Funded by the European Union

BEE WITH APEX

# E-Book über Low-Code-Entwicklung mit Oracle APEX

Praktischer Leitfaden für jeden zur Verbesserung der Arbeitsmarktfaehigkeit

**BeeAPEX Team**



Copyright © 2023 BeeAPEX Team

PUBLISHED BY SOFTWARE QUALITY AND TESTING LAB, UM FOV

BEEAPEX.EU

Lizenziert unter der Creative Commons Attribution-NonCommercial 4.0-Lizenz (die Lizenz"). Sie dürfen diese Datei nur in Übereinstimmung mit der Lizenz verwenden. Sie können eine Kopie der Lizenz unter https://creativecommons.org/licenses/by-nc-sa/4.0 erhalten. Sofern nicht durch geltendes Recht vorgeschrieben oder schriftlich vereinbart, wird die im Rahmen der Lizenz vertriebene Software auf einer "wie besehenBasis ohne Gewährleistungen oder Bedingungen jeglicher Art, weder ausdrücklich noch stillschweigend, vertrieben. Die spezifische Sprache, die die Berechtigungen und Einschränkungen im Rahmen der Lizenz regelt, finden Sie in der Lizenz.
*1st edition, 7. Dezember 2023*

# Inhaltsverzeichnis


Danksagung . . . . . 27

Vorwort . . . . . 29

I Anleitung in APEX

1 Wie wird Oracle APEX gestartet? . . . . . 35

Vjeran Strahonja, Dijana Oreški . . . . .

**1.1 Was ist Oracle APEX und wofür wird es eingesetzt? . . . . . 35**
1.1.1 Was ist Oracle APEX? . . . . . 35
1.1.2 Und wie sieht die Anwendung aus? . . . . . 35
1.1.3 Wie kann Oracle APEX bei der Anwendungsentwicklung helfen? . . . . . 36
1.1.4 Zyklus der Anwendungsentwicklung . . . . . 36
**1.2 Wie startet man Oracle APEX? . . . . . 43**
1.2.1 Wie hoch ist Ihr Kenntnisstand? . . . . . 43
1.2.2 APEX-Instanz vor Ort verwenden . . . . . 44
1.2.3 https://apex.oracle.com . . . . . 44
1.2.4 Virtual Box Appliance / Virtual Machine . . . . . 45
1.2.5 APEX docker . . . . . 46
1.2.6 APEX-Instanz in der Oracle Cloud-Infrastruktur . . . . . 48
1.2.7 APEX-Instanz in der Oracle Academy . . . . . 48
**1.3 Fragen . . . . . 48**
**1.4 Antworten . . . . . 49**

2 Wie wird eine Datenbank vorbereitet? . . . . . 51

E. Kapsammer, W. Retschitzegger, W. Schwinger . . . . .

**2.1 Wie man die Daten modelliert . . . . . 52**
2.1.1 Logisches Modell . . . . . 52
2.1.2 Relationales Modell . . . . . 56
2.1.3 Normalisierung eines RDB-Schemas . . . . . 61

**2.2 Wie man das RDBS-Schema verwaltet - SQL-DDL . . . 62**
2.2.1 Automatische Tabellengenerierung mit ODM . . . 63
2.2.2 Automatische Tabellengenerierung mit Quick SQL . . . 65
2.2.3 Manuelle Tabellenerstellung mit SQL-DDL . . . 67
2.2.4 Manuelle Tabellenverwaltung mit ORACLE Object Browser . . . 69
**2.3 Wie man Daten manipuliert - SQL-DML . . . 70**
2.3.1 Datenmanipulation mit SQL-DML . . . 70
2.3.2 Datenmanipulation mit dem ORACLE Object Browser ("OB") . . . 71
2.3.3 Datenmanipulation mit Quick SQL . . . 71
**2.4 Wie man Daten abfragt - SQL-DQL . . . 72**
2.4.1 Datenabfrage mit SQL-DQL . . . 72
2.4.2 Datenabfrage mit dem ORACLE Query Builder ("QB") . . . 77
**2.5 Aufbau der DB-Schicht - Das große Bild . . . 80**

**3 Wie navigiert man in APEX? . . . 81**
E. Kapsammer, W. Retschitzegger, W. Schwinger
**3.1 Webanwendungsentwicklungsprozess gefolgt von APEX . . . 81**
**3.2 Das Startmenü von APEX . . . 83**
**3.3 Übersicht über den App Builder - Erstellen und Verwalten Ihrer Apps . . . 83**
**3.4 Anwendung erstellen - Drei Anwendungsfälle . . . 84**
**3.5 Anwendung erstellen - Eigenschaften, Seiten, Funktionen und Einstellungen . . . 85**
**3.6 Seiten spezifizieren . . . 87**
**3.7 Pflegen und Ändern einer Seite – Page Designer . . . 89**
**3.8 Ausführen einer Anwendung oder Seite . . . 89**

**4 Wie man Daten mit APEX austauscht? . . . 93**
E. Kapsammer, W. Retschitzegger, W. Schwinger
**4.1 Importieren und Exportieren von Daten mit "Data Workshop" . . . 94**
4.1.1 Importieren von Daten . . . 94
4.1.2 Schritte zum Importieren . . . 96
4.1.3 Exportieren von Daten . . . 100
**4.2 Importieren und Exportieren von Daten mit dem "Object Browser" . . . 101**
**4.3 Exportieren des Ergebnisses eines SQL-Befehls . . . 102**
**4.4 Exportieren von Daten aus einem Anwendungsbericht . . . 103**
**4.5 Ermöglichung des Datenaustauschs mit RESTful Services . . . 104**
4.5.1 REST-Architektur-Muster . . . 105
4.5.2 Aktivieren eines DB-Schemas für den RESTful-Zugriff . . . 106
4.5.3 Ressourcenmodule | Templates | Handler . . . 107
4.5.4 Verwendung von "AutoREST" anstelle der manuellen Definition von Ressourcen 109

**5 Wie erstellt man den ersten Entwurf einer Anwendung? . . . 113**
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis
**5.1 Warum benötigen Unternehmen Anwendungen? . . . 113**

**5.2 Die Einrichtung der ORACLE APEX-Umgebung. . . . 113**
**5.3 Entwicklung von SQL-Workshop-Beispielen . . . 117**
5.3.1 Quick SQL . . . 120
5.3.2 SQL Skript . . . 121
**5.4 Datengesteuerter Teil der Anwendung . . . 124**
**5.5 Verwaltung der Anwendung . . . 127**
**5.6 Zugriffskontrolle . . . 127**
**5.7 Ergänzendes Lernmaterial . . . 133**
5.7.1 Exportierte Anwendungen . . . 133
5.7.2 Video-Anleitungen . . . 133
**5.8 Fragen . . . 133**
**5.9 Antworten . . . 134**

**6 Wie werden Berichte verwaltet? . . . 135**
Zuzana Žillová, Erik Malina, Matej Grochal, Andrej Staník, Andrea Meleková, Michal Kvet, Miroslav Potočár

**6.1 Bericht . . . 135**
**6.2 Klassischer Report . . . 145**
**6.3 Spaltenumschaltbericht . . . 146**
**6.4 Interaktiver Bericht . . . 146**
**6.5 Fragen . . . 155**
**6.6 Antworten . . . 156**

**7 Wie verwaltet man Formulare? . . . 157**
Veronika Šalgová, Michal Kvet, Miroslav Potočár; Übersetzer W. Schwinger

**7.1 Wie verwaltet man Formulare? . . . 157**
**7.2 Bearbeitbares Interaktives Raster . . . 157**
**7.3 Formular auf einer Tabelle . . . 160**
**7.4 Master-Detail-Formular . . . 161**
**7.5 Fragen . . . 164**
**7.6 Antworten . . . 164**

**8 Wie wandelt man Textberichte in Diagramme um? . . . 165**
Ivan Pastierik, Michal Kvet, Miroslav Potočár

**8.1 Diagramm . . . 165**
**8.2 Balkendiagramm erstellen . . . 166**
**8.3 Filterung zum Balkendiagramm hinzufügen . . . 170**
**8.4 Hinzufügen einer Sortierung zum Balkendiagramm . . . 175**
**8.5 Verschiedene Arten von Diagrammen erstellen . . . 179**
**8.6 Fragen . . . 181**
**8.7 Antworten . . . 182**

9 Wie verwaltet man Menüs? . . . 183
Veronika Šalgová, Michal Kvet, Miroslav Potočár; Übersetzer W. Schwinger . . .
9.1 Wie verwaltet man Menüs . . . 183
9.2 Side Menu . . . 183
9.3 Top Menu . . . 184
9.4 Mega Menu . . . 184
9.5 Bearbeiten von Menülisten . . . 185
9.6 Fragen . . . 186
9.7 Antworten . . . 186

10 Wie man im Team zusammenarbeitet? . . . 187
P. Staniszewski, M. Sońta, A. Kierzkowski . . .
10.1 Die kollaborative Wissensproduktion ist die Essenz der Low-Code-Entwicklung 187
10.2 Gemeinsam online sein . . . 187
10.3 Technischer Vorsprung durch Zusammenarbeit . . . 188
10.4 Beschreibung der Funktionen . . . 189
10.4.1 Seitensperre . . . 189
10.4.2 Kommentare . . . 190
10.4.3 Build-Optionen . . . 191
10.4.4 Team Development . . . 191
10.4.5 Feedback . . . 194
10.4.6 Standardisierung . . . 195
10.5 Schlussfolgerungen . . . 196
10.6 Fragen . . . 197
10.7 Antworten . . . 197

11 Wie kann man von der Galerie der Anwendungen und Plug-ins profitieren? . . . 199
Vjeran Strahonja, Dijana Oreški . . .
11.1 Wie installiert man Beispiel- und Starter-Apps? . . . 199
11.2 Starter-Apps . . . 200
11.3 Beispiel-Apps . . . 202
11.4 Plug-ins . . . 203
11.5 Fragen . . . 206
11.6 Antworten . . . 206

12 Wie verwaltet man verpackte und mehrsprachige Anwendungen? 209
R. Leskovar, U. Rajković, A.Baggia . . .
12.1 Anwendung und verpackte Anwendung . . . 209
12.2 Anwendung . . . 211
12.2.1 Umfang der Anwendung . . . 211
12.2.2 Tabellen erstellen . . . 211

12.2.3 Daten einfügen . . . 212
12.2.4 Anwendung generieren . . . 213
**12.3 Verpackte Anwendung . . . 217**
**12.4 Mehrsprachige Anwendung . . . 221**
**12.5 Ergänzendes Lernmaterial . . . 225**
12.5.1 Exportierte Anwendungen . . . 225
12.5.2 Video-Anleitungen . . . 225
**12.6 Fragen . . . 226**
**12.7 Antworten . . . 226**

II **Erstellen einer Anwendung in APEX**

**13 Intranet-Nachrichten für Mitarbeiter . . . 229**

R. Leskovar, U. Rajkovič, A.Baggia; Übersetzer W. Schwinger . . .

**13.1 Wirtschaftliche Sicht des Falles . . . 229**
**13.2 Problemstellung . . . 229**
**13.3 Anwendungsfälle . . . 230**
13.3.1 Narrative Beschreibung des Anwendungsfalls . . . 230
13.3.2 Semi-strukturierte Beschreibung . . . 230
13.3.3 Anwendungsfalldiagramm . . . 230
**13.4 Datenmodell . . . 234**
13.4.1 Narrative Beschreibung des Datenmodells . . . 234
13.4.2 Logisches Datenmodell . . . 234
13.4.3 Relationales Datenmodell . . . 235
**13.5 Anwendungsschnittstellen . . . 236**
**13.6 Ergänzendes Lernmaterial . . . 238**
13.6.1 Exportierte Anwendungen . . . 238
13.6.2 Video-Anleitungen . . . 238
**13.7 Fragen . . . 238**
**13.8 Antworten . . . 238**

**14 GreenDi - Pflanzenkatalog . . . 241**

Vjeran Strahonja, Dijana Oreški, Darko Andročec, Ana Kutnjak . . .

**14.1 Wirtschaftliche Sicht des Falles . . . 241**
**14.2 Problemstellung . . . 241**
**14.3 Anwendungsfälle . . . 242**
14.3.1 Narrative Beschreibung des Anwendungsfalls . . . 242
14.3.2 Semi-strukturierte Beschreibung . . . 242
14.3.3 Anwendungsfalldiagramm . . . 242
**14.4 Datenmodell . . . 244**
14.4.1 Narrative Beschreibung des Datenmodells . . . 244
14.4.2 Logisches Datenmodell . . . 244
14.4.3 Relationales Datenmodell . . . 246

**14.5 Anwendungsschnittstellen . . . 246**
**14.6 Ergänzendes Lernmaterial . . . 247**
14.6.1 Exportierte Anwendungen . . . 247
14.6.2 Video-Anleitungen . . . 248
**14.7 Fragen . . . 248**
**14.8 Antworten . . . 248**

**15 GreenDi - Benutzerautorisierung und -verwaltung . . . 251**
Vjeran Strahonja, Darko Andročec, Ana Kutnjak, Larisa Hrustek

**15.1 Wirtschaftliche Sicht des Falles . . . 251**
**15.2 Problemstellung . . . 251**
**15.3 Anwendungsfälle . . . 252**
15.3.1 Narrative Beschreibung des Anwendungsfalls . . . 252
15.3.2 Semi-strukturierte Beschreibung . . . 252
15.3.3 Anwendungsfalldiagramm . . . 252
**15.4 Data model . . . 252**
15.4.1 Narrative Beschreibung des Datenmodells . . . 252
15.4.2 Logisches Datenmodell . . . 254
15.4.3 Relationales Datenmodell . . . 254
**15.5 Anwendungsschnittstellen . . . 255**
**15.6 Ergänzendes Lernmaterial . . . 256**
15.6.1 Exportierte Anwendung . . . 256
15.6.2 Video-Anleitungen . . . 256
**15.7 Fragen . . . 256**
**15.8 Antworten . . . 256**

**16 Kleines Innovationssystem . . . 259**
R. Leskovar, U. Rajković, A.Baggia; Übersetzer W. Schwinger

**16.1 Wirtschaftliche Sicht des Falles . . . 259**
**16.2 Problemstellung . . . 259**
**16.3 Anwendungsfälle . . . 259**
16.3.1 Narrative Beschreibung des Anwendungsfalls . . . 259
16.3.2 Semi-strukturierte Beschreibung . . . 260
16.3.3 Anwendungsfalldiagramm . . . 260
**16.4 Datenmodell . . . 260**
16.4.1 Narrative Beschreibung des Datenmodells . . . 260
16.4.2 Logisches Datenmodell . . . 264
16.4.3 Relationales Datenmodell . . . 264
**16.5 Anwendungsschnittstellen . . . 265**
**16.6 Ergänzendes Lernmaterial . . . 268**
16.6.1 Exportierte Anwendungen . . . 268
16.6.2 Video-Anleitungen . . . 268
**16.7 Fragen . . . 268**
**16.8 Antworten . . . 269**

17 Management von Geschäftsprozessen ... 271
R. Leskovar, U. Rajković, A.Baggia; Übersetzer W. Schwinger ...
17.1 Wirtschaftliche Sicht des Falles ... 271
17.1.1 Flows für APEX installieren ... 274
17.1.2 Beispielhafte Prozessablaufanwendung installieren ... 274
17.1.3 Übungen lesen und üben ... 274
17.2 Problemstellung ... 275
17.3 Anwendungsfälle ... 276
17.3.1 Narrative Beschreibung ... 276
17.3.2 Semi-strukturierte Beschreibung ... 276
17.3.3 Anwendungsfalldiagramm ... 276
17.4 Workflow-Modell ... 278
17.5 Datenmodell ... 281
17.5.1 Narrative Beschreibung des Datenmodells ... 281
17.5.2 Logisches Datenmodell ... 281
17.5.3 Relationales Datenmodell ... 282
17.5.4 Quick SQL zur Erstellung von SQL-Skripten ... 282
17.5.5 Sequenz und zwei gespeicherte Funktionen ... 284
17.6 Anwendungsschnittstellen ... 285
17.6.1 Liste der Werte in Shared Components ... 285
17.6.2 Plug-ins in Shared Components ... 286
17.6.3 Verkaufsleiter ... 286
17.6.4 Produktionsleiter ... 291
17.6.5 Finanzverwalter ... 293
17.6.6 Hauptgeschäftsführer - Geschäftsleiter ... 295
17.7 Verknüpfung der Anwendung mit Flows for APEX ... 297
17.8 Benutzerrollen festlegen ... 299
17.9 Prüfung und Korrektur von Fehlern ... 299
17.10 Ergänzendes Lernmaterial ... 300
17.10.1 Exportierte Anwendungen ... 300
17.10.2 Video-Anleitungen ... 302
17.11 Fragen ... 302
17.12 Antworten ... 302
18 GreenDi - Tausch von Pflanzen und Saatgut ... 303
Vjeran Strahonja, Dijana Oreski, Darko Androcec, Ana Kutnjak ...
18.1 Wirtschaftliche Sicht des Falles ... 303
18.2 Problemstellung ... 303
18.3 Anwendungsfälle ... 304
18.3.1 Narrative Beschreibung des Anwendungsfalls ... 304
18.3.2 Semi-strukturierte Beschreibung ... 304
18.3.3 Anwendungsfalldiagramm ... 304
18.4 Datenmodell ... 304
18.4.1 Narrative Beschreibung des Datenmodells ... 306
18.4.2 Logisches Datenmodell ... 306
18.4.3 Relationales Datenmodell ... 306

**18.5 Anwendungsschnittstellen . . . 307**
**18.6 Ergänzendes Lernmaterial . . . 308**
18.6.1 Exportierte Anwendungen . . . 308
18.6.2 Video-Anleitungen . . . 308
**18.7 Fragen . . . 308**
**18.8 Antworten . . . 308**

**19 Verwaltungssystem für Buchrezensionen . . . 309**
A. Kutnjak, L. Hrustek, A. Baggia and R.Leskovar

**19.1 Wirtschaftliche Sicht des Falles . . . 309**
**19.2 Problemstellung . . . 309**
**19.3 Anwendungsfälle . . . 310**
19.3.1 Narrative Beschreibung des Anwendungsfalls . . . 310
19.3.2 Semi-strukturierte Beschreibung . . . 310
**19.4 Datenmodell . . . 312**
19.4.1 Narrative Beschreibung des Datenmodells . . . 312
19.4.2 Logisches Datenmodell . . . 313
19.4.3 Relationales Datenmodell . . . 313
19.4.4 SQL script . . . 314
19.4.5 Quick SQL . . . 315
**19.5 Anwendungsschnittstellen . . . 316**
19.5.1 Administrator . . . 316
19.5.2 Nutzer . . . 318
**19.6 Benutzerrollen definieren . . . 320**
**19.7 Ergänzendes Lernmaterial . . . 320**
19.7.1 Exportierte Anwendungen . . . 320
19.7.2 Video-Anleitungen . . . 321
**19.8 Fragen . . . 321**
**19.9 Antworten . . . 321**

**20 Stückliste und Kostenberechnung . . . 323**
R. Leskovar, U. Rajković, A.Baggia

**20.1 Wirtschaftliche Sicht des Falles . . . 323**
**20.2 Problemstellung . . . 323**
**20.3 Anwendungsfälle . . . 324**
20.3.1 Narrative Beschreibung . . . 324
20.3.2 Semi-strukturierte Beschreibung . . . 324
20.3.3 Anwendungsfalldiagramm . . . 324
**20.4 Data model . . . 324**
20.4.1 Narrative Beschreibung des Datenmodells . . . 324
20.4.2 Implementierung von Geschäftsregeln in die Datenbank . . . 328
20.4.3 Logisches Datenmodell . . . 329
20.4.4 Relationales Datenmodell . . . 329
20.4.5 Objekte in APEX . . . 330

**20.5 Anwendungsschnittstellen . . . 334**
**20.6 Ergänzendes Lernmaterial . . . 336**
20.6.1 Exportierte Anwendungen . . . 337
20.6.2 Video-Anleitungen . . . 337
**20.7 Fragen . . . 337**
**20.8 Antworten . . . 337**

**21 Ernährungs- und Diätmanagement . . . 339**
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . . .

**21.1 Wirtschaftliche Sicht des Falles . . . 339**
**21.2 Problemstellung . . . 339**
**21.3 Anwendungsfälle . . . 339**
21.3.1 Narrative Beschreibung des Anwendungsfalls . . . 339
21.3.2 Semi-strukturierte Beschreibung . . . 340
21.3.3 Anwendungsfalldiagramm . . . 340
**21.4 Datenmodell . . . 340**
21.4.1 Logisches Datenmodell . . . 340
21.4.2 Relationales Datenmodell . . . 342
21.4.3 QuickSQL und Script SQL Skript . . . 343
**21.5 Anwendungsschnittstellen . . . 346**
21.5.1 Wie man ein Dashboard erstellt . . . 350
21.5.2 Entwickler und Anwendungsnutzer . . . 354
**21.6 Ergänzendes Lernmaterial . . . 357**
21.6.1 Exportierte Anwendungen . . . 357
21.6.2 Video-Anleitungen . . . 357
**21.7 Fragen . . . 357**
**21.8 Antworten . . . 358**

**22 Sprechstundenplanung . . . 359**
J. Mańko, M. Sońta, R. Leskovar . . .

**22.1 Wirtschaftliche Sicht des Falles . . . 359**
**22.2 Problemstellung . . . 360**
**22.3 Anwendungsfälle . . . 360**
22.3.1 Narrative Beschreibung . . . 361
22.3.2 Semi-strukturierte Beschreibung . . . 361
22.3.3 Anwendungsfalldiagramm . . . 361
**22.4 Datenmodell . . . 364**
22.4.1 Narrative Beschreibung des Datenmodells . . . 364
22.4.2 Logisches Datenmodell . . . 365
22.4.3 Relationales Datenmodell . . . 365
22.4.4 Quick SQL . . . 366
22.4.5 SQL-Skript zum Erstellen von Tabellen . . . 367
22.4.6 Query builder in APEX . . . 367

**22.5 Anwendungsschnittstellen . . . 368**
22.5.1 Schnittstellen der Verwaltungsanwendungen . . . 368
22.5.2 Schnittstellen für Studentenbüros . . . 369
22.5.3 Schnittstellen für Studentenbewerbungen . . . 370
22.5.4 Bewerbungsschnittstellen für Lehrkräfte . . . 373
**22.6 Ergänzendes Lernmaterial . . . 375**
22.6.1 Exportierte Anwendungen . . . 376
22.6.2 Video-Anleitungen . . . 378
**22.7 Fragen . . . 378**
**22.8 Antworten . . . 378**

**23 Telco-Fall . . . 379**
Veronika Šalgová, Jozef Kostolný, Michal Mrena, Michal Kvet, Miroslav Potočár . . .
**23.1 Wirtschaftliche Sicht des Falles . . . 379**
**23.2 Problemstellung . . . 379**
**23.3 Anwendungsfälle . . . 379**
23.3.1 Narrative Beschreibung . . . 380
23.3.2 Semi-strukturierte Beschreibung . . . 380
23.3.3 Anwendungsfalldiagramm . . . 380
**23.4 Datenmodell . . . 382**
23.4.1 Narrative Beschreibung des Datenmodells . . . 382
23.4.2 Logisches Datenmodell . . . 382
23.4.3 Relationales Datenmodell . . . 383
**23.5 Benutzerauthentifizierung und Benutzerrollen . . . 383**
**23.6 Anwendungsschnittstellen . . . 386**
23.6.1 Anwendungsdesign . . . 387
**23.7 Skripte . . . 395**
**23.8 Erstellen einer Homepage . . . 397**
**23.9 Erstellen einer Kundenseite . . . 398**
**23.10 Erstellen einer Managerseite . . . 401**
**23.11 Ergänzendes Lernmaterial . . . 402**
23.11.1 Exportierte Anwendungen . . . 402
**23.12 Fragen . . . 402**
**23.13 Antworten . . . 403**

**24 Fall Autovermietung . . . 405**
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . . .
**24.1 Wirtschaftliche Sicht des Falles . . . 405**
**24.2 Problemstellung . . . 405**
**24.3 Anwendungsfälle . . . 406**
24.3.1 Narrative Beschreibung des Anwendungsfalls . . . 406
24.3.2 Semi-strukturierte Beschreibung . . . 406
24.3.3 Anwendungsfalldiagramm . . . 406

**24.4 Datenmodell . . . . . 408**
24.4.1 Narrative Beschreibung des Datenmodells . . . . . 408
24.4.2 Logisches Datenmodell . . . . . 409
24.4.3 Relationales Datenmodell . . . . . 409
24.4.4 SQL-Skript . . . . . 410
**24.5 Anwendungsschnittstellen . . . . . 411**
**24.6 Ergänzendes Lernmaterial . . . . . 416**
24.6.1 Exportierte Anwendungen . . . . . 416
24.6.2 Video-Anleitungen . . . . . 416
**24.7 Fragen . . . . . 416**
**24.8 Antworten . . . . . 416**

**Literatur . . . . . 417**
**Artikel . . . . . 417**
**Bücher . . . . . 417**

**Index . . . . . 419**

# Abbildungsverzeichnis


1.1 Zyklus der Anwendungsentwicklung. . . . . 37
1.2 Business Model Canvas (6) . . . . . 39
1.3 Die grundlegenden Konzepte des Anwendungsfalldiagramms. . . . . 40
1.4 Generalisierung/Spezialisierung von Akteuren. . . . . 41
1.5 Die Grenzen des Systems und Assoziationen „include“ und „extend“. . . . . 41
1.6 Diagramm der Anwendungsfälle des Hotelreservierungssystems. . . . . 42

2.1 DBS als Rückgrat von Webanwendungen. . . . . 51
2.2 Logisches Modell unseres laufenden HR-Beispiels mit Hilfe eines ER-Diagramms . 53
2.3 Verwaltung des logischen Modells mit ODM. . . . . 54
2.4 Attribute mit ODM verwalten. . . . . 55
2.5 Beziehungsmanagement mit ODM. . . . . 56
2.6 RDB-Schemata und Tabellenschemata . . . . . 57
2.7 Prozess zur RDB-Schema-Generierung aus dem logischen Modell. . . . . 58
2.8 Automatisch generiertes RDB-Schema. . . . . 58
2.9 GUI-Elemente zur Verwaltung von Spalten des RM. . . . . 60
2.10 GUI-Elemente für die Verwaltung des RM. . . . . 60
2.11 Normalisierungsschritte - Überblick . . . . . 61
2.12 Kategorien von SQL-Anweisungen . . . . . 63
2.13 Export des RDB-Schemas in ein DDL-Skript. . . . . 64
2.14 Hochladen und Ausführen eines SQL-Skripts . . . . . 65
2.15 Zugang zu Quick SQL über SQL Workshop . . . . . 66
2.16 Erstelltes SQL-Skript auf der Grundlage von Quick SQL. . . . . 67
2.17 SQL-Befehlseditor. . . . . 68
2.18 SQL-Anweisung zum Erstellen der Tabelle `Departments`. . . . . 68
2.19 SQL-Anweisung zum Löschen der Tabelle `Departments`. . . . . 69
2.20 SQL-Anweisung zum Ändern der Tabelle `Departments` . . . . . 69
2.21 Tabellenverwaltung mit dem ORACLE Object Browser . . . . . 69
2.22 SQL-Anweisung zum Einfügen neuer Daten in die Tabelle `Departments` . . . . . 70
2.23 SQL-Anweisung zum Aktualisieren des Gehalts aller Mitarbeiter um 10% in der Tabelle `Employees`. . . . . 70
2.24 SQL-Anweisung zur Aktualisierung des Gehalts von Mitarbeiter "Miller" . . . . . 70

2.25 SQL-Anweisung für die Aktualisierung von mehr als einem Attribut in der Tabelle Employees. . . . . 71
2.26 SQL-Anweisung zum Löschen aller Daten aus der Tabelle Employees. . . . . 71
2.27 SQL Statement for Deleting some Data from Table Employees. . . . . 71
2.28 Zufällige Dateneinfügung mit Quick SQL . . . . . 72
2.29 SQL-Anweisung zum Abrufen aller Daten aus der Tabelle Employees. . . . . 73
2.30 Auszug aus dem Abrufen aller Daten aus der Tabelle EMPLOYEES. . . . . 73
2.31 SQL-Anweisung zum Abrufen aller Daten aus der Tabelle Employees geordnet nach ihrem Gehalt. . . . . 73
2.32 Auszug aus dem Abrufen aller Daten aus der Tabelle Employees geordnet nach ihrem Gehalt . . . . . 74
2.33 SQL-Anweisung zum Abrufen bestimmter Daten aus der Tabelle Employees. . . . 74
2.34 Auszug aus dem Abrufen bestimmter Daten aus der Tabelle Employees. . . . . 74
2.35 SQL-Anweisung zum Abrufen bestimmter Daten aus der Tabelle Employees unter Verwendung der TO_CHAR()-Funktion. . . . . 75
2.36 Auszug aus dem Abrufen bestimmter Daten aus der Tabelle Mitarbeiter nur mit der TO_CHAR()-Funktion. . . . . 75
2.37 SQL-Anweisung zur Verknüpfung von Daten aus Tabelle Employees und Tabelle Departments. . . . . 75
2.38 Auszug aus dem Abrufen bestimmter Daten aus Tabelle Employees und Tabelle Departments. . . . . 76
2.39 SQL-Anweisung zum Zählen der Anzahl der Mitarbeiter und zum Bilden der Summe ihrer Gehälter aus der Tabelle Employees. . . . . 76
2.40 Auszug aus dem Abrufen der Anzahl der Mitarbeiter und Bildung der Summe ihrer Gehälter aus der Tabelle Employees. . . . . 76
2.41 SQL-Anweisung zum Gruppieren der Daten nach ihrem Gehalt und Ordnen des Ergebnisses nach der Anzahl der Mitarbeiter mit diesem Gehalt aus der Tabelle Employees. 77
2.42 Auszug aus dem Gruppieren der Daten nach ihrem Gehalt und Ordnen des Ergebnisses nach der Anzahl der Mitarbeiter mit diesem Gehalt aus der Tabelle Employees . 77
2.43 Überblick über den ORACLE Query Builder (“QB”). . . . . 78
2.44 Verwendung des Abfragegenerators zur Erstellung eines Join . . . . . 79
2.45 Übersicht über die Optionen zur Verwaltung der DB-Schicht. . . . . 80

3.1 Der gesamte Entwicklungsprozess in APEX. . . . . 82
3.2 Oracle APEX Workspace Homepage. . . . . 83
3.3 Überblick über den App Builder. . . . . 84
3.4 Create Application Wizard: Drei Anwendungsfälle beim Erstellen einer Anwendung. 85
3.5 Eigenschaften und Seiten. . . . . 86
3.6 Funktionen und Einstellungen. . . . . 86
3.7 Anwendungshomepage - Entwickleransicht. . . . . 87
3.8 Erstellen einer Seite. . . . . 88
3.9 Seitendesigner. . . . . 89
3.10 Gesamte Anwendung ausführen. . . . . 90
3.11 Einzelne Seiten ausführen. . . . . 91

4.1 Datenaustauschoptionen im APEX. . . . . 94
4.2 Verwendung des Data Workshops zum Laden von Daten - Zugriff über den SQL Workshop. . . . . 95
4.3 Verwendung des Data Workshops zum Laden von Daten - Zugriff über den App Builder. . . . . 96
4.4 Bereitstellung der Datenquelle. . . . . 97

4.5 Laden von Daten in eine neue Tabelle. . . . . 97
4.6 Laden von Daten in eine bestehende Tabelle. . . . . 99
4.7 Exportieren ("Entladen") von Daten. . . . . 100
4.8 Datenentlade-Assistent. . . . . 101
4.9 Import/Export von Tabellendaten mit dem Object Browser. . . . . 102
4.10 Datenexport des Ergebnisses einer SQL-Abfrage. . . . . 103
4.11 Export von Daten aus einem Anwendungsbericht. . . . . 104
4.12 RESTful Services für den Datenaustausch - Grundlegende Architektur. . . . . 105
4.13 Aktivieren eines DB-Schemas für den RESTful-Zugriff. . . . . 107
4.14 Beziehungen zwischen den verschiedenen Komponenten der ORDS RESTful Services. . . . . 108
4.15 Beispiel RESTful Services für die Tabelle Employees. . . . . 109
4.16 Aktivieren des REST-Dienstes im Objektbrowser. . . . . 110

5.1 Geben Sie den APEX-Link ein und melden Sie sich an. . . . . 114
5.2 Anforderung der ersten Schritte für einen Oracle APEX-Arbeitsbereich. . . . . 114
5.3 Fertigstellung der Anfrage eines Oracle APEX-Workspace. . . . . 115
5.4 Oracle APEX Workspace genehmigt. . . . . 115
5.5 Aktivierungs-E-Mail von APEX. . . . . 116
5.6 Richten Sie ein neues Passwort für Workspace APEX ein. . . . . 116
5.7 Arbeitsbereich APEX-Umgebung. . . . . 117
5.8 Logisches Modell für unser laufendes HR-Beispiel. . . . . 118
5.9 Relationales Modell für unser laufendes HR-Beispiel. . . . . 119
5.10 Quick SQL-Code in den APEX-Arbeitsbereich einfügen. . . . . 121
5.11 SQL-Skript ausführen. . . . . 123
5.12 App-Startprozess erstellen. . . . . 123
5.13 Anwendung aus Skript erstellen. . . . . 124
5.14 Webanwendung erstellt. . . . . 125
5.15 App-Builder-Umgebung. . . . . 125
5.16 App-Anmeldeseite. . . . . 126
5.17 Ihre neue App-Umgebung. . . . . 126
5.18 Verwaltung der Benutzer- und Gruppenumgebung. . . . . 127
5.19 Beispiel für eine Zugriffskontrolle. . . . . 128
5.20 Neue Benutzer hinzufügen. . . . . 129
5.21 Rollen für Benutzer hinzufügen. . . . . 130
5.22 Schaltflächenkonfiguration erstellen. . . . . 131
5.23 Contributor access and rights. . . . . 132
5.24 Readerzugang und Rechte. . . . . 133

6.1 Auswahl des Seitentyps. . . . . 136
6.2 Erstellung einer Seite mit Bericht. . . . . 137
6.3 Rendering. . . . . 137
6.4 Eigenschaften einer Region. . . . . 138
6.5 Verschiedene Datenquellen. . . . . 138
6.6 Layout. . . . . 139
6.7 Erscheinungsbild. . . . . 139
6.8 Vorlagen. . . . . 140
6.9 Erweitert. . . . . 140
6.10 Kopf- und Fußzeile. . . . . 140
6.11 Server-side Condition. . . . . 141

6.12 Read Only. . . . . 141
6.13 Security, Server Cache und Customization. . . . . 142
6.14 Seitennummerierung. . . . . 142
6.15 Anzahl der zu ladenden Zeilen. . . . . 142
6.16 Herunterladen und Drucken. . . . . 143
6.17 Seitenzentrums-Designer. . . . . 143
6.18 Type of column. . . . . 144
6.19 Spalteneinstellungen. . . . . 144
6.20 Format Mask. . . . . 144
6.21 Personalisierung. . . . . 145
6.22 Option zum Sortieren nach Spalten. . . . . 145
6.23 Position des grünen Pfeils für die Seitenpräsentation. . . . . 145
6.24 Beispiel für einen klassischen Bericht. . . . . 146
6.25 Möglichkeit zur Änderung der Sichtbarkeit von Spalten im Spaltenumschaltbericht. . . . . 146
6.26 Interaktiver Bericht. . . . . 147
6.27 Suchfeld. . . . . 147
6.28 Suche. . . . . 148
6.29 Aktionen-Schaltfläche. . . . . 148
6.30 Auswahl der anzuzeigenden Spalten. . . . . 149
6.31 Spalten-Filter. . . . . 149
6.32 Zeilen-Filter. . . . . 150
6.33 Nach Anklicken der Schaltfläche Data. . . . . 150
6.34 Sortieren. . . . . 151
6.35 Aggregate. . . . . 151
6.36 Flashback. . . . . 151
6.37 Control Break. . . . . 152
6.38 Control Break-Resultat. . . . . 152
6.39 Highlight. . . . . 152
6.40 Highlight-Resultat. . . . . 153
6.41 Row Per Page. . . . . 153
6.42 Diagramm im interaktiven Bericht. . . . . 154
6.43 Group By. . . . . 154
6.44 Save Report. . . . . 155
6.45 Herunterladen. . . . . 155

7.1 Interaktives Raster mit Kunden . . . . . 157
7.2 Aktionen des interaktiven Rasters . . . . . 158
7.3 Ausblenden-Symbol der Spalte Telefon . . . . . 158
7.4 Angezeigte Spalten . . . . . 159
7.5 Freeze-Symbol der Adressspalte . . . . . 159
7.6 Sortieren der Adressspalte . . . . . 159
7.7 Aggregate-Symbol der Adressspalte . . . . . 160
7.8 Arten von Diagrammen . . . . . 160
7.9 Formular zum Einfügen von Kunden . . . . . 161
7.10 Gestapeltes Master-Detail-Formular der Kundenpauschalen . . . . . 162
7.11 Side by Side Master-Detail-Formular der Kundenpauschalen . . . . . 162
7.12 Drill Down Master-Detail Formular – erste Seite . . . . . 163
7.13 Drill Down Master-Detail Formular – zweite Seite . . . . . 164

8.1 In der Beispielanwendung verwendetes Datenmodell. . . . . 165
8.2 Selecting and placing Chart region. . . . . 166
8.3 Layout des Hauptteils nach der Platzierung des Diagrammbereichs. . . . . 166
8.4 Baumübersicht mit Fehler in Diagrammreihen. . . . . 167
8.5 Endgültige Konfiguration der Verkaufsreihe . . . . . 168
8.6 Durch die Ausführung der SQL-Abfrage zurückgegebene Daten. . . . . 168
8.7 Beim Starten der Anwendung angezeigtes Diagramm. . . . . 169
8.8 Ändern des Titelattributs für die X-Achse. . . . . 169
8.9 Umsatzdiagramm nach Beschriftung der Achsen. . . . . 170
8.10 Auswählen und Platzieren von Checkbox-Gruppen-Elementen. . . . . 170
8.11 Einstellen der grundlegenden Attribute des Checkbox Group Elements. . . . . 171
8.12 Einstellen der Werteliste für das Checkbox Group Element. . . . . 171
8.13 Einstellen von Standardwerten für das Checkbox Group Element. . . . . 172
8.14 Verknüpfung der Verkaufsdiagrammregion mit dem Checkbox Group Element. 173
8.15 Dynamische Aktion für den Artikel P1_PRODUCTS_CHOICE erstellen. . . . . 173
8.16 Einstellung grundlegender Attribute der dynamischen Aktion bei Änderung. . 174
8.17 Einstellung der Attribute der Aktion Aktualisieren. . . . . 174
8.18 Dynamische Aktion zum Aktualisieren der Verkaufstabelle in der Baumübersicht. 175
8.19 Anwendung nach Implementierung der Produktfilterung. . . . . 175
8.20 Auswählen und Platzieren von Elementen der Auswahlliste. . . . . 176
8.21 Sales Chart Region nach dem Platzieren und Umbenennen eines Elements der Auswahlliste. . . . . 176
8.22 Einstellen der Werteliste für die Auswahlliste. . . . . 177
8.23 Einstellung statischer Werte für die Auswahlliste. . . . . 177
8.24 Hinzufügen einer dynamischen Aktion zur Auswahlliste. . . . . 177
8.25 Einstellen der Reihenfolge nach Artikel für die Sales Chart Region. . . . . 178
8.26 Festlegen von Sortierklauseln für Auswahllistenelemente, die für die Sortierung verwendet werden. . . . . 178
8.27 Anwendung nach Implementierung der Diagrammsortierung. . . . . 179
8.28 Ändern des Diagrammtyps in Line with Area. . . . . 180
8.29 Liste der Werte für die Auswahlliste, die zum Filtern der Produktverkäufe nach Jahr verwendet wird. . . . . 180
8.30 Einstellen der SQL-Quellabfrage und der Spaltenzuordnung für die Monthly Sales Chart Region. . . . . 181
8.31 Monthly Sales Chart. . . . . 181

9.1 Erweitertes seitliches Navigationsmenü . . . . . 184
9.2 Top Navigation Tabs . . . . . 184
9.3 Top Navigation Menu . . . . . 184
9.4 Top Navigation Mega Menu . . . . . 185
9.5 Attribute des Navigationsmenüs . . . . . 185

10.1 Seitensperrfunktion in APEX. . . . . 189
10.2 Sperren einer bestimmten Seite. . . . . 190
10.3 Hinzufügen von Kommentaren. . . . . 190
10.4 Build-Optionen. . . . . 191
10.5 Team Development-Funktion. . . . . 192
10.6 Aktivierung der Team Development-Funktion. . . . . 192
10.7 Anpassung der Team Development-Funktion. . . . . 193
10.8 Issue-bezogene Kommunikation innerhalb der Team Development-Funktion. . 193

10.9 Feedback in der Anwendungsdefinition zulassen. . . . . 194
10.10 Feedback in der Anwendung einreichen. . . . . 195
10.11 Utilities-Menü. . . . . 196

11.1 Oracle APEX-Workspaces werden mit Sample Apps und Starter Apps angeboten. 199
11.2 Starter-Apps in der Galerie. . . . . 200
11.3 Installation der Starter-App Team Calendar. . . . . 201
11.4 Beispiel-Apps. . . . . 202

12.1 Assistent für den Import und Export von Anwendungen. . . . . 210
12.2 Umwandlung von Quick SQL in SQL-Befehle. . . . . 212
12.3 Skript CH12INSERT Daten in drei Tabellen einfügen und Transaktionen festlegen. 212
12.4 Anwendung aus Skript erstellen. . . . . 213
12.5 Auswahl des Namens der Anwendung und aller Funktionen. . . . . 213
12.6 Shared Components auswählen. . . . . 214
12.7 List of Values von Grund auf neu erstellen. . . . . 214
12.8 Name und Typ festlegen: CH12_LOV_COMPETENCE_DESCRIPTION ist dynamisch. 215
12.9 Eingabe des SQL SELECT-Befehls. . . . . 215
12.10 Name und Typ definieren: CH12_LOV_COMPETENCE_LEVEL ist statisch. . . . . 216
12.11 Anzeige- und Rückgabewerte definieren: CH12_LOV_COMPETENCE_LEVEL ist statisch. . . . . 216
12.12 Wählen Sie Seite 7 (Ch12 Job Competences). . . . . 217
12.13 Seite 7 auswählen (Ch12 Job Competences). . . . . 217
12.14 Supporting Objects auswählen. . . . . 218
12.15 Voraussetzungen festlegen. . . . . 218
12.16 Prüfung auf Existenz von drei Tabellen einstellen. . . . . 218
12.17 Aufforderung zum Umbenennen der Anwendung einstellen. . . . . 219
12.18 Installationsskripte festlegen. . . . . 219
12.19 Deinstallationsskripte festlegen. . . . . 219
12.20 Anwendung in anderen Arbeitsbereich importieren. . . . . 220
12.21 Lizenzvereinbarung. . . . . 220
12.22 Importierte Anwendung umbenennen. . . . . 221
12.23 Hinzufügen einer Rolle zu einem Benutzer. . . . . 221
12.24 Definierte Sprachen für die Übersetzung. . . . . 222
12.25 Seed translatable text. . . . . 222
12.26 Zeichenfolgen für eine bestimmte Sprache und Seite exportieren. . . . . 223
12.27 Übersetzung der mit targetmarkierten Zeichenfolgen in den Zeilen 48, 52, 56, 60, 72 und 76. . . . . 223
12.28 Hochladen von XLDIFF-Übersetzungsdateien. . . . . 223
12.29 Änderungen übernehmen und veröffentlichen. . . . . 224
12.30 Endgültige Veröffentlichung der Übersetzung der Anwendung. . . . . 224
12.31 Einstellung der Primärsprache der Anwendung. . . . . 224
12.32 Übersetzte Seite. . . . . 224

13.1 Anwendungsfalldiagramm. . . . . 234
13.2 Logisches Datenmodell. . . . . 235
13.3 Relationales Datenmodell. . . . . 235
13.4 Das Dashboard der Mitwirkenden (Herausgeber). . . . . 236
13.5 Nachrichten mit Anhängen veröffentlichen. . . . . 236

13.6 Betrachter (Leser) Dashboard. . . . . 237
13.7 Betrachter (Leser) Zugriff auf Nachrichten. . . . . 237
13.8 Dashboard für Administratoren. . . . . 237
13.9 Bearbeitung der Mitarbeiterrolle durch den Administrator. . . . . 238

14.1 Anwendungsfalldiagramm - Pflanzenkatalog. . . . . 244
14.2 Logisches Datenmodell. . . . . 245
14.3 Relationales Datenmodell. . . . . 246
14.4 Interaktives Gitter für Plant Form. . . . . 247
14.5 Öffentlicher Teil - offenes Browsing von Pflanzendaten. . . . . 247

15.1 Anwendungsfalldiagramm - GreenDi Benutzerautorisierung und -verwaltung . 252
15.2 Logisches Datenmodell. . . . . 254
15.3 Relationales Datenmodell. . . . . 255
15.4 Benutzerformular. . . . . 255
15.5 Formular für die Benutzerhistorie. . . . . 256

16.1 Anwendungsfalldiagramm. . . . . 260
16.2 Logisches Datenmodell. . . . . 264
16.3 Relationales Datenmodell. . . . . 264
16.4 Das Mitarbeiter-Dashboard. . . . . 265
16.5 Idee mit Anhängen einreichen. . . . . 266
16.6 Organisatorische Struktur im Unternehmen. . . . . 266
16.7 Kleine Übersicht über Innovationsideen. . . . . 267
16.8 Prüfer Dashboard. . . . . 267
16.9 Seite zum Prüfen der Ideen. . . . . 268

17.1 Integration von Flows for APEX in die APEX-Anwendung. . . . . 273
17.2 Anwendungsfalldiagramm. . . . . 278
17.3 Workflow - Bearbeitungsanfrage - BPMN-Diagramm. . . . . 279
17.4 Erstellen von Instanzen. . . . . 279
17.5 Erzeugen einer Instanz des Flows CH17. . . . . 280
17.6 Demo-Instanz starten. . . . . 280
17.7 Schließen Sie die erste Aufgabe in der Demo-Instanz ab. . . . . 280
17.8 Ausführen von Aufgaben in parallelen Gates. . . . . 281
17.9 Abgeschlossene Demo-Instanz. . . . . 281
17.10 Logisches Datenmodell. . . . . 282
17.11 Relationales Datenmodell. . . . . 282
17.12 Startseite der Anwendung „CH17 Business Process Management“. . . . . 285
17.13 Verkäufe - initiierter Prozess. . . . . 287
17.14 Umsatz - Flow report. . . . . 287
17.15 Umsatz - Flow report für ausgewählte Instanz. . . . . 288
17.16 Verkäufe - Flow report für ausgewählte Instanz. . . . . 288
17.17 Verkauf - Anfragebericht. . . . . 289
17.18 Verkäufe - Liste der Dokumente. . . . . 289
17.19 Verkauf - Dokument für Anfrage 17 hochladen. . . . . 290
17.20 Verkäufe - Liste der Dokumente nach dem Hochladen und Aktualisieren. . . 290
17.21 Umsatz - Flow report für ausgewählte Instanz nach dem Hochladen von drei Belegen. . . . . 291

17.22 Umsatz - Flow report für ausgewählte Instanz nach dem Hochladen von drei Belegen. . . . . 291
17.23 Fertigung - Flow report. . . . . 292
17.24 Fertigung - BPMN-Diagramm - Zustand der Instanz. . . . . 292
17.25 Bewertung der Herstellung. . . . . 292
17.26 Fertigungsbewertung abgeschlossen, Instanz wartet auf die finanzielle Bewertung. . . . . 293
17.27 Finanzen - Flow-Report. . . . . 293
17.28 Finanzen - BPMN-Diagramm - Zustand der Instanz. . . . . 294
17.29 Finanzielle Bewertung. . . . . 294
17.30 Finanzielle Bewertung abgeschlossen, Instanz wartet auf die Unternehmensbewertung. . . . . 295
17.31 Finanzen - BPMN-Diagramm - Zustand der Instanz. . . . . 295
17.32 Business - Flow-Report. . . . . 296
17.33 Unternehmensbewertung. . . . . 296
17.34 Unternehmensbewertung abgeschlossen, Instanz beendet. . . . . 297
17.35 Zeigt den Zustand der Instanz im BPMN-Diagramm an. . . . . 298
17.36 Einstellen von Seitenelementen. . . . . 298
17.37 Schritt in Flows for Apex abschließen. . . . . 299
17.38 Definieren Sie Rollen und Benutzerrollen im Menü Application Access Control. 299
17.39 Erteilung des „Privilegs zum Erstellen von Aufträgen“ an den Arbeitsbereich. 300
17.40 Importieren Sie den Workflow namens CH17 in Flows for APEX - Schritt 1. . . . . 301
17.41 Importieren Sie den Workflow namens CH17 in Flows for APEX - Schritt 2. . . . . 301

18.1 Anwendungsfalldiagramm - Austausch von Pflanzen und Saatgut . . . . . 304
18.2 Logisches Datenmodell. . . . . 306
18.3 Relationales Datenmodell. . . . . 307
18.4 Angebote - interaktiver Bericht. . . . . 307
18.5 Formular für Nachrichten. . . . . 307

19.1 Anwendungsfalldiagramme: Administrator fügt neuen Kommentar hinzu. . . . 312
19.2 Logisches Datenmodell. . . . . 313
19.3 Relationales Datenmodell. . . . . 313
19.4 Erzeugen einer Anwendung aus dem Skript CH19CREATEINSERT - Teil 1. . . . . 316
19.5 Erzeugen einer Anwendung aus dem Skript CH19CREATEINSERT - Teil 2. . . . . 317
19.6 Hinzufügen eines Buches durch den Administrator. . . . . 318
19.7 Hinzufügen einer Kategorie durch den Administrator. . . . . 318
19.8 Benutzerregistrierung. . . . . 319
19.9 Durchsuchen und Hinzufügen von Bewertungen. . . . . 319
19.10 Formular zum Kommentieren einer Bewertung . . . . . 320

20.1 Anwendungsfalldiagramm. . . . . 324
20.2 Logisches Datenmodell. . . . . 329
20.3 Logisches Datenmodell. . . . . 329
20.4 Logisches Datenmodell. . . . . 329
20.5 Logisches Datenmodell. . . . . 331
20.6 Startseite der Anwendung. . . . . 334
20.7 Verwaltung von Basisdaten - Bericht. . . . . 335
20.8 Verwaltung von Basisdaten - Formular. . . . . 335
20.9 Verwaltung von Strukturdaten - Bericht. . . . . 335

20.10 Verwaltung von Strukturdaten - Formular. . . . . 336
20.11 Seite zur Berechnung der Stückliste. . . . . 336

21.1 Anwendungsfalldiagramm. . . . . 340
21.2 Logisches Datenmodell. . . . . 342
21.3 Relationales Datenmodell. . . . . 343
21.4 Quick-SQL-Tool zur Erstellung von SQL-Skripten. . . . . 344
21.5 SQL-Skript-Prozedur zur Erstellung von SEQUENCE. . . . . 346
21.6 Neue Anwendung erstellen. . . . . 347
21.7 Ernährungsanwendung. . . . . 348
21.8 Ernährungsanwendung - Rezeptseite. . . . . 349
21.9 Ernährungsanwendung - Zutaten-Seite. . . . . 349
21.10 Ernährungsanwendung - Rezept-Zutaten-Seite. . . . . 350
21.11 Ernährungsanwendung - Nährwerttabellen-Seite. . . . . 350
21.12 Erstellen Sie eine Dashboard-Seite für Ihre Anwendung. . . . . 351
21.13 Konfiguration von Titel und Chat-Typ in Dashboard-Berichten. . . . . 352
21.14 SOURCE mit SQL QUERY konfigurieren. . . . . 354
21.15 Erstellung eines neuen Benutzerkontos. . . . . 355
21.16 Benutzerkonto in der Anwendung erstellen. . . . . 356
21.17 Seiten mit Benutzerrechten konfigurieren. . . . . 357

22.1 Anwendungsfalldiagramm. . . . . 364
22.2 Logisches Datenmodell. . . . . 365
22.3 Relationales Datenmodell. . . . . 366
22.4 Datenmodell mit Quick SQL beschrieben. . . . . 366
22.5 Ein Beispiel für die Verwendung des Query Builders. . . . . 367
22.6 Startseite der Anwendung. . . . . 368
22.7 Management dashboard. . . . . 369
22.8 Interaktiver Bericht über die Bürozeiten für das Studentenbüro. . . . . 369
22.9 Umplanungsformular für das Studentenbüro. . . . . 370
22.10 Interaktiver Bericht über die Bürozeiten nach der Umplanung. . . . . 370
22.11 Terminkalender für Studenten. . . . . 371
22.12 Kalender aller Sprechstunden, die von den Lehrkräften angeboten werden, die am Studiengang des Studenten beteiligt sind. . . . . 371
22.13 Eintragung in die Sprechzeiten über den Kalender - Auswahl des Kalenderplatzes. . . . . 371
22.14 Eintragung in die Sprechzeiten über den Kalender - Auswahl des Zwecks. . . 372
22.15 Eintragung in die Sprechstunde über den Kalender - erfolgreiche Eintragung. 372
22.16 Überprüfung der Einschreibung über den Menüpunkt „Viev my appointments“. 372
22.17 Detaillierter interaktiver Bericht über alle Termine der Studenten. . . . . 373
22.18 Detaillierter Bericht über die Termine von Lehrern . . . . . 373
22.19 Lehrerkalender mit sichtbaren Schülernamen, ihren Sprachen und Zielen. . . 374
22.20 Neuplanung von Lehrersprechstunden mit NONE eingeschrieben - die Kalenderansicht. . . . . 374
22.21 Umplanung der Lehrersprechstunde mit NONE eingeschrieben - neues Datum eingegeben. . . . . 375
22.22 Die Ergebnisse der Neuplanung von Lehrersprechstunden, für die NONE eingeschrieben ist. . . . . 375

23.1 Anwendungsfalldiagramm. . . . . 382

23.2 Logical data model. . . . . 383
23.3 Relationales Datenmodell. . . . . 383
23.4 Kunden-Dashboard. . . . . 388
23.5 Kunden-Dashboard - Modales Fenster für Rechnungen. . . . . 389
23.6 Kunden-Dashboard - Rechnung im PDF-Format. . . . . 389
23.7 Kunden-Dashboard - Statistik der Minuten. . . . . 389
23.8 Kunden-Dashboard - Statistik der SMS. . . . . 390
23.9 Kunden-Dashboard - Statistik der Daten. . . . . 390
23.10 Manager-Dashboard - Kunde hinzufügen. . . . . 391
23.11 Manager-Dashboard - Verwalten von Kunden. . . . . 392
23.12 Manager-Dashboard - Kunden anzeigen. . . . . 392
23.13 Manager-Dashboard - Export von Kunden. . . . . 393
23.14 Manager-Dashboard - Kundenexport in XLS-Datei. . . . . 393
23.15 Administrator-Dashboard. . . . . 394
23.16 Eine Landing Page mit Login. . . . . 394
23.17 Login-Seite. . . . . 394
23.18 Bereiche des Bodies. . . . . 397
23.19 Liste der Inhalte. . . . . 397
23.20 HTML-Code eines statischen Bereichs. . . . . 398

24.1 Anwendungsfalldiagramm. . . . . 408
24.2 Logisches Modell des Autovermietungsprojekts . . . . . 409
24.3 Relationales Modell des Autovermietungsprojekts . . . . . 410
24.4 In der App anmelden. . . . . 412
24.5 Startseite der Anwendung. . . . . 412
24.6 Vorlage für Autos . . . . . 413
24.7 Wie man eine Kolumne mit Fotos erstellt. . . . . 414
24.8 Kundendaten. . . . . 415
24.9 Vorlage für die Autovermietung. . . . . 415



# Tabellenverzeichnis


13.1 c: Veröffentlichung von Internet-Nachrichten. . . . . 231
13.2 Beschreibung des Anwendungsfalls: Lesen von Intranet-Nachrichten . . . . . 232
13.3 Beschreibung des Anwendungsfalls: Verwaltung des Intranetportals. . . . . 233

14.1 Beschreibung des Anwendungsfalls: Durchsuchen des Pflanzenkatalogs . . . . 243

15.1 Beschreibung des Anwendungsfalls: Benutzerautorisierung und -verwaltung. . 253

16.1 Beschreibung des Anwendungsfalls: Einreichung von Ideen. . . . . 261
16.2 Beschreibung des Anwendungsfalls: Überblick über die Ideen. . . . . 262
16.3 Beschreibung des Anwendungsfalls: Bewertung von Ideen. . . . . 263

17.1 Beschreibung des Anwendungsfalls: Vorbereitung der Untersuchungsunterlagen. 276
17.2 Beschreibung des Anwendungsfalls: Fertigungsaspekte der Untersuchung bewerten. . . . . 277
17.3 Beschreibung des Anwendungsfalls: Bewertung der finanziellen Aspekte der Untersuchung. . . . . 277
17.4 Beschreibung des Anwendungsfalls: Bewertung der wirtschaftlichen Aspekte der Untersuchung. . . . . 278

18.1 Beschreibung des Anwendungsfalls: Austausch von Pflanzen und Saatgut. . . . 305

19.1 Beschreibung des Anwendungsfalls: Buchrezensionen Analyse . . . . . 311

20.1 Beschreibung des Anwendungsfalls: Bericht und Pflege von Basisdaten. . . . . 325
20.2 Beschreibung des Anwendungsfalls: Strukturdaten melden und pflegen. . . . . 326
20.3 Beschreibung des Anwendungsfalls: BOM berechnen. . . . . 327

21.1 Beschreibung des Anwendungsfalls: Zugang zu Rezepten, Zutaten und Nährwertberechnungen . . . . . 341

22.1 Beschreibung des Anwendungsfalls: Verlegung der Sprechstunden durch den Lehrer. . . . . 362
22.2 Beschreibung des Anwendungsfalls: Einschreibung von Studenten für Sprechstunden. . . . . 363

23.1 Beschreibung des Anwendungsfalls: Dienst hinzufügen. . . . . 381
23.2 Beschreibung des Anwendungsfalls: Status der Dienste anzeigen. . . . . 381
23.3 In der Tabelle CH23_Person gespeicherte Beispieldaten. . . . . 385

24.1 Beschreibung des Anwendungsfalls: Zugang zu Fahrzeugen, Kunden und Mietwagenreservierung . . . . . 407

# Acknowledgement

Die Projektpartners und das BEE APEX Team:

- University of Maribor Faculty of Organizational Sciences,
- University of Zagreb Faculty of Organization and Informatics,
- University of Žilina,
- Kozminski University,
- International Hellenic University and
- Johannes Kepler University

möchten sich für die finanzielle Unterstützung der European Commission durch die Förderaktion Erasmus + Better Employability for Everyone with APEX (project ID 2021-1-SI01-KA220-HED-000032218), finanziert durch das Erasmus+ Programm der European Union bedanken.

Die Unterstützung der Europäischen Kommission für die Erstellung dieser Veröffentlichung stellt keine Billigung des Inhalts dar, der ausschließlich die Ansichten der Autoren widerspiegelt, und die Kommission kann nicht für die Verwendung der darin enthaltenen Informationen verantwortlich gemacht werden.

*Danke an:*

- Oracle Academy *Program Manager Mr. Darko Jureković* für seine kontinuierliche Unterstützung bei der Verbreitung der Projektergebnisse und
- THE RIGHT THING SOLUTIONS *CEO Mr. Aljaž Mali* für seine wertvollen Ratschläge zu APEX vor und während des Projekts.

*Das BeeAPEX Projectteam*



# Vorwort

Willkommen zur Erkundung von Oracle Application Express (APEX) – einer intuitiven und leistungsstarken Low-Code-Entwicklungsplattform für die Erstellung datengesteuerter Webanwendungen. Dieses Lehrbuch soll Sie mit den Kompetenzen ausstatten, die Sie benötigen, um das volle Potenzial von Oracle APEX auszuschöpfen und hochmoderne Anwendungen zu erstellen, um reale geschäftliche Herausforderungen zu meistern.

Teil I dieses Lehrbuchs ist “How to” und widmet sich den grundlegenden Aspekten von Oracle APEX. In diesen zwölf Kapiteln begeben Sie sich auf eine Reise, um die Kernkonzepte zu verstehen, Zugang zu APEX zu erhalten und verschiedene Funktionalitäten zum Erstellen robuster Anwendungen zu erkunden. Jedes Kapitel befasst sich mit einem bestimmten Thema, bietet klare Anweisungen und erleichtert praktische Lernerfahrungen.

Kapitel 1: “Wie wird Oracle APEX gestartet“ erklärt, was es ist, wozu es dient und beschreibt verschiedene Möglichkeiten, die APEX-Umgebung vorzubereiten, um damit eigene praktische Erfahrungen zu sammeln und eigene Anwendungsentwicklung durchführen zu können.

Kapitel 2:“Wie bereite ich eine Datenbank vor¿‘ bietet eine Einführung in die Datenmodellierung, die Verwaltung von Datenbanken, die Bearbeitung von Daten und das Abfragen von Daten. Für Anfänger ist ein umfassender Überblick über Datenbankkonzepte ein Muss.

Kapitel 3: “Wie navigiert man in APEX¿‘ bietet einen Überblick über die APEX-Funktionen, die die Anwendungsentwicklung, Generierung und Anpassung verschiedener Webseiten und ihrer Komponenten ermöglichen. Das Ausführen und Testen der APEX-Anwendung ist nur einen Schritt von der Entwicklungsumgebung entfernt.

Kapitel 4: “Wie man Daten mit APEX austauscht¿‘ gibt Einblick in den Import und Export von Daten in APEX. Das Kapitel behandelt den Datenaustausch mit Dateien wie beispielsweise Tabellenkalkulationen sowie über RESTful-Dienste.

Kapitel 5: “Wie erstellt man den ersten Entwurf einer Anwendung¿‘ lädt Sie ein, die Entwicklungsleistung von APEX auszuloten. Sie werden feststellen, dass Sie, sobald die notwendigen Daten zu Verfügung stehen, ohne Programmierung sofort ansprechende und funktionale Anwendungen erstellen können. Außerdem kann sofort eine grundlegende Zugriffskontrolle für Endbenutzer mit unterschiedlichen Rollen generiert werden.

Kapitel 6: “Wie werden Berichte verwaltet?” führt Sie durch die Bereitstellung von Datenansichten mithilfe klassischer und interaktiver Berichtsassistenten. APEX-Berichte enthalten bereits Funktionalitäten für die individuelle Anpassung von Berichten durch den Endbenutzer, ganz ohne

Programmierung.

Kapitel 7: “Wie verwaltet man Formulare?” stellt drei gängige Arten von Webformularen vor, einschließlich Master-Detail-Web-Formularen. Sie werden Formularseiten ohne Programmierkenntnisse anpassen und generieren.

Kapitel 8: “Wie wandelt man Textberichte in Diagramme um?” ebnet den Weg zur Nutzung der APEX-Funktionen, um Daten als Diagramme direkt neben Textberichten darzustellen.

Kapitel 9: “Wie verwaltet man Menüs?” stellt verschiedene Arten von Navigationselementen für Ihre APEX-Anwendung vor.

Kapitel 10: “Wie man im Team zusammenarbeitet?” bietet Einblicke in APEX-Funktionen, die Teams nutzen, da es selten vorkommt, dass an der Anwendungsentwicklung nur ein Entwickler beteiligt ist.

Kapitel 11: “Wie kann man von der Galerie der Anwendungen und Plug-ins profitieren?” lädt Sie ein, leistungsstarke APEX-Funktionen durch die Wiederverwendung häufig vorkommender Muster anzuwenden.

Kapitel 12: “Wie verwaltet man verpackte und mehrsprachige Anwendungen?” zeigt den Weg zur Verteilung Ihrer Anwendung auf andere APEX-Umgebungen für Benutzer auf, die verschiedene Sprachen sprechen.

Teil I behandelt auch Themen, die für die Anwendungssicherheit, Verteilungsstrategien und die Bereitschaft für die Praxis von entscheidender Bedeutung sind.

Teil II dieses Lehrbuchs führt Sie über die Grundlagen hinaus und präsentiert zwölf praxisnahe Geschäftsfälle anhand einer verbalen Beschreibung bis hin zu technischen Details zur Lösung eines Problems. Jeder Fall wird sorgfältig dokumentiert, um ein ganzheitliches Verständnis der Anwendungsentwicklung aus Geschäfts-, Daten- und Benutzeroberflächenperspektive zu ermöglichen. Dieser Teil umfasst folgende Anwendungen für Unternehmen:

- Intranet-Nachrichten für Mitarbeiter,
- Kleines Innovationssystem,
- Management von Geschäftsprozessen,
- GreenDi - Tausch von Pflanzen und Saatgut,
- Verwaltungssystem für Buchrezensionen,
- Stückliste und Kostenberechnung,
- Ernährungs- und Diätmanagement,
- Sprechstundenplanung,
- Fall Autovermietung

for communities:

- GreenDi - Pflanzenkatalog,
- GreenDi - Tausch von Pflanzen und Saatgut

und allgemeingültige Benutzerautorisierung und -verwaltung. In jedem Geschäftsfall werden Sie Folgendes untersuchen:

- Geschäftsansicht des Falles: Ein Überblick über die Geschäftslage.
- Problemdefinition: Eine Suche nach Antworten darauf, wer und warum jemand Probleme hat.
- Anwendungsfälle: Es werden drei Arten der Beschreibung vorgestellt: narrativ, halbstrukturiert und grafisch, um die Dokumentation von UML-Anwendungsfällen vorzubereiten.
- Logisches und relationales Datenmodell: APEX ist voll ausgestattet, um neue Datenstrukturen zu erstellen, vorhandene zu verwenden und zu ändern, mit anderen Datenmodellierungstools zu kombinieren und Forward- oder Reverse-Engineering zu unterstützen. Die Bemühungen der Entwickler, geeignete Datenblöcke und die Beziehungen zwischen ihnen sicherzustellen und die Geschäftsanforderungen zu berücksichtigen, bilden eine Grundlage für die Weiterentwicklung von Benutzeroberflächen.

- Anwendungsschnittstellen: Das Lehrbuch stellt HTML-Seiten, Berichte, Formulare, Felder, Menüs, Schaltflächen und Hyperlinks bereit, die die Geschäftssituation, die Lösung von Geschäftsproblemen, Anwendungsfälle und Daten mit Blick auf den Endbenutzer konkretisieren.
- Ergänzendes Lernmaterial: Zur Erweiterung, Beschleunigung und Unterstützung auf dem Entwicklungspfad finden Sie für jedes Kapitel Links zu exportierten Anwendungen, Skripten, Daten und Video-Tutorials. Diese Ressourcen bieten Ihnen praktische Einblicke, sodass Sie Ihr Wissen vertiefen und direkt auf reale Projekte anwenden können.

Egal, ob Sie ein erfahrener Entwickler sind, der seine Fähigkeiten erweitern möchte, oder ein Anfänger, der die Welt von APEX erkunden möchte, dieses Lehrbuch ist Ihr definitiver Leitfaden. Wir hoffen, dass es auch für Nicht-IT-Lernende eine wertvolle Hilfe auf dem Weg zur Beherrschung von Oracle APEX und zur Entwicklung innovativer Anwendungen bietet.

Das Lehrbuch und die ergänzenden Materialien sind auf einen studentischen Aufwand von ca. 75 Stunden (3 ECTS) ausgelegt. Wir hoffen, dass verschiedene Studienmodi angewendet werden können:

- Lehrer, welche Vorlesungen und Laborübungen leiten,
- Blended Learning und auch
- Selbststudium.

Abhängig vom Hintergrundwissen der Lernenden und der verfügbaren Zeit für die Durchführung des Kurses können Lehrer problemlos eine Reihe von Kapiteln zusammenstellen, die zu den Lernsituationen passen, z. B. außerschulische Kurse, Sommerschulen, zeitlich begrenzte Veranstaltungen für Low-Code-Einführungen für alle Schüler (nicht nur in (IT oder CS) und Praktiker in verschiedenen Branchen.

Für die Entwicklung dieses Lehrbuchs und ergänzenden Lernmaterials wurden die APEX-Versionen 22 und 23 verwendet. Wir glauben, dass die in den Inhalten erläuterten und angewendeten Konzepte und Kerntechnologien auch für die Lernenden der zukünftigen APEX-Versionen von Vorteil sein werden.

Machen Sie sich bereit für ein spannendes Lernabenteuer mit Oracle APEX! Viel Spaß mit Assistenten und Low-Coding!

Professor Robert Leskovar
BeeAPEX Projektleiter, Leiter des IT Departments, UM, Faculty of Organizational Sciences

# Anleitung in APEX


1 **Wie wird Oracle APEX gestartet?** . . . . . 35

Vjeran Strahonja, Dijana Oreški . . . . . .

2 **Wie wird eine Datenbank vorbereitet?** 51

E. Kapsammer, W. Retschitzegger, W. Schwinger . . . . .

3 **Wie navigiert man in APEX?** . . . . . 81

E. Kapsammer, W. Retschitzegger, W. Schwinger . . . . .

4 **Wie man Daten mit APEX austauscht?** 93

E. Kapsammer, W. Retschitzegger, W. Schwinger . . . . .

5 **Wie erstellt man den ersten Entwurf einer Anwendung?** . . . . . 113

A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . .

6 **Wie werden Berichte verwaltet?** . . . . . 135

Zuzana Žillová, Erik Malina, Matej Grochal, Andrej Staník, Andrea Meleková, Michal Kvet, Miroslav Potočár

7 **Wie verwaltet man Formulare?** . . . . . 157

Veronika Šalgová, Michal Kvet, Miroslav Potočár; Übersetzer W. Schwinger . . . . . .

8 **Wie wandelt man Textberichte in Diagramme um?** . . . . . 165

Ivan Pastierik, Michal Kvet, Miroslav Potočár . . . . . .

9 **Wie verwaltet man Menüs?** . . . . . 183

Veronika Šalgová, Michal Kvet, Miroslav Potočár; Übersetzer W. Schwinger . . . . . .

10 **Wie man im Team zusammenarbeitet?** 187

P. Staniszewski, M. Sońta, A. Kierzkowski . . . . . .

11 **Wie kann man von der Galerie der Anwendungen und Plug-ins profitieren?** . . . . 199

Vjeran Strahonja, Dijana Oreški . . . . . .

12 **Wie verwaltet man verpackte und mehrsprachige Anwendungen?** . . . . . 209

R. Leskovar, U. Rajković, A.Baggia . . . . . .



# 1. Wie wird Oracle APEX gestartet?

VJERAN STRAHONJA, DIJANA OREŠKI

## 1.1 Was ist Oracle APEX und wofür wird es eingesetzt?

### 1.1.1 Was ist Oracle APEX?

Oracle APEX (auch bekannt als APEX oder Oracle Application Express) ist eine von der Oracle Corporation entwickelte Low-Code-Entwicklungsplattform für Unternehmen, die es den Benutzern ermöglicht, Anwendungen über eine webbasierte Schnittstelle zu erstellen und einzusetzen. Es handelt sich um eine Reihe von Werkzeugen zur Erstellung verschiedener Arten von Anwendungen, die aus Programmen für die Arbeit mit Daten in der Datenbank bestehen, einschließlich Eingabe, Speicherung, Anzeige, Änderung, Berechnung und logische Operationen mit Daten usw. Die Verwendung dieser Plattform ermöglicht die Entwicklung eigener Anwendungen für verschiedene Zwecke, z.B. die Erstellung einer Datenbank auf der Grundlage einer Reihe strukturierter Daten oder Datenmodelle, die direkte Arbeit mit den Daten in der Datenbank unter Verwendung der SQL-Sprache, die Erstellung von Benutzeroberflächen, die Erstellung verschiedener tabellarischer und grafischer Berichte und die Bereitstellung von Webanwendungen auf Oracle-Datenbanken.

### 1.1.2 Und wie sieht die Anwendung aus?

Eine Anwendung ist im Allgemeinen eine Reihe von Computerprogrammen, die einen bestimmten Zweck verfolgen und mit einer Reihe von Daten in einer Datenbank arbeiten. Wir sind besonders an Geschäftsanwendungen interessiert, d.h. an solchen, deren Programme zur Verarbeitung von Daten über Geschäftsprozesse und Transaktionen verwendet werden. Die Entwicklung einer Anwendung, selbst einer einfachen, ist ein komplexer Denk- und Gestaltungsprozess, der bestimmte Kenntnisse und Fähigkeiten erfordert. Zuallererst ist eine algorithmische Denkweise (algorithmische Kompetenz) erforderlich. Es handelt sich dabei um die Fähigkeit, Konzepte zu entwickeln, das Bewusstsein, dass ein Algorithmus in einer bestimmten Situation existiert, die Fähigkeit, ihn anzuwenden, das Wissen, wie ein Algorithmus entworfen und verwendet wird, und die Fähigkeit, Algorithmen kritisch zu bewerten. Erforderlich sind auch Kenntnisse und Fähigkeiten zur Erstellung hochwertiger Datensätze in einer Programmiersprache (Codierung), zum Testen und Debuggen und zu anderen Dingen, die wir gewöhnlich als Programmierung bezeichnen.

### 1.1.3 Wie kann Oracle APEX bei der Anwendungsentwicklung helfen?

Bei der Entwicklung von Computerprogrammen kommt man um die algorithmische Denkweise nicht herum. Die Verwendung von Plattformen und Tools für die Low-Code-/No-Code-Programmierung, wie Oracle APEX, ermöglicht es Nicht-Programmierern, einfach zu programmieren, ohne die Details einer Programmiersprache und fortgeschrittene Programmierkenntnisse kennen zu müssen. So können sie sich auf das Problem und den Algorithmus und nicht auf die Programmiersprache und die Technologie konzentrieren. Unser besonderes Interesse gilt der schnellen und benutzerfreundlichen Entwicklung von Anwendungen für den Zugriff, die Zusammenführung, die Analyse und die Darstellung von Daten aus offenen Datenquellen und aus Unternehmensdatenbanken. Offene Daten sind frei verfügbare Daten, auf die jeder zugreifen, sie nutzen und weitergeben kann. Sie können ohne Einschränkungen (oder mit einigen) genutzt, mit anderen geteilt oder zur Schaffung neuer Werke verwendet werden. Sie werden von Regierungen, Unternehmen und anderen Organisationen zur Verfügung gestellt, um Transparenz und Zusammenarbeit zu fördern. Heute gibt es viele leicht zugängliche Quellen für meteorologische, verkehrstechnische, geologische, gesundheitliche und ähnliche Daten sowie für Statistiken über Bildung, Gesundheit, Wirtschaft usw. Unternehmensdatenbanken enthalten Daten über Geschäftsvorgänge und andere Daten, die als Ergebnis von Geschäftsprozessen entstehen. Visualisierung, Analyse, Integration und verschiedene Ansichten dieser Daten können genutzt werden, um Geschäftsentscheidungen zu treffen oder andere Vorteile zu erzielen. Oracle APEX wird für die Entwicklung von Prototypen verwendet, die dazu dienen, zu besprechen, was das Programm tun soll, und dem IT-Spezialisten zu zeigen, was der Benutzer wünscht. Der Prototyp spiegelt die wesentlichen Merkmale des Systems wider. Später wird er dem zu verwendenden System hinzugefügt oder dient als Grundlage für die Erstellung eines Systems in einer anderen Programmiersprache, einem anderen Tool oder einer anderen Plattform.

### 1.1.4 Zyklus der Anwendungsentwicklung

Unabhängig davon, welche Art von Anwendung wir entwickeln und welche Art von methodischem Rahmen, Entwicklungswerkzeugen oder Plattform wir verwenden, durchläuft der Anwendungsentwicklungszyklus mehrere Phasen. Diese Phasen unterscheiden sich in den verschiedenen methodischen Ansätzen. Abbildung 1.1 zeigt ein mögliches Muster des Anwendungsentwicklungszyklus (siehe Abbildung 1.1).

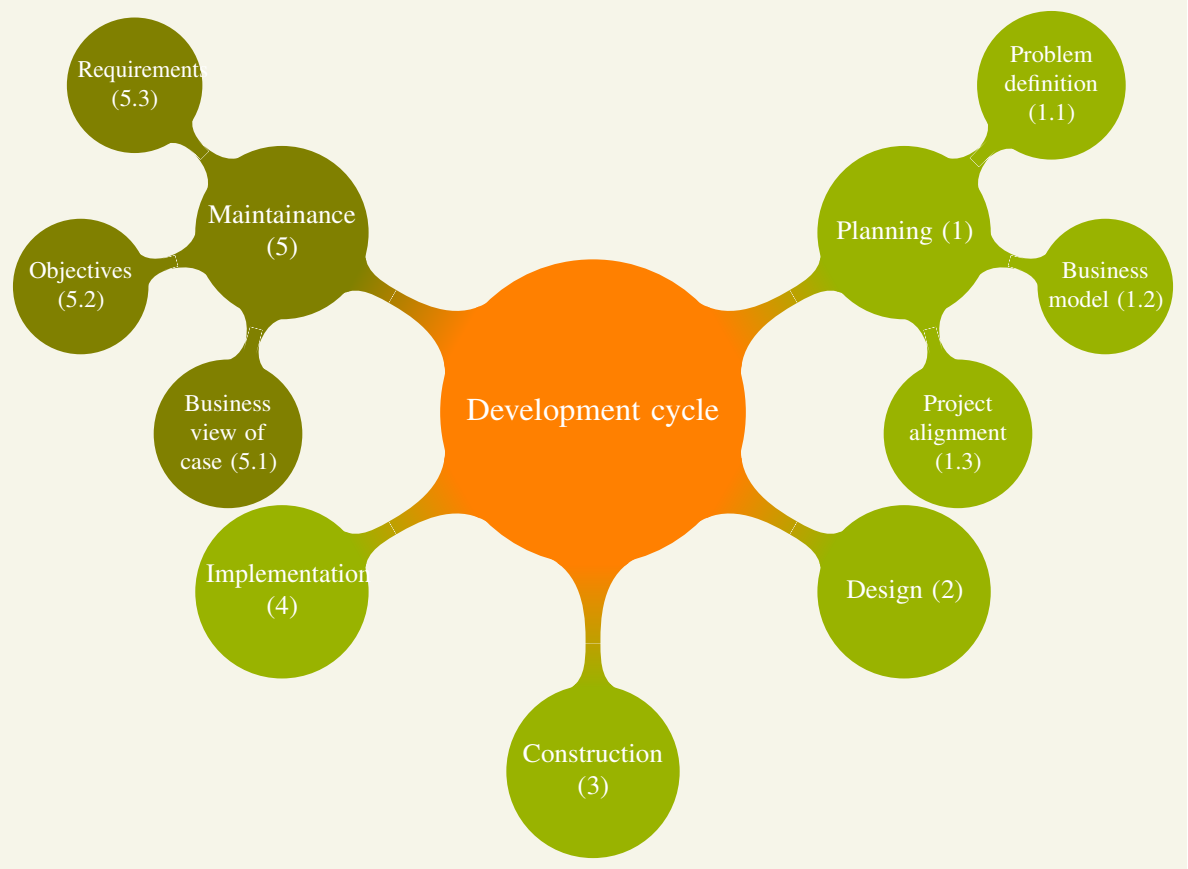



Abbildung 1.1: Zyklus der Anwendungsentwicklung.

Wie in der Abbildung dargestellt, umfasst der Zyklus der Anwendungsentwicklung die folgenden Phasen oder Prozesse:

1. Planung (Definition des Problems und des Projekts)
   - Definition und Analyse des Geschäftsmodells
   - Pflichtenheft
   - Ermittlung der grundlegenden Prozesse und Daten
   - Technologie- und Architekturentscheidungen
   - Entwicklungsplanung
2. Anwendungsdesign
   - Analyse und Spezifikation der Anforderungen
   - Logischer Anwendungsentwurf (Anwendungsfälle und logisches Datenmodell)
3. Aufbau der Anwendung
   - Datenbankdesign
   - Entwicklung (Prototyp) der Anwendung
   - Testen
4. Implementierung der Anwendung

Auf die Planung folgen Entwurf, Konstruktion und Implementierung. Diese Phasen oder Prozesse können sequentiell oder iterativ (evolutionär, inkrementell) durchgeführt werden, so dass wir den gesamten Entwicklungszyklus mehrmals durchlaufen und in jeder Iteration einen Teil der Funktionalität hinzufügen.

**Planung der Anwendung** Der Zyklus der Anwendungsentwicklung beginnt mit der Planung. Bei einfachen Anwendungen kann die Ausgangsinformation für die Planung der Anwendung die Problemdefinition sein, ein Dokument, welches das Problem klar benennt. Wer hat Kopfschmerzen?

Warum ist es ein echtes Kopfweh? Wie äußert sich das Problem? Das können ein paar Sätze oder Seiten Text sein. Im Falle komplexer Geschäftsanwendungen sollten detaillierte Spezifikationen berücksichtigt werden:

- Die betriebswirtschaftliche Sicht des Falles beschreibt, wie die Menschen im Unternehmen Situationen wahrnehmen oder was sie von der Anwendung erwarten.
- Die Ziele der Anwendungsentwicklung beschreiben, was die Anwendung in geschäftlicher, technologischer und sonstiger Hinsicht erreichen will, sowie die Indikatoren für den Erfolg.
- Das Geschäftsmodell beschreibt Werte, Kundensegmente, Kundenbeziehungen und -kanäle, wichtige Partner, Aktivitäten und Ressourcen, Struktur und Quellen von Einnahmen und Kosten usw.
- Die Software-Anforderungsspezifikation definiert die Anforderungen an Funktion, Verhalten, Leistung, Technologie usw.
- Der Projektplan beschreibt den Umfang und die Phasen des künftigen Projekts, die Aktivitäten (Aufgaben), die in den einzelnen Phasen durchgeführt werden, die Inputs und Outputs der einzelnen Aktivitäten, die benötigten Ressourcen, die Verknüpfung der Aktivitäten und ihre Dauer, Kosten, Risiken usw.

Für jede der aufgeführten Spezifikationen gibt es Methoden, die in der Literatur beschrieben sind. So hat Alexander Osterwalder [6] die Business Model Canvas-Methode entwickelt, die zur Entwicklung, Verbesserung und Dokumentation neuer und bestehender Geschäftsmodelle eingesetzt wird. Wie in Abbildung 1.2 dargestellt, handelt es sich dabei um ein visuelles Diagramm mit 9 Bausteinen, die das Wertangebot des Unternehmens, seine Kunden, die Beziehungen und Kanäle zu ihnen, die wichtigsten Partner, Aktivitäten und Ressourcen sowie die Kostenstruktur und die Ertragsströme beschreiben.




Abbildung 1.2: Business Model Canvas [6]

.

Während der Business Model Canvas (BMC) dazu dient, das Verständnis, die Analyse und den Entwurf des Geschäftsmodells zu fördern, und insbesondere, was von der von uns entwickelten Anwendung unterstützt werden muss, ist die Software Requirements Specification (SRS) ein Dokument, das beschreibt, was die Software tun wird und wie sie funktionieren soll. Es beschreibt die Funktionalität, Leistung und andere Merkmale, die die Anwendung haben muss, um das Geschäftsmodell und die Bedürfnisse aller Benutzer zu unterstützen.

**Anwendungsdesign** Von den ersten Computern und Computerprogrammen bis heute wurden hunderte von verschiedenen methodischen Ansätzen, Methoden und Techniken für die Anwendungsentwicklung entwickelt. Die Methodik der Programmentwicklung hat sich mit der technologischen Entwicklung verändert. Die Entwicklung von Low-Code-/No-Code-Programmen ist ein Ansatz, der einfache Methoden bevorzugt, die auch für Benutzer verständlich sind, die keine professionellen Softwareentwickler sind. Die Anwendungsentwicklung erfordert jedoch einen gewissen Entwurf vor der Programmerstellung. Wie bereits erwähnt, besteht eine Computeranwendung aus einer Reihe von Programmen, die mit Daten in einer Datenbank arbeiten. Um eine Anwendung zu entwickeln, müssen wir sowohl die Programme als auch die Datenbank entwerfen. Wenn die Datenbank bereits existiert, müssen wir wissen, wie man sie benutzt, und vor allem das Datenmodell verstehen. Für die Bedürfnisse der Low-Code/No-Code Programmentwicklung werden wir die notwendigsten

Entwurfsmethoden vorstellen. Dabei ist die Grundlage des Entwurfs die Erstellung von Modellen, die auch eine grafische Darstellung haben. Wir werden die folgenden Methoden anwenden:

- Use Case (UC) Modell, das die Beziehung zwischen der Anwendung und der Umgebung definiert.
- Ein Entity-Relationship (ER)-Modell, das die Struktur von Daten auf einer logischen Ebene beschreibt.

Use Case wird in der Fortsetzung dieses Kapitels erklärt, während Entity-Relationship (ER) in Kapitel 2 erläutert wird.

**Use Case Modell (UC)** Use Cases sind eine der Methoden der Softwareentwicklung. Sie beschreiben die Beziehung zwischen dem System (Anwendung) und der Umgebung (Benutzer und andere Systeme). Die Methode ist einfach, so dass auch nicht-professionelle Benutzer sie verstehen. Use Cases sind ein integraler Bestandteil der Unified Modeling Language (UML), der heute am häufigsten verwendeten Modellierungssprache im Bereich des Software-Engineerings, die eine Standardmethode zur Visualisierung des Entwurfs eines Systems bietet. Es gibt eine umfangreiche UML-Literatur und die Object Management Group (OMG) kümmert sich um die Entwicklung und Standardisierung der Sprache. UML enthält 13 Methoden und Diagrammtechniken zur Modellierung der Struktur, des Verhaltens und der Interaktion von Software, und wir werden nur Use Cases verwenden.

Das Anwendungsfallmodell besteht aus zwei Teilen:

- Spezifikationen (narrative Beschreibung) von Anwendungsfällen
- Anwendungsfalldiagramm

Zunächst werden wir das Anwendungsfalldiagramm untersuchen. Dieses Diagramm beschreibt, was das System aus der Sicht eines Beobachters außerhalb des Systems tut. Es spielt keine Rolle, wie das System intern funktioniert.

Die grundlegenden Konzepte des Anwendungsfalldiagramms sind in Abbildung 1.3 dargestellt.

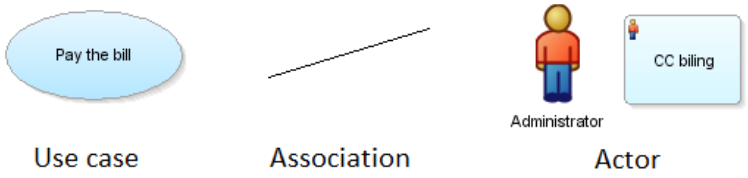



Abbildung 1.3: Die grundlegenden Konzepte des Anwendungsfalldiagramms.

Ein Anwendungsfall stellt eine Reihe von Aktionen dar, die von einem System (z.B. einer Softwarekomponente) in Interaktion mit externen Akteuren durchgeführt werden können. Es ist eine Geschichte, die beschreibt, wie Akteure das System nutzen, um bestimmte Ziele zu erreichen oder Aufgaben zu erfüllen. Er stellt eine abstrakte Aufgabe dar, die einen Zweck hat. Die Verknüpfung (Assoziation) verbindet die Teilnehmer an der Kommunikation, z.B. den Akteur und den Anwendungsfall, und stellt deren Interaktion und die Beziehung zwischen dem System und der Umgebung (Verhalten) dar. Ein Akteur stellt eine Reihe von Rollen dar, die auf die gleiche Weise mit dem System interagieren, z.B. eine Benutzerklasse, eine Art externes System usw. Ein Akteur ist jemand, der außerhalb des betrachteten Systems steht und mit diesem interagiert, aber nicht Teil davon ist. Dabei kann es sich um ein Lebewesen (Benutzer, Patient, Pilot) oder um ein anderes System (Abrechnungssystem, Bank, Spediteur) handeln.

Das Diagramm kann auch die Klassifizierungsstruktur zeigen (siehe Abbildung 1.4), d.h. die Generalisierung/Spezialisierung von Konzepten, z.B. Akteuren. Alle Unterklassen (children) erben Eigenschaften und Verhalten von der Oberklasse (parent), was bedeutet, dass sie auch die Verknüpfungen zu Anwendungsfällen von der Elternklasse erben. Gleichzeitig kann eine einzelne Unterklasse in eine Kommunikation eintreten, in die weder die Oberklasse noch andere Unterklassen

eintreten.

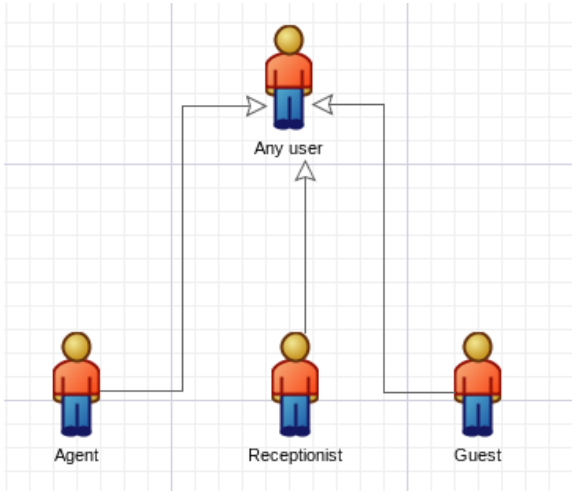



Abbildung 1.4: Generalisierung/Spezialisierung von Akteuren.

Abbildung 1.5 zeigt die Grenzen des Systems als ein Rechteck mit dem Namen des Systems, innerhalb dessen sich die Anwendungsfälle befinden.

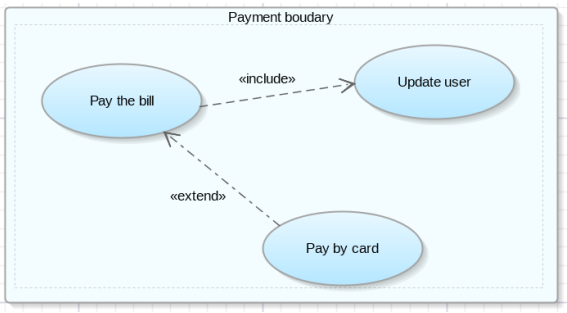



Abbildung 1.5: Die Grenzen des Systems und Assoziationen „include“ und „extend“.

UC-Diagramme zeigen nicht den Arbeitsablauf oder die Reihenfolge der Anwendungsfälle. Es gibt jedoch zwei zulässige Arten von Verbindungen zwischen Anwendungsfällen, die auch in Abbildung 1.5 dargestellt sind:

- „include“ ist eine Assoziation zwischen dem Basisanwendungsfall und dem eingeschlossenen Anwendungsfall, die anzeigt, dass der Basisanwendungsfall das Verhalten des eingeschlossenen Anwendungsfalls enthält. Auf diese Weise wird die Funktionalität und das Verhalten, das häufig verwendet wird, in einen Anwendungsfall getrennt, den wir nach Bedarf in andere Anwendungsfälle einbeziehen. Beachten Sie, dass der Basisanwendungsfall ohne die Einbeziehung nicht vollständig ist.
- „extend“ ist eine Assoziation von einem Anwendungsfall, der eine Erweiterung zu einem Basis-Anwendungsfall ist, und zeigt an, dass das Verhalten des Basis-Anwendungsfalls durch das Verhalten der Erweiterung erweitert wird. Gleichzeitig ist die Erweiterung nicht Teil des Basisanwendungsfalls, noch sollte diese Kommunikation immer existieren und einwandfrei funktionieren.

Die Verwendung der oben genannten Konzepte des Use-Cases-Diagramms lässt sich am besten anhand eines Beispiels erläutern, z.B. des Hotelreservierungssystems, das in Abbildung 1.6 dargestellt ist.

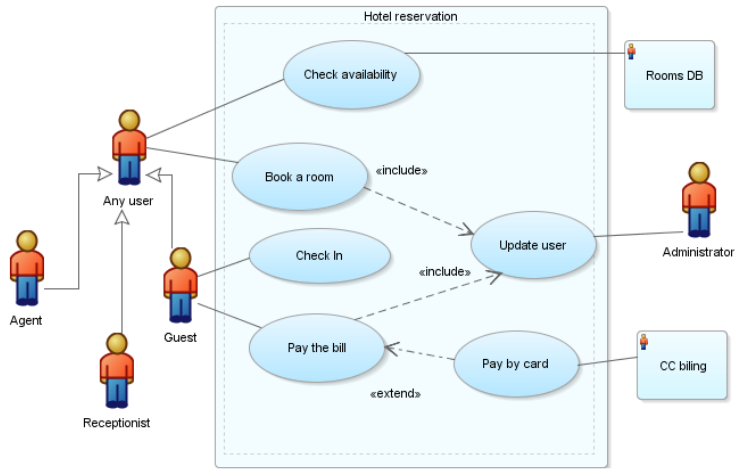



Abbildung 1.6: Diagramm der Anwendungsfälle des Hotelreservierungssystems.

Das System, dessen Verhalten wir beobachten, ist Hotel Reservation. Das gesamte System befindet sich innerhalb der durch das Rechteck dargestellten Grenzen. Die Umgebung des Systems liegt außerhalb des Rechtecks. Innerhalb der Systemgrenzen befinden sich sechs Anwendungsfälle, die durch Ellipsen mit Namen dargestellt sind. Außerhalb des Systems befinden sich die Akteure, die mit dem System, d.h. seinen Anwendungsfällen, kommunizieren, und diese Kommunikation wird durch Verbindungslinien, Assoziationen, dargestellt. Any User ist ein Akteur, der mit Check Availability kommuniziert. Dieser Anwendungsfall kommuniziert mit einem externen Akteur, der Rooms DB. Dabei handelt es sich um eine Art externes Reservierungssystem, das Informationen über Räume und Reservierungen in seiner Datenbank hat. Any User kann den Raum buchen. Dieser UC beinhaltet einen weiteren UC, Update User. Die Klassifizierungsstruktur des Akteurs Benutzer ist auf der linken Seite dargestellt. Agent, Receptionist und Guest sind „eine Art von Akteur Any User“. In dem gezeigten Fall kann nur Guest mit den UCs Check In und Pay the bill kommunizieren. UC Update User ist in Pay the bill enthalten, was bedeutet, dass Benutzerdaten auch innerhalb der Rechnungszahlungsfunktionalität aktualisiert werden können. UC Pay by card erweitert die Pay the bill-Funktionalität, d.h. die Kartenzahlung kann während der Rechnungszahlung aufgerufen werden. Pay by card kommuniziert mit einem externen Kreditkartenzahlungsdienst CC billing. Der Administrator kann mit UC Update user kommunizieren und Benutzerdaten aktualisieren. Wie wir bereits erwähnt haben, besteht das Anwendungsfallmodell aus zwei Teilen:

- Spezifikationen (narrative Beschreibung) von Anwendungsfällen
- Anwendungsfalldiagramm.

Die Use-Case-Spezifikation beschreibt Szenarien und interne Logik von Anwendungsfällen, Anfangszustände (Vorbedingungen) und Endzustände, Schnittstellen, Systemmeldungen, Spezifikation der Fehler- und Ausnahmeverarbeitung und ähnliches. In der Regel wird für die Use-Case-Spezifikation eine Art Vorlage verwendet, wie wir sie auch in diesem Buch verwenden werden.

**Aufbau, Implementierung und Pflege der Anwendung** Der Aufbau der Anwendung beinhaltet die Erstellung eines Programms und die Realisierung einer physischen Datenbank. Wir werden dafür die Oracle APEX-Plattform verwenden, wie später in diesem Buch erläutert wird. Die erste Version der Anwendung, die funktioniert und getestet werden kann, ist eigentlich ein Prototyp der Anwendung. Dieser Prototyp kann aktualisiert und erweitert werden. In Anbetracht dessen, was später mit dem Prototyp geschieht, unterscheiden wir zwischen zwei Arten von Prototypen: Ein ausbaufähiger Prototyp wird verbessert und in einen gebrauchstauglichen Zustand gebracht. Der „Wegwerf“-Prototyp dient zur Veranschaulichung der Funktionalität und als Modell, auf dem

die Anwendung aufgebaut werden soll, in der Regel auf einer anderen Plattform. Wenn wir beabsichtigen, den Anwendungsprototyp zu verwenden, müssen wir ihn implementieren, d.h. in der Produktionsumgebung installieren, die Benutzer schulen, die Datenbank vorbereiten und ähnliches.

Nach der Implementierungsphase folgt die Nutzungs- und Wartungsphase, die die Behebung und Beseitigung von Fehlern, die Verbesserung der Leistung und kleinere Funktionserweiterungen umfasst. Wenn größere Änderungen erforderlich sind, wird ein neuer Entwicklungszyklus der Anwendung gestartet.

## 1.2 Wie startet man Oracle APEX?

Software-Entwicklung gilt als komplex, da viel programmiert werden muss. Um die Komplexität zu überwinden, werden Low-Code-Lösungen vorgeschlagen. Der Low-Code-Ansatz ermöglicht die Entwicklung anspruchsvoller Anwendungen, die sowohl funktional als auch komplex sind, ohne eine Zeile Code zu schreiben. Oracle Application Express ist eine der besten Low-Code-Optionen für diesen Zweck. Es gibt verschiedene Möglichkeiten, Oracle APEX zu starten. Oracle APEX kann überall dort eingesetzt werden, wo eine Datenbank läuft, sowohl vor Ort als auch in einer privaten Cloud. Es kann ein physischer, dedizierter Server, eine virtualisierte Maschine oder ein Docker-Image sein (das besonders bei Oracle Application Express-Entwicklern beliebt ist und auf einem Laptop, in einem Bus oder in einem Flugzeug gestartet werden kann). Darüber hinaus kann Exadata genutzt werden, ein besonders leistungsfähiger physischer APEX-Server auf Cloud-Diensten. Alle diese Optionen werden hier besprochen. Wie man auf die Entwicklungsumgebung zugreift, wird in diesem Kapitel beschrieben.

### 1.2.1 Wie hoch ist Ihr Kenntnisstand?

Es gibt mehrere Möglichkeiten, mit Oracle Application Express - APEX zu beginnen. Ihr aktueller Kenntnisstand bestimmt den empfohlenen Weg zum Einstieg. Nehmen wir an, Sie haben spezifische Domänenkompetenzen in einem beliebigen organisatorischen Prozess (z.B. Verkauf, Einstellung, Herstellung, Kreditvergabe, Marketing, Rechtsangelegenheiten, Gesundheitswesen, Bauwesen, Bildung) und die folgenden digitalen Kompetenzstufen:

- Absoluter Anfänger: vertraut mit Webbrowser, keinerlei Programmiererfahrung; Empfehlung: On-Premise-Instanz, apex.oracle.com oder Oracle Academy verwenden
- Anfänger: vertraut mit Webbrowser, wenig Programmiererfahrung z.B. mit Tabellenkalkulationen und Formeln; jede andere Programmiersprache; Empfehlung: On-Premise-Instanz, apex.oracle.com oder Oracle Academy verwenden
- frischgebackener Entwickler: vertraut mit Webbrowser; wenig Programmiererfahrung; Verständnis für Datenbank, Tabelle, Tabellenspalte, Primärschlüssel, Fremdschlüssel; grundlegende Abfragen, Einfügen, Aktualisieren und Löschen; Empfehlung: On-Premise-Instanz, apex.oracle.com, Oracle Academy oder VBox-Appliance verwenden
- Erfahrener Entwickler: vertraut mit Webbrowser; mäßige Programmiererfahrung mit Prozeduren und Funktionen in einem beliebigen Datenbankmanagementsystem; mittlere Abfrage-, Einfüge-, Aktualisierungs- und Löschfunktionen; Empfehlung: on-Premise-Instanz, apex.oracle.com, Oracle Academy, VBox-Appliance oder Apex-Docker verwenden
- auf dem Weg zum professionellen Entwickler: Entwicklung von Fähigkeiten in PL/SQL und JavaScript; Empfehlung: On-Premise-Instanz, apex.oracle.com, APEX-Docker oder OCI APEX-Instanz verwenden

Ihre Benutzerrolle und der Ort, an dem Oracle APEX installiert ist, bestimmen, wie Sie sich anmelden und die Anwendung nutzen. Oracle APEX kann lokal auf Ihrem Computer oder in einer gehosteten Umgebung wie einem Oracle Cloud Service installiert werden. Je nach Installationsart variiert das Anmeldeverfahren. Die Benutzer müssen einen Arbeitsbereich erstellen, Oracle APEX-Benutzer hinzufügen und sich dann beim Arbeitsbereich anmelden, bevor sie Anwendungen

entwickeln oder installieren. Mehrere Benutzer können über einen Workspace an derselben Oracle APEX-Installation zusammenarbeiten, wobei die Privatsphäre ihrer Objekte, Daten und Anwendungen gewahrt bleibt. Wenn der Webbrowser JavaScript unterstützt, können sich die Benutzer bei einem Workspace anmelden, um auf die Oracle APEX-Homepage zuzugreifen. Jeder Workspace hat einen eigenen Namen und eine eigene ID. Innerhalb der Oracle APEX-Administrationsdienste kann ein Arbeitsbereich manuell von einem Instanzadministrator erstellt werden, oder Benutzer können Anfragen stellen. Ein unabhängiges Programm namens Oracle APEX Administration Services wird zur Überwachung einer kompletten Oracle APEX-Instanz verwendet.

### 1.2.2 APEX-Instanz vor Ort verwenden

Dieser Abschnitt beschreibt, wie Oracle APEX in einer lokalen Installation installiert wird. Die Installation von Oracle APEX umfasst mehrere Schritte. Diese Schritte sind:

- **Wählen Sie zwischen Voll- oder Laufzeitumgebung**. Die vollständige Umgebung ermöglicht den vollständigen Zugriff auf die Entwicklungsumgebung des App Builders. Für Produktionsimplementierungen, bei denen Sie unveränderliche Anwendungen ausführen möchten, ist eine Laufzeitumgebung eine gute Option. Oracle APEX bietet die Möglichkeit, nur eine Runtime-Version von Oracle APEX für Test- und Produktionsinstanzen zu installieren. Da Entwickler nicht versehentlich ein Produktionsprogramm in einer Laufzeitinstanz aktualisieren können, reduziert diese Laufzeitumgebung den installierten Fußabdruck und Rechte und erhöht die Anwendungssicherheit. Benutzer können Produktionsanwendungen in einer Oracle APEX-Laufzeitumgebung ausführen, aber sie verfügt nicht über eine Webschnittstelle für die Verwaltung. Eine Laufzeitumgebung ist sicherer, da sie nur die für die Ausführung des Programms erforderlichen Komponenten enthält.
- **Überprüfung der Installationsanforderungen** Überprüfen Sie, ob Ihr System die Mindestanforderungen für die Installation erfüllt. Es gibt fünf Gruppen von Anforderungen: a) Oracle Database Requirements (Oracle APEX Release 22 benötigt eine Oracle Database Release 12.1.0.2 oder höher. Oracle APEX läuft auf allen Datenbank-Editionen, einschließlich Enterprise Edition (EE), Standard Edition (SE) und Express Edition (XE). Oracle APEX kann in einer Single-Instance-Datenbank und in einer Oracle Real Application Clusters (Oracle RAC)-Datenbank installiert werden.), b) Browser-Anforderungen (Oracle APEX benötigt einen JavaScript-fähigen Browser und unterstützt die aktuellen und früheren Hauptversionen von Google Chrome, Mozilla Firefox, Apple Safari und Microsoft Edge.), c) Web Listener-Anforderungen (Oracle APEX benötigt Oracle REST Data Services (ORDS) 19.x oder höher), d) Anforderungen an den Festplattenspeicher (Oracle APEX benötigt 310 MB freien Speicherplatz für APEX-Softwaredateien im Dateisystem, wenn Sie nur den englischen Download verwenden, und 705 MB, wenn Sie den vollständigen Download verwenden, 220 MB freien Speicherplatz im APEX-Tablespace, 100 MB freien Speicherplatz im SYSTEM-Tablespace, 60 MB freien Speicherplatz im APEX-Tablespace für jede zusätzlich installierte Sprache (außer Englisch), e) Anforderungen an Oracle XML DB (Oracle XML DB muss in der Oracle-Datenbank installiert sein, die Sie verwenden möchten, wenn Sie eine vollständige Entwicklungsumgebung installieren. Wenn Sie eine vorkonfigurierte Datenbank verwenden, die entweder während einer Installation oder mit dem Database Configuration Assistant oder DBCA erstellt wurde, ist Oracle XML DB bereits installiert und konfiguriert).
- **Installieren** Sie die Software - Installieren Sie Oracle APEX, indem Sie eine ZIP-Datei von der Oracle APEX-Seite herunterladen und dann Oracle REST Data Services (ORDS) herunterladen und installieren.

### 1.2.3 https://apex.oracle.com

Navigieren Sie in Ihrem Browser zu diese URL. Die Beantragung eines kostenlosen Arbeitsbereichs ist der schnellste Weg, um mit Oracle APEX zu beginnen. Es dauert nur ein paar Sekunden, um

sich anzumelden, und der Arbeitsbereich ist für die Benutzer vorbereitet, um mit der Entwicklung von Anwendungen zu beginnen. Für diesen Ansatz ist kein Cloud-Konto erforderlich und er ist kostenlos. Um mit Oracle APEX zu beginnen:

- Geben Sie apex.oracle.com in Ihren Webbrowser ein.
- Klicken Sie auf die Schaltfläche Start for Free Today
- Wählen Sie auf der daraufhin angezeigten Webseite die Option Request a free Workspace
- Geben Sie Ihren Namen und Ihre E-Mail-Adresse ein, geben Sie Ihrem Arbeitsbereich einen Namen und klicken Sie auf Next
- Füllen Sie den Fragebogen aus und geben Sie an, warum Sie einen Arbeitsbereich beantragen. Lesen Sie die Bedingungen durch und akzeptieren Sie die Bedingungen, klicken Sie auf Next
- An dieser Stelle erhalten Sie eine E-Mail von Oracle APEX. Klicken Sie auf den Link, um sich zu registrieren und ein Passwort einzurichten.

### 1.2.4 Virtual Box Appliance / Virtual Machine

Navigieren Sie in Ihrem Browser zu dieser URL. Der Hauptzweck der Appliance ist die Entwicklung und das Testen. Die APEX-Version ist normalerweise nicht auf dem neuesten Stand. Laden Sie Oracle Virtual Box und Extension Manager herunter. Richten Sie Oracle Virtual Box und Extension Manager ein. Importieren Sie die .ova-Datei. Starten Sie die Appliance.

Die Oracle APEX Development VM ist eine gebrauchsfertige virtuelle Maschine, die Sie durch einfaches Importieren in VirtualBox nutzen können. Vorgefertigte virtuelle Maschinen von Oracle sind mit Oracle-Konten verfügbar. Alles, was Sie tun müssen, ist Oracle Virtual Box (Free Virtual Machine Client) zu installieren und dann eine beliebige Anwendung (Pre-Built VM) zu importieren. Oracle bietet mehrere vorgefertigte VMs unter dieser URL an.

Um die VM einzurichten, gehen Sie wie folgt vor:

- Laden Sie Oracle VM VirtualBox herunter und installieren Sie es auf Ihrem Host-System.
- Laden Sie die Dateien herunter (die Verwendung eines Download-Managers wird dringend empfohlen)
- Importieren Sie Ihre VM: File > Import Appliance, um den Assistenten für den Appliance-Import zu starten. Klicken Sie auf Choose..., um zu dem Verzeichnis zu wechseln, in dem Sie alle Dateien neu zusammengestellt haben, und wählen Sie die Datei OTN_Developer_Day_VM.ova. Klicken Sie dann auf Next, um mit dem Importieren der virtuellen Maschine zu beginnen. Sie werden während des Imports aufgefordert, den entsprechenden Entwicklerlizenzen zuzustimmen. Wenn der Import abgeschlossen ist, wird „Oracle Developer Days (Powered Off)“ angezeigt.
- Testen Sie Ihre VM: Sobald der Import abgeschlossen ist, doppelklicken Sie auf die OTN Developer Days VM. Klicken Sie auf OK, um die Virtualbox-Informationsdialoge zu schließen. Wenn Sie zum Bildschirm Enterprise Linux 6 gelangen, können Sie sich jetzt anmelden. (Benutzername und Passwort sind oracle.) Lassen Sie den Prozess ablaufen; er ist abgeschlossen, wenn Sie ein Terminalfenster sehen, das Sie schließen können. Wenn Sie die Arbeit in der Gast-VM beendet haben, können Sie sie über System > Shut Down herunterfahren; dadurch wird die Gast-VM in den ausgeschalteten Zustand versetzt.

Zu den wichtigsten Vorteilen des Einsatzes virtueller Maschinen gehören:

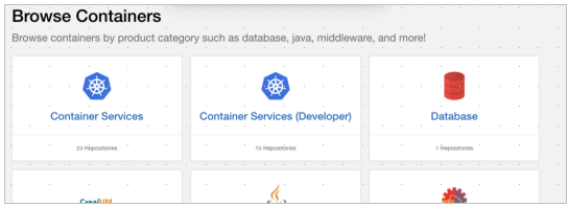
- Nutzung neuer Oracle-Software, ohne dass komplexe Installationen durchgeführt werden müssen oder die Beteiligung von Anbietern erforderlich ist.
- Experimentieren ohne Ihren Arbeitsplatz zu gefährden.
- Kostenlose geführte Schulungen, die von Oracle in Form von Hands On Labs angeboten werden.

### 1.2.5 APEX docker

Holen Sie sich das offizielle APEX-Docker auf der Oracle-Website. Navigieren Sie in Ihrem Browser zu diese URL.

Oracle APEX kann als Docker-Container installiert werden. Docker sind für Entwickler sehr bequem, da sie auf PCs und Laptops installiert werden können. Wir empfehlen dringend, APEX Docker nur von vertrauenswürdigen Quellen zu beziehen. Die Einrichtung ist in der Regel schnell und einfach. Für das Erlernen von Oracle Database und APEX ist die Version vielleicht nicht wichtig, aber für den produktiven Einsatz sollte man auf die Versionen achten. Der Rest dieses Abschnitts beschreibt einen möglichen Weg, APEX docker lokal auf Ihrem Computer zu installieren. In diesem Prozess klonen wir das offizielle Oracle Docker-Image, das Entwicklungswerkzeuge enthält.

1. Der erste Schritt ist die Installation von Docker auf dem Rechner. Docker lässt sich einfach von der Docker-Homepage herunterladen und installieren: `https://docs.docker.com/desktop/install/windows-install/`
2. Der zweite Schritt ist die Anmeldung für ein kostenloses Konto bei der Oracle Container Registry: `https://container-registry.oracle.com/ords/f?p=113:10` Docker-Images werden aus dieser Registry gezogen (siehe Abbildung 2).




3. Der dritte Schritt besteht darin, ein Terminalfenster zu öffnen und mit dem zuvor erstellten Benutzer auf die Oracle-Registrierung zuzugreifen.

   ```
   docker login container-registry.oracle.com
   ```

   Es ist besser, ein Netzwerk in einer Docker-Umgebung zu erstellen, damit Docker mit anderen Dockern über einen Hostnamen kommunizieren können

   ```
   docker network create ords-database-network
   ```

   Jetzt haben wir alles Nötige getan und können den Befehl ausführen, um unsere Oracle Database XE in einem Docker auf unserem Laptop auszuführen.

   ```
   docker run -d --name testapex --hostname database --network=ords-database-network -p 1
   ```

   Beachten Sie, dass hier die Parameter Name, Hostname und Network verwendet wurden. Erläuterung:
   - Das erste Flag -d bewirkt, dass der Container in einem losgelösten Modus läuft.
   - Der Parameter –name gibt den Containernamen an.
   - -p ordnet den Port 1521 auf dem Host-Rechner dem Port 1521 im Container zu, so dass wir eine Verbindung zur Datenbank herstellen können.
   - –network verbindet den Container mit dem von uns erstellten Netzwerk.
   - –hostname gibt dem DB-Server einen Namen.
   - Der letzte Parameter ist das Image, mit dem der Container gestartet werden soll.

   Wenn Sie eine Verbindung über sqlplus herstellen möchten, können Sie diesen Befehl ausführen:

   ```
   docker exec -it -u oracle testapex sqlplus / as sysdba
   ```

Mit dem Befehl *show pdbs* können Sie den Status von pdbs überprüfen. Ändern Sie das Standardpasswort mit dem unten stehenden Befehl (in diesem Fall lautet das neue Passwort Welcome1!!). Bevor Sie diesen Befehl ausführen, vergewissern Sie sich bitte, dass die pdb XEPDB1 zum Lesen und Schreiben geöffnet ist.

```
docker exec testapex ./setPassword.sh Welcome1!!
```

Um Apex für Oracle XE in Docker zu installieren, führen Sie weitere Schritte durch. Ziehen Sie in der Terminal-Session das Image von ords.

```
docker login container-registry.oracle.com
```

Führen Sie dann den Befehl pull aus, um das ords-Image herunterzuladen.

```
docker pull container-registry.oracle.com/database/ords:latest
```

Um die Apex-Installation zu konfigurieren, erstellen Sie ein Verzeichnis.

```
mkdir ~/APEX
```

Legen Sie die String-Informationen in einer Datei in diesem Verzeichnis ab.

```
echo 'CONN_STRING=sys/Welcome1##@database:1521/XEPDB1' >
~/APEX/conn_string.txt
```

Beachten Sie, dass der Parameter hostname von Oracle Database XE docker und der Dienst der pdb für die Stringverbindung verwendet wurden. Definieren Sie die STRING-Variable CONN_STRING wie folgt (muss in der Form unten, ohne einfache Anführungszeichen sein):

```
CONN_STRING=user/password@host:port/service_name
```

Jetzt können wir den Docker starten.

```
docker run --rm --name apex
-v /Users/lbindi_it/APEX:/opt/oracle/variables
--network=ords-database-network -p 8181:8181
container-registry.oracle.com/database/ords:latest
```

Hier wurde der gleiche Parameter network verwendet. Um die Installation zu überwachen, können Sie eine weitere Terminal-Session öffnen und diesen Befehl ausführen:

```
docker run - rm - name apex
-v /Users/lbindi_it/APEX:/opt/oracle/variables
--network=ords-database-network -p 8181:8181
container-registry.oracle.com/database/ords:latest
```

Ändern Sie das Passwort für den Benutzer APEX_PUBLIC_USER innerhalb der Datenbank. Melden Sie sich dazu in einem Oracle XE-Docker mit sqlplus an.

```
sqlplus sys/Welcome1##@//localhost:1521/XEPDB1 as sysdba
```

Legen Sie das Passwort fest.

```
alter user APEX_PUBLIC_USER identified by Welcome1##;
```

Jetzt können Sie sich mit Ihrem Browser mit der APEX-Umgebung auf Ihrem lokalen Computer verbinden.

```
http://localhost:8181/ords
```

### 1.2.6 APEX-Instanz in der Oracle Cloud-Infrastruktur

Oracle Cloud Infrastructure (OCI) bietet APEX Low-Code-Anwendungsentwicklung auf der Autonomous Infrastructure als vollständig verwalteten Service, der vorkonfiguriert und einsatzbereit ist. OCI bietet elastische Skalierbarkeit, Sicherheit, hohe Verfügbarkeit und globalen Zugang über regionale Cloud-Rechenzentren. Um eine APEX-Service-Instanz zu erstellen, werden in diesem Abschnitt die Verfahren für den Zugriff auf die Oracle Cloud Infrastructure (OCI) beschrieben. Die Schritte für den Zugriff auf die OCI sind wie folgt:

1. **Holen Sie sich ein OCI-Konto**. Der Benutzer muss über ein OCI-Konto mit Zugriff auf ein OCI-Konto verfügen, um den APEX-Service nutzen zu können. Der Benutzer kann ein bestehendes OCI-Konto verwenden, wenn das Unternehmen eine Vertriebsvereinbarung mit Oracle hat. Um den Zugang zu beantragen, setzen Sie sich zunächst mit dem OCI-Administrator Ihres Unternehmens in Verbindung. Wenn Sie als Einzelanwender bei Null anfangen oder nicht wissen, wo Sie anfangen sollen, sollten Sie sich für das Oracle Cloud Free Tier entscheiden. Neben einer kostenlosen anfänglichen Zuteilung von Cloud Credits bietet dies eine kostenlose, nicht auslaufende OCI-Miete und ein Konto. Während der Testphase können diese Credits für den Kauf von kostenpflichtigen OCI-Diensten wie dem APEX-Service verwendet werden. Wenn Sie den APEX-Service nach Ablauf der Probezeit oder nach Ablauf der kostenlosen Credits (je nachdem, was zuerst eintritt) weiter nutzen möchten, müssen Sie das Konto auf einen kostenpflichtigen Status hochstufen und weitere Credits kaufen. Ihr OCI-Konto wird in einen Zustand versetzt, in dem es nur noch OCI-Dienste nutzen kann, die eine winzige Always Free-Form haben, wenn Sie nicht vor Ablauf der Probezeit ein Upgrade durchführen. Eine geringfügige Always Free-Form ist im APEX-Service vorhanden. Oracle rät Ihnen, Ihr Konto in ein kostenpflichtiges umzuwandeln und entweder während oder nach der Testphase weitere Credits zu kaufen. Beginnen Sie den Anmeldeprozess, indem Sie das Oracle Cloud Infrastructure Free Tier prüfen. Um die Anmeldung zu starten, gehen Sie zu https://signup.oraclecloud.com/.
2. **Anmeldung bei der OCI-Konsole.** Verwenden Sie einen kompatiblen Webbrowser, um die Anmeldeseite der OCI-Konsole aufzurufen. Geben Sie Ihren Benutzernamen und Ihr Kennwort ein, gefolgt von dem Namen Ihres Cloud-Kontos (auch bekannt als Ihr Tenancy-Name). Ihre Willkommens-E-Mail enthält sowohl Ihren Benutzernamen als auch den Namen Ihres Cloud-Kontos.

### 1.2.7 APEX-Instanz in der Oracle Academy

Oracle Academy bietet institutionellen Mitgliedern und ihren Studenten Zugang zu Oracle Application Express (APEX) für praktische Übungen in der Cloud. Oracle Application Express (APEX) wird von der Oracle Academy für institutionelle Mitglieder und deren Studenten zur Verfügung gestellt, um mit der Cloud zu experimentieren. Zur Unterstützung von Übungen und angewandter Praxis für den Lehrplan, der Datenbankgrundlagen, Datenbankdesign und -programmierung mit SQL, Programmierung mit PL/SQL und Grundlagen der Oracle Application Express-Anwendungsentwicklung umfasst, bietet Oracle Academy Lehrkräften eine dedizierte Instanz an, die speziell für den Einsatz im Klassenzimmer mit bis zu 99 Studenten konzipiert ist. Um über Oracle Academy auf Oracle APEX zugreifen zu können, müssen sich die Benutzer beim Member Hub anmelden. Wenn sie keine Mitglieder sind, sollten sie sich kostenlos als institutionelle Mitglieder anmelden, um Zugang zu allen Materialien und anderen Vorteilen zu erhalten. Bitte navigieren Sie Ihren Browser zu https://academy.oracle.com/en/oa-web-overview.html.

## 1.3 Fragen

1. Welche sind die Phasen des Anwendungs-Entwicklungszyklus und welche Aktivitäten gehören zur dritten Phase?

2. Welche der Möglichkeiten, Oracle Application Express zu beginnen, ist die beste Lösung für absolute Anfänger?
3. Was sind die Vorteile der Verwendung einer APEX-Instanz in der Oracle Cloud Infrastructure?

## 1.4 Antworten

1. Die Phasen des Anwendungs-Entwicklungszyklus sind: (i) Planung (Definition des Problems und des Projekts), (ii) Anwendungsdesign, (iii) Konstruktion der Anwendung und (iv) Implementierung der Anwendung. Die Aktivitäten in der dritten Phase (Konstruktion der Anwendung) umfassen: Datenbankdesign, Entwicklung (Prototyp) der Anwendung und Tests.
2. Die Empfehlung für absolute Anfänger lautet, eine On-Premise-Instanz, apex.oracle.com oder die Oracle Academy zu verwenden.
3. Oracle Cloud Infrastructure (OCI) bietet APEX Low-Code-Anwendungsentwicklung auf der autonomen Infrastruktur als einen vollständig verwalteten Dienst, der vorab konfiguriert und einsatzbereit ist. OCI bietet elastische Skalierbarkeit, Sicherheit, hohe Verfügbarkeit und globalen Zugriff über regionale Cloud-Rechenzentren.

# 2. Wie wird eine Datenbank vorbereitet?

E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER

Nehmen Sie an, Sie arbeiten für *Walmart* und *Datenbanksysteme* (DBS) sind noch nicht erfunden worden. Nehmen wir weiter an, Sie sollen eine *Webanwendung* implementieren, die jeden einzelnen Verkauf in jeder der 10.500 Walmart-Filialen speichern, abrufen, visualisieren und weiterverarbeiten kann, zusammen mit Informationen über die verantwortlichen Abteilungen, Mitarbeiter und deren Jobs, was Petabytes an Daten sein könnten. Eine Aufgabe, die offensichtlich nicht zu bewältigen ist, denn Sie stehen vor zahlreichen Herausforderungen:

- Wie viele Dateien und Festplatten benötigen Sie zur Speicherung?
- Wie können Sie Daten finden und abrufen?
- Wie stellen Sie angemessene Reaktionszeiten sicher?
- Wie erlauben Sie Änderungen und gleichzeitigen Zugriff auf die Daten?
- Wie verhindern Sie den unbefugten Zugriff auf die Daten?

Wenn Sie ein DBS als konstituierendes Rückgrat Ihrer Webanwendung einsetzen (vgl. Abbildung 2.1), lassen sich diese Herausforderungen bewältigen, da DBS die entsprechenden Funktionen bieten, um sie zu bewältigen.




Abbildung 2.1: DBS als Rückgrat von Webanwendungen.

Ein DBS umfasst:

1. die Software zur Unterstützung bei der Verwaltung und Verarbeitung großer Datensammlungen, die von einer Webanwendung genutzt werden können, genannt Datenbankmanagementsysteme (DBMS)
2. die eigentliche Speicherung der Daten selbst, genannt Datenbank (), welche die DB-Schicht bildet, auf der die Webanwendung aufbaut.

Im Folgenden werden die wichtigsten Konzepte, die für den Aufbau der DB-Schicht, d.h. die *Entwicklung einer DB*, benötigt werden, näher erläutert. Obwohl es letztlich um das "*Anlegen geeigneter DB-Tabellen, Speichern von Daten, Manipulieren und Abfragen*" geht, müssen einige wichtige Konzepte eingeführt werden, um dieses Ziel zu erreichen.

Insbesondere beginnen wir in Abschnitt 2.1 mit dem Prozess der Modellierung der zu speichernden Daten, indem wir zwei verschiedene *Abstraktionsebenen* einführen, bestehend aus dem *logischen Modell* und dem *physikalischen Modell* im Sinne des *relationalen Modells*[1]. Im weiteren Verlauf des Abschnitts 2.2 werden DBS-spezifische Mechanismen und Werkzeuge zur *Erstellung der Datenstruktur innerhalb des DBS*, wie sie durch den Modellierungsprozess vorgegeben sind, in den Mittelpunkt gestellt. Darauf aufbauend befasst sich der Abschnitt 2.3 mit Fragen der Datenspeicherung und -manipulation und der Abschnitt 2.4 mit der Abfrage von gespeicherten Daten. Schließlich bietet der Abschnitt 2.5 eine Zusammenfassung aller Optionen und entsprechender Werkzeuge, die für den Aufbau der DB-Schicht einer Webanwendung zur Verfügung stehen.

In all diesen Abschnitten wird ein *laufendes Beispiel* verwendet, das sich an dem oben vorgestellten Anwendungsfall von Walmart orientiert und dessen *Human Resource (HR)*-Aspekt im Hinblick auf *Mitarbeiter*, *Abteilungen* und *Jobs* hervorhebt, ein Standardbeispiel, das auch von ORACLE seit der ersten Version seiner DBS im Jahr 1979 zu Demonstrationszwecken verwendet wird.

## 2.1 Wie man die Daten modelliert

Der erste Schritt bei der Realisierung der DB-Schicht einer Webanwendung besteht darin, sich auf die realen Daten zu konzentrieren, die gespeichert werden sollen. Da diese Daten recht umfangreich sein können, wie wir in unserem obigen Walmart-Beispiel gesehen haben, ist es sinnvoll, sich zunächst mit den Daten auf einer abstrakteren Ebene zu befassen und noch nicht alle Low-Level-Implementierungsdetails eines bestimmten DBS zu berücksichtigen. Dies ist der Schwerpunkt des logischen Modells, das in Abschnitt 2.1.1 behandelt wird. Danach, auf einer konkreteren und damit DBS-spezifischen Ebene, wird das relationale Modell in Abschnitt 2.1.2 vorgestellt.

### 2.1.1 Logisches Modell

**Motivation**. Ein *logisches (Daten-)Modell* beschreibt "Dinge" der *realen Welt* (d.h. der *Problemdomäne*), über die z.B. eine Organisation wie Walmart Daten sammeln und verarbeiten möchte. Logische Modelle werden mit Hilfe von Diagrammen visuell beschrieben, in der Praxis oft in Form von sogenannten *Entity-Relationship (ER)-Diagrammen*, ohne die Besonderheiten eines bestimmten DBS zu berücksichtigen. Die Vorteile dieser visuellen Darstellung sind vielfältig, z. B. die Verbesserung der Verständlichkeit und die Erleichterung der Kommunikation innerhalb eines Entwicklungsteams und mit dem Kunden.

Für die Modellierung von ER-Diagrammen gibt es zahlreiche verschiedene grafische Formalismen und Werkzeuge. Wir werden uns hier nur auf den *ORACLE SQL Developer Data Modeler (ODM)*, konzentrieren. Abbildung 2.2 zeigt ein ER-Diagramm für unser HR-Beispiel, das im Folgenden näher erläutert wird.

[1] Beachten Sie, dass für die Realisierung des physischen Modells nicht nur das relationale Modell verwendet werden kann, das im Mittelpunkt dieses Kapitels steht, sondern auch andere Datenrepräsentationsformate (auch bekannt als "Datenmodelle") zur Verfügung stehen, wie z.B. das *objektorientierte Datenmodell* oder *NoSQL-Datenmodelle*.




Abbildung 2.2: Logisches Modell unseres laufenden HR-Beispiels mit Hilfe eines ER-Diagramms

Das gleiche Beispiel ist in den Abbildungen 2.3, 2.4 und 2.5 dargestellt, diesmal jedoch innerhalb des ODM von ORACLE. Alle beschriebenen Konzepte sind mit Anmerkungen versehen, um die wichtigsten grafischen Notationen und Symbole zu erläutern.

**ER-Diagramme - Entitäten, Attribute, Beziehungen.** *ER-Diagramme* beschreiben die interessanten Konzepte und bestehen aus (i) *Entitäten* (z. B. Abteilungen, Mitarbeiter, Jobs) (ii) *Attributen* (z. B. "Abteilungsname" und "Standortname" von Abteilungen oder "Nachname" und "Gehalt" von Mitarbeitern) ihren Merkmalen und (iii) *Beziehungen* (z. B. eine Abteilung "umfasst" mehrere Mitarbeiter) zwischen Entitäten.

**Entitäten.** Einer *Entität* ist eine eindeutige Substantivphrase als Name zugewiesen und wird visuell als abgerundetes Rechteck dargestellt. Es werden sowohl Plural- als auch Singularformen verwendet, wobei in letzter Zeit die Singularform bevorzugt wird (und auch durch ISO 11179-5 standardisiert ist).

**Attribute.** Attribute sind immer mit Entitäten verbunden, d. h. sie können nicht unabhängig davon existieren. Im Gegensatz zu Entitäten ist ein *Attribut* "*atomar*", was bedeutet, dass es nicht wie eine Entität weiter strukturiert werden kann, sondern als singuläre Substantivphrase dargestellt wird. Jedem Attribut kann eine bestimmte *Domäne* (auch bekannt als *Datentyp*) zugeordnet werden, der die "Art" der zulässigen Werte angibt. Die gebräuchlichsten sind numerisch (`NUMERIC`), Zeichen (`VARCHAR`) und Datum/Uhrzeit (`DATE/TIME`).

Neben der Charakterisierung einer Entität können bestimmte Attribute dazu dienen, *jede Instanz* einer Entität (z. B. einen konkreten Mitarbeiter oder eine konkrete Abteilung) *eindeutig zu identifizieren*. Solche identifizierenden Attribute fungieren als sogenannter "*Primary Key (PK)*". Für eine Abteilung kann z.B. eine künstliche ID als PK definiert werden (die numerisch sein sollte, z.B. eine "departmentID") oder alternativ die Kombination bestehender Attribute wie "departmentName" und "location", wodurch ein sogenannter "*composite PK*" entsteht. Es ist ratsam, für jede Entität einen PK zu definieren, da sonst z. B. Mitarbeiter oder Abteilungen mit demselben Namen beim Abrufen der Daten nicht unterschieden werden können. Jeder Wert eines PK-Attributs muss also eindeutig sein und natürlich auch existieren.

Schließlich ist zu erwähnen, dass neben Domänen und PKs auch andere so genannte *constraints* für Datenwerte für bestimmte Attribute angegeben werden können, z. B. `NOT NULL`, d. h. "obligatorisch", um das Vorhandensein eines Wertes (z. B. für ein Attribut Gehalt) zu verlangen.

**Beziehung.** Eine *Beziehung* schließlich verbindet (in den meisten Fällen) zwei verschiedene Entitäten, wird visuell durch eine Linie zwischen den Rechtecken der Entitäten dargestellt und durch eine Verbphrase über der Linie benannt. Es gibt vor allem drei verschiedene Arten von Beziehungen, die in der Praxis häufig verwendet werden und sich durch die *Anzahl der Entitätsinstanzen* unterscheiden, die an jedem Ende der Beziehung beteiligt sein dürfen, die als "*Kardinalität*" bezeichnet wird (und somit eine weitere Art von Datenbeschränkung darstellt, die angegeben werden kann):

- **One-to-Many ("1:N"):** Eine Entitätsinstanz kann mit vielen Entitätsinstanzen auf der "many-Seite" der Beziehung in Beziehung stehen (z. B. hat eine Abteilung viele Mitarbeiter, d. h. eine "haveEmployees"-Beziehung), und eine Entität auf der "many-Seite" der Beziehung (ein Mitarbeiter) kann nur mit einer Entität auf der "one-Seite" (einer Abteilung) in Beziehung

stehen. Eine solche Beziehung wird als ”*Source-Target*” oder ”*Parent-Child*” *Beziehung* bezeichnet, wobei die ”Source”-Einheit auf der ”one-Seite” als ”Parent” und die beliebige Anzahl von ”Target”-Einheiten auf der ”many-Seite” als ihre ”Childs” fungieren. Im obigen Beispiel ist die Abteilung die ”Parent”-Einheit und die Mitarbeiter sind die ”Child”-Einheiten.

- **Many-to-Many (”M:N”):** Eine ”Source”-Entitätsinstanz kann mit *vielen ”Target”-Entitätsinstanzen* und eine ”Target”-Entitätsinstanz kann mit *vielen ”Source”-Entitätsinstanzen* verbunden sein. Ein Beispiel wären Mitarbeiter, die Jobs haben (d.h. eine ”haveJobs”-Beziehung), da ein Mitarbeiter im Laufe der Zeit mehr als einen Job haben kann und ein bestimmter Job wahrscheinlich von vielen Mitarbeitern ausgeübt wird. Es ist zu beachten, dass eine ”M:N”-Beziehung immer zwei ”1:N”-Beziehungen hervorbringt, wenn wir in der Datenbankmodellierung fortschreiten.
- **One-to-One (”1:1”):** Als letztes kann eine Entitätsinstanz auch nur mit *einer einzelnen Entitätsinstanz* am anderen Ende der Beziehung in Beziehung stehen und umgekehrt (z. B. eine Person ”wohnt an” einer bestimmten Adresse und diese Adresse ist nur die Adresse dieser bestimmten Person). Es ist jedoch anzumerken, dass diese Art von Beziehung häufig durch eine kompaktere Darstellung innerhalb einer einzigen Entität begünstigt wird (z. B. umfassen Entitätspersonen auch Adressattribute wie ”Straße” und ”Ort”) und daher in diesem Kapitel nicht weiter betrachtet wird.
- **Optionalität:** Für jede dieser drei verschiedenen Arten von Beziehungen ist es auch möglich, an jedem Ende für Source und Target anzugeben, ob das *Vorhandensein von mindestens einer Entitätsinstanz optional ist oder nicht*. Wird z.B. eine neue Abteilung gegründet, könnte es sein, dass für diese Abteilung noch keine Mitarbeiter eingestellt sind, d.h. das Ziel der ”haveEmployees”-Beziehung muss als optional definiert werden. Dies wird auch durch die gestrichelte Linie in ODM dargestellt.

Beziehungen werden in ODM einfach als ”Relationen” bezeichnet, die nicht mit den ”Relationen” des ”relationalen Modells” verwechselt werden sollten, die eigentlich ein Synonym für ”Tabellen” sind (vgl. Abschnitt 2.1.2). Daher halten wir uns weiterhin an den gebräuchlicheren Begriff ”Beziehung”.

**GUI-Elemente des ODM zur Verwaltung von ER-Diagrammen.** Die folgenden Abbildungen geben einen ersten Eindruck von den wichtigsten GUI-Elementen des ODM. Abbildung 2.3 veranschaulicht zunächst die Definition neuer Entitäten und neuer Beziehungen sowie die Visualisierung des logischen Modells in Form einer Listenansicht und einer grafischen Ansicht. In der grafischen Ansicht werden die wichtigsten grafischen Notationen und Symbole annotiert.

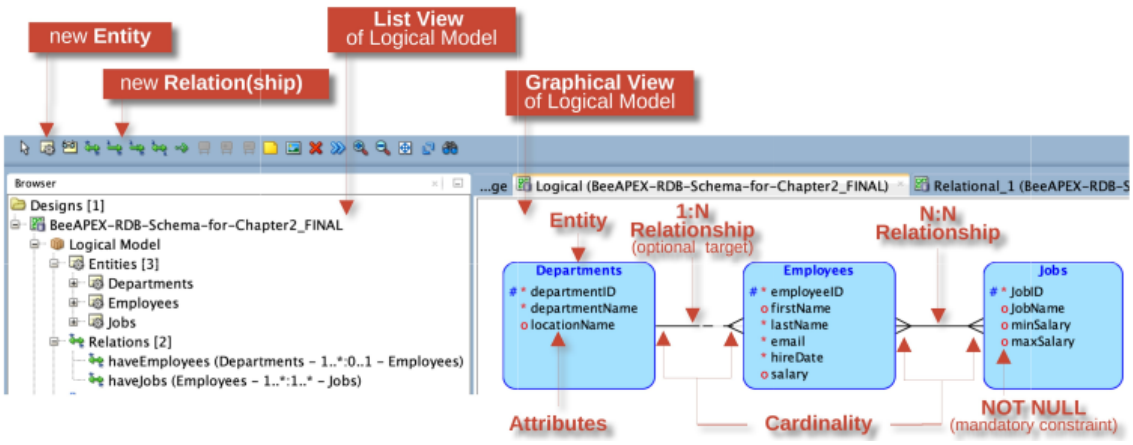



Abbildung 2.3: Verwaltung des logischen Modells mit ODM.

Abbildung 2.4 zeigt den Dialog für die Definition von Attributen, der durch einen einfachen Doppelklick auf eine bestimmte Entität geöffnet werden kann.

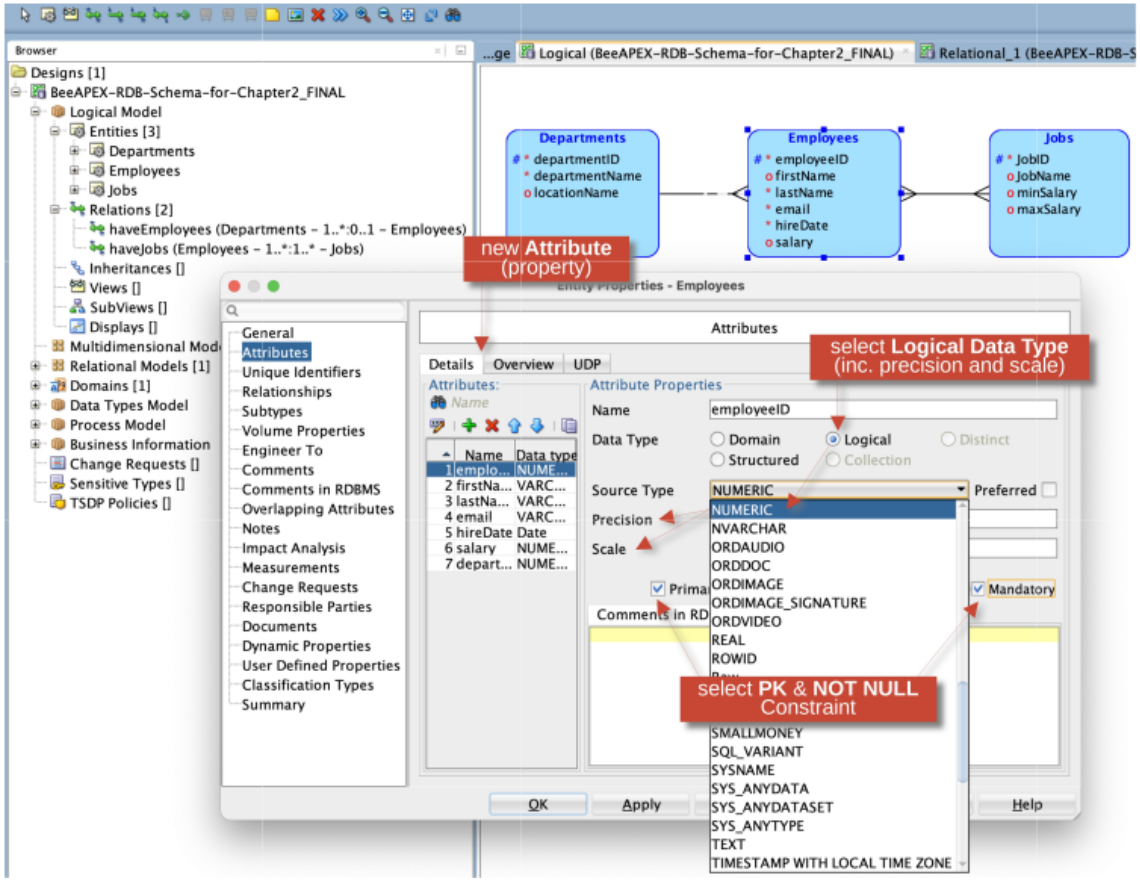



Abbildung 2.4: Attribute mit ODM verwalten.

Abbildung 2.5 zeigt den Dialog für die Definition von Beziehungen, der durch einfaches Doppelklicken auf eine bestimmte Zeile, die die Beziehung darstellt, gestartet werden kann.

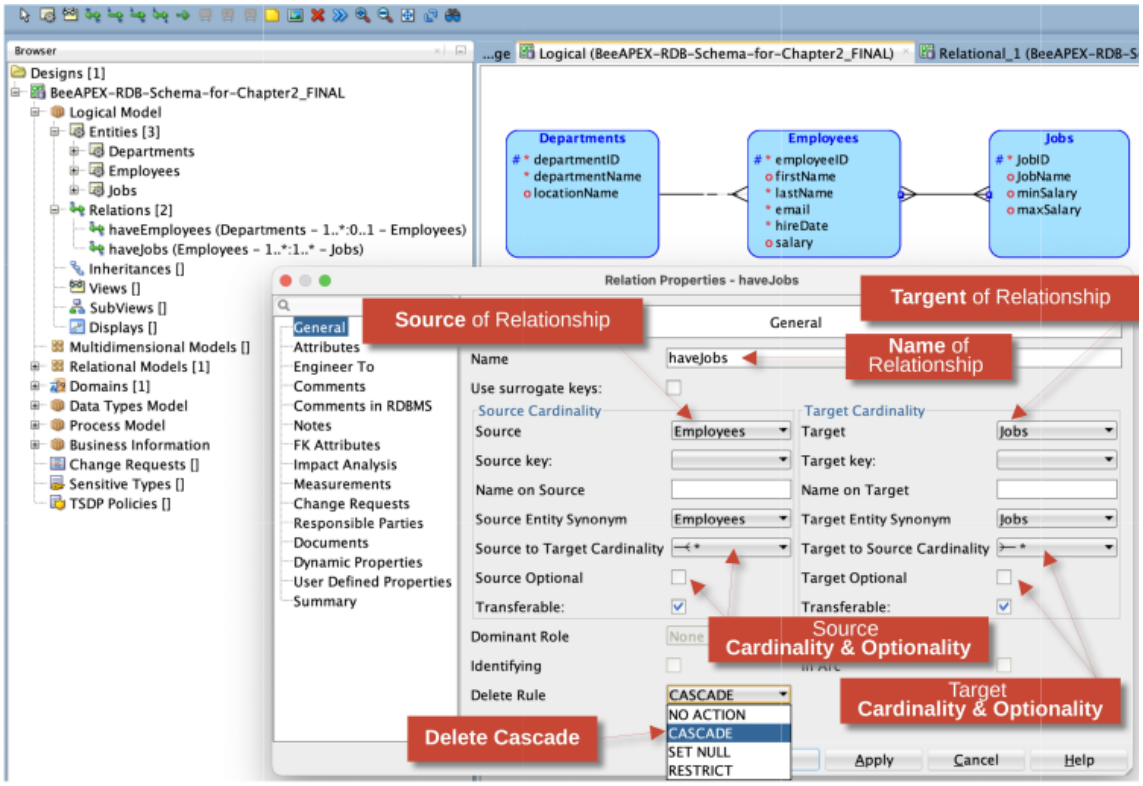



Abbildung 2.5: Beziehungsmanagement mit ODM.

### 2.1.2 Relationales Modell

**DB Schema.** Das logische Modell in Form eines ER-Diagramms, das unabhängig von einem bestimmten DBS ist, bildet die Grundlage für die Ableitung der DBS-spezifischen Struktur der DB, die als "*DB-Schema*" bezeichnet wird. Innerhalb eines spezifischen DBS ist es möglich, eine beliebige Anzahl verschiedener Schemata für verschiedene Benutzer und/oder Bereiche zu erstellen und zu speichern, z.B. eines für die Verwaltung von Personaldaten und ein anderes Schema für die Verwaltung von Daten über Produktverkäufe. Ein Schema funktioniert ähnlich wie ein Ordner in einem Dateisystem und ermöglicht die Gruppierung von Daten für einen bestimmten Zweck.

**Relationales DB-Schema.** Eine DB-spezifische Art von Schema ist das so genannte *Relationale DB (RDB)-Schema*, welches das *Relationale Modell (RM)* als Basisformalismus zur Beschreibung der Datenstruktur verwendet. Das RM ist recht einfach, da es das DB-Schema als eine Sammlung von *Relationen* alias *Tabellen* darstellt, die dem Konzept der Entitäten ähneln. Tabellen werden auf einfache Weise von Entitäten abgeleitet. Analog zu einer Entität besteht eine Tabelle aus einem Namen, *Spalten* (Ähnlich wie *Attribute*), ihren Datentypen und weiteren (optionalen) Einschränkungen (Constraints). Alle Tabellen bilden das *RDB-Schema* (vgl. Abbildung 2.6). Das Schema einer Tabelle ist weiterhin das "Fahrzeug" zur Speicherung der eigentlichen Daten, während jede Zeile in einer Tabelle als "*Tupel*" bezeichnet wird, das aus einer Sammlung zusammengehöriger Datenwerte besteht, die in den Spalten gespeichert sind, und somit eine reale *Entitätsinstanz* in Form von *relationalen* (*tabellarischen*) Daten darstellt.

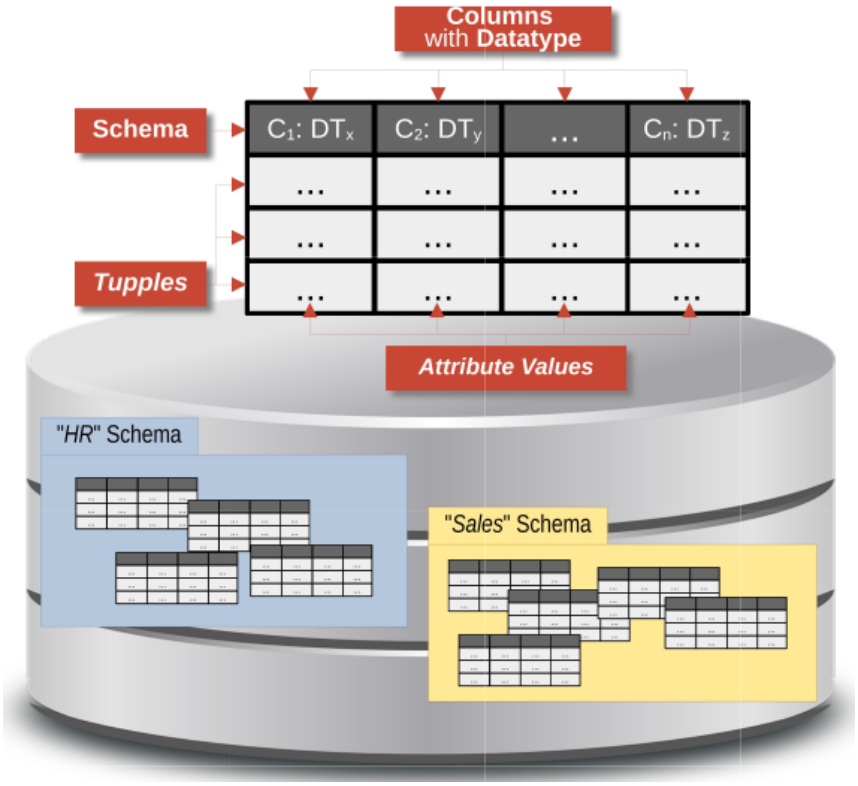



Abbildung 2.6: RDB-Schemata und Tabellenschemata

Interessant ist, dass *ODM* es ermöglicht, automatisch ein RDB-Schema aus dem logischen Modell in Form von Tabellen zu generieren, das die Entitäten unseres logischen Modells und alle Attribute und Beschränkungen jeder Entität widerspiegelt. Dies geschieht über den Menüeintrag *"Engineer to Relational Model"*, wie in Abbildung 2.7 dargestellt. Das sich daraus ergebende RDB-Schema wird auch grafisch visualisiert, indem eine Art ER-Diagramm-Formalismus verwendet wird, der auf einer etwas anderen, "tabellenorientierteren" Notation als das ER-Diagramm des logischen Modells basiert und nun z.B. auch die Datentypen aller Attribute anzeigt.

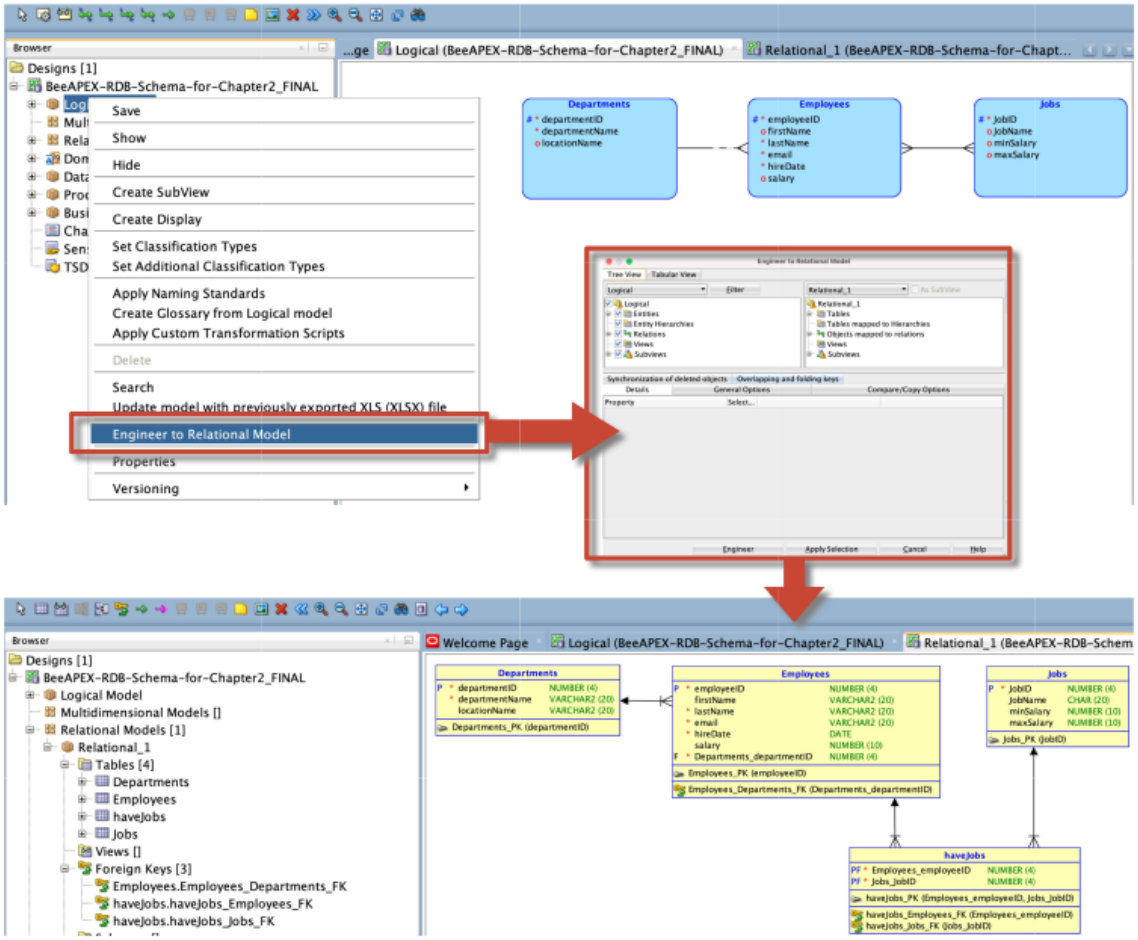



Abbildung 2.7: Prozess zur RDB-Schema-Generierung aus dem logischen Modell.

Betrachten wir nun unser beispielhaftes relationales Modell, das in Abbildung 2.8 dargestellt ist.

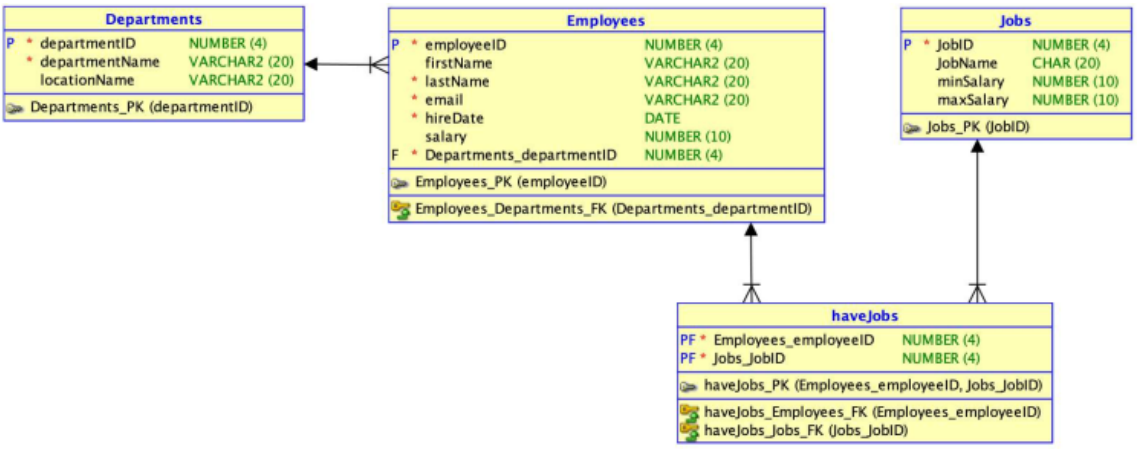



Abbildung 2.8: Automatisch generiertes RDB-Schema.

Zunächst einmal werden bei der Generierung des relationalen Modells aus dem logischen Modell neben der Umwandlung der vorhandenen Entitäten und Attribute in Tabellen und entsprechende Spalten automatisch PKs zu jeder Tabelle hinzugefügt und mit einem Namen versehen (vgl. z.B. ”Departments_PK (departmentID)”). Darüber hinaus werden die Beziehungen zwischen den

Entitäten innerhalb des logischen Modells automatisch in den Tabellen dargestellt, wobei das so genannte "*Foreign-Key (FK)*" *Konzept* auf folgende Weise dargestellt wird:

- **One-to-Many (1:N)-Beziehung:** Innerhalb der Tabelle auf der "Many-Seite" der Beziehung (d.h. der Child-Tabelle) wird ein zusätzliches Attribut eingeführt und als FK deklariert, das als eine Art Referenz auf die Werte des PK der Beziehung auf der "One-Seite" (d.h. der Parent-Tabelle) fungiert. Wenn es z.B. einen Mitarbeiter mit der PK ID 4711 gibt, dann muss im FK-Attribut der zugehörigen Abteilung auch der Wert 4711 stehen. Folglich muss das FK-Attribut den gleichen Datentyp haben wie der PK. In unserem Beispiel wird automatisch ein FK "Departments_departmentID" sowie ein FK mit dem Namen "Employees_Departments_FK" erzeugt. Folglich sorgt der FK jedes Mal, wenn Daten innerhalb dieser beiden Beziehungen im DBS manipuliert werden, für *Konsistenz* zwischen diesen Beziehungen, d. h. für korrekte Beziehungsdaten. Wird z. B. ein neuer Mitarbeiter in die Tabelle Mitarbeiter eingefügt, stellt das DBS sicher, dass er auch einer bestehenden Abteilung zugeordnet wird. Wenn Sie versuchen, eine Abteilung zu löschen, lehnt das DBS diesen Vorgang standardmäßig ab, falls es noch zugehörige Mitarbeiter gibt. Sie können dieses Standardverhalten ändern, indem Sie bei der Angabe des FK die so genannte "*Cascade-Delete Option*" verwenden, was zu einer automatischen Löschung (eventuell "Entlassung") aller mit der gelöschten Abteilung verbundenen Mitarbeiter führt.
- **Many-to-Many (M:N)-Beziehung:** Eine Many-to-Many-Beziehung zwischen zwei Entitäten wird durch die Einführung einer weiteren, dritten Tabelle realisiert, die als eine Art "*Mapping-Tabelle*" fungiert, in der hauptsächlich die Verweise auf jeden der PKs der verbundenen Entitäten in Form von zwei FK-Attributen gespeichert werden. In unserem Beispiel mit Mitarbeitern und Stellen wird automatisch eine dritte Relation mit dem Namen "haveJobs" erzeugt, die für die Verwaltung der korrekten Assoziationen zwischen Mitarbeitern und ihren jeweiligen Stellen zuständig ist. Der PK dieser Zuordnungstabelle ist ein zusammengesetzter PK, der aus den PKs der beiden verbundenen Tabellen besteht, d.h. "haveJobs_PK(Employees_employeeID, Jobs_jobID)". Jedes Attribut dieses PK wird auch zur Definition eines entsprechenden FK verwendet, der jeweils auf den PK der entsprechenden zugehörigen Tabelle verweist (d.h. "haveJobs_Employees_FK (Employees_employeeID)" und "haveJobs_Jobs_FK(Jobs_JobID)"). Es ist zu beachten, dass diese Zuordnungstabelle auch eine beliebige Anzahl zusätzlicher Attribute enthalten kann, die die Beziehung weiter charakterisieren, z.B. "startDate" und "endDate" eines Jobs.
- **Optionalität:** Standardmäßig ist für jede dieser Beziehungen das Vorhandensein mindestens einer einzigen Source- und einer einzigen Target-Instanz optional. Sollte sie obligatorisch sein, so muss eine weitere NOT NULL-Beschränkung mit dem entsprechenden FK-Attribut verbunden werden. In unserem Beispiel benötigen alle drei FKs eine NOT NULL-Beschränkung, da (i) Mitarbeiter mit einer bestimmten Abteilung (ii) sowie mit mindestens einer Stelle verbunden sein müssen und (iii) eine Stelle mit mindestens einem Mitarbeiter verbunden sein muss.

**GUI-Elemente des ODM für die Verwaltung des RM.** Die folgenden Abbildungen geben einen Überblick über die wichtigsten GUI-Elemente des *ODM* zur Verwaltung des RDB-Schemas, die denen zur Verwaltung des logischen Modells sehr ähnlich sind. Abbildung 2.9 veranschaulicht zunächst die Definition neuer Tabellen und neuer FKs sowie die Visualisierung des RM in Form einer Listenansicht und einer grafischen Ansicht. In der grafischen Ansicht werden die wichtigsten grafischen Notationen und Symbole annotiert.

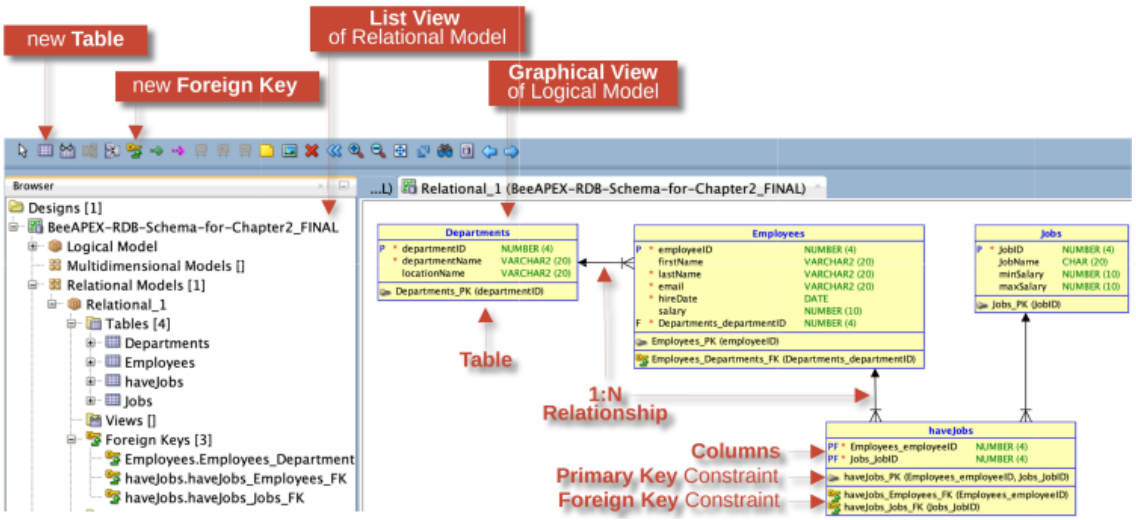



Abbildung 2.9: GUI-Elemente zur Verwaltung von Spalten des RM.

Abbildung 2.10 zeigt den Dialog für die Definition von Spalten, der durch einen einfachen Doppelklick auf eine bestimmte Tabelle gestartet werden kann. Beachten Sie, dass dieser Dialog weitgehend dem Dialog zur Verwaltung von Attributen ähnelt, der bereits in Abbildung 2.4 dargestellt ist.

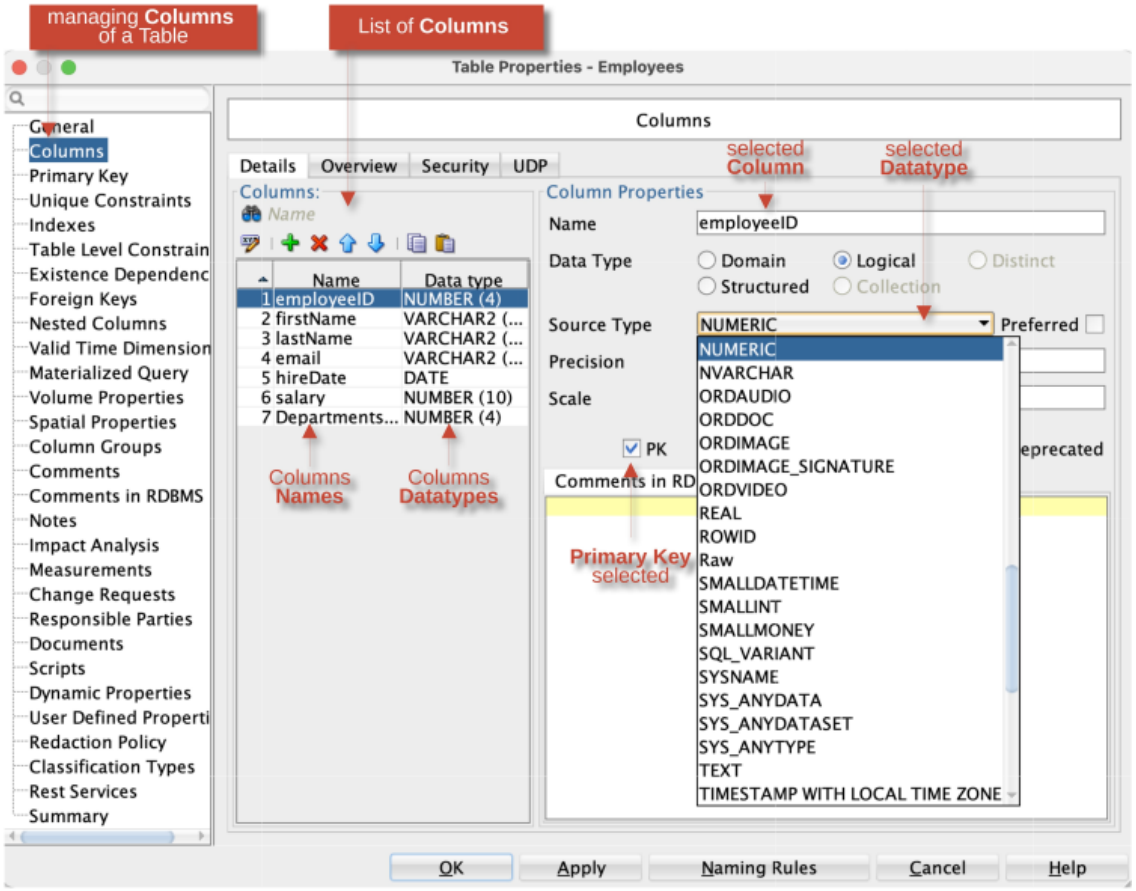



Abbildung 2.10: GUI-Elemente für die Verwaltung des RM.

**RDB-Schema unabhängig von physischen Speicheraspekten.** Obwohl ein RDB-Schema

auf Basis des relationalen Modells bereits konkreter und DBS-spezifischer ist als das logische Modell (z.B., die ”Implementierung” von Beziehungen auf der Basis des FK-Konzepts), so bleibt doch die Komplexität der zugrundeliegenden Speichermechanismen des DBS (auch bekannt als ”*physical*” *level*) verborgen, z.B. wie viele Dateien für die Speicherung der Tabellen verwendet werden oder auf welchen Servern die Daten gespeichert werden sollen. Ein weiterer großer Vorteil dieser Abstraktion von den physischen Speicheraspekten ist, dass neue RDB-Schemata realisiert oder bestehende geändert werden können, ohne dass man sich um die zugrunde liegende physische Datenorganisation kümmern muss.

### 2.1.3 Normalisierung eines RDB-Schemas

Ein RDB-Schema, das aus einem logischen Schema abgeleitet wurde, kann oft weiter verbessert werden, um bestimmte *Qualitätskriterien* zu erfüllen, die in DBS vorherrschen, wobei es einerseits am besten für den *Lesezugriff auf Daten* in Form von DB-Abfragen geeignet ist, andererseits aber auch eine geeignete Grundlage für die *Datenmanipulation* in Form von Einfügen, Aktualisieren und Löschen von Daten bietet. Obwohl dies in gewisser Weise widersprüchliche Ziele sind, gibt es für RDBS ein bekanntes Konzept im Sinne der sogenannten ”*Normalisierungstheorie*”, das einen systematischen Prozess von drei Schritten namens ”*erste Normalform*”, ”*zweite Normalform*” und ”*dritte Normalform*” beschreibt, der zu verbesserten RDB-Schemata führt (vgl. Abbildung 2.11 für einen Überblick). Im Folgenden werden diese drei Normalformen unter Verwendung leichter Variationen unseres laufenden Beispiels näher beschrieben.

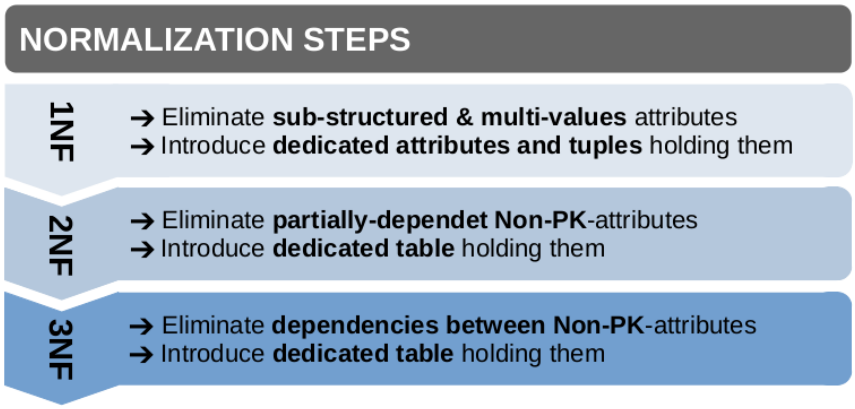



Abbildung 2.11: Normalisierungsschritte - Überblick

**Erste Normalform** Die erste Normalform setzt voraus, dass jedes Attribut einer Tabelle ”atomar” ist, was bedeutet, dass weder (i) *Substrukturen* erlaubt sind (z. B. wenn anstelle unseres einfachen Attributs ”Ortsname” ein alternatives Attribut ”Ort” Werte über Straße, Postleitzahl, Stadt und Land enthalten würde) noch (ii) *Mehrfachwerte* erlaubt sind (z. B. könnte das Attribut ”Ort” auch mehrere Adressen einer bestimmten Abteilung enthalten). Unterstrukturen innerhalb von Werten können *eliminiert* werden, indem die Unterstruktur einfach mit eigenen Attributen dargestellt wird (z. B. für jeden Teil der Adresse) und Mehrfachwerte, indem einfach ein eigenes Tupel für jeden der Werte verwendet wird (d. h. jede Adresse wird in eine eigene Zeile gestellt).

Ein *Hauptvorteil* der ersten Normalform besteht darin, dass Datenabfragen spezifischer sein können, da nun z. B. auf jeden Teil der Adresse sowie auf jedes Adresstupel separat zugegriffen werden kann.

**Zweite Normalform** Die zweite Normalform setzt voraus, dass (i) eine Tabelle in der *ersten Normalform* vorliegt und (ii) im Falle eines zusammengesetzten PK, d.h. eines PK, der aus zwei oder mehr Attributen besteht, alle anderen Nicht-PK-Attributwerte *nur durch den gesamten zusammengesetzten PK eindeutig identifiziert* werden. Nehmen wir zum Beispiel an, dass die Tabelle

Departments einen zusammengesetzten PK mit den Attributen ”departmentName” und ”locationName” enthält. Dieser PK bestimmt korrekt den Wert eines Nicht-PK-Attributs ”annualRevenue”, da wir natürlich davon ausgehen, dass der Wert von ”annualRevenue” sowohl von der Abteilung als auch von ihrem Standort abhängt. Die Werte eines anderen Nicht-PK-Attributs ”locationAddress” würden jedoch bereits von einem Teil des PK abhängen, nämlich dem Namen des Standorts. Um diese *Teilabhängigkeit* zu *beseitigen*, müssen wir die Tabelle ”Abteilungen” *aufteilen*, die Standortinformationen in eine separate Tabelle ”Standorte” auslagern und beide Tabellen über einen FK verbinden.

Ein wesentlicher *Vorteil* der zweiten Normalform ist, dass sie *redundante Datenspeicherung* verhindert, da in unserem Fall eine bestimmte Adresse nur einmal gespeichert wird und von verschiedenen Abteilungen wiederverwendet werden kann. Dies reduziert nicht nur den Speicherplatz, sondern verhindert vor allem auch fehlerhafte Daten, so genannte ”*Anomalien*”, wenn z.B. bestimmte Datenmanipulationen nicht an doppelten Daten vorgenommen werden (z.B. bei einem Umzug mehrerer Abteilungen an eine andere Adresse muss die Adresse nur einmal geändert werden). Das Gute ist schließlich, dass das RDB-Schema bei Tabellen mit nur nicht zusammengesetzten PKs (z.B. eine numerische ID für jede Tabelle) ”automatisch” die zweite Normalform erfüllt, da partielle Abhängigkeiten nicht existieren können.

**Dritte Normalform** Die dritte Normalform setzt voraus, dass (i) eine Tabelle in der *zweiten Normalform* vorliegt und (ii) dass jedes *Nicht-PK-Attribut nur durch den PK* und nicht durch ein anderes Nicht-PK-Attribut eindeutig identifiziert wird, was zu *transitiven Abhängigkeiten* führt. Wenn beispielsweise die Tabelle Abteilungen neben der PK employeeID und dem locationName auch eine locationAddress enthalten würde, dann wäre locationAdress bereits eindeutig durch locationName identifiziert (wenn wir davon ausgehen, dass jeder locationName mit einer anderen Adresse verbunden ist), was nach der dritten Normalform nicht zulässig ist. Um diese Situation zu vermeiden, müssen die Standortinformationen in eine separate Tabelle Standorte ausgelagert und über einen entsprechenden FK mit der Tabelle Abteilungen verbunden werden.

Der *Nutzen* ist derselbe wie bei der zweiten Normalform, nämlich die Verringerung der Redundanz und damit potenzieller Anomalien im Falle von Datenmanipulationen.

## 2.2 Wie man das RDBS-Schema verwaltet - SQL-DDL

Im vorangegangenen Abschnitt wurde das RDB-Schema auf verschiedenen Abstraktionsebenen grafisch modelliert, und zwar in einer Art *Low-Code-Entwicklung*. Nun ist es an der Zeit zu diskutieren, wie diese grafischen Modelle innerhalb einer RDBS ”programmatisch” implementiert werden können, so dass sie tatsächlich zur Verwaltung der eigentlichen Daten einer Web-Anwendung verwendet werden können. Die Fragen, die im Folgenden beantwortet werden sollen, sind daher:

- Wie kann man dem DBS mitteilen, die *erforderlichen Tabellen zu erstellen*, idealerweise aus den grafischen Modellen, und gibt es andere Alternativen, wenn keine solchen grafischen Modelle existieren?
- Wie sieht die *DBS ”Programmiersprache”* aus, mit der die Tabellen erstellt werden können?

**SQL = DDL | DML | DQL.** Um zunächst die zweite Frage zu beantworten: Die Sprache, die dafür und auch für andere Zwecke verwendet wird, ist die *standardized Structured Query Language (SQL)*. SQL bietet eine breite Palette verschiedener *”DBS-Programmieranweisungen”*, die je nach Zweck in verschiedene Kategorien eingeteilt werden können (vgl. Abbildung 2.12), wobei die wichtigsten (i) Anweisungen zur Verwaltung des RDB-Schemas (z.B., Anlegen, Ändern und Löschen von Tabellen), zusammengefasst als *DDL (Data Definition Language)*, auf die sich dieser Abschnitt konzentriert, (ii) Anweisungen zur Manipulation von Daten innerhalb des Schemas (z.B. Einfügen, Aktualisieren und Löschen von Daten innerhalb von Tabellen), genannt *DML (Data Manipulation Language)*, auf die in Abschnitt 2.3 eingegangen wird, und zuletzt (iii) Anweisungen zur Abfrage der Daten, genannt *DQL (Data Query Language)*, auf die in Abschnitt 2.4 näher eingegangen wird.

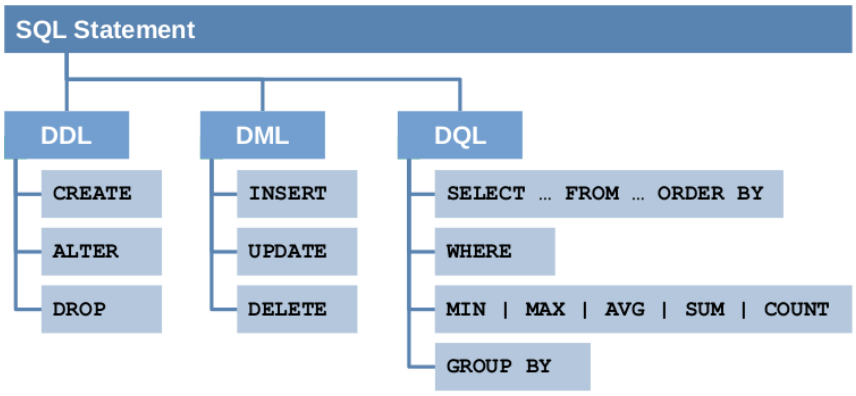



Abbildung 2.12: Kategorien von SQL-Anweisungen

**Fünf Optionen für die Erstellung der Tabellen.** Gemäß den von ORACLE zur Verfügung gestellten Tools gibt es fünf verschiedene Möglichkeiten, Tabellen zu erstellen, d.h. ein RDB-Schema zu implementieren (vgl. auch Abbildung 2.45 für einen Überblick):

1. **Automatische Tabellengenerierung mit ODM.** Dies ist die Option mit dem geringsten Aufwand, da es ausreichen würde, eine *automatische Generierung der Tabellen* durch einige Schaltflächenklicks *mit ODM* zu initiieren (vgl. Abschnitt 2.2.1).
2. **Automatische Tabellengenerierung mit Quick SQL.** Eine weitere Alternative wäre die Verwendung einer Art "Kurzschreibweise" für SQL, wie sie von ORACLEs Quick SQL Tool bereitgestellt wird (vgl. Abschnitt 2.2.2). Dies ist die perfekte Wahl, wenn keine grafischen (logischen und/oder relationalen) Modelle des RDB-Schemas verfügbar sind oder wenn einige schnelle Tests notwendig sind.
3. **Manuelle Tabellenerstellung mit SQL-DDL.** Es gibt natürlich auch die Möglichkeit, die Tabellen programmatisch mit einfachem SQL zu spezifizieren, was den Vorteil hat, dass jedes einzelne Detail der Tabellenspezifikation nach Bedarf konfiguriert werden kann und man nicht von manchmal suboptimalen automatischen Generierungsprozessen abhängig ist (vgl. Abschnitt 2.2.3).
4. **Manuelle Tabellenerstellung mit ORACLE Object Browser (OB).** Für Benutzer ohne SQL-Kenntnisse bietet der Object Browser die Möglichkeit, neue Tabellen anzulegen und diese gegebenenfalls formularbasiert zu ändern (vgl. Abschnitt 2.2.4).
5. **Automatische Tabellenerstellung mit ORACLE Data Workshop.** Schließlich gibt es eine weitere Möglichkeit der automatischen Tabellenerstellung, die jedoch nur genutzt werden kann, wenn bereits Daten in externen Dateien (z.B. in EXCEL-Dateien) gespeichert sind, mit Hilfe von ORACLE Data Workshop. Diese Option wird in diesem Kapitel nicht weiter behandelt, sondern in Kapitel 4 ausführlicher beschrieben.

Im Folgenden werden die ersten vier Optionen näher erläutert.

### 2.2.1 Automatische Tabellengenerierung mit ODM

Als die Tabellengenerierungsoption mit dem geringsten Aufwand sind nur zwei Schritte notwendig, um die benötigten DB-Tabellen zu erstellen, wenn ein RDB-Schema in *ODM* entwickelt wurde.

Zunächst muss das modellierte RDB-Schema in ein *SQL-Skript* exportiert werden (vgl. Abbildung 2.13, Schritte 1 bis 3), wobei es sich um eine einfache Textdatei handelt, die alle SQL-DDL-Anweisungen enthält, die zur Erstellung der Tabellen im RDBS erforderlich sind. Bitte beachten Sie, dass die Syntax der generierten SQL-Anweisungen in Abschnitt 2.2.3 näher beschrieben wird.

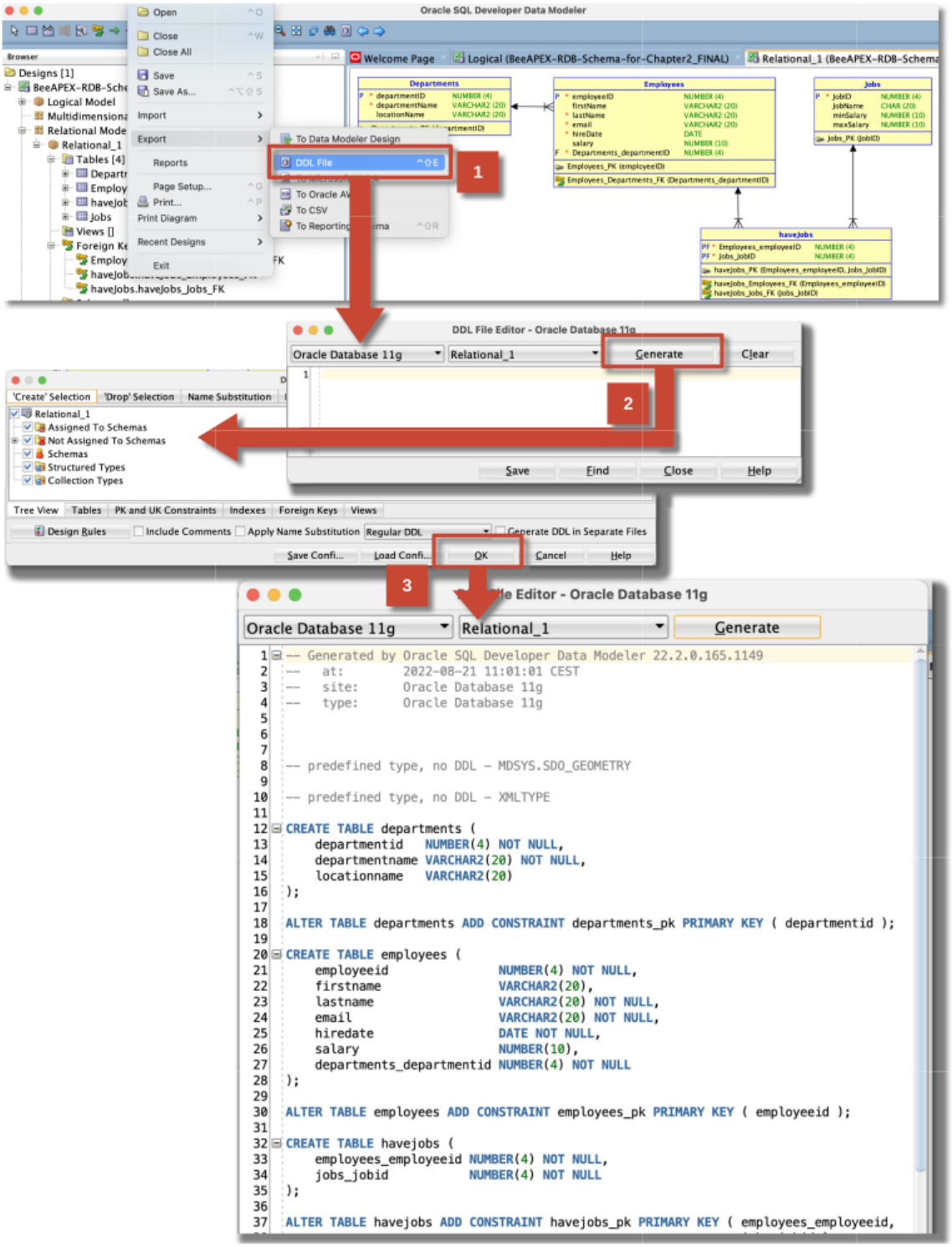



Abbildung 2.13: Export des RDB-Schemas in ein DDL-Skript.

Zweitens muss das generierte SQL-Skript ausgeführt werden, um das RDBS anzuweisen, tatsächlich alle Tabellen als leere Tabellen in seinem Speicherplatz aufzubauen. Dazu muss, wie in Abbildung 2.14 dargestellt, das Skript über ORACLE SQL Workshop hochgeladen (vgl. 1-2) und einfach ausgeführt werden (vgl. 3).

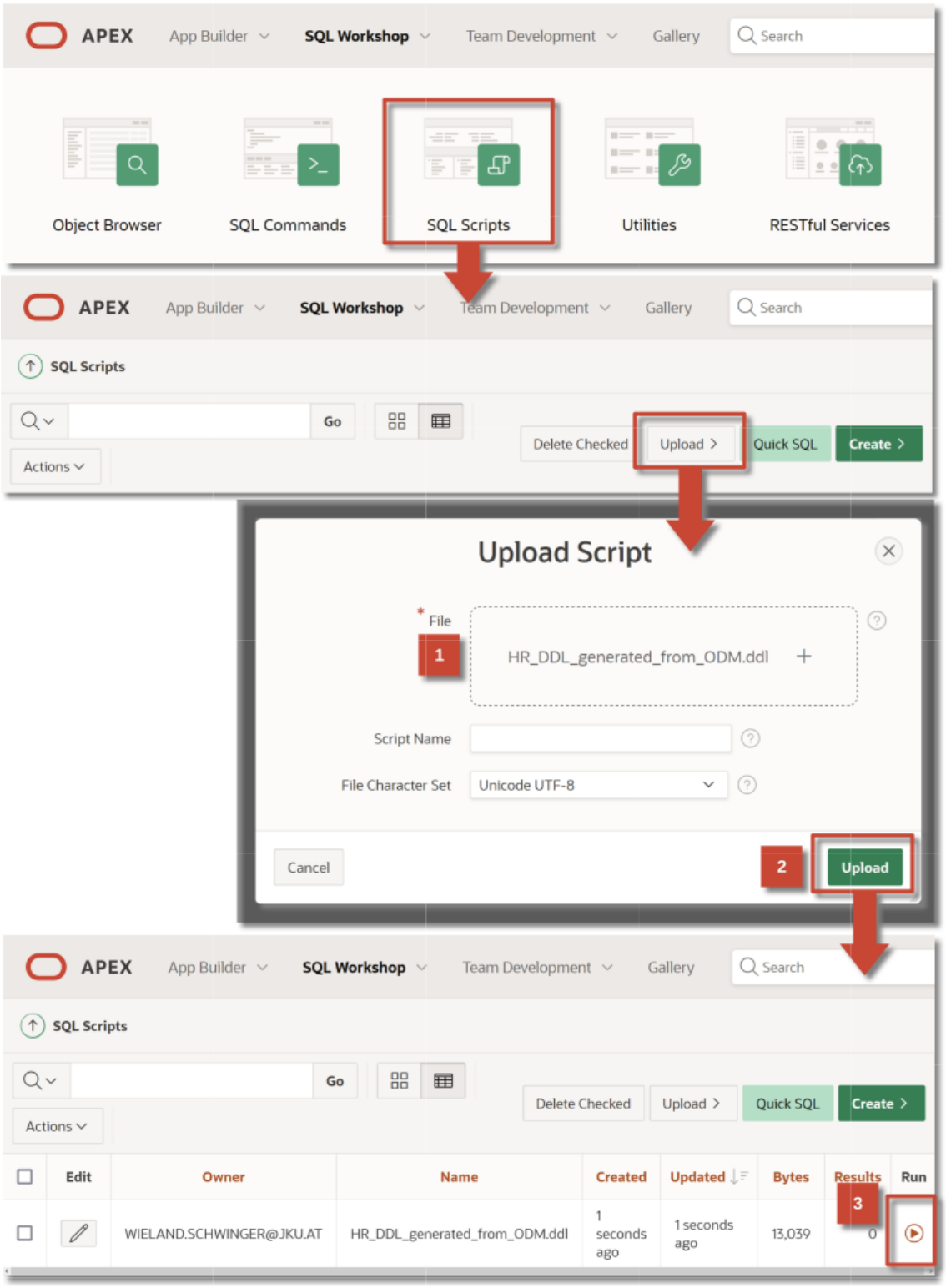



Abbildung 2.14: Hochladen und Ausführen eines SQL-Skripts

### 2.2.2 Automatische Tabellengenerierung mit Quick SQL

Beim Aufbau der DB-Schicht einer Webanwendung ist es sehr ratsam, die verschiedenen Abstraktionsebenen in Bezug auf logische und relationale Modelle zu beachten. Nichtsdestotrotz bietet *Quick SQL* eine einfache und intuitive Alternative für die Erstellung von Tabellen, die wesentlich einfacher ist als SQL, z.B. für schnelle Testzwecke. Quick SQL ermöglicht darüber hinaus die

automatische Generierung eines entsprechenden SQL-Skripts. Sobald das SQL-Skript generiert ist, kann es angepasst und erweitert werden und schließlich ausgeführt werden, um die Tabellen zu erstellen. Es muss noch einmal betont werden, dass Quick SQL zwar dazu dient, den Aufwand für die Erstellung von Tabellen zu verringern, aber keinesfalls als Ersatz für die Datenmodellierung gedacht ist (vgl. Abschnitt 2.1).

Der ORACLE Quick SQL-Editor kann über ”Utilities” vom SQL Workshop aufgerufen werden (vgl. Abbildung 2.15).

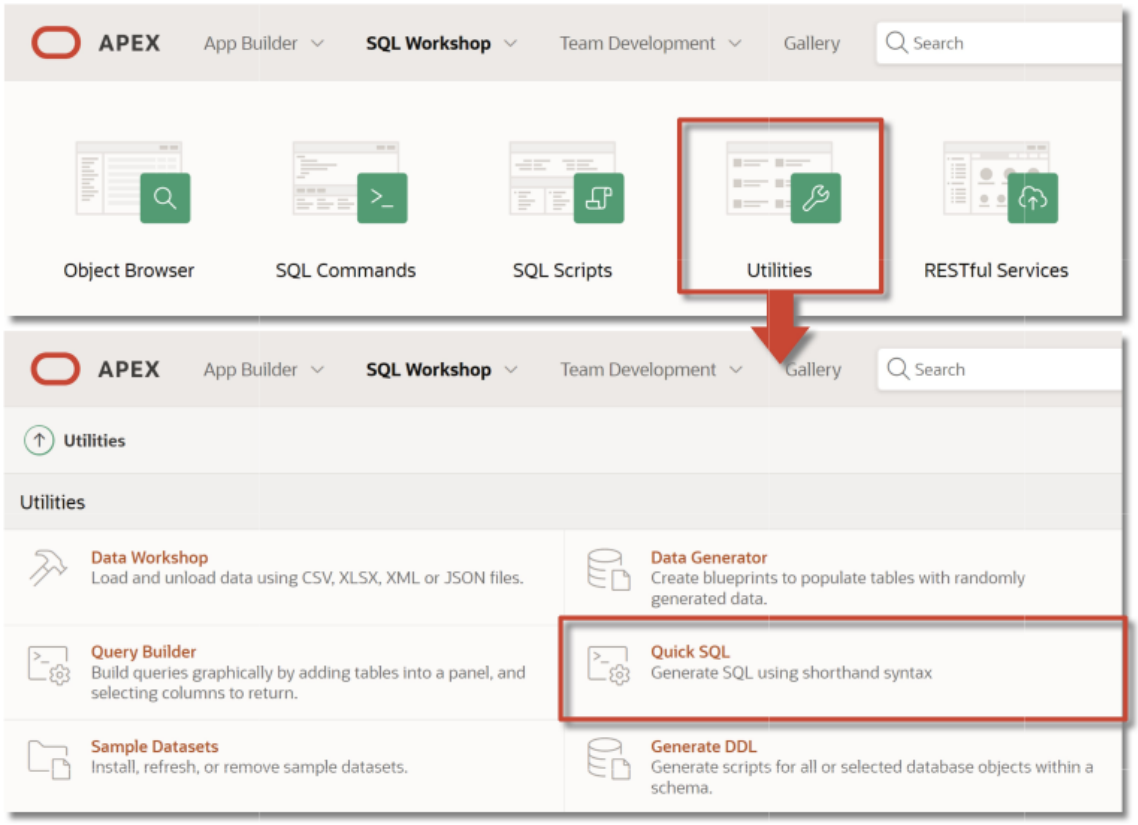



Abbildung 2.15: Zugang zu Quick SQL über SQL Workshop

Bleiben wir nun bei der einfachen Syntax von Quick SQL, die im Wesentlichen einige Formatierungsprinzipien verwendet, um die erforderlichen Tabellengenerationen anzugeben:

- **Parent-Tabellen:** Die Namen der übergeordneten Tabellen werden ohne Einrückung eingegeben. Tabellennamen werden automatisch formatiert, wobei Leerzeichen durch Unterstriche ersetzt werden.
- **Attribute:** Attributnamen werden mit einem einheitlichen Einzug von zwei oder mehr Leerzeichen eingegeben. Wie die Tabellennamen werden auch die Attributnamen automatisch formatiert, wobei Leerzeichen durch Unterstriche ersetzt werden.
- **Parent/Child-Beziehungen:** Parent-Child-Beziehungen werden durch Einrücken von Child-Tabellen unter Parent-Tabellen eingegeben. Untergeordnete Tabellen sollten auf derselben Ebene wie die Spalten der übergeordneten Tabelle eingefügt werden.
- **Datentypen:** Auf der Grundlage des im Spaltennamen enthaltenen englischen Textes und wenn kein Datentyp angegeben ist, leitet das RDBS automatisch eventuell geeignete Datentypen ab. Wenn eine Spalte einen bestimmten Datentyp haben soll, sind die folgenden Typen möglich: `NUM`, `INT`, `VC` (für `VARCHAR2`) oder `DATE`. Der Datentyp kann einfach am Ende eines Spaltennamens angegeben werden, getrennt durch ein Leerzeichen. Wenn eine bestimmte `VC`-Länge definiert werden soll, muss `VCn` angegeben werden, wobei `n` die Länge

des VARCHAR2 ist. Insgesamt wird die verfügbare Syntax über das ”Hilfe”-Menü angezeigt.

Abbildung 2.16 zeigt ein Beispiel für die Quick-SQL-Notation, wobei unsere übergeordnete Tabelle ”Abteilungen” und die entsprechende untergeordnete Tabelle ”Mitarbeiter” auf der linken Seite und das automatisch generierte SQL-Skript im SQL-Ausgabebereich dargestellt sind. Das SQL-Skript wird unmittelbar nach jedem Zeilenumbruch generiert. Das fertige SQL-Skript kann für die weitere Verwendung und eine eventuelle Anpassung gespeichert sowie ausgeführt werden, um die leeren Tabellen endgültig zu erzeugen.

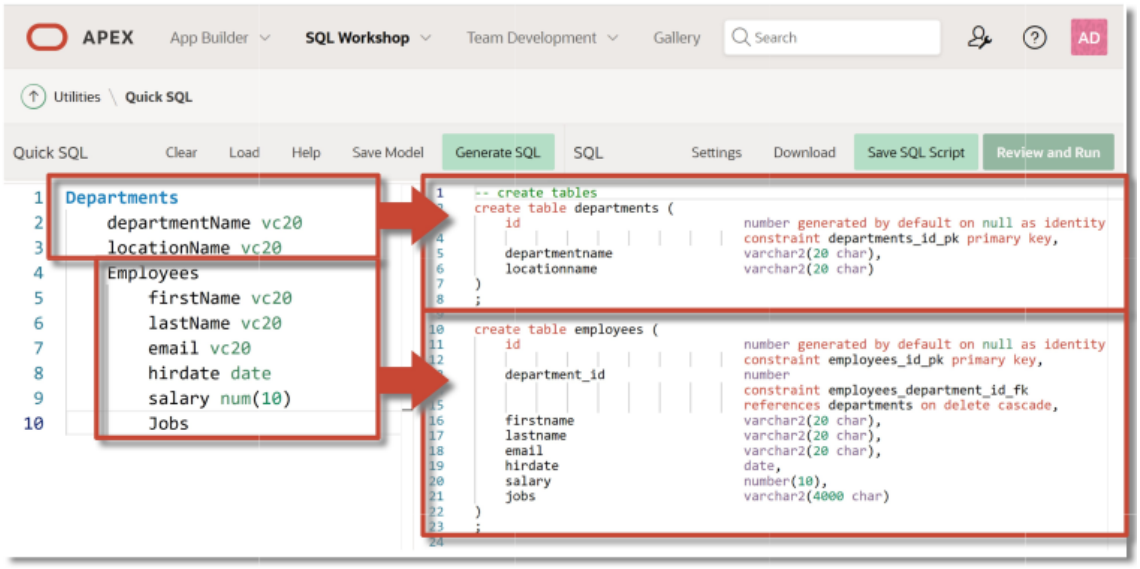



Abbildung 2.16: Erstelltes SQL-Skript auf der Grundlage von Quick SQL.

Quick SQL verfügt auch über eine Syntax zur Definition von Constraints (Fremdschlüssel und zulässige Werte) für bestimmte Datenfelder sowie zur Erzeugung von Testdaten.

### 2.2.3 Manuelle Tabellenerstellung mit SQL-DDL

Der programmatische Weg, Tabellen zu erstellen, besteht darin, sich von der automatischen Generierung von SQL-Skripten zu lösen (wie in Abschnitt 2.2.1 und Abschnitt 2.2.2 besprochen) und die Arbeit selbst zu übernehmen. Um konkreter zu werden, kann man die SQL-Skripte natürlich auch manuell mit der SQL-DDL `CREATE TABLE`-Anweisung im SQL-Command-Editor von ORACLEs *SQL-Workshop* (vgl. Abbildung 2.17) festlegen. Nach der manuellen Definition der SQL-Anweisungen mit Hilfe von einfachem SQL-DDL (1) kann das resultierende SQL-Skript gespeichert und ausgeführt werden (2), um die eigentlichen leeren Tabellen zu erstellen (vgl. Abbildung 2.17).

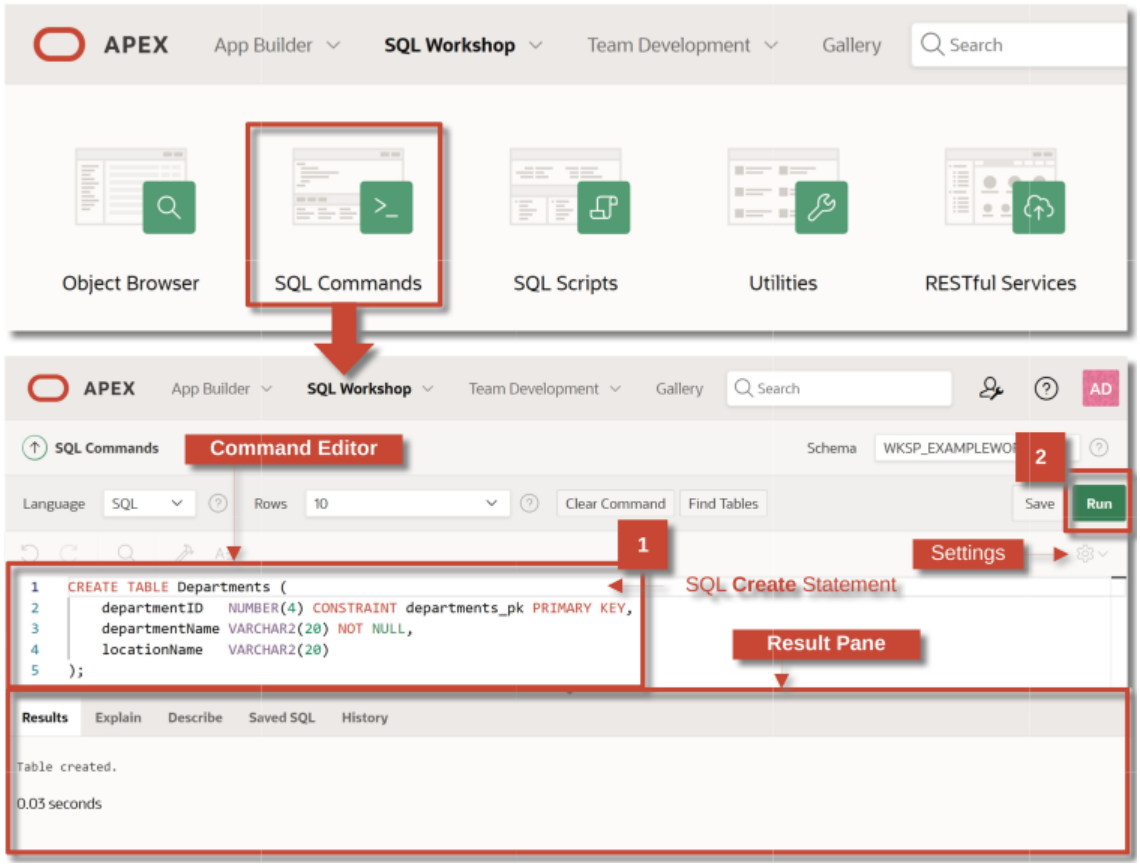



Abbildung 2.17: SQL-Befehlseditor.

SQL-Anweisungen sind mehr oder weniger wie einfaches Englisch, aber mit einer spezifischen Syntax. Die `CREATE TABLE` -Anweisung besteht aus *SQL-spezifischen Schlüsselwörtern* wie "`CREATE`" selbst sowie aus den *constraint-Definitionen* (Datentypen, Primär- und Fremdschlüssel). In Abbildung 2.18 sind alle Schlüsselwörter sowie andere SQL-spezifische syntaktische Symbole (wie Klammern und Kommas) in Großbuchstaben geschrieben und blau eingefärbt. Es ist zu beachten, dass bei SQL-Schlüsselwörtern die Groß- und Kleinschreibung keine Rolle spielt, d. h. "`CREATE`" bedeutet dasselbe wie "`create`". Die Attributdefinition ist in Klammern eingeschlossen, die Attribute sind durch Kommas getrennt, und jedes Attribut wird idealerweise in einer eigenen Zeile definiert (dies sind jedoch nur informelle Formatierungskonventionen und nicht Teil der SQL-Syntax). Einschränkungen können mit einem expliziten Namen definiert werden (z. B. departments_pk), was nicht zwingend erforderlich ist (vgl. z. B. `NOT NULL` für Abteilungsname), sondern lediglich der besseren Lesbarkeit dient. Jede SQL-Anweisung wird am Ende durch ein Semikolon abgegrenzt.

```
CREATE TABLE Departments
(   departmentID   NUMBER(4), CONSTRAINT departments_pk PRIMARY KEY,
    departmentName VARCHAR2(20) NOT NULL,
    locationName   VARCHAR2(20)
);
```

Abbildung 2.18: SQL-Anweisung zum Erstellen der Tabelle `Departments`.

Ergänzend zur `CREATE TABLE`-Anweisung gibt es natürlich auch eine SQL-Anweisung zum Löschen einer Tabelle - Sie funktioniert unabhängig davon, ob die Tabelle leer ist oder ob sie bereits

Daten enthält, die dann ebenfalls gelöscht werden (vgl. Abbildung 2.19).

```
DROP TABLE Departments;
```

Abbildung 2.19: SQL-Anweisung zum Löschen der Tabelle `Departments`.

Und schließlich kann auch die Definition bereits bestehender Tabellen geändert werden, z. B. durch Hinzufügen zusätzlicher Attribute oder Ändern eines Datentyps (vgl. Abbildung 2.20):

```
ALTER TABLE Departments
      ADD annualRevenue NUMBER(10);
```

Abbildung 2.20: SQL-Anweisung zum Ändern der Tabelle `Departments`

Wenn die Tabelle, die geändert werden soll, bereits Daten enthält, dann sind Änderungen etwas eingeschränkt, da z.B. die Änderung eines Datentyps zu unbeabsichtigten Auswirkungen auf diese Daten führen kann. Dieses Problem wird jedoch in diesem Abschnitt nicht weiter behandelt.

### 2.2.4 Manuelle Tabellenverwaltung mit ORACLE Object Browser

Der Object Browser des ORACLE SQL Workshop erlaubt es nicht nur, alle generierten Tabellen, in der ORACLE-Terminologie *"Objects"* genannt, zusammen mit den zugehörigen Attributen und Constraints zu betrachten, sondern auch, diese zu ändern und neue zu erstellen, alles basierend auf einem einfachen GUI-Wizard (vgl. Abbildung 2.21).

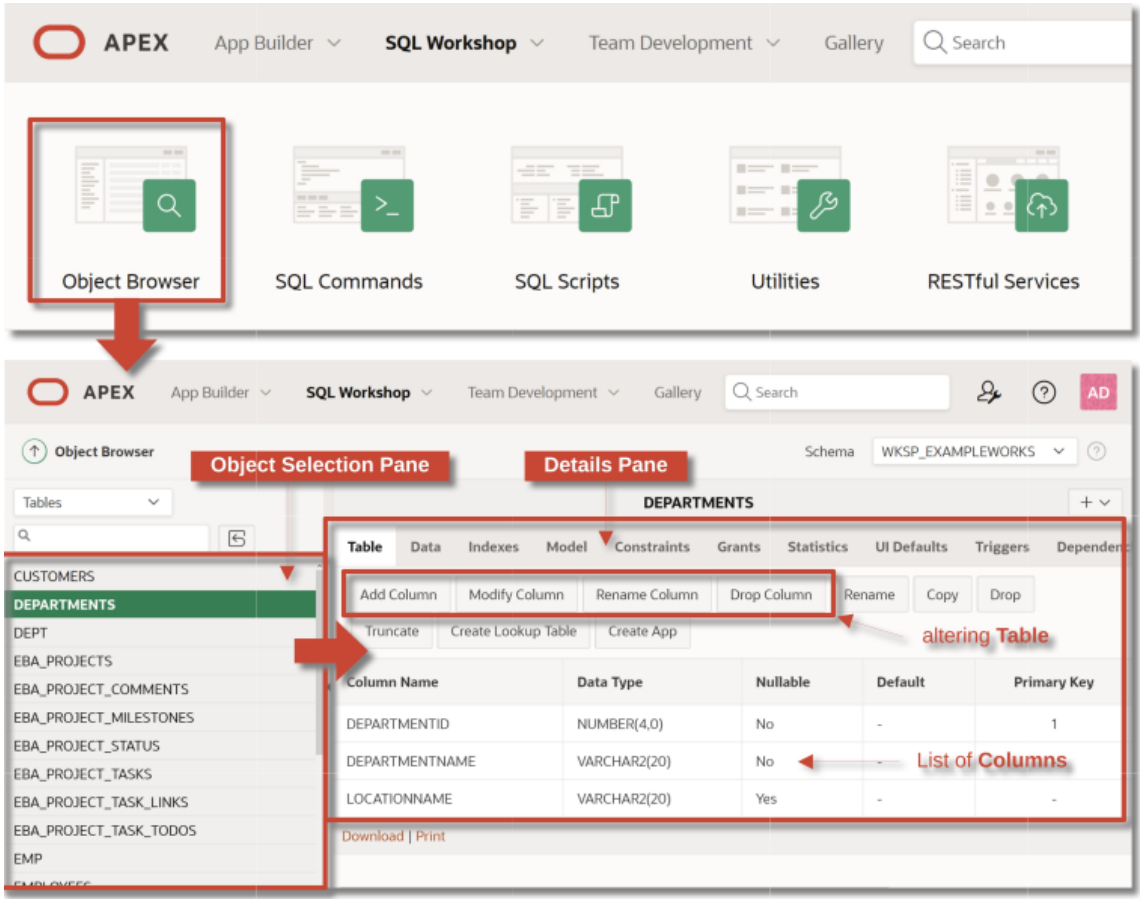



Abbildung 2.21: Tabellenverwaltung mit dem ORACLE Object Browser

## 2.3 Wie man Daten manipuliert - SQL-DML

Die im vorigen Abschnitt angelegten leeren Tabellen ”warten” nun darauf, mit entsprechenden Daten gefüllt zu werden, die dann weiter verändert und schließlich auch gelöscht werden können. Genau dafür sind die SQL-DML-Anweisungen `INSERT`, `UPDATE` und `DELETE` zuständig, die im folgenden Abschnitt 2.3.1 anhand unseres laufenden Beispiels kurz vorgestellt werden. Daneben steht mit dem Object Browser (”OB”) von ORACLE auch eine no-code Alternative zur Verfügung, die das manuelle Einfügen von Daten in Tabellen ermöglicht und für Benutzer ohne SQL-Kenntnisse geeignet ist (vgl. Abschnitt 2.3.2). Eine recht begrenzte Alternative zum Einfügen von Daten bietet schließlich Quick SQL (vgl. Abschnitt 2.3.3). Alle diese Alternativen sind auch in Abbildung 2.45 dargestellt.

### 2.3.1 Datenmanipulation mit SQL-DML

Für die programmatische Datenmanipulation mit SQL-DML wird der SQL-Befehlseditor verwendet (vgl. Abbildung 2.17), der auch für die Erstellung von Tabellen verwendet wurde. Im Folgenden werden einige Beispiele dargestellt, um einen ersten Eindruck über die Syntax und die Funktionalität von SQL-DML zu vermitteln.

`INSERT` wird verwendet, um zu einer Tabelle *neue Zeilen hinzuzufügen*, wie im folgenden Beispiel zu sehen ist (vgl. Abbildung 2.22):

```
INSERT INTO Departments (departmentID, departmentName, locationName)
     VALUES (10. 'ACCOUNTING', 'NEW YORK');
```

Abbildung 2.22: SQL-Anweisung zum Einfügen neuer Daten in die Tabelle `Departments`

.

Neben dieser programmatischen Option haben wir in Abschnitt 2.3.3 auch die Möglichkeit erwähnt, Quick SQL zum Einfügen von Daten zu verwenden, die jedoch auf das Einfügen von Zufallsdaten beschränkt ist.

`UPDATE` wird verwendet, um eine oder mehrere Datenzeilen innerhalb einer Tabelle zu aktualisieren. Das folgende Beispiel ist die einfachste Form einer `UPDATE`-Anweisung, die das Gehalt aller Mitarbeiter um 10% erhöht (vgl. Abbildung 2.23):

```
UPDATE Employees
     SET salary = salary * 1.1;
```

Abbildung 2.23: SQL-Anweisung zum Aktualisieren des Gehalts aller Mitarbeiter um 10% in der Tabelle `Employees`.

Eine leichte Erweiterung zeigt das nächste Beispiel, indem eine sogenannte ”`WHERE`-Klausel” hinzugefügt wird, um die Gehaltserhöhung auf den Mitarbeiter mit dem Nachnamen ”Miller” anzuwenden (vgl. Abbildung 2.24):

```
UPDATE Employees
     SET salary = salary * 1.1
     WHERE lastName = 'Miller';
```

Abbildung 2.24: SQL-Anweisung zur Aktualisierung des Gehalts von Mitarbeiter ”Miller”

Zuletzt kann auch mehr als ein Attribut einer Tabelle mit einer einzigen Anweisung aktualisiert werden, wie das folgende Beispiel zeigt (vgl. Abbildung 2.25):

```
UPDATE Employees SET
      jobID  = '4711',
      salary = salary + 1000,
      departmentID = 120;
```

Abbildung 2.25: SQL-Anweisung für die Aktualisierung von mehr als einem Attribut in der Tabelle `Employees`.

`DELETE` wird verwendet, um eine oder mehrere Datenzeilen aus einer Tabelle zu entfernen. Das folgende Beispiel ist die einfachste Form einer `DELETE`-Anweisung, die alle in der Tabelle mit den Mitarbeitern gespeicherten Daten löscht (vgl. Abbildung 2.26).

```
DELETE FROM Employees;
```

Abbildung 2.26: SQL-Anweisung zum Löschen aller Daten aus der Tabelle `Employees`.

Die Erweiterung der vorherigen Anweisung um eine "`WHERE`-Klausel" ermöglicht es, gezielt den Mitarbeiter mit der JobID "4711" zu löschen (vgl. Abbildung 2.27).

```
DELETE FROM Employees
      WHERE jobID  = '4711';
```

Abbildung 2.27: SQL Statement for Deleting some Data from Table `Employees`.

### 2.3.2 Datenmanipulation mit dem ORACLE Object Browser ("OB")

Neben dieser programmatischen Möglichkeit der Datenmanipulation wäre es auch möglich, Daten manuell in Tabellen einzufügen sowie bestehende Daten mit einem vom *Objektbrowser* bereitgestellten Editor zu ändern oder zu löschen, vgl. Abschnitt 2.2.4, Abbildung 2.21, unter Verwendung der "Daten"-Seite. Es ist zu beachten, dass alle Datenmanipulationen, die mit dem OB vorgenommen werden, von ORACLE automatisch in entsprechende SQL-DML-Anweisungen übersetzt werden.

### 2.3.3 Datenmanipulation mit Quick SQL

Mit Quick SQL ist es möglich, einige Daten automatisch in die Tabellen einzufügen. Ein Beispiel ist in Abbildung 2.28 mit den Anweisungen "`/insert 2`" und "`/insert 1`" dargestellt. Mit diesen Anweisungen wird die Anzahl der Tupel mit Zufallswerten definiert, die in die Tabellen eingefügt werden sollen.

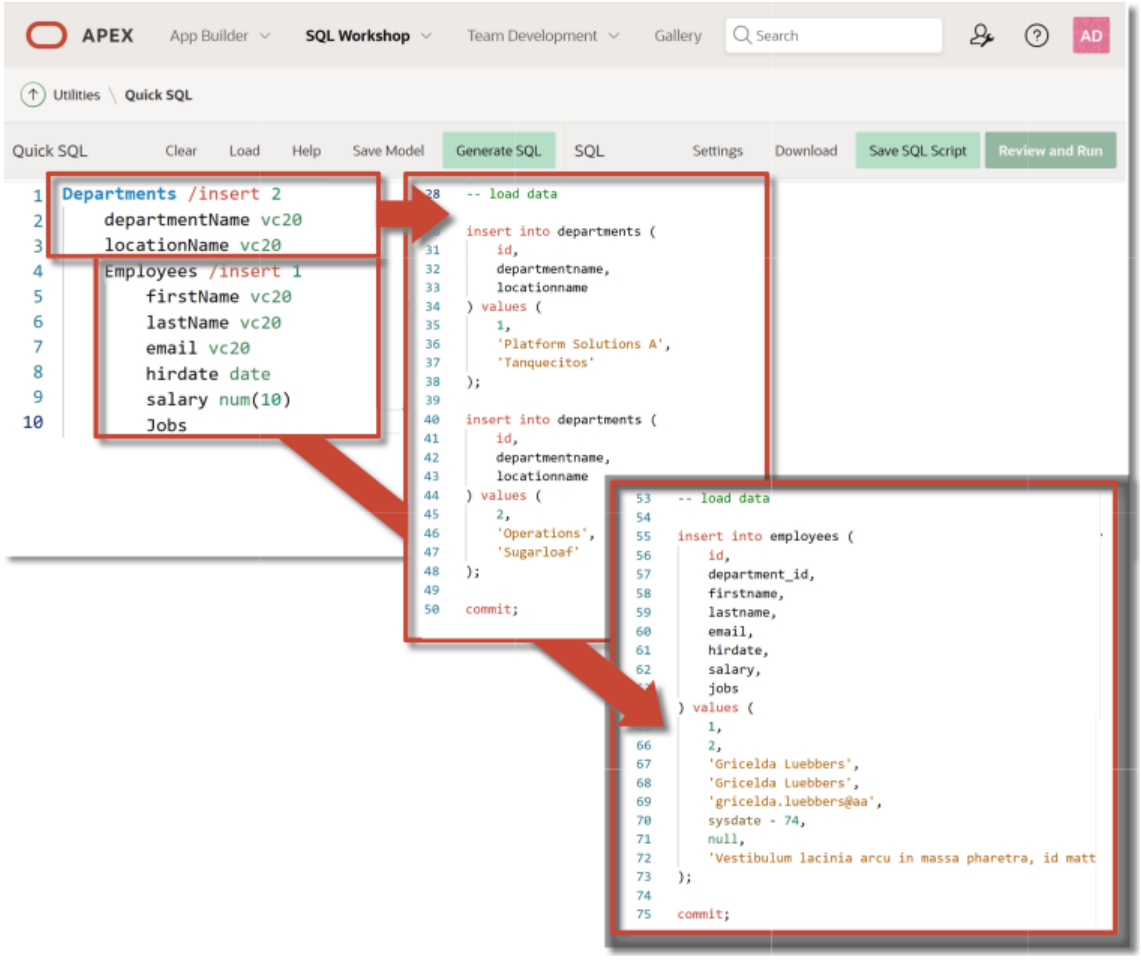



Abbildung 2.28: Zufällige Dateneinfügung mit Quick SQL

## 2.4 Wie man Daten abfragt - SQL-DQL

Nachdem wir in Abschnitt 2.2 verschiedene Möglichkeiten zur Erstellung von leeren Tabellen und in Abschnitt 2.3 die Möglichkeiten zur Datenmanipulation besprochen haben, wenden wir uns nun dem Abruf von Daten aus den Tabellen zu, indem wir zunächst die Abfragemöglichkeiten von SQL anhand der ”SELECT”-Anweisung besprechen (vgl. in Abschnitt 2.4.1), wiederum anhand unseres laufenden Beispiels, gefolgt von einem kurzen Überblick über das ORACLE-Tool Query Builder (”QB”), das eine ”Low-Code”-Abfragealternative für Benutzer mit geringen oder keinen SQL-Kenntnissen bietet (vgl. Abschnitt 2.4.2). Diese Alternativen sind auch in Abbildung 2.45 dargestellt.

### 2.4.1 Datenabfrage mit SQL-DQL

Zunächst ist anzumerken, dass die Ausdruckskraft von SQL-DQL immens ist, so dass wir uns nur auf eine recht kleine Teilmenge von Konzepten konzentrieren werden. Nichtsdestotrotz wollen wir einen ersten Eindruck von der Ausdruckskraft und der breiten Anwendbarkeit von SQL-DQL vermitteln, indem wir verschiedene Abfragemöglichkeiten exemplarisch vorstellen, im Folgenden geordnet nach steigender Komplexität. Als Hilfsmittel wird wieder der SQL-Befehlseditor verwendet (vgl. Abbildung 2.17), der auch für die Erstellung von Tabellen und die Manipulation von Daten verwendet wurde.

**(1) Beginnen wir mit der einfachsten Abfrage** - *”Alle Mitarbeiter abrufen”* (vgl. Abbildung 2.29).

```
SELECT * FROM Employees;
```

Abbildung 2.29: SQL-Anweisung zum Abrufen aller Daten aus der Tabelle `Employees`.

Die Zieltabelle der Abfrage (`Employees`) wird in der `FROM`-Klausel angegeben, das gewünschte Ergebnis der Abfrage wird direkt nach der `SELECT`-Klausel definiert - in diesem Beispiel unter Verwendung des Sterns als Jokerzeichen - was bedeutet, dass wir die Werte aller Spalten der Tabelle erhalten möchten.

Ein Auszug aus dem Ergebnis dieser Abfrage sieht wie folgt aus (vgl. Abbildung 2.30):

| EMPLOYEEID | FIRSTNAME | LASTNAME | EMAIL | HIREDATE | SALARY | DEPARTMENTS_DEPARTMENTID |
|---|---|---|---|---|---|---|
| 100 | Steven | King | SKING | 06/17/2003 | 24000 | 90 |
| 101 | Neena | Kochhar | NKOCHHAR | 09/21/2005 | 17000 | 90 |
| 102 | Lex | De Haan | LDEHAAN | 01/13/2001 | 17000 | 90 |
| 103 | Alexander | Hunold | AHUNOLD | 01/03/2006 | 9000 | 60 |
| 104 | Bruce | Ernst | BERNST | 05/21/2007 | 6000 | 60 |
| 105 | David | Austin | DAUSTIN | 06/25/2005 | 4800 | 60 |
| 106 | Valli | Pataballa | VPATABAL | 02/05/2006 | 4800 | 60 |
| 107 | Diana | Lorentz | DLORENTZ | 02/07/2007 | 4200 | 60 |
| 108 | Nancy | Greenberg | NGREENBE | 08/17/2002 | 12008 | 100 |
| 109 | Daniel | Faviet | DFAVIET | 08/16/2002 | 9000 | 100 |
| 110 | John | Chen | JCHEN | 09/28/2005 | 8200 | 100 |
| 111 | Ismael | Sciarra | ISCIARRA | 09/30/2005 | 7700 | 100 |
| 112 | Jose Manuel | Urman | JMURMAN | 03/07/2006 | 7800 | 100 |
| 113 | Luis | Popp | LPOPP | 12/07/2007 | 6900 | 100 |
| 114 | Den | Raphaely | DRAPHEAL | 12/07/2002 | 11000 | 30 |

Abbildung 2.30: Auszug aus dem Abrufen aller Daten aus der Tabelle `EMPLOYEES`.

Wenn wir das Ergebnis z.B. nach den Gehältern der Mitarbeiter ordnen möchten, müssen wir einfach eine `ORDER BY`-Klausel hinzufügen (vgl. Abbildung 2.31):

```
SELECT * FROM Employees ORDER BY salary;
```

Abbildung 2.31: SQL-Anweisung zum Abrufen aller Daten aus der Tabelle `Employees` geordnet nach ihrem Gehalt.

Das Ergebnis dieser Abfrage sieht wie folgt aus (vgl. Abbildung 2.32):

| EMPLOYEEID | FIRSTNAME | LASTNAME | EMAIL | HIREDATE | SALARY | DEPARTMENTS_ DEPARTMENTID |
|---|---|---|---|---|---|---|
| 100 | Steven | King | SKING | 06/17/2003 | 24000 | 90 |
| 101 | Neena | Kochhar | NKOCHHAR | 09/21/2005 | 17000 | 90 |
| 102 | Lex | De Haan | LDEHAAN | 01/13/2001 | 17000 | 90 |
| 108 | Nancy | Greenberg | NGREENBE | 08/17/2002 | 12008 | 100 |
| 114 | Den | Raphaely | DRAPHEAL | 12/07/2002 | 11000 | 30 |
| 103 | Alexander | Hunold | AHUNOLD | 01/03/2006 | 9000 | 60 |
| 109 | Daniel | Faviet | DFAVIET | 08/16/2002 | 9000 | 100 |
| 110 | John | Chen | JCHEN | 09/28/2005 | 8200 | 100 |
| 112 | Jose Manuel | Urman | JMURMAN | 03/07/2006 | 7800 | 100 |
| 111 | Ismael | Sciarra | ISCIARRA | 09/30/2005 | 7700 | 100 |
| 113 | Luis | Popp | LPOPP | 12/07/2007 | 6900 | 100 |
| 104 | Bruce | Ernst | BERNST | 05/21/2007 | 6000 | 60 |
| 105 | David | Austin | DAUSTIN | 06/25/2005 | 4800 | 60 |
| 106 | Valli | Pataballa | VPATABAL | 02/05/2006 | 4800 | 60 |
| 107 | Diana | Lorentz | DLORENTZ | 02/07/2007 | 4200 | 60 |

Abbildung 2.32: Auszug aus dem Abrufen aller Daten aus der Tabelle `Employees` geordnet nach ihrem Gehalt

**(2) Wie steht es um die Abfrage bestimmter Spalten und Zeilen -** *"Rufe nur Name, Adresse und Gehalt von Mitarbeitern mit einem Gehalt von mehr als 5.000 ab"*

Erstens müssen wir, um nur die Werte bestimmter Attribute abzurufen, diese anstelle des Sterns, durch Kommata getrennt, angeben. Zweitens müssen wir, um selektiv nur bestimmte Zeilen abzurufen, eine `WHERE`-Klausel einfügen und eine logische Bedingung für das entsprechende Attribut (in unserem Fall Gehalt) angeben. Beachten Sie, dass es mehrere solcher Bedingungen für verschiedene Attribute geben kann, die durch die Schlüsselwörter `AND` / `OR` getrennt werden (vgl. Abbildung 2.33).

```
SELECT lastName, hireDate, salary FROM Employees
WHERE salary > 5000;
```

Abbildung 2.33: SQL-Anweisung zum Abrufen bestimmter Daten aus der Tabelle `Employees`.

Das Ergebnis dieser Abfrage sieht wie folgt aus (vgl. Abbildung 2.34):

| LASTNAME | HIREDATE | SALARY |
|---|---|---|
| King | 06/17/2003 | 24000 |
| Kochhar | 09/21/2005 | 17000 |
| De Haan | 01/13/2001 | 17000 |
| Hunold | 01/03/2006 | 9000 |
| Ernst | 05/21/2007 | 6000 |
| Greenberg | 08/17/2002 | 12008 |
| Faviet | 08/16/2002 | 9000 |
| Chen | 09/28/2005 | 8200 |
| Sciarra | 09/30/2005 | 7700 |
| Urman | 03/07/2006 | 7800 |
| Popp | 12/07/2007 | 6900 |
| Raphaely | 12/07/2002 | 11000 |

Abbildung 2.34: Auszug aus dem Abrufen bestimmter Daten aus der Tabelle `Employees`.

Wenn wir die Werte des Attributs "`hireDate`" in einem anderen Format erhalten möchten, kann die von ORACLE SQL bereitgestellte "`TO_CHAR()`-Funktion" verwendet werden. Diese

Funktion nimmt einen DATE-Wert als erste Eingabe und ein gewünschtes Format als zweite Eingabe. Im Folgenden wird eine erweiterte Version des obigen Beispiels gezeigt (vgl. Abbildung 2.35), wobei nun die TO_CHAR()-Funktion für "hiredate" verwendet wird (eine vollständige Liste der möglichen Formatierungsoptionen ist natürlich in der Oracle-Dokumentation verfügbar):

```
SELECT lastName, TO_CHAR(hireDate, 'Month, DD, YYYY') hireDate, salary
FROM Employees
WHERE salary > 1000;
```

Abbildung 2.35: SQL-Anweisung zum Abrufen bestimmter Daten aus der Tabelle Employees unter Verwendung der TO_CHAR()-Funktion.

Das Ergebnis dieser Abfrage sieht wie folgt aus (vgl. Abbildung 2.36):

| LASTNAME | HIREDATE | SALARY |
|---|---|---|
| King | June, 17, 2003 | 24000 |
| Kochhar | September, 21, 2005 | 17000 |
| De Haan | January, 13, 2001 | 17000 |
| Hunold | January, 03, 2006 | 9000 |
| Ernst | May, 21, 2007 | 6000 |
| Austin | June, 25, 2005 | 4800 |
| Pataballa | February, 05, 2006 | 4800 |
| Lorentz | February, 07, 2007 | 4200 |
| Greenberg | August, 17, 2002 | 12008 |
| Faviet | August, 16, 2002 | 9000 |
| Chen | September, 28, 2005 | 8200 |
| Sciarra | September, 30, 2005 | 7700 |
| Urman | March, 07, 2006 | 7800 |
| Popp | December, 07, 2007 | 6900 |
| Raphaely | December, 07, 2002 | 11000 |

Abbildung 2.36: Auszug aus dem Abrufen bestimmter Daten aus der Tabelle Mitarbeiter nur mit der TO_CHAR()-Funktion.

**(3) Konzentrieren wir uns auf die Abfrage mehrerer Tabellen** - "*Abrufen aller Mitarbeiter und ihrer Abteilungen*" Der Abruf von Daten aus zwei oder mehr Tabellen innerhalb einer einzigen SELECT-Anweisung erfordert ein recht bekanntes SQL-Konzept, nämlich einen "*Join*". Ein Join kombiniert nämlich die Zeilen aus zwei oder mehr Tabellen, die in der FROM-Klausel angegeben werden müssen. Zusätzlich benötigen wir eine WHERE-Klausel, die die Rolle eines *Vermittlers* zwischen diesen beiden Tabellen spielt - da wir irgendwie definieren müssen, dass das DBS für jeden Mitarbeiter in der Tabelle Employees seinen Departments_ departmentID-Wert (der der FK ist) nimmt, weiter zur Tabelle Departments geht, diesen Wert mit dem entsprechenden PK-Wert von departmentID abgleicht und die Abteilungsdaten zusammen mit den Mitarbeiterdaten zurückgibt. Formal wird dieses Matchmaking durch eine so genannte "*join-condition*" spezifiziert, wie im folgenden Beispiel zu sehen ist (vgl. Abbildung 2.37):

```
SELECT lastName, salary, departmentName
FROM Employees, Departments
WHERE Employees.Departments_departmentID = Departments.departmentID;
```

Abbildung 2.37: SQL-Anweisung zur Verknüpfung von Daten aus Tabelle Employees und Tabelle Departments.

Das Ergebnis dieser Abfrage sieht wie folgt aus (vgl. Abbildung 2.38):

| LASTNAME | SALARY | DEPARTMENTNAME |
|---|---|---|
| King | 24000 | Executive |
| Kochhar | 17000 | Executive |
| De Haan | 17000 | Executive |
| Hunold | 9000 | IT |
| Ernst | 6000 | IT |
| Austin | 4800 | IT |
| Pataballa | 4800 | IT |
| Lorentz | 4200 | IT |
| Greenberg | 12008 | Finance |
| Faviet | 9000 | Finance |
| Chen | 8200 | Finance |
| Sciarra | 7700 | Finance |
| Urman | 7800 | Finance |
| Popp | 6900 | Finance |
| Raphaely | 11000 | Purchasing |

Abbildung 2.38: Auszug aus dem Abrufen bestimmter Daten aus Tabelle `Employees` und Tabelle `Departments`.

Beachten Sie, dass die entsprechenden Join-Attribute ebenfalls mit den entsprechenden Tabellennamen (durch einen "Punkt" getrennt) versehen sind, um eine bessere Lesbarkeit zu erreichen und - falls der Name des FK mit dem Namen des PK identisch ist - eine eindeutige Identifizierung zu gewährleisten.

**(4) Bleiben wir bei der Aggregation von Werten** - "*Abrufen der Anzahl von Mitarbeitern und der Summe ihrer Gehälter*"

Um die Anzahl aller Mitarbeiter in der Tabelle `Employees` zu ermitteln, können wir die SQL `COUNT`-Funktion über verschiedene `employeeIDs` verwenden. Zur Bildung einer Summe aller Werte können wir die `SUM`-Funktion über ein Attribut verwenden, in unserem Fall, da wir die Summe aller Gehälter berechnen wollen, das Attribut `salary` (vgl. Abbildung 2.39):

```
SELECT COUNT(employeeID), SUM(salary)
FROM Employees;
```

Abbildung 2.39: SQL-Anweisung zum Zählen der Anzahl der Mitarbeiter und zum Bilden der Summe ihrer Gehälter aus der Tabelle `Employees`.

Das Ergebnis dieser Abfrage sieht wie folgt aus (vgl. Abbildung 2.40):

| COUNT(EMPLOYEEID) | SUM(SALARY) |
|---|---|
| 15 | 149408 |

Abbildung 2.40: Auszug aus dem Abrufen der Anzahl der Mitarbeiter und Bildung der Summe ihrer Gehälter aus der Tabelle `Employees`.

**(5) Wie sieht es schließlich mit der Gruppierung bestimmter Zeilen aus** - "*Abrufen der Anzahl der Mitarbeiter, gruppiert nach ihren Gehältern in absteigender Reihenfolge*"

Um gleiche Daten zusammenzufassen, können wir die `GROUP BY`-Funktion in einer `SELECT`-Anweisung verwenden. Kombiniert mit einer SQL `COUNT`-Funktion über verschiedene `employeeIDs` können wir nun die Anzahl der Zeilen für jede Gruppe ermitteln. Zum Sortieren des Ergebnisses

können wir die `ORDER BY`-Funktion verwenden, die angibt, welches Attribut für die Sortierung verwendet werden soll. Dabei gibt `ASC` eine aufsteigende Reihenfolge an, während `DESC` eine absteigende Reihenfolge festlegt (vgl. Abbildung 2.41):

```
SELECT COUNT(employeeID), SUM(salary)
FROM Employees
GROUP BY salary
ORDER BY COUNT(employeeID) DESC;
```

Abbildung 2.41: SQL-Anweisung zum Gruppieren der Daten nach ihrem Gehalt und Ordnen des Ergebnisses nach der Anzahl der Mitarbeiter mit diesem Gehalt aus der Tabelle `Employees`.

Das Ergebnis dieser Abfrage sieht wie folgt aus (vgl. Abbildung 2.42):

| COUNT(EMPLOYEEID) | SUM(SALARY) |
|---|---|
| 2 | 9600 |
| 2 | 34000 |
| 2 | 18000 |
| 1 | 6900 |
| 1 | 7700 |
| 1 | 12008 |
| 1 | 11000 |
| 1 | 24000 |
| 1 | 6000 |
| 1 | 7800 |
| 1 | 4200 |
| 1 | 8200 |

Abbildung 2.42: Auszug aus dem Gruppieren der Daten nach ihrem Gehalt und Ordnen des Ergebnisses nach der Anzahl der Mitarbeiter mit diesem Gehalt aus der Tabelle `Employees`

### 2.4.2 Datenabfrage mit dem ORACLE Query Builder (”QB”)

Wie bereits erwähnt, ermöglichen die ORACLE Utilities in Form des sogenannten Query Builders (”QB”) auch Benutzern mit geringen oder gar keinen SQL-Kenntnissen, DB-Tabellen abzufragen und die Abfragen zur weiteren Verwendung zu speichern. Der QB kann auf folgende Weise aufgerufen werden (vgl. Abbildung 2.43):

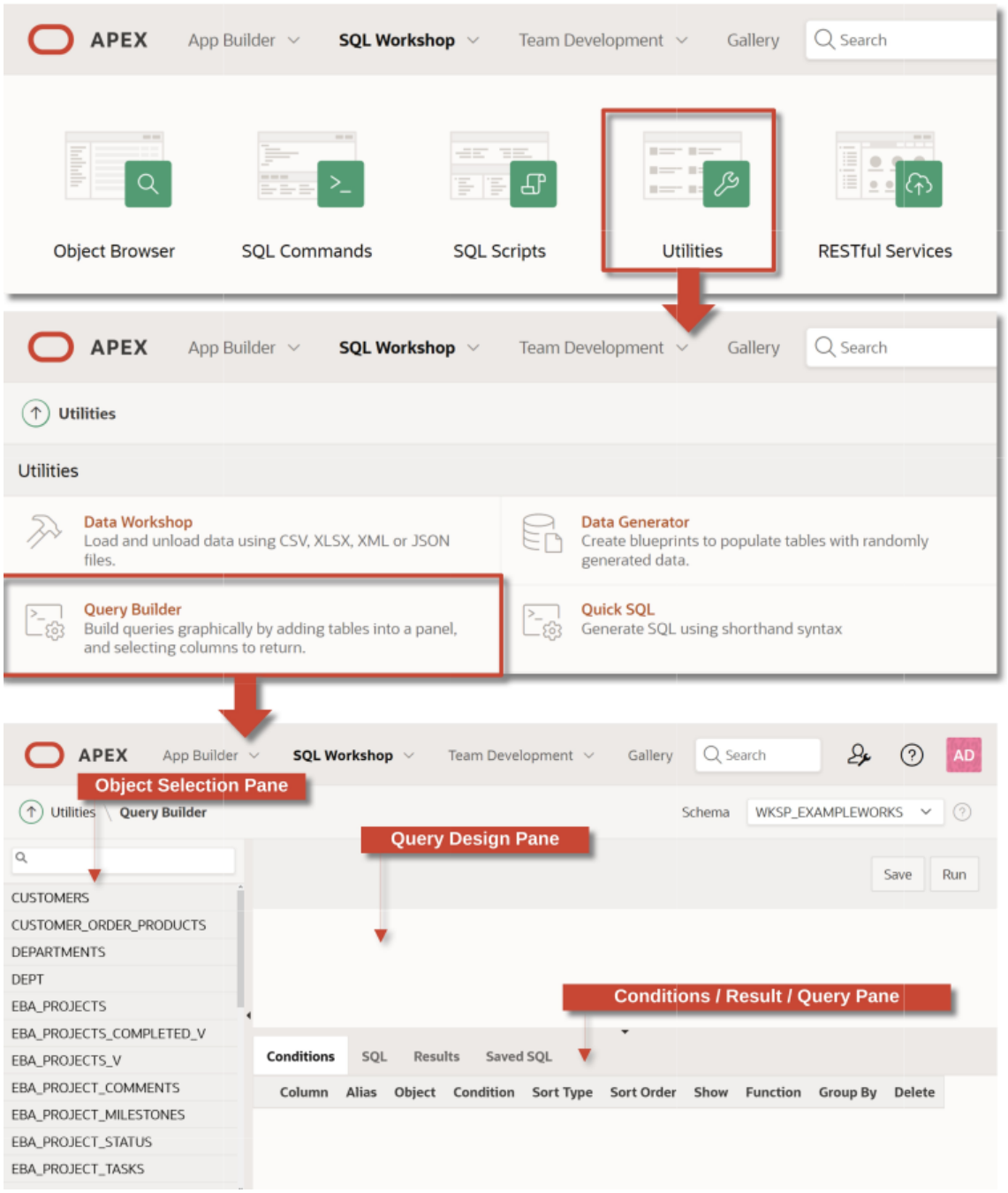



Abbildung 2.43: Überblick über den ORACLE Query Builder (”QB”).

Insgesamt sind die Schritte, die notwendig sind, um Abfragen zu erstellen, in Abbildung 2.44 beispielhaft dargestellt, anhand unseres in Abbildung 2.37 gezeigten ”Join-Beispiels” und im Folgenden kurz beschrieben:

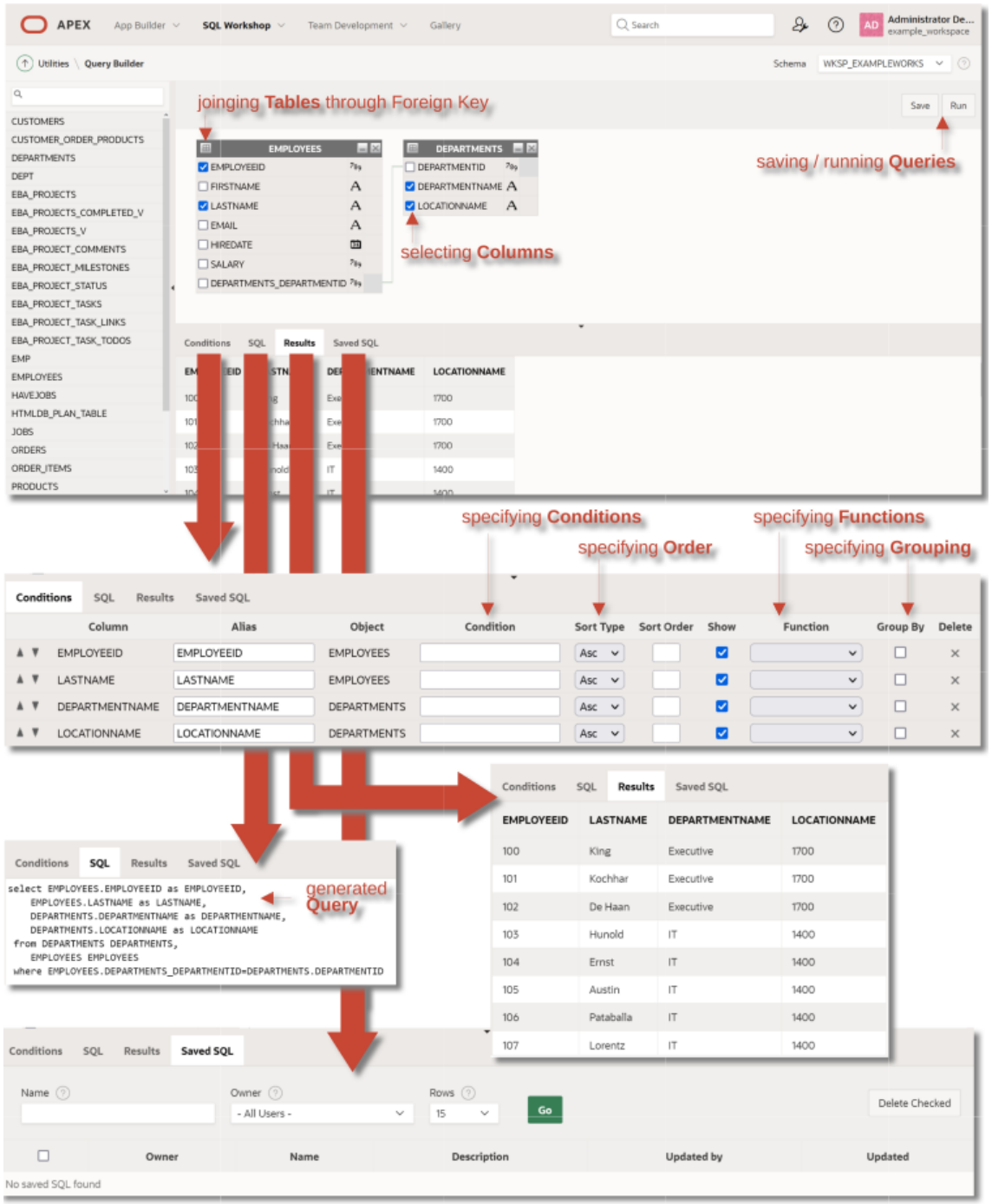



Abbildung 2.44: Verwendung des Abfragegenerators zur Erstellung eines Join

- Wählen Sie das *Ziel der Abfrage* in Form einer oder mehrerer DB-Tabellen (im QB "Objekte" genannt) aus dem "Objektauswahlfenster" (vgl. Abbildung 2.43). Dies bildet den "`FROM`"-Teil von SQL-DQL.
- Fügen Sie die ausgewählten Objekte zum "Entwurfsfenster" hinzu und wählen Sie die gewünschten Spalten weiter aus. Dies bildet den "`SELECT`"-Teil von SQL-DQL.
- Optional: Erstellen Sie Abfragebedingungen, d.h. den "`WHERE`"-Teil von SQL-DQL.
- Führen Sie die Abfrage aus und zeigen Sie die Ergebnisse an (klicken Sie auf die Schaltfläche "Ausführen").
- Optional: Sehen Sie sich den SQL-DQL-Code an, der automatisch generiert wird (klicken

Sie auf die Registerkarte ”SQL”).

- Optional: Speichern Sie die Abfrage zur weiteren Verwendung (klicken Sie auf den ”Speichern-Button”).

Schließlich ist zu beachten, dass die mit QB definierten Abfragen von dem RDBS automatisch in richtige SQL-DQL-Anweisungen übersetzt werden.

## 2.5 Aufbau der DB-Schicht - Das große Bild

Nach der Lektüre der vorangegangenen Abschnitte ist man vielleicht völlig verwirrt von all den verschiedenen Optionen und Alternativen, welche die ORACLE-Tooling-Landschaft bietet, um die DB-Schicht von Webanwendungen zu verwalten. Um sich einen Weg durch diesen Tooling-Dschungel zu bahnen, gibt Abbildung 2.45 einen zusammenfassenden Überblick über alle Optionen und die entsprechenden Tools, die zur Verfügung stehen, um die DB-Schicht einer Webanwendung zu verwalten.

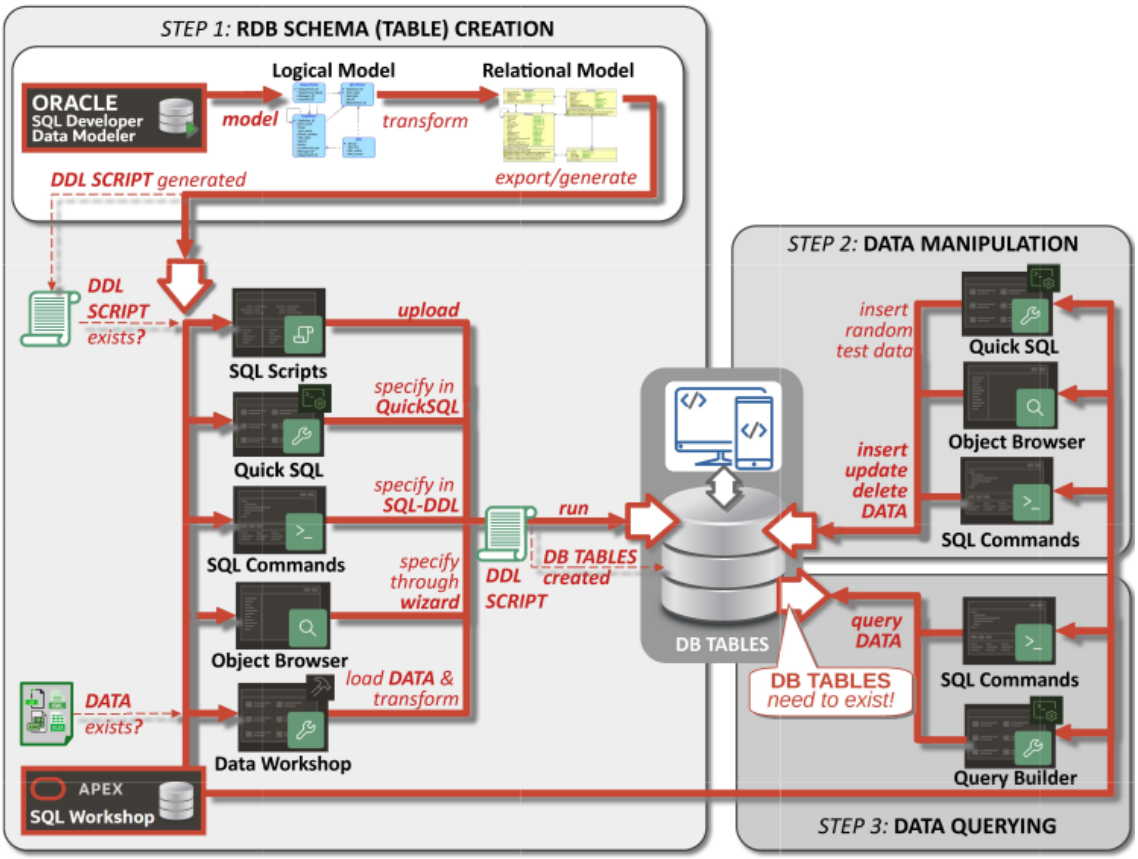



Abbildung 2.45: Übersicht über die Optionen zur Verwaltung der DB-Schicht.

Insgesamt werden auf der linken Seite zwei verschiedene Tooling-Umgebungen gezeigt, der ODM und der APEX SQL Workshop. Das letztendliche Ziel, die Erstellung des RDB-Schemas in Form von Tabellen (Schritt 1), kann auf verschiedenen Wegen erreicht werden. Alle führen jedoch zur Erstellung von SQL-Skripten, die schließlich ausgeführt werden, um leere Tabellen zu erzeugen. Von dort aus ist es möglich, Daten in die Tabellen einzufügen, zu aktualisieren, zu löschen (Schritt 2) oder Daten abzufragen (Schritt 3), indem man die Werkzeuge der SQL Workshop Workbench verwendet, wie in diesem Kapitel beschrieben.

Die Webanwendung könnte auf leeren Tabellen erstellt werden. Dieser Prozess wird jedoch im folgenden Kapitel 3 ausführlich behandelt.



# 3. Wie navigiert man in APEX?

E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER

**Was ist Oracle APEX?** Oracle Application Express (Oracle APEX) ist eine Webbrowser-basierte Low-Code-Entwicklungsumgebung für Oracle DB-gesteuerte Webanwendungen. „Low-Code“bedeutet, dass für den Aufbau einer vollwertigen Webanwendung nur wenige Programmierkenntnisse erforderlich sind - für einen ersten, einfachen Webanwendungsprototyp sind sogar überhaupt keine Programmierkenntnisse notwendig.

**Für welche Zwecke kann das APEX eingesetzt werden?** Insgesamt ermöglicht APEX nicht nur den Aufbau der DB-Schicht von Webanwendungen mit APEX „*SQL Workshop*“, wie in Kapitel 2 beschrieben, sondern auch den Aufbau von Desktop- oder mobilen Webanwendungen mit APEX „*App Builder*“. APEX wird sowohl von großen als auch von kleinen Kunden in einer Vielzahl von Anwendungsdomänen eingesetzt, um ein breites Spektrum von Geschäftsanforderungen zu erfüllen. Diese Geschäftsanforderungen können von der einfachen Umwandlung einer lokalen Tabellenkalkulation in eine webbasierte Anwendung bis hin zur Realisierung einer vollwertigen Webanwendung reichen (vgl. das in Kapitel 2 vorgestellte laufende Beispiel), die es ermöglicht, jeden einzelnen Verkauf in jeder der 10.500 Walmart-Filialen zusammen mit den zuständigen Abteilungen, Mitarbeitern und Informationen über ihre Arbeit zu speichern, abzurufen, zu visualisieren und weiter zu verarbeiten.

**Was ist der Grund für dieses Kapitel?** Der Grundgedanke dieses Kapitels ist es, zunächst den Gesamtprozess der Webanwendungsentwicklung zu erörtern, der von Oracle APEXs „App Builder“verfolgt wird. Dieser Prozess bildet die Grundlage für die Navigationsmöglichkeiten zwischen den verschiedenen App Builder-Werkzeugen, die als „Entwicklungskomponenten“bezeichnet werden, um eine Webanwendung zu erstellen. Zweitens wird ein eher grober Überblick über diese verschiedenen Entwicklungskomponenten gegeben, während die Details in den weiteren Kapiteln behandelt werden.

## 3.1 Webanwendungsentwicklungsprozess gefolgt von APEX

Bevor wir auf die verschiedenen Entwicklungskomponenten des APEX „App Builder“eingehen, ist es sinnvoll, einen Blick auf den Entwicklungsprozess von Webanwendungen zu werfen, den APEX verfolgt.

**APEX folgt einem mehrstufigen und zyklischen Entwicklungsprozess.** Der gesamte von APEX unterstützte Entwicklungsprozess, der in Abbildung 3.1 dargestellt ist, ist in einer mehrstufigen, teilweise zyklischen Weise organisiert, die eine schrittweise Verfeinerung der Webanwendung

und schließlich der DB-Schicht ermöglicht. Jeder der Entwicklungsschritte wird durch entsprechende Assistenten der Entwicklungskomponenten unterstützt. Das bedeutet, dass man von den Menüoptionen auf der obersten Ebene tiefer in die Funktionalität vom *App Builder* geführt wird, um jedes Detail einer Webanwendung zu spezifizieren und bei Bedarf schrittweise Verfeinerungen vorzunehmen.

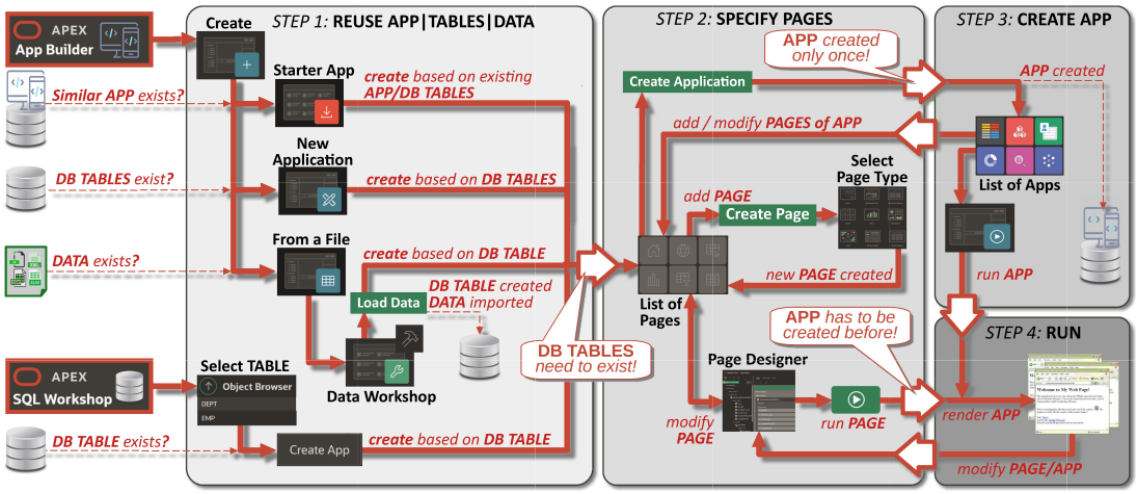



Abbildung 3.1: Der gesamte Entwicklungsprozess in APEX.

**APEX folgt einem „DB-Layer First“-Entwicklungsprozess.** Da sich APEX auf die Entwicklung von DB-gesteuerten Webanwendungen konzentriert, ist es notwendig, immer zuerst die DB-Schicht zu entwickeln (vgl. Abbildung 3.1 „SCHRITT 1: APP | TABLES | DATA WIEDERVERWENDEN“). Dies kann entweder wie in Kapitel 2 beschrieben oder direkt im App Builder erfolgen, wenn z.B. bereits externe Datendateien existieren, die für die Erstellung der DB-Tabellen wiederverwendet werden können (vgl. Abbildung 3.1 „From a File“). Falls die DB-Tabellen bereits existieren, kann der Entwicklungsprozess entweder (i) aus dem App Builder heraus gestartet werden, wobei eventuell eine bestehende Anwendung zusammen mit ihren DB-Tabellen wiederverwendet (und angepasst) wird (vgl. Abbildung 3.1 „Starter App“), oder es wird eine neue Anwendung von Grund auf neu erstellt, wiederum basierend auf bestehenden DB-Tabellen (vgl. Abbildung 3.1 „New Application“) oder (ii) aus SQL Workshop heraus, indem Sie einfach eine DB-Tabelle im Objektbrowser auswählen (vgl. Abbildung 3.1 „Select TABLE“) und die Anwendungsentwicklung auf der Grundlage dieser Tabelle starten.

**APEX folgt einem „seitengesteuerten“Entwicklungsprozess.** Sobald die DB-Tabellen für die Speicherung der Daten der Web-Anwendung vorhanden sind, können sie für die Erstellung einer neuen Web-Anwendung verwendet werden, wobei ein *seitengesteuerter Prozess* (vgl. Abbildung 3.1 „SCHRITT 2: SEITEN FESTLEGEN“) verfolgt wird. Das bedeutet, dass eine oder mehrere Webseiten definiert werden können, die die HTML-Seiten der endgültigen Webanwendung bilden. Jede dieser Seiten kann auf einer oder mehreren DB-Tabellen basieren und ermöglicht die Visualisierung ihrer Daten in Form verschiedener Seitentypen, die z.B. interaktive Berichte, Formulare, Listen, Diagramme oder Kalender umfassen, aber auch das interaktive „Spielen“mit den Daten (z.B. in Details zoomen) und sogar deren Manipulation und das Speichern der Änderungen zurück in die DB. Diese Seiten können mit Hilfe von Navigationsmenüs, Registerkarten, Schaltflächen oder Hypertext-Links miteinander verbunden werden.

Sobald eine oder mehrere Seiten erstellt wurden, ist es notwendig, die Gesamtanwendung zu erstellen (vgl. Abbildung 3.1 „SCHRITT 3: APP ERSTELLEN“), wobei einige allgemeine Eigenschaften der gesamten Webanwendung (z. B. das Aussehen der App) weiter spezifiziert oder einfach die Standardeinstellungen verwendet werden können. Zuletzt muss die App bzw. jede der

angegebenen Seiten gerendert werden, um die endgültigen HTML-Seiten zu generieren (Abbildung 3.1 „SCHRITT 4: STARTEN“), die weiterhin interaktiv getestet werden können, wobei gelegentlich ein Rückschritt zur weiteren Verfeinerung der Webanwendung erfolgt.

## 3.2 Das Startmenü von APEX

Wenn Sie sich bei Oracle APEX anmelden, erscheint die sogenannte *„Workspace Homepage“*. Diese Arbeitsbereichshomepage bietet Zugang zu vier verschiedenen Entwicklungskomponenten (vgl. Abbildung 3.2):

- *„App Builder“* wird hauptsächlich für die Erstellung von Webseiten verwendet und ist der Schwerpunkt dieses Kapitels.
- *„SQL Workshop“* wird für den Aufbau der DB-Schicht verwendet und wurde bereits in Kapitel 2 beschrieben.
- *„Team Development“* wird z.B. verwendet, um mit anderen Entwicklern über ein Ticket-System zu interagieren, um Meilensteine zu setzen usw.
- *„Gallery“* wird verwendet, um vorgefertigte Demo-Anwendungen zu installieren, die dann weiter modifiziert und an die eigenen Bedürfnisse angepasst werden können, wiederum mit dem „*App Builder*“.

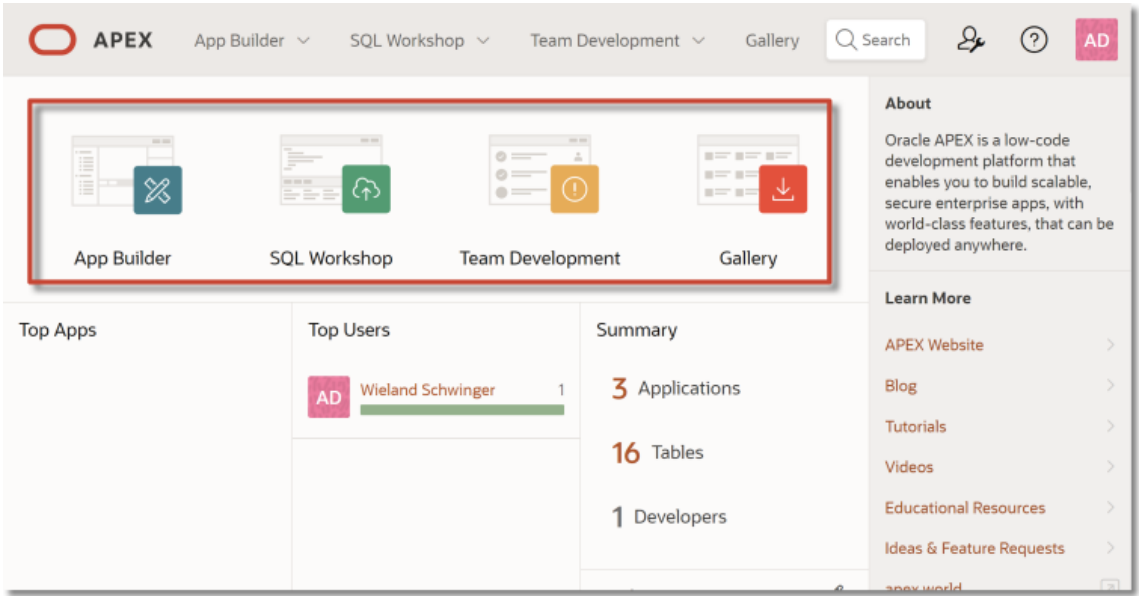



Abbildung 3.2: Oracle APEX Workspace Homepage.

Am unteren Rand der Arbeitsbereichshomepage (vgl. Abbildung 3.2) bieten die Bereiche „*Top Apps*“, „*Top Users*“und „*Summary*“Echtzeitinformationen über die Entwicklungsaktivitäten im aktuellen DB-Arbeitsbereich, während auf der rechten Seite auf weitere APEX-Lernressourcen verwiesen wird.

## 3.3 Übersicht über den App Builder - Erstellen und Verwalten Ihrer Apps

Wenn man zur App Builder Homepage navigiert, hat man folgende Möglichkeiten (vgl. Abbildung 3.3):

- *Erstellen* Sie eine neue Webanwendung.
- *Importieren* zuvor exportierter Webanwendungen.
- *Ansicht des „Dashboards“* mit Statistiken über bereits entwickelte Webanwendungen.

- *Zugang zu „Workspace Utilities“*, die eine breite Palette von Diensten wie Fernzugriff auf Daten oder Sicherungsfunktionen bieten.

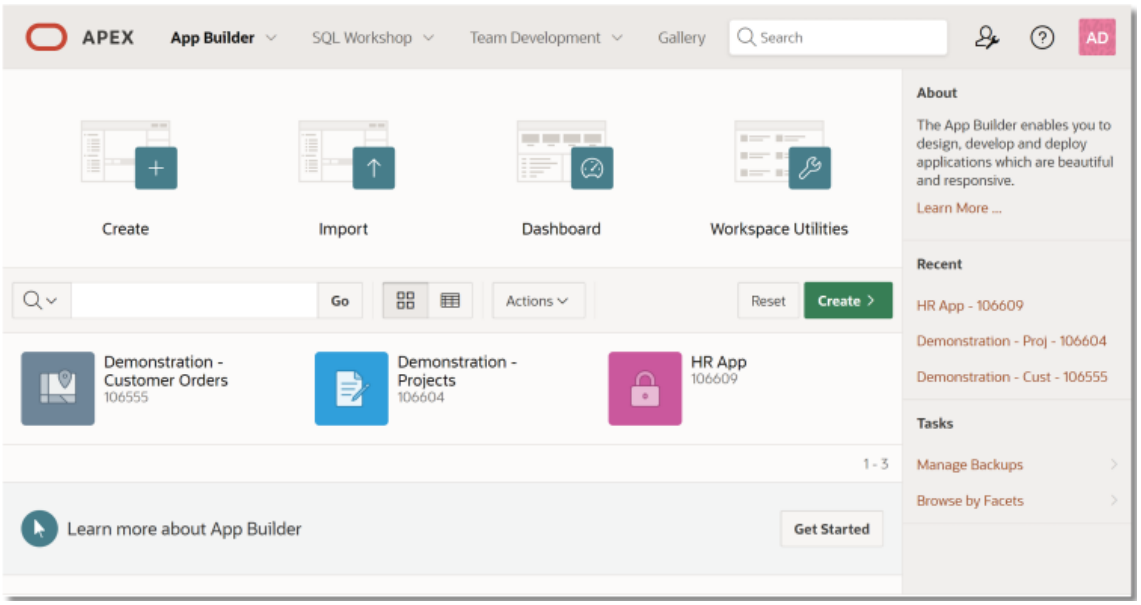



Abbildung 3.3: Überblick über den App Builder.

In der unteren Hälfte des Fensters (vgl. Abbildung 3.3) werden bereits vorhandene Webanwendungen aufgelistet, die zur weiteren Bearbeitung oder zum Rendering ausgewählt werden können (vgl. die folgenden Abschnitte).

## 3.4 Anwendung erstellen - Drei Anwendungsfälle

Bei der Auswahl von *„Create“* im App Builder zeigt der *„Create Application Wizard“* drei weitere Optionen an (vgl. Abbildung 3.4), die in der Tat reale Anwendungsfälle unterstützen, die das mögliche Wiederverwendungspotenzial hinsichtlich der Daten und der Anwendung selbst berücksichtigen:

- *„New Application“* sollte gewählt werden, wenn die DB-Schicht (oder ein Teil davon) bereits existiert oder wenn die DB-Schicht während der Anwendungsentwicklung „on-the-fly“entwickelt werden soll.
- *„From a File“* ist die Option der Wahl, wenn die DB-Schicht automatisch aus einer externen Datei, z.B. einer EXCEL-Datei, erzeugt werden soll.
- *„Starter App“* schließlich ermöglicht die Wiederverwendung (und möglicherweise weitere Modifizierung) einer bereits bestehenden Anwendung zusammen mit ihrer DB-Schicht aus der bereits oben erwähnten „Galerie“von Anwendungen.

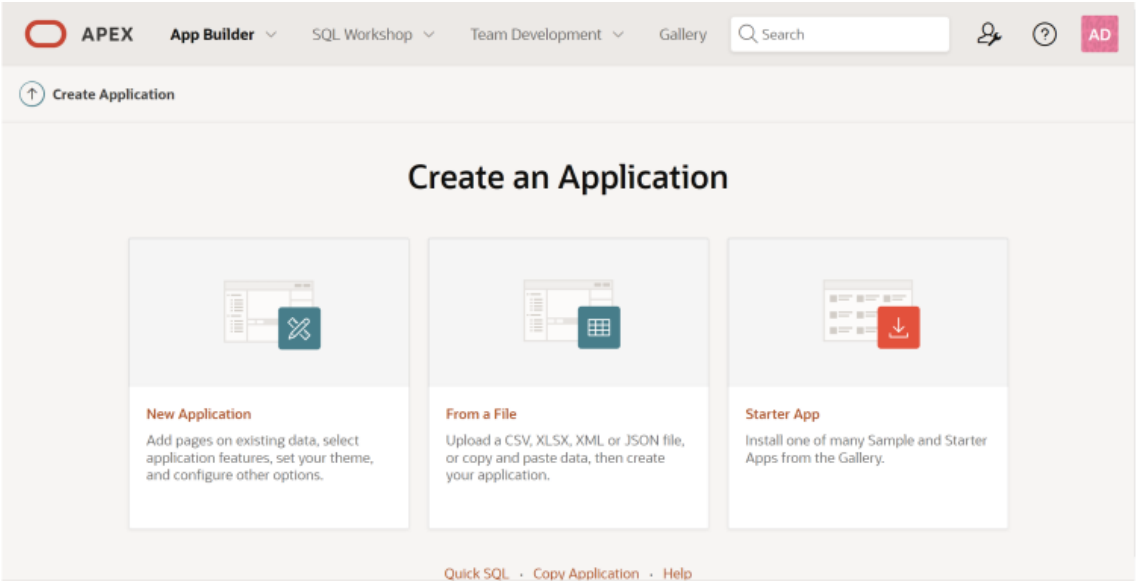



Abbildung 3.4: Create Application Wizard: Drei Anwendungsfälle beim Erstellen einer Anwendung.

Im Folgenden wird der Schwerpunkt auf die erste Option *„New Application“* gelegt, die zweite Option *„From a File“* wird in Kapitel 4 ausführlicher beschrieben, auf die dritte Option *„Starter App“* wird nicht weiter eingegangen.

## 3.5 Anwendung erstellen - Eigenschaften, Seiten, Funktionen und Einstellungen

Bei der Erstellung einer neuen Webanwendung müssen einige *gemeinsame Eigenschaften* definiert werden, die *Seiten* müssen erstellt werden, und einige zusätzliche recht nützliche *Funktionen* und *Einstellungen* können ausgewählt werden. Im Folgenden werden alle diese Definitionen näher erläutert, wobei Abbildung 3.5 den oberen Teil der GUI und Abbildung 3.6 den unteren Teil zeigt.

**Gemeinsame Eigenschaften - Icon, Name und Erscheinungsbild der Anwendung.** Bei den gemeinsamen Eigenschaften, wie in Abbildung 3.5 dargestellt, kann (1) ein beliebiges *Icon* ausgewählt oder hochgeladen werden, das als Favicon und Icon für PWA, Apple Touch oder App Builder verwendet wird, (2) ein *Name* muss eingegeben werden und (3) das *Erscheinungsbild* der Webanwendung kann aus einigen vordefinierten Optionen ausgewählt werden, die ihr „Look and Feel“in Bezug auf einen Themenstil (z.B, Farbe, Schriftgröße usw.) und die Art des Navigationsmenüs (z.B. Seitenmenü oder oberes Menü), wobei auch ein Standardaussehen verfügbar ist.

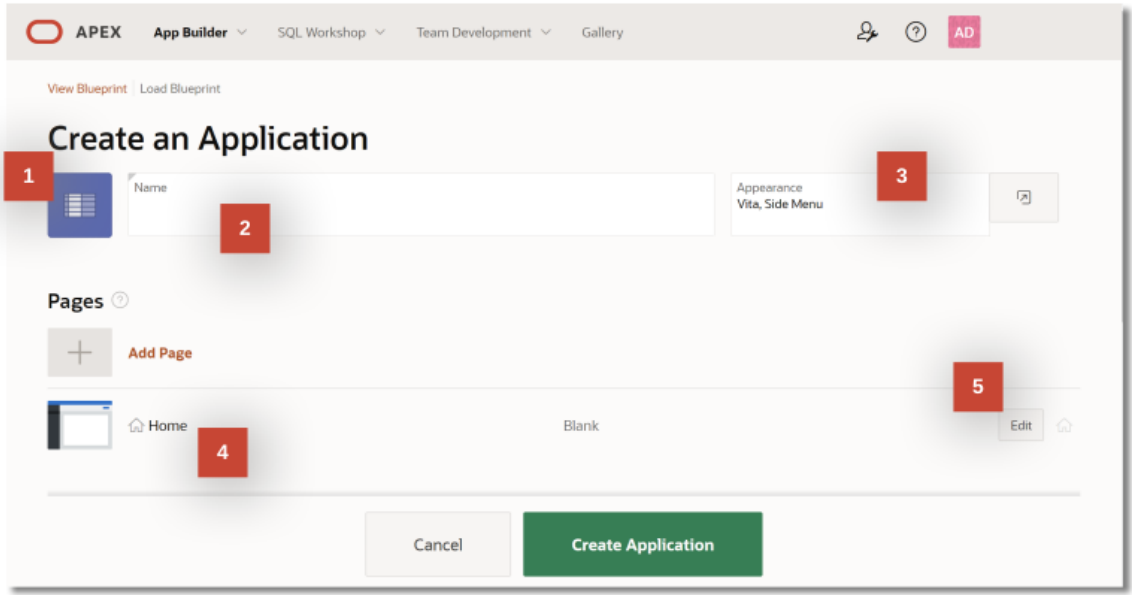



Abbildung 3.5: Eigenschaften und Seiten.

**Seitenerstellung für die Anwendung** Die Hauptaufgabe besteht jedoch darin, die Webanwendung schrittweise aufzubauen, indem sukzessive neue Seiten mithilfe des *„Create Page Wizard“* (gestartet durch *„Add Page“*, vgl. (4) in Abbildung 3.5) hinzugefügt werden. Sobald eine Seite erstellt ist, kann man während der Anwendungsentwicklung jederzeit die *Zusammensetzung* der Seite *editieren*, die Seitenreihenfolge *ändern* und diese *löschen* (vgl. (5) in Abbildung 3.5). Im unteren Teil der Abbildung 3.5 ist eine bereits erstellte Seite „Home“zu sehen. Weitere Einzelheiten über den „*Create Page Wizard*“finden Sie im Abschnitt 3.6.

**Funktionen und Einstellungen der Anwendung**. Zuletzt kann die Anwendung um einige recht nützliche *Funktionen* (vgl. (1) in Abbildung 3.6) wie eine „Info-Seite“, eine Seite für automatische „Zugangskontrolle“ oder eine Seite für „Benutzerfeedback“ sowie um einige allgemeine *Einstellungen* erweitert werden, z.B., den Namen des Schemas, das alle DB-Tabellen enthält und somit die DB-Schicht für die Anwendung oder die verwendete Sprache bildet (vgl. (2) in Abbildung 3.6).

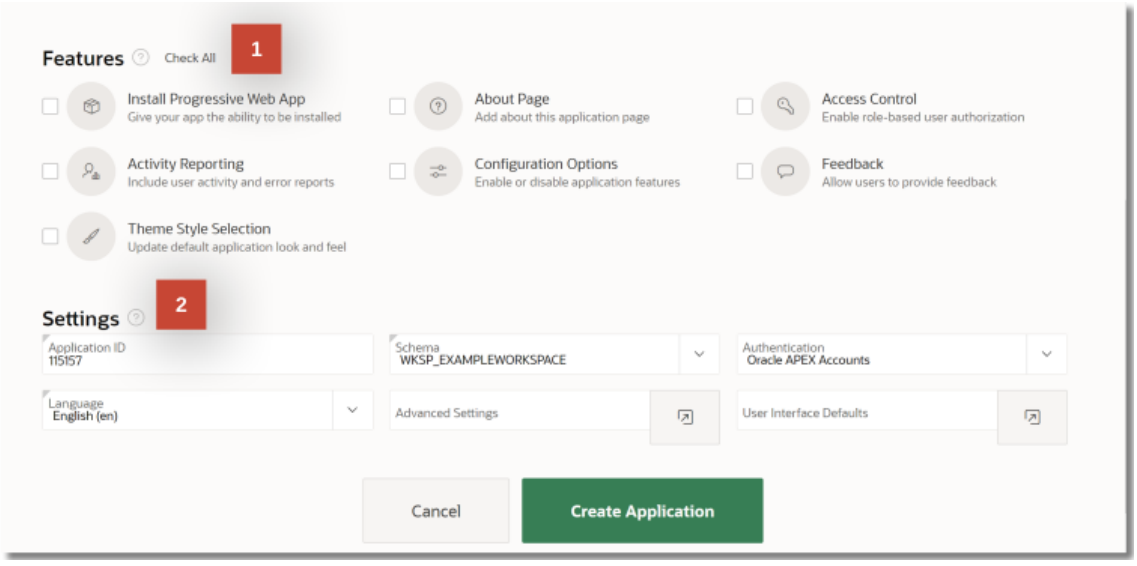



Abbildung 3.6: Funktionen und Einstellungen.

Nach einem Klick auf die Schaltfläche „*Create Application*“erscheint die „*App Builder Homepage*“, die die neu erstellte Anwendung mit ihren Seiten zeigt (vgl. Abbildung 3.7).

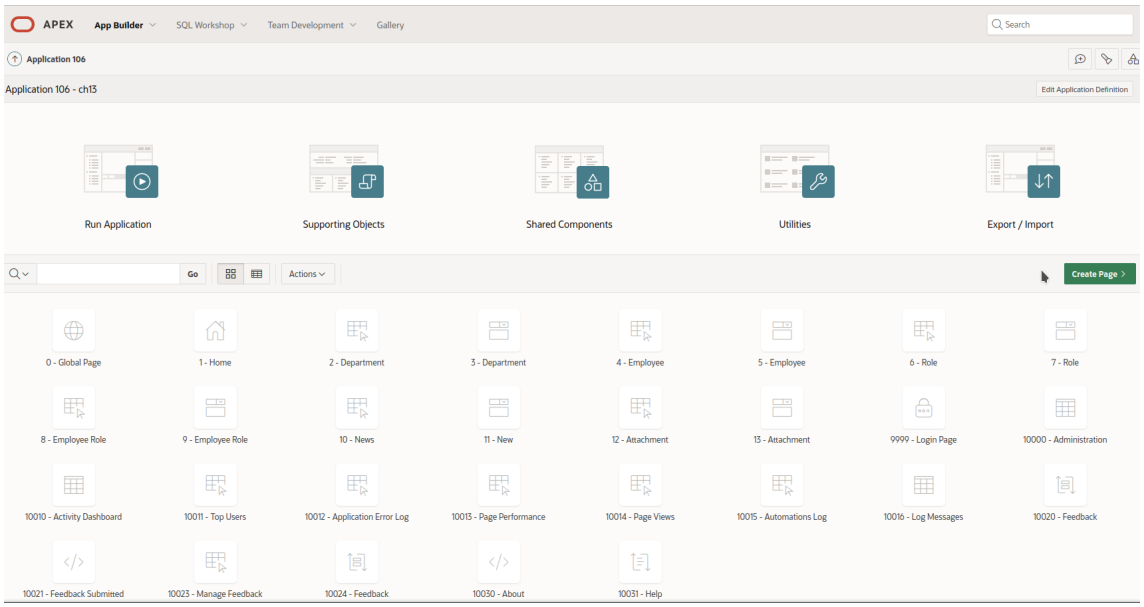



Abbildung 3.7: Anwendungshomepage - Entwickleransicht.

## 3.6 Seiten spezifizieren

**Was macht eine Seite aus?** Wie bereits erwähnt, ist eine Seite der Grundbaustein einer Webanwendung, d.h. jede Anwendung besteht aus mindestens einer oder mehreren Seiten. Jede Seite kann Schaltflächen und Felder (so genannte *„items“* ) enthalten, die in Containern namens *„regions“* gruppiert sind. Seiten können auch *Anwendungslogik* enthalten, z. B. *Berechnungen* (genannt *„computations“*) , und *Validierungen* durchführen , wie z.B. Korrektheitsprüfungen während der Datenmanipulation.

**Wählen Sie aus 18 verschiedenen Seitentypen.** Eine Seite kann Daten aus DB-Tabellen in verschiedenen Formen anzeigen, die durch so genannte „Seitentypen“bestimmt werden. Dazu gehören z.B. interaktive Berichte , Listen , Kalender oder Diagramme . Ein Seitentyp muss immer dann ausgewählt werden, wenn eine neue Seite erstellt wird, entweder über die Schaltfläche „Add Page“im Dialog „Create Application“ (vgl. Abbildung 3.5) oder über die Schaltfläche „Create Page“auf der Anwendungshomepage des Entwicklers (vgl. 3.8). Insgesamt werden 18 verschiedene Seitentypen angeboten, wie in Abbildung 3.8 dargestellt. Ein bestimmter Seitentyp bestimmt *Inhalt*, *Zusammensetzung* und *Layout* einer Seite, wie ein Blick auf die Seitentypen-Icons in Abbildung 3.8 erkennen lässt. Einzelheiten zu jedem dieser Seitentypen werden in den folgenden Kapiteln beschrieben.

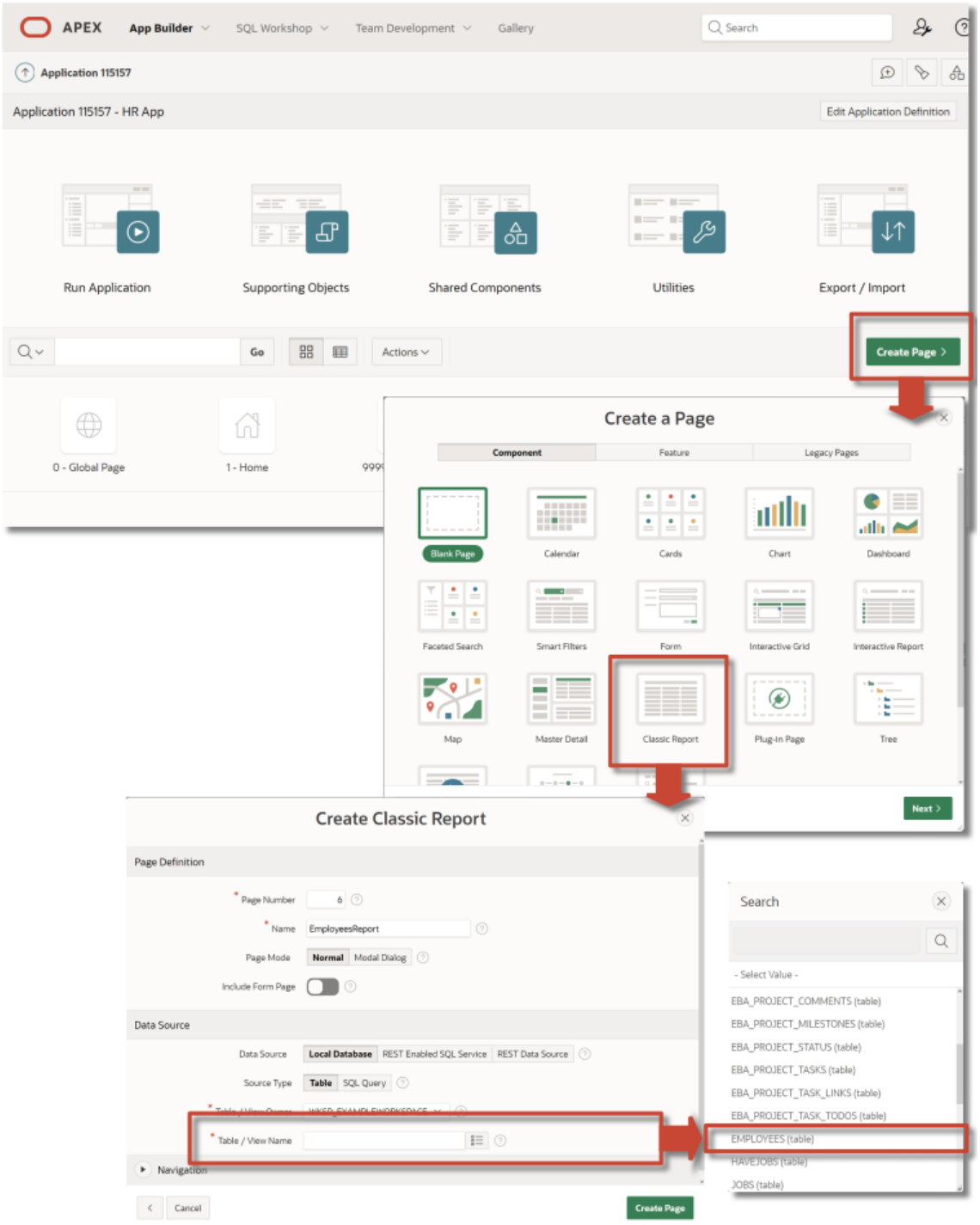



Abbildung 3.8: Erstellen einer Seite.

**Dialog zur Seitenspezifikation - Tabellen-Seiten-Zuordnung.** Bei der Entscheidung für einen bestimmten Seitentyp ist der konkrete Dialog zur Seitenspezifikation, den der „Page Creation Wizard“präsentiert, natürlich abhängig vom gewählten Seitentyp. Da jedoch fast alle verfügbaren Seitentypen auf DB-Tabellen basieren, besteht die Hauptaufgabe in der Regel darin, der Seite geeignete DB-Tabellen zuzuordnen, welche die Quelle für die Datenvisualisierung und das Ziel für die Datenmanipulation darstellen. Abbildung 3.8 zeigt einen Beispieldialog für den recht häufigen Seitentyp „*Classic Report*“. Es ist zu sehen, dass neben einigen grundlegenden Informationen wie „Name“auch die Zuordnung zur DB-Ebene angegeben werden muss, indem die „Datenquelle“in

Form der DB-Tabelle ausgewählt wird, die die Quelle für diesen Bericht sein soll (in Abbildung 3.8 wählen wir die Tabelle „Employees“unseres laufenden Beispiels). Es ist zu beachten, dass APEX von Haus aus SQL-DQL im Hintergrund verwendet, um die Daten aus der zugrundeliegenden DB-Tabelle anzuzeigen, sobald eine HTML-Seite gerendert wird.

Sobald eine neue Seite hinzugefügt wird, wird die „*Seitendesigner* “-Komponente von APEX (vgl. Abschnitt 3.7) automatisch für diese Seite geöffnet, so dass die Seitenspezifikation eingesehen und bearbeitet werden kann. Gleichzeitig kann die Seite über die Schaltfläche „Run“(vgl. Abschnitt 3.8) sofort automatisch in eine HTML-Seite umgewandelt werden, z.B. für Test- und Entwicklungszwecke.

## 3.7 Pflegen und Ändern einer Seite – Page Designer

Bereits erstellte Seiten können natürlich auch mit dem so genannten „*Page Designer*“weiter gepflegt und verbessert werden. Der Seitendesigner ist eine vollwertige *Integrated Development Environment (IDE)* mit einer Symbolleiste und mehreren Fenstern (vgl. Abbildung 3.9). Der Funktionsumfang des Page Designers ist enorm und ermöglicht unter anderem die vollständige Änderung der Zusammensetzung/des Layouts einer Seite und ihrer Komponenten (vgl. (1), (2) und (5)), einschließlich der Elemente, Schaltflächen und Bereiche (vgl. (3)), aber auch die Änderung der Darstellung der Seite oder der Anwendungslogik in Bezug auf Berechnungen und Validierungen (vgl. (4)).

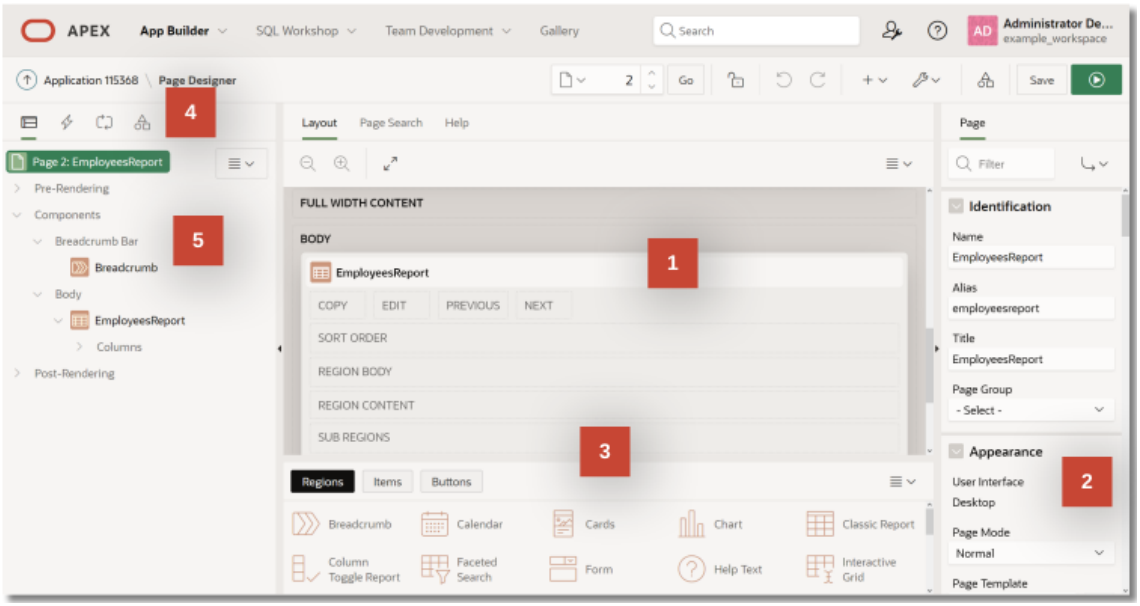



Abbildung 3.9: Seitendesigner.

Einzelheiten zu den Funktionen des Seitendesigners finden Sie in den folgenden Kapiteln.

## 3.8 Ausführen einer Anwendung oder Seite

Um eine gerenderte Version einer Anwendung oder einer einzelnen Seite anzuzeigen, muss diese der so genannten *Oracle APEX-Engine* vorgelegt werden, indem man auf der *Anwendungshomepage* auf die Schaltfläche „*Run*“klickt (vgl. Abbildung 3.10 und Abbildung 3.11). Die APEX-Engine rendert und verarbeitet die Seiten dynamisch in anzeigbare *HTML-Seiten*, basierend auf Daten über die Anwendung, die in „internen“DB-Tabellen gespeichert sind. Auf der *Anwendungshomepage* stehen zwei verschiedene Optionen zur Verfügung, je nachdem, ob Sie die gesamte Anwendung, d.h. alle Seiten, oder nur eine einzelne Seite, z.B. zu Testzwecken, ausführen möchten.

- **Die gesamte Anwendung ausführen**, d.h. alle Seiten: Diese Option ist auf der Homepage der Anwendung verfügbar, wie in Abbildung 3.10 zu sehen ist.

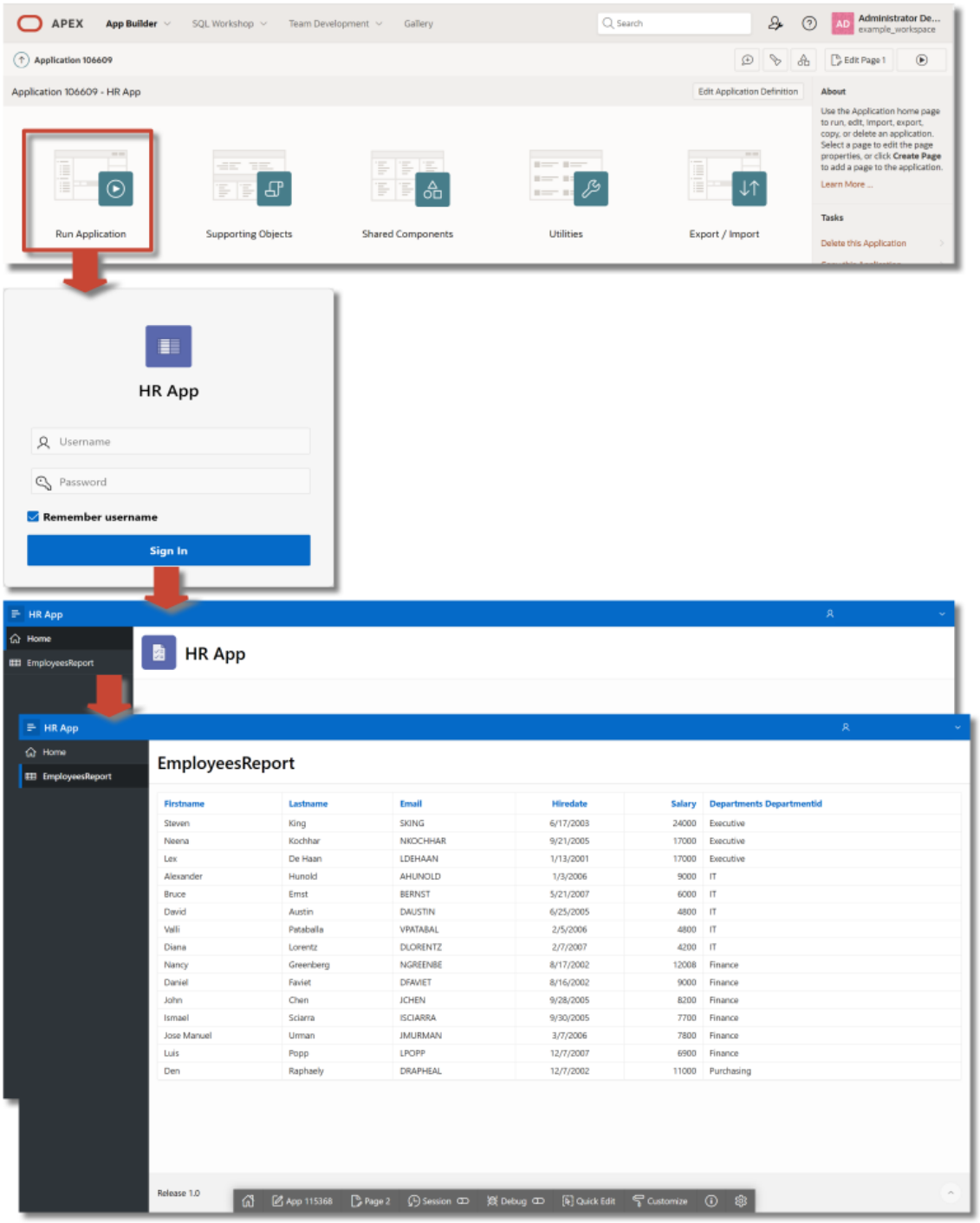



Abbildung 3.10: Gesamte Anwendung ausführen.

- **Einzelne Seiten ausführen**. Wenn Sie neue Seiten erstellen, können Sie auch jede Seite einzeln ausführen, um einen unmittelbaren Eindruck vom Aussehen und der Handhabung der aktuellen Seite zu erhalten. Die Schaltfläche „Run Page“ähnelt einem kleinen, schwarzen Play-Icon und wird auf der rechten Seite jeder Seite Ihrer Anwendung angezeigt, wie in Abbildung 3.11 zu sehen ist. Es ist jedoch zu beachten, dass auch wenn man nur eine einzelne Seite ausführt, auch alle anderen Seiten verfügbar sind.

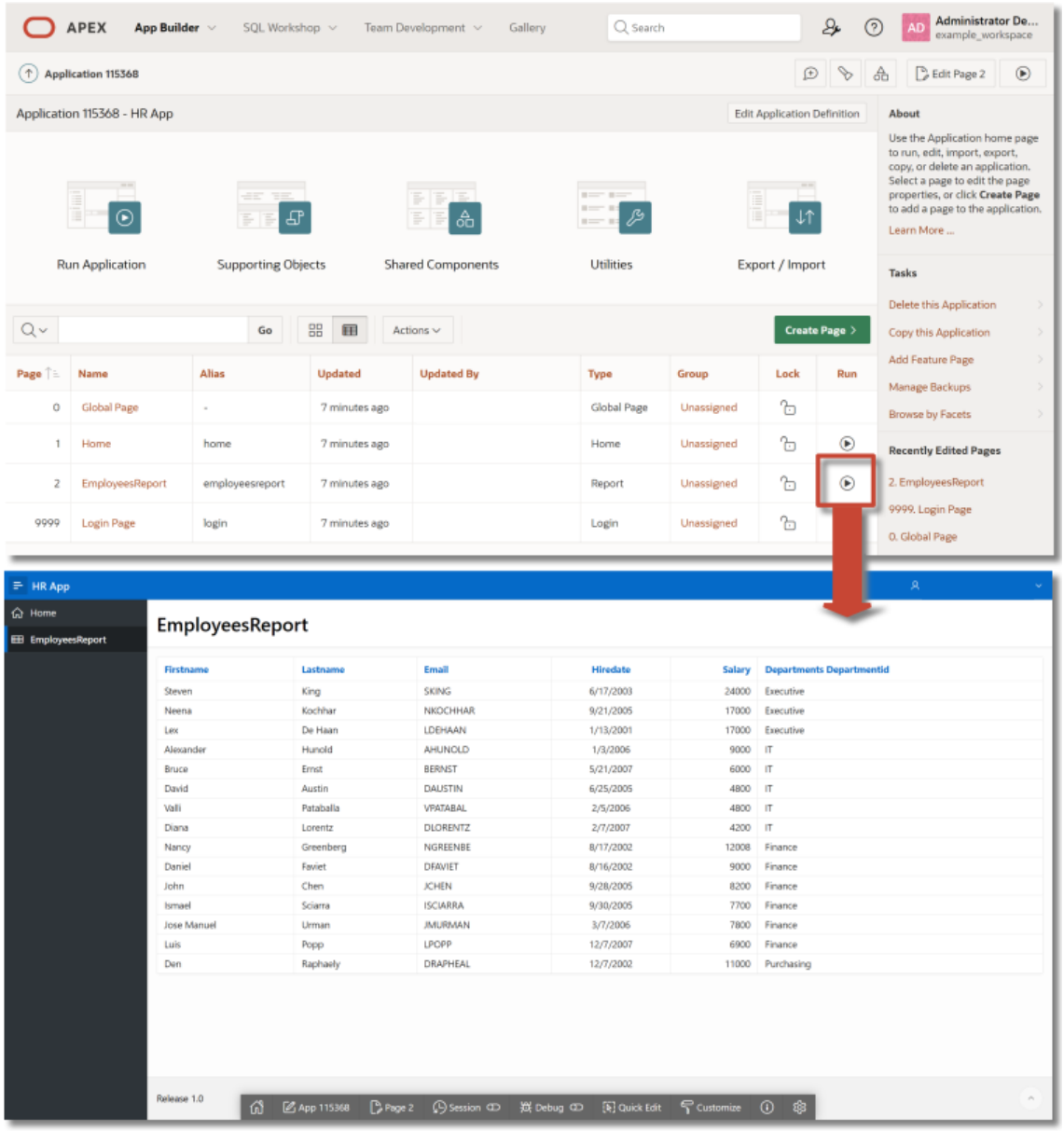



Abbildung 3.11: Einzelne Seiten ausführen.

Beachten Sie, dass die anderen Optionen, die neben „Run Application“auf der Anwendungshomepage zur Verfügung stehen (z. B. Unterstützende Objekte), in späteren Kapiteln behandelt werden.



# 4. Wie man Daten mit APEX austauscht?

E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER

Die Hauptaufgabe beim Aufbau der DB-Schicht für eine Web-Anwendung ist die Erstellung der DB-Tabellen und das Einfügen der entsprechenden Daten (siehe Kapitel 2). Dies kann jedoch nicht nur manuell erfolgen, wenn die Daten noch nicht vorhanden sind. Falls bereits Daten in externen Dateien vorhanden sind, können diese einfach wiederverwendet und in die Oracle DB *importiert* werden. Gleichzeitig könnte es wünschenswert sein, bereits vorhandene Daten von Zeit zu Zeit in eine Datei zu *exportieren*, damit sie in einem zweiten Schritt von einer anderen, externen Anwendung wiederverwendet werden können. Und schließlich besteht für den Fall, dass einigen externen Clients (z. B. anderen Web-/Mobil-/Legacy-Anwendungen oder Cloud-basierte Dienste) ein *online-Zugriff* zu den Daten innerhalb unserer DB zur Verfügung gestellt werden soll, statt nur von Zeit zu Zeit Dateien auszutauschen, auch die Möglichkeit, einen indirekten Online-Zugriff auf unsere DB-Daten zu ermöglichen.

Diese drei Anwendungsfälle für den Datenaustausch sind in Abbildung 4.1 dargestellt, die einen Überblick über die verschiedenen Optionen gibt, die Oracle APEX für den Import, Export und indirekten Zugriff auf DB-Daten bietet. Dieses Kapitel soll einen Überblick über diese verschiedenen Möglichkeiten geben.

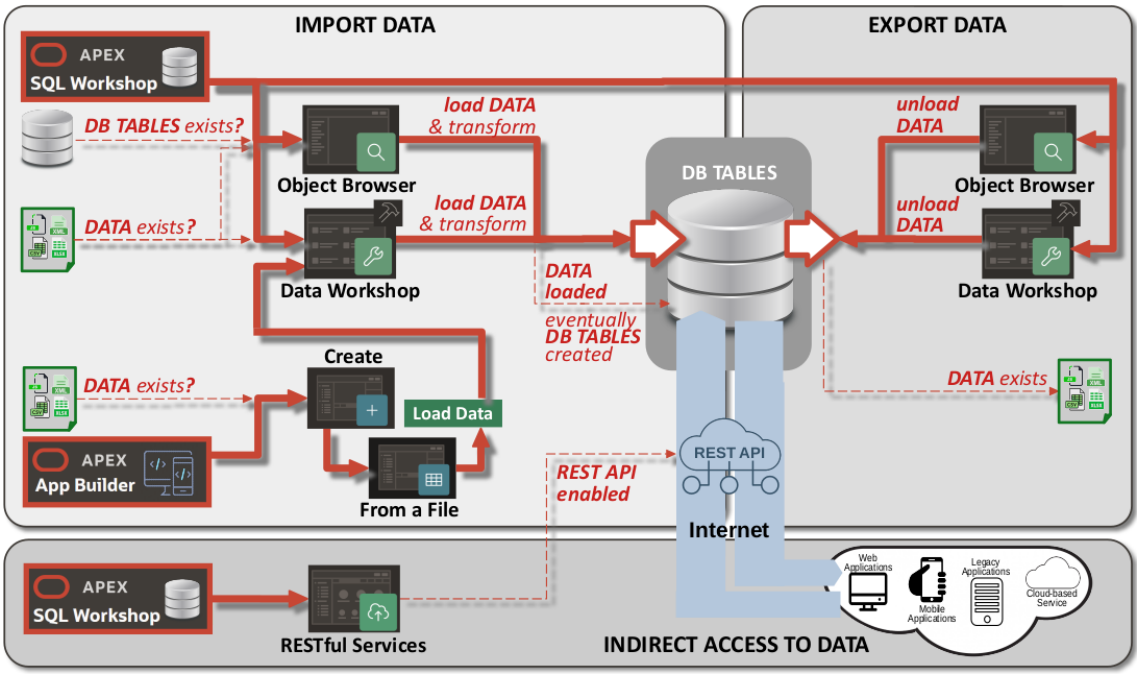



Abbildung 4.1: Datenaustauschoptionen im APEX.

## 4.1 Importieren und Exportieren von Daten mit "Data Workshop"

**Anwendungsszenario für den Data Workshop.** Der Data Workshop ist ein einfach zu bedienendes Werkzeug, das einen "Wizard" zur Verfügung stellt, um die Aufgabe des Imports/Exports von Daten aus/in externe/n Dateien zu vereinfachen. Der Data Workshop eignet sich besonders für Anwendungsszenarien, die sich mit Daten mittlerer Größe (weniger als 10 Tabellen) befassen, die nur einfache Standard-Datentypen haben (z.B. keine mehrwertigen Felder oder verschachtelte Strukturen). Es ist zu beachten, dass Oracle für die Wiederverwendung und den Import von *großen und komplexen Datensätzen* andere geeignete Werkzeuge zur Verfügung stellt, z.B. in Form des sogenannten "*SQL*Loader Utility*". Im Folgenden wird zunächst in den Abschnitten 4.1.1 und 4.1.2 auf den Import von Daten eingegangen, während in Abschnitt 4.1.3 der Export von Daten beschrieben wird.

### 4.1.1 Importieren von Daten

**Dateiformate.** Insgesamt ermöglicht der Data Workshop das Laden von Daten aus externen Dateien, die den folgenden Formaten entsprechen:

1. Ein beliebiges standardmäßig *abgrenztes Format*, das *tab-getrennt* oder *Komma-getrennt* ("*CSV*" - *Comma Separated Values*) ist.
2. *XLSX-Dateien* (d.h., *Excel-Arbeitsmappen*). Beachten Sie, dass, wenn die hochgeladene XLSX-Datei mehrere Arbeitsblätter enthält, das erste Blatt standardmäßig ausgewählt wird. Um ein anderes Blatt zu laden, können Sie es aus der Auswahlliste "Select Sheet" auswählen.
3. *JSON-Dateien* (*Java Script Object Notation*). Beachten Sie, dass nur eine Verschachtelungsebene unterstützt wird.
4. *XML-Dateien* (*eXtended Markup Language*). Beachten Sie, dass, ähnlich wie bei JSON-Dateien, nur eine Verschachtelungsebene unterstützt wird.

**Zugriff auf Data Workshop für Importe.** Der Zugriff auf den Data Workshop kann auf zwei verschiedene Arten erfolgen:

1. Starten Sie den Data Workshop aus dem SQL Workshop heraus, wie in Abbildung 4.2 dargestellt.

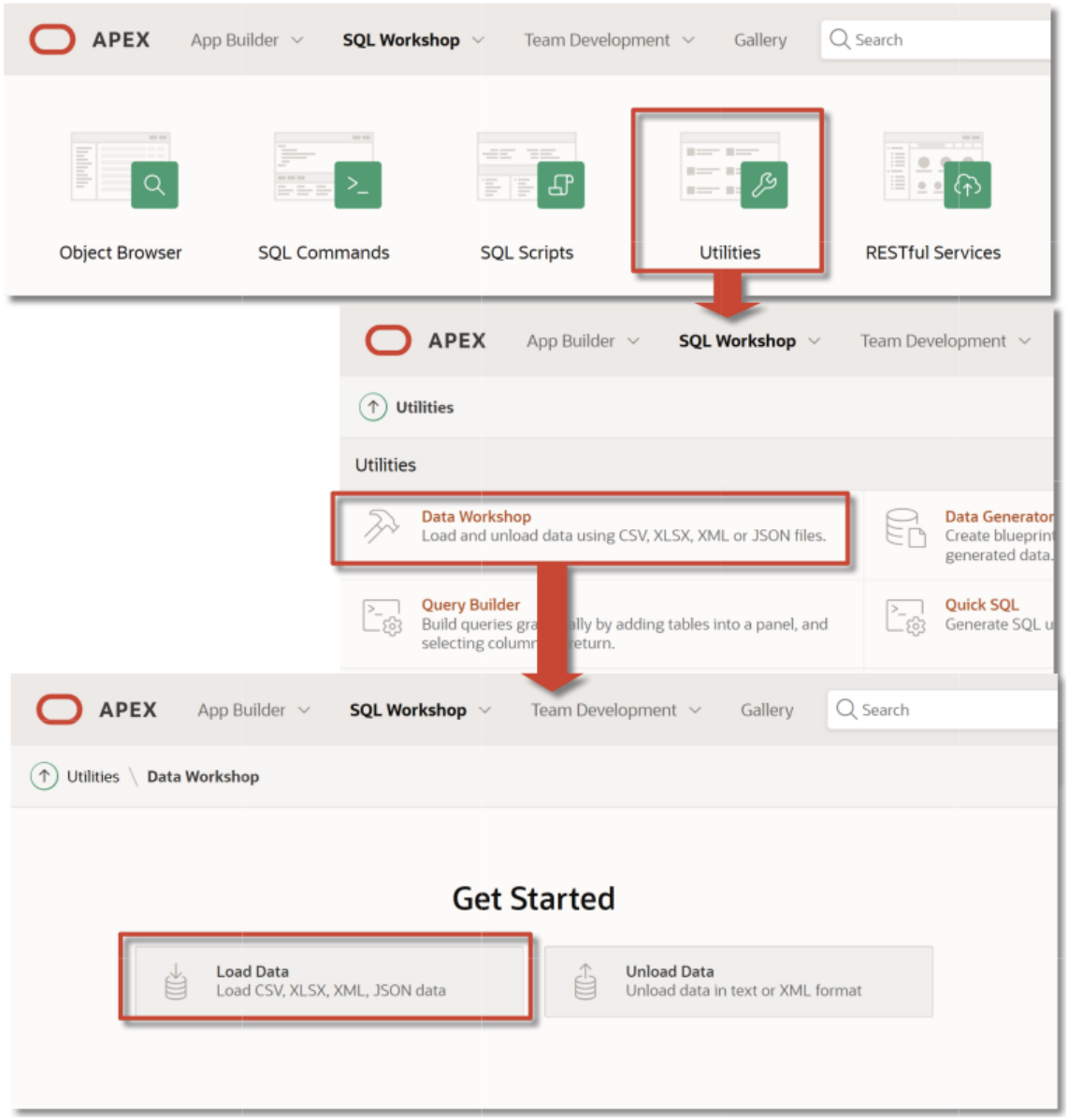



Abbildung 4.2: Verwendung des Data Workshops zum Laden von Daten - Zugriff über den SQL Workshop.

2. Alternativ kann der Data Workshop auch aus dem APEX App Builder bei der Erstellung einer Anwendung, wie bereits in Kapitel 3 erwähnt (vgl. Abbildung 4.3), über die Option "From a File" aufgerufen werden (vgl. Abbildung 4.1).

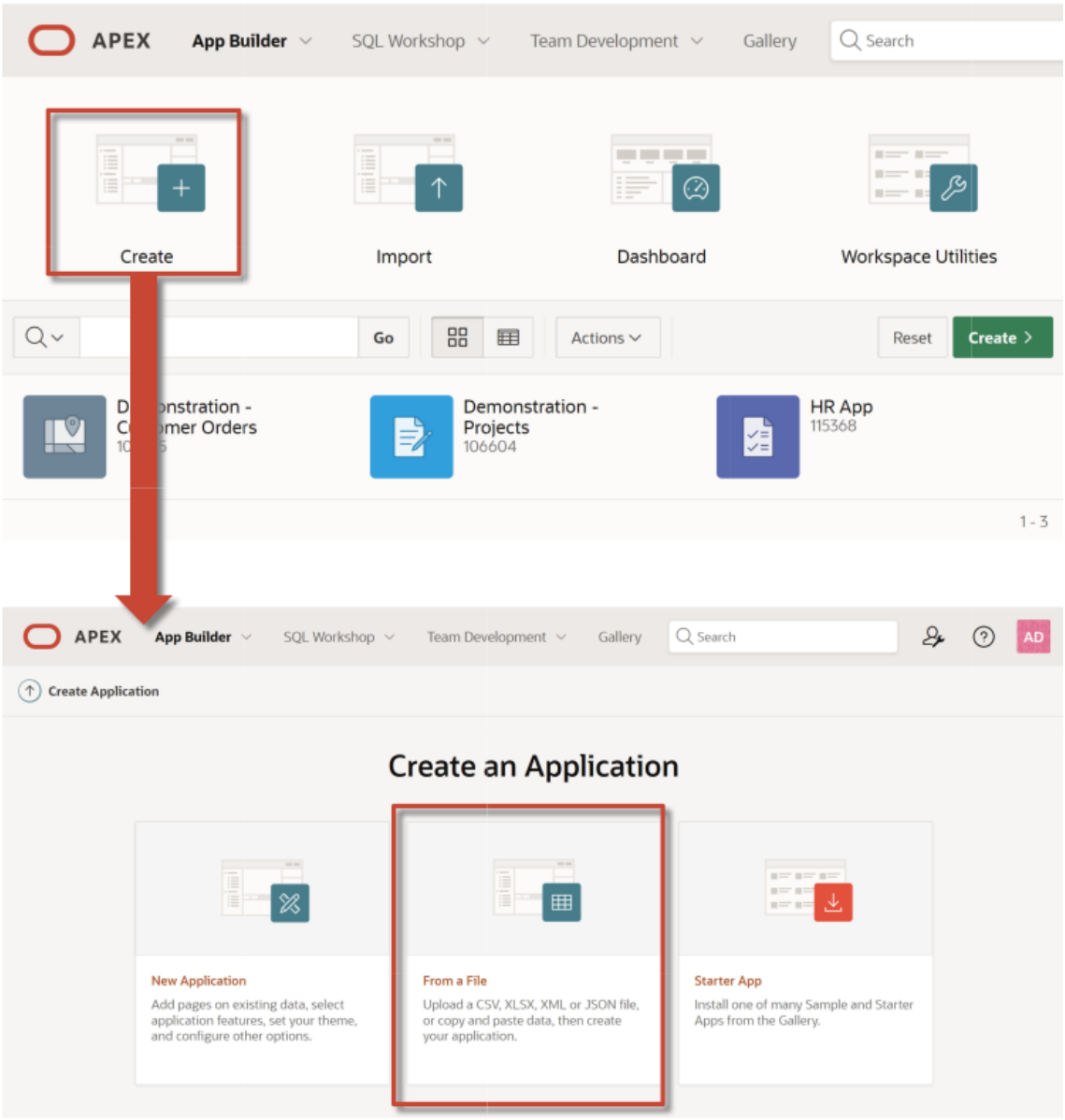



Abbildung 4.3: Verwendung des Data Workshops zum Laden von Daten - Zugriff über den App Builder.

### 4.1.2 Schritte zum Importieren

Im Folgenden werden die für den Datenimport erforderlichen Schritte kurz erläutert und einige wichtige Fragen zum Verständnis der Verwendung des "Load Wizard" gestellt:

**Bereitstellung der Datenquelle.** Der erste, ganz natürliche Schritt besteht darin, die Datenquelle bereitzustellen, was entweder durch Hochladen (per Drag/Drop oder Dateiauswahl) von durch Trennzeichen getrennten Dateien, XLSX-, JSON- oder XML-Dateien oder durch einfaches Kopieren/Einfügen von durch Trennzeichen getrennten Daten geschehen kann (vgl. Abbildung 4.4). Beachten Sie, dass dieser Dialog erscheint, wenn Sie entweder auf die Schaltfläche *"Load Data"* (vgl. Abbildung 4.2) oder auf die Schaltfläche *"From a File-button"* (vgl. Abbildung 4.12) klicken.

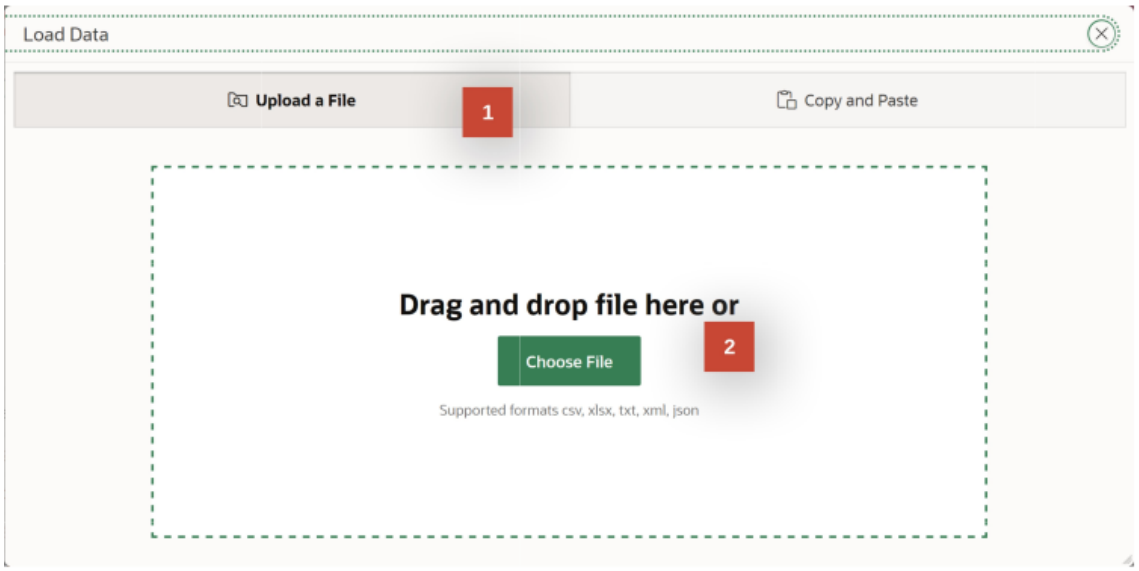



Abbildung 4.4: Bereitstellung der Datenquelle.

**Konfiguration des Data-to-Table Mapping.** Der zentrale Schritt des Ladeassistenten ist die Konfiguration der Daten-Tabellen-Zuordnung, die in Abbildung 4.5 und 4.6 visualisiert ist, wobei die wichtigsten Konfigurationsoptionen mit den Nummern (1)-(4) und (1)-(8) gekennzeichnet sind und im Folgenden beschrieben werden.

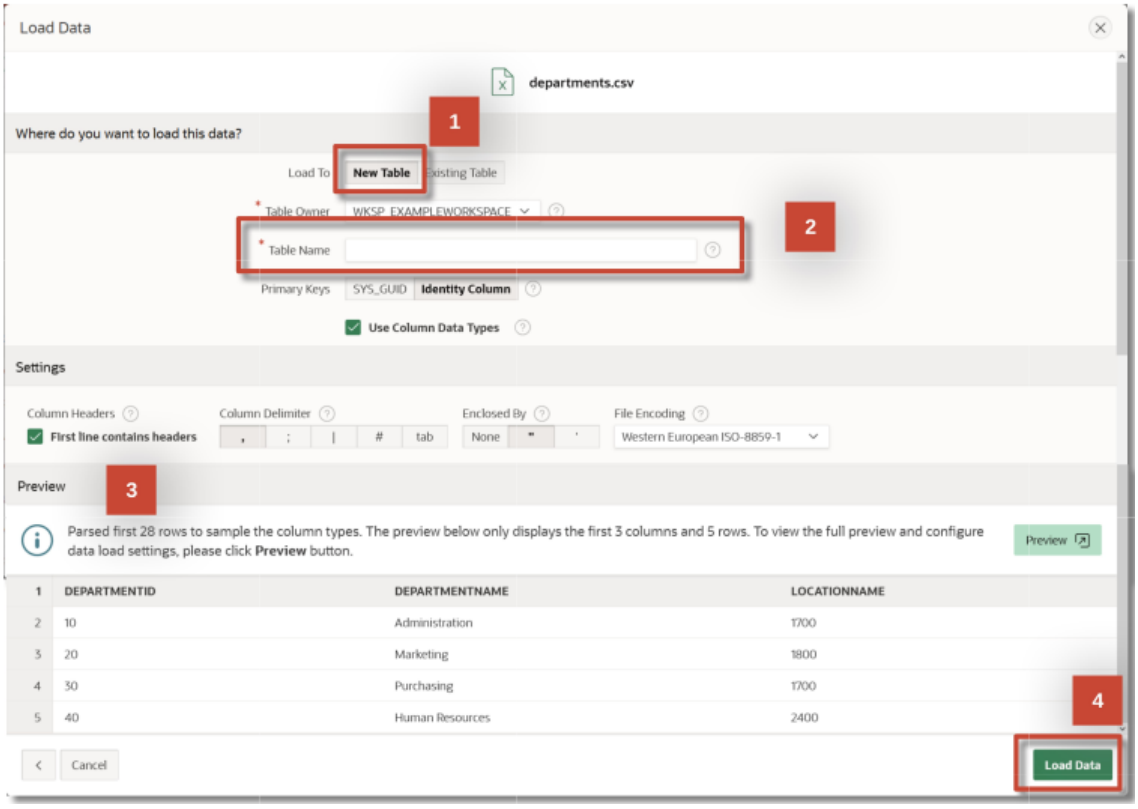



Abbildung 4.5: Laden von Daten in eine neue Tabelle.

1. **Daten in bestehende oder neue Tabelle laden.** Zunächst muss entschieden werden, ob die Daten in einer bereits vorhandenen DB-Tabelle gespeichert werden sollen oder ob eine neue

DB-Tabelle verwendet werden soll (vgl. Abbildung 4.5 (1) bzw. Abbildung 4.6 (1)).

2. **Eigentümer und Name der Tabelle angeben.** In beiden Fällen müssen Tabellenbesitzer und Tabellenname angegeben werden (vgl. Abbildung 4.5 (2) bzw. Abbildung 4.6 (2)).
3. **Einstellungen für Dateien mit Trennzeichen.** Der Abschnitt *Einstellungen* (vgl. Abbildung 4.5 (3)) ermöglicht drei verschiedene wichtige Konfigurationen, die speziell für Dateien mit Trennzeichen ("CSV") gelten. Mit der Konfigurationsoption "*Column Headers*" lässt sich festlegen, ob die erste Zeile der Daten die Spaltennamen enthält. Das "*Spaltentrennzeichen*" kann frei gewählt werden, wie im Abschnitt "Einstellungen" der Abbildung 4.5 (3) zu sehen ist, so dass Oracle die Struktur aus der externen Datei *extrahieren* kann - ein Prozess, der als "*parsing*" bezeichnet wird und die Voraussetzung für die Abbildung der Daten in der Tabelle ist. Durch die Einstellung "*Enclosed by*" kann die Anfangs- und Endgrenze eines Datenwertes abgegrenzt werden. Wenn Sie ein Abgrenzungszeichen angeben, ignoriert der Data Workshop Leerzeichen, die vor der Anfangs- und Endbegrenzung eines Datenwerts stehen. Sie können diese Option auch verwenden, um einen Datenwert mit dem angegebenen Abgrenzungszeichen zu umschließen.
4. **Textvorschau für geparste Daten.** Wie bereits erwähnt, parst Oracle die Daten automatisch, um die Struktur der Daten zu extrahieren und dabei z.B. auch geeignete Datentypen zu erkennen. Unten in Abbildung 4.5 (4) wird ein kleines Beispiel (bis zu 10 Spalten und 5 Zeilen) des Ergebnisses dieses Parsing-Prozesses gezeigt.
5. **Erweiterte Vorschau und weitere Konfigurationen.** Wenn man auf den "*Preview*"-Button klickt, werden maximal 100 Spalten und 100 Zeilen angezeigt und es werden weitere Konfigurationsmöglichkeiten angeboten, die es nun auch erlauben, die automatisch vorgeschlagenen Datentypen zu ändern und zu wählen, welche der Spalten tatsächlich geladen werden sollen. Hat man sich in Schritt (1) entschieden, die Daten in eine bestehende Tabelle zu laden, muss diese aus der Liste der verfügbaren Tabellen ausgewählt werden (vgl. Abbildung 4.6 (2) und (3)). Anschließend kann durch Anklicken der Schaltfläche *"Configure"* (vgl. Abbildung 4.6 (4)) die *Spaltenzuordnung* festgelegt werden, d.h. welche der automatisch identifizierten Spalten der geladenen Daten auf welche Spalten innerhalb der DB-Tabelle abgebildet werden sollen. Wie in Abbildung 4.6 (5) und (6) zu sehen ist, wird die Spalte `DEPARTMENTNAME` mit dem Datentyp `VARCHAR2`, die während des Parsens automatisch aus der externen Datei extrahiert wurde, nun auf die DB-Tabellenspalte namens "DEPARTMENTNAME" mit demselben Datentyp abgebildet.

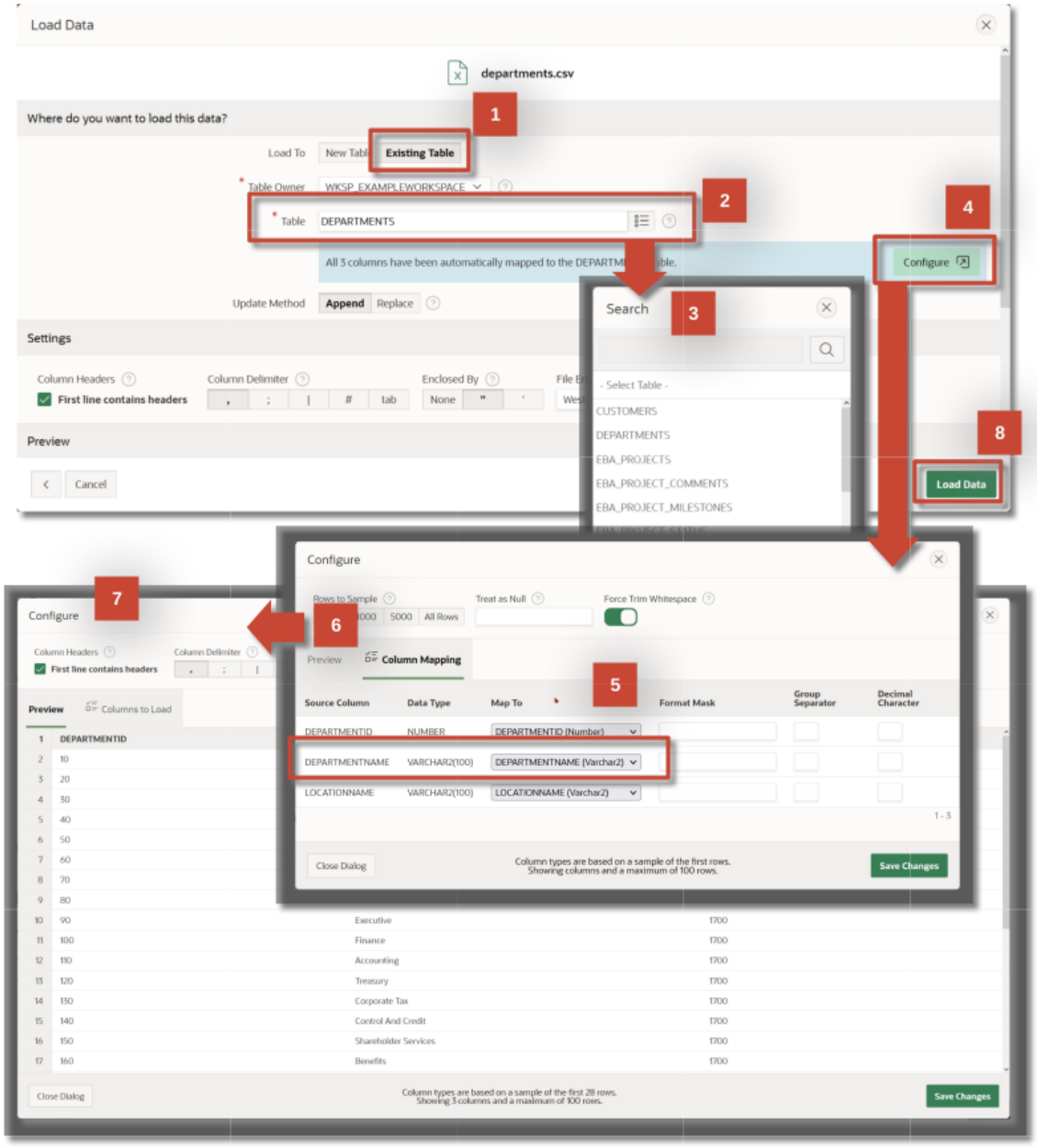



Abbildung 4.6: Laden von Daten in eine bestehende Tabelle.

6. **Daten laden.** Durch Anklicken der Schaltfläche *"Load Data"* (vgl. Abbildung 4.6 (8)) wird schließlich eine neue DB-Tabelle generiert (falls diese Option in (1) gewählt wurde) und die Daten in die neue bzw. die bestehende Tabelle geladen, wobei der Ladedialog darüber informiert, wie viele Zeilen geladen wurden. Das Laden der Daten läuft im Hintergrund ab, was beim Hochladen einer größeren Datei von Vorteil ist, da der Dialog dann ausgeblendet werden kann. Während dieses Ladevorgangs kann es auch vorkommen, dass man auf einige *fehlerhafte Zeilen* stößt, die nicht in die Zieltabelle eingefügt werden können, da z.B. der eine oder andere Wert in Konflikt mit einem Datentyp steht. In diesem Fall werden diese fehlerhaften Zeilen automatisch in einer *error table* gespeichert und können manuell nachbearbeitet werden. Man wird auch durch den Ladedialog über eventuelle fehlerhafte Zeilen informiert. Die resultierende Tabelle (und eine eventuelle Fehlertabelle) kann nun durch Navigation zum "Object Browser" eingesehen werden oder es kann sofort mit der Entwicklung der Webanwendung fortgefahren werden.

### 4.1.3 Exportieren von Daten

Der “Export-Assistent” des Data Workshops kann auf die gleiche Weise aufgerufen werden, wie im Abschnitt 4.1.1 für den “Lade-Assistenten” beschrieben (vgl. Abbildung 4.7).

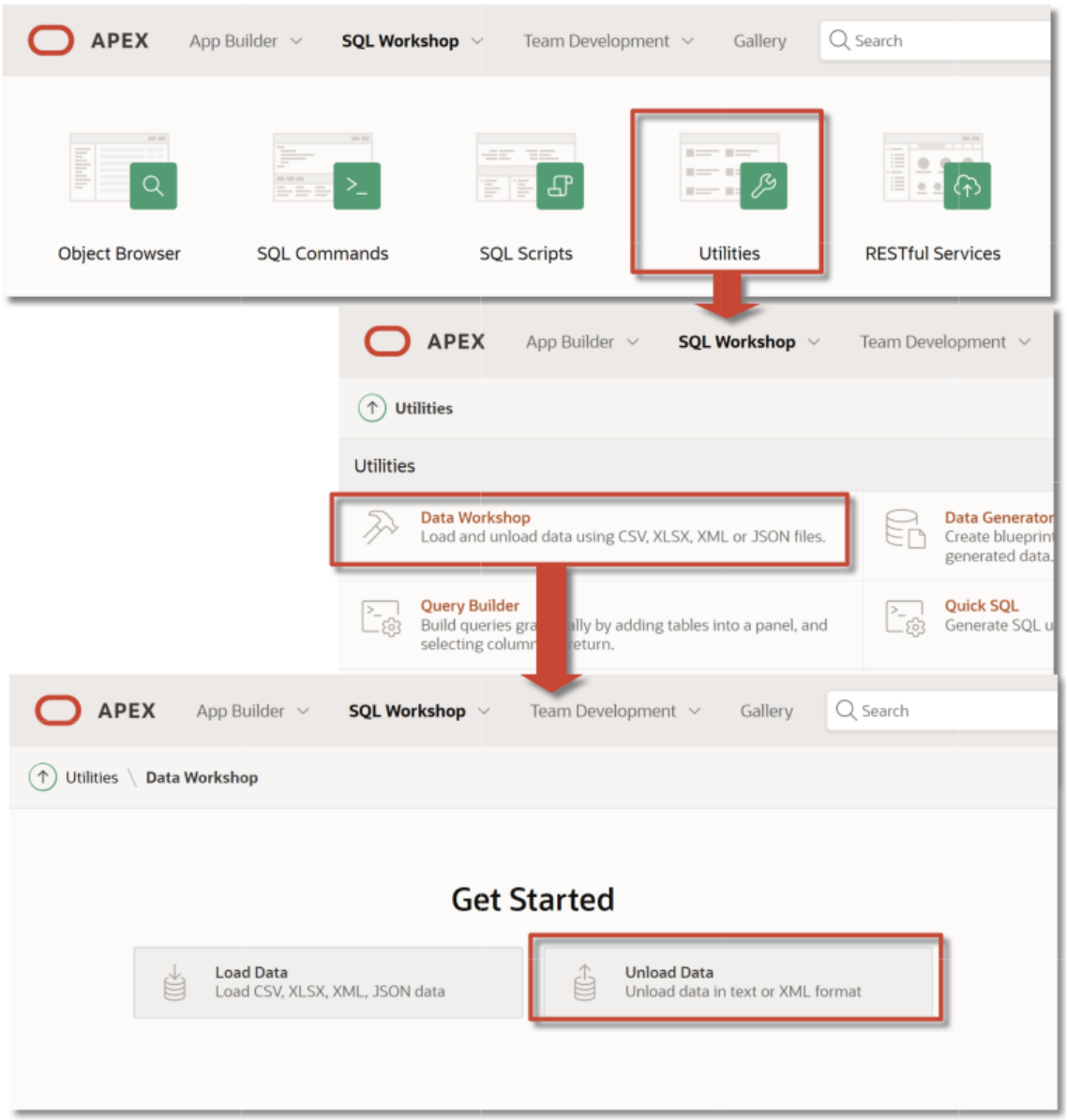



Abbildung 4.7: Exportieren (“Entladen”) von Daten.

In einem ersten Schritt muss das Dateiformat ausgewählt werden, wobei der Data Workshop es ermöglicht, Daten in externe Dateien zu exportieren, die in Form von CSV oder XML formatiert sind (vgl. Abbildung 4.8 (1)). Der nächste Schritt ist die Auswahl der Tabelle sowie der entsprechenden Spalten, deren Daten exportiert werden sollen (vgl. Abbildung 4.8 (2) bzw. (3)). Falls XML als Exportformat gewählt wurde, kann nun der eigentliche Export gestartet werden. Im Falle von CSV können einige weitere Optionen ausgewählt werden, die das Trennzeichen zwischen den verschiedenen Zeilen festlegen, ob die Zeilennamen in der Ausgabedatei enthalten sein sollen und ob das Datenformat DOS oder UNIX sein soll (vgl. Abbildung 4.8 (4)), bevor Sie schließlich “Unload Data” aktivieren, um den Datenexport zu speichern (vgl. Abbildung 4.8 (5)).

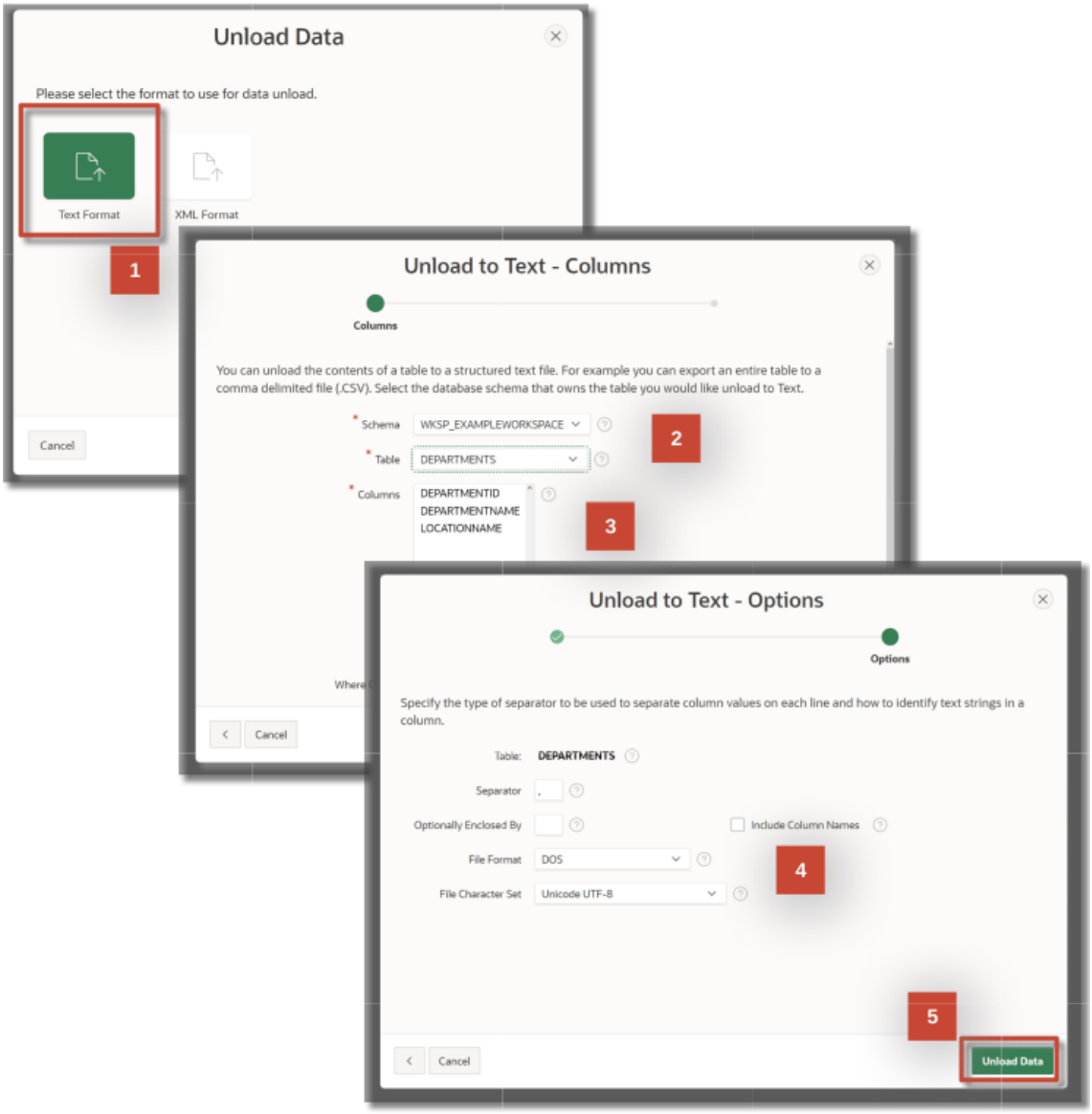



Abbildung 4.8: Datenentlade-Assistent.

Bezüglich DOS vs. UNIX ist zu beachten, dass es einen feinen Unterschied zwischen diesen beiden Optionen gibt: DOS-Dateien haben andere Zeilenenden als unter Unix/Linux erstellte Dateien. DOS verwendet carriage return und line feed ("`\r\n`") als Zeilenende, während Unix nur line feed ("`\n`") verwendet. Daher müssen Sie beim Übertragen von Dateien zwischen Windows- und Unix-Rechnern darauf achten, dass die Zeilenenden richtig gesetzt sind.

## 4.2 Importieren und Exportieren von Daten mit dem "Object Browser"

Eine weitere Alternative ist die Verwendung des "Object Browsers" von SQL Workshop zum Importieren von Daten aus und Exportieren von Daten in eine Datei, wobei dies tabellenweise erfolgen kann. Dazu muss zunächst die entsprechende Tabelle ausgewählt werden (vgl. Abbildung 4.9 (1)).

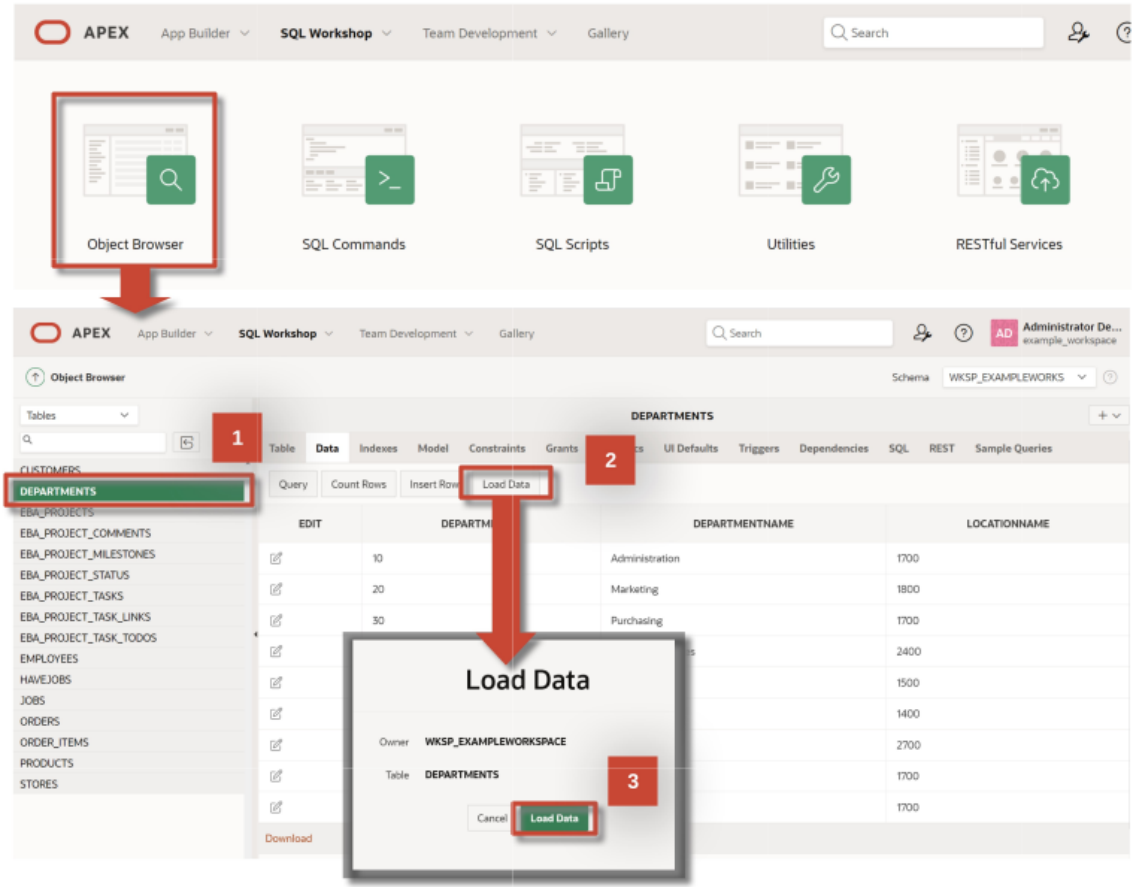



Abbildung 4.9: Import/Export von Tabellendaten mit dem Object Browser.

Sollen nun Daten *importiert* werden, so erscheint nach Anklicken der *"Load Data-Schaltfläche"* derselbe Dialog wie bei der Auswahl von "Load Data" (vgl. Abbildung 4.9 (2), (3)), wie bereits im Abschnitt 4.1.1 beschrieben - auch werden dieselben Dateiformate unterstützt. Auf der anderen Seite kann der *Export* von Daten einfach durch Anklicken der *"Download-Schaltfläche"* am unteren Ende der Seite erfolgen, was zur sofortigen Erzeugung einer CSV-Datei mit den Daten der DB-Tabelle führt.

## 4.3 Exportieren des Ergebnisses eines SQL-Befehls

Wie beim Export der gesamten in einer Tabelle gespeicherten Daten kann auch eine bestimmte Teilmenge dieser Daten - oder sogar die in mehreren Tabellen gespeicherten Daten - auf einmal exportiert werden. Vereinfacht ausgedrückt, kann die Ergebnismenge einer beliebig komplexen SQL-Abfrage exportiert werden (vgl. Abbildung 4.10 (1))

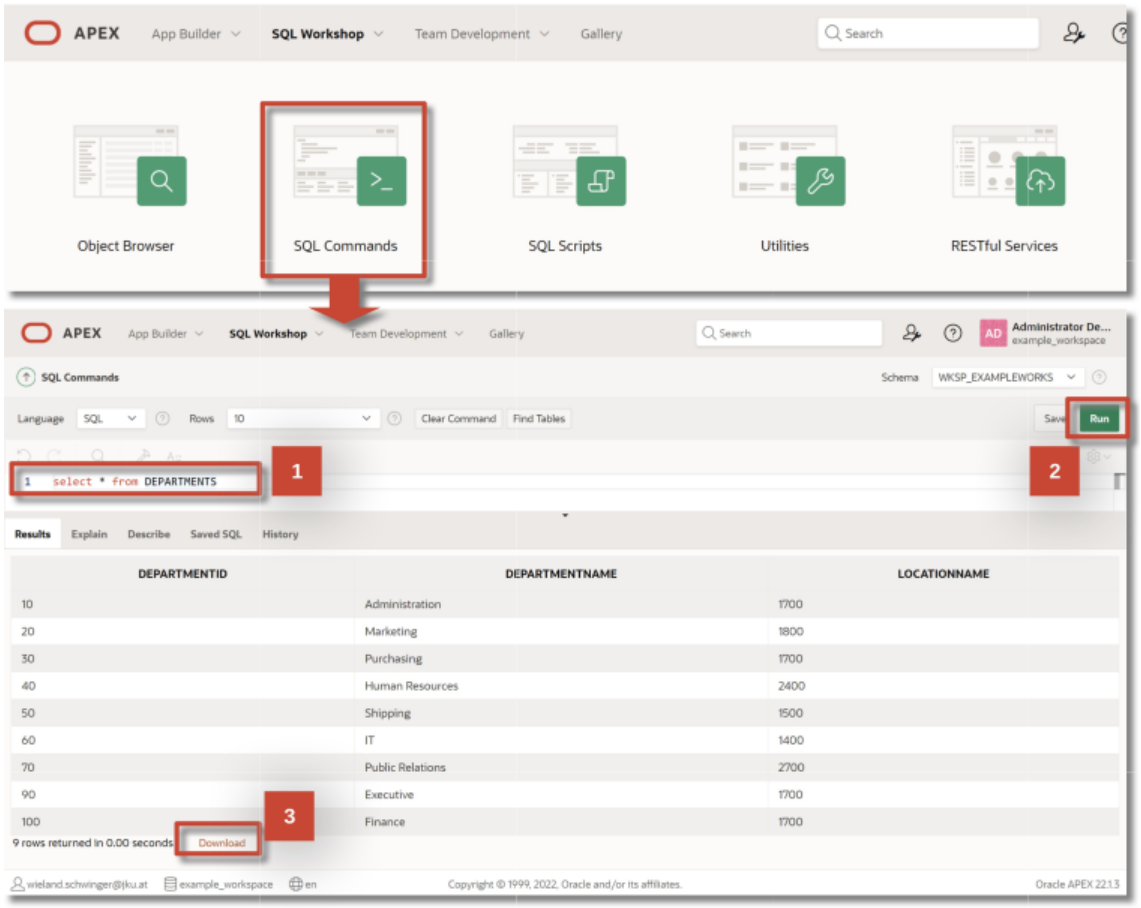



Abbildung 4.10: Datenexport des Ergebnisses einer SQL-Abfrage.

Nachdem die Abfrage im “SQL command”-Werkzeug des SQL Workshops ausgeführt wurde und die Ergebnismenge angezeigt wird (vgl. Abbildung 4.10 (2)), kann diese Ergebnismenge durch einen Klick auf den “Download”-Button heruntergeladen werden (vgl. Abbildung 4.10 (3)).

## 4.4 Exportieren von Daten aus einem Anwendungsbericht

Wie in Kapitel 3 beschrieben, ist es bei der Entwicklung einer Web-Anwendung mit dem APEX App Builder erforderlich, für jede Seite, die der Web-Anwendung hinzugefügt wird, einen bestimmten Seitentyp auszuwählen. Wenn man z.B. den Seitentyp “Interactive Report” ausgewählt hat, können die Daten, die dieser Bericht zur Laufzeit anzeigt, exportiert werden. Dies wird in Abbildung 4.11 veranschaulicht, die ein Beispiel für die interaktive Berichtsseite “DepartmentsReport” zeigt.

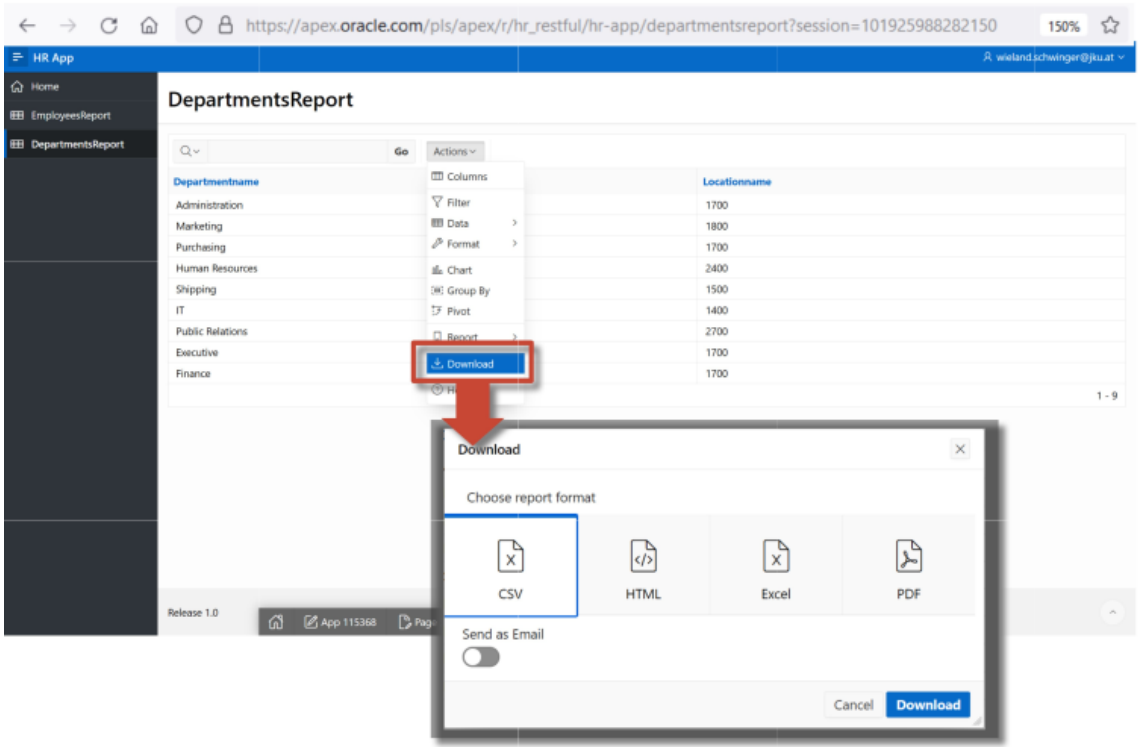



Abbildung 4.11: Export von Daten aus einem Anwendungsbericht.

Mögliche Dateiformate sind CSV, HTML, PDF und Excel, auch der Versand der Daten per eMail ist eine Option.

## 4.5 Ermöglichung des Datenaustauschs mit RESTful Services

Der Austausch von Daten der DB-Schicht unserer Webanwendung, indem wir externen Clients wie anderen Webanwendungen, mobilen Anwendungen, Legacy-Anwendungen oder Cloud-Systemen den Online-Zugriff auf unsere Daten ermöglichen, lässt sich mit sogenannten *RESTful services* leicht realisieren. *REST (Representational State Transfer)* ist eigentlich ein *Architekturmuster*, das von Roy Fielding und Richard Taylor im Jahr 2000 [2] vorgeschlagen wurde, um Interoperabilität zwischen beliebigen Systemen über das Internet zu ermöglichen. Es ermöglicht die Abfrage und Manipulation von Daten, *ohne dass ein direkter Zugriff* auf die zugrunde liegenden DB-Tabellen erforderlich ist. Abbildung 4.12 gibt einen Überblick über das grundlegende Architekturmuster von REST.

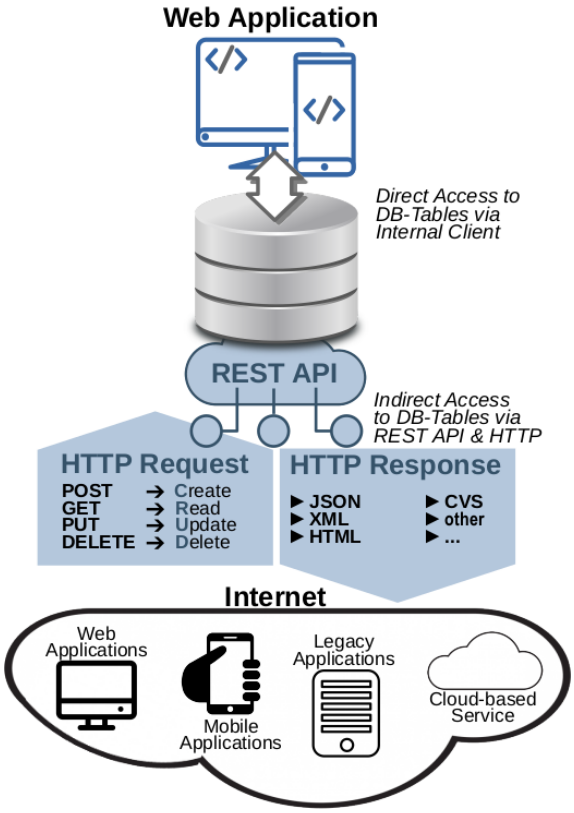



Abbildung 4.12: RESTful Services für den Datenaustausch - Grundlegende Architektur.

Um einen solchen *indirekten* Zugriff gemäß dem REST-Architekturmuster zu realisieren, müssen RESTful-Dienste in Form einer Reihe von *APIs (Application Programming Interfaces)* auf Ihren DB-Tabellen erstellt werden. Ohne eine direkte Verbindung zur zugrunde liegenden DB herzustellen, ermöglichen diese APIs externen Systemen eine sichere Interaktion mit den Daten durch Abfragen, Einfügen, Aktualisieren oder Löschen von Daten. Die Definitionen der in Oracle APEX erstellten RESTful Services werden im *Oracle REST Data Services (ORDS)*-Repository gespeichert und als *ORDS-based REST Services* bezeichnet.

### 4.5.1 REST-Architektur-Muster

Wie bereits erwähnt, wird ein Dienst als RESTful bezeichnet, wenn er den Grundsätzen von REST entspricht. Obwohl eine ausführliche Erörterung von REST den Rahmen dieses Kapitels sprengen würde, weist ein RESTful-Dienst die folgenden Merkmale auf (Beachten Sie, dass die allgemeinen Merkmale von REST, wie sie in [2] beschrieben sind, leicht an unseren DB-Kontext angepasst wurden):

1. **DB-Tabellen, die als Ressourcen für Dienste bereitgestellt werden**. Ein RESTful Service wird als ein Satz von Ressourcen modelliert, in unserem Fall auf Ihren DB-Tabellen. Diese Ressourcen werden durch URLs identifiziert und über die `HTTP` oder `HTTPS` Web-Protokolle zugegriffen.
2. **Operationen, die über HTTP-Anfragen gesendet werden**. Ein kleiner Satz von Operationen wird verwendet, um mit diesen Ressourcen in Form einer HTTP-Anfrage umzugehen, die aus den Operationen `POST`, `GET`, `PUT` und `DELETE` besteht und damit den CRUD-Operationen (Create, Read, Update, Delete) ähnelt.
3. **Dienste sind zustandslos**. RESTful-Dienste sind *zustandslos*, d.h. zwischen den Anfragen wird kein Client-Kontext auf dem Server gespeichert - der Sitzungsstatus wird vollständig auf dem Client aufbewahrt. Das bedeutet, dass jede Anfrage vom Client an den Server

alle Informationen enthalten muss, die der Server benötigt, um die Anfrage zu verstehen, und keinen auf dem Server gespeicherten Kontext nutzen kann. Der Hauptvorteil besteht darin, dass wiederverwendbare und skalierbare Dienste erstellt werden können, die verwaltet und aktualisiert werden können, ohne das System zu beeinträchtigen, selbst während der Laufzeit. Der Nachteil besteht darin, dass sich die Netzwerkleistung verringern kann, da mehr wiederkehrende Daten (Overhead pro Interaktion) in einer Reihe von Anfragen gesendet werden, da diese Daten nicht in einem gemeinsamen Kontext auf dem Server gespeichert werden können.

4. **HTTP-Responses**. Anfragen an einen RESTful-Dienst lösen immer eine *Antwort* aus. Diese Antwort liegt in Form von XML, JSON, HTML oder einem anderen definierten Format vor. Antworten enthalten Details zu einer Änderung der zugrunde liegenden Daten, Fehlermeldungen und Hypertext-Links zu anderen verwandten Ressourcen, je nach Vorgang.

### 4.5.2 Aktivieren eines DB-Schemas für den RESTful-Zugriff

Um ein DB-Schema für den RESTful-Zugriff zu aktivieren, muss es zunächst bei ORDS registriert werden. Dies kann ganz einfach geschehen, wie in Abbildung 4.13 dargestellt, indem man auf das Tool für RESTful-Dienste innerhalb des SQL Workshops zugreift (1), dann auf die Schaltfläche "Register Schema with ORDS" klickt (2), einen beliebigen Alias eingibt, der bei der Bildung der URL verwendet wird, die auf einen beliebigen RESTful-Dienst innerhalb des Schemas verweist (3), und schließlich, nachdem man auf die Schaltfläche *"Save Schema Attributes"* geklickt hat (4), das ORDS-Dashboard für RESTful-Dienste erscheint, das für die Verwaltung aller RESTful-Dienste verwendet wird.

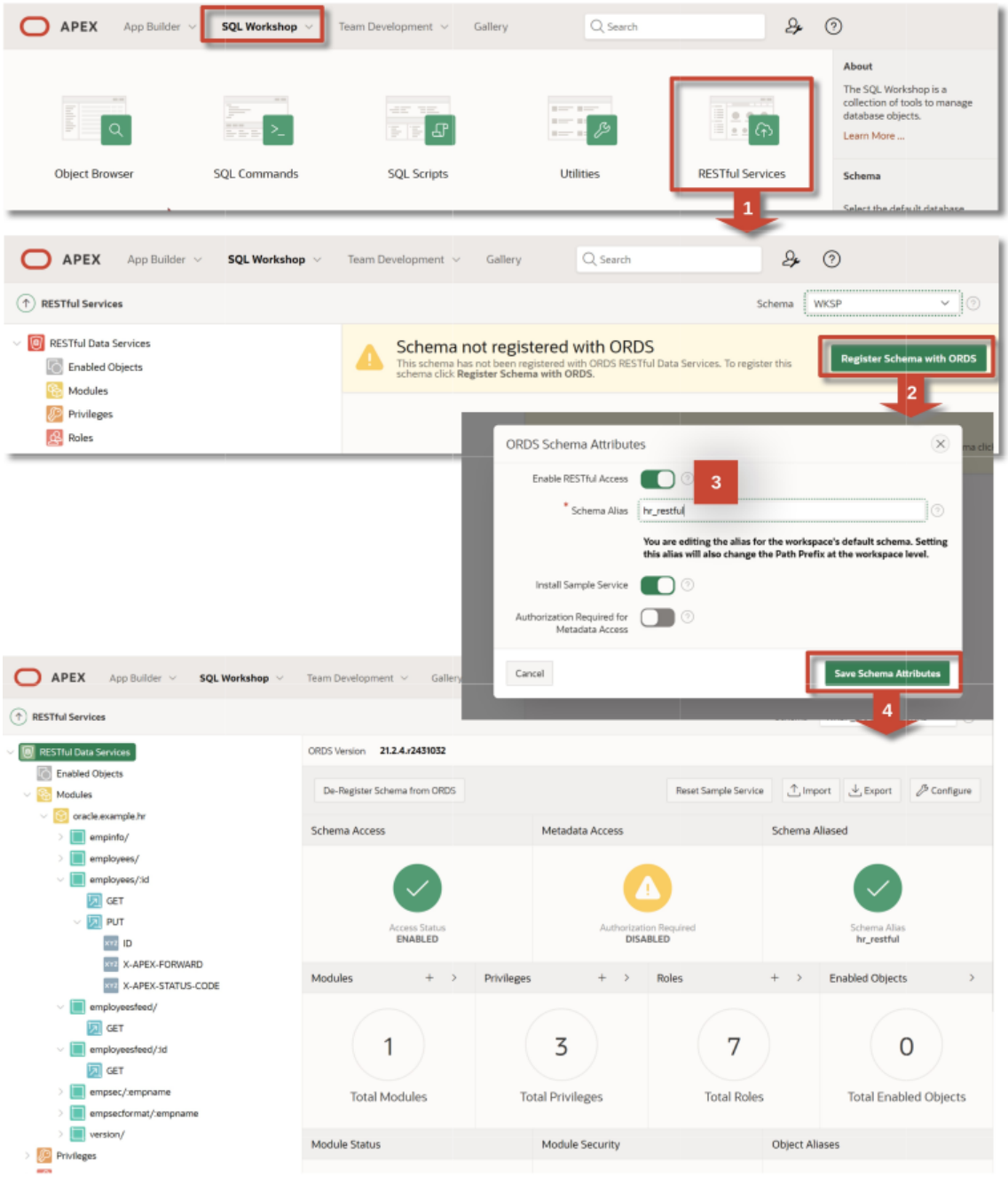



Abbildung 4.13: Aktivieren eines DB-Schemas für den RESTful-Zugriff.

Falls in Schritt (3) die Option “Install Sample Service” aktiviert wurde, wird das Beispiel *Modul* `oracle.example.hr` installiert, das beispielhafte *Ressourcen-Templates* und *Ressourcenhandler* (vgl. Abschnitt 4.5.3) bereitstellt, die mehrere verschiedene Operationen zum Abrufen und Anzeigen von Mitarbeiterinformationen aus der Tabelle “Employees” implementieren, wie in Abbildung 4.15 zu sehen ist.

### 4.5.3 Ressourcenmodule | Templates | Handler

In ORDS können RESTful-Dienste mit Hilfe von drei grundlegenden Komponenten erstellt werden, die Ressourcenmodule, Ressourcenvorlagen und Ressourcenhandler umfassen (siehe Abbildung 4.15 für einige Beispiele). Die allgemeinen Beziehungen zwischen diesen Grundbausteinen werden durch ein ER-Diagramm in Abbildung 4.14 veranschaulicht.




Abbildung 4.14: Beziehungen zwischen den verschiedenen Komponenten der ORDS RESTful Services.

- **Ressourcenmodule.** Ein Ressourcenmodul ist nichts anderes als ein Container, der eine Reihe von zusammenhängenden RESTful-Diensten auf der Grundlage eines bestimmten *REST-fähigen DB-Schemas* zusammenfasst. Das DB-Schema kann mehrere Ressourcenmodule enthalten. Ein Ressourcenmodul bietet nicht nur eine Möglichkeit zur eindeutigen Identifizierung der Gruppe, sondern definiert auch den eindeutigen Basispfad, der für einen URI *zum Zugriff auf die im Modul definierte Gruppe von Diensten* verwendet wird. Ein Ressourcenmodul, mit dem Sie auf Informationen über Mitarbeiter zugreifen können, heißt beispielsweise `oracle.example.hr`, und der Basispfad für den Dienst lautet `/hr/`, siehe Abbildung 4.15.
- **Ressourcen-Templates**. Ein Ressourcen-Template definiert einen einzelnen Dienst, der aufgerufen werden kann, z.B. `employees/:id`, Abbildung 4.15. Ressourcen-Templates sind in Ressourcenmodulen enthalten. Jedes Ressourcen-Template definiert ein *URI-Muster*, über das sie erreicht werden kann, z. B. “empinfo/” (Abbildung 4.15) und enthält mindestens einen Ressourcen-Handler (siehe unten).
- **Ressourcen-Handler**. Jeder Ressourcem-Handler implementiert eine (und nur eine) von vier verschiedenen HTTP-Operationen, die traditionellen CRUD-Operationen (Create, Read, Update, Delete) ähneln .
  - `POST` erstellt eine neue Ressource oder fügt eine Ressource zu einer Sammlung hinzu, was einer SQL `INSERT`-Anweisung entspricht.
  - `GET` ruft eine Darstellung einer Ressource ab, die einer SQL `SELECT`-Anweisung entspricht.
  - `PUT` aktualisiert die Werte einer vorhandenen Ressource, was einer SQL `UPDATE`-Anweisung entspricht.
  - `DELETE` löscht eine vorhandene Ressource, was einer SQL `DELETE`-Anweisung entspricht.

Sie müssen für jeden Vorgang, der mit demselben Ressourcen-Template verbunden ist, einen Ressourcen-Handler definieren. Um beispielsweise *eine Operation zur Rückgabe von Daten* und *eine weitere Operation zur Speicherung von Daten* bereitzustellen, müssen Sie für jede Operation einen Ressourcen-Handler definieren (vgl. Abb. 4.15 - für die Ressourcenvorlage `employees/:id`. Es wurden zwei Operationen `GET` (1) und `PUT` (2) definiert).

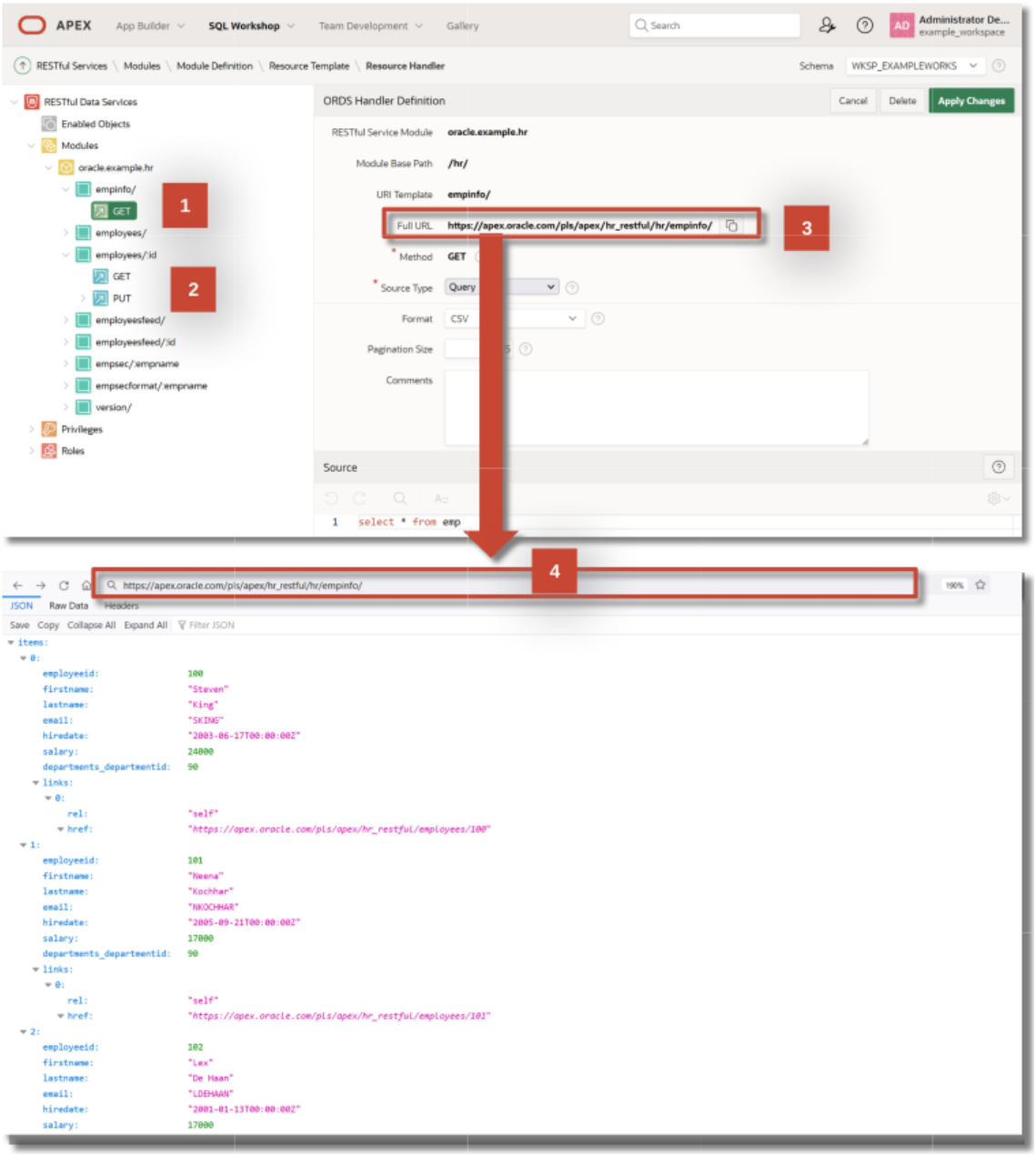



Abbildung 4.15: Beispiel RESTful Services für die Tabelle Employees.

Zusammenfassend gibt Abbildung 4.15 einen beispielhaften Überblick über die verschiedenen Bestandteile eines RESTful Service in Form von Ressourcenmodulen, Templates und Handlern, einschließlich der Auswahl des gewünschten Antwortformats (CSV) und der SQL-Abfrage, die den Operationstyp des Ressourcenhandlers implementiert. Nachdem Sie einen RESTful Service erstellt haben, können Sie ihn testen, indem Sie einfach die generierte Zugriffs-URL in Ihren Browser eingeben (vgl. Abb. 4.15 (3) bzw. (4)).

### 4.5.4 Verwendung von “AutoREST” anstelle der manuellen Definition von Ressourcen

Wie im vorherigen Abschnitt zu sehen war, sind für jede Art von DB-Operation, die externen Clients als RESTful Service für den indirekten Zugriff auf unsere DB angeboten werden soll, mehrere Schritte notwendig. Für die einfachste Form einer Abfrage, d.h. einen vollständigen

Tabellenscan, gibt es jedoch auch eine Art “Abkürzung”, die APEX in Form der sogenannten *“AutoREST”-Facility* bereitstellt. AutoREST ist eine schnelle und einfache Möglichkeit, DB-Tabellen als REST-Ressourcen bereitzustellen. Sie verlieren etwas Flexibilität und Anpassbarkeit, wenn Sie die AutoREST-Funktion verwenden, aber es reduziert Ihre Zeit und Ihren Aufwand in erheblichem Maße. Mit AutoRest können Sie Daten schnell freigeben, aber nur die gesamte Tabelle abfragen (d.h. einen Full-Table-Scan) und ein festes Ausgabeformat nur in Form von JSON haben.

AutoREST kann auf einfache Weise wie folgt verwendet werden: Nachdem Sie Ihr Schema für den RESTful-Zugriff aktiviert haben (vgl. Abschnitt 4.5.2), kann jede DB-Tabelle einzeln als Ressource mit dem Object Browser von SQL Workshop definiert werden, wie in Abbildung 4.16 zu sehen ist.

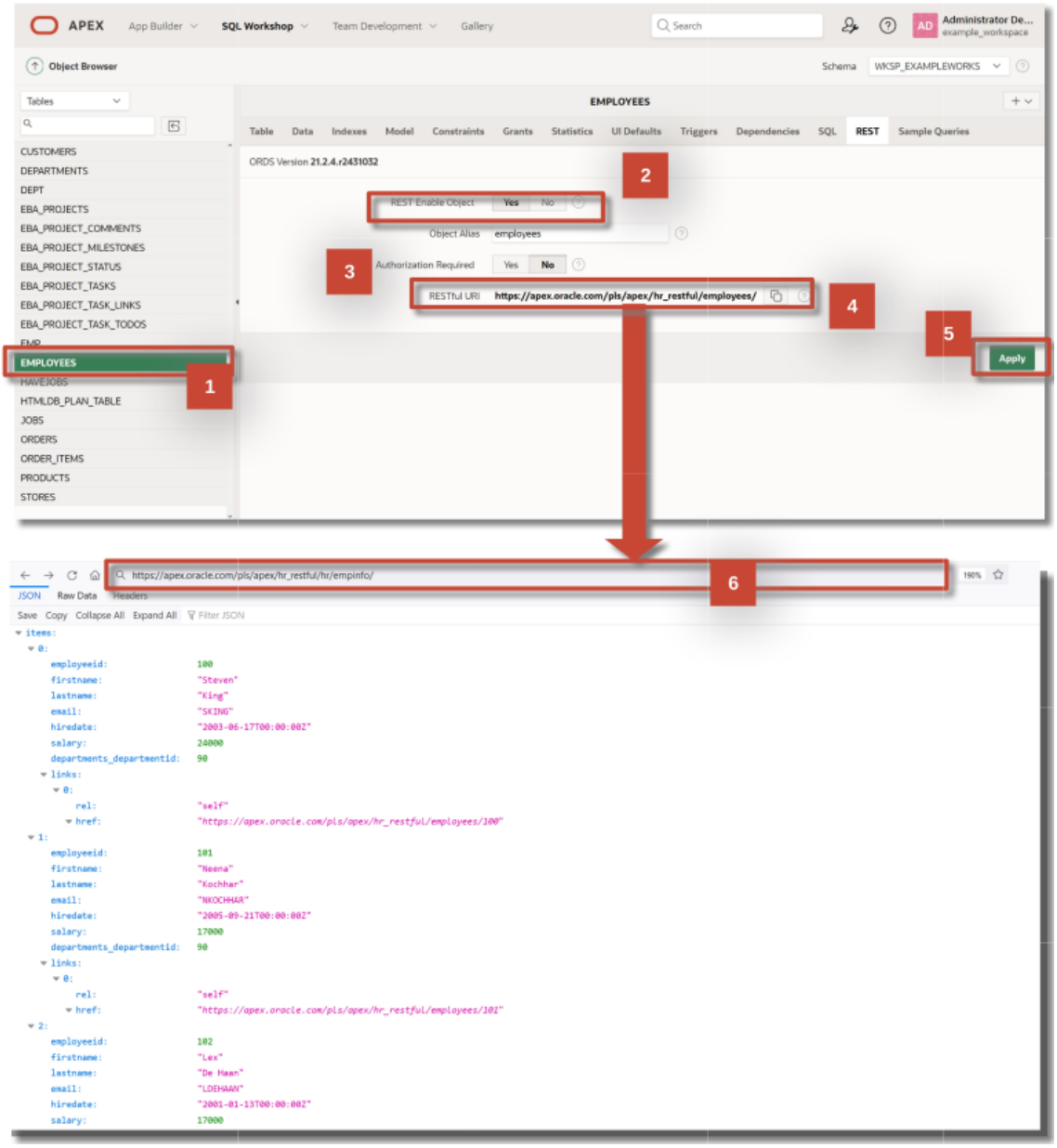



Abbildung 4.16: Aktivieren des REST-Dienstes im Objektbrowser.

Nach Auswahl der entsprechenden DB-Tabelle (1), Klick auf den REST-Reiter (2), Auswahl der gewünschten Berechtigung (3) und Übernahme all dieser Einstellungen (5) ist REST für die Tabelle aktiviert und die automatisch generierte Zugriffs-URL erscheint (4). Auch dieser REST-Dienst

kann einfach getestet werden, indem die URL in einem Webbrowser eingegeben wird (6).



# 5. Wie erstellt man den ersten Entwurf einer Anwendung?

A. ANGEIOPLASTIS, G. MYLLIS, A. TSIMPIRIS, D. VARSAMIS

## 5.1 Warum benötigen Unternehmen Anwendungen?

In der sich ständig wandelnden Geschäftswelt von heute sind Anwendungen zu einem entscheidenden Instrument für Unternehmen geworden, um wettbewerbsfähig zu bleiben und den ständig wechselnden Anforderungen ihrer Kunden gerecht zu werden. Die Entwicklung von Anwendungen ermöglicht es Unternehmen, ihre Abläufe zu optimieren, die Effizienz zu steigern und die Kundenerfahrung zu verbessern, was letztendlich zu einer höheren Rentabilität führen kann.

Einer der Hauptgründe, warum Unternehmen Anwendungen entwickeln, besteht darin, Prozesse zu automatisieren, die derzeit manuell durchgeführt werden. Zum Beispiel könnte ein E-Commerce-Unternehmen eine Anwendung entwickeln, die ihre Lagerverwaltung und Auftragsabwicklung automatisiert. Dadurch können sie die Wahrscheinlichkeit menschlicher Fehler reduzieren und die Geschwindigkeit und Genauigkeit ihrer Abläufe verbessern.

Ein weiterer wichtiger Grund, warum Unternehmen Anwendungen erstellen, ist die Verbesserung des Kundenerlebnisses. Anwendungen können entwickelt werden, um Kunden Selbstbedienungsoptionen anzubieten, wie die Möglichkeit, ihre Bestellungen zu verfolgen oder ihre Kontoinformationen zu aktualisieren. Dies kann dazu beitragen, die Wartezeiten für Kunden zu reduzieren und die Kundenzufriedenheit zu erhöhen.

Darüber hinaus können Anwendungen Unternehmen dabei helfen, wertvolle Daten über ihre Kunden zu sammeln, die zur Verbesserung ihrer Marketingstrategien und zur Entwicklung gezielter Werbekampagnen verwendet werden können. Zum Beispiel kann ein Einzelhändler eine Anwendung entwickeln, die das Einkaufsverhalten der Kunden verfolgt, sodass sie personalisierte Empfehlungen und Sonderangebote an ihre Kunden senden können.

Insgesamt ist die Entwicklung von Anwendungen zu einem starken Geschäftsbedarf geworden, da sie Unternehmen ermöglicht, effizienter zu arbeiten, bessere Kundenerlebnisse zu bieten und Einblicke in das Verhalten ihrer Kunden zu gewinnen. Da die Geschäftswelt weiterhin digitaler wird, wird die Bedeutung der Anwendungsentwicklung nur weiter zunehmen.

## 5.2 Die Einrichtung der ORACLE APEX-Umgebung.

Das Ziel dieses Beispiels besteht darin, Sie mit der ORACLE APEX-Umgebung vertraut zu machen, indem Sie durch die Einrichtung der APEX-Umgebung geführt werden, ein SQL-Workshop-Beispiel der employees-departments-projects-Datenbank vorstellen und zeigen, wie man eine

APEX-Webanwendung basierend auf dem Datenmodell erstellt.

Um dieses Beispiel umzusetzen, werden die folgenden Schritte befolgt:

Schritt 1: Öffnen Sie einen Webbrowser und geben Sie „https://apex.oracle.com/en/“ ein. Klicken Sie auf die Schaltfläche **Sign In**, wie in Abbildung 5.1 gezeigt.

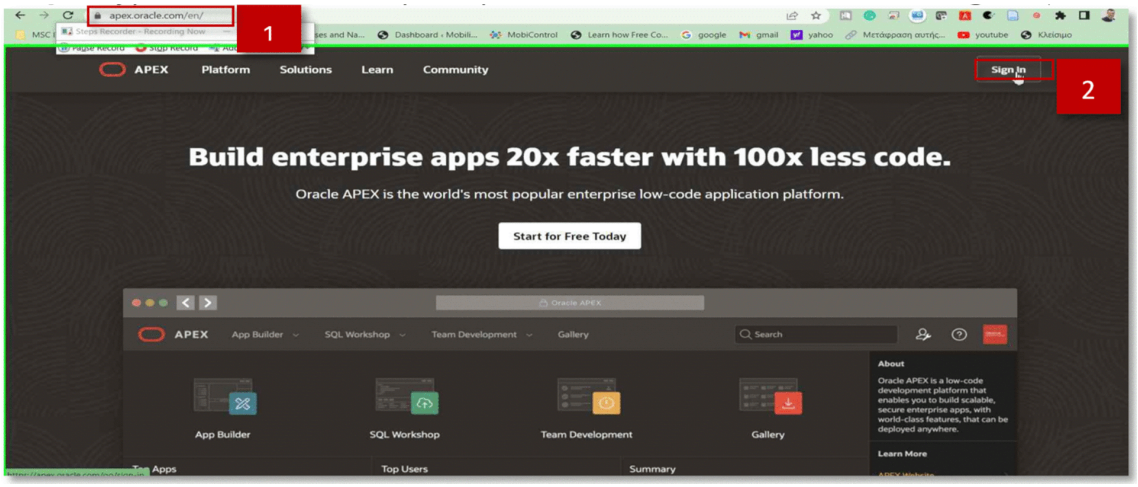



Abbildung 5.1: Geben Sie den APEX-Link ein und melden Sie sich an.

Schritt 2: Klicken Sie auf die Schaltfläche **Request a Workspace**. In dem sich öffnenden Popup-Fenster füllen Sie die erforderlichen Felder zur Identifikation aus und klicken Sie auf **Next**. Anschließend beantworten Sie die Umfragefragen im nächsten Popup-Fenster und klicken Sie auf **Next**, wie in Abbildung 5.2 dargestellt.

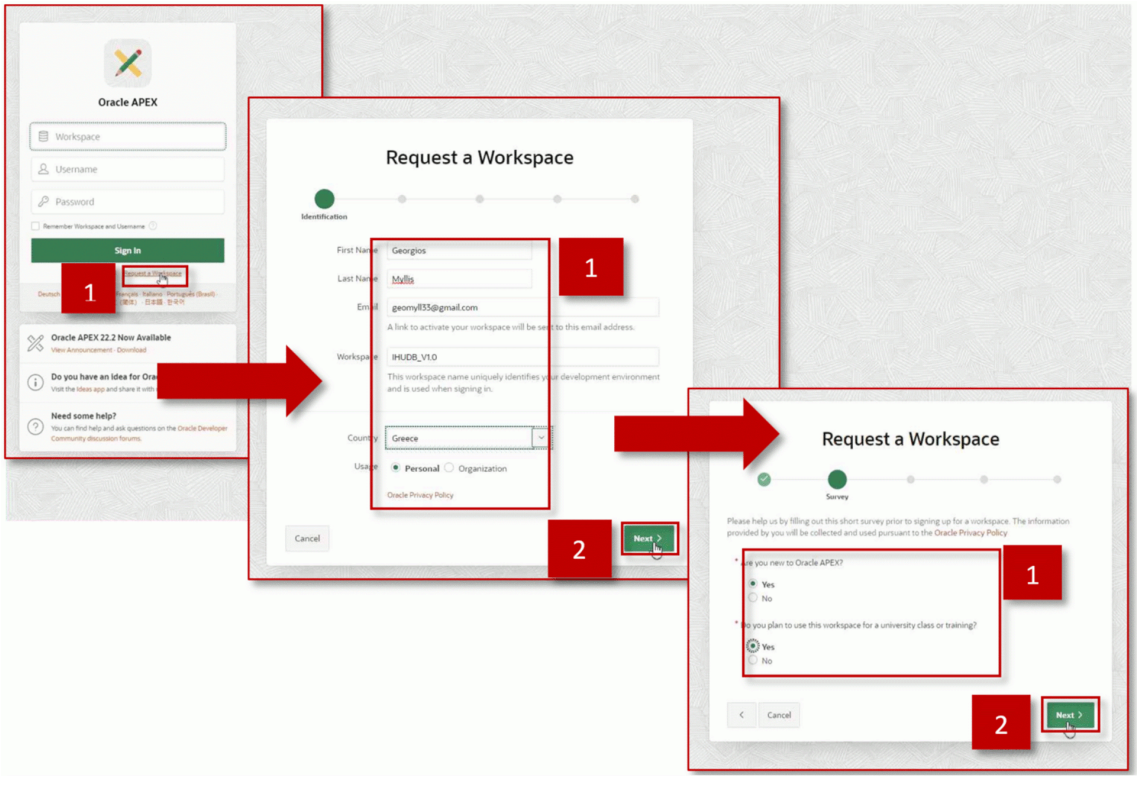



Abbildung 5.2: Anforderung der ersten Schritte für einen Oracle APEX-Arbeitsbereich.

Schritt 3: Beantworten Sie die angeforderten Informationen zur Begründung und klicken Sie auf **Next**. In dem anschließenden Popup-Fenster lesen Sie die Nutzungsbedingungen und akzeptieren Sie diese, indem Sie auf **Next** klicken. In dem folgenden Popup-Fenster überprüfen Sie die von Ihnen eingegebenen Details auf Fehler und klicken dann auf **Submit request**, wie in Abbildung 5.3 gezeigt.

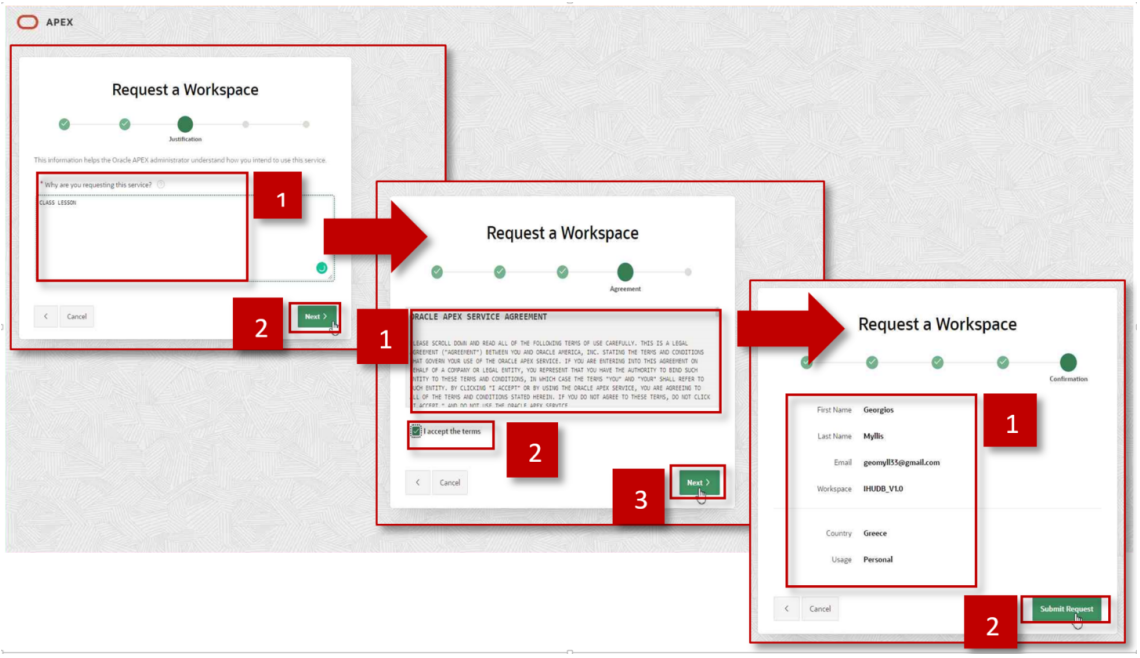



Abbildung 5.3: Fertigstellung der Anfrage eines Oracle APEX-Workspace.

Schritt 4: Nachdem Sie die Anfrage abgeschickt haben, wird ein Popup-Fenster erscheinen und bestätigen, dass Ihr Arbeitsbereich angefordert wurde. Eine Aktivierungs-E-Mail wird an die von Ihnen angegebene E-Mail-Adresse gesendet, wie in Abbildung 5.4 dargestellt.

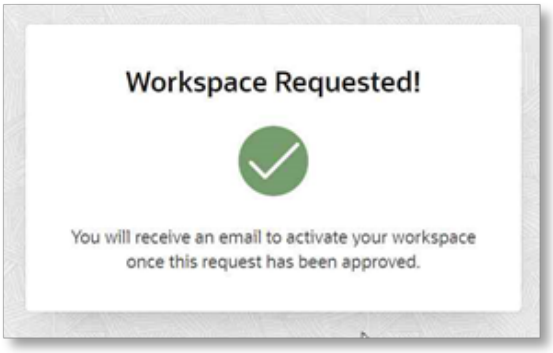



Abbildung 5.4: Oracle APEX Workspace genehmigt.

Schritt 5: Prüfen Sie Ihre E-Mail auf eine Nachricht vom APEX. Bevor Sie fortfahren können, muss Ihr Antrag genehmigt werden. Nach der Genehmigung klicken Sie auf **Create Workspace**, wie in Abbildung 5.5 gezeigt.

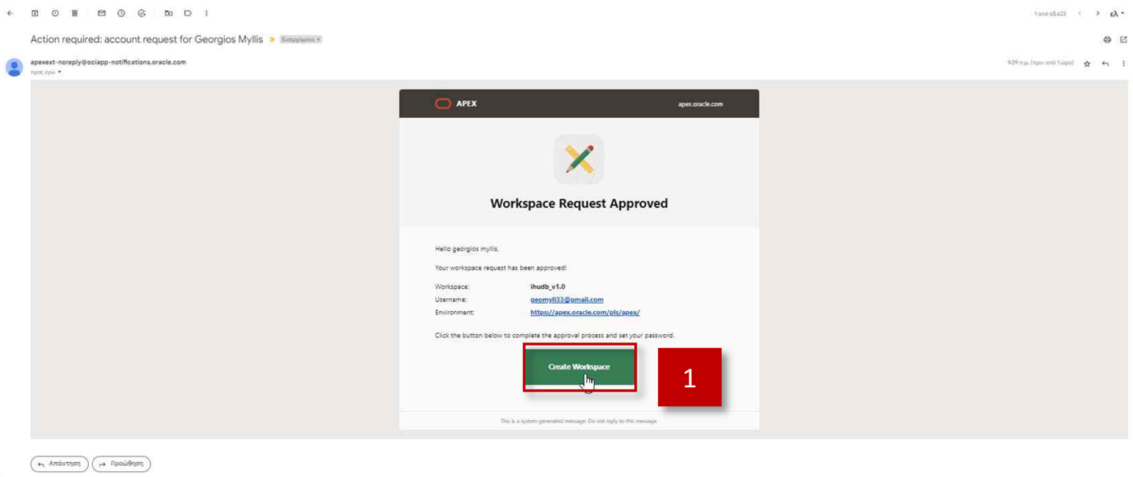



Abbildung 5.5: Aktivierungs-E-Mail von APEX.

Schritt 6: Warten Sie, bis der Prozess der Arbeitsbereichserstellung abgeschlossen ist. Sobald der Vorgang abgeschlossen ist, werden Sie in einem Popup-Fenster darüber informiert, dass der Arbeitsbereich erfolgreich erstellt wurde. Klicken Sie auf **Continue to Sign-in Screen**. Es erscheint ein neues Pop-up-Fenster, in dem Sie aufgefordert werden, Ihr Passwort zu ändern, bevor Sie sich anmelden. Geben Sie ein neues Kennwort ein, bestätigen Sie es und klicken Sie auf **Change Password**, wie in Abbildung 5.6 gezeigt.

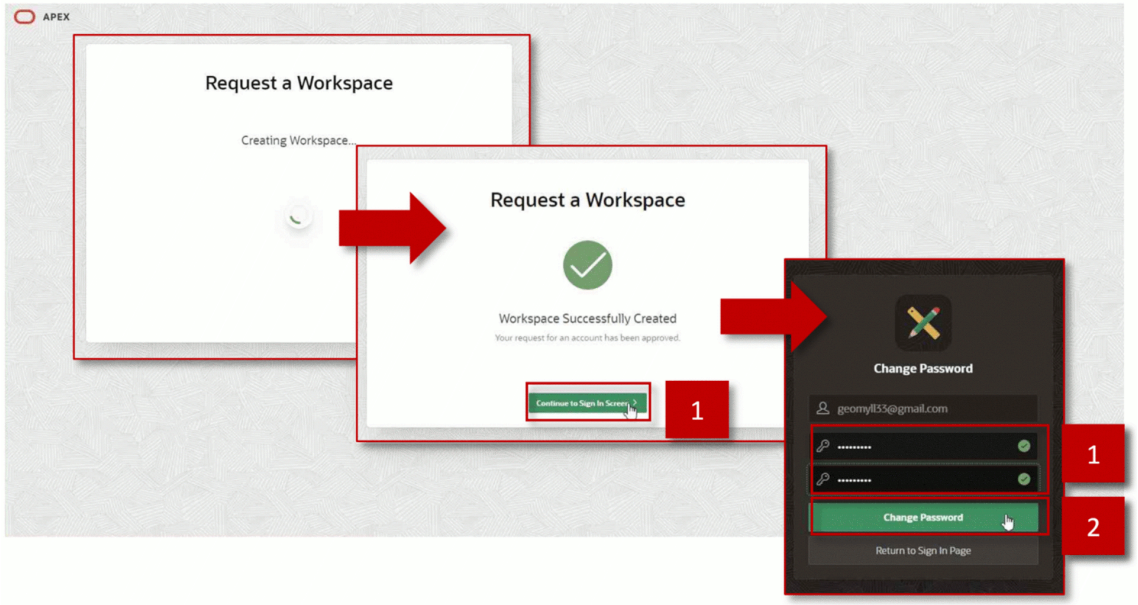



Abbildung 5.6: Richten Sie ein neues Passwort für Workspace APEX ein.

Schritt 7: Die Oracle APEX-Umgebung ist nun einsatzbereit, wie in Abbildung 5.7 dargestellt.

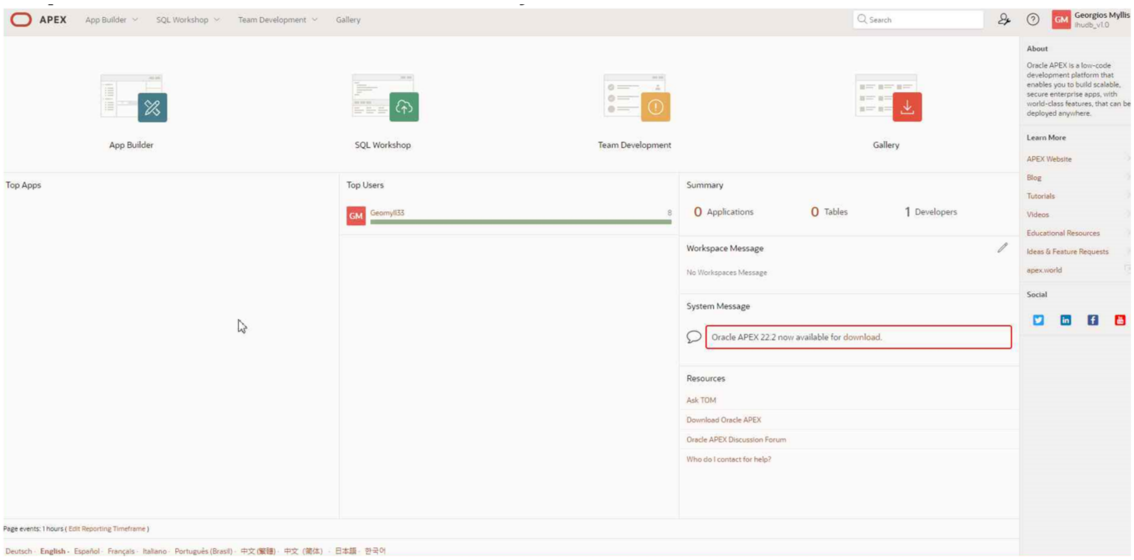



Abbildung 5.7: Arbeitsbereich APEX-Umgebung.

## 5.3 Entwicklung von SQL-Workshop-Beispielen

Um eine Datenbank in der APEX-Umgebung zu erstellen, gibt es mehrere Möglichkeiten. Eine Möglichkeit ist die Verwendung des Object Browsers, eines grafischen Werkzeugs, mit dem der Benutzer Tabellen und ihre jeweiligen Felder durch Ausfüllen einer vorgefertigten Vorlage erstellen kann.

Eine andere Alternative ist die Verwendung von SQL-Befehlen, mit denen der Benutzer Tabellen und andere Datenbankobjekte durch die Ausführung von SQL-Anweisungen erstellen kann. Diese Methode bietet mehr Flexibilität und Kontrolle über die Datenbankstruktur, erfordert jedoch fortgeschrittenere Kenntnisse.

Quick SQL hingegen ist ein von Oracle APEX bereitgestelltes Tool, das die schnelle Erstellung von SQL-Skripten zur Erstellung von Datenbanktabellen, Constraints und Beispieldaten ermöglicht. Quick SQL vereinfacht den Prozess der Erstellung von Datenbankobjekten, ohne dass Sie manuell komplizierte SQL-Skripte schreiben müssen.

Im vorliegenden Beispiel verwenden wir die Quick-SQL-Methode. Das Skript wurde auf der Grundlage des in Abbildung 5.8 dargestellten logischen Modells erstellt.

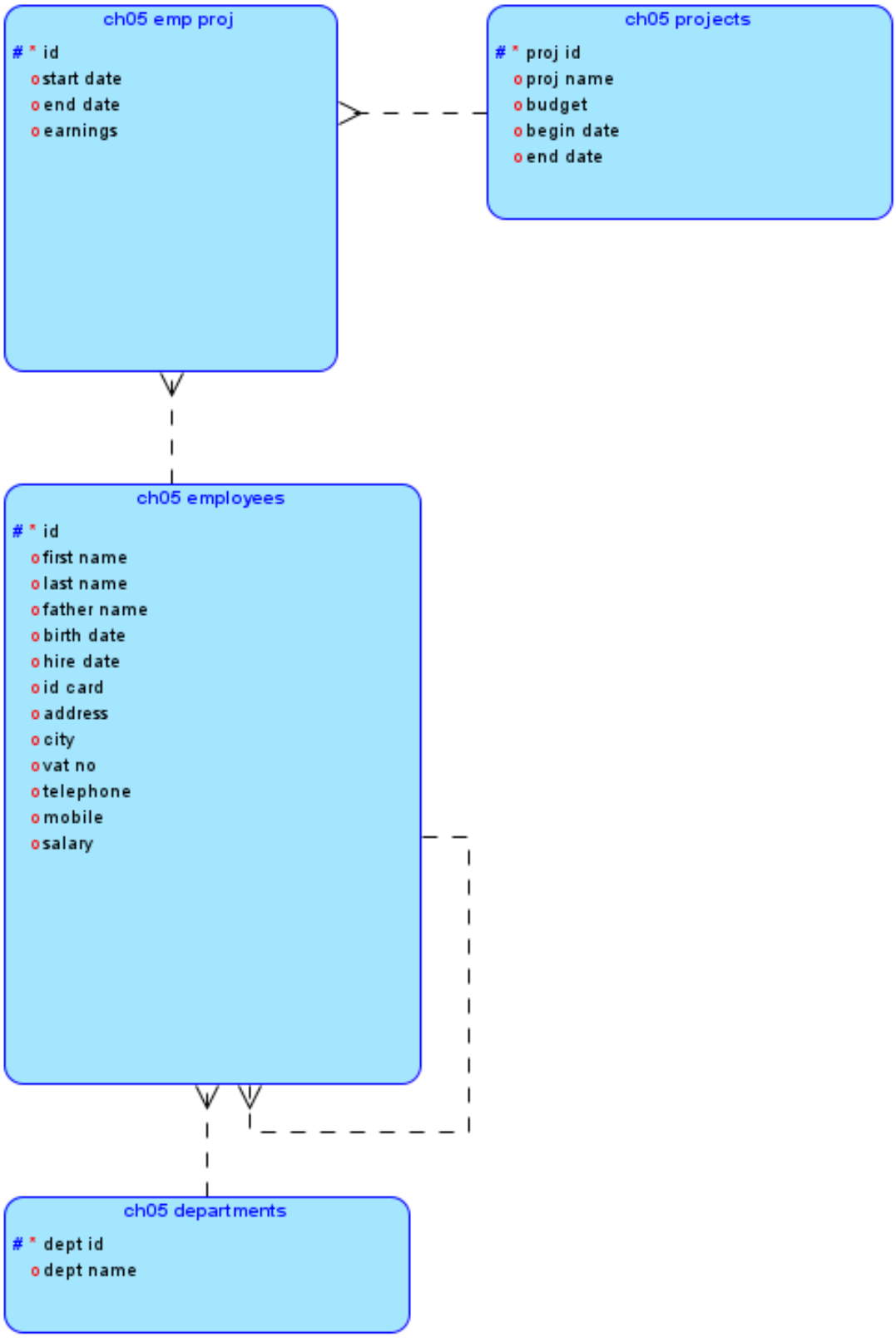



Abbildung 5.8: Logisches Modell für unser laufendes HR-Beispiel.

Anhand des relationalen Modells, das mit dem Oracle Data Modeler (ODM) aus dem logischen

Modell abgeleitet wurde (siehe Abbildung 5.9), können wir mit der Implementierung der folgenden Struktur in der Datenbank fortfahren.

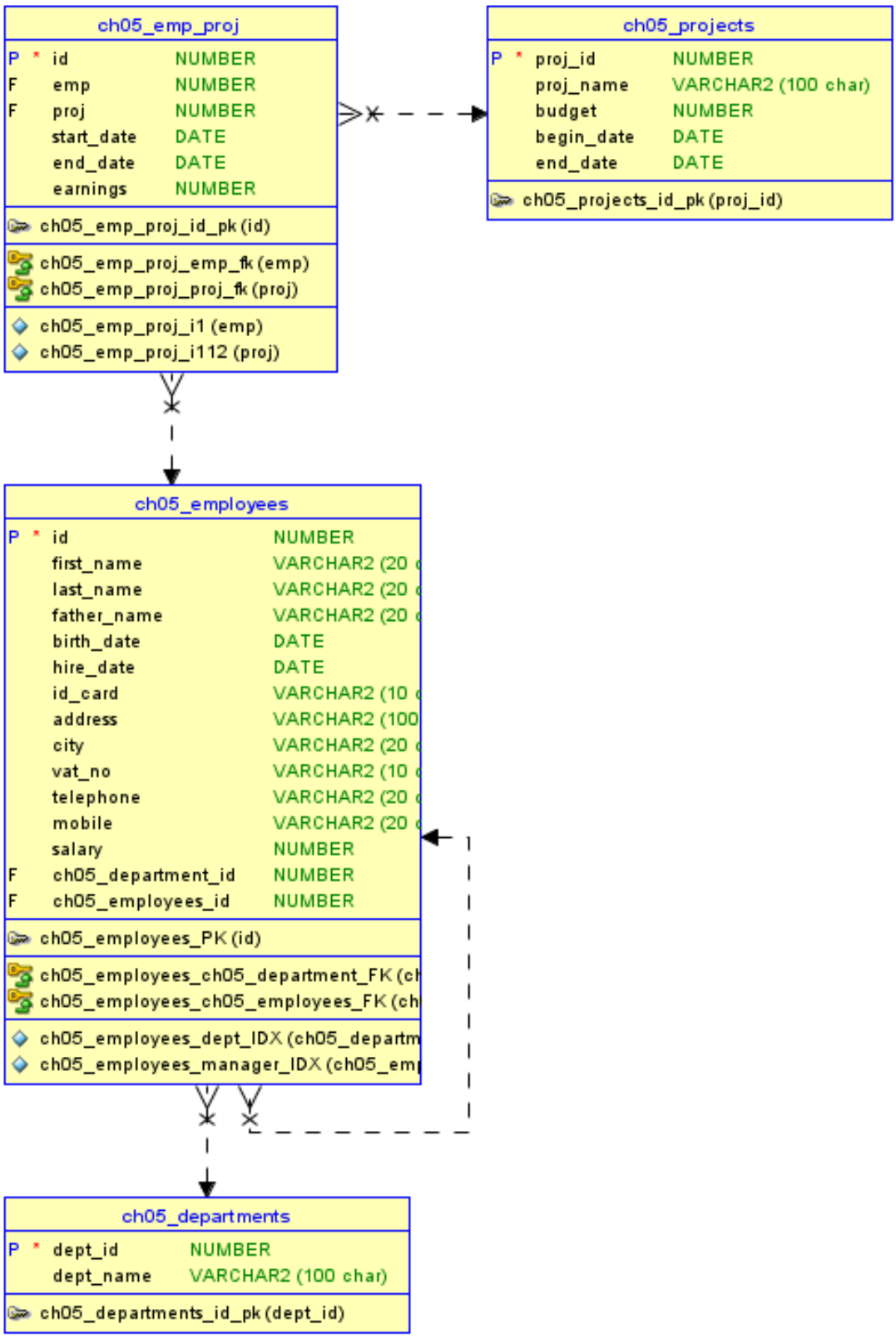



Abbildung 5.9: Relationales Modell für unser laufendes HR-Beispiel.

### 5.3.1 Quick SQL

Der bereitgestellte Code scheint eine Reihe von Anweisungen für die Erzeugung von Beispieldaten mit dem Quick SQL-Tool zu sein. Jeder Abschnitt beginnt mit einem Tabellennamen, gefolgt vom Schlüsselwort „/insert“ und einer Zahl, die die gewünschte Anzahl von Zeilen angibt, die in die Tabelle eingefügt werden sollen.

```
ch05_departments /insert 5
    dept_name vc100

ch05_projects /insert 15
    proj_name vc100
    budget num
    begin_date date
    end_date date

ch05_employees  /insert 50
    first_name vc20
    last_name  vc20
    father_name vc20
    birth_date date
    hire_date  date
    id_card    vc10
    address    vc100
    city       vc20
    vat_no     vc10
    telephone  vc20
    mobile     vc20
    salary     num
    dept num /fk ch05_departments
    manager num /fk ch05_employees

ch05_emp_proj /insert 500
    emp num /fk ch05_employees
    proj num /fk ch05_projects
    start_date date
    end_date date
    earnings num
```

Diese Anleitung enthält ein Schema und einen Beispielplan zur Datengenerierung für eine Datenbank, die sich auf Abteilungen, Projekte, Mitarbeiter und deren Zuordnungen bezieht.

Schritt 1: Führen Sie die folgenden Schritte aus, wie in Abbildung 5.10 dargestellt:

1. Navigieren Sie zum Abschnitt **SQL Workshop**.
2. Wählen Sie **SQL Scripts** in den Optionen.
3. Klicken Sie auf **Quick SQL**.
4. Fügen Sie den mitgelieferten Quick SQL-Code in den dafür vorgesehenen Bereich ein.
5. Klicken Sie auf **Generate SQL**. Das generierte SQL wird im Bereich Oracle SQL Output angezeigt.
6. Klicken Sie auf **Save SQL Script**.
7. Geben Sie einen Skriptnamen ein.
8. Klicken Sie auf **Save script**.
9. Klicken Sie auf **Review and Run**, um fortzufahren.

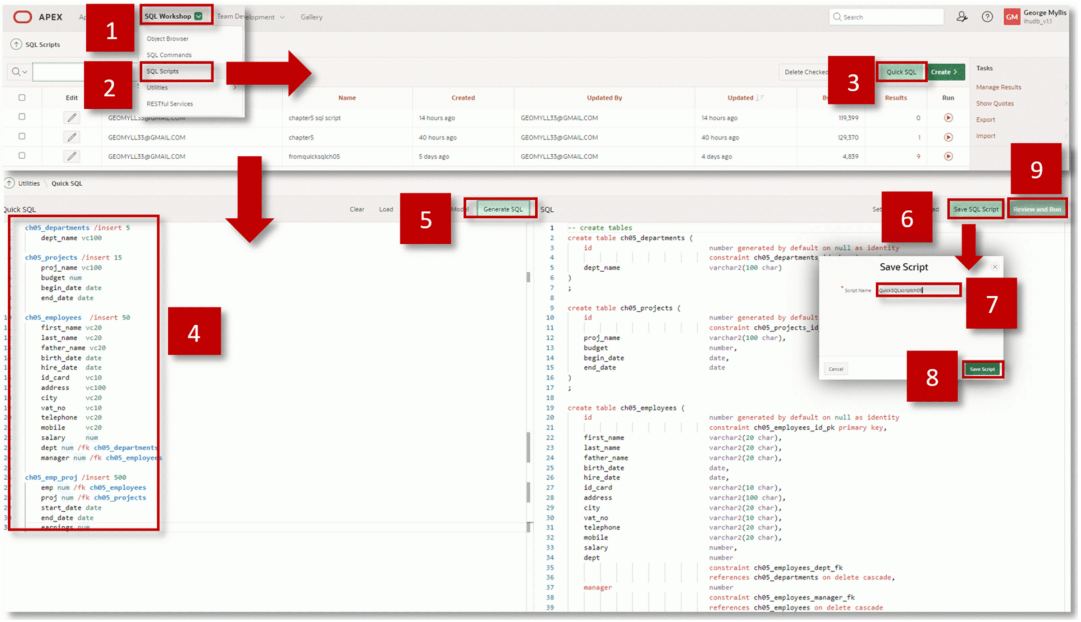



Abbildung 5.10: Quick SQL-Code in den APEX-Arbeitsbereich einfügen.

### 5.3.2 SQL Skript

Wie bereits erwähnt, wird im Folgenden ein SQL-Skript aus Quick SQL generiert, mit dem ein Datenbankschema für ein Beispiel-Personalverwaltungssystem erstellt wird. Das Schema umfasst Tabellen für departments, employees, projects und employee-project Beziehungen. Außerdem enthält es Indizes und Constraints, um die Datenkonsistenz und die referenzielle Integrität zu gewährleisten.

```
-- create tables
create table ch05_departments (
    id                             number generated by default on null as identity
                                   constraint ch05_departments_id_pk primary key,
    dept_name                      varchar2(100 char)
)
;

create table ch05_projects (
    id                             number generated by default on null as identity
                                   constraint ch05_projects_id_pk primary key,
    proj_name                      varchar2(100 char),
    budget                         number,
    begin_date                     date,
    end_date                       date
)
;

create table ch05_employees (
    id                             number generated by default on null as identity
                                   constraint ch05_employees_id_pk primary key,
    first_name                     varchar2(20 char),
    last_name                      varchar2(20 char),
```

```
    father_name                 varchar2(20 char),
    birth_date                  date,
    hire_date                   date,
    id_card                     varchar2(10 char),
    address                     varchar2(100 char),
    city                        varchar2(20 char),
    vat_no                      varchar2(10 char),
    telephone                   varchar2(20 char),
    mobile                      varchar2(20 char),
    salary                      number,
    dept                        number
                                constraint ch05_employees_dept_fk
                                references ch05_departments on delete cascade,
    manager                     number
                                constraint ch05_employees_manager_fk
                                references ch05_employees on delete cascade
)
;

-- table index
create index ch05_employees_i1 on ch05_employees (dept);
create index ch05_employees_i62 on ch05_employees (manager);

create table ch05_emp_proj (
    id                          number generated by default on null as identity
                                constraint ch05_emp_proj_id_pk primary key,
    emp                         number
                                constraint ch05_emp_proj_emp_fk
                                references ch05_employees on delete cascade,
    proj                        number
                                constraint ch05_emp_proj_proj_fk
                                references ch05_projects on delete cascade,
    start_date                  date,
    end_date                    date,
    earnings                    number
)
;

-- table index
create index ch05_emp_proj_i1 on ch05_emp_proj (emp);
create index ch05_emp_proj_i112 on ch05_emp_proj (proj);
```

Die Tabelle ch05_departments speichert Informationen über Abteilungen, einschließlich einer eindeutigen Kennung (id) und eines Namens (dept_name). Die Tabelle ch05_employees speichert Informationen über Mitarbeiter, einschließlich einer eindeutigen Kennung (id), Vorname, Nachname, Vatersname, Geburtsdatum, Einstellungsdatum, ID-Kartennummer, Adresse, Stadt, Umsatzsteuer-Identifikationsnummer, Telefonnummer, Handynummer, Gehalt sowie Abteilungs- (dept) und Manager-IDs (manager). Die Tabelle ch05_projects speichert Informationen über Projekte, einschließlich einer eindeutigen Kennung (id), Name, Budget, Start- und Enddatum. Die Tabelle ch05_emp_proj speichert Informationen über Projektbeziehungen von Mitarbeitern, einschließlich

einer eindeutigen Kennung (id), Mitarbeiter- (emp) und Projekt- (proj) IDs, Startdatum, Enddatum und Verdienst.

In der Tabelle ch05_emp_proj werden Indizes für die Spalten employee und project erstellt, um die Leistung bei der Abfrage der Tabelle anhand dieser Spalten zu verbessern. Aus demselben Grund werden auch für die Spalten department und manager Indizes für die Tabelle ch05_employee erstellt.

Schließlich werden Constraints hinzugefügt, um die Datenkonsistenz und die referenzielle Integrität zwischen den Tabellen zu gewährleisten. Die Constraints ch05_emp_proj_emp_fk und ch05_emp_proj_proj_fk stellen sicher, dass die Spalten employee und project in der Tabelle ch05_emp_proj auf gültige ID-Werte in den Tabellen ch05_employees bzw. ch05_project verweisen.

Die Constraints ch05_employees_dept_fk und ch05_employee_manager_fk stellen sicher, dass die Spalten departure und manager in der Tabelle ch05_employee auf gültige ID-Werte in den Tabellen ch05_departments bzw. ch05_employees verweisen.

Darüber hinaus legt die ON DELETE CASCADE-Klausel fest, dass, wenn eine Zeile in der referenzierten Tabelle gelöscht wird, alle zugehörigen Zeilen in der referenzierenden Tabelle ebenfalls gelöscht werden, um die referentielle Integrität zu wahren.

Schritt 2: Um die gewünschte Aktion auszulösen, sehen Sie sich bitte die Abbildung 5.11 an und klicken Sie auf die Schaltfläche **Run Now**.

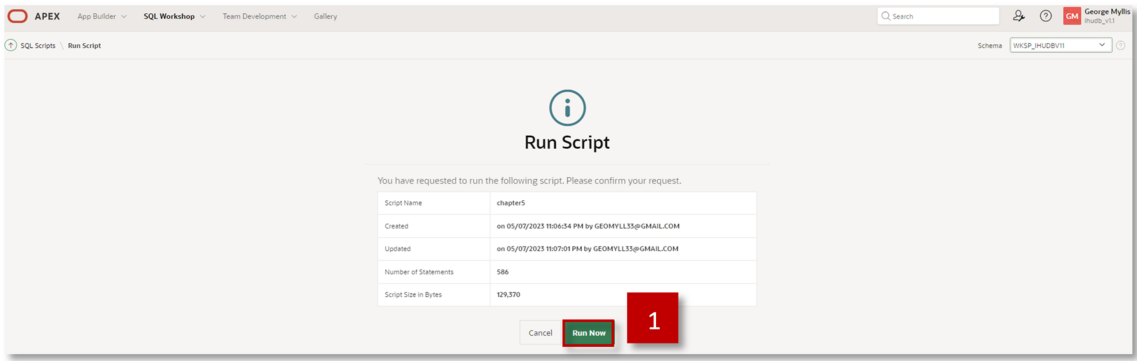



Abbildung 5.11: SQL-Skript ausführen.

Schritt 3: Wenn alle Anweisungen erfolgreich und ohne Fehler verarbeitet wurden, gehen Sie bitte zu Abbildung 5.12 und klicken Sie auf die Schaltfläche **Create App**.

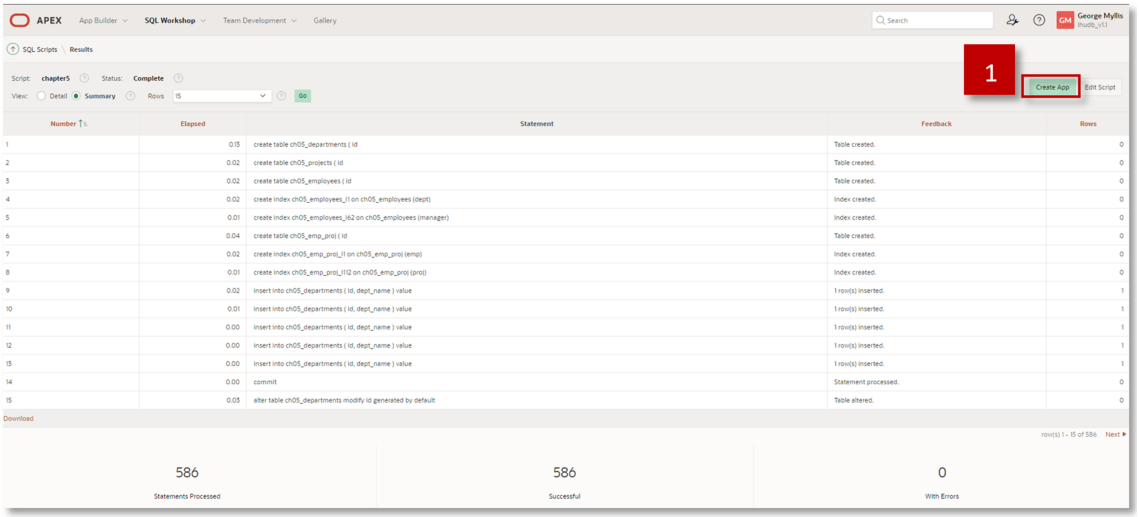



Abbildung 5.12: App-Startprozess erstellen.

## 5.4 Datengesteuerter Teil der Anwendung

Der datengesteuerte Teil der Anwendung konzentriert sich auf die Nutzung von Daten, um Erkenntnisse, Entscheidungsfindung und Benutzerinteraktionen zu fördern. In diesem Bereich stehen die Daten im Mittelpunkt und ermöglichen es den Nutzern, Informationen auf dynamische und intuitive Weise zu erkunden, zu analysieren und mit ihnen zu interagieren.

Das Hauptziel dieses Abschnitts ist es, den Nutzern einen umfassenden Überblick über die Daten zu verschaffen, damit sie wertvolle Erkenntnisse gewinnen und fundierte Entscheidungen treffen können. Durch verschiedene Datenvisualisierungstechniken, wie Diagramme, Grafiken und interaktive Dashboards, können die Nutzer komplexe Datensätze leicht verstehen und Muster, Trends und Ausreißer erkennen.

Darüber hinaus enthält der datengesteuerte Bereich robuste Datenverwaltungsfunktionen, die es den Benutzern ermöglichen, nach bestimmten Datenpunkten oder -segmenten zu filtern, zu sortieren und zu suchen. Erweiterte Funktionen wie Datenaggregation, Drill-Down und Datenexport verbessern die Benutzererfahrung weiter und ermöglichen eine tiefere Erkundung und Analyse.

In diesem Abschnitt wird auch die Bedeutung von Datenintegrität und Datenqualität hervorgehoben. Er umfasst Mechanismen zur Datenvalidierung, Fehlerbehandlung und Datenbereinigung, um sicherzustellen, dass die dargestellten Informationen genau, zuverlässig und konsistent sind.

Mit dem datengesteuerten Bereich können Benutzer Daten aus verschiedenen Blickwinkeln untersuchen, Ansichten nach ihren Wünschen anpassen und die Macht der Daten nutzen, um aussagekräftige Erkenntnisse zu gewinnen. Er dient als leistungsfähiges Werkzeug für die Datenanalyse, Berichterstattung und Entscheidungsunterstützung innerhalb der Anwendung und ermöglicht es den Benutzern, datengestützte Entscheidungen zu treffen und erfolgreiche Ergebnisse zu erzielen.

Schritt 1: Wenn das Popup-Fenster mit der Bezeichnung **AnCreate App from the Script** gemäß Abbildung 5.13 angezeigt wird, überprüfen Sie, ob die Tabellen erstellt wurden und in Ihrer neuen Anwendung verwendet werden sollen, und klicken Sie auf die Schaltfläche **Create Application**.

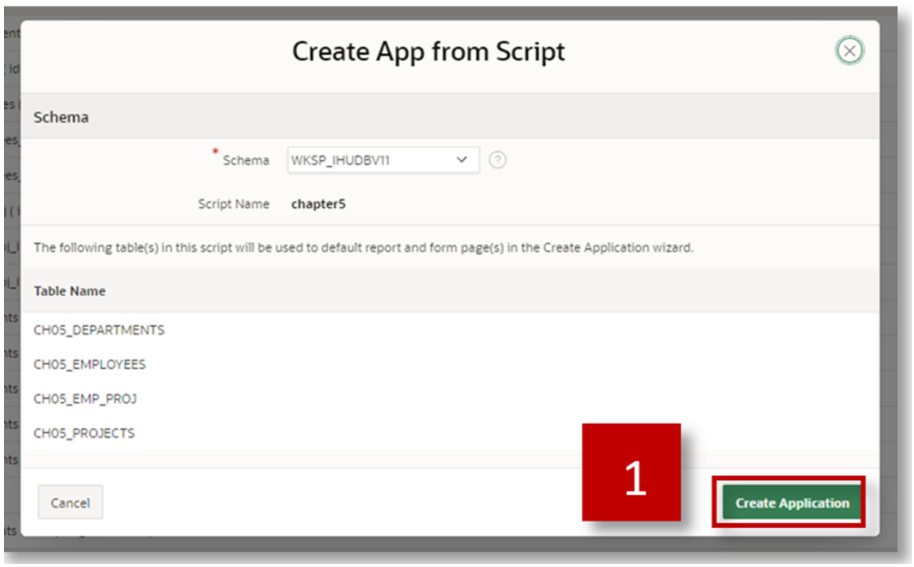



Abbildung 5.13: Anwendung aus Skript erstellen.

Schritt 2: Geben Sie einen Namen für Ihre Anwendung ein, überprüfen Sie, ob Berichte und Formularseiten aus den Tabellen hinzugefügt wurden und in Ihrer neuen Anwendung verwendet werden, und klicken Sie dann auf die Schaltfläche **Create Application**, wie in Abbildung 5.14 gezeigt.

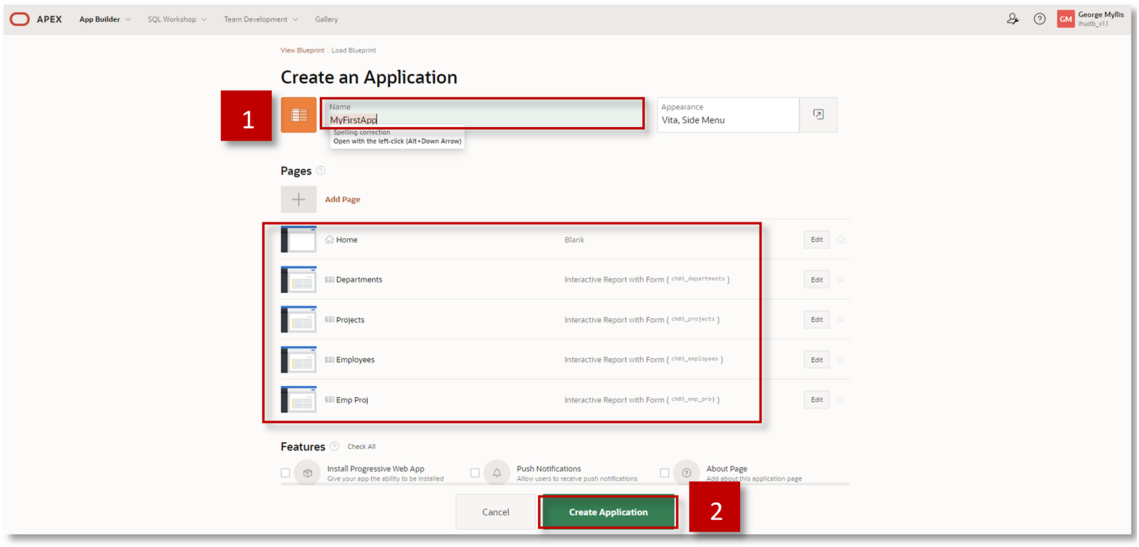



Abbildung 5.14: Webanwendung erstellt.

Schritt 3: Die App-Builder-Umgebung wurde für Ihre Anwendung erstellt, wie in Abbildung 5.15 dargestellt. Es ist wichtig zu wissen, dass Sie innerhalb dieser Umgebung die Möglichkeit haben, Ihre App umfassend anzupassen. Sie können Formularfelder hinzufügen, entfernen oder ändern, das Layout anpassen, Überprüfungsregeln definieren und zahlreiche andere Aktionen durchführen, um die Anwendung an Ihre speziellen Bedürfnisse anzupassen.

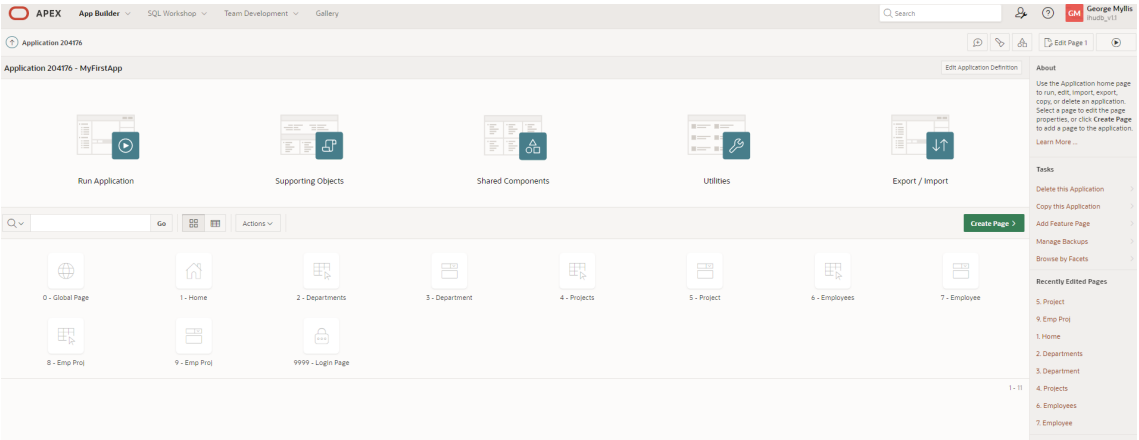



Abbildung 5.15: App-Builder-Umgebung.

Schritt 4: Klicken Sie auf das Symbol **Run Application** in der App-Builder-Umgebung, um die Anwendung auszuführen, wie in Abbildung 5.16 gezeigt. Dadurch wird eine neue Browserseite mit einem speziellen Link von APEX für Ihre Anwendung geöffnet. Geben Sie Ihre Anmeldedaten ein, die mit denen des WORKSPACE übereinstimmen, und klicken Sie dann auf **Sign in**. Sie werden dann auf die Startseite Ihrer App weitergeleitet.

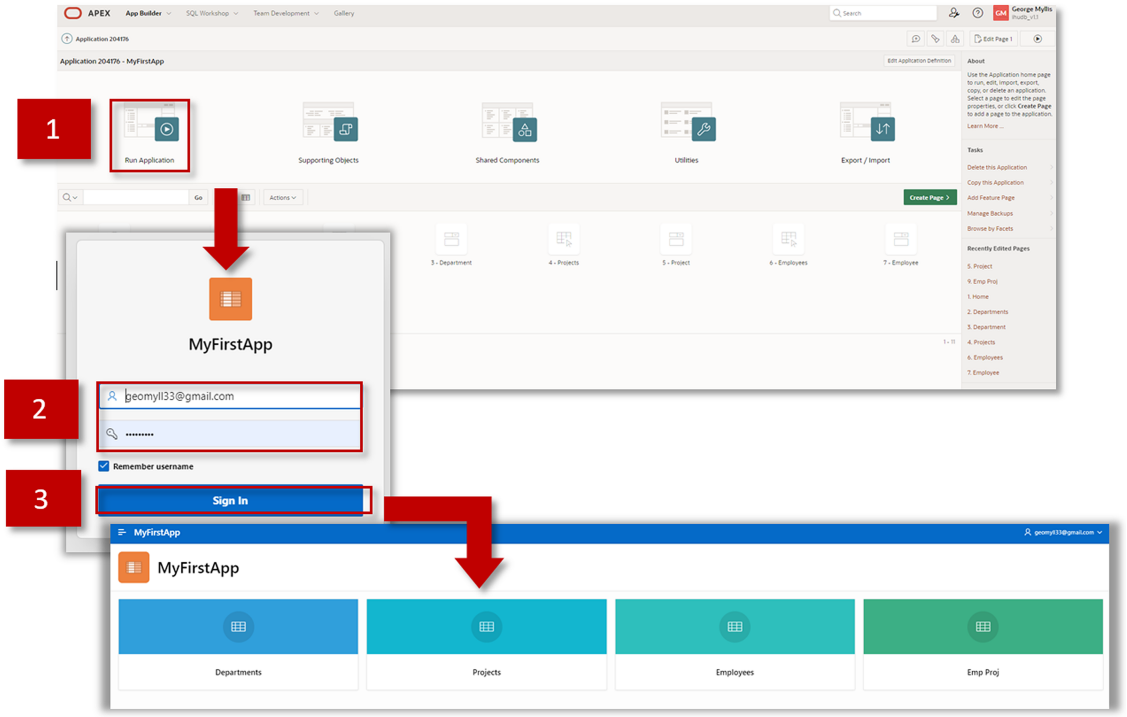



Abbildung 5.16: App-Anmeldeseite.

Schritt 5: Über das Seitenmenü können Sie einfach durch Ihre App navigieren, wie in Abbildung 5.17 dargestellt. Die App bietet eine benutzerfreundliche Umgebung, in der Sie Ihre Daten nach Ihren spezifischen Bedürfnissen und Vorlieben nahtlos hinzufügen, ändern und präsentieren können.

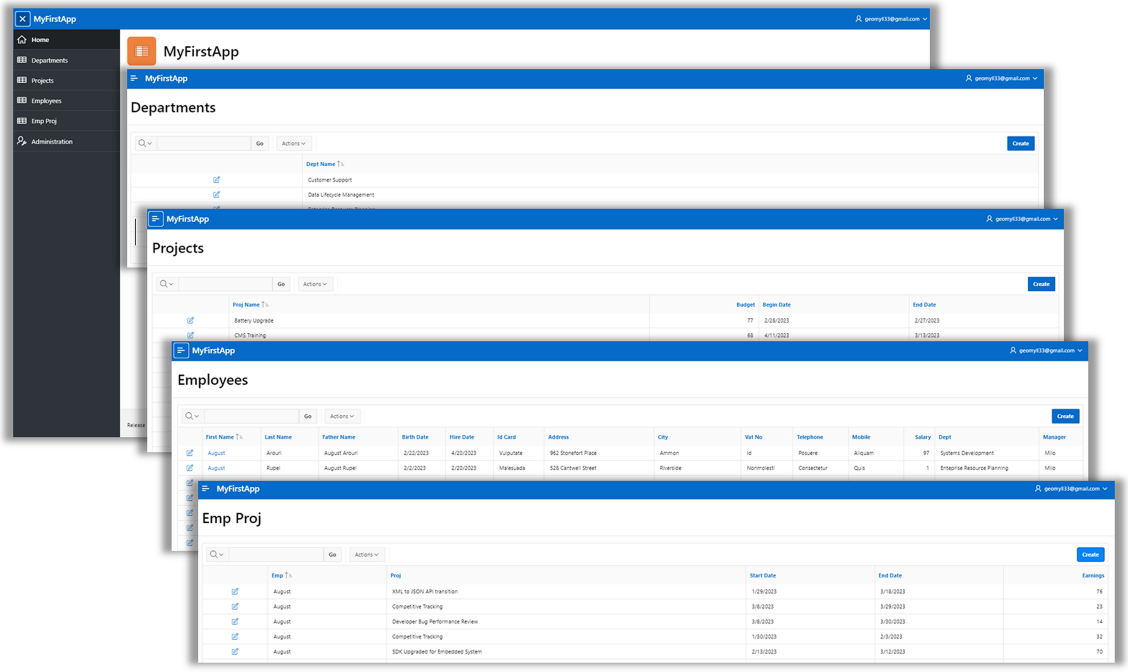



Abbildung 5.17: Ihre neue App-Umgebung.

## 5.5 Verwaltung der Anwendung

Bei der Application Accounts Authentication handelt es sich um eine integrierte Authentifizierungsmethode in Oracle Application Express, die die Erstellung und Verwaltung von Benutzerkonten im Application Express User Repository ermöglicht. Diese Methode ist besonders nützlich, wenn Ihre Anwendung die Application Express Accounts-Authentifizierung verwendet.

Klicken Sie in der Arbeitsbereichumgebung der Anwendung auf **Administration** und wählen Sie **Manage Users and Groups**. Das ursprünglich erstellte Konto kann als Administrator für die integrierte Anwendung verwendet werden. Oracle APEX User verwaltet Konten für Authentifizierungsschemata, die sich anhand des in diesen Konten gespeicherten Benutzernamens und Kennworts authentifizieren.

Arbeitsbereichsadministratoren haben die Berechtigung, Anwendungen und Datenbankobjekte zu erstellen und zu ändern sowie Benutzerkonten, Gruppen und Entwicklungsdienste zu verwalten. Entwickler können Anwendungen und Datenbankobjekte erstellen und ändern, während Endbenutzer keine Entwicklungsberechtigungen haben und nur auf Anwendungen zugreifen können, die kein externes Authentifizierungsschema verwenden, wie in Abbildung 5.18 dargestellt.

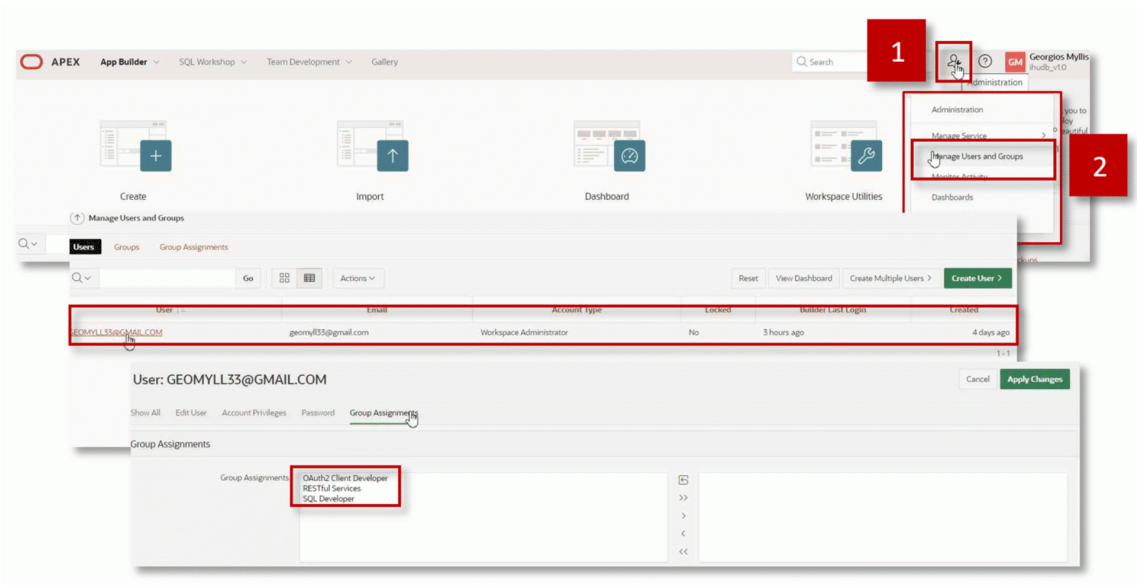



Abbildung 5.18: Verwaltung der Benutzer- und Gruppenumgebung.

## 5.6 Zugriffskontrolle

Als Nächstes werden wir, wie in Abbildung 5.19 gezeigt, ein Beispiel für die Zugriffskontrolle für Benutzer, die die Anwendung verwenden, nach der in der folgenden Abbildung dargestellten Logik erstellen.

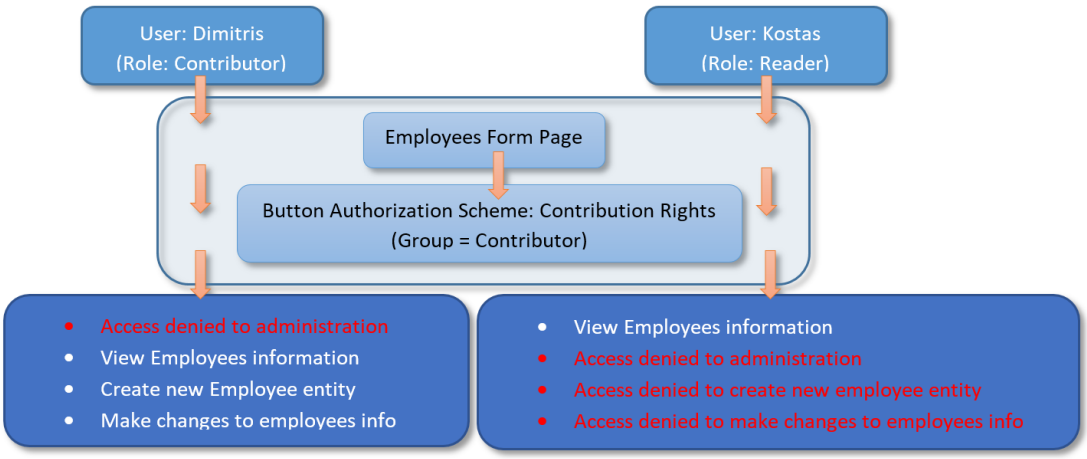



Abbildung 5.19: Beispiel für eine Zugriffskontrolle.

Oracle APEX bietet drei eingebaute Zugriffsrechte, die eine Zugriffskontrolle auf eine Anwendung oder ihre Komponenten ermöglichen. Diese Rechte sind Administration, Edit und View, und jedes entspricht einer Zugriffsrolle. Administration entspricht der Administratorrolle, Editieren entspricht der Contributorrolle und View entspricht der Readerrolle.

Um neue Benutzer hinzuzufügen und ihre Rollen zu definieren, müssen Sie zunächst in der APEX-Anzeigeumgebung zu **Manage Users and Groups** navigieren. Klicken Sie auf **Administration**, und wählen Sie **Manage Users and Groups**. Wählen Sie dann **Create User**, geben Sie die Benutzerkennung und das Kennwort ein und klicken Sie auf **Create User**, wie in Abbildung 5.20 dargestellt.

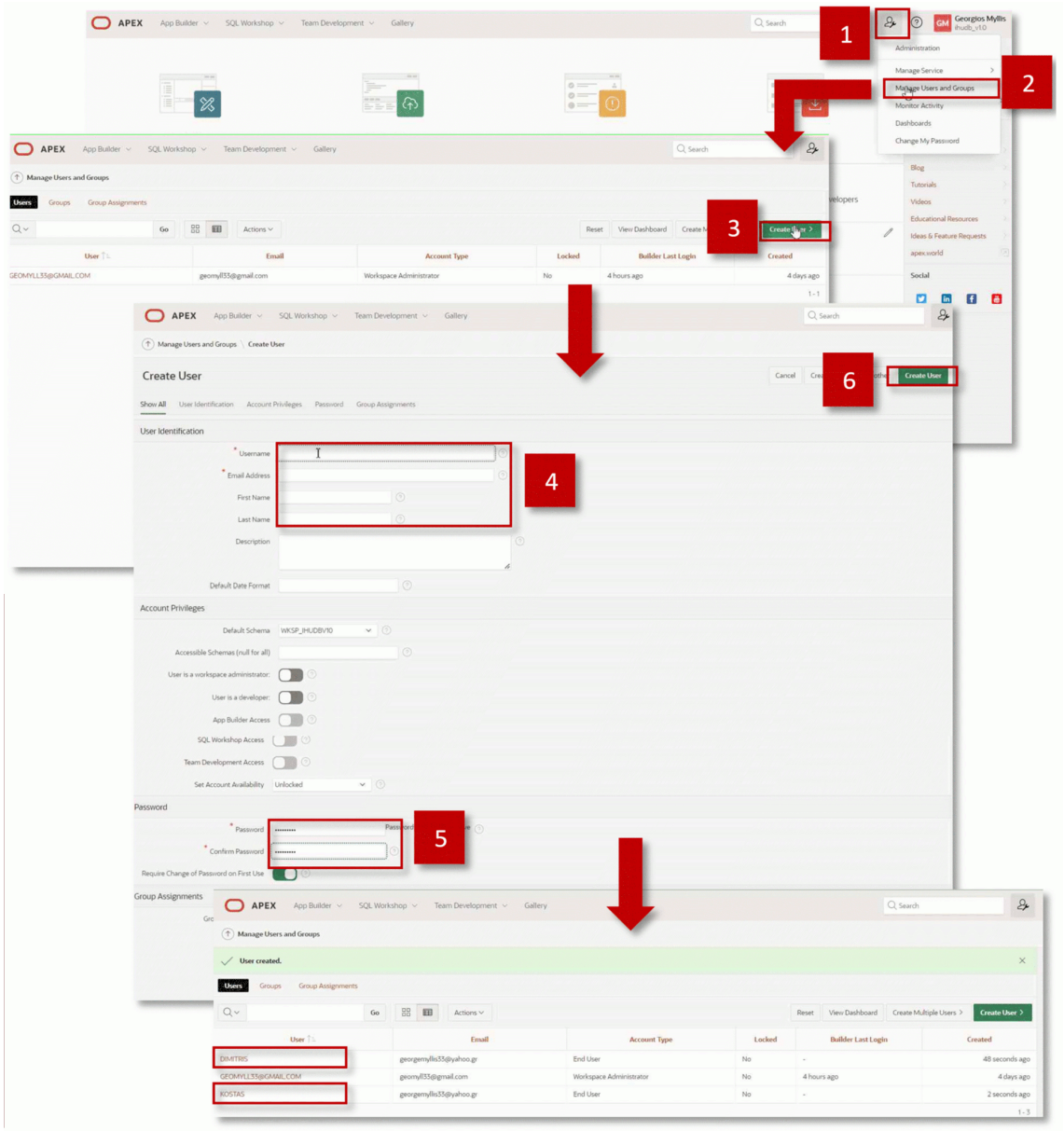



Abbildung 5.20: Neue Benutzer hinzufügen.

Als nächstes müssen Sie die Rollen für Ihre Benutzer definieren. In unserem Beispiel zeigen wir, wie dies innerhalb der App selbst geschehen kann.

Schritt 1: Melden Sie sich bei der App mit Administrator-Zugangsdaten an, gehen Sie zum Seitenmenü und klicken Sie auf **Administration**, klicken Sie auf **Add User**, geben Sie die Benutzerkennung und die Rolle ein und klicken Sie auf **Add User** in Abbildung 5.21.

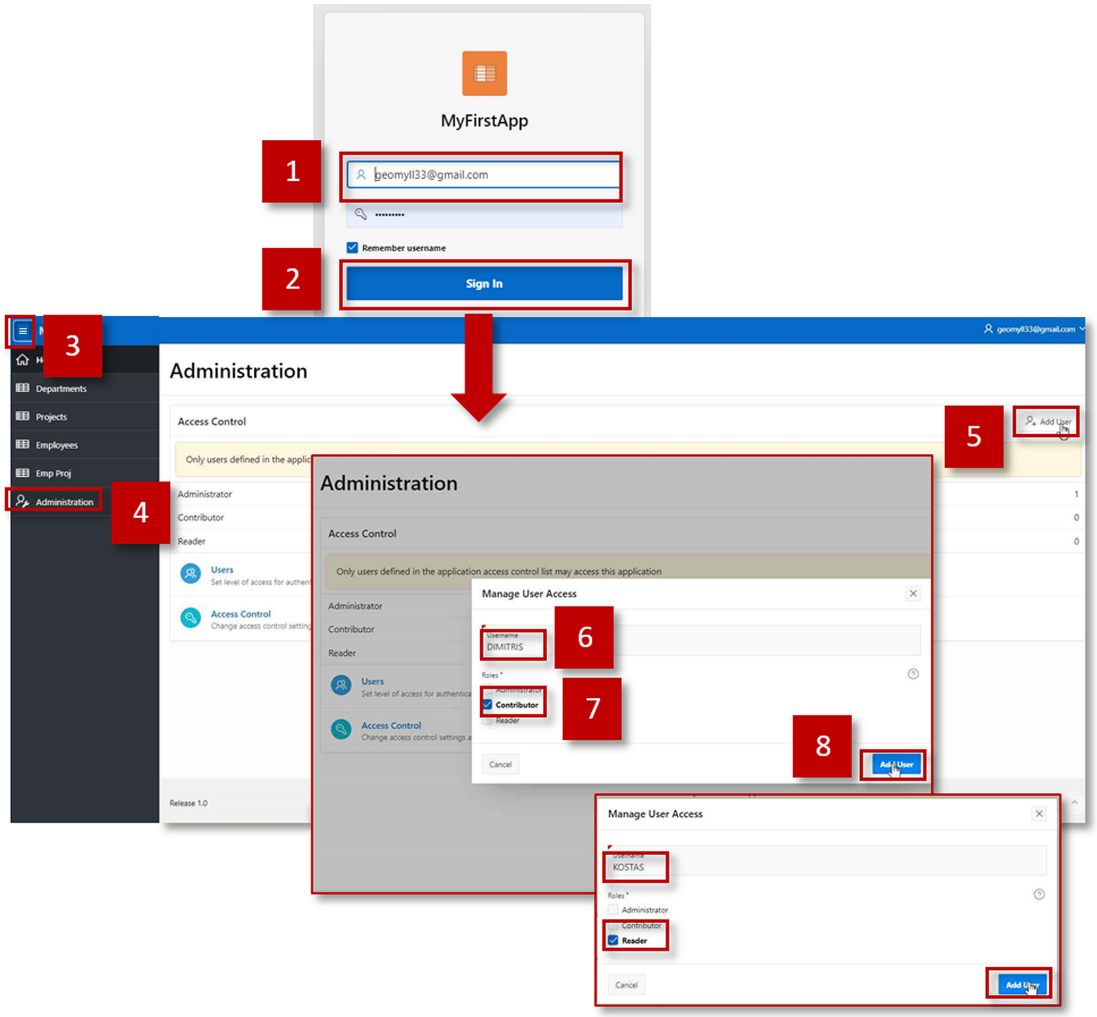



Abbildung 5.21: Rollen für Benutzer hinzufügen.

Schritt 2: Gehen Sie zur Arbeitsbereichumgebung der Anwendung und klicken Sie auf das Formular **employees-Form**. Wählen Sie im Seitendesigner des Formulars die Schaltfläche **Create**, und wählen Sie in der Sicherheits-Option **Contribution Rights** als Berechtigungsthema. Klicken Sie auf **Save**. Jetzt können nur Benutzer mit einer Beitragsrolle neue Mitarbeiter erstellen und hinzufügen, wie in Abbildung 5.22 dargestellt.

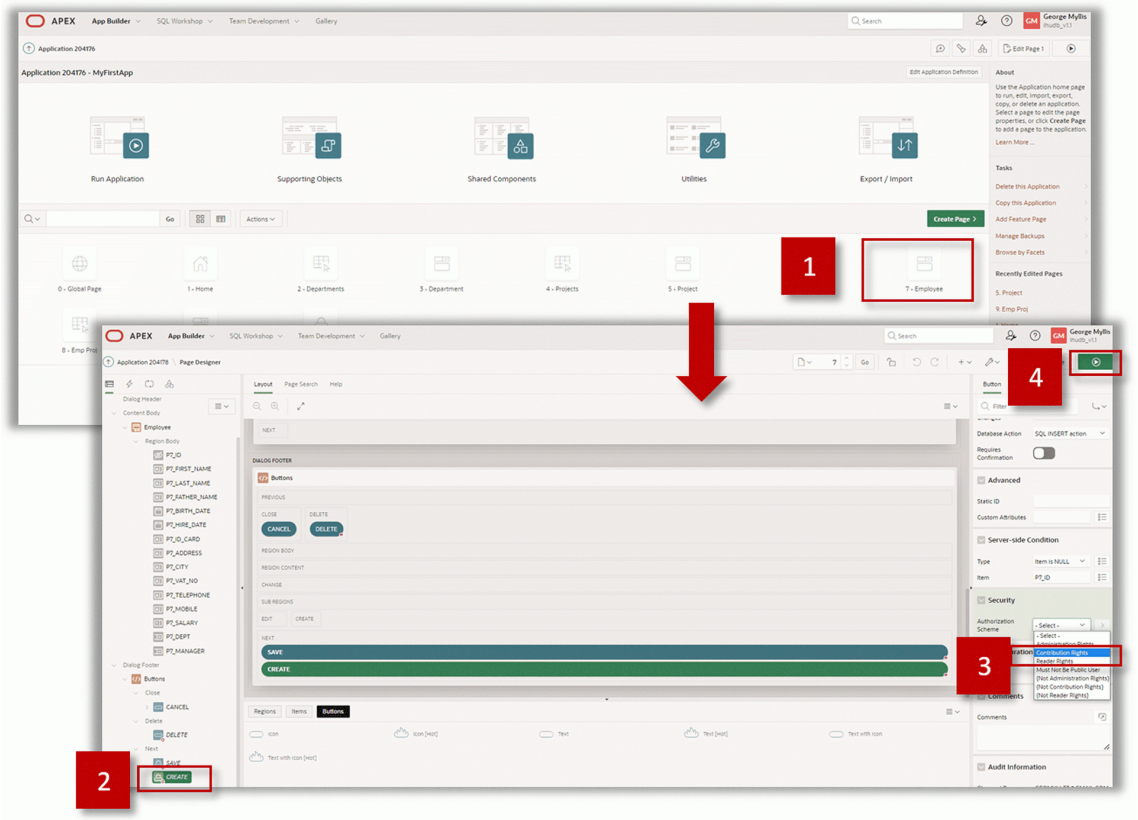



Abbildung 5.22: Schaltflächenkonfiguration erstellen.

Schritt 3: In diesem Schritt werden die Zugriffskontrollkonfigurationen für die Contributor-Rolle überprüft.

- Melden Sie sich bei der Anwendung mit den Anmeldedaten des Benutzers Dimitris an, dem die Rolle Contributor zugewiesen ist.
- Öffnen Sie das Menü und stellen Sie fest, dass die Option administration wie erwartet nicht vorhanden ist.
- Versuchen Sie, einen neuen Mitarbeiter als Contributor zu erstellen und hinzuzufügen, wie in Abbildung 5.23 gezeigt.
- Vergewissern Sie sich, dass Sie die Informationen zu den vorhandenen Mitarbeitern einsehen und Änderungen vornehmen können.
- Überprüfen Sie, ob Sie einen neuen Mitarbeiter erstellen können.

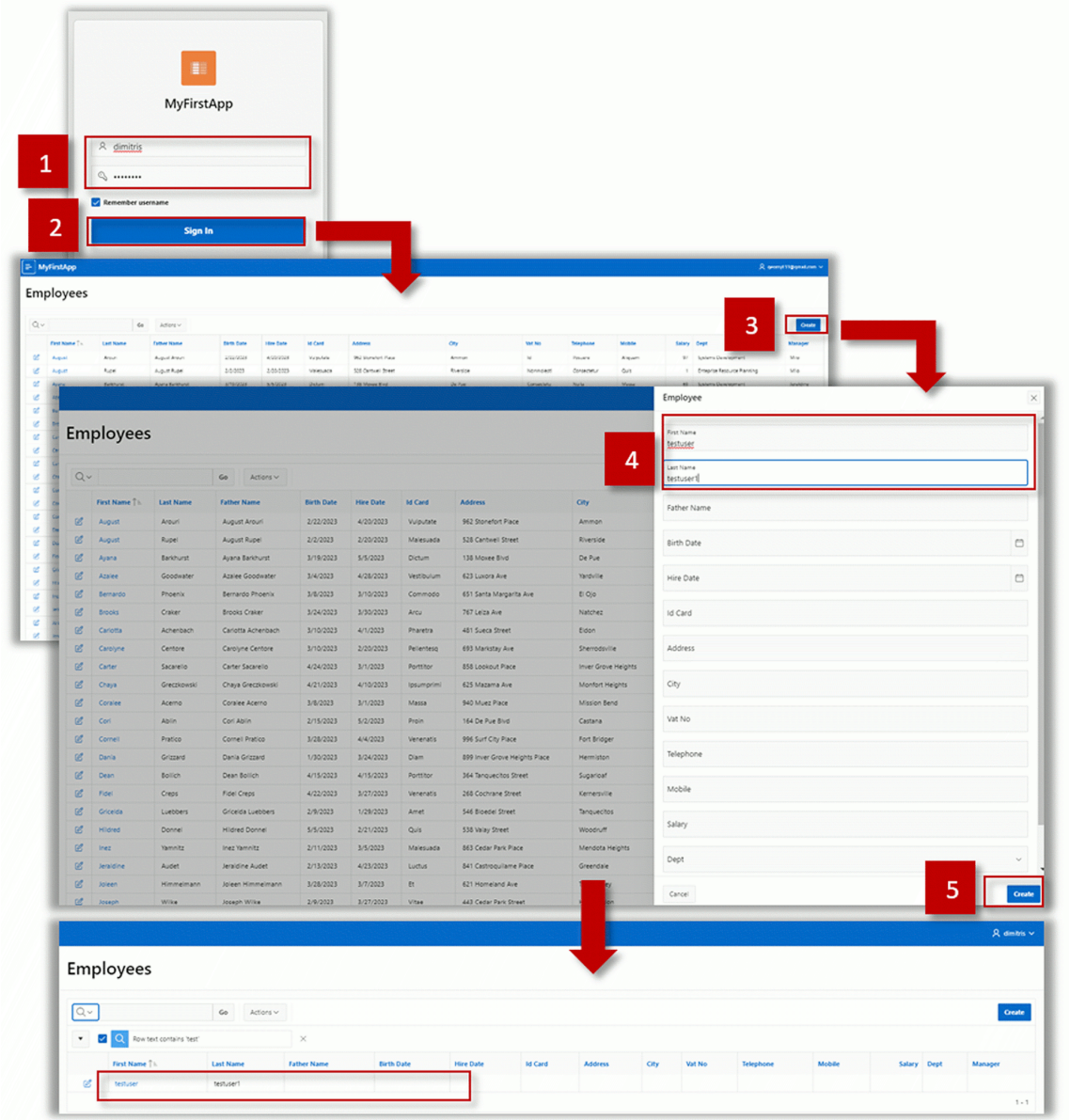



Abbildung 5.23: Contributor access and rights.

Schritt 4: In diesem Schritt werden die für die Readerrolle vorgenommenen Zugriffskontrollkonfigurationen überprüft.

- Melden Sie sich bei der Anwendung mit den Anmeldedaten des Benutzers Kostas an, dem die Rolle Reader zugewiesen ist.
- Öffnen Sie das Seitenmenü und stellen Sie fest, dass die Option administration wie erwartet nicht vorhanden ist.
- Klicken Sie auf das employees-Form in der Arbeitsplatzumgebung der Anwendung.
- Versuchen Sie, einen neuen Mitarbeiter zu erstellen oder hinzuzufügen. Sie erhalten die Meldung „access denied“, wie in Abbildung 5.24 dargestellt.
- Vergewissern Sie sich, dass Sie die Informationen zu den vorhandenen Mitarbeitern anzeigen können, aber keine Änderungen vornehmen können.

Diese Schritte bestätigen, dass die Zugriffskontrollkonfigurationen erfolgreich auf die Benutzer angewendet wurden.

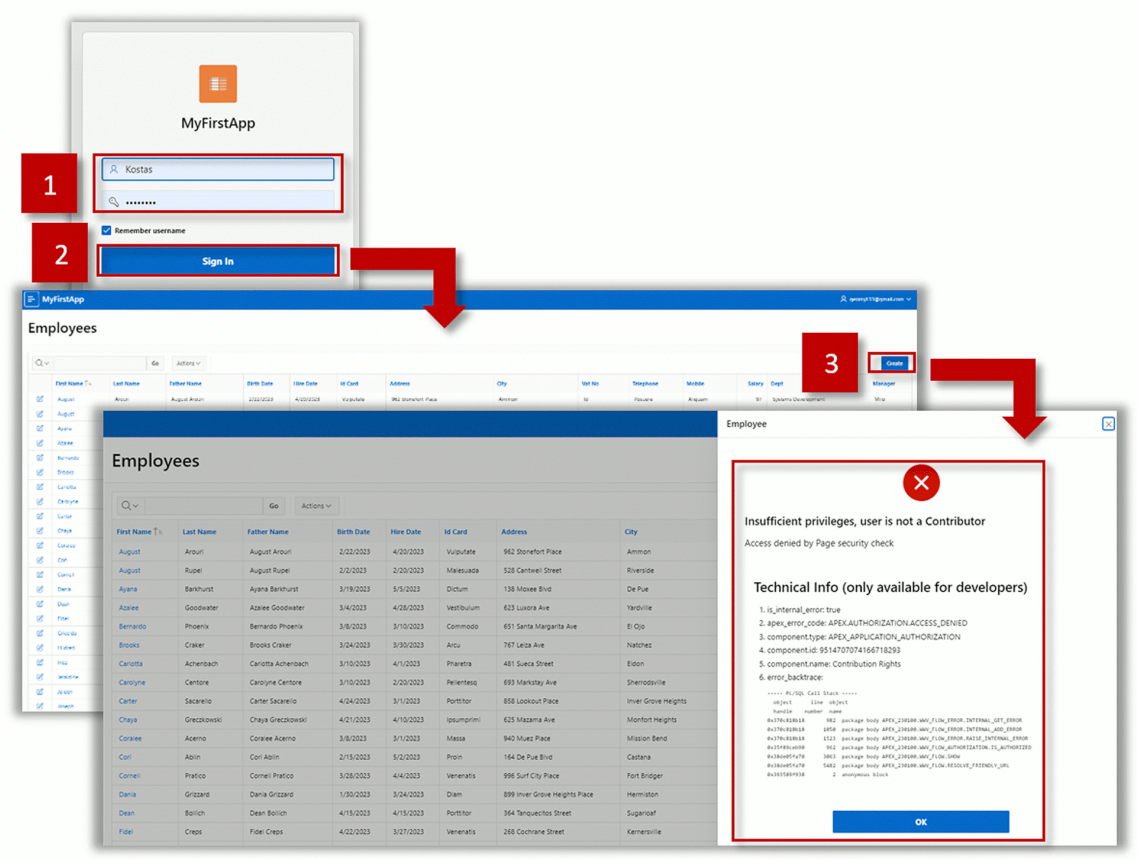



Abbildung 5.24: Readerzugang und Rechte.

## 5.7 Ergänzendes Lernmaterial

Sie können das folgende zusätzliche Lernmaterial finden:

- Skripte
- Anwendung
- Video-Guides

Alle zusätzlichen Lernmaterialien sind auf der öffentlichen BeeAPEX-Projektseite verfügbar. Melden Sie sich als Gastbenutzer an (kein Passwort erforderlich). Das Lehrbuch finden Sie im Abschnitt Books, die Skripte im Ordner Part 2 > Chapter 5 im Abschnitt Scripts und die Videoanleitungen in der Sammlung der Videoanleitungen. Material für Kurzkurse finden Sie in der Sektion Short courses.

### 5.7.1 Exportierte Anwendungen

Die exportierte Anwendung ist verpackt. Bei der Installation werden sowohl Tabellen erstellt als auch Daten eingefügt. Bei der Deinstallation werden alle in dieser Anwendung verwendeten Datenbankobjekte entfernt.

### 5.7.2 Video-Anleitungen

Videoanleitungen zeigen jeden Schritt der Anwendungsentwicklung.

## 5.8 Fragen

1. Wie viele Möglichkeiten gibt es, eine neue Anwendung in ORACLE APEX Application Builder zu erstellen?

2. Wie können Sie der Anwendung Benutzer hinzufügen?
3. Wie viele eingebaute Benutzerprivilegien gibt es in ORACLE APEX für die Zugriffskontrolle auf eine Anwendung oder ihre Komponenten?

## 5.9 Antworten

1. Es gibt 3 Möglichkeiten
   - Neue Anwendung
   - Aus einer Datei
   - Starter-App
2. Navigieren Sie zur Arbeitsbereichumgebung der Anwendung und wählen Sie Manage Users and Group
3. Es gibt drei
   - Administrator
   - Contributor
   - Reader



# 6. Wie werden Berichte verwaltet?

ZUZANA ŽILLOVÁ, ERIK MALINA, MATEJ GROCHAL, ANDREJ STANÍK, ANDREA MELEKOVÁ, MICHAL KVET, MIROSLAV POTOČÁR

Die Daten in den Datenbanken sind oft bedeutungslos, wenn wir keine Ergebnisse von ihnen erhalten, die uns mehr über die tatsächlichen Informationen, die sie enthalten, verraten. Um die Ergebnisse darzustellen, zu beobachten und herauszufinden, verwenden wir einfach Berichte, mit denen wir uns im folgenden Kapitel beschäftigen werden. Wir werden lernen, wie man einen solchen Bericht in der APEX-Umgebung einfach erstellt und wie man ihn bearbeitet. Wir werden uns mit verschiedenen Arten von Berichten beschäftigen, wie z.B. interaktiven Berichten, klassischen Berichten, Spaltenumschaltberichten oder Berichten mit grafischen Darstellungen.

## 6.1 Bericht

**Ein Bericht** ist ein *Bereich*, der das Durchsuchen von Daten vereinfacht und Daten in Zeilen anzeigt. Er kann als eine Einheit dargestellt werden, oder er kann verwendet werden, um Seiten in logische Einheiten zu unterteilen. Er kann statisch sein, nur mit Titel und Formatierung, aber ohne zusätzliche Funktionen. Es kann aber auch ein dynamischer Bereich sein, der spezifische Funktionen wie *Kalender, Bericht* oder *Liste* hat. Diese Bereiche sind mit ihrer Grundfunktionalität vorprogrammiert, die auch von den Benutzern leicht bearbeitet werden kann.

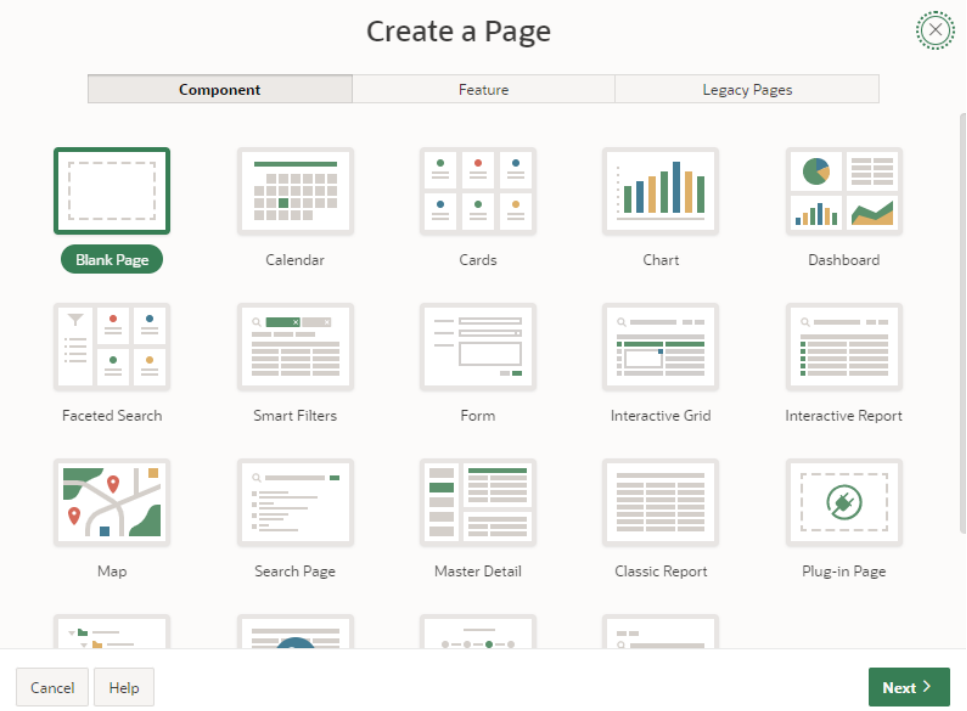



Abbildung 6.1: Auswahl des Seitentyps.

Wie in Abbildung 6.1 zu sehen ist, kann der Benutzer bei der Erstellung einer **Seite** aus vorgefertigten Seitentypen wählen, die bereits Berichte enthalten, so dass es nicht notwendig ist, einen Bereich durch den Benutzer selbst einzufügen. Ein Beispiel für ein Formular zur Berichtserstellung ist in Abbildung 6.2 dargestellt. Bei der Erstellung einer Seite, die einen Bericht enthält, müssen der Name der Seite und die Datenquelle der Seite angegeben werden. Die Daten können von einer *lokalen Datenbank*, einem *REST-fähiges SQL* oder *REST* geladen werden. Bei der Auswahl der Option „local database“muss ausgewählt werden, was genau als Berichtsquelle verwendet werden soll - der Benutzer kann nicht nur die Datenbank *Tabelle* selbst verwenden, sondern auch vorhandene *views* oder eigene *SQL query*-Anweisungen.

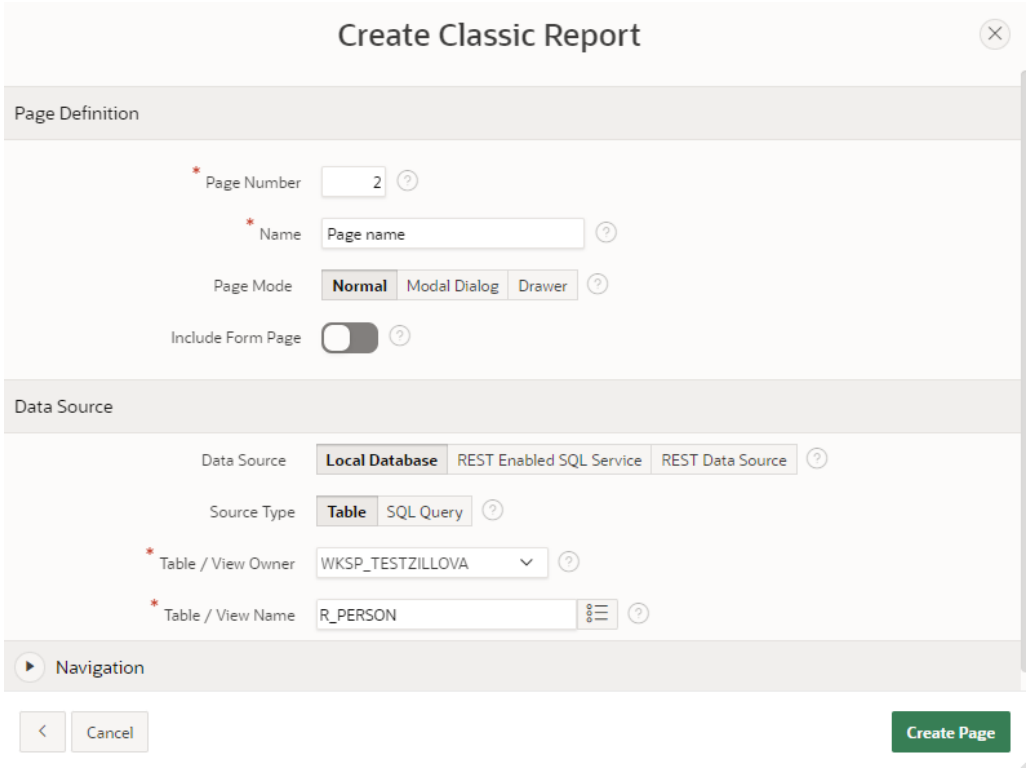



Abbildung 6.2: Erstellung einer Seite mit Bericht.

Wenn die Seite erstellt wird, erscheint auf der linken Seite ein Abschnitt ähnlich dem in Abbildung 6.3. In der seitlichen Navigationsleiste sehen Sie die Seite, ihren Namen und alle ihre Komponenten. Wenn Sie auf eine der Komponenten klicken, wird sie hervorgehoben. Dann sehen Sie die Spalten, die sie enthält, und auch ihre Unterregionen oder Elemente. Mit einem Rechtsklick können Sie einen neuen leeren Bereich erstellen.

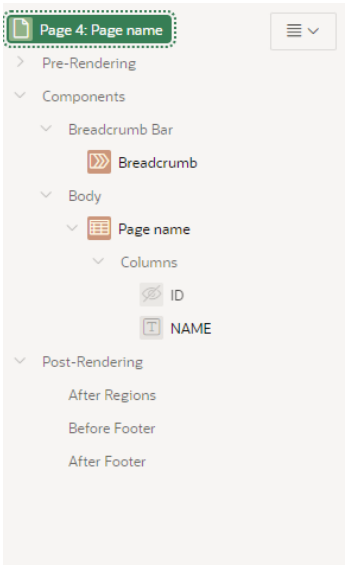



Abbildung 6.3: Rendering.

In Abbildung 6.4 sehen Sie die Region und ihre Attributdetails. **Identifikation** definiert den Seitentitel und die Art des Berichts (z. B. klassischer Bericht). Die Abbildung 6.5 zeigt mehrere

Beispiele für den Abschnitt Quelle, in dem die Quelle der Daten für unseren Bericht sowie Informationen über die Datenherkunft (wie oben erwähnt: *Tabelle, Ansicht, Funktion* oder *SQL-Abfrage*) definiert werden. **Where Clause** gibt an, welche Bedingung für die Datenfilterung angewendet werden soll.

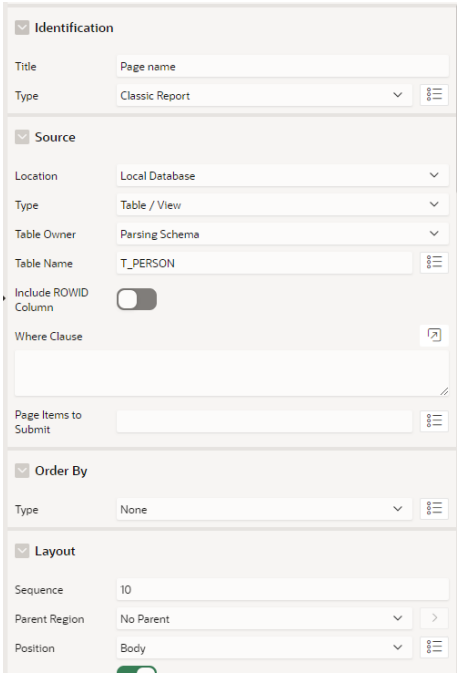



Abbildung 6.4: Eigenschaften einer Region.

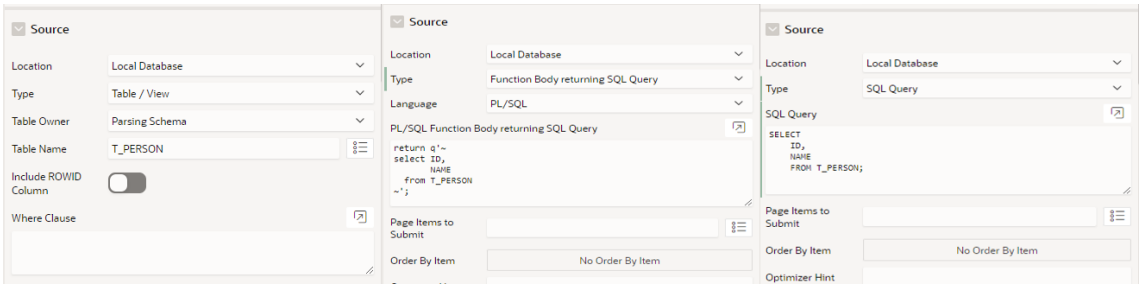



Abbildung 6.5: Verschiedene Datenquellen.

In Abbildung 6.6 sehen Sie **Layout**, wo Sie die Position der Region innerhalb der Seite angeben können. Mit **Parent Region** legen Sie fest, welcher Bereich unseren Bericht enthalten soll. Sie können dies auch über **Position** einstellen. Die Positionen werden durch die spezifische *Seitenvorlage* definiert. Nachdem die Seite erstellt wurde, wird sie mit einem bestimmten Thema und Layout erstellt, das von der gewählten Vorlage abhängt. Die Standardvorlage für das Thema **Universal** heißt **Standard**.

Die Regionen sind in einem Raster organisiert, d.h. jede Region hat ihren eigenen Standort, der durch Zeile und Spalte definiert ist. Diese Regionen sind nach ihrer **Folgenummer** geordnet. Die erste Region befindet sich in der ersten Zeile und der ersten Spalte des Rasters. Jede weitere Region wird dann nach einem dieser Muster angeordnet: in derselben Zeile und Spalte wie die vorherige Region, und sie wird unter der vorherigen Region angezeigt. Die Attribute **Start New Row** und **New Column** werden verwendet, um die Art der Organisation für diese Regionen festzulegen.

*Größe* und *Ort* werden nach der Zuordnung von Zeile und Spalte berechnet. Dies geschieht

durch die Berechnung der Belegung und Breite für jeden Teil der Zeile durch die Werte der Parameter **Column** und **Column Span**.

Die Einrückung des Bereichs in seiner Zeile wird durch das Attribut **Column** definiert. Der Wert dieser Spalte liegt im Bereich **1 - 12**, der sich auf **1** von **12 Punkten** des Gitters, das die Breite aufteilt, bezieht. Der Wert **1** steht für den äußersten linken Teil und **7** für die *Mitte* der Seite.

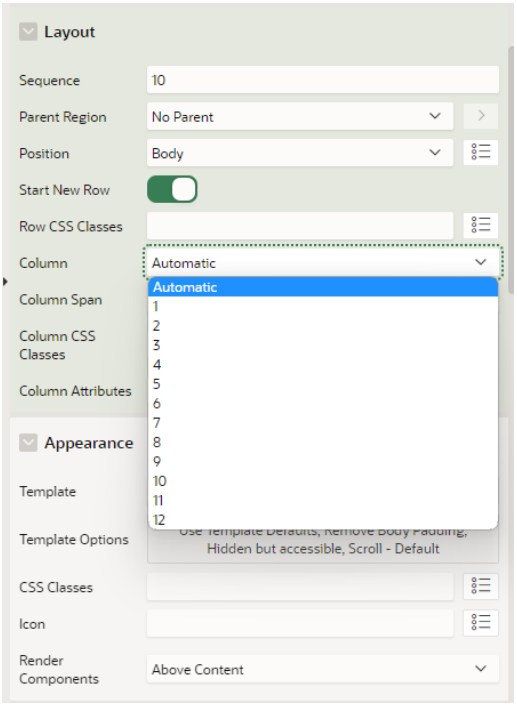



Abbildung 6.6: Layout.

In Abbildung 6.7 sehen Sie den Abschnitt **Auftritt**, der zur Bearbeitung der Vorlage für den angegebenen Bereich dient. Abbildung 6.8 zeigt vordefinierte Vorlagen, aus denen der Benutzer auswählen kann. Jede Vorlage kann durch Anklicken der Vorlagenoptionen oder durch Schreiben eigener **CSS**-Anweisungen angepasst werden. In den Vorlagenoptionen können Sie mehrere Regeln wie die *Texthöhe*, *Kopfzeilensichtbarkeit* oder sogar den *Abstand zwischen Elementen* definieren.

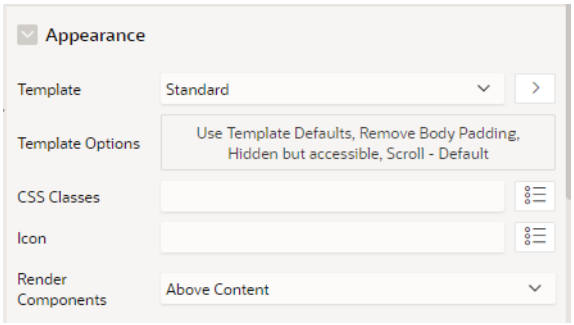



Abbildung 6.7: Erscheinungsbild.

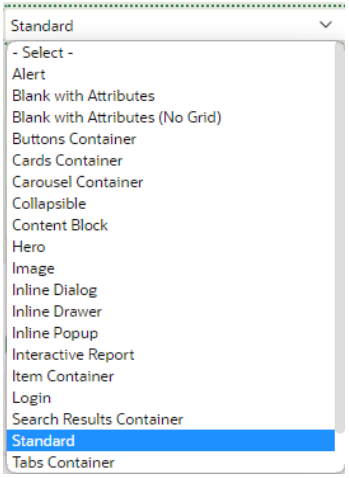



Abbildung 6.8: Vorlagen.

Abbildung 6.9 zeigt den Abschnitt **Advanced**, in dem Sie aus vier Optionen wählen können. **Static ID** ist für die Einrichtung des Bezeichners des Berichts bestimmt. Diese **Static ID** wird als **ID** für das **HTML**-Element innerhalb der Seite verwendet, wenn der Benutzer das Aussehen der Seite mit **CSS** konfigurieren möchte. **Custom Attributes** sind für zusätzliche Informationen für das **HTML**-Element. Benutzerdefinierte Attribute werden als Name/Wert gespeichert, zum Beispiel: *name* = *MY_REPORT*"

**Region Display Selector** ist eine Regionskomponente, die Navigationssteuerungen für andere Regionen innerhalb der Seite bietet. **Exclude Title from Translation** verhindert, dass der Titel von der Übersetzung ausgeschlossen wird, falls die Seite definierte Übersetzungen hat.

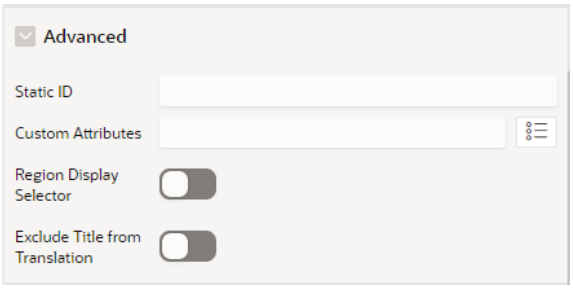



Abbildung 6.9: Erweitert.

**Kopf- und Fußzeile**, wie in Abbildung 6.10 gezeigt, werden zur Bearbeitung des Inhalts dieser Teile eines Berichts verwendet. Es ist wichtig zu erwähnen, dass sich die Fußzeile und die Kopfzeile auf den Bericht und nicht auf die gesamte Seite beziehen.

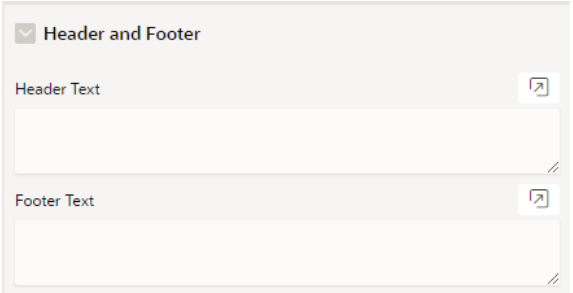



Abbildung 6.10: Kopf- und Fußzeile.

Die Arten von **Server-side Conditions** sind in Abbildung 6.11 aufgeführt. In Anwendungen, die mit Oracle APEX entwickelt wurden, werden **Server-side Conditions** in der Initialisierungsphase des Ladens der Website aufgelöst. Der Bericht wird nicht angezeigt, wenn die serverseitigen Bedingungen nicht erfüllt sind. Es gibt eine Vielzahl von Arten von Bedingungen. Zum Beispiel können Sie Werte von Elementen vergleichen oder es gibt Bedingungen für die Überprüfung von Anfragen.

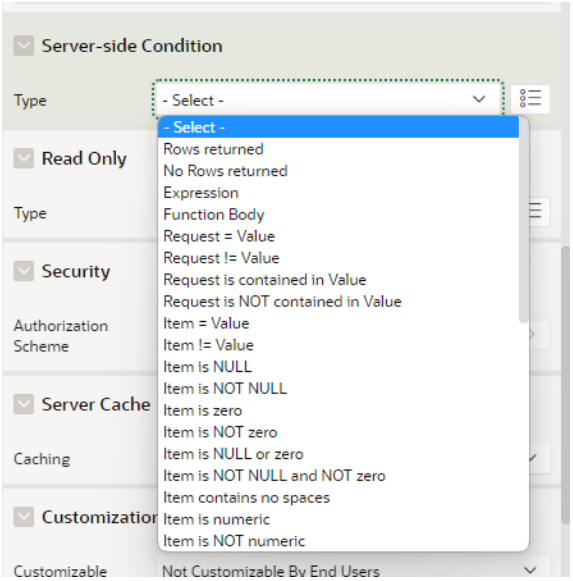



Abbildung 6.11: Server-side Condition.

Das in Abbildung 6.12 dargestellte Element **Read Only** hat dieselben Optionen wie das in Abbildung 6.11 dargestellte Element **Server-side Condition**. Natürlich ist es nicht möglich, die Ansicht der ausgewählten Region zu ändern, da sie schreibgeschützt ist. Im Falle des Basisberichts ist es nicht möglich, die Spaltenwerte zu sortieren.




Abbildung 6.12: Read Only.

In Abbildung 6.13 sehen Sie die letzten drei Abschnitte, die erklärt werden. Diese sind **Security, Server Cache** und **Customization**. Der Abschnitt **Security** enthält Optionen für Einschränkungen in Bezug auf die Anzeige der Website - z. B. *öffentliche Website, nur für autorisierte Benutzer*. Der Abschnitt **Server-Cache** enthält Einstellungen zur Cache-Verwaltung. Im Fenster **Customization** können Sie den Zugang zur Änderung des Berichts anpassen.

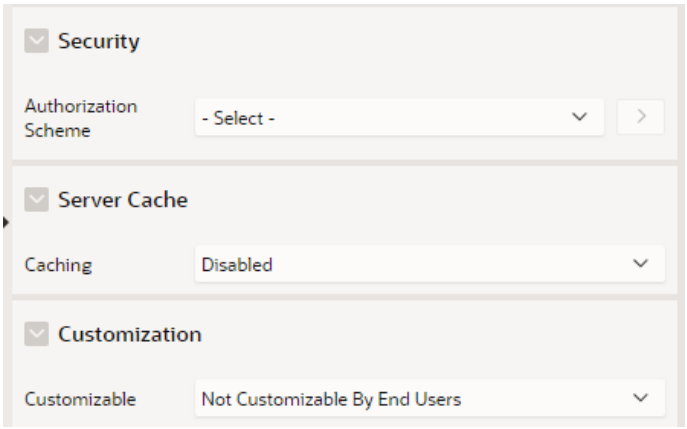



Abbildung 6.13: Security, Server Cache und Customization.

Eine weitere Gruppe von Abschnitten ist **Attributes**. Hier können Sie Dinge wie *die Anzahl der angezeigten Zeilen*, *die Möglichkeit der Auflistung von Berichtsergebnissen* und *die Art der Auflistung* im Bericht festlegen.

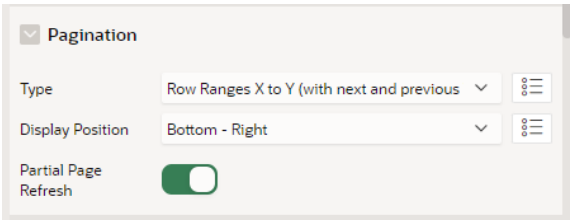



Abbildung 6.14: Seitennummerierung.

Eine weitere interessante Funktion ist die Festlegung der maximalen Anzahl der geladenen Zeilen. Sie können auch festlegen, was angezeigt werden soll, wenn *keine Daten gefunden werden*. Es ist möglich, Ihre Berichte als **.csv**-Datei herunterzuladen oder sie auszudrucken. Alles, was in diesem Abschnitt erwähnt wird, finden Sie in **Attributes**.

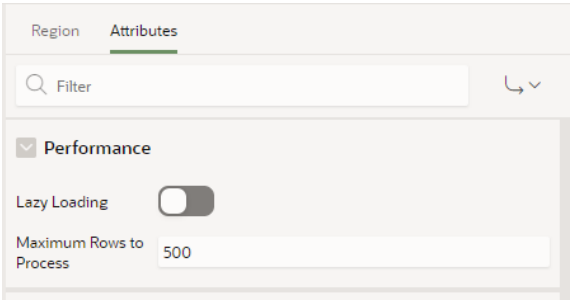



Abbildung 6.15: Anzahl der zu ladenden Zeilen.

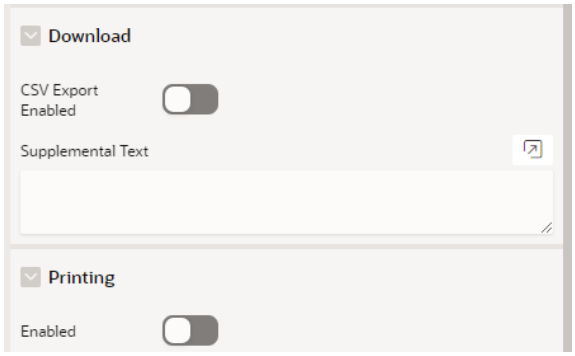



Abbildung 6.16: Herunterladen und Drucken.

Ein sehr wichtiger Teil ist das *Zentrum* der Anwendung. In Abbildung 6.17 sehen Sie **Layout**, das uns zeigt, wie alle Komponenten auf unserer Seite angeordnet sein werden. Die gesamte Seite ist in einem Raster strukturiert, so dass man sich intuitiv vorstellen kann, wo die Komponenten platziert werden. Um so benutzerfreundlich wie möglich zu sein, können Sie den Bereich/das Element auswählen und per *Drag & Drop* an die gewünschte Stelle verschieben. Für jede Region gibt es eine Standardvorlage, die beim Hinzufügen bereitgestellt wird. Alle notwendigen Einstellungen sind bereits vorgenommen.

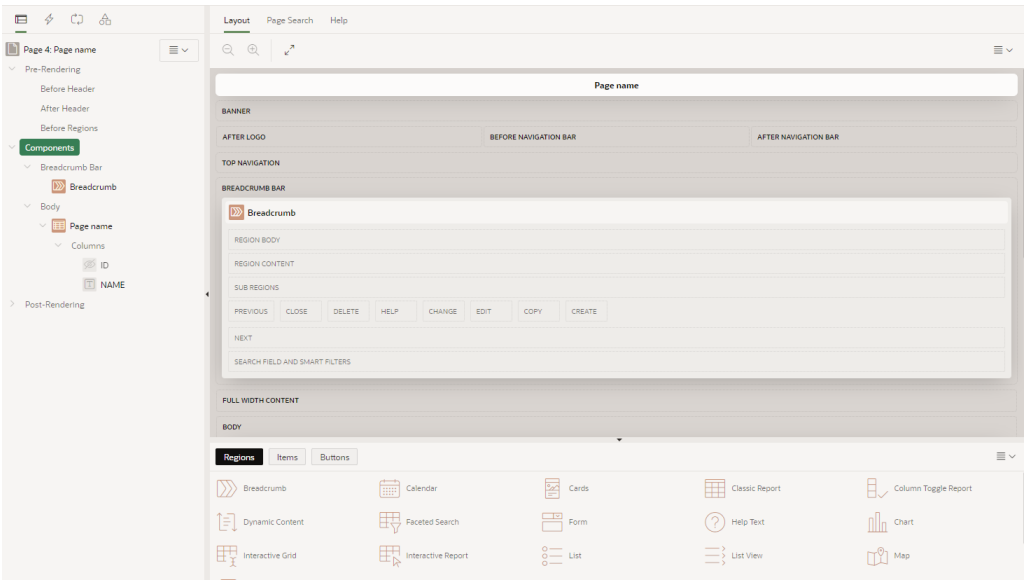



Abbildung 6.17: Seitenzentrums-Designer.

Wie Sie auf der linken Seite in Abbildung 6.17 sehen können, erstellt APEX standardmäßig einige Komponenten für den Bericht. Jede Komponente stellt eine Spalte dar, und es ist möglich, sie auf die gleiche Weise zu ändern, wie Sie es mit dem Bericht tun.

Die Spalteneinstellungen sind in Abbildung 6.19 dargestellt. In **Identification** sehen Sie zwei Merkmale - **Type** und **Name of Column**. Es ist nicht möglich, **Name of Column** zu ändern, da sein Wert aus der *SQL-Abfrage* bezogen wird. **Type of Column**, wie in Abbildung 6.18 gezeigt, bestimmt, wie die Spalte angezeigt wird - um einige von ihnen zu nennen: *Hidden* (es wird nichts angezeigt), *Plain Text*, *Bild anzeigen, Prozentdiagramm* und so weiter. Standardmäßig sind alle Spalten als *Plain Text* markiert.

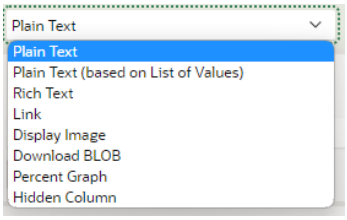



Abbildung 6.18: Type of column.

**Heading** besteht aus zwei Optionen für die Überschrift. Im Gegensatz zum Spaltennamen, kann **Heading** (Name) bearbeitet werden. Die Überschrift kann auch **HTML**-Code enthalten. Die Option **Alignment** spricht für sich selbst. Der Abschnitt **Layout** bietet die Option **Sequenz**, die festlegt, in welcher Reihenfolge die Spalten angezeigt werden, sowie die Option **Alignment**, wie im vorherigen Abschnitt.

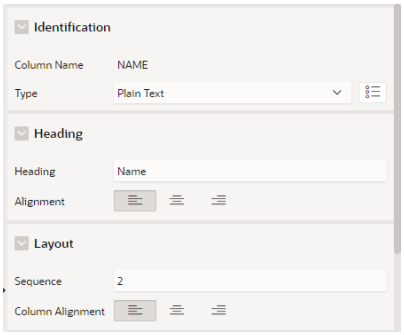



Abbildung 6.19: Spalteneinstellungen.

Der letzte wichtige Teil ist **Appearance**, der bearbeitet werden sollte, wenn Sie Zahlen als Zeichenkette in einem *besonderen Format* behandeln wollen. Wie Sie in Abbildung 6.20 sehen können, können Sie aus einer Vielzahl von Grundformaten wählen, wenn Sie auf **Format Mask** klicken. Diese Formate werden von APEX selbst angeboten, aber Sie können auch Ihr eigenes Format angeben, wenn Sie wollen.

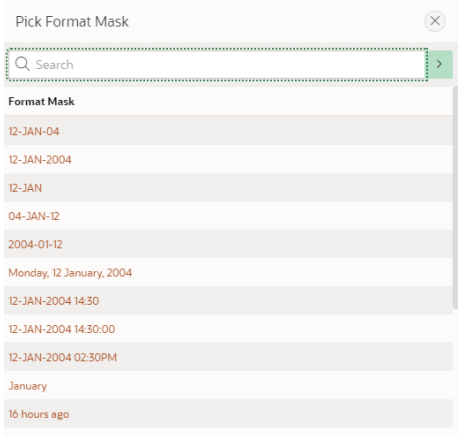



Abbildung 6.20: Format Mask.

## 6.2 Klassischer Report

Ein **klassischer Report** ist ein Berichtstyp, der nur die grundlegende Anzeige von Daten ermöglicht, ohne dass der Benutzer die Anzeigeoptionen ändern kann. Das Einzige, was dem Benutzer zur Verfügung steht, ist das Sortieren der Daten nach einzelnen Spalten, was ebenfalls eingestellt werden muss.

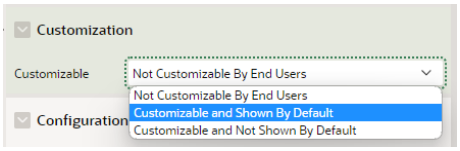



Abbildung 6.21: Personalisierung.

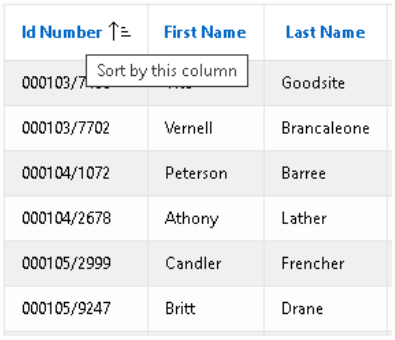



Abbildung 6.22: Option zum Sortieren nach Spalten.

Wenn Sie sehen möchten, wie die Seite aussehen wird, können Sie auf den grünen Pfeil im Kreis klicken. Wie Sie in Abbildung 6.23 sehen können, befindet sich der Pfeil in der oberen rechten Ecke. In Abbildung 6.24 können wir sehen, wie der klassische Bericht aussieht. Er hat Spalten mit Namen und Daten. Außer dem Sortieren nach Spalten ist keine Interaktion für den Benutzer möglich.

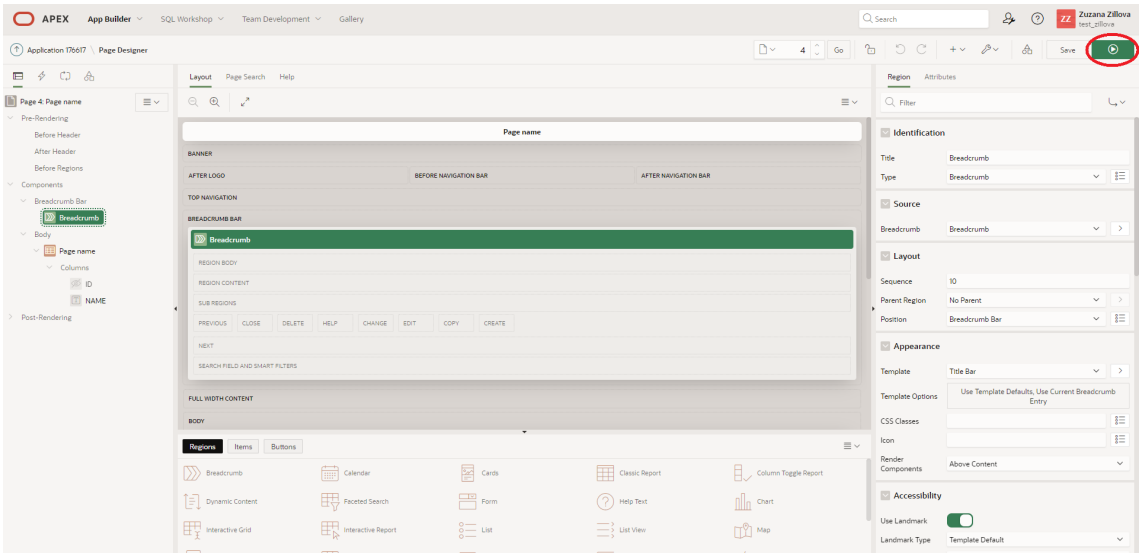
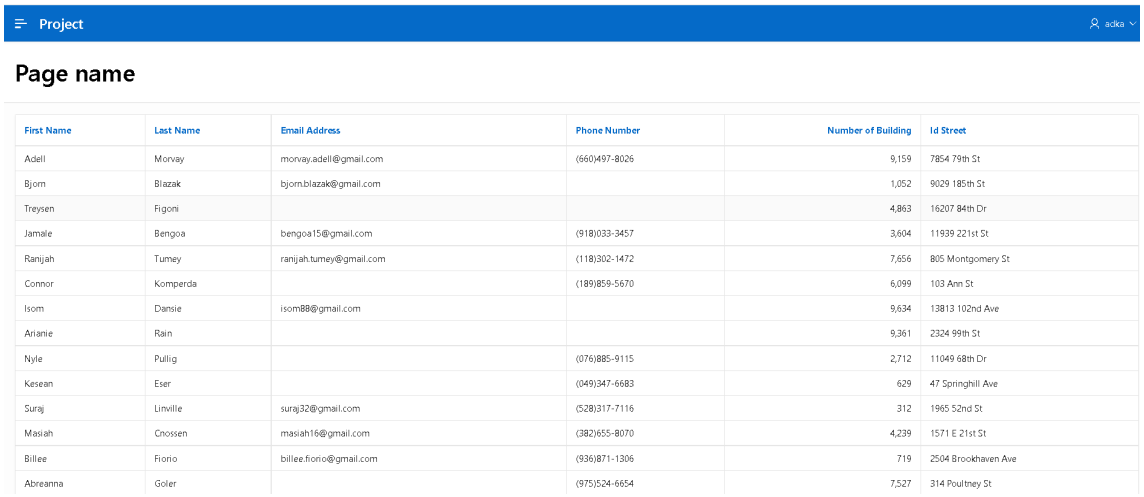



Abbildung 6.23: Position des grünen Pfeils für die Seitenpräsentation.

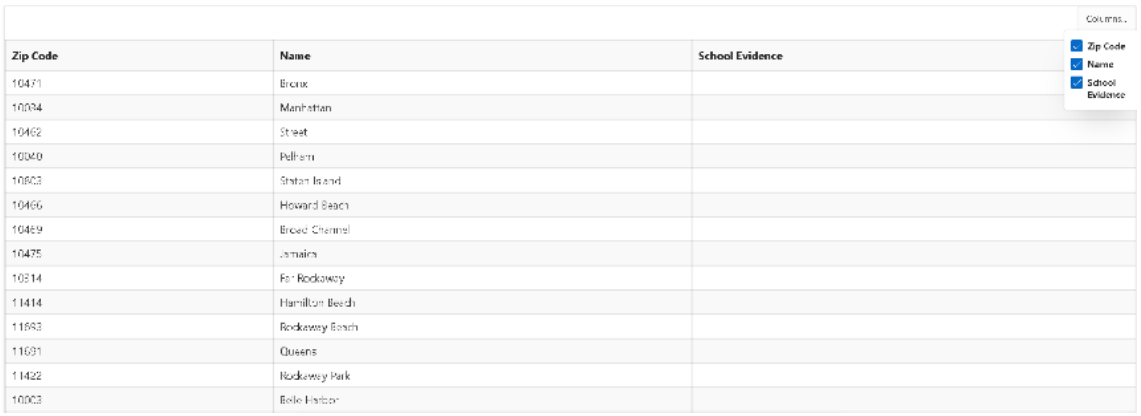



| First Name | Last Name | Email Address | Phone Number | Number of Building | Id Street |
|---|---|---|---|---|---|
| Adell | Morvay | morvay.adell@gmail.com | (660)497-8026 | 9,159 | 7854 79th St |
| Bjorn | Blazak | bjorn.blazak@gmail.com | | 1,052 | 9029 185th St |
| Treysen | Figoni | | | 4,863 | 16207 84th Dr |
| Jamale | Bengoa | bengoa15@gmail.com | (918)033-3457 | 3,604 | 11939 221st St |
| Ranijah | Tumey | ranijah.tumey@gmail.com | (118)302-1472 | 7,656 | 805 Montgomery St |
| Connor | Komperda | | (189)859-5670 | 6,099 | 103 Ann St |
| Isom | Dansie | isom88@gmail.com | | 9,634 | 13813 102nd Ave |
| Arianie | Rain | | | 9,361 | 2324 99th St |
| Nyle | Pullig | | (076)885-9115 | 2,712 | 11049 68th Dr |
| Kesean | Eser | | (049)347-6683 | 629 | 47 Springhill Ave |
| Suraj | Linville | suraj32@gmail.com | (528)317-7116 | 312 | 1965 52nd St |
| Masiah | Cnossen | masiah16@gmail.com | (382)655-8070 | 4,239 | 1571 E 21st St |
| Billee | Fiorio | billee.fiorio@gmail.com | (936)871-1306 | 719 | 2504 Brookhaven Ave |
| Abreanna | Goler | | (975)524-6654 | 7,527 | 314 Poultney St |

Abbildung 6.24: Beispiel für einen klassischen Bericht.

## 6.3 Spaltenumschaltbericht

Der **Spaltenumschaltbericht** ist fast dasselbe wie der klassische Bericht, nur eine zusätzliche Funktion ist die Angabe, welche Spalte sichtbar sein soll und welche nicht. Wir können es in der rechten oberen Ecke von *Abbildung 6.25* sehen.




| Zip Code | Name | School Evidence |
|---|---|---|
| 10471 | Bronx | |
| 10034 | Manhattan | |
| 10462 | Street | |
| 10940 | Pelham | |
| 10303 | Staten Island | |
| 10466 | Howard Beach | |
| 10469 | Broad Channel | |
| 10475 | Jamaica | |
| 10314 | Far Rockaway | |
| 11414 | Hamilton Beach | |
| 11693 | Rockaway Beach | |
| 11691 | Queens | |
| 11422 | Rockaway Park | |
| 10003 | Belle Harbor | |

Abbildung 6.25: Möglichkeit zur Änderung der Sichtbarkeit von Spalten im Spaltenumschaltbericht.

## 6.4 Interaktiver Bericht

Der **Interaktive Bericht**, der in Abbildung 6.26 dargestellt ist, ermöglicht es den Benutzern, einige Anpassungen des Berichts vorzunehmen, während sie die Seite betrachten, ohne auf den Kern der Anwendung zugreifen zu müssen. Ein interaktiver Bericht wird wie jeder andere erstellt und unterscheidet sich in seinem Aussehen nicht wesentlich. Im Folgenden werden wir uns die einzelnen Funktionen und Anpassungen, die der Benutzer im interaktiven Bericht vornehmen kann, genauer ansehen.

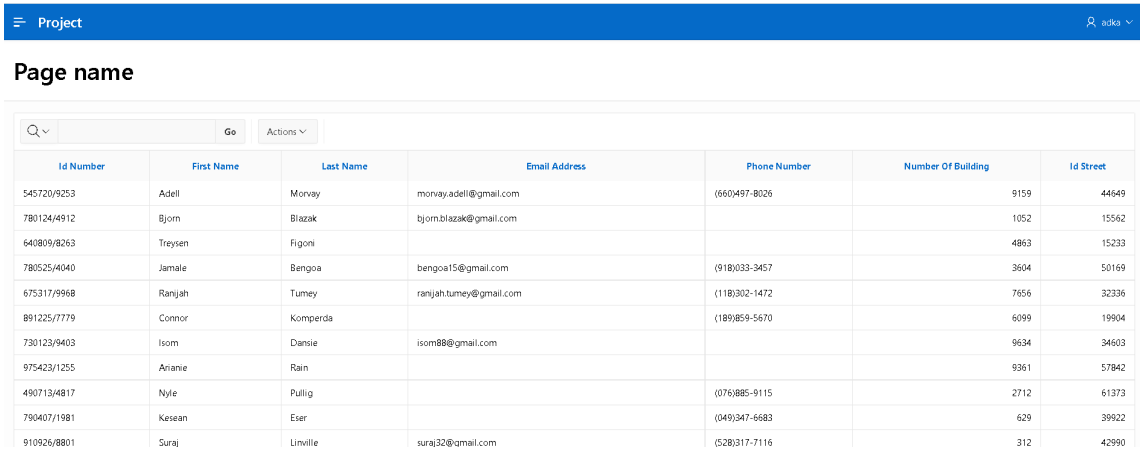



Abbildung 6.26: Interaktiver Bericht.

Das in Abbildung 6.27 dargestellte Suchfeld besteht aus vier Elementen:

- **Lupe,**
- **Textfeld,**
- **Startknopf,**
- **Aktionen-Schaltfläche**.

Die ersten drei Elemente ermöglichen es dem Benutzer, Berichtszeilen zu filtern. Wir geben einen Suchbegriff in das Feld ein, klicken auf die **Lupe**, wählen aus, welche Spalten angezeigt werden sollen, und klicken dann auf den **Startknopf**. Es werden die Datensätze angezeigt, die den Suchbegriff enthalten.

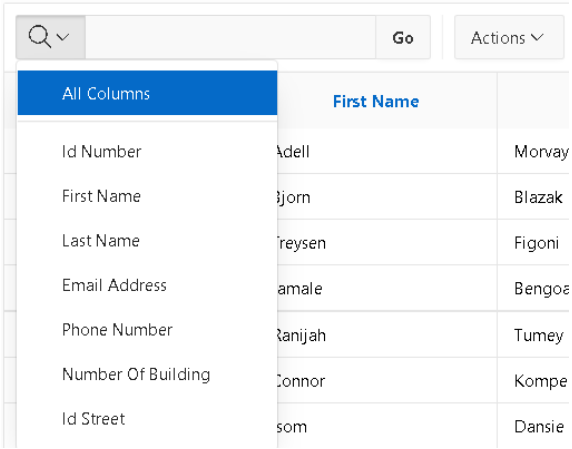



Abbildung 6.27: Suchfeld.

Wir möchten zum Beispiel nach Datensätzen suchen, die in einer beliebigen Spalte die Zahl **"730123/9403"** enthalten. Dieser Filter wird angezeigt und kann durch Klicken auf die Schaltfläche **X** deaktiviert werden. Die Situation ist in Abbildung 6.28 dargestellt. Es können mehrere Filter gleichzeitig in verschiedenen Spalten aktiv sein.

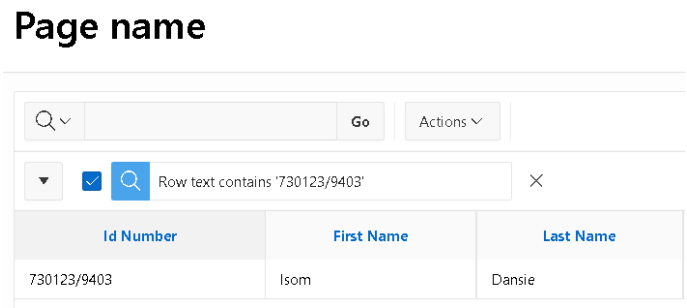



Abbildung 6.28: Suche.

Die **Aktionen**-Schaltfläche bietet weitere Möglichkeiten zur Anpassung der Anzeige des Berichts. Nach dem Anklicken erscheint ein Untermenü mit mehreren Elementen. Die aufgelisteten Elemente sind in Abbildung 6.29 dargestellt.

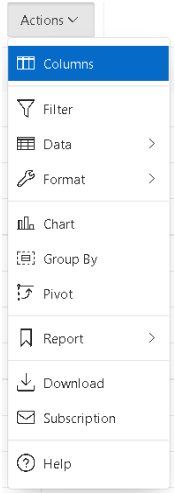



Abbildung 6.29: Aktionen-Schaltfläche.

Die Schaltfläche **Spalten** zeigt ein Feld an, das Sie in Abbildung 6.30 sehen können. Hier kann der Benutzer auswählen, welche Spalten im Bericht angezeigt werden sollen und in welcher Reihenfolge sie angezeigt werden sollen. Der Benutzer kann dies erreichen, indem er Spalten zwischen den beiden Abschnitten **Do not Display** und **Display in Report** verschiebt. Es gibt hilfreiche Pfeile, die die Arbeit mit dem Verschieben von Spalten erleichtern.

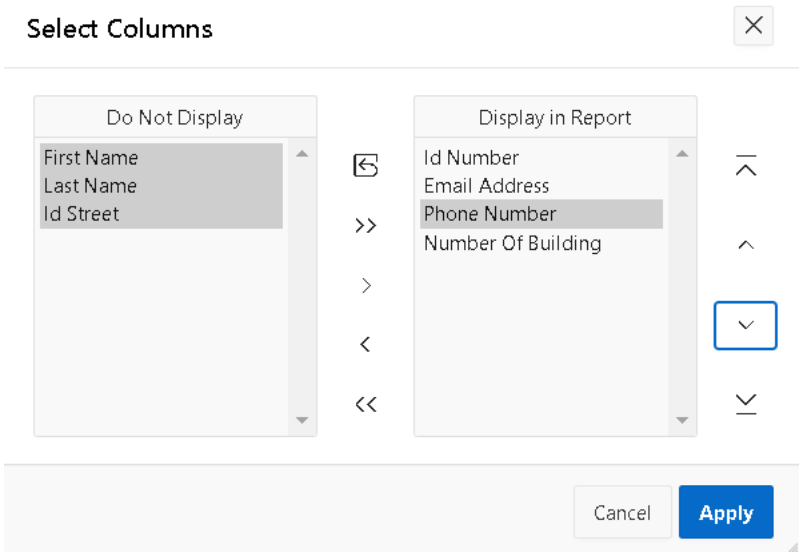



Abbildung 6.30: Auswahl der anzuzeigenden Spalten.

Als weitere Option gibt es den in Abbildung 6.31 gezeigten **Filter**. Wir können die Werte in der Spalte durch Ausdrücke wie *to be equal to, not equal, is null* usw. filtern. Sie können Ausdrücke definieren, indem Sie aus den vordefinierten Ausdrücken wählen oder Ihre eigenen schreiben.

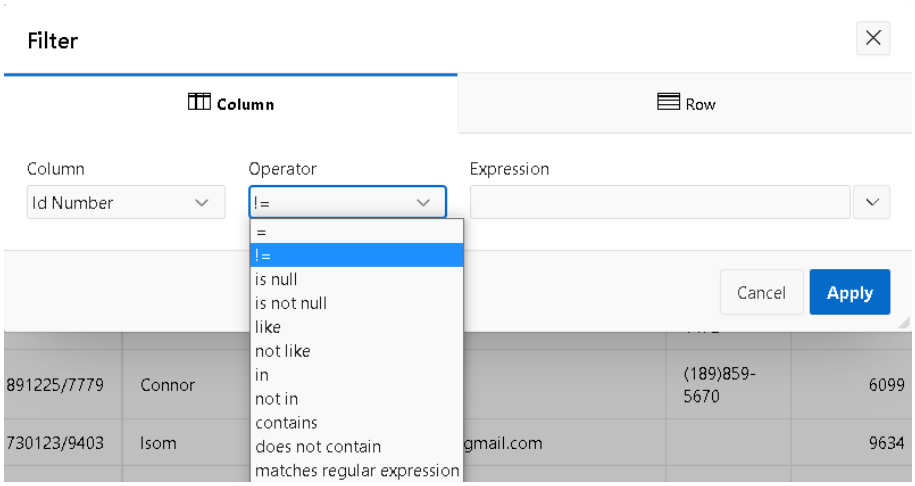



Abbildung 6.31: Spalten-Filter.

Beim Filtern von Zeilen haben wir die Wahl zwischen unzähligen Operationen und Ausdrücken, die wir zusammenstellen können. Dieser Filter erscheint im Fenster **Filter Expression**, das in Abbildung 6.32 dargestellt ist.

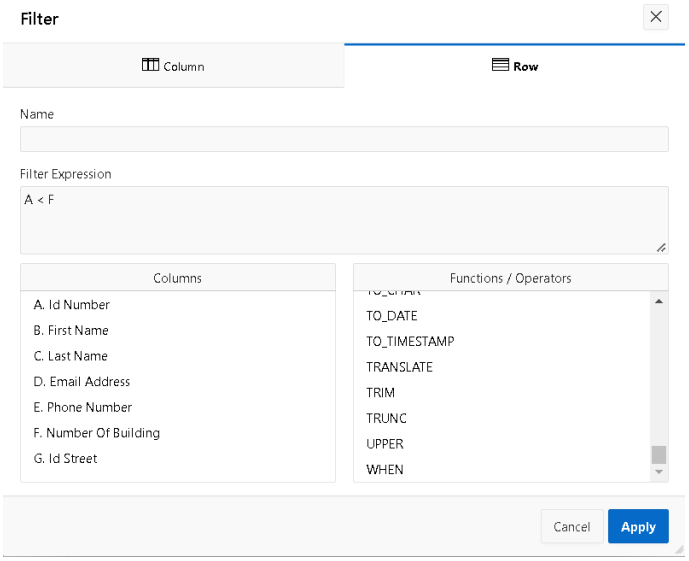



Abbildung 6.32: Zeilen-Filter.

Nachdem Sie auf die Schaltfläche **Data** geklickt haben, erscheint das in Abbildung 6.33 dargestellte Menü. Das Menü enthält vier weitere Möglichkeiten:

- sort,
- aggregate,
- compute,
- flashback.

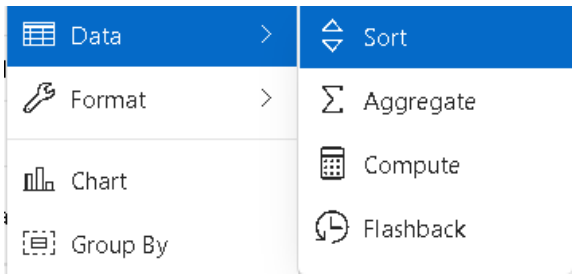



Abbildung 6.33: Nach Anklicken der Schaltfläche Data.

In Abbildung 6.34 sehen Sie die Option **Sort**, die auf jede Spalte angewendet werden kann, und der Benutzer kann festlegen, wie er mit *Nullwerten* arbeiten möchte.

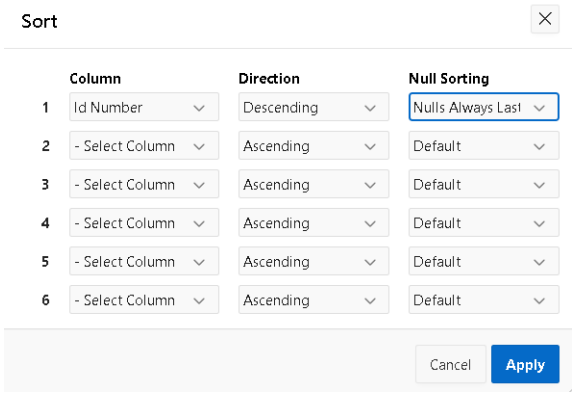



Abbildung 6.34: Sortieren.

Mit **Aggregate** können wir eine Aggregationsfunktion über eine der Spalten unseres Berichts ausführen. Die zur Auswahl stehenden Aggregationsfunktionen sind in Abbildung 6.35 dargestellt.

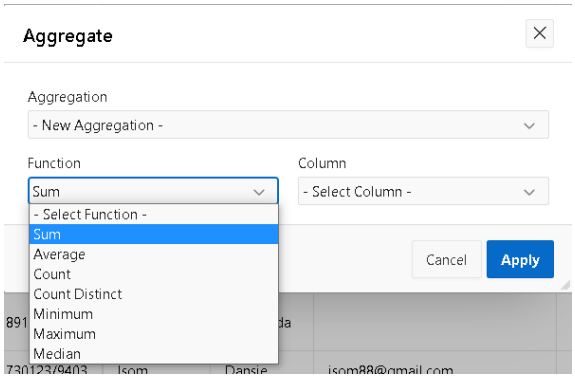



Abbildung 6.35: Aggregate.

Mit der Funktion **Flashback**, die in Abbildung 6.36 dargestellt ist, kann man zum Zustand des angezeigten Berichts zurückkehren, bevor er vor einigen Minuten geändert wurde.

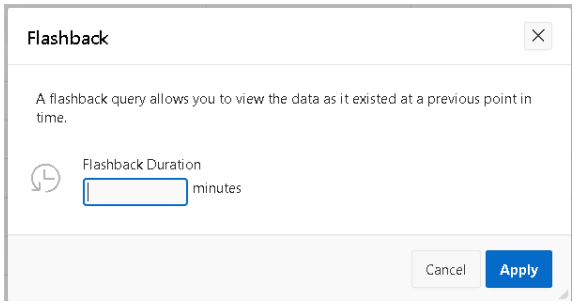



Abbildung 6.36: Flashback.

**Control Break**, wie in Abbildung 6.37 gezeigt, ermöglicht es uns, den Bericht in mehrere kleinere Einheiten zu unterteilen, als ob es sich um separate Berichte handeln würde. Diese kleinen Einheiten werden je nach Spaltenwerten unterteilt. Jede Einheit hat die gleichen Daten in der ausgewählten Spalte.

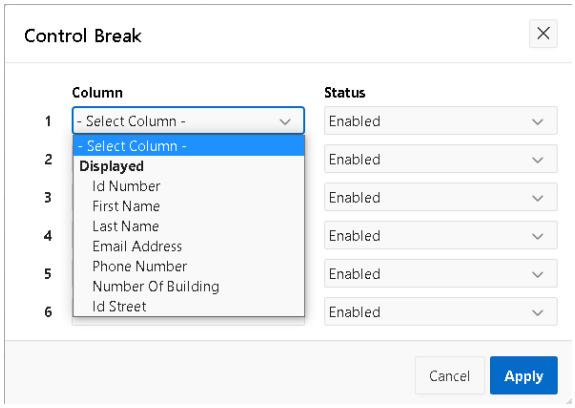



Abbildung 6.37: Control Break.

Abbildung 6.38 zeigt die Kontrollunterbrechung, die auf die Spalte *First Name* angewendet wird.

First Name: Aaban

| Id Number | Last Name | Email Address | Phone Number | Number Of Building | Id Street |
|---|---|---|---|---|---|
| 650724/0703 | Holmstedt | aaban4@gmail.com | (435)936-9323 | 4881 | 61479 |
| 640517/1114 | Sartor | aaban.sartor@gmail.com | | 8368 | 6494 |
| 921024/6657 | Kovari | aaban.kovari@gmail.com | (268)217-2739 | 1935 | 7728 |
| 740410/0666 | Dantico | aaban.dantico@gmail.com | | 3115 | 40604 |
| 870703/0560 | Labreck | aaban.labreck@gmail.com | (888)203-2292 | 2639 | 6159 |

Abbildung 6.38: Control Break-Resultat.

In Abbildung 6.39 sehen Sie die Funktion **Highlight**, die die Datensätze im Bericht transparent macht, ohne die anderen herauszufiltern. Wir können entweder eine Zelle oder die gesamte Zeile mit den von uns gewählten Farben hervorheben. Der Vergleich der Spaltenwerte funktioniert genauso wie bei **Filter**, nur dass ungefilterte Datensätze nicht aus dem Bericht verschwinden.

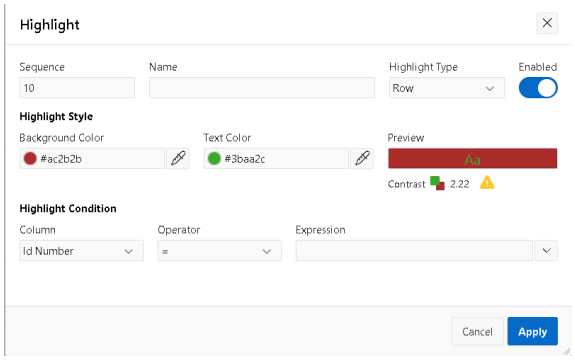



Abbildung 6.39: Highlight.

In Abbildung 6.40 sehen wir den Bericht mit hervorgehobenen Zeilen, wobei der Vorname gleich **"Aaban"** ist.

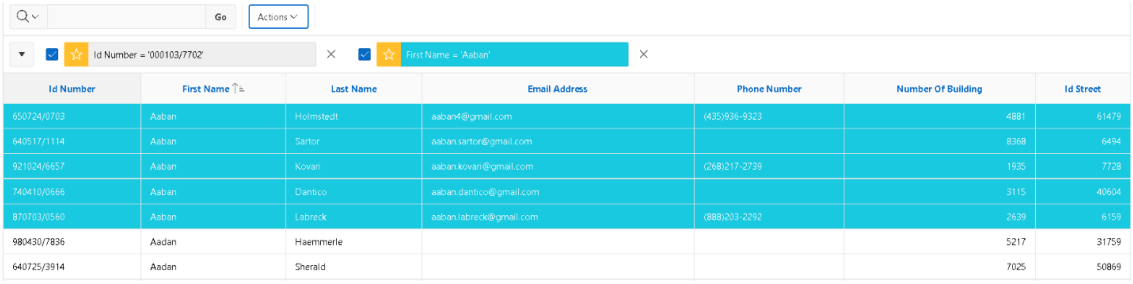



Abbildung 6.40: Highlight-Resultat.

Abbildung 6.41 zeigt **Rows Per Page**, ein Attribut, mit dem der Benutzer festlegen kann, wie viele Datensätze auf der Seite angezeigt werden sollen. Es ist eine ähnliche Einstellung wie die **Seitennummerierung**.

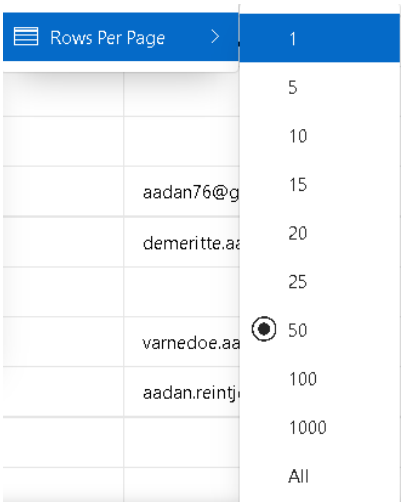



Abbildung 6.41: Row Per Page.

Wenn die Anzeige der Daten im Bericht nicht ausreicht, besteht auch die Möglichkeit, diese Daten über das in Abbildung 6.42 dargestellte Element **Chart** in ein Diagramm zu projizieren. Es stehen **vier** Basisgraphen zur Auswahl. Wir können auch eine Aggregationsfunktion wählen, die auf die gegebene Spalte angewendet wird.

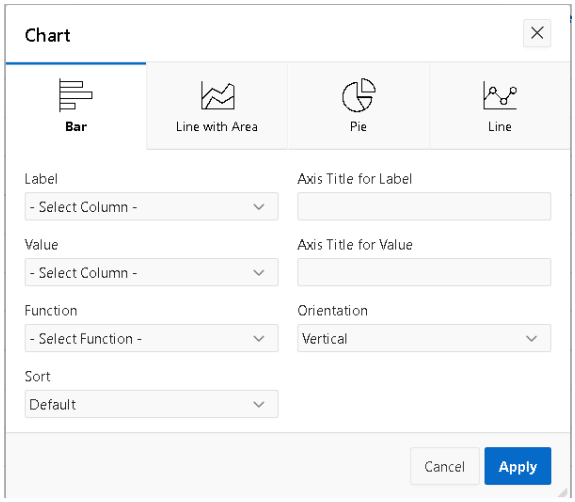



Abbildung 6.42: Diagramm im interaktiven Bericht.

Eine weitere sehr wichtige Funktion des interaktiven Berichts ist die in Abbildung 6.43 gezeigte Funktion **Group By**. Sie funktioniert auf die gleiche Weise wie bei einer SQL-Abfrage.

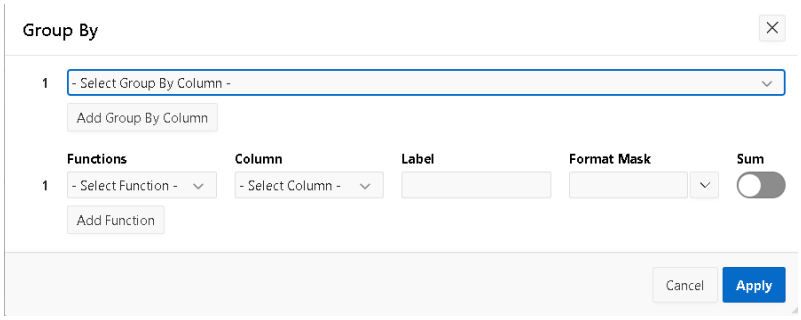



Abbildung 6.43: Group By.

**Save Report**, wie in Abbildung 6.44 dargestellt, speichert den Bericht in seinem aktuellen Zustand. Dieser Bericht ist jedoch nur für den Entwickler sichtbar, nachdem er sich bei der Anwendung angemeldet hat, und nicht für den Benutzer.

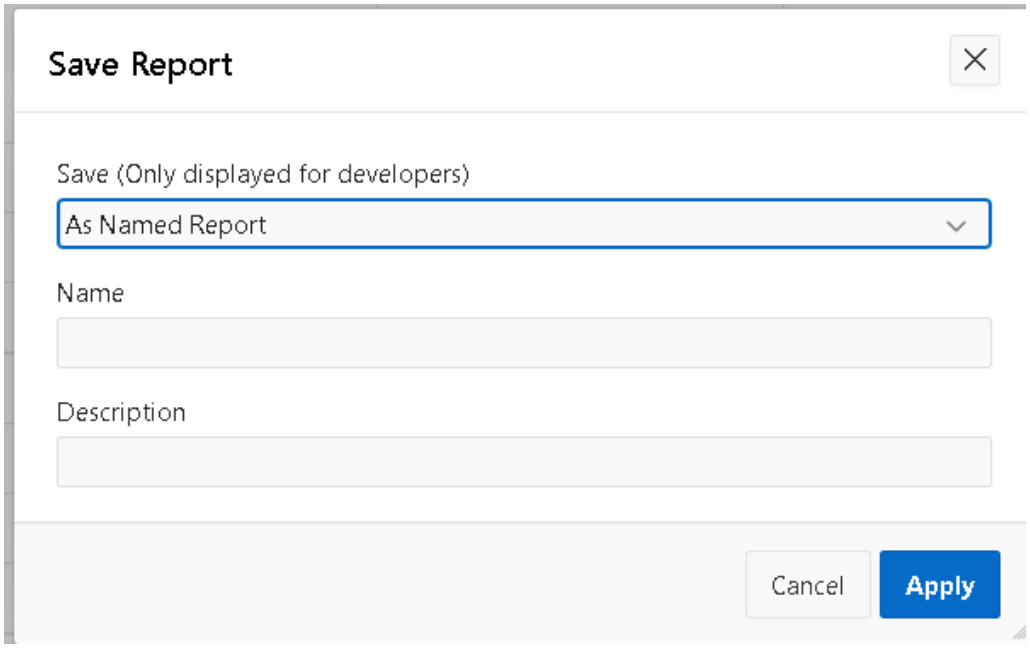



Abbildung 6.44: Save Report.

Wir können den Bericht auch in seinen ursprünglichen Zustand zurücksetzen oder ihn im ausgewählten Format herunterladen. Diese Option ist in Abbildung 6.45 zu sehen.

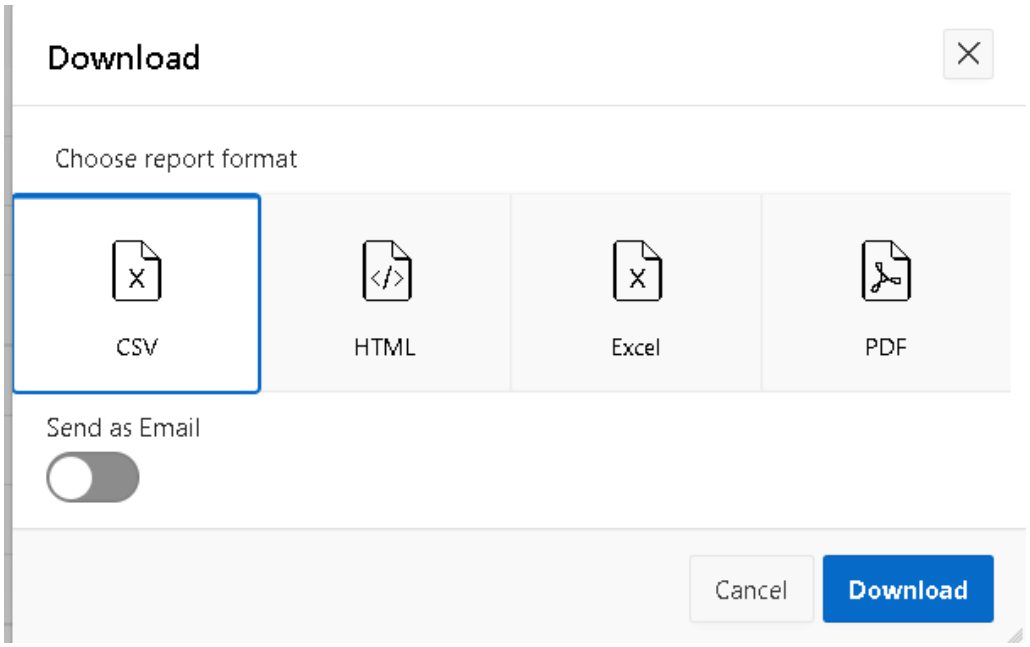



Abbildung 6.45: Herunterladen.

## 6.5 Fragen

1. Zu welcher Kategorie gehört der Berichtstyp? Regionen, Elemente oder Schaltflächen?
2. Was kann die Datenquelle eines Berichts sein?
3. Welche Quellenarten können für die Angabe eines klassischen Reports verwendet werden?
4. Was ist das Hauptmerkmal des klassischen Reports?

5. Welche Technologie ermöglicht es dem Benutzer, mit dem Interaktiven Bericht das Datenbild zu erhalten, wie es in der Vergangenheit existierte?

## 6.6 Antworten

1. Ein Bericht ist eine Region, die das Browsen und Suchen von Daten vereinfacht.
2. Die Daten können aus einer lokalen Datenbank, einer REST-fähigen SQL- oder REST-Quelle geladen werden.
3. Verfügbare Quellentypen sind Tabellen oder SQL-Querys.
4. Das Hauptmerkmal des klassischen Berichts ist der schreibgeschützte Modus, der es dem Benutzer nicht erlaubt, die Anzeigeoptionen oder Daten zu ändern.
5. Flashback-Funktionen ermöglichen es dem Benutzer, das Datenbild so zu erstellen, wie es zu einem bestimmten Zeitpunkt vorlag.

# 7. Wie verwaltet man Formulare?

VERONIKA ŠALGOVÁ, MICHAL KVET, MIROSLAV POTOČÁR; ÜBERSETZER W. SCHWINGER

## 7.1 Wie verwaltet man Formulare?

In Oracle APEX können wir drei grundlegende Formulartypen verwenden:

- ein bearbeitbares interaktives Raster,
- ein Formular auf einer Tabelle,
- ein Master-Detail-Formular.

## 7.2 Bearbeitbares Interaktives Raster

In dem bearbeitbaren interaktiven Bericht können Sie Daten hinzufügen, ändern, löschen und suchen. Es handelt sich um eine funktionsreiche Komponente mit Datenbearbeitungsfunktionen, die direkt auf der Seite verfügbar sind. Darüber hinaus kann dieser Bericht interaktiv mit der Maus angepasst und neu angeordnet werden, wobei die Struktur und der Inhalt eines Rasters direkt aktualisiert werden. Es ist möglich, ein bearbeitbares interaktives Raster als Konvertierung eines schreibgeschützten interaktiven Rasters zu erstellen, das durch Auswahl von *Report* und dann *Interactive Grid* erstellt wurde. Eine andere Möglichkeit besteht darin, *Report and Form* zu erstellen und als Berichtstyp *Interactive Grid* auszuwählen, das ebenfalls als schreibgeschützte Version erstellt wird, aber in ein bearbeitbares Formular umgewandelt werden kann.

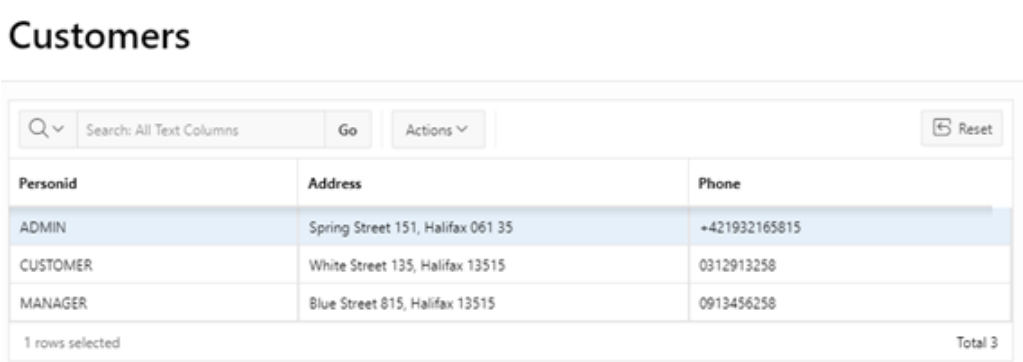



Abbildung 7.1: Interaktives Raster mit Kunden

Die Daten können mit Hilfe des Textes in der Suchleiste oben auf der Seite durchsucht werden. Andere Aktionen, die mit diesem interaktiven Gitter möglich sind, werden in Abbildung 7.2 gezeigt.

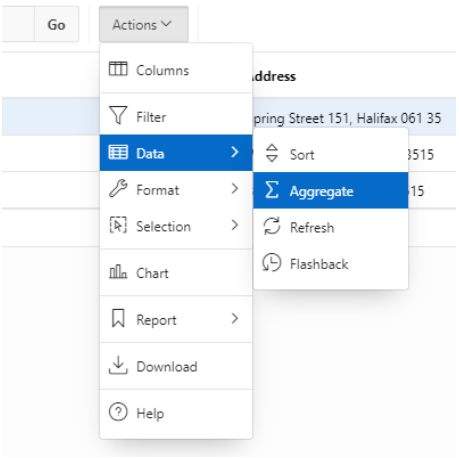



Abbildung 7.2: Aktionen des interaktiven Rasters

Sie können die Größe einer Rasterspalte ändern, indem Sie auf die Kante einer Spaltenüberschrift klicken, sie gedrückt halten und mit der Maus verschieben. Sie können eine Spalte auch ausblenden, indem Sie die Überschrift auswählen und auf das *Ausblenden*-Symbol klicken, wie in Abbildung 7.3 gezeigt.

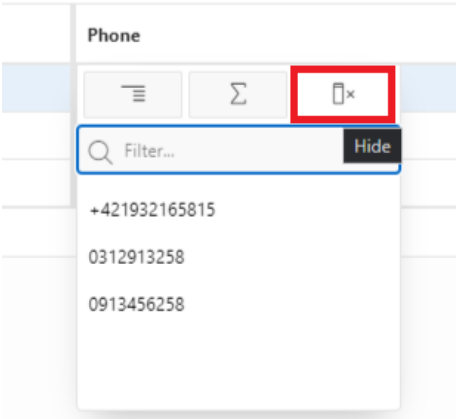



Abbildung 7.3: Ausblenden-Symbol der Spalte Telefon

Nachdem die Spalte ausgeblendet wurde, wird sie nicht mehr in der Tabelle angezeigt. Um die Spalte wieder anzuzeigen, müssen Sie auf *Actions* klicken, die Option *Columns* wählen und auf die anzuzeigenden Spalten klicken.

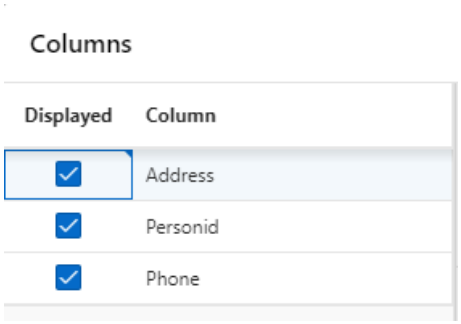



Abbildung 7.4: Angezeigte Spalten

Eine Spalte kann durch Klicken auf das *Freeze*-Symbol eingefroren werden, wie in Abbildung 7.5 gezeigt. Durch das Einfrieren einer Spalte wird diese aus dem rollbaren Bereich ausgeschlossen.

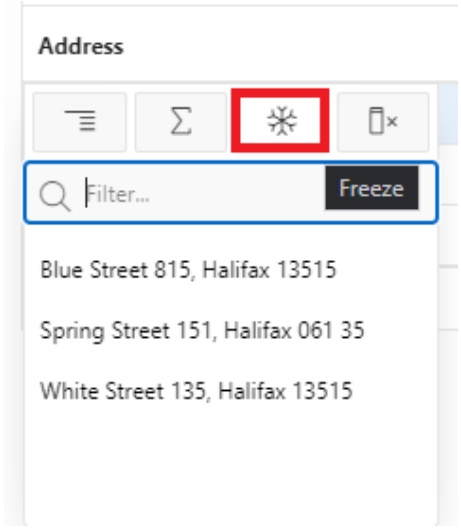



Abbildung 7.5: Freeze-Symbol der Adressspalte

Daten können nach einer Spalte sortiert werden, indem Sie das Symbol *Sort Ascending* oder *Sort Descending* auswählen, wie in Abbildung 7.6 gezeigt.

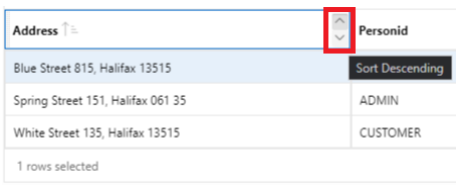



Abbildung 7.6: Sortieren der Adressspalte

Mithilfe des Symbols *Aggregate* der Spalte, wie in Abbildung 7.7 gezeigt, können Sie für die ausgewählte Spalte zwischen den Funktionen Count, Count Distinct, Minimum und Maximum wählen.

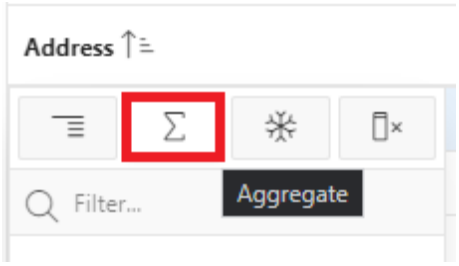



Abbildung 7.7: Aggregate-Symbol der Adressspalte

Daten aus dem Raster können zur Erstellung von Diagrammen verwendet werden, indem Sie die Optionen *Actions* und dann *Charts* wählen. Es gibt verschiedene Arten von Diagrammen, wie in Abbildung 7.8 gezeigt. Sie können aus einer Reihe von Aggregationsfunktionen wählen, die in den Diagrammen enthalten sind.

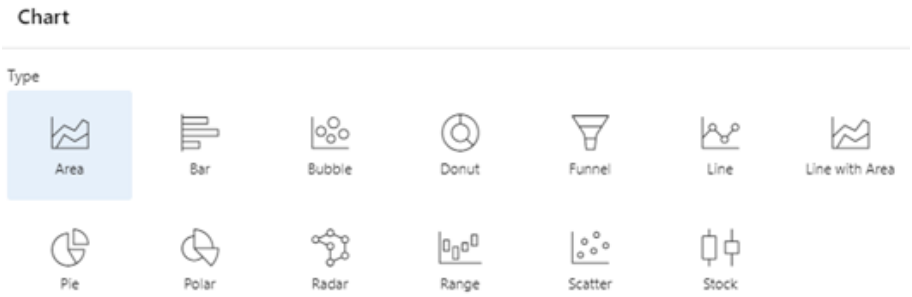



Abbildung 7.8: Arten von Diagrammen

## 7.3 Formular auf einer Tabelle

Oracle APEX stellt uns grundlegende Formulare zum Einfügen einer einzelnen Zeile in eine Tabelle zur Verfügung, wie in Abbildung 7.9 dargestellt. Nachdem Sie auf die Schaltfläche *Create a Page* geklickt haben, müssen Sie eine *Formular*-Option wählen. Sie müssen die Tabelle auswählen, die mit dem erstellten Formular verbunden ist, oder Sie können Ihre SQL-Abfrage und auch eine Primärschlüsselspalte der Tabelle schreiben.




Abbildung 7.9: Formular zum Einfügen von Kunden

## 7.4 Master-Detail-Formular

Es ist möglich, Werte aus zwei verbundenen Tabellen oder Ansichten abzufragen, einzufügen, zu aktualisieren und zu löschen. Nachdem Sie auf die Schaltfläche *Create a Page* geklickt haben, müssen Sie eine *Master-Detail*-Option wählen. Im folgenden Detail wählen Sie einen Master-Detail-Stil, z. B. *Stacked*, *Side by Side* oder *Drill Down*. Ein Master-Detail-Formular zeigt normalerweise eine Master- und mehrere Detailzeilen an. Das bedeutet, dass Sie eine Tabelle oder eine Ansicht als Master-Datenquelle und eine andere als Detail-Datenquelle auswählen müssen. Nach der Auswahl des Primärschlüssels jeder Tabelle müssen Sie den Master-Detail-Fremdschlüssel auswählen.

Abbildung 7.10 zeigt ein *gestapeltes* Master-Detail-Formular. Es enthält bearbeitbare interaktive Raster. Die Benutzer können eine Zeile im Master-Raster auswählen, um die Detail-Raster zu aktualisieren.

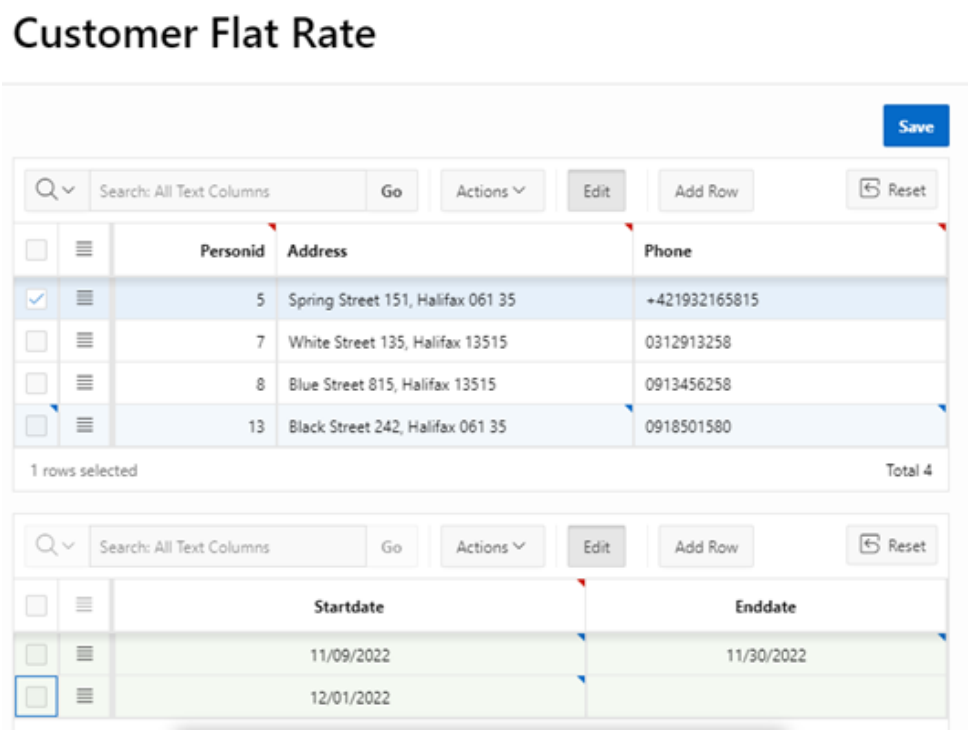



Abbildung 7.10: Gestapeltes Master-Detail-Formular der Kundenpauschalen

In Abbildung 7.11 ist ein *Side by Side*-Master-Detail-Formular zu sehen, das ein einseitiges Master-Detail enthält, welches ein Side-by-Side-Layout und Berichtsbereiche mit modalen Bearbeitungsfenstern verwendet. Auf der linken Seite können Sie aus einer Master-Liste auswählen, um zum Master-Satz zu navigieren. Die rechte Seite enthält den ausgewählten Master-Satz und die zugehörigen Detail-Berichte.

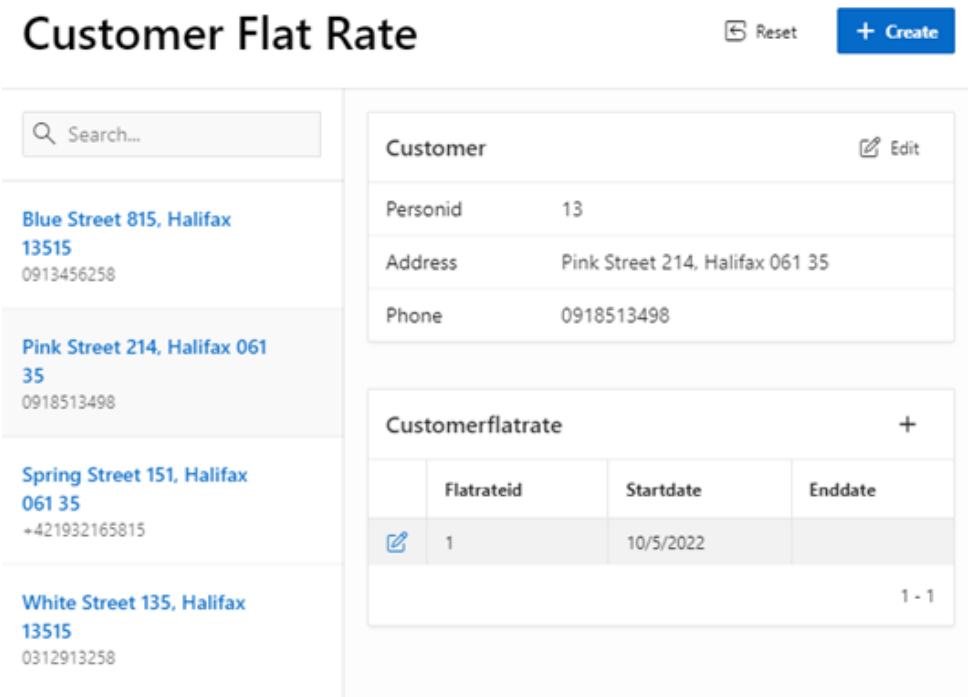



Abbildung 7.11: Side by Side Master-Detail-Formular der Kundenpauschalen

Ein *Drill Down* Master-Detail enthält zwei Seiten. Die erste Seite enthält einen interaktiven Bericht für die Master-Tabelle, wie in Abbildung 7.12 dargestellt. Die zweite Seite, wie in Abbildung 7.13 gezeigt, enthält ein Standardformular für die Master-Tabelle und bearbeitbare interaktive Raster für die Detail-Tabelle.

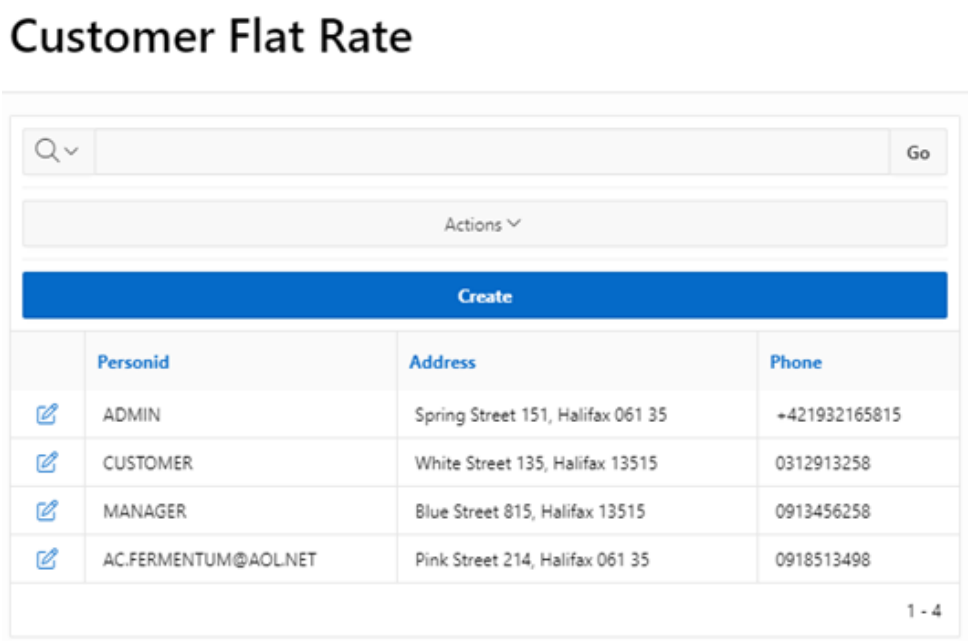



Abbildung 7.12: Drill Down Master-Detail Formular – erste Seite

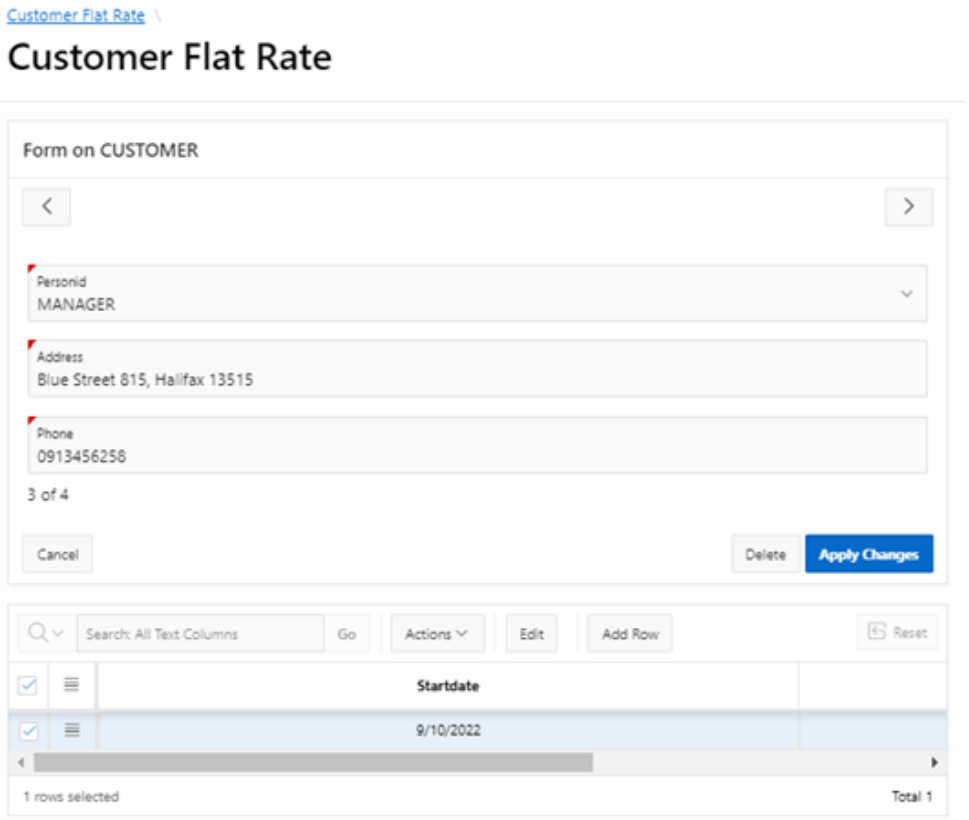



Abbildung 7.13: Drill Down Master-Detail Formular – zweite Seite

## 7.5 Fragen

1. Ist es möglich, die Daten in einem editierbaren interaktiven Raster anzuordnen?
2. Welche Tabellenkomponentendefinition muss ausgewählt werden, wenn ein Formular erstellt wird?
3. Welche Option schließt eine Spalte aus einem scrollbaren Bereich aus?

## 7.6 Antworten

1. Ja, die Berichte können interaktiv mit der Maus angepasst und neu angeordnet werden, wobei die Struktur und der Inhalt eines Rasters direkt aktualisiert werden.
2. Primärschlüsselattribute müssen angegeben werden.
3. Eine Spalte kann durch Klicken auf das Symbol Einfrieren eingefroren werden. Durch das Einfrieren einer Spalte wird diese aus dem scrollbaren Bereich ausgeschlossen.

# 8. Wie wandelt man Textberichte in Diagramme um?

IVAN PASTIERIK, MICHAL KVET, MIROSLAV POTOČÁR

## 8.1 Diagramm

Diagrammregionen in Oracle APEX sind vielseitige Komponenten, die es Entwicklern und Anwendern gleichermaßen ermöglichen, durch visuelle Darstellung tiefere Einblicke in Daten zu gewinnen. Diese dynamischen Diagramme, die von Balken- und Liniendiagrammen bis hin zu Kuchendiagrammen und mehr reichen, dienen als wichtige Hilfsmittel zur Verbesserung der Benutzererfahrung und zur Erleichterung der datengesteuerten Entscheidungsfindung.

In diesem Kapitel lernen Sie, wie Sie ein einfaches Diagramm mit den Funktionen **Filtern** und **Sortieren** erstellen können. **Filterung** und **Sortierung** können die Interaktivität und die Lesbarkeit von Diagrammen erheblich verbessern. Zu diesem Zweck haben wir ein einfaches Datenmodell erstellt, wie in Abbildung 8.1 dargestellt. Das Datenmodell besteht aus zwei Entitäten: **„product“** und **„sale_transaction“**. Die Entität **„product“** stellt ein einzelnes Produkt dar, dessen Primärschlüsselattribut **„product_id“** ist und dessen Name im Attribut **„product_name“** gespeichert ist. Die Entität **„sale_transaction“** speichert Informationen über alle verkauften Produkte mit dem Primärschlüssel **„sale_id“**. Das Attribut **„product_id“** ist ein Fremdschlüssel, der auf das Attribut **„product_id“** in der Entität **„product“** verweist. Das Attribut **„sell_date“** enthält das Datum, an dem das Produkt verkauft wurde.

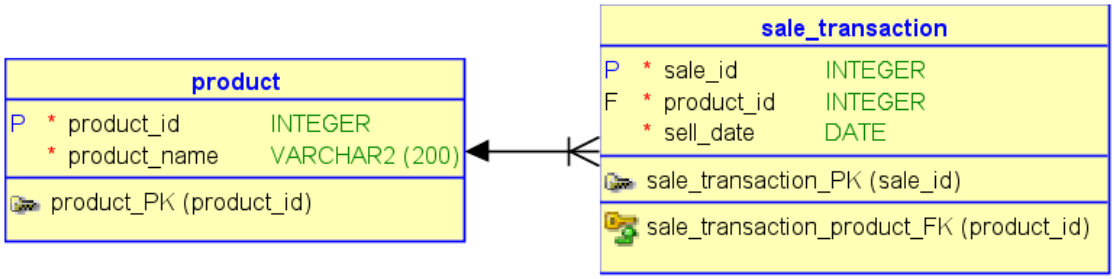



Abbildung 8.1: In der Beispielanwendung verwendetes Datenmodell.

Zu Beginn muss eine neue Anwendung erstellt werden, die wir **„Chart Showcase“** nennen werden. Diese Anwendung wird die Seiten **„Global Page“**, **„Home“** und **„Login Page“** enthalten. Wir werden Diagramme auf der Homepage erstellen. Detaillierte Anweisungen zum Erstellen von

Anwendungen wurden in Kapitel 5 beschrieben. Nachdem Sie die Anwendung erstellt haben, geben Sie den **Seitendesigner** für die Seite **„Home“** innerhalb unserer neu erstellten Anwendung ein.

## 8.2 Balkendiagramm erstellen

Um den in Abbildung 8.2 hervorgehobenen Diagrammbereich in unserer Anwendung zu platzieren, müssen wir ihn per Drag & Drop in den Layoutbereich des **Page Designers** ziehen.

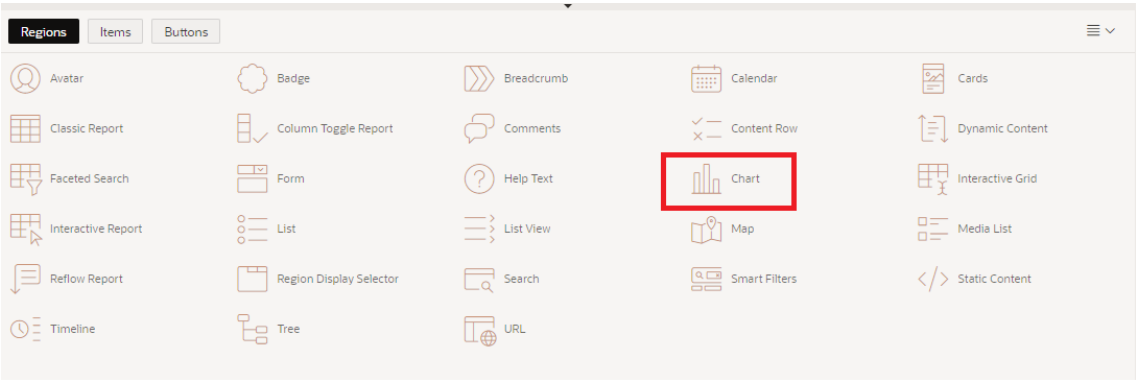



Abbildung 8.2: Selecting and placing Chart region.

Wir platzieren den Bereich **Chart** im Abschnitt **body** unserer Seite, da wir möchten, dass unser Diagramm der Hauptinhalt der Seite ist. Wir setzen den **title** unseres neu platzierten Diagrammbereichs auf **„Sales Chart“**. In Abbildung 8.3 sehen Sie, wie unser Hauptteil des Layouts nach dem Platzieren und Ändern des Titels unseres Bereichs **Chart** aussieht.

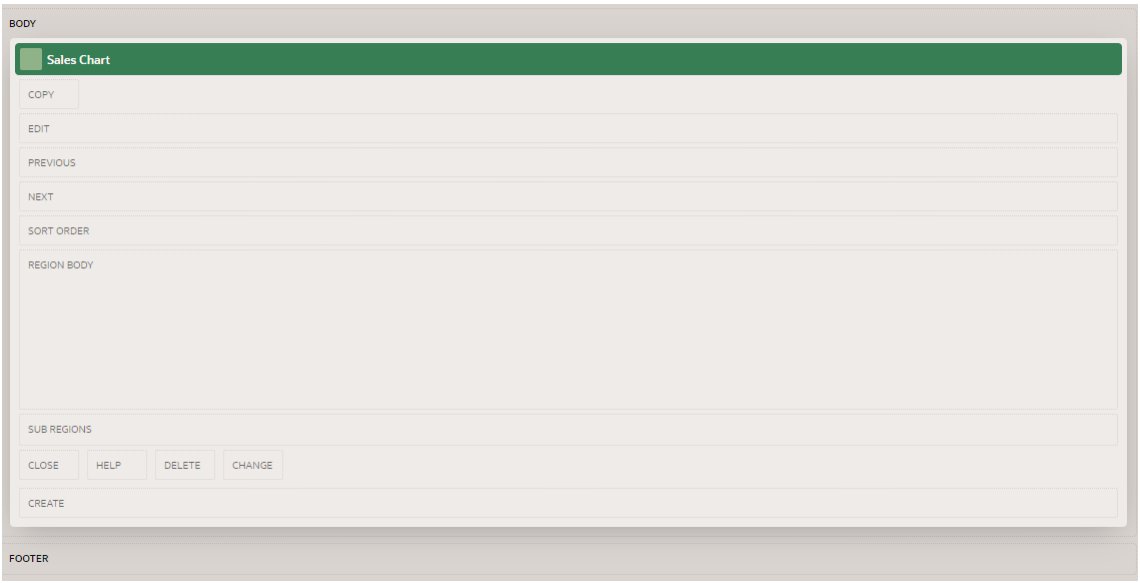



Abbildung 8.3: Layout des Hauptteils nach der Platzierung des Diagrammbereichs.

Im linken Teil des **Seitendesigners** sehen Sie eine **Baumübersicht** über alle unsere platzierten Komponenten. Wenn Sie sich unsere Komponente **Sales Chart** innerhalb von **„Series“** genau ansehen, wird ein Fehler angezeigt. Dieser Fehler, der in Abbildung 8.4 zu sehen ist, wird dadurch verursacht, dass wir die Quelle für die Daten, die wir in unserem Diagramm anzeigen möchten, nicht definiert haben.

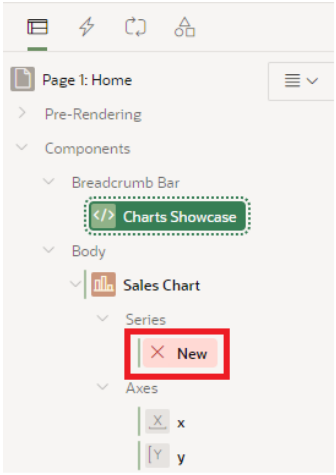



Abbildung 8.4: Baumübersicht mit Fehler in Diagrammreihen.

Wenn Sie in der Baumübersicht auf die Reihe mit der Bezeichnung **„New“** klicken, wird das Einstellungsmenü für die Reihe auf der rechten Seite des **Seitendesigners** geöffnet. In diesem Einstellungsmenü ändern wir den Namen der Reihe in **„sales“**, da wir die Anzahl der Verkäufe unserer einzelnen Produkte in unserem **Umsatzdiagramm** anzeigen möchten. Als Nächstes müssen wir die Reihe **source** definieren. Dazu ändern wir den Quellentyp in **„SQL Query“** und platzieren innerhalb der SQL Query eine **SELECT**-Anweisung, wie in Abbildung 8.5 gezeigt. Diese **SELECT**-Anweisung gibt die Gesamtzahl der Verkäufe gruppiert nach Produktnamen zurück. Nachdem wir nun die Quelle für unsere Daten definiert haben, die wir im Diagramm anzeigen möchten, müssen wir die von der Anweisung **SELECT** ausgewählten Spalten den Achsen **Label (X)** und **Value (Y)** zuordnen. Die endgültige Konfiguration unserer Reihe ist in Abbildung 8.5 dargestellt, wobei die wichtigsten Änderungen hervorgehoben sind.

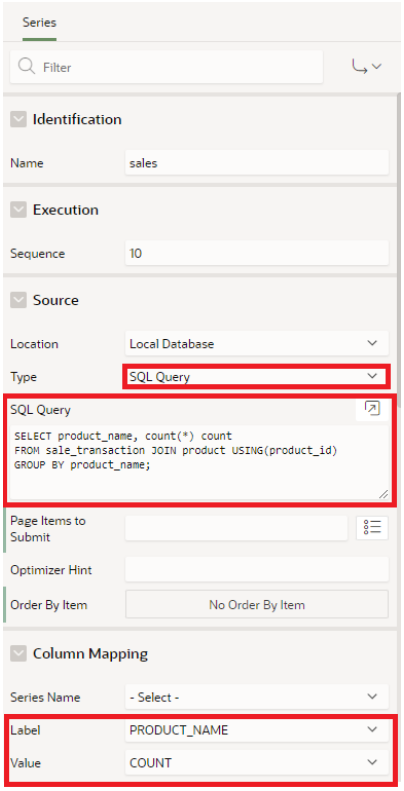



Abbildung 8.5: Endgültige Konfiguration der Verkaufsreihe

Wenn wir die SQL-Abfrage aus Abbildung 8.5 nehmen und sie ausführen, werden die Daten wie in Abbildung 8.6 gezeigt zurückgegeben.

| PRODUCT_NAME | COUNT |
|---|---|
| Smartwatch | 68 |
| Monitor | 67 |
| Keyboard | 66 |
| Drone | 67 |
| Virtual Reality Headset | 62 |
| Printer | 64 |
| Camera | 74 |
| Laptop | 59 |
| Tablet | 65 |
| PC | 72 |

Abbildung 8.6: Durch die Ausführung der SQL-Abfrage zurückgegebene Daten.

Jetzt können wir unsere Seite speichern und ausführen. Vergessen Sie nicht, die **Authentifizierung** der Seite auf **„Page Is Public“** zu setzen, damit keine Anmeldung erforderlich ist. Wenn Sie die Seite ausführen, sollten Sie die in Abbildung 8.7 gezeigten Ergebnisse erhalten. Sie werden jedoch feststellen, dass unser Diagramm keine beschrifteten **X**- und **Y**-Achsen hat, wodurch es schwerer zu lesen ist.

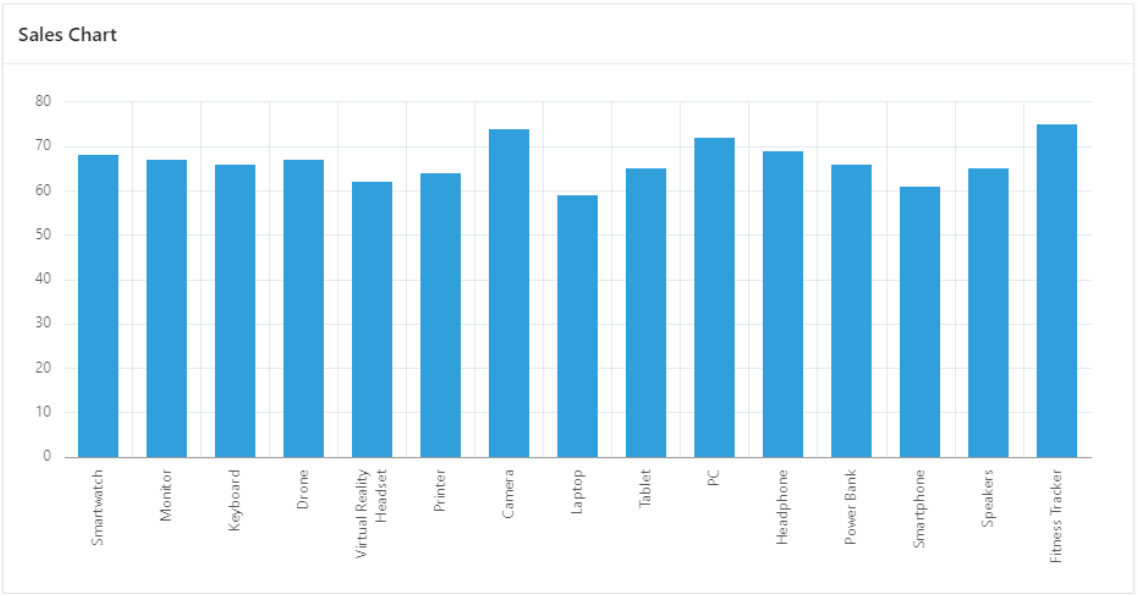



Abbildung 8.7: Beim Starten der Anwendung angezeigtes Diagramm.

Um dies zu beheben, können wir unsere Achsen auf ähnliche Weise benennen, wie wir unsere Reihen benannt haben. Indem wir in der linken Baumübersicht auf einzelne Achsen klicken und ihr Attribut **Title** ändern, wie in Abbildung 8.8 gezeigt, können wir die **X**-Achse **„Products“** und die **Y**-Achse **„Sales Count“** nennen.

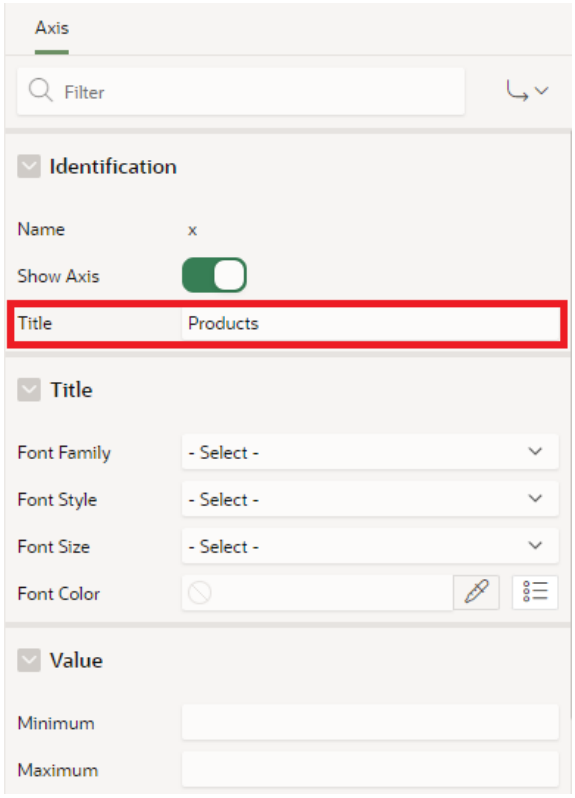



Abbildung 8.8: Ändern des Titelattributs für die X-Achse.

Nachdem wir beide Achsen unseres Diagramms beschriftet haben, sollte das Ergebnis dem in Abbildung 8.9 gezeigten Diagramm ähneln.

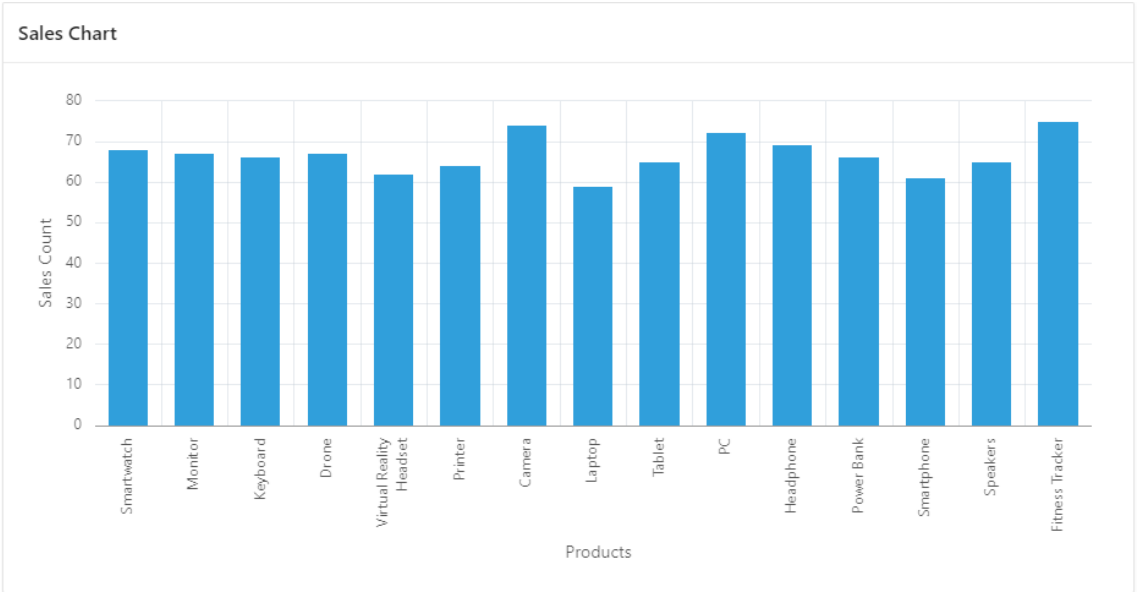



Abbildung 8.9: Umsatzdiagramm nach Beschriftung der Achsen.

## 8.3 Filterung zum Balkendiagramm hinzufügen

Fügen wir nun **Filterung** für Produkte zu unserem Diagramm hinzu, so dass wir auswählen können, welche Produkte angezeigt werden sollen. Dazu fügen wir ein Element **Checkbox Group** aus der unteren Symbolleiste des **Seitendesigners** hinzu und platzieren es direkt über unserem Bereich **Sales Chart** im Textkörper.

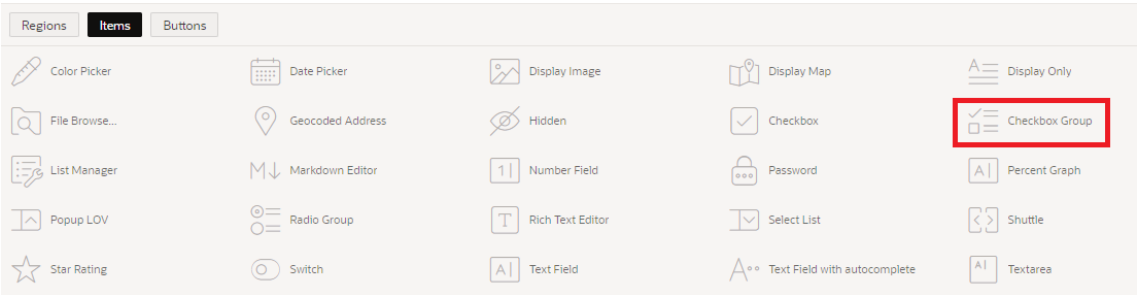



Abbildung 8.10: Auswählen und Platzieren von Checkbox-Gruppen-Elementen.

Wir nennen das **Checkbox Group** Element **„P1_PRODUCTS_CHOICE“** und setzen seine Bezeichnung auf **„Produktauswahl“**. Außerdem können wir das Layout so anpassen, dass es in drei Spalten angezeigt wird, wie in Abbildung 8.11 zu sehen ist.

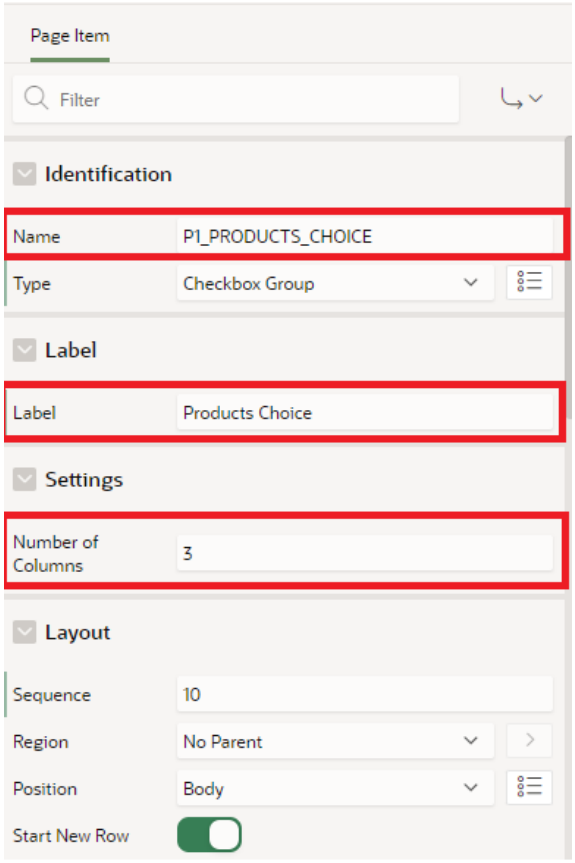



Abbildung 8.11: Einstellen der grundlegenden Attribute des Checkbox Group Elements.

Um unsere **Checkbox Group** mit Produkten zu füllen, müssen wir eine **Werteliste** definieren. Im Abschnitt **„List of Values“** der **Seitenelement** Einstellungen ändern wir das Attribut **„Type“** in **„SQL Query“** und schreiben eine **SELECT**-Anweisung, wie in Abbildung 8.12 gezeigt. Diese Anweisung sollte genau zwei Spalten auswählen: die erste Spalte mit den Anzeigewerten (mit dem Alias **„d“**) und die zweite Spalte mit den Rückgabewerten (mit dem Alias **„r“**).

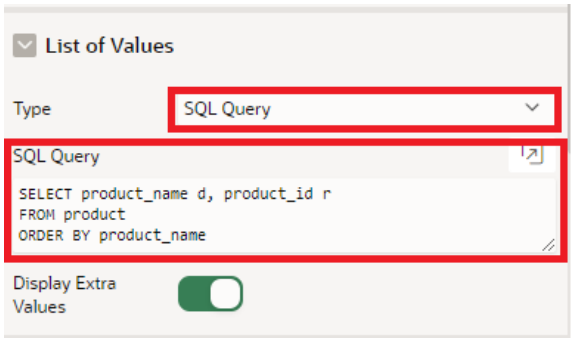



Abbildung 8.12: Einstellen der Werteliste für das Checkbox Group Element.

Standardmäßig sind bei **Checkbox Group** alle Kontrollkästchen nicht markiert, wenn Sie die Anwendung ausführen. Für eine bessere Benutzerfreundlichkeit bei der Verwendung mit Diagrammen ist es jedoch wünschenswert, dass die Benutzer standardmäßig alle Produkte im Diagramm angezeigt bekommen. Um dies zu erreichen, gehen wir zum Abschnitt **„Default“** der Einstellungen des **Checkbox Group** Elements und ändern das Attribut **„Type“** in **„SQL Query returning Colon Delimited List“**. Mit dieser Art von SQL-Abfrage können wir festlegen, welche

Werte standardmäßig ausgewählt werden. Die reguläre SQL-Abfrage kann nicht mit dem **Checkbox Group** Element verwendet werden, da sie nur einen Standardwert und nicht mehrere Standardwerte definieren kann. Innerhalb des **„SQL Query Colon“** schreiben wir eine **SELECT**-Anweisung, wie in Abbildung 8.13 gezeigt. Diese **SELECT**-Anweisung sollte genau eine Spalte zurückgeben, die intern in eine **Colon Delimited List** umgewandelt wird, die wie folgt aussehen könnte: **„1:5:7:2:3“**.

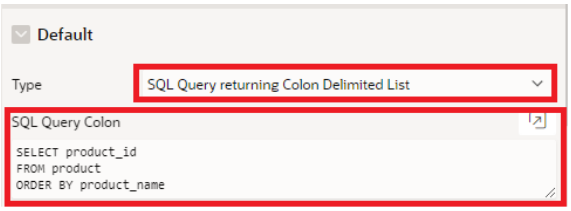



Abbildung 8.13: Einstellen von Standardwerten für das Checkbox Group Element.

Jetzt, wo das **Checkbox Group** Element nach unseren Vorstellungen konfiguriert ist, können wir es mit dem **Sales Chart** Bereich verknüpfen. Dazu fügen wir den Namen des **Checkbox Group** Elements in das Feld **„Page Items to Submit“** im Abschnitt **„Source“** der Reihe **sales** im **Sales Chart** Bereich ein. Es ist **erforderlich**, jedes Element, das in der SQL-Abfrage verwendet wird, in **„Page Items to Submit“** hinzuzufügen. Der Rückgabewert von **„P1_PRODUCTS_CHOICE“** ist eine **Colon Delimited List**, die aus Produkten besteht, die überprüft werden. Daher müssen wir die Funktion **INSTR** verwenden, die den Index des ersten Vorkommens einer im zweiten Parameter der Funktion definierten Teilzeichenkette innerhalb der im ersten Parameter der Funktion definierten Zeichenkette zurückgibt. Die Funktion **INSTR** gibt **0** zurück, wenn die Zeichenfolge nicht vorkommt. Wir verwenden die Funktion **INSTR**, um zu prüfen, ob die **„product_id“** jedes Produkts in der ausgewählten Liste der Produkt-IDs in der **„P1_PRODUCTS_CHOICE“** vorhanden ist. Die Funktion **INSTR** sucht nach dem Vorkommen von **':'||product_id||':'** innerhalb der verketteten Zeichenfolge **':'||:P1_PRODUCTS_CHOICE||':'**. Wenn eine Übereinstimmung gefunden wird, bedeutet dies, dass das Produkt zu den ausgewählten Produkten gehört. Wir müssen **':'** sowohl mit **„product_id“** als auch mit **„:P1_PRODUCTS_CHOICE“** verketten, weil wir einen genauen Abgleich zwischen den einzelnen Produkt-IDs vornehmen wollen. Auf diese Weise stellen wir sicher, dass zum Beispiel **„:1: “** mit **„:1:2:3:8:9:11: “** übereinstimmt. Auf diese Weise wird sichergestellt, dass wir nach exakten Übereinstimmungen suchen und dass jede ausgewählte Produkt-ID unabhängig betrachtet wird.

Wenn wir **“1”**, mit **“1:2:3:8:9:11”**, abgleichen, könnten wir möglicherweise ein Produkt mit der ID **“1”**, zeigen, wenn Kontrollkästchen für Produkte mit den IDs **“1”**, **“11”**, oder andere IDs, die den Wert **“1”**, enthalten, markiert sind. Mit anderen Worten, wenn wir das Produkt mit der ID **“11”**, prüfen, würde nicht nur das ausgewählte Produkt angezeigt werden, sondern auch das Produkt mit der ID **“1”**, Dies führt zu unerwünschten Ergebnissen und einer ungenauen Datendarstellung.

Durch das Hinzufügen von **':'** auf beiden Seiten der IDs wird ein konsistentes Begrenzungsmuster erstellt, das sicherstellt, dass die ausgewählten Produkt-IDs genau verglichen werden und jegliche Mehrdeutigkeit im Abgleichsprozess beseitigt wird. Auf diese Weise wird sichergestellt, dass das Diagramm die Daten auf der Grundlage der vom Benutzer getroffenen Auswahlen korrekt anzeigt.

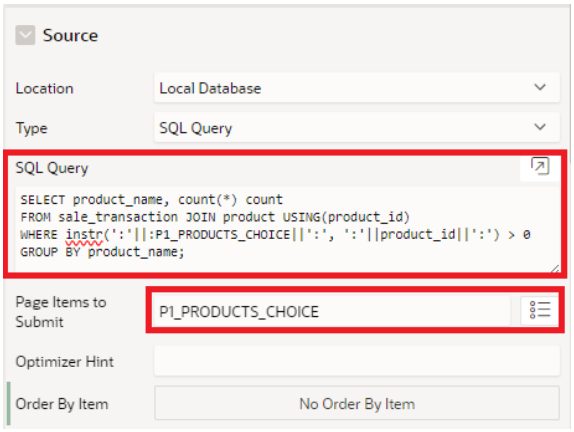



Abbildung 8.14: Verknüpfung der Verkaufsdiagrammregion mit dem Checkbox Group Element.

Der letzte Konfigurationsschritt besteht darin, die Aktualisierung unseres **Sales Chart** Bereichs einzurichten, wenn sich die Auswahl des **Checkbox Group** Elements ändert. Um dies zu erreichen, müssen wir eine **Dynamische Aktion** für das **„P1_PRODUCTS_CHOICE“** Checkbox-Gruppen Element erstellen, wie in Abbildung 8.15 gezeigt.

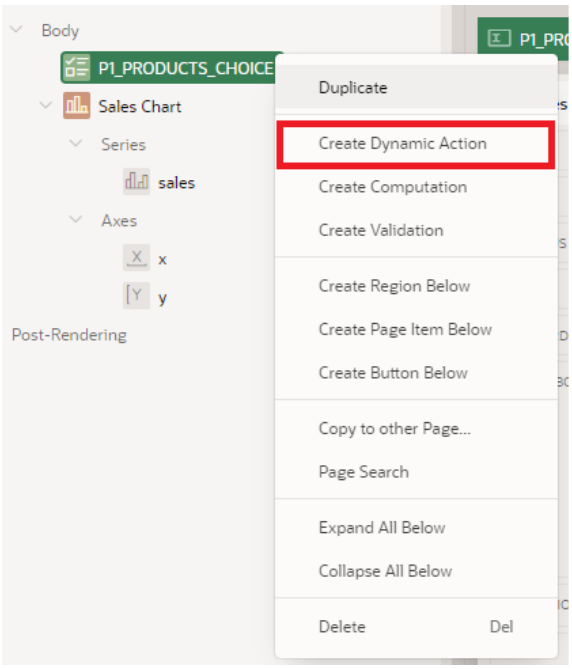



Abbildung 8.15: Dynamische Aktion für den Artikel P1_PRODUCTS_CHOICE erstellen.

Wir können das Attribut **„Name“** dieser **Dynamischen Aktion** auf **„PRODUCTS_CHOICE_REFRESH“** setzen, wie in Abbildung 8.16 zu sehen. Wir müssen auch sicherstellen, dass das Attribut **„Event“** im Abschnitt **„When“** auf **„Change“** gesetzt ist und dass das Attribut **„Selection Type“** auf **„Item(s)“** und das Attribut **„Item(s)“** auf unseren aktuellen Artikelnamen gesetzt ist, Diese **Dynamische Aktion** wird also ausgelöst, wenn sich der Wert von **„P1_PRODUCTS_CHOICE“** ändert.

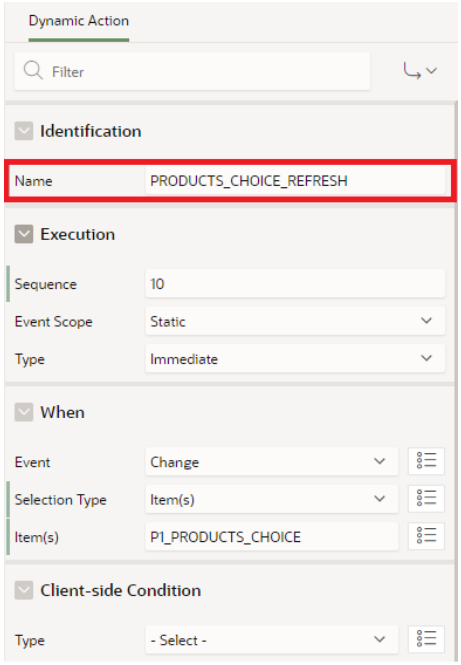



Abbildung 8.16: Einstellung grundlegender Attribute der dynamischen Aktion bei Änderung.

Klicken Sie in der Baumübersicht auf der linken Seite des **Seitendesigners** auf die Aktion, die ausgeführt wird, wenn die Bedingung **Dynamische Aktion wahr** ist, was der Fall ist, wenn sich unser **Checkbox Group** Element ändert. Wenn Sie eine neue **Dynamische Aktion** erstellen, wird die Aktion **„Show“** mit einem Fehler angezeigt, also klicken Sie darauf. Setzen Sie in den Einstellungen dieser Aktion das Attribut **„Action“** im Abschnitt **„Identification“** auf **„Refresh“**. Ändern Sie im Abschnitt **„Affected Elements“** das Attribut **„Selection Type“** auf **„Region“** und setzen Sie dann das Attribut **„Region“** auf **„Sales Chart“**, wie in Abbildung 8.17 zu sehen.

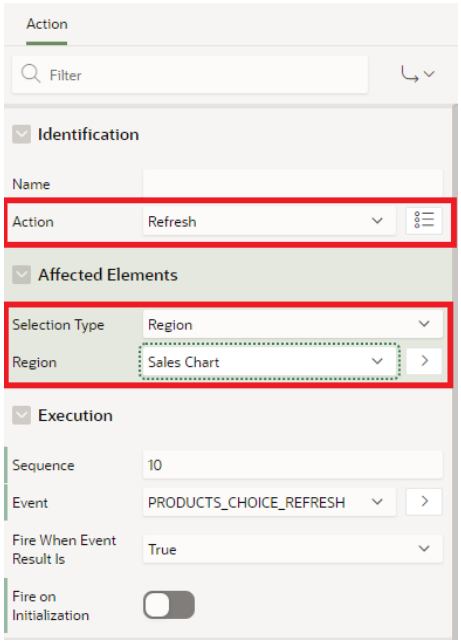



Abbildung 8.17: Einstellung der Attribute der Aktion Aktualisieren.

Abbildung 8.18 zeigt, wie Ihre Baumübersicht im linken Teil des **Seitendesigners** aussehen

sollte, wenn Sie alle Schritte korrekt ausgeführt haben.

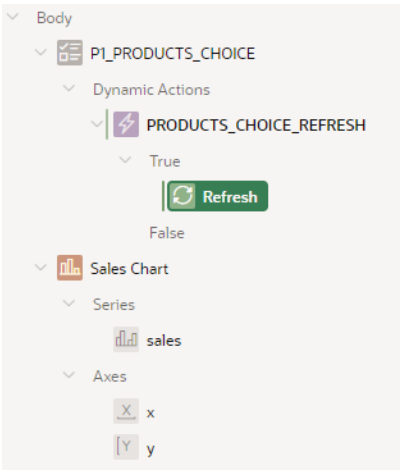



Abbildung 8.18: Dynamische Aktion zum Aktualisieren der Verkaufstabelle in der Baumübersicht.

Wenn wir unsere Anwendung ausführen und testen, sollten **Filtern** und **Aktualisieren** unseres Diagramms voll funktionsfähig sein, und die Anwendung sollte der in Abbildung 8.19 dargestellten ähneln.

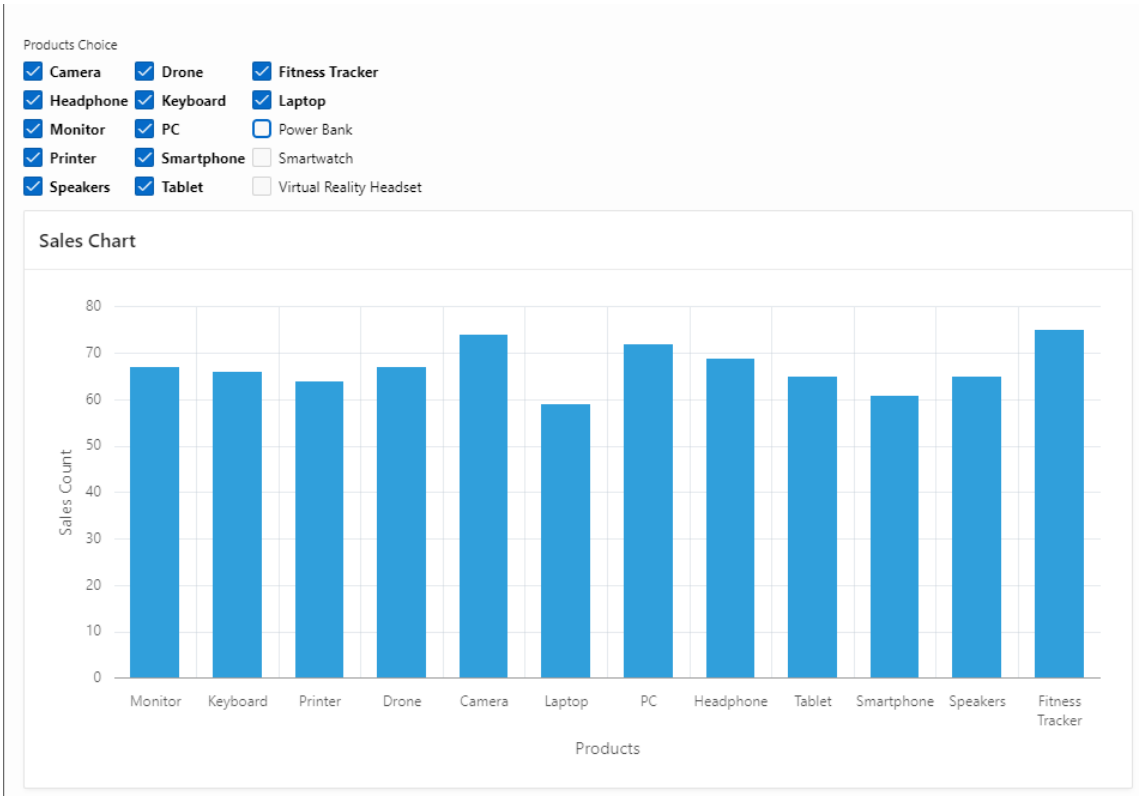



Abbildung 8.19: Anwendung nach Implementierung der Produktfilterung.

## 8.4 Hinzufügen einer Sortierung zum Balkendiagramm

Wir können die Benutzerfreundlichkeit verbessern, indem wir den Benutzern die Möglichkeit geben, die im Diagramm angezeigten Werte nach verschiedenen Kriterien zu sortieren. Zur Auswahl der Sortierkriterien wird eine **Auswahlliste** verwendet, wie in Abbildung 8.20 gezeigt.

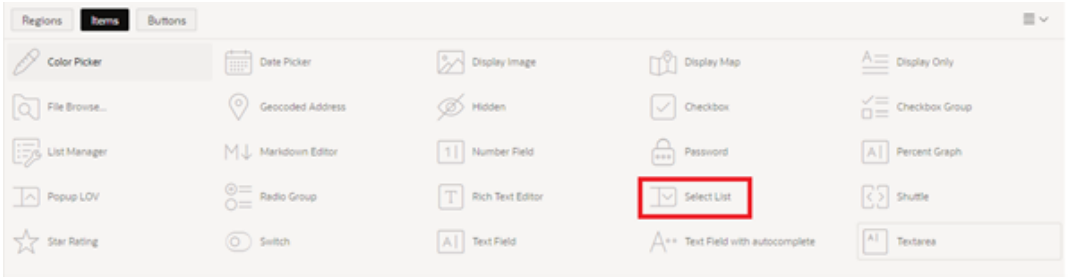



Abbildung 8.20: Auswählen und Platzieren von Elementen der Auswahlliste.

Wir platzieren die **Auswahlliste** in unserem **Sales Chart** Bereich unter dem Abschnitt **„Sort Order“** und nennen es **„P1_CHART_SORTING“**, wie in Abbildung 8.21 gezeigt.

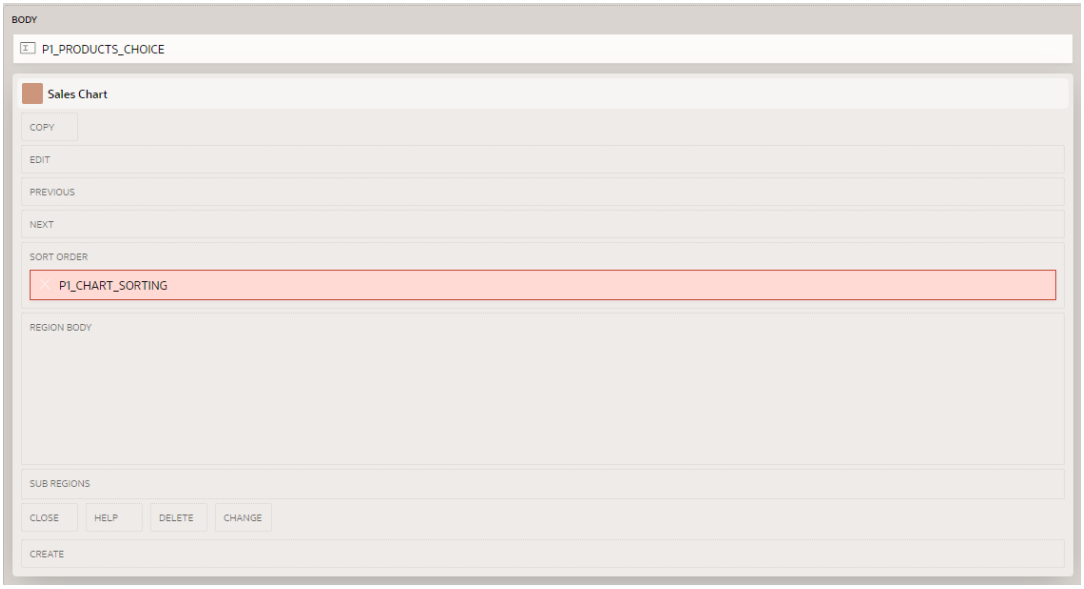



Abbildung 8.21: Sales Chart Region nach dem Platzieren und Umbenennen eines Elements der Auswahlliste.

Jetzt müssen wir eine Liste von Werten für die Auswahlliste definieren, die wir platziert haben. In den Einstellungen des Elements **„P1_CHART_SORTING“** müssen wir das Attribut **„Type“** im Abschnitt **„List of Values“** auf **„Static Values“** ändern. Dann müssen wir die Schaltfläche mit den Standardwerten **„Display1, Display2“** direkt neben dem Attribut **„Statische Werte“** drücken.

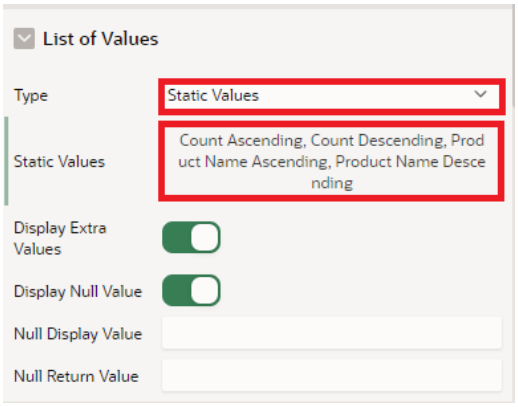



Abbildung 8.22: Einstellen der Werteliste für die Auswahlliste.

Nun sehen Sie, dass sich ein neues Fenster mit dem Titel **„Static Values“** geöffnet hat. In diesem Fenster müssen wir Anzeige- und Rückgabewerte definieren. Unter **Display**-Werte schreiben wir Sortierbezeichnungen, und unter **Return**-Werte schreiben wir Zahlen, die mit **1** beginnen, wie in Abbildung 8.23 zu sehen. Wenn wir wollen, dass diese Anzeigewerte so angeordnet werden, wie wir sie von oben nach unten eingegeben haben, müssen wir die Option **„Sort at Runtime“** deaktivieren.

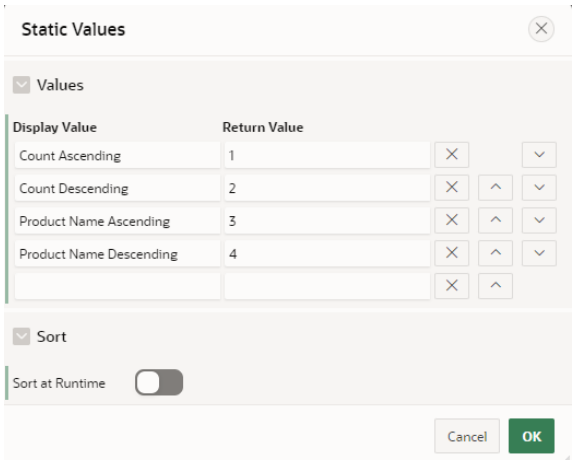



Abbildung 8.23: Einstellung statischer Werte für die Auswahlliste.

Wir müssen die Aktualisierung von **Sales Chart** auf dieselbe Weise hinzufügen, wie wir sie für das Element **„P1_PRODUCTS_CHOICE“** eingerichtet haben. Das Ergebnis ist in Abbildung 8.24 zu sehen.

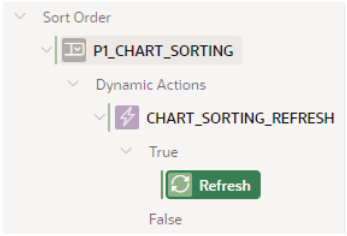



Abbildung 8.24: Hinzufügen einer dynamischen Aktion zur Auswahlliste.

Nachdem wir nun das Element **Auswahlliste** konfiguriert haben, müssen wir es mit dem Bereich **Sales Chart** verknüpfen. Dies geschieht durch Drücken der Schaltfläche direkt neben dem

Attribut **„Order By Item“**, das standardmäßig **„No Order By Item“** lautet, wie in Abbildung 8.25 hervorgehoben.

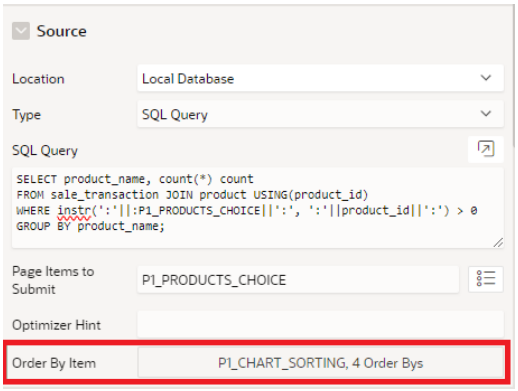



Abbildung 8.25: Einstellen der Reihenfolge nach Artikel für die Sales Chart Region.

Nachdem Sie auf die Schaltfläche geklickt haben, sollten Sie sehen, dass sich ein neues Fenster mit dem Titel **„Order By Item“** geöffnet hat. In diesem Fenster müssen wir zunächst das Attribut **„Name“** im Abschnitt **„Item“** auf **„P1_CHART_SORTING“** setzen. Dann müssen wir die Spalte „Klausel“ ausfüllen, wie in Abbildung 8.26 zu sehen ist. Diese Klauseln sind praktisch **„ORDER BY“**-Klauseln, die in **SELECT**-Anweisungen verwendet werden, aber ohne das **„ORDER BY“**Schlüsselwort.

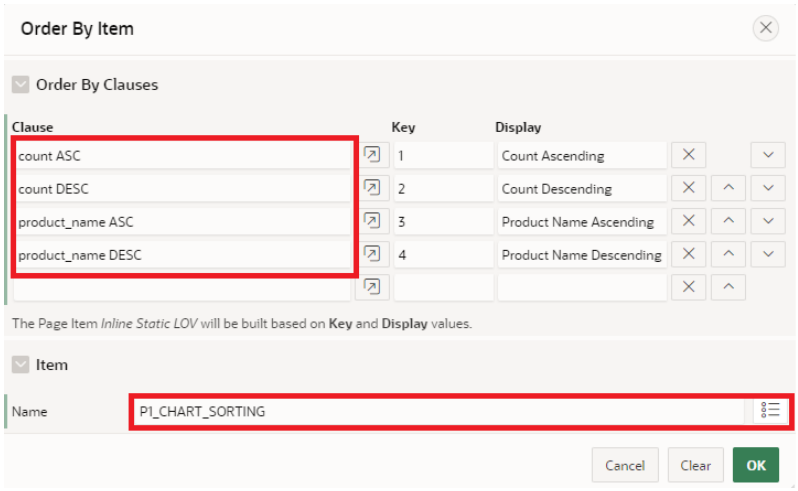



Abbildung 8.26: Festlegen von Sortierklauseln für Auswahllistenelemente, die für die Sortierung verwendet werden.

Wenn Sie alle diese Schritte genau befolgt haben, sollten **Sortierung** und **Aktualisierung** des Diagramms funktionieren, wenn Sie die Seite speichern und die Anwendung ausführen.

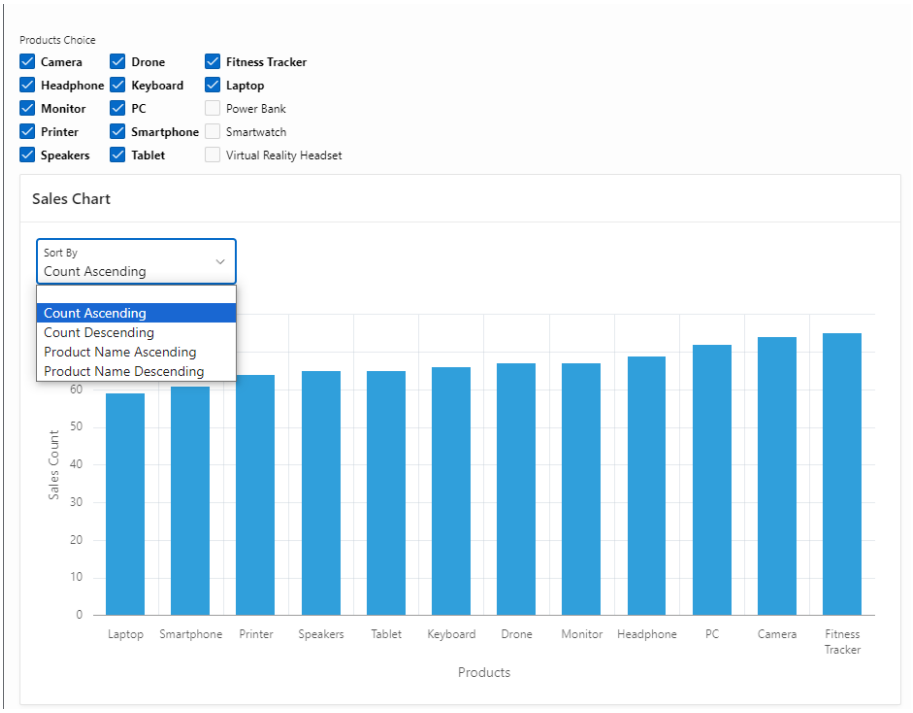



Abbildung 8.27: Anwendung nach Implementierung der Diagrammsortierung.

## 8.5 Verschiedene Arten von Diagrammen erstellen

Falls Sie eine andere Art von Diagramm implementieren möchten, können Sie dies tun, indem Sie den Diagrammbereich auswählen und die Registerkarte **„Attributes“** auf der rechten Seite des **Seitendesigners** aufrufen und dann den Wert des Attributs **„Type“** ändern. Wir erstellen ein weiteres **Line with Area**-Diagramm, das die monatlichen Verkaufszahlen anzeigt, und nennen es **„Monthly Sales Chart“**, und wir können es nach Jahr filtern.

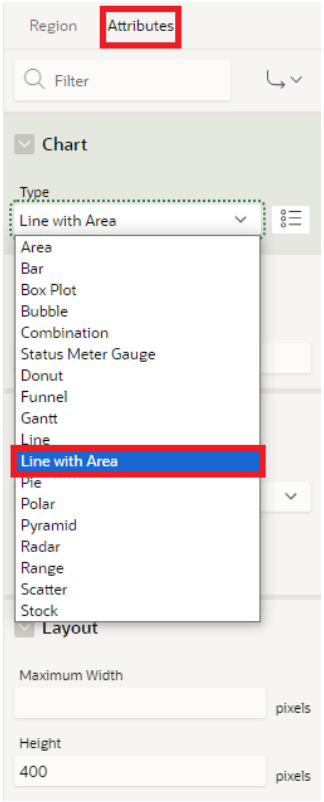



Abbildung 8.28: Ändern des Diagrammtyps in Line with Area.

Der nächste Schritt ist das Hinzufügen einer **Auswahlliste**, das wir zum **Filtern** des Diagramms nach **Jahr** verwenden werden. Wir platzieren dieses Element im Bereich **„Monthly Sales Chart“** unter dem Abschnitt **„Edit“**. Die SQL-Abfrage, mit der die Liste der Werte abgerufen wird, ist in Abbildung 8.29 dargestellt.

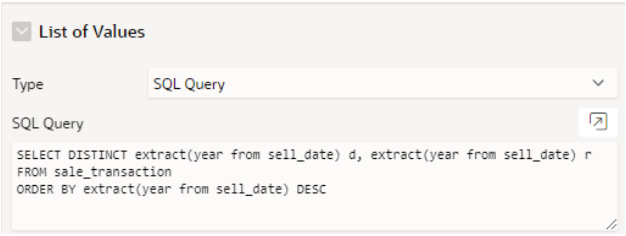



Abbildung 8.29: Liste der Werte für die Auswahlliste, die zum Filtern der Produktverkäufe nach Jahr verwendet wird.

Nach der Vorbereitung der **Auswahlliste**, die wir für die Filterung verwenden werden, können wir die Quell-SQL-Abfrage für die Reihe im Bereich **„Monthly Sales Chart“** schreiben. Die SQL-Abfrage, die wir verwenden werden, ist in Abbildung 8.30 dargestellt. Wir werden auch **„Label“** auf **„SELL_MONTH“** und **„Value“** auf **„COUNT“** abbilden. Außerdem ordnen wir die Werte nach der Monatsnummer aufsteigend an und zeigen den vollständigen Monatsnamen auf der **X** Achse an. Außerdem setzen wir die Beschriftung der **X** Achse auf **„Month“** und die **Y** Achse auf **„Sales Count“**.

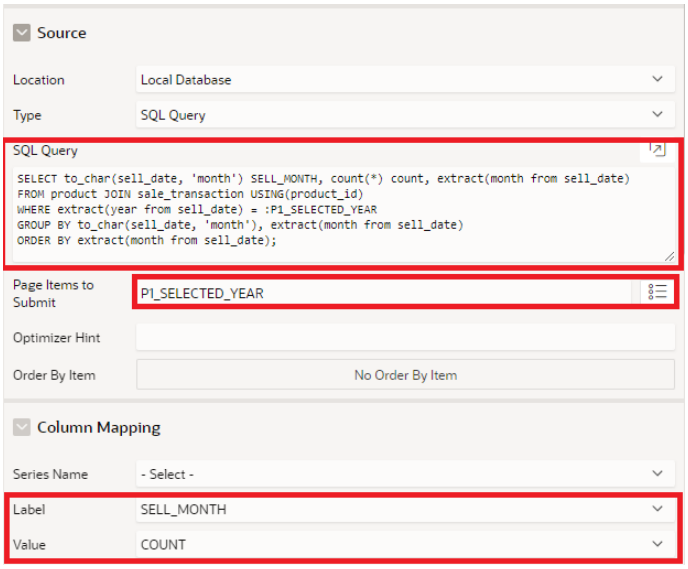



Abbildung 8.30: Einstellen der SQL-Quellabfrage und der Spaltenzuordnung für die Monthly Sales Chart Region.

Wenn Sie alles richtig konfiguriert haben, sollte Ihr Diagramm dem in Abbildung 8.31 dargestellten ähneln.

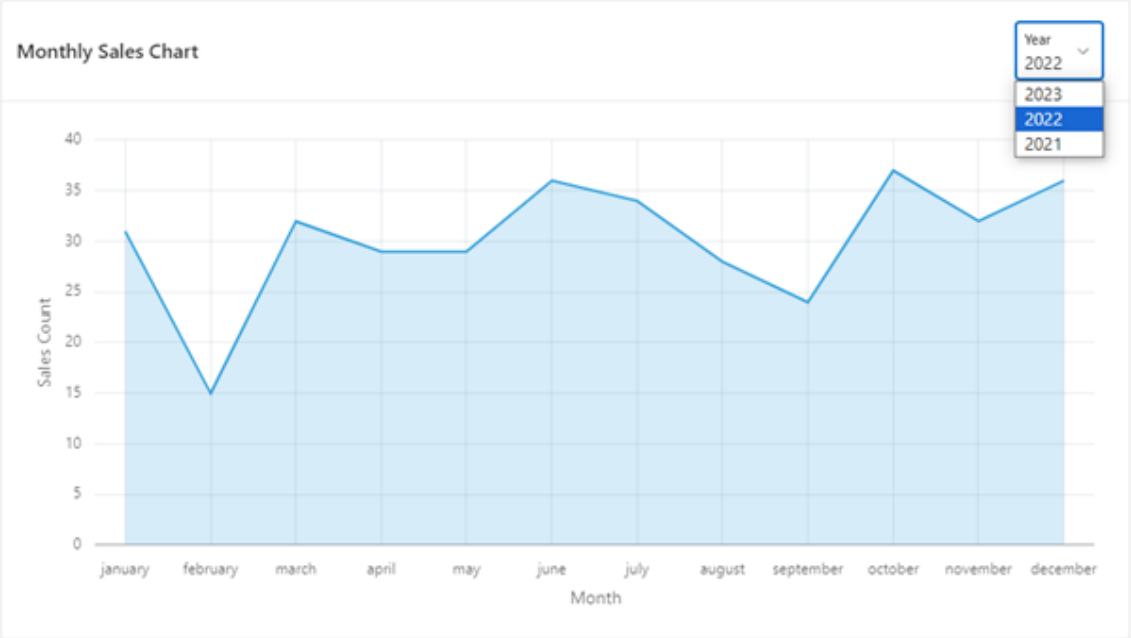



Abbildung 8.31: Monthly Sales Chart.

## 8.6 Fragen

1. Was ist der Hauptzweck von Oracle APEX-Diagrammregionen?
2. Welche Arten von Diagrammen können Sie mit Oracle APEX-Diagrammregionen erstellen?
3. Wie kann man einen Diagrammbereich in Oracle APEX aktualisieren, wenn Benutzerinteraktionen, wie z.B. Filtern oder Sortieren, auftreten?
4. Was bewirkt die Funktion INSTR, und warum wird sie in Oracle APEX-Diagrammregionen verwendet?

## 8.7 Antworten

1. Der Hauptzweck von Oracle APEX-Diagrammregionen besteht darin, Daten visuell darzustellen, damit die Benutzer Einblicke gewinnen und datengestützte Entscheidungen treffen können.
2. Oracle APEX-Diagrammregionen unterstützen verschiedene Diagrammtypen, einschließlich Balkendiagramme, Liniendiagramme, Tortendiagramme und mehr.
3. Sie können eine dynamische Aktion konfigurieren, um einen Diagrammbereich in Oracle APEX zu aktualisieren, wenn Benutzerinteraktionen die Daten ändern, wie z.B. Filtern oder Sortieren.
4. Die Funktion INSTR wird verwendet, um den Index des ersten Vorkommens einer Teilzeichenkette in einer Zeichenkette zu finden. In Oracle APEX-Diagrammregionen wird sie verwendet, um zu prüfen, ob bestimmte Werte mit ausgewählten Kriterien übereinstimmen, z.B. Produkt-IDs in einer Liste.



# 9. Wie verwaltet man Menüs?

VERONIKA ŠALGOVÁ, MICHAL KVET, MIROSLAV POTOČÁR; ÜBERSETZER W. SCHWINGER

## 9.1 Wie verwaltet man Menüs

Nach der Erstellung einer Anwendung wird automatisch ein Navigationsmenü erstellt. Es besteht aus Listeneinträgen, die mit den Seiten der Anwendung verknüpft sind. Es ist möglich zu ändern, wie und wo ein Navigationsmenü angezeigt wird, indem die Navigationsmenü-Attribute auf der Seite Benutzeroberfläche bearbeitet werden. Folgende Arten von Navigationsmenüs sind in Oracle APEX verfügbar:

- Side Menu,
- Top Menu,
- Mega Menu.

## 9.2 Side Menu

Diese Art von Menü kann durch Klicken auf das Menüsymbol in der Kopfzeile ein- und ausgeklappt werden. Die Navigationspunkte werden in Form einer Baumkomponente angezeigt. Auch die Unterpunkte können ein- und ausgeklappt werden. Es kann ein vollständiges Menü angezeigt oder auf eine schmale Symbolleiste reduziert werden.

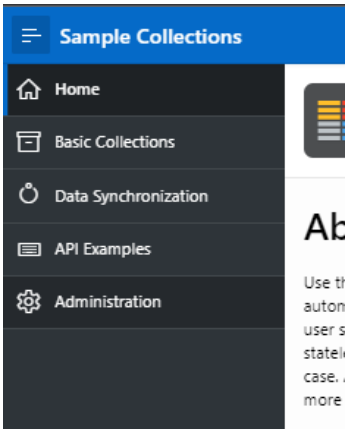



Abbildung 9.1: Erweitertes seitliches Navigationsmenü

Es ist auch möglich, ein Navigationsmenü so zu konfigurieren, dass es komplett ausgeblendet wird, was eine Standardoption ist. Eine schmale Symbolleiste kann angezeigt werden, wenn sie mithilfe der Vorlagenoptionen eingeklappt wird.

## 9.3 Top Menu

Am oberen Rand der Anwendung wird ein Top-Navigation-Menü angezeigt. Oracle APEX stellt zwei Vorlagen für die Anzeige des Menüs zur Verfügung, z.B. die *Top Navigation Tabs* und das *Top Navigation Menu*. Der erste Typ ist für einfache Anwendungen mit wenigen Registerkarten geeignet. Sie ist in Abbildung 9.2 dargestellt.

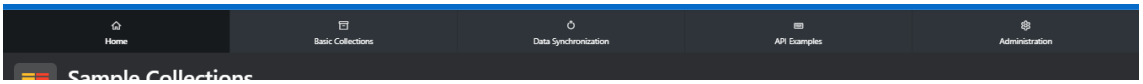



Abbildung 9.2: Top Navigation Tabs

Das Top Navigation Menu bietet eine Menüleiste. Es ist ideal für komplexere Anwendungen mit mehreren Hierarchieebenen innerhalb der Navigation. Es ist in Abbildung 9.3 dargestellt.

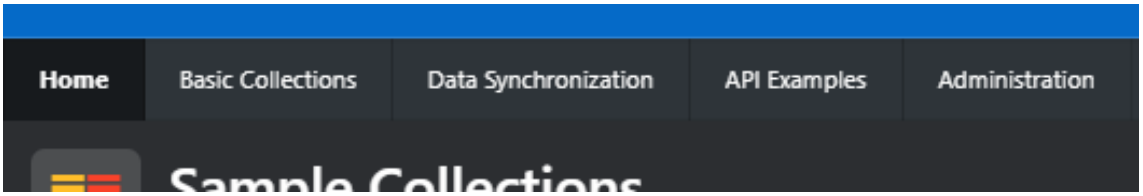



Abbildung 9.3: Top Navigation Menu

Die Vorlage Top Navigation Tabs positioniert sich automatisch am unteren Rand des Bildschirms für kleine Bildschirme oder mobile Geräte. Das Top Navigation Menu ähnelt den meisten Desktop-Anwendungen.

## 9.4 Mega Menu

Diese Art von Menü wird als Pop-up-Panel angezeigt, das über das Symbol in der Kopfzeile geöffnet oder geschlossen werden kann. Mega-Menus sind hilfreich, um alle Navigationspunkte auf einmal anzuzeigen.

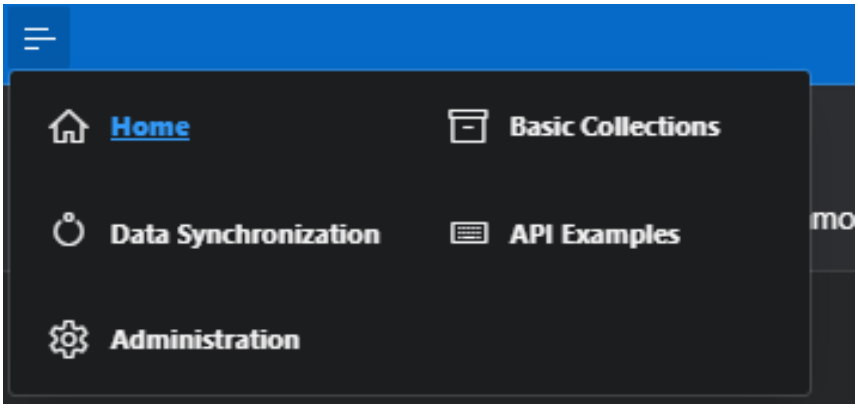



Abbildung 9.4: Top Navigation Mega Menu

Außerdem ist es möglich, die Anzahl der Spalten im Menü zu wählen.

## 9.5 Bearbeiten von Menülisten

Einzelne Menüpunkte bilden eine Liste, die ähnlich wie jede andere Liste erstellt und bearbeitet werden kann. Der Zugriff darauf erfolgt in *Shared Components* unter der Region Navigation. Sie können die Option *Navigation Menu* oder eine *Liste* wählen. Es ist möglich, die Position und die Art der Anzeige des Menüs zu ändern, indem Sie die Attribute des Navigationsmenüs auf der Seite User Interface bearbeiten. Sie gelangen dorthin, indem Sie auf *Shared Components* und *User Interface Attributes* klicken. Unter der Registerkarte *Navigation menu* können Sie die Attribute bearbeiten, die in Abbildung 9.5 dargestellt sind.

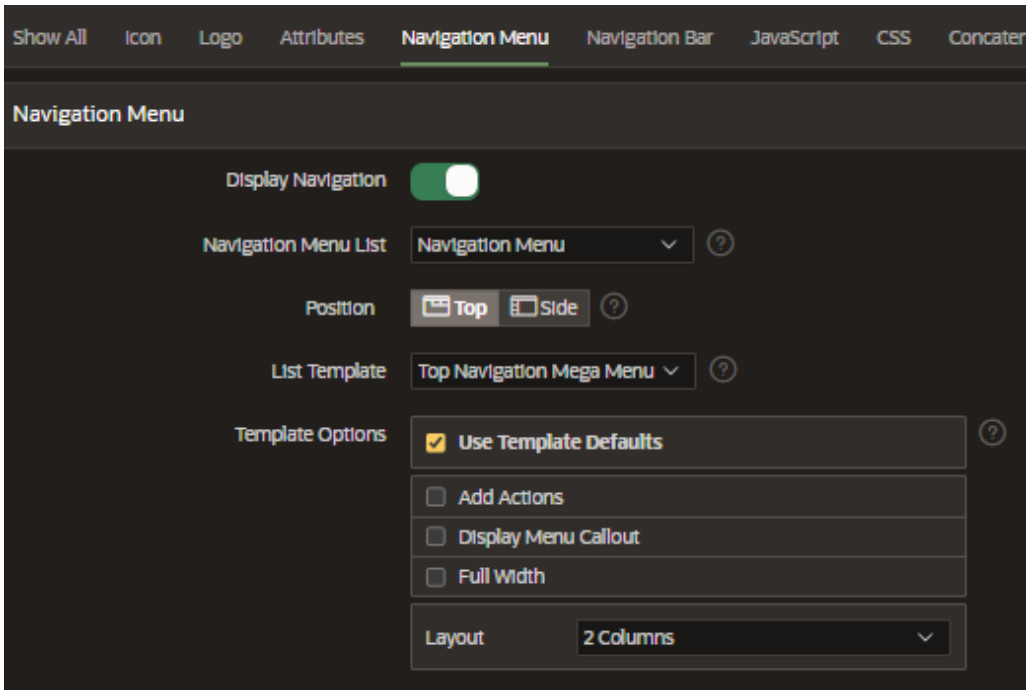



Abbildung 9.5: Attribute des Navigationsmenüs

- Navigation anzeigen - um die Navigation ein- oder auszuschalten.
- Navigationsmenüliste - zur Auswahl der Liste, die für das Navigationsmenü verwendet wird.
- Position - zur Auswahl der Position des Navigationsmenüs (*oben* oder *seitlich*).

- Listenvorlage - zur Auswahl aus verschiedenen Vorlagen für die Darstellung des Navigationsmenüs.
- Vorlagenoptionen - zum Festlegen der für die Navigationsmenüliste verwendeten Vorlagenoptionen.

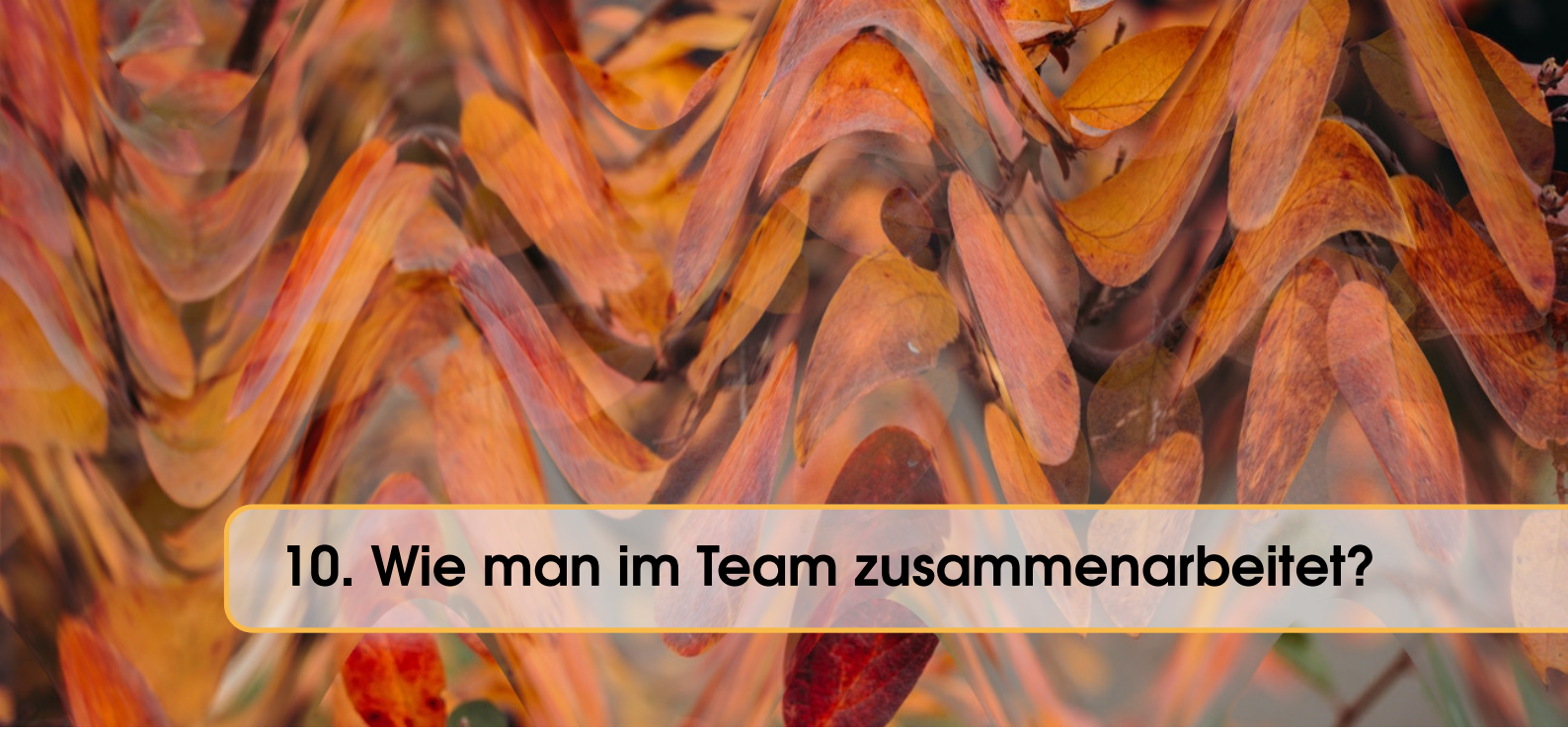
Nachdem Sie auf die Schaltfläche *Apply changes* geklickt haben, können Sie die Anwendung ausführen und die Änderungen sehen.

## 9.6 Fragen

1. Das in den Oracle APEX-Anwendungen verwendete Menü befindet sich in der Regel im linken Teil des Bildschirms. Auf welchen Typ bezieht es sich?
2. Wo können Sie die Definition der zu bearbeitenden Menüpunkte finden?
3. Welcher Menütyp ermöglicht es, alle Elemente gleichzeitig anzuzeigen, ohne dass das Untermenü erweitert werden muss?

## 9.7 Antworten

1. Side Menu.
2. Einzelne Menüpunkte bilden eine Liste, die ähnlich wie jede andere Liste erstellt und bearbeitet werden kann. Der Zugriff erfolgt in den Shared Components unter der Region Navigation.
3. Mit dem Mega Menu können Sie alle Elemente gleichzeitig mit nur einem Klick anzeigen, ohne dass Sie die Untermenüs erweitern müssen.



# 10. Wie man im Team zusammenarbeitet?

P. STANISZEWSKI, M. SOŃTA, A. KIERZKOWSKI

## 10.1 Die kollaborative Wissensproduktion ist die Essenz der Low-Code-Entwicklung

„Being collaborative distinguishes us as humans even more than our opposable thumbs“, da wir von Natur aus darauf programmiert sind, bevorzugt zu kooperieren (siehe z.B. [1]; [**Jemielniak2020**]), und vernetzte Plattformen sind mit Funktionen ausgestattet, die den kollektiven und gemeinsamen Wissensaustausch fördern.

Die Vorteile einer intelligenten Zusammenarbeit auf digitalen Plattformen sind zweifach:

- der Produktivitätsaspekt, wenn der Einzelne seine kollektive Intelligenz nutzt, um ein besseres Produkt zu entwickeln und
- der motivierende Aspekt, wenn der Prozess die kreativen Energien und den gemeinschaftsorientierten Ansatz zur Wissensentwicklung aktiviert.

## 10.2 Gemeinsam online sein

APEX von Oracle ist eine Plattform, die für die Vernetzung und Interaktion bei der Arbeit an den Anwendungen geschaffen wurde, und dieser Ansatz erfüllt die Definition einer Plattform für kollaborativen Aktivismus. Diese Art der kollaborativen Erfahrung erfordert jedoch „a particular way of being together“ ([3]). Ein Teil der kollaborativen Erfahrung ist die Möglichkeit, den Inhalt zu betrachten, zu kommentieren und mitzugestalten, um gemeinsam Wissen zu produzieren, und die Förderung von Kreativität und Geist durch die Gestaltung sozialer Beziehungen ist eine weitere Dimension der kollaborativen Nutzung der APEX-Geschichte. Die APEX-Gemeinschaft von Oracle mit mehr als zehn regionalen Gemeinschaftsveranstaltungen pro Jahr, neben vielen anderen:

- APEX World, Niederlande
- APEX World, APEX Alpe Adria, Österreich/Kroatien/Slowenien
- APEX World, APEX Connect, Deutschland
- APEX World, ODTUG Kscope, USA

haben das Entwicklungsstadium erreicht, um die kommerzielle Plattform zu entwickeln, die versprochen wurde, für die Nutzer für immer kostenlos zu bleiben, und zwar in einer kollaborativen Open-Source-Bewegung und mit dem Ehrgeiz, die Richtung des anregenden Gemeinschaftsgeistes zu verfolgen.

Mit der Anzahl der Interaktionen und Projekte wird die Art und Weise der Gemeinschafts-

entwicklung immer immersiver, was Möglichkeiten mit sich bringt „[...] modes of being together online, particularly in very immersive environments, may have an additional cognitive and perceptive layer that is yet unexplored“([3]) mit dem Ziel, die Reife des digitalen Aktivismus zu erreichen und verschiedene nicht-IT-geschulte Zielgruppen einzuladen.

## 10.3 Technischer Vorsprung durch Zusammenarbeit

Wenn wir nach akademischen Artikeln über APEX suchen, behandeln die meisten Abhandlungen geschäftliche Anwendungsfälle, Fälle von abteilungsübergreifenden Bemühungen, eine Anwendung zu erstellen, die das Personalwesen, die soziale Verantwortung von Unternehmen, die Kommunikation mit verschiedenen Interessengruppen und ähnliches einbezieht.

Diese Herangehensweise, die Zugänglichkeit der Plattform und die breite Palette an Bildungsinhalten über APEX machen dieses Tool auch zu einer Chance, in die Welt des Low-Coding einzusteigen und das Geschäftsbewusstsein für praktische App-Anwendungen zu schulen. Mit dieser Einstellung eröffnet APEX eine Möglichkeit für Nicht-IT-Fachleute, an IT-Projekten mitzuarbeiten, und bietet Raum, um das Niveau der technischen Kenntnisse zu erhöhen. Diese Kompetenzen und Einstellungen schaffen bessere Beschäftigungsmöglichkeiten für alle am Softwareentwicklungsprozess beteiligten Akteure.

Die Art und Weise der Zusammenarbeit hängt in der Regel mit der Methodik des Projekts zusammen. Auch wenn Low-Code-Plattformen wie APEX darauf ausgelegt sind, Ziele sehr schnell zu erreichen, müssen wir typischerweise mehr als einen Entwickler haben, um ein Projekt in der angenommenen Zeit und dem angenommenen Budget zu realisieren. In einer solchen Situation müssen wir das richtige Werkzeug haben, um den Entwicklern zu helfen, zusammenzuarbeiten und die Herausforderungen und Möglichkeiten zu meistern, die für die Zusammenarbeit in IT-Projekten typisch sind. Je nach Umfang und Anforderungen unserer Organisation kann dies mit Aspekten verbunden sein wie:

- Standardisierung - wenn wir sicherstellen wollen, dass alle Anwendungselemente einheitlich sind. Sie bezieht sich auf die UI-Standards, Namenskonventionen, Programmiertechniken, den Codierungsstil und alle anderen für die ausgewählte Technologie typischen Standards. In der APEX-Welt können Sie zum Beispiel über verschiedene Arten der Benennung von Elementen, Prozessen oder Validierungen nachdenken oder Standards bei der Auswahl von Seitenzahlen berücksichtigen.
- Schulung - Die Entwicklung von Anwendungen zusammen mit anderen Entwicklern hilft bei der Erweiterung der Fähigkeiten. Andere Teammitglieder sind vielleicht fortgeschrittener in Aspekten, auf die wir nicht besonders scharf sind. Selbst bei der Verwendung einer Low-Code-Plattform wie APEX, die es theoretisch jedem Benutzer ermöglicht, ein „Full-Stack“-Entwickler zu sein, können wir Spezialisierungen finden. Zum Beispiel fühlen sich einige Leute viel wohler mit der Datenbankseite, andere mit dem reinen APEX und wieder andere mit Frontend-Features und Sprachen. Ein größeres Team zu haben, hilft dem Rest, Fähigkeiten zu entwickeln, die sie nicht hatten und erhöht die allgemeine Qualität der Anwendung.
- Zeit - es liegt auf der Hand, dass mehr Entwickler Anwendungen schneller bereitstellen können, was zu einer besseren Leistung des Teams oder der Abteilung führt, die für ein bestimmtes System oder eine bestimmte Plattform zuständig ist.
- Qualität - das größere Team kann einen Code-Review-Prozess einführen und die Qualität der von den Teammitgliedern gelieferten Elemente überprüfen. Dies erhöht die Qualität des Endprodukts und begrenzt die Anzahl der Fehler.
- Feedback - Zusammenarbeit bedeutet auch Kommunikation zwischen Entwicklern und Benutzern. Es ist von Vorteil, von den Nutzern Informationen über die Qualität der Anwendung, gemeldete Fehler oder Änderungswünsche zu erhalten.

Oracle APEX, als ausgereifte Plattform, deckt verschiedene Strategien der Zusammenarbeit

mit eingebauten Funktionen ab, aber auch mit Optionen, die durch externe Tools wie Collaboration Suites, Task Management Anwendungen oder Code Verification Tools erweitert werden können.

Auch wenn wir gerade erst mit APEX angefangen haben, ist es wichtig zu wissen, wie wir mit anderen Anwendern zusammenarbeiten können, um diese Arbeit produktiver zu machen und sicher zu sein, wie man typische Probleme wie Aufgabenzuweisungen, die Arbeit an den genauen Seiten/Features oder die Kontrolle des Status der geplanten Arbeit angeht.

Das Wichtigste gleich zu Beginn ist die Kommunikation. Wenn das Team aus mehr als einer Person besteht, sollten Sie Möglichkeiten haben, die Arbeit für die nächsten Tage zu planen, über den Status zu informieren und die Zeit und die Fristen zu kontrollieren. Nichts ist wichtiger als regelmäßige Treffen und Kommunikation, wenn etwas unklar ist. Es ist gut, über tägliche Besprechungen, wöchentliche Planungen oder andere regelmäßige Veranstaltungen nachzudenken. Sie können auch eine der bekannten Methoden wie Scrum anwenden, um solche Gewohnheiten an einem Ort festzuhalten. Fangen Sie einfach damit an, und Sie werden die beste Art der Zusammenarbeit für Ihr Team herausfinden.

Am Anfang, wenn Sie über Tools für die Zusammenarbeit nachdenken, können Sie einfache Tabellenkalkulationen in Betracht ziehen, um zu überwachen, wer was tut. Natürlich gibt es auf dem Markt viele spezielle Anwendungen für die Aufgabenverwaltung wie Jira, Monday, ClickUp, Redmine usw., und natürlich können Sie mit wachsendem Team und Umfang auch die Verwendung einer dieser Anwendungen in Betracht ziehen. Die meisten dieser Aufgaben können jedoch mit den in der Oracle Application Express-Plattform enthaltenen Funktionen abgedeckt werden.

## 10.4 Beschreibung der Funktionen

### 10.4.1 Seitensperre

Die Seitensperre ist eine nützliche Funktion, die verhindert, dass mehrere Entwickler gleichzeitig dieselbe Seite bearbeiten. Wenn ein Entwickler eine Seite bearbeitet, verhindert die Seitensperrfunktion, dass andere Entwickler Änderungen an der gleichen Seite vornehmen, bis der erste Entwickler seine Arbeit beendet und die Seite freigibt. Dadurch werden Konflikte vermieden und sichergestellt, dass Änderungen geordnet und kontrolliert vorgenommen werden (siehe Abbildung 10.1).

| Page | Name | Alias | Updated | Updated By | Type | Group | Lock | Run |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 3.1 years ago | - | Global Page | Unassigned | | |
| 1 | Dashboard | dashboard | 9 seconds ago | pstaniszewski@pretius.com | Report | Customers | | |
| 2 | Customer Details | customer-details | 2 years ago | - | DML Form | Customers | | |
| 3 | Categories | categories | 1 seconds ago | pstaniszewski@pretius.com | Interactive Report | Administration | | |
| 4 | Category Details | category-details | 3.1 years ago | - | DML Form | Administration | | |
| 5 | Customer Types | customer-types | 1.6 years ago | - | Interactive Report | Administration | | |
| 6 | Status Details | status-details | 3.1 years ago | - | DML Form | Administration | | |
| 7 | Maintain Update | maintain-update | 2 years ago | - | DML Form | Customers | | |
| 8 | Contact Types | contact-types | 1.6 years ago | - | Interactive Report | Administration | | |
| 9 | Contact Type Details | contact-type-details | 3.1 years ago | - | DML Form | Administration | | |
| 10 | Validate Customer | validate-customer | 2 years ago | - | Report | Customers | | |
| 11 | Build Options | build-options | 1.6 years ago | - | Interactive Report | Administration | | |
| 12 | Administration | settings | 2 years ago | - | Report | Administration | | |
| 13 | Customer Usage Metrics | customer-usage-metrics | 2 years ago | - | Legacy Calendar | Customers | | |

Abbildung 10.1: Seitensperrfunktion in APEX.

Sie können eine bestimmte Seite sperren, indem Sie auf das Schlosssymbol in den Seitenlisten im Application Builder klicken oder dies auf der Ebene der Seitenansicht tun (siehe Abbildung 10.2).

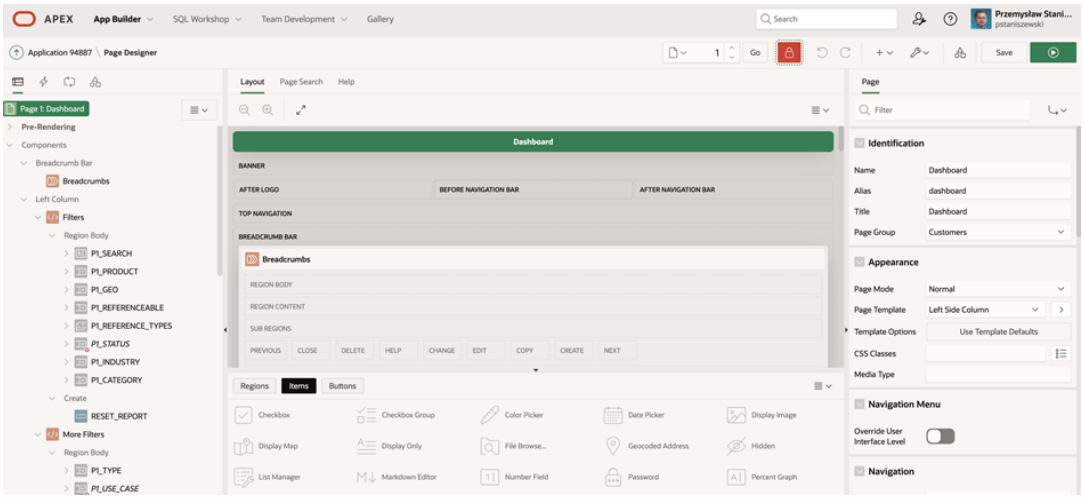



Abbildung 10.2: Sperren einer bestimmten Seite.

Während Sie die Seite sperren oder entsperren, können Sie einen Kommentar hinzufügen, damit Sie oder andere Entwickler weitere Informationen über den Grund oder den Arbeitsfortschritt sehen können.

### 10.4.2 Kommentare

Mit Oracle APEX können Entwickler Kommentare zu Seiten, Regionen, Elementen und vielen anderen Anwendungselementen hinzufügen. Kommentare können verwendet werden, um anderen Entwicklern einen Kontext zu liefern, die Gründe für eine Designentscheidung zu erläutern oder einfach Änderungen an einer Anwendung zu dokumentieren. Durch die Verwendung von Kommentaren können Entwickler besser miteinander kommunizieren und sicherstellen, dass alle auf der gleichen Seite sind (siehe Abbildung 10.3).

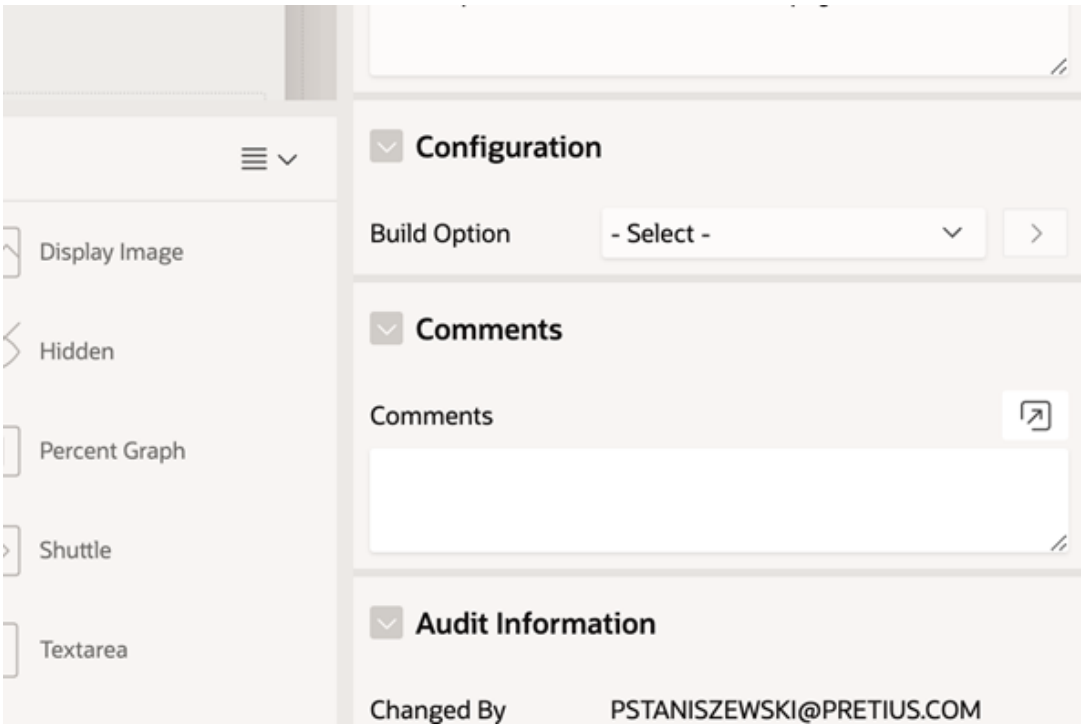



Abbildung 10.3: Hinzufügen von Kommentaren.

Die Dokumentation Ihres Codes für andere Teammitglieder oder künftige Entwickler, die sich um die Entwicklung oder Wartung der Anwendung kümmern, ist wichtig. Auch wenn die Verwendung einer Low-Code-Plattform selbst definieren kann, wie eine bestimmte Funktion ausgeführt wurde, ist das Hinzufügen einiger Informationen immer noch eine gute Praxis, wenn ein bestimmtes Element einmalig erstellt wird.

### 10.4.3 Build-Optionen

Build-Optionen sind eine weitere Funktion, die Teams helfen kann, in Oracle APEX effektiver zusammenzuarbeiten. Build-Optionen ermöglichen es Entwicklern, unterschiedliche Build-Einstellungen für verschiedene Umgebungen wie Entwicklung, Test und Produktion festzulegen. Dies erleichtert die Verwaltung von Änderungen in verschiedenen Umgebungen und stellt sicher, dass bei der Bereitstellung einer Anwendung die richtigen Einstellungen verwendet werden (siehe Abbildung 10.4).

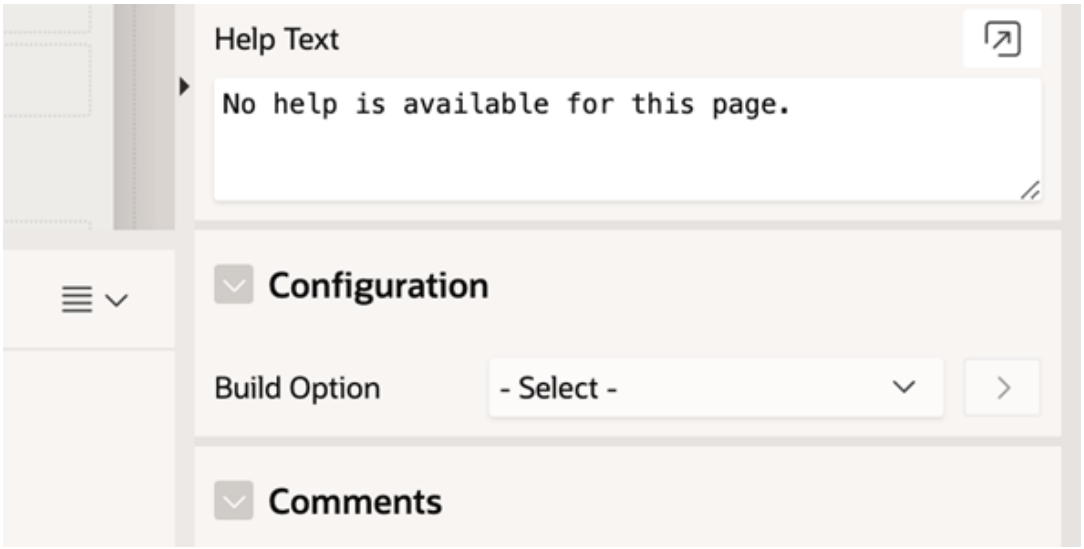



Abbildung 10.4: Build-Optionen.

Sie können es als Information betrachten, ob eine bestimmte Funktion (Seite, Element, Prozess) in die nächste Version aufgenommen werden soll, die in eine andere Umgebung exportiert/importiert wird. Beim Export/Import Ihrer Anwendung können Sie auswählen, ob ein bestimmter „Build“ enthalten sein soll. Stellen Sie sich die Situation vor, wenn ein Entwickler an einer Funktion arbeitet und es nicht mehr vor der Deadline schafft. Anstatt diese Seite aus der Anwendung zu löschen, können Sie einfach ein spezielles „Build“ für sie erstellen und sie beim Export der Anwendung ausschließen.

Die Verwendung dieser Funktion kann ein erster Schritt bei der Definition eines CI/CD-Prozesses für Ihre Anwendung sein, der in der professionellen Softwareentwicklung unerlässlich ist.

### 10.4.4 Team Development

Die Team Development-Funktion von Oracle APEX ermöglicht es Teams, in einer kontrollierten und kollaborativen Umgebung gemeinsam an einer Anwendung zu arbeiten. Mit dieser Funktion können Teammitglieder Projekte verwalten, vor allem:

- Meilensteine und Projektphasen definieren;
- Aufgaben/Probleme erstellen, zuweisen und überwachen;
- Klassifizierungen mit Hilfe von Label-Gruppen und Labeln anwenden;

- Themenvorlagen erstellen, um die Erstellung von Aufgaben oder Berichten einheitlich zu gestalten;
- Probleme mit einer Anwendung und einer Seite verknüpfen;
- Bestehende Probleme kommentieren und diskutieren. Den Status ändern und den Fortschritt der Projektphase überwachen;
- Verschiedene Arten von Problemen abonnieren;
- Feedback von Anwendungsnutzern überwachen und entscheiden, was damit geschehen soll.

Team Development ist einer der vier Hauptbereiche, die in der APEX IDE verfügbar sind. Seine Verfügbarkeit hängt von Ihrer APEX-Version ab. Sie wurde in APEX 19.2 eingeführt, ist aber in den neuesten Versionen nicht sofort verfügbar, sondern erfordert zusätzliche Aktionen, um sie verfügbar zu machen (siehe Abbildung 10.5).

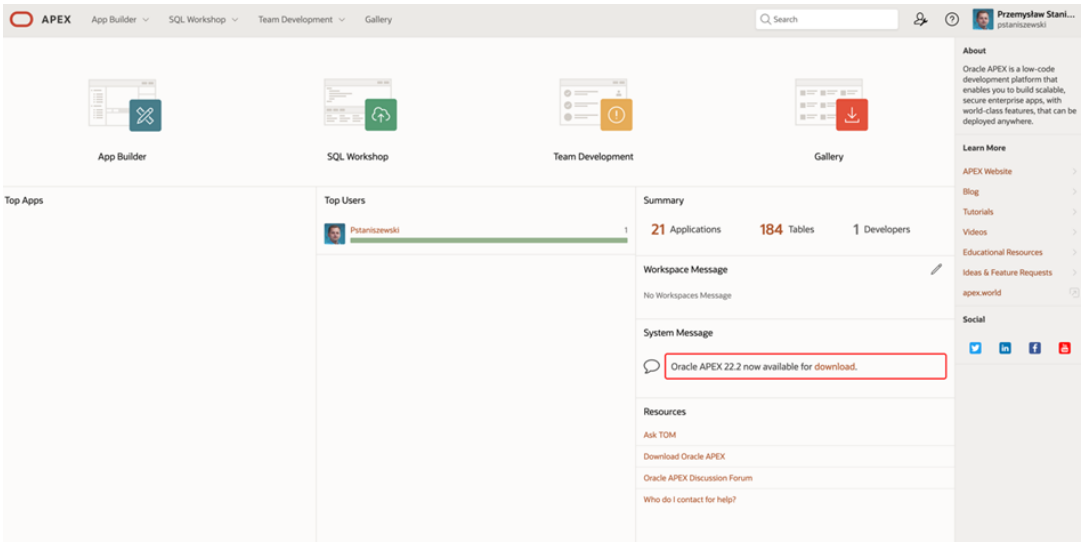



Abbildung 10.5: Team Development-Funktion.

Um es zu aktivieren, müssen Sie im Hauptmenü die Option „Team Development“ auswählen. Dann werden Sie auf die entsprechende Installationsseite weitergeleitet. Wenn Sie möchten, können Sie Beispiele für Probleme, Meilensteine, Vorlagen und Labels einfügen, aber Sie können diesen Teil auch überspringen und die Funktion ohne sie aktivieren (siehe Abbildung 10.6).

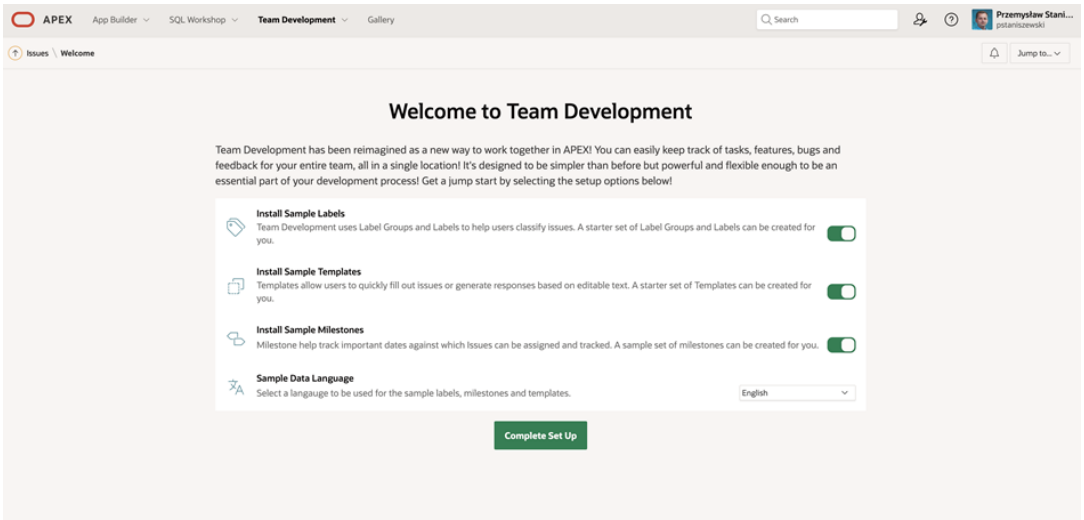



Abbildung 10.6: Aktivierung der Team Development-Funktion.

Wenn Ihr Team mit der Verwendung von „Team Development“ beginnen möchte, sollten Sie zunächst das Verzeichnis für Ihren Arbeitsstil konfigurieren. Gehen Sie dann zum Abschnitt Utilities und definieren Sie die Arten Ihrer Probleme, Projektphasen und Meilensteine oder Vorlagen für die typische Kommunikation im Projekt (siehe Abbildung 10.7).

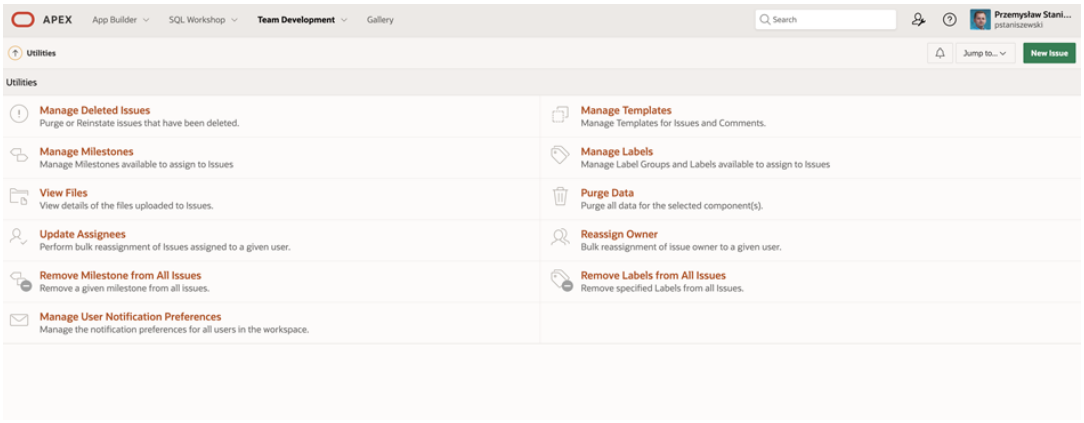



Abbildung 10.7: Anpassung der Team Development-Funktion.

Der wichtigste Teil dieser Funktion ist der Issue-Bereich, in dem Sie alle gemeldeten Issues finden, sie kategorisieren und bestimmten Personen zuordnen können. Jedes Issue kann mehr als einem Entwickler zugewiesen werden, der dann Kommentare hinzufügen, es in seinem Kontext diskutieren und seinen Status ändern kann. Bei Bedarf können die Benutzer Dateien hochladen und grundsätzlich die gesamte Issue-bezogene Kommunikation innerhalb von „Team Development“ durchführen. Wichtig ist auch, dass alle Issues mit einer bestimmten Anwendung und Seite verbunden werden können, so dass es einfach ist, eine Untersuchung zu starten und zu überwachen, welche Probleme mit welchem Teil der Anwendung zusammenhängen. Der Issue-Status ist eine einfache Version des Issue-Workflows, der in vielen Aufgabenverwaltungssystemen zu finden ist - Benutzer können sie melden, sie als „in Bearbeitung“markieren, zum Berichterstatter zurückkehren und mit Informationen über den Erfolg oder verschiedene Gründe für das Schließen beenden (siehe Abbildung 10.8).

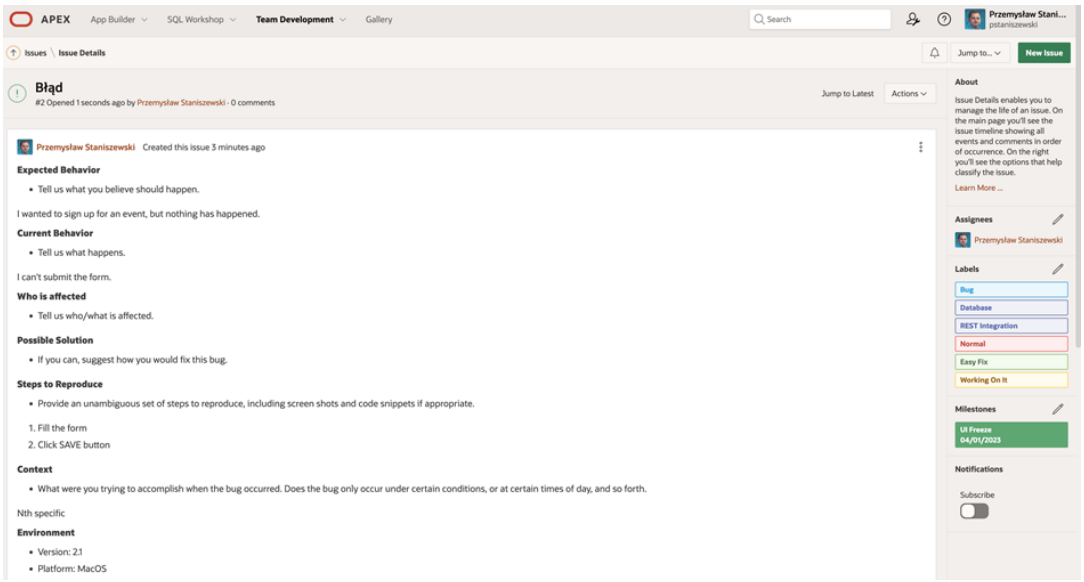



Abbildung 10.8: Issue-bezogene Kommunikation innerhalb der Team Development-Funktion.

Entwickler können auch ihnen zugewiesene Issues abonnieren, die sie per E-Mail informieren, wenn auf der Ebene des Issues eine Aktion stattfindet. Die Anwendung ermöglicht es den Nutzern, die Anzahl der E-Mails zu begrenzen, indem sie diese als stündliche oder tägliche Zusammenfassungen versenden.

Wir empfehlen dringend, „Team Development“ zu verwenden, wenn Ihr Team kein anderes Aufgabenverwaltungssystem implementiert hat oder Sie sehr einfache Tools wie Tabellenkalkulationen zur Verwaltung des Projekts verwenden.

### 10.4.5 Feedback

Feedback-Seiten sind eine integrierte Funktion in Oracle APEX, die es Endbenutzern ermöglicht, Feedback zur Funktionalität und Leistung der Anwendung zu geben. Feedback-Seiten können an die Bedürfnisse der Anwendung angepasst werden und können verschiedene Arten von Feedback enthalten, wie z.B. Fehlerberichte, Funktionsanforderungen und allgemeines Feedback.

Es ist von entscheidender Bedeutung, Informationen von den Endbenutzern zu erhalten und sie als eine Erweiterung des Entwicklungsteams zu behandeln. Je enger wir nicht nur intern, sondern auch mit den tatsächlichen Nutzern der Anwendung zusammenarbeiten, desto besser wird die Qualität der Anwendung sein. Außerdem kann das Team so die Anwendung an die tatsächlichen Bedürfnisse anpassen und das Vertrauen zwischen dem Entwicklungsteam und den Benutzern aufbauen (siehe Abbildung 10.9).

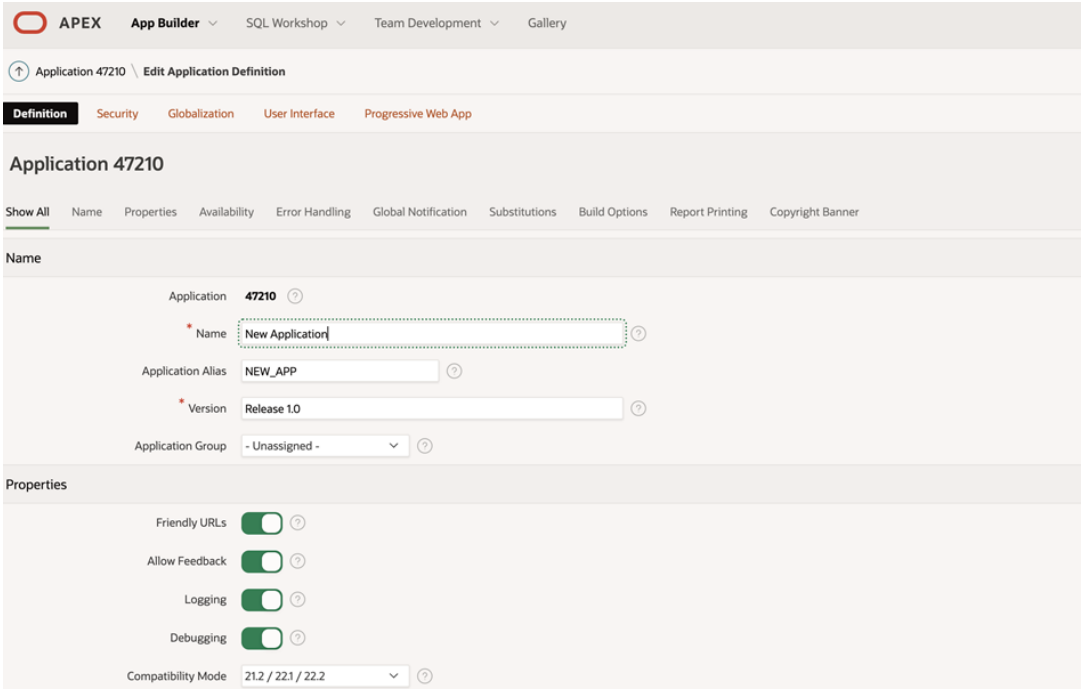



Abbildung 10.9: Feedback in der Anwendungsdefinition zulassen.

Um es zu aktivieren, müssen Sie bei der Anwendungserstellung eine Option auswählen. Wenn die Anwendung bereits existiert, gehen Sie zu „Edit Application Definition“ und markieren Sie die Option „Allow Feedback“ im Abschnitt Properties.

Nach der Aktivierung wird ein spezielles Symbol in das Anwendungsmenü eingefügt. Wenn der Endnutzer darauf klickt, kann er ein Feedback abgeben (siehe Abbildung 10.10).

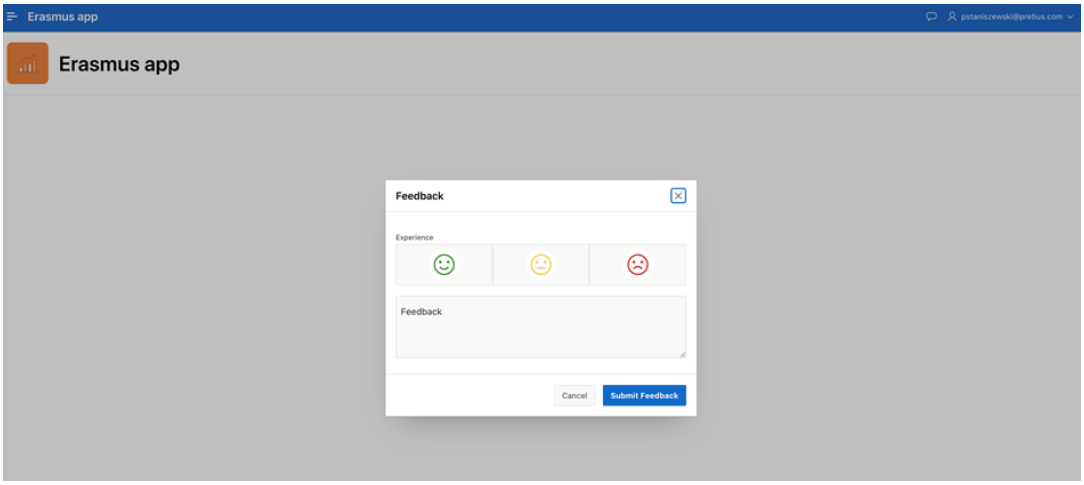



Abbildung 10.10: Feedback in der Anwendung einreichen.

Das Aussehen der Feedback-Seite kann angepasst werden, da es sich um eine Standard-APEX-Seite handelt. Alle Informationen, die der Endbenutzer zur Verfügung stellt, sind im Abschnitt Team Development verfügbar und können vom Team analysiert werden (es ist auch möglich, sie auf der Ebene einer bestimmten Anwendung in ihrem Verwaltungsabschnitt zu verfolgen). Außerdem erhalten die Entwickler detaillierte Informationen zu den gemeldeten Rückmeldungen (Anwendungs-ID, Seitennummer, Daten des Benutzers, Informationen über seine Umgebung usw.) und können sie in ein Issue umwandeln.

### 10.4.6 Standardisierung

Bei der Entwicklung von Anwendungen im Team ist es auch sehr wichtig, die Konsistenz zwischen den Entwicklungsstandards der verschiedenen Teammitglieder zu wahren. Wie bei jedem Aspekt der Zusammenarbeit ist die Kommunikation der Schlüssel zum Erfolg - zu Beginn des Projekts sollte sich das Team über die Art der Entwicklung und den Einsatz der APEX-Komponenten einigen. Zu den wichtigsten Aspekten dabei gehören:

- Namenskonventionen (für Elemente, Prozesse, Validierungen, dynamische Maßnahmen usw.);
- Seitennummerierung und Namensgebung (Alias);
- Erlaubte Elemente (z.B. verwenden wir für ein Verzeichnis nur interaktive Raster, jedes Formular muss als modale Seite angezeigt werden);
- UX/UI-Standards (z.B. Erscheinungsbild typischer Seiten, Ausrichtungsrichtlinien für verschiedene Datenzeiten, einheitliche Datenformate, empfohlene Symbole für bestimmte Aktionen usw.);
- Datenkonsistenz (z. B. einheitliche Verzeichnisse, die an einer Stelle für die gemeinsamen Komponenten definiert sind).

APEX stellt eine Reihe von Lösungen zur Verfügung, die den Teams beim Standardisierungsmanagement helfen. Die meisten davon sind im Abschnitt „Utilities“ auf der Ebene der ausgewählten Anwendung zu finden (siehe Abbildung 10.11).

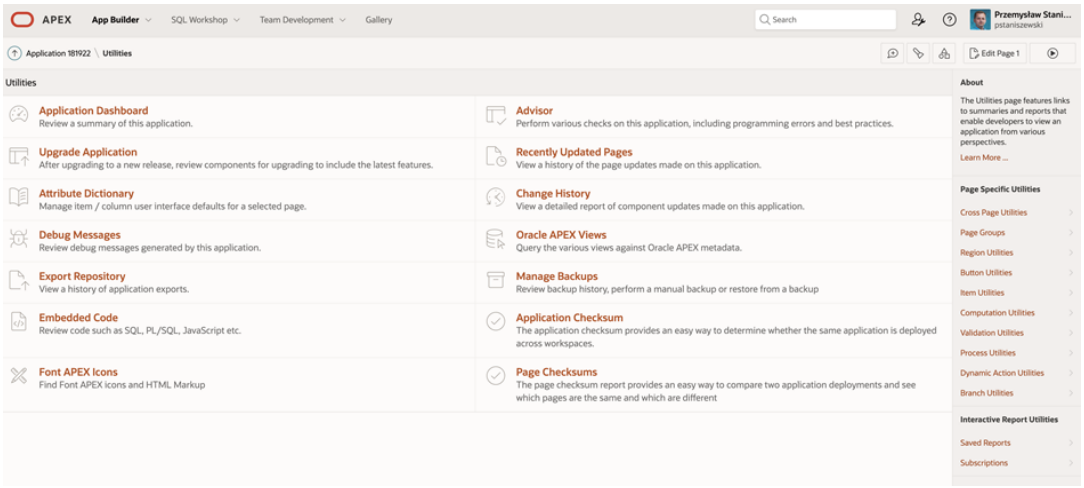



Abbildung 10.11: Utilities-Menü.

Zu den wichtigsten zu Beginn der Entwicklung gehören:

- Seitenübergreifende Berichte (Page Specific Utilities) - ermöglichen es Entwicklern, verschiedene APEX-Elemente (Seitenattribute, Berichte, Schaltflächen, Elemente usw.) in der Anwendung zu vergleichen, um die Konsistenz zu prüfen und sie auf einmal zu aktualisieren, ohne Attribute an mehreren Stellen manuell zu ändern.
- Änderungshistorie/kürzlich aktualisierte Seiten - ermöglicht die Verfolgung der von Entwicklern durchgeführten Aktionen.
- Standarddefinition (Attributverzeichnis) - ermöglicht die Erstellung eines gemeinsamen Verzeichnisses, um eine einheitliche Benennung, Formatierung, Datentyp usw. für Elemente zu gewährleisten, die an vielen Stellen in der Anwendung wiederkehren. Zum Beispiel kann „Update date“ in verschiedenen Berichten oder Elementen in der Anwendung vorkommen, aber wir wollen immer die gleiche Bezeichnung und das gleiche Datumsformat für jedes Vorkommen haben.

Natürlich bietet APEX viele zusätzliche Funktionen und Strategien, um die Konsistenz auf Datenbank- und Anwendungsebene zu gewährleisten. Diejenigen, die ihr Wissen erweitern möchten, sollten die APEX-Dokumentation lesen und fortgeschrittene Optionen wie Global Page, Blueprints, User Interface Defaults, Master Theme Application und Subscribe-Mechanismus, APEX Views oder Plug-ins prüfen.

## 10.5 Schlussfolgerungen

Als ausgereifte Entwicklungsplattform bietet Oracle APEX viele Funktionen, die bei der Zusammenarbeit helfen. Als Low-Code-Entwickler können wir diese nutzen und professionell APEX-basierte Anwendungen erstellen und pflegen, ohne die Plattform zu erweitern oder mit externen Tools zu integrieren. Wenn Sie neu in APEX und neu in Kooperationsstrategien in der Programmierung sind, beginnen Sie mit den grundlegendsten Schritten, die in diesem Kapitel beschrieben werden:

- Kommunikation und regelmäßige Treffen mit dem Team (live oder aus der Ferne - verwenden Sie Tools wie Google Meet, Zoom oder MS Teams für Video/Sprache und Slack, Google Chat oder MS Teams für schnelle Nachrichtenübermittlung);
- Beginnen Sie mit dem Einsatz von Kollaborationswerkzeugen, z.B. APEX Team Development, um die Entwicklung zu planen und den Fortschritt zu überprüfen;
- Beginnen Sie damit, Ihren Code und Ihre Standards zu dokumentieren. Und überprüfen Sie sie!

Die Zusammenarbeit bei der Low-Code-Entwicklung mag manchmal übertrieben klingen, aber sie ist einen Versuch wert und bringt genauso viel Gewinn wie bei der Standardprogrammierung.

Darüber hinaus können die meisten der typischen Kollaborationsstrategien wie „Code-Review“, „Pair Programming“ oder „Mob Programming“ auch auf der Low-Code-Plattform eingesetzt werden, auch wenn es sich bei dem überprüften Element manchmal nicht um reinen Code, sondern um einen Prozess, eine Seite oder eine Konfiguration handelt.

Je nach Umfang des Projekts oder der in der Organisation implementierten Standards müssen Sie zu einem bestimmten Zeitpunkt Ihren Kooperationsprozess erweitern. Bei großen Projekten müssen Sie beispielsweise Code-Repositories, automatisierte Tools für die Erstellung und Bereitstellung oder Software für das Testen und die Überprüfung der Codequalität einsetzen. Die Nutzung von Elementen professioneller CI/CD-Prozesse wie Git, Docker, Flyway, Liquibase, Jenkins, Gradle, SonarQube, Selenium, Cypress und vielen anderen ist mit der APEX-basierten Anwendung ebenfalls möglich. In vielen Fällen reichen jedoch die in der Plattform enthaltenen Elemente aus, um mit einem Team von Entwicklerkollegen eine bahnbrechende Anwendung zu erstellen.

## 10.6 Fragen

1. Was ist die Seitensperrfunktion und warum ist sie wichtig?
2. Wie können Entwickler während der Entwicklung mit APEX kommunizieren?
3. Was können Teammitglieder mit der Teamentwicklungsfunktion nutzen?

## 10.7 Antworten

1. Das Sperren von Seiten ist eine nützliche Funktion, die verhindert, dass mehrere Entwickler gleichzeitig dieselbe Seite bearbeiten und somit Konflikte vermeiden.
2. Entwickler können zu Seiten, Regionen, Elementen und vielen anderen Anwendungselementen Kommentare hinzufügen. Kommentare können verwendet werden, um anderen Entwicklern Kontext zu liefern, die Gründe für eine Designentscheidung zu erläutern oder einfach Änderungen an einer Anwendung zu dokumentieren. Durch die Verwendung von Kommentaren können Entwickler besser miteinander kommunizieren und die Gesamtqualität der Software sicherstellen.
3. Team Development ermöglicht es Teammitgliedern, Entwicklungsprojekte zu verwalten, insbesondere: Meilensteine und Projektphasen zu definieren, Aufgaben/Issues zu erstellen, zuzuweisen und zu überwachen, Klassifizierungen mit Hilfe von Label-Gruppen und Labels anzuwenden, Issue-Vorlagen zu erstellen, um die Erstellung von Aufgaben oder Berichten konsistent zu gestalten, Issues mit einer Anwendung und einer Seite zu verknüpfen, bestehende Issues zu kommentieren und zu diskutieren, ihren Status zu ändern und den Fortschritt der Projektphase zu überwachen und verschiedene Arten von Issues zu abonnieren.

# 11. Wie kann man von der Galerie der Anwendungen und Plug-ins profitieren?

VJERAN STRAHONJA, DIJANA OREŠKI

Dieses Kapitel befasst sich mit Starter-Apps, Beispiel-Apps und Plug-ins in Oracle APEX. Alle Oracle APEX-Workspaces werden mit Beispiel-Apps und Starter-Apps (siehe Abbildung 11.1) und Plug-ins angeboten.

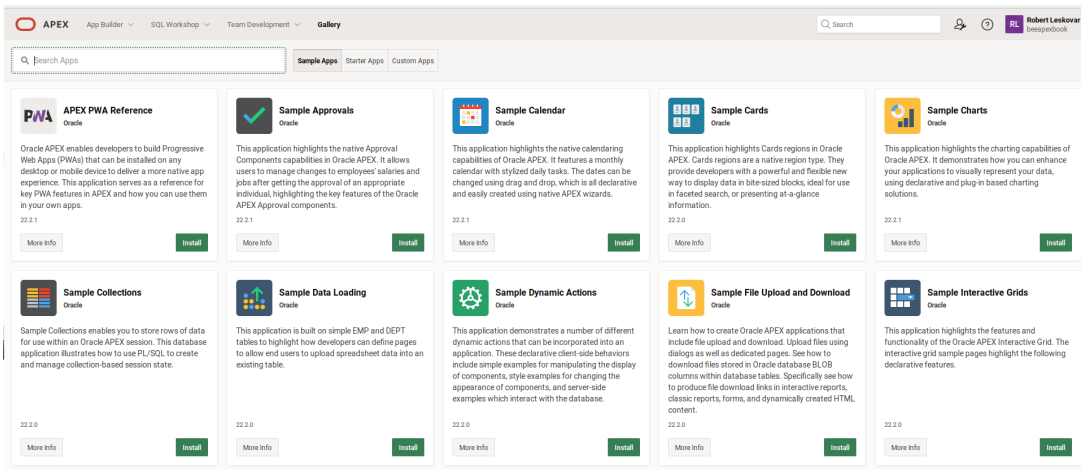



Abbildung 11.1: Oracle APEX-Workspaces werden mit Sample Apps und Starter Apps angeboten.

Starter-Apps sind eigenständige Punktlösungen, die einfache Anforderungen erfüllen und keine komplexe Lösung erfordern. Beispielanwendungen hingegen konzentrieren sich auf eine bestimmte Funktionalität und bieten Anleitungen zur Nutzung einer bestimmten Funktion. Plug-ins ermöglichen es den Benutzern, Anwendungen mit benutzerdefinierten Funktionen zu erweitern.

## 11.1 Wie installiert man Beispiel- und Starter-Apps?

Benutzer können Beispiel- und Starter-Apps aus der Galerie herunterladen. Wenn vom Instanzadministrator aktiviert, können Apps über die Galerie installiert werden.

Um Anwendungen aus der Galerie zu **installieren**:

1. Rufen Sie die Startseite des Arbeitsbereichs auf.
2. Wählen Sie **Gallery**. Die Beispielanwendungen, Starter-Apps und benutzerdefinierten An-

wendungen in Form von Karten werden angezeigt, wenn die Seite **Gallery** zum ersten Mal geladen wird.

3. Wählen Sie **Sample Apps** oder **Starter Apps**. Karten werden verwendet, um die Anwendungen aufzulisten. Sofern das Programm nicht bereits installiert wurde, enthält jede Karte eine Schaltfläche Installieren. Die Symbole Ausführen und App entfernen sind vorhanden, wenn die Anwendung bereits installiert wurde.
4. Wählen Sie **Install App**, um die Installation der App durchzuführen
5. Erweitern Sie den Abschnitt Erweiterte Einstellungen. (Nicht obligatorisch)
   a. Geben Sie an, ob die Anwendungs-ID manuell oder automatisch (Standard) zugewiesen wird.
   b. Ändern Sie das Parsing-Schema.
6. Klicken Sie auf Anwendung installieren, um die Installation abzuschließen.

Durch Auswahl des Remove App Symbols in der Galerie können die Benutzer die Apps deinstallieren, die von dort aus installiert wurden. Anwendungen, die von GitHub importiert und heruntergeladen wurden, haben kein Deinstallationssymbol, so dass sie gelöscht werden müssen, um aus dem Arbeitsbereich deinstalliert zu werden. Schritte zum **Deinstallieren** von Anwendungen aus der Galerie:

1. Rufen Sie die Startseite des Arbeitsbereichs auf.
2. Wählen Sie das Galerie-Symbol. Die Karten Beispiel-Apps, Starter-Apps und Benutzerdefinierte Apps werden angezeigt, wenn die Seite Galerie zum ersten Mal geladen wird.
3. Wählen Sie Starter Apps oder Sample Apps. Direkt aus der Galerie heruntergeladene Programme haben ein Symbol zum Entfernen und ein Ausführen-Symbol sichtbar.
4. Um die Anwendung zu entfernen, klicken Sie auf Remove.

## 11.2 Starter-Apps

Starter-Apps sind nützliche Anwendungen, die eigenständige Lösungen für einfache Anforderungen bieten, die keine umfangreiche und unnötig komplexe Antwort erfordern. Diese Apps können „out of the box“ verwendet oder mit Ihren eigenen Funktionen erweitert werden (siehe Abbildung 11.2).

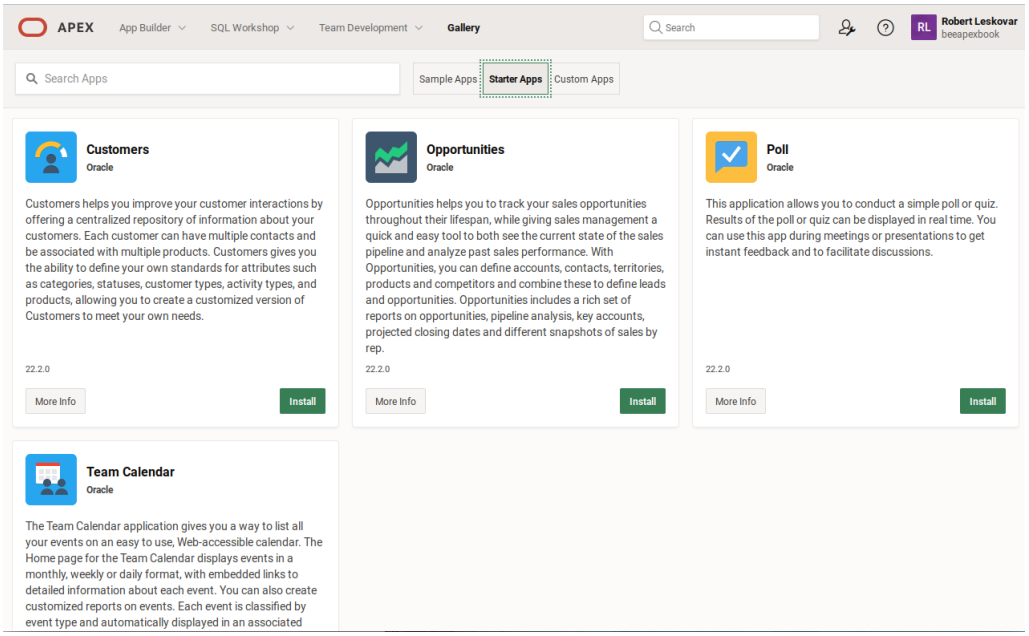



Abbildung 11.2: Starter-Apps in der Galerie.

Beispiele für Starter-Apps sind:

- **Customers** - Durch die Bereitstellung einer konsolidierten Datenbank mit Kundendaten ermöglicht die App Customers den Benutzern, die Interaktion mit Kunden zu verbessern. Jeder Kunde kann mit mehreren Kontakten und Artikeln verbunden sein. Mit Customers können Sie Ihre eigenen Kriterien für Funktionen wie Kategorien, Status, Kundentypen, Aktivitätstypen und Waren festlegen, so dass Sie eine spezielle Version von Customers entwerfen können, die Ihren Anforderungen entspricht.
- **Opportunities** - Diese Anwendung bietet dem Vertriebsmanagement eine schnelle und einfache Lösung, um sowohl den aktuellen Zustand der Vertriebs-Pipeline zu sehen als auch die vergangene Vertriebsleistung zu analysieren. Sie hilft Ihnen, Ihre Verkaufschancen während ihres gesamten Lebenszyklus zu verfolgen. Sie können Leads und Opportunities definieren, indem Sie Kunden, Kontakte, Regionen, Waren und Wettbewerber kombinieren. Opportunities bietet eine umfassende Sammlung von Informationen über Opportunities, Pipeline-Recherche, wichtige Kunden, voraussichtliche Abschlusstermine und andere Schnappschüsse von Vertriebsmitarbeitern.
- **Poll** – Benutzer können mit dieser Anwendung eine schnelle Umfrage oder einen Test erstellen. Die Ergebnisse der Umfrage oder des Quiz können sofort eingesehen werden. Dieses Tool kann verwendet werden, um Diskussionen zu erleichtern und sofortiges Feedback während Sitzungen oder Präsentationen zu erhalten.
- **Team Calendar** - Mit Hilfe der Teamkalender-App können Benutzer alle Ereignisse in einem einfach zu bedienenden, webbasierten Kalender auflisten. Auf der Startseite des Teamkalenders werden die Ereignisse in monatlicher, wöchentlicher oder täglicher Form aufgelistet, mit integrierten Links zu genaueren Details für jedes Ereignis. Darüber hinaus können Sie einzigartige Berichte über Ereignisse erstellen. Jedes Ereignis wird entsprechend seiner Kategorie eingeordnet, und die entsprechende Farbe wird automatisch angezeigt. Benutzer können neue Ereignistypen entwickeln oder die Attribute bereits vorhandener Ereignistypen ändern. Benutzer haben die Möglichkeit, ihre eigenen Gruppen zu erstellen, um ihre Bedürfnisse besser zu erfüllen, und sie können E-Mails an Einzelpersonen oder Gruppen senden, um sie über bevorstehende Treffen zu informieren.

Hier ein Beispiel für die Installation einer Anwendung für Team Calendar (siehe Abbildung 11.3).

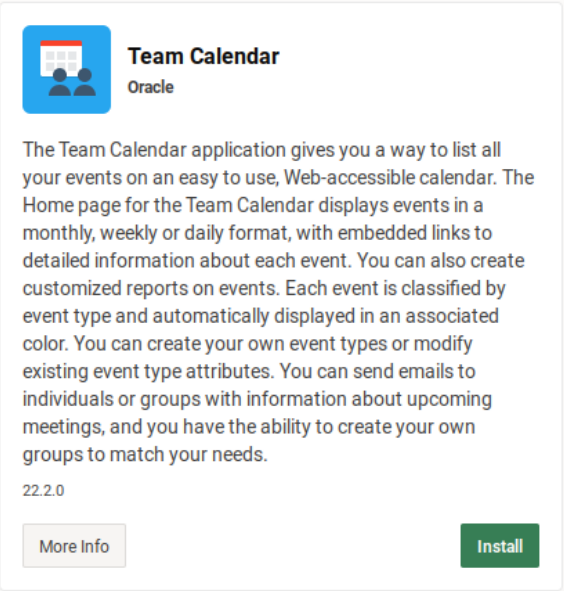



Abbildung 11.3: Installation der Starter-App Team Calendar.

Klicken Sie auf Install.

## 11.3 Beispiel-Apps

Beispiel-Apps werden erstellt, um bestimmte Funktionen hervorzuheben und als Entwicklerhandbuch für die Nutzung einer bestimmten Funktion zu dienen (siehe Abbildung 11.4).

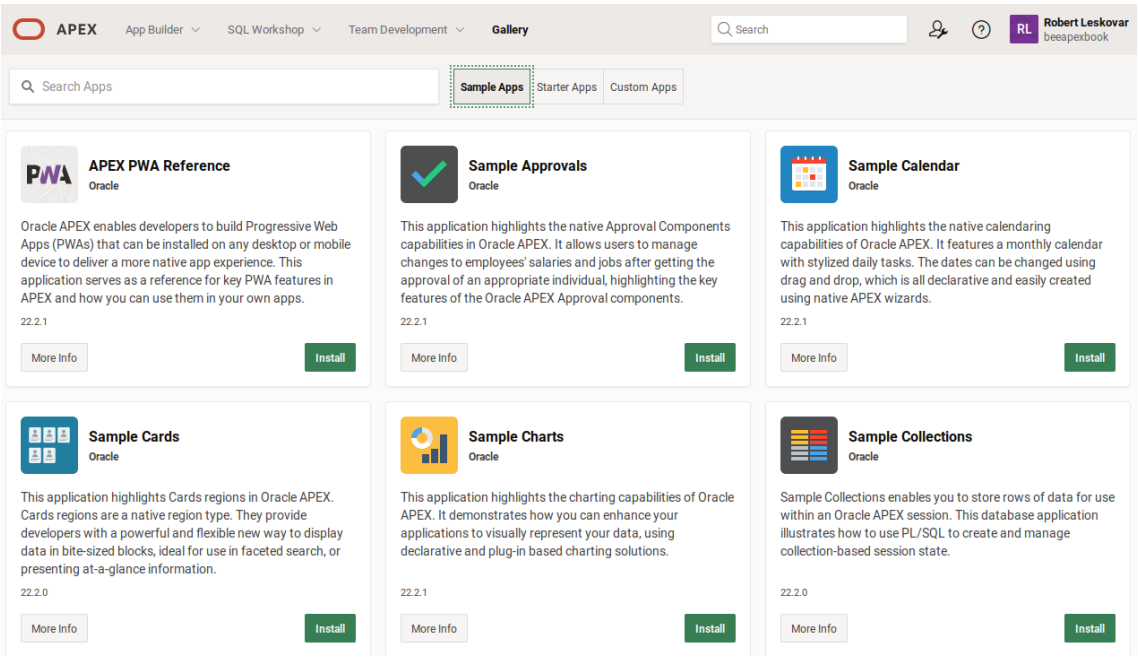



Abbildung 11.4: Beispiel-Apps.

Unter anderem können wir wählen:

- **APEX PWA Reference** - Um ein nativeres App-Erlebnis zu bieten, können Entwickler mit Oracle APEX Progressive Web Apps (PWAs) erstellen, die auf jedem Desktop oder Mobilgerät eingesetzt werden können. Die wichtigsten PWA-Funktionen in APEX und wie Sie sie in Ihren eigenen Apps anwenden können, sind in dieser Anwendung enthalten.
- **Sample Approvals** - Diese Anwendung zeigt die Fähigkeiten der nativen Genehmigungskomponenten von Oracle APEX. Die Komponenten von Oracle APEX Consent werden hervorgehoben, die es den Benutzern ermöglichen, Änderungen an den Zahlungen und Aufträgen der Mitarbeiter zu verwalten, nachdem sie von der richtigen Person genehmigt wurden.
- **Sample Calendar** - Diese Anwendung zeigt die integrierten Kalenderfunktionen von Oracle APEX. Ein Monatskalender mit gestylten täglichen Aufgaben ist enthalten. Drag and Drop, das vollständig deklarativ ist und schnell mit nativen APEX-Assistenten erzeugt wird, kann zur Aktualisierung der Daten verwendet werden.
- **Sample Cards** - In Oracle APEX hebt diese Anwendung die Cards-Regionen hervor. Regionen mit Karten sind eine Form von nativen Regionen. Sie bieten Entwicklern eine starke und anpassungsfähige neue Methode zur Präsentation von Daten in mundgerechten Bits, die sich perfekt für die Verwendung bei der Facettensuche oder die schnelle Präsentation von Informationen eignet.
- **Sample Charts** - Diese Anwendung stellt die Diagrammfunktionen von Oracle APEX vor. Es wird erklärt, wie Sie deklarative und Plug-in-basierte Diagrammwerkzeuge verwenden können, um Ihre Anwendungen so zu verbessern, dass sie Ihre Daten visuell darstellen.
- **Sample Collections** - Diese Anwendung bietet Benutzern die Möglichkeit, Datenzeilen in Beispielsammlungen zur Verwendung in einer Oracle APEX-Session zu speichern. Diese Datenbankanwendung demonstriert die Erstellung und Verwaltung von sammlungsbasierten

Session-States mit PL/SQL.

- **Sample Data Loading** - Diese Anwendung wurde unter Verwendung einfacher EMP- und DEPT-Tabellen erstellt, um zu zeigen, wie Entwickler Seiten entwerfen können, die es Benutzern ermöglichen, Tabellenkalkulationsdaten in eine bereits vorhandene Tabelle einzufügen.
- **Sample Dynamic Actions** - Diese Anwendung zeigt eine Vielzahl von dynamischen Aktionen, die zu Anwendungen hinzugefügt werden können. Diese deklarativen clientseitigen Verhaltensweisen umfassen einfache Beispiele für die Änderung der Anzeige von Komponenten, Stilbeispiele für die Anpassung des Aussehens von Komponenten und serverseitige Beispiele für die Interaktion mit Datenbanken.
- **Sample File Upload and Download** - Lernen Sie, wie Sie Oracle APEX-Anwendungen mit Datei-Upload- und Download-Funktionalität erstellen. Nutzen Sie sowohl dedizierte Seiten als auch Dialogfelder zum Hochladen von Dateien. Sehen Sie sich den Prozess zum Herunterladen von Dateien aus BLOB-Spalten in Datenbanktabellen der Oracle-Datenbank an. Sehen Sie insbesondere, wie Sie Links zum Herunterladen von Dateien in Formularen, interaktiven Berichten, traditionellen Berichten und dynamischen HTML-Inhalten erstellen können.
- **Sample Interactive Grids** - Die Merkmale und Funktionen des Oracle APEX Interactive Grid werden in dieser Anwendung hervorgehoben. Diese Funktionen werden auf den Beispielseiten des interaktiven Gitters hervorgehoben.
- **Sample Maps** - In dieser Anwendung finden Sie zahlreiche Beispiele für die Anzeige von Koordinatendaten auf einer Karte. Verwenden Sie die Heatmap-Funktion, Linien, Polygone oder Kartenmarker. Die Oracle Spatial-Funktionen (die in jeder Oracle-Datenbank verfügbar sind) können problemlos in die APEX Map Region integriert werden, um eine „Within Distance Search“, eine „Nearest Neighbor Search“ oder andere räumliche Analysen durchzuführen.
- **Sample Master Detail** - Diese Anwendung zeigt die nativen Stammdatenfunktionen von Oracle APEX. In der Anwendung gibt es vier mögliche Layouts für Stammdetail-Seiten. Die ersten beiden Layouts verwenden editierbare interaktive Raster, um die Stammdaten auf einer einzigen Seite anzuzeigen. In den letzten beiden Layouts werden die Stammdaten auf zwei Seiten dargestellt, wobei eine Kombination aus editierbaren interaktiven Rastern, Formulareingaben, alten Berichten und modalen Popups verwendet wird.
- **Sample Reporting** - Diese Anwendung demonstriert die Berichtsfunktionen von Oracle APEX. Mit Hilfe von SQL können Benutzer deklarativ interaktive Berichte, interaktive Raster, Berichte mit Facettensuche, Kartenberichte und klassische Berichte konstruieren.
- **Sample REST Services** - Diese Anwendung demonstriert, wie Oracle APEX zur Verbindung mit externen REST-Diensten verwendet werden kann. Oracle.example.hr, ein prototypischer RESTful Service, wird von der Anwendung verwendet. Die Beispiele in dieser Anwendung zeigen, wie man filtert, wie man eine Seitennummerierung hinzufügt und wie man einen einfachen tabellarischen Bericht über REST-Service-Daten erstellt.
- **Sample Trees** - Erfahren Sie, wie Sie eine SQL-Abfrage verwenden, um ein Tree Control zu erstellen. Die Oracle APEX-Anwendung in diesem Beispiel demonstriert mehrere Möglichkeiten zur Einbindung von Tree Controls.
- **Universal Theme Reference** - Diese App bietet eine einfache Möglichkeit, durch die zahlreichen Vorlagen, Vorlagenauswahlen und Themenstile zu navigieren und führt die Benutzer in Universal Theme ein. Die Beispiele zeigen, wie Sie Ihr Seitenlayout schnell verwalten können, um eine atemberaubende Anwendung zu erstellen.

## 11.4 Plug-ins

Plug-ins ermöglichen die einfache Erweiterung des Application Express-Frameworks um benutzerdefinierte Elementtypen, Regionstypen, Prozesse und dynamische Aktionen. Einmal definiert, werden Plug-in-basierte Komponenten ganz ähnlich wie Standardkomponenten von Application

Express erstellt und gepflegt. Mit Plug-ins können Entwickler hochgradig angepasste Komponenten erstellen, um die Funktionalität, das Aussehen und die Benutzerfreundlichkeit ihrer Anwendungen zu verbessern. Plug-Ins ermöglichen es den Benutzern, APEX-Anwendungen mit Plug-Ins zusätzliche Funktionalität hinzuzufügen, die nicht in die Plattform eingebaut ist. Die APEX-Community hat eine große Anzahl verschiedener Plug-Ins entwickelt, die für die Erweiterung bestehender Funktionen oder das Hinzufügen neuer Funktionen nützlich sind. Es gibt hunderte von Plugins für Oracle Apex, die in fünf Typen unterteilt sind:

- Dynamic action
- Region
- Item
- Process
- Authentication

Oracle Application Express bietet innerhalb jeder Kategorie verschiedene Plugins an. Plug-ins geben Ihnen die Möglichkeit, Ihrer Anwendung deklarativ neue Funktionen hinzuzufügen und so die eingebauten Typen zu erweitern. Da Plug-ins wiederverwendet werden sollen, können Entwickler sie über das Plug-in-Repository mit der Oracle Application Express Plug-in-Community teilen und sie in verschiedene Arbeitsbereiche exportieren und importieren. Die Schritte zur Implementierung eines Plug-ins sind wie folgt

- Fügen Sie ein Plug-in zu Ihrem Anwendungsarbeitsbereich hinzu oder erstellen Sie eines von Grund auf.
- Um das Plug-in zu verwenden, bearbeiten oder erstellen Sie eine dynamische Aktionsart, ein Element, einen Bereich, einen Prozess oder ein Berechtigungsschema.
- Starten Sie Ihre Anwendung, um die Funktionsfähigkeit des Plug-ins zu überprüfen.

Ein zentraler Ort für Entwickler, an dem sie Plug-ins freigeben und herunterladen können, ist das Plug-in-Repository. Das Repository ist im Oracle Technology Network zu finden.

- Das Repository für Plug-ins finden Sie auf der Seite für Shared Components:
  - Klicken Sie auf der Hauptseite des Arbeitsbereichs auf App Builder.
  - Entscheiden Sie sich für eine Anwendung.
  - Wählen Sie Shared Components auf der Startseite der Anwendung.
- Die Seite für Shared Components wird angezeigt.
- Wählen Sie Plug-ins aus dem Other Components Menü.
- Drücken Sie die Schaltfläche View Plug-in-Repository.

Das Repository für Oracle Application Express Plug-ins wird angezeigt. So greifen Sie auf die Plug-ins Seite zu:

- Rufen Sie die Seite Shared Components auf.
  - Klicken Sie auf der Hauptseite des Arbeitsbereichs auf App Builder.
  - Wählen Sie eine Anwendung.
  - Wählen Sie Shared Components auf der Startseite der Anwendung.
- Wählen Sie Plug-ins aus dem Other Components Menü. Die Registerkarte Plug-ins wird automatisch ausgewählt, wenn die Seite Plug-ins geladen wird. Die gesamte Liste der Plug-ins wird angezeigt.

So erstellen Sie ein Plug-in:

- Gehen Sie auf die Seite Shared Components.
  - Klicken Sie auf der Hauptseite des Arbeitsbereichs auf App Builder.
  - Wählen Sie eine Anwendung.
  - Wählen Sie Shared Components auf der Startseite der Anwendung. Die Seite Shared Components wird angezeigt.
- Wählen Sie Plug-ins unter Other Components,
- Klicken Sie auf Create. Der Assistent zum Erstellen von Plug-ins wird angezeigt.
- Klicken Sie auf Next, nachdem Sie unter Create Plug-in die Methode zum Erstellen eines

Plug-ins ausgewählt haben.
- Innerhalb von Name:
  - Name - Geben Sie den Namen des Plug-ins ein.
  - Internal Name - Geben Sie den internen Namen des Plug-ins an. Der Name sollte innerhalb der Anwendung eindeutig sein.
  - Type - Entscheiden Sie, welche Art von Komponente dieses Plug-in nutzen kann. Die Optionen unter Rückrufe und Standardattribute variieren je nach dem von Ihnen gewählten Plug-in-Typ. Weitere Informationen finden Sie in der Hilfe auf Feldebene.
  - Category - wird nur angezeigt, wenn die dynamische Aktion die angegebene Art ist. Wählen Sie auf der Benutzeroberfläche die Kategorie, unter der das Plug-in aufgeführt ist.
- Innerhalb von Source:
  - PL/SQL Code - Geben Sie einen anonymen PL/SQL-Codeblock ein, der die Schritte enthält, die zum Rendern, Validieren, Starten und Ausführen der Callbacks dieses Plug-ins erforderlich sind. Sie können diesen Code aus Leistungsgründen alternativ in einem PL/SQL-Paket in der Datenbank speichern.
  - Do not validate PL/SQL code - Um den PL/SQL-Code nur zur Laufzeit zu analysieren, wählen Sie diese Option. Andernfalls wird das Plug-in erstellt, sobald der Code geparst wurde.
- Konfigurieren Sie in Callbacks die Attribute. Der Plug-in-Typ bestimmt die Attribute, die angezeigt werden. Sehen Sie sich die Hilfe auf Feldebene an, um Beispiele zu prüfen und mehr über Attribute zu erfahren.
- Wählen Sie in User Interfaces, welche Anzeigen der App Builder für dieses Plug-in unterstützen muss. Die Optionen umfassen:
  - Desktop
  - Mobile
- Wählen Sie unter Standard Attributes die Attribute aus, die für dieses Plug-in gelten. Standardattribute machen bestimmte Plug-ins überflüssig.
- In Informationen:
  - Version - Geben Sie eine Zeichenkette ein, um die Plug-in-Version anzugeben.
  - About URL - Geben Sie eine URL ein, die zur Homepage des Plug-in-Autors oder zu weiteren Details über das Plug-in führt.
- Klicken Sie auf Create Plug-in.

Bitte beachten Sie, daß die obigen Angaben für die APEX-Version 22.1 gelten. Die neuere Version 23.1 bietet eine noch bessere Möglichkeit zur Erstellung von Plug-Ins. Um ein Plug-in zu importieren:

- Gehen Sie auf die Seite Shared Components:
  - Klicken Sie auf der Hauptseite des Arbeitsbereichs auf App Builder.
  - Wählen Sie eine Anwendung.
  - Wählen Sie Shared Components auf der Startseite der Anwendung.
- Wählen Sie Plug-ins innerhalb von Other Components.
- Wählen Sie Import. Die Seite Plug-in importieren wird angezeigt.
- Für Specify File:
  - Import file - Geben Sie den Namen der Importdatei ein oder navigieren Sie dorthin.
  - File Type - Plug-in wählen.
  - File Character Set - Wählen Sie die Zeichensatzkodierung für Importdateien.
  - Klicken Sie auf Next.
- Wählen Sie bei Datei-Import-Bestätigung die Option Next.
- Zur Installation wählen Sie Install Plug-in.

So exportieren Sie ein Plug-in von der Plug-in-Seite:

- Gehen Sie auf die Seite Shared Components:
  - Klicken Sie auf der Hauptseite des Arbeitsbereichs auf App Builder.
  - Wählen Sie eine Anwendung.
  - Wählen Sie Shared Components auf der Startseite der Anwendung.
- Wählen Sie Plug-ins innerhalb von Other Components.
- Wählen Sie unter Tasks die Option Export Plug-in.
- Auf der Seite Export Plug-in:
  - Application - Wählen Sie die Anwendung, aus der das Plug-in exportiert werden soll.
  - Plug-in - Wählen Sie das Plug-in.
  - File Format - Wählen Sie das Dateiformat für den Export.
  - Wählen Sie Export
- Wählen Sie Export.

In den nächsten Zeilen wird ein Beispiel für Plug-ins gegeben. Das Beispiel bezieht sich auf die aktualisierte sample-calendar-App und das dynamic action-Plugin. Dieses Beispiel-Plugin kann verwendet werden, um Tage in der neuen Datumsauswahl zu formatieren. Benutzer können es deaktivieren, Tooltips oder Klassen zu jedem Tag im Tagesraster der Datumsauswahl hinzufügen. Es gibt keine Voraussetzungen für die Verwendung dieses Plugins. Wie man es benutzt:

- Eine Anwendung erstellen
- Importieren Sie das Plug-in in Shared Components - Plug-ins
- Neue Datumsauswahl hinzufügen
- Fügen Sie eine dynamische Aktion beim Laden der Seite hinzu und wählen Sie als Aktion das importierte Plug-in
- Legen Sie eine URL eines iCalendar-Standarddateiformats fest oder geben Sie eine SQL-Abfrage ein, die die Kalendertage formatieren soll.
- Wählen Sie als Affected Element die gewünschte Datumsauswahl und führen Sie die Seite aus

Dieses Kapitel hat gezeigt, dass Plugins mehrere Vorteile bieten können. Keine noch so leistungsfähige Software kann alle Funktionen für jeden Benutzer bereitstellen. Die Lücke zwischen Formular und Funktion wird durch Plugins geschlossen. Sie machen es einfacher, Software und Anwendungen um bestimmte Funktionen zu erweitern. Plugins sind also zeitsparend. Der größte Vorteil des Einsatzes von Plugins besteht darin, dass sie den Zeitaufwand für die Entwicklung reduzieren und damit die Gesamtkosten senken. Darüber hinaus ermöglichen Plugins den Nutzern eine individuelle Anpassung der Merkmale und Funktionen. Die meisten Plugins ermöglichen es, bestimmte Einstellungen zu aktivieren und zu deaktivieren. Es ist ein einfacher Prozess, ein Plugin zu entfernen, wenn ein Bedarf dafür besteht.

## 11.5 Fragen

1. Was sind Beispiel-Apps und was sind ihre Vorteile?
2. Was sind Starter-Apps und was sind ihre Vorteile?
3. Was sind die Vorteile von Plugins?

## 11.6 Antworten

1. Beispiel-Apps werden erstellt, um bestimmte Funktionen hervorzuheben, und dienen als Handbuch für Entwickler zur Nutzung einer bestimmten Funktion.
2. Starter Apps sind nützliche Anwendungen, die eigenständige Punktlösungen bieten, die für einfache Anforderungen erstellt wurden, die keine umfangreiche und unnötig komplexe Lösung erfordern.
3. Plug-Ins ermöglichen es Entwicklern, hochgradig angepasste Komponenten zu erstellen, um die Funktionalität, das Aussehen und die Benutzerfreundlichkeit ihrer Anwendungen

zu verbessern. Plug-Ins ermöglichen es den Benutzern, APEX-Anwendungen zusätzliche Funktionen hinzuzufügen, die nicht eingebaut sind.



# 12. Wie verwaltet man verpackte und mehrsprachige Anwendungen?

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA

## 12.1 Anwendung und verpackte Anwendung

In diesem Abschnitt lernen Sie, wie Sie das Potenzial von Quick SQL nutzen können, um eine Anwendung fast ohne DDL-Kenntnisse und vollständig mit Oracle APEX zu erstellen. Nach der Generierung der Anwendung aus Quick SQL werden wir die verpackte Anwendung vorbereiten und sie in einen neuen Arbeitsbereich (d.h. Produktionsstandort") migrieren.

In Kapitel 11 haben wir Anwendungen vorgestellt, die von Oracle in APEX geliefert werden. Diese Anwendungen werden in zwei Gruppen eingeteilt: a) Produktivitätsanwendungen und b) Beispielanwendungen. Produktivitätsanwendungen sind auf spezifische Geschäftsanforderungen ausgerichtet. Die eingebetteten Funktionen erfüllen mehr oder weniger vollständig die Anforderungen, die sich aus der Geschäftssituation ergeben. Beispielanwendungen demonstrieren verschiedene Funktionalitäten. Beide können Entwickler dazu inspirieren, sie anzupassen und die geschäftlichen Anforderungen zu erfüllen. Diese Anwendungen sind Beispiele für verpackte Anwendungen. Wenn Sie diese verpackten Anwendungen installieren (der APEX-Begriff lautet ”importieren”), können Sie feststellen, dass Datenbankobjekte (Tabellen, Sequenzen, Trigger, Funktionen, Prozeduren und Pakete) erstellt und Daten in Tabellen eingefügt werden.

Die kürzeste Definition von **verpackte Anwendung** besagt, dass es sich um eine vorgefertigte Anwendung handelt, die in einem Oracle APEX-Arbeitsbereich installiert und konfiguriert werden kann. APEX verwendet zwei wichtige Konzepte, die eng mit der Migration von Anwendungen (Export und Import) zusammenhängen: a) unterstützende Objekte und b) gemeinsame Komponenten. Unterstützende Objekte steuern die Installation, das Upgrade und die Deinstallation, indem sie Voraussetzungen, Substitutionszeichenfolgen, Build-Optionen, Validierungen vor der Installation, Skripte und Meldungen definieren. Gemeinsame Komponenten definieren Anwendungslogik, Sicherheit, Navigation und Suche, Benutzeroberfläche, Dateien und Berichte, Datenquellen, Workflows (falls installiert), Globalisierung und andere Komponenten. Verpackte Anwendungen sind also solche mit definierten unterstützenden Objekten. In diesem Moment fragen Sie sich vielleicht: "Warum und wie können Entwickler benutzerdefinierte verpackte Anwendungen erstellen?".

Die Antwort auf die Frage nach dem "Warum”: Entwickler würden eine benutzerdefinierte verpackte Anwendung erstellen, wenn die Anwendung in einen neuen Arbeitsbereich mit unterstützenden Objekten und gemeinsam genutzten Komponenten migriert werden soll. Definitionen von Datenbankobjekten (mit DDL-Anweisungen) und Operationen mit diesen Objekten (DML-Anweisungen) werden während der Installation, Aktualisierung und Deinstallation ausgeführt.

Vereinfacht ausgedrückt: Während der Installation könnten Tabellen erstellt und mit Daten gefüllt werden, während der Aktualisierung könnten Tabellen geändert werden und während der Deinstallation könnten Tabellen abgeschnitten oder gelöscht werden. Ähnliches könnte für andere Datenbankobjekte wie Trigger, Sequenzen, gespeicherte Prozeduren, Funktionen und Pakete erzwungen werden. Der Hauptvorteil einer verpackten Anwendung besteht darin, dass sie mit Datendefinitionen und Daten selbst betriebsbereit ist.

Die Antwort auf die Frage nach dem "Wie" umfasst die folgenden Schritte:

- den Anwendungsbereich festlegen
- den Anwendungsbereich festlegen
- Skript zur Definition von Tabellen erstellen und ausführen
- Skript zum Einfügen von Beispieldaten erstellen und ausführen
- Anwendung erstellen
- Anpassungen in der Anwendung vornehmen
- verpackte Anwendung vorbereiten und im neuen Arbeitsbereich testen

Bevor wir Schritt für Schritt vorgehen, wollen wir uns den Assistenten für den Import und Export von Anwendungen in APEX ansehen. Um die Anwendung in einen neuen Arbeitsbereich zu verschieben, muss die Anwendung exportiert und dann importiert werden. Der Export-Assistent in APEX schreibt eine Textdatei (PL/SQL-Anweisungen ) a) auf den Computer des Entwicklers oder b) in einen externen Arbeitsbereich über einen REST-Dienst (siehe Abbildung 12.1).

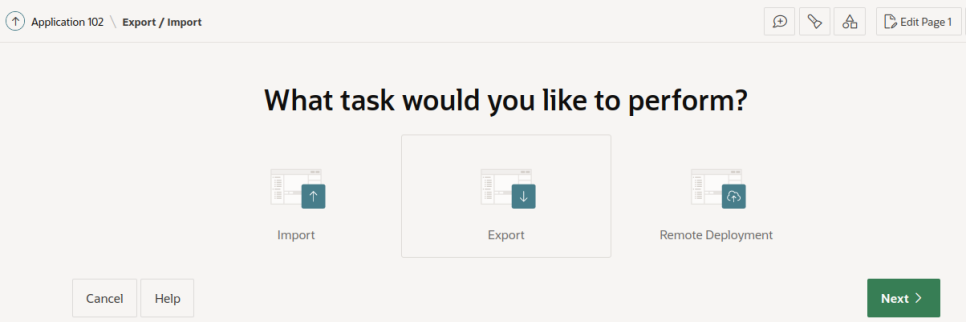



Abbildung 12.1: Assistent für den Import und Export von Anwendungen.

Es gibt mehrere Optionen für den Export, wie z.B. Aufteilung in mehrere Dateien, Auswahl des Zeichensatzes der Datei, öffentliche oder private Berichte und ähnliches. In den meisten Fällen lässt der Einsteiger APEX einfach die Standardeinstellungen übernehmen. Dann meldet sich der Entwickler in einem neuen Arbeitsbereich an und startet den Importassistenten, der die exportierte Textdatei lädt. Definitionen von Tabellen, Views, Triggern, Sequenzen, Funktionen, Prozeduren und Paketen werden nicht mit der Anwendung migriert, wenn wir keine unterstützenden Objekte vorbereiten. Wenn der Zielarbeitsbereich bereits Datenbankobjektdefinitionen und Daten in den Tabellen enthält, sollten wir unterstützende Objekte vorbereiten, um solche Situationen zu bewältigen.

Im ersten Teil dieses Kapitels werden die APEX-Assistenten so weit wie möglich genutzt. Dann werden einfache SQL INSERT-Anweisungen vorgestellt, die nicht vom Assistenten in APEX unterstützt werden. Um INSERT-Anweisungen zu generieren, kann der Leser Daten in eine Tabellenkalkulation importieren und dann INSERT-Anweisungen mit anderen Tools (z.B. SQL Developer oder TOAD) exportieren. In der Anwendung in diesem Kapitel wird eine Seitenkomponente namens lists of values (LOV) vorgestellt. Sie gehört zu den gemeinsam genutzten Komponenten. Lists of values dienen zwei Zwecken: Der Benutzer kann ein Formularfeld auf der Webseite leicht ausfüllen (Dropdown, Optionsfeld) und mit nur vorgegebenen Werten, um Tippfehler zu vermeiden. Es gibt zwei Arten von lists of values: dynamische basieren auf SQL-Abfragen, während statische einen festen Satz von Werten enthalten. Der Vorteil einer dynamischen list of values ist, dass der

Bereich von der Tabelle oder Ansicht abhängt. Der Endbenutzer kann den Bereich durch Einfügen, Aktualisieren und Löschen von Zeilen in der Tabelle ändern. Der Nachteil ist, dass eine Tabelle erstellt und gepflegt werden muss. Wenn sich die Domäne nicht ändert und die Anzahl der Zeilen in der Tabelle gering ist, kann eine statische lists of values verwendet werden. In diesem Fall kann nur der Entwickler den Bereich ändern (er ist "hard-coded").

Im zweiten Teil wird die Erstellung einer verpackten Anwendung, deren Export und Import in einen neuen Arbeitsbereich demonstriert.

Im dritten Teil des Kapitels wird eine Anleitung zur Erstellung mehrsprachiger Anwendungen gegeben. Als ergänzendes Studienmaterial werden drei exportierte Anwendungen sowie ein Videoclip bereitgestellt, der den Entwicklungsprozess im Detail zeigt.

## 12.2 Anwendung

### 12.2.1 Umfang der Anwendung

Ziel der Anwendung ist die Verwaltung von Stellen und Kompetenzen für die Personalabteilung. In einem Unternehmen kann es einige hundert Stellen geben. Eine Stelle kann eine bestimmte Teilmenge aller bekannten Kompetenzen umfassen. Das erforderliche Kompetenzniveau wird von der Personalabteilung für jede Stelle vorgeschrieben. Die Kompetenzniveaus beschreiben die Fähigkeit, zu helfen, zu nutzen, zu meistern, anzupassen und zu verbessern. Die Idee ist also, eine Anwendung zur Erstellung und Verwaltung von Stellenprofilen zu entwickeln. Die Anwendung verwendet drei Tabellen: JOB, COMPETENCE und JOB_COMPETENCE mit einer minimalen Anzahl von Datenfeldern (der Einfachheit und Prägnanz halber). Jede Tabelle hat einen Primärschlüssel und die Tabelle JOB_COMPETENCE hat zwei Fremdschlüssel, um jede Zeile mit den Tabellen JOB und COMPETENCE zu verknüpfen. Die Fremdschlüssel gewährleisten die Datenkonsistenz, d.h. eine Zeile in der Tabelle JOB_COMPETENCE kann nur existieren, wenn Zeilen mit denselben Primärschlüsseln in den Tabellen JOB und COMPETENCE vorhanden sind. Um eine gewisse Strenge zu erzwingen, wird den Tabellennamen das Präfix “CH12_” vorangestellt.

### 12.2.2 Tabellen erstellen

Wir können in APEX Tabellen erstellen:

- ohne Kenntnisse der Datenbeschreibungssprache durch Klicken im Objektbrowser. Diesen Prozess mit vielen Abbildungen zu illustrieren, würde jedoch zu viel Platz in diesem Buch einnehmen.
- mit Kenntnissen von DDL in SQL Workshop > SQL-Befehle durch Eingabe syntaktisch korrekter Befehle.
- mit Grundkenntnissen in Datenmodellierung (siehe Kapitel 2) und Quick SQL.

Um ein breites Verständnis und Prägnanz zu gewährleisten, wird Quick SQL verwendet (Quick SQL finden Sie unter SQL Workshop > Utilities > Quick SQL). Der folgende Text wird eingefügt:

```
CH12_JOB
  job_description vc100

CH12_COMPETENCE
  competence_description vc100

CH12_JOB_COMPETENCE
  id_job num /fk CH12_JOB
  id_competence num /fk CH12_COMPETENCE
  required_level vc10 /check 'assist','use','master', 'tailor','innovate'
```

Nach Eingabe des obigen Textes und Drücken der Schaltfläche **Generate SQL** erscheint der

folgende Bildschirm (siehe Abbildung 12.2):

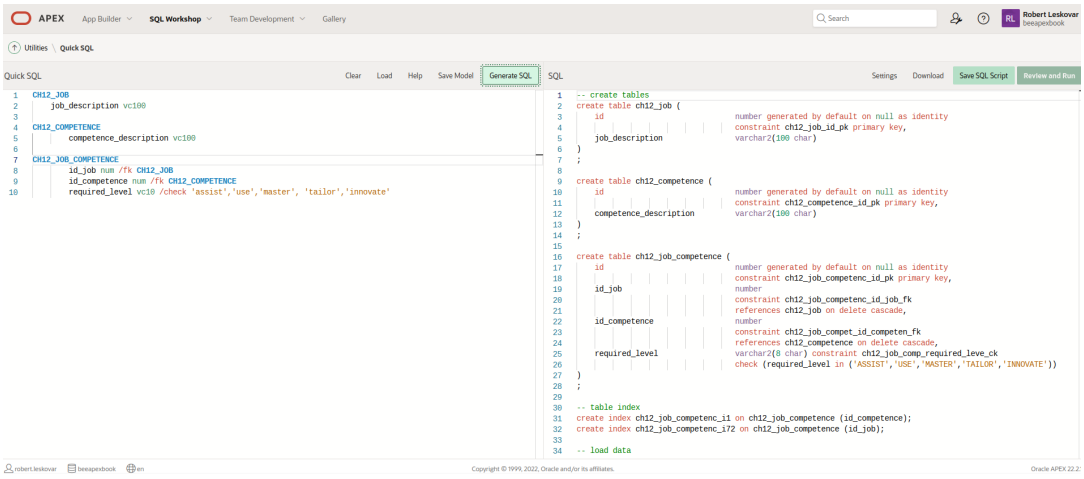



Abbildung 12.2: Umwandlung von Quick SQL in SQL-Befehle.

Klicken Sie dann auf die Schaltfläche **Save SQL Script** und benennen Sie das Skript als CH12CREATE. Jetzt können wir das Skript ausführen. Kehren Sie zu Scripts zurück und suchen Sie das Ausführen-Symbol in der Zeile, in der das Skript CH12CREATE angezeigt wird. Klicken Sie auf das **Start**-Symbol. Drei Tabellen und drei Indizes sollten erfolgreich erstellt worden sein. Sie können die Tabellen im Objektbrowser überprüfen.

### 12.2.3 Daten einfügen

Im APEX wählen Sie SQL Workshop > SQL Scripts > Create. Jetzt ist es an der Zeit, einige SQL INSERT-Befehle zu schreiben, um Daten in Tabellen zu füllen. Die Tabelle CH12_JOB wird sieben Zeilen enthalten, die jeweils einen bestimmten Job beschreiben. Die Tabelle CH12_COMPETENCE wird elf Zeilen enthalten, die elf digitale Kompetenzen beschreiben, die durch McKinsey's DELTAs definiert sind (unterschiedliche Elemente von Talenten). Und wir werden anweisen, fünf Kompetenzen in die Stelle 'Junior Apex Developer' einzufügen. Lassen Sie uns das Skript CH12INSERT nennen und das Skript schreiben (siehe Abbildung 12.3).

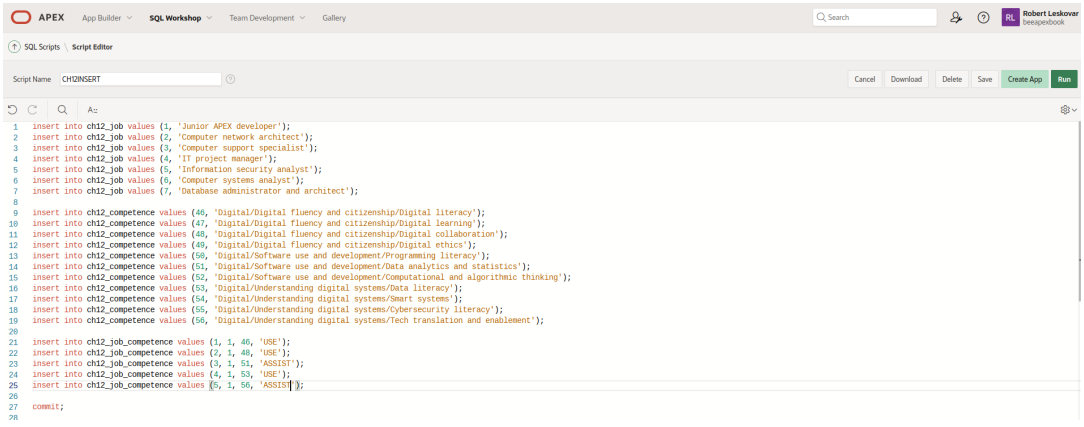



Abbildung 12.3: Skript CH12INSERT Daten in drei Tabellen einfügen und Transaktionen festlegen.

Klicken Sie auf die Schaltfläche **Create**. Führen Sie nun das Skript CH12INSERT aus. Kehren Sie zu Scripts zurück und klicken Sie auf das **Start**-Symbol in der Zeile, in der das Script CH12INSERT angezeigt wird. Klicken Sie auf die Schaltfläche **Run now**. Vierundzwanzig Zeilen

sollten erfolgreich erstellt worden sein. Sie können den Inhalt der Tabellen auch im Objektbrowser überprüfen.

### 12.2.4 Anwendung generieren

Nachdem Sie die Tabellen mit Testdaten gefüllt haben, navigieren Sie zu SQL Workshop > Scripts und bearbeiten das Script CH12CREATE. Klicken Sie auf die Schaltfläche **Create App** (siehe Abbildung 12.4). Sie können feststellen, dass drei Tabellen aufgelistet sind.

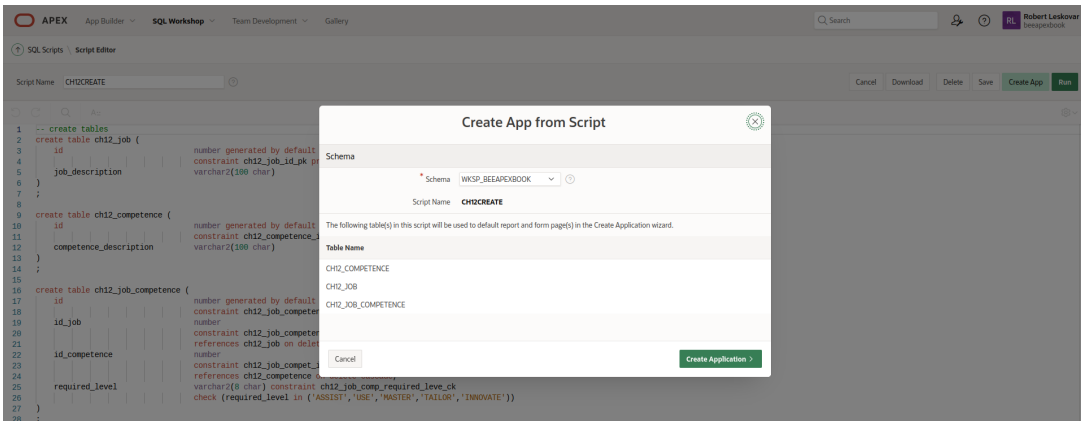



Abbildung 12.4: Anwendung aus Skript erstellen.

Klicken Sie auf die Schaltfläche Anwendung erstellen. Geben Sie den Namen der Anwendung ein (z.B. CH12 Application), klicken Sie auf Check all features und dann auf Create Application (siehe Abbildung 12.5).

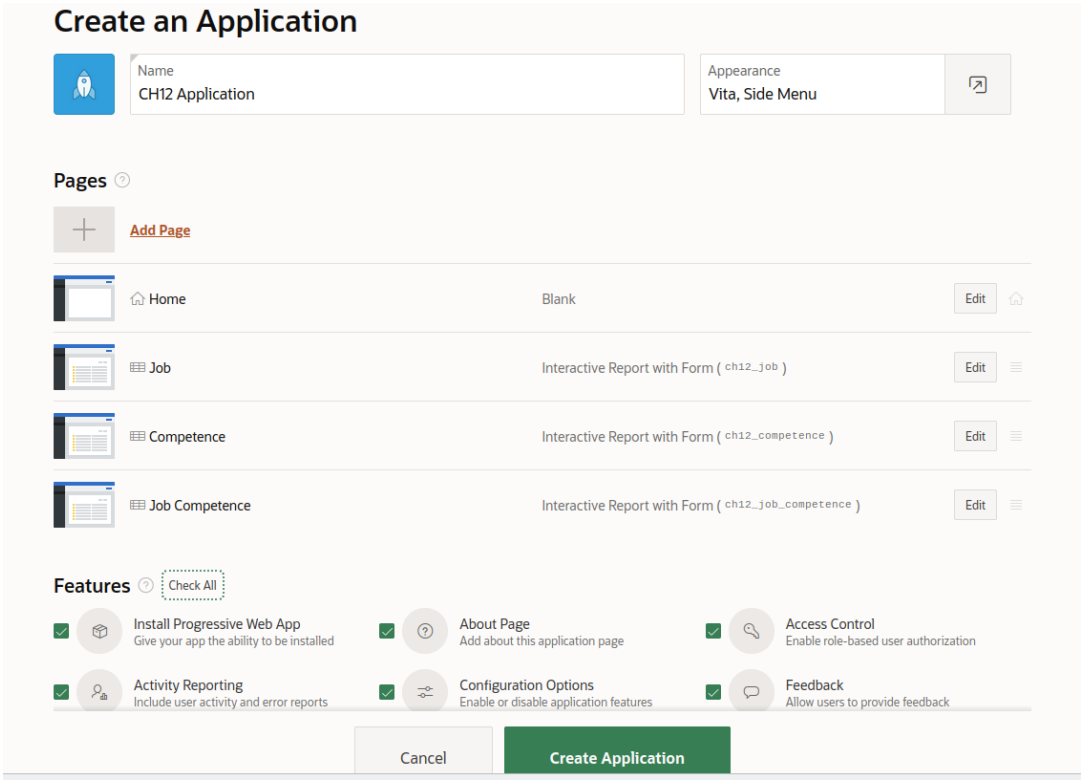



Abbildung 12.5: Auswahl des Namens der Anwendung und aller Funktionen.

Nachdem die Generierung abgeschlossen ist, können wir die Anwendung ausführen. In neueren Versionen von APEX erzeugt der Generator auch zwei Lists of Values (CH12_COMPETENCE.COMPETENCE_DESCR und CH12_JOB.JOB_DESCRIPTION), abgekürzt als LOV. Lists of Values sind Komponenten auf der Seite, bei denen ein Wert angezeigt und vom Benutzer verstanden wird, während die Seite selbst den entsprechenden Schlüssel oder Code zum Einfügen, Aktualisieren oder Löschen verwendet. Beispiel: Auf einem Formular wird der Nachname eines Mitarbeiters angezeigt und die zugehörige Identifikationsnummer des Mitarbeiters wird gespeichert, wenn die Bestätigungsschaltfläche gedrückt wird. Die generierten LOVs basieren auf Tabellen und es müssen keine SQL-Kenntnisse vorhanden sein. In der APEX-Terminologie werden sie als dynamisch bezeichnet, da eine Änderung in der Tabelle eine neue Liste von Werten erzeugt. Sie finden die generierten LOVs unter Shared Components > List of Values und können bestimmen, wie sie gesetzt werden. In diesem Kapitel lernen wir, wie man dynamische (basierend auf einer einfachen SQL-Abfrage) und statische LOVs (basierend auf einem festen Satz von Werten) erstellt. Durch den Einsatz von LOVs in Anwendungsseiten wird die Anwendung fehlerresistenter und auch benutzerfreundlicher. Das bedeutet, dass der Benutzer nicht jeden beliebigen Wert in ein Formularfeld eingeben kann, wodurch Tippfehler vermieden werden. Navigieren Sie zu Shared Components (siehe Abbildung 12.6).

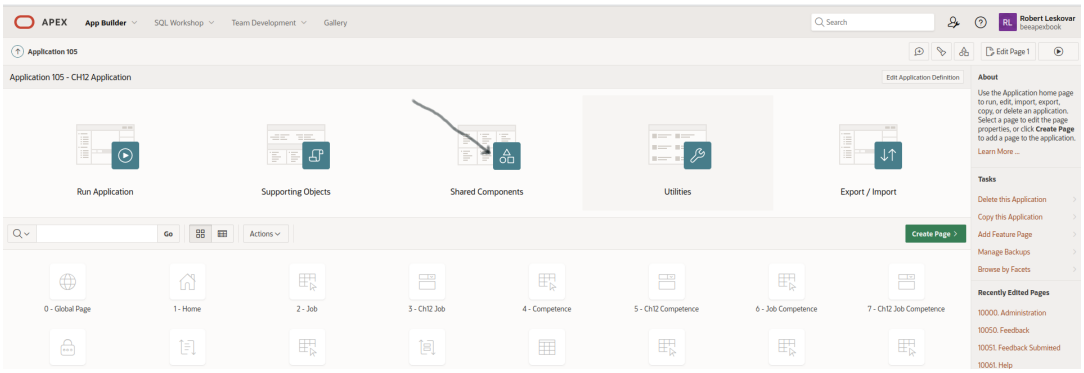



Abbildung 12.6: Shared Components auswählen.

Wählen Sie in den Shared Components List of Values (in Other Components) und klicken Sie auf Create. Wir werden die List of Values von Grund auf neu erstellen (siehe Abbildung 12.7).

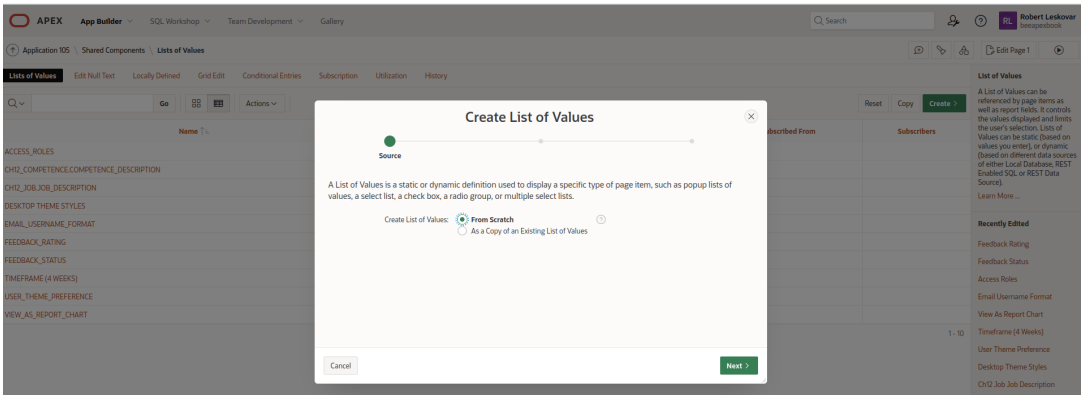



Abbildung 12.7: List of Values von Grund auf neu erstellen.

Benennen Sie LOV als CH12_LOV_COMPETENCE_DESCRIPTION und wählen Sie den dynamischen Typ (siehe Abbildung 12.8)

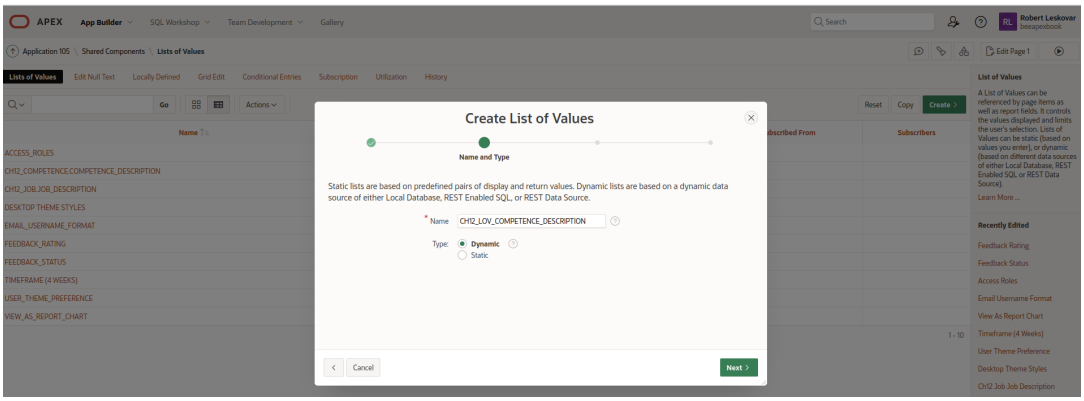



Abbildung 12.8: Name und Typ festlegen: CH12_LOV_COMPETENCE_DESCRIPTION ist dynamisch.

Wählen Sie SQL Query als Quelltyp und geben Sie die Abfrage “select competence_description as d, id as r from ch12_competence” als SQL select-Anweisung ein (siehe Abbildung 12.9).

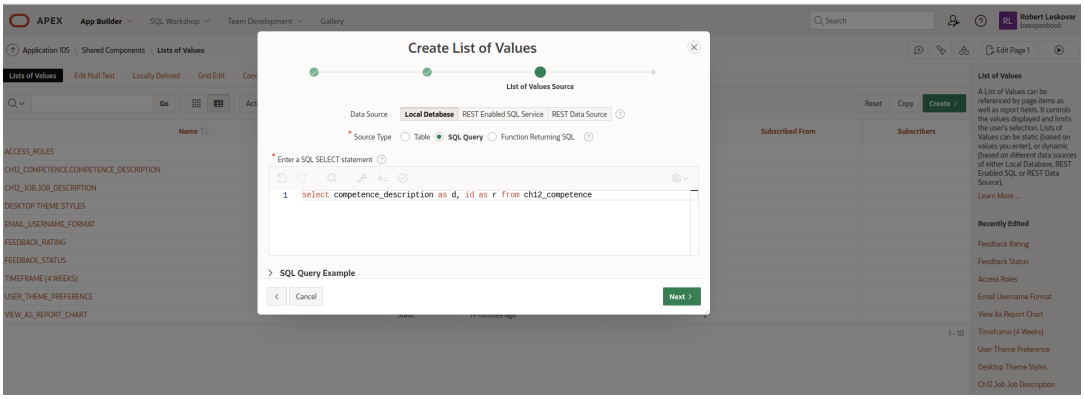



Abbildung 12.9: Eingabe des SQL SELECT-Befehls.

In dieser List of Values wird die Beschreibung der Kompetenz (Anzeigespalte) auf der Formularseite angezeigt, während die ID (Rückgabespalte) zurückgegeben wird. Die nächste LOV wird statisch sein und den Namen CH12_LOV_COMPETENCE_LEVEL tragen und wie in Abbildung 12.10 und Abbildung 12.11 dargestellt erstellt werden.

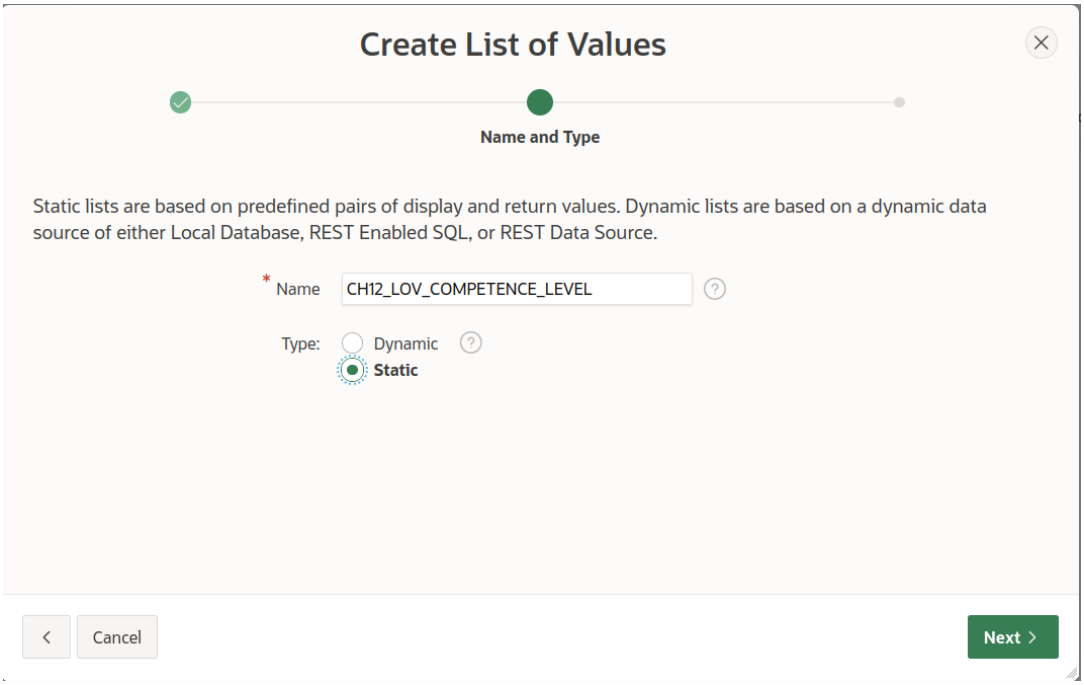



Abbildung 12.10: Name und Typ definieren: CH12_LOV_COMPETENCE_LEVEL ist statisch.

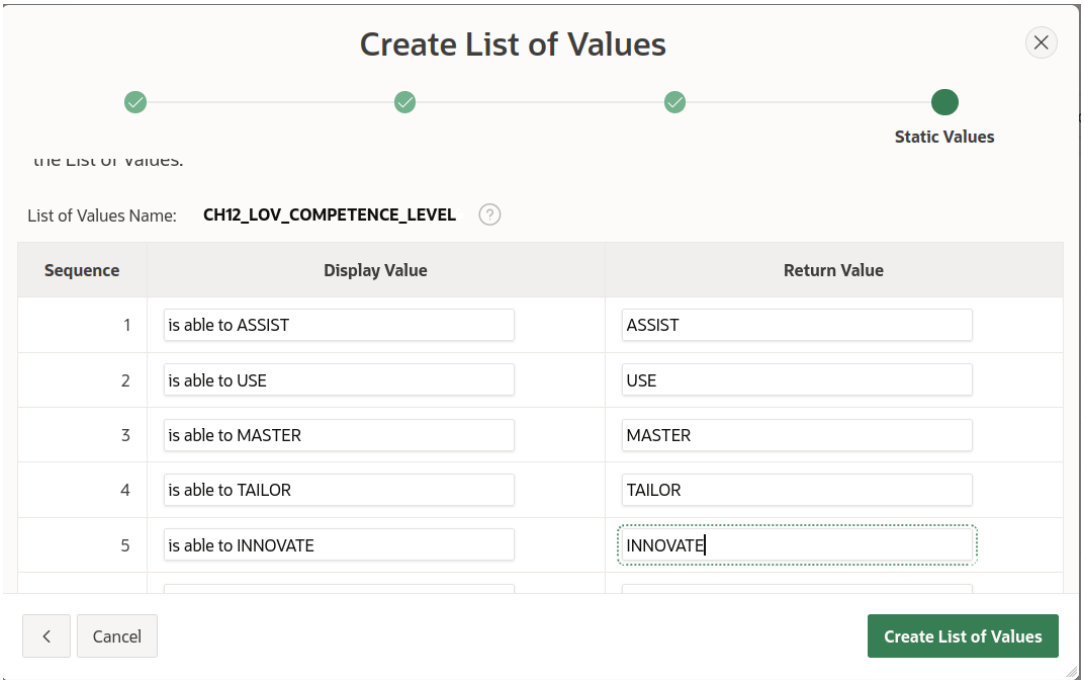



Abbildung 12.11: Anzeige- und Rückgabewerte definieren: CH12_LOV_COMPETENCE_LEVEL ist statisch.

Navigieren Sie zu den Anwendungsseiten (siehe Abbildung 12.12). Wählen Sie Seite 7 (Ch12 Job Competences).

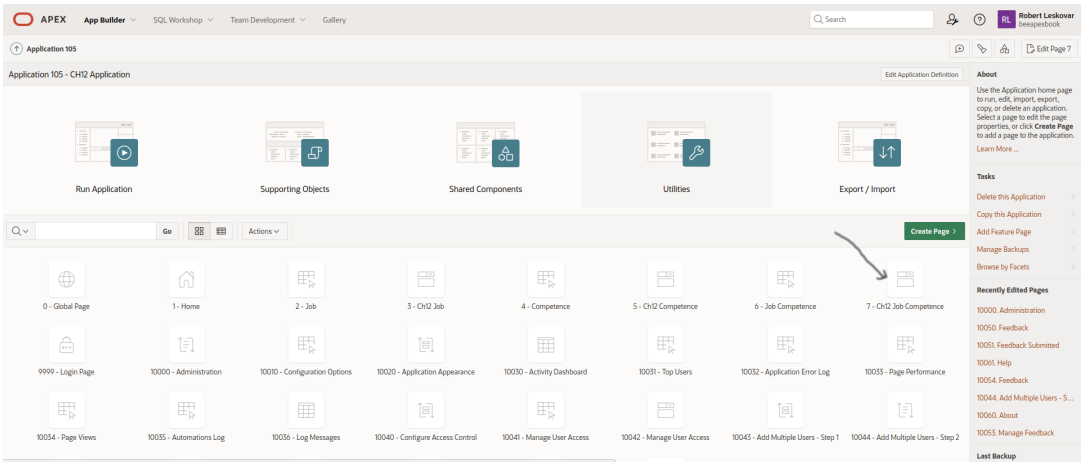



Abbildung 12.12: Wählen Sie Seite 7 (Ch12 Job Competences).

Ändern Sie den Seiteneintrag P7_ID_COMPETENCE:

- Typ ist SSelect listïn Identifikation
- Typ ist SShared componentsïn List of Values
- List of values ist CH12_LOV_COMPETENCE_DESCRIPTION List of Values

Ändern Sie den Seiteneintrag P7_REQUIRED_LEVEL:

- Typ ist SSelect listïn Identifikation
- Typ ist SShared componentsïn List of Values
- List of Values ist CH12_LOV_COMPETENCE_LEVEL List of Values

Speichern Sie die Seite 7, navigieren Sie zu Seite 6 und starten Sie die Anwendung. Sie sollten Abbildung 12.13 sehen.

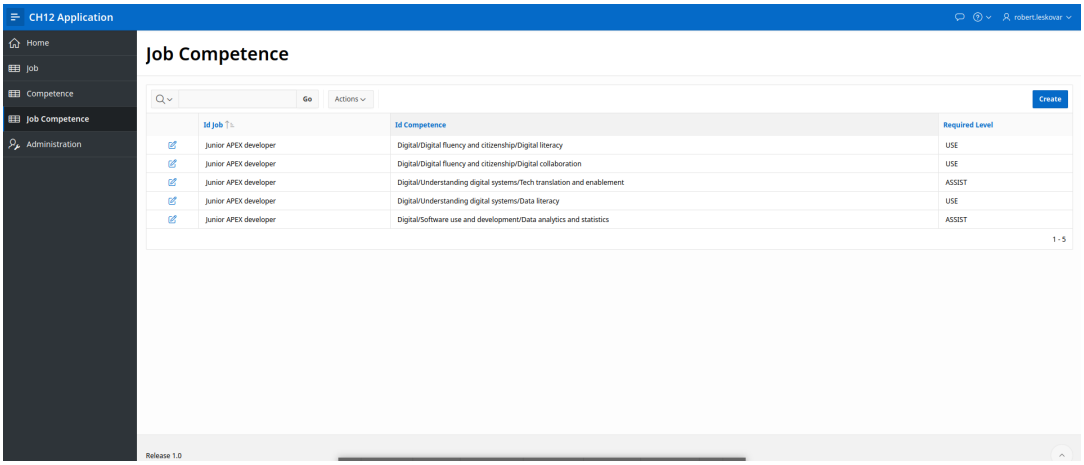



Abbildung 12.13: Seite 7 auswählen (Ch12 Job Competences).

Bearbeiten Sie nun den ersten Eintrag. Beide Auswahllisten liefern die erwarteten Daten.

## 12.3 Verpackte Anwendung

Da wir bereits einige Skripte (CH12CREATE, CH12INSERT) erstellt haben, werden wir die verpackte Anwendung sehr schnell erstellen. Navigieren Sie zur Anwendung und wählen Sie Supporting Objects (siehe Abbildung 12.14).

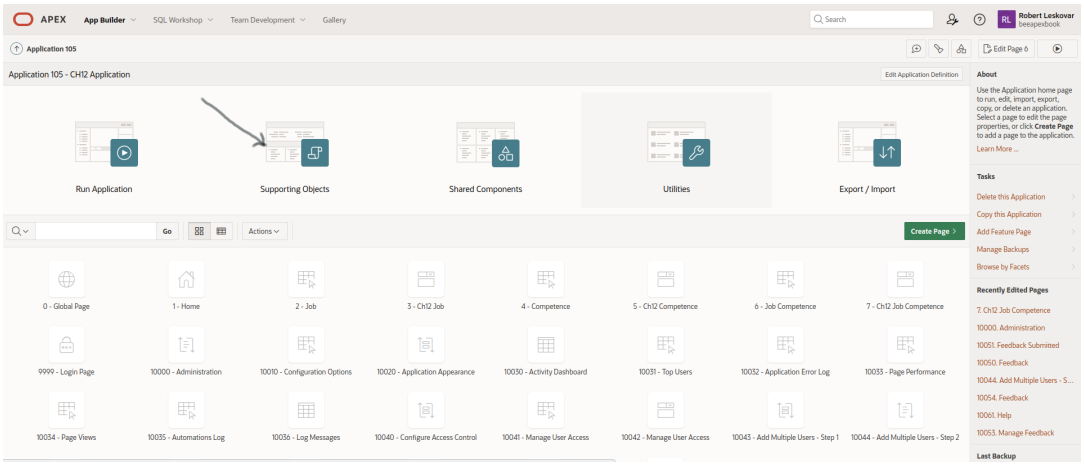



Abbildung 12.14: Supporting Objects auswählen.

Jetzt können wir die Installation (Voraussetzungen, Zeichenketten zur Ersetzung von Anwendungen, Build-Optionen, Validierungen vor der Installation, Installationsskripte und -meldungen), das Upgrade (Upgrade-Skripte, Upgrade-Meldung) und die Deinstallation (Deinstallationsskript, Deinstallationsmeldung) einstellen. Siehe Abbildung 12.15.

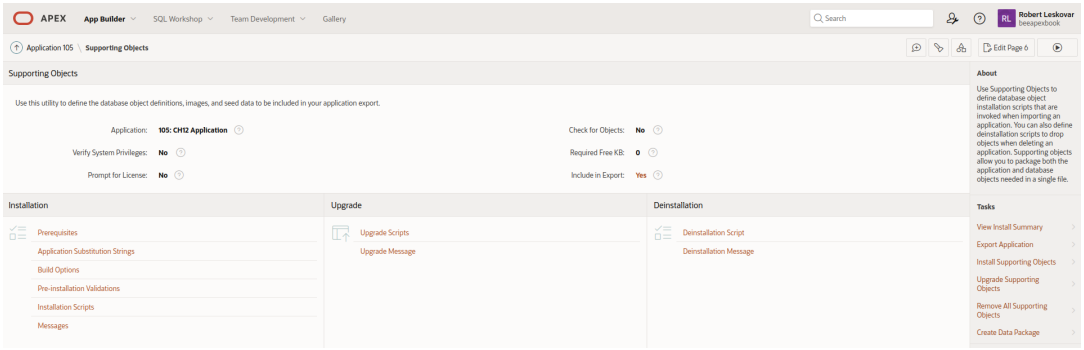



Abbildung 12.15: Voraussetzungen festlegen.

Um den Benutzer vor dem versehentlichen Löschen vorhandener Daten zu schützen, wird das Vorhandensein von drei Tabellen geprüft (siehe Abbildung 12.16). Klicken Sie auf die Schaltfläche Apply changes.

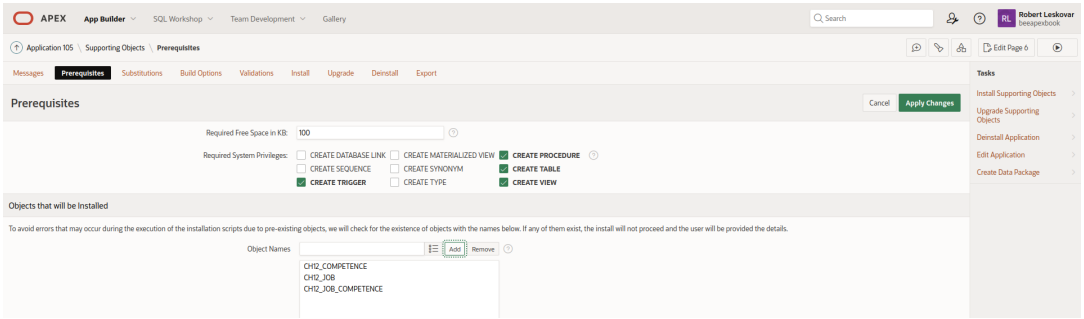



Abbildung 12.16: Prüfung auf Existenz von drei Tabellen einstellen.

In Application Substitution String wird gefragt, ob der Name der Anwendung “BeeAPEX Chapter 12 App” sein soll (siehe Abbildung 12.17).

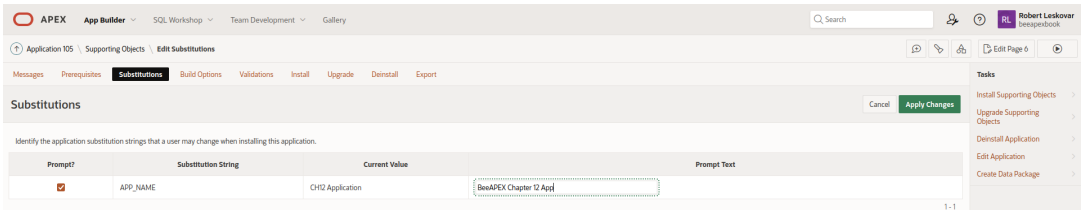



Abbildung 12.17: Aufforderung zum Umbenennen der Anwendung einstellen.

Klicken Sie auf die Schaltfläche Änderungen übernehmen.

Für diese verpackte Anwendung werden wir auch die folgende Einstellung in Supporting Objects anwenden:

- Voraussetzungen (siehe Abbildung 12.16)
- Anwendung Substitutionssubstrings (siehe Abbildung 12.17)
- Installationsskripte: Da wir bereits die Skripte CH12CREATE und CH12INSERT vorbereitet haben, werden wir deren Inhalt verwenden (siehe Abbildung 12.18)
- Deinstallationsskripte (siehe Abbildung 12.19)
- Meldungen setzen: Begrüßungsmeldung auf „Kapitel 12: ...“, Lizenzmeldung auf „CC BY“, Erfolgsmeldung auf „Erfolg!“, Fehlermeldung auf „Fehlschlag“, Deinstallationsmeldung auf „Anwendung CH12 deinstalliert“

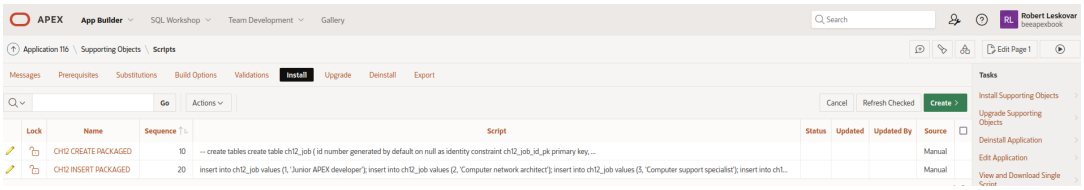



Abbildung 12.18: Installationsskripte festlegen.

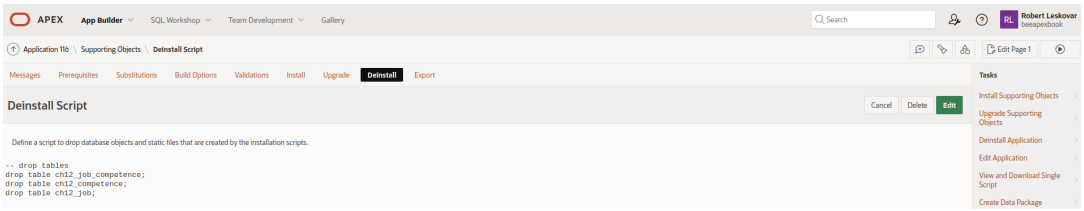



Abbildung 12.19: Deinstallationsskripte festlegen.

Exportieren Sie nun die verpackte Anwendung mit dem Assistenten (siehe Abbildung 12.1). Sie können alle von APEX vorgeschlagenen Exporteinstellungen übernehmen. Speichern Sie die exportierte Anwendung auf Ihrem Computer und merken Sie sich den Dateinamen. Melden Sie sich im neuen Arbeitsbereich an. Klicken Sie im neuen Arbeitsbereich auf den Import-Assistenten und ziehen Sie den Dateinamen in den Drag & Drop-Bereich (siehe Abbildung 12.20).

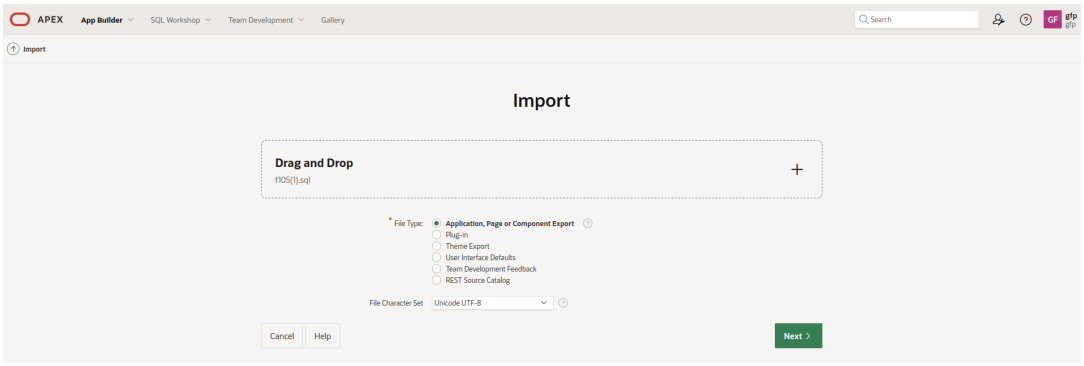



Abbildung 12.20: Anwendung in anderen Arbeitsbereich importieren.

Klicken Sie auf die Schaltfläche Next (auch zur Bestätigung des Datei-Imports) und auf die Schaltfläche Install application. Bestätigen Sie die Installation der unterstützenden Objekte mit der Schaltfläche Next, akzeptieren Sie die Lizenz (siehe Abbildung 12.21), benennen Sie die Anwendung in Ïmported CH12 Applicationüm (siehe Abbildung 12.22) und klicken Sie auf die Schaltfläche Install. Führen Sie die Anwendung noch nicht aus, da Sie die Berechtigungen für den neuen Benutzer separat festlegen müssen. Benutzer und Benutzerberechtigungen werden aus Sicherheitsgründen nicht exportiert. Nur eine Anmerkung: um Benutzer und Berechtigungen zu migrieren, könnten wir tief in die APEX API eintauchen und Installations- und Deinstallationsskripte schreiben. Navigieren Sie zu Shared Components in der importierten Anwendung. Wählen Sie application access control der Anwendung und geben Sie dem aktuell angemeldeten Benutzer die entsprechenden Rechte. In unserem Fall fügen wir dem APEX-Benutzer die Rollen GFP-Administrator, Contributor und Reader hinzu (siehe Abbildung 12.23). Bestätigen Sie mit einem Klick auf die Schaltfläche Add Assignment. Führen Sie nun die Anwendung als Benutzer mit den angegebenen Rechten aus.

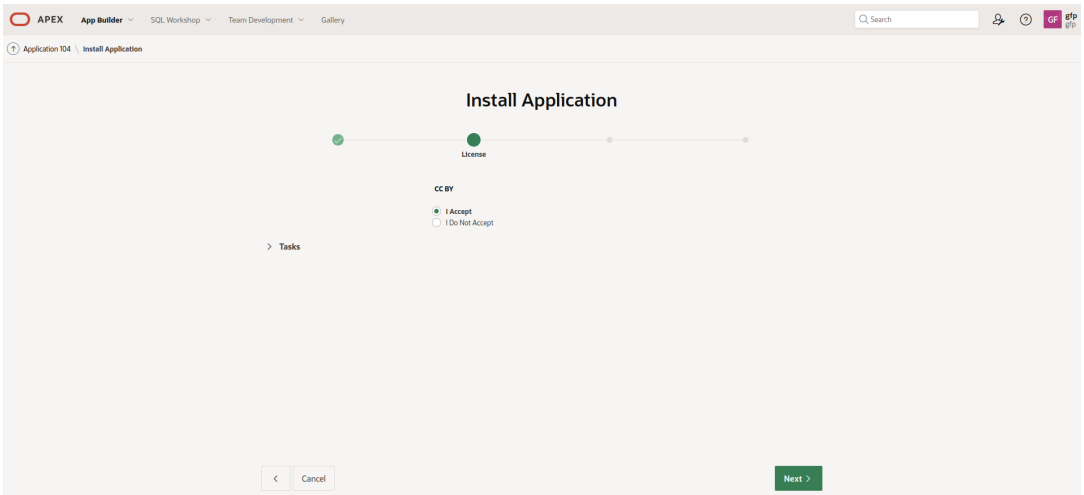



Abbildung 12.21: Lizenzvereinbarung.

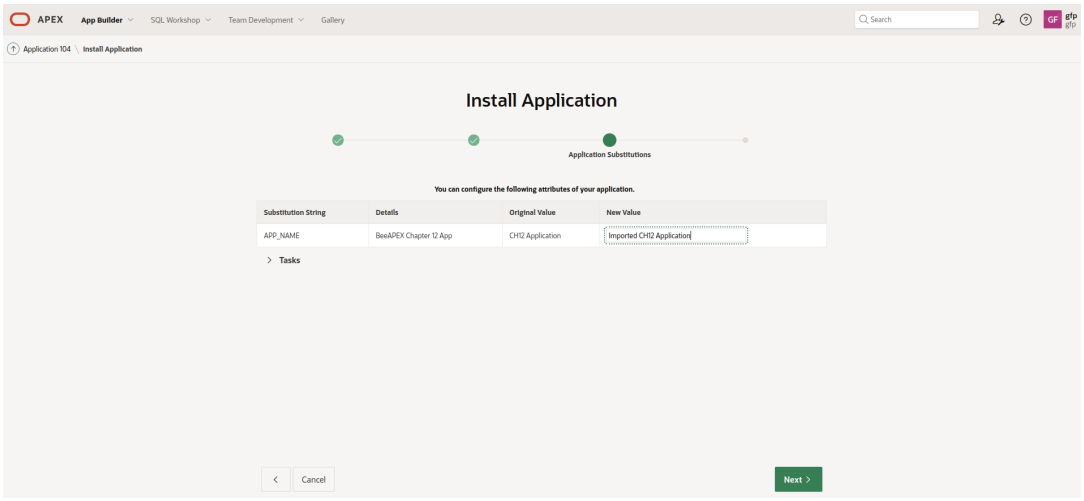



Abbildung 12.22: Importierte Anwendung umbenennen.

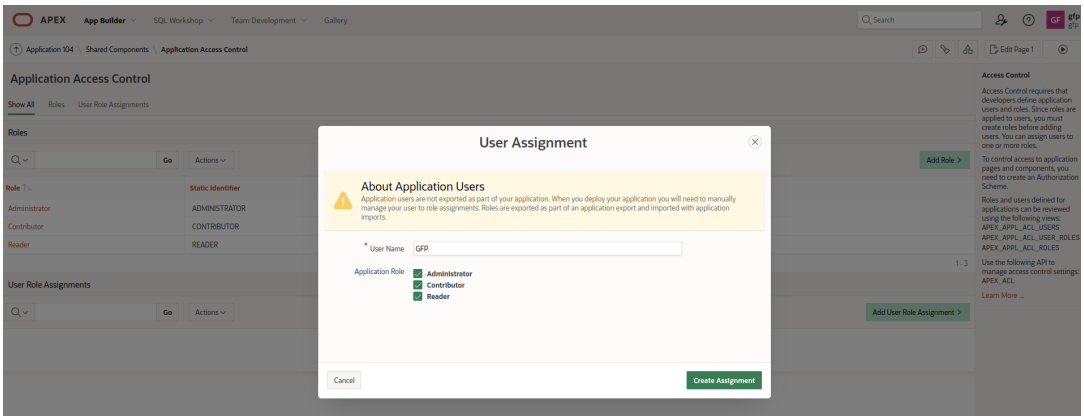



Abbildung 12.23: Hinzufügen einer Rolle zu einem Benutzer.

## 12.4 Mehrsprachige Anwendung

APEX hat viele Möglichkeiten der Übersetzung (basierend auf der primären Sprache der Anwendung, dem Browser, den Einstellungen der Anwendung, dem Element und der Sitzung). Der folgende Teil enthält die einfachsten Anweisungen, um mehrere Sprachen in die Anwendung zu implementieren, indem die primäre Sprache der Anwendung geändert wird. Die Schritte sind:

- Navigieren Sie zu Shared components > Globalization > Application Translations > Define application languages: Klicken Sie auf die Schaltfläche Create; legen Sie für jede Sprache eine eindeutige ganze Zahl und einen Wert fest (d.h. fügen Sie der Anwendungsnummer zwei Ziffern hinzu, z.B. 10801 für sl; 10802 für hr; 10803 für de-at; 10804 für el; 10805 für sk; 10806 für pl). Abbildung 12.24 zeigt alle eingestellten Sprachen.
- Navigieren Sie zu Shared components > Globalization > Application translations > Seed translatable text. Wählen Sie alle Sprachen aus, wie in Abbildung 12.25 gezeigt. Klicken Sie auf die Schaltfläche Seed und warten Sie, bis das Seeding abgeschlossen ist.
- Navigieren Sie zu Shared components > Globalization > Application Translations > Download XLIFF translation files; Sie können wählen, ob Sie alle übersetzbaren Elemente oder nur die zu übersetzenden Elemente herunterladen möchten. Das Beispiel in Abbildung 12.26 zeigt den Export von Seite 2, slowenischer Sprache und zu übersetzende Elemente.
- Bearbeiten Sie die exportierte Datei in Ihrem bevorzugten Editor (Notepad++, Kate, Sublime

usw.). Zu Demonstrationszwecken haben wir nur ein paar SZielmarkierte Zeichenfolgen geändert: in den Zeilen 48, 52, 56, 60, 72 und 76 (siehe Abbildung 12.27). Speichern Sie die Änderungen.

- Navigieren Sie zu Shared components > Globalization > Translate application > Apply XLDIFF translation files > Upload Files. Wählen Sie die Datei (siehe Abbildung 12.28) und klicken Sie auf Upload. Wählen Sie die richtige Sprache unter Äpply to Translation"(siehe Abbildung 12.29). Klicken Sie auf Apply checked. Klicken Sie auf Publish.
- Navigieren Sie zu Shared components > Globalization > Translate application > Publish translations. Wählen Sie alle Sprachen aus, die Sie übersetzt haben, und klicken Sie auf Publish (siehe Abbildung 12.30). Warten Sie, bis Sie die Meldung erhalten, dass die Anwendung erfolgreich veröffentlicht wurde.
- Navigieren Sie zu Shared components > Globalization > Globalization attributes. Ändern Sie die Primärsprache der Anwendung in die neue Übersetzung (siehe Abbildung 12.31). Klicken Sie auf Apply changes.

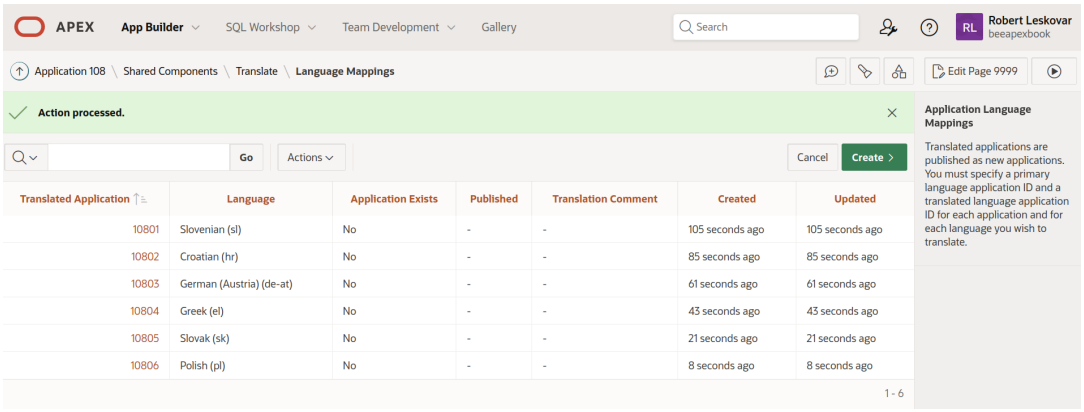



Abbildung 12.24: Definierte Sprachen für die Übersetzung.

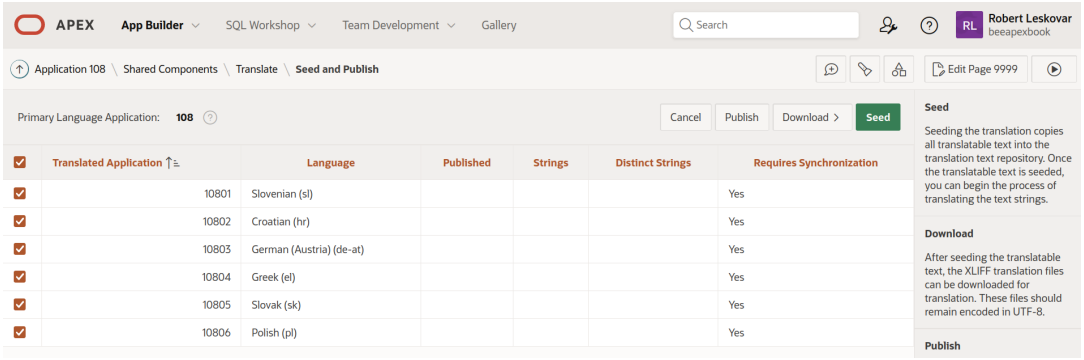



Abbildung 12.25: Seed translatable text.

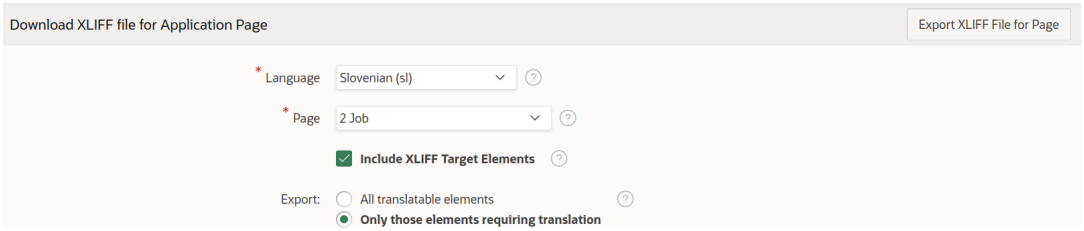



Abbildung 12.26: Zeichenfolgen für eine bestimmte Sprache und Seite exportieren.

```
45  </trans-unit>
46  <trans-unit id="S-13-23279527479873043-108">
47  <source>Create</source>
48  <target>Ustvari</target>
49  </trans-unit>
50  <trans-unit id="S-20-23278046036873035-108">
51  <source>Ch12 Job</source>
52  <target>Ch12 Del. mesto</target>
53  </trans-unit>
54  <trans-unit id="S-20-23280741017873048-108">
55  <source>Breadcrumb</source>
56  <target>Drobtina</target>
57  </trans-unit>
58  <trans-unit id="S-143-23278046036873035-108">
59  <source>Job</source>
60  <target>Del. mesto</target>
61  </trans-unit>
62  <trans-unit id="S-146-23278230062873035-108">
63  <source>The maximum row count for this report is #MAX_ROW_COUNT# rows.  Please apply a filter to reduce the num
64  <target>The maximum row count for this report is #MAX_ROW_COUNT# rows.  Please apply a filter to reduce the num
65  </trans-unit>
66  <trans-unit id="S-147-23278230062873035-108">
67  <source>No data found.</source>
68  <target>No data found.</target>
69  </trans-unit>
70  <trans-unit id="S-149-23278547600873039-108">
71  <source>ID</source>
72  <target>Šifra</target>
73  </trans-unit>
74  <trans-unit id="S-149-23278989319873042-108">
75  <source>Job Description</source>
76  <target>Opis delovnega mesta</target>
77  </trans-unit>
78  </body>
79  </file>
80  </xliff>
```

Abbildung 12.27: Übersetzung der mit targetmarkierten Zeichenfolgen in den Zeilen 48, 52, 56, 60, 72 und 76.

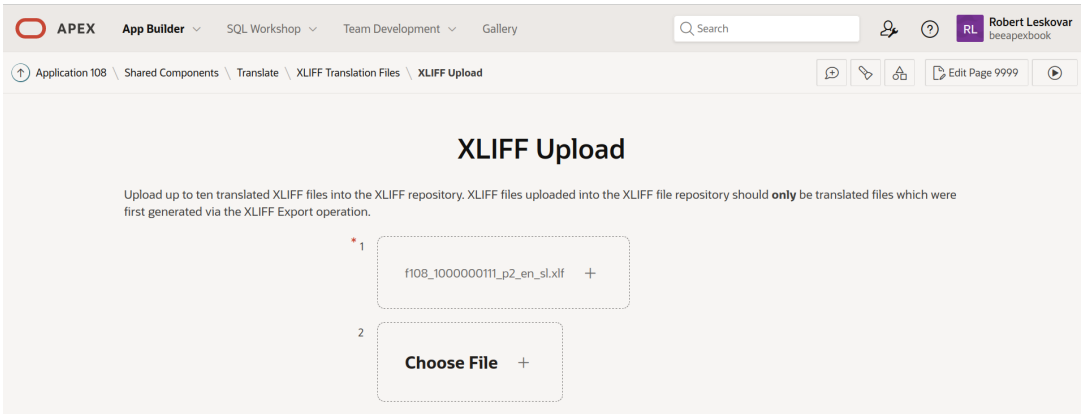



Abbildung 12.28: Hochladen von XLDIFF-Übersetzungsdateien.

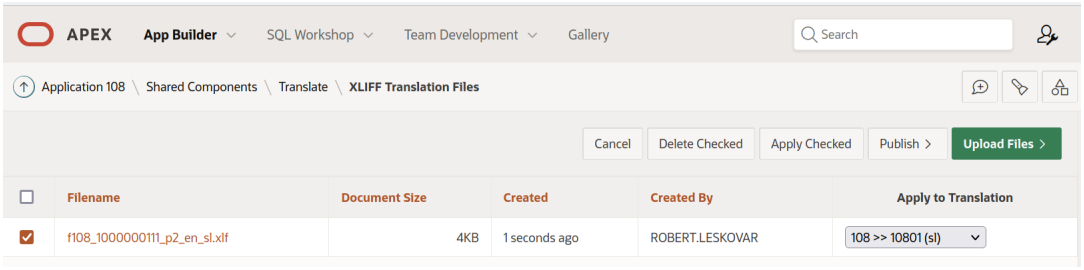



Abbildung 12.29: Änderungen übernehmen und veröffentlichen.

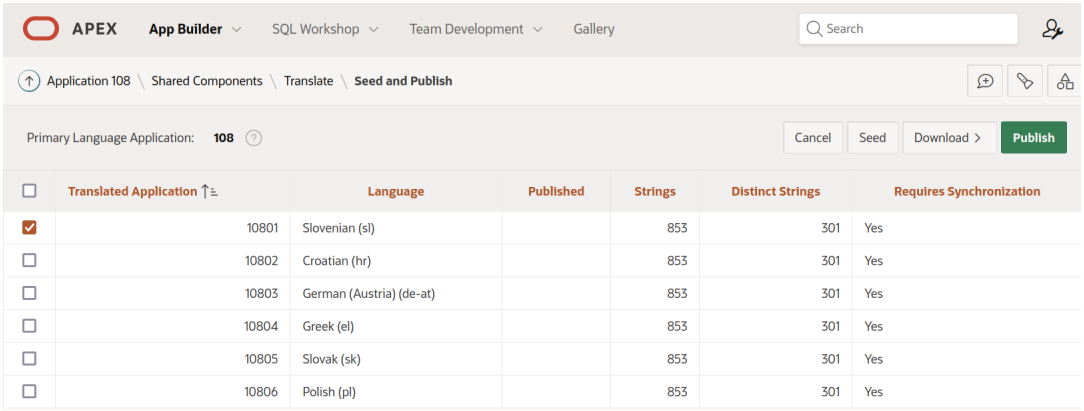



Abbildung 12.30: Endgültige Veröffentlichung der Übersetzung der Anwendung.

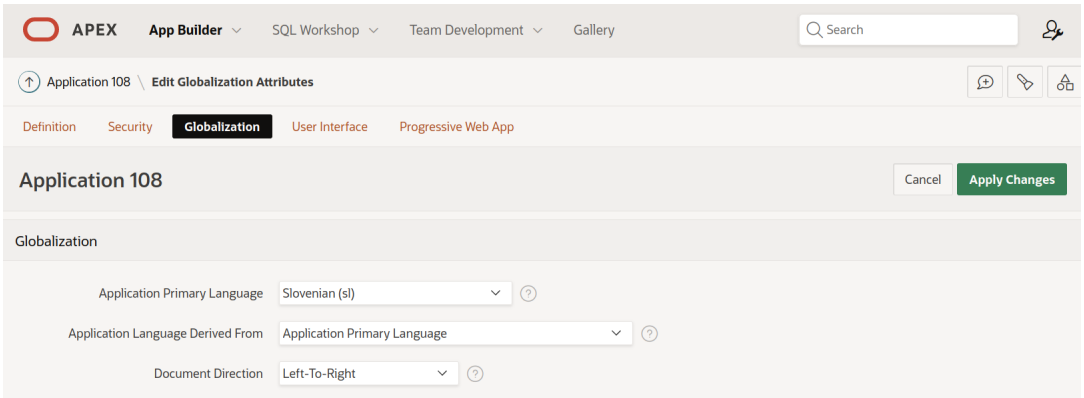



Abbildung 12.31: Einstellung der Primärsprache der Anwendung.

Überprüfen Sie nun Seite 2 in der übersetzten Anwendung (siehe Abbildung 12.32).

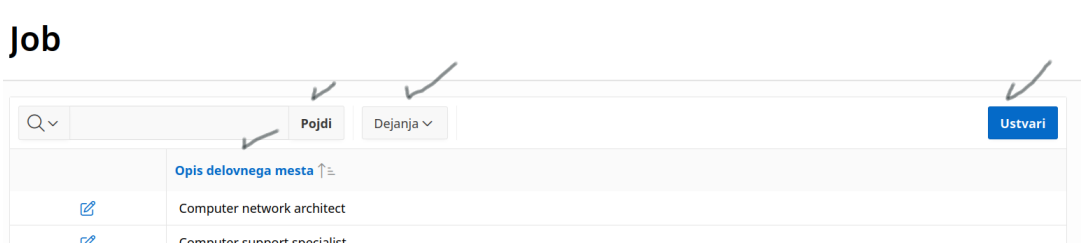



Abbildung 12.32: Übersetzte Seite.

**Abschließende Bemerkungen** Die Entwicklung findet normalerweise in einer Testumgebung statt. Bei der Migration einer genehmigten Anwendung in einen neuen Arbeitsbereich sind verpackte Anwendungen sehr praktisch, da sie den Zeitaufwand für die Migration verringern. Aus Sicherheitsgründen werden eventuelle Benutzer und ihre Rollen nicht migriert. Die Autorisierung von Benutzern im neuen Arbeitsbereich ist eine wichtige Aufgabe. Dieses Kapitel enthält auch grundlegende Anweisungen für mehrsprachige Anwendungen. Es ist zu beachten, dass bei großen und kritischen Anwendungen der Übersetzungsprozess durch leistungsfähige automatische Übersetzungswerkzeuge sowie durch Sprach- und Kulturspezialisten unterstützt werden muss.

## 12.5 Ergänzendes Lernmaterial

Hier finden Sie ergänzendes Lernmaterial:

- exportierte Anwendungen
- Video-Guides

Alle zusätzlichen Lernmaterialien sind auf der öffentlichen BeeAPEX-Projektseite verfügbar. Melden Sie sich als Gastbenutzer an (kein Passwort erforderlich). Das Lehrbuch finden Sie im Abschnitt Books, die Skripte im Ordner Part 1 > Chapter12 im Abschnitt Scripts und die Videoanleitungen in der Sammlung der Videoanleitungen. Material für Kurzkurse finden Sie in der Sektion Short courses.

### 12.5.1 Exportierte Anwendungen

Es gibt drei verwandte Anwendungen:

- Erstmalige Anwendung. Prüfen Sie, ob Tabellen, die mit CH12 beginnen, existieren. Wenn Tabellen vorhanden sind, können Sie sie mit dem Skript CH12DROP löschen. Importieren Sie dann die Anwendung in die Datei CH12_Application_initial.sql. Nachdem Sie die Anwendung importiert haben, müssen Sie Tabellen definieren. Führen Sie das Skript CH12CREATE aus und füllen Sie die Tabellen optional mit dem Skript CH12INSERT auf. Navigieren Sie zu imported application > Shared Components > Application Access Control. Fügen Sie die Benutzerrollenzuweisung für den aktuell angemeldeten Benutzer hinzu. Erlauben Sie alle Aktionen mit der Rolle Administrator, Auswählen, Einfügen, Aktualisieren und Löschen mit der Rolle Contributor und nur Auswählen mit der Rolle Reader.
- Verpackte Anwendung. Prüfen Sie, ob Tabellen, die mit CH12 beginnen, existieren. Wenn Tabellen vorhanden sind, müssen Sie sie mit dem Skript CH12DROP löschen. Importieren Sie dann die Anwendung in die Datei CH12_Application_packaged.sql. Navigieren Sie zu imported application > Shared Components > Application Access Control. Fügen Sie die Benutzerrollenzuweisung für den aktuell angemeldeten Benutzer hinzu. Erlauben Sie alle Aktionen mit der Rolle Administrator, Auswählen, Einfügen, Aktualisieren und Löschen mit der Rolle Contributor und nur Auswählen mit der Rolle Reader.
- Verpackte mehrsprachige Anwendung. Prüfen Sie, ob Tabellen, die mit CH12 beginnen, existieren. Wenn Tabellen vorhanden sind, müssen Sie sie mit dem Skript CH12DROP löschen. Importieren Sie dann die Anwendung in die Datei CH12_Application_packaged.sql. Navigieren Sie zu imported application > Shared Components > Application Access Control. Fügen Sie die Benutzerrollenzuweisung für den aktuell angemeldeten Benutzer hinzu. Erlauben Sie alle Aktionen mit der Rolle Administrator, Auswählen, Einfügen, Aktualisieren und Löschen mit der Rolle Contributor und nur Auswählen mit der Rolle Reader.

### 12.5.2 Video-Anleitungen

Die folgenden Videoanleitungen werden zur Verfügung gestellt:

- create initial application
- import initial application

- copy initial application to packaged and creating packaged application
- import packaged application
- import packaged multilingual application

## 12.6 Fragen

1. Was ist eine verpackte Anwendung und was ist ihr Nutzen?
2. Wie exportiert man eine verpackte Anwendung, um die Definition von Datenbankobjekten und Daten für Tabellen einzuschließen?
3. Welche Rolle spielt die XLDIFF-Datei bei der Erstellung mehrsprachiger Anwendungen?

## 12.7 Antworten

1. Eine verpackte Anwendung ist eine vorgefertigte Anwendung, die in einem Oracle APEX-Arbeitsbereich installiert und konfiguriert werden kann. Der Hauptvorteil einer verpackten Anwendung ist die einfache und reibungslose Migration sowie die sofortige Verfügbarkeit der vorbereiteten Daten.
2. Die verpackte Anwendung soll mit dem Assistenten exportiert werden. Man kann Skripte einfügen, die Datenbankobjekte (wie Tabellen) definieren und das Einfügen von Daten in Tabellen anweisen. Die Liste der Skripte und die Reihenfolge der Ausführung dieser Skripte wird im Assistenten festgelegt. Eine verpackte Anwendung kann auch De-Installationsskripte enthalten.
3. XLDIFF-Dateien sind mit Tags versehen und bieten daher einen einfachen Zugang zu den Zielwerten für Übersetzungsstrings. Diese Datei kann die gesamte Anwendung oder nur eine bestimmte Seite in der ausgewählten Sprache enthalten. Die Datei mit den übersetzten Zielzeichenketten wird in die Anwendung importiert. Das Übersetzungs-Repository muss veröffentlicht werden, um für den Endbenutzer verfügbar zu sein.

# II Erstellen einer Anwendung in APEX


13 Intranet-Nachrichten für Mitarbeiter . 229
R. Leskovar, U. Rajkovič, A.Baggia; Übersetzer W. Schwinger . . . . . .

14 GreenDi - Pflanzenkatalog . . . . . 241
Vjeran Strahonja, Dijana Oreški, Darko Andročec, Ana Kutnjak . . . . . .

15 GreenDi - Benutzerautorisierung und -verwaltung . . . . . 251
Vjeran Strahonja, Darko Andročec, Ana Kutnjak, Larisa Hrustek . . . . . .

16 Kleines Innovationssystem . . . . . 259
R. Leskovar, U. Rajkovič, A.Baggia; Übersetzer W. Schwinger . . . . . .

17 Management von Geschäftsprozessen 271
R. Leskovar, U. Rajkovič, A.Baggia; Übersetzer W. Schwinger . . . . . .

18 GreenDi - Tausch von Pflanzen und Saatgut . . . . . 303
Vjeran Strahonja, Dijana Oreski, Darko Androcec, Ana Kutnjak . . . . . .

19 Verwaltungssystem für Buchrezensionen 309
A. Kutnjak, L. Hrustek, A. Baggia and R.Leskovar . . .

20 Stückliste und Kostenberechnung . . . 323
R. Leskovar, U. Rajkovič, A.Baggia . . . . . .

21 Ernährungs- und Diätmanagement . . 339
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . .

22 Sprechstundenplanung . . . . . 359
J. Mańko, M. Sońta, R. Leskovar . . . . . .

23 Telco-Fall . . . . . 379
Veronika Šalgová, Jozef Kostolný, Michal Mrena, Michal Kvet, Miroslav Potočár . . . . . .

24 Fall Autovermietung . . . . . 405
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . .



# 13. Intranet-Nachrichten für Mitarbeiter

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA; ÜBERSETZER W. SCHWINGER

## 13.1 Wirtschaftliche Sicht des Falles

Das mittelgroße Unternehmen mit siebenhundert Mitarbeitern ist ein etablierter Akteur in der globalen Industrie für Spitzensport- und Freizeitausrüstung und fortschrittliche Verbundwerkstoffe. Der Hauptsitz befindet sich in Slowenien, Tochtergesellschaften gibt es in den USA, Kanada, Deutschland und Japan. Der Jahresumsatz liegt bei rund 90 Millionen Euro. Der Großteil der Mitarbeiter ist in Slowenien an einem Standort beschäftigt. Das Unternehmen hat vier Geschäftsbereiche, die jeweils mehrere Abteilungen haben. In der Vergangenheit wurde ein internes halbjährliches Bulletin verwendet, um verschiedene Erfolge, Erfolgsgeschichten, Ankündigungen neuer Produkte und ähnliches zu kommunizieren. Dann wurde eine statische (reine HTML) Intranetseite erstellt, um wichtige Neuigkeiten zu verbreiten. Doch diese Möglichkeit, die Mitarbeiter zu stärken und ihr Engagement für das Unternehmen zu erhöhen, war nur von kurzer Dauer. Der idyllische Garten hat sich in ein Problem verwandelt.

## 13.2 Problemstellung

Der Prozess der Veröffentlichung war nur lose definiert. IT-Mitarbeiter erhielten von den Herausgebern - Abteilungs- und Bereichsleitern - die Aufforderung, Nachrichten zu veröffentlichen. Die Anfrage wurde per Post, Telefon oder persönlich übermittelt, oft mit spärlichem Material (Text, Bilder, Video). Das Start- und Enddatum der Nachricht wurde nicht festgelegt. Die Häufigkeit der Anfragen zur Veröffentlichung von Nachrichten stieg auf etwa hundert Nachrichten pro Monat und einige Anfragen überschnitten sich (zwei oder mehr Abteilungsleiter hatten ähnliche oder gleiche Themen). Der IT-Administrator musste die Probleme zwischen und mit den Managern klären. Die Festlegung von Start- und Endterminen für das Erscheinen von Nachrichten wurde oft dem IT-Administrator überlassen, was zu schlechter Stimmung bei Managern und IT-Mitarbeitern führte. Auch die Verwaltung der statischen Seite wurde zeitaufwendig und extrem fehleranfällig. Die Lösung erforderte sowohl eine Prozessänderung als auch eine neue Veröffentlichungsplattform.

**Prozessänderung und neue Veröffentlichungsplattform** Für die Entwicklung der neuen Veröffentlichungsplattform wurde eine Low-Code-Anwendungsentwicklungsumgebung gewählt. Das Unternehmen verwendet bereits eine Oracle-Datenbank, daher wurde Oracle APEX als Entwicklungsumgebung ausgewählt. Es werden drei Arten von Benutzern der Plattform unterschieden. Zunächst der Administrator, der Zugang zu allen Daten und die Berechtigung hat, Benutzer und

Rollen zu verwalten. Zusätzlich zu den Basisdaten der Mitarbeiter werden auch deren Einsätze in den Abteilungen und deren Standorte erfasst. Jedem Mitarbeiter ist eine bestimmte Rolle zugewiesen, die sich im Laufe der Zeit ändern kann. Der zweite Benutzertyp ist der Herausgeber, der nicht das Recht hat, Mitarbeiterdaten zu verwalten und Berechtigungen zu definieren, aber das Recht hat, Nachrichten zu veröffentlichen. Der dritte Benutzertyp ist der Betrachter/Leser der Nachrichten, der das Recht hat, Internetnachrichten zu lesen.

**Anforderungsspezifikation** Auf der Grundlage des geänderten Prozesses bietet die Plattform folgende Möglichkeiten:

- Betrachter/Leser: Anmeldung, Zugriff auf das Dashboard, Anzeige von Nachrichten und Anhängen zu Nachrichten
- Herausgeber: Anmeldung, Zugriff auf das Dashboard, Anzeigen und Veröffentlichen von Nachrichten und Hinzufügen von Anhängen zu veröffentlichten Nachrichten
- Administrator: Anmeldung, Zugriff auf das Dashboard, Anzeigen und Veröffentlichen von Nachrichten, Anzeigen und Verwalten von Abteilungen, Rollen, Mitarbeitern, Mitarbeiterrollen, Verwalten von Anwendungen

## 13.3 Anwendungsfälle

### 13.3.1 Narrative Beschreibung des Anwendungsfalls

Die Veröffentlichungsplattform ermöglicht den Zugang zu drei verschiedenen Arten von Mitarbeitern: Administrator, Betrachter/Leser und Mitwirkende/Herausgeber, die jeweils unterschiedliche Berechtigungen auf dem Intranetportal haben. Der Herausgeber kann Nachrichten veröffentlichen, der Leser kann Nachrichten lesen und der Administrator kann die Plattform verwalten. Für jeden dieser Anwendungsfälle muss sich ein Benutzer bei der Intranet-Plattform anmelden.

### 13.3.2 Semi-strukturierte Beschreibung

Wir können drei verschiedene Benutzergeschichten oder uce-Fälle zusammenfassen: Veröffentlichung, Lesen und Verwaltung.

Das Hinzufügen eines Anhangs zu den Nachrichten ist eigentlich eine Erweiterung der oben beschriebenen UC und wir könnten es als neuen Anwendungsfall beschreiben. Aus Gründen der Lesbarkeit fahren wir mit den anderen beiden Hauptanwendungsfällen fort, jedoch wird die Anwendung eine Funktion zum Hinzufügen von Anhängen zu den Nachrichten haben.

### 13.3.3 Anwendungsfalldiagramm

Die obige Geschichte ist im Anwendungsfalldiagramm Figure 13.1 dargestellt.

Tabelle 13.1: c: Veröffentlichung von Internet-Nachrichten.

| **Keyword** | **Wert** |
|---|---|
| ID: | *Ch13-01* |
| Titel: | *Nachrichten im Intranet-Portal veröffentlichen* |
| Beschreibung: | *Eine verantwortliche Person aus der Personalabteilung, der eine Herausgeberrolle zugewiesen wurde, verwendet das APEX-Portal im Intranet, um Nachrichten zu veröffentlichen. Jede Nachricht hat ein Start- und Enddatum. Das Standard-Startdatum ist der Tag der Veröffentlichung.* |
| Primärer Akteur: | *Mitarbeiter mit Veröffentlichungsfunktion* |
| Voraussetzungen: | *Der Mitarbeiter muss in der Mitarbeitertabelle aufgeführt sein und der Benutzer muss der Rolle CH13 Publisher hinzugefügt werden. Der Zugriff auf die Webanwendung muss aktiviert werden.* |
| Nachbedingungen: | *Nach erfolgreicher Veröffentlichung von Nachrichten sind die Nachrichten und ihr Anhang für andere Benutzer des Intranetportals verfügbar.* |
| *Haupt* | *Szenarien* |
| Erfolgs-Szenario: | 1. *Öffnen Sie den Webbrowser und melden Sie sich bei der Intranet-Anwendung an*<br>2. *Wählen Sie Nachrichten veröffentlichen*<br>3. *Geben Sie den Titel, die Beschreibung und das Enddatum der Nachricht ein*<br>4. *Bestätigen Sie die Erstellung der Nachrichten*<br>5. *Anhang/Anhänge hinzufügen*<br>6. *Details zum Anhang eingeben*<br>7. *Bestätigen Sie das Hinzufügen des Anhangs*<br>8. *Überprüfen Sie die veröffentlichten Nachrichten* |
| Erweiterungen: | • *1a. Anmeldung schlägt fehl*<br>• *1a* Erweitern:*<br>• *1a1. Fehlermeldung anzeigen*<br>• *1a2. Anmeldefenster öffnen*<br>• *4a. Datentyp-Fehler*<br>• *4a* Erweitern:*<br>• *4a1. Fehlermeldung anzeigen*<br>• *7a. Datentyp-Fehler*<br>• *7a* Erweitern:*<br>• *7a1. Fehlermeldung anzeigen* |
| Häufigkeit der Nutzung: | *Die Betreiber veröffentlichen etwa 1000 Nachrichten pro Jahr, durchschnittlich 5 pro Tag* |
| Status: | *Fertiggestellt* |
| Eigentümer: | *Mitarbeiter mit Veröffentlichungsfunktion* |
| Priorität: | *moderat* |

Tabelle 13.2: Beschreibung des Anwendungsfalls: Lesen von Intranet-Nachrichten

| **Keyword** | **Wert** |
|---|---|
| ID: | *Ch13-02* |
| Titel: | *Nachrichten im Intranet-Portal lesen* |
| Beschreibung: | *Intranet-Benutzer liest Nachrichten* |
| Primärer Akteur: | *Intranet-Benutzer* |
| Voraussetzungen: | *Der Mitarbeiter muss in der Mitarbeitertabelle aufgeführt sein und der Benutzer muss der Rolle CH13 Viewer hinzugefügt werden. Der Zugriff auf die Webanwendung muss aktiviert sein.* |
| Nachbedingungen: | |
| *Haupt* | *Szenarien* |
| Erfolgs-Szenario: | 1. *Öffnen Sie den Webbrowser und melden Sie sich bei der Intranet-Anwendung an.*<br>2. *Wählen Sie Nachrichten anzeigen*<br>3. *Wählen Sie Download, um den Anhang anzuzeigen.* |
| Erweiterungen: | • *1a. Anmeldung schlägt fehl*<br>• *1a* : Erweitern*<br>• *1a1. Fehlermeldung anzeigen* |
| Häufigkeit der Nutzung: | *Mitarbeiter lesen täglich Nachrichten.* |
| Status: | *Fertiggestellt* |
| Eigentümer: | *Intranet-Benutzer* |
| Priorität: | *gering* |

Tabelle 13.3: Beschreibung des Anwendungsfalls: Verwaltung des Intranetportals.

| **Keyword** | **Wert** |
|---|---|
| ID: | *Ch13-03* |
| Titel: | *Verwaltung des Intranetportals* |
| Beschreibung: | *Verwalten des Intranetportals* |
| Primärer Akteur: | *Verwalter des Intranetportals* |
| Voraussetzungen: | *Der Mitarbeiter muss in der Tabelle „Mitarbeiter“ aufgeführt sein, und der Benutzer muss der Rolle „Administrator“ zugeordnet sein. Der Zugriff auf die Webanwendung muss aktiviert werden.* |
| Nachbedingungen: | *Benutzer, Rollen und Mitarbeiter sind bereit für die Verwendung im Intranetportal.* |
| *Haupt* | *Szenarien* |
| Erfolgs-Szenario: | 1. *Öffnen Sie den Webbrowser und melden Sie sich bei der Intranet-Anwendung an*<br>2. *Fügen Sie Daten zu Abteilungen hinzu und verwalten Sie diese*<br>3. *Fügen Sie Daten zu Rollen hinzu und verwalten Sie diese*<br>4. *Fügen Sie Mitarbeiter und ihre Rollen hinzu und verwalten Sie diese*<br>5. *Nachrichten veröffentlichen*<br>6. *Nachrichten ansehen*<br>7. *Verwaltung* |
| Erweiterungen: | • *1a. Anmeldung schlägt fehl*<br>• *1a* Erweitern:*<br>• *1a1. Fehlermeldung anzeigen*<br>• *1a2. Anmeldefenster öffnen*<br>• *2a. Datentyp-Fehler*<br>• *2a* Erweitern:*<br>• *2a1. Fehlermeldung anzeigen*<br>• *3a. Datentyp-Fehler*<br>• *3a* Erweitern:*<br>• *3a1. Fehlermeldung anzeigen*<br>• *4a. Datentyp-Fehler*<br>• *4a* Erweitern:*<br>• *4a1. Fehlermeldung anzeigen*<br>• *5a. Datentyp-Fehler*<br>• *5a* Erweitern:*<br>• *5a1. Fehlermeldung anzeigen* |
| Häufigkeit der Nutzung: | *Die Häufigkeit der Inanspruchnahme hängt von neuen Mitarbeitern und Änderungen der Privilegienregelung ab. Ungefähr 5 Mal pro Woche.* |
| Status: | *Fertiggestellt* |
| Eigentümer: | *Verwalter des Intranetportals* |
| Priorität: | *hoch* |

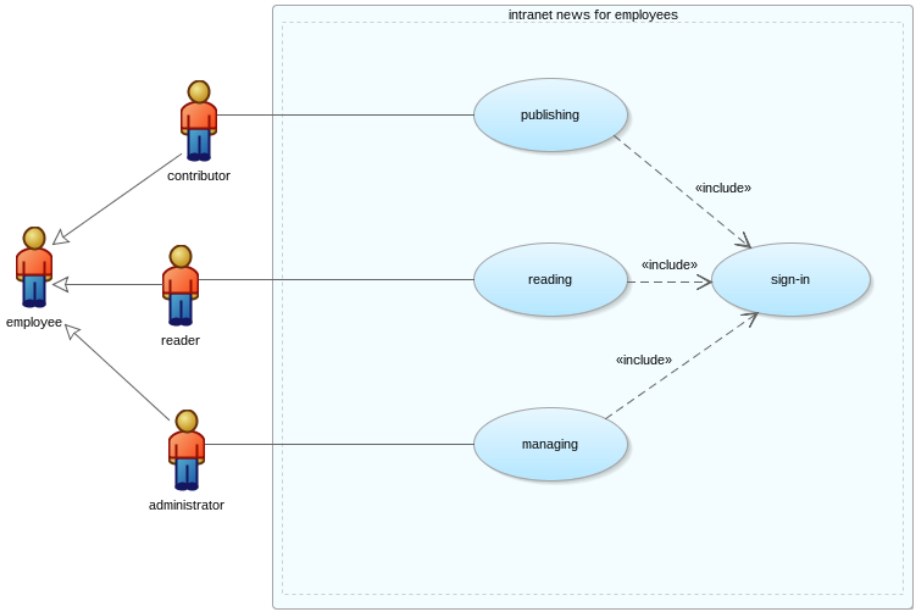



Abbildung 13.1: Anwendungsfalldiagramm.

Auch hier wird der Einfachheit halber der Anwendungsfall der Erweiterung (Hinzufügen von Anhängen zu den Nachrichten) in Figure 13.1 übersprungen.

## 13.4 Datenmodell

### 13.4.1 Narrative Beschreibung des Datenmodells

Es gibt sechs Entitäten im logischen Datenmodell. Die Entität CH13 DEPARTMENT hat drei Attribute: ID, Name und Standort der Abteilung. Jede Abteilung kann viele Mitarbeiter haben. Grundlegende Daten über Mitarbeiter werden in der Entität CH13 EMPLOYEE gespeichert: ID, Vor- und Nachname, Geburtsdatum und E-Mail. Der Mitarbeiter kann ein Manager sein. Jeder Mitarbeiter hat nur einen Vorgesetzten, während ein Vorgesetzter mehrere Mitarbeiter verwalten kann. Jeder Mitarbeiter kann eine Rolle haben. Rollen werden durch ID, Name und Beschreibung definiert. Für jeden Mitarbeiter und jede Rolle wird ein Anfangs- und Enddatum festgelegt. Im Laufe der Zeit kann ein Mitarbeiter mehrere Rollen haben, und es können auch mehrere Mitarbeiter einer Rolle zugewiesen werden. Allerdings kann eine Definition einer Mitarbeiterrolle nicht auf einen anderen Mitarbeiter oder eine andere Rolle übertragen werden. Ein Mitarbeiter kann im Laufe der Zeit mehrere Nachrichten veröffentlichen. Jede Nachricht wird durch ID, Titel, Beschreibung, Start- und Enddatum identifiziert. Jede Nachricht kann mehrere Anhänge haben. Ein Anhang wird mit ID, Dateiname, Mime-Typ, Erstellungsdatum und Inhalt identifiziert.

### 13.4.2 Logisches Datenmodell

Die obige Geschichte ist im logischen Datenmodell Figure 13.2 abgebildet.

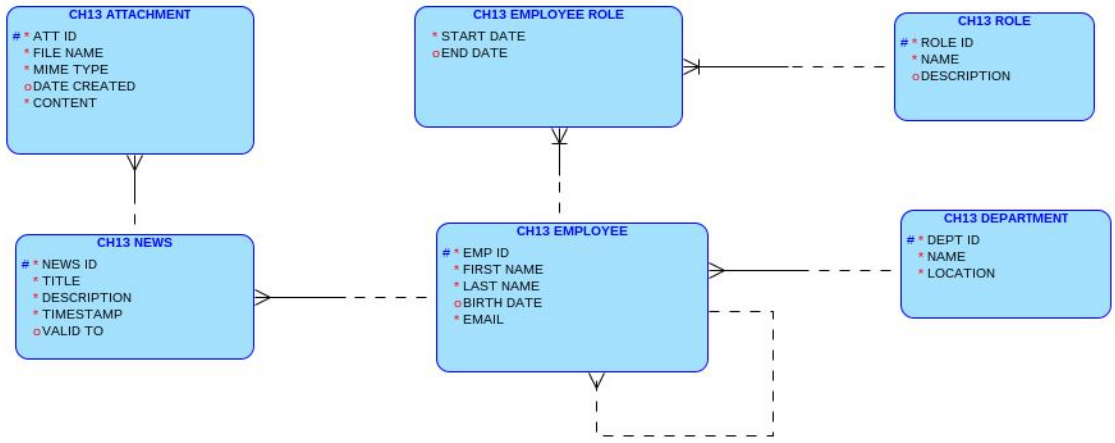



Abbildung 13.2: Logisches Datenmodell.

### 13.4.3 Relationales Datenmodell

Die automatische Umwandlung vom logischen Datenmodell in das relationale Datenmodell in Oracle SQL Data Modeler wird durch die Funktion *Engineering to relational* bereitgestellt. Das Ergebnis wird in Figure 13.3 angezeigt.

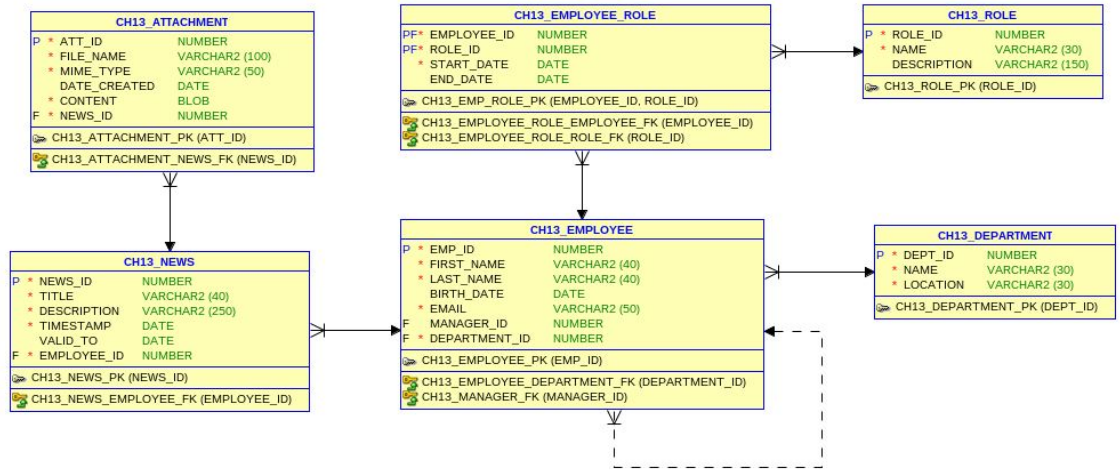



Abbildung 13.3: Relationales Datenmodell.

Der Oracle SQL Data Modeler generiert auch SQL-Skripte für die Erstellung von Tabellen, Sequenzen und Triggern. Wählen Sie alle Tabellen im relationalen Modell aus und verwenden Sie die Funktion *File* > *Export* > *DDL File*, um ein Skript wie dieses zu erhalten:

```
CREATE TABLE CH13_NEWS (
NEWS_ID NUMBER NOT NULL
...
);

CREATE TABLE CH13_DEPARTMENT (
DEPT_ID NUMBER NOT NULL
...
);
```

Jetzt ist es an der Zeit, Tabellen in der Oracle-Datenbank zu erzeugen. Wir können das generierte Skript importieren und es in APEX ausführen.

## 13.5 Anwendungsschnittstellen

In den folgenden Abbildungen werden die Anwendungsschnittstellen für alle drei Rollen dargestellt. Zunächst wird das Dashboard der Mitwirkenden (Herausgeber) in Figure 13.4 dargestellt.

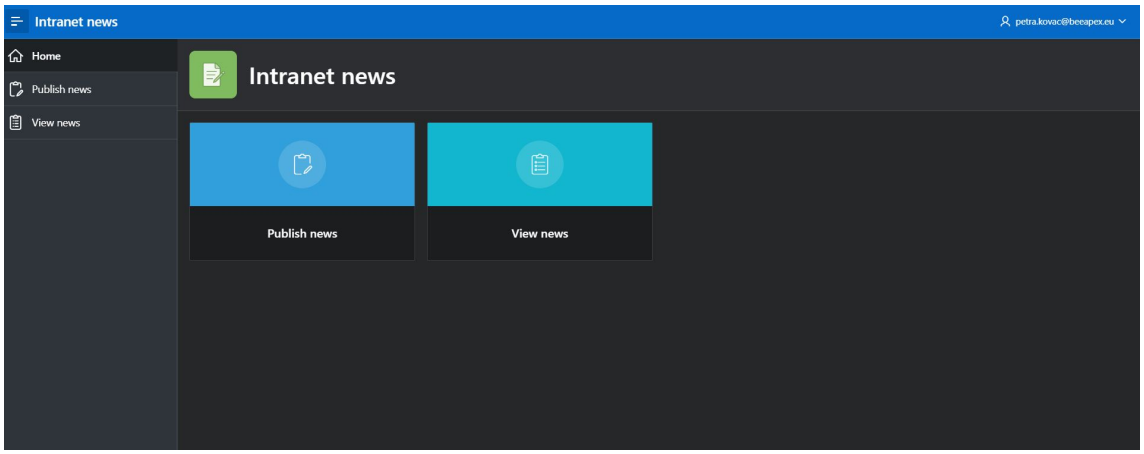



Abbildung 13.4: Das Dashboard der Mitwirkenden (Herausgeber).

Die Seite für die Veröffentlichung von Nachrichten mit Anhängen und ihre Einzelheiten sind in Figure 13.5 dargestellt.

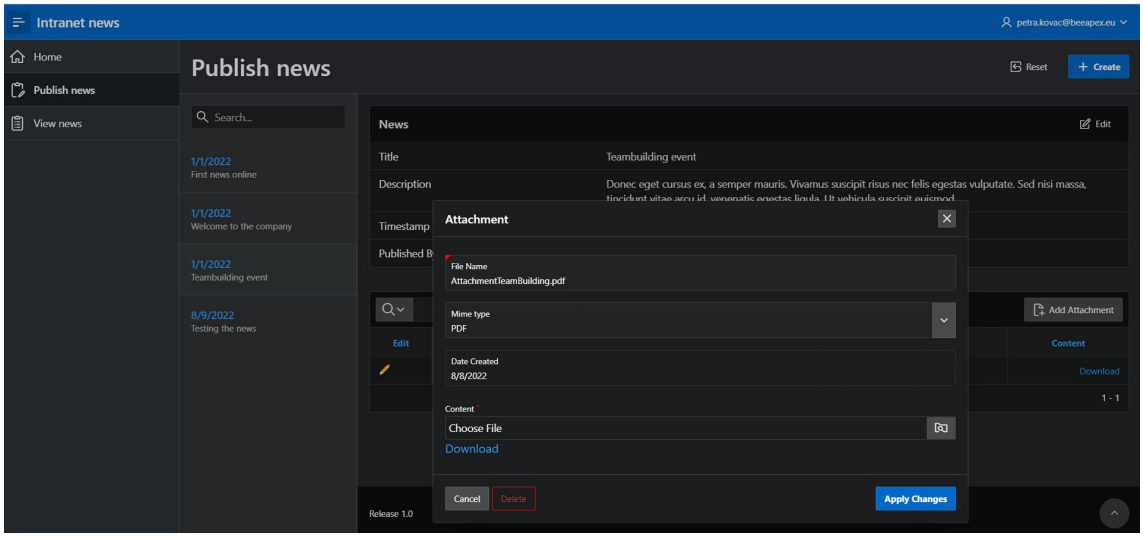



Abbildung 13.5: Nachrichten mit Anhängen veröffentlichen.

Die Rolle des Betrachters (Lesers) hat das am meisten eingeschränkte Dashboard. Im Moment sind nur die Nachrichten „Home“ und „View“ verfügbar, wie in Figure 13.6 dargestellt.

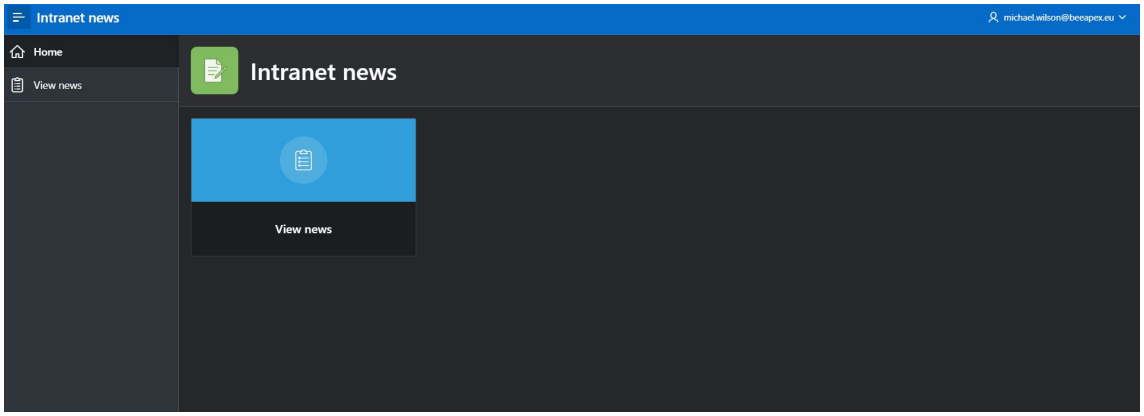



Abbildung 13.6: Betrachter (Leser) Dashboard.

Die wichtigste Funktion für den Betrachter (Leser) ist der Zugriff auf Nachrichten (siehe Figure 13.7).

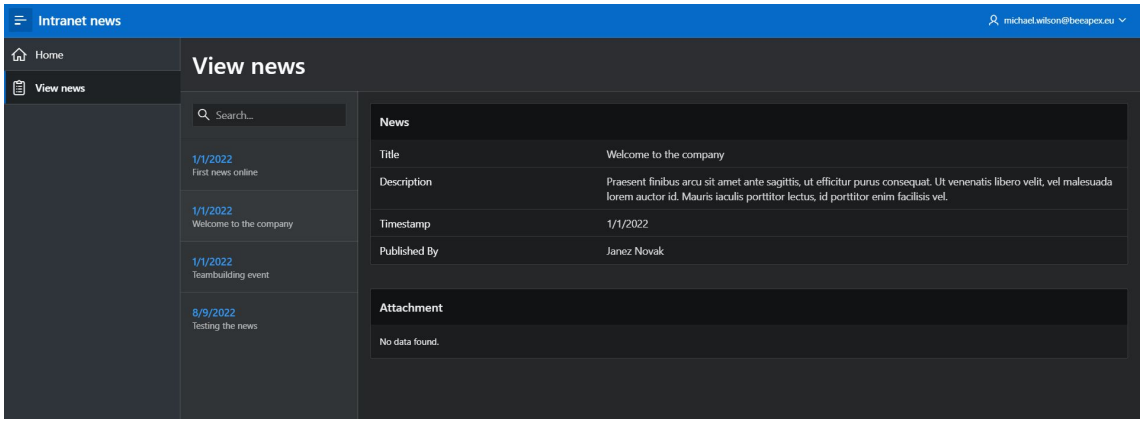



Abbildung 13.7: Betrachter (Leser) Zugriff auf Nachrichten.

Neben der Anzeige und Veröffentlichung von Nachrichten kann der Administrator Abteilungen, Rollen und Mitarbeiter mit Rollendaten verwalten, wie in Figure 13.8 dargestellt. Außerdem ist die Verwaltungsseite verfügbar, auf der die Benutzerverwaltung möglich ist.

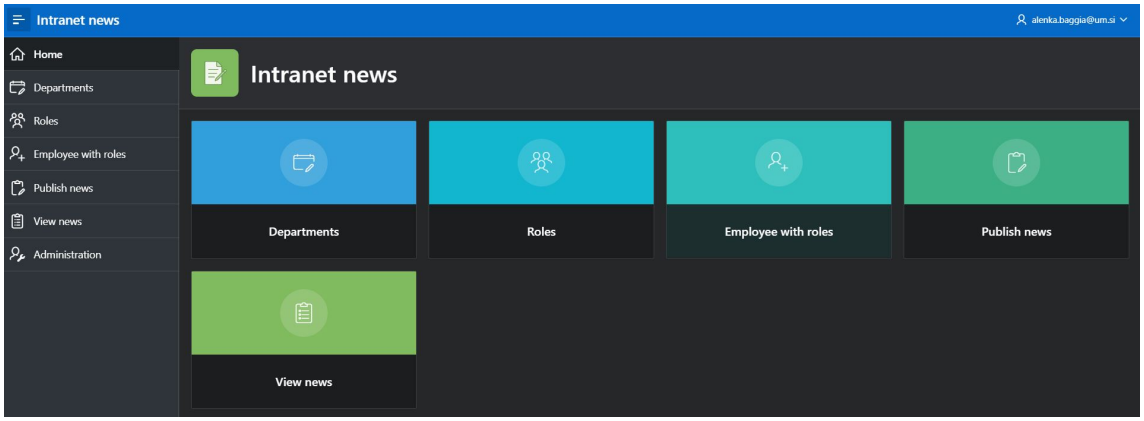



Abbildung 13.8: Dashboard für Administratoren.

Die Seite zur Bearbeitung der Daten von Mitarbeiterrollen wird in Figure 13.9 dargestellt. Diese

Seite ist nur für den Administrator des Intranetportals verfügbar.

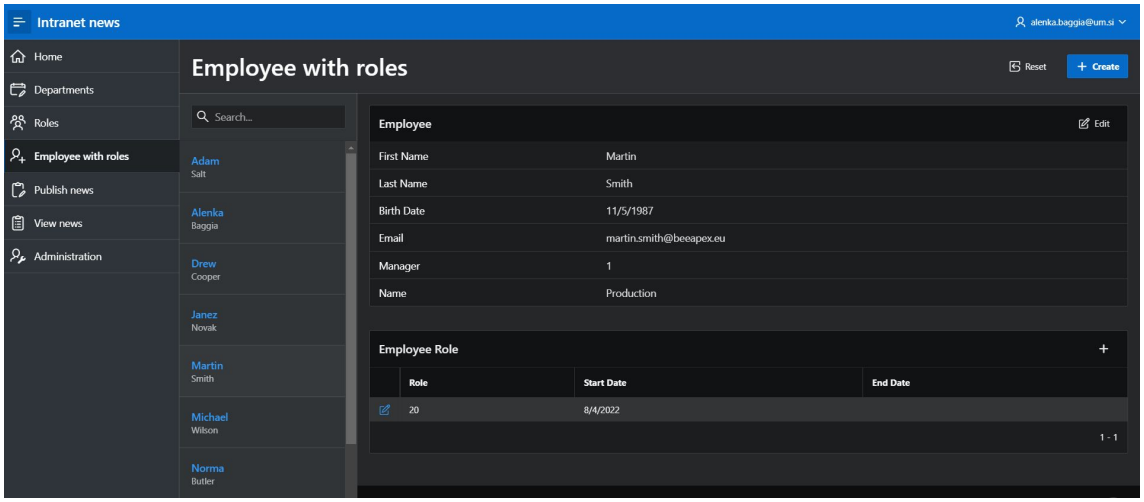



Abbildung 13.9: Bearbeitung der Mitarbeiterrolle durch den Administrator.

## 13.6 Ergänzendes Lernmaterial

Sie können das folgende zusätzliche Lernmaterial finden:

- exportierte Anwendung
- Video-Anleitungen

Alle zusätzlichen Lernmaterialien sind auf der öffentlichen BeeAPEX-Projektseite verfügbar. Melden Sie sich als Gastbenutzer an (kein Passwort erforderlich). Das Lehrbuch finden Sie im Abschnitt Books, die Skripte im Ordner Part 1 > Chapter12 im Abschnitt Scripts und die Videoanleitungen in der Sammlung der Videoanleitungen. Material für Kurzkurse finden Sie in der Sektion Short courses.

### 13.6.1 Exportierte Anwendungen

Die exportierte Anwendung wird verpackt. Bei der Installation werden Tabellen erstellt und Daten eingepflegt. Bei der Deinstallation werden alle in dieser Anwendung verwendeten Datenbankobjekte entfernt.

### 13.6.2 Video-Anleitungen

Videoanleitungen zeigen jeden Schritt der Anwendungsentwicklung.

## 13.7 Fragen

1. Wie würden Sie das logische Datenmodell ändern, um die folgende Anforderung umzusetzen: Ein Anhang kann zu mehreren Nachrichten gehören.
2. Wie würde sich das relationale Datenmodell ändern, wenn die Anforderung (eine Anlage kann zu mehreren Nachrichten gehören) umgesetzt wird?
3. Welche Benutzerrollen sind in der Anwendung in Kapitel 13 implementiert und warum?

## 13.8 Antworten

1. One-to-many-Beziehungen sollten in der Richtung vom Anhang zu den Nachrichten verbunden sein.
2. Das Paar attachment-news im logischen Modell würde in drei relationale Tabellen umgewandelt werden: attachment, news und attachment_news. Die letzte Tabelle würde mindestens

zwei Felder enthalten (IDs von Attachment und News), die beide als Fremdschlüssel dienen.

3. In Kapitel 13 sind die folgenden Benutzerrollen implementiert: Administrator, Mitwirkender und Leser. Diese Rollen sind implementiert, um autorisierte Benutzer in die Lage zu versetzen, ihre Arbeit zu erledigen, und um zu verhindern, dass nicht autorisierte Benutzer nicht genehmigte Aktionen durchführen.



# 14. GreenDi - Pflanzenkatalog

VJERAN STRAHONJA, DIJANA OREŠKI, DARKO ANDROČEC, ANA KUTNJAK

## 14.1 Wirtschaftliche Sicht des Falles

Eine Gruppe von Enthusiasten, die sich in der gemeinnützigen Organisation GreenDi zusammengeschlossen hat, hat verschiedene Initiativen im Zusammenhang mit dem Gärtnern in der Stadt, gesunder Ernährung, dem Anbau alter heimischer Sorten, der Zusammenarbeit mit örtlichen Landwirten usw. ins Leben gerufen und umgesetzt. Bislang haben sie bei ihrer Arbeit soziale Netzwerke genutzt. Deshalb haben sie beschlossen, ihr eigenes GreenDi zu gründen, eine Peer-to-Peer-Plattform für den Austausch von Saatgut und Pflanzen, mit dem Ziel, die biologische Vielfalt zu fördern und die Aktivitäten ihrer bisherigen Aktivitäten zu unterstützen und neue zu ermöglichen. Um eine solche Plattform zu entwickeln und zu implementieren, müssen die Anforderungen und die grundlegenden Modelle, einschließlich eines Datenmodells und der Anwendungsfälle, festgelegt werden.

## 14.2 Problemstellung

Der öffentliche Teil der Plattform weist die folgenden Merkmale auf:

- Die Nutzer von GreenDi sind nicht auf einen geografischen Ort beschränkt, sondern können weltweit sein. Da es sich jedoch um eine Pflanzenbörse handelt, ist die Bildung lokaler, auf ein bestimmtes geografisches Gebiet bezogener Netzwerke zu erwarten.
- Die Plattform ist in mehrere thematische Einheiten unterteilt, wie z.B. traditionelle und einheimische Sorten und Saatgut (Gemüse, Obst, Heilkräuter, Zimmer- und Zierpflanzen), wobei eine Ausweitung auf Industrie- und andere Pflanzen geplant ist.
- Die Grundlage des öffentliche Teils ist ein Pflanzenkatalog mit dem Namen in der Sprache des Benutzers, Latein und Englisch, der Klassifizierung der Pflanze (Taxonomie), der Beschreibung, den Lebensräumen, dem Anbau und der Verwendung, Fotos und anderen grundlegenden Informationen.
- Grundlegende Informationen über die Pflanze können ohne Registrierung gesucht und eingesehen werden.
- Der GreenDi-Pflanzenkatalog ist mit mehreren offenen Webseiten, Datenbanken und externen Diensten verknüpft, die Informationen über Pflanzen enthalten. Es werden URL-Adressen, Hyperlinks und andere Arten von markierten Daten verwendet.
- Der Katalog enthält Pflanzen, die auf der GreenDi-Plattform getauscht werden, aber auch

solche, die noch nicht getauscht oder in ein Angebot aufgenommen wurden. Der Katalog enthält auch Links zu kommerziellen Anbietern, mit denen die Mitglieder gute Erfahrungen gemacht haben und die den ökologischen Landbau fördern.

Die Katalogverwaltung wird mit den folgenden Sätzen beschrieben:

- Der Pflanzenkatalog wird von den Administratoren der GreenDi-Plattform verwaltet.
- Die Katalogverwaltung umfasst die Eingabe und Änderung von Informationen über Pflanzen.
- Einige Funktionalitäten der Katalogverwaltung stehen den vom Administrator autorisierten Benutzern zur Verfügung (autorisierter Benutzer).
- Zu den erweiterten Funktionen, die den Administratoren zur Verfügung stehen, gehört die Verknüpfung mit anderen Seiten, die Informationen über die Pflanze, Videos über den Anbau, die Verwendung der Pflanzen usw. enthalten.
- Alle registrierten Benutzer (Mitglieder) können, wenn sie es wünschen, ihre Meinung über die Pflanze mitteilen, eine Bewertung schreiben, weitere Informationen hinzufügen, usw., aber das ist nicht Teil dieser Funktion.

## 14.3 Anwendungsfälle

### 14.3.1 Narrative Beschreibung des Anwendungsfalls

Der öffentliche Teil der GreenDi-Plattform ermöglicht es jedem Nutzer von jedem Ort aus, die Daten des Pflanzenkatalogs ohne Registrierung einzusehen. Gleichzeitig sind die Pflanzen in Kategorien eingeteilt, die es dem Nutzer ermöglichen, sich leichter zurechtzufinden. Für jede Pflanze werden Name, Taxonomie, Beschreibung, Lebensräume, Anbau und Verwendung, Fotos und andere grundlegende Informationen sowie Links zu externen Datenquellen gespeichert.

### 14.3.2 Semi-strukturierte Beschreibung

Tabelle 14.1 zeigt den Anwendungsfall.

### 14.3.3 Anwendungsfalldiagramm

Dieser Vorgang ist im Anwendungsfalldiagramm 14.1 abgebildet. Auf der rechten Seite des Anwendungsfalldiagramms befindet sich der Akteur External Source. Dieser Akteur stellt ein externes System dar, von dem GreenDi Catalog of Plants Informationen über Pflanzen, Saatgut und ähnliches abruft. Es gibt viele Beispiele für solche Pflanzendatenbanken, die offene Dienste oder offene Datensätze anbieten. Das bedeutet, dass die Datenbank oder die Daten frei zur Verfügung stehen, ohne Zugangs- oder Nutzungsbeschränkungen. Einige Beispiele für Online-Pflanzendatenbanken, die offene Dienste oder offene Daten anbieten, sind:

1. Die Global Biodiversity Information Facility (GBIF, https://www.gbif.org ) bietet freien und öffentlichen Zugang zu Daten über die biologische Vielfalt, einschließlich Daten über Pflanzen. GBIF ermöglicht den Nutzern die Suche und den Zugang zu Daten aus einer Reihe von Quellen, darunter Herbarien, Museen und Forschungseinrichtungen. Er verweist auf den International Plant Names Index (IPNI), eine Datenbank mit den Namen und zugehörigen grundlegenden bibliografischen Angaben zu Samen, Pflanzen, Farnen und Lycophyten (https://www.gbif.org/dataset/046bbc50-cae2-47ff-aa43-729fbf53f7c5).
2. Die WFO-Pflanzenliste (https://wfoplantlist.org/plant-list) wurde im Mai 2021 eingeführt und bietet eine benutzerfreundliche, zitierfähige statische Liste aller Pflanzenarten. Der gesamte Datensatz der WFO-Pflanzenliste kann auch als Ganzes heruntergeladen werden. Die Daten können auch über die WFO-Pflanzenliste-API abgerufen werden.
3. Der Datensatz Flora of Italy kann von der Website der Universität Rom heruntergeladen werden und enthält hochauflösende Bilder von über 10.000 Pflanzenarten in Italien.

Dies sind nur einige Beispiele für verfügbare Datensätze, die Bilder von Pflanzen und Samen in Europa enthalten, aber es gibt viele andere Datensätze, die ähnliche Informationen liefern. In

Tabelle 14.1: Beschreibung des Anwendungsfalls: Durchsuchen des Pflanzenkatalogs

| **Keyword** | **Wert** |
|---|---|
| ID: | *ch14-01* |
| Titel: | *Katalog lokaler Pflanzen durchsuchen* |
| Beschreibung: | *Der Katalog der heimischen Pflanzen ist für alle Interessierten öffentlich zugänglich. Die Anwendung ermöglicht die Suche mit Filtern auf allen Datenbankfeldern.* |
| Primärer Akteur: | *Jede Person* |
| Voraussetzungen: | *Grundlegende Informationen über die Anlage können ohne Registrierung gesucht und eingesehen werden.* |
| Nachbedingungen: | *Wird das Durchsuchen des Pflanzenkatalogs aus einem anderen Anwendungsfall (Erweitern) aufgerufen, wird die ID der ausgewählten Pflanze übertragen.* |
| *Haupt* | *Pflanzenkatalog* |
| Erfolgs-Szenario: | *Erstes Szenario, Pflanzenkatalog:*<br>1. *Jeder Benutzer öffnet auf der Startseite das Formular „Pflanzenkatalog durchsuchen“, welches ein Suchformular ist, das auf dem Prinzip der „Query by Example“ (QBE) basiert.*<br>2. *Die Suche ist anhand der folgenden Daten möglich: a) Thematische Einheit - Liste der Codes (alle standardmäßig oder einige spezifische Namen, wie Gemüse, Obst, Heilkräuter, Zimmer- und Zierpflanzen usw.); b) Art der Pflanze - hierarchisch organisierte Klassifizierung der Pflanze (alle standardmäßig oder einige spezifische Namen), c) Name(n) in der Sprache des Benutzers, d) lateinischer Name e) englische(r) Name(n)*<br>3. *Das Suchergebnis wird in tabellarischer Form angezeigt.*<br>4. *Es ist möglich, das tabellarische Formular nach Spalten zu sortieren.*<br>5. *Wenn Sie eine Zeile auswählen, können Sie das Formular mit den Details des Pflanzenformulars öffnen, das Folgendes enthält: Art der Pflanze, thematische Einheit, lokale Namen, englische Namen, lateinische Namen, Beschreibung, Lebensraum, Anbau, Verwendung, Fotos, Hyperlinks und Status.*<br>*Zweites Szenario: Katalogverwaltung:*<br>1. *Da die Katalogverwaltung von einem aktiven Benutzer vom Typ Administrator durchgeführt werden kann, ist der erste Schritt das Anmeldeverfahren (UC: Log In), bei dem die Erfüllung dieser Bedingungen geprüft wird.*<br>2. *Die Katalogverwaltung umfasst das Ändern oder Hinzufügen von Informationen über eine Pflanze, die sich bereits im Katalog befindet, das Löschen dieser Pflanze (wobei sie eigentlich ausgeblendet wird, da die Pflanze nicht physisch gelöscht wird) oder das Hinzufügen einer neuen Pflanze.*<br>3. *Der Benutzer bestätigt oder lehnt die Änderung ab.* |
| Erweiterungen: | • *Suchformular (QBE)*<br>• *Tabellarische Form*<br>• *Details der Pflanzenform* |
| Häufigkeit der Nutzung: | *Das Maximum liegt bei ca. 1000 Suchen pro Minute.* |
| Status: | *Entwicklungsstand* |
| Eigentümer: | *öffentlicher, anonymer Benutzer* |
| Priorität: | *hoch* |

diesem Beispiel werden wir die Kommunikation mit einer externen Quelle nicht realisieren, da dies fortgeschrittenere Kenntnisse erfordert.

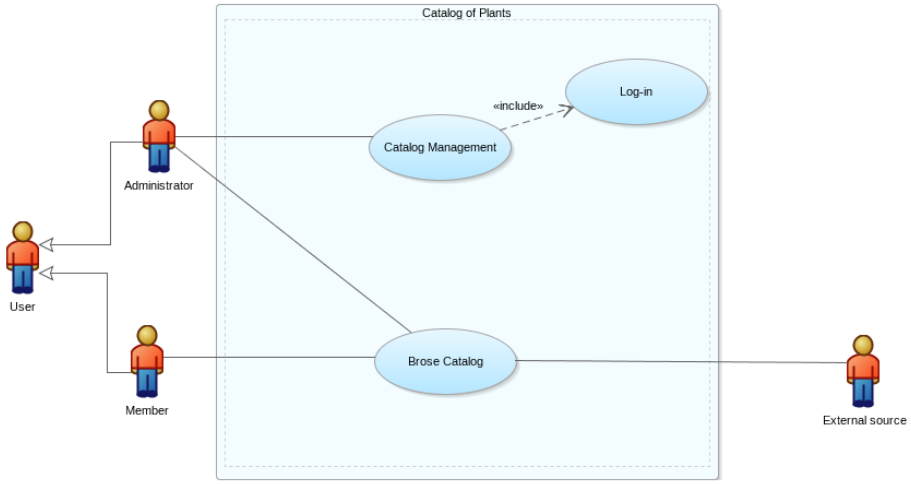



Abbildung 14.1: Anwendungsfalldiagramm - Pflanzenkatalog.

## 14.4 Datenmodell

### 14.4.1 Narrative Beschreibung des Datenmodells

Das logische Datenmodell (Entities-Relationship-Modell) enthält mehrere Entitäten. Das Basisattribut ist Plant, das mit den folgenden Attributen beschrieben wird: Plant_ID (Attribut zur Identifizierung der Pflanze); Name local (lokaler Name der Pflanze), Name English (englischer Name der Pflanze); Name Latin (lateinischer Name der Pflanze); Description (Beschreibung der Pflanze), Habitat (Beschreibung des Lebensraums, in dem die Pflanze vorkommt); Cultivation (Anbaumethode); Use (Verwendung der Pflanze); Photo (ein oder mehrere Fotos der Pflanze); Hyperlinks (Links zu externen Informationsquellen); Status (aktive Pflanze im Austausch, archivierte Pflanze...) Plant ist eine hierarchisch organisierte Codeliste. Das bedeutet, dass jeder Pflanzentyp 0, 1 oder mehr Untertypen von Pflanzen enthalten kann und als Untertyp in 0 oder 1 Supertyp von Plant enthalten sein kann. Die Thematic unit ist auch eine Codeliste, die Sorten von Pflanzen und Saatgut definiert (Gemüse, Obst, Heilkräuter, Zimmer- und Zierpflanzen...). Jede Pflanze gehört zu 0, 1 oder mehreren Thematic units. Jede Thematic unit klassifiziert 0, 1 oder mehr Pflanzen. Die Attribute der Thematic unit sind: Thematic unit_ID (Code oder Identifikationsattribut der Thematic unit; Name der Thematic unit (ein oder mehrere Namen). Das Modell zeigt auch den Entitätstyp User. Außerdem wurden einige Entitätstypen zu diesem Datenmodell hinzugefügt: Plant optional data, Type of optional data und Type of plant und Thematic unit. Optionale Daten zu Pflanzen umfassen die folgenden Attribute: Plant optional_ID (Identifikationsattribut); Description (Inhalt zusätzlicher optionaler Daten, d.h. Linkbeschreibung, falls vorhanden) Link (optional). Typ der optionalen Angaben enthält die Attribute: Type_ID (Identifikationsattribut); Name of type (allgemeine Informationen, Verwendung, Anbau, persönliche Meinung, Prüfung...). Jede Pflanze kann 0, 1 oder mehr Plant optional data haben. Jede Plant optional data bezieht sich auf 1 und nur 1 Plant. Jede Plant optional data gehört zu 1 und nur 1 Type of optional data. Jede Type of optional data klassifiziert 0, 1 oder mehrere Plant optional data. Plant optional data werden von 1 und nur 1 User eingegeben.

### 14.4.2 Logisches Datenmodell

Das logische Datenmodell wird in Figure 14.2 dargestellt.

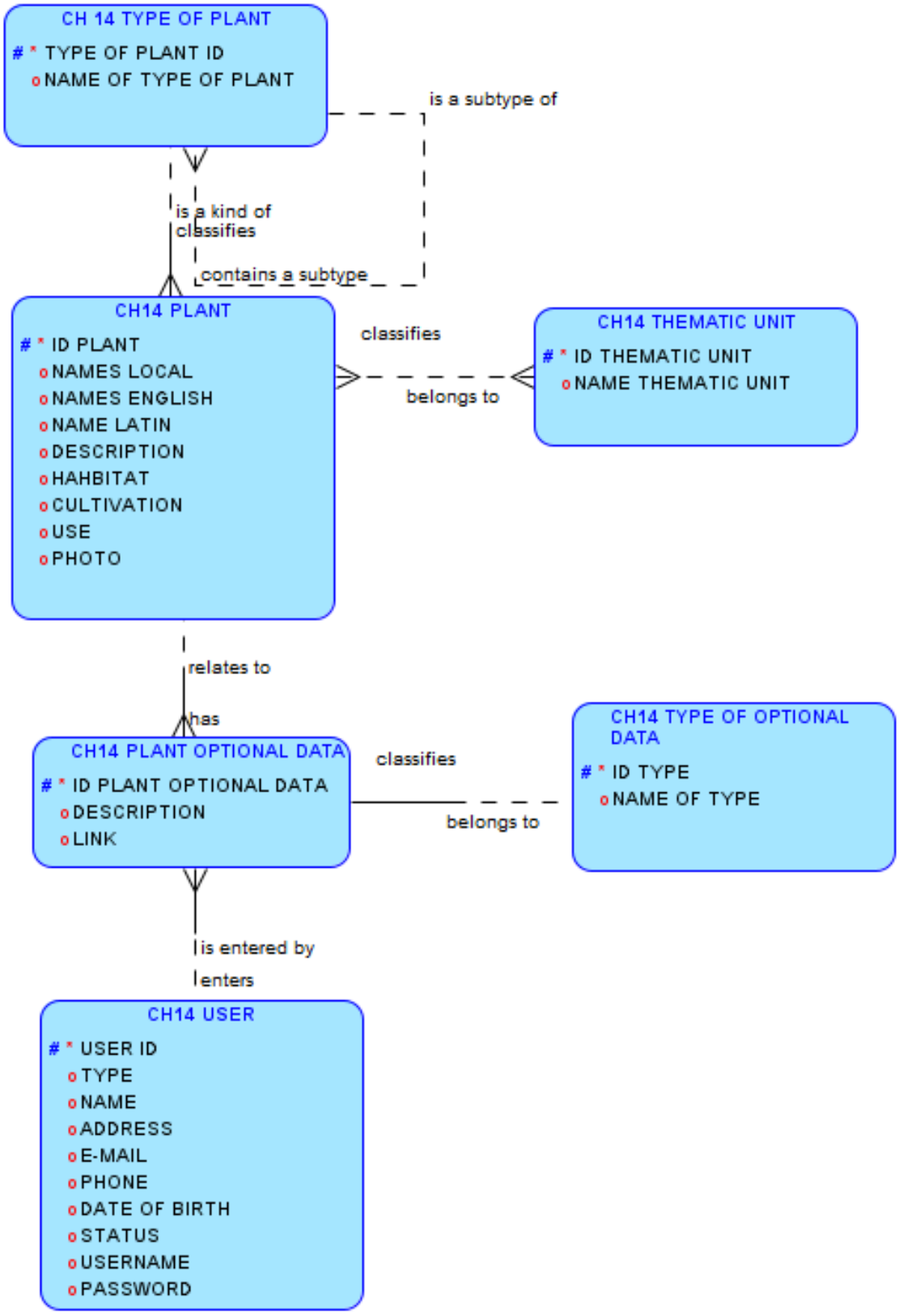



Abbildung 14.2: Logisches Datenmodell.

### 14.4.3 Relationales Datenmodell

Die automatische Umwandlung vom logischen Datenmodell in das relationale Datenmodell in Oracle SQL Data Modeler wird durch die Funktion *Engineering to relational* bereitgestellt. Das Ergebnis wird in Figure 14.3 angezeigt.

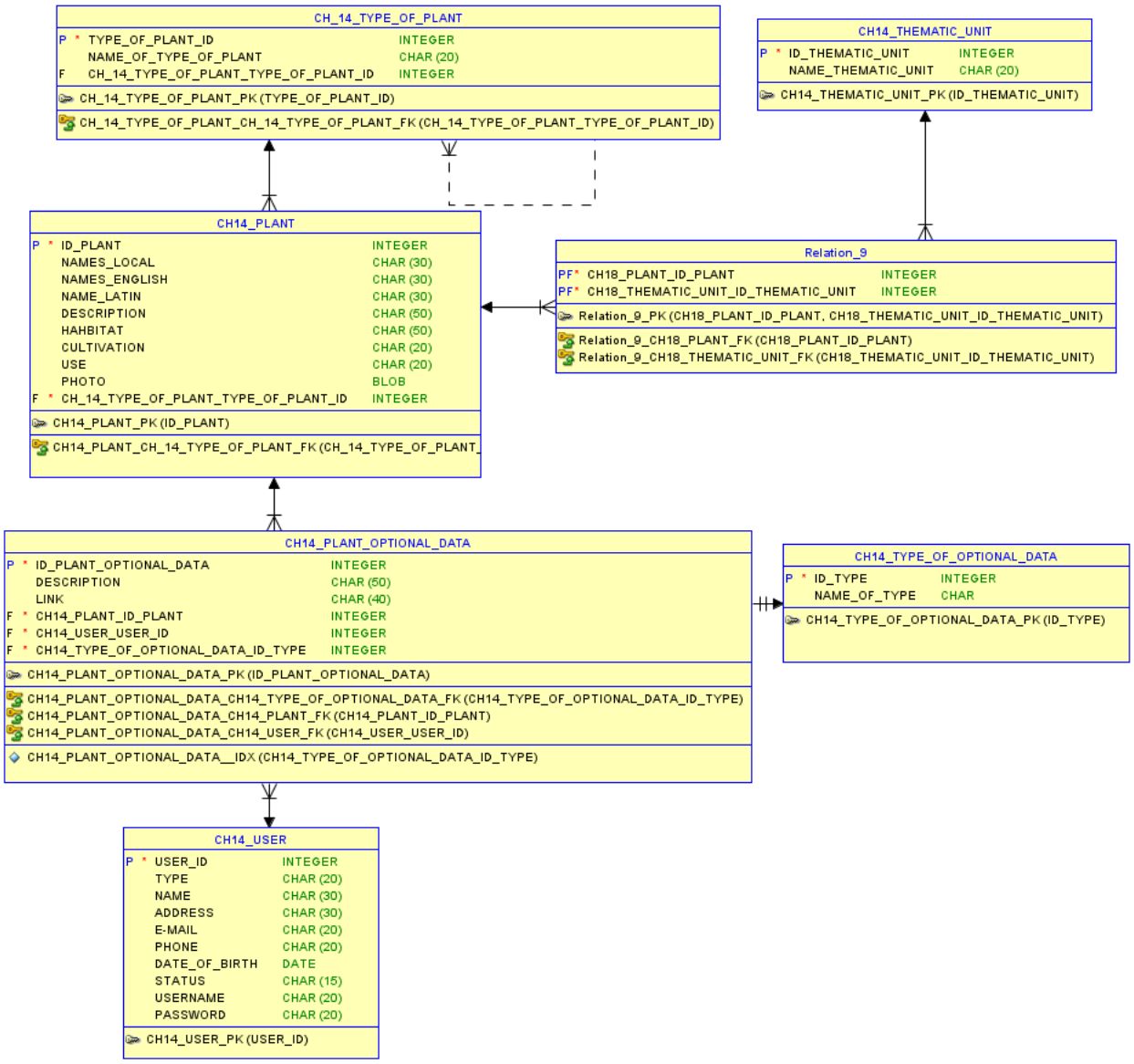



Abbildung 14.3: Relationales Datenmodell.

Lassen Sie nun Oracle SQL Data Modeler ein SQL-Skript generieren. Wählen Sie alle Tabellen im relationalen Modell und verwenden Sie die Funktion *File* > *Export* > *DDL File*, um ein Skript zu erhalten. Wir können das generierte Skript in APEX importieren und es ausführen.

## 14.5 Anwendungsschnittstellen

Einzelheiten zum interaktiven Raster für Plant Form sind in Figure 14.4 dargestellt.

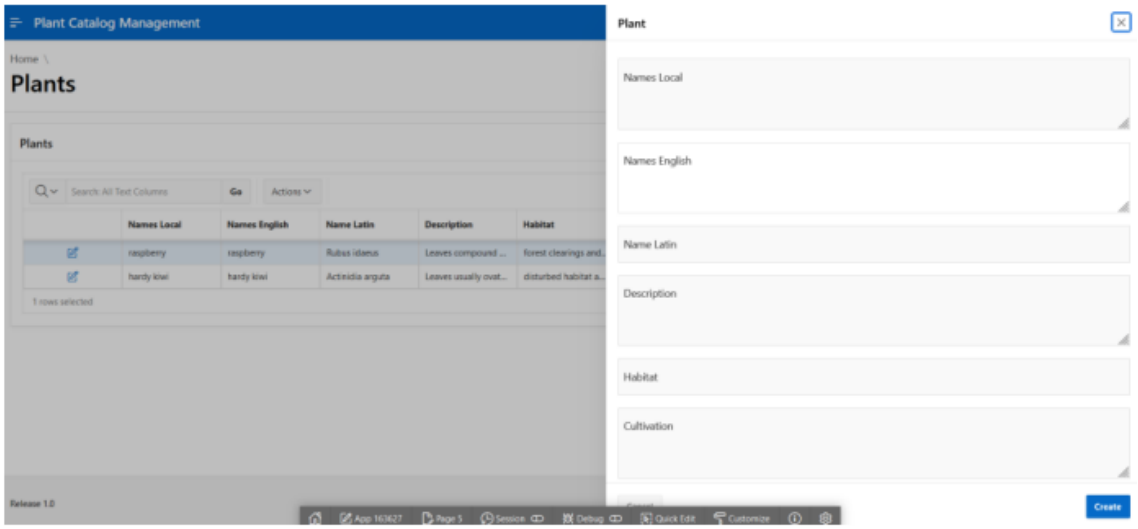



Abbildung 14.4: Interaktives Gitter für Plant Form.

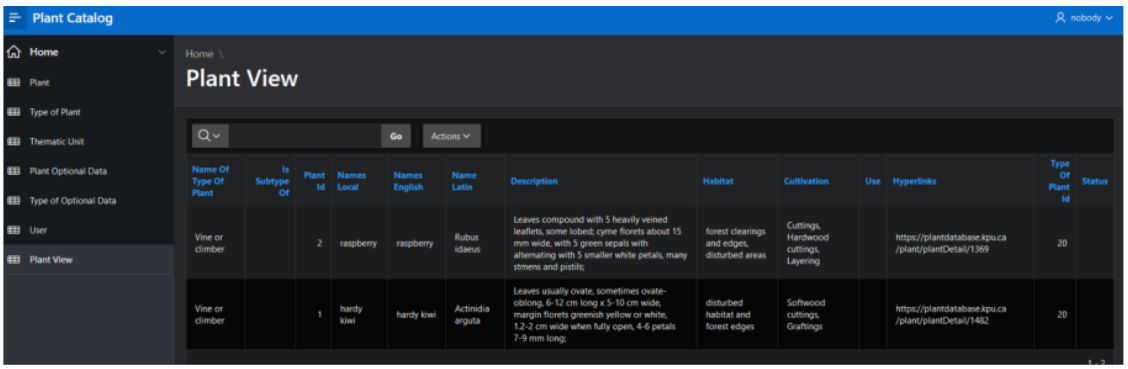



Abbildung 14.5: Öffentlicher Teil - offenes Browsing von Pflanzendaten.

Das Durchsuchen des Katalogs der lokalen Pflanzen ist mit einem Suchformular implementiert.

## 14.6 Ergänzendes Lernmaterial

Sie können das folgende zusätzliche Lernmaterial finden:

- Skript zum Erstellen und Auffüllen von Tabellen
- Skript zum Löschen von Tabellen
- exportierte verpackte Anwendung
- Video, das zeigt, wie man die Anwendung erstellt

Alle zusätzlichen Lernmaterialien sind auf der öffentlichen BeeAPEX-Projektseite verfügbar. Melden Sie sich als Gastbenutzer an (kein Passwort erforderlich). Das Lehrbuch finden Sie im Abschnitt Books, die Skripte im Ordner Part 2 > Chapter14 im Abschnitt Scripts und die Videoanleitungen in der Sammlung der Videoanleitungen. Material für Kurzkurse finden Sie in der Sektion Short courses.

### 14.6.1 Exportierte Anwendungen

Die exportierte Anwendung ist verpackt und verwendet nur zwei Tabellen (ch14_type_of_plant und ch14_plants). Bei der Installation werden sowohl Tabellen erstellt als auch Daten eingepflegt. Bei der Deinstallation werden alle in dieser Anwendung verwendeten Datenbankobjekte entfernt.

Die verpackte Anwendung ist getestet und wird im neuen Arbeitsbereich ausgeführt, wenn die folgenden Anforderungen erfüllt sind:

- Fügen Sie vor der Ausführung der Anwendung einen APEX-Benutzer hinzu. Nur im Entwicklungs- und Testarbeitsbereich navigieren Sie zu Shared Components > Application Access Control > Add User Role Assignment; geben Sie den APEX-Benutzer ein und legen Sie für diesen Benutzer die Rollen Administrator, Contributor und Reader fest. In der Produktion ist die Rücksprache mit einem Experten vor dem Einsatz ein Muss.

Wenn dem Benutzer keine entsprechende Rolle zugewiesen wird, stürzt die importierte Anwendung ab. Es ist notwendig, das Webbrowser-Cookie zu löschen (z.B. Firefox: Einstellungen > Cookies und Websitedaten > Daten verwalten), wenn die Anwendung aufgrund von nicht erfüllten Anforderungen abstürzt.

### 14.6.2 Video-Anleitungen

Videoanleitungen zeigen jeden Schritt der Anwendungsentwicklung.

## 14.7 Fragen

1. Untersuchen Sie, wie die Kommunikation mit einer externen Quelle oder einer anderen Datenbank oder einem System, das Pflanzendaten enthält, implementiert werden kann.
2. Nennen Sie einige der Formate, in denen Bilder von Pflanzen und Samen gespeichert sind.
3. In welchem Datentyp werden Bilder, Videos und ähnliche Objekte in der Oracle-Datenbank gespeichert?

## 14.8 Antworten

1. Im Prinzip gibt es zwei Möglichkeiten, die Kommunikation mit einer solchen externen Quelle zu realisieren. Eine Möglichkeit ist das Herunterladen des Datensatzes, d.h. der gesamten Datenbank in den Pflanzenkatalog. Eine andere Möglichkeit ist die Verbindung mit Hilfe einer Anwendungsprogrammierschnittstelle (API). Dabei handelt es sich um eine Reihe von Protokollen, Werkzeugen und Standards, die festlegen, wie Softwarekomponenten miteinander interagieren und kommunizieren sollen. APIs ermöglichen es verschiedenen Softwaresystemen, auf standardisierte Weise miteinander zu kommunizieren und Daten auszutauschen.
2. Einige der gängigsten Formate sind:
   a. JPEG (Joint Photographic Experts Group) - dies ist ein beliebtes Format für digitale Fotos, da es dank seiner Komprimierungsfunktionen kleinere Dateigrößen ohne wesentliche Qualitätsverluste ermöglicht. Es unterstützt Millionen von Farben und kann hochauflösende Bilder verarbeiten.
   b. BMP (Bitmap) - dieses Format wird häufig in Windows-Betriebssystemen verwendet und kann sowohl Graustufen- als auch Farbbilder unterstützen. Es unterstützt keine Komprimierung, so dass die Dateien groß sein können, aber es ist eine gute Option für die Speicherung hochwertiger Bilder.
   c. PNG (Portable Network Graphics) - dieses Format ist für seine verlustfreie Komprimierung bekannt, was bedeutet, dass die Bildqualität nicht beeinträchtigt wird, wenn die Dateigröße reduziert wird. Es unterstützt Transparenz und wird häufig für Webgrafiken und digitale Kunst verwendet.
   d. TIFF (Tagged Image File Format) - dieses Format wird häufig zum Speichern hochwertiger Bilder verwendet, wie sie z.B. beim Drucken verwendet werden. Es unterstützt eine verlustfreie Komprimierung, mehrere Ebenen und kann sowohl Graustufen- als auch Farbbilder verarbeiten.
3. Bilder, Videos und ähnliche Objekte werden in der Regel in der Oracle-Datenbank unter

Verwendung des Datentyps BLOB (Binary Large Object) gespeichert. BLOBs können binäre Daten, wie Bilder und Videos, mit einer Größe von bis zu 4 Gigabyte speichern. Dieser Datentyp ist ideal für die Speicherung großer Objekte wie Multimediadateien, da er eine effiziente Speicherung und Abfrage von Binärdaten ermöglicht.



# 15. GreenDi - Benutzerautorisierung und -verwaltung

VJERAN STRAHONJA, DARKO ANDROČEC, ANA KUTNJAK, LARISA HRUSTEK

## 15.1 Wirtschaftliche Sicht des Falles

Ein kurzer geschäftlicher Überblick über die GreenDi-Plattform wird in Kapitel 14 beschrieben. Er konzentriert sich auf den öffentlichen Teil der GreenDi-Plattform. Erweiterte Funktionalitäten sind nur für registrierte Benutzer (Mitglieder und Administratoren) möglich. Es ist notwendig, die Funktionalität der Benutzerregistrierung, des Logins der registrierten Benutzer in das System mit Authentifizierung und Autorisierung zur Durchführung bestimmter Aktivitäten zu aktivieren. Zu den erforderlichen Funktionen gehören die Sperrung des Benutzers unter bestimmten Bedingungen und die Überwachung der Datenhistorie des registrierten Benutzers.

## 15.2 Problemstellung

In den folgenden Sätzen wird das Problem der Benutzerautorisierung und -verwaltung beschrieben:

- Um einen tieferen Einblick in die Informationen über Pflanzen und andere erweiterte Funktionen zu erhalten, ist es notwendig, sich auf der Plattform anzumelden, d.h. dies ist nur für registrierte Benutzer (Mitglieder) möglich.
- Die Anmeldung ist für alle Funktionalitäten gleich und muss im Prinzip eine einmalige Registrierung sein.
- Der grundlegende Mechanismus zur Benutzerauthentifizierung ist Benutzername und Passwort.
- Die Informationen über das aktuelle Kennwort sind weder für den Administrator noch für andere Benutzer sichtbar.
- Die Registrierung umfasst die Eingabe grundlegender Benutzerinformationen wie Name, Adresse, E-Mail, Telefon, Benutzername und Passwort.
- Optionale Daten sind Nutzerinteressen, Geburtsdatum, etc.
- Nach erfolgreicher Registrierung erhält ein neuer Benutzer automatisch den Standardtyp „Mitglied“ und den Status „Aktiv“. Die Änderung des Status in Administrator und umgekehrt sowie des Typs kann nur von einem anderen aktiven Administrator vorgenommen werden.
- Mit der Anmeldung akzeptiert das Mitglied die Nutzungsbedingungen. Verstößt er oder sie dagegen, setzen die Administratoren die Mitgliedschaft aus, was ebenfalls in den Nutzungsbedingungen beschrieben ist.
- Die regelmäßige Anmeldung und Löschung der Mitgliedschaft ist freiwillig und erfolgt durch den Nutzer.

- Jede Änderung von Benutzerdaten wird in Form eines Protokolls aufgezeichnet. Der Administrator kann die Änderungshistorie jedes Benutzers, mit Ausnahme des Passworts, einsehen.
- Auf Wunsch des Benutzers ermöglicht das System das Zurücksetzen des Kennworts, so dass der Administrator dem Benutzer per E-Mail oder SMS ein vorläufiges Kennwort zusendet.
- Optionale Funktionalität: Zusätzlich zu Benutzername und Passwort ist eine Authentifizierung über Fb und Google Sign-In oder andere Dienste optional möglich. So muss beispielsweise eine Webanwendung, die OAuth 2.0 für den Zugriff auf Google-APIs verwendet, über Autorisierungsdaten verfügen, die die Anwendung gegenüber dem OAuth 2.0-Server von Google identifizieren.

## 15.3 Anwendungsfälle

### 15.3.1 Narrative Beschreibung des Anwendungsfalls

Das Anwendungsfalldiagramm beschreibt die Kommunikation von Akteuren und anderen Systemen mit dem von uns entwickelten System. Anwendungsfälle stellen die Funktionalität der Autorisierung und Benutzerverwaltung dar, die in Figure 15.1 beschrieben wird.

### 15.3.2 Semi-strukturierte Beschreibung

Tabelle 15.1 zeigt den Anwendungsfall.

### 15.3.3 Anwendungsfalldiagramm

Die obige Geschichte ist in dem Anwendungsfalldiagramm 15.1 abgebildet.

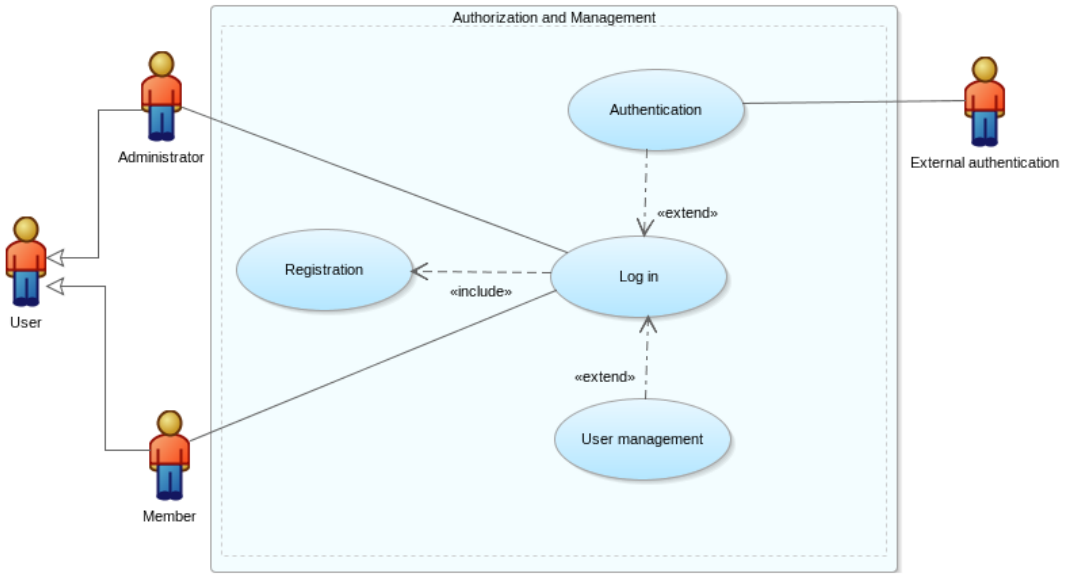



Abbildung 15.1: Anwendungsfalldiagramm - GreenDi Benutzerautorisierung und -verwaltung

## 15.4 Data model

### 15.4.1 Narrative Beschreibung des Datenmodells

Das logische Datenmodell (Entities-Relationship-Modell) enthält 2 Arten von Entitäten: User und User History. User hat die folgenden Attribute: User ID (Identifikationsattribut); Type (Benutzertyp: Mitglied, Administrator); Name (Vor- und Nachname des Benutzers); Address (Adresse des Benutzers); E-Mail; Phone; Date of birth (opt.); Interests (opt.); Status (Active, Suspended, Inactive);

Tabelle 15.1: Beschreibung des Anwendungsfalls: Benutzerautorisierung und -verwaltung.

| **Keyword** | **Wert** |
|---|---|
| ID: | *ch15-01* |
| Titel: | *Benutzerautorisierung und -verwaltung* |
| Beschreibung: | *Autorisierung und Verwaltung von Benutzern.* |
| Primärer Akteur: | *Jede Person, Mitglied, Administrator* |
| Voraussetzungen: | *Die Anmeldung erfolgt in der Regel automatisch aus anderen Anwendungsfällen, in denen der Benutzer als Mitglied oder Administrator eingestellt ist.* |
| Nachbedingungen: | *Das Ergebnis ist, dass der Benutzer nicht berechtigt ist, oder dass er berechtigt ist, aber sein Status (ausstehend, aktiv ...) und sein Typ (Mitglied, Administrator) bekannt sind.* |
| *Haupt* | - |
| Erfolgs-Szenario: | 1. Erweiterte Funktionen der GreenDi-Plattform erfordern eine Anmeldung am System Log In Form des Log In Use case. a. Bei der ersten Anmeldung muss sich der neue Benutzer über das Benutzerregistrierungsformular des Anwendungsfalls „Benutzerregistrierung“ registrieren, das im Anwendungsfall „Anmelden“ enthalten ist. Der Registrierungsprozess umfasst die Eingabe von obligatorischen Benutzerinformationen wie Name, Adresse, E-Mail und Telefon sowie von optionalen Informationen wie Benutzerinteressen und Geburtsdatum. b. Der Benutzer legt den Benutzernamen und das Passwort fest (sie werden in einer speziellen Tabelle in der Datenbank gespeichert, getrennt von anderen Benutzerdaten oder verschlüsselt). c. Der Nutzer akzeptiert die Nutzungsbedingungen und bestätigt die eingegebenen Daten. d. Der Standardbenutzertyp ist Mitglied und der Standardstatus ist Aktiv.<br>2. Die Benutzerdaten können vom Benutzer selbst bei jedem weiteren Einloggen in das System über das Benutzerregistrierungsformular geändert werden.<br>3. Das Zurücksetzen des Passworts wird vom Benutzer über das Anmeldeformular veranlasst, so dass eine Nachricht mit einem temporären Passwort, das 15 Minuten gültig ist, an die E-Mail des Benutzers gesendet wird.<br>4. Jede Änderung von Benutzerdaten wird in Form eines Protokolls in der Dateneinheit „Benutzerhistorie“ aufgezeichnet. Der Administrator kann die Änderungshistorie jedes Benutzers überprüfen, mit Ausnahme des Passworts.<br>5. Die Funktionalität des Anwendungsfalls „Benutzerverwaltung“ ermöglicht es einem Benutzer des Typs „Administrator“ und im Status „Aktiv“, den Typ und den Status eines anderen Benutzers zu ändern (Administrator zu Mitglied und umgekehrt, sowie den Status „Aktiv - Suspendiert - Inaktiv“).<br>6. Es gibt eine tabellarische Ansicht der Benutzer mit der Möglichkeit der Suche (Formular Tabelle der Benutzer). In diesem Formular kann der Administrator auf einem ähnlichen Formular wie dem Formular für die Benutzerregistrierung (Formular für Benutzerdetails) die Details eines beliebigen Benutzers auswählen, den Status und den Typ des Benutzers ändern, das Kennwort zurücksetzen und das Formular für die Benutzerhistorie öffnen, das die Änderungen der Daten über den ausgewählten Benutzer in Tabellenform anzeigt.<br>7. Jedes weitere Einloggen des registrierten Benutzers löst eine Überprüfung der Benutzerberechtigung aus, die den Benutzernamen und das Passwort, den Status des Benutzers und ob es sich um einen Administrator oder ein Mitglied handelt, überprüft. |
| Erweiterungen: | *Anmeldeformular; Benutzerregistrierungsformular; Formular für die Benutzertabelle; Formular für Benutzerdetails; Formular für die Benutzerhistorie* |
| Häufigkeit der Nut | *Ungefähres Maximum sind 100 pro Tag* |

User name (für den Administrator nicht sichtbar, verschlüsselt); Password (für den Administrator nicht sichtbar, verschlüsselt) Jeder Benutzer hat 1 oder mehrere User History Einträge . Die User History hat die gleichen Attribute wie User, sie ist eine Kopie des Benutzerdatensatzes nach der Änderung, mit der Tatsache, dass sie eine vom System generierte Änderungszeit und die ID der Person enthält, die die Änderung vorgenommen hat (das Mitglied selbst oder der Administrator). Das bedeutet, dass die geänderten Daten nach jeder Änderung in diesem Protokoll festgehalten werden.

### 15.4.2 Logisches Datenmodell

Das logische Datenmodell ist in Figure 15.2 dargestellt.

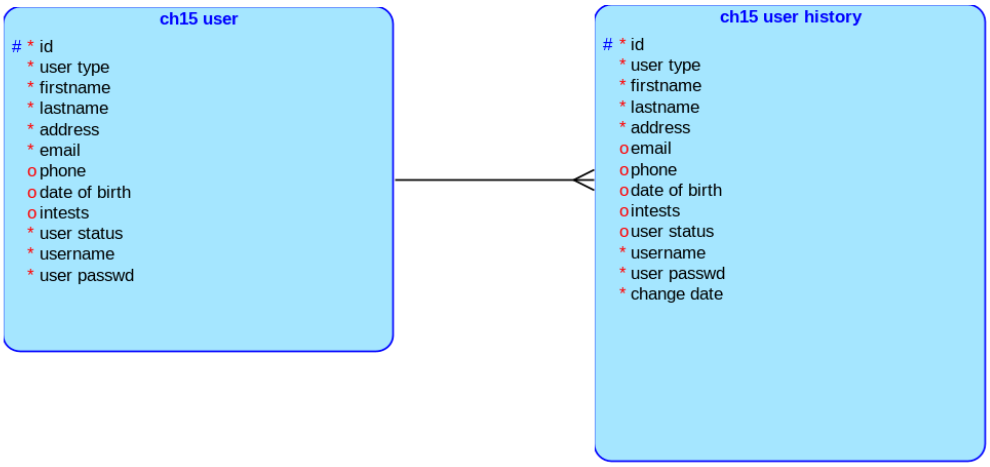



Abbildung 15.2: Logisches Datenmodell.

### 15.4.3 Relationales Datenmodell

Die automatische Umwandlung vom logischen Datenmodell in das relationale Datenmodell in Oracle SQL Data Modeler wird durch die Funktion *Engineering to relational* bereitgestellt. Das Ergebnis ist in Figure 15.3 dargestellt.

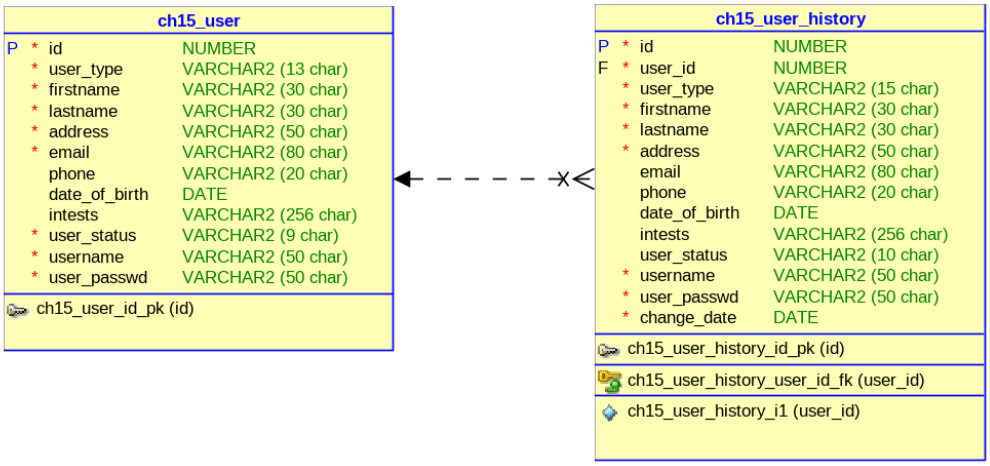



Abbildung 15.3: Relationales Datenmodell.

Lassen Sie nun Oracle SQL Data Modeler ein SQL-Skript generieren. Wählen Sie alle Tabellen im relationalen Modell aus und verwenden Sie die Funktion *File* > *Export* > *DDL File*, um ein Skript zu erhalten. Wir können das generierte Skript in APEX importieren und es ausführen.

## 15.5 Anwendungsschnittstellen

Das Benutzerformular ist in Figure 15.4 dargestellt.

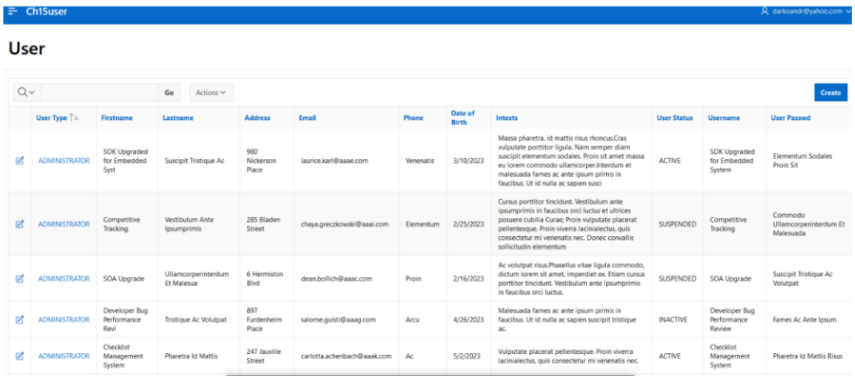



Abbildung 15.4: Benutzerformular.

Daten in das Formular Benutzerhistorie einfügen wird in Figure 15.5 dargestellt.

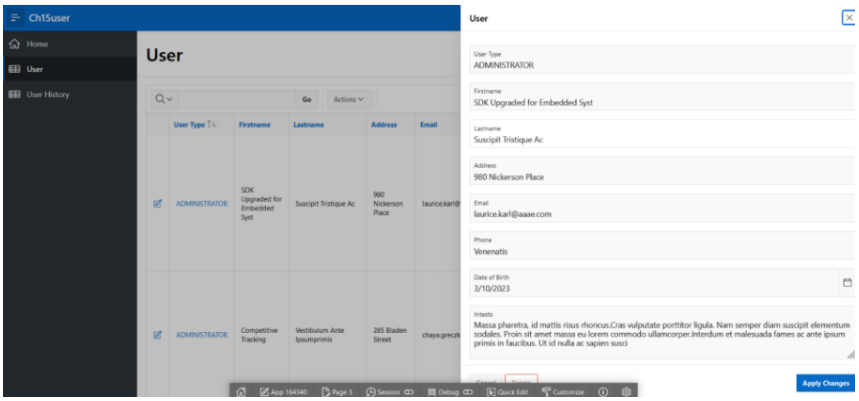



Abbildung 15.5: Formular für die Benutzerhistorie.

Funktionalitäten, die eine Benutzerautorisierung erfordern, sind im Kapitel 18 Webshop - GreenDi beschrieben.

## 15.6 Ergänzendes Lernmaterial

Sie können das folgende zusätzliche Lernmaterial finden:

- exportierte Anwendung
- Skripte zum Erstellen, Löschen und Einfügen
- Video-Anleitung

Alle zusätzlichen Lernmaterialien sind auf der öffentlichen BeeAPEX-Projektseite verfügbar. Melden Sie sich als Gastbenutzer an (kein Passwort erforderlich). Das Lehrbuch finden Sie im Abschnitt Books, die Skripte im Ordner Part 2 > Chapter15 im Abschnitt Scripts und die Videoanleitungen in der Sammlung der Videoanleitungen. Material für Kurzkurse finden Sie in der Sektion Short courses.

### 15.6.1 Exportierte Anwendung

Die exportierte Anwendung ist verpackt. Bei der Installation werden sowohl Tabellen erstellt als auch Daten eingefügt. Bei der Deinstallation werden alle in dieser Anwendung verwendeten Datenbankobjekte entfernt.

### 15.6.2 Video-Anleitungen

Videoanleitungen zeigen jeden Schritt der Anwendungsentwicklung.

## 15.7 Fragen

1. Was sind die eingebauten und anwendungsbezogenen Benutzerauthentifizierungs- und Autorisierungsfunktionen bei der Entwicklung von Anwendungen mit Oracle APEX?
2. Wie können wir externe Authentifizierung in APEX verwenden?
3. Welche Vorteile ergeben sich für die Nutzer, wenn sie sich mit ihren Facebook-, Google- oder ähnlichen Konten bei einer Anwendung anmelden?

## 15.8 Antworten

1. a) Eingebaute Benutzerauthentifizierung: APEX bietet eine integrierte Benutzerauthentifizierungsfunktion, die es Ihnen ermöglicht, Benutzerkonten direkt in der APEX-Umgebung zu definieren und zu verwalten. Benutzer können unterschiedliche Rollen und Rechte haben, und Sie können den Zugriff auf bestimmte Anwendungsseiten oder Komponenten auf der

Grundlage von Benutzerrollen steuern. b) Autorisierung auf Anwendungsebene: Mit APEX können Sie fein abgestufte Autorisierungsregeln auf der Anwendungsebene definieren. Sie können festlegen, welche Benutzer oder Rollen Zugriff auf bestimmte Seiten, Regionen, Schaltflächen oder andere Komponenten innerhalb Ihrer Anwendung haben.

2. Sie können Anmeldeoptionen wie Facebook, Google und andere Plattformen in einer Anwendung oder Website verwenden, die Sie selbst entwickeln. Viele beliebte Plattformen stellen Entwickler-APIs (Application Programming Interfaces) zur Verfügung, mit denen Sie ihre Anmeldefunktionen in Ihre eigenen Anwendungen oder Websites integrieren können. Facebook stellt beispielsweise die Facebook Login API zur Verfügung, mit der sich Benutzer mit ihren Facebook-Anmeldedaten bei Ihrer Anwendung oder Website anmelden können. In ähnlicher Weise stellt Google die Google Sign-In API zur Verfügung, die es Nutzern ermöglicht, sich mit ihren Google-Konten anzumelden. Um diese Anmeldeoptionen zu implementieren, müssen Sie Ihre Anwendung oder Website in der Regel beim Entwicklerportal der jeweiligen Plattform registrieren. Dieser Registrierungsprozess beinhaltet in der Regel den Erhalt von API-Schlüsseln oder Client-IDs, die Sie dann in Ihrem Code zur Authentifizierung von Benutzern und zur Verwaltung des Anmeldevorgangs verwenden werden.
3. Durch die Integration von Anmeldeoptionen dieser Plattformen können Sie Ihren Nutzern eine bequeme und vertraute Möglichkeit bieten, sich bei Ihrer Anwendung oder Website anzumelden, ohne dass sie neue Konten erstellen oder sich zusätzliche Anmeldedaten merken müssen. Es ist nicht erforderlich, sich zu registrieren und Namen, Adresse und ähnliche Daten einzugeben, es werden keine neuen Benutzernamen und Passwörter erstellt, die man sich merken muss.



# 16. Kleines Innovationssystem

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA; ÜBERSETZER W. SCHWINGER

## 16.1 Wirtschaftliche Sicht des Falles

In einem modernen Unternehmen sind innovative Ideen der Schlüssel zum Erfolg und zur Verbesserung der Unternehmensleistung. Aufgrund ihrer Kenntnisse über die Geschäftsprozesse sind die Mitarbeiter wahrscheinlich die beste Quelle für innovative Ideen. Diese Ideen umfassen verschiedene Gesichtspunkte wie Vorschläge zur Prozessoptimierung, Verbesserungen des Kundenbeziehungsmanagements, Verbesserung der Qualitätssicherungsaufgaben usw. Daher beschloss die Unternehmensleitung in diesem speziellen Fall, kleine innovative Ideen der Mitarbeiter zu sammeln und sie zu belohnen.

## 16.2 Problemstellung

Innovative Ideen im Unternehmen wurden früher per E-Mail und auf Papierformularen gesammelt. Eine Übersicht über die Ideen im Unternehmen war zeitaufwendig. Die Bewertungskriterien für die Beurteilung von Ideen waren nicht definiert. Aufgrund der großen Anzahl potenzieller Ideen, die für das Unternehmen von Nutzen sein könnten, ist es notwendig, einen guten Überblick zu haben. Die Plattform wird die Sammlung innovativer Ideen in einem kleinen Unternehmen ermöglichen. Die Mitarbeiter können sich bei der Plattform anmelden und ihre Ideen mit einfachen Beschreibungen oder Anhängen eingeben. Da die Ideen nicht den gleichen Wert haben (für das Management und den Betrieb), werden sie überprüft und bewertet. Wenn die Bewerter den Prozess abgeschlossen haben, wird die Idee als angenommen, abgelehnt oder überarbeitungsbedürftig eingestuft. Die innovativsten Ideen werden je nach ihrer Bewertung prämiert. Die folgenden Auszeichnungen können für die vorgeschlagene Idee vergeben werden: GOLD (100 EUR), SILBER (80 EUR), BRONZE (60 EUR), BETEILIGTER (40 EUR) oder DANKE. Nur Manager haben die Möglichkeit, Ideen zu bewerten. Die Manager werden die Bewertung und Belohnung regelmäßig (d.h. einmal pro Monat) veranlassen. Alle angemeldeten Benutzer können die Ideen nach der Bewertung überprüfen.

## 16.3 Anwendungsfälle

### 16.3.1 Narrative Beschreibung des Anwendungsfalls

Jeder Mitarbeiter hat die Möglichkeit, über das Antragsformular Innovationsideen vorzuschlagen. Alle Ideen werden gesammelt und von der Geschäftsleitung bewertet. Die Ideen werden nach

Nützlichkeit und Realisierungsmöglichkeiten geordnet und gute Ideen werden prämiert.

### 16.3.2 Semi-strukturierte Beschreibung

Wir können drei verschiedene Benutzergeschichten oder Anwendungsfälle zusammenfassen: Ideeneinreichung, Überblick über die Ideen und Ideenbewertung. Der letzte Schritt - die Auszahlung der Prämie - wird in diesem Geschäftsfall nicht behandelt.

Das Hinzufügen eines Anhangs zur Idee ist eigentlich eine Erweiterung der oben beschriebenen UC und wir könnten es als neuen Anwendungsfall beschreiben. Aus Gründen der Lesbarkeit fahren wir mit den anderen beiden Hauptanwendungsfällen fort, allerdings wird die Anwendung über eine Funktion zum Hinzufügen von Anhängen zur Idee verfügen.

### 16.3.3 Anwendungsfalldiagramm

Die obige Geschichte ist im Anwendungsfalldiagramm dargestellt (siehe Figure 16.1).

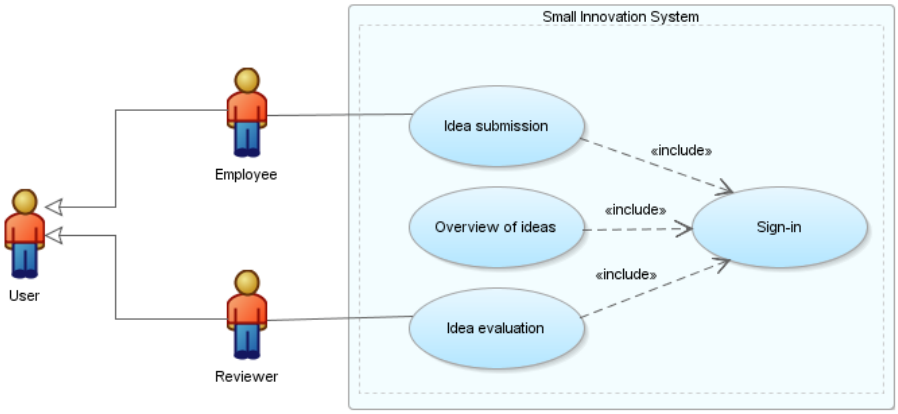



Abbildung 16.1: Anwendungsfalldiagramm.

Auch hier wird der Einfachheit halber der Anwendungsfall der Erweiterung (Hinzufügen eines Anhangs zur Idee) auf Figure 16.1 übersprungen.

## 16.4 Datenmodell

### 16.4.1 Narrative Beschreibung des Datenmodells

Es gibt drei Entitäten im logischen Datenmodell. Die Entität CH16 IDEA hat acht Attribute: ID, Einreichungsdatum, Titel, Beschreibung der Idee, Dateiname der Idee, Mime-Typ, Entscheidung und Entscheidungsdatum. Jede Idee wird von einem Mitarbeiter vorgeschlagen. Grundlegende Daten über Mitarbeiter werden in der Entität CH16 EMPLOYEE gespeichert: ID, Vor- und Nachname. Ein Mitarbeiter kann ein Manager für andere Mitarbeiter sein. Jeder Mitarbeiter hat nur einen Vorgesetzten, während ein Vorgesetzter mehrere Mitarbeiter führen kann. Manager bewerten Ideen. Ein Manager kann viele Ideen bewerten, während eine Idee nur von einem Manager bewertet werden kann. Jede Idee kann eine Auszeichnung erhalten. Die Entität CH16 AWARD hat drei Attribute: ID, Beschreibung des Preises und Betrag in Euro. Eine Art von Auszeichnung kann für mehrere Ideen vergeben werden. Die Bewertungsregeln werden nicht in das Datenmodell aufgenommen, da dies das Risiko mit sich bringen würde, dass die Anwendung bei jeder Änderung der Regeln geändert werden müsste. Mit dem vorgeschlagenen Datenmodell können wir Folgendes gewährleisten: Robustheit der Anwendung, schnelle Entwicklung und Bereitstellung, geringe Entwicklungs- und Wartungskosten und Flexibilität.

Tabelle 16.1: Beschreibung des Anwendungsfalls: Einreichung von Ideen.

| **Keyword** | **Wert** |
|---|---|
| ID: | *Ch16-01* |
| Titel: | *Einreichung von Ideen* |
| Beschreibung: | *Ein Mitarbeiter meldet sich bei der Plattform an und gibt die Idee ein. Jede Idee hat einen Titel und eine kurze Beschreibung, die als Anhang (Dokumentdatei) hinzugefügt wird.* |
| Primärer Akteur: | *Mitarbeiter* |
| Voraussetzungen: | *Der Mitarbeiter muss über das Oracle APEX-Benutzerkonto für die Anwendung Small Innovation System verfügen. Der Zugriff auf die Webanwendung muss aktiviert sein.* |
| Nachbedingungen: | *Nach erfolgreicher Eingabe der Idee sind die Idee und die ausführliche Beschreibung im Anhang für andere SIS-Portalnutzer verfügbar.* |
| *Haupt* | *Szenarien* |
| Erfolgs-Szenario: | 1. *Öffnen Sie den Webbrowser und melden Sie sich bei der SIS-Anwendung an.*<br>2. *Wählen Sie Ideen im Menü*<br>3. *Wählen Sie Neue Idee hinzufügen*<br>4. *Geben Sie den Titel der Idee ein und fügen Sie einen Anhang hinzu*<br>5. *Geben Sie den Dateinamen und die Art des Anhangs ein*<br>6. *Bestätigen Sie das Einfügen, indem Sie auf die Schaltfläche Create klicken.*<br>7. *Überprüfen Sie die eingegebenen Daten* |
| Erweiterungen: | • *1a. Anmeldung schlägt fehl*<br>• *1a* Erweitern:*<br>• *1a1. Fehlermeldung anzeigen*<br>• *1a2. Anmeldefenster öffnen*<br>• *4a. Datentyp-Fehler*<br>• *4a* Erweitern:*<br>• *4a1. Fehlermeldung anzeigen*<br>• *7a. Datentyp-Fehler*<br>• *7a* Erweitern:*<br>• *7a1. Fehlermeldung anzeigen* |
| Häufigkeit der Nutzung: | *Im Durchschnitt erhält das Unternehmen 3 innovative Ideen pro Woche.* |
| Status: | *Fertiggestellt* |
| Eigentümer: | *Mitarbeiter mit Zugriff auf die Anwendung* |
| Priorität: | *moderat* |

Tabelle 16.2: Beschreibung des Anwendungsfalls: Überblick über die Ideen.

| **Keyword** | **Wert** |
|---|---|
| ID: | *Ch16-02* |
| Titel: | *Überblick über die Ideen* |
| Beschreibung: | *Ein Mitarbeiter meldet sich bei der Plattform an und überprüft den Status der Idee oder anderer Ideen.* |
| Primärer Akteur: | *Mitarbeiter* |
| Voraussetzungen: | *Der Mitarbeiter muss über das Oracle APEX-Benutzerkonto für die Anwendung Small Innovation System verfügen. Der Zugriff auf die Webanwendung muss aktiviert sein.* |
| Nachbedingungen: | |
| *Haupt* | *Szenarien* |
| Erfolgs-Szenario: | 1. *Öffnen Sie den Webbrowser und melden Sie sich bei der SIS-Anwendung an*<br>2. *Wählen Sie Ideen im Menü*<br>3. *Durchsuchen Sie die aufgelisteten Ideen und ihren Status*<br>4. *Wählen Sie Übersicht im Menü*<br>5. *Sehen Sie sich die Diagramme zu den Daten der Ideeneinreichung an* |
| Erweiterungen: | • *1a. Anmeldung schlägt fehl*<br>• *1a* Erweitern:*<br>• *1a1. Fehlermeldung anzeigen* |
| Häufigkeit der Nutzung: | *Die Möglichkeit, Ideen einzusehen, haben Mitarbeiter, Prüfer und Administratoren. Insgesamt rufen sie die Plattform etwa 5 Mal pro Tag auf.* |
| Status: | *Fertiggestellt* |
| Owner: | *Mitarbeiter mit Anmeldemöglichkeit* |
| Priority: | *niedrig* |

Tabelle 16.3: Beschreibung des Anwendungsfalls: Bewertung von Ideen.

| **Keyword** | **Wert** |
|---|---|
| ID: | *Ch16-03* |
| Titel: | *Bewertung von Ideen* |
| Beschreibung: | *Ein Prüfer meldet sich bei der Plattform an und bewertet die Idee. In diesem Anwendungsfall wird auch der Preis definiert.* |
| Primärer Akteur: | *Prüfer* |
| Voraussetzungen: | *Der Benutzer des Mitarbeiters muss der Rolle CH16_Reviewer hinzugefügt werden. Der Zugriff auf die Webanwendung muss aktiviert werden.* |
| Nachbedingungen: | *Die ausgewählte Idee wurde bewertet und der vorgeschlagene Preis wurde festgelegt.* |
| *Haupt* | *Szenarien* |
| Erfolgs-Szenario: | 1. *Öffnen Sie den Webbrowser und melden Sie sich bei der SIS-Anwendung an.*<br>2. *Wählen Sie Ideen prüfen im Menü*<br>3. *Identifizieren Sie die Idee und fügen Sie Details hinzu, indem Sie die Schaltfläche Edit verwenden.*<br>4. *Wählen Sie eine Entscheidung aus der Liste der Werte (Annehmen, Ablehnen, Überarbeiten)*<br>5. *Wählen Sie die Art der Auszeichnung aus der Liste der Werte aus (Gold, Silber, Bronze, Mitwirkender, Dankeschön)*<br>6. *Bestätigen Sie die Aktualisierung* |
| Erweiterungen: | • *1a. Anmeldung schlägt fehl*<br>• *1a* Erweitern:*<br>• *1a1. Fehlermeldung anzeigen*<br>• *1a2. Anmeldefenster öffnen*<br>• *2a. Datentyp-Fehler*<br>• *2a* Erweitern:*<br>• *2a1. Fehlermeldung anzeigen*<br>• *3a. Datentyp-Fehler*<br>• *3a* Erweitern:*<br>• *3a1. Fehlermeldung anzeigen*<br>• *4a. Datentyp-Fehler*<br>• *4a* Erweitern:*<br>• *4a1. Fehlermeldung anzeigen*<br>• *5a. Datentyp-Fehler*<br>• *5a* Erweitern:*<br>• *5a1. Datentyp-Fehler* |
| Häufigkeit der Nutzung: | *Die Manager prüfen die vorgeschlagenen Ideen einmal im Monat.* |
| Status: | *Fertiggestellt* |
| Eigentümer: | *Prüfer* |
| Priorität: | *hoch* |

### 16.4.2 Logisches Datenmodell

Die obige Geschichte wird im logischen Datenmodell dargestellt (siehe Figure 16.2).

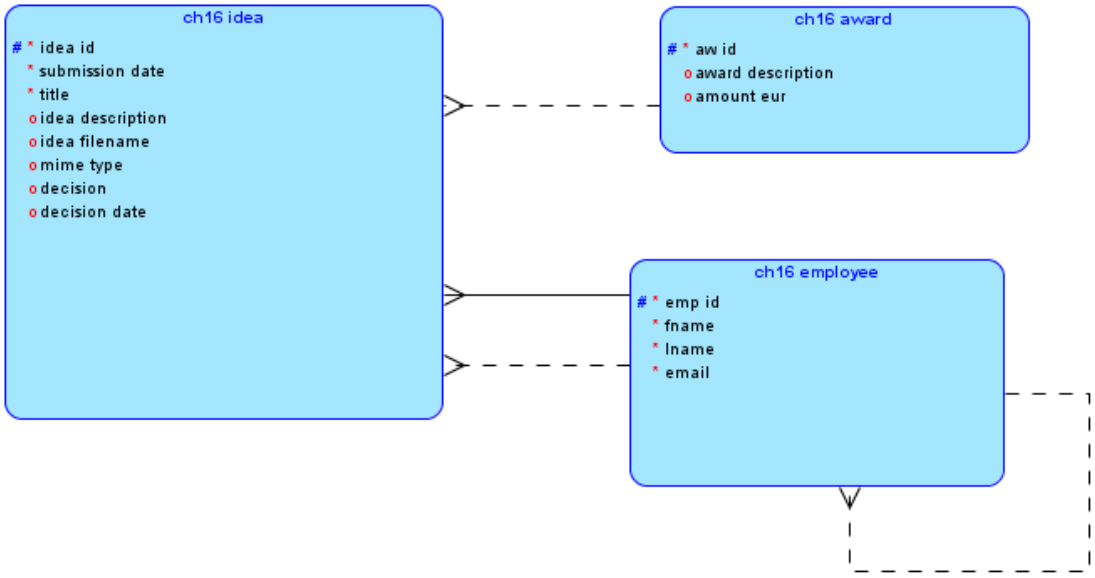



Abbildung 16.2: Logisches Datenmodell.

### 16.4.3 Relationales Datenmodell

Die automatische Umwandlung vom logischen Datenmodell in das relationale Datenmodell in Oracle SQL Data Modeler wird durch die Funktion *Engineering to relational* bereitgestellt. Das Ergebnis wird in Figure 16.3 angezeigt.

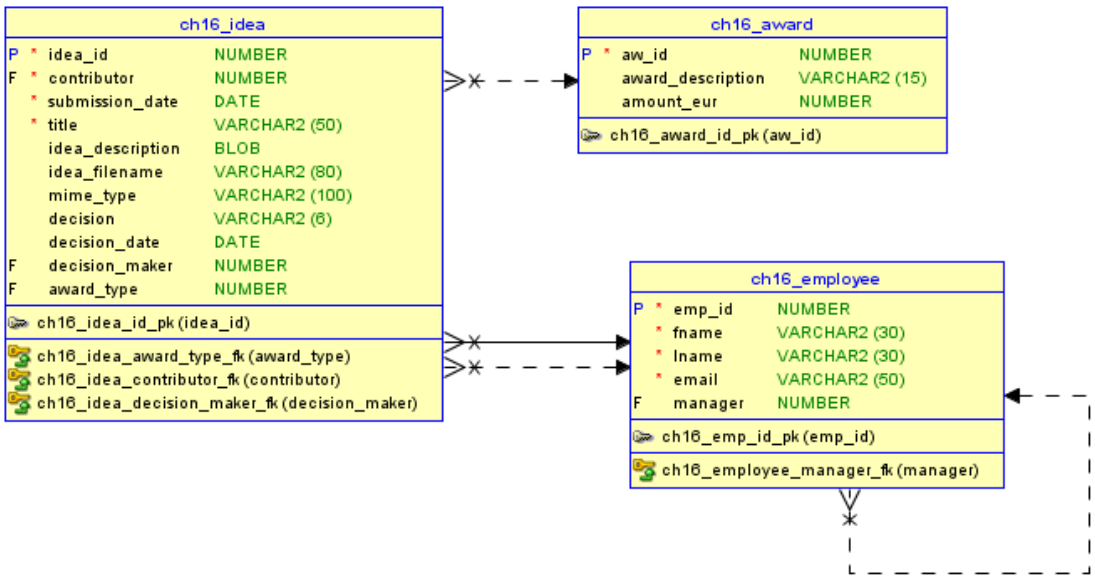



Abbildung 16.3: Relationales Datenmodell.

Der Oracle SQL Data Modeler generiert auch SQL-Skripte für die Erstellung von Tabellen, Sequenzen und Triggern. Wählen Sie alle Tabellen im relationalen Modell aus und verwenden Sie

die Funktion *File* > *Export* > *DDL File*, um ein Skript wie dieses zu erhalten:

```
CREATE TABLE CH16_AWARD (
AW_ID NUMBER NOT NULL
...
);

CREATE TABLE CH16_EMPLOYEE (
EMP_ID NUMBER NOT NULL
...
);
```

Jetzt ist es an der Zeit, Tabellen in der Oracle-Datenbank zu erzeugen. Wir können das generierte Skript importieren und es in APEX ausführen.

## 16.5 Anwendungsschnittstellen

In den folgenden Abbildungen werden die Anwendungsschnittstellen für beide Basisrollen (Mitarbeiter und Prüfer) dargestellt. Zunächst wird das Dashboard für Mitarbeiter in Figure 16.4 dargestellt.

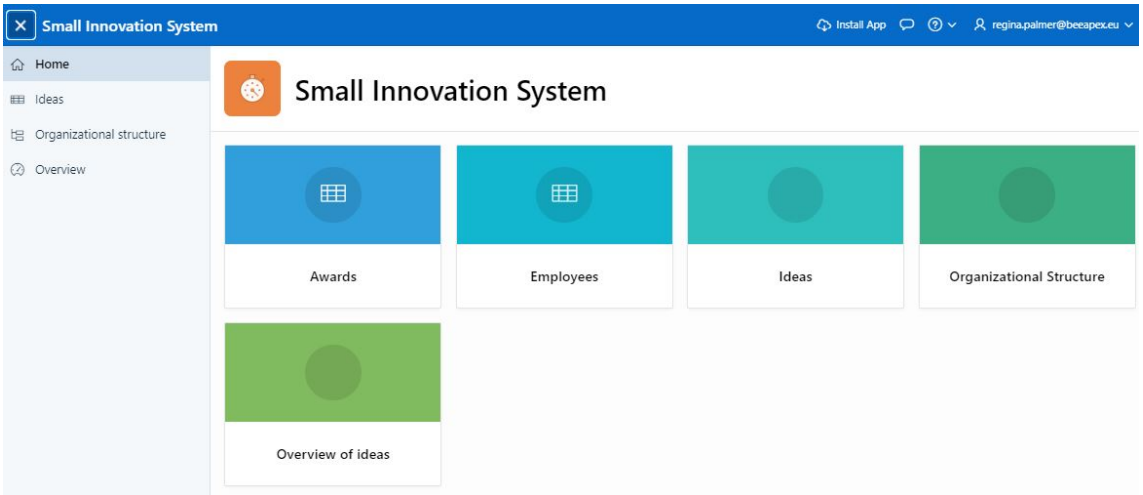



Abbildung 16.4: Das Mitarbeiter-Dashboard.

Die Seite für die Einreichung von Ideen mit Anhängen und ihre Einzelheiten sind in Figure 16.5 zu finden.

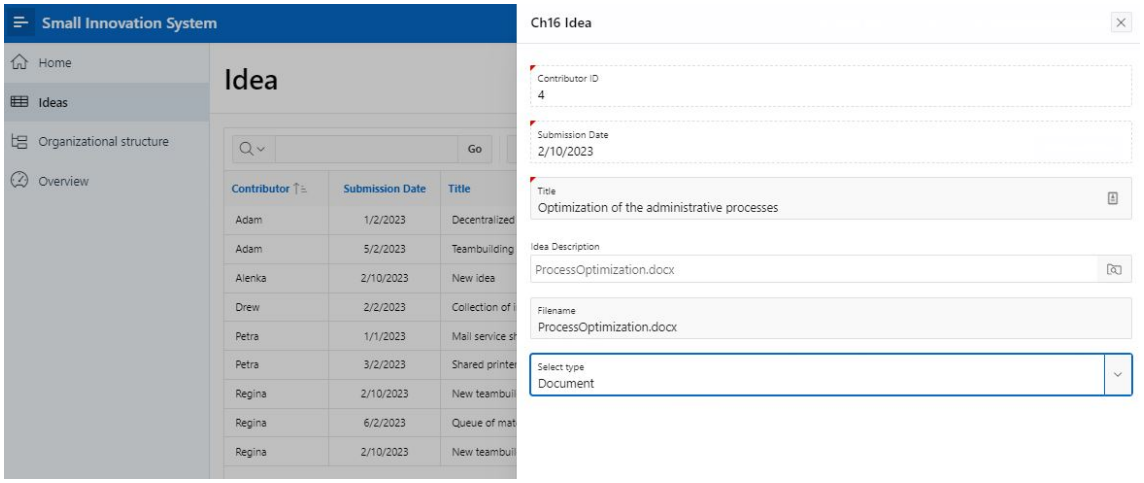



Abbildung 16.5: Idee mit Anhängen einreichen.

Jeder Mitarbeiter hat die Möglichkeit, die Organisationsstruktur in seinem Unternehmen einzusehen, wie sie in Figure 16.6 dargestellt ist.

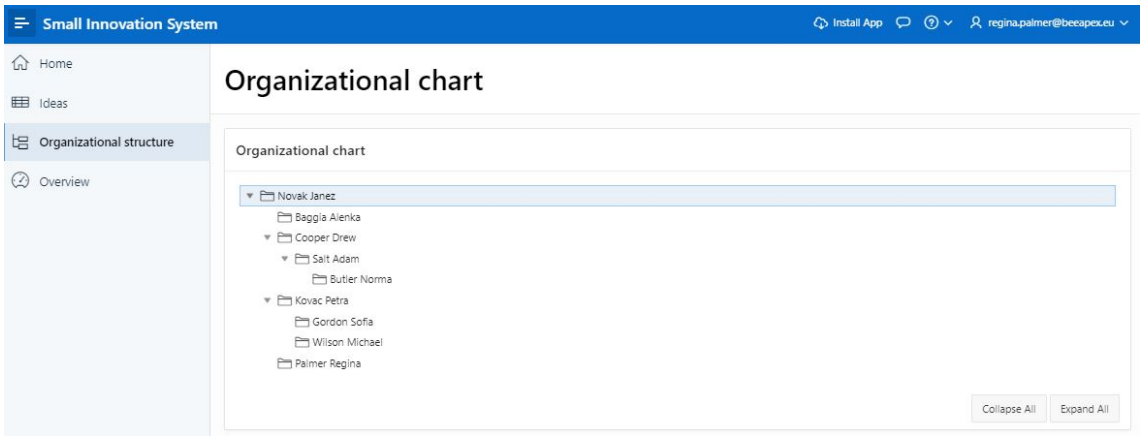



Abbildung 16.6: Organisatorische Struktur im Unternehmen.

Außerdem kann jeder Mitarbeiter das Dashboard mit den Diagrammen sehen, die die Verteilung der Ideen über die Zeit, den Anteil der angenommenen, abgelehnten und überarbeiteten Ideen und die Mitarbeiter mit der höchsten Anzahl innovativer Ideen im Unternehmen darstellen (siehe Figure 16.7).

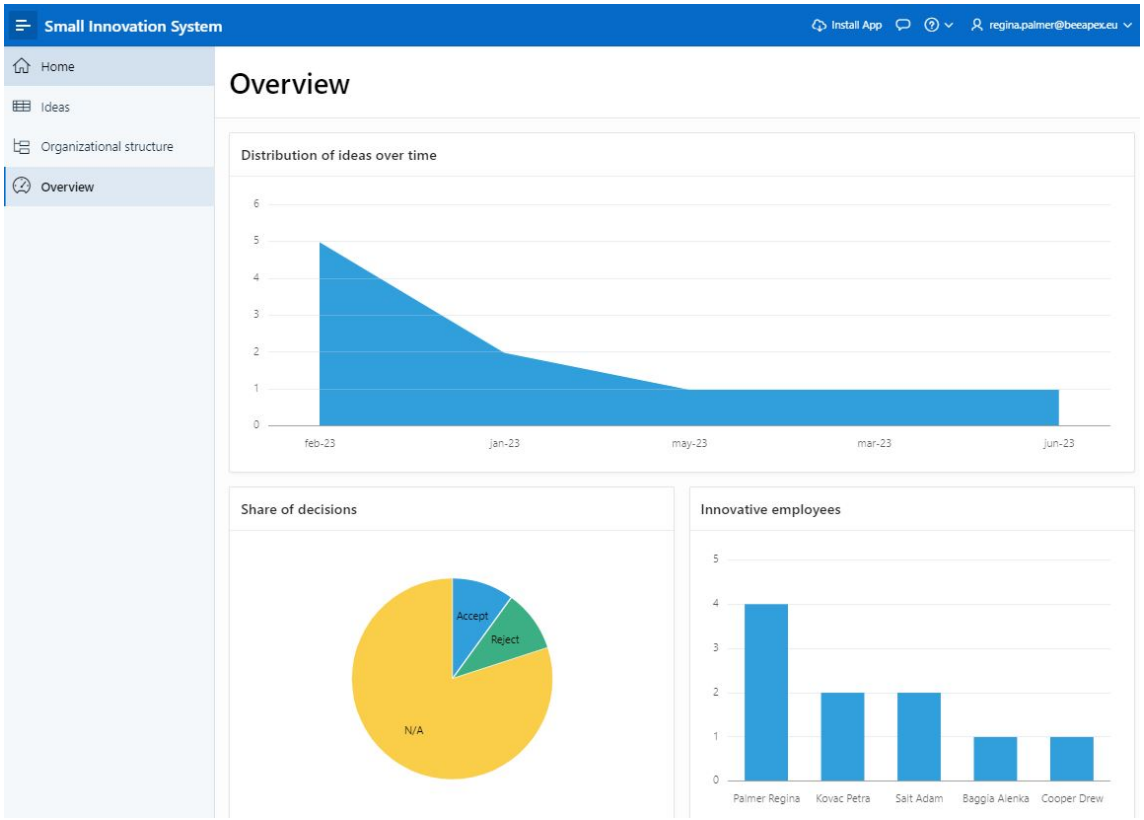



Abbildung 16.7: Kleine Übersicht über Innovationsideen.

Zusätzlich zu den beschriebenen Aktionen hat der Prüfer die Möglichkeit, Daten über Prämien und Mitarbeiter zu aktualisieren, wie in Figure 16.8 dargestellt. Die Verwaltungsseite ist nur für Administratoren, nicht für Prüfer zugänglich.

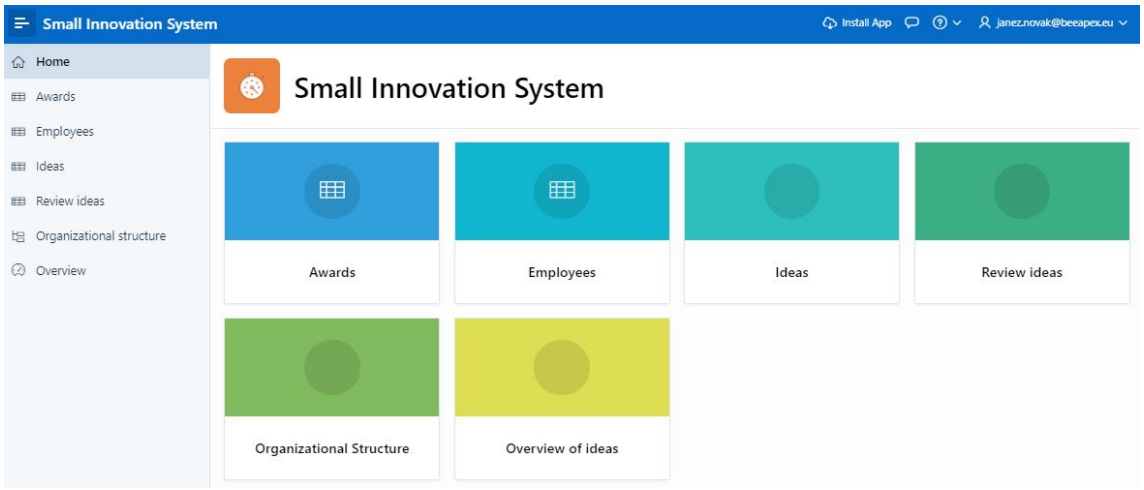



Abbildung 16.8: Prüfer Dashboard.

Die Seite zum Prüfen der Idee ist in Figure 16.9 zu finden. Diese Seite ist nur für den Prüfer zugänglich. Der Prüfer wählt die Entscheidung und ggf. die Art des Preises aus.

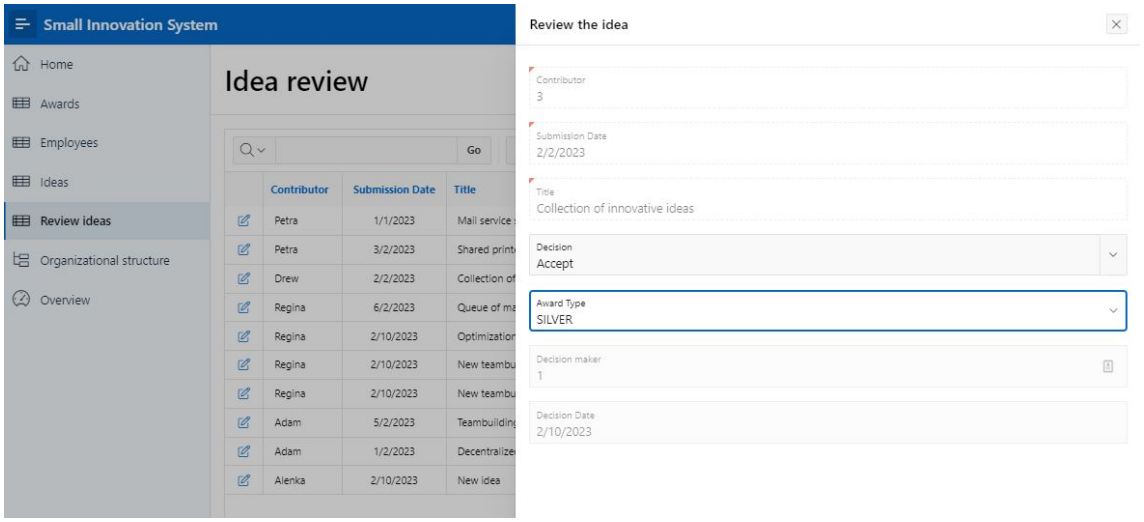



Abbildung 16.9: Seite zum Prüfen der Ideen.

## 16.6 Ergänzendes Lernmaterial

Sie können das folgende zusätzliche Lernmaterial finden:

- Skript zur Erstellung
- Skript zum Auffüllen von Tabellen
- Skript zum Löschen von Tabellen
- exportierte verpackte Anwendung
- Video, das zeigt, wie man eine Anwendung aus einem Skript erstellt

Alle zusätzlichen Lernmaterialien sind auf der öffentlichen BeeAPEX-Projektseite verfügbar. Melden Sie sich als Gastbenutzer an (kein Passwort erforderlich). Das Lehrbuch finden Sie im Abschnitt Books, die Skripte im Ordner Part 1 > Chapter12 im Abschnitt Scripts und die Videoanleitungen in der Sammlung der Videoanleitungen. Material für Kurzkurse finden Sie in der Sektion Short courses.

### 16.6.1 Exportierte Anwendungen

Die exportierte Anwendung wird verpackt. Bei der Installation werden sowohl Tabellen erstellt als auch Daten eingefügt. Bei der Deinstallation werden alle in dieser Anwendung verwendeten Datenbankobjekte entfernt.

Die verpackte Anwendung wird getestet und im neuen Arbeitsbereich ausgeführt, wenn die folgenden Anforderungen erfüllt sind:

- APEX-Benutzer vor der Ausführung der Anwendung hinzufügen.

Wenn dem Benutzer keine entsprechende Rolle zugewiesen wird, stürzt die importierte Anwendung ab. Es ist notwendig, das Webbrowser-Cookie zu löschen (z.B. Firefox: Einstellungen > Cookies und Websitedaten > Daten verwalten), wenn die Anwendung aufgrund von nicht erfüllten Anforderungen abstürzt.

### 16.6.2 Video-Anleitungen

Videoanleitungen zeigen jeden Schritt der Anwendungsentwicklung.

## 16.7 Fragen

1. Wo sind die Benutzerrollen in Kapitel 16 definiert?

2. Welche Beziehung im logischen Datenmodell ermöglicht die Darstellung der Organisationsstruktur in Kapitel 16?
3. Warum sind einige Felder auf der Seite für die Ideenprüfung in Kapitel 16 deaktiviert?

## 16.8 Antworten

1. Im Datenmodell des Kapitels 16 sind keine Rollen definiert. Sie werden nur in der APEX-Umgebung verwendet.
2. Die Beziehung, die die Tabelle Mitarbeiter mit sich selbst verbindet (rekursive Beziehung), ermöglicht die Darstellung der Organisationsstruktur in Kapitel 16.
3. In Kapitel 16 sind einige Felder deaktiviert, um die Benutzerfreundlichkeit zu erhöhen und die Konsistenz der Daten zu gewährleisten, da der Benutzer keine Daten hinzufügen muss, die bereits in der Datenbank vorhanden sind.



# 17. Management von Geschäftsprozessen

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA; ÜBERSETZER W. SCHWINGER

## 17.1 Wirtschaftliche Sicht des Falles

Der größte Teil dieses Buches betont die Bedeutung von Daten und datengesteuerten Ansätzen für die Entwicklung von Webanwendungen mit APEX. Aber der datengetriebene Ansatz hat ein Geschwisterchen: prozessgetriebene Anwendungen. Wie der Name schon sagt, konzentrieren sich prozessgesteuerte Anwendungen auf Prozesse. Weder kann eine gute datengetriebene Anwendung den Prozessaspekt vernachlässigen noch kann eine prozessgetriebene Anwendung den Datenaspekt vernachlässigen. Die implizite oder explizite Einbeziehung beider Aspekte macht jede Anwendung für ihren Zweck geeignet. Für Entwickler ist es wichtig, einen tiefen Einblick in die Geschäftsprozesse einer Organisation zu haben. Jede Art von Organisation, unabhängig von Größe, Jahreseinkommen, Eigentümerschaft, Branche und Aufgabe, führt Geschäftsprozesse durch. Die gängigen Klassen von Geschäftsprozessen, die für die meisten Organisationen gelten, fallen in die folgenden Kategorien:

- Vertrieb und Marketing umfassen Teilprozesse wie die Gewinnung und Bindung von Kunden, die Generierung von Interessenten, das Kundenbeziehungsmanagement und Marketingkampagnen.
- Das Lieferkettenmanagement umfasst Teilprozesse wie die Koordinierung und das Management von Waren- und Dienstleistungsströmen, Beschaffung, Produktion und Vertrieb.
- Das Finanzmanagement umfasst Teilprozesse wie die Verwaltung der finanziellen Ressourcen einer Organisation, Buchhaltung, Budgetierung und Finanzberichterstattung.
- Das Personalmanagement umfasst Teilprozesse wie die Verwaltung der Arbeitskräfte des Unternehmens, die Einstellung, die Ausbildung und die Entwicklung der Mitarbeiter.
- Der Kundendienst umfasst Teilprozesse wie die Unterstützung und Betreuung von Kunden, die Lösung von Beschwerden und die Behebung von Problemen.
- Das Betriebsmanagement umfasst Teilprozesse wie das Management des täglichen Geschäftsbetriebs, der Produktion, der Logistik und der Bestandsverwaltung.
- Die Informationstechnologie umfasst Teilprozesse wie die Verwaltung der informationstechnischen Ressourcen und Systeme einer Organisation, Hardware, Software und Datenmanagement.

Diese Prozesse werden als kritisch angesehen, da sie die Kernfunktionen einer Organisation unterstützen und das Erreichen ihrer Ziele und Vorgaben ermöglichen. Die Verbesserung der Effizienz und Effektivität dieser Prozesse kann einen erheblichen Einfluss auf die Gesamtleistung und den

Erfolg einer Organisation haben. In einem Versuch, ein klares Verständnis und eine Definition typischer Geschäftsprozesse zu schaffen, hat OASIS OPEN die Universal Business Language entwickelt, die Prozesse und Geschäftsdokumente beschreibt (siehe [5]). Die Softwareindustrie begann in den späten 1990er Jahren mit der Integration von Prozessen und erfand die Business Process Execution Language (BPEL) und später die Business Process Modeling Language (BPML). Große Unternehmen wie Microsoft, IBM, SAP, Oracle sowie unzählige kleinere Unternehmen trugen zu Softwareplattformen bei, die eine nahtlose Integration von Geschäftsfunktionen ermöglichen sollten. Abgesehen von der Softwarewelt haben auch andere Branchen den Fokus auf Prozesse durch die Linse spezifischer Unternehmen aus den 1980er Jahren mit Qualitätsmanagementinitiativen und -standards geschärft.

Seit den 1990er Jahren spielen die Disziplinen des Geschäftsprozessmanagements (BPM) und des Business Process Reengineering (BPR) eine wichtige Rolle, wenn Unternehmen ihre Leistung verbessern wollen. BPM konzentriert sich auf die Darstellung und Dokumentation der Arbeitsabläufe, Operationen und Prozesse eines Unternehmens mithilfe von Diagrammen, Flussdiagrammen oder anderen visuellen Hilfsmitteln. Ziel ist es, bestehende Prozesse zu verstehen und zu verbessern oder neue Prozesse zu entwerfen, um die Effizienz zu steigern und Ineffizienzen zu verringern. BPR wird als grundlegendes Überdenken und radikale Neugestaltung von Geschäftsprozessen angestrebt, um erhebliche Leistungs- und Produktivitätsverbesserungen zu erzielen. Es umfasst die Analyse und Gestaltung von Arbeitsabläufen und Prozessen innerhalb einer Organisation mit dem Ziel, effizienter und effektiver zu werden und sich an das sich verändernde Geschäftsumfeld anzupassen. BPR kann die Nutzung von Technologien und Informationssystemen beinhalten, um Prozesse zu automatisieren und die Entscheidungsfindung zu verbessern.

Bei BPM und BPR können mehrere Diagramme verwendet werden:

- Flussdiagramme: Eine einfache und intuitive Methode zur Darstellung eines Prozesses als eine Reihe von Schritten, Entscheidungen und Schleifen.
- Swimlane-Diagramme: Eine Art von Flussdiagramm, bei dem visuell dargestellt wird, wer oder was für die einzelnen Schritte eines Prozesses verantwortlich ist.
- Prozesslandkarten: Eine Übersicht über die an einem Prozess beteiligten Schritte und Aktivitäten, oft einschließlich Inputs, Outputs und Entscheidungspunkten.
- BPMN (Business Process Model and Notation [4]): Eine branchenübliche Notation für die Modellierung von Geschäftsprozessen, die von der Object Management Group (OMG) standardisiert wurde.
- EPC (Ereignisgesteuerte Prozesskette): Eine Art von Flussdiagramm, bei dem Ereignisse anstelle von Aktivitäten als treibende Kraft hinter einem Prozess verwendet werden.
- IDEF (Integrationsdefinition für Funktionsmodellierung): Eine Methode, die in der Softwareentwicklung zur Modellierung und Analyse von Prozessen und Systemen verwendet wird.

Die Wahl des Diagrammtyps zur Darstellung von Arbeitsabläufen hängt von der benötigten Detailtiefe, dem Zweck des Modells und der Zielgruppe ab. An dieser Stelle werden wir uns nur auf BPMN konzentrieren. Die Hauptgründe dafür sind: a) BPMN hat eine Standardnotation und eine breite Akzeptanz in verschiedenen Branchen und b) BPMN ist als Modellierungsdiagrammtyp in der Oracle Application Express (APEX) Funktion (manchmal auch als Erweiterung bezeichnet) namens Flows for APEX implementiert. Flows for APEX ist Open Source, lizenziert unter der sehr liberalen MIT-Lizenz. Flows for APEX ermöglicht es Entwicklern, webbasierte Anwendungen mit einer visuellen Drag-and-Drop-Schnittstelle zu erstellen und einzusetzen. Flows for APEX bietet eine Möglichkeit, komplexe Geschäftsprozesse zu automatisieren und zu vereinfachen, indem es sie in eine Reihe von Schritten und Aufgaben zerlegt und die Benutzer durch sie führt.

Flows for APEX kann eine Vielzahl von Elementen wie Seiten, Formulare, Berichte und Dialoge enthalten und kann so konfiguriert werden, dass es Verzweigungslogik, bedingte Verzweigungen, Datenvalidierung und Fehlerbehandlung enthält. Die visuelle Schnittstelle von Flows

for APEX macht es Entwicklern leicht, Geschäftsprozesse zu entwerfen, zu erstellen und zu testen, ohne dass eine umfangreiche Kodierung erforderlich ist. Flows for APEX kann mit anderen Oracle APEX-Komponenten wie SQL, PL/SQL und REST-Services integriert werden, um vollständige, webbasierte Anwendungen zu erstellen und kundenspezifische Lösungen für spezifische Geschäftsanforderungen wie Workflow-Management, Kunden-Onboarding und Datenmanagement zu entwickeln. Insgesamt bieten Flows for APEXs eine Möglichkeit für Unternehmen, komplexe Geschäftsprozesse zu rationalisieren und zu automatisieren, die Produktivität zu verbessern und Fehler und Ineffizienzen zu reduzieren. Ausgezeichnete Tutorials zu Flows for APEX sind auf dieser Seite verfügbar. Wie können wir also APEX als Low-Code-Programmierumgebung und Flows for APEX mit eingebetteter Standard-BPML-Darstellung verbinden?

Das Aktivitätsdiagramm (siehe Abbildung 17.1) stellt die vereinfachte Darstellung der Integration von Flows und APEX-Anwendung dar.

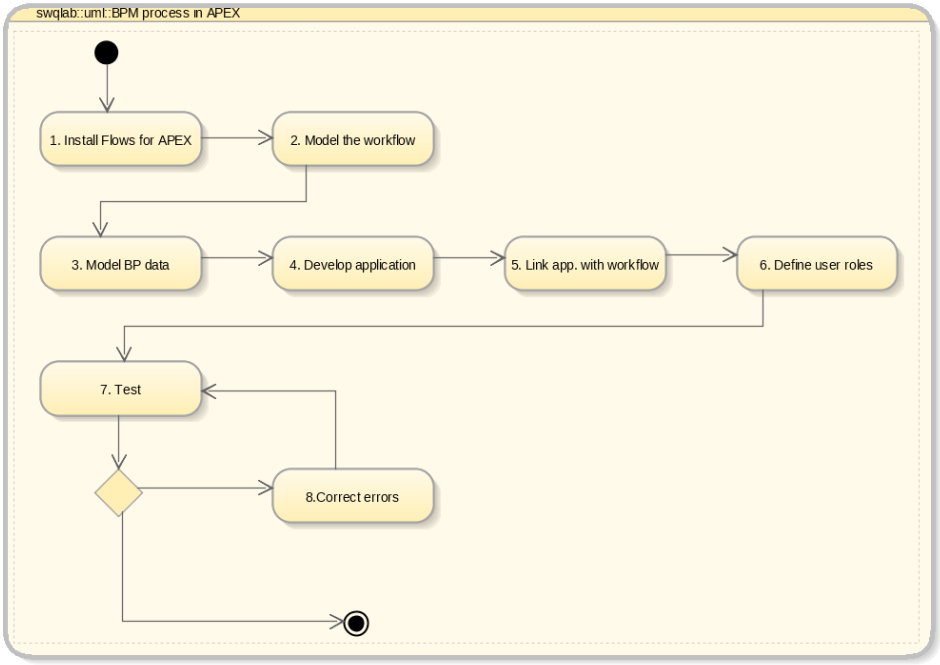



Abbildung 17.1: Integration von Flows for APEX in die APEX-Anwendung.

Die Schritte *3. Daten modellieren, 4. Anwendung entwickeln, 6. Benutzerrollen definieren, 7. Testen* und *8. Fehler aus der Anwendung entfernen* sind für alle APEX-Anwendungen gleich. Schritt *1. Installation von Flows for APEX* ist ähnlich wie die Installation jeder anderen verpackten Anwendung. Die wirklichen Neuerungen sind *2. Modellierung des Workflows* und *5. Verknüpfung der APEX-Anwendung mit dem Workflow*.

Lesern, die mit Business Process Management (BPM) und Business Process Modeling Notation (BPMN)-Diagrammen nicht vertraut sind, wird empfohlen, nach gängigen Lehrbüchern zu suchen und die bereits erwähnte BPMN-Spezifikation [4] zu studieren. Zur Erinnerung für alle Leser: Die folgenden Elemente von BPMN-Diagrammen sind die am häufigsten verwendeten:

- Start-Ereignis: Zeigt den Startpunkt eines Prozesses an.
- Aufgabe: Stellt eine einzelne Arbeitseinheit dar, wie z.B. eine Aktivität oder einen Schritt im Prozess.
- Tor: Stellt einen Entscheidungspunkt im Prozess dar, wie z.B. eine Gabelung im Flow oder eine bedingte Verzweigung.
- Sequence Flow: Verbindet die Elemente im Diagramm und stellt den Flow der Steuerung von einem Element zum anderen dar.

- End-Ereignis: Zeigt das Ende eines Prozesses an.
- Pool/Bahn: Stellt eine Gruppierung zusammengehöriger Aufgaben dar, die dazu dient, die Rollen und Zuständigkeiten der verschiedenen Prozessbeteiligten zu definieren.
- Nachrichtenfluss: Stellt den Flow von Nachrichten zwischen den Teilnehmern eines Prozesses dar, z.B. eine Kommunikation zwischen zwei Systemen.
- Datenobjekt: Stellt einen Teil der Daten dar, die in einem Prozess verwendet oder erzeugt werden, wie z.B. eine Rechnung oder ein Kundendatensatz.
- Datenspeicher: Stellt einen Container zum Speichern von Daten dar, die in einem Prozess verwendet oder erzeugt werden, wie z.B. eine Datenbank oder ein Dateisystem.

Diese Elemente können kombiniert werden, um detaillierte und genaue Modelle von Geschäftsprozessen zu erstellen, die es Unternehmen ermöglichen, verbesserungswürdige Bereiche zu identifizieren und ihre Arbeitsabläufe zu optimieren.

### 17.1.1 Flows für APEX installieren

Laden Sie die Zip-Datei von diese URL herunter und entpacken Sie alle Dateien auf Ihrem Computer. Folgen Sie den Anweisungen von diesem Link. Prüfen Sie vor der Installation des Flows for APEX-Plug-Ins die Anforderungen. Die Anforderungen sind erfüllt, wenn Sie Ihren kostenlosen APEX-Arbeitsbereich, Ihr kostenloses/zahlungspflichtiges OCI- oder Oracle Academy-Konto verwenden. Zwei zusätzliche Aufgaben nach der Installation sind:

- Erteilen Sie dem Arbeitsbereich die Berechtigung „Auftrag erstellen“. Sie oder der APEX-Instanzadministrator müssen eine Verbindung zur Datenbank herstellen (mit OCI-Schnittstelle, SQL Developer, SQL Plus, TOAD) und den Befehl erteilen. Sie können dieses Recht nicht innerhalb von APEX vergeben.
- Arbeitsbereich, Standardanwendung und Benutzerinformationen in der Anwendung „Flows for APEX“ konfigurieren. Sie können diesen Schritt jederzeit nachholen, wenn Sie Ihre eigene prozessgesteuerte Anwendung entwickeln.

Obwohl manchmal als Feature oder Plug-in bezeichnet, werden bei der Installation neue Datenbankobjekte (Tabellen, Ansichten) und eine Anwendung namens „Flows for APEX“ erstellt. Diese Anwendung ist eine administrative Schnittstelle, die es Ihnen ermöglicht:

- Instanzen von Workflows auf dem Dashboard zu überwachen (Dashboard),
- neue Workflows visuell zu erstellen (Flow Management) und
- Workflows auszuführen, indem Sie Instanzen erstellen, die zu Instanzen gehörenden Aufgaben erledigen und die Workflows testen (Flow Monitor).

### 17.1.2 Beispielhafte Prozessablaufanwendung installieren

Ein Beispiel für eine Prozessflussanwendung ist in der gleichen Zip-Datei wie Flows for APEX enthalten. Die Installation ist einfach. Sie befasst sich mit Spesenabrechnungen. Der Mitarbeiter erstellt eine Spesenabrechnung, die vom Vorgesetzten geprüft wird. Wird die Spesenabrechnung abgelehnt, so wird der Mitarbeiter darüber informiert. Für genehmigte Anträge bereitet die Buchhaltung die Zahlung vor, führt eine Banktransaktion durch und setzt den Status der Zahlung. Wenn der Antrag bezahlt wurde, ist die Instanz dieses Workflows abgeschlossen. Die Anwendung ist dem realen Prozess in jeder Organisation sehr ähnlich. Da sie so gut dokumentiert ist, bietet sie hervorragendes Studienmaterial.

### 17.1.3 Übungen lesen und üben

Es gibt ausgezeichnete Tutorials für Flows für APEX. Beginnen Sie mit diese Seite. Am Ende dieser einführenden Übungen finden Sie BPMN-Tutorial und APEX-Integrationstutorial.

Das Lesen und Üben wird Sie in die Lage versetzen, Ihre eigene prozessorientierte Anwendung zu entwickeln. Das Durcharbeiten aller Übungen wird den Leser wahrscheinlich einige Stunden in Anspruch nehmen. Der Leser wird nachdrücklich ermutigt, Learning-by-Doing zu betreiben.

Der Rest dieses Kapitels bietet die gleiche Vorlage wie andere Kapitel in diesem Lehrbuch (von der Problemdefinition bis zur Anwendung) mit zusätzlicher Einbeziehung von BPML und Anwendungsintegration mit Flows for APEX. Allerdings ist der Umfang der Anwendung, die im Kapitel 17 entwickelt wird, auf die Kernfunktionalität beschränkt und überspringt die Details der Autorisierung. Sie wird Anleitungen zu folgenden Themen bieten:

- Arbeitsabläufe in Flows for APEX definieren
- Start der Instanz des definierten Prozesses innerhalb der entwickelten Anwendung
- den Fluss der Instanzen innerhalb der entwickelten Anwendung überwachen
- jede Aufgabe einer bestimmten Instanz des Arbeitsablaufs innerhalb der entwickelten Anwendung abzuschließen

Eine eingeschränkte Anwendung kann auch eine schöne Herausforderung für begeisterte Lernende darstellen, oder?

## 17.2 Problemstellung

Ein mittelgroßes Fertigungsunternehmen erhält von seinen Kunden Anfragen für individuell gefertigte Produkte. Jede Anfrage kann mehrere Dokumente mit Spezifikationen, erforderlichen Normen, Schemata, Skizzen und Ähnlichem enthalten. Der Kunde kann mehrere Anfragen haben. Der Vertriebsmitarbeiter erstellt und klassifiziert die Anfragedokumente in drei Kategorien: Fertigung, Finanzen und Geschäft. Um eine Anfrage zu bewerten, muss der Vertrieb mehrere Dokumente für die Entscheidungsträger vorbereiten. Der Produktionsleiter und sein Team bewerten die Anfrage unter Fertigungsaspekten. Sie geben ein Gutachten über die Fähigkeit des Unternehmens ab, das gewünschte Produkt herzustellen. Wenn das Produkt hergestellt werden kann, gibt das Team die geschätzte Dauer des Herstellungsprozesses in Tagen an. Der Finanzmanager und sein Team bewerten den finanziellen Aspekt der Anfrage. Sie erstellen ein Gutachten über die Fähigkeit des Unternehmens, Finanzmittel bereitzustellen, und schätzen den erwarteten Gewinn. Die endgültige Bewertung und Entscheidung wird vom Vorstandsvorsitzenden und seinem Team getroffen. Sie prüfen die Machbarkeit (Herstellung, Finanzen), schätzen die Bedeutung des Kunden ein und treffen eine Entscheidung über die Anfrage. Abhängig von der Fähigkeit des Unternehmens, die Geschäftsmöglichkeit herzustellen und zu finanzieren, sowie von den Geschäftsaussichten entscheidet der Geschäftsführer, wie er dem Kunden antwortet: a) Übermittlung eines Geschäftsvorschlags mit dem Preis des Herstellers und der voraussichtlichen Lieferdauer oder b) Mitteilung an den Kunden, dass der Hersteller keinen Geschäftsvorschlag übermitteln kann.

Schlechte Geschäftsentscheidungen (Zusendung eines Geschäftsvorschlags oder Ablehnung des Geschäfts) können katastrophale Auswirkungen auf das produzierende Unternehmen haben, wie z. B. Gewinneinbußen, geringer Gewinn, schlechter Ruf aufgrund von überschrittenen Lieferterminen und sogar Konkurs. Die dargestellte Situation bereitet allen Beteiligten Kopfzerbrechen: dem Vertriebsmitarbeiter, dem Produktionsleiter, dem Finanzmanager und dem Vorstandsvorsitzenden. Die IT-Abteilung ist dafür verantwortlich, benutzerfreundliche, zuverlässige und sichere Softwareunterstützung für den beschriebenen Arbeitsablauf bereitzustellen. Ihr Problem ist es, eine solche Unterstützung mit Flows for APEX zu entwickeln.

Um die Anwendung zu entwickeln, müssen Sie Flows for APEX installiert haben. In diesem Kapitel wird der Geschäftsfall analysiert (Beschreibung, halbstrukturierte Beschreibung, Anwendungsfalldiagramm, Prozessmodell, Datenmodell) und die Anwendung entwickelt. Sie werden lernen:

- wie man ein Prozessmodell (Workflow) für diesen Geschäftsfall definiert,
- wie man ein Datenmodell erstellt und eine APEX-Anwendung entwickelt
- wie man die entwickelte Anwendung mit dem definierten Arbeitsablauf verknüpft

Bitte beachten Sie, dass eine Benutzerautorisierung nicht implementiert wird, da dies dieses Kapitel über einen angemessenen studentischen Aufwand hinaus erweitern würde.

Tabelle 17.1: Beschreibung des Anwendungsfalls: Vorbereitung der Untersuchungsunterlagen.

| **Keyword** | **Wert** |
|---|---|
| ID: | *ch17-01* |
| Titel: | *Dokumentation der Anfrage vorbereiten* |
| Beschreibung: | *Vertriebsmitarbeiter erstellen neue Anfragen und fügen zugehörige Dokumente hinzu* |
| Primärer Akteur: | *Vertriebsmitarbeiter* |
| Voraussetzungen: | Kunde hat bereits alle Dokumente zur Anfrage geschickt |
| Nachbedingungen: | |
| *Haupt* | - |
| Erfolgs-Szenario: | 1. Neue Anfrage hinzufügen<br>2. verwandte Dokumente zu einer bestimmten Anfrage hinzufügen<br>3. mit der Schaltfläche „Bestätigen„ bestätigen oder mit der Schaltfläche „Abbrechen„ verwerfen |
| Erweiterungen: | - |
| Häufigkeit der Nutzung: | *Ca. 250 pro Jahr, durchschnittlich 5 pro Woche.* |
| Status: | [Entwicklungsstand] |
| Eigentümer: | Vertriebsmitarbeiter |
| Priorität: | *hoch* |

## 17.3 Anwendungsfälle

### 17.3.1 Narrative Beschreibung

Vier Akteure sind an vier Anwendungsfällen beteiligt. Jeder Akteur ist für einen Anwendungsfall verantwortlich: Der Vertriebsmitarbeiter bereitet die Unterlagen für die Anfrage vor, der Produktionsleiter bewertet die Fertigungsaspekte, der Finanzleiter bewertet die finanziellen Aspekte und der Vorstandsvorsitzende trifft die Entscheidung. Das Anwendungsfalldiagramm (siehe Abbildung 17.2) geht davon aus, dass der Designer des Arbeitsablaufs entscheidet, ob Produktions- und Finanzmanager parallel oder sequentiell arbeiten sollen. Wenn sie parallel arbeiten und einer oder beide entscheiden, dass die Anforderungen der Anfrage nicht erfüllt werden können, dann entsteht zwar unnötiger Aufwand (Kosten), aber der Workflow kann schneller abgeschlossen werden. Entscheidet sich der Workflow-Designer für einen sequentiellen Ansatz, so wird der Workflow mehr Zeit in Anspruch nehmen, aber es entsteht weniger unnötiger Aufwand.

### 17.3.2 Semi-strukturierte Beschreibung

UML schlägt vor, dass jeder Anwendungsfall auch eine halbstrukturierte Beschreibung hat. Vier verschiedene halbstrukturierte Beschreibungen sind in den Tabellen 17.1, 17.2, 17.3 und 17.4 enthalten.

### 17.3.3 Anwendungsfalldiagramm

Ein Anwendungsfalldiagramm kann auch eine Generalisierung darstellen. In dem gegebenen Kontext hat ein Mitarbeiter vier spezielle Instanzen (Feldmanager) und jeder Manager ist mit einem Anwendungsfall verbunden. Alle vier Anwendungsfälle sind in Abbildung 17.2 dargestellt.

Tabelle 17.2: Beschreibung des Anwendungsfalls: Fertigungsaspekte der Untersuchung bewerten.

| **Keyword** | **Wert** |
|---|---|
| ID: | *ch17-02* |
| Titel: | *Fertigungsaspekte der Untersuchung bewerten* |
| Beschreibung: | *Der Produktionsleiter und sein Team geben ein Gutachten darüber ab, ob das Unternehmen in der Lage ist, ein Produkt herzustellen, und wenn ja, wie lange (in Tagen) die Herstellung einer bestimmten Menge dauern würde. Eine Begründung des Gutachtens ist obligatorisch.* |
| Primärer Akteur: | *Produktionsleiter* |
| Voraussetzungen: | Dokumentation der Anfrage vorbereitet |
| Nachbedingungen: | |
| *Haupt* | - |
| Erfolgs-Szenario: | 1. Meinung einbringen<br>2. mit der Schaltfläche „Bestätigen“ bestätigen oder mit der Schaltfläche „Abbrechen“ verwerfen |
| Erweiterungen: | - |
| Häufigkeit der Nutzung: | *Ca. 250 pro Jahr, durchschnittlich 5 pro Woche.* |
| Status: | [Entwicklungsstand] |
| Eigentümer: | Produktionsleiter |
| Priorität: | *hoch* |

Tabelle 17.3: Beschreibung des Anwendungsfalls: Bewertung der finanziellen Aspekte der Untersuchung.

| **Keyword** | **Wert** |
|---|---|
| ID: | *ch17-03* |
| Titel: | *Bewertung der finanziellen Aspekte der Untersuchung* |
| Beschreibung: | *Der Finanzmanager und sein Team geben ein Gutachten darüber ab, ob das Unternehmen in der Lage ist, das Produkt zu finanzieren, und wenn ja, wie hoch der erwartete Gewinn wäre. Eine Begründung des Gutachtens ist obligatorisch.* |
| Primärer Akteur: | *Finanzverwalter* |
| Voraussetzungen: | Dokumentation der Anfrage vorbereitet |
| Nachbedingungen: | |
| *Haupt* | - |
| Erfolgs-Szenario: | 1. Meinung einbringen<br>2. mit der Schaltfläche „Bestätigen“ bestätigen oder mit der Schaltfläche „Abbrechen“ verwerfen |
| Erweiterungen: | - |
| Häufigkeit der Nutzung: | *Ca. 250 pro Jahr, durchschnittlich 5 pro Woche.* |
| Status: | [Entwicklungsstand] |
| Eigentümer: | Finanzverwalter |
| Priorität: | *hoch* |

Tabelle 17.4: Beschreibung des Anwendungsfalls: Bewertung der wirtschaftlichen Aspekte der Untersuchung.

| **Keyword** | **Wert** |
|---|---|
| ID: | *ch17-04* |
| Titel: | *Bewertung der wirtschaftlichen Aspekte der Untersuchung* |
| Beschreibung: | *Der Geschäftsführer und sein Team treffen die Entscheidung, ob sie dem Kunden einen Geschäftsvorschlag unterbreiten oder das Angebot ausschlagen. Die Begründung der Stellungnahme ist obligatorisch und wird vom Vorstand verlangt.* |
| Primärer Akteur: | *Finanzverwalter* |
| Voraussetzungen: | Die Anwendungsfälle 17.1, 17.2 und 17.3 sind abgeschlossen. |
| Nachbedingungen: | - |
| *Haupt* | - |
| Erfolgs-Szenario: | 1. Meinung einbringen<br>2. mit der Schaltfläche „Bestätigen“ bestätigen oder mit der Schaltfläche „Abbrechen“ verwerfen |
| Erweiterungen: | - |
| Häufigkeit der Nutzung: | *Ca. 250 pro Jahr, durchschnittlich 5 pro Woche.* |
| Status: | [Entwicklungsstand] |
| Eigentümer: | Geschäftsführer/in |
| Priorität: | *hoch* |

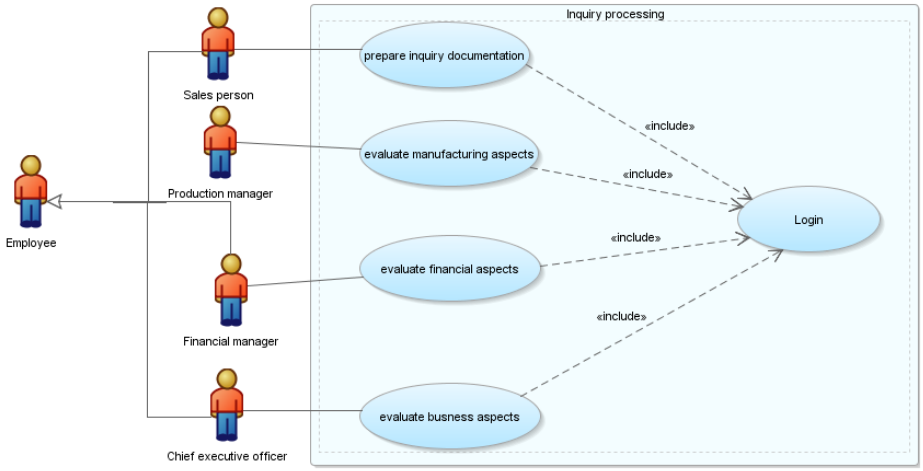



Abbildung 17.2: Anwendungsfalldiagramm.

## 17.4 Workflow-Modell

Das anfängliche Workflow-Modell führt eine Vorbereitungsaufgabe ein, die sich dann in ein paralleles Gateway mit zwei Aufgaben (Herstellung und finanzielle Bewertung) aufteilt. Der Arbeitsablauf wird mit der letzten Aufgabe (betriebswirtschaftliche Auswertung) fortgesetzt, wenn die beiden vorherigen Aufgaben abgeschlossen sind (siehe 17.3).

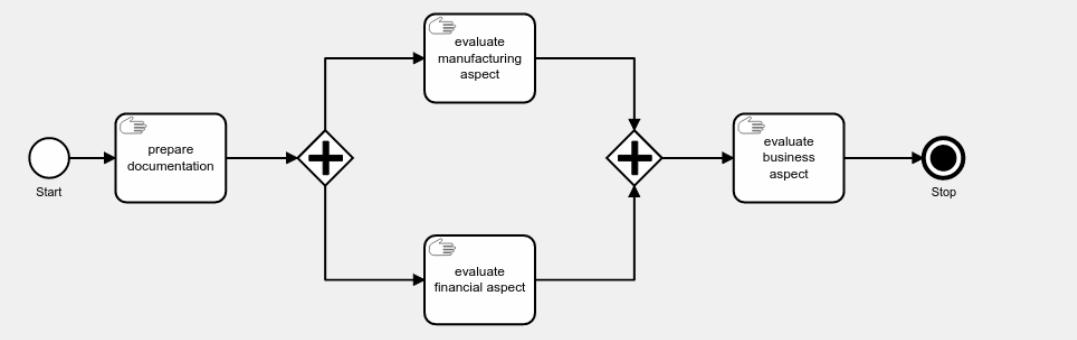



Abbildung 17.3: Workflow - Bearbeitungsanfrage - BPMN-Diagramm.

Alle vier Aufgaben sind als „Manuelle Aufgabe“ definiert und das Endereignis (Stop) ist als „Terminate end event“ definiert. Wir können den Workflow sofort im Flow Monitor (in der Anwendung „Flows for APEX“) testen. Speichern Sie den Ablauf als CH17. Navigieren Sie dann zum Flow Monitor. Klicken Sie auf die Schaltfläche „Create instance“ (siehe Abbildung 17.4).

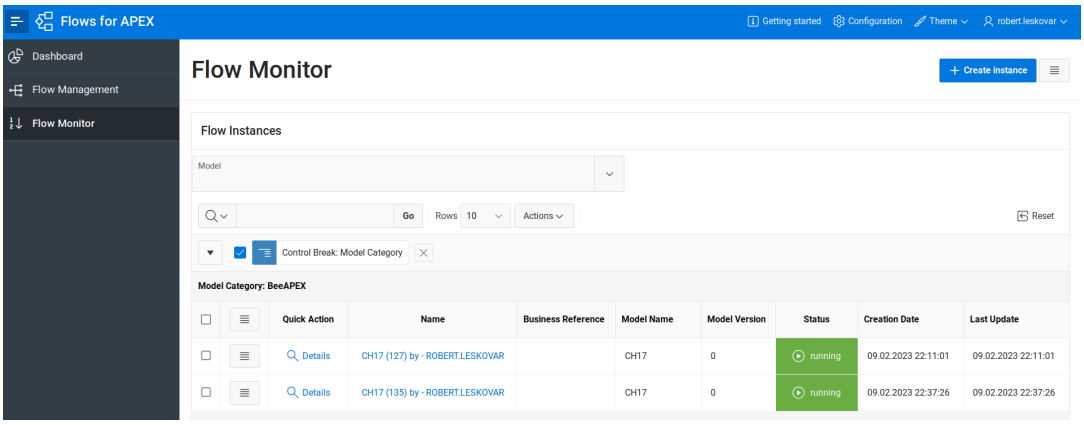



Abbildung 17.4: Erstellen von Instanzen.

Wählen Sie das Modell (CH17) und geben Sie den eindeutigen Namen der Instanz des Flows CH17 ein (siehe Abbildung 17.5).

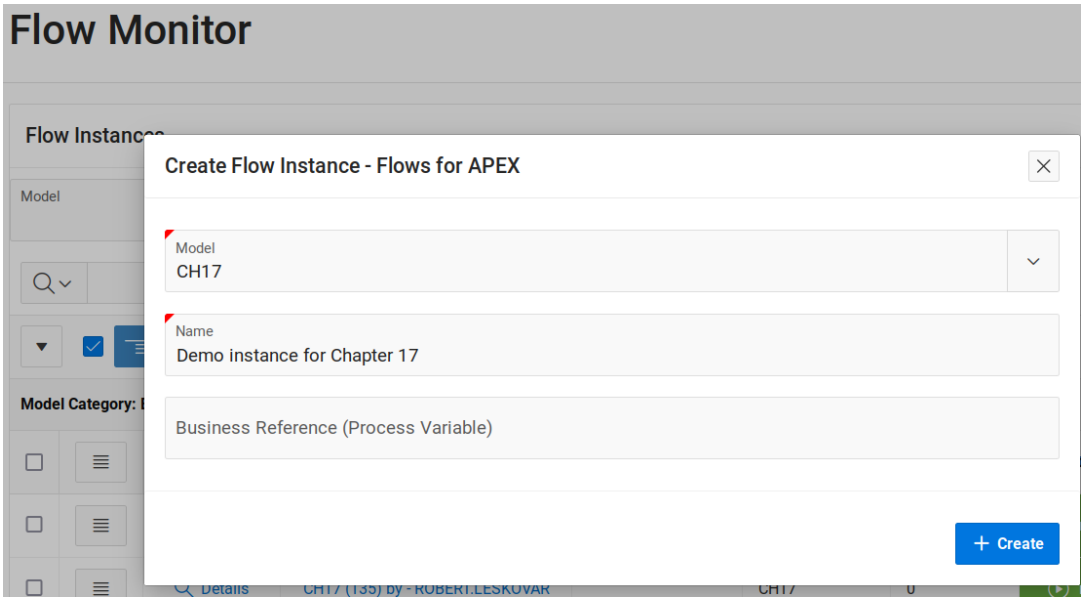



Abbildung 17.5: Erzeugen einer Instanz des Flows CH17.

Eine neue Instanz erscheint. Klicken Sie auf das angezeigte Symbol in Abbildung 17.6 und wählen Sie „Start“.




Abbildung 17.6: Demo-Instanz starten.

Der Status der Instanz ändert sich in „running“. Klicken Sie auf „Details“ für die Demo-Instanz. Stellen Sie die Einstellungen für die Side-by-Side-Anzeige auf und klicken Sie in der Spalte „Quick Action“ auf „Complete“ (siehe Abbildung 17.7).

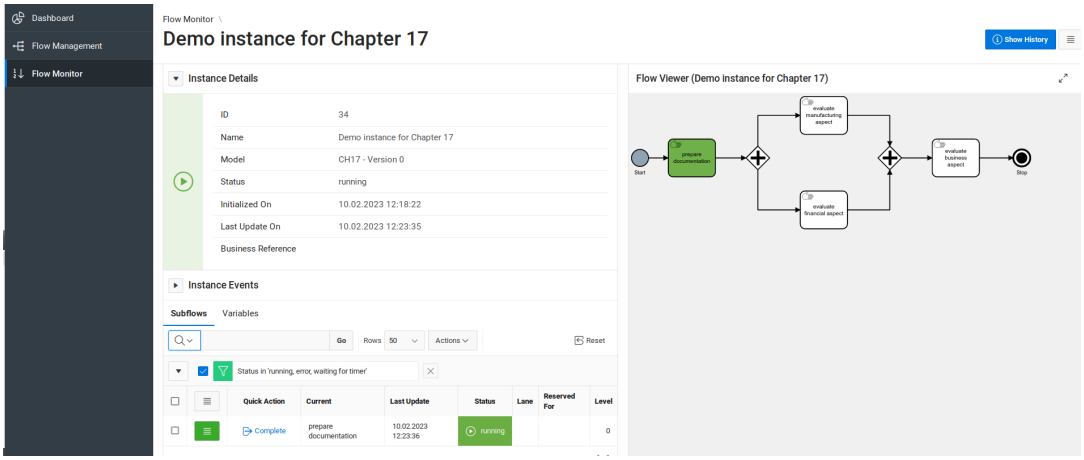



Abbildung 17.7: Schließen Sie die erste Aufgabe in der Demo-Instanz ab.

Jetzt läuft der Flow mit parallelem Tor und zwei Aufgaben weiter (siehe Abbildung 17.8).

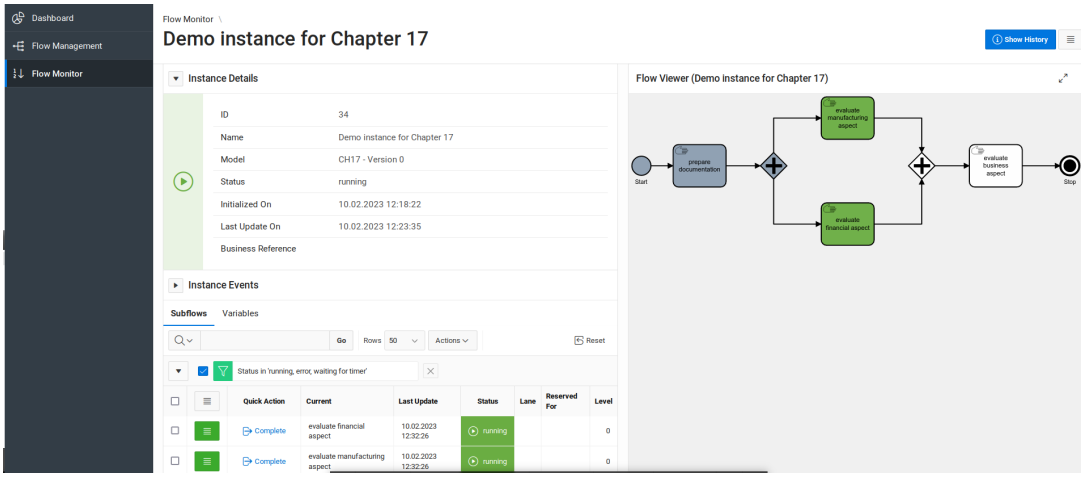



Abbildung 17.8: Ausführen von Aufgaben in parallelen Gates.

Klicken Sie in der Spalte „Quick Action“ für beide Aufgaben auf „Complete“ und fahren Sie mit der letzten Aufgabe fort. Wiederholen Sie den Abschluss für die letzte Aufgabe und die Instanz ist abgeschlossen (siehe Abbildung Abbildung 17.9).

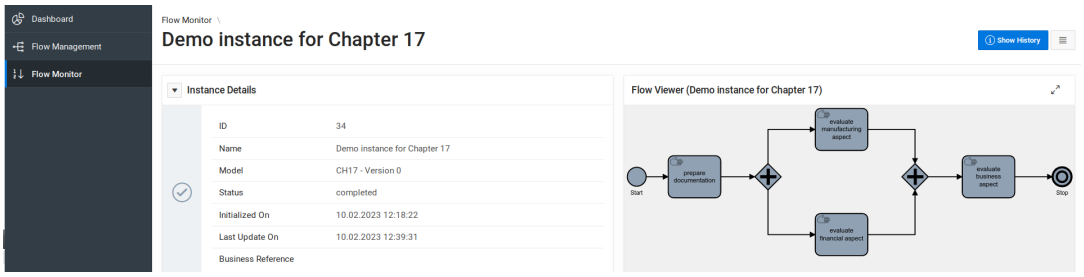



Abbildung 17.9: Abgeschlossene Demo-Instanz.

Der gesamte Verlauf der Instanz wird angezeigt, wenn Sie auf „Verlauf anzeigen“ klicken. Wir haben bewiesen, dass der Workflow CH17 ausführbar ist und wir können Daten und Anwendungsschnittstellen modellieren.

## 17.5 Datenmodell

### 17.5.1 Narrative Beschreibung des Datenmodells

Die wichtigste Entität ist **ch17 inquiry**. Eine Instanz von **ch17 inquiry** ist mit einer Instanz von **ch17 customer** verbunden, während ein Kunde mehrere Anfragen haben kann. Jede Instanz von **ch17 inquiry** muss mindestens drei Instanzen von **ch17 document** haben. Jede Instanz von **ch17 document** gehört zu einer Instanz von **ch17 document class**.

Die Attribute für die Entität **ch17 inquiry** sind: ID, Kundenanforderungen (Preis pro Auftrag, Menge, Liefertermin), Fertigungsaspekte (Fähigkeit, Lieferzeit für die angefragte Menge, Begründung), finanzielle Aspekte (finanzielle Fähigkeit, erwarteter Gewinn für die angefragte Menge) und geschäftliche Aspekte (Entscheidung, die Gelegenheit wahrzunehmen oder nicht, Begründung).

Wir werden nur eine begrenzte Anzahl von Attributen für andere Entitäten definieren, um diesen Fall kompakter zu gestalten.

### 17.5.2 Logisches Datenmodell

Das logische Datenmodell ist in Abbildung 17.10 dargestellt.

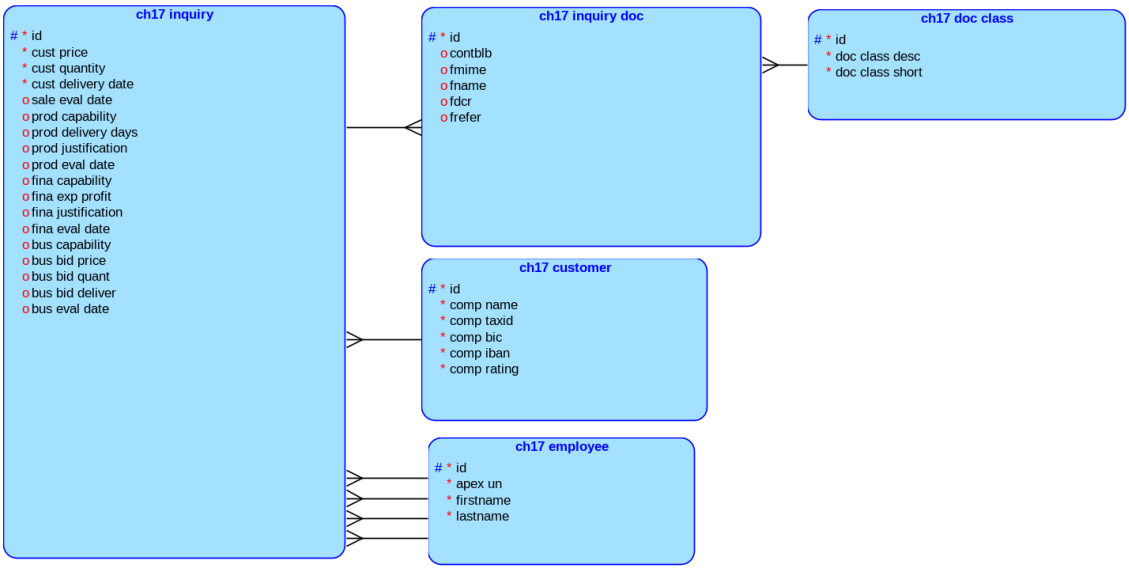



Abbildung 17.10: Logisches Datenmodell.

### 17.5.3 Relationales Datenmodell

Das relationale Datenmodell ist in Abbildung 17.11 dargestellt.

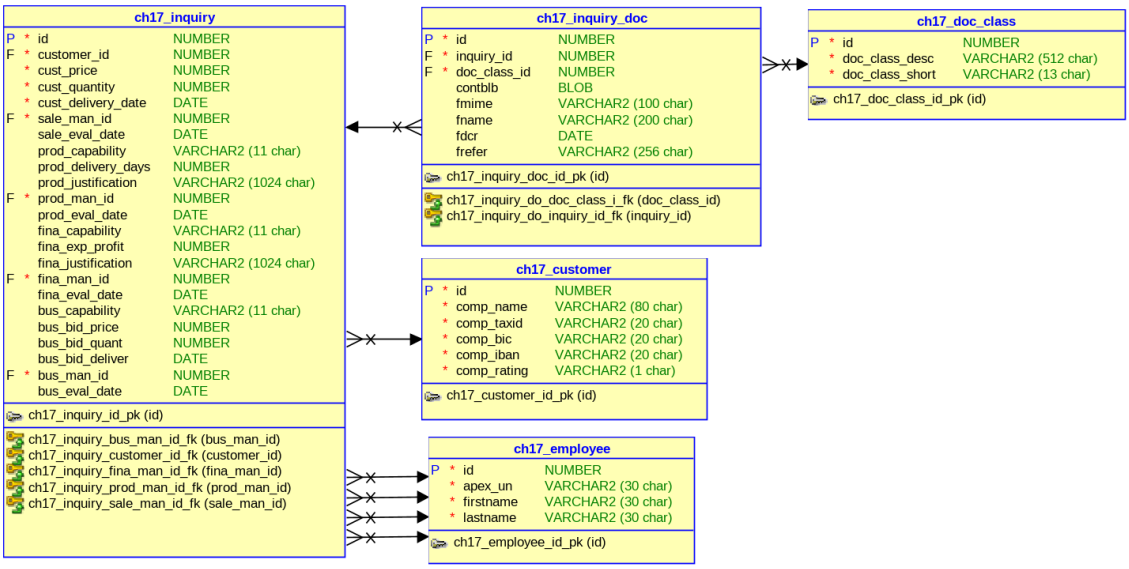



Abbildung 17.11: Relationales Datenmodell.

### 17.5.4 Quick SQL zur Erstellung von SQL-Skripten

Quick SQL ist ein praktisches Werkzeug zur Beschleunigung der Entwicklung von Datenmodellen. Es kann für das Prototyping durch Generierung von SQL-Skripten im APEX (Generierung von Anwendungen aus Skripten) oder für das Reverse Engineering von logischen Datenmodellen durch Import von SQL-Skripten in den SQL Developer Data Modeler und Transformation von relationalen in logische Modelle verwendet werden. Das Datenmodell, das in diesem Kapitel vorgestellt wird, kann durch das folgende Quick SQL beschrieben werden:

```
ch17_employee
apex_un vc30 /nn,
```

```
firstname vc30 /nn,
lastname  vc30 /nn

ch17_customer
comp_name vc80 /nn,
comp_taxid vc20 /nn,
comp_bic vc20 /nn,
comp_iban vc20 /nn,
comp_rating vc1 /check 'A','B','C','D' /nn

ch17_doc_class
doc_class_desc vc512 /nn,
doc_class_short vc13 /check 'MANUFACTURING','FINANCIAL','BUSINESS','OTHER' /nn

ch17_inquiry
customer_id num /fk ch17_customer /nn,
cust_price  num /nn,
cust_quantity num /nn,
cust_delivery_date date /nn,
cust_product_item vc50 /nn,
sale_man_id num /fk ch17_employee /nn,
sale_eval_date date,
prod_capability vc11 /check 'YES','NO','CONDITIONAL','N.A.',
prod_delivery_days num,
prod_justification vc1024,
prod_man_id num /fk ch17_employee /nn,
prod_eval_date  date,
fina_capability vc11 /check 'YES','NO','CONDITIONAL','N.A.',
fina_exp_profit num,
fina_justification vc1024,
fina_man_id num /fk ch17_employee /nn,
fina_eval_date date,
bus_capability vc11 /check 'YES','NO','CONDITIONAL','N.A.',
bus_bid_price num,
bus_bid_quant num,
bus_bid_deliver date,
bus_man_id num /fk ch17_employee /nn,
bus_eval_date date

ch17_inquiry_doc
inquiry_id /fk ch17_inquiry /nn,
doc_class_id /fk ch17_doc_class /nn,
contblb blob,
fmime vc100,
fname vc200,
fdcr date,
frefer vc256
```

Einer der Vorteile von Quick SQL ist seine Kompaktheit und Einfachheit. Die Anzahl der Zeilen in Quick SQL beträgt die Hälfte des SQL-Skripts und die Anzahl der Zeichen ist weniger als ein Drittel des SQL-Skripts. Zeit und Mühe für das Erlernen von Quick SQL sind eine gute Investition.

Das APEX bietet auch einen benutzerfreundlichen Editor mit Autovervollständigung, aber die Vorteile von Quick SQL werden eher von Benutzern genutzt, die SQL sprechen.

### 17.5.5 Sequenz und zwei gespeicherte Funktionen

In diesem Geschäftsfall zwingen uns die Anwendungsanforderungen dazu, einige leicht verständliche Konzepte der Oracle-Datenbank zu verwenden:

- der Mechanismus der Sequenz - er generiert ganzzahlige Werte, indem er Anfangswert, Maximalwert, Inkrement usw. definiert. Mit diesem Mechanismus generiert unsere Anwendung eine eindeutige Nummer, um die Anfrage zu identifizieren und den Namen der Flussinstanz zusammenzustellen.
- gespeicherte Funktionen in der Sprache PL/SQL nehmen Eingabeparameter entgegen und geben einen Wert zurück. Unsere Anwendung wird zwei benutzerdefinierte Funktionen verwenden: eine, um die Größe des in der BLOB-Spalte gespeicherten Dokuments zu ermitteln, und eine, um die Anzahl der Dokumente zu zählen, die einer bestimmten Anfrage zugeordnet sind.

Wir können Sequenzen und gespeicherte Funktionen in SQL Workshop > SQL Command definieren. Die Sequenz hat einen Namen, einen Minimalwert, einen Maximalwert, ein Inkrement, einen Startwert und einige andere Eigenschaften, die im Moment für uns nicht wichtig sind:

```
CREATE SEQUENCE   \glqq CH17_SEQ_INQUIRY\grqq\  MINVALUE 1 MAXVALUE 999999999999
INCREMENT BY 1 START WITH 100 CACHE 1
NOORDER  NOCYCLE  NOKEEP  NOSCALE  GLOBAL;
```

Wir werden auch zwei gespeicherte Funktionen definieren, die als Kommentare innerhalb des Codes erklärt werden - zwei Bindestriche am Anfang der Zeilen sind Kommentare. Die erste Funktion heißt ch17_doc_bytesize:

```
create or replace function ch17_doc_bytesize(xdocid in number) return number
-- Function ch17_doc_bytesize returns the size of the BLOB, stored in contblb
-- column in the table ch17_inquiry_doc for particular row.
-- Function parameter 1 is the ID of of ch17_inquiry_doc table.
-- The length of contblb column is provided by another system level PL/SQL
-- function called dbms_lob.getlength.
is
wdoc_size number := 0;
wblob blob;
Begin
select contblb into wblob from ch17_inquiry_doc where id = xdocid;
wdoc_size := dbms_lob.getlength(wblob);
RETURN  wdoc_size;
END ch17_doc_bytesize;
```

Die zweite Funktion heißt ch17_count_docs:

```
create or replace function ch17_count_docs(xinqid in number) return number
-- Function ch17_count_docs returns the number of documents associated with
-- particular inquiry. Function parameter 1 is the ID of inquiry.
-- If the length of the BLOB in the table ch17_inquiry_doc is greater than 0,
-- then we assume that the content is stored. So, function adds 1 to variable
-- wnumdocs, which is used as a counter. When all matching rows in the
-- table ch17_inquiry_doc are examined, than counter function returns counter.
is
```

```
wnumdocs number := 0;
CURSOR c_blob is
select contblb from ch17_inquiry_doc where inquiry_id = xinqid;
Begin
FOR r_blobs IN c_blob
LOOP
if dbms_lob.getlength(r_blobs.contblb) > 0 then
wnumdocs := wnumdocs + 1;
end if;
END LOOP;
RETURN  wnumdocs;
END ch17_count_docs;
```

## 17.6 Anwendungsschnittstellen

Wir empfehlen dem Leser, die verpackte Anwendung für dieses Kapitel zu installieren, bevor er die Anwendung entwirft. Den Link dazu finden Sie in den ergänzenden Lernmaterialien zu diesem Kapitel. Die angegebenen Seitenzahlen beziehen sich auf die verpackte Anwendung. Der Leser kann natürlich andere Seitenzahlen entwickeln und verwenden, um die gleiche Funktionalität wie in diesem Abschnitt beschrieben zu erhalten.

Die Anwendungsschnittstelle bietet Funktionalitäten für vier Akteure: Vertriebsleiter, Produktionsleiter, Finanzleiter und Geschäftsführer. Sie wurde zunächst mit einem Assistenten erstellt, der alle Funktionen enthält (progressiv, Web-App, Info-Seite, Zugangskontrolle, Aktivitätsberichte, Konfigurationsoptionen, Feedback und Auswahl des Themenstils, insgesamt 21 Seiten). Standardmäßig wird auch eine Anmeldeseite erstellt. Die Startseite enthält reines HTML, um die Geschäftssituation zu beschreiben. Sie enthält Links zum erforderlichen Plug-in und zum Tutorial (siehe Abbildung 17.12, Seite 1 in der verpackten Anwendung).

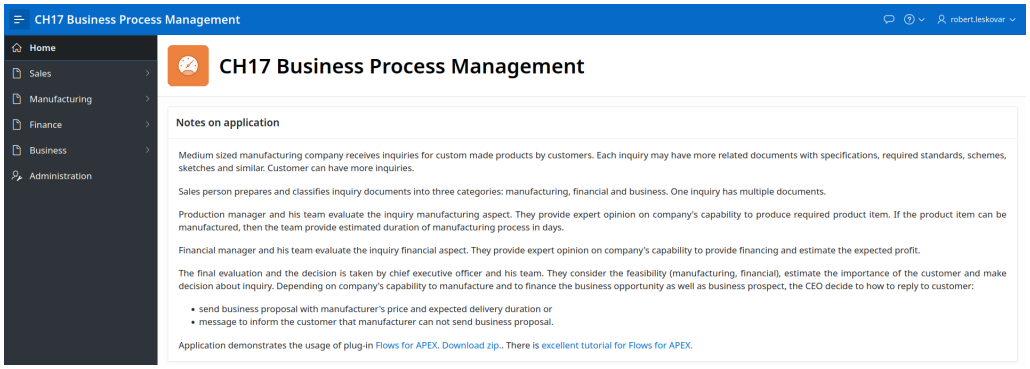



Abbildung 17.12: Startseite der Anwendung „CH17 Business Process Management“.

Einige LOVs und Plug-ins wurden vor der Anwendungsentwicklung im engeren Sinne eingerichtet oder einbezogen.

### 17.6.1 Liste der Werte in Shared Components

In dieser Anwendung werden die folgenden Wertelisten definiert:

- CH17_LOV_CAPABILITY: Typ ist statisch, Anzeigewerte sind (YES, NO, CONDITIONAL, Not Available), Rückgabewerte sind (YES, NO, CONDITIONAL, N.A.). Diese LOV wird verwendet, um drei verschiedene Fähigkeiten zu füllen (Fertigung, Finanzen und Geschäft).
- CH17_LOV_CRATING: Typ ist statisch, Anzeigewerte sind (Exzellente Bewertung, Gute Bewertung, Zufriedenstellende Bewertung, Nicht zufriedenstellende Bewertung), Rückgabe-

werte sind (A, B, C, D). Diese LOV wird für die Eingabe der Kundenbewertung verwendet.

- CH17_LOV_CUSTOMER: Typ ist dynamisch, basiert auf der Tabelle CH17_CUSTOMER, Anzeigewert ist DOC_CLASS_DESC und Rückgabewert ist ID. Diese LOV wird in Formularen und Berichten verwendet, um die Lesbarkeit zu verbessern und die Eingabe einer falschen Kunden-ID zu verhindern.
- CH17_LOV_DOC_CLASS: Typ ist dynamisch, basiert auf der Tabelle CH17_DOC_CLASS, Anzeigewert ist COMP_NAME und Rückgabewert ist ID. Diese LOV wird in Formularen und Berichten verwendet, um die Lesbarkeit zu verbessern und die Eingabe einer falschen Dokumentenklasse im Zusammenhang mit einer Anfrage zu verhindern.
- CH17_LOV_EMPLOYEE: Der Typ ist dynamisch und basiert auf einer SQL-Abfrage der Tabelle CH17_EMPLOYEE:

  ```
  SELECT FIRSTNAME || ' ' || LASTNAME as d,
  id as r
  FROM CH17_EMPLOYEE order by LASTNAME, FIRSTNAME
  ```

  Diese LOV wird auf Formularen und Berichten verwendet, um zur Lesbarkeit beizutragen und um zu verhindern, dass eine falsche Person eingegeben wird, die die Dokumentation erstellt und die Anfrage auswertet. Der Verkettungsoperator wird verwendet, um eine Zeichenfolge mit dem Vornamen des Mitarbeiters, einem Leerzeichen und dem Nachnamen zu bilden.
- CH17_LOV_INQUIRY: Typ ist dynamisch und basiert auf einer SQL-Abfrage in der Tabelle CH17_INQUIRY:

  ```
  SELECT
  'Instance:' || ch17_inquiry.id || ' (Customer: ' || comp_name || ',
  product:' || cust_product_item  as d,
  ch17_inquiry.id as r
  FROM ch17_inquiry, ch17_customer
  WHERE ch17_inquiry.customer_id = ch17_customer.id;
  ```

  Diese LOV wird in Formularen und Berichten verwendet, um die Lesbarkeit zu verbessern und die Eingabe einer falschen Anfrage-ID zu verhindern. Der Verkettungsoperator wird verwendet, um eine lange Zeichenkette mit vielen Informationen anzuzeigen.

### 17.6.2 Plug-ins in Shared Components

Die folgenden Plug-ins in Flows for APEX werden importiert: Flow-Instanz verwalten, Flow-Instanz-Schritt verwalten, Flow-Instanz-Variablen verwalten und Viewer. Diese Plug-ins werden aus dem installierten Flows for APEX-Plug-in kopiert.

### 17.6.3 Verkaufsleiter

Anwendungsschnittstellen ermöglichen es dem Vertriebsleiter:

- Workflow-Instanzen zu initiieren (siehe Abbildung 17.13, Seite 2 im Antrag)
- Arbeitsabläufe zu überwachen (siehe Abbildung 17.14, Seite 5 in der Anwendung)
- zu bestätigen, dass alle Dokumente für weitere Auswertungen in der Datenbank gespeichert sind (siehe Abbildung 17.15, Seite 4 im Antrag)
- Einblick in alle Anfragen zu erhalten (siehe Abbildung 17.17, Seite 6 in der Anwendung)
- Einsicht in alle Dokumente im Zusammenhang mit Anfragen zu erhalten (siehe Abbildung 17.17, Seite 8 im Antrag)
- das Dokument in die entsprechende Spalte des Anfragedokuments hochzuladen (siehe Abbildungen 17.18 und 17.20, Seite 8 und 9 im Antrag)

Kommentare zu Abbildung 17.13 (Zahlen und Pfeile) stellen die Reihenfolge der Aktionen dar: 1 (der Benutzer gibt Daten ein), 2 (der Benutzer bestätigt die eingegebenen Daten) und 3

(es wird eine Erfolgs- oder Fehlermeldung angezeigt, die Datenfelder werden für neue Eingaben freigegeben).

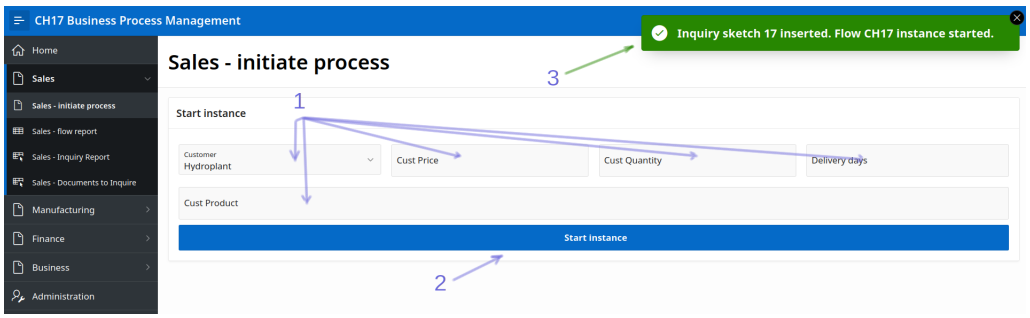



Abbildung 17.13: Verkäufe - initiierter Prozess.

Neben den sichtbaren Komponenten gibt es ein verstecktes Seitenelement (P2_WID), das einfach den nächsten numerischen Wert aus der Sequenz CH17_SEQ_INQUIRY erhält. Das war der Grund, warum wir die Sequenz CH17_SEQ_INQUIRY vor dem Seitendesign definiert haben. Die Schaltfläche „Start“ führt zwei Prozesse aus: „Anfrage hinzufügen“ fügt eine Zeile in die Tabelle CH17_INQUIRY und vier Zeilen in die Tabelle CH17_INQUIRY_DOC ein, während der Prozess „Erstellen und Starten“ die Aktion von Flows for APEX Plug-in ausführt (siehe Details in der verpackten Anwendung, Seite 2, Prozesse). Die Meldung über den erfolgreichen Wettbewerb wird wie in Abbildung 17.13 dargestellt zurückgegeben.

Abbildung 17.14 zeigt einen Bericht über die Verkaufsströme.

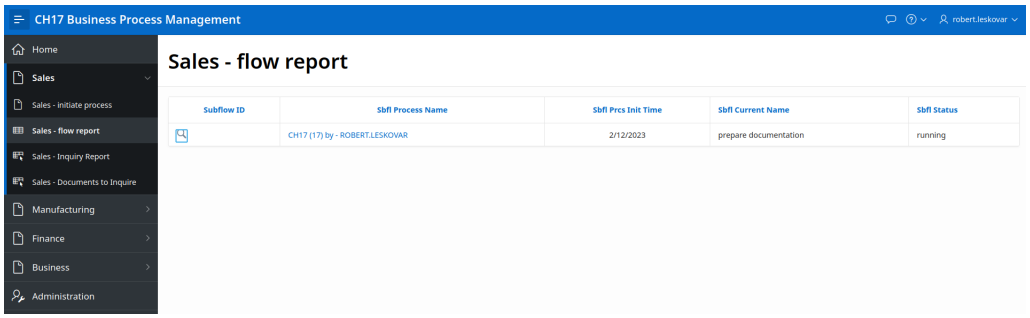



Abbildung 17.14: Umsatz - Flow report.

Der Flow report für den Verkauf ist ein klassischer Bericht auf der Grundlage einer SQL-Abfrage:

```
select sbfl_id, sbfl_prcs_id, sbfl_process_name, sbfl_prcs_init_ts,
sbfl_current_name, sbfl_step_key, sbfl_status
from flow_task_inbox_vw
where sbfl_dgrm_name = 'CH17' and
sbfl_current_name = 'prepare documentation'
```

Es werden nur Instanzen des Prozesses „CH17“ angezeigt, die den aktuellen Namen „prepare documentation“ tragen. Zwei Links sind auf Seite 4 des Antrags zu finden (siehe Abbildung 17.14):

1. Das Linsensymbol in der Spalte „Subflow-ID“ zeigt den Status der Instanz im Workflow-Diagramm an (siehe Abbildung 17.15, Seite 5 in der Anwendung).

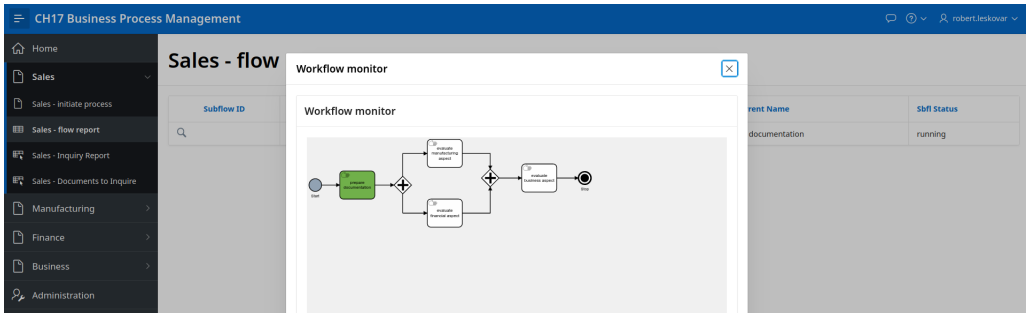



Abbildung 17.15: Umsatz - Flow report für ausgewählte Instanz.

2. Die Spalte „Sbfl Process Name“ zeigt ein modales Dialogfenster, in dem der Vertrieb bestätigen kann, dass die Dokumentation fertig ist (siehe Abbildung 17.16, Seite 7 in der Anwendung).

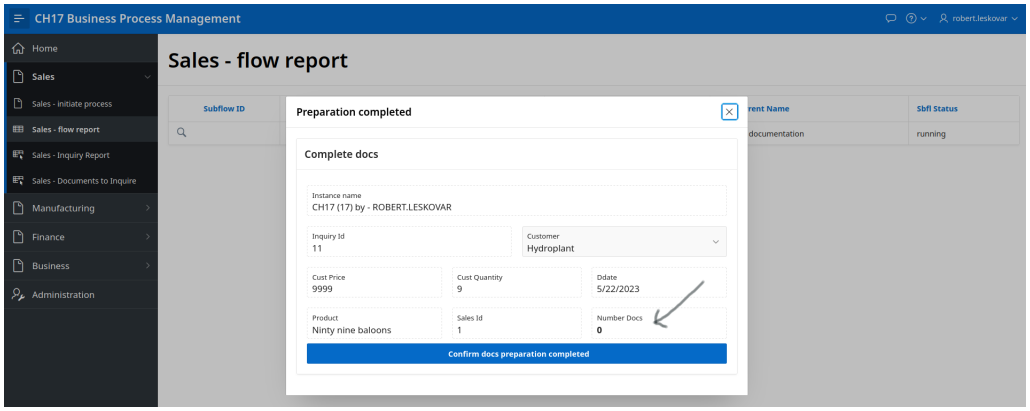



Abbildung 17.16: Verkäufe - Flow report für ausgewählte Instanz.

Zu diesem Zeitpunkt ist die Dokumentation noch nicht vorbereitet (0 Dokumente hochgeladen). Der Zähler ist ein Ergebnis des PL/SQL-Blocks als Quelle:

```
BEGIN
RETURN CH17_COUNT_DOCS(:P7_INQUIRY_ID);
END;
```

Die Funktion CH17_COUNT_DOCS (siehe Details in packaged application Supporting objects oder nach der Installation in Object Browser > Functions) zählt alle großen binären Objekte (BLOB), die in der Tabelle CH17_INQUIRY_DOC gespeichert sind und zu einer bestimmten Anfrage gehören. Wenn die BLOB-Größe größer als Null ist, wird der Inhalt gespeichert, ansonsten ist nur ein leerer Blob vorhanden. Gemäß den Geschäftsregeln müssen mindestens drei Dokumente erstellt werden (für Fertigung, Finanzen und Geschäft).

Um einen Einblick in alle Anfragen zu erhalten, wird ein interaktiver Bericht für den Verkauf erstellt (siehe Abbildung 17.17, Seite 6 in der Anwendung).

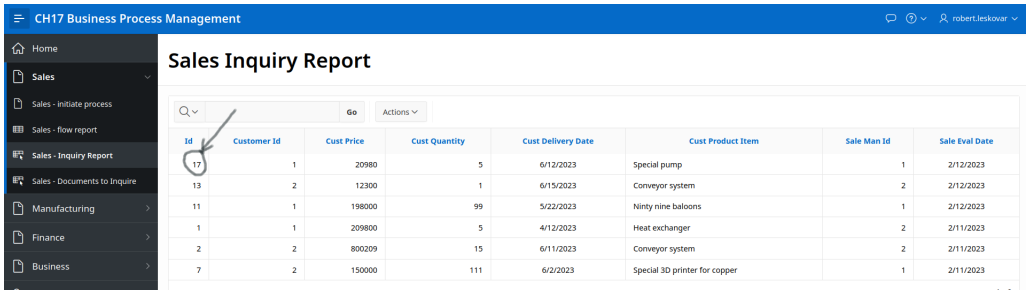



Abbildung 17.17: Verkauf - Anfragebericht.

Der Bericht über Verkaufsanfragen bietet Suchfunktionen und Aktionen (Auswahl von Spalten, Neuordnung, Filterung, Formatierung, Diagrammerstellung, Berichterstattung, Export usw.). Ziel dieses Berichts ist es, dem Vertriebsleiter eine ID der Anfrage zu liefern, die bei der Erstellung der Dokumentation verwendet werden kann (d.h. 17, wie in Abbildung 17.17). Der nächste Schritt für den Vertriebsleiter wäre die Erstellung der technischen, finanziellen, geschäftlichen und sonstigen Spezifikationen (siehe Abbildung 17.18, Seite 8 in der Anwendung).

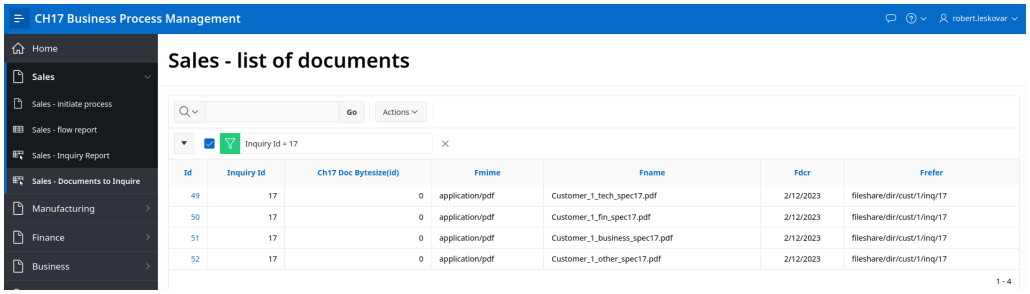



Abbildung 17.18: Verkäufe - Liste der Dokumente.

Abbildung 17.18 zeigt gefilterte Ergebnisse (Anfrage-ID ist 17). Es wird kein BLOB hochgeladen - die Spalte „Ch17 Doc Bytesize(id)“ zeigt alle Nullen. Seite 8 ist ein interaktiver Bericht, der auf der folgenden SQL-Abfrage basiert:

```
select ID, INQUIRY_ID,DOC_CLASS_ID, CH17_DOC_BYTESIZE(ID), CONTBLB,
FMIME,FNAME, FDCR, FREFER
from CH17_INQUIRY_DOC
```

Die PL/SQL-Funktion CH17_DOC_BYTESIZE gibt die Größe des BLOBs zurück. Die Funktion nimmt die ID der CH17_INQUIRY_DOC als Eingabeparameter und ist wie folgt definiert:

```
create or replace function CH17_DOC_BYTESIZE(xdocid in number) return number
is
wdoc_size number := 0;
wblob blob;
Begin
select contblb into wblob from ch17_inquiry_doc where id = xdocid;
wdoc_size := dbms_lob.getlength(wblob);
RETURN  wdoc_size;
END;
```

Die ID der Tabelle CH17_INQUIRY_DOC wird als Link zum Öffnen einer modalen Seite (Seite 9) in der Anwendung verwendet (siehe Abbildung 17.19)).

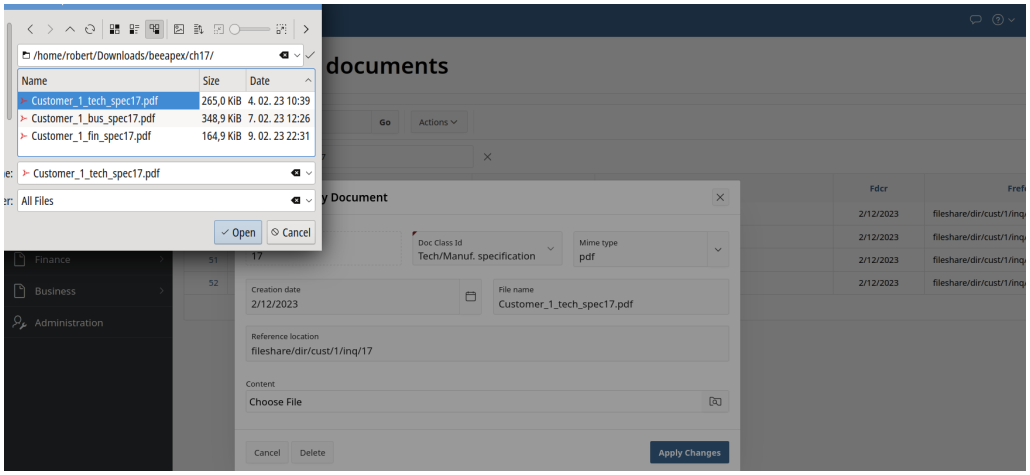



Abbildung 17.19: Verkauf - Dokument für Anfrage 17 hochladen.

Durch Aktualisieren der Seite 8 kann der Nutzer sicher sein, dass die Dokumente hochgeladen wurden (siehe Abbildung 17.20, Seite 8 in der Anwendung).

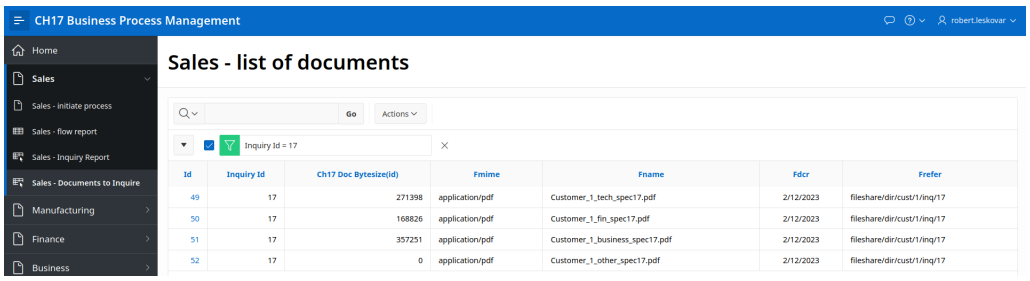



Abbildung 17.20: Verkäufe - Liste der Dokumente nach dem Hochladen und Aktualisieren.

Der Vertrieb würde nun bestätigen, dass die Vorbereitungsaufgabe abgeschlossen ist. Wenn Sie den Umsatzbericht öffnen und auf den Prozessnamen klicken, öffnet sich die modale Seite (siehe Abbildung 17.21, Seite 8 in der Anwendung).

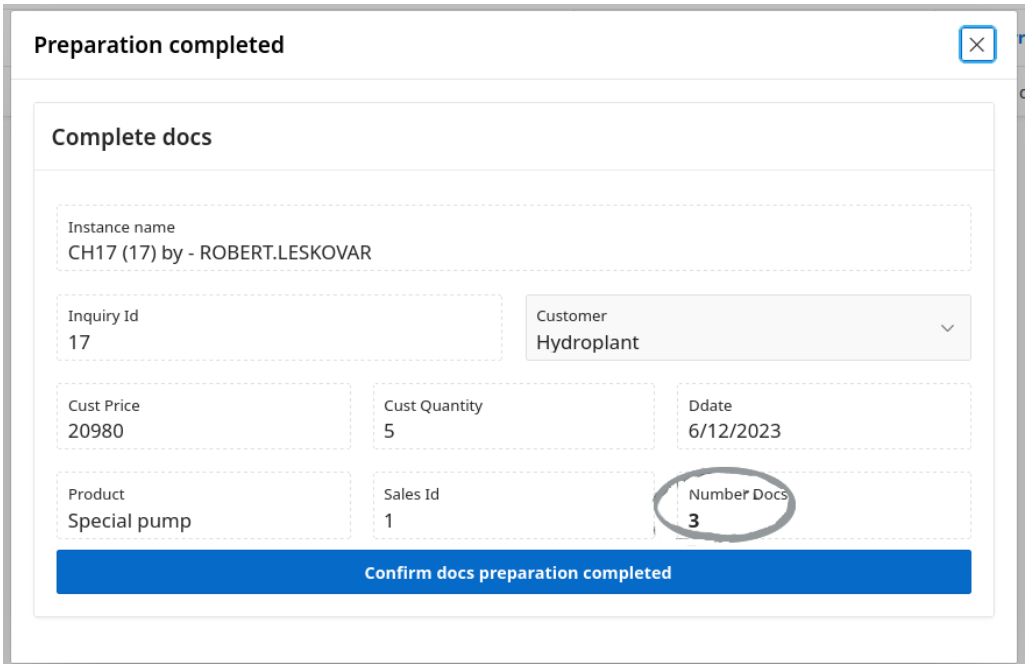



Abbildung 17.21: Umsatz - Flow report für ausgewählte Instanz nach dem Hochladen von drei Belegen.

Durch Klicken auf „Subflow-ID“ erhält der Vertriebsleiter eine visuelle Bestätigung, dass die Aufgabe der Dokumentenerstellung abgeschlossen ist (siehe Abbildung 17.22, Seite 5 in der Anwendung).

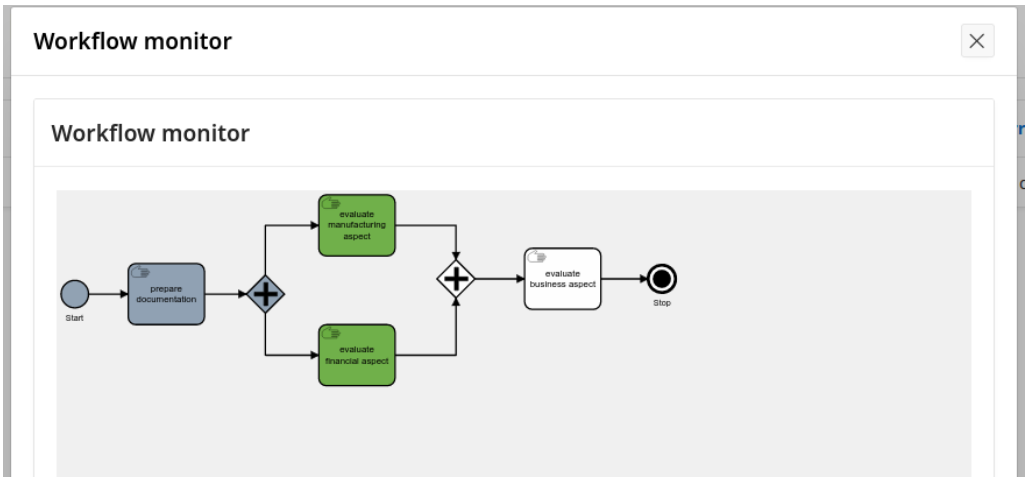



Abbildung 17.22: Umsatz - Flow report für ausgewählte Instanz nach dem Hochladen von drei Belegen.

### 17.6.4 Produktionsleiter

Produktionsleiter können:

- den Arbeitsablauf überwachen (Bericht auf Abbildung 17.23, Seite 20 im Antrag und Diagramm auf Abbildung 17.24, Seite 5 im Antrag)
- Fertigungsbewertungen eingeben (Abbildung 17.25, Seite 21 im Antrag)

Durch Anklicken von „Manufacturing flow report“ kann der Produktionsleiter nicht bewertete Anfragen verfolgen (d.h. Bericht in Abbildung 17.23, Seite 20 in der Anwendung).

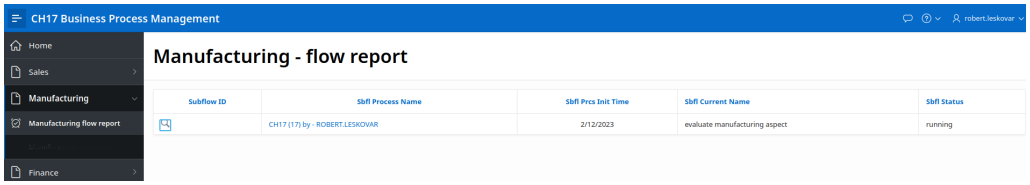



Abbildung 17.23: Fertigung - Flow report.

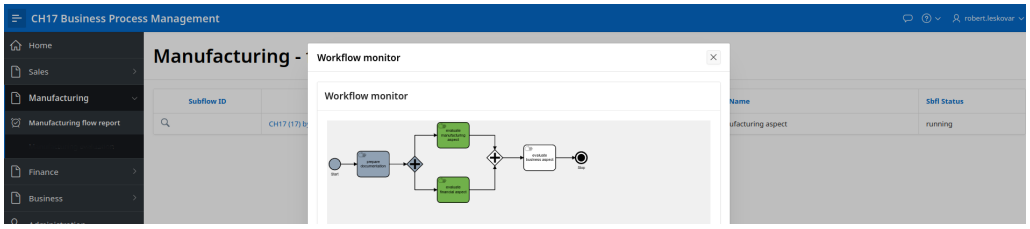



Abbildung 17.24: Fertigung - BPMN-Diagramm - Zustand der Instanz.

Durch Auswahl des Linsensymbols unter „Subflow ID“ wird Seite 5 (siehe Abbildung 17.24) geöffnet, die die Instanz des Flows CH17 zeigt. Die Instanz befindet sich im parallelen Tor, was bedeutet, dass beide Aufgaben (Fertigungs- und Finanzauswertung) abgeschlossen sein müssen, bevor die nächste Aufgabe (Geschäftsauswertung) beginnt. Die Fertigungsbewertung kann durch Klicken auf den Link „Sbfl Prozessname“ auf Seite 20 eingegeben werden (siehe Abbildung 17.25, Seite 21 in der Anwendung).

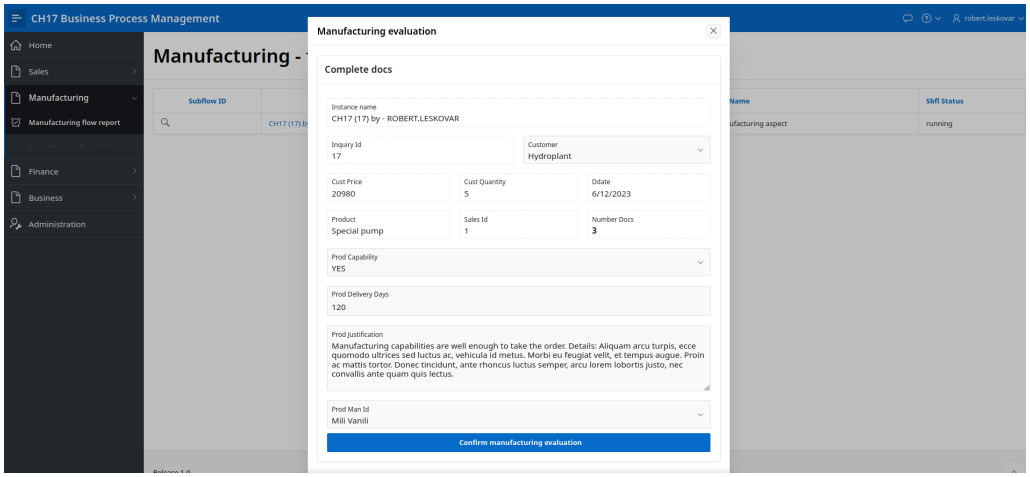



Abbildung 17.25: Bewertung der Herstellung.

Durch Drücken der Schaltfläche „Confirm manufacturing evaluation“ wird der Instance Flow in den nächsten Zustand versetzt, der in Abbildung 17.26 dargestellt ist.

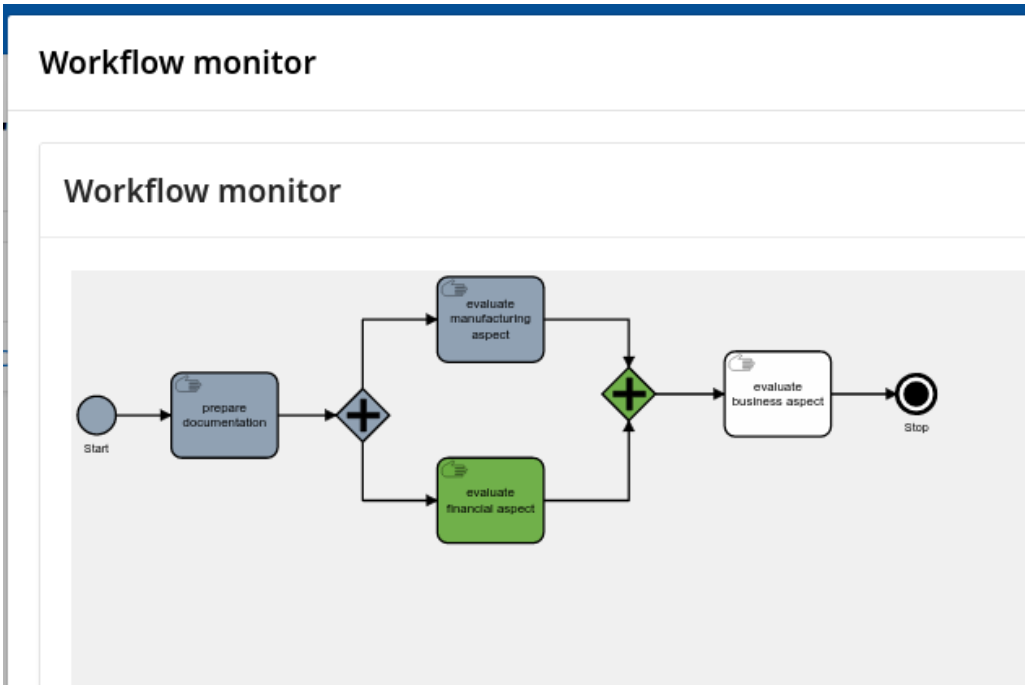



Abbildung 17.26: Fertigungsbewertung abgeschlossen, Instanz wartet auf die finanzielle Bewertung.

Wenn Sie die Seite 20 aktualisieren, zeigt der Bericht „Manufacturing has no flow instances“ an.

### 17.6.5 Finanzverwalter

Finanzmanager können:

- den Arbeitsablauf überwachen (Bericht auf Abbildung 17.27, Seite 30 im Antrag und Diagramm auf Abbildung 17.28, Seite 5 im Antrag)
- finanzielle Bewertungen eingeben (Abbildung 17.29, Seite 31 im Antrag)

Durch Anklicken von „Financial Flow report“ kann der Finanzverwalter nicht bewertete Anfragen verfolgen (d.h. Bericht in Abbildung 17.27, Seite 30 in der Anwendung).

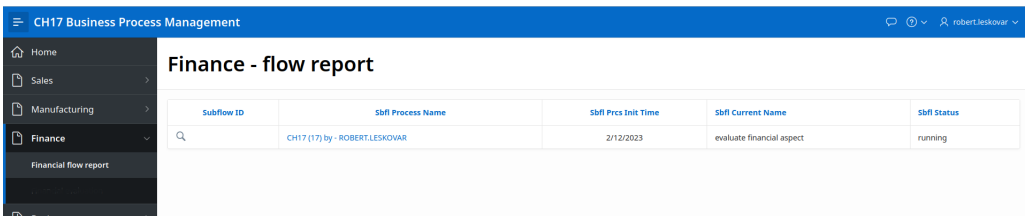



Abbildung 17.27: Finanzen - Flow-Report.

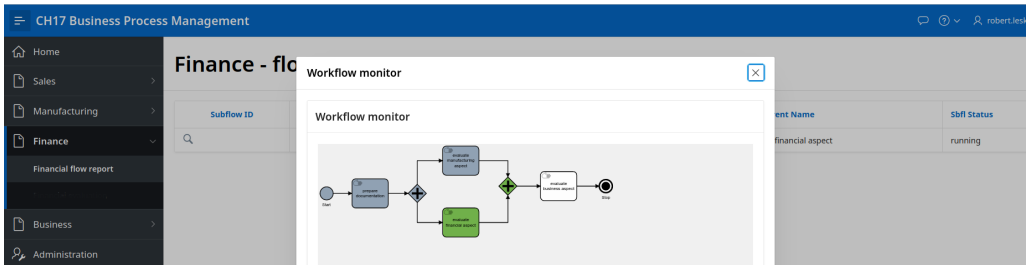



Abbildung 17.28: Finanzen - BPMN-Diagramm - Zustand der Instanz.

Durch Anklicken des Linsensymbols unter „Subflow ID“ öffnet die Anwendung Seite 5 (siehe Abbildung 17.28), die die Instanz des Flows CH17 zeigt. Die finanzielle Bewertung kann durch Anklicken des Links „Sbfl Process Name“ auf Seite 30 eingegeben werden (siehe Abbildung 17.29, Seite 31 in der Anwendung).

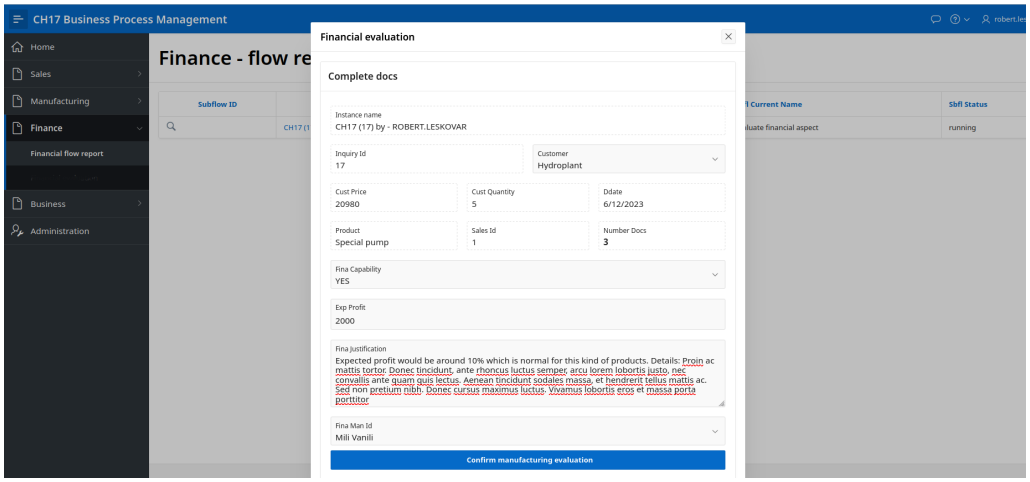



Abbildung 17.29: Finanzielle Bewertung.

Durch Drücken der Schaltfläche „Confirm financial evaluation“ wird der Instance Flow in den nächsten Zustand versetzt, der in Abbildung 17.30 dargestellt ist.

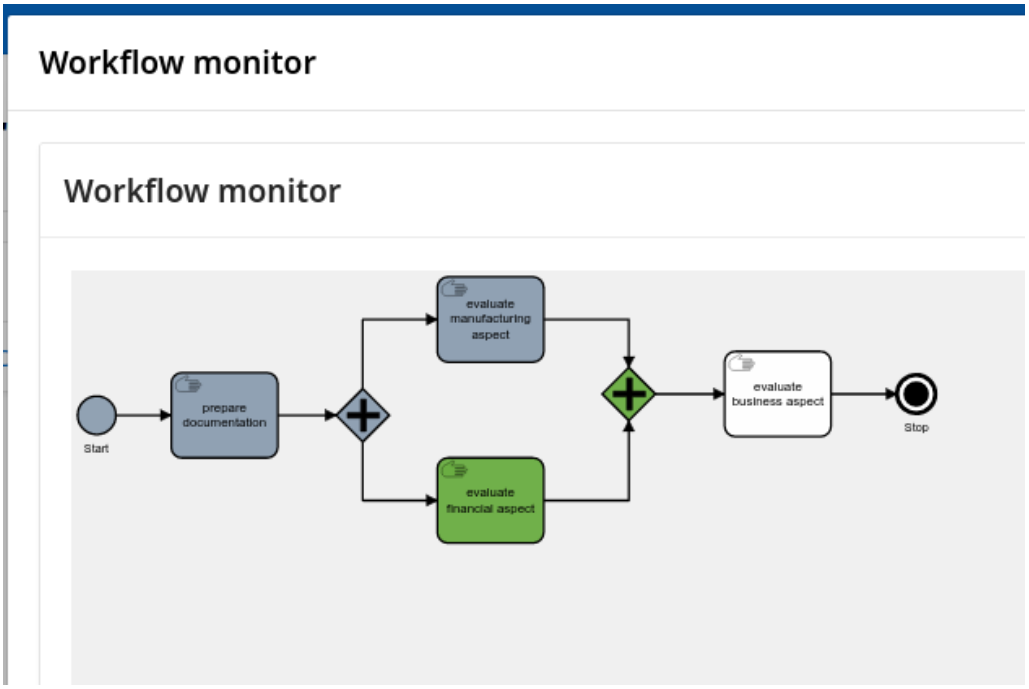



Abbildung 17.30: Finanzielle Bewertung abgeschlossen, Instanz wartet auf die Unternehmensbewertung.

Wenn Sie die Seite 30 aktualisieren, zeigt der Bericht „Finance has no flow instances“ an.

### 17.6.6 Hauptgeschäftsführer - Geschäftsleiter

Geschäftsleiter können:

- den Arbeitsablauf überwachen (Bericht auf Abbildung 17.32, Seite 40 im Antrag und Diagramm auf Abbildung 17.31, Seite 5 im Antrag)
- Unternehmensbewertungen eingeben (Abbildung 17.33, Seite 41 im Antrag)

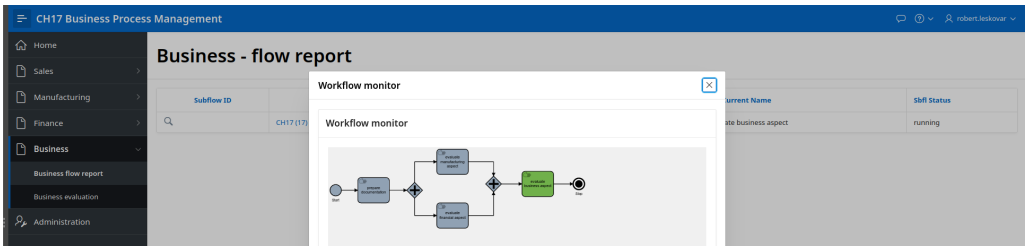



Abbildung 17.31: Finanzen - BPMN-Diagramm - Zustand der Instanz.

Durch Anklicken von „Business Flow-Report“ kann der CEO nicht bewertete Anfragen nachverfolgen (d.h. Bericht in Abbildung 17.32, Seite 40 in der Anwendung).

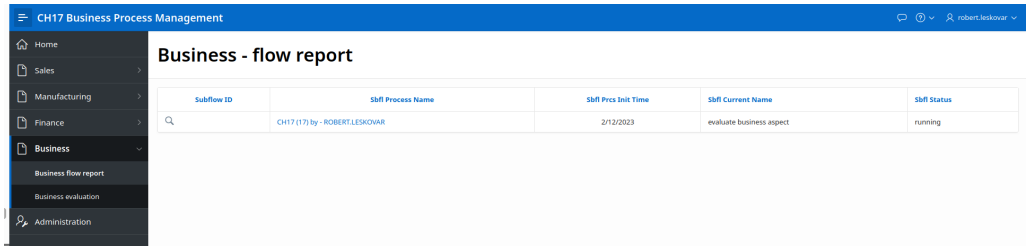



Abbildung 17.32: Business - Flow-Report.

Ein Klick auf das Linsensymbol unter „Subflow ID“ öffnet Seite 5 (siehe Abbildung 17.31), die die Instanz des Flows CH17 zeigt. Die finanzielle Bewertung kann durch Klicken auf den Link „Sbfl Process Name“ auf Seite 40 eingegeben werden (siehe Abbildung 17.33, Seite 41 in der Anwendung).

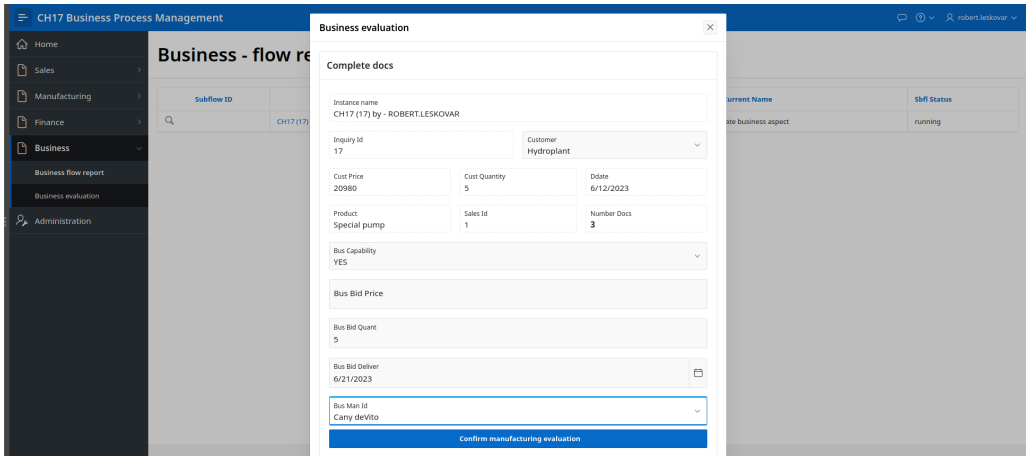



Abbildung 17.33: Unternehmensbewertung.

Durch Drücken der Schaltfläche „Confirm business evaluation“ wird die Instanz an das Ende des Workflows verschoben, wie in Abbildung 17.34 dargestellt.

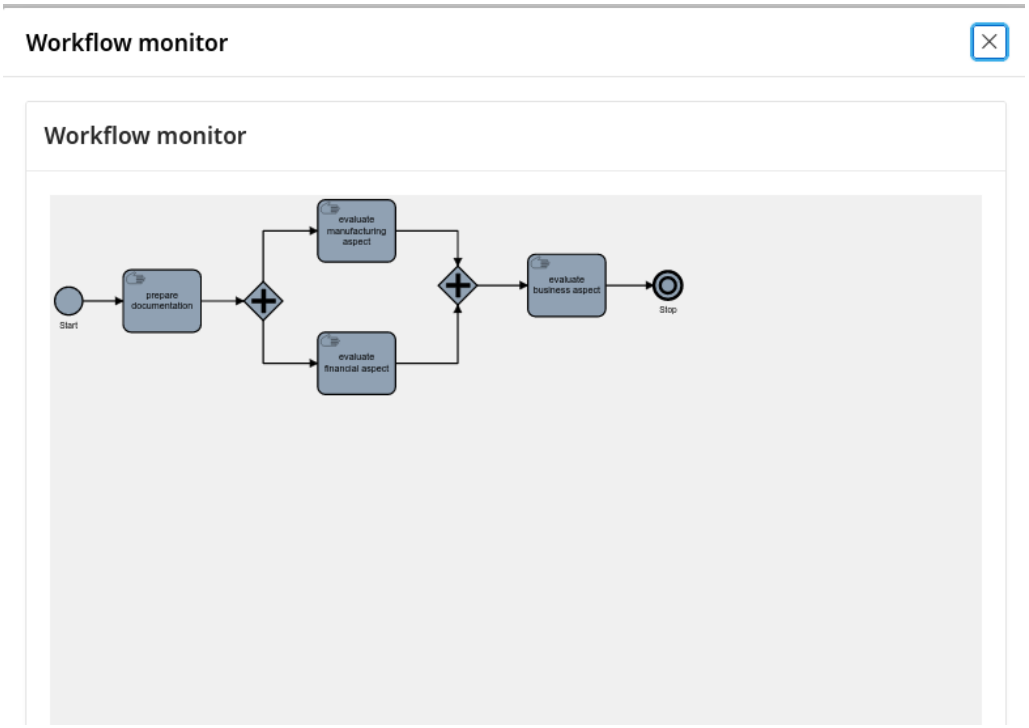



Abbildung 17.34: Unternehmensbewertung abgeschlossen, Instanz beendet.

Wenn Sie die Seite 40 aktualisieren, zeigt der Bericht „Business has no flow instances“ an.

## 17.7 Verknüpfung der Anwendung mit Flows for APEX

Die Anwendung, die in Kapitel 17 entwickelt wurde, verwendet zwei Funktionen, die von Flows for APEX bereitgestellt werden:

- eine Instanz auf der Seite 2 starten
- den Status der Instanz auf Seite 5 anzeigen
- Instanz zur nächsten Aufgabe auf den Seiten 7, 21, 31, 41 vorschieben

Der Start einer Instanz erfordert einen Prozess auf der Seite. Im Identifikationsbereich des Prozesstyps: „Flows for APEX - Flow-Instanz verwalten [Plug-In]“Die Aktion sollte „Erstellen und Starten“ sein. Wir setzen Flow instance Name auf:

```
CH17 (&P2_WID.) by - &APP_USER.
```

Denken Sie daran, dass das Seitenelement P2_WID eine eindeutige Nummer enthält, die von der Sequenz bereitgestellt wird. Der Name ist also eine verkettete Zeichenfolge, die aus der festen Zeichenfolge „CH17“, einem Leerzeichen, einer offenen Klammer, einer eindeutigen Sequenznummer, einer geschlossenen Klammer, einem Leerzeichen, der festen Zeichenfolge „by - “ und dem aktuellen Benutzer besteht. Select Flow verwendet „Static text“ und der statische Text ist „CH17“. Die Erfolgsmeldung lautet „Flow CH17 instance started.“ und die Fehlermeldung „Flow CH17 instance NOT started.“

Um den Zustand der Instanz auf Seite 5 darzustellen, wird der Seitenassistent für die Plug-in-Seite verwendet. Die Seite enthält ein Viewer-Plug-in und ein verstecktes Element - P5_PRCS_ID. Siehe Einstellungen in Abbildung 17.35.

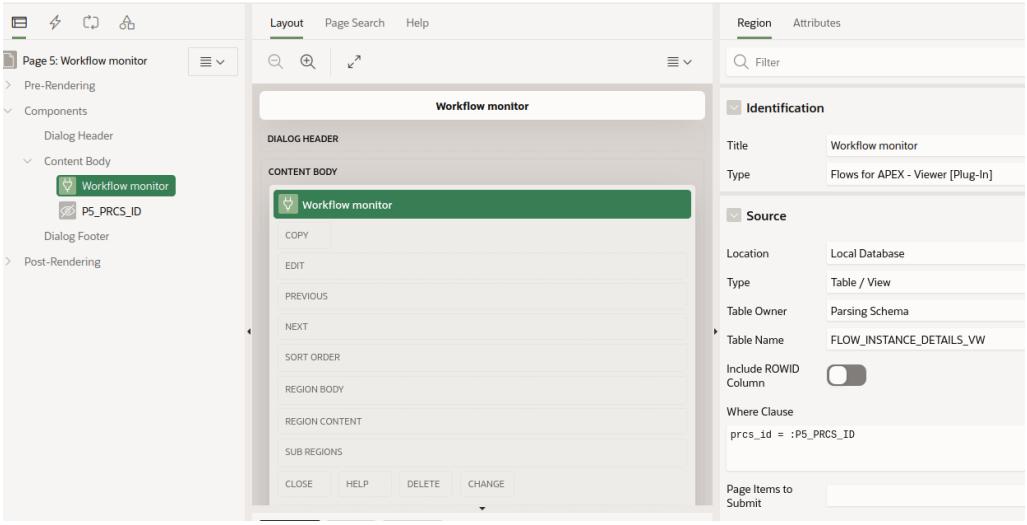



Abbildung 17.35: Zeigt den Zustand der Instanz im BPMN-Diagramm an.

Um die Instanz nach vorne zu schieben (d.h. die Seiten 7, 21, 31, 41), definieren Sie die Seitenelemente wie in Abbildung 17.36 gezeigt.

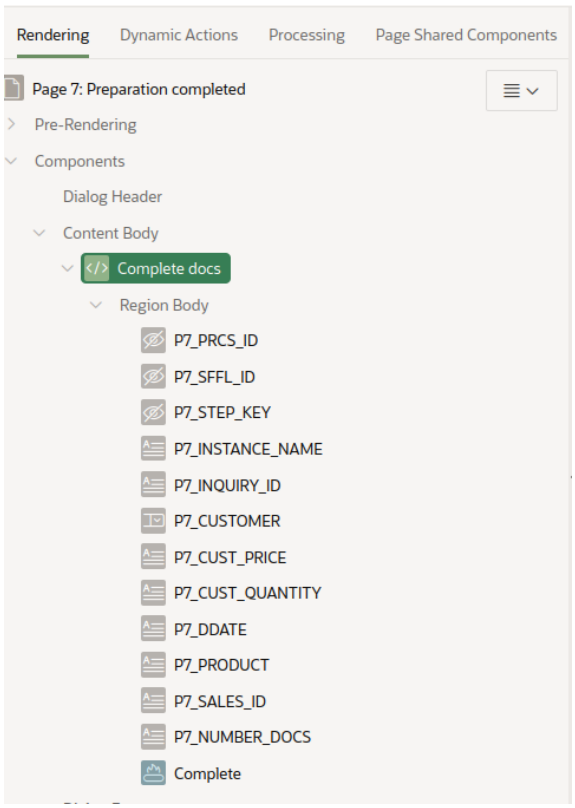



Abbildung 17.36: Einstellen von Seitenelementen.

Nur die ersten vier Seiteneinträge (von P7_PRCS_ID bis P7_INSTANCE_NAME) sind für die Flows for APEX-Integration relevant. Erstellen Sie einen neuen Prozess, der durch eine Schaltfläche ausgelöst wird. Der Prozess wird als „Manage Flow Instance Step“ Plug-in definiert. Legen Sie Aktion, Prozess-ID-Element, Subflow-ID-Element und Schrittschlüssel wie in Abbildung 17.37 dargestellt fest.

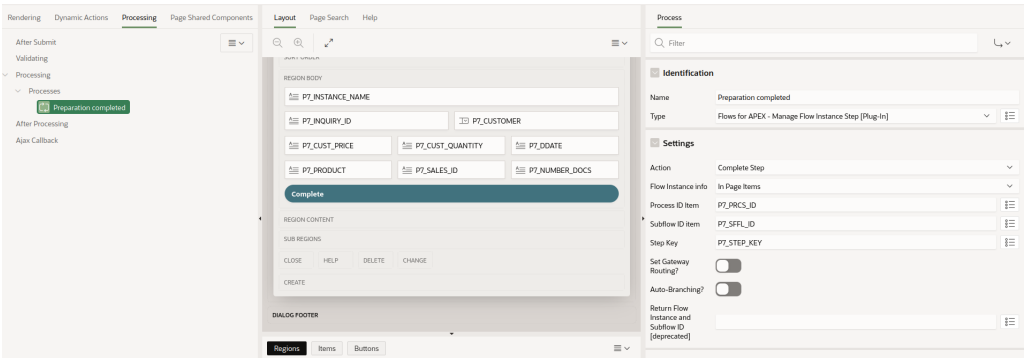



Abbildung 17.37: Schritt in Flows for Apex abschließen.

## 17.8 Benutzerrollen festlegen

Benutzerrollen können in der Application Access Control in Shared Components definiert werden (d.h. Abbildung 17.38).

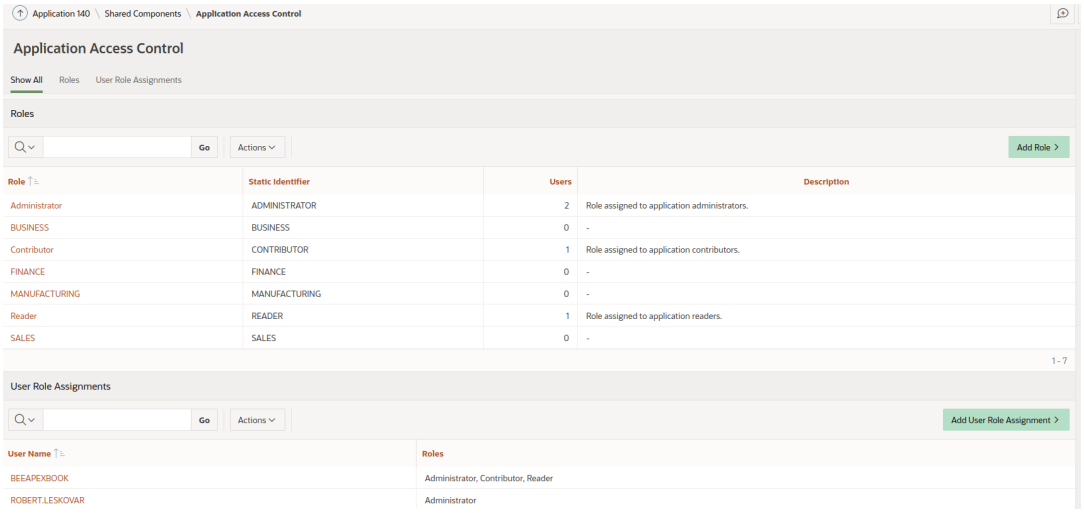



Abbildung 17.38: Definieren Sie Rollen und Benutzerrollen im Menü Application Access Control.

Zu diesem Zeitpunkt werden wir die Details und die Implementierung für diesen speziellen Fall auslassen. Für einen schnellen Überblick siehe Kapitel 13 oder für einen gründlichen Einblick die APEX-Dokumentation über Autorisierung.

## 17.9 Prüfung und Korrektur von Fehlern

Für Anfänger ist der Testprozeß sehr vereinfacht. Wenn Sie Ihre Hände auf Tastatur und Maus legen, um mit APEX zu entwickeln, sind Sie auf dem sicheren Weg, Fehler zu machen und diese zu korrigieren. Haben Sie keine Angst davor, Fehler zu machen. Ziehen Sie die Lehre aus jedem Fehler, den Sie selbst gelöst haben. Teilen Sie das Wissen in der offenen APEX-Gemeinschaft. Die Disziplin des Softwaretestens ist riesig und man kann sie lernen, indem man es tut. Es gibt keinen anderen Weg. Entwickler sind die erste Verteidigungslinie gegen Softwarefehler, -mängel und -ausfälle.

Überraschenderweise gingen gescheiterte Softwareentwicklungsprojekte in den meisten Fällen schief, bevor eine einzige Zeile Code geschrieben wurde. Die Anwendung, die für dieses Kapitel entwickelt wurde, enthält ebenfalls Softwarefehler. Einige von ihnen werden von cleveren Lesern entdeckt und behoben. Einige Fehler liegen auch tief in APEX selbst. Wir, die Autoren und

Entwickler dieses Kapitels, hoffen, dass das Testen und Debuggen für die Leser eine angenehme Herausforderung darstellt.

Die Beseitigung von Fehlern in Software ist manchmal ein Spaß und manchmal ein Fluch. Geben Sie nur nicht zu früh auf!

## 17.10 Ergänzendes Lernmaterial

Sie können das folgende zusätzliche Lernmaterial finden:

- exportierte verpackte Anwendung und Skripte
- Workflow-Modell namens CH17
- Video-Guides

Alle zusätzlichen Lernmaterialien sind auf der öffentlichen BeeAPEX-Projektseite verfügbar. Melden Sie sich als Gastbenutzer an (kein Passwort erforderlich). Das Lehrbuch finden Sie im Abschnitt Books, die Skripte im Ordner Part 1 > Chapter12 im Abschnitt Scripts und die Videoanleitungen in der Sammlung der Videoanleitungen. Material für Kurzkurse finden Sie in der Sektion Short courses.

### 17.10.1 Exportierte Anwendungen

Die exportierte Anwendung wird verpackt. Bei der Installation werden sowohl Tabellen erstellt als auch Daten eingefügt. Bei der Deinstallation werden alle in dieser Anwendung verwendeten Datenbankobjekte entfernt.

Die verpackte Anwendung wird getestet und im neuen Arbeitsbereich ausgeführt, wenn die folgenden Anforderungen erfüllt sind:

- Das Privileg „create job“ wird dem Schema gewährt, das den neuen Arbeitsbereich hostet. Der Datenbankadministrator kann dieses Privileg mit dem Befehl „grant create job to <schema>“ erteilen. Der APEX-Workspace ist mit einem Datenbankobjekt namens Schema verbunden. Wenn Sie OCI verwenden (kostenlos, kostenpflichtig oder von Oracle Academy zur Verfügung gestellt), dann verwenden Sie die Web OCI-Schnittstelle (Autonomous Database > specific instance > Database action) oder gehen Sie direkt in die Web-Datenbankadministrator-Schnittstelle. Sie können Abfragen und Skripte ausführen. Abbildung 17.39 zeigt einen Überblick über die Schnittstelle mit eingegebenem Befehl (1) und Datenbankantwort (2).

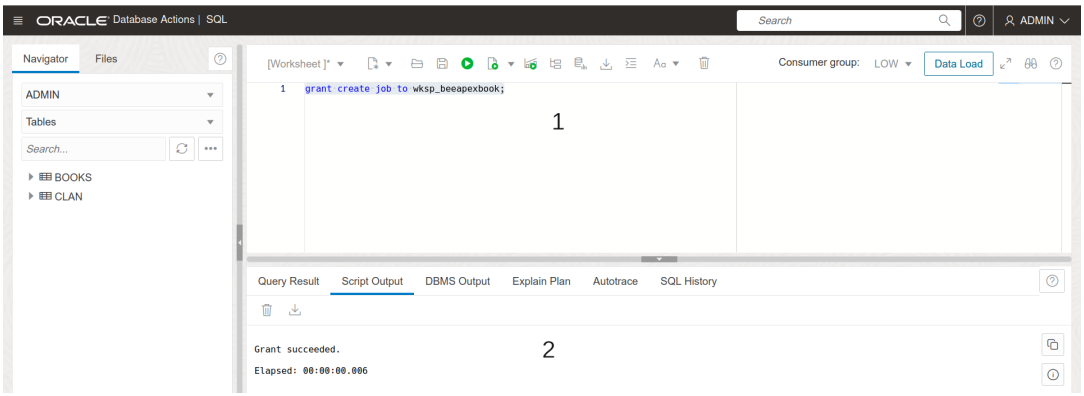



Abbildung 17.39: Erteilung des „Privilegs zum Erstellen von Aufträgen“ an den Arbeitsbereich.

Wenn Sie andere Tools (SQL Developer, TOAD) verwenden, ist das Verfahren bis auf das Aussehen der Schnittstelle weitgehend identisch.

- Flows for APEX muss im neuen Arbeitsbereich installiert werden (siehe vorherige Anweisungen).
- Das Workflow-Modell namens CH17 wird in Flows for APEX importiert. Die Datei *CH17.bpmn*

wird im Verzeichnis Flows4APEX in der verpackten Anwendung als statische Datei gespeichert. Gehen Sie zu Shared Components > Static Application Files und laden Sie sie auf Ihren lokalen Computer herunter. Die gleiche Datei ist auch in den Skripten der Lernmaterialien verfügbar. Dann importieren Sie die Datei *CH17.bpmn* mit Flows for APEX. Melden Sie sich bei der Anwendung Flows for APEX an. Wählen Sie Flow management und importieren Sie die Datei (siehe Abbildungen 17.40 - Schritt 1 und Abbildung 17.41 - Schritt 2).

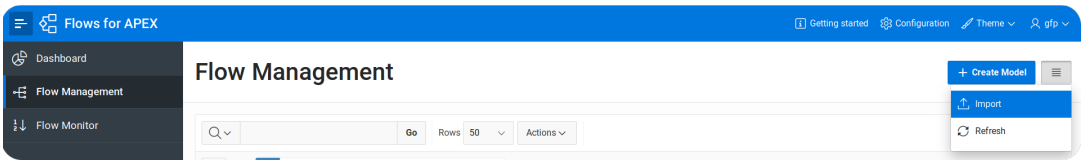



Abbildung 17.40: Importieren Sie den Workflow namens CH17 in Flows for APEX - Schritt 1.

Die Bedeutung der Zahlen in Abbildung 17.41 sind: 1 (beliebige Kategorie), 2 (Name des Modells muss genau CH17 sein), 3 (navigieren Sie in Ihrem Datei-Explorer zu CH17.bpmn) und 4 (bestätigen Sie mit Import)

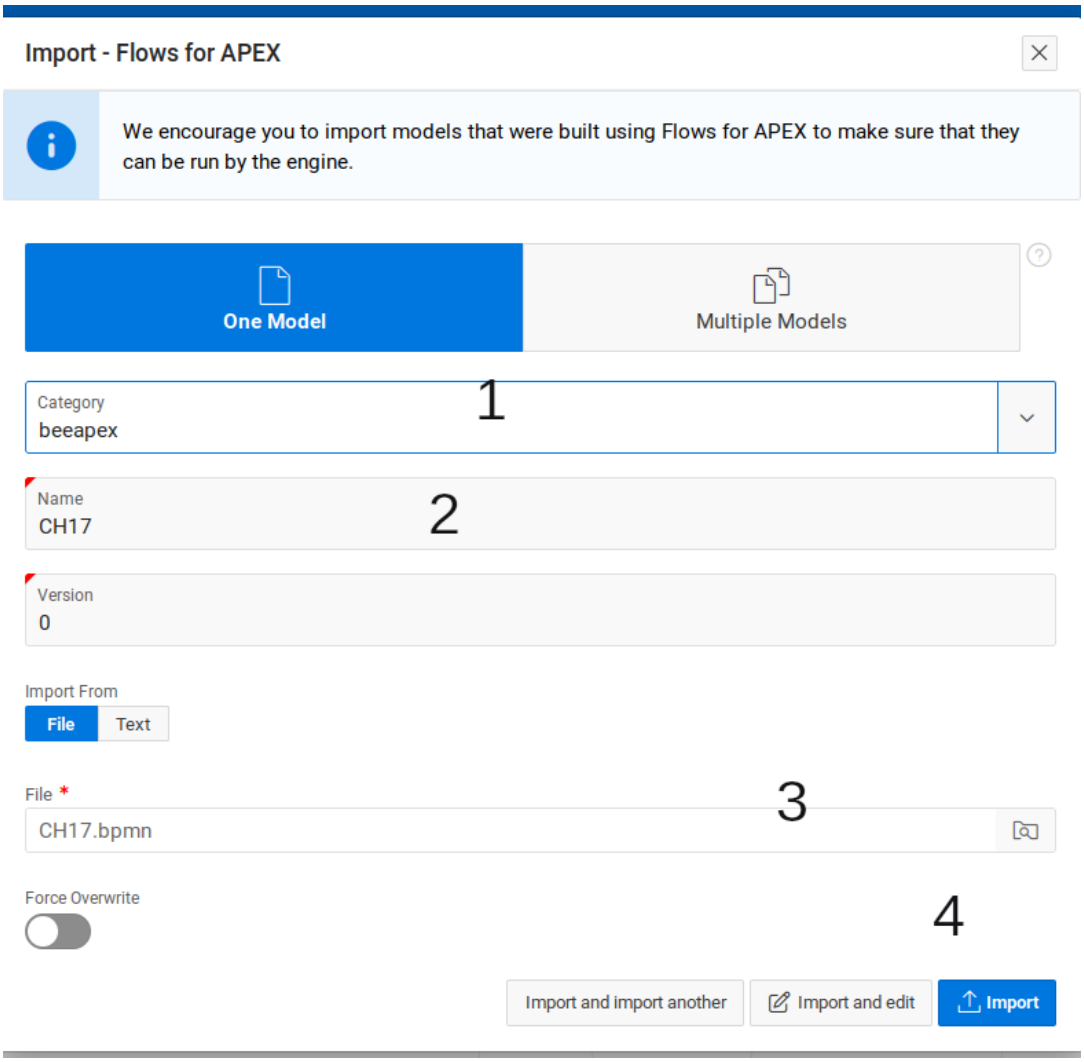



Abbildung 17.41: Importieren Sie den Workflow namens CH17 in Flows for APEX - Schritt 2.

- Fügen Sie nach der Installation der verpackten Anwendung eine Benutzerrollenzuweisung

hinzu (Shared components > Application Access Control > Add User Role Assignment)
Wenn eine der oben genannten Anforderungen nicht erfüllt ist, stürzt die importierte Anwendung ab. Es ist notwendig, das Webbrowser-Cookie zu löschen (z.B. Firefox: Einstellungen > Cookies und Websitedaten > Daten verwalten), wenn die Anwendung aufgrund von nicht erfüllten Anforderungen abstürzt.

### 17.10.2 Video-Anleitungen

Videoanleitungen zeigen jeden Schritt der Anwendungsentwicklung.

## 17.11 Fragen

1. Erklären Sie, was Flows for APEX ist?
2. Was sind die Artefakte eines Geschäftsprozessmodells, das mit einem BPMN-Diagramm dargestellt wird?
3. Welche Funktionen von Flows for APEX wurden in dieser Anwendung verwendet?

## 17.12 Antworten

1. Flows for APEX wird als Feature, als Plug-in und als Anwendung bezeichnet. Alle drei Bedeutungen sind richtig. Feature ist gemeint, wenn wir im Zusammenhang mit der Prozessmodellierung sprechen. Plug-in bezieht sich auf die Entwicklung einer eigenen Anwendung. Die Anwendung „Flows for APEX“ zielt darauf ab, Workflows zu entwerfen und die Instanzen im Falle von Problemen zu verwalten.
2. Die Artefakte eines Geschäftsprozessmodells, das mit einem BPMN-Diagramm dargestellt wird, sind:
   - Start-Ereignis: Zeigt den Startpunkt eines Prozesses an.
   - Aufgabe: Stellt eine einzelne Arbeitseinheit dar, wie z.B. eine Aktivität oder einen Schritt im Prozess.
   - Tor: Stellt einen Entscheidungspunkt im Prozess dar, wie z.B. eine Gabelung im Flow oder eine bedingte Verzweigung.
   - Sequence Flow: Verbindet die Elemente im Diagramm und stellt den Flow der Steuerung von einem Element zum anderen dar.
   - End-Ereignis: Zeigt das Ende eines Prozesses an.
   - Pool/Bahn: Stellt eine Gruppierung zusammengehöriger Aufgaben dar, die dazu dient, die Rollen und Zuständigkeiten der verschiedenen Prozessbeteiligten zu definieren.
   - Nachrichtenfluss: Stellt den Flow von Nachrichten zwischen den Teilnehmern eines Prozesses dar, z.B. eine Kommunikation zwischen zwei Systemen.
   - Datenobjekt: Stellt einen Teil der Daten dar, die in einem Prozess verwendet oder erzeugt werden, wie z.B. eine Rechnung oder ein Kundendatensatz.
   - Datenspeicher: Stellt einen Container zum Speichern von Daten dar, die in einem Prozess verwendet oder erzeugt werden, wie z.B. eine Datenbank oder ein Dateisystem.
3. Um die Instanz zu starten, wurde die Aktion „Create and Start“ verwendet. Um die Instanz im BPMB-Diagramm darzustellen, verwendete der Plug-in-Seiten-Assistent die Viewer-Komponente. Um zum nächsten Schritt überzugehen, wurde in der Anwendung die Komponente „manage Flow Instance Step“ verwendet.



# 18. GreenDi - Tausch von Pflanzen und Saatgut

VJERAN STRAHONJA, DIJANA ORESKI, DARKO ANDROCEC, ANA KUTNJAK

## 18.1 Wirtschaftliche Sicht des Falles

Ein kurzer Geschäftsüberblick über die GreenDi-Plattform wird in den Kapiteln 14 und 15 beschrieben. Kapitel 18 bezieht sich auf den Teil der Plattform, der dem Austausch von Pflanzen und Samen gewidmet ist. Alle registrierten Nutzer der Plattform können an diesem Austausch teilnehmen, indem sie ihr eigenes Angebot präsentieren oder auf das Angebot eines anderen Nutzers antworten. Gleichzeitig soll das System die Suche nach Angeboten über verschiedene Filter, Sortierspalten und ähnliche Funktionen ermöglichen. Darüber hinaus sollte die Kommunikation zwischen dem Bieter und anderen Nutzern in Bezug auf ein bestimmtes Angebot möglich sein. Die Kommunikation erfolgt in Form eines Chats.

## 18.2 Problemstellung

Die folgenden Sätze beschreiben das Problem:

- Die grundlegenden Funktionen des Saat- und Pflanzentauschs sind das Einstellen eigener Angebote (Bieten und Nachfragen) und das Reagieren auf die Angebote anderer Personen (Bieten und Nachfragen), was den Mitgliedern vorbehalten ist.
- Die Mitglieder können den Katalog nach verschiedenen Kriterien durchsuchen und eine Tabelle mit offenen Angeboten für jede Pflanzenart aus dem Katalog öffnen.
- Die Plattform ist nicht kommerziell, d.h. Pflanzen werden auf der Plattform nicht verkauft oder gekauft, sondern getauscht und gespendet. Daher unterliegen die Pflanzentransaktionen nicht der Besteuerung.
- Das Angebot enthält das angebotene oder angefragte Gewächs, die Menge, den Ort, den Bieter, Kommentare, das Angebotsdatum, das Gültigkeitsdatum des Angebots, Anweisungen für die Abholung oder den Versand, den Status usw.
- Die Angebote (Bieten und Nachfragen) werden in Tabellenform dargestellt, wobei die Möglichkeit besteht, nach Spalten zu sortieren und nach verschiedenen Kriterien zu suchen und zu filtern (Art der Pflanze, Datum des Angebots, Mitglied, Ort, Status usw. ...).
- Jedes Mitglied kann auf jedes offene Angebot reagieren.
- Die Konversation über ein Angebot erfolgt in Form eines öffentlichen oder privaten Chats.
- Die Grundeinheit einer Unterhaltung (Chat) ist eine Nachricht.
- Die Nachricht enthält Informationen über das Mitglied, das sie sendet (automatisch generiert),

den Text der Nachricht und die Sichtbarkeit (öffentlich, privat).

- Das System sollte alle Phasen des Lebenszyklusmanagements von Angeboten (Erstellen, Ändern, Schließen, Löschen und Archivieren) unterstützen. Das Angebot wird vom Angebotsinhaber erstellt, geändert, geschlossen, gelöscht und archiviert. Der Administrator kann ein Angebot löschen, wenn es gegen bestimmte Regeln verstößt, und er kann inaktive Angebote nach einer bestimmten Zeit archivieren.

## 18.3 Anwendungsfälle

### 18.3.1 Narrative Beschreibung des Anwendungsfalls

Die Kommunikation mit UC Offer Management steht jedem registrierten Nutzer oder Mitglied der GreenDi-Plattform zur Verfügung. Voraussetzung für die Kommunikation ist ein erfolgreicher Log-In (inklusive UC). Die Funktionalität UC Browse Pflanzenkatalog ist eine Erweiterung des UC Angebotsmanagements. Der Administrator ist auch ein externer Nutzer der UC-Angebotsverwaltung. Gleichzeitig kann er alle Angebote und Chats ohne Einschränkung einsehen und den Status jedes Angebots ändern.

### 18.3.2 Semi-strukturierte Beschreibung

Tabelle 18.1 zeigt UC.

### 18.3.3 Anwendungsfalldiagramm

Die obige Geschichte ist im Anwendungsfalldiagramm 18.1 dargestellt.

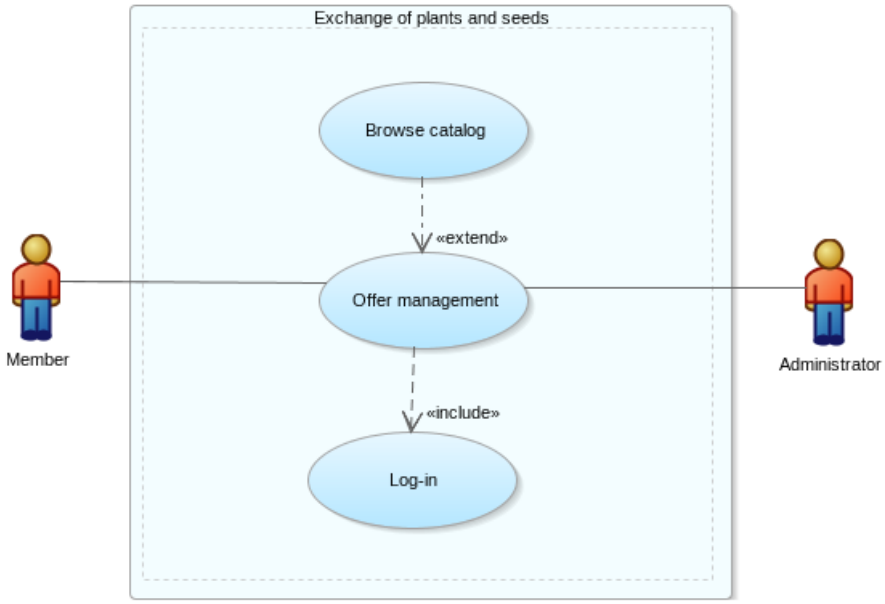



Abbildung 18.1: Anwendungsfalldiagramm - Austausch von Pflanzen und Saatgut

## 18.4 Datenmodell

Dieser Abschnitt erläutert das Datenmodell.

Tabelle 18.1: Beschreibung des Anwendungsfalls: Austausch von Pflanzen und Saatgut.

| **Keyword** | **Wert** |
|---|---|
| ID: | *ch18-01* |
| Titel: | *Austausch von Pflanzen und Saatgut* |
| Beschreibung: | *Austausch von Pflanzen.* |
| Primärer Akteur: | *Jede beliebige Person* |
| Voraussetzungen: | *Grundlegende Informationen über die Anlage können ohne Registrierung gesucht und eingesehen werden.* |
| Nachbedingungen: | *Wenn der Benutzer ein neues Angebot eingibt, werden die neuen Daten gespeichert..* |
| *Haupt* | - |
| Erfolgs-Szenario: | 1. Der Benutzer blättert im Katalog nach oben und unten, bis er eine Pflanze gefunden hat, oder wendet die Filtersuche an<br>2. Der Benutzer öffnet die Liste der Angebote für die ausgewählte Anlage oder alle Anlagen<br>3. Der Benutzer blättert in der Liste der Angebote nach oben und unten oder wendet die Filtersuche an<br>4. Der Benutzer öffnet ein neues Angebot für die ausgewählte Anlage, gibt die Details ein und zeigt das neue Angebot an.<br>5. Der Benutzer klickt auf ein ausgewähltes Angebot und öffnet das Pop-up-Fenster mit den Details des ausgewählten Angebots<br>6. Der Benutzer kann auf jedes aktive Angebot mit einem Gegenangebot oder einer Annahme reagieren oder einen Chat starten.<br>7. Der Chat erfolgt in Form von Nachrichten und Antworten auf Nachrichten<br>8. Der Link ist für den Anbieter und die anderen teilnehmenden Nutzer sichtbar |
| Erweiterungen: | • *Liste der Angebote (Bericht)*<br>• *Einzelheiten zum Angebot (neu oder bereits vorhanden)*<br>• *Chat (Liste der Nachrichten)* |
| Häufigkeit der Nutzung: | *Ungefähres Maximum sind 100 pro Tag.* |
| Status: | *Entwicklungsstand* |
| Eigentümer: | *öffentlich, anonymer Benutzer* |
| Priorität: | *hoch* |

### 18.4.1 Narrative Beschreibung des Datenmodells

Das logische Datenmodell (Entity-Relationship-Modell) ist eigentlich eine Erweiterung des Datenmodells aus den Kapiteln 14 und 15. Die Entitätstypen „Pflanze“ und „Thematische Einheit“ wurden aus Kapitel 14 und „Benutzer“ aus Kapitel 15 übernommen. Die grundlegenden Entitätstypen für den Tausch von Pflanzen und Saatgut sind Angebot und Nachricht. Das Angebot wird durch die folgenden Attribute beschrieben: IDoffer (Identifikationsattribut); Bieten oder Nachfragen (beschreibt, ob die Pflanze oder das Saatgut angeboten oder nachgefragt wird), Menge (der Pflanze oder des Saatguts), Ort (wo die Pflanze oder das Saatgut abgeholt oder geliefert werden kann), Bieter (der Nutzer, der bietet oder nachfragt), Kommentare (Kommentare zur Qualität, Anweisungen zur Lieferung usw.), Datum (Datum des Angebots), Gültig bis (Gültigkeitsdatum), Status (aktiv, zurückgezogen, realisiert). Angebot ist ein Originalangebot oder eine Antwort auf ein Originalangebot. Das bedeutet, dass jedes Angebot eine Antwort auf 0 oder ein Angebot ist UND 0, 1 oder mehr Antworten hat. Jedes Angebot wird von einem und nur einem Nutzer angeboten. 0, 1 oder mehr Nachrichten sind mit dem Angebot verbunden. Jede Nachricht bezieht sich auf 1 und nur 1 Angebot, das von 1 und nur 1 Nutzer (Absender) gesendet wird und für 1 und nur 1 Nutzer (Empfänger) bestimmt ist.

### 18.4.2 Logisches Datenmodell

Das logische Datenmodell wird in Figure 18.2 vorgestellt.

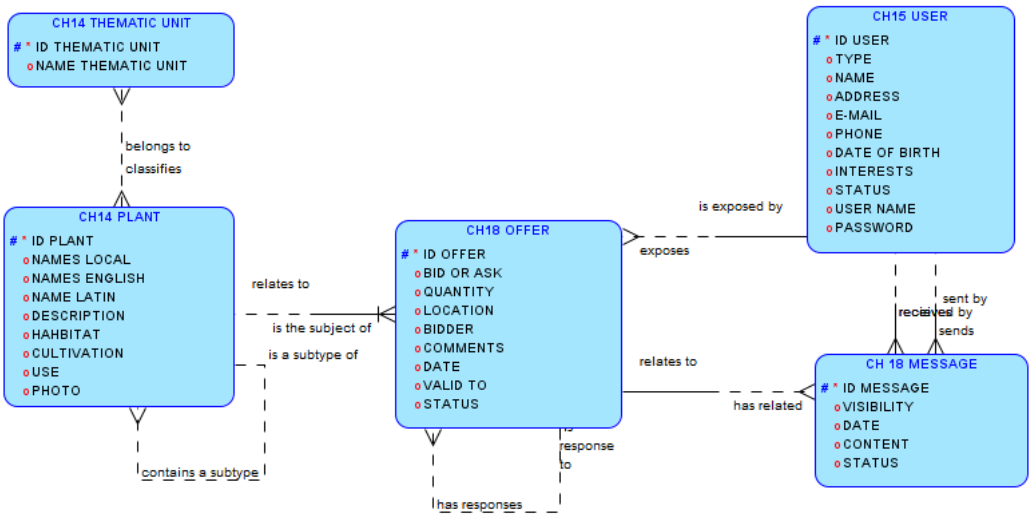



Abbildung 18.2: Logisches Datenmodell.

### 18.4.3 Relationales Datenmodell

Die automatische Umwandlung vom logischen Datenmodell in das relationale Datenmodell in Oracle SQL Data Modeler wird durch die Funktion *Engineering to relational* bereitgestellt. Das Ergebnis wird in Figure 18.3 angezeigt.

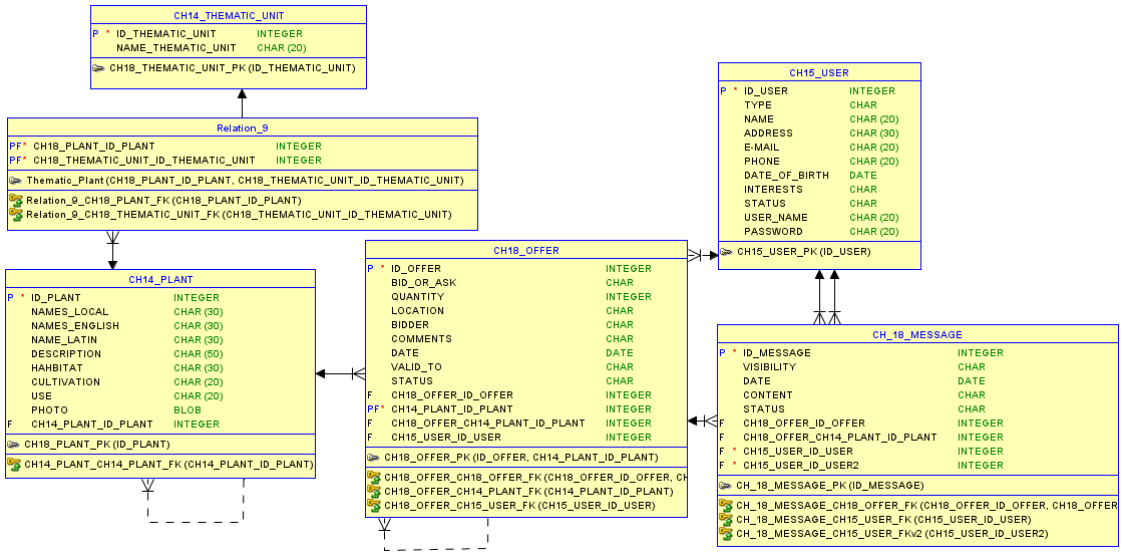



Abbildung 18.3: Relationales Datenmodell.

Lassen Sie nun Oracle SQL Data Modeler ein SQL-Skript generieren. Wählen Sie alle Tabellen im relationalen Modell und verwenden Sie die Funktion *File* > *Export* > *DDL File*, um ein Skript zu erhalten. Wir können das generierte Skript in APEX importieren und es ausführen.

## 18.5 Anwendungsschnittstellen

Die Angebote werden als interaktiver Bericht präsentiert (siehe Figure 18.4).

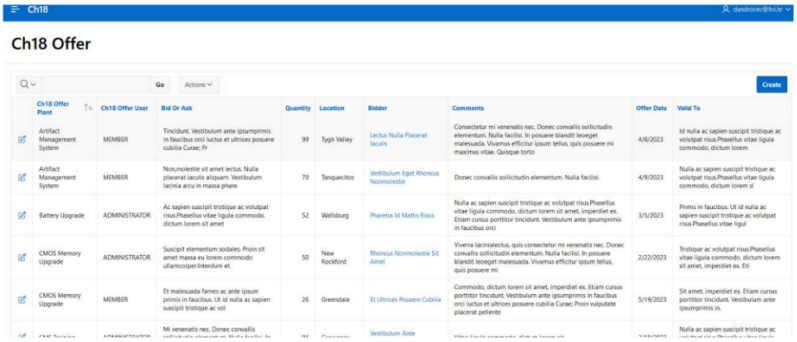



Abbildung 18.4: Angebote - interaktiver Bericht.

Das Nachrichtenformular wird in Figure 18.5 dargestellt.

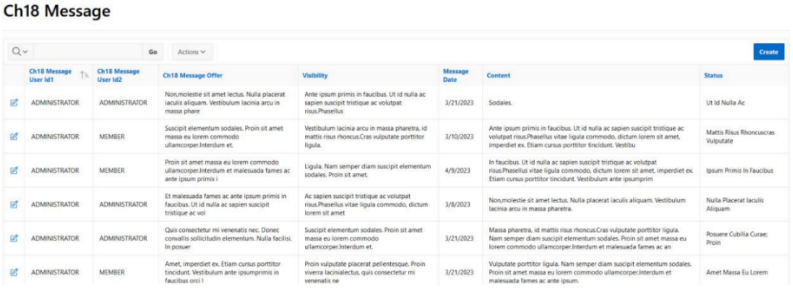



Abbildung 18.5: Formular für Nachrichten.

## 18.6 Ergänzendes Lernmaterial

Sie können das folgende zusätzliche Lernmaterial finden:

- exportierte Anwendung
- Skripte zum Erstellen, Löschen und Einfügen
- Video-Guide

Alle zusätzlichen Lernmaterialien sind auf der öffentlichen BeeAPEX-Projektseite verfügbar. Melden Sie sich als Gastbenutzer an (kein Passwort erforderlich). Das Lehrbuch finden Sie im Abschnitt Books, die Skripte im Ordner Part 2 > Chapter18 im Abschnitt Scripts und die Videoanleitungen in der Sammlung der Videoanleitungen. Material für Kurzkurse finden Sie in der Sektion Short courses.

### 18.6.1 Exportierte Anwendungen

Die exportierte Anwendung wird verpackt. Bei der Installation werden sowohl Tabellen erstellt als auch Daten eingefügt. Bei der Deinstallation werden alle in dieser Anwendung verwendeten Datenbankobjekte entfernt.

### 18.6.2 Video-Anleitungen

Videoanleitungen zeigen jeden Schritt der Anwendungsentwicklung.

## 18.7 Fragen

1. Wie könnten wir das Angebot mit einer Verzögerung sichtbar machen, ab einem bestimmten Zeitpunkt, wenn der Bieter es will?
2. Kann man mit dieser App auch über andere Themen als den eigentlichen Austausch von Pflanzen und Samen chatten? Wie?
3. In der Tabelle CH18_OFFER hat die Spalte VALID_TO den Datentyp CHAR. Welche Probleme können auftreten, wenn ein Datum mit dem Datentyp CHAR definiert ist?

## 18.8 Antworten

1. Zunächst würden wir eine Spalte in der Tabelle CH18_OFFER hinzufügen. Dann würden wir das Formular korrigieren, das die Eingabe neuer Angebote ermöglicht. Schließlich würden wir die Abfrage im Angebotsbericht korrigieren, indem wir eine Bedingung hinzufügen (where new_column <= SYSDATE).
2. Theoretisch ist das möglich. Der Moderator des Chats sollte ein Dummy-Angebot eröffnen, und alle Nutzer, die sich mit ihrem Dummy-Gegenangebot melden, könnten am Chat teilnehmen.
3. Die Speicherung von Daten als CHAR kann zu Problemen mit der Datenintegrität, Sortierproblemen, eingeschränkten Datumsoperationen, Formatierungsschwierigkeiten und einigen anderen Problemen führen. Im Allgemeinen wird empfohlen, die von Oracle bereitgestellten Datentypen wie DATE oder TIMESTAMP zu verwenden, um das Datum genau und effizient zu speichern und zu bearbeiten.



# 19. Verwaltungssystem für Buchrezensionen

A. KUTNJAK, L. HRUSTEK, A. BAGGIA AND R.LESKOVAR

## 19.1 Wirtschaftliche Sicht des Falles

Das System zur Verwaltung von Buchrezensionen wurde mit dem Ziel entwickelt, die Rezensionen von Büchern zu verwalten, die auf einer beliebigen Online-Plattform zum Verkauf angeboten werden. Das Unternehmen (d.h. Amazon) hat die Absicht, den Katalog der verfügbaren Bücher so zu verwalten, dass es auf der Grundlage der Prüfungen der Nutzer über das Angebot der Bücher entscheidet, indem es sie hinzufügt oder entfernt. Mit Hilfe des Rezensionsmanagementsystems sammelt das Unternehmen die Nutzerrezensionen, erstellt eine einzigartige Datenbank und entscheidet auf der Grundlage der gesammelten Daten über das weitere Vorgehen. Die Hauptakteure der Datenbank sind der Administrator und der Nutzer, der Buchleser. Der Anwendungsfall und die Idee wurden von Kaggle inspiriert und angeboten - einer Online-Community-Plattform für Datenwissenschaftler und Enthusiasten des maschinellen Lernens. Die Originaldaten sind unter hier verfügbar. Die Anwendungsfälle und das Datenmodell wurden tiefgreifend modifiziert und ergänzt. Das Ziel dieses Business Cases ist es, die Entwicklungsmöglichkeiten von APEX zu demonstrieren, die nur auf Assistenten basieren. Mit Ausnahme von Quick SQL (insgesamt 31 kurze Zeilen) wird kein weiterer Code vom Entwickler geschrieben.

## 19.2 Problemstellung

Das System zur Verwaltung der Bewertungen verfügt über die folgenden Funktionalitäten:

- In Anbetracht des breiten Spektrums an Büchern im weltweiten Angebot werden Daten zu den Rezensionen und Bewertungen gesammelt.
- Die allgemeine Ansicht der Plattform enthält eine Liste der Bücher mit Informationen zu Titel, Autoren, Inhaltsbeschreibung, Verlag, Erscheinungsdatum, Kategorien und Bewertungen.
- Zu den Optionen des Administrators gehören das Hinzufügen von Büchern zur Datenbank der Plattform sowie deren Bearbeitung und Auswahl nach Kategorien.
- Da die Datenbank das Hinzufügen von Rezensionen durch die Benutzer ermöglicht, kann der Administrator die Rezensionen und Kommentare einsehen. Auf dieser Grundlage trifft der Administrator Entscheidungen über Aktionen pro Buch (Hinzufügen oder Entfernen).
- Der Nutzer hat die Möglichkeit, sich auf der Plattform zu registrieren, um Buchdetails einzusehen, aber auch zu bewerten und Rezensionen zu kommentieren.
- Rezensionen auf der Nutzerseite haben zwei Vorteile. Erstens entscheidet sich der potenzielle

Nutzer aufgrund der Analyse der Buchrezensionen für den Kauf eines Buches. Zweitens trifft das Unternehmen auf der Grundlage der vorhandenen Bewertungen Maßnahmen in Bezug auf die Bücher, d.h. es fügt sie dem Katalog hinzu oder entfernt sie daraus.

## 19.3 Anwendungsfälle

### 19.3.1 Narrative Beschreibung des Anwendungsfalls

Der Administrator ist für die Eingabe oder Änderung von Daten über Bücher innerhalb des Rezensionsmanagementsystems des Unternehmens zuständig und kann verschiedene statistische Indikatoren in Bezug auf Bücher einsehen. Das Amazon-Rezensionsmanagementsystem basiert auf der Registrierung der Nutzer auf der Plattform. Nach der Registrierung können die Nutzer den Katalog der auf der Plattform verfügbaren Bücher einsehen. Der Buchkatalog enthält detaillierte Informationen über die verfügbaren Bücher und bildet die Grundlage für weitere Aktionen innerhalb der Plattform. Die Bücher sind in Kategorien eingeteilt, was die Suche erleichtert und die Eingabe der persönlichen Präferenzen des Nutzers ermöglicht. Registrierte Nutzer können eine Rezension hinzufügen und diese kommentieren.

### 19.3.2 Semi-strukturierte Beschreibung

Aktivitäten zur Eingabe neuer Buchdaten werden dem Administrator zugewiesen. Er ergänzt die aktuelle Datenbank mit neuen Büchern, was die Aufnahme des Angebots in den Buchkatalog zur Folge hat. Die Häufigkeit der Nutzung der Plattform durch den Administrator erfolgt täglich. Der Nutzer registriert sich auf der Plattform des Buchrezensionsmanagementsystems. Nach erfolgreicher Registrierung prüft der Nutzer den Katalog der Bücher und ihre Bewertungen und fügt schließlich seine eigene Bewertung und Kommentare zu den Bewertungen hinzu. Die Plattform wird von den Nutzern mehrmals im Monat genutzt.

Eine Beschreibung der Anwendungsfälle ist in Tabelle 19.1 enthalten.

Die obige Geschichte wird im Anwendungsfalldiagramm (siehe Abbildung 19.1) dargestellt.

Tabelle 19.1: Beschreibung des Anwendungsfalls: Buchrezensionen Analyse

| **Keyword** | **Wert** |
|---|---|
| ID: | *ch19-10* |
| Titel: | *Buchrezensionen Analyse* |
| Beschreibung: | *Das System zur Verwaltung von Buchrezensionen ist nach der Registrierung auf der Plattform verfügbar. Die Plattform ermöglicht es, Kataloge zu durchsuchen, sie zu filtern und nach Kategorien zu klassifizieren.* |
| Primärer Akteur: | *Benutzer* |
| Voraussetzungen: | Die obligatorische Registrierung auf der Plattform ermöglicht die Einsicht in die Buchdaten. |
| Nachbedingungen: | Durchführung weiterer Aktionen innerhalb der Plattform: Hinzufügen oder Entfernen von Büchern (Administrator) und Bewerten und Kommentieren oder Kaufen von Büchern (Nutzer). |
| *Haupt* | Amazon Review Management System |
| Erfolgs-Szenario: | Erstes Szenario (Administrator, neues Buch hinzufügen):<br>1. Der Administrator meldet sich bei der Plattform an.<br>2. Der Administrator wählt das Formular für die Eingabe eines neuen Buches aus.<br>3. Das Eingabeformular enthält Titel, Autoren, Beschreibung, Verlag, Veröffentlichungsdatum und Kategorieauswahl.<br>4. Der Administrator schließt das Verfahren ab, indem er einen neuen Eintrag erstellt.<br>5. Der Administrator meldet sich von der Plattform ab.<br>Zweites Szenario (Benutzer, Buchrezension):<br>1. Der Nutzer registriert sich auf der Plattform.<br>2. Der Nutzer durchsucht den Buchkatalog und wählt das Buch aus, das er bewerten möchte.<br>3. Das Bewertungsformular enthält Titel, Preis, Profilname, Nützlichkeit der Bewertung, Bewertungspunktzahl, Bewertungszusammenfassung und Bewertungstext.<br>4. Der Nutzer schließt das Verfahren ab, indem er eine neue Bewertung eingibt.<br>5. Der Nutzer meldet sich bei der Plattform ab.<br>Drittes Szenario (Nutzer, kommentierende Buchrezension):<br>1. Der Nutzer registriert sich auf der Plattform.<br>2. Der Benutzer durchsucht die Buchrezensionen.<br>3. Die Bewertung enthält Titel, Preis, Profilname, Nützlichkeit der Bewertung, Bewertung, Zusammenfassung und Text der Bewertung.<br>4. Der Nutzer fügt einen neuen Bewertungskommentar hinzu.<br>5. Der Nutzer meldet sich bei der Plattform ab. |
| Erweiterungen: | • *Buchbeilage*<br>• *Interaktiver Bericht pro Buch*<br>• *Buchrezensionen*<br>• *Kommentare zu den Bewertungen* |
| Häufigkeit der Nutzung: | *Täglich, mehrmals im Monat.* |
| Status: | [Entwicklungsstand] |
| Eigentümer: | Potenzielle Nutzer, basierend auf der Registrierung. |
| Priorität: | *hoch* |

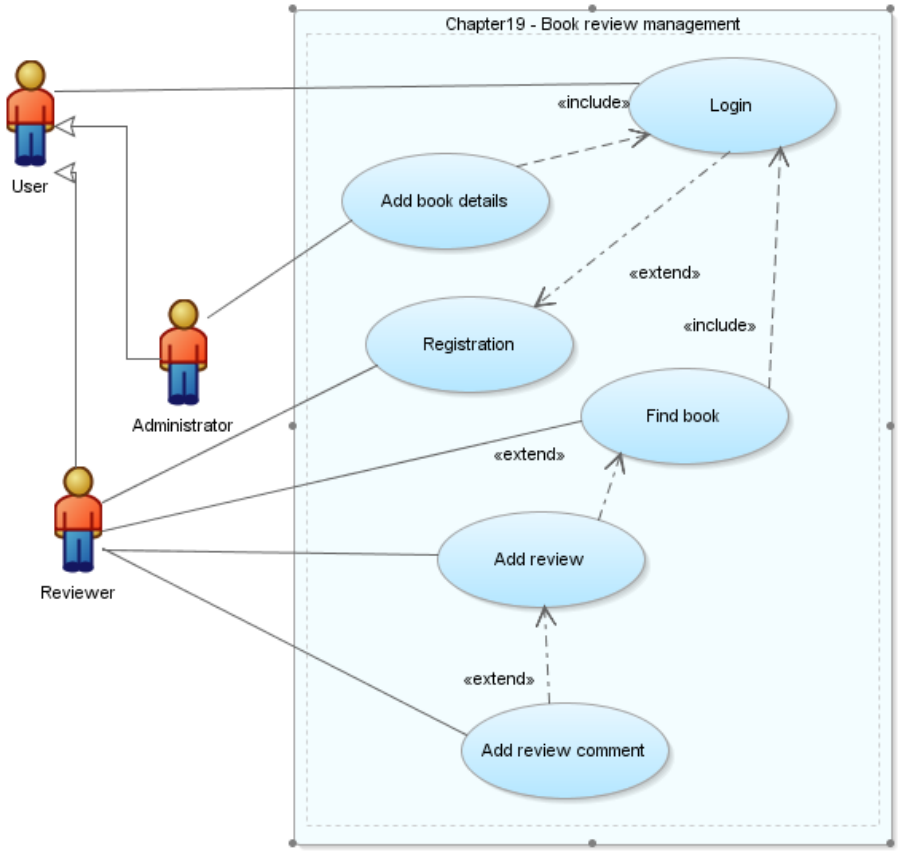



Abbildung 19.1: Anwendungsfalldiagramme: Administrator fügt neuen Kommentar hinzu.

## 19.4 Datenmodell

### 19.4.1 Narrative Beschreibung des Datenmodells

Das logische Datenmodell (Entity-Relationship-Modell) besteht aus mehreren Entitäten: Kategorie, book_data, book_user, book_review und book_review_comment. Die Entität category enthält das Attribut category_name. Die Entität book_data besteht aus mehreren Attributen: Titel, Autoren, About, Verlag, Veröffentlichungsdatum und Anzahl der Bewertungen. Die Entität book_user enthält die Attribute profile_name, firstname und lastname. Entität book_review hat Attribute: review und review_date und review_score. Die Entität review_comment enthält die Antworten auf Rezensionen: rev_comment, rev_helpfulness und rev_comm_score. Jedes Buch gehört zu einer und nur einer Kategorie. Eine Kategorie kann mehrere Bücher enthalten. Es kann mehrere Rezensionen zu einem Buch geben. Ein Benutzer kann mehrere Bücher und mehrere Bewertungen durch andere Benutzer kommentieren. Wir fügen jeder Entität eine primäre UID namens ID hinzu. Diese Attribute werden zu Primärschlüsseln in den entsprechenden Tabellen. Nachdem wir das SQL-Skript für die Tabellenerstellung generiert haben, fügen wir zwei Prüfbeschränkungen hinzu - wir lassen zu, dass die Bewertung von Rezensionen und die Kommentare zur Bewertung von Rezensionen nur numerische Werte zwischen 1 und 5 haben.

Entwickler haben mehrere Möglichkeiten, Tabellen zu erstellen. Eine davon ist, den SQL Developer Data Modeler zu verwenden und ein logisches Modell von Grund auf zu entwerfen, in ein relationales Modell umzuwandeln und ein SQL-Skript zu erzeugen. Eine andere Möglichkeit ist, Tabellen in APEX zu erstellen, Skripte zu exportieren, die Tabellen erstellen, Reverse Engineering

im SQL Develoepr Data Modeler anzuwenden, um ein relationales Modell zu erstellen und es dann in ein logisches Modell zu transformieren. Die Verwendung der APEX Quick SQL Funktionalität ist der einfachste Ansatz. Nachdem Quick SQL geschrieben wurde, kann es mit dem in APEX eingebauten Generator in ein SQL-Skript umgewandelt werden.

### 19.4.2 Logisches Datenmodell

Das logische Datenmodell ist in Abbildung 19.2 dargestellt.

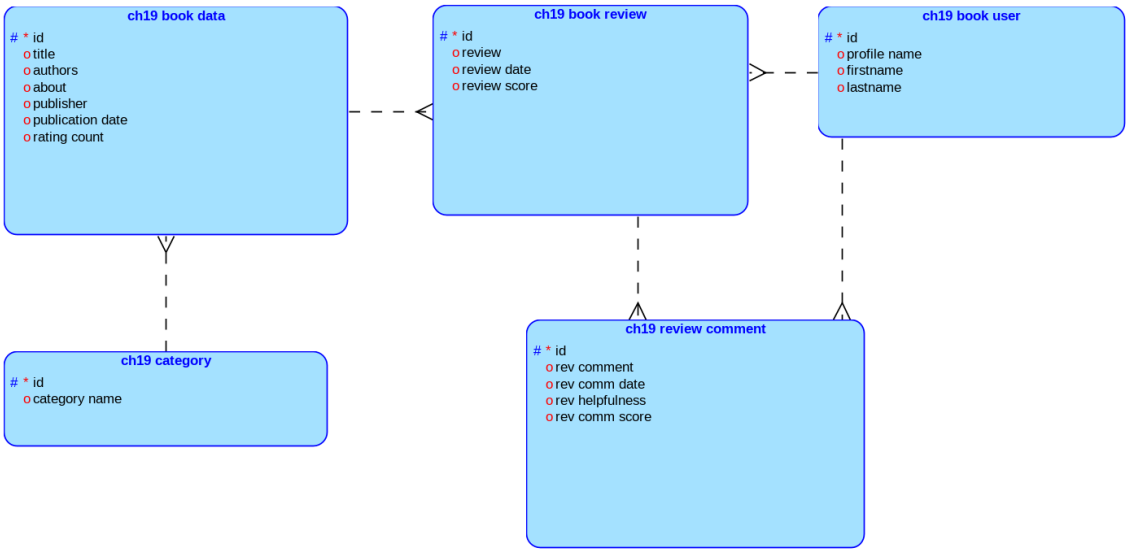



Abbildung 19.2: Logisches Datenmodell.

### 19.4.3 Relationales Datenmodell

Die automatische Umwandlung des zweiten logischen Datenmodells in das relationale Datenmodell in Oracle SQL Data Modeler wird durch die Funktion Engineer to Relational Model ermöglicht. Das Ergebnis ist in Abbildung 19.3 dargestellt.

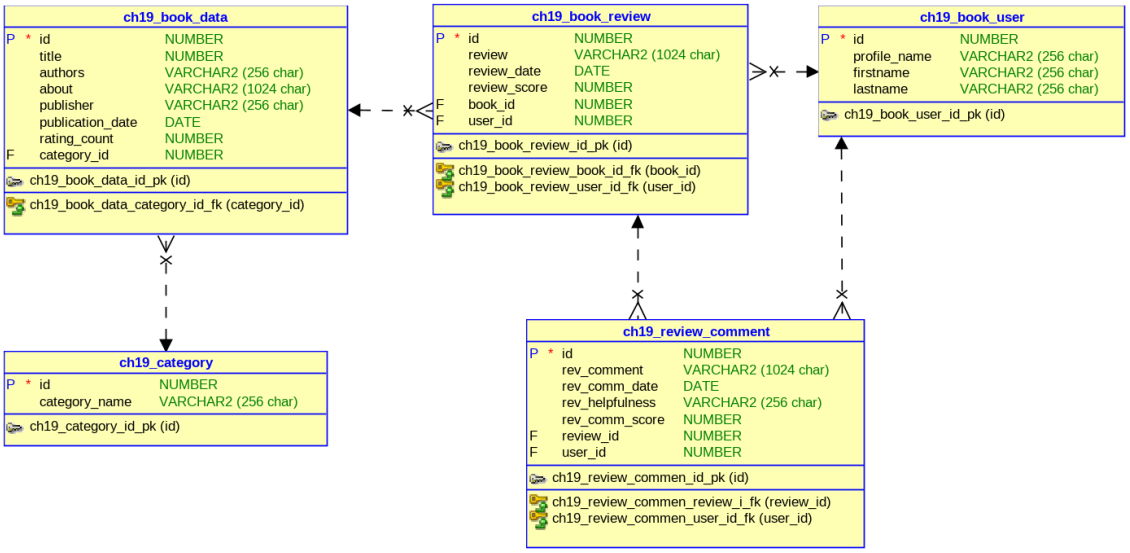



Abbildung 19.3: Relationales Datenmodell.

Lassen Sie nun den Oracle SQL Data Modeler ein SQL-Skript generieren. Wählen Sie alle

Tabellen im relationalen Modell und verwenden Sie die Funktion File > Export > DDL File, um das Skript zu erhalten. Wir können das generierte Skript in APEX importieren und es ausführen.

### 19.4.4 SQL script

Das Skript, das dem relationalen Modell entspricht, enthält die folgenden SQL-Befehle:

```
create table ch19_category (
        id number generated by default on null as identity
        constraint ch19_category_id_pk primary key,
        category_name varchar2(256 char));

create table ch19_book_user (
        id number generated by default on null as identity
        constraint ch19_book_user_id_pk primary key,
        profile_name varchar2(256 char),
        firstname varchar2(256 char),
        lastname varchar2(256 char));

create table ch19_book_data (
        id number generated by default on null as identity
                constraint ch19_book_data_id_pk primary key,
        title varchar2(256 char),
        authors varchar2(256 char),
        about varchar2(1024 char),
        publisher varchar2(256 char),
        publication_date date,
        rating_count number,
        category_id      number
        constraint ch19_book_data_category_id_fk
        references ch19_category on delete cascade);

create table ch19_book_review (
        id number generated by default on null as identity
        constraint ch19_book_review_id_pk primary key,
        review varchar2(1024 char),
        review_date date,
        review_score  number
                constraint ch19_book_revie_review_scor_ck
                check (review_score in (1,2,3,4,5)),
        book_id   number
        constraint ch19_book_review_book_id_fk
        references ch19_book_data on delete cascade,
        user_id   number
                constraint ch19_book_review_user_id_fk
references ch19_book_user on delete cascade);

create table ch19_review_comment (
        id  number generated by default on null as identity
                constraint ch19_review_commen_id_pk primary key,
        rev_comment varchar2(1024 char),
        rev_comm_date date,
```

```
        rev_helpfulness varchar2(256 char),
        rev_comm_score number
                constraint ch19_review_c_rev_comm_scor_ck
                check (rev_comm_score in (1,2,3,4,5)),
        review_id number
                constraint ch19_review_commen_review_i_fk
                references ch19_book_review on delete cascade,
        user_id   number
                constraint ch19_review_commen_user_id_fk
                references ch19_book_user on delete cascade);
```

### 19.4.5 Quick SQL

Die einfachste Methode, ein SQL-Skript zu erzeugen, ist die Verwendung der Quick SQL-Funktion in APEX. Die folgenden Zeilen in Quick SQL ermöglichen die Erzeugung des oben dargestellten SQL-Skripts:

```
ch19_category
        category_name vc256

ch19_book_user
        profile_name vc256
        firstname vc256
        lastname vc256

ch19_book_data
        title vc256
        authors vc256
        about   vc1024
        publisher  vc256
        publication_date date
        rating_count num
        category_id /fk ch19_category

ch19_book_review
        review vc1024
        review_date date
        review_score  num /check 1,2,3,4,5
        book_id /fk ch19_book_data
        user_id /fk ch19_book_user

ch19_review_comment
        rev_comment vc1024
        rev_comm_date date
        rev_helpfulness vc256
        rev_comm_score  num /check 1,2,3,4,5
        review_id /fk ch19_book_review
        user_id /fk ch19_book_user
```

Der Code ist sehr umfangreich. Er erfordert jedoch Grundkenntnisse der Datenmodellierung und der Syntax von Quick SQL.

## 19.5 Anwendungsschnittstellen

### 19.5.1 Administrator

In diesem Fall werden wir die Anwendung aus einem SQL-Skript generieren, das Tabellen erstellt und Beispieldaten einfügt. Suchen Sie die Datei CH19CREATEINSERT.sql im Lernmaterial und importieren Sie sie in Ihren Arbeitsbereich. Öffnen Sie sie nach dem Import und klicken Sie auf Create App (siehe Abbildung 19.4).

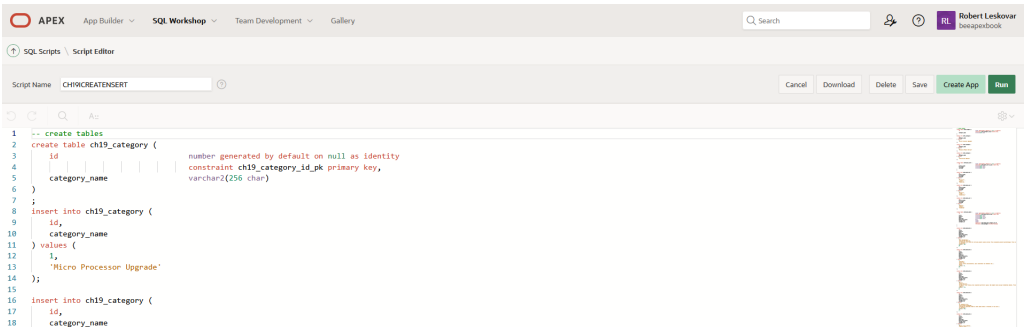



Abbildung 19.4: Erzeugen einer Anwendung aus dem Skript CH19CREATEINSERT - Teil 1.

Benennen Sie die Anwendung, wählen Sie alle Funktionen aus und klicken Sie auf Generate Application (siehe Abbildung 19.5).

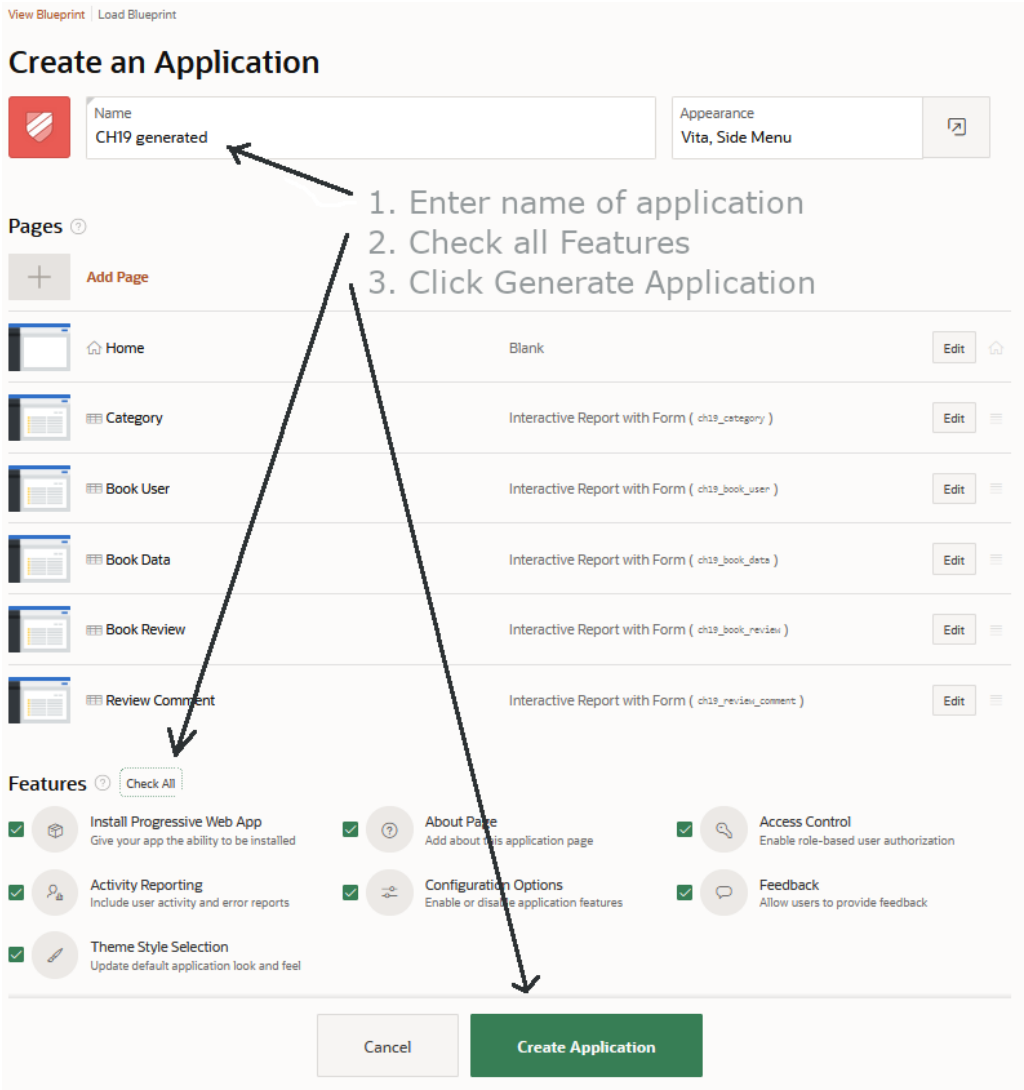



Abbildung 19.5: Erzeugen einer Anwendung aus dem Skript CH19CREATEINSERT - Teil 2.

Ein funktionierender Prototyp der Anwendung ist nun vorbereitet. Von nun an werden die Funktionen für den Administrator und den Endbenutzer vorgestellt. Bis jetzt wurde keine einzige Zeile Code vom Entwickler geschrieben (mit Ausnahme von Quick SQL) und keine Anwendungsschnittstelle wurde vom Entwickler angepasst. Der eigentliche Code hinter der Anwendung wurde vollständig von APEX-Assistenten zusammengestellt.

Abbildung 19.6 zeigt das Formular für die Eingabe eines neuen Buches durch den Administrator.

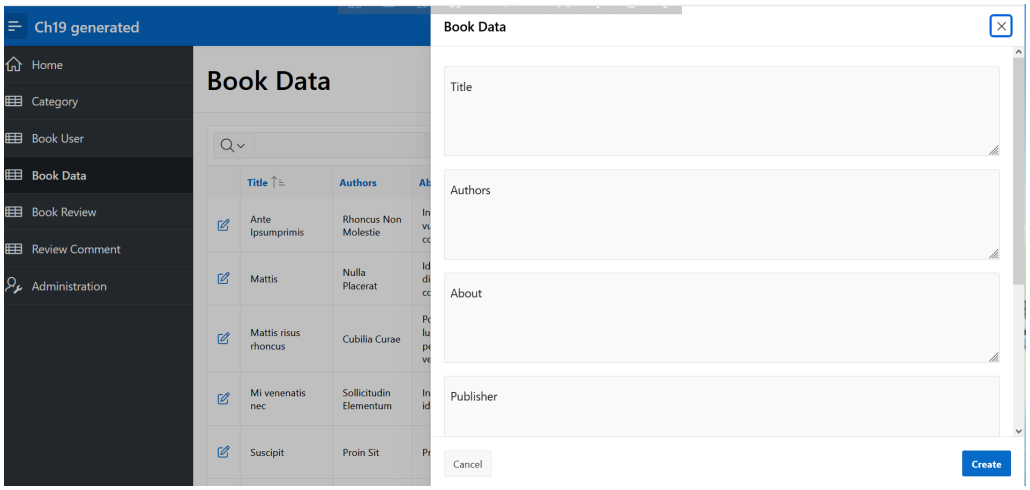



Abbildung 19.6: Hinzufügen eines Buches durch den Administrator.

Abbildung 19.7 zeigt das Formular zum Einfügen der Buchkategorie durch den Administrator.

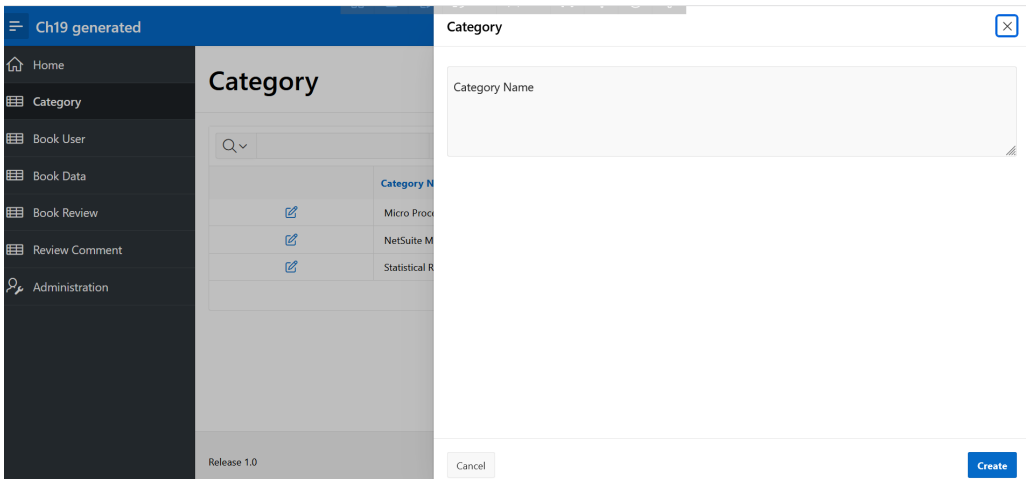



Abbildung 19.7: Hinzufügen einer Kategorie durch den Administrator.

### 19.5.2 Nutzer

Abbildung 19.8 zeigt das Benutzerregistrierungsformular.

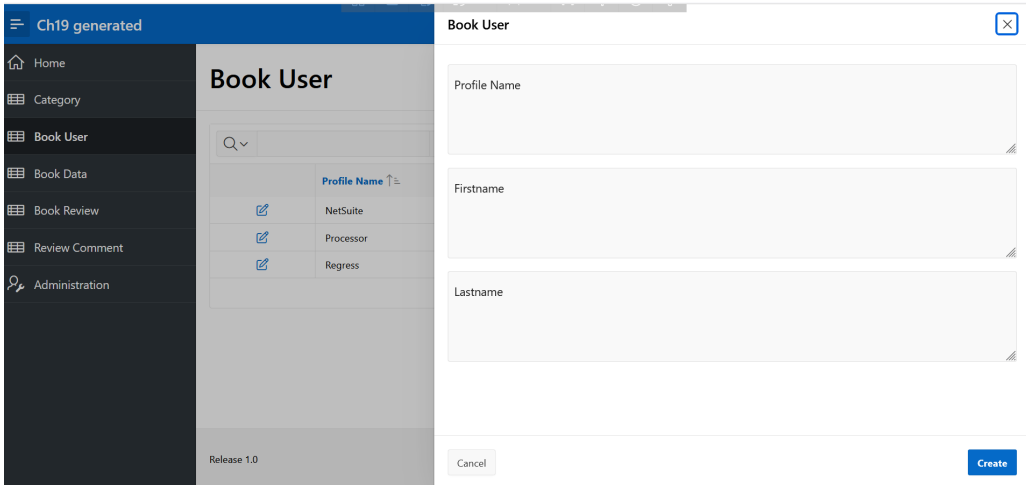



Abbildung 19.8: Benutzerregistrierung.

Abbildung 19.9 zeigt das Durchsuchen und Hinzufügen von Bewertungen.

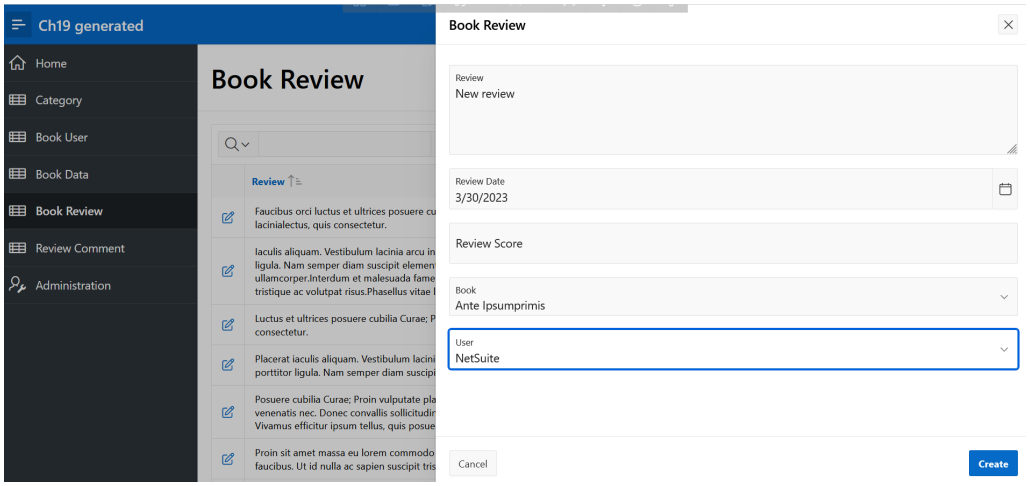



Abbildung 19.9: Durchsuchen und Hinzufügen von Bewertungen.

Abbildung 19.10 zeigt das Formular zum Kommentieren einer Bewertung.

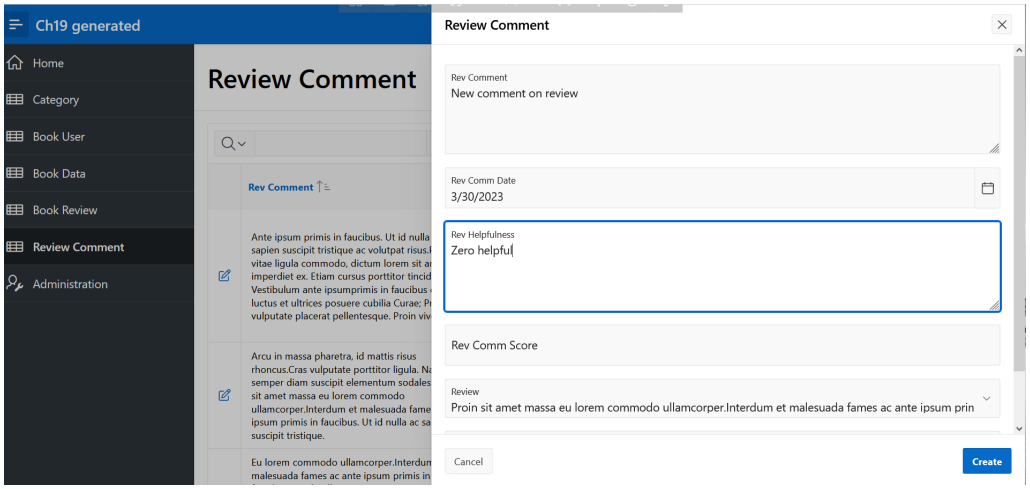



Abbildung 19.10: Formular zum Kommentieren einer Bewertung

Die Berichte sind standardmäßig tabellarisch. Der Endbenutzer kann mit dem eingebauten Assistenten ohne Programmierung Diagramme erstellen. Siehe Kapitel 8 zur Umwandlung von Berichten in Diagramme.

## 19.6 Benutzerrollen definieren

Benutzerrollen können in der Application Access Control in Shared Components definiert werden. Zu diesem Zeitpunkt werden wir die Details und die Implementierung für diesen speziellen Fall auslassen. Für einen schnellen Überblick siehe Kapitel 13 oder für einen tieferen Einblick in die APEX-Dokumentation über Autorisierung.

## 19.7 Ergänzendes Lernmaterial

Sie können das folgende zusätzliche Lernmaterial finden:

- Skript zum Erstellen und Auffüllen von Tabellen
- Skript zum Löschen von Tabellen
- exportierte verpackte Anwendung
- Video, das zeigt, wie man eine Anwendung aus einem Skript erstellt

Alle zusätzlichen Lernmaterialien sind auf der öffentlichen BeeAPEX-Projektseite verfügbar. Melden Sie sich als Gastbenutzer an (kein Passwort erforderlich). Das Lehrbuch finden Sie im Abschnitt Books, die Skripte im Ordner Part 1 > Chapter12 im Abschnitt Scripts und die Videoanleitungen in der Sammlung der Videoanleitungen. Material für Kurzkurse finden Sie in der Sektion Short courses.

### 19.7.1 Exportierte Anwendungen

Die exportierte Anwendung ist verpackt. Bei der Installation werden sowohl Tabellen erstellt als auch Daten eingefügt. Bei der Deinstallation werden alle in dieser Anwendung verwendeten Datenbankobjekte entfernt.

Die verpackte Anwendung wird getestet und im neuen Arbeitsbereich ausgeführt, wenn die folgenden Anforderungen erfüllt sind:

- Fügen Sie den APEX-Benutzer hinzu, bevor Sie die Anwendung starten. Nur im Entwicklungs- und Testarbeitsbereich navigieren Sie zu Shared Components > Application Access Control > Add User Role Assignment; geben Sie den APEX-Benutzer ein und legen Sie für diesen Benutzer die Rollen Administrator, Contributer und Reader fest. In der Produktion ist die Rücksprache mit einem Fachmann vor dem Einsatz ein Muss.

Wenn dem Benutzer keine entsprechende Rolle zugewiesen wird, stürzt die importierte Anwendung ab. Es ist notwendig, das Webbrowser-Cookie zu löschen (z.B. Firefox: Einstellungen > Cookies und Websitedaten > Daten verwalten), wenn die Anwendung aufgrund von nicht erfüllten Anforderungen abstürzt.

### 19.7.2 Video-Anleitungen

Videoanleitungen zeigen jeden Schritt der Anwendungsentwicklung.

## 19.8 Fragen

1. Warum wurde in diesem Fall ein Quick SQL-Skript verwendet?
2. Wie können Sie mehrere Formulare und Berichte auf einen Schlag erstellen?
3. Wie kann man den Benutzer der APEX-Anwendung in der Anwendung selbst autorisieren?

## 19.9 Antworten

1. Quick SQL-Skript wurde eingeführt, um die Entwicklung des Datenmodells zu beschleunigen. Quick-SQL-Skript erzeugt auf Anfrage ein SQL-Skript. Das SQL-Skript ist ausführbar, d.h. es erstellt Datenbankobjekte wie Tabellen und Ansichten.
2. Es ist möglich, mehrere Formulare und Berichte in einem Zug zu erstellen, indem man ein Skript schreibt, das die Definition mehrerer Tabellen enthält.
3. Um APEX-Anwendungsbenutzer in der Anwendung selbst zu autorisieren, muss die Anwendung mit einer Zugriffskontrollfunktion erstellt werden. Der autorisierte Benutzer (Administrator) kann dann andere Benutzer mit den Rollen Administrator, Contributor und Reader hinzufügen.



# 20. Stückliste und Kostenberechnung

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA

## 20.1 Wirtschaftliche Sicht des Falles

Das beobachtete kleine Unternehmen, das den Namen **OSC** tragen wird (10-20 Mitarbeiter), stellt Kabelkonfektionen, Kabelbündel, Leiter und Signallampen her. Sie bieten auch die Montage von elektromechanischen Teilen und mechanischen Halbprodukten an. Ihre Kunden sind größere Unternehmen, die Haushaltsgeräte, Messgeräte, Steuerungssysteme, Stromzähler, medizinische Geräte und ähnliches anbieten. Kundenbestellungen werden per Telefon und E-Mail entgegengenommen. Bei einfachen Produkten und Vorgängen sowie bei sich wiederholenden Aufträgen sendet **OSC** dem Kunden sehr schnell eine Bestätigung. In letzter Zeit häufen sich die Aufträge für mehrstufig zusammengesetzte Produkte und komplexere Halbfabrikate. Außerdem gibt es Dutzende von Positionen in einem Auftrag, und jede Position kann ein Produkt enthalten, das aus mehreren Halbprodukten und Materialien besteht. Halbfabrikate können aus Halbfabrikaten und Werkstoffen zusammengesetzt sein. Der Kunde möchte schnell ein Angebot erhalten. Die Erstellung einer Preiskalkulation für das Material und die Halbprodukte (Stücklistenkalkulation oder kurz BOM-Kalkulation) ist eine anspruchsvolle, fehleranfällige und zeitaufwändige Aufgabe für mehrstufig zusammengesetzte Produkte und Halbprodukte. Es kann mehrere Tage dauern, bis dem Kunden ein Angebot unterbreitet wird. Die Kalkulation der Arbeitskosten (Arbeitsgänge) ist eine weitere Kalkulation, die aus Gründen der Kürze des Kapitels ausgelassen wird. Der Umfang des Kapitels beschränkt sich daher auf die Stücklistenkalkulation. Der Firmeninhaber äußerte die folgenden Bedürfnisse:

- der Bericht über Produkte, Halbfabrikate, Material und eventuell Kundenanfragen
- der Bericht über die Materialliste
- ein Formular zur Bearbeitung von Produkten, Halbprodukten, Material und ggf. Kundenanfragen
- ein Formular für die Bearbeitung der Materialliste
- die Berechnung und baumartige Darstellung von Stücklisten

Eine Anwendung sollte auf einem Telefon, Tablet oder Desktop-Computer nutzbar sein.

## 20.2 Problemstellung

Der Unternehmensleiter steht vor mehreren Problemen im Zusammenhang mit der agilen Reaktion:

- Die Berechnung ist zeitaufwendig, fehleranfällig und anspruchsvoll, muss aber sehr schnell

vorbereitet werden.
- Überschreitungen und Kalkulationsfehler können sich auf verschiedene Weise auf ein Unternehmen auswirken, z.B. durch den Verlust potenzieller Kunden, geringere finanzielle Ergebnisse, einen schlechteren Ruf des Unternehmens, geringere Umsätze und Füllungen und ähnliches.

Die oben genannten Risiken sollten durch eine Webanwendung wirksam gemildert werden.

## 20.3 Anwendungsfälle

### 20.3.1 Narrative Beschreibung

Wir können die grundlegenden Aufgaben bestimmen, die zu erfüllen sind:
- Daten über Produkte, Halbprodukte und Materialien melden und pflegen. Wir behandeln das Konsumieren als Bericht in tabellarischer und Baumform und das Pflegen als Hinzufügen, Aktualisieren und Löschen
- Berechnung der Stückliste für eine bestimmte Anfrage, ein Produkt oder ein Halbprodukt.

Die Titel der Anwendungsfälle lauten daher „Grunddaten melden und pflegen“, „Strukturdaten melden und pflegen“ und „Stückliste berechnen“.

### 20.3.2 Semi-strukturierte Beschreibung

Die obige Geschichte wird in einem Anwendungsfalldiagramm dargestellt.

### 20.3.3 Anwendungsfalldiagramm

Die obige Geschichte ist im Anwendungsfalldiagramm dargestellt (siehe 20.1).

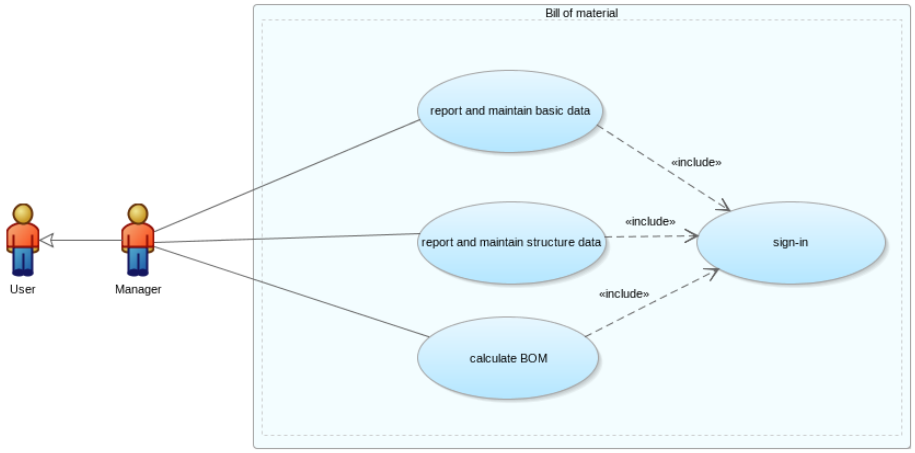



Abbildung 20.1: Anwendungsfalldiagramm.

## 20.4 Data model

### 20.4.1 Narrative Beschreibung des Datenmodells

Für diesen Fall gibt es nur zwei Entitäten im Datenmodell. Die Entität **ch20 psm** (später als Tabelle bezeichnet) beschreibt Produkte, Halbprodukte, Materialien und Kundenanfragen mit nur drei gemeinsamen Attributen: ID, Beschreibung, Mengeneinheit und Preis pro Mengeneinheit. Die Werte für alle Attribute sind erforderlich. Warum eigentlich? Es ist sinnlos, eine Instanz zu speichern, ohne zu wissen, wie sie bezeichnet wird, wie sie gemessen wird und wie hoch der Preis pro Einheit ist. Die ID ist eine eindeutige Zahl ohne Dezimalstellen. Die Beschreibung muss

Tabelle 20.1: Beschreibung des Anwendungsfalls: Bericht und Pflege von Basisdaten.

| **Keyword** | **Wert** |
|---|---|
| ID: | *Ch20-01* |
| Titel: | *Bericht und Pflege von Basisdaten* |
| Beschreibung: | *Der Manager verwendet die APEX-Anwendung, um grundlegende Daten zu melden und zu pflegen (auswählen, aktualisieren), unabhängig davon, ob es sich um ein Produkt, ein Halbprodukt, ein Material oder eine Anfrage handelt. Materialien und Aggregate werden mit demselben Satz von Attributen beschrieben. Eine Aktualisierung der Preise von Materialien führt nicht zu einer Neuberechnung der Preise der Aggregate, in denen das Material verwendet wird. Eine Kundenanfrage kann sich aus mehreren Produkten und Halbprodukten zusammensetzen. Die Mengeneinheit ist in diesem Fall das Stück.* |
| Primärer Akteur: | *Manager* |
| Voraussetzungen: | *Der Manager hat ein Konto für die APEX-Anwendung.* |
| Nachbedingungen: | *Nach der Meldung oder Pflege von Daten kann eine weitere Aktion durchgeführt werden, einschließlich der Stücklistenberechnung.* |
| *Haupt* | *Szenarien* |
| Erfolgs-Szenario: | 1. *Öffnen Sie den Webbrowser und melden Sie sich bei der Anwendung an.*<br>2. *Menüpunkt oder Seitennavigation „Prod./Semi/Material“ auswählen*<br>3. *Zum Bearbeiten blättern Sie auf der Seite nach unten oder filtern Sie die Daten, um ein bestimmtes Produkt, Halbprodukt oder Material zu finden.*<br>4. *Klicken Sie auf das Stiftsymbol für ein bestimmtes Produkt, Halbprodukt oder Material und aktualisieren Sie die Daten.*<br>5. *Klicken Sie auf „Apply changes“.* |
| Erweiterungen: | *keine* |
| Häufigkeit der Nutzung: | *durchschnittlich 5 pro Tag* |
| Status: | *Fertiggestellt* |
| Eigentümer: | *Manager* |
| Priorität: | *hoch* |

Tabelle 20.2: Beschreibung des Anwendungsfalls: Strukturdaten melden und pflegen.

| **Keyword** | **Wert** |
|---|---|
| ID: | *Ch20-02* |
| Titel: | *Strukturdaten melden und pflegen* |
| Beschreibung: | *Der Manager verwendet die APEX-Anwendung, um die Struktur der Produkte, Halbprodukte und Materialien zu melden (ausgewählte Daten anzeigen) und zu pflegen (einfügen und löschen). Eine Kundenanfrage kann sich aus mehreren Produkten und Halbprodukten zusammensetzen. Das Einfügen neuer Artikel oder das Löschen bestehender Artikel löst keine (Neu-)Berechnung der Preise der Aggregate aus.* |
| Primärer Akteur: | *Manager* |
| Voraussetzungen: | *Manager hat ein Konto für die APEX-Instanz.* |
| Nachbedingungen: | *Nach der Meldung oder Pflege von Daten kann eine weitere Aktion durchgeführt werden, einschließlich der Stücklistenberechnung..* |
| *Haupt* | *Szenarien* |
| Erfolgs-Szenario: | 1. *Öffnen Sie den Webbrowser und melden Sie sich bei der Anwendung an.*<br>2. *Menüpunkt oder Seitennavigation „Bill-of-material“ auswählen*<br>3. *Um eine Komponente hinzuzufügen, klicken Sie auf die Schaltfläche „Create“.*<br>4. *Um ein Element in der Struktur zu löschen, klicken Sie auf „Delete“.* |
| Erweiterungen: | *keine* |
| Häufigkeit der Nutzung: | *durchschnittlich 30 Mal pro Monat* |
| Status: | *Fertiggestellt* |
| Eigentümer: | *Manager* |
| Priorität: | *hoch* |

Tabelle 20.3: Beschreibung des Anwendungsfalls: BOM berechnen.

| **Keyword** | **Wert** |
|---|---|
| ID: | *Ch20-03* |
| Titel: | *BOM berechnen* |
| Beschreibung: | *Der Manager verwendet die APEX-Anwendung, um den Preis der durch die Stückliste beschriebenen Struktur für einen bestimmten Artikel zu berechnen, unabhängig davon, ob es sich um ein Produkt, ein Halbprodukt oder eine Anfrage handelt. Die sich daraus ergebenden Preise werden für alle in der Struktur aggregierten Positionen aktualisiert, unabhängig von ihrer Komplexität und Tiefe, einschließlich des obersten Aggregats.* |
| Primärer Akteur: | *Manager* |
| Voraussetzungen: | *Manager hat ein Konto für die APEX-Instanz.* |
| Nachbedingungen: | *Nach der Berechnung kann eine weitere Aktion durchgeführt werden, einschließlich einer weiteren Stücklistenberechnung.* |
| *Haupt* | *Szenarien* |
| Erfolgs-Szenario: | 1. *Öffnen Sie den Webbrowser und melden Sie sich bei der Anwendung an.*<br>2. *Wählen Sie den Menüpunkt oder die Seitennavigation „Tree view and calculation“.*<br>3. *Wählen Sie das zu berechnende Aggregat aus.*<br>4. *Klicken Sie auf die Schaltfläche „Calculate“. Es erscheint eine Meldung, die den Status der Berechnung und den berechneten Preis anzeigt.*<br>5. *Erweitern oder reduzieren Sie die Baumstruktur des ausgewählten Artikels, um die Daten der Unterteile anzuzeigen.* |
| Erweiterungen: | *keine* |
| Häufigkeit der Nutzung: | *durchschnittlich 30 Mal pro Monat* |
| Status: | *Fertiggestellt* |
| Eigentümer: | *Manager* |
| Priorität: | *hoch* |

mindestens ein druckbares Zeichen enthalten. Der Bereich der Maßeinheiten ist auf Millimeter, Zentimeter, Meter und Stück begrenzt. Aufgrund der geringen Anzahl unterschiedlicher Werte von Maßeinheiten wird keine zusätzliche Einheit eingeführt, um die Tatsache darzustellen, dass eine Maßeinheit viele Vorkommen in **ch20 psm** hat, während sich eine **ch20 psm**-Instanz auf nur eine Instanz einer Maßeinheit bezieht.

Zur Darstellung einer beliebigen hierarchischen Struktur (Stückliste beliebiger Breite und Tiefe) hat die Entität **ch20 bom** nur drei Attribute: Identifikation des Teilstücks, Identifikation des Aggregats und Menge des im Aggregat enthaltenen Teilstücks. Die Werte für alle Attribute müssen vorhanden sein. Das Paar (**Identifikation des Teilstücks, Identifikation des Aggregats**) ist eindeutig, wir werden jedoch einen Primärschlüssel hinzufügen, der als Folge von Ganzzahlwerten generiert wird, um die Vorteile der APEX-Umgebung mit wenig Code zu nutzen. Eine Instanz von **ch20 bom** kann eine doppelte Beziehung mit **ch20 psm** haben: Teilstück bezieht sich auf eine Instanz von **ch20 psm** und Aggregat bezieht sich auf eine Instanz von **ch20 psm**. In umgekehrter Richtung gilt: eine Instanz von **ch20 psm** kann auf viele Teilstücke und viele Aggregate in der Entität **ch20 bom** verweisen. Daher haben wir zwei 1-to-many-Beziehungen zwischen **ch20 psm** und **ch20 bom**. Auf eine weitere Sache des logischen Datenmodells muss noch hingewiesen werden: die Eindeutigkeit des Paares **(Identifikation des Teilstücks, Identifikation des Aggregats)**. Das logische Modell in Oracle Data Modeler hat keine „Syntax“, um diese Einzigartigkeit als spezielles Objekt in der Datenbank namens Index auszudrücken. Im relationalen Datenmodell ist es jedoch möglich, die Einzigartigkeit eines bestimmten Wertepaares als Index einzuführen. Im relationalen Datenmodell besteht der eindeutige Index aus zwei Feldern in der Tabelle. Dieser eindeutige Index garantiert, dass keine doppelten Wertepaare in der Datenbank gespeichert werden - die Datenbank-Engine schützt Entwickler und Endbenutzer vor Störungen der Datenintegrität.

### 20.4.2 Implementierung von Geschäftsregeln in die Datenbank

Das logische und relationale Datenmodell ist einfach. In der Mathematik ist die Stückliste ein spezieller Typ eines Graphen, bei dem jeder Knoten mit nur einem übergeordneten Knoten verbunden ist, mit Ausnahme des Wurzelknotens, der keinen übergeordneten Knoten hat. Es ist sehr einfach, eine Hierarchie aufzubrechen, indem man zusätzliche Verbindungen zwischen Knoten einführt oder eine Verbindung löscht. Im Zusammenhang mit Datenbanken wirken sich Änderungen in den Verbindungen auf die Datenintegrität aus. Die Speicherung einer zyklischen Struktur in **ch20 bom** würde dazu führen, dass die Berechnung der Stückliste nie endet. Im Kontext der Geschäftsregeln ist es unmöglich und verboten, dass ein Auto (Produkt) aus Rädern (Halbprodukt) und gleichzeitig das Rad aus Autos besteht. Auch ein Auto kann nicht aus sich selbst bestehen. Um die Verletzung von Geschäftsregeln zu verhindern, werden wir einen Trigger-Mechanismus (ein weiteres Datenbankobjekt) einführen, der normalerweise durch ein paar Zeilen Code in der Sprache PL/SQL dargestellt wird. Der Trigger wird ausgeführt, bevor in **ch20 bom** geschrieben oder aktualisiert wird. Ein solcher Code würde ein neues Einfügen in **ch20 bom** verhindern, wenn das Paar in umgekehrter Reihenfolge bereits existiert oder sowohl das neue Teilstück als auch das Aggregat denselben Wert haben. Die Implementierung dieser Art von Regeln kann nicht in einem logischen oder relationalen Datenmodell dargestellt werden.

Es ist auch nicht möglich, darzustellen, wie die Berechnung der Stückliste in Datenmodellen implementiert wird. In diesem Kapitel wird ein Einblick in den Code gegeben, der die Hierarchie auf den Kopf stellt (von unten nach oben) und mit der Summierung der Gesamtkosten von unten beginnt - d.h. zuerst werden die Materialkosten in den Halbprodukten und dann die Kosten der Halbprodukte im Produkt berechnet. Denken Sie daran, dass ein Datenmodell mit zwei Entitäten oder zwei Tabellen in der Lage ist, jede Komplexität (Breite und Tiefe der Hierarchie) der Stücklistenstruktur darzustellen.

Beginnen Sie also zunächst mit Datenmodellen.

### 20.4.3 Logisches Datenmodell

Das logische Datenmodell ist in Abbildung 20.2 dargestellt.

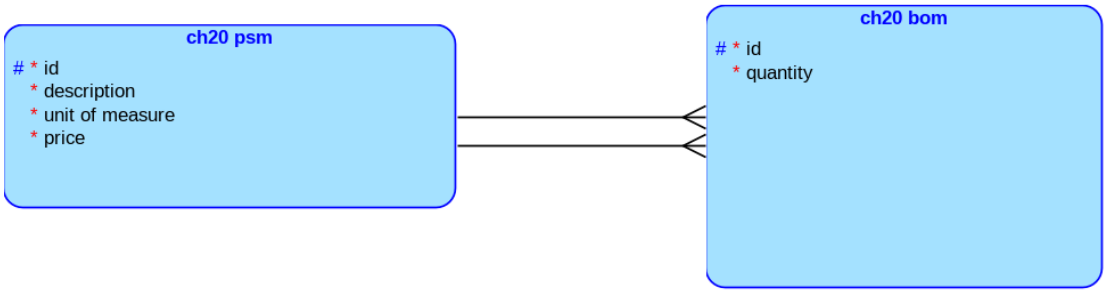



Abbildung 20.2: Logisches Datenmodell.

### 20.4.4 Relationales Datenmodell

Die automatische Umwandlung vom logischen Datenmodell in das relationale Datenmodell in Oracle SQL Data Modeler wird durch die Funktion *Engineering to relational* bereitgestellt. Dann führen wir einen eindeutigen Index *ch20_bom_ui* ein, der aus zwei Feldern *aggreg_id* und *subprt_id* besteht (siehe Abbildung 20.3). Das Ergebnis, ein relationales Datenmodell, das als SQL-Skript exportiert werden kann, ist in Abbildung 20.4 dargestellt.

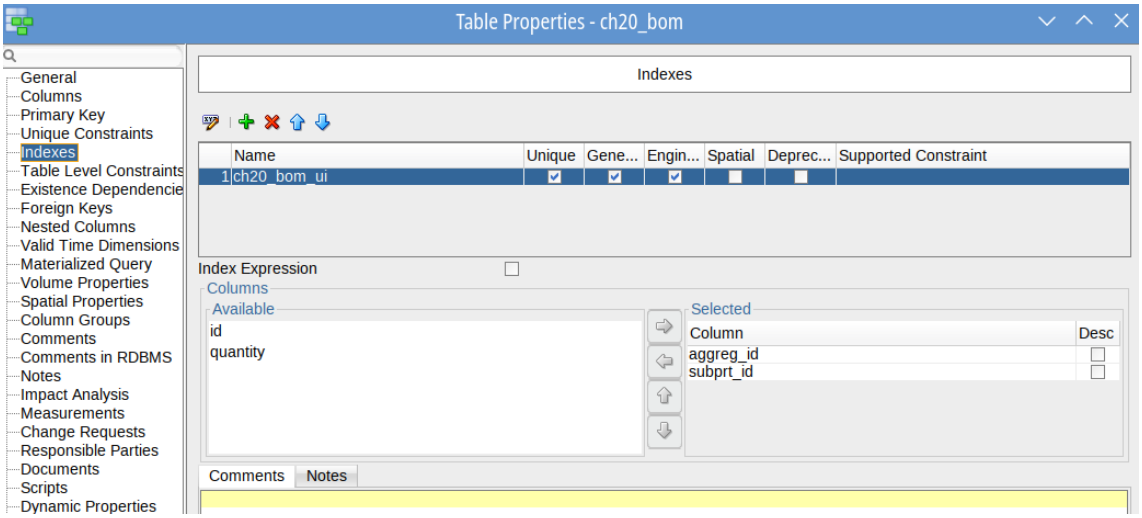



Abbildung 20.3: Logisches Datenmodell.

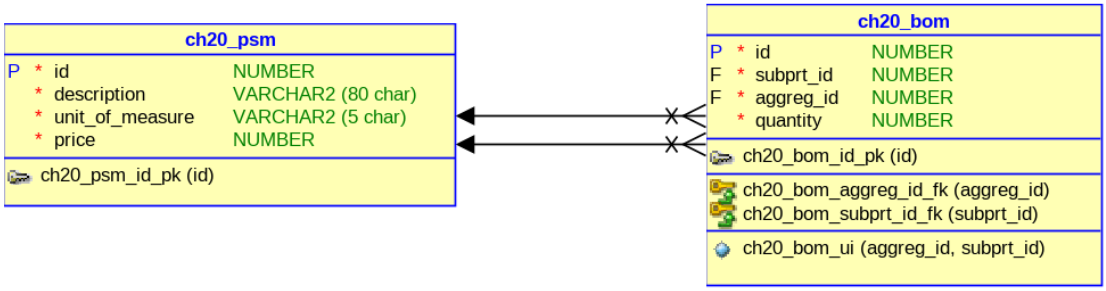



Abbildung 20.4: Logisches Datenmodell.

Der Oracle SQL Data Modeler generiert auch SQL-Skripte für die Erstellung von Tabellen, Sequenzen und Triggern. Wählen Sie alle Tabellen im relationalen Modell aus und verwenden Sie die Funktion *File > Export > DDL File*, um ein Skript zu erhalten. Speichern Sie das Skript und überprüfen Sie die Reihenfolge der Tabellendefinitionen. Die Tabelle *ch20_psm* muss zuerst definiert werden und dann *ch20_bom*. Der folgende Code definiert die Tabelle *ch20_psm* und legt Einschränkungen fest:

```
CREATE TABLE ch20_psm (
id NUMBER GENERATED BY DEFAULT ON NULL
AS IDENTITY ( START WITH 1 NOCACHE ) NOT NULL,
description     VARCHAR2(80 CHAR) NOT NULL,
unit_of_measure VARCHAR2(5 CHAR) NOT NULL,
price           NUMBER NOT NULL
) LOGGING;

ALTER TABLE ch20_psm ADD CONSTRAINT ch20_psm_id_pk PRIMARY KEY ( id );

ALTER TABLE ch20_psm ADD CONSTRAINT ch20_psm_unit_of_measure_ck
CHECK ( unit_of_measure IN ( 'CM', 'M', 'MM', 'PIECE' ) );
```

Der folgende Code definiert die Tabelle textitch20_bom, setzt Einschränkungen und definiert einen eindeutigen Index:

```
CREATE TABLE ch20_bom (
id        NUMBER GENERATED BY DEFAULT ON NULL
AS IDENTITY ( START WITH 1 NOCACHE ) NOT NULL,
subprt_id NUMBER NOT NULL,
aggreg_id NUMBER NOT NULL,
quantity  NUMBER NOT NULL
) LOGGING;

ALTER TABLE ch20_bom ADD CONSTRAINT ch20_bom_id_pk PRIMARY KEY ( id );

ALTER TABLE ch20_bom ADD CONSTRAINT
ch20_bom_subprt_id_fk FOREIGN KEY ( subprt_id )
REFERENCES ch20_psm ( id ) ON DELETE CASCADE NOT DEFERRABLE;

ALTER TABLE ch20_bom ADD CONSTRAINT
ch20_bom_aggreg_id_fk FOREIGN KEY ( aggreg_id )
REFERENCES ch20_psm ( id ) ON DELETE CASCADE NOT DEFERRABLE;

CREATE UNIQUE INDEX ch20_bom_ui ON ch20_bom (aggreg_id, subprt_id );
```

Speichern Sie das Skript mit den Anweisungen, die Tabellen mit Primär- und Fremdschlüsseln, Prüfbeschränkungen und eindeutigem Index erstellen. Jetzt ist es an der Zeit, Tabellen in der Oracle-Datenbank zu erzeugen. Wir können das generierte Skript importieren und es in APEX ausführen.

### 20.4.5 Objekte in APEX

Navigieren Sie mit Ihrem Browser zu Ihrem APEX-Arbeitsbereich. In diesem Abschnitt werden wir:

- Tabellen und Indizes erstellen

- Funktionen und Prozeduren erstellen
- Trigger erstellen

Es gibt zwei Möglichkeiten, Tabellen und Indizes zu erstellen. Wenn Sie den Oracle Data Modeler zur Erstellung des Skripts verwendet haben, importieren Sie es zuerst (SQL Workshop > SQL Scripts > Upload) und führen es dann aus. Eine andere Möglichkeit ist, das obige Skript mit Quick SQL zu generieren, wenn Sie nur APEX zum Generieren des Skripts verwenden möchten. Navigieren Sie zu SQL Workshop > Utilities > Quick SQL. Geben Sie den folgenden Text in den Quick SQL Textbereich ein (linker Teil des Fensters):

```
ch20_psm
description vc80 /nn,
unit_of_measure vc4  /check 'mm','cm','m', 'piece' /nn,
price num /nn

ch20_bom /unique
subprt_id num  /fk ch20_psm /nn,
aggreg_id num  /fk ch20_psm /nn,
quantity num /nn
```

Klicken Sie auf generate SQL. Die Tabellen haben die gleichen Eigenschaften wie die in Oracle Data Modeler generierten Tabellen (siehe Figure 20.5).

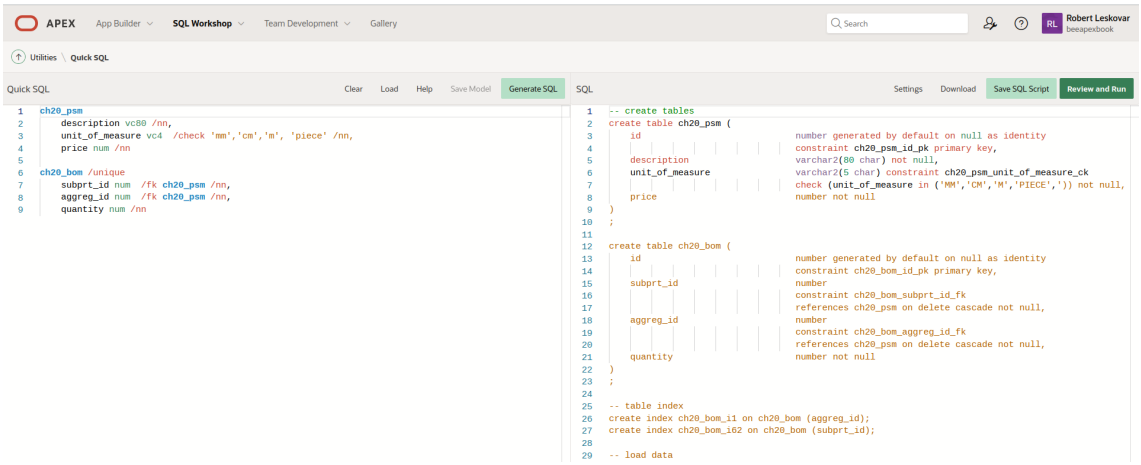



Abbildung 20.5: Logisches Datenmodell.

Schauen Sie sich die Zeilen 26 und 27 an. Diese beiden Zeilen sind zwar korrekt, werden aber später bearbeitet. Klicken Sie auf Save SQL Script und Review und Run. Ersetzen Sie die Zeilen 26 und 27 (zwei Indizes erstellen) und ersetzen Sie sie durch eine Anweisung:

```
create unique index ch20_bom_ui on ch20_bom (aggreg_id, subprt_id);
```

Führen Sie nun das Skript aus. Es werden zwei Tabellen und ein eindeutiger Index erzeugt. Beachten Sie, dass die Ausführung dieses Skripts nicht erfolgreich sein wird, wenn Sie bereits ein mit Oracle Data Modeler generiertes Skript ausführen. Der Grund dafür ist, dass bereits Datenbankobjekte existieren.

Um eine Funktion zu erstellen, geben Sie den folgenden Text in SQL Workshop > SQL Commands ein:

```
create or replace FUNCTION  CH20_GET_PRICE (
xpsm IN   NUMBER
```

```
) RETURN NUMBER
IS
y_price number;
BEGIN
SELECT price INTO y_price FROM ch20_psm  WHERE ID = xpsm;
return y_price;
EXCEPTION
WHEN OTHERS THEN
return(-505);
END CH20_GET_PRICE;
```

Diese Funktion gibt den Preis (Produkt, Halbprodukt, Material, Ware) für die angegebene ID zurück. Wenn etwas Unerwartetes mit der Datenbank passiert (Verbindungsabbruch, Serverausfall, Hardwarefehler, keine Daten für die angegebene ID), wird der negative Wert -505 zurückgegeben. Wählen Sie den gesamten Text aus und führen Sie ihn aus. Sie werden eine Rückmeldung erhalten, dass die Funktion erstellt wurde.

Um eine Prozedur zur Berechnung der Stücklistenkosten zu erstellen, geben Sie den folgenden Text in SQL Workshop > SQL Commands ein:

```
create or replace PROCEDURE  CH20_CALCULATE_BOM
(xpsm IN NUMBER,
ystatus OUT INTEGER,
yprice OUT NUMBER)
-- creation date: 20.1.2023
-- author: Robert Leskovar
-- input parameter: xpsm, id of product or semi-product
-- output parameters 1: status of calculation: 0=not calculated, 1=succesfully calculated
-- output parameters 2: price calculated
-- description: calculation of the total costs of given product and all underlaying levels
-- exceptions: system and user defined
-- date of testing: 20.1.2023
-- author of test: swqlab
-- status: APPROVED
is
v_rezult   NUMBER(38, 2);
v_psm        NUMBER;
v_price    NUMBER(38, 2);
e_negative_price EXCEPTION;
CURSOR c_ds IS
WITH STRUK(hier, subprt_id, aggreg_id, quantity) AS
(SELECT 1 AS hier, subprt_id, aggreg_id, quantity
FROM CH20_BOM JOIN CH20_PSM ON subprt_id = CH20_PSM.ID WHERE
aggreg_id = xpsm
UNION ALL
SELECT hier+1, C1.subprt_id, C1.aggreg_id, C1.quantity
FROM CH20_BOM C1 JOIN CH20_PSM P ON C1.subprt_id = P.ID
JOIN STRUK C2 ON C1.aggreg_id = C2.subprt_id)
SELECT hier, subprt_id, aggreg_id, quantity
FROM STRUK ORDER BY hier DESC, aggreg_id, subprt_id;
BEGIN
ystatus := 0;
```

```
v_rezult := 0.0;
v_psm := 0;
SAVEPOINT old_state;
FOR r_ds IN c_ds LOOP
v_price:= CH20_GET_PRICE(r_ds.subprt_id);
IF ( v_price <= 0 ) THEN
RAISE e_negative_price;
END IF;
IF ( r_ds.aggreg_id = v_psm ) THEN
v_rezult := v_rezult + v_price * r_ds.quantity;
UPDATE ch20_psm SET price = v_rezult WHERE ID = r_ds.aggreg_id;
ELSE
v_psm := r_ds.aggreg_id;
v_rezult := v_price * r_ds.quantity;
UPDATE ch20_psm SET price = v_rezult WHERE ID = r_ds.aggreg_id;
END IF;
END LOOP;
COMMIT;
ystatus := 1;
yprice := v_rezult;
EXCEPTION
WHEN e_negative_price THEN
ROLLBACK TO old_state;
ystatus := 0;
yprice := -505;
WHEN OTHERS THEN
ROLLBACK TO old_state;
ystatus := 0;
yprice := -100;
end CH20_CALCULATE_BOM;
```

Diese Prozedur gibt den Status der Berechnung und den Preis (Produkt, Halbprodukt, Material, Ware) für die angegebene ID zurück. Wenn etwas Unerwartetes mit der Datenbank passiert (Verbindungsabbruch, Serverausfall, Hardwarefehler, keine Daten für die angegebene ID), wird ein negativer Wert zurückgegeben. Wenn der Leser mit PL/SQL nicht vertraut ist, genügt es, die Kommentare in der Prozedur zu lesen. Ansonsten wird ein ziemlich komplexer Cursor mit einem allgemeinen Tabellenausdruck (WITH-Anweisung) erstellt. Dieser Cursor gibt die Vereinigung von zwei Mengen zurück. Er liefert die vollständige Struktur des Aggregats, das von unten nach oben geordnet ist. Die Preise der Zwischenaggregate werden in den entsprechenden Zeilen gespeichert und kumuliert, bis alle Zeilen im Cursor verarbeitet sind.

Das letzte zu erstellende Objekt ist der Trigger. Der Trigger stellt sicher, dass kein Aggregat und kein Unterteil in der Stückliste denselben Wert haben kann (Auto besteht aus Auto) und dass eine Traversalbeziehung erstellt wird (wenn Auto aus Rädern besteht, kann Rad nicht aus Auto bestehen). Um den Trigger zu erstellen, der die genannten Regeln implementiert, geben Sie den folgenden Text in SQL Workshop > SQL-Befehle ein:

```
create or replace trigger CH20_TRG_BOM_RULES
before insert or update on ch20_bom
for each row

declare
```

```
xsubprt numeric :=:new.subprt_id;
xaggreg numeric :=:new.aggreg_id;
xcount numeric;
begin
select count(subprt_id) into xcount from ch20_bom
where subprt_id = xaggreg and aggreg_id = xsubprt;
if xsubprt = xaggreg or xcount > 0 then
:new.subprt_id :=null;
end if;
end;
```

Mit diesem Trigger wird das Einfügen von Daten verhindert, indem ein NULL-Wert in einem Tabellenfeld erzwungen wird, das einen Wert haben muss. Der Vorrang der NOT NULL-Definition in der Tabellenspalte vor dem Trigger-Mechanismus bewirkt, dass keine Daten eingefügt werden, wenn diese Geschäftsregel verletzt wird. Wir haben jedoch dafür gesorgt, dass die Hierarchie zu einem Graphen mit Zyklus wird. Wenn wir die Erstellung dieses Triggers überspringen, sollten wir den Endbenutzer schulen, damit er nicht in die Zyklusfalle tappt. Vorbeugen ist viel besser als heilen, oder?

## 20.5 Anwendungsschnittstellen

In den folgenden Abbildungen werden die Anwendungsschnittstellen dargestellt. Zunächst wird die Skizze des Eingabefensters vorgestellt. Der Benutzer navigiert entweder über das Menü auf der linken Seite oder über die Seitennavigation zur Verwaltung der Basisdaten, der Strukturdaten und der Berechnung (siehe Abbildung 20.6).

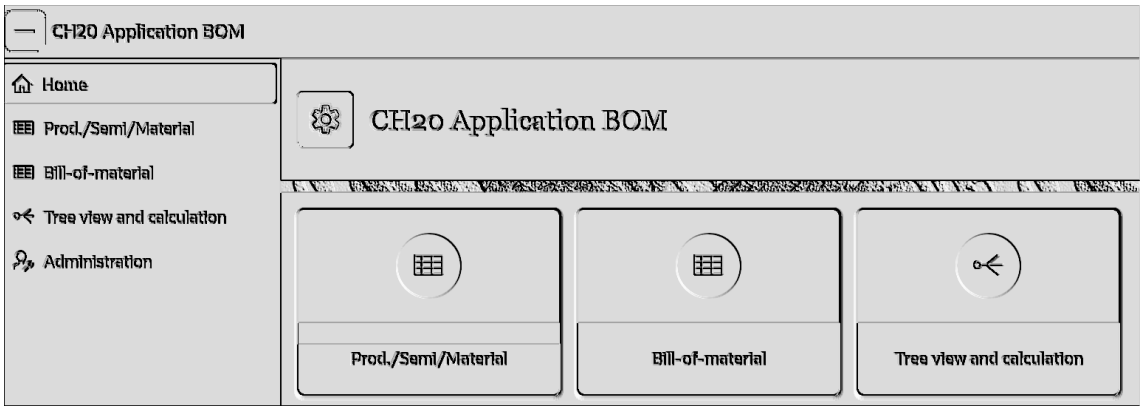



Abbildung 20.6: Startseite der Anwendung.

Die nächsten beiden Abbildungen (20.7 und 20.8) zeigen die Skizzen von Bericht und Formular für Produkte, Halbprodukte, Materialien und Anfragen. Einfügen, Aktualisieren und Löschen ist möglich.

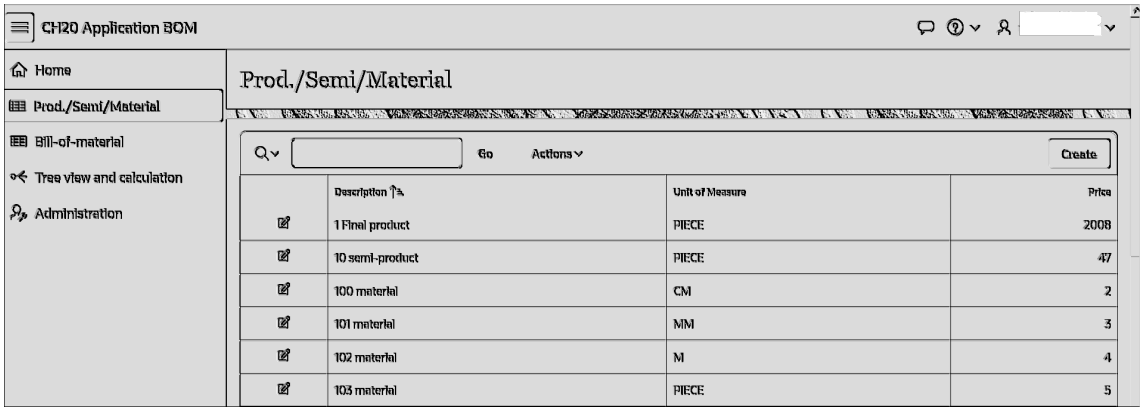



Abbildung 20.7: Verwaltung von Basisdaten - Bericht.

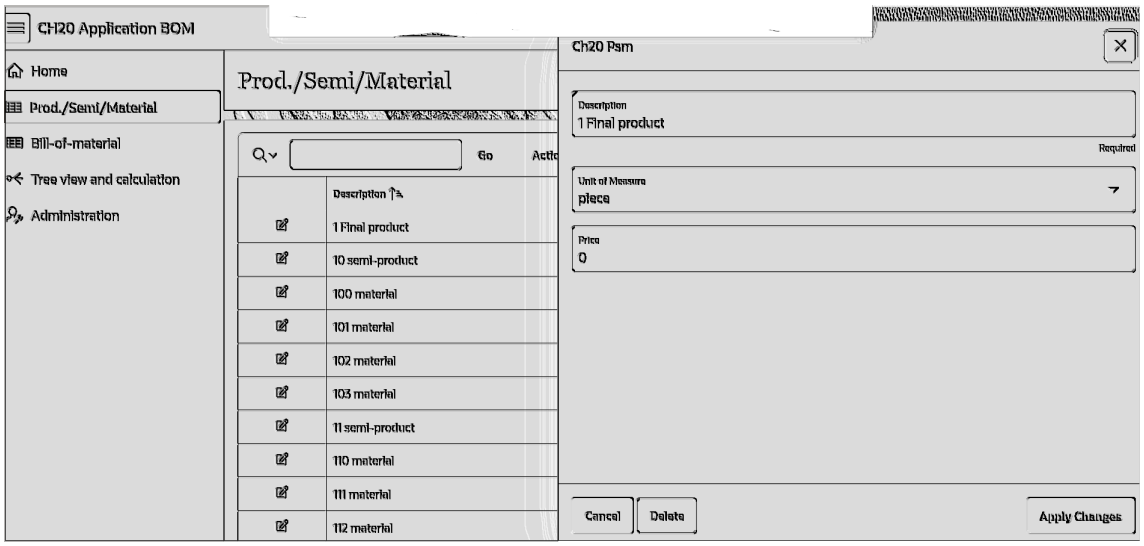



Abbildung 20.8: Verwaltung von Basisdaten - Formular.

Zwei weitere Abbildungen (20.9 und 20.10) zeigen die Skizzen des Berichts und des Formulars für die Struktur und die Stückliste. Einfügen, Aktualisieren und Löschen ist möglich.




Abbildung 20.9: Verwaltung von Strukturdaten - Bericht.

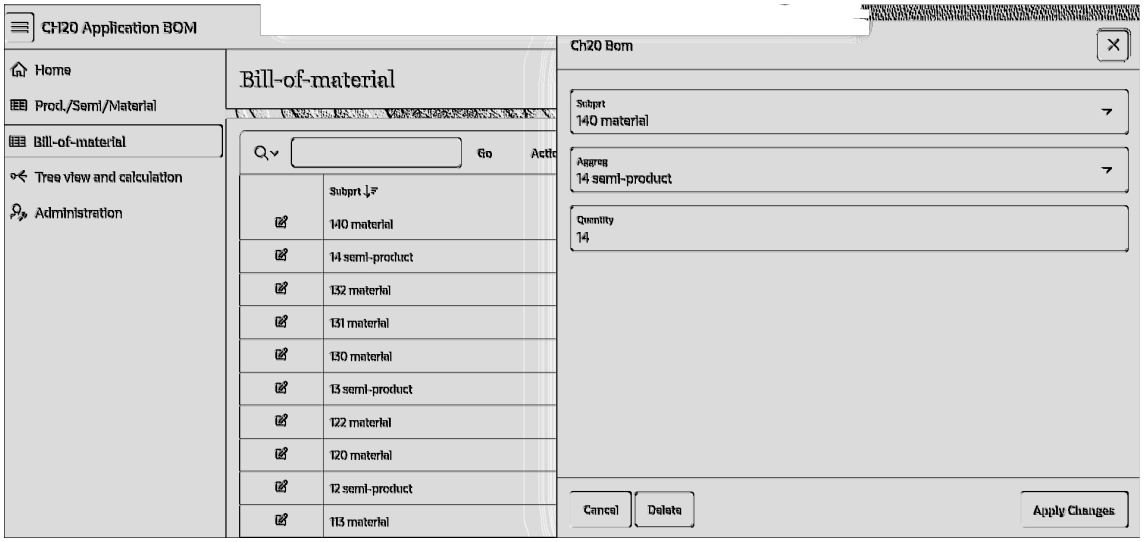



Abbildung 20.10: Verwaltung von Strukturdaten - Formular.

Die Skizze (20.11) zeigt die Schnittstelle zur Berechnung eines beliebigen Aggregats (Halbprodukt, Produkt, Anfrage). Der Benutzer wählt zunächst einen Artikel aus und drückt dann auf die Schaltfläche Calculate. Der Status der Berechnung und der berechnete Preis werden in einem Pop-up-Fenster angezeigt und eine Baumkomponente wird eingeblendet. Der Benutzer kann die Hierarchie auf- und zuklappen.

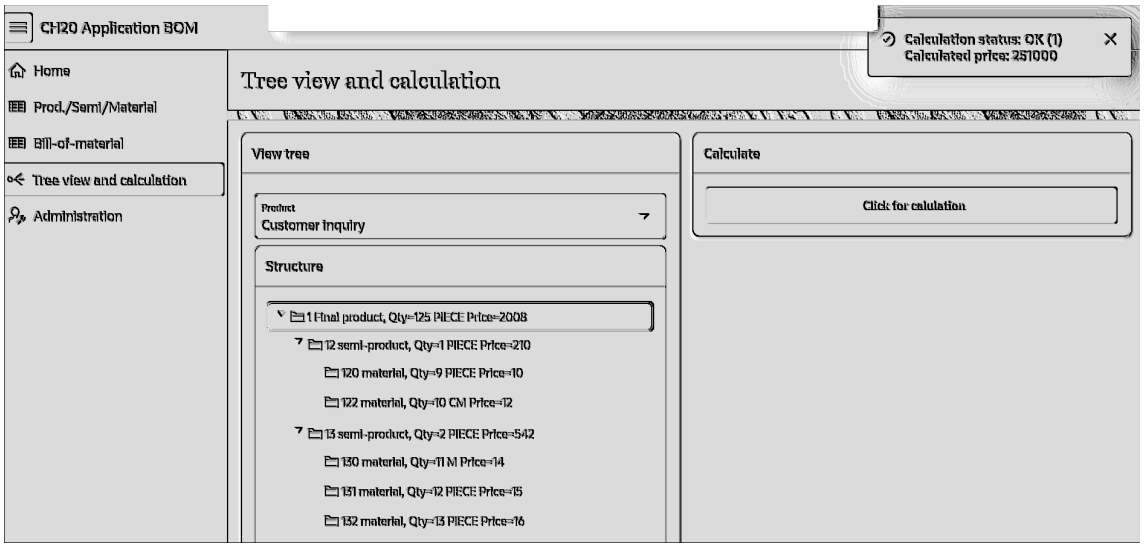



Abbildung 20.11: Seite zur Berechnung der Stückliste.

## 20.6 Ergänzendes Lernmaterial

Sie können das folgende zusätzliche Lernmaterial finden:

- exportierte Anwendung
- Video-Anleitungen

Alle zusätzlichen Lernmaterialien sind auf der öffentlichen BeeAPEX-Projektseite verfügbar. Melden Sie sich als Gastbenutzer an (kein Passwort erforderlich). Das Lehrbuch finden Sie im Abschnitt Books, die Skripte im Ordner Part 2 > Chapter20 im Abschnitt Scripts und die Videoanleitungen in der Sammlung der Videoanleitungen. Material für Kurzkurse finden Sie in der Sektion Short courses.

### 20.6.1 Exportierte Anwendungen

Die exportierte Anwendung wird verpackt. Bei der Installation werden sowohl Tabellen erstellt als auch Daten eingefügt. Bei der Deinstallation werden alle in dieser Anwendung verwendeten Datenbankobjekte entfernt.

### 20.6.2 Video-Anleitungen

Videoanleitungen zeigen jeden Schritt der Anwendungsentwicklung.

## 20.7 Fragen

1. Wie würden Sie das logische Datenmodell ändern, um eine neue Entität für Maßeinheiten zu implementieren, weil wir Zehnereinheiten einbeziehen wollen?
2. Wie lässt sich Quick SQL spezifizieren, um die oben genannte Änderung zu berücksichtigen?
3. Welche Folgen hat es, wenn der eindeutige Index ch20_bom_ui entfällt?

## 20.8 Antworten

1. Wir legen eine neue Entität *ch20_uom* an und setzen mindestens zwei Attribute (ID und Beschreibung der Maßeinheit). Dann setzen wir eine 1-to-many Beziehung zwischen *ch20_psm* und *ch20_uom*.
2. Das Paar attachment-news im logischen Modell würde in drei relationale Tabellen umgewandelt werden: attachment, news und attachment_news. Die letzte Tabelle würde mindestens zwei Felder enthalten (IDs von Attachment und News), die beide als Fremdschlüssel dienen.
3. In Quick SQL definieren wir zunächst die Tabelle *ch20_uom* als:
4. `ch20_uom`
   `description vc80`
   und ändern die Definition *ch20_psm* als:

   ```
   ch20_psm
   description vc80 /nn,
   unit_of_measure num /fk ch20_psm
   price num
   ```

5. Das Weglassen des eindeutigen Index ch20_bom_ui würde es ermöglichen, mehrere Paare (Unterteil, Aggregat) mit denselben Werten zu speichern. Die Tabelle würde mehr Speicherplatz verbrauchen. Die Berechnungsfunktionalität wäre davon nicht betroffen, die Verarbeitung würde jedoch länger dauern und mehr RAM verbrauchen.



# 21. Ernährungs- und Diätmanagement

A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis

## 21.1 Wirtschaftliche Sicht des Falles

Unser Beispiel ist Griechenlands beliebteste Lebensmittelmedienmarke mit einer Reichweite von über 500.000 Nutzern. Das Unternehmen ist stolzer Besitzer des marktführenden griechischen Food-Magazins und der digitalen Ausgabe, der führenden Food-Website des Landes, einer wachsenden Zahl von Live-Events, darunter die größte Food-Veranstaltung in Griechenland - die Greek Food Show - und einer äußerst erfolgreichen Kochbuchreihe.

Das Unternehmen hat beschlossen, ein Webtool zu entwickeln und zu bewerben, mit dem die Nutzer über ihre Website auf Tausende von Rezepten zugreifen können. Darüber hinaus können die Nutzer ihre Lieblingsrezepte durch Hinzufügen oder Entfernen von Zutaten personalisieren. Der Dienst liefert die Nährwertangaben der neuen Rezepte und ermöglicht es den Nutzern, ihr eigenes persönliches Kochbuch mit Lieblingsrezepten zu erstellen.

## 21.2 Problemstellung

Nutzer, die gerne kochen, stoßen oft an Grenzen, wenn es darum geht, was sie essen können, sei es aufgrund von gesundheitlichen Problemen wie Diabetes oder hohem Cholesterinspiegel oder aufgrund persönlicher Ernährungspräferenzen wie vegan oder glutenfrei. Ein Problem mit bestehenden webbasierten Rezeptressourcen ist, dass sie oft keine umfassenden Alternativen für diese Einschränkungen bieten. Dies kann dazu führen, dass die Nutzer von den begrenzten Informationen frustriert sind und die Rezeptsuche ganz aufgeben, was zu einer geringen Beteiligung führt.

## 21.3 Anwendungsfälle

### 21.3.1 Narrative Beschreibung des Anwendungsfalls

Wir werden eine Oracle APEX-Anwendung erstellen, mit der die Benutzer verschiedene gesunde Rezepte anzeigen und hinzufügen können. Die Anwendung wird eine detaillierte Beschreibung jedes Rezepts liefern, einschließlich einer Liste der Materialien und der Nährwertangaben für jede Zutat. Die Nutzer können auch ihre eigenen Rezepte und die entsprechenden Zutaten über ein Formular hinzufügen. Die Nährwerte jedes Rezepts, wie z.B. die Anzahl der Kalorien, die Gesamtproteinmenge, die Anzahl der Portionen und die Zubereitungszeit, werden zur besseren

Lesbarkeit in Diagrammen und nicht in einer einfachen Tabelle dargestellt.

### 21.3.2 Semi-strukturierte Beschreibung

Die semi-strukturierte Beschreibung ist in Tabelle 21.1 enthalten.

### 21.3.3 Anwendungsfalldiagramm

Die obige Geschichte wird im Anwendungsfalldiagramm dargestellt (siehe Abbildung 21.1).

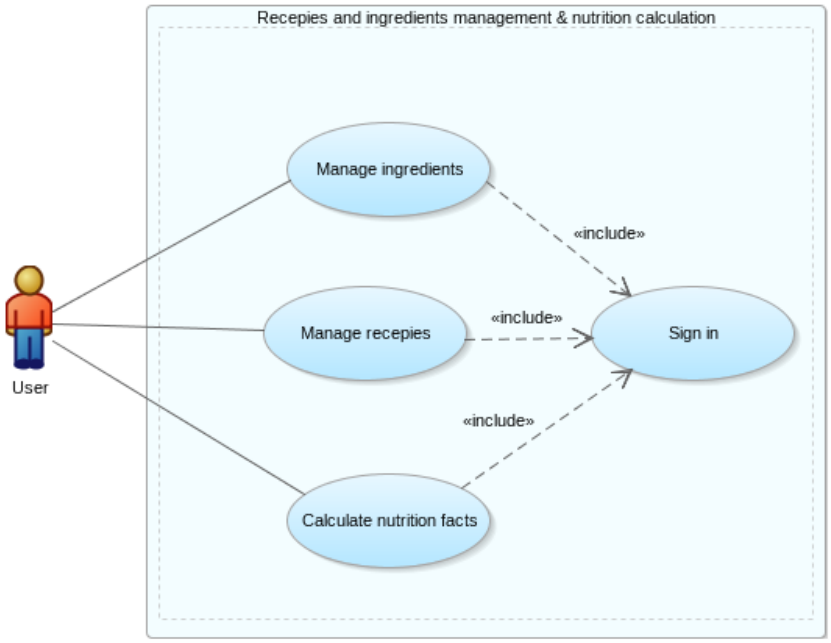



Abbildung 21.1: Anwendungsfalldiagramm.

## 21.4 Datenmodell

Das Datenmodell enthält drei Entitäten: Rezepte, Zutaten und Zutaten in Rezepten.

### 21.4.1 Logisches Datenmodell

Das logische Datenmodell ist in Abbildung 21.2 dargestellt.

Tabelle 21.1: Beschreibung des Anwendungsfalls: Zugang zu Rezepten, Zutaten und Nährwertberechnungen

| **Keyword&Wert** | |
|---|---|
| ID: &<br>*Ch21-01* | |
| Titel: &<br>*Die Plattform bietet Zugang zu einer Vielzahl von vorgefertigten Rezepten und Zutaten sowie die Möglichkeit, Zutaten hinzuzufügen oder zu entfernen, um die Rezepte an die individuellen Vorlieben anzupassen.Nutzer können ein neues personalisiertes Rezept erstellen, indem sie ihre bevorzugten Zutaten auswählen, und der Dienst berechnet die Nährwertangaben für das Rezept.* | |
| Beschreibung: | *Mit der Apex-Anwendung können die Benutzer Rezepte durchsuchen und auswählen sowie Rezeptdaten konfigurieren, z.B. Zutaten erstellen, aktualisieren und löschen. Benutzer können ein neues Rezept erstellen, indem sie Zutaten hinzufügen oder entfernen und die aktualisierten Nährwertangaben berechnen. Die Ergebnisse werden in Diagrammen angezeigt, die in Echtzeit auf der Grundlage der am Rezept vorgenommenen Änderungen aktualisiert werden.* |
| Primärer Akteur:&<br>*Benutzer* | |
| Voraussetzungen: | Benutzer hat ein Konto für die APEX-Instanz. |
| Nachbedingungen: & Nach dem Aktualisieren und Auswählen eines Rezepts kann der Benutzer die Nährwertangaben für ein Rezept berechnen und den Vorgang für andere Rezepte wiederholen, indem er Zutaten hinzufügt oder entfernt, um neue personalisierte Rezepte zu erstellen. | |

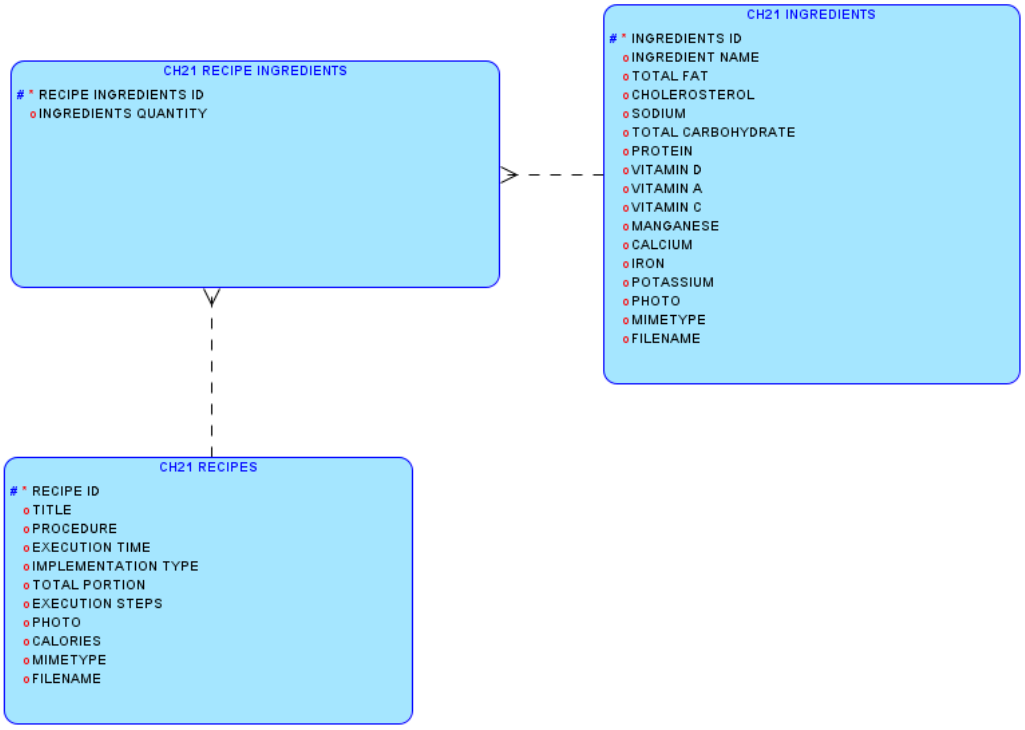



Abbildung 21.2: Logisches Datenmodell.

### 21.4.2 Relationales Datenmodell

Das relationale Datenmodell ist in Abbildung 21.3 dargestellt.

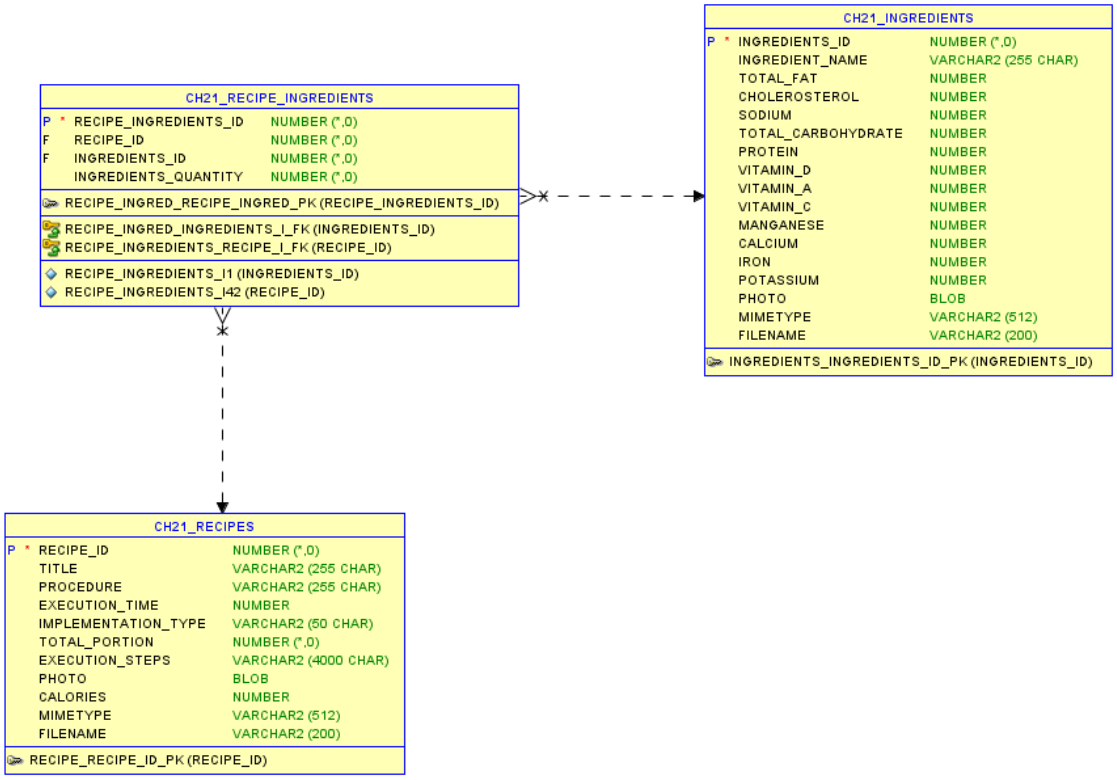



Abbildung 21.3: Relationales Datenmodell.

### 21.4.3 QuickSQL und Script SQL Skript

Die Tabellen, die wir für unsere Anwendung erstellen müssen, sind CH21_RECIPES, CH21_RECIPE_INGREDIENTS und CH21_INGREDIENTS. Die Felder für jede Tabelle sollten wie folgt strukturiert sein:

```
CH21_RECIPES
    RECIPE_ID int/pk,
    TITLE VC255,
    PROCEDURE VC255,
    EXECUTION_TIME NUM,
    IMPLEMENTATION_TYPE VC50,
    TOTAL_PORTION int,
    EXECUTION_STEPS  VC,
    PHOTO BLOB

CH21_INGREDIENTS
    INGREDIENTS_ID int/pk,
    INGREDIENT_NAME VC255,
    TOTAL_FAT NUM,
    CHOLEROSTEROL NUM,
    SODIUM NUM,
    TOTAL_CARBOHYDRATE NUM,
    PROTEIN NUM,
    VITAMIN_D NUM,
    VITAMIN_A NUM,
    VITAMIN_C NUM,
    MANGANESE NUM,
```

```
    CALCIUM NUM,
    IRON NUM,
    POTASSIUM NUM,
    PHOTO BLOB,
    MIMETYPE VC512,
    FILENAME VC200


CH21_RECIPE_INGREDIENTS
    RECIPE_INGREDIENTS_ID INT/PK,
    RECIPE_ID INT/FK RECIPE,
    INGREDIENTS_ID INT/FK INGREDIENTS,
    INGREDIENTS_QUANTITY INT
```

Um Quick SQL Code auszuführen, müssen Sie zunächst auf den **SQL Workshop** zugreifen und **SQL Scripts** auswählen. Von dort aus können wir **Quick SQL** wählen, um Tabellen zu erstellen. Der Vorgang ist einfach; wir müssen nur den Tabellennamen und seine Felder sowie alle untergeordneten Tabellen angeben, die mit unserer Tabelle verbunden sind. Abbildung 21.4 zeigt, wie man mit dieser Funktion schnell Tabellen erstellen kann.

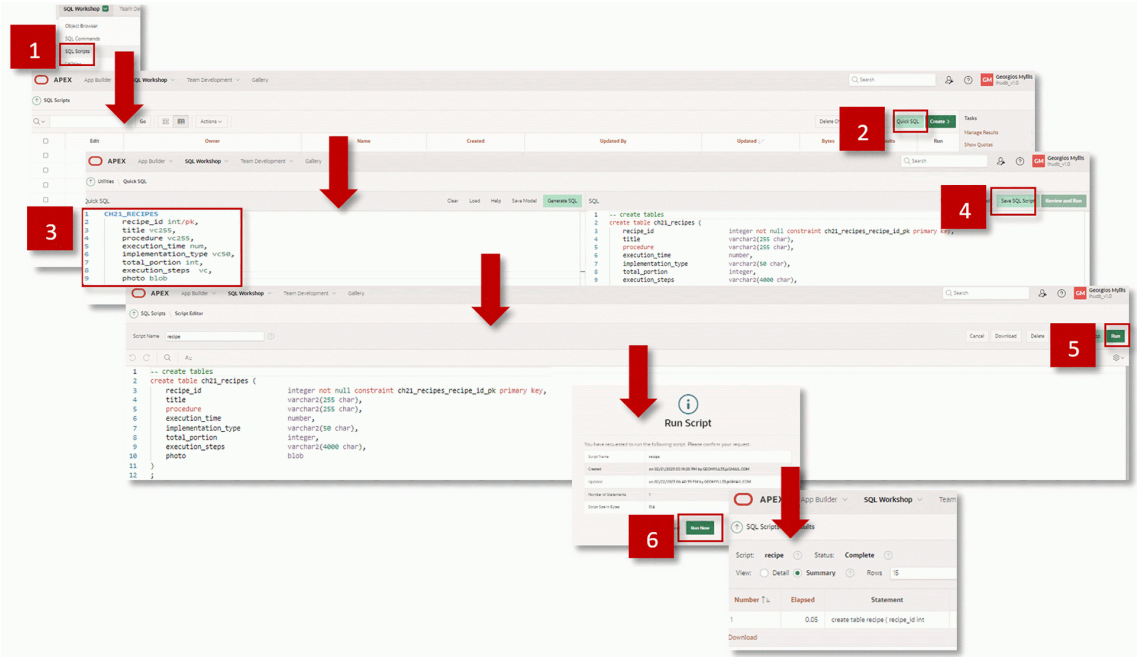



Abbildung 21.4: Quick-SQL-Tool zur Erstellung von SQL-Skripten.

Unser nächster Schritt für das ordnungsgemäße Funktionieren unserer Tabellen besteht darin, sicherzustellen, dass jede Hinzufügung zur Tabelle eindeutig ist und dass der Primärschlüssel jeder Tabelle automatisch inkrementiert wird. Um dies zu erreichen, müssen wir drei SEQUENCES erstellen.

Rezept-ID erhöhen

```
CREATE SEQUENCE seq_RECIPE
START WITH 1
INCREMENT BY 1
```

```
NOCACHE
NOCYCLE;

CREATE OR REPLACE TRIGGER RECIPE_trg
BEFORE INSERT
ON RECIPE
FOR EACH ROW
BEGIN
:new.RECIPE_ID:=seq_RECIPE.nextval;
END;
```

Zutaten-ID erhöhen

```
CREATE SEQUENCE seq_INGREDIENTS
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;

CREATE OR REPLACE TRIGGER INGREDIENTS_trg
BEFORE INSERT
ON INGREDIENTS
FOR EACH ROW
BEGIN
:new.INGREDIENTS_ID:=seq_INGREDIENTS.nextval;
END;
```

Rezept-Zutaten-ID erhöhen

```
CREATE SEQUENCE seq_RECIPE_INGREDIENTS
START WITH 31
INCREMENT BY 1
NOCACHE
NOCYCLE;

CREATE OR REPLACE TRIGGER RECIPE_INGREDIENTS_trg
BEFORE INSERT
ON RECIPE_INGREDIENTS
FOR EACH ROW
BEGIN
:new.RECIPE_INGREDIENTS_ID:=seq_RECIPE_INGREDIENTS.nextval;
END;
```

Um die SEQUENCE auszuführen, müssen wir zum **SQL Workshop** navigieren und auf **SQL Scripts** klicken. Dann können wir auf die Schaltfläche **Create** klicken und den Code für jede SEQUENCE einzeln kopieren und einfügen. Nachdem wir dem Skript einen Namen gegeben haben, können wir auf die Schaltfläche **Run** klicken, um das Skript auszuführen. Dadurch wird

sichergestellt, dass jeder Datensatz in den Tabellen eindeutig ist, und es ist nicht mehr erforderlich, den zuletzt hinzugefügten Datensatz manuell zu verfolgen. Alles wird automatisch erledigt. Ein Beispiel für die Ausführung der SEQUENCE in der APEX-Umgebung ist in Abbildung 21.5 dargestellt.

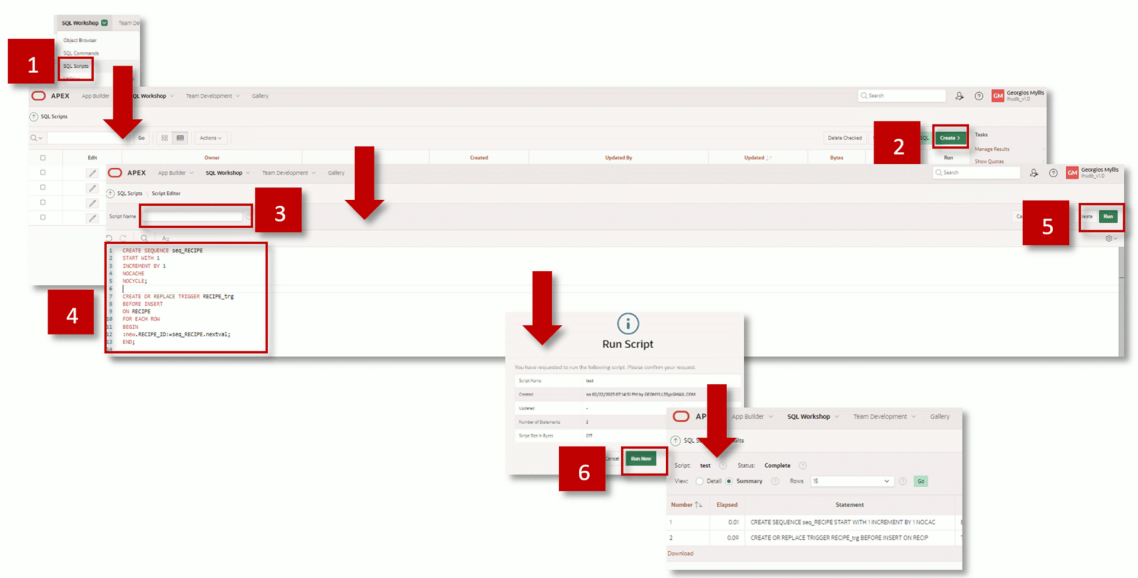



Abbildung 21.5: SQL-Skript-Prozedur zur Erstellung von SEQUENCE.

## 21.5 Anwendungsschnittstellen

Um die Anwendung wie in der Abbildung 21.6 zu erstellen, klicken Sie auf **App Builder** und dann auf **Create**. Daraufhin wird ein neues Fenster angezeigt, in dem wir den Namen unserer Anwendung angeben und die gewünschten Funktionen auswählen müssen. Die Tabellen, die wir für unsere Anwendung verwenden werden, sind: RECIPE, RECIPE INGREDIENTS, INGREDIENTS, wie bereits erklärt.

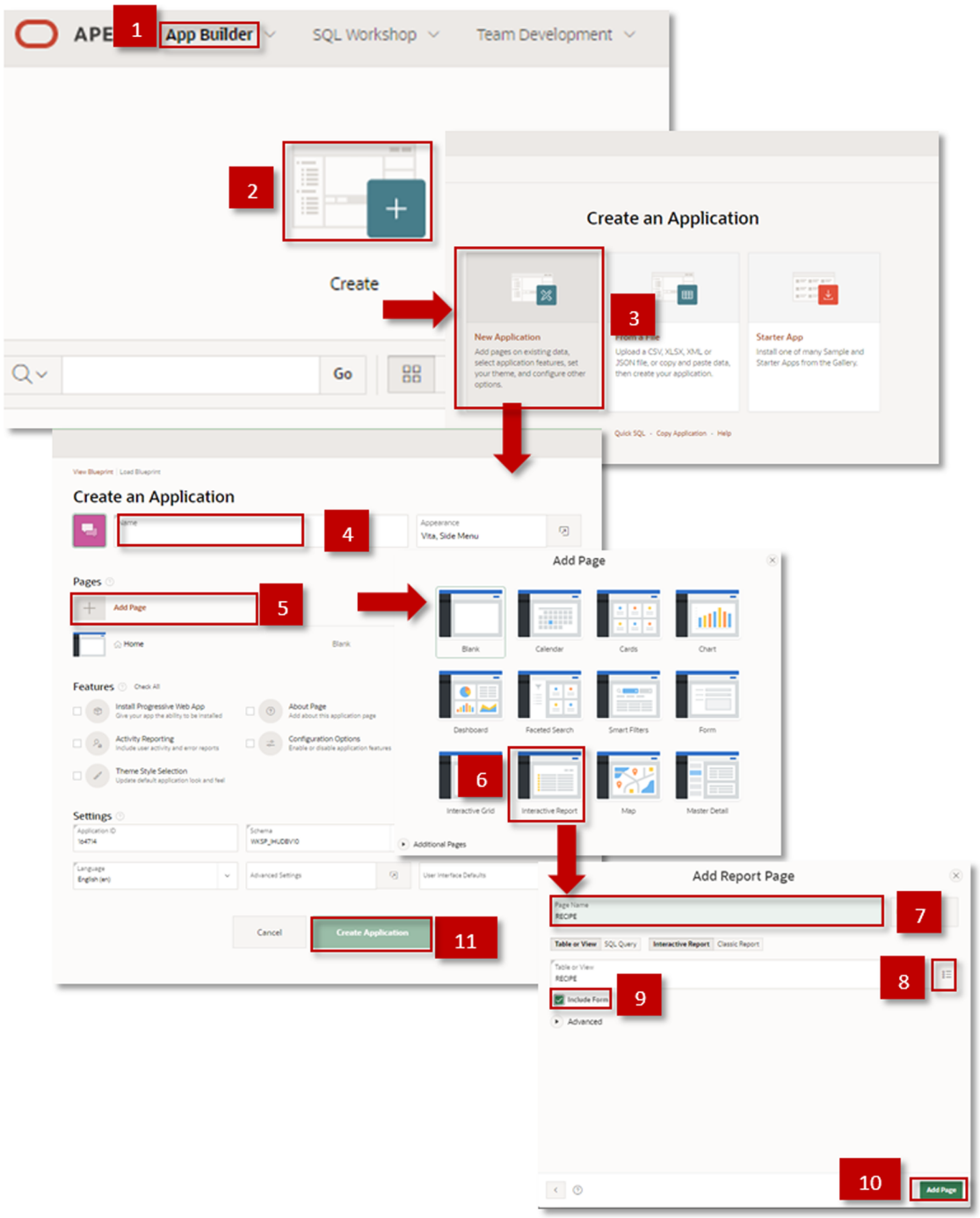



Abbildung 21.6: Neue Anwendung erstellen.

In Abbildung 21.7 wurde eine neue App Builder-Umgebung für Ihre neue Anwendung erstellt. Um auf Ihre Anwendung zuzugreifen, klicken Sie darauf und geben Sie Ihre Administrator-Anmeldedaten ein, die mit denen für den Arbeitsbereich übereinstimmen sollten. Sobald Sie sich angemeldet haben, können Sie über das Mega-Menü im oberen Bereich durch Ihre Anwendung navigieren.




Abbildung 21.7: Ernährungsanwendung.

Auf der Rezeptseite in Abbildung 21.8 können die Benutzer auf alle für die Zubereitung des Rezepts erforderlichen Informationen zugreifen, z.B. auf das Kochverfahren, die Ausführungszeit, die Gesamtportionen, die Ausführungsschritte und ein Foto des gewünschten Ergebnisses. Je nach ihren Berechtigungen können die Nutzer auch ein neues Rezept hinzufügen oder ein bestehendes nach ihren Wünschen aktualisieren.

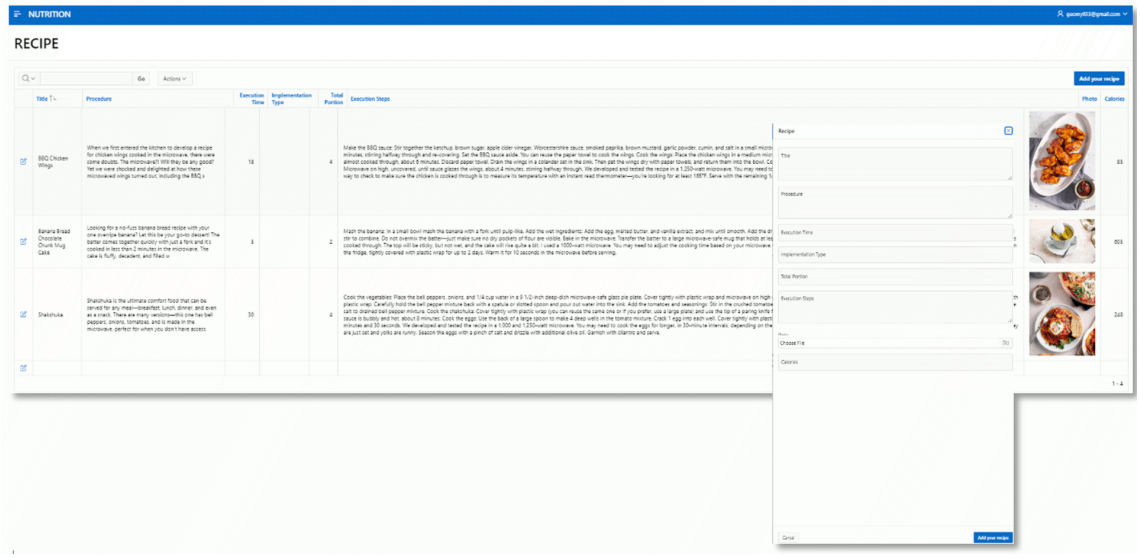



Abbildung 21.8: Ernährungsanwendung - Rezeptseite.

In Abbildung 21.9 werden auf der Seite mit den Zutaten verschiedene Attribute der Zutat angezeigt, z.B. Gesamtfett, Cholesterin, Natrium, Gesamtkohlenhydrate, Proteine usw. Diese Attribute werden verwendet, um die Nährwertangaben der Rezepte zu berechnen. Benutzer mit den entsprechenden Rechten können die Seite pflegen, indem sie bei Bedarf neue Zutaten hinzufügen oder bestehende aktualisieren.

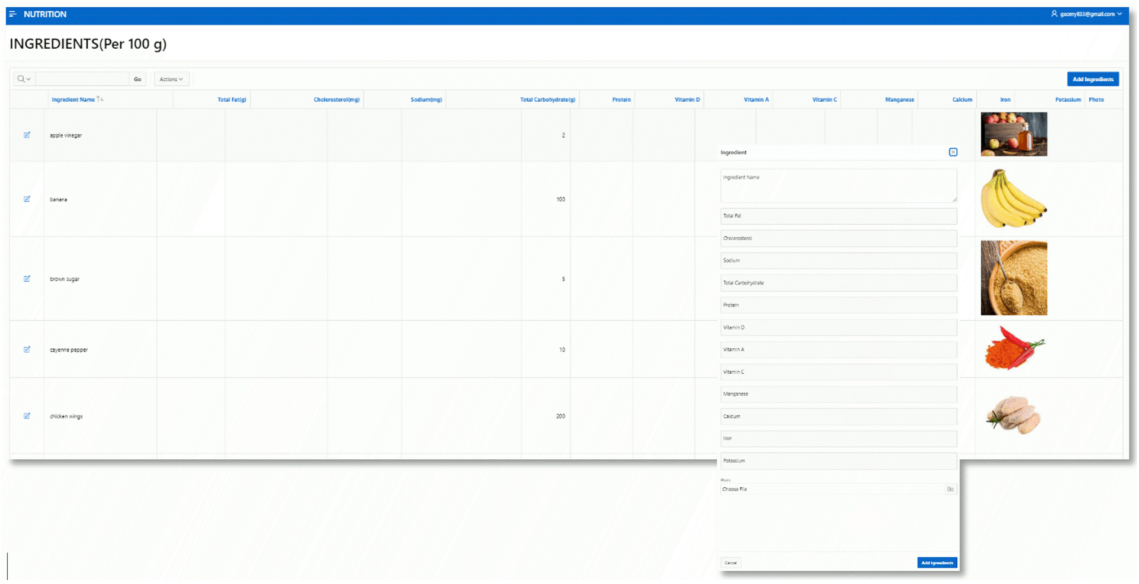



Abbildung 21.9: Ernährungsanwendung - Zutaten-Seite.

In Abbildung 21.10 bietet die Seite mit den Rezeptzutaten den Nutzern die Möglichkeit, diesen Webdienst zu nutzen, um Rezepte nach ihren Bedürfnissen zu erstellen oder bestehende zu ändern und anschließend die Nährwertangaben zu überprüfen. Benutzer mit den entsprechenden Berechtigungen auf dieser Seite können Zutaten zu den Rezepten hinzufügen oder aus ihnen entfernen.

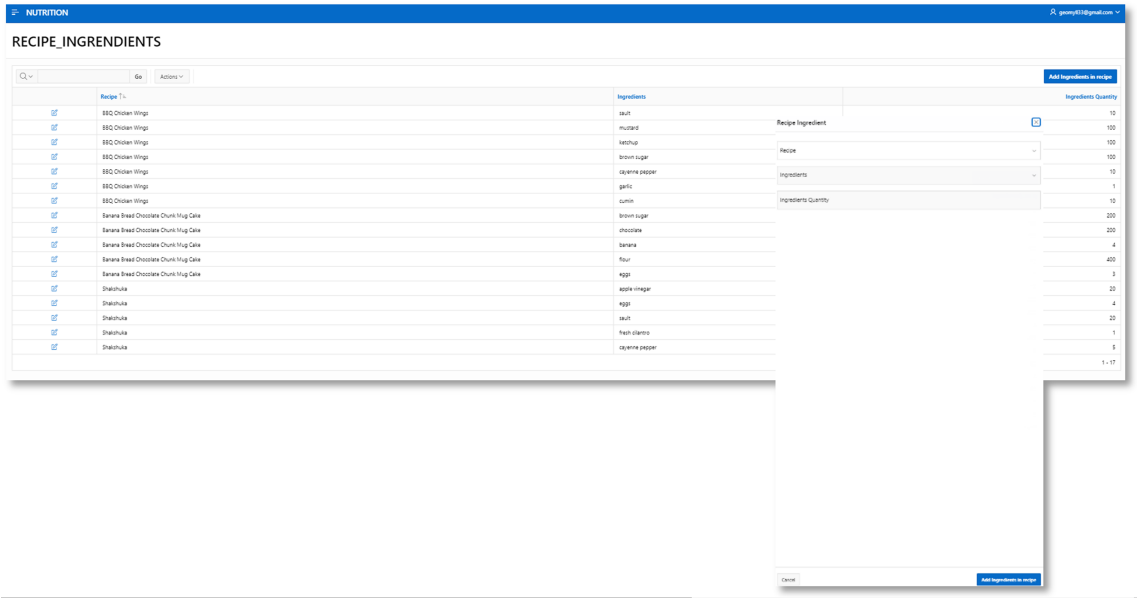



Abbildung 21.10: Ernährungsanwendung - Rezept-Zutaten-Seite.

Auf der Seite mit den Nährwertangaben in Abbildung 21.11 erhält der Benutzer alle notwendigen Informationen über das von ihm erstellte Rezept. Es werden Informationen wie der Gesamtproteingehalt des Rezepts, der Gesamtkaloriengehalt des Rezepts, die Anzahl der Portionen, für die das Rezept geeignet ist, und die Dauer der Zubereitung des Rezepts angegeben.

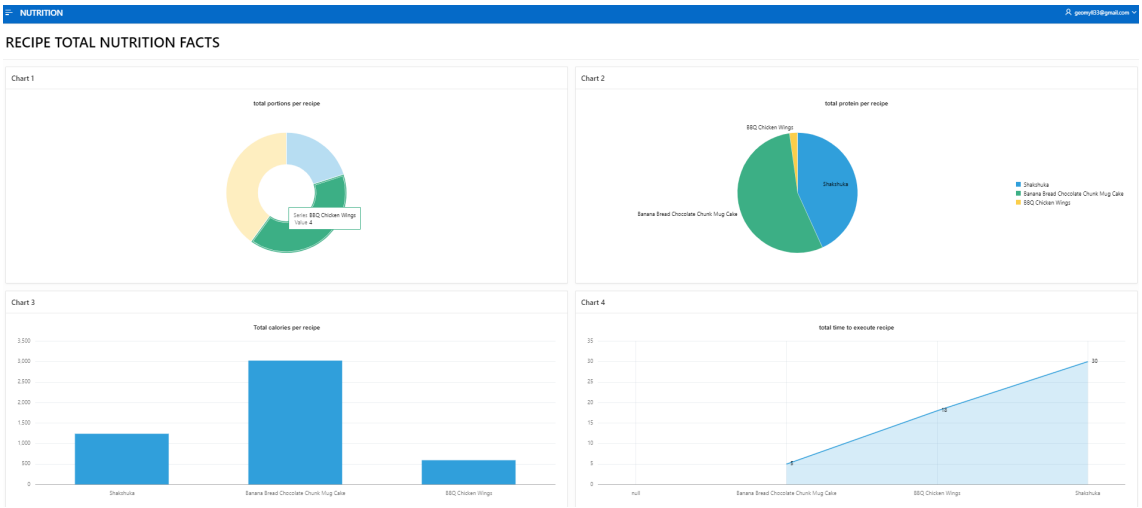



Abbildung 21.11: Ernährungsanwendung - Nährwerttabellen-Seite.

### 21.5.1 Wie man ein Dashboard erstellt

Um eine Dashboard-Seite für Ihre Anwendung zu erstellen, führen Sie die in Abbildung 21.12 gezeigten Schritte aus.

- Klicken Sie auf Ihre Anwendungsumgebung.
- Klicken Sie auf die Schaltfläche **Create Page**.
- Wählen Sie in der Liste der verfügbaren Seitentypen die Option **Dashboard**.
- Klicken Sie auf die Schaltfläche **Next**.
- Wählen Sie Ihr bevorzugtes Dashboard-Layout.
- Klicken Sie auf die Schaltfläche **Create Page**, um die Dashboard-Seite zu erstellen.

Diese Schritte führen Sie durch den Prozess der Erstellung einer Dashboard-Seite für Ihre Anwendung. Nach der Erstellung können Sie das Dashboard anpassen, indem Sie Widgets hinzufügen und das Layout so konfigurieren, dass die Daten angezeigt werden, die Sie zur Überwachung und Verwaltung Ihrer Anwendung benötigen.

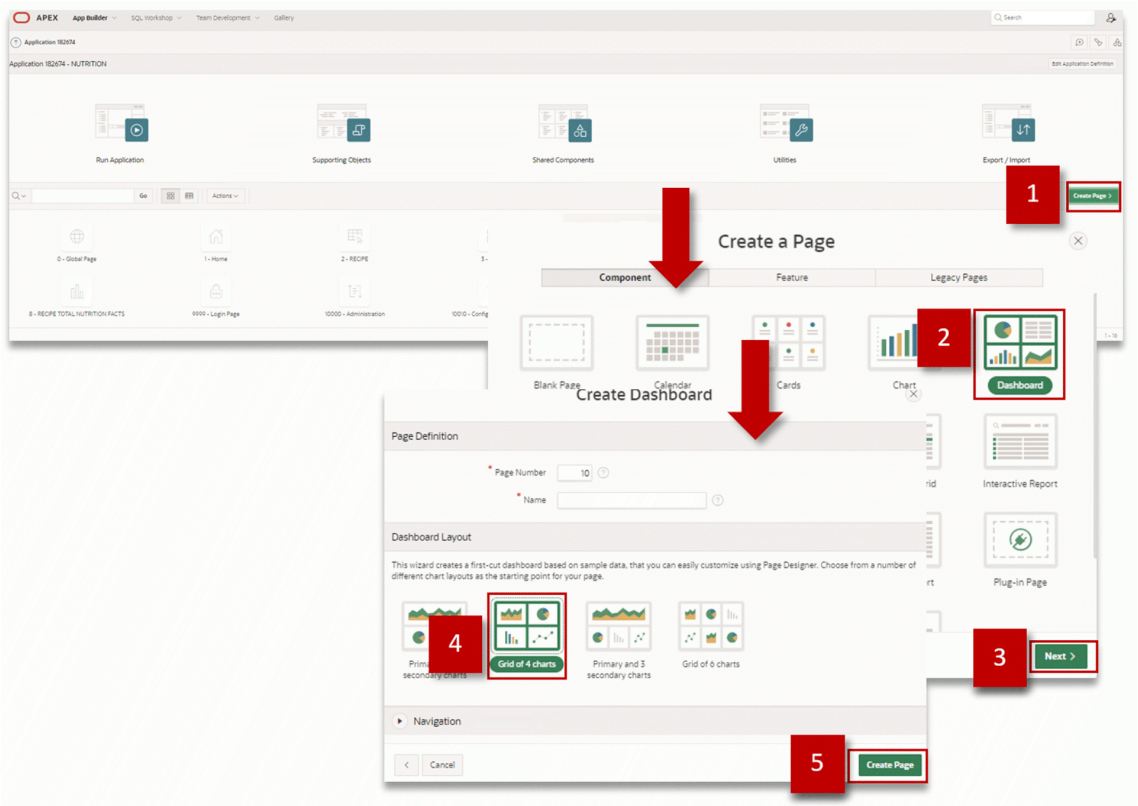



Abbildung 21.12: Erstellen Sie eine Dashboard-Seite für Ihre Anwendung.

Navigieren Sie nun zu Ihrem Dashboard-Bericht, den Sie in Ihrer Anwendungsumgebung finden, und rufen Sie den Seitendesigner auf. Wie in Abbildung 21.12 dargestellt, haben wir vier Diagramme ausgewählt und werden jedem davon mithilfe von SQL-Abfragen unterschiedliche Daten hinzufügen.

Wählen Sie zunächst Diagramm 1 und benennen Sie es in „Total Portion per Recipe“ um. Wählen Sie dann **Attribute** und wählen Sie den Diagrammtyp Donut. Wählen Sie als Nächstes Diagramm 2 und benennen Sie es in „Total Protein per Recipe“ um, damit die Rezepte mit dem meisten Protein angezeigt werden. Wählen Sie für diese Tabelle den Typ **Pie Chart**.

Gehen Sie zu Diagramm 3 über und benennen Sie es in „Total Calories per Recipe“ um. In diesem Diagramm wird die Gesamtzahl der Kalorien für jedes Rezept angezeigt, und wir verwenden den Diagrammtyp **Bar Chart**. Schließlich benennen Sie Diagramm 4 in „Total Time to Execute Recipe“ um. Wählen Sie **Attributes** und wählen Sie dann den Diagrammtyp **Line with Area** (siehe Abbildung 21.13).

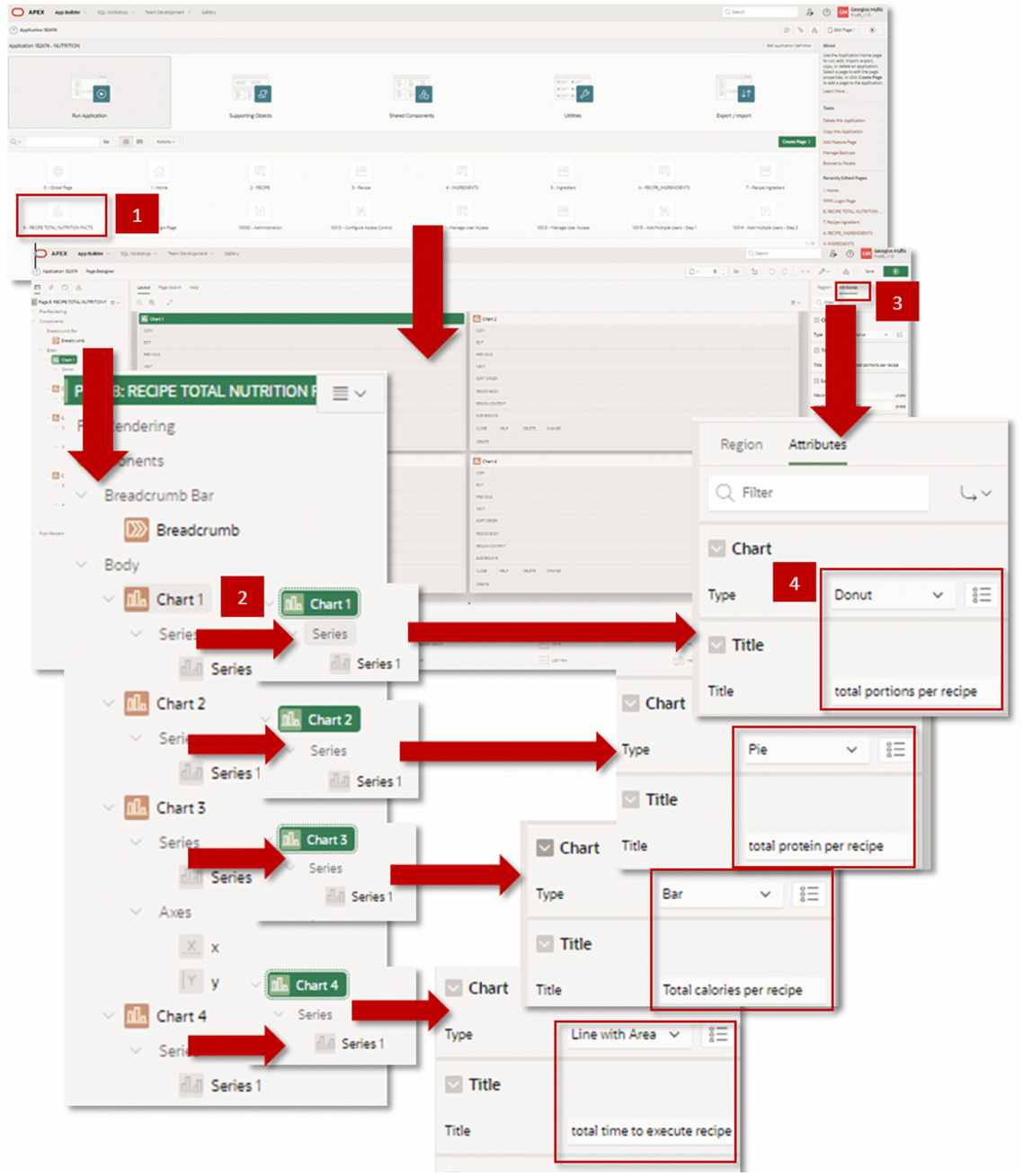



Abbildung 21.13: Konfiguration von Titel und Chat-Typ in Dashboard-Berichten.

Klicken Sie dann auf **Series 1** von Diagramm 1 und nehmen Sie unter Quelle die folgenden Einstellungen vor:

```
SELECT DISTINCT RECIPE.TITLE,
RECIPE.TOTAL PORTION
FROM RECIPE
```

Geben Sie in Reihe 1 von Diagramm 2 unter Quelle die folgenden Einstellungen ein

```
SELECT DISTINCT RECIPE.TITLE, SUM(protein) AS protein
FROM RECIPE
INNER JOIN RECIPE_INGREDIENTS ON
```

```
RECIPE.RECIPE_ID=RECIPE_INGREDIENTS.RECIPE_ID
INNER JOIN INGREDIENTS ON
INGREDIENTS.INGREDIENTS_ID=RECIPE_INGREDIENTS.INGREDIENTS_ID
GROUP BY RECIPE.TITLE
```

Geben Sie in Reihe 1 von Diagramm 3 unter Quelle die folgenden Einstellungen ein

```
select distinct recipe.title, sum(CALORIES) as CALORIES
from recipe
inner join recipe_ingredients on
recipe.recipe_id=recipe_ingredients.recipe_id
inner join ingredients on
ingredients.ingredients_id=recipe_ingredients.ingredients_id
group by recipe.title
```

und in Reihe 1 von Diagramm 4 unter Quelle die folgenden Einstellungen vornehmen

```
SELECT DISTINCT RECIPE.TITLE,RECIPE.EXECUTION_TIME
FROM RECIPE
```

Abbildung 21.14 enthält einige Hinweise zur Konfiguration.

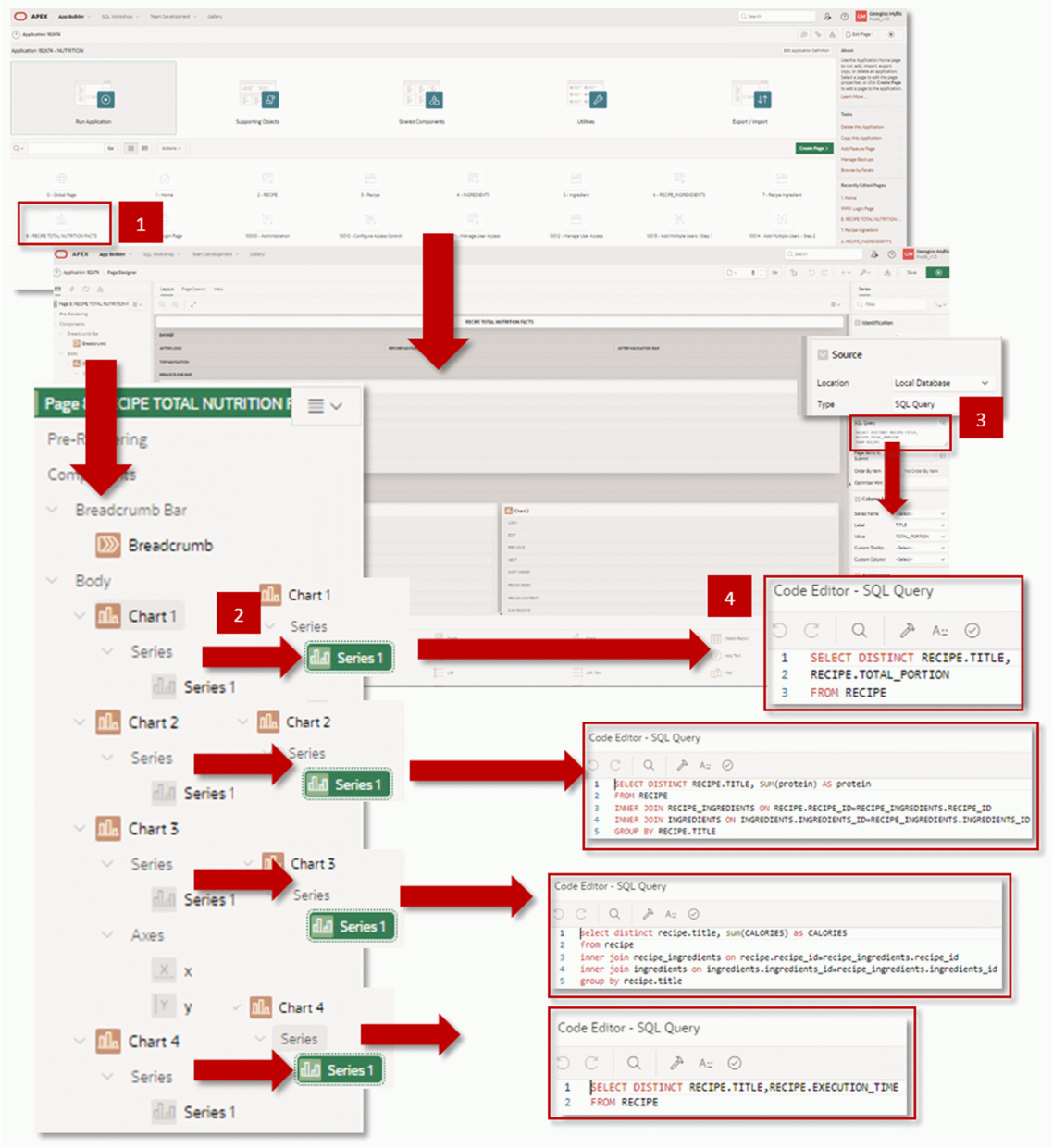



Abbildung 21.14: SOURCE mit SQL QUERY konfigurieren.

### 21.5.2 Entwickler und Anwendungsnutzer

In Oracle APEX müssen Entwickler und Endbenutzer über Konten verfügen, jedoch müssen wir zwischen den Privilegien für Arbeitsbereich, Anwendung und spezifische Seite unterscheiden. Workspace-Privilegien beziehen sich auf die Workspace-Administration, die Entwicklung, den Zugang zum App Builder, den Zugang zum SQL-Workshop und den Zugang zur Team-Entwicklung. Anwendungsbenutzer haben alle diese Rechte deaktiviert. Um ein neues Benutzerkonto zu erstellen, wählen Sie im Dropdown-Menü **Manage Users and Groups**. Wählen Sie dann **Create User**, geben Sie die Identifikation und das Kennwort des Benutzers ein, definieren Sie die Berechtigungen (Administrator, Entwickler, Zugriff auf Tools) und klicken Sie auf **Create User**, um den Vorgang abzuschließen (siehe Abbildung 21.15).

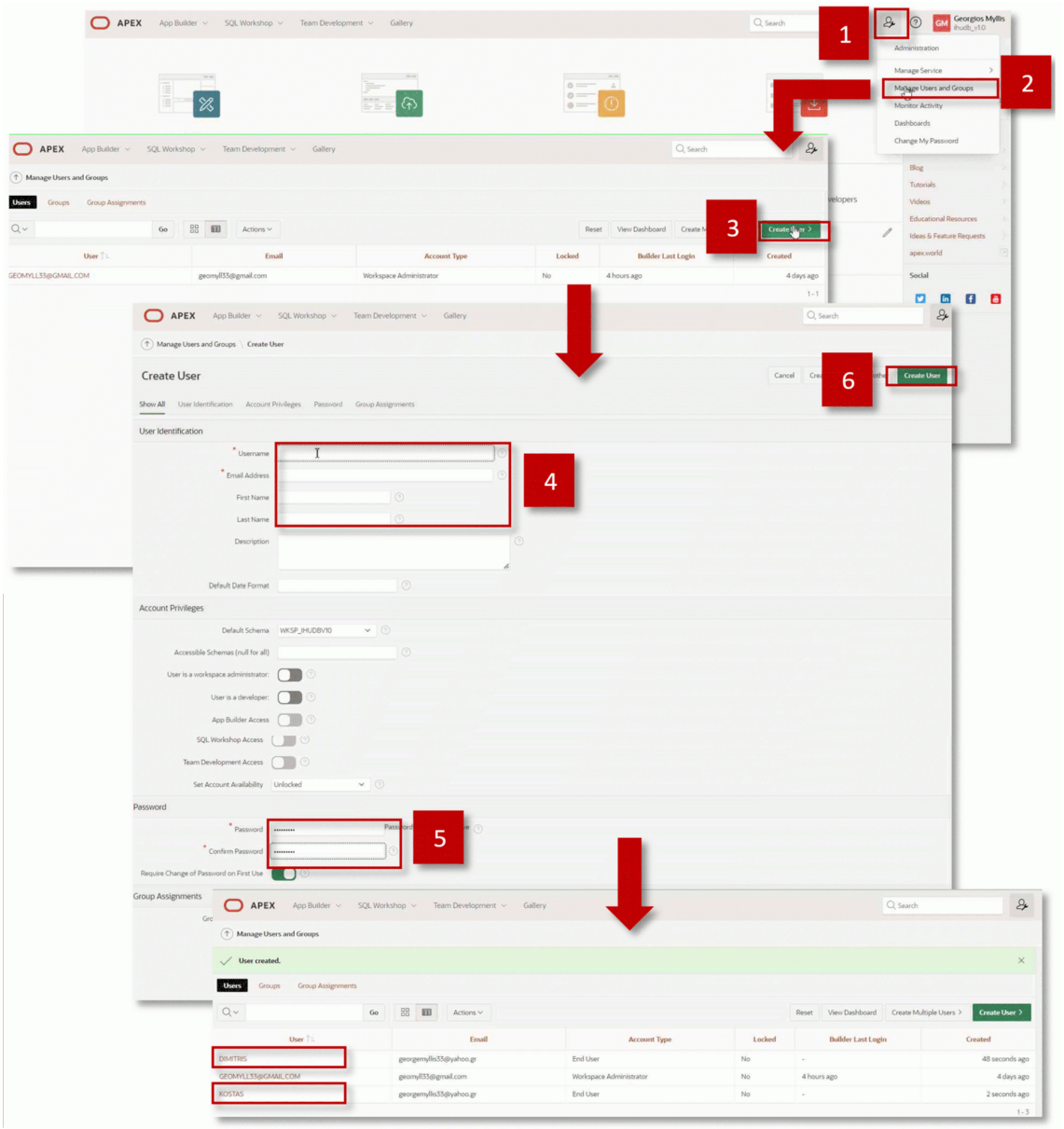



Abbildung 21.15: Erstellung eines neuen Benutzerkontos.

Anwendungsbenutzer und ihre Anwendungsprivilegien (Autorisierung) können auf der Seite Anwendungsadministration definiert (erstellt) werden. Um diese Seite zu erstellen, kreuzen Sie die Funktion Access Control beim Ausführen des Anwendungsgenerators an. Rufen Sie für diese spezielle Anwendung das Mega-Menü auf und klicken Sie auf **Administration**. Klicken Sie dort auf **Add User**, geben Sie die Identifikation und die Rolle des Benutzers ein, und klicken Sie schließlich auf **Add User**. Weitere Anleitungen finden Sie in der Abbildung 21.16.

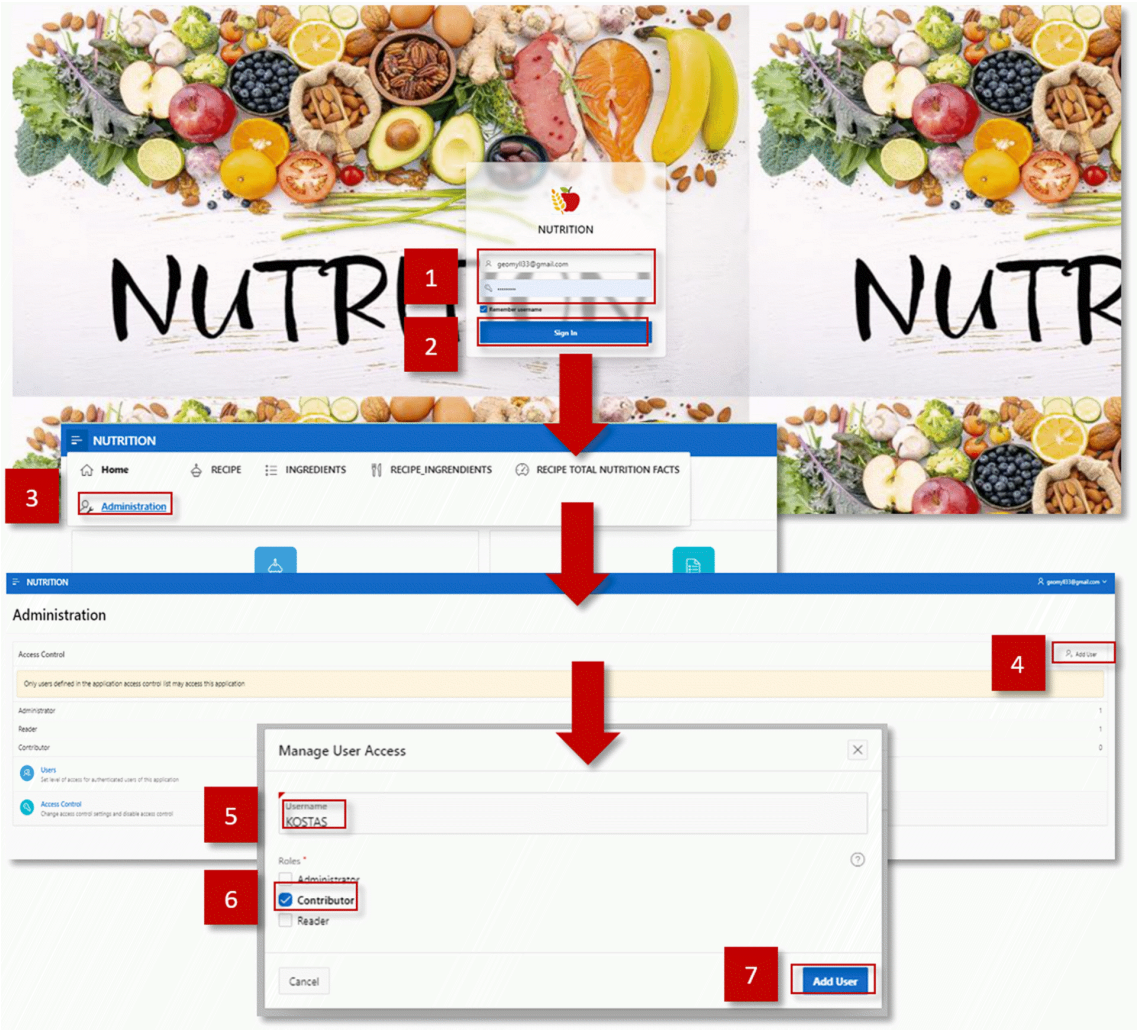



Abbildung 21.16: Benutzerkonto in der Anwendung erstellen.

Sie können einem bestimmten Benutzer oder einer bestimmten Rolle Zugriffsrechte gewähren. Im Seitendesigner (siehe Abbildung 21.17 für die Empfangsseite) finden Sie die Sicherheitsoptionen. Wählen Sie **Contribution Rights** als Berechtigungsschema. Klicken Sie auf **Save**. Dadurch wird sichergestellt, dass nur Benutzer mit der Beitragsrolle in der Lage sind, Rezepte zu erstellen.

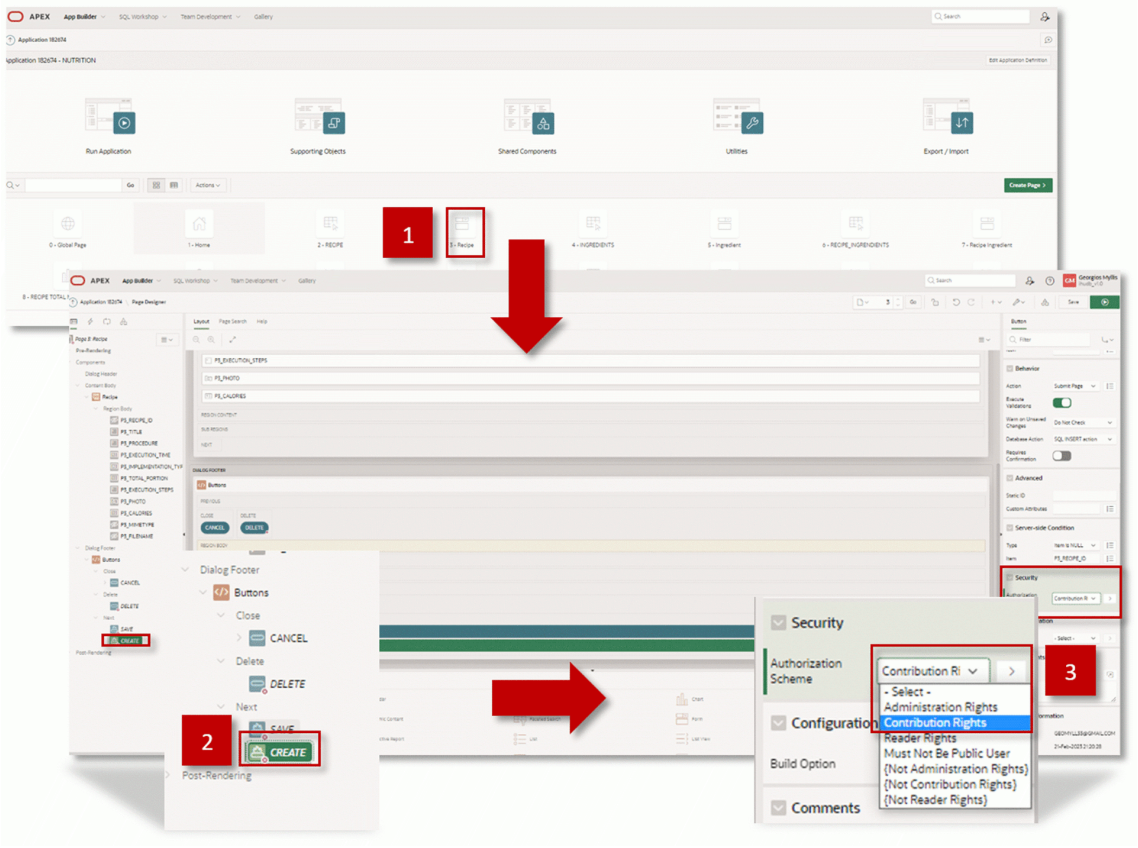



Abbildung 21.17: Seiten mit Benutzerrechten konfigurieren.

## 21.6 Ergänzendes Lernmaterial

Sie können das folgende zusätzliche Lernmaterial finden:

- exportierte Anwendung
- Video-Guides

Alle zusätzlichen Lernmaterialien sind auf der öffentlichen BeeAPEX-Projektseite verfügbar. Melden Sie sich als Gastbenutzer an (kein Passwort erforderlich). Das Lehrbuch finden Sie im Abschnitt Books, die Skripte im Ordner Part 2 > Chapter21 im Abschnitt Scripts und die Videoanleitungen in der Sammlung der Videoanleitungen. Material für Kurzkurse finden Sie in der Sektion Short courses.

### 21.6.1 Exportierte Anwendungen

Die exportierte Anwendung wird verpackt. Bei der Installation werden sowohl Tabellen erstellt als auch Daten eingefügt. Bei der Deinstallation werden alle in dieser Anwendung verwendeten Datenbankobjekte entfernt.

### 21.6.2 Video-Anleitungen

Videoanleitungen zeigen jeden Schritt der Anwendungsentwicklung.

## 21.7 Fragen

1. Wie kann man den Hintergrund der Anmeldeseite ändern?
2. Wo können wir sonst die Rollenverwaltung in der Oracle APEX-Umgebung finden, wenn sie in unserer Anwendung nicht aktiviert ist?

3. Wie können Sie den Namen der Schaltfläche eines Berichts konfigurieren?
4. Wie kann man den Seitennamen nach dem Erstellen der Seite ändern?

## 21.8 Antworten

1. Wählen Sie in Ihrer App-Umgebung **Shared Components** und dann **Static Application Files**, klicken Sie auf **Create File** und fügen Sie im Pop-up-Fenster das gewünschte Bild hinzu, kopieren Sie den Referenznamen des Bildes, gehen Sie zurück zu Ihrer App-Umgebung und wählen Sie **9999-Login Page**, gehen Sie zum Feld css und geben Sie inline ein

   ```
   body {
   background-image:
   url(\glqq #APP_FILES#\textit{Referenznamen des Bildes}\grqq);
   background-size:1700px 900 px:
   }
   ```

2. Wählen Sie in Ihrer Anwendungsumgebung **Shared Components** und dann **Application Access control**.
3. Um die Beschriftung der Schaltfläche **Create** in einen anderen Namen zu ändern, müssen Sie den Seitendesigner des Berichts aufrufen, die Schaltfläche **Create** suchen und die Beschriftung im Feld Identifikation ändern.
4. Um den Namen einer von Ihnen erstellten Seite zu ändern, melden Sie sich bei Ihrer App an und klicken Sie auf **Quick Edit**. Klicken Sie dann auf den Seitentitel und wählen Sie den Pfeil im Quellbereich des Breadcrumbs. Klicken Sie dann auf **Edit Component**, und aktualisieren Sie im Feld **Short Name** den Namen auf den gewünschten neuen Namen. Klicken Sie schließlich auf **Apply Changes**, und der neue Name wird im Seitentitel angezeigt.



# 22. Sprechstundenplanung

J. Mańko, M. Sońta, R. Leskovar

## 22.1 Wirtschaftliche Sicht des Falles

Eine universitäre Sprechstundenbuchungsanwendung kann aus betriebswirtschaftlicher Sicht als ein Werkzeug zur Rationalisierung und Optimierung des Planungsprozesses für das Universitätspersonal betrachtet werden. Sie kann dazu beitragen, die Effizienz der Terminplanung zu verbessern, die Arbeitsbelastung der Verwaltung zu verringern, die Zufriedenheit der Studierenden zu erhöhen und Verfügbarkeitsinformationen in Echtzeit bereitzustellen. Die Anwendung kann auch Daten und Erkenntnisse über Terminmuster und -auslastung generieren, die als Grundlage für die Entscheidungsfindung und Ressourcenzuweisung dienen können. Darüber hinaus kann die Anwendung eine bequeme und leicht zugängliche Buchungsplattform für Studierende bieten, die dazu beiträgt, eine proaktivere und engagiertere Studentengemeinschaft zu fördern. Das akademische Personal ist verpflichtet, jede Woche 60 Minuten seiner Zeit für die Studierenden zur Verfügung zu stellen, was intern als „Sprechstunde“ bezeichnet wird. Außerdem kommt es häufig vor, dass sich die Anfragen der Studierenden häufen, vor allem gegen Ende des Semesters, wenn die Anzahl und der Umfang der Anfragen zunimmt. In einem solchen Fall reichen 60 Minuten pro Woche nicht aus, da die Studierenden darauf bedacht sind, so schnell wie möglich ein dringend gewünschtes Ergebnis vom akademischen Personal zu erhalten. Es gibt zwar einige standardisierte Gesprächsthemen, wie z.B.: Diplomarbeit, Prüfungen, Abschlussarbeiten, aber einige der Fälle beziehen sich auf irrelevante Themen, die vielleicht besprochen werden sollten, aber nicht unbedingt während der Sprechstunde mit dem Dozenten (z.B.: administrative Anfragen oder Probleme, die leicht ohne das Eingreifen des akademischen Lehrers bewältigt werden können). In den letzten Jahren und insbesondere aufgrund der COVID-19-Pandemie haben die Universitäten einen radikalen technologischen Wandel vollzogen, der verschiedene Formen des Fernunterrichts über verfügbare Plattformen und Kommunikationsmittel ermöglicht, die den Online-Lehrprozess erleichtern. In der Tat wurden praktisch alle akademischen Aktivitäten online verlagert, darunter natürlich auch die Fernsprechstunden. Obwohl sich die akademische Gemeinschaft darüber einig ist, dass die Fernlehre die Lehre vor Ort niemals vollständig ersetzen wird, wird die Fernlehre und sogar die Fernarbeit in gewissem Umfang weltweit weitergeführt. Noch wichtiger ist, dass einige universitäre Aktivitäten und Prozesse bereits vollständig auf den Online-Modus umgestellt wurden oder in naher Zukunft umgestellt werden, um beispielsweise einen effizienten papierlosen Dokumentenumlauf zu gewährleisten. Diese Umstände verstärken noch den dringenden Bedarf an einem praktikablen digitalen Tool, das die Online-Buchung von Sprechstunden ermöglicht, die sowohl online als auch

offline stattfinden können. Es liegt auf der Hand, dass die gesamte akademische Gemeinschaft sehr von einer solchen Anwendung profitieren würde, die bald zum alltäglichen Werkzeug für jeden in der akademischen Gemeinschaft werden sollte, wie z.B. ein E-Mail-Postfach, und nicht eine ferne digitale Modeerscheinung.

## 22.2 Problemstellung

Derzeit ist es nicht möglich, die Zeit zu reservieren (sich anzumelden) und den Inhalt der Sitzung im Voraus zu planen. Vielmehr werden die Studenten, wenn sie in beliebiger Anzahl erscheinen, nach dem Prinzip „Wer zuerst kommt, mahlt zuerst“ bedient. Dies erhöht nicht nur die Frustration der Studenten, die in der Schlange warten müssen, ohne sicher sein zu können, dass ihr Termin überhaupt stattfindet, sondern verärgert auch das akademische Personal, das sich entscheiden muss, ob es einige Studenten umsonst warten lässt oder ob es seinen Plan für den Tag umstellt. Infolgedessen kommt es zu Situationen, in denen das akademische Personal den Studierenden pro bono beratend zur Seite steht oder sogar auf Kosten seines Privatlebens, nur um nicht als unzugänglich oder unfreundlich von den Studierenden wahrgenommen zu werden. Darüber hinaus führt dieser Status quo zu einer ineffizienten Zeiteinteilung sowohl für die Studierenden als auch für das akademische Personal und zu einer unnötigen Arbeitsbelastung für Letztere bei der Terminplanung.

Diese verwaltungstechnische Ineffizienz zwingt das akademische Personal zu zusätzlichen, oft unbezahlten Aufgaben, während die Studierenden über die Ergebnisse im Unklaren gelassen werden und höchstwahrscheinlich kaum mit dem gesamten Prozess zufrieden sind. Dies ist ein hervorragendes Beispiel dafür, dass die Folgen eines schlecht funktionierenden Terminvereinbarungssystems auf Gruppen verlagert werden, die in der akademischen Hierarchie niedriger angesiedelt sind (Studierende und akademisches Personal), anstatt eine von den Universitätsbehörden initiierte Top-Down-Lösung zu finden. Daher bietet die Technologie hier eine praktikable, befähigende und umsetzbare Lösung, die die Arbeitsbelastung des akademischen Personals erheblich verringern und gleichzeitig eine höhere Zufriedenheit und Effizienz bei der Nutzung durch die Studierenden gewährleisten würde.

Die Studenten wenden sich häufig an die Studentenkanzlei mit der Frage, wann und wo die Sprechstunden stattfinden. Da es keine verlässliche Quelle gibt, verlassen sie sich auf ihre Erfahrungen und verweisen die Studenten oft an falsche Orte und zu falschen Zeiten. Nach den Regeln der akademischen Einrichtung muss jede Änderung der Sprechzeiten vom Prodekan genehmigt werden. Die Genehmigungen werden in direkter Kommunikation zwischen Lehrer und Prodekan erteilt, und das Studentenbüro hat keine Informationen darüber. Der Prodekan wünscht daher, dass das Studentenbüro die geänderten Sprechzeiten einträgt.

## 22.3 Anwendungsfälle

Die Buchungsanwendung Office Hours ermöglicht den Zugang für die Verwaltung, das studentische Büropersonal, die Studierenden und das akademische Personal (auch Lehrkräfte genannt), die jeweils über unterschiedliche Berechtigungen verfügen:

- Das Management erhält auf dem Management-Dashboard zusammenfassende Informationen (Anzahl der Sprechstunden nach Jahr und Monat, Top 10 der Lehrer nach Anzahl der Schüler, Top 5 der überbuchten Lehrer und Top 5 der Lehrer ohne Schüler während der Sprechstunden).
- Die Mitarbeiter des Studentensekretariats greifen auf die interaktive Meldung der Sprechzeiten zu und verlegen die Sprechzeiten.
- Die Studenten haben Zugriff auf den Kalender mit ihren Terminen, auf den Kalender mit allen Sprechstunden der Lehrkräfte, die im Studiengang des Studenten tätig sind, die Einschreibung über den Kalender und einen detaillierten interaktiven Bericht über die Einschreibung in die Sprechstunden der Studenten.

- Akademische Mitarbeiter/Lehrer können einen detaillierten interaktiven Bericht über die Sprechzeiten der Lehrer (Dienstzeiten) und einen Kalender mit den Sprechzeiten der Lehrer (Dienstzeiten) abrufen und Sprechzeiten ohne Einschreibungen neu planen.

### 22.3.1 Narrative Beschreibung

Der Kürze halber werden wir nur zwei von mehreren Anwendungsfällen vorstellen:

- Um die Sprechstunden des Lehrers zu verschieben, für die keine Schüler eingeschrieben sind, muss sich der Lehrer einloggen, die geplanten Sprechstunden einsehen, für die keine Schüler eingeschrieben sind, und diese einzeln auf andere Termine „verschieben“.
- Die Anmeldung zu den Sprechstunden besteht aus einer Übersicht über den Kalender (alle Sprechstunden, die von allen Lehrkräften angeboten werden, die am Studiengang des Studenten beteiligt sind), der Auswahl der gewünschten Sprechstunde, der Eingabe des Zwecks und der Bestätigung der Anmeldung. Es ist nicht erlaubt, sich mehr als einmal pro geplante Sprechstunde anzumelden.

### 22.3.2 Semi-strukturierte Beschreibung

Der erste Anwendungsfall ist die Neuplanung von Sprechstunden durch den Lehrer, wenn kein Schüler eingeschrieben ist (siehe Tabelle 22.1). Wir beschränken die Neuplanung auf diesen Typ, da kein Schüler davon betroffen ist. Der zweite Anwendungsfall zielt auf die Einschreibung von Schülern für Sprechstunden ab (siehe Tabelle 22.2). Auch hier soll die Anwendung die Mehrfachanmeldung zu einem Sprechstundentermin verhindern.

### 22.3.3 Anwendungsfalldiagramm

Abbildung 22.1 zeigt die identifizierten Anwendungsfälle. Beachten Sie, dass der Anwendungsfall „Anmeldung“nur einmal pro Benutzersitzung durchgeführt wird und dass keiner der anderen Anwendungsfälle vor dem erfolgreichen Abschluss der „Anmeldung“ausgeführt werden kann. Drei Anwendungsfälle, nämlich „Verschiebung der Sprechzeiten“, „Einschreibung in die Sprechzeiten“und „Verschiebung der Sprechzeiten ohne Einschreibung“sind abhängig und mit der Assoziation „verlängern“ gekennzeichnet. Diese Anwendungsfälle werden auf Anfrage ausgeführt, nachdem der übergeordnete Anwendungsfall abgeschlossen ist.

Tabelle 22.1: Beschreibung des Anwendungsfalls: Verlegung der Sprechstunden durch den Lehrer.

| **Keyword** | **Wert** |
|---|---|
| ID: | *Ch22-01* |
| Titel: | *Verlegung der Sprechstunden durch den Lehrer* |
| Beschreibung: | *Die Lehrkraft nutzt die APEX-Anwendung, um Sprechstunden zu verlegen, in denen kein Schüler eingeschrieben ist.* |
| Primärer Akteur: | *Lehrkraft* |
| Voraussetzungen: | Zugang zu einem Webbrowser auf einem mobilen Gerät oder einem PC, der Nutzer verfügt über die Anmeldedaten und Berechtigungen des Lehrers, die Website der Anwendung ist verfügbar |
| Nachbedingungen: | Daten gespeichert in der Datenbanktabelle ch22_duty_hours |
| *Haupt* | - |
| Erfolgs-Szenario: | 1. Lehrer öffnet den Webbrowser und meldet sich bei der Anwendung an<br>2. Wählen Sie „Dienstzeiten neu planen“, wenn keiner eingetragen ist.<br>3. Klicken Sie auf den als nicht eingetragen markierten Steckplatz<br>4. Geben Sie auf dem Formular die Daten für den neuen Termin ein (Ort, Anfangs- und Endzeit)<br>5. Prüfen und bestätigen Sie die Neuplanung durch Anklicken der entsprechenden Schaltfläche |
| Erweiterungen: | - |
| Häufigkeit der Nutzung: | *Ungefähr 4 von 30 Pflichtterminen in einem akademischen Jahr.* |
| Status: | [Fertiggestellt] |
| Eigentümer: | *Benutzer mit den Rechten eines Lehrers* |
| Priorität: | *moderat* |

Tabelle 22.2: Beschreibung des Anwendungsfalls: Einschreibung von Studenten für Sprechstunden.

| **Keyword** | **Wert** |
|---|---|
| ID: | *Ch22-02* |
| Titel: | *Einschreibung von Studenten für Sprechstunden* |
| Beschreibung: | *Der Student verwendet die APEX-Anwendung, um sich für die Sprechstunden einzutragen. Der Kalender enthält alle geplanten Sprechzeiten für alle Lehrkräfte, die mit den Studiengängen (Abteilungen) der Studierenden befasst sind. Normalerweise ist ein Student für einen Studiengang eingeschrieben, aber in Ausnahmefällen können zwei oder mehr Studiengänge zugelassen werden. Die Anwendung verhindert die Mehrfacheinschreibung eines Studenten für bestimmte Sprechstunden.* |
| Primärer Akteur: | *Student* |
| Voraussetzungen: | Zugang zu einem Webbrowser auf einem mobilen Gerät oder einem PC, der Nutzer verfügt über die Anmeldedaten und Berechtigungen des Lehrers, die Website der Anwendung ist verfügbar |
| Nachbedingungen: | in der Datenbank in der Tabelle ch22_dh_slot gespeicherte Daten |
| *Haupt* | - |
| Erfolgs-Szenario: | 1. Der Student öffnet den Webbrowser und meldet sich bei der Anwendung an<br>2. Anmeldung über den Kalender<br>3. Klicken Sie auf den gewünschten Slot (Slots sind farblich gekennzeichnet und zeigen den Status an)<br>4. Auf dem neuen Formular klickt der Student auf die Schaltfläche und bestätigt die Einschreibung. |
| Erweiterungen: | - |
| Häufigkeit der Nutzung: | *Mehr als 1000 Studierende nutzen die Sprechstunden von 10 bis 15 Lehrkräften im Durchschnitt 2 Mal pro Semester.* |
| Status: | [Fertiggestellt] |
| Eigentümer: | *Student* |
| Priorität: | *hoch* |

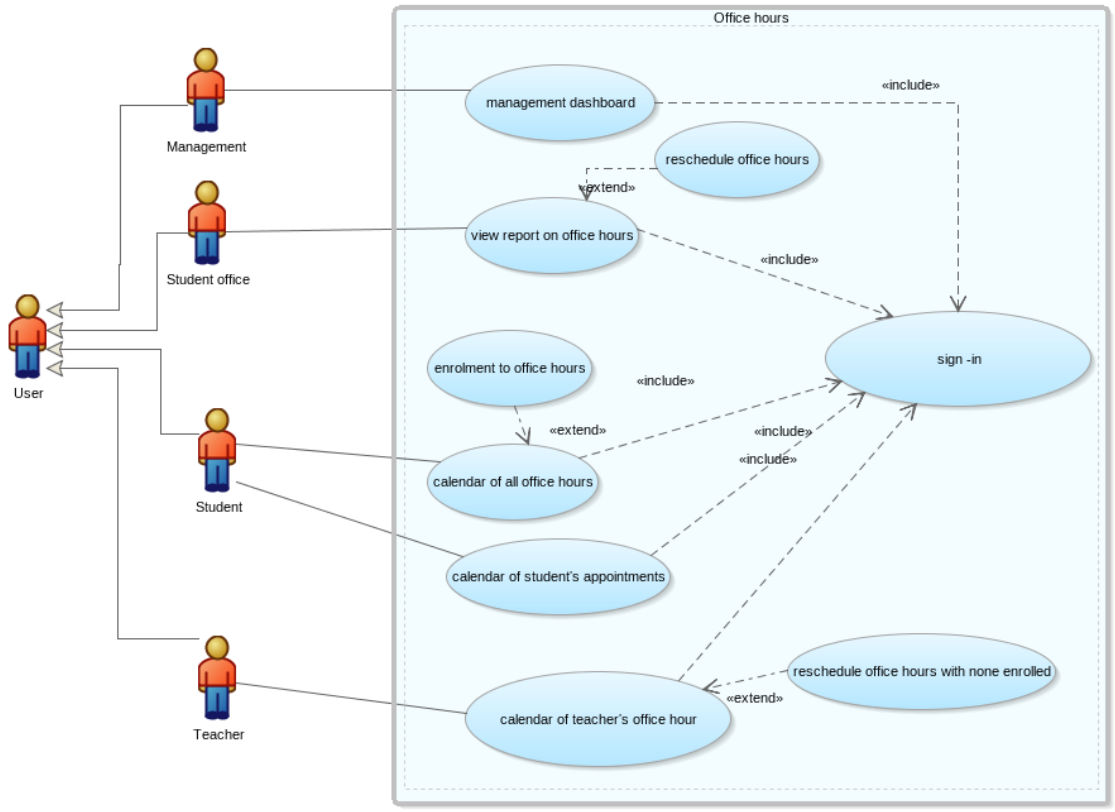



Abbildung 22.1: Anwendungsfalldiagramm.

## 22.4 Datenmodell

### 22.4.1 Narrative Beschreibung des Datenmodells

Das logische Datenmodell besteht aus zehn Entitäten:

- **ch22_dh_slot** ein Studenten-Lehrer-Gespräch führen;
- **ch22_duty_hour** Vorab zugewiesene Lehrerzeitscheiben (Sprechstunden), die auf mehrere Zeitfenster alias Studenten-Lehrer-Sitzungen hinweisen;
- **ch22_purpose** speichert Gründe und durchschnittliche Dauer von Studenten-Lehrer-Gesprächen;
- **ch22_student_dept** verweist auf einen Studiengang eines Studierenden. Ausnahmsweise können Studierende in einem akademischen Jahr in mehr als einem Studiengang (synonym für Fachbereich) eingeschrieben sein;
- **ch22_student** ist eine Person, die an einer Universität studiert und in den Studiengang eingebunden ist (synonym für Fachbereich);
- **ch22_teacher_dept** verweist auf einen Studiengang (auch Fachbereich genannt) eines Lehrers. Eine Lehrkraft ist in der Regel in mehr als einem Studiengang tätig;
- **ch22_teacher** ist eine Person, die akademische Aktivitäten leitet (Unterricht, Laborübungen usw.)
- **ch22_location** ist entweder physisch (R) oder virtuell (I) oder beides (B) - es ist ein Ort, an dem Bürozeiten stattfinden;
- **ch22_department** ist ein Synonym für Studiengang;
- **ch22_acad_cal_umb** enthält Angaben zu Arbeits- und Nichtarbeitstagen. Diese Entität ist mit allen Entitäten mit einigen Datumsattributen verbunden. Wir werden diese Beziehungen jedoch nicht explizit modellieren, da die implizite Beziehung für die Vollständigkeit des Modells ausreicht.

### 22.4.2 Logisches Datenmodell

Das logische Datenmodell wird in 22.2 vorgestellt.

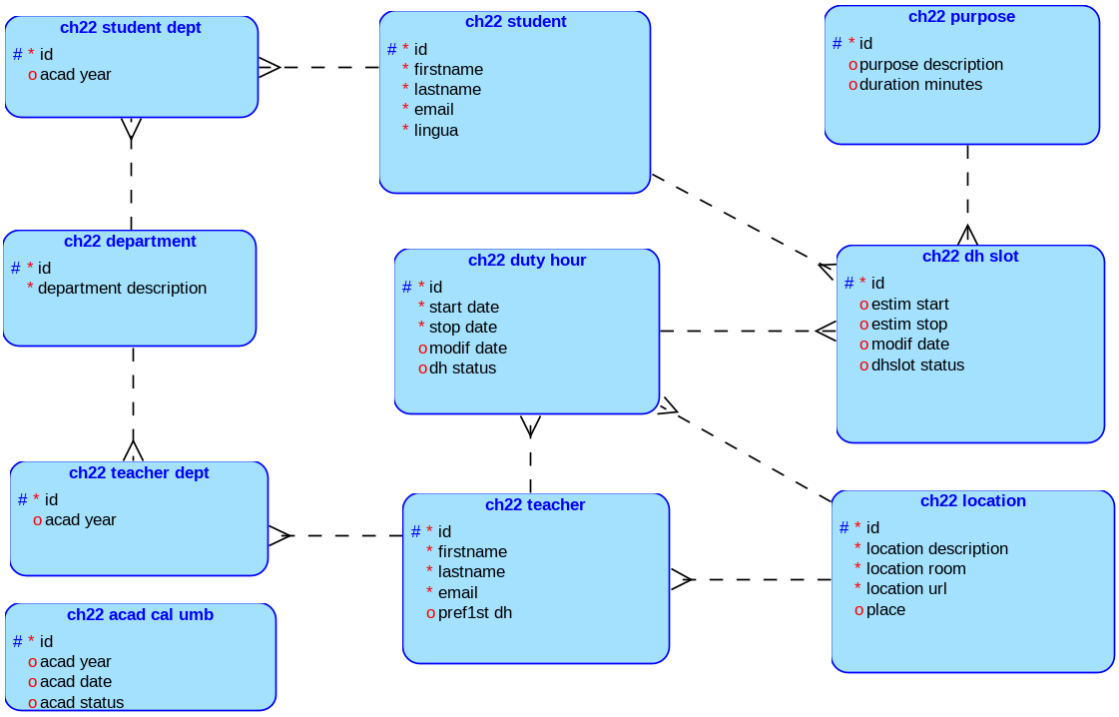



Abbildung 22.2: Logisches Datenmodell.

### 22.4.3 Relationales Datenmodell

Die automatische Umwandlung eines logischen Datenmodells in ein relationales Datenmodell in Oracle SQL Data Modeler wird durch die Funktion *Engineering to relational* bereitgestellt. Das Ergebnis, ein relationales Datenmodell, das als SQL-Skript exportiert werden kann, ist in Abbildung 22.3 dargestellt.

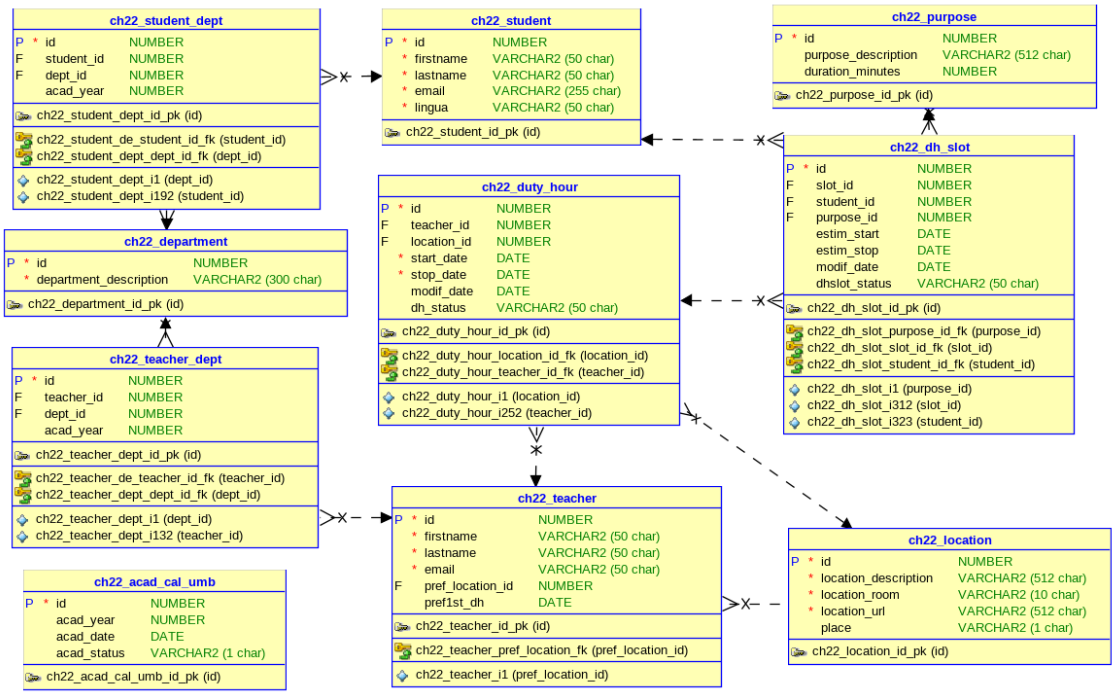



Abbildung 22.3: Relationales Datenmodell.

Relationale Datenmodelle können ohne SQL Data Developer Data Modeler entwickelt werden. Wir können Quick SQL in APEX verwenden.

### 22.4.4 Quick SQL

Das vorgestellte Quick SQL (siehe Abbildung 22.4) enthält alle notwendigen Informationen für die Erstellung eines SQL-Skripts.

```
ch22_acad_cal_umb
    acad_year num
    acad_date date
    acad_status vc1 /check y n
ch22_department
    department_description vc300 /nn
ch22_purpose
    purpose_description vc512
    duration_minutes num /check 10 15 20 25 30 45 60 /default 15
ch22_location
    location_description vc512 /nn
    location_room vc10 /nn /default R1
    location_url vc512 /nn /default https://beeapex.eu/bbb
    place vc1 /check R,I,B /default I
ch22_teacher
    firstname vc50 /nn
    lastname vc50 /nn
    email vc50 /nn
    pref_location_id num /fk ch22_location
    pref1st_dh date
ch22_teacher_dept
    teacher_id num /fk ch22_teacher
    dept_id num /fk ch22_department
    acad_year num
ch22_student_dept
    student_id num /fk ch22_student
    dept_id num /fk ch22_department
    acad_year num
ch22_duty_hour
    teacher_id /fk ch22_teacher
    location_id /fk ch22_location
    start_date date /nn
    stop_date date /nn
    modif_date date
    dh_status vc50
ch22_dh_slot
    slot_id /fk ch22_duty_hour
    student_id /fk ch22_student
    purpose_id /fk ch22_purpose
    estim_start date
    estim_stop date
    modif_date date
    dhslot_status vc50
ch22_student
    firstname vc50 /nn
    lastname vc50 /nn
    email vc255 /nn
    lingua vc50 /nn
```

Abbildung 22.4: Datenmodell mit Quick SQL beschrieben.

Relationale Datenmodelle können ohne SQL Data Developer Data Modeler entwickelt werden. Wir können Quick SQL in APEX verwenden.

### 22.4.5 SQL-Skript zum Erstellen von Tabellen

Das SQL-Skript zum Erstellen von Tabellen ist in den Lernmaterialien als Datei CH22CREATE.sql enthalten. Das Skript zum Erstellen von Tabellen ist auch in der verpackten Anwendung in den Lernmaterialien verfügbar.

### 22.4.6 Query builder in APEX

Diese spezielle Anwendung verwendet ziemlich lange Abfragen, die APEX-Anfänger abschrecken können. Aber es gibt Abhilfe - Query Builder, der in APEX eingebettet ist (SQL Workshop > Utilities > Query Builder). Er kann jedem helfen, durch Klicken und Ziehen korrekte Abfragen zu erstellen. Figure 22.5 zeigt eine Beispielabfrage für den Kalender der eingeschriebenen Studenten.

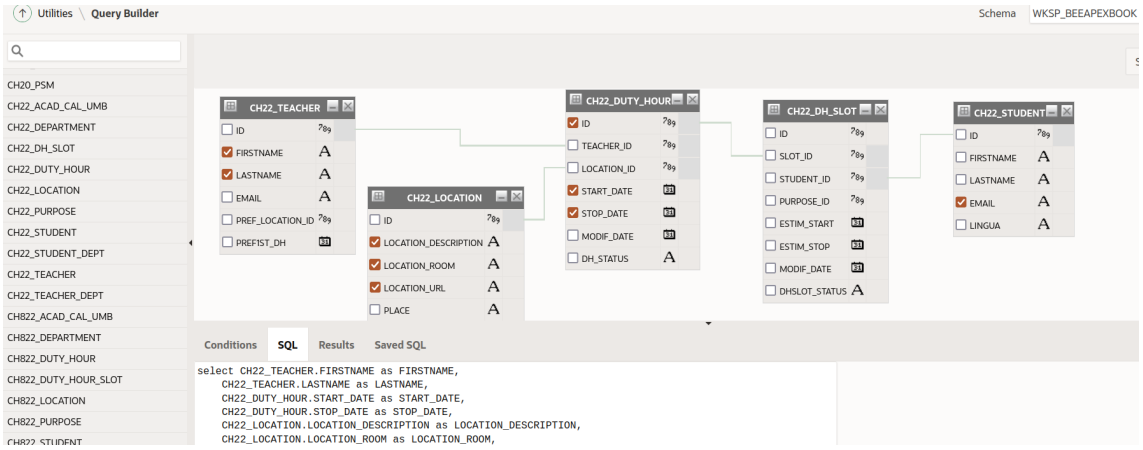



Abbildung 22.5: Ein Beispiel für die Verwendung des Query Builders.

Die generierte SQL-Anweisung hat die folgende Form:

```
select CH22_TEACHER.FIRSTNAME as FIRSTNAME,
       CH22_TEACHER.LASTNAME as LASTNAME,
       CH22_DUTY_HOUR.START_DATE as START_DATE,
       CH22_DUTY_HOUR.STOP_DATE as STOP_DATE,
       CH22_LOCATION.LOCATION_DESCRIPTION as
              LOCATION_DESCRIPTION,
       CH22_LOCATION.LOCATION_ROOM as LOCATION_ROOM,
       CH22_LOCATION.LOCATION_URL as LOCATION_URL,
       CH22_DUTY_HOUR.ID as ID,
       CH22_STUDENT.EMAIL as EMAIL
from CH22_DH_SLOT CH22_DH_SLOT,
       CH22_DUTY_HOUR CH22_DUTY_HOUR,
       CH22_STUDENT CH22_STUDENT,
       CH22_TEACHER CH22_TEACHER,
       CH22_LOCATION CH22_LOCATION
       where CH22_DH_SLOT.SLOT_ID=CH22_DUTY_HOUR.ID
       and CH22_DUTY_HOUR.LOCATION_ID=CH22_LOCATION.ID
       and CH22_DUTY_HOUR.TEACHER_ID=CH22_TEACHER.ID
       and CH22_STUDENT.ID=CH22_DH_SLOT.STUDENT_ID
```

Durch Hinzufügen einer zusätzlichen Bedingung am Ende „and CH22_TEACHER.EMAIL = :APP_USER“ wird die Abfrage eingeschriebene Schüler für diesen spezifischen Lehrer zurückgeben. Der Query Builder ist eine notwendige Funktion für absolute Anfänger, aber auch erfahrene Benutzer können davon profitieren.

Die in dieser Anwendung verwendeten Abfragen nutzen den Verkettungsoperator (||), die EXTRACT-Funktion, allgemeine Tabellenausdrücke, den Abruf der ersten 10 oder 5 Zeilen und die benutzerdefinierte Funktion CH22_dh_utilization. Für die Leser dieses Buches kann es von Vorteil sein, zuerst die verpackte Anwendung zu installieren und die in Kalendern, Dashboards und Berichten verwendeten Abfragen zu untersuchen.

## 22.5 Anwendungsschnittstellen

Anwendungen haben eine gemeinsame Seite - die Startseite. Diese Seite bietet einige grundlegende Informationen über die Anwendung (siehe Figure 22.6).

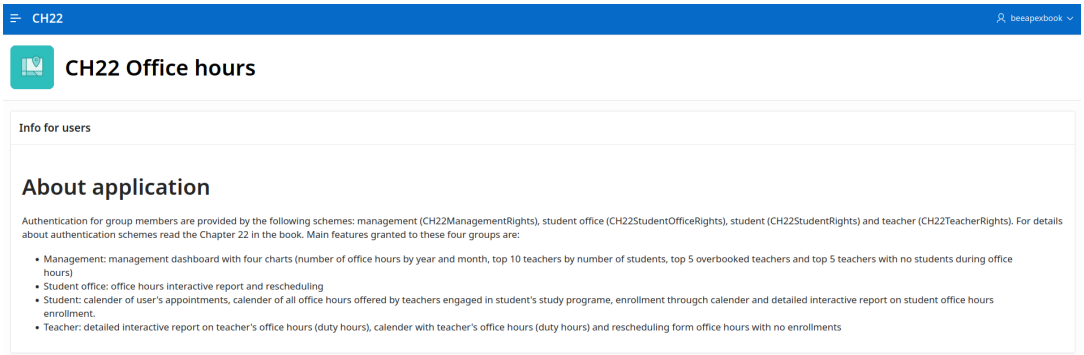



Abbildung 22.6: Startseite der Anwendung.

Die Verwaltung, die Mitarbeiter des Studentenbüros, die Studenten und die Lehrer werden nur mit angepassten Schnittstellen bedient.

### 22.5.1 Schnittstellen der Verwaltungsanwendungen

Benutzer mit Verwaltungsrechten können auf das Management-Dashboard zugreifen (siehe Figure 22.7). Es enthält Diagramme zur Unterstützung der Entscheidungsfindung in Bezug auf Sprechstunden und Lehrer:

- Anzahl der Bürostunden nach Jahr und Monat
- Top-10-Lehrer nach Anzahl der Schüler
- Top 5 der überlasteten Lehrer und
- Top 5 der Lehrer, die während der Sprechstunden keine Schüler haben

Gegenwärtig sind keine weiteren Anforderungen an die Verwaltung festgelegt.

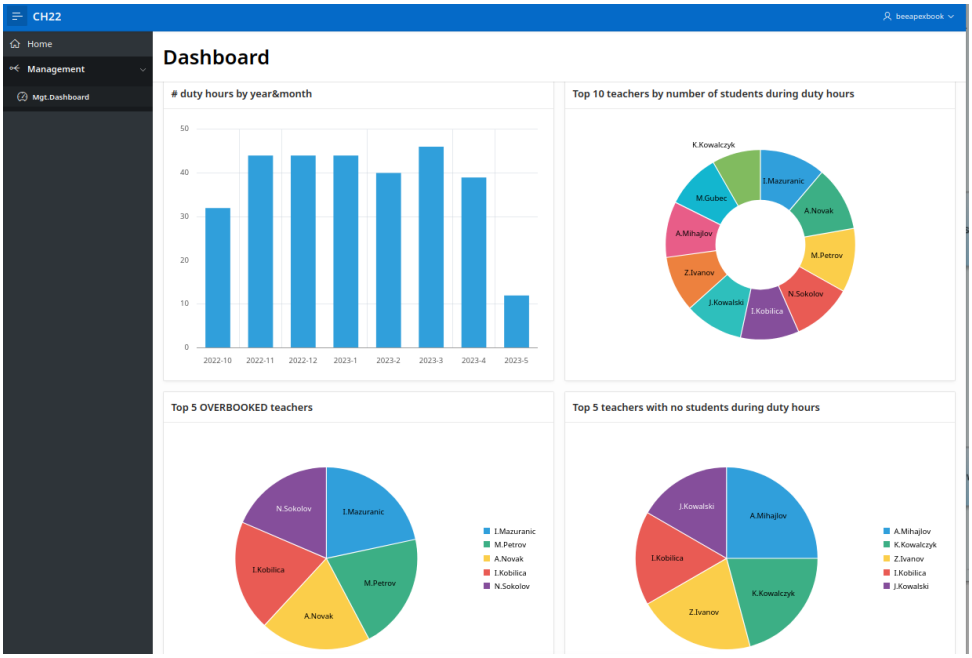



Abbildung 22.7: Management dashboard.

### 22.5.2 Schnittstellen für Studentenbüros

Benutzer mit Rechten für Studentenbüros können auf den interaktiven Bericht über die Sprechzeiten zugreifen und beliebige Sprechzeiten neu planen. Figure 22.8 zeigt den interaktiven Bericht mit aktiviertem Filter (nur ein Lehrer wird angezeigt). Als nächstes zeigt Figure 22.9 ein Formular, das eine Neuplanung ermöglicht. Der Benutzer kann den Lehrer, den Ort sowie das Start- und Enddatum der ausgewählten Sprechstunden ändern. Figure 22.10 zeigt den Bericht nach der Neuplanung. Zwei Felder werden automatisch aktualisiert: Änderungsdatum und Status der Sprechstunde. Letzteres ist eine verkettete Zeichenfolge, die auch den Benutzernamen enthält, der die Änderung vorgenommen hat.

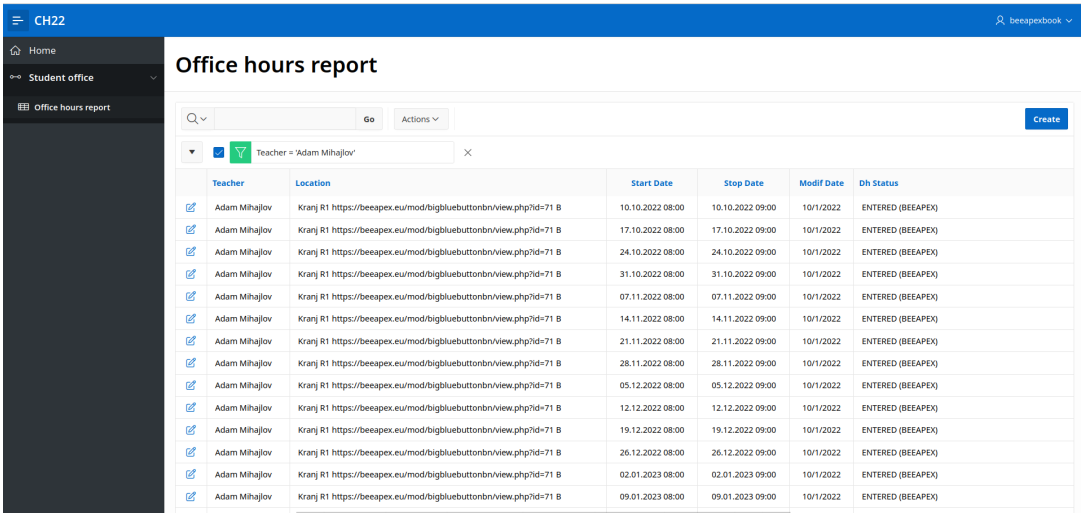



Abbildung 22.8: Interaktiver Bericht über die Bürozeiten für das Studentenbüro.

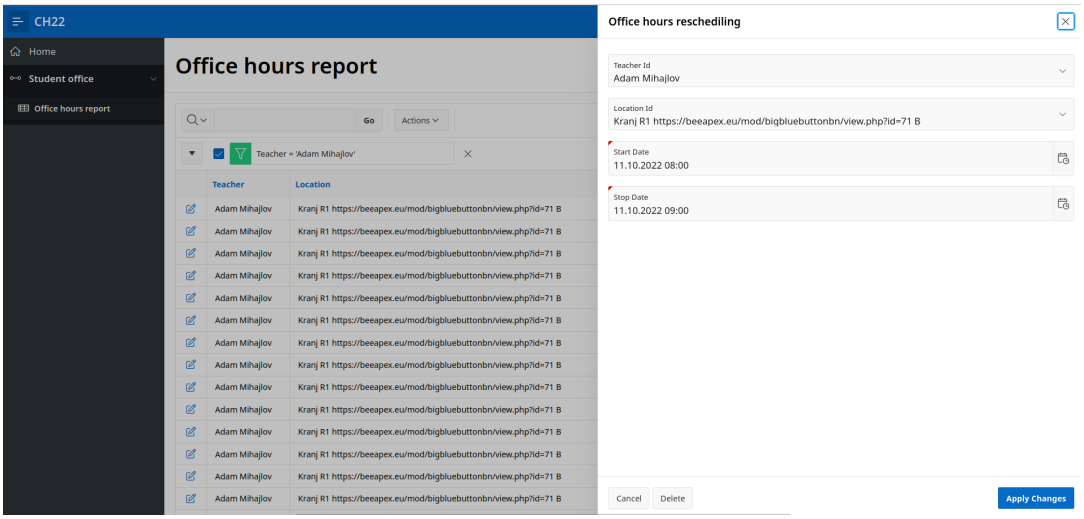



Abbildung 22.9: Umplanungsformular für das Studentenbüro.

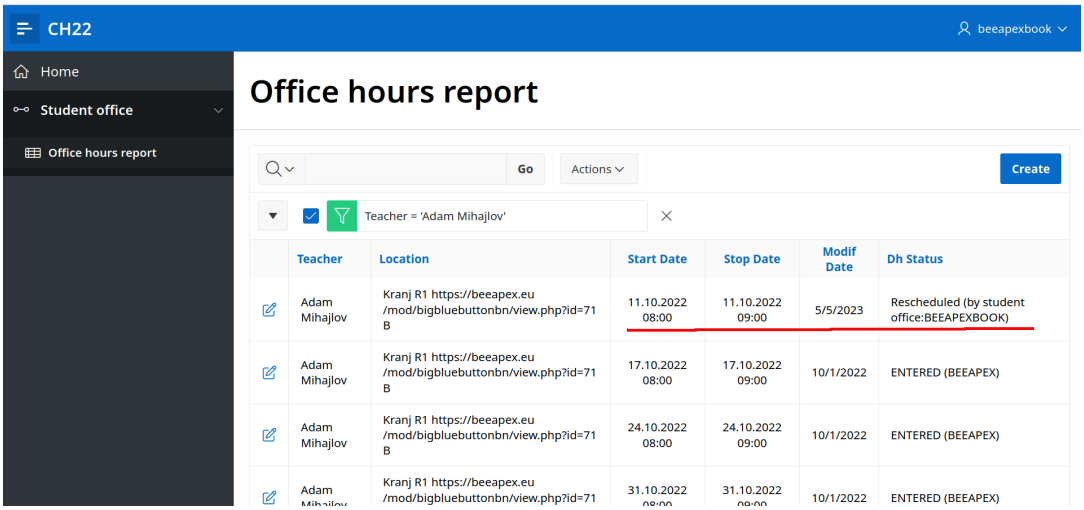



Abbildung 22.10: Interaktiver Bericht über die Bürozeiten nach der Umplanung.

### 22.5.3 Schnittstellen für Studentenbewerbungen

Kalendereinträge sind je nach Nutzung farblich gekennzeichnet. KEINE Einschreibung wird grün dargestellt, VERFÜGBAR wird blau dargestellt und ÜBERBUCHT wird braun dargestellt. In diesem Moment kann sich ein Student einschreiben, auch wenn der Status ÜBERBUCHT festgestellt wurde. Benutzer mit Studentenrechten können darauf zugreifen:

- Kalender seiner/ihrer Termine (siehe Figure 22.11).
- Kalender aller Sprechstunden der am Studiengang beteiligten Lehrkräfte (siehe Figure 22.12).
- Eintragung in die Sprechzeiten über den Kalender. Zuerst wählt der Benutzer die gewünschte Sprechstunde aus (siehe Figure 22.13). Im Formular wählt der Student nur den Zweck aus (siehe Figure 22.14) und klickt auf die Schaltfläche „Enroll“. Wenn die Einschreibung erfolgreich war, erhalten die Studierenden eine Rückmeldung (siehe Figure 22.15). Die Überprüfung der Einschreibung über den Menüpunkt „View my appointments“ bestätigt die erfolgreiche Einschreibung (siehe Figure 22.16).
- Detaillierter interaktiver Bericht, der alle Schülertermine in Tabellenform darstellt (siehe Figure 22.17).

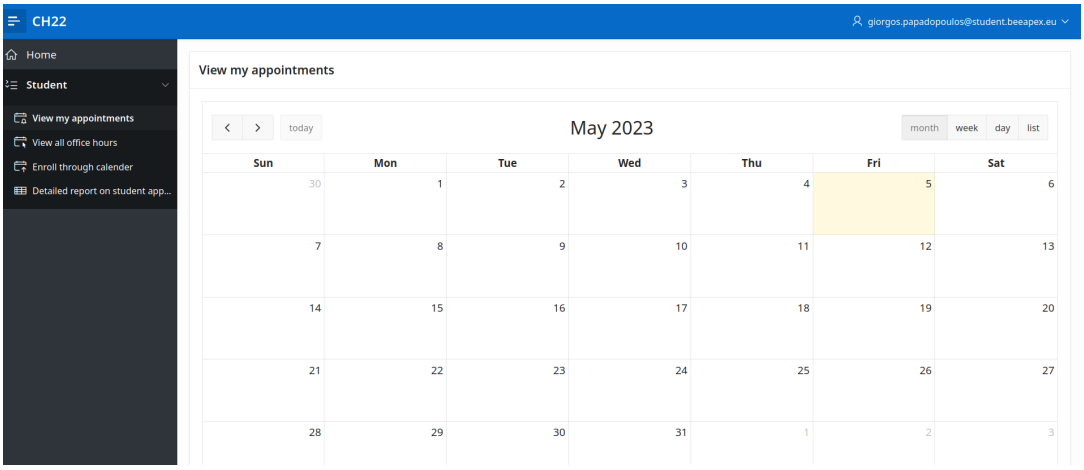



Abbildung 22.11: Terminkalender für Studenten.

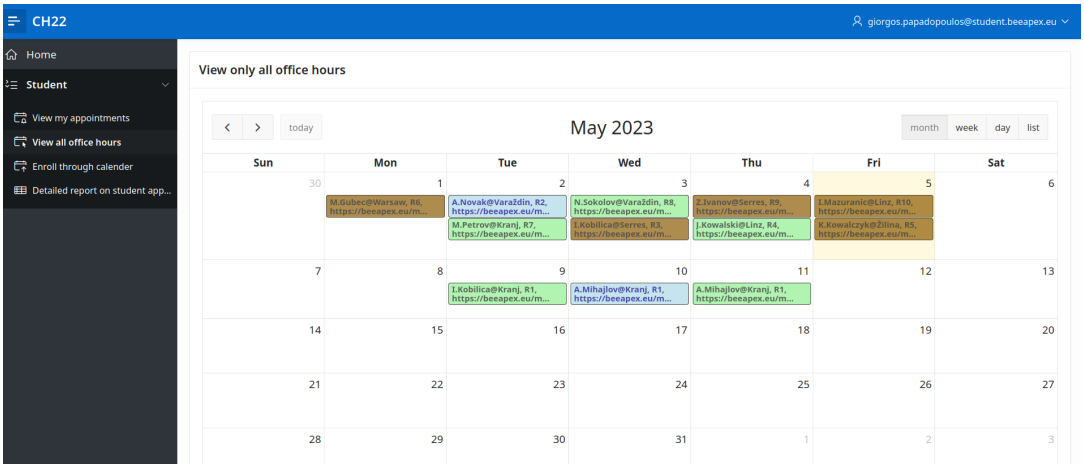



Abbildung 22.12: Kalender aller Sprechstunden, die von den Lehrkräften angeboten werden, die am Studiengang des Studenten beteiligt sind.

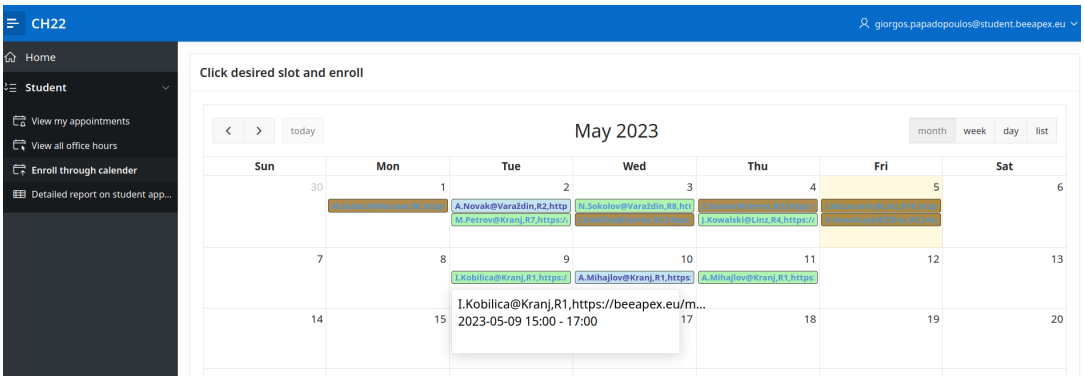



Abbildung 22.13: Eintragung in die Sprechzeiten über den Kalender - Auswahl des Kalenderplatzes.

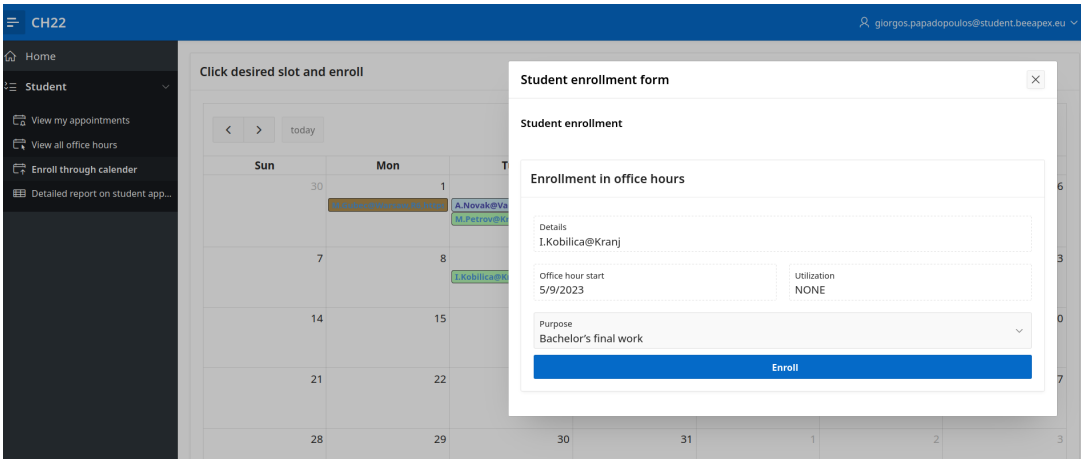



Abbildung 22.14: Eintragung in die Sprechzeiten über den Kalender - Auswahl des Zwecks.

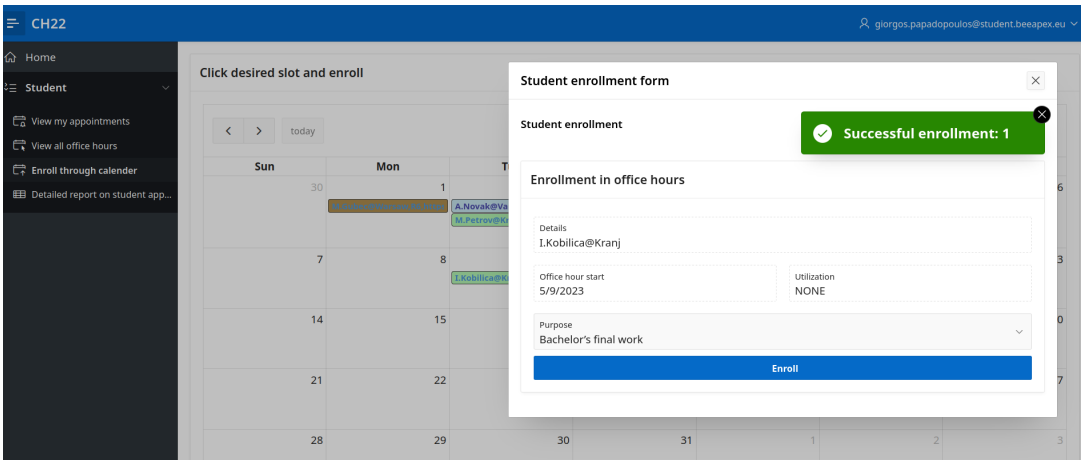



Abbildung 22.15: Eintragung in die Sprechstunde über den Kalender - erfolgreiche Eintragung.

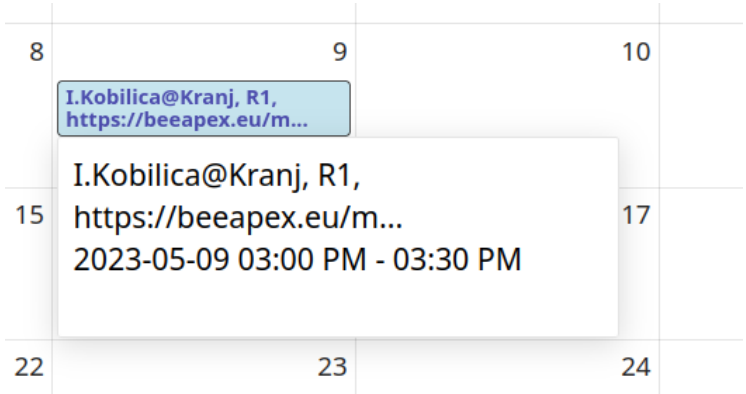



Abbildung 22.16: Überprüfung der Einschreibung über den Menüpunkt „Viev my appointments“.

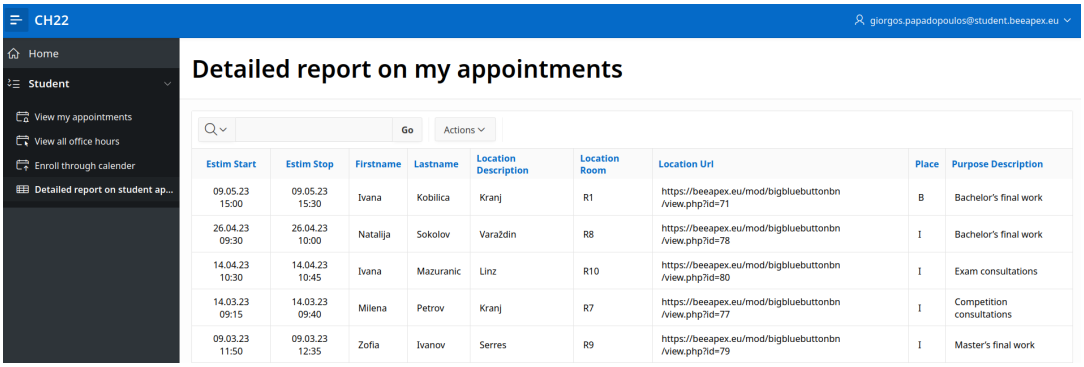



Abbildung 22.17: Detaillierter interaktiver Bericht über alle Termine der Studenten.

### 22.5.4 Bewerbungsschnittstellen für Lehrkräfte

Kalendereinträge sind je nach Auslastung farblich gekennzeichnet. Die Farben der Stati sind identisch mit denen der Schülerschnittstelle (KEINE ist grün, VERFÜGBAR ist blau und ÜBERBUCHT ist braun). Benutzer mit Lehrerrechten können:

- einen detaillierten Bericht über die Termine der Lehrer anzeigen (siehe Abbildung 22.18).
- den Lehrerkalender anzeigen. Der Name des Schülers, die Sprache und der Zweck werden für jeden eingeschriebenen Schüler angezeigt (siehe Abbildung 22.19).
- Sprechstunden neu planen, wenn keiner eingetragen ist. Zunächst wird der Kalender der Sprechstunden mit dem Status NONE angezeigt (siehe Abbildung 22.20). Durch Anklicken des Elements im Kalender wird ein modales Formular geöffnet, in das Start- und Enddatum (sowie Stunden und Minuten) eingegeben werden (siehe Abbildung 22.21). Durch Anklicken von „Apply changes“ werden die Bürozeiten auf das gewünschte neue Datum verschoben (siehe Abbildung 22.22).

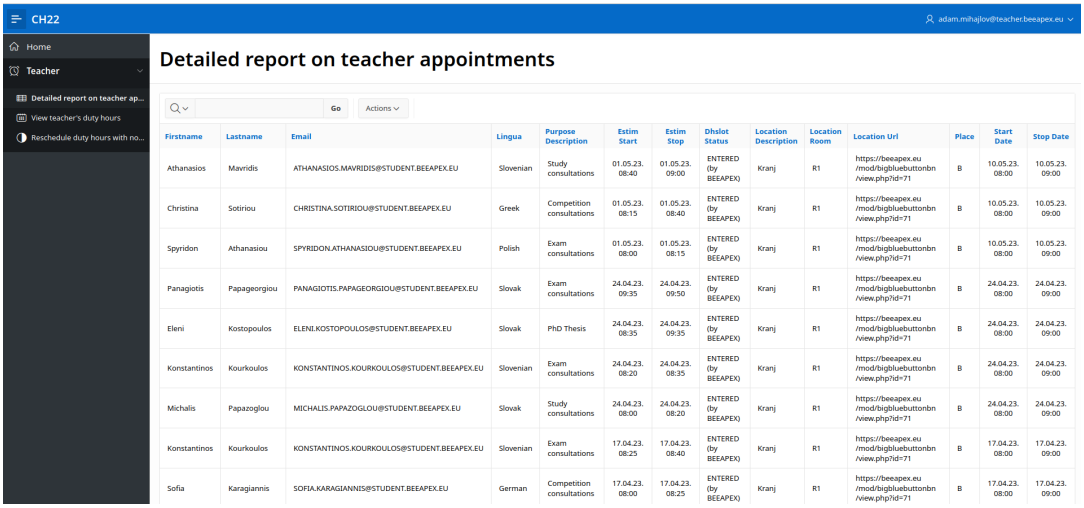



Abbildung 22.18: Detaillierter Bericht über die Termine von Lehrern

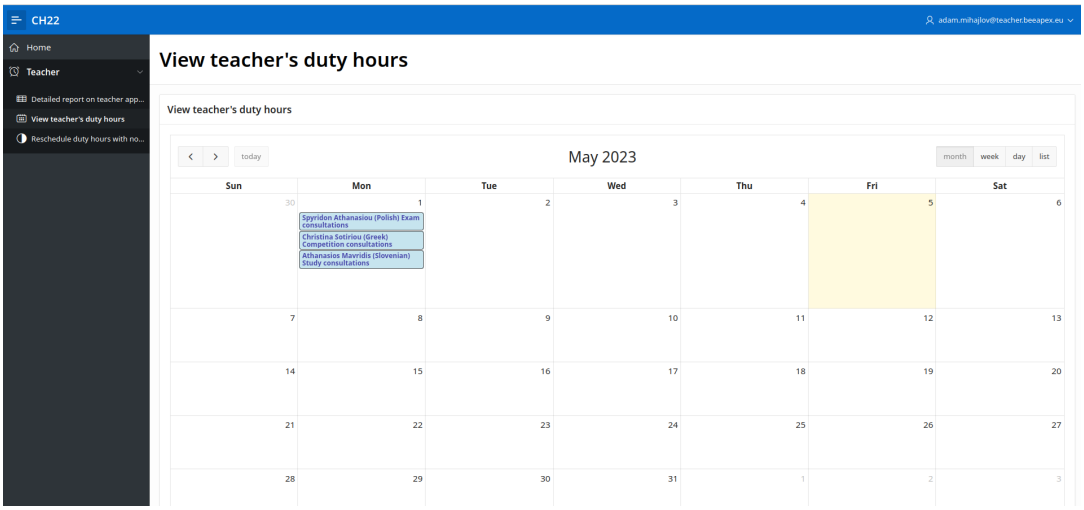



Abbildung 22.19: Lehrerkalender mit sichtbaren Schülernamen, ihren Sprachen und Zielen.

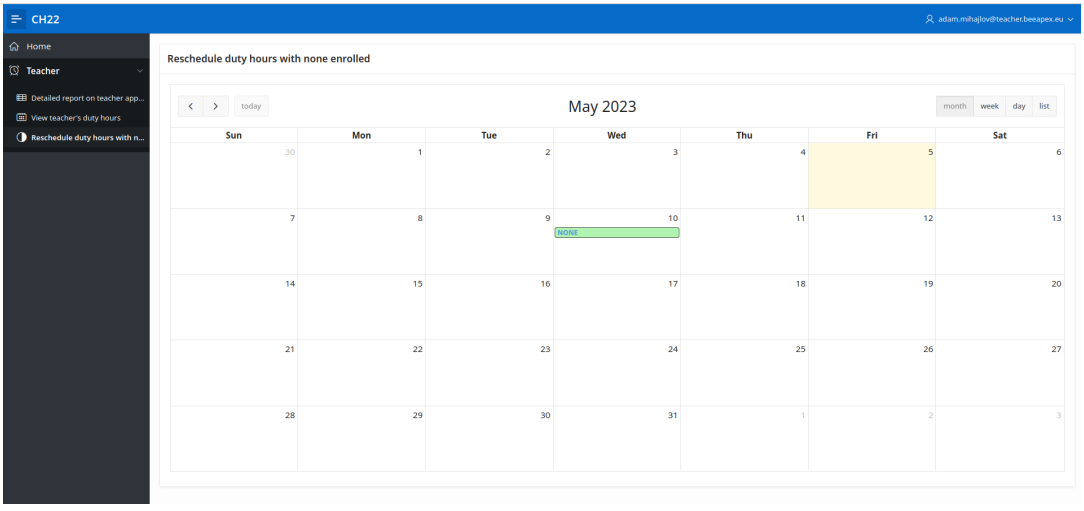



Abbildung 22.20: Neuplanung von Lehrersprechstunden mit NONE eingeschrieben - die Kalenderansicht.

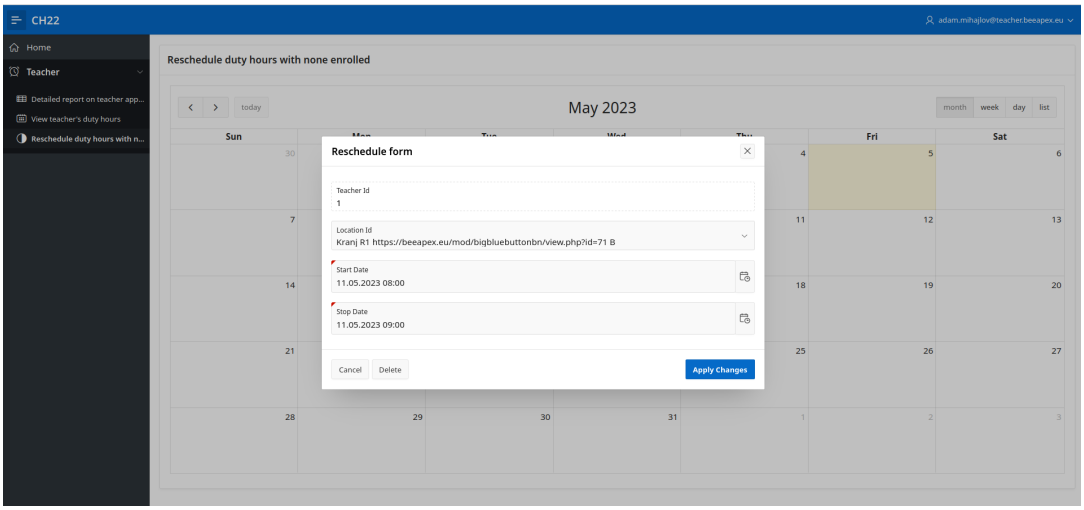



Abbildung 22.21: Umplanung der Lehrersprechstunde mit NONE eingeschrieben - neues Datum eingegeben.

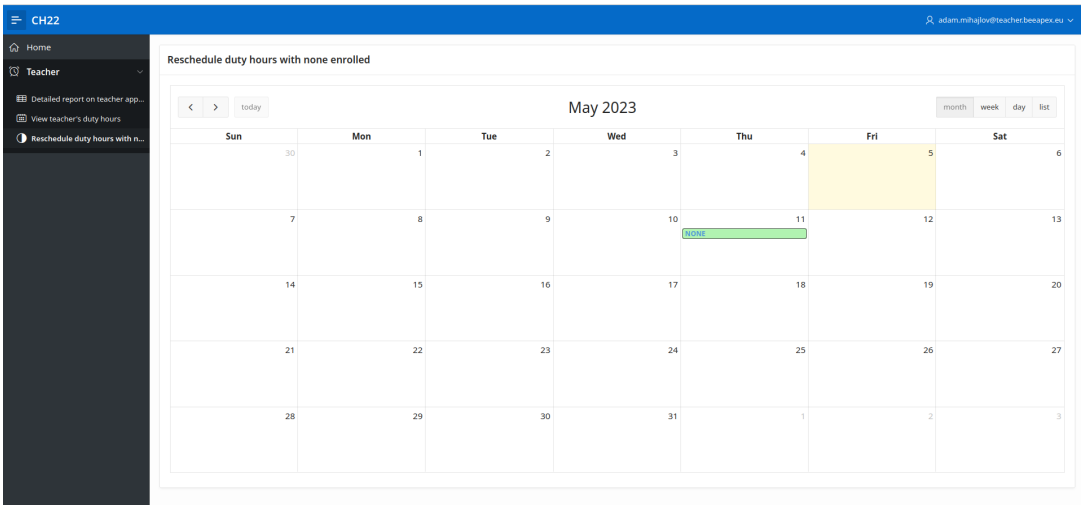



Abbildung 22.22: Die Ergebnisse der Neuplanung von Lehrersprechstunden, für die NONE eingeschrieben ist.

## 22.6 Ergänzendes Lernmaterial

Sie können das folgende zusätzliche Lernmaterial finden:

- exportierte Anwendung, Skripte zum Erstellen von Tabellen, Einfügen von Daten, Erstellen von PL/SQL-Funktionen und Löschen von Tabellen
- Videoanleitungen: a) Installation der verpackten Anwendung und Anlegen von Benutzern (Schüler und Lehrer) und b) Entwicklung der Anwendung von Grund auf

Alle zusätzlichen Lernmaterialien sind auf der öffentlichen BeeAPEX-Projektseite verfügbar. Melden Sie sich als Gastbenutzer an (kein Passwort erforderlich). Das Lehrbuch finden Sie im Abschnitt Books, die Skripte im Ordner Part 2 > Chapter22 im Abschnitt Scripts und die Videoanleitungen in der Sammlung der Videoanleitungen. Material für Kurzkurse finden Sie in der Sektion Short courses.

### 22.6.1 Exportierte Anwendungen

Die exportierte Anwendung ist verpackt. Bei der Installation werden Tabellen, Indizes und Funktionen erstellt sowie Daten eingefügt. Bei der Deinstallation werden alle Datenbankobjekte, die in dieser Anwendung verwendet werden, entfernt. Die verpackte Anwendung hat eine eingebettete CSS-Datei (ch22_cal.css), die von den meisten Kalenderseiten als #APP_FILES#ch22_cal#MIN#.css referenziert wird:

```
.fc-event .fc-content div.fc-time { display: none;}

.fc-event.my-cal-blue {
background-color: lightblue;
border: 0.5pt solid black;
opacity: 0.7;
}

.fc-event.my-cal-blue .fc-event-title {
color: darkblue;
font-weight: bold;
}

.fc-event.my-cal-orange {
background-color: orange;
border: 0.5pt solid black;
opacity: 0.7;
}

.fc-event.my-cal-orange .fc-title {
color: darkred;
font-weight: bold;
}

.fc-event.my-cal-dark-orange {
background-color: #8B5A00;
border: 0.5pt solid black;
opacity: 0.7;
}

.fc-event.my-cal-dark-orange .fc-title {
color: white;
font-weight: bold;
}

.fc-event.my-cal-white {
background-color: white;
border: 0.5pt solid black;
opacity: 0.7;
}

.fc-event.my-cal-white .fc-title {
color: black;
font-weight: bold;
```

```
}

.fc-event.my-cal-green {
background-color: lightgreen;
border: 0.5pt solid black;
opacity: 0.7;
}

.fc-event.my-cal-green .fc-title {
color: darkgreen;
font-weight: bold;
}
```

Das Management-Dashboard verwendet die folgenden Abfragen:

```
SELECT
        EXTRACT(YEAR FROM start_date) || '-' ||
        EXTRACT(MONTH FROM  start_date) AS "Year and Month",
        COUNT(id) AS count
FROM ch22_duty_hour
        GROUP BY EXTRACT(YEAR FROM start_date),
                EXTRACT(MONTH FROM start_date)
order by 1;

with teacher_appointmets (appointment, teacher) as
(select CH22_DH_SLOT.ID,
substr(CH22_TEACHER.FIRSTNAME,1,1) || '.' ||
        CH22_TEACHER.LASTNAME
from CH22_DH_SLOT CH22_DH_SLOT,
CH22_DUTY_HOUR CH22_DUTY_HOUR,
CH22_TEACHER CH22_TEACHER
where CH22_DUTY_HOUR.ID=CH22_DH_SLOT.SLOT_ID
and CH22_TEACHER.ID=CH22_DUTY_HOUR.TEACHER_ID)
select count(appointment), teacher from teacher_appointmets
        group by teacher
order by 1 desc
fetch  first 10 rows only;

with teacher_dh_statuses (duty_hour_id, teacher,dh_utilization) as
(select CH22_DUTY_HOUR.ID,
substr(CH22_TEACHER.FIRSTNAME,1,1) || '.' ||
        CH22_TEACHER.LASTNAME,
CH22_dh_utilization(CH22_DUTY_HOUR.ID)
from CH22_DUTY_HOUR CH22_DUTY_HOUR,
CH22_TEACHER CH22_TEACHER
where CH22_DUTY_HOUR.TEACHER_ID=CH22_TEACHER.ID and
CH22_dh_utilization(CH22_DUTY_HOUR.ID) = 'OVERBOOKED')
select count(duty_hour_id), teacher from teacher_dh_statuses
group by teacher
order by 1 DESC
fetch  first 5 rows only;

```

```
with teacher_dh_statuses (duty_hour_id, teacher,dh_utilization) as
(select CH22_DUTY_HOUR.ID,
substr(CH22_TEACHER.FIRSTNAME,1,1) || '.' || CH22_TEACHER.LASTNAME,
CH22_dh_utilization(CH22_DUTY_HOUR.ID)
from CH22_DUTY_HOUR CH22_DUTY_HOUR,
CH22_TEACHER CH22_TEACHER
where CH22_DUTY_HOUR.TEACHER_ID=CH22_TEACHER.ID and
CH22_dh_utilization(CH22_DUTY_HOUR.ID) = 'NONE')
select count(duty_hour_id), teacher from teacher_dh_statuses
group by teacher
order by 1 DESC
fetch  first 5 rows only;
```

Siehe andere Abfragen für Seiten, die auf Kalenderelementen in der verpackten Anwendung basieren.

### 22.6.2 Video-Anleitungen

Videoanleitungen zeigen die Installation der verpackten Anwendung und die Entwicklung der Anwendung von Grund auf.

## 22.7 Fragen

1. Wie kann ich einen Benutzer mit Lehrerrechten hinzufügen?
2. Wie würden Sie die Bewerbung verbessern, indem Sie den Studierenden die Möglichkeit geben, sich Notizen für bestimmte Sitzungen zu machen?
3. Wie würden Sie die Abfrageergebnisse auf Zeilen beschränken, deren Datumsspalten gleich oder größer als die aktuelle Zeit sind?

## 22.8 Antworten

1. Verwenden Sie Manage Users and Group. Fügen Sie einen neuen Benutzer hinzu. Verwenden Sie SQL Workshop > Object Browser und wählen Sie die Tabelle CH22_TEACHER. Wählen Sie die Registerkarte „Data“ und „Insert row“. Die E-Mail der eingefügten Zeile muss mit dem erstellten APEX-Benutzer übereinstimmen.
2. Erstellen Sie eine neue Tabelle, d.h. CH22_NOTES mit den Feldern: ID, note und DH_SLOT_ID als Fremdschlüssel, der auf die Tabelle CH22_DH_SLOT verweist. Erstellen Sie einen neuen Bericht und ein neues Formular für Schüler und autorisieren Sie die Seite für Schüler. Je nach den institutionellen Regeln können Sie Lehrern das Recht gewähren, die Notizen der Schüler einzusehen oder zu ändern.
3. Um die Abfrageergebnisse auf Zeilen zu beschränken, deren Datumsspalten gleich oder größer als die aktuelle Zeit sind, muss die Bedingung WHERE date_column >= SYSDATE hinzugefügt werden.



# 23. Telco-Fall

VERONIKA ŠALGOVÁ, JOZEF KOSTOLNÝ, MICHAL MRENA, MICHAL KVET, MIROSLAV POTOČÁR

## 23.1 Wirtschaftliche Sicht des Falles

Um den schrittweisen Prozess der Entwicklung eines Anwendungsprototyps anschaulich zu machen, betrachten wir das hypothetische Unternehmen VEYOMI als Beispiel. VEYOMI ist ein kleines Unternehmen, das sich auf den Verkauf von Telekommunikationsdiensten an seine Kunden konzentriert. Da VEYOMI sein Produktportfolio erweitern und einen wachsenden Kundenstamm bedienen möchte, ist es zwingend erforderlich, eine robuste Anwendung einzurichten und zu pflegen, die eine nahtlose Verwaltung verschiedener Aspekte wie Produktbestand, Kundeninformationen und automatische Abrechnungssysteme ermöglicht. In einer zunehmend wettbewerbsintensiven Marktlandschaft erkennt VEYOMI die überragende Bedeutung eines schnellen und bequemen Zugangs zu wichtigen Informationen für seine geschätzten Kunden. Durch die Entwicklung einer intuitiven Anwendung kann VEYOMI seine Reaktionsfähigkeit verbessern und sicherstellen, dass relevante Details zu Produkten, Dienstleistungen, Preisen und Abrechnungen für seine anspruchsvollen Kunden jederzeit verfügbar sind.

## 23.2 Problemstellung

In der Geschäftswelt ist es äußerst wichtig, Aufzeichnungen über Ihre Kunden, die von ihnen gekauften Produkte oder Dienstleistungen und die von ihnen in Anspruch genommenen laufenden Dienstleistungen zu führen. So haben Sie alle notwendigen Informationen zur Hand und können Ihre Kunden besser betreuen. Darüber hinaus ist es wichtig, dass Sie Ihren Kunden eine einfache und zugängliche Möglichkeit bieten, den Status ihrer Dienste zu überprüfen. Durch das Angebot einer benutzerfreundlichen Plattform oder Anwendung können Ihre Kunden leicht herausfinden, was mit ihren Diensten geschieht und auf dem Laufenden bleiben. Indem Sie genaue Aufzeichnungen führen und einen einfachen Zugang zu Service-Updates bieten, können Sie die Kundenzufriedenheit erhöhen, Vertrauen aufbauen und Ihre Geschäftsabläufe insgesamt verbessern.

## 23.3 Anwendungsfälle

Bei der Erstellung einer Anwendung ist es notwendig, mit der Spezifikation der Dienste zu beginnen, die die Anwendung bereitstellen soll und was sie abdecken soll. Daher ist die Erstellung einer

Liste der einzelnen Funktionalitäten und Dienste von entscheidender Bedeutung. Zunächst ist es jedoch notwendig, sich auf die Definition der Nutzertypen zu konzentrieren. Für unsere Anwendung können wir drei grundlegende Rollen in Betracht ziehen - Kunde, Manager und Administrator. Jede Rolle verfügt über spezifische Dienste, um die Gesamtabdeckung der Funktionalität der Anwendung zu gewährleisten.

### 23.3.1 Narrative Beschreibung

Der Kürze halber werden wir nur zwei von mehreren Anwendungsfällen vorstellen:

- Dienst hinzufügen - Kunden können ganz einfach beantragen, dass neue Dienste zu ihrem bestehenden Abonnement hinzugefügt werden. Das System aktiviert den neuen Dienst und aktualisiert die Abrechnung des Kunden entsprechend. Der Kunde erhält eine Bestätigung und Informationen über den hinzugefügten Dienst.
- Status der Dienste anzeigen - Es ist möglich, den aktuellen Status und die Details der vom Kunden abonnierten Dienste einfach einzusehen. Das System ruft eine Übersicht über die abonnierten Dienste des Kunden ab und zeigt sie an. Dazu gehören Informationen wie die Art des Dienstes, der Aktivierungsstatus, die verbleibende Nutzung oder Gültigkeitsdauer und alle damit verbundenen Funktionen oder Einschränkungen. Innerhalb der Dienstübersicht können die Kunden auf detailliertere Informationen über ihre Dienstnutzung zugreifen. Dazu können Datenverbrauch, Gesprächsminuten, gesendete Textnachrichten oder andere relevante Nutzungskennzahlen gehören.

### 23.3.2 Semi-strukturierte Beschreibung

Zwei Anwendungsfälle sind in den Tabellen 23.1 und 23.2 dargestellt.

### 23.3.3 Anwendungsfalldiagramm

Der Kunde muss den Kauf von Pauschal- und Zusatzleistungen sicherstellen und den Status der Leistungen anzeigen können, die er auch stornieren können sollte. Eine weitere wichtige Funktion ist die Anzeige einer Übersicht über die Rechnungen.

Die Benutzerrolle Manager kann auf die Übersicht der Kunden und deren Hinzufügen, Bearbeiten und Löschen zugreifen. Darüber hinaus kann der Manager auch Rechnungen erstellen.

Die letzte Benutzerrolle ist die des Administrators, die eine grundlegende Funktion hat: die Verwaltung von Managern. Die Darstellung dieser Rollen ist in Abbildung 23.1 zu sehen.

Tabelle 23.1: Beschreibung des Anwendungsfalls: Dienst hinzufügen.

| **Keyword** | **Wert** |
|---|---|
| ID: | *Ch23-01* |
| Wert: | *Dienst hinzufügen* |
| Beschreibung: | *Der Kunde verwendet die APEX-Anwendung, um einen neuen Dienst zu den verfügbaren Diensten hinzuzufügen.* |
| Primärer Akteur: | *Kunde* |
| Voraussetzungen: | *Zugang zur Anwendung, gültige Anmeldedaten, Webanwendung ist zugänglich.* |
| Nachbedingungen: | *Die Daten des neuen Dienstes werden in der Datenbank gespeichert.* |
| *Haupt:* | *Szenarien* |
| Erfolgs-Szenario: | 1. *Kunden-Login in die Anwendung.*<br>2. *Der Kunde navigiert auf der Startseite zum Bereich Extra kaufen.*<br>3. *Kunde gibt Menge ein.*<br>4. *Mit dem Dropdown-Menü wählt der Kunde eine neue Dienstleistung (Kaufart) aus.*<br>5. *Durch Anklicken der Schaltfläche „Buy“ bestätigt der Kunde die Aktivierung des neuen Dienstes.* |
| Erweiterungen: | - |
| Häufigkeit der Nutzung: | *Der Kunde wechselt eine Reihe von Diensten etwa einmal pro sechs Monate* |
| Status: | *Fertiggestellt* |
| Eigentümer: | *Kunde* |
| Priorität: | *hoch* |

Tabelle 23.2: Beschreibung des Anwendungsfalls: Status der Dienste anzeigen.

| **Keyword** | **Wert** |
|---|---|
| ID: | *Ch23-02* |
| Titel: | *Status der Dienste anzeigen* |
| Beschreibung: | *Kunde nutzt APEX-Anwendung, um Einblick in die Nutzung der Dienste zu erhalten.* |
| Primärer Akteur: | *Kunde* |
| Voraussetzungen: | *Zugang zur Anwendung, gültige Anmeldedaten, Webanwendung ist zugänglich.* |
| Nachbedingungen: | - |
| *Haupt:* | *Szenarien* |
| Erfolgs-Szenario: | 1. *Kunden-Login in die Anwendung.*<br>2. *Der Kunde wählt die Registerkarte „Stats“.*<br>3. *Mit dem Dropdown-Menü wählt der Kunde einen Dienst aus. Es wird ein Balkendiagramm für diesen speziellen Dienst angezeigt.* |
| Erweiterungen: | - |
| Häufigkeit der Nutzung: | *Kunden verlangen etwa einmal pro Monat einen Einblick* |
| Status: | *Fertiggestellt* |
| Eigentümer: | *Kunde* |
| Priorität: | *hoch* |

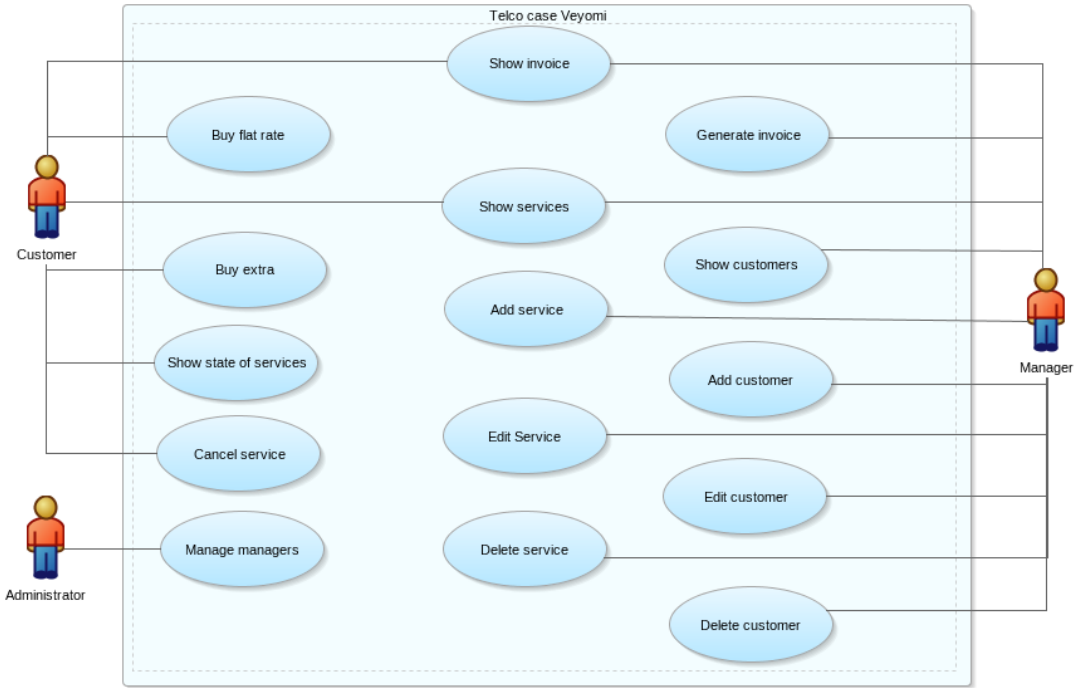



Abbildung 23.1: Anwendungsfalldiagramm.

## 23.4 Datenmodell

### 23.4.1 Narrative Beschreibung des Datenmodells

Die Tabelle **CH23_Person** speichert die primären Personendaten und die Art der Benutzerrolle. Wenn die Person ein Kunde ist, wird ein Datensatz in der Tabelle **CH23_Customer** angelegt, in der neben den Schlüsseln auch die Adresse und die Telefonnummer gespeichert werden. Die Tabelle **CH23_FlatRate** speichert Informationen über die Flatrate, nämlich den Namen und den Preis. In der Tabelle **CH23_Service** werden Daten zu einzelnen Leistungen, deren Bezeichnung, Preis und angebotene Einheiten gespeichert. Einzelne Leistungen, die Pauschalpreisen zugeordnet sind, werden in der Tabelle **CH23_FlatRateService** gespeichert, in der wir neben den Schlüsseln auch die zugeordnete Menge kennen. Bezieht der Kunde eine Flatrate, werden Informationen darüber in der Tabelle **CH23_CustomerFlatRate** gespeichert, einschließlich des Anfangs- und Enddatums. Die verbrauchten Einheiten der Flatrate werden in der Tabelle **CH23_UsageLog** registriert, wo wir wissen, welcher Kunde welche Leistung innerhalb welcher Flatrate verbraucht hat, zusammen mit dem Datum und der Menge. Zusätzlich zur Pauschale kann der Kunde zusätzliche Leistungen erwerben, die und deren Menge in der Tabelle **CH23_ExtraService** erfasst werden.

### 23.4.2 Logisches Datenmodell

Das logische Datenmodell ist in Abbildung 23.2 dargestellt.

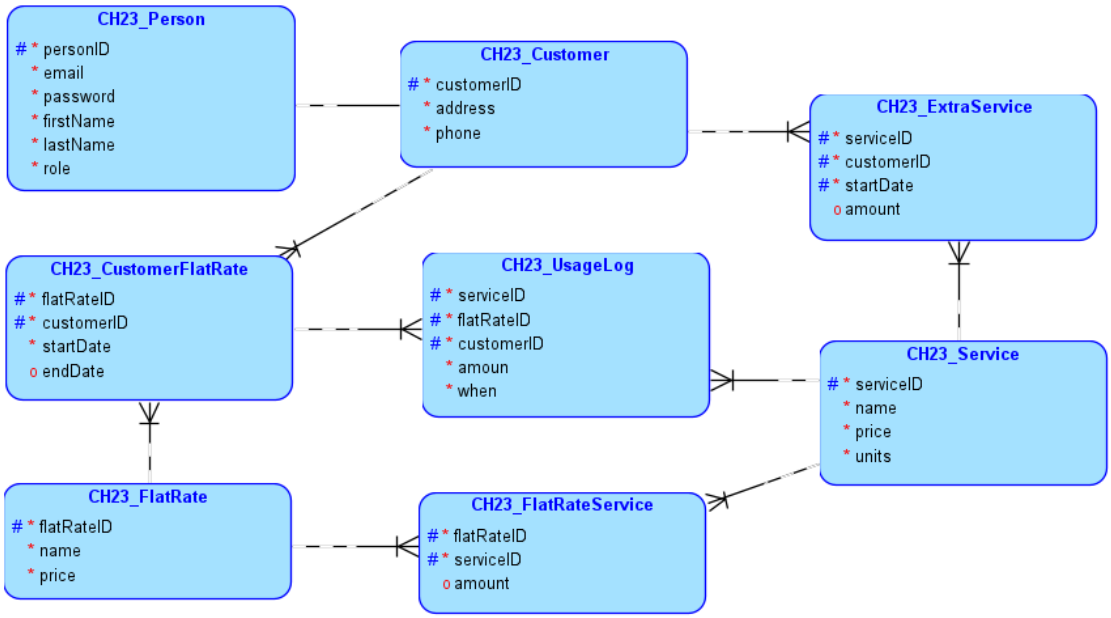



Abbildung 23.2: Logical data model.

### 23.4.3 Relationales Datenmodell

Das relationale Datenmodell ist in Abbildung 23.3 dargestellt.

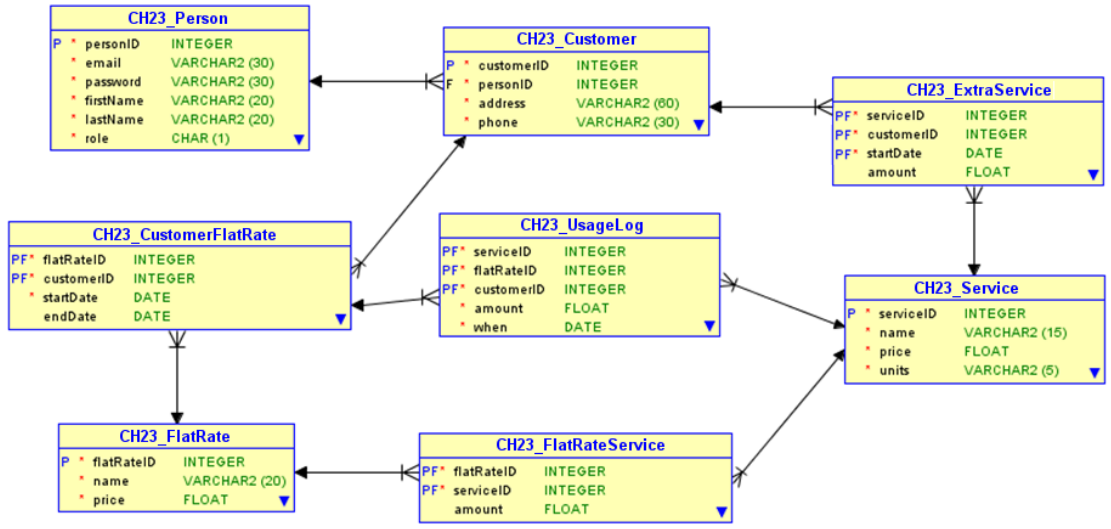



Abbildung 23.3: Relationales Datenmodell.

## 23.5 Benutzerauthentifizierung und Benutzerrollen

In der obigen Beschreibung haben wir drei Benutzerrollen identifiziert. Nach der Anmeldung stellt die Anwendung dem Benutzer je nach seiner Rolle Inhalte zur Verfügung. Der Anmeldevorgang selbst ist jedoch für alle Benutzer gleich. APEX bietet uns mehrere Möglichkeiten, Benutzer zu verwalten und zu authentifizieren. In der folgenden Liste stellen wir einige ausgewählte Methoden vor:

- Datenbank-Konto,
- LDAP-Server,

- Oracle APEX-Konto,
- Anmeldung bei einer sozialen Plattform,
- Benutzerdefiniert.

Im Oracle Apex-Tool wählen wir die Authentifizierungsmethode - **Authentication Scheme** - auf der Unterseite **Shared Components \ Authentication Schemes** aus. Auf dieser Unterseite finden Sie eine Liste der vorhandenen Anmeldeschemata, und Sie können hier auch ein neues Anmeldeschema erstellen. Bei der Erstellung eines neuen Schemas können wir eine der oben erwähnten Authentifizierungsmethoden wählen. Für die Bedürfnisse unserer Anwendung haben wir die allgemeinste Methode gewählt, **Benutzerdefiniert**.

Der Hauptteil der **Benutzerdefiniert**-Authentifizierung ist die sogenannte Authentifizierungsfunktion. Die Eingabe für diese Funktion ist der Benutzername und das Passwort, die der Benutzer auf der Anmeldeseite eingibt, und die Ausgabe ist ein **boolean**-Wert. Wie erwartet, gibt die Funktion **true** zurück, wenn der angegebene Name und das Kennwort mit einem bestehenden Benutzer übereinstimmen, und andernfalls **false**. In der Auflistung 23.1 ist eine Kopfzeile einer solchen Funktion zu sehen. Daher müssen wir beim Erstellen eines **Benutzerdefiniert**-Anmeldeschemas den Namen einer solchen Funktion angeben. Das heißt, eine solche Funktion muss bereits existieren.

```
CREATE OR REPLACE FUNCTION authenticate_user
(p_username IN varchar2,
p_password IN varchar2)
RETURN boolean;
```

Listing 23.1: Kopfzeile einer Authentifizierungsfunktion.

Wir implementieren den Hauptteil der Authentifizierungsfunktion mit Hilfe der PL/SQL-Sprache, d.h. es liegt an uns, wie wir den Benutzer authentifizieren. Das ist der Grund, warum die Methode **Benutzerdefiniert** die allgemeinste ist. Wir können z.B. einen externen Authentifizierungsdienst über die REST-API kontaktieren, in einer lokalen Datenbank nach einem Benutzer suchen oder die Authentifizierung einem anderen lokalen Prozess überlassen. In unserer Anwendung haben wir uns für das übliche Verfahren entschieden, den Benutzer in der lokalen Datenbank zu suchen und zu verifizieren.

Wir sehen die Tabelle CH23_Person in dem oben vorgestellten Datenmodell. Die Spalten **email** und **password** sind für die Authentifizierung relevant. Die **email** des Benutzers dient auch als eindeutiger Anmeldename. Die Spalte **password** enthält Benutzerpasswörter, die in einer sicheren Hash-Form gespeichert sind. Die Beschreibung der sicheren Passwortspeicherung würde den Rahmen dieses Kapitels sprengen. Der Leser kann in der Literatur mehr darüber erfahren. Vor der Eingabe von Daten oder der Bearbeitung der Spalte **password** in der Tabelle **CH23_Person** ist es jedoch erforderlich, das Kennwort zu hashen. Zu diesem Zweck haben wir eine Hilfsfunktion **hash_password** erstellt, deren Kopfzeile wir in der Auflistung 23.2 sehen können. Diese Funktion nimmt den Benutzernamen und das Kennwort im Klartext und generiert ein Kennwort in einem gesicherten gehashten Format, das in der Datenbank gespeichert werden kann.

```
CREATE OR REPLACE FUNCTION hash_password
(p_username IN varchar2,
p_password IN varchar2)
RETURN varchar2;
```

Listing 23.2: Die Kopfzeile der Hilfsfunktion hash_password.

Wir haben das automatische Hashing von Passwörtern bei der Arbeit mit der Tabelle **CH23_Person** durch die Erstellung von Triggern für die Tabelle CH23_Person gelöst. Die Trigger hashen das Klartext-Passwort mithilfe der oben beschriebenen Funktion **hash_password**, bevor sie die Spalte **password** einfügen oder aktualisieren. Damit wird sichergestellt, dass alle Kennwörter in der Tabelle **CH23_Person** sicher gespeichert werden - nicht einmal der Datenbankadministrator hat

Zugriff auf das tatsächliche Kennwort. In Tabelle 23.3 sehen wir ein Beispiel für die in der Tabelle **CH23_Person** gespeicherten Daten. Interessanterweise haben zwei in der Tabelle aufgeführte Benutzer das gleiche Kennwort. Diese Tatsache lässt sich jedoch nicht anhand der sicher gespeicherten Kennwörter feststellen.

| PERSONID | EMAIL | PASSWORD | ROLE |
|---|---|---|---|
| 5 | DUIS.A@OUTLOOK.CA | $2a$12$EfSjb2zSiKjqWCoZ47mQeOY6IPoHP/6LvECZxGEWjFZIZcXTF3UgGasd | 'a' |
| 11 | LOBORTIS@OUTLOOK.EDU | $2a$12$IUqALTj3reLAoTdtC5SNde2.C5A.DOm..7pFw9UBnIsaf7gSNcZyapql | 'c' |
| 14 | AC.FERMENTUM@AOL.NET | $2a$12$GuaO7Tjp.65d7NYLiEjSzeO6VWQfDz8eXja7dWubcjy77aZDAKRTutyq | 'm' |

Tabelle 23.3: In der Tabelle CH23_Person gespeicherte Beispieldaten.

Die Implementierung der Authentifizierungsfunktion ist nun einfach. In dieser Funktion suchen wir zunächst in der Tabelle **CH23_Person** nach dem Passwort des Benutzers mit dem angegebenen Benutzernamen. Anschließend wird das vom Benutzer eingegebene Kennwort mit der Funktion **hash_password** gehasht. Wenn beide Hashes übereinstimmen, ist die Benutzerauthentifizierung erfolgreich, und die Funktion gibt **true** zurück. Stimmen die Hashes hingegen nicht überein oder gibt es keinen Benutzer mit dem angegebenen Namen in der Tabelle CH23_Person, wird false zurückgegeben.

Der Nachteil der von uns gewählten **Benutzerdefiniert**-Methode ist die größere anfängliche Schwierigkeit bei der Erstellung einer Datenbank, innerhalb derer es notwendig ist, Hilfsfunktionen und Trigger zu entwickeln. Auf der anderen Seite liegt der Vorteil darin, dass wir, die Anwendungsentwickler, den gesamten Authentifizierungsprozess kontrollieren. Ein weiterer Vorteil ist, dass die verwendete Implementierung leicht skalierbar ist. Wenn wir zum Beispiel in Zukunft einen externen Dienst für die Authentifizierung nutzen möchten, müssen wir die Implementierung der Authentifizierungsfunktion so ändern, dass sie den externen Dienst statt der lokalen Datenbank kontaktiert.

Die zweite wichtige Aufgabe im Zusammenhang mit der Benutzerverwaltung ist die Verwaltung der Benutzerrollen. Während der Anmeldevorgang für alle Benutzer gleich war, mussten wir bei der Rollenverwaltung den Zugriff auf verschiedene Teile der Anwendung von der Rolle des Benutzers abhängig machen. Die Rollenverwaltung ist eng mit der Benutzerverwaltung verbunden. Zunächst müssen alle möglichen Benutzerrollen in der Oracle APEX-Umgebung festgelegt werden. Die Rollenverwaltung finden wir auf der Unterseite **Shared Components \ Application Access Control**. Auf dieser Unterseite sehen wir eine Liste der vorhandenen Rollen, und wir können hier auch eine neue Rolle erstellen. Für jede neue Rolle muss ein statischer Bezeichner angegeben werden, den wir später bei der Zuweisung der Rolle an einzelne Benutzer verwenden werden. In unserer Anwendung verwenden wir Rollen mit den folgenden statischen Bezeichnern:

- ADMIN,
- CUSTOMER,
- MANAGER.

Wir weisen einem Benutzer mit PL/SQL-Code eine Rolle zu. In der Auflistung 23.3 sehen wir einen Beispielcode für die Zuweisung der Rolle **CUSTOMER** (statische Rollenkennung) an den Benutzer **LOBORTIS@OUTLOOK.EDU**. Wir haben einen weiteren Satz von Triggern für die Tabelle Person verwendet, um diesen Prozess zu automatisieren. Die Spalte **Role** in dieser Tabelle kann Werte aus dem Set {'a', 'm', 'c'} annehmen, die mit den Anfangsbuchstaben der

oben definierten Rollen übereinstimmen. Im Hauptteil des Triggers wird vor dem Einfügen oder Aktualisieren von Daten die entsprechende Rolle für den gegebenen Benutzer entsprechend dem Wert der Spalte Role festgelegt, wobei Code ähnlich dem Code in der Auflistung 23.3 verwendet wird.

```
APEX_ACL.REPLACE_USER_ROLES (
 p_application_id    => 151905,
p_user_name          => 'LOBORTIS@OUTLOOK.EDU',
 p_role_static_ids   => wwv_flow_t_varchar2('CUSTOMER') );
RETURN boolean;
```

Listing 23.3: PL/SQL-Code, der einen bestimmten Benutzer mit einer bestimmten Rolle verknüpft.

Der letzte Schritt bei der Implementierung von Benutzerrollen in der Anwendung besteht darin, verschiedene Unterseiten auf der Grundlage der Rolle des angemeldeten Benutzers anzuzeigen. Für jede Unterseite, die nur für eine bestimmte Benutzerrolle bestimmt ist, muss die Benutzerrolle, für die die jeweilige Unterseite bestimmt ist, in den Einstellungen des **Seitendesigners** im Abschnitt **Security \ Authorization Scheme** ein. Dieser Schritt stellt sicher, dass Benutzer mit anderen Rollen nicht auf diese Unterseite zugreifen können. Die Startseite unserer Anwendung ist beispielsweise insofern spezifisch, als sie keine Anmeldung erfordert, um sie anzuzeigen. Wir legen dieses Verhalten auch im Abschnitt **Security** mit dem Element **Authentication** fest, wo wir die Option **Page Is Public** auswählen.

Mit den oben beschriebenen Einstellungen wird sichergestellt, dass der Benutzer nur auf die Unterseiten seiner Rolle Zugriff hat. Allerdings sehen alle Benutzer weiterhin alle Unterseiten im Navigationsmenü - auch die, auf die sie keinen Zugriff haben. Ein solches Verhalten ist sicherlich nicht wünschenswert. Damit die Anwendung den Benutzern nur die Unterseiten anzeigt, auf die sie Zugriff haben, ist es notwendig, die Einstellungen für das Navigationsmenü anzupassen. Dies kann in der Unterseite **Shared Components \ Navigation and Search \ Navigation Menu** einstellen. Hier muss in der Spalte **Authorization Scheme** für jeden Eintrag im Menü die Benutzerrolle festgelegt werden, für die der jeweilige Eintrag angezeigt wird.

## 23.6 Anwendungsschnittstellen

Die Anwendung ist so konzipiert, dass einzelne Unterseiten erstellt werden, auf denen die vorgeschlagenen Dienstleistungen angeboten werden. Die primäre Seite ist die Startseite, die einen Überblick über die angebotenen Dienstleistungen und das Portfolio der Pauschaldienstleistungen bietet. Auf dieser Seite ist es möglich, das Unternehmen und die angebotenen Dienstleistungen vorzustellen, die der Kunde erwerben kann. Diese Seite ist ohne Anmeldung für alle Besucher zugänglich. Die Hauptseite bietet eine weitere wichtige Funktion: den Zugang zur Verwaltung nach dem Einloggen. Unter Verwaltung verstehen wir eine Reihe von Unterseiten, die je nach Art des Kontos, mit dem man sich anmeldet, unterteilt sind - Kunde, Manager oder Administrator. Für jede dieser Rollen müssen also eigene Unterseiten erstellt werden, die die vorgeschlagenen Funktionen der Dienste der jeweiligen Rolle abdecken.

Die Kundenseite gibt in einzelnen Feldern einen Überblick über den aktuellen Stand der bestellten Leistungen und Zusatzleistungen. Ein weiterer extra Reiter ermöglicht den Kauf bestimmter Leistungen durch Auswahl der Leistungsart aus dem Kombinationsfeld. Eine weitere Funktionalität bietet die Anzeige und Erstellung von Rechnungen, die mit einem Pop-up-Fenster gelöst werden kann, in dem nach dem Öffnen die Rechnung in Form eines Berichts angezeigt wird, der die Konfiguration des Speicherns und Druckens im PDF-Format ermöglicht. Der Bericht für die Rechnung wird von einem bestimmten Skript aufgerufen, das die Daten für die im jeweiligen Monat erbrachte Leistung zusammenfasst. Ein weiterer wichtiger Teil der Kundenseite ist die Übersicht über die Leistungsnutzung mit Details wie Menge, Art der Leistung und Zeit. Diese Tabelle ermöglicht es, diese Übersicht mit Hilfe der Filterspezifikation und der Sortierung nach einzelnen Spalten zu

filtern und zu sortieren. Eine solche Funktionalität kann für den Kunden von Vorteil sein, wenn er z.B. einen Überblick über die Anzahl der von ihm in einem bestimmten Zeitraum versendeten SMS erhalten möchte. Schließlich hat der Kunde die Möglichkeit, sich die Statistiken in einem bestimmten Ordner grafisch darstellen zu lassen. Diese Funktionalität wird auf einer separaten Seite gehandhabt, die über eine Registerkarte mit der Seite des Kunden verbunden ist, und es ist möglich, zwischen den einzelnen Funktionalitäten zu wechseln. Die Statistikseite enthält eine Demonstration der grafischen Darstellung der Logos. Eine solche Darstellung ist geeignet, um Veränderungen im Zeitablauf zu zeigen. So lassen sich beispielsweise die abgerufenen Minuten in den einzelnen Monaten, die Anzahl der versendeten SMS oder die Menge der übertragenen Daten im bestellten Programm leicht darstellen.

### 23.6.1 Anwendungsdesign

In diesem Kapitel werden Screenshots der Anwendungsformulare für verschiedene Benutzerrollen gezeigt.

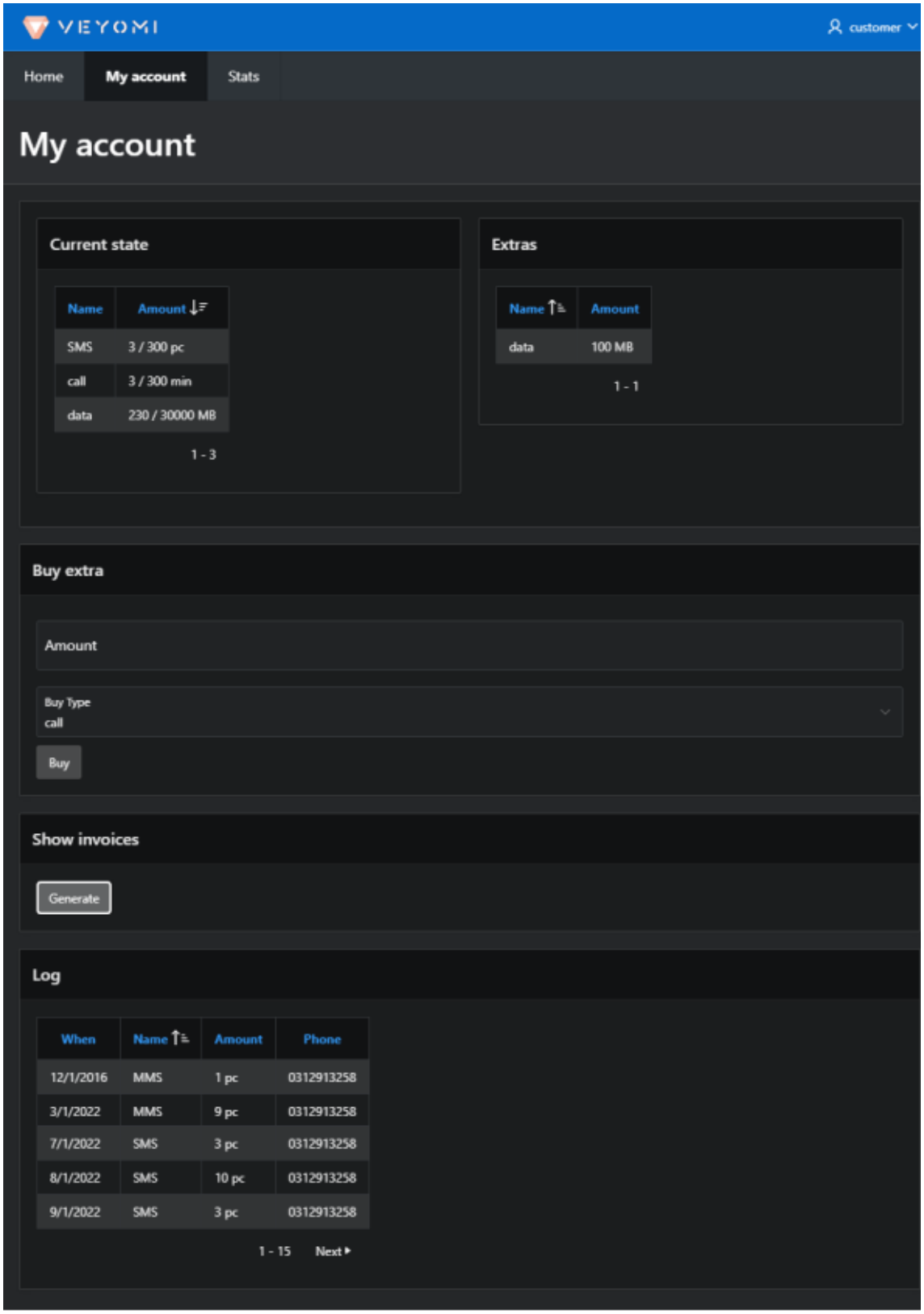



Abbildung 23.4: Kunden-Dashboard.

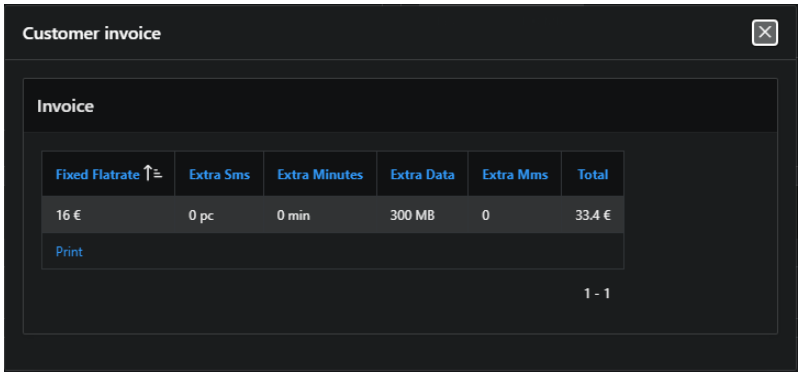



Abbildung 23.5: Kunden-Dashboard - Modales Fenster für Rechnungen.

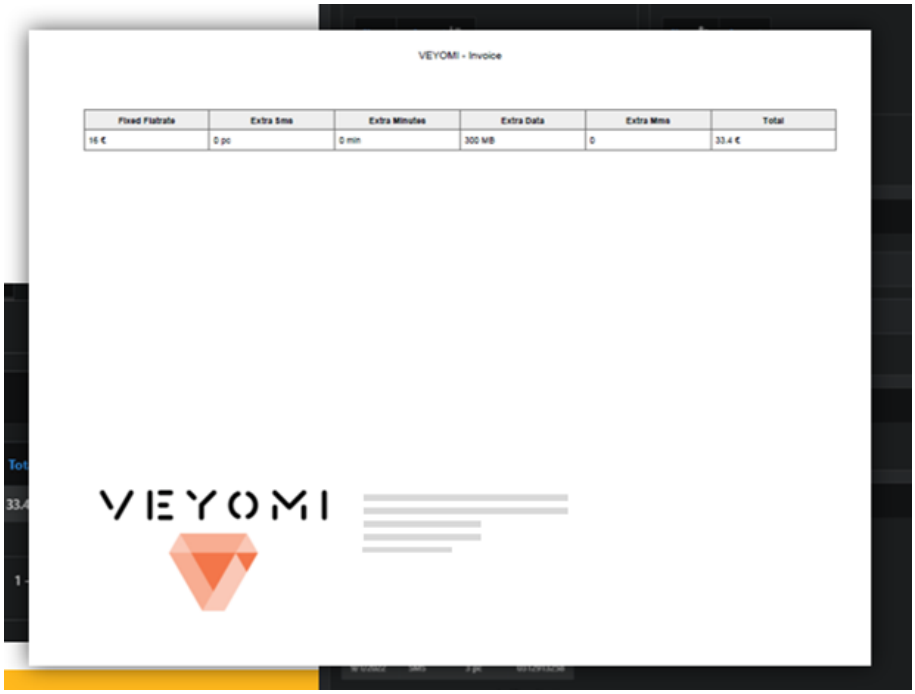



Abbildung 23.6: Kunden-Dashboard - Rechnung im PDF-Format.

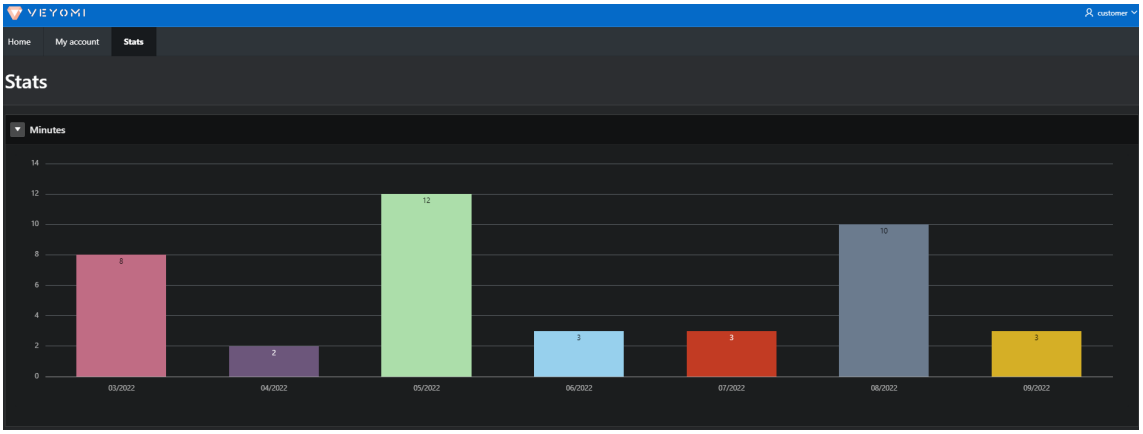



Abbildung 23.7: Kunden-Dashboard - Statistik der Minuten.




Abbildung 23.8: Kunden-Dashboard - Statistik der SMS.

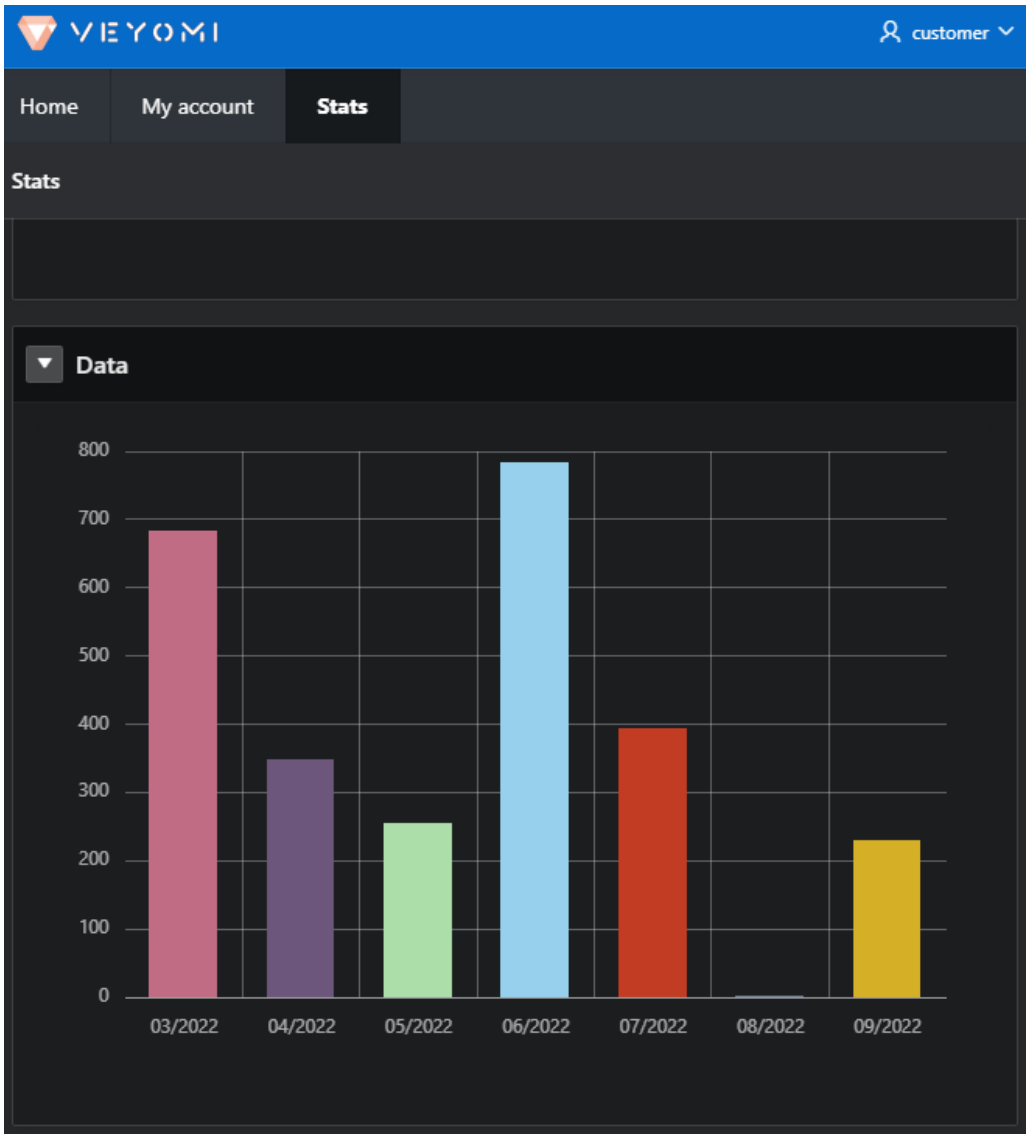



Abbildung 23.9: Kunden-Dashboard - Statistik der Daten.

Nachdem sich der Manager bei der Anwendung angemeldet hat, werden die Funktionen zur Kundenverwaltung auf separaten Registerkarten bereitgestellt, die Teil der Managerseite sind. Die

erste Funktion ist das Hinzufügen eines neuen Kunden, wo der Manager ein neues Kundenkonto hinzufügen und erstellen kann. Als Nächstes verwenden wir die Komponente für die Kundenverwaltung, um Kundeninformationen zu bearbeiten, die in einem Datengitter dargestellt sind. Diese Komponente bietet auch die Möglichkeit, einfache Berichte zu erstellen oder Kundeninformationen in andere Anwendungen von Drittanbietern zu exportieren, falls erforderlich. Die Registerkarte Kundenanzeige schließlich zeigt eine vereinfachte Form der Darstellung einer Übersicht über die Kundeninformationen. Dieses Raster ist so konfiguriert, dass es die Bearbeitung einzelner Positionen nicht zulässt, aber es bietet die Möglichkeit, Berichte und Exporte aus dieser Liste zu erstellen.

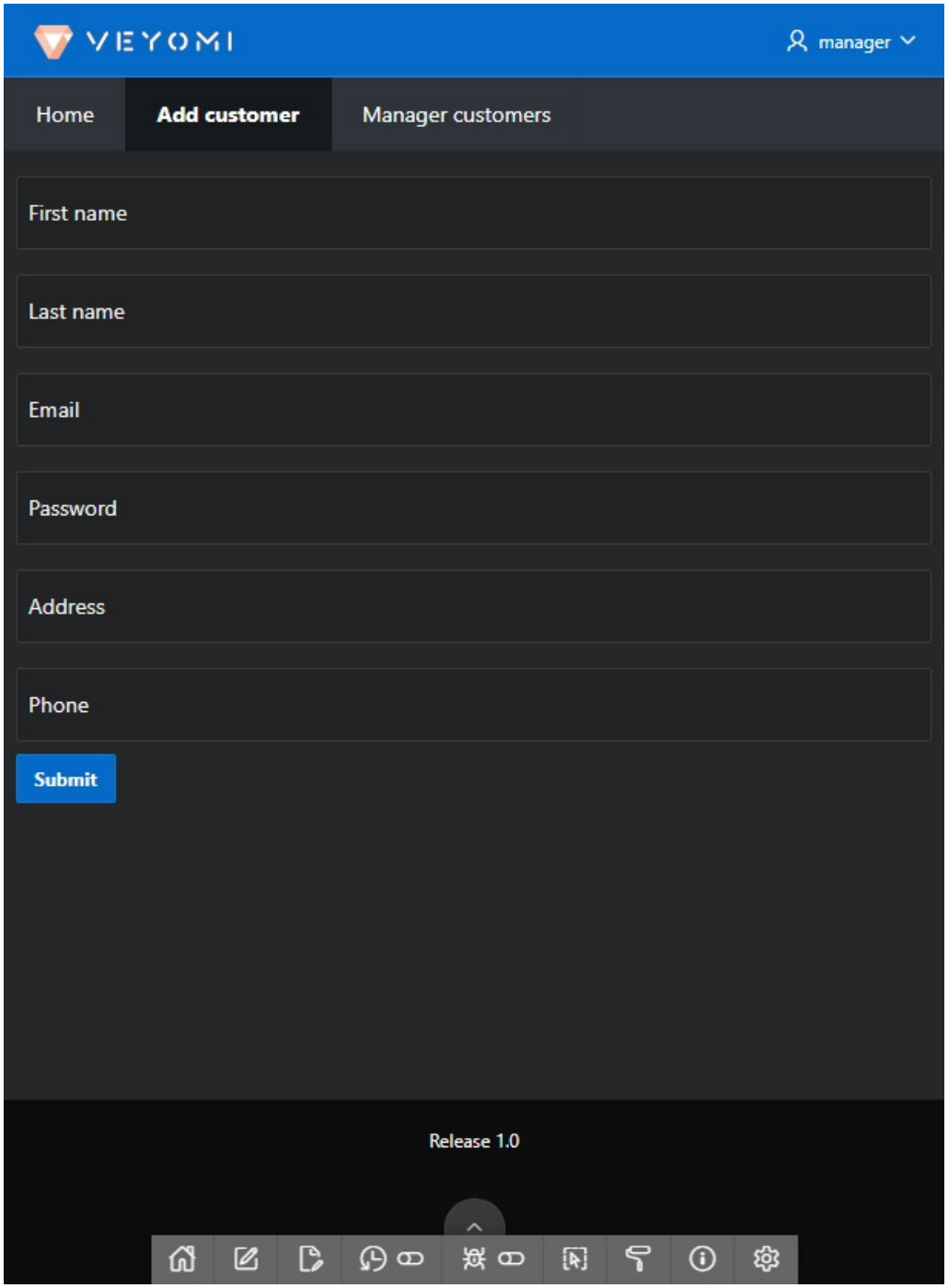



Abbildung 23.10: Manager-Dashboard - Kunde hinzufügen.

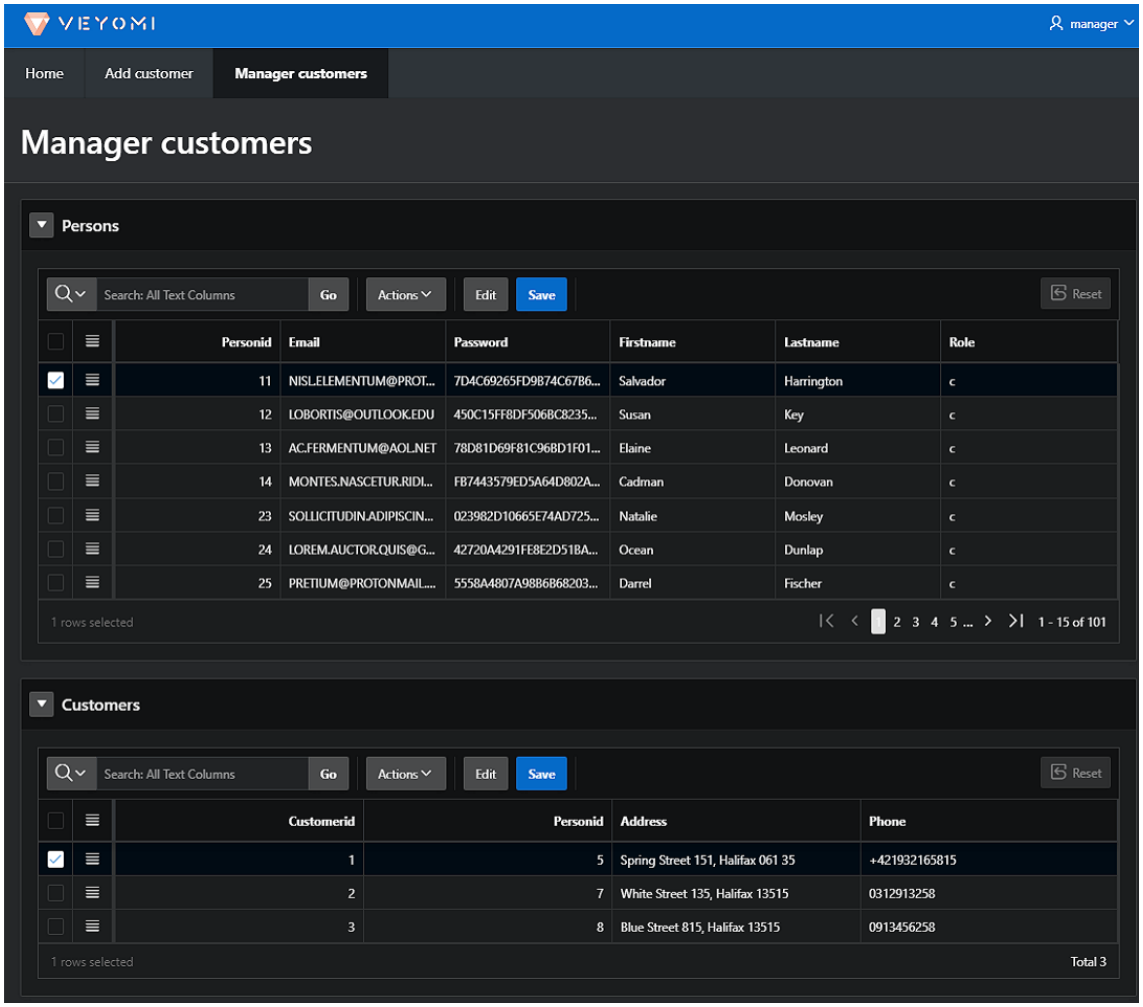



Abbildung 23.11: Manager-Dashboard - Verwalten von Kunden.

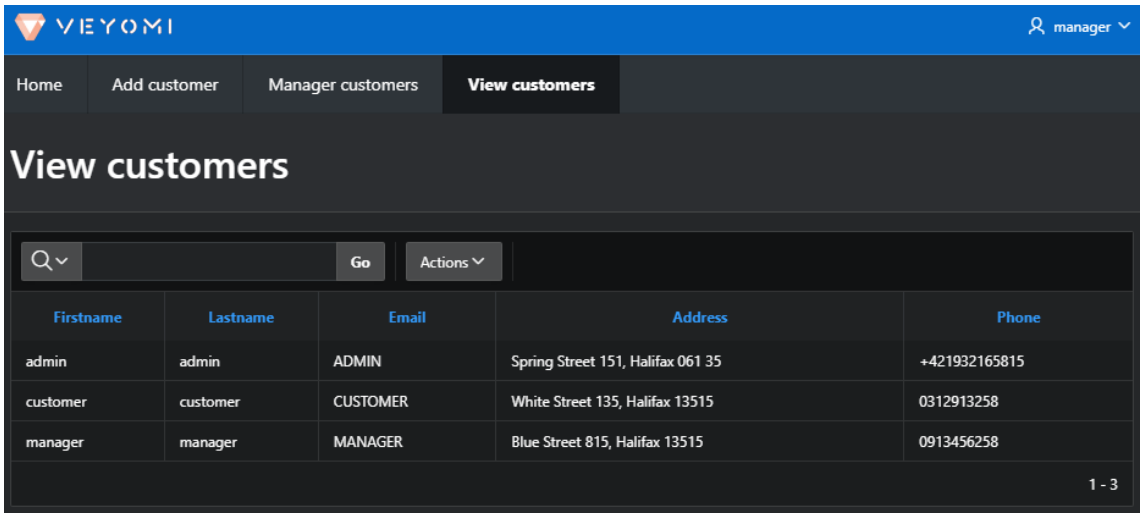



Abbildung 23.12: Manager-Dashboard - Kunden anzeigen.

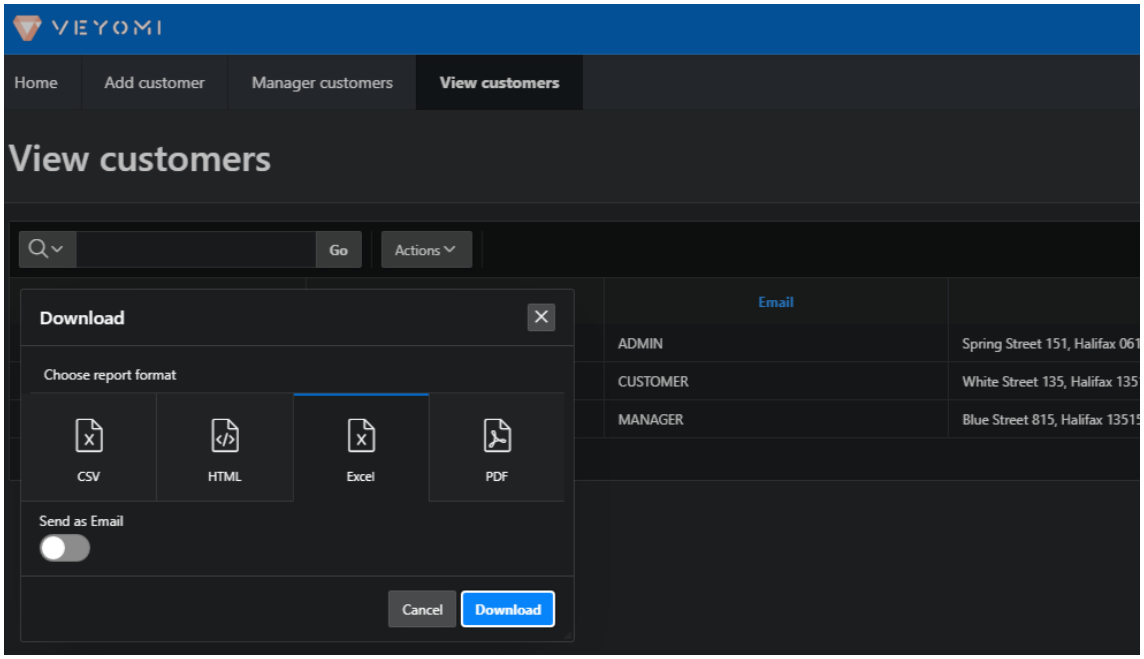



Abbildung 23.13: Manager-Dashboard - Export von Kunden.

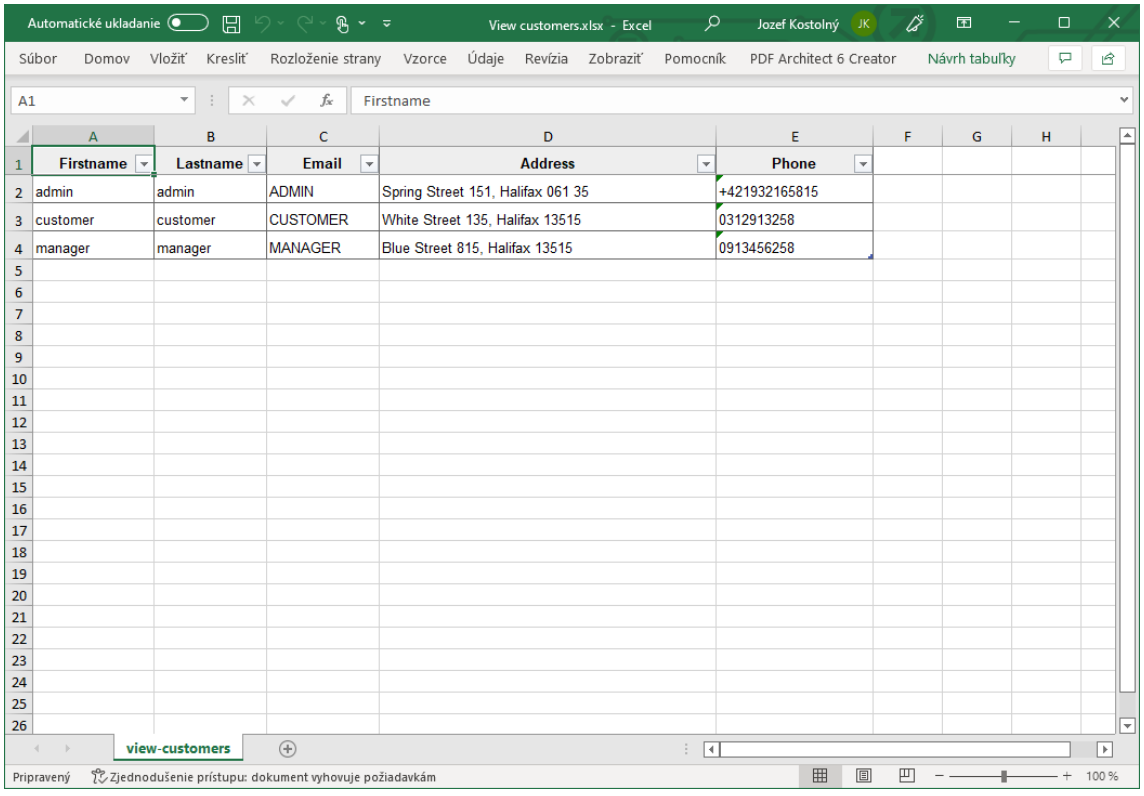



Abbildung 23.14: Manager-Dashboard - Kundenexport in XLS-Datei.

Die letzte Seite, die erstellt werden muss, ist die Seite für den Anwendungsadministrator. Auf dieser Seite bieten wir den Service, Manager in einem bearbeitbaren Datengitter zu verwalten, Exportberichte und Filterfunktionen zu ermöglichen.

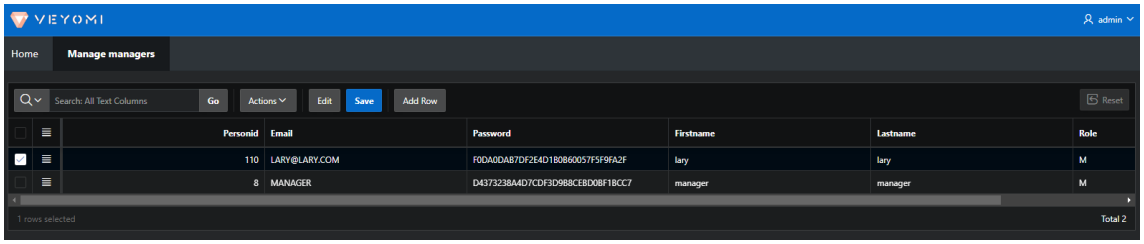



Abbildung 23.15: Administrator-Dashboard.

Ein wichtiger Teil ist die Erstellung des Anmeldeformulars. In unserer Anwendung erwägen wir, dieses Formular auf einer bestimmten Seite zu platzieren, die immer aufgerufen wird, wenn sichere Seiten aufgerufen werden, auf denen es notwendig ist, sich so schnell wie möglich zu authentifizieren.

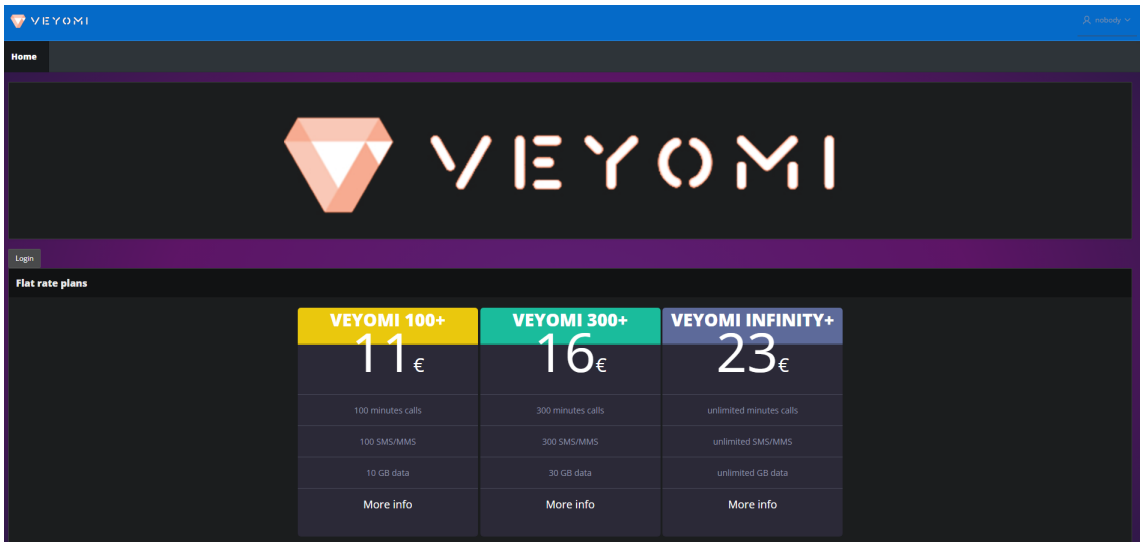



Abbildung 23.16: Eine Landing Page mit Login.

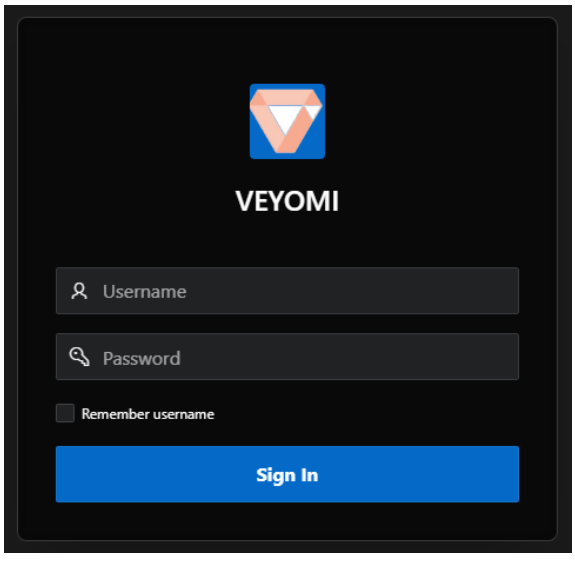



Abbildung 23.17: Login-Seite.

## 23.7 Skripte

```
CREATE SEQUENCE "CUSTOMER_SEQ"
MINVALUE 4 MAXVALUE 9999999999999999999999999999
INCREMENT BY 1
START WITH 4
NOCACHE NOORDER NOCYCLE NOKEEP NOSCALE GLOBAL
/

CREATE OR REPLACE EDITIONABLE TRIGGER "BI_CUSTOMER"
BEFORE INSERT ON CH23_CUSTOMER
FOR EACH ROW
BEGIN
SELECT CUSTOMER_SEQ.nextval INTO :new.customerid
FROM dual;
END;
/

ALTER TRIGGER "BI_CUSTOMER" ENABLE;
/
```

Listing 23.4: Trigger auf Kundensequenz.

```
CREATE SEQUENCE "PERSON_SEQ"
MINVALUE 1 MAXVALUE 9999999999999999999999999999
INCREMENT BY 1
START WITH 1
NOCACHE NOORDER NOCYCLE NOKEEP NOSCALE GLOBAL
/
CREATE OR REPLACE EDITIONABLE TRIGGER "BI_PERSON"
BEFORE INSERT ON CH23_PERSON
FOR EACH ROW
BEGIN
SELECT PERSON_SEQ.nextval INTO :new.personid
FROM dual;
:new.email := UPPER(:new.email);
:new.password := hash_password(UPPER(:new.email), :new.password);
END;
/

ALTER TRIGGER "BI_PERSON" ENABLE;
/

CREATE OR REPLACE EDITIONABLE TRIGGER "BU_PERSON"
BEFORE UPDATE ON CH23_PERSON
FOR EACH ROW
BEGIN
:new.email := UPPER(:new.email);
IF :new.password IS NOT NULL THEN
:new.password := hash_password(UPPER(:new.email), :new.password);
ELSE
```

```
:new.password := :old.password;
END IF;
END;
/
ALTER TRIGGER "BU_PERSON" ENABLE;
/
```

Listing 23.5: Trigger beim Einfügen einer Person.

```
CREATE OR REPLACE FUNCTION AUTHENTICATE_USER
(p_username in varchar2,
p_password in varchar2)
 return boolean
IS
l_user_name         CH23_PERSON.email%type := UPPER(p_username);
l_password          CH23_PERSON.password%type;
l_hashed_password varchar2(1000);
l_count             number;
BEGIN
SELECT COUNT(*) INTO l_count
FROM CH23_PERSON
WHERE email = l_user_name;

IF l_count > 0 THEN
l_hashed_password := hash_password(l_user_name, p_password);
SELECT password INTO l_password
FROM CH23_PERSON
WHERE email = l_user_name;

IF l_hashed_password = l_password THEN
APEX_UTIL.SET_AUTHENTICATION_RESULT(0);
 return true;
ELSE
APEX_UTIL.SET_AUTHENTICATION_RESULT(4);
 return false;
END IF;

ELSE
APEX_UTIL.SET_AUTHENTICATION_RESULT(1);
 return false;
END IF;
APEX_UTIL.SET_AUTHENTICATION_RESULT(7);
 return false;
EXCEPTION
WHEN OTHERS THEN
APEX_UTIL.SET_AUTHENTICATION_RESULT(7);
APEX_UTIL.SET_CUSTOM_AUTH_STATUS(sqlerrm);
 return false;
END AUTHENTICATE_USER;
/
```

Listing 23.6: Funktion AUTHENTICATE_USER.

## 23.8 Erstellen einer Homepage

Unter Create a Page können Sie zwischen verschiedenen Vorlagen, einer leeren Seite oder Funktionsseiten mit vordefinierten Funktionen wählen, z.B. Login, About page und Configuration Options. Zunächst müssen Sie grundlegende Informationen eingeben, z.B. eine Seitennummer und einen Namen, und ein Seitenmodell auswählen, z.B. Normal, Modal Dialog oder Drawer. Sie können auch zusätzliche Informationen eingeben, z.B. einen Navigationseintrag für die Seite festlegen (ob sie als oberster Eintrag oder in einem bestehenden Eintrag eingebettet sein soll) und ein Symbol auswählen, das in einem Dashboard-Menü angezeigt wird.

Nachdem Sie auf Create Page geklickt haben, wird das Dashboard zur Bearbeitung der Informationen und des Inhalts der Seite angezeigt. Der Hauptinhalt der Seite (Body) ist in Bereiche unterteilt.

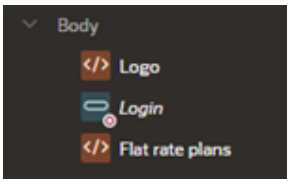



Abbildung 23.18: Bereiche des Bodies.

Jeder Bereich kann HTML oder verschiedene Inhalte enthalten, z.B. statisches HTML oder vordefinierte Inhalte.

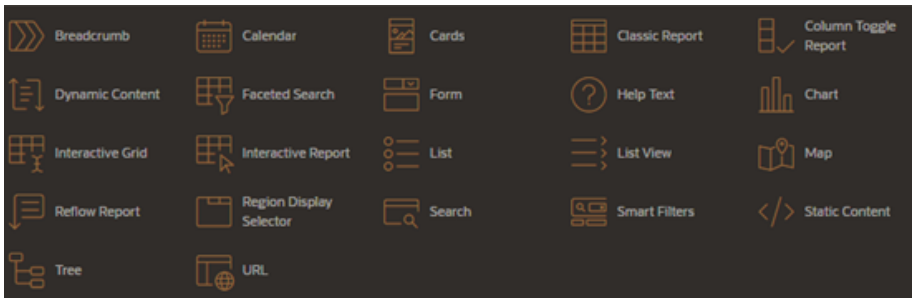



Abbildung 23.19: Liste der Inhalte.

Unser **Login** Bereich enthält eine Schaltfläche, die eine Seite zum Anmeldeformular des Benutzers weiterleitet. In der Eigenschaft Behaviour der Schaltfläche haben wir Action definiert, die in dieser Anwendung auf Redirect to Page gesetzt wurde. Dann haben wir Target auf eine vordefinierte Anmeldeseite gesetzt. Da es nicht notwendig ist, diese Schaltfläche anzuzeigen, wenn der Benutzer bereits angemeldet ist, müssen wir die Schaltfläche ausblenden, indem wir als Benutzer-Typ den öffentlichen Benutzer einstellen (der Benutzer hat sich nicht authentifiziert). Schließlich haben wir **Flat rate plans** als statischen Inhaltsbereich mit HTML-Code erstellt, für den wir CSS definieren können.




Abbildung 23.20: HTML-Code eines statischen Bereichs.

## 23.9 Erstellen einer Kundenseite

Wir haben diese Seite mit ähnlichen Schritten wie bei der Erstellung einer Homepage erstellt. Nach der Erstellung einer Seite fahren wir mit der Entwicklung eines Layouts für die Kundenseite fort, das in mehrere Bereiche unterteilt ist, z.B. für die Anzeige des aktuellen Stands der Dienstleistungen, den Kauf von Extras, die Anzeige von Rechnungen und die Anzeige von Protokollen.

Der Bereich **Current state** besteht aus zwei Unterbereichen, in denen numerische Daten angezeigt werden - der aktuelle Preis der verbrauchten Einheiten der Pauschale und der Extras. Die SQL-Abfrage wird verwendet, um diese Daten zu erhalten.

```
SELECT name, sum(u.amount) || ' ' || ' / ' || fs.amount || ' ' || s.units AS
    amount
FROM CH23_USAGELOG u
JOIN CH23_CUSTOMERFLATRATE cf ON (u.customerid = cf.customerid AND u.
    flatrateid = cf.flatrateid)
JOIN CH23_CUSTOMER c ON (cf.customerid = c.customerid)
JOIN CH23_PERSON p ON (c.personid = p.personid)
JOIN CH23_SERVICE s ON (u.serviceid = s.serviceid)
JOIN CH23_FLATRATESERVICE fs ON (fs.flatrateid = cf.flatrateid AND fs.serviceid = s
    .serviceid )
WHERE email = :APP_USER
AND EXTRACT(month from when) = EXTRACT(month from sysdate)
GROUP BY s.serviceid, s.name, s.units, fs.amount;
```

Listing 23.7: SQL-Abfrage - Verbrauchte Einheiten einer Pauschale.

```
SELECT name, SUM(amount) || ' ' || units AS amount
FROM CH23_EXTRASERVICE JOIN CH23_CUSTOMER USING(customerid)
JOIN CH23_PERSON USING (personid)
JOIN CH23_SERVICE USING (serviceid)
WHERE email = :APP_USER
AND EXTRACT(month from startDate) = EXTRACT(month from sysdate)
GROUP BY
 serviceid ,
name,
 units ;
```

Listing 23.8: SQL Abfrage - Verbrauchte Einheiten zusaetzlicher Services.

Der Bereich **Buy extra** enthält ein Zahlenfeld, eine Auswahlliste und eine Schaltfläche. Es ist möglich, einen Betrag festzulegen und auszuwählen, welche Zusatzleistung wir kaufen möchten. Die Bedienung erfolgt über eine Schaltfläche, die einen Prozess mit einem PL/SQL-Code ausführt:

```
INSERT INTO CH23_EXTRASERVICE VALUES (
:BUY_TYPE,
SELECT customerid
FROM CH23_CUSTOMER JOIN CH23_PERSON USING (personid)
WHERE email = :APP_USER),
sysdate ,
:BUY_AMOUNT
);
```

Listing 23.9: PL/SQL-Code zum Einfügen eines zusätzlichen Dienstes.

Der Bereich **Show invoices** besteht aus einer Schaltfläche, die zum Erstellen von Rechnungen verwendet wird. In der Eigenschaft *Behaviour* der Schaltfläche haben wir *Action* definiert, die als *Redirect to Page in this Application* festgelegt wurde. Dann haben wir *Target* auf eine vordefinierte Kundenrechnungsseite gesetzt, die als modal eingestellt wurde. Auf dieser Seite wird mithilfe einer SQL-Abfrage ein klassischer Bericht angezeigt, der auch gedruckt werden kann:

```
SELECT f.price || ' €' as "Fixed Flatrate ",
SUM(CASE e.serviceid WHEN 1 THEN NVL(e.amount, 0) ELSE 0 END)
|| ' ' || listagg (CASE s.serviceid WHEN 1 THEN s.units ELSE '' END, '') AS "Extra
    SMS",
SUM(CASE e.serviceid WHEN 2 THEN NVL(e.amount, 0) ELSE 0 END)
|| ' ' || listagg (CASE s.serviceid WHEN 2 THEN s.units ELSE '' END, '') AS "Extra
    Minutes",
SUM(CASE e.serviceid WHEN 3 THEN NVL(e.amount, 0) ELSE 0 END)
|| ' ' || listagg (CASE s.serviceid WHEN 3 THEN s.units ELSE '' END, '') AS "Extra
    Data",
SUM(CASE e.serviceid WHEN 4 THEN NVL(e.amount, 0) ELSE 0 END)
|| ' ' || listagg (CASE s.serviceid WHEN 4 THEN s.units ELSE '' END, '') AS "Extra
    MMS",
(f.price
+ SUM(CASE e.serviceid WHEN 1 THEN NVL(e.amount, 0) * s.price ELSE 0 END)
+ SUM(CASE e.serviceid WHEN 2 THEN NVL(e.amount, 0) * s.price ELSE 0 END)
+ SUM(CASE e.serviceid WHEN 3 THEN NVL(e.amount, 0) * s.price ELSE 0 END)
+ SUM(CASE e.serviceid WHEN 4 THEN NVL(e.amount, 0) * s.price ELSE 0 END)
|| ' €') AS "Total"
FROM CH23_CUSTOMER c
JOIN CH23_PERSON p ON (p.personid = c.personid)
JOIN CH23_CUSTOMERFLATRATE cf ON (cf.customerid = c.customerid)
JOIN CH23_FLATRATE f ON (f.flatrateid = cf. flatrateid )
JOIN CH23_FLATRATESERVICE fs ON (fs.flatrateid = f.flatrateid)
JOIN CH23_SERVICE s ON (s.serviceid = fs.serviceid)
LEFT JOIN CH23_EXTRASERVICE e ON (e.customerid = cf.customerid)
WHERE p.email = :APP_USER
AND (e. startdate IS NULL OR
(EXTRACT(month from e.startdate) = EXTRACT(month from sysdate)
AND EXTRACT(year from e.startdate) = EXTRACT(year from sysdate)))
GROUP BY f.price;
```

Listing 23.10: SQL-Abfrage für einen Rechnungsbericht.

Der Bereich **Log** enthält einen interaktiven Bericht zur Anzeige eines Nutzungsprotokolls mit Hilfe einer SQL-Abfrage:

```
SELECT name, phone, amount || ' ' || units AS amount, when
FROM CH23_USAGELOG JOIN CH23_CUSTOMER USING (customerid)
JOIN CH23_PERSON USING (personid)
JOIN CH23_SERVICE USING (serviceid)
WHERE email = :APP_USER
ORDER BY when DESC;
```

Listing 23.11: SQL-Abfrage - Nutzungsprotokoll.

Der Reiter **Stats** ist eine neue Seite, die aus drei Bereichen besteht, wie Minuten, SMS und Daten. Der Bereich ist auf den Typ *Static Content* eingestellt, einschließlich der Komponenten *Chart*, die Diagramme aus Reihen anzeigen. Seine Quelle wird durch eine SQL-Abfrage definiert. Standardmäßig ist die Farbe der Diagrammbalken auf einheitlich eingestellt. Sie kann in der SQL-Abfrage angepasst werden:

```
SELECT TO_CHAR(when, 'MM/YYYY') AS "Month",
SUM(amount) AS Amount,
CASE WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '01'
THEN '#F8B195'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '02'
THEN '#F67280'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '03'
THEN '#C06C84'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '04'
THEN '#6C567B'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '05'
THEN '#ACDEAA'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '06'
THEN '#97D0ED'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '07'
THEN '#C23B23'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '08'
THEN '#6B7B8E'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '09'
THEN '#D5AF26'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '10'
THEN '#9AB5C1'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '11'
THEN '#74698C'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '12'
THEN '#C1867B'
ELSE 'black'
END AS MY_COLOR
FROM CH23_USAGELOG
JOIN CH23_CUSTOMER USING (customerid)
JOIN CH23_PERSON USING (personid)
JOIN CH23_SERVICE USING (serviceid)
WHERE email = :APP_USER AND serviceid = 2
GROUP BY TO_CHAR(when, 'MM/YYYY')
ORDER BY 1;
```

Listing 23.12: SQL-Abfrage - Statistiken.

## 23.10 Erstellen einer Managerseite

Auf der Managerseite haben wir ähnliche Methoden und Komponenten verwendet, um Unterseiten zu erstellen. Jede Unterseite wird als neue Seite erstellt, z.B. Add customer, Manage customers und View customers.

Seite *Add customer* enthält ein einfaches Formular, das aus Komponenten erstellt wird. Der Prozess des Hinzufügens eines neuen Kunden wird durch PL/SQL-Code ausgeführt:

```
INSERT INTO CH23_PERSON (EMAIL, PASSWORD, FIRSTNAME, LASTNAME,
    ROLE)
VALUES (:CUSTOMER_EMAIL, :CUSTOMER_PASSWORD, :
    CUSTOMER_FIRSTNAME,
```

```
:CUSTOMER_LASTNAME, 'C');
```

Listing 23.13: PL/SQL-Code zum Einfügen einer Person.

Seite *Manage customers* wird mit zwei interaktiven Gittern erstellt: Personen verwalten und Aktuelle Kunden verwalten. Die interaktiven Gitter sind mit dem Prozess der Bearbeitung und Speicherung von Änderungen verbunden. Der Prozesstyp ist auf *Interactive Grid – Automatic Row Processing (DML)* eingestellt.

Seite *View customers* enthält einen interaktiven Bericht, um Daten anzuzeigen, ohne sie zu bearbeiten. Dieser Bericht wird mit einer SQL-Abfrage erstellt:

```
SELECT firstname, lastname, email, address, phone
FROM CH23_PERSON
JOIN CH23_CUSTOMER USING (personid);
```

Listing 23.14: SQL-Abfrage - Kunden.

## 23.11 Ergänzendes Lernmaterial

Sie können das folgende zusätzliche Lernmaterial finden:

- Skript zum Erstellen und Auffüllen von Tabellen
- Skript zum Löschen von Tabellen
- Exportierte verpackte Anwendung
- Video, das zeigt, wie man die Anwendung erstellt

Alle zusätzlichen Lernmaterialien sind auf der öffentlichen BeeAPEX-Projektseite verfügbar. Melden Sie sich als Gastbenutzer an (kein Passwort erforderlich). Das Lehrbuch finden Sie im Abschnitt Books, die Skripte im Ordner Part 2 > Chapter23 im Abschnitt Scripts und die Videoanleitungen in der Sammlung der Videoanleitungen. Material für Kurzkurse finden Sie in der Sektion Short courses.

### 23.11.1 Exportierte Anwendungen

Die exportierte Anwendung wird verpackt. Bei der Installation werden sowohl Tabellen erstellt als auch Daten eingefügt. Bei der Deinstallation werden alle in dieser Anwendung verwendeten Datenbankobjekte entfernt.

Die verpackte Anwendung ist getestet und wird im neuen Arbeitsbereich ausgeführt, wenn die folgenden Anforderungen erfüllt sind:

- Fügen Sie den APEX-Benutzer hinzu, bevor Sie die Anwendung starten. Nur im Entwicklungs- und Testarbeitsbereich navigieren Sie zu Shared Components > Application Access Control > Add User Role Assignment; geben Sie den APEX-Benutzer ein und legen Sie für diesen Benutzer die Rollen Administrator, Contributer und Reader fest. In der Produktion ist die Rücksprache mit einem Experten vor dem Einsatz ein Muss.

Wenn dem Benutzer keine entsprechende Rolle zugewiesen wird, stürzt die importierte Anwendung ab. Es ist notwendig, das Webbrowser-Cookie zu löschen (z.B. Firefox: Settings > Cookies and Site Data > Manage Data), wenn die Anwendung aufgrund von nicht erfüllten Anforderungen abstürzt.

subsectionVideo-Anleitungen Videoanleitungen zeigen jeden Schritt der Anwendungsentwicklung.

## 23.12 Fragen

1. Welche Unterseite wird für die Erstellung eines neuen Anmeldeschemas verwendet?
2. Nennen Sie mindestens zwei Authentifizierungsmethoden für die Benutzerverwaltung.

3. Während der Implementierung wurde in den Select-Anweisungen auf verschiedene Elemente Bezug genommen. Wie können Sie diese Elemente identifizieren?

## 23.13 Antworten

1. Im Oracle APEX-Tool wählen wir die Authentifizierungsmethode - Authentication Scheme - auf der Unterseite Shared Components => Authentication Schemes.
2. Es gibt mehrere Methoden, die für die Benutzerauthentifizierung verwendet werden können, wie LDAP-Server, Datenbankkonto, Oracle APEX-Konto usw.
3. Elemente in den Select-Anweisungen werden mit einem Doppelpunkt am Anfang des Elementnamens referenziert.



# 24. Fall Autovermietung

A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis

## 24.1 Wirtschaftliche Sicht des Falles

Ein Auto zu mieten kann eine bequeme Möglichkeit sein, neue Ziele zu erkunden, aber es gibt einige potenzielle Fallstricke, auf die man achten sollte. Es ist wichtig zu verstehen, dass eine Autovermietung eher mit der Buchung eines Hotelzimmers als mit der Buchung eines Flugtickets vergleichbar ist. Bei der Anmietung eines Autos gibt es zwei Arten von Gebühren: diejenigen, die Sie bei der Abholung des Wagens bezahlen, und im Voraus bezahlte Preise.

Für Erstmieter ist es in der Regel am besten, eine Pay-Later-Rate zu buchen, die Ihnen mehr Flexibilität bietet, falls Sie Ihre Pläne ändern müssen. Bei dieser Art von Mietpreisen fällt in der Regel keine Stornogebühr an, was bei unvorhergesehenen Umständen hilfreich sein kann.

Um zusätzliche Kosten zu vermeiden, ist es wichtig, den Mietvertrag und eventuell anfallende Zusatzgebühren, wie z.B. Versicherungs- oder Zusatzfahrergebühren, sorgfältig zu prüfen. Außerdem sollten Sie das Fahrzeug sorgfältig inspizieren, bevor Sie die Anmietstation verlassen, und eventuelle Schäden sofort dem Vermieter melden, um zu vermeiden, dass Sie später dafür verantwortlich gemacht werden.

Wenn Sie sich über den Anmietvorgang und mögliche Gebühren informieren, können Sie eine reibungslose und stressfreie Anmietung gewährleisten und die Freiheit und Flexibilität, die das Mieten eines Autos auf Reisen bieten kann, voll und ganz genießen.

## 24.2 Problemstellung

Ein Auto zu mieten mag mit all den Regeln und Bedingungen abschreckend wirken, aber es gibt sie aus gutem Grund. Ein fabrikneues Auto kostet im Durchschnitt etwa 36.000 Euro. Wenn Sie also ein Auto mieten, zahlen Sie im Grunde nur einen Bruchteil des Gesamtwerts für die vorübergehende Nutzung des Fahrzeugs.

Obwohl es wichtig ist, eine angemessene Mietwagenversicherung für den Fall eines Unfalls oder unvorhergesehener Umstände abzuschließen, verlaufen die meisten Anmietungen ohne Zwischenfälle. Dennoch ist es wichtig, den Mietvertrag sorgfältig zu lesen, die Bedingungen zu verstehen und das Fahrzeug auf Schäden zu überprüfen, bevor Sie es annehmen, um mögliche Probleme zu vermeiden.

Wenn Sie diese Richtlinien befolgen, können Sie eine sichere und angenehme Anmietung ohne unerwartete Überraschungen erleben. Und denken Sie daran, dass die Regeln und Bedingungen so-

wohl Sie als auch die Autovermietung schützen sollen. Es ist also immer am besten, sie einzuhalten, damit die Anmietung reibungslos verläuft.

## 24.3 Anwendungsfälle

### 24.3.1 Narrative Beschreibung des Anwendungsfalls

Das Ziel dieses Projekts ist die Entwicklung einer funktionalen Anwendung, die den Betrieb einer Autovermietung simuliert. Die Anwendung wird die Oracle APEX-Plattform nutzen, um Kunden-, Fahrzeug- und Mietinformationen im Detail zu verwalten und zu speichern.

Die Anwendung ist so konzipiert, dass sie für den Inhaber einer Autovermietung benutzerfreundlich ist und es ihm ermöglicht, alle notwendigen Informationen zu verwalten, einschließlich Vermietungen, verfügbare Mietwagen und Kundendaten. Die Benutzer können ein Fahrzeug anhand von Kriterien wie Marke, Modell, Baujahr und Kraftstofftyp auswählen, und alle für die Anmietung erforderlichen Details werden angezeigt.

Zusätzlich zeigt die Anwendung Daten wie aktuelle Vermietungen, verfügbare Fahrzeuge und Kundendaten im Detail an. Die aktuellen Vermietungen werden dem Benutzer in Tabellen angezeigt, je nachdem, in welcher Registerkarte sich der Benutzer befindet.

Insgesamt bietet diese Anwendung ein effizientes und organisiertes System für die Verwaltung einer Autovermietung mit detaillierten Informationen, die auf Knopfdruck verfügbar sind. Die benutzerfreundliche Oberfläche und die umfassenden Datenmanagement-Tools machen den Betrieb eines Autoverleihs einfacher und effizienter.

### 24.3.2 Semi-strukturierte Beschreibung

Ziel dieses Projekts ist die Entwicklung einer benutzerfreundlichen Anwendung, die Oracle APEX nutzt, um detaillierte Informationen über Kunden, Fahrzeuge und Vermietungen für eine Autovermietung zu verwalten. Die Anwendung wird es den Benutzern ermöglichen, Anmietungen auf der Grundlage bestimmter Kriterien vorzunehmen, die erforderlichen Anmietungsinformationen anzuzeigen und umfassende Datenverwaltungswerkzeuge bereitzustellen, um aktuelle Anmietungen, verfügbare Fahrzeuge und Kundendetails anzuzeigen. Durch die Bereitstellung eines effizienten und organisierten Systems für die Verwaltung einer Autovermietung wird diese Anwendung den Betrieb einer Autovermietung einfacher und effizienter machen. Die Halbstrukturbeschreibung ist in Tabelle 24.1 enthalten.

### 24.3.3 Anwendungsfalldiagramm

Die obige Geschichte wird in einem Anwendungsfalldiagramm dargestellt (siehe Abbildung 24.1).

Tabelle 24.1: Beschreibung des Anwendungsfalls: Zugang zu Fahrzeugen, Kunden und Mietwagenreservierung

| **Keyword** | **Wert** |
|---|---|
| ID: | *Ch24-01* |
| Titel: | *Zugriff auf die Liste der Fahrzeuge und Kunden, die Möglichkeit, Daten hinzuzufügen und zu entfernen und eine neue Mietwagenreservierung zu erstellen, indem die gewünschten Daten ausgewählt und die Kosten berechnet werden.* |
| Beschreibung: | *Der Benutzer verwendet die APEX-Anwendung, um auf die Fahrzeuge des Unternehmens zuzugreifen und alle Kunden, die zuvor eine Buchung vorgenommen haben, anzuzeigen. Sowohl Fahrzeuge als auch Kunden werden durch eine Reihe von Attributen beschrieben. Darüber hinaus hat der Benutzer die Möglichkeit, Daten für beide Kategorien zu erstellen, zu aktualisieren und zu löschen, was eine effiziente Verwaltung des Autovermietungssystems ermöglicht, und er kann eine neue Mietwagenreservierung durch Hinzufügen des Fahrzeugs, des Kunden und der Daten erstellen. Das System berechnet die Gesamtkosten der Reservierung auf der Grundlage der ausgewählten Daten und aktualisiert die Ergebnisse entsprechend. Der abschließende Bericht enthält die Details der Reservierung, einschließlich des Fahrzeugs, des Kunden, der Mietdaten und der Gesamtkosten.* |
| Primärer Akteur: | *Benutzer-Administrator* |
| Voraussetzungen: | Der Benutzer hat ein Admin-Konto für die APEX-Instanz. |
| Nachbedingungen: | Nach der Aktualisierung und Auswahl eines Fahrzeugs und eines Kunden kann der Benutzer eine Mietwagenreservierung innerhalb der APEX-Instanz erstellen. Nach Abschluss einer Mietwagenreservierung und Berechnung der Gesamtkosten. |
| *Haupt* | Szenario |
| Erfolgs-Szenario: | 1. Öffnen Sie einen Webbrowser und melden Sie sich bei der Anwendung der Autovermietung an.<br>2. Wählen Sie den Menüpunkt oder die Seitennavigation mit der Bezeichnung **Cars** oder **costumers**.<br>3. Um ein Auto hinzuzufügen, klicken Sie auf die Schaltfläche „Create“, und für Kunden tun Sie dasselbe.<br>4. Um einen Fahrzeug- oder Kundeneintrag zu bearbeiten, klicken Sie auf das Bleistiftsymbol, das mit dem jeweiligen Eintrag verbunden ist.<br>5. Um Daten hinzuzufügen oder zu aktualisieren, klicken Sie auf die Schaltfläche „Apply Changes“. Um Daten zu löschen, klicken Sie auf die Schaltfläche „Delete“.<br>6. Wählen Sie einen Menüpunkt oder die Seitennavigation **rentcar** aus.<br>7. Um eine neue Miete zu erstellen, klicken Sie auf **create**.<br>8. Um ein Fahrzeug zur Reservierung hinzuzufügen, klicken Sie auf **carid** und wählen Sie aus der Liste der verfügbaren Fahrzeuge.<br>9. Um einen Kunden zur Fahrzeugreservierung hinzuzufügen, klicken Sie auf **custid** und wählen Sie aus der Liste.<br>10. Um die Daten der Reservierung festzulegen, klicken Sie auf **rent date** und wählen Sie das gewünschte Startdatum der Reservierung.<br>11. Klicken Sie dann auf das Feld **return date** und wählen Sie das Datum, an dem die Autovermietung endet.<br>12. Bei der Rückgabe des Fahrzeugs werden die Gesamtkosten der Reservierung auf der Grundlage der Anzahl der gemieteten Tage und eventueller Zusatzkosten berechnet. Im Falle einer Beschädigung des Fahrzeugs wird ein Aufschlag für die Reparaturkosten erhoben.<br>13. Die endgültigen Kosten werden dem Kunden zusammen mit einer detaillierten Aufschlüsselung der Kosten im Mietbericht mitgeteilt. |
| Erweiterungen: | *Keine* |
| Häufigkeit der Nut | *Die Anzahl der Personen, [illegible]* |

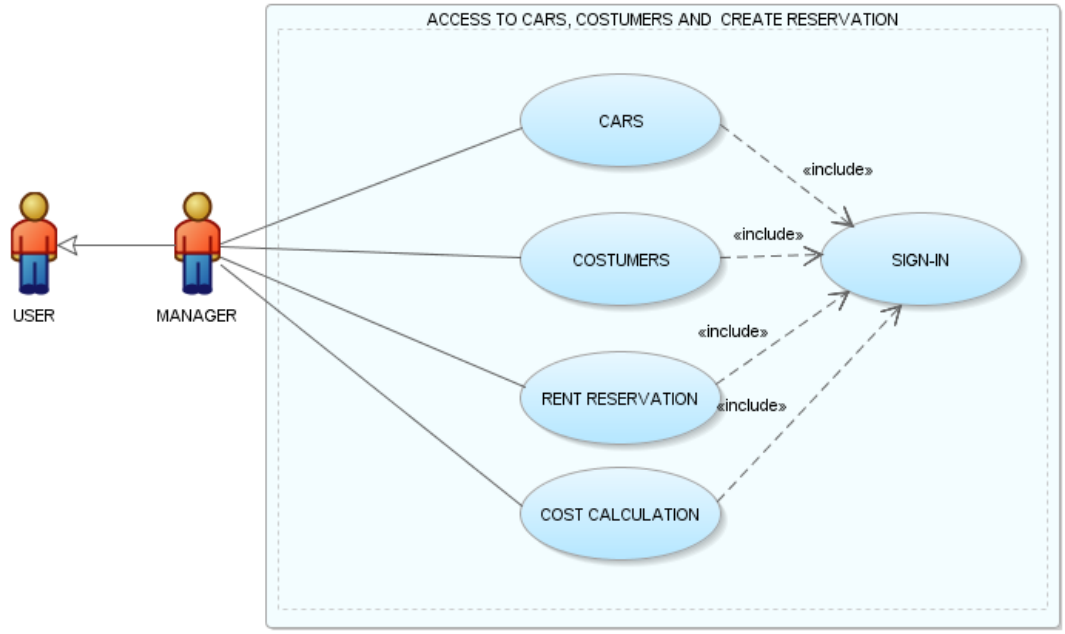



Abbildung 24.1: Anwendungsfalldiagramm.

## 24.4 Datenmodell

### 24.4.1 Narrative Beschreibung des Datenmodells

Das Lehrprojekt wird mit Hilfe von drei Tabellen implementiert, um die beteiligten Entitäten darzustellen: die Tabelle cars, die Tabelle customers und die Tabelle car rental. Diese Tabellen enthalten alle notwendigen Attribute und Daten, die für die Verwaltung des Autovermietungsprozesses über die APEX-Anwendung erforderlich sind.

1. Auto-Tabelle
   - **id** Identitätsnummer und Primärschlüssel,
   - **make** Automobilhersteller
   - **carmodel** das Modell des Fahrzeugs,
   - **fuel** Benzin, Diesel, LNG oder elektrischer Typ
   - **doors** wie viele Türen hat das Auto
   - **color** wie viele Türen hat das Auto
   - **rent** die Kosten für eine Tagesmiete eines bestimmten Fahrzeugs
   - **carphoto** ein Foto des Fahrzeugs
2. Kunden-Tabelle
   - **id** Identität die Nummer und der Primärschlüssel des des Kunden,
   - **fname** der Vorname des Kunden,
   - **lname** der Nachname des Kunden,
   - **address** die Adresse des Kunden,
   - **zipcode** die Postleitzahl der Adresse,
   - **city** die Stadt des Kunden,
   - **cardid** der Führerschein oder die Identitätskarte des Kunden
   - **telephone-work** ein Telefon der Arbeit des Kunden
   - **mobile** Mobiltelefon
   - **email** die E-Mail Adresse des Kunden
   - **comments** Kommentare, die sich auf das Verhalten der Person als Kunde beziehen

3. CarRental-Tabelle
    - **id** Identität die Nummer und der Primärschlüssel der Miettabelle
    - **carid id** Verbindungsfeld zwischen Autotabelle und Miettabelle
    - **custid id** Verbindungsfeld zwischen Kundentabelle und Miettabelle
    - **rent date** Mietdatum
    - **return date** Rückgabedatum,
    - **damage** Beschreibung eines eventuellen Schadens am Fahrzeug
    - **damage cost** die Schadenskosten
    - **total cost** die Gesamtkosten der Miete

### 24.4.2 Logisches Datenmodell

Das logische Datenmodell ist in Abbildung 24.2 dargestellt.

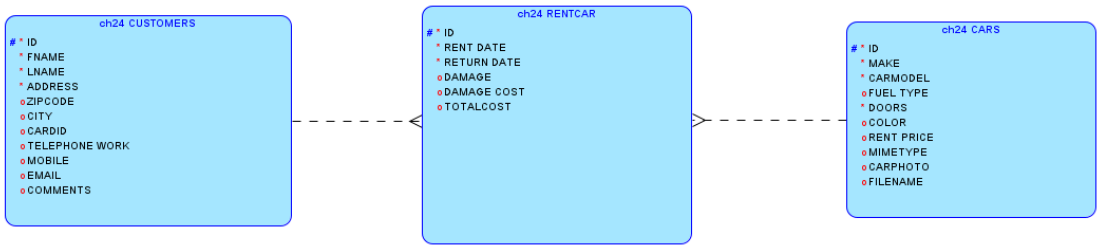



Abbildung 24.2: Logisches Modell des Autovermietungsprojekts

### 24.4.3 Relationales Datenmodell

Das relationale Datenmodell ist in Abbildung 24.3 dargestellt.

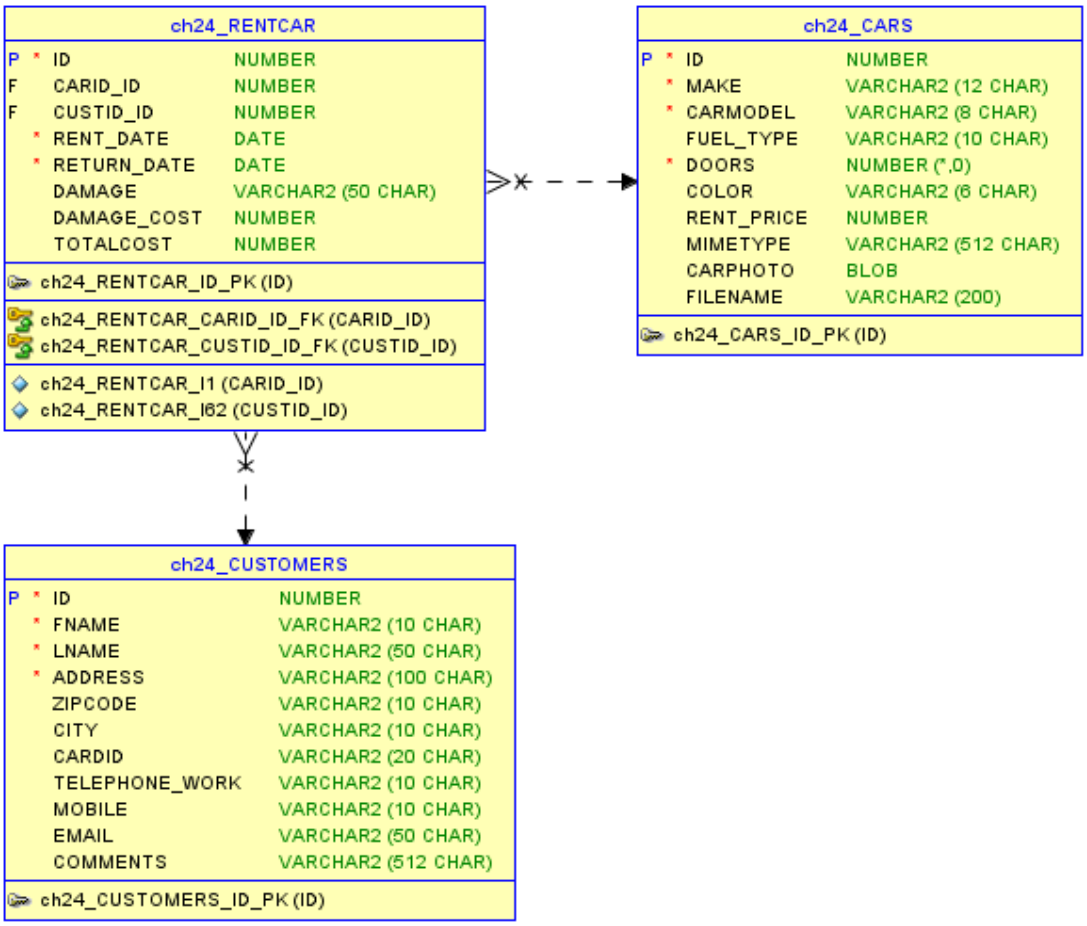



Abbildung 24.3: Relationales Modell des Autovermietungsprojekts

### 24.4.4 SQL-Skript

Die Tabellen, die wir für unsere Anwendung erstellt haben, sind CH24_CARS, CH24_CUSTOMERS und CH24_RENTCAR. Die Felder jeder Tabelle werden im Folgenden beschrieben:

```
drop table  ch24_cars;
drop table  ch24_customers;
drop table  ch24_rentcar;
-- create tables
create table ch24_cars (
    id                    number generated by default on null as identity
                            constraint ch24_cars_id_pk primary key,
    make                  varchar2(12 char) not null,
    carmodel              varchar2(8 char) not null,
    fuel_type             varchar2(10 char),
    doors                 integer not null,
    color                 varchar2(6 char),
    rent_price            number default '100',
    carphoto              varchar2(512 char)
)
;
```

```
create table ch24_customers (
    id                  number generated by default on null as identity
                           constraint ch24_customers_id_pk primary key,
    fname               varchar2(10 char) not null,
    lname               varchar2(50 char) not null,
    address             varchar2(100 char) not null,
    zipcode             varchar2(10 char),
    city                varchar2(10 char) default 'SERRES',
    cardid              varchar2(20 char),
    telephone_work      varchar2(10 char),
    mobile              varchar2(10 char),
    email               varchar2(50 char),
    comments            varchar2(512 char)
)
;
create table ch24_rentcar (
    id                  number generated by default on null as identity
                        constraint ch24_rentcar_id_pk primary key,
    carid_id            number
                        constraint ch24_rentcar_carid_id_fk
                        references ch24_cars on delete cascade,
    custid_id           number
                        constraint ch24_rentcar_custid_id_fk
                        references ch24_customers on delete cascade,
    rent_date           date not null,
    return_date         date not null,
    damage              varchar2(50 char),
    damage_cost         number,
    totalcost           number
)
;

-- table index
create index ch24_rentcar_i1 on ch24_rentcar (carid_id);
create index ch24_rentcar_i62 on ch24_rentcar (custid_id);
```

## 24.5 Anwendungsschnittstellen

Der Benutzer, der der Eigentümer der Autovermietung ist, muss seine Anmeldedaten angeben, um auf die Anwendung zuzugreifen. Die Anwendung verwendet die Application Express Accounts-Authentifizierung. Die Anmeldeinformationen des Eigentümers sind dieselben wie die in Abbildung 24.4 gezeigten WORKSPACE-Anmeldeinformationen.

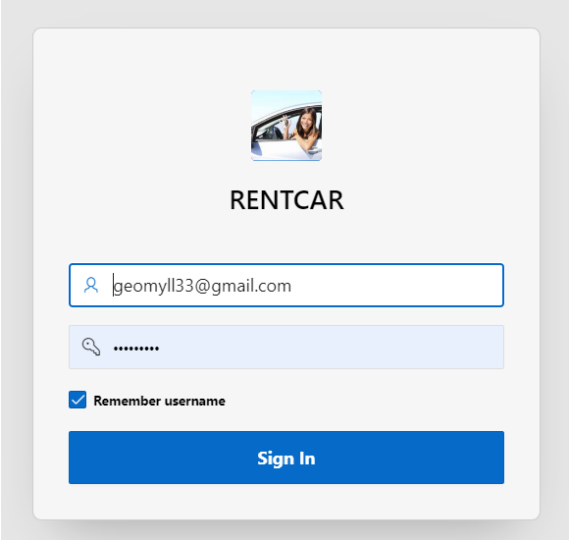



Abbildung 24.4: In der App anmelden.

Die Anwendung simuliert den Betrieb einer Autovermietung und umfasst die Kunden-, Fahrzeug- und Mietverwaltung. Sie liefert detaillierte Informationen über diese Einheiten mithilfe einer benutzerfreundlichen webbasierten Anwendung auf der Grundlage der Oracle APEX-Plattform, wie in Abbildung 24.5 dargestellt.

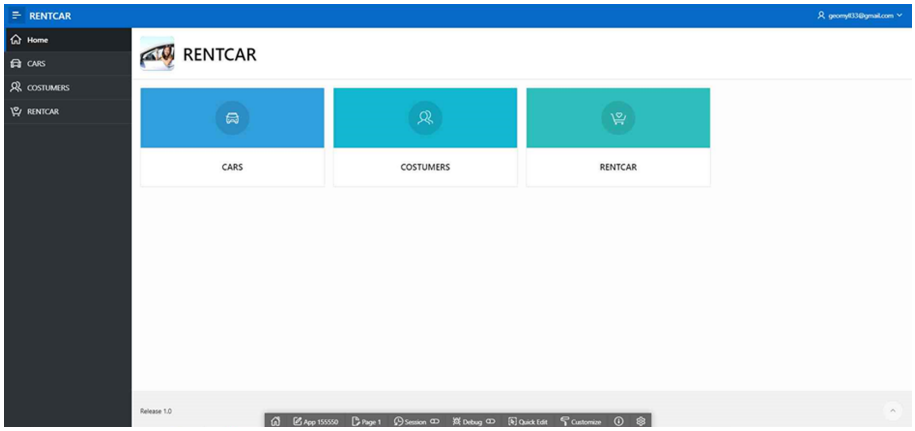



Abbildung 24.5: Startseite der Anwendung.

Die Vorlage in Abbildung 24.6 ist für die Verwaltung von Informationen über Autos in einer Autovermietung gedacht. Der Benutzer, der der Eigentümer des Ladens ist, kann alle notwendigen Informationen über die Autos speichern und verwalten, z.B. Marke, Modell, Baujahr, Kraftstofftyp, und auch ein Foto des Autos hochladen, um seinen Zustand zu zeigen.

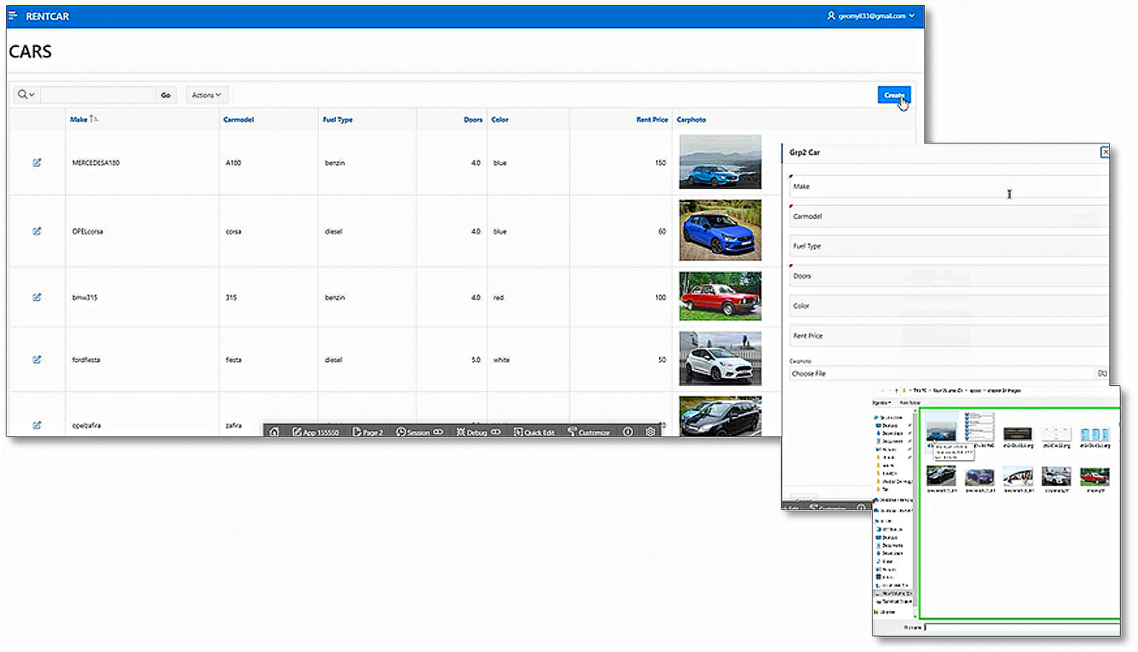



Abbildung 24.6: Vorlage für Autos

In der folgenden Abbildung 24.7 wird erklärt, wie Sie Ihrem Formular eine Spalte hinzufügen, in der Sie Fotos der Fahrzeuge anzeigen können. Um dies zu erreichen, benötigen Sie zwei Hilfsspalten, um die Informationen in Ihrer Tabelle zu speichern, eine für das Dateiformat und eine für den Dateinamen. Die Spalte, in der das Foto gespeichert wird, sollte auf den Speichertyp BLOB eingestellt werden. In unserem Fall lautet der Spaltenname **CARPHOTO**.

Im Anwendungsseitendesigner müssen Sie zum Bericht für die CAR-Vorlage gehen und die Spalte **CARPHOTO** als Dateibrowser konfigurieren. Ändern Sie dann die Einstellungen des Speichertyps auf „BLOB column specifier in item source attributes“.

Als Nächstes müssen Sie zum Formular für die CAR-Vorlage gehen und die Spalte **CARPHOTO** als Anzeigebild konfigurieren. Konfigurieren Sie die BLOB-Attribute, indem Sie die Tabelle **GRP2-CARS**, die Spalte **CARPHOTO** und den Primärschlüssel **ID** verbinden. Schließlich sollten die Spalten **MIMETYPE** und **FILENAME** auf versteckt gesetzt werden.

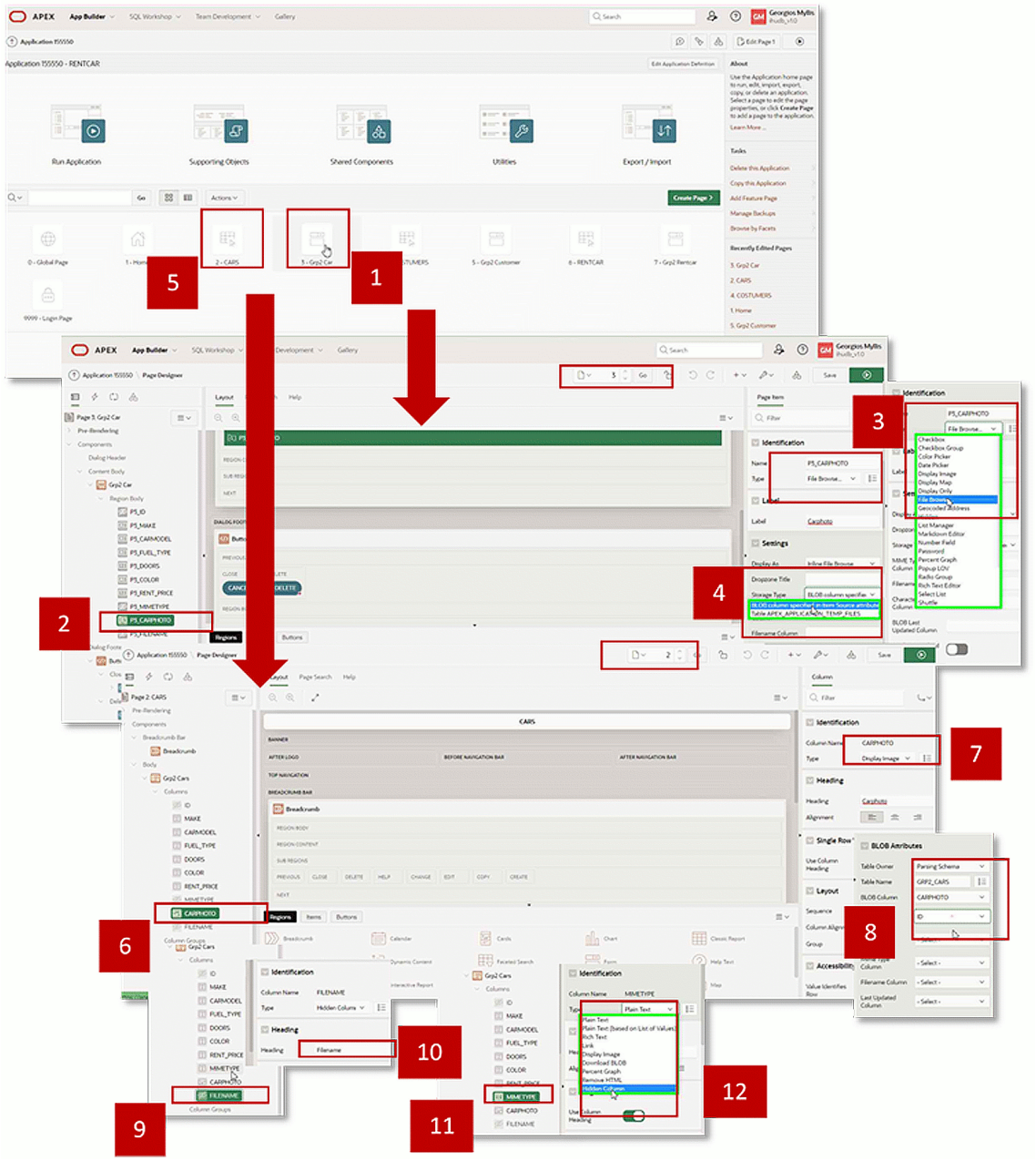



Abbildung 24.7: Wie man eine Kolumne mit Fotos erstellt.

In ähnlicher Weise können auch Kundendaten verwaltet werden (siehe Abbildung 24.8).

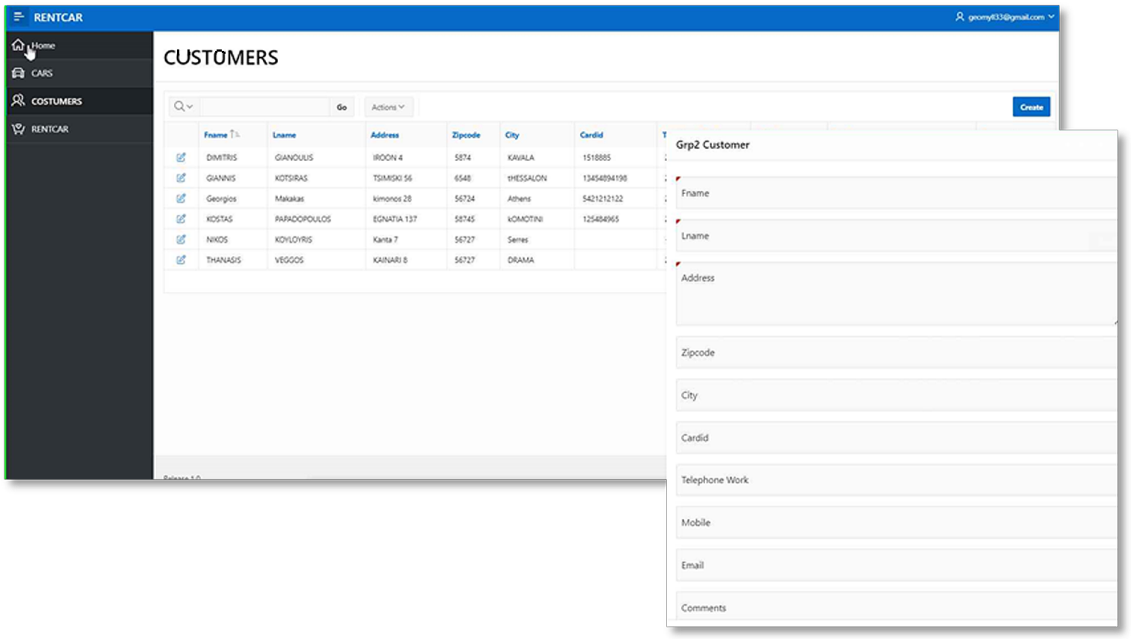



Abbildung 24.8: Kundendaten.

In der in Abbildung 24.9 gezeigten Anmiettabelle kann der Benutzer ein Auto anhand von Kriterien wie Hersteller, Modell und Baujahr auswählen. Die Anmietung wird dann mit dem Kunden verknüpft, und alle für die Anmietung erforderlichen Angaben wie Fahrzeug, Kunde, Start- und Enddatum der Anmietung sowie die Gesamtkosten werden angezeigt. Der Benutzer kann auch zusätzliche Kosten hinzufügen, wenn das Auto nach der Rückgabe einen Schaden aufweist. Sobald die Anmietung abgeschlossen ist, werden die Gesamtkosten berechnet und im Abschlussbericht dargestellt.

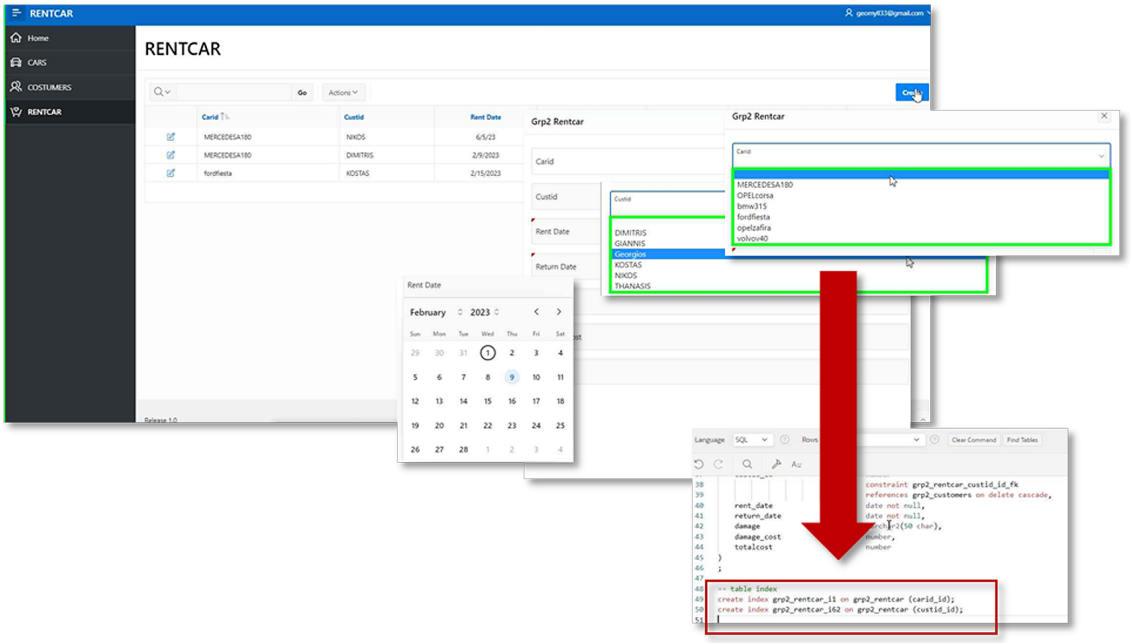



Abbildung 24.9: Vorlage für die Autovermietung.

## 24.6 Ergänzendes Lernmaterial

Sie können das folgende zusätzliche Lernmaterial finden:

- exportierte Anwendung
- Video-Guide

Alle zusätzlichen Lernmaterialien sind auf der öffentlichen BeeAPEX-Projektseite verfügbar. Melden Sie sich als Gastbenutzer an (kein Passwort erforderlich). Das Lehrbuch finden Sie im Abschnitt Books, die Skripte im Ordner Part 2 > Chapter24 im Abschnitt Scripts und die Videoanleitungen in der Sammlung der Videoanleitungen. Material für Kurzkurse finden Sie in der Sektion Short courses.

### 24.6.1 Exportierte Anwendungen

Die exportierte Anwendung ist verpackt. Bei der Installation werden sowohl Tabellen erstellt als auch Daten eingefügt. Bei der Deinstallation werden alle in dieser Anwendung verwendeten Datenbankobjekte entfernt.

### 24.6.2 Video-Anleitungen

Videoanleitungen zeigen jeden Schritt der Anwendungsentwicklung.

## 24.7 Fragen

1. Wie können Sie ein Symbol in Ihr App-Logo einfügen, wenn Sie die App bereits erstellt haben?
2. Wie können Sie Bilder zu Ihren Berichtsseiten hinzufügen?

## 24.8 Antworten

1. Wählen Sie in Ihrer Anwendungsumgebung **Shared Components** und dann **User Interface Attributes**. Klicken Sie im Popup-Fenster auf **Edit** und fügen Sie das gewünschte Bild hinzu.
2. Zunächst müssen Sie den Datentyp Ihrer Spalte als BLOB festlegen. Konfigurieren Sie dann im Anwendungsseitendesigner für Ihre Berichtsvorlage die Spalte als Dateibrowser und ändern Sie den Speichertyp in **BLOB column specifier in item Source Attributes**. Gehen Sie dann zum Formular dieser Vorlage in der App Builder Umgebung und konfigurieren Sie die Spalte als **Display image**. Konfigurieren Sie dann die BLOB-Attribute, indem Sie die Tabelle, die Spalte und die Primärschlüssel-ID, die sich auf diese Spalte bezieht, verbinden.

# Literaturverzeichnis

## Artikel

[1] Hill Kim. „Altruistic cooperation during foraging by the Ache, and the evolved human predisposition to cooperate“. In: *Human Nature* 13 (März 2002), Seiten 105–128. DOI: 10.1007/s12110-002-1016-3 (siehe Seite 187).

## Bücher

[2] Roy Fielding und Richard N. Taylor. *Principled design of the modern Web architecture*. Herausgegeben von Mehdi Jazayeri andAlexander L. Wolf Carlo Ghezzi. Association for Computing Machinery, 2010. ISBN: 978-1-58113-206-9. DOI: 10.1145/337180.337228 (siehe Seiten 104, 105).

[3] Dariusz Jemielniak und Aleksandra Przegalinska. *Collaborative society*. MIT Press, 2020 (siehe Seiten 187, 188).

[4] OMG. *Business Process Model and Notation (BPMN), Version 2.0*. 2011. URL: http://www.omg.org/spec/BPMN/2.0 (siehe Seiten 272, 273).

[5] OASIS OPEN. *Universal Business Language Version 2.1, OASIS Standard, 04 November 2013*. 2013. URL: http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.pdf (siehe Seite 272).

[6] A. Osterwalder und Y. Pigneur. *Business model generation: a handbook for visionaries, game changers, and challengers*. Band 1. John Wiley und Sons, 2010 (siehe Seiten 38, 39).

# Index


Abstraktionsebene, 52
Anwendung erstellen, 87
Anwendungsbericht, 103
Anwendungsfalldiagramm, 242, 252, 304, 340, 380, 406
Anwendungsfälle, 230, 242, 252, 259, 276, 304, 310, 324, 339, 360, 379, 406
Anwendungslogik, 87
Anwendungsschnittstellen, 236, 246, 255, 265, 285, 307, 316, 334, 346, 368, 386, 411
APEX docker, 46
Application Programming Interface, API, 105
attribute, 53
AutoREST, 109
Autovermietung, 405

Beispiel- und Startanwendungen, 199
Benutzerautorisierung, 251
Benutzerfeedback, 86
Berechnung, 87
Beziehung, 53

Create Application Wizard, 84
Create Page Wizard, 86
CRUD-Operation, 105, 108

Data model, 252, 324
data workshop, 94, 100
datenbank, 52
datenbank management system, 52
Datenbanksystem, 51
Datenmanipulation, 70
Datenmodell, 52, 234, 244, 260, 281, 304, 312, 340, 364, 382, 408
Datenstruktur, 52
datentyp, 53
DB, 52
DB schema, 56
DB Schicht, 52
DB-Schicht, 80
DBMS, 52
DBS, 51
DDL, 62
Diagramm, 87
DML, 70
domäne, 53
DQL, 72
Dritte Normalform, 3NF, 62

entity relationship diagram, ERD, 52
Entity-Relationship-Diagramm, ERD, 53
entität, 53
entitätsinstanz, 56
ER, 52
ER Diagramm, 52
ER-Diagramm, 53
Erste Normalform, 1NF, 61
Erstellung relationaler Modelle, 58
Export-Assistent, 100
Exportieren von Daten, 100

Flows für APEX installieren, 274
foreign key, 59

HTTP request, 108
HTTP response, 106
HTTP-Anfrage, 105
https://apex.oracle.com, 44

Importieren von Daten, 94
interaktiver Bericht, 87
item, 87

Kalender, 87
Kardinalität, 53

list of values, LOV, 214
Liste, 87
load wizard, 96
Logisches Datenmodell, 244, 254, 306, 340, 382, 409
logisches Datenmodell, 52
logisches Modell, 52

Management von Geschäftsprozessen, 271
mehrere Sprachen, 221

Normalisierung, 61

Object Browser, 69
object browser, 101
on-premise, 44
Oracle Academy, 48
ORACLE APEX, 81
ORACLE application builder, 83
Oracle Cloud Infrastruktur, 48
ORACLE Data Modeler, 63
ORACLE Object Browser, 71
ORACLE Query Builder, 77
Oracle REST data service, ORDS, 105, 106
ORACLE SQL Developer Data Modeler, 52
ORACLE SQL workshop, 69

page designer, 89
PL/SQL, 210
Problemstellung, 229, 241, 251, 259, 275, 303, 309, 323, 339, 360, 379, 405

Quick SQL, 65, 71, 120
QuickSQL und Script SQL Skript, 343

RDB, 56
RDB schema, 56, 60
RDB schema Erstellung, 57
RDBS-Schemaverwaltung, 62
region, 87
relation, 56
Relationales Datenmodell, 246, 254, 306, 342, 365, 383, 409
relationales DB-Schema, 56
relationales Modell, 52, 56
representational state transfer, REST, 104
Ressourcen-Handler, 108
Ressourcen-Template, 108
Ressourcenmodul, 108
REST, 104
RESTful service, 104
RESTful-Zugriff, 106

Seite, 90
seite, 87
Seitendesigner, 89
Seitenerstellung, 86
Seitentyp, 87
skill level, 43
Spalte, 56
Sprechstundenplanung, 359
SQL ALTER TABLE, 69
SQL COUNT, 76
SQL CREATE TABLE, 68
SQL data query language, SQL-DQL, 72
SQL DELETE, 71
SQL DROP TABLE, 68
SQL INSERT, 70
SQL JOIN, 75
SQL ORDER BY, 73, 76, 77
SQL SELECT, 73, 102
SQL Skript, 121
SQL SUM, 76
SQL TO_CHAR(), 74
SQL UPDATE, 70
SQL WHERE-Klausel, 70, 74
SQL-Aggregation, 76
SQL-Datendefinitionssprache, 67
SQL-Datendefinitionssprache, SQL-DDL, 62
SQL-Datenmanipulationssprache, SQL-DML, 70
SQL-DDL, 62, 70
SQL-DQL, 72
structured query language, SQL, 62
Stückliste, 323

tabelle, 56
Telekommunikationsdienste, 379

use case model, UC, 40

validierung, 87
verpackte Anwendung, 209
Virtual Box Appliance, 45

Webanwendung, 51, 90
Webanwendungsentwicklungsprozess, 81
Wie erstellt man den ersten Entwurf einer Anwendung, 113
wie kann man von der Galerie der Anwendungen und Plug-ins profitieren, 199
Wie man Daten mit APEX austauscht, 93
Wie man im Team zusammenarbeitet, 187
wie navigiert man in APEX, 81
Wie wird eine Datenbank vorbereitet, 51
Wie wird Oracle APEX gestartet, 35
Wirtschaftliche Sicht des Falles, 229, 241, 251, 259, 271, 303, 309, 323, 339, 359, 379, 405
Workflow-Modell, 278

Zugangskontrolle, 86
zustandsloses REST, 105
zweite Normalform, 2NF, 61