Funded by the European Union

BEE WITH APEX

E-book włączony Programowanie niskokodowe w APEX

Praktyczny przewodnik dotyczący lepszych zdolności do zatrudnienia dla każdego

Zespół BeeAPEX



Copyright © 2023 BeeAPEX Team

PUBLISHED BY SOFTWARE QUALITY AND TESTING LAB, UM FOV

BEEAPEX.EU

Licensed under the Creative Commons Attribution-NonCommercial 4.0 License (the "License"). You may not use this file except in compliance with the License. You may obtain a copy of the License at https://creativecommons.org/licenses/by-nc-sa/4.0. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1st edition, 7 grudnia 2023

Spis treści

Podziękowania 27

Wprowadzenie 29

I How to in APEX

1 Jak uruchomic Oracle APEX? 35

Vjeran Strahonja, Dijana Oreški . . . . .

1.1 Co to jest Oracle APEX i do czego służy? 35
1.1.1 Co to jest Oracle APEX? 35
1.1.2 Co to jest aplikacja? 35
1.1.3 Jak Oracle APEX może pomóc w tworzeniu aplikacji? 35
1.1.4 Cykl tworzenia aplikacji 36
1.2 Jak uruchomić Oracle APEX? 42
1.2.1 Jaki jest Twój poziom umiejętności? 42
1.2.2 Użyj lokalnej instancji APEX 43
1.2.3 https://apex.oracle.com 44
1.2.4 Virtual Box Appliance (maszyna wirtualna) 44
1.2.5 Docker APEX (klucz dostępu do APEX) 45
1.2.6 Instancja APEX w Oracle Cloud Infrastructure 47
1.2.7 Instancja APEX w Oracle Academy 47
1.3 Pytania 47
1.4 Answers 48

2 Jak przygotować bazę danych? 49

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

2.1 Jak modelować dane? 50
2.1.1 Model logiczny 50
2.1.2 Model relacyjny 54
2.1.3 Normalizacja schematu bazy danych RDB 59

2.2 Jak zarządzać schematami RDBS - SQL-DDL . . . 60
2.2.1 Automatyczne generowanie tabel przy użyciu ODM . . . 61
2.2.2 Automatyczne generowanie tabel przy użyciu Quick SQL . . . 63
2.2.3 Ręczne tworzenie tabel przy użyciu SQL-DDL . . . 65
2.2.4 Ręczne zarządzanie tabelami za pomocą przeglądarki obiektów . . . 67
2.3 Jak manipulować danymi - SQL-DM . . . 68
2.3.1 Manipulacja danymi przy użyciu SQL-DML . . . 68
2.3.2 Manipulacja danymi za pomocą przeglądarki obiektów ORACLE (OB) . . . 69
2.3.3 Manipulacja danymi przy użyciu Quick SQL . . . 69
2.4 Jak wyszukiwać dane - SQL-DQL . . . 70
2.4.1 Zapytanie o dane przy użyciu SQL-DQL . . . 70
2.4.2 Zapytanie o dane przy użyciu narzędzia Query Builder ("QB") firmy ORACLE . 75
2.5 Budowanie warstwy DB - szerszy obraz . . . 78
2.6 Pytania . . . 79
2.7 Odpowiedzi . . . 79

3 Jak się poruszać w APEX? . . . 81
E. Kapsammer, W. Retschitzegger, W. Schwinger . . .
3.1 Proces tworzenia aplikacji internetowych realizowany przez APEX . . . 81
3.2 Menu startowe APEX . . . 83
3.3 Przegląd aplikacji App Builder - tworzenie i zarządzanie aplikacjami . . 83
3.4 Tworzenie aplikacji - trzy przypadki użycia . . . 84
3.5 Tworzenie aplikacji - właściwości, strony, funkcje i ustawienia . . . 85
3.6 Określanie stron . . . 87
3.7 Utrzymywanie i modyfikowanie strony – Page Designer . . . 89
3.8 Uruchamianie aplikacji lub strony . . . 89
3.9 Pytania . . . 91
3.10 Answers . . . 91

4 Jak wymieniać dane w APEX? . . . 93
E. Kapsammer, W. Retschitzegger, W. Schwinger . . .
4.1 Importowanie i eksportowanie danych przy użyciu Data Workshop . . . 94
4.1.1 Importowanie danych . . . 94
4.1.2 Kroki importowanie . . . 96
4.1.3 Eksportowanie danych . . . 100
4.2 Importowanie i eksportowanie danych przy użyciu "Object Browser" . 101
4.3 Eksportowanie wyniku polecenia SQL . . . 102
4.4 Eksportowanie danych z raportu aplikacji . . . 103
4.5 Umożliwienie wymiany danych za pomocą usług RESTful . . . 104
4.5.1 Wzorzec architektoniczny REST . . . 105
4.5.2 Włączanie schematu DB dla dostępu RESTful . . . 106
4.5.3 Moduły zasobów | Szablony | Programy obsługi . . . 107
4.5.4 Używanie ĄutoRESTźamiast ręcznego definiowania zasobów . . . 109

4.6 Pytania . . . 111
4.7 Answers . . . 111

5 Jak wygenerować pierwszy szkic wniosku? . . . 113
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . . .

5.1 Dlaczego biznes potrzebuje aplikacji? . . . 113
5.2 Konfiguracja środowiska ORACLE APEX . . . 113
5.3 Tworzenie przykładów warsztatów SQL . . . 117
5.3.1 Quick SQL . . . 121
5.3.2 Skrypt SQL . . . 122
5.4 Część aplikacji oparta na danych . . . 125
5.5 Administrowanie aplikacją . . . 128
5.6 Kontrola dostępu . . . 128
5.7 Dodatkowe materiały edukacyjne . . . 134
5.7.1 Wyeksportowane aplikacje . . . 134
5.7.2 Przewodniki wideo . . . 134
5.8 Pytania . . . 134
5.9 Pytania . . . 135

6 Jak zarządzać raportami? . . . 137
Zuzana Žillová, Erik Malina, Matej Grochal, Andrej Staník, Andrea Meleková, Michal Kvet, Miroslav Potočár . . .

6.1 Raport . . . 137
6.2 Raport klasyczny . . . 146
6.3 Raport przełączania kolumn . . . 148
6.4 Raport interaktywny . . . 148
6.5 Pytania . . . 157
6.6 Answers . . . 157

7 Jak zarządzać formualrzami? . . . 159
Veronika Šalgová, Michal Kvet, Miroslav Potočár . . .

7.1 Rodzaje formularzy . . . 159
7.2 Edytowalna interaktywna siatka . . . 159
7.3 Formularz w tabeli . . . 162
7.4 Główny formularz szczegółowy . . . 162
7.5 Pytania . . . 165
7.6 Odpowiedzi . . . 165

8 Jak przekształcić raporty tekstowe w wykresy? . . . 167
Ivan Pastierik, Michal Kvet, Miroslav Potočár . . .

8.1 Wykresy . . . 167
8.2 Tworzenie wykresu słupkowego . . . 168

8.3 Dodawanie filtrowania do wykresu słupkowego . . . 172
8.4 Dodawanie sortowania do wykresu słupkowego . . . 177
8.5 Tworzenie różnych typów wykresów . . . 181
8.6 Pytania . . . 182
8.7 Odpowiedzi . . . 183

9 Jak zarządzać menu? . . . 185
Veronika Šalgová, Michal Kvet, Miroslav Potočár . . .

9.1 Jak zarządzać menu? . . . 185
9.2 Menu boczne . . . 185
9.3 Górne menu . . . 186
9.4 Mega Menu . . . 186
9.5 Edycja list menu . . . 187
9.6 Pytania . . . 187
9.7 Odpowiedzi . . . 187

10 Jak współpracować w zespole? . . . 189
P. Staniszewski, M. Sońta, A. Kierzkowski . . .

10.1 Wspólne tworzenie wiedzy jest istotą rozwoju niskokodowego . . . 189
10.2 Bycie razem online . . . 189
10.3 Znajomość technologii dzięki współpracy . . . 190
10.4 Opis funkcji . . . 191
10.4.1 Blokada strony . . . 191
10.4.2 Komentarze . . . 192
10.4.3 Opcje budowy . . . 193
10.4.4 Rozwój zespołu . . . 193
10.4.5 Informacje zwrotne . . . 196
10.4.6 Standaryzacja . . . 197
10.5 Wnioski . . . 198
10.6 Pytania . . . 199
10.7 Odpowiedzi . . . 199

11 Jak skorzystać z galerii aplikacji i wtyczek? . . . 201
Vjeran Strahonja, Dijana Oreški . . .

11.1 Jak zainstalować przykładowe i startowe aplikacje? . . . 201
11.2 Starter Apps (Aplikacje początkowe) . . . 202
11.3 Sample Apps (Aplikacje przykładowe) . . . 204
11.4 Wtyczki . . . 205
11.5 Pytania . . . 208
11.6 Odpowiedzi . . . 208

12 Jak zarządzać pakietowymi i wielojęzycznymi aplikacjami? ... 209
R. Leskovar, U. Rajkovič, A. Baggia
12.1 Aplikacja i aplikacja pakietowa ... 209
12.2 Aplikacja ... 211
12.2.1 Zakres zastosowania ... 211
12.2.2 Tworzenie tabel ... 211
12.2.3 Wstawianie danych ... 212
12.2.4 Generuj aplikacje ... 213
12.3 Aplikacja pakietowa ... 217
12.4 Aplikacja wielojęzyczna ... 221
12.5 Dodatkowe materiały szkoleniowe ... 225
12.5.1 Eksportowane aplikacje ... 225
12.5.2 Przewodniki video ... 225
12.6 Pytania ... 226
12.7 Odpowiedzi ... 226

II Constructing application in APEX

13 Intranetowe wiadomości dla pracowników ... 229
R. Leskovar, U. Rajkovič, A.Baggia
13.1 Biznesowy pogląd na sprawe ... 229
13.2 Definicja problemu ... 229
13.3 Przypadki użycia ... 230
13.3.1 Opis przypadku użycia ... 230
13.3.2 Opis częściowo ustrukturyzowany ... 230
13.3.3 Diagram przypadków użycia ... 230
13.4 Model danych ... 234
13.4.1 Opis modelu danych ... 234
13.4.2 Logiczny model danych ... 234
13.4.3 Relacyjny model danych ... 235
13.5 Interfejsy aplikacji ... 236
13.6 Dodatkowe materiały do nauki ... 238
13.6.1 Eksportowana aplikacja ... 238
13.6.2 Przewodniki video ... 238
13.7 Pytania ... 238
13.8 Odpowiedzi ... 238

14 GreenDi - Katalog roślin ... 241
Vjeran Strahonja, Dijana Oreški, Darko Andročec, Ana Kutnjak
14.1 Biznesowy pogląd na sprawę ... 241
14.2 Definicja problemu ... 241
14.3 Przypadki użycia ... 242
14.3.1 Opis przypadku użycia ... 242

14.3.2 Opis częściowo ustrukturyzowany . . . 242
14.3.3 Schemat przypadków użycia . . . 242
14.4 Model danych . . . 244
14.4.1 Opis modelu danych . . . 244
14.4.2 Logiczny model danych . . . 244
14.4.3 Relacyjny model danych . . . 246
14.5 Interfejsy aplikacji . . . 246
14.6 Dodatkowe materiały do nauki . . . 247
14.6.1 Eksportowana aplikacja . . . 247
14.6.2 Przewodniki wideo . . . 248
14.7 Pytania . . . 248
14.8 Odpowiedzi . . . 248

15 GreenDi - Autoryzacja i zarządzanie użytkownikami . . . 249
VJERAN STRAHONJA, DARKO ANDROČEC, ANA KUTNJAK, LARISA HRUSTEK . . .

15.1 Biznesowy pogląd na sprawę . . . 249
15.2 Definicja problemu . . . 249
15.3 Przypadki użycia . . . 250
15.3.1 Opis przypadku użycia . . . 250
15.3.2 Opis częściowo ustrukturyzowany . . . 250
15.3.3 Diagram przypadków użycia . . . 250
15.4 Model danych . . . 252
15.4.1 Opis modelu danych . . . 252
15.4.2 Logiczny model danych . . . 252
15.4.3 Relacyjny model danych . . . 253
15.5 Interfejsy aplikacji . . . 253
15.6 Dodatkowe materiały do nauki . . . 254
15.6.1 Eksportowana aplikacja . . . 254
15.6.2 Przewodniki wideo . . . 254
15.7 Pytania . . . 255
15.8 Odpowiedzi . . . 255

16 System małych innowacji . . . 257
R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA . . .

16.1 Biznesowy pogląd na sprawę . . . 257
16.2 Definicja problemu . . . 257
16.3 Przypadki użycia . . . 258
16.3.1 Opis przypadku użycia . . . 258
16.3.2 Opis częściowo ustrukturyzowany . . . 258
16.3.3 Schemat przypadków użycia . . . 258
16.4 Model danych . . . 258
16.4.1 Opis modelu danych . . . 258
16.4.2 Logiczny model danych . . . 262
16.4.3 Relacyjny model danych . . . 262

16.5 Interfejsy aplikacji . . . 263
16.6 Dodatkowe materiały do nauki . . . 266
16.6.1 Eksportowana aplikacja . . . 267
16.6.2 Przewodniki wideo . . . 267
16.7 Pytania . . . 267
16.8 Odpowiedzi . . . 267

17 Zarządzanie procesami biznesowymi . . . 269
R. Leskovar, U. Rajkovič, A.Baggia

17.1 Biznesowy pogląd na sprawę . . . 269
17.2 Rozpoczęcie pracy z aplikacją Flows dla APEX . . . 272
17.2.1 Instalowanie Flow dla APEX . . . 272
17.2.2 Instalowanie przykładowej aplikacji przepływu procesu . . . 272
17.2.3 Czytanie i ćwiczenia praktyczne . . . 272
17.3 Definicja problemu . . . 273
17.4 Przypadki użycia . . . 273
17.4.1 Opis narracji . . . 273
17.4.2 Opis częściowo ustrukturyzowany . . . 274
17.4.3 Schemat przypadków użycia . . . 274
17.5 Model przepływu pracy . . . 276
17.6 Model danych . . . 279
17.6.1 Opis modelu danych . . . 279
17.6.2 Logiczny model danych . . . 279
17.6.3 Relacyjny model danych . . . 280
17.6.4 Quick SQL do generowania skryptu SQL . . . 280
17.6.5 Sekwencja i dwie przechowywane funkcje . . . 282
17.7 Interfejsy aplikacji . . . 283
17.7.1 Lista wartości w komponentach współdzielonych . . . 283
17.7.2 Wtyczki w komponentach współużytkowanych . . . 284
17.7.3 Kierownik sprzedaży . . . 284
17.7.4 Kierownik produkcji . . . 289
17.7.5 Dyrektor finansowy . . . 291
17.7.6 Dyrektor Generalny - dyrektor handlowy . . . 292
17.8 Łączenie aplikacji z usługą Flows for APEX . . . 294
17.9 Definiowanie ról użytkowników . . . 296
17.10 Testowanie i korygowanie błędów . . . 296
17.11 Dodatkowy materiał do nauki . . . 297
17.11.1 Eksportowana aplikacja . . . 297
17.11.2 Przewodniki wideo . . . 299
17.12 Pytania . . . 299
17.13 Odpowiedzi . . . 299

18 GreenDi – wymiana roślin i nasion . . . 301
Vjeran Strahonja, Dijana Oreški, Darko Andročec, Ana Kutnjak

18.1 Biznesowy pogląd na sprawę . . . 301

18.2 Definicja problemu . . . 301
18.3 Przypadki użycia . . . 302
18.3.1 Opis przypadku użycia . . . 302
18.3.2 Opis częściowo ustrukturyzowany . . . 302
18.3.3 Diagram przypadków użycia . . . 302
18.4 Model danych . . . 302
18.4.1 Narrative description of data model . . . 302
18.4.2 Logiczny model danych . . . 304
18.4.3 Relacyjny model danych . . . 304
18.5 Interfejsy aplikacji . . . 305
18.6 Dodatkowe materiały do nauki . . . 306
18.6.1 Eksportowana aplikacja . . . 306
18.6.2 Przewodniki wideo . . . 306
18.7 Pytania . . . 306
18.8 Odpowiedzi . . . 306

19 System zarządzania recenzjami książek . . . 309
A. Kutnjak, L. Hrustek, A. Baggia and R.Leskovar . . .

19.1 Biznesowy pogląd na sprawę . . . 309
19.2 Definicja problemu . . . 309
19.3 Przypadki użycia . . . 310
19.3.1 Opis przypadku użycia . . . 310
19.3.2 Opis częściowo ustrukturyzowany . . . 310
19.3.3 Diagram przypadków użycia . . . 310
19.4 Model danych . . . 312
19.4.1 Opis modelu danych . . . 312
19.4.2 Logiczny model danych . . . 313
19.4.3 Relacyjny model danych . . . 313
19.4.4 Skrypt SQL . . . 314
19.4.5 Szybki SQL . . . 315
19.5 Interfejsy aplikacji . . . 316
19.5.1 Administrator . . . 316
19.5.2 Użytkownik . . . 318
19.6 Definiowanie ról użytkowników . . . 320
19.7 Dodatkowe materiały do nauki . . . 320
19.7.1 Eksportowana aplikacja . . . 320
19.7.2 Przewodniki wideo . . . 321
19.8 Pytania . . . 321
19.9 Odpowiedzi . . . 321

20 Zestawienie materiałów i kalkulacja kosztów . . . 323
R. Leskovar, U. Rajkovič, A.Baggia . . .

20.1 Biznesowy pogląd na sprawę . . . 323
20.2 Definicja problemu . . . 323

20.3 Przypadki użycia . . . 324
20.3.1 Opis narracyjny . . . 324
20.3.2 Opis częściowo ustrukturyzowany . . . 324
20.3.3 Diagram przypadków użycia . . . 327
20.4 Model danych . . . 327
20.4.1 Opis modelu danych . . . 327
20.4.2 Implementacja reguł biznesowych w bazie danych . . . 328
20.4.3 Logiczny model danych . . . 328
20.4.4 Relacyjny model danych . . . 328
20.4.5 Obiekty w APEX . . . 330
20.5 Interfejsy aplikacji . . . 333
20.6 Dodatkowe materiały do nauki . . . 336
20.6.1 Eksportowane aplikacje . . . 336
20.6.2 Przewodniki wideo . . . 336
20.7 Pytania . . . 336
20.8 Odpowiedzi . . . 336

21 Odżywianie i zarządzanie dietą . . . 339
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . . .

21.1 Biznesowy pogląd na sprawę . . . 339
21.2 Definicja problemu . . . 339
21.3 Przypadki użycia . . . 339
21.3.1 Opis narracyjny . . . 339
21.3.2 Opis częściowo ustrukturyzowany . . . 340
21.3.3 Schemat przypadków użycia . . . 340
21.4 Model danych . . . 340
21.4.1 Logiczny model danych . . . 340
21.4.2 Relacyjny model danych . . . 342
21.4.3 QuickSQL i skrypt SQL Script . . . 343
21.5 Interfejsy aplikacji . . . 346
21.5.1 Jak zbudować pulpit nawigacyjny? . . . 350
21.5.2 Programiści i użytkownicy aplikacji . . . 354
21.6 Dodatkowe materiały do nauki . . . 357
21.6.1 Eksportowana aplikacja . . . 357
21.6.2 Przewodniki wideo . . . 357
21.7 Pytania . . . 357
21.8 Odpowiedzi . . . 358

22 Planowanie godzin pracy biura . . . 359
J. Mańko, M. Sońta, R. Leskovar . . .

22.1 Biznesowy pogląd na sprawę . . . 359
22.2 Definicja problemu . . . 360
22.3 Przypadki użycia . . . 360
22.3.1 Opis narracyjny . . . 361
22.3.2 Opis częściowo ustrukturyzowany . . . 361
22.3.3 Schemat przypadków użycia . . . 361

22.4 Model danych . . . 364
22.4.1 Opis modelu danych . . . 364
22.4.2 Logiczny model danych . . . 365
22.4.3 Relacyjny model danych . . . 365
22.4.4 Szybki SQL . . . 366
22.4.5 Skrypt SQL do tworzenia tabel . . . 367
22.4.6 Konstruktor zapytań w APEX . . . 367
22.5 Interfejsy aplikacji . . . 368
22.5.1 Interfejsy aplikacji do zarządzania . . . 368
22.5.2 Interfejsy aplikacji biur studenckich . . . 369
22.5.3 Interfejsy aplikacji studenckich . . . 370
22.5.4 Interfejsy aplikacji dla nauczycieli . . . 373
22.6 Dodatkowe materiały do nauki . . . 375
22.6.1 Eksportowana aplikacja . . . 375
22.6.2 Przewodniki wideo . . . 378
22.7 Pytania . . . 378
22.8 Odpowiedzi . . . 378

23 Sprawa Telco . . . 379
Veronika Šalgová, Jozef Kostolný, Michal Mrena, Michal Kvet, Miroslav Potočár . . . .
23.1 Biznesowy pogląd na sprawę . . . 379
23.2 Definicja problemu . . . 379
23.3 Przypadki użycia . . . 379
23.3.1 Opis narracyjny . . . 380
23.3.2 Opis częściowo ustrukturyzowany . . . 380
23.3.3 Diagram przypadków użycia . . . 380
23.4 Model danych . . . 382
23.4.1 Opis modelu danych . . . 382
23.4.2 Logiczny model danych . . . 382
23.4.3 Relacyjny model danych . . . 382
23.5 Uwierzytelnianie użytkowników i role użytkowników . . . 383
23.6 Interfejsy aplikacji . . . 386
23.6.1 Projektowanie aplikacji . . . 386
23.7 Skrypty . . . 394
23.8 Tworzenie strony głównej . . . 395
23.9 Tworzenie strony klienta . . . 397
23.10 Tworzenie strony menedżera . . . 399
23.11 Dodatkowe materiały edukacyjne . . . 400
23.11.1 Wyeksportowana aplikacja . . . 400
23.11.2 Przewodniki wideo . . . 400
23.12 Pytania . . . 400
23.13 Odpowiedzi . . . 400

24 Sprawa wynajmu samochodu . . . 403
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . . .
24.1 Biznesowe spojrzenie na sprawę . . . 403
24.2 Definicja problemu . . . 403
24.3 Przypadki użycia . . . 404
24.3.1 Narracyjny opis przypadku użycia . . . 404
24.3.2 Opis częściowo ustrukturyzowany . . . 404
24.3.3 Diagram przypadków użycia . . . 404
24.4 Model danych . . . 406
24.4.1 Narracyjny opis modelu danych . . . 406
24.4.2 Logiczny model danych . . . 407
24.4.3 Relacyjny model danych . . . 407
24.4.4 Skrypt SQL . . . 408
24.5 Interfejsy aplikacji . . . 409
24.6 Dodatkowe materiały edukacyjne . . . 414
24.6.1 Eksportowane aplikacje . . . 414
24.6.2 Przewodniki wideo . . . 414
24.7 Pytania . . . 414
24.8 Odpowiedzi . . . 414

Bibliography . . . 415
Articles . . . 415
Books . . . 415

Index . . . 417

Spis rysunków

1.1 Cykl tworzenia aplikacji. 37
1.2 Business Model Canvas (6) 38
1.3 Podstawowe pojęcia diagramu przypadków użycia. 39
1.4 Generalizacja/specjalizacja "Aktorów" 40
1.5 Granice systemu i stowarzyszeń "uwzględniają"i żozszerzają". 40
1.6 Schemat przypadków użycia systemu rezerwacji hotelowej. 41
1.7 Szukanie APEX docker (kluczy dostępu APEX). 45

2.1 DBS jako podstawa aplikacji internetowych. 49
2.2 Logiczny model dla Zasobów Ludzkich HR przy użyciu ER-Diagram. 51
2.3 Zarządzanie modelem logicznym za pomocą ODM. 52
2.4 Zarządzanie atrybutami za pomocą ODM. 53
2.5 Zarządzanie relacjami przy pomocy ODM. 54
2.6 Schematy RDG i schematy tabeli. 55
2.7 Proces generownia schematu RDG z modelu logicznego. 56
2.8 Automatycznie generowany schemat RDB. 56
2.9 Zarządzanie kolumnami poprzez elementy GUI. 58
2.10 GUI-Elements for Managing the RM. 58
2.11 Etapy normalizacji - przegląd. 59
2.12 Kategorie instrukcji SQL. 61
2.13 Eksport schematu RDB do skryptu DDL 62
2.14 Przesyłanie i wykonywanie skryptu SQL. 63
2.15 Dostęp do Quick SQL poprzez SQL Workshop 64
2.16 Wygenerowany skrypt SQL oparty na Quick SQL. 65
2.17 Edytor poleceń SQL. 66
2.18 Instrukcja SQL tworzenia tabeli `Departaments`. 66
2.19 Polecenie SQL opuszczania tabeli `Departments`. 67
2.20 Instrukcja SQL dla zmiany tabeli `Departments` 67
2.21 Zarządzanie tabelami za pomocą przeglądarki obiektów ORACLE. 67
2.22 Instrukcja SQL dla wstawiania nowych danych do tabeli `Departments`. 68
2.23 Zapytanie SQL dotyczące aktualizacji wynagrodzenia wszystkich pracowników o 10% w tabeli `Employees`. 68

2.24 Instrukcja SQL dla aktualizacji wynagrodzenia pracownika "Miller", tylko i wyłącznie. 68
2.25 Instrukcja SQL dla aktualizacji więcej niż jednego atrybutu w tabeli Employees. 69
2.26 Instrukcja SQL usuwająca wszystkie dane z tabeli Employees. 69
2.27 Instrukcja SQL do usuwania niektórych danych z tabeli Employees. 69
2.28 Losowe wstawianie danych za pomocą Quick SQL. 70
2.29 Instrukcja SQL pobierająca wszystkie dane z tabeli Employees. 70
2.30 Fragment artykułu Pobieranie wszystkich danych z tabeli EMPLOYEES. 71
2.31 Instrukcja SQL pobierająca wszystkie dane z tabeli Employees w kolejności po ich Salary. 71
2.32 Fragment z Pobieranie wszystkich danych z tabeli Employees uporządkowanych według ich Salary. 71
2.33 Polecenie SQL do pobierania określonych danych z tabeli - Employees, tylko i wyłacznie. 72
2.34 Fragment artykułu Pobieranie określonych danych tylko z tabeli Employees. . . . 72
2.35 Instrukcja SQL pobierania danych z tabeli Employees przy użyciu funkcji TO_CHAR(). 72
2.36 Fragment z Pobieranie danych z tabeli Employees przy użyciu funkcji TO_CHAR(). 73
2.37 Instrukcja SQL łącząca dane z tabeli Employees i tabeli Departments. 73
2.38 Fragment artykułu Pobieranie określonych danych z tabeli Employees i tabeli Departments. 74
2.39 Instrukcja SQL zliczająca liczbę pracowników i tworząca sumę ich wynagrodzeń z tabeli Employees. 74
2.40 Fragment artykułu Pobieranie liczby pracowników i tworzenie sumy ich wynagrodzeń z tabeli Employees. 74
2.41 Instrukcja SQL grupująca dane według wynagrodzenia i porządkująca wynik według liczby pracowników z tym wynagrodzeniem z tabeli Employees. 75
2.42 Fragment Grupowanie danych według wynagrodzenia i porządkowanie wyników według liczby pracowników z tym wynagrodzeniem z tabeli Employees 75
2.43 Przegląd narzędzia ORACLE Query Builder ("QB"). 76
2.44 Używanie Query Builder do tworzenia Join. 77
2.45 Przegląd opcji zarządzania warstwą DB. 78

3.1 Ogólny proces rozwoju w APEX. 82
3.2 Oracle APEX Workspace Homepage. 83
3.3 Przegląd aplikacji App Builder. 84
3.4 Kreator tworzenia aplikacji: Trzy przypadki użycia podczas tworzenia aplikacji. . 85
3.5 Właściwości i strony. 86
3.6 Funkcje i ustawienia. 86
3.7 Strona główna aplikacji - widok dewelopera. 87
3.8 Tworzenie strony. 88
3.9 Kreator strony (Page Designer 89
3.10 Uruchamianie całej aplikacji. 90
3.11 Uruchamianie poszczególnych stron. 91

4.1 Opcje wymiany danych w APEX. 94
4.2 Używanie Data Workshop do ładowania danych - dostęp przez SQL Workshop. 95
4.3 c 96
4.4 Źródło danych. 97
4.5 Wczytaj dane do istniejącej lub nowej tabeli. 97
4.6 Ładowanie danych do istniejącej tabeli. 99
4.7 Eksportowanie ("rozładowywanie") danych. 100

4.8 Kreator rozładowywania danych "Unload Data Wizard". 101
4.9 Import/eksport danych tabeli przy użyciu przeglądarki obiektów. 102
4.10 Eksport danych wyniku zapytania SQL. 103
4.11 Eksport danych z raportu aplikacji. 104
4.12 Usługi RESTful do wymiany danych - podstawowa architektura. 105
4.13 Włączanie schematu DB dla dostępu RESTful. 107
4.14 Relacje między różnymi komponentami usług ORDS RESTful Services. 108
4.15 Przykładowe usługi RESTful dla tabeli Pracownicy. 109
4.16 Włączanie usługi REST w przeglądarce obiektów. 110

5.1 Proszę wpisać link APEX i zalogować się. 114
5.2 Proszę poprosić o Oracle APEX Workspace początkowe kroki. 114
5.3 Proszę poprosić o ukończenie Oracle APEX Workspace. 115
5.4 Oracle APEX Workspace zatwierdzony. 115
5.5 E-mail aktywacyjny od APEX. 116
5.6 Proszę ustawić nowe hasło dla Workspace APEX. 116
5.7 Workspace APEX environment. 117
5.8 Model logiczny naszego przykładu działającego HR. 118
5.9 Model relacyjny naszego przykładowego działającego HR. 120
5.10 Wstawianie kodu Quick SQL do obszaru roboczego APEX. 122
5.11 Uruchom skrypt SQL 124
5.12 Tworzenie procesu uruchamiania aplikacji. 124
5.13 Tworzenie aplikacji ze skryptu. 125
5.14 Utworzona aplikacja internetowa. 126
5.15 Środowisko tworzenia aplikacji. 126
5.16 Strona logowania do aplikacji. 127
5.17 Nasze nowe środowisko aplikacji. 127
5.18 Zarządzanie środowiskiem użytkowników i grup. 128
5.19 Przykład kontroli dostępu. 129
5.20 Dodawanie nowych użytkowników. 130
5.21 Dodawanie ról do użytkowników. 131
5.22 Tworzenie konfiguracji przycisków. 132
5.23 Dostęp i prawa współpracowników. 133
5.24 Dostęp i prawa czytelników. 134

6.1 Wybór typu strony. 138
6.2 Tworzenie strony z raportem. 139
6.3 Renderowanie. 139
6.4 Właściwości regionu. 140
6.5 Różne źródła danych. 140
6.6 Układ. 141
6.7 Wygląd. 141
6.8 Szablony. 142
6.9 Zaawansowane. 142
6.10 Nagłówek i stopka. 142
6.11 Stan po stronie serwera. 143
6.12 Tylko do odczytu. 143
6.13 Bezpieczeństwo, pamięć podręczna serwera i personalizacja. 144
6.14 Paginacja. 144

6.15 Liczba wierszy do załadowania. 144
6.16 Pobieranie i drukowanie. 145
6.17 Środek projektanta stron. 145
6.18 Typ kolumny. 146
6.19 Ustawienia kolumny. 146
6.20 Format maski. 146
6.21 Personalizacja. 147
6.22 Opcja sortowania według kolumny. 147
6.23 Lokalizacja zielonej strzałki dla prezentacji strony. 147
6.24 Klasyczny przykład raportu. 148
6.25 Możliwość zmiany widoczności kolumny w Column Toggle Report. 148
6.26 Raport interaktywny. 149
6.27 Panel wyszukiwania. 149
6.28 Wyszukiwanie. 150
6.29 Przycisk akcji. 150
6.30 Wybór kolumn do wyświetlenia. 151
6.31 Filtr kolumn. 151
6.32 Filtr wierszy. 152
6.33 Po kliknięciu przycisku Dane. 152
6.34 Sortowanie. 153
6.35 Agregowanie. 153
6.36 Flashback. 153
6.37 Control Break. 154
6.38 Wynik Control Break. 154
6.39 Highlight. 154
6.40 Najważniejszy wynik. 155
6.41 Wiersz na stronę. 155
6.42 Wykres w raporcie interaktywnym. 155
6.43 Grupuj. 156
6.44 Zapisz raport. 156
6.45 Pobierz. 157

7.1 Interaktywna siatka z klientam 159
7.2 Działania interaktywnej siatki 160
7.3 Ukryj ikonę kolumny Telefon 160
7.4 Wyświetlane kolumny 160
7.5 Ikona zamrożenia kolumny Adres 161
7.6 Sortowanie kolumny Adres 161
7.7 Ikona zagregowana kolumny Adres 161
7.8 Rodzaje wykresów 162
7.9 Formularz dodawania klientów 162
7.10 Skumulowany szczegółowy formularz zryczałtowanych stawek klienta 163
7.11 Obok siebie główny formularz szczegółowy zryczałtowanych stawek klienta . 163
7.12 Formularz Drill Down master detail – pierwsza strona 164
7.13 Formularz Drill Down master detail – druga strona 165

8.1 Model danych używany w przykładowej aplikacji. 167
8.2 Wybieranie i umieszczanie regionu wykresu. 168
8.3 Układ sekcji głównej po umieszczeniu regionu wykresu. 168

8.4 Przegląd drzewa pokazujący błąd w serii wykresów. 169
8.5 Końcowa konfiguracja serii sprzedaży 170
8.6 Dane zwrócone przez wykonanie zapytania SQL. 170
8.7 Wykres wyświetlany podczas uruchamiania aplikacji. 171
8.8 Zmiana atrybutu tytułu dla osi x. 171
8.9 Wykres sprzedaży po oznaczeniu osi. 172
8.10 Wybieranie i umieszczanie elementu Checkbox Group. 172
8.11 Ustawienie podstawowych atrybutów elementu Checkbox Group. 173
8.12 Ustawienie listy wartości dla elementu Checkbox Group. 173
8.13 Ustawianie wartości domyślnych dla elementu Checkbox Group. 174
8.14 Powiązanie regionu wykresu sprzedaży z elementem grupy pól wyboru 175
8.15 Tworzenie akcji dynamicznej dla elementu P1_PRODUCTS_CHOICE. 175
8.16 Ustawianie podstawowych atrybutów akcji dynamicznej przy zmianie. 176
8.17 Ustawianie atrybutów akcji Refresh 176
8.18 Dynamiczna akcja odświeżająca wykres sprzedaży w przeglądzie drzewa. . . 177
8.19 Aplikacja po wdrożeniu filtrowania produktów. 177
8.20 Wybieranie i umieszczanie elementu listy wyboru. 178
8.21 Region wykresu sprzedaży po umieszczeniu i zmianie nazwy elementu listy wyboru. 178
8.22 Ustawianie listy wartości dla pozycji Select List. 178
8.23 Ustawianie wartości statycznych dla elementu Select List. 179
8.24 Dodawanie odświeżania akcji dynamicznej dla elementu Select List. 179
8.25 Ustawianie kolejności według pozycji dla regionu wykresu sprzedaży. 179
8.26 Ustawienie klauzuli Order By dla elementu Select List używanego do sortowania. 180
8.27 Aplikacja po wdrożeniu sortowania wykresów. 180
8.28 Zmiana typu wykresu na Liniowy z obszarem. 181
8.29 Lista wartości dla elementu Select List używanego do filtrowania sprzedaży produktów według roku. 181
8.30 Ustawienie źródłowego zapytania SQL i mapowania kolumn dla regionu wykresu miesięcznej sprzedaży. 182
8.31 Miesięczny wykres sprzedaży. 182

9.1 Rozszerzone boczne menu nawigacyjne 185
9.2 Górne zakładki nawigacji 186
9.3 Górne menu nawigacyjne 186
9.4 Mega menu górnej nawigacji 186
9.5 Atrybuty menu nawigacji 187

10.1 Funkcja blokowania stron w APEX. 191
10.2 Blokowanie określonej strony. 192
10.3 Dodawanie komentarzy. 192
10.4 Opcje budowania. 193
10.5 Funkcja "Team Development" 194
10.6 Aktywacja funkcji "Team Development". 194
10.7 Dostosowanie funkcji rozwoju zespołu. 195
10.8 Komunikacja związana z problemem w ramach funkcji "Team Development". 195
10.9 Zezwolenie na informacje zwrotne w "Application Definition". 196
10.10 Przesyłanie informacji zwrotnych w aplikacji. 197
10.11 Menu "Utilities"(Narzedzia). 198

11.1 Obszary robocze Oracle APEX są dostarczane z przykładowymi aplikacjami i aplikacjami startowymi. 201
11.2 Aplikacje startowe w Galerii. 202
11.3 Instalacja aplikacji startowej Team Calender. 203
11.4 Przykładowe aplikacje. 204

12.1 Kreator importu i eksportu aplikacji. 210
12.2 Przekształcanie Quick SQL do poleceń SQL. 212
12.3 Skrypt CH12INSERT wstawia dane do trzech tabel i zatwierdza transakcje. . . . 212
12.4 Tworzenie aplikacji ze skryptu. 213
12.5 Wybieranie nazwy aplikacji i wszystkich funkcji. 213
12.6 Wybieranie "Shared components"(składników współużytkowanych). 214
12.7 Tworzenie listy wartości od podstaw. 214
12.8 Definiowanie nazwy i typu: CH12_LOV_COMPETENCE_DESCRIPTION jest dynamiczne. 215
12.9 Wprowadzanie polecenia SQL SELECT. 215
12.10 Zdefiniuj nazwę i typ: CH12_LOV_COMPETENCE_LEVEL jest statyczny. 216
12.11 Zdefiniuj wartości wyświetlane i zwracane: CH12_LOV_COMPETENCE_LEVEL jest statyczny. 216
12.12 Wybierz stronę 7 (Ch12 Job Competences). 217
12.13 Raport strona 7(Ch12 Job Competences). 217
12.14 Wybieranie obiektów pomocniczych. 218
12.15 Ustawianie wymagań wstępnych. 218
12.16 Sprawdzenie istnienia trzech tabel. 218
12.17 Ustaw prompt do zmiany nazwy aplikacji. 219
12.18 Ustawianie skryptów instalacyjnych. 219
12.19 Ustawianie skryptów deinstalacyjnych. 219
12.20 Importowanie aplikacji w innym obszarze roboczym. 220
12.21 Umowa licencyjna. 220
12.22 Zmiana nazwy importowanej aplikacji. 221
12.23 Dodawanie roli do użytkownika. 221
12.24 Określone języki do tłumaczenia. 222
12.25 Tekst do tłumaczenia. 222
12.26 Eksportowanie ciągów dla konkretnego języka i strony. 223
12.27 Tłumaczenie ciągów znaków oznaczonych jako "target"w liniach 48, 52, 56, 60, 72 i 76. 223
12.28 Wgrywanie plików tłumaczenia XLDIFF. 223
12.29 Zastosowanie zmian i publikacja 224
12.30 Ostateczna publikacja i tłumaczenie aplikacji. 224
12.31 Ustawienie głównego języka aplikacji. 224
12.32 Przetłumaczona strona. 224

13.1 Diagram przypadków użycia. 234
13.2 Logiczny model danych. 235
13.3 Relacyjny model danych. 235
13.4 Pulpit nawigacyjny współautora (wydawcy). 236
13.5 Publikowanie wiadomości z załącznikami. 236
13.6 Pulpit nawigacyjny czytelnika. 237
13.7 Dostęp przeglądarki (czytnika) do wiadomości. 237
13.8 Pulpit nawigacyjny administratora. 237
13.9 Edytowanie roli pracownika przez administratora. 238

14.1 Schemat przypadków użycia - "Catalog of plants" 244
14.2 Logiczny model danych. 245
14.3 Relacyjny model danych. 246
14.4 Interaktywna siatka dla formularza rośliny. 247
14.5 Część publiczna – otwarte przeglądanie danych roślin. 247

15.1 Diagram przypadków użycia - Autoryzacja i Zarządzanie Użytkownikami GreenDi 252
15.2 Logiczny model danych. 253
15.3 Relacyjny model danych. 253
15.4 Formularz użytkownika. 254
15.5 Formularz historii użytkownika. 254

16.1 Diagram przypadków użycia. 258
16.2 Logiczny model danych. 262
16.3 Relacyjny model danych. 263
16.4 Pulpit nawigacyjny pracownika. 264
16.5 Przesyłanie pomysłu z załącznikami. 264
16.6 Struktura organizacyjna w firmie. 265
16.7 Przegląd pomysłów na małe innowacje. 265
16.8 Pulpit nawigacyjny recenzenta. 266
16.9 Strona do przeglądania pomysłów. 266

17.1 Integracja przepływów dla APEX z aplikacją APEX. 271
17.2 Diagram przypadków użycia. 276
17.3 Workflow - przetwarzanie zapytania - diagram BPMN. 277
17.4 Tworzenie instancji. 277
17.5 Tworzenie instancji przepływu CH17. 278
17.6 Uruchom instancję demonstracyjną. 278
17.7 Wykonaj pierwsze zadanie w przykładowej instancji. 278
17.8 Wykonywanie zadań w bramie równoległej. 279
17.9 Ukończona instancja demo. 279
17.10 Logiczny model danych. 280
17.11 Relacyjny model danych. 280
17.12 Strona główna aplikacji ĆH17 Business Process Management". 283
17.13 Sprzedaż - proces zainicjowany. 285
17.14 Raport przepływu sprzedaży. 285
17.15 Raport Sales - flow dla wybranej instancji. 286
17.16 Raport Sales - flow dla wybranej instancji. 286
17.17 Sprzedaż - raport z zapytania. 287
17.18 Sprzedaż - lista dokumentów. 287
17.19 Sprzedaż - przesłanie dokumentu do zapytania. 288
17.20 Sprzedaż - lista dokumentów po przesłaniu i odświeżeniu. 288
17.21 Sprzedaż - raport dla wybranej instancji po przesłaniu trzech dokumentów. . 289
17.22 Sales - raport przepływu dla wybranej instancji po przesłaniu trzech dokumentów. 289
17.23 Raport przepływu produkcji - produkcja. 290
17.24 Produkcja - diagram BPMN - stan instancji. 290
17.25 Ocena produkcji. 290
17.26 Ocena produkcji zakończona, instancja oczekująca na ocenę finansową. . 291

17.27 Sprawozdanie finansowe - przepływ. 291
17.28 Finanse - diagram BPMN - stan instancji. 291
17.29 Ocena finansowa. 292
17.30 Ocena finansowa zakończona, instancja oczekuje na ocenę biznesową. . . 292
17.31 Finanse - diagram BPMN - stan instancji. 293
17.32 Raport przepływu - Biznes. 293
17.33 Ocena biznesowa. 293
17.34 Ocena biznesu zakończona, instancja zakończona. 294
17.35 Pokazanie stanu instancji na diagramie BPMN. 295
17.36 Elementy strony "Seting" 295
17.37 Zakończenie etapu w Flows for Apex. 296
17.38 Zdefiniuj role i role użytkowników w menu "Application Access Control". . . . 296
17.39 Przyznawanie "uprawnień do tworzenia zadań"do obszaru roboczego. 297
17.40 Importowanie przepływu pracy o nazwie CH17 do Flows for APEX - krok 1. . . 298
17.41 Importowanie przepływu pracy o nazwie CH17 do Flows for APEX - krok 2. . . 298

18.1 Diagram przypadków użycia - Wymiana roślin i nasion 302
18.2 Logiczny model danych. 304
18.3 Relacyjny model danych. 305
18.4 Oferty - raport interaktywny. 305
18.5 Formularz wiadomości. 306

19.1 Diagramy przypadków użycia. 312
19.2 Logiczny model danych. 313
19.3 Relacyjny model danych. 313
19.4 Generowanie aplikacji ze skryptu CH19CREATEINSERT - część 1. 316
19.5 Generowanie aplikacji ze skryptu CH19CREATEINSERT - część 2. 317
19.6 Dodawanie książki przez administratora. 318
19.7 Dodawanie kategorii przez administratora. 318
19.8 Rejestracja użytkownika. 319
19.9 Przeglądanie i dodawanie recenzji. 319
19.10 Formularz do komentowania recenzji 320

20.1 Use case diagram. 327
20.2 Logiczny model danych. 328
20.3 Definicja unikalnego indeksu w Oracle SQL Data Modeler. 329
20.4 Relational data model. 329
20.5 Generowanie skryptu SQL za pomocą narzędzia Quick SQL. 331
20.6 Strona główna aplikacji. 334
20.7 Zarządzanie podstawowymi danymi - raport. 334
20.8 Zarządzanie podstawowymi danymi - formularz. 334
20.9 Zarządzanie danymi struktury - raport. 335
20.10 Zarządzanie danymi struktury - formularz. 335
20.11 Strona do obliczania zestawienia materiałowego. 336

21.1 Diagram przypadków użycia. 340
21.2 Logiczny model danych. 342
21.3 Relacyjny model danych. 343
21.4 Szybkie narzędzie SQL do tworzenia skryptu SQL. 344
21.5 Procedura skryptu SQL do zbudowania SEQUENCE. 346

21.6 Utwórz nową aplikację. 347
21.7 Zastosowanie żywieniowe. 348
21.8 Aplikacja żywieniowa - strona przepisu. 349
21.9 Strona aplikacji żywieniowych - składniki. 349
21.10 Aplikacja żywieniowa - strona składników receptury. 350
21.11 Aplikacja żywieniowa - strona z informacjami o żywieniu. 350
21.12 Tworzenie strony pulpitu nawigacyjnego dla aplikacji. 351
21.13 Skonfiguruj tytuł i typ czatu w raportach pulpitu nawigacyjnego. 352
21.14 Konfigurowanie SOURCE za pomocą SQL QUERY. 354
21.15 Tworzenie nowego konta użytkownika. 355
21.16 Utwórz konto użytkownika w aplikacji. 356
21.17 Konfigurowanie stron z prawami użytkownika. 357

22.1 Diagram przypadków użycia. 364
22.2 Logiczny model danych. 365
22.3 Relacyjny model danych. 366
22.4 Model danych opisany za pomocą Quick SQL. 366
22.5 Przykładowe użycie konstruktora zapytań. 367
22.6 Strona główna aplikacji. 368
22.7 Pulpit nawigacyjny zarządzania. 369
22.8 Interaktywny raport godzin pracy dla biura studenta. 369
22.9 Formularz zmiany harmonogramu dla biura studenta. 370
22.10 Interaktywny raport godzin pracy po zmianie harmonogramu. 370
22.11 Kalendarz spotkań studenckich. 371
22.12 Kalendarz wszystkich godzin dyżurowych oferowanych przez nauczycieli w programie studiów studenta. 371
22.13 Rejestracja do godzin pracy biura za pośrednictwem kalendarza - wybieranie gniazda kalendarza. 371
22.14 Rejestracja do godzin pracy biura poprzez kalendarz - wybór celu. 372
22.15 Rejestracja do godzin pracy biura za pośrednictwem kalendarza - pomyślna rejestracja. 372
22.16 Sprawdzanie rejestracji za pomocą pozycji menu "Viev my appointments". . 372
22.17 Szczegółowy interaktywny raport ze wszystkich spotkań studentów. 373
22.18 Szczegółowy raport z nominacji nauczycieli. 373
22.19 Kalendarz nauczyciela z widocznymi imionami uczniów, ich językami i celami. 374
22.20 Zmiana harmonogramu godzin pracy nauczyciela z NONE enrolled - widok kalendarza. 374
22.21 Zmiana harmonogramu godzin pracy nauczyciela z NONE zapisanym - wprowadzono nową datę. 375
22.22 Wyniki przeplanowania godzin dyżuru nauczyciela, gdy nikt nie jest zapisany. 375

23.1 Diagram przypadków użycia. 381
23.2 Logiczny model danych. 382
23.3 Relacyjny model danych. 383
23.4 Pulpit nawigacyjny klienta. 387
23.5 Pulpit klienta - okno modalne faktury. 388
23.6 Pulpit klienta - faktura w formacie PDF. 388
23.7 Pulpit klienta - statystyki minut. 388
23.8 Pulpit klienta - Statystyki SMS. 389
23.9 Pulpit nawigacyjny klienta - statystyki danych. 389

23.10 Manager dashboard – Add customer. 390
23.11 Manager dashboard – Manage customer. 391
23.12 Manager dashboard – View customers. 391
23.13 Manager dashboard – Customer export. 392
23.14 Manager dashboard - Customer export in XLS file. 392
23.15 Pulpit nawigacyjny administratora. 393
23.16 Strona docelowa z logowaniem. 393
23.17 Strona logowania. 393
23.18 Regiony ciała. 396
23.19 Lista treści. 396
23.20 Kod HTML regionu statycznego. 396

24.1 Diagram przypadków użycia. 406
24.2 Model logiczny projektu wypożyczalni samochodów 407
24.3 Model relacyjny projektu wynajmu samochodów 408
24.4 Zaloguj się do aplikacji. 410
24.5 Strona główna aplikacji. 410
24.6 Szablon samochodu 411
24.7 Jak stworzyć kolumnę ze zdjęciami. 412
24.8 Dane klientów. 413
24.9 Szablon wypożyczalni samochodów. 413



Spis tablic

13.1 Opis przypadku użycia: publikowanie wiadomości internetowych. 231
13.2 Opis przypadku użycia: czytanie wiadomości intranetowych 232
13.3 Opis przypadku użycia: Zarządzanie portalem intranetowym 233

14.1 Opis przypadku użycia: przeglądanie katalogu roślin 243

15.1 Opis przypadku użycia: autoryzacja i zarządzanie użytkownikami. 251

16.1 Opis przypadku użycia: Zgłoszenie pomysłu. 259
16.2 Opis przypadku użycia: Przegląd pomysłów. 260
16.3 Opis przypadku użycia: Ocena pomysłu. 261

17.1 Opis przypadku użycia: przygotowanie dokumentacji zapytania. 274
17.2 Opis przypadku użycia: ocena aspektów zapytania dotyczących produkcji . 275
17.3 Opis przypadku użycia: ocena finansowych aspektów dochodzenia. 275
17.4 Opis przypadku użycia: ocena biznesowych aspektów zapytania 276

18.1 Opis przypadku użycia: Wymiana roślin i nasion. 303

19.1 Opis przypadku użycia: system zarządzania recenzjami książek 311

20.1 Opis przypadku użycia: raportowanie i przechowywanie podstawowych danych. 324
20.2 Opis przypadku użycia: raportowanie i utrzymywanie danych dotyczących konstrukcji. 325
20.3 Opis przypadku użycia: oblicz BOM. 326

21.1 Opis przypadku użycia: Dostęp do receptur i składników oraz obliczanie wartości odżywczych 341

22.1 Opis przypadku użycia: zmiana harmonogramu godzin pracy przez nauczyciela. 362
22.2 Opis przypadku użycia: rejestracja studentów na godziny pracy. 363

23.1 Opis przypadku użycia: dodaj usługę. 380
23.2 Opis przypadku użycia: pokaż stan usługi. 381
23.3 Przykładowe dane przechowywane w tabeli CH23_Person 384

24.1 Opis przypadku użycia: dostęp do samochodów, klientów i rezerwacji wynajmu samochodów 405

Podziękowania

Partnerzy projektu i zespół:

- Uniwersytet w Mariborze Wydział Nauk Organizacyjnych,
- Uniwersytet w Zagrzebiu Wydział Organizacji i Informatyki,
- Uniwersytet w Żylinie,
- Uniwersytet Kozmińskiego,
- Międzynarodowy Uniwersytet Helleński oraz i
- Uniwersytet Johanna Keplera

chcieliby podziękować za wsparcie finansowe udzielone przez European Commission Erasmus + Better Employability for Everyone with APEX (ID projektu 2021-1-SI01-KA220-HED- 000032218), współfinansowane przez program Erasmus+ Unii Europejskiej.

Wsparcie Komisji Europejskiej dla produkcji tej publikacji nie stanowi jej poparcia dla zawartości, która odzwierciedla jedynie poglądy autorów, a Komisja nie może być odpowiedzialna za jakiekolwiek użycie informacji w niej zawartych.

Wielkie podziękowania dla:

- Oracle Academy *Menadżera Programu Oracle Academy, pana Darko Jurekovicia* za jego ciągłe wsparcie w rozpowszechnianiu wyników projektu oraz i
- THE RIGHT THING SOLUTIONS *pana Aljaža Maliego, CEO* za jego cenne rady dotyczące APEX przed rozpoczęciem projektu i w trakcie jego trwania.

Zespół projektowy BeeAPEX



Wprowadzenie

Witamy w systemie Oracle Application Express (APEX) - intuicyjnej i wydajnej platformie do tworzenia aplikacji internetowych niewymagających umiejętności programowania. Ten podręcznik ma na celu wyposażenie Ciebie w kompetencje niezbędne do pełnego wykorzystania potencjału platformy Oracle APEX oraz stworzenie nowoczesnych aplikacji, które pozwolą Ci stawić czoła rzeczywistym wyzwaniom biznesowym.

Część I podręcznika pt. “Jak”, poświęcona jest fundamentalnym aspektom Oracle APEX. W tych dwunastu rozdziałach zapoznasz się z najważniejszymi koncepcjami, uzyskasz dostęp do platformy APEX i poznasz różne funkcje potrzebne do tworzenia rozbudowanych aplikacji. Każdy rozdział dotyczy konkretnego tematu, zawierającego jasne instrukcje i ułatwiającego praktyczne uczenie się.

W rozdziale 1: “Jak uruchomić Oracle APEX?” wyjaśniono co to jest Oracle APEX oraz opisano różne sposoby przygotowania środowiska APEX do praktycznego uczenia się i tworzenia aplikacji.

Rozdział 2: “Jak przygotować bazę danych?” zawiera wprowadzenie do modelowania danych, zarządzania bazą danych, przygotowywania danych i kierowania do nich zapytań. Dla początkujących niezbędny jest szersze spojrzenie na koncepcję bazy danych.

Rozdział 3: “Jak poruszać się po APEX?” zawiera prezentację funkcjonalności środowiska APEX, które umożliwiają tworzenie, generowanie i dostosowywanie różnych stron internetowych i komponentów. Uruchomienie i testowanie aplikacji APEX jest zaledwie jednym kliknięciem od środowiska deweloperskiego.

Rozdział 4: “Jak wymieniać dane w APEX?” umożliwia wgląd w import i eksport danych w APEX. Rozdział ten obejmuje wymianę danych z plikami, takimi jak arkusze kalkulacyjne, a także za pomocą usług RESTful.

Rozdział 5: “Jak wygenerować pierwszy projekt aplikacji?” zachęca do wypróbowania siły rozwojowej APEX. W momencie, gdy zdecydujesz, jakie są dane, możesz natychmiast wygenerować atrakcyjną i funkcjonalną aplikację bez programowania. Można też od razu wygenerować podstawową kontrolę dostępu dla użytkowników końcowych o różnych rolach.

Rozdział 6: “Jak zarządzać raportami?” prowadzi Cię przez udostępnianie widoków danych za pomocą klasycznych i interaktywnych kreatorów raportowania. Raporty APEX zawierają również funkcje dostosowywania raportów przez użytkowników końcowych bez konieczności programowania.

Rozdział 7: “Jak zarządzać formularzami?” przedstawia trzy typowe rodzaje formularzy

internetowych, w tym master-detail. Możesz również dostosowywać i generować strony formularzy bez umiejętności programowania.

Rozdział 8: “Jak przekształcić raporty tekstowe w wykresy?” przeprowadzi Cię przez drogę do wykorzystania funkcji APEX do prezentowania danych, jako wykresów bezpośrednio obok raportów tekstowych.

Rozdział 9: “Jak zarządzać menu?” przedstawia różne elementy nawigacyjne dla aplikacji APEX.

Rozdział 10: “Jak współpracować w zespole?” oferuje wgląd w funkcje APEX służące zespołom, ponieważ rzadko zdarza się, aby rozwój aplikacji angażował tylko jednego programistę.

Rozdział 11: "Jak skorzystać z galerii przykładowych aplikacji i wtyczek?żachęca do zastosowania zaawansowanych możliwości APEX poprzez ponowne wykorzystanie dobrych wzorców.

Rozdział 12: “Jak zarządzać pakietowymi i wielojęzycznymi aplikacjami?” pokazuje, jak dystrybuować Twoją aplikację do innych środowisk APEX dla użytkowników mówiących w różnych językach.

Część I obejmuje również tematy, które są niezbędne dla bezpieczeństwa aplikacji, strategii wdrażania i gotowości do zastosowania w świecie realnym.

Część II tego podręcznika wykracza poza podstawy, prezentując dwanaście angażujących przypadków biznesowych od opisu, przez każdy szczegół techniczny, po rozwiązanie problemu. Każdy przypadek jest starannie udokumentowany, tak aby uzyskać wszelkie informacje na temat tworzenia aplikacji z perspektywy firmy, danych i interfejsu użytkownika. Ta część obejmuje aplikacje dla przedsiębiorstw takich jak:

- wiadomości intranetowe dla pracowników,
- mały system innowacji,
- zarządzanie procesami biznesowymi przy pomocy workflow’ów,
- kalkulacja struktury materiałowej,
- system recenzji książek,
- zarządzanie żywieniem i dietą,
- planowanie godzin pracy,
- rozliczenia w firmie telekomunikacyjnej,
- wypożyczalnia samochodów

dla społeczności:

- katalog roślin,
- wymiana roślin

oraz ogólnie stosowalna autoryzacja i zarządzanie użytkownikami. W każdym przykładzie biznesowym znajdziesz:

- Biznesowy punkt widzenia: Przegląd sytuacji biznesowej.
- Definicję problemu: Poszukiwanie odpowiedzi, kto i dlaczego ma problem.
- Opisane przykłady: Prezentowane są trzy typy opisu: narracyjny, semistrukturalny i graficzny do przygotowania dokumentacji przypadku użycia UML.
- Model danych logicznych i relacyjnych: W pełni funkcjonalna platforma APEX umożliwia uruchamianie nowych struktur danych, wykorzystywanie i modyfikowanie istniejących, łączenie z innymi narzędziami w celu modelowania danych oraz wspomaganie inżynierii w przód lub wstecznej. Programiści starają się znaleźć relację między różnymi potrzebami biznesowymi, a odpowiednimi fragmentami danych, co jest podstawą do tworzenia interfejsu użytkownika.
- Interfejsy aplikacji: podręcznik dostarcza strony HTML, raporty, formularze, pola, menu, przyciski i hiperłącza, które materializują sytuację biznesową, rozwiązanie problemu biznesowego oraz use case’y, czy dane z myślą o użytkowniku końcowym.
- Dodatkowe materiały szkoleniowe: w celu wzbogacenia, przyspieszenia i pomocy w tej

ścieżce rozwoju, w każdym rozdziale znajdziesz łącza do wyeksportowanych aplikacji, skryptów, samouczków danych i wideo. Zasoby te dostarczą Ci praktyczną wiedzę, co pozwoli Ci wzmocnić swoją wiedzę i zastosować ją bezpośrednio do rzeczywistych projektów.

Niezależnie od tego, czy jesteś doświadczonym programistą chcącym poszerzyć swój zestaw umiejętności, czy początkującym chętnym do poznania świata APEX, ten podręcznik jest Twoim najlepszym przewodnikiem. Mamy nadzieję, że również osoby niezwiązane z branżą IT uznają go za nieocenionego towarzysza w podróży do opanowania usługi Oracle APEX i tworzenia innowacyjnych aplikacji, które mają pozytywny wpływ.

Podręcznik i materiały uzupełniające zostały zaprojektowane z myślą o około 75 godzinach pracy studenta (3 ECTS). W celu nauki można zastosować różne metody pracy takie jak::

- wykłady i ćwiczenia laboratoryjne prowadzone przez wykładowców,
- nauczanie mieszane,
- samodzielna nauka.

W zależności od wiedzy początkowej uczących się i dostępnego czasu na realizację kursu, nauczyciele mogą łatwo skomponować zestaw rozdziałów dostosowanych do potrzeb edukacyjnych, takich jak: zajęcia pozalekcyjne, szkoły letnie, ograniczone czasowo wydarzenia wprowadzające do programowania nisko-kodowego dla wszystkich studentów (nie tylko z branży IT czy nauk komputerowych) oraz praktyków z różnych gałęzi przemysłu.

Do opracowania tego podręcznika i materiałów uzupełniających użyto wersji APEX 22 i 23. Wierzymy, że koncepty i główne technologie omówione i zastosowane w treści będą korzystne również dla uczących się następnych wersji APEX.

Przygotuj się na ekscytującą przygodę z nauką dzięki Oracle APEX! Korzystaj z kreatorów i programowania niskokodowego!

Profesor Robert Leskovar
BeeAPEX kierownik projektu, Katedra Informatyki, UM, Wydział Nauk Organizacyjnych

How to in APEX

1 **Jak uruchomic Oracle APEX?** 35
Vjeran Strahonja, Dijana Oreški . . . . . .

2 **Jak przygotować bazę danych?** 49
E. Kapsammer, W. Retschitzegger, W. Schwinger . . . . .

3 **Jak się poruszać w APEX?** 81
E. Kapsammer, W. Retschitzegger, W. Schwinger . . . . .

4 **Jak wymieniać dane w APEX?** 93
E. Kapsammer, W. Retschitzegger, W. Schwinger . . . . .

5 **Jak wygenerować pierwszy szkic wniosku?** 113
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . .

6 **Jak zarządzać raportami?** 137
Zuzana Žillová, Erik Malina, Matej Grochal, Andrej Staník, Andrea Meleková, Michal Kvet, Miroslav Potočár . . .

7 **Jak zarządzać formualrzami?** 159
Veronika Šalgová, Michal Kvet, Miroslav Potočár . .

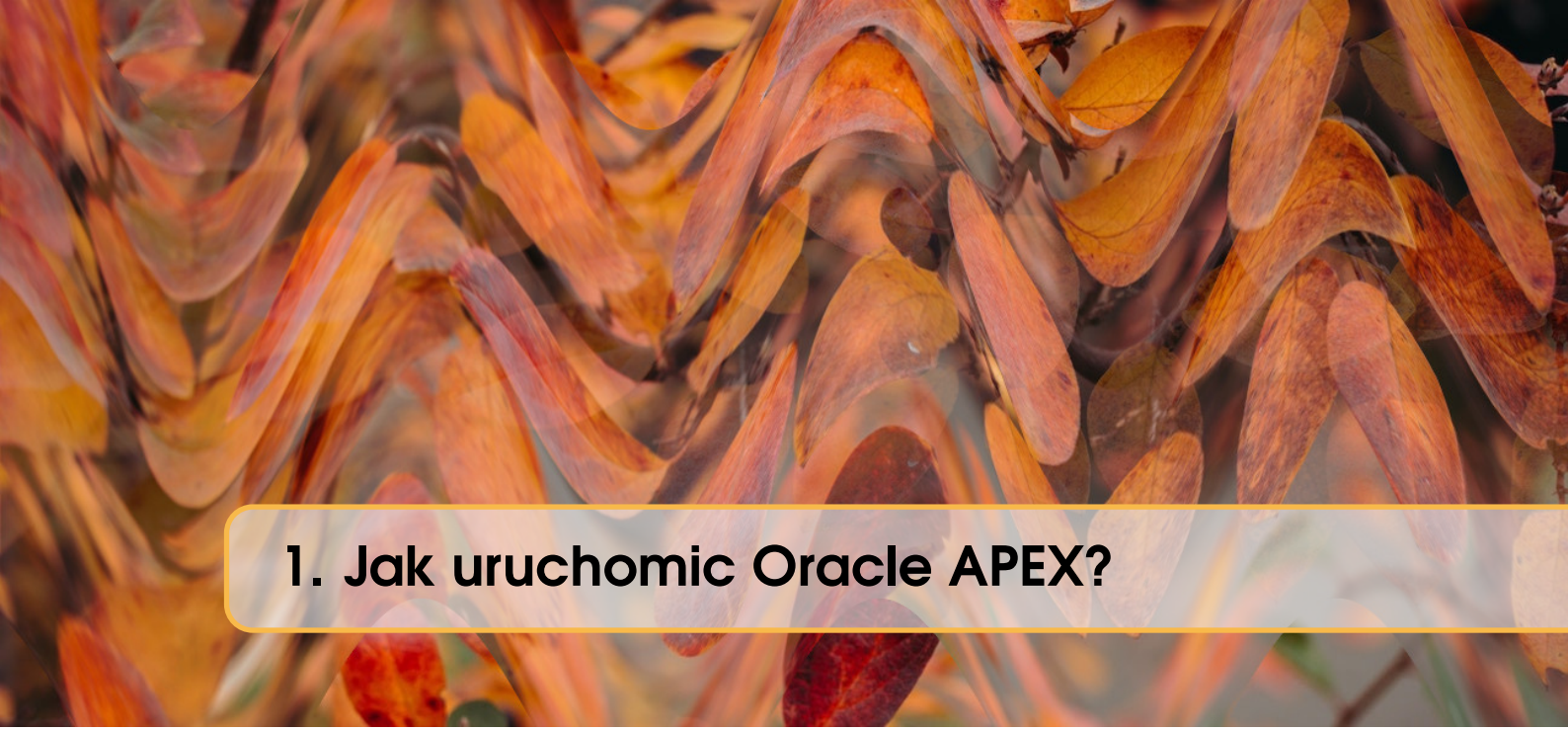
8 **Jak przekształcić raporty tekstowe w wykresy?** 167
Ivan Pastierik, Michal Kvet, Miroslav Potočár . . . . . .

9 **Jak zarządzać menu?** 185
Veronika Šalgová, Michal Kvet, Miroslav Potočár . .

10 **Jak współpracować w zespole?** 189
P. Staniszewski, M. Sońta, A. Kierzkowski . . . . . .

11 **Jak skorzystać z galerii aplikacji i wtyczek?** 201
Vjeran Strahonja, Dijana Oreški . . . . . .

12 **Jak zarządzać pakietowymi i wielojęzycznymi aplikacjami?** 209
R. Leskovar, U. Rajković, A. Baggia . . . . . .



1. Jak uruchomic Oracle APEX?

VJERAN STRAHONJA, DIJANA OREŠKI

1.1 Co to jest Oracle APEX i do czego służy?

1.1.1 Co to jest Oracle APEX?

Oracle APEX (znany także jako APEX lub Oracle Application Express) to platforma służąca do tworzenia aplikacji, wymagająca niewielkiej ilości programowania, opracowana przez Oracle Corporation, umożliwiająca użytkownikom tworzenie i wdrażanie aplikacji za pomocą interfejsu internetowego. Jest to zestaw narzędzi do tworzenia różnych rodzajów aplikacji składających się z programów do pracy z danymi w bazie danych, w tym: wprowadzania, przechowywania, wyświetlania, modyfikacji, obliczania i operacji logicznych z danymi itp. Korzystając z tej platformy można tworzyć własne aplikacje do różnych celów, w tym generować bazę danych na podstawie ustrukturyzowanych zbiorów danych lub modeli danych, pracować bezpośrednio z danymi w bazie danych za pomocą języka SQL, tworzyć interfejsy użytkownika, generować różne raporty tabelaryczne i graficzne oraz wdrażać aplikacje internetowe w bazach danych Oracle.

1.1.2 Co to jest aplikacja?

Aplikacja to na ogół zestaw programów komputerowych mających określony cel działając na zbiorze danych w bazie danych. Ze względu na tematykę podręcznika jesteśmy szczególnie zainteresowani aplikacjami biznesowymi, czyli tymi, których programy są wykorzystywane do przetwarzania danych na temat procesów biznesowych i transakcji. Opracowanie aplikacji, nawet prostej, jest złożonym procesem myślowym i twórczym, który wymaga pewnej wiedzy i umiejętności. Po pierwsze, wymagany jest algorytmiczny sposób. Jest to możliwość konceptualizacji, świadomość o istnieniu algorytmu w danej sytuacji, możliwość jego zastosowania, wiedza na temat projektowania i używania algorytmu oraz zdolność krytycznej oceny algorytmów. Wymagana jest także wiedza i umiejętności tworzenia wysokiej jakości kodu w języku programowania (kodowanie), testowania i usuwania błędów oraz inne rzeczy, które zwykle nazywamy programowaniem.

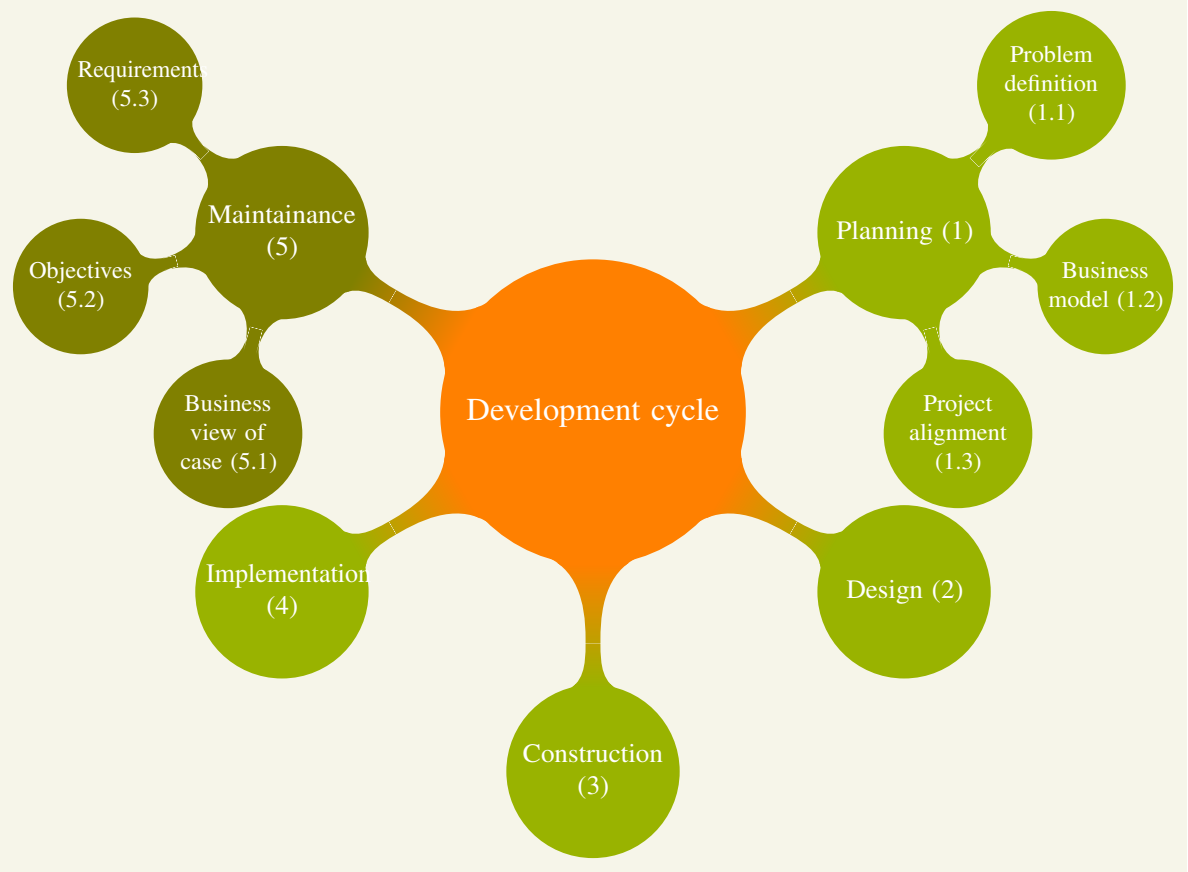
1.1.3 Jak Oracle APEX może pomóc w tworzeniu aplikacji?

Podczas tworzenia programów komputerowych nie możemy uniknąć algorytmicznego sposobu myślenia. Korzystanie z platform i narzędzi do programowania nisko kodowego (low code) i bez kodowego (no code), takich jak Oracle APEX, umożliwia osobom niebędącym programistami łatwe programowanie bez konieczności znajomości języka programowania i zaawansowanych

umiejętności programowania. Dzięki takim rozwiązaniom te osoby mogą skupić się na problemie i algorytmie, a nie na języku programowania i technologii. Jesteśmy szczególnie zainteresowani łatwym w obsłudze tworzeniem aplikacji umożliwiających dostęp, ujednolicanie, analizowanie i wyświetlanie danych z otwartych źródeł danych oraz z firmowych baz danych. Otwarte dane to dowolnie dostępne dane dla każdego; ktokolwiek może uzyskiwać do nich dostęp, przetwarzać i udostępniać je. Mogą być używane bez ograniczeń (lub z niektórymi), współdzielone z innymi lub wykorzystywane do tworzenia nowych projektów. Otwarte dane są udostępniane przez rządy, firmy i inne organizacje w celu promowania przejrzystości i współpracy. Obecnie istnieje wiele łatwo dostępnych źródeł danych meteorologicznych, drogowych, geologicznych i podobnych, a także statystyk dotyczących edukacji, zdrowia, gospodarki itp. Korporacyjne bazy danych zawierają dane dotyczące transakcji biznesowych i innych danych, które są tworzone w wyniku procesów biznesowych. Wizualizacje, analizy, integracje i różne widoki tych danych mogą być wykorzystane do podejmowania decyzji biznesowych lub osiągnięcia pewnych innych korzyści. Usługa Oracle APEX służy do tworzenia prototypów, które służą do omawiania zadań programu i pokazywania specjalistom ds. IT, czego użytkownik oczekuje od aplikacji. Celem prototypu jest odzwierciedlenie zasadniczych cech systemu. W następnej kolejności prototyp zostaje częścią systemu, który będzie używany przez użytkownika, lub służy jako podstawa do utworzenia systemu w innym języku programowania, narzędziu lub platformie.

1.1.4 Cykl tworzenia aplikacji

Niezależnie od tego, jaki rodzaj aplikacji rozwijamy i jakiego rodzaju ram metodologicznych, narzędzi programistycznych lub platform używamy, cykl tworzenia aplikacji przechodzi kilka etapów. Etapy te są różne w różnych podejściach metodologicznych. Rysunek 1.1 przedstawia jeden z możliwych schematów cyklu tworzenia aplikacji (zob. rys. 1.1).



Rysunek 1.1: Cykl tworzenia aplikacji.

Jak przedstawiono na rysunku, cykl tworzenia aplikacji składa się z następujących etapów lub procesów:

1. Planowanie (definiowanie problemu i projektu)
   - definicja i analiza modelu biznesowego,
   - specyfikacja wymagań,
   - określenie głównych procesów i danych,
   - wybór odpowiedniej technologii i podjęcie decyzji o architekturze aplikacji,
   - planowanie rozwoju.
2. Projektowanie aplikacji
   - analiza i specyfikacja wymagań,
   - lprojekt logicznej aplikacji (przypadki użycia i model danych logicznych).
3. Budowanie aplikacji
   - projektowanie baz danych,
   - tworzenie prototypu aplikacji,
   - testowanie
4. Wdrożenie aplikacji

Po etapie planowania następuje projektowanie, budowanie i wdrożenie aplikacji. Te fazy lub procesy mogą być realizowane sekwencyjnie lub iteracyjnie (w sposób ewolucyjny, przyrostowy), co umożliwia wielokrotne przechodzenie przez cały cykl rozwojowy i dodawanie nowych funkcji w każdej iteracji.

Planowanie aplikacji Cykl rozwoju aplikacji rozpoczyna się od etapu planowania. Dla prostych aplikacji początkową informacją w fazie planowania może być definicja problemu – dokument, który precyzyjnie określa problem. Jakie są objawy problemu? Kto go doświadcza? Dlaczego jest

to poważny problem? Opis może być krótki, składający się z kilku zdań lub kilku stron tekstu. Dla bardziej złożonych aplikacji biznesowych, dokładne specyfikacje powinny zawierać:

- Opis z perspektywy biznesowej, jak pracownicy wewnątrz organizacji postrzegają sytuację lub czego oczekują od aplikacji.
- Cele rozwoju aplikacji określające, co aplikacja ma na celu w kontekście biznesowym, technologicznym oraz inne cele i wskaźniki sukcesu.
- Model biznesowy prezentujący wartości, segmenty klientów, relacje i kanały komunikacji z klientem, kluczowych partnerów, działania, zasoby oraz strukturę przychodów i kosztów itp.
- Wymagania dotyczące aplikacji określające funkcjonalności, wydajność, technologie itp.
- Plan projektu zawierający zakres i etapy przyszłości projektu, działania realizowane w każdej fazie, zasoby, połączenia między działaniami, ich czas trwania, koszty i ryzyko itp,

Istnieją różne metody tworzenia każdej z powyższych specyfikacji, króre są one opisane w literaturze. Na przykład, Alexander Osterwalder [6] stworzył metodę Business Model Canvas (model biznesowy). Jest to narzędzie służące do opracowywania, ulepszania i dokumentowania nowych oraz istniejących modeli biznesowych Jak przedstawiono na rysunku 1.2, metoda Business Model Canvas to wizualny schemat składający się z dziewięciu bloków, które opisują propozycję wartości oferowaną przez organizację, jej klientów, relacje i kanały komunikacji z nimi, kluczowych partnerów, działania, zasoby oraz strukturę kosztów i strumienie przychodów.



Rysunek 1.2: Business Model Canvas [6]

.

Podczas gdy Business Model Canvas (BMC) jest narzędziem pomocnym w zrozumieniu, analizie i projektowaniu modelu biznesowego (zwłaszcza w aspekcie funkcji, które powinna pełnić rozwijana aplikacja), specyfikacja wymagań oprogramowania (SRS) to dokument precyzyjnie opisujący, jakie zadania powinno wykonywać oprogramowanie i w jaki sposób będzie funkcjonować. Opisuje on funkcje, wydajność oraz inne cechy, które aplikacja musi spełniać, aby sprostać wymaganiom modelu biznesowego i potrzebom wszystkich użytkowników.

Projektowanie aplikacji Od początków ery komputerowej do chwili obecnej, opracowano setki różnorodnych podejść metodologicznych, technik oraz metod projektowania aplikacji. Metody te ewoluowały wraz z postępem technologicznym. Programowanie nisko kodowe/bez kodowe (low code/ no code) to nowoczesne podejście, skierowane głównie dla osób, które nie są profesjonalnymi programistami, aczkolwiek tworzenie aplikacji z jego wykorzystaniem nadal wymaga odpowiednich umiejętności projektowania. Jak wcześniej podkreślono, aplikacja komputerowa to zestaw programów współpracujących z bazą danych. Projektując aplikację, musimy uwzględnić zarówno programy, jak i bazę danych. Jeśli baza danych już istnieje, kluczem jest zrozumienie jej modelu. W kontekście programowania nisko kodowego/bez kodowego skupimy się na kilku podstawowych technikach projektowania, które mają swoje graficzne reprezentacje:

- Model przypadków użycia (UC - use case) definiujący relacje między aplikacją, a jej otoczeniem.
- Model relacji między jednostkami (ER – entity-relationship) opisujący strukturę danych na poziomie logicznym.

W dalszej części tego rozdziału szczegółowo omówimy przypadki użycia, natomiast model relacji między jednostkami (ER) zostanie przedstawiony w rozdziale2.

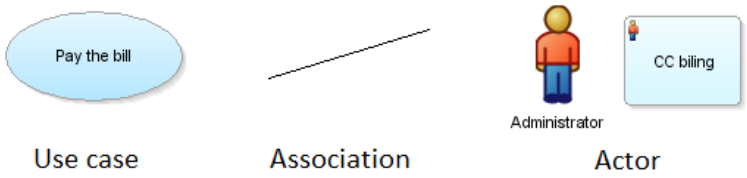
Przypadki użycia (UC) Przypadki użycia (UC) stanowią jedną z podstawowych metod projektowania oprogramowania. Ilustrują one interakcje między systemem (aplikacją) a jego otoczeniem (użytkownikami i innymi systemami). Jest to metoda na tyle intuicyjna, że zrozumiała jest nawet dla użytkowników, którzy nie są ekspertami w dziedzinie IT. Przypadki użycia są kluczowym elementem Unified Modeling Language (UML) – najczęściej stosowanego języka modelowania w inżynierii oprogramowania. UML oferuje standardowy sposób wizualizacji projektu systemu. Organizacja Object Management Group (OMG) jest odpowiedzialna za rozwój i standaryzację UML, który zawiera 13 technik schematycznych do modelowania struktury, zachowania i interakcji oprogramowania. W tym kontekście skupimy się głównie na przypadkach użycia.

Każdy model przypadku użycia składa się z dówch części:

- Specyfikacji przypadków użycia (dokładny opis)
- Diagramu przypadków użycia.

Rozpoczniemy od analizy diagramu przypadków użycia, który przedstawia funkcje systemu z perspektywy obserwatora spoza systemu, nie zagłębiając się w jego wewnętrzną strukturę.

Podstawowe pojęcia diagramu przypadków użycia przedstawiono na rysunku 1.3.

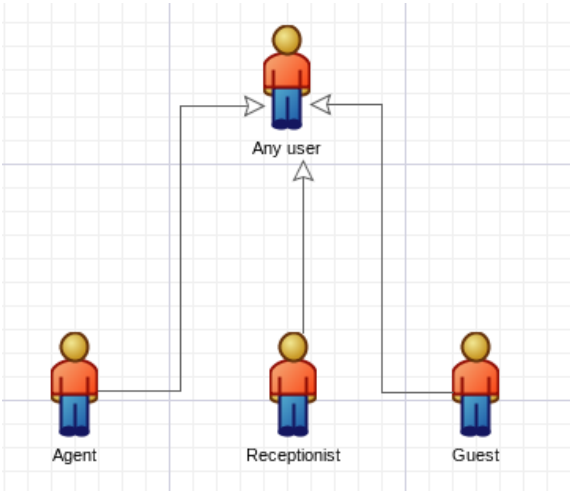


Rysunek 1.3: Podstawowe pojęcia diagramu przypadków użycia.

Przypadek użycia to zdefiniowany zbiór działań, które system (np. element oprogramowania) może realizować, współdziałając z podmiotami zewnętrznymi. Przedstawia on scenariusz, w którym określone aktory wchodzą w interakcję z systemem, aby osiągnąć konkretny cel lub spełnić określone zadanie. Taki przypadek jest abstrakcyjnym pojęciem z jasno określonym celem.

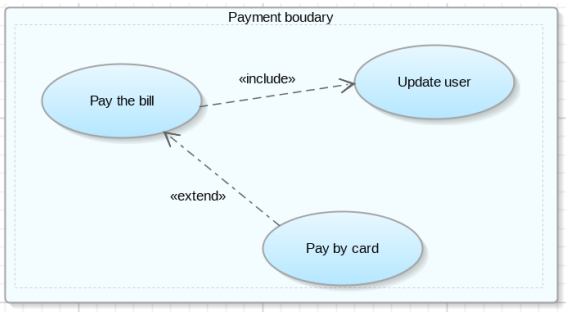
„Association” definiuje połączenie między uczestnikami interakcji, na przykład między aktorem, a przypadkiem użycia, ukazując tym samym ich współdziałanie oraz relację pomiędzy systemem, a otoczeniem. Aktor określa zestaw ról, które wchodzą w interakcję z systemem w taki sam sposób, takie jak określona grupa użytkowników lub zewnętrzny system. Aktor jest to podmiot spoza analizowanego systemu, który wchodzi z nim w interakcję, ale nie stanowi jego części. Może to być konkretny użytkownik (na przykład Lekarz, Kierowca) lub inny system (np. system płatnościowy, bank).

Diagram może również przedstawiać strukturę klasyfikacji (zob. rys. 1.4), czyli generalizację-/specyfikację określonych pojęć, jak na przykład Ąktorzy". Wszystkie klasy podrzędne (dzieci) dziedziczą atrybuty i zachowanie klasy nadrzędnej, czego skutkiem jest również dziedziczenie połączeń do przypadków użycia od klasy nadrzędnej. Jednakże konkretna klasa podrzędna może mieć unikatowe połączenia, których nie posiada klasa nadrzędna, ani inne klasy podrzędne.



Rysunek 1.4: Generalizacja/specjalizacja “Aktorów”

Rys. 1.5 przedstawia granice systemu w postaci prostokąta z nazwą systemu, wewnątrz którego znajdują się przypadki użycia.



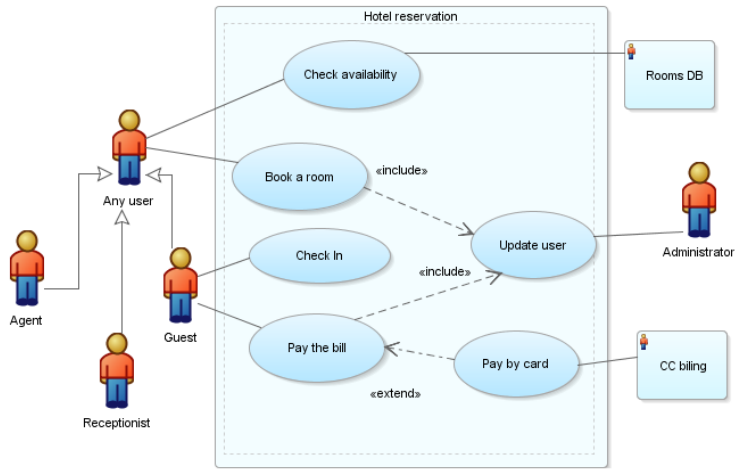
Rysunek 1.5: Granice systemu i stowarzyszeń "uwzględniają"i żozszerzają".

Diagramy przypadków użycia nie ilustrują sekwencji działań, ani przepływu pracy. Niemniej jednak istnieją dwa dozwolone rodzaje relacji między przypadkami użycia, które są przedstawione na rysunku 1.5:

- "include": jest to relacja, w której podstawowy przypadek użycia zawiera w sobie zachowanie innych przypadków użycia. W ten sposób często używane funkcje i zachowania są rozdzielane na przypadek użycia, który w razie potrzeby uwzględnimy w innych przypadkach użycia. Należy pamiętać, że podstawowy przypadek użycia nie jest kompletny bez uwzględnienia.

- ęxtend"to relacja, w której konkretny przypadek użycia działa jako rozszerzenie dla podstawowego przypadku użycia, wskazując, że zachowanie podstawowego przypadku użycia jest wzbogacane przez działania rozszerzenia. Jednocześnie, to rozszerzenie nie stanowi integralnej części podstawowego przypadku użycia, a sama komunikacja nie zawsze musi być obecna i działać bez zarzutu.

Zrozumienie tych pojęć diagramu przypadków użycia jest prostsze dzięki analizie praktycznych przykładów. Doskonałą ilustracją tych relacji jest przykład działania systemu rezerwacji hotelowej, przedstawiony na rysunku 1.6.



Rysunek 1.6: Schemat przypadków użycia systemu rezerwacji hotelowej.

System, którego zachowanie obserwujemy to system Rezerwacji hotelu. System znajduje się w granicach wskazanych przez prostokąt. Otoczenie systemu znajduje się poza tym prostokątem. W obrębie systemu widocznych jest sześć przypadków użycia, które są przedstawione za pomocą elips z odpowiednimi nazwami. Poza systemem znajdują się aktorzy, którzy wchodzą w interakcję z systemem, a więc z jego przypadkami użycia. Te interakcje są przedstawione jako linie łączące, nazywane „association". Każdy Użytkownik („any user") jest aktorem, który komunikuje się z funkcją Ćheck availiabillity". Ten przypadek użycia wchodzi w interakcję z zewnętrznym aktorem o nazwie Żooms DB". Jest to rodzaj zewnętrznego systemu rezerwacji przechowującego informacje o dostępności pokojów i dokonanych rezerwacjach. Każdy Użytkownik ma możliwość zarezerwowania pokoju. Ten przypadek użycia zawiera inny przypadek, którym jest Aktualizacja danych użytkownika („update user"). Struktura klasyfikacyjna aktora o nazwie Użytkownik jest przedstawiona po lewej stronie. Aktorzy: Agent („agent"), Żecepcjonista"(„receptionist") i "Gość"(„guest") są podtypami aktora Użytkownik („any user"). W prezentowanym scenariuszu tylko Gość ma możliwość komunikacji z funkcjami Zamelduj (ćheck in") oraz Zapłać za pobyt („pay the bill"). Funkcja Aktualizacja danych użytkownika jest częścią funkcji Zapłać za pobyt, co oznacza, że w trakcie płatności możliwa jest także aktualizacja danych użytkownika. Funkcja Zapłać kartą („pay by card") rozszerza funkcjonalność Zapłać za pobyt, co oznacza, że w trakcie płatności można dokonać transakcji kartą płatniczą. Zapłać kartą komunikuje się z zewnętrznym serwisem płatności kartą Fakturowanie CC („CC billing"). Administrator ma możliwość komunikacji z funkcją Aktualizacja danych użytkownika oraz aktualizacji informacji o użytkowniku. Jak wcześniej wspomniano, model przypadku użycia składa się z dwóch części:

- Specyfikacji przypadków użycia (dokładny opis)
- Diagramu przypadków użycia.

Specyfikacja przypadku użycia zawiera opisy scenariuszy, wewnętrzną logikę przypadków użycia, warunki początkowe, warunki końcowe, interfejsy, komunikaty systemowe, specyfikacje

błędów oraz przetwarzanie wyjątków. Zwykle korzysta się z określonych szablonów do specyfikacji przypadków użycia, tak jak to będzie prezentowane w tej książce.

Tworzenie, wdrażanie i utrzymanie aplikacji TTworzenie, wdrażanie i utrzymanie aplikacji polega na napisaniu programu oraz stworzeniu fizycznej bazy danych. W tym celu skorzystamy z platformy Oracle APEX, co zostanie dokładniej opisane w dalszej części książki. Pierwsza działająca wersja aplikacji jest de facto jej prototypem. Ten prototyp można modyfikować oraz rozwijać. W zależności od przyszłego wykorzystania prototypu, możemy wyróżnić dwa jego typy: rozwijalny prototyp, który jest stopniowo udoskonalany do wersji gotowej do użytku oraz prototyp "do odrzucenia", służący głównie jako ilustracja funkcjonalności i model dla przyszłej aplikacji, zazwyczaj budowanej na innej platformie. Jeżeli decydujemy się na wdrożenie prototypu, musimy go zintegrować z środowiskiem produkcyjnym, przeszkolić użytkowników, przygotować bazę danych itd.

Po etapie wdrażania następuje faza eksploatacji i utrzymania, która obejmuje korekty błędów, optymalizacje wydajności oraz niewielkie rozszerzenia funkcjonalne. W przypadku potrzeby wprowadzenia znaczących modyfikacji, rozpoczyna się kolejny cykl tworzenia aplikacji.

1.2 Jak uruchomić Oracle APEX?

Rozwój oprogramowania jest postrzegany jako skomplikowany z uwagi na konieczność wykonywania wielu zadań programistycznych. Aby przeciwdziałać tej złożoności, proponowane są rozwiązania oparte na podejściu niskiego kodu. Metodyka niskiego kodu pozwala na tworzenie zaawansowanych aplikacji, które są zarówno funkcjonalne, jak i skomplikowane, bez konieczności pisania pojedyńczej linijki kodu. Oracle Application Express to jedno z najlepszych narzędzi nisko-kodowych dostępnych na rynku. Istnieje wiele sposobów, aby rozpocząć pracę z Oracle APEX. Platformy można stosować wszędzie tam, gdzie wykorzystywana jest baza danych - zarówno lokalnie, jak i w prywatnej chmurze. Może to być dedykowany serwer fizyczny, maszyna wirtualna czy obraz docker (który jest szczególnie ceniony przez deweloperów Oracle Application Express i może być uruchamiany na laptopie, w autobusie czy samolocie). Co więcej można również korzystać z Exadata, czyli zaawansowanego fizycznego serwera APEX działającego w usługach chmurowych. Wszystkie te opcje są omawiane w niniejszym rozdziale, tak samo jak sposób dostępu do środowiska deweloperskiego.

1.2.1 Jaki jest Twój poziom umiejętności?

Istnieje wiele możliwości rozpoczęcia pracy z Oracle Application Express (APEX). Aktualny poziom twoich umiejętności decyduje o zalecanym sposobie rozpoczęcia. Zakładając, że posiadasz specjalistyczną wiedzę z różnych procesów organizacyjnych (np. sprzedaż, rekrutacja, produkcja, kredytowanie, marketing, prawo, opieka zdrowotna, budownictwo, edukacja) oraz następujące poziomy umiejętności cyfrowych:

- całkowity początkujący: swobodne korzystanie z przeglądarki internetowej, brak doświadczenia w programowaniu; zalecenie: skorzystaj z instancji lokalnej, apex.oracle.com lub Oracle Academy
- początkujący: swobodne korzystanie z przeglądarki internetowej, podstawowe doświadczenie w programowaniu, np. korzystanie z arkuszy kalkulacyjnych i tworzenie formuł; jakikolwiek inny język programowania; zalecenie: skorzystaj z instancji lokalnej, apex.oracle.com lub Oracle Academy
- początkujący deweloper: swobodne korzystanie z przeglądarki; podstawowe doświadczenie w programowaniu; zrozumienie pojęć bazy danych, tabeli, kolumny tabeli, klucza głównego, klucza obcego; podstawy tworzenia zapytań, wstawiania, aktualizacji i usuwania; zalecenie: skorzystaj z instancji lokalnej, apex.oracle.com, Oracle Academy lub urządzenia VBox
- zaawansowany deweloper: swobodne korzystanie z przeglądarki; średnio zaawansowane

umiejętności programistyczne związane z procedurami i funkcjami w różnych systemach zarządzania bazami danych; bardziej zaawansowane zapytania, wstawianie, aktualizowanie i usuwanie; zalecenie: skorzystaj z instancji lokalnej, apex.oracle.com, Oracle Academy, urządzenia VBox lub obrazu Docker dla APEX
- osoba w drodze do bycia profesjonalnym programistą: rozwijanie umiejętności w zakresie PL/SQL i JavaScript; zalecenie: skorzystaj z instancji lokalnej, apex.oracle.com, obrazu Docker dla APEX lub instancji OCI APEX.

Twoja rola jako użytkownika oraz lokalizacja systemu Oracle APEX decydują o sposobie logowania się i korzystania z aplikacji. Oracle APEX może być instalowany lokalnie na twoim komputerze lub w zdalnym środowisku, takim jak usługa Oracle Cloud. W zależności od rodzaju instalacji procedura logowania się różni. Użytkownicy muszą stworzyć obszar roboczy, dodać użytkowników Oracle APEX, a następnie zalogować się do tego obszaru przed tworzeniem lub instalacją aplikacji. Wielu użytkowników może współpracować przy tej samej instalacji Oracle APEX, korzystając z jednego obszaru roboczego, jednocześnie zachowując prywatność swoich danych i aplikacji. Jeśli przeglądarka internetowa obsługuje JavaScript, użytkownicy mogą logować się do obszaru roboczego, aby uzyskać dostęp do strony głównej Oracle APEX. Każdy obszar roboczy ma unikalną nazwę i identyfikator. W ramach usług administracyjnych Oracle APEX obszary robocze mogą być tworzone ręcznie przez administratora instancji lub użytkownicy mogą składać wnioski. Niezależny program o nazwie Oracle APEX Administration Services służy do nadzorowania całej instancji Oracle APEX.

1.2.2 Użyj lokalnej instancji APEX

W tej sekcji opisano sposób instalowania systemu Oracle APEX w instalacji lokalnej. Instalacja Oracle APEX składa się z kilku kroków. Są to:

- **Wybór pomiędzy Full, a Runtime Environment**. Full Environment daje pełny dostęp do środowiska rozwojowego App Builder. Dla implementacji produkcyjnych, w których chcesz uruchamiać niezmienne aplikacje, Runtime Environment jest dobrym wyborem. Oracle APEX umożliwia instalację tylko wersji runtime Oracle APEX do testowania i instancji produkcyjnych. Ponieważ deweloperzy nie mogą przypadkowo zaktualizować programu produkcyjnego w instancji runtime, to środowisko zmniejsza zainstalowany ślad i prawa oraz zwiększa bezpieczeństwo aplikacji. Użytkownicy mogą uruchamiać aplikacje produkcyjne w środowisku Oracle APEX runtime, ale nie ma ono interfejsu internetowego do administracji. Środowisko runtime jest bardziej bezpieczne, ponieważ zawiera tylko te komponenty, które są potrzebne do uruchomienia programu.
- **Weryfikacja wymagań instalacyjnych.** Sprawdź, czy Twój system spełnia minimalne wymagania instalacyjne. Istnieje pięć grup wymagań: a) Wymagania bazy danych Oracle (wersja Oracle APEX 22 wymaga wersji Oracle Database 12.1.0.2 lub nowszej. Oracle APEX działa na wszystkich edycjach bazy danych, w tym Enterprise Edition (EE), Standard Edition (SE) i Express Edition (XE). Oracle APEX można instalować zarówno w pojedynczej instancji bazy danych, jak i w bazie danych Oracle Real Application Clusters (Oracle RAC).), b) Wymagania dotyczące przeglądarki (Oracle APEX wymaga przeglądarki z włączonym JavaScriptem i obsługuje bieżącą i poprzednią główną wersję Google Chrome, Mozilla Firefox, Apple Safari oraz Microsoft Edge.), c) Wymagania dotyczące Web Listener (Oracle APEX wymaga Oracle REST Data Services (ORDS) w wersji 19.x lub nowszej), d) Wymagania dotyczące miejsca na dysku (Oracle APEX wymaga 310 MB wolnego miejsca na pliki oprogramowania APEX na systemie plików, jeśli korzystasz tylko z angielskiego pobierania i 705 MB przy pełnym pobieraniu, 220 MB wolnego miejsca w przestrzeni tabeli APEX, 100 MB wolnego miejsca w przestrzeni tabeli SYSTEM, 60 MB wolnego miejsca w przestrzeni tabeli APEX na każdy dodatkowy język (inny niż angielski), e) Wymagania dotyczące Oracle XML DB (Oracle XML DB musi być zainstalowany w bazie danych Oracle, której chcesz użyć,

jeśli instalujesz pełne środowisko rozwojowe. Jeśli korzystasz z wstępnie skonfigurowanej bazy danych stworzonej podczas instalacji lub przez Database Configuration Assistant lub DBCA, Oracle XML DB jest już zainstalowany i skonfigurowany.)

- **Zainstaluj** oprogramowanie - Zainstaluj Oracle APEX, pobierając plik ZIP ze strony Oracle APEX, a następnie pobierz i zainstaluj Oracle REST Data Services (ORDS)

1.2.3 https://apex.oracle.com

Przejdź w swojej przeglądarce pod ten link this URL. Stworzenie bezpłatnego obszaru roboczego to najszybszy sposób, aby rozpocząć pracę z Oracle APEX. Dołączenie zajmuje zaledwie kilka sekund, a obszar roboczy jest przygotowany dla użytkowników, aby zaczęli tworzyć aplikacje. Ten sposób nie wymaga posiadania konta w chmurze i jest bezpłatne. Aby rozpocząć pracę z Oracle APEX, należy:

- Wprowadź w swojej przeglądarce internetowej: apex.oracle.com
- Kliknij przycisk Śtart for Free Today"
- Na pojawiającej się stronie internetowej wybierz Żequest a free Workspace"
- Wprowadź swoje imię, adres e-mail, nadaj nazwę swojemu obszarowi roboczemu i kliknij Ńext"
- Wypełnij ankietę i podaj powód, dla którego składasz wniosek o obszar roboczy. Przeczytaj regulamin i zaakceptuj jego warunki, klikając Ńext"
- W tym momencie otrzymasz wiadomość e-mail od Oracle APEX. Kliknij w link w wiadomości, aby się zarejestrować i ustawić hasło

1.2.4 Virtual Box Appliance (maszyna wirtualna)

Wejdź w przeglądarce pod ten link this URL. Głównym celem tego urządzenia jest rozwijanie i testowanie aplikacji. Wersja APEX jest zwykle starsza niż najnowsza. Pobierz Oracle Virtual Box oraz Extension Manager. Skonfiguruj Oracle Virtual Box oraz Extension Manager. Zaimportuj plik .ova. Uruchom urządzenie.

Oracle APEX Development VM to gotowa do użycia maszyna wirtualna, którą można wykorzystać, po prostu importując ją do VirtualBox. Wstępnie skonfigurowane maszyny wirtualne od Oracle są dostępne dla użytkowników Oracle. Wystarczy zainstalować Oracle Virtual Box (darmowy klient do maszyn wirtualnych) i następnie zaimportować odpowiednie urządzenie (wstępnie skonfigurowana VM). Oracle udostępnia kilka takich maszyn tutaj URL.

Aby skonfigurować VM, wykonaj poniższe kroki:

- Pobierz i zainstaluj Oracle VM VirtualBox na swoim systemie głównym
- Pobierz odpowiednie pliki (zaleca się używanie menedżera pobierania)
- Zaimportuj swoją maszynę wirtualną, wybierz Plik > Importuj Urządzenie, co uruchomi Kreator Importu Urządzenia. Kliknij "Wybierz...", aby odnaleźć katalog, w którym zgromadzono wszystkie pliki, i wybierz plik OTN_Developer_Day_VM.ova. Następnie kliknij "Dalej", aby rozpocząć proces importu. Podczas importu zostaniesz poproszony o zaakceptowanie odpowiednich licencji dla deweloperów. Po zakończonym imporcie zobaczysz komunikat 'Oracle Developer Days (Wyłączony)'.
- Sprawdź działanie swojej VM: Po zakończeniu importu, dwukrotnie kliknij w aplikację OTN Developer Days. Kliknij ÓK", aby zamknąć okna informacyjne VirtualBox. Gdy pojawi się ekran Enterprise Linux 6, możesz się zalogować (nazwa użytkownika i hasło to óracle"). Poczekaj, aż proces się zakończy; będzie gotowy, gdy pojawi się okno terminala, które można zamknąć. Gdy zakończysz pracę w gościnnej maszynie wirtualnej, możesz ją wyłączyć wybierając System > Wyłącz, co przywróci maszynę wirtualną do stanu "Wyłączony".

Główne korzyści z korzystania z maszyn wirtualnych to:

- Możliwość korzystania z nowego oprogramowania Oracle bez konieczności przeprowadzania skomplikowanych instalacji lub angażowania dostawców.

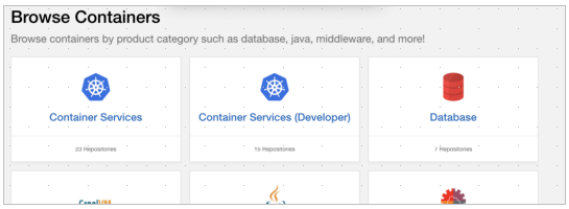
- Eksperymentowanie b. ez narażania swojego komputera.
- Darmowe, prowadzone przez Oracle szkolenia w formie praktycznych laboratoriów.

1.2.5 Docker APEX (klucz dostępu do APEX)

Pobierz oficjalny Docker APEX ze strony Oracle. Przejdź w przeglądarce pod wskazany link this URL.

Oracle APEX można zainstalować jako klucz dostępu. Klucze dostępu stanowią bardzo wygodne rozwiązanie dla deweloperów, gdyż można je instalować zarówno na komputerach stacjonarnych, jak i laptopach. Należy się jednak upewnić, by pobierać klucz dostępu APEX jedynie z wiarygodnych źródeł. Konfiguracja zwykle przebiega szybko i sprawnie. Podczas nauki Oracle Database i APEX wersja może nie być kluczowa, ale w przypadku środowiska produkcyjnego warto zwracać uwagę na konkretną wersję. W dalszej części tej sekcji przedstawiono jeden ze sposobów instalacji kluczy dostępów do APEX lokalnie na komputerze. W ramach tego procesu klonujemy oficjalny klucz dostępu od Oracle, który zawiera narzędzia deweloperskie.

1. Pierwszym krokiem jest instalacja kluczy na komputerze. Klucze można łatwo pobrać i zainstalować ze strony głównej: https://docs.docker.com/desktop/install/windows-install/
2. Drugim krokiem jest rejestracja bezpłatnego konta w rejestrze Oracle: https://container-registry.oracle.com/ords/f?p=113:10 Klucze dostępu są pobierane z tego rejestru (zob. rys. 1.7).



Rysunek 1.7: Szukanie APEX docker (kluczy dostępu APEX).

3. Trzeci etap to otwarcie okna terminala i uzyskanie dostępu do rejestru Oracle za pomocą wcześniej utworzonego użytkownika.

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

Lepiej jest utworzyć sieć w środowisku, aby klucze mogły się komunikować z innymi kluczami za pomocą nazwy hosta docker

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

W tym momencie zrobiliśmy wszystko, czego potrzebujemy i możemy wykonać polecenie, aby uruchomić Oracle Database XE na naszym laptopie.

```
docker run -d --name testapex --hostname database
  --network=ords-database-network -p 1521:1521
  container-registry.oracle.com/database/express:latest
```

Należy pamiętać, że nazwa parametru, nazwa hosta i sieć zostały tu użyte. Wyjaśnienie:

- -d uruchomi kontener w trybie odłączonym.
- Parametr -name określa nazwę pojemnika.
- -p odwzorowuje port 1521 na komputerze-hoście do portu 1521 w kontenerze, dzięki czemu możemy połączyć się z bazą danych.
- -network łączy kontener z utworzoną przez nas siecią.
- –hostname nadaje nazwę serwerowi DB.
- Ostatnim parametrem jest obraz, którego chcemy użyć do rozruchu kontenera.

Jeśli chcesz połączyć się przez sqlplus można wykonać to polecenie:

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

Za pomocą polecenia *show pdbs* można sprawdzić status pdbs. Proszę zmienić hasło domyślne, używając poniższego polecenia (w takim przypadku nowym hasłem będzie Welcome1!!). Przed wykonaniem tego polecenia proszę się upewnić, że pdb XEPDB1 jest otwarty do odczytu i zapisu).

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

Aby skonfigurować system Apex dla instalacji Oracle XE, należy wykonać kolejne czynności. Z sesji końcowej ściągnij obraz liczby porządkowej.

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

Następnie wykonaj polecenie pull, aby pobrać obraz ords.

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

Aby skonfigurować instalację APEX, należy utworzyć katalog.

```
mkdir ~/APEX
```

Umieść informacje o ciągu w pliku w tym katalogu.

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

Zauważ, że użyto parametru nazwy hosta dla dockera Oracle Database XE oraz usługi pdb dla ciągu połączenia. Zdefiniuj zmienną CONN_STRING w następujący sposób (powinna mieć poniższą formę, bez pojedynczego cudzysłowu):

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

W tym momencie możemy uruchomić docker.

```
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
```

W tym miejscu użyto tej samej sieci parametrów. Aby monitorować instalację, można otworzyć inną sesję terminala i wykonać to polecenie:

```
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
```

Zmień hasło użytkowanika APEX_PUBLIC_USER wewnątrz bazy danych. Aby to zrobić, proszę się zalogować do systemu Oracle XE przy użyciu programu sqlplus.

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

Ustaw hasło.

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

Teraz można się połączyć ze środowiskiem APEX na komputerze lokalnym za pomocą przeglądarki.

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

1.2.6 Instancja APEX w Oracle Cloud Infrastructure

Oracle Cloud Infrastructure (OCI) oferuje rozwój aplikacji APEX wymagającej niskiego poziomiu programowania przy użyciu infrastruktury Autonomous. Jest to w pełni zarządzaną usługa, która jest wstępnie skonfigurowana i gotowa do użycia. OCI zapewnia elastyczną skalowalność, bezpieczeństwo, wysoką dostępność oraz globalny dostęp poprzez regionalne centra danych w chmurze. W tej sekcji opisano procedury, jak uzyskać dostęp do Oracle Cloud Infrastructure (OCI) w celu utworzenia instancji usługi APEX .Poniżej przedstawiono etapy uzyskiwania dostępu do OCI:

1. **Załóż konto w OCI**. Aby korzystać z usługi APEX, użytkownik musi posiadać konto w OCI i mieć do niego dostęp. Jeżeli Twoja organizacja ma umowę sprzedaży z Oracle, możesz używać istniejącego już konta OCI. Aby uzyskać dostęp, skontaktuj się najpierw z administratorem OCI w Twojej firmie. Jeśli jesteś indywidualnym użytkownikiem zaczynającym od podstaw lub nie jesteś pewny, od czego zacząć, rozważ zapisanie się do Oracle Cloud Free Tier. Oprócz bezpłatnej początkowej puli kredytów na chmurę, oferuje ona nieograniczone w czasie bezpłatne konto i dzierżawę w OCI. W trakcie okresu próbnego te kredyty można wykorzystać na zakup płatnych usług OCI, takich jak usługa APEX. Jeżeli chcesz kontynuować korzystanie z usługi APEX po zakończeniu okresu próbnego lub po wyczerpaniu bezpłatnych kredytów (cokolwiek nastąpi wcześniej), musisz przekształcić swoje konto na płatne i zakupić więcej kredytów. Jeśli nie dokonasz aktualizacji przed zakończeniem wersji próbnej, Twoje konto OCI zostanie przekształcone tak, by mogło korzystać jedynie z usług OCI bezpłatnej wersji. Usługa APEX oferuje również ograniczoną, bezpłatną wersję. Oracle zaleca przekształcenie konta na płatne i zakup dodatkowych kredytów w trakcie lub po zakończeniu okresu próbnego. Aby rozpocząć proces rejestracji, zapoznaj się z ofertą Oracle Cloud Infrastructure Free Tier. Aby rozpocząć rejestrację, odwiedź stronę: https://signup.oraclecloud.com/.
2. **Zaloguj się do konsoli OCI.** W tym celu użyj przeglądarki internetowej, otwórz stronę logowania do konsoli OCI. Wprowadź swój login oraz hasło, a następnie nazwę Twojego konta w chmurze (znanej również jako nazwa twojego dzierżawcy). Twoja wiadomość powitalna zawiera zarówno twoją nazwę użytkownika, jak i nazwę konta w chmurze.

1.2.7 Instancja APEX w Oracle Academy

Oracle Academy umożliwia członkom instytucjonalnym i ich studentom dostęp do Oracle Application Express (APEX) w chmurze do praktycznych ćwiczeń. Oracle Application Express (APEX) jest udostępniany przez Oracle Academy członkom instytucjonalnym i ich studentom do eksperymentowania w chmurze. Aby wspierać laboratoria i praktyczne zastosowania w programie nauczania, który obejmuje Podstawy baz danych, Projektowanie i programowanie baz danych z użyciem SQL, Programowanie z PL/SQL i Podstawy tworzenia aplikacji w Oracle Application Express, Oracle Academy oferuje edukatorom dedykowaną instancję zaprojektowaną specjalnie do użytku w klasie dla maksymalnie 99 studentów. Aby uzyskać dostęp do Oracle APEX poprzez Oracle Academy, użytkownicy muszą zalogować się do Member Hub. Jeśli nie są członkami, aby uzyskać dostęp do wszystkich materiałów i otrzymywać inne korzyści, użytkownicy powinni zarejestrować się jako członkowie instytucjonalni za darmo. Przejdź do tej strony https://academy.oracle.com/en/oa-web-overview.html.

1.3 Pytania

1. Jakie są etapy cyklu tworzenia aplikacji i jakie działania dotyczą trzeciego etapu?
2. Która z możliwości rozpoczęcia pracy z Oracle Application Express to najlepsze rozwiązanie dla początkujących?
3. Jakie są korzyści z używania instancji APEX w Oracle Cloud Infrastructure?

1.4 Answers

1. Etapy cyklu tworzenia aplikacji to: (i) planowanie (definiowanie problemu i projektu), (ii) projektowanie aplikacji, (iii) budowa aplikacji, (iv) implementacja. Działania trzeciego etapu (budowa aplikacji) to: projekt bazy danych, opracowanie (prototyp) aplikacji i testowanie.
2. Zalecenie dla początkujących jest przeznaczone do korzystania z instancji lokalnej lub apex.oracle.com lub Oracle Academy.
3. Oracle Cloud Infrastructure (OCI) oferuje programistyczne aplikacje APEX wymagające niewielkiej ilości programowania, działające w infrastrukturze Autonomous, która jest wstępnie skonfigurowana i gotowa do użycia. OCI zapewnia elastyczną skalowalność, bezpieczeństwo, wysoką dostępność i globalny dostęp za pośrednictwem regionalnych centrów danych w chmurze.

2. Jak przygotować bazę danych?

E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER

Załóżmy, że pracujesz w *Walmart* i *systemy baz danych* (DBS) nie zostały jeszcze wymyślone. Załóżmy, że użytkownik jest proszony o wdrożenie *aplikacji internetowej* która może przechowywać, pobierać, wizualizować i dalej przetwarzać każdą pojedynczą sprzedaż w każdym z 10.500 sklepów Walmart, wraz z informacjami na temat odpowiedzialnych działów, pracowników i ich stanowisk, które mogą być petabajtami danych. Zadanie, które oczywiście nie jest możliwe, ponieważ stoisz przed licznymi wyzwaniami:

- Ile plików i dysków jest potrzebnych do przechowywania danych?
- Jak znaleźć i pobrać dane?
- Jak zapewnić odpowiedni czas reakcji?
- Jak zezwolić na modyfikacje i współbieżny dostęp do danych?
- Jak zapobiec nieautoryzowanemu dostępowi do danych?

Jeśli jako podstawę aplikacji internetowej zostałaby użyta struktura DBS (zob. rys. 2.1), wyzwania te są możliwe do zarządzania, ponieważ DBS zapewnia odpowiednie funkcje do radzenia sobie z nimi.



Rysunek 2.1: DBS jako podstawa aplikacji internetowych.

DBS obejmuje:

1. oprogramowanie pomagające w zarządzaniu dużymi zbiorami danych, które mogą być używane przez aplikację internetową o nazwie Database Management Systems (DBMS) (DBMS)
2. faktyczne przechowywanie samych danych, zwane (), które zapewnia warstwę DB, na której opiera się aplikacja internetowa.

Poniżej omówiono szczegółowo główne pojęcia potrzebne do tworzenia warstwy bazy danych, tj., *rozwój bazy danych*. Na samym końcu chodzi jednak o "*tworzenie odpowiednich tabel bazy danych, przechowywanie danych, manipulowanie nimi i kierowanie do nich zapytań*", ale są też pewne ważne koncepcje, które należy wprowadzić, aby osiągnąć ten cel.

W szczególności rozpoczniemy od sekcji 2.1 proces modelowania danych do przechowywania poprzez wprowadzenie dwóch różnych *poziomów abstrakcji* , obejmujących *model logiczny* and the *model fizyczny* w kontekście *modelu relacyjnego*[1]. Dalej w sekcjach 2.2, specyficzne dla DBS mechanizmy i narzędzia do *tworzenia struktury danych w DBS*, zgodnie z zaleceniami procesie modelowania, skupiamy się na. Na tej podstawie sekcja 2.3 zajmuje się kwestiami przechowywania i manipulacji danymi, a sekcja ma na celu omówienie zapytań przechowywanych danych. Na koniec, sekcja 2.5 zawiera podsumowanie wszystkich opcji i odpowiednich narzędzi, które są dostępne w celu zbudowania warstwy DB aplikacji internetowej.

We wszystkich sekcjach, wykorzystywany jest *ciągły przykład*, zainspirowany przedstawionym powyżej przypadkiem użycia Walmartu, podkreślającym jego aspekt *Zasoby ludzkie (HR)* w kontekście *pracowników*, *departamentów* and *stanowisk*. Jest to standardowy przykład, który ORACLE wykorzystuje w celach demonstracyjnych od pierwszej wersji swojego systemu baz danych (DBS) w 1979 roku.

2.1 Jak modelować dane?

Pierwszym krokiem podczas realizacji DB-Layer aplikacji internetowej jest skoncentrowanie się na rzeczywistych danych, które powinny być przechowywane. Dane te mogą być dość wyczerpujące, jak widzieliśmy w powyższym przykładzie Walmart. Warto najpierw pochylić się nad danymi na bardziej abstrakcyjnym poziomie, jeszcze nie biorąc pod uwagę wszystkich szczegółów wdrożenia na niskim poziomie DBS.Jest to omówione w sekcji 2.1.1 w ramach omawiania logicznego modelu danych. Następnie, na bardziej konkretnym przykładzie, a zatem specyficznym dla DBS poziomie, model relacyjny został wprowadzony w sekcji 2.1.2.

2.1.1 Model logiczny

Motywacja. *Logiczny model danych* opisuje rzeczy z *świata rzeczywistego* (tj. *obszar problemów*)), na temat których organizacja, taka jak Walmart, chce zbierać i przetwarzać dane. Modele logiczne są przedstawiane wizualnie za pomocą diagramów, w praktyce często jako tzw. *diagramy relacji między jednostkami (ER)*, bez uwzględniania specyfiki konkretnego systemu bazy danych (DBS). Korzyści wynikające z tej wizualnej reprezentacji są liczne, na przykład zwiększają zrozumienie i ułatwiają komunikację w ramach zespołu deweloperskiego oraz z klientem.

Dostępnych jest wiele różnych narzędzi do modelowania diagramów ER. Skoncentrujemy się wyłącznie na używaniu narzędziu *ORACLE SQL Developer Data Modeler (ODM)*, only. Zob. rys. 2.2 przedstawia diagram ER dla naszego przykładu dotyczącego Zasobów Ludzkich (HR), który zostanie dokładniej wyjaśniony poniżej.

[1] Należy zauważyć, że do realizacji modelu fizycznego można zastosować nie tylko model relacyjny, na którym skupiono się w tym rozdziale, ale dostępne są także inne formaty reprezentacji danych (tzw. „modele danych"), takie jak *obiektowy model danych* lub *modele danych NoSQL*.



Rysunek 2.2: Logiczny model dla Zasobów Ludzkich HR przy użyciu ER-Diagram.

Ten sam przykład przedstawiono na rysunkach 2.3, 2.4 i 2.5, tym razem jednak przy wykorzystaniu narzędzia ODM ORACLE. Wszystkie przedstawione pojęcia są opisywane w celu wyeksponowania najważniejszych notacji graficznych i symboli.

ER-Diagrams - obiekty, atrybuty, powiązania. *Wyresy ER* opisują pojęcia, które są przedmiotem zainteresowania i obejmują (i) *podmioty* (t.j.: Departments, Employees, Jobs) (ii) *atrybut* (np., "departmentName" (nazwę działów) i "locationName" (nazwę lokalizacji) departamentów, oraz "lastName" (nazwisko), and "salary" (płacę) pracowników)) ich cechy, oraz (iii) *relacji* (np. dział składa się z kilku pracowników) między podmiotami.

Podmioty. *Podmioty* (inaczej mówiąc obiekty) mają przypisaną unikatową nazwę i są wizualnie reprezentowane jako zaokrąglony prostokąt. W przypadku nazewnictwa obiektów, formy liczby mnogiej są używane, jednakże formy liczby pojedynczej są ostatnio bardziej preferowane (a także standaryzowane przez ISO 11179-5).

Atrybuty. Atrybuty są zawsze powiązane z podmiotami, tzn. nie mogą istnieć niezależnie. W przeciwieństwie do podmiotów *atrybut* jest "*autonomiczny*", co oznacza, że nie może być dalej ustrukturyzowany jak obiekt, który jest przedstawiony jako osobna fraza rzeczownika w liczbie pojedynczej. Dla każdego atrybutu można przypisać pewną *domenę* (tak zwany *typ danych*), oznaczającą "nature" dozwolonych wartości. Najczęściej są to cyfry (`NUMERIC`), pojedyńcze znaki (`VARCHAR`) i data/godzina (`DATE/TIME`).

Oprócz scharakteryzowania obiektów, niektóre atrybuty mogą mieć *jednoczesne pochodzenie każdego z podmiotów* (np. konkretnego pracownika lub konkretnego działu). Takie atrybuty identyfikujące działają jak tzw. "*klucz głównego (PK)*". W przypadku działu identyfikator można zdefiniować jako PK (który powinien być numeryczny, np. "departmentID") lub alternatywnie jako kombinację istniejących atrybutów, takich jak "departmentName" and "location", tworząc tzw. "*composite PK*". Wskazane jest zdefiniowanie PK dla każdego podmiotu, gdyż w przeciwnym razie przy wyszukiwaniu danych nie będzie można rozróżnić np. pracowników lub działów o tej samej nazwie. Zatem każda wartość atrybutu PK musi być unikalna i oczywiście musi istnieć.

Na koniec należy wspomnieć, że oprócz domen i kluczy głównych można także określić inne tzw. *ograniczenia* dotyczące wartości danych dla konkretnych atrybutów, np. `NOT NULL`, "obowiązkowe" wymaga istnienie wartości (np. dla atrybutu wynagrodzenia).

Relacja. *Relacja* łączy (najczęściej) dwie różne podmioty, jest wizualnie przedstawiana jako linia między prostokątami encji i nazwana za pomocą frazy czasownikowej nad linią. W praktyce najczęściej używane są trzy różne rodzaje relacji, rozróżniane przez *liczbę wystąpień podmiotów*, które mogą uczestniczyć w relacji z każdej strony. Liczbę wystąpień podmiotów nazywamy "*kardynalnością*"(reprezentujuje ona kolejny rodzaj ograniczenia danych, które można określić):

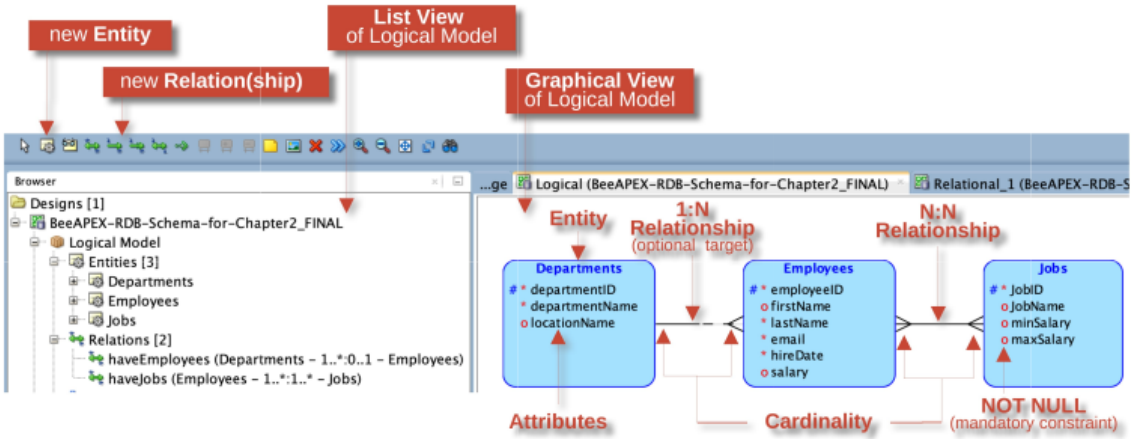
- **Jeden z wielu ("1:N"):** Jedna instancja podmiotów może być powiązana z wieloma instancjami innych podmiotów po "wielu stronach" relacji (np. jeden dział ma wielu pracowników, czyli "haveEmployee") i podmiot po "wielu stronach" relacji (employee) może być powiązany tylko z jednym podmiotem po "jednej stronie" (department). Taka relacja nazywa się "*Source-Target*" lub "*Parent-Child*" *relacja*, przy czym jednostka źródłowa po "jednej stronie" pełni rolę "Parent", a dowolna liczba docelowych jednostek po stronie wielu działających jako jej "Child". W powyższym przykładzie dział jest jednostką nadrzędną, a pracownicy są

jednostkami podrzędnymi.

- **Wiele z wielu (“M:N”):** Jedna instancja podmiotu źródłowego może być powiązana z *wieloma instancjami podmiotów docelowych*, a jedna instancja encji docelowej może być powiązana z *wieloma instancjami encji źródłowych instancje*. Przykładem mogą być pracownicy mający pracę (tj. relację “haveJobs”), ponieważ pracownik może mieć więcej niż jedną pracę w tym samym czasie, a określoną pracę prawdopodobnie wykonuje wielu pracowników. Należy zauważyć, że w miarę postępów w modelowaniu baz danych relacja “M:N” zawsze będzie generować dwie relacje “1:N”.
- **Jeden do jednego (“1:1”):** Wreszcie, jedna instancja podmiotów może być również powiązana z *instancją pojedyńczych podmiotów* na obu końcach relacji i odwrotnie (np. osoba “livesAt” (mieszka pod) konkretnym adresem, adres ten jest adresem tylko tej konkretnej osoby). Należy jednak zauważyć, że temu rodzajowi relacji często sprzyja bardziej zwarta reprezentacja w obrębie tylko jednego podmiotu (np. osoby podmiotu obejmują również atrybuty adresu, takie jak “street” i “city”), dlatego nie jest on dalej rozważany w tym rozdziale.
- **Opcjonalność:** Dla każdego z tych trzech różnych rodzajów relacji możliwe jest również określenie na każdym końcu źródła i celu, czy *istnienie co najmniej jedna instancja podmiotów, czy nie*. Na przykład, jeśli zostanie utworzony nowy dział, może się zdarzyć, że w tym dziale nie zatrudniono jeszcze żadnych pracowników, co oznacza, że cel relacji “haveEmployees” musi zostać zdefiniowany jako opcjonalny. Jest to również przedstawione linią przerywaną w narzędziu ODM.

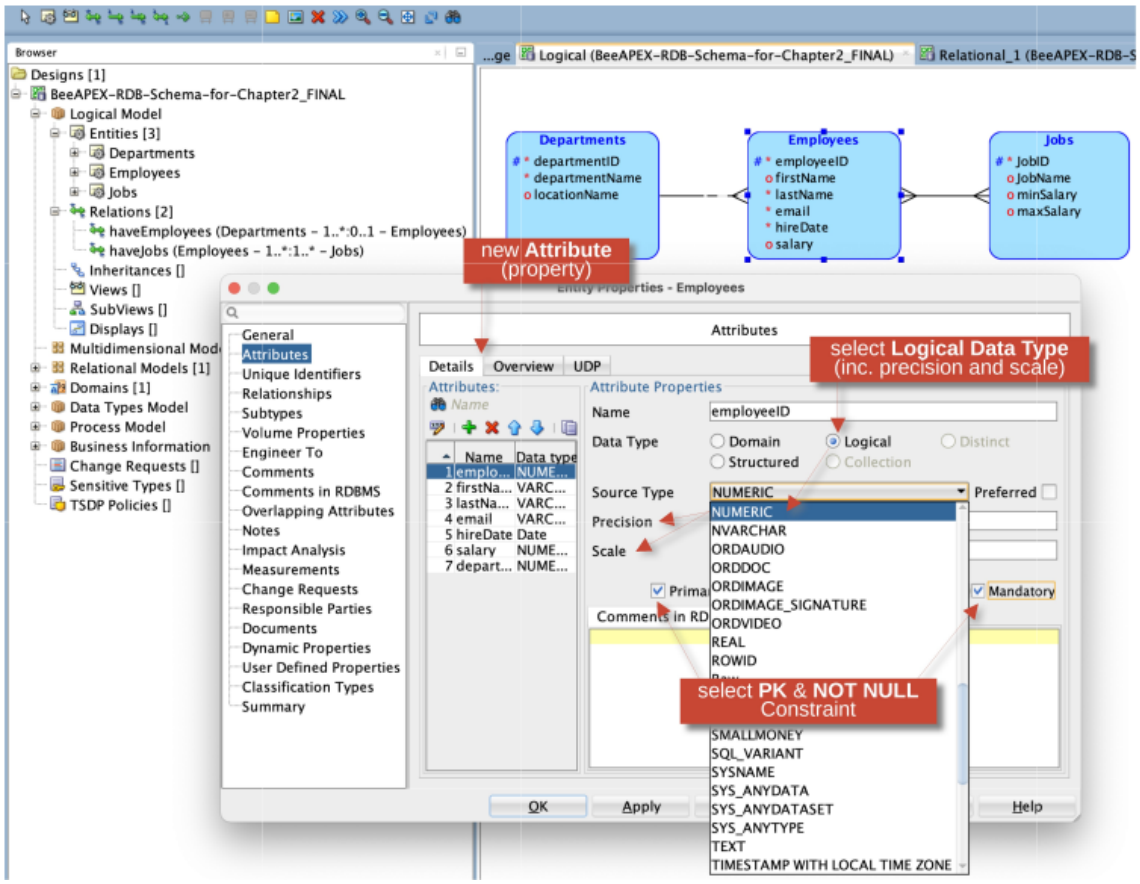
Relacje są określane w narzędziu ODM po prostu jako “ralations”, których nie należy mylić z “modelem relacyjnym”, który jest w rzeczywistości synonimem “tabel” (patrz sekcja 2.1.2). W związku z tym dalej stosujemy bardziej powszechny termin “relationship”.

Elementy GUI ODM do zarządzania diagramami ER. Poniższe rysunki dają pierwsze wrażenie najważniejszych elementów GUI ODM. Po pierwsze rysunek 2.3 ilustruje sposób definiowania nowych podmiotów i nowych relacji, a także wizualizację modelu logicznego w postaci widoku listy i widoku graficznego. W widoku graficznym najważniejsze notacje graficzne i symbole z adnotacjami.



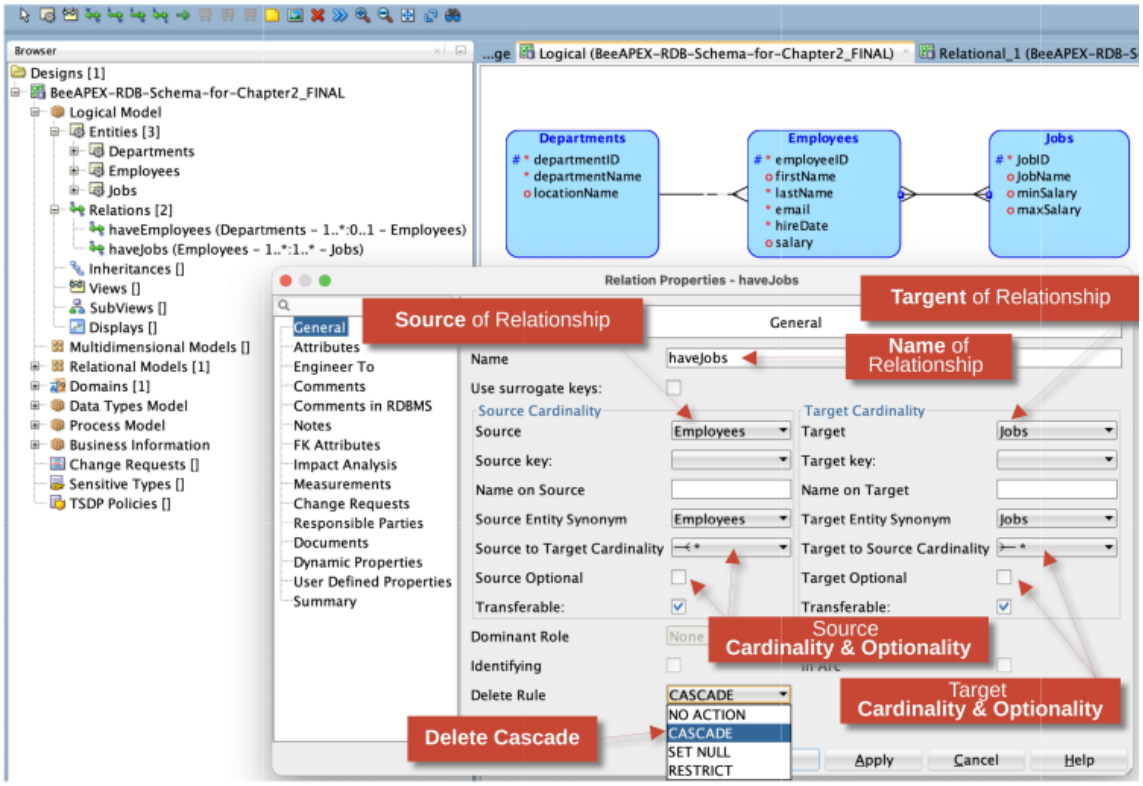
Rysunek 2.3: Zarządzanie modelem logicznym za pomocą ODM.

Rysunek 2.4 przedstawia okno dialogowe służące do definiowania atrybutów, które można otworzyć poprzez dwukrotne kliknięcie określonego podmiotu.



Rysunek 2.4: Zarządzanie atrybutami za pomocą ODM.

Rysunek 2.5 przedstawia okno dialogowe do definiowania relacji, które można uruchomić, klikając dwukrotnie określoną linię reprezentującą relację.

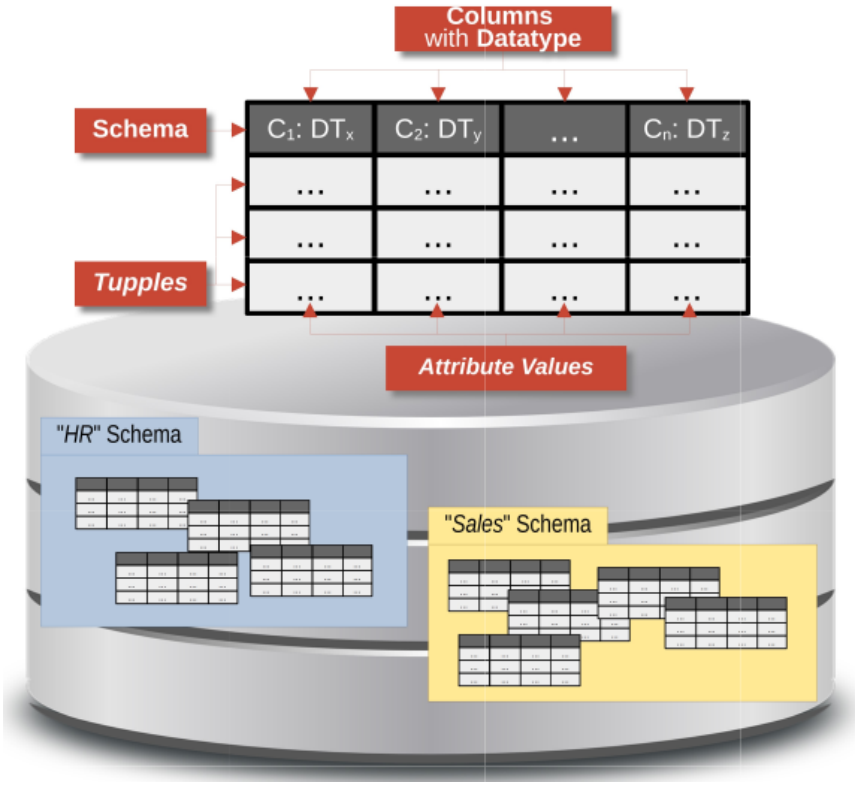


Rysunek 2.5: Zarządzanie relacjami przy pomocy ODM.

2.1.2 Model relacyjny

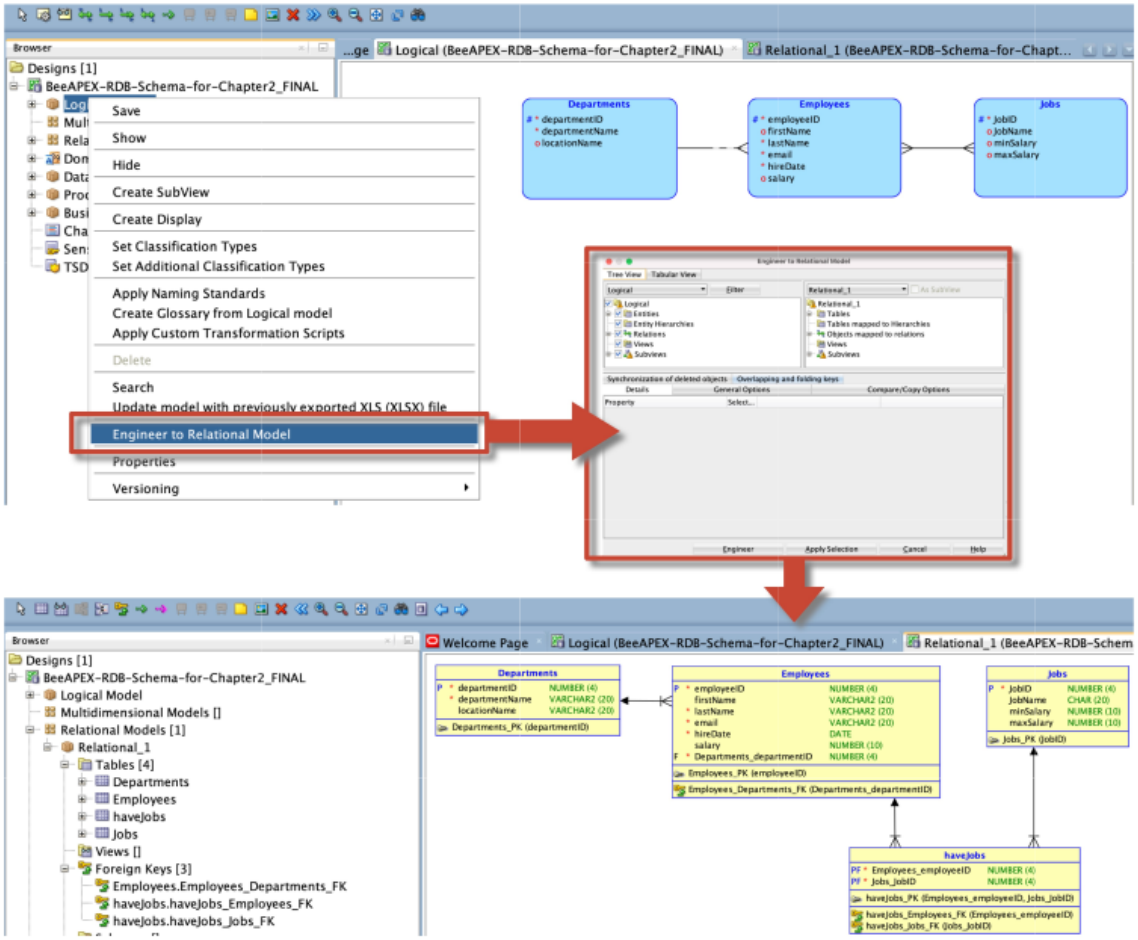
Schemat DB. Model logiczny w kategoriach diagramu ER niezależnego od konkretnego DBS jest podstawą do wyprowadzenia specyficznej dla DBS struktury DB zwanej "*schematem bazy danych*". W ramach konkretnego DBS możliwe jest tworzenie i przechowywanie dowolnej liczby różnych schematów dla różnych użytkowników i/lub domen, np. jednego do zarządzania danymi HR i innego schematu do zarządzania danymi o sprzedaży produktów. Schemat działa podobnie do folderu w systemie plików, umożliwiając grupowanie danych w określonym celu.

Relacyjny schemat bazy danych. Schemat specyficzny dla DB to tak zwany *Relacyjnych schemat DB (RDB)*, który wykorzystuje *relacyjny model (RM)* jako podstawowy formalizm do opisu struktury danych. RM jest dość prosty, ponieważ reprezentuje schemat DB jako zbiór *relacji* tzw. *tabel*, które przypominają koncepcję podmiotów. Tabele wywodzą się z podmiotów w prosty sposób. Analogicznie do podmiotów, tabela składa się z nazwy, *kolumny* (przypominającej *atrybuty*), ich typów danych i dalszych (opcjonalnych) ograniczeń. Wszystkie tabele tworzą *schemat RDB* (zob. rys. 2.6). Schemat tabeli jest dalej "nośnikiem", który przechowuje rzeczywiste dane, podczas gdy każdy wiersz w tabeli nazywany jest "*tuple*", który składa się ze zbioru powiązanych wartości danych przechowywanych w kolumnach, reprezentując w ten sposób rzeczywisty świat *instancji podmiotów* pod względem *relacji* (*tabelarycznych*) danych.



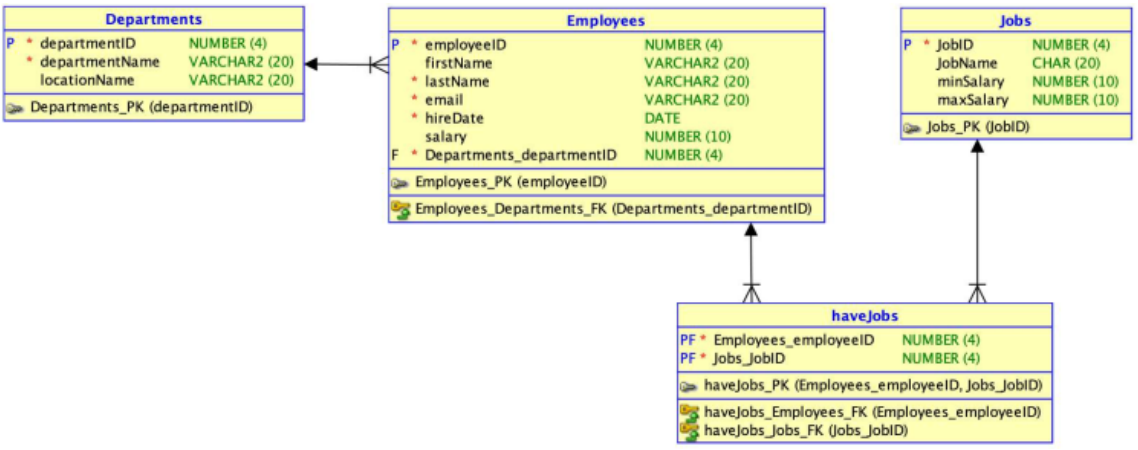
Rysunek 2.6: Schematy RDG i schematy tabeli.

Warto zauważyć, że *ODM* umożliwia automatyczne generowanie schematu RDB z modelu logicznego w kategoriach tabel, odzwierciedlając w ten sposób podmioty naszego modelu logicznego oraz wszystkie atrybuty i ograniczenia każdego podmiotu. Odbywa się to za pomocą pozycji menu *"Engineer to Relational Model"*, jak pokazano na rysunku 2.7. Ostateczny schemat RDB jest również wizualizowany graficznie przy użyciu pewnego rodzaju formalizmu diagramu ER, opartego na nieco innej, bardziej "zorientowanej na tabelę adnotacji" niż diagram ER modelu logicznego, pokazując np. również typy danych wszystkich atrybutów.



Rysunek 2.7: Proces generownia schematu RDG z modelu logicznego.

Skupmy się teraz na naszym przykładowym modelu relacyjnym przedstawionym na rysunku 2.8.



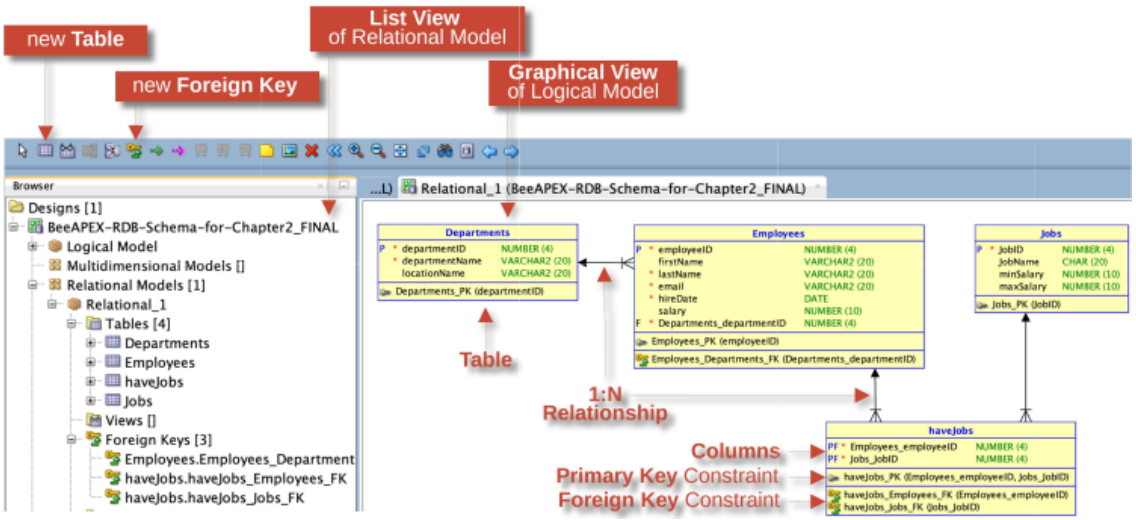
Rysunek 2.8: Automatycznie generowany schemat RDB.

Po pierwsze, podczas procesu generowania modelu relacyjnego z modelu logicznego, oprócz przekształcania istniejących encji i atrybutów w tabele i odpowiadające im kolumny, PK są automatycznie dodawane do każdej tabeli i otrzymują nazwę (np. “Departments_PK (departmentID)”. Ponadto relacje między podmiotami w modelu logicznym są automatycznie reprezentowane w

tabelach przy użyciu tak zwanego "*Foreign-Key (FK)*" *concept* w następujący sposób:

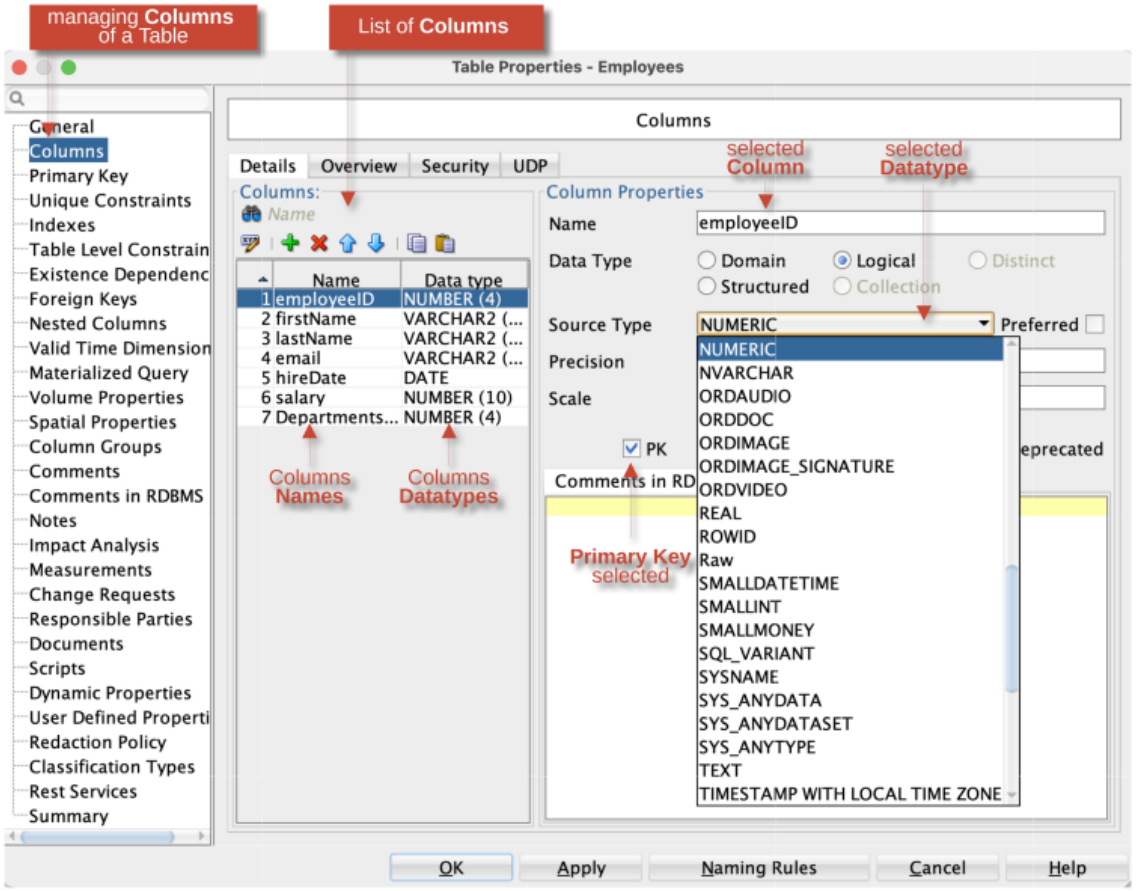
- **Relacja One-to-Many (1:N):** W tabeli po stronie "Many-side" relacji (tj. w tabeli Child), dodatkowy atrybut jest wprowadzany i deklarowany jako FK, działając jako rodzaj odniesienia do wartości PK relacji po stronie "One-side" (tj. w tabeli "Parent"). Jeśli istnieje np. pracownik o identyfikatorze PK 4711, wówczas w atrybucie FK powiązanego działu musi być również obecna wartość 4711. W związku z tym atrybut FK musi mieć ten sam typ danych co PK. W naszym przykładzie automatycznie generowany jest FK "Departments_departmentID" oraz FK o nazwie "Employees_Departments_FK". W konsekwencji, za każdym razem, gdy dane w ramach tych dwóch relacji są manipulowane w DBS, FK zapewnia *spójność* pomiędzy tymi relacjami, tj. poprawne dane relacji. Na przykład, jeśli nowy pracownik jest wstawiany do tabeli Pracownicy, DBS zapewnia, że jest on również przypisany do istniejącego działu. Jeśli spróbują Państwo usunąć dział, DBS domyślnie odrzuci tę operację, jeśli nadal istnieją powiązani pracownicy. Zmiana tego domyślnego zachowania jest możliwa poprzez użycie tak zwanego "*Cascade-Delete Option*" podczas określania FK, co spowoduje automatyczne usunięcie (być może "zwolnienie") wszystkich pracowników powiązanych z usuniętym działem.
- **Relacja wiele do wielu (M:N):** Relacja wiele-do-wielu między dwiema encjami jest realizowana poprzez wprowadzenie kolejnej, trzeciej tabeli, działającej jako swego rodzaju "*mapping table*", która przechowuje głównie odniesienia do każdego z PK powiązanych encji w kategoriach dwóch atrybutów FK. Jeśli chodzi o nasz przykład z pracownikami i miejscami pracy, automatycznie generowana jest trzecia relacja o nazwie "haveJobs", która jest teraz odpowiedzialna za zarządzanie prawidłowymi powiązaniami między pracownikami i ich odpowiednimi miejscami pracy. PK tej tabeli mapowania jest złożone, składające się z PK dwóch powiązanych tabel, tj. "haveJobs_PK(Employees_employeeID, Jobs_jobID)". Każdy atrybut tego PK jest również używany do zdefiniowania odpowiedniego FK, z których każdy odwołuje się do PK odpowiedniej powiązanej tabeli (tj. "haveJobs_Employees_FK (Employees_employeeID)" i "haveJobs_Jobs_FK(Jobs_JobID)"). Należy zauważyć, że ta tabela mapowania może również mieć dowolną liczbę dodatkowych atrybutów, np. "startDate" and "endDate" of a job.
- **Opcjonalność:** Domyślnie, dla każdej z tych relacji, istnienie co najmniej jednego źródła i jednej instancji docelowej jest opcjonalne. Jeśli jest to obowiązkowe, wówczas dalsze ograniczenie NOT NULL musi być powiązane z odpowiednim atrybutem FK. Biorąc pod uwagę nasz przykład, wszystkie trzy atrybuty FK wymagają ograniczenia NOT NULL, ponieważ (i) pracownicy muszą być powiązani z określonym działem (ii), a także z co najmniej jednym zadaniem oraz (iii) zadanie musi być powiązane z co najmniej jednym pracownikiem.

Elementy GUI ODM do zarządzania RM. Poniższe rysunki przedstawiają przegląd najważniejszych elementów GUI *ODM* do zarządzania schematem RDB, które są dość podobne do tych do zarządzania modelem logicznym. Po pierwsze, rysunek 2.9 ilustruje sposób definiowania nowych tabel i nowych FK, a także wizualizację RM w postaci widoku listy i widoku graficznego. W widoku graficznym najważniejsze notacje graficzne i symbole są opatrzone adnotacjami.



Rysunek 2.9: Zarządzanie kolumnami poprzez elementy GUI.

Rysunek 2.10 przedstawia okno dialogowe do definiowania kolumn, które można uruchomić, klikając dwukrotnie określoną tabelę. Proszę zauważyć, że to okno dialogowe w dużej mierze przypomina okno dialogowe zarządzania atrybutami przedstawione na rysunku 2.4.



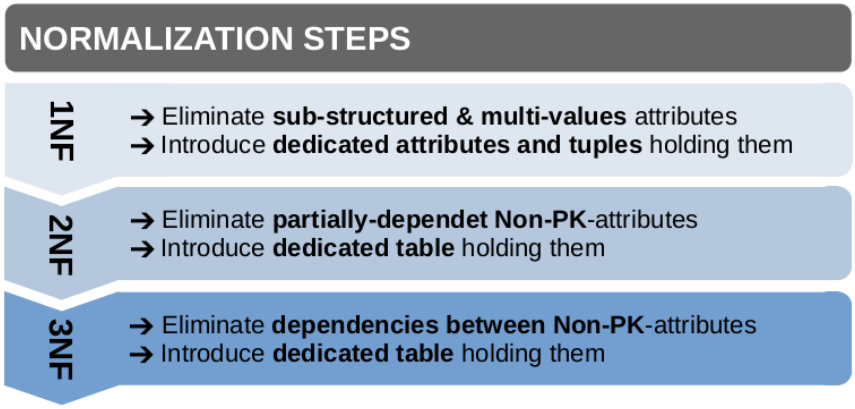
Rysunek 2.10: GUI-Elements for Managing the RM.

Schemat RDB niezależny od aspektów fizycznej pamięci masowej. Chociaż schemat RDB oparty na modelu relacyjnym jest już bardziej konkretny i specyficzny dla DBS niż model logiczny

(np, "wdrażanie" relacji opartych na koncepcji FK), należy zauważyć, że nadal ukrywa on złożoność podstawowych mechanizmów przechowywania DBS (aka. "*physical*" *level*), np. ile plików jest używanych do przechowywania tabel lub na których serwerach mają być przechowywane dane. Kolejną dużą zaletą tej abstrakcji od aspektów fizycznego przechowywania danych jest to, że można realizować nowe schematy RDB lub zmieniać istniejące bez konieczności dbania o podstawową fizyczną organizację danych.

2.1.3 Normalizacja schematu bazy danych RDB

Schemat RDB, który został wyprowadzony z logicznego, może być często dalej ulepszany w celu lepszego osiągnięcia pewnych "kryteriów jakości", które są powszechne w DBS, będąc z jednej strony najlepiej dostosowanym do "dostępu do odczytu danych" w zakresie zapytań DB, ale z drugiej strony zapewniając również odpowiednią podstawę do "manipulacji danymi" w zakresie wstawiania, aktualizowania i usuwania danych. Chociaż są to do pewnego stopnia sprzeczne cele, istnieje dobrze znana koncepcja dostępna dla RDBS pod względem tak zwanej "teorii normalizacji", która opisuje systematyczny proces trzech kroków o nazwie "*pierwsza postać normalna*", "*druga postać normalna*" i "*trzecia postać normalna*", prowadzący do ulepszonych schematów RDB (zob. rysunek 2.11 dla przeglądu). Poniżej, te trzy normalne formy są opisane bardziej szczegółowo, przy użyciu niewielkich odmian naszego działającego przykładu.



Rysunek 2.11: Etapy normalizacji - przegląd.

Pierwsza Forma Normalna. Pierwsza postać normalna wymaga, aby każdy atrybut tabeli był "atomic", co oznacza, że ani (i) *podstruktury* nie są dozwolone (np. jeśli zamiast naszego prostego atrybutu "locationName", alternatywny atrybut "location" zawierałby wartości dotyczące ulicy, kodu pocztowego, miasta i kraju), ani (ii) *wielowartości* nie są dozwolone (np. atrybut "location" może również zawierać kilka adresów określonego działu. Podstruktury w ramach wartości można *wyeliminować*, po prostu reprezentując podstrukturę za pomocą dedykowanych atrybutów (np. dla każdej części adresu), po prostu używając własnej "tuple" dla każdej z wartości (tj. umieszczając każdy adres w osobnym wierszu).

Jedną z głównych *zalet* pierwszej postaci normalnej jest to, że zapytania o dane mogą być bardziej szczegółowe, ponieważ każda część adresu, jak również każda "tuple" adresu może być dostępna oddzielnie.

Druga postać normalna. Druga postać normalna wymaga (i) aby tabela była w *pierwszej postaci normalna* oraz (ii) w przypadku złożonego PK, tj. składającego się z dwóch lub więcej atrybutów, wszystkie inne wartości atrybutów niebędące PK są *wyłącznie identyfikowane przez cały złożony PK*. Proszę na przykład założyć, że tabela Departments zawiera złożony PK z atrybutami "departmentName" i "locationName". Ten PK poprawnie określa wartość atrybutu niebędącego PK "annualRevenue", ponieważ naturalnie zakładamy, że wartość "annualRevenue" zależy zarówno od

działu, jak i jego lokalizacji. Wartości innego atrybutu nie będącego PK "locationAddress" zależą już jednak od części PK, a mianowicie od nazwy lokalizacji. Aby *wyeliminować* tę *częściową zależność*, musimy *rozdzielić* tabelę Departments, wydzielając informacje o lokalizacji do oddzielnej tabeli "Locations" i połączyć obie tabele za pomocą FK.

Jedną z głównych *zalet* drugiej postaci normalnej jest to, że zapobiega ona *nadmiarowemu przechowywaniu danych*, ponieważ w naszym przypadku określony adres jest przechowywany tylko raz i może być ponownie wykorzystany przez różne działy. Zmniejsza to nie tylko przestrzeń dyskową, ale przede wszystkim zapobiega błędnym danym, tak zwanym "*anomalią*" jeśli np. pewne manipulacje danymi nie są przetwarzane na żadnych zduplikowanych danych (np. w przypadku przeniesienia kilku działów pod inny adres, adres musi zostać zmieniony tylko raz). Wreszcie, dobrą rzeczą jest to, że mając tabele tylko z niezłożonymi PK (np. numeryczny identyfikator dla każdej tabeli), schemat RDB "automatycznie" spełnia drugą postać normalną, ponieważ częściowe zależności nie mogą istnieć.

Trzecia forma normalna Trzecia postać normalna wymaga (i) aby tabela znajdowała się w *drugiej postaci normalnej* i (ii) każdy atrybut *niebędący PK był jednoznacznie identyfikowany tylko przez PK*, a nie przez jakikolwiek inny atrybut niebędący PK, co prowadzi do *zależności przechodnich*. Na przykład, jeśli tabela Departments zawierałaby oprócz PK employeeID i locationName również locationAddress, wówczas locationAdress byłby już jednoznacznie identyfikowany przez locationName (jeśli założymy, że każda locationName jest powiązana z innym adresem), co nie jest dozwolone przez trzecią postać normalną. Aby wyeliminować tę sytuację, informacje o lokalizacji muszą zostać uwzględnione w oddzielnej tabeli Locations i połączone za pomocą odpowiedniego FK z tabelą Departments.

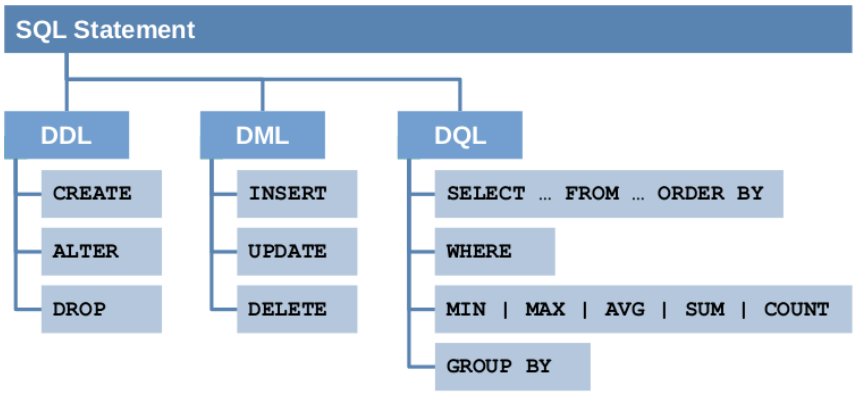
Zalety są takie same, jak w przypadku drugiej postaci normalnej - zmniejszenie nadmiarowości, a tym samym potencjalnych anomalii w przypadku manipulacji danymi.

2.2 Jak zarządzać schematami RDBS - SQL-DDL

W poprzedniej sekcji, schemat RDB został graficznie zamodelowany na różnych poziomach abstrakcji w sposób tekstowy *low-code development fashion*. Teraz nadszedł czas, aby omówić, w jaki sposób te modele graficzne mogą być implementowane "programowo" w RDBS, tak aby naprawdę mogły być wykorzystywane do zarządzania rzeczywistymi danymi aplikacji internetowej. Poniżej znajdą Państwo odpowiedzi na następujące pytania:

- Jak doradzić DBS, aby *utworzył wymagane tabele*, najlepiej z modeli graficznych i czy mamy jakieś inne alternatywy, jeśli takie modele graficzne nie istnieją?
- Jak wygląda *"język programowania" DBS*, który jest w stanie tworzyć tabele?

SQL = DDL | DML | DQL. Aby najpierw odpowiedzieć na drugie pytanie, językiem, który jest używany do tego, a także do innych celów. Język programowania DBS jest *znormalizowany strukturalny język zapytań (SQL)*. SQL zapewnia szeroki zakres różnych *"instrukcji programowania DBS"* które można, zgodnie z ich przeznaczeniem, pogrupować w różne kategorie (zob. rys Figure 2.12), z których najważniejsze to (i) instrukcje do zarządzania schematem RDB (np, tworzenie, zmienianie i usuwanie tabel), podsumowane jako *DDL (Data Definition Language)*, na którym koncentruje się ta sekcja, (ii) instrukcje do manipulowania danymi w schemacie (np. wstawianie, aktualizowanie i usuwanie danych w tabelach) zwane *DML (Data Manipulation Language)* które zostaną omówione w sekcji 2.3 i wreszcie (iii) instrukcje do wysyłania zapytań do danych zwane *DQL (Data Query Language)*, omówione dalej w sekcji 2.4.



Rysunek 2.12: Kategorie instrukcji SQL.

Pięć opcji tworzenia tabel. Zgodnie z narzędziami dostarczonymi przez ORACLE, istnieje pięć różnych opcji tworzenia tabel, tj. implementacji schematu RDB (zob. rys. 2.45 dla lepszego zrozumienia):

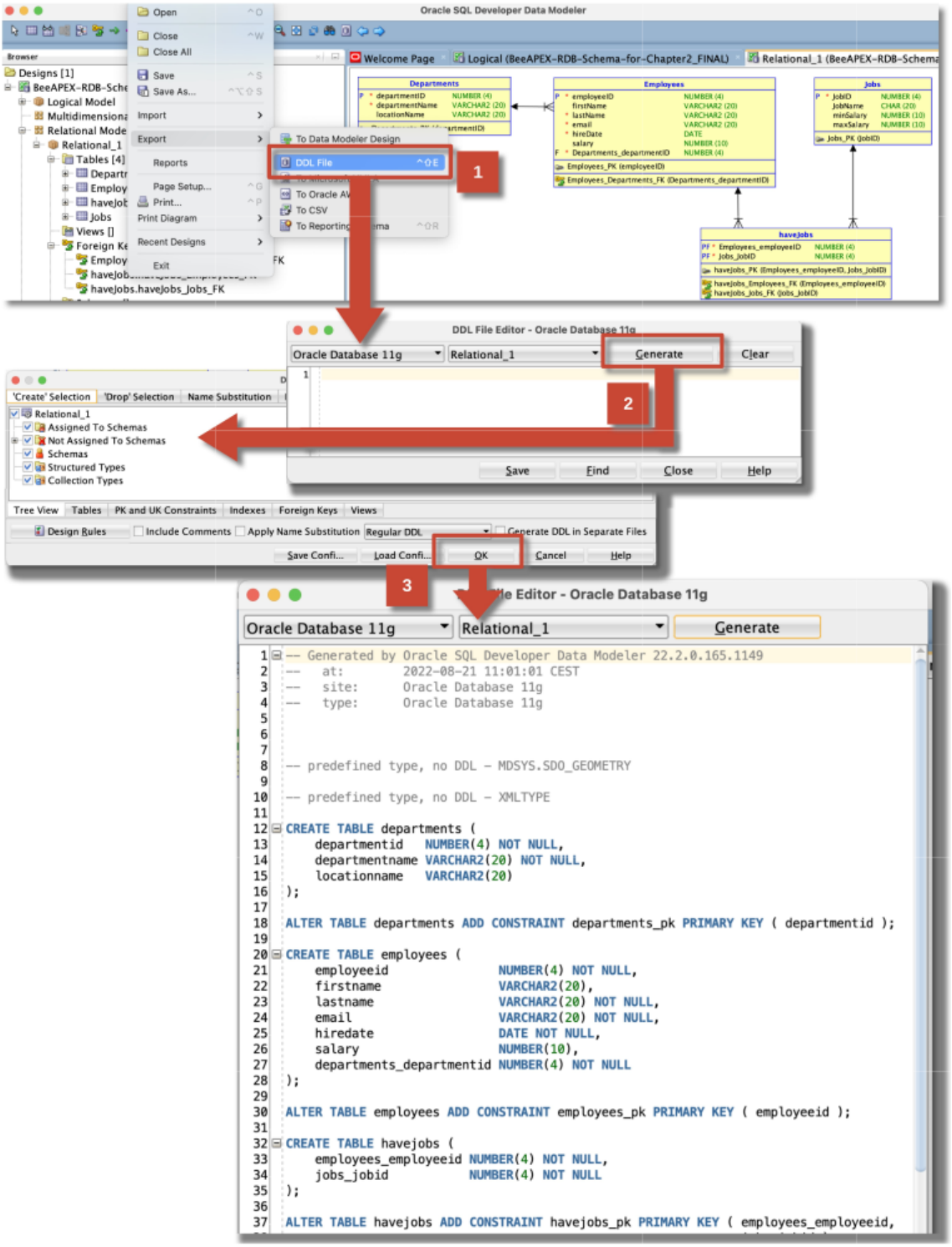
1. **Automatyczne generowanie tabel przy użyciu ODM.** Jest to opcja wymagająca najmniejszego wysiłku, ponieważ wystarczy zainicjować *automatyczne generowanie tabel* za pomocą kilku kliknięć przycisku *używając ODM* (zob. sekcję 2.2.1).
2. **Automatyczne generowanie tabel przy użyciu Quick SQL.** Inną alternatywą byłoby użycie pewnego rodzaju "skrótu-notacji" dla SQL, dostarczanego przez narzędzie ORACLES Quick SQL (zob. sekcję 2.2.2). Jest to idealny wybór, jeśli nie ma dostępnych graficznych (logicznych i relacyjnych) modeli schematu RDB lub jeśli konieczne jest szybkie testowanie.
3. **Ręczne tworzenie tabel przy użyciu SQL-DDL.** Oczywiście istnieje również możliwość ideogramatycznego określenia tabel za pomocą zwykłego języka SQL, co zapewnia korzyści polegające na tym, że każdy szczegół dotyczący specyfikacji tabeli można skonfigurować zgodnie z potrzebami i nie jest zależny od, czasami nieoptymalnych, automatycznych procesów generowania (zob. sekcję 2.2.3).
4. **Ręczne tworzenie tabel za pomocą przeglądarki obiektów ORACLE (OB).**Dla tych użytkowników, którzy nie znają SQL, Przeglądarka obiektów umożliwia tworzenie nowych tabel i ich modyfikowanie w razie potrzeby w sposób oparty na formularzach (zob. sekcję 2.2.4).
5. **Automatyczne tworzenie tabel przy użyciu ORACLE Data Workshop.** Istnieje jeszcze inna możliwość automatycznego tworzenia tabel, która może być jednak używana tylko wtedy, gdy istnieją już dane przechowywane w plikach zewnętrznych (np. w plikach EXCEL) przy użyciu ORACLE Data Workshop. Opcja ta nie jest dalej omawiana w tym rozdziale, ale raczej opisana bardziej szczegółowo w sekcji chapter 4.

Poniżej omówiono bardziej szczegółowo pierwsze cztery opcje.

2.2.1 Automatyczne generowanie tabel przy użyciu ODM

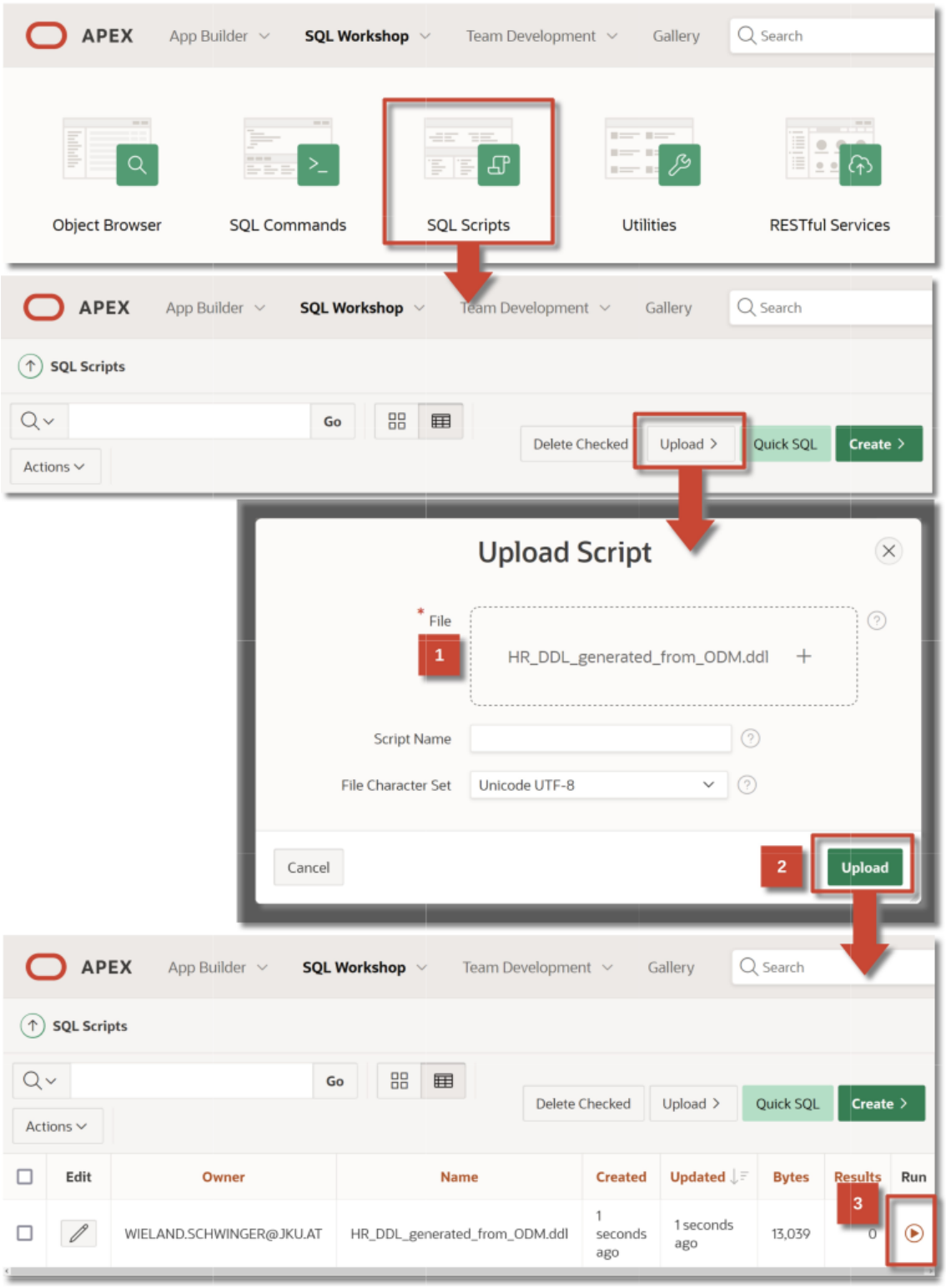
ODM to skrót od narzędzia o nazwie Oracle SQL Developer Data Modeler. Jako opcja generowania tabel przy najmniejszym wysiłku, tylko dwa kroki są niezbędne do utworzenia wymaganych tabel DB w przypadku, gdy schemat RDB został opracowany w *ODM*.

Najpierw należy wyeksportować modelowany schemat RDB do *skryptu SQL* (zob. rysunek 2.13, (kroki od 1 do 3), będącego w rzeczywistości prostym plikiem tekstowym, który zawiera wszystkie instrukcje SQL-DDL niezbędne do utworzenia tabel w RDBS. Proszę zauważyć, że składnia wygenerowanych instrukcji SQL jest opisana bardziej szczegółowo w sekcji 2.2.3.



Rysunek 2.13: Eksport schematu RDB do skryptu DDL

Po drugie, wygenerowany skrypt SQL musi zostać wykonany, aby doradzić RDBS, aby faktycznie utworzył wszystkie tabele jako puste w swojej przestrzeni dyskowej. W tym celu, jak pokazano na rys. 2.14, skrypt musi być załadowany przez ORACLE SQL Workshop (zob. 1-2) i po prostu wykonany (zob. 3).



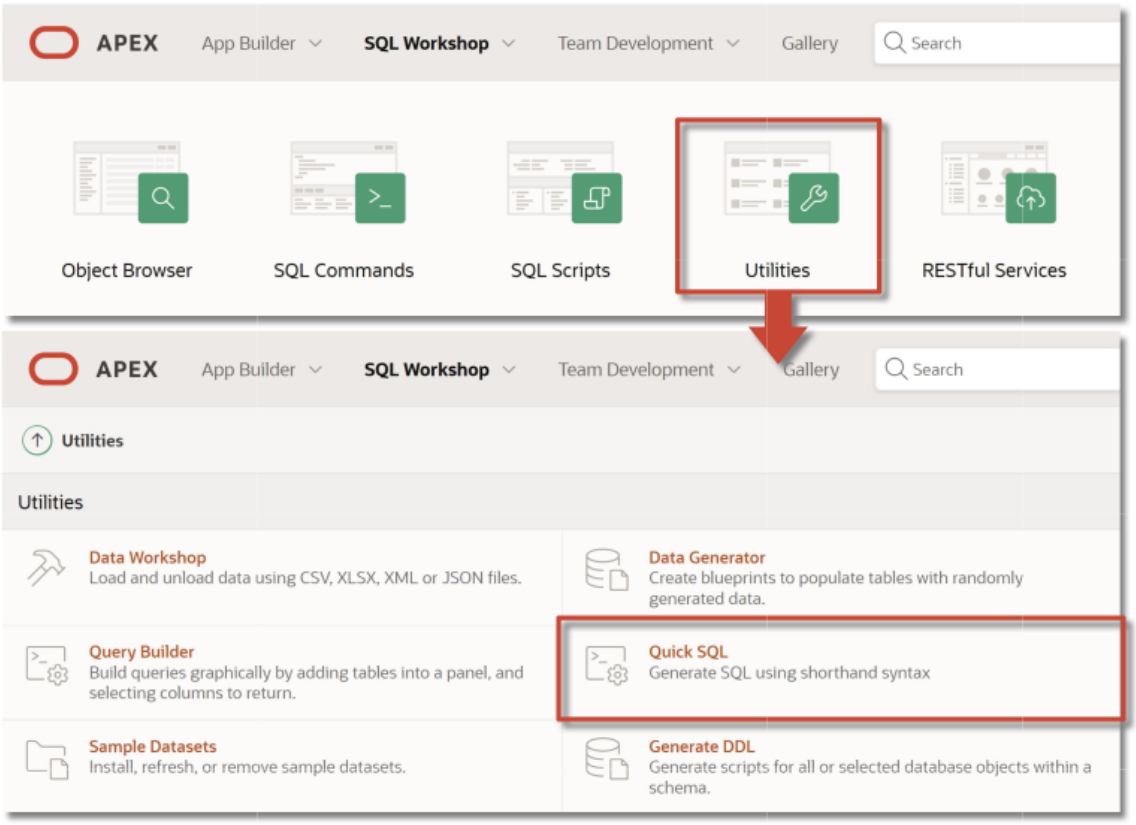
Rysunek 2.14: Przesyłanie i wykonywanie skryptu SQL.

2.2.2 Automatyczne generowanie tabel przy użyciu Quick SQL

Budując warstwę DB aplikacji internetowej, wysoce wskazane jest przestrzeganie różnych poziomów abstrakcji pod względem modeli logicznych i relacyjnych. Niemniej jednak, *szybki SQL* zapewnia prostą i intuicyjną alternatywę do tworzenia tabel, znacznie prostszą niż SQL, np. do celów szybkiego testowania. Quick SQL pozwala również na automatyczne wygenerowanie odpo-

wiedniego skryptu SQL. Po wygenerowaniu skryptu SQL można go dostosować i rozszerzyć, a następnie wykonać w celu utworzenia tabel. Należy jeszcze raz podkreślić, że chociaż Quick SQL został zaprojektowany w celu zmniejszenia wysiłku wymaganego do tworzenia tabel, zdecydowanie nie został zaprojektowany jako zamiennik modelowania danych (zob. sekcję 2.1).

Dostęp do edytora ORACLE Quick SQL można uzyskać poprzez "Utilities" w SQL Workshop (zob. rys. 2.15.



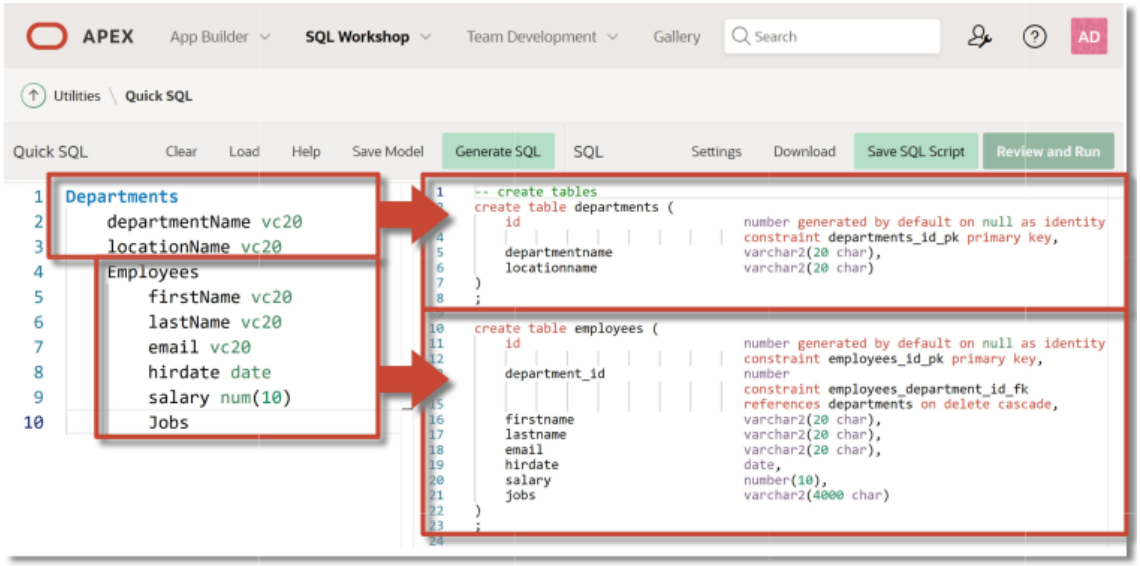
Rysunek 2.15: Dostęp do Quick SQL poprzez SQL Workshop

Pozostańmy teraz przy prostej składni dostarczanej przez Quick SQL, która zasadniczo wykorzystuje kilka zasad formatowania w celu określenia wymaganych generacji tabel:

- **Tabele nadrzędne:** Nazwy tabel nadrzędnych są wprowadzane bez wcięć. Nazwy tabel są automatycznie formatowane, zastępując spacje podkreśleniami
- **Atrybuty:** Nazwy atrybutów są wprowadzane z jednolitym wcięciem dwóch lub więcej spacji. Podobnie jak w przypadku nazw tabel, nazwy atrybutów są automatycznie formatowane, zastępując spacje podkreśleniami.
- **Relacje nadrzędnych i podrzędnych tabel:** Relacje nadrzędnych i podrzędnych tabel są wprowadzane poprzez wcięcie tabel podrzędnych pod tabelami nadrzędnymi. Tabele podrzędne powinny znajdować się na tym samym poziomie co kolumny w tabeli nadrzędnej.
- **Typ danych:** W oparciu o angielski tekst zawarty w nazwie kolumny i w przypadku braku określonego typu danych, RDBS automatycznie wyprowadza ewentualnie odpowiednie typy danych. Jeśli kolumna powinna mieć dedykowany typ danych, możliwe są następujące typy: `NUM`, `INT`, `VC` (`VARCHAR2`), czy `DATE`. Typ danych można po prostu określić na końcu nazwy kolumny, oddzielając go spacją. Jeśli określony `VC` długość powinna być zdefiniowana, `VCn` należy wprowadzić długość, gdzie `n` to długość `VARCHAR2`. Ogólnie rzecz biorąc, dostępna składnia jest wyświetlana za pośrednictwem "Help" menu.

Rysunek 2.16 przedstawia przykład notacji Quick SQL, przedstawiający naszą tabelę nadrzędną

"Departments" i odpowiednią tabelę podrzędną "Employees" po lewej stronie oraz automatycznie wygenerowany skrypt SQL w panelu SQL Output. SQL jest generowany natychmiast po każdym powrocie karetki. Ostateczny skrypt SQL może zostać zapisany w celu dalszego wykorzystania i ewentualnej adaptacji, a także wykonany w celu ostatecznego wygenerowania pustych tabel.

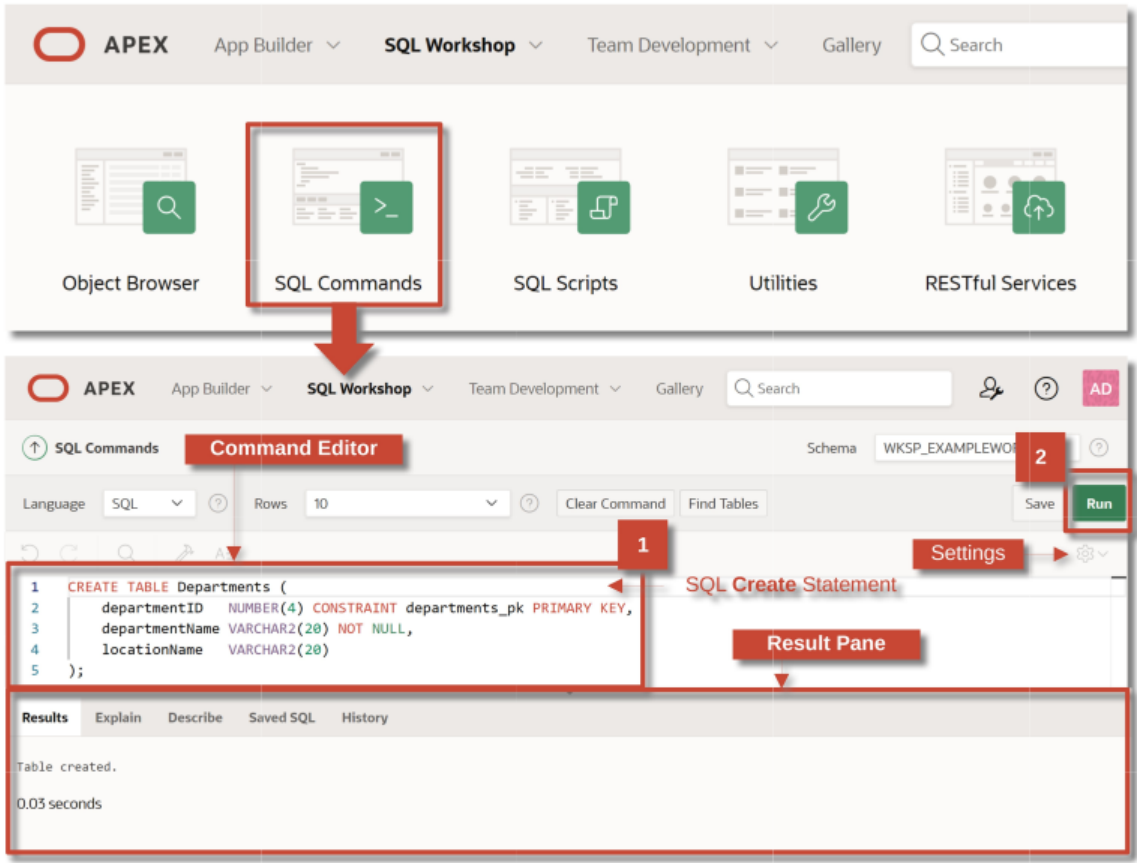


Rysunek 2.16: Wygenerowany skrypt SQL oparty na Quick SQL.

Quick SQL ma również składnię do definiowania ograniczeń (klucz obcy i dozwolone wartości) dla określonego pola danych, a także do generowania danych testowych.

2.2.3 Ręczne tworzenie tabel przy użyciu SQL-DDL

Programistyczny sposób tworzenia tabel polega na oderwaniu się od automatycznego generowania skryptów SQL (jak omówiono w sekcji 2.2.1 i sekcji 2.2.2) i samodzielnym wykonaniu pracy. Aby być bardziej konkretnym, można oczywiście również ręcznie określić skrypty SQL za pomocą instrukcji `CREATE TABLE` w edytorze SQL-Command-Editor dostarczanym przez Oracle *SQL-Workshop* (zob. rys. 2.17). Po ręcznym zdefiniowaniu instrukcji SQL za pomocą zwykłego SQL-DDL (1), wynikowy skrypt SQL można zapisać i wykonać (2) w celu utworzenia rzeczywistych pustych tabel (cf. Figure 2.17).



Rysunek 2.17: Edytor poleceń SQL.

Instrukcje SQL są mniej lub bardziej proste jak zwykły angielski, ale mają specyficzną składnię. Instrukcja `CREATE TABLE` składa się ze *słów kluczowych SQL* takich jak "`CREATE`" a także *definicji ograniczeń* (typy danych, klucz podstawowy i klucz obcy). Na rysunku 2.18, wszystkie słowa kluczowe, a także inne specyficzne dla SQL symbole składniowe (takie jak nawiasy i przecinki) są zapisane wielkimi literami i oznaczone kolorem niebieskim. Proszę zauważyć, że w słowach kluczowych SQL wielkość liter nie ma znaczenia, co oznacza na przykład, że "`CREATE`". Definicja atrybutu jest zawarta w nawiasie, atrybuty są oddzielone przecinkami, a każdy z nich jest idealnie zdefiniowany w osobnym wierszu (chociaż są to tylko nieformalne konwencje formatowania, a nie część składni SQL). Ograniczenia mogą być definiowane przy użyciu jawnej nazwy (np. departments_PK), co nie jest wymagane (np., `NOT NULL` dla departmentName), ale służy jedynie lepszej czytelności. Każda instrukcja SQL jest na końcu oddzielona średnikiem.

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

Rysunek 2.18: Instrukcja SQL tworzenia tabeli `Departaments`.

Uzupełniając instrukcję `CREATE TABLE`, istnieje oczywiście również instrukcja SQL do usuwania tabeli – działa poprawnie, niezależnie od tego, czy tabela jest pusta, czy też zawiera już jakieś dane, które są następnie usuwane (zob. rys. 2.19).

```
DROP TABLE Departments;
```

Rysunek 2.19: Polecenie SQL opuszczania tabeli `Departments`.

I wreszcie, definicja już istniejących tabelmoże również zostać zmieniona, np. poprzez dodanie dodatkowych atrybutów lub zmianę typu danych atrybuty lub zmiana typu danych (zob. rys. 2.20):

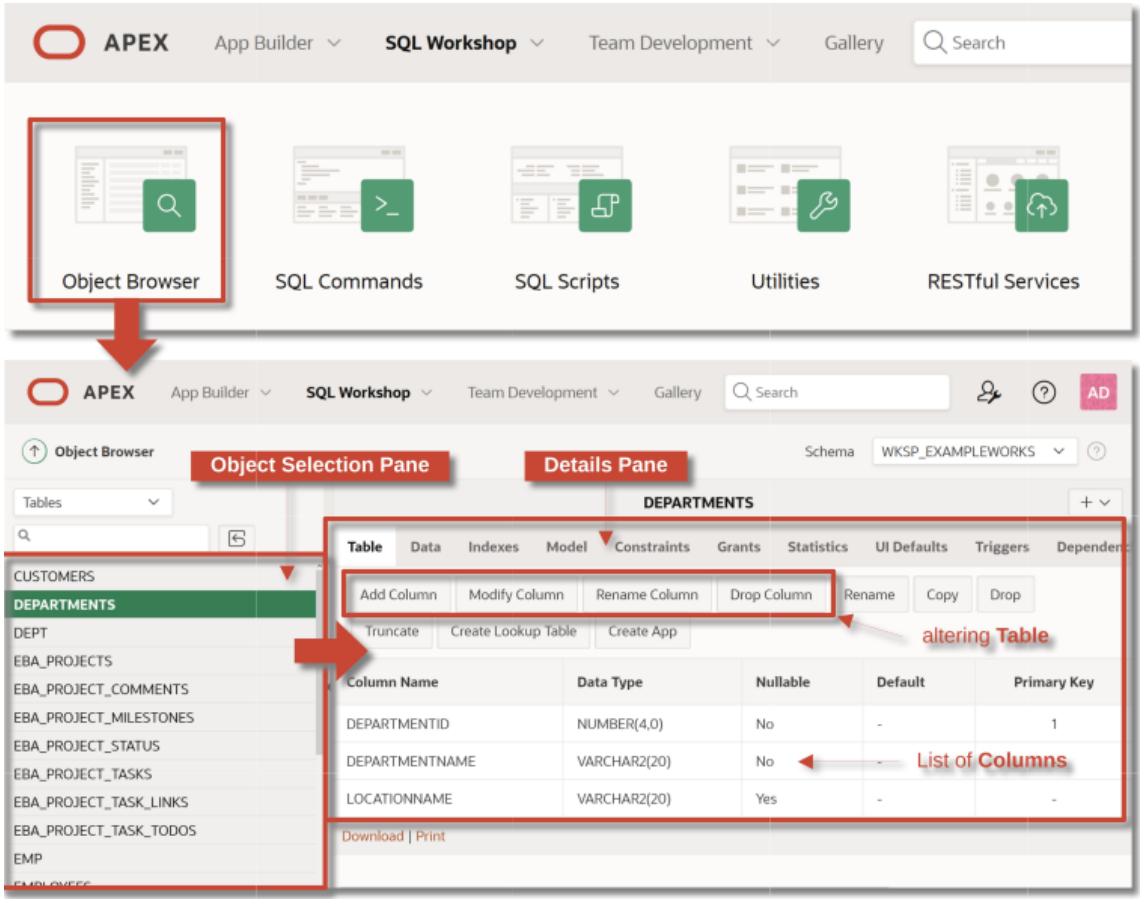
```
ALTER TABLE Departments
      ADD annualRevenue NUMBER(10);
```

Rysunek 2.20: Instrukcja SQL dla zmiany tabeli `Departments`

Jeśli tabela, która ma zostać zmieniona, zawiera już dane, wówczas modyfikacje są nieco ograniczone, ponieważ np. zmiana typu danych może prowadzić do niezamierzonych skutków dla tych danych. Kwestia ta nie jest jednak dalej omawiana w tej sekcji.

2.2.4 Ręczne zarządzanie tabelami za pomocą przeglądarki obiektów

Przeglądarka obiektów w SQL Workshop pozwala nie tylko przeglądać wszystkie wygenerowane tabele, tak zwane *Objects* w terminologii ORACLE, wraz z ich odpowiednimi atrybutami i ograniczeniami, ale także zmieniać je i tworzyć nowe, a wszystko to w oparciu o prosty kreator GUI (zob. rys. 2.21).



Rysunek 2.21: Zarządzanie tabelami za pomocą przeglądarki obiektów ORACLE.

2.3 Jak manipulować danymi - SQL-DM

Puste tabele utworzone w poprzedniej sekcji, czekają teraz na wypełnienie odpowiednimi danymi, które mogą być dalej modyfikowane i ostatecznie również usuwane. Za to właśnie odpowiedzialne są instrukcje SQL-DML `INSERT`, `UPDATE` i `DELETE`, które zostaną pokrótce przedstawione w następnej sekcji 2.3.1 przy użyciu naszego przykładu. Poza tym istnieje również alternatywa bez kodu dostępna za pośrednictwem przeglądarki obiektów ORACLE (OB), która umożliwia ręczne wstawianie danych do tabel, co jest odpowiednie dla użytkowników bez znajomości języka SQL (cf. Section 2.3.2). Wreszcie, dość ograniczoną alternatywę wstawiania danych zapewnia Quick SQL (zob. sekcję 2.3.3). Wszystkie te alternatywy zostały również przedstawione na rysunku 2.45.

2.3.1 Manipulacja danymi przy użyciu SQL-DML

Do programowej manipulacji danymi przy użyciu SQL-DML używany jest edytor poleceń SQL (cf. Figure 2.17) Poniżej zilustrowano kilka przykładów, aby dać pierwsze wrażenie na temat składni i funkcjonalności zapewnianej przez SQL-DML.

`INSERT` jest użyty, żeby *dodać nowe wiersze* do tabeli, jak widać na poniższym przykładzie (zob. rys. 2.22):

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

Rysunek 2.22: Instrukcja SQL dla wstawiania nowych danych do tabeli `Departments`.

Należy zauważyć, że oprócz tej opcji programowej wspomnieliśmy w sekcji 2.3.3 również o możliwości korzystania z Quick SQL do wstawiania danych, która jest jednak ograniczona tylko do wstawiania losowych danych.

`UPDATE` służy do aktualizacji jednego lub więcej wierszy danych w tabeli. Poniższy przykład jest najprostszą formą instrukcji `UPDATE` która zwiększa wynagrodzenie wszystkich pracowników o 10%. (cf. Figure 2.23):

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

Rysunek 2.23: Zapytanie SQL dotyczące aktualizacji wynagrodzenia wszystkich pracowników o 10% w tabeli `Employees`.

Niewielkie rozszerzenie jest pokazane w następnym przykładzie, dodając tak zwane polecenie "`WHERE`" w celu zastosować podwyżkę wynagrodzenia do pracownika o nazwisku (zob. rys. 2.24):

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

Rysunek 2.24: Instrukcja SQL dla aktualizacji wynagrodzenia pracownika "Miller", tylko i wyłącznie.

Wreszcie, również więcej niż jeden atrybut tabeli może zostać zaktualizowany za pomocą pojedynczej instrukcji, jak widać na poniższym przykładzie (zob. rys. 2.25):

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

Rysunek 2.25: Instrukcja SQL dla aktualizacji więcej niż jednego atrybutu w tabeli `Employees`.

`DELETE` służy do usuwania jednego lub więcej wierszy danych z tabeli. Poniższy przykład jest najprostszą formą instrukcji `DELETE`, usuwającą wszystkie dane przechowywane w tabeli przechowującej pracowników (zob. rys. 2.26).

```
DELETE FROM Employees;
```

Rysunek 2.26: Instrukcja SQL usuwająca wszystkie dane z tabeli `Employees`.

Rozszerzenie poprzedniej instrukcji o polecenie "`WHERE`" pozwala na selektywne usunięcie pracownika o identyfikatorze jobID "4711" (zob. rys. 2.27).

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

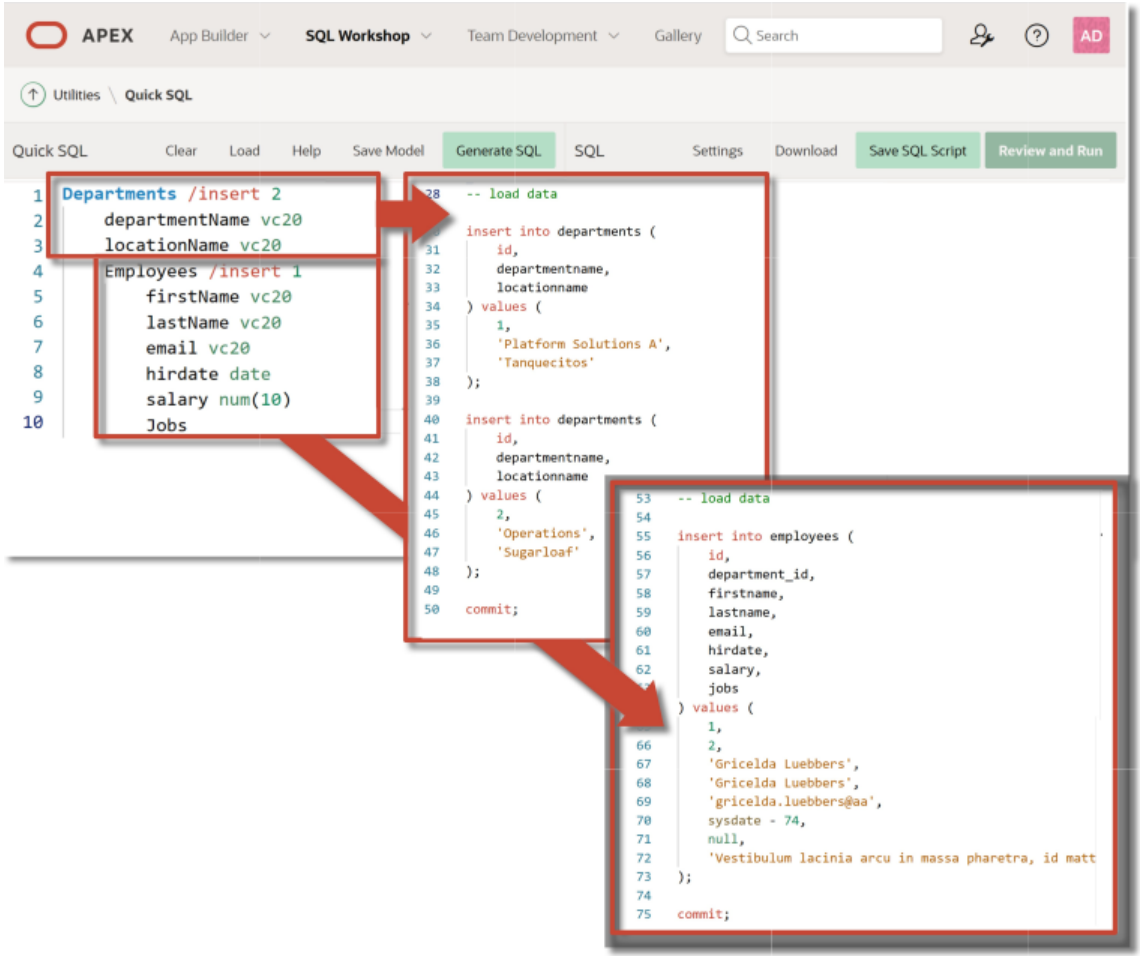
Rysunek 2.27: Instrukcja SQL do usuwania niektórych danych z tabeli `Employees`.

2.3.2 Manipulacja danymi za pomocą przeglądarki obiektów ORACLE (OB)

Oprócz tej programowej opcji manipulowania danymi, możliwe byłoby również ręczne wstawianie danych do tabel, a także zmienianie lub usuwanie istniejących danych za pomocą edytora dostępnego w *Object Browser*, zob. sekcję 2.2.4, rys. 2.21, przy użyciu zakładki "Dane". Należy zauważyć, że wszystkie manipulacje danymi wykonywane za pomocą OB są automatycznie tłumaczone przez ORACLE na odpowiednie instrukcje SQL-DML.

2.3.3 Manipulacja danymi przy użyciu Quick SQL

Za pomocą Quick SQL możliwe jest automatyczne wstawianie niektórych danych do tabel. Przykład pokazano na rysunku 2.28 przy użyciu instrukcji "`/insert 2`" and "`/insert 1`". Za pomocą tych instrukcji definiowana jest liczba "tuples" o losowych wartościach, które mają zostać wstawione do tabel.



Rysunek 2.28: Losowe wstawianie danych za pomocą Quick SQL.

2.4 Jak wyszukiwać dane - SQL-DQL

Po omówieniu różnych możliwości tworzenia pustych tabel w sekcji 2.2 i opcji manipulowania danymi w sekcji 2.3, przejdziemy teraz do pobierania danych z tabel, najpierw omawiając możliwości zapytań SQL w zakresie instrukcji "SELECT" (zob. w sekcji 2.4.1), ponownie za pomocą naszego działającego przykładu, a następnie krótki przegląd narzędzia ORACLE Query Builder ("QB") zapewniającego alternatywę zapytań "niskokodowych" dla użytkowników posiadających niewielką lub żadną wiedzę SQL (zob. sekcję

2.4.1 Zapytanie o dane przy użyciu SQL-DQL

Przede wszystkim należy zauważyć, że ekspresywność SQL-DQL jest ogromna, więc skupimy się tylko na dość niewielkim podzbiorze pojęć. Niemniej jednak, naszym celem jest danie pierwszego wrażenia na temat ekspresywności i szerokiego zastosowania SQL-DQL, poprzez wprowadzenie różnych możliwości zapytań na przykładzie, uporządkowanych w dalszej części według rosnącej złożoności. Jeśli chodzi o narzędzia, ponownie używany jest edytor poleceń SQL (zob. rys. 2.17) ten sam, który był używany do tworzenia tabel i manipulowania danymi.

(1) Zacznijmy od najprostszego zapytania – *"Retrieve all Employees"* (zob. rys. 2.29).

```
SELECT * FROM Employees;
```

Rysunek 2.29: Instrukcja SQL pobierająca wszystkie dane z tabeli `Employees`.

Tabela docelowa zapytania (`Employees`) jest określona w polecniu `FROM`, wymagany wynik zapytania jest zdefiniowany zaraz po poleceniu `SELECT` - w tym przykładzie przy użyciu gwiazdki jako znaku jokera - co oznacza, że chcielibyśmy uzyskać wartości wszystkich kolumn tabeli.

Fragment wyniku tego zapytania wygląda następująco (zob. rys. 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

Rysunek 2.30: Fragment artykułu Pobieranie wszystkich danych z tabeli `EMPLOYEES`.

Jeśli chcielibyśmy uporządkować wyniki, np. według wynagrodzeń pracowników, musimy po prostu dodać polecnie `ORDER BY` (zob. rys. 2.31):

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

Rysunek 2.31: Instrukcja SQL pobierająca wszystkie dane z tabeli `Employees` w kolejności po ich Salary.

The result of this query looks as follows (cf. Figure 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

Rysunek 2.32: Fragment z Pobieranie wszystkich danych z tabeli `Employees` uporządkowanych według ich Salary.

(2) A co z zapytaniem dotyczącym tylko określonych kolumn i określonych wierszy – *"Retrieve name, address and salary of employees having a salary higher than 5.000, only"*

Po pierwsze, aby odzyskać wartości tylko niektórych atrybutów, wystarczy określić je zamiast gwiazdy, oddzielając przecinkami. Po drugie, aby selektywnie pobrać tylko niektóre wiersze, musimy wprowadzić poleceniewHERE i określić warunek logiczny na odpowiednim atrybucie.Proszę zauważyć, że może istnieć kilka takich warunków dla różnych atrybutów, oddzielonych słowami kluczowymi AND / OR (zob. rys. 2.33).

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

Rysunek 2.33: Polecenie SQL do pobierania określonych danych z tabeli - Employees, tylko i wyłacznie.

Wynik tego zapytania wygląda następująco (zob. rys. 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

Rysunek 2.34: Fragment artykułu Pobieranie określonych danych tylko z tabeli Employees.

Jeśli chcielibyśmy uzyskać wartości atrybutu "hireDate" w innym formacie, można użyć funkcjiTO_CHAR()" dostarczonej przez ORACLE SQL. Funkcja ta przyjmuje wartość DATE jako pierwsze wejście i żądany format jako drugie wejście. Poniżej przedstawiono rozszerzoną wersję powyższego przykładu (zob. rys. 2.35), teraz z wykorzystaniem funkcji TO_CHAR() dla "hiredate" (pełna lista możliwych opcji formatowania jest oczywiście dostępna w dokumentacji Oracle):

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

Rysunek 2.35: Instrukcja SQL pobierania danych z tabeli Employees przy użyciu funkcji TO_CHAR().

The result of this query looks as follows (cf. Figure 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

Rysunek 2.36: Fragment z Pobieranie danych z tabeli `Employees` przy użyciu funkcji `TO_CHAR()`.

(3) Skupmy się na zapytaniu do kilku tabel – "*Retrieve all employees and their departments*" Pobieranie danych z dwóch lub więcej tabel w ramach pojedynczej instrukcji `SELECT` wymaga dość ważnej koncepcji SQL, czyli "*Join*". Złączenie łączy wiersze z dwóch lub więcej tabel, które muszą być określone w poleceniu `FROM`. Dodatkowo potrzebujemy polecnia `WHERE` która odgrywa rolę *matchmaker* między tymi dwiema tabelami, ponieważ musimy w jakiś sposób zdefiniować, że DBS powinien, dla każdego pracownika w tabeli `Employees`, pobrać jego wartość Departments_ departmentID (która jest FK), przejść dalej do tabeli `Departments`, dopasować tę wartość do odpowiedniej wartości PK departmentID i zwrócić dane departamentów wraz z danymi pracowników. Bardziej formalnie, to dopasowanie jest określone przez tak zwany "*join-condition*" jak można zobaczyć w poniższym przykładzie (zob. rys. 2.37):

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

Rysunek 2.37: Instrukcja SQL łącząca dane z tabeli `Employees` i tabeli `Departments`.

Wynik tego zapytania wygląda następująco (cf. Figure 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

Rysunek 2.38: Fragment artykułu Pobieranie określonych danych z tabeli `Employees` i tabeli `Departments`.

Proszę zauważyć, że odpowiednie atrybuty łączenia są również *"qualified"* przy użyciu odpowiednich nazw tabel (oddzielonych "kropką"), dążąc w ten sposób do lepszej czytelności i - w przypadku, gdy nazwa FK byłaby taka sama jak nazwa PK - zapewniając unikalną identyfikację.

(4) Pozostańmy przy agregowaniu wartości – "*Proszę pobrać liczbę pracowników i sumę ich wynagrodzeń*"

Aby uzyskać liczbę wszystkich pracowników w tabeli `Employees` możemy użyć funkcji `COUNT` nad różnymi identyfikatorami pracowników `employeeIDs`. Aby utworzyć sumę wszystkich wartości, możemy użyć funkcji `SUM` nad atrybutem, w naszym przypadku, ponieważ chcemy obliczyć sumę wszystkich wynagrodzeń , atrybut wynagrodzenie`salary` (zob. rys. 2.39):

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

Rysunek 2.39: Instrukcja SQL zliczająca liczbę pracowników i tworząca sumę ich wynagrodzeń z tabeli `Employees`.

Wynik tego zapytania wygląda następująco (zob. rys. 2.40):

COUNT(EMPLOYEEID)	SUM(SALARY)
15	149408

Rysunek 2.40: Fragment artykułu Pobieranie liczby pracowników i tworzenie sumy ich wynagrodzeń z tabeli `Employees`.

(5) Wreszcie, co z grupowaniem określonych wierszy – "*Proszę pobrać liczbę pracowników pogrupowanych według ich wynagrodzeń w porządku malejącym*"

Aby pogrupować te same dane, możemy użyć funkcji `GROUP BY` z instrukcją `SELECT`. W połączeniu z funkcją `COUNT` dla różnych `employeeIDs` jesteśmy teraz w stanie określić liczbę wierszy dla każdej grupy. Do sortowania wyników możemy użyć funkcji `ORDER BY` funkcji określającej, który atrybut powinien zostać użyty do uporządkowania. W ten sposób `ASC` deklaruje porządek rosnący, podczas gdy `DESC` deklaruje porządek malejący (zob. rys 2.41):

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

Rysunek 2.41: Instrukcja SQL grupująca dane według wynagrodzenia i porządkująca wynik według liczby pracowników z tym wynagrodzeniem z tabeli `Employees`.

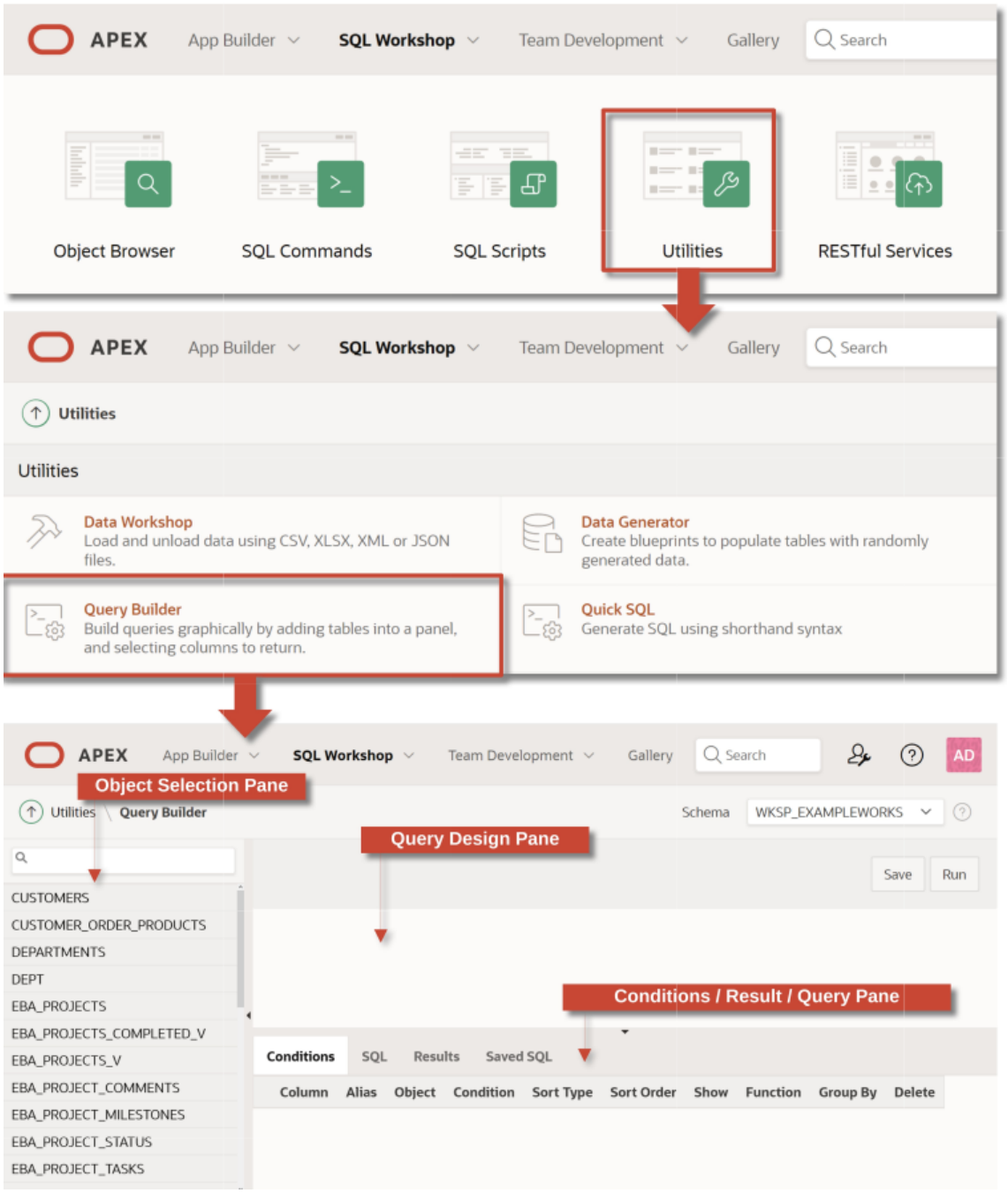
Wynik tego zapytania wygląda następująco (zob. rys 2.42):

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

Rysunek 2.42: Fragment Grupowanie danych według wynagrodzenia i porządkowanie wyników według liczby pracowników z tym wynagrodzeniem z tabeli `Employees`

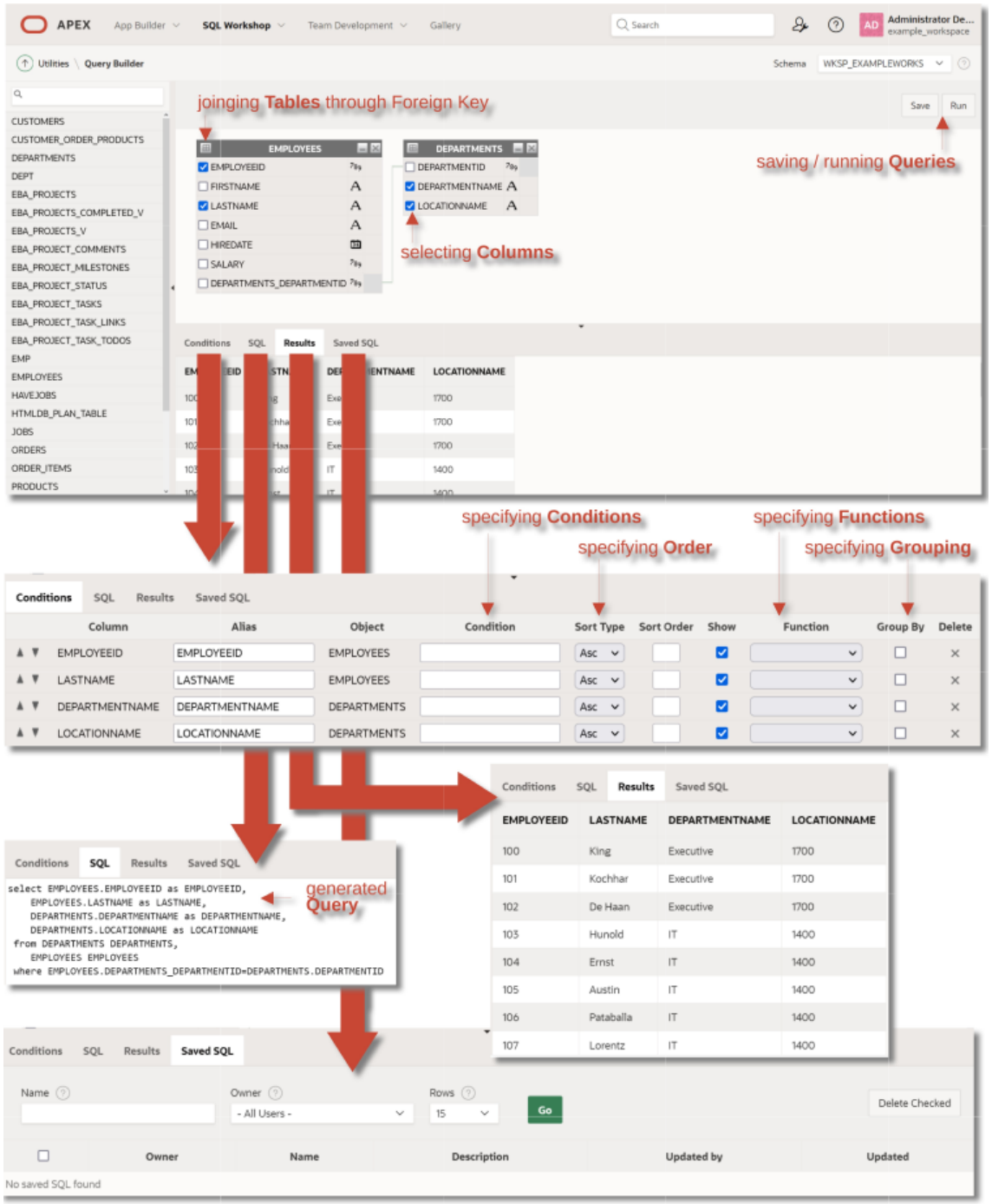
2.4.2 Zapytanie o dane przy użyciu narzędzia Query Builder (“QB”) firmy ORACLE

Jak już wspomniano, ORACLE Utilities w postaci tak zwanego Query Builder (“QB”) umożliwia użytkownikom z niewielką lub nawet żadną znajomością SQL zadawanie zapytań do tabel DB i zapisywanie zapytań do dalszego wykorzystania. Dostęp do QB można uzyskać w następujący sposób (zob. rys 2.43):



Rysunek 2.43: Przegląd narzędzia ORACLE Query Builder (“QB”).

Ogólnie rzecz biorąc, kroki niezbędne do tworzenia zapytań zostały przykładowo przedstawione na rysunku 2.44, przy użyciu naszego przykładu “Join” pokazanego na rysunku 2.37 i krótko opisane poniżej:



Rysunek 2.44: Używanie Query Builder do tworzenia Join.

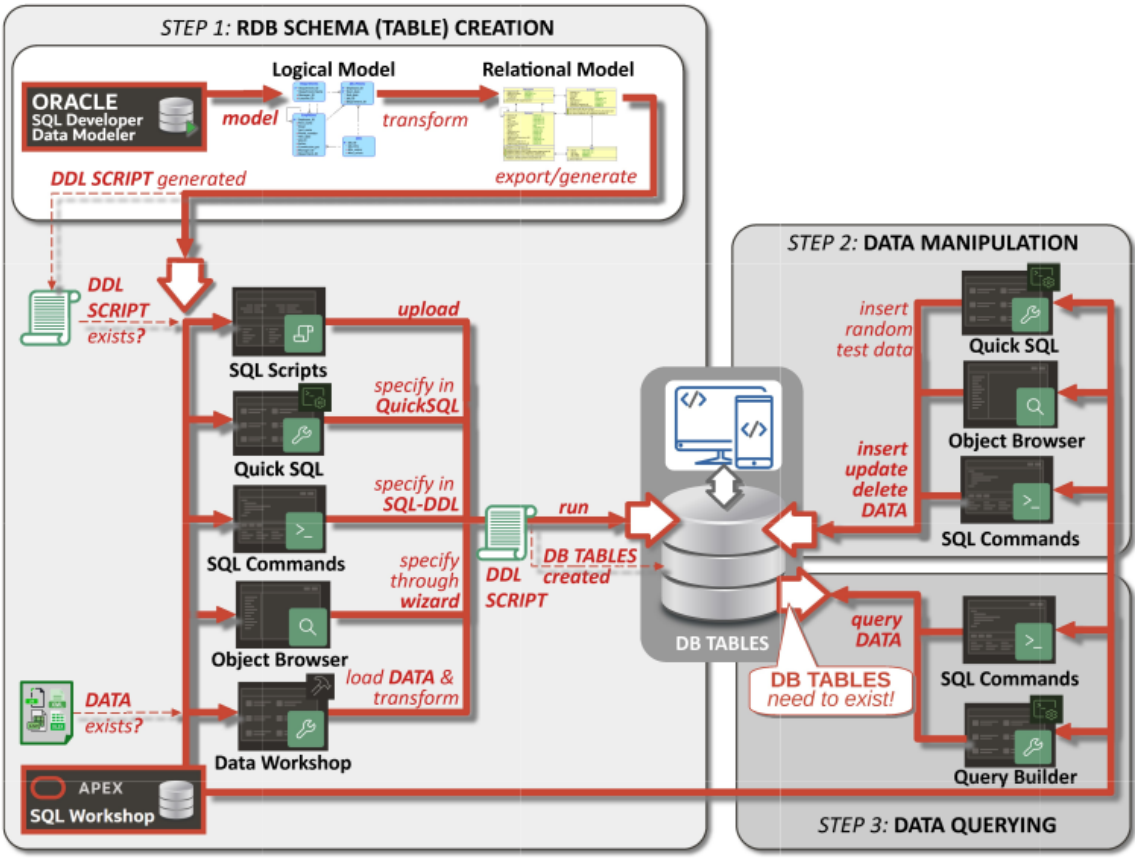
- Wybierz *cel zapytania* w postaci jednej lub więcej tabel DB (zwanych "obiektami" w QB) z panelu "Object Selection panel" (zob. rys. 2.43). Tworzy to instruckja "`FROM`"-część SQL-DQL.
- Proszę dodać wybrane obiekty do "Panelu projektu" i wybrać żądane kolumny. Tworzy to część "`SELECT`" SQL-DQL.
- Opcjonalnie: Proszę utworzyć warunki zapytania, tj. część "`WHERE`" SQL-DQL.
- Proszę wykonać zapytanie i wyświetlić wyniki (proszę kliknąć przycisk "Run").
- Opcjonalnie: Proszę wyświetlić automatycznie wygenerowany kod SQL-DQL (proszę kliknąć na "SQL-tab").

- Opcjonalnie: Proszę zapisać zapytanie do dalszego użytku (proszę kliknąć przycisk “Save”).

Na koniec należy zauważyć, że zapytania zdefiniowane za pomocą QB są ponownie automatycznie tłumaczone przez RDBS na odpowiednie instrukcje SQL-DQL.

2.5 Budowanie warstwy DB - szerszy obraz

Po zapoznaniu się z poprzednimi sekcjami, mogą Państwo być całkowicie zaskoczeni tymi wszystkimi różnymi opcjami i alternatywami, które są dostarczane przez środowisko narzędziowe ORACLE w celu zarządzania warstwą DB aplikacji internetowych. W związku z tym, aby ułatwić Państwu poruszanie się w tym gąszczu narzędzi, rysunek 2.45 przedstawia podsumowujący przegląd, ilustrujący wszystkie opcje i narzędzia, które są dostępne w celu zarządzania warstwą DB aplikacji internetowej.



Rysunek 2.45: Przegląd opcji zarządzania warstwą DB.

Ogólnie rzecz biorąc, po lewej stronie pokazane są dwa różne środowiska narzędziowe, ODM i APEX SQL Workshop. Ostateczny cel utworzenia schematu RDB w postaci tabel (Krok 1) można osiągnąć różnymi ścieżkami. Wszystkie jednak prowadzą do wygenerowania skryptów SQL, które są ostatecznie wykonywane w celu wygenerowania pustych tabel. Stamtąd można wstawiać dane do tabel, aktualizować je, usuwać (Krok 2) lub wyszukiwać dane (Krok 3) przy użyciu narzędzi środowiska roboczego SQL Workshop, jak opisano w tym rozdziale.

Aplikacja internetowa może być generowana na pustych tabelach. Proces ten został jednak szczegółowo omówiony w następnym rozdziale 3.

2.6 Pytania

1. Jaka jest różnica między modelem logicznym a modelem relacyjnym? Jakie są elementy tych dwóch modeli?
2. Czym jest normalizacja schematu RDB?
3. Proszę opisać opcje tworzenia tabel przy użyciu zestawu narzędzi Oracle.

2.7 Odpowiedzi

1. Model logiczny jest bardziej abstrakcyjny. Elementami modelu logicznego są encje, atrybuty i relacje. Model relacyjny reprezentuje bardziej konkretny, a zatem specyficzny dla DBS poziom. Elementami modelu relacyjnego są tabele, kolumny tabeli (pola danych) i relacje.
2. Normalizacja schematu RDB jest systematycznym procesem składającym się z trzech etapów zwanych "pierwszą formą normalną", "drugą formą normalną" i "trzecią formą normalną", co prowadzi do ulepszenia schematów RDB.
3. Istnieje co najmniej pięć sposobów tworzenia tabel: automatyczne generowanie tabel za pomocą Oracle Data Modeler, automatyczne generowanie tabel za pomocą Quick SQL w Oracle APEX, ręczne tworzenie tabel za pomocą SQL-DDL w Oracle APEX SQL Commands, ręczne tworzenie tabel za pomocą Oracle APEX Object Browser i automatyczne tworzenie tabel za pomocą Oracle APEX Data Workshop.



3. Jak się poruszać w APEX?

E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER

Czym jest Oracle APEX? Oracle Application Express (Oracle APEX) to oparte na przeglądarce internetowej środowisko programistyczne o niskim kodzie dla aplikacji internetowych opartych na Oracle DB. Ńiskokodoweóznacza, że do zbudowania pełnoprawnej aplikacji internetowej wymagane są tylko niewielkie umiejętności programistyczne - w przypadku pierwszego, prostego prototypu aplikacji internetowej umiejętności programistyczne nie są nawet w ogóle potrzebne.

Do jakich celów można używać APEX? Ogólnie rzecz biorąc, APEX umożliwia nie tylko tworzenie warstwy DB aplikacji internetowych za pomocą APEX "*SQL Workshop*", jak omówiono w rozdziale 2, ale także tworzenie aplikacji desktopowych lub mobilnych za pomocą APEX "*App Builder*". APEX jest wykorzystywany zarówno przez dużych, jak i małych klientów, w wielu domenach aplikacji, radząc sobie z szerokim spektrum potrzeb biznesowych. Potrzeby te mogą sięgać od prostego przekształcenia lokalnego arkusza kalkulacyjnego w aplikację internetową, aż po realizację pełnoprawnej aplikacji internetowej (patrz przykład przedstawiony w rozdziale 2) umożliwiającej przechowywanie, pobieranie, wizualizację i dalsze przetwarzanie każdej pojedynczej sprzedaży w każdym z 10 500 sklepów Walmart wraz z odpowiedzialnymi działami, pracownikami i informacjami o ich pracy.

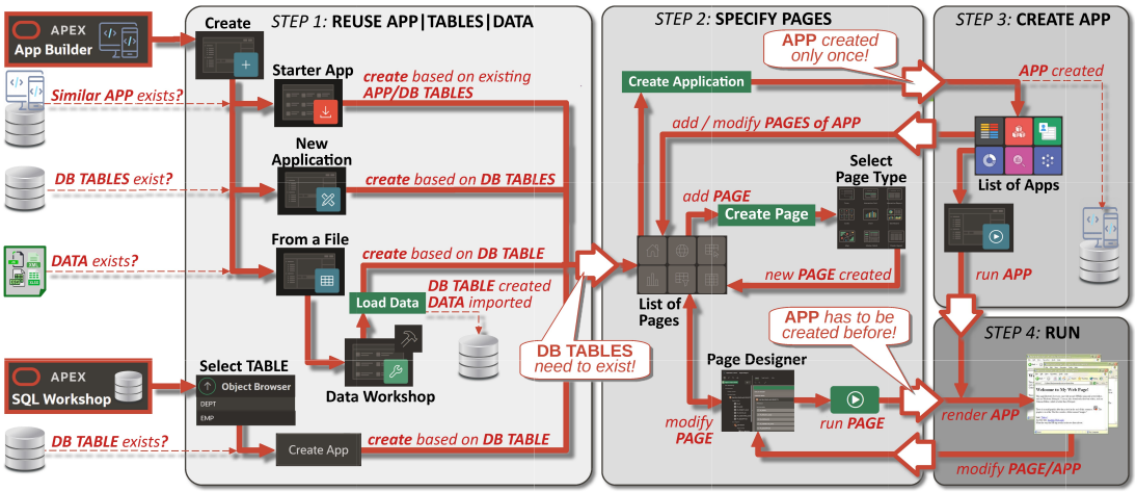
Jakie jest uzasadnienie tego rozdziału? Celem tego rozdziału jest omówienie ogólnego procesu tworzenia aplikacji internetowych, który jest realizowany przez Oracle APEXs Ąpp Builder". Proces ten stanowi podstawę dla możliwości nawigacji pomiędzy różnymi narzędziami App Builder zwanymi "komponentami programistycznymi"w celu zbudowania aplikacji internetowej. Po drugie, przedstawiono raczej ogólny przegląd tych różnych komponentów programistycznych, podczas gdy szczegóły zostaną omówione w dalszych rozdziałach.

3.1 Proces tworzenia aplikacji internetowych realizowany przez APEX

Zanim przejdziemy do szczegółów dotyczących różnych komponentów programistycznych Ąpp BuilderĄPEX, warto przyjrzeć się procesowi tworzenia aplikacji internetowych, który jest realizowany przez APEX.

APEX podąża za wieloetapowym i cyklicznym procesem rozwoju. Ogólny proces rozwoju wspierany przez APEX, który jest zilustrowany na rysunku 3.1, jest zorganizowany w wieloetapowy *wieloetapowy, częściowo cykliczny sposób* umożliwiający *przyrostowy rozwój*,stopniowe udosko-

nalanie aplikacji internetowej i ostatecznie warstwy DB. Każdy z etapów rozwoju jest wspierany przez odpowiednie kreatory komponentów programistycznych. Oznacza to, że z opcji menu na najwyższym poziomie, użytkownik jest prowadzony głębiej w funkcjonalność *App Builder* w celu określenia każdego szczegółu aplikacji internetowej, cofając się w celu wykonania przyrostowych udoskonaleń w razie potrzeby.



Rysunek 3.1: Ogólny proces rozwoju w APEX.

APEX podąża za procesem rozwoju "DB-Layer First".. Należy podkreślić, że ponieważ APEX koncentruje się na rozwoju aplikacji internetowych opartych na DB, konieczne jest, aby zawsze najpierw rozpocząć od opracowania warstwy DB (zob. rys. 3.1 ŚTEP 1: REUSE APP | TABLES | DATA") . Można to zrobić w sposób opisany w rozdziale 2 lub bezpośrednio w App Builder, jeśli np. istnieją już zewnętrzne pliki danych, które można ponownie wykorzystać do utworzenia tabel DB (zob. rys. 3.1 “From a File”). W przypadku, gdy tabele DB już istnieją, proces rozwoju można rozpocząć (i) z poziomu App Builder, tym samym ostatecznie ponownie wykorzystując (i dostosowując) istniejącą aplikację wraz z jej tabelami DB (zob. rys. 3.1 Śtarter App") lub tworząc nową aplikację od podstaw, ponownie w oparciu o istniejące tabele DB (zob. rys. 3.1 Ńew Application") lub (ii) z poziomu SQL Workshop, po prostu wybierając tabelę DB w przeglądarce obiektów (zob. rys. 3.1 Ślect TABLE"), i rozpoczynając proces rozwoju aplikacji na podstawie tej tabeli.

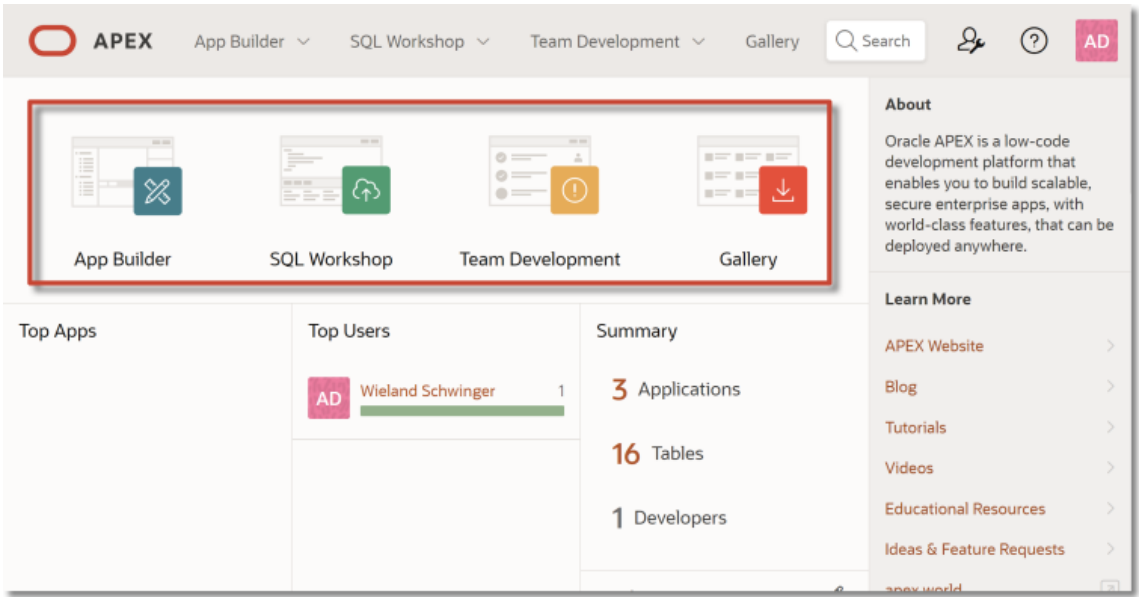
APEX podąża za procesem rozwoju "Page-Driven. Gdy tylko istnieją tabele DB do przechowywania danych aplikacji internetowej, można je wykorzystać do utworzenia nowej aplikacji internetowej, zgodnie z procesem *page-driven process* (zob. rys. 3.1 ŚTEP 2: SPECIFY PAGES"). Oznacza to, że można zdefiniować jedną lub więcej stron internetowych, tworząc strony HTML ostatecznej aplikacji internetowej. Każda z tych stron może być oparta na jednej lub kilku tabelach DB, umożliwiając wizualizację ich danych pod względem różnych (typów stron) *page types* zawierających np. interaktywne raporty, formularze, listy, wykresy lub kalendarze, ale także interaktywną żabawęź danymi np. powiększanie szczegółów), a nawet manipulowanie nimi i zapisywanie zmian z powrotem w DB. Te *strony mogą być ze sobą połączone* za pomocą menu nawigacyjnych, kart, przycisków lub łączy hipertekstowych..

Po utworzeniu jednej lub więcej stron konieczne jest utworzenie całej aplikacji (zob. rys. 3.1 ŚTEP 3: CREATE APP"), określając w ten sposób niektóre ogólne właściwości całej aplikacji internetowej (np. wygląd aplikacji) lub po prostu korzystając z ustawień domyślnych. Wreszcie, aplikacja i/lub każda z określonych stron musi zostać wyrenderowana w celu wygenerowania ostatecznych stron HTML (rysunek 3.1 ŚTEP: 4 RUN"), które mogą być dalej interaktywnie testowane, od czasu do czasu cofając się w celu dalszego udoskonalenia aplikacji internetowej.

3.2 Menu startowe APEX

Po zalogowaniu się do Oracle APEX pojawia się tak zwana *"Workspace Homepage"*. Strona główna obszaru roboczego zapewnia dostęp do czterech różnych komponentów programistycznych (zob. rys. 3.2):

- *"App Builder"* służy głównie do tworzenia stron internetowych i będzie głównym tematem tego rozdziału.
- *"SQL Workshop"* służy do tworzenia warstwy DB i został już opisany w Rozdziale 2.
- *"Team Development"* służy np. do interakcji z innymi programistami za pośrednictwem systemu zgłoszeń w celu ustalenia kamieni milowych itp.
- *"Gallery"* służy do prostego instalowania gotowych aplikacji demonstracyjnych, które można następnie następnie zmodyfikować i dostosować do własnych potrzeb, ponownie korzystając z App Builder".



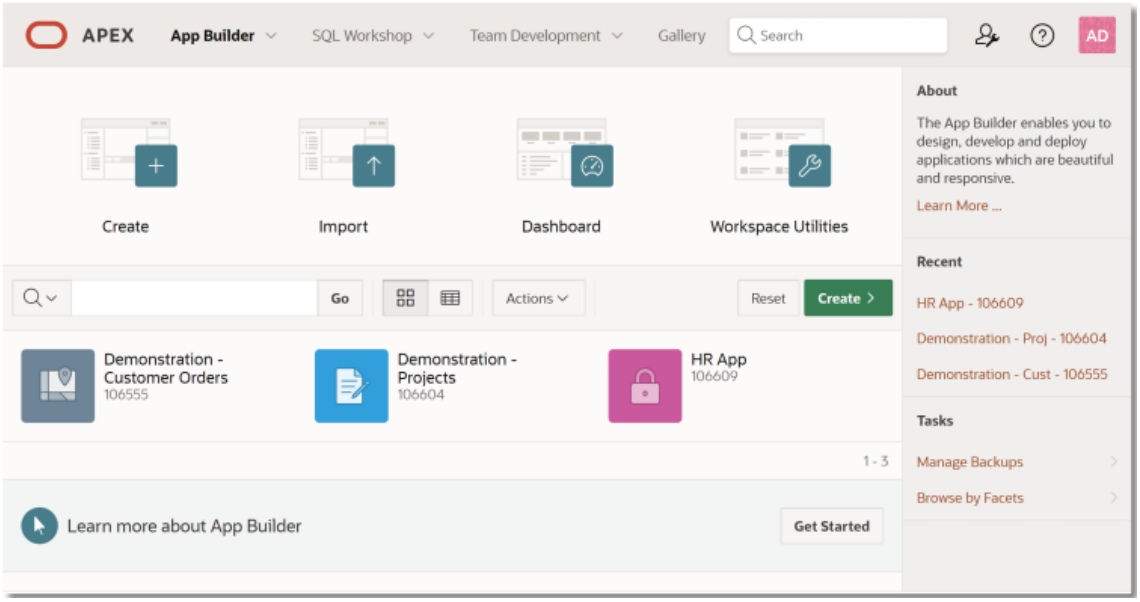
Rysunek 3.2: Oracle APEX Workspace Homepage.

W dolnej części strony głównej obszaru roboczego (cf. Figure 3.2), okna "*Top Apps*", "*Top Users*", i "*Summary*" oferują informacje w czasie rzeczywistym na temat działań rozwojowych w bieżącym obszarze roboczym DB, podczas gdy po prawej stronie znajdują się odniesienia do dalszych zasobów edukacyjnych APEX.

3.3 Przegląd aplikacji App Builder - tworzenie i zarządzanie aplikacjami

Po przejściu do strony głównej App Builder dostępne są następujące opcje (zob. rys. 3.3):

- *Create* nową aplikację internetową.
- *Import* wcześniej wyeksportowane aplikacje internetowe.
- *View the "Dashboard"* dostarczający statystyki dotyczące już stworzonych aplikacji internetowych.
- *Access "Workspace Utilities"* zapewniających szeroki zakres usług, takich jak zdalny dostęp do danych lub funkcje tworzenia kopii zapasowych.



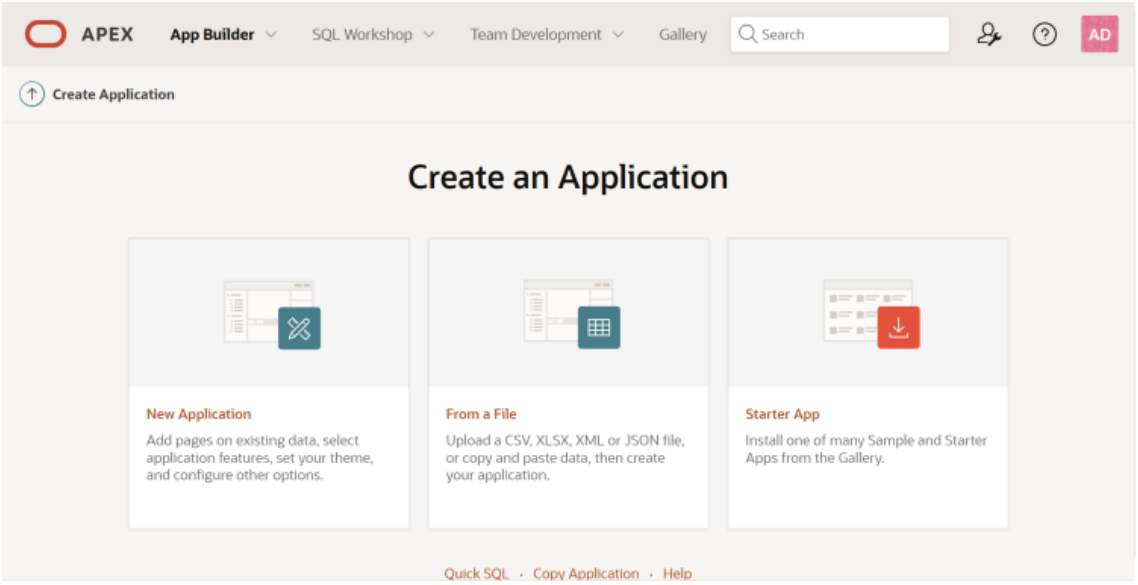
Rysunek 3.3: Przegląd aplikacji App Builder.

W dolnej połowie okna (cf. Figure 3.3), znajduje się lista już istniejących aplikacji internetowych, które można wybrać do dalszej edycji lub renderowania (patrz następne sekcje).

3.4 Tworzenie aplikacji - trzy przypadki użycia

Po wybraniu opcji *"Create"* w App Builder, "*Create Application Wizard*" wyświetla trzy dodatkowe opcje (zob. rys. 3.4),wspierając w rzeczywistości rzeczywiste przypadki użycia, które uwzględniają możliwy potencjał ponownego wykorzystania danych i samej aplikacji:

- *"New Application"* należy wybrać, jeśli warstwa DB (lub jej część) już istnieje lub jeśli warstwa DB powinna zostać utworzona "w locie"podczas tworzenia aplikacji.
- *"From a File"* ijest opcją wyboru, jeśli warstwa DB powinna być automatycznie generowana z zewnętrznego pliku, np. pliku EXCEL.
- *"Starter App"* pozwala na ponowne wykorzystanie (i prawdopodobnie dalszą modyfikację) już istniejącej aplikacji wraz z jej warstwą DB z "Galeriiąplikacji wspomnianych powyżej.



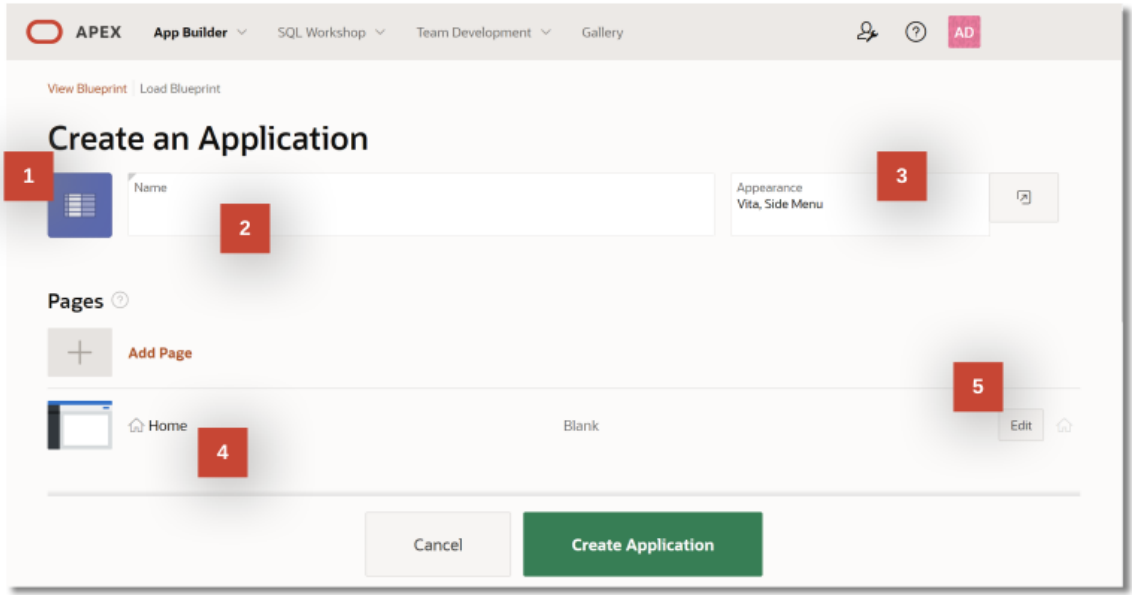
Rysunek 3.4: Kreator tworzenia aplikacji: Trzy przypadki użycia podczas tworzenia aplikacji.

Poniżej skupimy się na pierwszej opcji *"New Application"*, druga opcja *"From a File"* zostanie opisana bardziej szczegółowo w rozdziale 4, a trzecia opcja *Śtarter App"* nie będzie dalej omawiana..

3.5 Tworzenie aplikacji - właściwości, strony, funkcje i ustawienia

Podczas tworzenia nowej aplikacji internetowej należy zdefiniować niektóre *typowe właściwości*, *strony* muszą zostać utworzone, kilka dodatkowych *funkcji* and *ustawień* muszą zostać wybrane. Poniżej, wszystkie te definicje są omówione bardziej szczegółowo, gdzie rysunek 3.5) pokazuje górną część GUI, a rysunek 3.6) pokazuje jego dolną część..

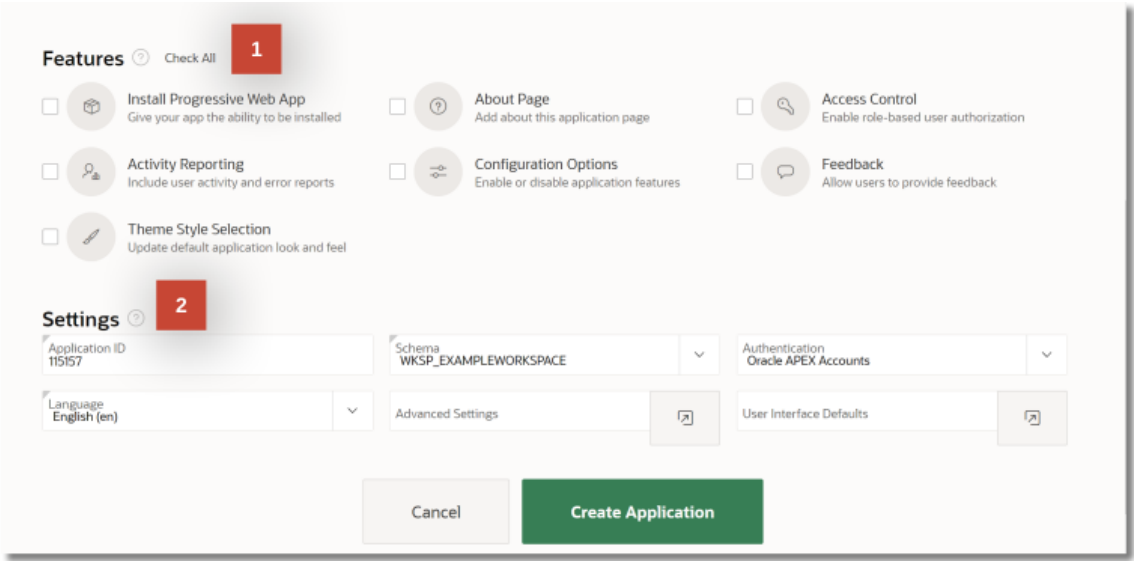
Wspólne właściwości - ikona, nazwa i wygląd aplikacji. eśli chodzi o wspólne właściwości, jak pokazano na rysunku 3.5, (1) (1) można wybrać lub załadować *icon*, używaną jako Favicon i ikona dla PWA, Apple touch lub App Builder, (2) *nazwa* musi zostać wpisana, (3) *wygląd* aplikacji internetowej można wybrać spośród kilku wstępnie zdefiniowanych opcji, określając jej "wygląd i styl"pod względem stylu motywu (np. kolor, rozmiar czcionki itp.) oraz *rodzaj menu nawigacyjnego* (np. menu boczne lub menu górne), przy czym dostępny jest również wygląd domyślny.



Rysunek 3.5: Właściwości i strony.

Tworzenie stron dla aplikacji. Głównym zadaniem jest jednak przyrostowe tworzenie aplikacji internetowej poprzez sukcesywne *dodawanie nowych stron* używając *"Create Page Wizard"* (uruchamianego przez *"Add Page"*, patrz (4) na rysunku 3.5. Po utworzeniu strony można, w dowolnym momencie podczas tworzenia aplikacji, *edytować skład* strony, *zmieniać* kolejność stron, i *usuwać* je (zob. (5) rysunek 3.5). W dolnej części rysunku 3.5, pokazano już utworzoną stronę "Home". Dalsze szczegóły dotyczące "*Create Page Wizard*" są przedstawion w sekcji 3.6.

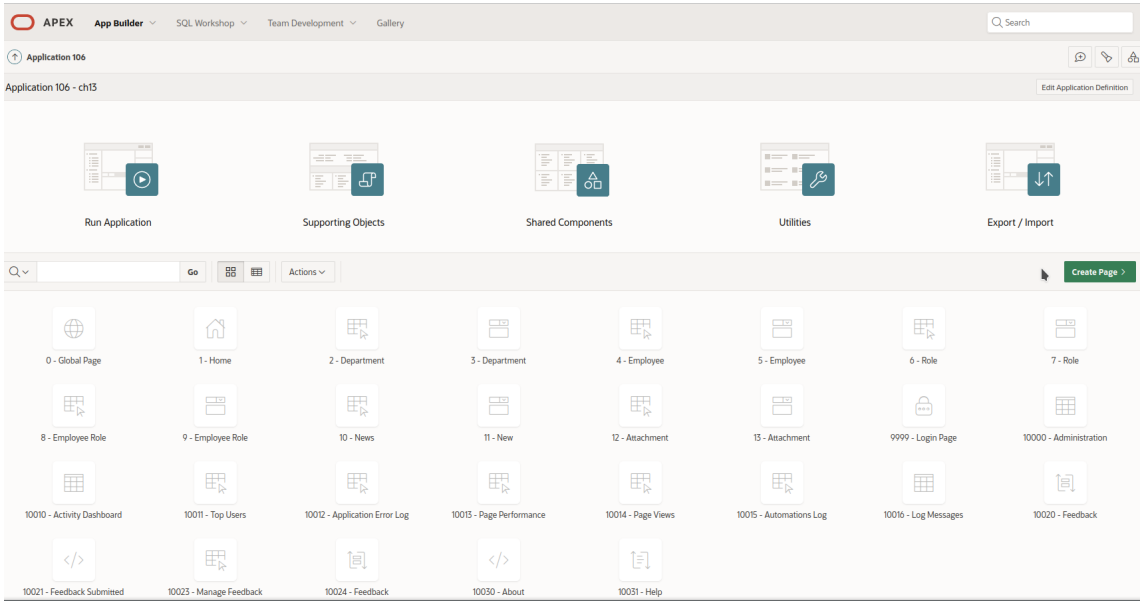
Funkcje i ustawienia aplikacji. Wreszcie, aplikacja może zostać wzbogacona o pewne dość przydatne *funkcje* (zob. (1) na rysunku 3.6) takie jak "About Page", strona automatycznej "Access Control" albo strona "User Feedback" a także niektóre typowe *ustawienia*, np. nazwa schematu zawierającego wszystkie tabele DB, a tym samym tworzącego warstwę DB dla aplikacji lub używanego języka (zob. (2) rys. 3.6).



Rysunek 3.6: Funkcje i ustawienia.

Po kliknięciu "*Create Application*"przycisku w "*App Builder Homepage*"pojawi się nowo

utworzona aplikacja wraz z jej stronami (zob. rys. 3.7).

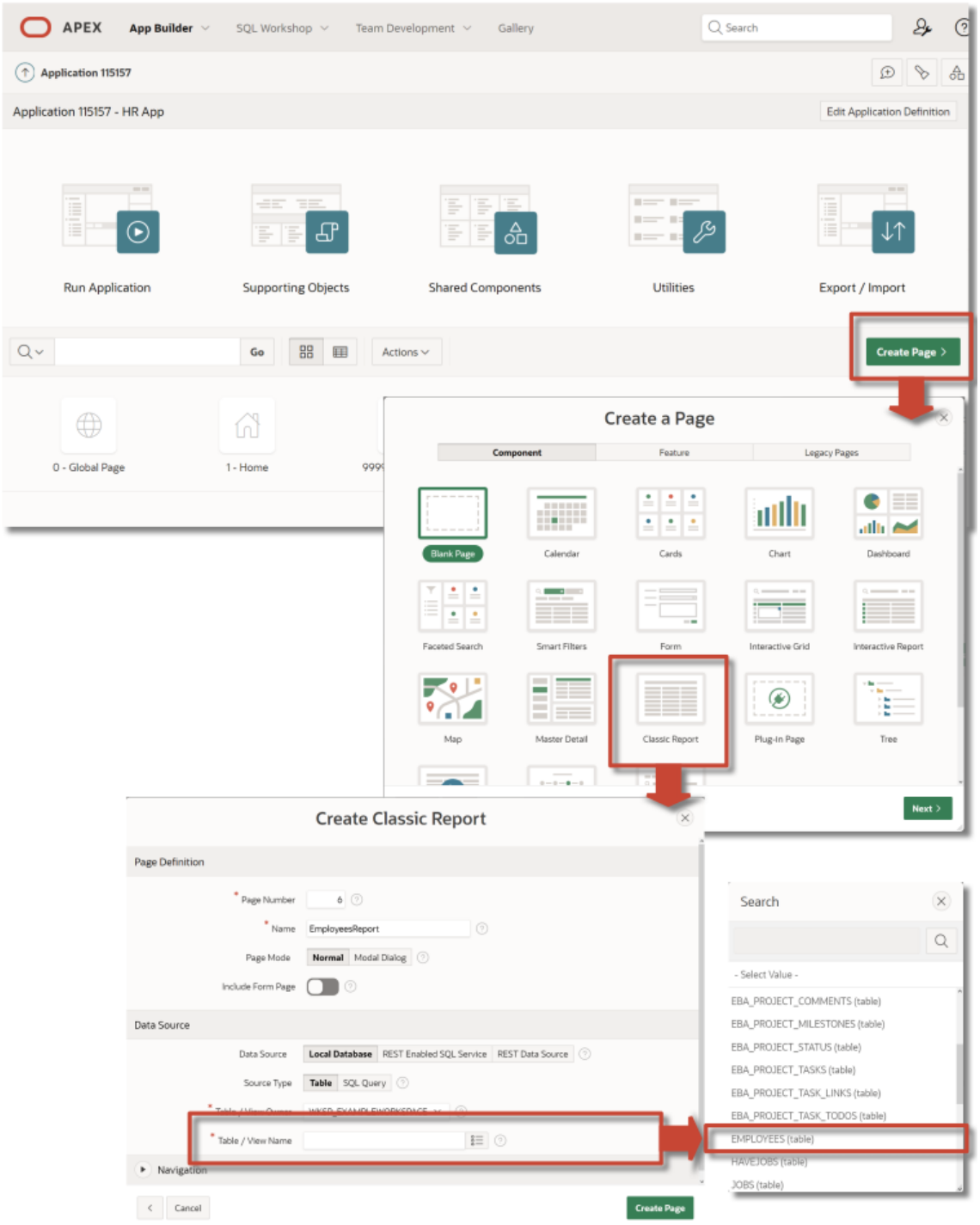


Rysunek 3.7: Strona główna aplikacji - widok dewelopera.

3.6 Określanie stron

Co składa się na stronę? Jak już wspomniano, strona jest podstawowym elementem składowym aplikacji internetowej, tj. każda aplikacja składa się z co najmniej jednej lub wielu stron. Każda strona może zawierać przyciski i pola (zwabe *"items"*) które są pogrupowane w "konteneryźwane *"regions"*. Strony mogą również zawierać *application logic*, np. wykonywać *obliczenia* (called *"computations"*) , i przeprowadzać *validations* takie jak sprawdzanie poprawności podczas manipulacji danymi.

Proszę wybierać spośród 18 różnych typów stron Strona może wyświetlać dane tabel DB w różnych formach określonych przez tak zwane "typy stron", obejmujące np. interaktywne raporty , listy , kalendarze czy wykresy . Typ strony należy wybrać za każdym razem, gdy tworzona jest nowa strona, albo za pomocą przycisku "Dodaj stronę" w oknie dialogowym "Utwórz aplikację"(zob. rys. 3.5) albo za pomocą przycisku "Utwórz stronęńa stronie głównej aplikacji dewelopera (zob. rys. 3.8). Ogólnie dostępnych jest 18 różnych typów stron, jak pokazano na rysunku 3.8. Określony typ strony określa *zawartość*, *kompozycję*, i *układ* strony, co można rozpoznać, patrząc na ikony typów stron na rysunku 3.8. Szczegółowe informacje na temat każdego z tych typów stron znajdują się w kolejnych rozdziałach.



Rysunek 3.8: Tworzenie strony.

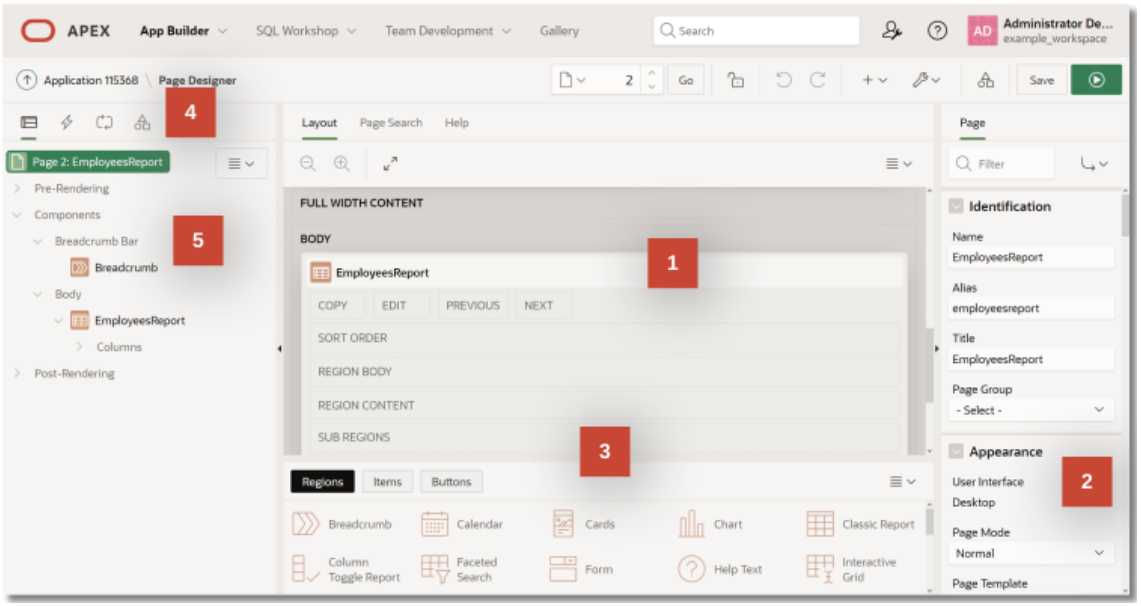
Okno dialogowe specyfikacji strony - mapowanie tabeli na stronę. Decydując się na określony typ strony, konkretny dialog specyfikacji strony, który jest prezentowany przez "Kreator tworzenia strony", jest naturalnie zależny od wybranego typu strony. Niemniej jednak, ponieważ prawie wszystkie dostępne typy stron są oparte na tabelach DB, głównym zadaniem jest zwykle mapowanie odpowiednich tabel DB na stronę, zapewniając źródło wizualizacji danych i cel manipulacji danymi. Rysunek 3.8 przedstawia przykładowe okno dialogowe dla dość popularnego typu strony "*Classic Report*". Można zauważyć, że oprócz kilku podstawowych informacji, takich jak Ńazwa", należy również określić mapowanie do warstwy DB, wybierając "Źródło danych"w

odniesieniu do tabeli DB, która powinna być źródłem tego raportu (na rys. 3.8 wybieramy tabelę Employeesź naszego uruchomionego przykładu). Należy zauważyć, że w celu wyświetlenia danych z bazowej tabeli DB, APEX z natury wykorzystuje SQL-DQL w tle, gdy tylko strona HTML jest renderowana.

Jak tylko nowa strona zostanie dodana, komponent "*Page Designer*" APEX (patrz sekcja 3.7) jest automatycznie otwierany dla tej strony, umożliwiając przeglądanie i edycję specyfikacji strony. W tym samym czasie strona może być również natychmiast i automatycznie renderowana do strony HTML, np. do celów testowania i przyrostowego rozwoju, za pomocą przycisku Żun"(patrz sekcja 3.8).

3.7 Utrzymywanie i modyfikowanie strony – Page Designer

Już utworzone strony mogą być oczywiście dalej utrzymywane i ulepszane za pomocą tak zwanego "*Page Designer*". The Page Designer to w pełni funkcjonalne *zintegrowane środowisko programistyczne (IDE)* które zawiera pasek narzędzi i wiele paneli (patrz rys. 3.9). Funkcjonalność zapewniana przez Page Designer jest ogromna, pozwalając między innymi na pełną modyfikację kompozycji/układu strony i jej komponentów (por. (1), (2) i (5)), w tym elementów, przycisków i regionów (por. (3)), ale także na modyfikację renderowania strony (por. (lub logiki aplikacji w zakresie obliczeń i walidacji (por. (4))



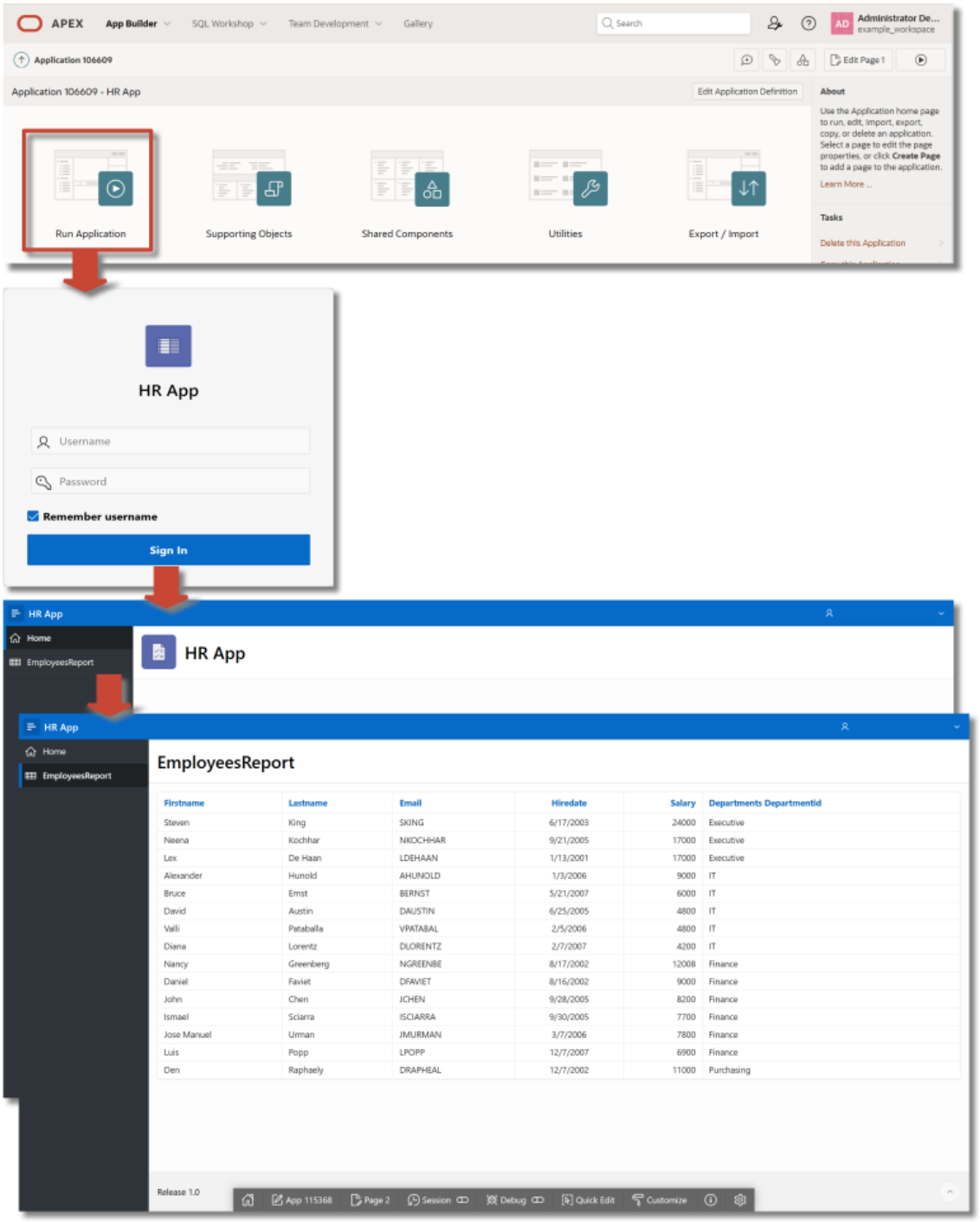
Rysunek 3.9: Kreator strony (Page Designer

Szczegóły dotyczące funkcjonalności kreatora stron znajdą Państwo w kolejnych rozdziałach.

3.8 Uruchamianie aplikacji lub strony

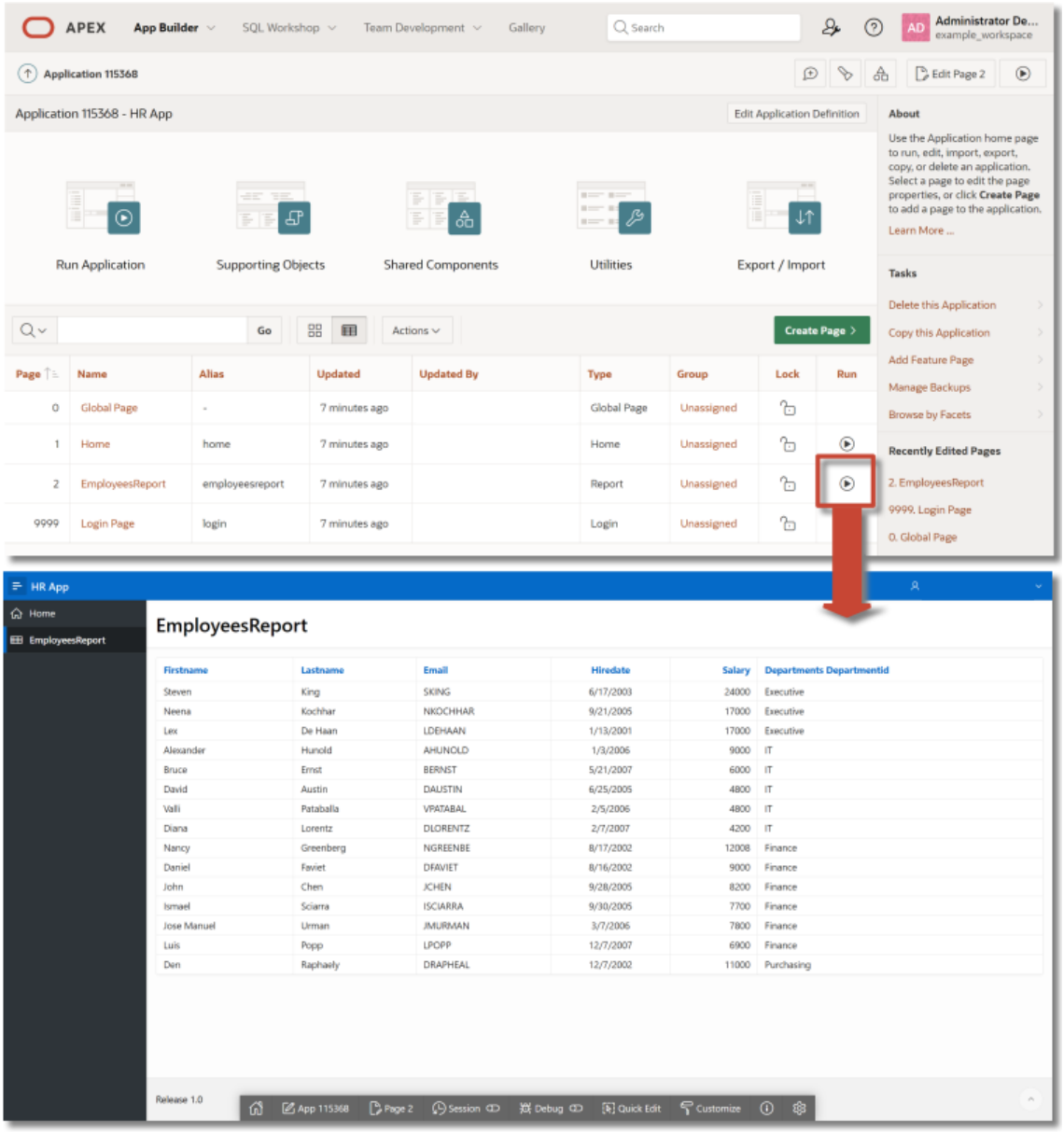
Aby wyświetlić wyrenderowaną wersję aplikacji lub pojedynczej strony, należy przesłać ją do tak zwanego *Oracle APEX engine* klikając przycisk "*Run*" na *stronie głównej aplikacji* (patrz rys. 3.10 i rys. 3.11). Silnik APEX dynamicznie renderuje i przetwarza strony na widoczne strony HTML w oparciu o dane o aplikacji, które są przechowywane w "wewnętrznych"tabelach DB. Istnieją dwie różne opcje dostępne na stronie *głównej aplikacji*, w zależności od tego, czy chcą Państwo uruchomić całą aplikację, tj. wszystkie strony, czy tylko pojedynczą stronę, np. do celów testowych.

- **Uruchom aplikację lub stronę**, i.e., all pages: This option is available on the Application Homepage as can be seen in Figure 3.10.



Rysunek 3.10: Uruchamianie całej aplikacji.

- **Uruchamianie poszczególnych stron**. Podczas tworzenia nowych stron, możliwe jest również uruchomienie każdej strony indywidualnie, aby uzyskać natychmiastowe wrażenie wyglądu i działania rzeczywistej strony. Przycisk "Uruchom stronę"przypomina małą, czarną ikonę gry i wyświetla się po prawej stronie każdej ze stron aplikacji, jak widać na rysunku 3.11. Należy jednak zauważyć, że nawet jeśli uruchamia się tylko pojedynczą stronę, dostępne są również wszystkie inne strony.



Rysunek 3.11: Uruchamianie poszczególnych stron.

Proszę zauważyć, że inne opcje, które są dostępne na stronie głównej aplikacji oprócz "Uruchom aplikację"(np. Obiekty pomocnicze) są omówione w kolejnych rozdziałach.

3.9 Pytania

1. Jakie opcje udostępnia kreator tworzenia aplikacji i do czego one służą?
2. Jakie funkcje oferuje kreator Ńowa aplikacja"?
3. Jakiego rodzaju strony może generować APEX?

3.10 Answers

1. Dostępne są trzy opcje: a) Ńowa aplikacja"pozwala deweloperowi dodawać strony dla istniejących danych, wybierać funkcje aplikacji, ustawiać motyw i konfigurować ją, b) Ź pliku"pozwala deweloperowi przesyłać pliki w formatach takich jak CSV, XLSX, XML i

JSON lub kopiować/wklejać dane i inicjować tworzenie aplikacji oraz c) Ąplikacja startowa"pozwala deweloperowi zainstalować jedną lub wiele przykładowych i startowych aplikacji z Galerii.

2. Dostępnych jest kilka przydatnych funkcji, takich jak automatyczne tworzenie strony Ąbout Page", udostępnianie administratorowi aplikacji strony do nadawania uprawnień użytkownikom w funkcji Ąccess Control", automatyczne tworzenie strony "User Feedback", raportowanie aktywności, wybór stylu motywu, opcje konfiguracji i wybór instalacji progresywnej aplikacji internetowej.
3. APEX może generować szeroką gamę stron: formularz, raport (interaktywna siatka, interaktywny raport, klasyczny raport, główny szczegół), pusta strona, kalendarz, karty, wykres, pulpit nawigacyjny, wyszukiwanie fasetowe, inteligentny filtr, mapa, strona wyszukiwania, strona wtyczki, drzewo, ładowanie danych, kreator i ujednolicona lista zadań.

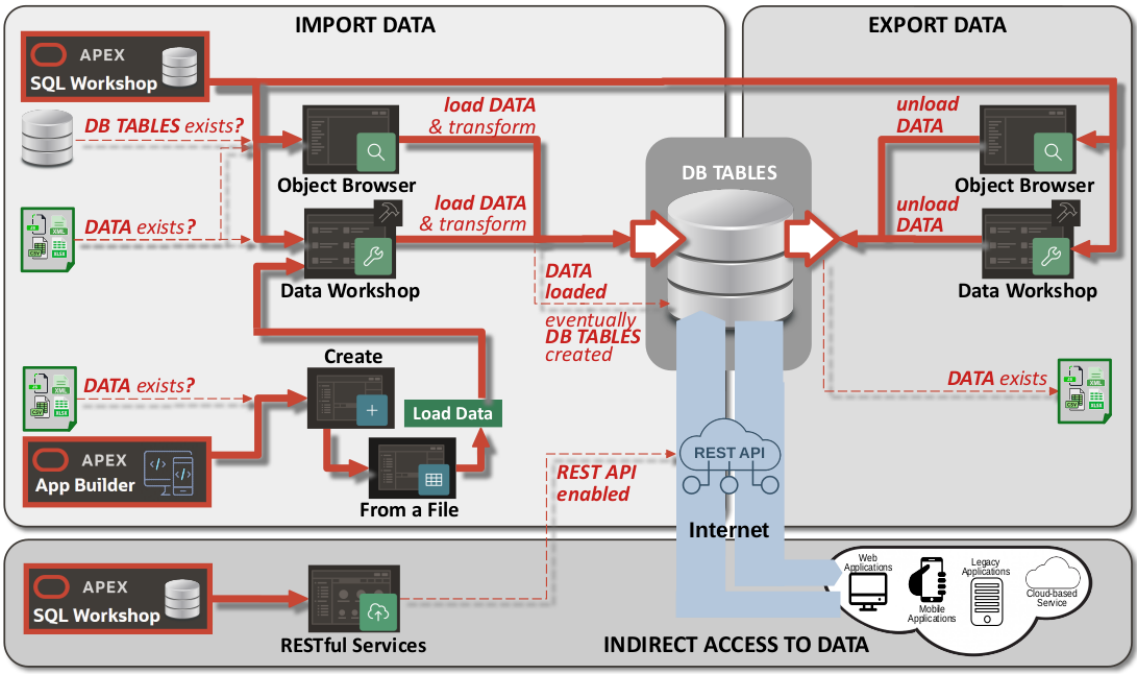


4. Jak wymieniać dane w APEX?

E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER

Głównym zadaniem podczas tworzenia warstwy DB dla aplikacji internetowej jest utworzenie tabel DB i wstawienie odpowiednich danych (patrz rozdział 2). Można to jednak zrobić nie tylko ręcznie, jeśli dane jeszcze nie istnieją. W przypadku, gdy istnieją już dane dostępne w plikach zewnętrznych, można je po prostu ponownie wykorzystać i *zaimportować* do bazy danych Oracle. Jednocześnie może być pożądane, aby od czasu do czasu *eksportować* już istniejące dane do pliku, aby mogły być ponownie wykorzystane w drugim kroku przez inną, zewnętrzną aplikację. I wreszcie, w przypadku, gdy niektórzy klienci zewnętrzni (np. inne aplikacje internetowe / mobilne / legacy lub usługi oparte na chmurze) powinni mieć zapewniony *dostęp online* do danych w naszej bazie danych, zamiast tylko wymieniać pliki danych od czasu do czasu, istnieje również możliwość zapewnienia pośredniego dostępu online do naszych danych DB.

Te trzy przypadki użycia dla zostały zilustrowane na rysunku 4.1, przedstawiającym podsumowanie różnych opcji Oracle APEX do importowania, eksportowania i pośredniego dostępu do danych DB. Niniejszy rozdział ma na celu przedstawienie przeglądu tych różnych opcji.



Rysunek 4.1: Opcje wymiany danych w APEX.

4.1 Importowanie i eksportowanie danych przy użyciu Data Workshop

Scenariusz zastosowania Data Workshop. Data Workshop jest łatwym w użyciu narzędziem, zapewniającym "kreatora" w celu uproszczenia zadania importowania/eksportowania danych z/do plików zewnętrznych. Data Workshop jest szczególnie odpowiedni dla scenariuszy aplikacji zajmujących się danymi o umiarkowanym rozmiarze (mniej niż 10 tabel) posiadającymi tylko proste standardowe typy danych (np. bez pól wielowartościowych lub struktur zagnieżdżonych). Należy zauważyć, że do ponownego wykorzystania i importowania *ogromne i złożone zbiory danych*, Oracle zapewnia inne odpowiednie narzędzia, np. w zakresie tzw. *SQL*Loader Utility*. Poniżej, po pierwsze, sekcja 4.1.1 i sekcja 4.1.2 dotyczą importowania danych, podczas gdy sekcja 4.1.3 opisuje eksportowanie danych.

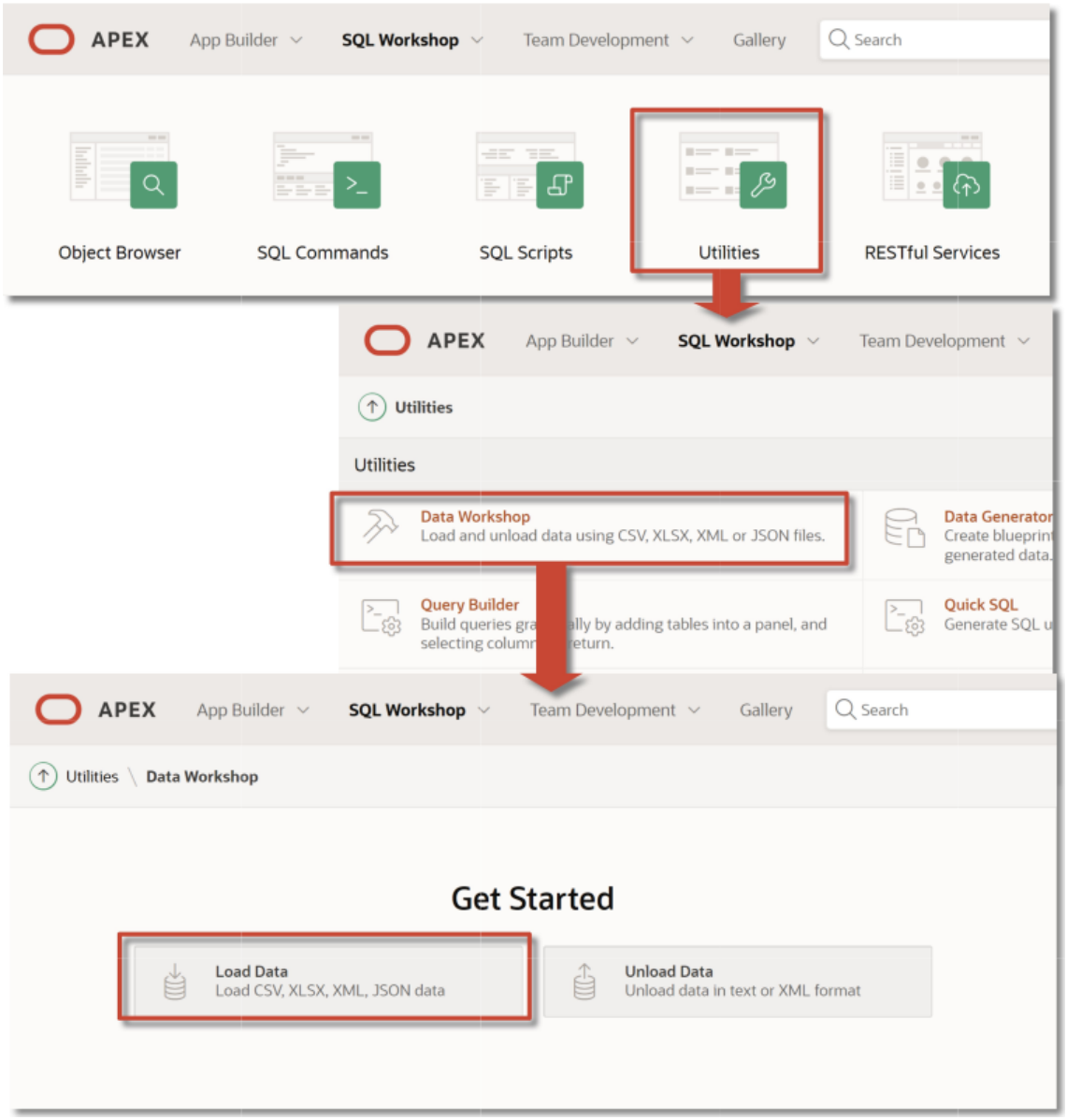
4.1.1 Importowanie danych

Formaty plików. Ogólnie rzecz biorąc, Data Workshop umożliwia ładowanie danych z plików zewnętrznych zgodnych z następującymi formatami:

1. Dowolny standardowy format rozdzielony (tabulator lub przecinek) – *Comma Separated Values*).
2. *Pliki XLSX* (i.e., *Excel workbooks*).). Należy zauważyć, że jeśli przesłany plik XLSX zawiera wiele arkuszy, domyślnie wybierany jest pierwszy arkusz. Aby wczytać inny arkusz, można go wybrać z listy wyboru "Wybierz arkusz".
3. *Pliki JSON* (*Java Script Object Notation*). Należy zauważyć, że obsługiwany jest tylko jeden poziom.
4. *Pliki XML* (*eXtended Markup Language*). Należy zauważyć, że podobnie jak w przypadku plików JSON, obsługiwany jest tylko jeden poziom.

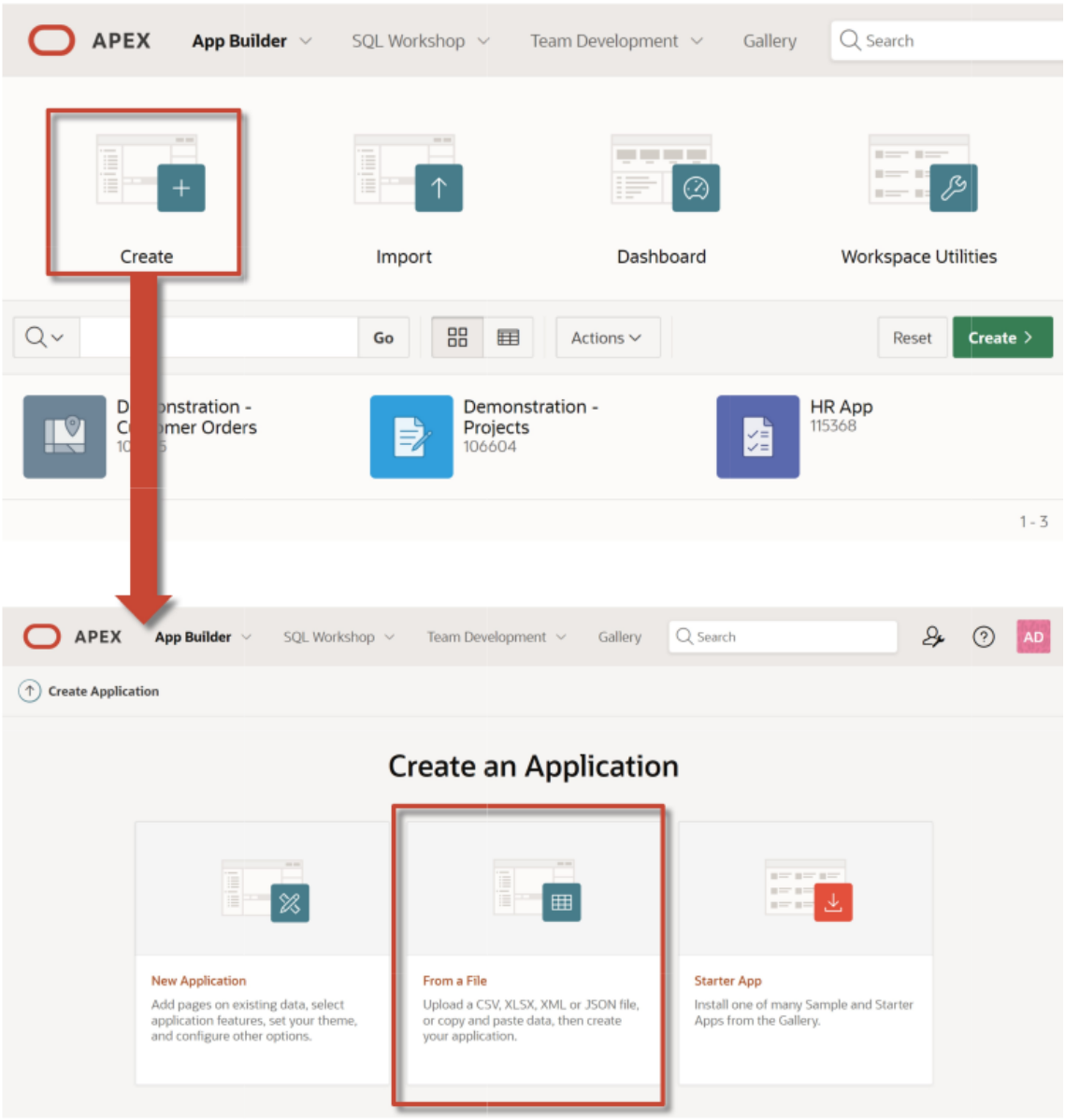
Dostęp do Data Workshop dla importów. Dostęp do Data Workshop można uzyskać na dwa alternatywne sposoby:

1. Proszę uruchomić Data Workshop z SQL Workshop, jak pokazano na rysunku 4.2.



Rysunek 4.2: Używanie Data Workshop do ładowania danych - dostęp przez SQL Workshop.

2. Alternatywnie, dostęp do Data Workshop można również uzyskać z poziomu APEX App Builder podczas tworzenia aplikacji, jak już wspomniano w rozdziale 3 (patrz rys. 4.3), używanie opcji “From a File” (patrz rysunek 4.1).

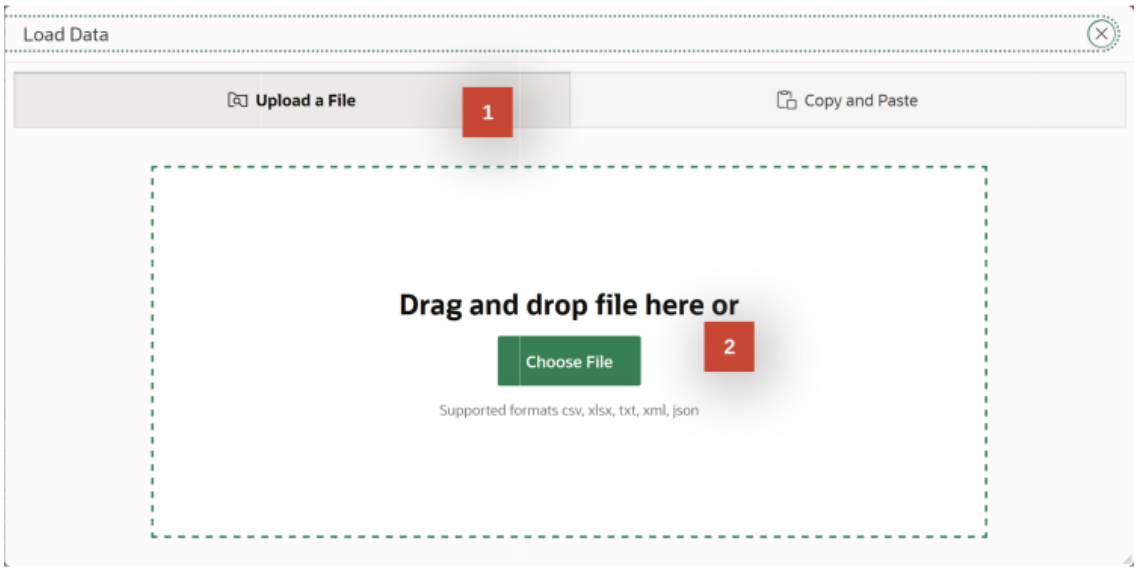


Rysunek 4.3: c

4.1.2 Kroki importowanie

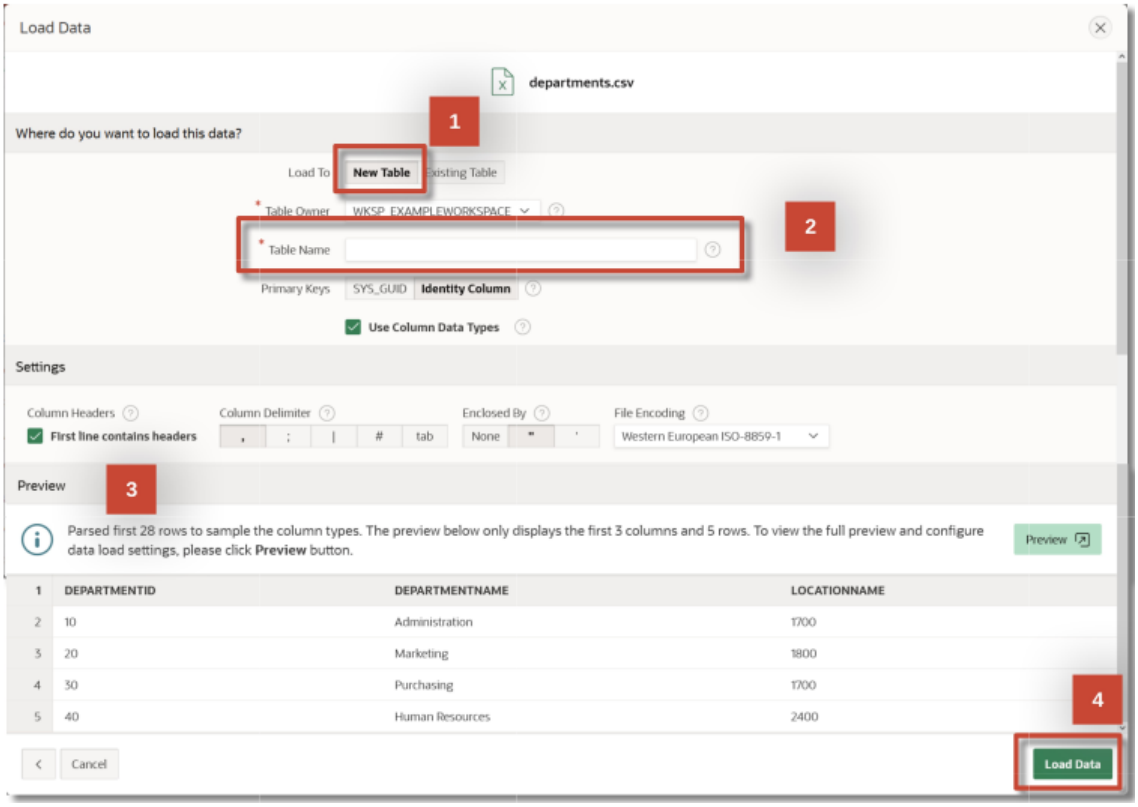
Poniżej pokrótce omówiono kroki niezbędne do zaimportowania danych, wraz z kilkoma ważnymi kwestiami przydatnymi do zrozumienia sposobu korzystania z “Load Wizard”:

Dostarczenie źródła danych. Pierwszym, dość naturalnym krokiem jest dostarczenie źródła danych, co można zrobić poprzez przesłanie (przeciągnij / upuść lub wybierz plik) plików delimited, XLSX, JSON lub XML lub poprzez proste kopiowanie / wklejanie danych delimited, tylko (patrz rys. 4.4). Proszę zauważyć, że to okno dialogowe pojawia się po kliknięciu przycisku *Load Data* (patrz rys. 4.2) lub poprzez kliknięcie przycisku *From a File* (patrz rys. 4.12).



Rysunek 4.4: Źródło danych.

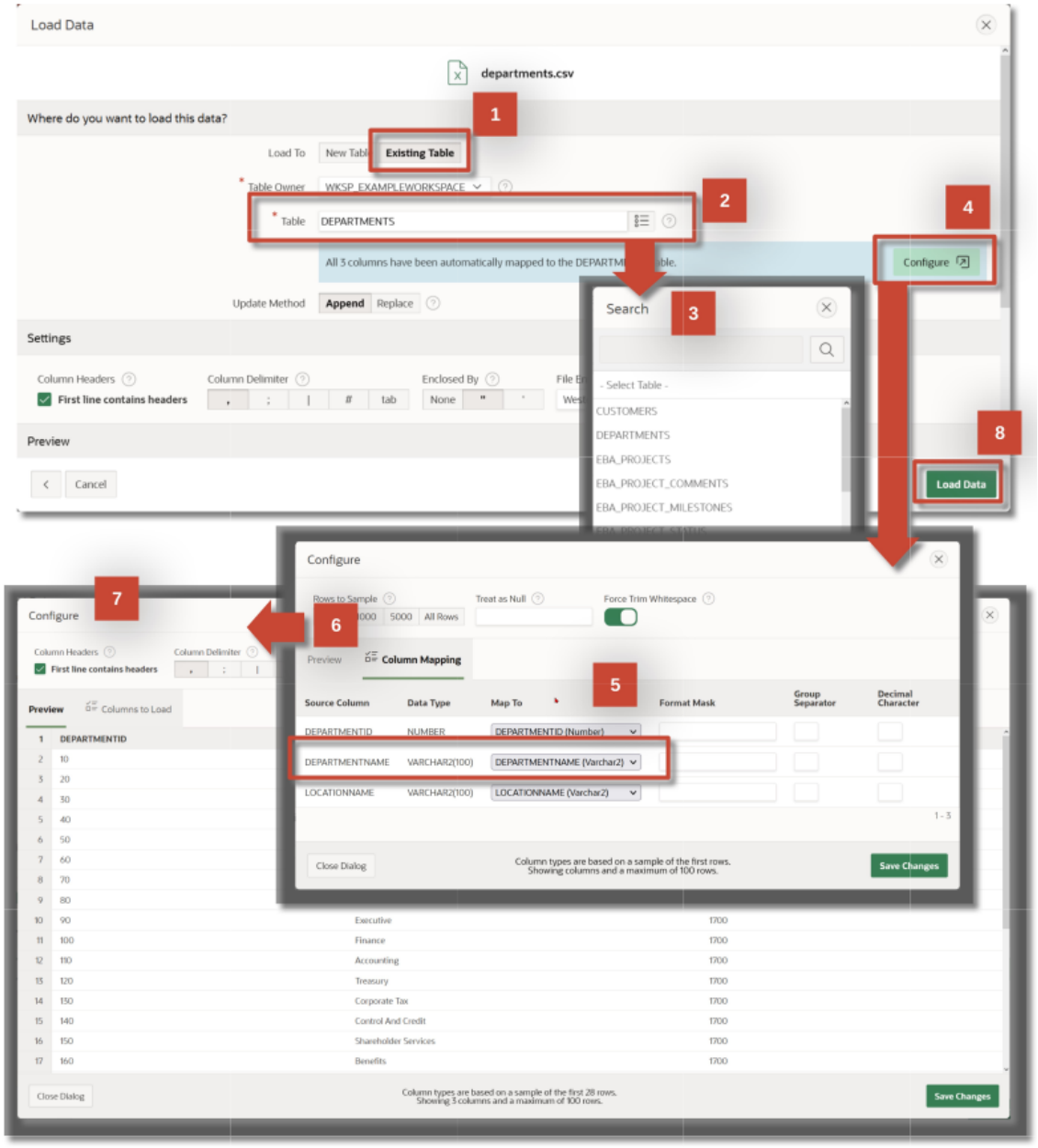
Konfiguracja mapowania danych do tabeli. Najważnieszym krokiem jest konfigutacja Load Wizard is the konfiguracja odwzorowania danych do tabeli, co jest zobrazowane na rysunkach 4.5 and 4.6, gdzie najważniejsze opcje konfiguracji są opisane numerami (1)-(4) and (1)-(8).



Rysunek 4.5: Wczytaj dane do istniejącej lub nowej tabeli.

1. **Wczytaj dane do istniejącej lub nowej tabeli.** Przede wszystkim należy zdecydować, czy dane powinny być przechowywane w już istniejącej tabeli DB, czy też należy użyć nowej tabeli DB (patrz rys. 4.5 (1) i rys. 4.6 (1)).

2. **Określ właściciela i nazwę tabeli.** W obu przypadkach należy określić właściciela tabeli i nazwę tabeli (patrz rys. 4.5 (2) i rys. 4.6 (2)).
3. **Ustawienia dla plików rozdzielonych.** *Sekcja ustawień* (patrz rys. 4.5 (3)) pozwala na trzy różne ważne konfiguracje specyficzne dla plików delimitowanych (ĆSV"). Opcja konfiguracji \textit{Nagłówki kolumn} pozwalają określić, czy pierwszy wiersz danych zawiera nazwy kolumn. *Column Delimiter* można dowolnie wybrać, jak widać w sekcji "Ustawieniańa rysunku 4.5 (3) umożliwiając Oracle *wyodrębnienie struktury z pliku zewnętrznego* – proces, który nazywa się *parsowanie (parsing)*, jest warunkiem wstępnym mapowania danych do tabeli. Za pomocą ustawienia *Enclosed by* można określić początkową i końcową granicę wartości danych. Jeśli określą Państwo znak ogranicznika, program Data Workshop zignoruje białe znaki występujące przed początkową i końcową granicą wartości danych. Można również użyć tej opcji, aby otoczyć wartość danych określonym znakiem ogranicznika.
4. **Sample Preview for Parsed Data.** Oracle automatycznie analizuje dane, jak już wspomniano, w celu wyodrębnienia struktury danych, a tym samym wykrycia np. odpowiednich typów danych. Na dole rysunku 4.5 (4) pokazana jest mała próbka (do 10 kolumn i 5 wierszy) wyniku tego procesu parsowania.
5. **Rozszerzony podgląd i dalsze konfiguracje.** Po kliknięciu przycisku "*Preview*"wyświetlane jest maksymalnie 100 kolumn i 100 wierszy oraz oferowane są dalsze możliwości konfiguracji, pozwalające teraz również na zmianę automatycznie sugerowanych typów danych i wybór, które z kolumn powinny zostać faktycznie załadowane. W przypadku, gdy w kroku (1) zdecydowano się załadować dane do istniejącej tabeli, należy wybrać tabelę z listy dostępnych (por. Rysunek 4.6 (2) i (3). Następnie, klikając przycisk *Ćonfigure-button"* (por. Rysunek 4.6 (4), można *determine the column mapping*, tj. które z automatycznie zidentyfikowanych kolumn załadowanych danych powinny zostać zmapowane do których kolumn w tabeli DB. Jak widać na rysunku 4.6 (5) i (6), kolumna `DEPARTMENTNAME` o typie danych `VARCHAR2`, która została automatycznie wyodrębniona z zewnętrznego pliku danych podczas parsowania, jest teraz mapowana na kolumnę tabeli DB o nazwie "DEPARTMENTNAMEó tym samym typie danych.

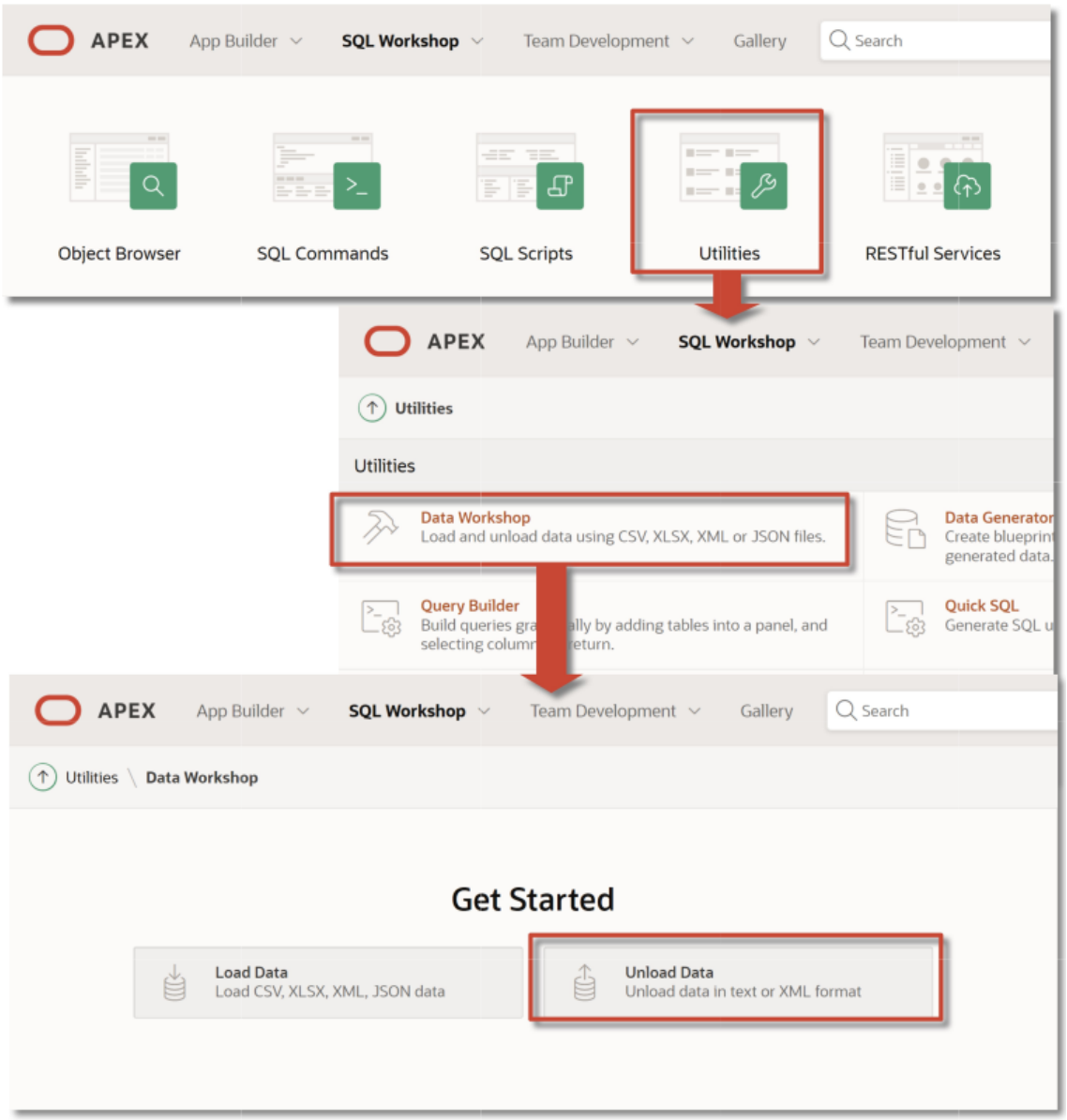


Rysunek 4.6: Ładowanie danych do istniejącej tabeli.

6. **Load Data.** Wreszcie, po kliknięciu przycisku *"Load Data"* (por. Rysunek 4.6 (8)), generowana jest nowa tabela DB (w przypadku wybrania tej opcji w (1)), a dane są ładowane do nowej lub istniejącej tabeli, przy czym okno dialogowe ładowania informuje, ile wierszy zostało załadowanych. Proces ładowania danych przebiega w tle, co jest korzystne, jeśli przesyłany jest większy plik, ponieważ okno dialogowe można zamknąć. Podczas tego procesu ładowania może się również zdarzyć, że napotkane zostaną pewne *błędne wiersze*, których nie można wstawić do tabeli docelowej, ponieważ np. jedna lub druga wartość jest sprzeczna z pewnym typem danych. W takim przypadku te błędne wiersze są automatycznie zapisywane w tabeli *error table* i mogą być przetwarzane ręcznie. Użytkownik jest również informowany przez okno dialogowe ładowania o ewentualnych błędnych wierszach. Wynikową tabelę (i ewentualną tabelę błędów) można teraz przeglądać, przechodząc do "Object browser" lub można natychmiast kontynuować tworzenie aplikacji internetowej.

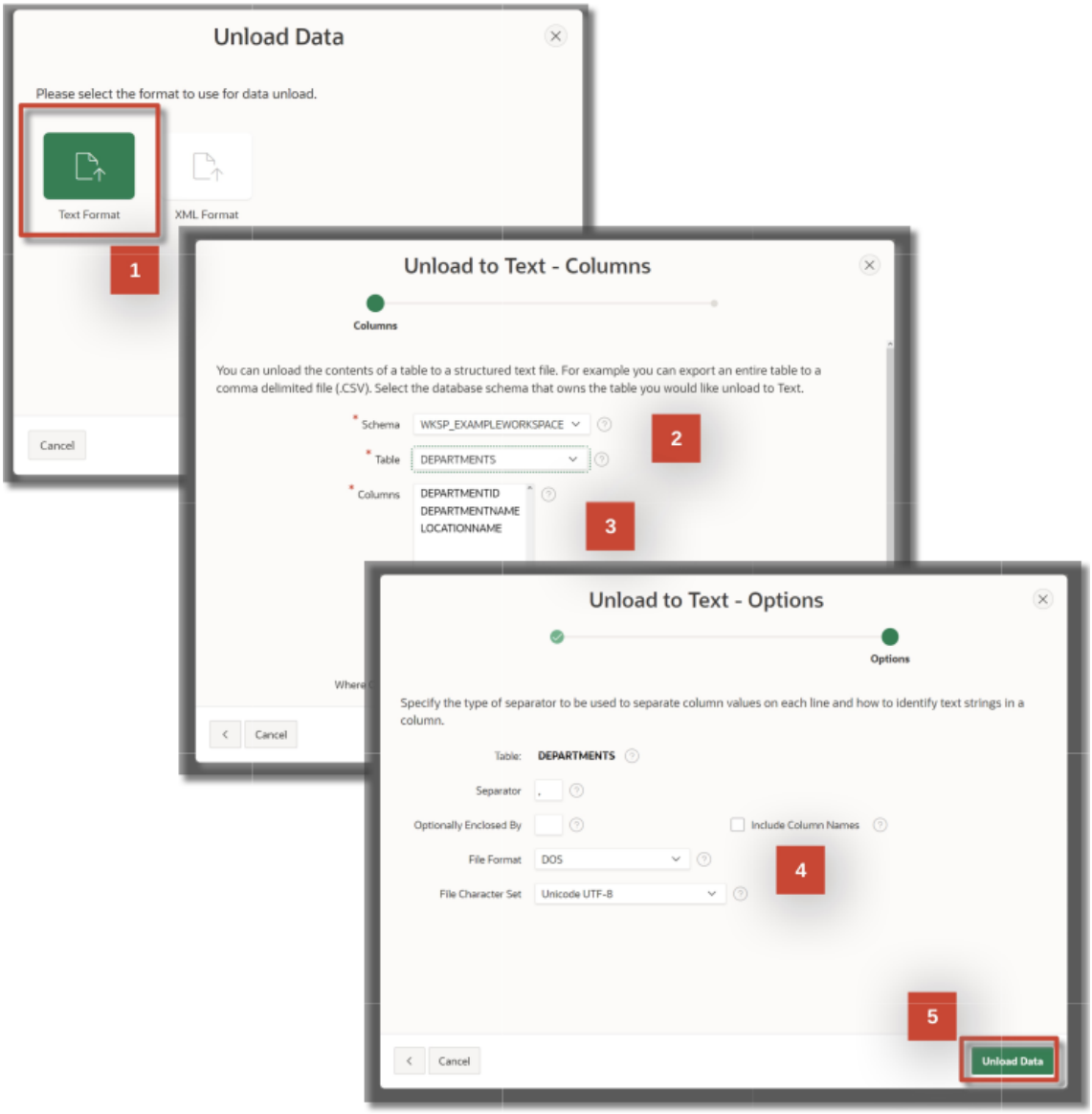
4.1.3 Eksportowanie danych

Dostęp do “Export Wizard” programu Data Workshop można uzyskać w taki sam sposób, jak opisano w sekcji 4.1.1 dla “Kreatora ładowania” (patrz rys. 4.7).



Rysunek 4.7: Eksportowanie (“rozładowywanie”) danych.

W pierwszym kroku należy wybrać format pliku, przy czym Data Workshop umożliwia eksport danych do plików zewnętrznych sformatowanych w formacie CSV lub XML (por. rysunek 4.8 (1)). Kolejnym krokiem jest wybranie tabeli oraz odpowiednich kolumn, których dane mają zostać wyeksportowane (por. Rysunek 4.8 odpowiednio (2) i (3)). W przypadku, gdy jako format eksportu wybrano XML, teraz można rozpocząć faktyczny eksport. W przypadku CSV można wybrać kilka dalszych opcji, określając separator między różnymi wierszami, czy nazwy wierszy powinny być zawarte w pliku wyjściowym i czy format danych powinien być DOS lub UNIX (por. Rysunek 4.8 (4)), a następnie proszę aktywować opcję "Unload Data", aby zapisać eksport danych (por. Rysunek 4.8). (5)).

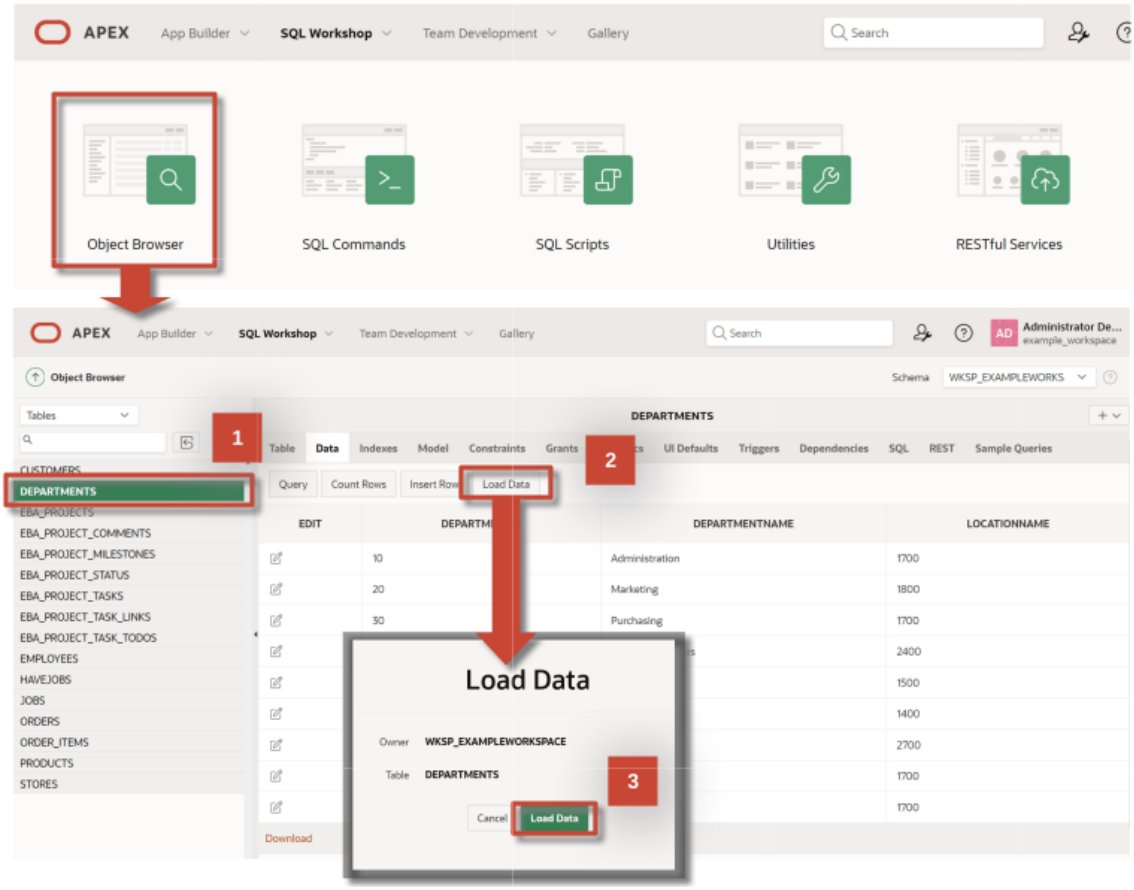


Rysunek 4.8: Kreator rozładowywania danych "Unload Data Wizard".

Jeśli chodzi o DOS vs. UNIX, proszę zauważyć, że istnieje subtelna różnica między tymi dwiema opcjami: Pliki DOS mają inne zakończenia linii niż pliki utworzone w systemie Unix/Linux. DOS używa powrotu karetki i podawania linii (“\texttt{\textbackslash{}r\textbackslash{}n}”) jako zakończenia linii, podczas gdy Unix używa tylko podawania linii (“\texttt{\textbackslash{}n}”). Dlatego należy zachować ostrożność podczas przesyłania plików między komputerami z systemem Windows i Unix, aby upewnić się, że zakończenia linii są ustawione prawidłowo.

4.2 Importowanie i eksportowanie danych przy użyciu “Object Browser”

Inną alternatywą jest użycie “Przeglądarki obiektów” SQL Workshop do importowania danych z i eksportowania danych do pliku, przy czym można to zrobić w sposób tabelaryczny. W tym celu należy najpierw wybrać odpowiednią tabelę (patrz rys. 4.9 (1)).

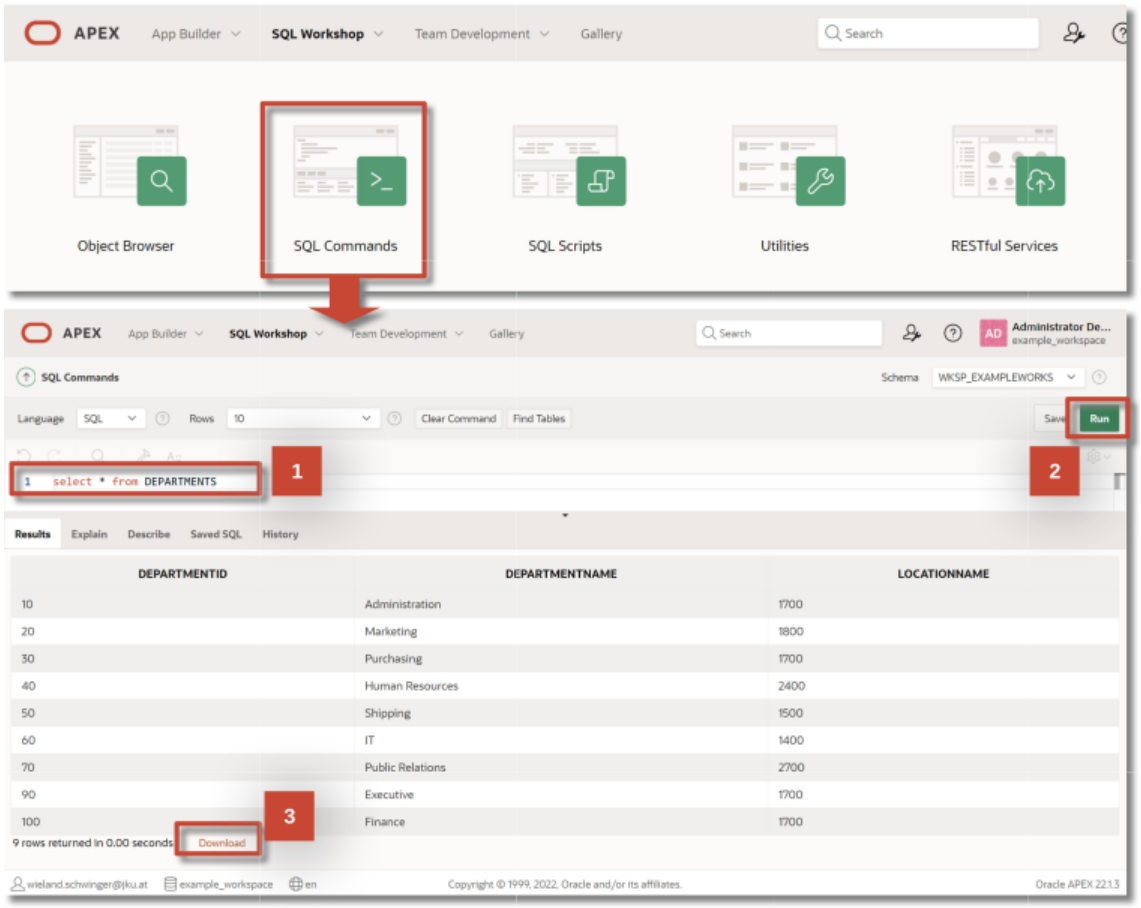


Rysunek 4.9: Import/eksport danych tabeli przy użyciu przeglądarki obiektów.

Jeśli teraz dane mają zostać zaimportowane, po kliknięciu przycisku “Załaduj dane” pojawi się to samo okno dialogowe, co po wybraniu opcji “Załaduj dane” (por. rysunek 4.9 (2), (3)), jak już opisano w sekcji 4.1.1 - obsługiwane są również te same formaty plików. Z drugiej strony, *eksport* danych można po prostu wykonać, klikając przycisk *“Download-button”* u dołu strony, co prowadzi do natychmiastowego wygenerowania pliku CSV zawierającego dane tabeli DB.

4.3 Eksportowanie wyniku polecenia SQL

Podobnie jak w przypadku eksportu wszystkich danych przechowywanych w tabeli, można również wyeksportować pewien podzbiór tych danych - lub nawet dane przechowywane w kilku tabelach - jednocześnie. Mówiąc prościej, można wyeksportować zestaw wyników dowolnego złożonego zapytania SQL (zob. rys. 4.10 (1))

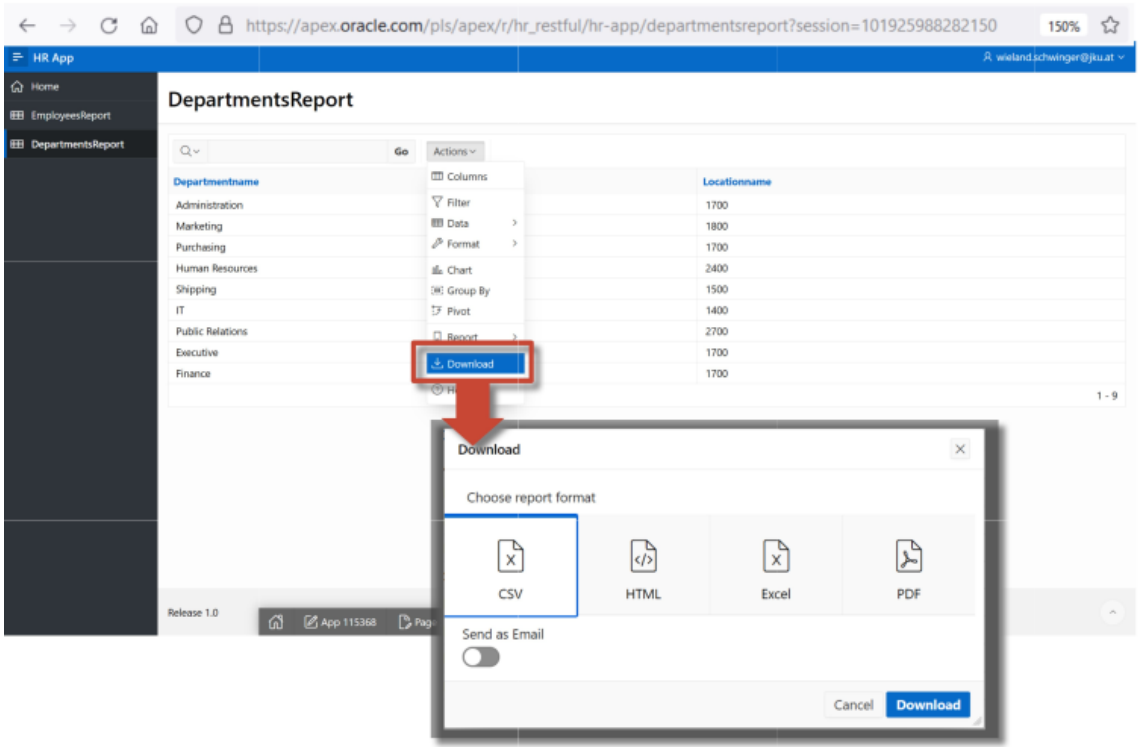


Rysunek 4.10: Eksport danych wyniku zapytania SQL.

W tym celu, po wykonaniu zapytania w narzędziu "Polecenie SQL"programu SQL Workshop i wyświetleniu zestawu wyników (por. rysunek 4.10), zestaw ten można pobrać, klikając przycisk "Pobierz"(por. rysunek 4.10). (2)), ten zestaw wyników można pobrać, klikając przycisk "Pobierz"(por. Rysunek 4.10). (3)).

4.4 Eksportowanie danych z raportu aplikacji

Jak omówiliśmy w rozdziale 3, tworzenie aplikacji internetowej za pomocą APEX App Builder wymaga wybrania określonego typu strony dla każdej strony dodawanej do aplikacji internetowej. W przypadku wybrania np. typu strony Żaport interaktywny", dane wyświetlane przez ten raport w czasie wykonywania mogą zostać wyeksportowane. Zostało to zilustrowane na rysunku 4.11, przedstawiającym przykładową stronę raportu interaktywnego "DepartmentsReport".

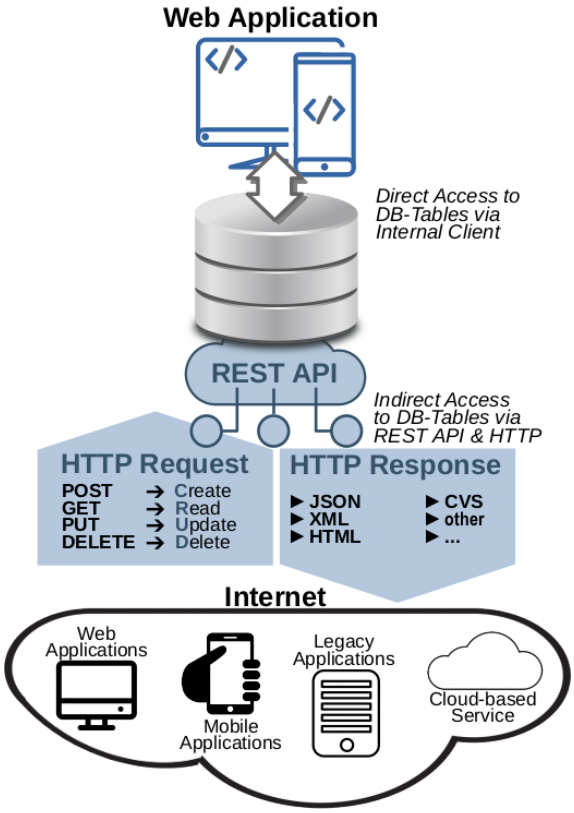


Rysunek 4.11: Eksport danych z raportu aplikacji.

Możliwe formaty plików to CSV, HTML, PDF i Excel, możliwe jest również wysyłanie danych pocztą elektroniczną.

4.5 Umożliwienie wymiany danych za pomocą usług RESTful

Wymiana danych warstwy DB naszej aplikacji internetowej poprzez umożliwienie klientom zewnętrznym, takim jak inne aplikacje internetowe, aplikacje mobilne, starsze aplikacje lub systemy chmurowe, dostępu online do naszych danych może być łatwo zrealizowana przy użyciu tak zwanych *RESTful services*. *REST (Representational State Transfer)* to w rzeczywistości *wzorzec architektoniczny* zaproponowany przez Roya Fieldinga i Richarda Taylora w 2000 roku [2] w celu zapewnienia interoperacyjności między dowolnymi systemami przez Internet. Umożliwia on wysyłanie zapytań i manipulowanie danymi *bez potrzeby bezpośredniego dostępu* do bazowych tabel DB. Rysunek 4.12 przedstawia ogólny przegląd podstawowego wzorca architektonicznego REST.



Rysunek 4.12: Usługi RESTful do wymiany danych - podstawowa architektura.

Aby zrealizować taki *bezpośredni* dostęp zgodnie z wzorcem architektonicznym REST, usługi RESTful muszą być tworzone w kategoriach zestawu *APIs (Application Programming Interfaces)* na Państwa tabelach DB. Bez bezpośredniego łączenia się z bazową bazą danych, te interfejsy API umożliwiają zewnętrznym systemom bezpieczną interakcję z danymi poprzez wysyłanie zapytań, wstawianie, aktualizowanie lub usuwanie danych. Definicje usług RESTful utworzonych w Oracle APEX są przechowywane w repozytorium *Oracle REST Data Services (ORDS)* i są określane jako *ORDS-based REST Services*.

4.5.1 Wzorzec architektoniczny REST

Jak już wspomniano, usługa jest opisywana jako RESTful, gdy jest zgodna z założeniami REST. Chociaż pełne omówienie REST wykracza poza zakres tego rozdziału, usługa RESTful ma następujące cechy (proszę zauważyć, że ogólna charakterystyka REST opisana w [2] została nieznacznie dostosowana do naszego kontekstu DB):

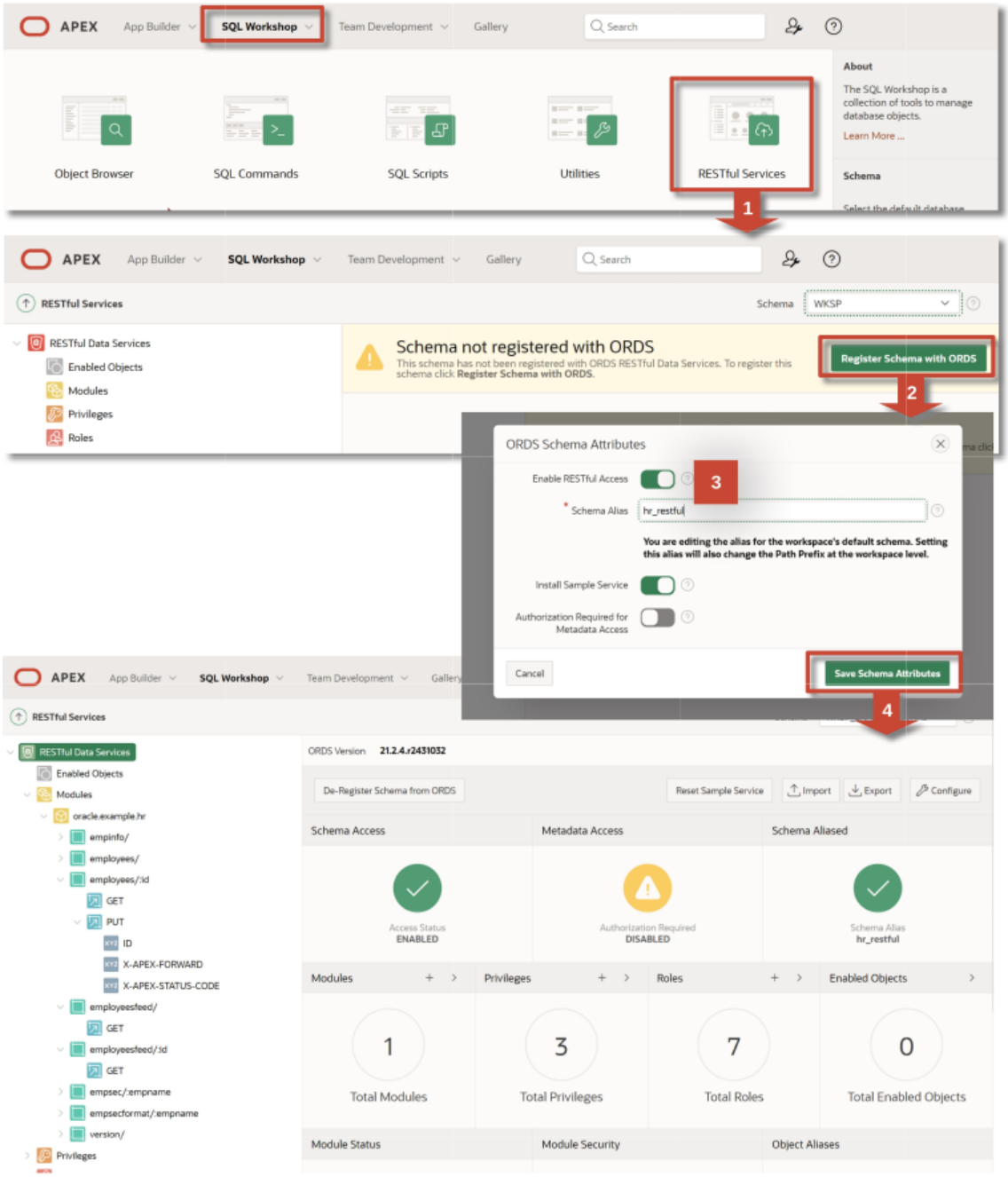
1. **Tabele DB dostarczane jako zasoby dla usług**. Usługa RESTful jest modelowana jako zestaw zasobów, w naszym przypadku na podstawie Państwa tabel DB. Zasoby te są identyfikowane przez adresy URL i dostępne za pośrednictwem protokołów internetowych `HTTP` lub `HTTPS`. protokołów sieciowych.
2. **Operacje wysyłane za pośrednictwem HTTP-Requests**. Niewielki zestaw operacji jest używany do radzenia sobie z tymi zasobami w kategoriach żądania `HTTP` składającego się z `POST`, `GET`, `PUT`, `DELETE`, przypominając w ten sposób operacje CRUD (Create, Read, Update, Delete) .
3. **Usługi są bezstanowe**. Usługi RESTful są *stateless*, tj. żaden kontekst klienta nie jest przechowywany na serwerze między żądaniami - stan sesji jest przechowywany w całości na kliencie. Oznacza to, że każde żądanie od klienta do serwera musi zawierać wszystkie informacje niezbędne serwerowi do zrozumienia żądania i nie może korzystać z żadnego

kontekstu przechowywanego na serwerze. Główną zaletą jest to, że można tworzyć usługi wielokrotnego użytku i skalowalne, którymi można zarządzać i aktualizować bez wpływu na system, nawet w czasie wykonywania. Wadą jest to, że może to zmniejszyć wydajność sieci poprzez zwiększenie powtarzających się danych (narzut na interakcję) wysyłanych w serii żądań, ponieważ dane te nie mogą pozostać na serwerze we współdzielonym kontekście.

4. **Odpowiedzi HTTP**. Żądania do usługi RESTful zawsze wywołują *odpowiedź*. Ta odpowiedź ma postać XML, JSON, HTML lub innego zdefiniowanego formatu. Odpowiedzi zawierają szczegóły dotyczące pewnego rodzaju zmian w danych bazowych, komunikaty o błędach i łącza hipertekstowe do innych powiązanych zasobów w zależności od operacji.

4.5.2 Włączanie schematu DB dla dostępu RESTful

Aby włączyć schemat DB dla dostępu RESTful, należy go najpierw zarejestrować w ORDS . Można to zrobić w prosty sposób, jak pokazano na rysunku 4.13, uzyskując dostęp do narzędzia usług RESTful w SQL Workshop (1), a następnie klikając przycisk Żegister Schema with ORDS"(2), wprowadzając dowolny alias używany w tworzeniu adresu URL odwołującego się do dowolnej usługi RESTful w schemacie (3) i wreszcie, po kliknięciu przycisku *Śave Schema Attributes-button"* (4) pojawi się ORDS RESTful Services Dashboard, używany do zarządzania wszystkimi usługami RESTful..



Rysunek 4.13: Włączanie schematu DB dla dostępu RESTful.

Jeśli w kroku (3) włączono opcję "Install Sample Service", zainstalowany zostanie również przykładowy *module* `oracle.example.hr` jest również zainstalowany, dostarczając przykładowe szablony *resource templates* i *resource handlers* (por. sekcja 4.5.3), które implementują kilka różnych operacji pobierania i wyświetlania informacji o pracownikach z tabeli Employees, jak widać na rysunku 4.15.

4.5.3 Moduły zasobów | Szablony | Programy obsługi

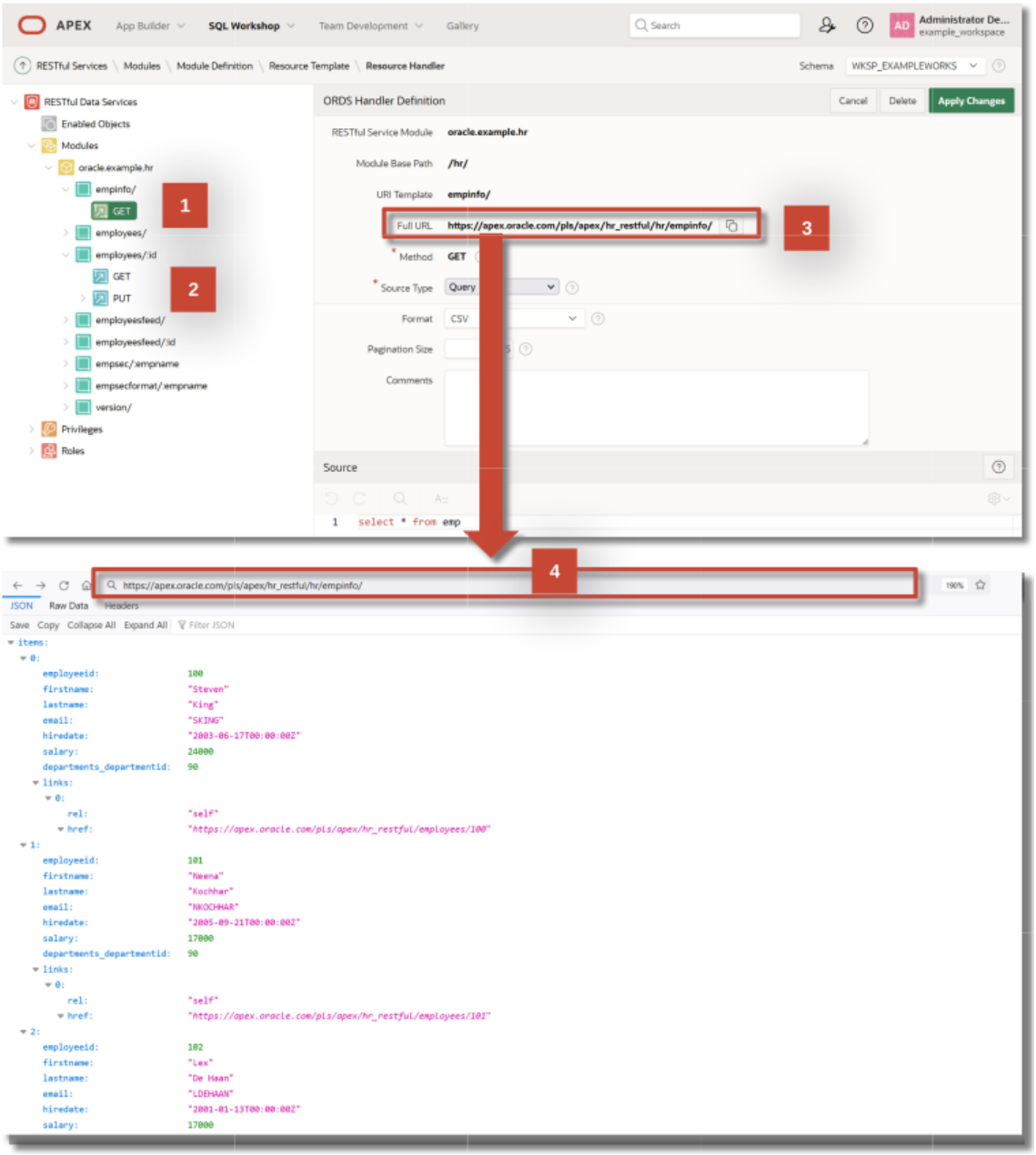
W ORDS usługi RESTful mogą być tworzone przy użyciu trzech podstawowych komponentów, obejmujących moduły zasobów, szablony zasobów i programy obsługi zasobów (por. rysunek 4.15 dla niektórych przykładów). Ogólne relacje między tymi podstawowymi elementami konstrukcyjnymi ilustruje diagram ER na rysunku 4.14.



Rysunek 4.14: Relacje między różnymi komponentami usług ORDS RESTful Services.

- **Moduły zasobów.** Moduł zasobów to nic innego jak kontener, który grupuje zestaw powiązanych usług RESTful, w oparciu o określony *Schemat DB z obsługą REST*. Schemat DB może zawierać kilka modułów zasobów. Moduł zasobów nie tylko zapewnia sposób na jednoznaczną identyfikację grupy, ale także definiuje unikalną ścieżkę bazową używaną w URI *do uzyskania dostępu do zestawu usług* zdefiniowanych w module. Na przykład moduł zasobów, który umożliwia dostęp do informacji o pracownikach, nosi nazwę `oracle.example.hr`, a wartość ścieżki bazowej dla usługi to `/hr/`, por. rysunek 4.15.
- **szablony zasobów**. Szablon zasobów definiuje indywidualną usługę, którą można wywołać, np. `employees/:id`, rysunek 4.15. Szablony zasobów są zawarte w modułach zasobów. Każdy szablon zasobów definiuje *URI Pattern*, gdzie można go uzyskać, np. ępminfo/"(Rysunek 4.15) i zawiera co najmniej jeden moduł obsługi zasobów (por. poniżej).
- **Obsługa zasobów.** Każdy Resource Handler implementuje jedną (i tylko jedną) z czterech różnych operacji HTTP, które przypominają tradycyjne operacje CRUD (Create, Read, Update, Delete) .
  - `POST` tworzy nowy zasób lub dodaje zasób do kolekcji, co jest równoważne instrukcji SQL `INSERT`.
  - `GET` pobiera reprezentację zasobu, równoważną instrukcji SQL `SELECT`.
  - `PUT` aktualizuje wartości istniejącego zasobu, co jest równoważne instrukcji SQL `UPDATE`.
  - `DELETE` usuwa istniejący zasób, co jest równoważne instrukcji SQL `DELETE`.

Należy zdefiniować Resource Handler dla każdej operacji powiązanej z tym samym Resource Template. Na przykład, aby zapewnić *operacja zwracająca dane* i *inna operacja przechowująca dane*, należy zdefiniować Resource Handler dla każdej operacji (por. Rys. 4.15 - dla szablonu zasobów `employees/:id` zdefiniowano dwie operacje `GET` (1) i `PUT` (2)).



Rysunek 4.15: Przykładowe usługi RESTful dla tabeli Pracownicy.

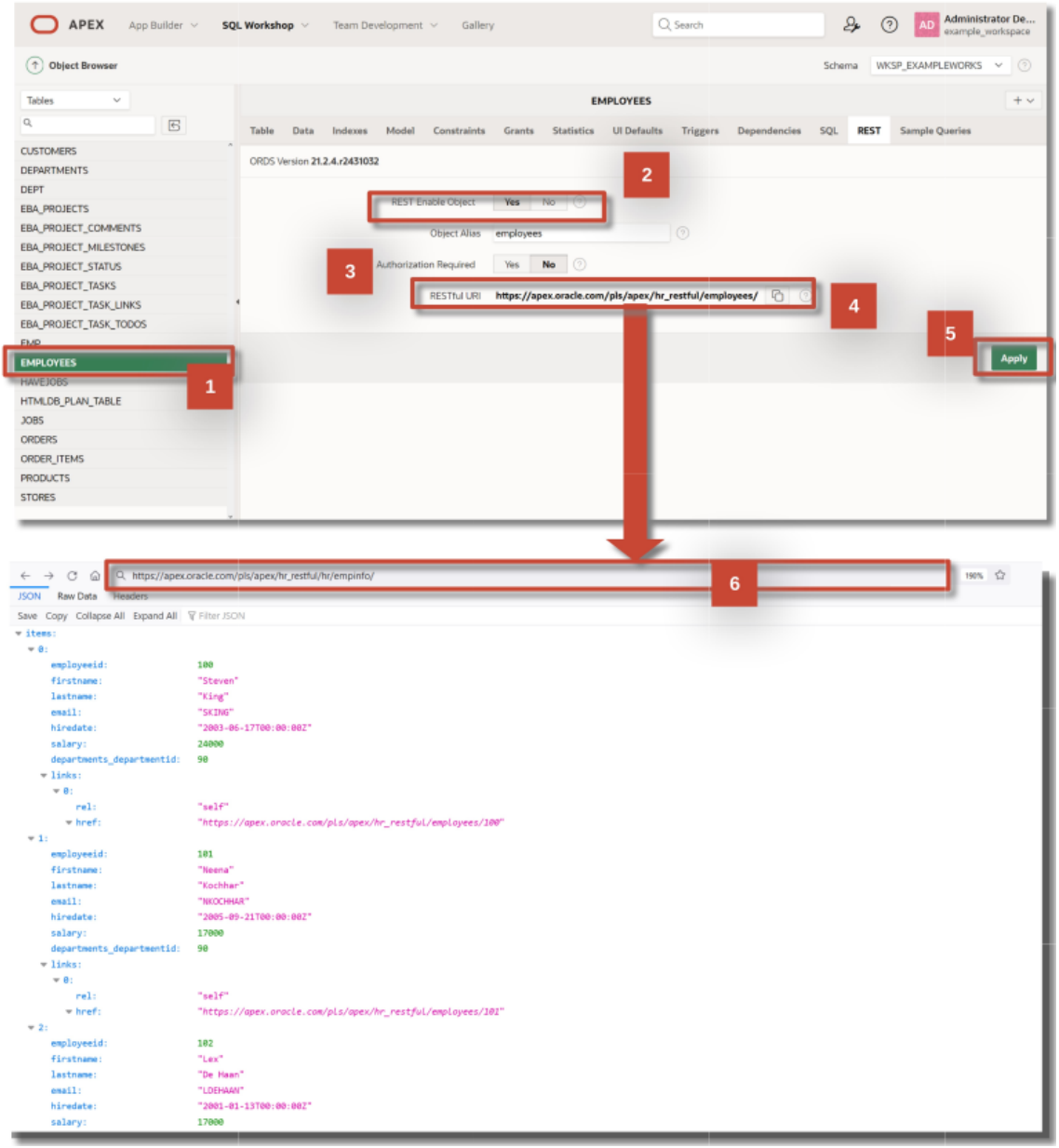
Podsumowując, rysunek 4.15 przedstawia przykładowy przegląd różnych składników usługi RESTful pod względem modułów zasobów, szablonów i procedur obsługi, w tym także wybór żądanego formatu odpowiedzi (CSV) i zapytania SQL implementującego typ operacji procedury obsługi zasobów. Po utworzeniu usługi RESTful można ją przetestować, po prostu wprowadzając wygenerowany adres URL dostępu w przeglądarce (por. Rys. 4.15 (3) i (4) odpowiednio)

4.5.4 Używanie ĄutoRESTźamiast ręcznego definiowania zasobów

Jak można było zauważyć w poprzedniej sekcji, dla każdego rodzaju operacji DB, które powinny być oferowane klientom zewnętrznym jako usługa RESTful w celu pośredniego dostępu do naszej bazy danych, konieczne jest wykonanie kilku kroków. W przypadku najprostszej formy zapytania, tj. pełnego skanowania tabeli, istnieje jednak również pewien rodzaj śkrótuźapewniany przez APEX w postaci tak zwanego *ĄutoREST-facility*. AutoREST to szybki i łatwy sposób na udostępnienie

tabel DB jako zasobów REST. Korzystając z funkcji AutoREST, tracą Państwo pewną elastyczność i możliwość dostosowywania, ale znacznie skraca to czas i wysiłek. AutoRest pozwala na szybką ekspozycję danych, ale umożliwia jedynie odpytywanie całej tabeli (tj. pełne skanowanie tabeli), mając stały format wyjściowy w postaci JSON.

AutoREST można wykorzystać w następujący sposób: Po włączeniu schematu dla dostępu RESTful (por. Sekcja 4.5.2), każda tabela DB może być indywidualnie zdefiniowana jako zasób przy użyciu Przeglądarki obiektów SQL Workshop, jak widać na Rysunku 4.16.



Rysunek 4.16: Włączanie usługi REST w przeglądarce obiektów.

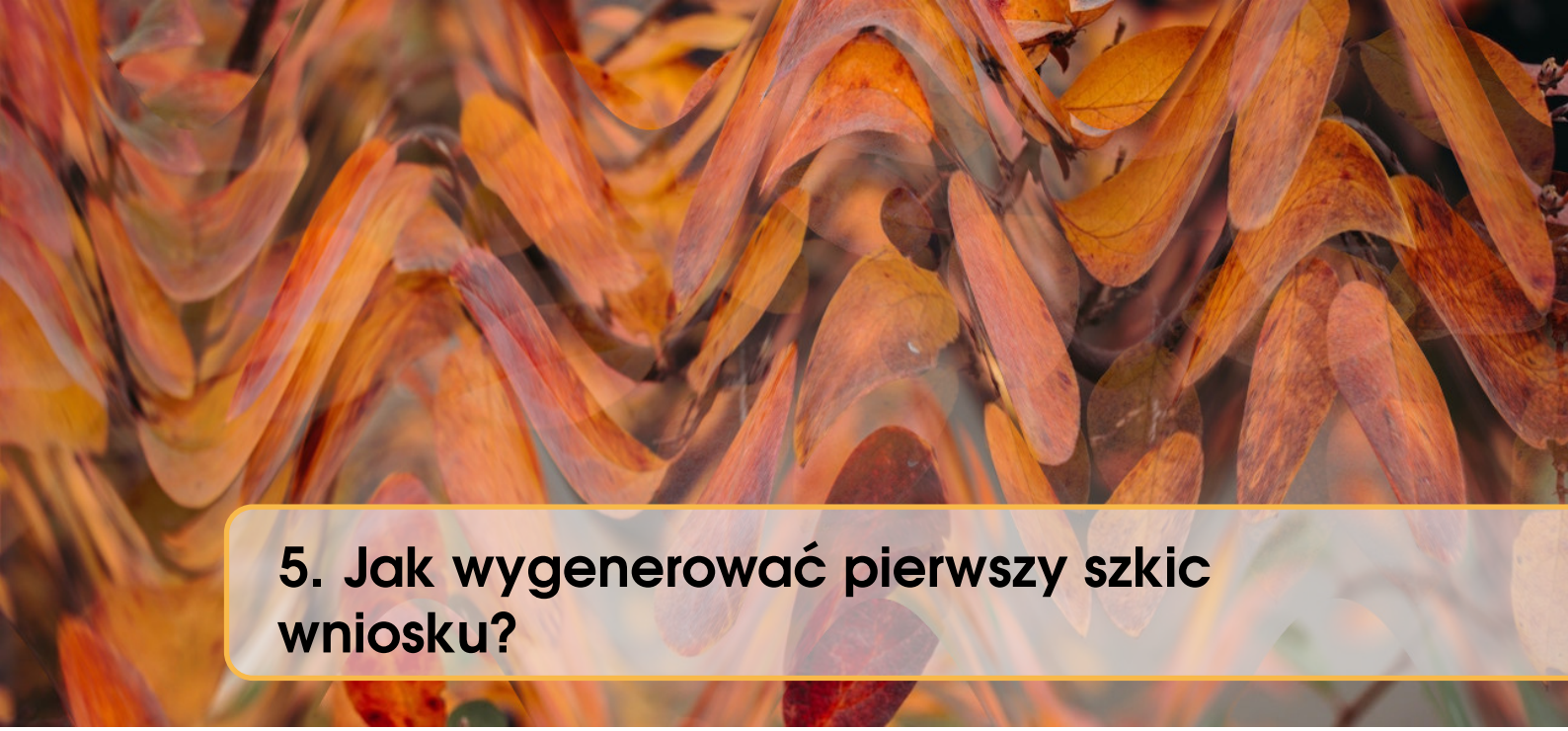
Po wybraniu odpowiedniej tabeli DB (1), kliknięciu zakładki REST (2), wybraniu wymaganej autoryzacji (3) i zastosowaniu wszystkich tych ustawień (5), REST został aktywowany dla tabeli i pojawił się automatycznie wygenerowany adres URL dostępu (4). Ponownie, ta usługa REST może być po prostu przetestowana poprzez wprowadzenie adresu URL w przeglądarce internetowej (6).

4.6 Pytania

1. Jakie są przypadki użycia do importowania danych za pomocą SQL Workshop i App Builder?
2. Jakie są różne sposoby eksportowania danych?
3. Do czego służy REST? Jaki jest jego cel lub korzyść?

4.7 Answers

1. Data Workshop umożliwia importowanie danych z plików zewnętrznych za pomocą "Kreatora" oraz importowanie danych do tabeli za pomocą "Przeglądarki obiektów". Import danych z Data Workshop można również zainicjować za pomocą App Builder.
2. Eksport danych do pliku w sposób tabelaryczny jest obsługiwany przez "Object browser" SQL Workshop. Data Workshop udostępnia "Kreator eksportu", który pozwala wybrać format pliku i tabele, a także odpowiednie kolumny, których dane mają zostać wyeksportowane.
3. Usługi RESTful zapewniają interoperacyjność między dowolnymi systemami za pośrednictwem Internetu, umożliwiając wyszukiwanie danych i manipulowanie nimi.



5. Jak wygenerować pierwszy szkic wniosku?

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

5.1 Dlaczego biznes potrzebuje aplikacji?

W dzisiejszym szybko zmieniającym się krajobrazie biznesowym aplikacje stały się kluczowym narzędziem dla firm, aby pozostać konkurencyjnymi i sprostać stale zmieniającym się wymaganiom klientów. Tworzenie aplikacji pozwala firmom usprawnić swoje operacje, zwiększyć wydajność i poprawić jakość obsługi klienta, co ostatecznie może prowadzić do większej rentowności.

Jednym z głównych powodów, dla których firmy tworzą aplikacje, jest automatyzacja procesów, które są obecnie wykonywane ręcznie. Przykładowo, firma zajmująca się handlem elektronicznym może opracować aplikację, która zautomatyzuje procesy zarządzania zapasami i realizacji zamówień. W ten sposób mogą zmniejszyć prawdopodobieństwo wystąpienia błędu ludzkiego oraz poprawić szybkość i dokładność swoich operacji.

Innym kluczowym powodem, dla którego firmy tworzą aplikacje, jest poprawa obsługi klienta. Aplikacje mogą być tworzone w celu zapewnienia klientom opcji samoobsługi, takich jak możliwość śledzenia zamówień lub aktualizacji informacji o koncie. Może to pomóc skrócić czas oczekiwania klientów i zwiększyć ich zadowolenie.

Ponadto aplikacje mogą pomóc firmom w gromadzeniu cennych danych o ich klientach, które można wykorzystać do ulepszenia strategii marketingowych i opracowania ukierunkowanych kampanii reklamowych. Na przykład sprzedawca detaliczny może opracować aplikację, która śledzi wzorce zakupów klientów, umożliwiając im wysyłanie spersonalizowanych rekomendacji i promocji do swoich klientów.

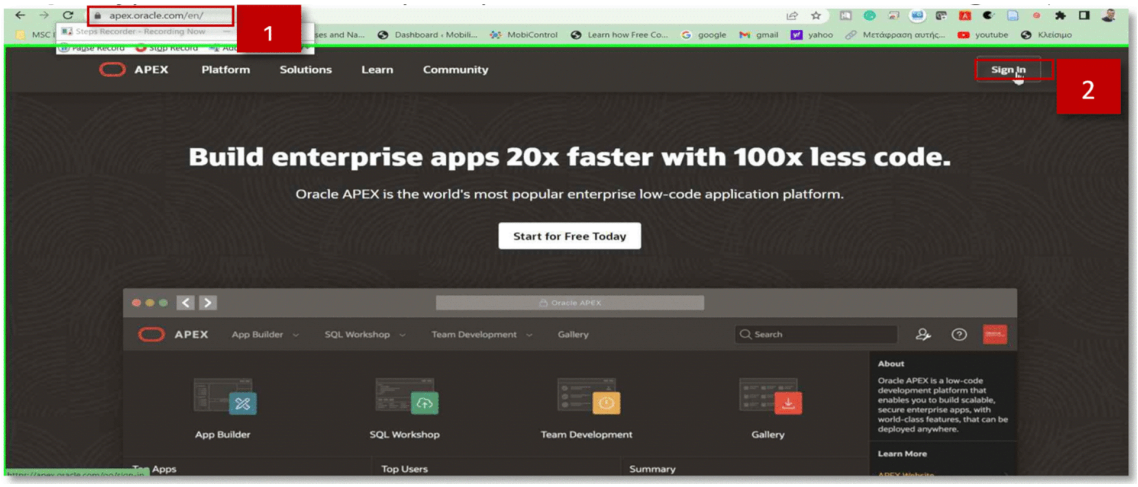
Ogólnie rzecz biorąc, tworzenie aplikacji stało się silną potrzebą biznesową, ponieważ pozwala firmom działać wydajniej, zapewniać lepsze doświadczenia klientów i uzyskiwać wgląd w ich zachowania. Ponieważ świat biznesu staje się coraz bardziej cyfrowy, znaczenie tworzenia aplikacji będzie nadal rosło.

5.2 Konfiguracja środowiska ORACLE APEX

Celem tego przykładu jest zapoznanie Państwa ze środowiskiem ORACLE APEX, poprzez poprowadzenie Państwa przez konfigurację środowiska APEX, zademonstrowanie przykładu warsztatu SQL bazy danych employees-departments-projects oraz zilustrowanie sposobu tworzenia aplikacji internetowej APEX w oparciu o model danych.

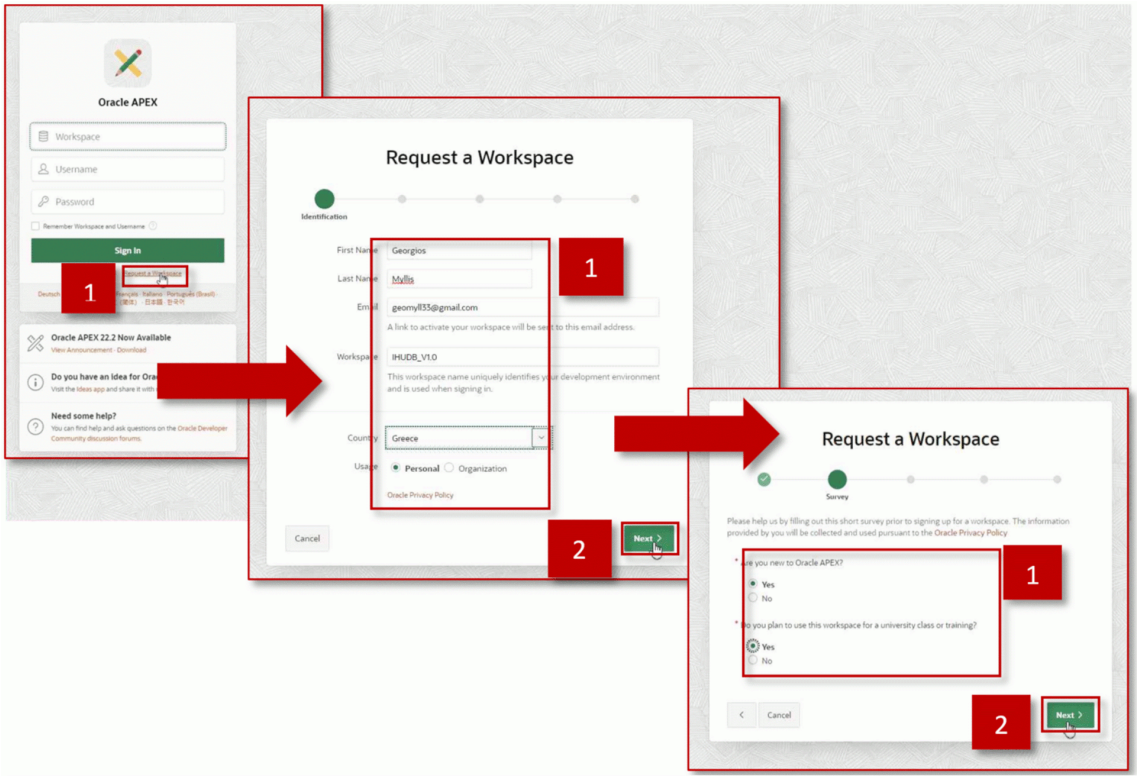
Aby wdrożyć ten przykład, należy wykonać następujące kroki:

Krok 1: Proszę otworzyć przeglądarkę internetową i wpisać adres "https://apex.oracle.com/en/". Proszę kliknąć przycisk **Sign In**, jak pokazano na rysunku 5.1.



Rysunek 5.1: Proszę wpisać link APEX i zalogować się.

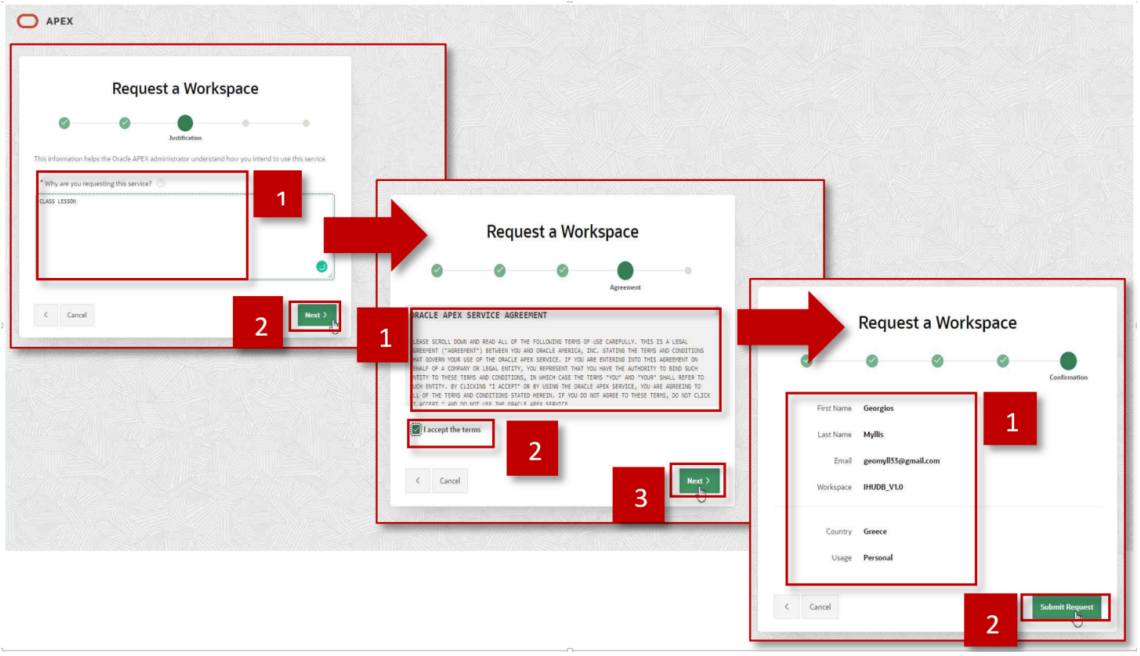
Krok 2: Proszę kliknąć przycisk **Request a Workspace**. W wyskakującym oknie, które się pojawi, proszę wypełnić pola wymagane do identyfikacji i kliknąć **Next**. Następnie proszę odpowiedzieć na pytania ankietowe w kolejnym wyskakującym oknie i kliknąć **Next**, jak pokazano na rysunku 5.2.



Rysunek 5.2: Proszę poprosić o Oracle APEX Workspace początkowe kroki.

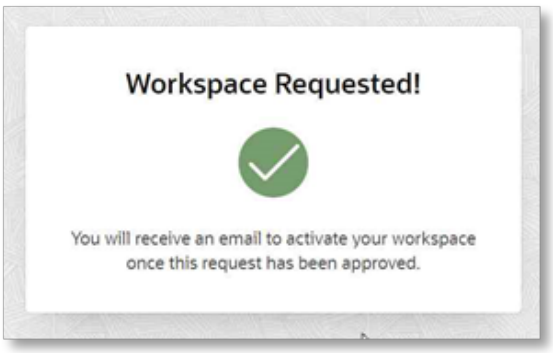
Request an Oracle APEX Workspace initial steps.Step 3: Proszę odpowiedzieć na informacje wymagane do uzasadnienia i kliknąć **Next**. W kolejnym wyskakującym okienku proszę przeczytać

warunki umowy, a następnie zaakceptować je, klikając **Next**. W kolejnym wyskakującym okienku proszę sprawdzić wprowadzone dane pod kątem błędów, a następnie kliknąć **Prześlij żądanie**, jak pokazano na rysunku 5.3.



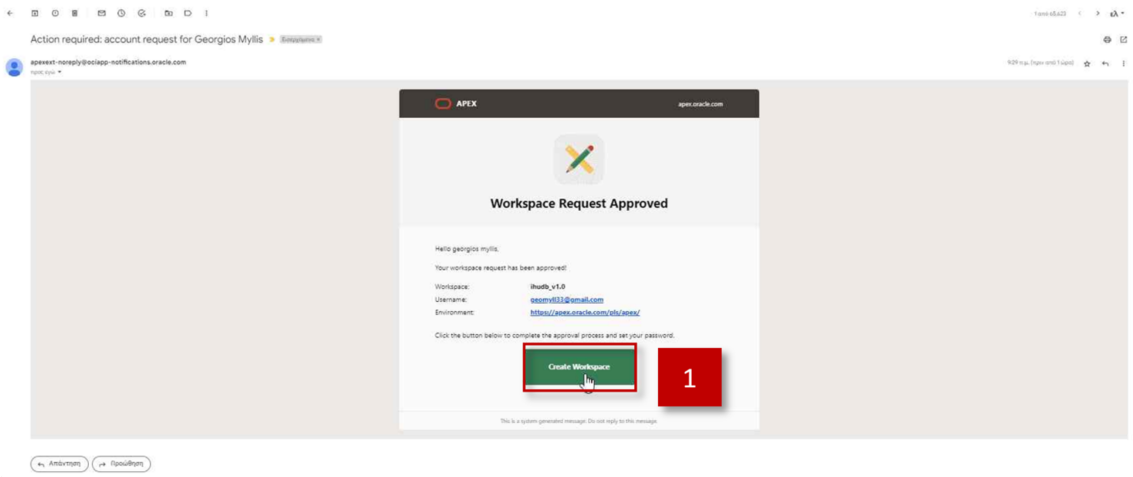
Rysunek 5.3: Proszę poprosić o ukończenie Oracle APEX Workspace.

Krok 4: Po przesłaniu żądania pojawi się wyskakujące okienko potwierdzające, że Państwa obszar roboczy został zamówiony, a na podany adres e-mail zostanie wysłana wiadomość aktywacyjna, jak pokazano na rysunku 5.4.



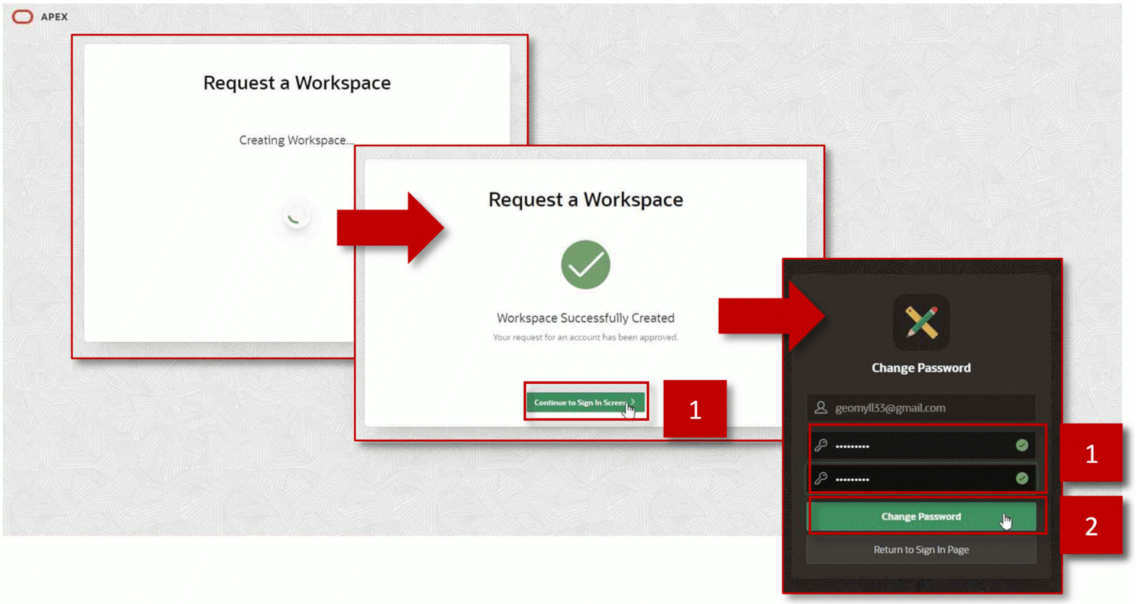
Rysunek 5.4: Oracle APEX Workspace zatwierdzony.

Krok 5: Proszę sprawdzić swoją pocztę e-mail pod kątem wiadomości od APEX. Państwa wniosek musi zostać zatwierdzony, zanim będzie można kontynuować. Po zatwierdzeniu proszę kliknąć **Create Workspace**, jak pokazano na rysunku 5.5.



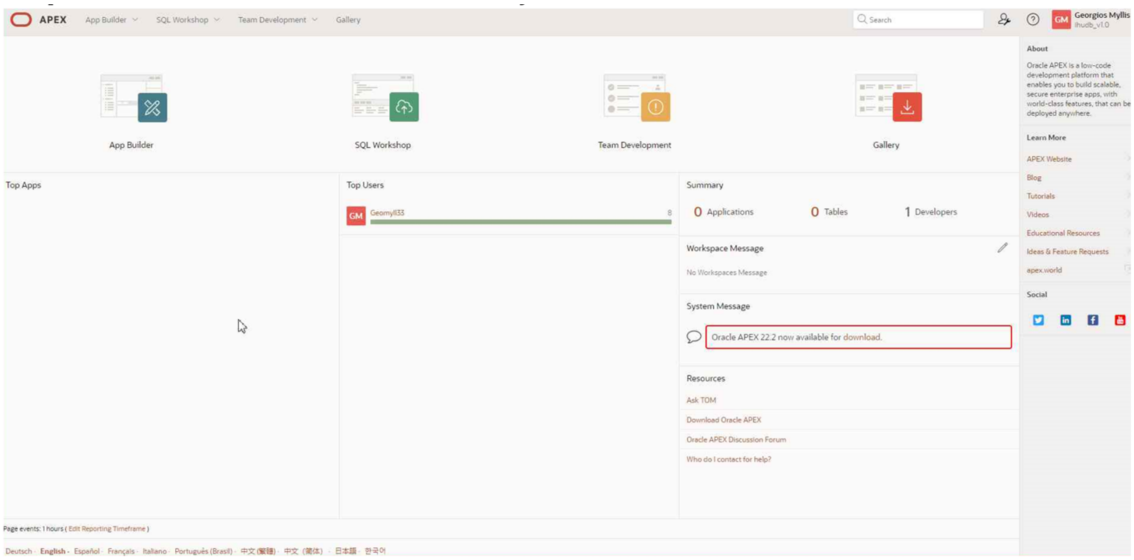
Rysunek 5.5: E-mail aktywacyjny od APEX.

Krok 6: Proszę poczekać na zakończenie procesu tworzenia przestrzeni roboczej. Po zakończeniu pojawi się wyskakujące okienko informujące, że obszar roboczy został pomyślnie utworzony. Proszę kliknąć **Continue to Sign-in Screen**. Pojawi się nowe okno z prośbą o zmianę hasła przed zalogowaniem. Proszę ustawić nowe hasło, potwierdzić je i kliknąć **Change Password**, jak pokazano na rysunku 5.6.



Rysunek 5.6: Proszę ustawić nowe hasło dla Workspace APEX.

Krok 7: Środowisko Oracle APEX jest teraz gotowe do użycia, jak pokazano na rysunku 5.7.



Rysunek 5.7: Workspace APEX environment.

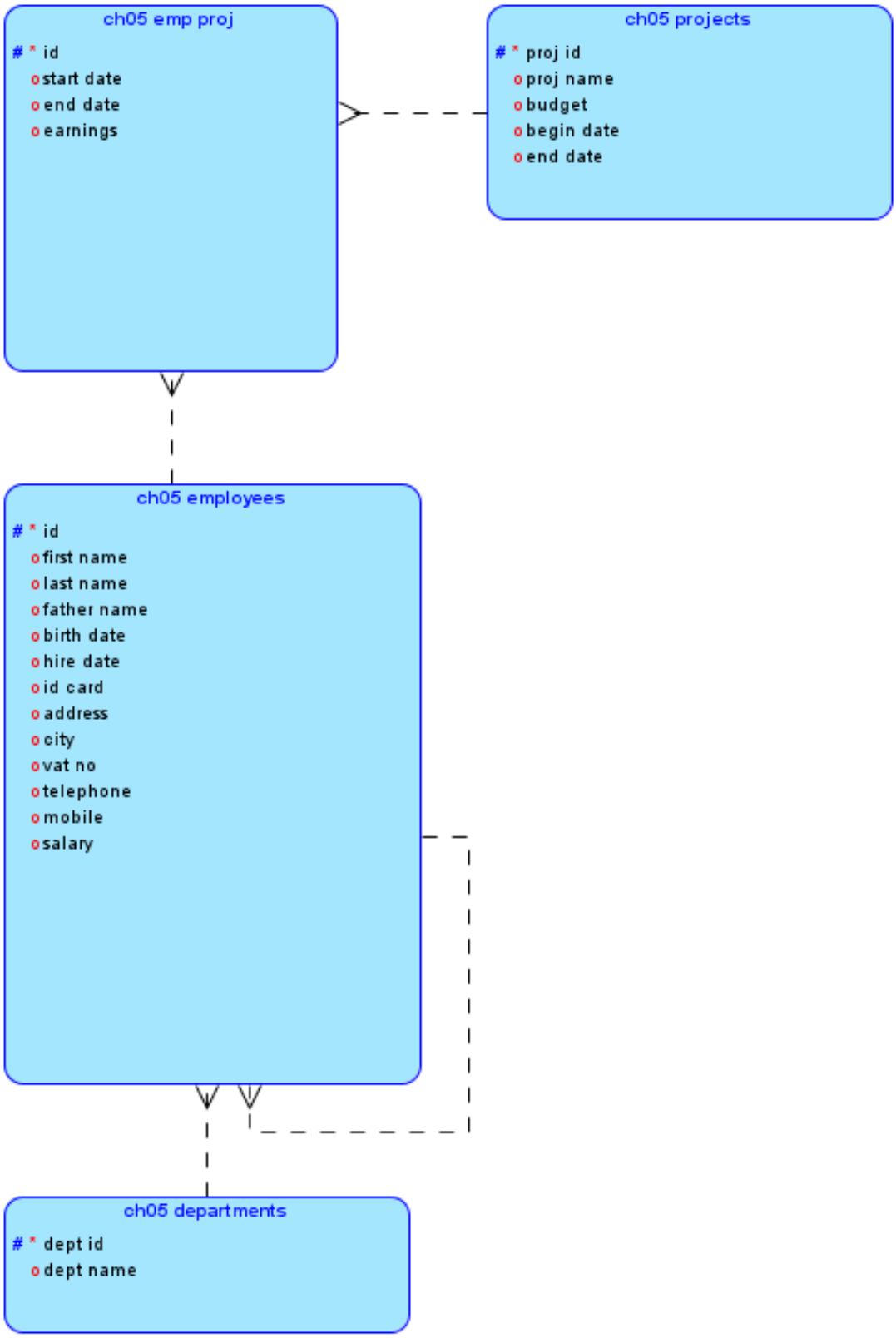
5.3 Tworzenie przykładów warsztatów SQL

Aby utworzyć bazę danych w środowisku APEX, dostępnych jest wiele podejść. Jedną z opcji jest wykorzystanie Object Browser, graficznego narzędzia, które umożliwia użytkownikom tworzenie tabel i ich odpowiednich pól poprzez wypełnienie wstępnie zaprojektowanego szablonu.

Inną alternatywą jest wykorzystanie poleceń SQL, które umożliwiają użytkownikom tworzenie tabel i innych obiektów bazy danych poprzez wykonywanie instrukcji SQL. Metoda ta oferuje większą elastyczność i kontrolę nad strukturą bazy danych, choć wymaga bardziej zaawansowanej wiedzy.

Z drugiej strony, Quick SQL to narzędzie dostarczane przez Oracle APEX, które ułatwia szybkie generowanie skryptów SQL do tworzenia tabel bazy danych, ograniczeń i przykładowych danych. Quick SQL upraszcza proces tworzenia obiektów bazy danych bez konieczności ręcznego pisania skomplikowanych skryptów SQL.

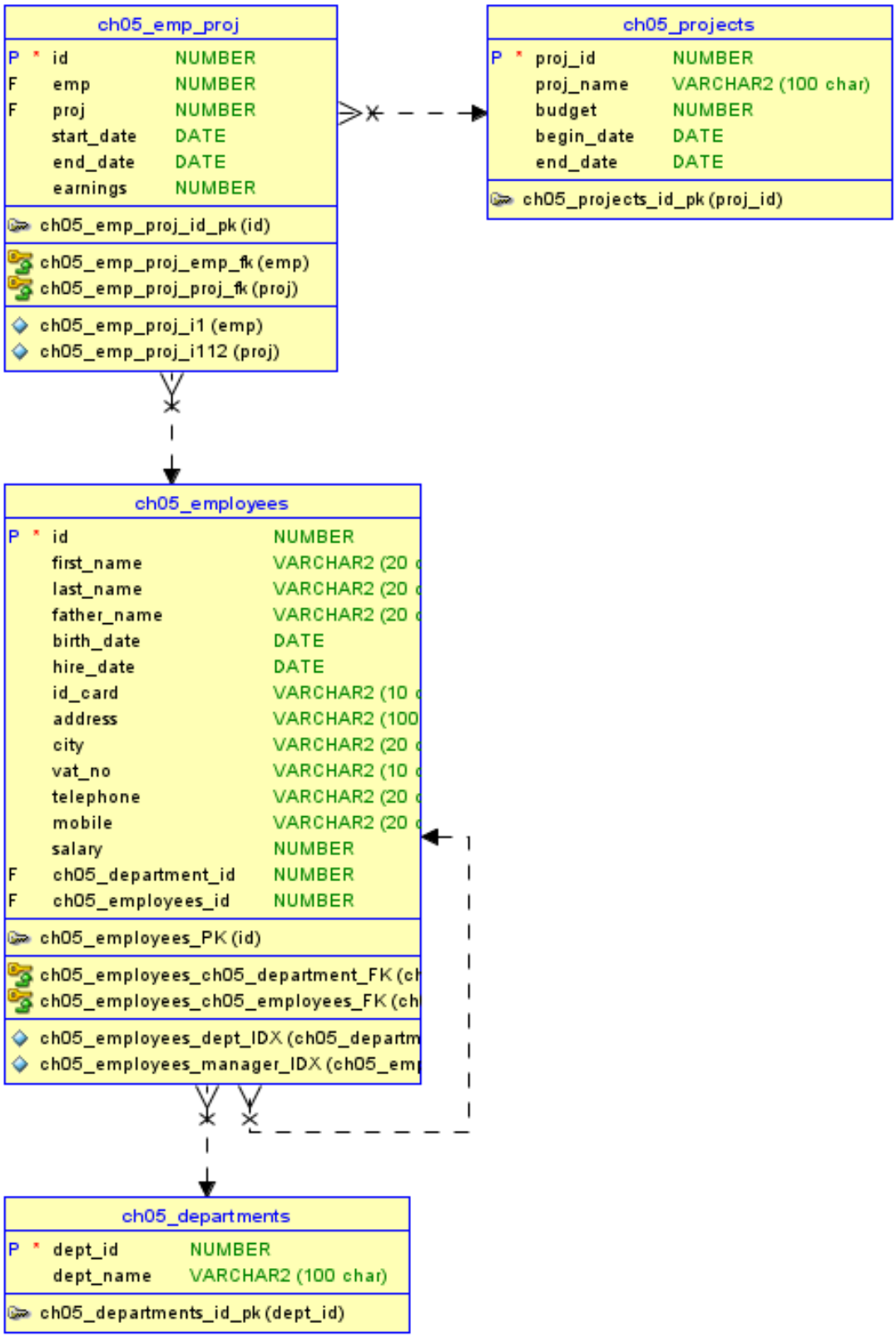
W niniejszym przykładzie wykorzystamy metodę Quick SQL. Skrypt został wygenerowany w oparciu o model logiczny przedstawiony na rysunku 5.8.



Rysunek 5.8: Model logiczny naszego przykładu działającego HR.

Zgodnie z modelem relacyjnym wyprowadzonym z modelu logicznego przy użyciu Oracle Data

Modeler (ODM), jak pokazano na rysunku 5.9, możemy przystąpić do implementacji następującej struktury w bazie danych.



Rysunek 5.9: Model relacyjny naszego przykładowego działającego HR.

5.3.1 Quick SQL

Dostarczony kod wydaje się być zestawem instrukcji do generowania przykładowych danych za pomocą narzędzia Quick SQL. Każda sekcja zaczyna się od nazwy tabeli, po której następuje słowo kluczowe "/insert"i liczba wskazująca żądaną liczbę wierszy do wstawienia do tabeli..

```
ch05_departments /insert 5
    dept_name vc100

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

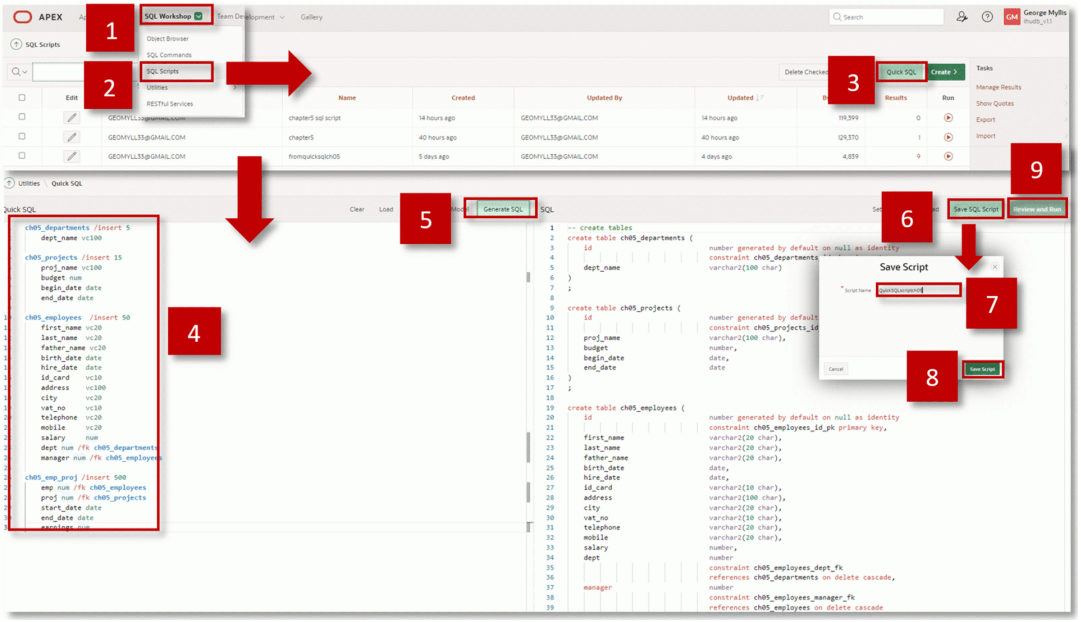
ch05_employees  /insert 50
    first_name vc20
    last_name  vc20
    father_name vc20
    birth_date date
    hire_date  date
    id_card    vc10
    address    vc100
    city       vc20
    vat_no     vc10
    telephone  vc20
    mobile     vc20
    salary     num
    dept num /fk ch05_departments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

Niniejsza instrukcja zawiera schemat i przykładowy plan generowania danych dla bazy danych związanej z działami, projektami, pracownikami i ich powiązaniami.

Krok 1: Jak pokazano na rysunku 5.10, wykonajnastępujące kroki:

1. Proszę przejść do sekcji **SQL Workshop**.
2. W opcjach proszę wybrać **SQL Scripts**.
3. Proszę kliknąć **Quick SQL**.
4. Proszę wstawić dostarczony kod Quick SQL do wyznaczonego obszaru..
5. Proszę kliknąć na **Generuj SQL**. Wygenerowany kod SQL zostanie wyświetlony w oknie Oracle SQL Output.
6. Proszę kliknąć **Save script SQL**.
7. Proszę wprowadzić nazwę skryptu.
8. Proszę kliknąć **Save script**.
9. Proszę kontynuować, klikając **Review and Run**.



Rysunek 5.10: Wstawianie kodu Quick SQL do obszaru roboczego APEX.

5.3.2 Skrypt SQL

Jak wspomniano wcześniej, poniżej znajduje się skrypt SQL wygenerowany z Quick SQL, który tworzy schemat bazy danych dla przykładowego systemu zarządzania zasobami ludzkimi. Schemat obejmuje tabele dla działów, pracowników, projektów i relacji pracownik-projekt. Dodatkowo zawiera indeksy i ograniczenia, aby zagwarantować spójność danych i integralność referencyjną.

```
-- 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);
```

Tabela ch05_departments przechowuje informacje o działach, w tym unikatowy identyfikator (id) i nazwę (dept_name). Tabela ch05_employees przechowuje informacje o pracownikach, w tym unikatowy identyfikator (id), imię (first_name), nazwisko (last_name), imię ojca (father_name), datę urodzenia (birth_date), datę zatrudnienia (hire_date), numer karty ID (ID_card number), adres (address), miasto (city), numer VAT (VAT_number), numer telefonu (phone number), numer telefonu komórkowego (mobile number), wynagrodzenie (salary) oraz identyfikatory działu (dept) i kierownika (manager). Tabela ch05_projects przechowuje informacje o projektach, w tym unikalny identyfikator (id), nazwę, budżet, datę rozpoczęcia i zakończenia. Tabela ch05_emp_proj przechowuje informacje o relacjach pracowników z projektami, w tym unikatowy identyfikator (id),

identyfikatory pracownika (emp) i projektu (proj), datę rozpoczęcia_date, datę zakończenia_date i zarobki.

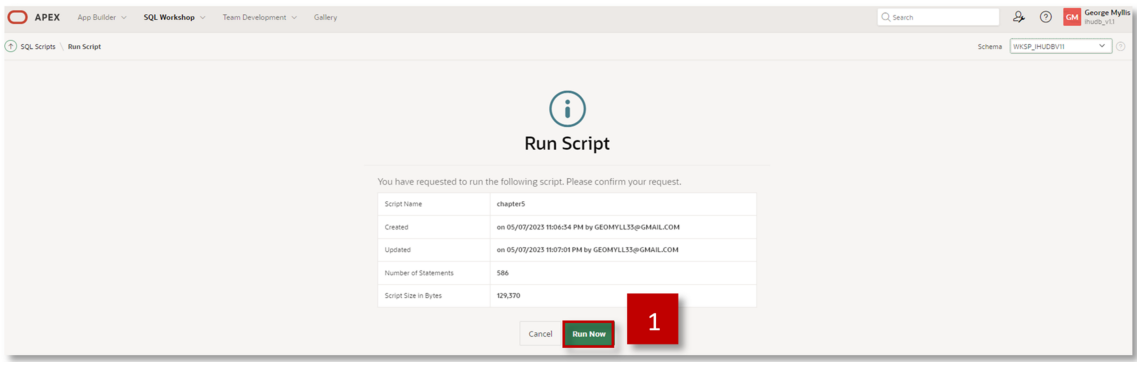
Indeksy są tworzone w tabeli ch05_emp_proj dla kolumn employee i project w celu poprawy wydajności podczas wykonywania zapytań do tabeli na podstawie tych kolumn. Indeksy są również tworzone w tabeli ch05_employee dla kolumn department i manager z tego samego powodu.

Na koniec dodawane są ograniczenia w celu zapewnienia spójności danych i integralności referencyjnej między tabelami. Ograniczenia ch05_emp_proj_emp_fk i ch05_emp_proj_proj_fk zapewniają, że kolumny employee i project w tabeli ch05_emp_proj odwołują się do prawidłowych wartości id odpowiednio w tabelach ch05_employees i ch05_project.

Ograniczenia ch05_employees_dept_fk i ch05_employee_manager_fk zapewniają, że kolumny departure i manager w tabeli ch05_employee odwołują się do prawidłowych wartości id odpowiednio w tabelach ch05_departments i ch05_employees.

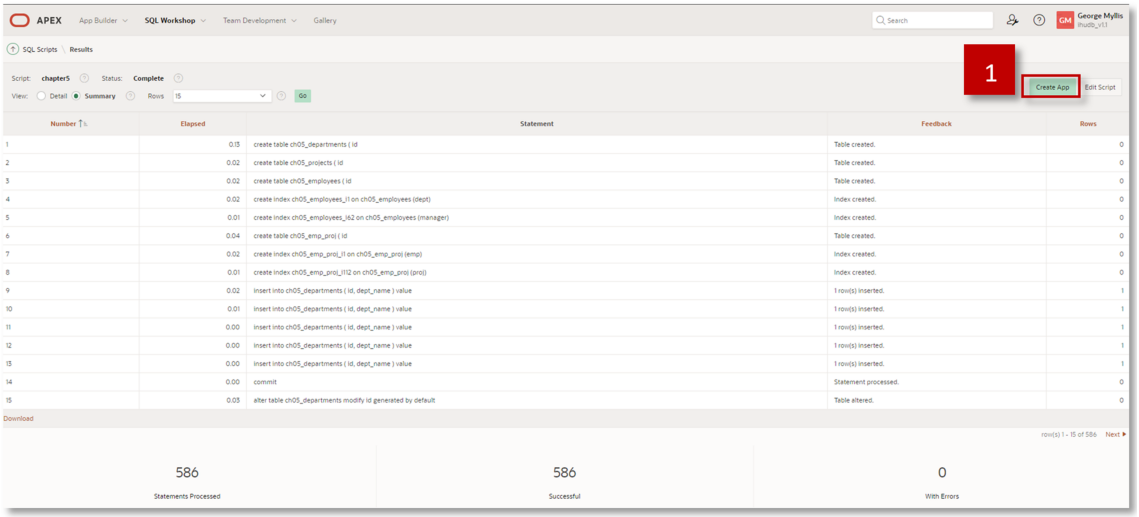
Dodatkowo klauzula ON DELETE CASCADE określa, że jeśli wiersz w tabeli, do której się odwołujemy, zostanie usunięty, wszystkie powiązane wiersze w tabeli, do której się odwołujemy, również zostaną usunięte, aby zachować integralność referencyjną.

Krok 2: Aby zainicjować żądaną akcję, proszę zapoznać się z rysunkiem 5.11 i kliknąć przycisk **Run Now**.



Rysunek 5.11: Uruchom skrypt SQL

SKrok 3: Po pomyślnym przetworzeniu wszystkich wyciągów bez napotkania żadnych błędów, proszę przejść do rysunku 5.12 i kliknąć przycisk **Create App**.



Rysunek 5.12: Tworzenie procesu uruchamiania aplikacji.

5.4 Część aplikacji oparta na danych

Sekcja aplikacji oparta na danych koncentruje się na wykorzystaniu mocy danych do kierowania spostrzeżeniami, podejmowania decyzji i interakcji z użytkownikami. W tej sekcji dane zajmują centralne miejsce, umożliwiając użytkownikom eksplorację, analizę i interakcję z informacjami w dynamiczny i intuicyjny sposób.

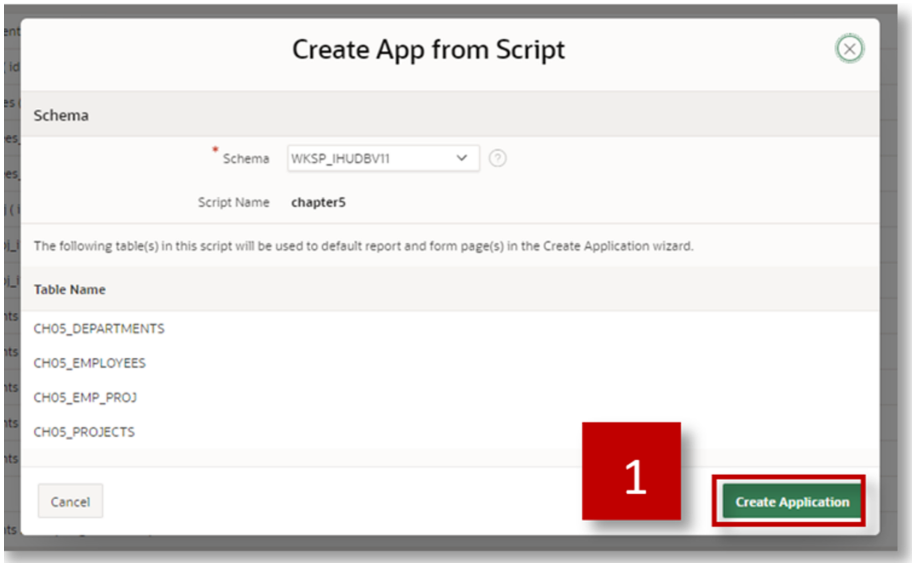
Głównym celem tej sekcji jest zapewnienie użytkownikom kompleksowego widoku danych, umożliwiając im uzyskanie cennych spostrzeżeń i podejmowanie świadomych decyzji. Dzięki różnym technikom wizualizacji danych, takim jak wykresy, grafy i interaktywne pulpity nawigacyjne, użytkownicy mogą łatwo zrozumieć złożone zestawy danych i zidentyfikować wzorce, trendy i wartości odstające.

Dodatkowo, sekcja oparta na danych zawiera solidne możliwości zarządzania danymi, umożliwiając użytkownikom filtrowanie, sortowanie i wyszukiwanie określonych punktów danych lub segmentów. Zaawansowane funkcje, takie jak agregacja danych, drążenie i eksport danych, dodatkowo poprawiają komfort użytkowania, umożliwiając głębszą eksplorację i analizę.

Sekcja ta podkreśla również znaczenie integralności i jakości danych. Obejmuje ona mechanizmy walidacji danych, obsługi błędów i czyszczenia danych, aby zapewnić, że prezentowane informacje są dokładne, wiarygodne i spójne.

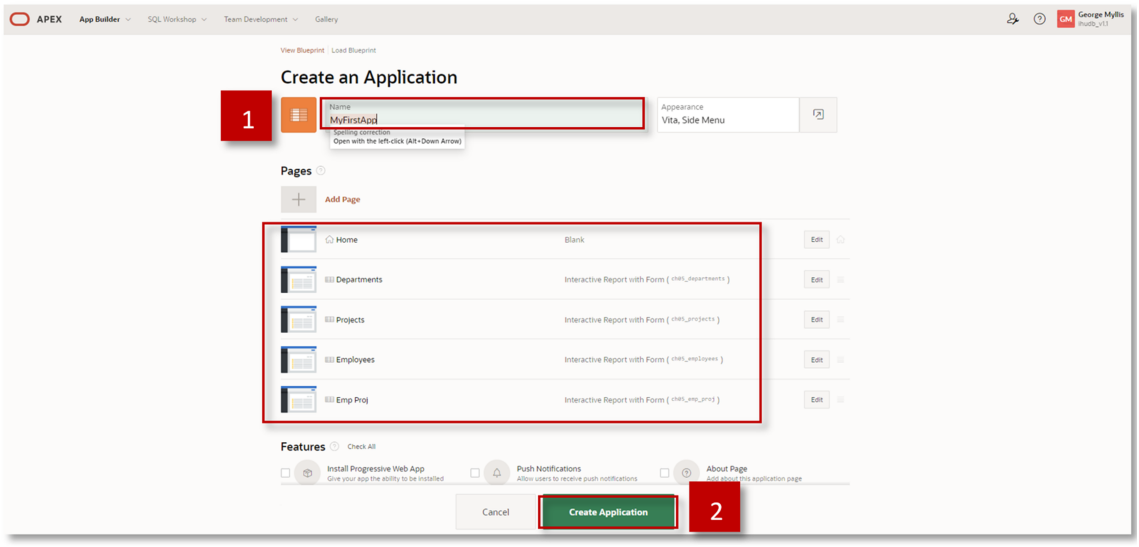
Dzięki sekcji opartej na danych użytkownicy mogą eksplorować dane pod wieloma kątami, dostosowywać widoki w oparciu o swoje preferencje i wykorzystywać moc danych w celu uzyskania znaczących spostrzeżeń. Służy jako potężne narzędzie do analizy danych, raportowania i wspomagania decyzji w ramach aplikacji, umożliwiając użytkownikom dokonywanie wyborów opartych na danych i osiąganie pomyślnych wyników.

Krok 1: Po pojawieniu się wyskakującego okna o nazwie **Create App from the Script**, zgodnie z rysunkiem 5.13, proszę sprawdzić, czy tabele zostały utworzone i będą używane w nowej aplikacji, a następnie kliknąć przycisk **Create Application**.



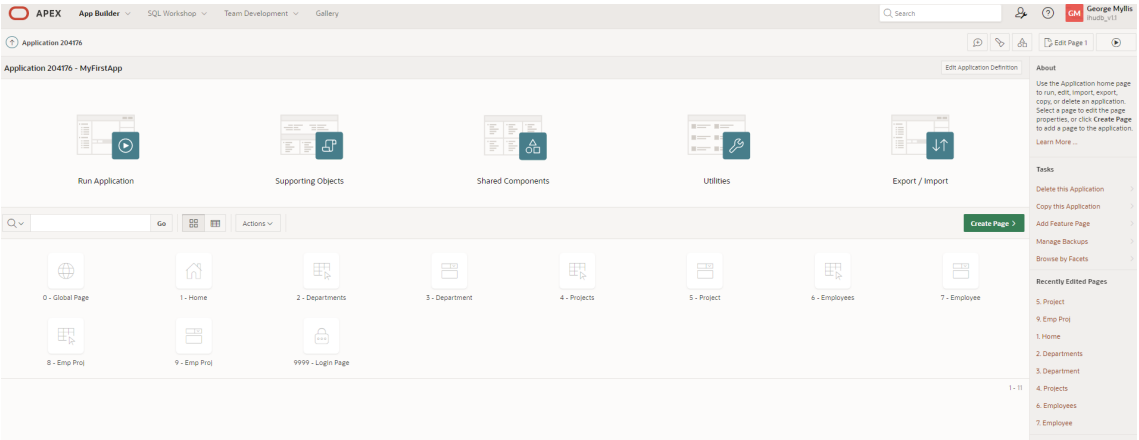
Rysunek 5.13: Tworzenie aplikacji ze skryptu.

Krok 2: Proszę wprowadzić nazwę aplikacji, sprawdzić, czy raporty i strony formularzy z tabel zostały dodane i będą używane w nowej aplikacji, a następnie kliknąć przycisk **Create Application**, jak pokazano na rysunku 5.14.



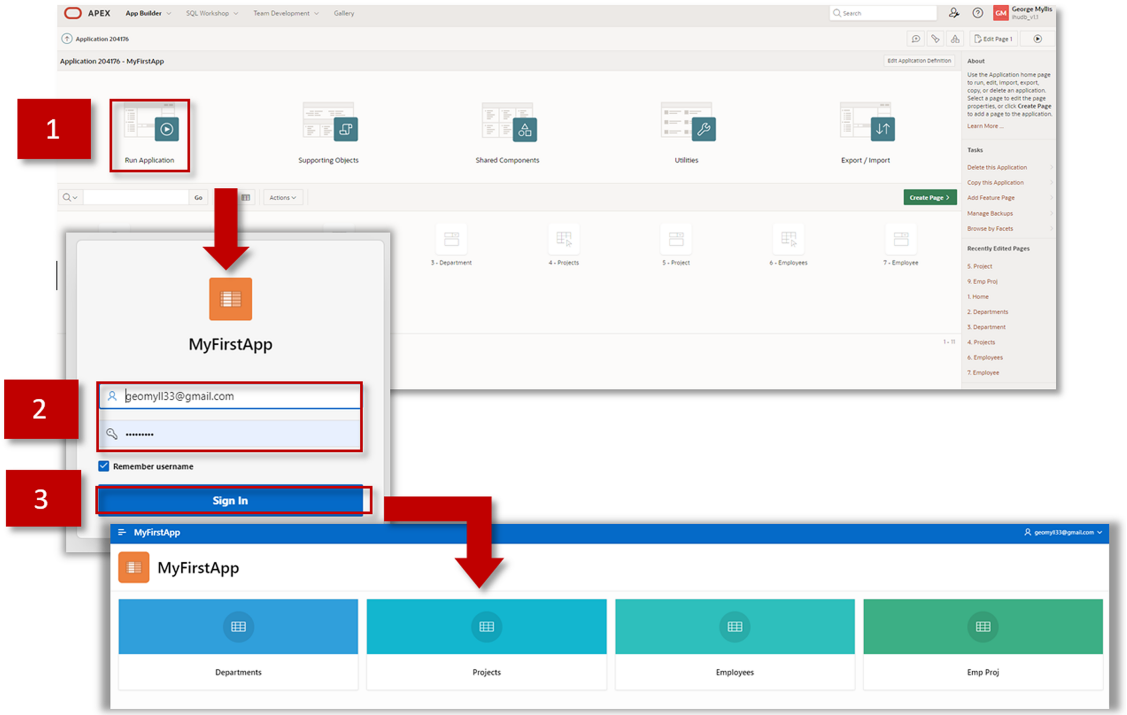
Rysunek 5.14: Utworzona aplikacja internetowa.

Krok 3: Dla Państwa aplikacji zostało utworzone środowisko App Builder, jak pokazano na rysunku 5.15. Ważne jest, aby pamiętać, że w tym środowisku mają Państwo możliwość szerokiego dostosowania aplikacji. Można dodawać, usuwać lub modyfikować pola formularzy, dostosowywać układ, definiować reguły walidacji i wykonywać wiele innych czynności w celu dostosowania aplikacji do konkretnych potrzeb.



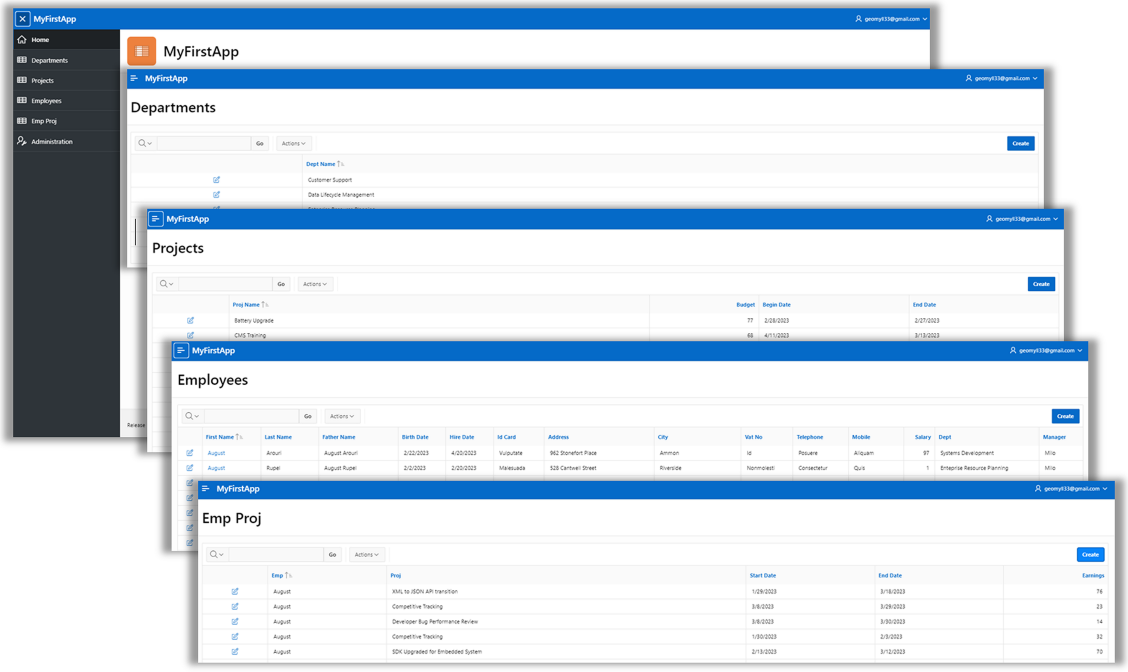
Rysunek 5.15: Środowisko tworzenia aplikacji.

Krok 4: Proszę kliknąć ikonę **Run Application** w środowisku App Builder, aby uruchomić aplikację, jak pokazano na rysunku 5.16. Spowoduje to otwarcie nowej strony przeglądarki z dedykowanym łączem z APEX dla Państwa aplikacji. Proszę wprowadzić swoje dane uwierzytelniające, które są takie same jak te w WORKSPACE, a następnie kliknąć **Sign in**. Zostaną Państwo przekierowani na stronę główną swojej aplikacji.



Rysunek 5.16: Strona logowania do aplikacji.

Krok 5: Można łatwo poruszać się po aplikacji za pomocą menu bocznego, jak pokazano na rysunku 5.17. Aplikacja zapewnia przyjazne dla użytkownika środowisko, które umożliwia płynne dodawanie, modyfikowanie i prezentowanie danych zgodnie z określonymi potrzebami i preferencjami.



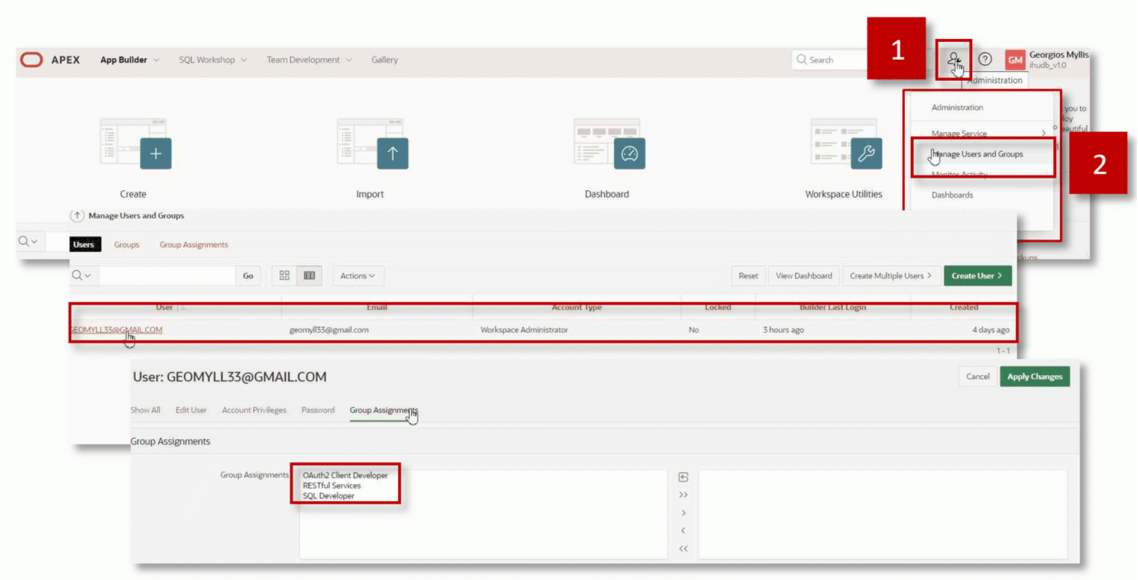
Rysunek 5.17: Nasze nowe środowisko aplikacji.

5.5 Administrowanie aplikacją

Jeśli chodzi o uwierzytelnianie kont aplikacji, jest to wbudowana metoda uwierzytelniania w Oracle Application Express, która umożliwia tworzenie i zarządzanie kontami użytkowników w repozytorium użytkowników Application Express. Metoda ta jest szczególnie przydatna, jeśli Państwa aplikacja korzysta z uwierzytelniania kont Application Express.

W środowisku Workspace aplikacji proszę kliknąć na **Administration** i wybrać **Manage Users and Groups**. Początkowo utworzone konto może być używane jako administrator dla wbudowanej aplikacji. Oracle APEX User utrzymuje konta dla schematów uwierzytelniania, które uwierzytelniają się na podstawie nazwy użytkownika i hasła przechowywanego na tych kontach.

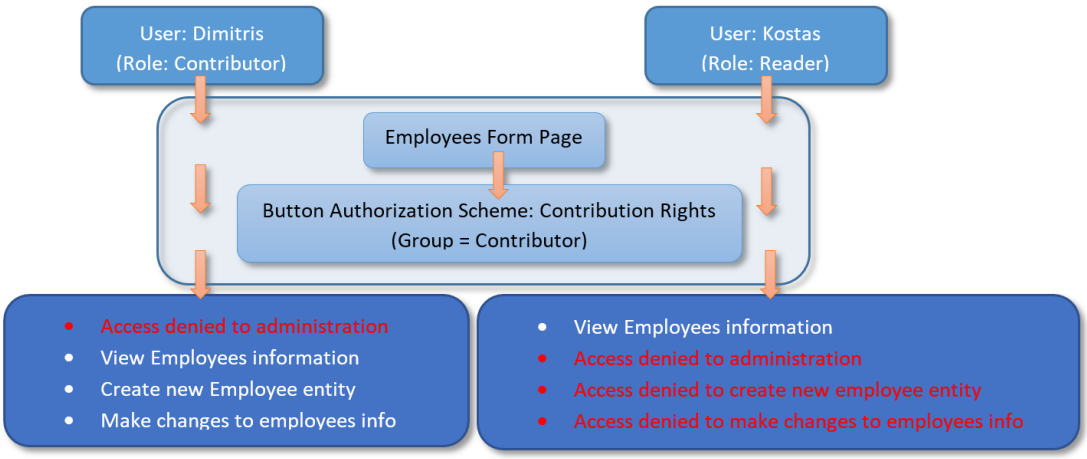
Administratorzy obszaru roboczego mają uprawnienia do tworzenia i modyfikowania aplikacji i obiektów bazy danych, a także do zarządzania kontami użytkowników, grupami i usługami programistycznymi. Programiści mogą tworzyć i modyfikować aplikacje i obiekty baz danych, podczas gdy użytkownicy końcowi nie mają uprawnień programistycznych i mogą uzyskać dostęp tylko do aplikacji, które nie korzystają z zewnętrznego schematu uwierzytelniania, jak pokazano na rysunku. 5.18.



Rysunek 5.18: Zarządzanie środowiskiem użytkowników i grup.

5.6 Kontrola dostępu

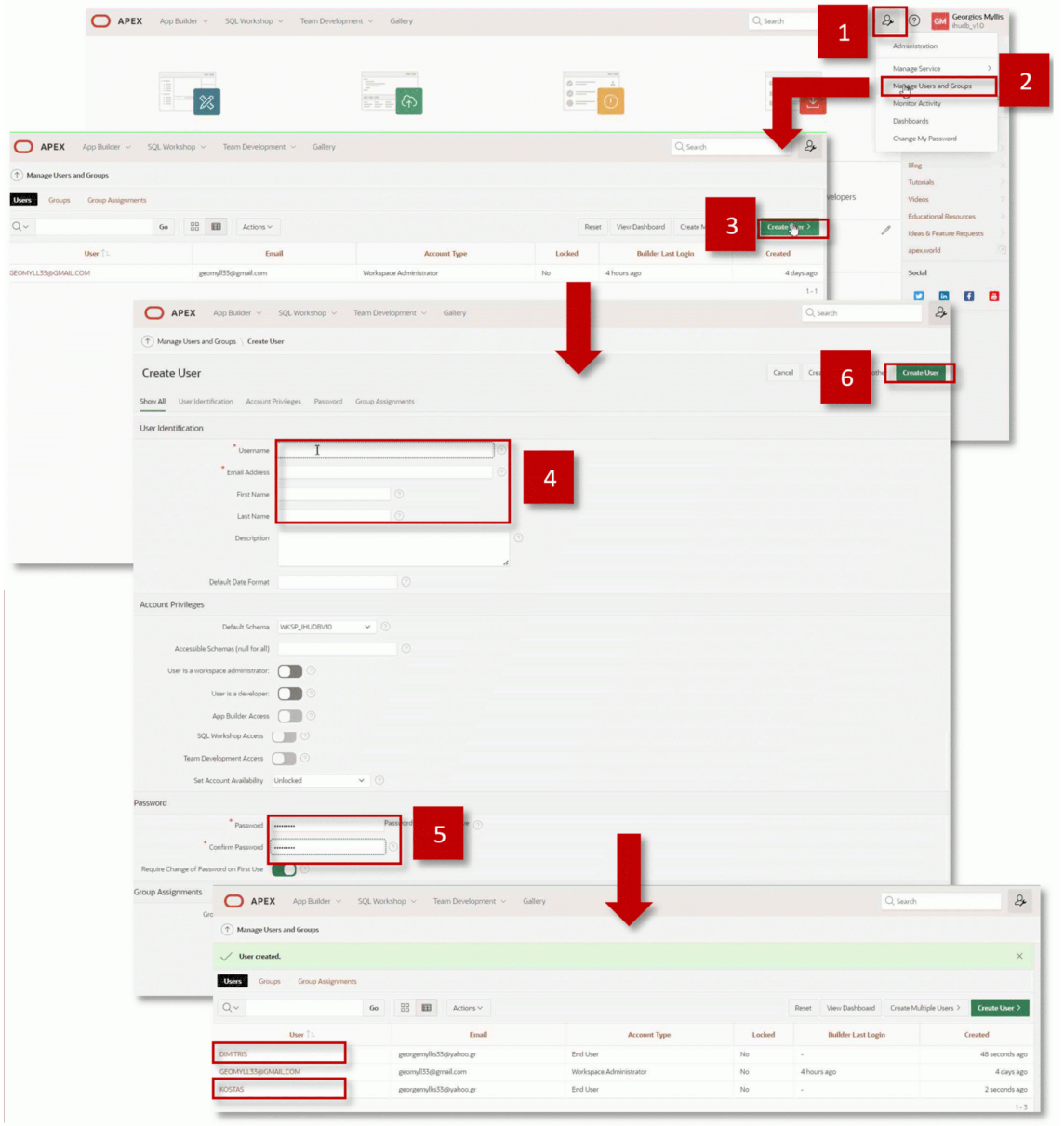
Następnie, jak pokazano na rysunku 5.19, przedstawimy przykład kontroli dostępu dla użytkowników korzystających z aplikacji, zgodnie z logiką przedstawioną na poniższym rysunku.



Rysunek 5.19: Przykład kontroli dostępu.

Oracle APEX zapewnia trzy wbudowane uprawnienia, które pozwalają na kontrolę dostępu do aplikacji lub jej komponentów. Uprawnienia te to administracja, edycja i widok, a każde z nich odpowiada roli dostępu. Administration odpowiada roli administratora, edit odpowiada roli Contributor, a View odpowiada roli Reader.

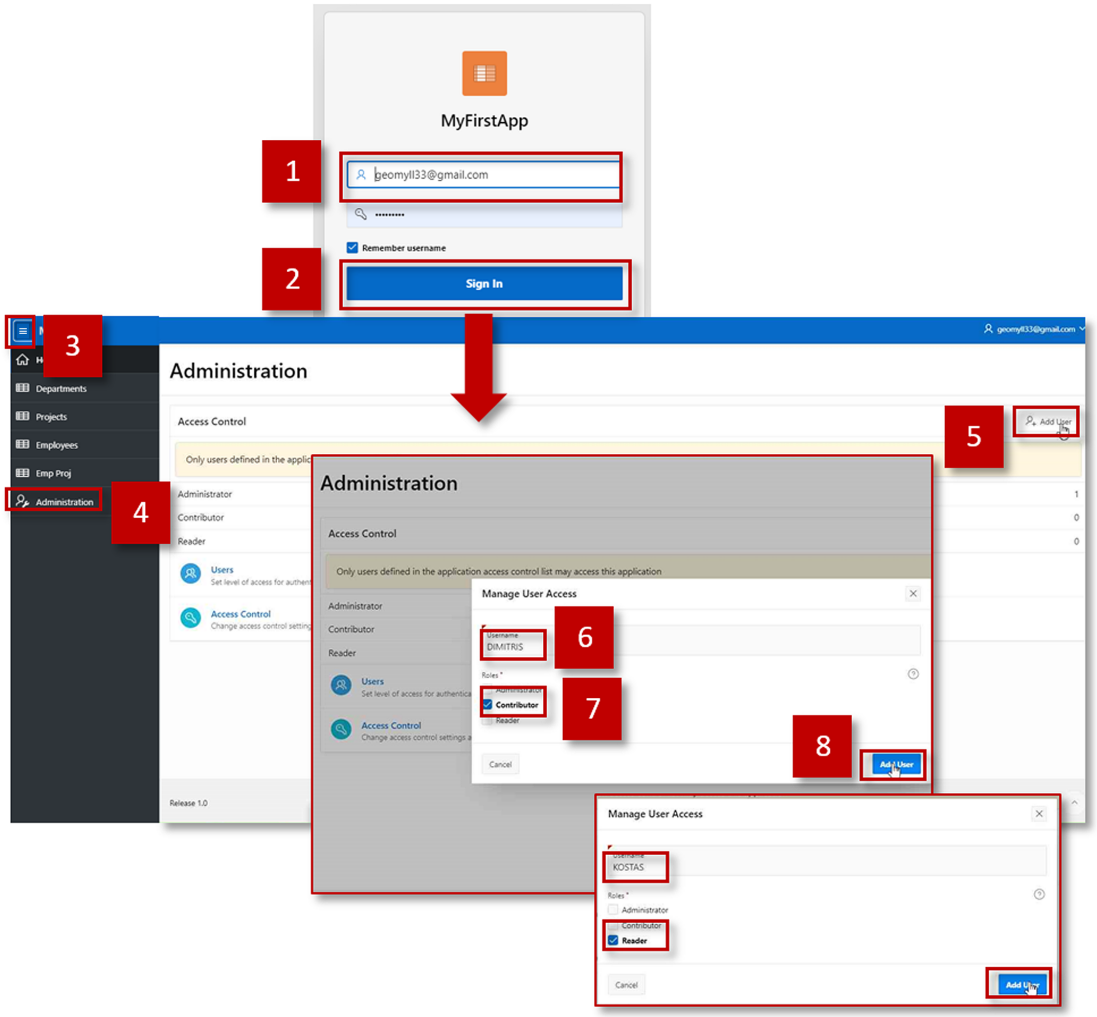
Aby dodać nowych użytkowników i zdefiniować ich role, proszę przejść do **Manage Users and Groups** ze środowiska wyświetlania APEX. Proszę kliknąć **Administration** i wybrać **Manage Users and Groups**. Następnie proszę wybrać **Create User**, wprowadzić identyfikator użytkownika i hasło, a następnie kliknąć **Create User**, jak pokazano na rysunku 5.20.



Rysunek 5.20: Dodawanie nowych użytkowników.

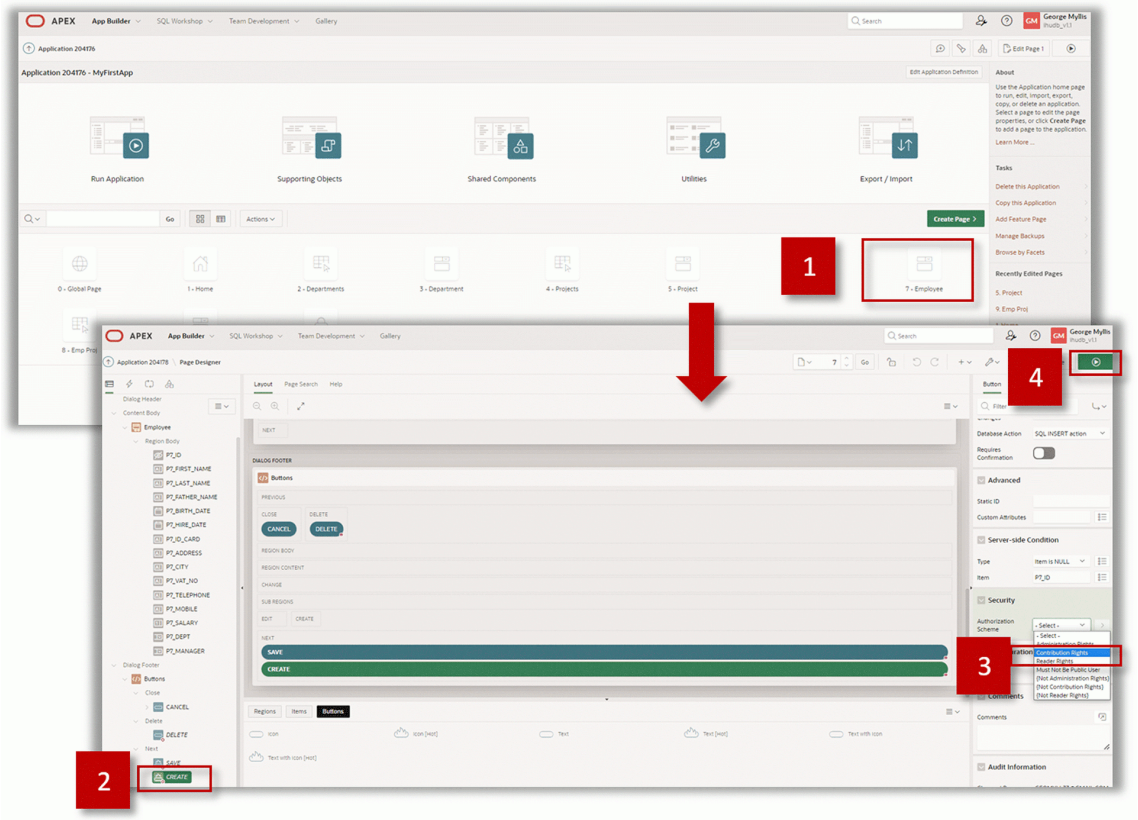
Następnie należy zdefiniować role dla użytkowników. W naszym przykładzie pokazujemy, jak można to zrobić w samej aplikacji

Krok 1: Proszę zalogować się do aplikacji za pomocą poświadczeń administratora, przejść do menu bocznego i kliknąć **Administration**, kliknąć **Add User**, wprowadzić identyfikator użytkownika i rolę, a następnie kliknąć **Add User** na Rysunku. 5.21.



Rysunek 5.21: Dodawanie ról do użytkowników.

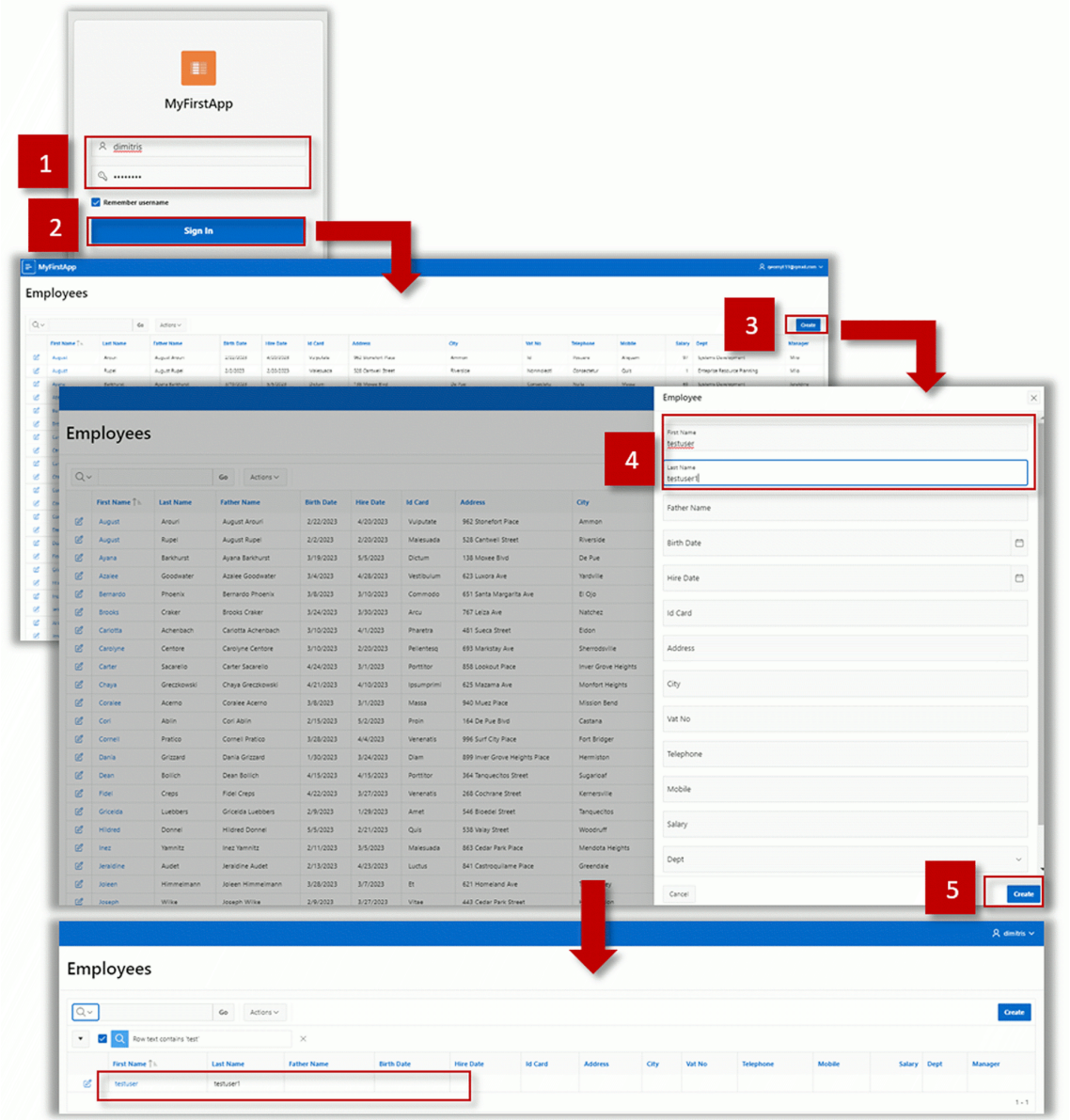
Krok 2: Proszę przejść do środowiska obszaru roboczego aplikacji i kliknąć **employees-Form**. W kreatorze stron formularza proszę wybrać przycisk **Create** i w opcji Security wybrać **Contribution Rights** jako motyw autoryzacji. Proszę kliknąć **Zapisz**. Teraz tylko użytkownicy z rolą wkładu będą mogli tworzyć i dodawać nowych pracowników, jak pokazano na rysunku 5.22.



Rysunek 5.22: Tworzenie konfiguracji przycisków.

Krok 3: Ten krok służy do weryfikacji konfiguracji kontroli dostępu dokonanych dla roli Contributor.

- Proszę zalogować się do aplikacji przy użyciu poświadczeń użytkownika Dimitris, który ma przypisaną rolę Contributor.
- Proszę otworzyć menu i sprawdzić, czy brakuje opcji administracyjnej, zgodnie z oczekiwaniami.
- Proszę spróbować utworzyć i dodać nowego pracownika jako współpracownika, jak pokazano na rysunku 5.23.
- Proszę sprawdzić, czy mogą Państwo przeglądać informacje o istniejących pracownikach i wprowadzać zmiany.
- Proszę sprawdzić, czy można utworzyć nowy podmiot pracowniczy.

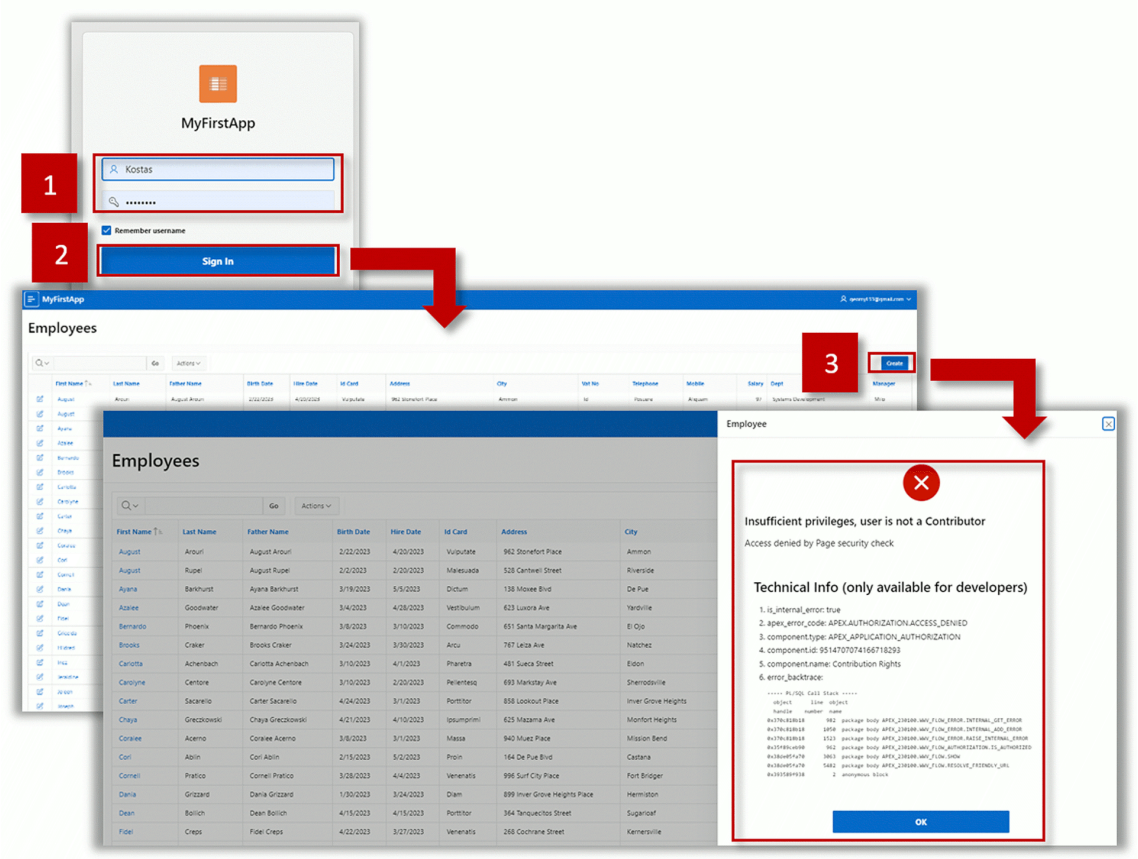


Rysunek 5.23: Dostęp i prawa współpracowników.

Krok 4: Ten krok służy do weryfikacji konfiguracji kontroli dostępu dokonanych dla roli Reader.

- Proszę zalogować się do aplikacji przy użyciu poświadczeń użytkownika Kostas, który ma przypisaną rolę Reader.
- Proszę otworzyć menu boczne i sprawdzić, czy brakuje opcji administracyjnej, zgodnie z oczekiwaniami.
- Proszę kliknąć na formularz Pracownicy w środowisku pracy aplikacji.
- Proszę spróbować utworzyć lub dodać nowego pracownika. Otrzymają Państwo komunikat ódmowa dostępu", jak pokazano na rysunku 5.24.
- Proszę sprawdzić, czy mogą Państwo przeglądać informacje o istniejących pracownikach, ale nie mogą Państwo wprowadzać żadnych zmian.

Kroki te potwierdzają, że konfiguracje kontroli dostępu zostały pomyślnie zastosowane do użytkowników.



Rysunek 5.24: Dostęp i prawa czytelników.

5.7 Dodatkowe materiały edukacyjne

Mogą Państwo znaleźć następujące dodatkowe materiały edukacyjne:

- skrypty
- aplikacje
- video.

Wszystkie dodatkowe materiały edukacyjne są dostępne na stronie publiczna strona projektu BeeAPEX. Proszę zalogować się jako gość (hasło nie jest wymagane). Podręcznik można znaleźć w sekcji Książki, skrypty w folderze Część 2 > Rozdział 5 w sekcji Skrypty, a przewodniki wideo w sekcji Kolekcja przewodników wideo. Materiały do krótkich kursów znajdują się w sekcji Krótkie kursy.

5.7.1 Wyeksportowane aplikacje

Wyeksportowana aplikacja jest spakowana. Instalacja tworzy tabele, indeks, funkcję, procedurę i wyzwalacz, a także wypełnia dane. Deinstalacja usuwa wszystkie obiekty bazy danych używane w tej aplikacji.

5.7.2 Przewodniki wideo

Przewodnik wideo pokazuje każdy krok w tworzeniu aplikacji.

5.8 Pytania

1. Na ile sposobów można utworzyć nową aplikację w ORACLE APEX Application Builder?
2. Jak można dodać użytkowników do aplikacji?

3. Ile jest wbudowanych uprawnień użytkownika w ORACLE APEX do kontroli dostępu do aplikacji lub jej komponentów?

5.9 Pytania

1. Istnieją 3 sposoby
   - Nowa aplikacja
   - Z pliku
   - Aplikacja startowa
2. Proszę przejść do środowiska obszaru roboczego aplikacji i wybrać Zarządzaj użytkownikami i grupami.
3. Istnieją trzy
   - Administrator
   - Współtwórca
   - Czytelnik



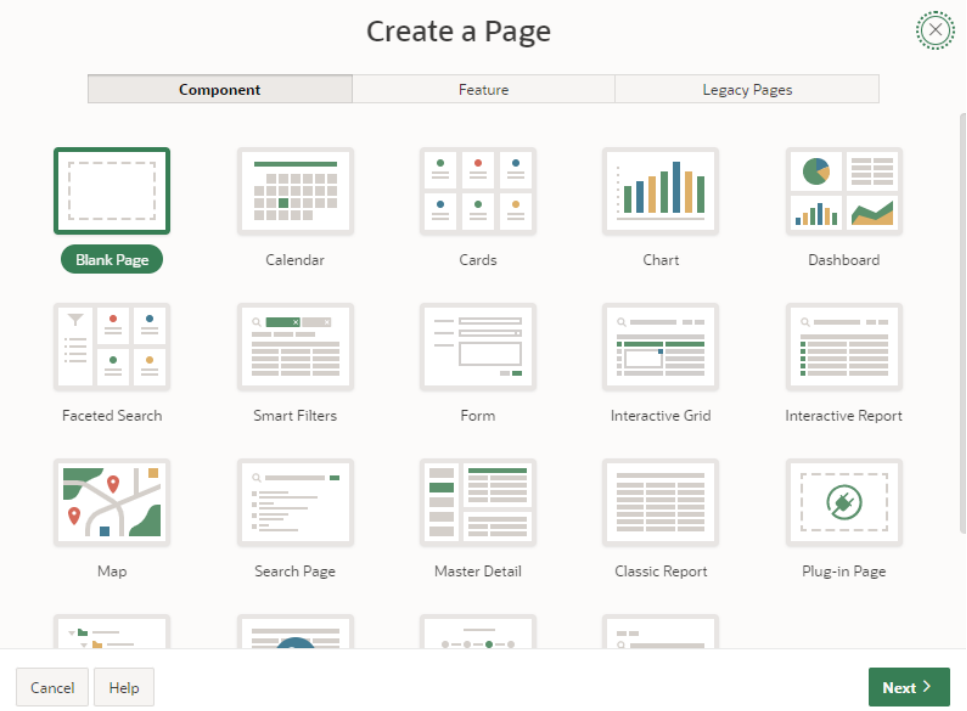
6. Jak zarządzać raportami?

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

Dane w bazach danych są często bez znaczenia, jeśli nie otrzymamy z nich danych wyjściowych, które powiedzą nam więcej o prawdziwych informacjach, które zawierają. Aby przedstawić wyniki, obserwować i dowiedzieć się, używamy po prostu raportów, którymi zajmiemy się w następnym rozdziale. Dowiemy się, jak łatwo utworzyć taki raport w środowisku APEX i jak go edytować. Zajmiemy się różnymi typami raportów, takimi jak raporty interaktywne, raporty klasyczne, raporty z przełączaniem kolumn lub raporty z wizualizacjami wykresów.

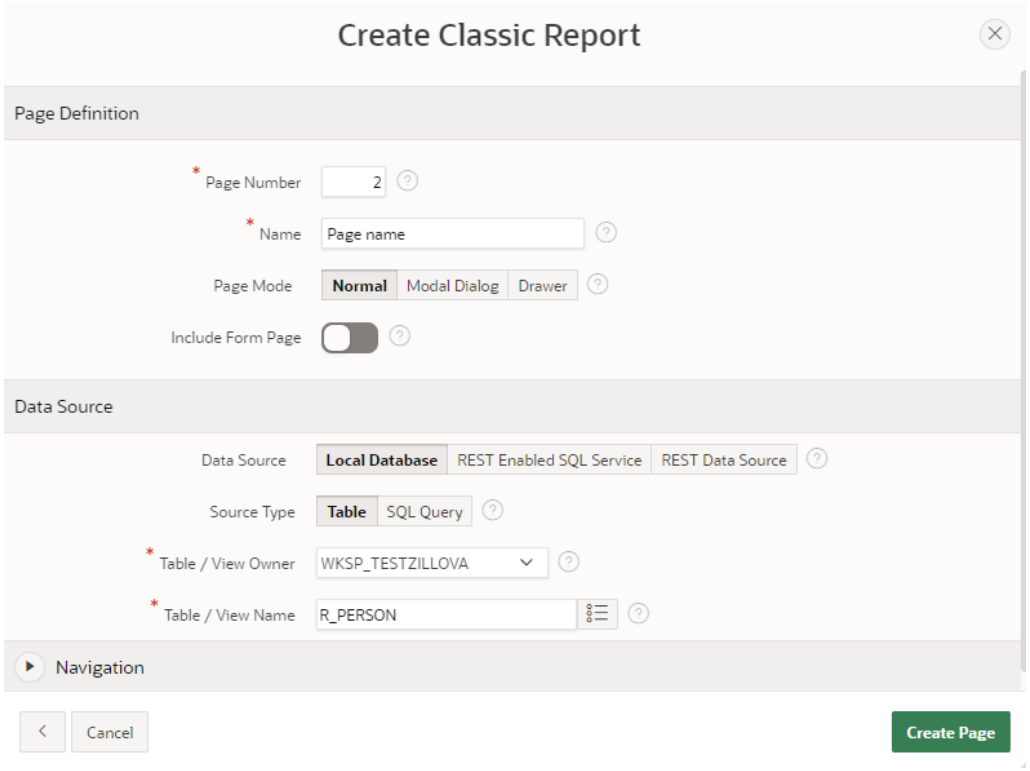
6.1 Raport

Raport to *region* który upraszcza przeglądanie danych i wyświetla dane w wierszach. Może występować jako jedna jednostka lub może być używany do dzielenia stron na logiczne jednostki. Może być statyczny, z samym tytułem i formatowaniem, ale bez żadnych dodatkowych funkcji. Może też być dynamicznym regionem, posiadającym określoną funkcjonalność, taką jak *kalendarz, raport* lub *lista*. Regiony te są wstępnie zaprogramowane z podstawowymi funkcjami, które mogą być również łatwo edytowane przez użytkowników.



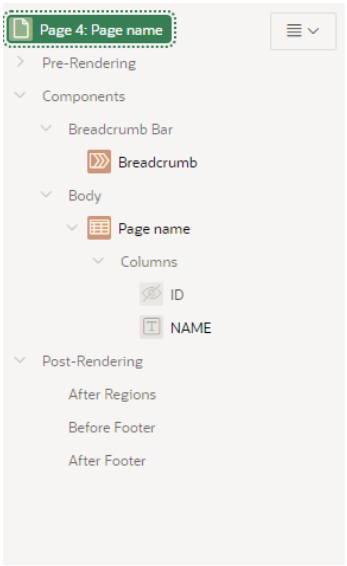
Rysunek 6.1: Wybór typu strony.

Jak widać na rysunku 6.1, podczas tworzenia strony **page** użytkownicy mogą wybierać spośród gotowych typów stron, które już zawierają raporty, więc nie jest konieczne samodzielne wstawianie regionu. Przykład formularza do tworzenia raportu pokazano na rysunku 6.2. Podczas tworzenia strony zawierającej raport należy podać nazwę strony i źródło danych strony. Dane mogą być ładowane ze źródła *local database, REST enabled SQL* lub *REST*. Wybierając opcję lokalnej bazy danych, należy wybrać, co dokładnie będzie używane jako źródło raportów - użytkownik może korzystać nie tylko z samej bazy danych *table*, ale także z istniejących *views* lub własnych instrukcji *SQL query*.



Rysunek 6.2: Tworzenie strony z raportem.

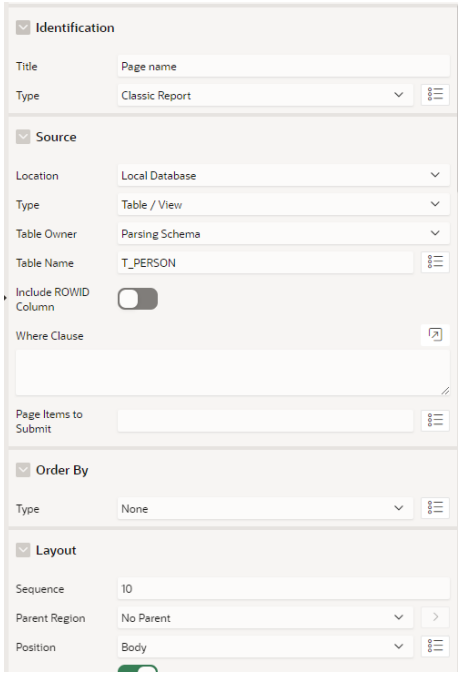
Po utworzeniu strony po lewej stronie pojawi się sekcja podobna do tej na rysunku 6.3. W bocznym pasku nawigacyjnym można zobaczyć stronę, jej nazwę i wszystkie jej komponenty. Po kliknięciu na jeden z komponentów zostanie on podświetlony. Następnie można zobaczyć kolumny, które zawiera, a także jego podregiony lub elementy. Za pomocą prawego przycisku myszy można utworzyć nowy pusty region.



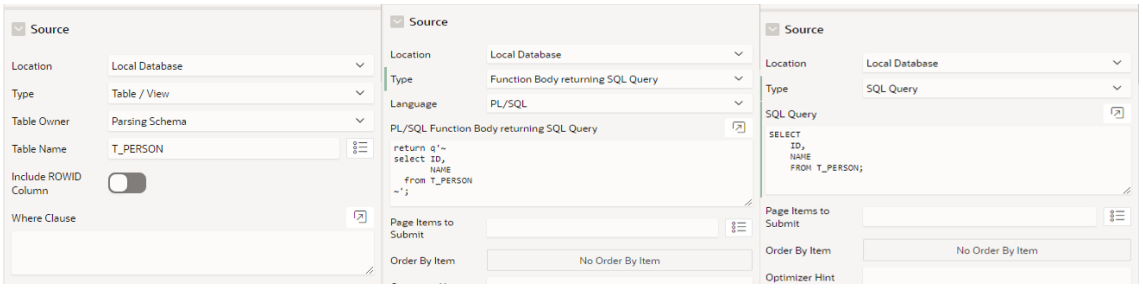
Rysunek 6.3: Renderowanie.

Na rysunku 6.4 można zobaczyć region i szczegóły jego atrybutu. **Identification** określa tytuł strony i typ raportu (np. raport klasyczny). Rysunek 6.5 przedstawia kilka przykładów sekcji

źródłowej, w której definiowane jest źródło danych dla naszego raportu, a także informacje o pochodzeniu danych (jak wspomniano powyżej: *tabela, widok, funkcja* lub *zapytanie SQL*). **Where Clause** mówi o tym, jaki warunek zostanie zastosowany do filtrowania danych.



Rysunek 6.4: Właściwości regionu.



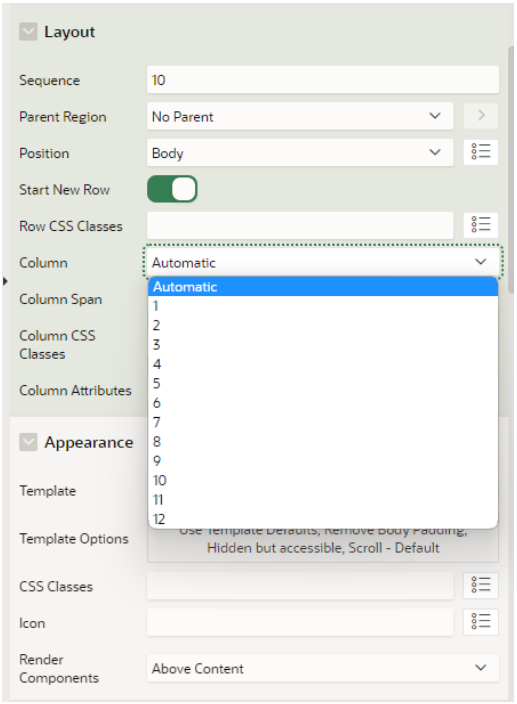
Rysunek 6.5: Różne źródła danych.

Na rysunku 6.6 można zobaczyć **Layout**, gdzie można określić lokalizację regionu na stronie. **Region nadrzędny** służy do określenia, który region będzie zawierał nasz raport. Można to również dostosować za pomocą **Position**. Pozycje są definiowane przez określony *szablon strony*. Po utworzeniu strony, jest ona budowana z określonym motywem i układem w zależności od wybranego szablonu. Domyślny szablon dla motywu **Universal** nazywa się **Standard**.

Regiony są zorganizowane w *grid* - tak więc każdy region ma swoją własną lokalizację zdefiniowaną przez wiersz i kolumnę. Regiony te są uporządkowane według ich **numeru sekwencji**. Pierwszy region znajduje się w pierwszym wierszu i pierwszej kolumnie siatki. Każdy następny region jest następnie zorganizowany według jednego z tych wzorców: w tym samym wierszu i kolumnie co poprzedni region i jest wyświetlany pod poprzednim regionem. Atrybuty **Start New Row** i **New Column** służą do określania sposobu organizacji tych regionów.

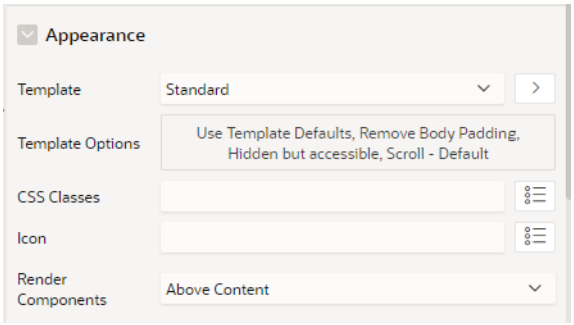
Atrybuty *Size* i *location* są obliczane po przypisaniu wiersza i kolumny. Odbywa się to poprzez obliczenie zajętości i szerokości dla każdej części wiersza za pomocą wartości parametrów **Column** i **Column Span**.

Wcięcie obszaru w wierszu jest definiowane przez atrybut **Column**. Wartość tej kolumny mieści się w zakresie **1 - 12**, który odnosi się do **1** z **12**. **points** siatki, która dzieli szerokość. Wartość **1** oznacza najbardziej *lewa część*, a **7** oznacza *środek* strony.

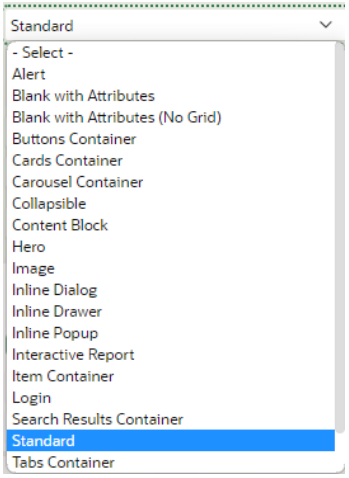


Rysunek 6.6: Układ.

Na rysunku 6.7 można zobaczyć sekcję **Appearance**, która służy do edycji szablonu dla określonego regionu. Rysunek 6.8 przedstawia wstępnie zdefiniowane szablony, które użytkownik może wybrać. Każdy szablon można dostosować, klikając opcje szablonu lub pisząc własne instrukcje **CSS**. W opcjach szablonu można zdefiniować wiele reguł, takich jak *wysokość treści*, *widoczność nagłówka* lub nawet *odległość między elementami*.



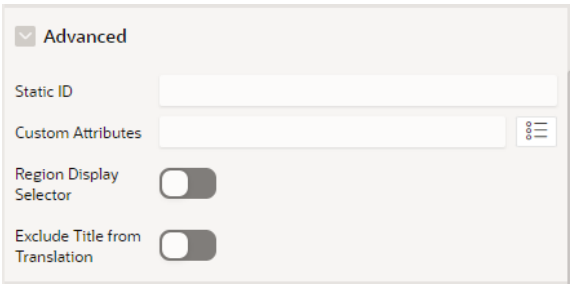
Rysunek 6.7: Wygląd.



Rysunek 6.8: Szablony.

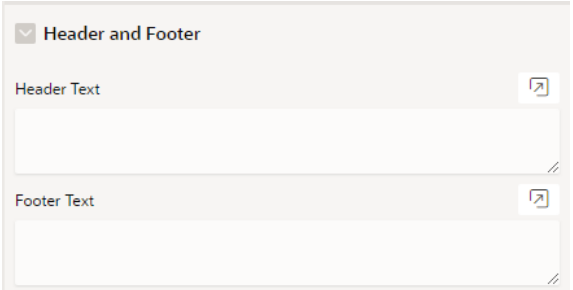
Rysunek 6.9 przedstawia sekcję o nazwie **Advanced**, w której można wybrać jedną z czterech opcji. **Static ID** służy do ustawienia identyfikatora raportu. Ten **Static ID** jest używany jako **ID** dla elementu **HTML** na stronie w sytuacjach, gdy użytkownik chce skonfigurować wygląd strony za pomocą **CSS**. **Custom Attributes** są dla dodatkowych informacji dla elementu **HTML**. Atrybuty niestandardowe są przechowywane jako nazwa/wartość, na przykład: *name* = ”*MY_REPORT*”

Region Display Selector to komponent regionu, który zapewnia elementy sterujące nawigacją dla innych regionów na stronie. **Exclude Title from Translation** pozwala uniknąć tłumaczenia tytułu w przypadku, gdy strona ma zdefiniowane tłumaczenia. .



Rysunek 6.9: Zaawansowane.

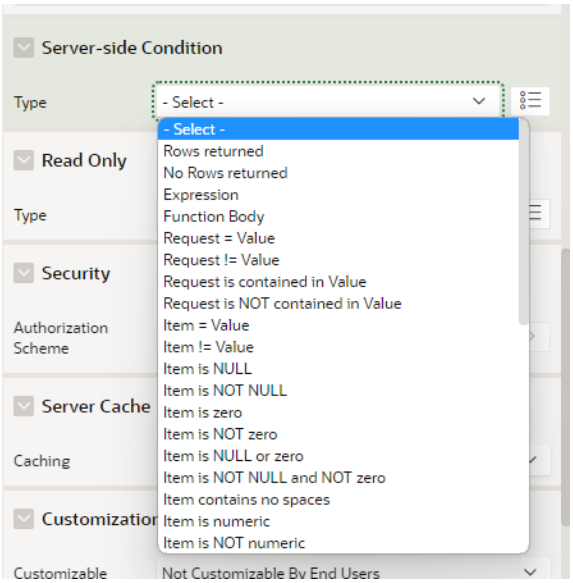
Nagłówek i stopka pokazane na rysunku 6.10 służą do edycji zawartości tych części raportu. Należy wspomnieć, że stopka i nagłówek są powiązane z raportem, a nie z całą stroną.



Rysunek 6.10: Nagłówek i stopka.

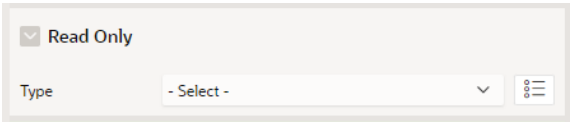
Rodzaje **Server-side Conditions** są wymienione na rysunku 6.11. W aplikacji opracowanej

przez Oracle APEX, **Server-side Conditions** są rozwiązywane w fazie inicjalizacji ładowania strony internetowej. Raport nie jest wyświetlany, jeśli warunki po stronie serwera nie są spełnione. Istnieje wiele rodzajów warunków. Przykładem może być porównywanie wartości elementów lub warunki sprawdzania żądań.



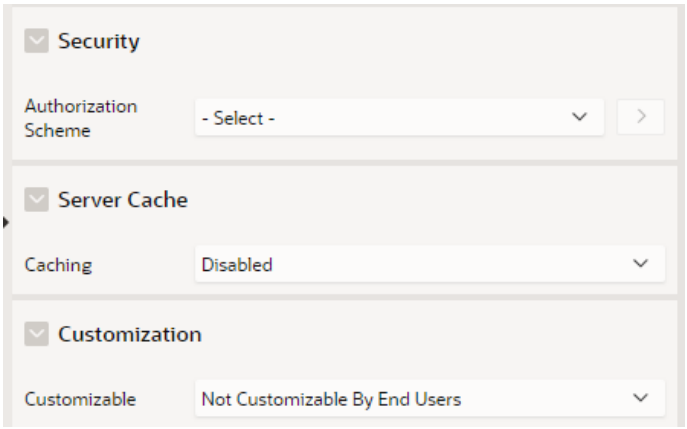
Rysunek 6.11: Stan po stronie serwera.

Read Only przedstawiony na rysunku 6.12 ma takie same opcje jak element **Server-side Condition** przedstawiony na rysunku 6.11. Oczywiście nie ma możliwości zmiany widoku wybranego regionu, ponieważ jest on tylko do odczytu. W przypadku raportu podstawowego nie ma możliwości sortowania wartości kolumn.



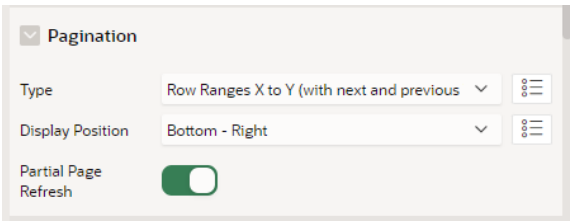
Rysunek 6.12: Tylko do odczytu.

Na rysunku 6.13 można zobaczyć trzy ostatnie sekcje, które zostaną wyjaśnione. Są to **Security, Server Cache** i **Customization**. Sekcja **Security** zawiera opcje ograniczeń w zakresie wyświetlania strony internetowej - np. *strona publiczna, tylko dla autoryzowanych użytkowników*. Sekcja **Server Cache** zawiera ustawienia dotyczące zarządzania pamięcią podręczną. W oknie **Customization** można dostosować dostęp do modyfikacji raportu.



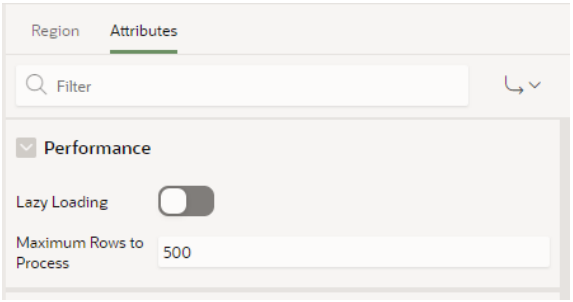
Rysunek 6.13: Bezpieczeństwo, pamięć podręczna serwera i personalizacja.

Kolejną grupą sekcji jest **Attributes**. Można tu ustawić takie elementy jak liczba wyświetlanych wierszy, czy możliwe jest wyświetlanie wyników raportu oraz w jaki sposób będą one wyświetlane w całym raporcie.

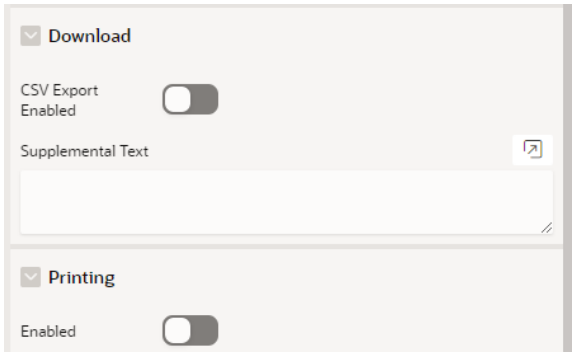


Rysunek 6.14: Paginacja.

Inną interesującą funkcją jest ustawienie maksymalnej liczby załadowanych wierszy. Można również zdefiniować, co ma być wyświetlane w przypadku braku danych. Istnieje możliwość pobrania raportów w postaci pliku **.csv** lub ich wydrukowania. Wszystko, o czym mowa w tym akapicie, można znaleźć w **Attributes**.

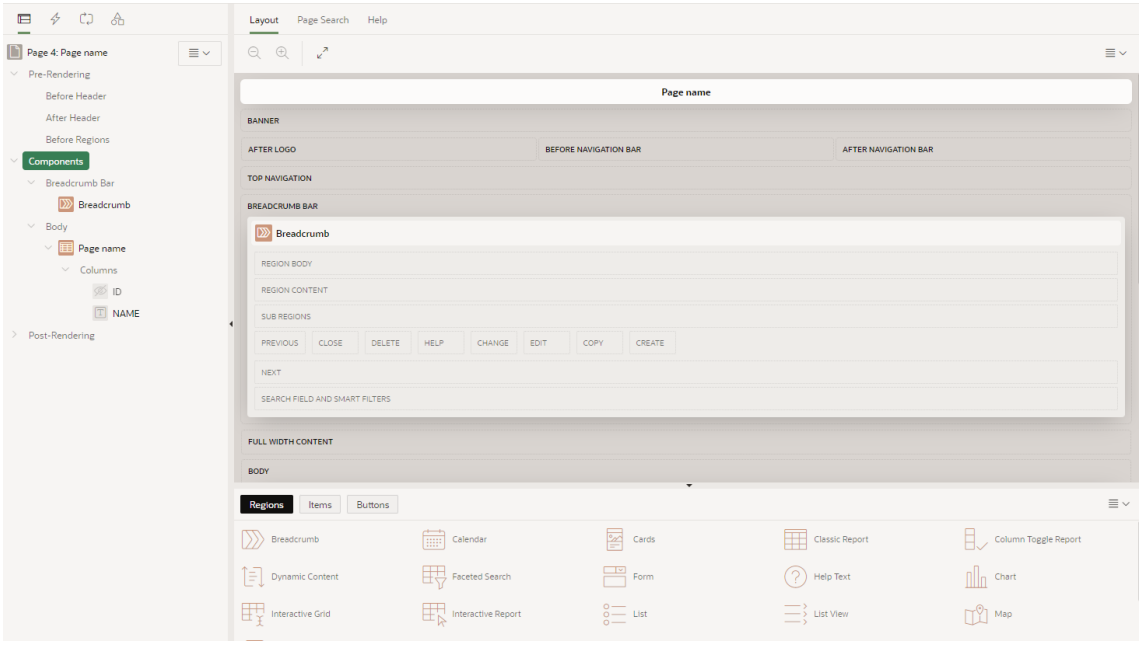


Rysunek 6.15: Liczba wierszy do załadowania.



Rysunek 6.16: Pobieranie i drukowanie.

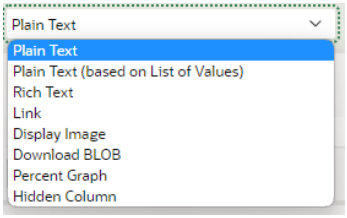
Bardzo ważną częścią jest *center* aplikacji. Na rysunku 6.17 można zobaczyć **Layout**, który pokazuje nam, jak wszystkie komponenty będą zorganizowane na naszej stronie. Cała strona ma strukturę siatki, więc bardziej intuicyjne jest wyobrażenie sobie, gdzie zostaną umieszczone komponenty. Aby być jak najbardziej przyjaznym dla użytkownika, można wybrać region/element i przeciągnąć go w wybrane miejsce. Dla każdego regionu istnieje domyślny szablon, który jest dostarczany po jego wprowadzeniu. Wszystkie niezbędne ustawienia są już skonfigurowane.



Rysunek 6.17: Środek projektanta stron.

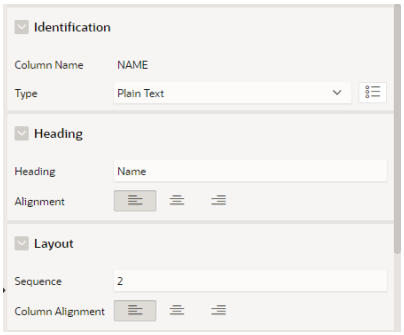
Jak widać po lewej stronie na rysunku 6.17, APEX domyślnie tworzy kilka komponentów dla raportu. Każdy komponent reprezentuje jedną kolumnę i można go zmienić w taki sam sposób, jak w przypadku raportu.

Ustawienia kolumny pokazano na rysunku 6.19. W polu **Identification** widoczne są dwie cechy - **Type** i **Name of Column**. Nie można zmienić **Name of Column**, ponieważ jego wartość jest uzyskiwana z *SQL query*. **Type of Column** pokazany na rysunku 6.18 określa sposób wyświetlania kolumny - proszę wskazać niektóre z nich: *Hidden* (nic nie jest wyświetlane), *Plain Text*, *Display Image, Percent Graph* itd. Domyślnie wszystkie kolumny są oznaczone jako *Tekst zwykły*.



Rysunek 6.18: Typ kolumny.

Heading zawiera dwie opcje dotyczące nagłówka. W przeciwieństwie do nazwy kolumny, **Heading** (nazwa) może być edytowana. Nagłówek może również zawierać kod **HTML**. Opcja **Alignment** mówi sama za siebie. Sekcja **Layout** oferuje opcję **Sequence**, która określa, w jakiej kolejności będą wyświetlane kolumny, a także opcję **Alignment**, taką samą jak w poprzedniej sekcji



Rysunek 6.19: Ustawienia kolumny.

Ostatnią istotną częścią jest **Wygląd**, który należy edytować, gdy chcą Państwo traktować liczbę jako ciąg znaków w jakimś *specjalny format*. Jak widać na rysunku 6.20, można wybierać spośród wielu podstawowych formatów po kliknięciu na **Maska formatu**. Te formaty są oferowane przez sam APEX, ale można wyrazić swój własny format, jeśli chcesz.

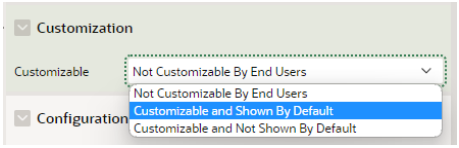


Rysunek 6.20: Format maski.

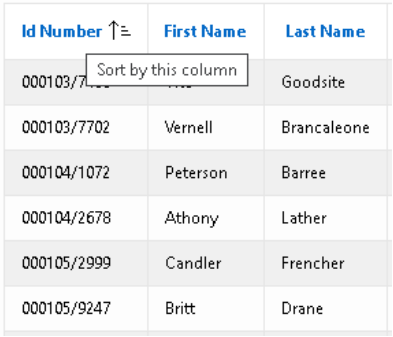
6.2 Raport klasyczny

A **Classic Report** jest typem raportu, który pozwala tylko na podstawowe wyświetlanie danych, bez możliwości modyfikacji opcji wyświetlania przez użytkownika. Jedyną rzeczą dostępną dla

użytkownika jest sortowanie danych według poszczególnych kolumn, co również należy ustawić.

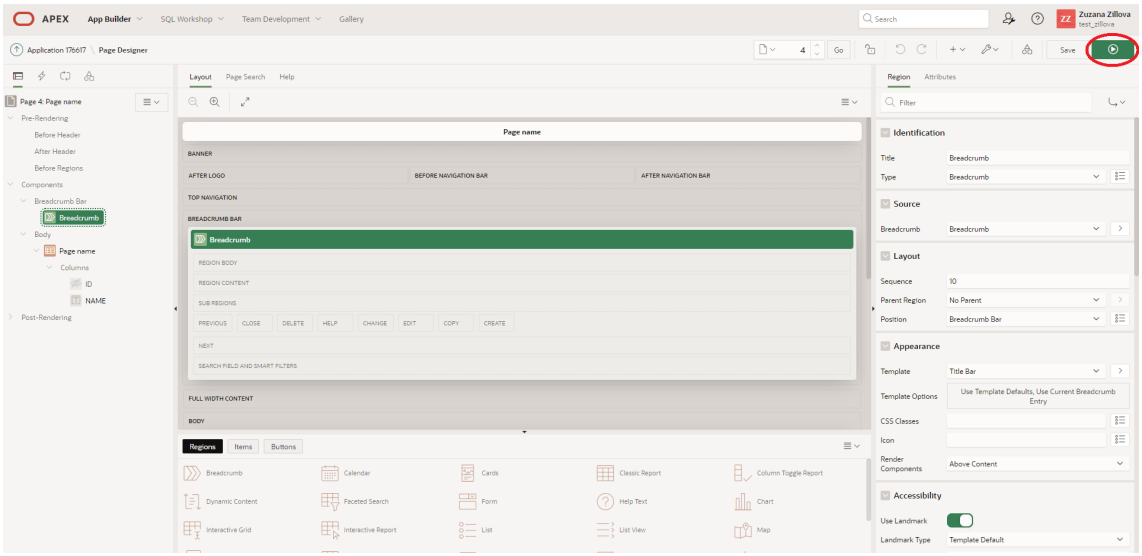


Rysunek 6.21: Personalizacja.

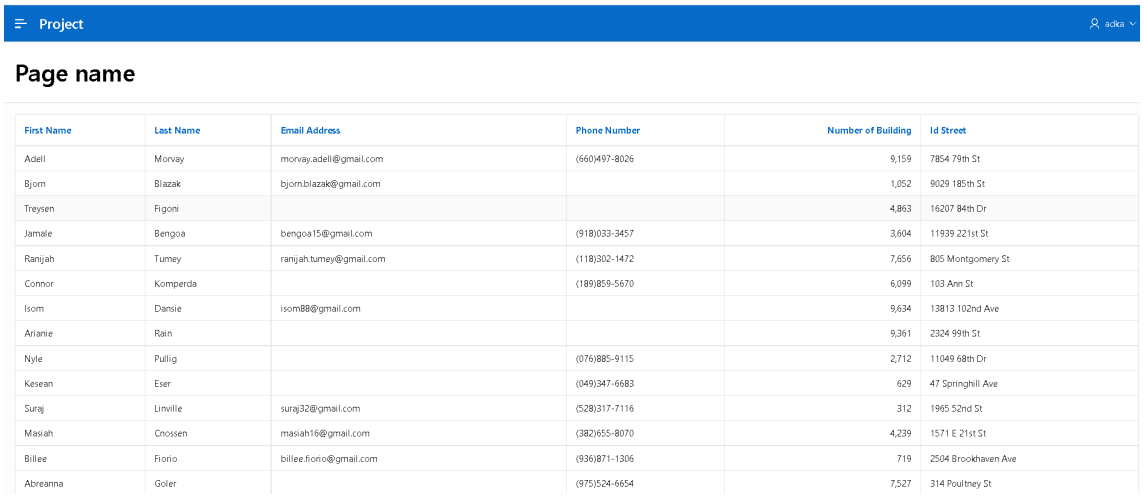


Rysunek 6.22: Opcja sortowania według kolumny.

Jeśli chcą Państwo zobaczyć, jak będzie wyglądać strona, można kliknąć zieloną strzałkę w okręgu. Jak widać na rysunku 6.23, strzałka znajduje się w prawym górnym rogu. Na rysunku 6.24 możemy zobaczyć, jak wygląda klasyczny raport. Ma on kolumny z nazwami i danymi. Żadna interakcja, z wyjątkiem sortowania według kolumn, nie jest dozwolona dla użytkownika.



Rysunek 6.23: Lokalizacja zielonej strzałki dla prezentacji strony.

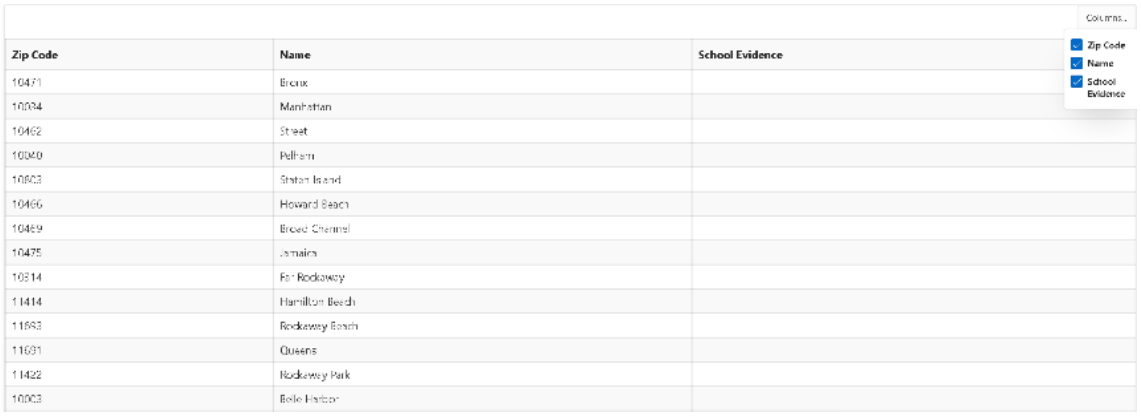


First Name	Last Name	Email Address	Phone Number	Number of Building	Id Street
Adell	Morvay	morvay.adell@gmail.com	(660)497-8026	9,159	7854 79th St
Bjorn	Blazak	bjorn.blazak@gmail.com		1,052	9029 185th St
Treysen	Figoni			4,863	16207 84th Dr
Jamale	Bengoa	bengoa15@gmail.com	(918)033-3457	3,604	11939 221st 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

Rysunek 6.24: Klasyczny przykład raportu.

6.3 Raport przełączania kolumn

Column Toggle Report jest prawie taki sam jak klasyczny raport, tylko jedną dodatkową funkcją jest określenie, która kolumna powinna być widoczna, a która nie. Możemy to zobaczyć w prawym górnym rogu *Rysunek 6.25*.

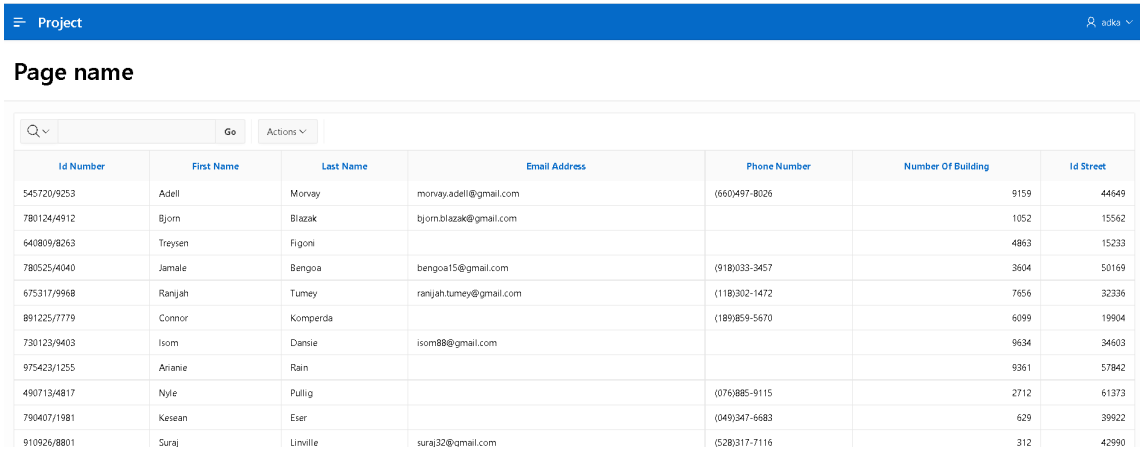


Zip Code	Name	School Evidence
10471	Bronx	
10004	Manhattan	
10462	Street	
10040	Pelham	
10603	Staten Island	
10466	Howard Beach	
10469	Broad Channel	
10475	Jamaica	
10314	Far Rockaway	
11414	Hamilton Beach	
11693	Rockaway Beach	
11691	Queens	
11422	Rockaway Park	
10003	Belle Harbor	

Rysunek 6.25: Możliwość zmiany widoczności kolumny w Column Toggle Report.

6.4 Raport interaktywny

Raport interaktywny przedstawiony na rysunku 6.26 pozwala użytkownikom na dokonywanie pewnych dostosowań raportu podczas przeglądania strony bez dostępu do rdzenia aplikacji. Raport interaktywny jest tworzony tak samo jak każdy inny i nie różni się zbytnio wyglądem. Teraz przyjrzymy się bliżej poszczególnym funkcjom i dostosowaniom, które użytkownik może dostosować w raporcie interaktywnym.

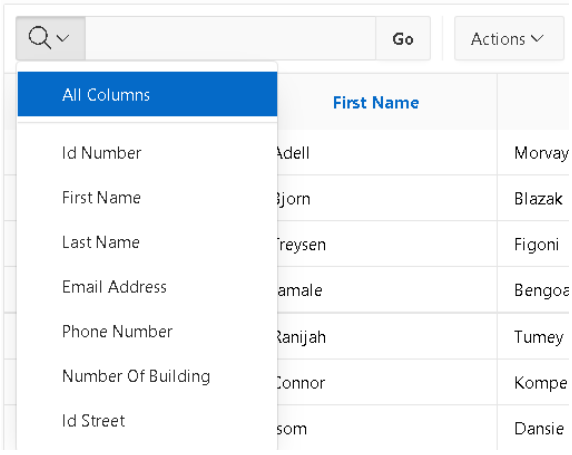


Rysunek 6.26: Raport interaktywny.

Panel wyszukiwania pokazany na rysunku 6.27 składa się z czterech elementów:

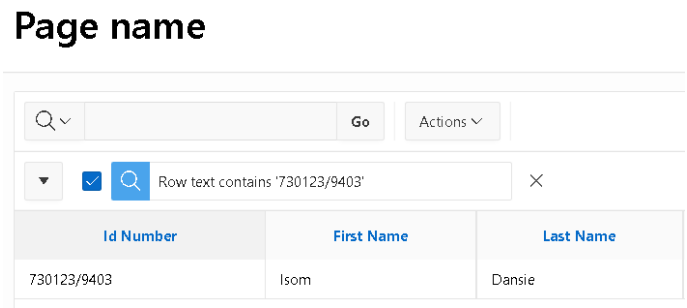
- **lupa,**
- **pole tekstowe,**
- **przycisk idź,**
- **przycisk akcji.**.

Pierwsze trzy elementy pozwalają użytkownikowi filtrować wiersze raportu. Wprowadzamy wyszukiwany ciąg w polu, klikamy na **lupa**, gdzie wybieramy, które kolumny mają być wyświetlane, a następnie klikamy na **Przycisk idź**. Zobaczymy te rekordy, które zawierają wyszukiwany ciąg.



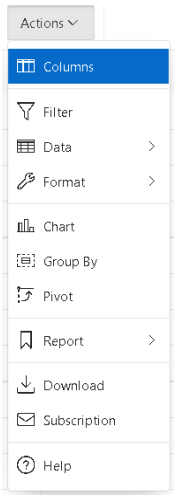
Rysunek 6.27: Panel wyszukiwania.

Na przykład, chcemy wyszukać rekordy, które zawierają liczbę **"730123/9403"** w dowolnej kolumnie. Ten filtr zostanie wyświetlony i można go wyłączyć, klikając przycisk **X**. Sytuację pokazano na rysunku 6.28. Możemy mieć wiele filtrów aktywnych w tym samym czasie w różnych kolumnach.



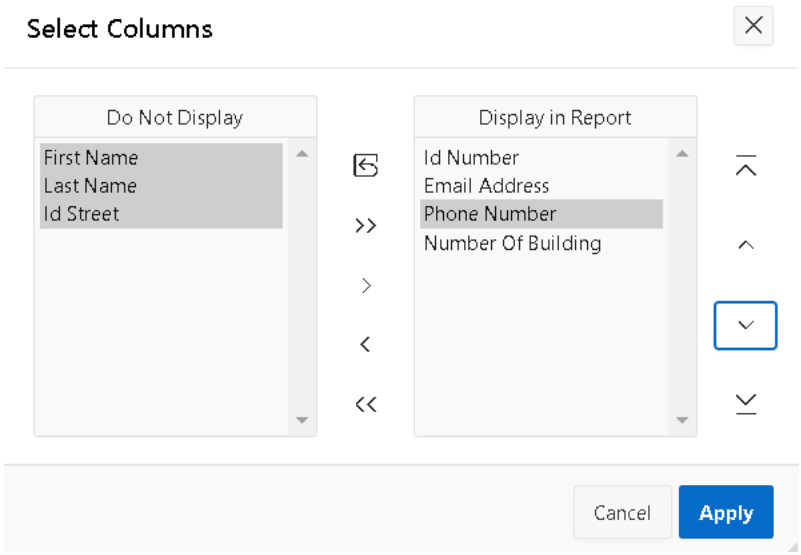
Rysunek 6.28: Wyszukiwanie.

Przycisk **Actions** oferuje więcej możliwości dostosowania wyświetlania raportu. Po kliknięciu pojawi się podmenu zawierające kilka pozycji. Wymienione pozycje pokazano na rysunku 6.29.



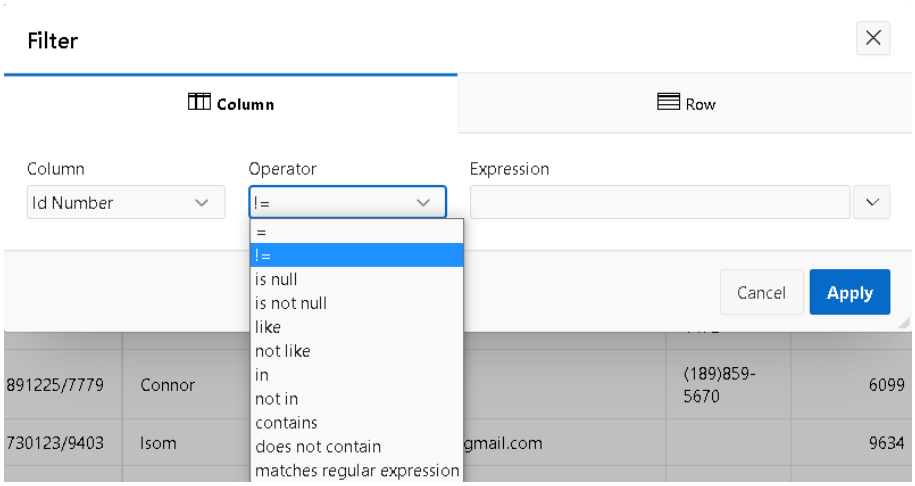
Rysunek 6.29: Przycisk akcji.

Przycisk **Columns** wyświetli okno, które mogą Państwo zobaczyć na rysunku 6.30. Tutaj użytkownik może wybrać, które kolumny będą wyświetlane w raporcie, a także kolejność, w jakiej powinny być wyświetlane. Użytkownik może to osiągnąć, przenosząc kolumny między dwiema sekcjami **Nie wyświetlaj** i **Wyświetl w raporcie**. Pomocne są strzałki, które ułatwiają pracę z przenoszeniem kolumn.



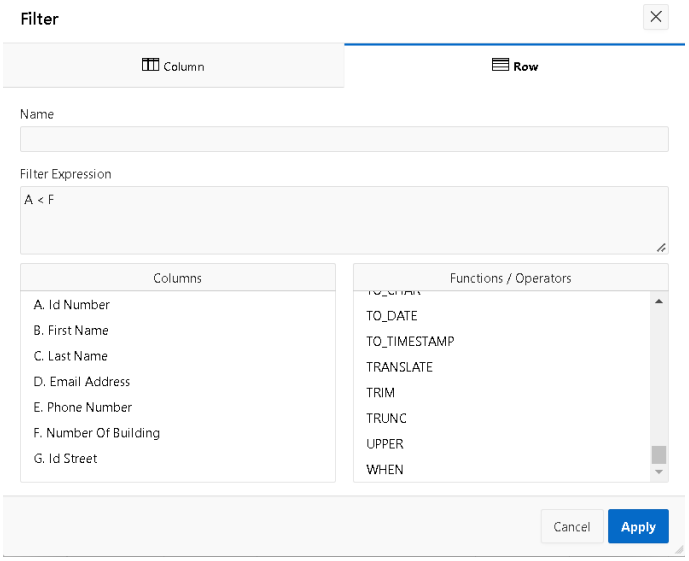
Rysunek 6.30: Wybór kolumn do wyświetlenia.

Inną opcją jest **Filter** pokazany na rysunku 6.31. Możemy filtrować wartości w kolumnie za pomocą wyrażeń takich jak *to be equal to, not equal, is null* itp. Wyrażenie można zdefiniować, wybierając spośród predefiniowanych lub pisząc własne.



Rysunek 6.31: Filtr kolumn.

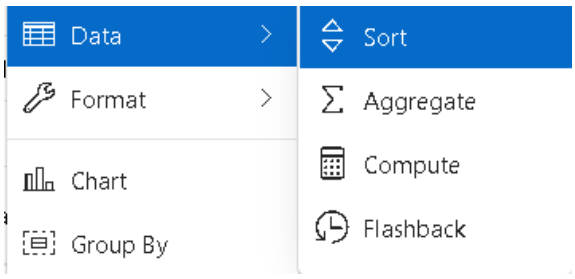
W przypadku filtrowania wierszy mamy do wyboru niezliczone operacje i wyrażenia, które możemy skomponować. Filtr ten pojawia się w oknie **Filter Expression** przedstawionym na rysunku 6.32.



Rysunek 6.32: Filtr wierszy.

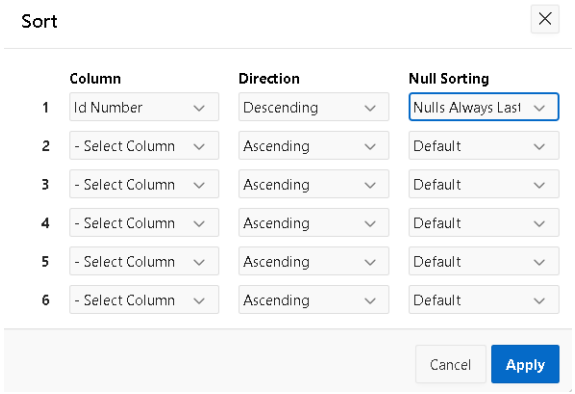
Po kliknięciu przycisku **Data** pojawi się menu pokazane na rysunku 6.33. Menu zawiera kolejne cztery możliwości:

- sortowanie,
- agregatowanie,
- obliczanie,
- flashback.



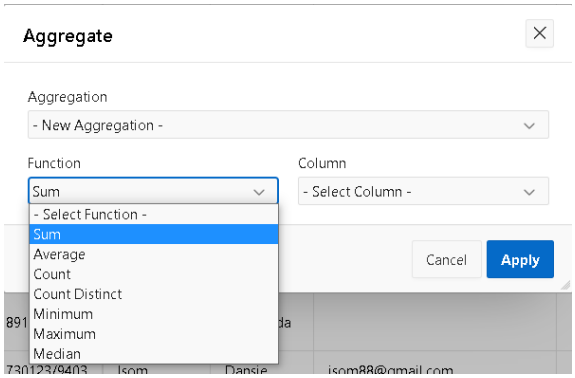
Rysunek 6.33: Po kliknięciu przycisku Dane.

Na rysunku 6.34 widać opcję **Sort**, którą można zastosować do każdej kolumny, a użytkownik może określić sposób pracy z opcją *null values*..



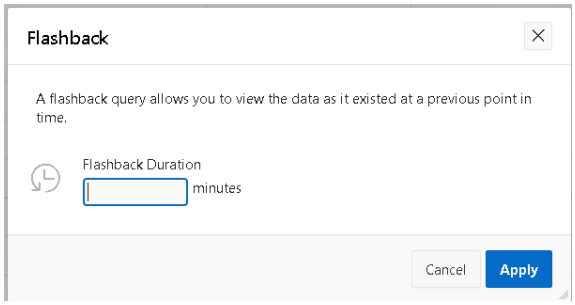
Rysunek 6.34: Sortowanie.

Za pomocą **Aggregate** możemy wykonać funkcję agregacji nad jedną z kolumn naszego raportu. Funkcje agregacji, które możemy wybrać, są pokazane na rysunku 6.35.



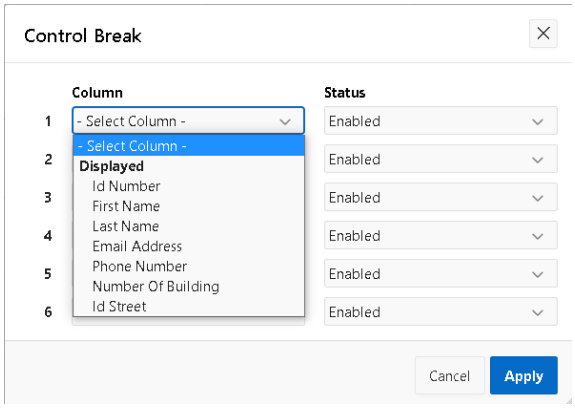
Rysunek 6.35: Agregowanie.

Funkcja Flashback przedstawiona na rysunku 6.36 pozwala nam powrócić do stanu wyświetlanego raportu przed zmianami sprzed kilku minut.



Rysunek 6.36: Flashback.

Control Break pokazany na rysunku 6.37 pozwala nam podzielić raport na kilka mniejszych jednostek, tak jakby były oddzielnymi raportami. Te małe jednostki są podzielone w zależności od wartości kolumn. Każda jednostka współdzieli te same dane w wybranej kolumnie.



Rysunek 6.37: Control Break.

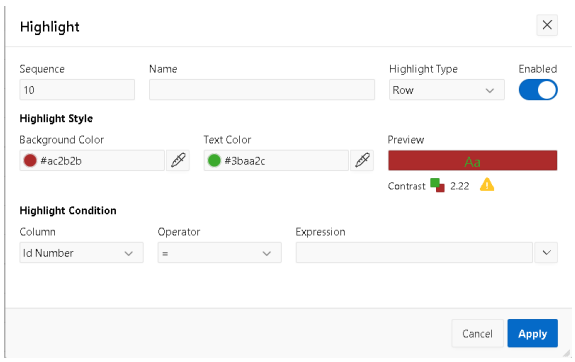
Rysunek 6.38 pokazuje nam przerwę kontrolną zastosowaną w kolumnie *First Name*.

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

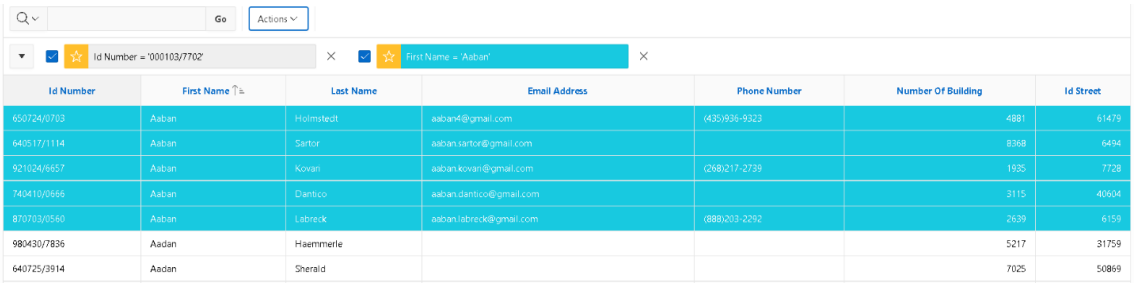
Rysunek 6.38: Wynik Control Break.

Na rysunku 6.39 można zobaczyć funkcję **Highlight**, która w prosty sposób sprawia, że rekordy w raporcie są przezroczyste, bez odfiltrowywania pozostałych. Możemy podświetlić jedną komórkę lub cały wiersz wybranymi kolorami. Porównywanie wartości kolumn działa tak samo jak w przypadku **Filter**, z tą różnicą, że niefiltrowane rekordy nie znikają z raportu.



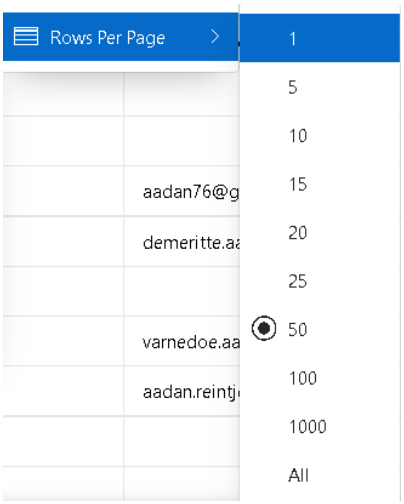
Rysunek 6.39: Highlight.

Na rysunku 6.40 możemy zobaczyć raport z podświetlonymi wierszami, z imieniem równym **Aaban"**.



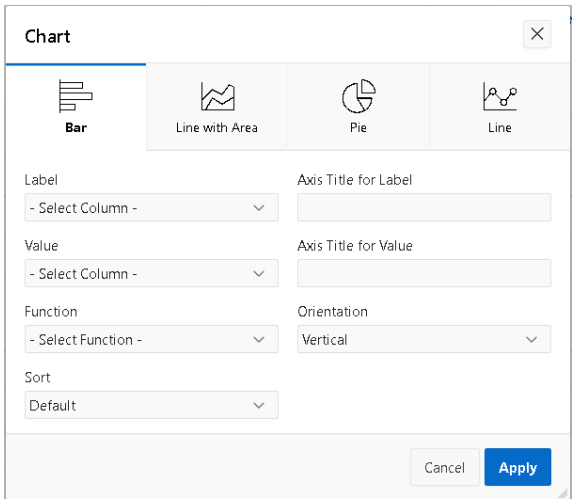
Rysunek 6.40: Najważniejszy wynik.

Rysunek 6.41 pokazuje **Rows Per Page**, który jest atrybutem, w którym użytkownik może wybrać, ile rekordów będzie wyświetlanych na stronie. Jest to ustawienie podobne do **Pagination**.



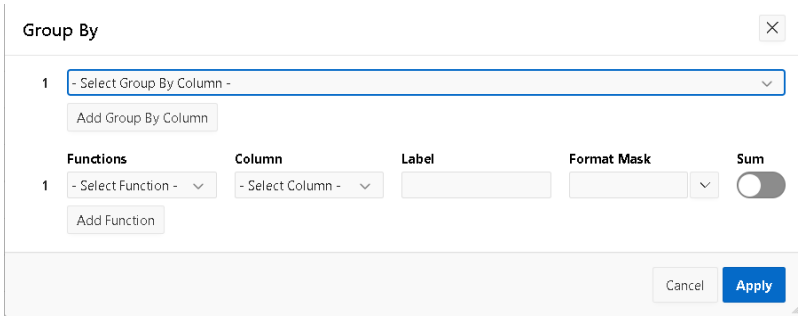
Rysunek 6.41: Wiersz na stronę.

Jeśli wyświetlanie danych w raporcie jest niewystarczające, istnieje również możliwość rzutowania tych danych na wykres za pomocą elementu **Chart** przedstawionego na rysunku 6.42. Do wyboru są **cztery** podstawowe wykresy. Możemy również wybrać funkcję agregacji, która zostanie zastosowana do danej kolumny.



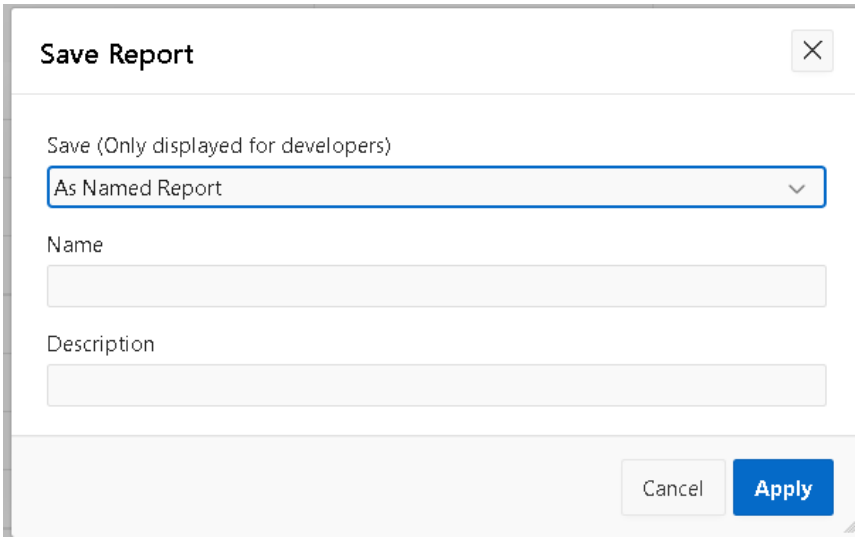
Rysunek 6.42: Wykres w raporcie interaktywnym.

Ponadto, bardzo ważną funkcjonalnością raportu interaktywnego jest **Group By** pokazane na rysunku 6.43. Działa ona w taki sam sposób jak w zapytaniu SQL.



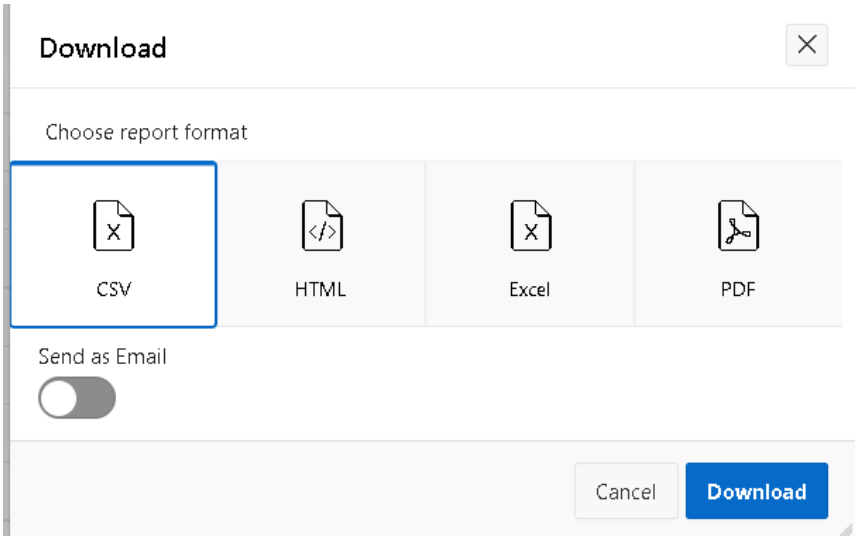
Rysunek 6.43: Grupuj.

Save Report przedstawiony na rysunku 6.44 zapisuje raport w jego bieżącym stanie. Jednak raport ten może być widoczny tylko dla dewelopera po zalogowaniu się do aplikacji, a nie dla użytkownika.



Rysunek 6.44: Zapisz raport.

Możemy również przywrócić raport do pierwotnego stanu lub pobrać go w wybranym formacie. Mogą Państwo zobaczyć tę opcję na rysunku 6.45.



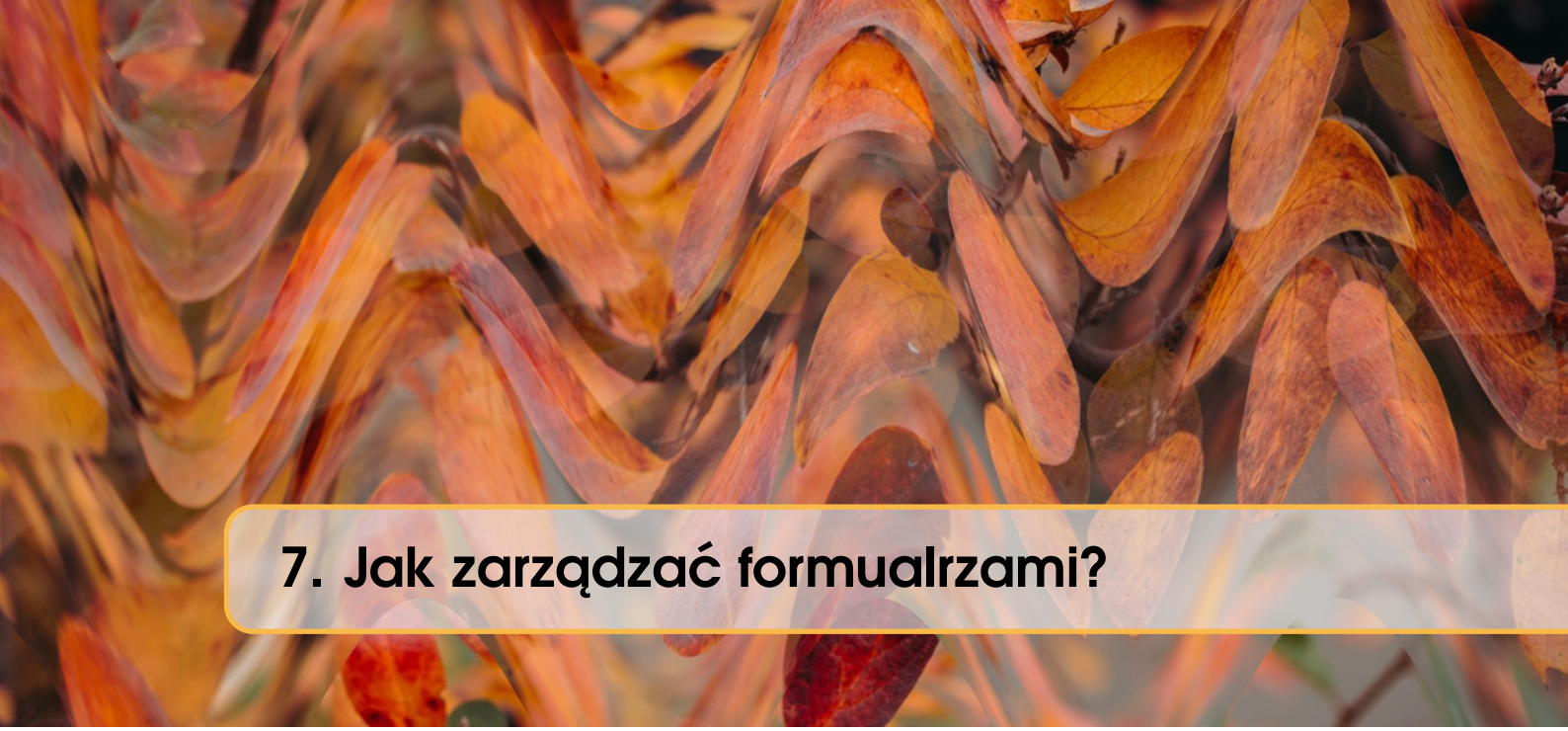
Rysunek 6.45: Pobierz.

6.5 Pytania

1. Do jakiej kategorii należy typ raportu? Regiony, elementy czy przyciski?
2. Co może być źródłem danych dla raportu?
3. Jakich typów źródeł można użyć do określenia raportu klasycznego?
4. Jaka jest główna cecha raportu Classic?
5. Która technologia pozwala użytkownikowi uzyskać obraz danych, jaki istniał w przeszłości przy użyciu raportu interaktywnego?

6.6 Answers

1. Raport to region upraszczający przeglądanie i wyszukiwanie danych.
2. Dane mogą być ładowane z lokalnej bazy danych, źródła SQL lub źródła REST.
3. Dostępne typy źródeł to Tabela lub Zapytanie SQL.
4. Główną cechą klasycznego raportu jest tryb tylko do odczytu, uniemożliwiający użytkownikowi modyfikowanie opcji wyświetlania lub danych.
5. Funkcje Flashback pozwalają użytkownikowi skonstruować obraz danych, jaki istniał w określonym punkcie czasowym.



7. Jak zarządzać formualrzami?

VERONIKA ŠALGOVÁ, MICHAL KVET, MIROSLAV POTOČÁR

7.1 Rodzaje formularzy

W Oracle APEX możemy korzystać z trzech podstawowych typów formularzy:

- edytowalna interaktywna siatka,
- formularz w tabeli,
- główny formularz szczegółowy.

7.2 Edytowalna interaktywna siatka

W edytowalnym interaktywnym raporcie można dodawać, modyfikować, usuwać i wyszukiwać dane. Jest to bogaty w funkcje komponent z możliwościami edycji danych dostępnymi bezpośrednio na stronie. Ponadto, raport ten może być dostosowywany i zmieniany interaktywnie za pomocą myszy, bezpośrednio aktualizując strukturę i zawartość siatki. Możliwe jest utworzenie edytowalnej interaktywnej siatki jako konwersji interaktywnej siatki tylko do odczytu, która została utworzona poprzez wybranie *Report*, a następnie *Interactive Grid*. Inną możliwością jest utworzenie *Report and Form* i wybranie typu raportu jako *Interactive Grid*, który również zostanie utworzony jako wersja tylko do odczytu, ale może zostać przekonwertowany na formę edytowalną.

Customers

Search: All Text Columns | Go | Actions | Reset

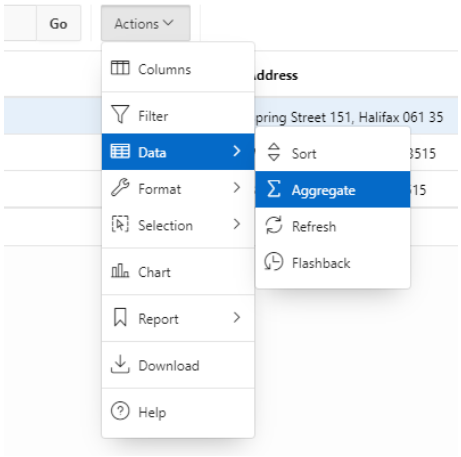
Personid	Address	Phone
ADMIN	Spring Street 151, Halifax 061 35	+421932165815
CUSTOMER	White Street 135, Halifax 13515	0312913258
MANAGER	Blue Street 815, Halifax 13515	0913456258

1 rows selected | Total 3

Rysunek 7.1: Interaktywna siatka z klientam

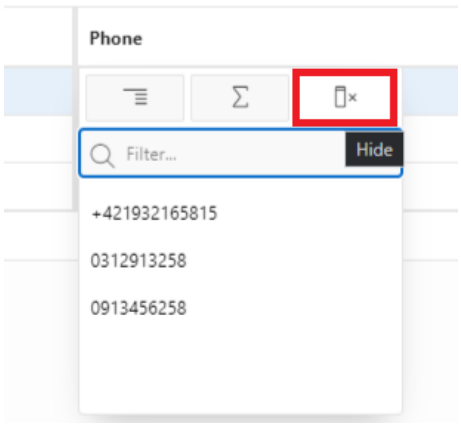
Dane można wyszukiwać za pomocą tekstu na pasku wyszukiwania w górnej części strony.

Inne działania dostępne w tej interaktywnej siatce pokazano na rysunku 7.2.



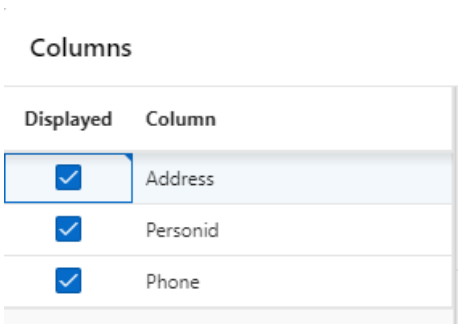
Rysunek 7.2: Działania interaktywnej siatki

Można zmienić rozmiar kolumny siatki, klikając i przytrzymując krawędź nagłówka kolumny i dostosowując ją za pomocą myszy. Można również ukryć kolumnę, wybierając nagłówek i klikając ikonę *Hide*, jak pokazano na rysunku 7.3.



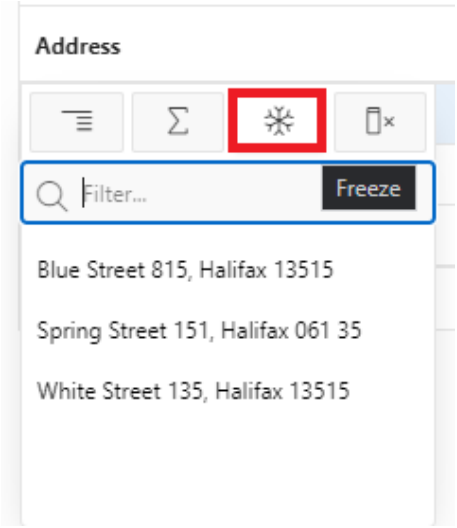
Rysunek 7.3: Ukryj ikonę kolumny Telefon

Po ukryciu kolumny nie będzie ona już wyświetlana w siatce. Aby ponownie wyświetlić kolumnę, należy kliknąć opcję *Actions*, wybrać opcję *Columns* i kliknąć kolumny, które mają zostać wyświetlone.



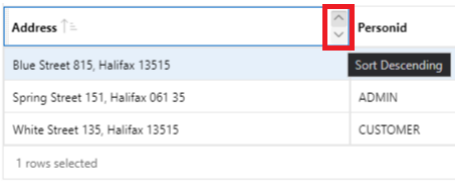
Rysunek 7.4: Wyświetlane kolumny

Kolumnę można zamrozić po kliknięciu ikony *Freeze*, jak pokazano na rysunku 7.5. Zamrożenie kolumny wyklucza ją z obszaru przewijania.



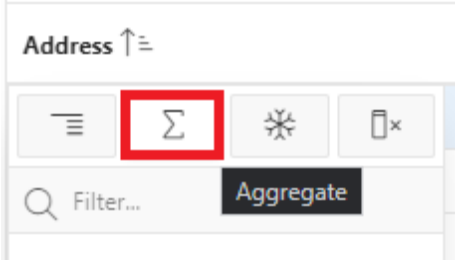
Rysunek 7.5: Ikona zamrożenia kolumny Adres

Dane można sortować według kolumny, wybierając ikonę *Sort Ascending* lub *Sort Descending*, jak pokazano na rysunku 7.6.



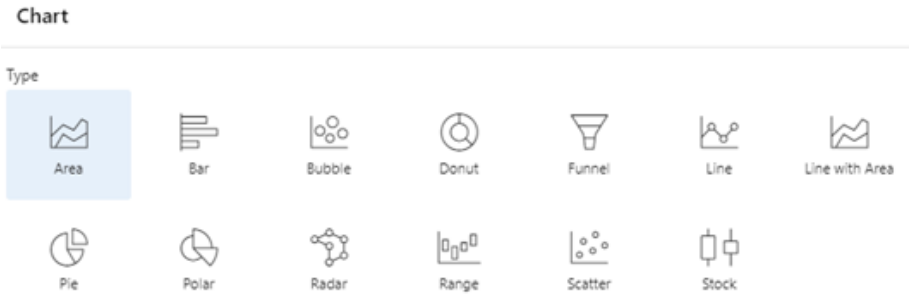
Rysunek 7.6: Sortowanie kolumny Adres

Za pomocą ikony *Aggregate* kolumny, jak pokazano na rysunku 7.7, można wybrać funkcje Count, Count Distinct, Minimum i Maximum dla wybranej kolumny.



Rysunek 7.7: Ikona zagregowana kolumny Adres

Dane z siatki można wykorzystać do tworzenia wykresów, wybierając opcje *Actions*, a następnie *Charts*. Dostępne są różne typy wykresów, jak pokazano na rysunku 7.8. Można wybierać spośród różnych funkcji agregacji dostarczonych z wykresami.



Rysunek 7.8: Rodzaje wykresów

7.3 Formularz w tabeli

Oracle APEX zapewnia nam podstawowe formularze do wstawiania pojedynczego wiersza do tabeli, jak pokazano na rysunku 7.9. Po kliknięciu przycisku *Create a Page* należy wybrać opcję *Form*. Należy wybrać tabelę połączoną z tworzonym formularzem lub można napisać zapytanie SQL, a także kolumnę klucza podstawowego tabeli.

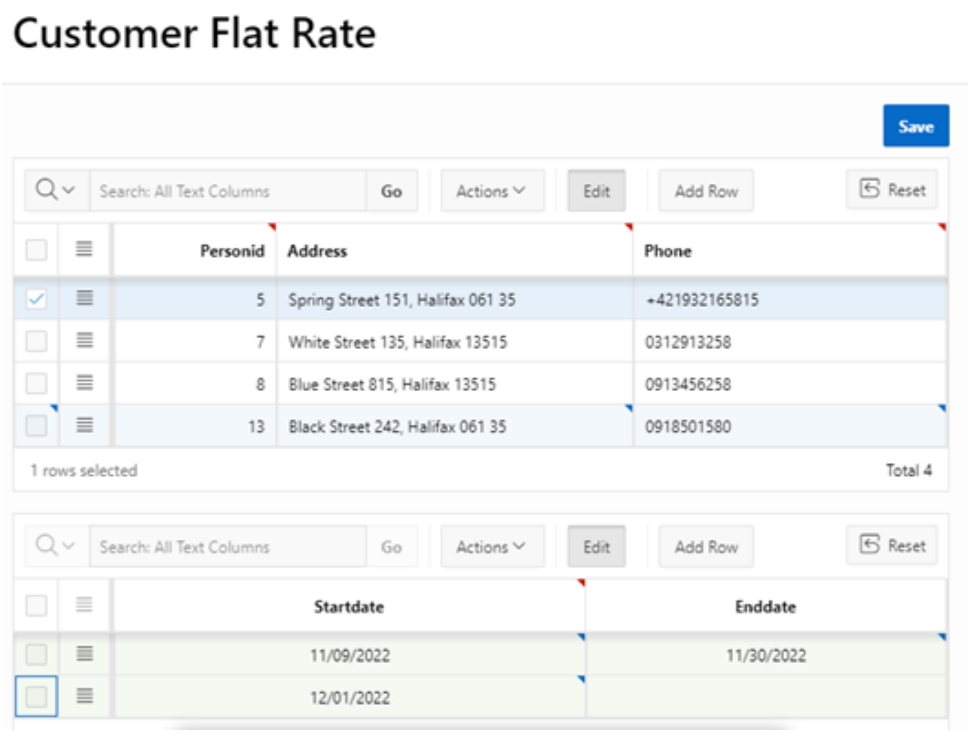


Rysunek 7.9: Formularz dodawania klientów

7.4 Główny formularz szczegółowy

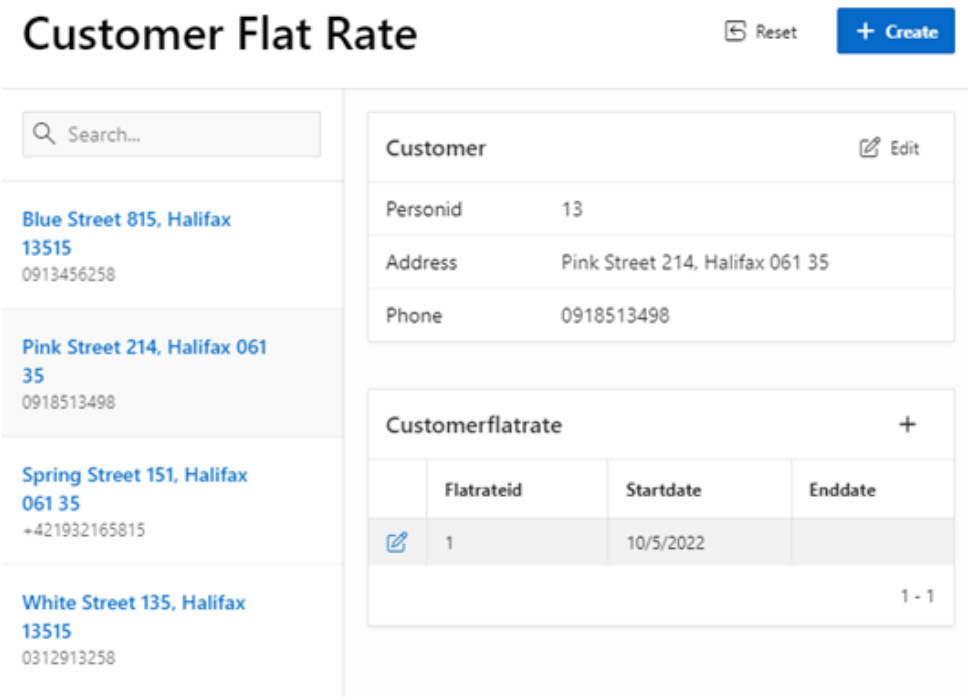
Możliwe jest odpytywanie, wstawianie, aktualizowanie i usuwanie wartości z dwóch powiązanych tabel lub widoków. Po kliknięciu przycisku *Create a Page* należy wybrać opcję *Master Detail*. W kolejnym kroku należy wybrać styl szczegółów głównych, taki jak *Stacked*, *Side by Side* lub *Drill Down*. Formularz szczegółów głównych zazwyczaj wyświetla wiersz główny i wiele wierszy szczegółowych. Oznacza to, że należy wybrać tabelę lub widok jako główne źródło danych, a inny jako szczegółowe źródło danych. Po wybraniu klucza głównego każdej tabeli, należy wybrać klucz obcy szczegółu głównego.

Rysunek 7.10 przedstawia główny formularz szczegółów *Stacked*. Zawiera on edytowalne interaktywne siatki. Użytkownicy mogą wybrać wiersz w siatce głównej, aby zaktualizować siatki szczegółowe.



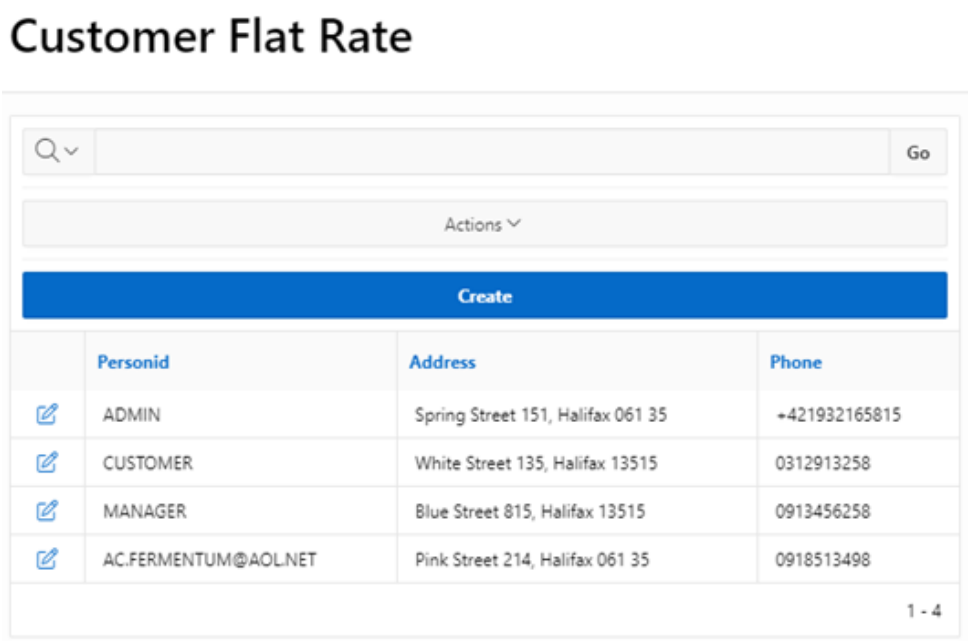
Rysunek 7.10: Skumulowany szczegółowy formularz zryczałtowanych stawek klienta

Na rysunku 7.11 znajduje się formularz szczegółów wzorcowych *Side by Side* zawierający jednostronicowe szczegóły wzorcowe wykorzystujące układ side-by-side i regiony raportów z modalnymi oknami edycji. Po lewej stronie można wybrać z listy głównej, aby przejść do rekordu głównego. Prawa strona zawiera wybrany rekord główny i powiązane raporty szczegółowe.

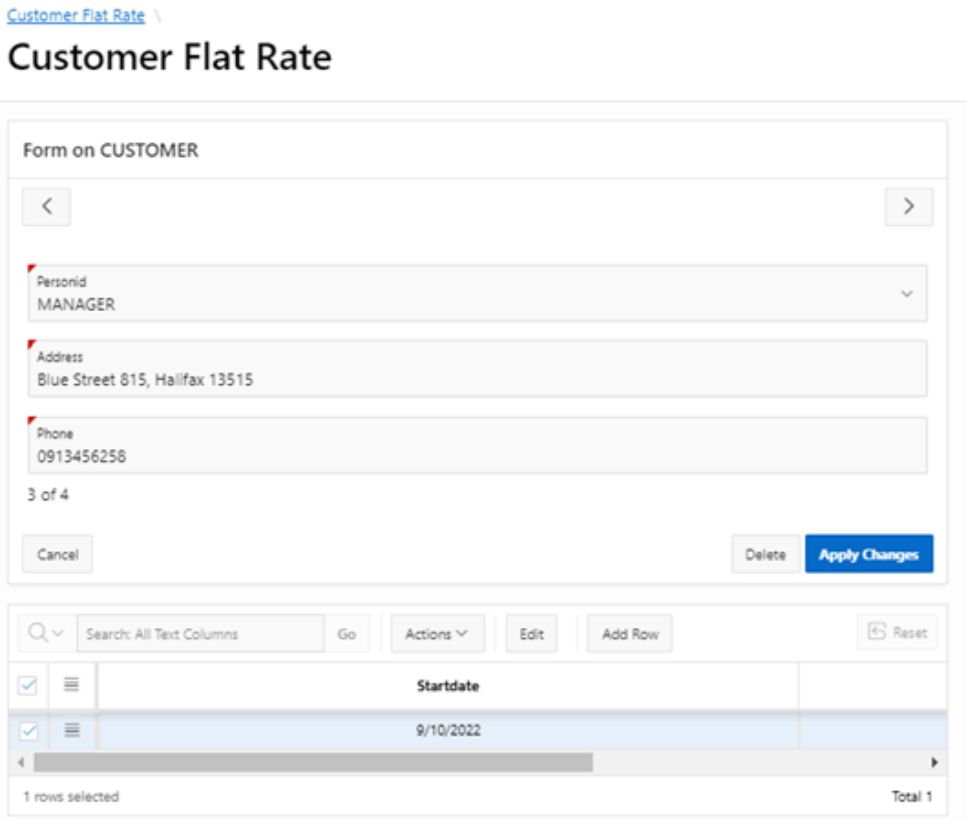


Rysunek 7.11: Obok siebie główny formularz szczegółowy zryczałtowanych stawek klienta

Master detail *Drill Down* zawiera dwie strony. Pierwsza strona zawiera interaktywny raport dla tabeli głównej, jak pokazano na rysunku 7.12. Na drugiej stronie, jak pokazano na rysunku 7.13, znajduje się standardowy formularz dla tabeli głównej i edytowalne interaktywne siatki dla szczegółów.



Rysunek 7.12: Formularz Drill Down master detail – pierwsza strona



Rysunek 7.13: Formularz Drill Down master detail – druga strona

7.5 Pytania

1. Czy możliwe jest uporządkowanie danych za pomocą edytowalnej interaktywnej siatki?
2. Którą definicję komponentu tabeli należy wybrać podczas tworzenia formularza?
3. Która opcja wyklucza kolumnę z obszaru przewijania?

7.6 Odpowiedzi

1. Tak, raporty mogą być dostosowywane i zmieniane interaktywnie za pomocą myszy, bezpośrednio aktualizując strukturę i zawartość siatki..
2. Należy określić atrybuty klucza głównego.
3. Kolumnę można zamrozić po kliknięciu ikony Zamroź. Zamrożenie kolumny wyklucza ją z obszaru przewijalnego.

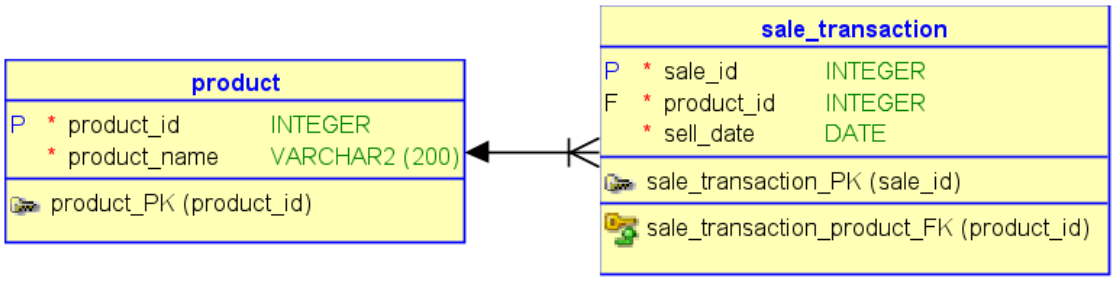
8. Jak przekształcić raporty tekstowe w wykresy?

IVAN PASTIERIK, MICHAL KVET, MIROSLAV POTOČÁR

8.1 Wykresy

Wykresy w Oracle APEX to wszechstronne komponenty, które umożliwiają programistom i użytkownikom uzyskanie głębszego wglądu w dane poprzez ich wizualną reprezentację. Te dynamiczne wykresy, począwszy od wykresów słupkowych i liniowych, a skończywszy na wykresach kołowych i innych, służą jako ważne zasoby poprawiające wrażenia użytkownika i ułatwiające podejmowanie decyzji opartych na danych.

W tym rozdziale dowiedzą się Państwo, jak utworzyć prosty wykres z funkcjami **filtrowanie** i **sortowanie**. **Filtrowanie** i **sortowanie** mogą znacznie poprawić interaktywność i czytelność wykresów. W tym celu stworzyliśmy prosty model danych pokazany na rysunku 8.1. Model danych składa się z dwóch encji: **“product”** i **“sale_transaction”**. Podmiot **“product”** reprezentuje pojedynczy produkt, którego atrybut klucza głównego to **“sale_id”**, a nazwa produktu jest przechowywana w atrybucie **“product_name”**. Podmiot **“sale_transaction”** przechowuje informacje o wszystkich naszych sprzedanych produktach z kluczem podstawowym **“sale_id”**. Atrybut **“product_id”** jest kluczem obcym, który odwołuje się do atrybutu **“product_id”** w encji **“product”**. Atrybut **“sell_date”** zawiera datę sprzedaży produktu.



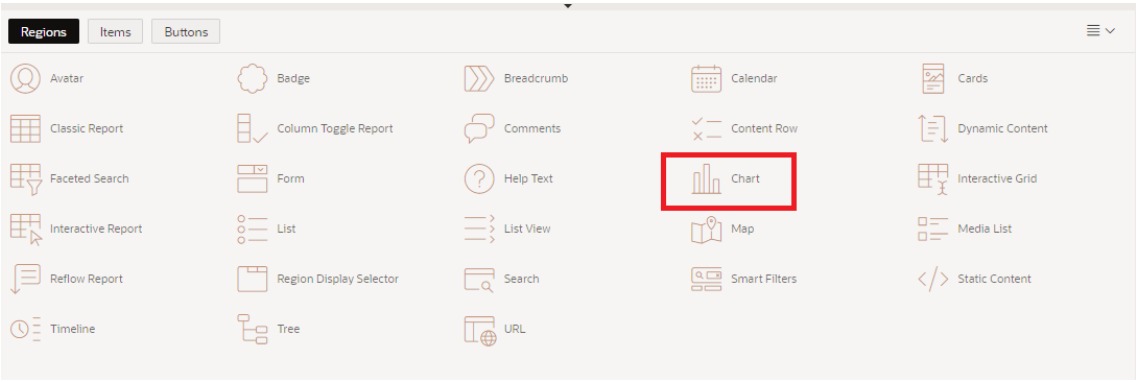
Rysunek 8.1: Model danych używany w przykładowej aplikacji.

Aby rozpocząć, należy utworzyć nową aplikację, którą nazwiemy **“Chart Showcase”**. Aplikacja ta będzie zawierać **“Global Page”**, **“Home”** i **“Login Page”**. Będziemy tworzyć wykresy na stronie głównej. Szczegółowe instrukcje dotyczące tworzenia aplikacji zostały opisane w rozdziale

5. Po utworzeniu aplikacji proszę wprowadzić **Page Designer** dla strony **“Home”** w naszej nowo utworzonej aplikacji.

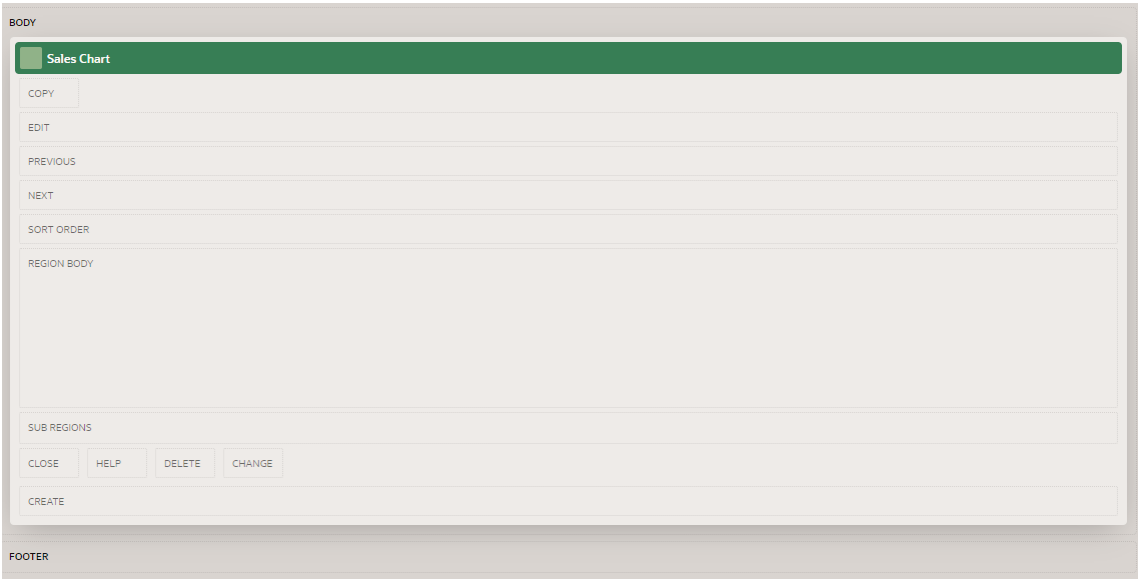
8.2 Tworzenie wykresu słupkowego

Aby umieścić region wykresu, podświetlony na rysunku 8.2, w naszej aplikacji, musimy przeciągnąć i upuścić go do sekcji układu **Page Designer**.



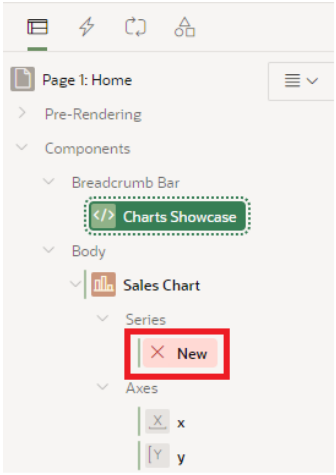
Rysunek 8.2: Wybieranie i umieszczanie regionu wykresu.

Umieścimy komponent **Chart** w sekcji **body** naszej strony, ponieważ chcemy, aby nasz wykres był główną treścią strony. Ustawimy **title** naszego nowo umieszczonego regionu Chart na **“Sales Chart”**. Na rysunku 8.3 można zobaczyć, jak wygląda sekcja główna naszego układu po umieszczeniu i zmianie tytułu naszego regionu **Chart**.



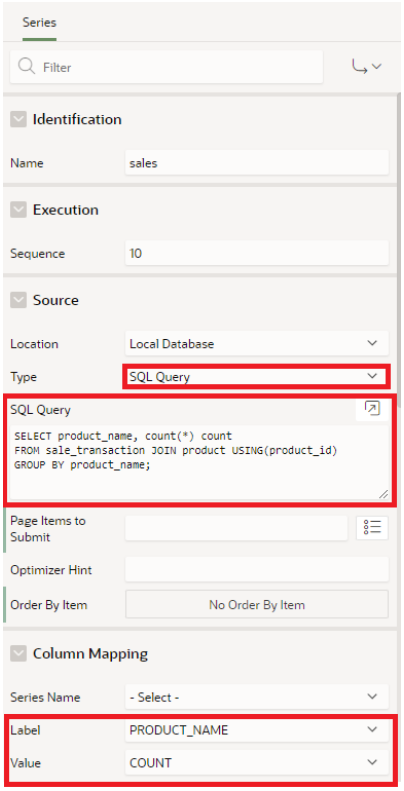
Rysunek 8.3: Układ sekcji głównej po umieszczeniu regionu wykresu.

W lewej części **Page Designer** można zobaczyć **przegląd drzewa** wszystkich umieszczonych przez nas komponentów. Jeśli przyjrzeć się bliżej naszemu komponentowi **Wykres sprzedaży**, wewnątrz **“Series”**, pokazuje on błąd. Błąd ten, jak widać na rysunku 8.4, jest spowodowany tym, że nie zdefiniowaliśmy źródła naszych danych, które chcemy wyświetlić na naszym wykresie.



Rysunek 8.4: Przegląd drzewa pokazujący błąd w serii wykresów.

Po kliknięciu na serię oznaczoną jako **"New"** w przeglądzie drzewa, menu ustawień dla serii otworzy się po prawej stronie **Page Designer**. Wewnątrz tego menu ustawień zmienimy nazwę serii na **"Sales"**, ponieważ chcemy pokazać liczbę sprzedaży naszych poszczególnych produktów w naszym **Sales Chart**. Następnie musimy zdefiniować serię **source**. Aby to zrobić, zmienimy typ źródła na **"SQL Query"**, a wewnątrz zapytania SQL umieścimy instrukcję **SELECT**, jak pokazano na rysunku 8.5. Ta instrukcja **SELECT** zwróci całkowitą liczbę sprzedaży pogrupowanych według nazw produktów. Teraz, gdy zdefiniowaliśmy źródło naszych danych, które chcemy pokazać na wykresie, musimy zmapować kolumny wybrane przez instrukcję **SELECT** na osie **Label (X)** i **Value (Y)**. Ostateczna konfiguracja naszej serii jest pokazana na rysunku 8.5 z zaznaczonymi ważnymi zmianami.



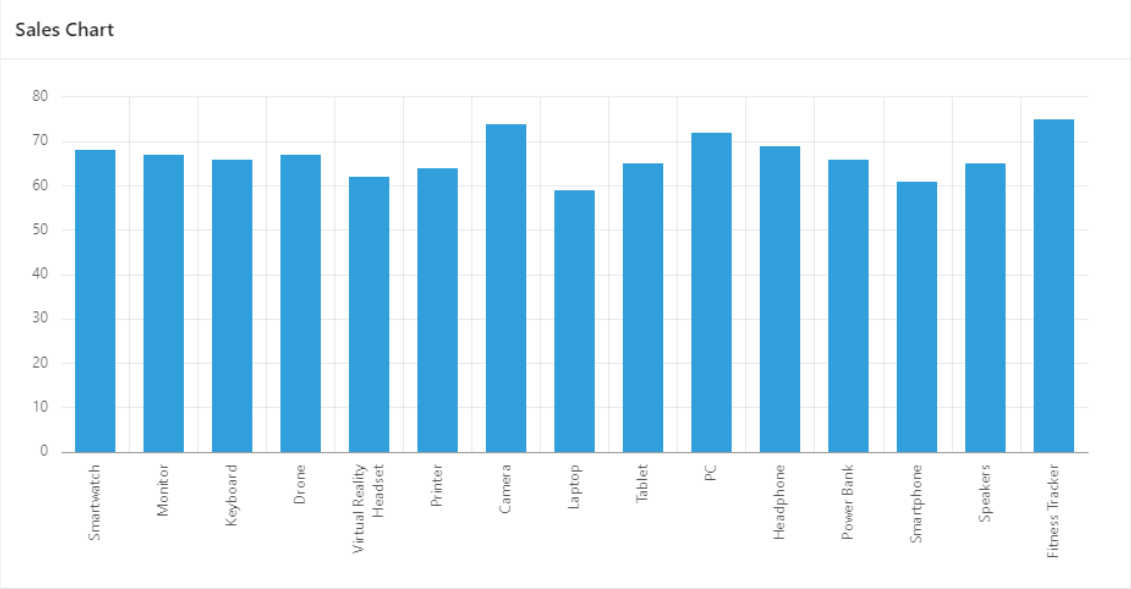
Rysunek 8.5: Końcowa konfiguracja serii sprzedaży

Jeśli weźmiemy zapytanie SQL z rysunku 8.5 i wykonamy je, zwróci ono dane pokazane na rysunku 8.6.

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

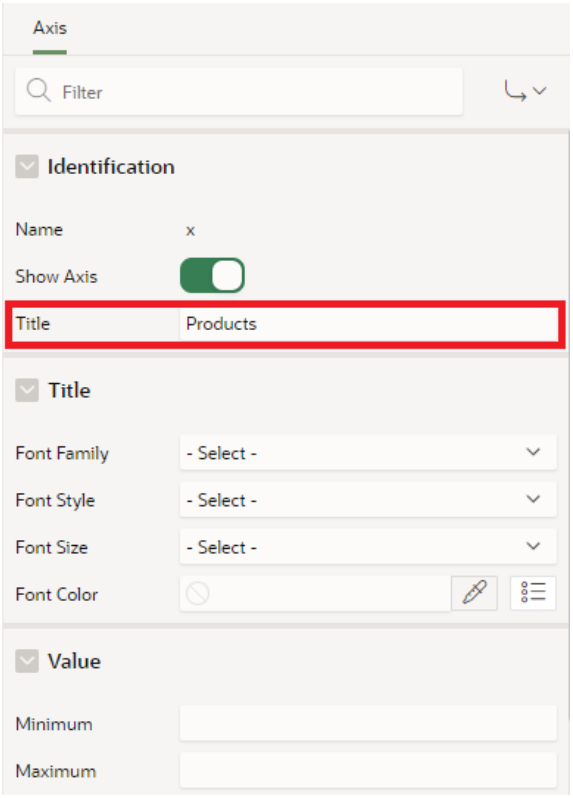
Rysunek 8.6: Dane zwrócone przez wykonanie zapytania SQL.

Teraz możemy zapisać naszą stronę i uruchomić ją. Proszę nie zapomnieć ustawić **authentication** strony na **“Page Is Public”**, aby nie wymagała logowania. Po uruchomieniu strony powinni Państwo otrzymać wyniki podobne do tych pokazanych na rysunku 8.7. Można jednak zauważyć, że nasz wykres nie ma oznaczonych osi **X** i **Y**, co utrudnia jego odczyt.



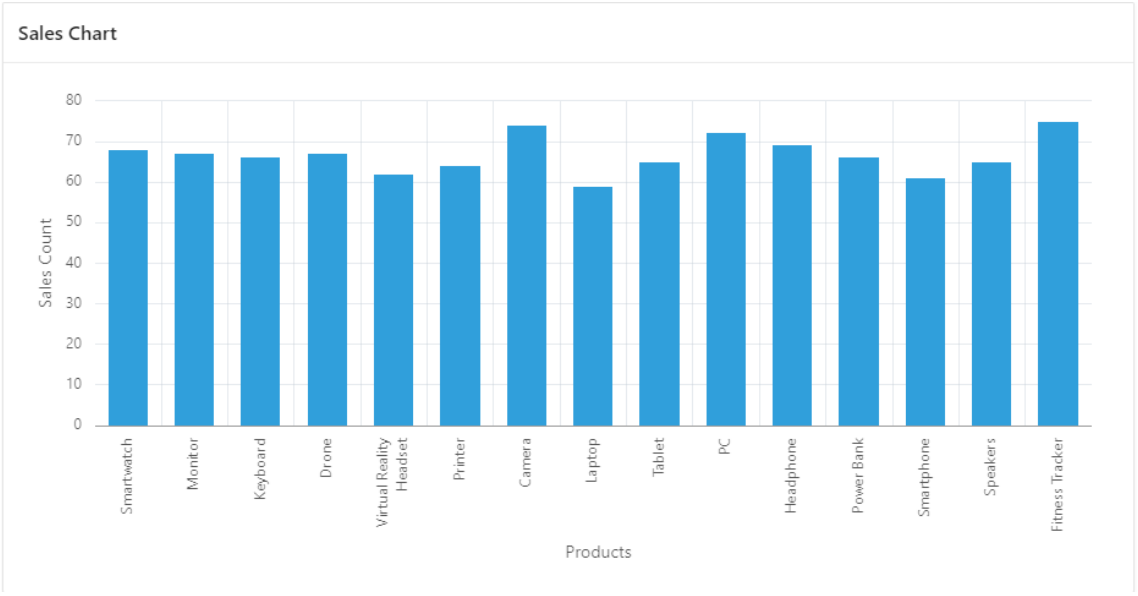
Rysunek 8.7: Wykres wyświetlany podczas uruchamiania aplikacji.

Aby temu zaradzić, możemy oznaczyć nasze osie podobnie do tego, jak oznaczyliśmy nasze serie. Klikając poszczególne osie w lewym przeglądzie drzewa i zmieniając ich atrybut **Title**, jak pokazano na rysunku 8.8, możemy nazwać oś **X "Products"**, a oś **Y "Sales Count"**.



Rysunek 8.8: Zmiana atrybutu tytułu dla osi x.

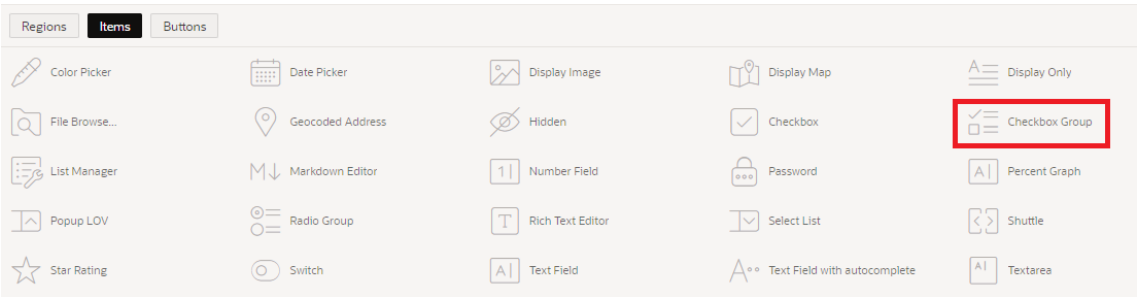
Po oznaczeniu obu osi naszego wykresu, wynik powinien przypominać wykres pokazany na rysunku 8.9.



Rysunek 8.9: Wykres sprzedaży po oznaczeniu osi.

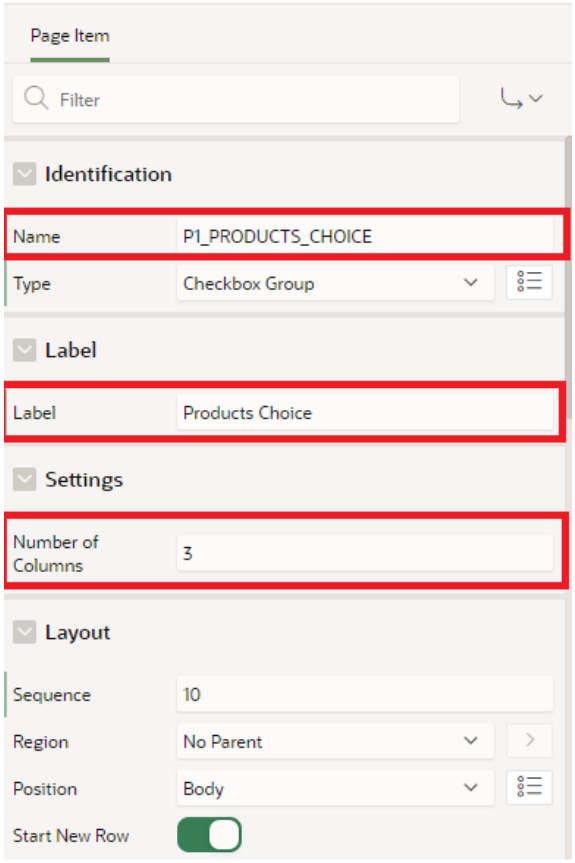
8.3 Dodawanie filtrowania do wykresu słupkowego

Teraz dodajmy **filtrowanie** dla produktów do naszego wykresu, pozwalając nam wybrać, które produkty mają być wyświetlane. Aby to osiągnąć, dodamy element **Checkbox Group** z dolnego paska narzędzi **Page Designer**, umieszczając go tuż nad naszym regionem **Sales Chart** w treści.



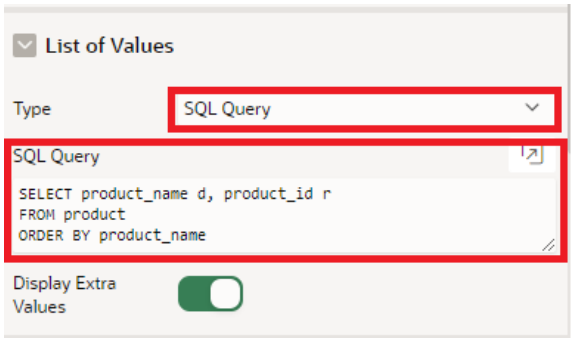
Rysunek 8.10: Wybieranie i umieszczanie elementu Checkbox Group.

Nazwiemy element **Checkbox Group "P1_PRODUCTS_CHOICE"** i ustawimy jego etykietę na **"Products Choice"**. Dodatkowo możemy dostosować układ, aby wyświetlić go w trzech kolumnach, co pokazano na rysunku 8.11.



Rysunek 8.11: Ustawienie podstawowych atrybutów elementu Checkbox Group.

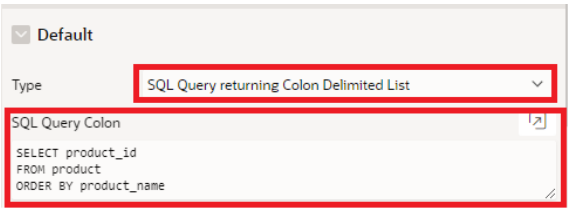
Aby wypełnić naszą grupę **Checkbox Group** produktami, musimy zdefiniować **list of values**. W sekcji **"List of Values"** ustawień **Page Item** zmienimy atrybut **"Type"** na **"SQL Query"** i napiszemy instrukcję **SELECT**, jak pokazano na rysunku 8.12. Instrukcja ta powinna wybrać dokładnie dwie kolumny: pierwszą kolumnę zawierającą wartości wyświetlane (z aliasem **"d"**) i drugą kolumnę zawierającą wartości zwracane (z aliasem **"r"**).



Rysunek 8.12: Ustawienie listy wartości dla elementu Checkbox Group.

Domyślnie element **Checkbox Group** ma zazwyczaj odznaczone wszystkie pola wyboru po uruchomieniu aplikacji. Jednak dla większej łatwości obsługi w przypadku korzystania z wykresów, lepiej jest, aby użytkownicy domyślnie widzieli wszystkie produkty wyświetlane na wykresie. Aby to osiągnąć, przejdziemy do sekcji **"Default"** w ustawieniach elementu **Checkbox Group** i zmienimy atrybut **"Type"** na **"SQL Query returning Colon Delimited List"**. Ten typ zapytania SQL pozwala nam zdefiniować, które wartości są wybierane domyślnie. Zwykłe zapytanie SQL

nie może być używane z elementem **Grupa pól wyboru**, ponieważ może zdefiniować tylko jedną wartość domyślną, a nie wiele wartości domyślnych. Wewnątrz **"SQL Query Colon"**, napiszemy instrukcję **SELECT**, jak pokazano na rysunku 8.13. Ta instrukcja **SELECT** powinna zwrócić dokładnie jedną kolumnę, która zostanie wewnętrznie przekonwertowana na listę **Colon Delimited List**, która może wyglądać następująco: **"1:5:7:2:3"**.

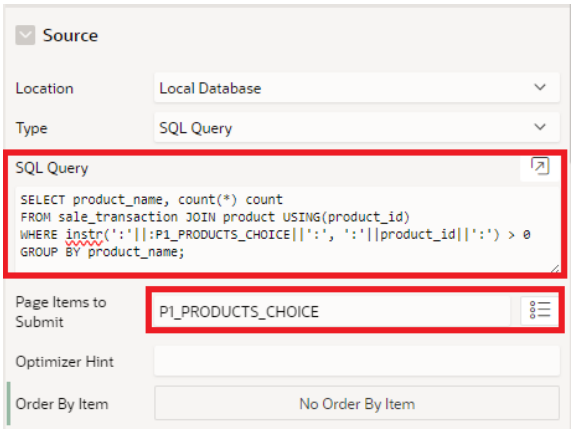


Rysunek 8.13: Ustawianie wartości domyślnych dla elementu Checkbox Group.

Teraz, po skonfigurowaniu elementu **Checkbox Group** zgodnie z naszymi upodobaniami, możemy połączyć go z naszym regionem **Sales Chart**. Aby to zrobić, dodamy nazwę naszego elementu **Checkbox Group** do pola **"Page Items to Submit"** w sekcji **"Source"** naszej serii **sales** wewnątrz regionu **Sales Chart**. Wymagane jest dodanie każdego elementu użytego w zapytaniu SQL do sekcji **"Page Items to Submit"**. Wartością zwrotną **"P1_PRODUCTS_CHOICE"** jest **Colon Delimited List**, która składa się z produktów, które są sprawdzane. Dlatego musimy użyć funkcji **INSTR**, która zwraca indeks pierwszego wystąpienia podciągu zdefiniowanego w drugim parametrze funkcji w ciągu zdefiniowanym w pierwszym parametrze funkcji. Funkcja **INSTR** zwróci **0**, jeśli w ciągu nie ma żadnego wystąpienia. Użyjemy funkcji **INSTR**, aby sprawdzić, czy identyfikator **"product_id"** każdego produktu znajduje się na wybranej liście identyfikatorów produktów w **"P1_PRODUCTS_CHOICE"**. Funkcja **INSTR** wyszukuje wystąpienie **':'||product_id||':'** w połączonym ciągu **':'||:P1_PRODUCTS_CHOICE||':'**. Jeśli znajdzie dopasowanie, oznacza to, że produkt znajduje się wśród wybranych produktów. Musimy połączyć **':'** zarówno z **"product_id"**, jak i **":P1_PRODUCTS_CHOICE"**, ponieważ chcemy wykonać precyzyjne dopasowanie między poszczególnymi identyfikatorami produktów. W ten sposób zapewniamy, że na przykład **":1: "** jest dopasowywany do **":1:2:3:8:9:11: "**. Gwarantuje to, że szukamy dokładnych dopasowań, zapewniając, że każdy wybrany identyfikator produktu jest rozpatrywany niezależnie

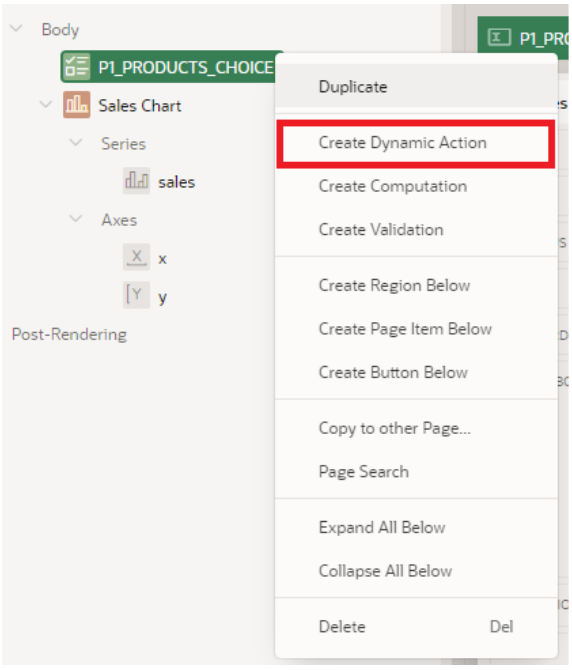
Gdybyśmy dopasowali **"1"** do **"1:2:3:8:9:11"**, moglibyśmy potencjalnie wyświetlić produkt o ID "1", gdy zaznaczone są pola wyboru dla produktów o ID **"1"**, **"11"** lub innych ID zawierających wartość **"1"**. Innymi słowy, jeśli sprawdzimy produkt o ID **"11"**, wyświetlony zostanie nie tylko wybrany produkt, ale także produkt o ID **"1"**. Prowadzi to do niepożądanych wyników i niedokładnej reprezentacji danych.

Dodając **':'** po obu stronach identyfikatorów, tworzymy spójny wzorzec ogranicznika, upewniając się, że wybrane identyfikatory produktów są dokładnie porównywane i eliminując wszelkie niejasności w procesie dopasowywania. Zapewnia to, że wykres wyświetla dane dokładnie na podstawie wyborów użytkownika.



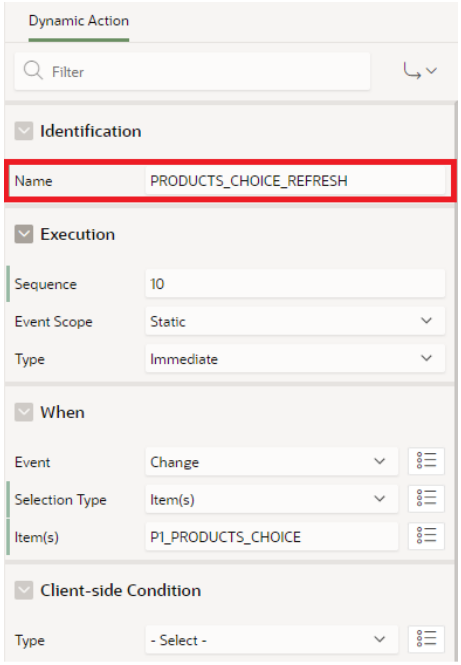
Rysunek 8.14: Powiązanie regionu wykresu sprzedaży z elementem grupy pól wyboru

Ostatnim krokiem konfiguracji jest ustawienie odświeżania naszego regionu **Sales Chart**, gdy nastąpi zmiana w wyborze elementu **Checkbox Group**. Aby to osiągnąć, musimy utworzyć **Dynamic Action** na naszym elemencie **"P1_PRODUCTS_CHOICE"**. Checkbox Group, jak pokazano na rysunku 8.15.



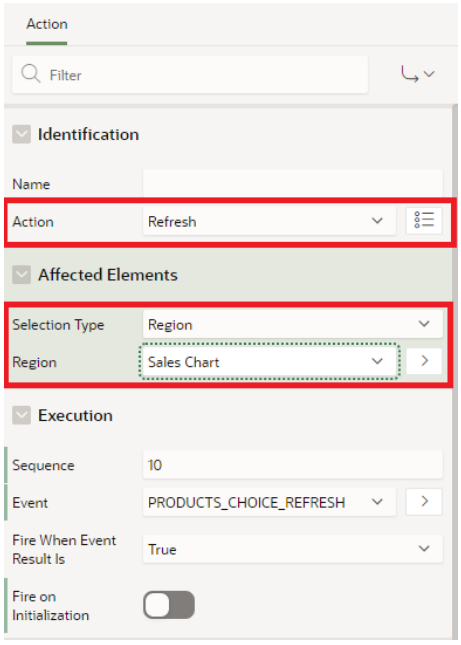
Rysunek 8.15: Tworzenie akcji dynamicznej dla elementu P1_PRODUCTS_CHOICE.

Możemy ustawić atrybut **"Name"** tego **Dynamic Action** na **"PRODUCTS_CHOICE_REFRESH"**, jak widać na rysunku 8.16. Musimy również upewnić się, że atrybut **"Event"** w sekcji **"When"** jest ustawiony na **"Change"** i że atrybut **"Selection Type"** jest ustawiony na **"Item(s)"**, a atrybut **"Item(s)"** jest ustawiony na naszą aktualną nazwę elementu, więc ta **Akcja dynamiczna** zostanie uruchomiona, gdy zmieni się wartość **"P1_PRODUCTS_CHOICE"**.



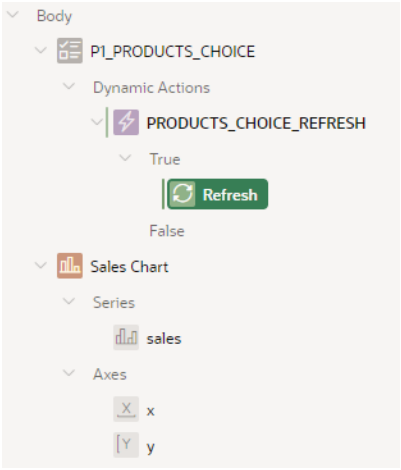
Rysunek 8.16: Ustawianie podstawowych atrybutów akcji dynamicznej przy zmianie.

W przeglądzie drzewa po lewej stronie **Page Designer**, proszę kliknąć na akcję, która wykonuje się, gdy warunek **Dynamic Action** ma wartość **true**, co ma miejsce, gdy zmienia się nasz element **Checkbox Group**. Po utworzeniu nowej akcji **Dynamic Action** zobaczą Państwo akcję **"Show"** z błędem, więc proszę ją kliknąć. W ustawieniach tej akcji proszę ustawić atrybut **"Action"** w sekcji **"Identification"** na **"Refresh"**. W sekcji **"Affected Elements"** proszę zmienić **"Selection Type"** na **"Region"**, a następnie ustawić **"Region"** na **"Sales Chart"**, jak pokazano na rysunku 8.17.



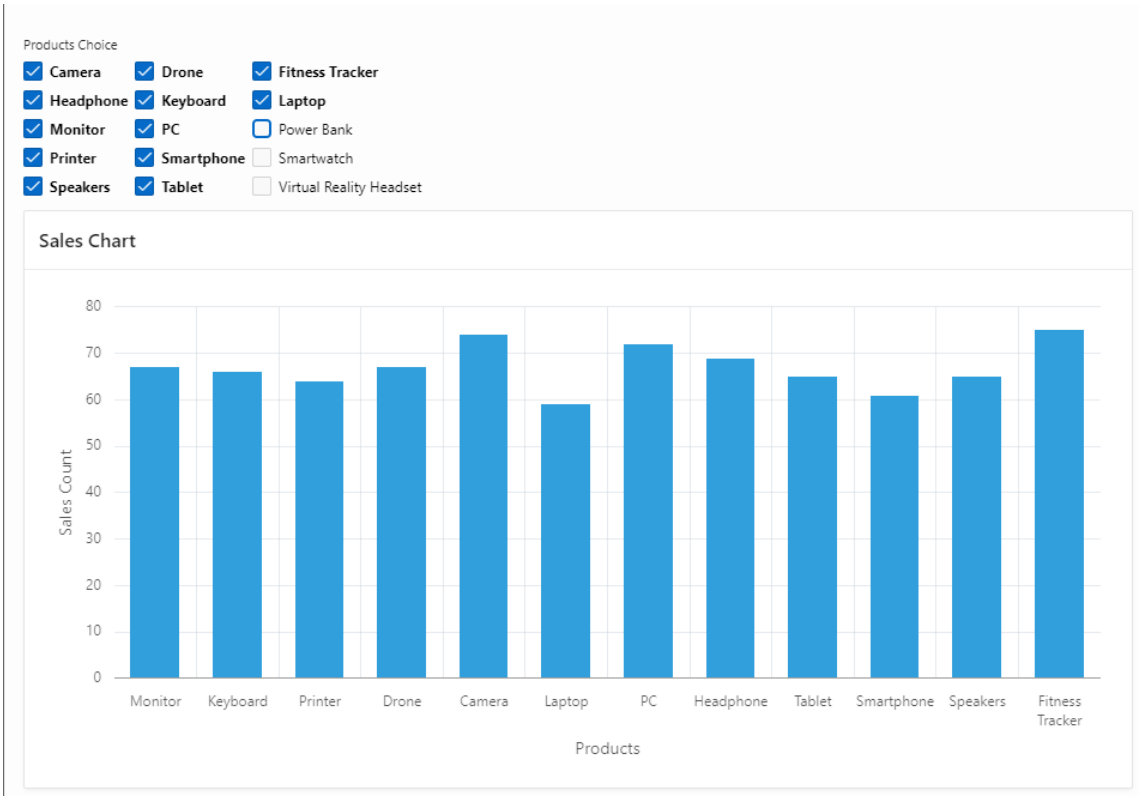
Rysunek 8.17: Ustawianie atrybutów akcji Refresh

Rysunek 8.18 pokazuje, jak powinien wyglądać przegląd drzewa w lewej części **Page Designer**, jeśli wszystkie kroki zostały wykonane poprawnie.



Rysunek 8.18: Dynamiczna akcja odświeżająca wykres sprzedaży w przeglądzie drzewa.

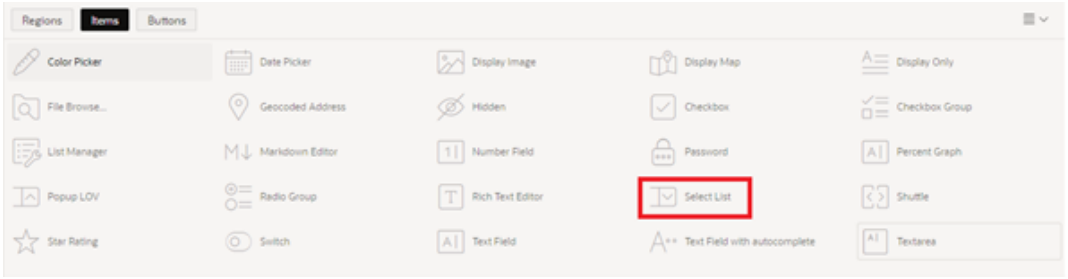
Kiedy uruchomimy i przetestujemy naszą aplikację, **filtrowanie** i **odświeżanie** naszego wykresu powinny być w pełni funkcjonalne, a aplikacja powinna przypominać tę pokazaną na rysunku 8.19.



Rysunek 8.19: Aplikacja po wdrożeniu filtrowania produktów.

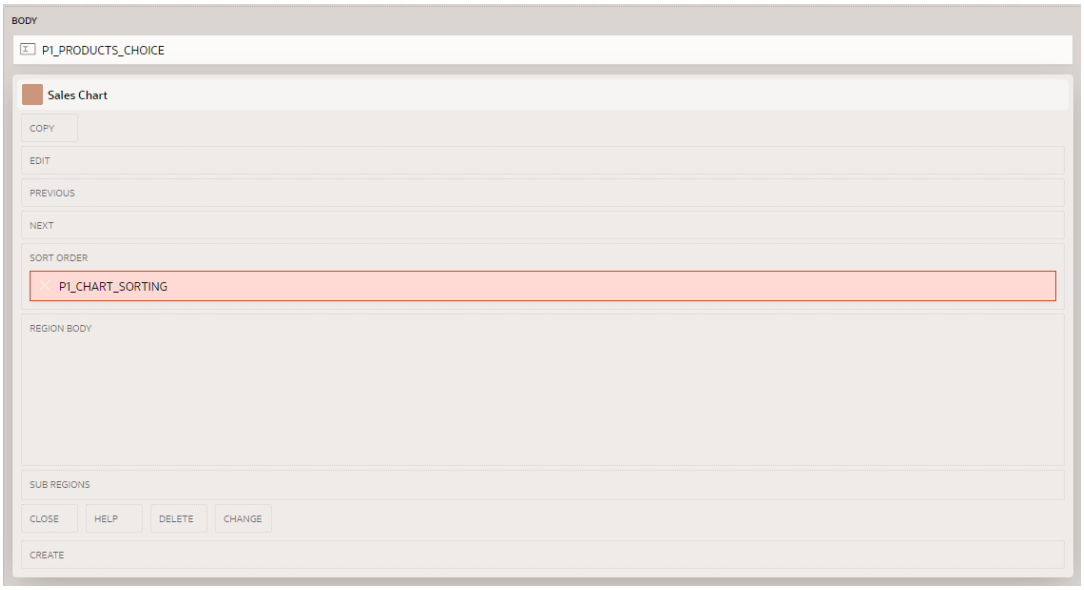
8.4 Dodawanie sortowania do wykresu słupkowego

Możemy poprawić wrażenia użytkownika, dodając możliwość sortowania wartości wyświetlanych na wykresie według różnych kryteriów. Aby wybrać kryteria sortowania, użyjemy elementu **Select List**, jak pokazano na rysunku 8.20.



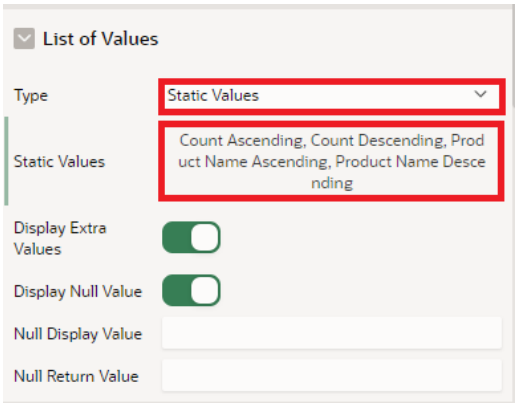
Rysunek 8.20: Wybieranie i umieszczanie elementu listy wyboru.

Umieścimy element **Select List** w naszym regionie **Sales Chart** w sekcji **"Sort Order"** i nazwiemy go **"P1_CHART_SORTING"**, jak pokazano na rysunku 8.21.



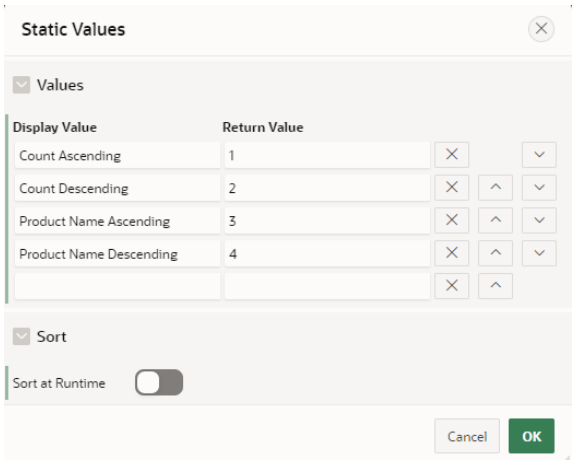
Rysunek 8.21: Region wykresu sprzedaży po umieszczeniu i zmianie nazwy elementu listy wyboru.

Teraz musimy zdefiniować listę wartości dla elementu Select List, który umieściliśmy. W ustawieniach elementu **"P1_CHART_SORTING"** musimy zmienić atrybut **"Type"** wewnątrz sekcji **"List of Values"** na **"Static Values"**. Następnie musimy nacisnąć przycisk pokazujący wartości domyślne **"Display1, Display2"** tuż obok atrybutu **"Static Values"**.



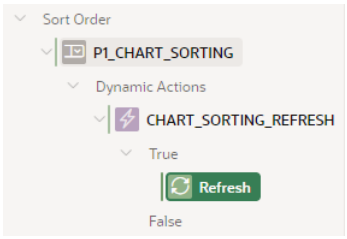
Rysunek 8.22: Ustawianie listy wartości dla pozycji Select List.

Teraz mogą Państwo zobaczyć, że otworzyło się nowe okno o tytule **“Static Values.”** W tym oknie musimy zdefiniować wartości wyświetlane i zwracane. W polu **display** values wpiszemy opisy sortowania, a w polu **return** values wpiszemy liczby zaczynające się od **1**, jak widać na rysunku 8.23. Jeśli chcemy, aby te wyświetlane wartości były uporządkowane tak, jak je wprowadziliśmy, od góry do dołu, musimy wyłączyć opcję **“Sort at Runtime”**.



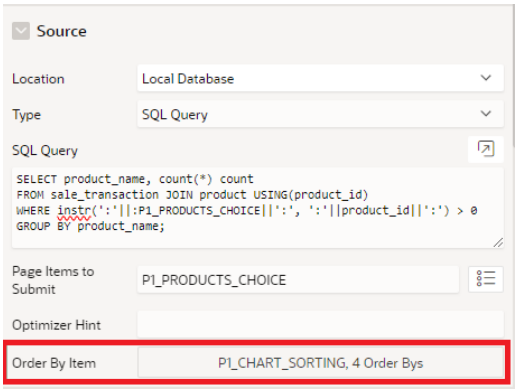
Rysunek 8.23: Ustawianie wartości statycznych dla elementu Select List.

Musimy dodać **Wykres sprzedaży** odświeżający **Akcja dynamiczna** w ten sam sposób, w jaki skonfigurowaliśmy go dla elementu **“P1_PRODUCTS_CHOICE”**. Wynik tego można zobaczyć na rysunku 8.24.



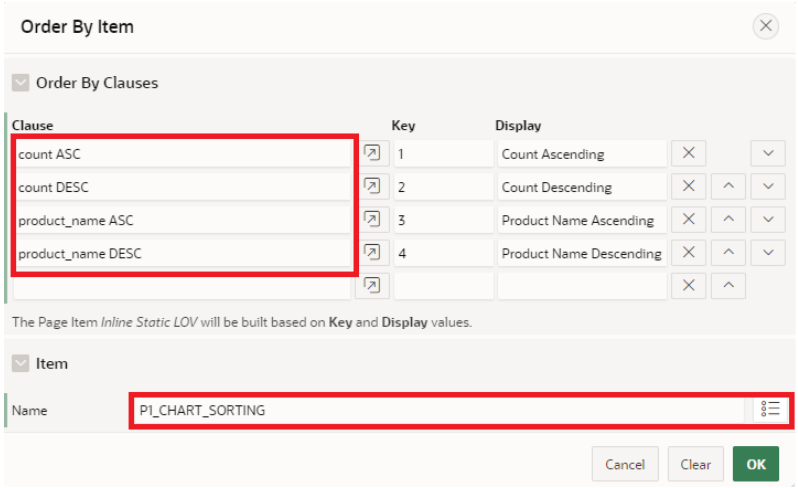
Rysunek 8.24: Dodawanie odświeżania akcji dynamicznej dla elementu Select List.

Teraz, gdy skonfigurowaliśmy element **Lista wyboru**, musimy połączyć go z regionem **Wykres sprzedaży**. Możemy to zrobić, naciskając przycisk tuż obok atrybutu **“Order By Item”**, który domyślnie mówi **“No Order By Item”**, jak zaznaczono na rysunku 8.25.



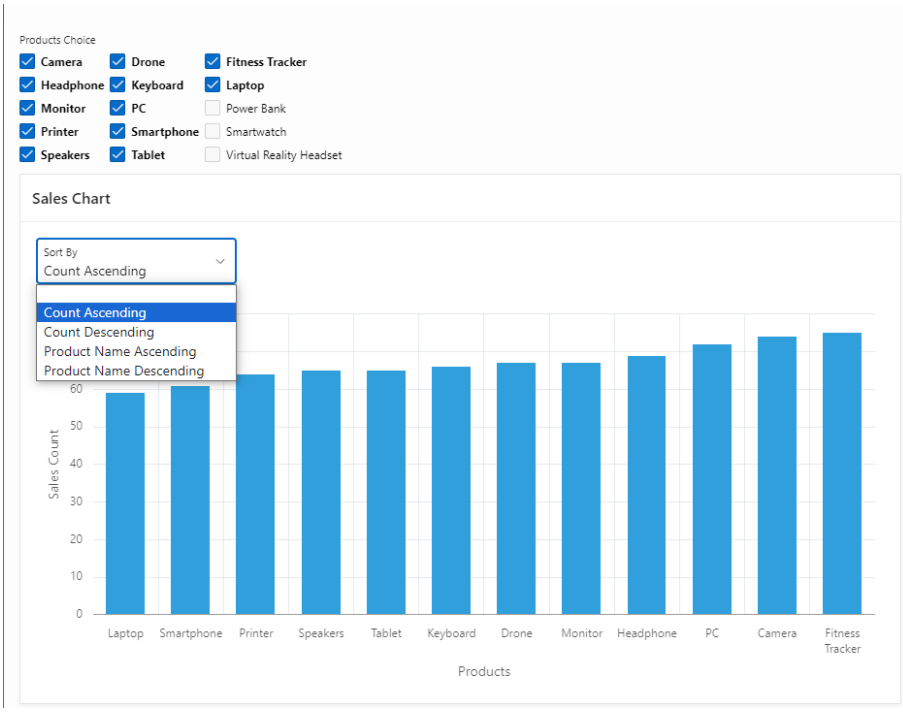
Rysunek 8.25: Ustawianie kolejności według pozycji dla regionu wykresu sprzedaży.

Po naciśnięciu przycisku powinni Państwo zobaczyć, że otworzyło się nowe okno z tytułem **“Order By Item.”**. W tym oknie musimy najpierw ustawić atrybut **“Name”** w sekcji **“Item”** na **“P1_CHART_SORTING”**. Następnie musimy wypełnić kolumnę “Clause”, jak widać na rysunku 8.26. “Clause” są praktycznie **“ORDER BY”** używanymi w instrukcjach **SELECT**, ale bez użycia słowa kluczowego **“ORDER BY”**.



Rysunek 8.26: Ustawienie klauzuli Order By dla elementu Select List używanego do sortowania.

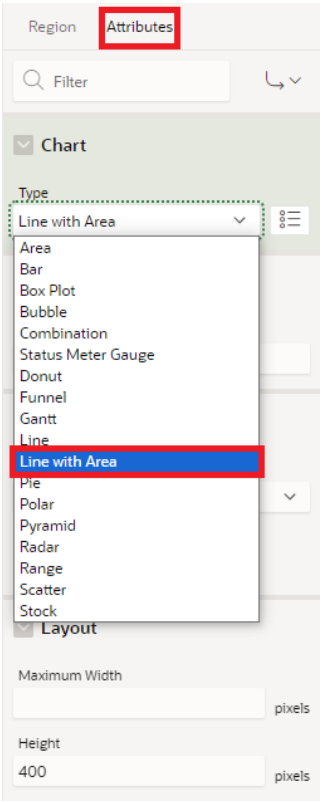
Jeśli dokładnie wykonali Państwo wszystkie powyższe kroki, wykresy **sortowanie** i **odświeżanie** powinny działać po zapisaniu strony i uruchomieniu aplikacji.



Rysunek 8.27: Aplikacja po wdrożeniu sortowania wykresów.

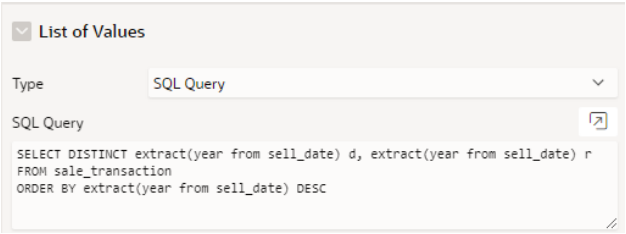
8.5 Tworzenie różnych typów wykresów

Jeśli chcą Państwo zaimplementować inny typ wykresu, można to zrobić, wybierając region wykresu i przechodząc do zakładki **“Attributes”** po prawej stronie **Page Designer**, a następnie zmieniając wartość atrybutu **“Type”**. Utworzymy jeszcze jeden wykres **Line with Area** przedstawiający miesięczną sprzedaż, który nazwiemy **“Monthly Sales Chart”** i będziemy mogli filtrować go według roku.



Rysunek 8.28: Zmiana typu wykresu na Liniowy z obszarem.

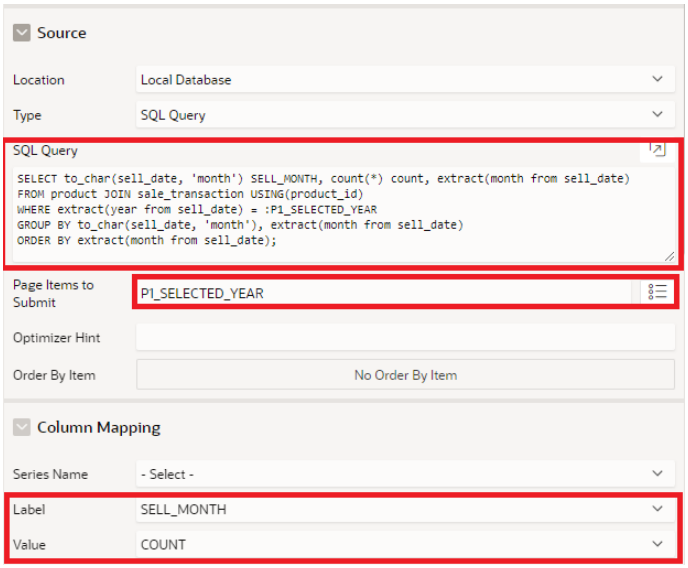
Następnym krokiem jest dodanie elementu **Select List**, którego użyjemy do **filtrowania** wykresu według **roku**. Element ten umieścimy w regionie **“Monthly Sales Chart”** w sekcji **“Edit”**. Zapytanie SQL używane do uzyskania listy wartości jest pokazane na rysunku 8.29.



Rysunek 8.29: Lista wartości dla elementu Select List używanego do filtrowania sprzedaży produktów według roku.

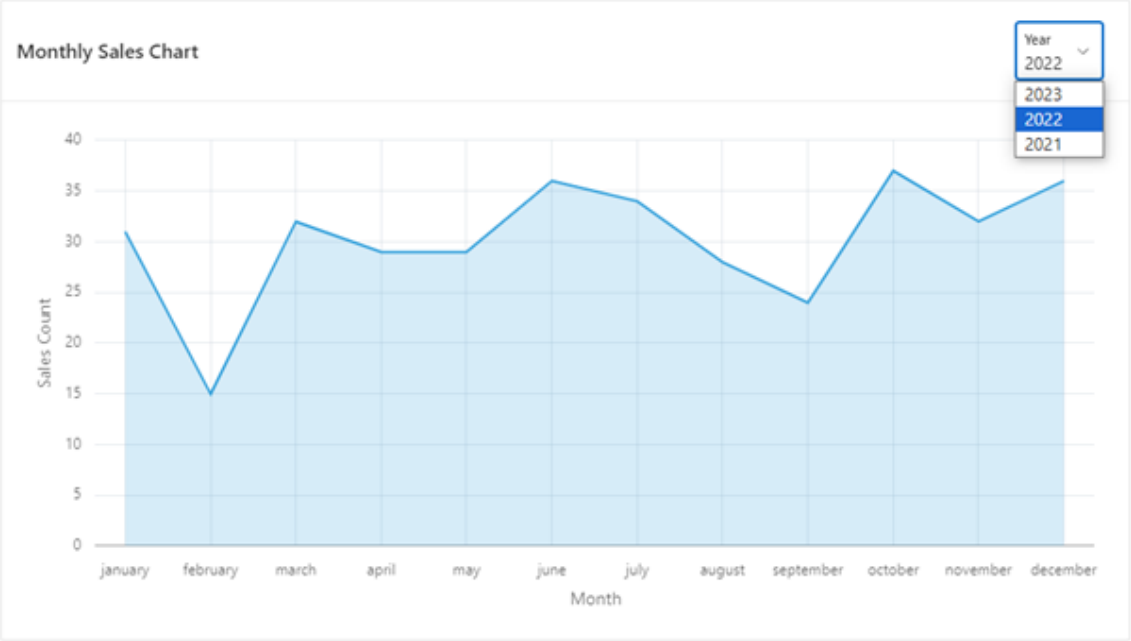
Po przygotowaniu elementu **Select List**, którego będziemy używać do filtrowania, możemy napisać źródłowe zapytanie SQL dla serii w regionie **“Monthly Sales Chart”**. Zapytanie SQL, którego użyjemy, pokazano na rysunku 8.30. Zamapujemy również **“Label”** na **“SELL_MONTH”** i **“Value”** na **“COUNT”**. Porządkujemy również wartości według rosnącego numeru miesiąca i

wyświetlamy pełną nazwę miesiąca na osi **X**. Ustawimy również etykietę osi **X** na **"Month"**, a osi **Y** na **"Sales Count"**.



Rysunek 8.30: Ustawienie źródłowego zapytania SQL i mapowania kolumn dla regionu wykresu miesięcznej sprzedaży.

Jeśli wszystko zostało poprawnie skonfigurowane, wykres powinien przypominać ten pokazany na rysunku 8.31.



Rysunek 8.31: Miesięczny wykres sprzedaży.

8.6 Pytania

1. Jaki jest główny cel regionów wykresów Oracle APEX?
2. Jakie typy wykresów można tworzyć za pomocą regionów wykresów Oracle APEX?

3. W jaki sposób można odświeżyć region wykresu w Oracle APEX, gdy wystąpią interakcje użytkownika, takie jak filtrowanie lub sortowanie?
4. Co robi funkcja INSTR i dlaczego jest używana w regionach wykresów Oracle APEX?

8.7 Odpowiedzi

1. Podstawowym celem regionów wykresów Oracle APEX jest wizualna reprezentacja danych, umożliwiająca użytkownikom uzyskanie wglądu i podejmowanie decyzji opartych na danych.
2. Regiony wykresów Oracle APEX obsługują różne typy wykresów, w tym wykresy słupkowe, liniowe, kołowe i inne.
3. Można skonfigurować akcję dynamiczną w celu odświeżenia regionu wykresu w Oracle APEX, gdy interakcje użytkownika zmieniają dane, takie jak filtrowanie lub sortowanie.
4. Funkcja INSTR służy do znajdowania indeksu pierwszego wystąpienia podciągu w ciągu znaków. W regionach wykresów Oracle APEX służy do sprawdzania, czy określone wartości spełniają wybrane kryteria, takie jak identyfikatory produktów na liście.

9. Jak zarządzać menu?

VERONIKA ŠALGOVÁ, MICHAL KVET, MIROSLAV POTOČÁR

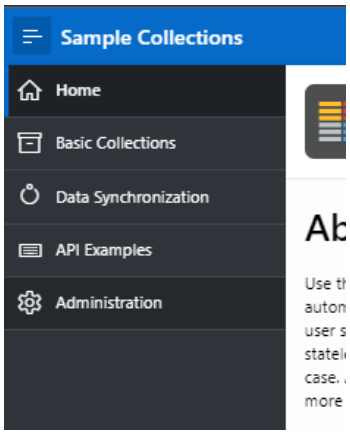
9.1 Jak zarządzać menu?

Menu nawigacyjne jest tworzone automatycznie po utworzeniu aplikacji. Składa się ono z pozycji listy odsyłających do stron aplikacji. Istnieje możliwość zmiany sposobu i miejsca wyświetlania menu nawigacyjnego poprzez edycję atrybutów menu nawigacyjnego na stronie interfejsu użytkownika. Typy menu nawigacji dostępne w Oracle APEX to:

- Menu boczne,
- Menu górne,
- Menu mega.

9.2 Menu boczne

Ten typ menu można rozwinąć lub zwinąć, klikając ikonę menu w nagłówku. Elementy nawigacyjne są wyświetlane za pomocą elementu drzewa. Elementy podrzędne można również rozwijać lub zwijać. Możemy wyświetlić pełne menu lub zwinąć je do wąskiego paska ikon.



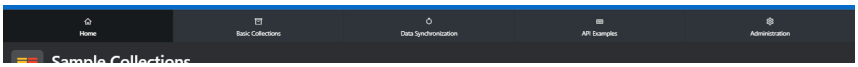
Rysunek 9.1: Rozszerzone boczne menu nawigacyjne

Możliwe jest również skonfigurowanie menu nawigacyjnego tak, aby było całkowicie ukryte,

co jest opcją domyślną. Wąski pasek ikon może być wyświetlany po zwinięciu za pomocą opcji szablonu.

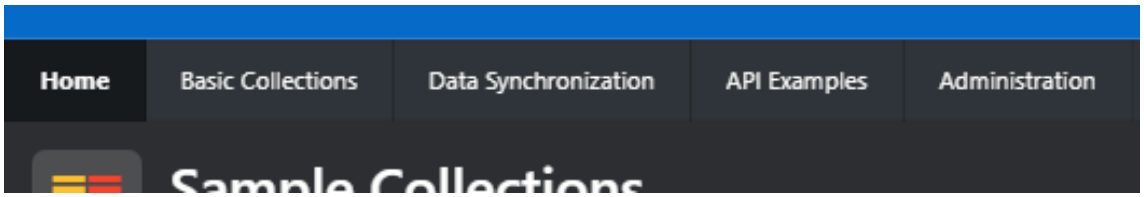
9.3 Górne menu

Top Navigation Menu jest wyświetlane w górnej części aplikacji. Oracle APEX udostępnia dwa szablony do wyświetlania menu, takie jak *Top Navigation Tabs* i *Top Navigation Menu*. Pierwszy typ jest odpowiedni dla prostych aplikacji z kilkoma zakładkami. Jest on pokazany na rysunku 9.2.



Rysunek 9.2: Górne zakładki nawigacji

Top Navigation Menu to pasek menu. Jest idealne dla bardziej złożonych aplikacji z kilkoma warstwami hierarchii w nawigacji. Jest ono pokazane na rysunku 9.3.

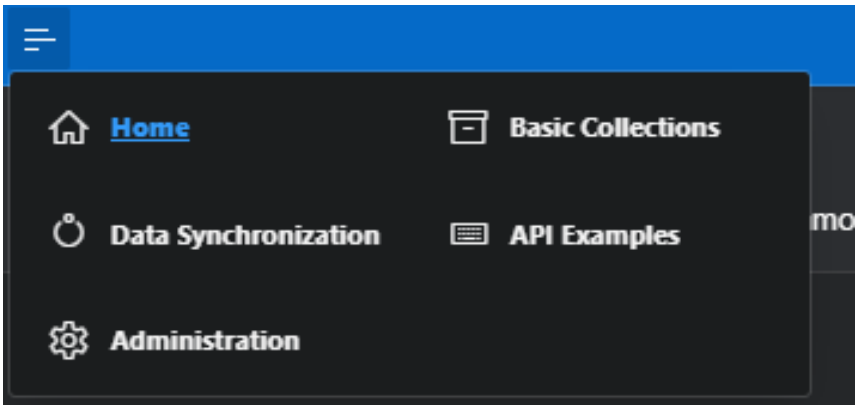


Rysunek 9.3: Górne menu nawigacyjne

Szablon Top Navigation Tabs automatycznie pozycjonuje się na dole ekranu dla małych ekranów lub urządzeń mobilnych. Top Navigation Menu przypomina większość aplikacji desktopowych.

9.4 Mega Menu

Ten typ menu jest wyświetlany jako wyskakujący panel, który można otworzyć lub zamknąć za pomocą ikony w nagłówku. Mega menu są przydatne do wyświetlania wszystkich elementów nawigacyjnych jednocześnie.

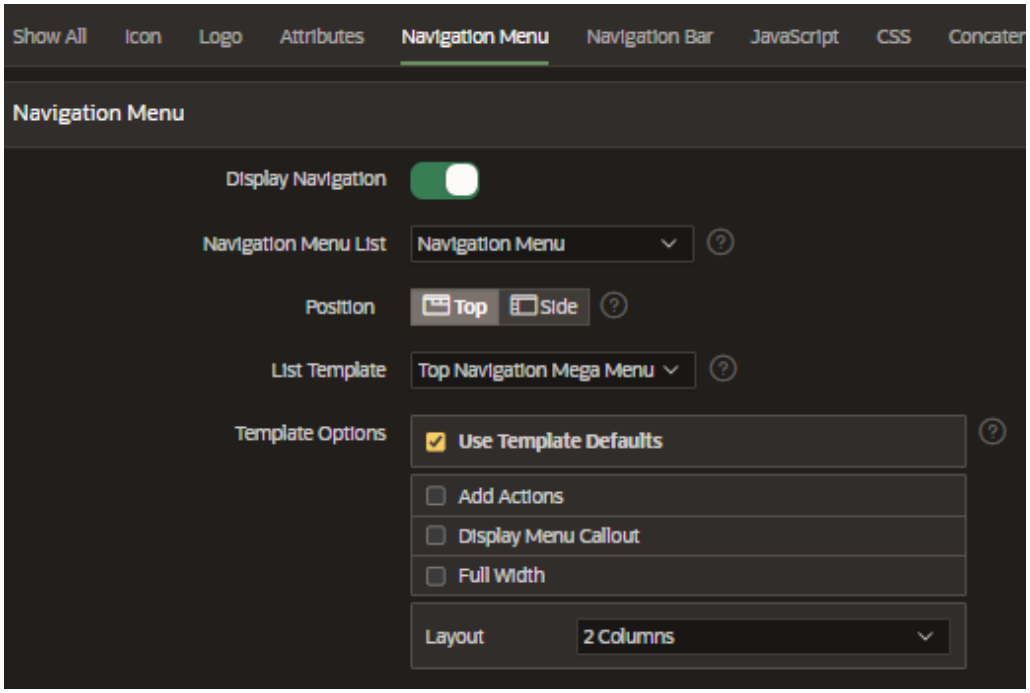


Rysunek 9.4: Mega menu górnej nawigacji

Ponadto możliwe jest wybranie liczby kolumn w menu.

9.5 Edycja list menu

Poszczególne pozycje menu tworzą listę, którą można tworzyć i edytować podobnie jak każdą inną listę. Dostęp do niej można uzyskać w *Shared Components* w regionie Navigation. Można wybrać opcję *Menu nawigacji* lub *Lista*. Można zmienić pozycję i sposób wyświetlania menu, edytując atrybuty menu nawigacji na stronie interfejsu użytkownika. Można się tam dostać klikając na *Shared Components* i *User Interface Attributes*. W zakładce *Menu nawigacji* można edytować atrybuty, które pokazano na rysunku 9.5.



Rysunek 9.5: Atrybuty menu nawigacji

- Wyświetl nawigację - aby włączyć lub wyłączyć nawigację.
- Navigation Menu List (Lista menu nawigacji) - służy do wyboru listy wykorzystywanej przez menu nawigacji.
- Position (Pozycja) - służy do wyboru pozycji *Top* lub *Side* menu nawigacyjnego.
- Szablon listy - aby wybrać spośród różnych szablonów do renderowania menu nawigacyjnego.
- Opcje szablonu - aby ustawić opcje szablonu używane dla listy menu nawigacyjnego.

Po kliknięciu przycisku *Zastosuj zmiany* można uruchomić aplikację i zobaczyć zmiany.

9.6 Pytania

1. Zazwyczaj menu używane w aplikacjach Oracle APEX znajduje się w lewej części ekranu. Do jakiego typu się ono odnosi?
2. Gdzie można znaleźć definicję pozycji menu do edycji?
3. Który typ menu umożliwia wyświetlenie wszystkich pozycji jednocześnie bez konieczności rozwijania podmenu?

9.7 Odpowiedzi

1. Menu boczne.

2. Poszczególne pozycje menu tworzą listę, którą można tworzyć i edytować podobnie jak każdą inną listę. Dostęp do niej można uzyskać w Shared Components w regionie Navigation.
3. Mega menu umożliwia wyświetlenie wszystkich elementów jednocześnie za pomocą jednego kliknięcia, bez konieczności rozwijania podmenu.



10. Jak współpracować w zespole?

P. STANISZEWSKI, M. SOŃTA, A. KIERZKOWSKI

10.1 Wspólne tworzenie wiedzy jest istotą rozwoju niskokodowego

"Współpraca wyróżnia nas jako ludzi nawet bardziej niż nasze przeciwstawne kciuki", ponieważ jesteśmy naturalnie zaprogramowani do współpracy jako preferencyjnego podejścia (patrz [1]; [3]), a platformy sieciowe są wyposażone w funkcje zachęcające do zbiorowego i kolektywnego dzielenia się wiedzą.

Korzyści płynące z inteligentnej współpracy na platformach cyfrowych są podwójne:

- aspekt produktywności, gdy jednostki wykorzystują zbiorową inteligencję do opracowania lepszego produktu i
- oraz aspekt motywacyjny, gdy proces aktywuje kreatywną energię i podejście społeczno-ściowe do rozwoju wiedzy.

10.2 Bycie razem online

APEX firmy Oracle to jedna z platform stworzonych do tworzenia sieci i interakcji podczas pracy nad aplikacjami, a takie podejście spełnia definicję platformy aktywizmu opartego na współpracy. Jednak ten rodzaj współpracy wymaga szczególnego sposobu bycia razem"([3]). Jedną z części doświadczenia współpracy jest możliwość przeglądania, komentowania i współtworzenia treści w celu wspólnego tworzenia wiedzy, a pobudzanie kreatywności i ducha poprzez kształtowanie relacji społecznych to kolejny wymiar wspólnego korzystania z historii APEX. Społeczność APEX Oracle organizuje ponad dziesięć regionalnych wydarzeń społecznościowych rocznie:

- APEX World, The Netherlands
- APEX World, APEX Alpe Adria, Austria/Croatia/Slovenia
- APEX World, APEX Connect, Germany
- APEX World, ODTUG Kscope, USA

osiągnęły etap rozwoju, aby rozwijać komercyjną platformę, która została obiecana na zawsze za darmo dla użytkowników w sposób oparty na współpracy open-source i ambicji podążania za kierunkami energetyzującego ducha społeczności.

Wraz z liczbą interakcji i projektów sposób rozwoju społeczności staje się jeszcze bardziej immersyjny, co stwarza możliwości "[...] sposoby bycia razem online, szczególnie w bardzo immersyjnych środowiskach, mogą mieć dodatkową, niezbadaną jeszcze warstwę poznawczą i percepcyjną"([3]) z ambicją osiągnięcia dojrzałości cyfrowego aktywizmu i zapraszania różnych

odbiorców niewykwalifikowanych w dziedzinie IT.

10.3 Znajomość technologii dzięki współpracy

Jeśli szukamy artykułów akademickich na temat APEX, większość z nich obejmuje przypadki użycia biznesowego, przypadki wysiłków międzywydziałowych w celu stworzenia aplikacji, która obejmuje zasoby ludzkie, społeczną odpowiedzialność biznesu, komunikację z różnymi interesariuszami i tym podobne.

Takie podejście, dostępność platformy i szeroki zakres treści edukacyjnych na temat APEX sprawiają, że narzędzie to jest również szansą na wejście w świat niskiego kodowania i praktykowania świadomości biznesowej na temat praktycznych zastosowań aplikacji. Dzięki takiemu podejściu, APEX otwiera możliwość współpracy przy projektach IT dla osób niezwiązanych z branżą IT i umożliwia podniesienie poziomu wiedzy technologicznej. Te kompetencje i postawy tworzą lepsze warunki zatrudnienia dla wszystkich interesariuszy zaangażowanych w proces tworzenia oprogramowania.

Sposoby współpracy są zwykle związane z metodologią projektu. Nawet jeśli platformy niskokodowe, takie jak APEX, są zaprojektowane tak, aby osiągać cele bardzo szybko, zazwyczaj potrzebujemy więcej niż jednego programisty, aby zrealizować projekt w założonym czasie i budżecie. W takiej sytuacji potrzebujemy odpowiedniego narzędzia, które pomoże programistom współpracować i sprostać wyzwaniom i możliwościom typowym dla coworkingu w projektach IT. W zależności od skali i wymagań naszych lub naszej organizacji, może to być związane z takimi aspektami jak:

- Standaryzacja - gdy chcemy zapewnić spójność wszystkich elementów aplikacji. Jest ona związana ze standardami UI, konwencjami nazewnictwa, technikami programowania, stylem kodowania i wszystkimi innymi standardami typowymi dla wybranej technologii. Na przykład w świecie APEX można po prostu pomyśleć o różnych sposobach nazywania elementów, procesów lub walidacji lub rozważyć standardy w wyborze numerów stron.
- Szkolenie - tworzenie aplikacji z innymi programistami pomaga w podnoszeniu umiejętności. Inni członkowie zespołu mogą być bardziej zaawansowani w aspektach, na których nam nie zależy.Nawet w przypadku korzystania z platformy niskokodowej, takiej jak APEX, która teoretycznie pozwala każdemu użytkownikowi być programistą "full-stack", możemy znaleźć specjalizacje. Na przykład, niektórzy ludzie czują się bardziej komfortowo z bazą danych, inni z czystym APEX-em, a jeszcze inni z funkcjami i językami frontendowymi. Posiadanie większego zespołu pomaga pozostałym rozwinąć umiejętności, których nie posiadali i zwiększyć ogólną jakość aplikacji.
- Czas - naturalne jest, że większa liczba programistów może dostarczać aplikacje szybciej, co przekłada się na lepszą wydajność zespołu lub działu odpowiedzialnego za dany system lub platformę.
- Jakość elementów - większy zespół może wdrożyć proces code review i sprawdzać jakość elementów dostarczanych przez członków zespołu. Zwiększa to jakość produktu końcowego i ogranicza liczbę błędów.
- Informacje zwrotne - współpraca oznacza również komunikację między programistami a użytkownikami. Korzystne jest posiadanie informacji od użytkowników na temat jakości aplikacji, zgłoszonych błędów lub żądań zmian.

Oracle APEX, jako dojrzała platforma, obejmuje różne strategie współpracy z wbudowanymi funkcjami, ale także z opcjami rozszerzenia o zewnętrzne narzędzia, takie jak pakiety współpracy, aplikacje do zarządzania zadaniami lub narzędzia do weryfikacji kodu.

Nawet jeśli dopiero zaczynamy naszą przygodę z APEX, ważne jest, aby wiedzieć, w jaki sposób możemy współpracować z innymi użytkownikami za pomocą natywnych funkcji, aby taka praca była bardziej produktywna i mieć pewność, jak rozwiązać typowe problemy, takie jak przydzielanie zadań, praca nad konkretnymi stronami / funkcjami lub kontrolowanie statusu

zaplanowanej pracy.

Najważniejszą częścią na samym początku jest komunikacja. Jeśli zespół składa się z więcej niż jednej osoby, należy mieć sposoby na planowanie pracy na kolejne dni, informowanie o statusie oraz kontrolowanie czasu i terminów. Nie ma nic ważniejszego niż regularne spotkania i komunikacja, jeśli coś jest niejasne. Warto pomyśleć o codziennych spotkaniach, cotygodniowym planowaniu lub innych regularnych wydarzeniach. Można również wdrożyć jedną ze znanych metodologii, takich jak Scrum, aby mieć takie nawyki zapisane w jednym miejscu. Proszę po prostu zacząć to robić, a znajdą Państwo najlepszy sposób współpracy dla swojego zespołu.

Na początku, myśląc o narzędziach do współpracy, można rozważyć podstawowe arkusze kalkulacyjne do monitorowania, kto co robi. Oczywiście na rynku dostępnych jest wiele dedykowanych aplikacji do zarządzania zadaniami, takich jak Jira, Monday, ClickUp, Redmine itp. i oczywiście wraz ze wzrostem zespołu i skali można rozważyć skorzystanie z jednej z nich. Jednak większość z tych działań można wykonać za pomocą funkcji zawartych w platformie Oracle Application Express..

10.4 Opis funkcji

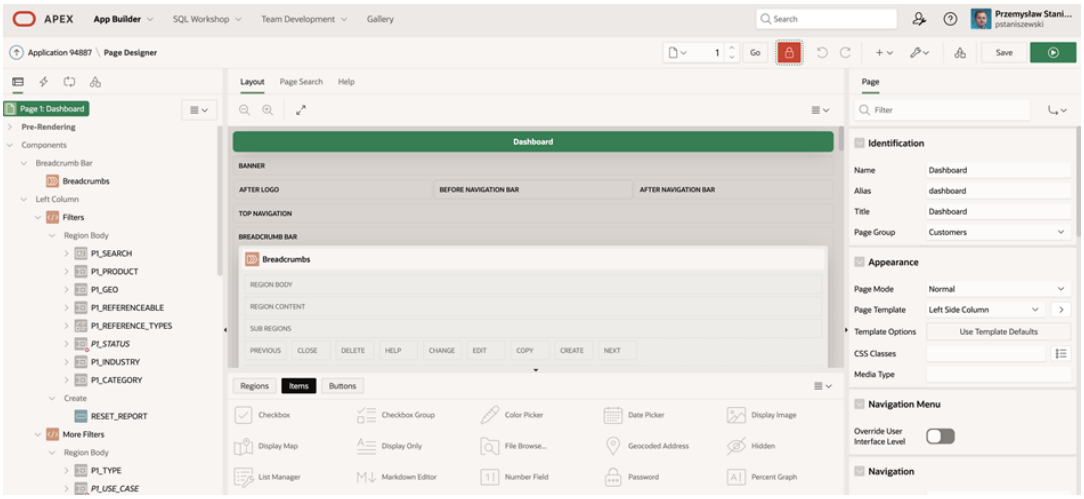
10.4.1 Blokada strony

Blokowanie stron to przydatna funkcja, która zapobiega jednoczesnej edycji tej samej strony przez wielu deweloperów. Gdy deweloper edytuje stronę, funkcja blokowania stron uniemożliwia innym deweloperom wprowadzanie zmian na tej samej stronie, dopóki pierwszy deweloper nie zakończy pracy i nie zwolni strony. Pomaga to zapobiegać konfliktom i zapewnia, że zmiany są wprowadzane w sposób uporządkowany i kontrolowany (patrz rys. 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		

Rysunek 10.1: Funkcja blokowania stron w APEX.

Można zablokować określoną stronę, klikając ikonę blokady na listach stron w kreatorze aplikacji lub zrobić to na poziomie widoku strony (patrz rysunek 10.2).

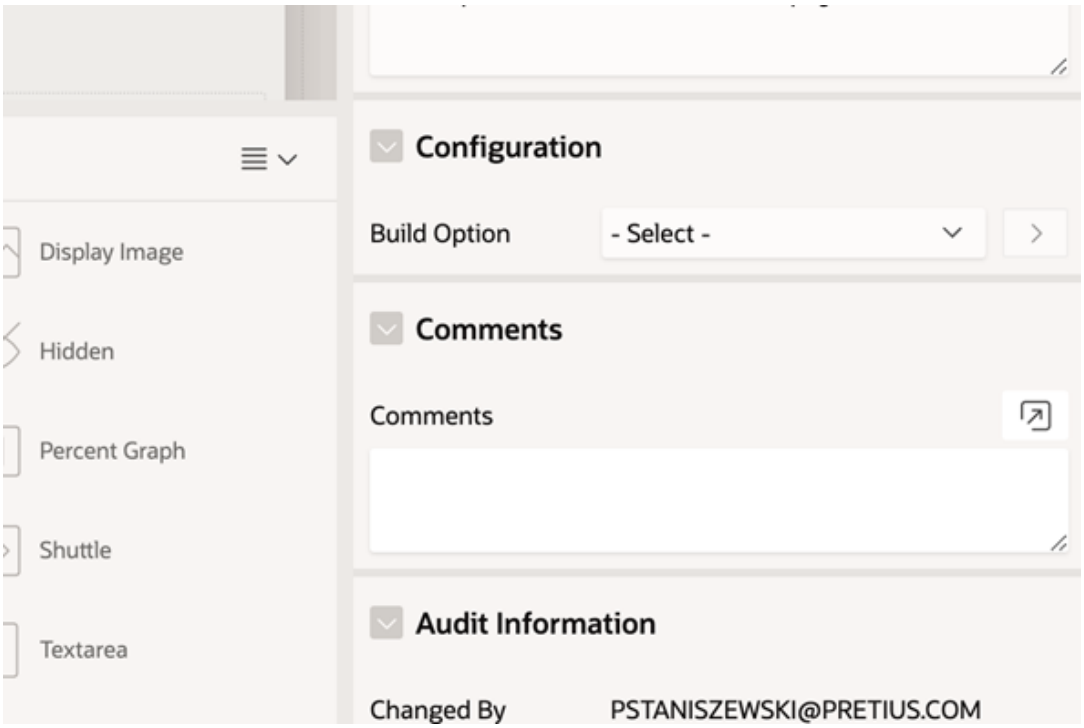


Rysunek 10.2: Blokowanie określonej strony.

Podczas blokowania lub odblokowywania strony można dodać komentarz, dzięki czemu Państwo lub inni deweloperzy będą mogli zobaczyć więcej informacji na temat przyczyny lub postępu prac.

10.4.2 Komentarze

Oracle APEX umożliwia programistom dodawanie komentarzy do stron, regionów, elementów i wielu innych elementów aplikacji. Komentarze mogą być używane w celu zapewnienia kontekstu dla innych programistów, wyjaśnienia powodów decyzji projektowych lub po prostu udokumentowania zmian wprowadzonych w aplikacji. Korzystając z komentarzy, programiści mogą lepiej komunikować się ze sobą i upewnić się, że wszyscy są na tej samej stronie (patrz rysunek 10.3)

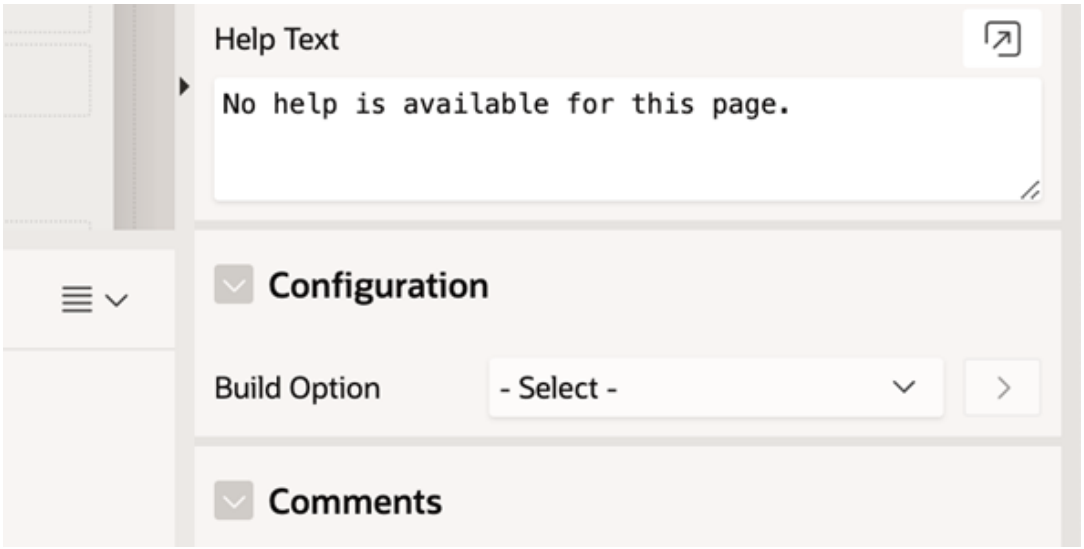


Rysunek 10.3: Dodawanie komentarzy.

Dokumentowanie kodu dla innych członków zespołu lub przyszłych programistów, którzy zajmą się rozwojem lub utrzymaniem aplikacji, jest ważne. Nawet jeśli korzystanie z platformy niskokodowej pozwala na samodzielne zdefiniowanie sposobu wykonania określonej funkcji, dodanie pewnych informacji jest nadal dobrą praktyką, jeśli określony element jest tworzony w sposób unikalny.

10.4.3 Opcje budowy

Opcje kompilacji to kolejna funkcja, która może pomóc zespołom w bardziej efektywnej współpracy w Oracle APEX. Opcje kompilacji umożliwiają programistom określenie różnych ustawień kompilacji dla różnych środowisk, takich jak programowanie, testowanie i produkcja. Ułatwia to zarządzanie zmianami w różnych środowiskach i zapewnia, że prawidłowe ustawienia są stosowane podczas wdrażania aplikacji (patrz rysunek 10.4).



Rysunek 10.4: Opcje budowania.

Można je traktować jako informację, czy dana funkcja (strona, element, proces) powinna zostać uwzględniona w kolejnym wydaniu, które zostanie wyeksportowane/zaimportowane do innego środowiska. Podczas eksportowania/importowania aplikacji można wybrać, czy dany "build"ma zostać uwzględniony. Proszę sobie wyobrazić sytuację, w której jeden z deweloperów pracuje nad jakąś funkcją i nie może zdążyć przed upływem terminu. Zamiast usuwać tę stronę z aplikacji, można po prostu utworzyć dla niej dedykowaną "kompilację"i wykluczyć ją podczas eksportu aplikacji.

Korzystanie z tej funkcji może być pierwszym krokiem do zdefiniowania procesu CI/CD dla Państwa aplikacji, który jest niezbędny w profesjonalnym tworzeniu oprogramowania.

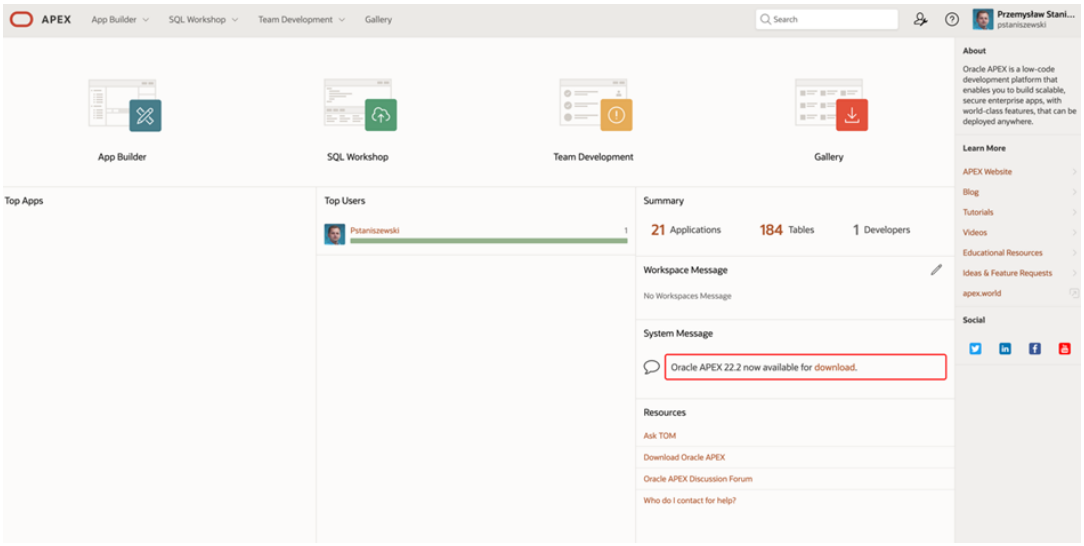
10.4.4 Rozwój zespołu

Funkcja Oracle APEX “Team Development” umożliwia zespołom wspólną pracę nad tą samą aplikacją w kontrolowanym i opartym na współpracy środowisku. Dzięki tej funkcji członkowie zespołu mogą zarządzać projektami, w szczególności :

- Definiować kamienie milowe i etapy projektu;
- Tworzyć, przypisywać i monitorować zadania/problemy;
- Stosować klasyfikacje przy użyciu grup etykiet i etykiet;
- Tworzyć szablony spraw, aby tworzenie zadań lub raportowanie było spójne;
- Łączyć Problemy z Aplikacją i Stroną;

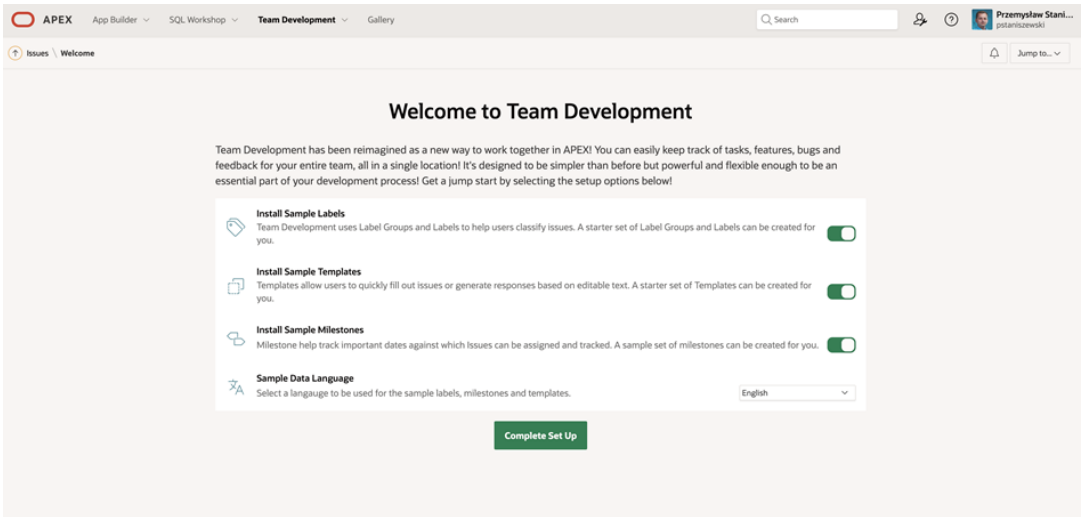
- Komentować i dyskutować istniejące problemy. Zmieniać ich status i monitorować postęp etapu projektu;
- Subskrybować różne rodzaje problemów;
- Monitorować informacje zwrotne dostarczane przez użytkowników aplikacji i decydować, co z nimi zrobić.

Team Development jest jedną z czterech głównych sekcji dostępnych w APEX IDE. Jej dostępność zależy od posiadanej wersji APEX. Został wprowadzony w APEX 19.2, ale w najnowszych wersjach nie jest dostępny od razu, wymaga dodatkowych działań, aby go udostępnić (zob.rysunek 10.5).



Rysunek 10.5: Funkcja “Team Development”

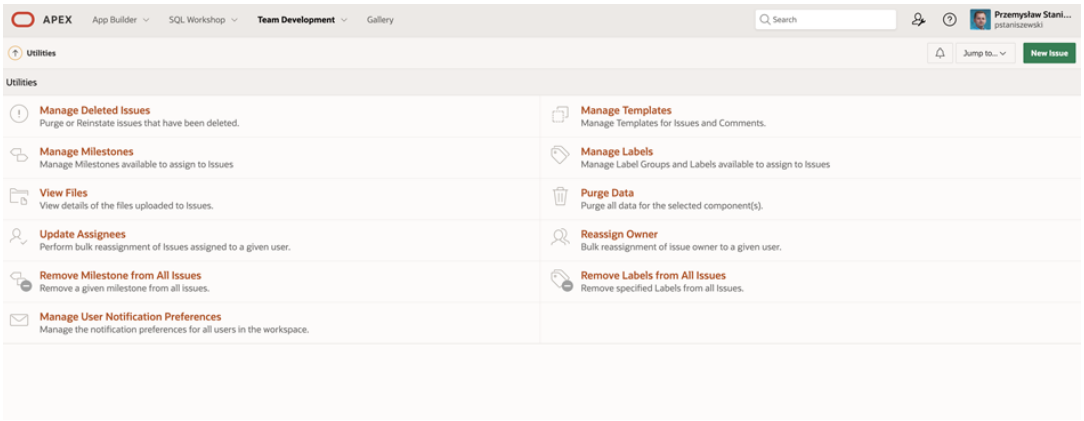
Aby go włączyć, musisz wybrać opcję "Team Development"w głównym menu. Następnie zostaniesz przeniesiony na dedykowaną stronę instalacji. Jeśli chcesz, możesz zawierać przykłady Zadań, Kamieni Milowych, Szablonów i Etykiet, ale możesz także pominąć ten krok i aktywować funkcję bez nich (zob.rysunek 10.6).



Rysunek 10.6: Aktywacja funkcji “Team Development”.

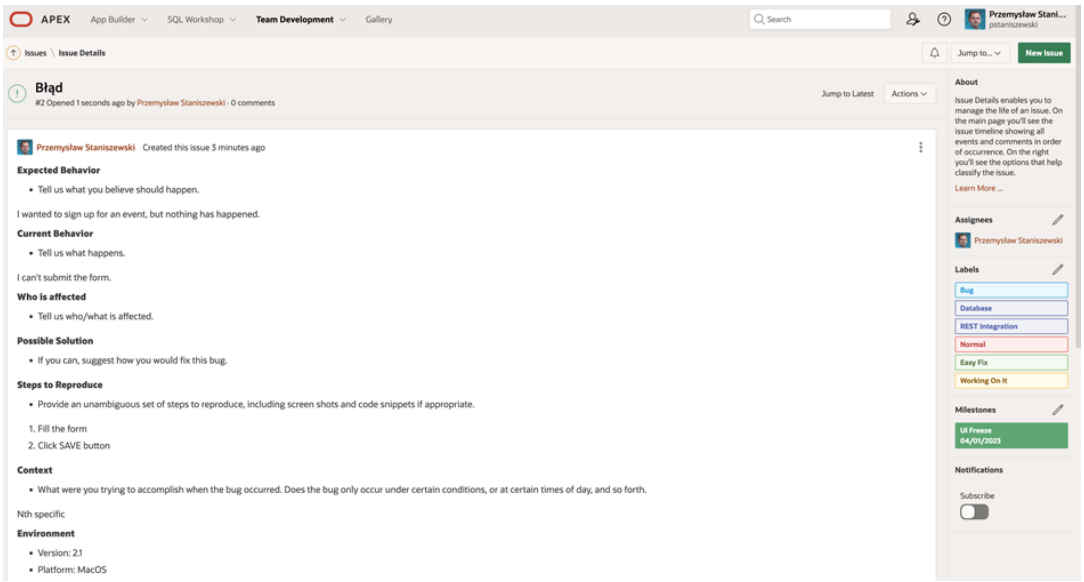
Jeśli twój zespół chciałby zacząć korzystać z funkcji "Team Development", powinieneś najpierw

skonfigurować jego słownik zgodnie ze swoim stylem pracy. Następnie przejdź do sekcji Narzędzia i zdefiniuj typy problemów, etapy projektu i kamienie milowe lub szablony do typowej komunikacji w projekcie (zob.rysunek 10.7).



Rysunek 10.7: Dostosowanie funkcji rozwoju zespołu.

Najważniejszą częścią tej funkcji jest sekcja “Issues”, gdzie można znaleźć wszystkie zgłoszone problemy, kategoryzować je i przypisywać konkretnym osobom. Każdy problem może być przypisany do więcej niż jednego dewelopera, co umożliwia dodawanie komentarzy, dyskusję w jego kontekście oraz zmianę jego statusu. W razie potrzeby użytkownicy mogą również przesyłać pliki i praktycznie prowadzić całą komunikację związana z problemem w ramach "Team Development". Ważne jest również to, że wszystkie problemy można połączyć z konkretną aplikacją i stroną, dzięki czemu łatwo rozpocząć dochodzenie i monitorować, które problemy są związane z konkretną częścią aplikacji. Status problemu to prosty rodzaj workflow problemu, który można znaleźć w wielu systemach zarządzania zadaniami - użytkownicy mogą zgłaszać problemy, oznaczać je jako "w toku", wracać do zgłaszającego i zakończyć informacją o sukcesie lub innych powodach zamknięcia. (zob.rysunek 10.8).



Rysunek 10.8: Komunikacja związana z problemem w ramach funkcji “Team Development”.

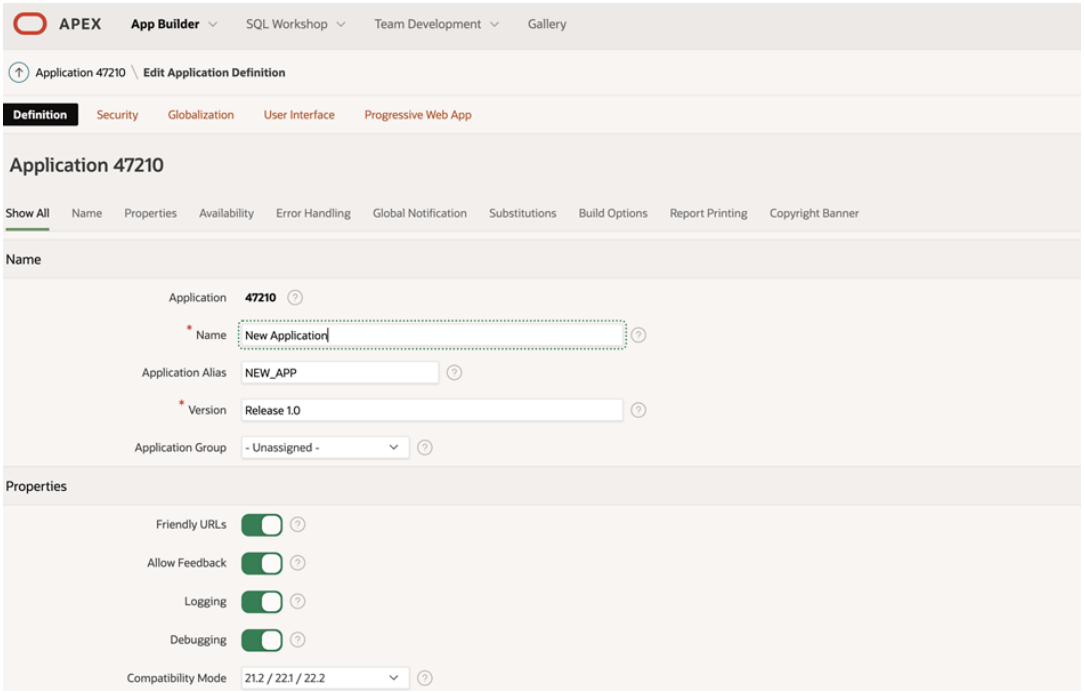
Deweloperzy mogą również subskrybować problemy przypisane do nich, co pozwala na otrzymywanie powiadomień e-mailowych przy każdej akcji na poziomie problemu. Aplikacja umożliwia

użytkownikom ograniczenie liczby e-maili, wysyłając je w formie podsumowań godzinnych lub dziennych. Zdecydowanie polecamy rozpoczęcie korzystania z "Team Development, jeśli twój zespół nie ma wdrożonego innego systemu zarządzania zadaniami, lub korzysta z bardzo podstawowych narzędzi, takich jak arkusze kalkulacyjne, do zarządzania projektem.

10.4.5 Informacje zwrotne

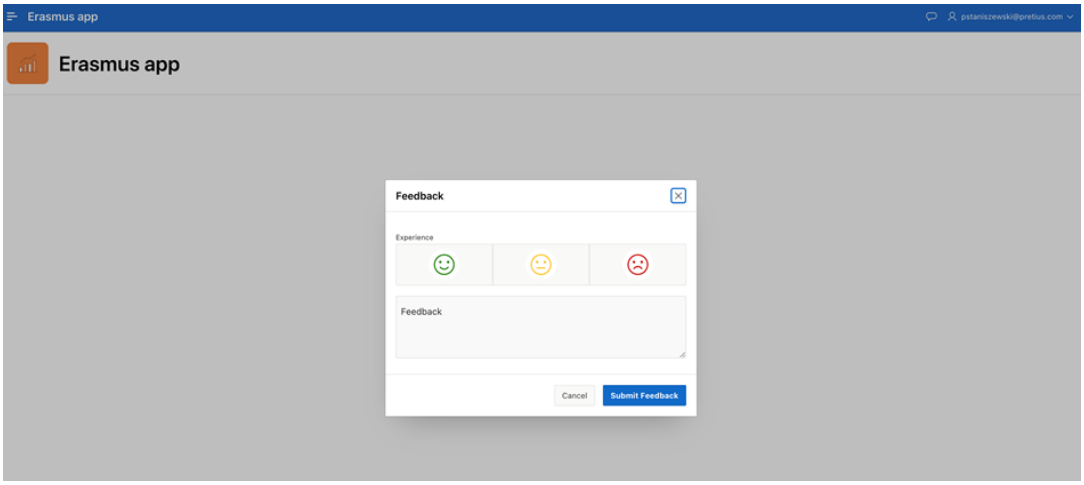
Strony z informacjami zwrotnymi są wbudowaną funkcją w Oracle APEX, która umożliwia użytkownikom końcowym dostarczanie opinii na temat funkcjonalności i wydajności aplikacji. Strony te i można dostosować do potrzeb aplikacji i mogą zawierać różne rodzaje informacji zwrotnych, takich jak zgłoszenia błędów, prośby o dodanie nowych funkcji oraz ogólna opinia.

To kluczowe, aby uzyskać informacje od użytkowników końcowych i traktować ich jak rozszerzenie zespołu deweloperskiego. Im bliżej współpracujemy nie tylko wewnętrznie, ale także z rzeczywistymi użytkownikami aplikacji, tym lepsza jakość aplikacji, którą uzyskamy. To także pozwala zespołowi dostosować aplikację do rzeczywistych potrzeb i pomaga budować zaufanie między zespołem deweloperskim a użytkownikami (zob.rysunek 10.9).



Rysunek 10.9: Zezwolenie na informacje zwrotne w “Application Definition”.

Aby ją włączyć, musisz wybrać odpowiednią opcję podczas tworzenia aplikacji. Jeśli aplikacja już istnieje, przejdź do Ędycji Definicji Aplikacji"i zaznacz opcję Źezwalaj na Informacje Zwrotne"w sekcji Właściwości. Po aktywacji w menu aplikacji zostanie dodana dedykowana ikona. Po jej kliknięciu użytkownik końcowy może przesłać informacje zwrotne (zob.rysunek 10.10).



Rysunek 10.10: Przesyłanie informacji zwrotnych w aplikacji.

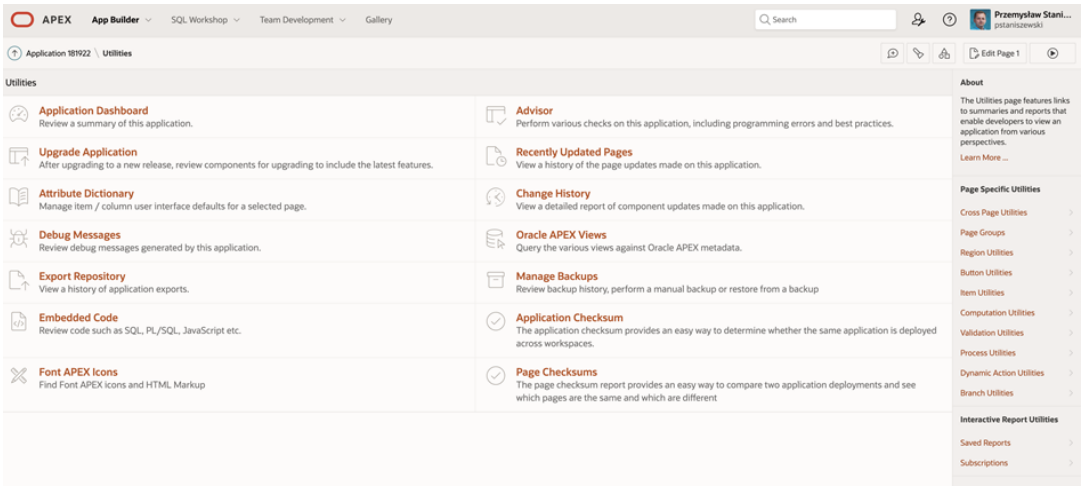
Wygląd strony informacji zwrotnych można dostosować, ponieważ jest to standardowa strona APEX. Wszystkie informacje dostarczone przez użytkownika końcowego są dostępne w sekcji "Team Development"i mogą być analizowane przez zespół (możliwe jest również śledzenie ich na poziomie konkretnej aplikacji w sekcji Ądministration"). Ponadto developerzy otrzymują szczegółowe informacje związane z raportowanymi informacjami zwrotnymi (ID aplikacji, numer strony, dane użytkownika, informacje o jego środowisku itp.) i mogą przekształcać je w problem.

10.4.6 Standaryzacja

Podczas tworzenia aplikacji w zespole bardzo ważne jest również zachowanie spójności między standardami rozwoju różnych członków zespołu. Tak jak w każdym aspekcie współpracy komunikacja jest kluczem do sukcesu - na początku projektu zespół powinien uzgodnić sposób rozwoju i wykorzystania komponentów APEX. Do najważniejszych aspektów należą:

- Konwencja nazewnictwa (dla elementów, procesów, walidacji, akcji dynamicznych, itp.);
- Numerowanie i nazewnictwo stron (alias);
- Dozwolone elementy (np. w słownikach używamy tylko interaktywnych siatek, każdy formularz musi być wyświetlany jako strona modalna);
- Standardy UX/UI (np. wygląd typowych stron, polityka wyrównywania różnych danych czasowych, spójne formaty danych, zalecane ikony dla konkretnych działań, itp.);
- Spójność danych (np. definiowanie wspólnych słowników w jednym miejscu jako współdzielone komponenty).

APEX dostarcza wiele rozwiązań, które pomagają zespołom w zarządzaniu standaryzacją. Większość z nich znajduje się w sekcji "Utilities ńa poziomie wybranej aplikacji (zob.rysunek 10.11).



Rysunek 10.11: Menu "Utilities"(Narzedzia).

Najważniejsze na początku rozwoju to:

- Raporty wielostronicowe (narzędzia specyficzne dla strony) - umożliwiają programistom porównywanie różnych elementów APEX (atrybuty strony, raporty, przyciski, pozycje itp.) w aplikacji w celu sprawdzania spójności i aktualizowania ich jednocześnie bez ręcznej zmiany atrybutów w wielu miejscach.
- Zmień historię/Ostatnio aktualizowane strony - pozwala na śledzenie działań podjętych przez programistów.
- Definicja domyślna (Słownik atrybutów) - umożliwia utworzenie wspólnego słownika w celu zachowania spójnego nazewnictwa, formatowania, typu danych itp. dla elementów powtarzających się w wielu miejscach w aplikacji. Na przykład "Data aktualizacji"może istnieć w różnych raportach lub elementach aplikacji, ale zawsze chcemy mieć tę samą etykietę i format daty dla każdego wystąpienia

Oczywiście APEX zapewnia wiele dodatkowych funkcji i strategii, aby zachować spójność na poziomie bazy danych i aplikacji. Osoby, które chcą poszerzyć swoją wiedzę, powinny zapoznać się z dokumentacją APEX i zapoznać się z bardziej zaawansowanymi opcjami, takimi jak strona globalna, schematy, ustawienia domyślne interfejsu użytkownika, aplikacja motywu głównego i mechanizm subskrypcji, widoki APEX lub wtyczki.

10.5 Wnioski

Jako dojrzała platforma programistyczna, Oracle APEX zapewnia wiele funkcji, które ułatwiają współpracę. Będąc developerami niskokodowymi, możemy ich używać, profesjonalnie tworzyć i utrzymywać aplikacje oparte na APEX bez rozszerzania platformy lub integrowania jej z zewnętrznymi narzędziami. Jeśli jesteś nowym użytkownikiem APEX i nie jesteś jeszcze zaznajomiony z strategiami współpracy w programowaniu, zacznij od najbardziej fundamentalnych kroków opisanych w tym rozdziale:

- Komunikacja i regularne spotkania z zespołem (na żywo lub zdalnie - użyj narzędzi takich jak Google Meet, Zoom lub MS Teams wideokonferencji i rozmów głosowych oraz Slack, Google Chat lub MS Teams do szybkiego przesyłania wiadomości);
- Zacznij korzystać z narzędzi współpracy, np. APEX Team Development, w celu planowania rozwoju i weryfikacji postępów;
- Zacznij dokumentować swój kod i standardy. I zweryfikuj je!

Współpraca w programowaniu niskokodowym może czasami wydawać się przesadzona, ale warto spróbować, ponieważ przynosi tyle samo korzyści, ile w standardowym programowaniu. Co więcej, większość typowych strategii współpracy, takich jak "przegląd kodu", "programowanie w

parachłub "programowanie w grupie"można również stosować w platformie niskiego kodu, nawet jeśli sprawdzony element nie jest czystym kodem, ale procesem, stroną lub konfiguracją.

W zależności od skali projektu lub standardów przyjętych w organizacji, w pewnym momencie może być konieczne rozszerzenie procesu współpracy. Na przykład w dużych projektach będziesz musiał zacząć używać repozytoriów kodów, narzędzi do automatycznego budowania i wdrażania, lub oprogramowania dedykowanego do testowania i weryfikacji jakości kodu. Wykorzystanie elementów profesjonalnych procesów CI/CD, takich jak Git, Docker, Flyway, Liquibase, Jenkins, Gradle, SonarQube, Selenium, Cypress i wiele innych, jest również możliwe w przypadku aplikacji opartych na APEX. Jednak w wielu sytuacjach elementy zawarte w platformie są wystarczające do stworzenia rewolucyjnej aplikacji przy współpracy z zespołem deweloperów.

10.6 Pytania

1. Czym jest funkcja blokowania stron i dlaczego ma to znaczenie?
2. Jak programiści mogą komunikować się podczas pracy nad projektem za pomocą platformy APEX?
3. Jak członkowie zespołu mogą korzystać, używając funkcji Team Development?

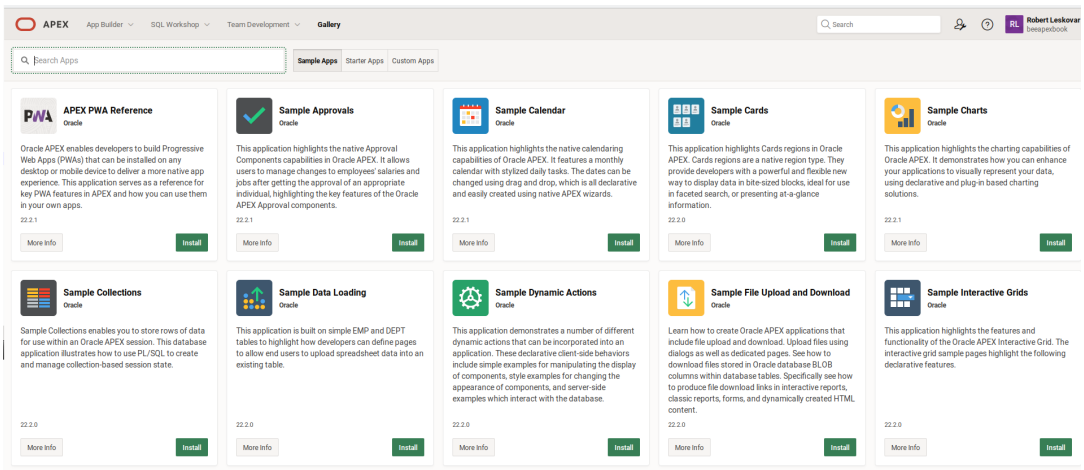
10.7 Odpowiedzi

1. Blokowanie stron jest przydatną funkcją, która uniemożliwia wielu programistom jednoczesne edytowanie tej samej strony przez co zapobiega konfliktom.
2. Programiści mogą dodawać komentarze do stron, regionów, elementów i wielu innych elementów aplikacji. Komentarze mogą być używane do dostarczania kontekstu innym programistom, wyjaśniania przyczyn decyzji projektowej lub po prostu dokumentowania zmian wprowadzonych w aplikacji. Korzystając z komentarzy, developerzy mogą lepiej komunikować się między sobą i zapewnić ogólną jakość oprogramowania.
3. Funkcja Team Development umożliwia członkom zespołu zarządzanie projektami rozwojowymi, w szczególności: definiowanie kamieni milowych i etapów projektu, tworzenie, przydzielanie i monitorowanie zadań/problemów, stosowanie klasyfikacji za pomocą grup etykiet i etykiet, tworzenie szablonów problemów, aby ujednolicić tworzenie lub raportowanie zadań. Łączenie problemów z aplikacją i stroną, komentowanie i omawianie istniejących problemów, zmienianie ich statusu i monitorowanie postępu etapu projektu oraz subskrybowanie różnych rodzajów problemów.

11. Jak skorzystać z galerii aplikacji i wtyczek?

VJERAN STRAHONJA, DIJANA OREŠKI

W tym rozdziale omówiono aplikacje początkowe, przykładowe aplikacje i wtyczki w systemie Oracle APEX. Wszystkie obszary robocze Oracle APEX są wyposażone w przykładowe aplikacje i startowe aplikacje (zob.rysunek 11.1) oraz wtyczki.



Rysunek 11.1: Obszary robocze Oracle APEX są dostarczane z przykładowymi aplikacjami i aplikacjami startowymi.

Aplikacje początkowe to autonomiczne rozwiązania punktowe, które spełniają proste wymagania i nie wymagają złożonych rozwiązań. Z drugiej strony przykładowe aplikacje koncentrują się na konkretnej funkcjonalności i zawierają wskazówki dotyczące korzystania z określonej funkcji. Wtyczki umożliwiają użytkownikom rozszerzanie aplikacji o funkcje niestandardowe.

11.1 Jak zainstalować przykładowe i startowe aplikacje?

Użytkownicy mogą pobrać przykładowe i startowe aplikacje z galerii. Jeśli jest to włączone przez administratora instancji, aplikacje można zainstalować za pomocą galerii.

Aby **zainstalować** aplikacje z galerii należy:

1. Przejść na stronę startową obszaru roboczego.

2. Wybrać opcje **"Gallery"**. Podczas pierwszego ładowania strony **"Gallery"** wyświetlane są przykładowe aplikacje, aplikacje startowe i aplikacje niestandardowe za pomocą kart.
3. Wybrać **Śample Apps"** albo **Śtarter Apps"**. Karty służą do wyszczególniania aplikacji. Jeśli program nie został wcześniej zainstalowany, każda karta zawiera przycisk “Install”. Jeśli aplikacja została już zainstalowana, będą dostępne ikony Żun"i Żemove App".
4. Aby przeprowadzić instalacje wybierz opcje **"Install App"** .
5. Rozwiń sekcje Ądvanced Settings". (nieobowiązkowe)
   a. Należy określić, czy ID aplikacji zostanie przydzielony ręcznie czy automatycznie (ustawienie domyślne).
   b. Zmienić schemat analizy składniowej.
6. Aby zakończyć instalację, proszę nacisnąć przycisk "Install Application".

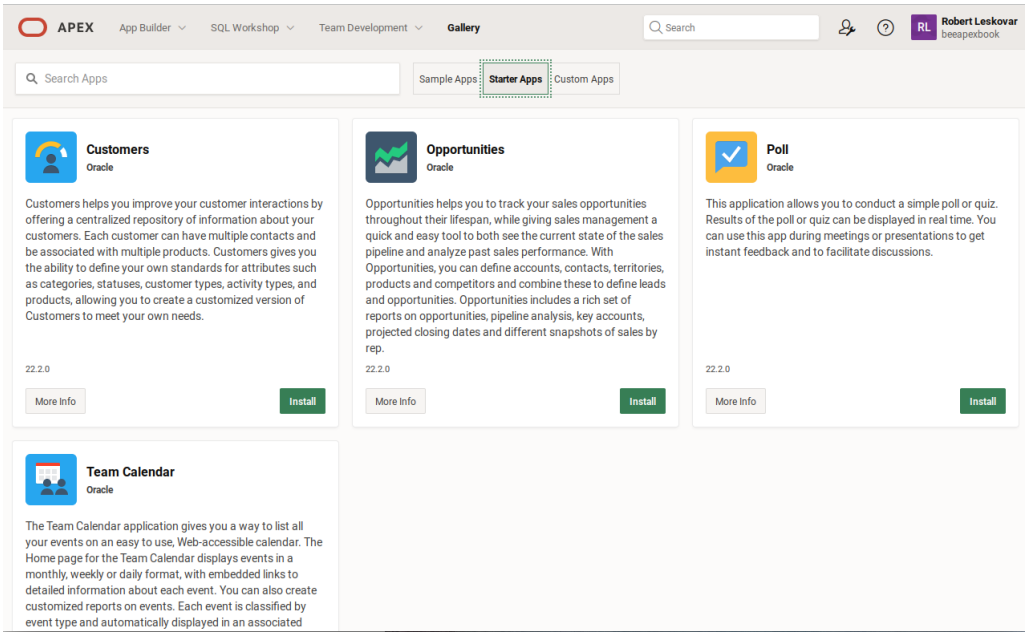
Wybierając z galerii ikonę “Remove App”, użytkownicy mogą odinstalowywać aplikacje, które zostały tam zainstalowane. Aplikacje, które zostały zaimportowane z pliku GitHub i pobrane, nie mają ikony odinstalowywania, dlatego - aby można było je odinstalować z obszaru roboczego - trzeba je usunąć.

Kroki do **odinstalowywania** aplikacji z Galerii:

1. Przejść na stronę startową obszaru roboczego.
2. Wybrać ikone "Gallery". "The Sample Apps", Śtarter Apps", i Ćustom Appssą wyświetlane po pierwszym załadowaniu strony Galeria.
3. Wybierz Śtarter Appsąlbo Śample Apps". Programy pobrane bezpośrednio z galerii będą miały ikonę Żemove"i ikonę Żun".
4. Aby usunąć aplikację, naciśnij przycisk Żemove".

11.2 Starter Apps (Aplikacje początkowe)

Aplikacje początkowe to przydatne aplikacje, które oferują autonomiczne rozwiązania punktowe, stworzone w celu spełnienia prostych wymagań, które nie wymagają znacznej i niepotrzebnej złożonej odpowiedzi. Aplikacje te mogą być używane ź pudełkałub mogą być rozszerzane o własne unikatowe funkcje (zob.rysunek 11.2).

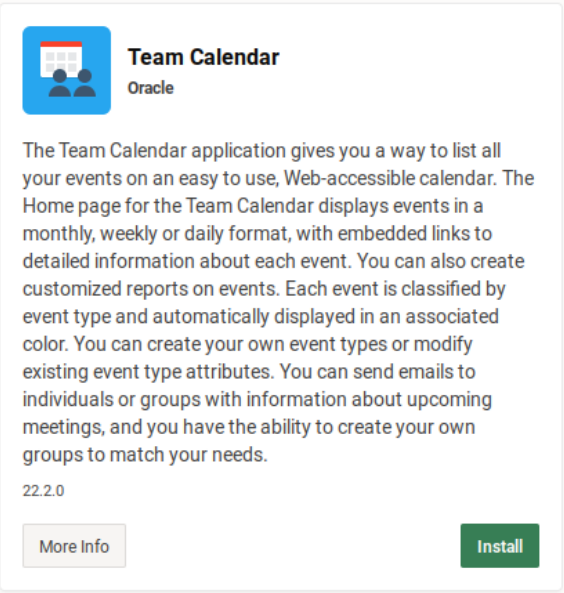


Rysunek 11.2: Aplikacje startowe w Galerii.

Przykładowe aplikacje startowe to:

- **Customers (Klienci)** – udostępniając skonsolidowaną bazę danych danych klientów, aplikacja Klienci pozwala użytkownikom usprawniać interakcje z klientami. Każdy klient może być połączony z kilkoma osobami kontaktowymi i przedmiotami. Klienci mogą ustawić własne kryteria dotyczące funkcji, takich jak kategorie, status, rodzaje klientów, typy aktywności i towary, dzięki czemu można zaprojektować specjalną wersję Klientów dostosowaną do własnych wymagań.
- **Opportunities (Okazje)** – Ta aplikacja zapewnia zarządzanie sprzedażą szybkim i prostym rozwiązaniem, sprawdzając obecny stan planowanej sprzedaży i analizując historyczne wyniki sprzedaży. Pomaga śledzić możliwości sprzedaży w całym cyklu życia. Można definiować potencjalnych klientów i okazje, łącząc konta, kontakty, regiony, produkty i konkurentów. Okazje oferują kompleksowy zbiór informacji na temat możliwości, badań planowanej sprzedaży, znaczących podmiotów, przewidywanych dat zamknięcia i zapisów stanu bieżącego innych przedstawicieli handlowych.
- **Poll (Ankieta)** – użytkownicy mogą używać tej aplikacji do tworzenia szybkiej ankiety lub testu. Wyniki ankiety lub quizu można zobaczyć natychmiast. Narzędzie to może być używane do ułatwiania dyskusji i uzyskiwania natychmiastowych informacji zwrotnych podczas spotkań lub prezentacji.
- **Team Calendar (Kalendarz Zespołu)** - za pomocą tej aplikacji użytkownicy zespołu mogą łatwo wyświetlać listę wszystkich wydarzeń w kalendarzu internetowym, który jest łatwy w użyciu. Strona startowa "Kalendarz Zespołuźawiera listę zdarzeń w stylu miesięcznym, tygodniowym lub dziennym wraz ze zintegrowanymi łączami do bardziej szczegółowych informacji o każdym wydarzeniu. Ponadto można projektować unikatowe raporty dotyczące zdarzeń. Każde zdarzenie jest klasyfikowane według kategorii, a odpowiedni kolor jest wyświetlany automatycznie. Użytkownicy mogą tworzyć nowe typy zdarzeń lub zmieniać atrybuty już istniejących typów zdarzeń. Użytkownicy mają możliwość tworzenia własnych grup, aby lepiej zaspokoić swoje potrzeby oraz mogą wysyłać e-maile do osób lub grup informujących ich o nadchodzących spotkaniach.

Oto przykład instalacji aplikacji dla kalendarza zespołu (zob.rysunek 11.3).

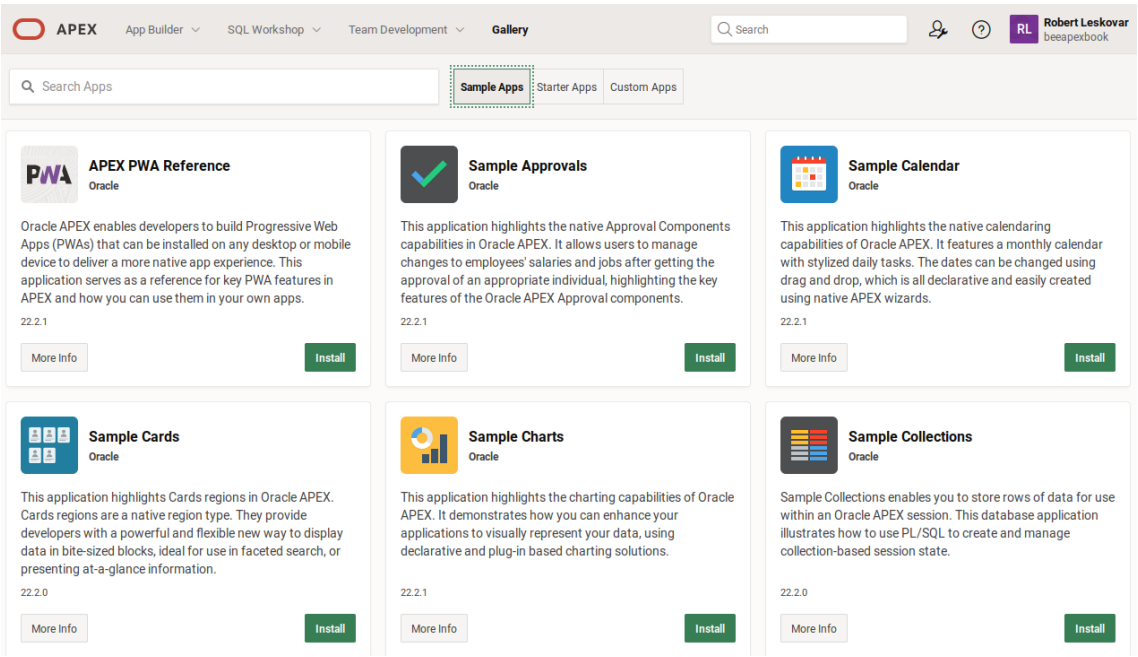


Rysunek 11.3: Instalacja aplikacji startowej Team Calender.

Kliknij przycisk "Install".

11.3 Sample Apps (Aplikacje przykładowe)

Przykładowe aplikacje są tworzone w celu wyróżnienia określonych możliwości i działania jako podręcznik programisty w celu wykorzystania określonej funkcji (zob.rysunek 11.4).



Rysunek 11.4: Przykładowe aplikacje.

Można między innymi wybrać:

- **APEX PWA Reference (Referencja APEX PWA)** - Aby zapewnić bardziej natywne środowisko aplikacji, programiści mogą tworzyć Progressive Web Apps (PWA) przy użyciu systemu Oracle APEX, który można wdrożyć na dowolnym komputerze lub urządzeniu mobilnym. Główne możliwości PWA w APEX i jak je stosować we własnych aplikacjach są zawarte w tej aplikacji.
- **Sample Approvals (Przykład Zatwierdzeń)** - ta aplikacja prezentuje możliwości natywnych komponentów do zatwierdzania w Oracle APEX. Wyróżnione są komponenty zgody Oracle APEX, które pozwalają użytkownikom zarządzać zmianami w płacach i stanowiskach pracowników po uzyskaniu zgody odpowiedniej osoby.
- **Sample Calendar (Przykładowy Kalendarz)** -ta aplikacja prezentuje wbudowane funkcje kalendarza w Oracle APEX. Zawiera ona miesięczny kalendarz z zaznaczonymi zadaniami na każdy dzień. Można używać funkcji przeciągnij i upuść, które są całkowicie deklaratywne i szybko generowane za pomocą natywnych kreatorów APEX, do aktualizacji dat.
- **Sample Cards (Przykładowe Karty)** – w systemie Oracle APEX ta aplikacja wyróżnia regiony kart. Regiony z kartami są formą regionu macierzystego. Dają programistom silną i adaptowalną nową metodę prezentowania danych w bitach o rozmiarze bitu, która jest idealna do użycia w wyszukiwaniu fasetowym lub szybkiego prezentowania informacji.
- **Sample Charts (Przykładowe Wykresy)** – ta aplikacja przedstawia funkcje wykresu Oracle APEX. Wyjaśnia, jak za pomocą narzędzi do tworzenia wykresów deklaratywnych i wtyczek można ulepszyć swoje aplikacje, tak aby wizualnie przedstawiały swoje dane.
- **Sample Collections (Przykładowe Kolekcje)** – ta aplikacja umożliwia użytkownikom przechowywanie wierszy danych w przykładowych kolekcjach w celu użycia ich w sesji Oracle APEX. Ta aplikacja bazy danych demonstruje tworzenie stanu sesji opartej na windykacji i zarządzanie nim za pomocą kodu PL/SQL.

- **Sample Data Loading (Przykładowe Ładowanie Danych)** - ta aplikacja jest tworzona przy użyciu prostych tabel EMP i DEPT w celu pokazania, w jaki sposób programiści mogą projektować strony umożliwiające użytkownikom dodawanie danych arkusza kalkulacyjnego do tabeli, która już istnieje.
- **Sample Dynamic Actions (Przykładowe Czynności Dynamiczne)** – ta aplikacja przedstawia szereg dynamicznych czynności, które można dodać do aplikacji. Te deklaratywne zachowania po stronie klienta obejmują proste przykłady modyfikowania sposobu wyświetlania składników, przykłady stylów dostosowywania wyglądu składników oraz przykłady interakcji z bazami danych po stronie serwera.
- **Sample File Upload and Download (Przykładowe wysyłanie i pobieranie plików)** – dowiedz się, jak budować aplikacje Oracle APEX z funkcjonalnością przesyłania i pobierania plików. Wykorzystaj dedykowane strony oraz okienka dialogowe do przesyłania plików. Sprawdź proces pobierania plików z kolumn BLOB bazy danych Oracle w tabelach bazodanowych. Zobacz w szczególności, jak tworzyć linki do pobierania plików w formularzach, interaktywnych raportach, tradycyjnych raportach i dynamicznych treściach HTML.
- **Sample Interactive Grids (Przykładowe siatki interaktywne)** – w tej aplikacji wyróżnione są funkcje i możliwości Oracle APEX Interactive Grid. Funkcje te są podkreślone na przykładowych stronach interaktywnej siatki.
- **Sample Maps (Przykładowe mapy)** – w tej aplikacji dostępnych jest wiele przykładów wyświetlania danych współrzędnych na mapie. Użyj elementu mapy termicznej, linii, wielkątów lub znaczników mapy. Możliwości przestrzenne Oracle (dostępne w każdej bazie danych Oracle) można łatwo zintegrować z Regionem Mapy APEX, aby przeprowadzić "Within Distance Search", Ńearest Neighbor Searchłub inne analizy przestrzenne.
- **Sample Master Detail (Przykładowy szczegół nadrzędny)** – ta aplikacja przedstawia funkcje nadrzędnych szczegółów Oracle APEX. W aplikacji istnieją cztery możliwe główne układy stron szczegółów. Pierwsze dwa układy korzystają z dostępnych do edycji tabel interaktywnych, aby wyświetlić szczegóły główne na jednej stronie. Ostatnie dwa układy prezentują główne szczegóły na dwóch stronach, używając kombinacji edytowalnych siatek interaktywnych, danych wejściowych formularza,tradycyjnych raportów i modalnych okienek.
- **Sample Reporting (Przykładowe raportowanie)** – ta aplikacja przedstawia funkcje raportowania Oracle APEX. Używając języka SQL, użytkownicy mogą deklaratywnie konstruować raporty interaktywne, siatki interaktywne, raporty z wyszukiwaniem, raporty w formie kart i raporty klasyczne.
- **Sample REST Services (Przykładowe usługi REST)** – ta aplikacja pokazuje, jak używać Oracle APEX do nawiązywania połączenia z zewnętrznymi usługami REST. Oracle.example.hr, prototyp RESTful Service, jest używany przez aplikację. Przykłady w tej aplikacji pokazują, jak filtrować, jak dodawać paginacje oraz jak tworzyć proste raporty tabelaryczne dotyczące danych usługi REST.
- **Sample Trees (Przykładowe drzewa)** – dowiedz się, jak używać zapytania SQL do budowania kontrolki drzewa. Aplikacja Oracle APEX w tym przykładzie prezentuje kilka sposobów na wykorzystanie kontrolki drzewa.
- **Universal Theme Reference (Universal Theme Reference)** - ta aplikacja wprowadza użytkowników w Uniwersalny Motyw, zapewniając prosty sposób nawigacji przez liczne szablony, wybory szablonów i style motywu. Przykłady pokazują, jak użytkownicy mogą szybko zarządzać układem strony, aby stworzyć efektowną aplikację.

11.4 Wtyczki

"Plug-ins"(Wtyczki) umożliwiają łatwe rozszerzanie środowiska Application Express o niestandardowe typy elementów, typy regionów, procesy i czynności Dynamic. Po zdefiniowaniu składniki

oparte na wtyczkach są tworzone i utrzymywane w bardzo podobny sposób jak standardowe składniki aplikacji Express. Wtyczki umożliwiają programistom tworzenie wysoce dostosowanych komponentów w celu zwiększenia funkcjonalności, wyglądu i przyjazności dla użytkownika aplikacji. Wtyczki umożliwiają użytkownikom dodawanie dodatkowych funkcji do aplikacji APEX, które nie są wbudowane w platformę. Społeczność APEX opracowała szeroką gamę różnych wtyczek przydatnych do rozszerzania istniejących funkcji lub dodawania nowych funkcji. Istnieją setki wtyczek dla Oracle Apex, podzielonych na pięć typów:

- Dynamic action (Dynamiczna Akcja)
- Region (Region)
- Item (Element)
- Process (Proces)
- Authentication (Autoryzacja)

Oracle Application Express oferuje różne wtyczki w każdej kategorii. Wtyczki dają możliwość deklaratywnego dodawania nowych funkcji do aplikacji, co pozwala rozszerzać wbudowane typy. Ponieważ wtyczki są przeznaczone do ponownego użycia, programiści mogą udostępnić je społeczności Oracle Application Express, używając repozytorium wtyczek, a także eksportować je i importować do różnych obszarów roboczych. Czynności związane z wdrożeniem wtyczki są następujące:

- Należy dodać wtyczkę do obszaru roboczego aplikacji lub utworzyć wtyczkę od podstaw.
- Aby korzystać z wtyczki, edytuj lub twórz typ dynamicznej akcji, element, region, proces lub schemat autoryzacji.
- Uruchom aplikację, aby sprawdzić funkcjonalność wtyczki.

Centralną lokalizacją dla programistów, którzy mogą udostępniać i pobierać wtyczki, jest Repozytorium Wtyczek. Można je znaleźć na stronie Oracle Technology Network.

- Aby zobaczyć repozytorium wtyczek, proszę przejść na stronę składników współużytkowanych:
  - Naciśnij przycisk App Builderńa stronie głównej obszaru roboczego.
  - Wybierz aplikacje.
  - Na stronie startowej aplikacji wybierz opcję “Shared Components” (Wspólne komponenty).
- Pojawi się strona współużytkowanych składników.
- Z menu “Other Components” (Inne składniki) wybrać opcję "Plug-ins"(Wtyczki).
- Kliknij przycisk "View Plug-in Repository".

Zostanie wyświetlone repozytorium dla wtyczek Oracle Application Express. Aby uzyskać dostęp do strony "Wtyczki", należy:

- Przejść do strony “Shared Components”.
  - Nacisnąć przycisk “App Builder” na stronie głównej obszaru roboczego.
  - Wybierz aplikację.
  - Na stronie startowej aplikacji wybierz opcję “Shared Components”. Pojawi się strona wspólnych komponentów.
- Z menu Other Components (Inne składniki) wybierz wtyczki. Karta Plug-ins jest automatycznie wybierana po załadowaniu strony. Zostanie wyświetlona cała lista wtyczek.

Aby utworzyć wtyczkę, należy:

- Przejść do strony Śhared Components".
  - Kliknąć przycisk App Builderńa stronie głównej obszaru roboczego.
  - Wybrać aplikacje.
  - Na stronie startowej aplikacji wybrać opcję Śhared Components". Strona zostanie wyświetlona.
- Wybierz “Plug-ins” w obszarze “Other Components”.
- Kliknij przycisk “Create". Zostanie wyświetlony kreator Plug-in (Tworzenie wtyczki).

- Po wybraniu metody tworzenia wtyczki w obszarze Ćreate Plug-in"kliknij przycisk Ńext".
- W obrębie nazwy:
  - Ńame"(Nazwa) - podać nazwę wtyczki.
  - "Internal Name"(Nazwa wewnętrzna) - należy podać wewnętrzną nazwę wtyczki. Nazwa musi być unikatowa w obrębie aplikacji.
  - "Type"(Typ) - Zdecyduj, który składnik może korzystać z tej wtyczki. Opcje w obszarze Wywołania zwrotne i Atrybuty standardowe różnią się w zależności od wybranego typu wtyczki. Więcej informacji można znaleźć w Pomocy na poziomie pola.
  - Ćategory- pojawia się tylko wtedy, gdy określona jest Dynamic Action (czynność dynamiczna). W interfejsie użytkownika wybierz kategorię, w której znajduje się wtyczka.
- W źródle:
  - "PL/SQL Code"(Kod PL/SQL) - Wprowadz anonimowy blok kodu PL/SQL zawierający kroki wymagane do renderowania, sprawdzania poprawności, uruchamiania i wykonywania wywołań zwrotnych tej wtyczki. Kod ten można także przechowywać w pakiecie PL/SQL w bazie danych ze względu na wydajność.
  - "Do not validate PL/SQL code"(Nie sprawdzaj poprawności kodu PL/SQL) - Aby wykonać analizę składniową kodu PL/SQL w trybie wykonawczym, wybierz tę opcję. Jeśli nie, wtyczka jest tworzona po przeanalizowaniu kodu.
- W obszarze Ćallbacks"(połączeń zwrotnych) skonfiduruj atrybuty. Typ wtyczki określa atrybuty, które są wyświetlane. Sprawdź pomoc na poziomie pola, aby sprawdzić próbki i dowiedzieć się więcej o atrybucie.
- W sekcji "User Interfaces", wybierz, które wyświetlenia musi obsługiwać App Builder dla tej wtyczki. Dostępne opcje to:
  - Desktop
  - Mobile
- W przypadku “Standard Attributes”(atrybutów standardowych) wybierz atrybuty, które mają zastosowanie do tej wtyczki. Są dostępne tylko dla niektórych wtyczek.
- W ramach informacji:
  - Version (Wersja)- aby określić wersję wtyczki, wprowadź ciąg znaków.
  - Informacje o adresie URL - należy podać adres URL prowadzący do strony startowej autora wtyczki lub aby uzyskać więcej informacji na temat wtyczki.
- Naciśnij przycisk Ćreate Plug-in".

Należy pamiętać, że powyższe dotyczy APEX w wersji 22.1. Nowsza wersja 23.1 zapewnia jeszcze lepszy sposób tworzenia wtyczki. Aby zaimportować wtyczkę, należy: To import a plug-in:

- Przejść do strony Śhared Components":
  - Wybrać aplikacje Ąpp Builderńa stronie startowej obszaru roboczego.
  - Wybrać aplikacje.
  - Wybrać opcje Śhared Componentsńa stronie startowej aplikacji.
- Wybierz "Plug-ins"w obrębie Óther Components".
- Wybierz opcje "Import". Zostanie wyświetlona strona "Import Plug-in"(Import wtyczki).
- Dla określonego pliku:
  - “Import file“ (Plik importu) - wprowadz lub przejdz do nazwy pliku importu.
  - “File Type” (Typ pliku) - wybierz wtyczkę.
  - “File Character Set” (Zestaw znaków pliku) - Wybierz kodowanie zestawu znaków dla plików importu.
  - Kliknij przycisk “Next”(Dalej).
- Dla opcji "File Import Confirmation” (Potwierdzenie importu pliku) wybierz opcję Ńext".
- Dla opcji “Install” wybierz opcję “Install Plug-in”.

Aby wyeksportować wtyczkę ze strony "Plug-in":

- Przejdź do strony Śhared Components":
  - Na stronie startowej obszaru roboczego naciśnij przycisk Ąpp Builder"
  - Wybierz aplikację.
  - Na stronie startowej Aplikacja nacisnij przycisk Śhared Components".
- W obszarze “Other Components” wybierz opcję “Plug-ins”.
- W obszarze ”Tasks” wybierz opcję “Export” wtyczkę.
- Na stronie “Export Plug-in”(Eksport wtyczki):
  - Application - Należy wybrać aplikację, z której ma zostać wyeksportowana wtyczka.
  - Plug-in - Wybierz wtyczkę.
  - File Format- należy wybrać format pliku eksportu.
  - Wybierz Ęxport"
- Kliknij przycisk Ęxport".

W następnych kilku liniach podano przykład wtyczki. Przykład odnosi się do zaktualizowanej aplikacji kalendarza i wtyczki dynamicznej akcji. Ta przykładna wtyczka może być używana do formatowania dni w nowym elemencie wybierania daty. Użytkownicy mogą ją wyłączyć, dodać etykiety narzędziowe lub klasy do każdego dnia w siatce dni w elemencie wybierania daty. Nie ma wymagań wstępnych dotyczących korzystania z tej wtyczki. Jak jej używać:

- Utwórz aplikacje
- Zaimportuj wtyczkę w Śhared Componets- "Plug-ins".
- Dodaj nowy selektor daty
- Dodaj czynność dynamiczną przy ładowaniu stron i wybierz jako czynność zaimportowaną wtyczkę
- Ustaw adres URL standardowego formatu pliku iCalendar lub wprowadz zapytanie SQL, które ma sformatować dni kalendarza
- Wybierz odpowiedni element wybierania daty jako Ąffected Element” (Element Dotknięty) i uruchom stronę

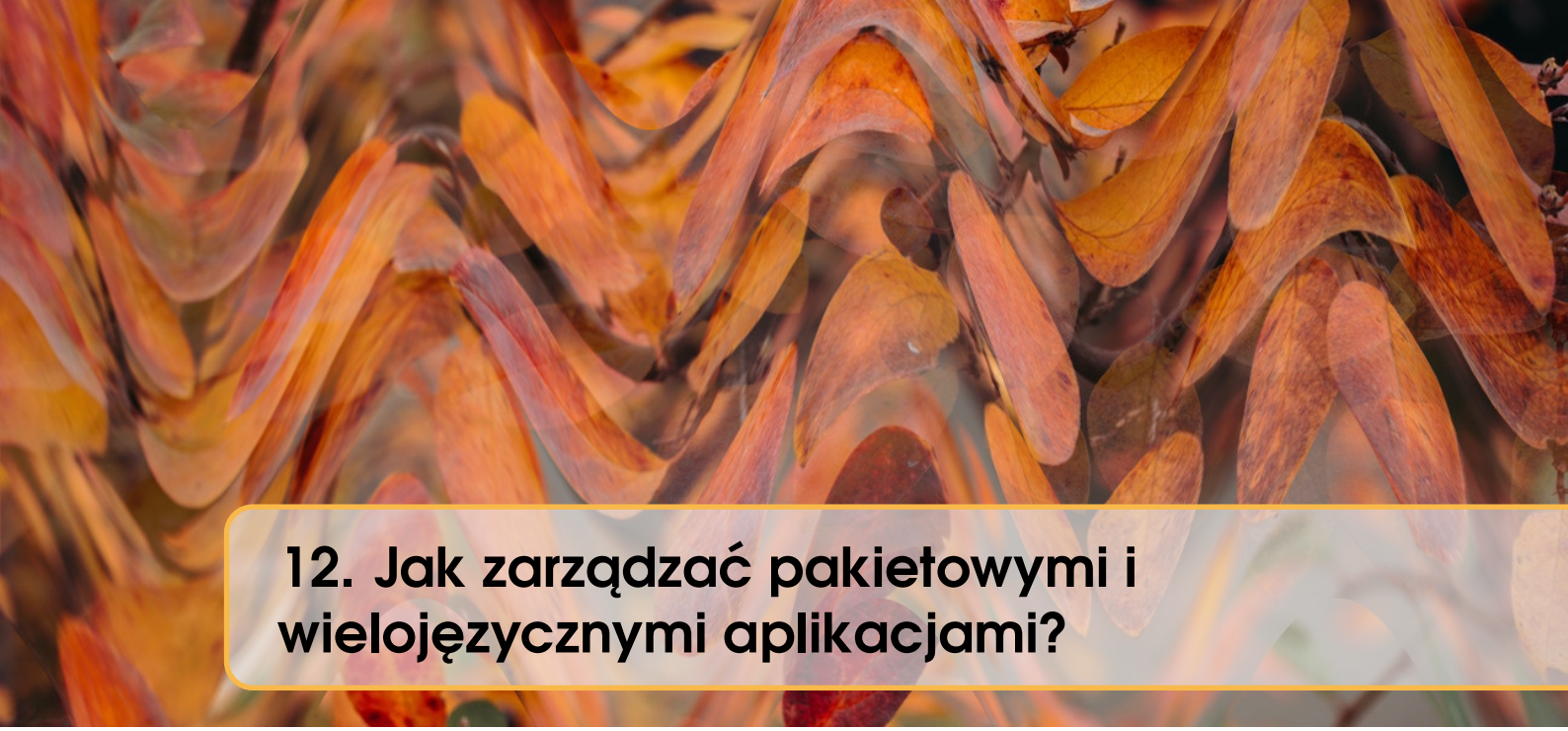
W tym rozdziale wykazano, że wtyczki mogą oferować kilka korzyści. Bez względu na to, jak silna może być, żadne pojedyncze oprogramowanie nie może zapewnić każdej funkcji dla każdego użytkownika. Luka między formą a funkcją jest wypełniana przez wtyczki. Ułatwiają dodawanie do oprogramowania i aplikacji określonych funkcji. W związku z tym wtyczki oszczędzają czas. Największą korzyścią z używania wtyczek jest skrócenie czasu potrzebnego na rozwój, co obniża całkowity koszt. Ponadto wtyczki umożliwiają użytkownikom dostosowywanie funkcji i możliwości. Większość wtyczek umożliwia włączanie i wyłączanie określonych ustawień. Usunięcie wtyczki, jeśli zajdzie taka potrzeba, jest prostym procesem.

11.5 Pytania

1. Jakie są sample apps(przykładowe aplikacje) i jakie są ich korzyści?
2. Jakie są starter apps (aplikacje startowe) i jakie są ich zalety?
3. Jakie są korzyści z wtyczek?

11.6 Odpowiedzi

1. Sample Apps (Przykładowe aplikacje) są tworzone w celu wyróżnienia określonych możliwości i pełnienia funkcji podręcznika programisty.
2. Starter Apps (Aplikacje startowe) to przydatne aplikacje, które oferują autonomiczne rozwiązania punktowe, stworzone w celu spełnienia prostych wymagań, które nie wymagają znacznej i niepotrzebnej złożonej odpowiedzi.
3. Wtyczki umożliwiają programistom tworzenie wysoce dostosowanych komponentów w celu zwiększenia funkcjonalności, wyglądu i przyjazności dla użytkownika aplikacji. Wtyczki umożliwiają użytkownikom dodawanie funkcji do aplikacji APEX, które nie są wbudowane.



12. Jak zarządzać pakietowymi i wielojęzycznymi aplikacjami?

R. LESKOVAR, U. RAJKOVIČ, A. BAGGIA

12.1 Aplikacja i aplikacja pakietowa

W tej sekcji opisano, jak wykorzystać potencjał funkcji Quick SQL do generowania aplikacji praktycznie bez znajomości języka DDL i całkowicie w ramach systemu Oracle APEX. Po wygenerowaniu aplikacji z Quick SQL przygotujemy zapakowaną aplikację i przeniesie ją do nowego obszaru roboczego (tj. “production site”(strona produkcyjna)).

W rozdziale 11 przedstawiono aplikacje dostarczane w APEX przez Oracle. Wnioski te są podzielone na dwie grupy: a) aplikacje produktywnościowe i b) przykładowe aplikacje. Aplikacje produktywnościowe zaspokajają specyficzne potrzeby biznesowe. Wbudowane funkcje w większym lub mniejszym stopniu spełniają wymagania narzucone przez sytuację biznesową. Przykładowe aplikacje demonstrują różne funkcje. Obydwa rodzaje mogą zainspirować programistów do dostosowania ich i spełnienia potrzeb biznesowych. Aplikacje te to przykłady aplikacji pakietowych. Podczas instalowania (terminem APEX jest import) tych pakietów aplikacji można zauważyć, że obiekty bazy danych (takie jak tabele, sekwencje, triggery, funkcje, procedury i pakiety) są tworzone, a dane wprowadzane do tabel.

Najkrótsza definicja **aplikacji pakietowej** mówi, że jest to aplikacja wbudowana, którą można zainstalować i skonfigurować w obszarze roboczym Oracle APEX. APEX wykorzystuje dwie ważne koncepcje ściśle związane z migracją aplikacji (eksportem i importem): a) obiekty pomocnicze i b) elementy wspólne. Obiekty pomocnicze kontrolują instalację, aktualizację i deinstalację, definiując wymagania wstępne, ciągi zastępcze, opcje kompilacji, walidacje przed instalacją, skrypty i komunikaty. Elementy wspólne definiują logikę aplikacji, zabezpieczenia, nawigację i wyszukiwanie, interfejs użytkownika, pliki i raporty, źródła danych, procesy Workflow (jeśli zostały zainstalowane), globalizację i inne. Tak więc aplikacje pakietowe są aplikacjami ze zdefiniowanymi obiektami pomocniczymi. W tej chwili możesz zadać sobie pytanie: "Dlaczego i w jaki sposób programiści mogą tworzyć niestandardowe zapakowane aplikacje?".

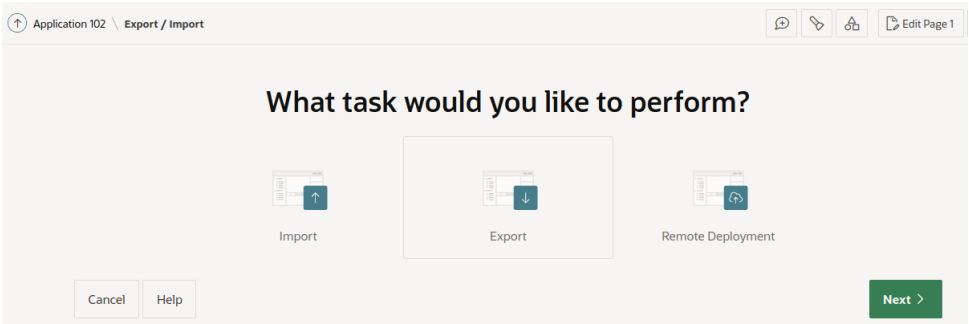
Odpowiedź na pytanie “dlaczego”: programiści tworzą niestandardowe aplikacje pakietowe, jeśli istnieje wymóg migracji aplikacji do nowego obszaru roboczego z obiektami pomocniczymi i składnikami współużytkowanymi. Definicje obiektów bazy danych (za pomocą instrikcji DDL) i operacje na tych obiektach (za pomocą instrukcji DML) są wykonywane podczas instalacji, aktualizacji i deinstalacji. Uproszczając: podczas instalacji tabele mogą być tworzone i wypełniane danymi, podczas uaktualniania tabele mogą być modyfikowane, a podczas deinstalacji tabele mogą być oczyszczane lub usuwane. To samo może być wymuszone dla innych obiektów bazy danych,

takich jak wyzwalacze, sekwencje, procedury składowane, funkcje i pakiety. Główną korzyścią pakietu jest to, że aplikacja jest gotowa do działania z definicjami danych i samymi danymi.

Odpowiedź na pytanie "jak" obejmuje następujące kroki:

- zdefiniuj zakres aplikacji
- stwórz i wykonaj skrypt do zdefiniowania tabel
- stwórz i wykonaj skrypt do wstawienia przykładowych danych
- wygeneruj aplikację
- dokonaj dostosowań w aplikacji
- przygotuj zapakowaną aplikację i przetestuj ją w nowym obszarze roboczym

Zanim przejdziemy krok po kroku, spójrzmy na kreator importu i eksportu aplikacji w APEX. Aby przenieść aplikację do nowego obszaru roboczego, aplikację trzeba wyeksportować, a następnie zaimportować. Kreator eksportu w APEX zapisuje plik tekstowy (instrukcje PL/SQL) na: a) komputerze programisty lub b) na zdalnym obszarze roboczym za pośrednictwem usługi REST (zob.rysunek 12.1).



Rysunek 12.1: Kreator importu i eksportu aplikacji.

W opcjach eksportu dostępnych jest kilka możliwości, takich jak podział na wiele plików, wybór zestawu znaków pliku, opcja zaznaczenia lub odznaczenia raportów publicznych lub prywatnych oraz podobne. W większości przypadków początkujący użytkownicy pozwalają APEX na zastosowanie domyślnych ustawień. Następnie programista loguje się w nowym obszarze roboczym i uruchamia kreator importu, który wczytuje wyeksportowany plik tekstowy. Definicje tabel, widoków, wyzwalaczy, sekwencji, funkcji, procedur i pakietów nie są przenoszone wraz z aplikacją, jeśli nie przygotujemy obiektów wspierających. Jeśli docelowy obszar roboczy już zawiera definicje obiektów bazy danych i dane w tabelach, to powinniśmy przygotować obiekty wspierające, do obsługi w takiej sytuacji.

Pierwsza część tego rozdziału będzie w możliwie największym stopniu korzystać z kreatorów APEX. Następnie zostaną przedstawione proste instrukcje SQL INSERT, które nie są obsługiwane przez kreator w APEX. Aby wygenerować instrukcje INSERT, czytnik może importować dane do arkusza kalkulacyjnego, a następnie wyeksportować instrukcje INSERT za pomocą innych narzędzi (tj. SQL Developer lub TOAD). Aplikacja w tym rozdziale przedstawi komponent strony o nazwie listy wartości (LOV). Należy do składników współużytkowanych. Lista wartości ma dwa cele: użytkownik łatwo wypełni pole formularza na stronie internetowej lista rozwijana, przycisk opcji) oraz zapewnienie ustalonych wartości, aby uniknąć błędów przy wpisywaniu. Dostępne są dwa typy list wartości: dynamiczne są oparte na zapytaniu SQL, a statyczne oferują ustalony zestaw wartości. Zaletą dynamicznej listy wartości jest to, że domena zależy od tabeli lub widoku. Użytkownik końcowy może zmieniać domenę, wstawiając, aktualizując i usuwając wiersze w tabeli. Wady to konieczność tworzenia i zarządzania tabelą. Jeśli domena nie ulegnie zmianie, a liczba wierszy w tabeli jest mniejsza niż można zastosować statyczną listę wartości. W takim przypadku tylko deweloper może zmienić domenę (jest to "hard-coded").

Druga część pokaże tworzenie zapakowanej aplikacji, jej eksport i import do nowego obszaru

roboczego.

Trzecia część rozdziału zawiera instrukcje tworzenia aplikacji wielojęzycznej. Jako dodatkowy materiał dostarczane są trzy wyeksportowane aplikacje oraz nagranie wideo, które szczegółowo pokazuje proces tworzenia.

12.2 Aplikacja

12.2.1 Zakres zastosowania

Aplikacja ma na celu zarządzanie stanowiskami pracy i kompetencjami w dziale HR. W firmie może być kilkaset stanowisk pracy. Jedno miejsce pracy może obejmować określony podzbiór wszystkich znanych kompetencji. Ten wymagany jest określony przez dział kadr. Poziomy kompetencji to opisowe zdolnosci do: pomocy, użytkowania, opanowania, dostosowania i innowacji. Celem jest zatem opracowanie aplikacji do przygotowywania i zarządzania profilami stanowisk pracy. Aplikacja wykorzystuje trzy tabele: JOB, COMPETENCE and JOB_COMPETENCE z minimalną liczbą pól danych (w celu uproszczenia i zwięzłości). Każda tabela zawiera klucz główny, a tabela JOB_COMPETENCE ma dwa klucze obce, aby powiązać każdy wiersz z wierszami tabeli JOB i COMPETENCE. Klucze obce zapewniają spójność danych, a mianowicie wiersz w tabeli JOB_COMPETENCE może istnieć tylko wtedy, gdy wiersze z tymi samymi kluczami głównymi istnieją w tabelach JOB i COMPETENCE. Aby wprowadzić pewną dyscypline, nazwy tabel mają przedrostek ĆH12_".

12.2.2 Tworzenie tabel

Możemy tworzyć tabele w APEX:

- Bez znajomości języka opisu danych, klikając w przeglądarkę obiektów. Ilustrowanie tego procesu za pomocą wielu rysunków zabrałoby zbyt wiele miejsca w tej książce.
- Z wiedzą z zakresu DDL (Data Definition Language) w SQL Workshop > SQL Commands, wpisując składniowo poprawne polecenia.
- Z podstawową wiedzą z zakresu modelowania danych (zob. rozdział 2 2) i narzędziem Quick SQL.

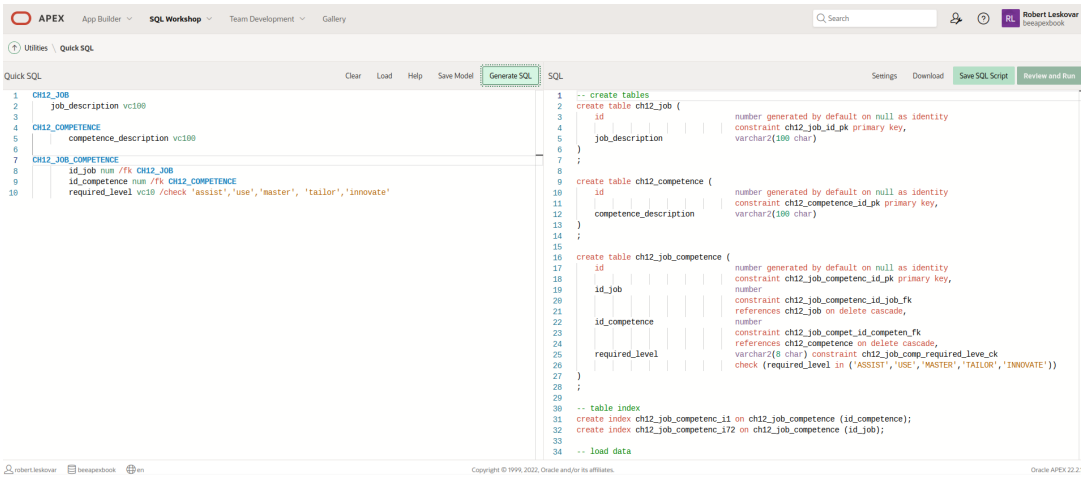
Aby zapewnić szerokie zrozumienie i zwięzłość, zostanie użyta funkcja Quick SQL (znajdź funkcję Quick SQL w oknie SQL Workshop > Utilities > Quick SQL). Zostanie wstawiony następujący tekst:

```
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'
```

Po wprowadzeniu powyższego tekstu i naciśnięciu przycisku **Generate SQL** pojawia się następujący ekran(zob.rysunek 12.2):

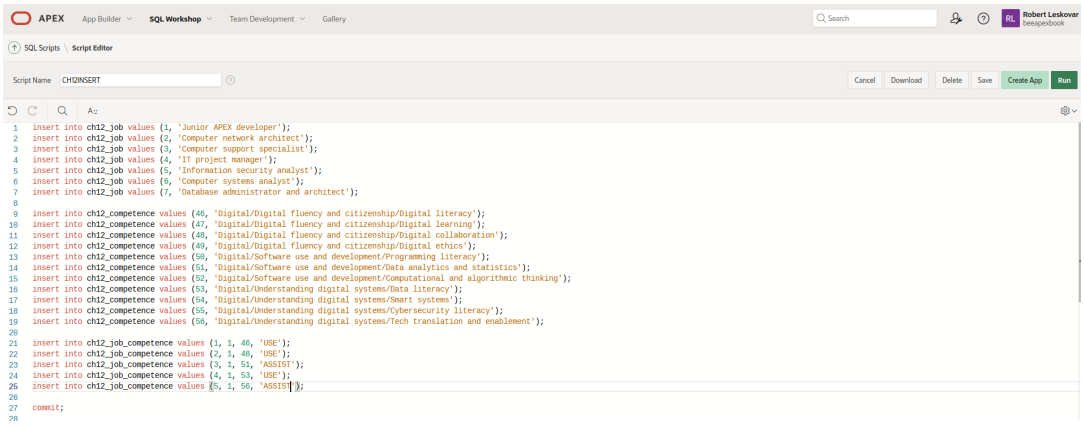


Rysunek 12.2: Przekształcanie Quick SQL do poleceń SQL.

Następnie naciśnij przycisk Śave SQL Script"(Zapisz skrypt SQL) i nadaj skryptowi nazwę CH12CREATE. Teraz możemy uruchomić skrypt. Wróć do Skryptów, znajdź ikonę **Run** w wierszu, w którym jest wyświetlany skrypt CH12CREATE. Kliknij ikonę. Trzy tabele i trzy indeksy powinny zostać pomyślnie utworzone. Możesz sprawdzić tabele w “Object Browser"(Przeglądarce obiektów).

12.2.3 Wstawianie danych

W APEX wybierz SQL Workshop > SQL Scripts > Create. Teraz nadszedł czas, aby napisać kilka poleceń SQL INSERT, aby wypełnić dane w tabelach. Tabla CH12_JOB będzie miała siedem wierszy opisujących różne stanowiska. Tabla CH12_COMPETENCE będzie miała jedenaście wierszy opisujących jedenaście kompetencji cyfrowych zdefiniowanych przez McKinsey’s DELTAs (odrębne elementy talentu). Polecimy wstawić pięć kompetencji do stanowiska "Junior Apex developer". Nazwijmy skrypt CH12INSERT i napiszmy skrypt (zob. rysunek 12.3).

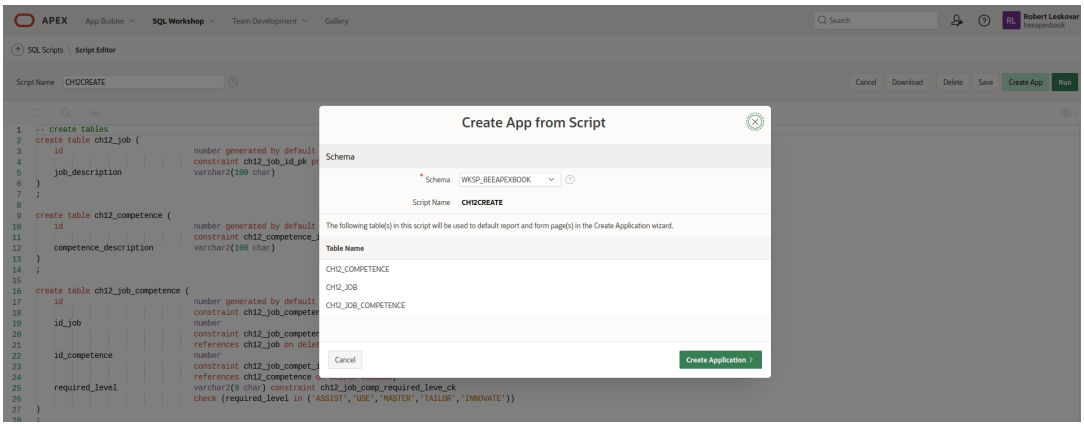


Rysunek 12.3: Skrypt CH12INSERT wstawia dane do trzech tabel i zatwierdza transakcje.

Kliknij przycisk **Create** . Teraz należy wykonać skrypt CH12INSERT. Wróć do skryptów, kliknij ikonę **Run** w wierszu, gdzie jest pokazany skrypt CH12INSERT. Kliknij **Run now** . Powinno być pomyślnie utworzonych dwadzieścia cztery wiersze. Zawartość tabel można także sprawdzić w “Object browser” (Przeglądarce obiektów).

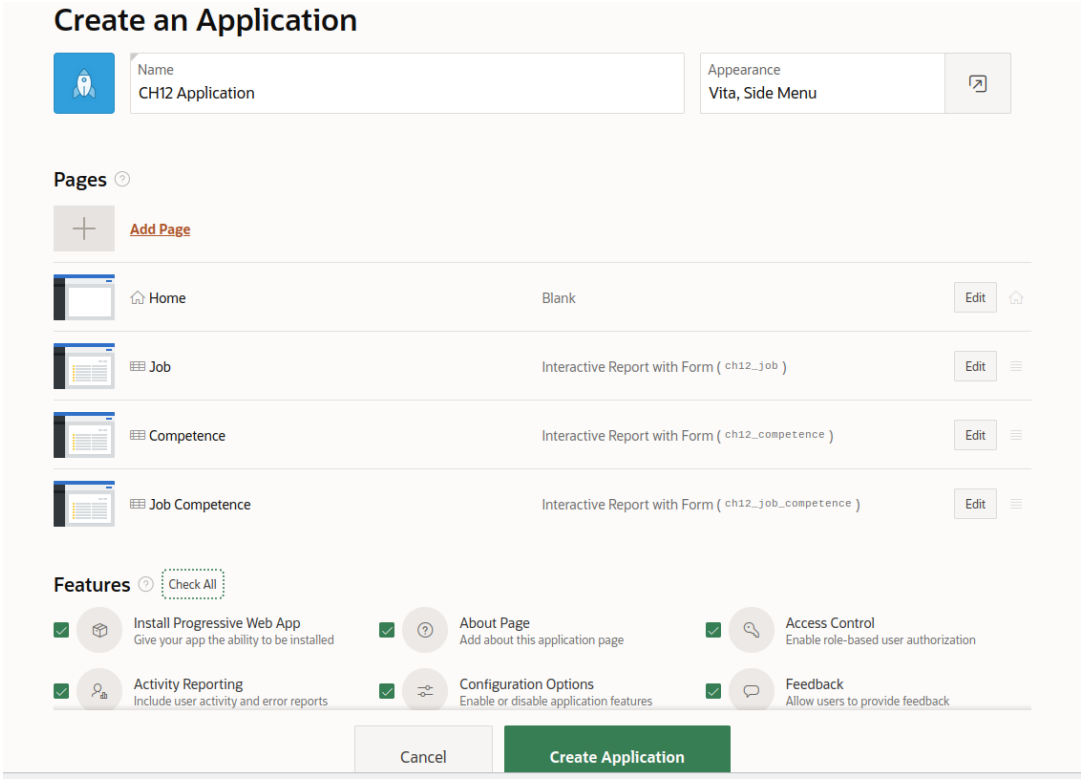
12.2.4 Generuj aplikacje

Po wypełnieniu tabel danymi testowymi przejdź do polecenia SQL Workshop > Scripts i edytuj skrypt CH12CREATE. Kliknij przycisk **Create App** (zob.rysunek 12.4). Możesz zauważyć, że na liście znajdują się trzy tabele.



Rysunek 12.4: Tworzenie aplikacji ze skryptu.

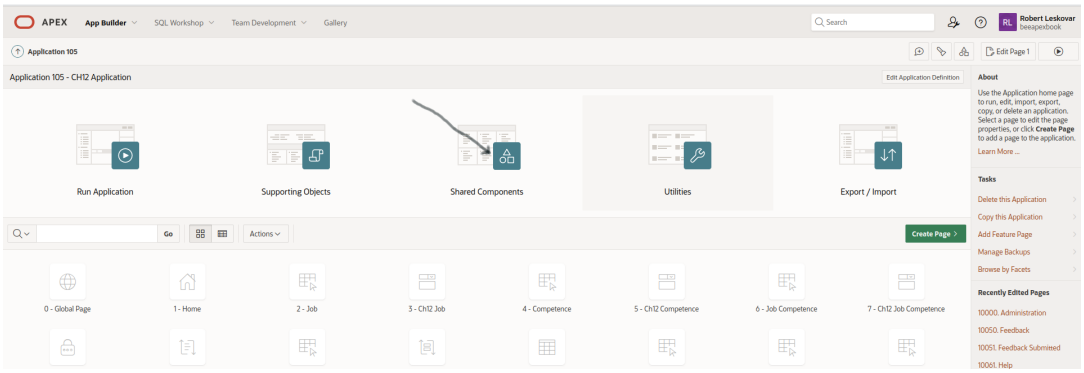
Nacisnąć przycisk Ćreate Application". Podaj nazwę aplikacji (np. CH12 Application), nacisnij przycisk Ćheck all features" (Sprawdź wszystkie funkcje), a następnie Ćreate Application" (zob.rys 12.5).



Rysunek 12.5: Wybieranie nazwy aplikacji i wszystkich funkcji.

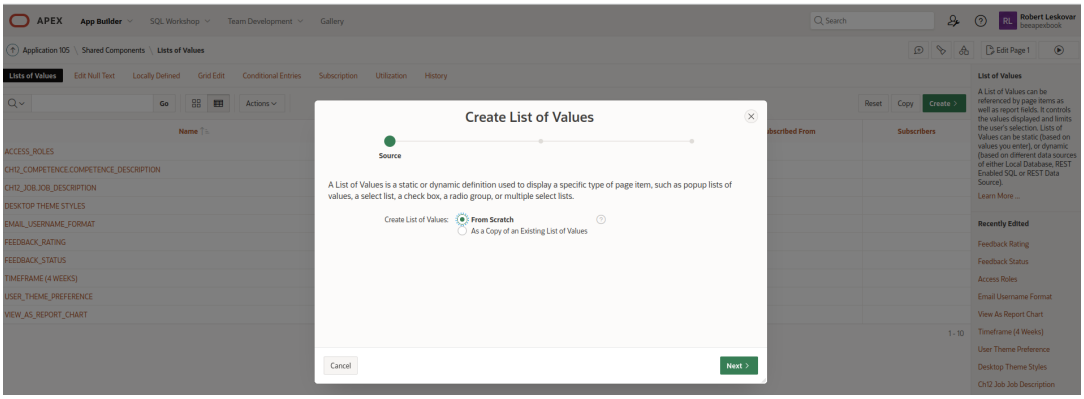
Po zakończeniu generowania możemy uruchomić aplikację. W nowszych wersjach APEX generator tworzy także dwie listy wartości (CH12_COMPETENCE.COMPETENCE_DESCRIPTION i CH12_JOB.JOB_DESCRIPTION), skrótowo nazywane LOV. Listy wartości są komponentami

na stronie, gdzie jedna wartość jest wyświetlana i zrozumiana przez użytkownika, podczas gdy strona używa odpowiadającego klucza lub kodu do operacji dodawania, aktualizacji lub usuwania. Na przykład: na formularzu na stronie jest wyświetlana nazwisko pracownika, a po naciśnięciu przycisku potwierdzenia przechowywany jest odpowiadający numer identyfikacyjny pracownika. Wygenerowane listy wartości są oparte na tabelach i nie jest wymagana uprzednia znajomość języka SQL. W terminologii APEX są one nazywane dynamicznymi, ponieważ zmiana w tabeli wygenerowałaby nową listę wartości. Wygenerowane listy wartości można znaleźć w menu Shared Components > List of Values i określić sposób ich ustawienia. W tym rozdziale dowiemy się, jak tworzyć dynamiczne listy wartości (oparte na prostym zapytaniu SQL) oraz statyczne, oparte na stałym zestawie wartości. Poprzez zastosowanie list wartości (LOV) na stronach aplikacji uczynimy ją bardziej odporną na błędy i bardziej przyjazną dla użytkownika. Oznacza to, że użytkownik nie może wprowadzić dowolnej wartości w polu formularza, co ograniczy liczbę błędów przy wpisywaniu. Przejdź do “Shared components"(składników współużytkowanych)(zob.rysunek 12.6).



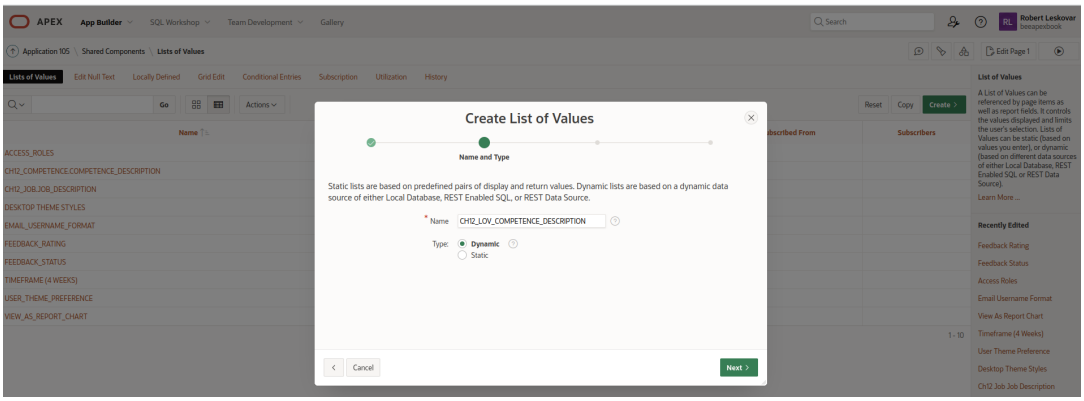
Rysunek 12.6: Wybieranie “Shared components“(składników współużytkowanych).

W Śhared components” wybierz “list of values”(listę wartości)w innych składnikach i kliknij przycisk “Create”. Utworzymy listę wartości od podstaw (zob.rysunek 12.7).



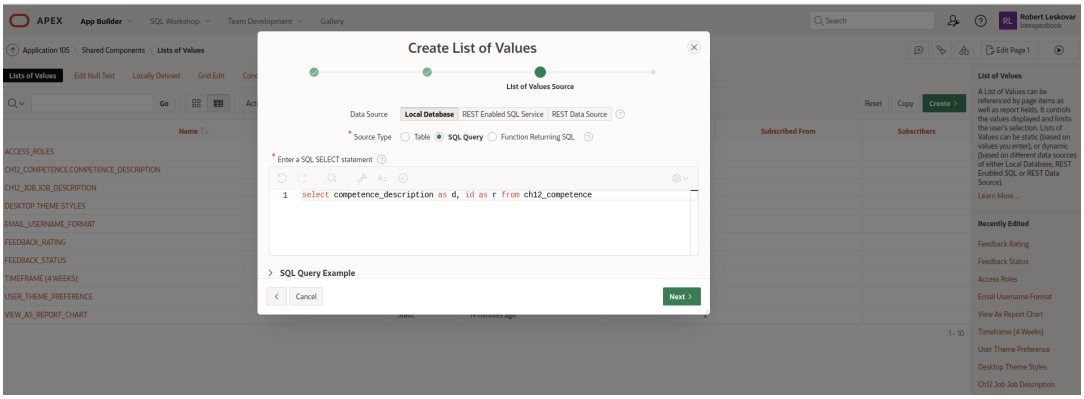
Rysunek 12.7: Tworzenie listy wartości od podstaw.

Nazwa listy wartości LOV jako CH12_LOV_COMPETENCE_DESCRIPTION i wybór typu Dynamicznego (zob.rysunek 12.8)



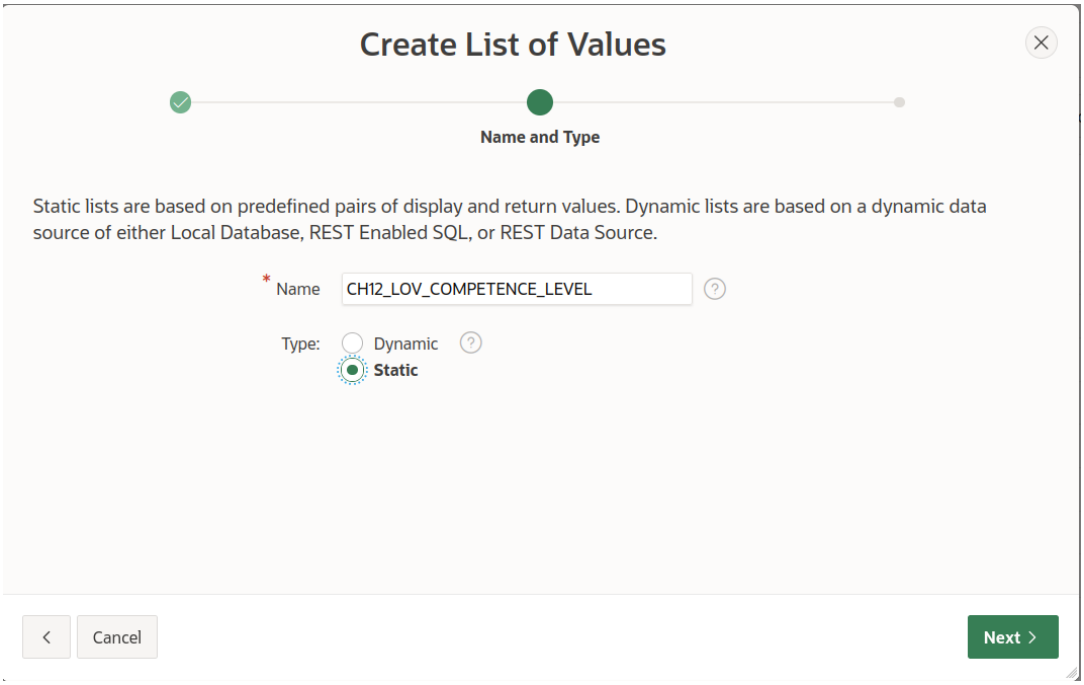
Rysunek 12.8: Definiowanie nazwy i typu: CH12_LOV_COMPETENCE_DESCRIPTION jest dynamiczne.

Jako typ źródła wybierz SQL, po czym wpisz zapytanie ślect competence_description as d, id as r from ch12_competence"jako instrukcje SQL (zob.rysunek 12.9).

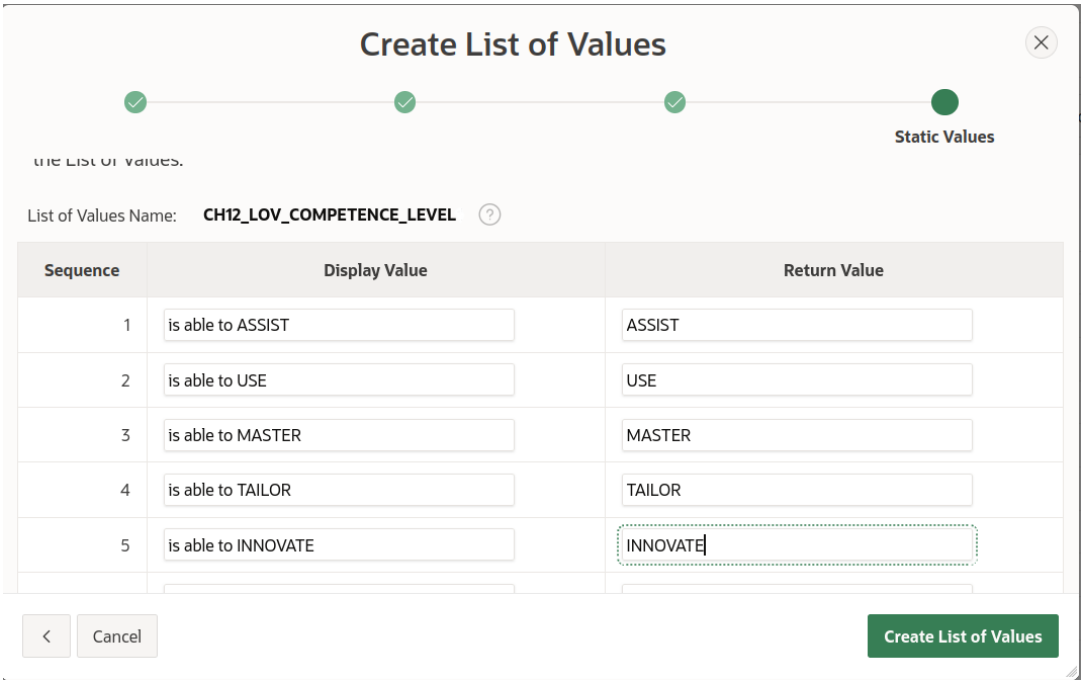


Rysunek 12.9: Wprowadzanie polecenia SQL SELECT.

Ten zestaw wartości będzie wyświetlać opis kompetencji (kolumna wyświetlana) na stronie formularza, podczas gdy zwraca ID (kolumna zwrotna). Kolejne LOV będzie statyczne, o nazwie CH12_LOV_COMPETENCE_LEVEL i zostanie utworzone zgodnie z rysunkiem 12.10 i rysunkiem 12.11.

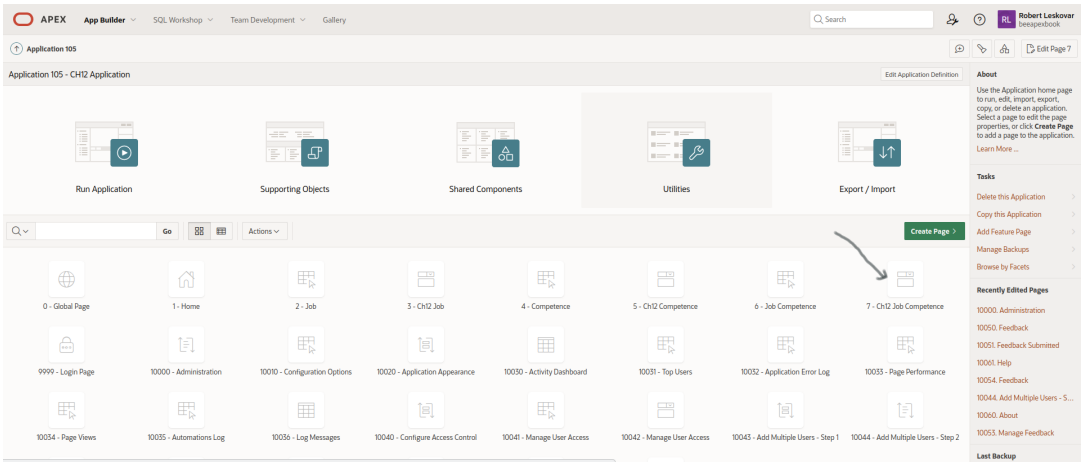


Rysunek 12.10: Zdefiniuj nazwę i typ: CH12_LOV_COMPETENCE_LEVEL jest statyczny.



Rysunek 12.11: Zdefiniuj wartości wyświetlane i zwracane: CH12_LOV_COMPETENCE_LEVEL jest statyczny.

Przejść do stron aplikacji (zob.rysunek 12.12). Wybierz stronę (Ch12 Job Competences).



Rysunek 12.12: Wybierz stronę 7 (Ch12 Job Competences).

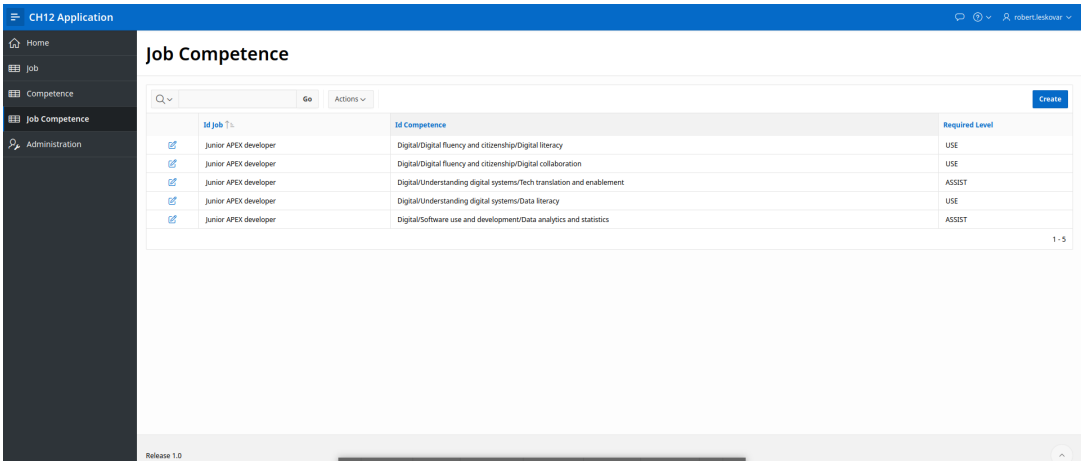
Modyfikowanie elementu strony P7_ID_COMPETENCE:

- Typ to Łista wyboru"w sekcji Identyfikacja.
- Typ na "Współdzielone komponenty"w sekcji Listy wartości.
- Wybierz listę wartości CH12_LOV_COMPETENCE_DESCRIPTION

Zmień element strony P7_REQUIRED_LEVEL:

- Typ to Łista wyboru"w sekcji Identyfikacja.
- Typ na "Współdzielone komponenty"w sekcji Listy wartości.
- Wybierz listę wartości CH12_LOV_COMPETENCE_LEVEL

Zapisz strone 7, przejdz do strony 6 i uruchom aplikacje. Powinieneś zobaczyć rysunek 12.13.

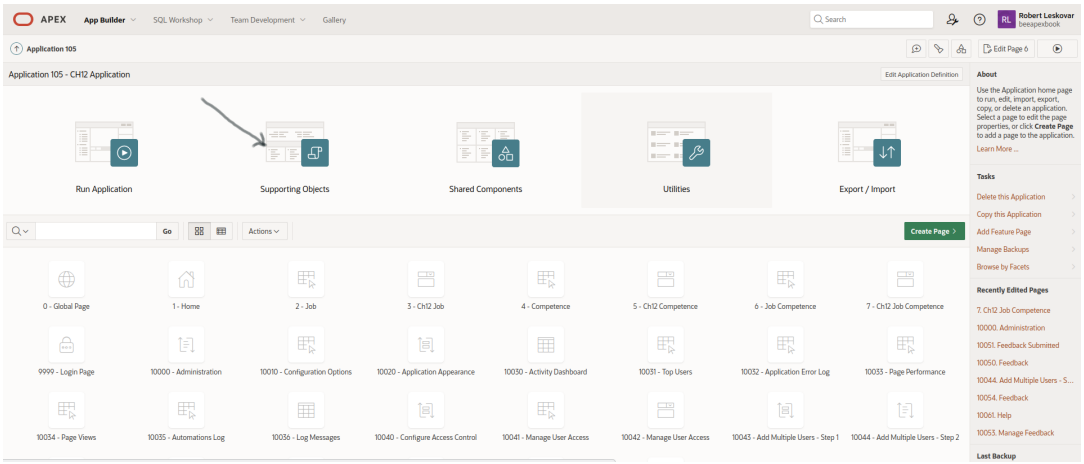


Rysunek 12.13: Raport strona 7(Ch12 Job Competences).

Teraz edytuj pierwszy wpis. Obie listy wyboru powinny zawierać oczekiwane dane.

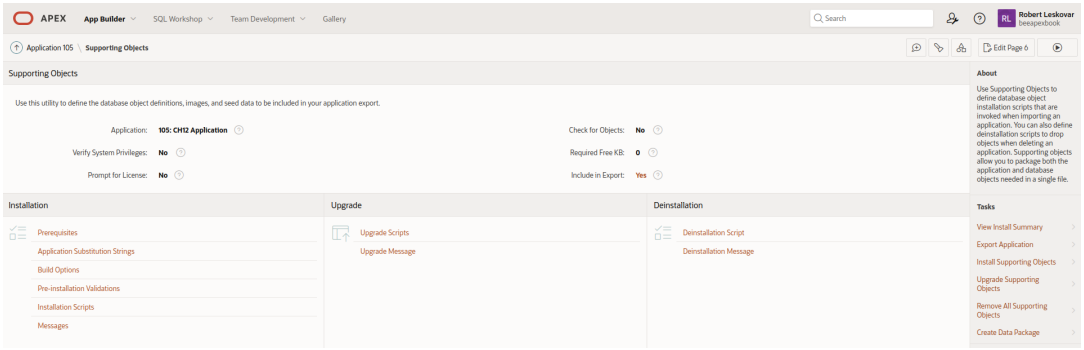
12.3 Aplikacja pakietowa

Ponieważ utworzyliśmy już niektóre skrypty (CH12CREATE, CH12INSERT) bardzo szybko zbudujemy teraz aplikację spakowaną. Przejdź do aplikacji i wybierz opcję “Supporting Objects” (Obiekty pomocnicze)(zob.rysunek 12.14).



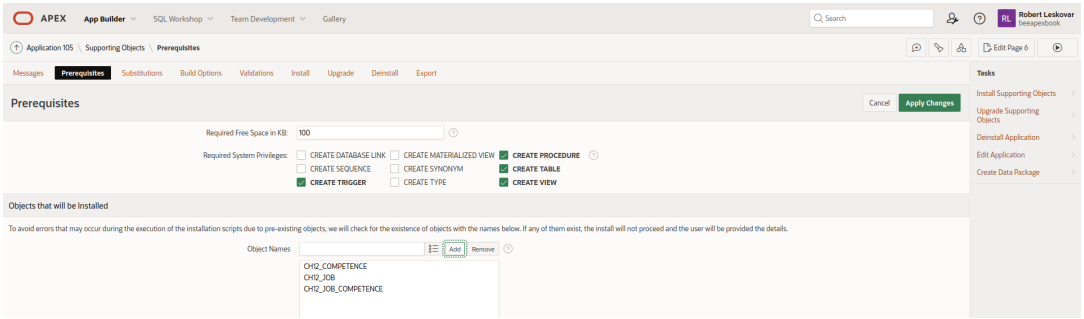
Rysunek 12.14: Wybieranie obiektów pomocniczych.

Teraz możemy ustawić instalację (wymagania wstępne, napisy substytucyjne aplikacji, opcje kompilacji, weryfikacje przedinstalacyjne, skrypty instalacyjne i komunikaty), uaktualnienie (skrypty uaktualniania, komunikat uaktualniania), deinstalację (skrypt deinstalacji, komunikat deinstalacji). Zob. rysunek 12.15.



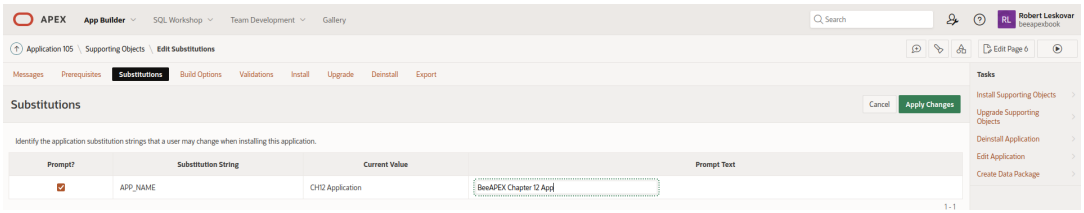
Rysunek 12.15: Ustawianie wymagań wstępnych.

W celu ochrony użytkownika przed przypadkowym usunięciem istniejących danych, sprawdzimy istnienie trzech tabel (zob. rysunek 12.16). Kliknij przycisk “Apply changes” (Zastosuj zmiany).



Rysunek 12.16: Sprawdzenie istnienia trzech tabel.

W Application Substitution String zapytamy, czy nazwa aplikacji powinna być "BeeAPEX Rozdział 12 App"(zob.rysunek 12.17).

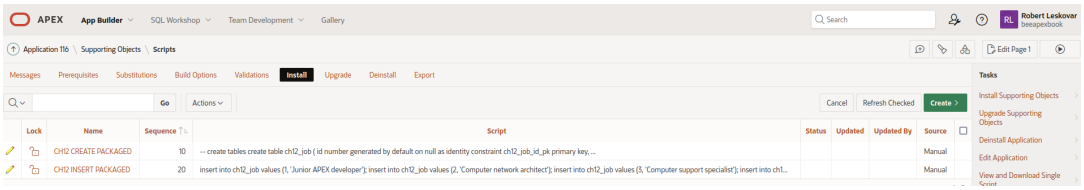


Rysunek 12.17: Ustaw prompt do zmiany nazwy aplikacji.

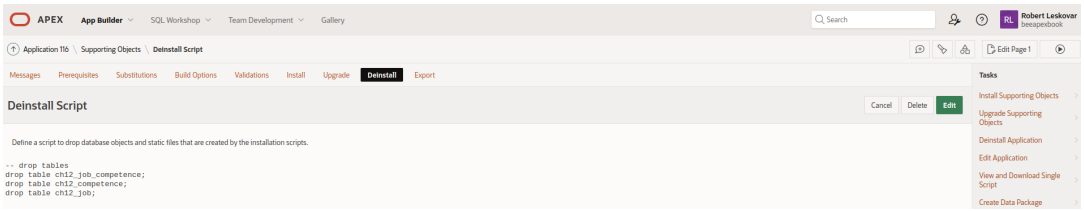
Kliknij przycisk “Apply changes”.

Dla tej aplikacji pakietowej zastosujemy również następujące ustawienie w “Supporting Objects”(Obsługujących obiektach):

- wymagania wstępne (zob.rysunek 12.16)
- zastępowanie ciągów w aplikacji (zob.rysunek 12.17)
- skrypty instalacyjne: ponieważ już przygotowaliśmy skrypty CH12CREATE i CH12INSERT, wykorzystamy ich treść (zob.rysunek 12.18)
- skrypty deinstalacyjne (see Figure 12.19)
- ustawienia komunikatów: komunikat powitalny Żozdział 12: ...", komunikat licencyjny ĆC BY", komunikat o powodzeniu instalacji Śukces!", komunikat o niepowodzeniu instalacji "Błąd", komunikat deinstalacji Aplikacja CH12 odinstalowana"

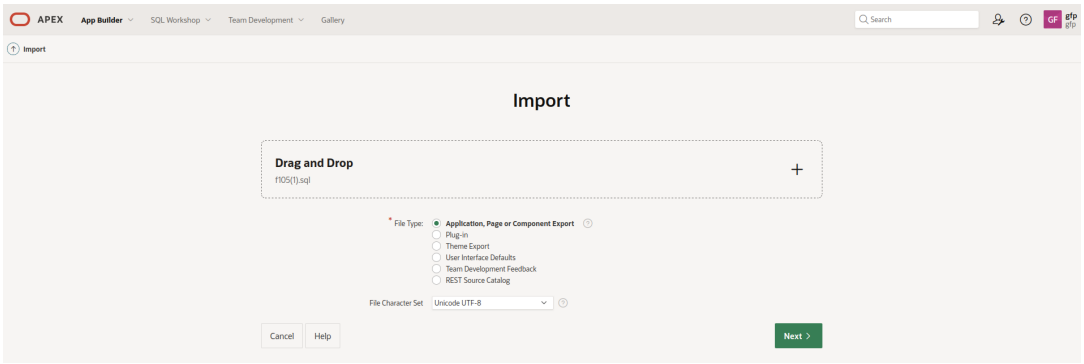


Rysunek 12.18: Ustawianie skryptów instalacyjnych.



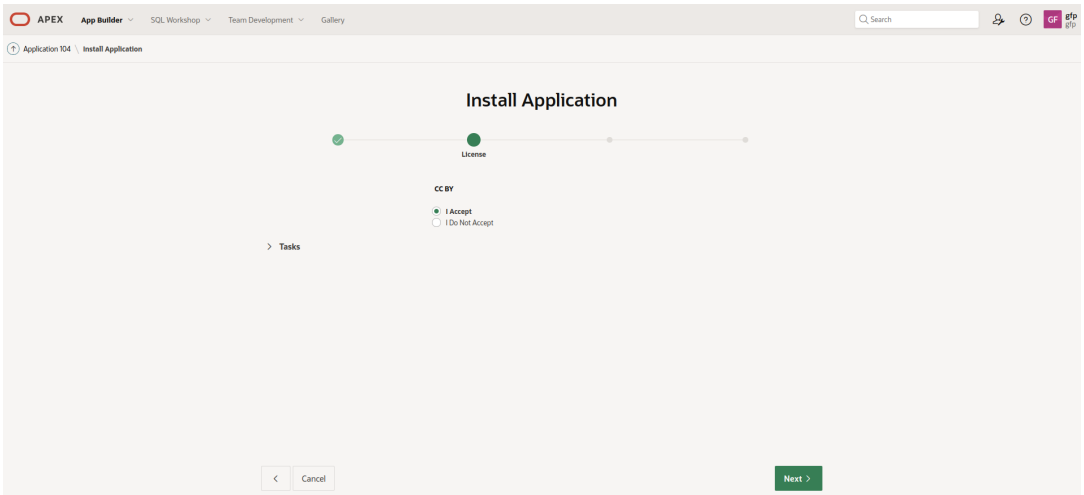
Rysunek 12.19: Ustawianie skryptów deinstalacyjnych.

Teraz wyeksportuj aplikację pakietową za pomocą kreatora eksportu (zob.rysunek 12.1). Możesz zaakceptować wszystkie ustawienia eksportu zaproponowane przez APEX.Zapisz wyeksportowaną aplikację na swoim komputerze i zapamiętaj nazwę pliku. Zaloguj się do nowego obszaru roboczego. Kliknij w kreatora importu w nowym obszarze roboczym, po czym przeciągnij nazwę pliku do obszaru przeciągania i upuszczania(zob.rysunek 12.20).

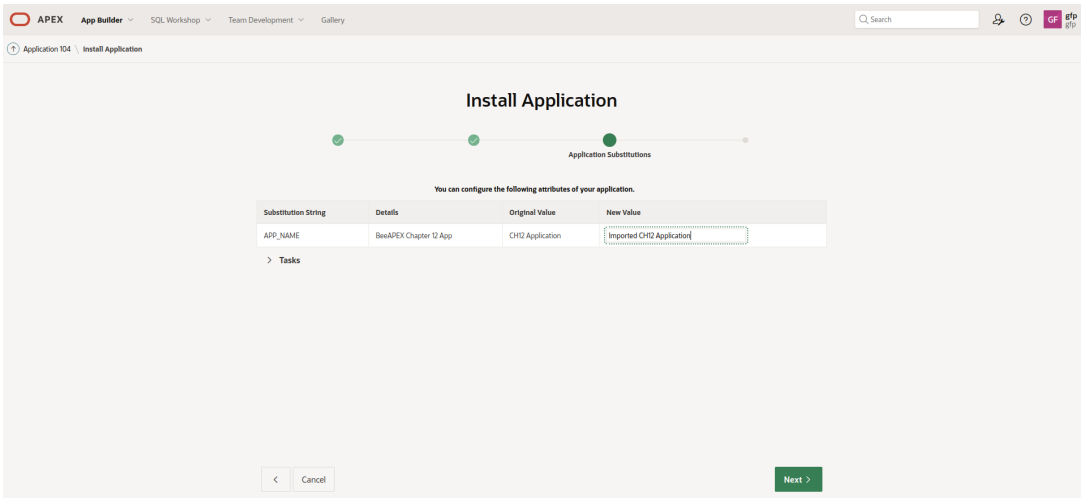


Rysunek 12.20: Importowanie aplikacji w innym obszarze roboczym.

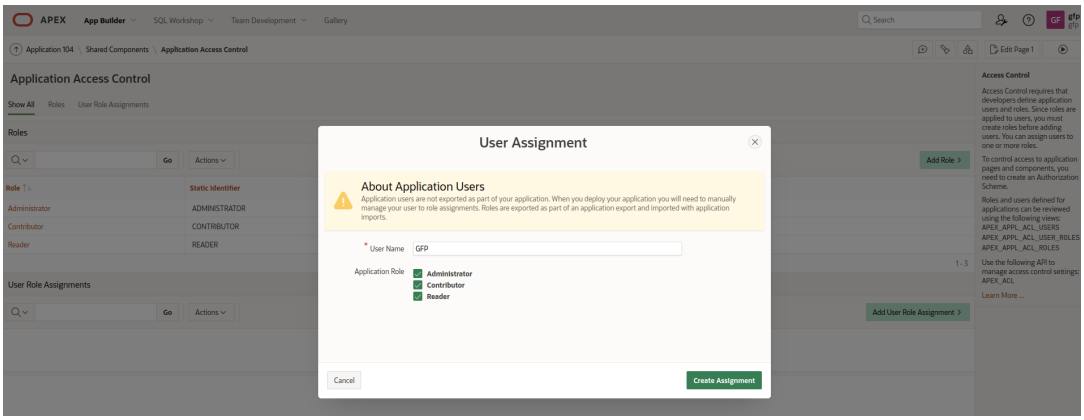
Kliknij przycisk “Next”(aby potwierdzić import pliku) i następnie “Install application”. Potwierdź instalację obiektów pomocniczych za pomocą przycisku “Next”, zaakceptuj licencję (zob.rysunek 12.21), zmień nazwę aplikacji na "Imported CH12 Application"(zob.rysunek 12.22) i kliknij przycisk "Install". Nie uruchamiaj jeszcze aplikacji, ponieważ musisz ustawić uprawnienia dla nowego użytkownika. Użytkownicy i uprawnienia użytkownika nie są eksportowane ze względów bezpieczeństwa. Jedna uwaga: aby przeprowadzić migrację użytkowników i uprawnień, możemy dogłębnie zanurkować w APEX API oraz napisać skrypty instalacyjne i deinstalacyjne. Przejdź do “shared components” (składników współużytkowanych) w importowanej aplikacji. Wybierz kontrolę dostępu do aplikacji i nadaj zalogowanemu użytkownikowi odpowiednie uprawnienia. W naszym przypadku dodamy rolę administratora, współtwórcy i czytelnika (zob.rysunek 12.23). Potwierdź, naciskając przycisk Ądd Assignment” (Dodaj przydział). Teraz uruchom aplikację jako użytkownik z nadanymi uprawnieniami.



Rysunek 12.21: Umowa licencyjna.



Rysunek 12.22: Zmiana nazwy importowanej aplikacji.



Rysunek 12.23: Dodawanie roli do użytkownika.

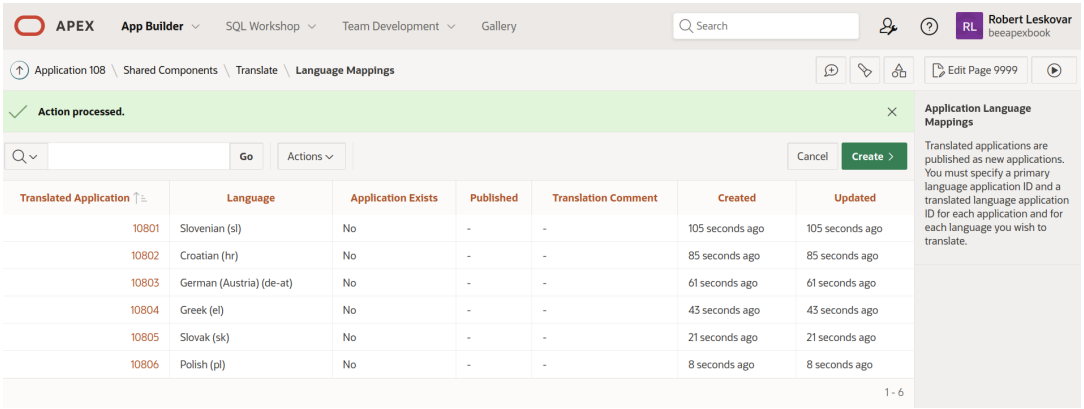
12.4 Aplikacja wielojęzyczna

Ten temat będzie dotyczył tylko jednego aspektu aplikacji wielojęzycznych - napisów na stronach aplikacji, takich jak raporty i formularze. Głównymi przyczynami dostosowania aplikacji do konkretnego języka są: a) użytkownicy posługującymi się tylko innymi językami, b) istnieją wymagania narzucone przez organizacje lub państwa, które wymagają obsługi więcej niż jednego języka. Skala adaptacji może być ograniczona do tłumaczenia etykiet elementów stron, komunikatów aplikacji, wewnętrznych komunikatów APEX, formatów liczb i dat lub tak skomplikowanych, jak tłumaczenie tekstu przechowywanego w tabelach bazy danych (tj. status, poziom, ocena). Konkretny kontekst kulturowy i językowy może sprawić, że czyste tłumaczenie ciągów znaków będzie śmieszne, obraźliwe lub niekompetentne. Dlatego lokalizacja aplikacji to znacznie więcej niż tylko tłumaczenie tekstów. APEX oferuje wiele możliwości zastosowania tłumaczenia (na podstawie głównego języka aplikacji, przeglądarki, preferencji aplikacji, elementu, sesji). W poniższej części przedstawiono najprostsze instrukcje implementowania wielu języków w aplikacji, poprzez zmiane głównego języka aplikacji.

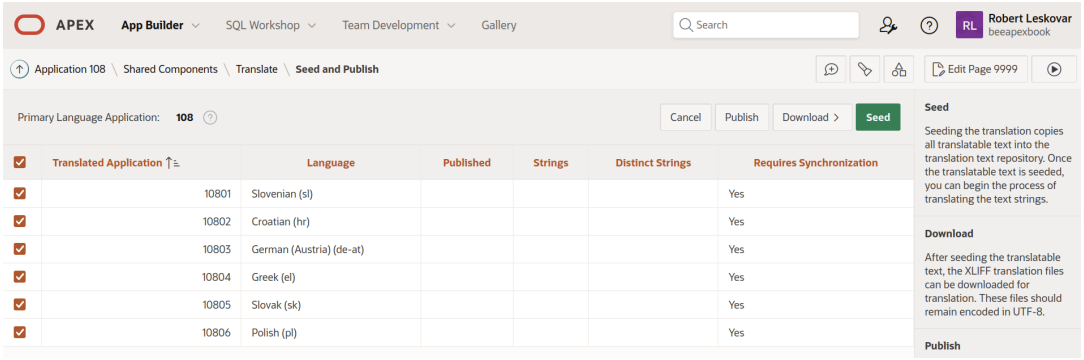
- przejdź do Shared components > Globalization > Application Translations > Define application languages: kliknij przycisk “Create” (Utwórz); dla każdego języka ustaw unikatową liczbę całkowitą i wartość (tj. dodaj dwie cyfry do numeru aplikacji, np. 10801 dla sl; 10802 dla hr; 10803 dla de-at; 10804 dla el; 10805 dla sk; 10806 dla pl). Rysunek 12.24 przedstawia

wszystkie ustawione języki.

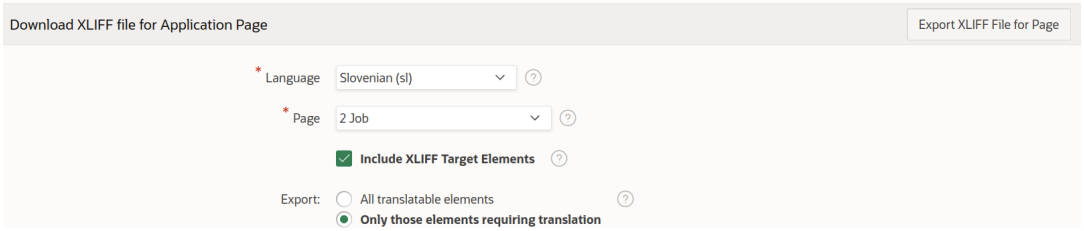
- przejdź do opcji Shared components > Globalization > Application translations > Seed translatable text. Wybierz wszystkie języki, jak pokazano na rysunku 12.25. Naciśnij przycisk Śeed" i poczekaj aż proces się zakończy.
- przejdź do sekcji Shared components > Globalization > Application Translations > Download XLIFF translation files; możesz pobrać wszystkie możliwe do przetłumaczenia elementy lub tylko te elementy, które wymagają tłumaczenia. Przykład na rysunku 12.26 pokazuje eksport strony 2, języka słoweńskiego i elementów wymagających tłumaczenia.
- edytuj wyeksportowany plik w preferowanym edytorze (Notepad++, Kate, Sublime itp.). W celach demonstracyjnych zmieniliśmy tylko kilka "docelowychńapisów: w liniach 48, 52, 56, 60, 72 i 76 (zob.rysunek 12.27). Zapisz zmiany.
- przejdź do menu Shared components > Globalization > Translate application > Apply XLDIFF translation files > Upload Files. Wybierz plik (zob.rysunek 12.28) i kliknij przycisk "Upload". Wybierz odpowiedni język Ąpply to Translation"(zob.rysunek 12.29). Kliknij Ąpply checked". Kliknij "Publish".
- przejdź do opcji Shared components > Globalization > Translate application > Publish translations. Wybierz wszystkie przetłumaczone języki i kliknij przycisk “Publish” (zob.rysunek 12.30).Poczekaj, aż otrzymasz komunikat, że aplikacja została pomyślnie opublikowana.
- przejść do opcji Shared components > Globalization > Globalization attributes. Zmień język podstawowy aplikacji na nowe tłumaczenie (zob.rysunek 12.31).Kliknij przycisk “Apply changes”.



Rysunek 12.24: Określone języki do tłumaczenia.



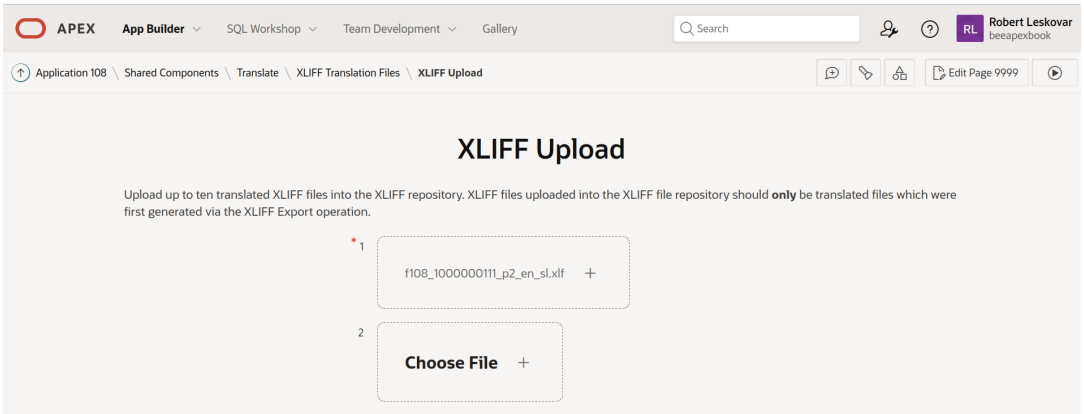
Rysunek 12.25: Tekst do tłumaczenia.



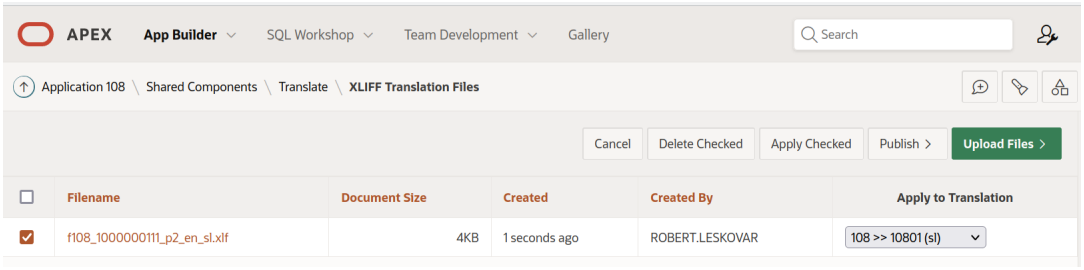
Rysunek 12.26: Eksportowanie ciągów dla konkretnego języka i strony.

```
</trans-unit>
<trans-unit id="S-13-23279527479873043-108">
<source>Create</source>
<target>Ustvari</target>
</trans-unit>
<trans-unit id="S-20-23278046036873035-108">
<source>Ch12 Job</source>
<target>Ch12 Del. mesto</target>
</trans-unit>
<trans-unit id="S-20-23280741017873048-108">
<source>Breadcrumb</source>
<target>Drobtina</target>
</trans-unit>
<trans-unit id="S-143-23278046036873035-108">
<source>Job</source>
<target>Del. mesto</target>
</trans-unit>
<trans-unit id="S-146-23278230062873035-108">
<source>The maximum row count for this report is #MAX_ROW_COUNT# rows.  Please apply a filter to reduce the num
<target>The maximum row count for this report is #MAX_ROW_COUNT# rows.  Please apply a filter to reduce the num
</trans-unit>
<trans-unit id="S-147-23278230062873035-108">
<source>No data found.</source>
<target>No data found.</target>
</trans-unit>
<trans-unit id="S-149-23278547600873039-108">
<source>ID</source>
<target>Šifra</target>
</trans-unit>
<trans-unit id="S-149-23278989319873042-108">
<source>Job Description</source>
<target>Opis delovnega mesta</target>
</trans-unit>
</body>
</file>
</xliff>
```

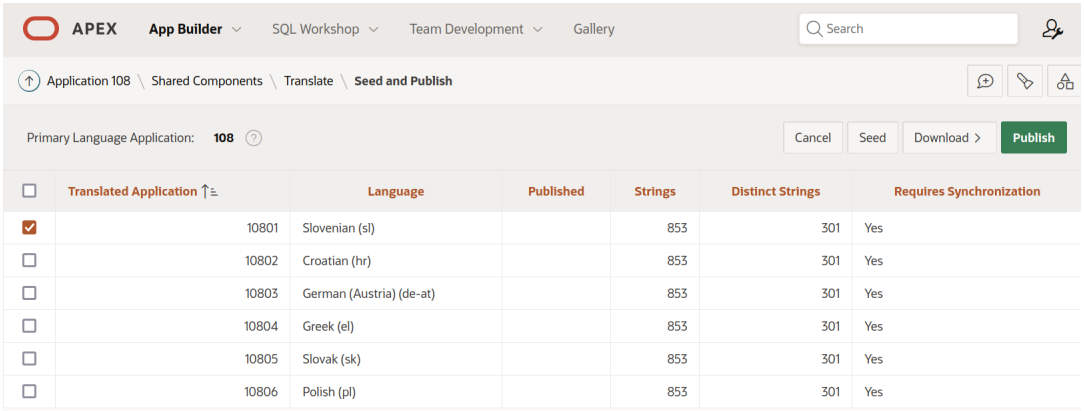
Rysunek 12.27: Tłumaczenie ciągów znaków oznaczonych jako "target"w liniach 48, 52, 56, 60, 72 i 76.



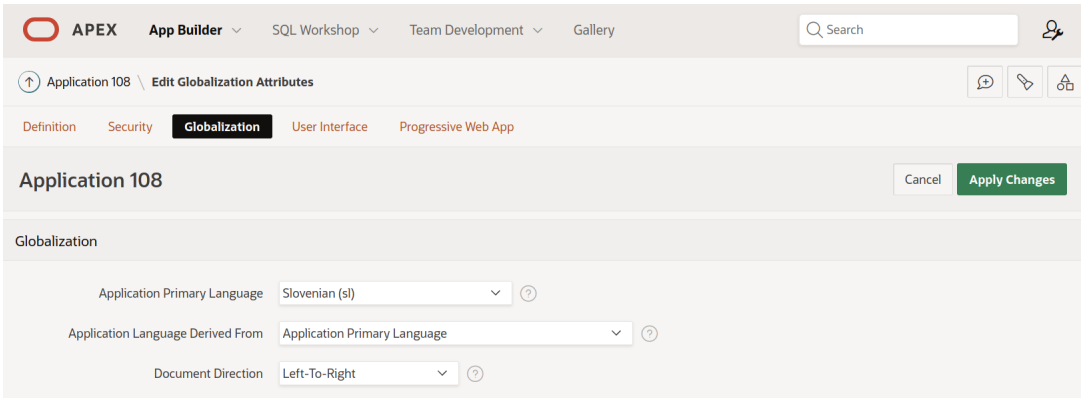
Rysunek 12.28: Wgrywanie plików tłumaczenia XLDIFF.



Rysunek 12.29: Zastosowanie zmian i publikacja

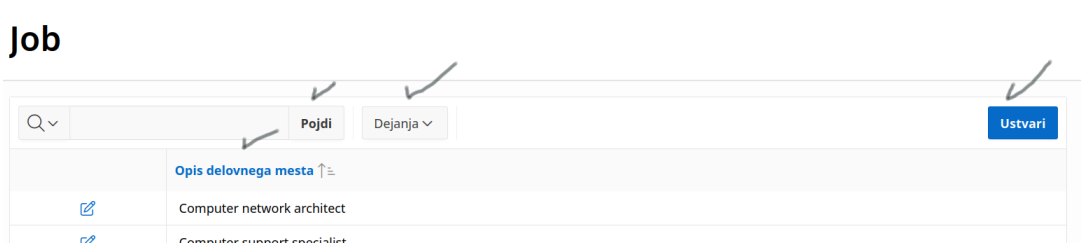


Rysunek 12.30: Ostateczna publikacja i tłumaczenie aplikacji.



Rysunek 12.31: Ustawienie głównego języka aplikacji.

Teraz należy sprawdzić stronę 2 w przetłumaczonej aplikacji (zob.rysunek 12.32).



Rysunek 12.32: Przetłumaczona strona.

Podsumowanie Rozwój aplikacji odbywa się zazwyczaj w środowisku testowym. Przy migracji zaakceptowanej aplikacji do nowej przestrzeni roboczej, aplikacje spakowane są bardzo wygodne, ponieważ skracają czas migracji. Ze względów bezpieczeństwa użytkownicy końcowi i ich role nie są migrowane. Autoryzacja użytkowników w nowym obszarze roboczym to poważne zadanie. Rozdział ten zawierał również podstawowe instrukcje dotyczące aplikacji wielojęzycznych. Należy zauważyć, że w przypadku dużych i krytycznych zastosowań, proces tłumaczenia musi być wspierany przez zdolne narzędzia automatycznego tłumaczenia, a także językowych i kulturowych specjalistów.

12.5 Dodatkowe materiały szkoleniowe

Możesz wyszukać następujące materiały dotyczące szkolenia dodatkowego:

- wyeksportowane aplikacje
- przewodniki wideo

Wszystkie dodatkowe materiały są dostępne na publicznej stronie BeeAPEX . Zaloguj się jako użytkownik gość (nie jest wymagane hasło). Znajdź podręcznik w sekcji “Books”(Książki), skrypty w folderze Part 1 > Chapter12 w sekcji “Scripts” (Skrypty) i przewodniki wideo w “Collection of video guides”. Materiał na krótkie kursy w “Short courses section”.

12.5.1 Eksportowane aplikacje

Istnieją trzy powiązane aplikacje:

- początkowa aplikacja. Proszę sprawdź, czy istnieją tabele zaczynające się od CH12. Jeśli tabele istnieją, możesz je usunąć za pomocą skryptu CH12DROP. Następnie zaimportuj aplikację w plikuCH12_Application_initial.sql. o zaimportowaniu aplikacji musisz zdefiniować tabele. Uruchom skrypt CH12CREATE i opcjonalnie wypełnij tabele za pomocą skryptu CH12INSERT. Przejdź do imported application > Shared Components > Application Access Control. Dodaj “User Role Assignment”(przypisanie roli użytkownika) dla aktualnie zalogowanego użytkownika. Pozwól na wszystkie czynności z rolą Administratora “wybieraj, wstawiaj, aktualizuj i usuwaj“,w roli Współtwórcy oraz tylko “wybieraj” w roli Czytelnika.
- spakowana aplikacja. Sprawź, czy istnieją tabele zaczynające się od CH12. Jeśli tabele istnieją, musisz usunąć je za pomocą skryptu CH12DROP. Następnie należy zaimportuj aplikację z pliku CH12_Application_packaged.sql. Przejdź do imported application > Shared Components > Application Access Control. Dodaj “User Role Assignment”(przypisanie roli użytkownika) dla aktualnie zalogowanego użytkownika. Pozwól na wszystkie czynności z rolą Administratora “wybieraj, wstawiaj, aktualizuj i usuwaj“,w roli Współtwórcy oraz tylko “wybieraj” w roli Czytelnika.
- spakowana aplikacja wielojęzyczna. Sprawdź, czy istnieją tabele zaczynające się, od CH12. Jeśli istnieją, musisz usunąć je skryptem CH12DROP. Następnie należy zaimportuj aplikację w pliku CH12_Application_packaged.sql. Przejdź do imported application > Shared Components > Application Access Control. Dodaj “User Role Assignment”(przypisanie roli użytkownika) dla aktualnie zalogowanego użytkownika. Pozwól na wszystkie czynności z rolą Administratora “wybieraj, wstawiaj, aktualizuj i usuwaj“,w roli Współtwórcy oraz tylko “wybieraj” w roli Czytelnika.

12.5.2 Przewodniki video

Następujące przewodniki wideo są prywatne:

- tworzenie początkowej aplikacji
- importowanie początkowej aplikacji
- kopiowanie początkowej aplikacji do pakietu i tworzenie aplikacji spakowanej
- import aplikacji spakowanej

- importowanie spakowanej aplikacji wielojęzycznej

12.6 Pytania

1. Co to jest aplikacja spakowana i jakie są jej korzyści?
2. Jak wyeksportować spakowaną aplikację, aby uwzględnić definicję obiektu bazy danych i dane dla tabel?
3. Jaka jest rola pliku XLDIFF w tworzeniu aplikacji wielojęzycznej?

12.7 Odpowiedzi

1. Aplikacja spakowana to gotowa aplikacja, którą można zainstalować i skonfigurować w środowisku Oracle APEX. Główną korzyścią aplikacji jest łatwa i płynna migracja, a także natychmiastowa dostępność przygotowanych danych.
2. Aplikacja spakowana jest eksportowana za pomocą kreatora. Można uwzględnić skrypty definiujące obiekty bazy danych (np. tabele) i instruujące wstawianie danych do tabel. Lista skryptów i kolejność wykonywania tych skryptów jest ustawiana w kreatorze. Aplikacja spakowana może również zawierać skrypty deinstalacyjne.
3. Plik XLDIFF jest oznaczony i zapewnia łatwy dostęp do docelowych wartości dla tłumaczonych ciągów tekstowych. Plik ten może zawierać całą aplikację lub tylko określoną stronę w wybranym języku. Plik z przetłumaczony importowany jest do aplikacji. Repozytorium tłumaczeń musi być opublikowane, aby było dostępne dla użytkownika końcowego.

II Constructing application in APEX

13 Intranetowe wiadomości dla pracowników 229
R. Leskovar, U. Rajkovič, A.Baggia . . . . . .

14 GreenDi - Katalog roślin 241
Vjeran Strahonja, Dijana Oreški, Darko Andročec, Ana Kutnjak . . . . . .

15 GreenDi - Autoryzacja i zarządzanie użytkownikami 249
Vjeran Strahonja, Darko Andročec, Ana Kutnjak, Larisa Hrustek . . . . . .

16 System małych innowacji 257
R. Leskovar, U. Rajkovič, A.Baggia . . . . . .

17 Zarządzanie procesami biznesowymi 269
R. Leskovar, U. Rajkovič, A.Baggia . . . . . .

18 GreenDi – wymiana roślin i nasion . . 301
Vjeran Strahonja, Dijana Oreški, Darko Andročec, Ana Kutnjak . . . . . .

19 System zarządzania recenzjami książek 309
A. Kutnjak, L. Hrustek, A. Baggia and R.Leskovar . . .

20 Zestawienie materiałów i kalkulacja kosztów 323
R. Leskovar, U. Rajkovič, A.Baggia . . . . . .

21 Odżywianie i zarządzanie dietą 339
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . .

22 Planowanie godzin pracy biura 359
J. Mańko, M. Sońta, R. Leskovar . . . . . .

23 Sprawa Telco 379
Veronika Šalgová, Jozef Kostolný, Michal Mrena, Michal Kvet, Miroslav Potočár . . . . . .

24 Sprawa wynajmu samochodu 403
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . .

13. Intranetowe wiadomości dla pracowników

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA

13.1 Biznesowy pogląd na sprawe

Średniej wielkości firma zatrudniająca siedemset pracowników jest uznanym graczem w globalnego rynku sprzętu sportowego i rekreacyjnego oraz zaawansowanych kompozytów. Główna siedziba znajduje się w Słowenii, a spółki zależne w USA, Kanadzie, Niemczech i Japonii. Ich roczny przychód wynosi około 90 milionów euro. Większość pracowników znajduje się w Słowenii w jednym miejscu. Firma dzieli się na 4 dywizje, a każdy z nich ma kilka działów. W przeszłości wykorzystywała wewnętrznego półrocznego biuletynu do komunikowania różnych osiągnięć, historii sukcesu, ogłaszania nowych produktów i tym podobnych. Następnie stworzyli (czysty HTML) stronę intranetową do rozpowszechniania ważnych wiadomości. Jednak ta okazja do wzmocnienia zaangażowania pracowników w firmę była krótkotrwała. Idylliczny ogród zamienił się w problem.

13.2 Definicja problemu

Proces publikacji był luźno zdefiniowany. Pracownik IT otrzymywał prośby o opublikowanie wiadomości od wydawców - kierowników działów i dywizji. Prośby były przekazywane drogą mailową, telefonicznie lub osobiście, często z niewielkimi materiałami (tekstem, zdjęciami, wideo). Nie określano daty rozpoczęcia i zakończenia publikacji wiadomości. Częstotliwość prośb o publikację wiadomości wzrosła do około stu wiadomości miesięcznie, a niektóre prośby się pokrywały (dwóch lub więcej kierowników działów miało podobne lub te same tematy). Administrator IT musiał rozwiązywać problemy zarówno między kierownikami, jak i z kierownikami. Często to na administratorze IT spoczywało odpowiedzialność za ustalanie dat rozpoczęcia i zakończenia publikacji wiadomości, co powodowało złe samopoczucie u kierowników i pracownika IT. Zarządzanie statyczną stroną stało się czasochłonne i pełne błędów. Rozwiązanie wymagało zarówno modyfikacji procesu, jak i nowej platformy publikacyjnej.

Modyfikacja procesu i nowa platforma wydawnicza Do opracowania nowej platformy publikacyjnej wybrano środowisko niskiego kodu. Firma korzysta już z bazy danych Oracle, dlatego wybrano Oracle APEX jako środowisko developerskie. Wyróżnia się trzy typy użytkowników platformy. Pierwszy to Administrator, który ma dostęp do wszystkich danych i uprawnień do zarządzania użytkownikami i rolami. Oprócz podstawowych danych pracowników śledzimy również ich zatrudnienie w działach i ich lokalizacjach. Każdy pracownik ma określoną rolę, która

może się zmieniać w czasie. Drugim typem użytkownika jest Wydawca, który nie ma uprawnień do zarządzania danymi pracowników i definiowania uprawnień, ale ma uprawnienia do publikowania wiadomości. Trzecim typem użytkownika jest Widz/Czytelnik wiadomości, który może czytać wiadomości internetowe.

Specyfikacja wymagań W oparciu o zmodyfikowany proces platforma umożliwi użytkownikom:

- Reader/Viewer (Widz/Czytelnik): logowanie, dostęp do pulpitu, przeglądanie wiadomości i załączników do wiadomości
- Publisher (Wydawca): logowanie, dostęp do pulpitu, przeglądanie i publikowanie wiadomości oraz dodawanie załączników do opublikowanych wiadomości
- Administrator: logowanie, dostęp do pulpitu, przeglądanie i publikowanie wiadomości, przeglądanie i zarządzanie działami, rolami, pracownikami, rolami pracowników, zarządzanie aplikacją

13.3 Przypadki użycia

13.3.1 Opis przypadku użycia

Platforma publikacyjna umożliwia dostęp trzem różnych rodzajom pracowników: administratorowi, wydawcy i czytelnikowi, z których każdy ma inne uprawnienia na portalu intranetowym. Wydawca może publikować wiadomości, czytelnik może czytać wiadomości, a administrator może zarządzać platformą. Każdy z tych przypadków użycia wymaga od użytkownika zalogowania się do platformy intranetowej.Patrz tabele 13.1, 13.2 i 13.3,

13.3.2 Opis częściowo ustrukturyzowany

Możemy wyszczególnić trzy różne historie lub przypadki użycia przez użytkowników: publikowanie, czytanie i zarządzanie.

Dodanie załącznika do wiadomości jest w rzeczywistości rozszerzeniem UC opisanym powyżej i możemy opisać to jako nowy przypadek użycia. Ze względu na czytelność przechodzimy do innych dwóch głównych przypadków użycia, jednakże aplikacja będzie miała funkcję dodawania załączników do wiadomości.

13.3.3 Diagram przypadków użycia

Powyższa historia jest przedstawiona na schemacie przypadków użycia (zob.rys 13.1).

Tablica 13.1: Opis przypadku użycia: publikowanie wiadomości internetowych.

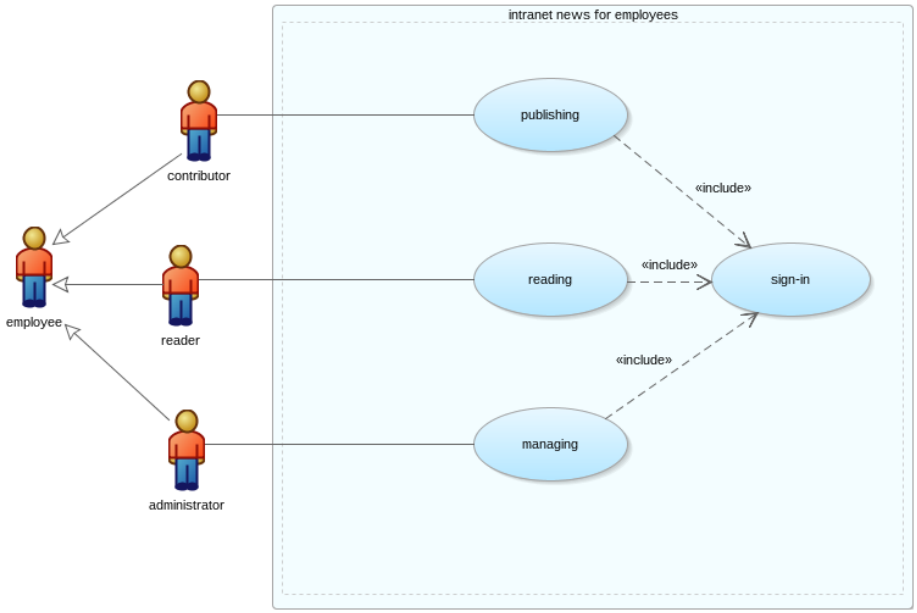
Słowo kluczowe	**Wartość**
ID:	*Ch13-01*
Tytuł:	*Publikowanie wiadomości w portalu intranetowym*
Opis:	*Osoba odpowiedzialna z działu HR, posiadająca zdefiniowaną rolę "wydawcy", korzysta z wewnętrznego portalu APEX, aby opublikować wiadomości. Każda wiadomość ma datę rozpoczęcia i zakończenia. Domyślną datą rozpoczęcia jest dzień publikacji.*
Główny aktor:	*Pracownik z rolą wydawniczą*
Warunki wstępne:	*Pracownik musi być wpisany do tabeli pracowników, a użytkownik musi zostać dodany do roli CH13 Publisher. Dostęp do aplikacji internetowej musi być włączony.*
Warunki końcowe:	*Po pomyślnym opublikowaniu wiadomości wraz z załącznikami, są one dostępne dla innych użytkowników portalu intranetowego.*
Główne:	*Scenariusze*
Scenariusz sukcesu:	1. *Otwórz przeglądarkę internetową i zaloguj się do aplikacji intranetowej.* 2. *Wybierz opcję "Publish news"* 3. *Wprowadź tytuł wiadomości, opis oraz datę zakończenia.* 4. *Potwierdź utworzenie wiadomości.* 5. *Dodaj załącznik(i).* 6. *Wprowadź szczegóły załącznika.* 7. *Potwierdź dodanie załącznika.* 8. *Przejrzyj opublikowaną wiadomość.*
Rozszerzenia:	• *1a. Logowanie nie powiodło się.* • *1a* Rozszerzenie:* • *1a1. Wyświetl komunikat błędu.* • *1a2. Otwórz okno logowania.* • *4a. Błąd typu danych.* • *4a* Rozszerzenie:* • *4a1. Wyświetl komunikat błędu.* • *7a. Błąd typu danych.* • *7a* Rozszerzenie:* • *7a1. Wyświetl komunikat błędu.*
Częstotliwość stosowania:	*Wydawcy publikują około 1000 wiadomości rocznie, co daje średnio 5 dziennie*
Status:	*Ukończony*
Właściciel:	*Pracownik z rolą wydawniczą*
Priorytet:	*umiarkowany*

Tablica 13.2: Opis przypadku użycia: czytanie wiadomości intranetowych

Słowo kluczowe	**Wartość**
ID:	*Ch13-02*
Tytuł:	*Czytaj wiadomości na portalu intranetowym*
Opis:	*Użytkownik intranetu czyta wiadomości*
Główny aktor:	*Użytkownik intranetu*
Warunki wstępne:	*Pracownik musi być wpisany w tabeli pracowników, a użytkownik musi zostać dodany do roli CH13 Viewer. Dostęp do aplikacji internetowej musi być włączony.*
Warunki końcowe:	-
Główne	*Scenariusze*
Scenariusz sukcesu:	1. *Otwórz przeglądarkę internetową i zaloguj się do aplikacji intranetowej* 2. *Wybierz "View" (Wyświetl) wiadomości* 3. *Wybierz pozycję "Download" (Pobierz), aby wyświetlić załącznik*
Rozszerzenia:	• *1a. Logowanie kończy się niepowodzeniem* • *1a* Rozszerz:* • *1a1. Pokaż komunikat o błędzie*
Częstotliwość stosowania:	*Pracownicy codziennie czytają wiadomości.*
Status:	*Ukończony*
Właściciel:	*Użytkownik intranetu*
Priorytet:	*niski*

Tablica 13.3: Opis przypadku użycia: Zarządzanie portalem intranetowym

Słowo kluczowe	**Wartość**
ID:	*Ch13-03*
Tytuł:	*Zarządzanie portalem intranetowym*
Opis:	*Zarządzanie portalem intranetowym*
Główny aktor:	*Administrator portalu intranetowego*
Warunki wstępne:	*Pracownik musi być wpisany do tabeli pracowników, a użytkownik musi zostać dodany do roli Administratora. Dostęp do aplikacji internetowej musi być włączony.*
Warunki końcowe:	*Użytkownicy, role i pracownicy są gotowi do użycia w portalu intranetowym*
Główne	*Scenariusze*
Scenariusz sukcesu:	1. *Otwórz przeglądarkę internetową i zaloguj się do aplikacji internetowej.* 2. *Dodaj i zarządzaj danymi departamentów.* 3. *Dodaj i zarządzaj danymi ról.* 4. *Dodaj i zarządzaj danymi pracowników oraz ich rolami.* 5. *Publikuj wiadomości.* 6. *Przeglądaj wiadomości.* 7. *Zarządzaj administracją.*
Rozszerzenia:	• *1a. Nieudana próba logowania się.* • *1a* Rozwiń:* • *1a1. Wyświetl komunikat błędu.* • *1a2. Otwórz okno logowania się.* • *2a. Błąd typu danych.* • *2a* Rozwiń:* • *2a1. Wyświetl komunikat błędu.* • *3a. Błąd typu danych.* • *3a* Rozwiń:* • *3a1. Wyświetl komunikat błędu.* • *4a. Błąd typu danych.* • *4a* Rozwiń:* • *4a1. Wyświetl komunikat błędu.* • *5a. Błąd typu danych.* • *5a* Rozwiń:* • *5a1. Wyświetl komunikat błędu.*
Częstotliwość stosowania:	*Częstotliwość użycia zależy od nowych pracowników i zmian w systemie uprawnień. Szacunkowo koło 5 razy w tygodniu.*
Status:	*Ukończony*
Właściciel:	*Administrator portalu intranetowego*
Priorytet:	*wysoki*



Rysunek 13.1: Diagram przypadków użycia.

Ponownie, dla uproszczenia, przypadki użycia rozszerzenia (dodawanie załącznika do wiadomości) są pominięte na (zob.rys 13.1).

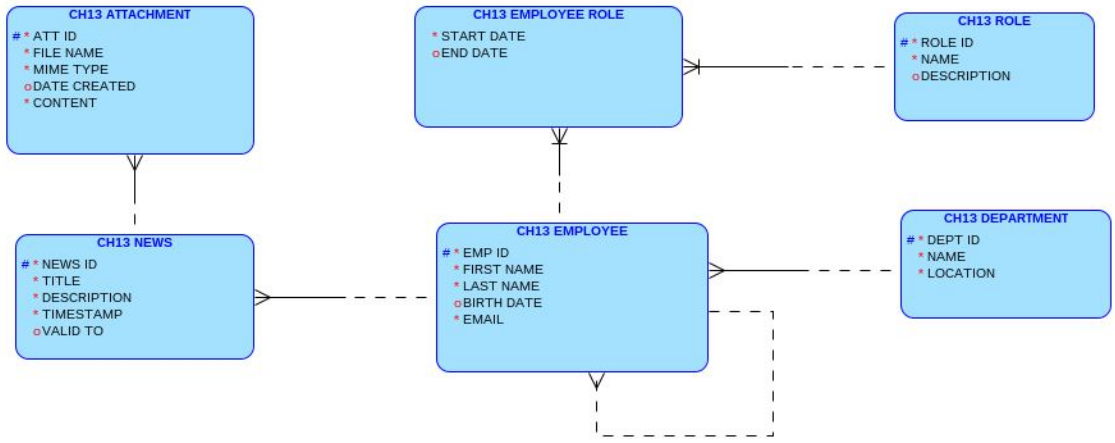
13.4 Model danych

13.4.1 Opis modelu danych

W logicznym modelu danych znajduje się sześć jednostek. Jednostka CH13 DEPARTMENT ma trzy atrybuty: ID, nazwę i lokalizację działu. Każdy dział może mieć wielu pracowników. Podstawowe dane o pracownikach są przechowywane w jednostce CH13 EMPLOYEE: ID, imię i nazwisko, data urodzenia i adres e-mail. Pracownik może być menedżerem. Każdy pracownik ma tylko jednego menedżera, podczas gdy menedżer może zarządzać kilkoma pracownikami. Każdy pracownik może mieć swoją rolę. Role są definiowane przez ID, nazwę i opis. Dla każdego pracownika i każdej roli definiowana jest data rozpoczęcia i zakończenia. Z biegiem czasu pracownik może mieć kilka ról, a także więcej pracowników może być przypisanych do jednej roli. Niemniej jednak jedna definicja roli pracownika nie może zostać przeniesiona na innego pracownika lub rolę. Pracownik może publikować więcej wiadomości w miarę upływu czasu. Każda wiadomość jest identyfikowana przez ID, tytuł, opis, datę rozpoczęcia i datę zakończenia. Każda wiadomość może zawierać wiele załączników. Załącznik jest identyfikowany za pomocą ID, nazwy pliku, typu MIME, daty utworzenia i zawartości.

13.4.2 Logiczny model danych

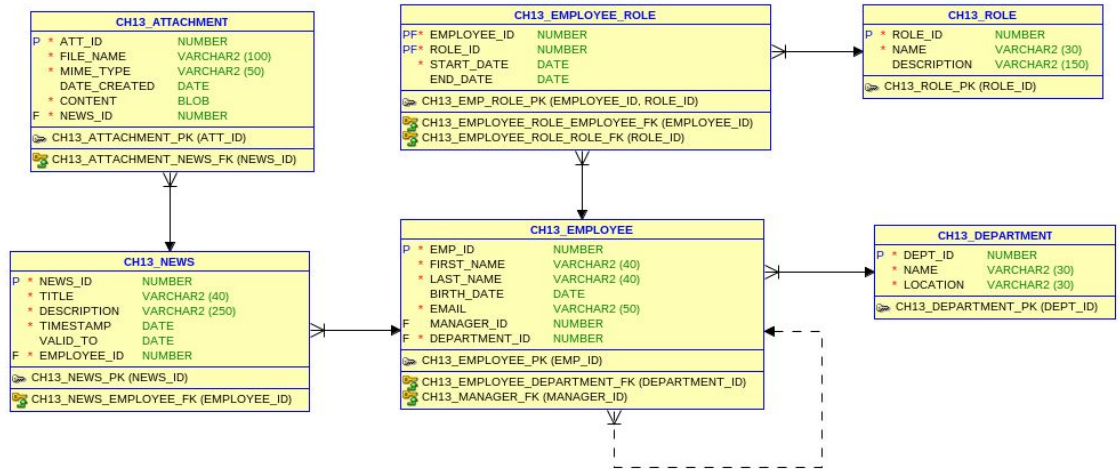
Powyższa historia jest przedstawiona na logicznym modelu danych (zob.rys 13.2).



Rysunek 13.2: Logiczny model danych.

13.4.3 Relacyjny model danych

Automatyczna transformacja z logicznego modelu danych do relacyjnego modelu danych w Oracle SQL Data Modeler jest zapewniana przez funkcję *Engineering to relational*. Wynik przedstawiono na (zob.rys 13.3).



Rysunek 13.3: Relacyjny model danych.

Oracle SQL Data Modeler generuje również skrypt SQL do tworzenia tabel, sekwencji i wyzwalaczy. Wybierz wszystkie tabele w modelu relacyjnym i użyj funkcji *File* > *Export* > *DDL File* aby uzyskać skrypt taki jak ten:

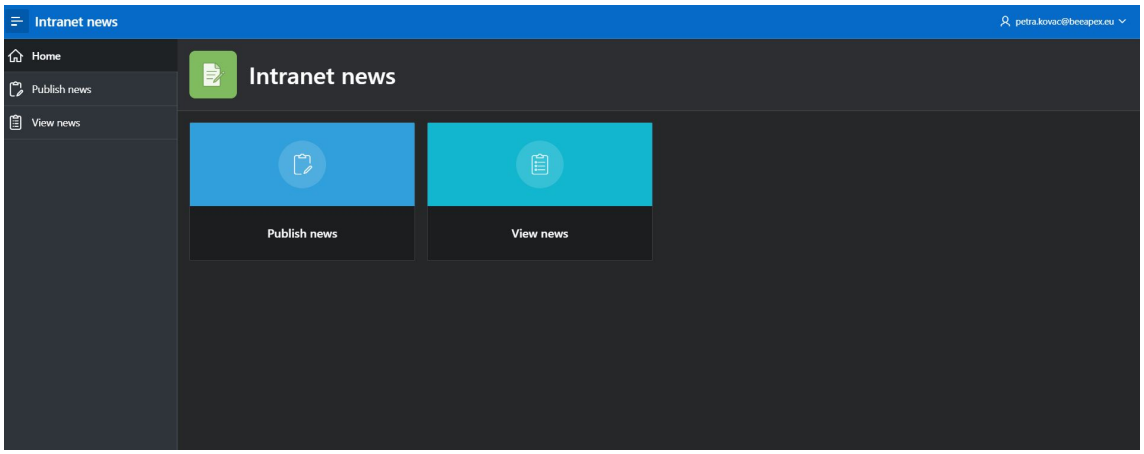
```
CREATE TABLE CH13_NEWS (
NEWS_ID NUMBER NOT NULL
...
);

CREATE TABLE CH13_DEPARTMENT (
DEPT_ID NUMBER NOT NULL
...
);
```

Teraz nadszedł czas na wygenerowanie tabel w bazie danych Oracle. Możemy zaimportować wygenerowany skrypt i wykonać go w APEX.

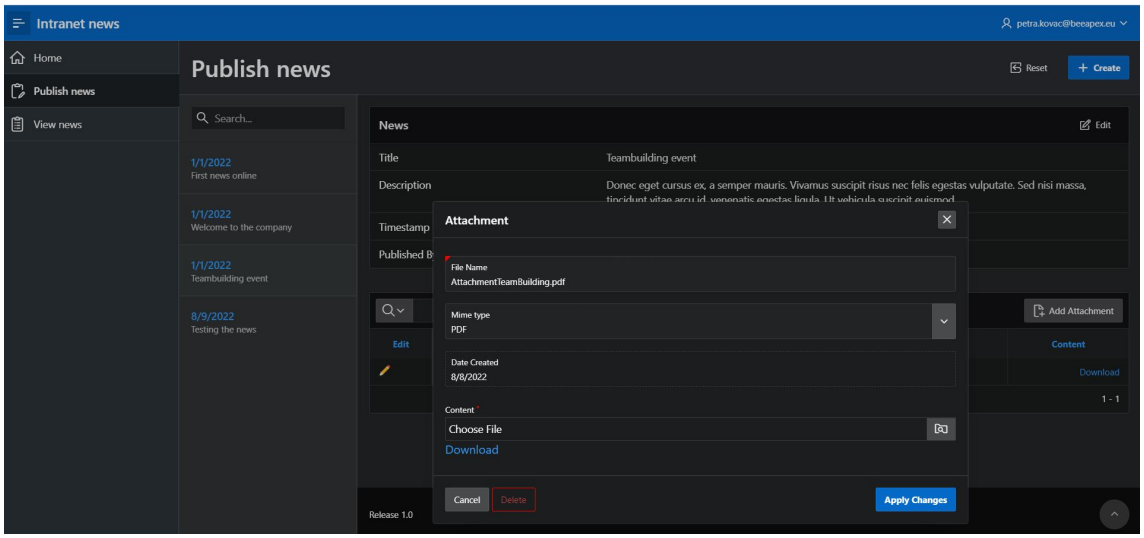
13.5 Interfejsy aplikacji

Poniżej przedstawiam interfejsy aplikacji dla wszystkich trzech ról. Na początek, na Na rysunku 13.4 pokazano panel główny dla użytkownika, którym jest “Publisher” (Wydawca).



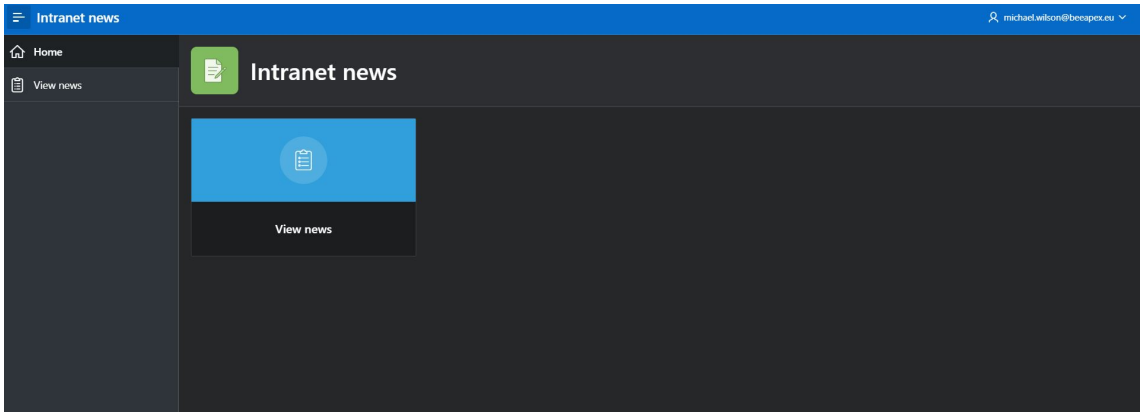
Rysunek 13.4: Pulpit nawigacyjny współautora (wydawcy).

Strone do publikowania wiadomości z załącznikami i ich szczegółami przedstawiono na rysunku 13.5.



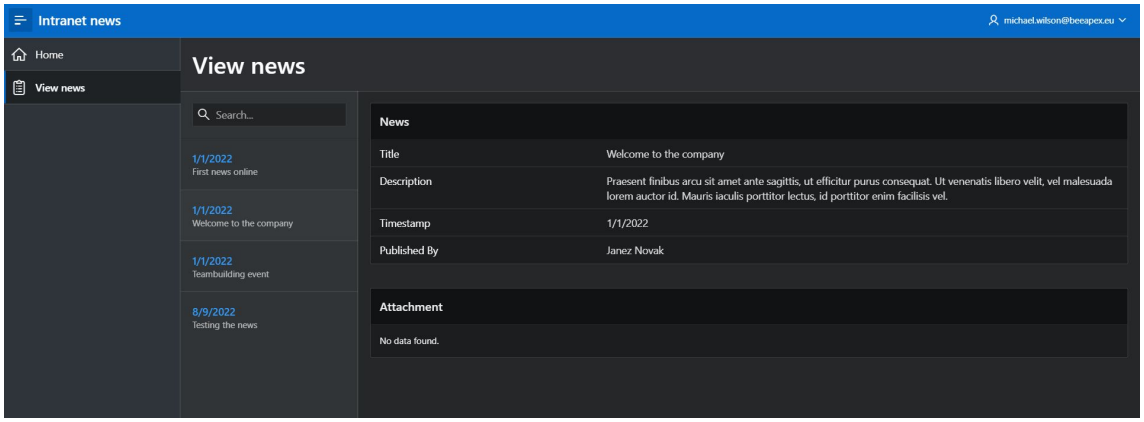
Rysunek 13.5: Publikowanie wiadomości z załącznikami.

Rola “Reader” (Czytelnik) ma najbardziej ograniczony pulpit nawigacyjny. Na razie dostępne są tylko wiadomości Home i View, jak pokazano na rysunku 13.6.



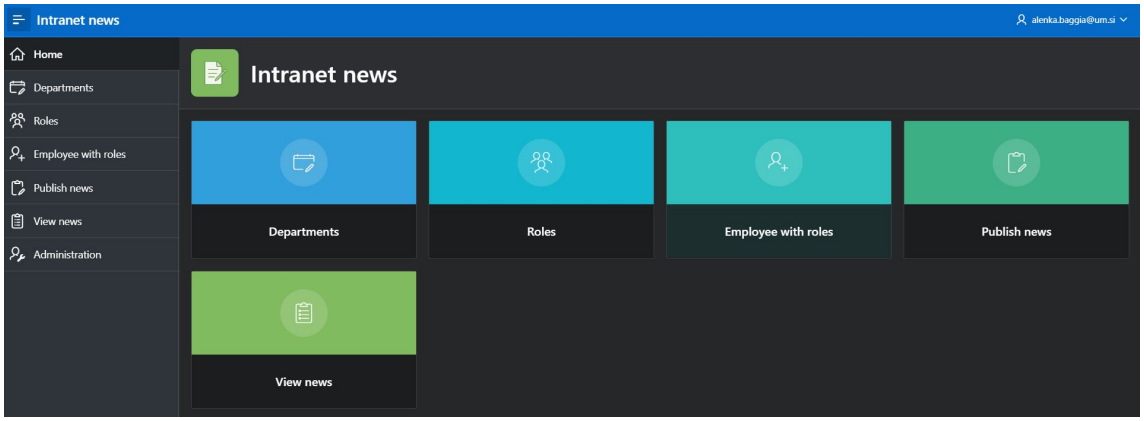
Rysunek 13.6: Pulpit nawigacyjny czytelnika.

Główną cechą przeglądarki czytelnika jest dostęp do wiadomości (patrz rysunek 13.7).



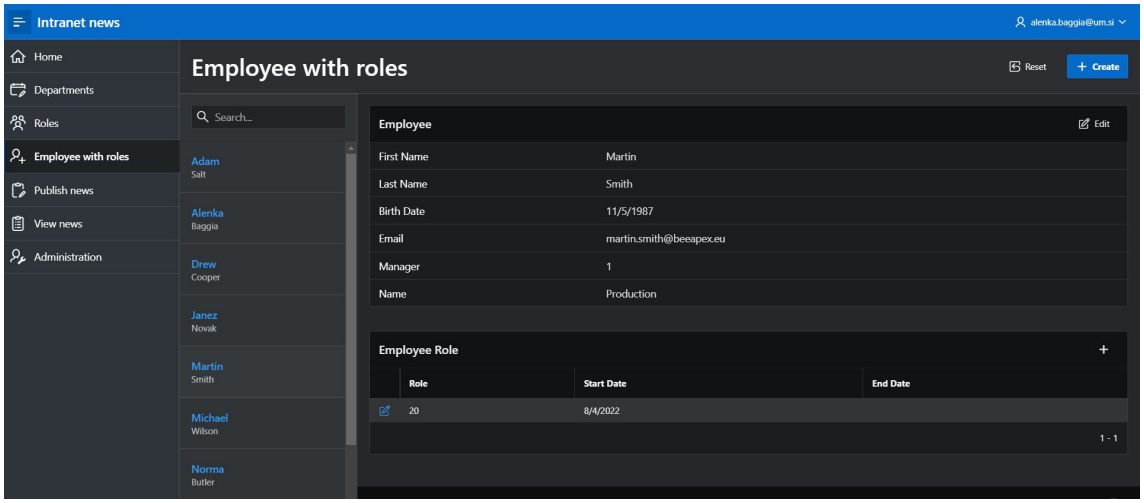
Rysunek 13.7: Dostęp przeglądarki (czytnika) do wiadomości.

Oprócz przeglądania i publikowania wiadomości administrator może zarządzać działami, rolami i pracownikami za pomocą danych ról, jak pokazano na rysunku 13.8. Dostępna jest również strona administracyjna, na której możliwe jest zarządzanie użytkownikami.



Rysunek 13.8: Pulpit nawigacyjny administratora.

Strona do edycji danych ról pracowników przedstawiono na rysunku 13.9. Jest ona dostępna tylko dla administratora portalu intranetowego.



Rysunek 13.9: Edytowanie roli pracownika przez administratora.

13.6 Dodatkowe materiały do nauki

Możesz znaleźć następujące dodatkowe materiały do nauki:

- wyeksportowana aplikacja
- skrypty do tworzenia, usuwania i wstawiania danych
- przewodnik wideo

Wszystkie dodatkowe materiały są dostępne na publicznej stronie BeeAPEX . Zaloguj się jako użytkownik gość (nie jest wymagane hasło). Znajdź podręcznik w sekcji “Books”(Książki), skrypty w folderze Part 1 > Chapter13 w sekcji “Scripts” (Skrypty) i przewodniki wideo w “Collection of video guides”. Materiał na krótkie kursy w “Short courses section”.

13.6.1 Eksportowana aplikacja

Eksportowana aplikacja w formie spakowanej. Podczas instalacji tworzone są tabele i wprowadzane dane. Deinstalacja usuwa wszystkie obiekty bazy danych używane w tej aplikacji.

13.6.2 Przewodniki video

Video guide show every step in application development.

13.7 Pytania

1. Jak zmienić logiczny model danych, aby zaimplementować następujące wymaganie: jeden załącznik może należeć do większej liczby wiadomości.
2. Jak zmieni się relacyjny model danych, jeśli zostanie zrealizowane wymaganie (jeden załącznik może należeć do większej liczby wiadomości)?
3. Jakie role użytkowników są wdrożone w aplikacji rozdziału 13 i dlaczego?

13.8 Odpowiedzi

1. W modelu logicznym danych, powinna zostać wprowadzona relacja jeden-do-wielu w kierunku od załącznika do wiadomości.
2. Para załącznik-wiadomość w modelu logicznym zostałaby przekształcona na trzy tabele relacyjne: załącznik, wiadomość i załącznik_wiadomość. Ostatnia z nich zawierałaby co najmniej dwie kolumny (ID załącznika i wiadomości), obie pełniące rolę kluczy obcych.
3. W rozdziale 13 zaimplementowano następujące role użytkowników: Administrator, Współ-

twórca i Czytelnik. Te role są implementowane, aby umożliwić autoryzowanym użytkownikom wykonywanie ich pracy oraz zapobiec nieautoryzowanym użytkownikom do wykonywania nieautoryzowanych czynności.



14. GreenDi - Katalog roślin

VJERAN STRAHONJA, DIJANA OREŠKI, DARKO ANDROČEC, ANA KUTNJAK

14.1 Biznesowy pogląd na sprawę

Grupa entuzjastów skupionych w organizacji non-profit GreenDi rozpoczęła i realizuje różne inicjatywy związane z ogrodnictwem miejskim, zdrowym odżywianiem, uprawą starych odmian domowych, współpracą z lokalnymi rolnikami itp. Dotychczasz w swojej pracy wykorzystywali oni głównie sieci społecznościowe. Dlatego postanowili uruchomić własną platformę GreenDi, miejsce do wymiany nasion i roślin typu peer-to-peer, w celu promowania różnorodności biologicznej wspierania swoich istniejących inicjatyw oraz umożliwienia tworzenia nowych. Aby opracować i wdrożyć taką platformę, konieczne jest przygotowanie dokładnego zestawu wymagań oraz podstawowych modeli, w tym modelu danych i przypadków użycia.

14.2 Definicja problemu

Otwarta część platformy ma następujące cechy:

- Użytkownicy GreenDi nie są ograniczeni do konkretnej lokalizacji geograficznej, ale mogą pochodzić z różnych części świata. Ponieważ jednak jest to wymiana roślin, oczekuje się tworzenia lokalnych sieci związanych z określonym obszarem geograficznym.
- Platforma jest podzielona na kilka jednostek tematycznych, takich jak tradycyjne i rodzime odmiany i nasiona (warzywa, owoce, zioła lecznicze, rośliny domowe i ozdobne), z myślą o rozszerzeniu na rośliny przemysłowe i inne.
- Podstawą części otwartej jest katalog roślin, z nazwą (lub nazwami) w języku użytkownika, po łacinie i angielsku, klasyfikacją roślin (taksonomia), opisem, siedliskami, uprawą i użytkowaniem, zdjęciami i innymi podstawowymi informacjami.
- Podstawowe informacje o roślinie są dostępne do wyszukiwania i przeglądania bez konieczności rejestracji.
- Katalog roślin GreenDi jest połączony z kilkoma otwartymi stronami internetowymi, bazami danych i zewnętrznymi serwisami, które zawierają informacje związane z roślinami. Używane są adresy URL, hiperłącza i inne rodzaje oznaczonych danych.
- Katalog obejmuje rośliny, które są wymieniane na platformie GreenDi, ale także te, które nie zostały jeszcze wymienione lub uwzględnione w ofercie. Katalog zawiera również linki do dostawców komercyjnych, z którymi członkowie mają dobre doświadczenia i promują uprawę organiczną.

Zarządzanie katalogiem jest opisane następująco:

- Katalogiem roślin zarządzają administratorzy platformy GreenDi .
- Zarządzanie katalogiem obejmuje wprowadzanie i zmienianie informacji o roślinach.
- Niektóre funkcjonalności Zarządzania Katalogami są udostępniane użytkownikom upoważnionym przez administratora (uprawnionego użytkownika).
- Zaawansowane funkcje, są dostępne dla administratorów, obejmują linkowanie do innych stron, które zawierają informacje o roślinie, filmy dotyczące uprawy, zastosowanie roślin itp.
- Wszyscy zarejestrowani użytkownicy (członkowie) mogą, jeśli chcą, podzielić się swoją opinią na temat zakładu, napisać recenzję, dodać więcej informacji itp., Ale nie jest to część tej funkcji.

14.3 Przypadki użycia

14.3.1 Opis przypadku użycia

Otwarta część platformy GreenDi umożliwia każdemu użytkownikowi z dowolnego miejsca przeglądanie danych Katalogu roślin, bez konieczności rejestracji. Jednocześnie rośliny podzielone są na kategorie, co ułatwia użytkownikowi nawigację. Dla każdej rośliny dostępne są takie informacje jak nazwa, taksonomia, opis, siedliska, uprawa i użytkowanie, zdjęcia i inne podstawowe informacje, a także linki do zewnętrznych źródeł danych.

14.3.2 Opis częściowo ustrukturyzowany

Tabela 14.1 przedstawia UC.

14.3.3 Schemat przypadków użycia

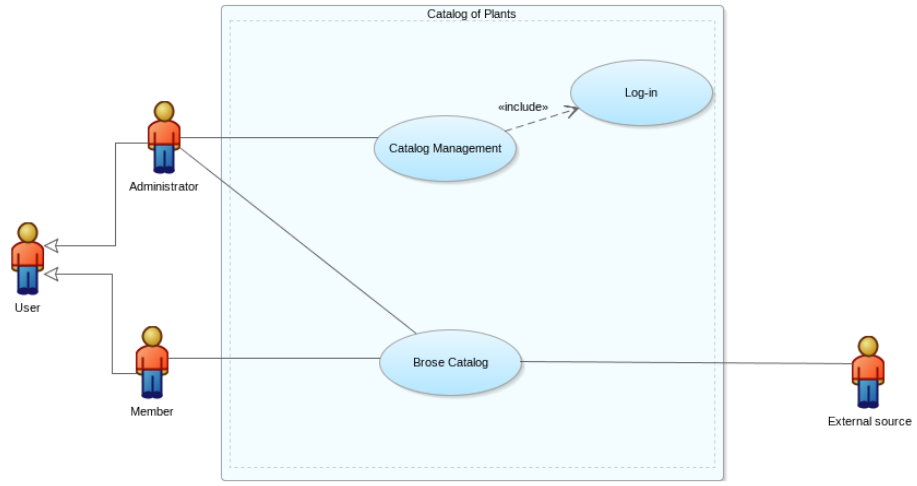
Ta historia jest przedstawiona na diagramie przypadków użycia 14.1.Po prawej stronie diagramu przypadków użycia znajduje się aktor Ęxternal Source"(Źródło Zewnętrzne). Przedstawia on zewnętrzny system, z którego GreenDi Catalog of Plants pobiera informacje o roślinach, nasionach i tym podobnych. Istnieje wiele przykładów takich baz danych roślin, które oferują otwarte usługi lub zbiory danych. Oznacza to, że baza danych lub dane są swobodnie dostępne do użytku, bez żadnych ograniczeń w dostępie lub użytkowaniu. Oto kilka przykładów internetowych baz danych zakładów, które zapewniają otwarte usługi lub otwarte dane:

1. Global Biodiversity Information Facility (GBIF, https://www.gbif.org) zapewnia bezpłatny i otwarty dostęp do danych dotyczących różnorodności biologicznej, w tym danych o roślinach. GBIF umożliwia użytkownikom wyszukiwanie i dostęp do danych z różnych źródeł, w tym zielników, muzeów i instytucji badawczych. Odnosi się do The International Plant Names Index (IPNI), bazy danych nazw i powiązanych podstawowych danych bibliograficznych nasion, roślin, paproci i likofitów (https://www.gbif.org/dataset/046bbc50-cae2-47ff-aa43-729fbf53f7c5).
2. WFO Plant List (https://wfoplantlist.org/plant-list) została uruchomiona w maju 2021 r. i zapewnia przyjazną dla użytkownika, możliwą do cytowania statyczną listę wszystkich gatunków roślin. Cały zbiór danych WFO Plant List można również pobrać jako całość. Dane można również uzyskać za pośrednictwem interfejsu API WFO Plant List.
3. Zestaw danych Flora of Italy można pobrać ze strony internetowej Uniwersytetu w Rzymie i zawiera w wysokiej rozdzielczości zdjęcia ponad 10 000 gatunków roślin we Włoszech.

To tylko kilka przykładów dostępnych zbiorów danych, które zawierają zdjęcia roślin i nasion w Europie, ale istnieje wiele innych zbiorów danych dostarczających podobnych informacji. W tym przykładzie nie zrealizujemy komunikacji ze Źródłem Zewnętrznym, ponieważ wymaga to bardziej zaawansowanej wiedzy.

Tablica 14.1: Opis przypadku użycia: przeglądanie katalogu roślin

Słowo kluczowe	**Wartość**
ID:	*ch14-01*
Tytuł:	*Przeglądaj katalog lokalnych roślin*
Opis:	*Katalog lokalnych roślin jest publicznie dostępny dla wszystkich zainteresowanych. Aplikacja umożliwia wyszukiwanie za pomocą filtrów na wszystkich polach bazy danych.*
Główny aktor:	*Każda osoba*
Warunki wstępne:	*Podstawowe informacje o zakładzie są otwarte do wyszukiwania i przeglądania bez rejestracji.*
Warunki końcowe:	*Jeśli Przeglądaj katalog roślin zostanie wywołany z innego przypadku użycia ("extend"[rozszerzenie]), identyfikator wybranej rośliny zostanie przeniesiony.*
Główny	*Katalog roślin*
Scenariusz sukcesu:	*Pierwszy scenariusz, Katalog roślin:* 1. *Każdy użytkownik na stronie głównej otwiera formularz "Browse catalog of plants" (Przeglądaj katalog roślin), który jest formularzem wyszukiwania, opartym na zasadzie Zapytanie przez przykład (QBE).* 2. *Wyszukiwanie jest możliwe według następujących danych: a) Jednostka tematyczna - lista kodów (wszystkie domyślnie lub niektóre specyficzne nazwy, takie jak warzywa, owoce, zioła lecznicze, rośliny domowe i ozdobne itp.); b) Rodzaj rośliny - hierarchicznie zorganizowana klasyfikacja roślin (domyślnie lub jakaś konkretna nazwa); c) Nazwa (nazwy) w języku użytkownika; d) Nazwa łacińska; e) Nazwa angielska (nazwy)* 3. *Wynik wyszukiwania jest wyświetlany w formie tabeli.* 4. *Możliwe jest sortowanie formularza tabeli według kolumn.* 5. *Wybierając wiersz, można otworzyć formularz ze szczegółami Formularza Rośliny, który zawiera: rodzaj rośliny, jednostkę tematyczną, nazwy lokalne, nazwy angielskie, nazwy łacińskie, opis, siedlisko, uprawę, zastosowanie, zdjęcia, hiperłącza i status.* *Drugi scenariusz, zarządzanie katalogiem:* 1. *Zarządzanie katalogiem może być wykonywane przez aktywnego użytkownika typu Administrator, więc pierwszym krokiem jest procedura logowania (UC: Log In), gdzie sprawdzane są te warunki.* 2. *Zarządzanie Katalogiem obejmuje zmianę lub dodanie informacji o roślinie, która już znajduje się w katalogu, usunięcie tej rośliny (w rzeczywistości ukrycie jej, ponieważ roślina nie jest fizycznie usuwana) lub dodanie nowej rośliny.* 3. *Użytkownik potwierdza lub odrzuca zmiany.*
Rozszerzenia:	• *Formularz wyszukiwania (QBE)* • *Formularz w formie tabeli* • *Szczegóły formy roślinnej*
Częstotliwość stosowania:	*Maksymalnie 1000 wyszukiwań na minutę.*
Status:	*Rozwoju*
Właściciel:	*publiczny, anonimowy użytkownik*
Priorytet:	*wysoki*



Rysunek 14.1: Schemat przypadków użycia - “Catalog of plants”

14.4 Model danych

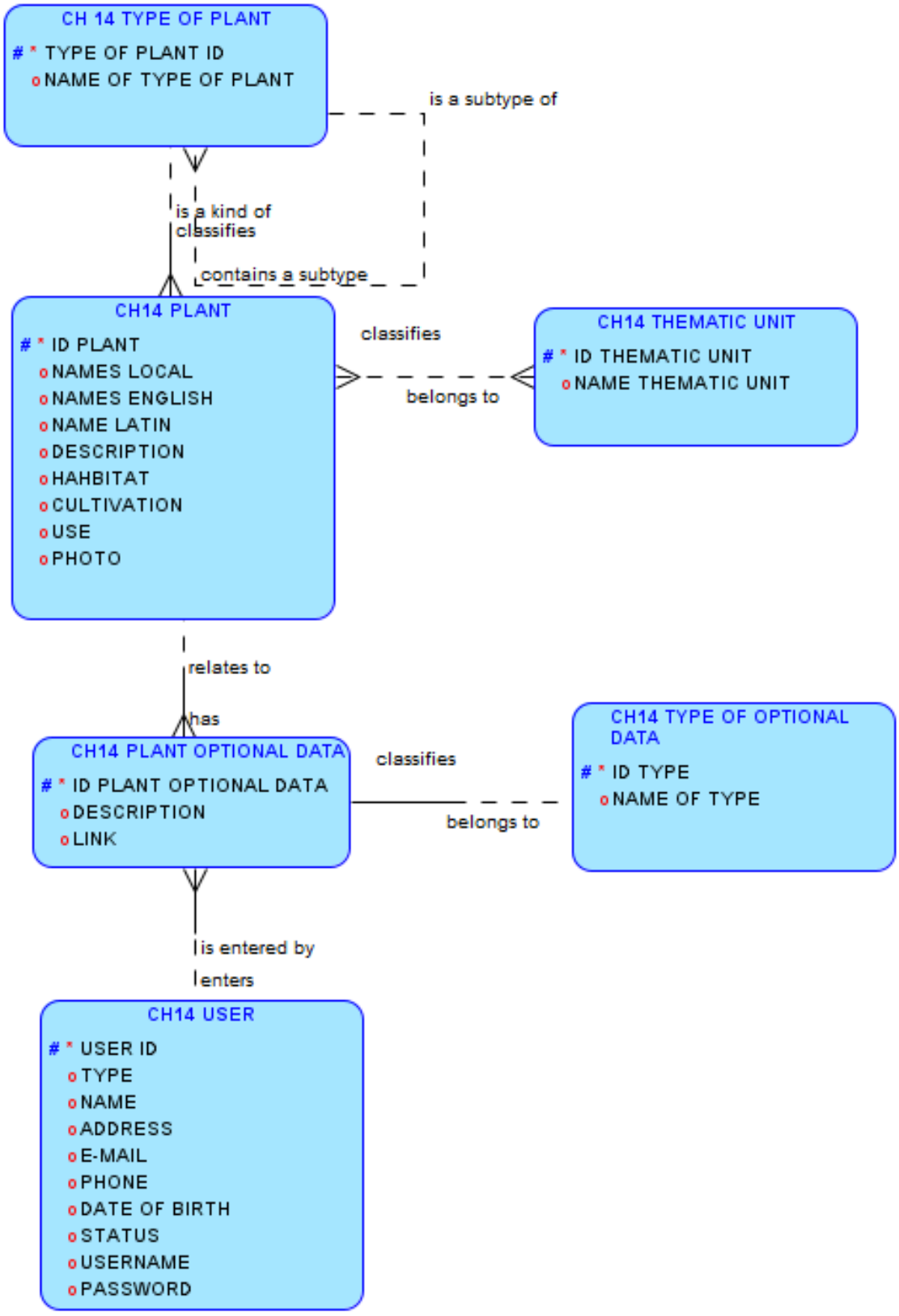
14.4.1 Opis modelu danych

Model danych logicznych (model encji-relacji) zawiera kilka jednostek. Podstawowym z nich jest Roślina, opisana następującymi cechami: Plant _ID (atrybut identyfikujący roślinę); Nazwa lokalna (lokalna nazwa rośliny), Name English (nazwa rośliny w języku angielskim); Name Latin (nazwa rośliny po łacinie); Description (opis rośliny), Habitat (opis siedliska, w którym można znaleźć roślinę); Cultivation (metoda uprawy); Use (wykorzystanie rośliny); Photo (jedno lub więcej zdjęć rośliny); Hyperlinks (linki do zewnętrznych źródeł informacji); Status aktywna roślina będąca przedmiotem wymiany, zarchiwizowana roślina...) Roślina jest hierarchicznie zorganizowanym zestawem kodów. Oznacza to, że każdy rodzaj rośliny może zawierać 0, 1 lub więcej podtypów roślin i może być włączony jako podtyp do 0 lub 1 nadtypu rośliny. Jednostka tematyczna jest również listą kodową, która definiuje odmiany roślin i nasion (warzywa, owoce, zioła lecznicze, rośliny domowe i ozdobne ...). Każda roślina należy do 0, 1 lub więcej jednostek tematycznych. Każda jednostka tematyczna klasyfikuje 0, 1 lub więcej roślin.

Atrybuty jednostek tematycznych to: unit _ID tematyczne (kod lub atrybut identyfikacyjny jednostki tematycznej); Name of thematic unit (jedna lub więcej nazw). Model pokazuje również typ encji Użytkownik. Ponadto do tego modelu danych dodano kilka typów jednostek: : “Plant optional data“(Opcjonalne dane roślin), “Type of optional data”(Typ opcjonalnych danych) oraz “Type of plant”(Typ rośliny) i “Thematic unit”(Jednostka tematyczna). Dane opcjonalne roślin obejmują następujące atrybuty: Plant optional _ID (atrybut identyfikacyjny); Description (treść dodatkowych danych opcjonalnych, tj. opis linku, jeśli istnieje) Link (opcjonalnie). Rodzaj danych opcjonalnych zawiera atrybuty: Type _ID (atrybut identyfikacyjny); Name of type (informacje ogólne, zastosowanie, uprawa, osobista opinia, recenzja...). Każda roślina może mieć 0, 1 lub więcej opcjonalnych danych rośliny. Każde Dane opcjonalne Rośliny odnoszą się do 1 i tylko 1 Rośliny. Każde Dane opcjonalne Rośliny należy do 1 i tylko 1 Typu “optional data”(danych opcjonalnych). Każdy Typ danych opcjonalnych klasyfikuje 0, 1 lub więcej Danych opcjonalnych Rośliny. Dane opcjonalne Rośliny wprowadzane są przez 1 i tylko 1 Użytkownika.

14.4.2 Logiczny model danych

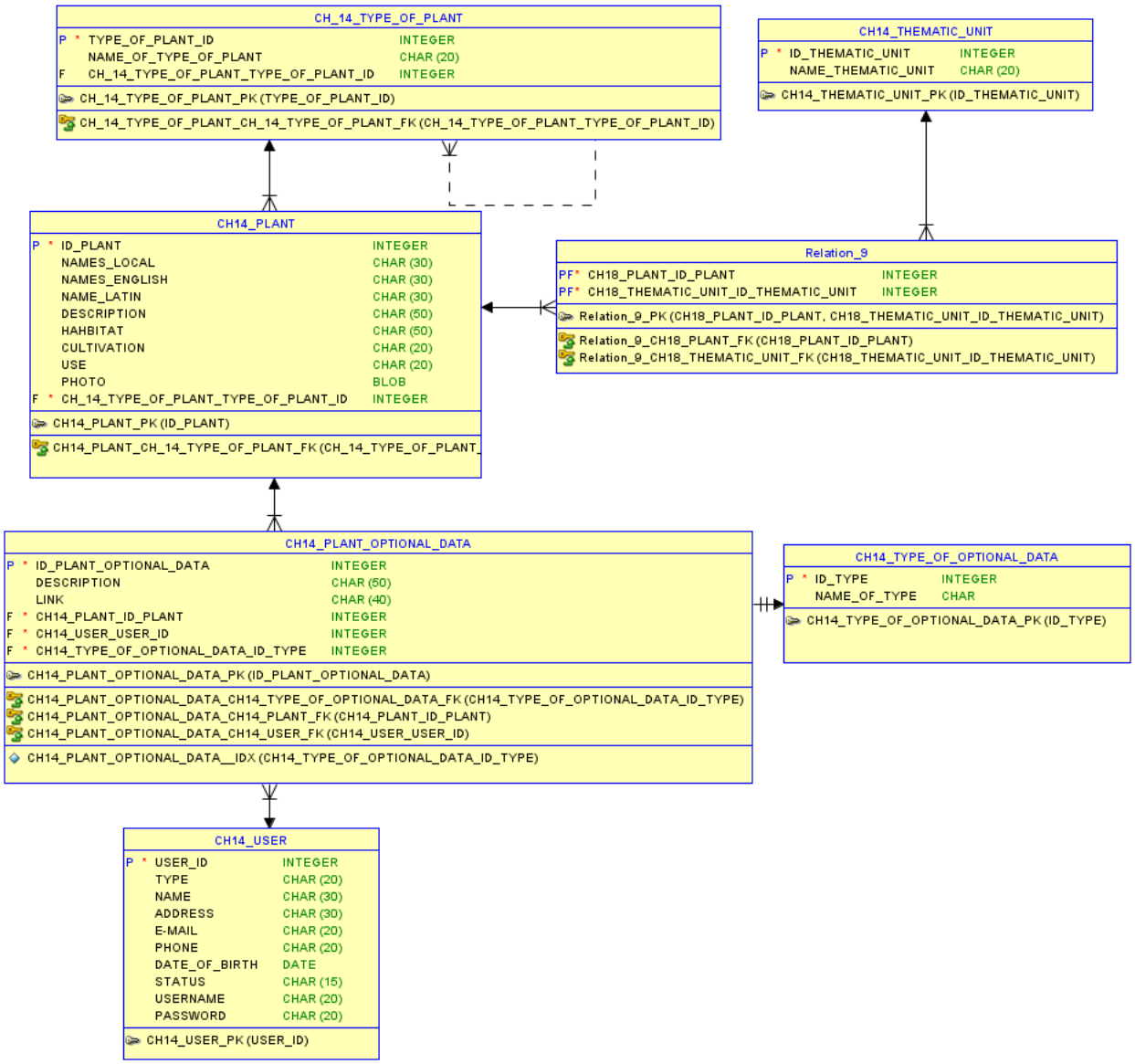
Logiczny model danych przedstawiono na rysunku 14.2.



Rysunek 14.2: Logiczny model danych.

14.4.3 Relacyjny model danych

Automatyczna transformacja z logicznego modelu danych do relacyjnego modelu danych w Oracle SQL Data Modeler jest zapewniana przez funkcję Ęngineering to Relational"(Inżynieria do relacyjnej). Wynik pokazano na rysunku 14.3.

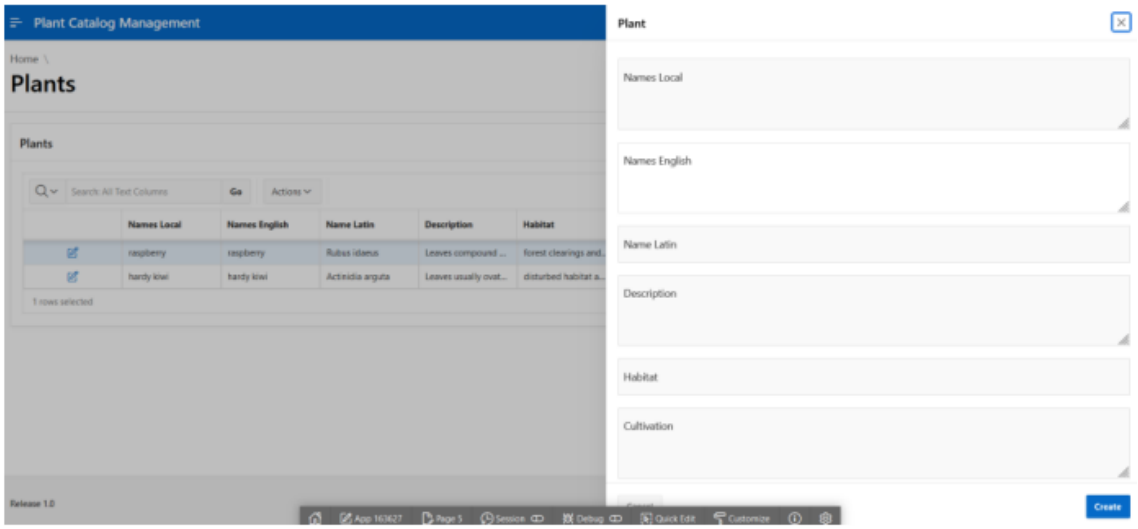


Rysunek 14.3: Relacyjny model danych.

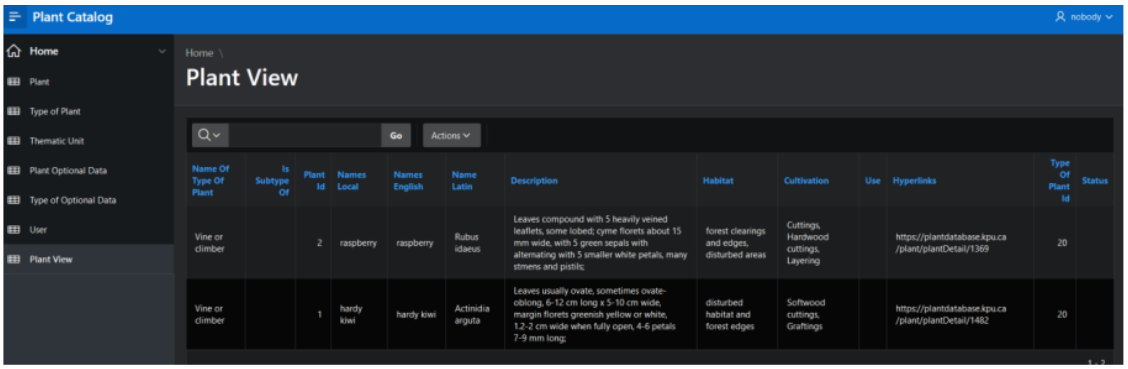
Teraz Oracle SQL Data Modeler wygeneruje skrypt SQL. Wybierz wszystkie tabele w modelu relacyjnym i użyj funkcji *File* > *Export* > *DDL File* , aby uzyskać skrypt. Możemy zaimportować wygenerowany skrypt w APEX i wykonać go.

14.5 Interfejsy aplikacji

Szczegóły interaktywnej siatki dla formularza rośliny przedstawiono na rysunku 14.4.



Rysunek 14.4: Interaktywna siatka dla formularza rośliny.



Rysunek 14.5: Część publiczna – otwarte przeglądanie danych roślin.

Przeglądanie katalogu lokalnych roślin odbywa się za pomocą formularza wyszukiwania.

14.6 Dodatkowe materiały do nauki

Możesz znaleźć następujące dodatkowe materiały do nauki:

- skrypt do tworzenia i wypełniania tabel
- skrypt do usuwania tabel
- wyeksportowaną aplikacje pakietową
- wideo, które demonstruje, jak generować aplikację

Wszystkie dodatkowe materiały są dostępne na publicznej stronie BeeAPEX . Zaloguj się jako użytkownik gość (nie jest wymagane hasło). Znajdź podręcznik w sekcji “Books”(Książki), skrypty w folderze Part 2 > Chapter14 w sekcji “Scripts” (Skrypty) i przewodniki wideo w “Collection of video guides”. Materiał na krótkie kursy w “Short courses section”.

14.6.1 Eksportowana aplikacja

Wyeksportowana aplikacja jest zapakowana i używa tylko dwóch tabel (ch14_type_of_plant i ch14_plants). Instalacja tworzy tabele oraz wypełnia je danymi. De-instalacja usuwa wszystkie obiekty bazy danych używane w tej aplikacji.

Testowana jest aplikacja zapakowana i będzie działać w nowej przestrzeni roboczej, jeśli

spełnione zostaną następujące wymagania:

- dodaj użytkownika APEX przed uruchomieniem aplikacji. Tylko w obszarze roboczym programowania i testowania przejdź do Shared Components > Application Access Control > Add User Role Assignment; wprowadź użytkownika APEX i ustaw te role użytkownika "Administrator, Contributor and Reader"(Administrator, Współautor i Czytelnik). W przypadku wdrożenia w produkcji skonsultuj się z osobą o odpowiednich umiejętnościach.

Jeśli użytkownik nie otrzyma odpowiedniej roli, to zaimportowana aplikacja ulegnie awarii. Konieczne będzie wyczyszczenie plików cookie w przeglądarkach internetowych (np. Firefox: Settings > Cookies and Site Data > Manage Data) po awarii aplikacji z powodu niespełnionych wymagań.

14.6.2 Przewodniki wideo

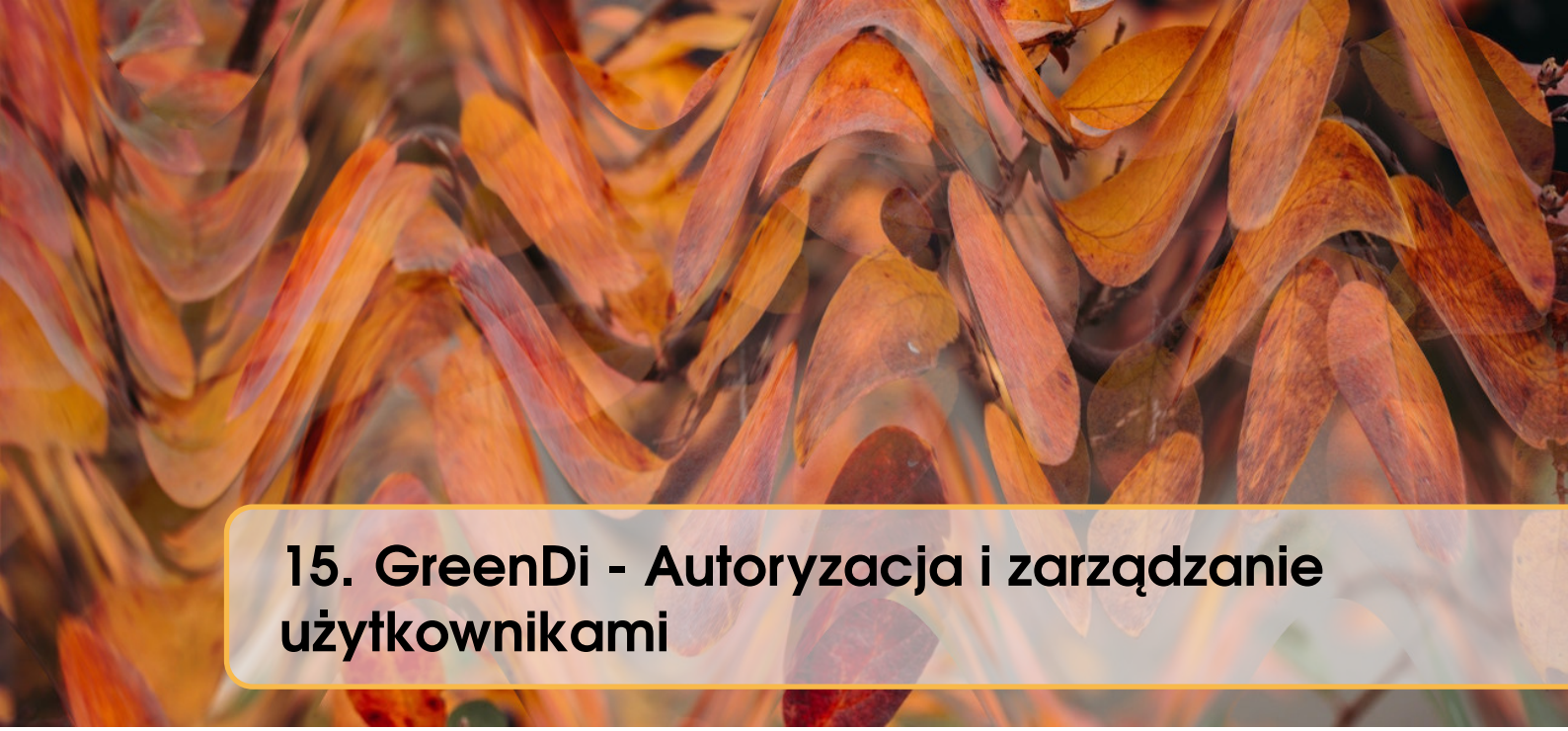
Przewodnik wideo pokazuje wszystkie kroki w tworzeniu aplikacji

14.7 Pytania

1. Zbadaj, w jaki sposób można zaimplementować komunikację ze źródłem zewnętrznym lub inną bazą danych lub systemem zawierającym dane o roślinach.
2. Wymień kilka formatów, w jakich przechowywane są obrazy roślin i nasion.
3. W jakim typie danych zapisywane są obrazy, filmy i podobne obiekty w bazie danych Oracle?

14.8 Odpowiedzi

1. Zasadniczo istnieją dwa sposoby realizacji komunikacji z takim zewnętrznym źródłem. Jednym ze sposobów jest pobranie zbioru danych, czyli kompletnej bazy danych do Katalogu roślin. Innym sposobem jest połączenie za pomocą interfejsu programowania aplikacji (API). Jest to zestaw protokołów, narzędzi i standardów, który określa, w jaki sposób składniki oprogramowania powinny współdziałać i komunikować się ze sobą. Interfejsy API umożliwiają różnym systemom oprogramowania komunikowanie się i udostępnianie danych w ustandaryzowany sposób.
2. Niektóre z najpopularniejszych formatów to:
   a. JPEG (Joint Photographic Experts Group) - jest to popularny format zdjęć cyfrowych ze względu na możliwości kompresji, które pozwalają na mniejsze rozmiary plików bez znacznej utraty jakości. Obsługuje miliony kolorów i może obsługiwać obrazy o wysokiej rozdzielczości.
   b. BMP (Bitmap) - ten format jest powszechnie używany w systemach operacyjnych Windows i może obsługiwać zarówno obrazy w skali szarości, jak i kolorowe. Nie obsługuje kompresji, więc pliki mogą być duże, ale jest to dobra opcja do przechowywania obrazów wysokiej jakości.
   c. PNG (Portable Network Graphics) - ten format jest znany z bezstratnej kompresji, co oznacza, że jakość obrazu nie jest zagrożona po zmniejszeniu rozmiaru pliku. Obsługuje przezroczystość i jest często używany do grafiki internetowej i sztuki cyfrowej.
   d. TIFF (Tagged Image File Format) - ten format jest powszechnie używany do przechowywania wysokiej jakości obrazów, takich jak te używane w druku. Obsługuje kompresję bezstratną, wielowarstwowość i może obsługiwać zarówno obrazy w skali szarości, jak i kolorowe.
3. Obrazy, filmy i podobne obiekty są zwykle przechowywane w bazie danych Oracle przy użyciu typu danych BLOB (Binary Large Object). BLOB mogą przechowywać dane binarne, takie jak obrazy i filmy, o rozmiarze do 4 gigabajtów. Ten typ danych jest idealny do przechowywania dużych obiektów, takich jak pliki multimedialne, ponieważ pozwala na wydajne przechowywanie i pobieranie danych binarnych.



15. GreenDi - Autoryzacja i zarządzanie użytkownikami

VJERAN STRAHONJA, DARKO ANDROČEC, ANA KUTNJAK, LARISA HRUSTEK

15.1 Biznesowy pogląd na sprawę

Krótki przegląd biznesowy platformy GreenDi opisano w rozdziale 14. Skupia się na otwartej części platformy GreenDi. Zaawansowane funkcje są możliwe tylko dla zarejestrowanych użytkowników (członków i administratorów) Konieczne jest umożliwienie funkcji rejestracji użytkowników, logowania zarejestrowanych użytkowników do systemu z uwierzytelnianiem i autoryzacją do wykonywania określonych działań. Niezbędne funkcjonalności obejmują zawieszenie użytkownika pod pewnymi warunkami i monitorowanie historii danych o zarejestrowanym użytkowniku.

15.2 Definicja problemu

Poniższe zdania opisują problem autoryzacji i zarządzania użytkownikami:

- Aby uzyskać głębszy wgląd w informacje o roślinach i innych zaawansowanych funkcjonalnościach konieczne jest zalogowanie się do platformy, czyli jest to możliwe tylko dla zarejestrowanych użytkowników (członków).
- Logowanie jest wspólne dla wszystkich funkcjonalności i w zasadzie musi być jednym logowaniem.
- Podstawowym mechanizmem uwierzytelniania użytkownika jest nazwa użytkownika i hasło.
- Informacja o bieżącym haśle nie jest widoczna dla administratora ani żadnego innego użytkownika.
- Proces rejestracji obejmuje wprowadzenie podstawowych wymaganych informacji o użytkowniku, takich jak imię i nazwisko, adres, adres e-mail, telefon, nazwa użytkownika i hasło.
- Opcjonalne dane to zainteresowania użytkownika, data urodzenia itp.
- Po udanej rejestracji nowy użytkownik automatycznie otrzymuje domyślny typ “Member"(Członka) i status “Active” (Aktywny). Zmiana statusu na Administrator i odwrotnie, a także typ może być wykonana tylko przez innego aktywnego administratora.
- Po rejestracji członek akceptuje warunki użytkowania. Jeśli je naruszy, administratorzy zawieszają członkostwo, co również jest opisane w regulaminie.
- Regularna rejestracja i usunięcie członkostwa jest dobrowolne i wykonywane przez użytkownika.
- Każda zmiana jakichkolwiek danych użytkownika jest rejestrowana w formie dziennika.

Administrator może przejrzeć historię zmian każdego użytkownika, z wyjątkiem hasła.
- Na życzenie użytkownika system umożliwia resetowanie hasła, w taki sposób, że administrator wysyła użytkownikowi hasło tymczasowe za pośrednictwem poczty elektronicznej lub SMS.
- Funkcjonalność opcjonalna: Oprócz nazwy użytkownika i hasła uwierzytelnianie jest opcjonalnie możliwe za pomocą Fb i Google Sign-In lub innych usług. Na przykład aplikacja internetowa, która korzysta z protokołu OAuth 2.0 w celu uzyskania dostępu do interfejsów API Google, musi mieć poświadczenia autoryzacji identyfikujące aplikację na serwerze OAuth 2.0 firmy Google.

15.3 Przypadki użycia

15.3.1 Opis przypadku użycia

Autoryzacja i Zarządzanie Użytkownikami są realizowane poprzez przypadki użycia (patrz 15.1). Każda osoba może rozpocząć procedurę Logowania. Podobnie, przypadek użycia Loginu jest uwzględniony w tych przypadkach użycia platformy GreenDi , z którymi mogą komunikować się tylko zarejestrowani użytkownicy. Jeśli użytkownik nie jest zarejestrowany, jest kierowany do przypadku użycia Rejestracji użytkownika, który jest zawarty w Logowaniu. Podstawowa funkcjonalność tego przypadku użycia jest wprowadzanie danych użytkownika, w tym nazw użytkowników i haseł. Rejestrację Użytkownika można rozpocząć z poziomu Logowania również w przypadku, gdy użytkownik chce zmienić swoje dane. Zarządzanie użytkownikami to przypadek użycia, z którym może komunikować się tylko zarejestrowany użytkownik typu administratora. Funkcjonalność tego przypadku użycia pozwala administratorowi na zmianę statusu i typu dowolnego innego zarejestrowanego użytkownika, w tym zawieszenie go. Użytkownik może zmienić hasło, to znaczy system generuje tymczasowe hasło na żądanie, które trwa krótko, i wysyła je do użytkownika. Uwierzytelnianie zewnętrzne to dodatkowa funkcja, która powinna umożliwić autoryzacje przy użyciu Fb i Google Sign-In lub innych usług. Na przykład aplikacja internetowa, która korzysta z protokołu OAuth 2.0 w celu uzyskania dostępu do interfejsów API Google, musi mieć poświadczenia autoryzacji identyfikujące aplikację na serwerze OAuth 2.0 firmy Google. Więcej informacji na temat używania protokołu OAuth 2.0 do uzyskiwania dostępu do interfejsów API Google można znaleźć na stronie https://developers.google.com/identity/protocols/oauth2

15.3.2 Opis częściowo ustrukturyzowany

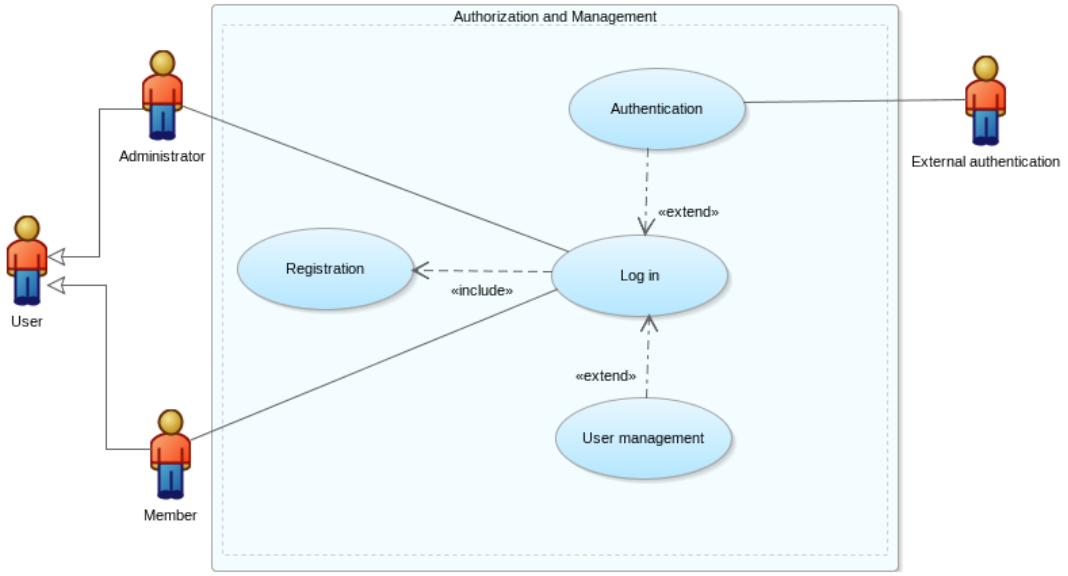
Komentarz RL, 20230711: Ten opis UC powinien zostać poprawiony. W obecnym stanie jest to mieszanka wymagań i przypadków użycia. Tabla 15.1 przedstawia UC.

15.3.3 Diagram przypadków użycia

Opisana historia jest przedstawiona na schemacie przypadków użycia 15.1.

Tablica 15.1: Opis przypadku użycia: autoryzacja i zarządzanie użytkownikami.

Słowo kluczowe	**Wartość**
ID:	*ch15-01*
Tytuł:	*Autoryzacja użytkowników i zarządzanie nimi*
Opis:	*Autoryzowanie użytkowników i zarządzanie nimi.*
Główny aktor:	*Dowolna osoba, Członek, Administrator*
Warunki wstępne:	*Logowanie zazwyczaj rozpoczyna się automatycznie od innych przypadków użycia, w których użytkownik jest ustawiony jako członek lub administrator.*
Warunki końcowe:	*Wynikowa sytuacja jest taka, że użytkownik nie jest autoryzowany lub że jest autoryzowany, ale jego status (oczekujący, aktywny ...) i typ (członek, administrator) są znane.*
Główny	-
Scenariusz sukcesu:	1. Zaawansowane funkcje platformy GreenDi wymagają zalogowania się do systemu za pomocą formularza logowania. a. Przy pierwszym logowaniu nowy użytkownik musi zarejestrować się za pomocą Formularza Rejestracji Użytkownika, dołączonego do przypadku użycia logowania. Proces rejestracji obejmuje wprowadzenie obowiązkowych informacji o użytkowniku, takich jak imię i nazwisko, adres, adres e-mail i telefon, a także informacji opcjonalnych, takich jak zainteresowania użytkownika i data urodzenia. b.Użytkownik ustala nazwę użytkownika i hasło (są one zapisywane w specjalnej tabeli w bazie danych, oddzielane od innych danych użytkownika lub zaszyfrowane). c.Użytkownik akceptuje regulamin i potwierdza wprowadzone dane. d.Domyślnym typem użytkownika jest Członek, a domyślnym stanem jest Aktywny. 2. Dane Użytkownika mogą być zmieniane przez samego użytkownika przy każdym kolejnym Logowaniu do systemu, poprzez Formularz Rejestracyjny Użytkownika. 3. Resetowanie hasła jest inicjowane przez użytkownika z formularza logowania, dzięki czemu wiadomość z tymczasowym hasłem, które trwa 15 minut, jest wysyłana na adres e-mail użytkownika. 4. Każda zmiana jakichkolwiek danych użytkownika jest rejestrowana w formie dziennika w jednostce danych Historia użytkownika. Administrator może przeglądać historię zmian każdego użytkownika, z wyjątkiem hasła. 5. Funkcjonalność przypadku użycia Zarządzanie użytkownikami umożliwia użytkownikowi typu Administrator i w stanie Aktywny zmianę typu i statusu innego (Administrator na Członka i odwrotnie, a także status Aktywny - Wstrzymany - Nieaktywny). 6. Istnieje widok tabeli użytkowników z możliwością wyszukiwania (Formularz Tabeli Użytkowników). Na tym formularzu administrator może wybrać szczegóły dowolnego użytkownika, na podobnym formularzu jak Formularz Rejestracji Użytkownika (Formularz Szczegółów Użytkownika), zmieniać status i typ użytkownika, resetować hasło oraz otwierać Formularz Historii Użytkownika, który przedstawia zmiany w danych dotyczących wybranego użytkownika w formie tabelarycznej. 7. Każde kolejne Logowanie zarejestrowanego użytkownika inicjuje sprawdzenie autoryzacji Użytkownika, które sprawdza nazwę użytkownika i hasło, status użytkownika oraz to, czy jest administratorem lub członkiem.
Rozszerzenia:	*Formularz logowania ;Formularz rejestracyjny użytkownika ;Formularz spisu użytkowników ; Formularz danych użytkownika ;Formularz historii użytkownika*
Częstotliwość stosowania:	*Maksymalnie 100 dziennie.*
Status:	*Rozwoju*
Właściciel:	*publiczny, anonimowy użytkownik*
Priorytet:	*wysoki*



Rysunek 15.1: Diagram przypadków użycia - Autoryzacja i Zarządzanie Użytkownikami GreenDi

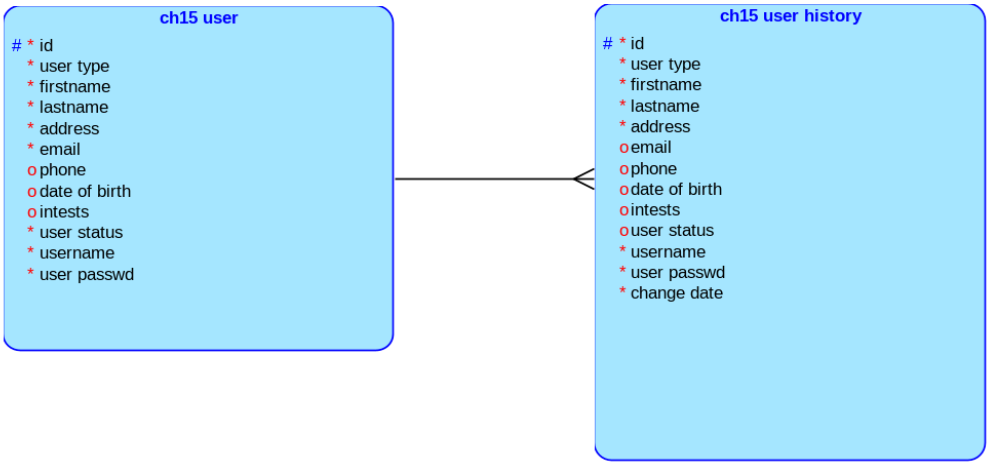
15.4 Model danych

15.4.1 Opis modelu danych

Dokumentacja logicznego modelu danych (modelu encji-relacji) zawiera 2 rodzaje encji: Użytkownik i Historia Użytkownika. Encja Użytkownik posiada następujące atrybuty: User ID (atrybut identyfikacyjny); Type (typ użytkownika: Członek, Administrator); Name (imię i nazwisko użytkownika); Address (adres użytkownika); Email; Telefon; Data urodzenia (opcjonalnie); Interests (zainteresowania - opcjonalnie); Status (Aktywny, Zawieszony, Nieaktywny); User name (niewidoczny dla administratora, zaszyfrowany); “Password” (hasło - niewidoczne dla administratora, zaszyfrowane). Każdy Użytkownik ma 1 lub więcej rekordów Historii Użytkownika. Historia Użytkownika ma te same atrybuty co Użytkownik, jest to kopia rekordu Użytkownika po dokonanej zmianie, z tym że zawiera ona automatycznie wygenerowany czas zmiany i ID osoby dokonującej zmiany (członek lub administrator). Oznacza to, że zmienione dane są rejestrowane w tym dzienniku po każdej zmianie.

15.4.2 Logiczny model danych

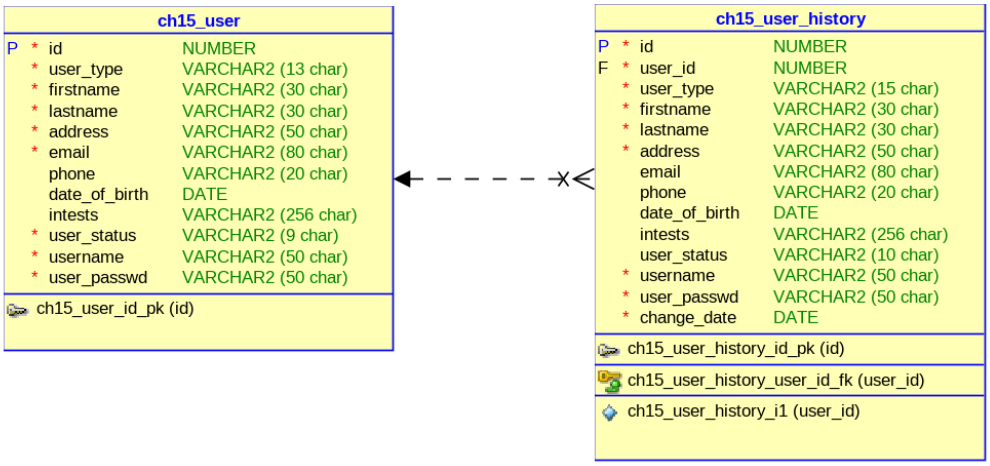
Logiczny model danych przedstawiono na rysunku 15.2.



Rysunek 15.2: Logiczny model danych.

15.4.3 Relacyjny model danych

Automatyczna transformacja z logicznego modelu danych do relacyjnego modelu danych w Oracle SQL Data Modeler jest zapewniana przez funkcję *Engineering to relational*.Wynik pokazano na rysunku 15.3.

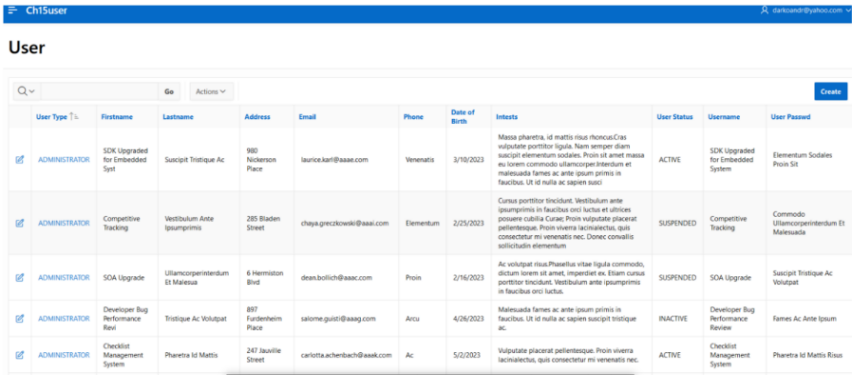


Rysunek 15.3: Relacyjny model danych.

Pozwól teraz Oracle SQL Data Modeler wygenerować skrypt SQL. Wybierz wszystkie tabele w modelu relacyjnym i użyj funkcji *File* > *Export* > *DDL File* , aby uzyskać skrypt. Możemy zaimportować wygenerowany skrypt w APEX i wykonać go.

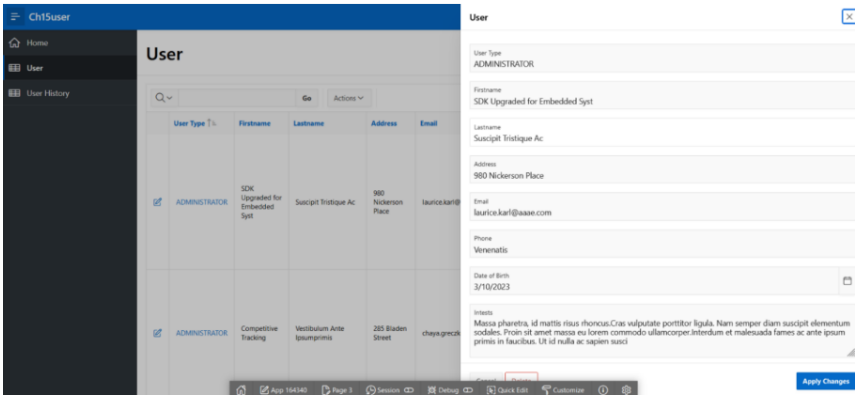
15.5 Interfejsy aplikacji

Formularz użytkownika przedstawiono na rysunku 15.4.



Rysunek 15.4: Formularz użytkownika.

Wstawianie danych do formularza historii użytkownika jest przedstawione na rysunku 15.5.



Rysunek 15.5: Formularz historii użytkownika.

Funkcjonalności wymagające autoryzacji użytkownika zostały opisane w rozdziale 18 Sklep internetowy - GreenDi.

15.6 Dodatkowe materiały do nauki

Możesz znaleźć następujące dodatkowe materiały do nauki:

- eksportowana aplikacja
- skrypty do tworzenia, upuszczania i wstawiania
- przewodnik wideo

Wszystkie dodatkowe materiały są dostępne na publicznej stronie BeeAPEX . Zaloguj się jako użytkownik gość (nie jest wymagane hasło). Znajdź podręcznik w sekcji “Books”(Książki), skrypty w folderze Part 2 > Chapter15 w sekcji “Scripts” (Skrypty) i przewodniki wideo w “Collection of video guides”. Materiał na krótkie kursy w “Short courses section”.

15.6.1 Eksportowana aplikacja

Eksportowana aplikacja jest spakowana. Instalacja tworzy tabele i wypełnia dane. Dezinstalacja usuwa wszystkie obiekty bazy danych używane w tej aplikacji.

15.6.2 Przewodniki wideo

Przewodnik wideo pokazuje każdy krok w tworzeniu aplikacji.

15.7 Pytania

1. Jakie są wbudowane funkcje uwierzytelniania i autoryzacji użytkowników na poziomie aplikacji podczas tworzenia aplikacji przy użyciu Oracle APEX?
2. Jak możemy korzystać z uwierzytelniania zewnętrznego w APEX?
3. Jakie są korzyści dla użytkowników, którzy logują się do aplikacji za pomocą swoich kont Facebook, Google lub podobnych?

15.8 Odpowiedzi

1. a) Wbudowane uwierzytelnianie użytkownika: APEX zapewnia wbudowaną funkcję uwierzytelniania użytkowników, umożliwiając definiowanie i zarządzanie kontami użytkowników bezpośrednio w środowisku APEX. Użytkownicy mogą mieć różne role i uprawnienia, a dostęp do określonych stron aplikacji lub składników można kontrolować na podstawie ról użytkowników. b) Autoryzacja na poziomie aplikacji: APEX umożliwia definiowanie szczegółowych reguł autoryzacji na poziomie aplikacji. Można określić, którzy użytkownicy lub role mają dostęp do określonych stron, regionów, przycisków lub innych składników w aplikacji.
2. Możemy korzystać z opcji logowania, takich jak Facebook, Google i inne platformy w aplikacji lub na stronie internetowej, którą sam opracowujesz. Wiele popularnych platform udostępnia interfejsy API programistów (Application Programming Interfaces), które umożliwiają integrację funkcji logowania z własnymi aplikacjami lub stronami internetowymi. Na przykład Facebook udostępnia interfejs API logowania Facebooka, który umożliwia użytkownikom logowanie się do aplikacji lub witryny przy użyciu danych logowania do Facebooka. Podobnie Google udostępnia interfejs Google Sign-In API, który umożliwia użytkownikom logowanie się przy użyciu kont Google. Aby zaimplementować te opcje logowania, zazwyczaj trzeba zarejestrować aplikację lub witrynę internetową w portalu dla programistów odpowiedniej platformy. Ten proces rejestracji zwykle obejmuje uzyskanie kluczy API lub identyfikatorów klientów, które następnie zostaną użyte w kodzie do uwierzytelniania użytkowników i zarządzania procesem logowania.
3. Wprowadzając opcje logowania z tych platform, możesz zaoferować użytkownikom wygodny i znajomy sposób logowania się do aplikacji lub witryny internetowej bez konieczności tworzenia nowych kont lub zapamiętywania dodatkowych danych logowania. Nie ma potrzeby rejestracji i wprowadzania nazwy, adresu i podobnych danych, nie są tworzone nowe nazwy użytkowników i hasła, które należy zapamiętać.



16. System małych innowacji

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA

16.1 Biznesowy pogląd na sprawę

We współczesnym przedsiębiorstwie innowacyjne pomysły są kluczem do sukcesu i poprawy wyników biznesowych. Dzięki swojej wiedzy na temat procesów biznesowych, pracownicy są prawdopodobnie najlepszym źródłem innowacyjnych pomysłów. Pomysły te obejmują różne punkty widzenia, takie jak sugestie dotyczące optymalizacji procesów, ulepszenia zarządzania relacjami z klientami, ulepszenia zadań zapewnienia jakości i podobne. Dlatego kierownictwo w tym konkretnym przypadku postanowiło zebierać małe innowacyjne pomysły od pracowników i nagradzać ich.

16.2 Definicja problemu

Dotychczas innowacyjne pomysły w przedsiębiorstwie były zbierane za pośrednictwem poczty elektronicznej i formularzy papierowych. Przegląd pomysłów w przedsiębiorstwie był czasochłonny. Kryteria oceny pomysłów nie były zdefiniowane. Ze względu na dużą liczbę potencjalnych pomysłów, które mogłyby przynieść korzyści przedsiębiorstwu, istnieje potrzeba posiadania dobrego przeglądu. Platforma powinna umożliwiać gromadzenie innowacyjnych pomysłów w małym przedsiębiorstwie. Pracownicy powinni mieć możliwość zalogowania się na platformie i wprowadzenia swoich pomysłów za pomocą prostych opisów lub załączników. Ponieważ pomysły nie mają takiej samej wartości (dla kierownictwa i operacji), zostaną poddane przeglądowi i ocenie. Kiedy oceniający zakończą proces, pomysł jest oceniany jako zaakceptowany, odrzucony lub wymaga poprawy. Najbardziej innowacyjne pomysły zostaną nagrodzone w zależności od uzyskanej oceny. Za zaproponowany pomysł można uzyskać następujące nagrody: GOLD (100 EUR), SILVER (80 EUR), BRONZE (60 EUR), CONTRIBUTOR (40 EUR) lub THANK YOU (Podziękowanie). Tylko zarząd będzie mieli możliwość oceny pomysłów. Kierownicy będą regularnie inicjować proces oceny i nagradzania (np. raz w miesiącu). Wszyscy zalogowani użytkownicy będą mogli przeglądać pomysły po ich ocenie.

16.3 Przypadki użycia

16.3.1 Opis przypadku użycia

Każdy pracownik ma możliwość zaproponowania pomysłów innowacyjnych poprzez formularz zgłoszeniowy. Wszystkie pomysły są zbierane i oceniane przez kierownictwo. Pomysły są klasyfikowane według przydatności i możliwości realizacji, a dobre pomysły są nagradzane.

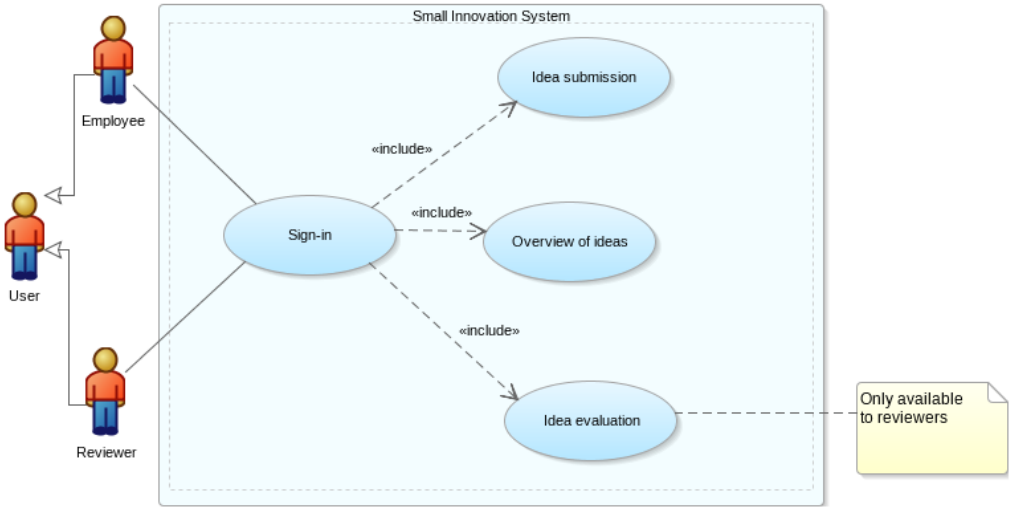
16.3.2 Opis częściowo ustrukturyzowany

Możemy podsumować trzy różne historie użytkowników lub przypadki użycia: zgłoszenie pomysłu, przegląd pomysłów i ocena pomysłu. Ostatni krok - wypłata nagrody nie jest objęty tym przypadkiem biznesowym. Dodanie załącznika do pomysłu jest właściwie rozszerzeniem opisanego powyżej przypadku użycia, i moglibyśmy opisać go jako nowy przypadek użycia. Dla czytelności kontynuujemy jednak z pozostałymi dwoma głównymi przypadkami użycia, ale aplikacja będzie miała funkcję dodawania załączników do pomysłu.

Dodanie załącznika do pomysłu jest w rzeczywistości rozszerzeniem opisanego powyżej UC i moglibyśmy opisać go jako nowy przypadek użycia. Ze względu na czytelność, kontynuujemy pozostałe dwa główne przypadki użycia, jednak aplikacja będzie miała funkcję dodawania załączników do pomysłu.

16.3.3 Schemat przypadków użycia

Powyższa historia jest przedstawiona na diagramie przypadków użycia(zob.rysunek 16.1).



Rysunek 16.1: Diagram przypadków użycia.

Ponownie, ze względu na prostotę, przypadek użycia rozszerzenia (dodanie załącznika do pomysłu) został pominięty na rysunku 16.1.

16.4 Model danych

16.4.1 Opis modelu danych

W logicznym modelu danych istnieją trzy encje. Jednostka CH16 IDEA ma osiem atrybutów: ID, datę przesłania, tytuł, opis pomysłu, nazwę pliku pomysłu, typ MIME, decyzję i datę decyzji. Każdy pomysł jest proponowany przez jednego pracownika. Podstawowe dane o pracownikach przechowywane są w jednostce CH16 PRACOWNIK: ID, imię i nazwisko. Pracownik może być

Tablica 16.1: Opis przypadku użycia: Zgłoszenie pomysłu.

Słowo kluczowe	**Wartość**
ID:	*Ch16-01*
Tytuł:	*Zgłaszanie pomysłów*
Opis:	*Pracownik loguje się na platformie i wprowadza pomysł. Każdy pomysł ma tytuł i krótki opis dodany jako załącznik (plik dokumentu).*
Główny aktor:	*Pracownik*
Warunki wstępne:	*Pracownik musi mieć konto użytkownika Oracle APEX dla aplikacji Small Innovation System. Dostęp do aplikacji internetowej musi być włączony.*
Warunki końcowe:	*Po pomyślnym wprowadzeniu pomysłu, pomysł i szczegółowy opis w załączniku są dostępne dla innych użytkowników portalu SIS.*
Główne	*Scenariusze*
Scenariusz sukcesu:	1. *Otwórz przeglądarkę internetową i zaloguj się do aplikacji SIS* 2. *Wybierz Pomysły z menu* 3. *Wybierz Dodaj nowy pomysł* 4. *Wpisz Tytuł pomysłu i dodaj załącznik* 5. *Wprowadź nazwę pliku i typ załącznika* 6. *Potwierdź wstawienie, klikając przycisk "Create"* 7. *Przejrzyj wstawione dane*
Rozszerzenia:	• *1a. Logowanie kończy się niepowodzeniem* • *1a* Rozszerz:* • *1a1. Pokaż komunikat o błędzie* • *1a2. Otwórz okno logowania* • *4a. Błąd typu danych* • *4a* Rozszerz:* • *4a1. Pokaż komunikat o błędzie* • *7a. Błąd typu danych* • *7a* Rozszerz:* • *7a1. Pokaż komunikat o błędzie*
Częstotliwość stosowania:	*Średnio firma otrzymuje 3 innowacyjne pomysły tygodniowo.*
Status:	*Gotowy*
Właściciel:	*Pracownik z dostępem do aplikacji*
Priorytet:	*Umiarkowany*

Tablica 16.2: Opis przypadku użycia: Przegląd pomysłów.

Słowo kluczowe	**Wartość**
ID:	*Ch16-02*
Tytuł:	*Przegląd pomysłów*
Opis:	*Pracownik loguje się na platformie i sprawdza status pomysłu lub innych pomysłów.*
Główny aktor:	*Pracownik*
Warunki wstępne:	*Pracownik musi mieć konto użytkownika Oracle APEX dla aplikacji Small Innovation System. Dostęp do aplikacji internetowej musi być włączony.*
Warunki końcowe:	-
Główne	*Scenariusze*
Scenariusz sukcesu:	1. *Otwórz przeglądarkę internetową i zaloguj się do aplikacji SIS* 2. *Wybierz Pomysły z menu* 3. *Przeglądaj listę pomysłów i ich status* 4. *Wybierz Przegląd w menu* 5. *Zobacz wykresy danych przesyłania pomysłów*
Rozszerzenia:	• *1a. Logowanie kończy się niepowodzeniem* • *1a* Rozszerz:* • *1a1. Pokaż komunikat o błędzie*
Częstotliwość stosowania:	*Opcja przeglądania pomysłów jest dostępna dla pracowników, recenzentów i administratorów. Łącznie uzyskują oni dostęp do platformy około 5 razy dziennie.*
Status:	*Finshed*
Właściciel:	*Pracownik z opcją logowania*
Priorytet:	*niski*

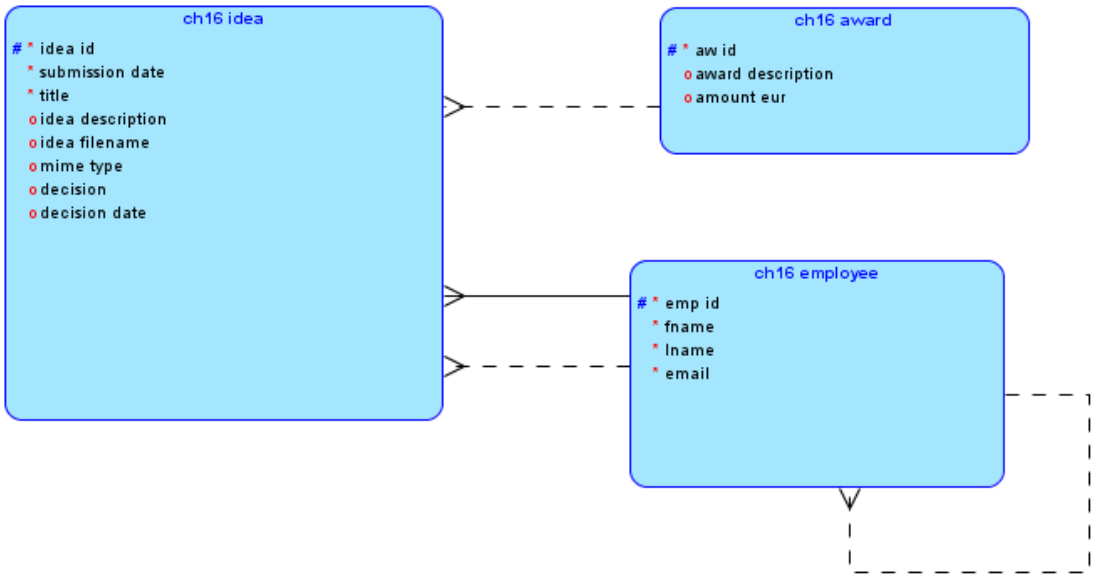
Tablica 16.3: Opis przypadku użycia: Ocena pomysłu.

Słowo kluczowe	**Wartość**
ID:	*Ch16-03*
Tytuł:	*Ocena pomysłu*
Opis:	*Recenzent loguje się na platformie i sprawdza pomysł. Nagroda jest również zdefiniowana w tym przypadku użycia.*
Główny aktor:	*Recenzent*
Warunki wstępne:	*Użytkownik pracownik musi zostać dodany do roli CH16_Reviewer. Dostęp do aplikacji internetowej musi być włączony.*
Warunki końcowe:	*Wybrany pomysł jest oceniany i definiowana jest proponowana nagroda.*
Główny	*Scenariusze*
Scenariusz sukcesu:	1. *Otwórz przeglądarkę internetową i zaloguj się do aplikacji SIS* 2. *Wybierz Przejrzyj pomysły z menu* 3. *Zidentyfikuj pomysł i dodaj szczegóły za pomocą przycisku "Edit"(Edytuj)* 4. *Wybierz decyzję z listy wartości " Accept, Reject, Revise" (Zaakceptuj, Odrzuć, Popraw zmianę)* 5. *Wybierz rodzaj nagrody z listy wartości (Gold, Silver, Bronze, Contributor, Thank you)* 6. *Potwierdź aktualizację*
Rozszerzenia:	• *1a. Logowanie kończy się niepowodzeniem* • *1a* Rozszerz:* • *1a1. Pokaż komunikat o błędzie* • *1a2. Otwórz okno logowania* • *2a. Błąd typu danych* • *2a* Rozszerz:* • *2a1. Pokaż komunikat o błędzie* • *3a. Błąd typu danych* • *3a* Rozszerz:* • *3a1. Pokaż komunikat o błędzie* • *4a. Błąd typu danych* • *4a* Rozszerz:* • *4a1. Pokaż komunikat o błędzie* • *5a. Błąd typu danych* • *5a* Rozszerz:* • *5a1. Pokaż komunikat o błędzie*
Częstotliwość stosowania:	*Menedżerowie raz w miesiącu przeglądają proponowane pomysły.*
Status:	*Gotowy*
Właściciel:	*Recenzent*
Priorytet:	*wysoki*

menedżerem dla innych pracowników. Każdy pracownik ma tylko jednego menedżera, podczas gdy menedżer może zarządzać kilkoma pracownikami. Menedżerowie oceniają pomysły. Jeden menedżer może ocenić wiele pomysłów, podczas gdy jeden pomysł może być oceniony tylko przez jednego menedżera. Każdy pomysł może otrzymać jedną nagrodę. CH16 AWARD ma trzy atrybuty: ID, opis nagrody i kwotę w euro. Jeden rodzaj nagrody może być przyznany większej liczbie pomysłów. Reguły oceny nie są uwzględniane w modelu danych, ponieważ stwarzałoby to ryzyko zmiany aplikacji przy każdej zmianie reguł. Dzięki proponowanemu modelowi danych zapewniliśmy: niezawodność aplikacji, szybki rozwój i wdrożenie, niskie koszty rozwoju i utrzymania oraz elastyczność.

16.4.2 Logiczny model danych

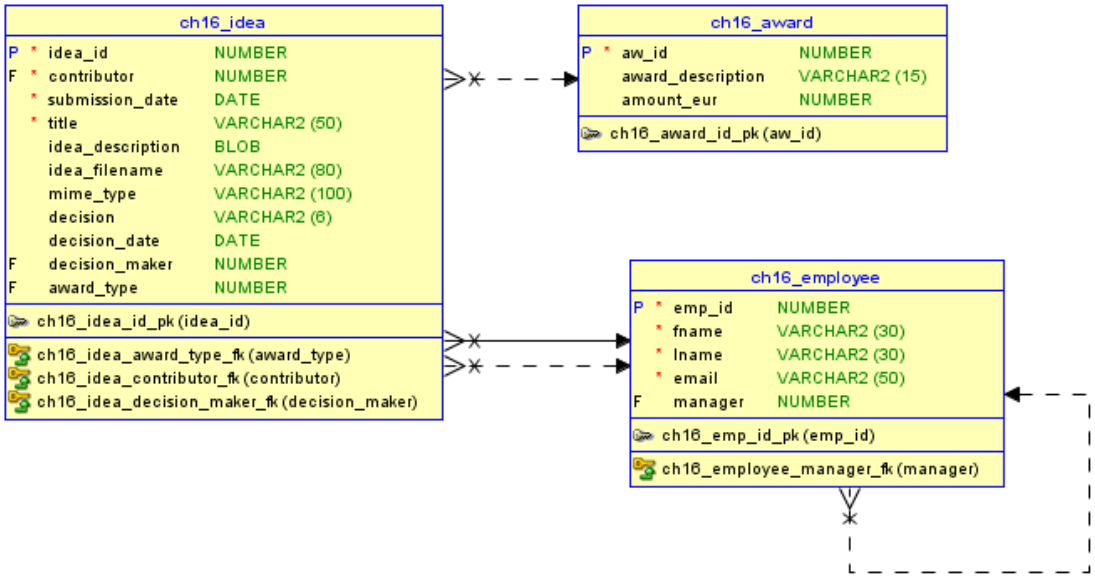
Powyższy opis przedstawiono na logicznym modelu danych (patrz rysunek 16.2).



Rysunek 16.2: Logiczny model danych.

16.4.3 Relacyjny model danych

Automatyczna transformacja z modelu danych logicznych na model danych relacyjnych w Oracle SQL Data Modeler jest dostarczana przez funkcję *Engineering to relational*. Wynik jest przedstawiony na rysunku 16.3.



Rysunek 16.3: Relacyjny model danych.

Oracle SQL Data Modeler generuje również skrypt SQL do tworzenia tabel, sekwencji i wyzwalaczy. Wybierz wszystkie tabele w modelu relacyjnym i użyj funkcji *File* > *Export* > *DDL File* , aby uzyskać skrypt podobny do tego:

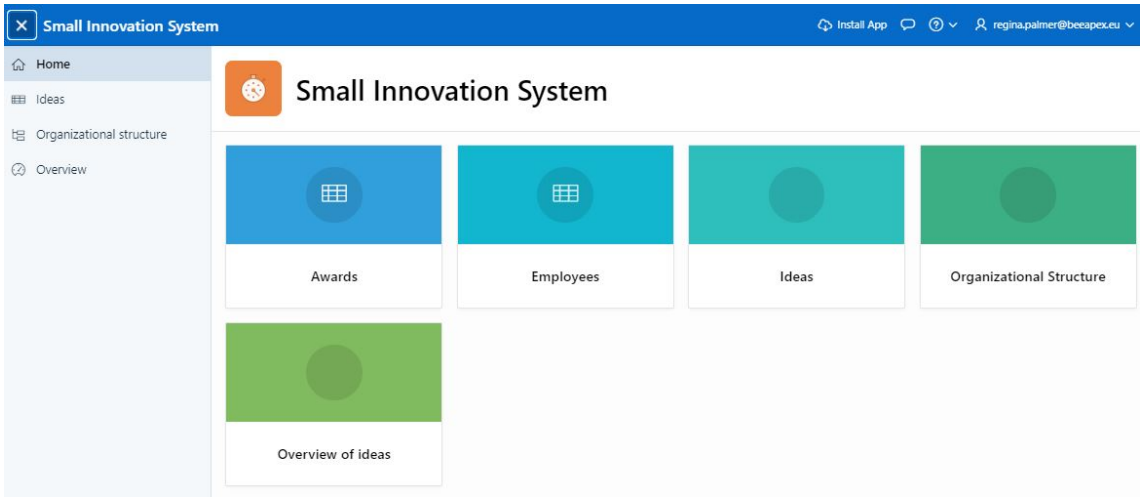
```
CREATE TABLE CH16_AWARD (
    AW_ID NUMBER NOT NULL
...
);

CREATE TABLE CH16_EMPLOYEE (
    EMP_ID NUMBER NOT NULL
...
);
```

Teraz nadszedł czas na wygenerowanie tabel w bazie danych Oracle. Możemy zaimportować wygenerowany skrypt i wykonać go w APEX.

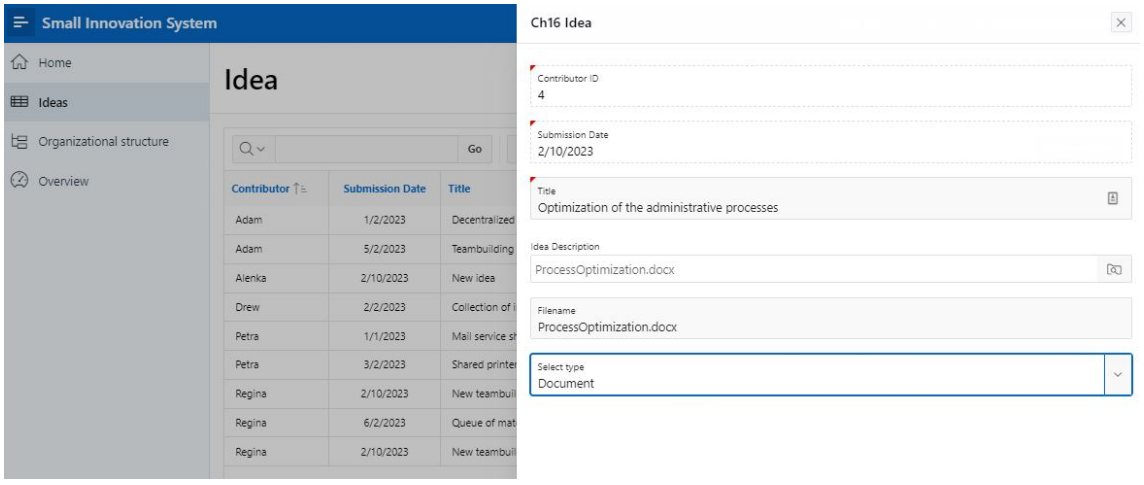
16.5 Interfejsy aplikacji

Na poniższych rysunkach przedstawiono interfejsy aplikacji dla obu podstawowych ról (pracownika i recenzenta). Jako pierwszy, pulpit nawigacyjny dla pracownika na rysunku 16.4.



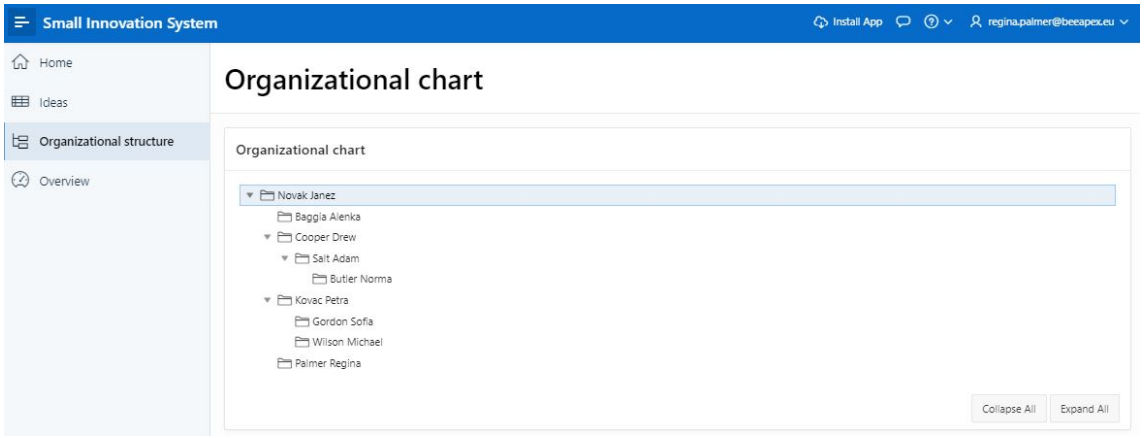
Rysunek 16.4: Pulpit nawigacyjny pracownika.

Strone do zgłaszania pomysłów wraz z załącznikami i ich szczegółami przedstawiono na rysunku 16.5.



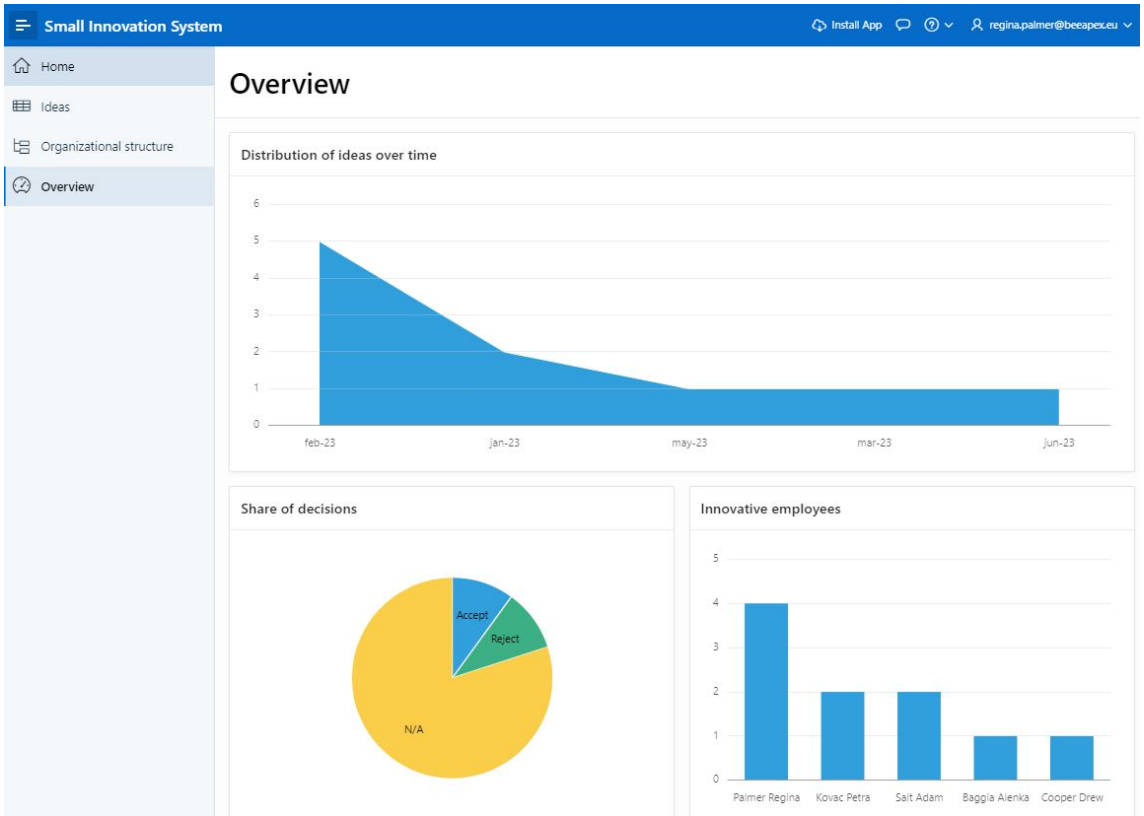
Rysunek 16.5: Przesyłanie pomysłu z załącznikami.

Każdy pracownik ma możliwość przeglądania struktury organizacyjnej w firmie, jak pokazano na rysunku 16.6.



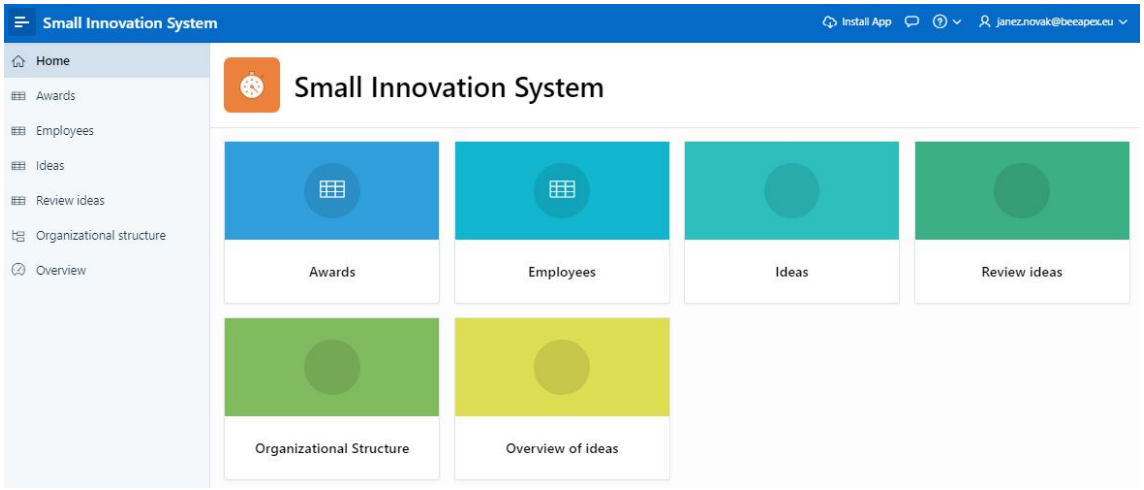
Rysunek 16.6: Struktura organizacyjna w firmie.

Ponadto każdy pracownik może zobaczyć pulpit nawigacyjny wykresów przedstawiających rozkład pomysłów w czasie, udział zaakceptowanych, odrzuconych i zrewidowanych pomysłów oraz pracowników z największą ilością innowacyjnych pomysłów w firmie(patrz rysunek 16.7).



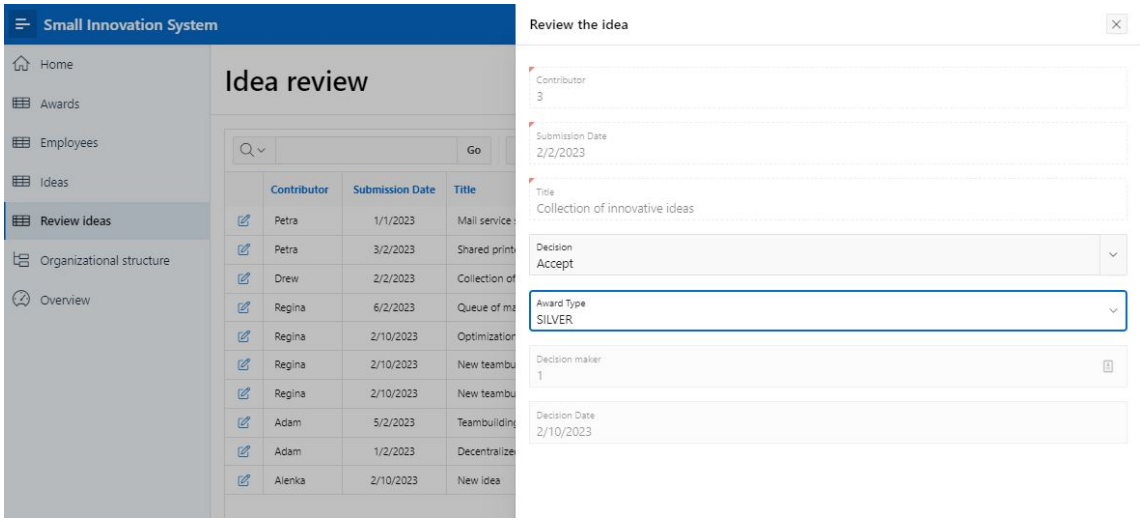
Rysunek 16.7: Przegląd pomysłów na małe innowacje.

Oprócz opisanych działań, recenzent ma możliwość aktualizacji danych o nagrodach i pracownikach, jak pokazano na rysunku 16.8. Strona administracyjna jest dostępna tylko dla administratorów, a nie dla recenzentów.



Rysunek 16.8: Pulpit nawigacyjny recenzenta.

Strona do przeglądu pomysłu jest przedstawiona na rysunku 16.9. Ta strona jest dostępna tylko dla Recenzenta. Recenzent wybiera decyzję i rodzaj nagrody, jeśli dotyczy.



Rysunek 16.9: Strona do przeglądania pomysłów.

16.6 Dodatkowe materiały do nauki

Możesz znaleźć następujące dodatkowe materiały do nauki:

- skrypt do tworzenia
- skrypt do wypełniania tabel
- skrypt do usuwania tabel
- wyeksportowaną i zapakowaną aplikację
- film instruktażowy, który prezentuje, jak wygenerować aplikację na podstawie skryptu

Wszystkie dodatkowe materiały są dostępne na publicznej stronie BeeAPEX . Zaloguj się jako użytkownik gość (nie jest wymagane hasło). Znajdź podręcznik w sekcji “Books”(Książki), skrypty w folderze Part 2 > Chapter16 w sekcji “Scripts” (Skrypty) i przewodniki wideo w “Collection of video guides”. Materiał na krótkie kursy w “Short courses section”.

16.6.1 Eksportowana aplikacja

Wyeksportowana aplikacja jest spakowana. Instalacja tworzy tabele, a także wypełnia dane. Deinstalacja usuwa wszystkie obiekty baz danych używane w tej aplikacji. Spakowana aplikacja jest testowana i będzie działać w nowym obszarze roboczym, jeśli spełnione są następujące wymagania:

- dodaj użytkownika APEX przed uruchomieniem aplikacji.

Jeśli użytkownikowi nie zostanie przypisana odpowiednia rola, zaimportowana aplikacja ulegnie awarii. Konieczne jest wyczyszczenie plików cookie przeglądarki internetowej (i.e. Firefox: Settings > Cookies and Site Data > Manage Data)) po awarii aplikacji z powodu niespełnionych wymagań.

16.6.2 Przewodniki wideo

Przewodnik wideo pokazuje wszystkie kroki tworzenia aplikacji.

16.7 Pytania

1. Gdzie zdefiniowano role użytkowników w rozdziale 16?
2. Która relacja w logicznym modelu danych umożliwia prezentację struktury organizacyjnej w rozdziale 16?
3. Dlaczego niektóre pola na stronie do przeglądu pomysłów są wyłączone w rozdziale 16?

16.8 Odpowiedzi

1. W modelu danych rozdziału 16 nie są zdefiniowane żadne role. Są one używane tylko w środowisku APEX.
2. Relacja łącząca tabelę Employee z samą sobą (relacja rekurencyjna) umożliwia prezentację struktury organizacyjnej w rozdziale 16.
3. W rozdziale 16 niektóre pola są wyłączone, aby zapewnić lepsze wrażenia użytkownika i spójność danych, ponieważ użytkownik nie jest zobowiązany do dodawania danych, które już znajdują się w bazie danych.



17. Zarządzanie procesami biznesowymi

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA

17.1 Biznesowy pogląd na sprawę

Większość tej książki podkreśla znaczenie danych i podejścia opartego na danych w tworzeniu aplikacji internetowych za pomocą APEX. Ale podejście oparte na danych ma swojego brata bliźniaka: aplikacje oparte na procesach. Jak sama nazwa wskazuje, aplikacje oparte na procesach skupiają się na procesach. Dobra aplikacja oparta na danych nie może zaniedbywać aspektu procesowego, podobnie jak aplikacja oparta na procesach nie może zaniedbywać aspektu danych. Ukryte lub wyraźne włączenie obu, sprawia, że każda aplikacja jest odpowiednia do celu. Ważne jest, aby programista miał głęboki wgląd w procesy biznesowe w organizacji. Każdy rodzaj organizacji, niezależnie od wielkości, rocznych zarobków, własności, branży i misji, przeprowadza procesy biznesowe. Typowe klasy procesów biznesowych, które mają zastosowanie do większości organizacji, należą do następujących kategorii:

- Sprzedaż i marketing obejmują podprocesy, takie jak pozyskiwanie i zatrzymywanie klientów, generowanie potencjalnych klientów, zarządzanie relacjami z klientami i kampanie marketingowe.
- Zarządzanie łańcuchem dostaw obejmuje podprocesy, takie jak koordynacja i zarządzanie przepływem towarów i usług, zaopatrzenie, produkcja i dystrybucja.
- Zarządzanie finansami obejmuje podprocesy, takie jak zarządzanie zasobami finansowymi organizacji, księgowość, budżetowanie i sprawozdania finansowe.
- Zarządzanie zasobami ludzkimi obejmuje podprocesy, takie jak zarządzanie siłą roboczą organizacji, rekrutacja, szkolenia i rozwój pracowników.
- Obsługa klienta obejmuje podprocesy, takie jak zapewnianie wsparcia i pomocy klientom, rozwiązywanie skarg i rozwiązywanie problemów.
- Zarządzanie operacyjne obejmuje podprocesy, takie jak zarządzanie codziennymi operacjami biznesowymi, produkcją, logistyką i zarządzaniem zapasami.
- Technologia informacyjna obejmuje podprocesy, takie jak zarządzanie zasobami i systemami informatycznymi organizacji, sprzęt, oprogramowanie i zarządzanie danymi.

Procesy te są uważane za krytyczne, ponieważ wspierają podstawowe funkcje organizacji i umożliwiają osiągnięcie jej celów i zadań. Poprawa wydajności i efektywności tych procesów może mieć znaczący wpływ na ogólną wydajność i sukces organizacji. Próbując jasno zrozumieć i zdefiniować typowy biznes, OASIS OPEN opracował uniwersalny język biznesowy, który opisuje procesy i dokumenty biznesowe (patrz [5]) . Przemysł oprogramowania rozpoczął się pod koniec

1990 roku od integracji procesów, wynalezionych przez Business Process Execution Language (BPEL), a później Business Process Modeling Language (BPML). Wielcy gracze, tacy jak Microsoft, IBM, SAP, Oracle, a także niezliczone mniejsze firmy przyczyniły się do powstania platform oprogramowania, które miały na celu zapewnienie bezproblemowej integracji funkcji biznesowych. Oprócz świata oprogramowania inne branże skupiły się na procesach poprzez pryzmat konkretnych przedsiębiorstw od lat 80. za pomocą inicjatyw związanych z zarządzaniem jakością i normami.

Od 1990 roku dyscypliny zarządzania procesami biznesowymi (BPM) i re-engineering procesów biznesowych (BPR) odgrywają ważną rolę, gdy organizacje chcą poprawić swoją wydajność. BPM koncentruje się na reprezentacji i dokumentacji przepływów pracy, operacji i procesów firmy za pomocą diagramów, schematów blokowych lub innych pomocy wizualnych. Celem jest zrozumienie i ulepszenie istniejących procesów lub zaprojektowanie nowych, aby zwiększyć wydajność i zmniejszyć nieefektywność. BPR jest poszukiwany jako fundamentalne przemyślenie i radykalne przeprojektowanie procesów biznesowych w celu osiągnięcia znacznej poprawy wydajności i produktywności. Obejmuje analizę i projektowanie przepływów pracy i procesów w organizacji, w celu zwiększenia wydajności, skuteczności i zdolności do dostosowywania się do zmieniającego się środowiska biznesowego. BPR może obejmować technologię wykorzystania i systemy informacyjne w celu automatyzacji procesów i usprawnienia procesu decyzyjnego.

W BPM i BPR można użyć kilku diagramów:

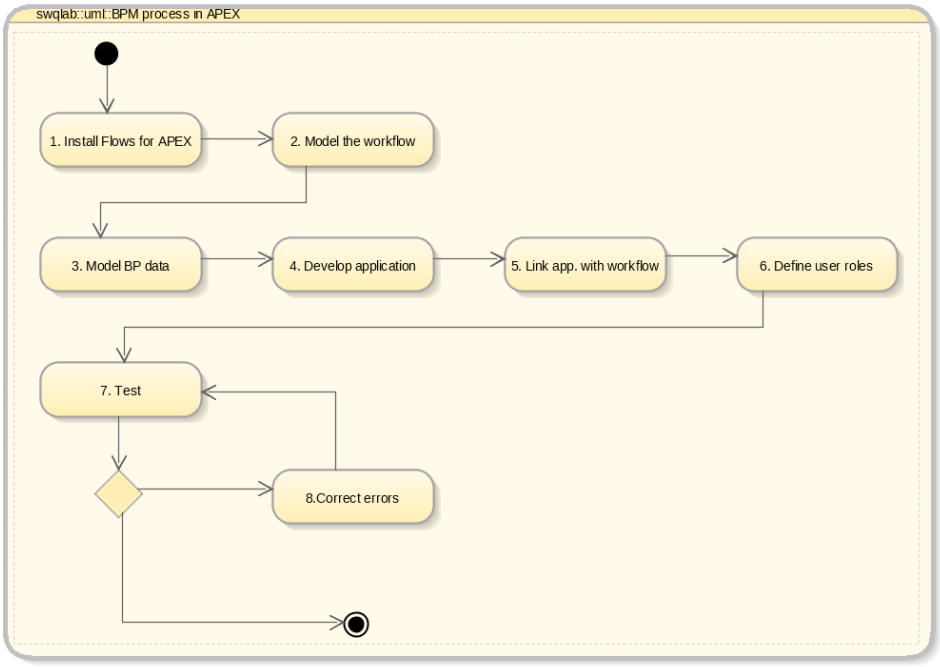
- Flowcharts: prosty i intuicyjny sposób przedstawiania procesu jako serii kroków, decyzji i pętli.
- Swimlane diagrams:typ schematu blokowego, który dodaje wizualną reprezentację tego, kto lub co jest odpowiedzialne za każdy krok w procesie.
- Process maps: ogólny widok kroków i działań związanych z procesem, często obejmujący dane wejściowe, wyjściowe i punkty decyzyjne.
- BPMN (Business Process Model and Notation [4]): Standardowa notacja branżowa używana do modelowania procesów biznesowych, standaryzowana przez Object Management Group (OMG).
- EPC (Event-driven Process Chain): Typ schematu blokowego, który wykorzystuje zdarzenia jako siłę napędową procesu, a nie działania.
- IDEF (Integration Definition for Function Modeling): Metoda stosowana w inżynierii oprogramowania do modelowania i analizowania procesów i systemów.

Wybór typu diagramu do prezentacji przepływu pracy zależy od poziomu wymaganych szczegółów, przeznaczenia modelu i grupy docelowej. W tym momencie skupimy się tylko na BPMN. Główne powody to: a) BPMN ma standardową notację i szeroką akceptację w różnych branżach oraz b) BPMN jest zaimplementowany jako typ diagramu modelowania w funkcji Oracle Application Express (APEX) (czasami określanej jako rozszerzenie) o nazwie Flows for APEX. Flows for APEX są open source, licencjonowane na podstawie bardzo liberalnej licencji MIT. Przepływy dla APEX umożliwiają deweloperom tworzenie i wdrażanie aplikacji internetowych przy użyciu wizualnego interfejsu typu "przeciągnij i upuść". Przepływy dla APEX umożliwiają automatyzację i uproszczenie złożonych procesów biznesowych poprzez podzielenie ich na szereg kroków i zadań oraz przeprowadzenie użytkowników przez nie.

Przepływy dla APEX mogą zawierać różne elementy, takie jak strony, formularze, raporty i okna dialogowe, i mogą być skonfigurowane tak, aby obejmowały logikę rozgałęzień, rozgałęzianie warunkowe, sprawdzanie poprawności danych i obsługę błędów. Interfejs wizualny aplikacji Flows for APEX ułatwia deweloperom projektowanie, tworzenie i testowanie procesów biznesowych bez konieczności rozbudowanego kodowania. Można go zintegrować z innymi komponentami Oracle APEX, takimi jak usługi SQL, PL/SQL i REST, w celu tworzenia kompletnych aplikacji internetowych, a także tworzenia niestandardowych rozwiązań dla konkretnych potrzeb biznesowych, takich jak zarządzanie przepływem pracy, wdrażanie klientów i zarządzanie danymi. Ogólnie rzecz biorąc, przepływy dla APEX zapewniają sposób organizacji na usprawnienie i automatyzację

złożonych procesów biznesowych, zwiększenie produktywności oraz zmniejszenie liczby błędów i nieefektywności. Doskonałe samouczki na temat przepływów dla APEX są dostępne na na tej stronie . Jak więc możemy połączyć APEX jako środowisko programowania low-code i Flows for APEX z wbudowaną standardową prezentacją BPMN?

Diagram aktywności (zob.rysunek 17.1) przedstawia uproszczoną prezentację integracji przepływów i aplikacji APEX.



Rysunek 17.1: Integracja przepływów dla APEX z aplikacją APEX.

Kroki *3. modeling data, 4. developing application, 6. defining user roles, 7. testing* i *8. removing errors from application* są wspólne dla wszystkich aplikacji APEX. Krok *1. installing Flows for APEX* jest podobny do instalowania dowolnej innej spakowanej aplikacji. Prawdziwe nowości to *2. modeling the workflow* i *5. linking APEX application with workflow*.

Czytelnikom niezaznajomionym z diagramami zarządzania procesami biznesowymi (BPM) i notacji modelowania procesów biznesowych (BPMN) zaleca się poszukiwanie popularnych podręczników i studiowanie wspomnianej już spefyfikacji BPMN [4] . Jako krótkie przypomnienie dla wszystkich czytelników, następujące elementy diagramów BPMN są najczęściej używane:

- Start Event: Wskazuje punkt początkowy procesu.
- Task: Reprezentuje pojedynczą jednostkę pracy, taką jak działanie lub krok w procesie.
- Gateway: Reprezentuje punkt decyzyjny w procesie, taki jak rozwidlenie w przepływie lub gałąź warunkowa.
- Sequence Flow: Łączy elementy na diagramie i reprezentuje przepływ sterowania z jednego elementu do drugiego.
- End Event: Wskazuje koniec procesu.
- Pool/Lane: Reprezentuje grupę powiązanych zadań, używaną do definiowania ról i obowiązków różnych uczestników procesu.
- Message Flow: Reprezentuje przepływ komunikatów między uczestnikami procesu, takiego jak komunikacja między dwoma systemami.
- Data Object: Reprezentuje fragment danych, który jest używany lub produkowany w procesie, takim jak faktura lub rekord klienta.
- Data Store: Reprezentuje kontener do przechowywania danych, które są używane lub

tworzone w procesie, takim jak baza danych lub system plików.

Elementy te można łączyć w celu stworzenia szczegółowych i dokładnych modeli procesów biznesowych, umożliwiając organizacjom identyfikację obszarów wymagających poprawy i optymalizację przepływów pracy.

17.2 Rozpoczęcie pracy z aplikacją Flows dla APEX

17.2.1 Instalowanie Flow dla APEX

Pobierz plik zip z z tego adresu URL i rozpakuj wszystkie pliki na swój komputer. Postępuj zgodnie z instrukcjami podanymi pod tym linkiem. Przed zainstalowaniem wtyczki Flows for APEX sprawdź wymagania. Wymagania są spełnione, jeśli korzystasz z bezpłatnej przestrzeni roboczej APEX, bezpłatnego/płatnego OCI lub konta Oracle Academy. Dodatkowo po instalacji:

- Udziel opcji ćreate job" do obszaru roboczego. Ty lub administrator instancji APEX musisz nawiązać połączenie z bazą danych (za pomocą interfejsu OCI, SQL Developer, SQL Plus, TOAD) i wydaj polecenie. Nie można przyznać tego uprawnienia bezpośrednio w ramach APEX.
- skonfiguruj obszar roboczy, domyślną aplikację i informacje o użytkowniku w aplikacji "Flows for APEX". Ten krok można wykonać w dowolnym momencie później, gdy opracujesz własną aplikację opartą na procesach.

Chociaż czasami nazywane jest funkcją lub wtyczką, instalacja tworzy nowe obiekty bazy danych (tabele, widoki) i aplikacji, o nazwie "Przepływy dla APEX". Ta aplikacja jest interfejsem administracyjnym, który umożliwia:

- monitorowanie wystąpień przepływów pracy na panelu nawigacyjnym (Dashboard),
- wizualne tworzenie nowych przepływów pracy (Flow Management) oraz
- wykonywania przepływów pracy poprzez tworzenie wystąpień, wykonywanie zadań należących do instancji i testowanie przepływów prac (Flow Monitor).

17.2.2 Instalowanie przykładowej aplikacji przepływu procesu

Przykładowa aplikacja przepływu procesu jest również dostarczana w tym samym pliku ZIP co "Flows for APEX". Instalacja jest prosta. Dotyczy ona rozliczeń kosztów. Pracownik przygotowuje wniosek o zwrot kosztów, który jest walidowany przez kierownika. Jeśli wniosek zostanie odrzucony, to pracownik jest informowany. W końcu, dla zatwierdzonych wniosków, dział księgowości przygotowuje płatność, wykonuje transakcję bankową i ustawia status płatności. Jeśli wniosek zostanie opłacony, to instancja tego przepływu pracy jest zakończona. Aplikacja jest bardzo podobna do rzeczywistego procesu w każdej organizacji. Ponieważ jest tak dobrze udokumentowana, stanowi doskonały materiał do nauki.

17.2.3 Czytanie i ćwiczenia praktyczne

Istnieje doskonały samouczek dla przepływów dla APEX. Zacznij od tej strony. Na końcu tego ćwiczenia wprowadzającego znajdź samouczek BPMN i samouczek integracji APEX. Czytanie i praktykowanie umożliwi ci rozwijanie własnej aplikacji opartej na procesach. Praktykowanie wszystkich ćwiczeń prawdopodobnie zajmie kilka godzin, aby je zakończyć. Czytelnika zdecydowanie zachęca się do nauki przez działanie. Reszta tego rozdziału będzie zawierać ten sam szablon co inne rozdziały w tej książce (od definicji problemu po aplikację), z dodatkowym uwzględnieniem BPMN i integracji aplikacji z Flows for APEX. Jednak zakres aplikacji rozwijanej w rozdziale 17 zostaje zawężony do podstawowej funkcjonalności, pomijając szczegóły autoryzacji. Będzie ona zawierać przewodniki dotyczące:

- definiowania przepływu pracy w Flows for APEX
- uruchamiania instancji zdefiniowanego procesu w opracowanej aplikacji
- monitorowania przebiegu instancji w opracowanej aplikacji

- wykonywania każdego zadania w konkretnej instancji przepływu pracy w opracowanej aplikacji

Zawężona aplikacja może również stanowić ciekawe wyzwanie dla entuzjastycznego ucznia, prawda?

17.3 Definicja problemu

Średniej wielkości firma produkcyjna otrzymuje zapytania od klientów dotyczące zamówień na produkty dostosowane do ich potrzeb. Każde zapytanie może zawierać wiele dokumentów związanych ze specyfikacjami, wymaganymi standardami, schematami, szkicami i innymi podobnymi materiałami. Klient może składać wiele zapytań. Pracownik działu sprzedaży przygotowuje i klasyfikuje dokumenty związane z zapytaniem na trzy kategorie: produkcja, finanse i biznes. Aby ocenić jedno zapytanie, pracownicy działu sprzedaży muszą przygotować wiele dokumentów dla osób podejmujących decyzje. Kierownik produkcji i jego zespół oceniają aspekt produkcyjny zapytania. Dają ekspertyzę dotyczącą zdolności firmy do produkcji żądanego produktu. Jeśli produkt można wyprodukować, zespół podaje przewidywany czas trwania procesu produkcyjnego w dniach. Kierownik finansowy i jego zespół oceniają aspekt finansowy zapytania. Dają ekspertyzę dotyczącą zdolności firmy do udzielenia finansowania i szacują oczekiwany zysk. Ostateczna ocena i decyzja są podejmowane przez dyrektora generalnego i jego zespół. Rozważają oni możliwość (produkcyjną, finansową), oceniają znaczenie klienta i podejmują decyzję dotyczącą zapytania. W zależności od zdolności firmy do produkcji i finansowania okazji biznesowej oraz perspektyw biznesowych, dyrektor generalny decyduje, w jaki sposób odpowiedzieć klientowi: a) przesłać propozycję biznesową z ceną producenta i przewidywanym terminem dostawy lub b) wysłać wiadomość informującą klienta, że producent nie może przesłać propozycji biznesowej. Złe decyzje biznesowe (wysłanie propozycji biznesowej lub odrzucenie oferty biznesowej) mogą mieć katastrofalne skutki dla firmy produkcyjnej, takie jak utrata dochodów, niski zysk, zła reputacja z powodu przekroczenia terminów dostawy, a nawet bankructwo. Sytuacja przedstawiona powoduje ból głowy dla wszystkich zaangażowanych: pracownika działu sprzedaży, kierownika produkcji, kierownika finansowego i dyrektora generalnego. Dział IT jest odpowiedzialny za dostarczenie wsparcia oprogramowania zapewniającego łatwe w użyciu, niezawodne i bezpieczne wsparcie dla opisanego procesu. Ich zadaniem jest opracowanie takiego wsparcia za pomocą Flows for APEX. Aby opracować aplikację, musisz zainstalować Flows for APEX. W tym rozdziale zostanie przeanalizowany przypadek biznesowy (opis, opis półstrukturalny, diagram przypadków użycia, model procesu, model danych) i opracowana zostanie aplikacja. Dowiecie się Państwo:

- jak zdefiniować model procesu (workflow) dla tego przypadku biznesowego
- jak stworzyć model danych i opracować aplikację APEX
- jak połączyć opracowaną aplikację z zdefiniowanym workflowem

Proszę pamiętać, że autoryzacja użytkowników nie będzie wdrożona, ponieważ wymagałoby to rozszerzenia tego rozdziału poza rozsądny nakład pracy studenta.

17.4 Przypadki użycia

17.4.1 Opis narracji

Cztery podmioty są zaangażowane w cztery przypadki użycia. Każdy podmiot jest odpowiedzialny za jeden przypadek użycia: sprzedawca przygotowuje dokumentację związaną z zapytaniem, kierownik produkcji zapewnia ocenę aspektów produkcyjnych, kierownik finansowy zapewnia ocenę aspektów finansowych, a dyrektor generalny podejmuje decyzje. Diagram przypadków użycia (zob.rysunek 17.2)) zakłada, że projektant przepływu pracy zdecyduje, czy kierownik produkcji i finansów powinien pracować równolegle czy sekwencyjnie. Jeśli pracują równolegle i jeden lub obaj zdecydują, że wymagania zapytania nie mogą być spełnione, pojawia się niepotrzebny

Tablica 17.1: Opis przypadku użycia: przygotowanie dokumentacji zapytania.

Słowo kluczowe	**Wartość**
ID:	*ch17-01*
Tytuł:	*Przygotuj dokumentację zapytania ofertowego*
Opis:	*Sprzedawca tworzy nowe zapytanie i dodaje powiązane dokumenty*
Główny podmiot:	*Sprzedawca*
Warunki wstępne:	Klient wysłał już wszystkie dokumenty związane z zapytaniem
Warunki końcowe:	-
Główny	-
Scenariusz sukcesu:	1. dodaj nowe zapytanie 2. dodawanie powiązanych dokumentów do konkretnego zapytania 3. potwierdź przyciskiem Ćonfirmłub odrzuć przyciskiem “Cancel”
Rozszerzenia:	-
Częstotliwość stosowania:	*Ok. 250 rocznie, średnio 5 tygodniowo.*
Status:	[rozwoju]
Właściciel:	sprzedawca
Priorytet:	*wysoki*

wysiłek (koszty), ale przepływ pracy można wykonać szybciej. Jeśli projektant przepływów pracy zdecyduje się na podejście sekwencyjne, przepływ pracy zajmie więcej czasu, ale pojawi się mniej niepotrzebnego wysiłku.

17.4.2 Opis częściowo ustrukturyzowany

UML proponuje, aby każdy przypadek użycia miał również częściowo ustrukturyzowany opis. W tabelach 17.1, 17.2, 17.3 i 17.4 przedstawiono cztery odrębne opisy częściowo ustrukturyzowane.

17.4.3 Schemat przypadków użycia

Diagram przypadków użycia może również przedstawiać uogólnienie. W danym kontekście pracownik ma cztery specjalne instancje (menedżerowie terenowi), a każdy menedżer jest powiązany z jednym przypadkiem użycia. Wszystkie cztery przypadki użycia przedstawiono na rysunku 17.2.

Tablica 17.2: Opis przypadku użycia: ocena aspektów zapytania dotyczących produkcji

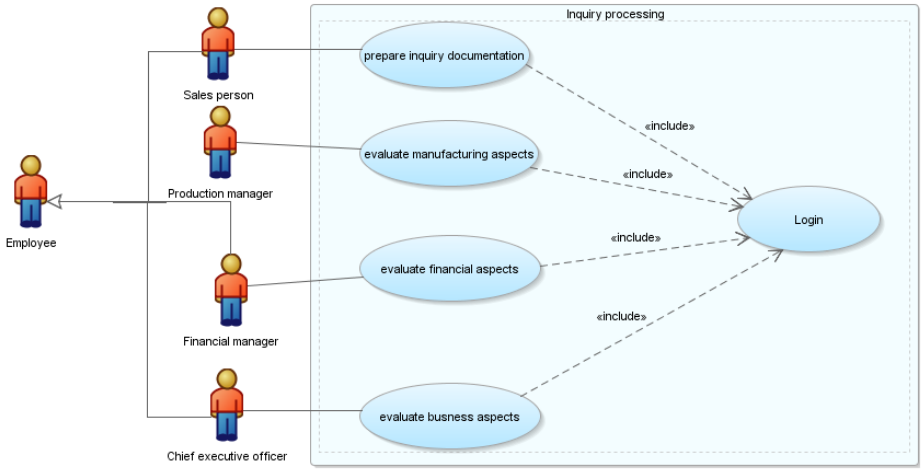
Słowo kluczowe	**Wartość**
ID:	*ch17-02*
Tytuł:	*Oceń aspekty produkcyjne zapytania*
Opis:	*Kierownik produkcji i jego zespół przedstawiają ekspertyzę, czy firma ma możliwość wytworzenia produktu, a jeśli tak, to jak długo (w dniach) zajmie produkcja danej ilości. Uzasadnienie opinii jest obowiązkowe.*
Główny aktor:	*Kierownik produkcji*
Warunki wstępne:	Przygotowana dokumentacja zapytania ofertowego
Warunki końcowe:	-
Główny	-
Scenariusz sukcesu:	1. wpisz opinię 2. potwierdź przyciskiem Ćonfirmłub odrzuć przyciskiem “Cancel”
Rozszerzenia:	-
Częstotliwość stosowania:	*Ok. 250 rocznie, średnio 5 tygodniowo.*
Status:	[rozwoju]
Właściciel:	Kierownik produkcji
Priorytet:	*wysoki*

Tablica 17.3: Opis przypadku użycia: ocena finansowych aspektów dochodzenia.

Słowo kluczowe	**Wartość**
ID:	*ch17-03*
Tytuł:	*Oceń finansowe aspekty dochodzenia*
Opis:	*Dyrektor finansowy i jego zespół przedstawiają ekspertyzę, czy firma ma możliwość sfinansowania pozycji produktu, a jeśli tak, to jaki byłby oczekiwany zysk. Uzasadnienie opinii jest obowiązkowe.*
Główny aktor:	*Dyrektor finansowy*
Warunki wstępne:	Przygotowana dokumentacja zapytania ofertowego
Warunki końcowe:	-
Główny	-
Success Scenario:	1. wpisz opinię 2. potwierdź przyciskiem Ćonfirmłub odrzuć przyciskiem “Cancel”
Rozszerzenia:	-
Częstotliwość stosowania:	*Ok. 250 rocznie, średnio 5 tygodniowo.*
Status:	[rozwoju]
Właściciel:	Dyrektor finansowy
Priorytet:	*wysoki*

Tablica 17.4: Opis przypadku użycia: ocena biznesowych aspektów zapytania

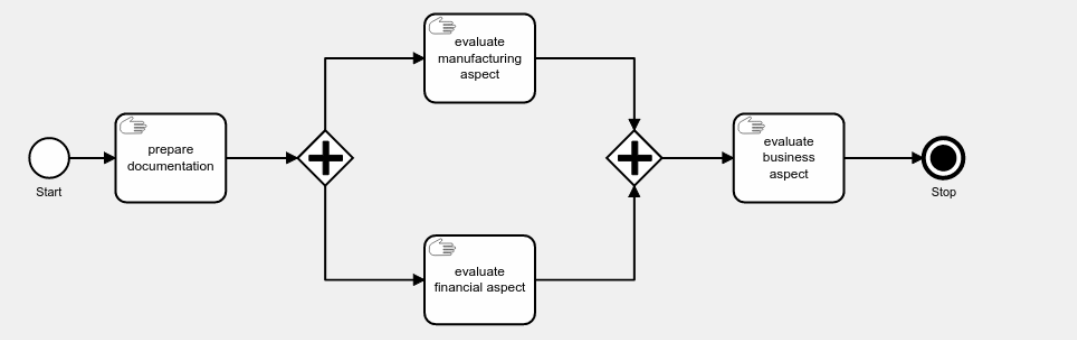
Słowo kluczowe	**Wartość**
ID:	*ch17-04*
Tytuł:	*Oceń biznesowe aspekty zapytania*
Opis:	*Dyrektor generalny i jego/z jej zespół podejmują decyzję, czy przekazać klientowi propozycję biznesową, czy też zrezygnować z oferty. Uzasadnienie opinii jest obowiązkowe i wymagane przez radę dyrektorów.*
Główny aktor:	*Dyrektor Generalny*
Warunki wstępne:	Przypadki użycia 17.1, 17.2 i 17.3 zakończone.
Warunki końcowe:	-
Główny	-
Scenariusz sukcesu:	1. wpisz opinię 2. potwierdź przyciskiem Ćonfirmłub odrzuć przyciskiem “Cancel”
Rozszerzenia:	-
Częstotliwość stosowania:	*Ok. 250 rocznie, średnio 5 tygodniowo.*
Status:	[rozwoju]
Właściciel:	Dyrektor generalny
Priorytet:	*wysoki*



Rysunek 17.2: Diagram przypadków użycia.

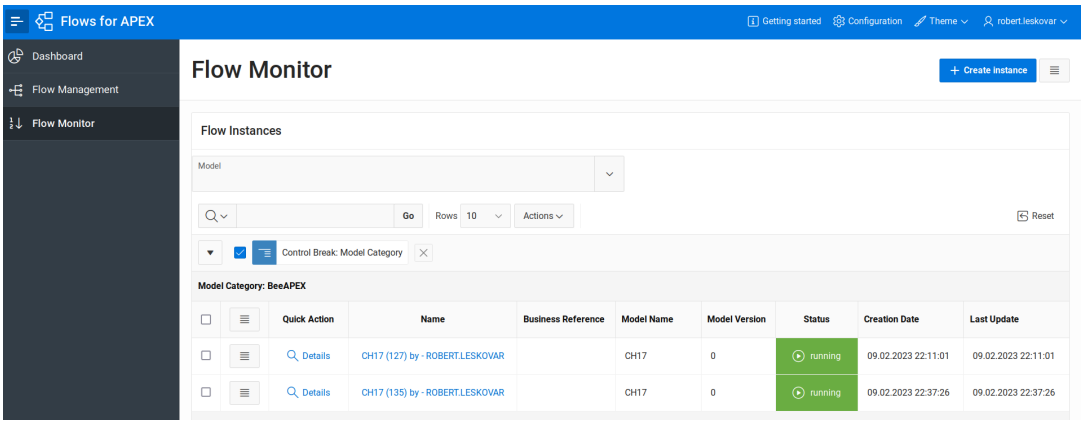
17.5 Model przepływu pracy

Początkowy model workflowu wprowadza zadanie przygotowania, które następnie dzieli się na bramkę równoległą z dwoma zadaniami (ocena produkcji i ocena finansowa). Workflow przechodzi do ostatniego zadania (ocena biznesowa), gdy oba poprzednie zadania są zakończone (zob.rys Figure 17.3).



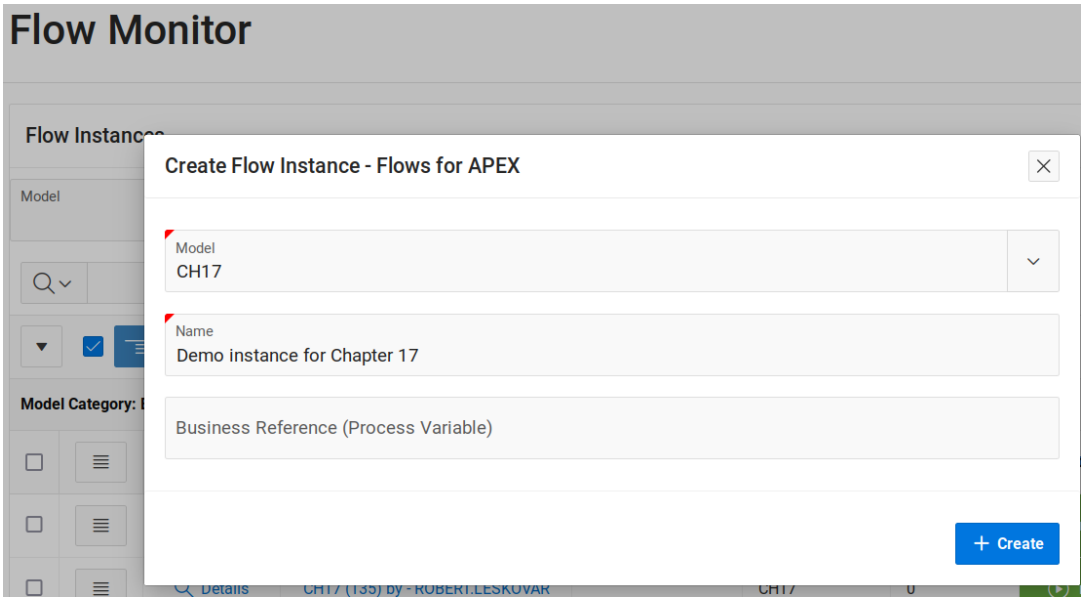
Rysunek 17.3: Workflow - przetwarzanie zapytania - diagram BPMN.

Wszystkie cztery zadania są zdefiniowane jako Źadanie manualne", a zdarzenie zakończenia (Stop) jest zdefiniowane jako Źdarzenie zakończenia przerwania". Możemy natychmiast przetestować workflow w Flow Monitor (w aplikacji "Flows for APEX"). Zapisz workflow jako CH17, a następnie przejdź do Flow Monitor. Kliknij przycisk Ćreate instance”(zob.rysunek 17.4).



Rysunek 17.4: Tworzenie instancji.

Wybierz model (CH17) i wprowadź odrębną nazwę instancji przepływu CH17 (zob.rysunek 17.5).



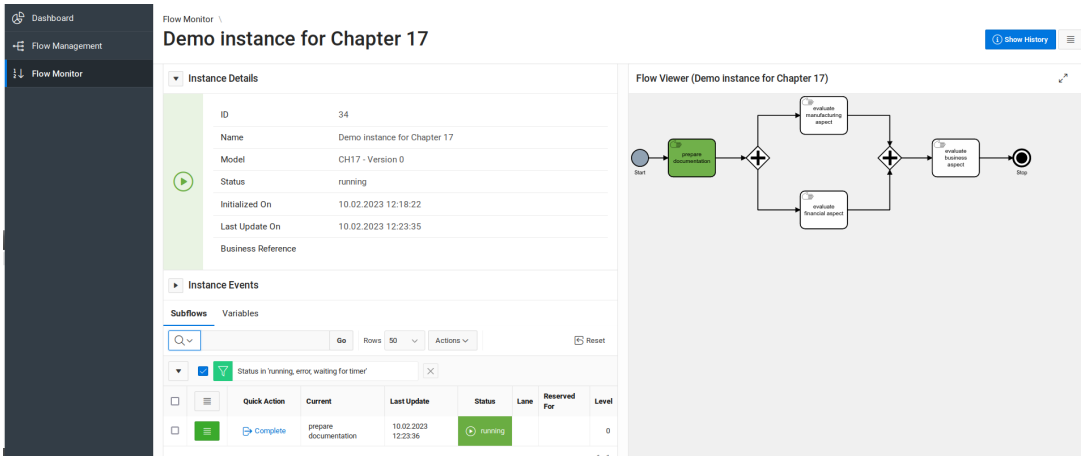
Rysunek 17.5: Tworzenie instancji przepływu CH17.

Pojawi się nowa instancja. Kliknij wskazaną ikonę na rysunku 17.6 i wybierz Śtart".



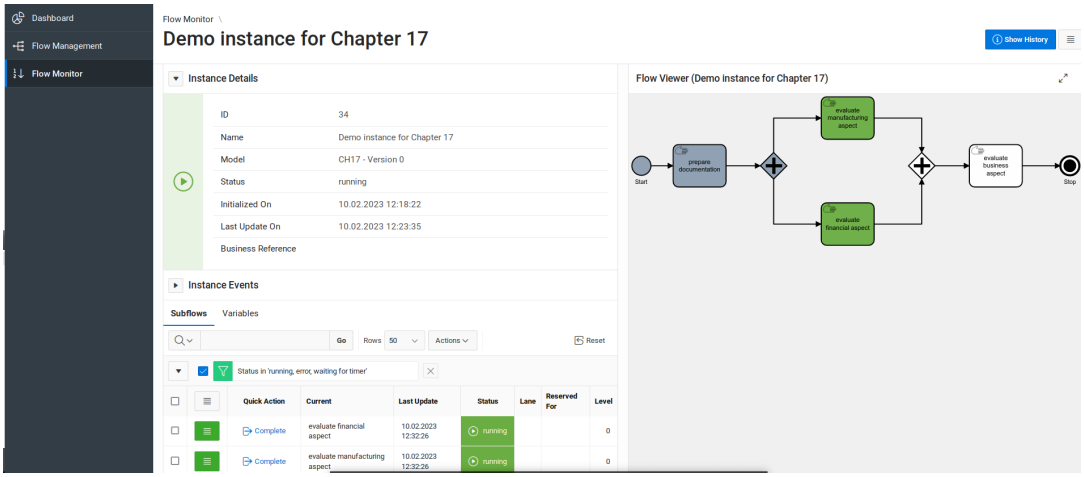
Rysunek 17.6: Uruchom instancję demonstracyjną.

Stan instancji zmieni się na żunning". Kliknij "Details" dla instancji demonstracyjnej. Ustaw ustawienia wyświetlania obok siebie na i kliknij Ćomplete" w kolumnie "Quick Action" (zob.rysunek 17.7).



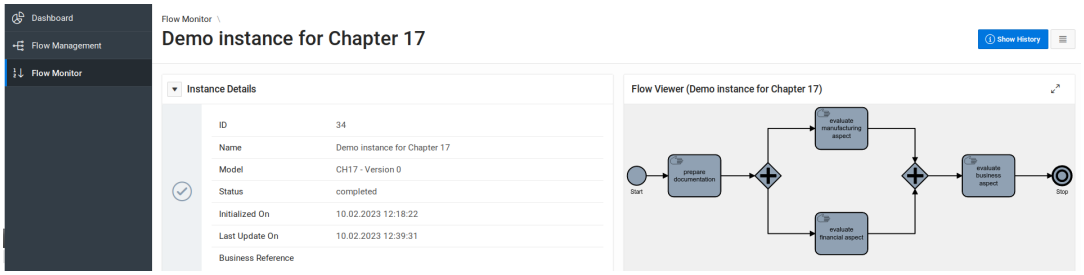
Rysunek 17.7: Wykonaj pierwsze zadanie w przykładowej instancji.

Teraz przepływ przebiega z równoległą bramą i dwoma zadaniami (zob.rysunek 17.8).



Rysunek 17.8: Wykonywanie zadań w bramie równoległej.

Kliknij Ćomplete” w kolumnie "Quick Action” dla obu zadań i kontynuuj ostatnie zadanie. Powtórz ukończenie ostatniego zadania, a instancja zostaje zakończona. (zob.rysunek 17.9).



Rysunek 17.9: Ukończona instancja demo.

Cała historia instancji zostanie wyświetlona po kliknięciu Śhow history”. Udowodniliśmy, że przepływ pracy CH17 jest wykonywalny, i możemy modelować dane i interfejs aplikacji.

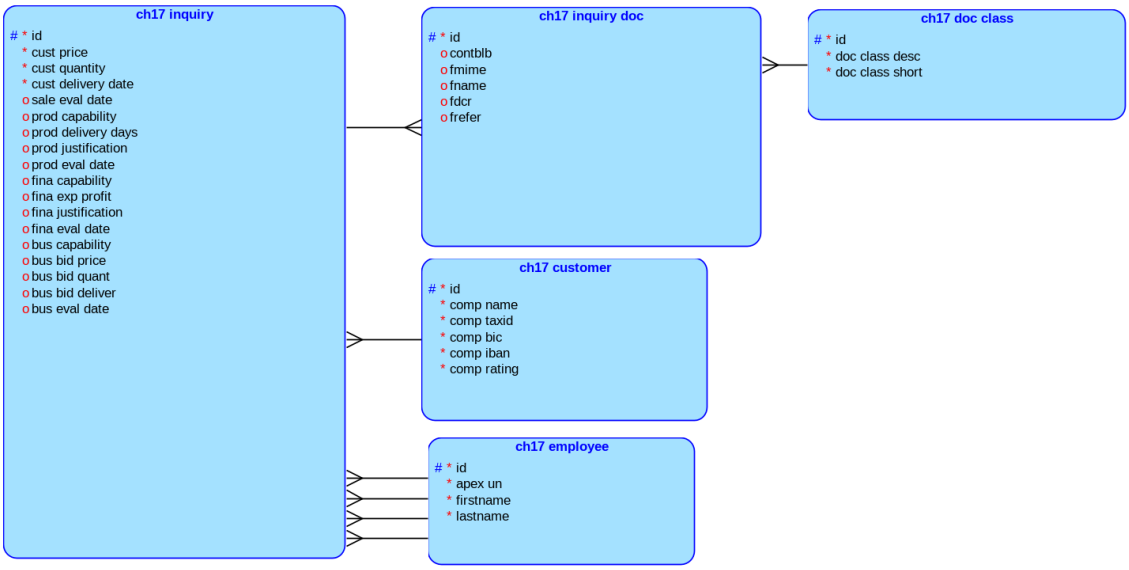
17.6 Model danych

17.6.1 Opis modelu danych

Najważniejszą jednostką jest zapytanie CH17. Instancja zapytania CH17 jest powiązana z jedną instancją klienta CH17, podczas gdy klient może mieć wiele zapytań. Każda instancja zapytania CH17 musi mieć co najmniej trzy instancje dokumentów CH17. Każda instancja dokumentu CH17 należy do jednej instancji klasy dokumentu CH17. Atrybuty jednostki zapytania CH17 to: ID, wymagania klienta (cena za zamówienie, ilość, data dostawy), aspekty produkcyjne (zdolności, czas dostawy dla wymaganej ilości, uzasadnienie), aspekty finansowe (zdolność finansowa, oczekiwany zysk dla wymaganej ilości) oraz aspekty biznesowe (decyzja o podjęciu lub odrzuceniu okazji, uzasadnienie). Ograniczymy zdefiniowanie atrybutów dla innych jednostek, aby zachować kompaktowość tego przypadku.

17.6.2 Logiczny model danych

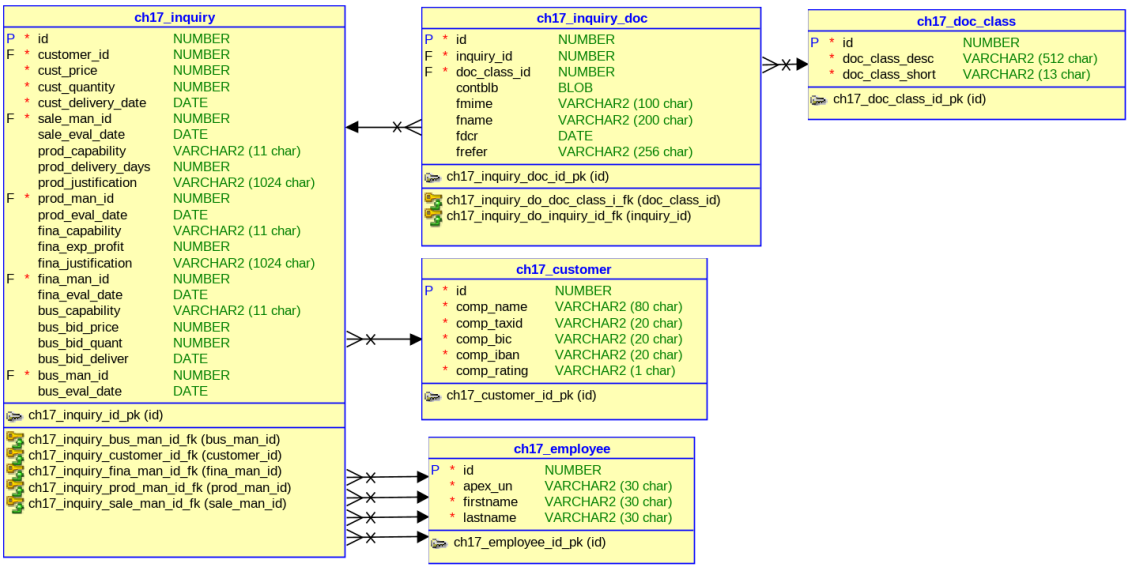
Logiczny model danych przedstawiono na rysunku 17.10.



Rysunek 17.10: Logiczny model danych.

17.6.3 Relacyjny model danych

Relacyjny model danych przedstawiono na rysunku 17.11.



Rysunek 17.11: Relacyjny model danych.

17.6.4 Quick SQL do generowania skryptu SQL

Quick SQL to przydatne narzędzie, które przyspiesza rozwijanie modelu danych. Może być używane do prototypowania poprzez generowanie skryptu SQL w środowisku APEX (tworzenie aplikacji na podstawie skryptu) lub do przeprowadzania inżynierii wstecznej logicznego modelu danych poprzez importowanie skryptu SQL do SQL Developer Data Modeler i przekształcanie modelu relacyjnego na model logiczny. Model danych, przedstawiony w tym rozdziale, może być opisany za pomocą poniższego Quick SQL:

```
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
```

Jednym z korzyści Quick SQL jest jego zwartość i prostota. Liczba linii w Quick SQL wynosi połowę skryptu SQL, a liczba znaków wynosi mniej niż jedną trzecią skryptu SQL. Czas i wysiłek poświęcone na naukę Quick SQL to dobra inwestycja. APEX dostarcza także przyjaznego dla użyt-

kownika edytora z automatycznym uzupełnianiem, jednak korzyści z Quick SQL lepiej wykorzysta użytkownik, który zna SQL.

17.6.5 Sekwencja i dwie przechowywane funkcje

W tym przypadku biznesowym wymagania aplikacji zmuszają nas do korzystania z kilku łatwych do zrozumienia koncepcji bazy danych Oracle:

- mechanizm sekwencji - generuje liczby całkowite, określając początkową wartość, maksymalną wartość, przyrost itp. Dzięki temu mechanizmowi nasza aplikacja będzie generować unikalne numery do identyfikacji zapytań i komponować nazwę instancji workflowu.
- funkcje składowane w języku PL/SQL, przyjmujące parametry wejściowe i zwracające jedną wartość. Nasza aplikacja będzie korzystać z dwóch zdefiniowanych przez użytkownika funkcji: jednej do uzyskiwania rozmiaru dokumentu przechowywanego w kolumnie BLOB oraz drugiej do zliczania liczby dokumentów powiązanych z konkretnym zapytaniem.

Możemy zdefiniować sekwencję i funkcje składowane w SQL Workshop > SQL Command. Sekwencja ma nazwę, wartość minimalną, wartość maksymalną, przyrost, wartość początkową i kilka innych właściwości, które w obecnej chwili nie są dla nas istotne.

```
CREATE SEQUENCE   "CH17_SEQ_INQUIRY"  MINVALUE 1 MAXVALUE 999999999999
  INCREMENT BY 1 START WITH 100 CACHE 1
  NOORDER  NOCYCLE  NOKEEP  NOSCALE  GLOBAL;
```

Zdefiniujemy również dwie przechowywane funkcje, które są wyjaśnione jako komentarze w kodzie - dwie myślniki na początku wierszy są komentarzami. Pierwsza funkcja nazywa się 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;
```

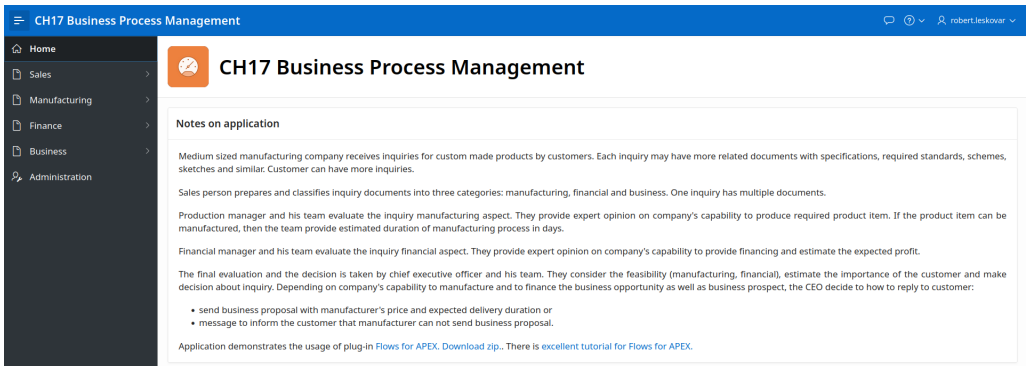
Druga funkcja nazywa się ch17_count_docs:

```
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;
```

17.7 Interfejsy aplikacji

Sugerujemy czytelnikowi zainstalowanie spakowanej aplikacji dla tego rozdziału przed zaprojektowaniem aplikacji. Link znajduje się w Dodatkowych materiałach do nauki do tego rozdziału. Podane numery stron, odnoszą się do przygotowanej aplikacji pakietowej. Czytelnik może oczywiście opracowywać i używać innych numerów stron, aby zapewnić taką samą funkcjonalność, jak opisano w tej sekcji. Interfejs aplikacji zapewnia funkcjonalności dla czterech podmiotów: kierownika sprzedaży, kierownika produkcji, kierownika finansowego i dyrektora generalnego. Został po raz pierwszy wygenerowany za pomocą kreatora i wszystkich zawartych funkcji (progresywny, aplikacja internetowa, o stronie, kontrola dostępu, raportowanie aktywności, opcje konfiguracji, opinie i wybór stylu motywu, łącznie 21 stron). Domyślnie generowana jest również strona logowania. Strona główna zawiera czysty HTML opisujący sytuację biznesową. Zawiera linki do wymaganej wtyczki i samouczka (zob.rysunek 17.12, strona 1 w aplikacji pakietowe).



Rysunek 17.12: Strona główna aplikacji ĆH17 Business Process Management".

Niektóre LOV-y i wtyczki zostały skonfigurowane lub obejmowały wcześniejsze tworzenie aplikacji w węższym znaczeniu.

17.7.1 Lista wartości w komponentach współdzielonych

W tej aplikacji zdefiniowane są następujące listy wartości:

- CH17_LOV_CAPABILITY: typ to statyczny, wartości wyświetlane to (TAK, NIE, WARUNKOWY, Niedostępny), wartości zwracane to (TAK, NIE, WARUNKOWY, N.A.). Ta lista wartości jest używana do wypełnienia trzech różnych zdolności (produkcyjnej, finansowej i biznesowej).
- CH17_LOV_CRATING: typ to statyczny, wartości wyświetlane to (Ocena doskonała, Ocena dobra, Ocena dopuszczalna, Ocena niedopuszczalna), wartości zwracane to (A, B, C, D). Ta lista wartości jest używana do wprowadzania oceny klienta.
- CH17_LOV_CUSTOMER: to typ dynamiczny, oparty na tabeli CH17_CUSTOMER, war-

tość wyświetlana to DOC_CLASS_DESC , a wartość zwracana to ID. Ta lista wartości jest używana w formularzach i raportach, aby przyczynić się do czytelności i zapobiec wprowadzaniu nieprawidłowego identyfikatora klienta.

- CH17_LOV_DOC_CLASS: typ to dynamiczny, oparty na tabeli CH17_DOC_CLASS, wartość wyświetlana to COMP_NAME , a wartość zwracana to ID. Ta lista wartości jest używana w formularzach i raportach, aby przyczynić się do czytelności i zapobiec wprowadzaniu nieprawidłowej klasy dokumentu związanego z zapytaniem.
- CH17_LOV_EMPLOYEE: typ to dynamiczny, oparty na zapytaniu SQL na tabeli CH17_EMPLOYEE:

```
SELECT FIRSTNAME || ' ' || LASTNAME as d,
id as r
FROM CH17_EMPLOYEE order by LASTNAME, FIRSTNAME
```

Ten LOV jest używany w formularzach i raportach, aby przyczynić się do czytelności i zapobiec wprowadzeniu niewłaściwej osoby, która przygotowuje dokumentację i ocenia zapytanie. Operator Concatenate służy do tworzenia ciągu wyświetlanego z imieniem, spacją i nazwiskiem pracownika.

- CH17_LOV_INQUIRY: typ jest dynamiczny, oparty na zapytaniu SQL na CH17_INQUIRY: tabeli

```
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;
```

Ten LOV jest używany w formularzach i raportach, aby przyczynić się do czytelności i zapobiec wprowadzaniu nieprawidłowego identyfikatora zapytania. Operator Concatenate służy do tworzenia długiego ciągu wyświetlania z dużą ilością informacji.

17.7.2 Wtyczki w komponentach współużytkowanych

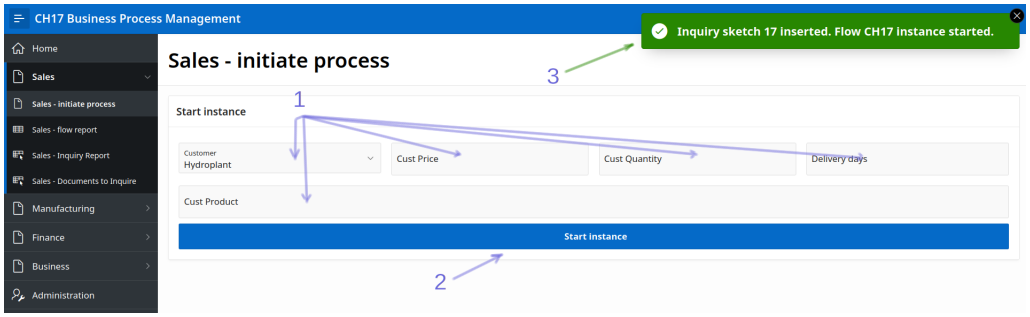
Do usługi Flows for APEX są importowane następujące wtyczki: Manage Flow Instance, Manage Flow Instance Step, Manage Flow Instance Variables i Viewer. Te dodatki są kopiowane z zainstalowanej wtyczki Flows for APEX.

17.7.3 Kierownik sprzedaży

Interfejs aplikacji pozwala kierownikowi ds. sprzedaży na:

- rozpoczęcie instancji workflowu (zob.rysunek 17.13, Strona 2 w aplikacji)
- monitorowanie workflowu (zob.rysunek 17.14,Strona 5 w aplikacji)
- potwierdzenie, że wszystkie dokumenty do dalszych ocen są przechowywane w bazie danych (zob.rysunek 17.15, Strona 4 w aplikacji)
- pozyskanie wglądu we wszystkie zapytania (zob.rysunek 17.17, Strona 6 w aplikacji)
- pozyskanie wglądu we wszystkie dokumenty związane z zapytaniami (zob.rysunek 17.17, Strona 8 w aplikacji)
- przesyłanie dokumentów do odpowiedniej kolumny w dokumencie zapytania (zob.rysunek 17.18 i 17.20, Strona 8 i 9 w aplikacji)

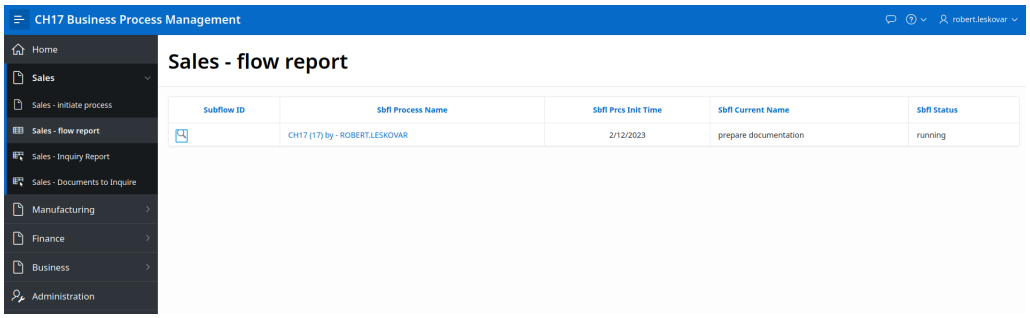
Komentarze dotyczące Rysunku 17.13 (numery i strzałki) przedstawiają sekwencję działań: 1 (użytkownik wprowadza dane), 2 (użytkownik potwierdza wprowadzone dane) i 3 (wyświetlane jest komunikat o sukcesie lub niepowodzeniu, pola danych są czyścić do wprowadzenia nowych danych).



Rysunek 17.13: Sprzedaż - proces zainicjowany.

Oprócz widocznych komponentów istnieje także ukryty element strony ((P2_WID) który po prostu pobiera następną wartość numeryczną z sekwencji CH17_SEQ_INQUIRY . To było powodem, dla którego zdefiniowaliśmy sekwencję CH17_SEQ_INQUIRY przed projektowaniem strony. Przycisk Śtart"wykonuje dwie procesy: : Ądd inquiry"(Dodaj zapytanie) dodaje jeden wiersz do tabeli CH17_INQUIRY i cztery wiersze do tabeli CH17_INQUIRY_DOC,podczas gdy proces Ćreate and start"(Utwórz i uruchom) wykonuje akcję z wtyczki Flows for APEX (szczegóły można znaleźć w aplikacji pakietowej, Strona 2, Procesy). Komunikat o udanym zakończeniu jest zwracany, jak widać na Rysunku 17.13.

Raport przepływu dla sprzedaży przedstawiony na rysunku 17.14.



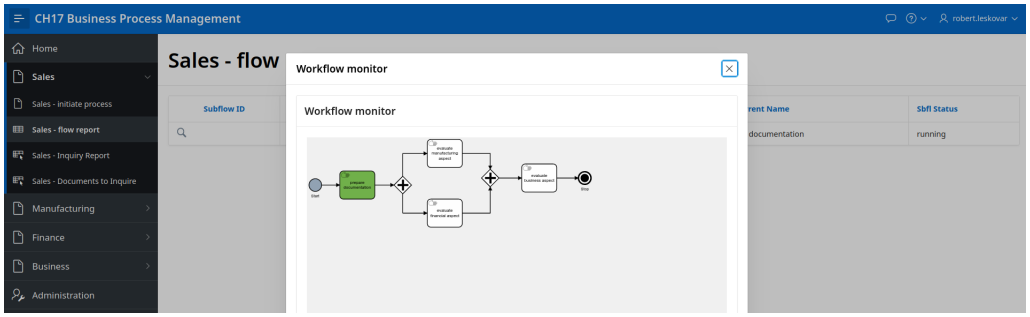
Rysunek 17.14: Raport przepływu sprzedaży.

Raport przepływu dla sprzedaży jest klasycznym raportem opartym na zapytaniu SQL:

```
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'
```

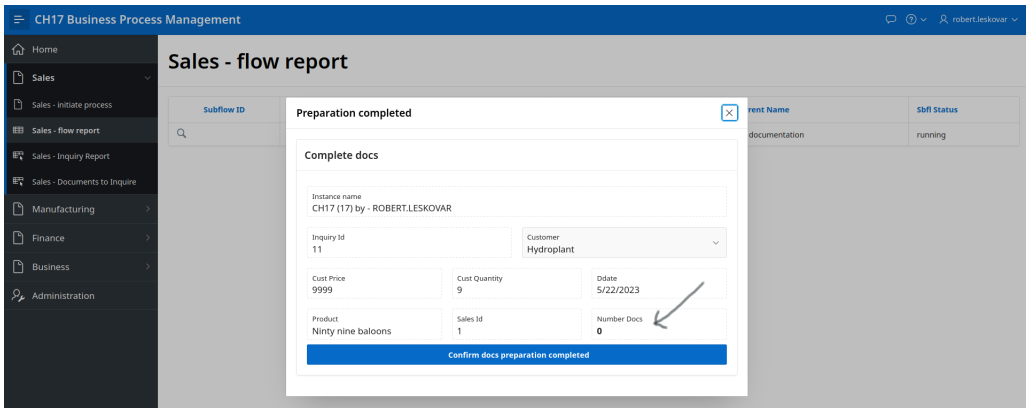
Pokazuje tylko instancje procesu ĆH17", które mają aktualną nazwę "przygotowanie dokumentacji". Dwa linki znajdują się na stronie 4 w aplikacji(zob.rysunek 17.14):

1. ikona soczewki w kolumnie Śubflow ID"pokazuje stan instancji na diagramie przepływu pracy. (zob.rysunek 17.15,strona 5 aplikacji).



Rysunek 17.15: Raport Sales - flow dla wybranej instancji.

2. Kolumna Śbfl Process Name"wyświetla modalne okno dialogowe, w którym sprzedaż może potwierdzić, że dokumentacja jest gotowa. (zob.rysunek 17.16, strona 7 w aplikacji).



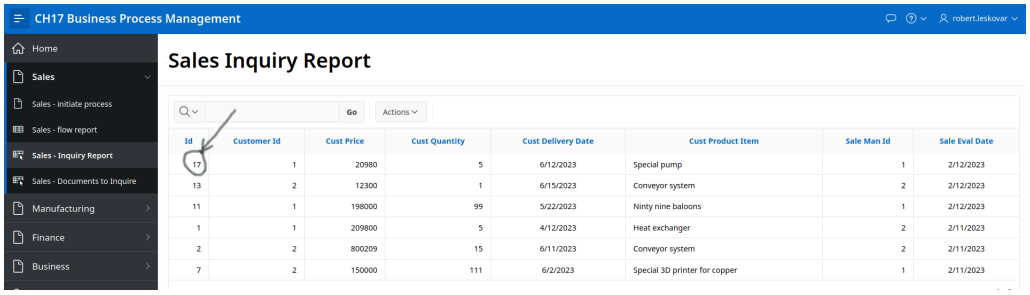
Rysunek 17.16: Raport Sales - flow dla wybranej instancji.

W tym momencie dokumentacja nie jest jeszcze przygotowana (przesłano 0 dokumentów). Licznik jest wynikiem bloku PL/SQL jako źródła:

```
BEGIN
RETURN CH17_COUNT_DOCS(:P7_INQUIRY_ID);
END;
```

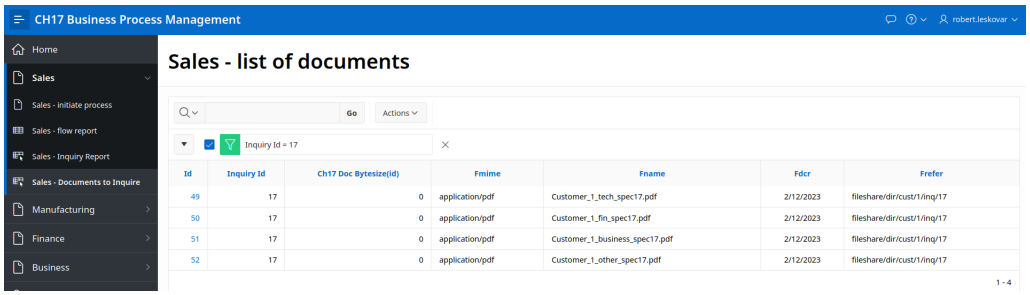
Funkcja CH17_COUNT_DOCS w Object Browser > Functions) liczy wszystkie duże obiekty binarne (BLOB), przechowywane w tabeli CH17_INQUIRY_DOC które są związane z konkretnym zapytaniem.. Jeśli rozmiar BLOB jest większy niż zero, to treść jest przechowywana, w przeciwnym razie istnieje tylko pusty blob. Zgodnie z zasadami biznesowymi przynajmniej trzy dokumenty muszą być przygotowane (dla produkcji, finansów i biznesu).

Aby uzyskać wgląd we wszystkie zapytania, przygotowany jest interaktywny raport dla działu sprzedaży (zob.rysunek 17.17, strona 6 w aplikacji).



Rysunek 17.17: Sprzedaż - raport z zapytania.

Raport z zapytaniami sprzedażowymi oferuje wyszukiwanie oraz różne akcje (wybieranie kolumn, zmienianie kolejności, filtrowanie, formatowanie, tworzenie wykresów, generowanie raportów, eksportowanie itp.). Celem tego raportu jest dostarczenie kierownikowi ds. sprzedaży identyfikatora zapytania, który może być wykorzystywany w procesie przygotowywania dokumentacji (np. 17, jak wskazano na Rysunku 17.17). Następnym krokiem dla kierownika ds. sprzedaży byłoby przygotowanie specyfikacji technicznej, finansowej, biznesowej i innych (patrz Rysunek 17.18 , Strona 8 w aplikacji).



Rysunek 17.18: Sprzedaż - lista dokumentów.

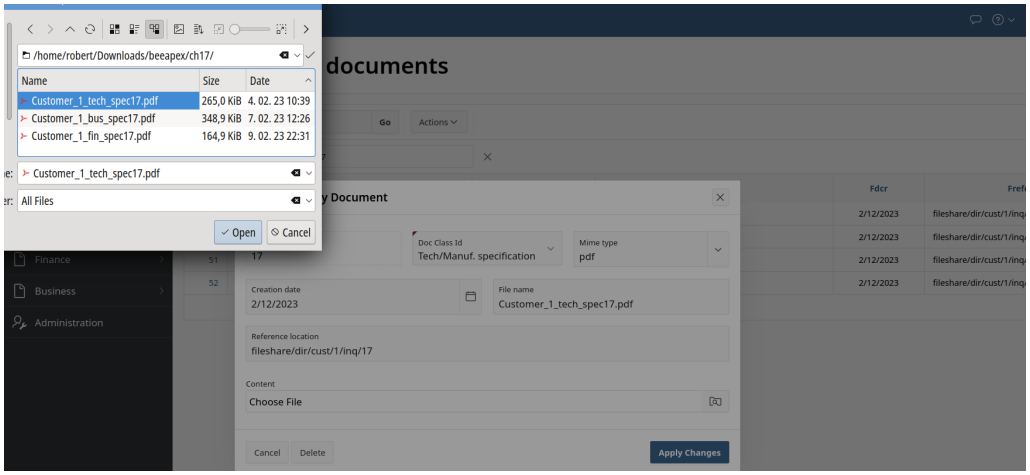
Rysunek 17.18 pokazuje przefiltrowane wyniki (identyfikator zapytania to 17). Nie przesłano BLOB - kolumna Ćh17 Doc Bytesize(id)"pokazuje wszystkie zera. Strona 8 to interaktywny raport oparty na następującym zapytaniu SQL:

```
select ID, INQUIRY_ID,DOC_CLASS_ID, CH17_DOC_BYTESIZE(ID), CONTBLB,
       FMIME,FNAME, FDCR, FREFER
from CH17_INQUIRY_DOC
```

Funkcja CH17_DOC_BYTESIZE zwraca rozmiar obiektu BLOB. Funkcja przyjmuje identyfikator CH17_INQUIRY_DOC jako parametr wejściowy i jest zdefiniowana w następujący sposób:

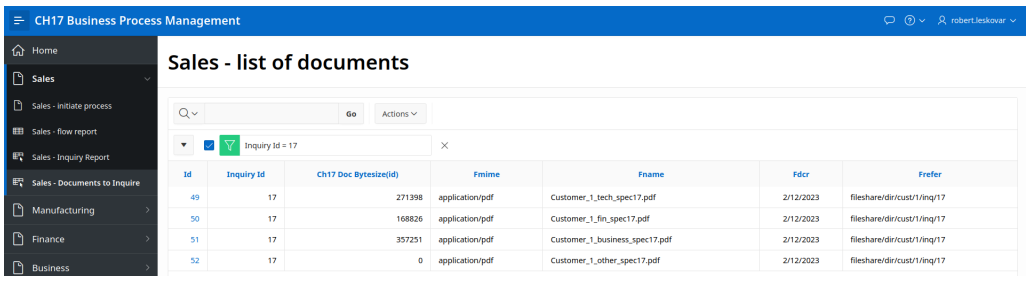
```
create or replace function CH17_DOC_BYTESIZE(xdocid in number) return number
is
  wdoc_size number := 0;
  wblob blob;
Begin
  select contblb into wblob from ch17_inquiry_doc where id = xdocid;
  wdoc_size := dbms_lob.getlength(wblob);
  RETURN  wdoc_size;
END;
```

Identyfikator tabeli CH17_INQUIRY_DOC służy jako link do otwarcia strony modalnej Strona 9 w aplikacji (zob.rysunek 17.19).



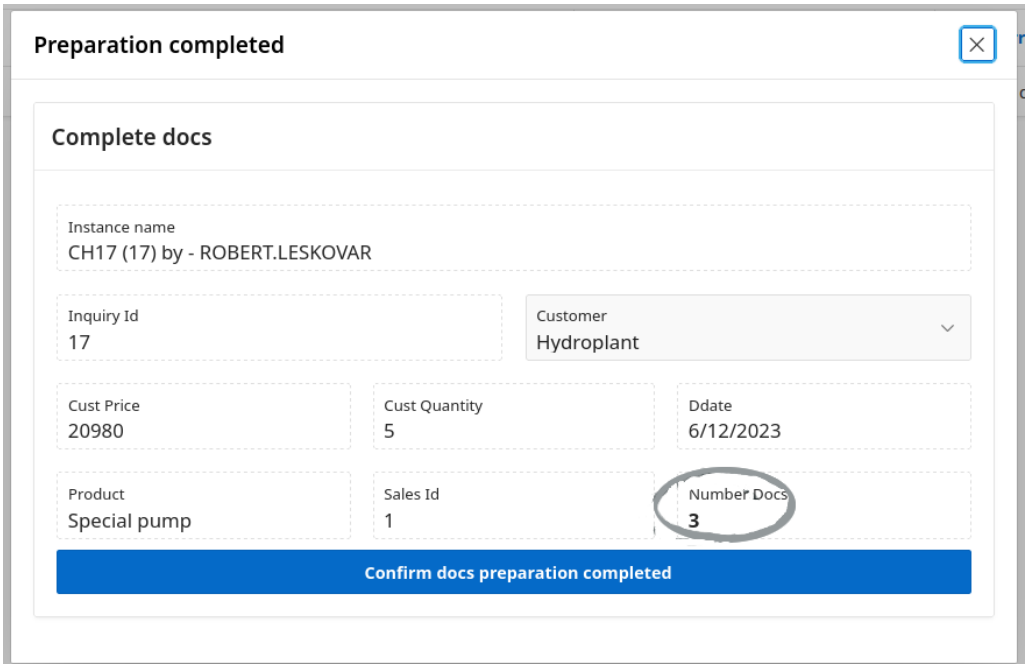
Rysunek 17.19: Sprzedaż - przesłanie dokumentu do zapytania.

Poprzez odświeżenie Strony 8 użytkownik może upewnić się, że dokumenty zostały przesłane (patrz Rysunek 17.20 , Strona 8 w aplikacji).



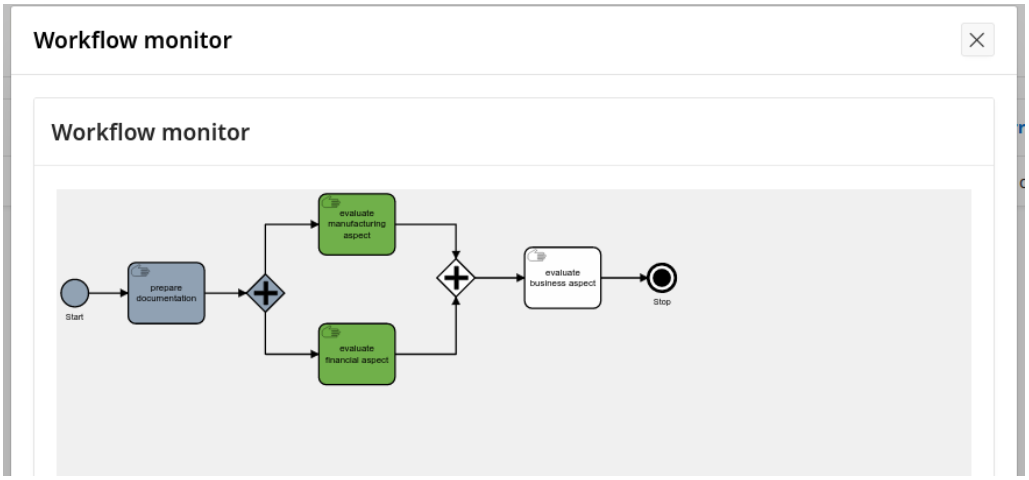
Rysunek 17.20: Sprzedaż - lista dokumentów po przesłaniu i odświeżeniu.

Teraz dział sprzedaży potwierdzi, że zadanie przygotowawcze zostało zakończone. Otwierając raport z działu sprzedaży i klikając link na nazwie procesu, otwiera się strona modalna (patrz Rysunek 17.21 , Strona 8 w aplikacji).



Rysunek 17.21: Sprzedaż - raport dla wybranej instancji po przesłaniu trzech dokumentów.

Klikając Śubflow ID", kierownik sprzedaży otrzymuje wizualne potwierdzenie, że zadanie przygotowania dokumentu zostało zakończone (zob.rysunek 17.22, strona 5 aplikacji).



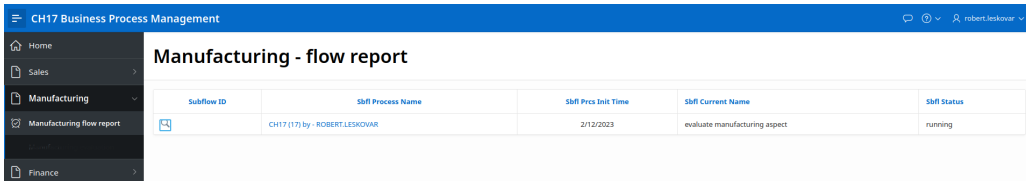
Rysunek 17.22: Sales - raport przepływu dla wybranej instancji po przesłaniu trzech dokumentów.

17.7.4 Kierownik produkcji

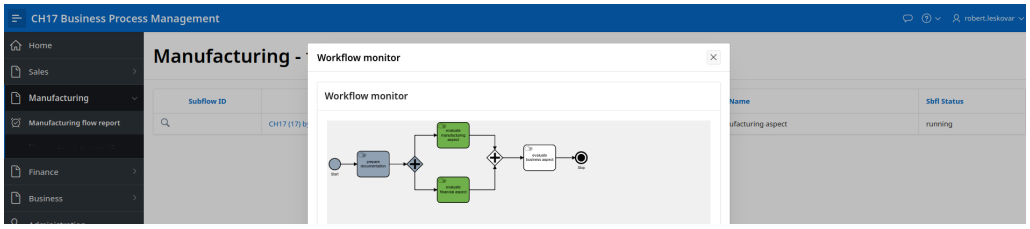
Kierownik produkcji może:

- monitorować workflow (report na rysunku 17.23, Strona 20 w aplikacji i diagram na Rysunku 17.24, Strona 5 w aplikacji)
- wprowadzać oceny produkcyjne (zob.rysunek 17.25, strona 21 w aplikacji)

Klikając Żaport workflowu produkcyjnego", kierownik produkcji może śledzić nieszacowane zapytania (czyli raport na Rysunku 17.23 , Strona 20 w aplikacji).

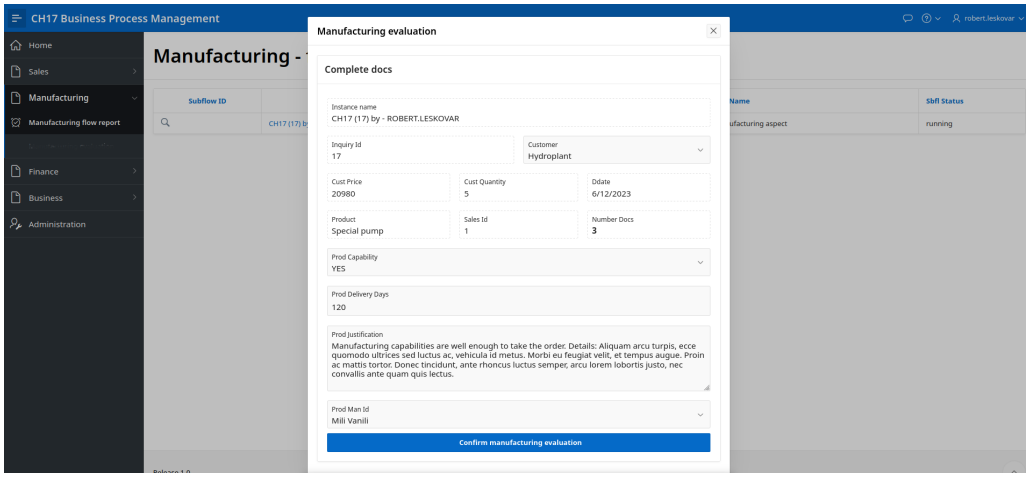


Rysunek 17.23: Raport przepływu produkcji - produkcja.



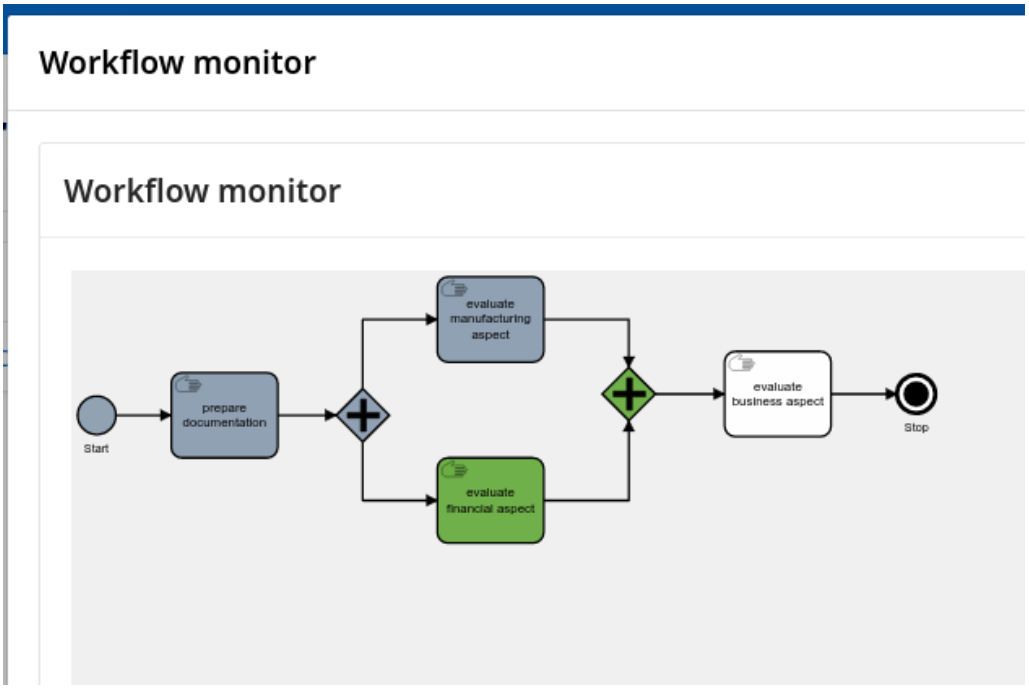
Rysunek 17.24: Produkcja - diagram BPMN - stan instancji.

Wybierając ikonę soczewki pod Śubflow IDótwiera się Strona 5 (patrz Rysunek 17.24), która przedstawia instancję workflowu CH17. Instancja znajduje się w bramce równoległej, co oznacza, że oba zadania (ocena produkcyjna i ocena finansowa) muszą być zakończone, zanim rozpocznie się kolejne zadanie (ocena biznesowa). Ocena produkcyjna może być wprowadzona, klikając link Śbfl Process Name"(Nazwa procesu podpopytu) na Stronie 20 (patrz Rysunek 17.25, Strona 21 w aplikacji).



Rysunek 17.25: Ocena produkcji.

Naciśnięcie przycisku Ćonfirm manufacturing evaluation” spowoduje przesunięcie przepływu instancji w następny stan, przedstawiony na rysunku 17.26.



Rysunek 17.26: Ocena produkcji zakończona, instancja oczekująca na ocenę finansową.

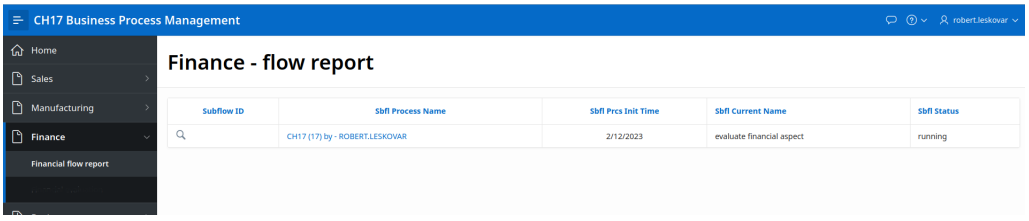
Po odświeżeniu strony 20 raport wyświetla komunikat "Manufacturing has no flow instances"

17.7.5 Dyrektor finansowy

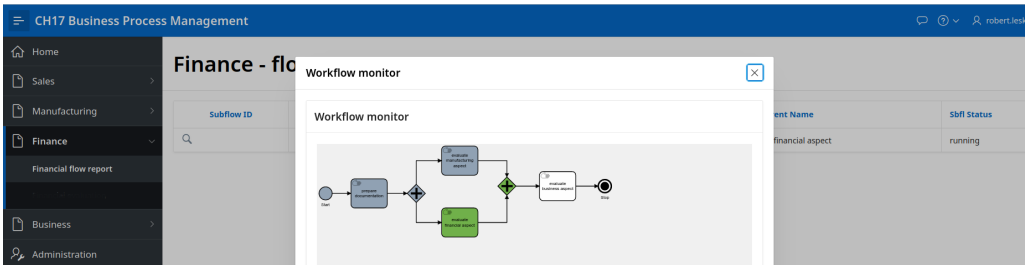
Menedżer finansowy może:

- monitorować workflow (raport na Rysunku 17.27, Strona 30 w aplikacji i diagram na Rysunku 17.28 , Strona 5 w aplikacji).
- wprowadzać oceny finansowe (Rysunek 17.29 , Strona 31 w aplikacji).

Klikając Żaport workflowu finansowego", kierownik finansowy może śledzić nieszacowane zapytania (czyli raport na Rysunku 17.27 , Strona 30 w aplikacji).

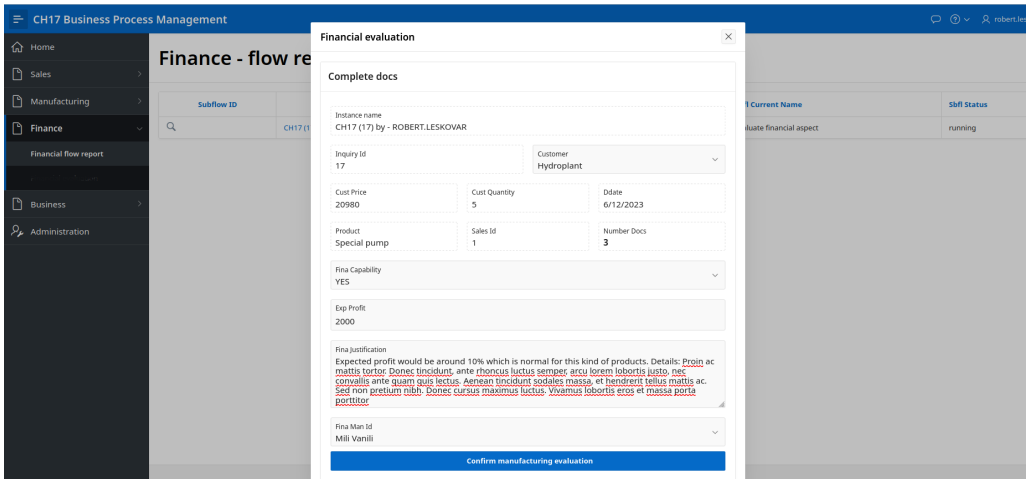


Rysunek 17.27: Sprawozdanie finansowe - przepływ.



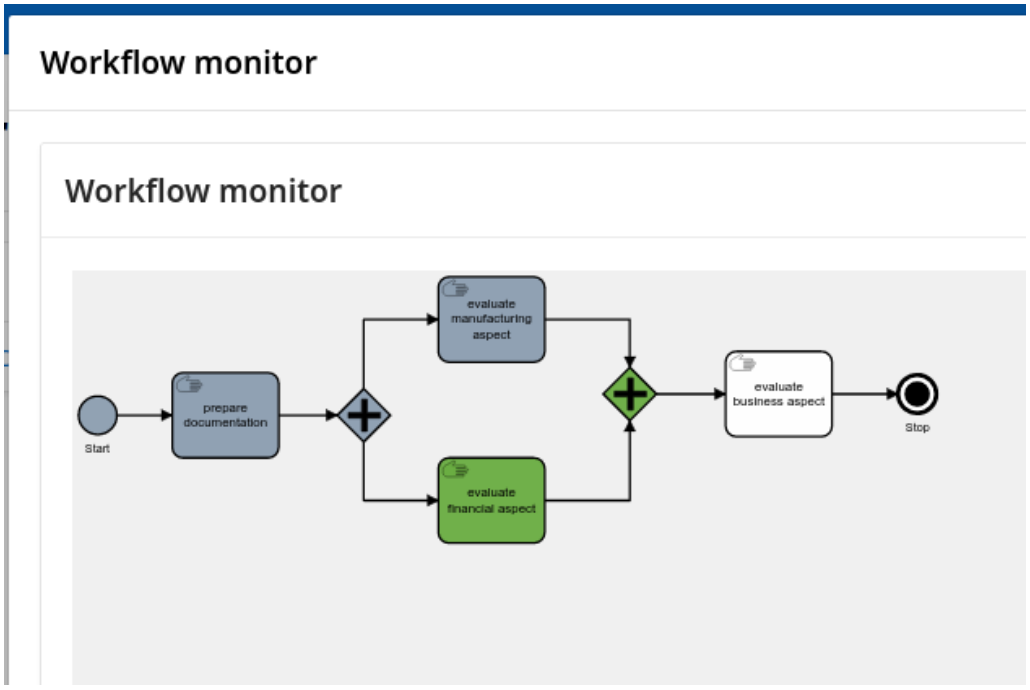
Rysunek 17.28: Finanse - diagram BPMN - stan instancji.

Klikając ikonę obiektywu pod Śubflow ID", aplikacja otwiera stronę 5 (patrz rysunek 17.28), która pokazuje instancję przepływu CH17. Ocenę finansową można wprowadzić, klikając link Śbfl Process Name na stronie 30 (patrz rysunek 17.29, strona 31 we wniosku).



Rysunek 17.29: Ocena finansowa.

Naciśnięcie przycisku Ćonfirm financial evaluation"przeniesie przepływ instancji w następny stan, przedstawiony na rysunku 17.30 .



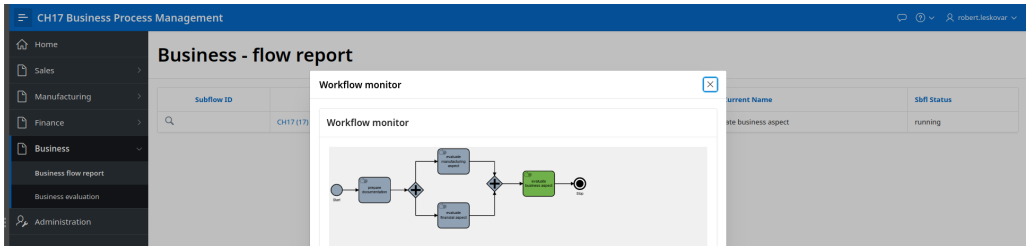
Rysunek 17.30: Ocena finansowa zakończona, instancja oczekuje na ocenę biznesową.

Po odświeżeniu strony 30 raport wyświetla komunikat "Finance has no flow instances"(Finanse nie mają wystąpień przepływu).

17.7.6 Dyrektor Generalny - dyrektor handlowy

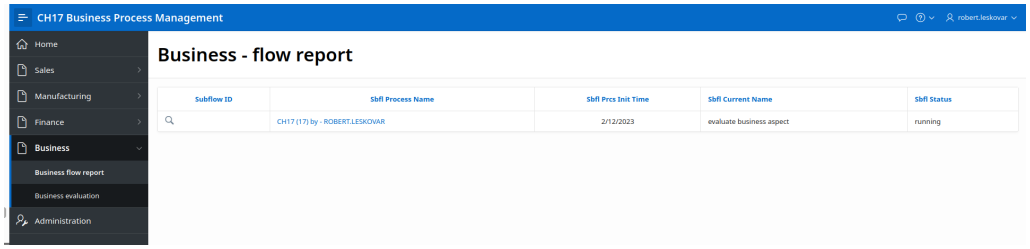
Menedżer firmy może:

- monitorować przepływu pracy (sprawozdanie na rysunku 17.32, strona 40 w aplikacji i diagram na rysunku 17.31, Strona 5 w aplikacji)
- wprowadzać ocenę biznesową (rysunek 17.33 , strona 41 w aplikacji)



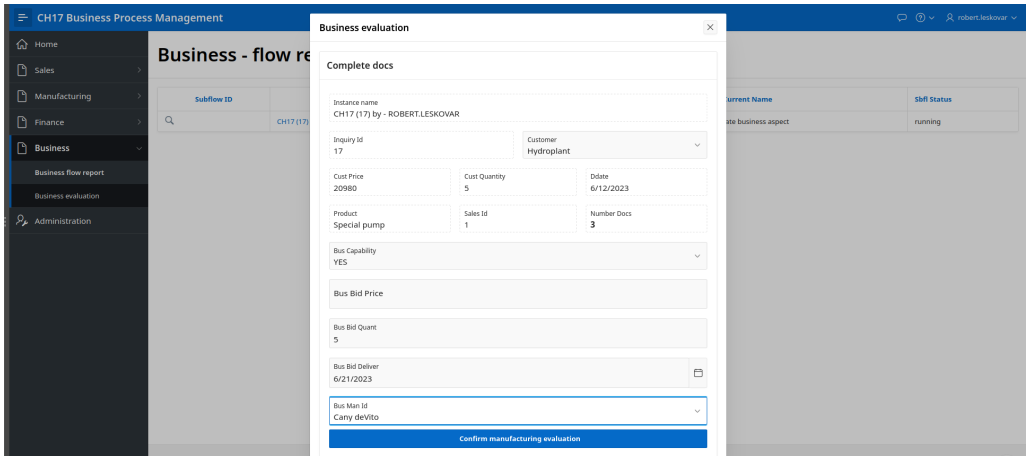
Rysunek 17.31: Finanse - diagram BPMN - stan instancji.

Klikając "Business flow reportĆEO może śledzić nieocenione zapytania (tj. raport na rysunku 17.32 , strona 40 w aplikacji).



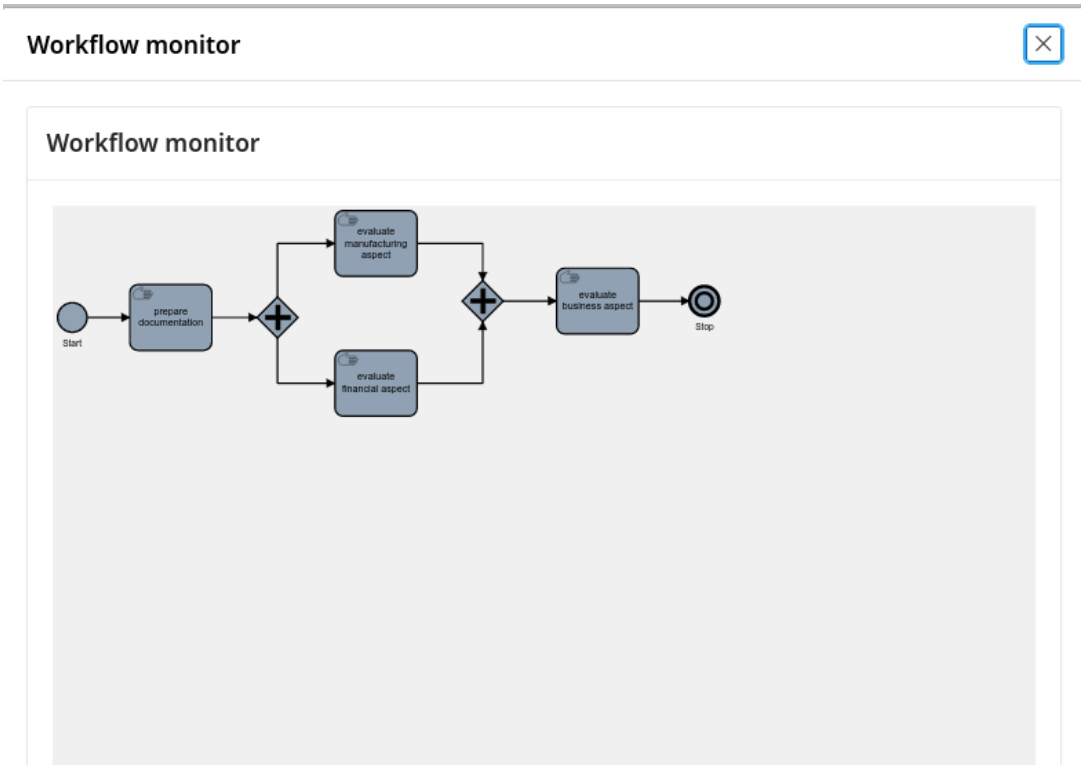
Rysunek 17.32: Raport przepływu - Biznes.

Kliknięcie ikony obiektywu w sekcji Śubflow IDśpowoduje otwarcie strony 5 (patrz rysunek 17.31), która pokazuje wystąpienie przepływu CH17. Ocenę przedsiębiorstwa można wprowadzić, klikając link Śbfl Process Name ńa stronie 40 (patrz rysunek 17.33, strona 41 w aplikacji).



Rysunek 17.33: Ocena biznesowa.

Naciśnięcie przycisku Ćonfirm business evaluationśpowoduje przesunięcie instancji na koniec przepływu pracy, jak pokazano na rysunku 17.34.



Rysunek 17.34: Ocena biznesu zakończona, instancja zakończona.

By refreshing Page 40, report displays "Business has no flow instances".

17.8 Łączenie aplikacji z usługą Flows for APEX

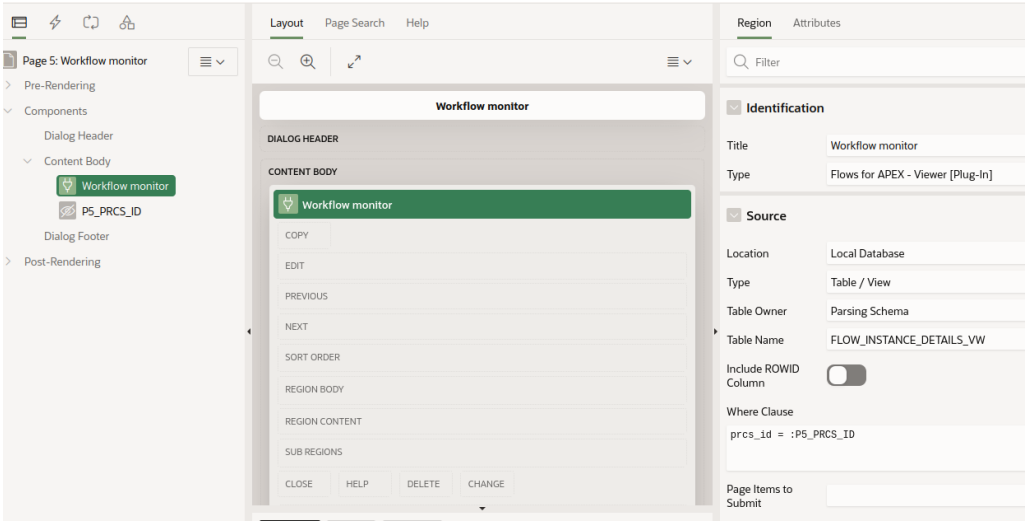
Aplikacja, opracowana w rozdziale 17, wykorzystywała dwie funkcje dostarczane przez Flows dla APEX:

- uruchamianie instancji na stronie 2
- wyświetlanie stanu instancji na stronie 5
- przesyłanie instancji do następnego zadania na stronach 7, 21, 31, 41

Uruchomienie instancji wymaga procesu na stronie. W obszarze identyfikacji procesu: "Flows for APEX - Managing Flow Instance [Plug-In]". Akcja powinna mieć postać Ćreate and Start"(Utwórz i uruchom). Ustawiamy Flow instance Name na:

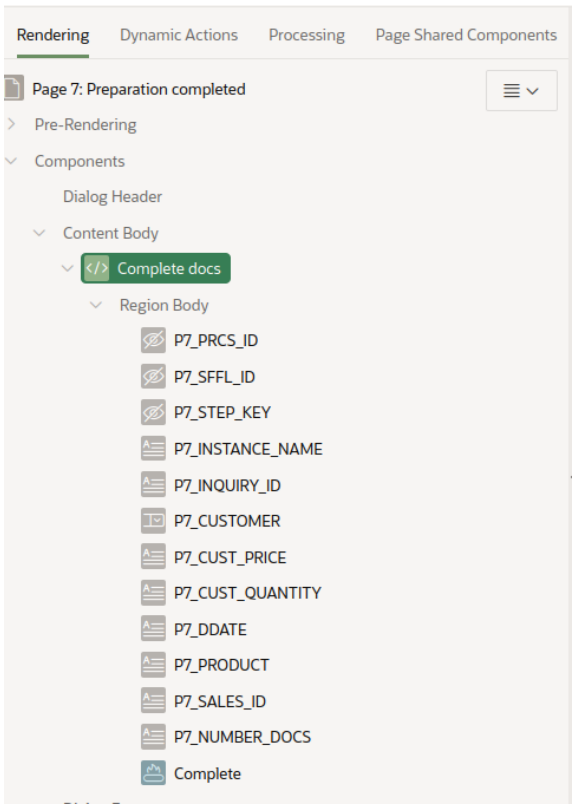
```
CH17 (&P2_WID.) by - &APP_USER.
```

Pamiętaj, że element strony P2_WID zawiera unikalny numer, dostarczany przez sekwencję. Nazwa jest konkatenacją stałego ciągu ĆH17", spacji, nawiasu otwierającego, unikalnego numeru sekwencji, nawiasu zamykającego, spacji, stałego ciągu "by - "i aktualnego użytkownika. Wybór przepływu używa "Tekstu statycznego", a Tekst statyczny to ĆH17". Komunikat o sukcesie to "Flow CH17 instance started"(Uruchomiono instancję przepływu CH17), a komunikat o błędzie to "Flow CH17 instance NOT started."(Nie uruchomiono instancji przepływu CH17). Aby przedstawić stan instancji na stronie 5, używamy kreatora strony dla strony wtyczki. Strona zawiera wtyczkę przeglądarki i ukryty element - P5_PRCS_ID. Zobacz ustawienia na Rysunku 17.35.



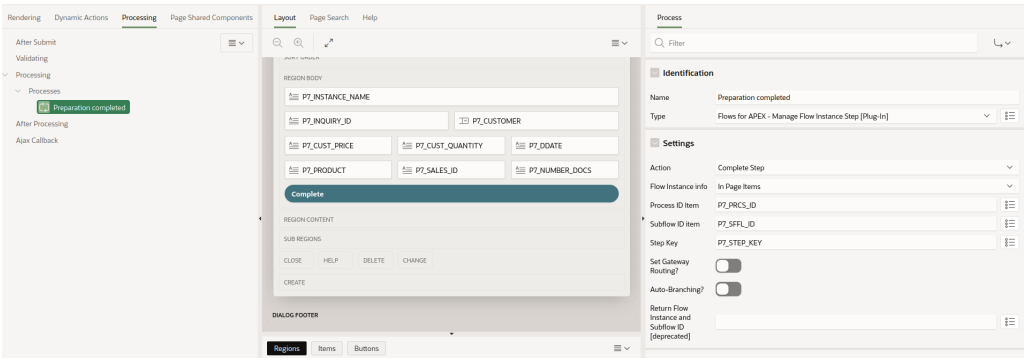
Rysunek 17.35: Pokazanie stanu instancji na diagramie BPMN.

Aby przesunąć instancję do przodu (np. Strony 7, 21, 31, 41), zdefiniuj elementy strony, tak jak pokazano na rysunku 17.36.



Rysunek 17.36: Elementy strony “Seting” .

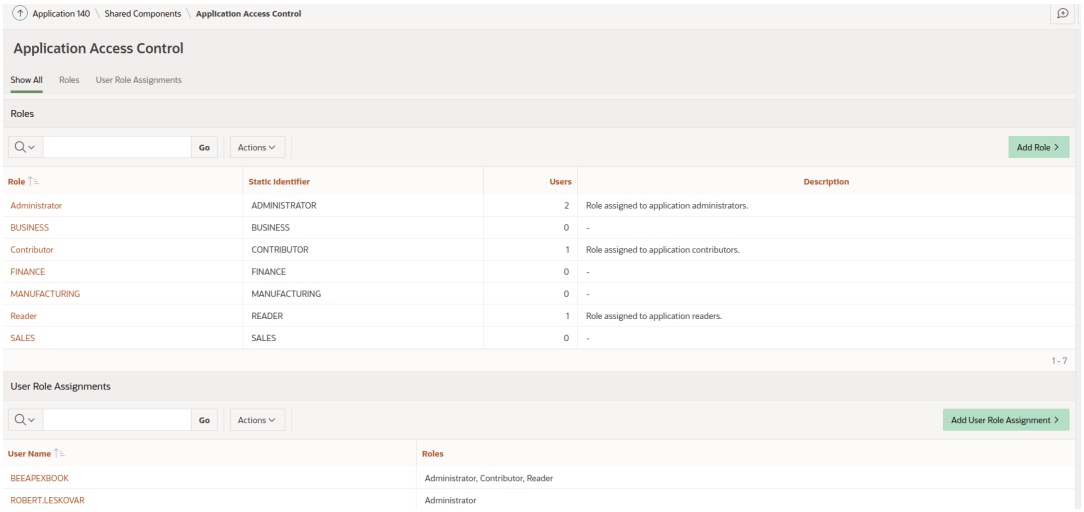
Tylko pierwsze cztery elementy strony (od P7_PRCS_ID do P7_INSTANCE_NAME) są istotne dla integracji Flows for APEX. Utwórz nowy proces, który jest uruchamiany przez przycisk. Proces jest zdefiniowany jako wtyczka "Manage Flow Instance Step"(Zarządzanie etapem instancji przepływu). Ustaw “Action” (akcję), “Process ID Item”(element ID procesu),”Subflow ID”(element ID podprzepływu) i Śubflow ID” (klucz etapu) zgodnie z Rysunkiem 17.37.



Rysunek 17.37: Zakończenie etapu w Flows for Apex.

17.9 Definiowanie ról użytkowników

Role użytkowników można zdefiniować w “Access Control”(Kontroli dostępu) do aplikacji w “Shared Components” komponentach współdzielonych (np. Rysunek 17.38).



Rysunek 17.38: Zdefiniuj role i role użytkowników w menu “Application Access Control”.

Na ten moment pominiemy szczegóły i implementację w tym konkretnym przypadku. Dla szybkiego przeglądu, zajrzyj do Rozdziału 13, a dla dokładniejszych informacji, zgłęb dokumentację APEX dotyczącą autoryzacji.

17.10 Testowanie i korygowanie błędów

Dla początkujących, proces testowania będzie bardzo uproszczony. Kiedy zaczniesz używać klawiatury i myszki do pracy z APEX, będziesz pewnie popełniać błędy i potem je poprawiać. Nie bój się popełniać błędów. Wynieś naukę z każdego, który rozwiązałeś samodzielnie. Dziel się swoją wiedzą w otwartej społeczności APEX. Dyscyplina testowania oprogramowania jest ogromna i możesz ją nauczyć się, działając. Nie ma innej drogi. Programiści są pierwszą linią obrony przed błędami, usterkami i awariami oprogramowania. Niezwykle, większość nieudanych projektów rozwoju oprogramowania szła źle już długo przed napisaniem choćby jednej linii kodu. Aplikacja rozwijana w tym rozdziale ma również błędy oprogramowania. Część z nich zostanie odkryta i zdebugowana przez bystrych czytelników. Niektóre błędy są również ukryte głęboko w samym APEX. My, autorzy i deweloperzy tego rozdziału, mamy nadzieję, że testowanie i debugowanie

staną się przyjemnym wyzwaniem dla czytelników. Usuwanie błędów z oprogramowania bywa czasem zabawą, a czasem przekleństwem. Po prostu nie poddawaj się zbyt szybko!

17.11 Dodatkowy materiał do nauki

Możesz znaleźć następujące dodatkowe materiały do nauki:

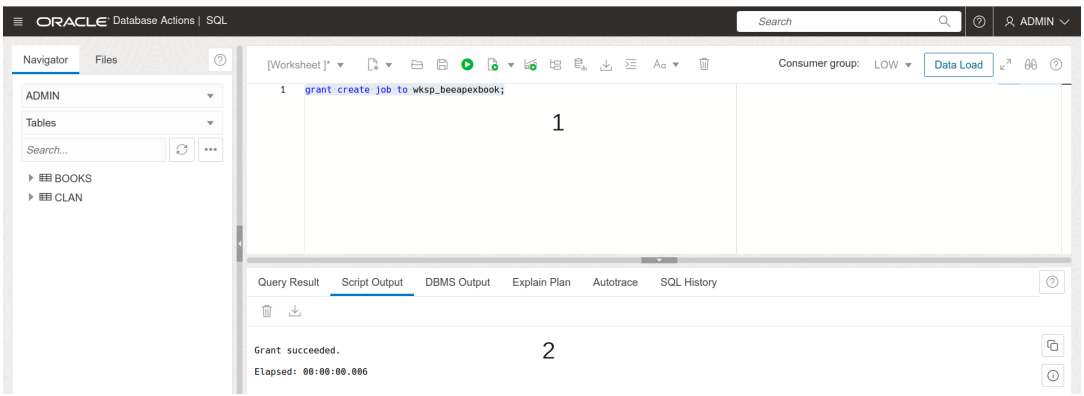
- wyeksportowaną aplikację w postaci pakietu oraz skrypty
- model przepływu pracy o nazwie CH17
- materiały wideo i przewodniki

Wszystkie dodatkowe materiały są dostępne na publicznej stronie BeeAPEX . Zaloguj się jako użytkownik gość (nie jest wymagane hasło). Znajdź podręcznik w sekcji “Books”(Książki), skrypty w folderze Part 2 > Chapter17 w sekcji “Scripts” (Skrypty) i przewodniki wideo w “Collection of video guides”. Materiał na krótkie kursy w “Short courses section”.

17.11.1 Eksportowana aplikacja

Eksportowana aplikacja jest spakowana. Instalacja tworzy tabele, sekwencje, funkcje i wypełnia danymi. Dezinstalacja usuwa wszystkie obiekty bazy danych używane w tej aplikacji. Spakowana aplikacja jest przetestowana i będzie działać w nowej przestrzeni roboczej, jeśli zostaną spełnione następujące wymagania:

- uprawnienie ćreate job"jest przyznane schematowi, który hostuje nową przestrzeń roboczą. Administrator bazy danych może przyznać to uprawnienie za pomocą polecenia "grant create job to <schema>". Przestrzeń robocza APEX jest powiązana z obiektem bazy danych o nazwie schemat. Jeśli korzystasz z OCI (darmowego, płatnego lub dostarczanego przez Oracle Academy), użyj interfejsu webowego OCI (Autonomous Database > specific instance > Database action) lub przejdź bezpośrednio do interfejsu webowego administratora bazy danych. Możesz wykonywać zapytania i skrypty. Rysunek 17.39 przedstawia wygląd interfejsu z wprowadzonym poleceniem (1) i odpowiedzią bazy danych (2).

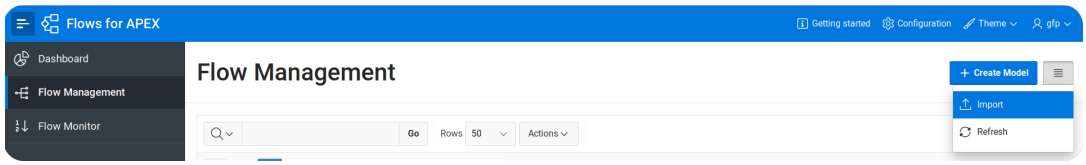


Rysunek 17.39: Przyznawanie "uprawnień do tworzenia zadań"do obszaru roboczego.

Jeśli używasz innych narzędzi (SQL Developer, TOAD), procedura jest prawie taka sama, z wyjątkiem wyglądu interfejsu.

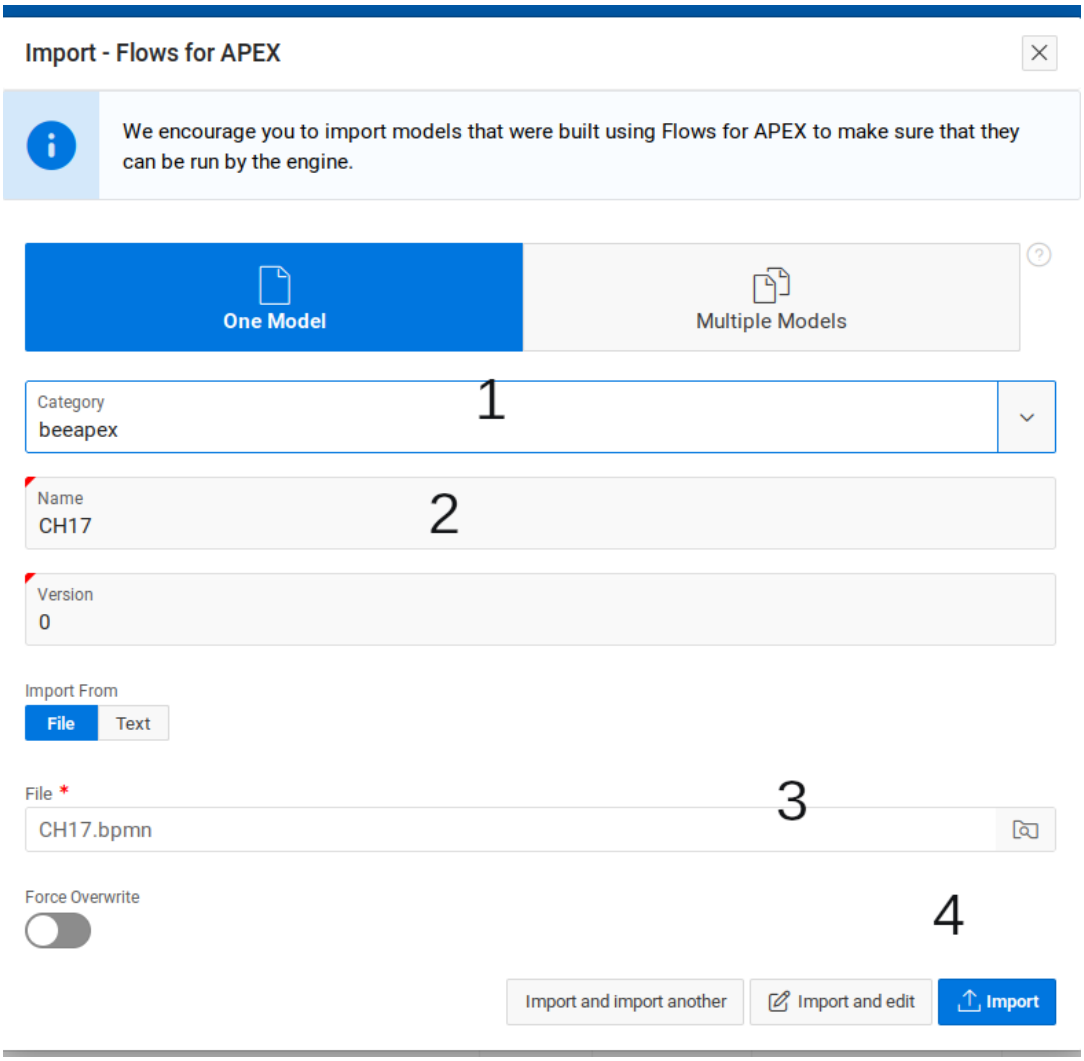
- Flows for APEX musi być zainstalowane w nowej przestrzeni roboczej (patrz wcześniejsze instrukcje).
- Model przepływu pracy o nazwie CH17 jest importowany do Flows for APEX. Plik *CH17.bpmn* jest przechowywany w katalogu Flows4APEX w spakowanej aplikacji jako plik statyczny. Przejdź do Shared Components > Static Application Files i pobierz go na swój komputer lokalny. Ten sam plik jest także dostępny w skryptach materiałów edukacyjnych. Następnie zaimportuj plik *CH17.bpmn* za pomocą Flows for APEX. Zaloguj się do aplikacji Flows

for APEX, wybierz “Flow management” (Zarządzanie przepływem) i zaimportuj plik (patrz Rysunek 17.40 - krok 1 i Rysunek 17.41 - krok 2).



Rysunek 17.40: Importowanie przepływu pracy o nazwie CH17 do Flows for APEX - krok 1.

Znaczenie numerów na Rysunku 17.41 to: 1 (dowolna kategoria), 2 (nazwa modelu musi być dokładnie ĆH17"), 3 (przejdź do eksploratora plików i wybierz plik CH17.bpmn) oraz 4 (potwierdź, naciskając przycisk "Import").



Rysunek 17.41: Importowanie przepływu pracy o nazwie CH17 do Flows for APEX - krok 2.

- Po zainstalowaniu spakowanej aplikacji dodaj przypisanie roli użytkownika (Shared components > Application Access Control > Add User Role Assignment)).

Jeśli którykolwiek z powyższych wymagań nie jest spełniony, to zaimportowana aplikacja może ulec awarii. Konieczne jest wtedy wyczyszczenie ciasteczka w przeglądarce internetowej (np.

w przeglądarce Firefox: Settings > Cookies and Site Data > Manage Data) po awarii aplikacji spowodowanej niespełnieniem wymagań.

17.11.2 Przewodniki wideo

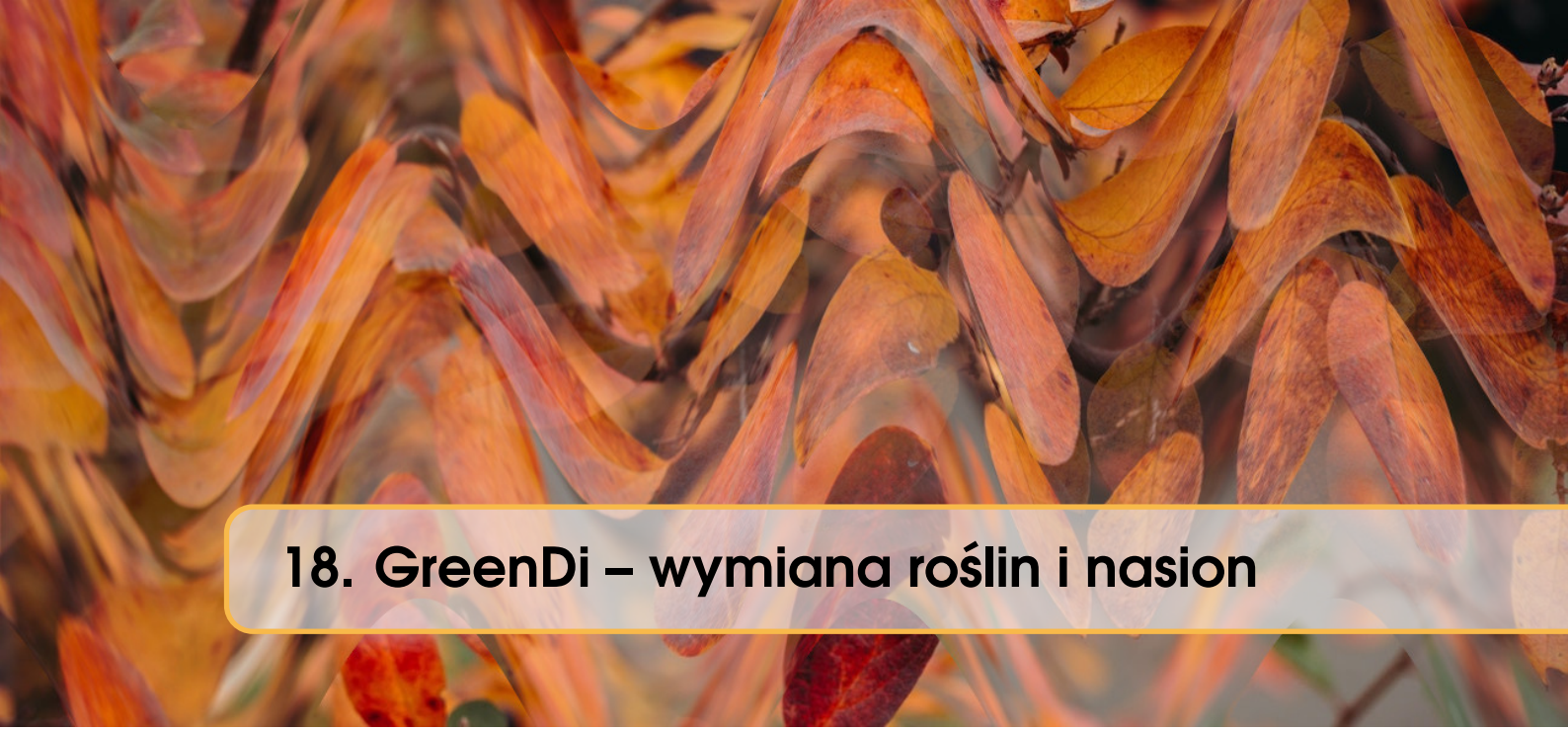
Przewodnik wideo pokazuje wstępne kroki w tworzeniu aplikacji.

17.12 Pytania

1. Wyjaśnij, co to jest Flows for APEX?
2. Jakie artefakty modelu procesów biznesowych przedstawiono za pomocą diagramu BPMN?
3. Jakie funkcje Flows for APEX zostały użyte w tej aplikacji?

17.13 Odpowiedzi

1. Flows for APEX jest nazywane funkcją, wtyczką i aplikacją. Wszystkie trzy znaczenia są poprawne. Mówimy o funkcji, gdy rozmawiamy w kontekście modelowania procesów. Wtyczka odnosi się do tworzenia niestandardowych aplikacji. Aplikacja "Flows for APEX"ma na celu projektowanie przepływów pracy i zarządzanie instancjami w przypadku problemów.
2. Artefakty modelu procesu biznesowego przedstawione na diagramie BPMN to:
   - Start Event: Wskazuje punkt początkowy procesu.
   - Task: Reprezentuje pojedynczą jednostkę pracy, taką jak działanie lub krok w procesie.
   - Gateway: Reprezentuje punkt decyzyjny w procesie, taki jak rozgałęzienie w przepływie lub warunkowa gałąź.
   - Sequence Flow: Łączy elementy na diagramie i reprezentuje przepływ kontroli między nimi.
   - End Event: Wskazuje zakończenie procesu.
   - Pool/Lane: Reprezentuje grupowanie powiązanych zadań, używane do określenia ról i odpowiedzialności różnych uczestników w procesie.
   - Message Flow: Reprezentuje przepływ wiadomości między uczestnikami w procesie, taką jak komunikacja między dwoma systemami.
   - Data Object: Reprezentuje kawałek danych używanych lub produkowanych w procesie, takie jak faktura lub rekord klienta.
   - Data Store: Reprezentuje kontener do przechowywania danych używanych lub produkowanych w procesie, takie jak baza danych lub system plików.
3. Aby uruchomić instancję, użyto akcji Ćreate and Start"(Utwórz i Uruchom). Aby przedstawić instancję na diagramie BPMN, użyto kreatora strony wtyczki z komponentem przeglądarki. Aby przejść do następnego kroku, użyto komponentu manage Flow Instance Step"(Zarządzaj krokiem instancji przepływu) w aplikacji.



18. GreenDi – wymiana roślin i nasion

VJERAN STRAHONJA, DIJANA OREŠKI, DARKO ANDROČEC, ANA KUTNJAK

18.1 Biznesowy pogląd na sprawę

Krótka prezentacja biznesowa platformy GreenDi jest opisana w rozdziałach 14 i 15. Rozdział 18 odnosi się do części platformy poświęconej wymianie roślin i nasion. Wszyscy zarejestrowani użytkownicy platformy mogą uczestniczyć w tej wymianie, prezentując swoją własną ofertę lub odpowiadając na oferty innych osób. Jednocześnie system powinien umożliwiać wyszukiwanie ofert za pomocą różnych filtrów, sortowania kolumn i podobnych funkcji. Dodatkowo, powinna być możliwość komunikacji między oferentem a innymi użytkownikami związanymi z określoną ofertą. Komunikacja odbywa się w formie czatu.

18.2 Definicja problemu

Poniższe zdania opisują problem:

- Podstawowe funkcje wymiany nasion i roślin to składanie własnych ofert (kupna i sprzedaży) oraz odpowiadanie na oferty innych użytkowników (kupna i sprzedaży), co jest zarezerwowane dla członków.
- Członkowie mogą przeszukiwać katalog według różnych kryteriów i otwierać tabelę z otwartymi ofertami dla każdego rodzaju roślin z katalogu.
- Platforma nie jest komercyjna, tzn. rośliny nie są sprzedawane ani kupowane na platformie, ale wymieniane i ofiarowywane. W związku z tym transakcje zakładowe nie podlegają opodatkowaniu.
- Oferta zawiera informacje o rodzaju rośliny, która jest oferowana lub poszukiwana, ilości, miejscu, uwagach, dacie oferty, dacie ważności oferty, instrukcjach dotyczących odbioru lub wysyłki, statusie itp.
- Oferty (kupna lub sprzedaży) prezentowane są w formie tabeli, z możliwością sortowania według kolumn oraz wyszukiwania i filtrowania według różnych kryteriów (rodzaj zakładu, daty oferty, członeka, miejsca, statusu itp.)
- Każdy członek może odpowiedzieć na dowolną otwartą ofertę.
- Rozmowy dotyczące oferty przyjmują formę publicznego lub prywatnego czatu.
- Podstawową jednostką rozmowy (czatu) jest wiadomość.
- Wiadomość zawiera informacje o użytkowniku, który ją wysyła (automatycznie generowane), treść wiadomości i widoczność (publiczna, prywatna).

- System powinien wspierać wszystkie etapy zarządzania cyklem życia oferty (tworzenie, modyfikowanie, zamykanie, usuwanie i archiwizacja). Ofertę tworzy, modyfikuje, zamyka, usuwa i archiwizuje właściciel oferty. Administrator może usunąć ofertę, jeśli narusza pewne zasady, oraz może archiwizować nieaktywne oferty po określonym czasie.

18.3 Przypadki użycia

18.3.1 Opis przypadku użycia

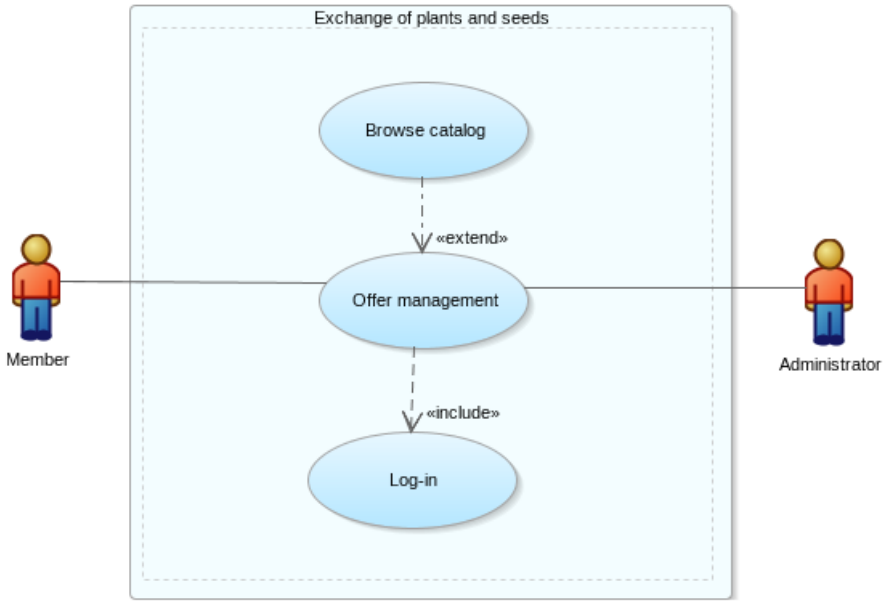
Komunikacja z zarządzaniem ofertami (UC Offer management) jest dostępna dla każdego zarejestrowanego użytkownika lub członka platformy GreenDi. Warunkiem koniecznym do komunikacji jest udane zalogowanie się (w ramach UC, tj. interfejsu użytkownika). Funkcjonalność przeglądania katalogu roślin (UC Browse catalog of plants) stanowi rozszerzenie zarządzania ofertami (UC Offer management). Administrator jest również zewnętrznym użytkownikiem zarządzania ofertami (UC Offer management). Jednocześnie może on przeglądać wszystkie oferty i rozmowy bez ograniczeń oraz zmieniać status każdej oferty.

18.3.2 Opis częściowo ustrukturyzowany

Tabela 18.1 przedstawia UC.

18.3.3 Diagram przypadków użycia

Powyższa historia jest przedstawiona na schemacie przypadków użycia 18.1.



Rysunek 18.1: Diagram przypadków użycia - Wymiana roślin i nasion

18.4 Model danych

W tej sekcji wyjaśniono model danych.

18.4.1 Narrative description of data model

Model danych logicznych (model encji i relacji) jest faktycznie ulepszeniem modelu danych z Rozdziałów 14 i 15). Typy encji Jednostka roślin i tematyczna zostały zaczerpnięte z Rozdziału

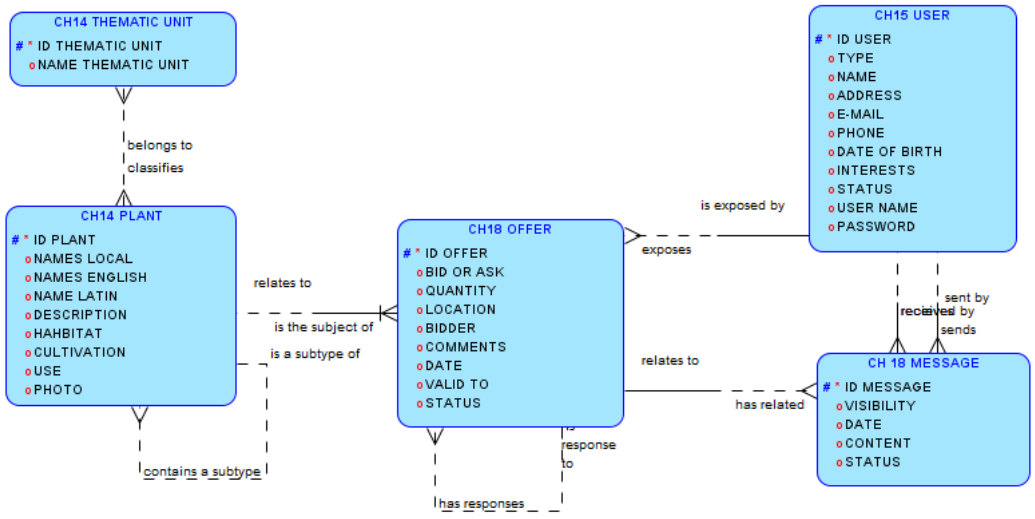
Tablica 18.1: Opis przypadku użycia: Wymiana roślin i nasion.

Słowo kluczowe	**Wartość**
ID:	*ch18-01*
Tytuł:	*Wymiana roślin i nasion*
Opis:	*Wymiana roślin.*
Główny aktor:	*Każda osoba*
Warunki wstępne:	*Podstawowe informacje o zakładzie są otwarte do wyszukiwania i przeglądania bez rejestracji.*
Warunki końcowe:	*Jeśli użytkownik wprowadzi nową ofertę, zapisywane są nowe dane.*
Główny	-
Scenariusz sukcesu:	1. Użytkownik przewija katalog w górę i w dół, aż znajdzie roślinę lub stosuje filtry wyszukiwania. 2. Użytkownik otwiera listę ofert dla wybranej rośliny lub wszystkich roślin. 3. Użytkownik przewija listę ofert w górę i w dół lub stosuje filtr wyszukiwania. 4. Użytkownik tworzy nową ofertę dla wybranej rośliny, wprowadza szczegóły i udostępnia nową ofertę. 5. Użytkownik klika na wybraną ofertę, otwiera okno pop-up z danymi wybranej oferty. 6. Użytkownik może odpowiedzieć na dowolną aktywną ofertę kontrpropozycją lub akceptacją, lub rozpoczynając czat. 7. Rozmowa odbywa się w formie wiadomości i odpowiedzi na wiadomości. 8. Link jest widoczny dla oferenta i innych uczestniczących użytkowników.
Rozszerzenia:	• *Lista ofert (raport)* • *Szczegóły oferty (nowe lub istniejące)* • *Czat (lista wiadomości)*
Częstotliwość stosowania:	*Maksymalnie 100 dziennie.*
Status:	*Status rozwoju*
Właściciel:	*publiczny, anonimowy użytkownik*
Priorytet:	*wysoki*

14 , a Użytkownik z Rozdziału 15 . Podstawowe typy encji dla Wymiany Roślin i Nasion to Oferta i Wiadomość. Oferta jest opisana przez następujące atrybuty: ID offer (atrybut identyfikacyjny); “Bid or ask” (Licytacja lub pytanie - określa, czy roślina lub nasienie jest oferowane lub żądane), “Quantity”(Ilość - rośliny lub nasion), “Location” (Lokalizacja- gdzie można odebrać lub dostarczyć roślinę lub nasienie), “Bidder” (Licytujący - użytkownik, który licytuje lub pyta), “Comments”(Komentarze dotyczące jakości, instrukcje dotyczące dostawy itp.), “Date”(data oferty), “Valid to” (data ważności), “Status” (aktywny, wycofany, zrealizowany). Oferta jest oryginalną ofertą lub odpowiedzią na oryginalną ofertę. Oznacza to, że każda oferta jest odpowiedzią na 0 lub jedną ofertę i ma 0, 1 lub więcej odpowiedzi. Każda Oferta jest oferowana przez jednego i tylko jednego Użytkownika. 0, 1 lub więcej Wiadomości jest związanych z Ofertą. Każda Wiadomość jest powiązana z 1, i tylko 1 ofertą, wysłaną przez 1, i tylko 1 Użytkownika (nadawcę) i przeznaczoną dla 1, i tylko 1 użytkownika (odbiorcy).

18.4.2 Logiczny model danych

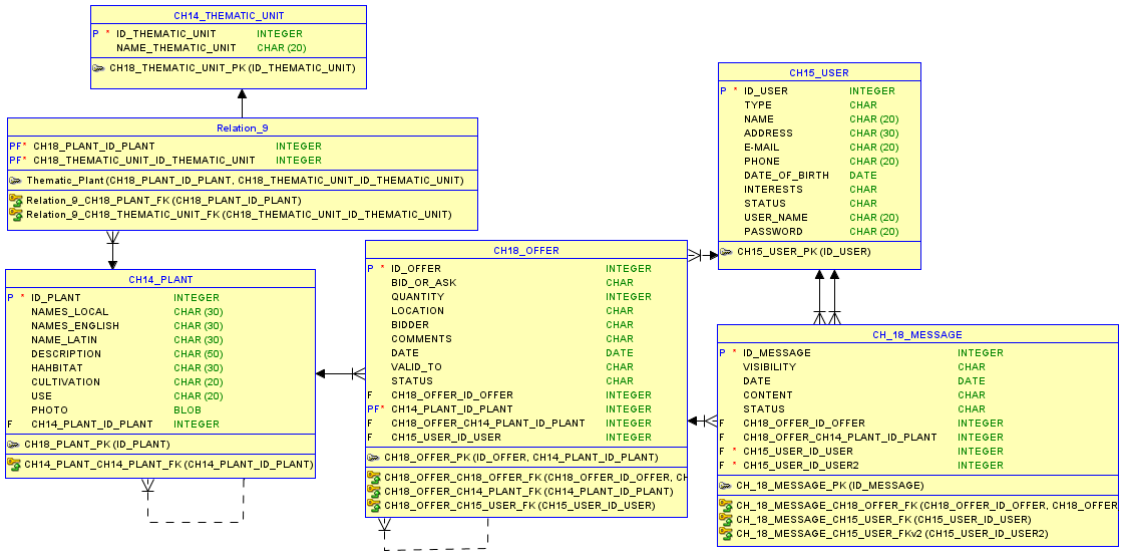
Logiczny model danych przedstawiono na rysunku 18.2.



Rysunek 18.2: Logiczny model danych.

18.4.3 Relacyjny model danych

Automatyczna transformacja z modelu danych logicznych do modelu danych relacyjnych w Oracle SQL Data Modeler jest dostarczana przez funkcję *Engineering to relational* . Wynik jest przedstawiony na Rysunku 18.3.

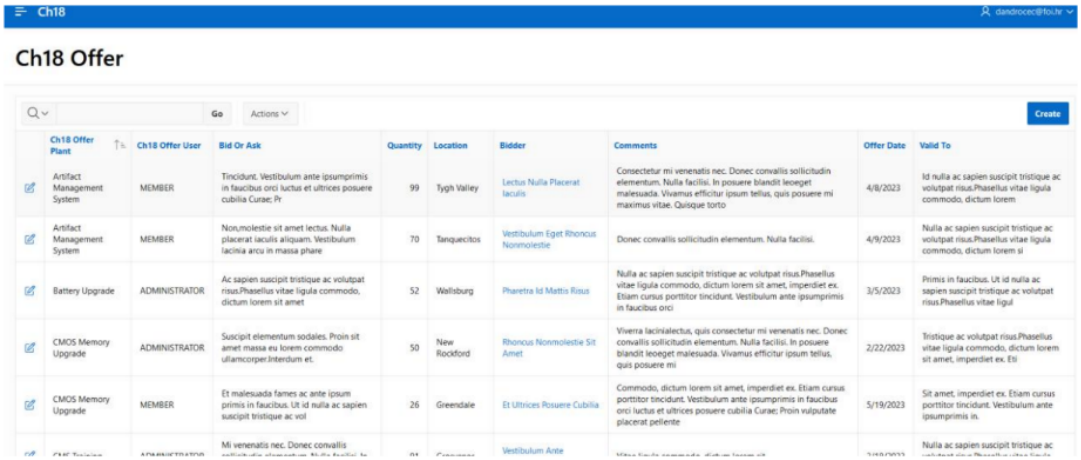


Rysunek 18.3: Relacyjny model danych.

Teraz pozwól Oracle SQL Data Modeler wygenerować skrypt SQL. Wybierz wszystkie tabele w modelu relacyjnym i użyj funkcji File > Export > DDL File, aby uzyskać skrypt. Możemy zaimportować wygenerowany skrypt do APEX i go wykonać.

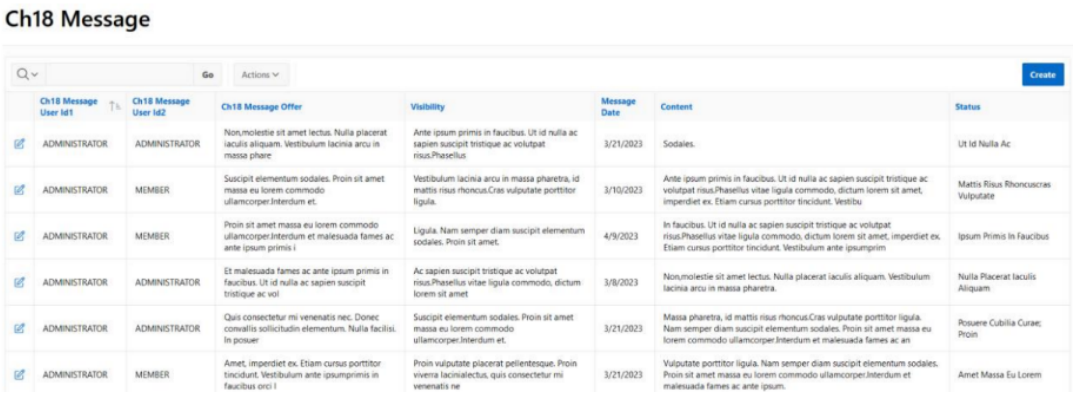
18.5 Interfejsy aplikacji

Oferty są prezentowane w formie interaktywnego raportu (zob.rysunek 18.4).



Rysunek 18.4: Oferty - raport interaktywny.

Formularz wiadomości przedstawiono na rysunku 18.5.



Rysunek 18.5: Formularz wiadomości.

18.6 Dodatkowe materiały do nauki

Możesz znaleźć następujące dodatkowe materiały do nauki:

- wyeksportowana aplikacja
- skrypty do tworzenia, upuszczania i wstawiania
- przewodnik wideo

Wszystkie dodatkowe materiały są dostępne na publicznej stronie BeeAPEX . Zaloguj się jako użytkownik gość (nie jest wymagane hasło). Znajdź podręcznik w sekcji “Books”(Książki), skrypty w folderze Part 2 > Chapter18 w sekcji “Scripts” (Skrypty) i przewodniki wideo w “Collection of video guides”. Materiał na krótkie kursy w “Short courses section”.

18.6.1 Eksportowana aplikacja

Wyeksportowana aplikacja jest spakowana. Podczas instalacji tworzone są tabele i wypełniane danymi. Odinstalowanie usuwa wszystkie obiekty bazy danych używane w tej aplikacji.

18.6.2 Przewodniki wideo

Przewodnik wideo pokazuje każdy krok w tworzeniu aplikacji.

18.7 Pytania

1. Jak możemy sprawić, by oferta była widoczna z opóźnieniem, od pewnego momentu, kiedy oferent tego chce?
2. Czy ta aplikacja może być używana do czatowania na inny temat niż faktyczna wymiana roślin i nasion? Jak?
3. W tabeli CH18_OFFER kolumna VALID_TO ma typ danych CHAR. Jakie są potencjalne problemy, jeśli data jest zdefiniowana jako typ CHAR?

18.8 Odpowiedzi

1. Po pierwsze, dodalibyśmy kolumnę w tabeli CH18_OFFER. Następnie poprawilibyśmy formularz umożliwiający wprowadzanie nowych ofert. Ostatecznie poprawilibyśmy zapytanie w raporcie ofert poprzez dodanie warunku (where new_column <= SYSDATE).
2. Teoretycznie jest to możliwe. Moderator czatu powinien otworzyć fikcyjną ofertę, a wszyscy użytkownicy, którzy pojawią się ze swoją fikcyjną kontrofertą, mogliby wziąć udział w czacie.
3. Przechowywanie dat jako CHAR może wprowadzić problemy z integralnością danych, problematykę sortowania, ograniczone operacje na dacie, trudności formatowania oraz inne

problemy. Zaleca się ogólnie stosowanie odpowiednich typów danych dotyczących dat dostarczanych przez Oracle, takich jak DATE lub TIMESTAMP, dla dokładnego i wydajnego przechowywania i manipulacji danymi datowymi.



19. System zarządzania recenzjami książek

A. KUTNJAK, L. HRUSTEK, A. BAGGIA AND R.LESKOVAR

19.1 Biznesowy pogląd na sprawę

System zarządzania recenzjami książek został zaprojektowany w celu zarządzania recenzjami książek dostępnych do sprzedaży na dowolnej platformie internetowej. Firma (np. Amazon) rozważa zarządzanie katalogiem dostępnych książek w taki sposób, że na podstawie recenzji użytkowników podejmuje decyzję o dodaniu lub usunięciu książki z oferty. Korzystając z systemu zarządzania recenzjami, firma zbiera opinie użytkowników, tworzy unikalną bazę danych i podejmuje dalsze działania na podstawie zebranych danych. Podstawowymi interesariuszami bazy danych są administrator i użytkownik czytający książki. Przypadek użycia i pomysł są inspirowane i oferowane przez Kaggle - platformę społecznościową dla naukowców danych i entuzjastów uczenia maszynowego. Oryginalne dane są dostępne na stronie www.kaggle.com. Przypadki użycia oraz model danych są gruntownie zmodyfikowane i uzupełnione. Celem tego przypadku biznesowego jest demonstracja możliwości rozwoju APEX opierającego się wyłącznie na kreatorach. Za wyjątkiem Quick SQL (łącznie 31 krótkich linii) żaden inny kod nie będzie pisany przez dewelopera.

19.2 Definicja problemu

System zarządzania recenzjami posiada następujące funkcjonalności:

- Z uwagi na szeroki wybór książek w globalnej ofercie, gromadzone są dane dotyczące ich recenzji i ocen.
- Wspólny widok platformy obejmuje listę książek z informacjami o tytule, autorach, opisie treści, wydawcy, dacie publikacji, kategoriach i recenzjach.
- Opcje administratora obejmują dodawanie książek do bazy danych platformy, a także ich edycję i wybór według kategorii.
- Ponieważ baza danych umożliwia użytkownikom dodawanie recenzji, administrator może przeglądać recenzje i komentarze do recenzji. Na tej podstawie podejmuje decyzje dotyczące działań związanych z książką (dodawanie lub usuwanie).
- Użytkownik ma możliwość zarejestrowania się na platformie, co umożliwia mu przeglądanie szczegółów książki, ale także recenzowanie i komentowanie recenzji.
- Recenzje po stronie użytkownika mają dwie korzyści. Po pierwsze, na podstawie analizy recenzji książki potencjalny użytkownik decyduje się na zakup. Po drugie, na podstawie istniejących recenzji firma podejmuje działania dotyczące książek, tj. ich dodawanie lub

usuwanie z katalogu.

19.3 Przypadki użycia

19.3.1 Opis przypadku użycia

Administrator jest odpowiedzialny za wprowadzanie lub zmianę danych dotyczących książek w ramach systemu zarządzania recenzjami firmy i ma możliwość przeglądania różnych wskaźników statystycznych z nimi związanych. System zarządzania recenzjami Amazon opiera się na rejestracji użytkowników na platformie. Rejestrując się, użytkownicy mogą przeglądać katalog dostępnych książek na platformie. Katalog książek zawiera szczegółowe informacje o dostępnych książkach i stanowi podstawę do podejmowania dalszych działań w ramach platformy. Książki są klasyfikowane na kategorie, co ułatwia wyszukiwanie i pozwala na wprowadzenie osobistych preferencji użytkownika. Zarejestrowany użytkownik może dodawać recenzje i komentować recenzje.

19.3.2 Opis częściowo ustrukturyzowany

Aktywności związane z wprowadzaniem danych dotyczących nowych książek są przypisane do administratora. Uzupełnia on bieżącą bazę danych o nowe książki, co skutkuje dodaniem oferty do katalogu książek. Częstotliwość korzystania z platformy przez administratora jest codzienna. Użytkownik rejestruje się na platformie systemu zarządzania recenzjami książek. Po pomyślnej rejestracji użytkownik przegląda katalog książek, ich recenzje, a ostatecznie dodaje swoją własną recenzję i komentarze do recenzji. Częstotliwość korzystania z platformy przez użytkowników to kilka razy w miesiącu.

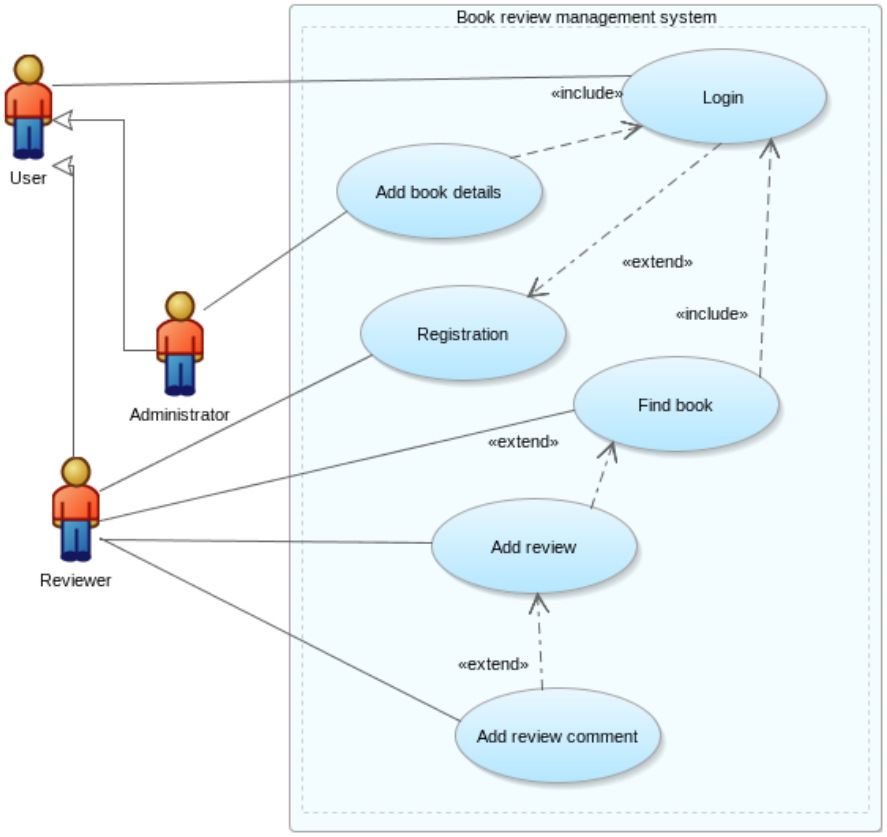
Opis przypadków użycia został przedstawiony w Tabli 19.1.

19.3.3 Diagram przypadków użycia

Powyższa historia jest przedstawiona na schemacie przypadków użycia (zob.rysunek 19.1).

Tablica 19.1: Opis przypadku użycia: system zarządzania recenzjami książek

Słowo kluczowe	**Wartość**
ID:	*ch19-10*
Tytuł:	*Analiza recenzji książek*
Opis:	*System zarządzania recenzjami książek jest dostępny na podstawie rejestracji na platformie. Platforma umożliwia przeglądanie katalogów, filtrowanie ich i klasyfikowanie według kategorii.*
Główny aktor:	*Użytkownik*
Warunki wstępne:	przeglądarka na komputerze lub smartfonie, użytkownik ma poświadczenia, aplikacja jest dostępna.
Warunki końcowe:	Jeśli dodano szczegóły książki, recenzję lub komentarz do recenzji, niż baza danych ją zapisuje.
Główny	Scenariusz
Scenariusz sukcesu:	Pierwszy scenariusz (administrator dodaje nową książkę): 1. Administrator loguje się do aplikacji. 2. Administrator wybiera formularz do dodawania nowej książki. 3. Formularz wprowadzania obejmuje tytuł, autorów, opis, wydawcę, datę publikacji i wybór kategorii. 4. Administrator kończy procedurę tworzenia nowego wpisu. 5. Administrator wylogowuje się z platformy. Drugi scenariusz (użytkownik, recenzja książki): 1. Użytkownik rejestruje się na platformie. 2. Użytkownik przegląda katalog książek i wybiera książkę, którą chce zrecenzować. 3. Formularz recenzji obejmuje tytuł, cenę, nazwę profilu, pomocność recenzji, ocenę recenzji, podsumowanie recenzji i treść recenzji. 4. Użytkownik kończy procedurę dodawania nowej recenzji. 5. Użytkownik wylogowuje się z platformy. Trzeci scenariusz (użytkownik, komentowanie recenzji książki): 1. Użytkownik rejestruje się na platformie. 2. Użytkownik przegląda recenzje książek.. 3. Recenzja obejmuje tytuł, cenę, nazwę profilu, pomocność recenzji, ocenę recenzji, podsumowanie recenzji i treść recenzji. 4. Użytkownik dodaje nowy komentarz do recenzji. 5. Użytkownik wylogowuje się z platformy.
Rozszerzenia:	• *Dodawanie książek* • *Interaktywny raport dla każdej książki* • *Recenzje książek* • *Komentarze do recenzji*
Częstotliwość stosowania:	*Codziennie, kilka razy w miesiącu.*
Status:	[rozwój]
Właściciel:	Potencjalni użytkownicy, na podstawie rejestracji.
Priorytet:	*wysoki*



Rysunek 19.1: Diagramy przypadków użycia.

19.4 Model danych

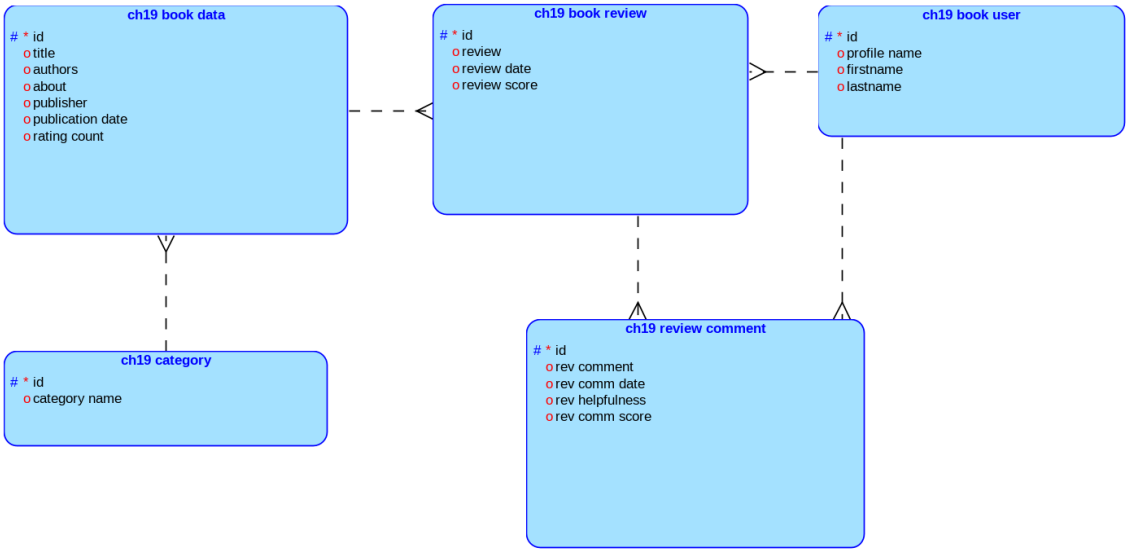
19.4.1 Opis modelu danych

Model danych logicznych (model encji i relacji) składa się z kilku encji: ch19_category, ch19_book_data, ch19_book_review, ch19_book_user i ch19_book_review_comment. Encja book_data składa się z kilku atrybutów: tytuł, autorzy, informacje, wydawca, data publikacji i liczba ocen. Encja book_user zawiera nazwę profilu, imię i nazwisko. Encja book_review ma atrybuty: recenzja, data recenzji i ocena recenzji. Encja review_comment zawiera odpowiedzi na recenzje: komentarz do recenzji, pomocność recenzji i ocena komentarza do recenzji. Każda książka należy do jednej i tylko jednej kategorii. Kategoria może zawierać wiele książek. Może być wiele recenzji związanych z jedną książką. Użytkownik może komentować wiele książek i wiele recenzji innych użytkowników. Dodajemy unikatowy identyfikator (UID) o nazwie ID do każdej encji. Te atrybuty staną się kluczami głównymi w odpowiadających im tabelach. Po wygenerowaniu skryptu SQL tworzącego tabele dodajemy dwa ograniczenia sprawdzające - pozwalamy tylko na wartości numeryczne ocen recenzji i komentarzy do recenzji z zakresu od 1 do 5.

Programista ma kilka możliwości tworzenia tabel. Jednym z nich jest użycie SQL Developer Data Modeler i zaprojektowanie modelu logicznego od podstaw, przekształcenie w relacyjny i wygenerowanie skryptu SQL. Innym jest tworzenie tabel w APEX, eksportowanie skryptów, które tworzą tabele, stosowanie inżynierii wstecznej w SQL Develoepr Data Modeler w celu wygenerowania modelu relacyjnego, a następnie przekształcenie go w model logiczny. Używanie tylko funkcji APEX Quick SQL jest najprostszym podejściem. Po Quick SQL jest writen, można go przekształcić w skrypt SQL za pomocą wbudowanego generatora APEX.

19.4.2 Logiczny model danych

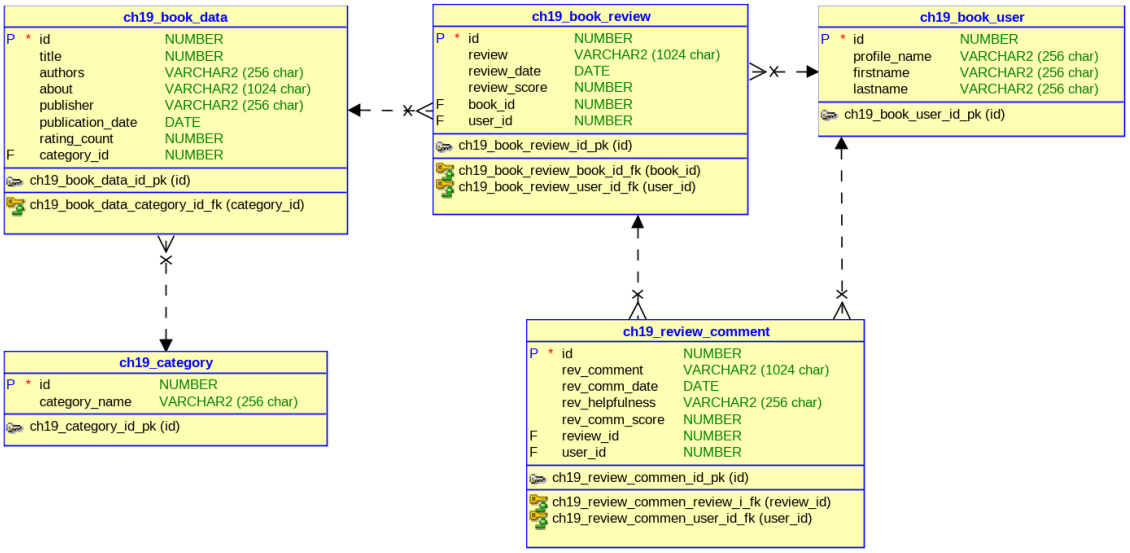
Logiczny model danych przedstawiono na rysunku 19.2.



Rysunek 19.2: Logiczny model danych.

19.4.3 Relacyjny model danych

Automatyczna transformacja z drugiego modelu logicznego na model danych relacyjnych w Oracle SQL Data Modeler jest dostarczana przez funkcję Inżynier do Modelu Relacyjnego. Wynik przedstawiony jest na Rysunku 19.3.



Rysunek 19.3: Relacyjny model danych.

Teraz pozwól Oracle SQL Data Modeler wygenerować skrypt SQL. Wybierz wszystkie tabele w modelu relacyjnym i użyj funkcji File > Export > DDL File, aby uzyskać skrypt. Możemy zaimportować wygenerowany skrypt do APEX i go wykonać.

19.4.4 Skrypt SQL

Skrypt odpowiadający modelowi relacyjnemu zawiera następujące polecenia SQL:

```
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 Szybki SQL

Najprostszą procedurą generowania skryptu SQL jest użycie funkcji Quick SQL w APEX. Poniższe wiersze w Quick SQL umożliwiają generowanie skryptu SQL, przedstawione powyżej:

```
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

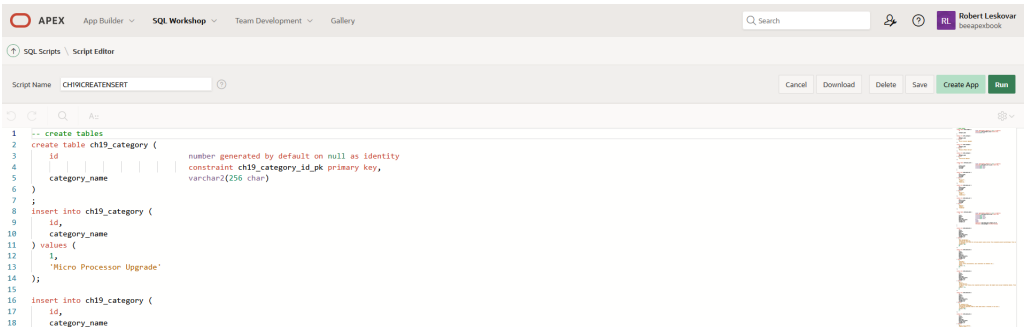
ch19_review_comment
        rev_comment vc1024
        rev_comm_date date
        rev_helpfulness vc256
        rev_comm_score  num /check 1,2,3,4,5
        review_id /fk ch19_book_review
        user_id /fk ch19_book_user
```

Kod jest bardzo gęsty. Wymaga jednak podstawowej wiedzy z zakresu modelowania danych i składni Quick SQL.

19.5 Interfejsy aplikacji

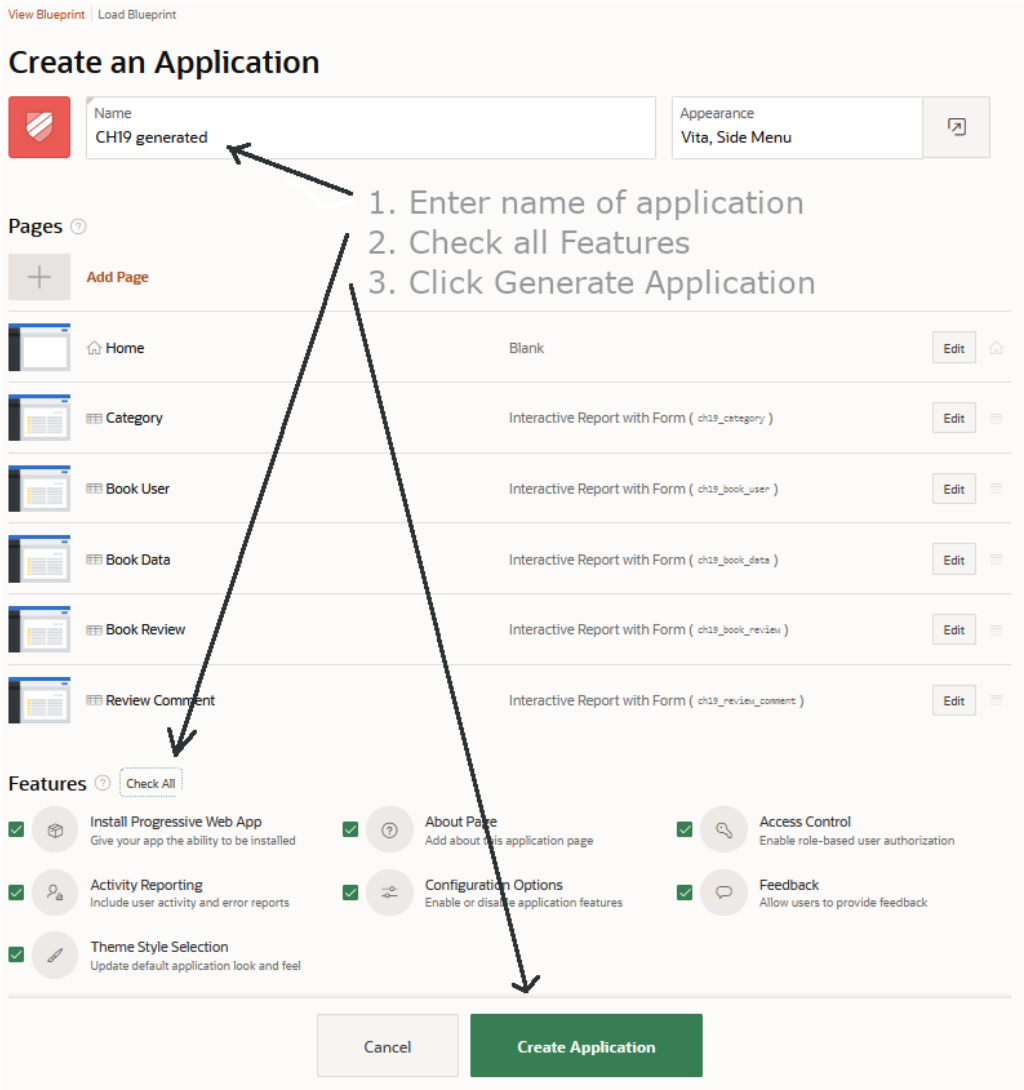
19.5.1 Administrator

W tym przypadku wygenerujemy aplikację ze skryptu SQL, który tworzy tabele i wypełnia przykładowe dane. Znajdź CH19CREATEINSERT.sql plików w materiałach edukacyjnych i zaimportuj je do swojego obszaru roboczego. Po zaimportowaniu otwórz go i kliknij Utwórz aplikację (patrz rysunek 19.4).



Rysunek 19.4: Generowanie aplikacji ze skryptu CH19CREATEINSERT - część 1.

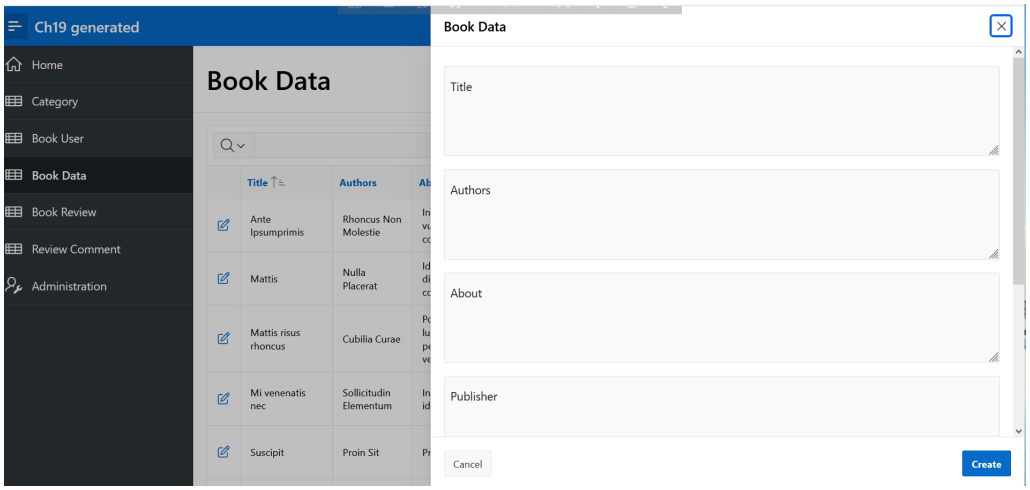
Podaj nazwę aplikacji, wybierz wszystkie funkcje i kliknij “Generate Application” (patrz Rysunek 19.5)(zob.rysunek 19.5).



Rysunek 19.5: Generowanie aplikacji ze skryptu CH19CREATEINSERT - część 2.

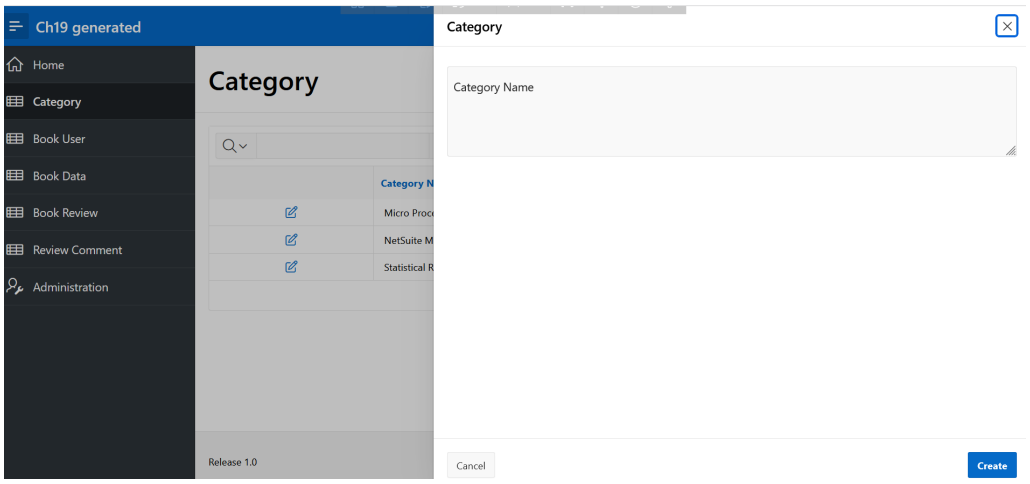
Prototyp działającej aplikacji jest teraz przygotowany. Od teraz będą prezentowane funkcje dla administratora i użytkownika końcowego. Dotychczas żaden kod nie został napisany przez programistę (z wyjątkiem Quick SQL), a interfejs aplikacji nie został dostosowany przez programistę. Cały rzeczywisty kod aplikacji został stworzony wyłącznie przez kreatory APEX.

Rysunek 19.6 pokazuje formularz wprowadzania nowej książki przez administratora.



Rysunek 19.6: Dodawanie książki przez administratora.

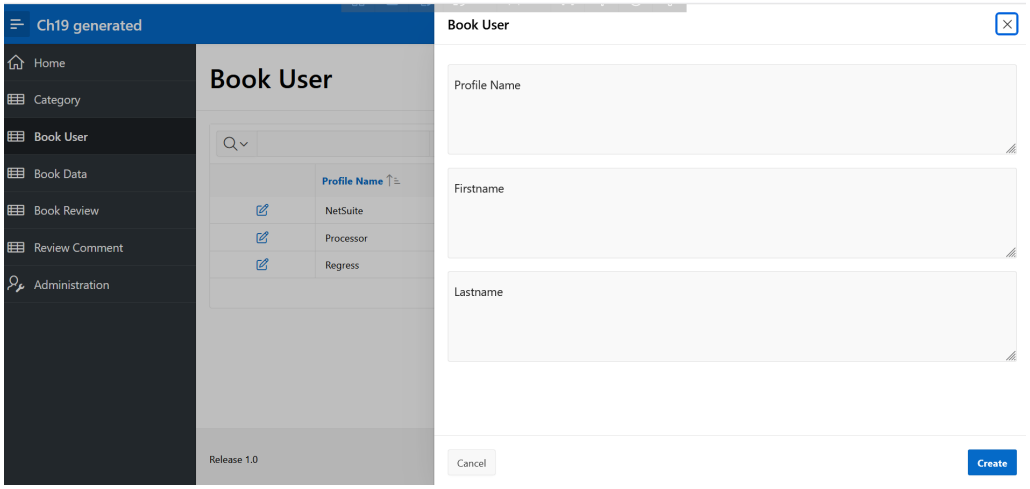
Rysunek 19.7 pokazuje formularz do wstawiania kategorii książki przez administratora.



Rysunek 19.7: Dodawanie kategorii przez administratora.

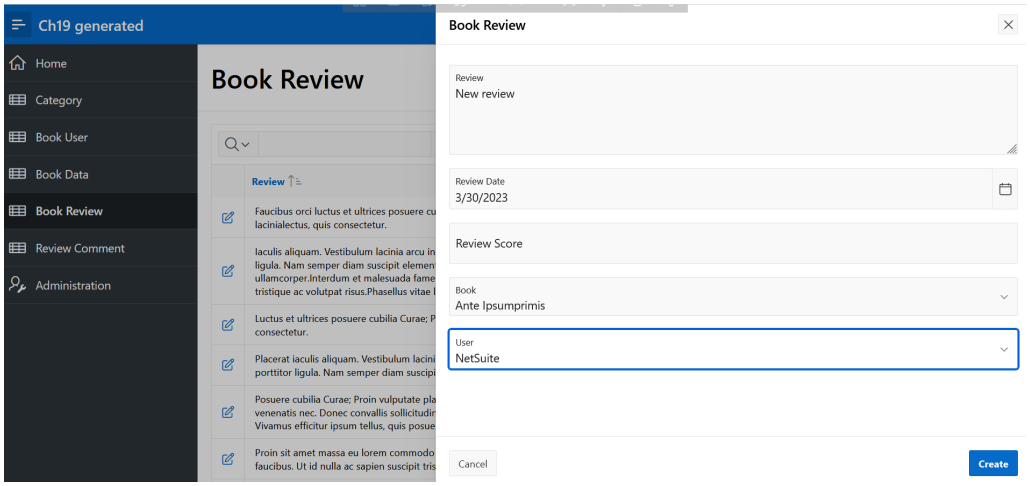
19.5.2 Użytkownik

Rysunek 19.8 przedstawia formularz rejestracyjny użytkownika.



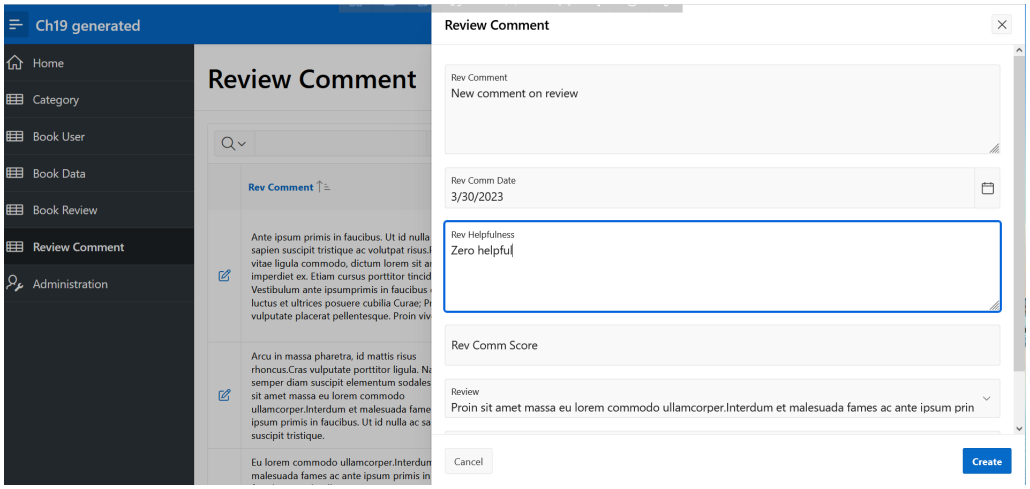
Rysunek 19.8: Rejestracja użytkownika.

Rysunek 19.9 pokazuje przeglądanie i dodawanie recenzji.



Rysunek 19.9: Przeglądanie i dodawanie recenzji.

Rysunek 19.10pokazuje formularz do komentowania recenzji.



Rysunek 19.10: Formularz do komentowania recenzji

Raporty domyślnie są w formie tabeli. Użytkownik końcowy może generować wykresy bez konieczności programowania, korzystając wyłącznie z wbudowanego kreatora. Zobacz Rozdział 2, aby dowiedzieć się, jak przekształcić raporty w wykresy.

19.6 Definiowanie ról użytkowników

Role użytkowników można zdefiniować w Kontroli dostępu do aplikacji w Składnikach wspólnych. W tym momencie pomińmy szczegóły i implementację dla tego konkretnego przypadku. Aby uzyskać szybki przegląd, zobacz Rozdział 13, a dla dogłębnej wiedzy zapoznaj się z dokumentacją APEX dotyczącą autoryzacji.

19.7 Dodatkowe materiały do nauki

Możesz znaleźć następujące dodatkowe materiały do nauki:

- skrypt do tworzenia i wypełniania tabel
- skrypt do usuwania tabel
- wyeksportowaną spakowaną aplikację
- wideo, które demonstruje, jak wygenerować aplikację zgodnie ze skryptem

Wszystkie dodatkowe materiały są dostępne na publicznej stronie BeeAPEX . Zaloguj się jako użytkownik gość (nie jest wymagane hasło). Znajdź podręcznik w sekcji “Books”(Książki), skrypty w folderze Part 2 > Chapter19 w sekcji “Scripts” (Skrypty) i przewodniki wideo w “Collection of video guides”. Materiał na krótkie kursy w “Short courses section”.

19.7.1 Eksportowana aplikacja

Wyeksportowana aplikacja jest spakowana. Instalacja tworzy tabele oraz wypełnia je danymi. Dezinstalacja usuwa wszystkie obiekty bazy danych używane w tej aplikacji. Spakowana aplikacja została przetestowana i będzie działać w nowej przestrzeni roboczej, jeśli spełnione są następujące wymagania:

Spakowana aplikacja została przetestowana i będzie działać w nowej przestrzeni roboczej, jeśli spełnione są następujące wymagania:

- Dodaj użytkownika APEX przed uruchomieniem aplikacji. Tylko w przestrzeni roboczej do rozwoju i testów przejdź do Shared Components > Application Access Control > Add User Role Assignment; wprowadź użytkownika APEX i ustaw dla tego użytkownika role Administrator, Contributer i Reader. W przypadku produkcji przed wdrożeniem należy skonsultować się z wykwalifikowanym personelem.

Jeśli użytkownikowi nie została przyznana odpowiednia rola, zaimportowana aplikacja ulegnie awarii. Konieczne jest wyczyszczenie pliku cookie w przeglądarce internetowej (np. Firefox: Settings > Cookies and Site Data > Manage Data) po awarii aplikacji spowodowanej niespełnieniem wymagań.

19.7.2 Przewodniki wideo

Przewodnik wideo pokazuje wszystkie kroki tworzenia aplikacji.

19.8 Pytania

1. Dlaczego w tym przypadku użyto skryptu Quick SQL?
2. Jak wygenerować wiele formularzy i raportów jednocześnie?
3. Jak autoryzować użytkownika aplikacji APEX wewnątrz samej aplikacji?

19.9 Odpowiedzi

1. Skrypt Quick SQL został wprowadzony w celu przyspieszenia opracowywania modelu danych. Skrypt Quick SQL generuje skrypt SQL na żądanie. Skrypt SQL jest wykonywalny, czyli tworzy obiekty bazy danych, takie jak tabele i widoki.
2. Możliwe jest wygenerowanie wielu formularzy i raportów jednocześnie poprzez napisanie skryptu, który zawiera definicję kilku tabel.
3. Aby autoryzować użytkownika aplikacji APEX wewnątrz samej aplikacji, aplikacja musi zostać wygenerowana z funkcją “Control Feature” (Kontroli dostępu). Następnie uprawniony użytkownik (administrator) może dodać innych użytkowników z rolami Administrator, Contributor i Reader.



20. Zestawienie materiałów i kalkulacja kosztów

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA

20.1 Biznesowy pogląd na sprawę

Obserwowana mała firma, którą nazwiemy **OSC** (10-20 pracowników), produkuje zespoły kablowe, wiązki kabli, przewody i lampy sygnalizacyjne. Oferują również montaż części elektromechanicznych i półproduktów mechanicznych. Ich klientami są większe firmy, które dostarczają sprzęt AGD, sprzęt pomiarowy, systemy kontroli, liczniki elektryczne, urządzenia medyczne i podobne. Zamówienia klientów są przyjmowane telefonicznie i drogą mailową. Dla prostych produktów i operacji, a także dla powtarzalnych zamówień, **OSC** wysyła potwierdzenie do klienta bardzo szybko. Ostatnio wzrosła liczba zamówień na wielopoziomowe produkty złożone oraz bardziej skomplikowane półprodukty. Ponadto w jednym zamówieniu może być dziesiątki pozycji, a każda pozycja może zawierać produkt składający się z kilku półproduktów i materiałów. Półprodukty mogą składać się z półproduktów i materiałów. Klienci chcą otrzymać ofertę szybko. Obliczenie ceny materiału i półproduktów (kalkulacja struktury wyrobu, w skrócie BOM) jest wymagające, podatne na błędy i czasochłonne, zwłaszcza dla wielopoziomowych produktów złożonych i półproduktów. Może minąć kilka dni, zanim odpowie się klientowi ofertą. Obliczenia kosztów pracy (operacji) są innym obliczeniem, które zwięzłość rozdziału pomija. Zasięg rozdziału obejmie więc wyłącznie kalkulację BOM. Właściciel firmy wyraził następujące potrzeby:

- raport dotyczący produktów, półproduktów, materiałów oraz ewentualnych zapytań klientów
- raport dotyczący struktury wyrobu (BOM)
- formularz do edycji produktów, półproduktów, materiałów oraz ewentualnych zapytań klientów
- formularz do edycji struktury wyrobu (BOM)
- kalkulacja i prezentacja struktury wyrobu (BOM) w formie drzewa

Aplikacja powinna być użyteczna na telefonie, tablecie oraz komputerze stacjonarnym.

20.2 Definicja problemu

Manager firmy staje w obliczu kilku problemów związanych z dynamiczną reakcją:

- Obliczenia są czasochłonne, podatne na błędy i wymagające, jednak muszą być przygotowane bardzo szybko.
- Opóźnienia i błędy w obliczeniach mogą wpływać na biznes w różny sposób, takie jak utrata potencjalnych klientów, niższe wyniki finansowe, pogorszenie reputacji firmy, spadek

Tablica 20.1: Opis przypadku użycia: raportowanie i przechowywanie podstawowych danych.

Słowo kluczowe	**Wartość**
ID:	*Ch20-01*
Tytuł:	*Raportowanie i obsługa podstawowych danych*
Opis:	*Manager korzysta z aplikacji APEX do raportowania i utrzymania (wyboru, aktualizacji) podstawowych danych, czy to będzie produkt, półprodukt, materiał czy zapytanie. Materiały i zbiory są opisywane za pomocą tego samego zestawu atrybutów. Aktualizacja cen materiałów nie powoduje ponownego obliczenia cen składników, w których używany jest materiał. Zapytanie klienta może składać się z kilku produktów i półproduktów. Jednostką miary w tym przypadku będzie sztuka.*
Główny aktor:	*Menedżer*
Warunki wstępne:	*Menedżer ma konto dla aplikacji APEX.*
Warunki końcowe:	*Po zgłoszeniu lub utrzymaniu danych można wykonać inne działanie, w tym obliczenie zestawienia komponentów.*
Główny	*Scenariusz*
Scenariusz sukcesu:	1. *Otwórz przeglądarkę internetową i zaloguj się do aplikacji.* 2. *Wybierz element menu lub nawigację strony "Prod./Semi/Material".* 3. *Aby edytować, przewiń stronę w dół lub przefiltruj dane, aby znaleźć określony produkt, półprodukt lub materiał.* 4. *Kliknij ikonę ołówka dla konkretnego produktu, półproduktu lub materiału i zaktualizuj dane.* 5. *Kliknij Apply changes" (Zastosuj zmiany).*
Rozszerzenia:	*brak*
Częstotliwość stosowania:	*average 5 per day*
Status:	*ukończony*
Właściciel:	*Menadżer*
Priorytet:	*wysoki*

sprzedaży oraz zatrudnienia i podobne.

Powyższe zagrożenia powinny być skutecznie ograniczane przez aplikację internetową.

20.3 Przypadki użycia

20.3.1 Opis narracyjny

Możemy określić podstawowe zadania do wykonania:

- Raportowanie i utrzymanie danych dotyczących produktów, półproduktów i materiałów. Traktujemy konsumpcję jako raportowanie w formach tabelarycznych i drzewnych oraz utrzymanie jako dodawanie, aktualizację i usuwanie (patrz Tabela 20.1).
- Obliczenie struktury wyrobu (BOM) dla określonego zapytania, produktu lub półproduktu (patrz Tabela 20.2).

Tytuły przypadków użycia to zatem żaportowanie i utrzymanie podstawowych danych", żaportowanie i utrzymanie danych strukturyóraz óbliczanie BOM".

20.3.2 Opis częściowo ustrukturyzowany

Powyższa historia jest przedstawiona na schemacie przypadków użycia.

Tablica 20.2: Opis przypadku użycia: raportowanie i utrzymywanie danych dotyczących konstrukcji.

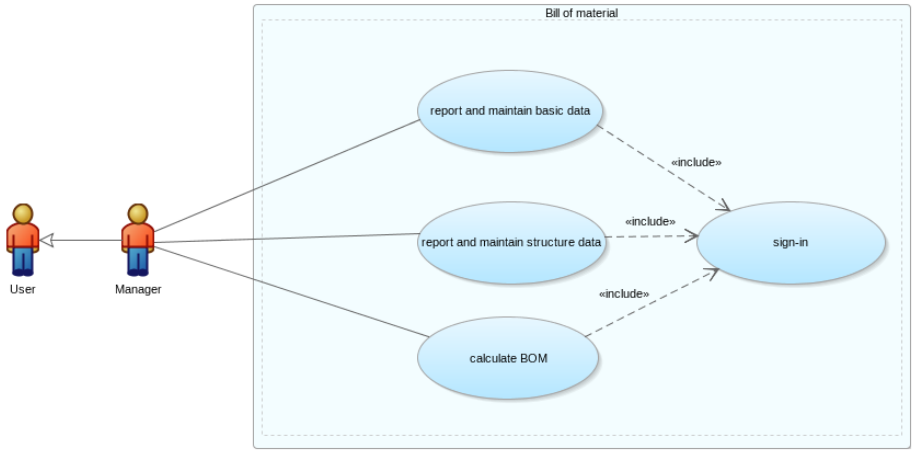
Słowo kluczowe	**Wartość**
ID:	*Ch20-02*
Tytuł:	*Raport i obsługa danych strukturalnych*
Opis:	*Manager wykorzystuje aplikację APEX do raportowania (przeglądania wybranych danych) oraz utrzymywania (wstawiania i usuwania) struktury produktów, półproduktów i materiałów. Zapytanie klienta może składać się z kilku produktów i półproduktów. Wstawienie nowych pozycji lub usunięcie istniejących elementów nie spowoduje (ponownego) obliczenia cen agregatów*
Główny aktor:	*Menedżer*
Warunki wstępne:	*Menedżer ma konto dla instancji APEX.*
Warunki końcowe:	*Po zgłoszeniu lub utrzymaniu danych można wykonać inne działanie, w tym obliczenie zestawienia komponentów.*
Główny	*Scenariusze*
Scenariusz sukcesu:	1. *Otwórz przeglądarkę internetową i zaloguj się do aplikacji.* 2. *Wybierz element menu lub nawigację strony "Bill-of-material"(Lista materiałów).* 3. *Aby dodać komponent, kliknij przycisk Ćreate" (Utwórz).* 4. *Aby usunąć element w strukturze, kliknij "Delete" (Usuń).*
Rozszerzenia:	*brak*
Częstotliwość stosowania:	*średnio 30 miesięcznie*
Status:	*ukończony*
Właściciel:	*Menedżer*
Priorytet:	*wysoki*

Tablica 20.3: Opis przypadku użycia: oblicz BOM.

Słowo kluczowe	**Wartość**
ID:	*Ch20-03*
Tytuł:	*Obliczanie zestawienia komponentów*
Opis:	*Menedżer używa aplikacji APEX do obliczania ceny struktury opisanej przez BOM dla konkretnego towaru, niezależnie od tego, czy jest to produkt, półprodukt czy zapytanie. Uzyskane ceny są aktualizowane dla wszystkich towarów, zagregowanych w strukturze, niezależnie od ich złożoności i głębokości, w tym najwyższego agregatu*
Główny aktor:	*Menadżer*
Warunki wstępne:	*Menedżer ma konto dla instancji APEX.*
Warunki końcowe:	*Po obliczeniu można wykonać dowolną inną akcję, w tym inne obliczenie BOM.*
Główny	*Scenariusze*
Scenariusz sukcesu:	1. *Otwórz przeglądarkę internetową i zaloguj się do aplikacji.* 2. *Wybierz element menu lub nawigację strony "Tree view and calculation"(Widok drzewa i obliczenia).* 3. *Wybierz agregację do obliczenia.* 4. *Kliknij przycisk Ćalculate" (Oblicz). Pojawia się komunikat pokazujący stan obliczeń i obliczoną cenę.* 5. *Rozwiń lub zwiń strukturę drzewa zaznaczonego elementu, aby wyświetlić dane części podrzędnych.*
Rozszerzenia:	*brak*
Częstotliwość stosowania:	*średnio 30 miesięcznie*
Status:	*ukończony*
Właściciel:	*Menadżer*
Priorytet:	*wysoki*

20.3.3 Diagram przypadków użycia

Powyższa historia jest przedstawiona na diagramie przypadków użycia (patrz rysunek 20.1).



Rysunek 20.1: Use case diagram.

20.4 Model danych

20.4.1 Opis modelu danych

W modelu danych dla tego przypadku istnieją tylko dwie jednostki. Encja **ch20_psm** (później nazywana tabelą) opisuje produkty, półprodukty, materiały i zapytania klientów, posiadając tylko trzy atrybuty wspólne dla wszystkich: ID, “description”(opis), “unit of measure”(jednostkę miary) oraz “price per unit of measure”(cenę za jednostkę miary). Wartości wszystkich atrybutów są wymagane. Dlaczego? Przechowywanie instancji bez informacji o nazwie, miarze oraz cenie za jednostkę jest bezsensowne. ID jest unikalnym numerem bez części dziesiętnej. Opis musi zawierać co najmniej jeden znak. Domena jednostek miary jest ograniczona do milimetra, centymetra, metra i sztuki. Ze względu na niewielką liczbę różnych wartości jednostek miary nie wprowadzamy dodatkowej encji, aby przedstawić fakt, że jedna jednostka miary ma wiele wystąpień w **ch20_psm**, podczas gdy jedna instancja **ch20_psm** odnosi się tylko do jednej instancji jednostki miary.

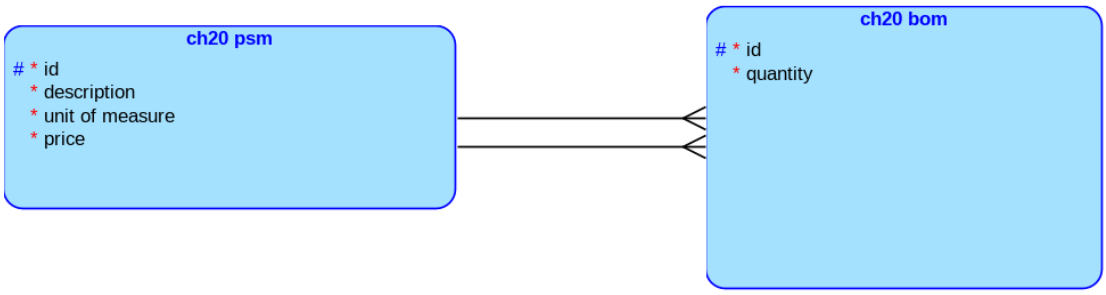
Aby przedstawić arbitralną hierarchiczną strukturę (strukturę wyrobu o dowolnej szerokości i dowolnej głębokości), encja **ch20_bom** ma tylko trzy atrybuty: “identification of sub part”(identyfikację podzespołu),” identification of aggregate”(identyfikację agregatu) i "quantity of sub part contained in aggregate”(ilość podzespołu zawartego w agregacie). Wszystkie atrybuty muszą mieć wartości. Para (identyfikacja podzespołu, identyfikacja agregatu) jest unikalna, jednak dodamy klucz główny generowany jako sekwencja liczb całkowitych, aby skorzystać z niskokodowego środowiska APEX. Jedna instancja **ch20_bom**może mieć podwójną relację z **ch20_psm**: podzespoły odnoszą się do jednej instancji **ch20_psm**, a agregaty odnoszą się do jednej instancji **ch20_psm** W przeciwnym kierunku mamy: jedna instancja **ch20_psm** może odnosić się do wielu podzespołów i wielu agregatów w encji **ch20_bom**. Dlatego mamy dwie relacje 1-do-wielu między **ch20_psm** a **ch20_bom**. Kolejna ważna kwestia dotycząca modelu logicznego danych to unikalność pary (identyfikacja podzespołu, identyfikacja agregatu). Model logiczny w Oracle Data Modeler nie ma śkładni"do wyrażenia tej unikalności jako specjalnego obiektu w bazie danych, zwanej indeksem. Jednak w modelu danych relacyjnych możliwe jest wprowadzenie unikalności określonej pary wartości jako indeksu. W modelu danych relacyjnych unikalny indeks składa się z dwóch pól w tabeli. Ten unikalny indeks gwarantuje, że żadne zduplikowane pary wartości nie zostaną przechowywane w bazie danych - silnik bazy danych będzie chronił programistę i użytkownika końcowego przed naruszeniem integralności danych.

20.4.2 Implementacja reguł biznesowych w bazie danych

Model logiczny i relacyjny jest prosty. W matematyce struktura wyrobu jest specjalnym rodzajem grafu, w którym każdy węzeł jest połączony tylko z jednym węzłem nadrzędnym, z wyjątkiem węzła korzenia, który nie ma węzła nadrzędnego. Bardzo łatwo jest przerwać hierarchię poprzez wprowadzenie dodatkowego połączenia między węzłami lub usunięcie połączenia. W kontekście bazy danych zmiany w połączeniach mają wpływ na integralność danych. Przechowywanie struktury cyklicznej w **ch20_bom** spowodowałoby, że obliczenia struktury wyrobu nigdy się nie kończą. W kontekście reguł biznesowych jest niemożliwe i zabronione, aby samochód (produkt) składał się z kół (półproduktów), a jednocześnie koło składało się z samochodów. Samochód nie może również składać się z samego siebie. Aby zapobiec naruszeniu reguł biznesowych, wprowadzimy mechanizm wyzwalacza (innego obiektu bazy danych), który zwykle jest reprezentowany przez kilka linii kodu w języku PL/SQL. Wyzwalacz zostanie wykonany przed zapisaniem lub aktualizacją **ch20_bom** . Taki kod uniemożliwi nowe wstawienie do **ch20_bom** , jeśli para w odwrotnej kolejności już istnieje lub oba nowe podzespoły i agregaty mają tę samą wartość. Implementacja tego rodzaju reguł nie może zostać przedstawiona w modelu logicznym ani relacyjnym. Niemożliwe jest również przedstawienie sposobu, w jaki zostanie zaimplementowane obliczenie BOM w modelach danych. W tym rozdziale zostanie przedstawiony kod, który odwraca hierarchię (od dołu do góry) i zaczyna sumować całkowity koszt od dołu - czyli najpierw oblicza koszt materiału w półproduktach, a następnie koszt półproduktów w produkcie. Pamiętaj, że model danych składający się z dwóch jednostek lub dwóch tabel jest zdolny do przedstawienia dowolnej złożoności (szerokości i głębokości hierarchii) struktury BOM. Zacznij więc od modeli danych.

20.4.3 Logiczny model danych

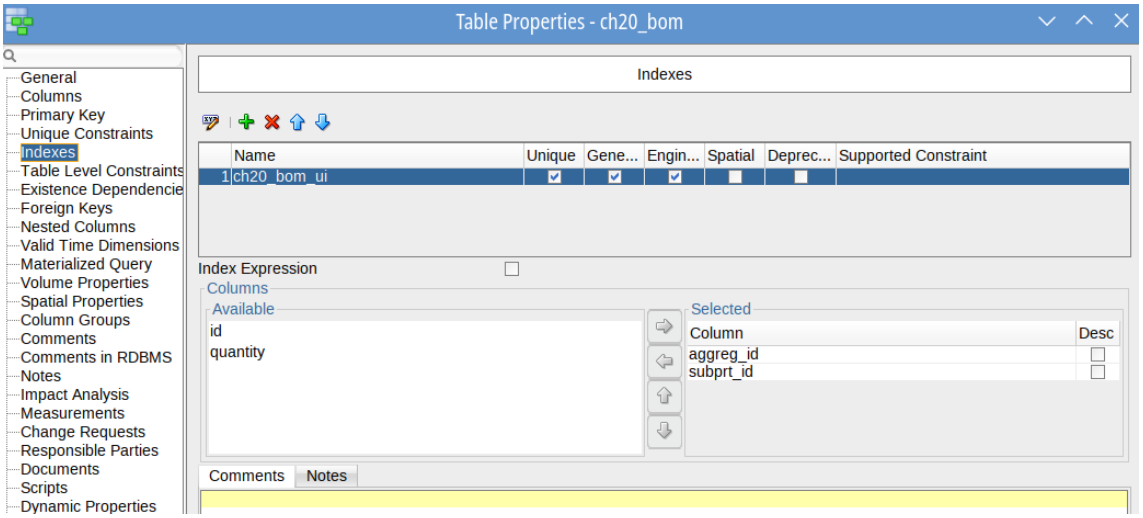
Logiczny model danych przedstawiono na rysunku 20.2.



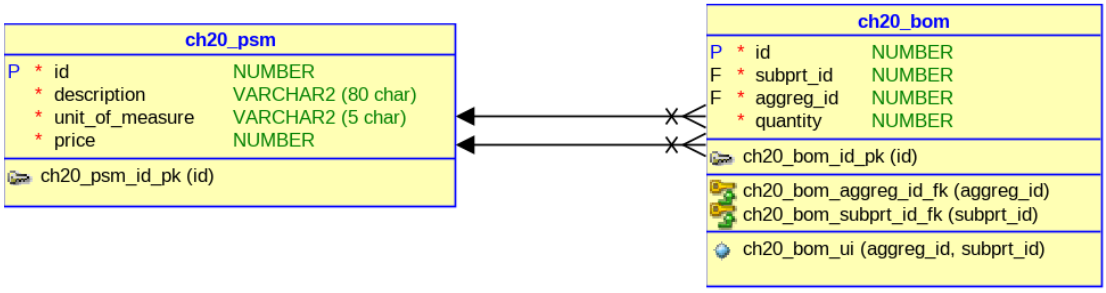
Rysunek 20.2: Logiczny model danych.

20.4.4 Relacyjny model danych

Automatyczna transformacja z modelu logicznego danych do modelu relacyjnego danych w Oracle SQL Data Modeler jest dostarczana przez funkcję Inżynieria do modelu relacyjnego. Następnie wprowadzamy unikalny indeks *ch20_bom_ui* , składający się z dwóch pól: *aggreg_id* i *subprt_id* (zob.rysunek 20.3). Wynikowy model relacyjny danych gotowy do wyeksportowania jako skrypt SQL jest pokazany na rysunku 20.4.



Rysunek 20.3: Definicja unikalnego indeksu w Oracle SQL Data Modeler.



Rysunek 20.4: Relational data model.

Oracle SQL Data Modeler generuje również skrypt SQL do tworzenia tabeli, sekwencji i wyzwalacza. Wybierz wszystkie tabele w modelu relacyjnym i użyj funkcji *File* > *Export* > *DDL File* , aby uzyskać skrypt. Zapisz skrypt i sprawdź kolejność definicji tabel. Tabela *ch20_psm* musi być zdefiniowana pierwsza, a następnie *ch20_bom*. Poniższy kod definiuje tabelę *ch20_psm* i ustawia ograniczenia:

```
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' ) );
```

Poniższy kod definiuje *ch20_bom* tabeli, ustawia ograniczenia i definiuje unikalny indeks:

```
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 );
```

Zapisz skrypt z instrukcjami, które tworzą tabele z kluczami podstawowymi i obcymi, sprawdzają ograniczenia i unikalny indeks. Teraz nadszedł czas na wygenerowanie tabel w bazie danych Oracle. Możemy zaimportować wygenerowany skrypt i wykonać go w APEX.

20.4.5 Obiekty w APEX

Przejdź do przeglądarki internetowej w przestrzeni roboczej APEX. W tej sekcji wykonamy następujące czynności:

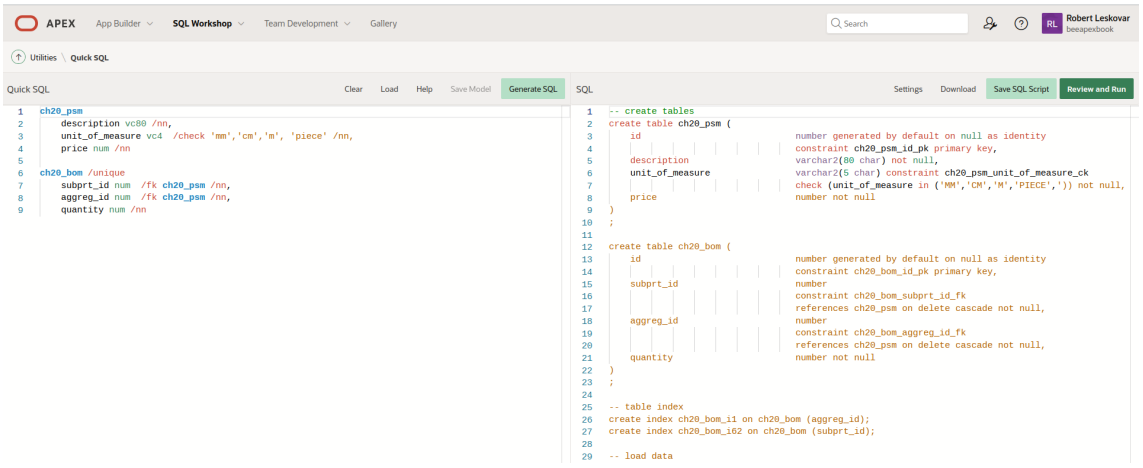
- utworzymy tabele i indeksy
- utworzymy funkcję i procedurę
- utworzymy wyzwalacz

Istnieją dwie opcje tworzenia tabel i indeksów. Jeśli używasz Oracle Data Modeler do wygenerowania skryptu, najpierw zaimportuj go (SQL Workshop > SQL Scripts > Upload) i uruchom. Inną opcją jest wygenerowanie powyższego skryptu za pomocą Quick SQL, jeśli chcesz użyć tylko APEX do wygenerowania takiego skryptu. Przejdź do SQL Workshop > Utilities > Quick SQL. Umieść następujący tekst w obszarze tekstowym Quick SQL (lewa część okna):

```
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
```

Kliknij przycisk “Generate SQL”. Tabele mają takie same właściwości jak tabele wygenerowane w Oracle Data Modeler (zob. Figure 20.5).



Rysunek 20.5: Generowanie skryptu SQL za pomocą narzędzia Quick SQL.

Spójrz na linie 26 i 27. Te dwie linie, choć poprawne, zostaną później edytowane. Kliknij przycisk "Click Save SQL Script and Review and Run". Zastąp linie 26 i 27 (utwórz dwa indeksy) jednym poleceniem:

```
create unique index ch20_bom_ui on ch20_bom (aggreg_id, subprt_id);
```

Teraz uruchom skrypt. Wygenerowane zostaną dwie tabele i jeden unikalny indeks. Zauważ, że uruchomienie tego skryptu nie powiedzie się, jeśli już uruchomiłeś skrypt wygenerowany za pomocą Oracle Data Modeler. Powodem jest istnienie obiektów bazy danych. Aby utworzyć funkcję, wprowadź następujący tekst w SQL Workshop > SQL Commands:

```
1  create or replace FUNCTION  CH20_GET_PRICE (
2    xpsm IN   NUMBER
3    ) RETURN NUMBER
4  IS
5    y_price number;
6  BEGIN
7    SELECT price INTO y_price FROM ch20_psm  WHERE ID = xpsm;
8    return y_price;
9  EXCEPTION
10   WHEN OTHERS THEN
11     return(-505);
12 END CH20_GET_PRICE;
```

Ta funkcja zwróci cenę (produktu, półproduktu, materiału, zapytania) dla danego identyfikatora. Jeśli wystąpią nieoczekiwane problemy z bazą danych (utracenie połączenia, awaria serwera, błąd sprzętowy, brak danych dla określonego identyfikatora), zostanie zwrócona wartość ujemna -505. Zaznacz cały tekst i uruchom. Otrzymasz informację zwrotną, że funkcja została utworzona. Aby utworzyć procedurę obliczającą koszty struktury wyrobu, wprowadź następujący tekst w SQL Workshop > SQL Commands:

```
1  create or replace PROCEDURE  CH20_CALCULATE_BOM
2    (xpsm IN NUMBER,
3    ystatus OUT INTEGER,
4    yprice OUT NUMBER)
5  -- creation date: 20.1.2023
6  -- author: Robert Leskovar
7  -- input parameter: xpsm, id of product or semi-product
```

```
8  -- output parameters 1: status of calculation:
9                  0=not calculated, 1=succesfully calculated
10 -- output parameters 2: price calculated
11 -- description: calculation of the total costs of given product
12 --                              and all underlaying levels
13 -- exceptions: system and user defined
14 -- date of testing: 20.1.2023
15 -- author of test: swqlab
16 -- status: APPROVED
17 is
18   v_rezult   NUMBER(38, 2);
19   v_psm        NUMBER;
20   v_price    NUMBER(38, 2);
21   e_negative_price EXCEPTION;
22   CURSOR c_ds IS
23    WITH STRUK(hier, subprt_id, aggreg_id, quantity) AS
24     (SELECT 1 AS hier, subprt_id, aggreg_id, quantity
25      FROM CH20_BOM JOIN CH20_PSM ON subprt_id = CH20_PSM.ID
26         WHERE aggreg_id = xpsm
27    UNION ALL
28     SELECT hier+1, C1.subprt_id, C1.aggreg_id, C1.quantity
29      FROM CH20_BOM C1 JOIN CH20_PSM P ON C1.subprt_id = P.ID
30       JOIN STRUK C2 ON C1.aggreg_id = C2.subprt_id)
31    SELECT hier, subprt_id, aggreg_id, quantity
32     FROM STRUK ORDER BY hier DESC, aggreg_id, subprt_id;
33 BEGIN
34   ystatus := 0;
35   v_rezult := 0.0;
36   v_psm := 0;
37   SAVEPOINT old_state;
38   FOR r_ds IN c_ds LOOP
39    v_price:= CH20_GET_PRICE(r_ds.subprt_id);
40    IF ( v_price <= 0 ) THEN
41     RAISE e_negative_price;
42    END IF;
43    IF ( r_ds.aggreg_id = v_psm ) THEN
44     v_rezult := v_rezult + v_price * r_ds.quantity;
45     UPDATE ch20_psm SET price = v_rezult WHERE ID = r_ds.aggreg_id;
46    ELSE
47     v_psm := r_ds.aggreg_id;
48     v_rezult := v_price * r_ds.quantity;
49     UPDATE ch20_psm SET price = v_rezult WHERE ID = r_ds.aggreg_id;
50    END IF;
51   END LOOP;
52   COMMIT;
53   ystatus := 1;
54   yprice := v_rezult;
55 EXCEPTION
56   WHEN e_negative_price THEN
57    ROLLBACK TO old_state;
```

```
58     ystatus := 0;
59     yprice := -505;
60   WHEN OTHERS THEN
61     ROLLBACK TO old_state;
62     ystatus := 0;
63     yprice := -100;
64 end CH20_CALCULATE_BOM;
```

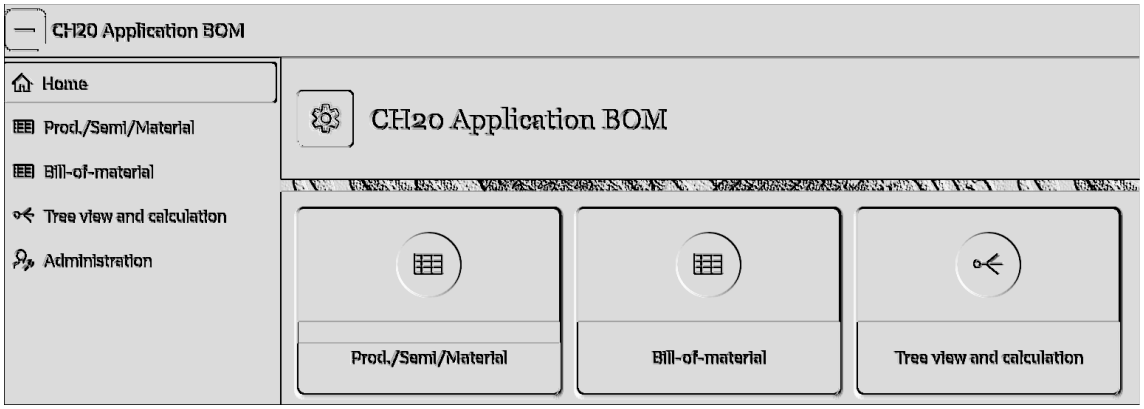
Ta procedura zwróci stan obliczeń i cenę (produktu, półproduktu, materiału, zapytania) dla danego identyfikatora, chyba że którykolwiek składnik ma wartość ujemną. Jeśli wystąpią nieoczekiwane problemy z bazą danych (utracenie połączenia, awaria serwera, błąd sprzętowy, brak danych dla określonego identyfikatora), zostanie zwrócona wartość ujemna. Jeśli czytelnik nie jest zaznajomiony z PL/SQL, wystarczy przeczytać komentarze w procedurze. W przeciwnym razie zostanie utworzony dość złożony kursor z wyrażeniem zapytania wspólnego (WITH). Ten kursor zwraca sumę dwóch zbiorów. Zapewnia kompletną strukturę agregatu, która jest uporządkowana od dołu do góry. Ceny pośrednich agregatów są przechowywane w odpowiednich wierszach i sumowane, aż wszystkie wiersze w kursorze zostaną przetworzone. Ostatnim obiektem do utworzenia jest wyzwalacz. Wyzwalacz zapewni, że żaden agregat i podzespoły w strukturze wyrobu nie mogą mieć tej samej wartości (samochód składa się z samochodu), a utworzona jest relacja przejścia (jeśli samochód składa się z kół, koło nie może składać się z samochodów). Aby utworzyć wyzwalacz, który stosuje określone reguły, wprowadź następujący tekst w SQL Workshop > SQL Commands:

```
1  create or replace trigger CH20_TRG_BOM_RULES
2    before insert or update on ch20_bom
3    for each row
4
5    declare
6     xsubprt numeric :=:new.subprt_id;
7     xaggreg numeric :=:new.aggreg_id;
8     xcount numeric;
9    begin
10    select count(subprt_id) into xcount from ch20_bom
11     where subprt_id = xaggreg and aggreg_id = xsubprt;
12    if xsubprt = xaggreg or xcount > 0 then
13     :new.subprt_id :=null;
14    end if;
15 end;
```

Dzięki temu wyzwalaczowi zapobiegniemy wstawianiu poprzez wymuszenie wartości NULL w polu tabeli, które musi mieć wartość. Pierwszeństwo definicji NOT NULL w kolumnie tabeli nad mechanizmem wyzwalacza spowoduje, że żadne dane nie zostaną wstawione, jeśli ta reguła biznesowa zostanie naruszona. Jednakże zabezpieczyliśmy hierarchię przed przekształceniem się w graf z cyklem. Jeśli pominęlibyśmy utworzenie tego wyzwalacza, powinniśmy przeszkolić użytkownika końcowego, aby nie wpadał w pułapkę cyklu. O wiele lepiej jest zapobiegać niż leczyć, prawda?

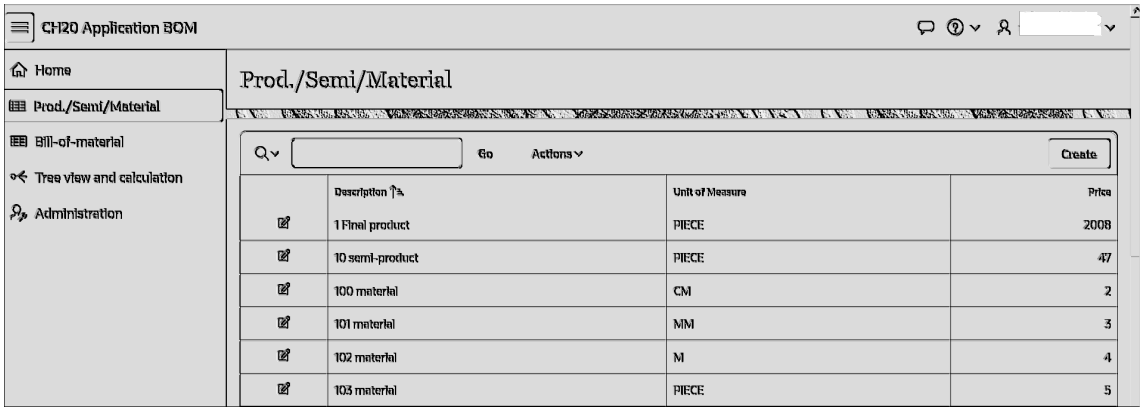
20.5 Interfejsy aplikacji

W poniższych rysunkach przedstawiono interfejsy aplikacji. Na początek prezentujemy szkic okna wejściowego. Użytkownik będzie nawigował po zarządzaniu podstawowymi danymi, zarządzaniu danymi struktury oraz obliczeniach, zarówno za pomocą menu po lewej stronie, jak i nawigacji strony (patrz rysunek 20.6).

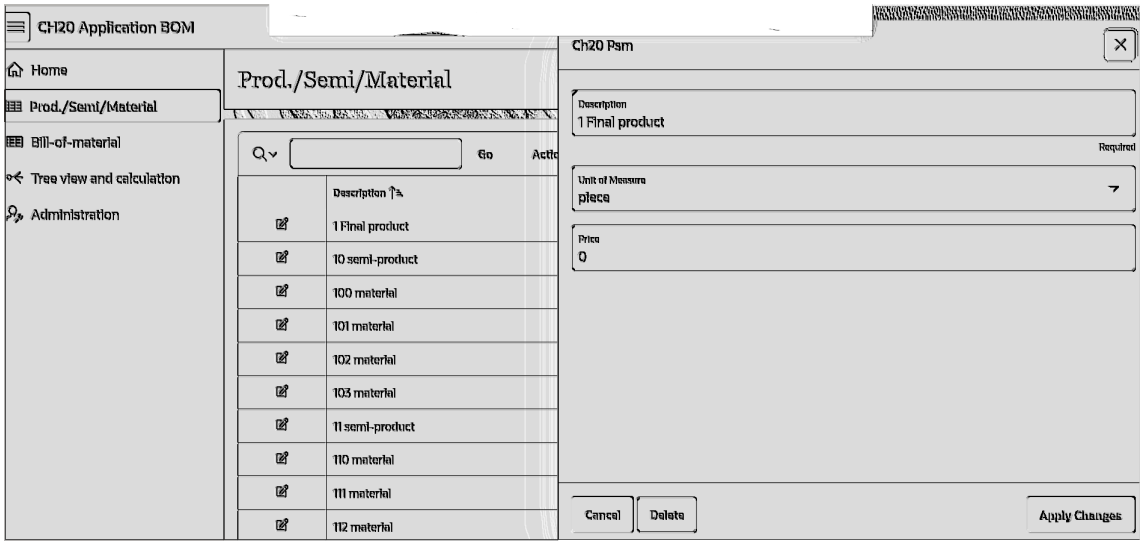


Rysunek 20.6: Strona główna aplikacji.

Kolejne dwa rysunki (20.7 i 20.8) przedstawiają szkice raportu i formularza dla produktów, półproduktów, materiałów i zapytań. Dostępne są opcje dodawania, aktualizacji i usuwania.



Rysunek 20.7: Zarządzanie podstawowymi danymi - raport.



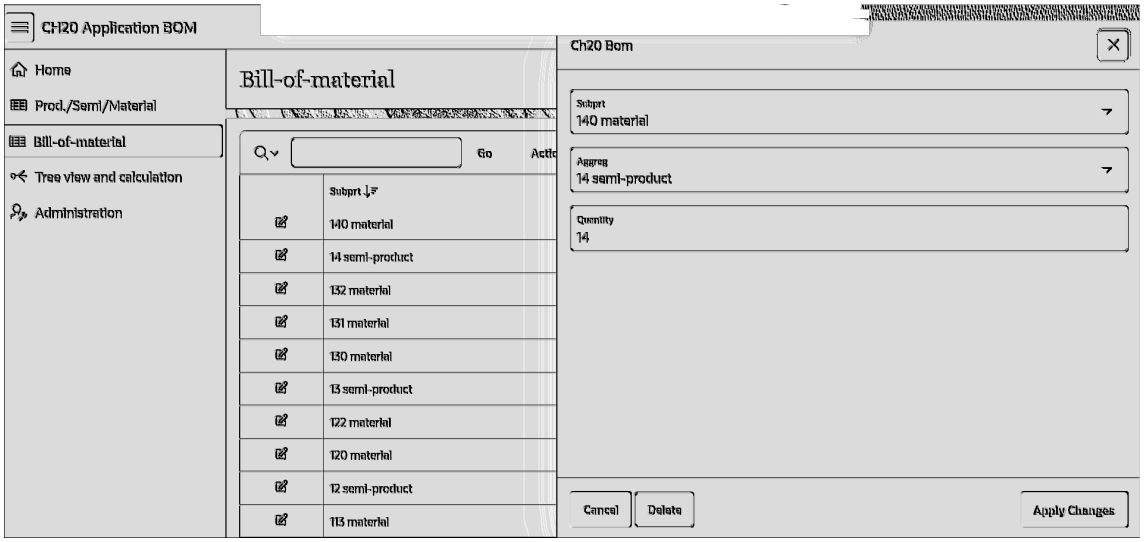
Rysunek 20.8: Zarządzanie podstawowymi danymi - formularz.

Kolejne dwa rysunki (20.9 i 20.10) przedstawiają szkice raportu i formularza dla struktury, listy

materiałów. Dostępne są opcje dodawania, aktualizacji i usuwania.

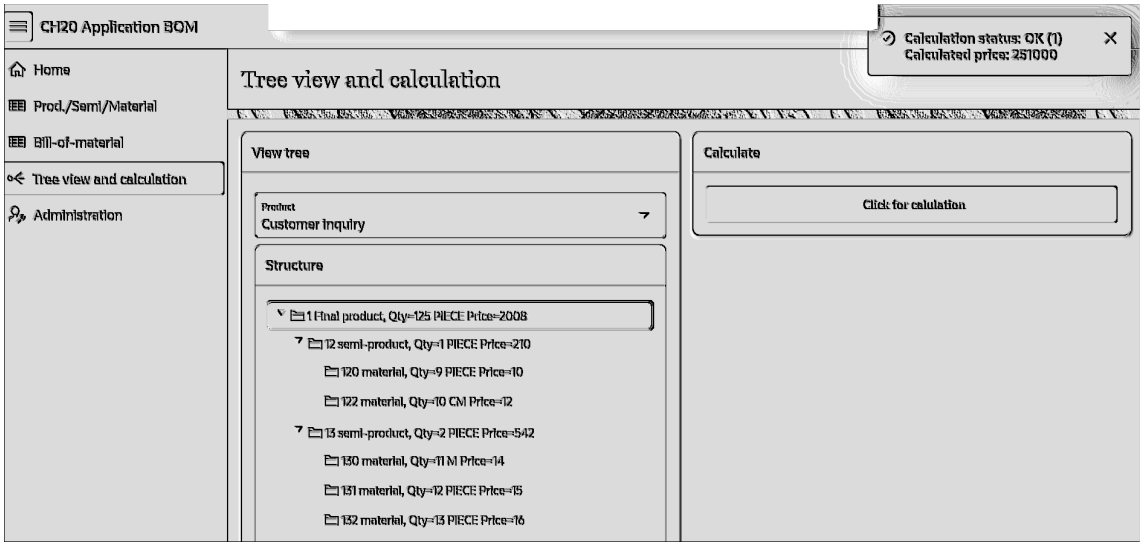


Rysunek 20.9: Zarządzanie danymi struktury - raport.



Rysunek 20.10: Zarządzanie danymi struktury - formularz.

Szkic (20.11) prezentuje interfejs do przeprowadzania obliczeń dla dowolnego agregatu (półprodukt, produkt, zapytanie). Użytkownik najpierw wybiera element, a następnie naciska przycisk “Calculate”. Status obliczeń i obliczona cena są wyświetlane jako wyskakujące okno, a komponent drzewa jest pokazywany. Użytkownik może rozszerzać i zwijać hierarchię.



Rysunek 20.11: Strona do obliczania zestawienia materiałowego.

20.6 Dodatkowe materiały do nauki

Możesz znaleźć następujące uzupełniające materiały szkoleniowe:

- wyeksportowaną aplikację
- poradniki wideo

Wszystkie dodatkowe materiały są dostępne na publicznej stronie BeeAPEX . Zaloguj się jako użytkownik gość (nie jest wymagane hasło). Znajdź podręcznik w sekcji “Books”(Książki), skrypty w folderze Part 2 > Chapter20 w sekcji “Scripts” (Skrypty) i przewodniki wideo w “Collection of video guides”. Materiał na krótkie kursy w “Short courses section”.

20.6.1 Eksportowane aplikacje

Wyeksportowana aplikacja jest spakowana. Podczas instalacji tworzone są tabele, indeksy, funkcje, procedury i wyzwalacze, a także dane są wypełniane. Deinstalacja usuwa wszystkie obiekty bazy danych używane w tej aplikacji.

20.6.2 Przewodniki wideo

Przewodnik wideo pokazuje każdy krok w tworzeniu aplikacji.

20.7 Pytania

1. Jak zmieniłbyś logiczny model danych, aby zaimplementować nową encję dla jednostek miary, ponieważ chcemy uwzględnić dziesiątki jednostek?
2. W jaki sposób można określić Quick SQL, aby odzwierciedlić powyższą zmianę?
3. Jakie będą konsekwencje rezygnacji z unikalnego indeksu ch20_bom_ui?

20.8 Odpowiedzi

1. Stosujemy nowe *ch20_uom* encji i ustawiamy co najmniej dwa atrybuty (ID i opis jednostki miary). Następnie ustawiamy relację 1 do wielu między *ch20_psm* a *ch20_uom*.
2. W Quick SQL najpierw definiujemy *ch20_uom* tabeli jako:

```
ch20_uom
  description vc80
```

and change the definition *ch20_psm* as:

```
ch20_psm
  description vc80 /nn,
  unit_of_measure num /fk ch20_uom
  price num
```

3. Rezygnacja z unikalnego indeksu ch20_bom_ui umożliwiłaby przechowywanie wielu par (podczęść, agregacja) o tych samych wartościach. Tabela zajmowałaby więcej miejsca na dysku. Nie wpłynie to na funkcjonalność obliczeń, jednak przetwarzanie potrwa dłużej i zużyje więcej pamięci RAM.



21. Odżywianie i zarządzanie dietą

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

21.1 Biznesowy pogląd na sprawę

Nasz przykład to najpopularniejsza w Grecji marka mediów o tematyce kulinarnej, docierająca do ponad 500 000 użytkowników. Firma jest dumną właścicielką czołowego czasopisma o tematyce kulinarnej w Grecji oraz edycji cyfrowej, najpopularniejszej witryny internetowej poświęconej kulinariom w kraju, rosnącej liczby wydarzeń na żywo, w tym największej imprezy kulinarnej w Grecji - Greek Food Show - oraz ogromnie udanej serii książek kucharskich. Firma postanowiła stworzyć i promować narzędzie internetowe, które umożliwia użytkownikom dostęp do tysięcy przepisów za pośrednictwem ich strony internetowej. Ponadto użytkownicy mogą spersonalizować ulubione przepisy, dodając lub usuwając składniki, a usługa dostarczy informacje o wartości odżywczej nowych przepisów, pozwalając użytkownikom stworzyć własną spersonalizowaną książkę kucharską ulubionych przepisów.

21.2 Definicja problemu

Użytkownicy, którzy lubią gotować, często napotykają ograniczenia dotyczące tego, co mogą zjeść, czy to z powodu problemów zdrowotnych, takich jak cukrzyca, wysoki poziom cholesterolu, lub z osobistych preferencji żywieniowych, takich jak weganizm czy dieta bezglutenowa. Jednym z problemów istniejących zasobów internetowych z przepisami jest to, Jedną z kwestii związanych z istniejącymi internetowymi zasobami przepisów jest to, że często nie oferują kompleksowych alternatyw, aby sprostać tym ograniczeniom. W rezultacie użytkownicy mogą odczuwać frustrację z powodu ograniczonej dostępności informacji i całkowicie porzucić poszukiwania przepisów, co prowadzi do niskiego zaangażowania.

21.3 Przypadki użycia

21.3.1 Opis narracyjny

Stworzymy aplikację Oracle APEX, która umożliwi użytkownikom przeglądanie i dodawanie różnorodnych zdrowych przepisów. Aplikacja będzie zawierała szczegółowy opis każdego przepisu, w tym listę składników oraz informacje o wartościach odżywczych każdego składnika. Użytkownicy będą mogli również dodawać swoje własne przepisy i odpowiadające im składniki za pomocą formularza. Wartości odżywcze każdego przepisu, takie jak liczba kalorii, całkowita ilość białka,

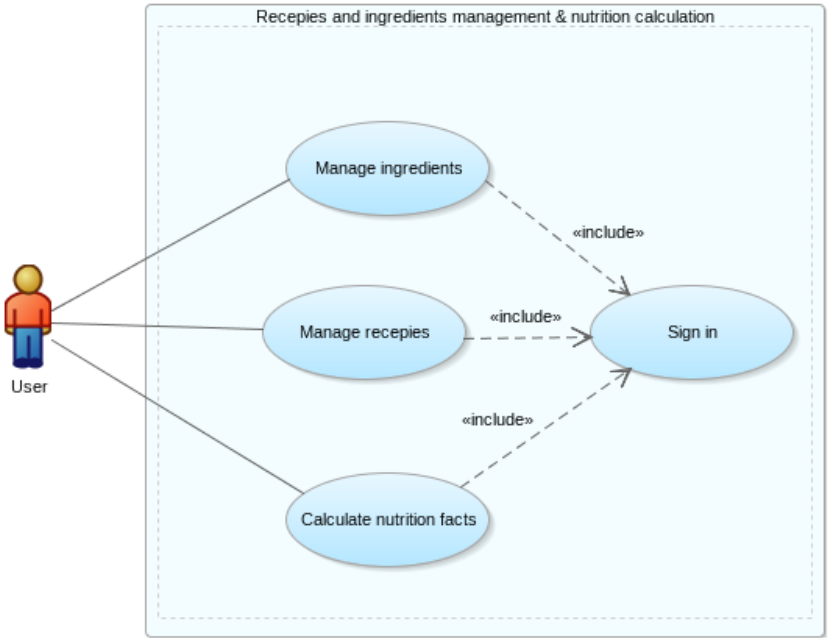
liczba porcji i czas przygotowania, zostaną przedstawione w postaci diagramów, aby ułatwić odczyt, zamiast prostych tabel z danymi.

21.3.2 Opis częściowo ustrukturyzowany

Opis częściowo ustrukturyzowany przedstawiono w tabeli 21.1.

21.3.3 Schemat przypadków użycia

Powyższa historia jest przedstawiona na schemacie przypadków użycia (zob.rysunek 21.1).



Rysunek 21.1: Diagram przypadków użycia.

21.4 Model danych

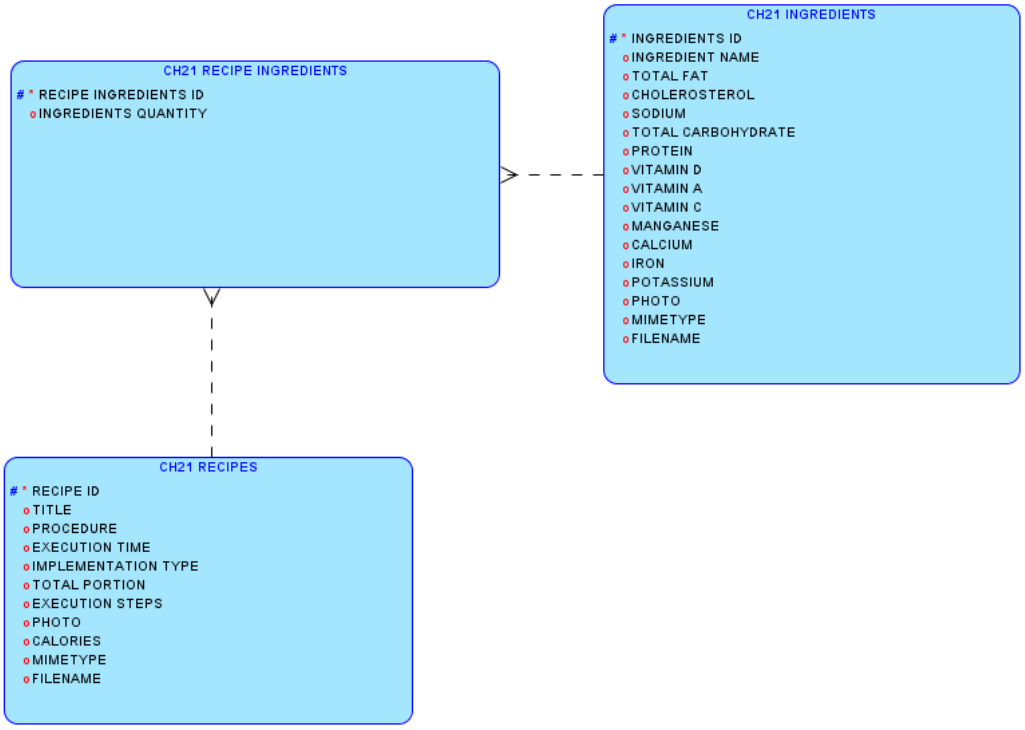
Model danych przedstawia trzy jednostki: “recepies, ingredients and ingrediants in recepies “

21.4.1 Logiczny model danych

Logiczny model danych pokazano na rysunku 21.2.

Tablica 21.1: Opis przypadku użycia: Dostęp do receptur i składników oraz obliczanie wartości odżywczych

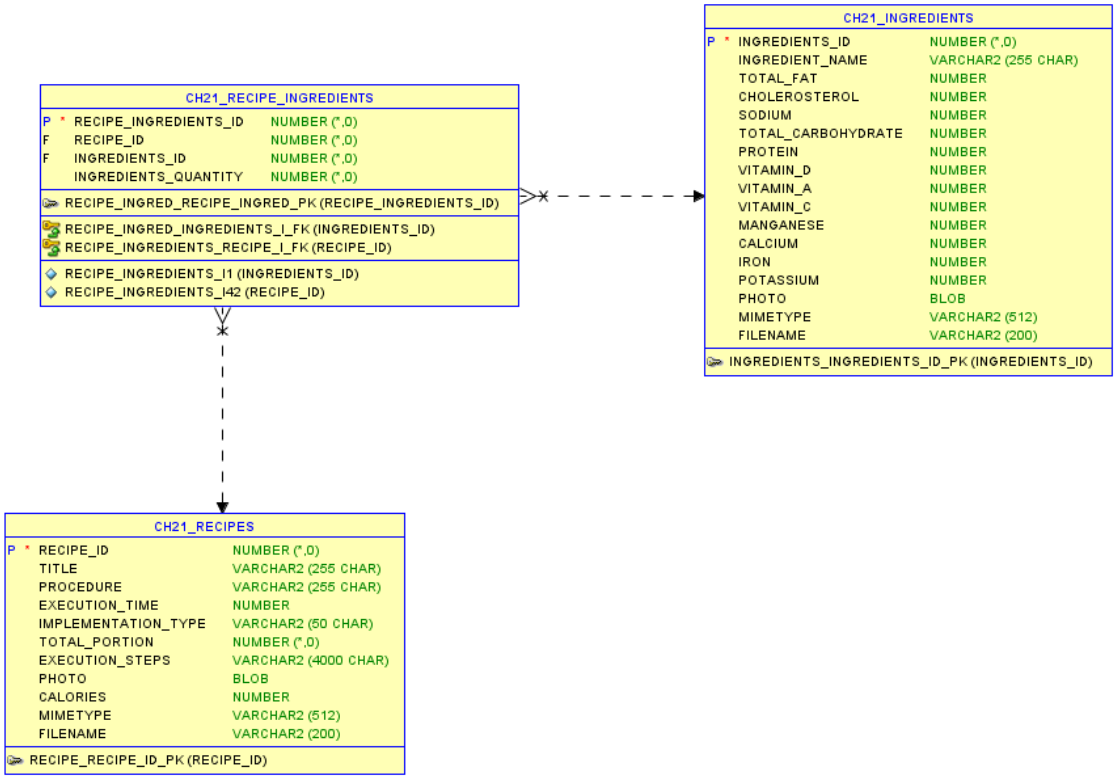
Słowo kluczowe	**Wartość**
ID:	*Ch21-01*
Tytuł:	*Platforma oferuje dostęp do różnych gotowych receptur i składników, a także możliwość dodawania lub usuwania składników w celu dostosowania przepisów do indywidualnych preferencji. Użytkownicy mogą utworzyć nowy spersonalizowany przepis, wybierając preferowane składniki, a usługa obliczy wartości odżywcze dla przepisu.*
Opis:	*Korzystając z aplikacji Apex, użytkownicy mogą przeglądać i wybierać przepisy, a także konfigurować dane receptur, takie jak tworzenie, aktualizowanie i usuwanie składników. Zarówno receptury, jak i składniki są zdefiniowane przez zestaw atrybutów. Użytkownicy mogą utworzyć nowy przepis, dodając lub usuwając składniki i obliczając zaktualizowane fakty żywieniowe. Wyniki są wyświetlane na diagramach, które są aktualizowane w czasie rzeczywistym na podstawie zmian wprowadzonych w recepturze.*
Główny aktor:	*Użytkownik*
Warunki wstępne:	Użytkownik ma konto dla instancji APEX.
Warunki postu:	Po aktualizacji i wyborze przepisu użytkownik może obliczyć fakty dotyczące składników . Po obliczeniu wartości odżywczych dla przepisu użytkownicy mogą powtórzyć proces dla innych przepisów, dodając lub usuwając składniki, aby utworzyć nowe spersonalizowane przepisy.
Główny	Scenariusz
Scenariusz sukcesu:	1. Otwórz przeglądarkę internetową i zaloguj się do aplikacji. 2. Wybierz opcję “recipes” lub “ingredients” z menu lub nawigacji strony. 3. Aby dodać przepis, kliknij przycisk “add a recipe“. Aby dodać składnik, kliknij przycisk “Add Ingredient”. 4. Aby edytować przepis lub składnik, kliknij symbol ołówka do konkretnego wpisu. 5. Po wprowadzeniu zmian kliknij przycisk “Apply Changes”, aby je zapisać. Aby usunąć dane, kliknij przycisk “Delete”. 6. Aby sprawdzić informacje o wartości odżywczej dla przepisu, przejdź do menu lub nawigacji strony i wybierz “Recipe total nutrition facts “. 7. Podane szczegóły żywieniowe obejmują wielkość porcji żywności, którą można przygotować w oparciu o określone ilości składników, całkowitą liczbę białka i kalorii w przepisie oraz całkowity czas potrzebny na przygotowanie przepisu.
Rozszerzenia:	*brak*
Częstotliwość stosowania:	*Użytkownicy serwisu publikują średnio 5 przepisów dziennie, czyli około 1 tysiąc przepisów rocznie. Dziennie jest około 10 tysięcy użytkowników, którzy tylko przeglądają.*
Stan:	Gotowy
Właściciel:	Użytkownik
Priorytet:	*umiarkowany*



Rysunek 21.2: Logiczny model danych.

21.4.2 Relacyjny model danych

Relacyjny model danych pokazano na rysunku 21.3.



Rysunek 21.3: Relacyjny model danych.

21.4.3 QuickSQL i skrypt SQL Script

Podany kod wygląda na zestaw instrukcji generujących przykładowe dane za pomocą narzędzia Quick SQL. Tabele, które musimy utworzyć dla naszej aplikacji, to CH21_RECIPES, CH21_RECIPE_INGREDIENTS oraz CH21_INGREDIENTS. Pola dla każdej tabeli powinny być ustrukturyzowane w następujący sposób:

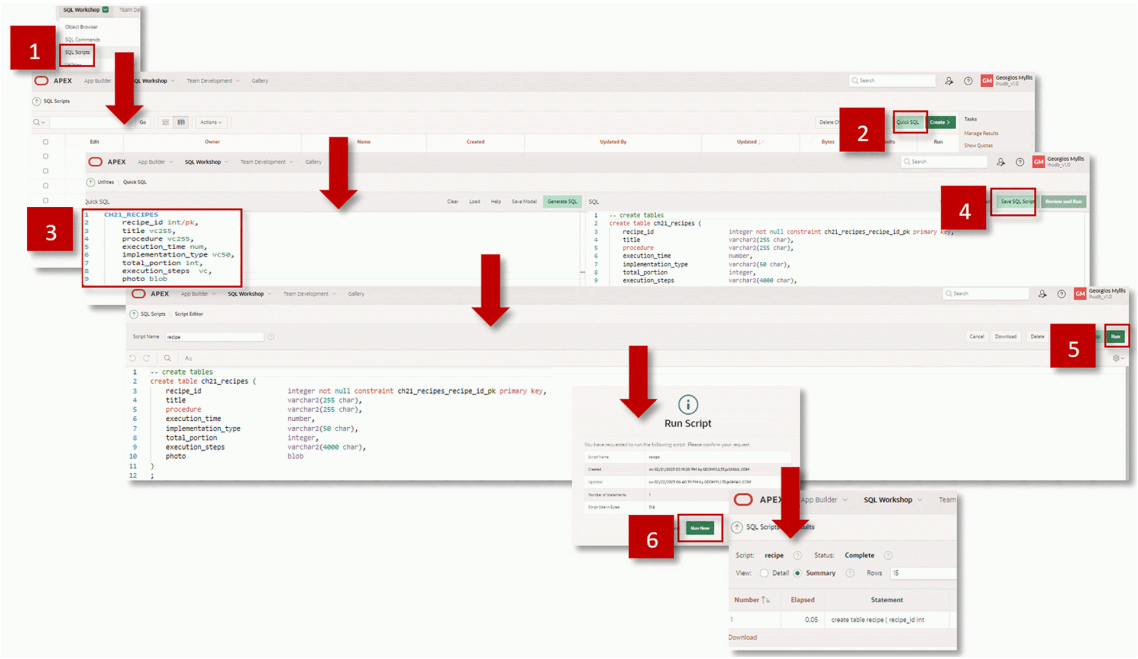
```
CH21_RECIPES
    RECIPE_ID int/pk,
    TITLE VC255,
    PROCEDURE VC255,
    EXECUTION_TIME NUM,
    IMPLEMENTATION_TYPE VC50,
    TOTAL_PORTION int,
    EXECUTION_STEPS  VC,
    PHOTO BLOB

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
```

Aby uruchomić kod Quick SQL, musimy najpierw uzyskać dostęp do warsztatu SQL i wybrać **SQL Scripts**. Następnie możemy wybrać **Quick SQL**, aby utworzyć tabele. Proces jest prosty; musimy tylko określić nazwę tabeli i jej pola, a także wszystkie tabele podrzędne związane z naszą tabelą. Rysunek 21.4 ilustruje, jak szybko tworzyć tabele za pomocą tej funkcji.



Rysunek 21.4: Szybkie narzędzie SQL do tworzenia skryptu SQL.

Nasz następny krok w celu prawidłowego funkcjonowania naszych tabel to zapewnienie, że każde dodanie do tabeli jest unikalne i że główny klucz każdej tabeli jest automatycznie zwiększany. Aby to osiągnąć, musimy utworzyć trzy SEKWENCJE.

Zwiększ identyfikator przepisu:

```
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;
```

Zwiększ ID składnika:

```
CREATE SEQUENCE seq_INGREDIENTS
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;

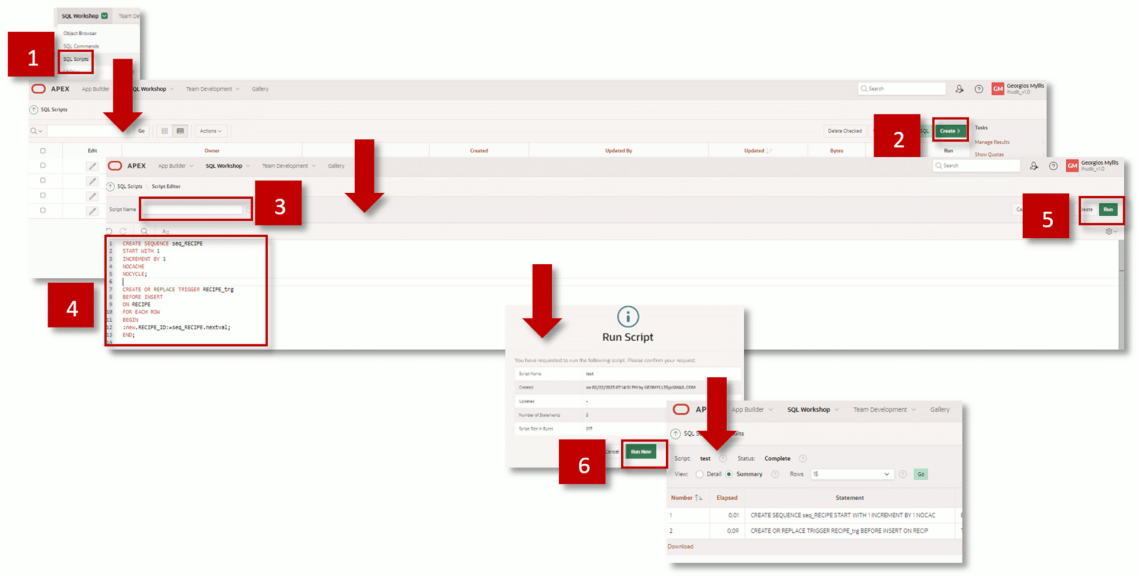
CREATE OR REPLACE TRIGGER INGREDIENTS_trg
BEFORE INSERT
ON INGREDIENTS
FOR EACH ROW
BEGIN
:new.INGREDIENTS_ID:=seq_INGREDIENTS.nextval;
END;
```

Zwiększ identyfikator składnika przepisu:

```
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;
```

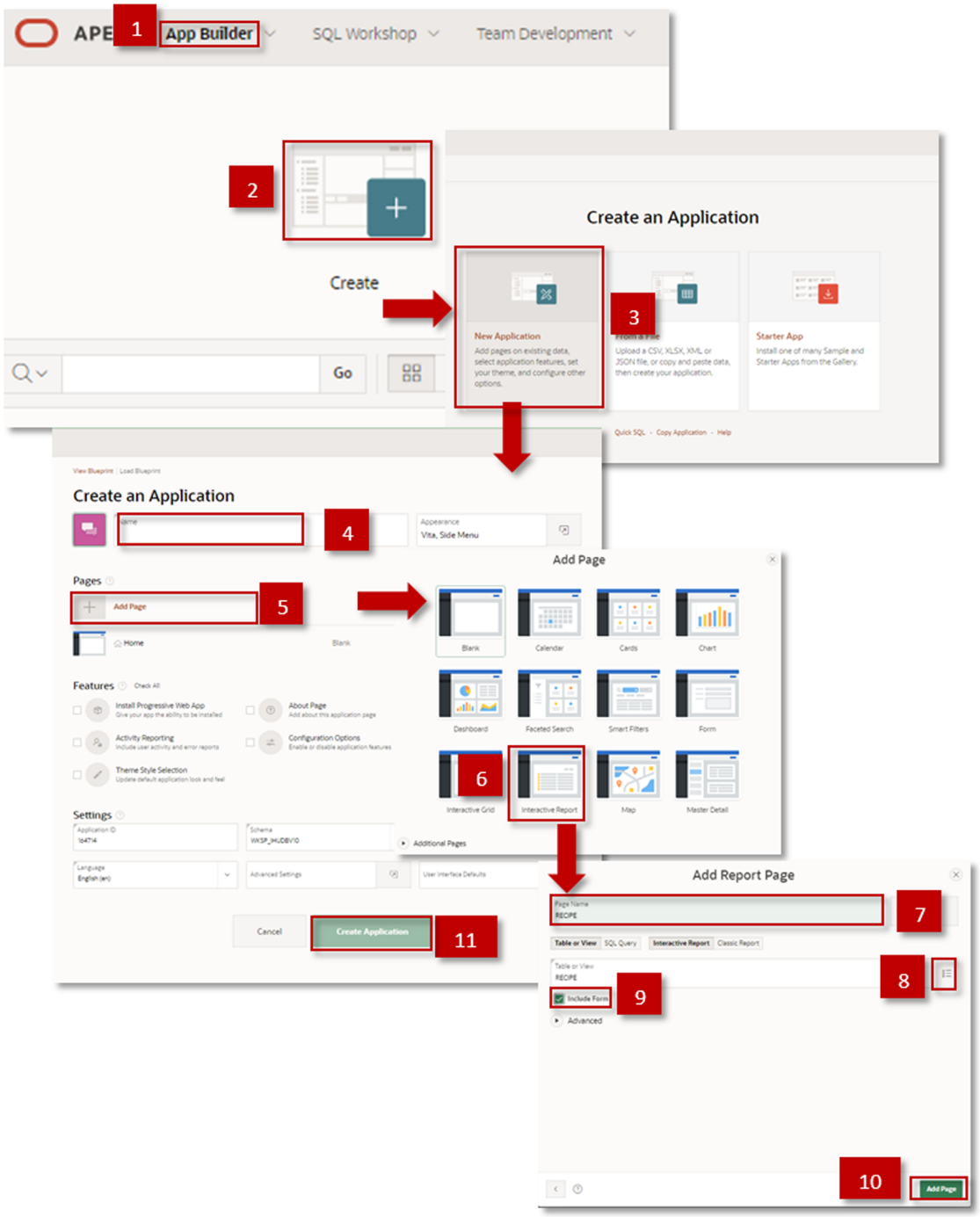
Aby uruchomić SEKWENCJĘ, musimy przejść do **SQL Workshop** i kliknąć **SQL Scripts**. Następnie możemy kliknąć przycisk **Create** i osobno skopiować i wkleić kod dla każdej SEKWENCJI. Po nadaniu nazwy skryptowi możemy kliknąć przycisk "Run", aby wykonać skrypt. Zapewni to, że każdy rekord w tabelach jest unikalny i eliminuje konieczność ręcznego śledzenia ostatniego dodanego rekordu. Wszystko będzie robione automatycznie. Przykład uruchamiania SEKWENCJI w środowisku APEX został przedstawiony na rysunku 21.5.



Rysunek 21.5: Procedura skryptu SQL do zbudowania SEQUENCE.

21.5 Interfejsy aplikacji

Aby stworzyć aplikację tak jak na rysunku 21.6, kliknij w **App Builder** i następnie na **Create** .Następnie pojawi się nowe okno, w którym musimy podać nazwę naszej aplikacji oraz wybrać funkcje, których chcemy, aby nasza aplikacja była wyposażona. Tabele, które będziemy używać w naszej aplikacji to: RECIPE, RECIPE INGREDIENTS, INGREDIENTS, jak wcześniej wyjaśniono.



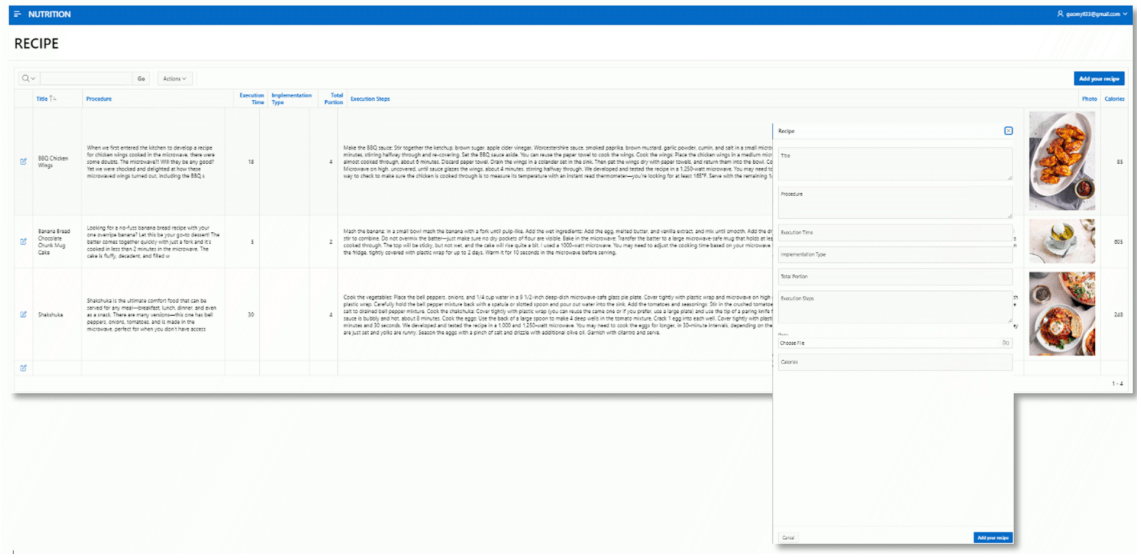
Rysunek 21.6: Utwórz nową aplikację.

Na rysunku 21.7 zostało utworzone nowe środowisko App Builder dla twojej nowej aplikacji. Aby uzyskać dostęp do swojej aplikacji, kliknij na nią i podaj swoje poświadczenia administratora, które powinny być takie same jak te dla obszaru roboczego. Po zalogowaniu możesz nawigować po swojej aplikacji za pomocą menu mega znajdującego się na górze.



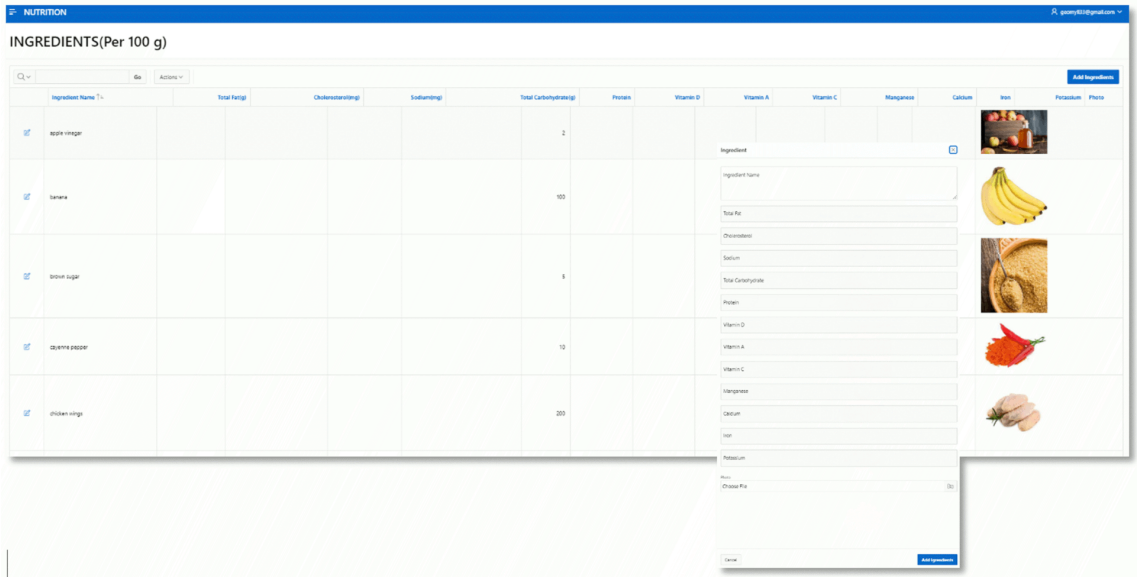
Rysunek 21.7: Zastosowanie żywieniowe.

Na rysunku 21.8 na stronie przepisu użytkownicy mogą uzyskać dostęp do wszystkich niezbędnych informacji dotyczących przygotowania przepisu, takich jak procedura, czas wykonania, całkowita liczba porcji, kroki wykonania oraz zdjęcie oczekiwanego rezultatu. W zależności od ich uprawnień użytkownicy mogą również dodać nowy przepis lub zaktualizować istniejący, aby dostosować go do swoich preferencji.



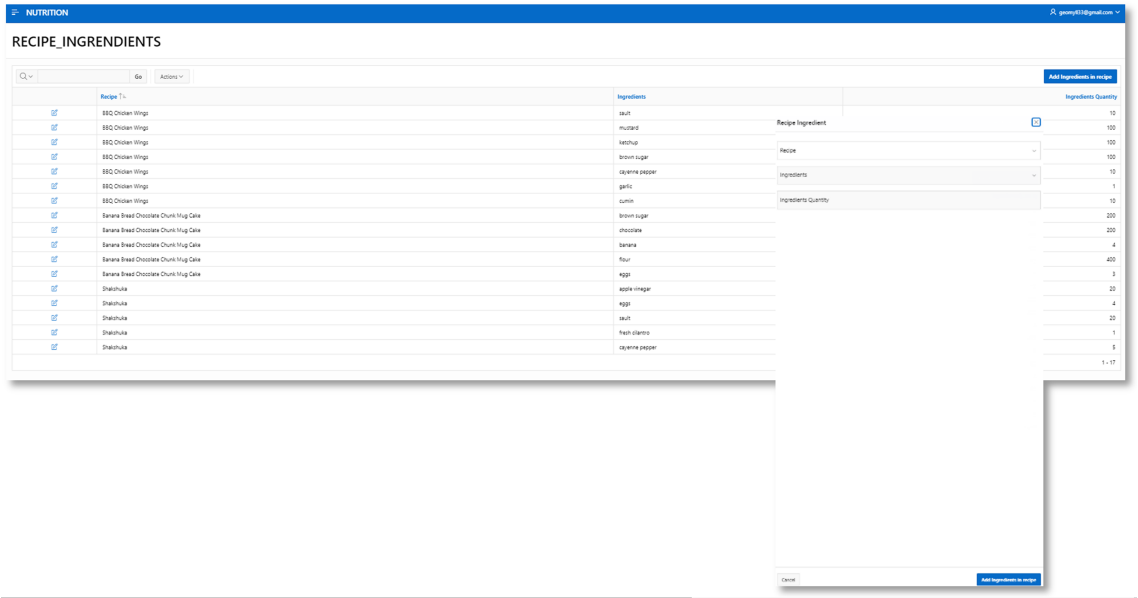
Rysunek 21.8: Aplikacja żywieniowa - strona przepisu.

Na rysunku 21.9 strona składników wyświetla różne atrybuty składnika, takie jak ogólna zawartość tłuszczu, cholesterolu, sodu, węglowodanów, białka itp. Te atrybuty są używane do obliczania wartości odżywczych przepisów. Użytkownicy posiadający odpowiednie uprawnienia mogą zarządzać stroną, dodając nowe składniki lub aktualizując istniejące, według potrzeb.



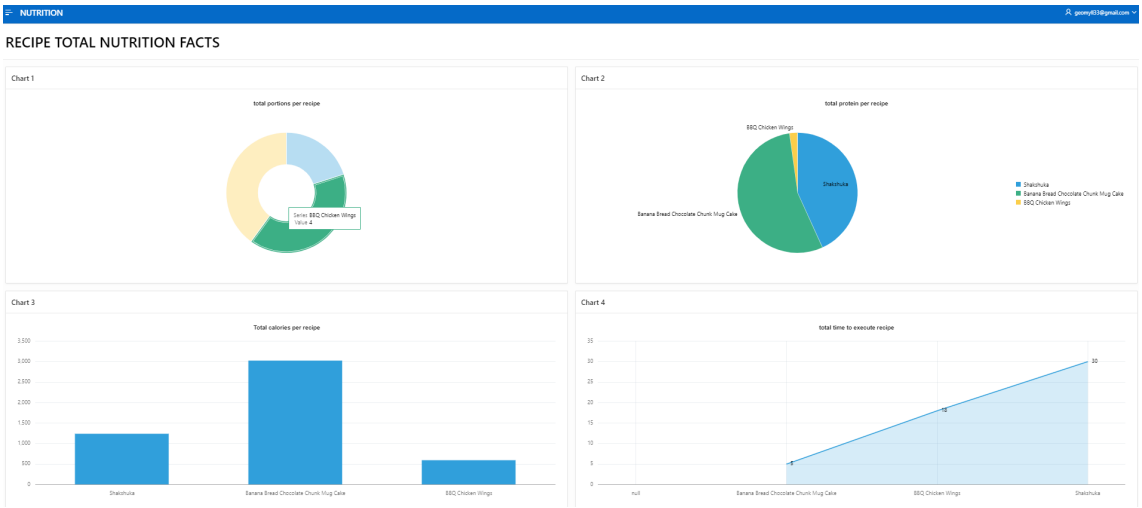
Rysunek 21.9: Strona aplikacji żywieniowych - składniki.

Na rysunku 21.10 strona składników przepisu umożliwia użytkownikom korzystanie z korzyści tej usługi internetowej, pozwalając im tworzyć przepisy według własnych potrzeb lub modyfikować istniejące, a następnie sprawdzać wartości odżywcze. Użytkownicy posiadający odpowiednie uprawnienia na tej stronie mogą dodawać lub usuwać składniki z przepisów.



Rysunek 21.10: Aplikacja żywieniowa - strona składników receptury.

Na rysunku 21.11 strona z raportem faktów dotyczących wartości odżywczych dostarcza użytkownikowi wszystkich niezbędnych informacji dotyczących stworzonego przez niego przepisu. Udostępnione są informacje takie jak całkowite białko w przepisie, całkowite kalorie przepisu, liczba porcji, które przepis zawiera, oraz czas potrzebny na przygotowanie przepisu.



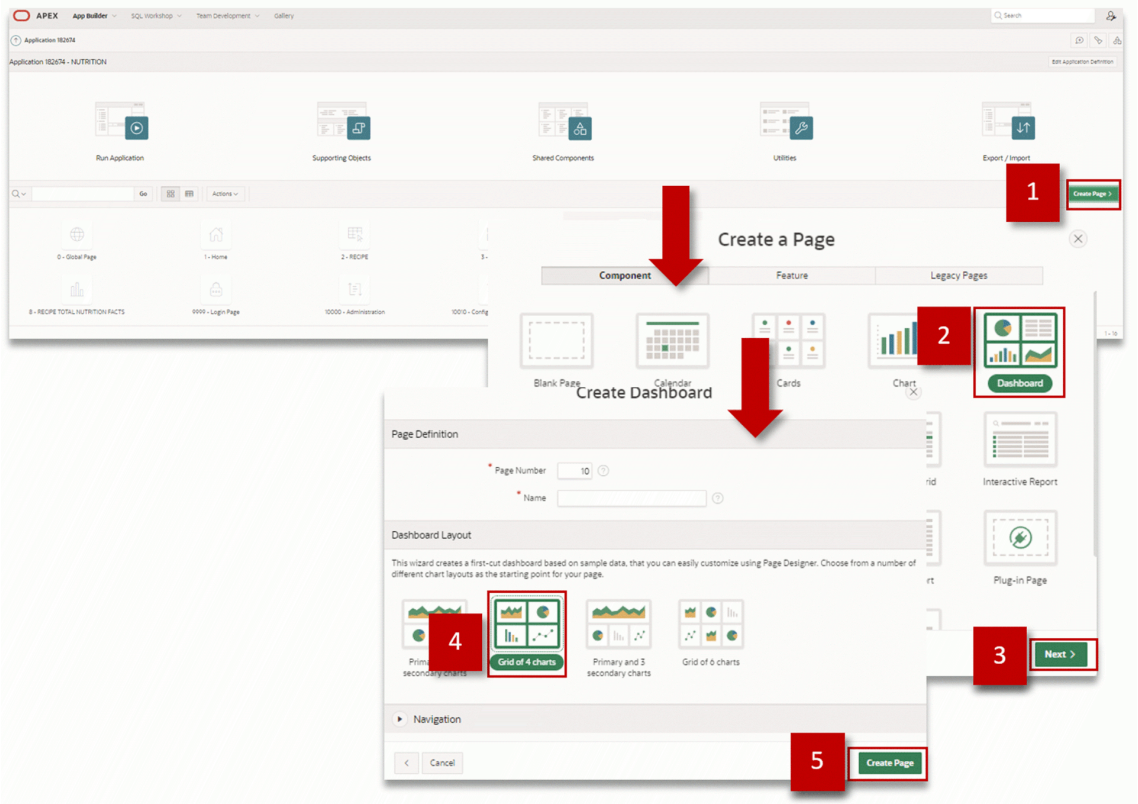
Rysunek 21.11: Aplikacja żywieniowa - strona z informacjami o żywieniu.

21.5.1 Jak zbudować pulpit nawigacyjny?

Aby utworzyć stronę pulpitu nawigacyjnego dla aplikacji, wykonaj kroki pokazane na rysunku 21.12.

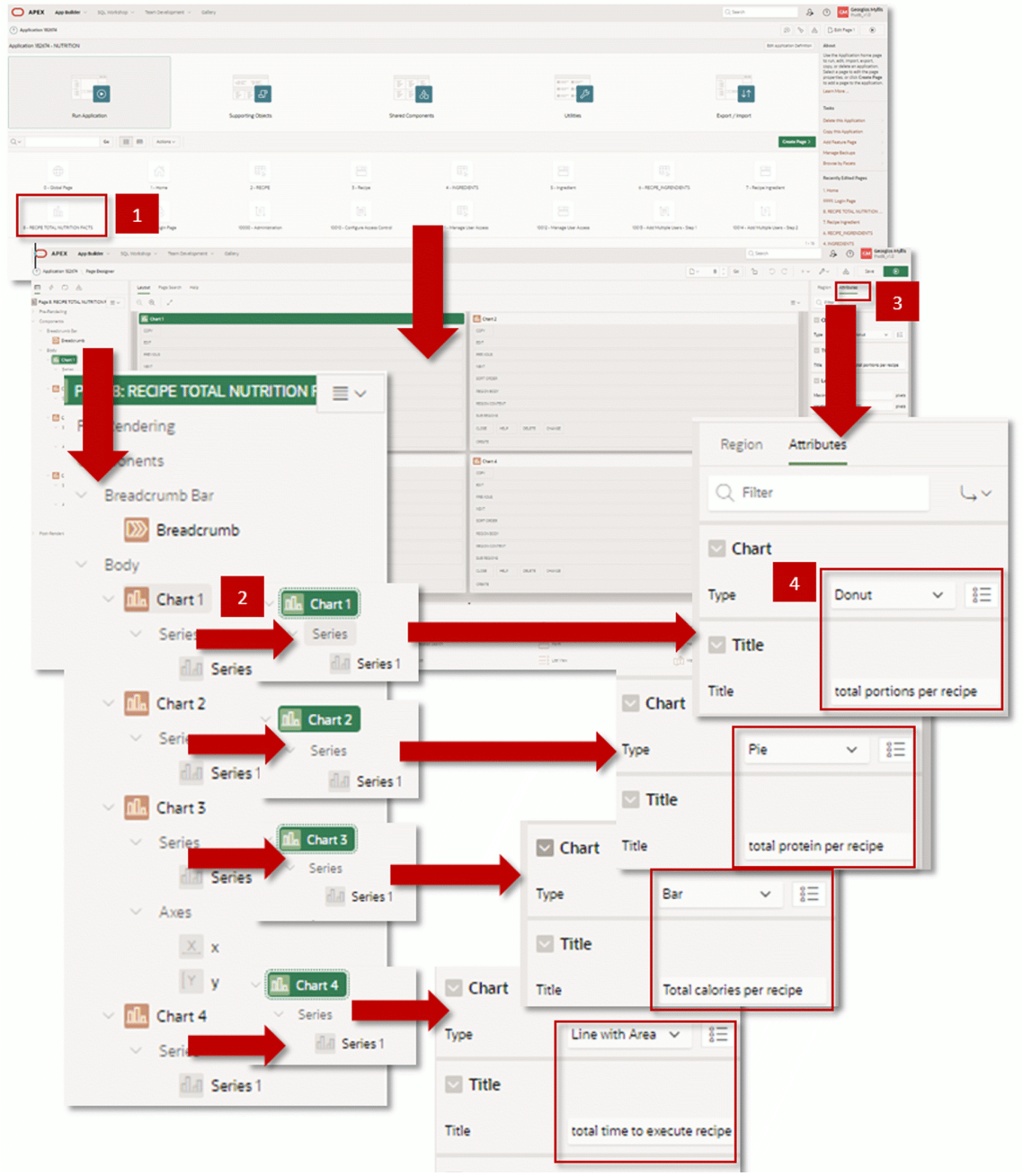
- Kliknij w swoje środowisko aplikacji.
- Kliknij przycisk **Create Page**.
- Wybierz **Dashboard** z listy dostępnych typów stron.
- Kliknij przycisk **Next**.
- Wybierz preferowany układ pulpitu nawigacyjnego.
- Kliknij przycisk **Create Page**, aby utworzyć stronę pulpitu nawigacyjnego.

Te kroki przeprowadzą Cię przez proces tworzenia strony pulpitu nawigacyjnego dla aplikacji. Po utworzeniu można dostosować pulpit nawigacyjny, dodając widżety i konfigurując układ w celu wyświetlania danych potrzebnych do monitorowania aplikacji i zarządzania nią.



Rysunek 21.12: Tworzenie strony pulpitu nawigacyjnego dla aplikacji.

Teraz przejdź do swojego raportu z panelami nawigacyjnymi, który znajdziesz w środowisku aplikacji, a następnie wejdź do projektanta stron. Jak pokazano na rysunku 21.12, wybraliśmy cztery wykresy i dodamy do każdego z nich różne dane za pomocą zapytań SQL. Aby zacząć, wybierz Wykres 1 i zmień jego nazwę na Ćałkowita ilość porcji na przepis ". Następnie wybierz **Attribute** i wybierz typ Wykresu Pierścieni. Następnie wybierz Wykres 2 i zmień jego nazwę na Ćałkowity białek na przepis", który wyświetli przepisy z największą ilością białka. Dla tego wykresu wybierz typ “ **Pie Chart**. Przechodząc do Wykresu 3, zmień jego nazwę na Ćałkowita liczba kalorii na przepis". Ten wykres będzie wyświetlał całkowitą liczbę kalorii dla każdego przepisu, a my użyjemy typu **Bar Chart type**. Na koniec zmień nazwę Wykresu 4 na Ćałkowity czas przygotowania przepisu". Wybierz **Attributes** , a następnie wybierz typ wykresu **Line with Area** (patrz rysunek 21.13).



Rysunek 21.13: Skonfiguruj tytuł i typ czatu w raportach pulpitu nawigacyjnego.

Następnie kliknij **Series 1** wykresu 1 i w "Sources" wprowadź następujące ustawienia:

```
SELECT DISTINCT RECIPE.TITLE,
RECIPE.TOTAL PORTION
FROM RECIPE
```

Na serii 1 wykresu 2 w obszarze Źródło umieść następujące ustawienia:

```
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
```

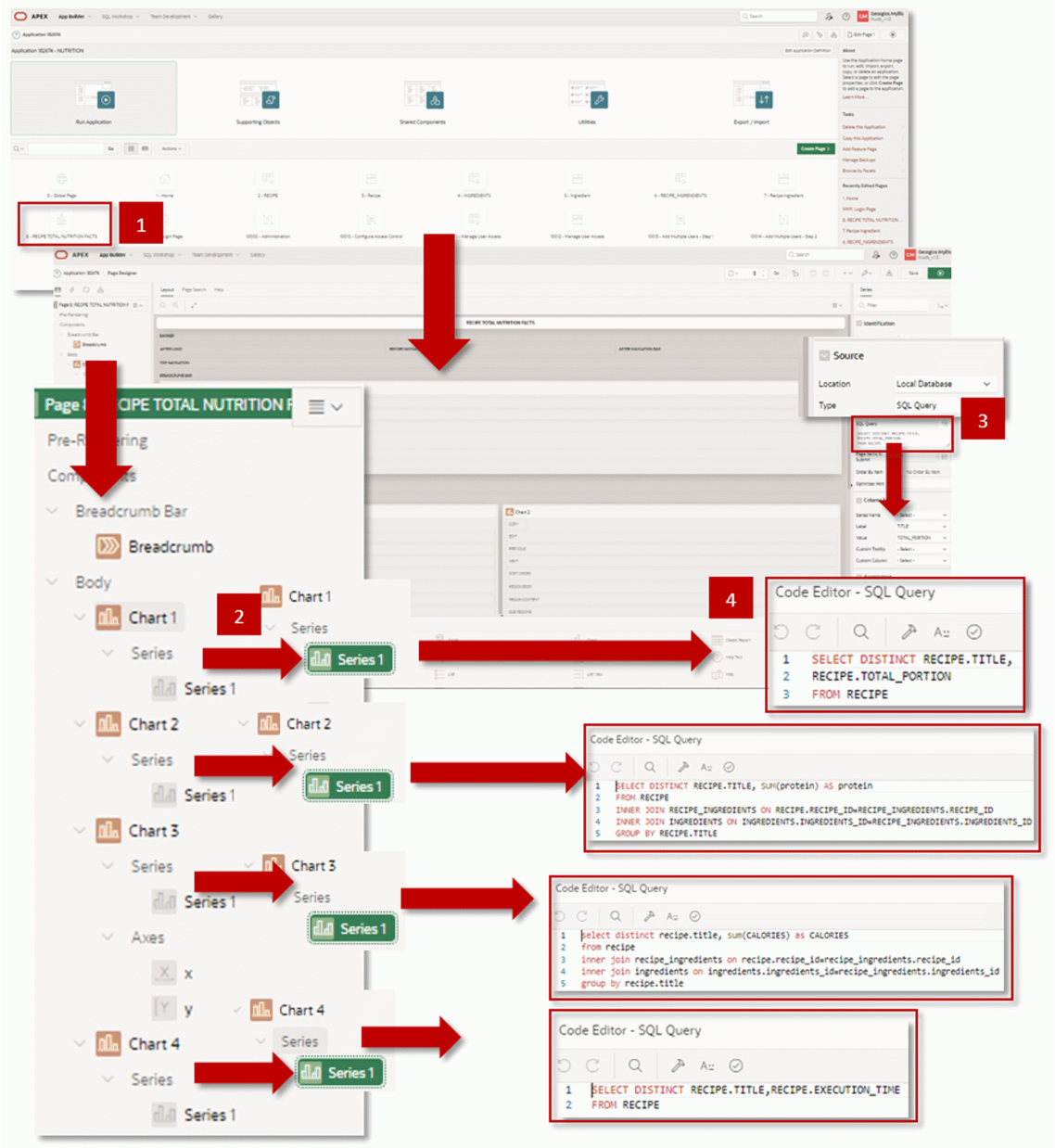
Na serii 1 wykresu 3 w obszarze Źródło umieść następujące ustawienia:

```
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
```

i na serii 1 wykresu 4 w obszarze Źródło umieść następujące ustawienia:

```
SELECT DISTINCT RECIPE.TITLE,RECIPE.EXECUTION_TIME
FROM RECIPE
```

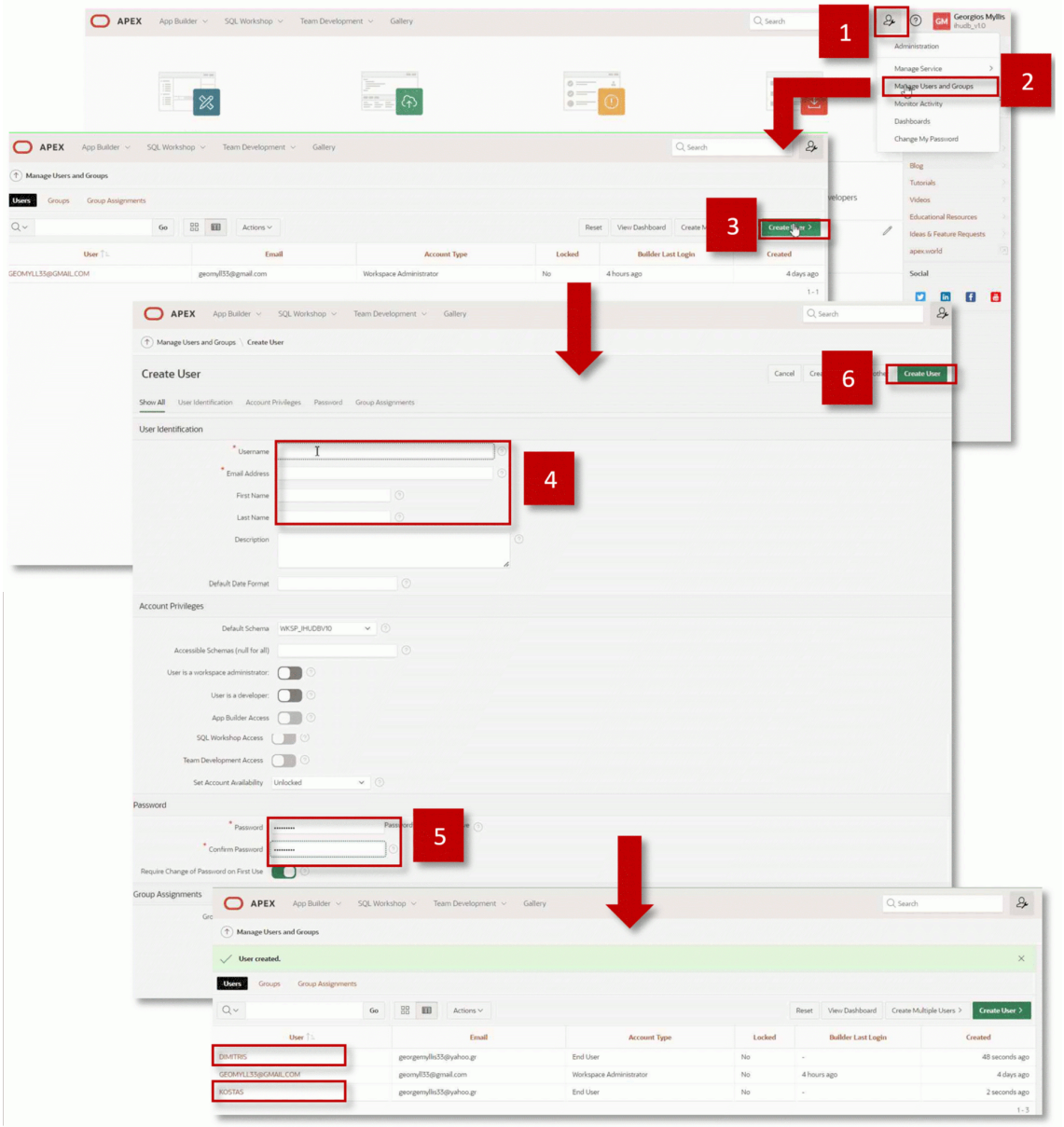
Rysunek 21.14 zawiera kilka wskazówek, jak go skonfigurować.



Rysunek 21.14: Konfigurowanie SOURCE za pomocą SQL QUERY.

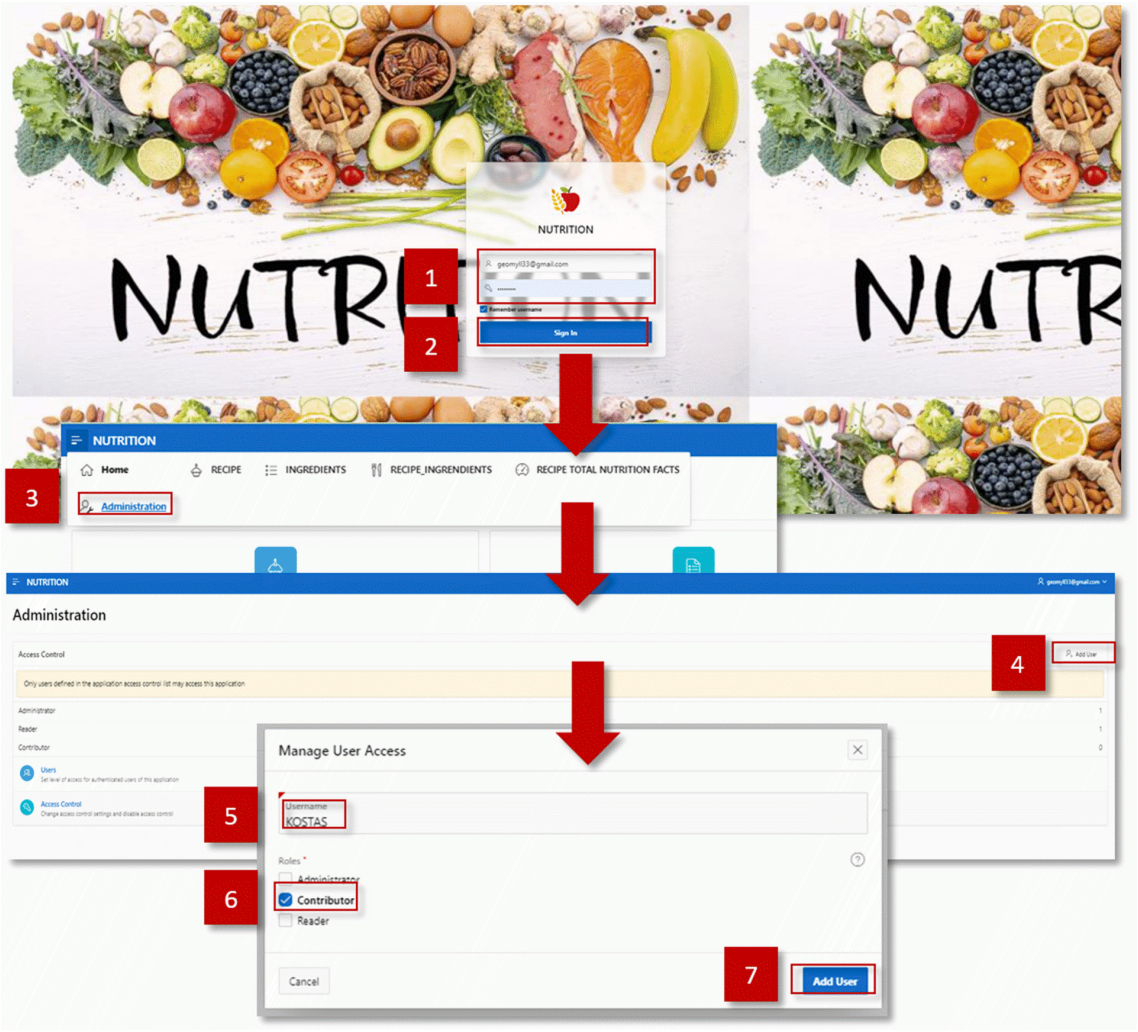
21.5.2 Programiści i użytkownicy aplikacji

W Oracle APEX deweloperzy i użytkownicy końcowi muszą mieć konta, jednak musimy odróżnić uprawnienia dla obszaru roboczego, aplikacji i konkretnej strony. Uprawnienia obszaru roboczego dotyczą administracji obszaru roboczego, rozwoju, dostępu do kreatora aplikacji, dostępu do warsztatu SQL i dostępu do rozwoju zespołu. Użytkownicy aplikacji mają wyłączone wszystkie te uprawnienia. Aby utworzyć nowe konto użytkownika, wybierz **Manage Users and Groups** z menu rozwijanego. Następnie wybierz **Create User**, wprowadź identyfikację i hasło użytkownika, zdefiniuj uprawnienia (administrator, deweloper, dostęp do narzędzi) i kliknij **Create User** , aby zakończyć proces (zob.rysunek 21.15).



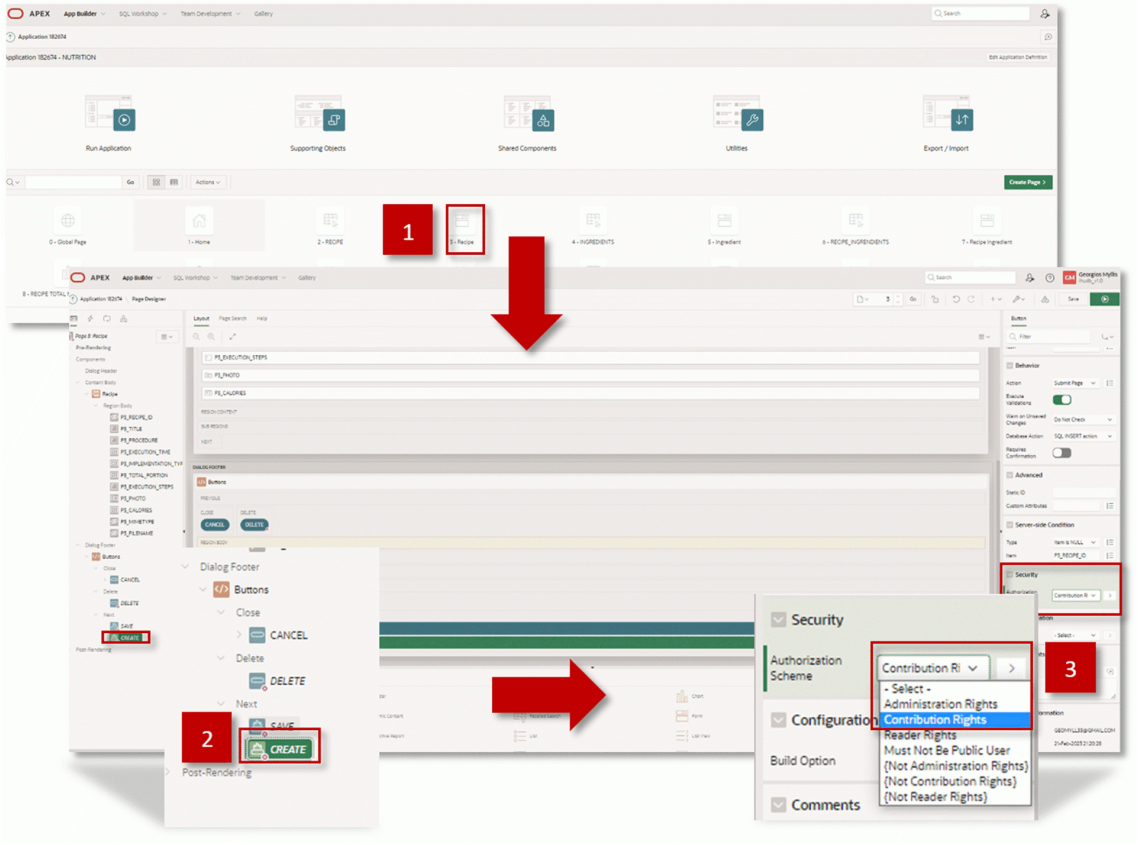
Rysunek 21.15: Tworzenie nowego konta użytkownika.

Użytkownicy aplikacji i ich uprawnienia aplikacyjne (autoryzacja) mogą być zdefiniowane (utworzone) na stronie administracyjnej aplikacji. Aby wygenerować tę stronę, zaznacz funkcję Kontroli dostępu wykonując generator aplikacji. Aby uzyskać dostęp do tej konkretnej aplikacji, przejdź do menu rozwijanego i kliknij pozycję **Administration**. Następnie kliknij **Add User**, wprowadź identyfikację użytkownika i rolę, a następnie kliknij **Add User**. Możesz odnieść się do rysunku 21.16 w celu uzyskania dodatkowych wskazówek.



Rysunek 21.16: Utwórz konto użytkownika w aplikacji.

Możesz przyznać uprawnienia do dostępu określonym użytkownikom lub roli. W projektancie stron (zob.rysunek 21.17 dla strony przepisów) znajdź opcje zabezpieczeń. Wybierz **Contribution Rights** (Prawa dotyczące współpracy) jako schemat autoryzacji. Kliknij przycisk **Save**. Spowoduje to, że tylko użytkownicy posiadający rolę współpracy będą mogli tworzyć przepisy.



Rysunek 21.17: Konfigurowanie stron z prawami użytkownika.

21.6 Dodatkowe materiały do nauki

Możesz znaleźć następujące dodatkowe materiały do nauki:

- eksportowana aplikacja
- przewodniki wideo

Wszystkie dodatkowe materiały są dostępne na publicznej stronie BeeAPEX . Zaloguj się jako użytkownik gość (nie jest wymagane hasło). Znajdź podręcznik w sekcji “Books”(Książki), skrypty w folderze Part 2 > Chapter21 w sekcji “Scripts” (Skrypty) i przewodniki wideo w “Collection of video guides”. Materiał na krótkie kursy w “Short courses section”.

21.6.1 Eksportowana aplikacja

Wyeksportowana aplikacja jest spakowana. Instalacja tworzy tabele, indeks, funkcję, procedurę i wyzwalacz, a także wypełnia dane. Deinstalacja usuwa wszystkie obiekty bazy danych używane w tej aplikacji.

21.6.2 Przewodniki wideo

Przewodnik wideo pokazuje każdy krok w tworzeniu aplikacji.

21.7 Pytania

1. Jak zmienić tło na stronie logowania?
2. Gdzie jeszcze możemy znaleźć zarządzanie rolami w środowisku Oracle APEX, jeśli nie jest ono włączone w naszej aplikacji?
3. Jak skonfigurować nazwę przycisku raportu?

4. Jak zmienić nazwę strony po jej utworzeniu?

21.8 Odpowiedzi

1. W swoim środowisku aplikacji wybierz **Shared Components**, a następnie wybierz **Static Application Files**, kliknij **Create File**, a w oknie pop-up dodaj preferowany obraz. Następnie skopiuj nazwę odwołania obrazu, wróć do swojego środowiska aplikacji i wybierz **9999-Login Page**, przejdź do pola css i wpisz wiersz:

```
body {
background-image:
url("#APP_FILES#\textit{the reference name of the image}");
background-size:1700px 900 px:
}
```

2. W środowisku aplikacji wybierz pozycję **Shared Components**, a następnie wybierz pozycję **Application Access control** .
3. Aby zmienić etykietę przycisku **Create** na inną nazwę, należy wprowadzić Projektanta strony raportu, zlokalizować przycisk **Create** i zmienić etykietę w polu Identyfikacja.
4. Aby zmienić nazwę utworzonej strony, zaloguj się do aplikacji i kliknij **Quick Edit**. Następnie kliknij tytuł strony i wybierz strzałkę w obszarze źródłowym bułki tartej. Następnie kliknij **Edit Component**, a w polu **Short Name** zaktualizuj nazwę do żądanej nowej nazwy. Na koniec kliknij **Apply Changes**, a nowa nazwa pojawi się w tytule strony.



22. Planowanie godzin pracy biura

J. MAŃKO, M. SOŃTA, R. LESKOVAR

22.1 Biznesowy pogląd na sprawę

Aplikacja rezerwacji godzin konsultacji na uniwersytecie z perspektywy biznesowej może być postrzegana jako narzędzie usprawniające i optymalizujące proces harmonogramowania dla personelu uniwersyteckiego. Może ona pomóc w poprawie wydajności harmonogramowania, zmniejszeniu obciążenia administracyjnego, zwiększeniu satysfakcji studentów oraz dostarczeniu informacji o dostępności w czasie rzeczywistym. Aplikacja może również generować dane i wskazówki dotyczące wzorców i wykorzystania spotkań, które mogą informować podejmowanie decyzji i alokację zasobów. Ponadto, aplikacja może oferować wygodną i dostępną platformę rezerwacji dla studentów, pomagając w stymulowaniu bardziej aktywnej i zaangażowanej społeczności studenckiej. Każdego tygodnia personel akademicki zobowiązany jest do udostępnienia 60 minut swojego czasu studentom, co wewnętrznie nazywane jest "Godzinami Konsultacyjnymi". Ponadto, zdarza się to dość często, zwłaszcza w okresie końca semestru, gdy zapytania od studentów nasilają się zarówno pod względem liczby, jak i zakresu. W takim przypadku 60 minut tygodniowo to za mało, ponieważ studenci są zdeterminowani, aby jak najszybciej uzyskać pożądany wynik od pracownika naukowego. Mimo że istnieją pewne standardowe tematy do dyskusji, takie jak: praca dyplomowa, egzaminy, prace końcowe, niektóre z przypadków dotyczą nieistotnych tematów, które być może powinny być omawiane, ale niekoniecznie w godzinach konsultacyjnych z nauczycielem (na przykład: wnioski administracyjne lub problemy, które można łatwo załatwić bez interwencji nauczyciela akademickiego). W ostatnich latach, zwłaszcza z powodu pandemii COVID-19, uniwersytety przeszły radykalną transformację technologiczną umożliwiającą różne formy zdalnego nauczania za pomocą dostępnych platform i komunikatorów, które ułatwiały proces nauczania online. W rzeczywistości praktycznie wszystkie działania akademickie zostały przeniesione online, w tym oczywiście zdalne godziny konsultacyjne. Mimo że istnieje porozumienie wśród społeczności akademickiej, że zdalne nauczanie nigdy nie zastąpi w pełni nauczania stacjonarnego, zdalne nauczanie i praca nadal są wdrażane w pewnym stopniu na całym świecie. Co więcej, niektóre działania i procesy uniwersyteckie już przeszły lub będą w przyszłości przechodzić do trybu online, na przykład w celu zapewnienia efektywnego obiegu dokumentów bez użycia papieru. Te okoliczności jeszcze bardziej umacniają pilną potrzebę biznesową wiarygodnego narzędzia cyfrowego, które umożliwi rezerwację godzin konsultacyjnych online, które mogą odbywać się zarówno online, jak i offline. Wyraźnie, cała społeczność akademicka mogłaby odnieść wielką korzyść z takiej aplikacji, która wkrótce powinna stać się codziennym narzędziem dla wszystkich w społeczności

akademickiej do użytku, tak jak skrzynka e-mail, a nie jakiś odległy cyfrowy fanaberia.

22.2 Definicja problemu

Na chwilę obecną niemożliwe jest rezerwowanie (zapisywanie) slotu czasowego i planowanie treści spotkania z wyprzedzeniem. Faktycznie studenci, jeśli się pojawią w jakiejkolwiek liczbie, są obsługiwani na zasadzie "kto pierwszy, ten lepszy". Nie tylko zwiększa to frustrację studentów, którzy muszą czekać w kolejce bez żadnej pewności, czy ich spotkanie w ogóle się odbędzie, ale także denerwuje personel akademicki, który musi wybierać między tym, czy pozwolić niektórym studentom czekać na próżno, czy przekształcić swoje plany na dzień. W rezultacie dochodzi do sytuacji, w których personel akademicki faktycznie udziela bezpłatnej porady dla studentów, a nawet kosztem swojego życia prywatnego, tylko po to, aby uniknąć bycia postrzeganym jako niedostępny lub nieprzyjazny przez studentów. Ponadto, ten status quo powoduje niewydajne alokowanie czasu zarówno dla studentów, jak i personelu akademickiego, a także niepotrzebne obciążenie harmonogramem dla tego ostatniego. Ta administracyjna nieskuteczność systemowa zmusza personel akademicki do angażowania się w dodatkowe, często niepłatne obowiązki, jednocześnie pozostawiając studentów w niepewności co do wyników i prawdopodobnie ledwo zadowolonych z całego procesu. Jest to najważniejszy przykład przenoszenia konsekwencji niesprawnego systemu umawiania spotkań na grupy zlokalizowane niżej w hierarchii akademickiej (studentów i personel akademicki), zamiast znalezienia rozwiązania z góry inicjowanego przez władze uniwersyteckie. Dlatego technologia oferuje tutaj wiarygodne, upoważniające i wykonalne rozwiązanie, które w znacznym stopniu złagodzi obciążenie pracy personelu akademickiego, jednocześnie zapewniając wyższą satysfakcję i efektywność użytkowania dla studentów. Studenci często zgłaszają się do biura studenckiego z pytaniami dotyczącymi dat i miejsc godzin konsultacyjnych. Ponieważ nie ma wiarygodnego źródła, polegają na doświadczeniach i często kierują studentów w niewłaściwe miejsca o niewłaściwych porach. Zgodnie z zasadami instytucji akademickiej każda zmiana godzin konsultacyjnych powinna być zatwierdzona przez prodziekana. Zatwierdzenia są udzielane w bezpośredniej komunikacji między nauczycielem a prodziekanem, a biuro studenckie nie ma o tym informacji. Prodziekan chciałby, aby biuro studenckie wprowadzało przełożone godziny konsultacyjne. Zarząd ma niewiele danych na temat ogólnego obrazu godzin konsultacyjnych w instytucji akademickiej. Plotki i inne niewiarygodne informacje nie stanowią dobrych podstaw do racjonalnych decyzji dotyczących zwiększenia lub zmniejszenia liczby godzin dyżurów. Może to również być związane z nagradzaniem nauczycieli za ich dodatkowe wysiłki.

22.3 Przypadki użycia

Aplikacja rezerwacji godzin konsultacyjnych umożliwia dostęp zarządzaniu, personelowi biura studenckiego, studentom oraz personelowi akademickiemu (nazywanemu także nauczycielami), z różnymi uprawnieniami:

- Zarząd otrzymuje podsumowanie informacji zebranych na panelu zarządzania (liczba godzin konsultacyjnych według roku i miesiąca, 10 nauczycieli z największą liczbą studentów, 5 najbardziej przeciążonych nauczycieli i 5 nauczycieli bez studentów w czasie godzin konsultacyjnych).
- Personel biura studenckiego ma dostęp do interaktywnego raportu z godzinami konsultacyjnymi i możliwość przełożenia dowolnych godzin konsultacyjnych.
- Studenci mogą uzyskać dostęp do kalendarza swoich spotkań, kalendarza wszystkich godzin konsultacyjnych oferowanych przez nauczycieli zaangażowanych w program studiów studenta, zapisywanie się przez kalendarz oraz szczegółowy interaktywny raport dotyczący zapisów na godziny konsultacyjne w biurze studenckim.
- Personel akademicki/nauczyciele mogą uzyskać szczegółowy interaktywny raport na temat godzin konsultacyjnych nauczycieli (godzin obowiązkowych), kalendarz z godzinami konsul-

tacyjnymi nauczycieli (godzin obowiązkowych) oraz mogą przełożyć godziny konsultacyjne, na które nie zapisali się studenci.
- Zapisywanie się studentów na godziny dyżuru obejmuje przegląd kalendarza (wszystkie godziny dyżuru oferowane przez wszystkich nauczycieli zaangażowanych w program studiów studenta), wybór pożądanych godzin dyżuru, wpisanie celu i potwierdzenie zapisu. Nie wolno zapisać się więcej niż raz na zaplanowaną godzinę dyżuru.

22.3.1 Opis narracyjny

Dla zwięzłości przedstawimy tylko dwa przypadki użycia spośród kilku:

- Zmiana harmonogramu godzin pracy przez nauczyciela, gdy żaden uczeń nie jest zapisany, wymaga od nauczyciela zalogowania się, przejrzenia zaplanowanych godzin pracy, w których nikt nie jest zapisany, i "przeniesienia"jednej po drugiej na inne daty.
- Zapisanie się studenta na godziny pracy polega na przejrzeniu kalendarza (wszystkie godziny pracy oferowane przez wszystkich nauczycieli zaangażowanych w program studiów studenta), wybraniu żądanych godzin pracy, wpisaniu celu i potwierdzeniu zapisu. Nie można zapisać się więcej niż raz na zaplanowane godziny zajęć.

22.3.2 Opis częściowo ustrukturyzowany

Pierwszy przypadek użycia to zmiana terminu godzin dyżuru przez nauczyciela, gdy nikt się na nie nie zapisuje (patrz Tabela 22.1). Ograniczamy przekształcenie w tym przypadku, ponieważ żaden student nie jest dotknięty. Drugi przypadek użycia dotyczy zapisów studentów na godziny dyżuru (patrz Tabela 22.2). Aplikacja powinna również zapobiegać wielokrotnym zapisom na ten sam slot godzin dyżuru.

22.3.3 Schemat przypadków użycia

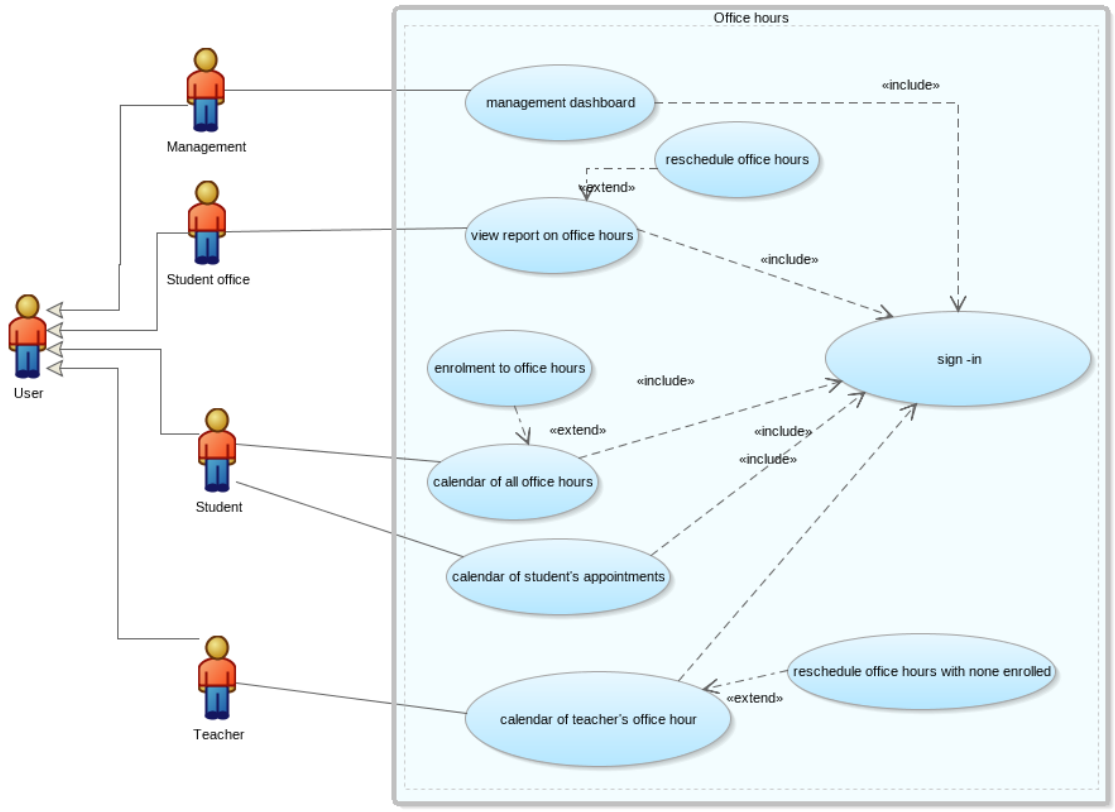
Rysunek 22.1 przedstawia zidentyfikowane przypadki użycia. Zauważ, że przypadku użycia „zaloguj się” wykonuje się tylko raz w sesji użytkownika i że żaden z pozostałych przypadków użycia nie może zostać wykonany przed pomyślnym zakończeniem „zaloguj się”. Trzy przypadki użycia, a mianowicie „zmiana godzin dyżuru”, „zapisy na godziny dyżuru” i „zmiana godzin dyżuru bez zapisanych”, są zależne i oznaczone związkiem „rozszerzenie”. Te przypadki użycia są wykonywane na żądanie po zakończeniu przypadku nadrzędnego.

Tablica 22.1: Opis przypadku użycia: zmiana harmonogramu godzin pracy przez nauczyciela.

Słowo kluczowe	**Wartość**
ID:	*Ch22-01*
Tytuł:	*Zmiana harmonogramu godzin pracy przez nauczyciela*
Opis:	*Nauczyciel używa aplikacji APEX, aby przełożyć godziny pracy, w których nie ma zapisanego ucznia.*
Główny aktor:	*nauczyciel*
Warunki wstępne:	dostęp do przeglądarki internetowej na urządzeniu mobilnym lub komputerze, użytkownik posiada poświadczenia i uprawnienia nauczyciela, dostępna jest strona internetowa aplikacji
Warunki końcowe:	Dane przechowywane w tabeli bazy danych ch22_duty_hours
Główny	-
Scenariusz sukcesu:	1. Nauczyciel otwiera przeglądarkę internetową i loguje się do aplikacji. 2. Wybiera przekładanie godzin dyżuru bez zapisanych studentów. 3. Klika wybrany slot oznaczony jako bez zapisu. 4. W formularzu wpisuje dane dotyczące nowych ustaleń (miejsce, czas rozpoczęcia i zakończenia). 5. Przegląda i potwierdza przekładanie, klikając odpowiedni przycisk.
Rozszerzenia:	-
Częstotliwość stosowania:	*Około 4 z 30 obowiązkowych terminów w jednym roku akademickim.*
Status:	[gotowe]
Właściciel:	*Użytkownik z uprawnieniami nauczyciela*
Priorytet:	*umiarkowanye*

Tablica 22.2: Opis przypadku użycia: rejestracja studentów na godziny pracy.

Słowo kluczowe	**Wartość**
ID:	*Ch22-02*
Tytuł:	*Rejestracja studentów na godziny pracy biura*
Opis:	*Student korzysta z aplikacji APEX, aby zapisać się na godziny dyżuru. Kalendarz zawiera wszystkie zaplanowane godziny dyżuru dla wszystkich nauczycieli zaangażowanych w programy studiów studentów (wydziały). Zazwyczaj jeden student jest zapisany na jeden program studiów, ale wyjątkowi studenci mogą być uprawnieni do nauki dwóch lub więcej. Aplikacja zapobiega wielokrotnym zapisom jednego studenta na konkretne godziny dyżuru.*
Główny aktor:	*student*
Warunki wstępne:	Dostęp do przeglądarki internetowej na urządzeniu mobilnym lub komputerze, użytkownik ma poświadczenia nauczycieli i odpowiednie uprawnienia, witryna aplikacji jest dostępna.
Warunki końcowe:	dane przechowywane w bazie danych w tabeli ch22_dh_slot
Główny	-
Scenariusz sukcesu:	1. Student otwiera przeglądarkę internetową i loguje się do aplikacji 2. Klika żądane gniazdo (gniazda są kolorowe i wskazują statusy) 3. click the desired slot (slots are colored indicating the statuses) 4. W nowym formularzu uczeń klika przycisk i potwierdza rejestrację
Rozszerzenia:	-
Częstotliwość stosowania:	*Ponad 1000 studentów korzysta z godzin pracy średnio 2 razy w semestrze dla 10 do 15 nauczycieli.*
Status:	[gotowe]
Właściciel:	*Student*
Priorytet:	*wysoki*



Rysunek 22.1: Diagram przypadków użycia.

22.4 Model danych

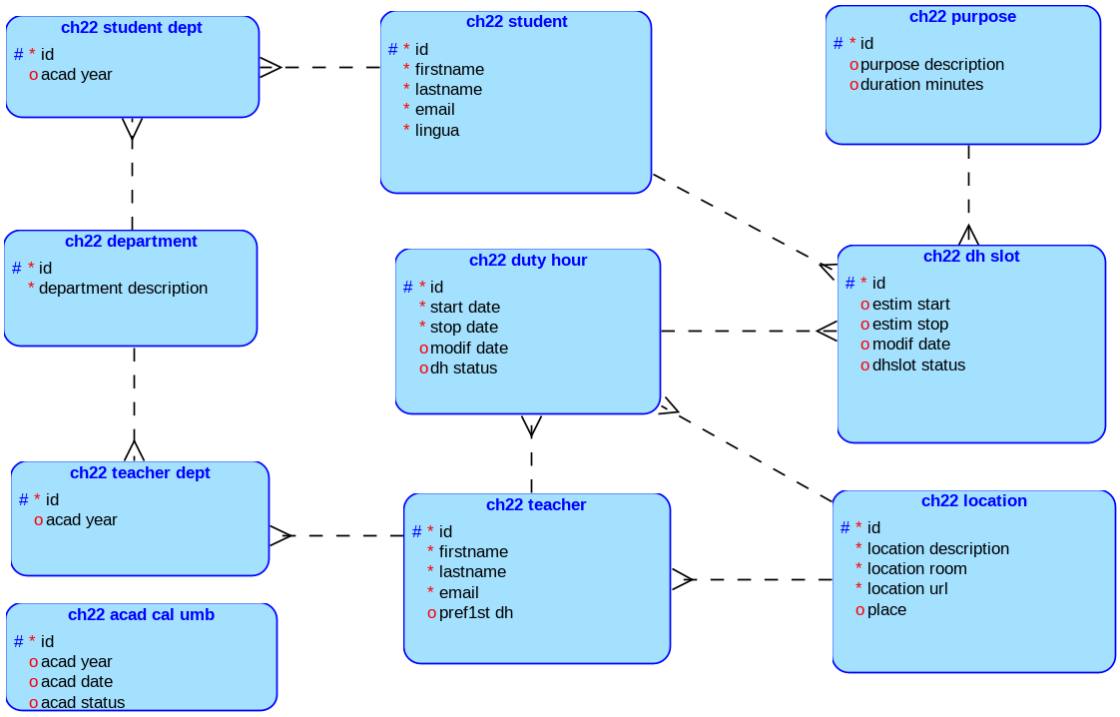
22.4.1 Opis modelu danych

W modelu danych logicznych znajduje się dziesięć jednostek:

- **ch22_dh_slot** prezentuje spotkanie jednego studenta z nauczycielem;
- **ch22_duty_hour** prezentuje uprzednio przydzielony przedział czasu nauczyciela (godziny konsultacyjne), który wskazuje na kilka slotów, zwanych spotkaniami student-nauczyciel;
- **ch22_purpose** przechowuje powody i średni czas trwania spotkań student-nauczyciel;
- **ch22_student_dept** wskazuje na jeden program studiów jednego studenta. Wyjątkowi studenci mogą być zapisani na więcej niż jeden program w jednym roku akademickim;
- **ch22_student** to osoba studiująca na uczelni i uczestnicząca w programie studiów;
- **ch22_teacher_dept** wskazuje na jeden program studiów (inaczej wydział) jednego nauczyciela. Zwykle jeden nauczyciel jest zaangażowany w więcej niż jeden program studiów;
- **ch22_teacher** to osoba prowadząca działalność naukową (lekcje, ćwiczenia laboratoryjne itp.);
- **ch22_location** jest albo fizyczna (R), albo wirtualna (I), albo obie (B) - jest to miejsce, w którym odbywają się godziny konsultacyjne;
- **ch22_department** to synonim programu studiów;
- **ch22_acad_cal_umb** przechowuje określone dni robocze/nie robocze. Ta jednostka jest powiązana z wszystkimi jednostkami posiadającymi pewne atrybuty daty, jednak nie będziemy modelować tych relacji w sposób explicite, ponieważ relacja domyślna wystarczy dla kompletności modelu.

22.4.2 Logiczny model danych

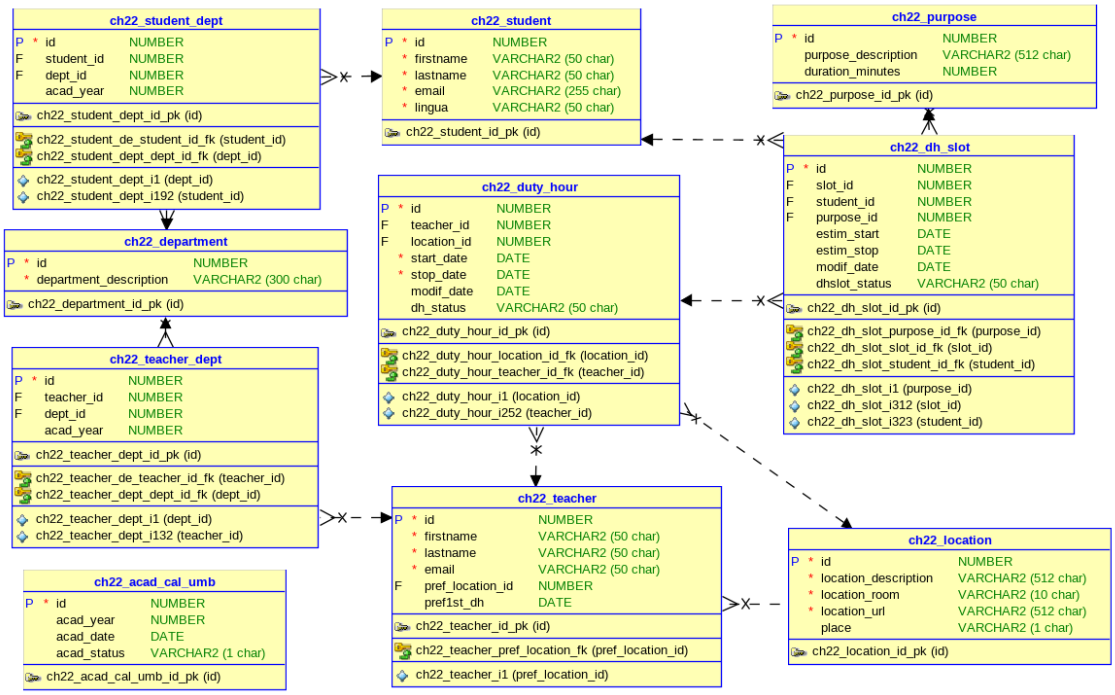
Logiczny model danych przedstawiono na rysunku 22.2.



Rysunek 22.2: Logiczny model danych.

22.4.3 Relacyjny model danych

Automatyczna transformacja z modelu danych logicznych do modelu danych relacyjnych w Oracle SQL Data Modeler jest realizowana przez funkcję *Engineering to relational*. Wynikowy model danych relacyjnych gotowy do wyeksportowania jako skrypt SQL został przedstawiony na Rysunku 22.3.



Rysunek 22.3: Relacyjny model danych.

Relacyjny model danych można opracować bez SQL Data Developer Data Modeler. Możemy użyć Quick SQL w APEX.

22.4.4 Szybki SQL

Przedstawiony Quick SQL (patrz Rysunek 22.4) zawiera wszystkie niezbędne informacje do wygenerowania skryptu SQL.

```
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
```

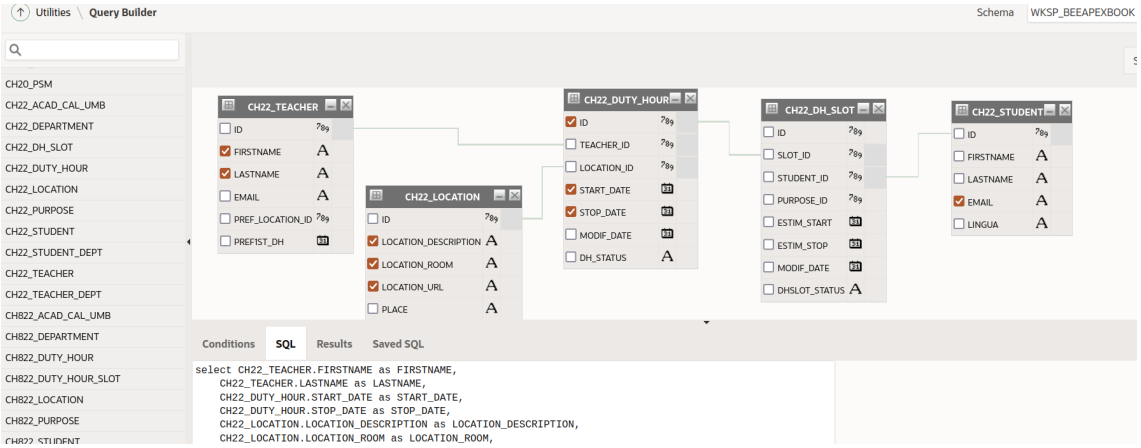
Rysunek 22.4: Model danych opisany za pomocą Quick SQL.

22.4.5 Skrypt SQL do tworzenia tabel

Skrypt SQL do tworzenia tabel jest dostępny w materiałach edukacyjnych jako plik CH22CREATE.sql. Skrypt do tworzenia tabel jest także dostępny w spakowanej aplikacji w materiałach edukacyjnych.

22.4.6 Konstruktor zapytań w APEX

Ta konkretna aplikacja będzie korzystać z dość długich zapytań, które mogą przestraszyć początkujących użytkowników APEX. Ale jest na to lekarstwo - Query Builder, który jest wbudowany w APEX (SQL Workshop > Utilities > Query Builder). Może on pomóc każdemu w konstruowaniu poprawnych zapytań, wystarczy kliknąć i przeciągnąć. Figure 22.5 przedstawia przykładowe zapytanie dla kalendarza zapisów studentów.



Rysunek 22.5: Przykładowe użycie konstruktora zapytań.

Wygenerowana instrukcja SQL ma następującą postać:

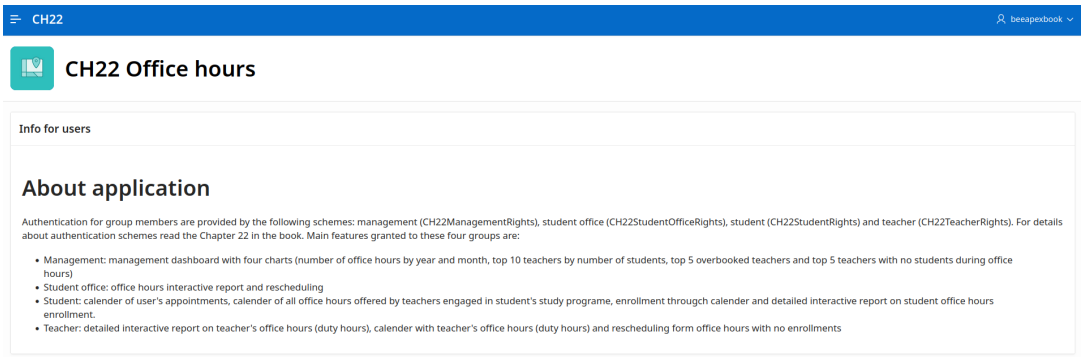
```
select CH22_TEACHER.FIRSTNAME as FIRSTNAME,
       CH22_TEACHER.LASTNAME as LASTNAME,
       CH22_DUTY_HOUR.START_DATE as START_DATE,
       CH22_DUTY_HOUR.STOP_DATE as STOP_DATE,
       CH22_LOCATION.LOCATION_DESCRIPTION as
              LOCATION_DESCRIPTION,
       CH22_LOCATION.LOCATION_ROOM as 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
```

Dodając dodatkowy warunek na końcu ąnd CH22_TEACHER.EMAIL = :APP_USER", zapytanie zwróci zapisanych studentów dla tego konkretnego nauczyciela. Kreator zapytań byłby potrzebną funkcją dla absolutnych początkujących, ale doświadczeni użytkownicy mogą również skorzystać

z niego. Zapytania używane w tej aplikacji wykorzystują operator konkatenacji (||), funkcję EXTRACT, wspólne wyrażenia tabeli, pobieranie tylko pierwszych 10 lub 5 wierszy, a także funkcję niestandardową CH22_dh_utilization. Czytelnicy tej książki mogą skorzystać, instalując najpierw spakowaną aplikację i przyjrzeć się zapytaniom używanym w kalendarzach, panelu sterowania i raportach.

22.5 Interfejsy aplikacji

Strona główna aplikacji zawiera podstawowe informacje o aplikacji (zob. Figure 22.6).



Rysunek 22.6: Strona główna aplikacji.

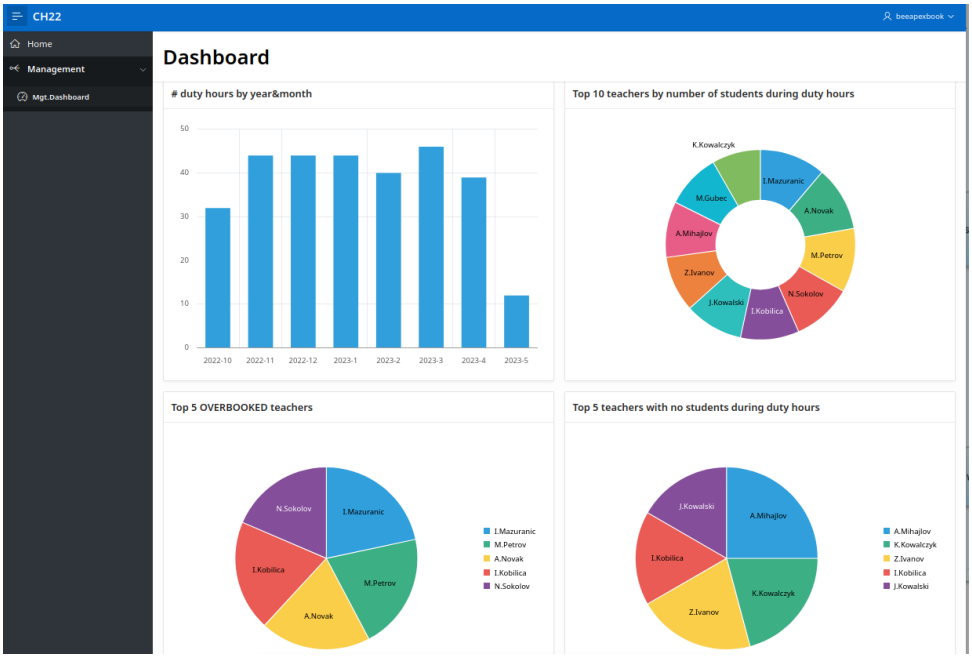
Zarówno zarządzanie, personel biura studenckiego, studenci, jak i nauczyciele będą obsługiwani wyłącznie za pomocą spersonalizowanych interfejsów.

22.5.1 Interfejsy aplikacji do zarządzania

Użytkownicy z uprawnieniami do zarządzania mogą uzyskać dostęp do pulpitu nawigacyjnego zarządzania (patrz Figure 22.7). Zawiera on wykresy wspierające podejmowanie decyzji związane z godzinami pracy i nauczycielami:

- Liczba godzin pracy w poszczególnych latach i miesiącach
- 10 najlepszych nauczycieli według liczby uczniów
- 5 najlepszych nauczycieli overbookingu i
- 5 najlepszych nauczycieli bez uczniów w godzinach pracy

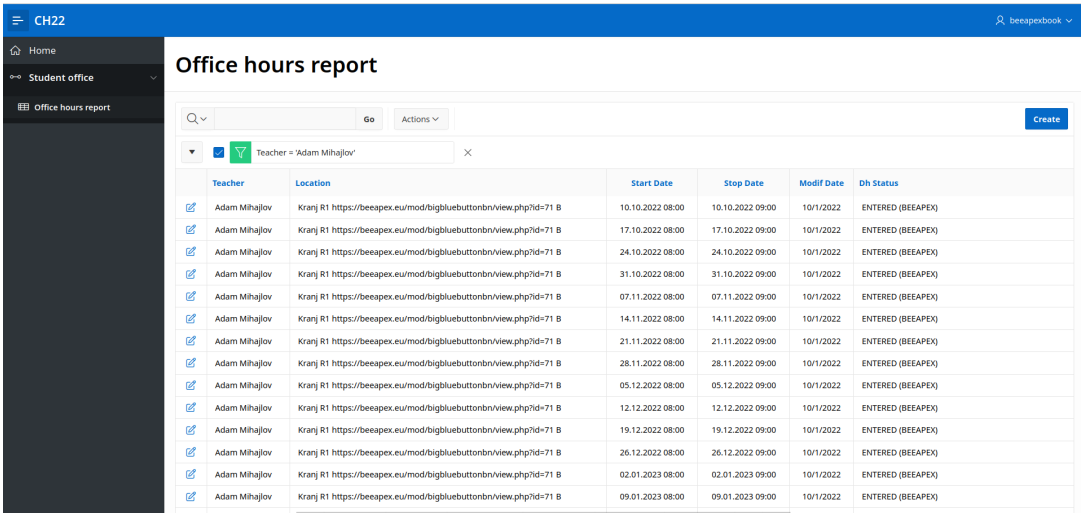
W chwili obecnej kierownictwo nie definiuje żadnych dalszych wymagań dotyczących zarządzania.



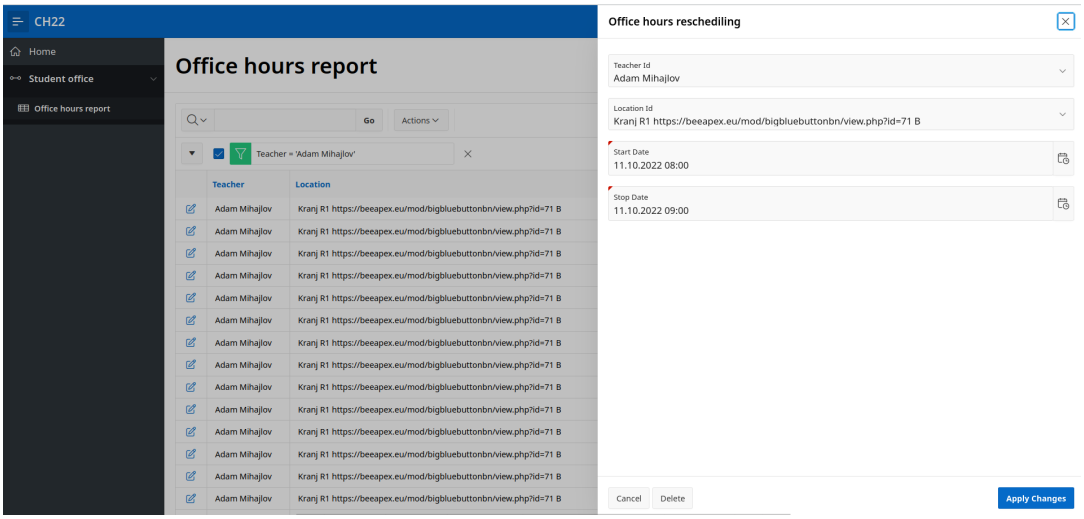
Rysunek 22.7: Pulpit nawigacyjny zarządzania.

22.5.2 Interfejsy aplikacji biur studenckich

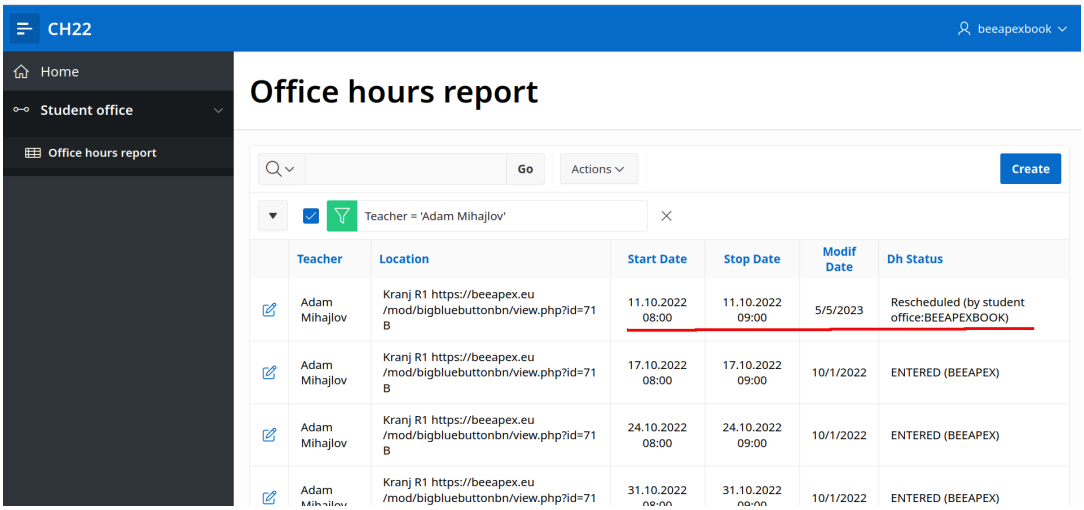
Użytkownicy posiadający prawa biura studenckiego mogą uzyskać dostęp do interaktywnego raportu godzin dyżurów i zmieniać harmonogram każdych godzin dyżurowych. Na rysunku Figure 22.8 przedstawiono interaktywny raport z aktywowanym filtrem (wyświetlany jest tylko jeden nauczyciel). Następnie, na rysunku Figure 22.9 pokazano formularz umożliwiający dokonywanie zmian w harmonogramie. Użytkownik może zmienić nauczyciela, lokalizację oraz daty rozpoczęcia i zakończenia wybranych godzin dyżurowych. Na rysunku Figure 22.10 przedstawiono raport po zmianach w harmonogramie. Dwa pola są automatycznie aktualizowane: data modyfikacji i status godzin dyżurowych. Ten ostatni to sklejony ciąg znaków zawierający również nazwę użytkownika, który dokonał zmiany.



Rysunek 22.8: Interaktywny raport godzin pracy dla biura studenta.



Rysunek 22.9: Formularz zmiany harmonogramu dla biura studenta.

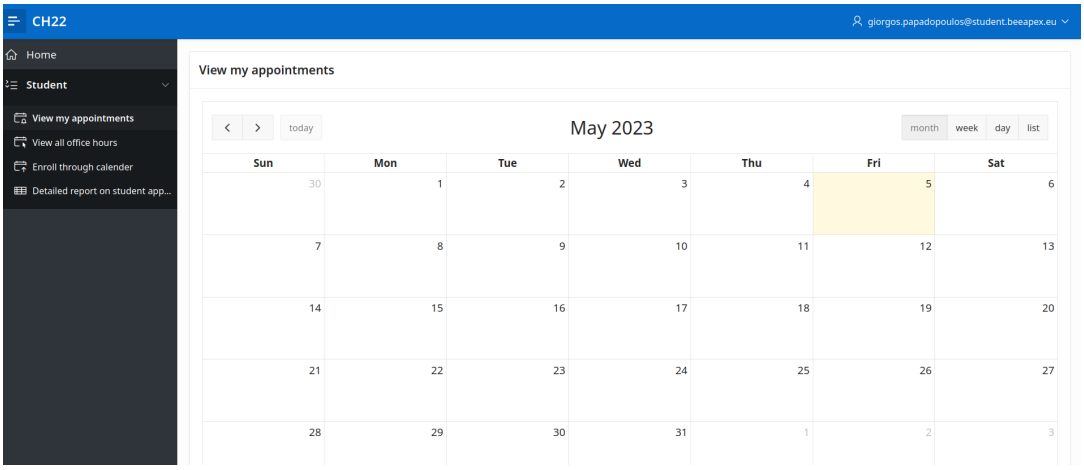


Rysunek 22.10: Interaktywny raport godzin pracy po zmianie harmonogramu.

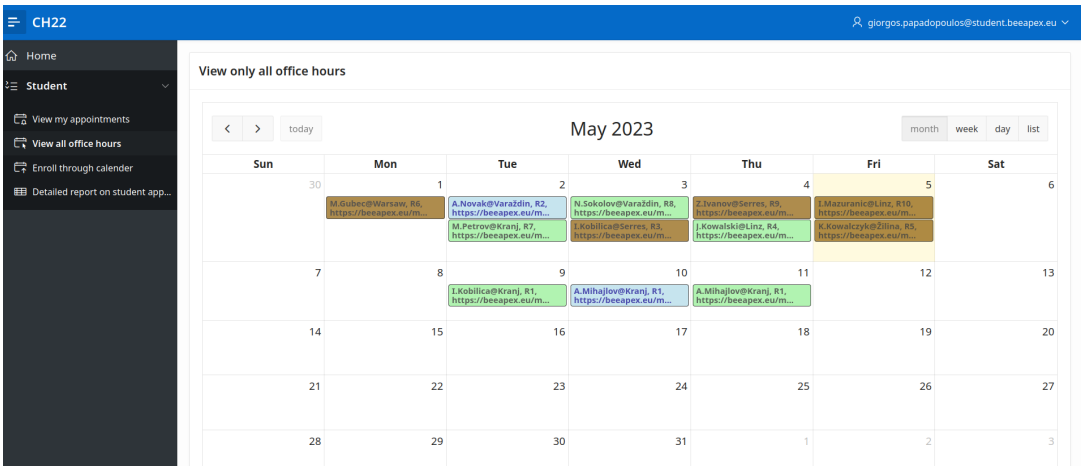
22.5.3 Interfejsy aplikacji studenckich

Elementy kalendarza są kolorowane w zależności od wykorzystania. BRAK ZAPISU jest oznaczony kolorem zielonym, DOSTĘPNY kolorem niebieskim, a PRZEŁADOWANY kolorem brązowym. W chwili obecnej student może zapisać się nawet jeśli ustalony zostanie status PRZEŁADOWANY. Użytkownicy posiadający prawa studenckie mają dostęp do:

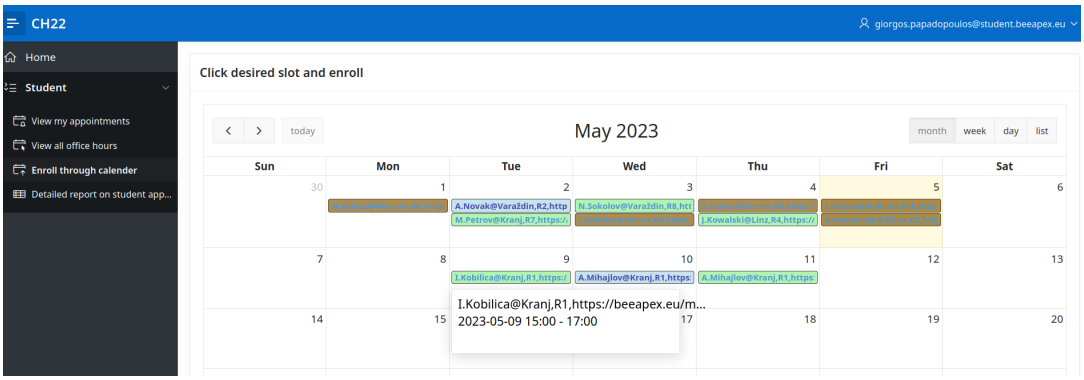
- kalendarza swoich umówionych spotkań (patrz Figure 22.11).
- kalendarza wszystkich godzin dyżurowych oferowanych przez nauczycieli zaangażowanych w program studiów studenta (patrz Figure 22.12).
- zapisu na godziny dyżurowe za pomocą kalendarza. Najpierw użytkownik wybiera żądane godziny dyżurowe (patrz Figure 22.13). Na formularzu student wybiera tylko cel (patrz Figure 22.14)) i klika przycisk “Enroll”. Jeśli zapis przebiegnie pomyślnie, student otrzymuje informację zwrotną (patrz Figure 22.15). Sprawdzenie zapisu za pomocą opcji “View my appointments” potwierdza udany zapis (patrz Figure 22.16).
- szczegółowego interaktywnego raportu przedstawiającego wszystkie umówione spotkania studentów w formie tabelarycznej (patrz Figure 22.17).



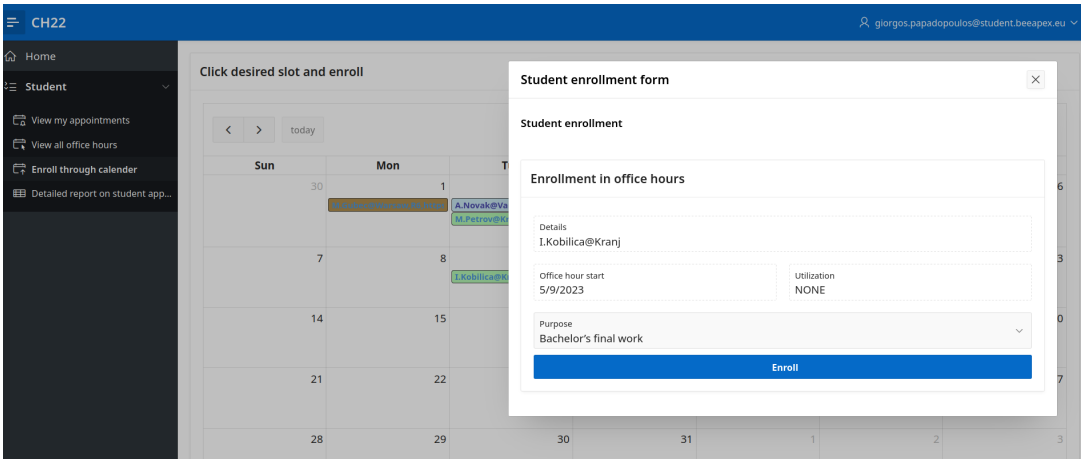
Rysunek 22.11: Kalendarz spotkań studenckich.



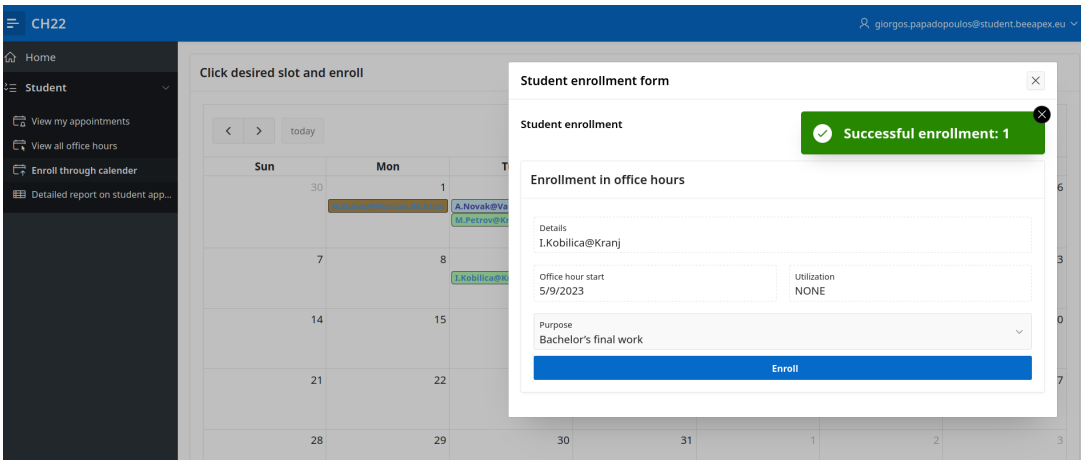
Rysunek 22.12: Kalendarz wszystkich godzin dyżurowych oferowanych przez nauczycieli w programie studiów studenta.



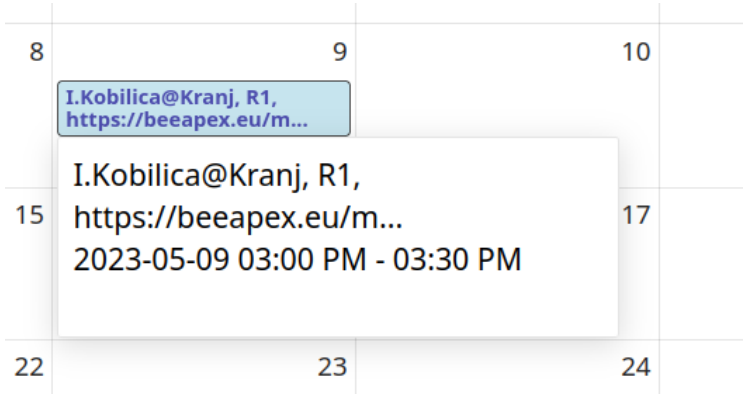
Rysunek 22.13: Rejestracja do godzin pracy biura za pośrednictwem kalendarza - wybieranie gniazda kalendarza.



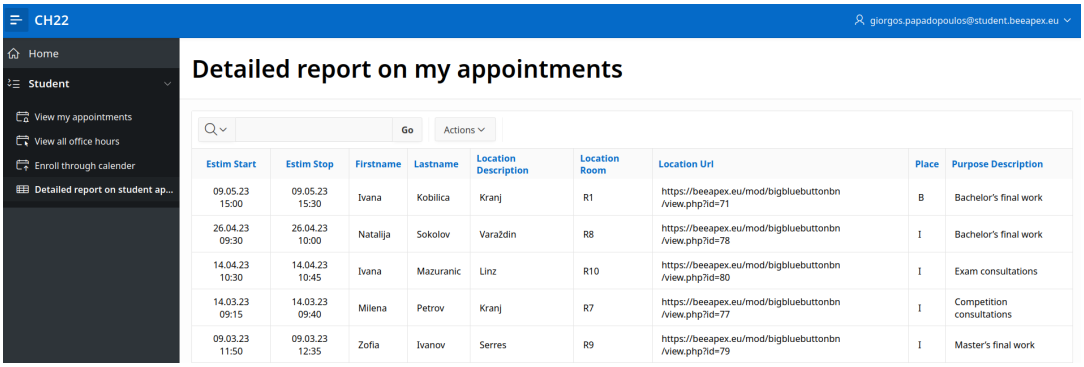
Rysunek 22.14: Rejestracja do godzin pracy biura poprzez kalendarz - wybór celu.



Rysunek 22.15: Rejestracja do godzin pracy biura za pośrednictwem kalendarza - pomyślna rejestracja.



Rysunek 22.16: Sprawdzanie rejestracji za pomocą pozycji menu "Viev my appointments".

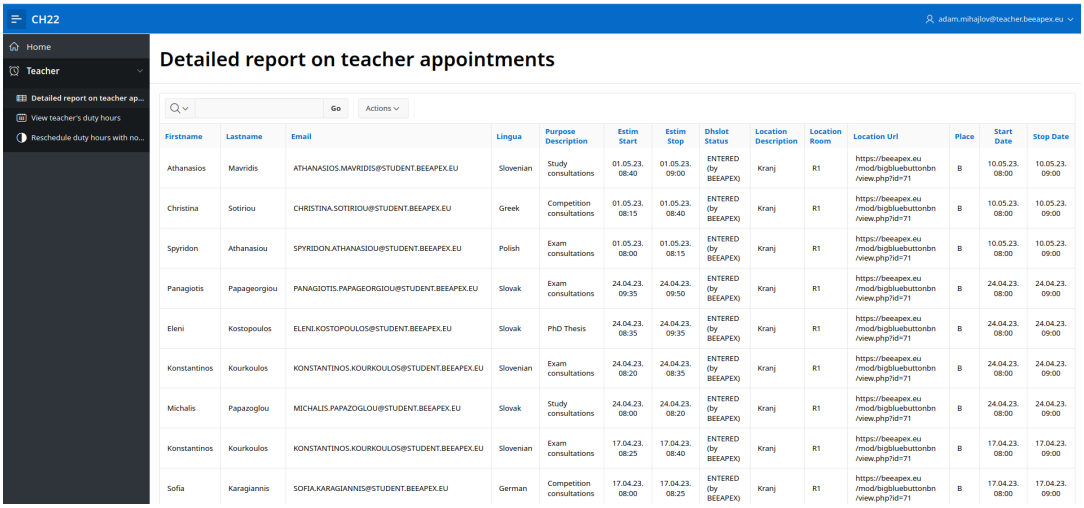


Rysunek 22.17: Szczegółowy interaktywny raport ze wszystkich spotkań studentów.

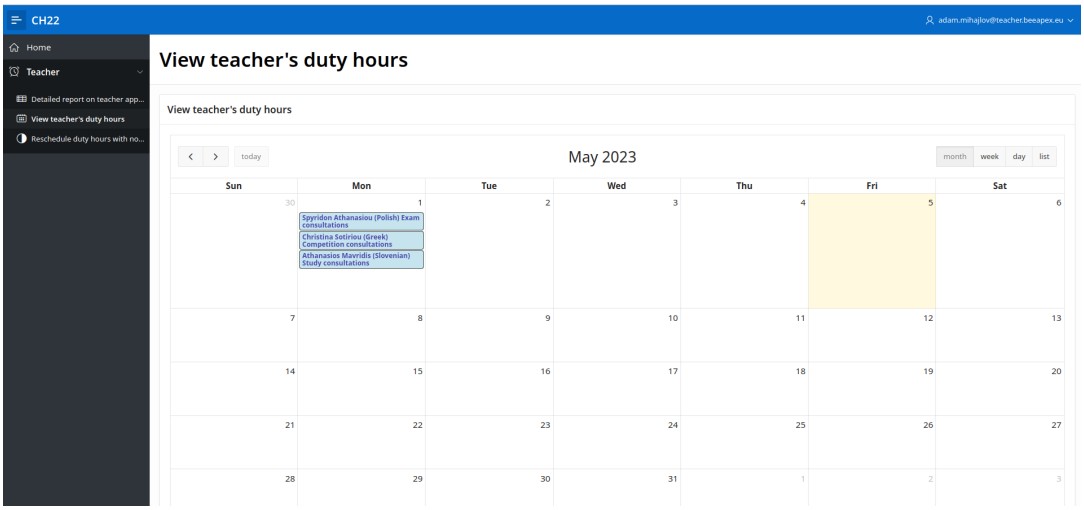
22.5.4 Interfejsy aplikacji dla nauczycieli

Elementy kalendarza są kolorowane w zależności od wykorzystania. Kolory statusów są identyczne jak w interfejsie dla studentów (BRAK jest zielony, DOSTĘPNY jest niebieski, a PRZEPEŁNIONY jest brązowy). Użytkownicy posiadający uprawnienia nauczyciela mogą:

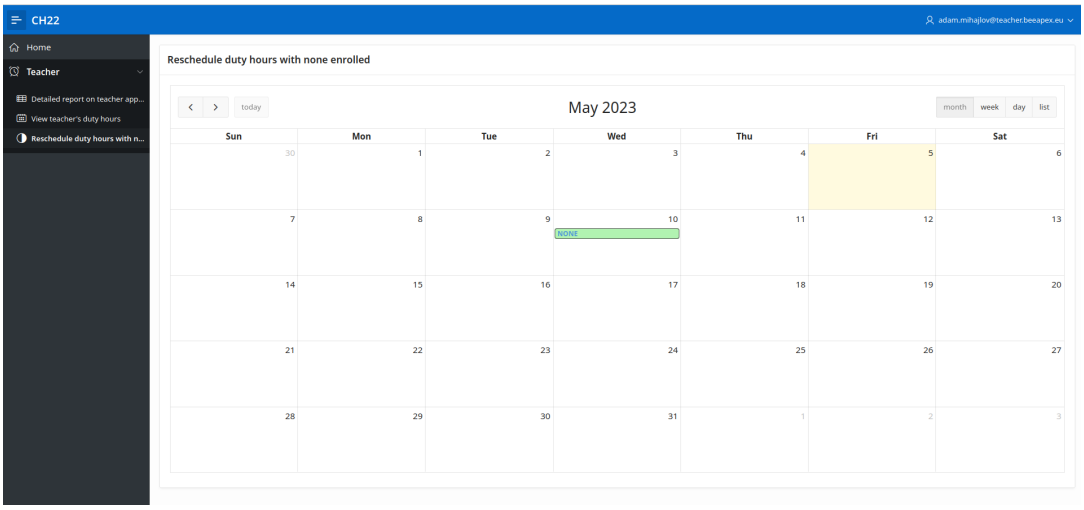
- przeglądać szczegółowy raport dotyczący spotkań nauczycieli (patrz Rysunek 22.18).
- przeglądać kalendarz nauczyciela. Dla każdego zapisanego studenta wyświetlane są imię, język i cel (patrz Rysunek 22.19).
- przeplanowywać godziny dyżurów, jeśli nikt nie jest zapisany. Najpierw wyświetlany jest kalendarz godzin dyżurów ze statusem BRAK (patrz Rysunek 22.20). Kliknięcie pozycji w kalendarzu otwiera formularz modalny, w którym wprowadzane są daty rozpoczęcia i zakończenia (również godziny i minuty) (patrz Rysunek 22.21). Po kliknięciu przycisku “Apply changes” (Zastosuj zmiany) godziny dyżuru są przenoszone na nową wybraną datę (patrz Rysunek 22.22).



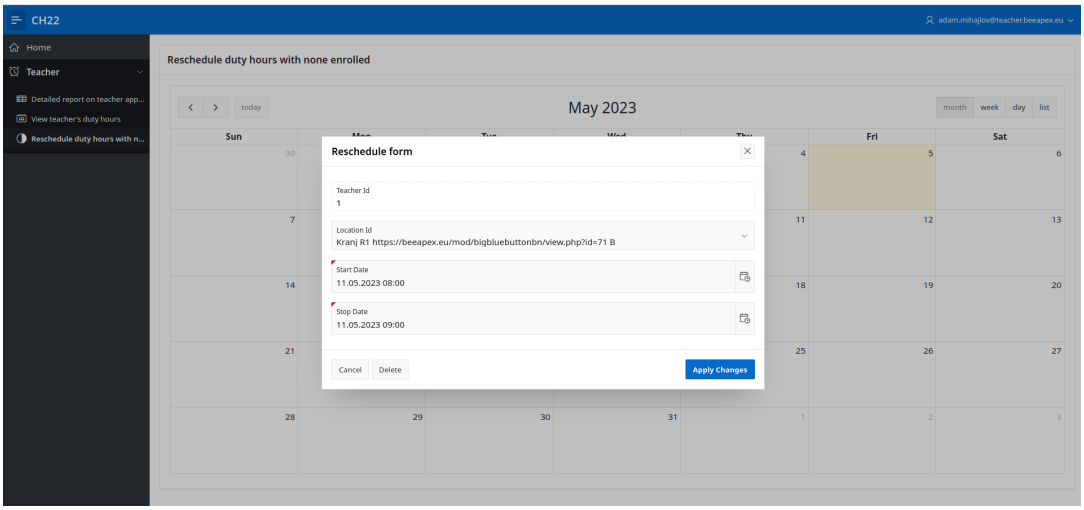
Rysunek 22.18: Szczegółowy raport z nominacji nauczycieli.



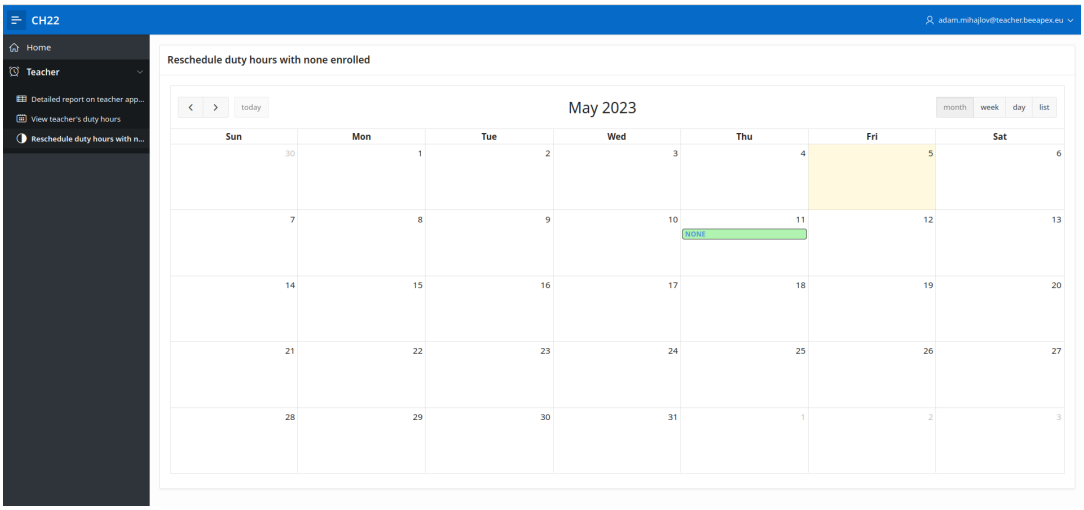
Rysunek 22.19: Kalendarz nauczyciela z widocznymi imionami uczniów, ich językami i celami.



Rysunek 22.20: Zmiana harmonogramu godzin pracy nauczyciela z NONE enrolled - widok kalendarza.



Rysunek 22.21: Zmiana harmonogramu godzin pracy nauczyciela z NONE zapisanym - wprowadzono nową datę.



Rysunek 22.22: Wyniki przeplanowania godzin dyżuru nauczyciela, gdy nikt nie jest zapisany.

22.6 Dodatkowe materiały do nauki

Możesz znaleźć następujące materiały dodatkowe:

- wyeksportowana aplikacja, skrypty do tworzenia tabel, wstawiania danych, tworzenia funkcji PL/SQL i usuwania tabel
- poradniki wideo: a) instalacja aplikacji pakietowej i tworzenie użytkowników (studentów i nauczycieli) oraz b) tworzenie aplikacji od podstaw

Wszystkie dodatkowe materiały są dostępne na publicznej stronie BeeAPEX . Zaloguj się jako użytkownik gość (nie jest wymagane hasło). Znajdź podręcznik w sekcji “Books”(Książki), skrypty w folderze Part 2 > Chapter22 w sekcji Scripts (Skrypty) i przewodniki wideo w Collection of video guides. Materiał na krótkie kursy w Short courses section.

22.6.1 Eksportowana aplikacja

Eksportowana aplikacja jest spakowana. Instalacja tworzy tabele, indeks, funkcję, a także wypełnia danymi. Deinstalacja usuwa wszystkie obiekty bazy danych używane w tej aplikacji. Aplikacja

pakietowa zawiera osadzony plik CSS (ch22_cal.css), który jest odwoływany przez większość stron kalendarza jako #APP_FILES#ch22_cal#MIN#.css:

```
.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;
}
```

Pulpit nawigacyjny zarządzania używa następujących zapytań:

```
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;
```

Zobacz inne zapytania dotyczące stron, które są oparte na elemencie kalendarza w spakowanej aplikacji.

22.6.2 Przewodniki wideo

Przewodniki wideo pokazują instalację spakowanej aplikacji i tworzenie aplikacji od podstaw.

22.7 Pytania

1. Jak dodać użytkownika z uprawnieniami nauczyciela?
2. W jaki sposób ulepszyłbyś aplikację, umożliwiając uczniowi robienie notatek na konkretne spotkanie?
3. Jak ograniczyć wyniki zapytania do wierszy, których kolumny dat są równe lub większe niż aktualny czas?

22.8 Odpowiedzi

1. Użyj opcji Zarządzanie użytkownikami i grupami. Dodaj nowego użytkownika. Użyj programu SQL Workshop > Object Browser i wybierz CH22_TEACHER tabeli. Wybierz zakładkę "Data" i "Insert row" . Adres e-mail wstawionego wiersza musi pasować do utworzonego użytkownika APEX.
2. Utwórz nową tabelę, czyli CH22_NOTES z polami: ID, note i DH_SLOT_ID jako klucz obcy, który odwołuje się do tabeli CH22_DH_SLOT. Utwórz nowy raport i formularz dla ucznia i autoryzuj stronę dla ucznia. W zależności od reguł instytucjonalnych możesz przyznać nauczycielom prawo do przeglądania lub modyfikowania notatek uczniów.
3. Aby ograniczyć wyniki kwerendy do wierszy, których kolumny dat są równe lub większe niż bieżąca godzina, należy dodać warunek WHERE date_column >= SYSDATE.



23. Sprawa Telco

VERONIKA ŠALGOVÁ, JOZEF KOSTOLNÝ, MICHAL MRENA, MICHAL KVET, MIROSLAV POTOČÁR

23.1 Biznesowy pogląd na sprawę

Aby skutecznie przedstawić krok po kroku proces tworzenia prototypu aplikacji, rozważmy hipotetyczną firmę VEYOMI jako przykład. VEYOMI to mała firma skupiająca się na świadczeniu usług telekomunikacyjnych dla swoich klientów. Ponieważ VEYOMI dąży do poszerzenia swojego portfolio produktów i obsłużenia rosnącej bazy klientów, niezwykle istotne staje się ustanowienie i utrzymanie solidnej aplikacji umożliwiającej płynne zarządzanie różnymi aspektami, w tym inwentarzem produktów, informacjami o klientach i zautomatyzowanymi systemami fakturowania. W coraz bardziej konkurencyjnym krajobrazie rynkowym VEYOMI uznaje, że niezmiernie ważne jest zapewnienie szybkiego i wygodnego dostępu do istotnych informacji dla swoich cennych klientów. Poprzez opracowanie intuicyjnej aplikacji, VEYOMI może zwiększyć swoją reaktywność, zapewniając, że istotne szczegóły dotyczące produktów, usług, cen oraz faktur są łatwo dostępne dla swojej wymagającej klienteli.

23.2 Definicja problemu

W świecie biznesu niezwykle istotne jest prowadzenie ewidencji swoich klientów, produktów lub usług, które zakupili, oraz wszelkich trwających usług, których są beneficjentami. Pozwala to mieć wszystkie niezbędne informacje na wyciągnięcie ręki, ułatwiając efektywne obsłużenie klientów. Dodatkowo istotne jest zapewnienie prostego i dostępnego sposobu dla klientów, aby sprawdzali status swoich usług. Poprzez oferowanie przyjaznej dla użytkownika platformy lub aplikacji umożliwiasz swoim klientom łatwe dowiedzenie się, co dzieje się z ich usługami i pozostanie na bieżąco. Poprzez utrzymywanie dokładnych danych oraz zapewnienie łatwego dostępu do aktualizacji usług, możesz zwiększyć satysfakcję klientów, zbudować zaufanie i poprawić ogólną działalność biznesową.

23.3 Przypadki użycia

Przy tworzeniu aplikacji konieczne jest rozpoczęcie od specyfikacji usług, które aplikacja powinna świadczyć oraz tego, co powinna obejmować. Dlatego też kluczowe jest przygotowanie listy poszczególnych funkcjonalności i usług. Niemniej jednak, konieczne jest najpierw skupienie się na

Tablica 23.1: Opis przypadku użycia: dodaj usługę.

Słowo kluczowe	**Wartość**
ID:	*Ch23-01*
Tytuł:	*Dodaj usługę*
Opis:	*Klient korzysta z aplikacji APEX, aby dodać nową usługę wśród dostępnych.*
Główny aktor:	*Klient*
Warunki wstępne:	*Dostęp do aplikacji, ważne poświadczenia, aplikacja internetowa jest dostępna.*
Warunki postu:	*Dane o nowej usłudze są przechowywane w bazie danych.*
Główny:	*Scenariusz*
Scenariusz sukcesu:	1. *Logowanie klienta do aplikacji.* 2. *Klient przechodzi do opcji Buy extra region (Kup dodatkowy region) na stronie głównej.* 3. *Klient wprowadza kwotę.* 4. *Za pomocą rozwijanego menu klient wybiera nową usługę (typ zakupu).* 5. *Klikając przycisk Buy (Kup), klient potwierdza aktywację nowej usługi.*
Rozszerzenia:	-
Częstotliwość stosowania:	*Klient zmienia zestaw usług mniej więcej raz na sześć miesięcy*
Status:	*Gotowy*
Właściciel:	*Klient*
Priorytet:	*wysoki*

określeniu rodzajów użytkowników. Dla naszego zastosowania możemy rozważyć trzy podstawowe role - klienta, menedżera i administratora. Każda z tych ról ma określone usługi, aby zapewnić ogólne pokrycie funkcjonalności aplikacji.

23.3.1 Opis narracyjny

Dla zwięzłości przedstawimy tylko dwa przypadki użycia spośród wielu:

- Dodawanie usługi - Klienci mogą łatwo poprosić o dodanie nowych usług do swojej istniejącej subskrypcji. System aktywuje nową usługę i aktualizuje rozliczenie klienta zgodnie z tym. Klient otrzymuje potwierdzenie oraz informacje dotyczące dodanej usługi.
- Pokaż stan usług - Można łatwo wyświetlić aktualny status i szczegóły subskrybowanych przez klienta usług. System pobiera i prezentuje przegląd subskrybowanych usług klienta. Obejmuje to informacje, takie jak rodzaj usługi, status aktywacji, pozostałe użycie lub okres ważności oraz wszelkie związane funkcje lub ograniczenia. W ramach przeglądu usług klienci mogą uzyskać dostęp do bardziej szczegółowych informacji dotyczących korzystania z usług. Może to obejmować zużycie danych, minuty rozmów, wysłane wiadomości tekstowe lub inne istotne wskaźniki wykorzystania.

23.3.2 Opis częściowo ustrukturyzowany

Dwa przypadki użycia przedstawiono w tabelach 23.1 i 23.2.

23.3.3 Diagram przypadków użycia

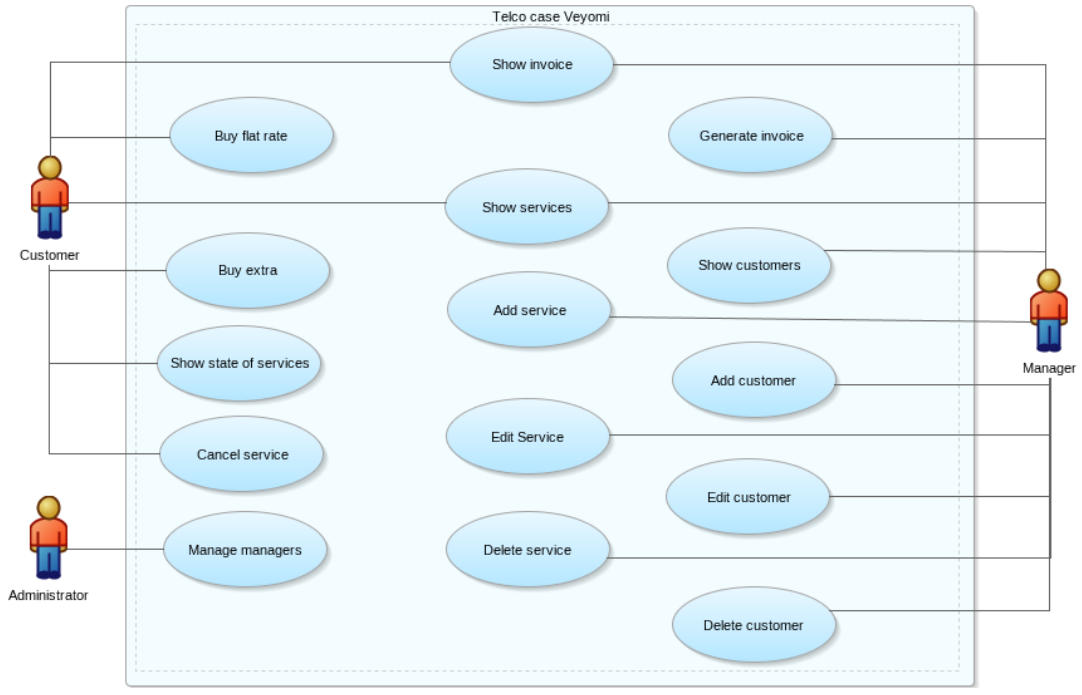
Klient musi upewnić się, że zakupuje usługi stałego taryfikatora oraz dodatkowe usługi, wyświetlając status usług, które powinien również móc anulować. Inną istotną funkcjonalnością do uwzględnienia jest wyświetlanie przeglądu faktur.

Tablica 23.2: Opis przypadku użycia: pokaż stan usługi.

Słowo kluczowe	**Wartość**
ID:	*Ch23-02*
Tytuł:	*Pokaż użycie usług*
Opis:	*Klient korzysta z aplikacji APEX, aby uzyskać wgląd w wykorzystanie usług.*
Główny aktor:	*Klient*
Warunki wstępne:	*Dostęp do aplikacji, ważne poświadczenia, aplikacja internetowa jest dostępna.*
Warunki postu:	-
Główny:	*Scenariusz*
Scenariusz sukcesu:	1. *Logowanie klienta do aplikacji.* 2. *Klient wybiera zakładkę Stats (Statystyki).* 3. *Za pomocą rozwijanego menu klient wybiera usługę. Zostanie wyświetlony wykres słupkowy tej konkretnej usługi.*
Rozszerzenia:	-
Częstotliwość stosowania:	*Klient wymaga wglądu mniej więcej raz w miesiącu*
Status:	*Gotowy*
Właściciel:	*Klient*
Priorytet:	*wysoki*

Rola użytkownika menedżera może uzyskać dostęp do przeglądu klientów oraz ich dodawania, edytowania i usuwania. Ponadto menedżer może również generować faktury.

Ostatnią rolą użytkownika jest administrator, który ma jedną podstawową funkcjonalność: zarządzanie menedżerami. Reprzentacja tych ról została przedstawiona na ilustracji 23.1.



Rysunek 23.1: Diagram przypadków użycia.

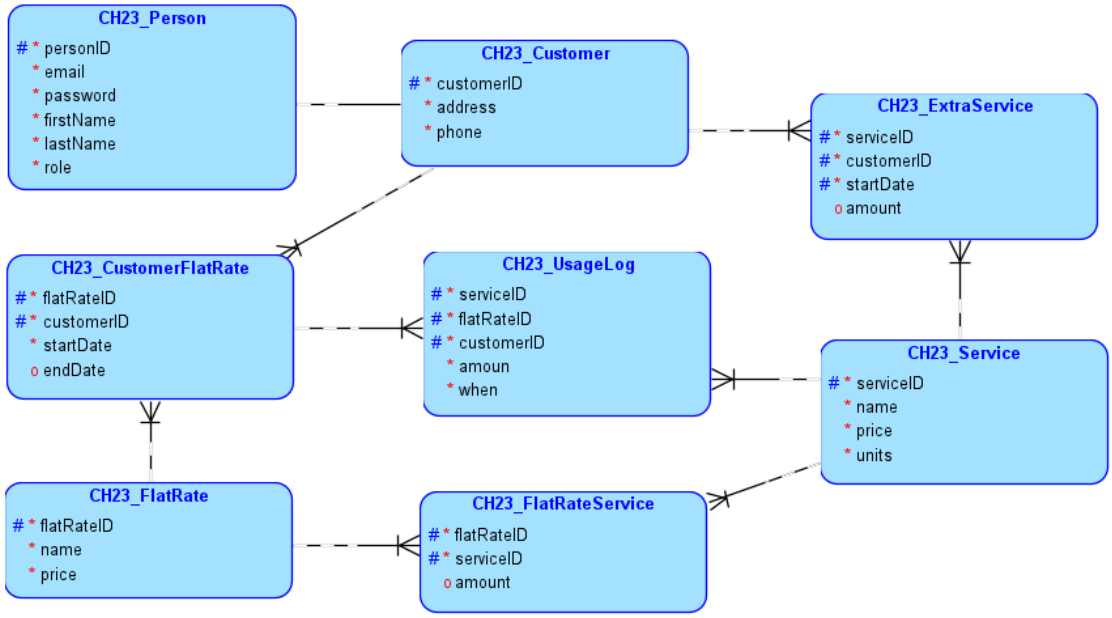
23.4 Model danych

23.4.1 Opis modelu danych

Tabela **CH23_Person** przechowuje podstawowe dane osobowe oraz typ roli użytkownika. Gdy osoba jest klientem, tworzony jest wpis w tabeli **CH23_Customer** , gdzie przechowywane są jego adres oraz numer telefonu oprócz kluczy. Tabela **CH23_FlatRate** przechowuje informacje dotyczące taryfy ryczałtowej, mianowicie nazwę i cenę. Dane dotyczące poszczególnych usług, ich nazwa, cena i oferowane jednostki, przechowywane są w tabeli **CH23_Service** . Poszczególne usługi przypisane do taryf ryczałtowych przechowywane są w tabeli **CH23_FlatRateService** gdzie oprócz kluczy mamy informację o przydzielonej ilości. Jeśli klient nabywa taryfę ryczałtową, informacje na ten temat przechowywane są w tabeli **CH23_CustomerFlatRate**, w tym data rozpoczęcia i zakończenia. Zużyte jednostki z taryfy ryczałtowej są rejestrowane w tabeli **CH23_UsageLog**, gdzie wiemy, który klient korzystał z której usługi w ramach której taryfy ryczałtowej, wraz z datą i ilością. Oprócz taryfy ryczałtowej, klient może nabywać dodatkowe usługi, które są rejestrowane w tabeli **CH23_ExtraService** wraz z ich ilością.

23.4.2 Logiczny model danych

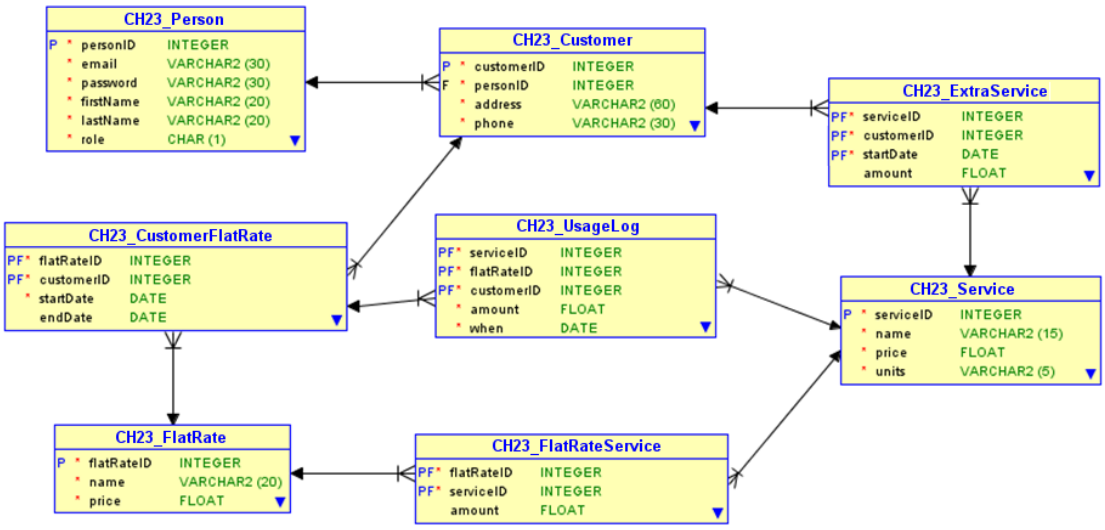
Logiczny model danych przedstawiono na rysunku 23.2.



Rysunek 23.2: Logiczny model danych.

23.4.3 Relacyjny model danych

Relacyjny model danych przedstawiono na rysunku 23.3.



Rysunek 23.3: Relacyjny model danych.

23.5 Uwierzytelnianie użytkowników i role użytkowników

W powyższym opisie zidentyfikowaliśmy trzy role użytkowników. Po zalogowaniu się aplikacja udostępnia użytkownikowi treść zgodnie z jego rolą. Jednak sam proces logowania jest taki sam dla wszystkich użytkowników. APEX oferuje nam kilka sposobów zarządzania użytkownikami i uwierzytelniania ich. W poniższym wykazie prezentujemy kilka wybranych metod:

- Konto bazy danych,
- Serwer LDAP,
- Konto Oracle APEX,
- Logowanie za pomocą platformy społecznościowej,
- Niestandardowe.

W narzędziu Oracle Apex wybieramy metodę uwierzytelniania - **Authentication Scheme**- w podstronie **Shared Components \ Authentication Schemes** w Na tej podstronie znajduje się lista istniejących schematów logowania, a także można tutaj utworzyć nowy schemat logowania. Tworząc nowy schemat, możemy wybrać jedną z powyższych metod uwierzytelniania. Na potrzeby naszej aplikacji wybraliśmy najbardziej ogólną metodę, czyli Niestandardową.

Główną częścią niestandardowego uwierzytelniania jest tzw. funkcja uwierzytelniania. Wejściem do tej funkcji jest nazwa użytkownika i hasło, które użytkownik wprowadza na stronie logowania, a wyjściem jest wartość logiczna. Zgodnie z oczekiwaniami funkcja zwraca wartość **true**, jeśli podana nazwa i hasło pasują do istniejącego użytkownika, a w przeciwnym razie zwraca wartość **false**. Na listingu możemy zobaczyć nagłówek takiej funkcji. Dlatego przy tworzeniu niestandardowego schematu logowania musimy podać nazwę takiej funkcji. Innymi słowy, taka funkcja musi już istnieć.

```
# The header of an authentication function.
CREATE OR REPLACE FUNCTION authenticate_user
(p_username IN varchar2,
p_password IN varchar2)
RETURN boolean;
```

Implementujemy ciało funkcji uwierzytelniania za pomocą języka PL/SQL, czyli to od nas zależy, jak uwierzytelniamy użytkownika. Dlatego metoda Niestandardowa jest najbardziej ogólna. Możemy na przykład skontaktować się z zewnętrzną usługą uwierzytelniania za pomocą interfejsu

API REST, wyszukać użytkownika w lokalnej bazie danych lub pozostawić uwierzytelnianie innemu procesowi lokalnemu. W naszej aplikacji wybraliśmy powszechnie stosowaną procedurę wyszukiwania i weryfikacji użytkownika w lokalnej bazie danych.

W modelu danych, który przedstawiliśmy powyżej, można zauważyć tabelę CH23_Person Kolumny **email** i **password** są istotne dla uwierzytelniania. Adres e-mail użytkownika służy również jako unikalna nazwa logowania. Kolumna hasła zawiera hasła użytkowników przechowywane w bezpiecznej zahaszowanej formie. Opis bezpiecznego przechowywania hasła wykracza poza zakres tego rozdziału. Czytelnik może dowiedzieć się więcej na ten temat w literaturze. Niemniej jednak, przed wprowadzeniem danych lub edycją kolumny hasła w tabeli **CH23_Person**, konieczne jest zahashowanie hasła. W tym celu stworzyliśmy pomocniczą funkcję **hash_password**, której nagłówek możemy zobaczyć w listingu. Ta funkcja przyjmuje nazwę użytkownika i hasło w postaci zwykłego tekstu i generuje hasło w zabezpieczonej zahaszowanej formie, które może zostać przechowane w bazie danych.

```
# The header of the auxiliary function hash\_password.
CREATE OR REPLACE FUNCTION hash_password
(p_username IN varchar2,
p_password IN varchar2)
RETURN varchar2;
```

Rozwiązaliśmy automatyczne hashowanie hasła podczas pracy z tabelą **CH23_Person** poprzez stworzenie wyzwalaczy dla tabeli CH23_Person. Wyzwalacze haszują hasło w postaci zwykłego tekstu za pomocą funkcji **hash_password** opisanej powyżej przed wstawieniem lub aktualizacją kolumny hasła. Zapewni to, że wszystkie hasła w tabeli **CH23_Person** są przechowywane w sposób bezpieczny - nawet administrator bazy danych nie ma dostępu do rzeczywistego hasła. W Tabeli 23.3 możemy zobaczyć przykład danych przechowywanych w tabeli **CH23_Person**. Co ciekawe, dwaj użytkownicy wymienieni w tabeli mają takie samo hasło. Jednak na podstawie bezpiecznie przechowywanych haseł nie możemy stwierdzić tego faktu.

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'

Tablica 23.3: Przykładowe dane przechowywane w tabeli CH23_Person .

I Implementacja funkcji uwierzytelniania jest teraz prosta. W tej funkcji najpierw przeszukujemy tabelę **CH23_Person** w poszukiwaniu hasła użytkownika o podanej nazwie użytkownika. Następnie haszujemy hasło, które użytkownik wprowadził, za pomocą funkcji **hash_password**. Jeśli oba hashe się zgadzają, uwierzytelnienie użytkownika powiedzie się, i zwracamy wartość **true** z funkcji. Z drugiej strony, jeśli hasze się nie zgadzają lub nie ma użytkownika o podanej nazwie w tabeli CH23_Person, zwracamy wartość **false**.

Wada wybranej przez nas metody Niestandardowej polega na większej początkowej trudności w tworzeniu bazy danych, w ramach której konieczne jest opracowanie funkcji i wyzwalaczy pomocniczych. Z drugiej strony zaleta polega na tym, że my, programiści aplikacji, kontrolujemy cały proces uwierzytelniania. Inną zaletą jest łatwa skalowalność zastosowanej implementacji. Jeśli

na przykład w przyszłości chcielibyśmy użyć zewnętrznej usługi do uwierzytelniania, musielibyśmy zmienić implementację funkcji uwierzytelniania, aby kontaktowała się z zewnętrzną usługą zamiast lokalnej bazy danych.

Drugim ważnym zadaniem związanym z zarządzaniem użytkownikami jest zarządzanie rolami użytkowników. Chociaż proces logowania był taki sam dla wszystkich użytkowników, podczas zarządzania rolami musieliśmy zapewnić dostęp do różnych części aplikacji w zależności od roli użytkownika. Zarządzanie rolami jest ściśle powiązane z zarządzaniem użytkownikami. Najpierw w środowisku Oracle APEX należy określić wszystkie możliwe role użytkowników. Zarządzanie rolami znajdziemy na podstronie **Shared Components \ Application Access Control**. Na tej podstronie widzimy listę istniejących ról, ale możemy tu także utworzyć nową rolę. Dla każdej nowej roli konieczne jest podanie jej statycznego identyfikatora, z którego będziemy później korzystać przy przypisywaniu roli poszczególnym użytkownikom. W naszej aplikacji wykorzystujemy role posiadające następujące statyczne identyfikatory:

- ADMIN,
- CUSTOMER,
- MANAGER.

Przypisujemy rolę do użytkownika za pomocą kodu PL/SQL. W Listingu możemy zobaczyć przykładowy kod przypisania roli **CUSTOMER** (statyczny identyfikator roli) użytkownikowi **LOBORTIS@OUTLOOK.EDU**. Użyliśmy kolejnego zestawu wyzwalaczy dla tabeli Person, aby zautomatyzować ten proces. Kolumna **Role** w tej tabeli może przyjmować wartości z zbioru (a, m, c), które zgadzają się z początkowymi literami powyżej zdefiniowanych ról. W ciele wyzwalacza, przed wstawieniem lub aktualizacją danych, ustawiamy odpowiednią rolę dla danego użytkownika zgodnie z wartością kolumny Role, korzystając z kodu podobnego do kodu w Listingu.

```
#PL/SQL Code that associates a given user with a given role.
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;

INSERT INTO CH23_PERSON (EMAIL, PASSWORD, FIRSTNAME, LASTNAME, ROLE)
VALUES (:CUSTOMER_EMAIL, :CUSTOMER_PASSWORD, :CUSTOMER_FIRSTNAME,
:CUSTOMER_LASTNAME, 'C');
```

Ostatnim krokiem w implementacji ról użytkowników w aplikacji jest wyświetlanie różnych podstron w zależności od roli zalogowanego użytkownika. Dla każdej podstrony, która jest przeznaczona wyłącznie dla określonej roli użytkownika, konieczne jest wybranie roli użytkownika, dla której dana podstrona jest przeznaczona, w ustawieniach **Page Designer** w sekcji **Security \ Authorization Scheme**. Ten krok zapewnia, że użytkownicy z innymi rolami nie mogą uzyskać dostępu do tej podstrony. Na przykład strona główna naszej aplikacji jest specyficzna w tym sensie, że nie wymaga logowania się, aby ją zobaczyć. Ustawiliśmy również to zachowanie w sekcji **Security**, w której wybraliśmy opcję **Page Is Public**.

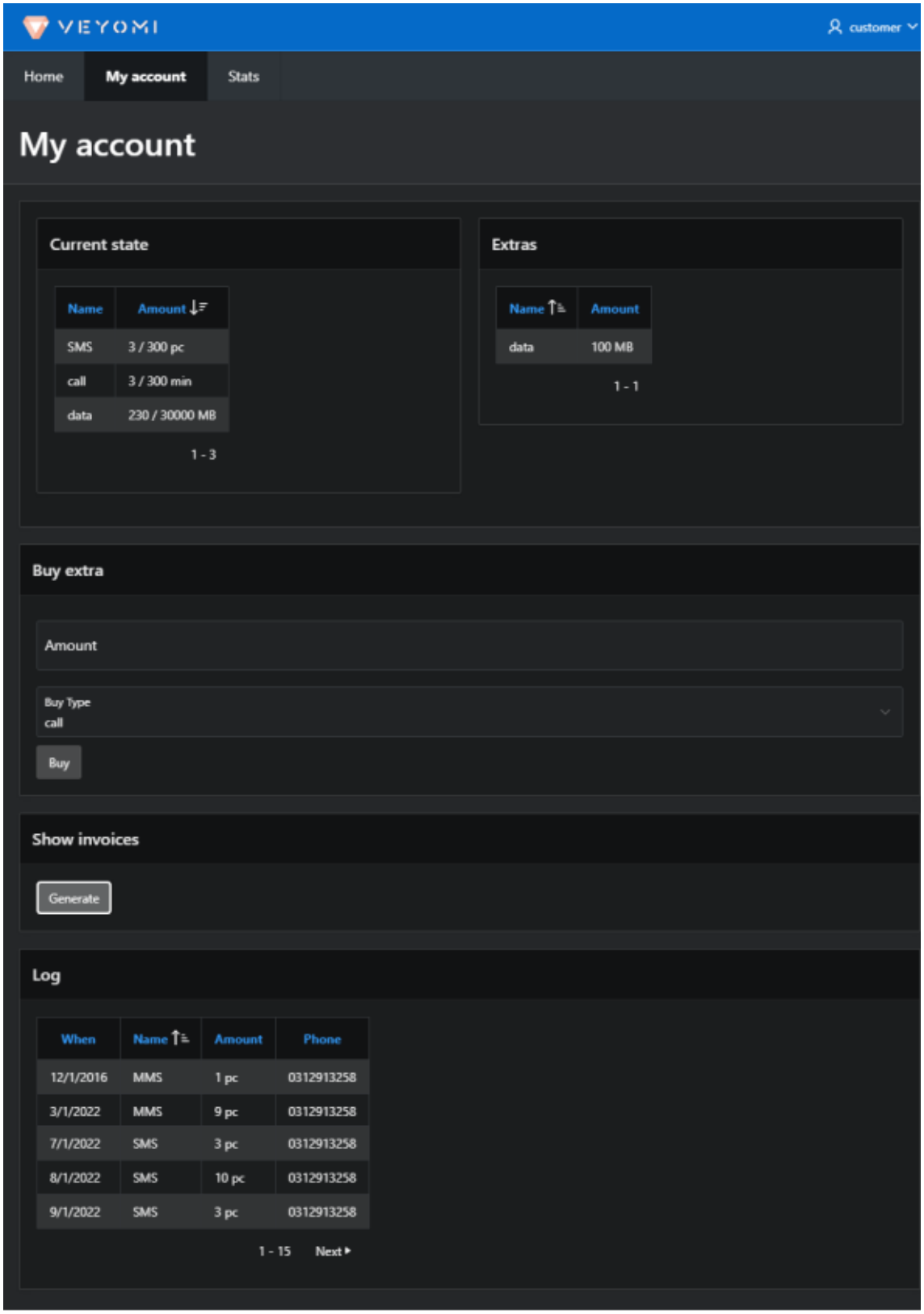
Opisane powyżej ustawienia zapewnią, że użytkownik ma dostęp tylko do podstron swojej roli. Niemniej jednak wszyscy użytkownicy wciąż zobaczą wszystkie podstrony w menu nawigacyjnym - nawet te, do których nie mają dostępu. Takie zachowanie z pewnością nie jest pożądane. Aby aplikacja pokazywała użytkownikom tylko te podstrony, do których mają dostęp, konieczne jest dostosowanie ustawień menu nawigacyjnego. Możemy to ustawić w podstronie **Shared Components \ Navigation and Search \ Navigation Menu**. Tutaj, w kolumnie **Authorization Scheme** dla każdego elementu w menu, konieczne jest ustawienie roli użytkownika, dla której dany element będzie wyświetlany.

23.6 Interfejsy aplikacji

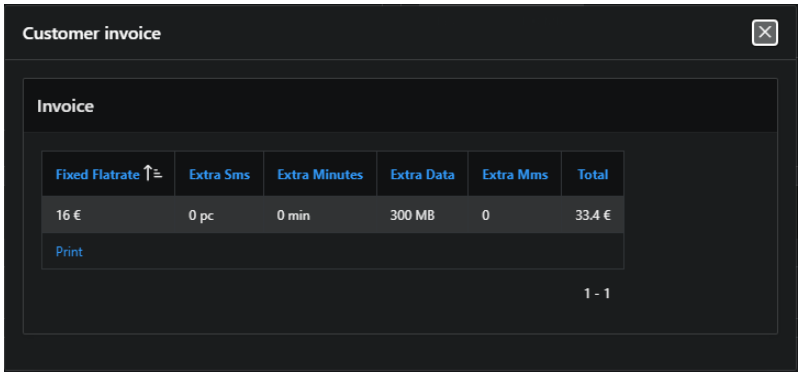
Projekt aplikacji polega na stworzeniu indywidualnych podstron, które będą oferować proponowane usługi. Główną stroną jest strona główna, która zapewnia przegląd oferowanych usług i portfela usług taryfowych. Można wykorzystać tę stronę do prezentacji firmy i świadczonych usług, które klient może nabyć. Ta strona jest dostępna bez konieczności logowania się dla wszystkich odwiedzających. Główna strona oferuje jeszcze jedną ważną funkcjonalność: dostęp do administracji po zalogowaniu. Przez administrację rozumiemy zbiór podstron podzielonych według typu konta, do którego się logujemy - klienta, menedżera lub administratora. Dlatego dla każdego z nich konieczne jest stworzenie osobnych podstron, które obejmują proponowaną funkcjonalność usług danej roli. Strona klienta zapewnia przegląd bieżącego statusu zamówionych usług i dodatkowych usług w poszczególnych polach. Kolejna dodatkowa karta umożliwia zakup określonych usług poprzez wybranie typu usługi z pola kombi. Kolejna funkcjonalność umożliwia wyświetlanie i generowanie faktur, co można rozwiązać za pomocą wyskakującego okna, w którym po otwarciu faktura jest wyświetlana w formie raportu, co pozwala na konfigurację zapisywania i drukowania w formacie PDF. Raport dla faktury jest wywoływany przez określony skrypt podsumowujący dane dla świadczonej usługi w danym miesiącu. Kolejną istotną częścią strony klienta jest przegląd wykorzystania usług z szczegółami, takimi jak ilość, typ usługi i czas. Ta tabela umożliwia przegląd filtrowania i sortowania za pomocą specyfikacji filtru i sortowania według poszczególnych kolumn. Taka funkcjonalność może być przydatna dla klienta, gdy chce uzyskać, na przykład, przegląd liczby wysłanych wiadomości SMS w danym okresie. Na koniec klientowi udostępniana jest funkcjonalność wyświetlania statystyk w postaci wykresów w osobnym folderze. Ta funkcjonalność jest obsługiwana na osobnej stronie, która jest dołączona do strony klienta za pomocą karty, i można przełączać się między poszczególnymi funkcjonalnościami. Strona ze statystykami zawiera prezentację graficznej reprezentacji logów. Taka reprezentacja jest odpowiednim sposobem prezentacji zmiany w czasie. Na przykład możemy łatwo wyświetlić minuty rozmów podczas poszczególnych miesięcy, liczbę wysłanych wiadomości SMS lub ilość przesłanych danych w zamówionym programie.

23.6.1 Projektowanie aplikacji

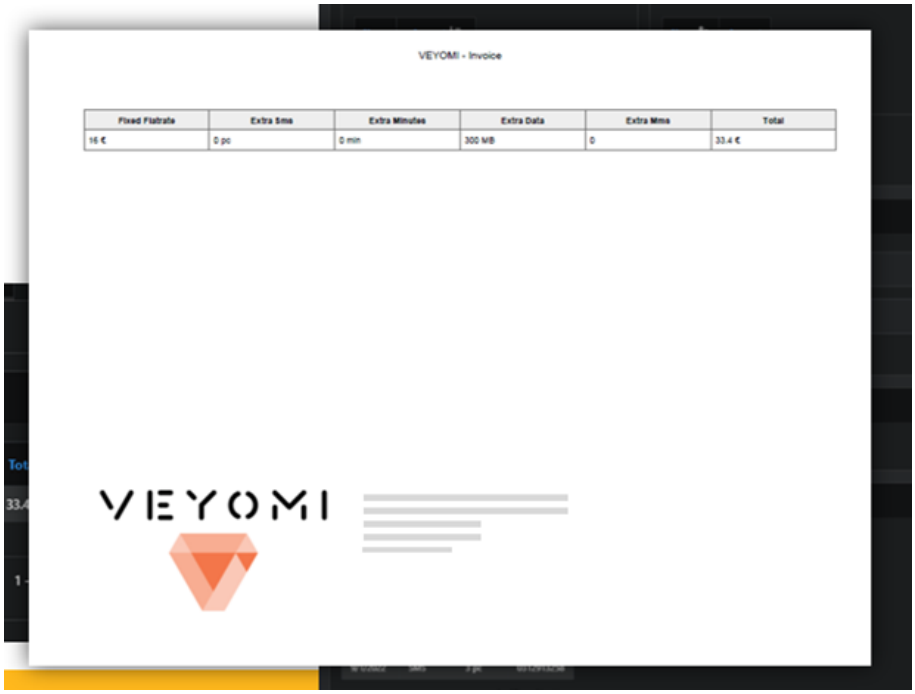
W tym rozdziale wyświetlane są zrzuty ekranu formularzy aplikacji różnych ról użytkowników.



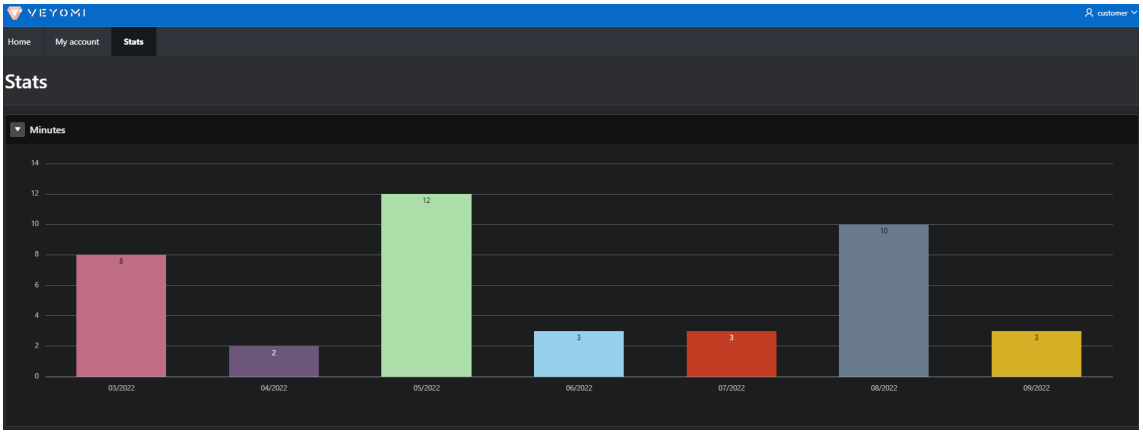
Rysunek 23.4: Pulpit nawigacyjny klienta.



Rysunek 23.5: Pulpit klienta - okno modalne faktury.



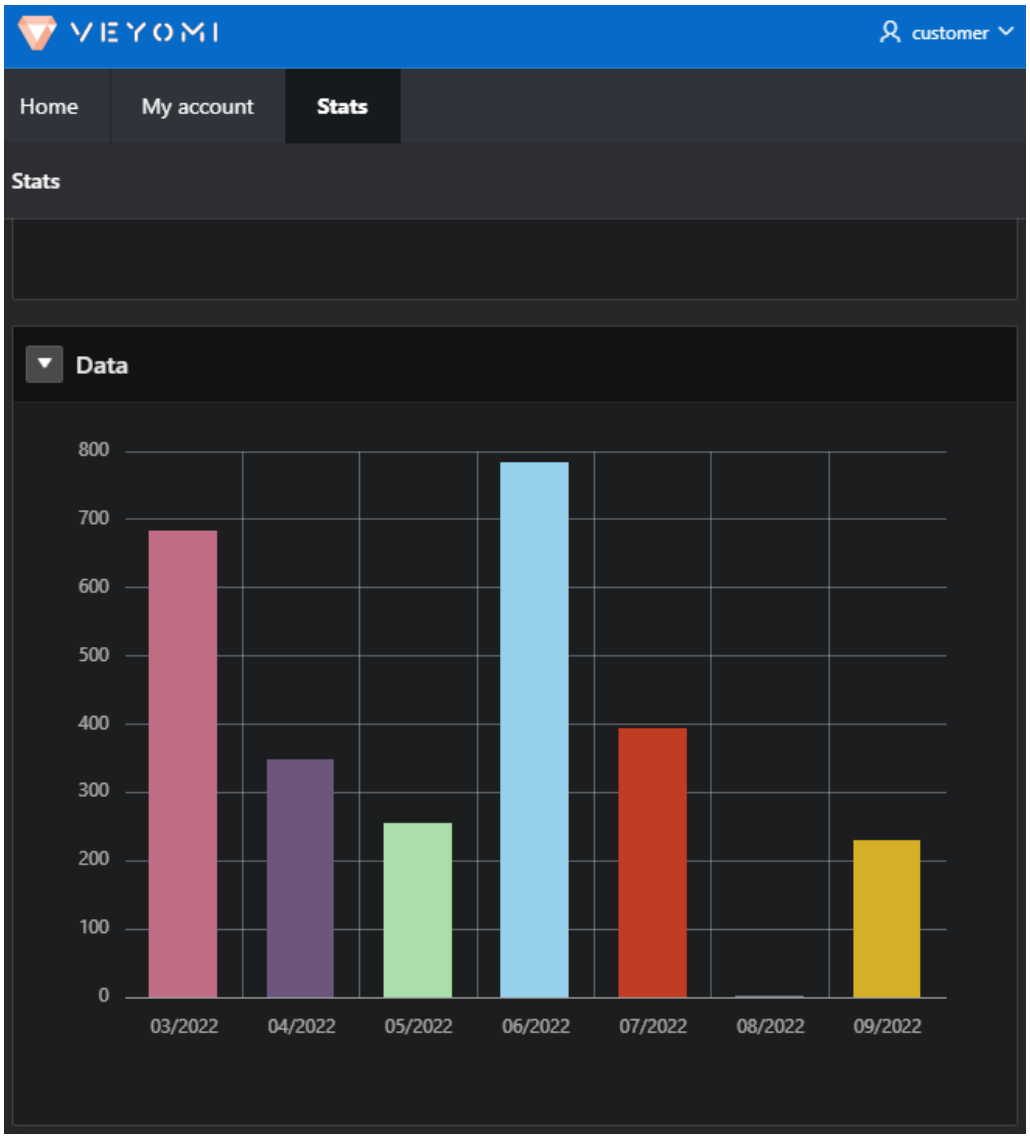
Rysunek 23.6: Pulpit klienta - faktura w formacie PDF.



Rysunek 23.7: Pulpit klienta - statystyki minut.



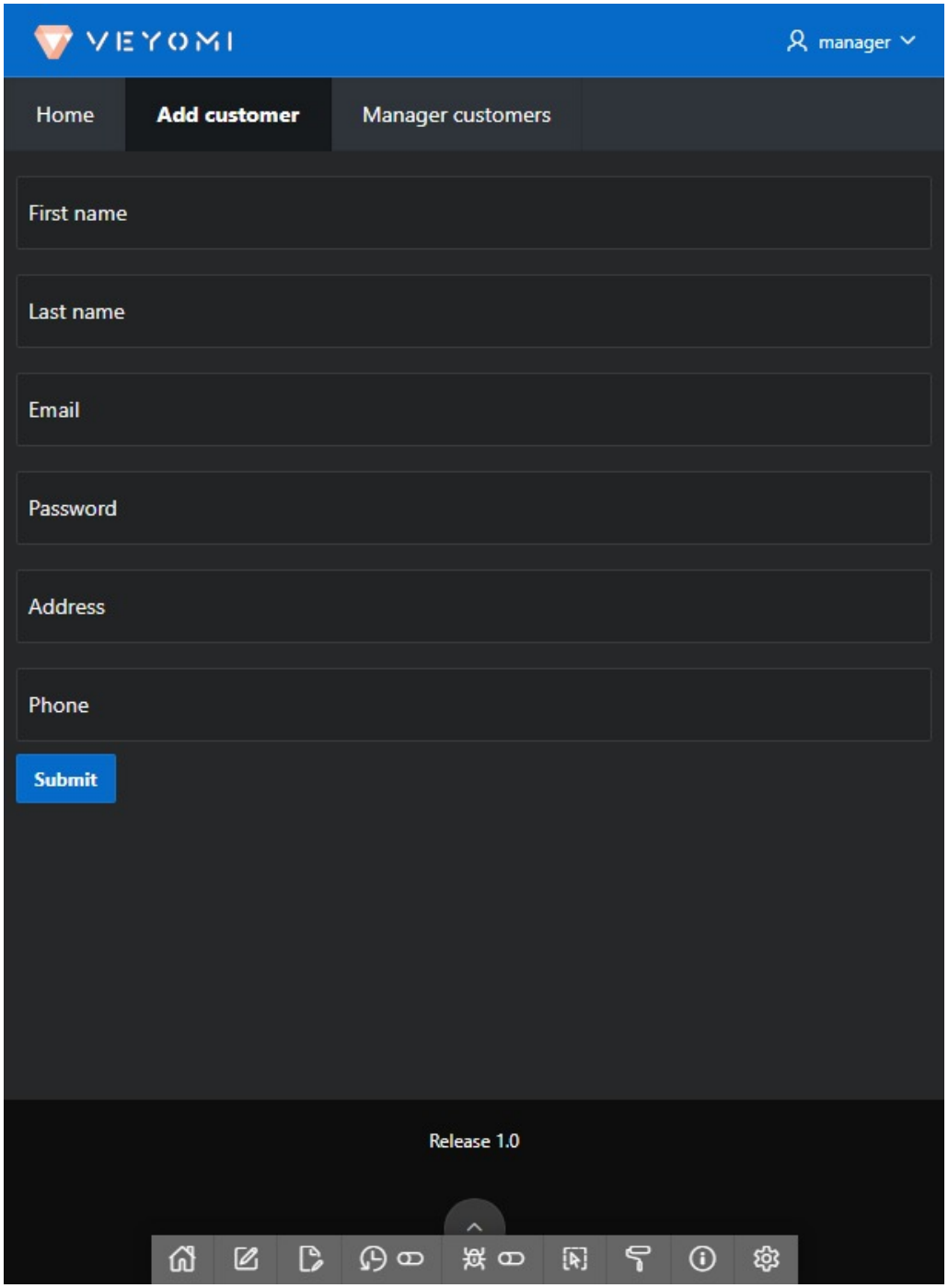
Rysunek 23.8: Pulpit klienta - Statystyki SMS.



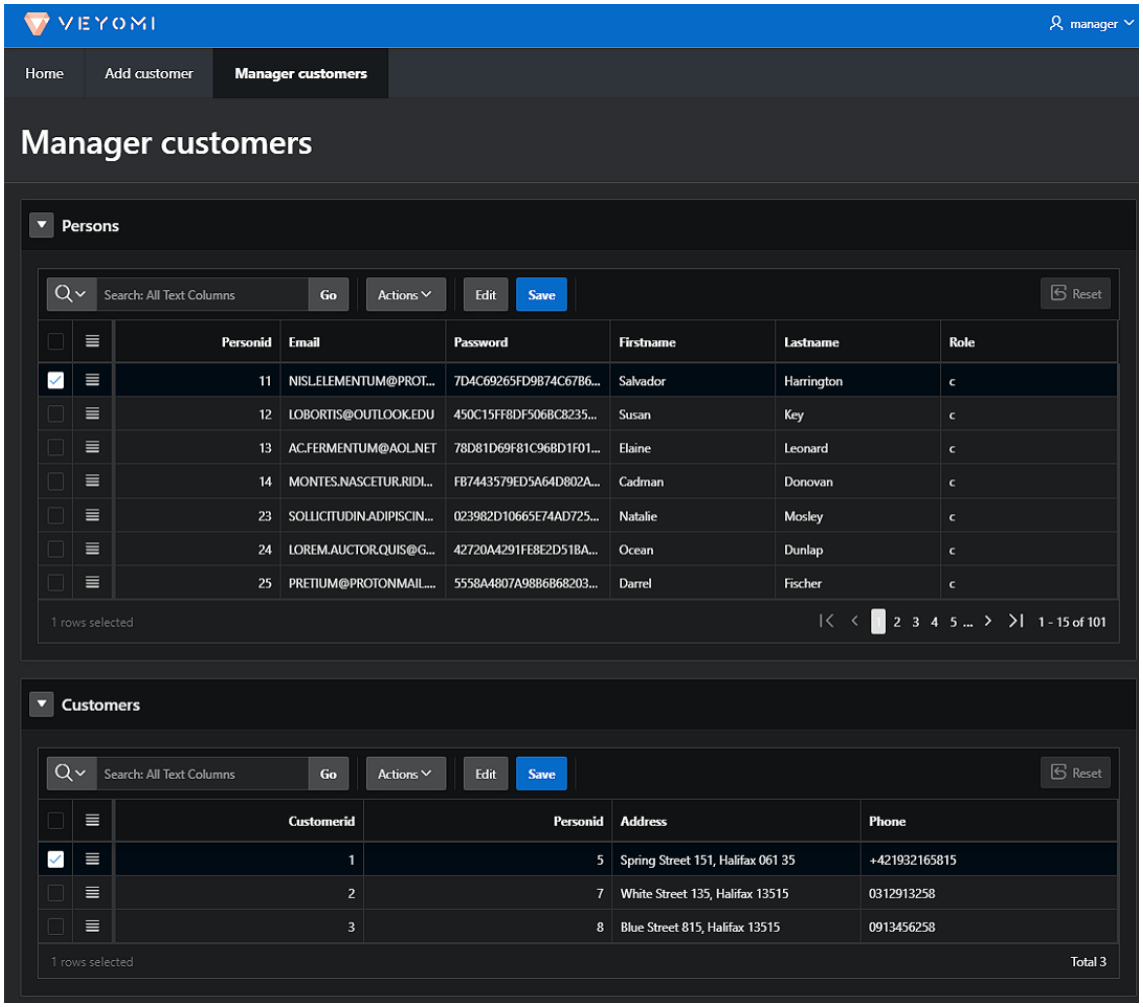
Rysunek 23.9: Pulpit nawigacyjny klienta - statystyki danych.

Po zalogowaniu się menedżera do aplikacji, funkcjonalność zarządzania klientami jest udostępniana na osobnych zakładkach, które są częścią strony menedżera. Pierwszą funkcjonalnością jest

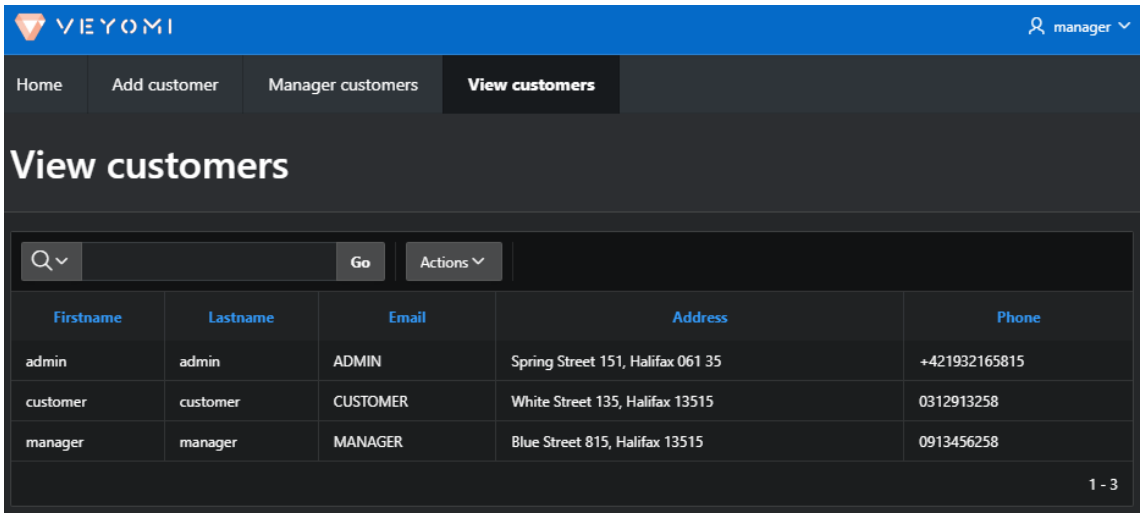
dodanie nowego klienta, gdzie menedżer może dodać i utworzyć nowe konto klienta. Następnie używamy komponentu zarządzania klientami do edycji informacji o klientach, które są projektowane w siatce danych. Ten komponent zapewnia również funkcjonalność prostego raportowania lub eksportowania informacji o klientach do aplikacji innych firm, jeśli jest to konieczne. Na koniec karta wyświetlania odbiorcy przedstawia uproszczoną formę wyświetlania przeglądu informacji o kliencie. Ta siatka jest tak skonfigurowana, że nie pozwala na edycję poszczególnych elementów, ale ma możliwość tworzenia raportów i eksportów z tej listy.



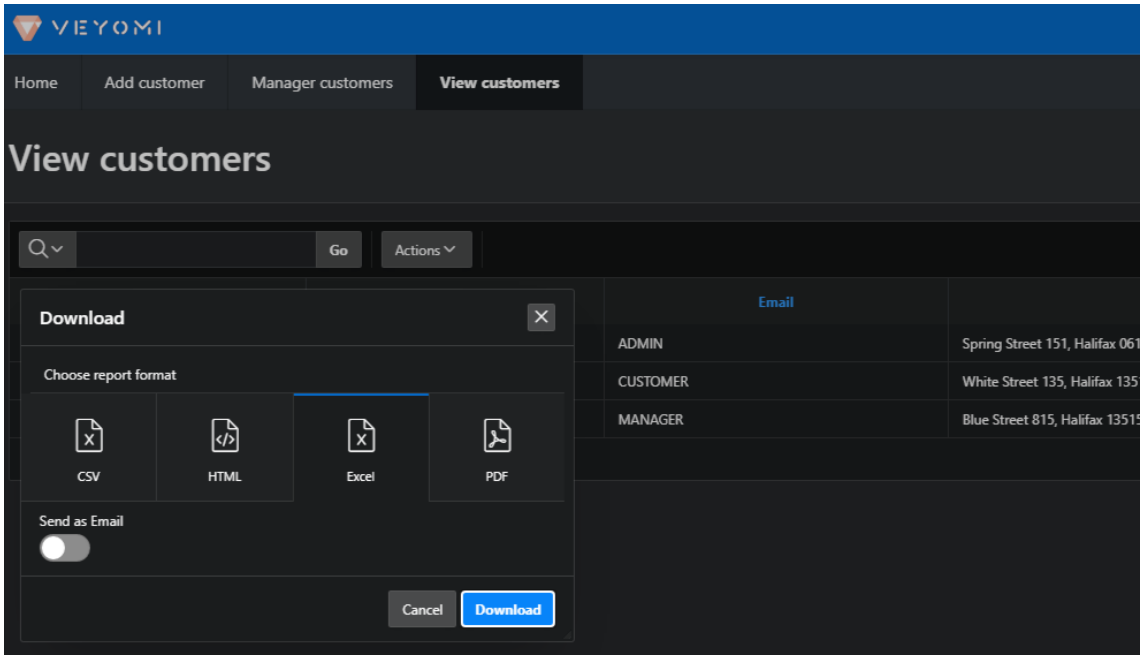
Rysunek 23.10: Manager dashboard – Add customer.



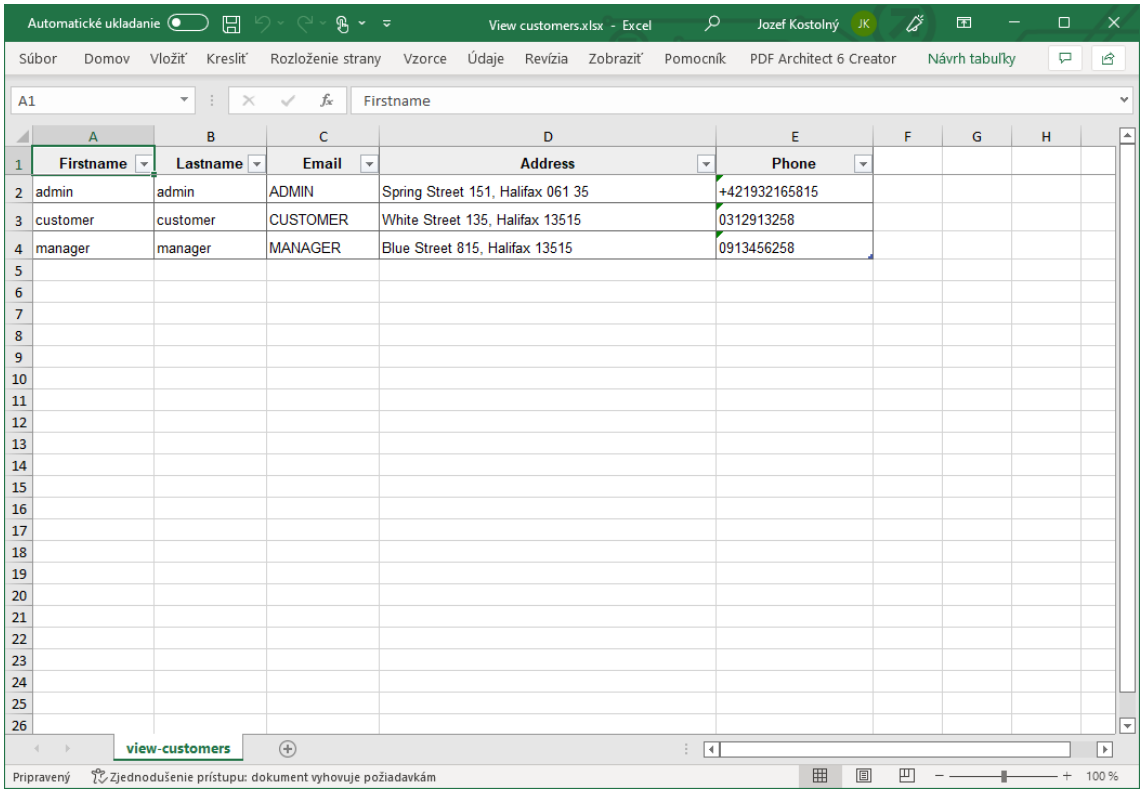
Rysunek 23.11: Manager dashboard – Manage customer.



Rysunek 23.12: Manager dashboard – View customers.

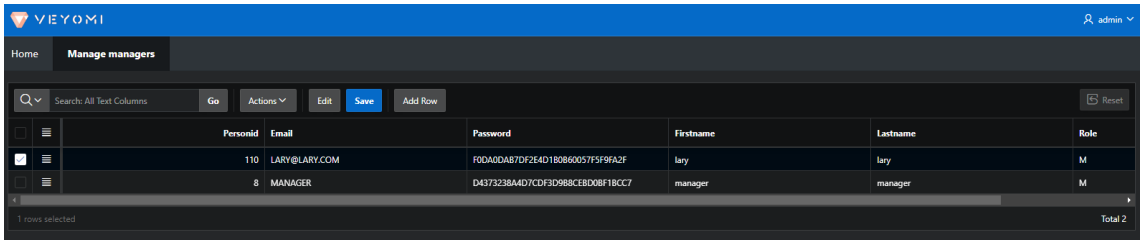


Rysunek 23.13: Manager dashboard – Customer export.



Rysunek 23.14: Manager dashboard - Customer export in XLS file.

Ostatnią stroną, którą należy utworzyć, jest strona dla administratora aplikacji. Na tej stronie zapewniamy usługę zarządzania menedżerami w edytowalnej siatce danych, umożliwiając eksportowanie raportów i filtrowanie funkcji.

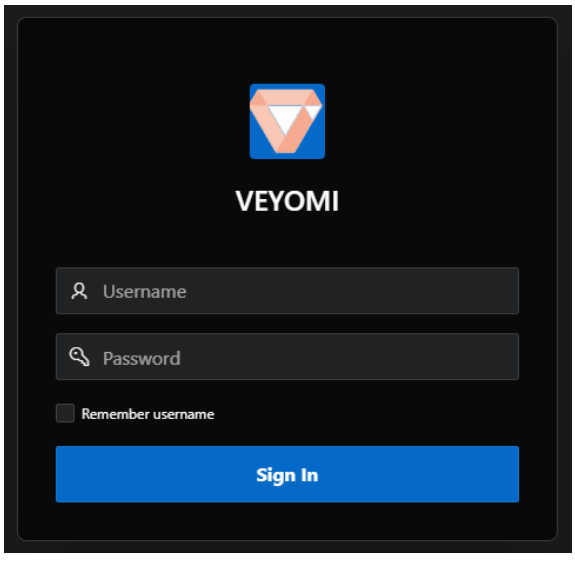


Rysunek 23.15: Pulpit nawigacyjny administratora.

Ważną częścią jest utworzenie formularza logowania. W naszej aplikacji rozważamy umieszczenie tego formularza na określonej stronie, która jest zawsze wywoływana podczas wchodzenia na bezpieczne strony, gdzie konieczne jest jak najszybsze uwierzytelnienie.



Rysunek 23.16: Strona docelowa z logowaniem.



Rysunek 23.17: Strona logowania.

23.7 Skrypty

```
#Trigger on customer sequence.
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;
/

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;
```

```
/

[caption={Function AUTHENTICATE\_USER.
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;

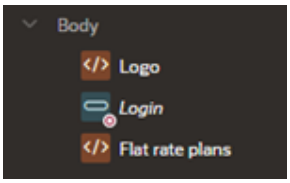
ELSE
APEX_UTIL.SET_AUTHENTICATION_RESULT(1);
return false;
END IF;
APEX_UTIL.SET_AUTHENTICATION_RESULT(7);
return false;
EXCEPTION
WHEN OTHERS THEN
APEX_UTIL.SET_AUTHENTICATION_RESULT(7);
APEX_UTIL.SET_CUSTOM_AUTH_STATUS(sqlerrm);
return false;
END AUTHENTICATE_USER;
/
```

23.8 Tworzenie strony głównej

W zakładce Create a Page można wybierać spośród różnych szablonów, pustych stron lub stron z predefiniowanymi funkcjami, takimi jak Login, About page i Configuration Options. Najpierw należy wypełnić podstawowe informacje, takie jak numer i nazwa strony oraz wybrać model strony, taki jak Normalny, Modalne okno dialogowe lub Szuflada. Można również wypełnić dodatkowe

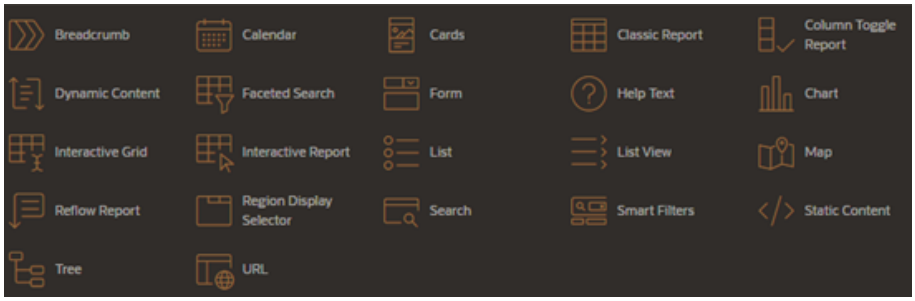
informacje, takie jak ustawienie wpisu nawigacyjnego dla strony (czy będzie to górny wpis, czy zagnieżdżony w istniejącym wpisie) i wybranie ikony wyświetlanej w menu pulpitu nawigacyjnego.

Po kliknięciu przycisku Create Page (Utwórz stronę) wyświetlony zostanie pulpit nawigacyjny umożliwiający edycję informacji i treści strony.Główna zawartość strony (Body) jest podzielona na regiony.



Rysunek 23.18: Regiony ciała.

Każdy region może zawierać HTML lub różne treści, takie jak statyczny HTML lub predefiniowane treści.



Rysunek 23.19: Lista treści.

Our **Login** zawiera przycisk przekierowujący stronę do formularza logowania użytkownika. We właściwości Behaviour przycisku zdefiniowaliśmy Action, która w tej aplikacji została ustawiona jako Redirect to Page. Następnie ustawiliśmy Target na predefiniowaną stronę logowania. Ponieważ nie jest konieczne wyświetlanie tego przycisku, gdy użytkownik jest już zalogowany, musimy ukryć przycisk, ustawiając Typ jako Użytkownik publiczny (użytkownik nie jest uwierzytelniony). Na koniec utworzyliśmy **Plany ryczałtowe** jako statyczny region zawartości zawierający kod HTML, dla którego możemy zdefiniować CSS.



```
11
12  </head>
13  <body>
14  <div id="container">
15      <div class="whole">
16      <div class="type">
17          <p>VEYOMI 100+</p>
18          </div>
19      <div class="plan">
20
21          <div class="header">
22              11<span>€</span>
23              <br>
24          </div>
25          <div class="content">
26              <ul>
27                  <li>100 minutes calls</li>
28                  <li>100 SMS/MMS</li>
29                  <li>10 GB data</li>
30              </ul>
31
32          </div>
33
34          <div class="price">
```

Rysunek 23.20: Kod HTML regionu statycznego.

23.9 Tworzenie strony klienta

Stworzyliśmy tę stronę przy użyciu podobnych kroków, jak podczas tworzenia strony głównej. Po utworzeniu strony kontynuujemy opracowywanie układu strony klienta, która jest podzielona na kilka regionów, takich jak wyświetlanie aktualnego stanu usług, kupowanie dodatków, wyświetlanie faktur i wyświetlanie dzienników.

Obecny stan składa się z dwóch podregionów, które wyświetlają dane liczbowe - aktualną cenę zużytych jednostek ryczałtu i dodatków. Zapytanie SQL służy do uzyskania tych danych.

```
#Zapytanie SQL - Zużyte jednostki stawki ryczałtowej.
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;

#Zapytanie SQL - Zużyte jednostki usług dodatkowych.
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;
```

Kup dodatkowo zawiera pole liczbowe, listę wyboru i przycisk. Możliwe jest ustawienie kwoty i wybranie dodatkowej usługi, którą chcemy kupić. Jest on obsługiwany za pomocą przycisku, który wykonuje proces uruchamiający kod PL/SQL:

```
#Kod PL/SQL do wstawienia dodatkowej usługi..
INSERT INTO CH23_EXTRASERVICE VALUES (
:BUY_TYPE,
(SELECT customerid
FROM CH23_CUSTOMER JOIN CH23_PERSON USING (personid)
WHERE email = :APP_USER),
sysdate,
:BUY_AMOUNT
);
```

Pokaż faktury składa się z przycisku służącego do generowania faktur. W obszarze *Zachowanie* właściwości przycisku, zdefiniowaliśmy *Działanie*, który został ustawiony jako *Przekierowanie do strony w tej aplikacji*. Następnie ustawiamy *Cel* do predefiniowanej strony faktury klienta, która została ustawiona jako modalna. Na tej stronie za pomocą zapytania SQL wyświetlany jest klasyczny raport, który można również wydrukować:

```
#Zapytanie SQL dla raportu faktur.
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;
```

Region **Dziennika** zawiera interaktywny raport do wyświetlania dziennika użytkowania za pomocą zapytania SQL:

```
#Zapytanie SQL - Dziennik użycia.
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;
```

Zakładka **Statystyki** to nowa strona składająca się z trzech regionów, takich jak Minuty, SMS i Dane. Typ regionu jest ustawiony na *Zawartość statyczna*, w tym *Wykres* komponent wyświetlający wykresy z serii. Jego źródło jest definiowane przez zapytanie SQL. Domyślnie kolor słupków wykresu jest jednolity. Można go dostosować w zapytaniu SQL:

```
#Zapytanie SQL - Statystyki.
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;
```

23.10 Tworzenie strony menedżera

Na stronie menedżera użyliśmy podobnych metod i komponentów do tworzenia podstron. Każda podstrona jest tworzona jako nowa strona, taka jak Dodaj klienta, Zarządzaj klientami i Wyświetl klientów.

Strona *Dodaj klienta* zawiera prosty formularz utworzony z komponentów. Proces dodawania nowego klienta jest wykonywany przez kod PL/SQL:

```
#Kod PL/SQL do wstawiania osoby.
INSERT INTO CH23_PERSON (EMAIL, PASSWORD, FIRSTNAME, LASTNAME, ROLE)
VALUES (:CUSTOMER_EMAIL, :CUSTOMER_PASSWORD, :CUSTOMER_FIRSTNAME,
:CUSTOMER_LASTNAME, 'C');
```

Strona *Zarządzanie klientami* składa się z dwóch interaktywnych siatek: Zarządzaj osobami i Zarządzaj bieżącymi klientami. Interaktywne siatki są połączone z procesem edycji i zapisywania zmian. Typ procesu jest ustawiony na *Interaktywny Grid - Automatyczne przetwarzanie wierszy (DML)*.

Strona *Wyświetlanie klientów* zawiera interaktywny raport do wyświetlania danych bez ich edycji. Raport ten jest tworzony przy użyciu zapytania SQL:

```
#Zapytanie SQL - Klienci.
SELECT firstname, lastname, email, address, phone
FROM CH23_PERSON
JOIN CH23_CUSTOMER USING (personid);
```

23.11 Dodatkowe materiały edukacyjne

Dostępne są następujące dodatkowe materiały edukacyjne:

- skrypt do tworzenia i wypełniania tabel
- skrypt do usuwania tabel
- wyeksportowana spakowana aplikacja
- wideo, które pokazuje, jak wygenerować aplikację

Wszystkie dodatkowe materiały edukacyjne są dostępne na stronie Publiczna strona projektu BeeAPEX. Zaloguj się jako gość (hasło nie jest wymagane). Podręcznik znajduje się w sekcji Książki, skrypty w folderze Część 2 > Rozdział23 w sekcji Skrypty, a przewodniki wideo w sekcji Kolekcja przewodników wideo. Materiały do krótkich kursów znajdują się w sekcji Krótkie kursy.

23.11.1 Wyeksportowana aplikacja

Wyeksportowana aplikacja jest spakowana. Instalacja tworzy tabele i wypełnia je danymi. Deinstalacja usuwa wszystkie obiekty bazy danych używane w tej aplikacji.

Spakowana aplikacja jest testowana i zostanie uruchomiona w nowej przestrzeni roboczej, jeśli spełnione zostaną następujące wymagania:

- Dodaj użytkownika APEX przed uruchomieniem aplikacji. Tylko w obszarze roboczym rozwoju i testowania przejdź do Shared Components > Application Access Control > Add User Role Assignment; wprowadź użytkownika APEX i ustaw role użytkownika Administrator, Contributer i Reader. W produkcji konieczna jest konsultacja z wykwalifikowanym personelem przed wdrożeniem.

Jeśli użytkownikowi nie zostanie przyznana odpowiednia rola, zaimportowana aplikacja ulegnie awarii. Konieczne jest wyczyszczenie plików cookie przeglądarki internetowej (i.e. Firefox: Settings > Cookies and Site Data > Manage Data) po awarii aplikacji z powodu niespełnionych wymagań.

23.11.2 Przewodniki wideo

Przewodnik wideo pokazuje wszystkie etapy tworzenia aplikacji.

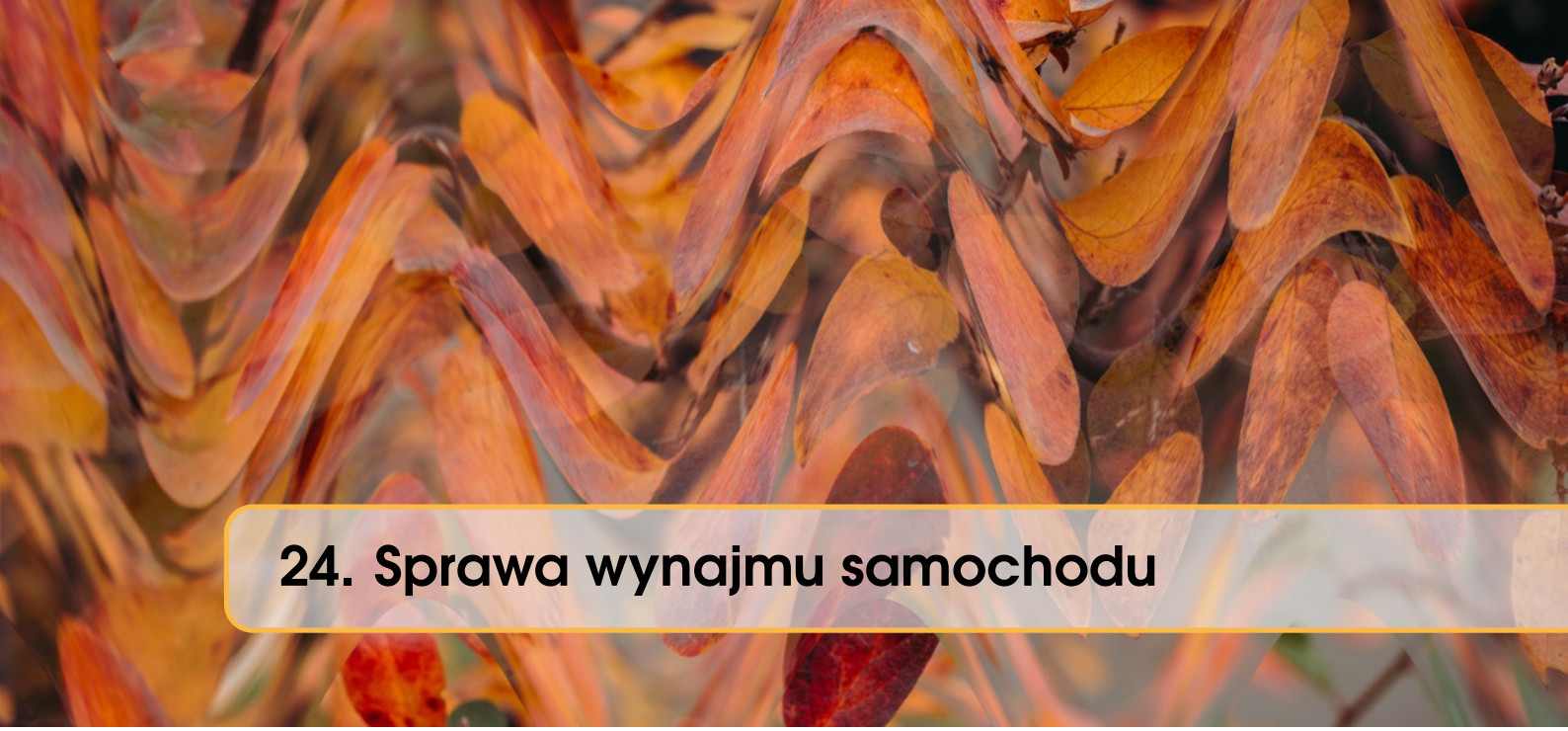
23.12 Pytania

1. Która podstrona służy do tworzenia nowego schematu logowania?
2. Wymień co najmniej dwie metody uwierzytelniania na potrzeby zarządzania użytkownikami.
3. W trakcie wdrożenia różne elementy były przywoływane w instrukcjach Select. Jak można zidentyfikować te elementy?

23.13 Odpowiedzi

1. W narzędziu Oracle APEX wybieramy metodę uwierzytelniania - Authentication Scheme - na podstronie Shared Components => Authentication Schemes.
2. Istnieje wiele metod, które mogą być używane do uwierzytelniania użytkowników, takich jak serwer LDAP, konto bazy danych, konto Oracle APEX itp.

3. Pozycje w instrukcjach Select są przywoływane za pomocą dwukropka na początku nazwy pozycji.



24. Sprawa wynajmu samochodu

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

24.1 Biznesowe spojrzenie na sprawę

Wynajem samochodu może być wygodnym sposobem na odkrywanie nowych miejsc docelowych, ale istnieje kilka potencjalnych pułapek, których należy być świadomym. Ważne jest, aby zrozumieć, że wynajem samochodu jest bardziej podobny do rezerwacji pokoju hotelowego niż rezerwacji biletu lotniczego. Podczas wynajmu samochodu istnieją dwa rodzaje opłat: te, które płacisz przy odbiorze samochodu, oraz ceny opłacone z góry.

Dla osób wynajmujących po raz pierwszy, zwykle najlepiej jest zarezerwować stawkę Pay Later, która pozwala na większą elastyczność w przypadku konieczności zmiany planów.Ten rodzaj stawki wynajmu zazwyczaj nie wiąże się z karą za anulowanie, co może być pomocne w przypadku wystąpienia nieprzewidzianych okoliczności.

Aby uniknąć dodatkowych opłat, ważne jest, aby dokładnie zapoznać się z umową najmu i wszelkimi dodatkowymi opłatami, które mogą zostać poniesione, takimi jak ubezpieczenie lub dodatkowe opłaty za kierowcę. Ponadto należy dokładnie sprawdzić pojazd przed opuszczeniem wypożyczalni i natychmiast zgłosić wszelkie uszkodzenia firmie wynajmującej, aby uniknąć późniejszej odpowiedzialności.

Rozumiejąc proces wynajmu i potencjalne opłaty, możesz zapewnić sobie płynne i bezstresowe doświadczenie wynajmu i w pełni cieszyć się swobodą i elastycznością, jaką może zaoferować wynajem samochodu podczas podróży.

24.2 Definicja problemu

Wynajem samochodu może wydawać się zniechęcający ze względu na wszystkie zasady i warunki, ale istnieją one nie bez powodu. Fabrycznie nowy samochód kosztuje średnio około 36 000 euro, więc wynajmując samochód, zasadniczo płacisz niewielki ułamek jego całkowitej wartości za tymczasowe użytkowanie pojazdu.

Chociaż posiadanie odpowiedniego ubezpieczenia wynajmu samochodu na wypadek wypadku lub nieprzewidzianych okoliczności ma kluczowe znaczenie, większość doświadczeń związanych z wynajmem jest wolna od incydentów. Jednak nadal ważne jest, aby uważnie przeczytać umowę najmu, zrozumieć warunki i sprawdzić pojazd pod kątem wszelkich uszkodzeń przed jego przyjęciem, aby uniknąć potencjalnych problemów.

Postępując zgodnie z tymi wskazówkami, możesz mieć bezpieczne i przyjemne doświadczenie

wynajmu bez żadnych nieoczekiwanych niespodzianek. I pamiętaj, że zasady i warunki mają na celu ochronę zarówno Ciebie, jak i wypożyczalni samochodów, więc zawsze najlepiej jest ich przestrzegać, aby wynajem przebiegał sprawnie.

24.3 Przypadki użycia

24.3.1 Narracyjny opis przypadku użycia

Celem tego projektu jest opracowanie funkcjonalnej aplikacji symulującej działanie wypożyczalni samochodów. Aplikacja będzie wykorzystywać platformę Oracle APEX do szczegółowego zarządzania i przechowywania informacji o klientach, samochodach i wypożyczalniach.

Aplikacja ma być przyjazna dla właściciela wypożyczalni samochodów, umożliwiając mu zarządzanie wszystkimi niezbędnymi informacjami, w tym wypożyczalniami, dostępnymi samochodami do wynajęcia i danymi klientów. Użytkownicy będą mogli dokonywać wynajmu, wybierając samochód na podstawie kryteriów takich jak marka, model, rocznik i rodzaj paliwa, a wszystkie niezbędne szczegóły wymagane do ukończenia wynajmu zostaną wyświetlone.

Ponadto aplikacja będzie wyświetlać dane, takie jak bieżące wypożyczenia, dostępne samochody do wynajęcia i szczegółowe dane klienta. Bieżące wypożyczenia będą wyświetlane użytkownikowi w tabelach, w zależności od tego, w której zakładce się znajduje.

Ogólnie rzecz biorąc, aplikacja ta zapewni wydajny i zorganizowany system zarządzania wypożyczalnią samochodów, ze szczegółowymi informacjami dostępnymi po kliknięciu przycisku. Przyjazny dla użytkownika interfejs i wszechstronne narzędzia do zarządzania danymi sprawią, że prowadzenie wypożyczalni samochodów będzie prostsze i bardziej wydajne.

24.3.2 Opis częściowo ustrukturyzowany

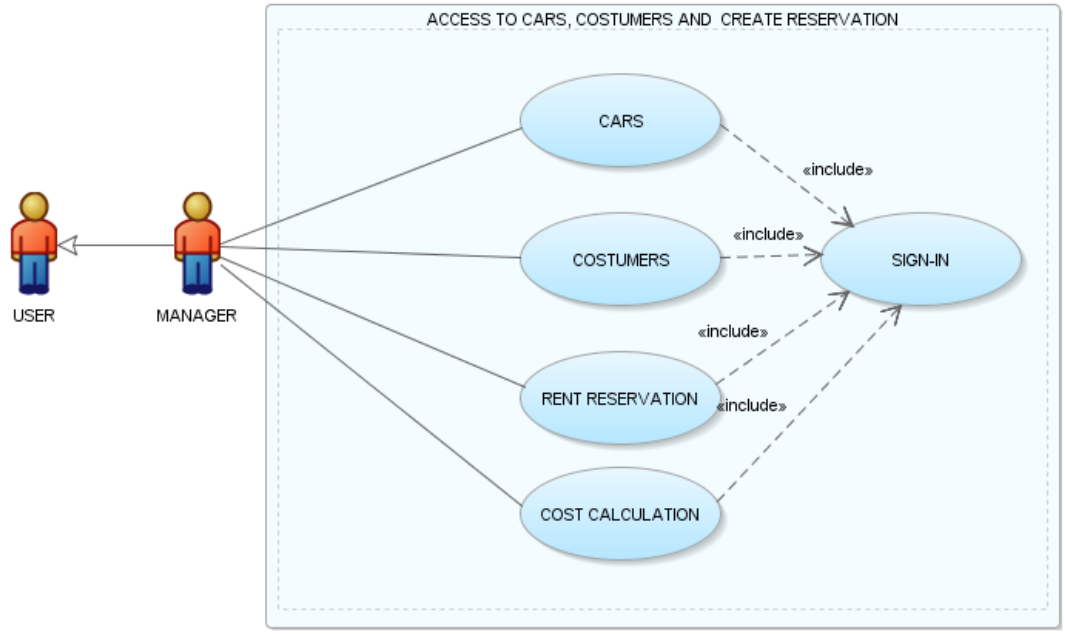
Celem tego projektu jest opracowanie przyjaznej dla użytkownika aplikacji wykorzystującej Oracle APEX do zarządzania szczegółowymi informacjami o klientach, samochodach i wypożyczalniach dla wypożyczalni samochodów. Aplikacja umożliwi użytkownikom dokonywanie wynajmu w oparciu o określone kryteria, wyświetlanie niezbędnych informacji o wynajmie oraz zapewni kompleksowe narzędzia do zarządzania danymi w celu wyświetlania bieżących wynajmów, dostępnych samochodów do wynajęcia i danych klientów. Zapewniając wydajny i zorganizowany system do zarządzania wypożyczalnią samochodów, aplikacja ta sprawi, że prowadzenie wypożyczalni samochodów będzie prostsze i bardziej wydajne tabela 24.1.

24.3.3 Diagram przypadków użycia

Powyższa historia jest przedstawiona na diagramie przypadków użycia (patrz na 24.1).

Tablica 24.1: Opis przypadku użycia: dostęp do samochodów, klientów i rezerwacji wynajmu samochodów

Słowo kluczowe	**Wartość**
ID:	*Ch24-01*
Tytuł:	*Dostęp do listy samochodów i klientów, możliwość dodawania i usuwania danych oraz tworzenia nowych rezerwacji wynajmu samochodów poprzez wybór preferowanych dat i obliczanie kosztów.*
Opis:	*Użytkownik będzie korzystał z aplikacji APEX, aby uzyskać dostęp i przeglądać samochody należące do firmy, a także wszystkich klientów, którzy wcześniej dokonali rezerwacji. Zarówno samochody, jak i klienci są opisani przez zestaw atrybutów. Ponadto użytkownik będzie miał możliwość tworzenia, aktualizowania i usuwania danych dla obu kategorii, co pozwoli na efektywne zarządzanie systemem wynajmu samochodów. Ponadto użytkownik może utworzyć nową rezerwację wynajmu samochodu, dodając samochód, klienta i daty. System oblicza całkowity koszt rezerwacji na podstawie wybranych dat i odpowiednio aktualizuje wyniki. Raport końcowy przedstawia szczegóły rezerwacji, w tym samochód, klienta, daty wynajmu i całkowity koszt.*
Główny aktor:	*Administrator użytkownika*
Warunki wstępne:	Użytkownik posiada konto administratora dla instancji APEX.
Warunki końcowe:	Po aktualizacji i wybraniu samochodu i klienta, użytkownik będzie mógł utworzyć rezerwację wynajmu samochodu w instancji APEX. Po zakończeniu rezerwacji wynajmu samochodu i obliczeniu całkowitego kosztu.
Główny	Scenariusz
Scenariusz sukcesu:	1. Otwórz przeglądarkę internetową i zaloguj się do aplikacji wypożyczalni samochodów. 2. Wybierz pozycję menu lub nawigację po stronie z etykietą **samochody** lub **klienci**. 3. Aby dodać samochód, kliknij przycisk "Utwórz", a w przypadku klientów zrób to samo. 4. Aby edytować wpis dotyczący samochodu lub klienta, kliknij symbol ołówka powiązany z danym wpisem. 5. Aby dodać lub zaktualizować dane, kliknij przycisk Źastosuj zmiany". Aby usunąć dane, kliknij przycisk "Usuń". 6. Wybierz pozycję menu lub nawigację na stronie **wynajem samochodu**. 7. Aby utworzyć nowy czynsz, kliknij **Utwórz**. 8. Aby dodać samochód do rezerwacji, kliknij **carid** i wybierz z listy dostępnych samochodów. 9. Aby dodać klienta do rezerwacji samochodu, kliknij **custid** i wybierz z listy. 10. Aby ustawić daty rezerwacji, kliknij **data wynajmu** i wybierz preferowaną datę rozpoczęcia rezerwacji. 11. Następnie kliknij pole **data zwrotu** i wybierz datę zakończenia wynajmu samochodu. 12. Po zwrocie samochodu całkowity koszt rezerwacji jest obliczany na podstawie liczby dni wynajmu i wszelkich dodatkowych opłat. W przypadku uszkodzenia samochodu zostanie doliczona dodatkowa opłata na pokrycie kosztów naprawy. 13. Ostateczny koszt jest przedstawiany klientowi wraz ze szczegółowym zestawieniem opłat w raporcie z wynajmu.
Rozszerzenia:	*Brak*
Częstotliwość użytkowania:	*Liczba rezerwacji dziennie różni się w zależności od sezonu turystycznego, ale średnio otrzymujemy około pięciu rezerwacji dziennie.*
Status:	Zakończony
Właściciel:	Użytkownik



Rysunek 24.1: Diagram przypadków użycia.

24.4 Model danych

24.4.1 Narracyjny opis modelu danych

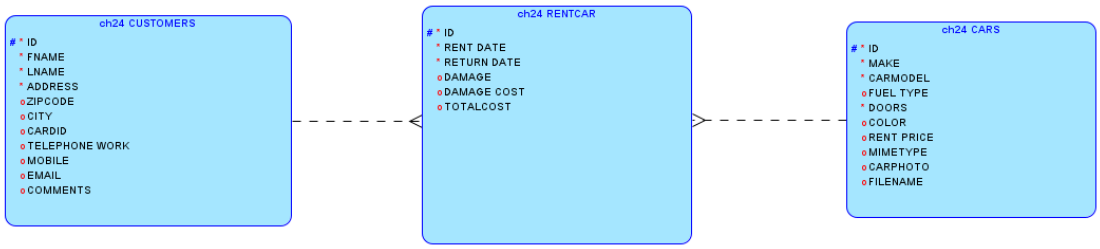
Projekt edukacyjny jest realizowany przy użyciu trzech tabel do reprezentowania zaangażowanych podmiotów: tabeli samochodów, tabeli klientów i tabeli wynajmu samochodów. Tabele te zawierają wszystkie niezbędne atrybuty i dane potrzebne do zarządzania procesem wynajmu samochodów za pośrednictwem aplikacji APEX.

1. Tabela samochodów
   - **id** numer identyfikacyjny i klucz podstawowy,
   - **marka** producent samochodów,
   - **model samochodu** model samochodu,
   - **paliwo** typ benzin, diesel, LNG lub elektryczny
   - **drzwi** ile drzwi ma samochód
   - **color** kolor samochodu
   - **wynajem** cena koszt dziennego wynajmu konkretnego samochodu
   - **auto-foto** zdjęcie samochodu
2. Tabela klientów
   - **id** tożsamość numer i klucz główny klienta,
   - **fname** imię klienta,
   - **lname** nazwisko klienta,
   - **address** adres klienta,
   - **zipcode** kod pocztowy adresu,
   - **city** miasto klienta,
   - **cardid** dokument tożsamości lub dowód osobisty klienta
   - **telephone-work** telefon do pracy klienta
   - **mobile** telefon komórkowy
   - **email** adres e-mail klienta
   - **comments** komentarze związane z zachowaniem danej osoby jako klienta

3. Tabela wynajmu samochodów
    - **id** tożsamość numer i klucz podstawowy tabela wypożyczeń
    - **carid id** pole połączenia między samochodem a tabelą wypożyczalni
    - **custid id** pole połączenia między tabelą klientów a tabelą wypożyczeń
    - **rent date** data wynajmu
    - **return date** data zwrotu,
    - **damage** opis uszkodzenia samochodu jeśli wystąpi
    - **damage cost** koszt uszkodzenia
    - **total cost** całkowity koszt wynajmu

24.4.2 Logiczny model danych

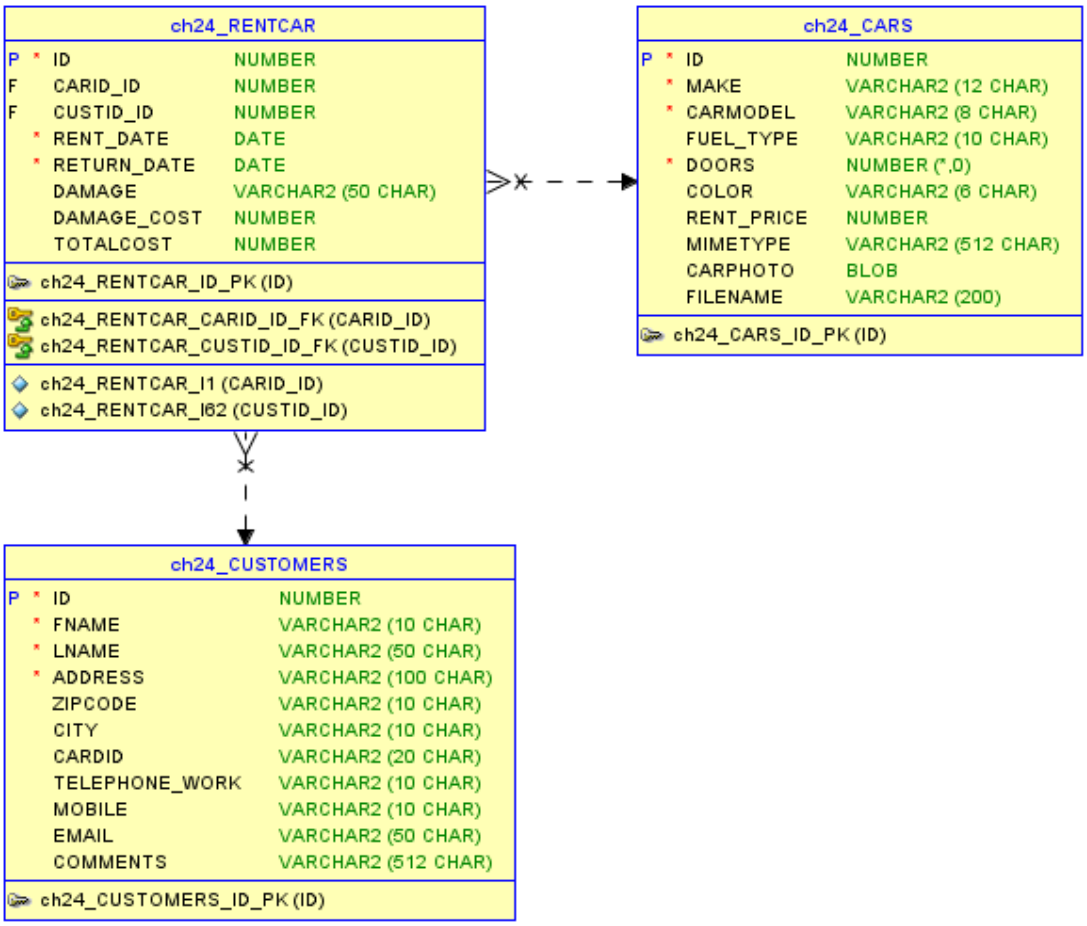
Logiczny model danych przedstawiony na rysunku 24.2.



Rysunek 24.2: Model logiczny projektu wypożyczalni samochodów

24.4.3 Relacyjny model danych

Relacyjny model danych przedstawiony na rysunku 24.3.



Rysunek 24.3: Model relacyjny projektu wynajmu samochodów

24.4.4 Skrypt SQL

Tabele, które utworzyliśmy dla naszej aplikacji to CH24_CARS, CH24_CUSTOMERS i CH24_RENTCAR. Pola każdej tabeli zostały opisane poniżej:

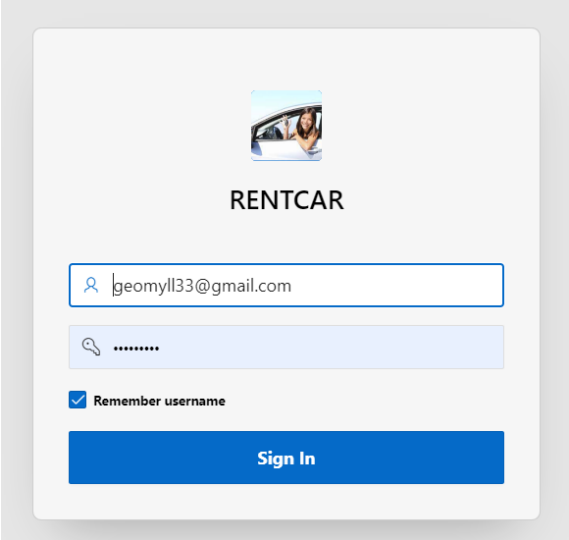
```
drop table  ch24_cars;
drop table  ch24_customers;
drop table  ch24_rentcar;
-- create tables
create table ch24_cars (
    id                  number generated by default on null as identity
                        constraint ch24_cars_id_pk primary key,
    make                varchar2(12 char) 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);
```

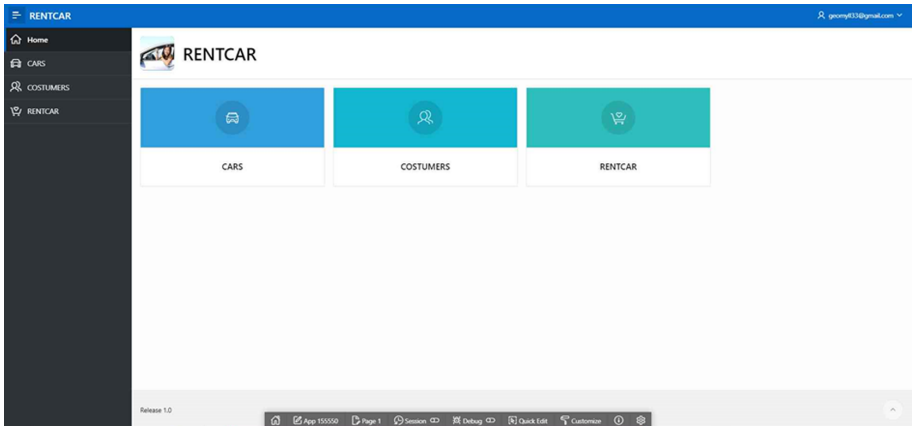
24.5 Interfejsy aplikacji

Użytkownik, który jest właścicielem wypożyczalni samochodów, musi podać swoje dane uwierzytelniające, aby uzyskać dostęp do aplikacji. Aplikacja korzysta z uwierzytelniania kont Application Express. Poświadczenia właściciela są takie same jak poświadczenia WORKSPACE pokazane na rysunku 24.4.



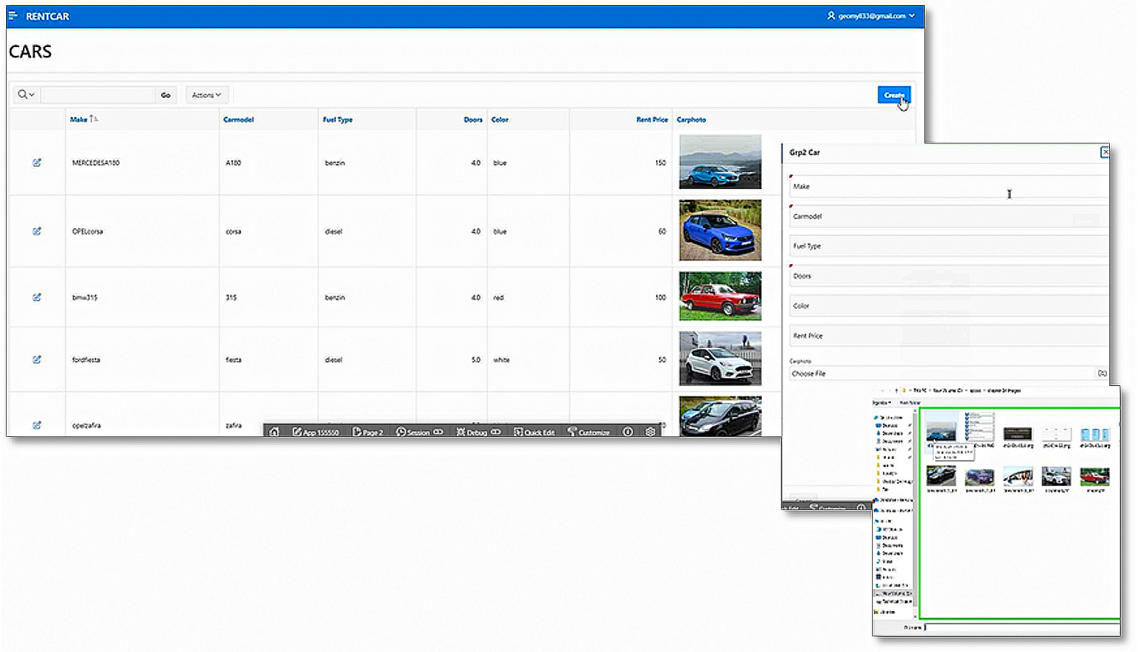
Rysunek 24.4: Zaloguj się do aplikacji.

Aplikacja symuluje działanie wypożyczalni samochodów i obejmuje zarządzanie klientami, samochodami i wypożyczalniami. Zapewnia szczegółowe informacje o tych podmiotach za pomocą przyjaznej aplikacji internetowej opartej na platformie Oracle APEX, jak pokazano na rysunku 24.5.



Rysunek 24.5: Strona główna aplikacji.

Szablon na rysunku 24.6 jest przeznaczony do zarządzania informacjami związanymi z samochodami w wypożyczalni samochodów. Użytkownik, który jest właścicielem sklepu, może przechowywać i zarządzać wszystkimi niezbędnymi informacjami o samochodach, takimi jak marka, model, rok, rodzaj paliwa, a także przesłać zdjęcie samochodu, aby pokazać jego stan.

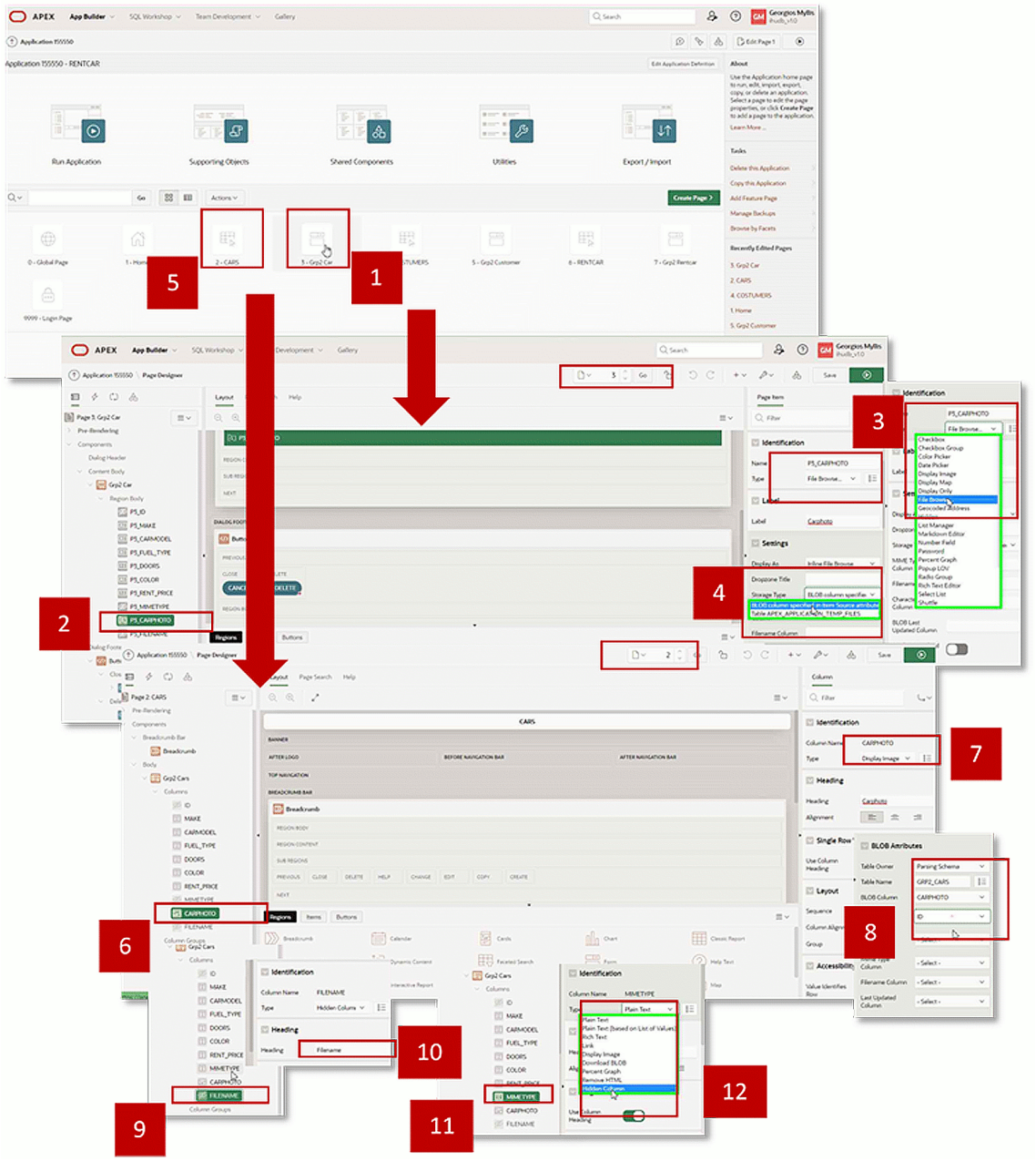


Rysunek 24.6: Szablon samochodu

Na poniższym rysunku 24.7 wyjaśniono, jak dodać kolumnę do formularza samochodu, która umożliwi wyświetlanie zdjęć samochodów. Aby to osiągnąć, potrzebne są dwie kolumny pomocnicze do zapisywania informacji w tabeli, jedna dla mime i jedna dla nazwy pliku. Kolumna, która będzie przechowywać zdjęcie, powinna być ustawiona na typ przechowywania BLOB. W naszym przypadku nazwa kolumny to **CARPHOTO**.

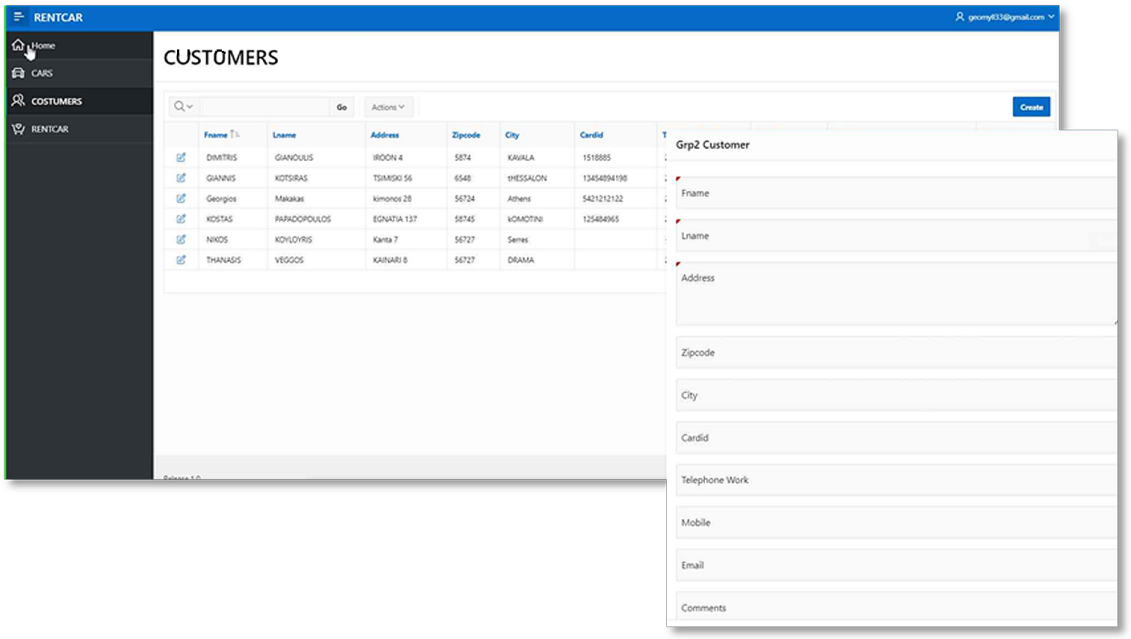
W kreatorze stron aplikacji należy przejść do raportu dla szablonu CAR i skonfigurować kolumnę **CARPHOTO** jako przeglądarkę plików. Następnie należy zmienić ustawienia typu przechowywania na "BLOB column specifier in item source attributes".

Następnie należy przejść do formularza szablonu CAR i skonfigurować kolumnę **CARPHOTO** jako obraz Display. Skonfiguruj atrybuty BLOB, łącząc tabelę **GRP2-CARS**, kolumnę **CARPHOTO** i klucz podstawowy **ID**. Na koniec kolumny **MIMETYPE** i **FILENAME** powinny być ustawione jako ukryte.



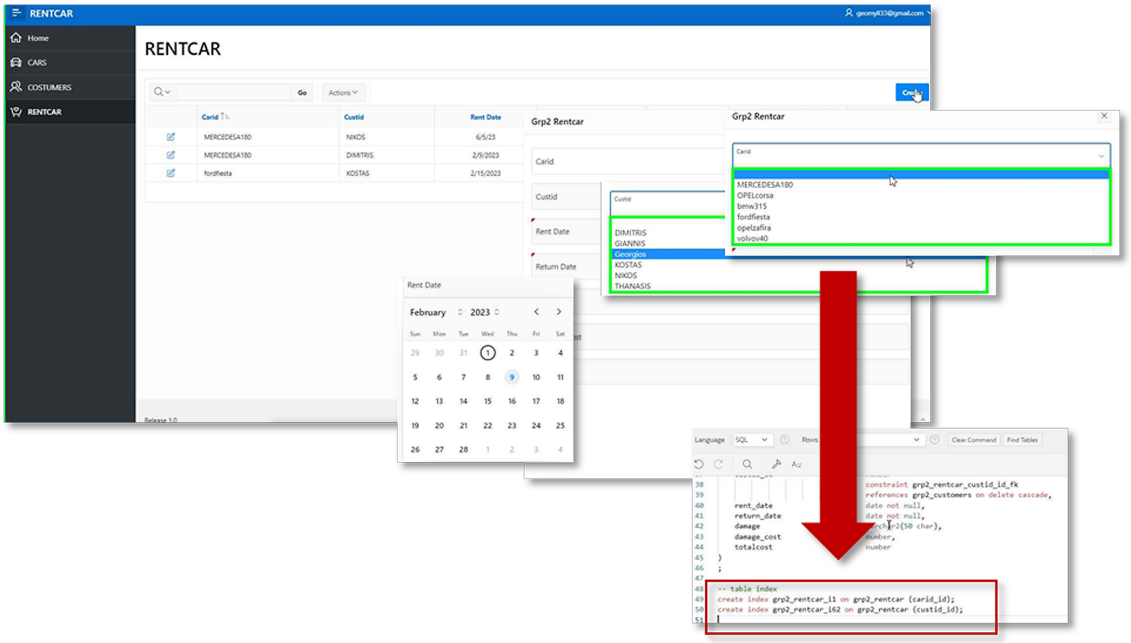
Rysunek 24.7: Jak stworzyć kolumnę ze zdjęciami.

Podobnie można zarządzać danymi związanymi z klientami (patrz rysunek 24.8).



Rysunek 24.8: Dane klientów.

W tabeli wynajmu, jak pokazano na rysunku 24.9, użytkownik może wybrać samochód na podstawie kryteriów takich jak producent, model i rok. Wypożyczenie jest następnie łączone z klientem i wyświetlane są wszystkie niezbędne szczegóły potrzebne do wypożyczenia, takie jak samochód, klient, data rozpoczęcia i zakończenia wypożyczenia oraz całkowity koszt. Użytkownik może również dodać dodatkowy koszt, jeśli samochód został uszkodzony po zwrocie. Po zakończeniu wynajmu całkowity koszt zostanie obliczony i przedstawiony w raporcie końcowym.



Rysunek 24.9: Szablon wypożyczalni samochodów.

24.6 Dodatkowe materiały edukacyjne

Dostępne są następujące dodatkowe materiały edukacyjne:

- wyeksportowana aplikacja
- przewodniki wideo

Wszystkie dodatkowe materiały edukacyjne są dostępne na stronie publiczna strona projektu BeeAPEX. Zaloguj się jako gość (hasło nie jest wymagane). Podręcznik można znaleźć w sekcji Książki, skrypty w folderze Część 2 > Rozdział 24 w sekcji Skrypty, a przewodniki wideo w sekcji Kolekcja przewodników wideo. Materiały do krótkich kursów znajdują się w sekcji Krótkie kursy.

24.6.1 Eksportowane aplikacje

Wyeksportowana aplikacja jest spakowana. Instalacja tworzy tabele, indeks, funkcję, procedurę i wyzwalacz, a także wypełnia dane. Deinstalacja usuwa wszystkie obiekty bazy danych używane w tej aplikacji.

24.6.2 Przewodniki wideo

Przewodnik wideo pokazuje każdy krok w tworzeniu aplikacji.

24.7 Pytania

1. Jak dodać ikonę do logo aplikacji po jej utworzeniu?
2. Jak dodać obrazy do stron raportów?

24.8 Odpowiedzi

1. W środowisku aplikacji wybierz **Komponenty współdzielone**, a następnie wybierz **Atrybuty interfejsu użytkownika**, kliknij **Edytuj** w wyskakującym oknie i dodaj preferowany obraz.
2. Najpierw należy ustawić typ danych kolumny jako BLOB. Następnie w kreatorze stron aplikacji dla szablonu raportu skonfiguruj kolumnę jako przeglądarkę plików i zmień Storage Type na **określnik kolumny BLOB w elemencie Source Attributes**. Następnie przejdź do formularza tego szablonu w środowisku App Builder i skonfiguruj kolumnę jako **Wyświetl obraz**. Następnie skonfiguruj atrybuty BLOB, łącząc tabelę, kolumnę i identyfikator klucza podstawowego, który jest powiązany z tą kolumną.

Bibliography

Articles

[1] Hill Kim. “Altruistic cooperation during foraging by the Ache, and the evolved human predisposition to cooperate”. W: *Human Nature* 13 (mar. 2002), strony 105–128. DOI: 10.1007/s12110-002-1016-3 (cytowane na stronie 189).

Books

[2] Roy Fielding i Richard N. Taylor. *Principled design of the modern Web architecture*. Redagował Mehdi Jazayeri andAlexander L. Wolf Carlo Ghezzi. Association for Computing Machinery, 2010. ISBN: 978-1-58113-206-9. DOI: 10.1145/337180.337228 (cytowane na stronach 104, 105).

[3] Dariusz Jemielniak i Aleksandra Przegalinska. *Collaborative society*. MIT Press, 2020 (cytowane na stronie 189).

[4] OMG. *Business Process Model and Notation (BPMN), Version 2.0*. 2011. URL: http://www.omg.org/spec/BPMN/2.0 (cytowane na stronach 270, 271).

[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 (cytowane na stronie 269).

[6] A. Osterwalder i Y. Pigneur. *Business model generation: a handbook for visionaries, game changers, and challengers*. Tom 1. John Wiley i Sons, 2010 (cytowane na stronie 38).

Indeks

abstraction level, 50
access control, 86
APEX docker, 45
application builder, 83
application logic, 87
application programming interface, API, 105
application report, 103
attribute, 51
AutoREST, 109

bill-of-material, 323
Biznesowe spojrzenie na sprawę, 403
biznesowy pogląd na sprawe, 229
biznesowy pogląd na sprawę, 241, 249, 257, 269, 339, 359, 379
business view of the case, 301, 309, 323

calendar, 87
cardinality, 51
chart, 87
column, 54
computation, 87
create application, 87
create application wizard, 84
create page wizard, 86
CRUD operation, 108

data manipulation, 68
data model, 50
Data Modeler, 61
data structure, 50
data type, 51
data workshop, 94, 100
database, 50
database management system, 50
database system, 49
DB, 50
DB layer, 50, 78
DB schema, 54
DBMS, 50
DBS, 49
DDL, 60
definicja problem, 249
Definicja problemu, 403
definicja problemu, 241, 257, 273, 301, 309, 323, 339, 360, 379
definicja problemu , 229
Diagram przypadków użycia, 310, 404
diagram przypadków użycia, 302, 327, 380
diagram przypadków użycia , 230, 250
DML, 68
domain, 51
DQL, 70

entity, 51
entity instance, 54
entity relationship diagram, ERD, 50, 51
ER, 50
ER diagram, 50, 51
export data, 100
export wizard, 100

first normal form, 1NF, 59
Flows dla APEX, 272
foreign key, 57

HTTP request, 105, 108
HTTP response, 106
https://apex.oracle.com, 44

import data, 94
interactive report, 87
interfejs aplikacji, 236
Interfejsy aplikacji, 409
interfejsy aplikacji, 246, 253, 263, 283, 305, 316, 333, 346, 368, 386
intranet, 229
item, 87

Jak przygotować bazę danych?, 49
Jak się poruszać w APEX?, 81
Jak skorzystać z galerii aplikacji i wtyczek?, 201
Jak uruchomic Oracle APEX?, 35
Jak współpracować w zespole?, 189
Jak wygenerować pierwszy szkic wniosku?, 113
Jak wymieniać dane w APEX?, 93
Jak zarządzać formualrzami?, 159

Jak zarządzać menu?, 185
Jak zarządzać pakietowanymi i wielojęzycznymi aplikacjami?, 209
Jak zarządzać raportami?, 137

list, 87
list of values, LOV, 214
load wizard, 96
logical data model, 50
logical model, 50
logiczny model danych, 234, 244, 252, 262, 279, 304, 313, 328, 340, 365, 407
logiczny model danych , 382

Model danych, 406
model danych, 244, 252, 258, 279, 302, 312, 327, 340, 364, 382
model danych , 234
model przepływu pracy, 276

normalization, 59

object browser, 67, 69, 101
on-premise, 43
operation CRUD, 105
Oracle Academy, 47
ORACLE APEX, 81
Oracle Cloud Infrastructure, 47
Oracle REST Data Service, ORDS, 105, 106

packaged application, 209
page, 87, 90
page creation, 86
page designer, 89
page type, 87
PL/SQL, 210
planowanie godzin pracy biura, 359
proces tworzenia aplikacji internetowych, 81
przechowywane funkcje, 282
Przykładowe aplikacje i aplikacje startowe, 201
Przypadki użycia, 404
przypadki użycia, 242, 250, 258, 273, 302, 310, 324, 339, 360, 379
przypadki użycia , 230

query builder, 75
Quick SQL, 63, 69, 121, 209
QuickSQL i skrypt SQL Script, 343

RDB, 54
RDB schema, 54, 58
RDB schema generation, 55
RDBS schema management, 60
region, 87
Relacyjny model danych, 328
relacyjny model danych, 246, 262, 280, 304, 342, 407
relacyjny model danych , 235, 253
relation, 54
relational data model, 313, 365, 382
relational DB schema, 54
relational model, 50, 54
relational model generation, 56
relationship, 51
representational state transfer, REST, 104
resource handler, 108
resource module, 108
REST, 104
RESTful access, 106
RESTful service, 104

Schemat przypadków użycia, 340
schemat przypadków użycia, 258, 274, 361
schemat przypadków użycia , 242
second normal form, 2NF, 59
sekwencja , 282
skill level, 42
SQL aggregation, 74
SQL ALTER TABLE, 67
SQL COUNT, 74
SQL CREATE TABLE, 66
SQL data definition language, 65
SQL data definition language, SQL-DDL, 60
SQL data manipulation language, SQL-DML, 68
SQL data query language, SQL-DQL, 70
SQL DELETE, 69
SQL Developer Data Modeler, 50
SQL DROP TABLE, 66
SQL INSERT, 68
SQL JOIN, 73
SQL ORDER BY, 71, 74
SQL Script, 122
SQL SELECT, 71, 102
SQL SUM, 74
SQL TO_CHAR(), 72
SQL UPDATE, 68
SQL WHERE clause, 68, 72
SQL workshop, 67
SQL-DDL, 60, 68, 209
SQL-DQL, 70
stateless REST, 105
structured query language, SQL, 60

system małych innowacji, 257
szablon zasobów, 108

table, 54
third normal form, 3NF, 60

use case model, UC, 39
user authorisation, 249
user feedback, 86
usługi telekomunikacyjne, 379

validation, 87
Virtual Box Appliance, 44

web application, 49, 90
wiele języków, 221
wielojęzyczny, 221
wymiany danych, 93
wynajem samochodów, 403

zarządzanie procesami biznesowymi , 269