Funded by the European Union

BEE WITH APEX

E-kniha
Nízkokódové programovanie s APEX

Praktická príručka na zlepšenie zamestnateľnosti

Projektová skupina BeeAPEX



Copyright © 2023 BeeAPEX Team

PUBLIKOVALO SOFTWARE QUALITY AND TESTING LAB, UM FOV

BEEAPEX.EU

Licencované pod Creative Commons Attribution-NonCommercial 4.0 License (“Licencia”). Tento súbor môžete používať len v súlade s Licenciou. Kópiu Licencie môžete získať na adrese https://creativecommons.org/licenses/by-nc-sa/4.0. Pokiaľ to nevyžaduje platný zákon alebo to nie je písomne odsúhlasené, softvér distribuovaný na základe licencie sa distribuuje tak, “AKO JE”, BEZ AKÝCHKOĽVEK ZÁRUK ALEBO PODMIENOK, či už výslovných alebo predpokladaných. Pozrite si Licenciu, kde nájdete konkrétne znenie oprávnení a obmedzení v rámci Licencie.

1st edícia, 7. decembra 2023

Obsah

Poďakovanie 27

Predslov 29

I Ako na to v APEX

1 Ako spustiť Oracle APEX? 35

Vjeran Strahonja, Dijana Oreški . . . . . .

1.1 Čo je Oracle APEX a na čo slúži? 35

1.1.1 Čo je Oracle APEX? 35

1.1.2 A čo je aplikácia? 35

1.1.3 Ako môže Oracle APEX pomôcť pri vývoji aplikácií? 35

1.1.4 Cyklus vývoja aplikácie 36

1.2 Ako spustiť Oracle APEX? 42

1.2.1 Aká je úroveň vašich zručností? 42

1.2.2 Použitie lokálnej inštancie APEX 43

1.2.3 https://apex.oracle.com 43

1.2.4 Zariadenie Virtual Box / Virtual Machine 44

1.2.5 APEX docker 44

1.2.6 APEX inštancia v službe Oracle Cloud Infrastructure 46

1.2.7 APEX inštancia v Oracle Academy 47

1.3 Otázky 47

1.4 Odpovede 47

2 Ako pripraviť databázu? 49

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

2.1 Ako modelovať dáta 50

2.1.1 Logický model 50

2.1.2 Relačný model 54

2.1.3 Normalizácia RDB-schémy 59

2.2 Ako spravovať RDBS-schému – SQL-DDL . . . 60
2.2.1 Automatické generovanie tabuliek pomocou ODM . . . 61
2.2.2 Automatické generovanie tabuliek pomocou Quick SQL . . . 63
2.2.3 Manuálne vytváranie tabuliek pomocou SQL-DDL . . . 65
2.2.4 Manuálna správa tabuliek pomocou ORACLE Object Browser . . . 67
2.3 Ako manipulovať s dátami – SQL-DML . . . 68
2.3.1 Manipulácia s údajmi pomocou SQL-DML . . . 68
2.3.2 Manipulácia s údajmi pomocou ORACLE Object Browser ("OB") . . . 69
2.3.3 Manipulácia s údajmi pomocou Quick SQL . . . 69
2.4 Ako získavať údaje – SQL-DQL . . . 70
2.4.1 Dotazovanie údajov pomocou SQL-DQL . . . 70
2.4.2 Dotazovanie údajov pomocou ORACLE Query Builder ("QB") . . . 75
2.5 Vytvorenie DB-vrstvy – The Big Picture . . . 78
2.6 Otázky . . . 78
2.7 Odpovede . . . 79

3 Ako sa pohybovať v nástroji APEX? . . . 81
E. Kapsammer, W. Retschitzegger, W. Schwinger . . .

3.1 Proces vývoja webových aplikácií pomocou APEX . . . 81
3.2 Start Menu v APEX-e . . . 82
3.3 Prehľad nástroja App Builder – vytvorenie a správa Aplikácie . . . 83
3.4 Vytvorenie aplikácie – tri prípady použitia . . . 84
3.5 Vytvorenie aplikácie – vlastnosti, stránky, funkcie a nastavenia . . . 85
3.6 Špecifikácia stránok . . . 87
3.7 Udržiavanie a upravovanie stránky – Page Designer . . . 89
3.8 Spustiť aplikáciu alebo stránku . . . 89

4 Ako si vymieňať dáta v APEX-e? . . . 93
E. Kapsammer, W. Retschitzegger, W. Schwinger . . .

4.1 Import a export údajov pomocou "Data Workshop" . . . 94
4.1.1 Import údajov . . . 94
4.1.2 Kroky importu . . . 96
4.1.3 Export údajov . . . 100
4.2 Import a export údajov pomocou "Object Browser" . . . 101
4.3 Export výsledku SQL-príkazu . . . 102
4.4 Export údajov z reportu aplikácie . . . 103
4.5 Povolenie výmeny údajov so službou RESTful Servis . . . 104
4.5.1 Architektonický vzor REST . . . 105
4.5.2 Povolenie DB-schémy pre RESTful prístup . . . 106
4.5.3 Moduly Zdrojov | Templates | Handlers . . . 107
4.5.4 Používanie "AutoREST" namiesto manuálneho definovania zdrojov . . . 109
4.6 Otázky . . . 111
4.7 Odpovede . . . 111

5 Ako vygenerovať prvý návrh aplikácie? . . . 113
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis

5.1 Prečo podniky potrebujú aplikácie . . . 113
5.1.1 Nastavenie prostredia ORACLE APEX . . . 113
5.1.2 Príklad vývoja SQL workshopu . . . 117
5.1.3 Quick SQL . . . 119
5.1.4 SQL Script . . . 121
5.2 Dátovo riadená časť aplikácie . . . 123
5.3 Administrácia aplikácie . . . 127
5.4 Riadenie prístupu . . . 127
5.5 Doplnkový učebný materiál . . . 133
5.5.1 Exportované aplikácie . . . 133
5.5.2 Video sprievodcovia . . . 133
5.6 Otázky . . . 133
5.7 Odpovede . . . 134

6 Ako spravovať reporty? . . . 135
Zuzana Žillová, Erik Malina, Matej Grochal, Andrej Staník, Andrea Meleková, Michal Kvet, Miroslav Potočár

6.1 Report . . . 135
6.2 Klasická zostava . . . 144
6.3 Zostava s možnosťou prepínania stĺpcov . . . 146
6.4 Interaktívna zostava . . . 146
6.5 Otázky . . . 155
6.6 Answers . . . 155

7 Ako spravovať formuláre? . . . 157
Veronika Šalgová, Michal Kvet, Miroslav Potočár

7.1 Ako spravovať formuláre . . . 157
7.2 Editovateľná interaktívna mriežka . . . 157
7.3 Formulár založený na tabuľke . . . 160
7.4 Hlavný podrobný formulár - Master detail form . . . 160
7.5 Otázky . . . 163
7.6 Odpovede . . . 163

8 Ako previesť textové reporty na grafy? . . . 165
Ivan Pastierik, Michal Kvet, Miroslav Potočár

8.1 Graf . . . 165
8.2 Vytvorenie stĺpcového grafu . . . 166
8.3 Pridanie filtrovania pre stĺpcový graf . . . 169
8.4 Pridanie triedenia stĺpcového grafu . . . 175
8.5 Vytvorenie iných typov grafov . . . 178

8.6 Otázky . . . 180
8.7 Odpovede . . . 181

9 Ako spravovať menu? . . . 183
Veronika Šalgová, Michal Kvet, Miroslav Potočár

9.1 Ako spravovať menu . . . 183
9.2 Side Menu . . . 183
9.3 Top Menu . . . 184
9.4 Mega Menu . . . 184
9.5 Úprava menu zoznamov . . . 184
9.6 Otázky . . . 185
9.7 Odpovede . . . 185

10 Ako spolupracovať v tíme? . . . 187
P. Staniszewski, M. Sońta, A. Kierzkowski

10.1 Kolaboratívne vytváranie znalostí je podstatou vývoja v nízkom kóde 187
10.2 Byť spolu online . . . 187
10.3 Technická zdatnosť vďaka spolupráci . . . 188
10.4 Popis funkcií . . . 189
10.4.1 Uzamknutie stránky . . . 189
10.4.2 Komentáre . . . 190
10.4.3 Možnosti zostavovania . . . 190
10.4.4 Tímový vývoj . . . 191
10.4.5 Spätná väzba . . . 194
10.4.6 Štandardizácia . . . 195
10.5 Záver . . . 196
10.6 Otázky . . . 197
10.7 Odpovede . . . 197

11 Ako využiť galériu vzorových aplikácií a pluginov? . . . 199
Vjeran Strahonja, Dijana Oreški

11.1 Ako nainštalovať vzorové a štartovacie aplikácie? . . . 199
11.2 Štartovacie aplikácie . . . 200
11.3 Vzorové aplikácie (Sample Apps) . . . 201
11.4 Plug-iny . . . 203
11.5 Otázky . . . 206
11.6 Odpovede . . . 206

12 Ako spravovať zabalené a viacjazyčné aplikácie? . . . 207
R. Leskovar, U. Rajkovič, A.Baggia

12.1 Aplikácia a zabalená aplikácia . . . 207

12.2 Aplikácia . . . 209
12.2.1 Rozsah aplikácie . . . 209
12.2.2 Vytvorenie tabuliek . . . 209
12.2.3 Vloženie dát . . . 210
12.2.4 Generovanie aplikácie . . . 211
12.3 Zabalená aplikácia . . . 215
12.4 Viacjazyčná aplikácia . . . 219
12.5 Doplnkový učebný materiál . . . 223
12.5.1 Exportované aplikácie . . . 223
12.5.2 Video návody . . . 223
12.6 Otázky . . . 224
12.7 Odpovede . . . 224

II **Vytváranie aplikácie v APEX**

13 Intranetové novinky pre zamestnancov . . . 227
R. Leskovar, U. Rajkovič, A.Baggia . . .

13.1 Biznis pohľad prípadu . . . 227
13.2 Definícia problému . . . 227
13.3 Prípady použitia . . . 228
13.3.1 Naratívny popis . . . 228
13.3.2 Pološtruktúrovaný popis . . . 228
13.3.3 Diagram prípadov použitia . . . 228
13.4 Dátový model . . . 232
13.4.1 Naratívny popis dátového modelu . . . 232
13.4.2 Logický dátový model . . . 232
13.4.3 Relačný dátový model . . . 232
13.5 Rozhrania aplikácie . . . 233
13.6 Otázky . . . 236
13.7 Odpovede . . . 236

14 GreenDi - Katalóg rastlín . . . 237
Vjeran Strahonja, Dijana Oreski, Darko Androcec, Ana Kutnjak . . .

14.1 Biznis pohľad prípadu . . . 237
14.2 Definícia problému . . . 237
14.3 Prípady použitia . . . 238
14.3.1 Naratívny popis . . . 238
14.3.2 Pološtruktúrovaný popis . . . 238
14.3.3 Diagram prípadov použitia . . . 238
14.4 Dátový model . . . 240
14.4.1 Naratívny popis dátového modelu . . . 240
14.4.2 Logický dátový model . . . 240
14.4.3 Relačný dátový model . . . 241

14.5 Rozhrania aplikácie . . . 242
14.6 Doplnkový učebný materiál . . . 243
14.6.1 Exportovaná aplikácia . . . 243
14.6.2 Video sprievodca . . . 244
14.7 Otázky . . . 244
14.8 Odpovede . . . 244

15 GreenDi - Autorizácia a správa používateľov . . . 247
VJERAN STRAHONJA, DARKO ANDROCEC, ANA KUTNJAK, LARISA HRUSTEK . . .

15.1 Biznis pohľad prípadu . . . 247
15.2 Definícia problému . . . 247
15.3 Prípady použitia . . . 248
15.3.1 Príbehový popis prípadu použitia . . . 248
15.3.2 Pološtruktúrovaný popis . . . 248
15.3.3 Diagram prípadov použitia . . . 248
15.4 Dátový model . . . 248
15.4.1 Príbehový popis prípadu použitia . . . 248
15.4.2 Logický dátový model . . . 250
15.4.3 Relačný dátový model . . . 250
15.5 Rozhrania aplikácie . . . 251
15.6 Doplnkový učebný materiál . . . 251
15.6.1 Exportované aplikácie . . . 251
15.6.2 Video sprievodcovia . . . 252
15.7 Otázky . . . 252
15.8 Odpovede . . . 252

16 Malý inovačný systém . . . 253
R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA . . .

16.1 Biznis pohľad prípadu . . . 253
16.2 Definícia problému . . . 253
16.3 Prípady použitia . . . 253
16.3.1 Naratívny popis . . . 253
16.3.2 Pološtruktúrovaný popis . . . 254
16.3.3 Diagram prípadov použitia . . . 254
16.4 Dátový model . . . 254
16.4.1 Príbehový popis dátového modelu . . . 254
16.4.2 Logický dátový model . . . 254
16.4.3 Relačný dátový model . . . 258
16.5 Rozhrania aplikácie . . . 259
16.6 Doplnkový učebný materiál . . . 261
16.6.1 Exportovaná aplikácia . . . 262
16.6.2 Video sprievodca . . . 262
16.7 Otázky . . . 262
16.8 Odpovede . . . 262

17 Manažment podnikových procesov . . . 263
R. Leskovar, U. Rajkovič, A.Baggia

17.1 Biznis pohľad prípadu . . . 263
17.1.1 Inštalácia Flows for APEX . . . 266
17.1.2 Inštalovanie Sample Process Flow aplikácie . . . 266
17.1.3 Cvičenia zamerané na čítanie a prax . . . 266
17.2 Definovanie problému . . . 267
17.3 Prípady použitia . . . 267
17.3.1 Naratívny popis . . . 267
17.3.2 Pološtruktúrovaný popis . . . 267
17.3.3 Diagram prípadu použitia . . . 270
17.4 Model pracovného toku (Workflow model) . . . 270
17.5 Dátový model . . . 273
17.5.1 Naratívny popis dátového modelu . . . 273
17.5.2 Logický dátový model . . . 273
17.5.3 Relačný dátový model . . . 273
17.5.4 Quick SQL for generating SQL script . . . 274
17.5.5 Sekvencia a dve uložené funkcie . . . 275
17.6 Rozhrania aplikácie . . . 276
17.6.1 Zoznam hodnôt v Shared Components . . . 277
17.6.2 Plug-iny v Shared Components . . . 278
17.6.3 Manažér predaja (Sales manager) . . . 278
17.6.4 Manažér výroby . . . 282
17.6.5 Finančný manažér . . . 284
17.6.6 Generálny riaditeľ - manažér podniku . . . 285
17.7 Prepojenie aplikácie s Flows for APEX . . . 287
17.8 Definovanie užívateľských rolí . . . 289
17.9 Testovanie a oprava chýb . . . 289
17.10 Doplnkový učebný materiál . . . 290
17.10.1 Exportovaná aplikácia . . . 290
17.10.2 Video návody . . . 292
17.11 Otázky . . . 292
17.12 Odpovede . . . 292

18 GreenDi – Výmena rastlín a semien . . . 293
Vjeran Strahonja

18.1 Biznis pohľad prípadu . . . 293
18.2 Definícia problému . . . 293
18.3 Prípady použitia . . . 294
18.3.1 Naratívny popis . . . 294
18.3.2 Pološtruktúrovaný popis . . . 294
18.3.3 Diagram prípadov použitia . . . 294
18.4 Dátový model . . . 294
18.4.1 Naratívny popis dátového modelu . . . 294
18.4.2 Logický dátový model . . . 296
18.4.3 Relačný dátový model . . . 296

18.5 Rozhrania aplikácie . . . 297
18.6 Doplnkový učebný materiál . . . 298
18.6.1 Exportovaná aplikácia . . . 298
18.6.2 Video sprievodca . . . 298
18.7 Otázky . . . 298
18.8 Odpovede . . . 298

19 Systém správy knižných recenzií . . . 299
A. Kutnjak, L. Hrustek, A. Baggia and R.Leskovar

19.1 Biznis pohľad prípadu . . . 299
19.2 Definícia problému . . . 299
19.3 Prípady použitia . . . 300
19.3.1 Naratívny popis . . . 300
19.3.2 Pološtruktúrovaný popis . . . 300
19.4 Dátový model . . . 302
19.4.1 Naratívny popis dátového modelu . . . 302
19.4.2 Logický dátový model . . . 302
19.4.3 Relačný dátový model . . . 302
19.4.4 SQL skript . . . 303
19.4.5 Quick SQL . . . 304
19.5 Rozhrania aplikácie . . . 305
19.5.1 Administrátor . . . 305
19.5.2 Používateľ . . . 307
19.6 Definovanie užívateľských rolí . . . 309
19.7 Doplnkový učebný materiál . . . 309
19.7.1 Exportovaná aplikácia . . . 309
19.7.2 Video návody . . . 310
19.8 Otázky . . . 310
19.9 Odpovede . . . 310

20 Kusovník a kalkulácia nákladov . . . 311
R. Leskovar, U. Rajkovič, A.Baggia

20.1 Biznis pohľad prípadu . . . 311
20.2 Definícia problému . . . 311
20.3 Prípady použitia . . . 312
20.3.1 Naratívny popis . . . 312
20.3.2 Pološtruktúrovaný popis . . . 312
20.3.3 Diagram prípadov použitia . . . 312
20.4 Dátový model . . . 315
20.4.1 Naratívny popis . . . 315
20.4.2 Implementácia podnikových pravidiel v databáze . . . 315
20.4.3 Logický dátový model . . . 316
20.4.4 Relačný dátový model . . . 316
20.4.5 Objekty v APEX . . . 318

20.5 Rozhrania aplikácie . . . 322
20.6 Doplnkové účebné materiály . . . 324
20.6.1 Exportované aplikácie . . . 324
20.6.2 Video návody . . . 324
20.7 Otázky . . . 324
20.8 Odpovede . . . 324

21 Manažment výživy a stravovania . . . 327
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . . .

21.1 Biznis pohľad prípadu . . . 327
21.2 Definícia problému . . . 327
21.3 Prípady použitia . . . 327
21.3.1 Naratívny popis . . . 327
21.3.2 Pološtruktúrovaný popis . . . 328
21.3.3 Diagram prípadov použitia . . . 328
21.4 Dátový model . . . 328
21.4.1 Logický dátový model . . . 328
21.4.2 Relačný dátový model . . . 330
21.4.3 QuickSQL a Script SQL Script . . . 331
21.5 Rozhrania aplikácie . . . 334
21.5.1 Ako vytvoriť report . . . 338
21.5.2 Vytvorenie prístupu pre používateľa . . . 342
21.5.3 Exportovaná aplikácia . . . 345
21.5.4 Video návody . . . 345
21.6 Otázky . . . 345
21.7 Odpovede . . . 345

22 Plánovanie konzultačných hodín . . . 347
J. Mańko, M. Sońta, R. Leskovar . . .

22.1 Biznis pohľad prípadu . . . 347
22.2 Definícia problému . . . 348
22.3 Prípady použitia . . . 348
22.3.1 Naratívny popis . . . 349
22.3.2 Pološtruktúrovaný popis . . . 349
22.3.3 Diagram prípadu použitia . . . 349
22.4 Dátový model . . . 351
22.4.1 Naratívny popis dátového modelu . . . 351
22.4.2 Logický dátový model . . . 352
22.4.3 Relačný dátový model . . . 352
22.4.4 Quick SQL . . . 353
22.4.5 SQL Skript na vytváranie tabuliek . . . 354
22.4.6 Query builder v APEX . . . 354
22.5 Rozhrania aplikácie . . . 355
22.5.1 Rozhrania aplikácie pre manažment . . . 355
22.5.2 Rozhrania aplikácie pre študijné oddelenie . . . 356
22.5.3 Rozhrania študentskej aplikácie . . . 357

22.5.4 Rozhrania aplikácie pre učiteľov . . . 360
22.6 Doplnkový učebný materiál . . . 362
22.6.1 Exportovaná aplikácia . . . 363
22.6.2 Videonávody . . . 365
22.7 Otázky . . . 365
22.8 Odpovede . . . 365

23 Prípad Telco . . . 367
Veronika Šalgová, Jozef Kostolný, Michal Mrena, Michal Kvet, Miroslav Potočár . . . .
23.1 VEYOMI aplikácia . . . 367
23.2 Diagram prípadov použitia . . . 367
23.3 Dátový model . . . 368
23.3.1 Naratívny popis . . . 368
23.3.2 Logický dátový model . . . 369
23.3.3 Relačný dátový model . . . 369
23.4 Autentifikácia používateľa a používateľské roly . . . 369
23.5 Rozhrania aplikácie . . . 372
23.6 Dizajn aplikácie . . . 372
23.7 Skripty . . . 380
23.8 Vytvorenie domovskej stránky . . . 382
23.9 Vytvorenie stránky zákazníka . . . 383
23.10 Vytvorenie stránky manažéra . . . 386
23.11 Doplnkové výučbové materiály . . . 386
23.11.1 Exportovaná aplikácia . . . 386
23.11.2 Video návody . . . 386
23.12 Otázky . . . 386
23.13 Odpovede . . . 387

24 Prípad prenájmu automobilov . . . 389
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . . . .
24.1 Biznis pohľad prípadu . . . 389
24.2 Definícia problému . . . 389
24.3 Prípady použitia . . . 390
24.3.1 Naratívny popis . . . 390
24.3.2 Pološtruktúrovaný popis . . . 390
24.3.3 Diagram prípadov použitia . . . 392
24.4 Dátový model . . . 392
24.4.1 Textový popis dátového modelu . . . 392
24.4.2 Logický dátový model . . . 393
24.4.3 Relačný dátový model . . . 393
24.4.4 SQL skript . . . 394

24.5 Rozhrania aplikácie **395**
24.6 Doplnkové výučbové materiály **400**
24.6.1 Exportovaná aplikácia 400
24.6.2 Video návody 400
24.7 Otázky **400**
24.8 Odpovede **400**

Literatúra **401**
Články **401**
Knihy **401**

Register **403**

Zoznam obrázkov

1.1 Cyklus vývoja aplikácie. 36
1.2 Business Model Canvas (6). 38
1.3 Základné koncepty Use Case diagramu. 39
1.4 Zovšeobecnenie/špecializácia hercov. 40
1.5 Hranice systému a asociácie "include" a "extend". 40
1.6 Use Cases diagram hotelového rezervačného systému. 41

2.1 DBS ako chrbtica Webových aplikácií. 49
2.2 Logický Model nášho príkladu o Ľudských Zdrojoch reprezentovaný ER-diagramom. 51
2.3 Správa logického modelu pomocou ODM. 52
2.4 Správa atribútov pomocou ODM. 53
2.5 Správa vzťahov pomocou ODM. 54
2.6 RDB-schémy a tabuľkové schémy. 55
2.7 Proces generovania RDB-schémy z logického Modelu. 56
2.8 Automaticky vygenerovaná RDB-schéma. 56
2.9 GUI-elementy pre správu stĺpcov RM. 58
2.10 GUI-elementy pre správu RM. 58
2.11 Kroky normalizácie – Prehľad. 59
2.12 Kategórie SQL príkazov. 61
2.13 Export RDB-schémy do DDL-skriptu. 62
2.14 Nahratie a Spustenie SQL-Skriptu. 63
2.15 Prístup ku Quick SQL cez SQL Workshop 64
2.16 Vygenerovaný SQL-Script pomocou Quick SQL. 65
2.17 SQL Command Editor. 66
2.18 SQL príkaz pre vytvorenie tabuľky `Departments`. 66
2.19 SQL príkaz na vymazanie tabuľky `Departments`. 67
2.20 SQL príkaz na zmenu tabuľky `Departments` 67
2.21 Správa tabuliek pomocou ORACLE Object Browser. 67
2.22 SQL príkaz na vkladanie nových údajov do tabuľky `Departments`. 68
2.23 SQL príkaz na aktualizáciu platu všetkých zamestnancov o 10% v tabuľke `Employees`. 68
2.24 SQL príkaz na aktualizáciu platu zamestnanca s priezviskom "Miller". 68
2.25 SQL príkaz na aktualizáciu viac ako jedného atribútu v tabuľke `Employees`. . . . 69

2.26 SQL príkaz na vymazanie všetkých údajov z tabuľky Employees. 69
2.27 SQL príkaz na vymazanie niektorých údajov z tabuľky Employees. 69
2.28 Náhodné vkladanie údajov pomocou Quick SQL. 70
2.29 SQL príkaz na získanie všetkých údajov z tabuľky Employees. 70
2.30 Časť z výsledku všetkých údajov z tabuľky EMPLOYEES. 71
2.31 SQL príkaz na získanie všetkých údajov z tabuľky Employees zotriedené podľa atribútu Salary. 71
2.32 Časť z výsledku všetkých údajov z tabuľky Employees zotriedené podľa atribútu Salary. 71
2.33 SQL Príkaz na získanie len určitých údajov z tabuľky Employees. 72
2.34 Časť výsledku len určitých údajov z tabuľky Employees. 72
2.35 SQL príkaz na získanie údajov z tabuľky Employees s použitím funkcie TO_CHAR(). 72
2.36 Časť výsledku získaných údajov z tabuľky Employees s použitím Funkcie TO_CHAR(). 73
2.37 SQL príkaz na spojenie tabuľky Employees a tabuľky Departments. 73
2.38 Časť výsledku určitých údajov spojením tabuľky Employees a tabuľky Departments. 74
2.39 SQL príkaz na výpočet počtu zamestnancov a zostavenie súčtu ich platov z tabuľky Employees. 74
2.40 Výsledok získania počtu zamestnancov a zostavenia súčtu ich úlatov z tabuľky Employees. 74
2.41 SQL príkaz pre zoskupenie údajov podľa platu zamestnancov a zoradenie výsledku podľa počtu zamestnancov s týmto platom z tabuľky Employees. 75
2.42 Časť z výsledku zoskupenia údajov podľa platu zamestnancov a zoradenie výsledku podľa počtu zamestnancov s týmto platom z tabuľky Employees 75
2.43 Prehľad nástroja ORACLE Query Builder ("QB"). 76
2.44 Použitie nástroja Query Builder na vytvorenie spojenia. 77
2.45 Prehľad možností správy DB-Layer. 78

3.1 Vývojový proces v APEX-e. 82
3.2 Oracle APEX Workspace Homepage. 83
3.3 Prehľad nástroja App Builder. 84
3.4 Create Application Wizard: Tri prípady použitia pri vytváraní Aplikácie. 85
3.5 Vlastnosti a stránky. 86
3.6 Funkcie a Nastavenia. 86
3.7 Application Homepage - pohľad vývojára. 87
3.8 Vytvorenie stránky. 88
3.9 Page Designer. 89
3.10 Spustenie Celej Aplikácie. 90
3.11 Spustenie jednotlivých stránok. 91

4.1 Možnosti výmeny dát v APEX-e. 94
4.2 Použitie Data Workshop na načítanie údajov - prístup cez SQL Workshop. 95
4.3 Použitie Data Workshop na načítanie údajov – prístup cez App Builder. 96
4.4 Poskytovanie zdroja údajov. 97
4.5 Načítavanie údajov do novej tabuľky. 97
4.6 Načítanie údajov do existujúcej tabuľky. 99
4.7 Export ("Unloading") údajov. 100
4.8 Unload Data Wizard. 101
4.9 Import/Export údajov tabuľky pomocou Object Browser. 102
4.10 Export údajov výsledku SQL-Príkazu. 103

4.11 Export údajov z reportu aplikácie 104
4.12 RESTful Servis pre výmenu údajov – základná architektúra. 105
4.13 Povolenie DB-schémy pre RESTful prístup. 107
4.14 Vzťahy medzi rôznymi komponentmi ORDS RESTful Services. 108
4.15 Príklad RESTful služieb pre tabuľku Employees. 109
4.16 Povolenie služby REST v prehliadači objektov. 110

5.1 Zadajte APEX odkaz a prihláste sa. 114
5.2 Úvodné kroky žiadosti o Oracle APEX Workspace. 114
5.3 Dokončenie žiadosti o Oracle APEX Workspace. 115
5.4 Schválený Oracle APEX Workspace. 115
5.5 Aktivačný e-mail od APEX. 116
5.6 Nastavenie nového hesla pre Workspace APEX. 116
5.7 Prostredie Workspace APEX. 117
5.8 Logický model nášho HR príkladu. 118
5.9 Relačný model nášho HR príkladu. 119
5.10 Vloženie kódu Quick SQL do pracovného priestoru APEX. 121
5.11 Spustiť SQL skript. 123
5.12 Proces spustenia aplikácie. 123
5.13 Vytvorenie aplikácie zo skriptu. 124
5.14 Vytvorená webová aplikácia. 125
5.15 App builder prostredie. 125
5.16 Prihlasovacia stránka aplikácie. 126
5.17 Prostredie vašej novej aplikácie. 126
5.18 Prostredie pre správu používateľov a skupín. 127
5.19 Príklad riadenia prístupu. 128
5.20 Pridanie nových používateľov. 129
5.21 Pridanie rolí používateľom. 130
5.22 Vytvorenie konfigurácie tlačidla. 131
5.23 Prístup a práva pre rolu Contributor. 132
5.24 Prístup a práva pre rolu Reader. 133

6.1 Výber typu stránky. 136
6.2 Tvorba stránky s reportom. 137
6.3 Vykresľovanie. 137
6.4 Vlastnosti oblasti. 138
6.5 Rôzne zdroje údajov. 138
6.6 Rozloženie. 139
6.7 Vzhľad. 139
6.8 Šablóny. 140
6.9 Pokročilé. 140
6.10 Hlavička a päta. 140
6.11 Podmienky na strane servera. 141
6.12 Len na čítanie. 141
6.13 Zabezpečenie, Vyrovnávacia pamäť servera a Prispôsobenie. 142
6.14 Stránkovanie. 142
6.15 Počet riadkov, ktoré sa majú načítať. 142
6.16 Stiahnutie a tlač. 143
6.17 Stred dizajnéra stránky. 143

6.18 Typ stĺpca. 144
6.19 Nastavenia stĺpcov. 144
6.20 Formát masky. 144
6.21 Prispôsobenie. 145
6.22 Možnosť triedenia podľa stĺpcov. 145
6.23 Umiestnenie zelenej šípky pre prezentačnú stránku. 145
6.24 Príklad klasickej zostavy. 146
6.25 Možnosť zmeny viditeľnosti stĺpca v Zostava s možnosťou prepínania stĺpcov. 146
6.26 Interaktívna správa. 147
6.27 Vyhľadávací panel. 147
6.28 Vyhľadávanie. 148
6.29 Tlačidlo Akcie. 148
6.30 Výber stĺpcov na zobrazenie. 149
6.31 Stĺpcový filter. 149
6.32 Riadkový filter. 150
6.33 Po kliknutí na tlačidlo Údaje. 150
6.34 Triediť. 151
6.35 Agregácia. 151
6.36 Retrospektíva. 151
6.37 Kontrolná prestávka. 152
6.38 Výsledok kontrolnej prestávky. 152
6.39 Zvýraznenie. 152
6.40 Zvýraznenie výsledku. 153
6.41 Riadky na stránke. 153
6.42 Graf v interaktívnej správe. 153
6.43 Zgrupovať podľa. 154
6.44 Uložiť zostavu. 154
6.45 Stiahnuť. 155

7.1 Interaktívna mriežka s údajmi o zákazníkoch. 157
7.2 Možnosti interaktívnej mriežky. 158
7.3 Ikona skrytia v stĺpci Phone column. 158
7.4 Zobrazené stĺpce. 158
7.5 Ikona zmrazenia v stĺpci adresa. 159
7.6 Zoradenie stĺpca s adresami. 159
7.7 Ikona súčtu v stĺpci adresy. 159
7.8 Typy grafov. 160
7.9 Formulár pre vkladanie zákazníkov. 160
7.10 Stacked master detail form zákazníckych paušálnych sadzieb. 161
7.11 Side by Side master detail form zákazníckych paušálnych sadzieb. 161
7.12 Drill Down master detail form – prvá strana. 162
7.13 Drill Down master detail form – druhá strana. 162

8.1 Dátový model používaný vo vzorovej aplikácii. 165
8.2 Výber a pokladanie regiónu "Chart". 166
8.3 Sekcia telo v Page Designeri po položení a premenovaní regiónu "Chart". . . . 166
8.4 Stromový prehľad komponentov stránky s chybou v časti "Series" pod regiónom "Sales Chart". 167

8.5 Konečná konfigurácia série "sales". 167
8.6 Dáta vrátené vykonaním SQL dotazu. 168
8.7 Graf zobrazený po spustení aplikácie. 168
8.8 Zmena názvu osi x. 169
8.9 Graf po pomenovaní osí. 169
8.10 Výber a pokladanie predmetu "Checkbox Group". 170
8.11 Nastavenie základných atribútov predmetu "Checkbox Group". 170
8.12 Nastavenie zoznamu hodnôt pre predmet "Checkbox Group". 171
8.13 Nastavenie predvolených hodnôt pre predmet "Checkbox Group". 171
8.14 Prepojenie regiónu "Sales Chart" s predmetom "Checkbox Group". 172
8.15 Vytvorenie dynamickej akcie pre predmet s názvom "P1_PRODUCTS_CHOICE". 173
8.16 Nastavenie základných atribútov dynamickej akcie vykonanej pri zmene. . . . 173
8.17 Nastavenie atribútov akcie obnovenia. 174
8.18 Dynamická akcia pre obnovenie regiónu "Sales Chart" v stromovom prehľade komponentov stránky. 174
8.19 Aplikácia po implementovaní filtrovania produktov. 175
8.20 Výber a pokladanie predmetu "Select List". 175
8.21 Región "Sales Chart" po položení a premenovaní predmetu "Select List". . . . 176
8.22 Nastavenie zoznamu hodnôt pre predmet "Select List". 176
8.23 Nastavenie statických hodnôt pre predmet "Select List". 177
8.24 Pridanie dynamickej akcie obnovenia pre predmet "Select List". 177
8.25 Nastavenie atribútu "Order By Item" pre región "Sales Chart". 177
8.26 Nastavenie "Order By" klauzúl používaných pre triedenie na základe predmetu "P1_CHART_SORTING". 178
8.27 Aplikácia po implementovaní triedenia grafu. 178
8.28 Zmenenie typu grafu na "Line with Area". 179
8.29 Zoznam hodnôt pre predmet "Select List" použitý pre filtrovanie predajov produktov podľa roku. 179
8.30 Nastavenie zdrojového SQL dotazu a mapovania stĺpcov pre región "Monthly Sales Chart". 180
8.31 Graf mesačných predajov. 180

9.1 Rozšírené bočné navigačné menu 183
9.2 Top Navigation Tabs 184
9.3 Top Navigation Menu 184
9.4 Top Navigation Mega Menu 184
9.5 Atribúty navigačného menu 185

10.1 Funkcia blokovania stránok v APEX. 189
10.2 Uzamknutie konkrétnej stránky. 189
10.3 Pridávanie komentárov. 190
10.4 Možnosti zostavovania. 191
10.5 Funkcia tímového vývoja. 192
10.6 Aktivácia funkcie tímového vývoja. 192
10.7 Prispôsobenie funkcie Team Development. 193
10.8 Komunikácia súvisiaca s problémami v rámci funkcie Team Development. . . 193
10.9 Povolenie spätnej väzby v definícii aplikácie. 194
10.10 Odoslanie spätnej väzby v aplikácii. 195
10.11 Ponuka pomocných programov. 196

11.1 Pracovné priestory Oracle APEX vzorové a štartovacie aplikácie. 199
11.2 Štartovacie aplikácie v galérii. 200
11.3 Inštalácia štartovacej aplikácie Team Calendar. 201
11.4 Vzorové aplikácie. 202

12.1 Sprievodca importom a exportom aplikácie. 208
12.2 Transformácia Quick SQL na SQL príkazy. 210
12.3 Skript CH12INSERT vloží údaje do troch tabuliek a potvrdí transakcie. 210
12.4 Vytvorenie aplikácie zo skriptu. 211
12.5 Výber názvu aplikácie a všetkých funkcií. 211
12.6 Výber Zdieľaných komponentov. 212
12.7 Vytvorenie zoznamu hodnôt od začiatku. 212
12.8 Definovanie názvu a typu: CH12_LOV_COMPETENCE_DESCRIPTION je dynamický. 213
12.9 Zadanie príkazu SQL SELECT. 213
12.10 Definovanie názvu a typu: CH12_LOV_COMPETENCE_LEVEL je statický. 214
12.11 Definovanie zobrazovaných a návratových hodnôt: CH12_LOV_COMPETENCE_LEVEL je statický. 214
12.12 Výber strany 7 (Ch12 Job Competences). 215
12.13 Výber strány 7 (Ch12 Job Competences). 215
12.14 Výber podporných objektov. 216
12.15 Nastavenie prerekvizít. 216
12.16 Nastavenie kontroly na existenciu troch tabuliek. 216
12.17 Nastavenie výzvy na premenovanie aplikácie. 217
12.18 Nastavenie inštalačných skriptov. 217
12.19 Nastavenie deinštalačných skriptov. 217
12.20 Import aplikácie do iného pracovného priestoru. 218
12.21 Licenčná zmluva. 218
12.22 Premenovanie importovanej aplikácie. 219
12.23 Pridanie roly používateľovi. 219
12.24 Definované jazyky na preklad. 220
12.25 Seed preložiteľného textu. 220
12.26 Export reťazcov pre konkrétny jazyk a stránku. 221
12.27 Preklad reťazcov označených ako "target" v riadkoch 48, 52, 56, 60, 72 a 76. 221
12.28 Nahrávanie prekladových súborov XLDIFF. 221
12.29 Aplikovanie zmien a publikovanie. 222
12.30 Finálne zverejnenie prekladu aplikácie. 222
12.31 Nastavenie hlavného jazyka aplikácie. 222
12.32 Preložená stránka. 222

13.1 Diagram prípadov použitia. 228
13.2 Logický dátový model. 232
13.3 Logický dátový model. 233
13.4 Hlavný panel prispievateľa/vydavateľa. 233
13.5 Publikovanie noviniek s prílohami. 234
13.6 Hlavný panel čitateľa. 234
13.7 Prístup čitateľa k novinkám. 235
13.8 Panel správcu. 235
13.9 Úpravy roly zamestnanca administrátorom. 236

14.1 Diagram prípadov použitia - katalóg rastlín. 240

14.2 Logický dátový model. 241
14.3 Relačný dátový model. 242
14.4 Interaktívna mriežka pre formulár Plant. 243
14.5 Verejná časť – otvorené prehliadanie údajov o rastlinách. 243

15.1 Diagram prípadov použitia - Autorizácia a správa používateľov GreenDi 248
15.2 Logický dátový model. 250
15.3 Relačný dátový model. 250
15.4 Formulár používateľa. 251
15.5 Formulár histórie používateľov. 251

16.1 Diagram prípadov použitia. 254
16.2 Logický dátový model. 258
16.3 Relačný dátový model. 258
16.4 Ovládací panel pre zamestnancov. 259
16.5 Odoslanie nápadu s prílohami. 259
16.6 Organizačná štruktúra spoločnosti. 260
16.7 Prehľad Malého Inovačného Systému. 260
16.8 Ovládací panel recenzenta. 261
16.9 Stránka na hodnotenie nápadov. 261

17.1 Integrácia Flows for APEX s aplikáciou APEX. 265
17.2 Use case diagram. 270
17.3 Workflow - processing inquiry - BPMN diagram. 270
17.4 Vytváranie inštancií. 271
17.5 Vytvorenie inštancie toku CH17. 271
17.6 Spustenie demo inštancie. 271
17.7 Dokončenie prvej úlohy v demo inštancii. 272
17.8 Vykonávanie úloh v paralelnej bráne. 272
17.9 Dokončená demo inštancia. 272
17.10 Logický dátový model. 273
17.11 Relačný dátový model. 274
17.12 Domovská stránka "CH17 Business Process Management" aplikácie. 277
17.13 Predaj - iniciovaný proces. 278
17.14 Sales - flow report. 279
17.15 Predaj - prehľad toku pre vybranú inštanciu. 279
17.16 Predaj - prehľad toku pre vybranú inštanciu. 279
17.17 Predaj -prehľad dopytov. 280
17.18 Predaj -zoznam dokumentov. 280
17.19 Predaj - nahratie dokumenty pre dopyt 17. 281
17.20 Predaj - zoznam dokumentov po nahraní a obnovení. 281
17.21 Predaj - prehľad toku pre vybratú inštanciu po nahratí troch dokumentov. . 282
17.22 Predaj - prehľad toku pre vybratú inštanciu po nahratí troch dokumentov. . 282
17.23 Výroba - prehľad toku. 283
17.24 Výroba - BPMN diagram - stav inštancie. 283
17.25 Hodnotenie z výrobného hľadiska. 283
17.26 Výrobné hodnotenie ukončené, inštancia čaká na finančné hodnotenie. . . 284
17.27 Financie - prehľad toku. 284
17.28 Financie - BPMN diagram - stav inštancie. 284
17.29 Finančné hodnotenie. 285

17.30 Finančné hodnotenie ukončené, inštancia čaká na hodnotenie z podnikateľského hľadiska. 285
17.31 Podnikové hodnotenie - BPMN diagram - stav inštancie. 286
17.32 Podnikové hodnotenie - prehľad toku. 286
17.33 Hodnotenie na najvyššej úrovni. 286
17.34 Hodnotenie na najvyššej úrovni skončilo, inštancia ukončená. 287
17.35 Zobrazenie stavu inštancie na diagrame BPMN. 288
17.36 Nastavenie položiek stránky. 288
17.37 Dokončenie kroku vo Flows for Apex. 289
17.38 Definujte roly a užívateľské roly v Application Access Control menu. 289
17.39 Udelenie "create job privilege" workspace-u. 290
17.40 Import pracovného toku s názvom CH17 do Flows for APEX - krok 1. 291
17.41 Import pracovného toku s názvom CH17 do Flows for APEX - krok 2. 291

18.1 Diagram prípadov použitia - Výmena rastlín a semien 294
18.2 Logický dátový model. 296
18.3 Relačný dátový model. 297
18.4 Ponuka (Offer). 297
18.5 Správa (Message). 297

19.1 Use case diagram: administrátor pridáva nový komentár. 300
19.2 Logický dátový model. 302
19.3 Relačný dátový model. 303
19.4 Generovanie aplikácie zo skriptu CH19CREATEINSERT - časť 1. 305
19.5 Generovanie aplikácie zo skriptu CH19CREATEINSERT - časť 2. 306
19.6 Pridanie knihy správcom. 307
19.7 Pridanie kategórie správcom. 307
19.8 Registrácia používateľa 308
19.9 Prehliadanie a pridávateni recenzií. 308
19.10 Formulár na komentovanie recenzie 309

20.1 Diagram prípadu použitia. 315
20.2 Logický dátový model. 316
20.3 Definícia jedinečného indexu v programe Oracle SQL Data Modeler. 317
20.4 Relačný dátový model. 317
20.5 Generovanie skriptu SQL pomocou nástroja Quick SQL. 319
20.6 Domovská stránka aplikácie. 322
20.7 Správa základných údajov - report. 322
20.8 Správa základných údajov - formulár. 323
20.9 Správa štruktúry - report. 323
20.10 Správa štruktúry dát - form. 323
20.11 Strana na výpočet kusovníka. 324

21.1 Diagram prípadov použitia. 328
21.2 Logický diagram. 330
21.3 Relačný diagram. 331
21.4 Quick SQL procedure to create tables. 332
21.5 Procedúra s SQL skriptom na vytvorenie SEQUENCE. 334
21.6 Vytvorenie novej aplikácie. 335
21.7 Nutričná aplikácia. 336

21.8 Nutričná aplikácia - stránka receptu. 337
21.9 Nutričná aplikácia - stránka ingrediencií. 337
21.10 Nutričná aplikácia - stránka s ingredienciami receptov. 338
21.11 Nutričná aplikácia - stránka s informáciami o výživovej hodnote. 338
21.12 Vytvorenie dashboardu pre aplikáciu. 339
21.13 Konfigurácia názvu a typu grafu v dashboard reportoch. 340
21.14 Konfigurácia SOURCE s SQL QUERY. 342
21.15 Vytvorenie užívateľského účtu. 343
21.16 Vytvorenie užívateľského účtu. 344
21.17 Konfigurácia stránok s užívateľskými oprávneniami. 345

22.1 Diagram prípadu použitia. 351
22.2 Logický dátový model. 352
22.3 Relačný dátový model. 353
22.4 Dátový model vytvorený pomocou Quick SQL. 353
22.5 Príklad použitia Query Builder. 354
22.6 Domovská stránka aplikácie. 355
22.7 Manažment dashboard. 356
22.8 Interaktívny prehľad o konzultačných hodinách pre študijné oddelenie. 356
22.9 Formulár na zmenu termínu pre študijné oddelenie. 357
22.10 Interaktívny prehľad o konzultačných hodinách po zmene termínu. 357
22.11 Kalendár konzultácií študentov. 358
22.12 Kalendár všetkých konzultačných hodín učiteľov zapojených do študijného programu študenta. 358
22.13 Zápis na konzultačné hodiny prostredníctvom kalendára - výber okienka v kalendári. 358
22.14 Zápis na konzultačné hodiny prostredníctvom kalendára - výber účelu. 359
22.15 Zápis do konzultačných hodín prostredníctvom kalendára - úspešný zápis. . 359
22.16 Kontrola zápisu prostredníctvom položky "View my meetings". 359
22.17 Podrobný interaktívny prehľad o všetkých konzultáciách študentov. 360
22.18 Detailný prehľad o konzultáciách učiteľov 360
22.19 Kalendár učiteľa s viditeľnými menami žiakov, ich jazykmi a cieľmi. 361
22.20 Zmena konzultačných hodín učiteľa, na ktoré nie je niekto zapísaný - zobrazenie kalendára. 361
22.21 Zmena termínu konzultačných hodín učiteľa, na ktorý nie je niekto zapísaný - zadaný nový termín. 362
22.22 Výsledky preloženia konzultačných hodín učiteľa, na ktoré nebol nik zapísaný. 362

23.1 Diagram prípadov použitia. 368
23.2 Logický dátový model. 369
23.3 Dátový model. 369
23.4 Zákaznícky dashboard. 373
23.5 Zákaznícky dashboard – Modálne okno faktúry. 374
23.6 Zákaznícky dashboard – Faktúra v PDF. 374
23.7 Zákaznícky dashboard – Štatistika minút. 374
23.8 Zákaznícky dashboard – Štatistika SMS. 375
23.9 Zákaznícky dashboard – Štatistika dát. 375
23.10 Manažérsky dashboard – Pridanie zákazníka. 376
23.11 Manažérsky dashboard – Pridanie zákazníka. 377
23.12 Manažérsky dashboard – Zobrazenie zákazníkov. 377

23.13 Manažérsky dashboard – Export zákazníkov. 378
23.14 Manažérsky dashboard – Export zákazníkov do XLS súboru. 378
23.15 Dashboard administrátora. 379
23.16 Úvodná stránka s prihlásením. 379
23.17 Prihlasovacia stránka. 379
23.18 Regióny časti body. 382
23.19 Zoznam obsahu. 382
23.20 HTML kód statického regiónu. 383

24.1 Diagram prípadu použitia. 392
24.2 Logický model projektu autopožičovne. 393
24.3 Relačný model projektu autopožičovne. 394
24.4 Prihlásenie do aplikácie. 396
24.5 Domovská stránka aplikácie. 396
24.6 Šablóna Cars (Autá). 397
24.7 Ako vytvoriť stĺpec s fotografiami. 398
24.8 Šablóna Customers (Zákazníci). 399
24.9 Šablóna Rentcar (Prenájom áut). 399



Zoznam tabuliek

13.1 Popis prípadu použitia: publikovanie internetových správ. 229
13.2 Popis prípadu použitia: čítanie intranetových správ 230
13.3 Popis prípadu použitia: Spravovanie intranetového portálu. 231

14.1 Popis prípadu použitia: prehliadanie katalógu rastlín 239

15.1 Popis prípadu použitia: Autorizácia a správa používateľov. 249

16.1 Popis prípadu použitia: Predloženie nápadu. 255
16.2 Popis prípadu použitia: Prehľad nápadov. 256
16.3 Popis prípadu použitia: Hodnotenie nápadu. 257

17.1 Popis prípadu použitia: príprava dokumentácie k dopytu. 268
17.2 Popis prípadu použitia: zhodnotenie výrobného aspektu dopytu. 268
17.3 Popis prípadu použitia: zhodnotenie finančného aspektu dopytu. 269
17.4 Popis prípadu použitia: zhodnotenie dopytu z hľadiska celého podniku. 269

18.1 Popis prípadu použitia: Výmena rastlín a semien. 295

19.1 Popis prípadu použitia: analýza recenzií kníh 301

20.1 Popis prípadu použitia: vykazovanie a údržba základných údajov. 312
20.2 Popis prípadu použitia: vykazovanie a údržba dát o štruktúre. 313
20.3 Popis prípadu použitia: vytvorenie kusovníka. 314

21.1 Popis prípadu použitia: Prístup k receptom, ingredienciám a k výpočtu nutričných hodnôt 329

22.1 Popis prípadu použitia: preloženie konzultačných hodín učiteľom. 349
22.2 Popis prípadu použitia: zápis študenta na konzultačné hodiny. 350

23.1 Vzorové údaje uložené v tabuľke CH23_Person. 371

24.1 Popis prípadu použitia: Sprístupňovanie vozidiel, zákazníkov a rezervácií prenájmov áut 391

Poďakovanie

Partneri projektu a BEE APEX tím:

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

chceli by sme vyjadriť vďaku za finančnú podporu, ktorú nám poskytla European Commission prostredníctvom projektu Erasmus + Better Employability for Everyone with APEX (ID projektu 2021-1-SI01-KA220-HED-000032218), spolufinancovaného z programu Erasmus+ Európskej únie.

Podpora Európskej komisie na vydanie tejto publikácie nepredstavuje schválenie jej obsahu, ktorý vyjadruje len názory autorov, a Komisia nenesie zodpovednosť za akékoľvek použitie informácií v nej obsiahnutých.

Veľká vďaka patrí:

- Oracle Academy *Programovému manažérovi pánovi Darkovi Jurekovićovi* za neustálu podporu pri propagácii výsledkov projektu a
- THE RIGHT THING SOLUTIONS *CEO pánovi Aljažovi Malimu* za jeho cenné rady týkajúce sa systému APEX pred začiatkom a počas realizácie projektu.

Tím projektu BeeAPEX



Predslov

Vitajte pri objavovaní Oracle Application Express (APEX) - intuitívnej a výkonnej platformy na vývoj s malým množstvom kódu na vytváranie webových aplikácií na základe údajov. Táto učebnica je navrhnutá tak, aby vás vybavila schopnosťami potrebnými na plné využitie potenciálu Oracle APEX a vytváranie špičkových aplikácií na riešenie reálnych obchodných výziev.

Prvá časť tejto učebnice "Ako na to", je venovaná základným aspektom Oracle APEX. V týchto dvanástich kapitolách sa vydáte na cestu k pochopeniu základných pojmov, získate prístup k systému APEX a preskúmate rôzne funkcie na vytváranie robustných aplikácií. Každá kapitola sa venuje konkrétnej téme, poskytuje jasné pokyny a uľahčuje praktické učenie.

Kapitola 1: "Ako spustiť Oracle APEX?" vysvetľuje, čo to je, na čo slúži, a opisuje rôzne spôsoby prípravy prostredia APEX na praktické učenie a vývoj aplikácií.

Kapitola 2: "Ako pripraviť databázu?" poskytuje úvod do modelovania údajov, správy databázy, manipulácie s údajmi a dopytovania sa na údaje. Pre začínajúcich používateľov je nevyhnutný široký prehľad o koncepciách bázy dát.

Kapitola 3: "Ako sa pohybovať v nástroji APEXe?" poskytuje prehľad funkcií APEXu, ktoré umožňujú vývoj aplikácií, generovanie a prispôsobovanie rôznych webových stránok a ich komponentov. Spustenie a testovanie aplikácie APEX je vzdialené len jednu kartu od vývojového prostredia.

Kapitola 4: "Ako si vymieňať údaje v APEX-e?" poskytuje prehľad o importe a exporte údajov v rámci APEXu. Kapitola sa zaoberá výmenou údajov pomocou súborov, ako sú tabuľky, a tiež prostredníctvom RESTful služieb.

Kapitola 5: "Ako vygenerovať prvý návrh aplikácie?" vás pozýva vyskúšať vývojovú silu APEXu. Zistíte, že keď sa raz rozhodnete, aké sú vaše údaje, môžete okamžite vygenerovať atraktívnu a funkčnú aplikáciu bez programovania. Okamžite sa dá vygenerovať aj základné riadenie prístupu pre koncových používateľov s rôznymi rolami.

Kapitola 6: "Ako spravovať reporty?" vás prevedie poskytovaním pohľadu na dáta prostredníctvom klasických a interaktívnych sprievodcov reportovaním. Reporty APEX už obsahujú funkcie na prispôsobenie reportov koncovým používateľom bez akéhokoľvek programovania.

Kapitola 7: "Ako spravovať formuláre?" vám predstaví tri bežné typy webových formulárov vrátane hlavného a detailného. Budete prispôsobovať a vytvárať stránky formulárov bez programátorských zručností.

Kapitola 8: "Ako previesť textové reporty na grafy?" pripravuje cestu k využitiu možností

APEX na prezentáciu údajov vo forme grafov priamo popri textových reportoch.

Kapitola 9: "Ako spravovať menu?" predstavuje typy rôznych navigačných prvkov pre vašu APEX aplikáciu.

Kapitola 10: "Ako spolupracovať v tíme?" ponúka pohľad na funkcie APEX, ktoré slúžia tímom, keďže vývoj aplikácií len zriedkavo zahŕňa len jedného vývojára.

Kapitola 11: "Ako využiť galériu vzorových aplikácií a pluginov?" vás vyzýva na použitie rozsiahlych možností APEXu prostredníctvom opätovného použitia osvedčených vzorov.

Kapitola 12: "Ako spravovať zabalené a viacjazyčné aplikácie?" určuje cestu distribúcie vašej aplikácie do iných prostredí APEX pre používateľov, ktorí hovoria rôznymi jazykmi.

Časť I sa zaoberá aj témami, ktoré sú dôležité pre bezpečnosť aplikácií, stratégie nasadenia a pripravenosť na použitie v reálnom svete.

Časť II tejto učebnice vás prevedie nad rámec základov a predstaví dvanásť pútavých biznis prípadov od slovného opisu, cez každý technický detail až po riešenie problému. Každý prípad je starostlivo zdokumentovaný tak, aby poskytol ucelené pochopenie vývoja aplikácií z pohľadu podnikania, údajov a používateľského rozhrania. Táto časť obsahuje aplikácie pre podnikateľov týkajúce sa:

- intranetové novinky pre zamestnancov,
- malý inovačný systém,
- riadenie obchodných procesov s pracovnými postupmi,
- výpočet kusovníka,
- systém recenzovania kníh,
- riadenie výživy a stravovania,
- plánovanie úradných hodín,
- fakturácia telekomunikačnej spoločnosti,
- požičovňa áut

pre komunity:

- katalóg rastlín,
- burza rastlín

a všeobecnú uplatniteľnú autorizáciu a správu používateľov. V každom biznis prípade preskúmate:

- Biznis pohľad na danú vec: Prehľad biznis situácie.
- Definícia problému: Hľadanie odpovede na otázku, kto a prečo potrebuje daný problém vyriešiť.
- Prípady použitia: Prezentujú sa tri typy popisu: naratívny, pološtruktúrovaný a grafický na prípravu dokumentácie prípadov použitia v jazyku UML.
- Logický a relačný dátový model: APEX ponúka všetky funkcie na vytváranie nových dátových štruktúr, používanie a úpravu existujúcich štruktúr, kombináciu s inými nástrojmi na modelovanie dát a podporu dopredného alebo spätného inžinierstva. Úsilie vývojárov o zabezpečenie vhodných častí dát a vzťahov medzi nimi a zohľadnenie biznis potrieb je základom pre napredovanie používateľského rozhrania.
- Aplikačné rozhrania: učebnica poskytuje HTML stránky, reporty, formuláre, polia, menu, tlačidlá, hypertextové odkazy, ktoré zhmotňujú biznis situáciu, riešenie biznis problému, prípady použitia a údaje s ohľadom na koncového používateľa.
- Doplnkový učebný materiál: V každej kapitole nájdete odkazy na exportované aplikácie, skripty, údaje a videonávody, ktoré vám pomôžu pri vývoji. Tieto zdroje vám poskytnú praktické poznatky, ktoré vám umožnia upevniť si vedomosti a aplikovať ich priamo na reálne projekty.

Či už ste skúsený vývojár, ktorý chce rozšíriť svoje zručnosti, alebo začiatočník, ktorý túži preskúmať svet APEX, táto učebnica je vaším dokonalým sprievodcom. Dúfame, že aj pre študentov, ktorí nie sú odborníkmi na IT, bude neoceniteľným spoločníkom na ceste k zvládnutiu systému

Oracle APEX a vytváraniu inovatívnych aplikácií, ktoré majú pozitívny vplyv.

Učebnica a doplnkové materiály sú určené na približne 75 hodín práce študentov (3 ECTS). Dúfame, že je možné uplatniť rôzne spôsoby štúdia:

- prednášky a laboratórne cvičenia vedené učiteľom,
- kombinované vzdelávanie a tiež
- samoštúdium.

V závislosti od základných znalostí študentov a dostupného času na vedenie kurzu môžu učitelia ľahko zostaviť sadu kapitol, ktorá vyhovuje učebným príležitostiam, ako sú: mimoškolské kurzy, letné školy, časovo obmedzené podujatia na oboznámenie sa s low-code pre všetkých študentov (nielen IT alebo CS) a odborníkov z praxe v rôznych odvetviach priemyslu.

Pri tvorbe tejto učebnice a doplnkových učebných materiálov boli použité verzie 22 a 23 systému APEX. Veríme, že pojmy a základné technológie vysvetlené a aplikované v jej obsahu budú prínosom aj pre žiakov budúcich verzií systému APEX.

Pripravte sa na vzrušujúce vzdelávacie dobrodružstvo so systémom Oracle APEX! Užite si používanie asistentov a low-coding!

Profesor Robert Leskovar
Vedúci projektu BeeAPEX, Vedúci katedry informatiky, UM, Faculty of Organizational Sciences

Ako na to v APEX

1 **Ako spustiť Oracle APEX?** 35
VJERAN STRAHONJA, DIJANA ORE[Š]KI . . . . . .

2 **Ako pripraviť databázu?** 49
E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER . . . . .

3 **Ako sa pohybovať v nástroji APEX?** . . . 81
E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER . . . . .

4 **Ako si vymieňať dáta v APEX-e?** 93
E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER . . . . .

5 **Ako vygenerovať prvý návrh aplikácie?** 113
A. ANGEIOPLASTIS, G. MYLLIS, A. TSIMPIRIS, D. VARSAMIS . .

6 **Ako spravovať reporty?** 135
ZUZANA ŽILLOVÁ, ERIK MALINA, MATEJ GROCHAL, ANDREJ STANÍK, ANDREA MELEKOVÁ, MICHAL KVET, MIROSLAV POTOČÁR . . .

7 **Ako spravovať formuláre?** 157
VERONIKA ŠALGOVÁ, MICHAL KVET, MIROSLAV POTOČÁR . .

8 **Ako previesť textové reporty na grafy?** 165
IVAN PASTIERIK, MICHAL KVET, MIROSLAV POTOČÁR . . . . . .

9 **Ako spravovať menu?** 183
VERONIKA ŠALGOVÁ, MICHAL KVET, MIROSLAV POTOČÁR . .

10 **Ako spolupracovať v tíme?** 187
P. STANISZEWSKI, M. SOŃTA, A. KIERZKOWSKI . . . . . .

11 **Ako využiť galériu vzorových aplikácií a pluginov?** 199
VJERAN STRAHONJA, DIJANA ORE[Š]KI . . . . . .

12 **Ako spravovať zabalené a viacjazyčné aplikácie?** 207
R. LESKOVAR, U. RAJKOVI[Č], A.BAGGIA . . . . . .



1. Ako spustiť Oracle APEX?

VJERAN STRAHONJA, DIJANA OREŠKI

1.1 Čo je Oracle APEX a na čo slúži?

1.1.1 Čo je Oracle APEX?

Oracle APEX (tiež známy ako APEX alebo Oracle Application Express) je podniková low-code vývojová platforma vyvinutá Oracle Corporation, ktorá umožňuje používateľom vytvárať a nasadzovať aplikácie využívajúce webové rozhranie. Ide o súbor nástrojov na vytváranie rôznych typov aplikácií, ktoré pozostávajú z programov na prácu s dátami v databáze vrátane zadávania, ukladania, zobrazovania, zmeny, výpočtu a logických operácií s dátami atď. Využitie tejto platformy umožňuje vývoj vlastných aplikácií na rôzne účely, ktoré zahŕňajú generovanie databázy na základe súboru štruktúrovaných dát alebo dátových modelov, prácu priamo s dátami v databáze pomocou jazyka SQL, vytváranie užívateľských rozhraní, generovanie rôznych tabuľkových a grafických reportov a nasadenie webových aplikácií v databázach Oracle.

1.1.2 A čo je aplikácia?

Aplikácia je vo všeobecnosti súbor počítačových programov, ktorý má špecifický účel a pracuje so súborom údajov v databáze. Zaujímajú nás najmä obchodné aplikácie, teda tie, ktorých programy sa používajú na spracovanie údajov o obchodných procesoch a transakciách. Vývoj aplikácie, aj jednoduchej, je zložitý myšlienkový a tvorivý proces, ktorý si vyžaduje určité znalosti a zručnosti. V prvom rade je potrebný algoritmický spôsob myslenia (algoritmická gramotnosť). Je to schopnosť konceptualizácie, uvedomenie si, že v danej situácii existuje určitý algoritmus, schopnosť ho aplikovať, znalosť ako navrhnúť a použiť algoritmus a schopnosť kriticky hodnotiť algoritmy. Potrebné sú aj znalosti a zručnosti tvorby kvalitných záznamov v programovacom jazyku (kódovanie), testovanie a odladťovanie a ďalšie znalosti, ktoré zvyčajne nazývame programovanie.

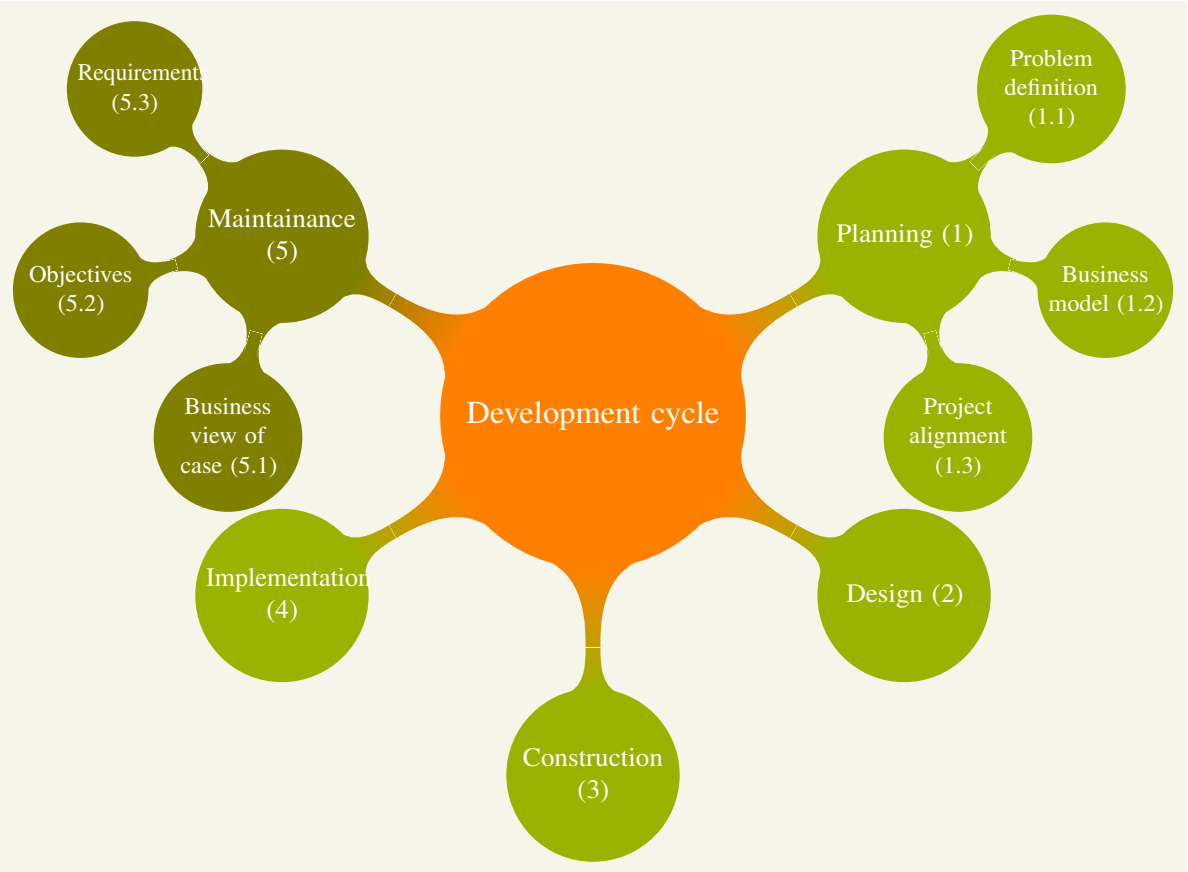
1.1.3 Ako môže Oracle APEX pomôcť pri vývoji aplikácií?

Pri vývoji počítačových programov sa nevyhneme algoritmickému spôsobu myslenia. Používanie platforiem a nástrojov na programovanie pre low-code / no-code programovanie, ako je napríklad Oracle APEX, umožňuje neprogramátorom jednoducho programovať bez toho, aby museli poznať detaily programovacieho jazyka a mať pokročilé pogramovacie znalosti. Znamená to, že sa môžu sústrediť na problém a algoritmus, a nie na programovací jazyk a technológiu. Zaujíma nás najmä rýchly, užívateľsky prívetivý vývoj aplikácií pre prístup, zjednotenie, analýzu a zobrazenie dát

z otvorených dátových zdrojov a z podnikových databáz. Otvorená dáta (open data) sú voľne dostupné dáta, ku ktorým má ktokoľvek prístup, môže ich používať a zdieľať. Môžu byť použité bez obmedzení (alebo s niektorými), zdieľané s ostatnými alebo použité na vytváranie nových diel. Sprístupňujú ich vlády, spoločnosti a iné organizácie na podporu transparentnosti a spolupráce. Dnes existuje veľa ľahko dostupných zdrojov meteorologických, dopravných, geologických zdravotných a podobných údajov, ako aj štatistík o školstve, zdravotníctve, hospodárstve atď. Podnikové databázy obsahujú údaje o obchodných transakciách a ďalšie údaje, ktoré vznikajú ako výsledok obchodných procesov. Vizualizáciu, analýzu, integráciu a rôzne pohľady na tieto údaje je možné použiť na prijímanie manažérskych rozhodnutí alebo na dosiahnutie niektorých ďalších výhod. Okrem používateľského vývoja sa Oracle APEX používa aj na vývoj prototypov, pomocou ktorých sa rozhoduje o tom, čo má program robiť a ukazuje IT špecialistovi, čo používateľ chce. Prototyp odráža základné vlastnosti systému. Neskôr je pridaný do systému alebo slúži ako základ pre vytvorenie systému v inom programovacom jazyku, nástroji alebo platforme.

1.1.4 Cyklus vývoja aplikácie

Bez ohľadu na to, aký druh aplikácie vyvíjame a aký druh metodického rámca, vývojových nástrojov alebo platformy používame, cyklus vývoja aplikácie prechádza niekoľkými fázami. Tieto štádiá sa líšia v rôznych metodických prístupoch. Obrázok 1.1 ukazuje jeden možný vzor cyklu vývoja aplikácie (viď. Obrázok 1.1).



Obr. 1.1: Cyklus vývoja aplikácie.

Ako je znázornené na obrázku, cyklus vývoja aplikácie obsahuje nasledujúce fázy alebo procesy:

1. Plánovanie (definovanie problému a projektu)

- definícia a analýza biznis modelu
- špecifikácia požiadavky
- stanovenie základných procesov a údajov
- technologické a architektonické rozhodnutia
- plánovanie vývoja

2. Dizajn aplikácie
   - analýza a špecifikácia požiadaviek
   - logický dizajn aplikácie (prípady použitia a logický dátový model)
3. Vytvorenie aplikácie
   - návrh databázy
   - vývoj (prototyp) aplikácie
   - testovanie
4. Implementácia aplikácie

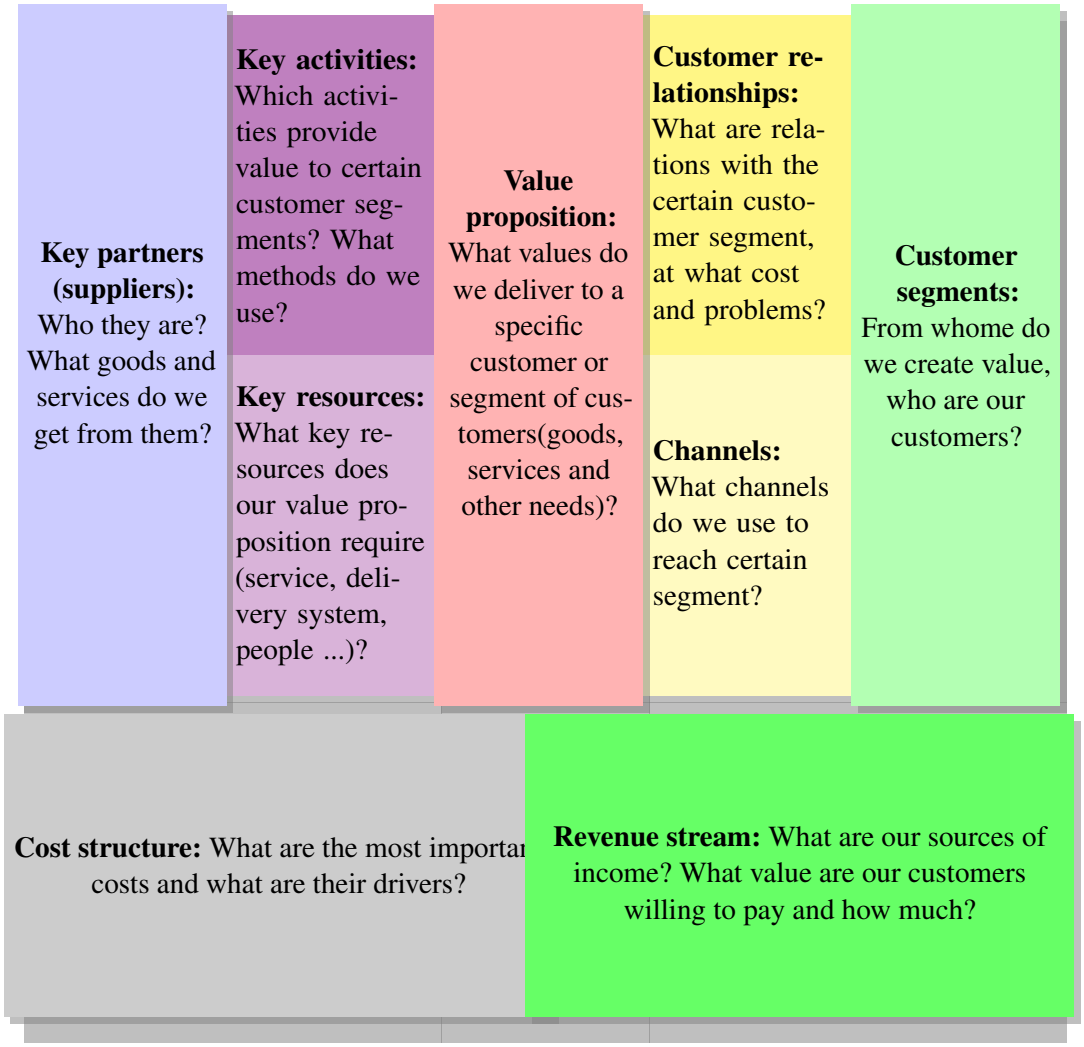
Po plánovaní, nasleduje návrh, vytvorenie a implementácia. Tieto fázy alebo procesy je možné vykonávať sekvenčne alebo iteratívne (evolučne, inkrementálne), a to tak, že celý vývojový cyklus prejdeme niekoľkokrát a v každej iterácii pridáme časť funkcionality.

Plánovanie vývoja aplikácie Cyklus vývoja aplikácie začína plánovaním. Pri jednoduchých aplikáciách môže byť počiatočnou informáciou pre plánovanie aplikácie definícia problému, dokument, ktorý jasne uvádza problém. Koho tento problém trápi? Prečo je to pre nich problém? Ako sa tento problém prejavuje? Môže to byť niekoľko viet, ale aj niekoľko strán textu.

V prípade komplexných podnikových aplikácií sú to však podrobné špecifikácie, ako napríklad:

- Podnikový pohľad na prípad popisuje ako ľudia v podniku vnímajú situácie alebo aké očakávania majú pred vývojom aplikácie.
- Ciele vývoja aplikácie popisujú, čo chce aplikácia dosiahnuť z podnikového, technologického a iného hľadiska a ukazovatele úspešnosti.
- Biznis model popisuje hodnoty, zákaznícke segmenty, zákaznícke vzťahy a kanály, kľúčových partnerov, aktivity a zdroje, štruktúru a zdroje výnosov a nákladov atď.
- Špecifikácia softvérových požiadaviek definuje požiadavky týkajúce sa funkcie, správania, výkonu, technológie atď.
- Projektový plán popisuje rozsah a fázy projektu, činnosti (úlohy), ktoré sa vykonávajú v jednotlivých fázach, vstupy a výstupy jednotlivých činností, zdroje, ktoré sú potrebné, prepojenie činností a ich trvanie, náklady, riziká , atď.

Na vytvorenie každej z uvedených špecifikácií existujú metódy, ktoré sú opísané v literatúre. Napríklad, Alexander Osterwalder [6] vyvinul metódu s názvom Business Model Canvas, ktorá sa používa na vývoj, zlepšovanie a dokumentovanie nových a existujúcich obchodných modelov. Ako je znázornené na Obrázku 1.2, je to v skutočnosti tabuľka s 9 blokmi, ktoré opisujú hodnotovú ponuku poskytovanú podnikom, jej zákazníkmi, vzťahmi a kanálmi s nimi, kľúčovými partnermi, aktivitami a zdrojmi ako aj nákladovú štruktúru a toky príjmov.



Obr. 1.2: Business Model Canvas [6].

Zatiaľ čo BMC slúži na podporu pochopenia, analýzy a návrhu obchodného modelu a najmä toho, čo potrebuje podporovať aplikácia, ktorú vyvíjame, Software Requirements Specification (SRS) je dokument, ktorý popisuje, čo bude softvér robiť a ako aký výkon sa od neho očakáva. Popisuje funkčnosť, výkon a ďalšie vlastnosti, ktoré musí mať aplikácia, aby podporovala obchodný model a potreby všetkých používateľov.

Dizajn aplikácie Od prvých počítačov a počítačových programov až po dnešok boli vyvinuté stovky rôznych metodických prístupov, metód a techník návrhu aplikácií. Technologickým vývojom sa zmenila metodika tvorby programov. Vývoj programov pomocou Low-code / no-code je prístup, ktorý preferuje jednoduché metódy, zrozumiteľné pre používateľov, ktorí nie sú profesionálnymi vývojármi softvéru. Vývoj aplikácií si však pred vygenerovaním programu vyžaduje určitý návrh. Ako sme už spomenuli, počítačová aplikácia je súbor programov, ktoré pracujú s dátami v databáze. Aby sme mohli vyvinúť aplikáciu, musíme navrhnúť programy aj databázu. Ak už databáza existuje, musíme ju vedieť používať a predovšetkým rozumieť dátovému modelu. Pre potreby vývoja low-code / no-code programu uvedieme najnutnejšie metódy návrhu. Základom dizajnu je zároveň tvorba modelov, ktoré majú aj grafické znázornenie.

Nasledujúce metódy budú použité:

- Use Case (UC) model, ktorý definuje vzťah medzi aplikáciou a prostredím.
- Model vzťahu entít (entity-relationship (ER)) model, ktorý popisuje štruktúru údajov na logickej úrovni.

Prípad použitia (Use Case) bude vysvetlený v pokračovaní tejto kapitoly, zatiaľ čo entity-relationship (ER) bude vysvetlený v kapitole 2.

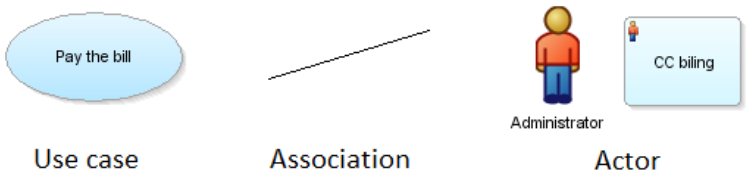
Use Case Model (UC) Use Cases sú jednou z metód vývoja softvéru. Popisujú vzťah medzi systémom (aplikáciou) a prostredím (používatelia a iné systémy). Metóda je jednoduchá, takže jej rozumejú aj neprofesionálni používatelia. Use Cases sú neoddeliteľnou súčasťou Unified Modeling Language (UML), dnes najpoužívanejšieho modelovacieho jazyka v oblasti softvérového inžinierstva, ktorý poskytuje štandardný spôsob vizualizácie návrhu systému. Existuje množstvo UML literatúry a Object Management Group (OMG) sa stará o jeho vývoj a štandardizáciu. UML obsahuje 13 metód a diagramových techník na modelovanie štruktúry, správania a interakcie softvéru. My budeme používať iba prípady použitia (Use Cases).

Use Case Model pozostáva z dvoch častí:

- Use case špecifikácie (naratívny popis)
- Use case diagramu

Najprv si preštudujeme Use Case diagram. Tento diagram popisuje, čo systém robí z pohľadu pozorovateľa mimo systému. Nezáleží na tom, ako systém interne funguje.

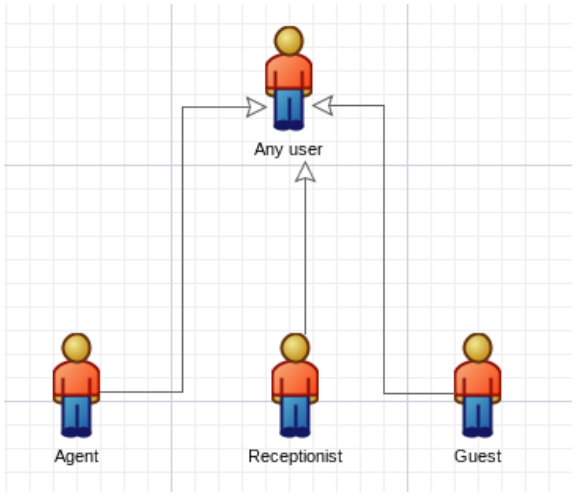
Základné koncepty Use Case diagramu sú znázornené na Obrázku 1.3.



Obr. 1.3: Základné koncepty Use Case diagramu.

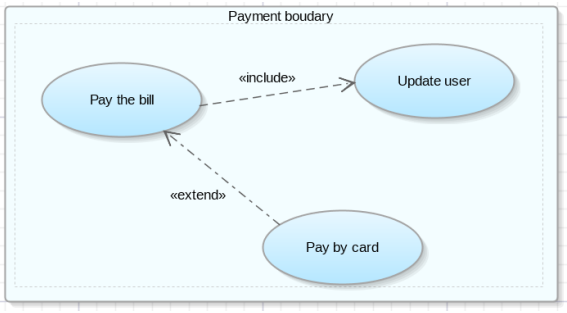
Prípad použitia (use case) predstavuje súbor činností, ktoré môže systém (napr. softvérový komponent) vykonať v interakcii s externými aktérmi. Je to príbeh, ktorý popisuje, ako aktéri využívajú systém na dosiahnutie určitých cieľov alebo na plnenie úloh. Predstavuje abstraktnú úlohu, ktorá má svoj účel. Prepojenie (Asociácia) spája účastníkov komunikácie, napríklad aktéra a prípad použitia, a predstavuje ich interakciu a vzťah medzi systémom a prostredím (správanie). Aktér predstavuje súbor rolí, ktoré interagujú so systémom rovnakým spôsobom, napríklad trieda používateľov, nejaký druh externého systému atď. Aktér je niekto mimo diskutovaného systému, ktorý s ním interaguje, ale nie je jeho súčasťou. Môže to byť živá bytosť (používateľ, pacient, pilot) alebo iný systém (fakturačný systém, banka, dopravca).

Diagram môže tiež znázorňovať klasifikačnú štruktúru (viď Obrázok 1.4), teda zovšeobecnenie/špecializáciu pojmov, napríklad aktérov. Všetky podtriedy (deti) dedia vlastnosti a správanie od nadtriedy (rodiča), čo znamená, že od rodiča dedia aj odkazy na prípady použitia. Zároveň môže jednotlivá podtrieda vstúpiť do komunikácie, do ktorej nevstupuje ani nadtrieda, ani iné podtriedy.



Obr. 1.4: Zovšeobecnenie/špecializácia hercov.

Obrázok 1.5 zobrazuje hranice systému ako obdĺžnik s názvom systému, v ktorom sa nachádzajú prípady použitia.

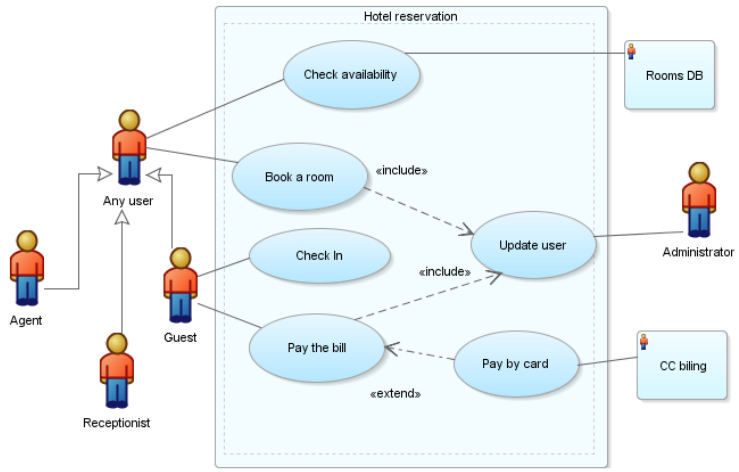


Obr. 1.5: Hranice systému a asociácie “include” a “extend”.

UC diagramy nezobrazujú pracovný tok ani postupnosť prípadov použitia. Existujú však dva povolené typy asociácií medzi prípadmi použitia, ktoré sú znázornené na Obrázku 1.5:

- “include” je priradenie od základného prípadu použitia k zahrnutému prípadu použitia, čo naznačuje, že základný prípad použitia obsahuje správanie zahrnutého prípadu použitia. Týmto spôsobom sú často používané funkcie a správanie rozdelené do prípadov použitia, ktoré podľa potreby zahrnieme do iných prípadov použitia. Všimnite si, že základný prípad použitia nie je úplný bez zahrnutia.
- “extend” je asociácia z prípadu použitia, ktorý je rozšírením, k základnému prípadu použitia, čo naznačuje, že správanie základného prípadu použitia je rozšírené o správanie rozšírenia. Rozšírenie zároveň nie je súčasťou základného prípadu použitia, ani by táto komunikácia nemala vždy existovať a fungovať bezchybne.

Použitie vyššie uvedených konceptov Use Cases diagramu je najlepšie vysvetliť na príklade, akým je hotelový rezervačný systém zobrazený na Obrázku 1.6.



Obr. 1.6: Use Cases diagram hotelového rezervačného systému.

Systém, ktorého správanie sledujeme, je hotelová rezervácia. Celý systém je v rámci hraníc znázornených obdĺžnikom. Okolie systému je mimo obdĺžnika. V rámci hraníc systému je šesť prípadov použitia, znázornených elipsami s názvami. Mimo systému sú aktéri, ktorí komunikujú so systémom, teda jeho prípadmi použitia a táto komunikácia je znázornená spojovacími čiarami, asociáciami. Každý používateľ je účastníkom, ktorý komunikuje s Check Availability. Tento prípad použitia komunikuje s externým aktérom Rooms DB. Je to nejaký externý rezervačný systém, ktorý má vo svojej databáze informácie o izbách a rezerváciách. Každý užívateľ môže rezervovať izbu. Tento UC zahŕňa ďalší UC - Update User. Štruktúra klasifikácie užívateľského aktéra je znázornená vľavo. Agent, Receptionist a Guest sú druhom aktéra Any user. V zobrazenom prípade môže hosť komunikovať s UC Check In (ubytovať sa) a Pay the bill (zaplatiť účet). Používateľ aktualizácie UC je zahrnutý v časti Zaplatiť účet, čo znamená, že údaje používateľa možno aktualizovať aj v rámci funkcie platby faktúr. UC Update User je zahrnutý v Pay the bill, čo znamená, že používateľské údaje možno aktualizovať aj v rámci funkcie platby faktúr.

UC Pay by card rozširuje funkcionalitu Pay the bill, čo znamená, že platbu kartou je možné vykonať počas Pay the bill (platenia účtu). Pay by card komunikuje s externou službou platby platobnou kartou CC billing. Administrátor môže komunikovať s UC Update user a aktalizovať informácie o používateľovi.

Ako sme už spomenuli, use case model pozostáva z dvoch častí:

- Use case špecifikácie (naratívny popis)
- Use case diagramu.

Use case špecifikácia popisuje scenáre a internú logiku prípadov použitia, počiatočné stavy (predpoklady) a konečné stavy, rozhrania, systémové správy, špecifikáciu spracovania chýb a výnimiek a podobne. Zvyčajne sa pre Use Case špecifikáciu používa nejaký druh šablóny, napr. taký, ktorý budeme používať v tejto knihe.

Vytvorenie, implementácia a údržba aplikácie Vytvorenie aplikácie zahŕňa vytvorenie samotného programu a fyzickej databázy. Použijeme na to platformu Oracle APEX, ako bude vysvetlené v knihe neskôr. Prvá verzia aplikácie, ktorá funguje a je možné ju testovať, je vlastne prototypom aplikácie. Tento prototyp je možné upgradovať a rozšíriť.

Vzhľadom na to, čo sa stane s prototypom neskôr, rozlišujeme dva typy prototypov: Rozšíriteľný prototyp je vylepšený a upgradovaný do stavu, ktorý je vhodný na použitie. “Throw away” prototyp slúži ako ilustrácia funkcionality a modelu, na ktorom bude aplikácia postavená, zvyčajne na inej platforme. Ak chceme prototyp aplikácie použiť, musíme ho implementovať, to znamená nainštalovať do produkčného prostredia, edukovať používateľov, pripraviť databázu a podobne.

Po fáze implementácie nasleduje používanie a údržba, ktorá zahŕňa opravu a odstránenie chýb, zlepšenie výkonu a menšie rozšírenia funkcionality. Ak sú potrebné zásadné zmeny, spustí sa nový vývojový cyklus aplikácie.

1.2 Ako spustiť Oracle APEX?

Vývoj softvéru sa považuje za zložitý kvôli veľkému objemu programovania. Na prekonanie zložitosti sa navrhujú riešenia založené na low code prístupe. Low code prístup umožňuje vývoj sofistikovaných aplikácií, ktoré sú funkčné a zložité bez písania riadku kódu. Oracle Application Express je na to jednou z najlepších možností. Existujú rôzne spôsoby, ako spustiť Oracle APEX. Oracle APEX je možné použiť kdekoľvek, kde je spustená databáza, a to lokálne aj v súkromnom cloude. Môže to byť fyzický dedikovaný server, virtualizovaný stroj alebo docker image (ktorý majú radi najmä vývojári Oracle Application Express a možno ho spustiť na prenosnom počítači, v autobuse alebo v lietadle). Okrem toho je možné využiť Exadata, super výkonný fyzický APEX server v cloudových službách. Všetky možnosti sú popísané v tejto knihe. To ako sa dostať do vývojového prostredia je popísané v tejto kapitole.

1.2.1 Aká je úroveň vašich zručností?

Existuje niekoľko možností ako začať s Oracle Application Express - APEX. Vaša aktuálna úroveň zručností určuje odporúčaný spôsob akým začať. Predpokladajme, že máte kompetencie v špecifických doménach v akomkoľvek podnikovom procese (napr. predaj, výroba, prenájom marketing, právne záležitosti, zdravotná starostlivosť, stavebníctvo, vzdelávanie) a nasledujúce úrovne digitálnych zručností:

- absolútny začiatočník: skúsený s používaním webového prehliadača, žiadne skúsenosti s programovaním; odporúčanie: použite inštanciu on-premise alebo apex.oracle.com alebo Oracle Academy
- začiatočník: skúsený s používaním webového prehliadača, málo skúseností s programovaním napr. používanie tabuliek a nastavovacích vzorcov; akýkoľvek iný programovací jazyk; odporúčanie: použite inštanciu on-premise alebo apex.oracle.com alebo Oracle Academy
- vývojár nováčik: skúsený s používaním webového prehliadača, malé skúsenosti s programovaním; chápe, čo je databáza, tabuľka, stĺpec tabuľky, primárny kľúč, cudzí kľúč; základné dopytovanie, vkladanie, aktualizácia a mazanie; odporúčanie: použite lokálnu inštanciu alebo apex.oracle.com alebo Oracle Academy alebo zariadenie VBox
- skúsený vývojár: skúsený s používaním webového prehliadača, pokročilé znalosti v oblasti programovania postupov a funkcií v akomkoľvek systéme správy databáz; pokročilé dopytovanie, vkladanie, aktualizácia a mazanie; odporúčanie: použite lokálnu inštanciu alebo apex.oracle.com alebo Oracle Academy alebo zariadenie VBox alebo Apex docker
- Takmer profesionálny vývojár: rozvoj zručností v PL/SQL a Javascript; odporúčanie: použite lokálnu inštanciu alebo apex.oracle.com alebo APEX docker alebo OCI APEX instance

Vaša používateľská rola a umiestnenie Oracle APEX určia, ako sa prihlásite do aplikácie a ako ju používate. Oracle APEX je možné nainštalovať lokálne na vašom počítači alebo v hosťovanom prostredí, ako je služba Oracle Cloud. V závislosti od typu inštalácie sa postup prihlásenia líši. Používatelia si pred vývojom alebo inštaláciou aplikácií musia vytvoriť pracovný priestor, pridať používateľov Oracle APEX a potom sa prihlásiť do pracovného priestoru. Viacerí používatelia môžu spolupracovať na rovnakej inštalácii Oracle APEX pomocou pracovného priestoru pri zachovaní súkromia svojich objektov, údajov a aplikácií. Ak webový prehliadač podporuje JavaScript, používatelia sa môžu prihlásiť do pracovného priestoru, aby získali prístup na domovskú stránku Oracle APEX. Každý pracovný priestor má odlišný názov a ID. V rámci Oracle APEX Administration Services môže administrátor inštancie manuálne vytvoriť pracovný priestor alebo môžu používatelia zadávať požiadavky. Nezávislý program s názvom Oracle APEX Administration

Services sa používa na dohľad nad kompletnou inštanciou Oracle APEX.

1.2.2 Použitie lokálnej inštancie APEX

Táto časť popisuje, ako nainštalovať Oracle APEX lokálne. Inštalácia Oracle APEX zahŕňa niekoľko krokov. Tie kroky sú:

- **Vyberte si medzi Full alebo Runtime Environment**. Full Environment poskytuje úplný prístup k vývojovému prostrediu App Builder. Pre produkčné implementácie, kde chcete spúšťať nemenné aplikácie, je dobrou možnosťou Runtime Environment. Oracle APEX umožňuje inštaláciu iba runtime verzie Oracle APEX pre testovacie a produkčné inštancie. Keďže vývojári nemôžu náhodne aktualizovať produkčný program v inštancii runtime, Runtime Environemnt znižuje inštalovanú stopu a práva a zvyšuje bezpečnosť aplikácie. Používatelia môžu spúšťať produkčné aplikácie v Runtime prostredí Oracle APEX, ktoré však nemá webové rozhranie na správu. Runtime Environment je bezpečnejšie, pretože obsahuje iba komponenty potrebné na spustenie programu.
- **Overte požiadavky na inštaláciu** Overte, či váš systém spĺňa minimálne požiadavky na inštaláciu. Existuje päť skupín požiadaviek: a) Požiadavky na Oracle databázu (Oracle APEX 22 vyžaduje Oracle Database 12.1.0.2 alebo novšiu. Oracle APEX beží na všetkých edíciách databáz, vrátane Enterprise Edition (EE), Standard Edition (SE) a Express Edition (XE). Oracle APEX je možné nainštalovať do databázy s jednou inštanciou a do databázy Oracle Real Application Clusters (Oracle RAC)), b) Požiadavky na prehliadač (Oracle APEX vyžaduje prehliadač s JavaScriptom a podporuje aktuálne a predchádzajúce hlavné vydanie Google Chrome, Mozilla Firefox, Apple Safari a Microsoft Edge.), c) Web Listener požiadavky (Oracle APEX požaduje Oracle REST Data Services (ORDS) 19.x alebo novšiu verziu), d) Požiadavky na miesto na disku (Požiadavky Oracle APEX na miesto na disku sú: 310 MB voľného miesta pre softvérový súbory APEX v súborovm systéme ak sa sťahuje len v angličtine a 705 MB ak sa sťahuje celý balík, 220 MB voľného miesta v APEX tablespace, 100 MB voľného miesta v SYSTEM tablespace, 60 MB voľného miesta v APEX tablespace pre každý ďalší nainštalovaný jazyk (okrem angličtiny), e) Oracle XML DB požiadavky (Ak inštalujete úplné vývojové prostredie, Oracle XML DB musí byť nainštalovaný v Oracle databáze, ktorú chcete použiť. Ak používate predkonfigurovanú databázu vytvorenú počas inštalácie alebo pomocou nástroja Database Configuration Assistant alebo DBCA, Oracle XML DB je už nainštalovaný a nakonfigurovaný.)
- **Nainštalujte** softvér - Nainštalujte Oracle APEX stiahnutím ZIP súboru zo stránky Oracle APEX a a potom si stiahnite a nainštalujte Oracle REST Data Services (ORDS).

1.2.3 https://apex.oracle.com

Prejdite v prehliadači na toto URL. Požiadanie o bezplatný workspace je najrýchlejší spôsob, ako začať s Oracle APEX. Pripojenie trvá len niekoľko sekúnd a workspace je pripravený na to, aby používatelia mohli začať vyvíjať aplikácie. Tento prístup nevyžaduje cloudový účet a je bezplatný. Ak chcete začať s Oracle APEX:

- Zadajte do webového prehliadača: apex.oracle.com
- Kliknite na tlačidlo Start for Free Today
- Na webovej stránke, ktorá sa zobrazí, vyberte možnosť Request a free Workspace
- Zadajte svoje meno, e-mailovú adresu, pomenujte svoj workspace (pracovný priestor) a kliknite na next (ďalej)
- Vyplňte prieskum a uveďte, prečo požadujete workspace. Prečítajte si zmluvné podmienky a prijmite zmluvné podmienky, kliknite na tlačidlo next
- V tomto bode dostanete e-mail od Oracle APEX, kliknutím na odkaz sa zaregistrujte a nastavíte heslo

1.2.4 Zariadenie Virtual Box / Virtual Machine

Prejdite v prehliadači na toto URL. Testovacou prerekviziou je najnovšia APEX verzia. Stiahnite Oracle Virtual Box a Extension manager. Nastavte Oracle Virtual Box a Extension manager. Importujte .ova súbor. Spustite nástroj.

Oracle APEX Development VM je virtuálny nástroj, ktorý môžete použiť jednoduchým importovaním do VirtuallBox-u. Oracle Pre-Built Virtual Machines sú dostupné s účtami Oracle. Všetko, čo musíte urobiť, je nainštalovať Oracle Virtual Box (Free Virtual Machine Client) a potom importovať akékoľvek zariadenie (Pre-Built VM). Oracle má k dispozícii niekoľko predpripravených virtuálnych nástrojov dostupných na URL.

Ak chcete nastaviť VM, postupujte nasledovne:

- Stiahnite si a nainštalujte Oracle VM VirtualBox do svojho hostiteľského systému.
- Stiahnite si súbory (odporúča sa použiť downolad manažéra).
- Importujte VM: File > Import Appliance to launch Appliance Import Wizard. Kliknite Choose... prejdite do adresára, v ktorom ste znova zostavili všetky súbory, a vyberte OTN_Developer_Day_VM.ova. Potom kliknutím na tlačidlo Next spustite import virtuálneho počítača. Počas importu vás vyzve, aby ste vyjadrili súhlas s príslušnými vývojárskymi licenciami. Po dokončení importu sa zobrazí 'Oracle Developer Days (Powered Off).
- Otestujte svoju VM: Po dokončení importu dvakrát kliknite na OTN Developer Days VM. Kliknutím na tlačidlo OK zatvorte dialógové okná Virtualbox Information. Keď sa dostanete na obrazovku Enterprise Linux 6 môžete sa prihlásiť. (Používateľské meno a heslo je oracle.) Nechajte proces dokončiť; je pripravený, keď uvidíte okno terminálu, ktoré môžete následne zavrieť. Po dokončení práce na hosťujúcej VM ju môžete vypnúť cez Systém > Vypnúť (System > Shut Down); toto vráti hosťovanú VM do vypnutého stavu.

Medzi hlavné výhody využívania Virtual Machines patria:

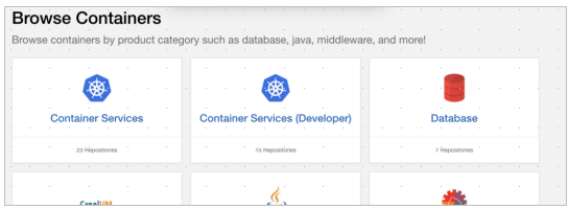
- Využívanie nového softvéru Oracle bez zložitých inštalácií alebo zapojenia dodávateľa.
- Možnosť experimentovania bez vystavenia vašej pracovnej stanice riziku.
- Bezplatné, vedené školenie poskytované spoločnosťou Oracle vo forme Hands On Labs.

1.2.5 APEX docker

Získajte oficiálny APEX docker na stránke Oracle. V prehliadači prejdite na toto URL.

Oracle APEX je možné nainštalovať ako Docker kontajner. Dockery sú pre vývojárov veľmi pohodlným spôsobom, pretože sa dajú nainštalovať na počítače a notebooky. Dôrazne odporúčame získať APEX docker len z dôveryhodných zdrojov. Inštalácia je zvyčajne rýchla a jednoduchá. Na učenie sa nie sú verzie Oracle Database a APEX veľmi dôležité ale na produkčné použitie je vhodné venovať verziám pozornosť. Zvyšok tejto časti opisuje jeden z možných spôsobov ako nainštalovať APEX docker lokálne na váš počítač. Pri tomto postupe klonujeme oficiálny Oracle docker obraz, ktorý obsahuje vývojové nástroje.

1. Prvým krokom je inštalácia nástroja docker do počítača. Docker je možné ľahko stiahnuť a nainštalovať z jeho domovskej stránky: `https://docs.docker.com/desktop/install/windows-install/`
2. Druhým krokom je zaregistrovanie bezplatného účtu v oracle container registry: `https://container-registry.oracle.com/ords/f?p=113:10` Docker obrazy sú vytiahnuté z tohto registra (viď. Obrázok 2).



3. Tretím krokom je otvorenie okna terminálu a prístup do registra Oracle pomocou predtým vytvoreného používateľa.

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

V prostredí dockera je lepšie vytvoriť sieť, aby dockery mohli komunikovať s inými dockermi pomocou názvu hostiteľa

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

V tomto okamihu sme urobili všetko potrebné a môžeme spustiť príkaz na spustenie databázy Oracle Database XE v dockeri na našom počítači.

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

Všimnite si, že boli použité parametre názov, názov hostiteľa a sieť. Vysvetlenie:
- Prvý príznak -d spustí kontajner v odpojenom režime.
- Parameter –name určuje názov kontajnera.
- -p mapuje port 1521 na hostiteľskom počítači na port 1521 v kontajneri, aby sme sa mohli pripojiť k databáze.
- –network pripojí kontajner k sieti, ktorú sme vytvorili.
- –hostname definuje meno DB servera.
- Posledný parameter je obraz, ktorý chceme použiť na spustenie kontajnera.

Ak sa chcete pripojiť pomocou sqlplus, môžete vykonať tento príkaz:

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

Pomocou príkazu *show pdbs* zistíte stav pdbs. Zmeňte predvolené heslo pomocou nižšie uvedeného príkazu (v tomto prípade bude nové heslo Welcome1!!). Pred vykonaním tohto príkazu sa uistite, že pdb XEPDB1 je otvorená na čítanie a zápis.

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

Ak chcete nastaviť inštaláciu Apex for Oracle XE v dockeri, vykonajte ďalšie kroky. Z relácie terminálu vytiahnite obraz ords.

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

Potom vykonajte príkaz pull na stiahnutie obrazu ords.

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

Ak chcete nakonfigurovať inštaláciu Apex, vytvorte adresár.

```
mkdir ~/APEX
```

Informácie o reťazci vložte do súboru v tomto adresári.

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

Všimnite si, že bol použitý hostname parameter Oracle Database XE docker a služba pdb pre reťazcové pripojenie. Definujte premennú CONN_STRING takto. Musí byť v nasledujúcom tvare (bez jednoduchých úvodzoviek):

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

V tomto okamihu môžeme spustiť 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
```

V tomto okamihu môžeme spustiť 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
```

Bola tu použitá rovnaká sieť parametrov. Ak chcete sledovať inštaláciu, môžete otvoriť ďalšie relácie terminálu a vykonať tento príkaz:

```
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
```

Zmeňte heslo pre APEX_PUBLIC_USER user v databáze. Prihláste sa do Oracle XE docker pomocou sqlplus.

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

Nastavte heslo.

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

Teraz sa môžete pripojiť k prostrediu APEX na lokálnom počítači pomocou prehliadača.

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

1.2.6 APEX inštancia v službe Oracle Cloud Infrastructure

Oracle Cloud Infrastructure (OCI) ponúka vývoj low-code aplikácií APEX na autonómnej infraštruktúre ako plne spravovanú službu, ktorá je vopred nakonfigurovaná a pripravená na používanie. OCI poskytuje elastickú škálovateľnosť, bezpečnosť, vysokú dostupnosť a globálny prístup prostredníctvom regionálnych cloudových dátových centier. Na vytvorenie inštancie služby APEX sú v tejto časti opísané postupy na získanie prístupu k Oracle Cloud Infrastructure (OCI). Kroky spojené s prístupom k OCI sú nasledovné:

1. **Získajte OCI účet**. Používateľ musí mať OCI účet, aby mohol používať APEX Service. Používateľ môže použiť existujúce konto OCI, ak má organizácia uzatvorenú predajnú zmluvu so spoločnosťou Oracle. Ak chcete získať prístup, najprv sa spojte so správcom OCI pre vašu spoločnosť. Ak ste jediný používateľ, ktorý začína od nuly alebo si nie ste istí, kde začať, zvážte možnosť registrácie do Oracle Cloud Free Tier. Okrem bezplatného počiatočného pridelenia Cloud kreditov ponúka aj bezplatný prenájom a účet OCI, bez obmedzenej platnosti. Počas skúšobnej lehoty môžete tieto kredity použiť na nákup platených služieb OCI, napríklad APEX Service. Ak chcete APEX Service používať aj po skončení skúšobnej lehoty alebo po vyčerpaní bezplatných kreditov (podľa toho, čo nastane skôr), musíte zmeniť svoj účet na platený a zakúpiť si ďalšie kredity. Ak neprejdete na platený účet pred uplynutím skúšobnej doby, vaše konto OCI sa zmení na také, v ktorom môžete využívať len služby OCI, ktoré majú k dispozícii oklieštenú formu s názvom Always Free. Nižšia Always Free verzia je k dispozícii v službe APEX. Spoločnosť Oracle odporúča, aby ste svoje

konto zmenili na platené a zakúpili si ďalšie kredity buď počas skúšobného obdobia, alebo po jeho skončení. Začnite proces registrácie tým, že si pozriete Oracle Cloud Infrastructure Free Tier. Ak chcete začať registráciu, prejdite na adresu https://signup.oraclecloud.com/.

2. **Prihláste sa do OCI Console.** Pomocou kompatibilného webového prehliadača vyhľadajte stránku na prihlásenie do OCI Console. Zadajte svoje prihlasovacie meno a heslo, za ktorým nasleduje názov vášho cloudového účtu (známy aj ako tenancy). Váš uvítací e-mail obsahuje vaše používateľské meno aj názov cloudového účtu.

1.2.7 APEX inštancia v Oracle Academy

Oracle Academy im poskytuje inštitucionálnym členom a ich študentom prístup k Oracle Application Express (APEX) na praktické cvičenia v cloude. Oracle Application Express (APEX) im sprístupňuje Oracle Academy na experimentovanie s využitím cloudu. S cieľom podporiť laboratóriá a aplikovanú prax pre učebné osnovy, ktoré zahŕňajú základy databáz, návrh a programovanie databáz s SQL, programovanie s PL/SQL a základy vývoja aplikácií s Oracle APEX, ponúka Oracle Academy pedagógom vyhradenú inštanciu určenú špeciálne na používanie v triede s maximálne 99 študentmi. Ak chcú mať používatelia prístup k Oracle APEX prostredníctvom Oracle Academy, musia sa prihlásiť do členského centra (Member Hub). Ak nie sú členmi, aby mali prístup ku všetkým materiálom a získali ďalšie výhody, mali by sa používatelia bezplatne zaregistrovať ako inštitucionálni členovia. Prejdite vo svojom prehliadači na adresu: https://academy.oracle.com/en/oa-web-overview.html.

1.3 Otázky

1. Aké sú fázy vývojového cyklu aplikácie a aké činnosti sú realizované v jeho tretej fáze?
2. torá z možností začať s Oracle Application Express je najlepším riešením pre úplného začiatočníka?
3. Aké sú výhody používania inštancie APEX v Oracle Cloud Infrastructure?

1.4 Odpovede

1. Fázy vývojového cyklu aplikácie sú: (i) plánovanie (definovanie problému a projektu), (ii) návrh aplikácie, (iii) vytvorenie aplikácie, (iv) implementácia aplikácie. Činnosti tretej fázy (vytvorenia aplikácie) sú: návrh databázy, vývoj (prototyp) aplikácie a testovanie.
2. Odporúčanie pre úplných začiatočníkov je použiť lokálnu inštanciu alebo apex.oracle.com alebo Oracle Academy.
3. Oracle Cloud Infrastructure (OCI) ponúka vývoj low-code aplikácií APEX na autonómnej infraštruktúre ako plne spravovanej službe, ktorá je vopred nakonfigurovaná a pripravená na používanie. OCI poskytuje pružnú škálovateľnosť, bezpečnosť, vysokú dostupnosť a globálny prístup prostredníctvom regionálnych cloudových dátových centier.

2. Ako pripraviť databázu?

E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER

Predpokladajte, že pracujete pre *Walmart* a *Database Systems* (DBS) ešte neboli vynájdené. Ďalej predpokladajte, že Vás požiadali o implementáciu *Webovej aplikácie*, ktorá umožní ukladať, získavať, vizualizovať a ďalej spracovávať každý jeden predaj v každom z 10 500 obchodov Walmart spolu s informáciami o zodpovedných oddeleniach, zamestnancoch a ich pracovných pozíciach, ktoré by mohli predstavovať Petabajty údajov. Úloha, ktorá sa zjavne nedá zvládnuť, pretože čelíte mnohým výzvam:

- Koľko súborov a diskov potrebujete na uloženie?
- Ako vyhľadáte a získate údaje?
- Ako zabezpečíte adekvátne časy odozvy?
- Ako povolíte úpravy a súbežný prístup k údajom?
- Ako zabrátnite neoprávnenému prístupu k údajom?

Ak by ste použili DBS ako základnú chrbticu Vašej Webovej aplikácie (pozri Obrázok 2.1), tak by sa tieto výzvy dali zvládnuť, pretože DBS poskytuje správnu funkčnosť na ich riešenie.



Obr. 2.1: DBS ako chrbtica Webových aplikácií.

DBS obsahuje:

1. softvér, ktorý pomáha pri správe a spracovaní veľkého množstva údajov, ktoré môžu byť používané Webovou aplikáciou, nazývaný Systém správy databáz (Database Management System -DBMS)
2. samotné ukladanie údajov, nazývané databáza (DB), ktorá poskytuje DB-vrstvu, na ktorej je budovaná Webová aplikácia.

V nasledujúcom texte je podrobnejšie vysvetlený hlavný koncept vytvorenia DB-vrstvy , t.j., *vývoj DB*. Hoci je to všetko o "*vytváraní vhodných DB tabuliek, ukladaní údajov, manipulácii s nimi a ich dotazovaní*", existuje niekoľko dôležitých konceptov, ktoré je potrebné zaviesť na to, aby sa tento cieľ dosiahol.

Konkrétne začneme v Časti 2.1 s procesom modelovania dát, ktoré majú byť ukladané, zavedením dvoch rôznych *úrovní abstrakcie* , ktoré obsahujú *logický model* a *fyzický model* v zmysle *relačného modelu*[1]. Ďalej, Časť 2.2, je zameraná na mechanizmy a nástroje pre *vytvorenie dátovej štruktúry v rámci DBS*, ako je predpísané procesom modelovania. Na základe toho sa Časť 2.3 zaoberá otázkou ukladania a manipulácie s údajmi a Časť 2.4 sa zameriava na získanie uložených údajov. Nakoniec, Časť 2.5 poskytuje súhrn všetkých možností a nástrojov, ktoré sú k dispozícii na vytvorenie DB-vrstvy Webovej aplikácie.

Vo všetkých týchto častiach je použitý *príklad*, inšpirovaný prípadom použitia spoločnosti Walmart, uvedeným vyššie, ktorý zdôrazňuje aspekt *Ľudských Zdrojov (HR)* z hľadiska *zamestnancov*, *oddelení* a *pracovných miest*, štandardný príklad, ktorý používa aj ORACLE na demonštračné účely od prvej verzie ich DBS v roku 1979.

2.1 Ako modelovať dáta

Prvým krokom pri realizácii DB-Vrstvy Webovej aplikácie je zamerať sa na reálne dáta, ktoré by mali byť ukladané. Keďže tieto údaje môžu byť komplexné, ako sme videli v našom príklade pre Walmart vyššie, má zmysel zaoberať sa údajmi najprv na abstraktnejšej úrovni, pričom zatiaľ nezohľadňujeme všetky podrobnosti nízko-úrovňovej implementácie určitého DBS. Toto je súčasťou logického modelu, ako je uvedené v Časti 2.1.1. Potom, na konkrétnejšej a teda špecifickejšej úrovni DBS, je relačný model predstavený v Časti 2.1.2.

2.1.1 Logický model

Motivácia. *Logický (dátový) model* popisuje "veci" *reálneho sveta* (t.j., *problémová doména (problem domain)*), ktoré napr.organizácia ako Walmart, chce zhromažďovať a spracovávať dáta. Logické modely sú graficky popísané pomocou diagramov, v praxi často pomocou takzvaných *Entitno-Relačných (ER)-diagramov*, bez zohľadnenia špečifík určitého DBS. Výhod tohto spôsobu grafickej reprezentácie je mnoho, napr. zlepšenie zrozumiteľnosti a uľahčenie komunikácie v rámci vývojového tímu a so zákazníkom.

Na modelovanie ER-diagramov je k dispozícii množstvo rôznych grafických formalizmov a nástrojov. My sa zameriame len na *ORACLE SQL Developer Data Modeler (ODM)*. Obrázok 2.2 zobrazuje ER-diagram nášho príkladu o Ľudských Zdrojoch, ktorý bude podrobnejšie vysvetlený nižšie.

[1] Všimnite si, že na realizáciu fyzického modelu možno použiť nielen relačný model, na ktorý sa zameriava táto Kapitola, ale sú dostupné aj iné formáty reprezentácie údajov (známe aj ako "dátové modely"), ako napr. *objektovo-orientovaný dátový model* alebo *NoSQL dátové modely*.



Obr. 2.2: Logický Model nášho príkladu o Ľudských Zdrojoch reprezentovaný ER-diagramom.

Rovnaký príklad je zobrazený na Obrázkoch 2.3, 2.4 a 2.5, tentoraz však v rámci ODM spoločnosti ORACLE. Všetky pojmy sú popísané tak, aby sa vysvetlili najdôležitejšie grafické označenia a symboly.

ER-Diagramy – Entity, Atribúty, Vzťahy. *ER-diagramy* popisujú koncept záujmu a pozostávajú z (i) *entít* (napr. Departments (Oddelenia), Employees (Zamestnanci), Jobs (Pracovné miesta)) (ii) *atribútov* (napr., "departmentName" - názov oddelenia a "locationName" - názov miesta oddelenia alebo "lastName" - priezvisko a "salary" - plat zamestnancov) ich charakteristiky, a (iii) *vzťahov* (napr., oddelenie "zahŕňa" niekoľko zamestnancov) medzi entitami .

Entity. *Entita* má pridelený názov v podobe podstatného mena a je graficky reprezentovaná ako zaoblený obdĺžnik. Používajú sa názvy v množnom alebo v jednotnom čísle, avšak preferovaný je tvar v jednotnom čísle (a tiež je to štandard podľa ISO 11179-5) .

Atribúty. *Atribúty* sú vždy asociované s entitami, to znamená, že nemôžu existovať samostatne. Na rozdiel od entít, *atribút* je "*atomický*", čo znamená, že nemôže byť ďalej štrukturovaný ako entita. Je reprezentovaný podstatným menom v jednotnom čísle. Každému atribútu možno prideliť *doménu* (*dátový typ*), ktorá označuje "typ" povolených hodnôt. Najbežnejšie domény sú numerické (`NUMERIC`), znakové (`VARCHAR`) a dátumové/časové (`DATE/TIME`).

Okrem charakteristiky entity môžu určité atribúty slúžiť na *jedinečnú identifikáciu každej inštancie* entity (napr. konkrétneho zamestnanca alebo konkrétneho oddelenia). Takýto atribút pre identifikáciu entity sa nazýva "*Primárny Kľúč (PK)*". Pre oddelenie, možno definovať "umelé" ID ako PK (ktoré by malo mať číselnu hodnotu, napr. "departmentID") alebo kombináciu existujúcich atribútov ako "departmentName" a "location", ktoré tvoria takzvaný "*kompozitný PK*". Pre každú entitu je potrebné definovať PK, pretože v opačnom prípade nemožno pri získavaní údajov rozlíšiť napríklad zamestnancov alebo oddelenia s rovnakým názvom. Každá hodnota atribútu PK preto musí byť jedinečná a samozrejme musí existovať.

Taktiež je potrebné spomenúť, že okrem domén a PK je možné špecifikovať pre atribúty aj iné *obmedzenia* hodnôt, napr. `NOT NULL`, teda "povinné" definovanie hodnôt atribútu (napr. pre atribút salary).

Vzťah. *Vzťah* spája (najčastejšie) dve rôzne entity. Je vizuálne znázornený čiarou medzi entitami a pomenovaný slovesom nad čiarou. Tri rôzne druhy vzťahov sa bežne používajú v praxi, pričom sa líšia *počtom inštancií entít*, ktoré môžu byť súčasťou vzťahu. "*Kardinalita vzťahu*" (predstavuje iný druh obmedzenia údajov, ktoré je možné špecifikovať):

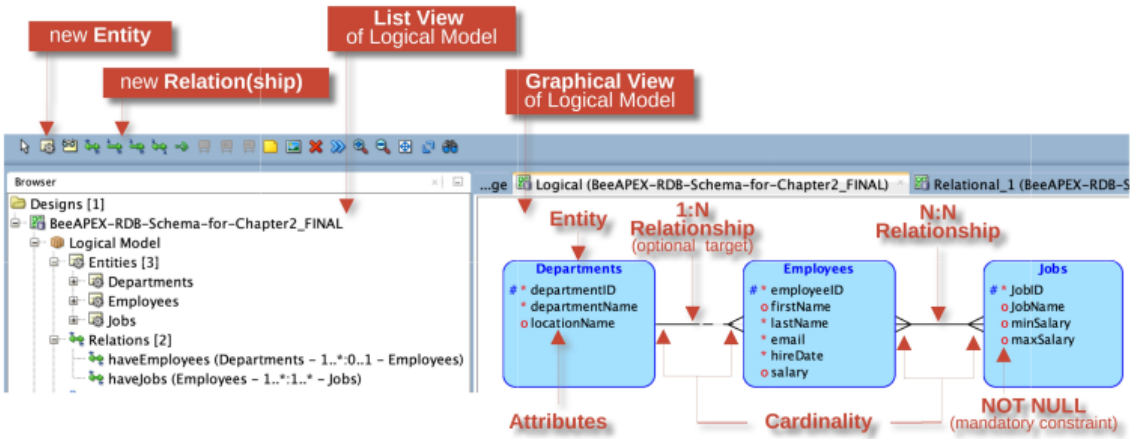
- **One-to-Many ("1:N"):** Jedna inštancia entity môže byť vo vzťahu s mnohými inštanciami inej entity, (napr. jedno oddelenie má mnoho zamestnancov, to je vzťah s názvom "haveEmployees") a entita na strane vzťahu "many" (zamestnanec) môže byť prepojená len s jednou inštanciou entity na strane vzťahu "one" (oddelenie). Takýto vzťah sa nazýva "*Source-Target*" alebo "*Parent-Child*" *vzťah*. Zdrojová (Source) entita na strane "one" vystupuje ako "Parent" (Rodič) a ľubovoľný počet cieľových (target) entít na strane "many" vystupujú ako "Childs" (Deti). Vo vyššie uvedenom príklade reprezentuje entita oddelenie nadradenú entitu a entita zamestnanci reprezentuje podriadenú entitu.
- **Many-to-Many ("M:N"):** Jedna inštancia zdrojovej entity môže byť vo vzťahu s *mnohými inštanciami cieľovej entity* a jedna inštancia cieľovej entity môže byť vo vzťahu s *mnohými*

inštanciami zdrojovej entity. Príkladom by mohli byť Zamestnanci, ktorí majú rôzne pracovné pozície (t.j. vzťah "haveJobs"), keďže zamestnanec môže mať v priebehu času viac ako jednu pracovnú pozíciu a určitú pracovnú pozíciu pravdepodobne zastáva viac zamestnancov. Je potrebné poznamenať, že vzťah "M:N" vždy vytvorí dva vzťahy "1:N", ako bude vidieť pri modelovaní databázy.

- **One-to-One ("1:1"):** Jedna inštancia zdrojovej entity môže byť vo vzťahu s *jednou inštanciou cieľovej entity* a naopak jedna inštancia cieľovej entity môže byť vo vzťahu s jednou inštanciou zdrojovej entity (napr. osoba "livesAt" (býva na) konkrétnej adrese a táto adresa je adresou len tejto konkrétnej osoby). Treba však poznamenať, že tento druh vzťahu je často uprednostňovaný kompaktnejším zastúpením iba v rámci jednej entity (napr. entita osoba zahŕňa atribúty adresy ako "street" a "city"), a preto sa o ňom už ďalej v tejto Kapitole neuvažuje.
- **Voliteľné:** Pre každý z týchto troch rôznych druhov vzťahov je tiež možné špecifikovať na každom konci zdroja a cieľa, či je *existencia aspoň 1 inštancie entity voliteľná alebo nie*. Napríklad, ak sa založí nové oddelenie, môže sa stať, že pre toto oddelenie ešte nie sú prijatí žiadni zamestnanci, t. j. cieľ vzťahu „haveEmployees" musí byť definovaný ako voliteľný. To je znázornené bodkovanou čiarou v ODM.

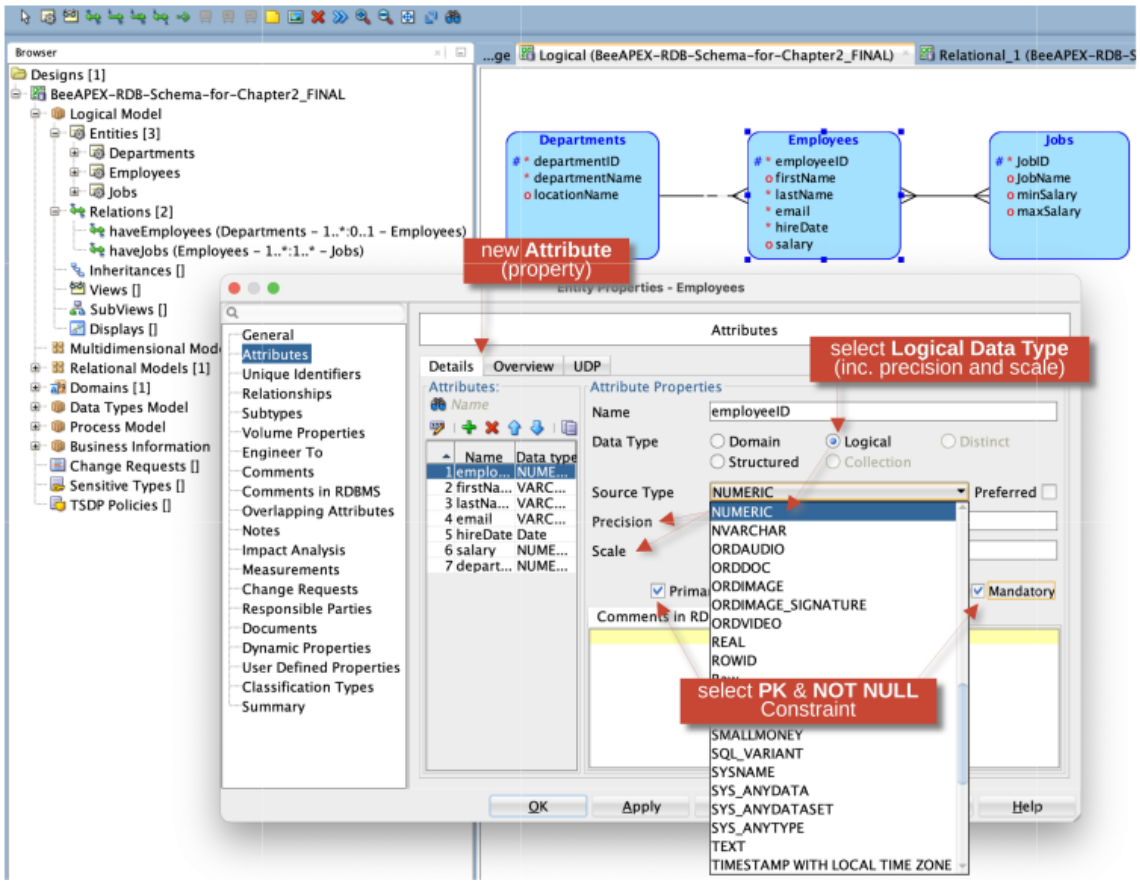
Vzťahy sa v ODM nazývajú jednoducho ako "vzťahy", čo by sa nemalo zamieňať so "czťahmi relačnéhol modelu", ktoré sú v skutočnosti synonymom pre "tabuľky" (pozri Časť 2.1.2). Preto sa ďalej pridržiavame bežnejšieho pojmu "vzťah".

GUI-Elementy ODM pre Správu ER-Diagramov. Nasledujúce Obrázky poskytujú informácie o najdôležitejších prvkoch GUI ODM. Prvý z obrázkov, Obrázok 2.3 ilustruje, ako definovať nové entity a nové vzťahy, ako aj vizualizáciu logického modelu vo forme zoznamu a grafického zobrazenia. V grafickom zobrazení sú označené najdôležitejšie grafické zápisy a symboly.



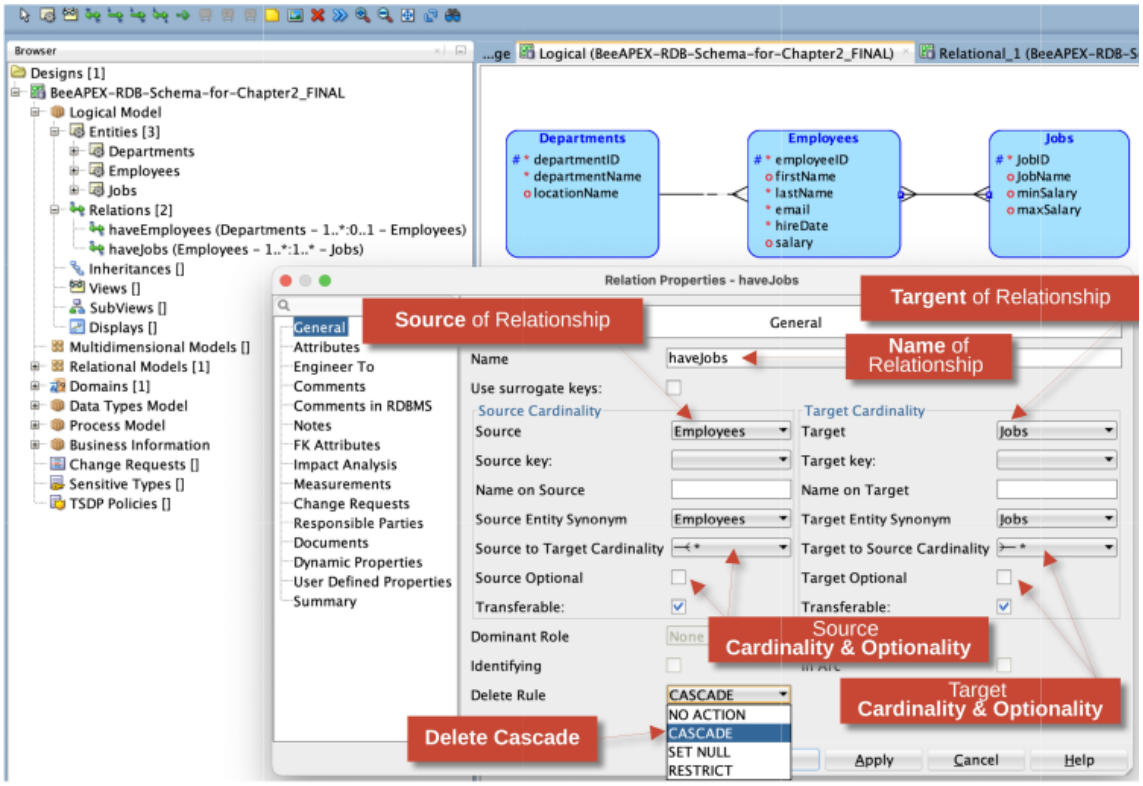
Obr. 2.3: Správa logického modelu pomocou ODM.

Obrázok 2.4 zobrazuje dialógové okno pre definíciu atribútov, ktoré je možné otvoriť jednoduchým dvojklikom na určitú entitu.



Obr. 2.4: Správa atribútov pomocou ODM.

Obrázok 2.5 zobrazuje dialógové okno na definovanie vzťahov, ktoré možno spustiť jednoduchým dvojklikom na čiaru predstavujúcu vzťah.

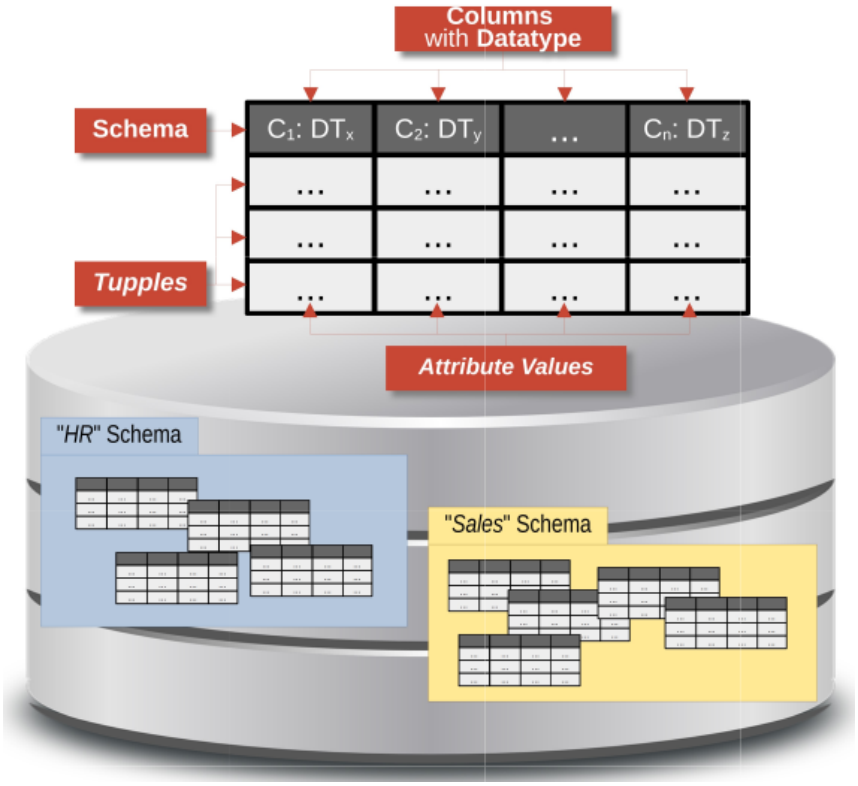


Obr. 2.5: Správa vzťahov pomocou ODM.

2.1.2 Relačný model

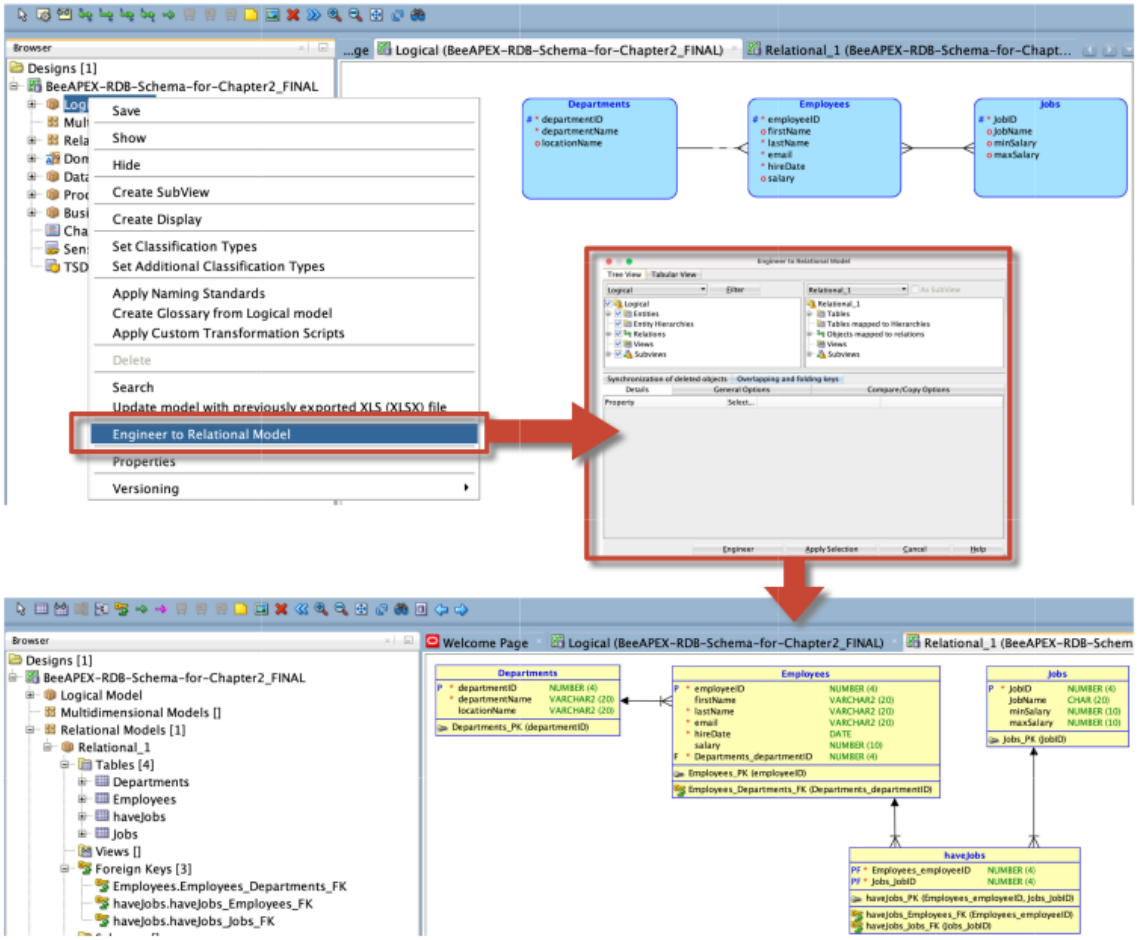
DB Schéma. Logický model v zmysle ER-diagramu, ktorý je nezávislý od konkrétneho DBS, je základom na odvodenie štruktúry DB špecifickej pre DBS nazývanej "*DB-schéma*". V rámci špecifického DBS je možné vytvárať a uchovávať ľubovoľný počet rôznych schém pre rôznych používateľov a/alebo domén, napr. jednu na správu HR údajov a druhú schému na správu údajov o predaji produktov. Schéma funguje podobne ako priečinok v súborovom systéme, ktorý umožňuje zoskupovať údaje na konkrétny účel.

Schéma relačnej DB. Špecifickým druhom schémy pre DB je takzvaná *Schéma Relačnej DB (RDB)* ktorá používa *Relačný Model (RM)* ako základný formalizmus na popis štruktúry údajov. RM je pomerne jednoduchý, pretože predstavuje schému DB ako kolekciu *vzťahov* tiež známu ako *tabuľky*, ktoré pripomínajú koncept entít. Tabuľky sú odvodené od entít jednoduchým spôsobom. Podobne ako entita, tabuľka pozostáva z názvu, *stĺpcov* (pripomínajúcich *atribúty*), ich dátových typov a ďalších (voliteľných) obmedzení. Všetky tabuľky tvoria *RDB-schému* (pozri Obrázok 2.6). Schéma tabuľky je taktiež "nástroj" na ukladanie aktuálnych údajov, zatiaľ čo každý riadok v tabuľke sa nazýva "*tuple*" ktorý pozostáva z kolekcie súvisiacich údajových hodnôt uložených v stĺpcoch, čím reprezentujú reálny svet *inštancie entity* v zmysle *relačných* (*tabuľkových*) údajov.



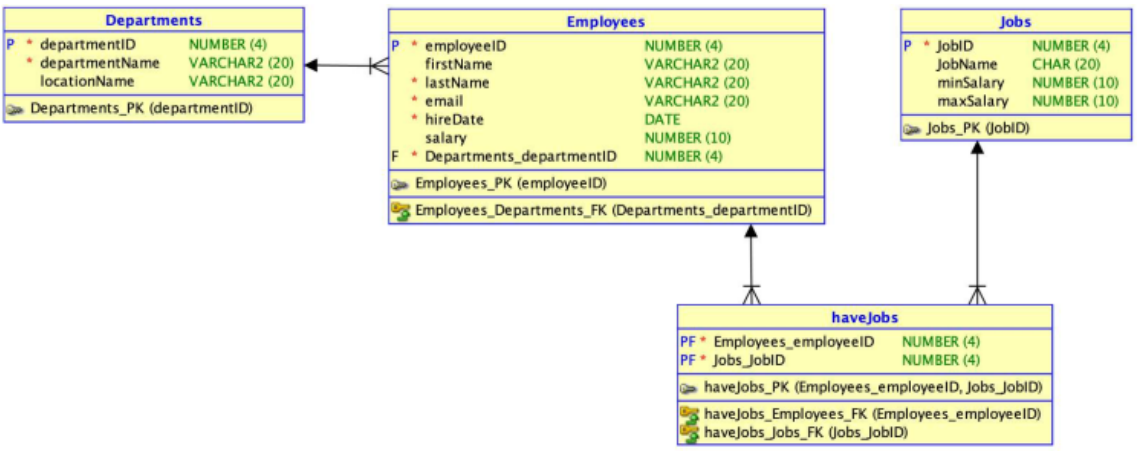
Obr. 2.6: RDB-schémy a tabuľkové schémy.

Je zaujímavé poznamenať, že *ODM* umožňuje automaticky generovať RDB-Schemy z logického modelu v podobe tabuliek, čím zobrazuje entity nášho logického modelu a všetky atribúty a obmedzenia každej entity. Je to možné vykonať pomocou položky v ponuke menu *"Engineer to Relational Model"*, ako je znázornené na Obrázku 2.7. Výsledná RDB-schéma je tiež graficky vizualizovaná pomocou ER-diagramu, založeného na trochu inom, viac "tabuľkovo-orientovanom zápise" ako ER-diagram logického modelu, ktorý teraz zobrazuje napr. typy všetkých atribútov.



Obr. 2.7: Proces generovania RDB-schémy z logického Modelu.

Teraz sa zamerajme na náš relačný model znázornený na Obrázku 2.8.

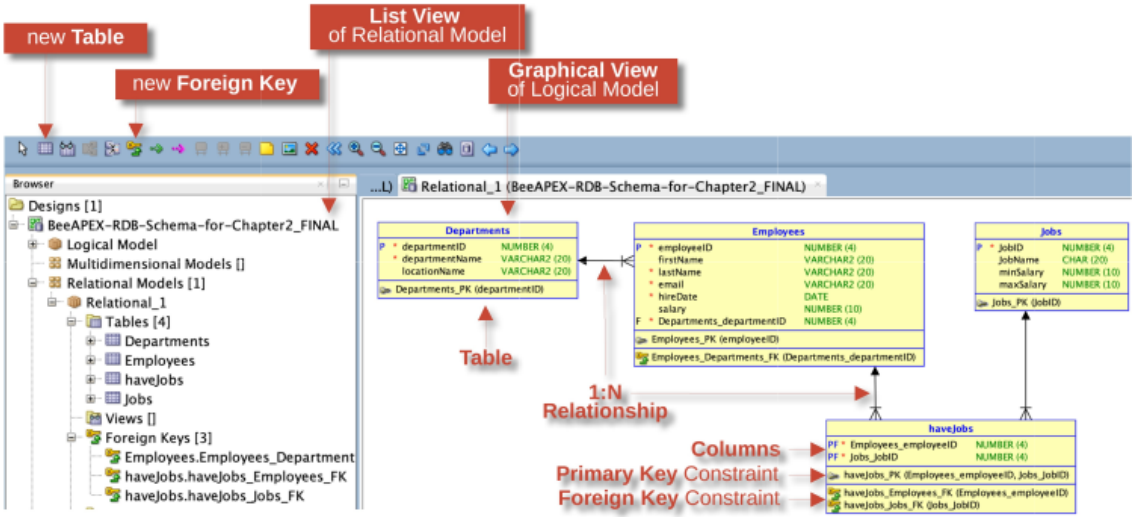


Obr. 2.8: Automaticky vygenerovaná RDB-schéma.

Počas procesu generovania relačného modelu z logického modelu sa okrem transformácie existujúcich entít a atribútov do tabuliek a zodpovedajúcich stĺpcov automaticky pridávajú PK do každej tabuľky a priraďujú sa im názvy (napr. “Departments_PK (departmentID)”. Okrem toho sú vzťahy medzi entitami v rámci logického modelu automaticky reprezentované v tabuľkách pomocou takzvaného *konceptu* “*Foreign-Key (FK) - Cudzieho kľúča*” nasledujúcim spôsobom:

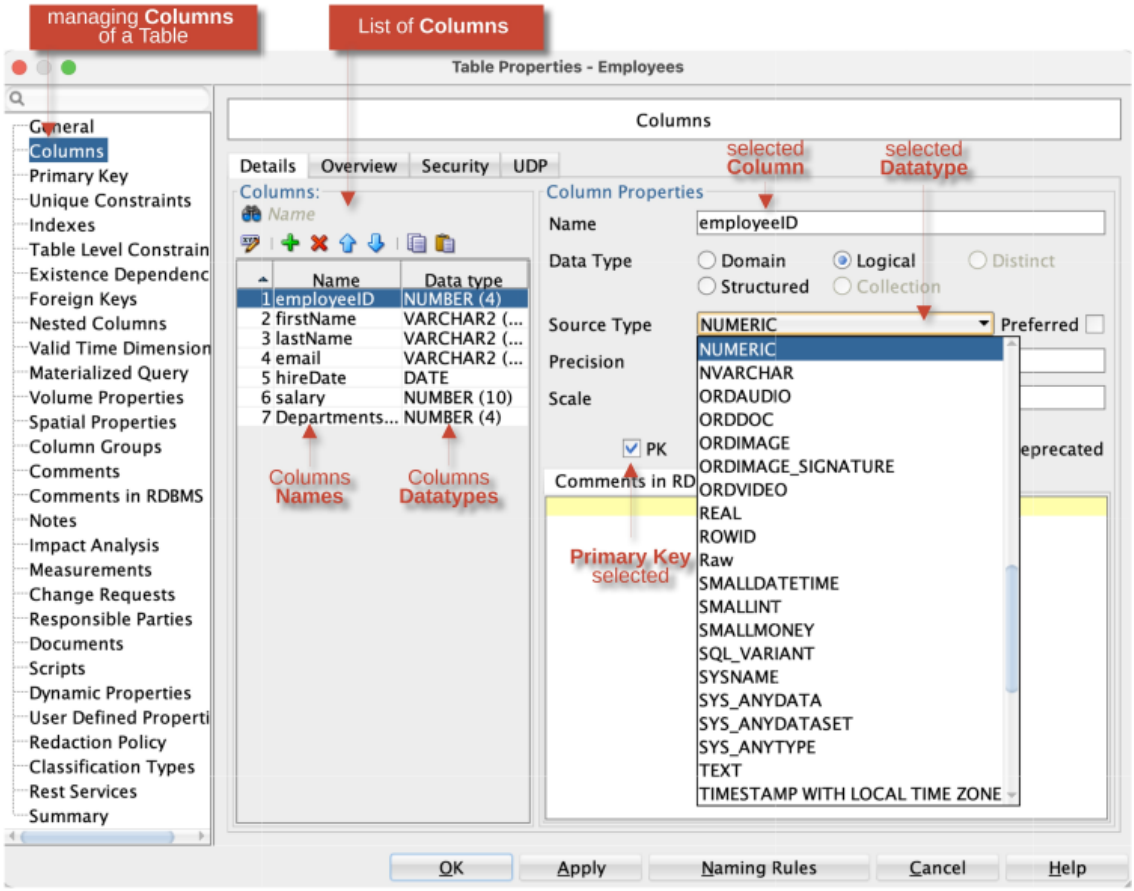
- **One-to-Many (1:N) Vzťah:** V tabuľke na strane vzťahu "Many" (t.j. Child tabuľka) je zavedený ďalší atribút a deklarovaný ako FK, ktorý slúži ako odkaz na hodnoty PK vzťahu na strane "One" (t.j. Parent tabuľka). Ak existuje napr. zamestnanec s PK ID 4711, potom v atribúte FK daného oddelenia musí byť prítomná aj hodnota 4711. V dôsledku toho sa vyžaduje, aby atribút FK mal rovnaký dátový typ ako PK. V našom príklade sa automaticky vygeneruje FK "Departments_departmentID" ako aj FK s názvom "Employees_Departments_FK". V dôsledku toho pri každej manipulácii s údajmi v rámci týchto dvoch vzťahov v DBS, FK zaisťuje *konzistenciu* medzi týmito vzťahmi, t.j. správne údaje v rámci vzťahu. Ak sa napríklad do tabuľky Employees vloží nový zamestnanec, DBS zabezpečí, aby bol priradený aj k existujúcemu oddeleniu. Ak sa pokúsite odstrániť oddelenie, DBS štandardne odmietne túto operáciu v prípade, že stále existujú pridružení zamestnanci. Zmena tohto predvoleného správania je možná pomocou takzvanej "*Cascade-Delete Option*", pri zadávaní FK, čo má za následok automatické vymazanie ("prepustenie") všetkých zamestnancov spojených s vymazaným oddelením.
- **Many-to-Many (M:N) Vzťah:** Vzťah many-to-many medzi dvoma tabuľkami sa realizuje zavedením ďalšej, tretej tabuľky, fungujúcej ako nejaký druh "*mapovacej tabuľky*", v ktorej sú uložené odkazy na každý z PK pridružených tabuliek v zmysle dvoch atribútov FK. Čo sa týka nášho spustiteľného príkladu Employees a Jobs, automaticky sa vygeneruje tretí vzťah s názvom "haveJobs", ktorý je teraz zodpovedný za riadenie správnych asociácií medzi zamestnancami a ich príslušnými pracovnými pozíciami. PK tejto mapovacej tabuľky je kompozitný PK, ktorý pozostáva s PK oboch tabuliek t.j. "haveJobs_PK(Employees_employeeID, Jobs_jobID)". Každý atribút tohto PK sa používa aj na definovanie príslušného FK, pričom každý odkazuje na PK priradenej tabuľky (t.j. "haveJobs_Employees_FK (Employees_employeeID)" a "haveJobs_Jobs_FK(Jobs_JobID)"). Je potrebné poznamenať, že táto mapovacia tabuľka môže mať tiež ľubovoľný počet ďalších atribútov, ktoré ďalej charakterizujú vzťah, napr. "startDate" (dátum začiatku) a "endDate" (dátum konca) určitej pracovnej pozície zamestnanca.
- **Voliteľné:** Štandardne je pre každý z týchto vzťahov existencia aspoň jednej zdrojovej a jednej cieľovej inštancie voliteľná. Ak by to bolo povinné, potom s príslušným atribútom FK musí byť spojené ďalšie obmedzenie NOT NULL. Ak vezmeme do úvahy náš príklad, všetky tri FK potrebujú obmedzenie NOT NULL, pretože (i) Employees - Zamestnanci musia byť pridružení k určitému (ii) Department - oddeleniu ako aj k aspoň jednému (iii) Job - pracovnému miestu a pracovné miesto musí byť priradené aspoň k jednému zamestnancovi.

GUI-elementy ODM pre správu RM. Nasledujúce Obrázky poskytujú prehľad najdôležitejších prvkov GUI *ODM* na správu RDB-Schémy, ktoré sú dosť podobné tým na správu logického modelu. Obrázok 2.9 ilustruje, ako definovať nové tabuľky a nové FK, ako aj vizualizáciu RM vo forme zoznamu a grafického zobrazenia. V grafickom zobrazení sú popísané najdôležitejšie grafické zápisy a symboly.



Obr. 2.9: GUI-elementy pre správu stĺpcov RM.

Obrázok 2.10 zobrazuje dialógové okno pre definíciu stĺpcov, ktorý možno spustiť jednoduchým dvojklikom na určitú tabuľku. Všimnite si, že toto dialógové okno sa do značnej miery podobá dialógovému oknu na správu atribútov, ako je už znázornené na Obrázku 2.4.



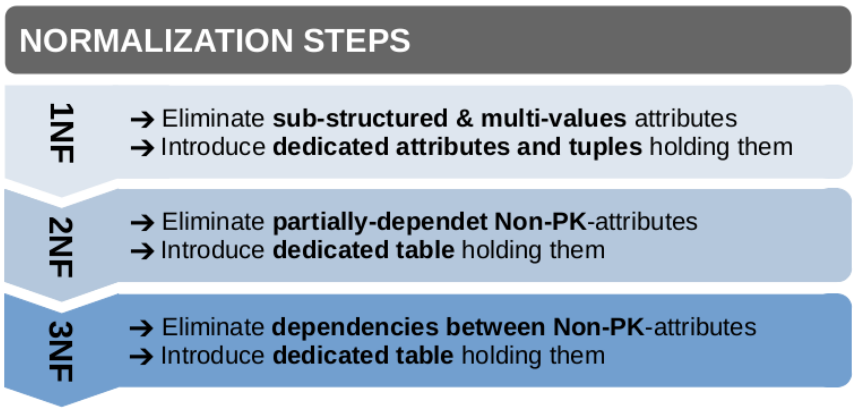
Obr. 2.10: GUI-elementy pre správu RM.

RDB-Schéma nezávislá od aspektov fyzického Úložiska. Hoci RDB-schéma založená na relačnom modeli je už konkrétnejšia a špecifickejšia pre DBS ako logický model (napr. “imple-

mentácia" vzťahov založených na koncepte FK) je potrebné poznamenať, že stále skrýva zložitosť základných mechanizmov ukladania DBS (známe ako "*fyzický*" *level*), napr. koľko súborov sa používa na uloženie tabuliek alebo na ktoré servery sa majú dáta ukladať. Ďalšou veľkou výhodou tejto abstrakcie aspektov fyzického ukladania je, že nové RDB-schémy možno realizovať alebo meniť existujúce bez toho, aby ste sa museli starať o základnú organizáciu fyzických údajov.

2.1.3 Normalizácia RDB-schémy

RDB-Schéma, ktorá bola odvodená z logickej schémy, môže byť často vylepšená tak, aby sa dosiahli určité *kritéria kvality*, ktoré prevládajú v DBS, pričom sú na jednej strane najvhodnejšie pre *prístup na čítanie údajov* z hľadiska DB-dotazov, ale na druhej strane poskytujú vhodný základ pre *manipuláciu s údajmi* z hľadiska vkladania, aktualizácie a vymazávania údajov. Hoci ide do určitej miery o protichodné ciele, pre RDBS je k dispozícii dobre známy koncept takzvanej "*teórie normalizácie*" , ktorý popisuje systematický proces troch krokov "*prvá normálna forma*", "*druhá normálna forma*" a "*tretia normálna forma*", čo vedie k vylepšeniu RDB-Schém (pozri Obrázok 2.11). V nasledujúcom texte sú tieto tri normálne formy popísané podrobnejšie s ukážkou na našom príklade.



Obr. 2.11: Kroky normalizácie – Prehľad.

Prvá Normálna Forma. Prvá normálna forma vyžaduje, aby bol každý atribút tabuľky "atomický" čo znamená, že nie sú povolené ani (i) *podštruktúry* (napr. ak namiesto nášho atribútu "locationName" použijete alternatívny atribút "location", ktorý by obsahoval hodnoty o ulici, poštovom smerovacom čísle, meste a krajine) ani (ii) *viaceré-hodnoty* (napr. atribút "location" môže obsahovať aj niekoľko adries určitého oddelenia). Podštruktúry v rámci hodnôt môžu byť *eliminované* jednoduchou reprezentáciou podštruktúry s jednotlivými atribútmi (napr. pre každú časť adresy) a viacerými hodnotami s použitím vlastnej n-tice pre každú z hodnôt (t.j. umiestnením každej adresy do samostatného riadku).

Jednou z hlavných *výhod* prvej normálnej formy je, že dotazy na dáta môžu byť špecifickejšie, pretože teraz je možné napríklad pristupovať ku každej časti adresy, ako aj ku každej n-tici adries samostatne.

Druhá Normálna Forma. Druhá normálna forma vyžaduje, (i) aby bola tabuľka v *prvej normálnej forme* a (ii) v prípade kompozitného PK, t. j. pozostávajúceho z dvoch alebo viacerých atribútov, sú všetky ostatné hodnoty atribútov mimo PK *jedinečne identifikované iba celým kompozitným PK*. Predpokladajme napríklad, že tabuľka Departments obsahuje kompozitný PK s atribútmi "departmentName" a "locationName". Tento PK správne určuje hodnotu atribútu "annualRevenue", keďže predpokladáme, že hodnota "annualRevenue" závisí od oddelenie a lokality. Hodnoty atribútu "locationAddress" by však už záviseli od časti PK, konkrétne od názvu lokality. Aby sme *eliminovali* túto *čiastočnú závislosť*, musíme *rozdeliť* tabuľku Departments, odčleniť

informácie o umiestnení do samostatnej tabuľky "Locations" a spojiť obe tabuľky cez FK.

Hlavná *výhoda* druhej normálnej formy je, že zabraňuje *ukladaniu redundantných dát*, keďže v našom prípade je určitá adresa uložená iba raz a môže byť opätovne použitá rôznymi oddeleniami. Tým sa zmenšuje nielen úložný priestor, ale predovšetkým sa predchádza aj chybným údajom, takzvaným "*anomáliam*", ak sa napríklad pri manipulácii s údajmi nezapracujú zmeny aj na duplicitných údajoch (napr. v prípade presunu viacerých oddelení na inú adresu, by adresa bola zmenená len raz). Je dobré vedieť, že ak máme tabuľky iba s nekompozitnými PK (napr. číselné ID pre každú tabuľku), RDB-Schéma "automaticky" spĺňa druhú normálnu formu, pretože čiastočné závislosti nemôžu existovať.

Tretia Normálna Forma. Tretia normálna forma vyžaduje, aby (i) tabuľka bola v *druhej normálnej forme* a (ii) aby každý *nekľúčový atribút bol jednoznačne identifikovaný primárnym kľúčom* a nie akýmkoľvek iným nekľúčovým atribútom, čo vedie k *tranzitívnym závislostiam*. Napríklad, ak by tabuľka Departments obsahovala okrem PK employeeID a locationName aj locationAddress, potom locationAdress by už bola jednoznačne identifikovaná pomocou locationName (ak predpokladáme, že každý locationName je asociovaný s inou adresou), čo nie je povolené tretou normálnou formou. Aby sa táto situácia eliminovala, informácie o lokalite sa musia vyčleniť do samostatnej tabuľky Locations a prepojiť cez príslušný FK s tabuľkou Departments.

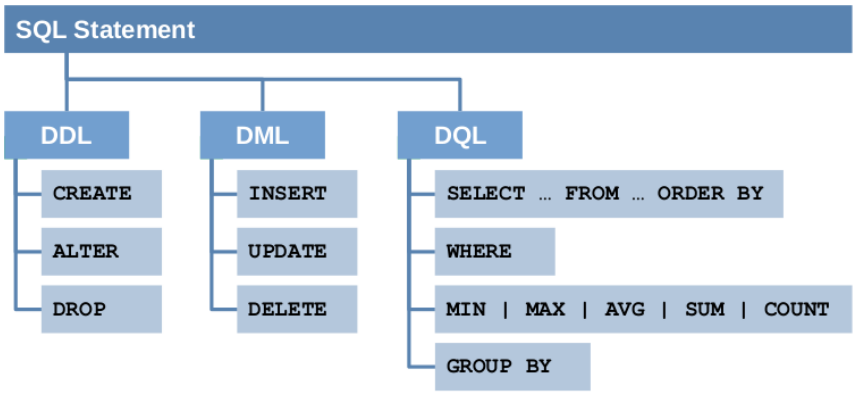
Výhoda je rovnaká, ako už bolo spomenuté pri druhej normálnej forme – zníženie redundancie a tým aj potenciálnych anomálií v prípade manipulácie s dátami.

2.2 Ako spravovať RDBS-schému – SQL-DDL

V predchádzajúcej Časti bola RDB-Schéma graficky modelovaná na rôznych úrovniach abstrakcie pre *návrh nízko-úrovňového vývoja*. Nasledujúca Časť hovorí o tom, ako môžu byť tieto grafické modely implementované "programovo" v rámci RDBS tak, aby sa dali použiť na správu skutočných údajov Webovej aplikácie. Otázky, ktoré budú zodpovedané v nasledujúcom texte, sú:

- Ako poradiť DBS, aby *vytvoril požadované tabuľky*, ideálne z grafických modelov a máme nejaké iné alternatívy, ak takéto grafické modely neexistujú?
- Ako vyzerá *DBS "Programovací Jazyk"*, ktorý je schopný vytvárať tabuľky?

SQL = DDL | DML | DQL. Aby sme najskôr odpovedali na druhú otázku, jazyk, ktorý sa na tento a aj na iné účely používa, je *štandardizovaný Structured Query Language (SQL) - štruktúrovaný dopytovací jazyk*. SQL poskytuje širokú škálu rôznych *"DBS programovacích príkazov"* ktoré môžu byť podľa ich účelu zoskupené do rôznych kategórií (pozri Obrázok 2.12), z ktorých najdôležitejšie sú (i) príkazy na správu RDB-Schémy (napr. vytváranie, zmena a mazanie tabuliek) nazývané ako *DDL (Data Definition Language)*, na ktoré sa zameriava táto Časť, (ii) príkazy na manipuláciu s údajmi v rámci schémy (napr. vkladanie, aktualizácia a mazanie údajov v tabuľkách) nazývané *DML (Data Manipulation Language)*, ktoré budú bližšie popísané v Časti 2.3 a (iii) príkazy na dopytovanie údajov s názvom *DQL (Data Query Language)*, ktoré sú ďalej spracované v Časti 2.4.



Obr. 2.12: Kategórie SQL príkazov.

Päť možností na vytváranie tabuliek. Podľa nástrojov poskytovaných ORACLE existuje päť rôznych možností na vytváranie tabuliek, t. j. implementácia RDB-Schémy (pozri tiež Obrázok 2.45):

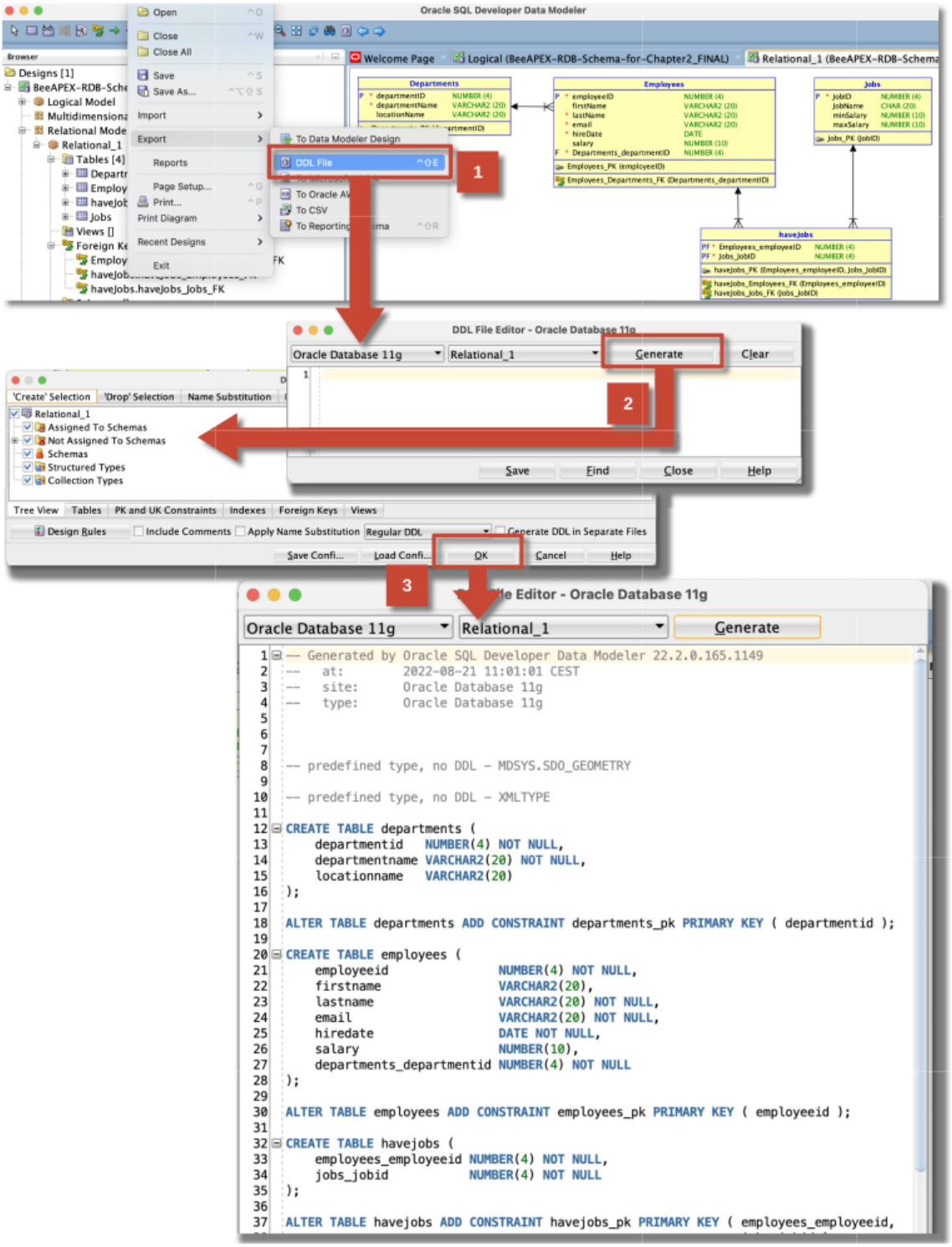
1. **Automatické generovanie tabuliek pomocou ODM.** Toto je jeden z najjednoduchších spôsobov generovania tabuliek, pretože by stačilo jednoduché spustenie *automatického generovania tabuliek* na niekoľko klikov *pomocou ODM* (pozri Časť 2.2.1).
2. **Automatické generovanie tabuliek pomocou Quick SQL.** Ďalšou alternatívou by bolo použitie nejakého druhu "skráteného zápisu" pre SQL, ktorý poskytuje nástroj ORACLES Quick SQL (pozri Časť 2.2.2). Toto je ideálna voľba, ak nie sú k dispozícii žiadne grafické (logické a / alebo relačné) modely RDB-Schémy alebo ak je potrebné rýchle testovanie.
3. **Manuálne vytváranie tabuliek pomocou SQL-DDL.** Samozrejme je tu aj možnosť programovo špecifikovať tabuľky pomocou obyčajného SQL, čo poskytuje výhody, že každý detail týkajúci sa špecifikácie tabuľky je možné nakonfigurovať podľa potreby a nie je závislý na niekedy suboptimálnych procesoch automatického generovania (pozri Časť 2.2.3).
4. **Manuálne vytváranie tabuliek pomocou ORACLE Object Browser (OB).** Pre tých používateľov, ktorí nemajú žiadne znalosti SQL, Object Browser umožňuje vytvárať nové tabuľky a v prípade potreby ich meniť spôsobom založeným na formulári (pozri Časť 2.2.4).
5. **Automatické vytváranie tabuliek pomocou ORACLE Data Workshop.** Existuje ďalšia možnosť automatického vytvárania tabuliek, ktorú je však možné použiť iba vtedy, ak sú už nejaké údaje uložené v externých súboroch (napr. v súboroch EXCEL). V takom prípade je možné použiť automatické vytváranie tabuliek pomocou ORACLE Data Workshop. Táto možnosť je podrobnejšie opísaná v Kapitole 4.

V nasledujúcom texte sa podrobnejšie rozoberú prvé štyri možnosti.

2.2.1 Automatické generovanie tabuliek pomocou ODM

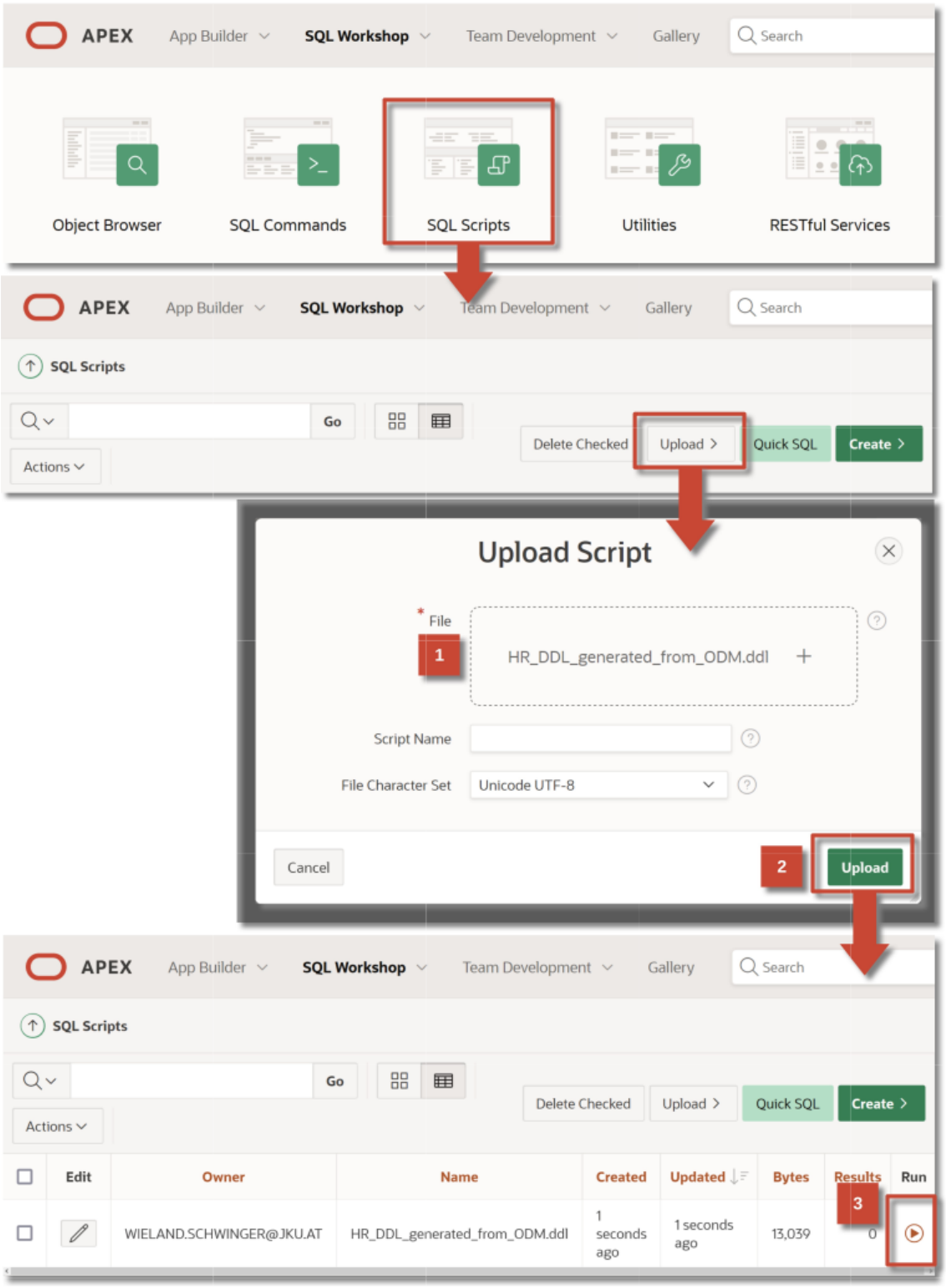
ODM je skratkou pre nástroj nazývaný Oracle SQL Developer Data Modeler. Keďže ide o možnosť generovania tabuliek s najmenšou námahou, na vytvorenie požadovaných DB tabuliek sú potrebné len dva kroky v prípade, že RDB-Schéma bola vytvorená v *ODM*.

Najskôr je potrebné vytvoriť export modelovanej RDB-Schémy do *SQL-skriptu* (pozri Obrázok 2.13, kroky 1 až 3), pričom ide v skutočnosti o jednoduchý textový súbor, ktorý obsahuje všetky príkazy SQL-DDL potrebné na vytvorenie tabuliek v rámci RDBS. Syntax vygenerovaných príkazov SQL je podrobnejšie opísaná v časti 2.2.3.



Obr. 2.13: Export RDB-schémy do DDL-skriptu.

Druhým krokom je vykonanie vygenerovaného SQL skriptu, vďaka ktorému sa vytvoria všetky tabuľky ako prázdne v úložnom priestore. Na to musí byť skript nahraný cez ORACLE SQL Workshop (pozri 1-2) a jednoducho spustený (pozri 3). Pozri Obrázok 2.14.



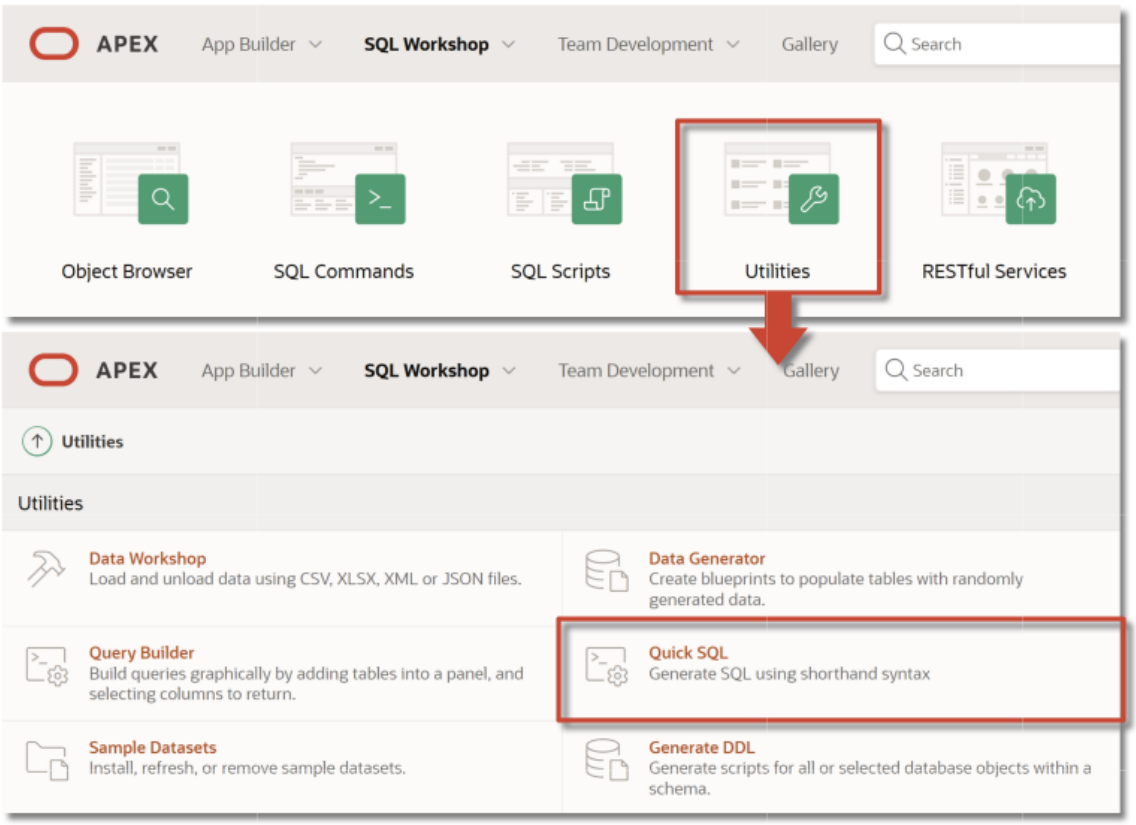
Obr. 2.14: Nahratie a Spustenie SQL-Skriptu.

2.2.2 Automatické generovanie tabuliek pomocou Quick SQL

Pri vytváraní DB-vrstvy webovej aplikácie je veľmi vhodné sledovať rôzne úrovne abstrakcie z hľadiska logických a relačných modelov. *Quick SQL* poskytuje jednoduchú a intuitívnu alternatívu na vytváranie tabuliek, celkom jednoduchšiu ako SQL, napr. na účely rýchleho testovania. Quick SQL ďalej umožňuje automaticky vygenerovať SQL skript. Akonáhle je SQL-skript vygenerovaný,

môže byť vylepšený a rozšírený a nakoniec spustený, aby sa vytvorili tabuľky. Znovu je potrebné zdôrazniť, že aj keď je Quick SQL navrhnutý tak, aby znížil úsilie potrebné na vytváranie tabuliek, rozhodne nie je navrhnutý ako náhrada za dátové modelovanie (pozri Časť 2.1).

ORACLE Quick SQL editor je prístupný cez možnosť "Utilities" v SQL Workshop (pozri Obrázok 2.15.

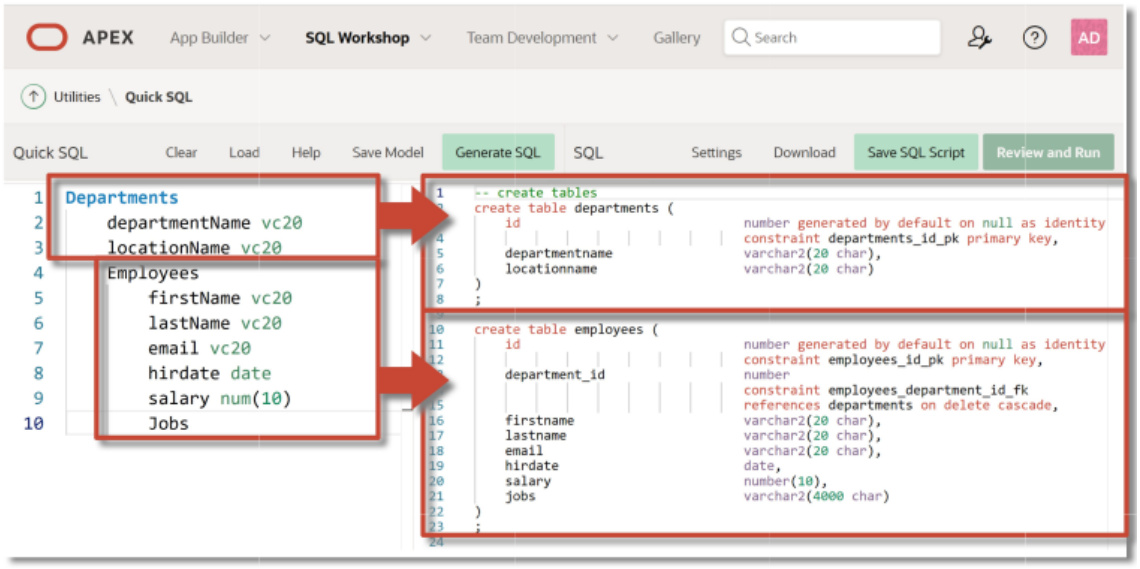


Obr. 2.15: Prístup ku Quick SQL cez SQL Workshop

Teraz si prejdeme syntax poskytovanú Quick SQL, ktorá v podstate používa niekoľko princípov formátovania na špecifikovanie požadovaného generovania tabuliek:

- **Parent tabuľky:** Názvy parent tabuliek sa zadávajú bez akéhokoľvek odsadenia. Názvy tabuliek sú automaticky naformátované a všetky medzery sa nahradia podčiarkovníkmi.
- **Atribúty:** Názvy atribútov sa zadávajú s jednotným odsadením dvoch alebo viacerých medzier. Rovnako ako v prípade názvov tabuliek sú názvy atribútov automaticky naformátované a všetky medzery sa nahradia podčiarkovníkmi.
- **Parent/child vzťahy:** Parent/child vzťahy sa zadávajú odsadením child tabuliek pod parent tabuľkami. Child tabuľky by mali mať rovnakú úroveň ako stĺpce v parent tabuľke.
- **Dátové Typy:** Na základe anglického textu obsiahnutého v názve stĺpca a pri absencii akéhokoľvek špecifikovaného dátového typu, RDBS automaticky odvodí vhodné dátové typy. Ak má mať stĺpec určený dátový typ, sú možné tieto: `NUM`, `INT`, `VC` (pre `VARCHAR2`) alebo `DATE`). Dátový typ môže byť jednoducho špecifikovaný na konci názvu stĺpca oddelený medzerou. Ak má byť definovaná dĺžka reťazca `VC` musí sa zadať `VCn`, kde `n` je dĺžka `VARCHAR2`. Dostupná syntax sa zobrazuje prostredníctvom ponuky "Help".

Obrázok 2.16 ukazuje príklad Quick SQL zápisu, ktorý zobrazuje našu Parent tabuľku "Departments" a príslušnú Child tabuľku "Employees" na ľavej strane a automaticky vygenerovaný skript SQL v paneli SQL Output. SQL-skript sa generuje okamžite po každom vykonaní. Finálny SQL-skript je možné uložiť pre ďalšie použitie, ako aj spustiť, aby sa vytvorili prázdne tabuľky.

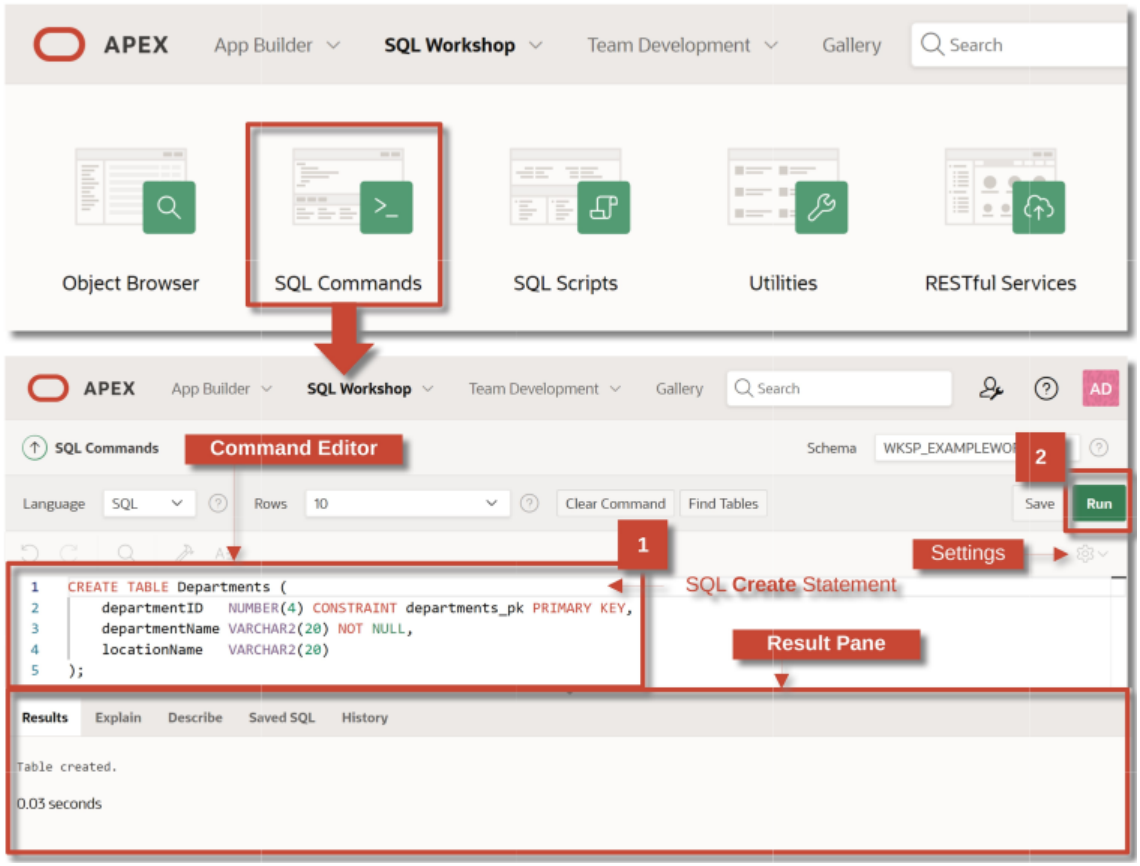


Obr. 2.16: Vygenerovaný SQL-Script pomocou Quick SQL.

Quick SQL má tiež možnosť generovať integritné obmedzenia (cudzie kľúče a povolené hodnoty) pre špecifické dátové pole, ako aj na generovanie testovacích údajov.

2.2.3 Manuálne vytváranie tabuliek pomocou SQL-DDL

Cieľom programového spôsobu vytvárania tabuliek je vyhnúť sa akémukoľvek návrhu "nízko-úrovňového" automatického generovania SQL-skriptov (ako bolo popísané v Časti 2.2.1 a Časti 2.2.2) a pripraviť SQL-skript manuálne. Aby sme boli konkrétnejší, je možné špecifikovať SQL-skripty aj manuálne pomocou príkazu SQL-DDL `CREATE TABLE` v rámci SQL-Command-Editoru poskytovaného spoločnosťou ORACLE *SQL-Workshop* (pozri Obrázok 2.17). Po manuálnom definovaní príkazov SQL pomocou SQL-DDL (1) možno výsledný SQL-skript uložiť a spustiť (2), aby sa vytvorili prázdne tabuľky (pozri Obrázok 2.17).



Obr. 2.17: SQL Command Editor.

Príkazy SQL sú viac-menej jednoduché, ale so špecifickou syntaxou. Príkaz `CREATE TABLE` pozostáva z *kľúčových slov špecifických pre SQL* ako "`CREATE`", ako aj z *definície obmedzení* (dátové typy, primárne kľúče a cudzie kľúče). Na Obrázku 2.18 sú všetky kľúčové slová, ako aj ostatné syntaktické symboly špecifické pre SQL (zátvorky a čiarky) napísané veľkými písmenami a zafarbené modrou farbou. Je dôležité poznamenať, že kľúčové slová SQL nerozlišujú veľké a malé písmená, čo znamená, že napr. "`CREATE`" je to isté ako "`create`". Definícia atribútu sa nachádza v zátvorkách, atribúty sú oddelené čiarkami, každý je ideálne definovaný v samostatnom riadku (hoci ide len o neformálne konvencie formátovania, nie sú súčasťou syntaxe SQL). Obmedzenia možno definovať pomocou explicitného názvu (napr. departments_pk), ktorý nie je povinný (napr. `NOT NULL` pre departmentName), ale slúži len na účely lepšej čitateľnosti. Každý príkaz SQL je na konci oddelený bodkočiarkou.

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

Obr. 2.18: SQL príkaz pre vytvorenie tabuľky `Departments`.

Doplnkom príkazu `CREATE TABLE` je SQL-príkaz na vymazanie tabuľky – funguje rovnako, bez ohľadu na to, či je tabuľka prázdna alebo už obsahuje niektoré údaje, ktoré sa potom tiež vymažú (pozri Obrázok 2.19).

```
DROP TABLE Departments;
```

Obr. 2.19: SQL príkaz na vymazanie tabuľky `Departments`.

A nakoniec, definícia už existujúcich tabuliek môže byť tiež zmenená, napr. pridaním ďalších atribútov alebo zmenou dátových typov (pozri Obrázok 2.20):

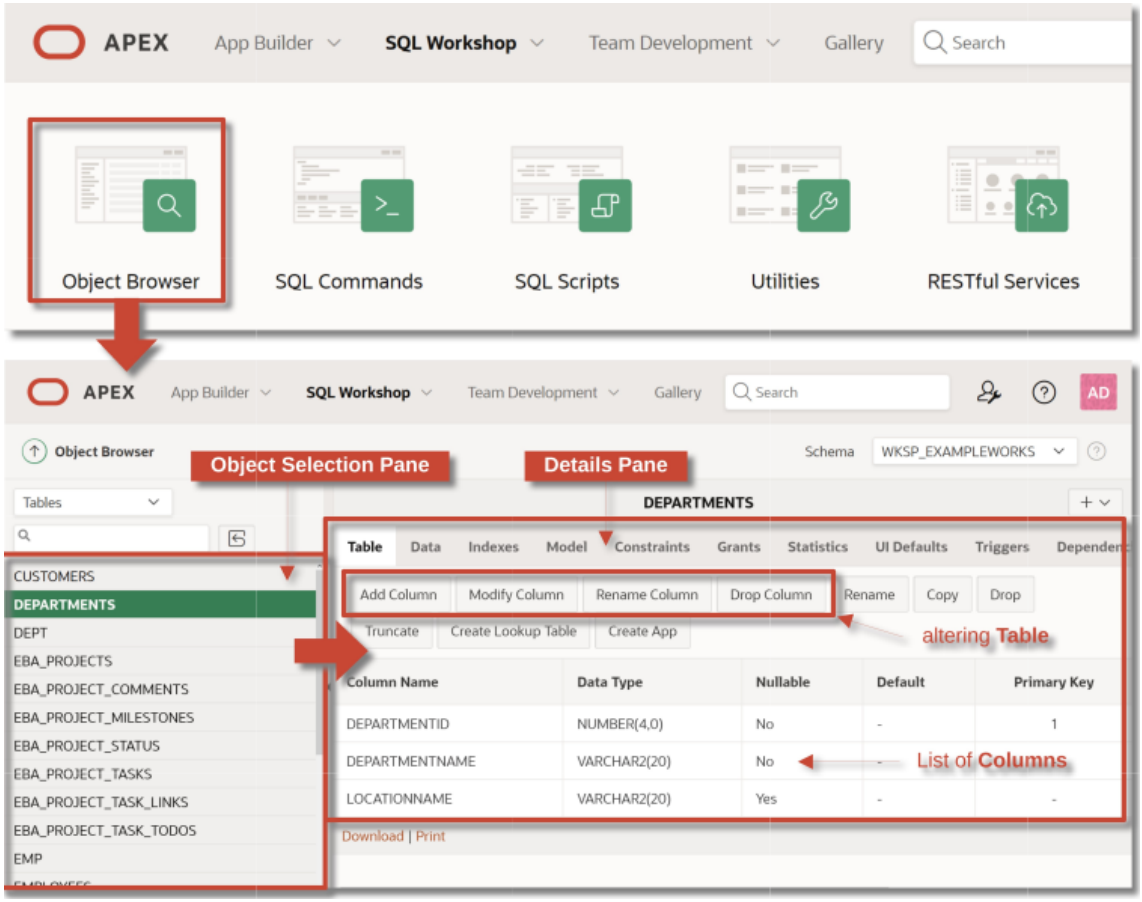
```
ALTER TABLE Departments
      ADD annualRevenue NUMBER(10);
```

Obr. 2.20: SQL príkaz na zmenu tabuľky `Departments`

Ak tabuľka, ktorá by sa mala zmeniť, už obsahuje údaje, potom sú úpravy trochu obmedzené, pretože napríklad zmena typu údajov môže viesť k ďalším problémom. Touto otázkou sa však táto Časť ďalej nezaoberá.

2.2.4 Manuálna správa tabuliek pomocou ORACLE Object Browser

Object Browser od ORACLE's SQL Workshop umožňuje nielen prezerať všetky vygenerované tabuľky, takzvané *"Objects"* v terminológii ORACLE, spolu s ich príslušnými atribútmi a obmedzeniami, ale aj ich meniť a vytvoriť nové, všetky založené na jednoduchom GUI-Wizard (pozri Obrázok 2.21).



Obr. 2.21: Správa tabuliek pomocou ORACLE Object Browser.

2.3 Ako manipulovať s dátami – SQL-DML

Prázdne tabuľky vytvorené v predchádzajúcej Časti teraz "čakajú"na naplnenie príslušnými údajmi, ktoré je možné ďalej upravovať a prípadne aj mazať. Presne za to zodpovedajú príkazy SQL-DML `INSERT`, `UPDATE` a `DELETE`, , ktoré budú stručne predstavené v nasledujúcej Časti 2.3.1 na našom ukážkovom príklade. Okrem toho je cez ORACLE Object Browser (“OB”) dostupná aj alternatíva bez kódu, ktorá umožňuje manuálne vkladať dáta do tabuliek, čo je vhodné pre používateľov bez akýchkoľvek znalostí SQL (pozri Časť 2.3.2). Nakoniec pomerne obmedzenú alternatívu vkladania údajov poskytuje Quick SQL (pozri Časť 2.3.3). Všetky tieto alternatívy sú tiež znázornené na Obrázku 2.45.

2.3.1 Manipulácia s údajmi pomocou SQL-DML

Na programovú manipuláciu s dátami pomocou SQL-DML sa používa SQL Command Editor (pozri Obrázok 2.17), ktorý sa použil na vytváranie tabuliek. V nasledujúcom texte sú uvedené niektoré príklady.

`INSERT` sa používa na *pridanie nových riadkov* do tabuľky, ako je možné vidieť v nasledujúcom príklade (pozri Obrázok 2.22):

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

Obr. 2.22: SQL príkaz na vkladanie nových údajov do tabuľky `Departments`.

Je potrebné poznamenať, že okrem tejto programovej možnosti sme v Časti 2.3.3 spomenuli aj možnosť použiť na vkladanie údajov Quick SQL, ktoré je však obmedzené len na vkladanie náhodných údajov.

`UPDATE` sa používa na aktualizáciu jedného alebo viacerých riadkov v tabuľke. Nasledujúci príklad je najjednoduchšou formou príkazu `UPDATE`, ktorý zvyšuje plat všetkých zamestnancov o 10% (pozri Obrázok 2.23):

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

Obr. 2.23: SQL príkaz na aktualizáciu platu všetkých zamestnancov o 10% v tabuľke `Employees`.

Mierne rozšírenie príkazu `UPDATE` je zobrazené v nasledujúcom príklade pridaním takzvanej “`WHERE`-klauzuly” , aby sa zvýšenie platu uplatnilo na zamestnanca, ktorý má priezvisko “Miller” (pozri Obrázok 2.24):

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

Obr. 2.24: SQL príkaz na aktualizáciu platu zamestnanca s priezviskom “Miller”.

Tiež je možné aktualizovať aj viac ako len jeden atribút tabuľky pomocou jedného príkazu, ako je možné vidieť v nasledujúcom príklade (pozri Obrázok 2.25):

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

Obr. 2.25: SQL príkaz na aktualizáciu viac ako jedného atribútu v tabuľke `Employees`.

`DELETE` sa používa na odstránenie jedného alebo viacerých riadkov údajov z tabuľky. Nasledujúci príklad je najjednoduchšou formou príkazu `DELETE`, ktorý vymaže všetky údaje uložené v tabuľke Employees (pozri Obrázok 2.26).

```
DELETE FROM Employees;
```

Obr. 2.26: SQL príkaz na vymazanie všetkých údajov z tabuľky `Employees`.

Rozšírenie predchádzajúceho príkazu o "`WHERE`-klauzulu" umožňuje vymazať zamestnanca s jobID "4711" (pozri Obrázok 2.27).

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

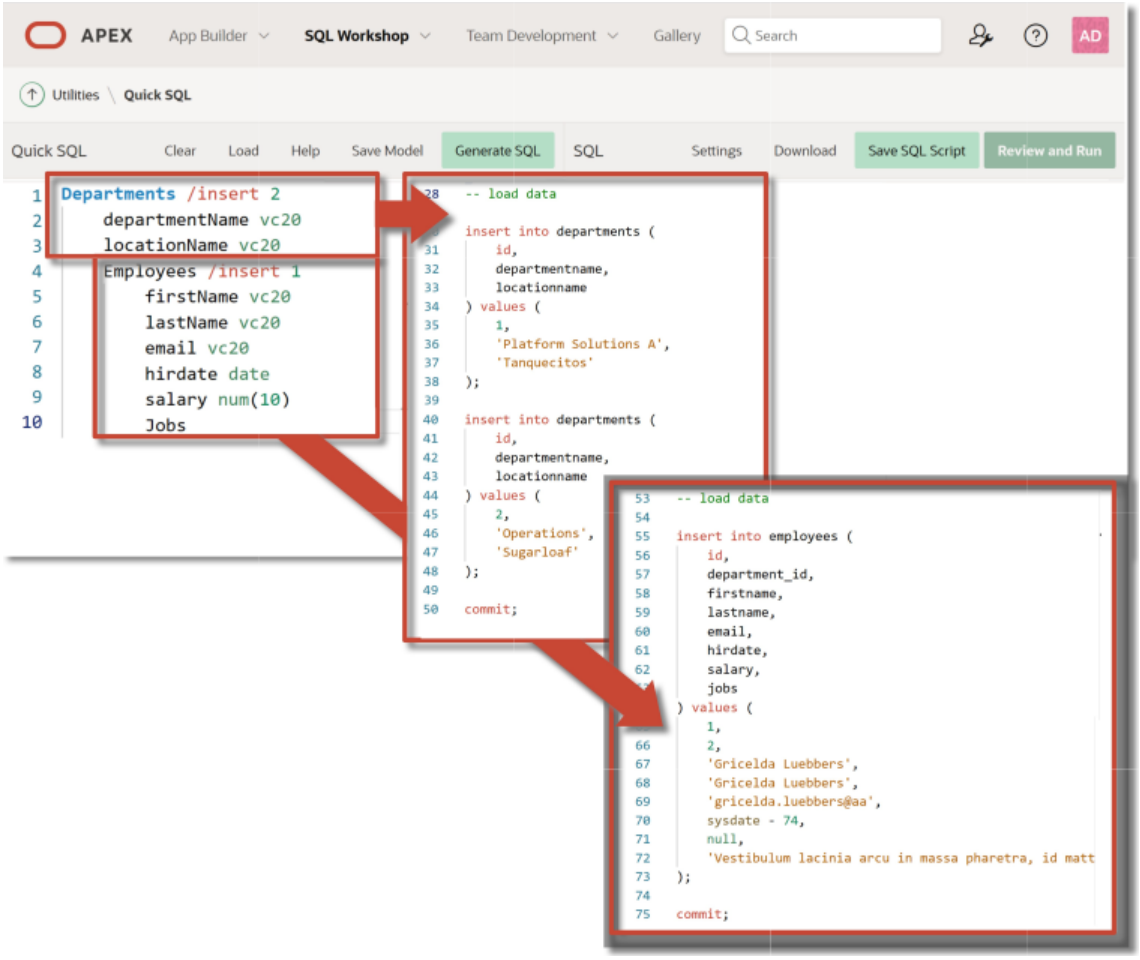
Obr. 2.27: SQL príkaz na vymazanie niektorých údajov z tabuľky `Employees`.

2.3.2 Manipulácia s údajmi pomocou ORACLE Object Browser ("OB")

Okrem tejto programovej možnosti manipulácie s údajmi by bolo možné aj manuálne vkladať údaje do tabuliek, ako aj meniť alebo mazať existujúce údaje pomocou editora, ktorý poskytuje *Object Browser*, pozri Časť 2.2.4, Obrázok 2.21, použitím "Datapane. Je potrebné poznamenať, že každá manipulácia s údajmi, ktorá sa vykonáva pomocou OB, je opäť automaticky preložená serverom ORACLE do zodpovedajúcich príkazov SQL-DML.

2.3.3 Manipulácia s údajmi pomocou Quick SQL

Pomocou Quick SQL je možné do tabuliek automaticky vkladať niektoré údaje. Príklad je zobrazený na Obrázku 2.28 s použitím príkazov "`/insert 2`" a "`/insert 1`". Pomocou týchto príkazov sa definujú početné n-tice s náhodnými hodnotami, ktoré sa majú vložiť do tabuliek.



Obr. 2.28: Náhodné vkladanie údajov pomocou Quick SQL.

2.4 Ako získavať údaje – SQL-DQL

Po diskusii o rôznych možnostiach vytvárania prázdnych tabuliek v Časti 2.2 a možnostiach manipulácie s údajmi v Časti 2.3, sa teraz zameriame na získavanie údajov z tabuliek pomocou SQL príkazu "SELECT" (pozri Časť 2.4.1), pomocou nášho príkladu, po ktorom nasleduje krátky prehľad o nástroji ORACLE Query Builder ("QB"), ktorý poskytuje alternatívu pre "nízko-úrovňové" dopytovanie pre používateľov s malými alebo žiadnymi znalosťami SQL (pozri Časť 2.4.2). Tieto alternatívy sú tiež znázornené na Obrázku 2.45.

2.4.1 Dotazovanie údajov pomocou SQL-DQL

V prvom rade je potrebné poznamenať, že expresivita SQL-DQL je obrovská, takže sa zameriame len na pomerne malú podmnožinu konceptov. Napriek tomu sa snažíme poskytnúť prehľad o širokej použiteľnosti SQL-DQL zavedením rôznych možností dotazovania na príkladoch, ktoré sú usporiadané v podľa zvyšujúcej sa zložitosti. Čo sa týka nástrojov, opäť sa používa SQL Command Editor (pozri Obrázok 2.17), ktorý bol použitý na vytváranie tabuliek a manipuláciu s údajmi.

(1) Začnime najjednoduchším dotazom – *"Získanie všetkých zamestnancov"* (pozri Obrázok 2.29).

```
SELECT * FROM Employees;
```

Obr. 2.29: SQL príkaz na získanie všetkých údajov z tabuľky `Employees`.

Cieľová tabuľka (`Employees`) v dotaze, je špecifikovaná vo `FROM`-klauzule, požadovaný výsledok dotazu je definovaný hneď za `SELECT`-klauzulou - v tomto príklade používame hviezdičku – to znamená, že by sme chceli získať hodnoty všetkých stĺpcov tabuľky.

Časť z výsledku tohto dotazu vyzerá nasledovne (pozri Obrázok 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

Obr. 2.30: Časť z výsledku všetkých údajov z tabuľky `EMPLOYEES`.

Ak by sme chceli výsledok zoradiť napr. podľa platu zamestnancov, musíme jednoducho pridať `ORDER BY`-klauzulu (pozri Obrázok 2.31):

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

Obr. 2.31: SQL príkaz na získanie všetkých údajov z tabuľky `Employees` zotriedené podľa atribútu Salary.

Výsledok tohto dotazu vyzerá nasledovne (pozri Obrázok 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

Obr. 2.32: Časť z výsledku všetkých údajov z tabuľky `Employees` zotriedené podľa atribútu Salary.

(2) Ako je to s dotazovaním len na určité stĺpce a riadky? – *"Získajte meno, adresu a plat*

zamestnancov s platom vyšším ako 5.000"

Po prvé, aby sme získali späť hodnoty iba určitých atribútov, musíme ich špecifikovať namiesto hviezdičky a oddeliť čiarkami. Po druhé, aby sme získali iba určité riadky, musíme zaviesť `WHERE`-klauzuku a zadať logickú podmienku pre príslušný atribút (v našom prípade plat). Pre rôzne atribúty môže existovať niekoľko takýchto podmienok oddelených kľúčovými slovami `AND` / `OR` (pozri Obrázok 2.33).

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

Obr. 2.33: SQL Príkaz na získanie len určitých údajov z tabuľky `Employees`.

Výsledok tohto dotazu vyzerá nasledovne (pozri Obrázok 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

Obr. 2.34: Časť výsledku len určitých údajov z tabuľky `Employees`.

Ak by sme chceli získať hodnoty atribútu "`hireDate`" v inom formáte, mohla by sa použiť funkcia `TO_CHAR()`" poskytovaná ORACLE SQL. Táto funkcia používa hodnotu `DATE` ako prvý vstupný parameter, a druhý parameter požadovaný formát. V nasledujúcom texte je zobrazená rozšírená verzia vyššie uvedeného príkladu (pozri Obrázok 2.35), ktorá teraz používa funkciu `TO_CHAR()` pre "`hiredate`" (úplný zoznam možných spôsobov formátovania je k dispozícii v dokumentácii Oracle):

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

Obr. 2.35: SQL príkaz na získanie údajov z tabuľky `Employees` s použitím funkcie `TO_CHAR()`.

Výsledok tohto dotazu vyzerá nasledovne (pozri Obrázok 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

Obr. 2.36: Časť výsledku získaných údajov z tabuľky `Employees` s použitím Funkcie `TO_CHAR()`.

(3) Ako je to s dotazovaním niekoľkých tabuliek? – "*Získajte všetkých zamestnancov a ich oddelenia*" Načítanie údajov dvoch alebo viacerých tabuliek v rámci jedného príkazu `SELECT` si vyžaduje pridanie klauzuly "*Join*". Spojenie v skutočnosti kombinuje riadky z dvoch alebo viacerých tabuliek, ktoré musia byť špecifikované v klauzule `FROM`. Okrem toho potrebujeme `WHERE`-klauzulu, ktorá hrá úlohu *sprostredkovateľa* medzi týmito dvoma tabuľkami – keďže musíme nejako definovať, že DBS by mal pre každého zamestnanca v tabuľke `Employees`, vziať hodnotu Departments_ departmentID (čo je FK) a porovná túto hodnotu s príslušnou hodnotou PK departmentID z tabuľky `Departments` a vráti údaje oddelení spolu s údajmi o zamestnancoch. Formálnejšie je toto vytváranie spojení špecifikované takzvaným "*join-condition*" ako je možné vidieť v nasledujúcom príklade (pozri Obrázok 2.37):

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

Obr. 2.37: SQL príkaz na spojenie tabuľky `Employees` a tabuľky `Departments`.

Výsledok tohto dotazu vyzerá nasledovne (pozri Obrázok 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

Obr. 2.38: Časť výsledku určitých údajov spojením tabuľky `Employees` a tabuľky `Departments`.

Všimnite si, že príslušné atribúty spájania sú tiež *"kvalifikované"* pomocou zodpovedajúcich názvov tabuliek (oddelených "bodkou"), čím sa usiluje o lepšiu čitateľnosť - v prípade, že názov FK by bol rovnaký ako názov PK - zabezpečenie jednoznačnej identifikácie.

(4) Ako je to s agregovaním hodnôt? – "*Získajte počet zamestnancov a súčet ich platu*"

Na získanie počtu všetkých zamestnancov v tabuľke `Employees` môžeme použiť funkciu SQL `COUNT` cez rôzne `employeeID`. Na zostavenie súčtu všetkých hodnôt môžeme použiť funkciu `SUM` nad atribútom `salary`, keďže v našom prípade chceme vypočítať súčet všetkých platov (pozri Obrázok 2.39):

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

Obr. 2.39: SQL príkaz na výpočet počtu zamestnancov a zostavenie súčtu ich platov z tabuľky `Employees`.

Výsledok tohto dotazu vyzerá nasledovne (pozri Obrázok 2.40):

COUNT(EMPLOYEEID)	SUM(SALARY)
15	149408

Obr. 2.40: Výsledok získania počtu zamestnancov a zostavenia súčtu ich úlatov z tabuľky `Employees`.

(5) Ako je to s grupovaním určitých riadkov – "*Získajte počet zamestnancov zoskupených podľa ich platov v zostupnom poradí*"

Na zoskupenie rovnakých údajov môžeme použiť klauzulu `GROUP BY` v príkaze `SELECT`. V kombinácii s SQL funkciou `COUNT` cez rôzne `employeeID` sme teraz schopní určiť počet riadkov pre každú skupinu. Na triedenie výsledku môžeme použiť funkciu `ORDER BY` ktorá triedi záznamy podľa zvoleného atribútu. `ASC` definuje vzostupné poradie, zatiaľ čo `DESC` definuje zostupné poradie (pozri Obrázok 2.41):

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

Obr. 2.41: SQL príkaz pre zoskupenie údajov podľa platu zamestnancov a zoradenie výsledku podľa počtu zamestnancov s týmto platom z tabuľky `Employees`.

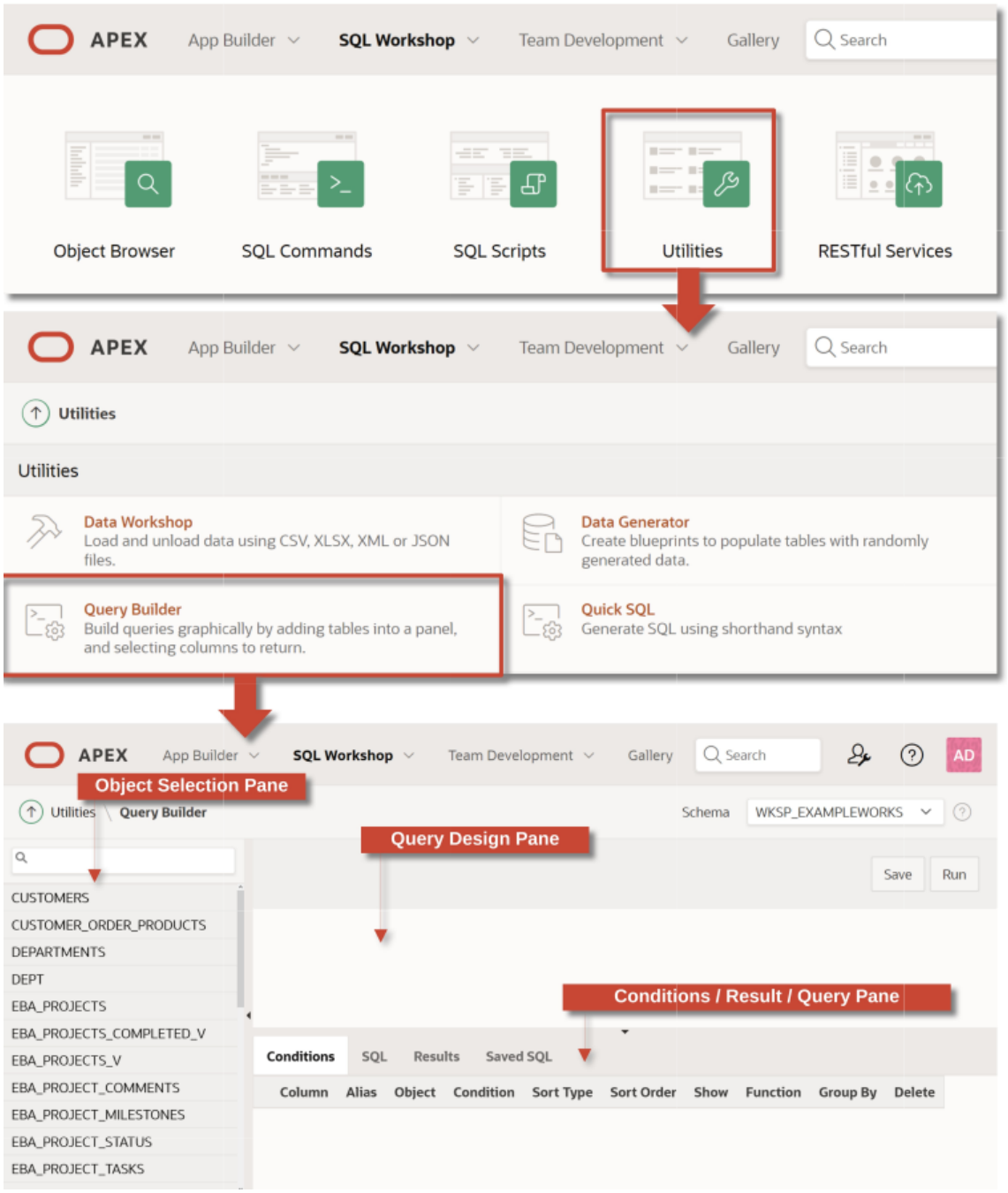
Výsledok tohto dotazu vyzerá nasledovne (pozri Obrázok 2.42):

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

Obr. 2.42: Časť z výsledku zoskupenia údajov podľa platu zamestnancov a zoradenie výsledku podľa počtu zamestnancov s týmto platom z tabuľky `Employees`

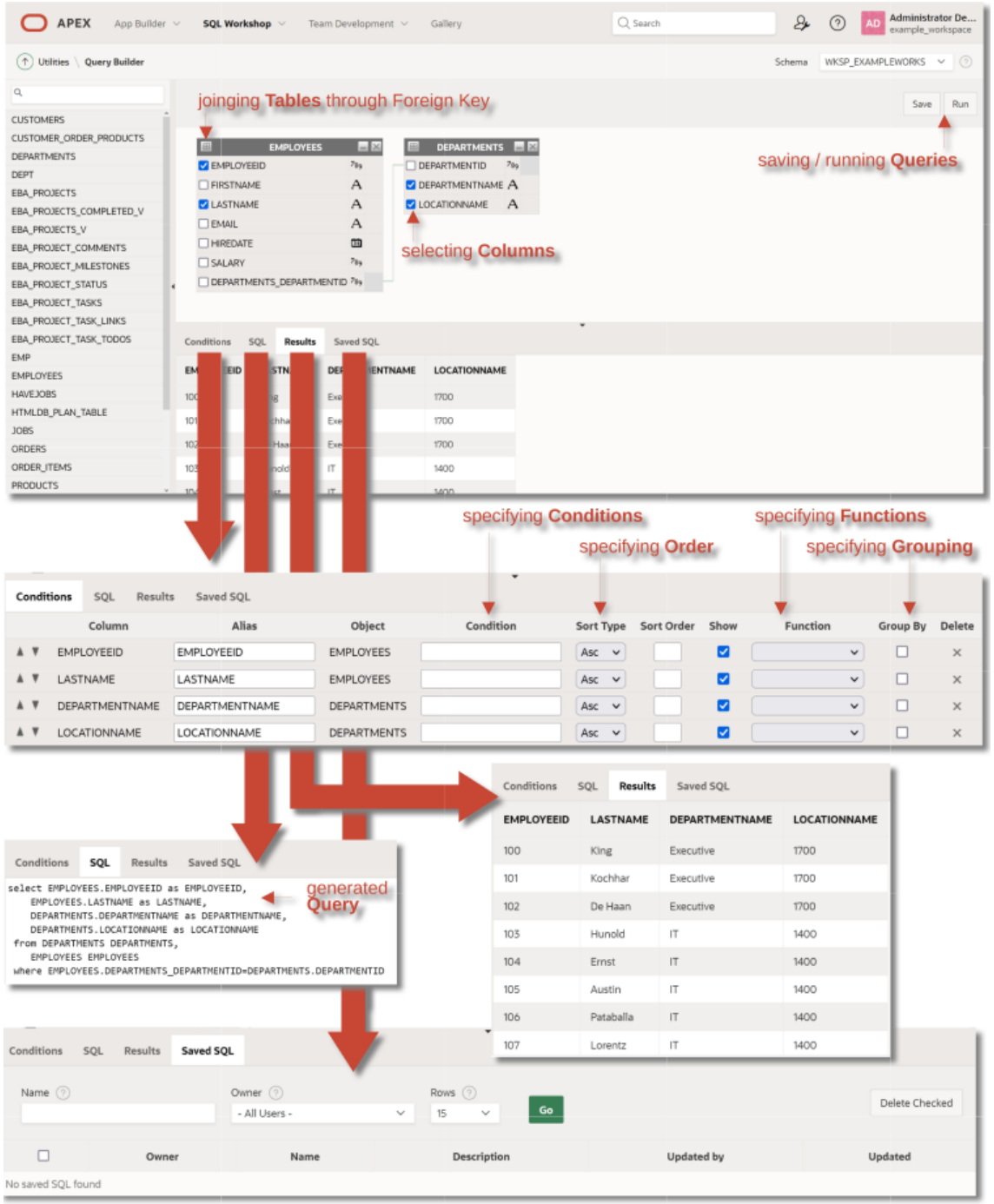
2.4.2 Dotazovanie údajov pomocou ORACLE Query Builder (“QB”)

Ako už bolo spomenuté, ORACLE Utilities vo forme takzvaného Query Builder (“QB”) umožňuje užívateľom s malými alebo dokonca žiadnymi znalosťami SQL dotazovať sa na DB tabuľky a ukladať dopyty pre ďalšie použitie. Ku QB je možné pristupovať nasledujúcim spôsobom (pozri Obrázok 2.43):



Obr. 2.43: Prehľad nástroja ORACLE Query Builder (“QB”).

Kroky, ktoré sú potrebné na zostavenie dopytov, sú znázornené na Obrázku 2.44, s použitím príkladu spojenia "Join"znázorneného na Obrázku 2.37 a stručným popisom v nasledujúcom texte:



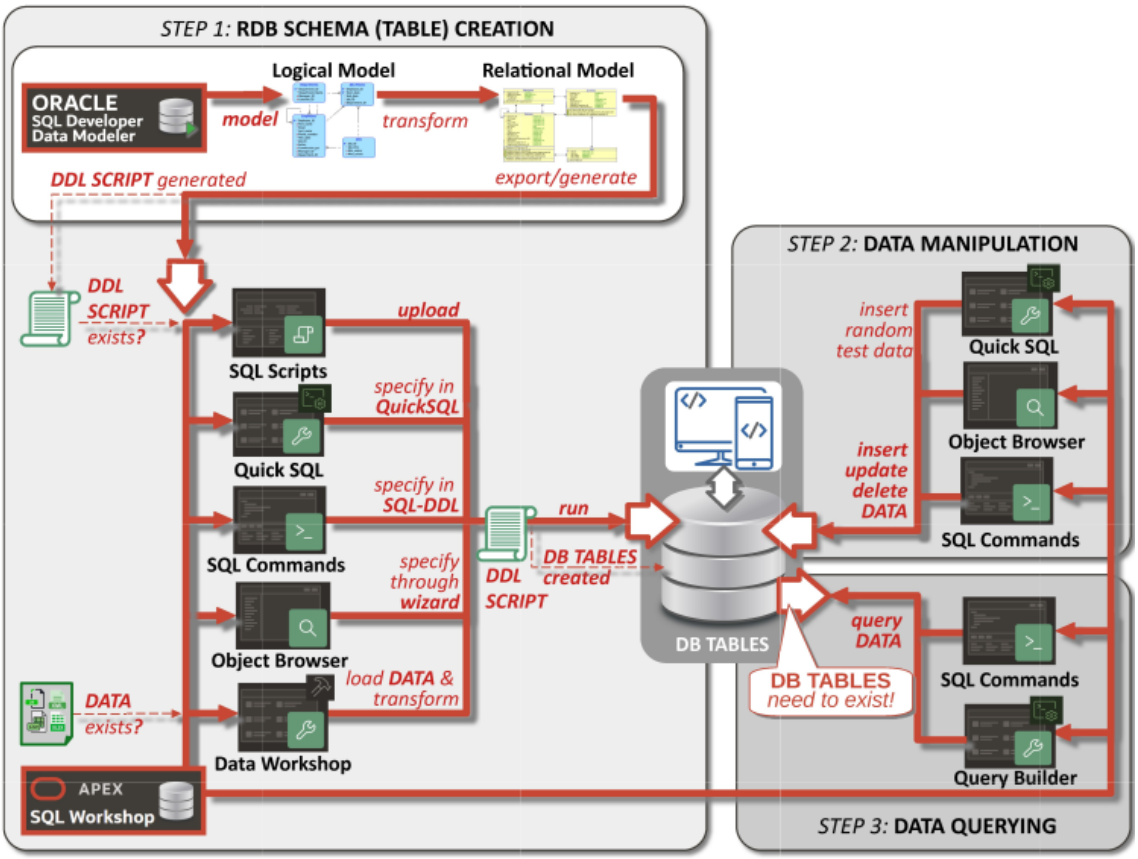
Obr. 2.44: Použitie nástroja Query Builder na vytvorenie spojenia.

- Vyberte *cieľ dopytu* z jednej alebo viacerých DB tabuliek (nazývaných „objects“ v QB) z panelu „Object Selection“ (pozri Obrázok 2.43). Toto tvorí “`FROM`”-časť SQL-DQL.
- Pridajte vybraté objekty do panelu “Design” a ďalej vyberte požadované stĺpce. Toto tvorí “`SELECT`”-časť v SQL-DQL.
- Voliteľné: Vytvorte podmienky dotazu, t. j. časť “`WHERE`” v SQL-DQL.
- Vykonajte dotaz a zobrazte výsledky (kliknite na tlačidlo “Run”).
- Voliteľné: Pozrite si kód SQL-DQL, ktorý sa generuje automaticky (kliknite na záložku “SQL”).
- Voliteľné: Uložte dotaz pre ďalšie použitie (kliknite na tlačidlo “Save”).

Nakoniec je potrebné poznamenať, že dotazy definované pomocou QB sú opäť automaticky preložené RDBS do správnych príkazov SQL-DQL.

2.5 Vytvorenie DB-vrstvy – The Big Picture

Po prečítaní predchádzajúcich častí môže byť človek zmätený všetkými tými rôznymi možnosťami a alternatívami, ktoré poskytuje prostredie nástrojov ORACLE na správu DB-vrstvy webových aplikácií. Obrázok 2.45 poskytuje súhrnný prehľad, ilustrujúci všetky možnosti a nástroje, ktoré sú k dispozícii, aby bolo možné spravovať DB-vrstvu pre webovú aplikáciu.



Obr. 2.45: Prehľad možností správy DB-Layer.

Celkovo sú na ľavej strane zobrazené dve rôzne prostredia nástrojov, ODM a APEX SQL Workshop. Konečný cieľ vytvorenia schémy RDB z hľadiska tabuliek (Step 1) možno dosiahnuť rôznymi spôsobmi. Všetky z nich však vedú ku generovaniu SQL skriptov, ktoré sa nakoniec vykonajú na generovanie prázdnych tabuliek. Odtiaľ je možné vkladať údaje do tabuliek, aktualizovať ich, mazať (Step 2) alebo vyhľadávať údaje (Step 3) pomocou nástrojov SQL Workshopako je popísané v tejto kapitole.

Webová aplikácia môže byť generovaná na prázdnych tabuľkách. Tento proces je podrobne popísaný v Kapitole 3.

2.6 Otázky

1. Aký je rozdiel medzi logickým a relačným modelom? Aké sú prvky týchto dvoch modelov?
2. Čo je to normalizácia RDB-schémy?
3. Popíšte možnosti vytvárania tabuliek pomocou sady nástrojov Oracle.

2.7 Odpovede

1. Logický model je abstraktnejší. Prvky logického modelu sú entity, atribúty a vzťahy. Relačný model predstavuje konkrétnejšiu, a teda pre DBS špecifickejšiu úroveň. Prvky relačného modelu sú tabuľky, stĺpce tabuliek (dátové polia) a vzťahy.
2. Normalizácia RDB-schém je systematický proces s tromi krokmi nazývanými "prvá normálna forma", "druhá normálna forma" a "tretia normálna forma", ktorý vedie k zlepšeniu RDB-schém.
3. Existuje najmenej päť spôsobov vytvárania tabuliek: automatické vytváranie tabuliek pomocou nástroja Oracle Data Modeler, automatické vytváranie tabuliek pomocou nástroja Quick SQL v nástroji Oracle APEX, ručné vytváranie tabuliek pomocou nástroja SQL-DDL v nástroji Oracle APEX SQL Commands, ručné vytváranie tabuliek pomocou nástroja Oracle APEX Object Browser a automatické vytváranie tabuliek pomocou nástroja Oracle APEX Data Workshop.



3. Ako sa pohybovať v nástroji APEX?

E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER

Čo je Oracle APEX? Oracle Application Express (Oracle APEX) je low-code vývojové prostredie založené na webovom prehliadači pre Webové aplikácie využívajúce Oracle DB. "Low-code" znamená, že na vytvorenie plnohodnotnej webovej aplikácie je potrebných len málo programátorských zručností – pre prvý jednoduchý prototyp webovej aplikácie nie sú programátorské zručnosti vôbec potrebné.

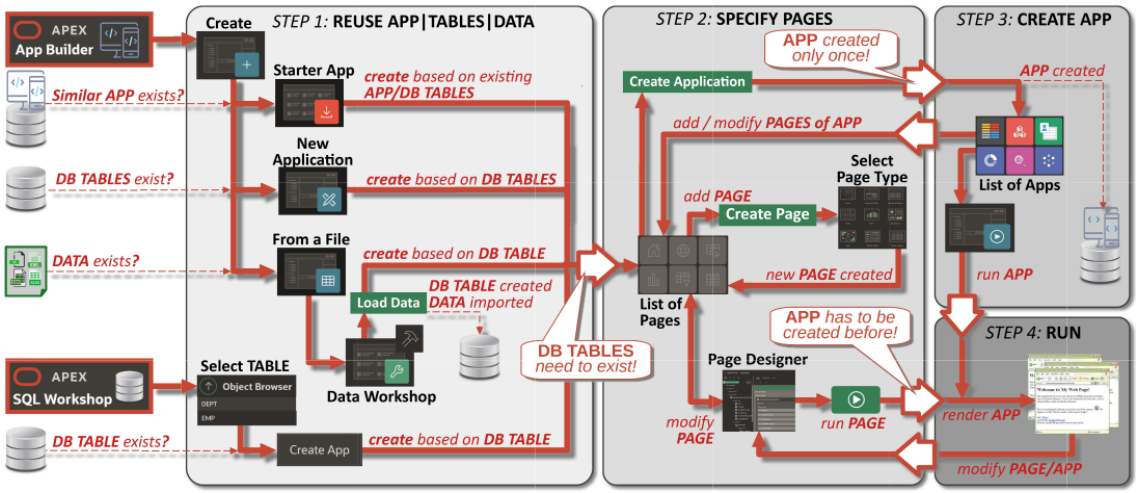
Na aké účely možno APEX použiť? APEX umožňuje nielen zostaviť DB-vrstvu webových aplikácií pomocou APEX "*SQL Workshop*", ako bolo popísané v Kapitole 2, ale aj zostaviť desktopové alebo mobilné webové aplikácie pomocou APEX "*App Builder*". APEX využívajú veľkí aj malí zákazníci v širokom počte aplikačných domén, čím sa vyrovnávajú so širokým spektrom obchodných potrieb. Tieto obchodné potreby môžu siahať od jednoduchej transformácie lokálnej aplikácie na webovú až po realizáciu plnohodnotnej webovej aplikácie (pozri spustiteľný príklad uvedený v Kapitole 2), ktorý umožňuje ukladať, získavať, vizualizovať a ďalej spracovať každý jeden predaj v každom z 10 500 obchodov Walmart spolu s oddeleniami, zamestnancami a informáciami o ich práci.

Aké sú dôvody tejto kapitoly? Dôvodom tejto kapitoly je popísať celkový proces vývoja webových aplikácií, po ktorom nasleduje popis Oracle APEX "App Builder". Tento proces tvorí základ pre možnosti navigácie medzi rôznymi nástrojmi App Builder nazývanými "vývojové komponenty" s cieľom vytvoriť webovú aplikáciu. Po druhé, je uvedený pomerne vysoký prehľad týchto rôznych vývojových komponentov, zatiaľ čo podrobnosti budú popísané v ďalších kapitolách.

3.1 Proces vývoja webových aplikácií pomocou APEX

Predtým, ako prejdeme do podrobností týkajúcich sa rôznych vývojových komponentov APEX "App Builder", je užitočné pozrieť sa na proces vývoja webových aplikácií, po ktorom nasleduje APEX.

APEX využíva viackrokový a cyklický vývojový proces. vývojový proces podporovaný APEX-om, ktorý je znázornený na Obrázku 3.1, e organizovaný *viackrokovým, čiastočne cyklickým spôsobom* umožňujúcim *inkrementálny vývoj*, t.j. postupné zdokonaľovanie webovej aplikácie a prípadne aj DB-vrstvy. Každý z vývojových krokov je podporovaný príslušnými sprievodcami vývojových komponentov. To znamená, že z možností ponuky na najvyššej úrovni sa dostanete hlbšie do funkčnosti *App Builder*, aby ste mohli špecifikovať každý detail webovej aplikácie, krokovať tam a späť, aby ste v prípade potreby vykonali postupné spresnenie.



Obr. 3.1: Vývojový proces v APEX-e.

APEX využíva "DB-Layer First" vývojový proces. Je potrebné zdôrazniť, že keďže sa APEX zameriava na vývoj webových aplikácií riadených DB, je potrebné vždy začať najskôr vývojom DB-vrstvy (pozri Obrázok 3.1 "STEP 1: REUSE APP | TABLES | DATA") . Dá sa to urobiť buď tak, ako je popísané v kapitole 2, alebo priamo v aplikácii App Builder, ak napríklad už existujú nejaké externé dátové súbory, ktoré možno znova použiť na vytvorenie tabuliek DB (pozri Obrázok 3.1 "From a File"). V prípade, že DB tabuľky už existujú, vývojový proces možno spustiť buď (i) z App Builder-a, čím sa prípadne znova použije (a prispôsobí) existujúca aplikácia spolu s jej DB tabuľkami (pozri Obrázok 3.1 "Starter App") alebo vytvorením novej aplikácie od začiatku, opäť na základe existujúcich tabuliek DB (pozri Obrázok 3.1 "New Application"), alebo (ii) z SQL Workshop-u, jednoduchým výberom tabuľky DB v prehliadači objektov (pozri Obrázok 3.1 "Select TABLE") a spustením procesu vývoja aplikácie na základe tejto tabuľky.

APEX využíva "Page-Driven" vývojový proces. Hneď ako budú existovať DB tabuľky na ukladanie údajov webovej aplikácie, môžu sa použiť na vytvorenie novej webovej aplikácie podľa *procesu page-driven* (pozri Obrázok 3.1 "STEP 2: SPECIFY PAGES"). To znamená, že je možné definovať jednu alebo viac webových stránok, ktoré tvoria HTML stránky finálnej webovej aplikácie. Každá z týchto stránok môže byť založená na jednej alebo viacerých DB tabuľkách, čo umožňuje vizualizovať ich údaje z hľadiska rôznych *typov stránok* , ktoré zahŕňajú napríklad interaktívne zostavy, formuláre, zoznamy, grafy alebo kalendáre, ale aj interaktívne "využívanie" údajov (napr. priblížiť detaily), a dokonca s nimi manipulovať a ukladať zmeny späť do databázy. Tieto *stránky môžu byť navzájom prepojené* pomocou navigačného panela, kariet, tlačidiel alebo hypertextových odkazov.

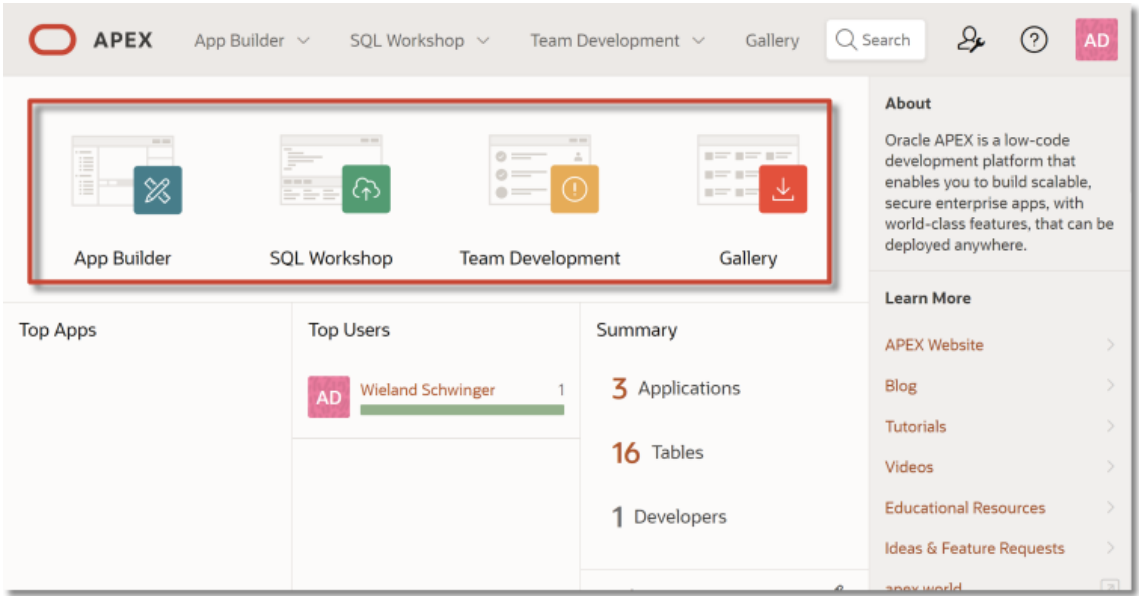
Po vytvorení jednej alebo viacerých stránok je potrebné vytvoriť celkovú aplikáciu (pozri Obrázok 3.1 "STEP 3: CREATE APP"), čím sa ďalej špecifikujú niektoré ďalšie vlastnosti celej webovej aplikácie (napr. vzhľad aplikácie) alebo dodatočné predvolené nastavenia. Nakoniec je potrebné vykresliť aplikáciu a/alebo každú zo špecifikovaných stránok, aby sa vygenerovali finálne HTML stránky (Obrázok 3.1 "STEP: 4 RUN"), ktoré môžu byť ďalej testované a vylepšené

3.2 Start Menu v APEX-e

Keď sa prihlásite do Oracle APEX, zobrazí sa takzvaná *"Workspace Homepage"*. Táto domovská stránka pracovného priestoru poskytuje prístup k štyrom rôznym vývojovým komponentom (pozri Obrázok 3.2):

- *"App Builder"* sa používa hlavne na vytváranie webových stránok a bude bližšie popísaný v tejto kapitole.

- *"SQL Workshop"* sa používa na vytvorenie DB-vrstvy a bol popísaný v Kapitole 2.
- *"Team Development"* sa používa napríklad na interakciu s inými vývojármi prostredníctvom systému lístkov na nastavenie míľnikov, atď.
- *"Gallery"* slúži na jednoduchú inštaláciu vopred zostavených demo aplikácií, ktoré je možné následne ďalej upravovať a prispôsobovať vlastným potrebám, pomocou "*App Builder*".



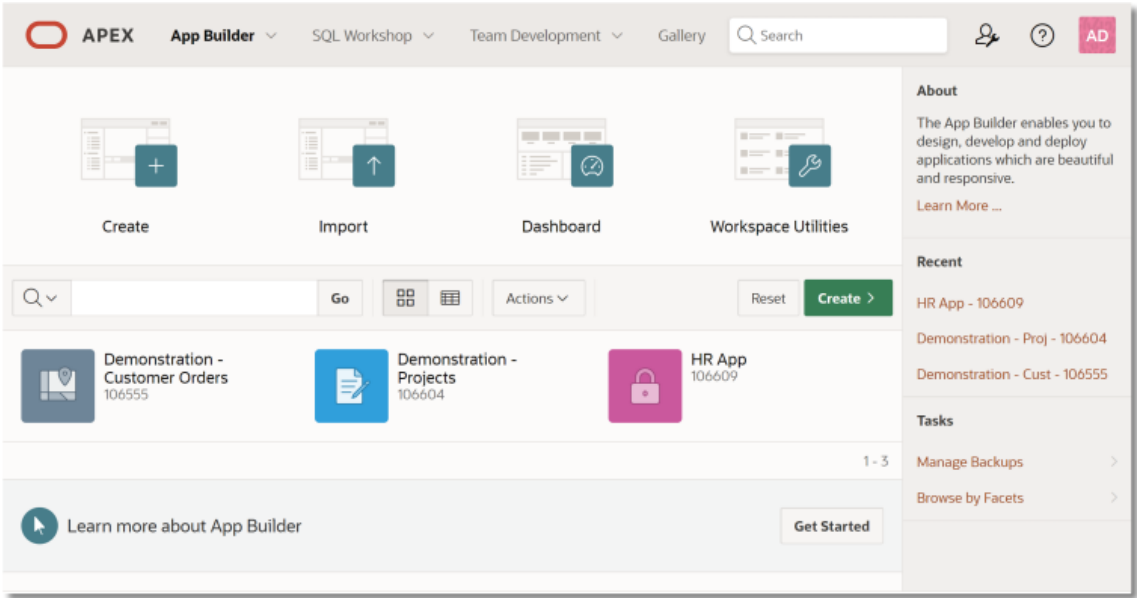
Obr. 3.2: Oracle APEX Workspace Homepage.

V spodnej časti domovskej stránky (pozri Obrázok 3.2), oblasti "*Top Apps*", "*Top Users*" a "*Summary*" ponúkajú informácie o vývojových aktivitách v aktuálnom pracovnom priestore DB v reálnom čase, zatiaľ čo na pravej strane sú uvedené ďalšie vzdelávacie zdroje APEX.

3.3 Prehľad nástroja App Builder – vytvorenie a správa Aplikácie

Pri prechode na domovskú stránku aplikácie App Builder máte k dispozícii nasledujúce možnosti (pozri Obrázok 3.3):

- *Create* - vytvoriť novú webovú aplikáciu.
- *Import* - importovať predtým exportované webové aplikácie.
- *"Dashboard"* - poskytuje štatistiky o už vyvinutých webových aplikáciách.
- *"Workspace Utilities"* - poskytuje širokú škálu služieb, ako je vzdialený prístup k dátam alebo funkcia zálohovania.



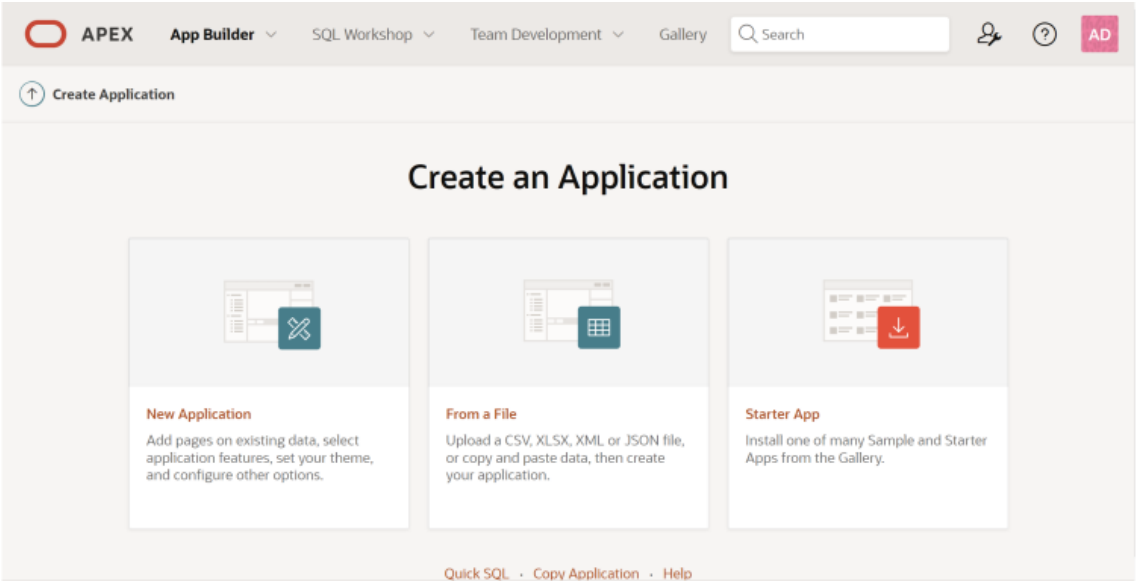
Obr. 3.3: Prehľad nástroja App Builder.

V spodnej polovici okna (pozri Obrázok 3.3) sú uvedené už existujúce webové aplikácie, ktoré je možné vybrať na ďalšie úpravy alebo vykreslenie (pozri nasledujúce časti).

3.4 Vytvorenie aplikácie – tri prípady použitia

Keď v nástroji App Builder vyberiete *"Create"*, "*Create Application Wizard*" zobrazí tri ďalšie možnosti (pozri Obrázok 3.4), ktoré v skutočnosti podporujú prípady použitia v reálnom svete, ktoré zohľadňujú možný potenciál opätovného použitia údajov a samotnej aplikácie:

- *"New Application"* by sa malo vybrať, ak už existuje DB-vrstva (alebo jej časť) alebo ak by sa DB-vrstva mala vyvíjať „za behu" počas vývoja aplikácie.
- *"From a File"* - je možnosť voľby, ak sa má DB-vrstva automaticky generovať z externého súboru, napr. zo súboru EXCEL.
- *"Starter App"* umožňuje opätovné použitie (a pravdepodobne aj ďalšie úpravy) už existujúcej aplikácie spolu s jej DB-vrstvou z "Gallery" aplikácií už spomenutých vyššie.



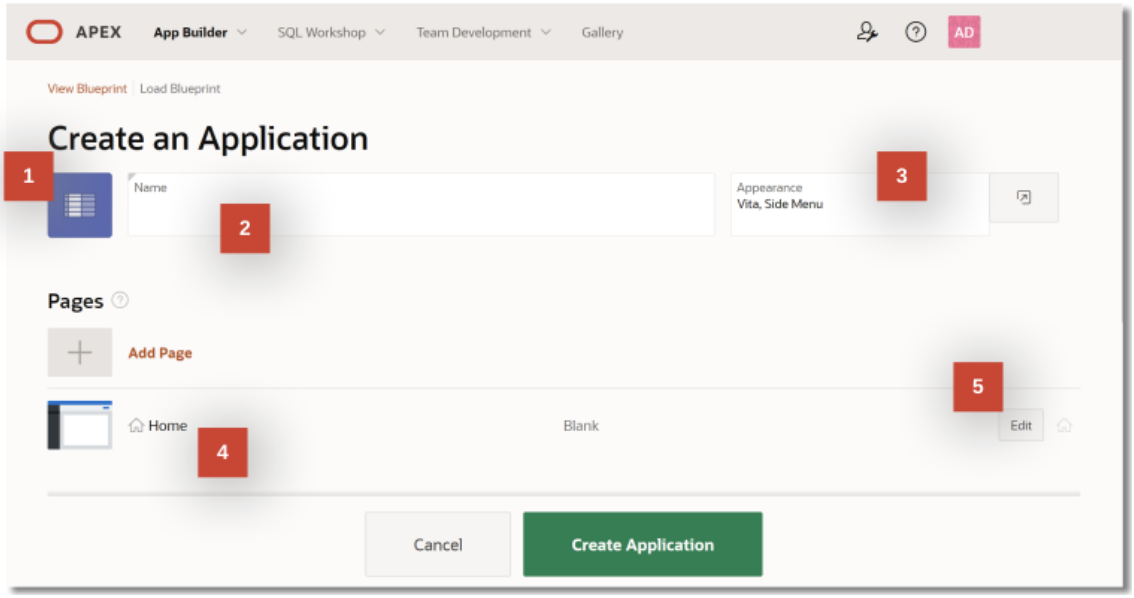
Obr. 3.4: Create Application Wizard: Tri prípady použitia pri vytváraní Aplikácie.

Ďalej sa zameriame na prvú možnosť *"New Application"*, druhú možnosť *"From a File"* popíšeme podrobnejšie v Kapitole 4 a treťou možnosťou *"Starter App"* sa ďalej nebudeme zaoberať.

3.5 Vytvorenie aplikácie – vlastnosti, stránky, funkcie a nastavenia

Pri vytváraní novej webovej aplikácie je potrebné definovať niektoré *spoločné vlastnosti*, vytvoriť *stránky*, a vybrať niektoré ďalšie celkom užitočné *funkcie* a *nastavenia*. Všetky tieto vlastnosti sú bližšie popísané nižšie, pričom Obrázok 3.5 zobrazuje hornú časť GUI a Obrázok 3.6 zobrazuje jeho nižšiu časť.

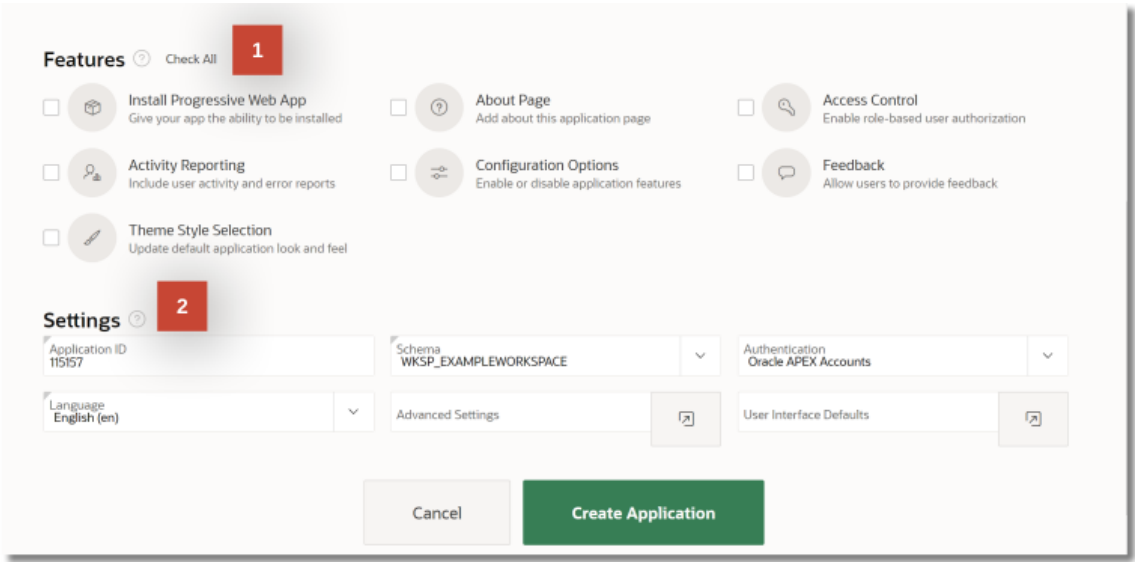
Spoločné vlastnosti – Ikona, Názov a Vzhľad Aplikácie. Čo sa týka bežných vlastností, ako je znázornené na Obrázku 3.5, (1) je možné vybrať alebo nahrať ľubovoľnú *ikonu*, ktorá sa použije ako ikona Favicon a ikona pre PWA, Apple touch alebo App Builder, (2) je potrebné zadať *názov* a (3) *vzhľad* webovej aplikácie je možné vybrať z niektorých preddefinovaných možností, ktoré určujú jej "vzhľad a dojem" z hľadiska štýlu témy (napr. farba, veľkosť písma atď.) a *druh navigačnej ponuky* (napr. bočná ponuka alebo horná ponuka), pričom je k dispozícii aj predvolený vzhľad.



Obr. 3.5: Vlastnosti a stránky.

Vytvorenie stránky pre aplikáciu.. Hlavnou úlohou je však postupné vytváranie webovej aplikácie postupným *pridávaním nových stránok* pomocou *"Create Page Wizard"* (spustený príkazom *"Add Page"*, pozri (4) na Obrázku 3.5. Po vytvorení stránky je možné kedykoľvek počas vývoja aplikácie *upravit' kompozíciu* stránky, *zmenit'* poradie stránok a *odstránit'* ich (pozri (5) na Obrázku 3.5). V spodnej časti Obrázku 3.5 je zobrazená už vytvorená stránka "Home". Ďalšie podrobnosti o "*Create Page Wizard*" sú uvedené v časti 3.6.

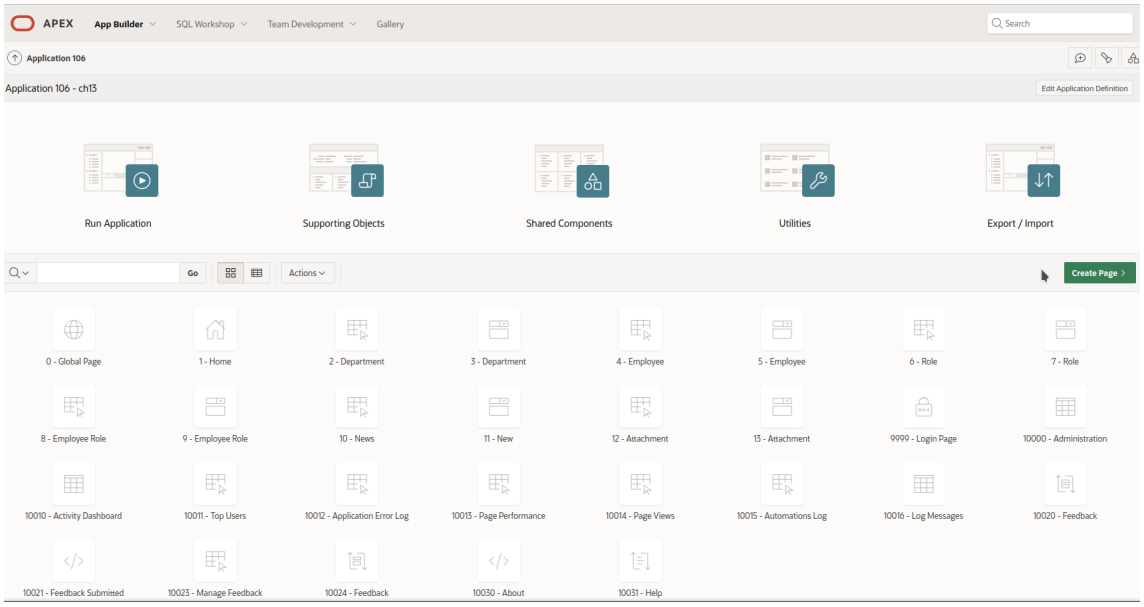
Funkcie a nastavenia aplikácie. Nakoniec môže byť aplikácia vylepšená o určité celkom užitočné *funkcie* (pozri (1) na Obrázku 3.6) , ako je napríklad stránka "About Page", "Access Control" alebo stránka Üser Feedback" ako aj niektoré bežné *nastavenia*, napr. názov schémy obsahujúcej všetky DB-tabuľky a teda tvoriacu DB-vrstvu pre aplikáciu alebo použitý jazyk (pozri (2) na Obrázku 3.6).



Obr. 3.6: Funkcie a Nastavenia.

Po kliknutí na tlačidlo "*Create Application*" sa zobrazí "*App Builder Homepage*" s novovytvo-

renou aplikáciou s jej stránkami (pozri Obrázok 3.7).

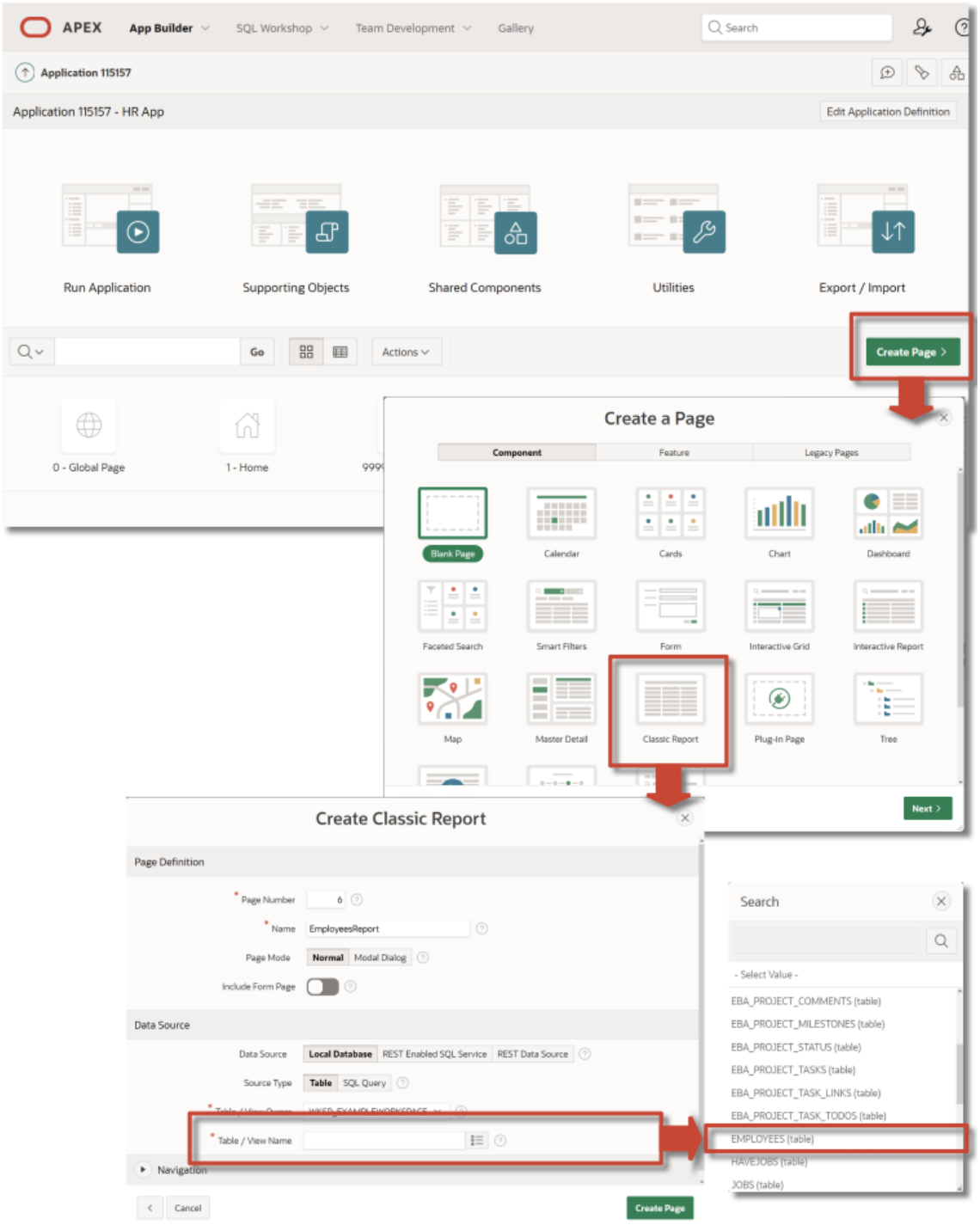


Obr. 3.7: Application Homepage - pohľad vývojára.

3.6 Špecifikácia stránok

Čo tvorí stránku? Ako už bolo spomenuté, stránka je základným stavebným kameňom webovej aplikácie, t.j. každá aplikácia pozostáva aspoň z jednej alebo viacerých stránok. Každá stránka môže mať tlačidlá a polia (nazývané *"items"*) , ktoré sú zoskupené do kontajnerov s názvom *"regions"*. Stránky môžu obsahovať aj *logiku aplikácie*, napr. vykonávať *výpočty* (nazývané *"computations"*) , a vykonávať *validácie* ako sú kontroly správnosti počas manipulácie s údajmi.

Vyberte si z 18 rôznych typov stránok. Stránka môže zobrazovať údaje DB tabuliek v rôznych formách určených takzvanými "page types", ktoré zahŕňajú napr. interaktívne zostavy , zoznamy , kalendáre alebo grafy . Pri každom vytvorení novej stránky je potrebné vybrať typ stránky, a to buď pomocou tlačidla "Add Page" v dialógu "Create Application" (pozri Obrázok 3.5) alebo pomocou tlačidla "Create Page" na domovskej stránke aplikácie vývojára (pozri 3.8). Celkovo je poskytnutých 18 rôznych typov stránok, ako je znázornené na Obrázku 3.8. Určitý typ stránky určuje *obsah*, *kompozíciu*, a *rozloženie* stránky, ako je možné rozpoznať pri pohľade na ikony typu stránky na Obrázku 3.8. Podrobnosti o každom z týchto typov stránok sú uvedené v nasledujúcich kapitolách.



Obr. 3.8: Vytvorenie stránky.

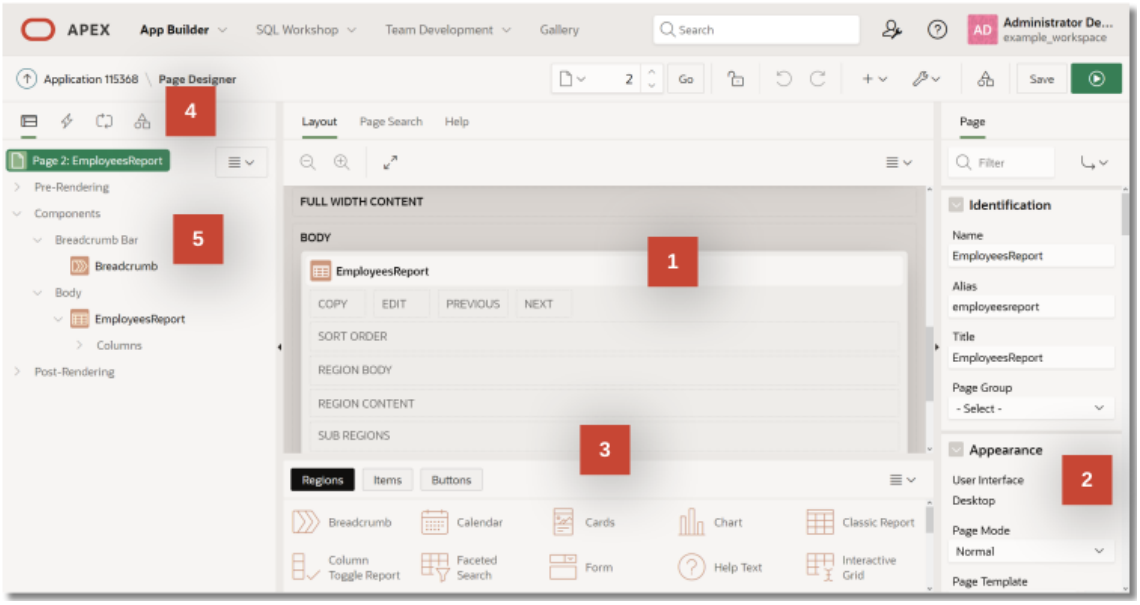
Dialóg so špecifikáciou stránky – mapovanie z tabuľky na stránku. Pri rozhodovaní pre určitý typ stránky je konkrétny dialóg pre špecifikáciu stránky, ktorý predstavuje “Page Creation Wizard” prirodzene závislý od zvoleného typu stránky. Napriek tomu, keďže takmer všetky dostupné typy stránok sú založené na DB-tabuľkách, hlavnou úlohou je obyčajne mapovať vhodné DB-tabuľky na stránku, poskytujúc zdroj pre vizualizáciu údajov a cieľ pre manipuláciu s údajmi. Obrázok 3.8 ukazuje príklad dialógu pre celkom bežný typ stránky “*Classic Report*”. Je vidieť, že okrem niektorých základných informácií, ako je “Name”, je potrebné špecifikovať aj mapovanie na DB-vrstvu výberom “Data Source” v zmysle DB-tabuľky, ktorá by mala byť zdrojom pre túto zo-

stavu (na Obrázku 3.8 vyberieme tabuľku “Employees” nášho príkladu). Je potrebné poznamenať, že na zobrazenie údajov zo základnej DB-tabuľky, APEX vo svojej podstate využíva SQL-DQL na pozadí hneď po vykreslení stránky HTML.

Hneď po pridaní novej stránky sa pre túto stránku automaticky otvorí APEX komponent “*Page Designer*” (pozri Sekciu 3.7), čo umožňuje zobraziť a upraviť špecifikáciu stránky.Zároveň môže byť stránka okamžite a automaticky vykreslená do HTML stránky, napr. na účely testovania a vývoja, pomocou tlačidla “Run” (pozri Sekciu 3.8).

3.7 Udržiavanie a upravovanie stránky – Page Designer

Už vytvorené stránky je samozrejme možné ďalej udržiavať a vylepšovať pomocou takzvaného “*Page Designer*”. Page Designer je plnohodnotné *Integrated Development Environment (IDE)*, ktoré obsahuje panel nástrojov a viacero panelov (pozri Obrázok 3.9). Funkcionalita, ktorú poskytuje Page Designer, je obrovská, okrem iného umožňuje plne upraviť kompozíciu/rozloženie stránky a jej komponentov (pozri (1), (2) a (5)), vrátane položiek, tlačidiel a regiónov (pozri (3)) ale aj na úpravu vykresľovania stránky (pozri (alebo aplikačnú logiku z hľadiska výpočtov a validácií (pozri (4)).



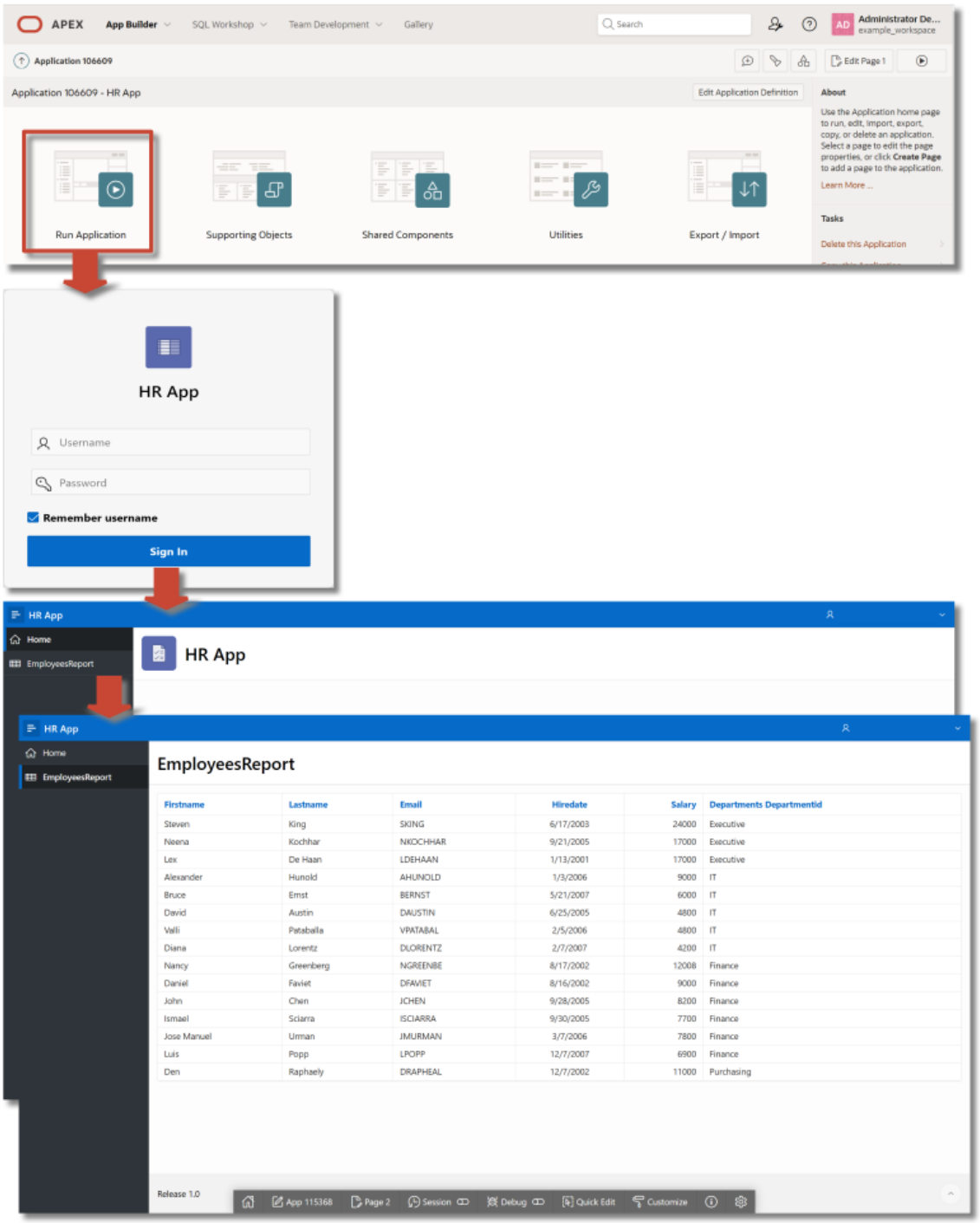
Obr. 3.9: Page Designer.

Podrobnosti o funkciách, ktoré poskytuje Page Designer, nájdete v nasledujúcich kapitolách.

3.8 Spustiť aplikáciu alebo stránku

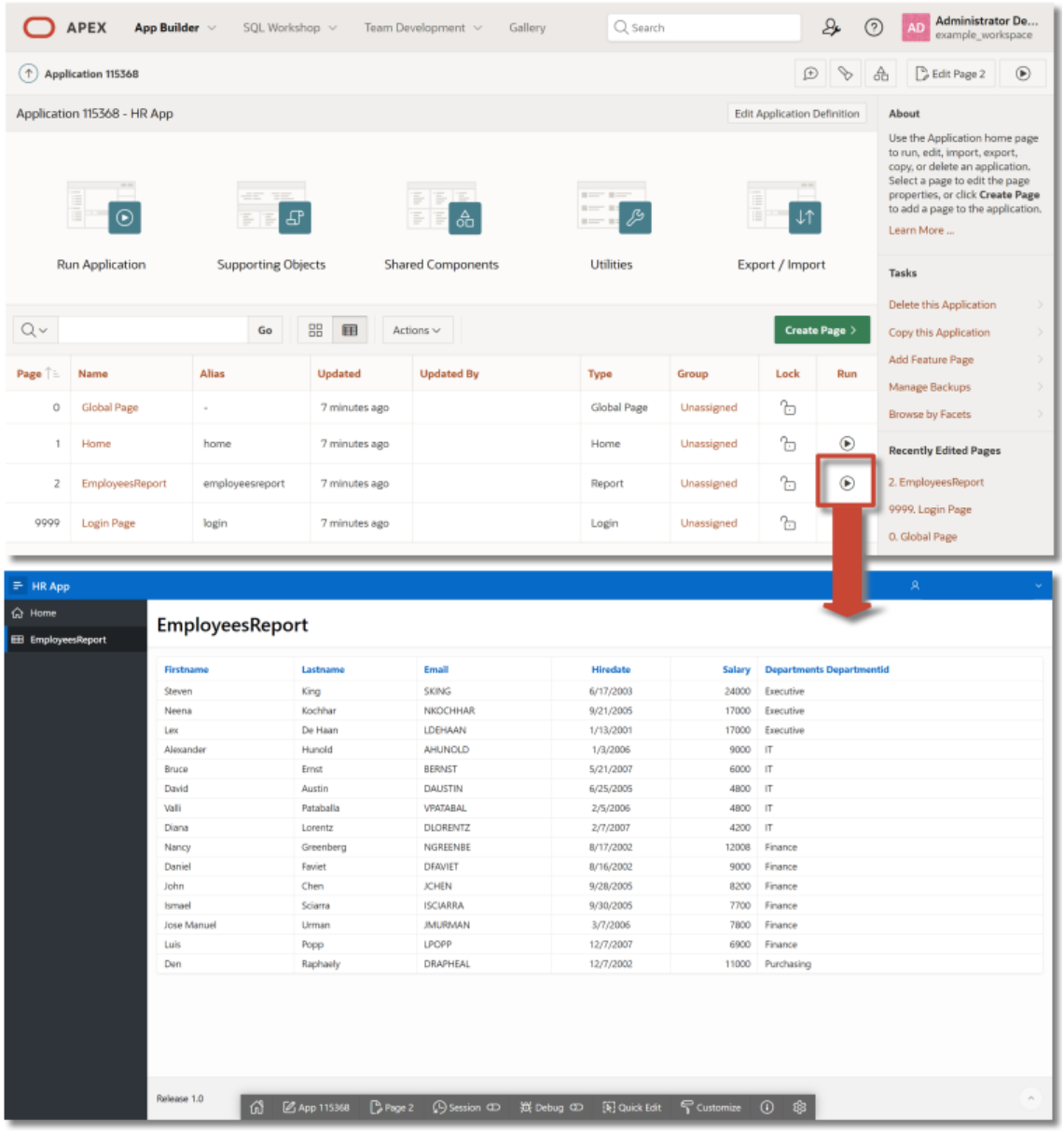
Ak chcete zobraziť vykreslenú verziu aplikácie alebo jednotlivú stránku, musíte ju odoslať do takzvaného *Oracle APEX engine* kliknutím na tlačidlo “*Run*” na *domovskej stránke aplikácie* (pozri Obrázok 3.10 a Obrázok 3.11). APEX engine dynamicky vykresľuje a spracováva stránky do viditeľných *HTML stránok* na základe údajov o aplikácii, ktoré sú uložené v “interných” DB tabuľkách. Na *Application Homepage* sú k dispozícii dve rôzne možnosti v závislosti od toho, či chcete spustiť celú aplikáciu, t.j. všetky stránky, alebo len jednu stránku, napr. na testovacie účely.

- **Spustiť celú aplikáciu**, t.j. všetky stránky: Táto možnosť je dostupná na domovskej stránke aplikácie, ako je vidieť na Obrázku 3.10.



Obr. 3.10: Spustenie Celej Aplikácie.

- **Spustiť jednotlivé stránky**. Pri vytváraní nových stránok je tiež možné spustiť každú stránku samostatne, aby ste získali okamžitý dojem o vzhľade a štýle skutočnej stránky. Tlačidlo “Run Page” pripomína malú čiernu ikonu prehrávania a zobrazuje sa na pravej strane každej zo stránok vašej aplikácie, ako môžete vidieť na Obrázku 3.11. Treba však poznamenať, že aj keď sa spustí len jedna stránka, dostupné sú aj všetky ostatné stránky.



Obr. 3.11: Spustenie jednotlivých stránok.

Všimnite si, že ďalšie možnosti, ktoré sú k dispozícii na domovskej stránke aplikácie okrem “Run Application” (napr. podporné objekty), sú popísané v nasledujúcich kapitolách.

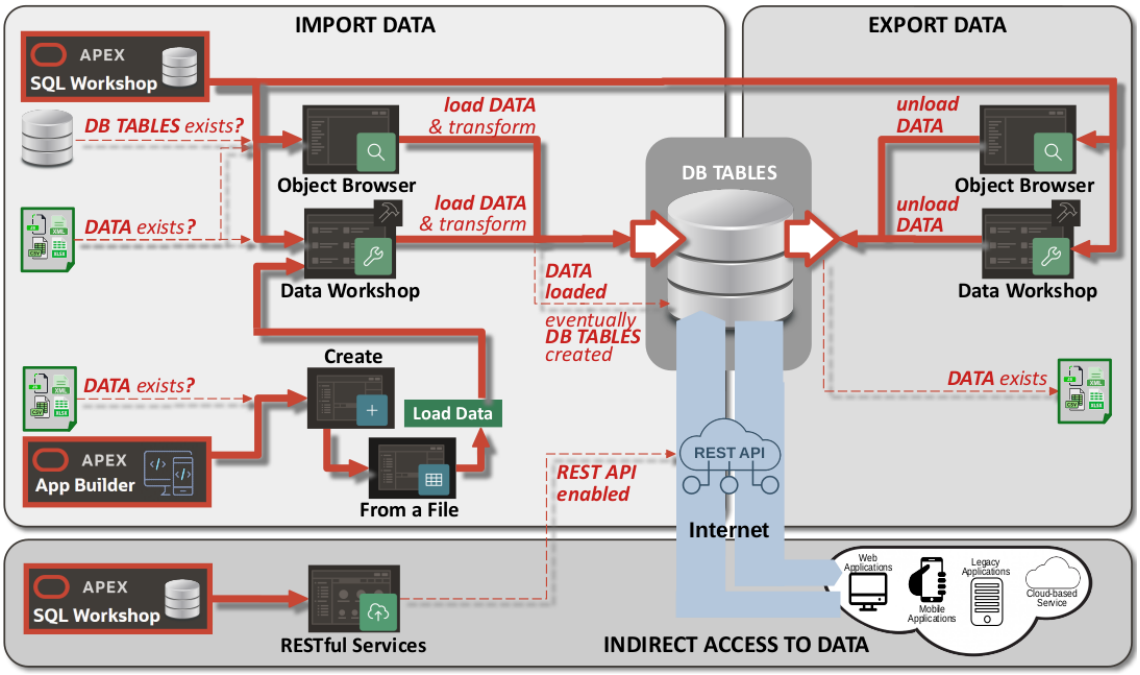


4. Ako si vymieňať dáta v APEX-e?

E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER

Hlavnou úlohou pri vytváraní DB-vrstvy pre webovú aplikáciu je vytvorenie DB-tabuliek a vloženie príslušných údajov (pozri Kapitolu 2)). Toto však nie je možné vykonať iba manuálne, ak údaje ešte neexistujú. V prípade, že sú už existujúce dáta dostupné v externých súboroch, môžu byť jednoducho znovu použité a *importované* do DB Oracle. Zároveň môže byť žiaduce z času na čas *exportovať* už existujúce dáta do súboru, aby ich v druhom kroku mohla znovu použiť iná externá aplikácia. A napokon v prípade, že niektorým externým klientom (napr. iným webovým/mobilným/starším aplikáciám alebo cloudovým službám) by mal byť poskytnutý *online prístup* k údajom v rámci našej databázy, namiesto toho, aby sme si len priebežne vymieňali dátové súbory časom je tu aj možnosť poskytovania nepriameho online prístupu k našim DB údajom.

Tieto tri prípady použitia na výmenu údajov sú znázornené na Obrázku 4.1, ktorý poskytuje súhrn rôznych možností, ktoré Oracle APEX poskytuje na import, export a nepriamy prístup k údajom DB. Cieľom tejto kapitoly je poskytnúť prehľad týchto rôznych možností.



Obr. 4.1: Možnosti výmeny dát v APEX-e.

4.1 Import a export údajov pomocou "Data Workshop"

Scenár Aplikácie pre Data Workshop. Data Workshop je ľahko použiteľný nástroj, ktorý poskytuje "Wizard" na zjednodušenie úlohy importovania/exportovania údajov z/do externých súborov. Data Workshop je obzvlášť vhodný pre aplikačné scenáre, ktoré sa zaoberajú údajmi strednej veľkosti (menej ako 10 tabuliek), ktoré majú iba jednoduché štandardné dátové typy (napr. žiadne polia s viacerými hodnotami alebo vnorenými štruktúrami). Je potrebné poznamenať, že na opätovné použitie a import *obrovských a zložitých súborových údajov* poskytuje Oracle ďalšie vhodné nástroje, napr. takzvaný "*SQL*Loader*". V nasledujúcom texte sa najskôr sekcia 4.1.1 a sekcia 4.1.2 zaoberajú importom údajov, zatiaľ čo sekcia 4.1.3 popisuje export údajov.

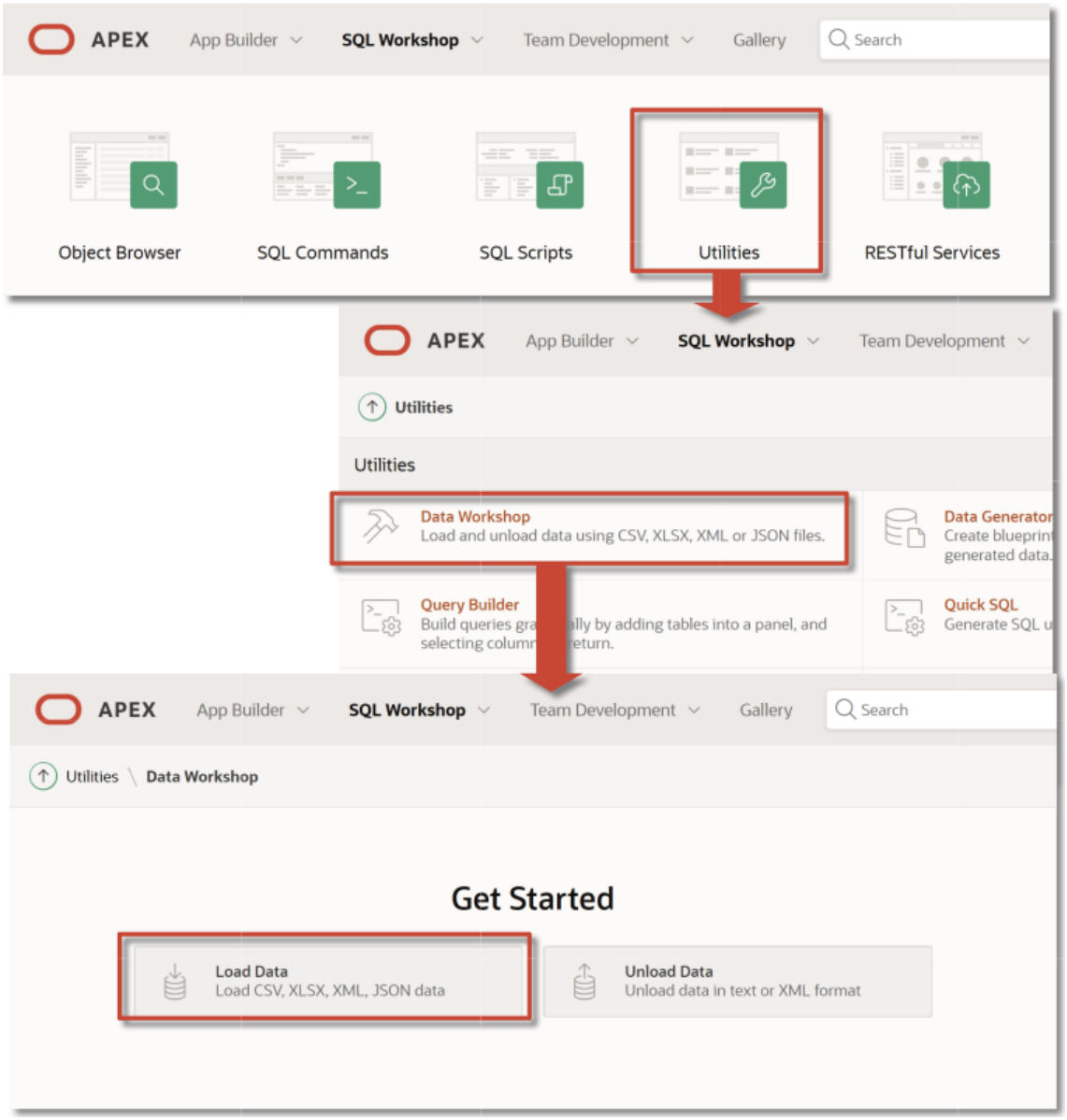
4.1.1 Import údajov

Formáty Súborov. Celkovo umožňuje Data Workshop načítať údaje z externých súborov v nasledujúcich formátoch:

1. Akýkoľvek štandardný *formát s oddeľovačmi*, ktorý je *oddelený tabulátormi* alebo *oddelený čiarkou* ("*CSV*" – *Hodnoty oddelené čiarkou*).
2. *Súbory XLSX* (t.j. *zošity programu Excel*). Upozorňujeme, že ak nahraný súbor XLSX obsahuje viacero pracovných hárkov, predvolene sa vyberie prvý hárok. Ak chcete vložiť ďalší hárok, môžete ho vybrať z výberového zoznamu "Vybrať Hárok".
3. *Súbory JSON* (*Java Script Object Notation*). Upozorňujeme, že je podporovaná iba jedna úroveň vnorenia.
4. *Súbory XML* (*eXtended Markup Language*). Všimnite si, že podobne ako pri súboroch JSON je podporovaná iba jedna úroveň vnorenia.

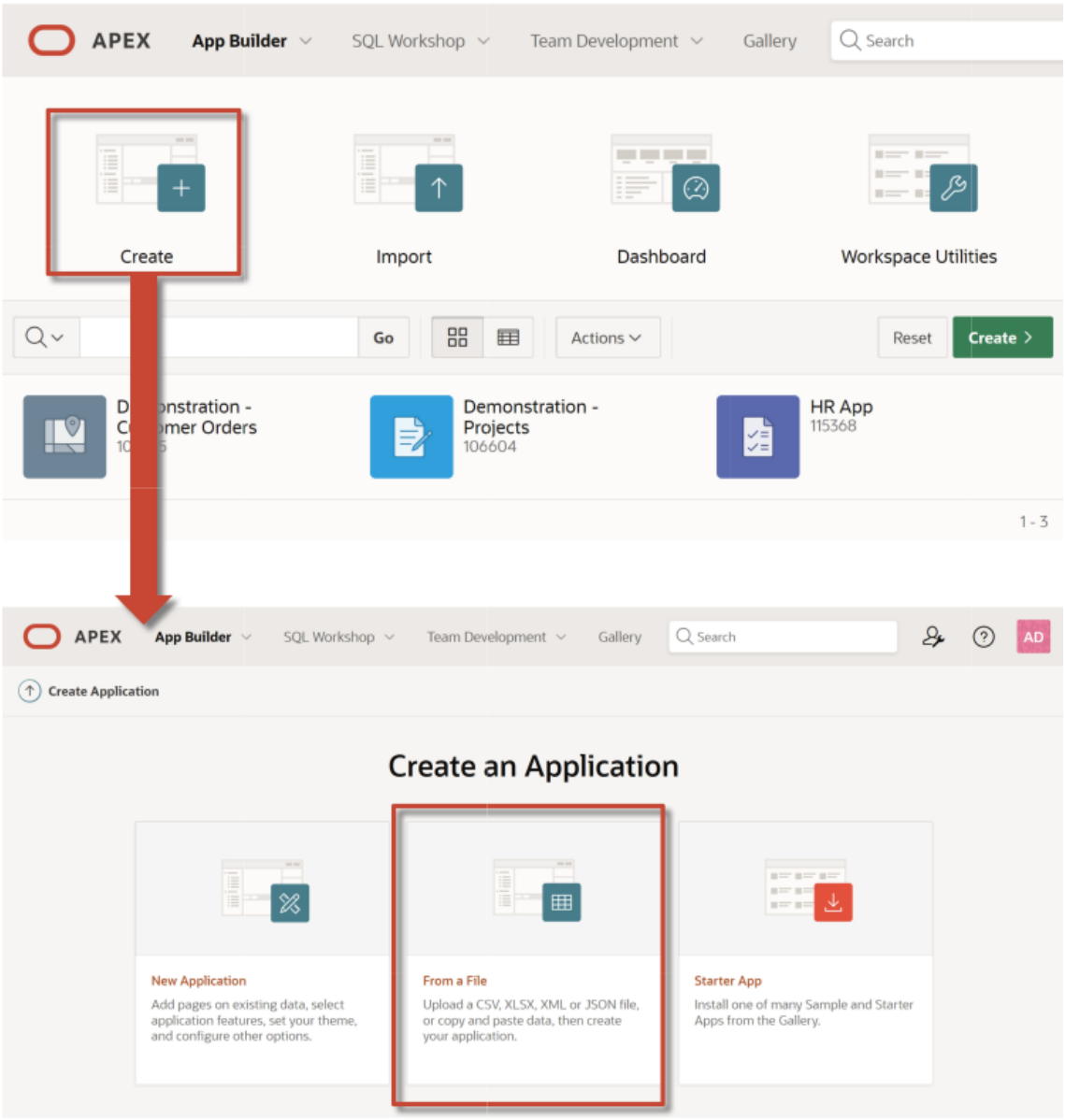
Prístup k Data Workshop pre importy. K Data Workshop je možné pristupovať dvoma alternatívnymi spôsobmi:

1. Spustite Data Workshop mimo SQL Workshop , ako je znázornené na Obrázku 4.2.



Obr. 4.2: Použitie Data Workshop na načítanie údajov - prístup cez SQL Workshop.

2. Alternatívne je možné k Data Workshopu pristupovať aj z nástroja APEX App Builder pri vytváraní aplikácie, ako už bolo uvedené v Kapitole 3 (pozri Obrázok 4.3), pomocou "From a File" možnosti (pozri Obrázok 4.1).

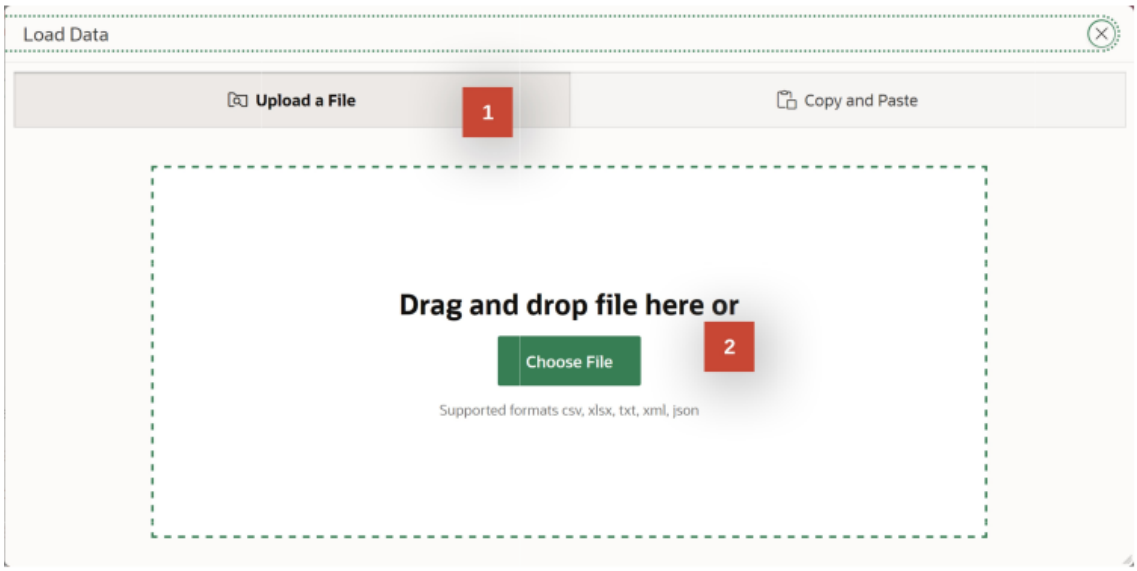


Obr. 4.3: Použitie Data Workshop na načítanie údajov – prístup cez App Builder.

4.1.2 Kroky importu

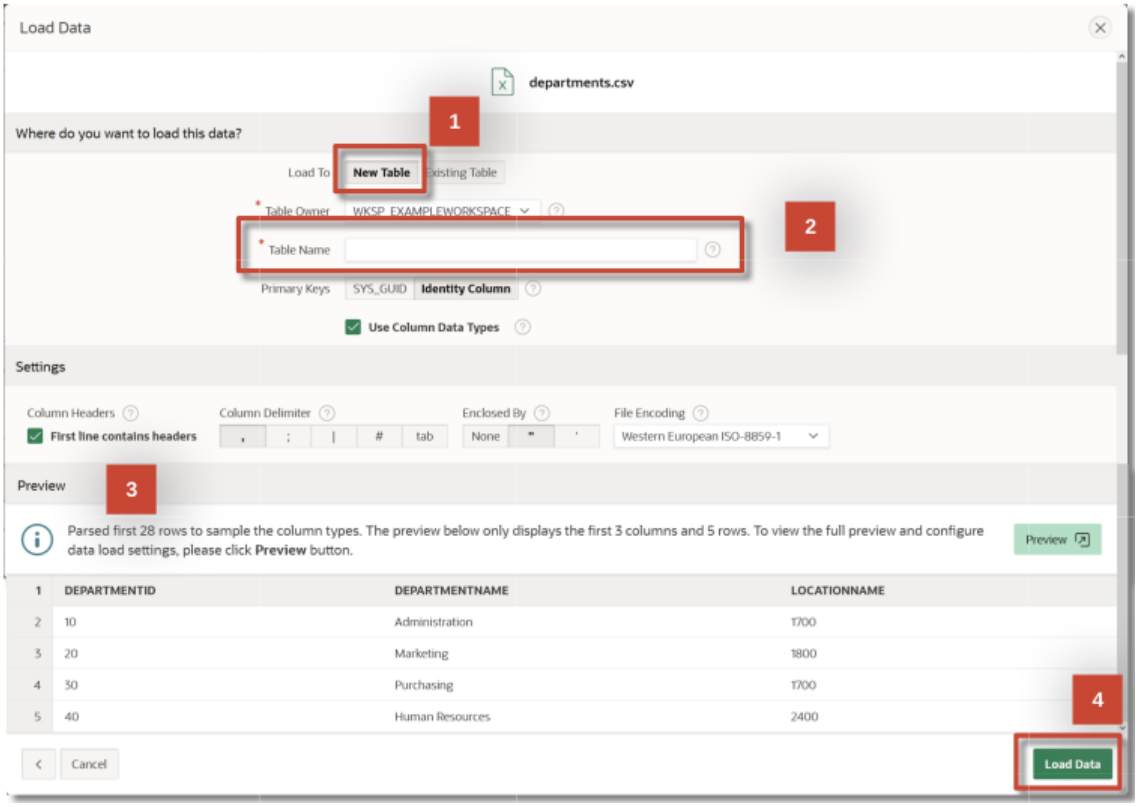
V nasledujúcom texte sú stručne popísané kroky potrebné na import údajov spolu s niektorými dôležitými poznatkami, užitočnými na pochopenie toho, ako používať "Load Wizard":

Poskytovanie zdroja údajov. Prvým, celkom prirodzeným krokom je poskytnutie zdroja údajov, ktorý je možné vykonať buď nahraním (pretiahnutím myšou alebo výberom súboru) oddelených súborov, XLSX, JSON alebo XML súborov, alebo jednoduchým skopírovaním/vložením oddelených údajov (pozri Obrázok 4.4). Všimnite si, že tento dialóg sa objaví buď po kliknutí na tlačidlo *"Load Data"* (pozri Obrázok 4.2) alebo pomocou tlačidla *"From a File"* (pozri Obrázok 4.12).



Obr. 4.4: Poskytovanie zdroja údajov.

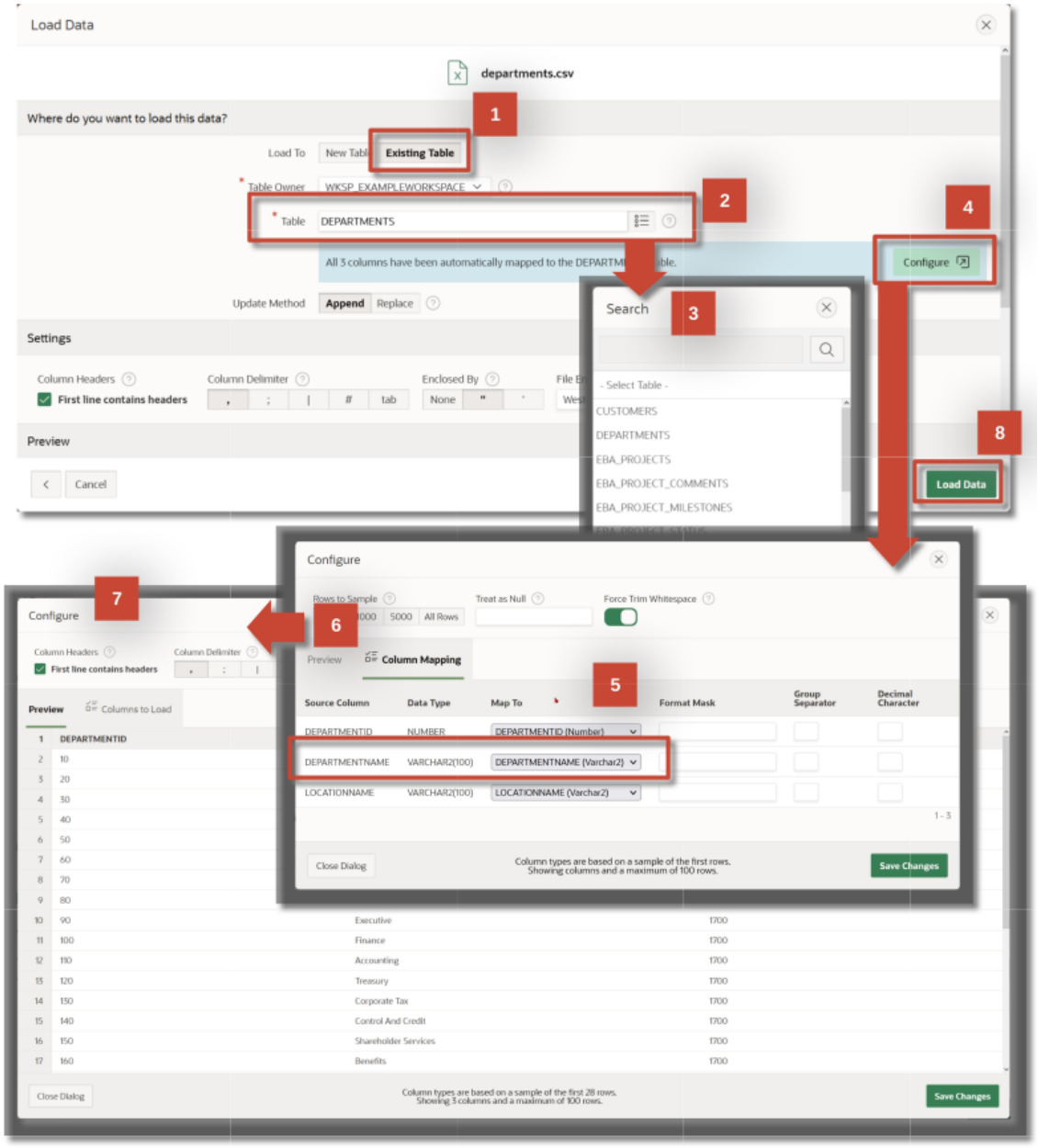
Konfigurácia mapovania dát na tabuľku. Ústredným krokom Load Wizard je konfigurácia mapovania dát na tabuľku, ktorá je znázornená na Obrázku 4.5 a 4.6, kde sú najdôležitejšie možnosti konfigurácie označené číslami (1)-(4) a (1)-(8), ktoré sú popísané nižšie.



Obr. 4.5: Načítavanie údajov do novej tabuľky.

1. **Načítanie údajov do existujúcej alebo novej tabuľky.** Najprv sa treba rozhodnúť, či sa majú údaje uložiť do už existujúcej DB tabuľky alebo či sa má použiť nová DB tabuľka (pozri Obrázok 4.5 (1) a Obrázok 4.6 (1)).

2. **Určenie vlastníka tabuľky a názvu.** V oboch prípadoch je potrebné zadať vlastníka tabuľky a názov tabuľky (pozri Obrázok 4.5 (2) a Obrázok 4.6 (2)).
3. **Nastavenia pre súbory s oddeľovačmi.** *Sekcia nastavení* (pozri Obrázok 4.5 (3)) umožňuje tri rôzne konfigurácie špecifické pre oddeľovače ("CSV") súborov. Voľba konfigurácie "*Column Headers*" umožňuje určiť, či prvý riadok údajov obsahuje názvy stĺpcov. "*Column Delimiter*" je možné ľubovoľne vybrať, ako je možné vidieť v časti "Settings" na Obrázku 4.5 (3) čo umožňuje spoločnosti Oracle *extrahovať štruktúru externého súboru* – proces, ktorý sa nazýva "*parsing*" - parsovanie, ktorý je nevyhnutným predpokladom na mapovanie údajov do tabuľky. Prostredníctvom nastavenia "*Enclosed by*" je možné vymedziť počiatočnú a koncovú hranicu hodnoty údajov. Ak zadáte oddeľovací znak, Data Workshop bude ignorovať medzery vyskytujúce sa pred začiatočnou a koncovou hranicou hodnoty údajov. Túto voľbu môžete použiť aj na uzavretie hodnoty údajov so zadaným oddeľovacím znakom.
4. **Jednoduchá ukážka Parsovaných Údajov** Oracle automaticky parsuje údaje, ako už bolo spomenuté, s cieľom extrahovať štruktúru údajov, a tým tiež zistiť, napr., vhodné typy údajov. V spodnej časti Obrázku 4.5 (4) je zobrazená malá ukážka (do 10 stĺpcov a 5 riadkov) výsledku tohto procesu parsovania.
5. **Rozšírený Náhľad a Ďalšie Konfigurácie.** Po kliknutí na tlačidlo "*Preview*" sa zobrazí maximálne 100 stĺpcov a 100 riadkov a ponúkajú sa ďalšie možnosti konfigurácie, ktoré teraz umožňujú zmeniť aj automaticky navrhované dátové typy a vybrať, ktorý zo stĺpcov sa má skutočne načítať. V prípade, že ste sa v kroku (1) rozhodli načítať údaje do existujúcej tabuľky, tabuľku je potrebné vybrať zo zoznamu dostupných (pozri Obrázok 4.6 (2) a (3)). Potom kliknutím na tlačidlo *"Configure"* (pozri Obrázok 4.6 (4)) je možné *určiť stĺpec mapovania*, t.j. ktoré z automaticky identifikovaných stĺpcov načítaných údajov by sa mali mapovať na ktoré stĺpce v DB-tabuľke. Ako je možné vidieť na Obrázku 4.6 (5) a (6), stĺpec `DEPARTMENTNAME` s dátovým typom `VARCHAR2`, ktorý bol automaticky extrahovaný z externého dátového súboru počas parsovania, je teraz namapovaný na stĺpec DB-tabuľky s názvom "DEPARTMENTNAME"s rovnakým dátovým typom.

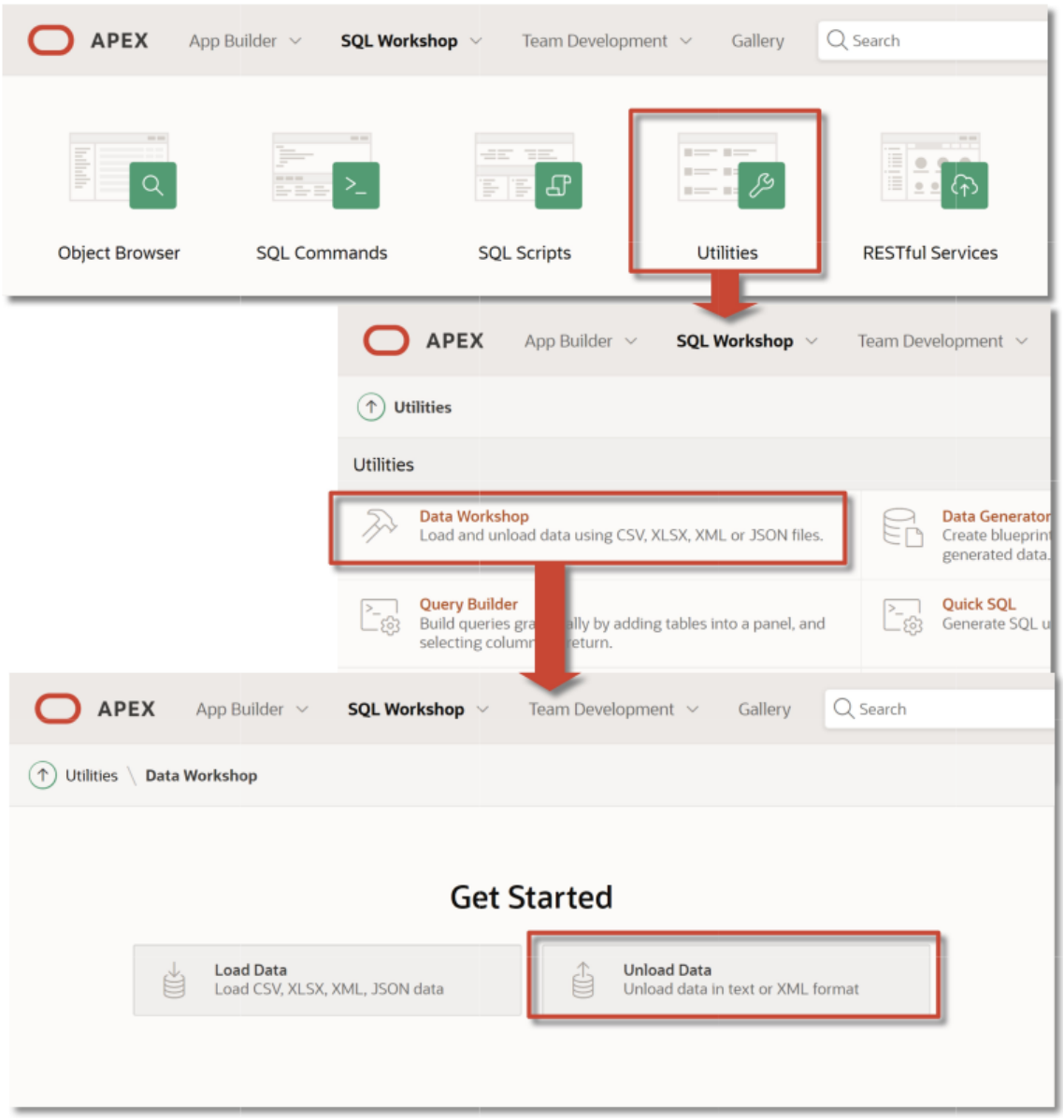


Obr. 4.6: Načítanie údajov do existujúcej tabuľky.

6. **Načítanie údajov.** Nakoniec kliknutím na tlačidlo *"Load Data"* (pozri Obrázok 4.6 (8)) sa vygeneruje nová DB-tabuľka (v prípade táto možnosť bola zvolená v (1)) a údaje sa načítajú do novej alebo existujúcej tabuľky, pričom dialógové okno načítania informuje o počte načítaných riadkov. Proces načítania údajov prebieha na pozadí, čo je výhodné, ak sa nahráva väčší súbor, pretože dialógové okno je možné zavrieť. Počas tohto procesu načítania sa tiež môže stať, že sa vyskytnú niektoré *chybné riadky*, ktoré nie je možné vložiť do cieľovej tabuľky, pretože napr. jedna alebo druhá hodnota je v konflikte s nejakým dátovým typom. V tomto prípade sa tieto chybné riadky automaticky uložia do *tabuľky chýb* a možno ich dodatočne spracovať manuálne. Výslednú tabuľku (a prípadnú tabuľku chýb) je teraz možné zobraziť prechodom do "Object Browser" alebo môžete okamžite pokračovať vo vývoji webovej aplikácie.

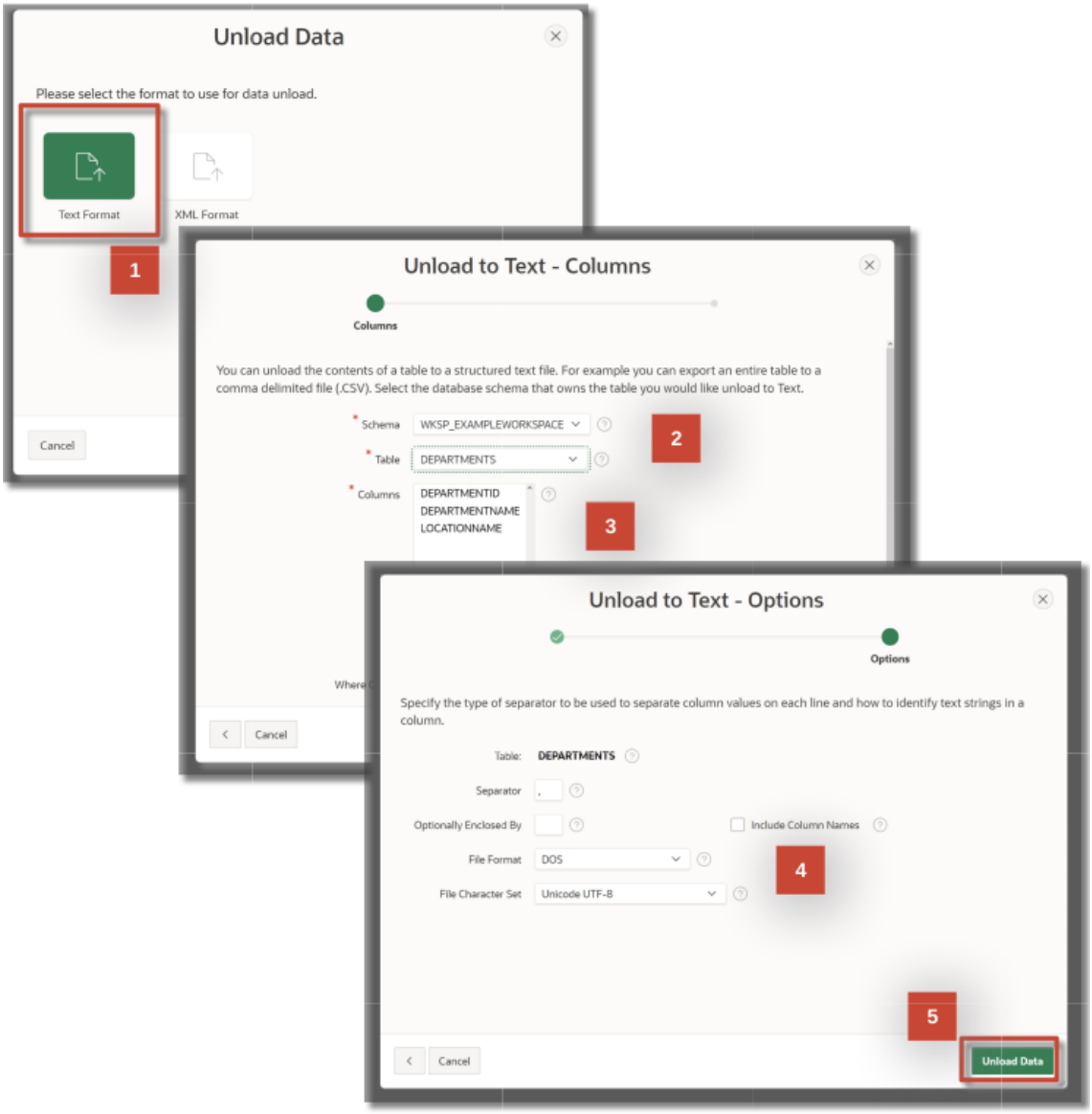
4.1.3 Export údajov

"Export Wizard" Data Workshop-u je možné získať rovnakým spôsobom, ako je popísané v časti 4.1.1 pre "Load Wizard" (pozri Obrázok 4.7).



Obr. 4.7: Export ("Unloading") údajov.

V prvom kroku je potrebné zvoliť formát súboru, pričom Data Workshop umožňuje exportovať dáta do externých súborov formátovaných v CSV alebo XML (pozri Obrázok 4.8 (1)). Ďalším krokom je výber tabuľky, ako aj príslušných stĺpcov, ktorých údaje sa majú exportovať (pozri Obrázok 4.8 (2) a (3) v tomto poradí). V prípade, že bol ako formát exportu zvolený XML, je možné spustiť samotný export. V prípade CSV možno vybrať niektoré ďalšie možnosti, ktoré určujú oddeľovač medzi rôznymi riadkami, či majú byť názvy riadkov zahrnuté vo výstupnom súbore a či má byť formát údajov DOS alebo UNIX (pozri Obrázok 4.8 (4)), predtým ako nakoniec aktivujte "Unload Data" na uloženie exportu dát (pozri Obrázok 4.8 (5)).

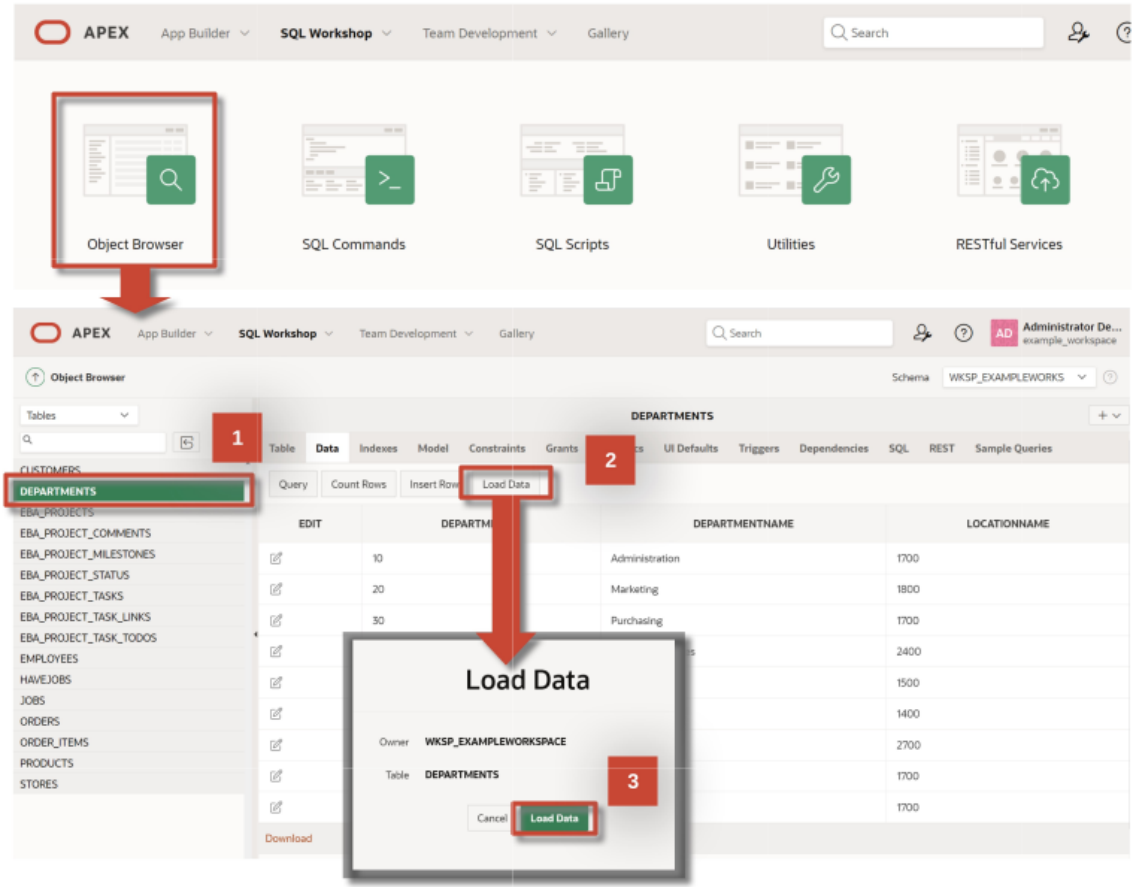


Obr. 4.8: Unload Data Wizard.

Pokiaľ ide o DOS vs. UNIX, uvedomte si, že medzi týmito dvoma možnosťami je jemný rozdiel: súbory DOS majú iné konce riadkov ako súbory vytvorené v systéme Unix/Linux. DOS používa návrat kurzora a posun riadku ("`\r\n`") ako ukončenie riadku, zatiaľ čo Unix používa iba posun riadku ("`\n`"). Preto musíte byť opatrní pri prenose súborov medzi systémom Windows a Unix, aby ste sa uistili, že konce riadkov sú správne nastavené.

4.2 Import a export údajov pomocou "Object Browser"

Ďalšou alternatívou je použiť "Object Browser" SQL Workshop na importovanie údajov zo súboru a export údajov do súboru, pričom to možno vykonať tabuľkovým spôsobom. Najprv je teda potrebné vybrať príslušnú tabuľku (pozri Obrázok 4.9 (1)).

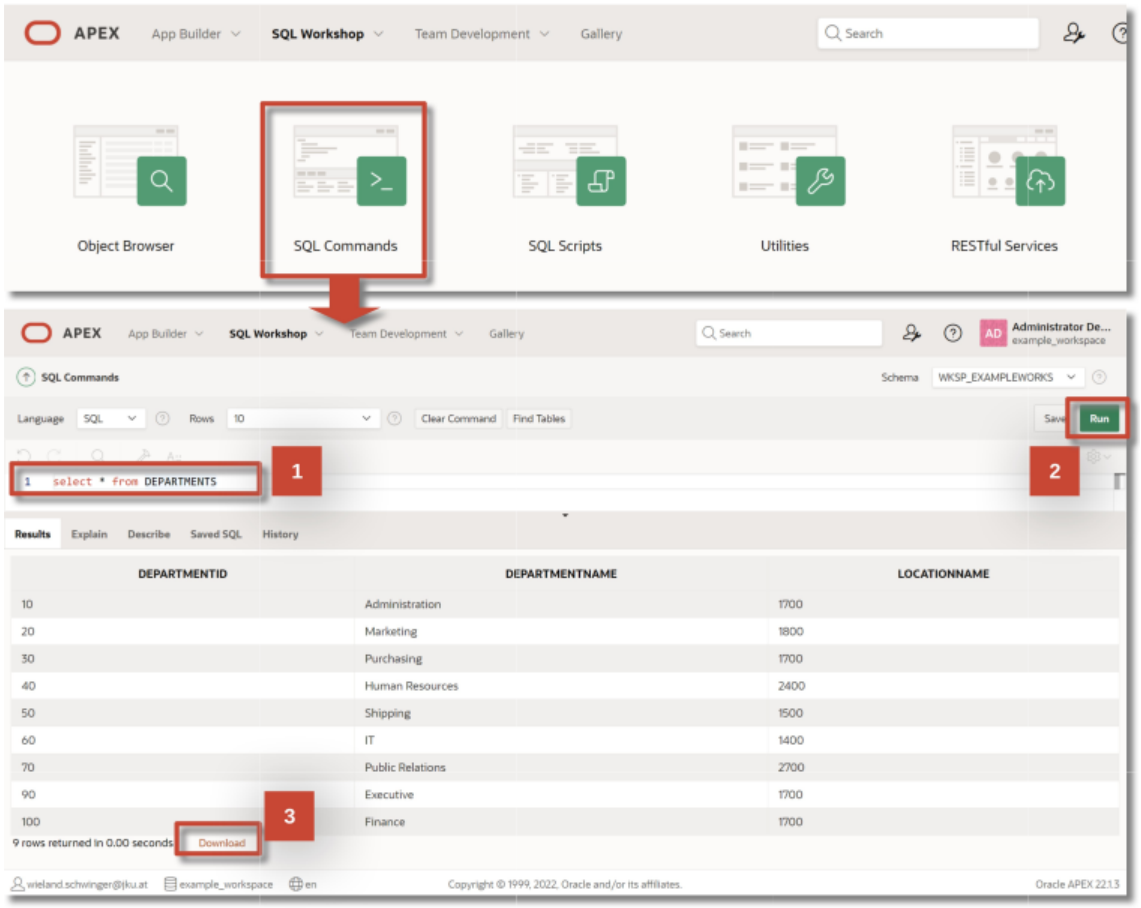


Obr. 4.9: Import/Export údajov tabuľky pomocou Object Browser.

Ak teraz majú byť dáta *importované*, po kliknutí na tlačidlo *"Load Data"* sa zobrazí rovnaký dialóg ako pri výbere "Load Data" (pozri Obrázok 4.9 (2), (3)), ako už bolo popísané v časti 4.1.1 - podporované sú aj rovnaké formáty súborov. Na druhej strane, *export* údajov možno jednoducho vykonať kliknutím na tlačidlo *"Download"* v spodnej časti stránky, čo vedie k okamžitému vygenerovaniu súboru CSV obsahujúceho údaje DB-tabuľky.

4.3 Export výsledku SQL-príkazu

Podobne ako pri exporte celých údajov uložených v tabuľke je možné naraz exportovať aj určitú podmnožinu týchto údajov – alebo dokonca údaje uložené v niekoľkých tabuľkách. Jednoducho povedané, výsledná sada ľubovoľného komplexného SQL-Príkazu môže byť exportovaná (pozri Obrázok 4.10 (1))

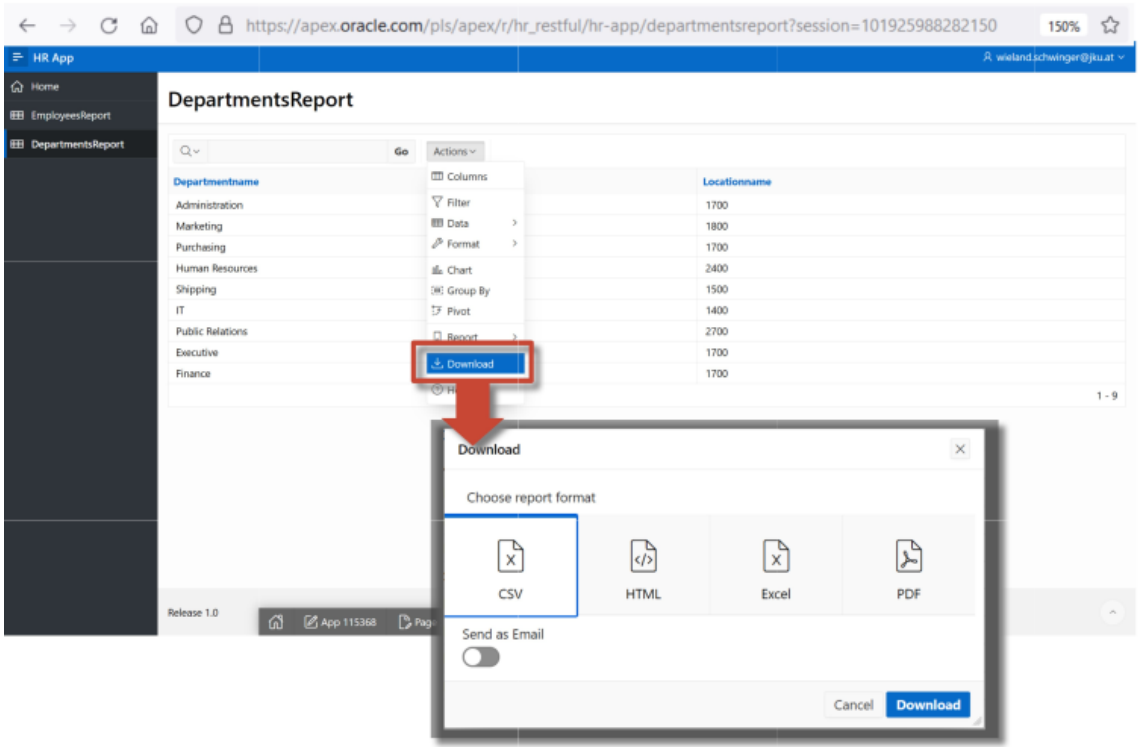


Obr. 4.10: Export údajov výsledku SQL-Príkazu.

Po vykonaní SQL-dotazu v nástroji SQL Workshop a zobrazení sady výsledkov (pozri Obrázok 4.10 (2)) môže byť táto sada výsledkov stiahnutá jednoduchým kliknutím na tlačidlo “Download” (pozri Obrázok 4.10 (3)).

4.4 Export údajov z reportu aplikácie

Ako sme diskutovali v Kapitole 3, vývoj webovej aplikácie pomocou nástroja APEX App Builder vyžaduje výber určitého typu stránky pre každú stránku, ktorá sa pridá do webovej aplikácie. V prípade, že ste zvolili napr. typ stránky “Interactive Report”, údaje zobrazené touto zostavou je možné exportovať. Toto je znázornené na Obrázku 4.11, , ktorý ukazuje jednoduchú stránku interaktívnej zostavy “DepartmentsReport”.

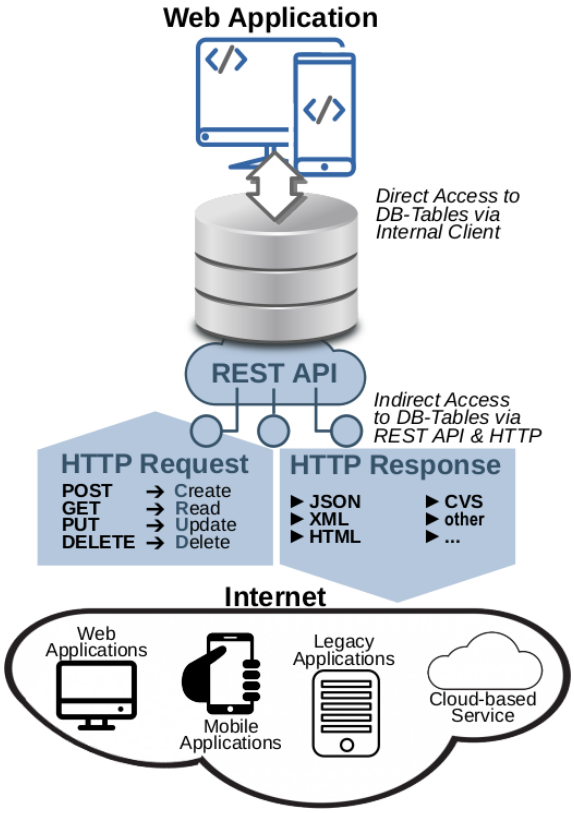


Obr. 4.11: Export údajov z reportu aplikácie

Možné formáty súborov sú CSV, HTML, PDF a Excel, možnosť je aj odoslanie údajov e-mailom.

4.5 Povolenie výmeny údajov so službou RESTful Servis

Výmenu dát DB-vrstvy našej webovej aplikácie tým, že externým klientom, ako sú iné webové aplikácie, mobilné aplikácie, staršie aplikácie alebo cloudové systémy, online prístup k našim dátam možno jednoducho realizovať pomocou takzvaných *RESTful servis*. *REST (Representational State Transfer)* je v skutočnosti *architektonický vzor* navrhnutý Royom Fieldingom a Richardom Taylorom v roku 2000 [2] na zabezpečenie interoperability medzi ľubovoľnými systémami Internetu. Umožňuje dopytovanie a manipuláciu s údajmi *bez potreby priameho prístupu* k podkladovým DB-tabuľkám. Obrázok 4.12 poskytuje prehľad základného architektonického vzoru REST na vysokej úrovni.



Obr. 4.12: RESTful Servis pre výmenu údajov – základná architektúra.

Na realizáciu takéhoto *nepriameho* prístupu podľa architektonického vzoru REST musia byť služby RESTful vytvorené v zmysle sady *APIs (Application Programming Interfaces)* navrchu vašich DB-tabuľiek. Tieto rozhrania API bez priameho pripojenia k základnej databáze umožňujú externým systémom bezpečnú interakciu s údajmi prostredníctvom dotazovania, vkladania, aktualizácie alebo odstraňovania údajov. Definície RESTful služieb vytvorené v rámci Oracle APEX sú uložené v úložisku *Oracle REST Data Services (ORDS)* , a označujú sa ako *ORDS-based REST Services*.

4.5.1 Architektonický vzor REST

Ako už bolo spomenuté, služba sa označuje ako RESTful, keď je v súlade s princípmi REST. Hoci úplná diskusia o REST je mimo rozsahu tejto kapitoly, služba RESTful má nasledujúce charakteristiky (Všimnite si, že všeobecné charakteristiky REST opísané v [2] boli mierne prispôsobené nášmu kontextu DB):

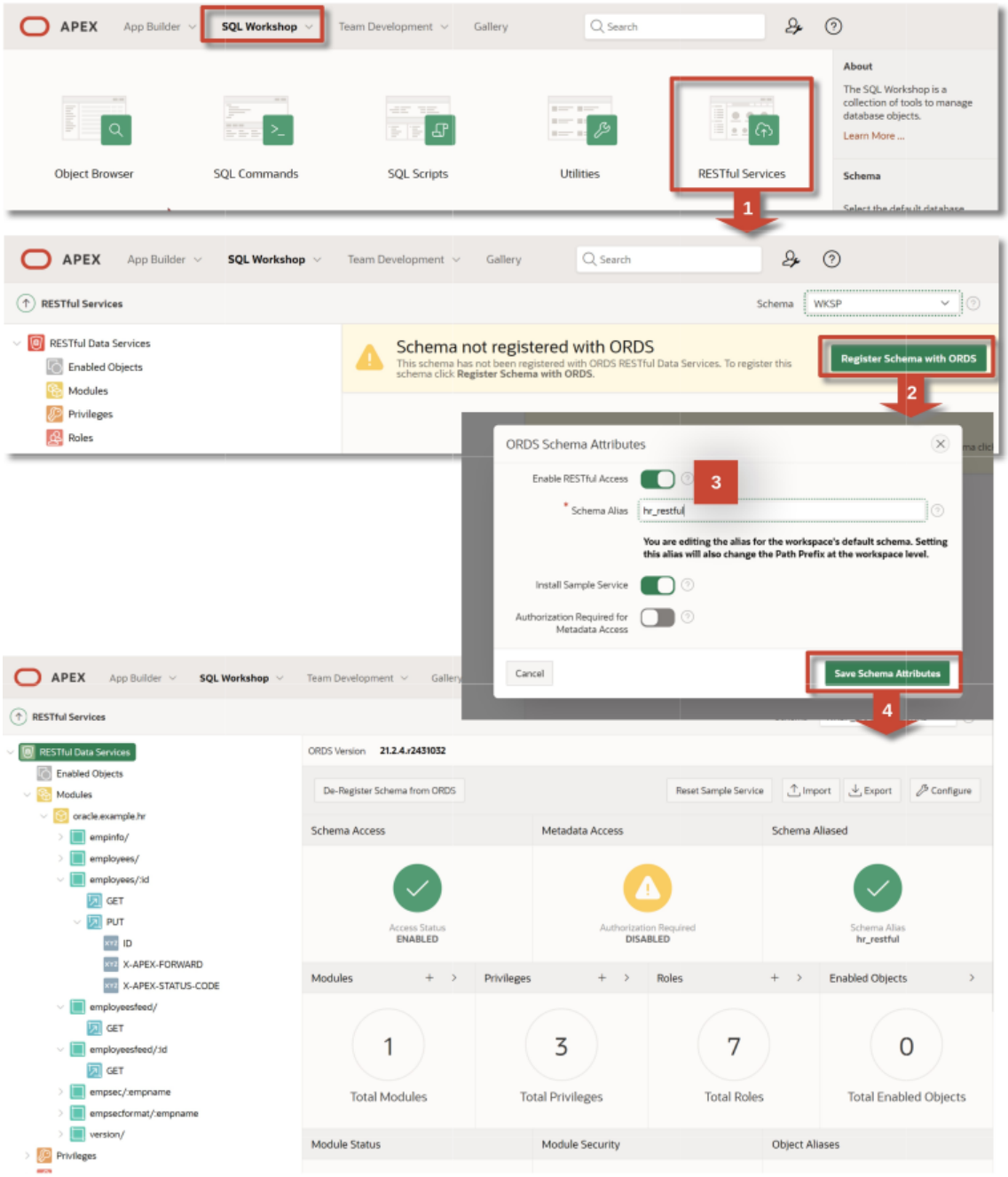
1. **DB-tabuľky poskytnuté ako zdroje pre služby**. Služba RESTful je modelovaná ako množina zdrojov, v našom prípade nad vašimi DB-tabuľkami. Tieto zdroje sú identifikované URL a prístupné cez webové protokoly `HTTP` alebo `HTTPS`.
2. **Operácie odoslané prostredníctvom HTTP-Requests**. Na prácu s týmito zdrojmi sa používa malá skupina operácií v zmysle `HTTP`-požiadavky obsahujúcej `POST`, `GET`, `PUT`, `DELETE`, čím sa podobá operáciam CRUD (Vytvoriť, Prečítať, Aktualizovať, Odstrániť).
3. **Služby sú bezstavové**. Služby RESTful sú *bezstavové*, t.j. medzi požiadavkami nie je na serveri uložený žiadny kontext klienta – stav relácie je ponechaný výlučne na klientovi. To znamená, že každá požiadavka od klienta k serveru musí obsahovať všetky informácie potrebné na to, aby server porozumel požiadavke a nemôže využiť žiadny uložený kontext na serveri. Hlavnou výhodou je, že je možné vytvárať opätovne použiteľné a škálovateľné služby, ktoré možno spravovať a aktualizovať bez ovplyvnenia systému, dokonca aj počas

behu. Nevýhodou je, že môže znížiť výkon siete zvýšením počtu opakujúcich sa údajov (režie pre jednotlivé interakcie) odosielaných v sérii požiadaviek, pretože tieto údaje nemožno ponechať na serveri v zdieľanom kontexte.

4. **HTTP-Responses**. Požiadavky na RESTful Service vždy vyvolajú *response - odpoveď*. Táto odpoveď je vo forme XML, JSON, HTML alebo iného definovaného formátu. Odpovede poskytujú podrobnosti o nejakom druhu zmien základných údajov, chybových hláseniach a hypertextových odkazoch na iné súvisiace zdroje v závislosti od operácie.

4.5.2 Povolenie DB-schémy pre RESTful prístup

Aby bolo možné povoliť DB-schému pre RESTful prístup, musí byť najprv zaregistrovaná v ORDS . Dá sa to jednoducho urobiť, ako je znázornené na Obrázku 4.13 tak, že vstúpite do nástroja RESTful services v rámci SQL Workshopu (1), potom kliknete na tlačidlo "Register Schema with ORDS" (2), a zadáte ľubovoľný alias použitý pri vytváraní adresy URL odkazujúcej na akúkoľvek službu RESTful v rámci schémy (3) a nakoniec sa po kliknutí na tlačidlo *"Save Schema Attributes"* (4) zobrazí panel ORDS RESTful Services Dashboard, ktorý sa používa na správu všetkých RESTful služieb.



Obr. 4.13: Povolenie DB-schémy pre RESTful prístup.

V prípade, že v kroku (3) bola povolená možnosť "Install Sample Service", nainštaluje sa aj vzorový *modul* `oracle.example.hr` s príkladmi *resource templates* a *resource handlers* (pozri Sekciu 4.5.3) , ktoré implementujú niekoľko rôznych operácií na získanie a zobrazenie informácií o zamestnancoch z tabuľky Employees ako je možné vidieť na Obrázku 4.15.

4.5.3 Moduly Zdrojov | Templates | Handlers

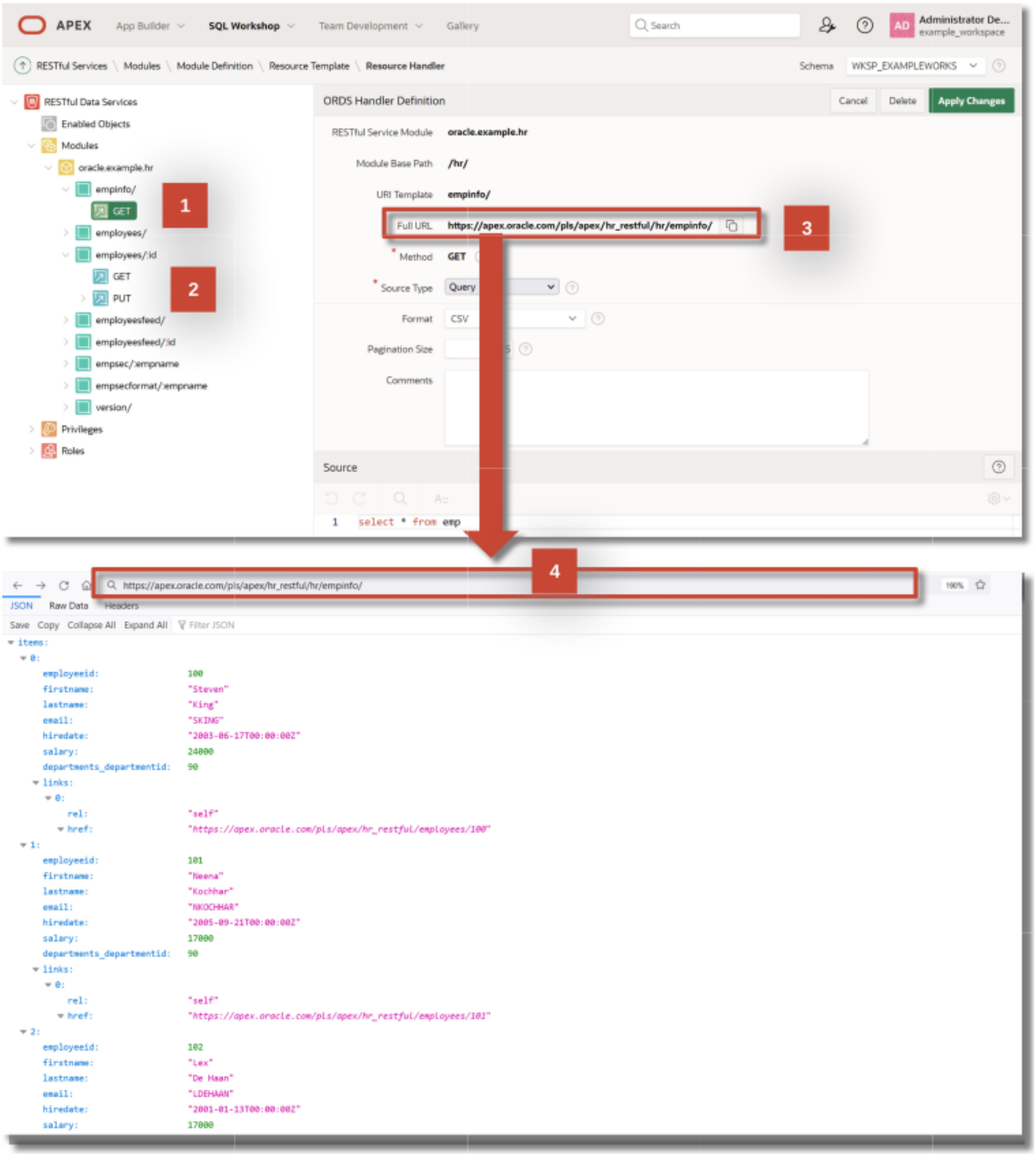
V ORDS môžu byť služby RESTful vytvorené pomocou troch základných komponentov, ktoré zahŕňajú moduly zdrojov, šablóny zdrojov a obslužné programy zdrojov (niekoľko príkladov pozri na Obrázku 4.15). Celkové vzťahy medzi týmito základnými stavebnými blokmi sú znázornené ER-diagramom na Obrázku 4.14.



Obr. 4.14: Vzťahy medzi rôznymi komponentmi ORDS RESTful Services.

- **Resources Modules.** Zdrojový modul nie je nič iné ako kontajner, ktorý zoskupuje množinu súvisiacich RESTful služieb na základe určitej *REST-umožňujúcej DB-Schémy*. DB Schéma môže obsahovať niekoľko zdrojových modulov. Zdrojový modul poskytuje nielen spôsob, ako jednoznačne identifikovať skupinu, ale tiež definuje jedinečnú základnú cestu použitú na URI *na prístup k množine služieb* definovanej v module. Napríklad modul zdrojov, ktorý vám umožňuje prístup k informáciám o zamestnancoch, má názov `oracle.example.hr` a základná hodnota cesty pre službu je `/hr/`, pozri Obrázok 4.15.
- **Resource Templates**. Šablóna zdrojov definuje individuálnu službu, ktorú možno volať, napr. `employees/:id`, Obrázok 4.15. Šablóny zdrojov sú obsiahnuté v moduloch zdrojov. Každá šablóna zdroja definuje *vzor URI*, kde sa dá dosiahnuť, napr. "empinfo/" (Obrázok 4.15) a obsahuje aspoň jeden obslužný program zdroja (pozri nižšie).
- **Resource Handlers.** Každý Resource Handler implementuje jednu (a iba jednu) zo štyroch rôznych HTTP operácií, ktoré sa podobajú tradičným CRUD operáciám (Create, Read, Update, Delete) .
  - `POST` vytvorí nový zdroj alebo pridá zdroj do kolekcie, ekvivalentný príkazu SQL `INSERT`.
  - `GET` získa reprezentáciu zdroja, ekvivalentnú príkazu SQL `SELECT`.
  - `PUT` aktualizuje hodnoty existujúceho zdroja, ekvivalentné príkazu SQL `UPDATE`.
  - `DELETE` vymaže existujúci zdroj, ekvivalentný príkazu SQL `DELETE`.

Pre každú operáciu spojenú s rovnakou šablónou prostriedkov musíte definovať obslužný program zdroja. Ak napríklad chcete poskytnúť *operáciu na vrátenie údajov* a *ďalšiu operáciu na ukladanie údajov*, musíte pre každú operáciu definovať obslužný program prostriedkov (pozri Obr. 4.15 – pre Resource Template `employees/:id`. boli definované dve operácie `GET` (1) a `PUT` (2)).



Obr. 4.15: Príklad RESTful služieb pre tabuľku Employees.

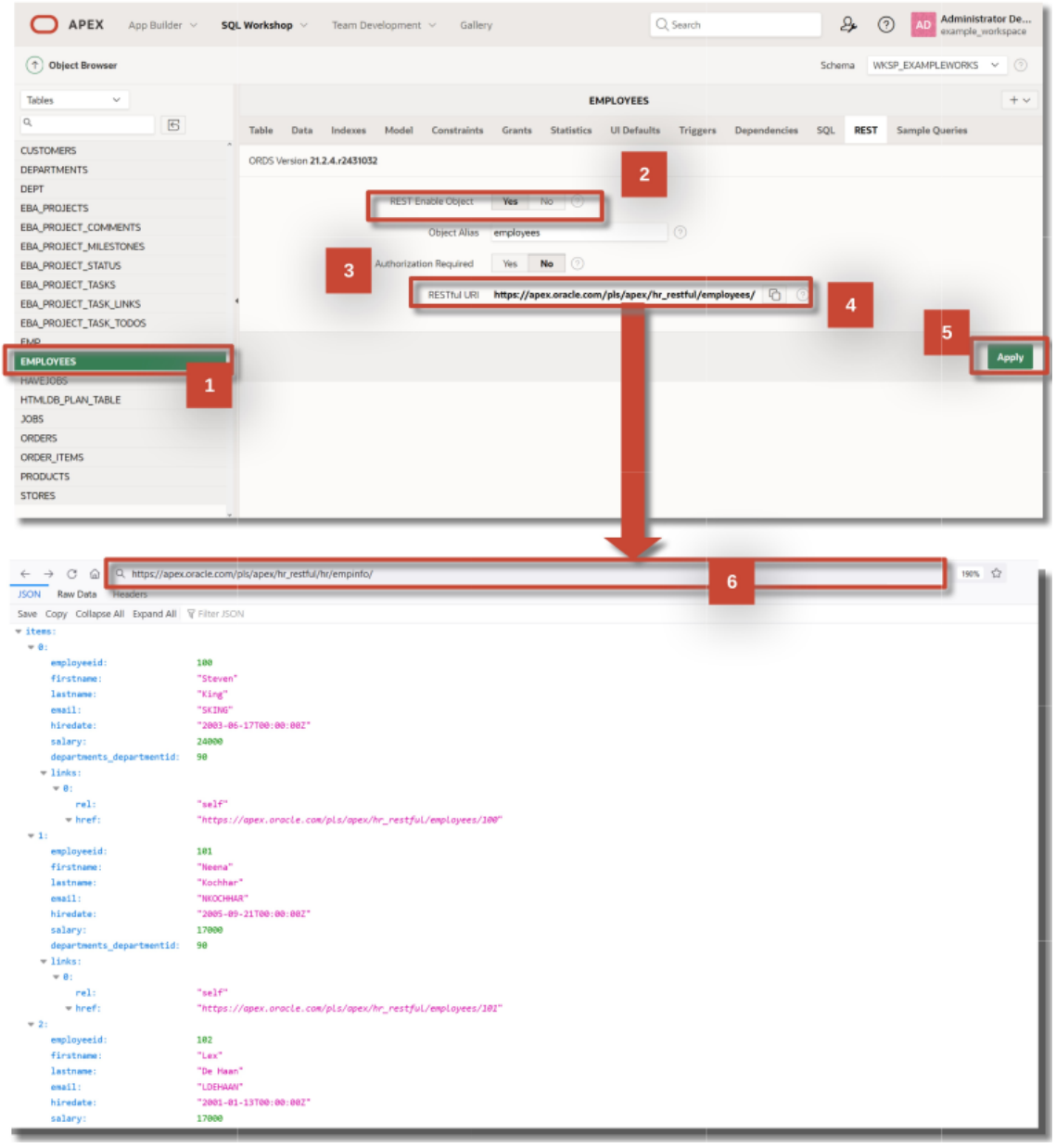
Zhrnutie, Obrázok 4.15, poskytuje príkladný prehľad rôznych zložiek RESTful služby, pokiaľ ide o Resource Modules, Templates a Handlers, vrátane výberu požadovaného formátu odpovede (CSV) a SQL dotaz implementujúci typ operácie Resource Handler. Po vytvorení RESTful Service ju môžete otestovať jednoduchým zadaním vygenerovanej prístupovej URL do vášho prehliadača (pozri Obr. 4.15 (3) a (4) v tomto poradí).

4.5.4 Používanie “AutoREST” namiesto manuálneho definovania zdrojov

Ako bolo vidieť v predchádzajúcej časti, pre každý druh DB operácie, ktorá by mala byť ponúkaná externým klientom ako RESTful služba na nepriamy prístup k našej DB, je potrebných niekoľko krokov. Pre najjednoduchšiu formu dotazu, t. j. úplné skenovanie tabuľky, však existuje aj určitá “skratka”, ktorú poskytuje APEX v rámci takzvanej *“AutoREST”-možnosti*. AutoREST je rýchly a jednoduchý spôsob, ako vystaviť DB tabuľky ako prostriedky REST. Ak použijete funkciu

AutoREST, stratíte určitú flexibilitu a prispôsobiteľnosť, ale výrazne zníži váš čas a úsilie. AutoRest vám umožňuje rýchlo odhaliť údaje, ale umožňuje iba dotazovať sa na celú tabuľku (t. j. skenovanie celej tabuľky), pričom má pevný výstupný formát iba JSON.

AutoREST sa dá jednoducho použiť nasledujúcim spôsobom: Po povolení vašej schémy pre RESTful prístup (pozri Časť 4.5.2) je možné každú DB tabuľku individuálne definovať ako zdroj pomocou Object Browser SQL Workshopu , ako je možné vidieť na Obrázku 4.16.



Obr. 4.16: Povolenie služby REST v prehliadači objektov.

Po zvolení príslušnej DB tabuľky (1), kliknutí na záložku REST (2), zvolení požadovanej autorizácie (3) a použití všetkých týchto nastavení (5) je pre tabuľku aktivovaný REST a automaticky vygenerovaná prístupová URL sa objaví (4). Túto službu REST možno opäť jednoducho otestovať zadaním adresy URL do webového prehliadača (6).

4.6 Otázky

1. Aké sú prípady použitia importu údajov prostredníctvom programu SQL Workshop a App Builder?
2. Aké sú rôzne spôsoby exportu údajov?
3. Na čo slúži REST ? Aký je jeho účel alebo prínos?

4.7 Odpovede

1. Data Workshop umožňuje importovať údaje z externých súborov prostredníctvom "Sprievodcu" a importovať údaje do tabuľky prostredníctvom "Prehliadača objektov". Import údajov z Data Workshop možno iniciovať aj prostredníctvom aplikácie App Builder.
2. Exportovanie údajov do súboru podľa tabuliek je podporované nástrojom SQL Workshop "Object Browser". Data Workshop poskytuje "Sprievodcu exportom", ktorý umožňuje vybrať formát súboru a tabuľky, ako aj príslušné stĺpce, ktorých údaje sa majú exportovať.
3. Služby RESTful poskytujú súčinnosť medzi ľubovoľnými systémami cez internet tým, že umožňujú vyhľadávať údaje a manipulovať s nimi.



5. Ako vygenerovať prvý návrh aplikácie?

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

5.1 Prečo podniky potrebujú aplikácie

V dnešnom rýchlo sa rozvíjajúcom podnikateľskom prostredí sa aplikácie stali pre spoločnosti kľúčovým nástrojom na udržanie konkurencieschopnosti a splnenie neustále sa meniacich požiadaviek zákazníkov. Vytváranie aplikácií umožňuje podnikom zefektívniť svoju činnosť, zvýšiť efektivitu a zlepšiť skúsenosti zákazníkov, čo v konečnom dôsledku môže viesť k vyššej ziskovosti. Jedným z hlavných dôvodov, prečo podniky budujú aplikácie, je automatizácia procesov, ktoré sa v súčasnosti vykonávajú manuálne. Napríklad spoločnosť zaoberajúca sa elektronickým obchodom môže vyvinúť aplikáciu, ktorá automatizuje procesy riadenia zásob a plnenia objednávok. Týmto spôsobom môžu znížiť pravdepodobnosť ľudskej chyby a zvýšiť rýchlosť a presnosť svojich operácií.

Ďalším kľúčovým dôvodom, prečo podniky vytvárajú aplikácie, je zlepšenie zákazníckej skúsenosti. Aplikácie môžu byť vyvinuté tak, aby zákazníkom poskytovali samoobslužné možnosti, napríklad možnosť sledovať svoje objednávky alebo aktualizovať informácie o svojom účte. To môže pomôcť skrátiť čas čakania zákazníkov a zvýšiť ich spokojnosť.

Okrem toho môžu aplikácie pomôcť podnikom zhromažďovať cenné údaje o svojich zákazníkoch, ktoré sa dajú využiť na zlepšenie marketingových stratégií a vývoj cielených reklamných kampaní. Napríklad maloobchodný predajca môže vyvinúť aplikáciu, ktorá sleduje nákupné zvyklosti zákazníkov, čo mu umožní posielať zákazníkom personalizované odporúčania a propagačné akcie.

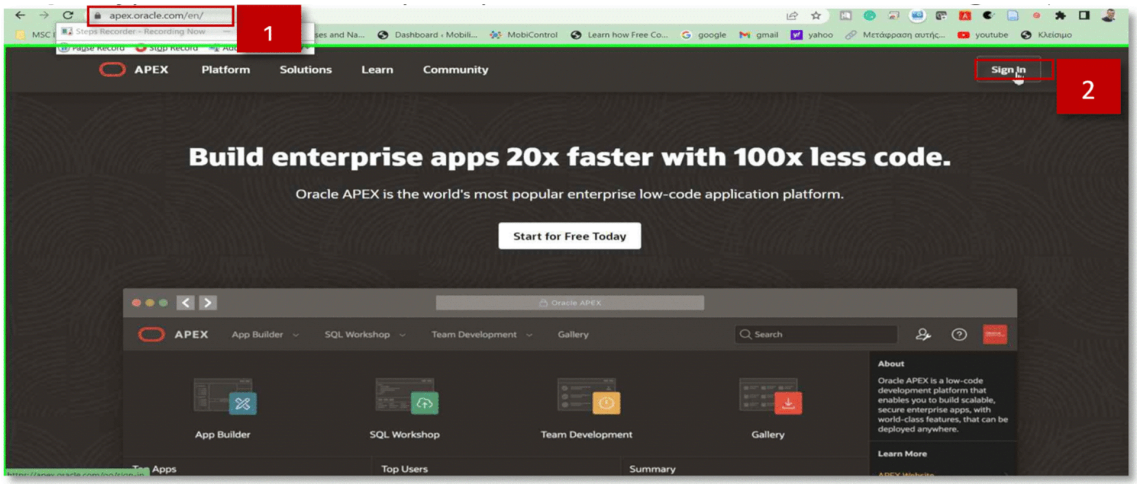
Celkovo sa vytváranie aplikácií stalo silnou obchodnou potrebou, pretože umožňuje spoločnostiam fungovať efektívnejšie, poskytovať lepšie služby zákazníkom a získať prehľad o správaní svojich zákazníkov. Keďže svet podnikania sa naďalej stáva digitálnejším, význam budovania aplikácií bude naďalej rásť.

5.1.1 Nastavenie prostredia ORACLE APEX

Účelom tohto príkladu je predstaviť a oboznámiť vás s prostredím ORACLE APEX, ktoré obsahuje kroky na nastavenie prostredia APEX a príklad SQL workshopu s databázou pracovníkov projektu.

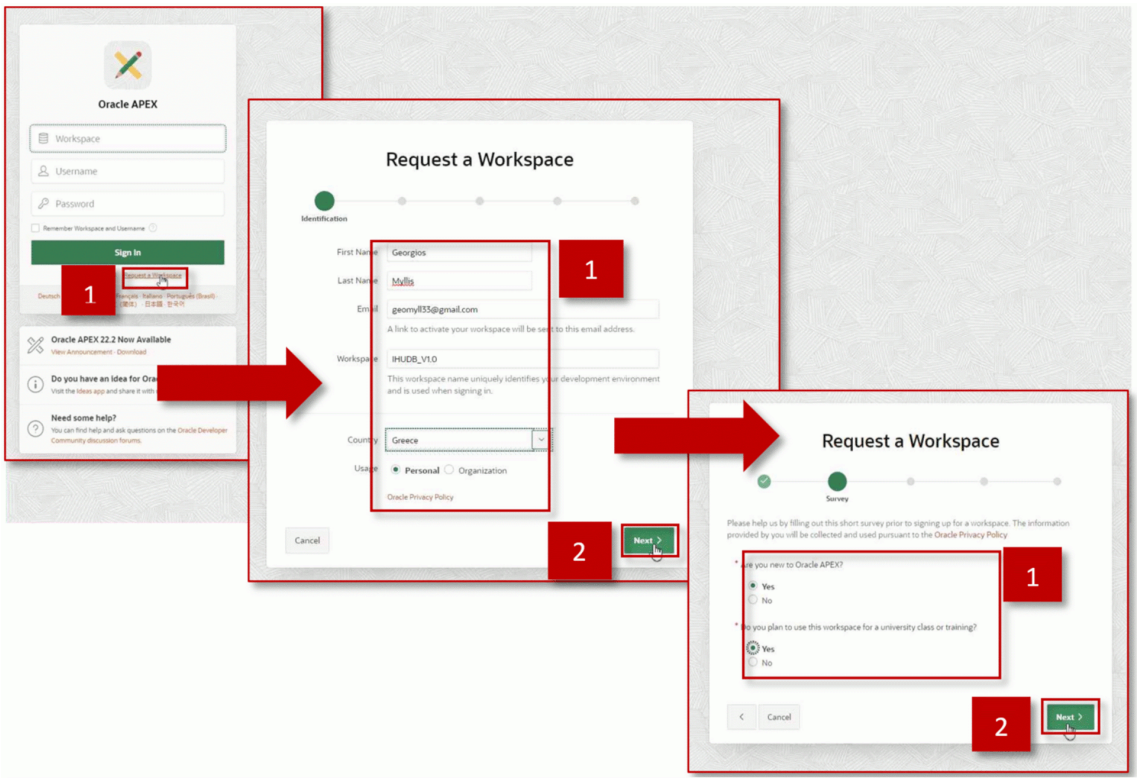
Na implementáciu tohto príkladu budú dodržané nasledujúce kroky:

Krok 1: do prehliadača zadajte “https://apex.oracle.com/en/” a kliknite na **Sign in** na Obrázku 5.1.



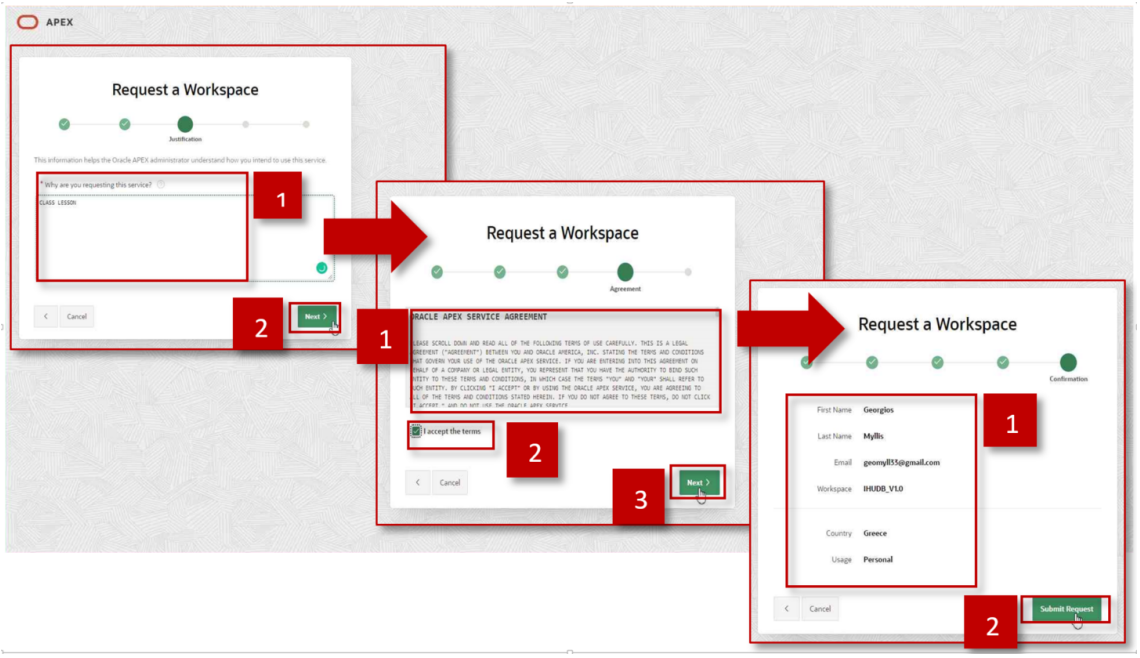
Obr. 5.1: Zadajte APEX odkaz a prihláste sa.

Krok 2: Kliknite na **Request a Workspace**, v ďalšom vyskakovacom okne vyplňte požadované polia na identifikáciu a kliknite na **Next**, v ďalšom vyskakovacom okne odpovedze na informácie požadované pre prieskum a kliknite na **Next** na Obrázku 5.2.



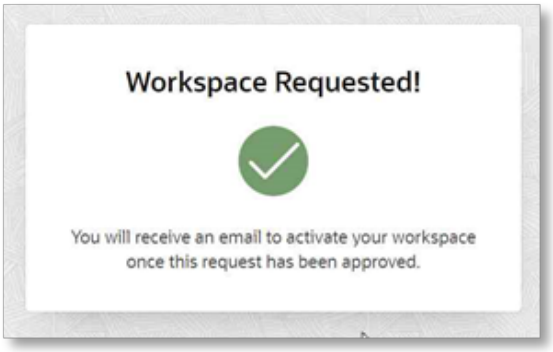
Obr. 5.2: Úvodné kroky žiadosti o Oracle APEX Workspace.

Krok 3: Odpovedzte na informácie požadované pre oprávnenie a zvoľte **Next**, v ďalšom vyskakovacom okne si prečítajte podmienky zmluvy a potom, aby ste mohli pokračovať, akceptujte podmienky zmluvy a zvoľte **Next**, v ďalšom vyskakovacom okne skontrolujte, či v zadaných údajoch nie sú nejaké chyby a potom kliknite na **Submit request** na Obrázku 5.3.



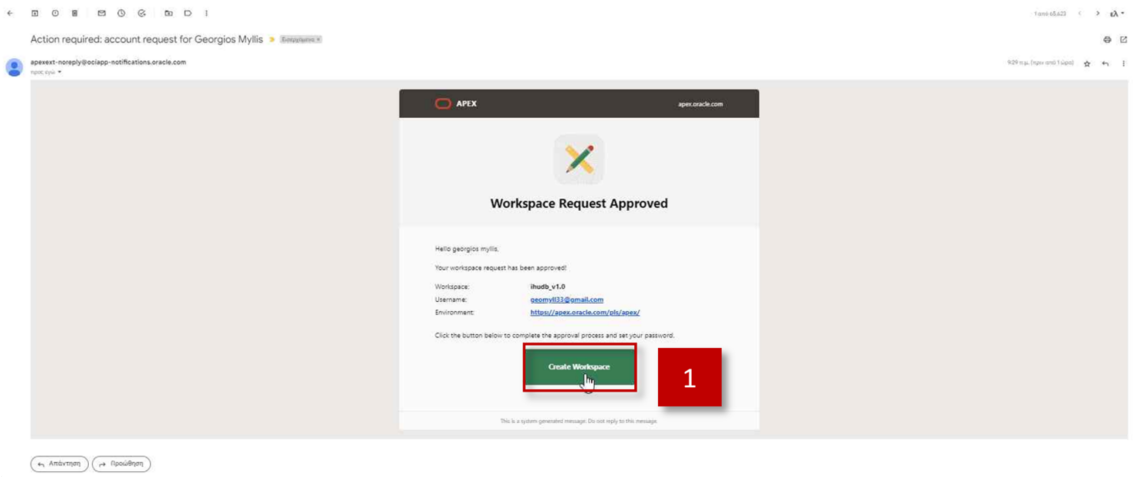
Obr. 5.3: Dokončenie žiadosti o Oracle APEX Workspace.

Krok 4: Vyskakovacie okno vás bude informovať, že váš pracovný priestor bol vyžiadaný a aktivačný e-mail bude odoslaný na e-mailový účet, ktorý ste uviedli na Obrázku 5.4.



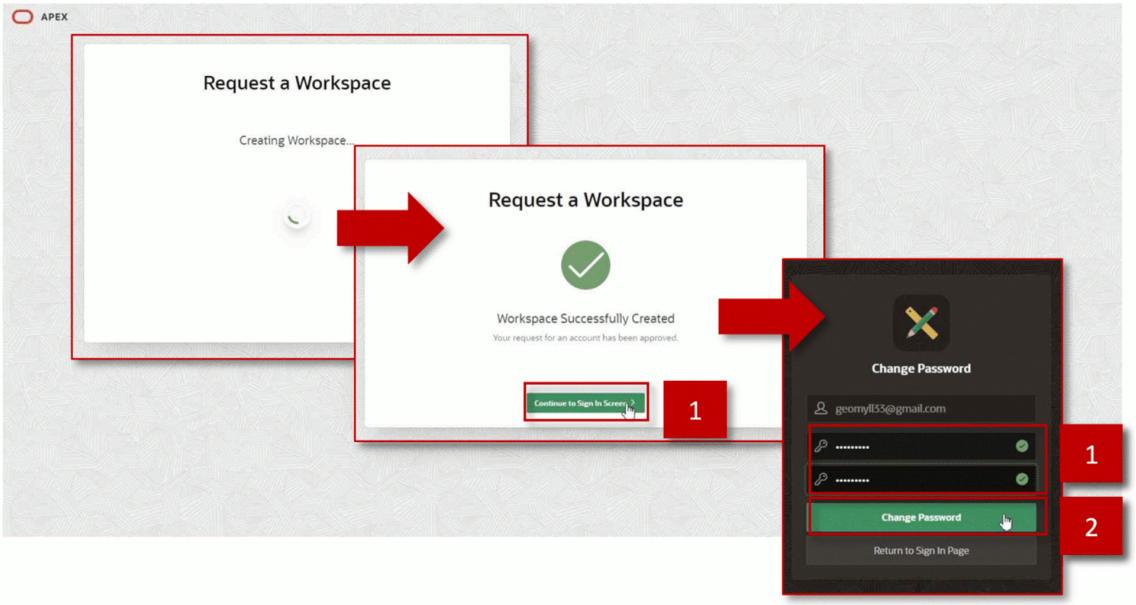
Obr. 5.4: Schválený Oracle APEX Workspace.

Krok 5: Vyžaduje sa akcia, skontrolujte svoj e-mail pre príjem z apexu, vaša žiadosť musí byť schválená, kliknite na **Create workspace** na Obrázku 5.5.



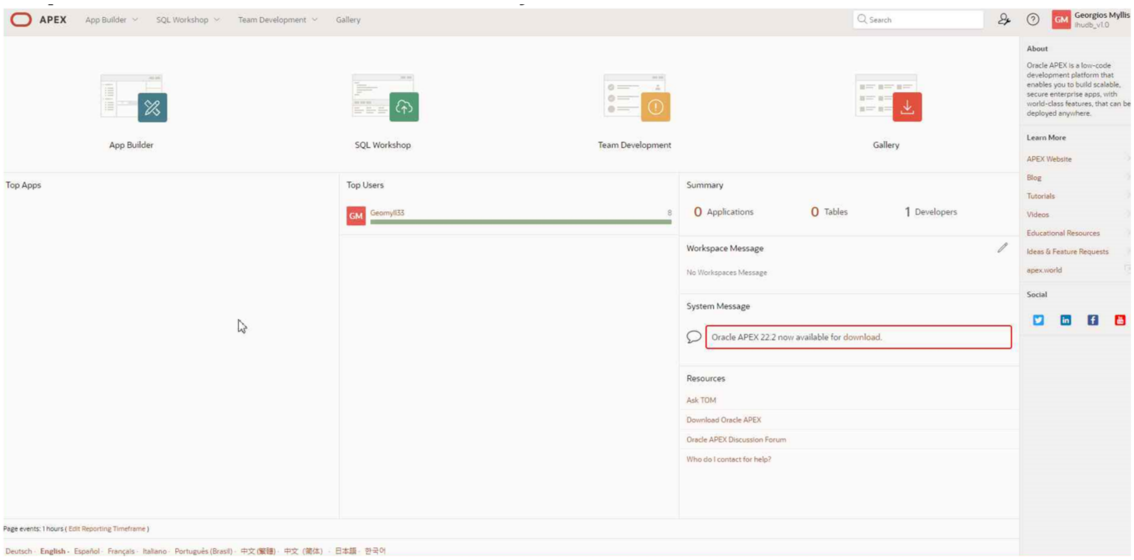
Obr. 5.5: Aktivačný e-mail od APEX.

Krok 6: Počkajte, kým sa dokončí vytváranie Oracle APEX Workspace, po jeho dokončení vás vyskakovacie okno informuje, že Oracle APEX Workspace bol úspešne vytvorený, kliknite na **Continue to Sign in Screen**. Vyskakovacie okno bude vyžadovať zmenu hesla pred prihlásením užívateľa, je potrebné nastaviť nové heslo, potvrdiť ho a kliknúť na **Change Password** na Obrázku 5.6.



Obr. 5.6: Nastavenie nového hesla pre Workspace APEX.

Krok 7: Prostredie Oracle APEX je pripravené na použitie na Obrázku 5.7.



Obr. 5.7: Prostredie Workspace APEX.

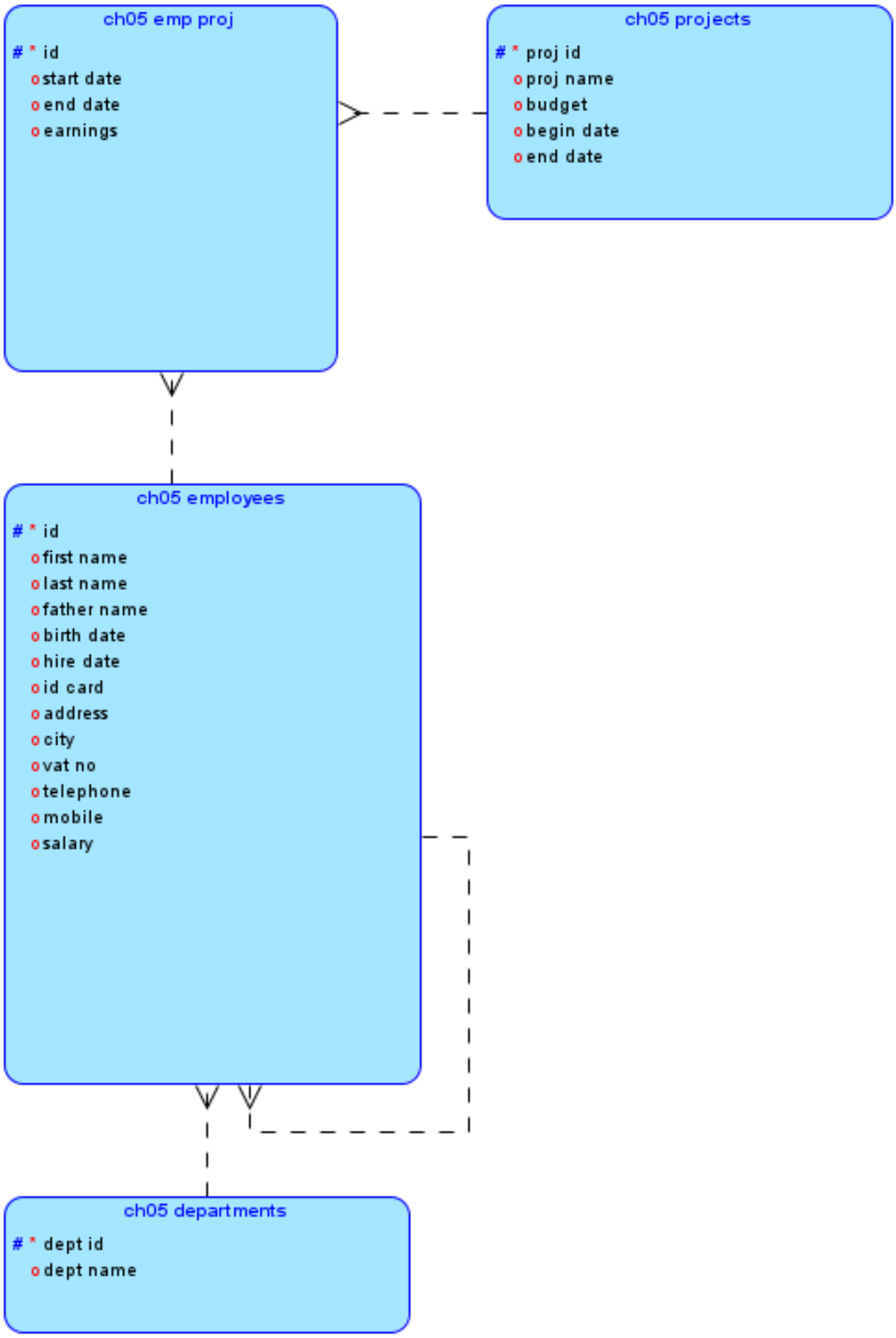
5.1.2 Príklad vývoja SQL workshopu

Na vytvorenie databázy v prostredí APEX je k dispozícii viacero prístupov. Jednou z možností je využiť Object Browser (Prehliadač objektov), grafický nástroj, ktorý umožňuje používateľom vytvárať tabuľky a ich príslušné polia vyplnením vopred navrhnutej šablóny.

Ďalšou alternatívou je použitie príkazov SQL, ktoré umožňujú používateľom vytvárať tabuľky a iné databázové objekty vykonávaním príkazov SQL. Táto metóda ponúka väčšiu flexibilitu a kontrolu nad štruktúrou databázy, hoci si vyžaduje pokročilejšie znalosti.

Na druhej strane, Quick SQL je nástroj poskytovaný systémom Oracle APEX, ktorý uľahčuje rýchle generovanie skriptov SQL na vytváranie databázových tabuliek, obmedzení a vzorových údajov. Quick SQL zjednodušuje proces vytvárania databázových objektov bez potreby ručného písania zložitých SQL skriptov.

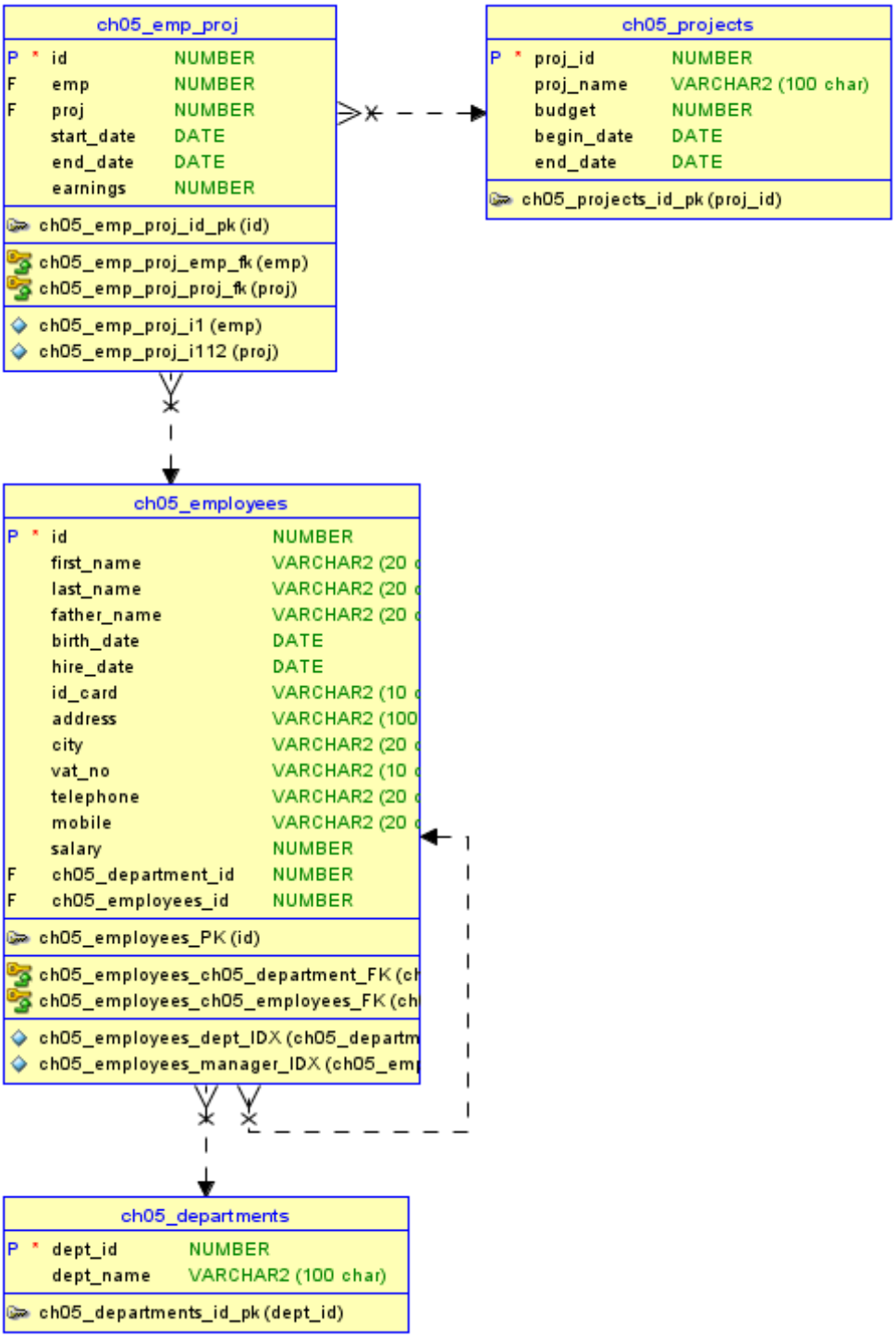
V tomto príklade použijeme metódu Quick SQL. Skript bol vytvorený na základe logického modelu zobrazeného na Obrázku 5.8.



Obr. 5.8: Logický model nášho HR príkladu.

Podľa relačného modelu odvodeného z logického modelu pomocou programu Oracle Data Mo-

deler (ODM), ako je znázornené na Obrázku 5.9, môžeme pokračovať v implementácii nasledujúcej štruktúry v databáze.



Obr. 5.9: Relačný model nášho HR príkladu.

5.1.3 Quick SQL

Poskytnutý kód sa javí ako súbor pokynov na generovanie vzorových údajov pomocou nástroja Quick SQL. Každá časť začína názvom tabuľky, za ktorým nasleduje kľúčové slovo “/insert” a

číslo označujúce požadovaný počet riadkov, ktoré sa majú do tabuľky vložiť.

```
ch05_departments /insert 5
dept_name vc100

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

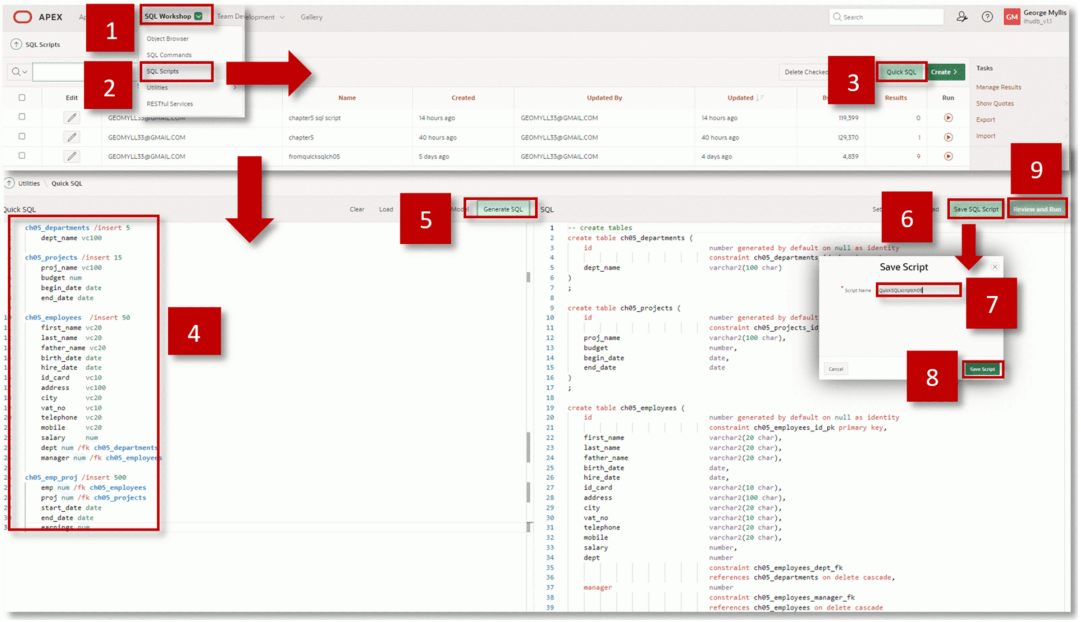
ch05_employees  /insert 50
first_name vc20
last_name  vc20
father_name vc20
birth_date date
hire_date  date
id_card    vc10
address    vc100
city       vc20
vat_no     vc10
telephone  vc20
mobile     vc20
salary     num
dept num /fk ch05_departments
manager num /fk ch05_employees

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

Tieto pokyny predstavujú schému a vzorový plán generovania údajov pre databázu týkajúcu sa oddelení, projektov, zamestnancov a ich asociácií.

Krok 1: Ako je znázornené na Obrázku 5.10, postupujte podľa nasledujúcich krokov:

1. Prejdite do časti **SQL Workshop**.
2. Z možností vyberte položku **SQL Scripts**.
3. Kliknite na **Quick SQL**.
4. Vložte poskytnutý kód Quick SQL do určenej oblasti.
5. Kliknite na **Generate SQL**. Vygenerovaný SQL sa zobrazí v paneli Oracle SQL Output.
6. Kliknite na **Save SQL Script**.
7. Zadajte názov skriptu.
8. Kliknite na **Save script**.
9. Pokračujte kliknutím na **Review and Run**.



Obr. 5.10: Vloženie kódu Quick SQL do pracovného priestoru APEX.

5.1.4 SQL Script

Ako už bolo spomenuté, nižšie je uvedený SQL skript vygenerovaný z programu Quick SQL, ktorý vytvára databázovú schému pre príklad systému riadenia ľudských zdrojov. Schéma zahŕňa tabuľky pre oddelenia, zamestnancov, projekty a vzťahy medzi zamestnancami a projektmi. Okrem toho obsahuje indexy a obmedzenia na zaručenie konzistencie údajov a referenčnej integrity.

```
-- 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);
```

V tabuľke ch05_departments sú uložené informácie o oddeleniach vrátane jedinečného identifikátora (id) a názvu (dept_name). V tabuľke ch05_employees sa uchovávajú informácie o zamestnancoch vrátane jedinečného identifikátora (id), first_name, last_name, father_name, birth_date, hire_date, ID_card number, address, city, VAT_number, telephone number, mobile number, salary, and department (dept) a manager (manager) ID. V tabuľke ch05_projects sú uložené informácie o projektoch vrátane jedinečného identifikátora (id), name, budget, start_date a end_ date. V tabuľke ch05_emp_proj sa uchovávajú informácie o projektových vzťahoch zamestnancov vrátane jedinečného identifikátora (id), identifikátorov employee (emp) a project (proj), start_date, end_date a earnings.

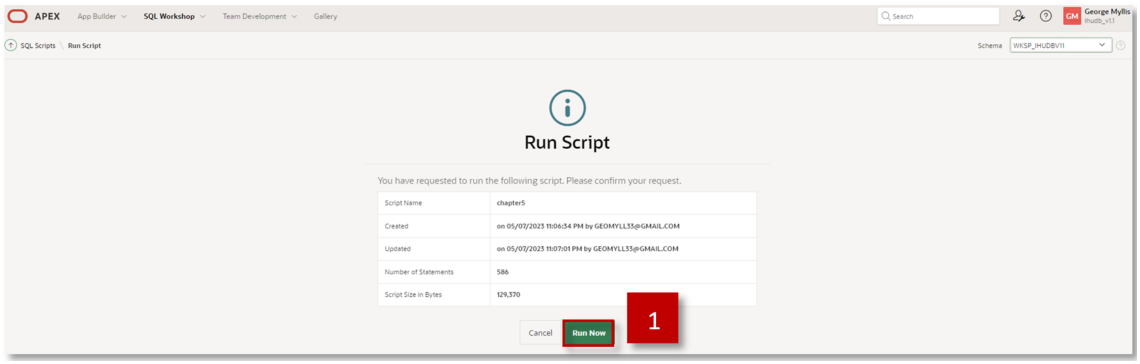
V tabuľke ch05_emp_proj sú vytvorené indexy pre stĺpce employee a project, aby sa zlepšil výkon pri vyhľadávaní v tabuľke na základe týchto stĺpcov. Z rovnakého dôvodu sú vytvorené indexy aj v tabuľke ch05_employee pre stĺpce department a manager.

Nakoniec sa pridajú obmedzenia na zabezpečenie konzistentnosti údajov a referenčnej integrity medzi tabuľkami.

Obmedzenia ch05_emp_proj_emp_fk a ch05_emp_proj_proj_fk zabezpečujú, aby stĺpce employee a project v tabuľke ch05_emp_proj odkazovali na platné hodnoty id v tabuľkách ch05_employees a ch05_project.

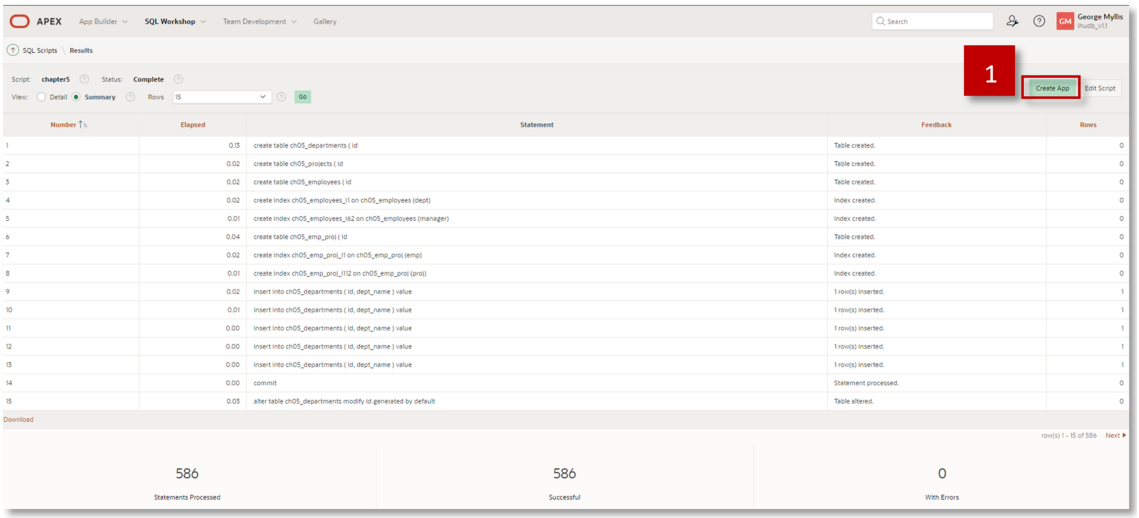
Obmedzenia ch05_employees_dept_fk a ch05_employee_manager_fk zabezpečujú, aby stĺpce departure a manager v tabuľke ch05_employee odkazovali na platné hodnoty id v tabuľkách ch05_departments a ch05_employees.

Krok 2: Ak chcete spustiť požadovanú akciu, pozrite si Obrázok 5.11 a kliknite na tlačidlo **Run Now**.



Obr. 5.11: Spustiť SQL skript.

Krok 3: Po úspešnom vykonaní všetkých príkazov bez výskytu chýb prejdite na Obrázok 5.12 a kliknite na tlačidlo **Create App**.



Obr. 5.12: Proces spustenia aplikácie.

5.2 Dátovo riadená časť aplikácie

Časť aplikácie založená na dátach sa zameriava na využitie sily dát na riadenie poznatkov, rozhodovania a interakcií s používateľmi. V tejto časti sú dáta v stredobodom pozornosti a umožňujú

používateľom skúmať, analyzovať a interagovať s informáciami dynamickým a intuitívnym spôsobom.

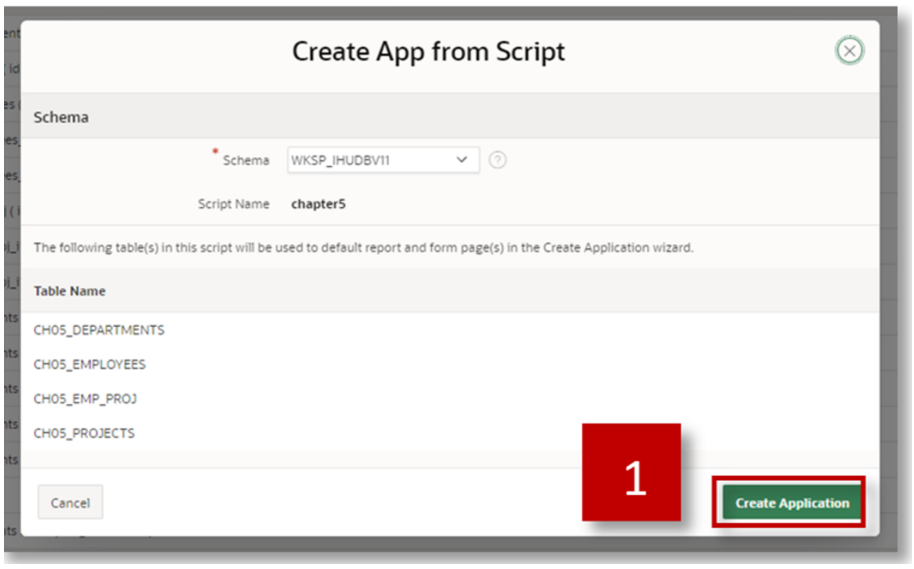
Hlavným cieľom tejto časti je poskytnúť používateľom komplexný pohľad na dáta a umožniť im získať cenné poznatky a prijímať informované rozhodnutia. Prostredníctvom rôznych techník vizualizácie údajov, ako sú grafy, diagramy a interaktívne informačné panely, môžu používatelia ľahko pochopiť komplexné súbory údajov a identifikovať vzory, trendy a odľahlé hodnoty.

Okrem toho časť založená na dátach obsahuje rozsiahle možnosti správy údajov, ktoré umožňujú používateľom filtrovať, triediť a vyhľadávať konkrétne dátové body alebo segmenty. Pokročilé funkcie, ako sú agregácia údajov, rozbaľovanie a funkcie exportu údajov, ďalej zlepšujú používateľský zážitok a umožňujú hlbšie skúmanie a analýzu.

Táto sekcia tiež zdôrazňuje dôležitosť integrity a kvality údajov. Zahŕňa mechanizmy validácie údajov, spracovania chýb a čistenia dát s cieľom zabezpečiť, aby boli prezentované informácie presné, spoľahlivé a konzistentné.

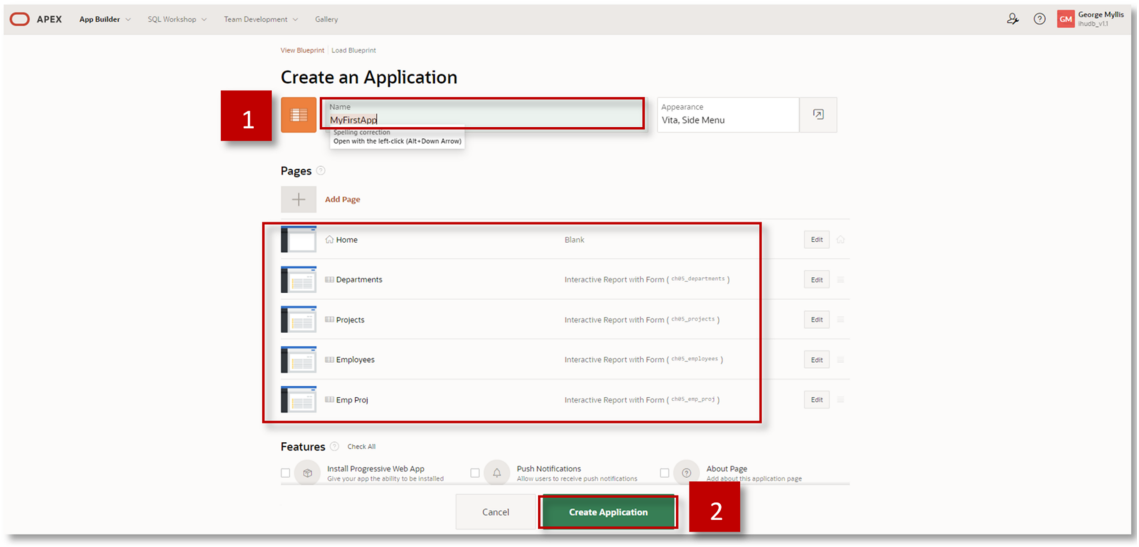
V sekcii založenej na dátach môžu používatelia skúmať údaje z viacerých uhlov pohľadu, prispôsobiť si zobrazenia podľa svojich preferencií a využiť silu údajov na získanie zmysluplných poznatkov. Slúži ako výkonný nástroj na analýzu dát, reportovanie a podporu rozhodovania v rámci aplikácie, čím umožňuje používateľom robiť rozhodnutia založené na dátach a dosahovať priaznivé výsledky.

Krok 1: Keď sa zobrazí vyskakovacie okno s označením **Create App from the Script**, v súlade s Obrázkom 5.13, overte, že tabuľky boli vytvorené a budú použité vo vašej novej aplikácii, kliknite na tlačidlo **Create Application**.



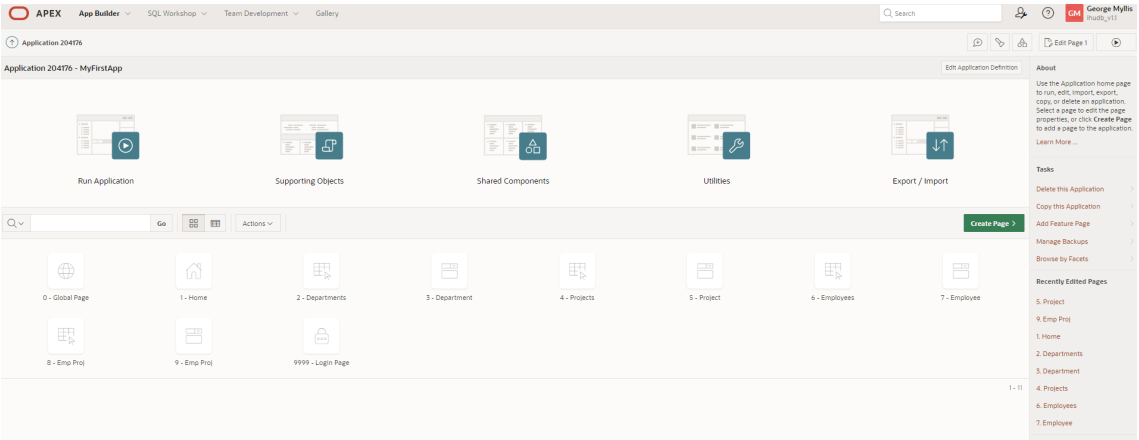
Obr. 5.13: Vytvorenie aplikácie zo skriptu.

Krok 2: Zadajte názov aplikácie, skontrolujte, či boli pridané reporty a stránky formulárov z tabuliek a či budú použité vo vašej novej aplikácii, a potom kliknite na **Create Application**, ako je znázornené na Obrázku 5.14



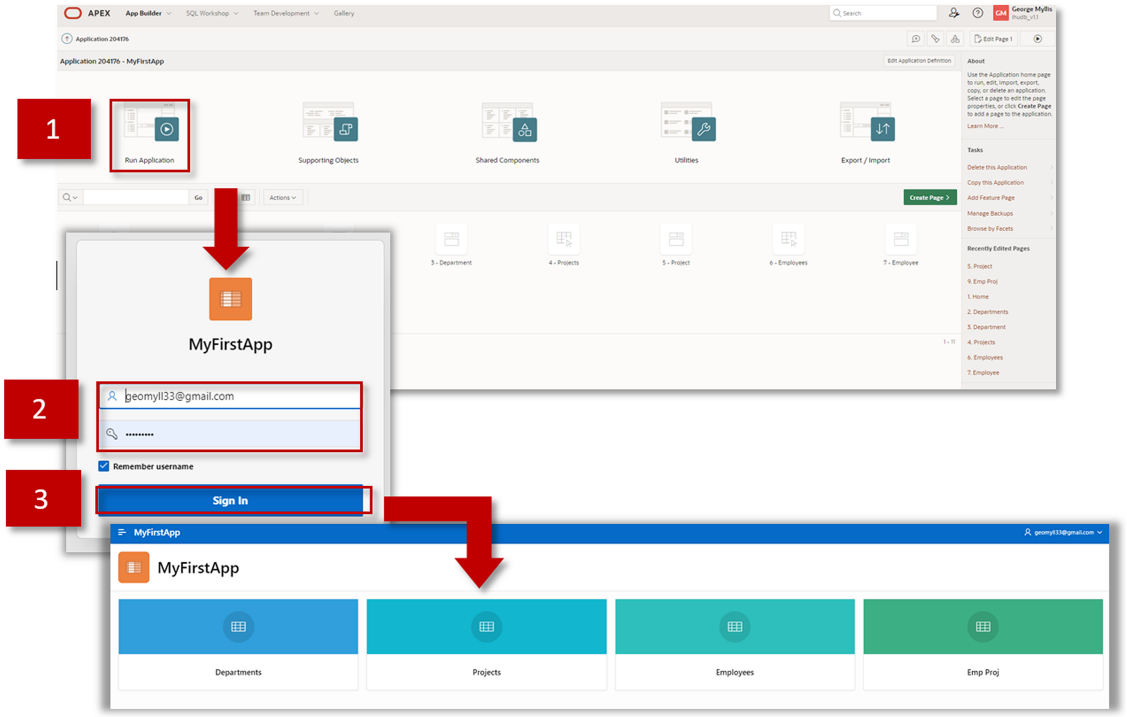
Obr. 5.14: Vytvorená webová aplikácia.

Krok 3: Pre vašu aplikáciu bolo vytvorené prostredie app builder, ako je znázornené na Obrázku 5.15. Je dôležité poznamenať, že v tomto prostredí máte možnosť rozsiahleho prispôsobovania aplikácie. Môžete pridávať, odstraňovať alebo upravovať polia formulára, upravovať rozloženie, definovať pravidlá overovania a vykonávať množstvo ďalších činností, aby ste aplikáciu prispôsobili svojim špecifickým potrebám.



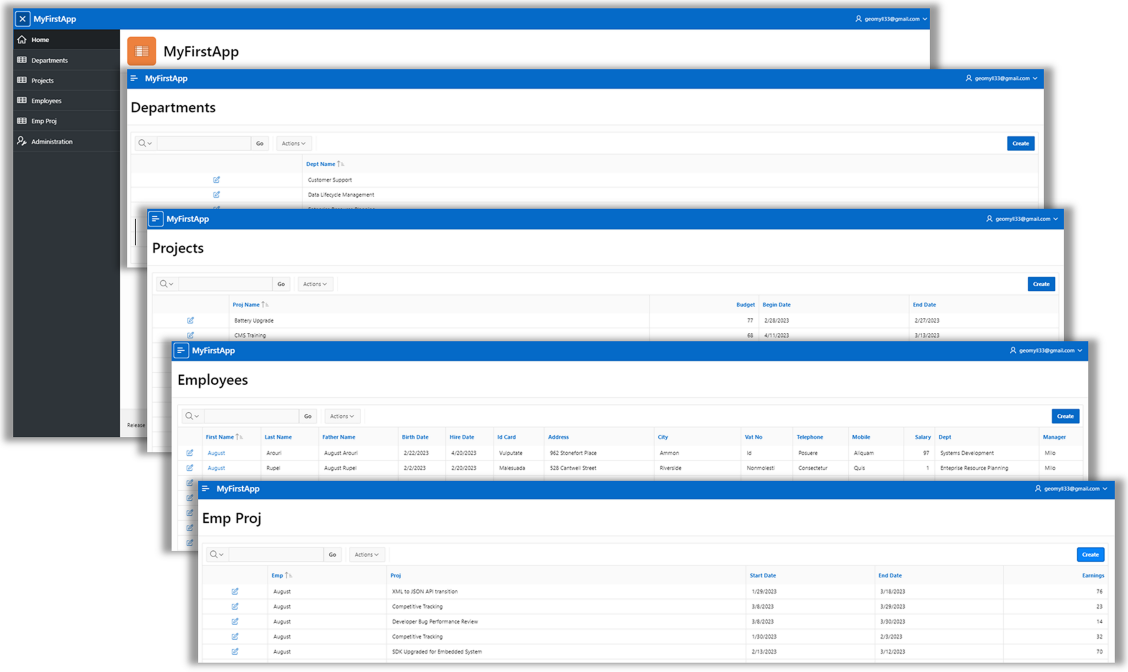
Obr. 5.15: App builder prostredie.

Krok 4: Kliknutím na ikonu **Run Application** v prostredí App builder spustíte aplikáciu, ako je znázornené na Obrázku 5.16. Tým sa otvorí nová stránka prehliadača so špeciálnym odkazom APEX pre vašu aplikáciu. Zadajte svoje prihlasovacie údaje, ktoré sú rovnaké ako vo WORKSPACE, a potom kliknite na tlačidlo **Sign in**. Následne budete presmerovaní na domovskú stránku vašej aplikácie.



Obr. 5.16: Prihlasovacia stránka aplikácie.

Krok 5: V aplikácii sa môžete ľahko pohybovať pomocou bočného menu, ako je znázornené na Obrázku 5.17. Aplikácia poskytuje používateľsky prívetivé prostredie, ktoré vám umožňuje bezproblémovo pridávať, upravovať a prezentovať dáta podľa vašich špecifických potrieb a preferencií.

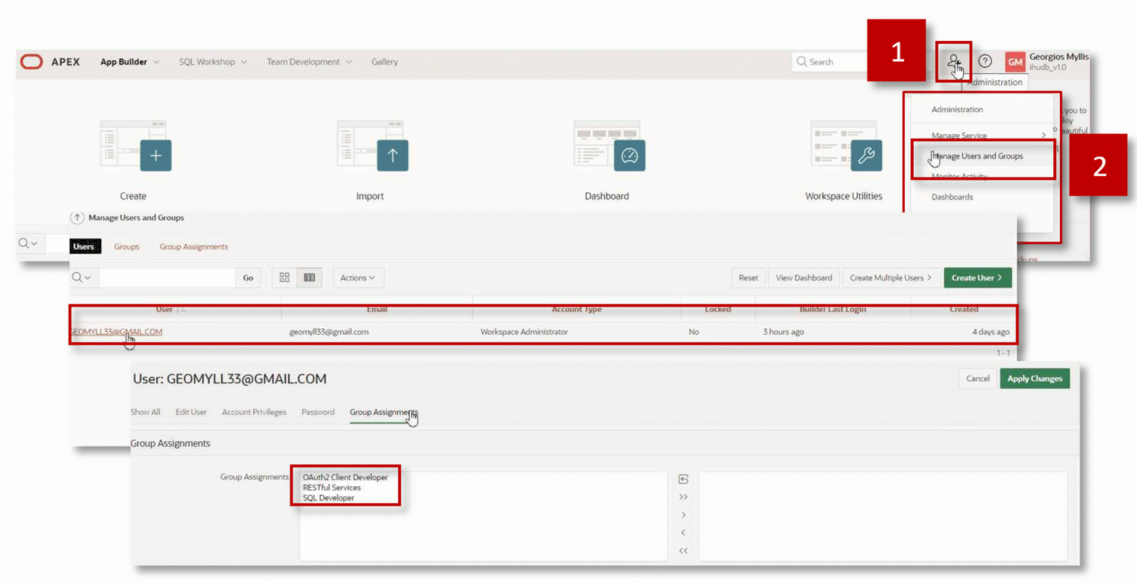


Obr. 5.17: Prostredie vašej novej aplikácie.

5.3 Administrácia aplikácie

Pokiaľ ide o autentifikáciu používateľských účtov, ide o integrovanú metódu autentifikácie v aplikácii Oracle Application Express, ktorá umožňuje vytváranie a správu používateľských účtov v úložisku používateľov aplikácie Application Express. Táto metóda je užitočná najmä vtedy, ak vaša aplikácia používa overovanie Application Express Accounts.

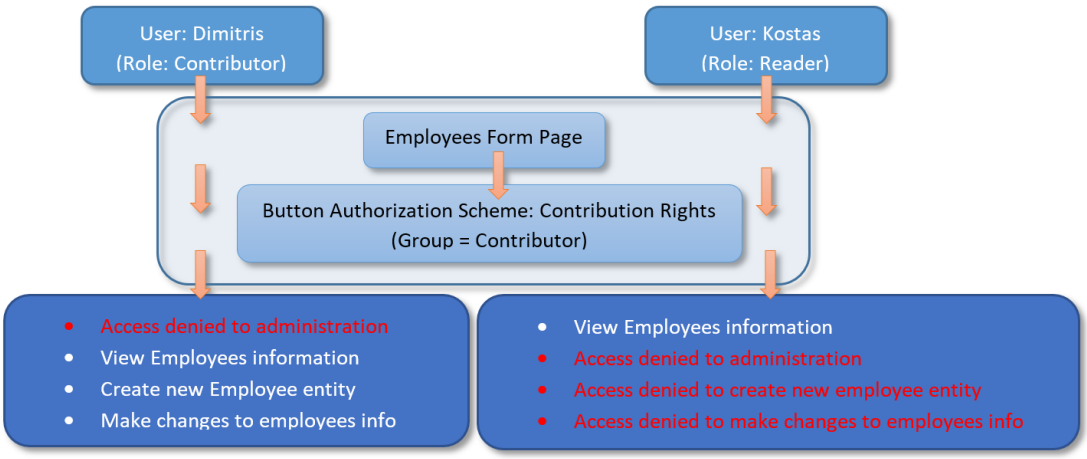
V aplikácii v prostredí Workspace kliknite na **Administration** a vyberte **Manage Users and Groups**. Vytvorené počiatočné konto možno použiť ako administrátora pre zabudovanú aplikáciu. Oracle APEX User udržiava účty pre autentifikačné schémy, ktoré sa autentifikujú na základe mena používateľa a hesla uložených v týchto účtoch.



Obr. 5.18: Prostredie pre správu používateľov a skupín.

5.4 Riadenie prístupu

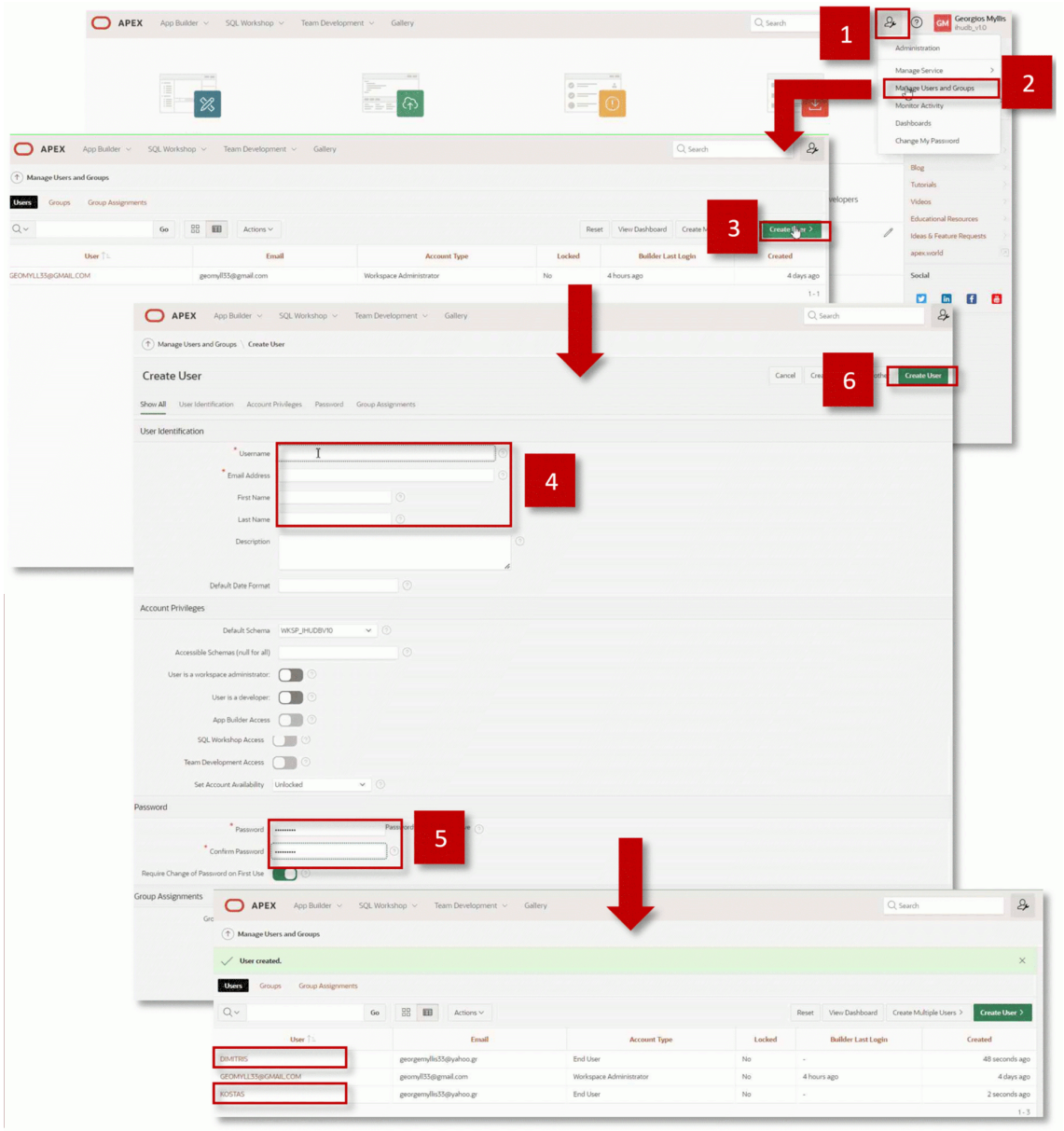
Ďalej, ako je znázornené na Obrázku 5.19, uvedieme príklad riadenia prístupu pre používateľov, ktorí používajú aplikáciu, podľa logiky znázornenej na obrázku nižšie.



Obr. 5.19: Príklad riadenia prístupu.

Oracle APEX poskytuje tri vstavané prístupové práva, ktoré umožňujú spravovať prístup k aplikácii alebo jej súčastiam. Tieto oprávnenia sú administrácia, editácia a zobrazenie a každé z nich zodpovedá prístupovej role. Administrácia zodpovedá role Administrator, editácia zodpovedá role Contributor a zobrazenie zodpovedá role Reader.

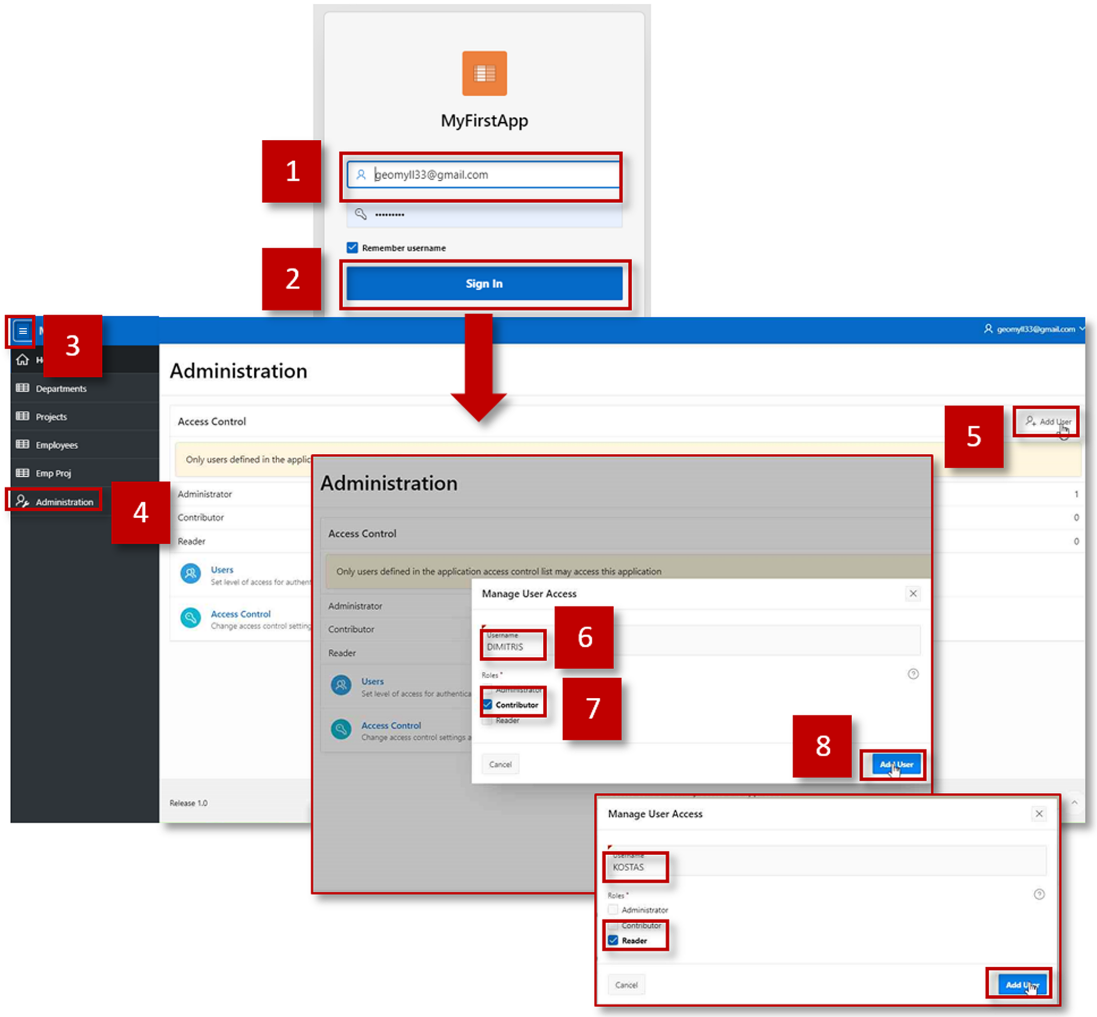
Ak chcete pridať nových používateľov a definovať ich roly, prejdite na **Manage Users and Groups** zo zobrazeného prostredia APEX. Kliknite na **Administration** a vyberte **Manage Users and Groups**. Potom vyberte **Create User**, zadajte identifikačné údaje a heslo používateľa a kliknite na **Create User**, ako je znázornené na Obrázku 5.20.



Obr. 5.20: Pridanie nových používateľov.

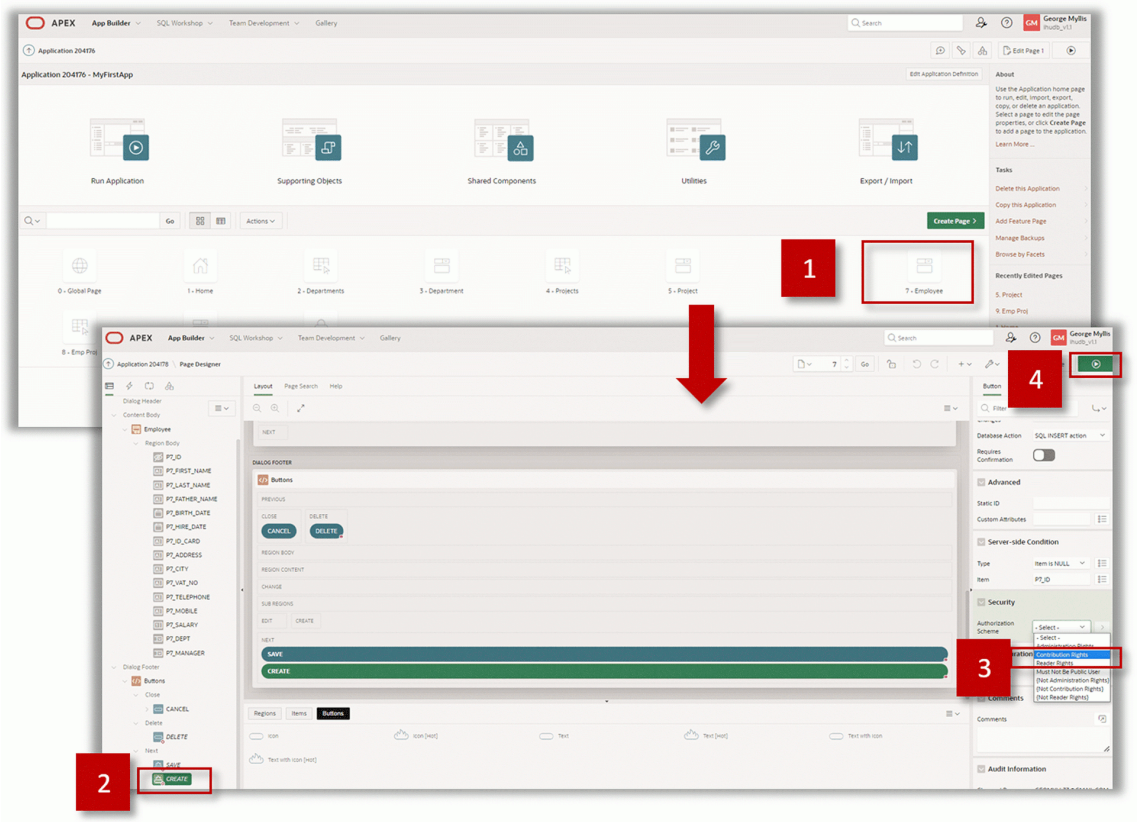
Ďalej je potrebné definovať roly pre používateľov. V našom príklade si ukážeme, ako sa to dá urobiť v samotnej aplikácii.

Krok 1: Prihláste sa do aplikácie pomocou poverení správcu, prejdite do bočného menu a kliknite na **Administration**, kliknite na **Add User**, zadajte identifikačné údaje používateľa a rolu a kliknite na **Add User** na Obrázku 5.21.



Obr. 5.21: Pridanie rolí používateľom.

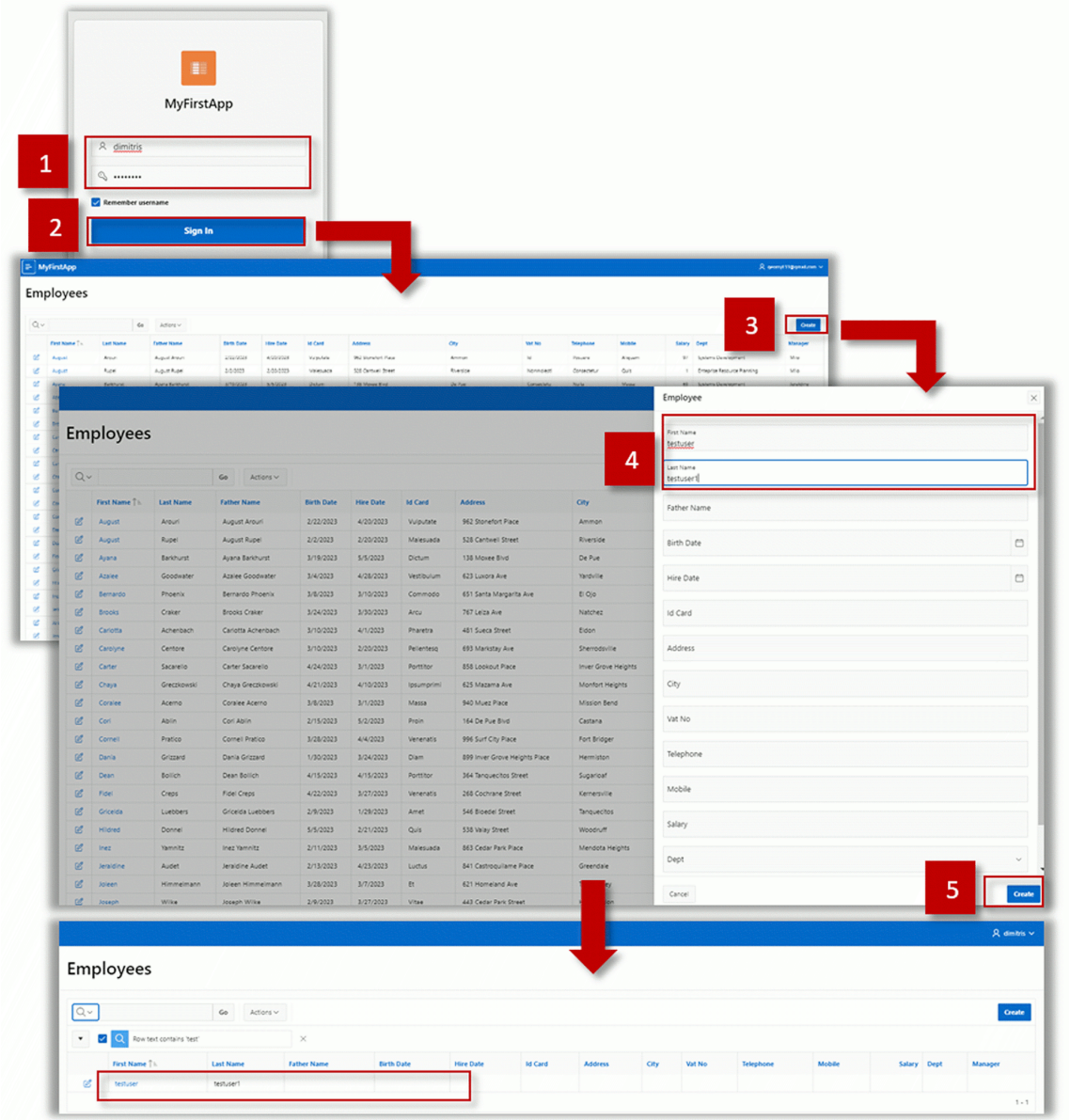
Krok 2: Prejdite do pracovného prostredia aplikácie a kliknite na **employees-Form**. V návrhárovi stránky formulára vyberte tlačidlo **Create** a v možnosti Zabezpečenie vyberte ako tému oprávnenia **Contribution Rights**. Kliknite na **Save**. Teraz budú môcť vytvárať a pridávať nových zamestnancov len používatelia s rolou Contributor, ako je znázornené na Obrázku 5.22.



Obr. 5.22: Vytvorenie konfigurácie tlačidla.

Krok 3: V tomto kroku sa overujú konfigurácie riadenia prístupu vykonané pre rolu Contibutor.

- Prihláste sa do aplikácie pomocou prihlasovacích údajov používateľa Dimitris, ktorý má pridelenú rolu Contributor.
- Otvorte ponuku a overte, že možnosť administrácie podľa očakávania chýba.
- Skúste vytvoriť a pridať nového zamestnanca s rolou Contributor, ako je uvedené na Obrázku 5.23.
- Overte, či môžete zobraziť existujúce informácie o zamestnancoch a či môžete vykonať zmeny.
- Overte, či môžete vytvoriť novú zamestnaneckú entitu.

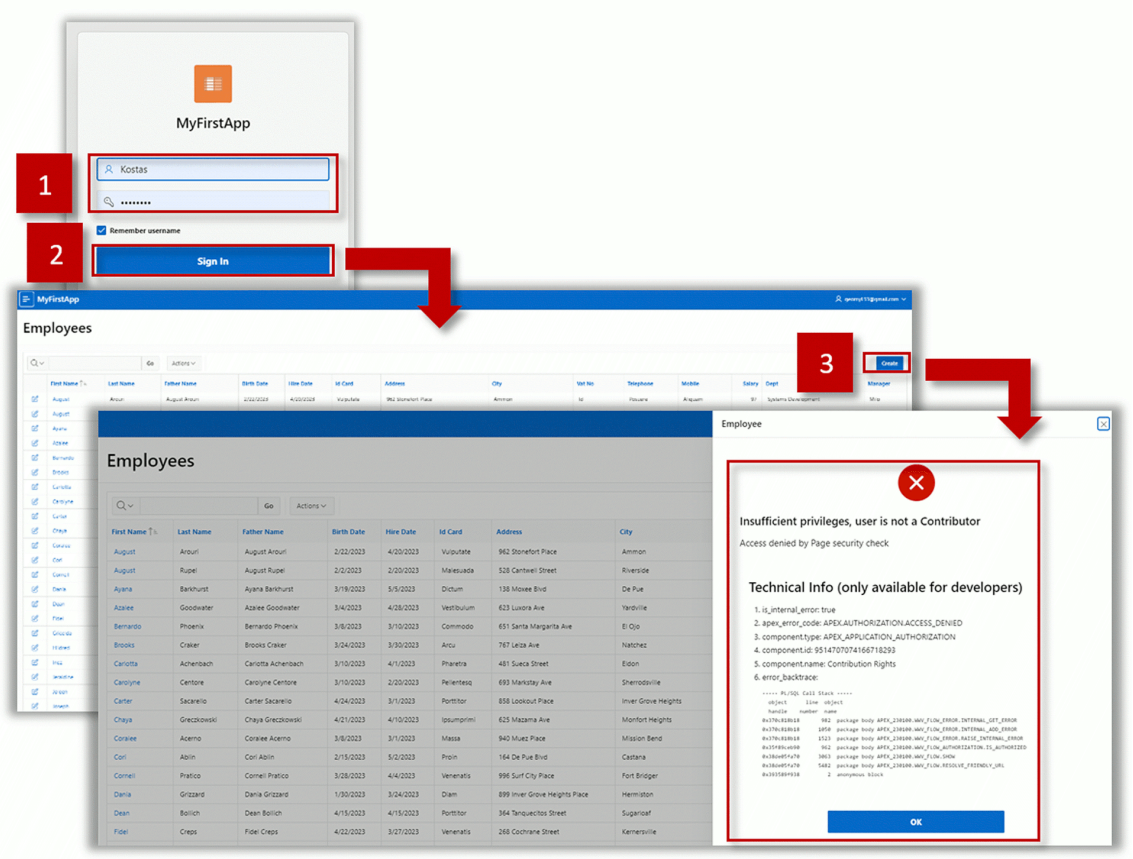


Obr. 5.23: Prístup a práva pre rolu Contributor.

Krok 4: Tento krok slúži na overenie konfigurácií riadenia prístupu pre rolu Reader.

- Prihláste sa do aplikácie pomocou prihlasovacích údajov používateľa Kostas, ktorý má pridelenú rolu Reader.
- Otvorte bočné menu a overte, či podľa očakávania chýba možnosť administrácie.
- Kliknite na Employees-Form v pracovnom prostredí aplikácie.
- Skúste vytvoriť alebo pridať nového pracovníka. Zobrazí sa správa “access denied”, ako je znázornené na obrázku 5.24.
- Skontrolujte, či si môžete prezerať informácie o existujúcich zamestnancoch, ale nemôžete vykonávať žiadne zmeny.

Tieto kroky potvrdzujú, že konfigurácia riadenia prístupu bola úspešne aplikovaná na používateľov.



Obr. 5.24: Prístup a práva pre rolu Reader.

5.5 Doplnkový učebný materiál

Nájdete tu tieto doplnkové učebné materiály:

- skripty
- aplikácia
- video návod

Všetky doplnkové vzdelávacie materiály sú k dispozícii na verejnej stránke projektu BeeAPEX. Prihláste sa ako hosťujúci používateľ (nevyžaduje sa heslo). Učebnicu nájdete v časti Knihy, skripty v priečinku Časť 2 > Kapitola 5 v časti Skripty a video návody v časti Zbierka video návodov. Materiály pre krátke kurzy sa nachádzajú v časti Krátke kurzy.

5.5.1 Exportované aplikácie

Exportovaná aplikácia je zabalená. Inštalácia vytvorí tabuľky, indexy, funkcie, procedúry a spúšťače, ako aj naplní údaje. Odinštalovaním sa odstránia všetky objekty databázy použité v tejto aplikácii.

5.5.2 Video sprievodcovia

Video sprievodca ukazuje každý krok pri vývoji aplikácie.

5.6 Otázky

1. Koľkými spôsobmi je možné vytvoriť novú aplikáciu v nástroji ORACLE APEX application Builder?
2. Ako môžete pridať používateľov do aplikácie?

3. Koľko zabudovaných používateľských oprávnení existuje v ORACLE APEX na riadenie prístupu k aplikácii alebo jej súčastiam?

5.7 Odpovede

1. Existujú 3 spôsoby
   - Nová aplikácia
   - Zo súboru
   - Štartovacia aplikácia
2. Prejdite do pracovného prostredia aplikácie a vyberte položku Manage Users and Group.
3. Existujú tri
   - Administrator
   - Contributor
   - Reader



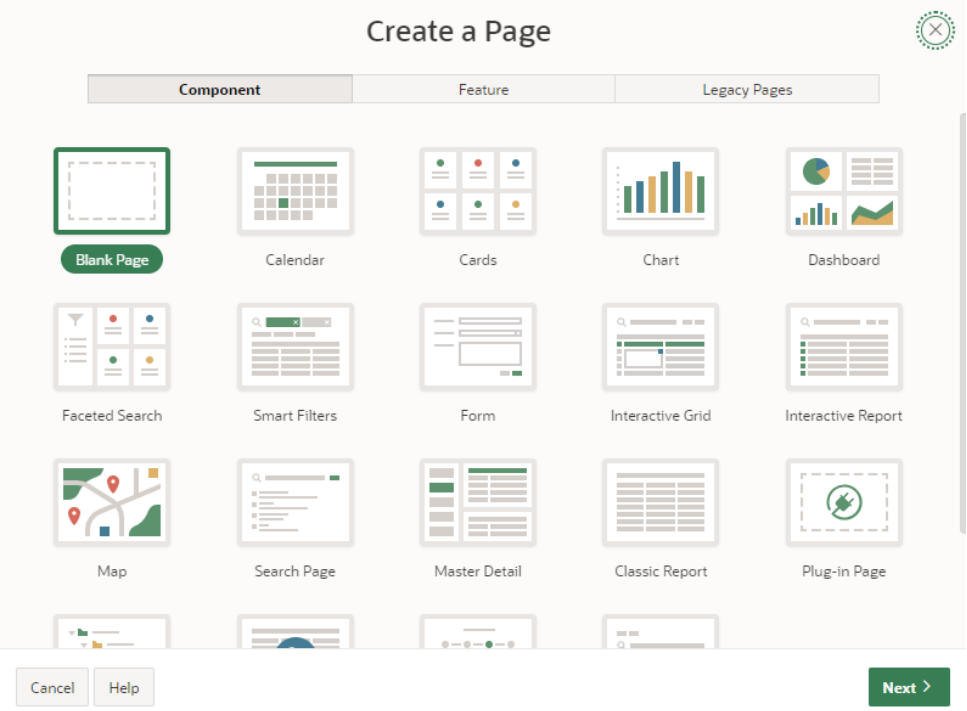
6. Ako spravovať reporty?

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

Údaje v databázach často nemajú význam, ak z nich nedostaneme výstupy, ktoré by nám povedali viac o skutočných informáciách, ktoré obsahujú. Na reprezentáciu výsledkov, pozorovanie a zisťovanie nám slúžia práve reporty, ktorým sa budeme venovať v nasledujúcej kapitole. Naučíme sa, ako takýto report jednoducho vytvoriť v prostredí APEX a ako ho upravovať. Budeme sa zaoberať rôznymi typmi reportov, ako sú interaktívne reporty, klasické reporty, Column Toggle Report alebo reporty s vizualizáciou grafov.

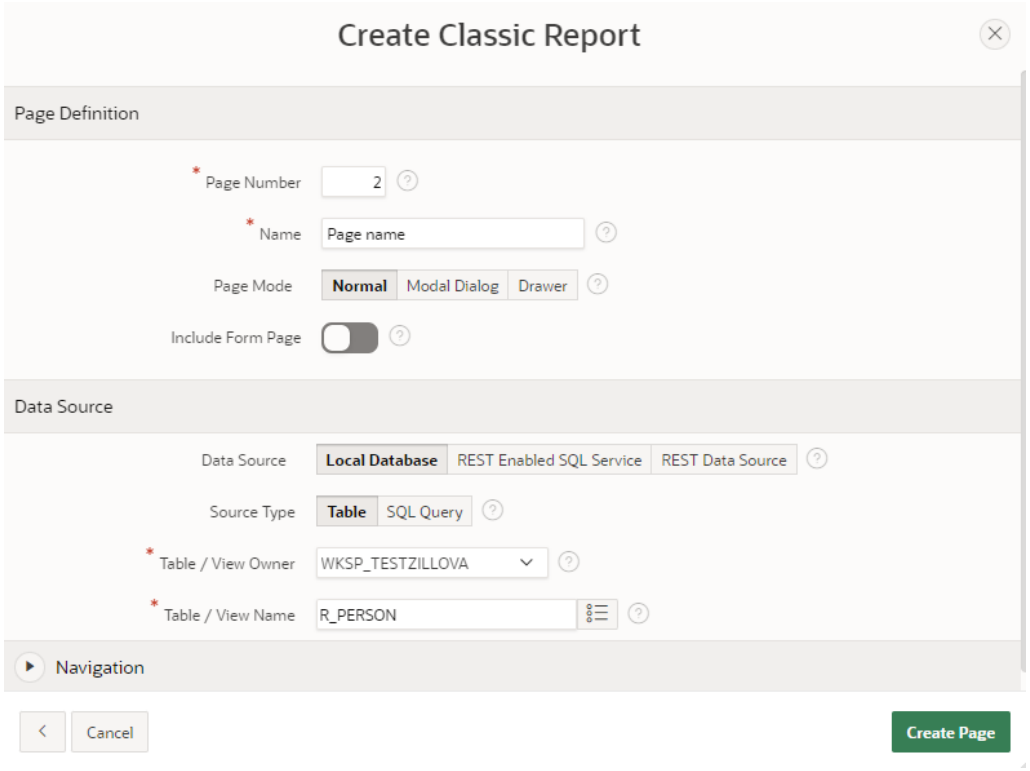
6.1 Report

Report je *oblasť*, ktorá zjednodušuje prehľadávanie údajov a zobrazuje údaje v riadkoch. Môže figurovať ako jedna jednotka alebo sa môže použiť na rozdelenie stránok do logických jednotiek. Môže byť statický, len s nadpisom a formátovaním, ale bez ďalších funkcií. Alebo to môže byť dynamická oblasť, ktorá má špecifickú funkčnosť, ako napríklad *kalendár, report* alebo *zoznam*. Tieto regióny sú predprogramované so svojou základnou funkčnosťou, ktorú môžu používatelia tiež ľahko upravovať.



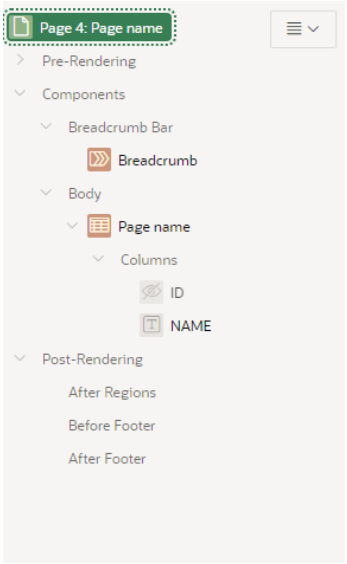
Obr. 6.1: Výber typu stránky.

Ako vidíte na obrázku 6.1, pri vytváraní **stránky** si používatelia môžu vybrať z predpripravených typov stránok, ktoré už obsahujú zostavy, takže nie je potrebné, aby používateľ sám vkladal oblasť. Príklad formulára na vytvorenie zostavy je zobrazený na obrázku 6.2. Pri vytváraní stránky obsahujúcej report je potrebné vyplniť názov stránky a zdroj údajov stránky. Údaje sa môžu načítať z *lokálnej databázy, z SQL s podporou REST* alebo zo zdroja *REST*. Pri výbere možnosti lokálnej databázy je potrebné vybrať, čo presne sa použije ako zdroj zostáv - používateľ môže použiť nielen samotnú *tabuľku* databázy, ale aj existujúce *pohľady* alebo vlastné *príkazy dotazu SQL*.



Obr. 6.2: Tvorba stránky s reportom.

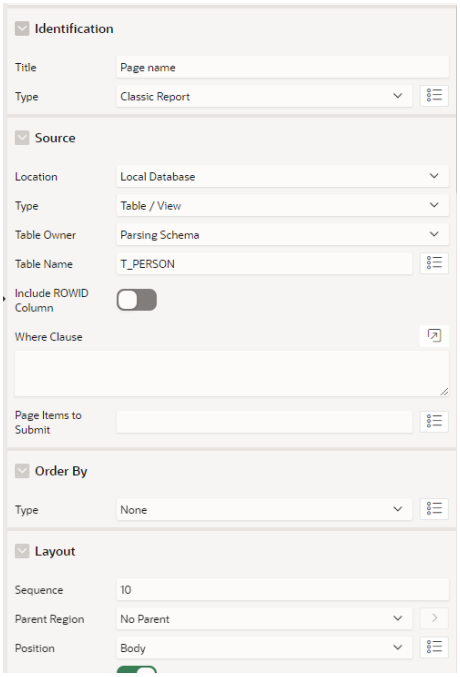
Po vytvorení stránky sa na ľavej strane zobrazí časť podobná tej na Obrázku 6.3. V bočnom navigačnom paneli môžete vidieť stránku, jej názov a všetky jej súčasti. Keď kliknete na niektorý z komponentov, zvýrazní sa. Potom môžete vidieť stĺpce, ktoré obsahuje, a tiež jej podoblasti alebo položky. Kliknutím pravým tlačidlom myši môžete vytvoriť novú prázdnu oblasť.



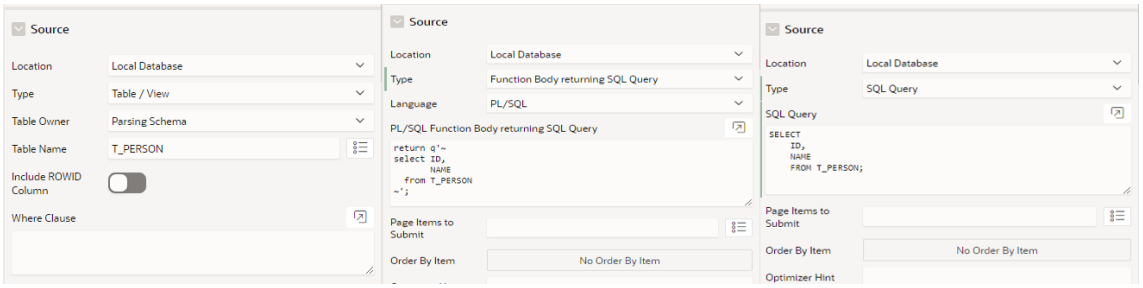
Obr. 6.3: Vykresľovanie.

Na obrázku 6.4 môžete vidieť oblasť a podrobnosti o jej atribútoch. **Identifikácia** definuje názov stránky a typ reportu (napr. klasický report). Na obrázku 6.5 je zobrazených niekoľko príkladov zdrojovej časti, v ktorej je definovaný zdroj údajov pre náš report, ako aj informácie o

pôvode údajov (ako bolo uvedené vyššie: *tabuľka, pohľad, funkcia* alebo *dotaz SQL*). **Klauzula where** hovorí o tom, aká podmienka sa použije na filtrovanie údajov.



Obr. 6.4: Vlastnosti oblasti.



Obr. 6.5: Rôzne zdroje údajov.

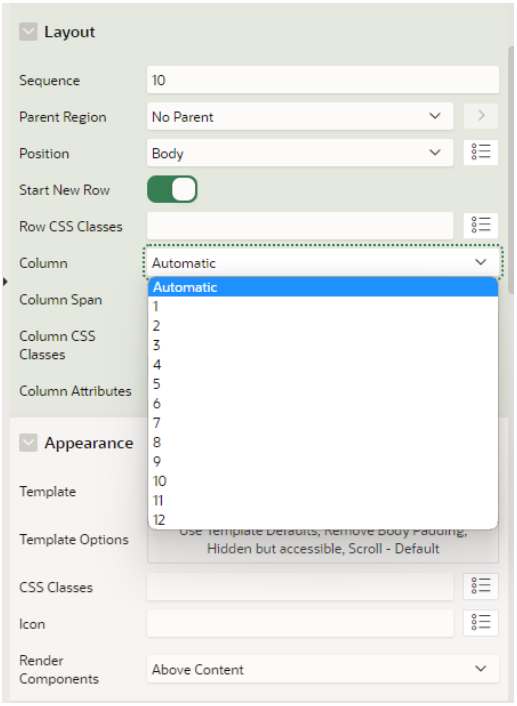
Na obrázku 6.6 môžete vidieť **Rozloženie**, kde môžete určiť umiestnenie oblasti na stránke. **Nadradená oblasť** slúži na určenie, ktorá oblasť bude obsahovať našu správu. Túto oblasť môžete upraviť aj prostredníctvom položky **Pozícia**. Pozície sú definované konkrétnou *šablónou stránky*. Po vytvorení stránky sa vytvorí s konkrétnou témou a rozložením v závislosti od zvolenej šablóny. Predvolená šablóna pre tému **Univerzálna** sa nazýva **Štandardná**.

Oblasti sú usporiadané do *mriežky* - každá oblasť má teda svoje vlastné umiestnenie definované riadkom a stĺpcom. Tieto oblasti sú usporiadané podľa svojho **čísla poradia**. Prvá oblasť je v prvom riadku a prvom stĺpci mriežky. Každá ďalšia oblasť je potom usporiadaná podľa jedného z týchto vzorov: v rovnakom riadku a stĺpci ako predchádzajúca oblasť a je zobrazená pod predchádzajúcou oblasťou. Atribúty **Začať nový riadok** a **Nový stĺpec** sa používajú na určenie spôsobu usporiadania týchto oblastí.

Veľkosť a *Umiestnenie* sa vypočítajú po priradení riadku a stĺpca. To sa dosiahne výpočtom obsadenia a šírky pre každú časť riadku pomocou hodnôt parametrov **Stĺpec** a **Priestor stĺpca**.

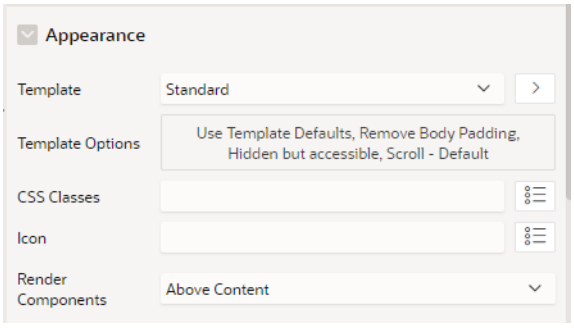
Odsadenie oblasti v jej riadku je definované atribútom **Stĺpec**. Hodnota tohto stĺpca je v rozsahu **1 - 12**, ktorý sa vzťahuje na **1** z **12**. **bodov** mriežky, ktorá rozdeľuje šírku. Hodnota **1** znamená

najviac *ľavú časť* a **7** je pre *stred* stránky.

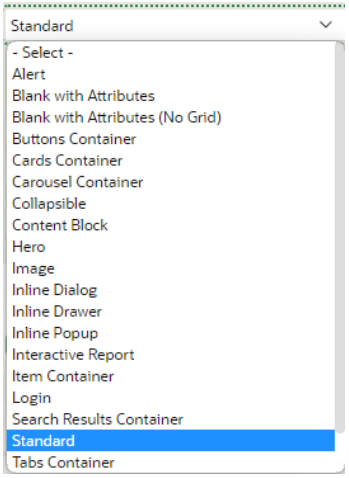


Obr. 6.6: Rozloženie.

Na obrázku 6.7 môžete vidieť časť **Vzhľad**, ktorá sa používa na úpravu šablóny pre zadanú oblasť. Na obrázku 6.8 sú zobrazené preddefinované šablóny, z ktorých si používateľ môže vybrať. Každú šablónu je možné prispôsobiť kliknutím na možnosti šablóny alebo napísaním vlastných príkazov **CSS**. V možnostiach šablóny môžete definovať viaceré pravidlá, ako napríklad *výška tela*, *viditeľnosť hlavičky* alebo dokonca *vzdialenosť medzi položkami*.



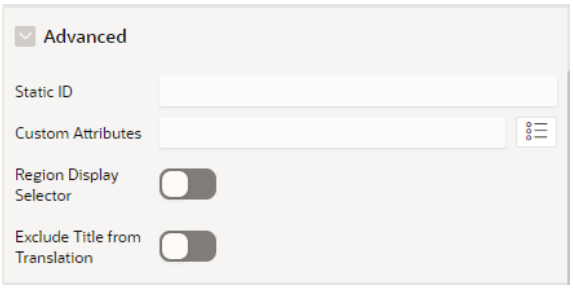
Obr. 6.7: Vzhľad.



Obr. 6.8: Šablóny.

Obrázok 6.9 zobrazuje časť s názvom **Pokročilé**, kde si môžete vybrať zo štyroch možností. **Statické ID** je určené na nastavenie identifikátora správy. Toto **Statické ID** sa používa ako **ID** pre prvok **HTML** v rámci stránky pre situácie, keď chce používateľ nastaviť vzhľad stránky pomocou **CSS**. **Vlastné atribúty** slúžia na doplňujúce informácie pre prvok **HTML**. Vlastné atribúty sú uložené ako názov/hodnota, napr: $name = "MY_REPORT"$

Výber zobrazenia oblasti je komponent oblasti, ktorý poskytuje navigačné ovládacie prvky pre iné oblasti na stránke. **Vylúčiť nadpis z prekladu** zabráni prekladu nadpisu v prípade, že stránka má definované preklady.



Obr. 6.9: Pokročilé.

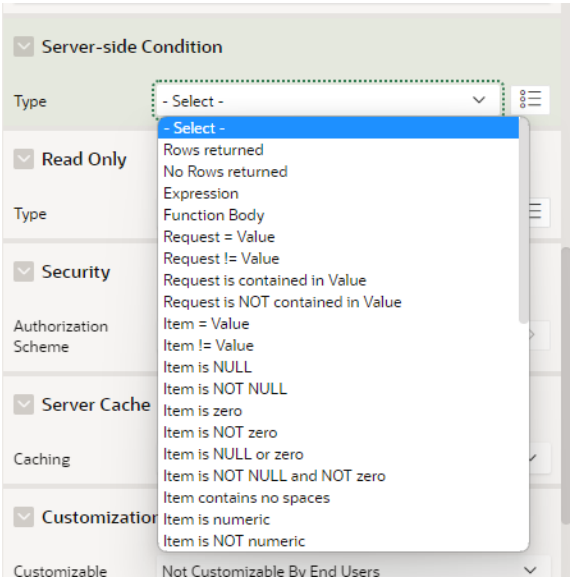
Hlavička a päta zobrazené na obrázku 6.10 sa používajú na úpravu obsahu týchto častí správy. Je dôležité spomenúť, že hlavička a päta sa týkajú reportu a nie celej stránky.



Obr. 6.10: Hlavička a päta.

Typy **Podmienok na strane servera** sú uvedené na obrázku 6.11. V aplikácii vyvinutej pomocou Oracle APEX sa **podmienky na strane servera** riešia v inicializačnej fáze načítania

webovej stránky. Správa sa nezobrazí, ak nie sú tieto podmienky splnené. Existuje množstvo typov podmienok. Ako príklad možno uviesť porovnávanie hodnôt položiek alebo existencia podmienok na kontrolu požiadaviek.



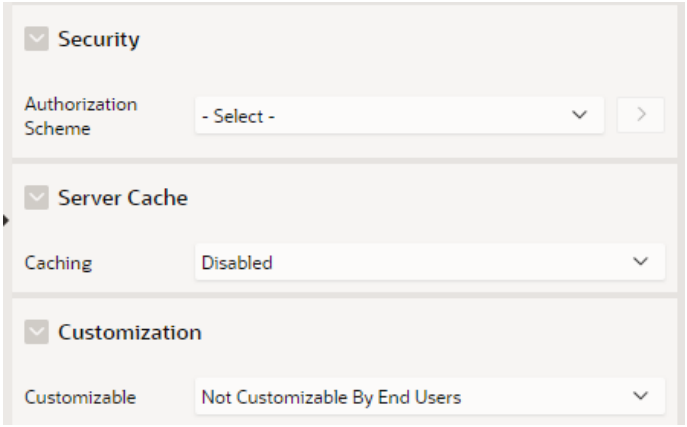
Obr. 6.11: Podmienky na strane servera.

Položka **len na čítanie** zobrazená na obrázku 6.12 má rovnaké možnosti ako položka **podmienka na strane servera** zobrazená na obrázku 6.11. Prirodzene, nie je možné zmeniť zobrazenie vybranej oblasti, kým je len na čítanie. V prípade základnej správy nie je možné triediť hodnoty stĺpcov.



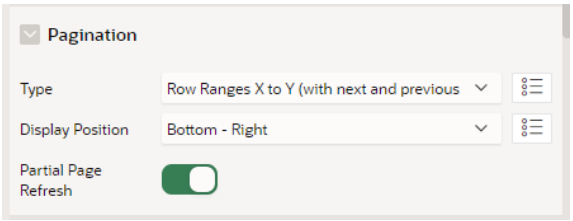
Obr. 6.12: Len na čítanie.

Na obrázku 6.13 môžete vidieť posledné tri časti, ktoré budú vysvetlené. Sú to **Zabezpečenie, Vyrovnávacia pamäť servera** a **Prispôsobenie**. Sekcia **Zabezpečenie** obsahuje možnosti obmedzení z hľadiska zobrazovania webovej stránky - napr. *verejná webová stránka, len pre oprávnených používateľov*. Sekcia **Vyrovnávacia pamäť servera** obsahuje nastavenia týkajúce sa správy vyrovnávacej pamäte. V okne **Prispôsobenie** môžete nastaviť prístup pre úpravu zostavy.



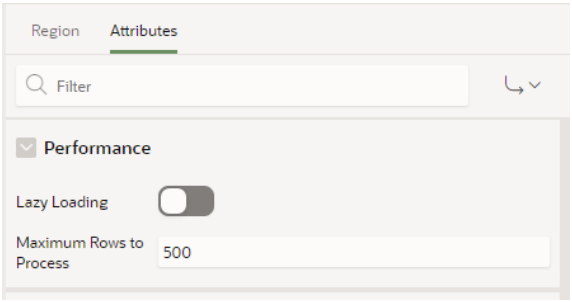
Obr. 6.13: Zabezpečenie, Vyrovnávacia pamäť servera a Prispôsobenie.

Ďalšou skupinou sekcií sú **Atribúty**. Môžete tu nastaviť veci ako *počet zobrazených riadkov*, či je možné *vypisovať cez výsledky správy* a akým *spôsobom sa budú vypisovať* v celej zostave.

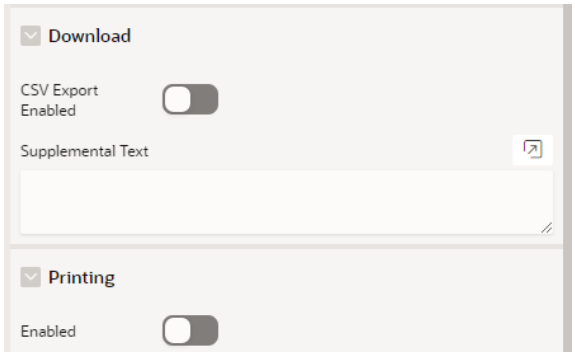


Obr. 6.14: Stránkovanie.

Ďalšou zaujímavou funkciou je nastavenie maximálneho počtu načítaných riadkov. Môžete tiež definovať, čo sa má zobraziť, keď *nie sú nájdené žiadne údaje*. Správy je možné stiahnuť ako súbor **.csv** alebo vytlačiť. Všetko, čo je uvedené v tomto odseku, nájdete v časti **Atribúty**.

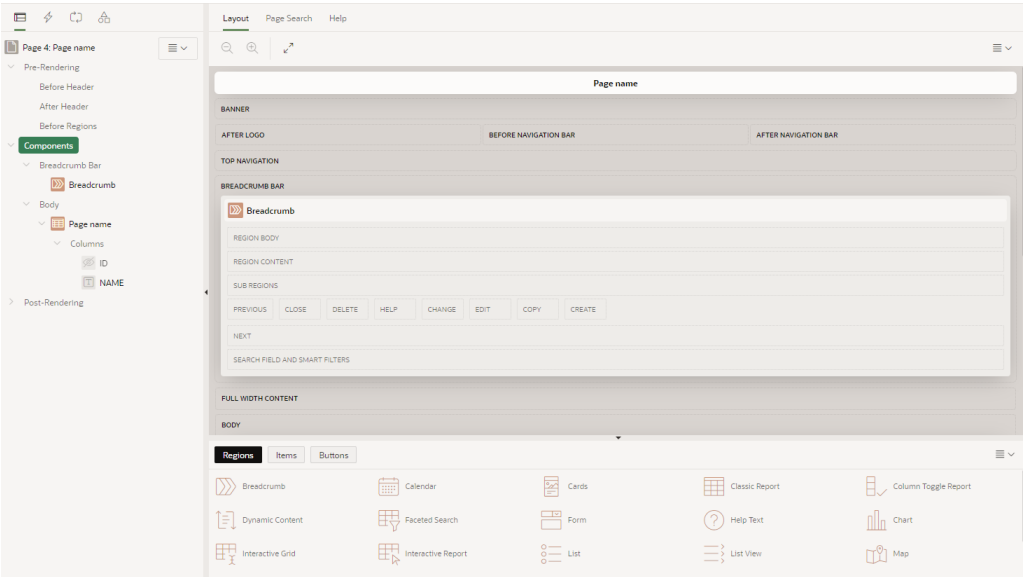


Obr. 6.15: Počet riadkov, ktoré sa majú načítať.



Obr. 6.16: Stiahnutie a tlač.

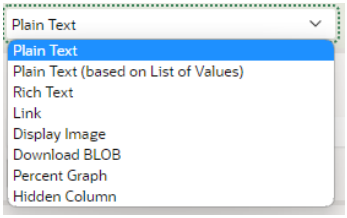
Veľmi dôležitou súčasťou je *stred* aplikácie. Na obrázku 6.17 môžete vidieť **Rozloženie**, ktoré nám ukazuje, ako budú všetky komponenty usporiadané na našej stránke. Celá stránka je štruktúrovaná do mriežky, takže je intuitívnejšie si predstaviť, kde budú komponenty umiestnené. Aby sme boli používateľsky čo najprívetivejší, môžete si vybrať oblasť/položku a *pretiahnuť & pustiť* ju na požadované miesto. Pre každý región existuje jeho predvolená šablóna, ktorá sa poskytuje pri jeho vložení. Všetky potrebné nastavenia sú už nastavené.



Obr. 6.17: Stred dizajnéra stránky.

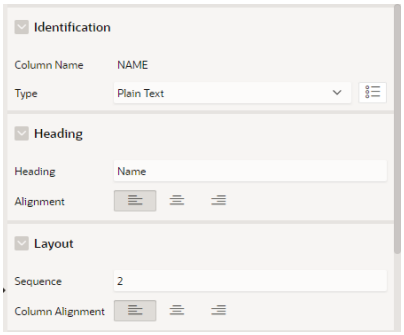
Ako môžete vidieť na ľavej strane stránky na obrázku 6.17, APEX štandardne vytvára niektoré komponenty pre správu. Každá položka predstavuje jeden stĺpec a je možné ju meniť rovnakým spôsobom ako pri zostave.

Nastavenia stĺpcov sú znázornené na obrázku 6.19. V časti **Identifikácia** môžete vidieť dve funkcie - **Typ** a **Názov stĺpca**. Nie je možné zmeniť **Názov stĺpca**, pretože jeho hodnota sa získava z *SQL dopytu*. **Typ stĺpca** zobrazený na obrázku 6.18 určuje, ako sa bude stĺpec zobrazovať - na niektoré z nich upozorníme: *Skrytý* (nezobrazí sa nič), *Obyčajný text*, *Zobraziť obrázok, percentuálny graf* atď. V predvolenom nastavení sú všetky stĺpce označené ako *Obyčajný text*.



Obr. 6.18: Typ stĺpca.

Záhlavie pozostáva z dvoch možností hlavičky. Na rozdiel od názvu stĺpca, **Záhlavie** (názov) možno upravovať. Záhlavie môže obsahovať aj kód **HTML**. Možnosť **Zarovnanie** hovorí sama o sebe. Sekcia **Layout** ponúka možnosť **Sekvencie**, ktorá určuje, v akom poradí budú stĺpce uvedené, a tiež možnosť **Alignment**, rovnakú ako v predchádzajúcej časti.



Obr. 6.19: Nastavenia stĺpcov.

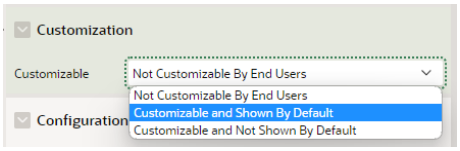
Poslednou dôležitou časťou je **Vzhľad**, ktorý by mal byť upravený, ak chcete číslo spracovať ako reťazec v nejakom *špeciálnom formáte*. Ako môžete vidieť na obrázku 6.20, po kliknutí na **Formát masky** si môžete vybrať z množstva základných formátov. Tieto formáty ponúka samotný APEX, ale ak chcete, môžete vytvoriť svoj vlastný formát.



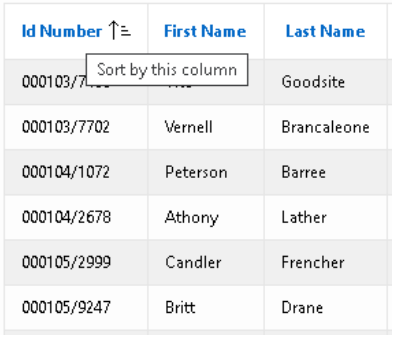
Obr. 6.20: Formát masky.

6.2 Klasická zostava

Klasická zostava je typ zostavy, ktorá umožňuje len základné zobrazenie údajov bez možnosti úpravy zobrazenia. Jediné, čo má používateľ k dispozícii, je triedenie údajov podľa jednotlivých stĺpcov, aj to je potrebné nastaviť.

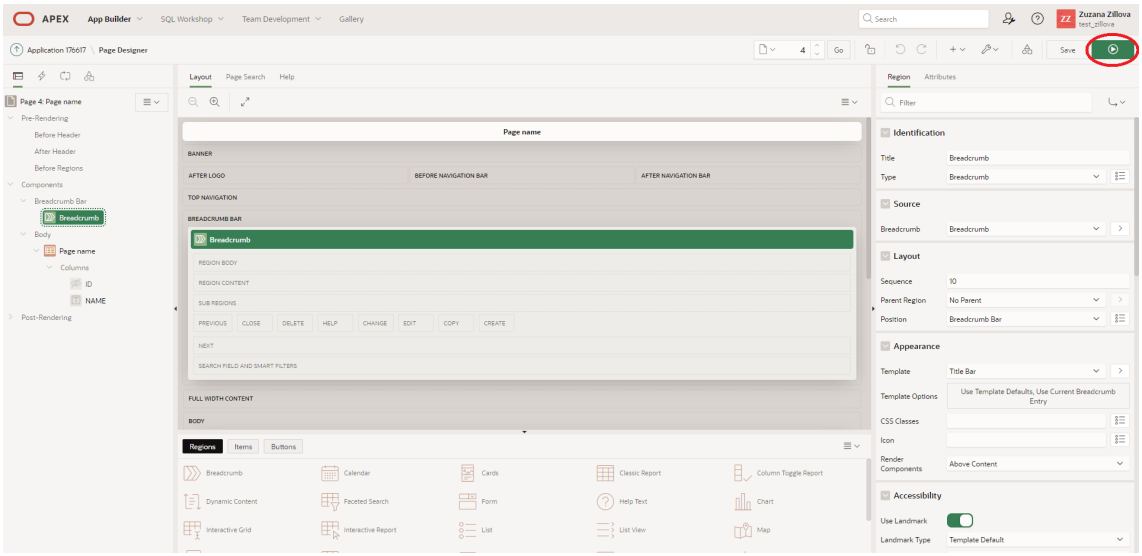


Obr. 6.21: Prispôsobenie.

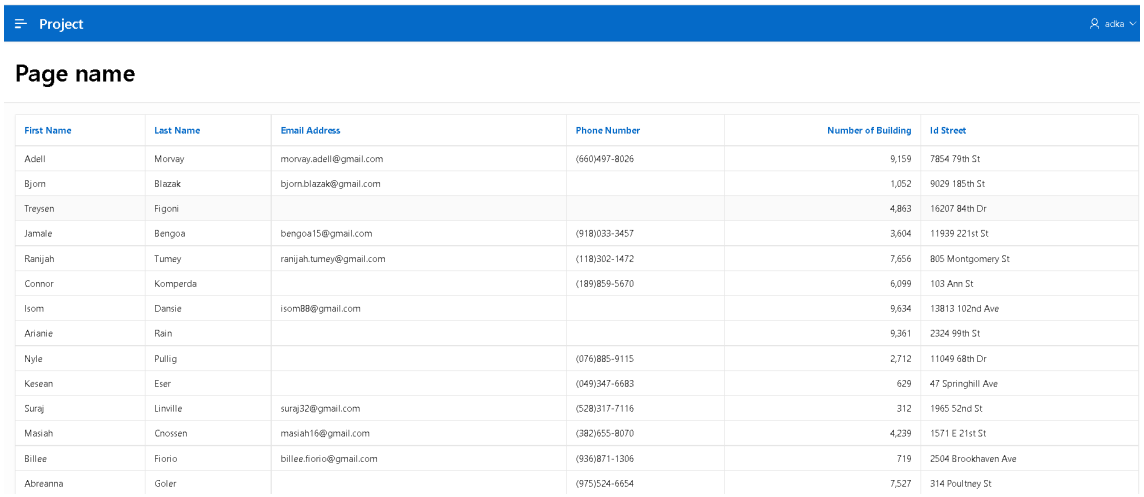


Obr. 6.22: Možnosť triedenia podľa stĺpcov.

Ak chcete vidieť, ako bude stránka vyzerať, môžete kliknúť na zelenú šípku v kruhu. Ako vidíte na obrázku 6.23, šípka sa nachádza v pravom hornom rohu. Na obrázku 6.24 vidíme, ako vyzerá klasická správa. Má stĺpce s názvami a údajmi. Používateľovi nie je umožnená žiadna interakcia okrem triedenia podľa stĺpcov.



Obr. 6.23: Umiestnenie zelenej šípky pre prezentačnú stránku.

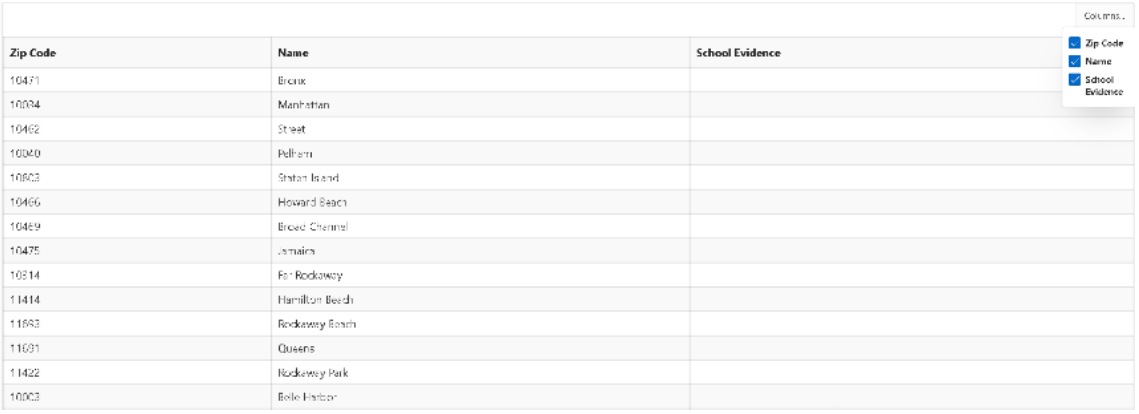


First Name	Last Name	Email Address	Phone Number	Number of Building	Id Street
Adell	Morvay	morvay.adell@gmail.com	(660)497-8026	9,159	7854 79th St
Bjorn	Blazak	bjorn.blazak@gmail.com		1,052	9029 185th St
Treysen	Figoni			4,863	16207 84th Dr
Jamale	Bengoa	bengoa15@gmail.com	(918)033-3457	3,604	11939 221st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

Obr. 6.24: Príklad klasickej zostavy.

6.3 Zostava s možnosťou prepínania stĺpcov

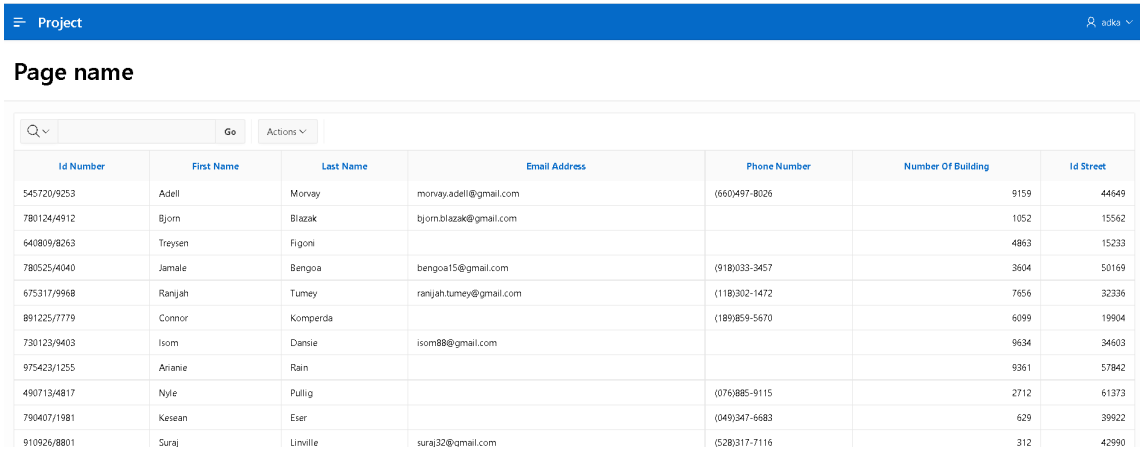
Zostava s možnosťou prepínania stĺpcov je takmer rovnaký ako klasická zostava, len s jednou funkciou navyše, ktorá umožňuje určiť, ktorý stĺpec má byť viditeľný a ktorý nie. Vidíme to v pravom hornom rohu *Obrázok 6.25*.



Obr. 6.25: Možnosť zmeny viditeľnosti stĺpca v Zostava s možnosťou prepínania stĺpcov.

6.4 Interaktívna zostava

Interaktívna zostava znázornená na obrázku 6.26 umožňuje používateľom vykonať niektoré úpravy správy pri prezeraní stránky bez toho, aby mali prístup k jadru aplikácie. Interaktívna správa sa vytvára rovnako ako každá iná a vzhľadovo sa veľmi nelíši. Teraz sa bližšie pozrieme na jednotlivé funkcie a prispôsobenia, ktoré môže používateľ v interaktívnej správe upraviť.

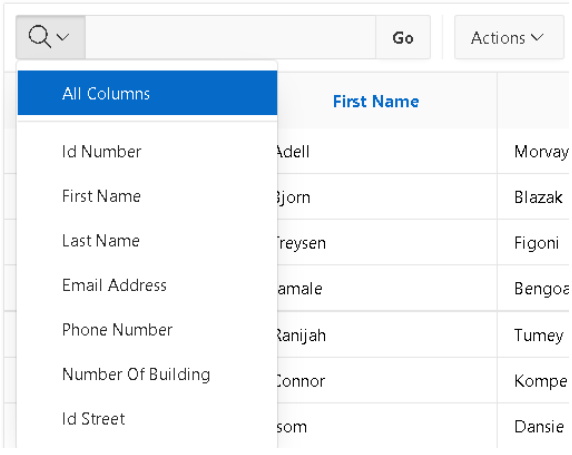


Obr. 6.26: Interaktívna správa.

Panel vyhľadávania zobrazený na obrázku 6.27 pozostáva zo štyroch položiek:

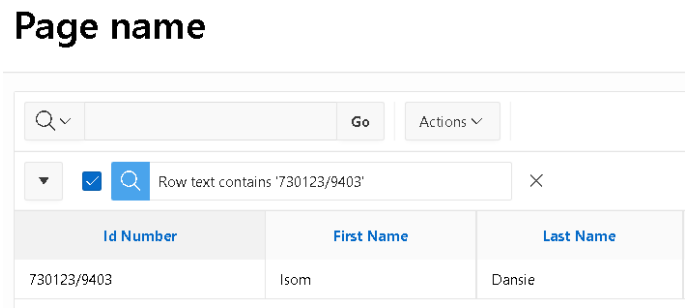
- **Lupa,**
- **Textové pole,**
- **Tlačidlo Prejsť,**
- **Tlačidlo Akcií**.

Prvé tri položky umožňujú používateľovi filtrovať riadky zostavy. Do poľa zadáme vyhľadávací reťazec, klikneme na **Lupa**, kde vyberieme, ktoré stĺpce sa majú zobraziť, a potom klikneme na **Tlačidlo Prejsť**. Zobrazia sa tie záznamy, ktoré obsahujú hľadaný reťazec.



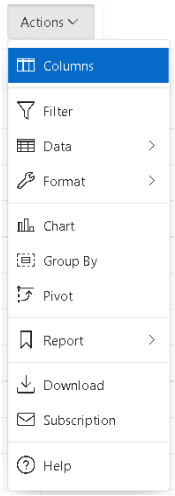
Obr. 6.27: Vyhľadávací panel.

Napríklad chceme vyhľadať záznamy, ktoré obsahujú číslo **“730123/9403”** v ľubovoľnom stĺpci. Tento filter sa zobrazí a možno ho vypnúť kliknutím na **tlačidlo X**. Situácia je znázornená na obrázku 6.28. Môžeme mať súčasne aktívnych viacero filtrov v rôznych stĺpcoch.



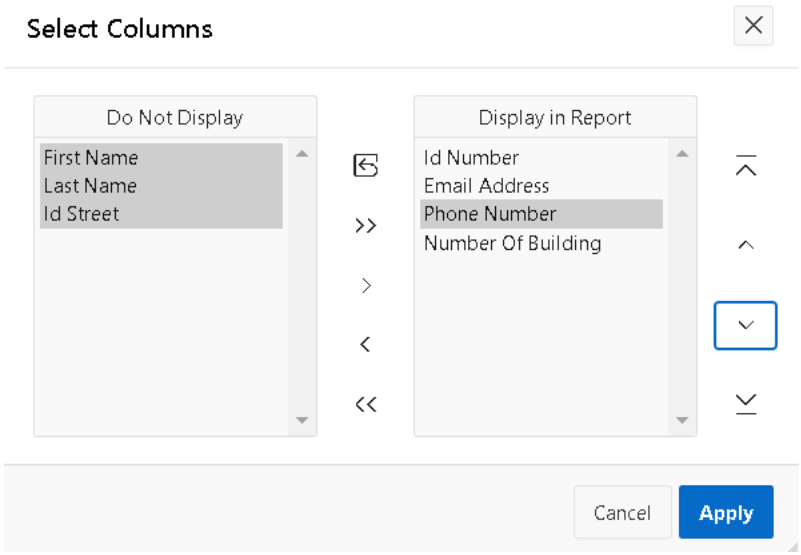
Obr. 6.28: Vyhľadávanie.

Tlačidlo Akcií ponúka viac možností nastavenia zobrazenia zostavy. Po kliknutí sa zobrazí ďalšia ponuka obsahujúca niekoľko položiek. Uvedené položky sú znázornené na obrázku 6.29.



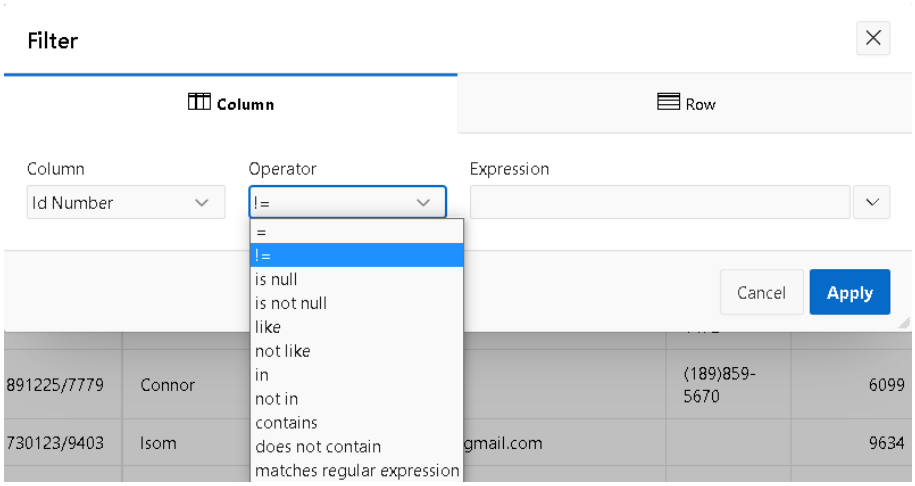
Obr. 6.29: Tlačidlo Akcie.

Tlačidlo **Stĺpce** nám zobrazí okno, ktoré môžete vidieť na obrázku 6.30. Tu si používateľ môže vybrať, ktoré stĺpce sa v správe zobrazia, a tiež poradie, v akom sa majú zobraziť. Používateľ to môže dosiahnuť presunutím stĺpcov medzi dvoma sekciami **Nezobrazovať** a **Zobraziť v zostave**. K dispozícii sú užitočné šípky, ktoré uľahčujú prácu s presúvaním stĺpcov.



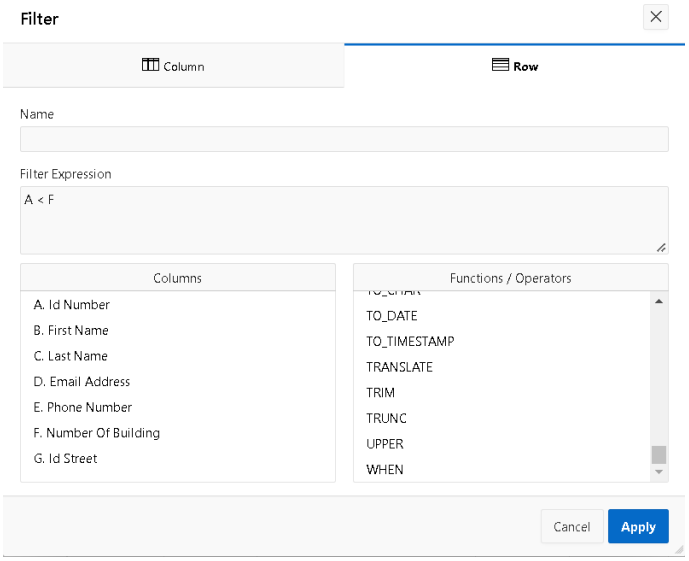
Obr. 6.30: Výber stĺpcov na zobrazenie.

Ďalšou možnosťou je **Filter** zobrazený na obrázku 6.31. Hodnoty v stĺpci môžeme filtrovať pomocou výrazov ako *byť rovný, nie je rovný, je nulový* atď. Výraz môžete definovať výberom z preddefinovaných alebo napísaním vlastného.



Obr. 6.31: Stĺpcový filter.

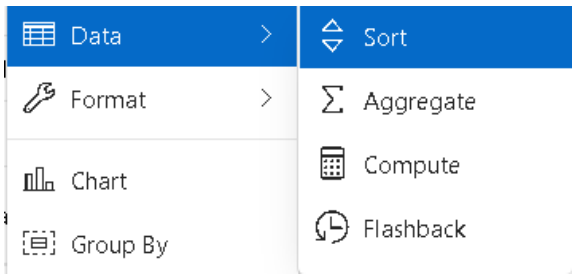
V prípade filtrovania riadkov máme na výber z nespočetného množstva operácií a výrazov, ktoré môžeme zostaviť. Tento filter sa zobrazí v okne **Výraz filtra** zobrazenom na obrázku 6.32.



Obr. 6.32: Riadkový filter.

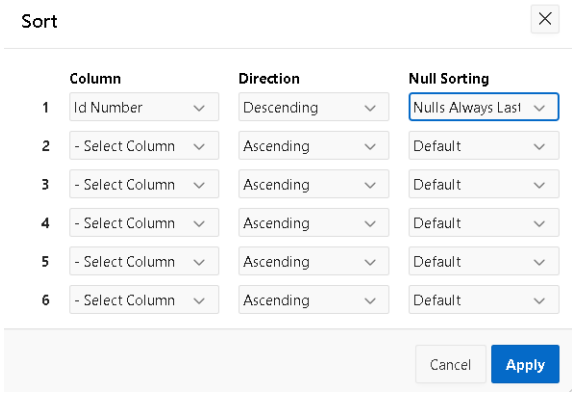
Po kliknutí na tlačidlo **Údaje** sa zobrazí ponuka znázornená na obrázku 6.33. Ponuka obsahuje ďalšie štyri možnosti:

- triedenie,
- agregácia,
- vypočítať,
- retrospektíva.



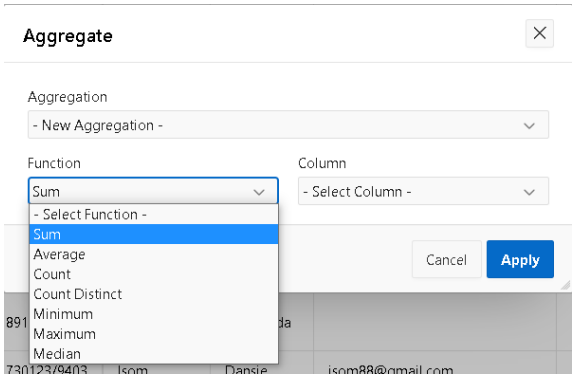
Obr. 6.33: Po kliknutí na tlačidlo Údaje.

Na obrázku 6.34 môžete vidieť možnosť **Triedit'**, ktorú možno použiť na každý stĺpec a používateľ môže určiť, ako pracovať s *null hodnotami*.



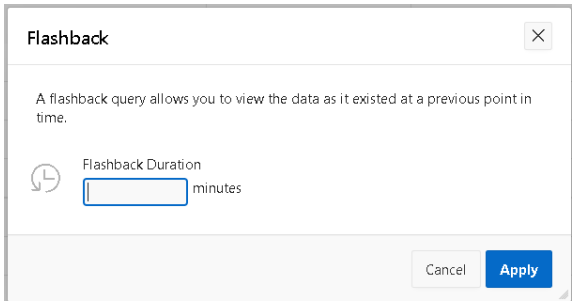
Obr. 6.34: Triediť.

Pomocou **Agregácie** môžeme vykonať agregačnú funkciu nad jedným zo stĺpcov našej zostavy. Agregačné funkcie, z ktorých si môžeme vybrať, sú zobrazené na obrázku 6.35.



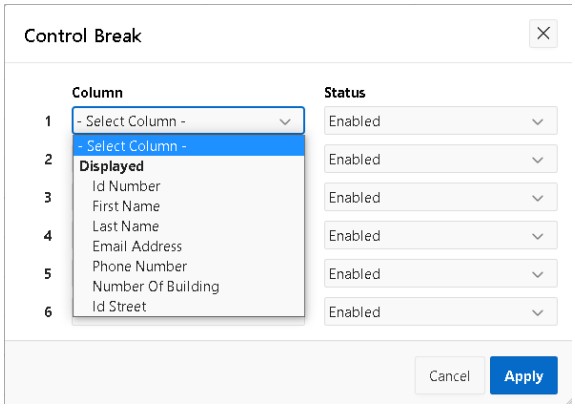
Obr. 6.35: Agregácia.

Funkcia **retrospektívy** znázornená na obrázku 6.36 nám umožňuje vrátiť sa do stavu zobrazenej správy pred zmenami pred niekoľkými minútami.



Obr. 6.36: Retrospektíva.

Kontrolná prestávka zobrazená na obrázku 6.37 nám umožňuje rozdeliť správu na niekoľko menších celkov, akoby to boli samostatné správy. Tieto malé jednotky sú rozdelené v závislosti od hodnôt stĺpcov. Každá jednotka zdieľa rovnaké údaje vo vybranom stĺpci.



Obr. 6.37: Kontrolná prestávka.

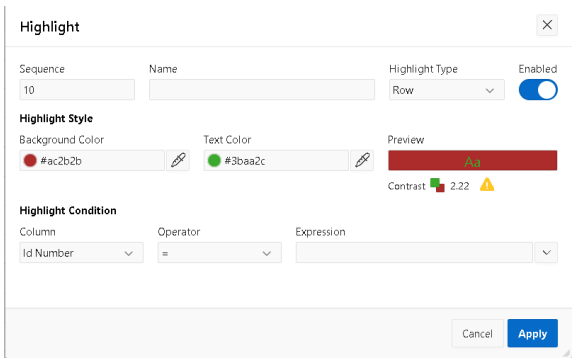
Obrázok 6.38 nám ukazuje kontrolnú prestávku aplikovanú na stĺpec *Meno*.

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

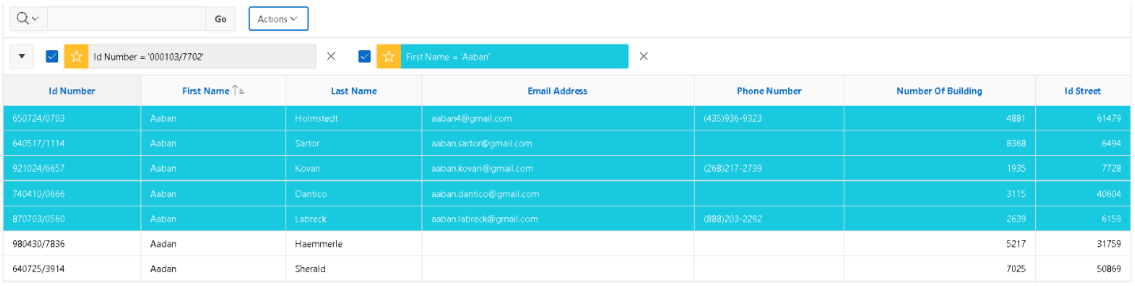
Obr. 6.38: Výsledok kontrolnej prestávky.

Na obrázku 6.39 môžete vidieť funkciu **Zvýraznenie**, ktorá ľahko sprehľadní záznamy v zostave bez filtrovania ostatných. Môžeme zvýrazniť buď jednu bunku, alebo celý riadok farbami, ktoré si zvolíme. Porovnávanie hodnoty stĺpca funguje rovnako ako pri funkcii **Filter**, s tým rozdielom, že nefiltrované záznamy zo zostavy nezmiznú.



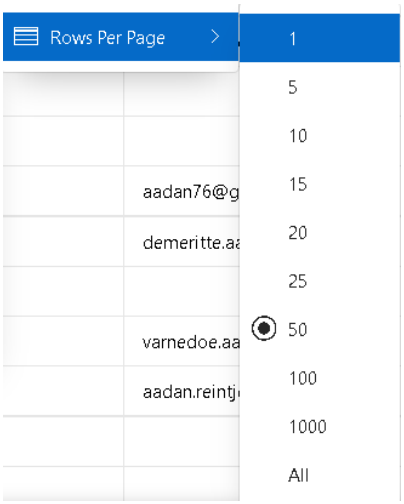
Obr. 6.39: Zvýraznenie.

Na obrázku 6.40 vidíme správu so zvýraznenými riadkami, ktorých prvé meno sa rovná **“Aaban”**.



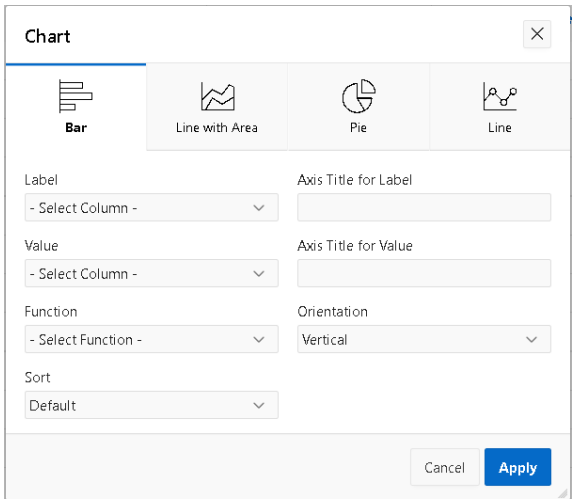
Obr. 6.40: Zvýraznenie výsledku.

Obrázok 6.41 zobrazuje **Riadky na stránke**, čo je atribút, pri ktorom si používateľ môže vybrať, koľko záznamov sa zobrazí na stránke. Je to podobné nastavenie ako **Stránkovanie**.



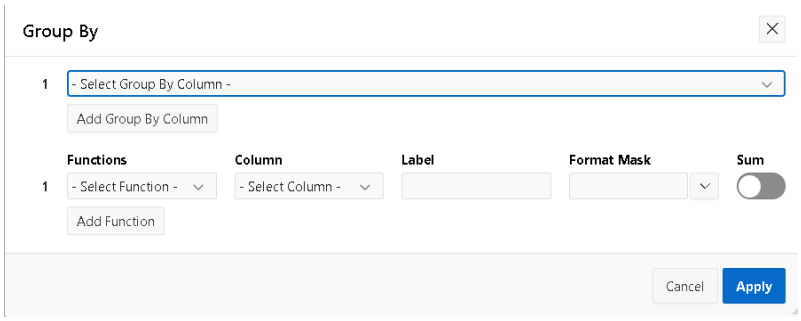
Obr. 6.41: Riadky na stránke.

Ak je zobrazenie údajov v správe nedostatočné, existuje aj možnosť premietnuť tieto údaje do grafu pomocou položky **Graf** zobrazenej na obrázku 6.42. Na výber sú štyri základné grafy. Môžeme si tiež vybrať agregačnú funkciu, ktorá sa aplikuje na daný stĺpec.



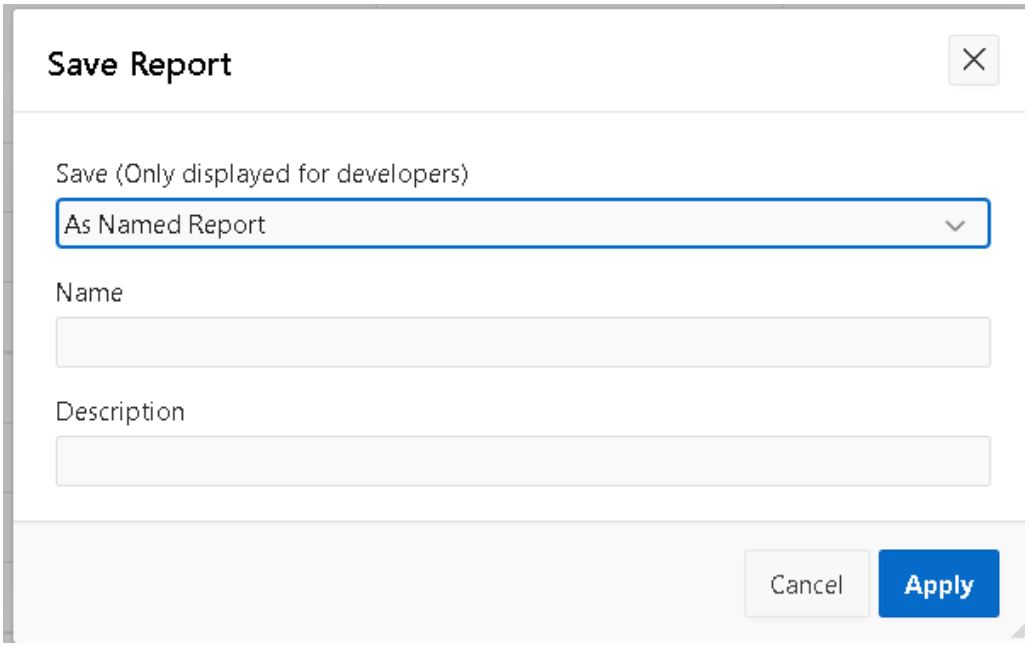
Obr. 6.42: Graf v interaktívnej správe.

Veľmi dôležitou funkciou interaktívnej správy je aj **Zgrupovať podľa** zobrazený na obrázku 6.43. Funguje rovnakým spôsobom ako v SQL dotaze.



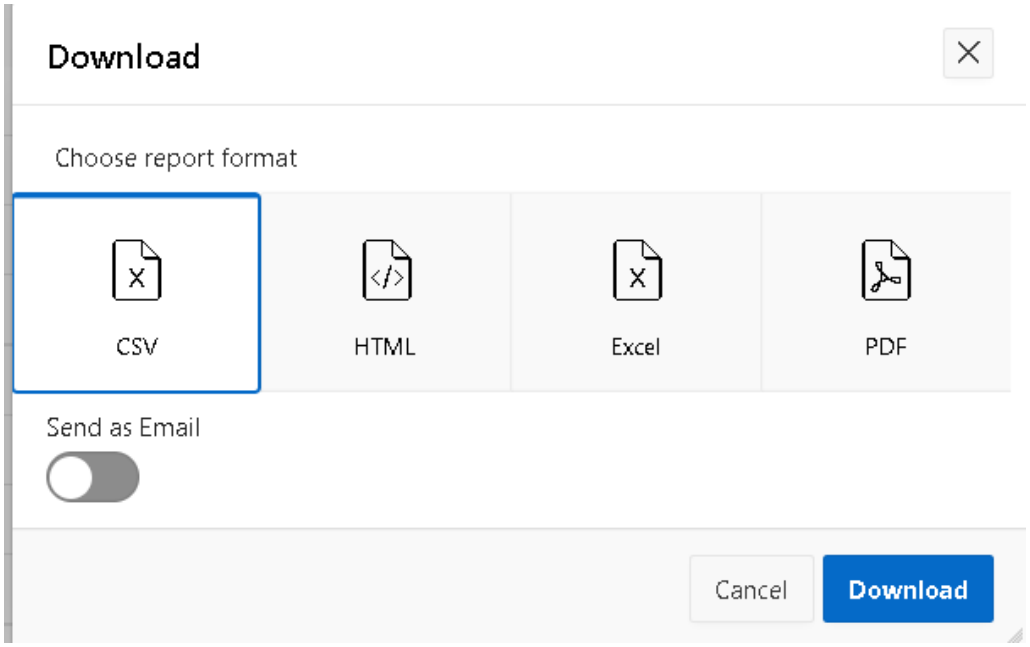
Obr. 6.43: Zgrupovať podľa.

Funkcia **Uložiť zostavu** zobrazená na obrázku 6.44 uloží zostavu v jej aktuálnom stave. Túto zostavu však môže vidieť len vývojár po prihlásení do aplikácie a nie používateľ.



Obr. 6.44: Uložiť zostavu.

Môžeme tiež obnoviť pôvodný stav zostavy alebo ju stiahnuť vo vybranom formáte. Túto možnosť môžete vidieť na obrázku 6.45.



Obr. 6.45: Stiahnuť.

6.5 Otázky

1. Do akej kategórie patrí typ zostavy? Oblasti, položky alebo tlačidlá?
2. Čo môže byť zdrojom údajov pre zostavu?
3. Aké typy zdrojov je možné použiť na definovanie klasickej zostavy?
4. Aká je hlavná funkcia klasickej zostavy?
5. Ktorá technológia umožňuje používateľovi získať obraz údajov, aký existoval v minulosti, pomocou interaktívnej správy?

6.6 Answers

1. Zostava je oblasť určená na zjednodušenie prehľadávanie a vyhľadávanie údajov.
2. Údaje je možné načítať z lokálnej databázy, SQL s podporou RESTu alebo z REST zdroja.
3. Dostupné typy zdrojov sú Tabuľka alebo SQL dotaz.
4. Hlavnou vlastnosťou klasickej zostavy je režim len na čítanie, ktorý používateľovi znemožňuje možnosti upravovať zobrazenie alebo údaje.
5. Funkcie retrospektívy umožňujú používateľovi skonštruovať obraz údajov tak, ako existoval v definovanom časovom bode.

7. Ako spravovať formuláre?

VERONIKA ŠALGOVÁ, MICHAL KVET, MIROSLAV POTOČÁR

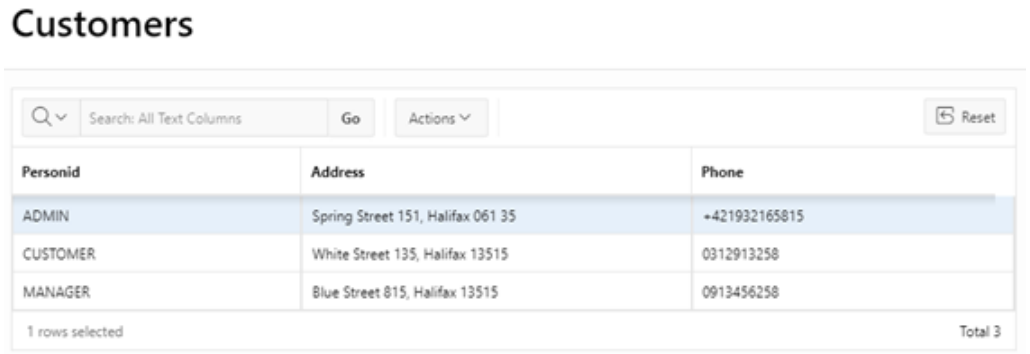
7.1 Ako spravovať formuláre

V Oracle APEX je možné použiť tri základné typy formulárov:

- editovateľnú interaktívnu mriežku,
- formulár založený na tabuľke,
- hlavný podrobný formulár.

7.2 Editovateľná interaktívna mriežka

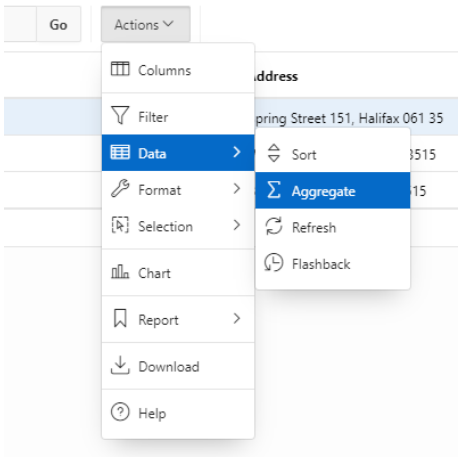
v editovateľnom interaktívnom prehľade môžete pridávať, upravovať, mazať a vyhľadávať údaje. Je to komponent bohatý na funkcie s možnosťami úpravy údajov dostupnými priamo na stránke. Okrem toho je možné túto zostavu prispôsobiť a preusporiadať interaktívne pomocou myši, čím as priamo aktualizuje štruktúra a obsah mriežky. Editovateľnú interaktívnu mriežku je možné vytvoriť konverziou interaktívnej mriežky určenej len na čítanie, ktorá bola vytvorená vybratím *Report* a potom *Interactive Grid*. Ďalšou možnosťou je vytvoriť *Report and Form* a vybrať typ reportu *Interactive Grid*, ktorý bude tiež vytvorený len ako verzia na čítanie, ale dá sa previesť do editovateľnej formy.



Obr. 7.1: Interaktívna mriežka s údajmi o zákazníkoch.

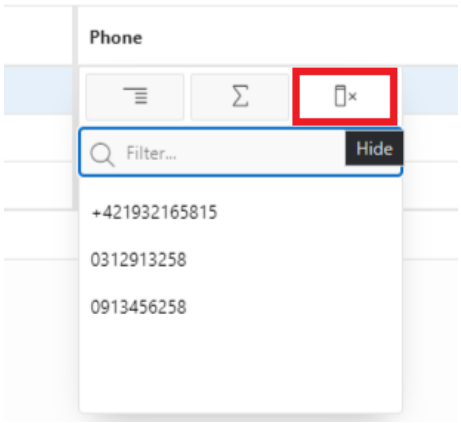
Údaje je možné vyhľadávať pomocou textu vo vyhľadávacom paneli v hornej časti stránky.

Ďalšie akcie dostupné s touto interaktívnou mriežkou sú znázornené na Obrázku 7.2.



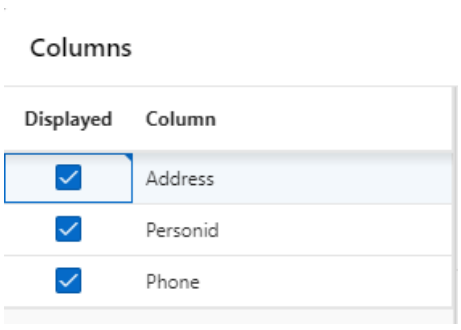
Obr. 7.2: Možnosti interaktívnej mriežky.

Veľkosť stĺpca mriežky je možné zmeniť kliknutím a podržaním okraja hlavičky stĺpca a jeho úpravou pomocou myši. Stĺpec môžete skryť aj tak, že vyberiete hlavičku a kliknete na ikonu *Hide*, ako je to znázornené na Obrázku 7.3.



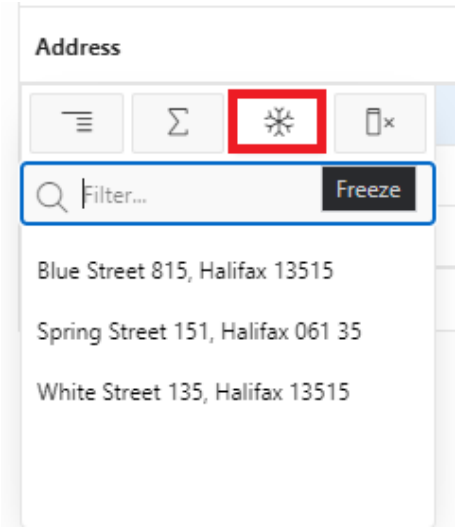
Obr. 7.3: Ikona skrytia v stĺpci Phone column.

Po skrytí, sa už stĺpec nezobrazuje v mriežke. Ak chcete stĺpec znova zobraziť, musíte kliknúť na *Actions*, vybrať možnosť *Columns* a kliknúť na stĺpce, ktoré sa majú zobraziť.



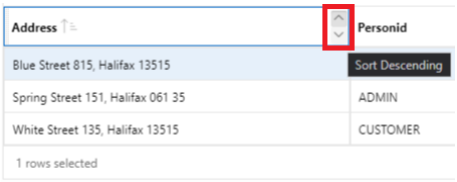
Obr. 7.4: Zobrazené stĺpce.

Stĺpec je taktiež možné zmraziť kliknutím na *Freeze* ikonu, ako je znázornené na Obrázku 7.5. Zmrazenie vylúči stĺpec z rolovateľnej oblasti.



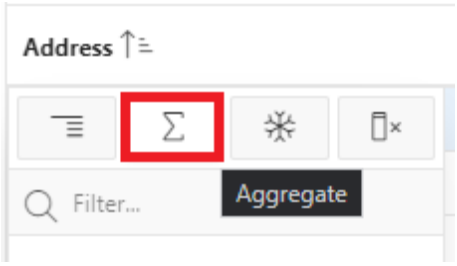
Obr. 7.5: Ikona zmrazenia v stĺpci adresa.

Údaje je možné zoradiť podľa stĺpca kliknutím na *Sort Ascending* alebo *Sort Descending* ikonu, ako je znázornené na Obrázku 7.6.



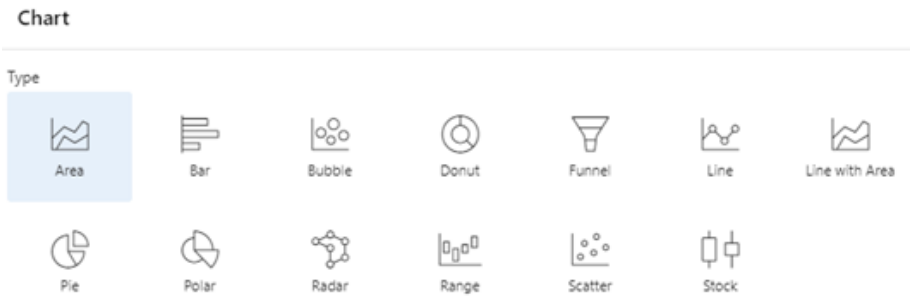
Obr. 7.6: Zoradenie stĺpca s adresami.

Použitím *Aggregate* ikony v stĺpci, ako je znázornené na Obrázku 7.7, je možné na vybranom stĺpci zvoliť funkciu Count (počet), Count Distinct (počet unikátnych hodnôt), Minimum a Maximum.



Obr. 7.7: Ikona súčtu v stĺpci adresy.

Dáta z mriežky je možné využiť na vytváranie grafov, výberom možností *Actions* a potom *Charts*. Existujú rôzne typy grafov, ako je možné vidieť na Obrázku 7.8. Je možné si vybrať z rôznych agregačných funkcií poskytovaných s grafmi.



Obr. 7.8: Typy grafov.

7.3 Formulár založený na tabuľke

Oracle APEX poskytuje základné formuláre na vloženie jedného riadku do tabuľky, ako je znázornené na Obrázku 7.9. Po kliknutí na *Create a Page* je potrebné vybrať možnosť *Form*. Je potrebné vybrať tabuľku pripojenú k vytvorenému formuláru alebo je možné napísať SQL Query a tiež primárny kľúčový stĺpec tabuľky.

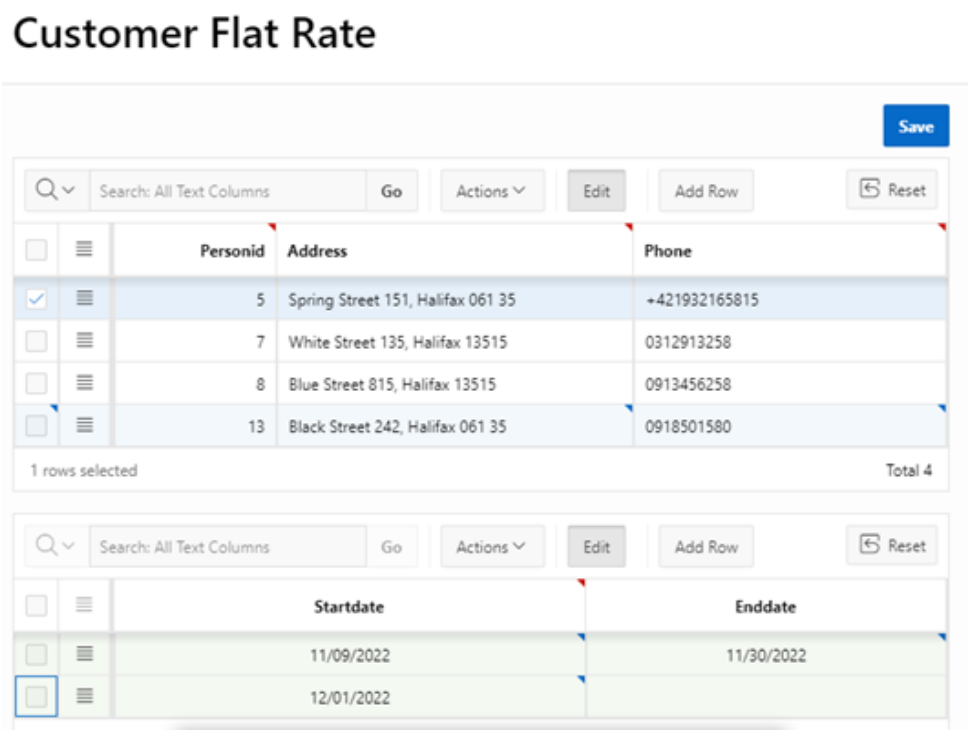


Obr. 7.9: Formulár pre vkladanie zákazníkov.

7.4 Hlavný podrobný formulár - Master detail form

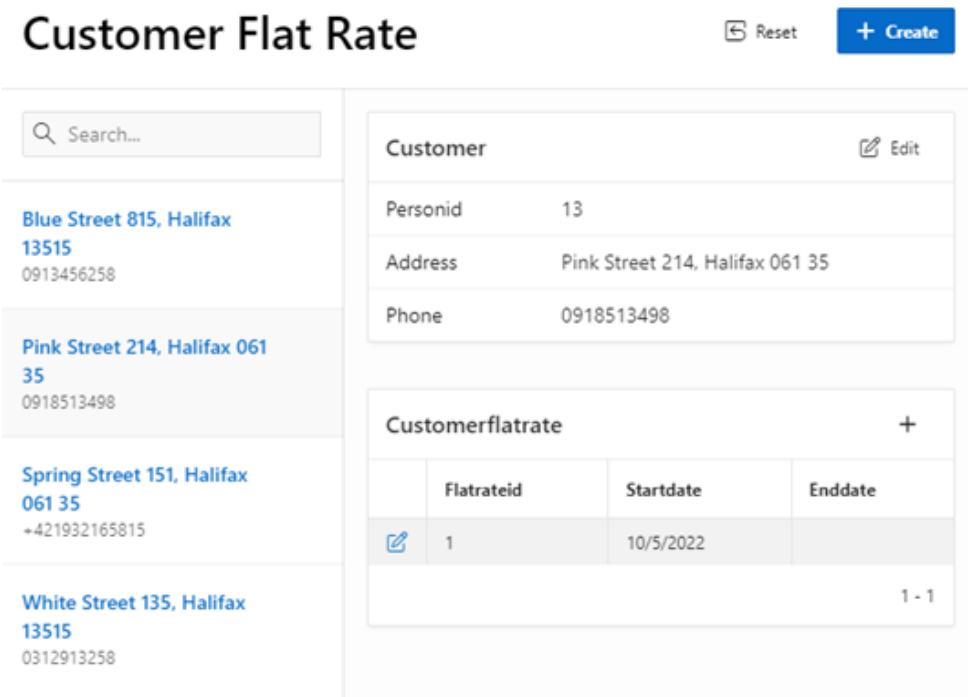
V tomto formulári je možné dopytovať, vkladať, aktualizovať a zmazávať hodnoty z dvoch súvisiach tabuliek alebo zobrazení. Po kliknutí na *Create a Page*, je potrebné zvoliť možnosť *Master Detail*. V nasledujúcom detaile si vyberáte master-detail štýl, napr. *Stacked*, *Side by Side*, alebo *Drill Down*. Tento formulár zvyčajne zobrazuje hlavný riadok a viacero podrobných riadkov. Znamená to, že si musíte vybrať tabuľku alebo pohľad ako hlavný zdroj údajov a ďalší ako zdroj podrobných údajov. Po výbere primárneho kľúča každej tabuľky je potrebné zvoliť cudzí kľúč hlavného detailu.

Obrázok 7.10 zobrazuje *Stacked* master detail form. Obsahuje upraviteľné interaktívne mriežky. Používatelia môžu vybrať riadok v hlavnej mriežke a aktualizovať mriežky s detailami.



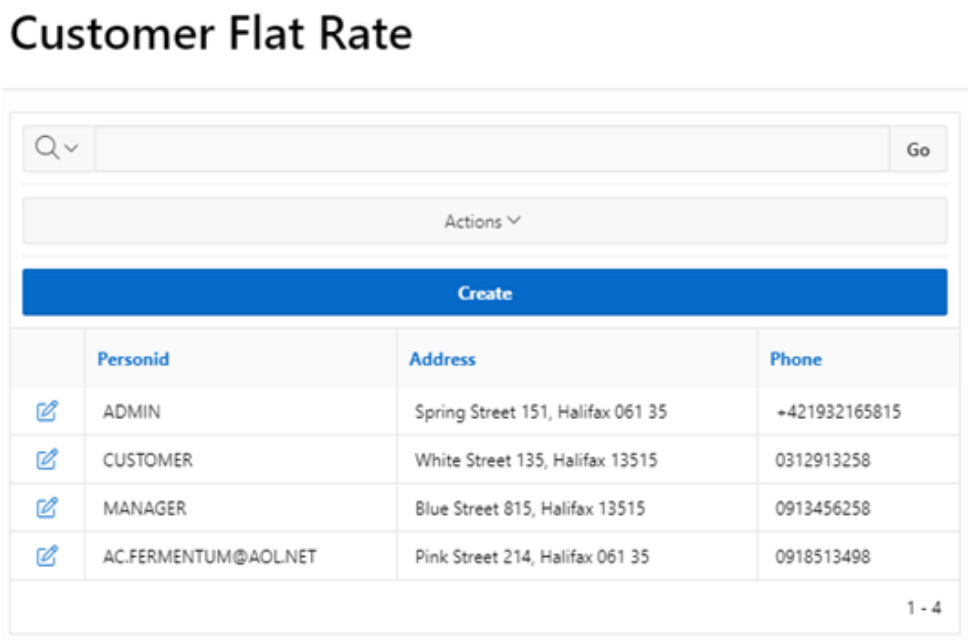
Obr. 7.10: Stacked master detail form zákazníckych paušálnych sadzieb.

Na Obrázku 7.11 je *Side by Side* master detail form obsahujúci detail predlohy jednej strany využívajúci rozloženie vedľa seba a oblasti zostavy s modálnymi oknami úprav. Na ľavej strane si môžete vybrať z hlavného zoznamu a prejsť na hlavný záznam. Pravá strana obsahuje vybraný kmeňový záznam a súvisiace podrobné správy.

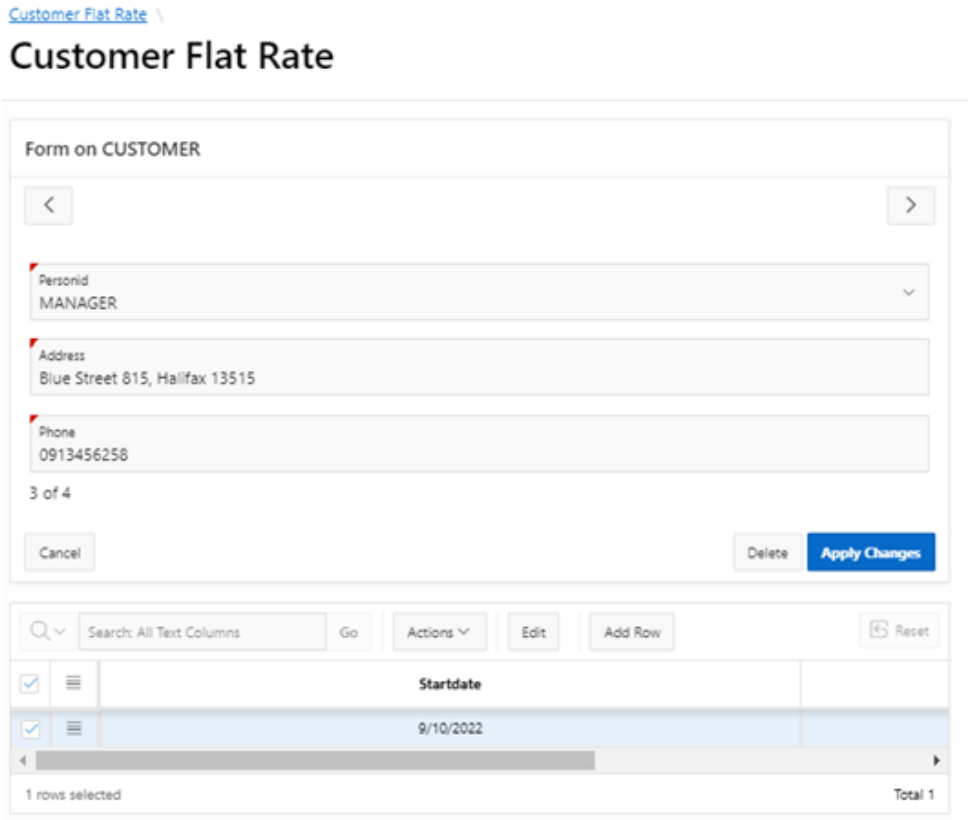


Obr. 7.11: Side by Side master detail form zákazníckych paušálnych sadzieb.

Drill Down master detail obsahuje dve strany. Prvá strana obsahuje interaktívny prehľad pre hlavnú tabuľku, ako je znázornené na Obrázku 7.12. Na druhej strane, ako je znázornené na Obrázku 7.13, je štandardný formulár a editovateľné interaktívne mriežky pre zobrazenie detailov.



Obr. 7.12: Drill Down master detail form – prvá strana.



Obr. 7.13: Drill Down master detail form – druhá strana.

7.5 Otázky

1. Je možné usporiadať dáta pomocou editovateľnej interaktívnej mriežky?
2. Ktorá definícia komponentu tabuľky musí byť zvolená pri vytváraní formulára?
3. Ktorá možnosť vylúči stĺpec z rolovateľnej oblasti?

7.6 Odpovede

1. Áno, zostavy je možné prispôsobovať a preusporiadať interaktívne pomocou myši a priamo aktualizovať štruktúru a obsah mriežky.
2. Musia byť špecifikované primárne kľúčové atribúty.
3. Stĺpec môže byť zmrazený po kliknutí na ikonu “Freeze”. Zmrazenie stĺpca ho vylúči z rolovateľnej oblasti.

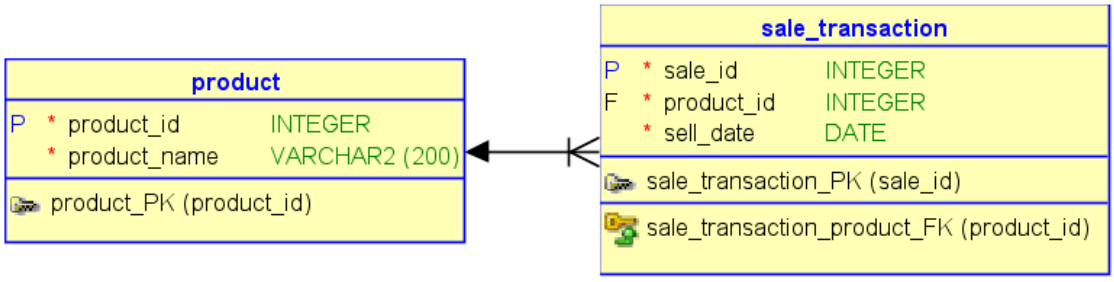
8. Ako previesť textové reporty na grafy?

IVAN PASTIERIK, MICHAL KVET, MIROSLAV POTOČÁR

8.1 Graf

Regióny **“Chart”** (Graf) v Oracle APEX sú univerzálne komponenty, ktoré umožňujú vývojárom aj používateľom získavať hlbšie pochopenie dát prostredníctvom vizualizácie. Tieto dynamické grafy, ktoré môžu mať napríklad formu stĺpcových, čiarových, alebo koláčových grafov slúžia ako dôležitý nástroj na zlepšenie používateľskej prívetivosti aplikácie a uľahčenie rozhodovania založeného na dátach.

V tejto kapitole sa dozviete, ako vytvoriť jednoduchý graf s možnosťami **filtrovania** a **triedenia**. **Filtrovanie** a **triedenie** môžu dramaticky zlepšiť interaktivitu a čitateľnosť grafov. Pre tento účel sme vytvorili jednoduchý dátový model, ako je ukázané na Obrázku 8.1. Dátový model sa skladá z dvoch entít: **“product”** a **“sale_transaction”**. Entita **“product”** reprezentuje jeden produkt, pričom jej primárny kľúč je **“product_id”** a názov produktu je uložený v atribúte **“product_name”**. Entita **“sale_transaction”** obsahuje informácie o všetkých predaných produktoch s primárnym kľúčom **“sale_id”**. Atribút **“product_id”** je cudzím kľúčom, ktorý odkazuje na atribút **“product_id”** v entite **“product”**. Atribút **“sell_date”** obsahuje dátum predaja produktu.

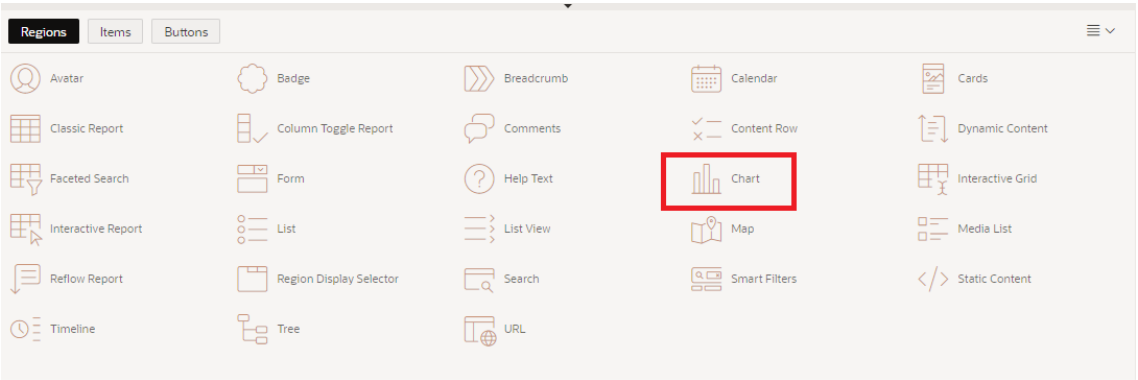


Obr. 8.1: Dátový model používaný vo vzorovej aplikácii.

Na začiatok je potrebné vytvoriť novú aplikáciu, ktorú nazveme **“Chart Showcase”**. Táto aplikácia bude obsahovať stránky **“Global Page”**, **“Home”** a **“Login Page”**. Grafy budeme vytvárať na domácej stránke **“Home”**. Podrobné pokyny na vytvorenie aplikácií boli popísané v Kapitole 5. Po vytvorení aplikácie vstúpte do “**Page Designeru”** (Návrhár stránky) pre stránku **“Home”** v našej novovytvorenej aplikácii, ktorý slúži na úpravu vzhľadu a obsahu tejto stránky.

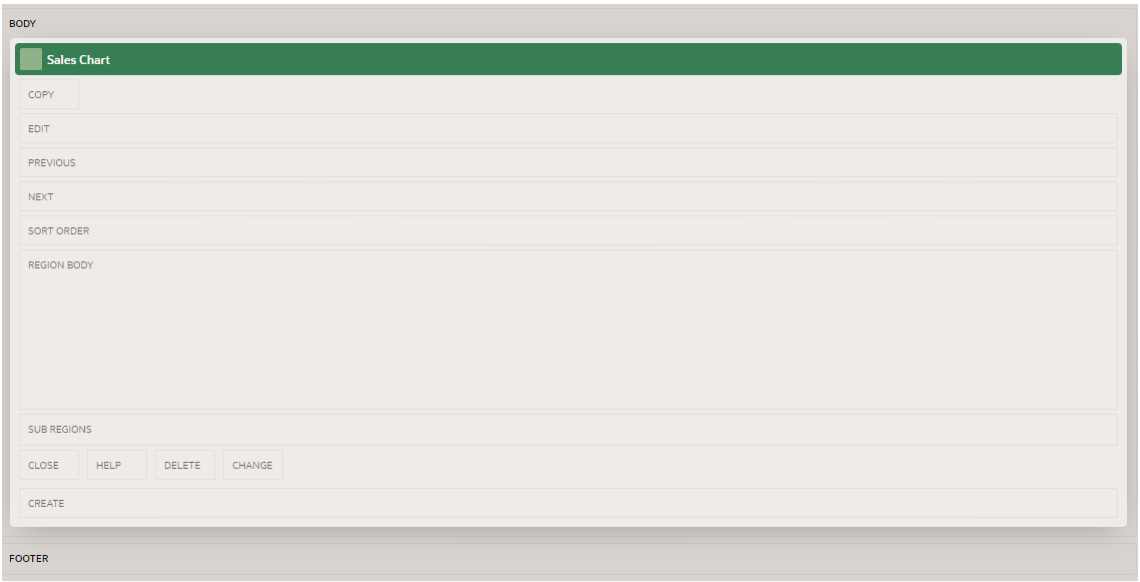
8.2 Vytvorenie stĺpcového grafu

Pre umiestnenie regiónu **“Chart”**, zvýraznenej na Obrázku 8.2, do našej aplikácie, ju musíme potiahnuť a pustiť do sekcie rozvrhnutia stránky v **“Page Designeri”**, ktorá sa nachádza v strede obrazovky.



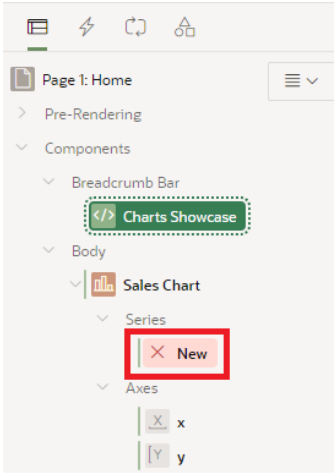
Obr. 8.2: Výber a pokladanie regiónu “Chart”.

Región **“Chart”** umiestnime do časti stránky **“Body”** (Telo), pretože chceme, aby náš graf bol hlavným obsahom našej stránky. Po umiestnení regiónu **“Chart”** nastavíme jeho názov na **“Sales Chart”** (Graf predajov). Na Obrázku 8.3 môžete vidieť, ako vyzerá región **“Chart”** po umiestnení a zmenení názvu.



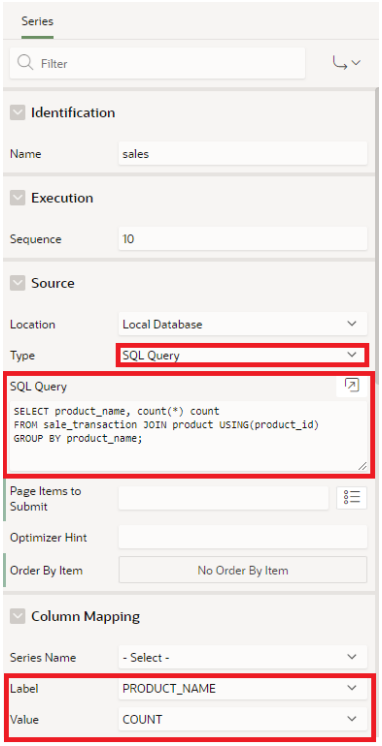
Obr. 8.3: Sekcia telo v Page Designeri po položení a premenovaní regiónu “Chart”.

V ľavej časti **“Page Designeru”** môžete vidieť prehľad všetkých umiestnených komponentov vo forme stromového diagramu. Ak sa pozriete pozorne na región **“Sales Chart”**, v časti **“Series”** (série) je hlásená chyba. Táto chyba, ako je vidieť na Obrázku 8.4, je spôsobená tým, že sme nedefinovali zdroj pre dáta, ktoré chceme zobrazovať v našom grafe. Zdroje dát sú pre zobrazenie v regióne **“Chart”** definované v sekcii **“Series”**.



Obr. 8.4: Stromový prehľad komponentov stránky s chybou v časti "Series" pod regiónom "Sales Chart".

Ak kliknete na sériu označenú ako **"New"** zo stromového prehľadu komponentov stránky, otvorí sa nastavenie pre sériu na pravej strane **"Page Designeru"**. V tomto nastavení zmeníme názov série na **"sales"**, pretože chceme zobraziť počet predajov našich jednotlivých produktov v našom **"Sales Chart"** regióne. Ďalej musíme definovať zdroj série. Aby sme toho docielili, zmeníme typ zdroja na **"SQL Query"** (SQL dotaz) a do vnútra časti **"SQL Query"** umiestnime **SELECT** príkaz, ako je ukázané na Obrázku 8.5. Tento **SELECT** príkaz vráti celkový počet predajov zoskupených podľa názvov produktov. Teraz, keď sme definovali zdroj pre dáta, ktoré chceme zobraziť v grafe, musíme namapovať stĺpce vybrané **SELECT** príkazom na osi **"Label"** (Štítok predstavujúci os **x**) a **"Value"** (Hodnota predstavujúca os **y**). Konečná konfigurácia našej série je ukázaná na Obrázku 8.5 s dôležitými zmenami zvýraznenými v červenom rámčeku.



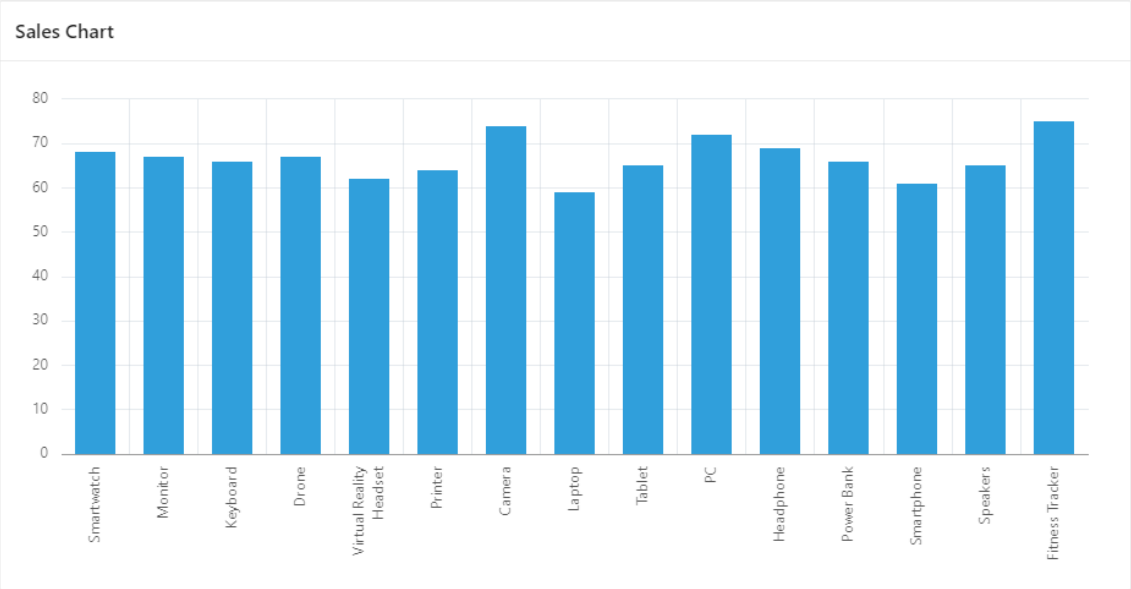
Obr. 8.5: Konečná konfigurácia série "sales".

Ak zoberieme SQL dotaz z Obrázku 8.5 a spustíme ho, vráti výstup, ktorý je zobrazený na Obrázku 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

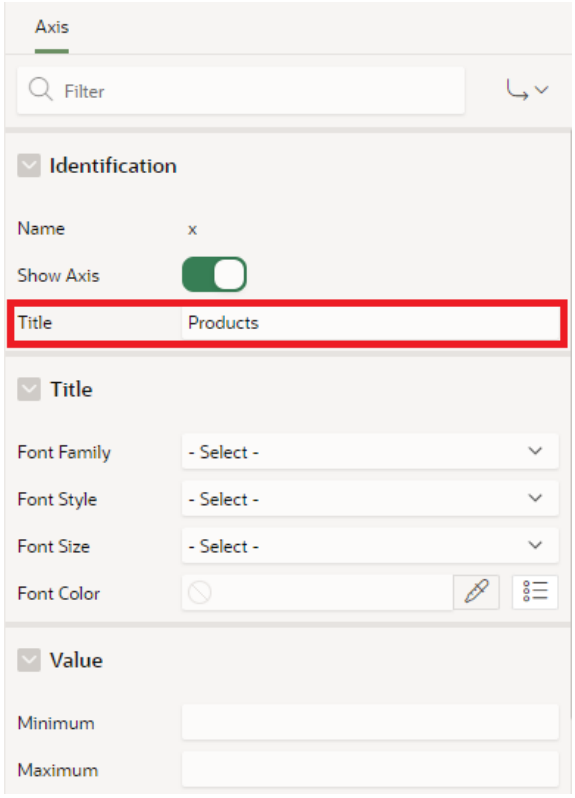
Obr. 8.6: Dáta vrátené vykonaním SQL dotazu.

Teraz môžeme uložiť našu stránku a spustiť ju. Pred spustením stránky nastavíme atribút stránky **“Authentication”** na **“Page Is Public”**, takže stránka nevyžaduje prihlásenie používateľa. Keď spustíte stránku, mali by ste dostať výsledky podobné tým na Obrázku 8.7. Môžete však pozorovať, že náš graf nemá pomenované osi **x** a **y**, čo ho robí ťažšie čitateľným.



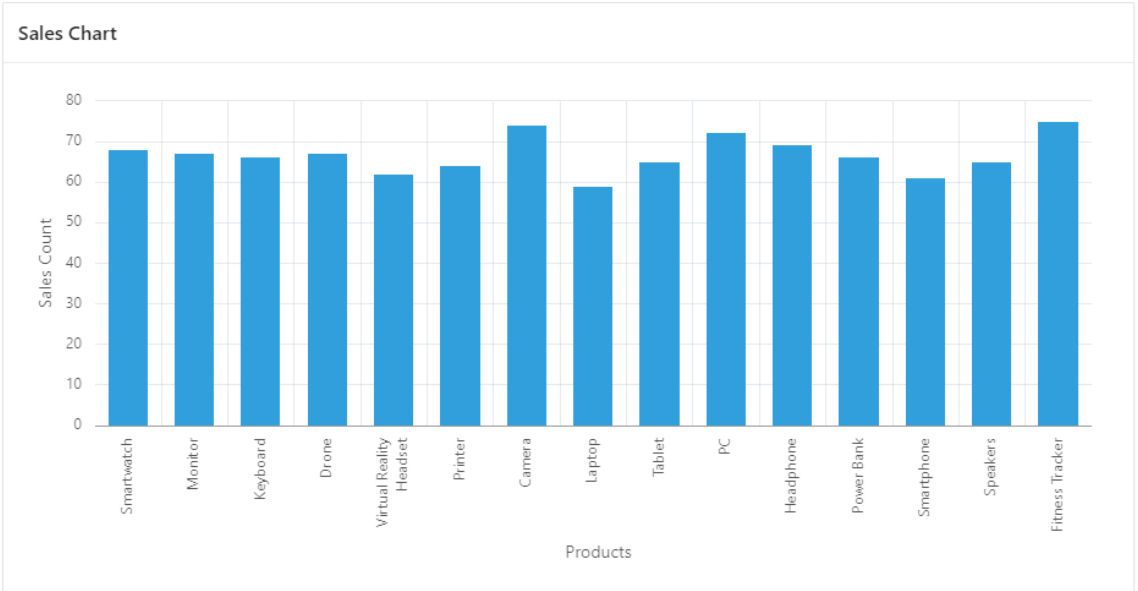
Obr. 8.7: Graf zobrazený po spustení aplikácie.

Pre vyriešenie problému s nepomenovanými osami **x** a **y** môžeme naše osi pomenovať podobne, ako sme pomenovali sériu. Kliknutím na jednotlivé osi v stromovom prehľade komponentov stránky a zmenou ich atribútu **“title”** (názov) môžeme osi pomenovať. Os **x** nazveme **“Products”** (Produkty) a os **y** nazveme **“Sales Count”** (Počet predajov).



Obr. 8.8: Zmena názvu osi x.

Po označení oboch osí náš graf bude vyzerať podobne ako graf na Obrázku 8.9.

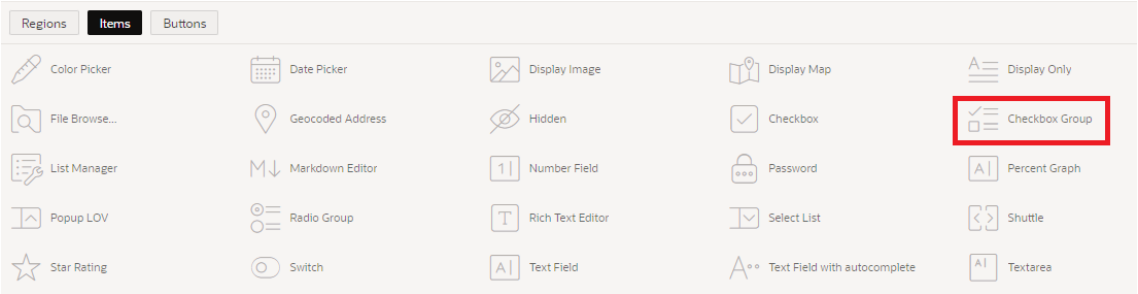


Obr. 8.9: Graf po pomenovaní osí.

8.3 Pridanie filtrovania pre stĺpcový graf

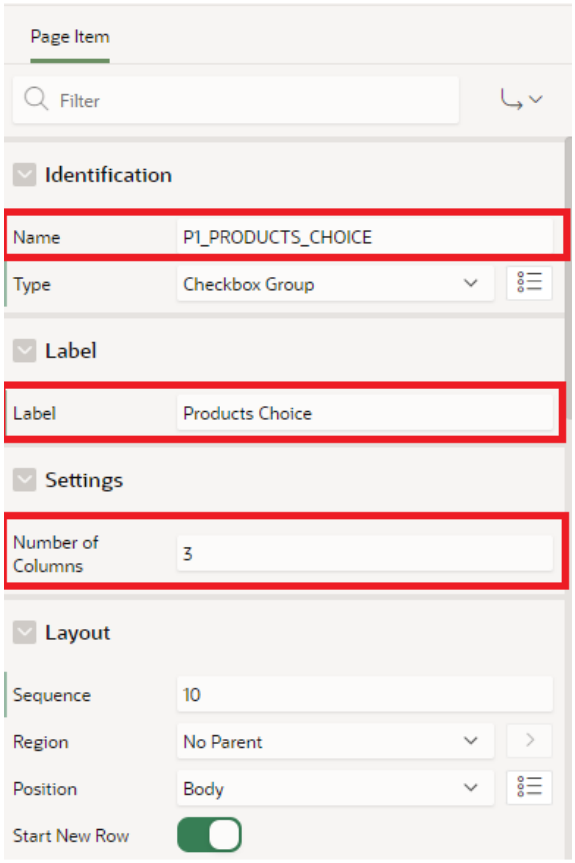
Teraz chceme pridať **filtrovanie** produktov do nášho grafu, aby sme mohli vybrať ktoré produkty sa majú zobraziť na grafe. Pre tento účel pridáme predmet **“Checkbox Group”** (Skupina zaškrtávacích políčok) zo sekcie **“Items”** (Predmety) z dolného panela nástrojov **“Page Designeru”** a umiestnime

ho nad náš región s názvom **“Sales Chart”** v sekcii **“Body”**.



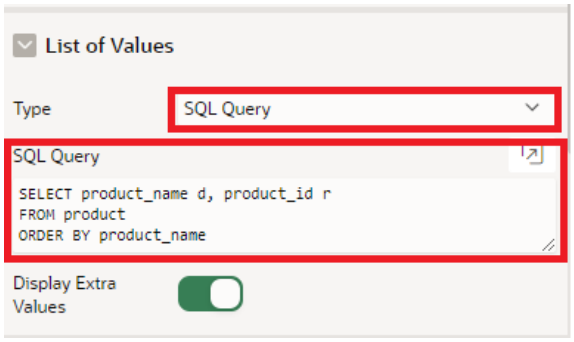
Obr. 8.10: Výber a pokladanie predmetu “Checkbox Group”.

Predmet **“Checkbox Group”** pomenujeme **“P1_PRODUCTS_CHOICE”** a nastavíme jeho atribút **“Label”** (štítok) na hodnotu **“Products Choice”** (Voľba produktov). Okrem toho môžeme upraviť rozloženie tohto predmetu tak, aby sa produkty zobrazovali v troch stĺpcoch zmenou hodnoty atribútu **“Number of Columns”** (Počet stĺpcov) tak, ako je ukázané na Obrázku 8.11.



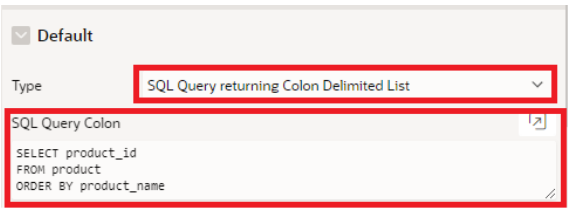
Obr. 8.11: Nastavenie základných atribútov predmetu “Checkbox Group”.

Pre naplnenie nášho predmetu **“Checkbox Group”** produktami musíme definovať zoznam hodnôt. V sekcii **“List of Values”** (Zoznam hodnôt) zmeníme atribút **“Type”** (Typ) na hodnotu **“SQL Query”** a napíšeme **SELECT** príkaz, ktorý je ukázaný na Obrázku 8.12. Tento príkaz musí vybrať presne dva stĺpce: prvý stĺpec obsahujúci zobrazené hodnoty (s aliasom **“d”**) a druhý stĺpec obsahujúci návratové hodnoty (s aliasom **“r”**).



Obr. 8.12: Nastavenie zoznamu hodnôt pre predmet "Checkbox Group".

Predmet **"Checkbox Group"** má predvolené pri spustení aplikácie všetky políčka nezaškrtnuté. Avšak pre zlepšenie používateľskej prívetivosti, keď je predmet **"Checkbox Group"** používaný na filtrovanie regiónu **"Chart"**, tak je vhodné, aby používatelia po spustení aplikácie videli zobrazené všetky produkty na grafe. Na dosiahnutie tohto efektu prejdeme do sekcie **"Default"** (Predvolený) v nastaveniach predmetu **"Checkbox Group"** a zmeníme atribút **"Type"** na **"SQL Query returning Colon Delimited List"** (SQL dotaz vracajúci dvojbodkou oddelený zoznam). Tento typ SQL dotazu nám umožňuje definovať, ktoré hodnoty sú predvolené vybrané. Bežný SQL dotaz nemôže byť použitý s predmetom **"Checkbox Group"**, pretože môže definovať iba jednu predvolenú hodnotu, pričom my potrebujeme definovať viacero predvolených hodnôt. V sekcii **"SQL Query Colon"** napíšeme **SELECT** príkaz, ukázaný na Obrázku 8.13. Tento **SELECT** príkaz musí vrátiť práve jeden stĺpec, ktorý bude interne konvertovaný na **dvojbodkou oddelený zoznam**, ktorý by mohol vyzerať napríklad takto: **"1:5:7:2:3"**.



Obr. 8.13: Nastavenie predvolených hodnôt pre predmet "Checkbox Group".

Teraz, keď je predmet **"Checkbox Group"** nakonfigurovaný podľa našich predstáv, môžeme ho prepojiť s regiónom **"Sales Chart"**. Aby všetko správne fungovalo, tak pridáme názov predmetu **"Checkbox Group"** do ako hodnotu atribútu **"Page Items to Submit"** (Predmety stránky na odoslanie) v sekcii **"Source"** (Zdroj) našej série **"sales"** vo vnútri regiónu **"Sales Chart"**. Je potrebné pridať do **"Page Items to Submit"** každú položku, ktorá je použitá v SQL dotaze. Návratová hodnota predmetu **"P1_PRODUCTS_CHOICE"** je **dvojbodkou oddelený zoznam**, ktorý pozostáva z produktov, ktoré sú zaškrtnuté. Preto musíme použiť funkciu **INSTR**, ktorá vracia index prvého výskytu podreťazca, definovaného v druhom parametri funkcie, v reťazci definovanom v prvom parametri funkcie.

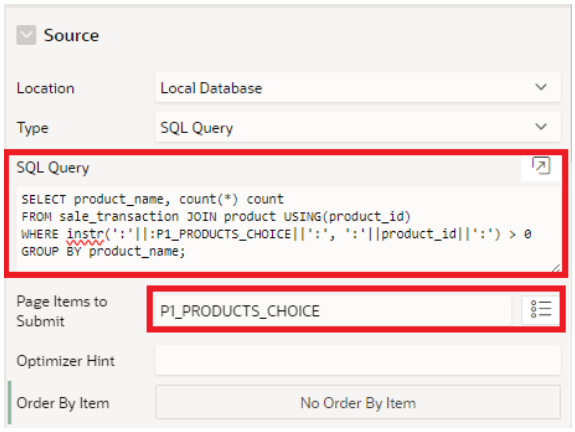
Funkcia **INSTR** vráti hodnotu **0**, ak sa v reťazci nenachádza žiadny výskyt podreťazca. Funkciu **INSTR** použijeme na kontrolu, či je **"product_id"** každého produktu prítomné v zozname vybraných produktov v **"P1_PRODUCTS_CHOICE"**. Funkcia **INSTR** vyhľadáva výskyt podreťazca **':'||product_id||':'** v reťazci **':'||:P1_PRODUCTS_CHOICE||':'**. Ak nájde zhodu, znamená to, že produkt je medzi vybranými produktmi.

Musíme pridať **':'** k **"product_id"** a **":P1_PRODUCTS_CHOICE"**, pretože chceme vykonať presnú zhodu medzi jednotlivými hodnotami **"product_id"**. Týmto spôsobom zabezpečíme, že napríklad **":1:"** sa určite nachádza v reťazci **":1:2:3:8:9:11:"**. Vďaka tomu si môžeme byť istý,

že hľadáme presné zhody a každá hodnota **“product_id”** je posudzovaná nezávisle od ostatných hodnôt.

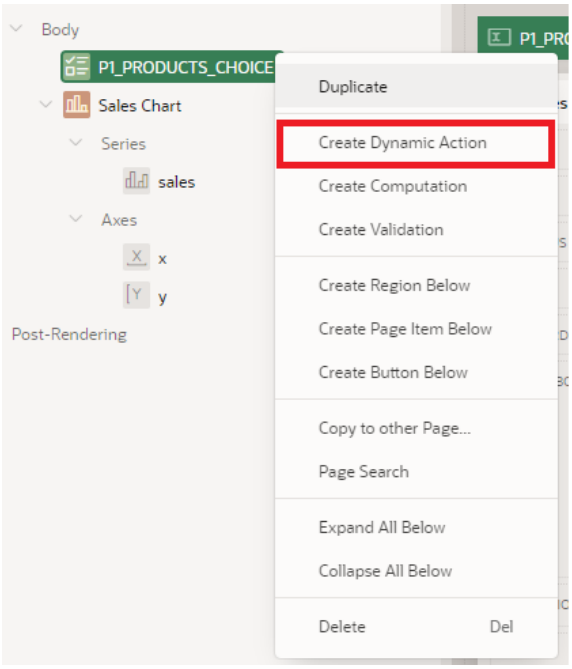
Ak by sme hľadali výskyt podreťazca **“1”** v reťazci **“1:2:3:8:9:11”**, tak by sme mohli zobraziť produkt s ID **“1”** pri zaškrtnutí políčok produktov, ktoré majú ID **“1”**, **“11”**, alebo ľubovoľné iné ID, ktoré v sebe obsahuje hodnotu **“1”**. Čiže pri zaškrtnutí produktu s ID **“11”** by sa v našom prípade nezobrazil len ten daný produkt, ktorý sme zaškrtli, ale zobrazil by sa aj produkt s ID **“1”**, čo vedie k nežiaducim výsledkom a nepresnej reprezentácii dát.

Pridaním **’:’** na oboch stranách ID vytvoríme konzistentný vzor oddeľovača, ktorý zabezpečuje, že zhoda hodnôt **“product_id”** je vždy presná a tým pádom sa odstránia akékoľvek problémy, ktoré môžu nastať pri porovnávaní. Tým zabezpečíme, že graf zobrazí dáta presne podľa výberu používateľa.



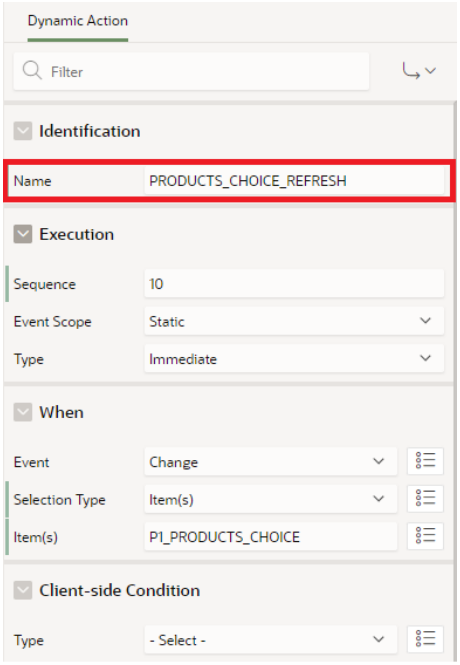
Obr. 8.14: Prepojenie regiónu “Sales Chart” s predmetom “Checkbox Group”.

Posledným krokom konfigurácie je nastavenie obnovy nášho regiónu **“Sales Chart”** pri zmene výberu predmetu **“Checkbox Group”**. Preto musíme vytvoriť **“Dynamic Action”** (Dynamická akcia) pre predmet **“Checkbox Group”** s názvom **“P1_PRODUCTS_CHOICE”** tak, ako je ukázané na Obrázku 8.15.



Obr. 8.15: Vytvorenie dynamickej akcie pre predmet s názvom “P1_PRODUCTS_CHOICE”.

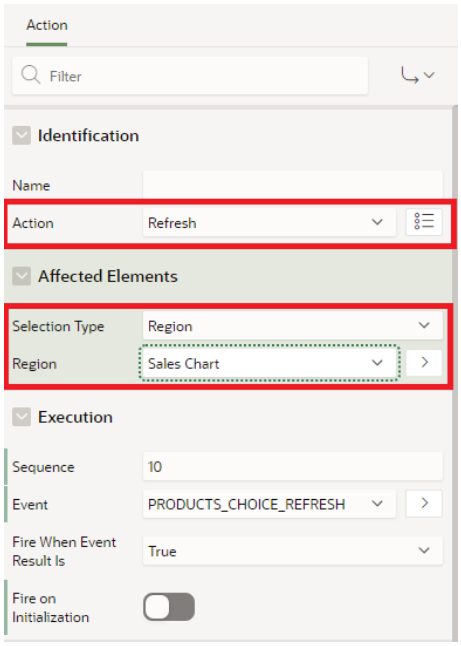
Túto dynamickú akciu môžeme pomenovať **“PRODUCTS_CHOICE_REFRESH”**, ako je vidieť na Obrázku 8.16. Je potrebné sa tiež uistiť, že atribút **“Event”** (Udalosť) v sekcii **“When”** (Kedy) je nastavený na hodnotu **“Change”** (Zmena) a že atribút **“Selection Type”** (Typ výberu) je nastavený na hodnotu **“Item(s)”** (Predmet(y)) s atribútom **“Item(s)”** nastaveným na hodnotu mena predmetu, pre ktorý túto dynamickú akciu vytvárame. Týmto spôsobom bude dynamická akcia spúšťaná, keď sa zmení hodnota predmetu **“P1_PRODUCTS_CHOICE”**.



Obr. 8.16: Nastavenie základných atribútov dynamickej akcie vykonanej pri zmene.

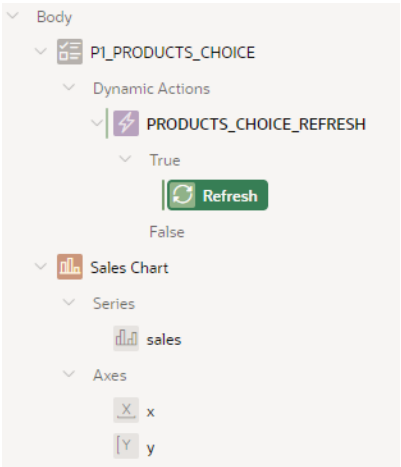
V stromovom prehľade komponentov stránky na ľavej strane **“Page Designeru”** kliknite

na akciu, ktorá sa vykoná, keď je podmienka dynamickej akcie pravdivá. Táto podmienka je pravdivá vtedy, keď sa zmení výber predmetu **“Checkbox Group”**. Pri vytvorení dynamickej akcie, uvidíte predvolenú akciu “Show” s chybou, takže na ňu kliknite. V nastaveniach tejto akcie nastavte atribút **“Action”** (Akcia) v sekcii **“Identification”** (Identifikácia) na hodnotu **“Refresh”** (Obnoviť). V sekcii **“Affected Elements”** (Ovplyvnené prvky) zmeňte **“Selection Type”** (Typ výberu) na hodnotu **“Region”** a potom nastavte **“Region”** na hodnotu **“Sales Chart”**, ako je vidieť na Obrázku 8.17.



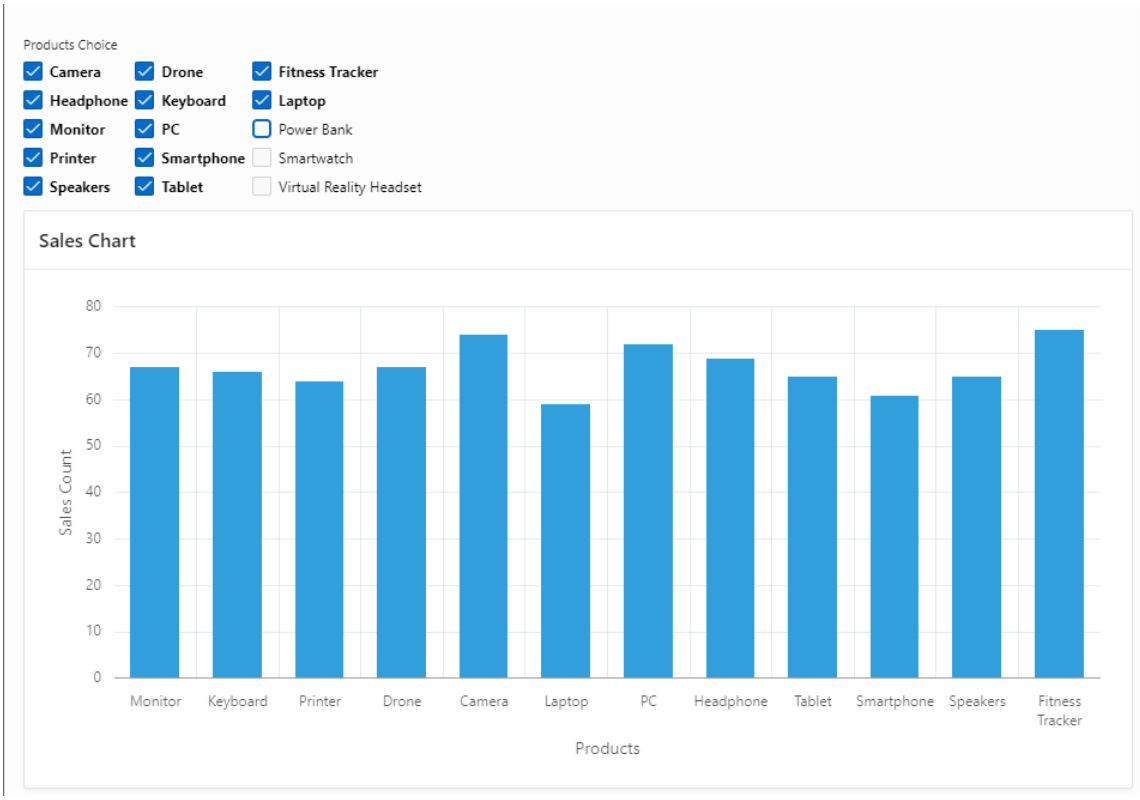
Obr. 8.17: Nastavenie atribútov akcie obnovenia.

Obrázok 8.18 ukazuje, ako by mal vyzerať stromový prehľad komponentov stránky v ľavej časti **“Page Designeru”**, ak ste všetky kroky dodržali správne.



Obr. 8.18: Dynamická akcia pre obnovenie regiónu “Sales Chart” v stromovom prehľade komponentov stránky.

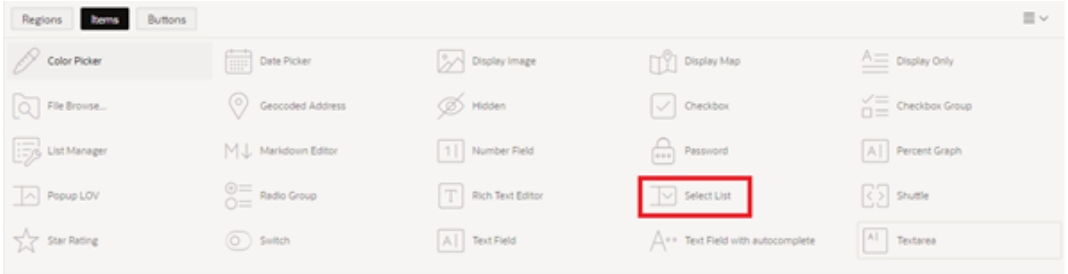
Keď spustíme a otestujeme našu aplikáciu, **filtrovanie** a **obnovenie** nášho grafu by mali byť plne funkčné, a aplikácia by mala pripomínať tú na Obrázku 8.19.



Obr. 8.19: Aplikácia po implementovaní filtrovania produktov.

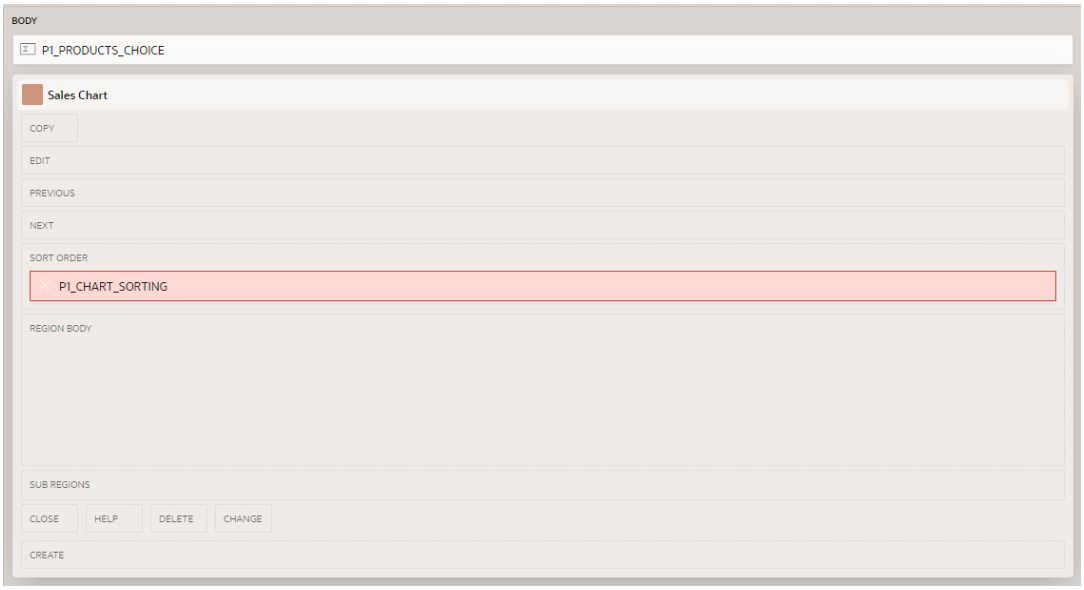
8.4 Pridanie triedenia stĺpcového grafu

Používateľskú prívetivosť aplikácie môžeme ešte zlepšiť pridaním možnosti **triedenia** hodnôt zobrazených v grafe podľa rôznych kritérií. Na výber kritérií triedenia použijeme predmet **“Select List”** (Výber zo zoznamu), ako je to vidieť na Obrázku 8.20.



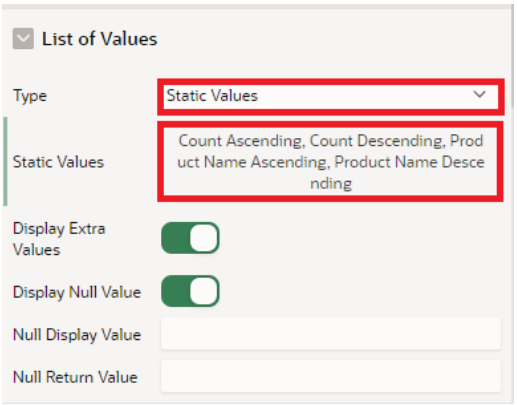
Obr. 8.20: Výber a pokladanie predmetu “Select List”.

Predmet **“Select List”** umiestnime do nášho regiónu **“Sales Chart”** v časti **“Sort Order”** (Poradie triedenia) a nazveme ju **“P1_CHART_SORTING”**, ako je to vidieť na Obrázku 8.21.



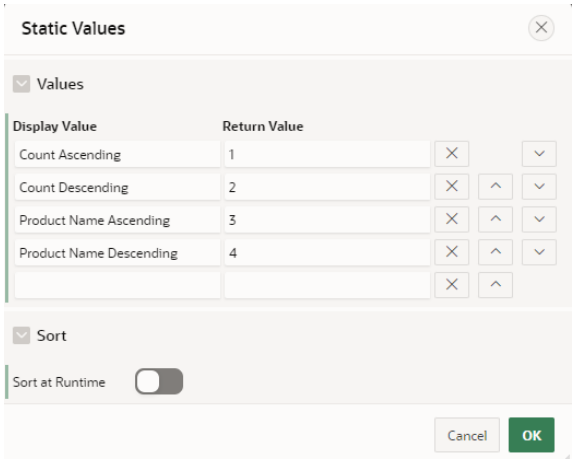
Obr. 8.21: Región "Sales Chart" po položení a premenovaní predmetu "Select List".

Teraz musíme definovať zoznam hodnôt pre náš položený predmet **"Select List"**. V nastaveniach predmetu **"P1_CHART_SORTING"** musíme zmeniť atribút **"Type"** v časti **"List of Values"** na hodnotu **"Static Values"** (Statické hodnoty). Potom musíme stlačiť tlačidlo zobrazujúce predvolené hodnoty **"Display1, Display2"** v rámci atribútu **"Static Values"**.



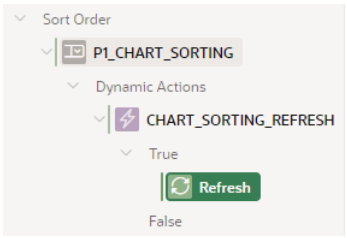
Obr. 8.22: Nastavenie zoznamu hodnôt pre predmet "Select List".

Následne sa otvorí nové okno s názvom **"Static Values"**. V tomto okne musíme definovať hodnoty zobrazenia a návratu. Pod hodnotami **zobrazenia** (Display Value) napíšeme popisy triedenia a pod hodnotami **návratu** (Return Value) napíšeme čísla začínajúce od 1, ako je to vidieť na Obrázku 8.23. Ak chceme, aby sa tieto hodnoty zobrazenia zoradili tak, ako sme ich zadali od vrchu nadol, musíme zakázať možnosť **"Sort at Runtime"** (Triediť počas behu).



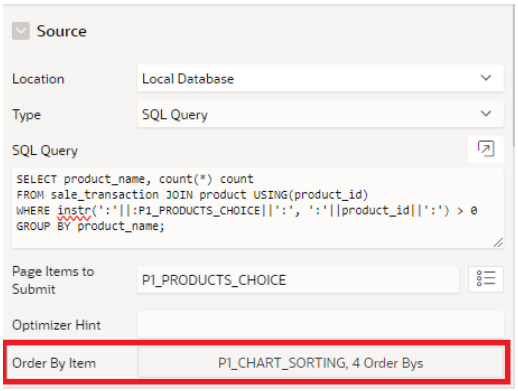
Obr. 8.23: Nastavenie statických hodnôt pre predmet “Select List”.

Aby všetko fungovalo, tak musíme pridať dynamickú akciu **“Sales Chart refresh”** rovnakým spôsobom, ako sme ju pridali a nastavili pre predmet **“P1_PRODUCTS_CHOICE”**. Výsledok je možné vidieť na Obrázku 8.24.



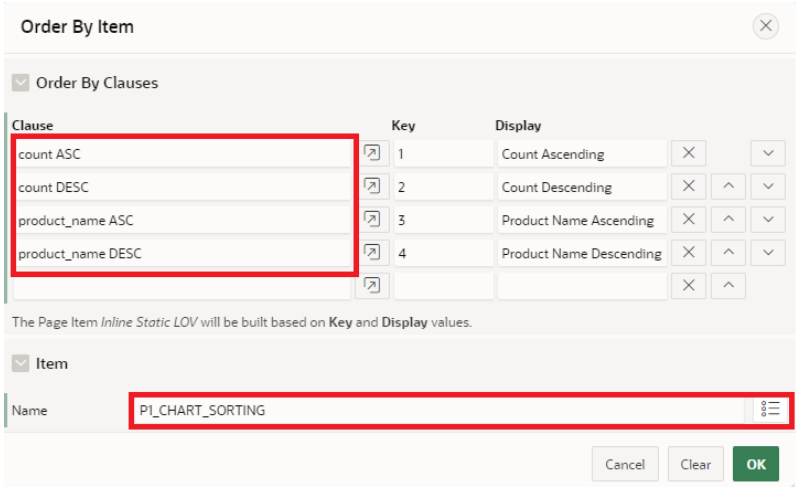
Obr. 8.24: Pridanie dynamickej akcie obnovenia pre predmet “Select List”.

Teraz, keď sme nakonfigurovali predmet **“Select List”**, musíme ho prepojiť s regiónom **“Sales Chart”**. To môžeme urobiť stlačením tlačidla hneď vedľa atribútu **“Order By Item”**, ktoré má predvolenú hodnotu **“No Order By Item”**, ako je to zvýraznené na Obrázku 8.25.



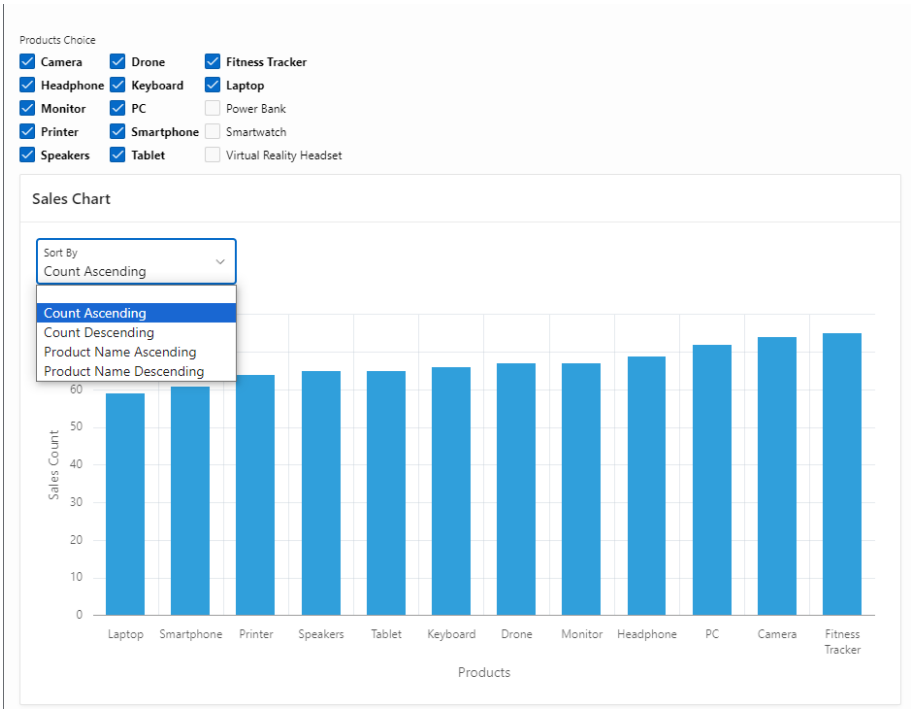
Obr. 8.25: Nastavenie atribútu “Order By Item” pre región “Sales Chart”.

Po stlačení tlačidla uvidíte, že sa otvorilo nové okno s názvom **“Order By Item”**. V tomto okne najskôr musíme nastaviť atribút **“Name”** v sekcii **“Item”** na hodnotu **“P1_CHART_SORTING”**. Potom musíme vyplniť stĺpec **“Clause”** (Klauzula) ako je to vidieť na Obrázku 8.26. Tieto klauzuly sú prakticky **“ORDER BY”** klauzuly používané v príkazoch **SELECT**, ale bez použitia kľúčového slova **“ORDER BY”**.



Obr. 8.26: Nastavenie "Order By" klauzúl používaných pre triedenie na základe predmetu "P1_CHART_SORTING".

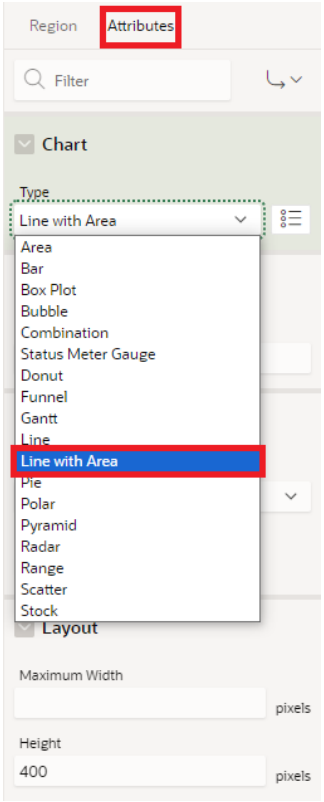
Ak ste tieto kroky dodržali podrobne, **triedenie** a **obnovenie** grafu by malo fungovať po uložení a spustení stránky aplikácie.



Obr. 8.27: Aplikácia po implementovaní triedenia grafu.

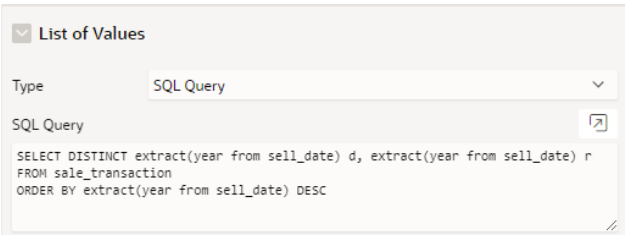
8.5 Vytvorenie iných typov grafov

Ak chcete implementovať iný typ grafu, môžete to urobiť tak, že vyberiete región **"Graph"** a prejdete do karty **"Attributes"** v pravej časti **"Page Designeru"** a zmeníte hodnotu atribútu **"Type"**. Vytvoríme ešte jeden graf typu **"Line with Area"** (Čiarový graf s plochou), ktorý ukazuje mesačný počet predajov, a nazveme ho **"Monthly Sales Chart"** (Graf mesačných predajov). Tento graf budeme môcť dodatočne filtrovať podľa roku.



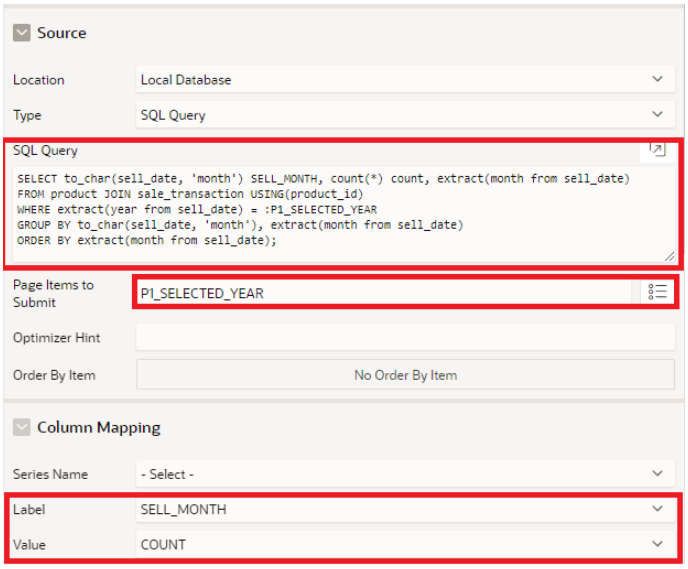
Obr. 8.28: Zmenenie typu grafu na “Line with Area”.

Ďalším krokom je pridanie predmetu **“Select List”**, ktorý použijeme na filtrovanie grafu podľa roka. Tento predmet umiestnime do časti **“Edit”** regiónu **“Monthly Sales Chart”**. SQL dotaz použitý na získanie zoznamu hodnôt je zobrazený na Obrázku 8.29.



Obr. 8.29: Zoznam hodnôt pre predmet “Select List” použitý pre filtrovanie predajov produktov podľa roku.

Po príprave predmetu **“Select List”**, ktorý budeme používať na filtrovanie, môžeme napísať SQL dotaz zdroja pre sériu v regióne **“Monthly Sales Chart”**. SQL dotaz, ktorý použijeme, je zobrazený na Obrázku 8.30. Taktiež budeme mapovať **“Label”** na **“SELL_MONTH”** (Mesiac predaja) a **“Value”** na **“COUNT”** (Počet). Hodnoty budeme tiež zoraďovať podľa čísla mesiaca vzostupne, pričom na osi **x** budeme zobrazovať plné názvy mesiacov. Názov osi **x** nastavíme na hodnotu **“Month”** (Mesiac) a hodnotu osi **y** na hodnotu **“Sales Count”** (Počet predajov).



Obr. 8.30: Nastavenie zdrojového SQL dotazu a mapovania stĺpcov pre región "Monthly Sales Chart".

Ak ste všetko nakonfigurovali správne, váš graf by mal pripomínať ten na Obrázku 8.31.



Obr. 8.31: Graf mesačných predajov.

8.6 Otázky

- Aký je hlavný účel regiónov "Chart" v Oracle APEX?
- Aké typy grafov je možné vytvoriť pomocou regiónov "Chart" v Oracle APEX?
- Ako môžeme obnoviť región "Chart" v Oracle APEX, keď dochádza k interakciám používateľa, ako je napríklad filtrovanie alebo triedenie?
- Čo robí funkcia INSTR a prečo sa používa pri regiónoch "Chart" v Oracle APEX?

8.7 Odpovede

- Hlavným účelom regiónov “Chart” v Oracle APEX je vizuálne reprezentovať dáta, čo umožňuje používateľom získavať poznatky a robiť rozhodnutia založené na dátach.
- Regióny “Chart” v Oracle APEX podporujú rôzne typy grafov, vrátane stĺpcových grafov, čiarových grafov, koláčových grafov a ďalších.
- Môžeme nastaviť dynamickú akciu na obnovenie regiónu “Chart” v Oracle APEX, keď sa menia dáta v dôsledku interakcií používateľa, ako napríklad filtrovanie alebo triedenie.
- Funkcia INSTR sa používa na nájdenie indexu prvého výskytu podreťazca v reťazci. Pri regiónoch “Chart” v Oracle APEX sa používa na kontrolu, či sa konkrétne hodnoty zhodujú s vybranými kritériami, napríklad s hodnotami ID produktov a na základe toho sa určí, ktoré produkty sa majú zobraziť na grafe.

9. Ako spravovať menu?

VERONIKA ŠALGOVÁ, MICHAL KVET, MIROSLAV POTOČÁR

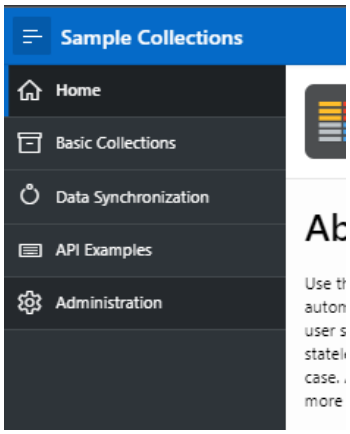
9.1 Ako spravovať menu

Navigačné menu sa vytvorí automaticky po vytvorení aplikácie. Pozostáva z položiek zoznamu odkazujúcich na stránky aplikácie. Upravením atribútov navigačného menu na stránke používateľského rozhrania (User Interface) je možné zmeniť, ako a kde sa zobrazí navigačné menu. Typy navigačného menu dostupné v Oracle APEX sú:

- Side Menu (bočné menu),
- Top Menu (horné menu),
- Mega Menu.

9.2 Side Menu

Tento typ menu je možné rozbaliť alebo zbaliť kliknutím na menu ikonu v hlavičke. Navigačné položky sa zobrazujú pomocou stromovej zložky. Podpoložky možno tiež rozbaliť alebo zbaliť. Môžeme vidieť celé menu alebo ho zbaliť na úzky panel s ikonami.

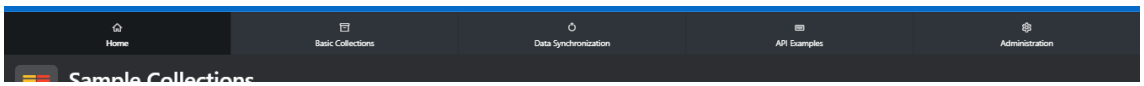


Obr. 9.1: Rozšírené bočné navigačné menu

Je tiež možné nakonfigurovať, aby bolo navigačné menu úplne skryté, čo je predvolená možnosť. Po zbalení možno pomocou možnosti Template Options zobraziť úzky panel s ikonami.

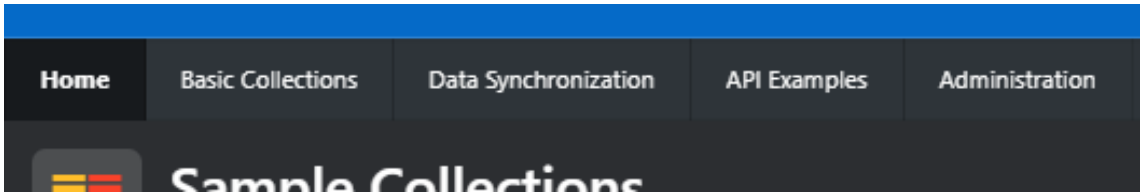
9.3 Top Menu

Horné navigačné menu sa zobrazí v hornej časti aplikácie. Oracle APEX poskytuje dve šablóny na zobrazenie menu, ako napríklad *Top Navigation Tabs* (Horné navigačné záložky) a *Top Navigation Menu* (Horné navigačné menu). Prvý typ je vhodný pre jednoduché aplikácie s niekoľkými záložkami. Je znázornený na Obrázku 9.2.



Obr. 9.2: Top Navigation Tabs

Horné navigačné menu poskytuje panel s menu. Je ideálny pre zložitejšie aplikácie s niekoľkými vrstvami hierarchie v rámci navigácie. Je znázornené na Obrázku 9.3.

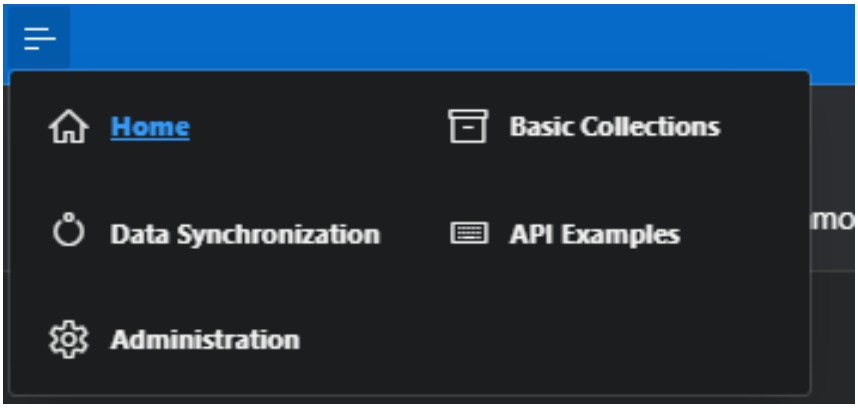


Obr. 9.3: Top Navigation Menu

Šablóna horných navigačných záložiek sa automaticky umiestňuje do spodnej časti obrazovky pre malé obrazovky alebo mobilné zariadenia. Horné navigačné menu sa najviac podobá desktopovým aplikáciám.

9.4 Mega Menu

Tento typ menu sa zobrazuje ako vyskakovací panel, ktorý je možné otvoriť alebo zatvoriť pomocou ikony v hlavičke. Mega menu sú užitočné na zobrazenie všetkých položiek navigácie naraz.



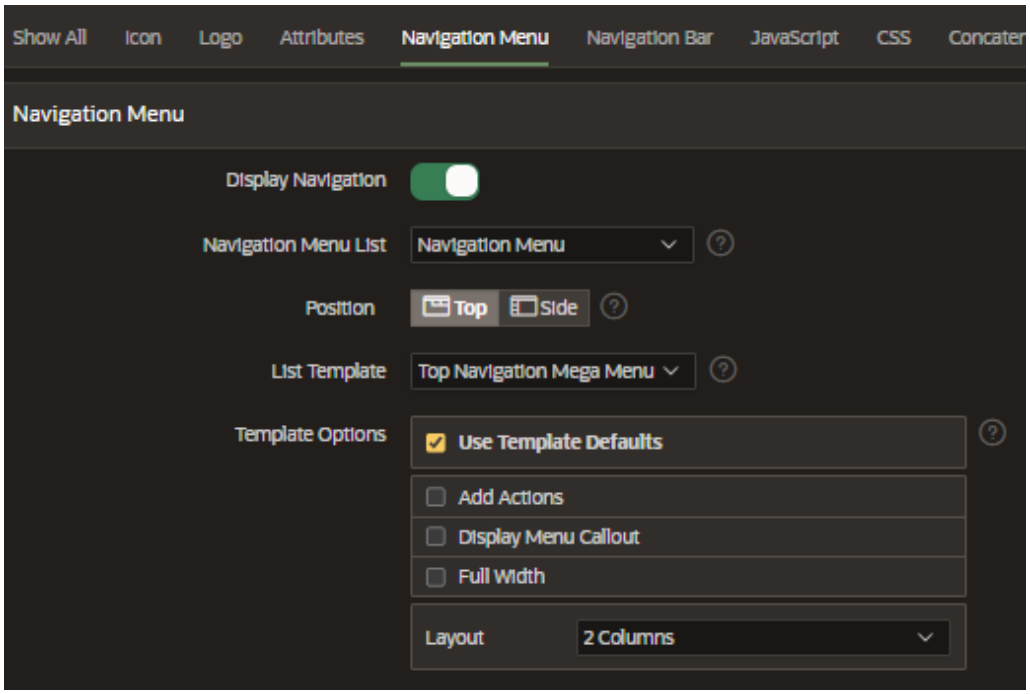
Obr. 9.4: Top Navigation Mega Menu

Okrem toho je možné v menu zvoliť počet stĺpcov.

9.5 Úprava menu zoznamov

Jednotlivé položky menu tvoria zoznam, ktorý je možné vytvárať a upravovať podobne ako akýkoľvek iný zoznam. Je to prístupné v zdieľaných komponentoch (*Shared Components*) v časti

navigácie (Navigation region). Môžete si vybrať možnosť *Navigation Menu* alebo *List*. Pozíciu a spôsob zobrazenia menu je možné zmeniť úpravou atribútov navigačného menu na stránke používateľského rozhrania (User Interface). Dostanete sa tam kliknutím na zdieľané komponenty (*Shared Components*) a atribúty používateľského rozhrania (*User Interface Attributes*). Na karte navigačného menu (*Navigation menu*) môžete upraviť atribúty, ktoré sú znázornené na Obrázku 9.5.



Obr. 9.5: Atribúty navigačného menu

- Display Navigation (zobraziť navigáciu) – zapnutie alebo vypnutie navigácie.
- Navigation Menu List (zoznam navigačného menu) – výber zoznamu používaného na navigačné menu.
- Position (pozícia) – výber *Top* alebo *Side* pozície navigačného menu.
- List Template (šablóna zoznamu) – výber z rôznych šablón na vykreslenie navigačného menu.
- Template Options (možnosti šablóny) – nastavenie možností šablóny používaných pre zoznam navigačného menu.

Po kliknutí na tlačidlo *Apply changes* môžete spustiť aplikáciu a vidieť zmeny.

9.6 Otázky

1. Bežne sa menu používané v aplikáciách Oracle APEX nachádza v ľavej časti obrazovky. Na ktorý typ sa vzťahuje?
2. Kde môžete nájsť definíciu položiek menu, ktoré sa majú upraviť?
3. Ktorý typ menu vám umožňuje zobraziť všetky položky súčasne bez potreby rozbaľovania submenu?

9.7 Odpovede

1. Side Menu (bočné menu).
2. Jednotlivé položky menu tvoria zoznam, ktorý je možné vytvárať a upravovať podobne ako akýkoľvek iný zoznam. Je to prístupné v zdieľaných komponentoch (*Shared Components*) v

časti navigácie (Navigation region).

3. Mega menu vám umožňuje zobraziť všetky položky súčasne jediným kliknutím bez nutnosti rozbaľovania submenu.



10. Ako spolupracovať v tíme?

P. Staniszewski, M. Sońta, A. Kierzkowski

10.1 Kolaboratívne vytváranie znalostí je podstatou vývoja v nízkom kóde

„Spolupráca nás ako ľudí odlišuje ešte viac ako naše protichodné palce“, keďže sme prirodzene naprogramovaní tak, že naším preferovaným prístupom je spolupráca (pozri napr. [1]; [3]) a sieťové platformy sú vybavené funkciami, ktoré podporujú kolektív a kolektívne zdieľanie vedomostí.

Výhody inteligentnej spolupráce na digitálnych platformách sú dvojaké:

- aspekt produktivity, keď jednotlivci využívajú kolektívnu inteligenciu na vývoj lepšieho produktu a
- motivačný aspekt, keď proces aktivuje tvorivé energie a komunitou riadený prístup k rozvoju vedomostí.

10.2 Byť spolu online

Oracle APEX je platforma vytvorená na sieťovanie a interakciu pri práci na aplikáciách a tento prístup spĺňa definíciu platformy kolaboratívneho aktivizmu. Napriek tomu si tento typ spolupráce vyžaduje „konkrétny spôsob bytia spolu“ ([3]). Jednou súčasťou zážitku zo spolupráce je možnosť prezerať, komentovať a spoluvytvárať obsah na kolektívne vytváranie vedomostí a posilňovanie kreativity a ducha prostredníctvom formovania sociálnych vzťahov je ďalšou dimenziou spoločného využitia príbehu APEX. Komunita APEX spoločnosti Oracle s viac ako desiatimi podujatiami regionálnej komunity ročne, okrem iného:

- APEX World, Holandsko
- APEX World, APEX Alpe Adria, Rakúsko/Chorvátsko/Slovinsko
- APEX World, APEX Connect, Nemecko
- APEX World, ODTUG Kscope, USA

dosiahla štádium vývoja na vývoj komerčnej platformy, ktorá bola sľúbená, že zostane pre používateľov navždy bezplatná, v spolupráci s otvoreným zdrojom a s ambíciou riadiť sa smermi energizujúceho ducha komunity.

S množstvom interakcií a projektov sa spôsob rozvoja komunity ešte viac pohltí, čo prináša príležitosti „[...] spôsoby bytia spolu online, najmä vo veľmi pohlcujúcich prostrediach, môžu mať ďalšiu kognitívnu a vnímavú vrstvu, ktorá je zatiaľ nepreskúmaná. “ ([3]) s ambíciou dosiahnuť zrelosť digitálneho aktivizmu a pozývať rôzne publikum, ktoré nie je vyškolené v oblasti IT.

10.3 Technická zdatnosť vďaka spolupráci

Ak hľadáme akademické články o APEX, väčšina článkov pokrýva prípady obchodného použitia, ktoré ukazujú príklady medzirezortných snáh o vytvorenie aplikácie: Ľudské zdroje, Spoločenská zodpovednosť spoločnosti, Komunikácia s rôznymi používateľmi, ktorí zadávajú nielen údaje, ale aj príspevok pri konceptualizácii dizajnu samotnej platformy.

Tento prístup, prístupnosť k platforme a široká škála vzdelávacieho obsahu o APEX robia z tohto nástroja aj šancu vstúpiť do sveta nízkeho kódovania a precvičiť obchodné povedomie o praktických aplikáciách aplikácií. S týmto zmýšľaním otvára APEX príležitosť pre profesionálov, ktorí nie sú v oblasti IT, na spoluprácu na IT projektoch a poskytuje priestor na zvýšenie úrovne technologickej zdatnosti. Tieto kompetencie a postoje vytvárajú lepšie pracovné prostredie pre všetky zainteresované strany zapojené do procesu vývoja softvéru.

Spôsoby spolupráce zvyčajne súvisia s metodikou projektu. Aj keď sú platformy s nízkym kódom, ako je APEX, navrhnuté tak, aby dosahovali ciele veľmi rýchlo, zvyčajne potrebujeme viac ako jedného vývojára na realizáciu projektu v predpokladanom čase a rozpočte. V takejto situácii potrebujeme ten správny nástroj, ktorý pomôže vývojárom spolupracovať a riešiť výzvy a príležitosti typické pre coworking v IT projektoch. V závislosti od rozsahu a požiadaviek našej alebo našej organizácie to môže súvisieť s aspektmi, ako sú:

- Štandardizácia – keď chceme zabezpečiť, aby boli všetky prvky aplikácie konzistentné. Týka sa to štandardov používateľského rozhrania, názvových konvencií, programovacích techník, štýlu kódovania a všetkých ostatných štandardov typických pre vybranú technológiu. Napríklad vo svete APEX môžete len premýšľať o rôznych spôsoboch pomenovania položiek, procesov alebo validácií alebo zvážiť štandardy pri výbere čísel strán.
- Vzdelávanie – vývoj aplikácie s ďalšími vývojármi pomáha zvyšovať zručnosti. Ostatní členovia tímu môžu byť pokročilejší v aspektoch, ktoré nás veľmi nezaujímajú. Dokonca aj pri použití platformy s nízkym kódom, ako je APEX, ktorá teoreticky umožňuje každému používateľovi byť „full-stack“ vývojárom, môžeme nájsť špecializácie. Niektorí ľudia sa napríklad cítia oveľa pohodlnejšie so stránkou databázy, iní s čistým APEX a ďalší s funkciami a jazykmi frontendu. Väčší tím pomáha ostatným rozvíjať zručnosti, ktoré nemali a zvyšovať všeobecnú kvalitu aplikácie.
- Čas – je prirodzené, že viac vývojárov môže dodávať aplikácie rýchlejšie, čo vedie k lepšiemu výkonu tímu alebo oddelenia zodpovedného za konkrétny systém alebo platformu.
- Kvalita – väčší tím môže implementovať proces kontroly kódu a kontrolovať kvalitu prvkov dodaných členmi tímu. Zvyšuje kvalitu konečného produktu a obmedzuje počet chýb.
- Spätná väzba – spolupráca znamená aj komunikáciu medzi vývojármi a používateľmi. Je výhodné mať od používateľov informácie o kvalite aplikácie, nahlásených chybách, či požiadavkách na zmenu.

Oracle APEX ako vyspelá platforma pokrýva rôzne stratégie spolupráce so vstavanými funkciami, ale aj s možnosťami rozšírenia o externé nástroje, ako sú sady na spoluprácu, aplikácie na správu úloh alebo nástroje na overovanie kódu.

Aj keď sme práve začali našu cestu s APEX, je nevyhnutné vedieť, ako môžeme spolupracovať s ostatnými používateľmi s natívnymi funkciami, aby bola takáto práca produktívnejšia, a byť si istí, ako riešiť typické problémy, ako je zadávanie úloh, práca na presných stránkach a funkciách, alebo kontrola stavu plánovanej práce.

Na samom začiatku je najdôležitejšia komunikácia. Ak je tím viac ako jedna osoba, mali by ste mať spôsoby, ako si naplánovať prácu na nasledujúce dni, informovať o stave a kontrolovať čas a termíny. Nič nie je dôležitejšie ako pravidelné stretnutia a komunikácia, ak je niečo nejasné. Je dobré myslieť na denné stretnutia, týždenné plánovanie alebo iné pravidelné udalosti. Môžete tiež implementovať jednu zo známych metodík, ako je Scrum, aby ste mali takéto zvyky napísané na jednom mieste. Začnite to robiť a zistíte, aký spôsob spolupráce je pre váš tím najlepší.

Na začiatku, keď premýšľate o niektorých nástrojoch spolupráce, môžete zvážiť základné

tabuľky na sledovanie toho, kto čo robí. Samozrejme, na trhu existuje množstvo špecializovaných aplikácií na správu úloh, ako je Jira, Monday, ClickUp, Redmine atď. a samozrejme s rastúcim tímom a rozsahom môžete zvážiť použitie jednej z nich. Väčšinu týchto akcií však možno pokryť funkciami zahrnutými v platforme Oracle Application Express.

10.4 Popis funkcií

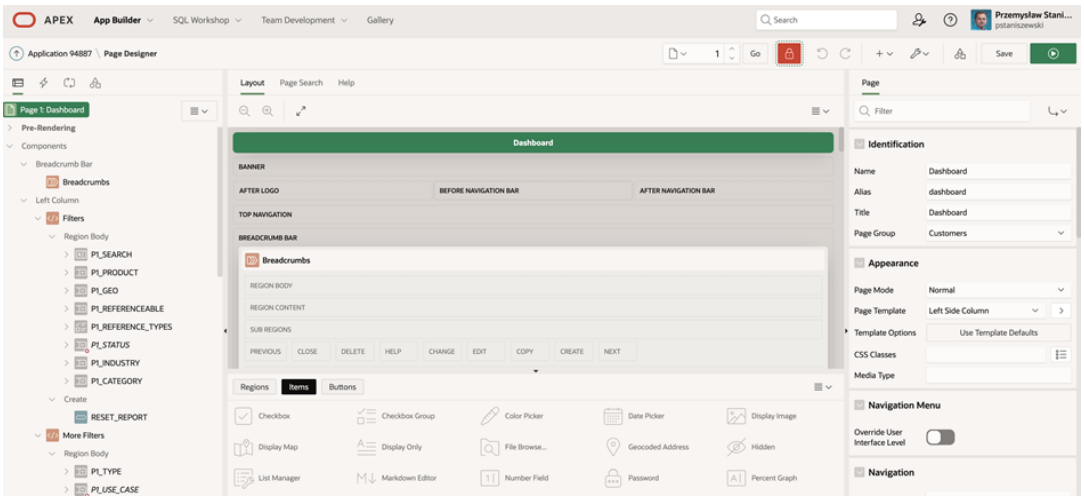
10.4.1 Uzamknutie stránky

Uzamknutie stránky je užitočná funkcia, ktorá zabraňuje viacerým vývojárom upravovať rovnakú stránku súčasne. Keď vývojár upravuje stránku, funkcia Blokovanie stránky bráni ostatným vývojárom vykonávať zmeny na tej istej stránke, kým prvý vývojár nedokončí svoju prácu a stránku neuvoľní. To pomáha predchádzať konfliktom a zaisťuje, že zmeny sú vykonávané usporiadane a kontrolovane (pozri obrázok 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		

Obr. 10.1: Funkcia blokovania stránok v APEX.

Konkrétnu stránku môžete uzamknúť kliknutím na ikonu zámku v zoznamoch stránok v aplikácii Application Builder alebo to môžete urobiť na úrovni zobrazenia stránky. (pozri obrázok 10.2).



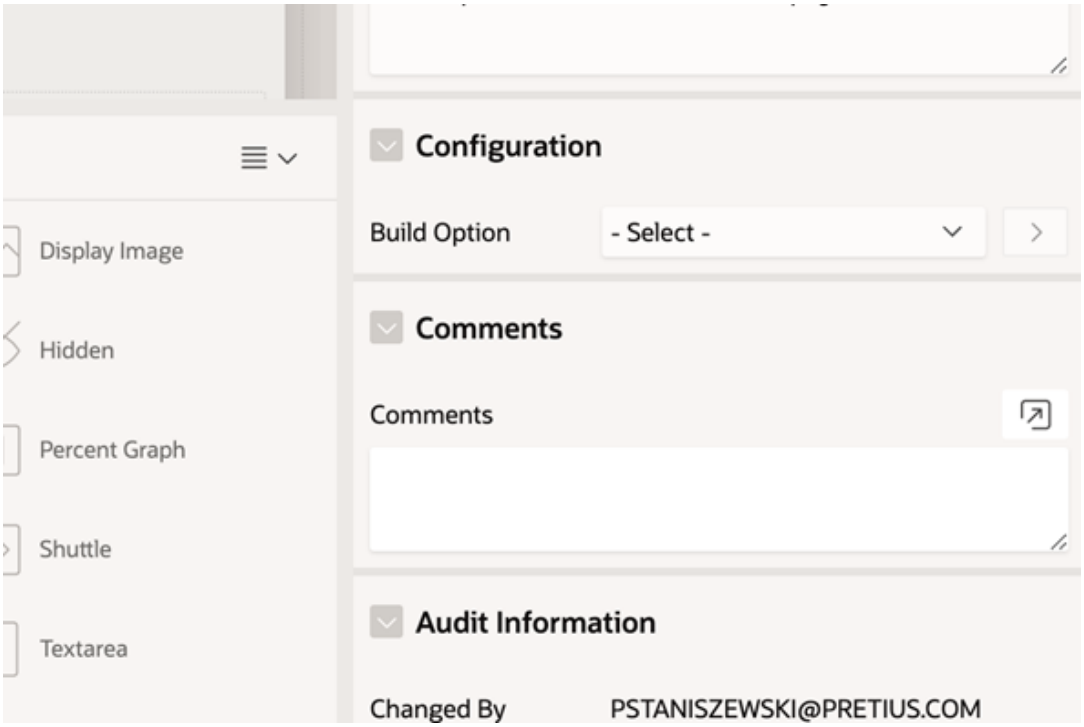
Obr. 10.2: Uzamknutie konkrétnej stránky.

Počas zamykania alebo odomykania stránky môžete pridať komentár, aby ste vy alebo iní

vývojári mohli vidieť viac informácií o dôvode alebo postupe práce.

10.4.2 Komentáre

Oracle APEX umožňuje vývojárom pridávať komentáre k stránkam, regiónom, položkám a mnohým ďalším prvkom aplikácie. Komentáre možno použiť na poskytnutie kontextu pre iných vývojárov, vysvetlenie dôvodov rozhodnutia o dizajne alebo jednoducho zdokumentovanie zmien vykonaných v aplikácii. Pomocou komentárov môžu vývojári navzájom lepšie komunikovať a zabezpečiť, aby boli všetci na rovnakej stránke (pozri obrázok 10.3).

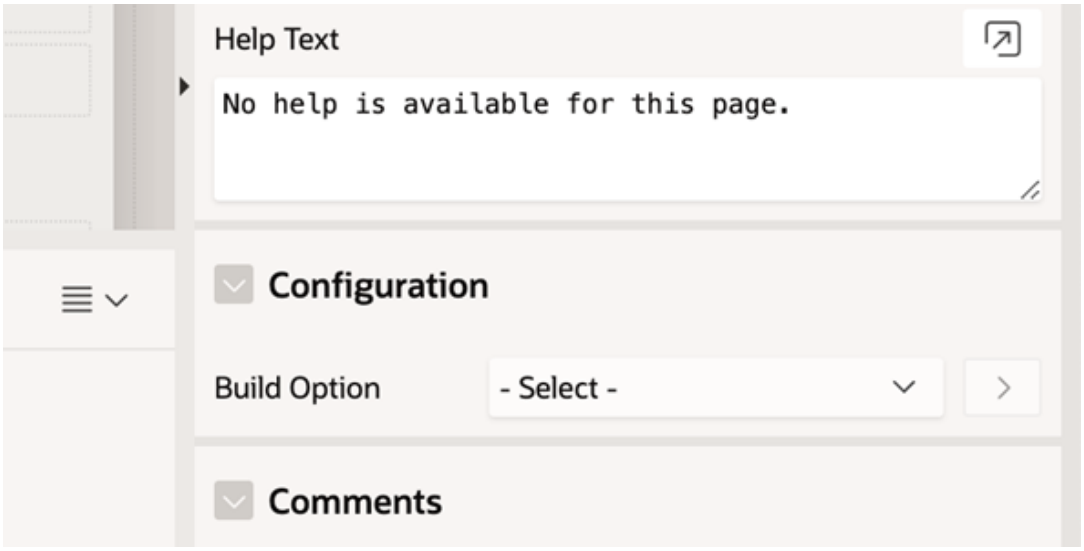


Obr. 10.3: Pridávanie komentárov.

Zdokumentovanie kódu pre ostatných členov tímu alebo budúcich vývojárov, ktorí sa budú starať o vývoj alebo údržbu aplikácií, je dôležité. Aj keď je možné pomocou platformy s nízkym kódom definovať, ako bola konkrétna funkcia vykonaná, pridanie niektorých informácií je stále dobrým postupom, ak je špecifický prvok vytvorený jedinečne.

10.4.3 Možnosti zostavovania

Možnosti zostavovania (Build options) sú ďalšou funkciou, ktorá môže pomôcť tímom efektívnejšie spolupracovať v Oracle APEX. Možnosti zostavenia umožňujú vývojárom špecifikovať rôzne nastavenia zostavy pre rôzne prostredia, ako je vývoj, testovanie a výroba. To uľahčuje správu zmien v rôznych prostrediach a zaisťuje, že pri nasadení aplikácie sa použijú správne nastavenia (pozri obrázok 10.4).



Obr. 10.4: Možnosti zostavovania.

Môžete s nimi zaobchádzať ako s informáciami, ak má byť konkrétna funkcia (stránka, položka, proces) zahrnutá v nasledujúcom vydaní, ktoré bude exportované/importované do iného prostredia. Pri exporte/importe vašej aplikácie si môžete vybrať, či je zahrnutá konkrétna „zostava“. Len si predstavte situáciu, keď jeden vývojár pracuje na nejakej funkcii a nestihne to do termínu. Namiesto odstránenia tejto stránky z aplikácie môžete pre ňu vytvoriť vyhradenú „zostavu“ a vylúčiť ju pri exporte aplikácie.

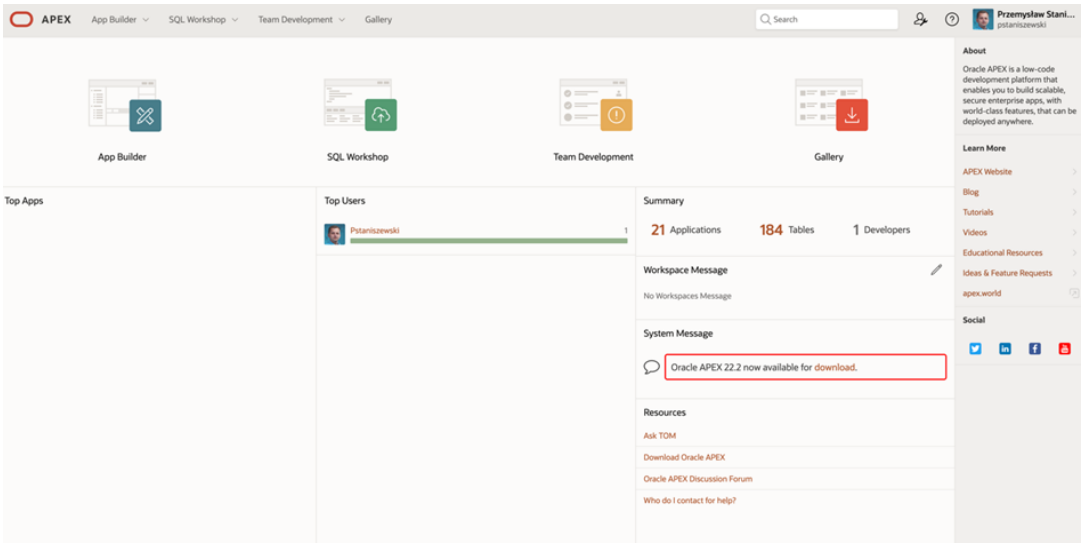
Použitie tejto funkcie môže byť prvým krokom pri definovaní procesu CI/CD pre vašu aplikáciu, ktorý je nevyhnutný pri profesionálnom vývoji softvéru.

10.4.4 Tímový vývoj

Funkcia Team Development Oracle APEX umožňuje tímom spolupracovať na rovnakej aplikácii v kontrolovanom a kolaboratívnom prostredí. Pomocou tejto funkcie môžu členovia tímu riadiť projekty, najmä:

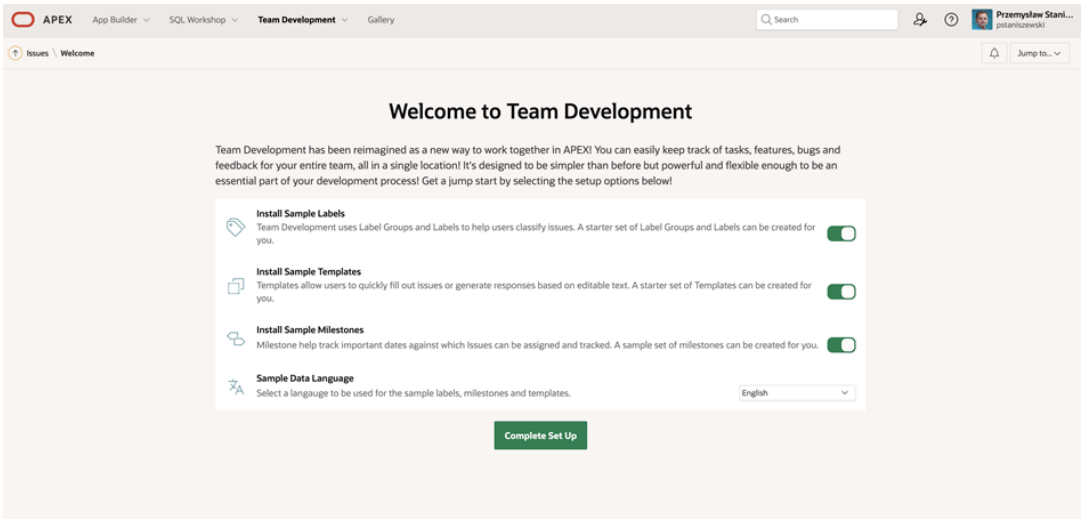
- Definovať míľniky a štádiá projektu;
- Vytvárať, prideľovať a monitorovať úlohy/problémy;
- Aplikovať klasifikácie pomocou skupín štítkov a štítkov;
- Vytvárať šablóny problémov, aby bolo vytváranie úloh alebo podávanie správ konzistentné;
- Prepojiť problémy s aplikáciou a stránkou;
- Komentovať a diskutovať o existujúcich problémoch. Zmeniť ich stav a monitorovať priebeh projektovej fázy;
- Prihlásiť sa na odber rôznych typov problémov;
- Monitorovať spätnú väzbu od používateľov aplikácie a rozhodovať sa, čo s ňou urobiť.

Team Development je jednou zo štyroch hlavných sekcií dostupných v APEX IDE. Jeho dostupnosť závisí od vašej verzie APEX. Bol predstavený v APEX 19.2, ale v najnovších verziách nie je k dispozícii hneď po vybalení, ale vyžaduje ďalšie akcie na jeho sprístupnenie (pozri obrázok 10.5).



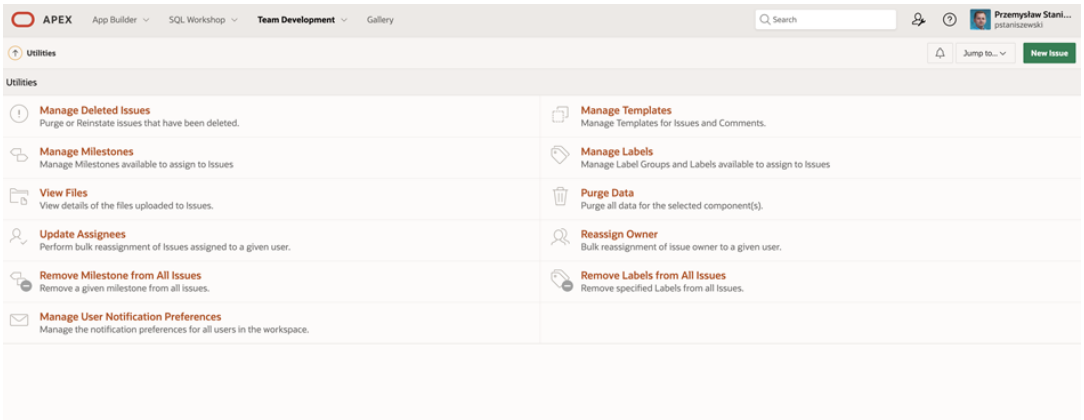
Obr. 10.5: Funkcia tímového vývoja.

Ak ho chcete zapnúť, musíte v hlavnom menu vybrať možnosť „Team Development“. Potom sa presuniete na vyhradenú inštalačnú stránku. Ak chcete, môžete zahrnúť príklady problémov, míľnikov, šablón a štítkov, ale túto časť môžete tiež preskočiť a aktivovať funkciu bez nej (pozri obrázok 10.6).



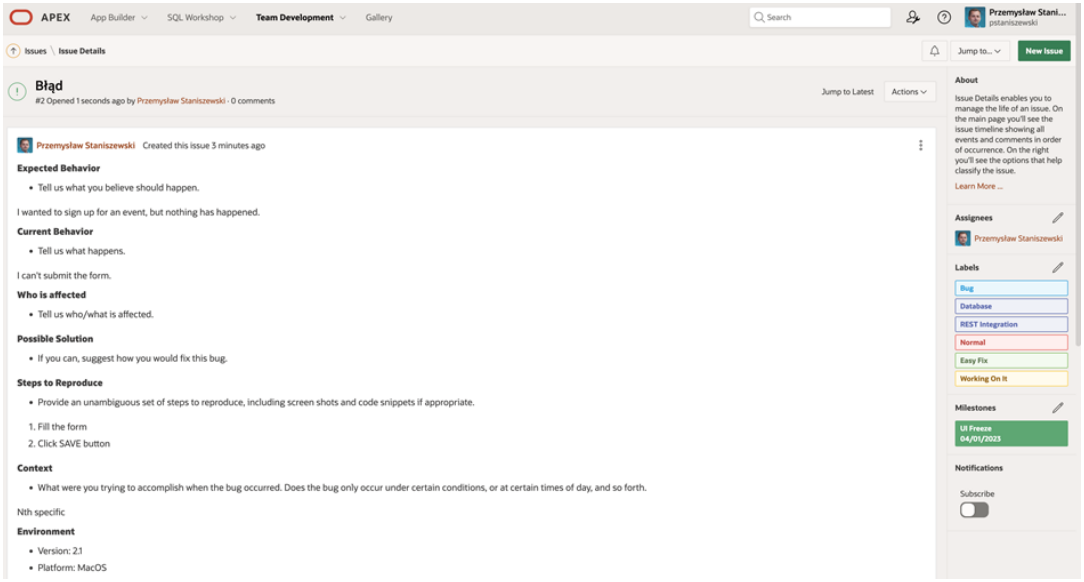
Obr. 10.6: Aktivácia funkcie tímového vývoja.

Ak by váš tím chcel začať používať „Team Development“, mali by ste najprv nakonfigurovať jeho slovník pre váš pracovný štýl. Potom prejdite do časti Utilities a definujte typy svojich problémov, fázy projektu a míľniky alebo šablóny pre typickú komunikáciu v projekte (pozri obrázok 10.7).



Obr. 10.7: Prispôsobenie funkcie Team Development.

Najdôležitejšou časťou tejto funkcie je sekcia Issue, kde nájdete všetky nahlásené problémy, kategorizuje ich a priraďuje ich konkrétnym osobám. Každý problém môže byť nastavený na viac ako jedného vývojára, ktorý bude môcť pridávať komentáre, diskutovať o ňom v kontexte a meniť jeho stav. V prípade potreby môžu používatelia nahrať súbory a v podstate spustiť všetku komunikáciu súvisiacu s problémami v rámci „Team Development“. Je tiež dôležité, aby sa všetky problémy dali prepojiť s konkrétnou aplikáciou a stránkou, takže je jednoduché začať vyšetrovanie a sledovať, ktoré problémy súvisia s ktorou časťou aplikácie. Stav problému je jednoduchá verzia pracovného toku problému, ktorý možno nájsť v mnohých systémoch správy úloh – používatelia ich môžu nahlásiť, označiť ako prebiehajúce - „in progress“, vrátiť sa k reportérovi a ukončiť s informáciami o úspechu alebo rôznych dôvodoch ukončenia (pozri obrázok 10.8).



Obr. 10.8: Komunikácia súvisiaca s problémami v rámci funkcie Team Development.

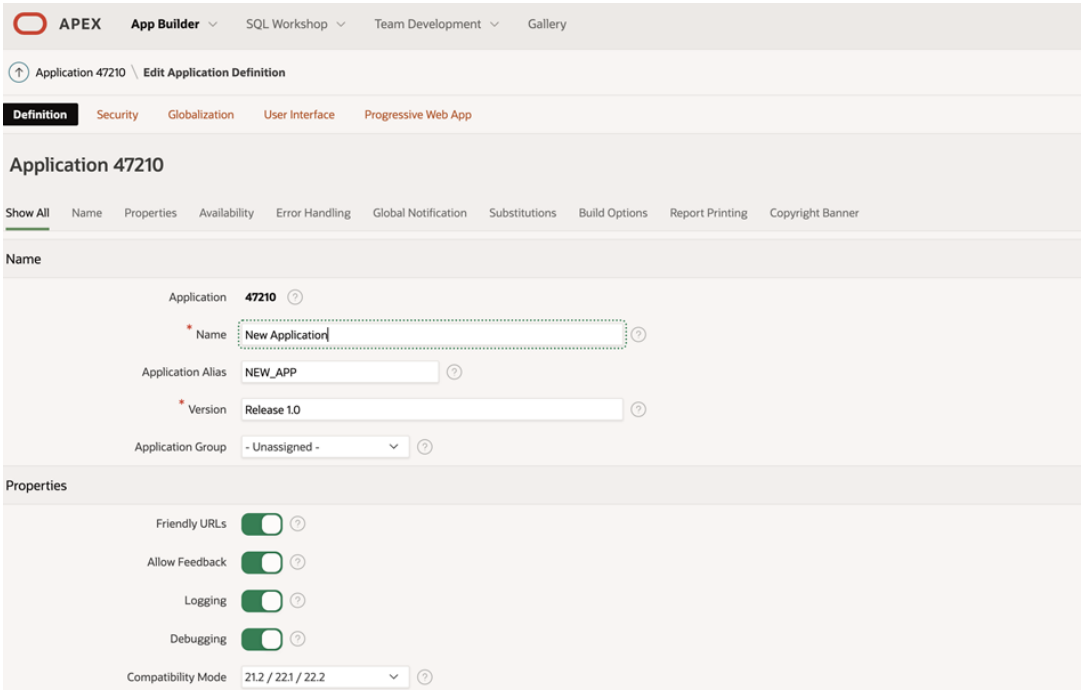
Vývojári sa tiež môžu prihlásiť na odber problémov, ktoré im boli pridelené, čo poskytuje e-mailové informácie, keď dôjde k akejkoľvek akcii na úrovni problému. Aplikácia umožňuje používateľom obmedziť počet e-mailov ich odosielaním ako hodinové alebo denné súhrny.

Dôrazne odporúčame začať používať „Team Development“, ak váš tím nemá implementovaný iný systém riadenia úloh alebo ak na riadenie projektu používate veľmi základné nástroje, ako sú tabuľky.

10.4.5 Spätná väzba

Stránky spätnej väzby sú vstavanou funkciou v Oracle APEX, ktorá umožňuje koncovým používateľom poskytovať spätnú väzbu o funkčnosti a výkone aplikácie. Stránky spätnej väzby možno prispôsobiť potrebám aplikácie a môžu obsahovať rôzne typy spätnej väzby, ako sú hlásenia o chybách, požiadavky na funkcie a všeobecná spätná väzba.

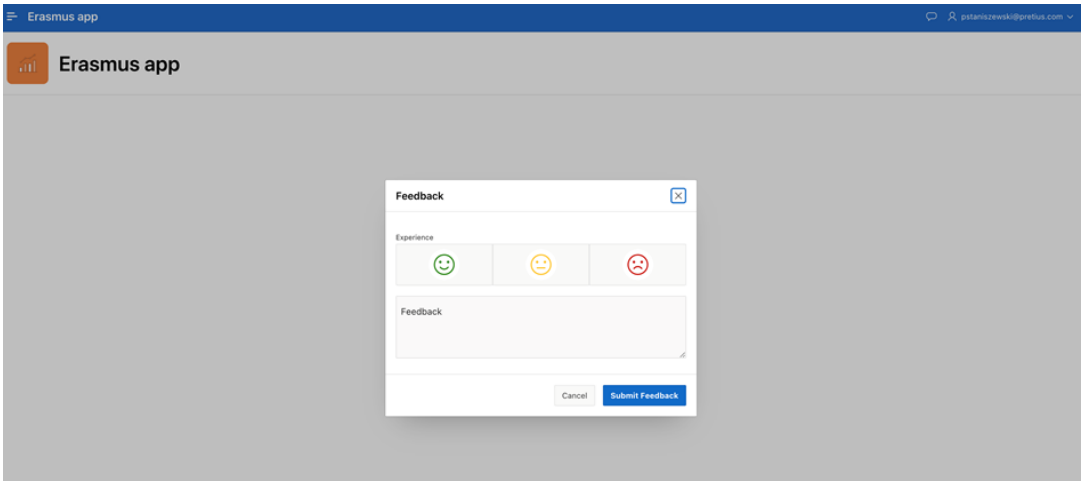
Je dôležité získať informácie od koncových používateľov a považovať ich za rozšírenie vývojového tímu. Čím užšie budeme spolupracovať nielen interne, ale aj so skutočnými používateľmi aplikácie, tým lepšiu kvalitu aplikácie budeme mať. Umožňuje tiež tímu prispôsobiť aplikáciu skutočným potrebám a pomáha budovať dôveru medzi vývojárskym tímom a používateľmi (pozri obrázok 10.9).



Obr. 10.9: Povolenie spätnej väzby v definícii aplikácie.

Ak ju chcete zapnúť, musíte vybrať možnosť počas vytvárania aplikácie. Ak aplikácia už existuje, prejdite na upravenie definície aplikácie - „Edit Application Definition“ a v časti vlastností označte možnosť povolenia spätnej väzby - „Allow Feedback“.

Po aktivácii sa pridá vyhradenú ikonu v menu aplikácie. Po kliknutí naň môže koncový používateľ odoslať spätnú väzbu (pozri obrázok 10.10).



Obr. 10.10: Odoslanie spätnej väzby v aplikácii.

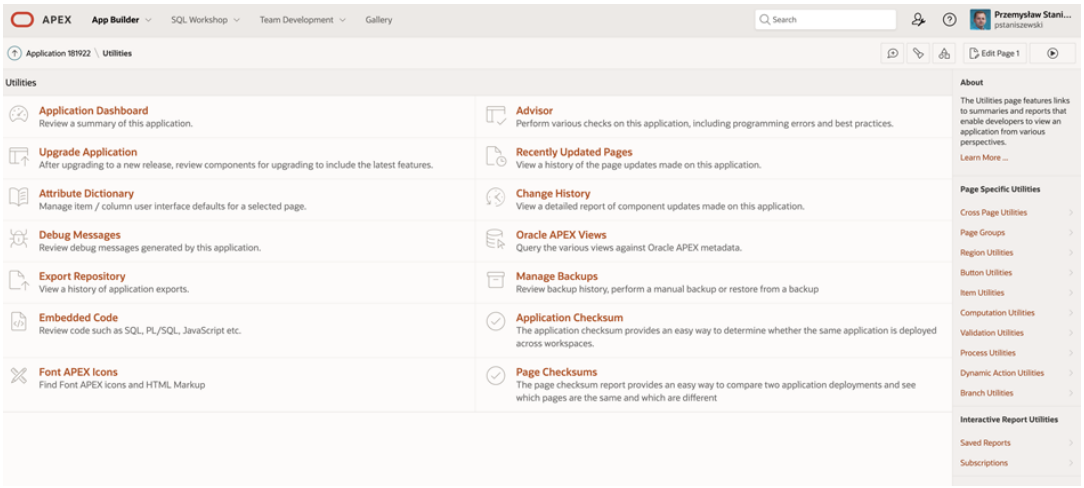
Vzhľad stránky spätnej väzby je možné upraviť, keďže ide o štandardnú stránku APEX. Všetky informácie, ktoré koncový používateľ poskytuje, sú dostupné v sekcii Team Development a môžu byť analyzované tímom (tiež je možné ich sledovať na úrovni konkrétnej aplikácie v jej sekcii administrácie). Okrem toho vývojári dostávajú podrobné informácie súvisiace s nahlásenou spätnou väzbou (ID aplikácie, číslo stránky, údaje používateľa, informácie o jej prostredí atď.) a môžu ju premeniť na problém (issue).

10.4.6 Štandardizácia

Pri vývoji aplikácií v tíme je tiež veľmi dôležité udržiavať konzistentnosť medzi vývojovými štandardmi rôznych členov tímu. Ako v každom aspekte spolupráce je kľúčom k úspechu komunikácia – na začiatku projektu by sa mal tím dohodnúť na spôsobe vývoja a využitia komponentov APEX. K tým najdôležitejším aspektom patria:

- Konvencia pomenovania (pre položky, procesy, validácie, dynamické akcie atď.);
- Číslovanie a pomenovanie stránok (alias);
- Povolené prvky (napr. pre slovníky používame len interaktívne mriežky, každý formulár musí byť zobrazený ako modálna stránka);
- UX/UI štandardy (napr. vzhľad a štýl typických stránok, politika zosúladenia pre rôzne dátové časy, konzistentné formáty dát, odporúčané ikony pre konkrétne akcie atď.);
- Konzistencia údajov (napr. mať spoločné slovníky definované na jednom mieste so zdieľanými komponentmi).

APEX ponúka niekoľko riešení, ktoré pomáhajú tímom so správou štandardizácie. Väčšinu z nich možno nájsť v sekcii pomocných nástrojov - „Utilities“ na úrovni vybranej aplikácie (pozri obrázok 10.11).



Obr. 10.11: Ponuka pomocných programov.

Medzi najdôležitejšie na začiatku vývoja patria:

- Medzistránkové zostavy (Page Specific Utilities) – umožňujú vývojárom porovnávať rôzne prvky APEX (atribúty stránok, zostavy, tlačidlá, položky atď.) v aplikácii, aby skontrolovali konzistenciu a aktualizovali ich naraz bez manuálnej zmeny atribútov na viacerých miestach.
- História zmien/Nedávno aktualizované stránky – umožňuje sledovanie akcií vykonaných vývojármi.
- Definícia predvolených nastavení (Attribute Dictionary) – umožňuje vytvorenie spoločného slovníka, aby sa zachovalo konzistentné pomenovanie, formátovanie, typ údajov atď. pre prvky, ktoré sa opakujú na mnohých miestach v aplikácii. Napríklad dátum aktualizácie - „Update date“ môže existovať v rôznych zostavách alebo položkách v aplikácii, ale vždy chceme mať rovnaký štítok a formát dátumu pre každý výskyt.

Samozrejme, APEX poskytuje mnoho ďalších funkcií a stratégií na udržanie konzistencie na úrovni databázy a aplikácie. Pre tých, ktorí by si chceli rozšíriť svoje znalosti, si prečítajte dokumentáciu APEX a skontrolujte pokročilejšie možnosti, ako je globálna stránka, plány, predvolené nastavenia používateľského rozhrania, mechanizmus aplikácie hlavnej témy a prihlásenie na odber, zobrazenia APEX alebo doplnky.

10.5 Záver

Oracle APEX ako vyspelá vývojová platforma poskytuje mnoho funkcií, ktoré pomáhajú pri spolupráci. Ako vývojári s nízkym kódom ich môžeme využiť a profesionálne vytvárať a udržiavať aplikácie založené na APEX bez rozširovania platformy alebo jej integrácie s externými nástrojmi. Ak ste v APEX noví a v stratégiách spolupráce v programovaní, začnite s najzákladnejšími krokmi opísanými v tejto kapitole:

- Komunikácia a pravidelné stretnutia s tímom (naživo alebo na diaľku – použite nástroje ako Google Meet, Zoom alebo MS Teams na video/hlas a Slack, Google Chat alebo MS Teams na rýchle posielanie správ);
- Začnite používať nástroje spolupráce, napr. APEX Team Development, plánovať vývoj a overovať pokrok;
- Začnite dokumentovať svoj kód a štandardy. A overte si ich!

Spolupráca pri vývoji malého kódu môže niekedy znieť prehnane, ale oplatí sa vyskúšať a prináša rovnaký zisk ako pri štandardnom programovaní. A čo viac, väčšinu typických stratégií spolupráce, ako je „kontrola kódu“, „párové programovanie“ alebo „mob programovanie“, možno použiť aj s platformou s nízkym kódom, aj keď niekedy kontrolovaný prvok nie je čistý kód, ale je proces, stránka alebo konfigurácia.

V závislosti od rozsahu projektu alebo štandardov implementovaných v organizácii môže byť niekedy potrebné rozšíriť proces spolupráce. Napríklad vo veľkých projektoch budete musieť začať používať úložiská kódu, nástroje na automatizované vytváranie a nasadzovanie alebo softvér určený na testovanie a overovanie kvality vášho kódu. Využitie prvkov profesionálnych procesov CI/CD ako Git, Docker, Flyway, Liquibase, Jenkins, Gradle, SonarQube, Selenium, Cypress a mnoho ďalších je možné aj s aplikáciou založenou na APEX. V mnohých situáciách však prvky zahrnuté v platforme stačia na vytvorenie revolučnej aplikácie s tímom kolegov vývojárov.

10.6 Otázky

1. Čo je funkcia uzamykania stránky a prečo je dôležitá?
2. Ako môžu vývojári komunikovať počas vývoja s APEX?
3. Čo môžu členovia tímu využiť pomocou funkcie Team Development?

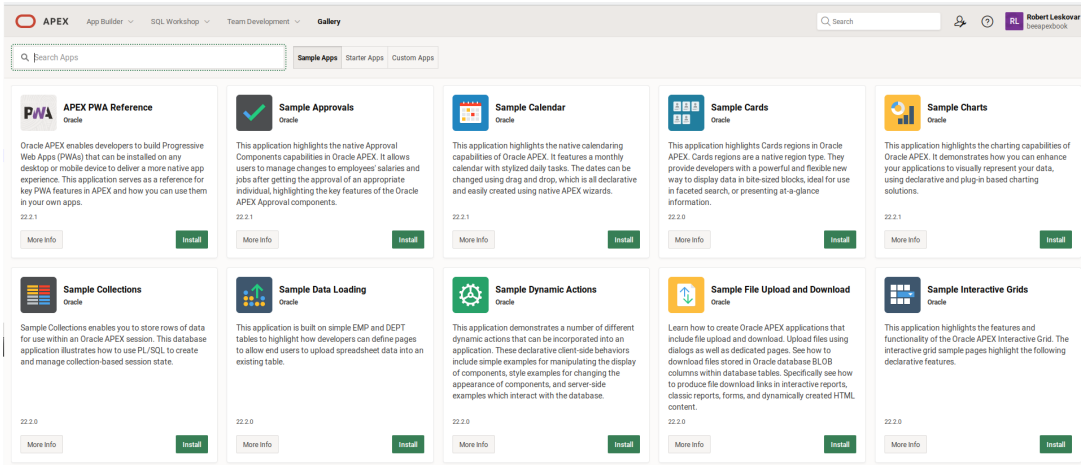
10.7 Odpovede

1. Uzamykanie stránky je užitočná funkcia, ktorá bráni viacerým vývojárom upravovať rovnakú stránku súčasne, a tým predchádzať konfliktom.
2. Vývojári môžu pridávať komentáre na stránky, oblasti, položky a mnoho ďalších prvkov aplikácie. Komentáre možno použiť na poskytnutie kontextu pre iných vývojárov, vysvetlenie dôvodov rozhodnutia o dizajne alebo jednoducho zdokumentovanie zmien vykonaných v aplikácii. Pomocou komentárov môžu vývojári medzi sebou lepšie komunikovať a zabezpečiť celkovú kvalitu softvéru.
3. Funkcia Team Development umožňuje členom tímu riadiť vývojové projekty, najmä: definovať míľniky a fázy projektu, vytvárať, priraďovať a monitorovať úlohy/problémy, aplikovať klasifikácie pomocou skupín štítkov a štítkov, vytvárať šablóny problémov, aby bolo vytváranie úloh alebo vykazovanie konzistentné, prepojiť problémy s aplikáciou a stránkou, komentovať a diskutovať o existujúcich problémoch, meniť ich stav a sledovať priebeh fázy projektu a prihlásiť sa na odber rôznych typov problémov.

11. Ako využiť galériu vzorových aplikácií a pluginov?

VJERAN STRAHONJA, DIJANA OREŠKI

Používatelia si môžu vyskúšať, učiť sa, používať a experimentovať so štartovacími aplikáciami, vzorovými aplikáciami a pluginmi v Oracle APEX. Všetky pracovné prostredia Oracle APEX sa dodávajú so vzorovými aplikáciami (sample apps) a štartovacími aplikáciami (starter apps) (viď Obrázok 11.1).



Obr. 11.1: Pracovné priestory Oracle APEX vzorové a štartovacie aplikácie.

Novšie verzie APEX nahrávajú štartovacie aplikácie, vzorové aplikácie a plug-iny z cloudového úložiska git.

11.1 Ako nainštalovať vzorové a štartovacie aplikácie?

Používatelia si môžu z galérie stiahnuť vzorové a štartovacie aplikácie. Ak to správca inštancie povolí, prostredníctvom Galérie je možné inštalovať aj vlastné aplikácie.

Ak chcete **inštalovať** aplikácie z galérie:

1. Prejdite na domovskú stránku pracovného priestoru.
2. Vyberte **Gallery**. Karty vzorových, štartovacích a vlastných aplikácií sa zobrazia, keď sa prvýkrát načíta stránka **Gallery**.

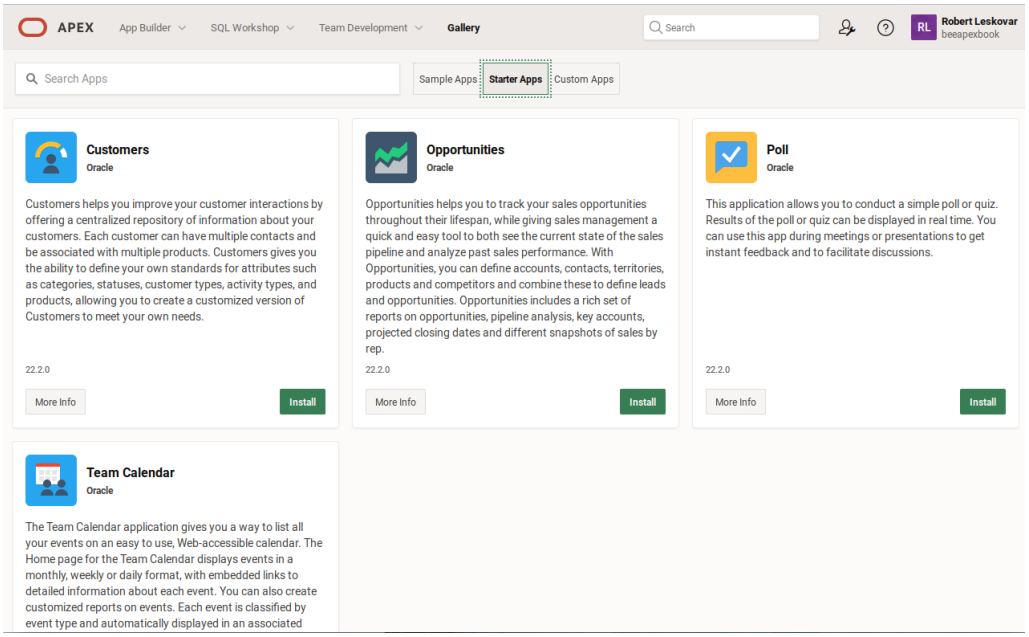
3. Vyberte **Sample Apps** alebo **Starter Apps**. Karty sú používané pre zobrazenie zoznamu aplikácií. Ak program nebol predtým nainštalovaný, každá karta obsahuje tlačidlo Install. Spustenie a odstránenie aplikácie sú prítomné, ak už bola aplikácia nainštalovaná.
4. Výberom položky **Install App** vykonáte inštaláciu aplikácie.
5. Rozbaľte sekciu Advanced Settings (nie je povinné).
   a. Uveďte, či bude ID aplikácie priradené ručne alebo automaticky (predvolené nastavenie).
   b. Zmeňte Parsing Schema.
6. Kliknite na Install Application, aby ste dokončili inštaláciu.

Odinštalovanie aplikácií z galérie vykonáte výberom ikony Remove App. Používatelia môžu odinštalovať aplikácie, ktoré boli nainštalované. Aplikácie, ktoré boli importované z GitHub a stiahnuté, nemajú ikonu na odinštalovanie, takže aby sa dali odinštalovať, musia byť odstránené z pracovného priestoru. Kroky na **odinštalovanie** aplikácií z galérie:

1. Prejdite na domovskú stránku pracovného priestoru.
2. Vyberte ikonu Gallery. Zobrazia sa karty vzorových aplikácií, štartovacích a vlastných aplikácií, keď sa stránka galéria načíta po prvýkrát.
3. Vyberte štartovacie aplikácie alebo vzorové aplikácie. Priamo stiahnuté programy z galérie budú mať viditeľné ikony Remove a Run.
4. Pre odstránenie aplikácie kliknite na Remove.

11.2 Štartovacie aplikácie

Štartovacie aplikácie sú užitočné aplikácie, ktoré ponúkajú samostatné bodové riešenia vytvorené na splnenie jednoduchých požiadaviek, ktoré nevyžadujú podstatnú a zbytočnú komplexnú odpoveď. Tieto aplikácie môžete používať "out of the box" alebo ich môžete rozšíriť o vlastné jedinečné funkcie (viď Obrázok 11.2).



Obr. 11.2: Štartovacie aplikácie v galérii.

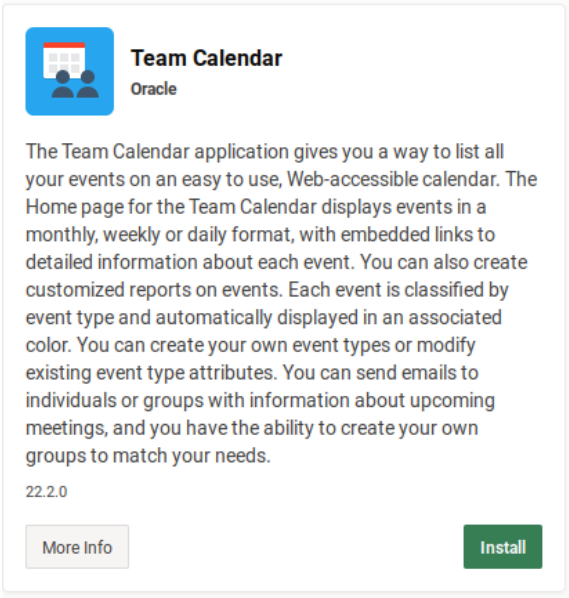
Príklady štartovacích aplikácií:

- Zákazníci (Customers) - aplikácia Customers poskytuje konsolidovanú databázu údajov o klientoch a umožňuje používateľom zlepšiť interakciu so zákazníkmi. Každý zákazník má potenciál byť prepojený s viacerými kontaktmi a položkami. Aplikácia Customers

umožňuje nastaviť vlastné kritériá pre funkcie, ako sú kategórie, stavy, druhy zákazníkov, typy činností a tovaru, takže môžete navrhnúť špeciálnu verziu aplikácie Customers podľa svojich požiadaviek.

- Príležitosti (Opportunities) – Táto aplikácia poskytuje manažmentu predaja rýchle a jednoduché riešenie na zistenie aktuálneho stavu predajného potrubia a analýzu historických výsledkov predaja. Pomáha sledovať predajné príležitosti počas celého ich životného cyklu. V aplikácii Opportunities môžete definovať potenciálnych zákazníkov a príležitosti kombináciou účtov, kontaktov, regiónov, tovarov a konkurentov. Aplikácia Opportunities ponúka komplexnú zbierku informácií o príležitostiach, prieskume potrubia, významných účtoch, predpokladaných termínoch uzavretia a ďalších prehľadoch obchodných zástupcov.
- Anketa (Poll) – používatelia môžu pomocou tejto aplikácie vytvoriť rýchly prieskum alebo test. Výsledky ankety alebo kvízu sa zobrazia okamžite. Tento nástroj možno použiť na uľahčenie diskusií a získanie okamžitej spätnej väzby počas stretnutí alebo prezentácií.
- Tímový kalendár (Team Calendar) - Pomocou aplikácie Team Calendar môžu používatelia ľahko vytvárať zoznamy všetkých udalostí vo webovom kalendári, ktorý sa jednoducho používa. Na domovskej stránke aplikácie sú uvedené udalosti v mesačnom, týždennom alebo dennom štýle spolu s integrovanými odkazmi na konkrétnejšie údaje o každej udalosti. Okrem toho môžete navrhnúť jedinečné správy o udalostiach. Každá udalosť je kategorizovaná podľa svojej kategórie a automaticky sa zobrazuje príslušná farba. Používatelia môžu vytvárať nové typy udalostí alebo meniť atribúty už existujúcich typov udalostí. Používatelia majú možnosť vytvoriť si vlastné skupiny, ktoré budú lepšie slúžiť ich potrebám, a taktiež môžu jednotlivcom alebo skupinám posielať e-maily s informáciami o nadchádzajúcich stretnutiach.

Tu je príklad inštalácie aplikácie Team Calendar (viď Obrázok 11.3).

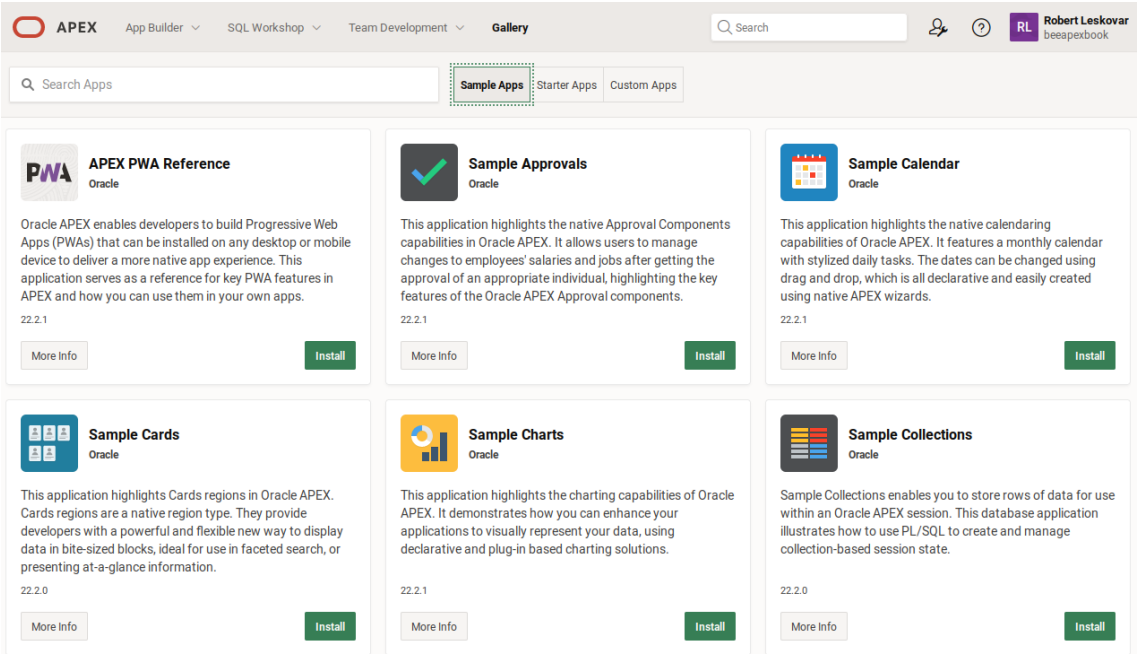


Obr. 11.3: Inštalácia štartovacej aplikácie Team Calendar.

Kliknite Install.

11.3 Vzorové aplikácie (Sample Apps)

Vzorové aplikácie sú vytvorené s cieľom zdôrazniť konkrétne možnosti a slúžia ako príručka pre vývojárov na používanie konkrétnej funkcie (viď Obrázok 11.4).



Obr. 11.4: Vzorové aplikácie.

Okrem iného je možné si vybrať:

- APEX PWA Reference - Vývojári môžu vytvárať progresívne webové aplikácie (PWA) pomocou Oracle APEX, ktoré sa dajú nasadiť na ľubovoľný počítač alebo mobilné zariadenie. V tejto aplikácii sú uvedené hlavné možnosti PWA v APEX a spôsoby ich použitia vo vlastných aplikáciách.
- Sample Approvals - Táto aplikácia predstavuje možnosti natívnych schvaľovacích komponentov Oracle APEX. Zvýraznené sú komponenty súhlasu Oracle APEX, ktoré umožňujú používateľom spravovať zmeny miezd a pracovných miest zamestnancov po získaní súhlasu oprávnenej osoby.
- Sample Calendar - Táto aplikácia predstavuje vstavané funkcie kalendára Oracle APEX. Súčasťou je mesačný kalendár so štylizovanými dennými úlohami. Na aktualizáciu dátumov možno použiť funkciu drag and drop, ktorá je úplne deklaratívna a rýchlo sa vytvára pomocou natívnych sprievodcov APEX.
- Sample Cards – V systéme Oracle APEX táto aplikácia zvýrazňuje oblasti kariet. Oblasti s kartami sú formou pôvodnej oblasti. Poskytujú vývojárom silnú a prispôsobivú novú metódu na prezentáciu údajov v malých častiach, ktorá je ideálna na použitie pri fazetovom vyhľadávaní alebo rýchlej prezentácii informácií.
- Sample Charts – Táto aplikácia predstavuje funkcie grafov Oracle APEX. Vysvetľuje, ako môžete pomocou deklaratívnych a plug-in nástrojov na tvorbu grafov vylepšiť svoje aplikácie tak, aby vizuálne zobrazovali vaše údaje.
- Sample Collections – Táto aplikácia umožňuje používateľom ukladať riadky údajov vo vzorových kolekciách na použitie v relácii Oracle APEX. Táto databázová aplikácia demonštruje vytváranie a správu stavu relácie založenej na kolekciách pomocou jazyka PL/SQL.
- Sample Data Loading - Táto aplikácia je vytvorená pomocou jednoduchých tabuliek EMP a DEPT. Cieľom je ukázať, ako môžu vývojári navrhovať stránky, ktoré umožňujú používateľom pridávať tabuľkové údaje do už existujúcej tabuľky.
- Sample Dynamic Actions – Táto aplikácia predstavuje rôzne dynamické akcie, ktoré možno pridať do aplikácií. Tieto deklaratívne správania na strane klienta zahŕňajú priame príklady na úpravu zobrazenia komponentov, príklady štýlov na prispôsobenie vzhľadu komponentov

a príklady na strane servera na interakciu s databázami.

- Sample File Upload and Download – Naučte sa vytvárať aplikácie Oracle APEX s funkciou odosielania a sťahovania súborov. Využívajte na odosielanie súborov špecializované stránky aj dialógové okná. Pozrite si postup sťahovania súborov zo stĺpcov BLOB databázy Oracle v databázových tabuľkách. Pozrite si najmä, ako vytvoriť odkazy na sťahovanie súborov vo formulároch, interaktívnych zostavách, tradičných zostavách a dynamickom obsahu HTML.
- Sample Interactive Grids – V tejto aplikácii sú zvýraznené vlastnosti a funkcie interaktívnej siete Oracle APEX. Na vzorových stránkach interaktívnej mriežky sú zvýraznené tieto deklaratívne funkcie.
- Sample Maps – V tejto aplikácii sa nachádza množstvo príkladov zobrazenia súradnicových údajov na mape. Použite funkciu tepelnej mapy, čiary, polygóny alebo mapové značky. Funkcie Oracle Spatial (dostupné v každej databáze Oracle) možno ľahko integrovať s mapovou oblasťou APEX na vykonávanie "Within Distance Search", "Nearest Neighbor Search" alebo iných priestorových analýz.
- Sample Master Detail – This application showcases Oracle APEX's native master detail capabilities. There are four possible master detail page layouts in the application. The first two layouts use editable Interactive Grids to display master detail on a single page. The final two layouts present the master detail on two pages, using a combination of editable Interactive Grids, form inputs, vintage reports, and modal popups.
- Sample Reporting – Táto aplikácia predstavuje funkcie reportovania Oracle APEX. Pomocou jazyka SQL môžu používatelia deklaratívne vytvárať interaktívne zostavy (Interactive Reports), interaktívne mriežky (Interactive Grids), zostavy s fazetovým vyhľadávaním (Faceted Search Reports), zostavy s kartami (Cards Reports) a klasické zostavy (Classic Reports).
- Sample REST Services – Táto aplikácia demonštruje, ako používať Oracle APEX na pripojenie k externým službám REST. Aplikácia používa prototyp služby Oracle.example.hr, ktorá je založená na protokole REST. Príklady v tejto aplikácii ukazujú, ako filtrovať, ako pridať stránkovanie a ako vytvoriť jednoduchý tabuľkový report o údajoch služby REST.
- Sample Trees – Zistite, ako použiť dotaz SQL na vytvorenie ovládacieho prvku stromu. Aplikácia Oracle APEX v tomto príklade demonštruje niekoľko spôsobov začlenenia stromových ovládacích prvkov.
- Universal Theme Reference - Táto aplikácia, ktorá poskytuje jednoduchý spôsob navigácie v mnohých šablónach, možnostiach šablón a štýloch tém, zoznamuje používateľov s aplikáciou Universal Theme. Príklady ukazujú, ako môžu používatelia rýchlo spravovať rozloženie stránky a vytvoriť tak skvelú aplikáciu.

11.4 Plug-iny

Plug-iny umožňujú ľahko rozšíriť rámec Application Express o vlastné typy položiek, typy oblastí, procesy a dynamické akcie. Po definovaní sa komponenty založené na plug-inoch vytvárajú a udržiavajú podobne ako štandardné komponenty Application Express. Plug-iny umožňujú vývojárom vytvárať vysoko prispôsobené komponenty na zlepšenie funkčnosti, vzhľadu a používateľskej prívetivosti ich aplikácií. Plug-iny umožňujú používateľom pridávať do aplikácií APEX ďalšie funkcie pomocou plug-inov, ktoré nie sú zabudované v platforme. Komunita APEX vyvinula veľké množstvo rôznych plug-inov užitočných na rozšírenie existujúcich funkcií alebo pridanie nových funkcií. Existujú stovky plug-inov pre Oracle APEX, ktoré sú rozdelené do piatich typov:

- Dynamic action (dynamická akcia),
- Region (región),
- Item (položka),
- Process (proces),
- Authentication (autentifikácia).

Oracle Application Express podporuje určitú kolekciu typov položiek, oblastí, dynamických akcií,

procesov, schém overovania a autorizačných schém. Plug-iny dávajú možnosť deklaratívne pridávať nové typy do aplikácie, čím sa rozšíria zabudované typy. Keďže plug-iny sú určené na opakované použitie, vývojári ich môžu zdieľať s komunitou Oracle Application Express Plug-in pomocou úložiska plug-inov, ako aj exportovať a importovať ich do rôznych pracovných priestorov. Pri implementácii plug-inu sa postupuje nasledovne:

- Pridajte plug-in do pracovného priestoru aplikácie alebo ho vytvorte od začiatku.
- Ak chcete použiť plug-in, upravte alebo vytvorte typ dynamickej akcie, položku, oblasť, proces alebo autorizačnú schému.
- Spustite aplikáciu a skontrolujte plug-in.

Centrálnym miestom pre vývojárov na zdieľanie a sťahovanie plug-inov je Plug-in Repository. Úložisko nájdete v sieti Oracle Technology Network.

- Pre zobrazenie úložiska plug-inov prejdite na stránku pre zdieľané komponenty (Shared Components):
  - Kliknite na App Builder na hlavnej stránke pracovného priestoru.
  - Vyberte aplikáciu.
  - Zvoľte Shared Components (zdieľané komponenty) na domovskej stránke aplikácie.
- Zobrazí sa stránka pre zdieľané komponenty.
- Vyberte plug-in z ponuky Other Components.
- Stlačte View Plug-in Repository.

Zobrazí sa úložisko pre Oracle Application Express Plug-iny. Pre prístup na stránku plug-inov:

- Prejdite na stránke Shared Components.
  - Kliknite na položku App Builder na hlavnej stránke pracovného priestoru (workspace).
  - Vyberte aplikáciu.
  - Vyberte položku Shared Components na domovskej stránke aplikácie.
- V ponuke Other Components vyberte položku Plug-ins. Karta Plug-ins sa automaticky vyberie pri načítaní stránky Plug-ins. Zobrazí sa celý zoznam plug-inov.

Pre vytvorenie plug-inu.

- Prejdite na stránku Shared Components.
  - Kliknite na App Builder na hlavnej stránke pracovného priestoru.
  - Vyberte aplikáciu.
  - Na domovskej stránke aplikácie vyberte položku Shared Components. Následne sa zobrazí stránka Shared Components.
- V časti Other Components vyberte položku Plug-ins.
- Kliknite na Create. Zobrazí sa sprievodca vytvorením plug-inu.
- Po výbere metódy vytvorenia plug-inu v časti Create Plug-in klinite na tlačidlo Next.
- V rámci časti Name:
  - Name – Vyplňte názov plug-inu.
  - Internal Name - Vyplňte interný názov plug-inu. Názov by mal byť v rámci aplikácie jedinečný.
  - Type - Rozhodnite, ktorý druh komponentu môže tento plug-in využívať. Možnosti v časti Callbacks (Spätné volania) a Standard Attributes (Štandardné atribúty) sa líšia v závislosti od zvoleného typu plug-inu. Ďalšie informácie nájdete v pomocníkovi (field-level Help).
  - Category - sa zobrazí len vtedy ak je vybratá dynamická akcia. V používateľskom rozhraní vyberte kategóriu, v ktorej má byť zásuvný modul uvedený.
- V rámci Subscription reference Master Plug-in From - Vyberte plug-in zo zoznamu, ktorý sa má použiť ako vzor pre ostatné plug-iny tohto pracovného priestoru. Ak nechcete, aby to bola hlavná kópia tohto plug-inu, nechajte pole prázdne.
- V rámci časti Source:
  - PL/SQL Code - Zadajte PL/SQL anonymný blok kódu, ktorý obsahuje kroky potrebné

na vykreslenie, overenie, spustenie a vykonanie spätných volaní tohto plug-Ajax inu. Tento kód môžete z výkonnostných dôvodov alternatívne uložiť do balíka PL/SQL v databáze.
  - Do not validate PL/SQL code (parse PL/SQL code at runtime only) - Ak chcete analyzovať kód PL/SQL len za behu, vyberte túto možnosť. Ak nie, zásuvný modul sa zostaví po rozparsovaní kódu.
- V rámci Callbacks, nakonfigurujte príslušné atribúty. Typ plug-inu určuje atribúty, ktoré sa zobrazia. Pozrite si nápovedu na úrovni polí, aby ste preskúmali ukážky a dozvedeli sa viac o atribútoch.
- V rámci User Interfaces, vyberte zobrazenia, ktoré musí App Builder podporovať pre tento plug-in. Medzi možnosti patrí:
  - Desktop
  - Mobile
- Pre Standard Attributes, vyberte atribúty, ktoré sa vzťahujú na tento plug-in. Štandardné atribúty sa zaobídu bez určitých plug-inov.
- V rámci Information:
  - Version - Ak chcete určiť verziu plug-inu, zadajte reťazec.
  - About URL - Zadajte adresu URL vedúcu na domovskú stránku plug-inu alebo na ďalšie podrobnosti o plug-ine.
- Kliknite na Create Plug-in.

Pre importovanie plug-inu:

- Prejdite na stránku Shared Components:
  - Kliknite na App Builder na hlavnej stránke pracovného priestoru.
  - Vyberte aplikáciu.
  - Na domovskej stránke aplikácie vyberte položku Shared Components.
- Vyberte Plug-ins v rámci Other Components.
- Vyberte Import. Zobrazí sa stránka pre importovanie plug-inu.
- V rámci časti Specify File:
  - Import file - Zadajte alebo prejdite na názov importovaného súboru.
  - File Type - Vyberte Plug-in.
  - File Character Set - Vyberte kódovanie znakovej sady pre importované súbory.
  - Kliknite na Next.
- Pri File Import Confirmation, vyberte Next.
- V časti Install, vyberte Install Plug-in.

Pre exportovanie plug-inu zo stránky Plug-in:

- Prejdite na stránku Shared Components:
  - Na domovskej stránke Workspace kliknite na položku App Builder.
  - Vyberte aplikáciu.
  - Na domovskej stránke aplikácie kliknite na položku Shared Components.
- V rámci Other Components, vyberte Plug-ins.
- V časti Tasks, vyberte Export Plug-in.
- Na stránke Export Plug-in:
  - Application - Vyberte aplikáciu, z ktorej chcete exportovať zásuvný modul.
  - Plug-in - Zvoľte plug-in.
  - File Format - Vyberte formát súboru exportu.
  - File Format - Vyberte formát súboru exportu.
  - Zvoľte Export.
- Následne zvoľte opäť Export.

Príklad: Aktualizovaná ukážka aplikácie kalendára a dynamického akčného plug-inu. Tento vzorový plug-in možno použiť na formátovanie dní v novej položke výberu dátumu. Používatelia môžu

vypnúť, pridať tooltipy alebo triedy ku každému dňu v mriežke dní v položke výberu dátumu. Na používanie tohto plug-inu nie sú žiadne predpoklady. Ako ho používať:

- Vytvorte aplikáciu
- Importujte plug-In v časti Shared Components - Plug-Ins
- Pridajte nový vyberač dátumu na stránku APEX
- Pridajte dynamickú akciu pri načítaní stránky a ako akciu vyberte importovaný plug-in
- Nastavte URL adresu ICS súboru alebo zadajte SQL Query, ktorý by mal formátovať kalendárne dni
- Vyberte ako Affected Element požadovaný vyberač dátumu a spustite stránku

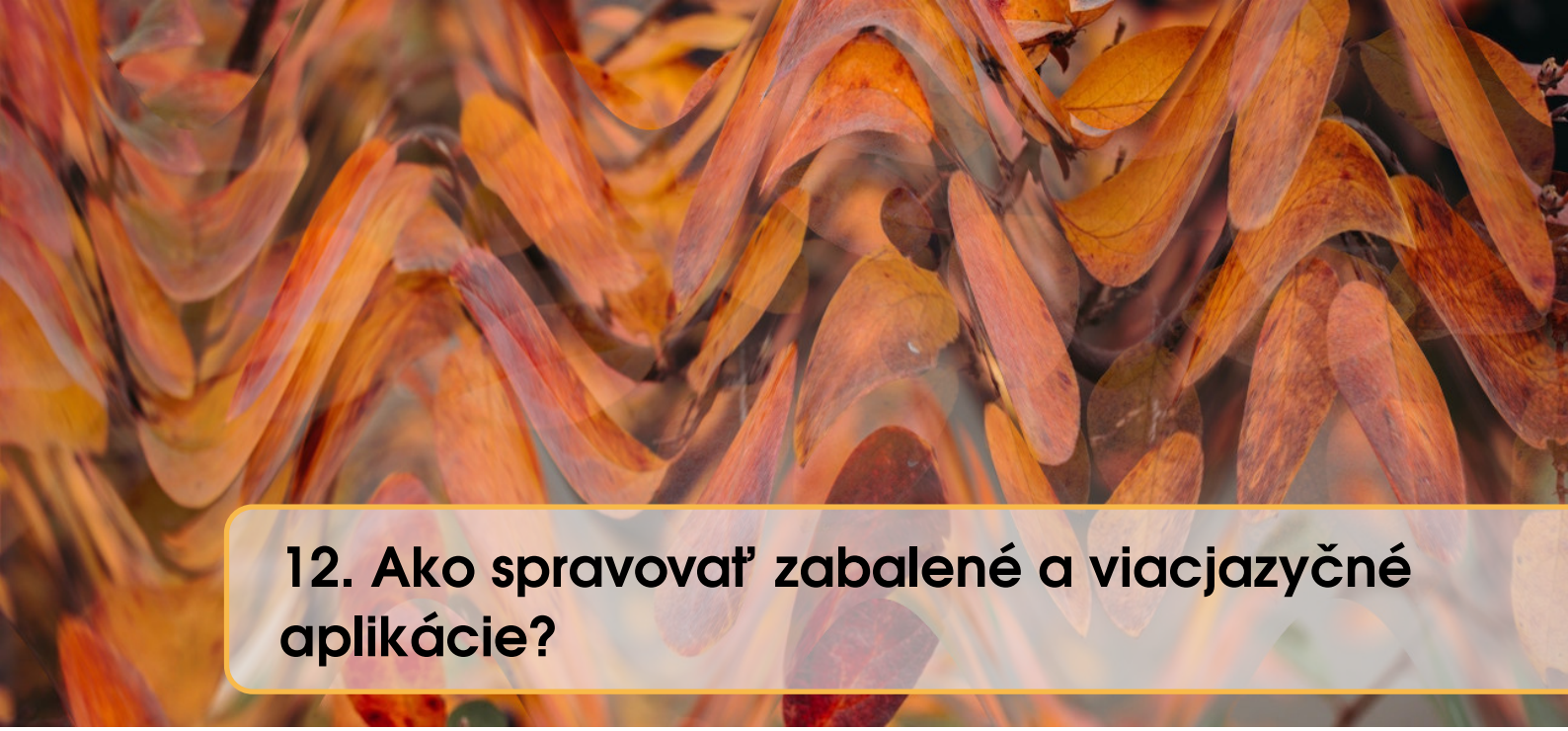
Plug-iny môžu ponúkať niekoľko výhod. Bez ohľadu na to, aký silný môže byť, žiadny softvér nemôže poskytovať všetky funkcie pre každého používateľa. Medzeru medzi formou a funkciou vypĺňajú plug-iny. Vďaka nim je jednoduchšie pridať do softvéru a aplikácií konkrétne funkcie. Plug-iny ako také šetria čas. Najväčšou výhodou ich používania je, že skracujú čas potrebný na vývoj, čo znižuje celkové náklady. Okrem toho umožňujú používateľom prispôsobiť vlastnosti a funkcie. Väčšina plug-in-ov umožňuje zapínať a vypínať konkrétne nastavenia. V prípade potreby je ich možné jednoducho odstrániť.

11.5 Otázky

1. Čo sú to vzorové aplikácie a aké sú ich výhody?
2. Čo sú to štartovacie aplikácie a aké sú ich výhody?
3. Aké sú výhody pluginov?

11.6 Odpovede

1. Vzorové aplikácie sú vytvorené na zdôraznenie konkrétnych možností a slúžia ako príručka pre vývojárov na používanie konkrétnej funkcie.
2. Štartovacie aplikácie sú užitočné aplikácie, ktoré ponúkajú samostatné bodové riešenia, vytvorené na uspokojenie jednoduchých požiadaviek, ktoré nevyžadujú podstatnú a zbytočnú komplexnú odpoveď.
3. Pluginy umožňujú vývojárom vytvárať vysoko prispôsobené komponenty na zlepšenie funkčnosti, vzhľadu a používateľskej prívetivosti ich aplikácií. Plug-iny umožňujú používateľom pridať do aplikácií APEX ďalšie funkcie pomocou plug-inov, ktoré nie sú zabudované v platforme.



12. Ako spravovať zabalené a viacjazyčné aplikácie?

R. Leskovar, U. Rajkovič, A.Baggia

12.1 Aplikácia a zabalená aplikácia

Táto časť vás naučí ako využiť potenciál funkcie Quick SQL na generovanie aplikácie takmer bez znalosti DDL a úplne v rámci Oracle APEX. Po vygenerovaní aplikácie z Quick SQL pripravíme zabalenú aplikáciu a premigrujeme ju do nového pracovného priestoru (t.j. "production site").

V Kapitole 11 sme predstavili aplikácie, ktoré sú dodané v Oracle APEX. Tieto aplikácie sú rozdelené do dvoch skupín: a) produktívne a b) vzorové aplikácie. Aplikácie produktivity riešia špecifické obchodné potreby. Vstavané funkcie viac-menej úplne spĺňajú požiadavky kladené obchodnou situáciou. Vzorové aplikácie demonštrujú rôzne funkcie. Obe môžu inšpirovať vývojárov, aby ich prispôsobili a splnili obchodné potreby. Tieto aplikácie sú príkladmi zabalených aplikácií. Keď inštalujete (APEX pojem je import) tieto zabalené aplikácie, môžete si všimnúť, že sa vytvárajú databázové objekty (tabuľky, sekvencie, triggre, funkcie, procedúry a balíky) a údaje sú naplnené do tabuliek.

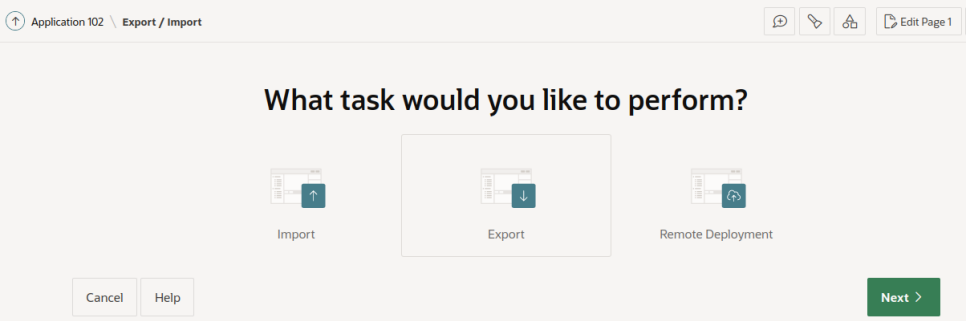
Najkratšia definícia **zabalenej aplikácie** hovorí, že ide o vpred zostavenú aplikáciu, ktorú je možné nainštalovať a nakonfigurovať v pracovnom priestore Oracle APEX. APEX používa dva dôležité koncepty, ktoré úzko súvisia s migráciou aplikácií (export a import): a) podporné objekty a b) zdieľané komponenty. Podporné objekty riadia inštaláciu, aktualizáciu a odinštalovanie definovaním prerekvizít, substitučných reťazcov, vstavaných možností, predinštalačných overení, skriptov a správ. Zdieľané komponenty definujú aplikačnú logiku, bezpečnosť, navigáciu a vyhľadávanie, používateľské rozhranie, súbory a zostavy, zdroje údajov, pracovné postupy (ak sú nainštalované), globalizáciu a ďalšie komponenty. Zabalené aplikácie sú teda tie, ktoré majú definované podporné objekty. V tejto chvíli si možno položíte otázku: „Prečo a ako môžu vývojári vytvoriť vlastnú zabalenú aplikáciu?".

Odpoveď na otázku „prečo": vývojári by vytvorili vlastnú zabalenú aplikáciu, ak existuje požiadavka na migráciu aplikácie do nového pracovného priestoru s podpornými objektmi a zdieľanými komponentmi. Definície databázových objektov (s príkazmi DDL) a operácie na týchto objektoch (príkazy DML) sa vykonávajú počas inštalácie, aktualizácie a odinštalovania. Zjednodušene: počas inštalácie bolo možné vytvárať tabuľky a napĺňať ich údajmi, počas aktualizácie bolo možné tabuľky upravovať a počas odinštalovania možno tabuľky orezávať alebo rušiť. To isté by sa dalo vynútiť aj pre iné databázové objekty, ako sú triggre, sekvencie, uložené procedúry, funkcie a balíky. Kľúčovou výhodou zabalenej aplikácie je, že je pripravená na spustenie s definíciami údajov a samotnými údajmi.

Odpoveď na otázku „ako“ obsahuje nasledujúce kroky:

- definovanie rozsahu aplikácie
- vytvorenie a spustenie skriptu na definovanie tabuliek
- vytvorenie a spustenie skriptu na vloženie vzorových údajov
- generovanie aplikácie
- vykonanie úprav v aplikácii
- pripravenie zabalenej aplikácie a jej otestovanie v novom pracovnom priestore

Skôr než prejdeme krok za krokom, pozrime sa na sprievodcu importom a exportom aplikácie v APEX. Ak chcete aplikáciu presunúť do nového pracovného priestoru, aplikáciu je potrebné exportovať a následne importovať. Sprievodca exportom v APEX zapíše textový súbor (príkazy PL/SQL) na: a) počítač vývojára alebo b) do vzdialeného pracovného priestoru prostredníctvom služby REST (Obrázok 12.1).



Obr. 12.1: Sprievodca importom a exportom aplikácie.

Existuje niekoľko možností exportu, ako je rozdelenie do viacerých súborov, výber znakovej sady súboru, aktivácia alebo deaktivácia verejných alebo súkromných správ a podobne. Vo väčšine prípadov začiatočník jednoducho nechá APEX použiť predvolené nastavenia. Potom sa vývojár prihlási do nového pracovného priestoru a spustí sprievodcu importom, ktorý načíta exportovaný textový súbor. Definície tabuliek, pohľadov, triggrov, sekvencií, funkcií, procedúr a balíkov sa s aplikáciou nemigrujú, ak nepripravíme podporné objekty. Ak cieľový pracovný priestor už obsahuje definície databázových objektov a údaje v tabuľkách, mali by sme pripraviť podporné objekty na zvládnutie takýchto situácií.

V prvej časti tejto kapitoly sa budú čo najviac využívať sprievodcovia APEX. Potom budú prezentované jednoduché príkazy SQL INSERT, ktoré sprievodca v APEX nepodporuje. Na generovanie príkazov INSERT môže čitateľ importovať údaje v tabuľke a potom exportovať príkazy INSERT pomocou iných nástrojov (napr. SQL Developer alebo TOAD). Aplikácia v tejto kapitole predstaví komponent stránky, ktorý sa nazýva zoznamy hodnôt (LOV - list of values). Patrí k zdieľaným komponentom. Zoznam hodnôt má dva účely: používateľ jednoducho vyplní pole formulára na webovej stránke (rozbaľovací zoznam, prepínač) a iba predpísanými hodnotami, aby sa vyhol preklepom. K dispozícii sú dva typy zoznamov hodnôt: dynamické sú založené na SQL dotaze, zatiaľ čo statické ponúkajú pevnú množinu hodnôt. Výhodou dynamického zoznamu hodnôt je, že doména závisí od tabuľky alebo pohľadu. Koncový používateľ môže zmeniť doménu vložením, aktualizáciou a odstránením riadkov v tabuľke. Nevýhodou je, že tabuľka musí byť vytvorená a udržiavaná. Ak sa doména nezmení a počet riadkov v tabuľke je malý, je možné použiť statický zoznam hodnôt. V takom prípade môže doménu zmeniť iba vývojár (je “napevno naprogramovaná”).

Druhá časť bude demonštrovať vytvorenie zabalenej aplikácie, jej exportovanie a importovanie do nového pracovného priestoru.

Tretia časť kapitoly bude obsahovať návod na vytvorenie viacjazyčnej aplikácie. Ako doplnkový študijný materiál sú poskytnuté tri exportované aplikácie, ako aj videoklip, ktorý detailne zobrazuje

proces vývoja.

12.2 Aplikácia

12.2.1 Rozsah aplikácie

Aplikácia je zameraná na riadenie pracovných miest a kompetencií pre HR oddelenie. Vo firme môže byť niekoľko stoviek pracovných miest. Jedno pracovné miesto môže zahŕňať špecifickú podskupinu všetkých známych kompetencií. Požadovanú úroveň kompetencií predpisuje HR oddelenie pre každú pracovnú pozíciu. Úrovne kompetencií sú opisné schopnosti: pomáhať, používať, ovládať, prispôsobovať a inovovať. Myšlienkou je teda vyvinúť aplikáciu na prípravu a správu pracovných profilov. Aplikácia využíva tri tabuľky: JOB, COMPETENCE a JOB_COMPETENCE s minimálnym počtom dátových polí (kvôli jednoduchosti a stručnosti). Každá tabuľka má primárny kľúč a tabuľka JOB_COMPETENCE má dva cudzie kľúče na spojenie každého riadku s riadkami JOB a tabuľkou COMPETENCE. Cudzie kľúče zabezpečujú konzistenciu údajov, konkrétne riadok v tabuľke JOB_COMPETENCE môže existovať iba vtedy, ak v tabuľkách JOB a COMPETENCE existujú riadky s rovnakými primárnymi kľúčmi. Aby sa zabezpečila malá presnosť, názvy tabuliek majú predponu "CH12_".

12.2.2 Vytvorenie tabuliek

Môžeme vytvárať tabuľky v APEX:

- bez znalosti jazyka popisu údajov kliknutím v prehliadači objektov. Ale ilustrovať tento proces množstvom obrázkov by v tejto knihe zabralo príliš veľa miesta.
- so znalosťou DDL v SQL Workshop > Príkazy SQL zadaním syntakticky správnych príkazov.
- so základnými znalosťami dátového modelovania (Kapitola 2) a Quick SQL.

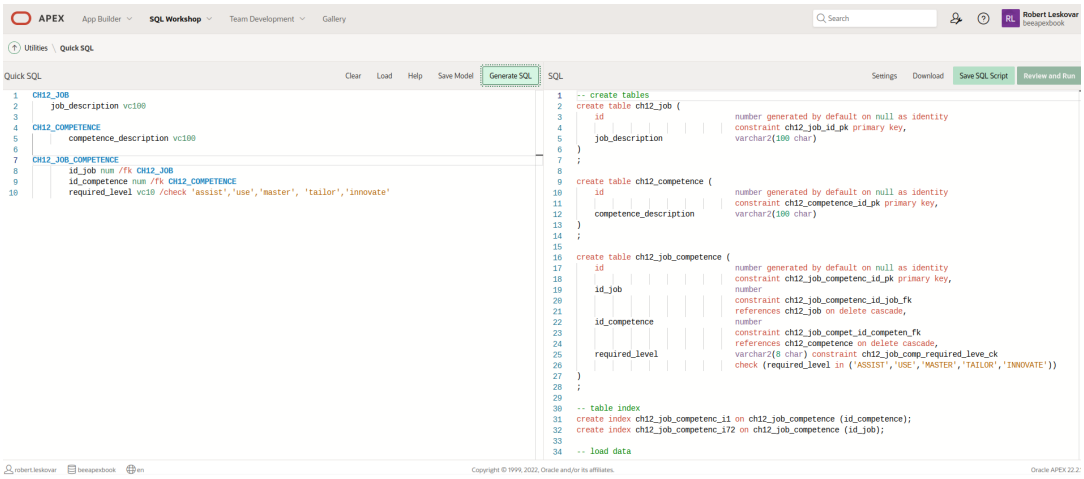
Na zabezpečenie širokého pochopenia a stručnosti sa použije Quick SQL (nájdite Quick SQL v časti SQL Workshop > Utilities > Quick SQL). Vloží sa nasledujúci text:

```
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 zadaní vyššie uvedeného textu a stlačení tlačidla **Generate SQL** sa zobrazí nasledujúca obrazovka (Obrázok 12.2):

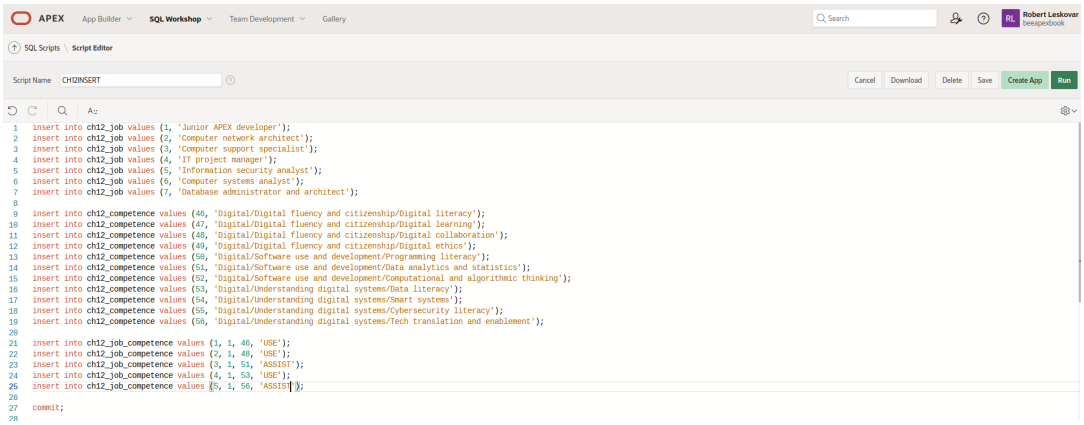


Obr. 12.2: Transformácia Quick SQL na SQL príkazy.

Potom kliknite na tlačidlo **Uložiť SQL skript** a pomenujte skript ako CH12CREATE. Teraz môžeme spustiť skript. Vráťte sa do Skriptov, nájdite ikonu Run v riadku, kde je zobrazený skript CH12CREATE. Kliknite na ikonu **Run**. Tri tabuľky a tri indexy by mali byť úspešne vytvorené. Tabuľky si môžete pozrieť v prehliadači objektov.

12.2.3 Vloženie dát

V APEX vyberte SQL Workshop > SQL Scripts > Create. Teraz je čas napísať niekoľko príkazov SQL INSERT na vyplnenie údajov do tabuliek. Tabuľka CH12_JOB bude mať sedem riadkov, z ktorých každý bude popisovať odlišnú pracovnú pozíciu. Tabuľka CH12_COMPETENCE bude mať jedenásť riadkov popisujúcich jedenásť digitálnych kompetencií definovaných McKinsey's DELTAs (výrazné prvky talentu). A dáme pokyn vložiť päť kompetencií do pozície 'Junior Apex developer'. Pomenujme skript CH12INSERT a napíšme skript (Obrázok 12.3).

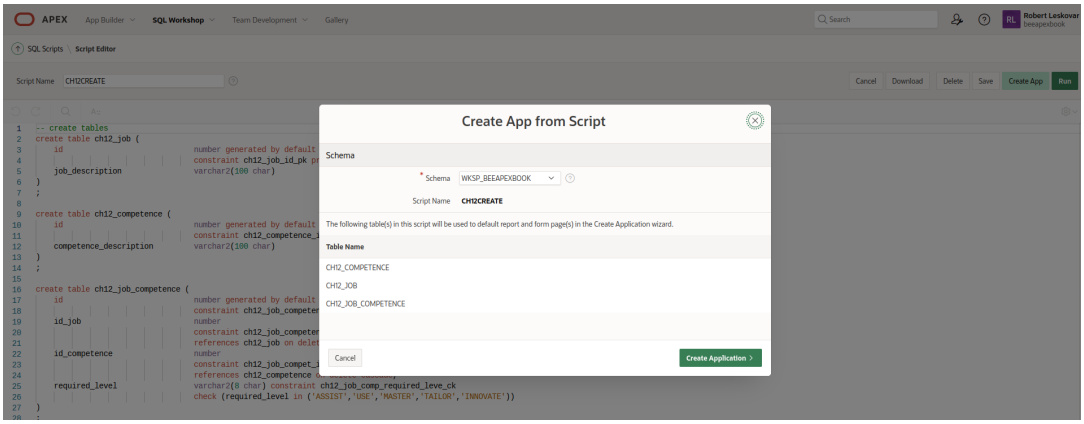


Obr. 12.3: Skript CH12INSERT vloží údaje do troch tabuliek a potvrdí transakcie.

Kliknite na tlačidlo **Create**. Teraz spustite skript CH12INSERT. Vráťe sa do Skriptov, kliknite na ikonu **Run** v riadku, kde je zobrazený skript CH12INSERT. Kliknite na tlačidlo **Run now**. Úspešne by sa malo vytvoriť dvadsaťštyri riadkov. Obsah tabuliek môžete skontrolovať aj v prehliadači objektov.

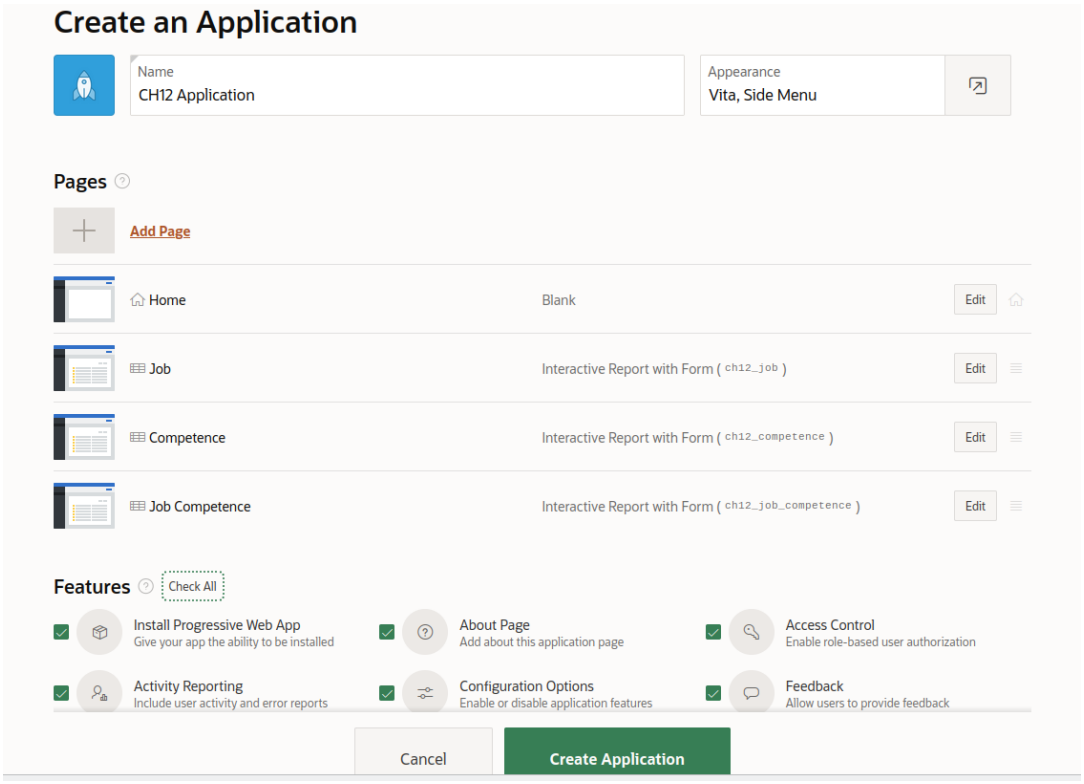
12.2.4 Generovanie aplikácie

Po naplnení tabuliek testovacími údajmi prejdite do SQL Workshop > Scripts a upravte skript CH12CREATE. Kliknite na tlačidlo **Create App** (Obrázok 12.4). Môžete si všimnúť, že sú zobrazené tri tabuľky.



Obr. 12.4: Vytvorenie aplikácie zo skriptu.

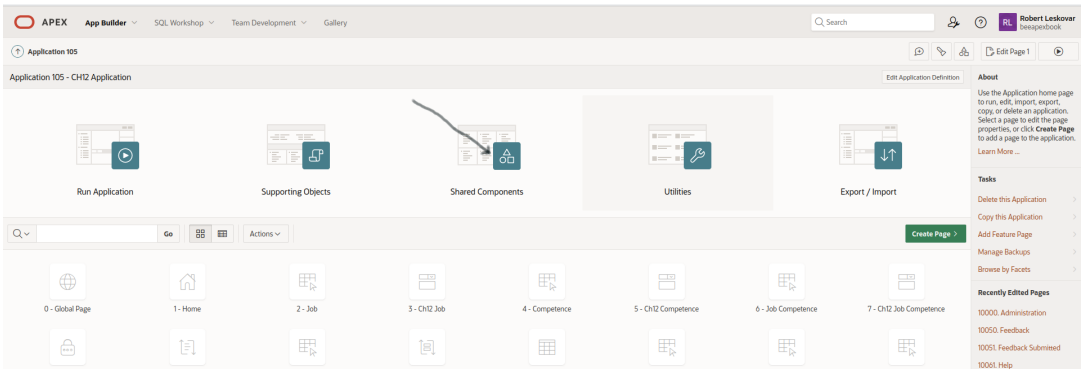
Kliknite na tlačidlo Create Application (Vytvoriť aplikáciu). Zadajte názov aplikácie (t.j. CH12 Application), kliknite na Check all features a potom na tlačidlo Create Application (Obrázok 12.5).



Obr. 12.5: Výber názvu aplikácie a všetkých funkcií.

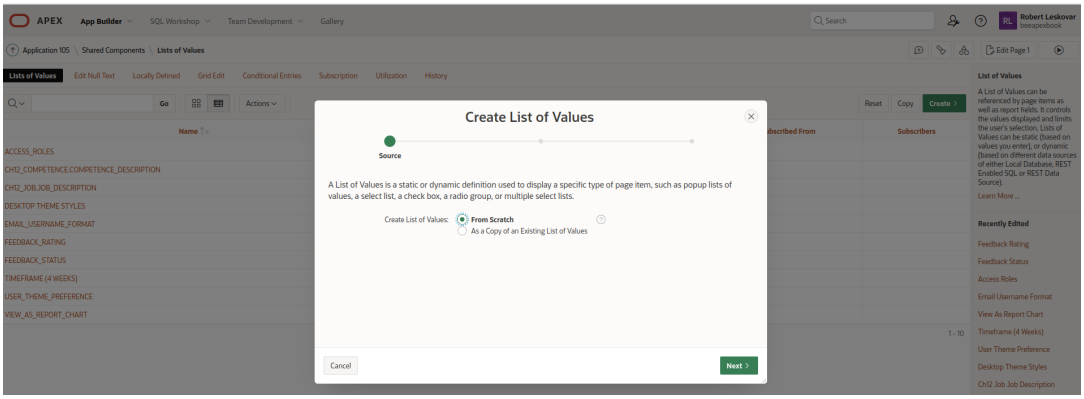
Po dokončení generovania môžeme spustiť aplikáciu. V novších verziách APEX generátor vytvára aj dva zoznamy hodnôt (CH12_COMPETENCE.COMPETENCE_DESCRIPTION a CH12_JOB.JOB_DESCRIPTION), skrátene LOV (list of values). Zoznam hodnôt sú komponenty

na stránke, kde je jedna hodnota zobrazená a pochopená používateľom, pričom stránka v skutočnosti používa zodpovedajúci kľúč alebo kód na vloženie, aktualizáciu alebo odstránenie. Napríklad: na stránke formulára sa po stlačení potvrdzovacieho tlačidla zobrazí priezvisko zamestnanca a príslušné identifikačné číslo zamestnanca. Vygenerované hodnoty LOV sú založené na tabuľkách a nie sú potrebné žiadne predchádzajúce znalosti SQL. V terminológii APEX sa nazývajú dynamické, pretože zmena v tabuľke by viedla k novému zoznamu hodnôt. Vygenerované LOV môžete nájsť v časti Shared Components > List of Values a určiť ako sú nastavené. V tejto kapitole sa naučíme, ako vytvoriť dynamický zoznam hodnôt (založený na jednoduchom SQL dotaze) a statický na základe pevnej množiny hodnôt. Aplikovaním zoznamu hodnôt (LOV) na stránky aplikácie urobíme aplikáciu odolnejšou voči chybám a tiež užívateľsky príjemnejšou. To znamená, že používateľ nemôže vyplniť ľubovoľnú hodnotu v poli formulára, čím sa obmedzia preklepy. Prejdite na časť Shared components (Obrázok 12.6).



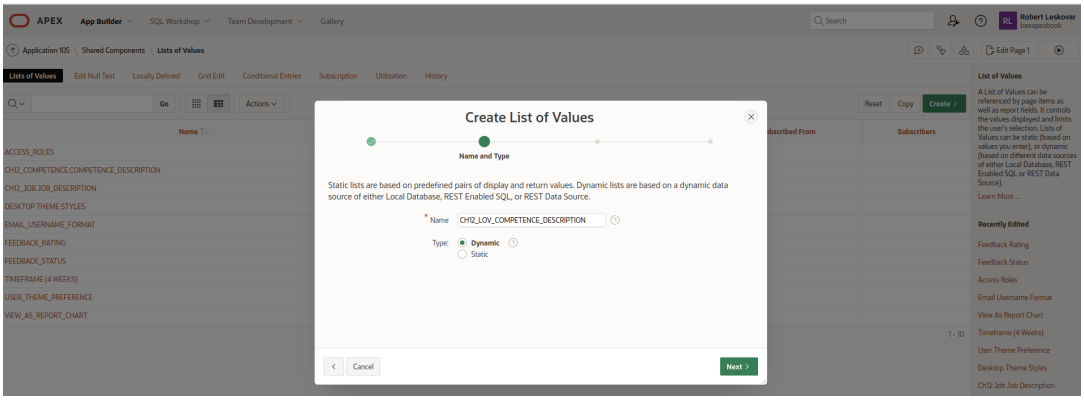
Obr. 12.6: Výber Zdieľaných komponentov.

V zdieľaných komponentoch (Shared components) vyberte zoznam hodnôt (List of values) (v časti Other Components) a kliknite na tlačidlo Create. Vytvoríme zoznam hodnôt úplne od začiatku (Obrázok 12.7).



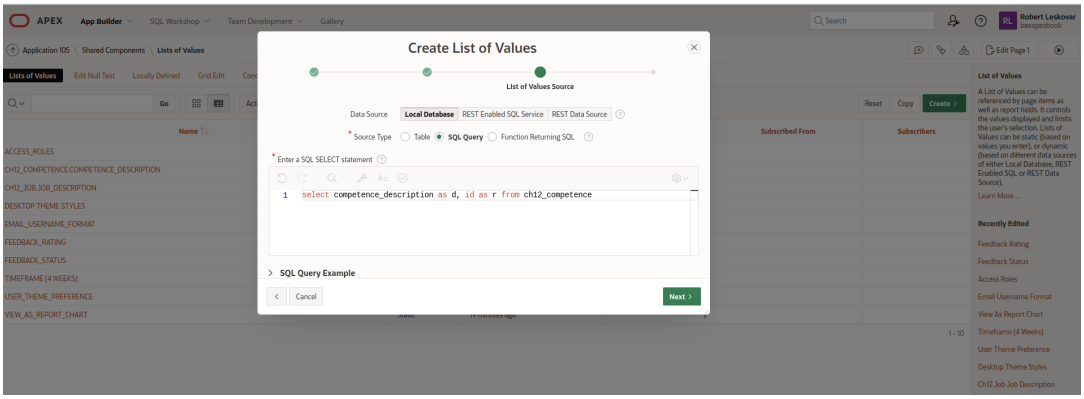
Obr. 12.7: Vytvorenie zoznamu hodnôt od začiatku.

Pomenujte LOV ako CH12_LOV_COMPETENCE_DESCRIPTION a vyberte Dynamický typ (Obrázok 12.8)



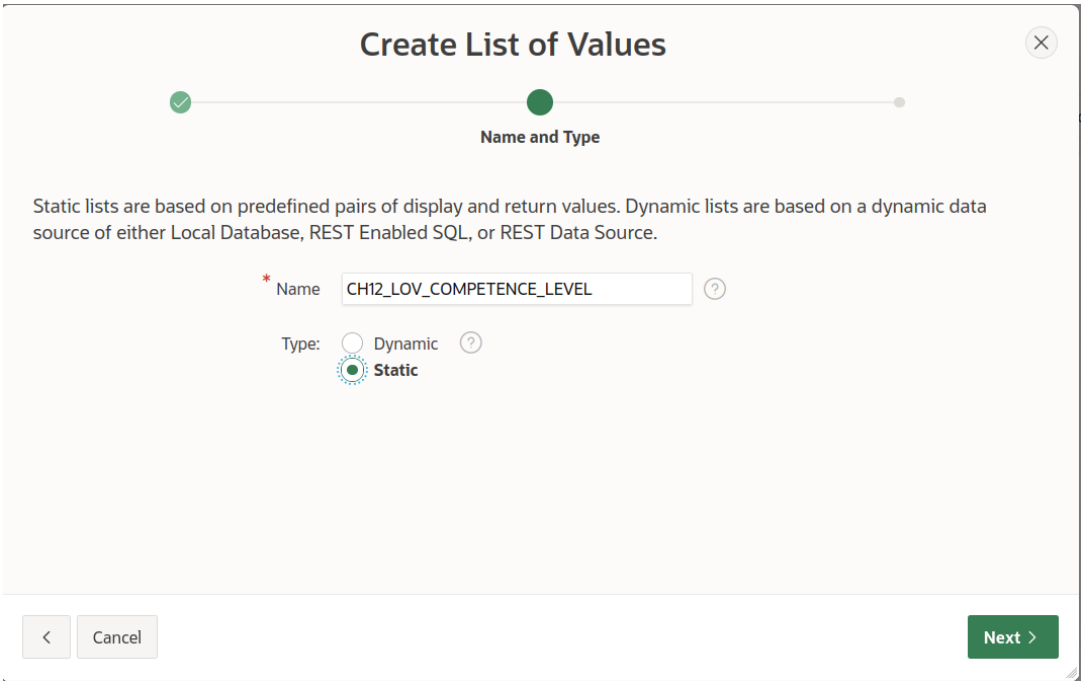
Obr. 12.8: Definovanie názvu a typu: CH12_LOV_COMPETENCE_DESCRIPTION je dynamický.

Vyberte SQL Query ako typ zdroja a zadajte dotaz “select competence_description as d, id as r from ch12_competence” ako príkaz SQL (Obrázok 12.9).

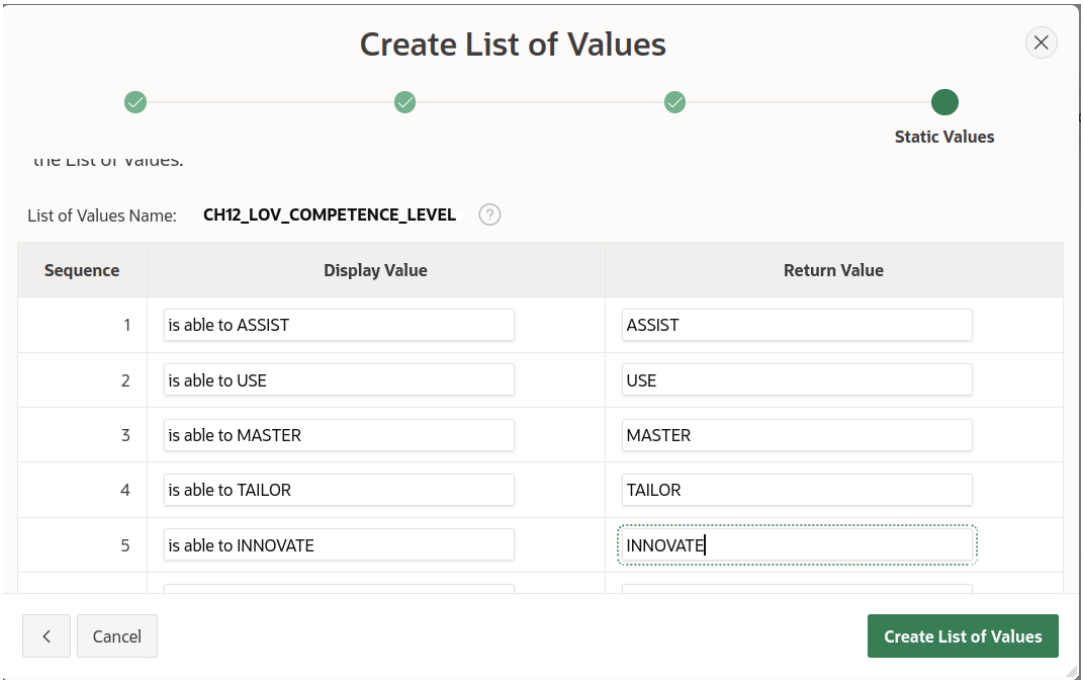


Obr. 12.9: Zadanie príkazu SQL SELECT.

Tento zoznam hodnôt zobrazí popis kompetencie (stĺpec zobrazenia) na stránke formulára, pričom vráti ID (stĺpec návratu). Ďalšia LOV bude statická s názvom CH12_LOV_COMPETENCE_LEVEL a bude vytvorená tak, ako je znázornené na Obrázku 12.10 a Obrázku 12.11.

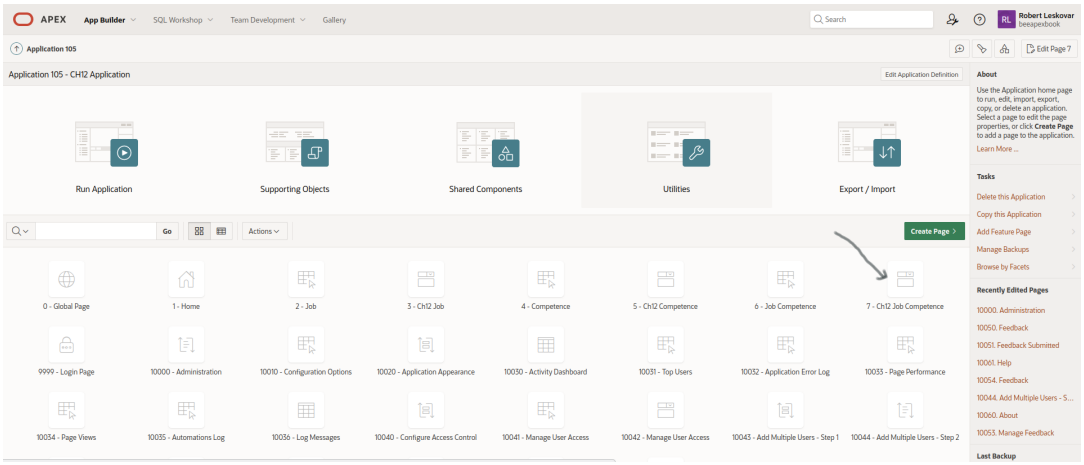


Obr. 12.10: Definovanie názvu a typu: CH12_LOV_COMPETENCE_LEVEL je statický.



Obr. 12.11: Definovanie zobrazovaných a návratových hodnôt: CH12_LOV_COMPETENCE_LEVEL je statický.

Prejdite na stránky aplikácie (Obrázok 12.12). Vyberte stranu 7 (Ch12 Job Competences).



Obr. 12.12: Výber strany 7 (Ch12 Job Competences).

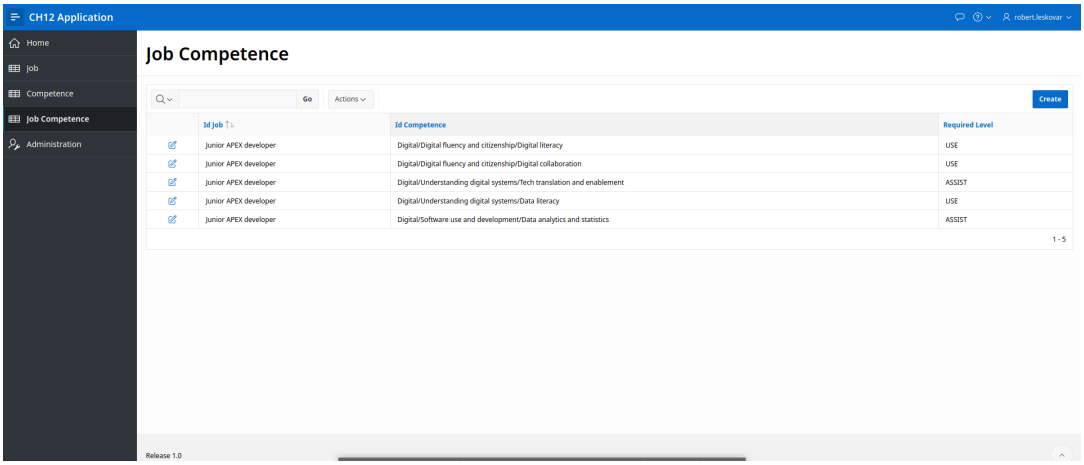
Upravte položku stránky P7_ID_COMPETENCE:

- typ je “Select list” v časti Identification
- typ je “Shared components” v časti List of values
- časť list of values je CH12_LOV_COMPETENCE_DESCRIPTION List of values

Upravte položku stránky P7_REQUIRED_LEVEL:

- typ je “Select list” v časti Identification
- typ je “Shared components” v časti List of values
- časť list of values je CH12_LOV_COMPETENCE_LEVEL List of values

Uložte stránku 7, prejdite na stranu 6 a spustite aplikáciu. Mali by ste vidieť Obrázok 12.13.

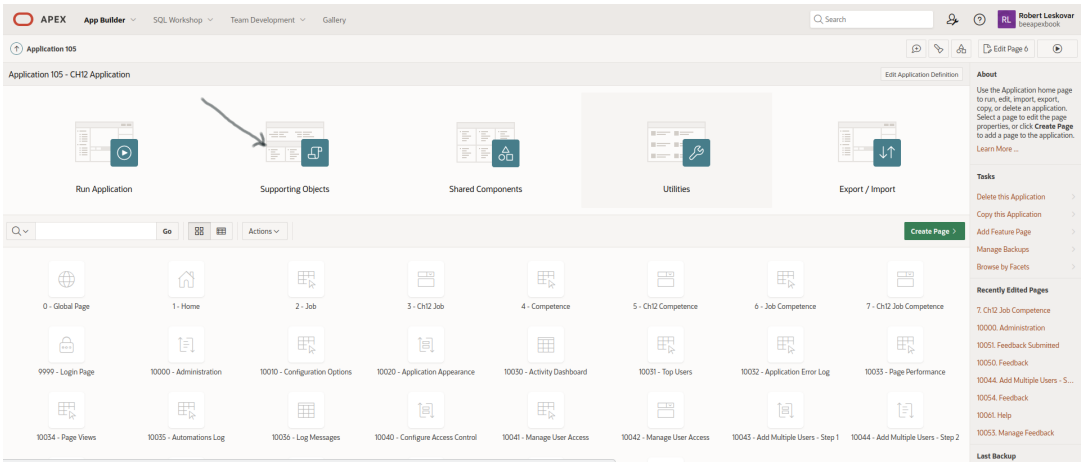


Obr. 12.13: Výber strany 7 (Ch12 Job Competences).

Teraz upravte prvý záznam. Obidva vybrané zoznamy poskytujú očakávané údaje.

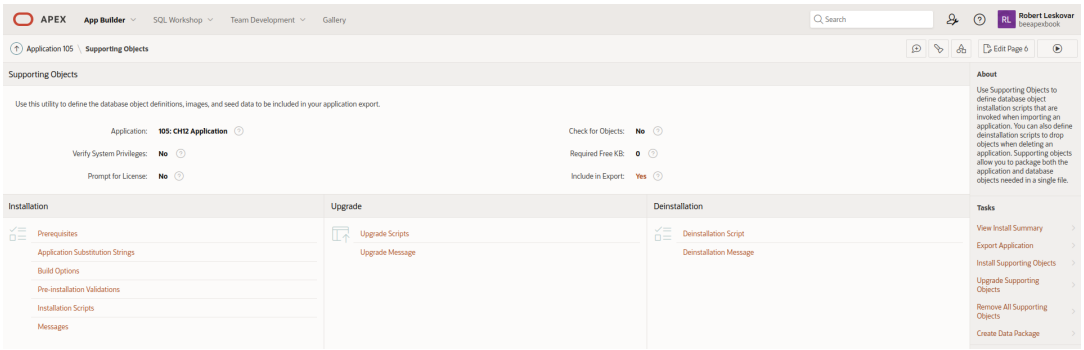
12.3 Zabalená aplikácia

Keďže už máme vytvorené nejaké skripty (CH12CREATE, CH12INSERT), vytvoríme zabalenú aplikáciu veľmi rýchlo. Prejdite do aplikácie a vyberte Supporting Objects (Obrázok 12.14).



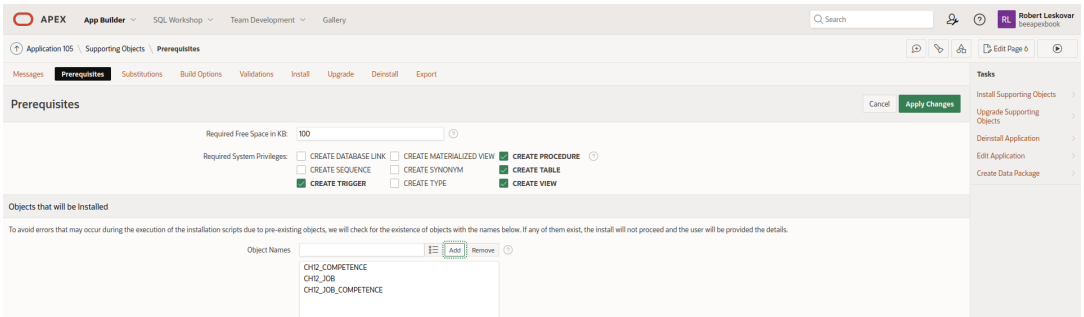
Obr. 12.14: Výber podporných objektov.

Teraz môžeme nastaviť inštaláciu (prerekvizity, substitučné reťazce aplikácie, možnosti zostavenia, overenie pred inštaláciou, inštalačné skripty a správy), inováciu (skripty aktualizácie, správa o aktualizácii), odinštalovanie (deinštalačný skript, správa o odinštalovaní). Pozrite si Obrázok 12.15.



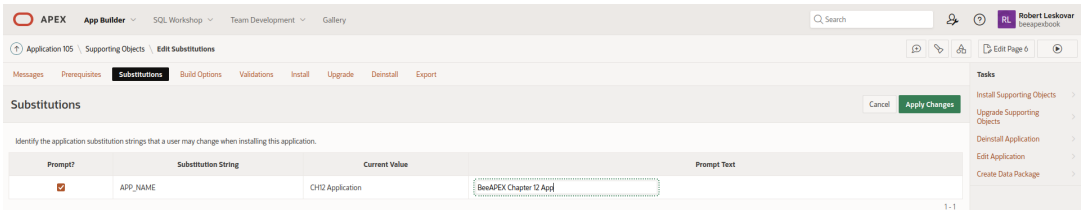
Obr. 12.15: Nastavenie prerekvizít.

Na ochranu používateľa pred náhodným vymazaním existujúcich údajov použijeme kontrolu existencie troch tabuliek (Obrázok 12.16). Kliknite na tlačidlo Apply changes.



Obr. 12.16: Nastavenie kontroly na existenciu troch tabuliek.

V reťazci Application Substitution String sa opýtame, či má byť názov aplikácie “BeeAPEX Chapter 12 App” (Obrázok 12.17).



Obr. 12.17: Nastavenie výzvy na premenovanie aplikácie.

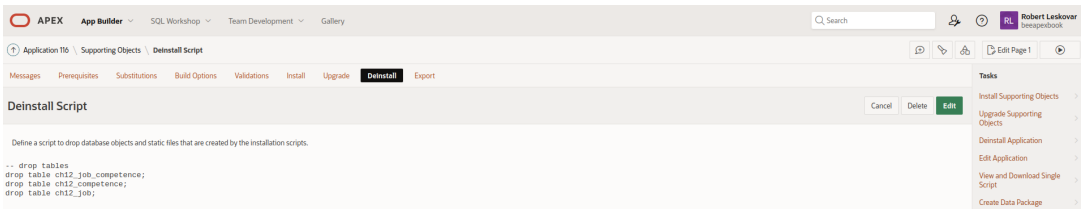
Kliknite na tlačidlo Apply changes.

Pre túto zabalenú aplikáciu použijeme aj nasledujúce nastavenie v časti Supporting Objects:

- prerekvizity (Obrázok 12.16)
- substitučné reťazce aplikácie (Obrázok 12.17)
- inštalačné skripty: keďže sme už pripravili skripty CH12CREATE a CH12INSERT použijeme ich obsah (Obrázok 12.18)
- deinštalačné skripty (Obrázok 12.19)
- nastavené správy: uvítacia správa pre “Chapter 12: ...”, licenčná správa pre “CC BY”, správa o úspešnej inštalácii na “Success!”, správa o zlyhaní inštalácie na “Failure”, správa o odinštalovaní na “Application CH12 deinstalled”

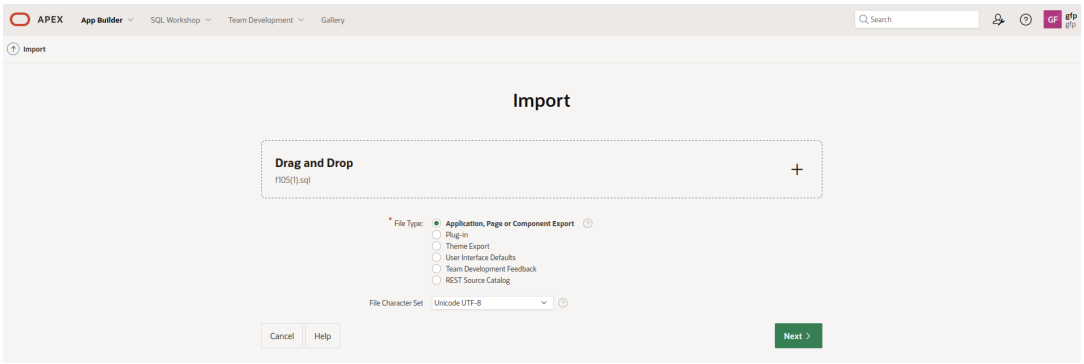


Obr. 12.18: Nastavenie inštalačných skriptov.



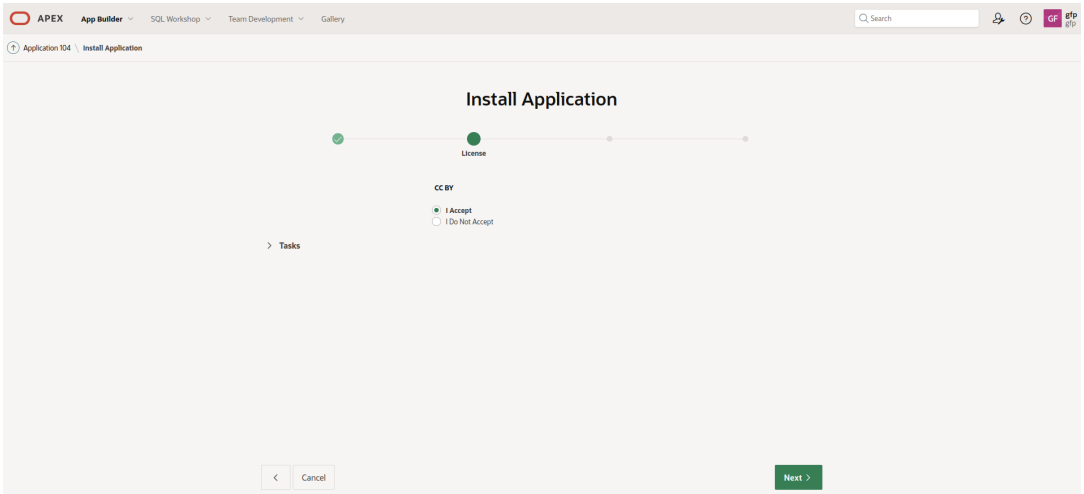
Obr. 12.19: Nastavenie deinštalačných skriptov.

Teraz exportujte zabalenú aplikáciu pomocou sprievodcu (Obrázok 12.1). Môžete akceptovať všetky nastavenia exportu navrhnuté APEX. Uložte exportovanú aplikáciu do počítača a zapamätajte si názov súboru. Prihláste sa do nového pracovného priestoru. Kliknite na Sprievodcu importom v novom pracovnom priestore a presuňte názov súboru do oblasti drag and drop (Obrázok 12.20).

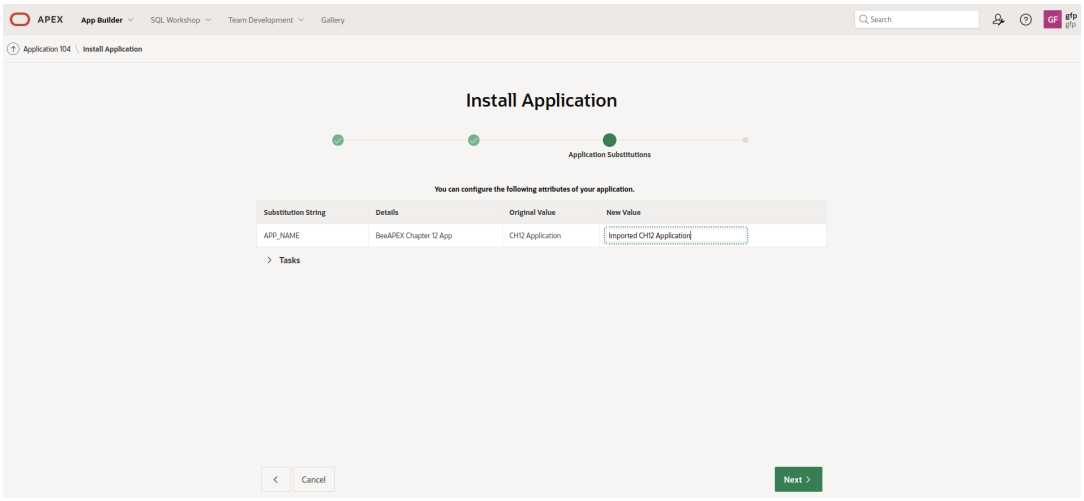


Obr. 12.20: Import aplikácie do iného pracovného priestoru.

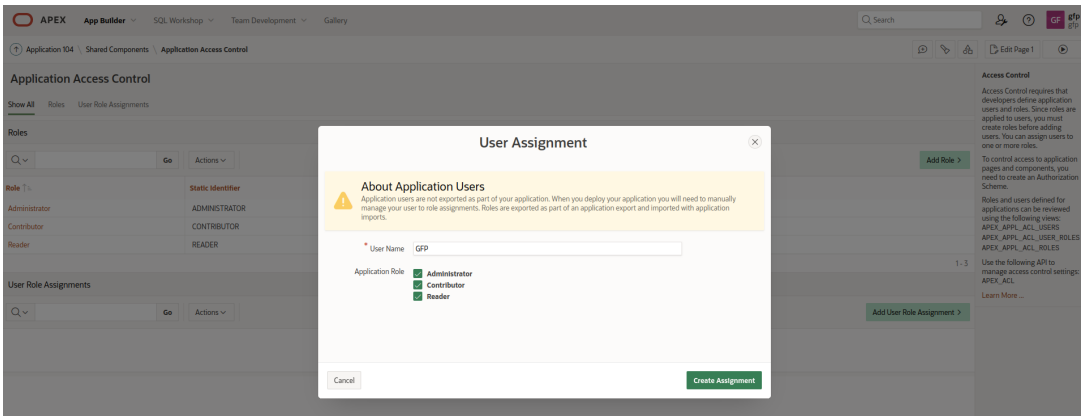
Pokračujte kliknutím na tlačidlo Next (tiež na potvrdenie importu súboru) a kliknutím na tlačidlo Install application. Potvrďte inštaláciu podporných objektov tlačidlom Next, akceptujte licenciu (Obrázok 12.21), premenujte aplikáciu na "Imported CH12 Application" (Obrázok 12.22) and kliknite na tlačidlo Install. Aplikáciu zatiaľ nespúšťajte, pretože pre nového používateľa musíte nastaviť oprávnenia samostatne. Používatelia a používateľské oprávnenia sa samozrejme neexportujú z bezpečnostných dôvodov. Len jedna poznámka: na migráciu používateľov a oprávnení by sme sa mohli hlboko ponoriť do APEX API a písať inštalačné a deinštalačné skripty. V importovanej aplikácii prejdite na zdieľané komponenty (Shared components). Vyberte application access control a pridelťte aktuálne prihlásenému používateľovi príslušné oprávnenia. V našom prípade pridáme APEX používateľovi rolu GFP administrátora, prispievateľa a čitateľa (Obrázok 12.23). Potvrďte kliknutím na tlačidlo Add Assignment. Teraz spustite aplikáciu ako užívateľ s danými oprávneniami.



Obr. 12.21: Licenčná zmluva.



Obr. 12.22: Premenovanie importovanej aplikácie.



Obr. 12.23: Pridanie roly používateľovi.

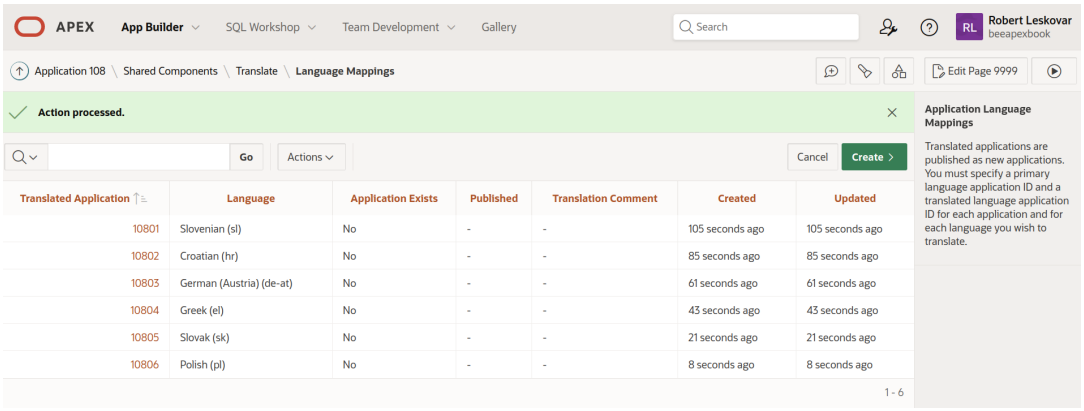
12.4 Viacjazyčná aplikácia

Táto téma bude pokrývať iba jeden aspekt viacjazyčných aplikácií – reťazce na stránkach aplikácie, ako sú reporty a formuláre. Primárne dôvody na prispôsobenie aplikácie konkrétnemu jazyku sú: a) používatelia hovoria plynule iba inými jazykmi a b) organizácie alebo štáty ukladajú požiadavky na podporu viac ako jedného jazyka. Rozsah prispôsobenia môže byť obmedzený na preklad štítkov položiek stránok, správ aplikácií, interných správ APEX, formátov čísel a dátumov alebo môže byť taký zložitý, ako je preklad textu uloženého v databázových tabuľkách (t. j. stav, úroveň, stupeň). Špecifický kultúrny a jazykový kontext môže spôsobiť, že čistý preklad reťazcov bude vtipný, urážlivý alebo nekompetentný. Lokalizácia aplikácie je teda oveľa viac. APEX má veľa možností na aplikáciu prekladu (na základe primárneho jazyka aplikácie, prehliadača, preferencie aplikácie, položky, relácie). Nasledujúca časť poskytuje najjednoduchšie pokyny na implementáciu viacerých jazykov v aplikácii zmenou primárneho jazyka aplikácie. Postup je nasledovný:

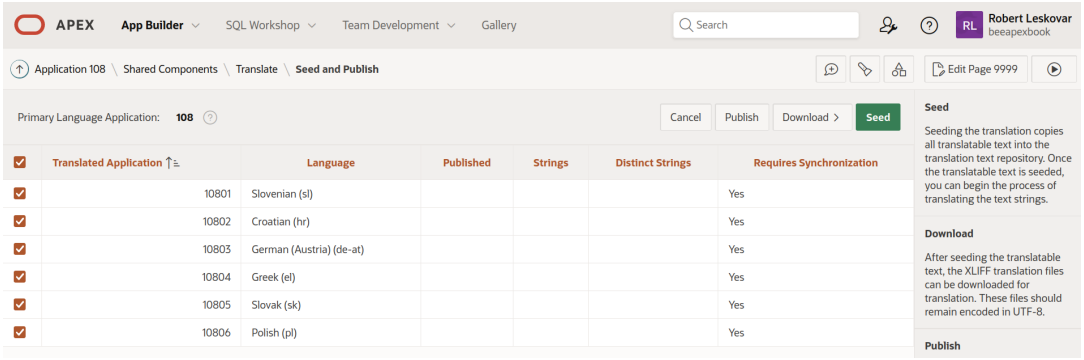
- prejdite do časti Shared components > Globalization > Application Translations > Define application languages: kliknite na tlačidlo Create; pre každý jazyk nastavte jedinečné celé číslo a hodnotu (t.j. pridajte dve číslice k číslu aplikácie, napríklad 10801 pre sl; 10802 pre hr; 10803 pre de-at; 10804 pre el; 10805 pre sk; 10806 pre pl). Na Obrázku 12.24 sú zobrazené všetky nastavené jazyky.
- prejdite na Shared components > Globalization > Application translations > Seed translatable

text. Vyberte všetky jazyky, ako je znázornené na Obrázku 12.25. Kliknite na tlačidlo Seed a počkajte, kým sa nedokončí proces.

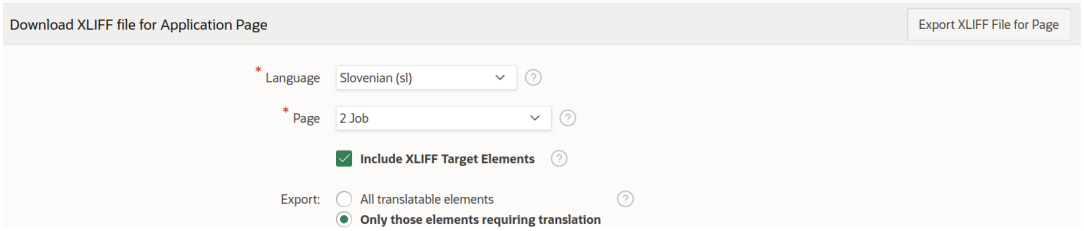
- prejdite na Shared components > Globalization > Application Translations > Download XLIFF translation files; môžeme si vybrať, či stiahneme všetky preložiteľné prvky alebo len tie prvky, ktoré si preklad vyžadujú. Príklad na Obrázku 12.26 ukazuje export strany 2, slovinského jazyka a prvkov vyžadujúcich preklad.
- upravte exportovaný súbor v preferovanom editore (Notepad++, Kate, Sublime etc.). Na demonštračné účely sme zmenili len niekoľko reťazcov označených ako target": v riadkoch 48, 52, 56, 60, 72 a 76 (Obrázok 12.27). Uložte zmeny.
- prejdite na Shared components > Globalization > Translate application > Apply XLDIFF translation files > Upload Files. Vyberte súbor (Obrázok 12.28) a kliknite na tlačidlo Upload. Vyberte správny jazyk v časti “Apply to Translation” (Obrázok 12.29). Kliknite na tlačidlo Apply changes a následne na Publish.
- prejdite na Shared components > Globalization > Translate application > Publish translations. Vyberte všetky jazyky, ktoré ste preložili, a kliknite na tlačidlo Publish (Obrázok 12.30). Počkajte, kým dostanete správu, že aplikácia bola úspešne zverejnená.
- prejdite na Shared components > Globalization > Globalization attributes. Zmeňte primárny jazyk aplikácie na nový preklad (Obrázok 12.31). Kliknite na tlačidlo Apply changes.



Obr. 12.24: Definované jazyky na preklad.



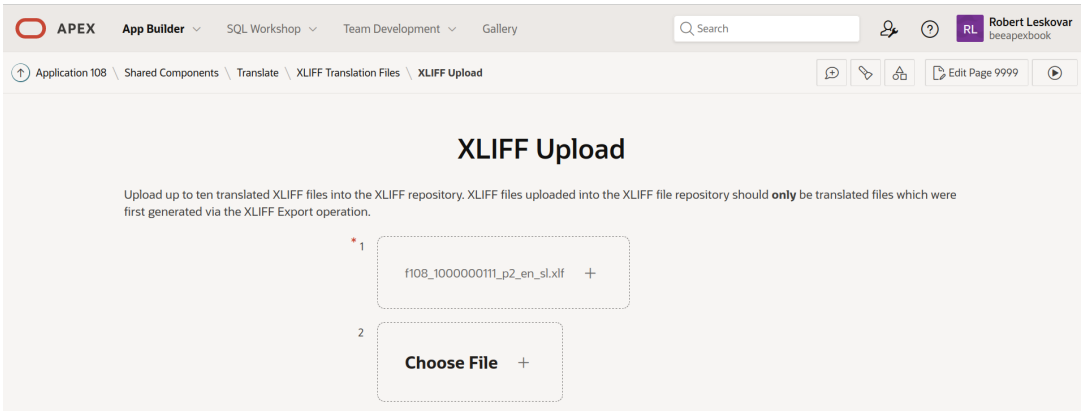
Obr. 12.25: Seed preložiteľného textu.



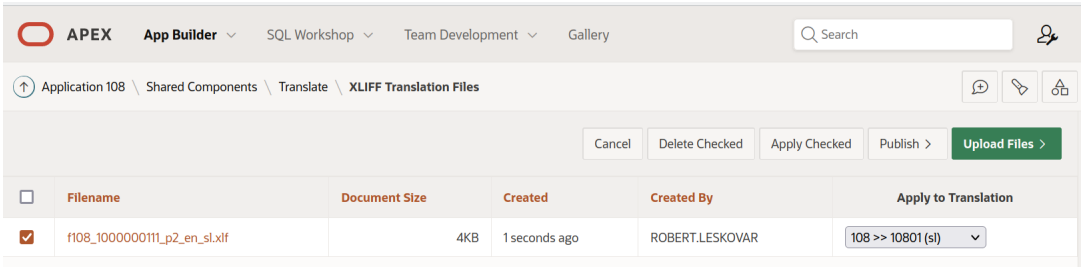
Obr. 12.26: Export reťazcov pre konkrétny jazyk a stránku.

```
45  </trans-unit>
46  <trans-unit id="S-13-23279527479873043-108">
47  <source>Create</source>
48  <target>Ustvari</target>
49  </trans-unit>
50  <trans-unit id="S-20-23278046036873035-108">
51  <source>Ch12 Job</source>
52  <target>Ch12 Del. mesto</target>
53  </trans-unit>
54  <trans-unit id="S-20-23280741017873048-108">
55  <source>Breadcrumb</source>
56  <target>Drobtina</target>
57  </trans-unit>
58  <trans-unit id="S-143-23278046036873035-108">
59  <source>Job</source>
60  <target>Del. mesto</target>
61  </trans-unit>
62  <trans-unit id="S-146-23278230062873035-108">
63  <source>The maximum row count for this report is #MAX_ROW_COUNT# rows.  Please apply a filter to reduce the num
64  <target>The maximum row count for this report is #MAX_ROW_COUNT# rows.  Please apply a filter to reduce the num
65  </trans-unit>
66  <trans-unit id="S-147-23278230062873035-108">
67  <source>No data found.</source>
68  <target>No data found.</target>
69  </trans-unit>
70  <trans-unit id="S-149-23278547600873039-108">
71  <source>ID</source>
72  <target>Šifra</target>
73  </trans-unit>
74  <trans-unit id="S-149-23278989319873042-108">
75  <source>Job Description</source>
76  <target>Opis delovnega mesta</target>
77  </trans-unit>
78  </body>
79  </file>
80  </xliff>
```

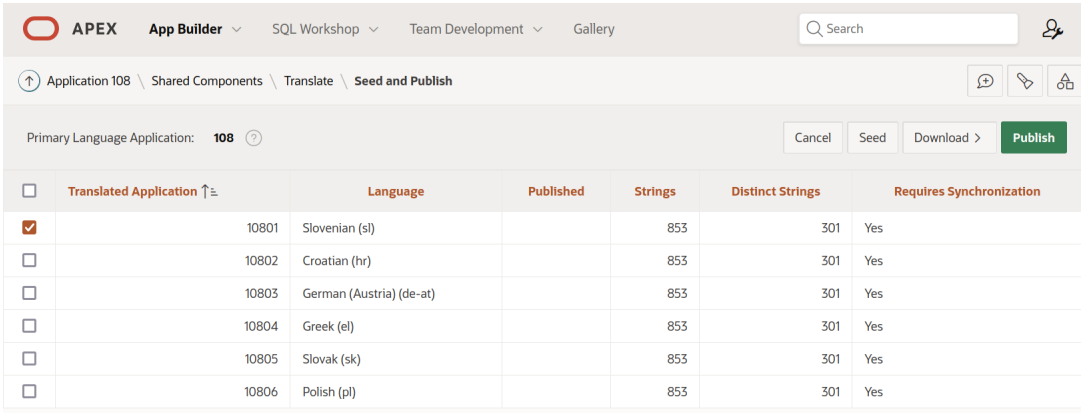
Obr. 12.27: Preklad reťazcov označených ako “target” v riadkoch 48, 52, 56, 60, 72 a 76.



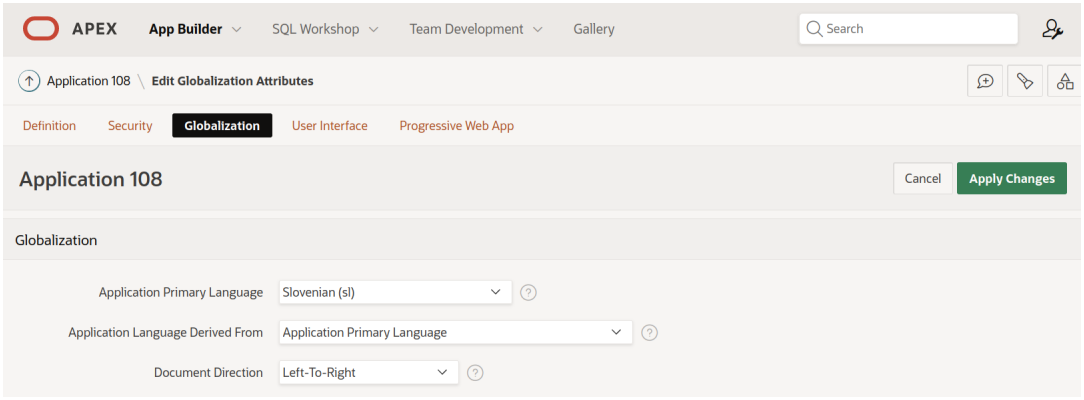
Obr. 12.28: Nahrávanie prekladových súborov XLDIFF.



Obr. 12.29: Aplikovanie zmien a publikovanie.

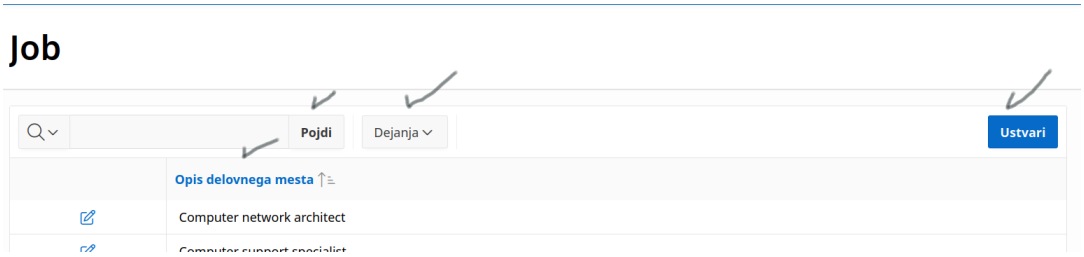


Obr. 12.30: Finálne zverejnenie prekladu aplikácie.



Obr. 12.31: Nastavenie hlavného jazyka aplikácie.

Teraz skontrolujte stranu 2 v preloženej aplikácii (Obrázok 12.32).



Obr. 12.32: Preložená stránka.

Poznámky na záver Vývoj zvyčajne prebieha v testovacom prostredí. Pri migrácii schválenej aplikácie do nového pracovného priestoru sú zabalené aplikácie veľmi pohodlné, pretože skracujú čas potrebný na migráciu. Z bezpečnostných dôvodov sa prípadní používatelia a ich roly nemigrujú. Autorizácia používateľov v novom pracovnom priestore je vážnou úlohou. V tejto kapitole sú uvedené aj základné pokyny pre viacjazyčné aplikácie. Treba poznamenať, že v prípade veľkých a kritických misií musí byť proces prekladu podporovaný schopnými automatickými prekladateľskými nástrojmi, ako aj odborníkmi na jazyk a kultúru.

12.5 Doplnkový učebný materiál

Môžete nájsť nasledujúci doplnkový učebný materiál:

- exportované aplikácie
- video návody

Všetky doplnkové učebné materiály sú dostupné na verejná stránka projektu BeeAPEX. Prihláste sa ako hosť (nevyžaduje sa žiadne heslo). Učebnicu nájdete v sekcii Books, skriptá v priečinku Part 1 > Chapter12 v sekcii Scripts a video návody v časti Collection of video guides. Materiál pre krátke kurzy v sekcii Short courses.

12.5.1 Exportované aplikácie

Sú to tri súvisiace aplikácie:

- Prvotná aplikácia. Skontrolujte, či existujú tabuľky začínajúce na CH12. Ak tabuľky existujú, môžete ich zrušiť pomocou skriptu CH12DROP. Následne importujte aplikáciu do súboru CH12_Application_initial.sql. Po naimportovaní aplikácie je potrebné definovať tabuľky. Spustite skript CH12CREATE a voliteľne naplňte tabuľky skriptom CH12INSERT. Prejdite na importovanú aplikáciu > Shared Components > Application Access Control. Pridajte User Role Assignment pre aktuálne prihláseného používateľa. Povoľte všetky akcie s rolou Administrator, pre rolu Contributor povoľte select, insert, update, delete a pre rolu Reader iba select.
- zabalená aplikácia. Skontrolujte, či existujú tabuľky začínajúce na CH12. Ak tabuľky existujú, musíte ich zrušiť pomocou skriptu CH12DROP. Následne importujte aplikáciu do súboru CH12_Application_packaged.sql. Prejdite na importovanú aplikáciu > Shared Components > Application Access Control. Pridajte User Role Assignment pre aktuálne prihláseného používateľa. Povoľte všetky akcie s rolou Administrator, pre rolu Contributor povoľte select, insert, update, delete a pre rolu Reader iba select.
- zabalená viacjazyčná aplikácia. Skontrolujte, či existujú tabuľky začínajúce na CH12. Ak tabuľky existujú, musíte ich zrušiť pomocou skriptu CH12DROP. Následne importujte aplikáciu do súboru CH12_Application_packaged.sql. Prejdte na importovanú aplikáciu > Shared Components > Application Access Control. Pridajte User Role Assignment pre aktuálne prihláseného používateľa. Povoľte všetky akcie s rolou Administrator, pre rolu Contributor povoľte select, insert, update, delete a pre rolu Reader iba select.

12.5.2 Video návody

K dispozícii sú nasledujúce videonávody:

- vytvorenie úvodnej aplikácie
- import úvodnej aplikácie
- skopírovanie úvodnej aplikácie do zabalenej a vytvorenie zabalenej aplikácie
- import zabalenej aplikácie
- import zabalenej viacjazyčnej aplikácie

12.6 Otázky

1. Čo je to zabalená aplikácia a aká je jej výhoda?
2. Ako exportujete zabalenú aplikáciu tak, aby obsahovala definíciu databázového objektu a údaje pre tabuľky?
3. Akú úlohu zohráva súbor XLDIFF pri vytváraní viacjazyčnej aplikácie?

12.7 Odpovede

1. Zabalená aplikácia je vopred zostavená aplikácia, ktorú je možné nainštalovať a nakonfigurovať v pracovnom priestore Oracle APEX. Hlavnou výhodou zabalenej aplikácie je ľahká a plynulá migrácia ako aj okamžitá dostupnosť pripravených dát.
2. Zabalenú aplikáciu je potrebné exportovať pomocou sprievodcu. Jedna môže obsahovať skripty, ktoré definujú databázové objekty (ako tabuľky) a inštruujú vkladanie údajov do tabuliek. Zoznam skriptov a poradie vykonávania týchto skriptov sa nastavuje v sprievodcovi. Zabalená aplikácia môže obsahovať aj deinštalačné skripty.
3. XLDIFF súbor je označený a preto poskytuje jednoduchý prístup k cieľovým hodnotám pre prekladové reťazce. Tieto súbory môžu obsahovať celú aplikáciu alebo len konkrétnu stránku vo vybranom jazyku. Súbor s preloženými cieľovými reťazcami sa importuje do aplikácie. Úložisko prekladov musí byť zverejnené, aby bolo dostupné pre koncového používateľa.

II Vytváranie aplikácie v APEX

13 Intranetové novinky pre zamestnancov 227
R. Leskovar, U. Rajkovič, A.Baggia . . . . . .

14 GreenDi - Katalóg rastlín 237
Vjeran Strahonja, Dijana Oreski, Darko Androcec, Ana Kutnjak . . . . . .

15 GreenDi - Autorizácia a správa používateľov 247
Vjeran Strahonja, Darko Androcec, Ana Kutnjak, Larisa Hrustek . . . . . .

16 Malý inovačný systém 253
R. Leskovar, U. Rajkovič, A.Baggia . . . . . .

17 Manažment podnikových procesov . 263
R. Leskovar, U. Rajkovič, A.Baggia . . . . . .

18 GreenDi – Výmena rastlín a semien . . 293
Vjeran Strahonja . . . . . .

19 Systém správy knižných recenzií 299
A. Kutnjak, L. Hrustek, A. Baggia and R.Leskovar . . . .

20 Kusovník a kalkulácia nákladov 311
R. Leskovar, U. Rajkovič, A.Baggia . . . . . .

21 Manažment výživy a stravovania 327
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . .

22 Plánovanie konzultačných hodín 347
J. Mańko, M. Sońta, R. Leskovar . . . . . .

23 Prípad Telco 367
Veronika Šalgová, Jozef Kostolný, Michal Mrena, Michal Kvet, Miroslav Potočár . . . . . .

24 Prípad prenájmu automobilov 389
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . .



13. Intranetové novinky pre zamestnancov

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA

13.1 Biznis pohľad prípadu

Stredne veľká spoločnosť so sedemsto zamestnancami je etablovaným hráčom v globálnom priemysle špičkového športového a rekreačného vybavenia a pokročilých kompozitov. Centrála je v Slovinsku a dcérske spoločnosti sú v USA, Kanade, Nemecku a Japonsku. Ich ročné tržby sa pohybujú okolo 90 miliónov eur. Väčšina zamestnancov sa nachádza v Slovinsku na jednom mieste. Spoločnosť má štyri divízie a každá má niekoľko oddelení. V minulosti využívali interný polročný bulletin na komunikáciu rôznych úspechov, úspešných príbehov, ohlásenia nových produktov a podobne. Potom vytvorili statickú (čisto HTML) intranetovú stránku na šírenie dôležitých správ. Táto príležitosť posilniť postavenie zamestnancov a posilniť ich záväzok voči spoločnosti však trvala krátko. Idylická záhrada sa zmenila na problém.

13.2 Definícia problému

Proces publikovania bol voľne definovaný. Požiadavku na zverejnenie noviniek dostal pracovník IT od vydavateľov - manažérov oddelení a divízií. Požiadavka bola komunikovaná poštou, telefonicky alebo osobne voľne často s nedostatkovým materiálom (text, obrázky, video). Dátum začiatku a konca spravodajstva nebol stanovený. Frekvencia žiadostí o zverejňovanie noviniek sa zvýšila na približne sto správ mesačne a niektoré požiadavky sa prekrývali (dvaja a viacerí manažéri oddelení mali podobné alebo rovnaké témy). IT administrátor musel riešiť problémy medzi manažérmi a s manažérmi. Dátumy začatia a ukončenia vystúpenia noviniek boli tak často ponechané na pleciach administrátorov IT, čo spôsobovalo zlú náladu manažérom a zamestnancom IT. Taktiež správa statickej stránky sa stala časovo náročnou a extrémne obmedzovanou chybou. Riešenie si vyžiadalo úpravu procesov a novú publikačnú platformu.

Úprava procesov a nová publikačná platforma Na vývoj novej publikačnej platformy bolo vybrané prostredie na vývoj aplikácií s nízkym kódom. Spoločnosť už používa databázu Oracle, preto je ako vývojové prostredie vybrané Oracle APEX. Identifikujú sa tri typy používateľov platformy. Prvým je administrátor, ktorý má prístup ku všetkým údajom a previlégiám na správu používateľov a rolí. Okrem základných údajov o zamestnancoch sledujeme aj ich zamestnania na oddeleniach a ich umiestnenie. Každý zamestnanec má definovanú špecifickú rolu, ktorá sa môže časom meniť. Druhým typom používateľa je vydavateľ (publisher), ktorý nemá právo spravovať údaje o zamestnancoch a definovať privilégiá, ale má právo publikovať novinky. Tretím typom

používateľa je divák/čitateľ (viewer/reader) správ, ktorý má privilégium čítať internetové správy.

Špecifikácia požiadaviek Na základe upraveného procesu platforma používateľom umožní:

- Reader/Viewer: prihlásenie, prístup na dashboard, zobrazenie noviniek a príloh k novinkám
- Publisher: prihlásenie, prístup na dashboard, zobrazenie a publikovanie noviniek a pridávanie príloh k publikovaným novinkám
- Administrator: prihlásenie, prístup na dashboard, zobrazenie a publikovanie noviniek, zobrazenie a spravovanie oddelení, rolí, zamestnancov, rolí zamestnancov, spravovanie aplikácie

13.3 Prípady použitia

13.3.1 Naratívny popis

Publikačná platforma umožňuje prístup k trom rôznym typom zamestnancov: administrátor, divák/čitateľ (viewer/reader) a prispievateľ/vydavateľ (contributor/publisher), pričom každý má na intranetovom portáli iné privilégiá. Vydavateľ môže publikovať novinky, čitateľ môže čítať novinky a administrátor môže spravovať platformu. Každý z týchto prípadov použitia vyžaduje, aby sa používateľ prihlásil na intranetovú platformu.

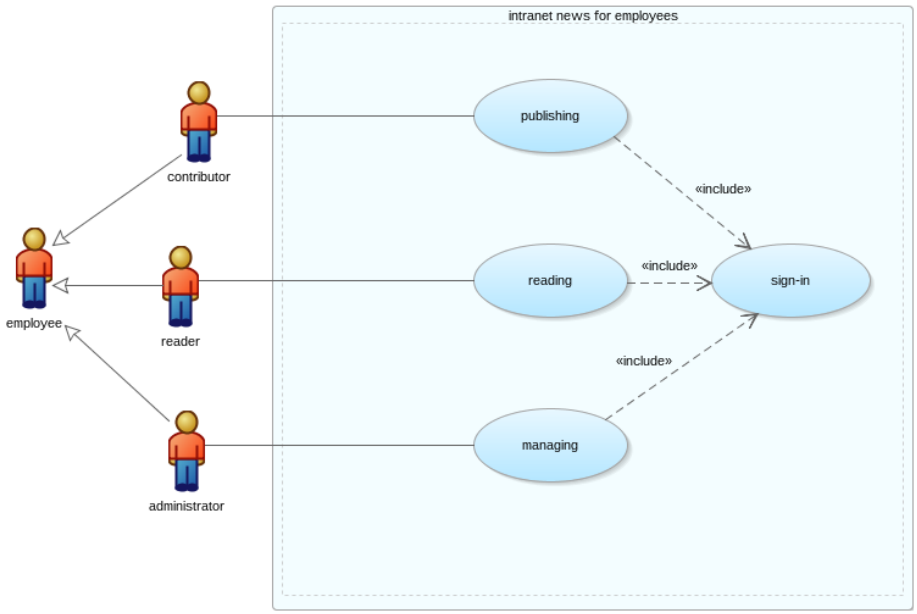
13.3.2 Pološtruktúrovaný popis

Môžeme zhrnúť tri odlišné používateľské príbehy alebo prípady použitia: publikovanie, čítanie a spravovanie.

Pridanie prílohy k novinkám je v skutočnosti rozšírením vyššie opísaného prípadu použitia a mohli by sme ho označiť ako nový prípad použitia. Kvôli prehľadnosti pokračujeme ďalšími dvoma hlavnými prípadmi použitia, avšak aplikácia bude mať funkciu na pridávanie príloh k novinkám.

13.3.3 Diagram prípadov použitia

Vyššie uvedený príbeh je znázornený na diagrame prípadu použitia, viď Obrázok 13.1.



Obr. 13.1: Diagram prípadov použitia.

Opäť sa kvôli jednoduchosti vynecháva rozšírenie prípadu použitia (pridanie prílohy k novin-

Tabuľka 13.1: Popis prípadu použitia: publikovanie internetových správ.

Keyword	**Value**
ID:	*Ch13-01*
Nadpis:	*Publikovanie noviniek na intranetovom portáli*
Popis:	*Zodpovedná osoba z HR oddelenia s definovanou rolou vydavateľa využíva intranetový portál APEX na zverejňovanie noviniek. Každá novinka má dátum začatia a ukončenia. Predvolený počiatočný dátum je deň zverejnenia.*
Hlavný aktér:	*Zamestnanec s rolou vydavateľa*
Predpoklady:	*Zamestnanec musí byť uvedený v tabuľke zamestnancov a používateľ musí byť pridaný do roly CH13 Publisher. Musí byť povolený prístup k webovej aplikácii.*
Podmienky po skončení:	*Po úspešnom zverejnení noviniek sú novinky a ich príloha k dispozícii ostatným užívateľom intranetového portálu.*
Hlavné	*scenáre*
Úspešný scenár:	1. Otvorte webový prehliadač a prihláste sa do intranetovej aplikáci 2. Vyberte možnosť Publish news 3. Zadajte názov, popis a dátum ukončenia správ 4. Potvrďte vytvorenie správ 5. Pridajte prílohu/y 6. Zadajte podrobnosti prílohy 7. Potvrďte pridanie prílohy 8. Prezrite si zverejnené novinky
Rozšírenia:	• *1a. Prihlásenie zlyhá* • *1a* Rozšírenie:* • *1a1. Zobrazenie chybového hlásenia* • *1a2. Otvorte prihlasovacie okno* • *4a. Chyba dátového typu* • *4a* Rozšírenie:* • *4a1. Zobrazenie chybového hlásenia* • *7a. Chyba dátového typu* • *7a* Rozšírenie:* • *7a1. Zobrazenie chybového hlásenia*
Frekvencia používania:	*Vydavatelia publikujú približne 1000 noviniek ročne, priemerne 5 denne*
Status:	*Dokončené*
Vlastník:	*Zamestnanec s rolou vydavateľa*
Priorita:	*stredná*

Tabuľka 13.2: Popis prípadu použitia: čítanie intranetových správ

Kľúčové slovo	**Hodnota**
ID:	*Ch13-02*
Názov:	*Čítanie správ na intranetovom portáli*
Popis:	*Používateľ intranetu číta novinky*
Hlavný aktér:	*Používateľ intranetu*
Predpoklady:	*Zamestnanec musí byť uvedený v tabuľke zamestnancov a používateľ musí byť pridaný do roly CH13 Viewer. Musí byť povolený prístup k webovej aplikácii.*
Podmienky po skončení:	-
Hlavné	*scenáre*
Úspešný scenár:	1. Otvorte webový prehliadač a prihláste sa do intranetovej aplikácie 2. Vyberte možnosť Publish news 3. Vyberte možnosť Download pre zobrazenie prílohy
Rozšírenia:	• *1a. Prihlásenie zlyhá* • *1a* Rozšírenie:* • *1a1. Zobrazenie chybového hlásenia*
Frekvencia používania:	*Zamestnanci čítajú správy na dennej báze.*
Status:	*Dokončené*
Vlastník:	*Intranetový používateľ*
Priorita:	*nízka*

Tabuľka 13.3: Popis prípadu použitia: Spravovanie intranetového portálu.

Kľúčové slovo	**Hodnota**
ID:	*Ch13-03*
Nadpis:	*Spravovanie intranetového portálu*
Popis:	*Spravovanie intranetového portálu*
Hlavný aktér:	*Administrátor intranetového portálu*
Predpoklady:	*Zamestnanec musí byť uvedený v tabuľke zamestnancov a používateľ musí byť pridaný do roly administrátora. Prístup do webovej aplikácie musí byť povolený.*
Podmienky po skončení:	*Používatelia, role a zamestnanci sú pripravení na použitie v intranetovom portáli.*
Hlavné	*scenáre*
Úspešný scenár:	1. Otvorte webový prehliadač a prihláste sa do intranetovej aplikácie 2. Pridajte a spravujte údaje oddelení 3. Pridajte a spravujte údaje rolí 4. Pridajte a spravujte zamestnancov a ich role 5. Publikujte novinky 6. Zobrazte novinky 7. Administrácia
Rozšírenia:	• *1a. Prihlásenie zlyhá* • *1a* Rozšírenie:* • *1a1. Zobrazenie chybového hlásenia* • *1a2. Otvorenie prihlasovacieho okna* • *2a. Chyba dátového typu* • *2a* Rozšírenie:* • *2a1. Zobrazenie chybového hlásenia* • *3a. Chyba dátového typu* • *3a* Rozšírenie:* • *3a1. Zobrazenie chybového hlásenia* • *4a. Chyba dátového typu* • *4a* Rozšírenie:* • *4a1. Zobrazenie chybového hlásenia* • *5a. Chyba dátového typu* • *5a* Rozšírenie:* • *5a1. Zobrazenie chybového hlásenia*
Frekvencia používania:	*Frekvencia používania závisí od nových zamestnancov a zmien v schéme privilégií. Približne 5-krát týždenne.*
Status:	*Dokončené*
Vlastník:	*Administrátor intranetového portálu*
Priorita:	*vysoká*

kám) na Obrázku 13.1.

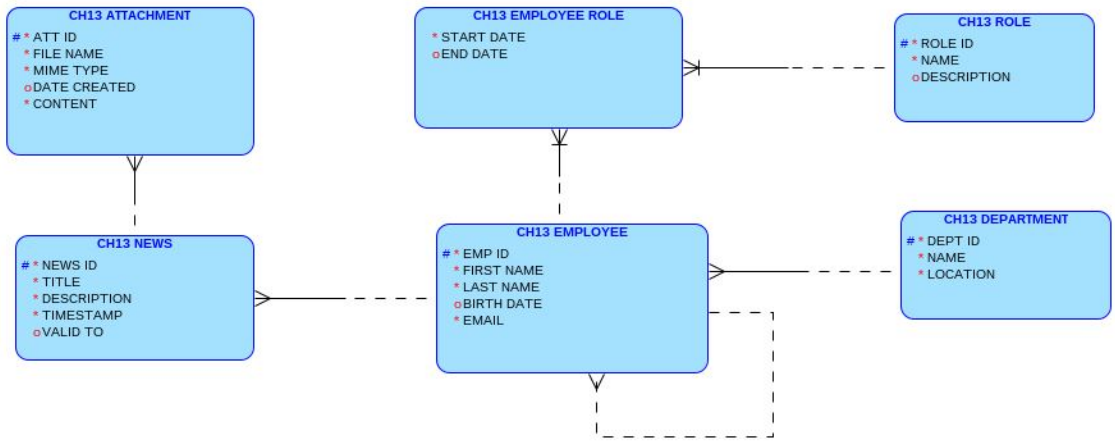
13.4 Dátový model

13.4.1 Naratívny popis dátového modelu

Zamestnanec môže byť manažér. Každý zamestnanec má iba jedného manažéra, zatiaľ čo manažér môže riadiť niekoľko zamestnancov. Každý zamestnanec môže mať svoju rolu. Roly sú definované pomocou ID, názvu a popisu. Pre každého zamestnanca a každú rolu je definovaný dátum začiatku a konca. V priebehu času môže mať zamestnanec viacero rolí, k jednej role môže byť priradených aj viac zamestnancov. Jedna definícia roly zamestnanca sa však nemôže preniesť na iného zamestnanca alebo rolu. Zamestnanec môže v priebehu času publikovať viac noviniek. Každá novinka je identifikovaná podľa ID, názvu, popisu, dátumu začiatku a dátumu ukončenia. Každá správa môže mať viac príloh. Príloha je identifikovaná pomocou ID, názvu súboru, typu MIME, dátumu vytvorenia a obsahu.V logickom dátovom modeli je šesť entít. Entita CH13 DEPARTMENT má tri atribúty: ID, názov a umiestnenie oddelenia. Každé oddelenie môže mať veľa zamestnancov. V entite CH13 EMPLOYEE sú uložené základné údaje o zamestnancoch: ID, meno, priezvisko, dátum narodenia a email. Zamestnanec môže byť manažér. Každý zamestnanec má iba jedného manažéra, zatiaľ čo manažér môže riadiť niekoľko zamestnancov. Každý zamestnanec môže mať svoju rolu. Roly sú definované pomocou ID, názvu a popisu. Pre každého zamestnanca a každú rolu je definovaný dátum začiatku a konca. V priebehu času môže mať zamestnanec viacero rolí, k jednej role môže byť priradených aj viac zamestnancov. Jedna definícia roly zamestnanca sa však nemôže preniesť na iného zamestnanca alebo rolu. Zamestnanec môže v priebehu času publikovať viac správ. Každá sráva je identifikovaná podľa ID, názvu, popisu, dátumu začiatku a dátumu ukončenia. Každá správa môže mať viac príloh. Príloha je identifikovaná pomocou ID, názvu súboru, typu internetového média, dátumu vytvorenia a obsahu.

13.4.2 Logický dátový model

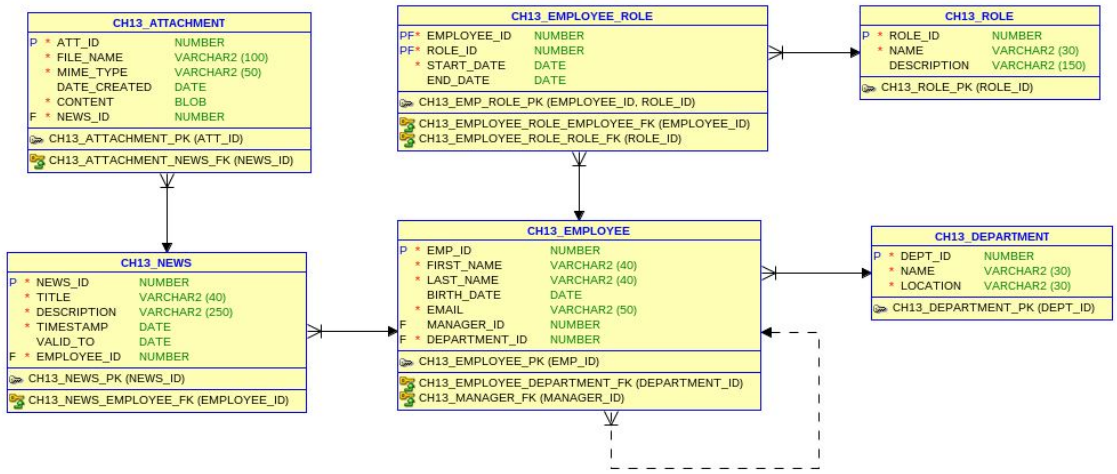
Vyššie uvedený príbeh je znázornený na logickom dátovom modeli, viď Obrázok 13.2.



Obr. 13.2: Logický dátový model.

13.4.3 Relačný dátový model

Automatickú transformáciu z logického dátového modelu na relačný dátový model v Oracle SQL Data Modeler zabezpečuje funkcia *Engineering to relational*. Výsledok je zobrazený na Obrázku 13.3.



Obr. 13.3: Logický dátový model.

Oracle SQL Data Modeler tiež generuje SQL skript na vytváranie tabuliek, sekvencií a triggrov. Vyberte všetky tabuľky relačného modelu a použite funkciu *File* > *Export* > *DDL File* aby ste získali skript, ako je tento:

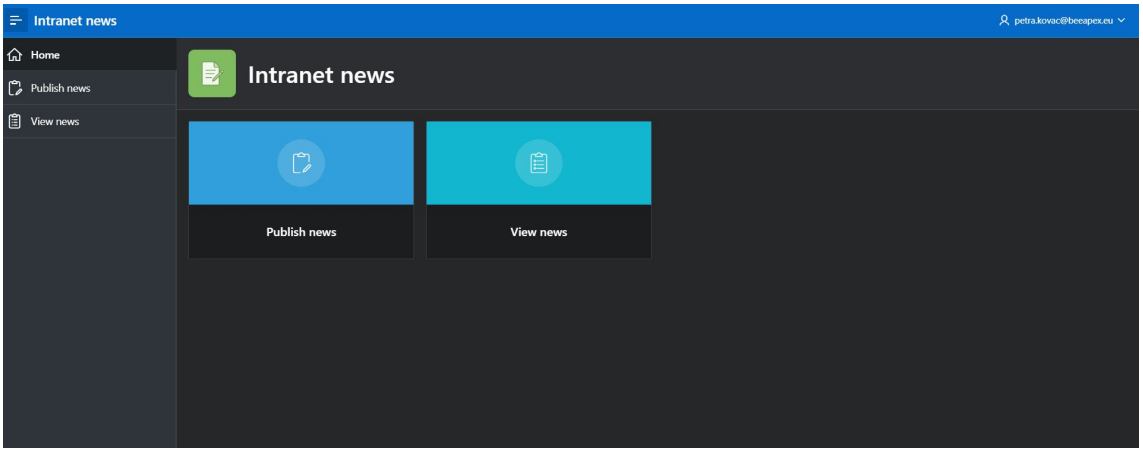
```
CREATE TABLE CH13_NEWS (
NEWS_ID NUMBER NOT NULL
...
);

CREATE TABLE CH13_DEPARTMENT (
DEPT_ID NUMBER NOT NULL
...
);
```

Teraz je čas vygenerovať tabuľky v databáze Oracle. Vygenerovaný skript môžeme importovať a spustiť v APEXe.

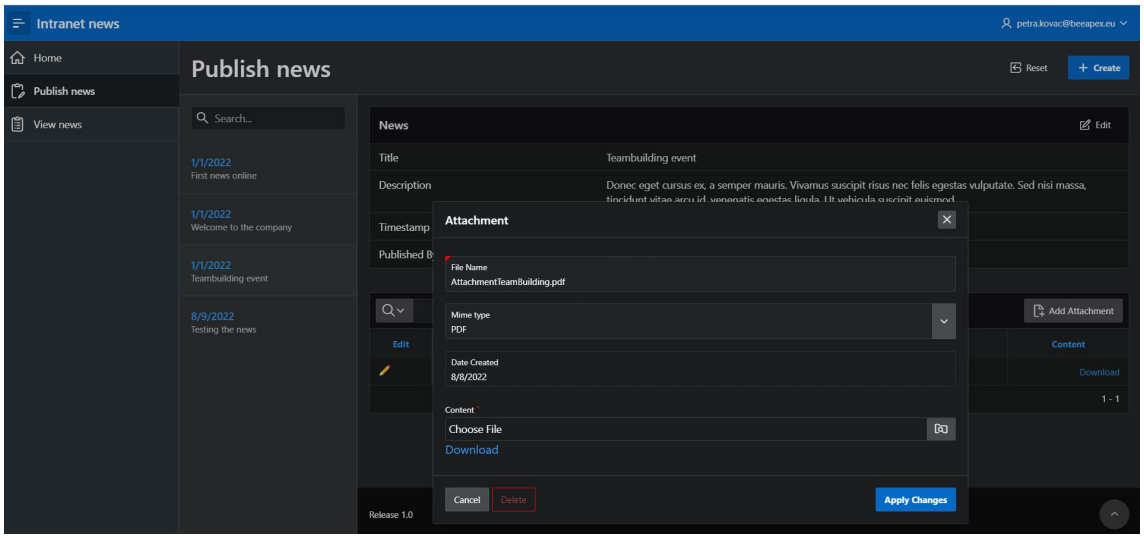
13.5 Rozhrania aplikácie

Na nasledujúcich obrázkoch sú zobrazené rozhrania aplikácie pre všetky tri roly. Najprv sa zobrazí hlavný panel pre prispievateľa (Contributor/Publisher) na Obrázku 13.4.



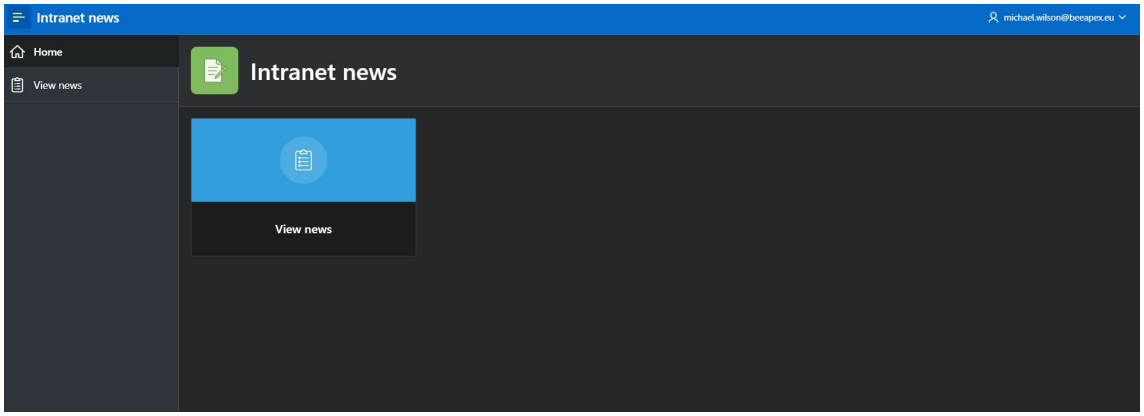
Obr. 13.4: Hlavný panel prispievateľa/vydavateľa.

Stránka na publikovanie noviniek s prílohami a ich podrobnosťami je uvedená na Obrázku 13.5.



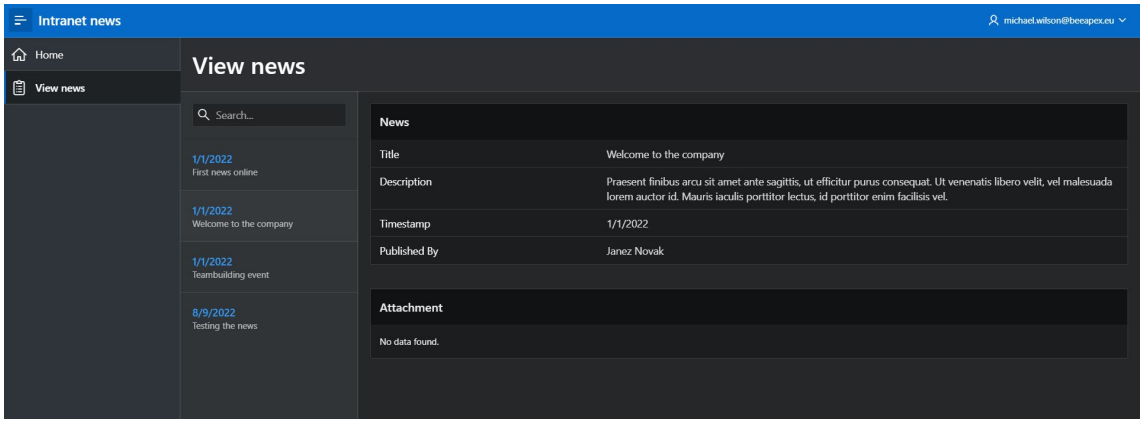
Obr. 13.5: Publikovanie noviniek s prílohami.

Rola čitateľa (Viewer/Reader) má najviac obmedzený informačný panel. Zatiaľ sú k dispozícii iba novinky Domov a Zobraziť, ako je uvedené na Obrázku 13.6.



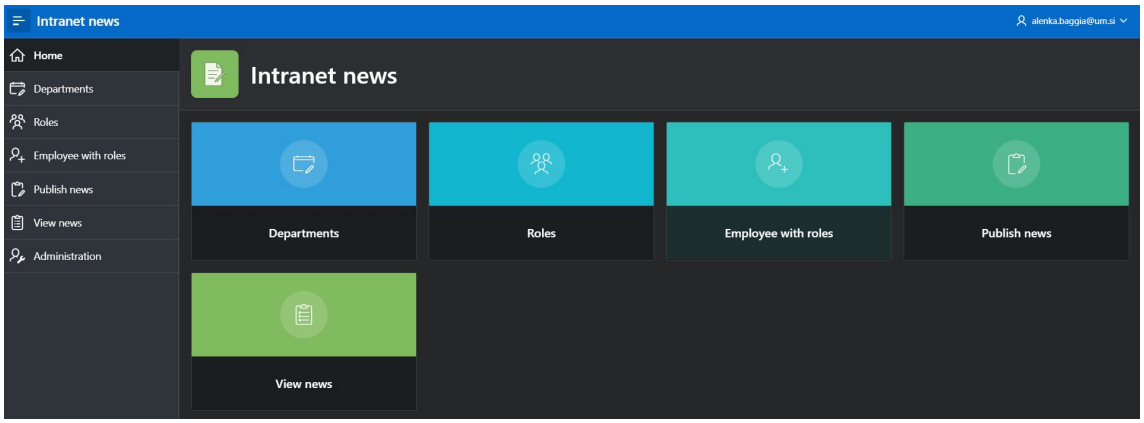
Obr. 13.6: Hlavný panel čitateľa.

Hlavnou funkciou čitateľa (Viewer/Reader) je prístup k novinkám (Obrázok 13.7).



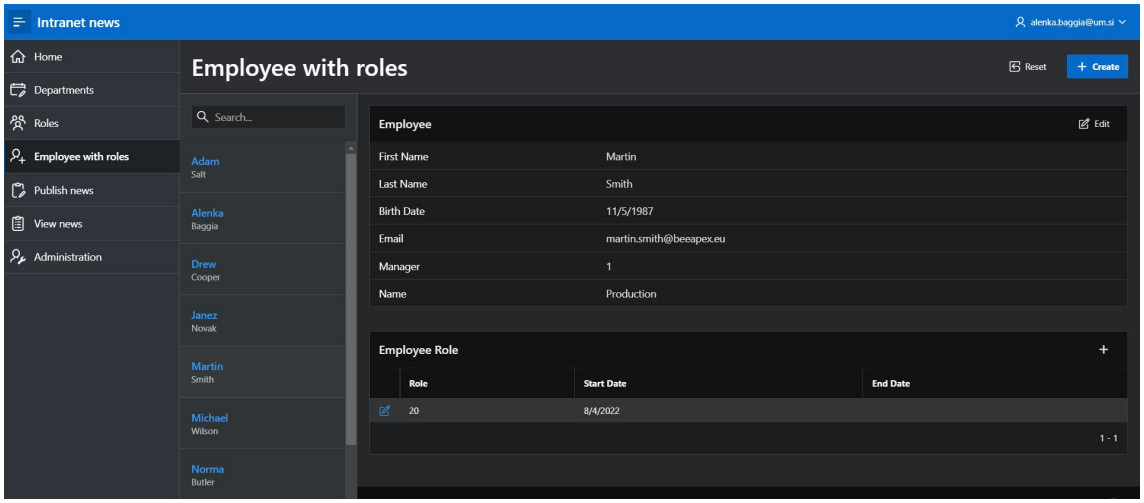
Obr. 13.7: Prístup čitateľa k novinkám.

Okrem prezerania a zverejňovania noviniek môže administrátor spravovať oddelenia, roly a zamestnancov s údajmi o rolách, ako je znázornené na Obrázku 13.8. K dispozícii je aj administračná stránka, kde je možná správa používateľov.



Obr. 13.8: Panel správcu.

Stránka na úpravu údajov o rolách zamestnancov sa nachádza na Obrázku 13.9. Táto stránka je prístupná iba pre administrátora intranetového portálu.



Obr. 13.9: Úpravy roly zamestnanca administrátorom.

13.6 Otázky

1. Ako by ste zmenili logický dátový model tak, aby implementoval nasledujúcu požiadavku: jedna príloha môže patriť viacerým novinkám.
2. Ako by sa zmenil relačný dátový model, ak by sa implementovala požiadavka (jedna príloha môže patriť viacerým novinkám)?
3. Ktoré používateľské roly sú implementované v aplikácii v Kapitole 13 a prečo?

13.7 Odpovede

1. Vzťah jeden k mnohým by mal byť spojený v smere od prílohy k správam.
2. Dvojica príloha-novinky v logickom modeli by sa transformovala do troch relačných tabuliek: príloha, novinky a správy príloh_. Posledná by mala obsahovať minimálne dve polia (ID prílohy a novinky), obe slúžiace ako cudzie kľúče.
3. V Kapitole 13 sú implementované nasledujúce užívateľské roly: Administrátor, Prispievateľ (Contributor) a Čitateľ (Reader). Tieto roly sú implementované, aby umožnili oprávneným používateľom vykonávať svoju prácu a zabránili neoprávneným používateľom vykonávať neschválené akcie.



14. GreenDi - Katalóg rastlín

VJERAN STRAHONJA, DIJANA ORESKI, DARKO ANDROCEC, ANA KUTNJAK

14.1 Biznis pohľad prípadu

Skupina nadšencov zhromaždená v neziskovej organizácii GreenDi rozbieha a realizuje rôzne iniciatívy súvisiace s mestským záhradkárčením, zdravým stravovaním, pestovaním starých odrôd rastlín, spoluprácou s miestnymi farmármi a pod. Doteraz pri svojej práci využívali sociálne siete. Rozhodli sa spustiť vlastnú GreenDi platformu na zdieľanie semien a rastlín medzi používateľmi, s cieľom podporovať biodiverzitu a uľahčiť aktivity, ktorými sa doteraz zaoberali, i umožniť nové aktivity. Aby sme mohli takúto platformu vyvinúť a implementovať, je potrebné špecifikovať požiadavky a vytvoriť základné modely, vrátane dátového modelu a prípadov použitia.

14.2 Definícia problému

Otvorená časť platformy má nasledujúce charakteristiky:

- Používatelia GreenDi nie sú obmedzení len na geografickú polohu, môžu fungovať aj globálne. Keďže však ide o výmenu rastlín, očakáva sa vytvorenie miestnych sietí súvisiacich s určitou geografickou oblasťou.
- Platforma je rozdelená do niekoľkých tematických jednotiek, ako sú tradičné a domorodé odrody, semená (zelenina, ovocie, liečivé byliny, izbové a okrasné rastliny), s myšlienkou rozšírenia na priemyselné a iné rastliny.
- Základom otvorenej časti je katalóg rastlín obsahujúci názov (alebo názvy) v jazyku používateľa, v latinčine a angličtine, klasifikáciu rastliny (taxonómiu), popis, biotopy, pestovanie a použitie, fotografie a ďalšie základné informácie.
- Základné informácie o rastline sú dostupné pre vyhľadávanie a prezeranie bez akejkoľvek registrácie.
- GreenDi katalóg rastlín je prepojený s niekoľkými otvorenými webovými stránkami, databázami a externými službami, ktoré obsahujú informácie týkajúce sa rastlín. Používajú sa URL adresy, odkazy a iné typy označených údajov.
- Katalóg obsahuje rastliny, ktoré sa vymieňajú na GreenDi platforme, ale aj tie, ktoré ešte neboli vymieňané alebo zaradené do ponuky. V katalógu sa tiež nachádzajú odkazy na komerčných dodávateľov, s ktorými majú členovia dobré skúsenosti a ktorí sa venujú ekologickej pestovateľskej činnosti.

Správa katalógu (Catalog Management) je popísaná nasledujúcimi vetami:

- Katalóg rastlín je spravovaný administrátormi GreenDi platformy.
- Správa katalógu zahŕňa zadávanie a zmenu informácií o rastlinách.
- Niektoré funkcionality správy katalógu sú dostupné pre používateľov s autorizáciou od administrátora (autorizovaný používateľ).
- Pokročilé funkcionality sú dostupné pre administrátorov, zahŕňajú prepojenia na ďalšie stránky, ktoré obsahujú informácie o rastline, videá o pestovaní, využití rastlín atď.
- Všetci registrovaní používatelia (členovia) môžu, ak si to želajú, zdieľať svoj názor na rastlinu, napísať recenziu, pridať ďalšie informácie atď. (ale to nie je súčasťou tejto funkcionality).

14.3 Prípady použitia

14.3.1 Naratívny popis

Otvorená časť platformy GreenDi umožňuje každému používateľovi z akéhokoľvek miesta prezerať si údaje z katalógu rastlín bez registrácie. Rastliny sú zároveň zaradené do kategórií, ktoré používateľovi umožňujú jednoduchšiu navigáciu. Pri každej rastline sa uchováva názov, taxonómia, popis, biotopy, pestovanie a využitie, fotografie a ďalšie základné informácie, ako aj odkazy na externé zdroje údajov.

14.3.2 Pološtruktúrovaný popis

Tabuľka 14.1 predstavuje prípady použitia.

14.3.3 Diagram prípadov použitia

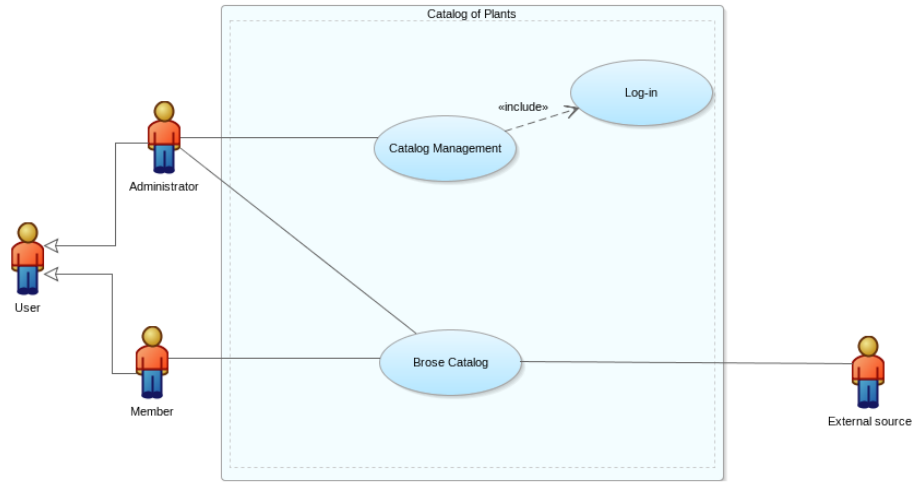
Tento príbeh je znázornený na diagrame prípadov použitia 14.1. Na pravej strane diagramu je aktér externý zdroj (External Source). Tento aktér nie je reprezentovaný stereotypom človeka, pretože ide o externý systém, z ktorého GreenDi katalóg rastlín získava informácie o rastlinách, semenách a podobne. Existuje mnoho príkladov takýchto databáz rastlín, ktoré ponúkajú otvorené služby alebo otvorené datasety, čo znamená, že databáza alebo údaje sú voľne dostupné na používanie, bez akýchkoľvek obmedzení prístupu alebo používania. Niektoré príklady online databáz rastlín, ktoré poskytujú otvorené služby alebo otvorené dáta, zahŕňajú:

1. Globálny informačný nástroj o biodiverzite (Global Biodiversity Information Facility - GBIF, https://www.gbif.org) poskytuje bezplatný a otvorený prístup k údajom o biodiverzite vrátane údajov o rastlinách. GBIF umožňuje používateľom vyhľadávať a pristupovať k údajom z rôznych zdrojov vrátane herbárií, múzeí a výskumných inštitúcií. Odkazuje na medzinárodný index názvov rastlín (International Plant Names Index - IPNI), databázu názvov a súvisiacich základných bibliografických údajov semien, rastlín, papradí a lykofytov (https://www.gbif.org/dataset/046bbc50-cae2-47ff-aa43-729fbf53f7c5).
2. Zoznam rastlín WFO (WFO Plant List, https://wfoplantlist.org/plant-list) bol spustený v máji 2021 a poskytuje používateľsky prívetivý, citovateľný statický zoznam všetkých druhov rastlín. Celý dataset WFO Plant List je tiež možné stiahnuť ako celok. K údajom je možné pristupovať aj cez rozhranie WFO Plant List API.
3. Dataset Flora of Italy je možné stiahnuť z webovej stránky University of Roma, obsahuje obrázky vo vysokej kvalite pre viac ako 10 000 druhov rastlín v Taliansku.

Toto je len niekoľko príkladov dostupných datasetov, ktoré zahŕňajú obrázky rastlín a semien v Európe, ale existuje ešte mnoho ďalších dostupných datasetov, ktoré poskytujú podobné informácie. V tomto príklade nebudeme realizovať komunikáciu s externým zdrojom, pretože to vyžaduje pokročilejšie znalosti.

Tabuľka 14.1: Popis prípadu použitia: prehliadanie katalógu rastlín

Kľúčové slovo	**Hodnota**
ID:	*ch14-01*
Názov:	*Prehliadanie katalógu miestnych rastlín*
Popis:	*Katalóg miestnych rastlín je verejne prístupný pre všetkých záujemcov. Aplikácia umožňuje vyhľadávanie s filtrom na všetkých atribútoch v databáze.*
Hlavný aktér:	*Akákoľvek osoba*
Predpoklady:	*Základné informácie o rastline sú dostupné pre vyhľadávanie a prezeranie bez potreby akejkoľvek registrácie.*
Podmienky po skončení:	*Ak je z iného používateľského prípadu (extend) vyvolané prehliadanie katalógu rastlín (Browse catalog of plants), prenáša sa ID vybranej rastliny.*
Hlavný	*Katalóg rastlín*
Úspešný scenár:	*Prvý scenár, katalóg rastlín (Catalog of plants):* 1. *Akýkoľvek používateľ na domovskej stránke otvorí formulár pre prehliadanie katalógu rastlín (Browse catalog of plants), ktorý je vyhľadávacím formulárom založeným na princípe dotazu podľa vzoru (Query by Example - QBE).* 2. *Vyhľadávanie je možné podľa nasledujúcich údajov: a) Tematická jednotka (Thematic unit) - zoznam kódov (všetky predvolene alebo niektorý konkrétny názov, napríklad zelenina, ovocie, liečivé byliny, izbové a okrasné rastliny, atď.), b) Typ rastliny (Type of plant) - hierarchicky organizovaná klasifikácia rastliny (všetky predvolene alebo niektorý konkrétny názov), c) Názov/názvy (Name/Names) v jazyku používateľa, d) Latinský názov/názvy (Latin name/names), e) Anglický názov/názvy (English name/names).* 3. *Výsledok vyhľadávania sa zobrazí vo forme tabuľky (Tabular Form).* 4. *Tabuľkový formulár je možné triediť podľa stĺpcov.* 5. *Výberom riadku je možné otvoriť formulár s detailmi rastliny (Details of the Plant), ktorý obsahuje: typ rastliny (type of plant), tematickú jednotku (thematic unit), miestne názvy (local names), anglické názvy (English names), latinské názvy (Latin names), popis (description), biotop (habitat), pestovanie (cultivation), použitie (use), fotografie (photos), odkazy (hyperlinks) a stav (status).* *Druhý scenár, správa katalógu (Catalog managemet):* 1. *Správu katalógu môže vykonávať aktívny používateľ s oprávneniami administrátora, preto prvým krokom je procedúra prihlásenia (používateľský prípad Log In), kde sa overuje splnenie týchto podmienok.* 2. *Správa katalógu zahŕňa zmenu alebo pridanie informácií o rastline, ktorá sa už nachádza v katalógu, odstránenie tejto rastliny (v skutočnosti ju skryje, pretože rastlina nie je fyzicky odstránená), alebo pridanie novej rastliny.* 3. *Používateľ potvrdí alebo odmietne zmenu.*
Rozšírenia:	• *Formulár pre vyhľadávanie (Search Form - QBE)* • *Tabuľkový formulár (Tabular Form)* • *Formulár s detailmi rastliny (Details of the Plant Form)*
Frekvencia použitia:	*Približne maximálne 1000 vyhľadávaní za minútu.*
Status:	*Vo vývoji*
Vlastník:	*verejnosť, anonymný používateľ*
Priorita:	*vysoká*



Obr. 14.1: Diagram prípadov použitia - katalóg rastlín.

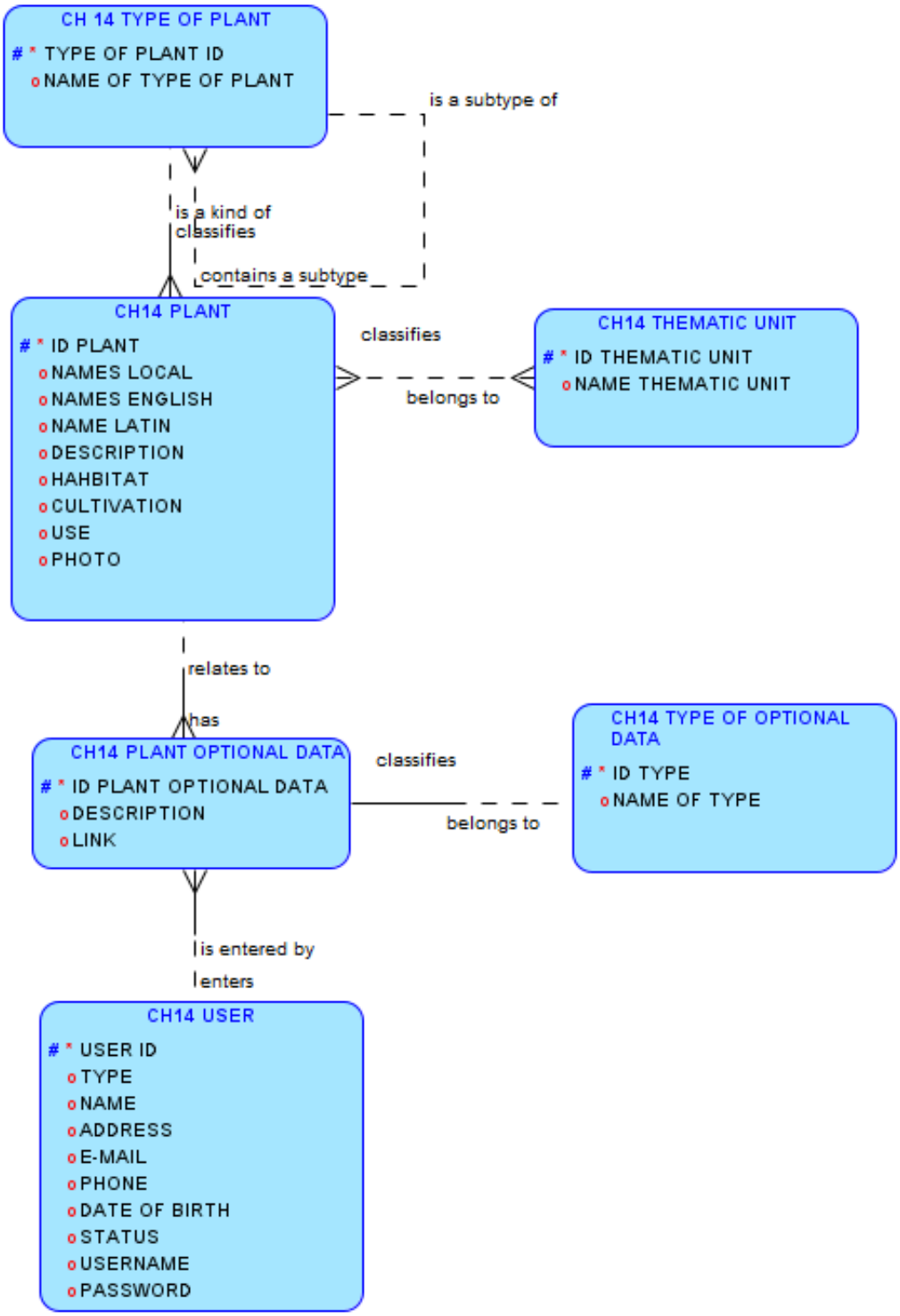
14.4 Dátový model

14.4.1 Naratívny popis dátového modelu

Logický dátový model (entitno-relačný model) obsahuje niekoľko entít. Základná entita je Plant (Rastlina), ktorá je popísaná nasledujúcimi atribútmi: ID Plant (identifikačný atribút rastliny), Names local (miestne meno rastliny), Names English (názov rastliny v angličtine), Name Latin (názov rastliny v latinčine), Description (popis rastliny), Habitat (popis prostredia, kde sa rastlina nachádza), Cultivation (spôsob pestovania rastliny), Use (účel alebo využitie rastliny), Photo (jeden alebo viacero fotografií rastliny), Hyperlinks (odkazy na externé zdroje informácií) a Status (či je rastlina aktívna a vymieňa sa alebo je archivovaná, ...). Plant je hierarchicky organizovaný kódový zoznam. To znamená, že každý typ rastliny môže obsahovať 0, 1 alebo viacero podtypov rastlín a môže byť zaradený ako podtyp do 0 alebo 1 nadradeného typu rastliny. Thematic unit (Tematická jednotka) je tiež kódový zoznam, ktorý definuje druhy rastlín a semien (zelenina, ovocie, liečivé byliny, izbové a okrasné rastliny, ...). Každá rastlina patrí do 0, 1 alebo viacerých tematických jednotiek. Každá tematická jednotka klasifikuje 0, 1 alebo viacero rastlín. Atribúty Thematic unit sú: ID Thematic unit (kód alebo identifikačný atribút tematickej jednotky) a Name (názov tematickej jednotky, môže byť viacero). Model taktiež zobrazuje entitu typu User (Používateľ). Okrem toho boli do tohto dátového modelu pridané aj niektoré ďalšie entity: Plant optional data (Voliteľné údaje o rastline), Type of optional data (Typ voliteľných údajov) a Type of plant (Typ rastliny). Plant optional data zahŕňa nasledujúce atribúty: ID Plant optional data (identifikačný atribút voliteľných údajov o rastline), Description (obsah ďalších voliteľných údajov, napr. popis odkazu, ak existuje) a Link (voliteľný odkaz). Type of optional data obsahuje atribúty: ID Type (identifikačný atribút), Name of type (názov typu, napríklad všeobecné informácie, využitie, pestovanie, osobný názor, recenzia, ...). Každá rastlina (Plant) môže mať 0, 1 alebo viacero voliteľných údajov (Plant optional data) o rastline. Každý voliteľný údaj (Plant optional data) sa vzťahuje k 1 a len 1 rastline (Plant). Každý voliteľný údaj (Plant optional data) patrí len k 1 a len 1 typu voliteľných údajov (Type of optional data). Každý typ voliteľných údajov (Type of optional data) klasifikuje 0, 1 alebo viacero voliteľných údajov o rastline (Plant optional data). Voliteľné údaje (Plant optional data) zapisuje len 1 a iba 1 používateľ (User).

14.4.2 Logický dátový model

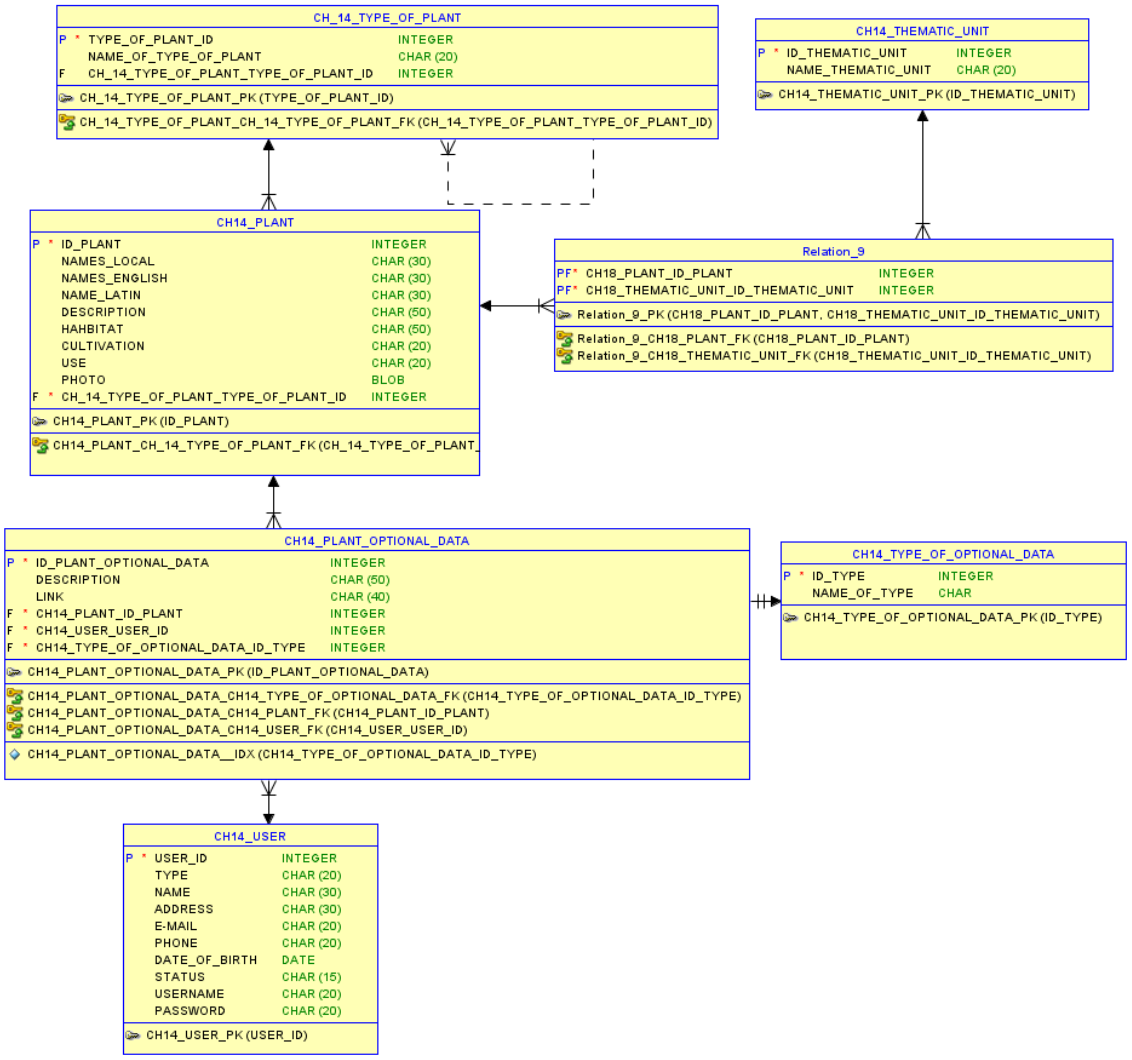
Logický dátový model je prezentovaný na Obrázku 14.2.



Obr. 14.2: Logický dátový model.

14.4.3 Relačný dátový model

Automatická transformácia z logického dátového modelu na relačný dátový model je v Oracle SQL Data Modeler poskytovaná funkciou *Engineering to relational*. Výsledok je zobrazený na Obrázku 14.3.

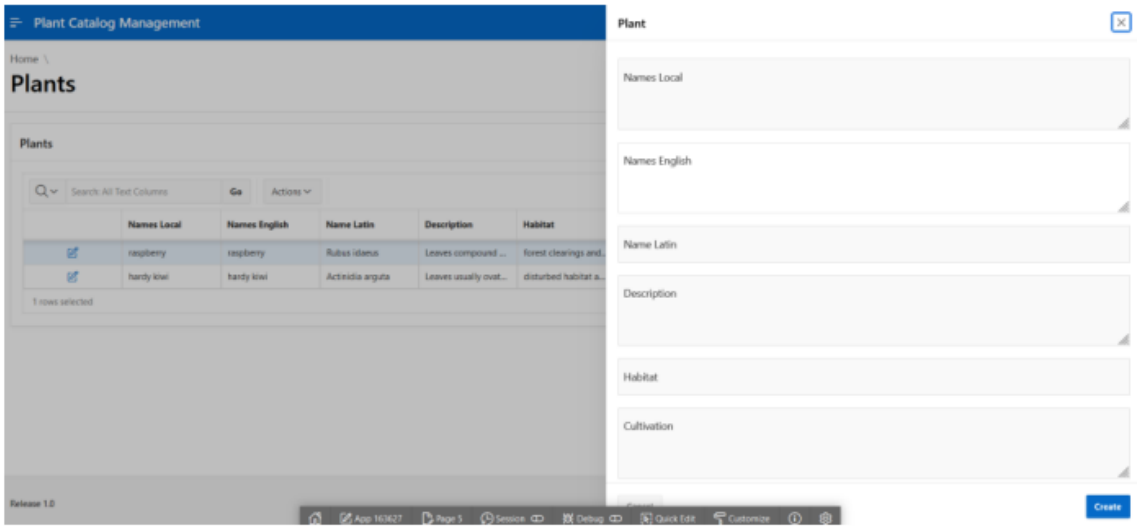


Obr. 14.3: Relačný dátový model.

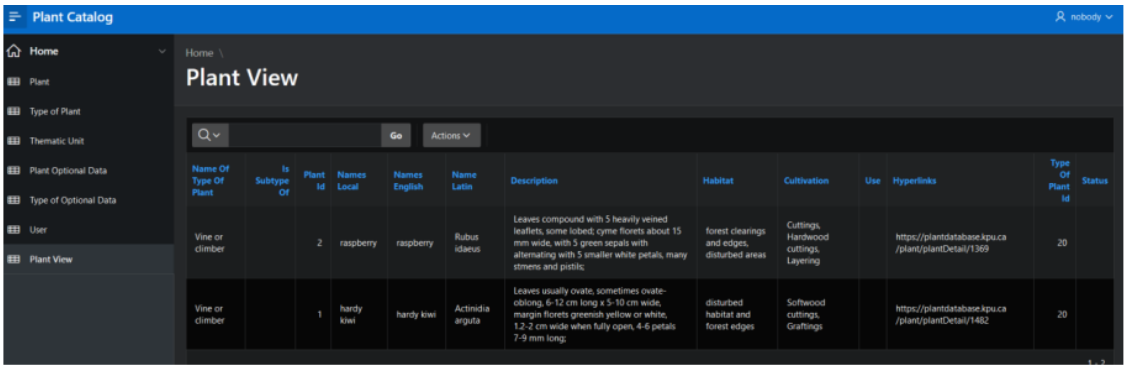
Nechajme Oracle SQL Data Modeler vygenerovať SQL skript. Vyberte všetky tabuľky v relačnom modeli a použite funkciu *File* > *Export* > *DDL File* na získanie skriptu. Vygenerovaný skript môžeme importovať do APEX-u a spustiť ho.

14.5 Rozhrania aplikácie

Podrobnosti interaktívnej mriežky pre formulár Plant sú zobrazené na Obrázku 14.4.



Obr. 14.4: Interaktívna mriežka pre formulár Plant.



Obr. 14.5: Verejná časť – otvorené prehliadanie údajov o rastlinách.

Prehliadanie katalógu miestnych rastlín je implementované pomocou vyhľadávacieho formulára.

14.6 Doplnkový učebný materiál

Môžete nájsť nasledujúci doplnkový učebný materiál:

- skript na vytvorenie a naplnenie tabuliek,
- skript na zrušenie tabuliek,
- exportovaná zabalená aplikácia,
- video, ktoré ukazuje, ako generovať aplikáciu.

Všetky doplnkové učebné materiály sú dostupné na verejnej stránke projektu BeeAPEX. Prihláste sa ako hosť (nevyžaduje sa žiadne heslo). Učebnicu nájdete v sekcii Books, skripty v priečinku Part2 > Chapter14 v sekcii Scripts, a videopríručky v Collection of video guides. Materiál pre krátke kurzy je v sekcii Short courses.

14.6.1 Exportovaná aplikácia

Exportovaná aplikácia je zabalená. Inštalácia vytvorí tabuľky a naplní ich údajmi. Odinštalovanie odstráni všetky objekty databázy, ktoré boli použité v tejto aplikácii.

Zabalená aplikácia je testovaná a bude fungovať v novom pracovnom priestore, ak sú splnené

nasledujúce požiadavky:

- používateľ APEX je pridaný pred spustením aplikácie. V pracovnom priestore pre vývoj a testovanie prejdite na Shared Components (Zdieľané komponenty) > Application Access Control (Riadenie prístupu k aplikáciám) > Add User Role Assignment (Pridať pridelenie roly používateľovi), zadajte používateľa APEX a nastavte mu roly Administrator (Administrátor), Contributer (Prispievateľ) a Reader (Čitateľ). V prípade produkcie je nevyhnutná konzultácia s kvalifikovaným personálom pred nasadením.

Ak používateľ nezíska príslušnú rolu, importovaná aplikácia spadne. Po zlyhaní aplikácie z dôvodu nesplnených požiadaviek je potrebné vymazať súbor cookie z webového prehliadača (napríklad v prehliadači Firefox: Settings (Nastavenia) > Cookies and Site Data (Cookies a údaje stránok) > Manage Data (Spravovať údaje)).

14.6.2 Video sprievodca

Video návod ukazuje všetky kroky pri vývoji aplikácie.

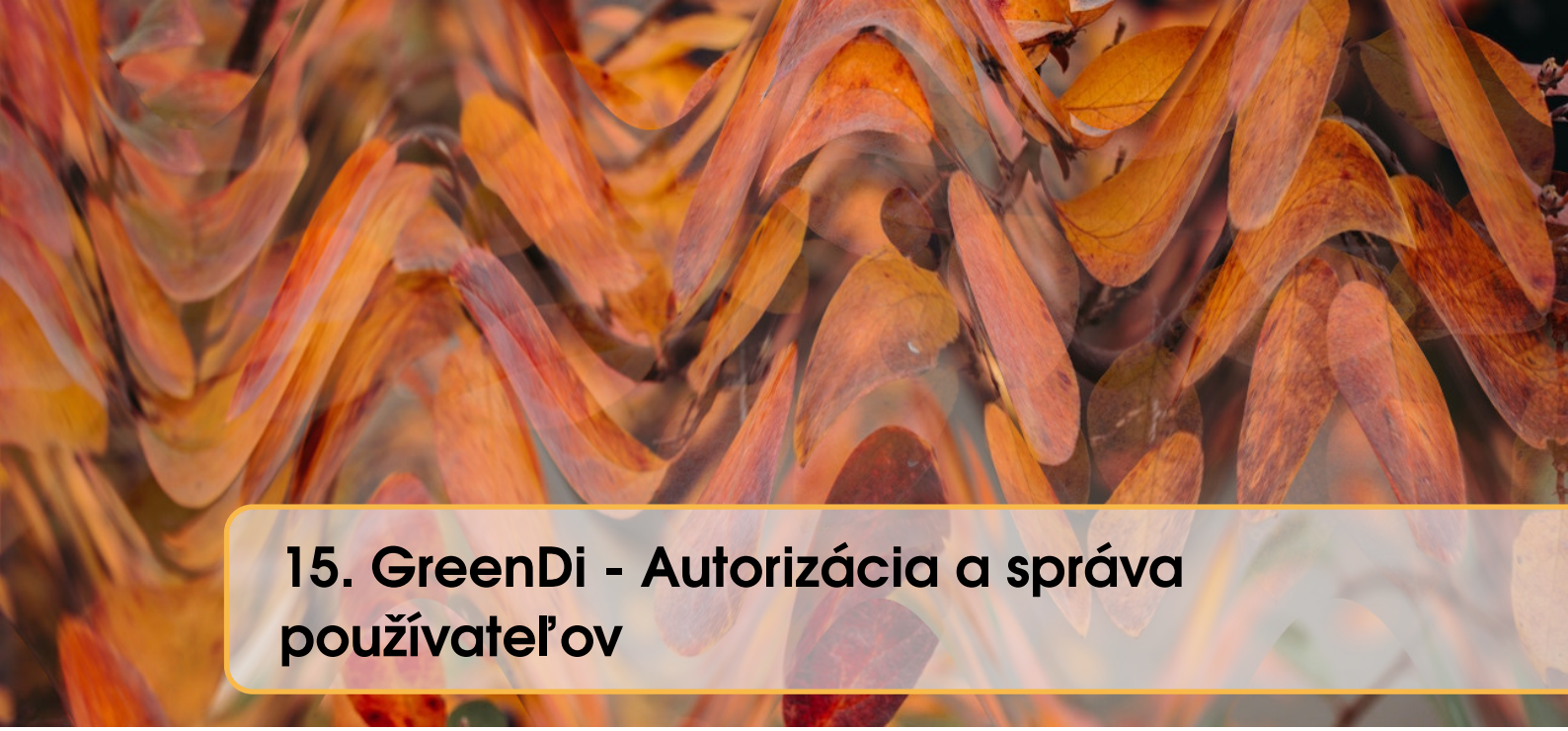
14.7 Otázky

1. Preskúmajte, ako by sa mohla implementovať komunikácia s externým zdrojom (External Source) alebo inou databázou, alebo systémom, ktorý obsahuje údaje o rastlinách.
2. Vymenujte niektoré z formátov, v ktorých sa ukladajú obrázky rastlín a semien.
3. V akom dátovom type sú obrázky, videá a podobné objekty uložené v databáze Oracle?

14.8 Odpovede

1. V princípe existujú dva spôsoby realizácie komunikácie s takýmto externým zdrojom. Jedným zo spôsobov je stiahnuť si dataset, teda kompletnú databázu do katalógu rastlín. Ďalším spôsobom je pripojenie pomocou aplikačného programového rozhrania (Application Programming Interface - API). Je to súbor protokolov, nástrojov a štandardov, ktoré definujú, ako by mali softvérové komponenty vzájomne pôsobiť a komunikovať. Rozhrania API umožňujú rôznym softvérovým systémom komunikovať a zdieľať údaje medzi sebou štandardizovaným spôsobom.
2. Niektoré z najbežnejších formátov sú:
   a. JPEG (Joint Photographic Experts Group) - obľúbený formát pre digitálne fotografie vďaka svojim schopnostiam kompresie, ktoré umožňujú menšie veľkosti súborov bez výraznej straty kvality. Podporuje milióny farieb a dokáže spracovať obrázky vo vysokom rozlíšení.
   b. BMP (Bitmap) - tento formát sa bežne používa v operačných systémoch Windows a môže podporovať obrázky v odtieňoch, ako aj farebné obrázky. Nepodporuje kompresiu, takže súbory môžu byť veľké, ale je to dobrá voľba na ukladanie obrázkov vo vysokej kvalite.
   c. PNG (Portable Network Graphics) - tento formát je známy svojou bezstratovou kompresiou, čo znamená, že pri zmenšení veľkosti súboru nie je narušená kvalita obrazu. Podporuje priehľadnosť a často sa používa na webovú grafiku a digitálnu umeleckú tvorbu.
   d. TIFF (Tagged Image File Format) - tento formát sa bežne používa na ukladanie obrázkov vysokej kvality, ako sú obrázky používané pri tlači. Podporuje bezstratovú kompresiu, viacero vrstiev a zvláda obrázky v odtieňoch šedej, ako aj farebné obrázky.
3. Obrázky, videá a podobné objekty sú zvyčajne uložené v databáze Oracle pomocou dátového typu BLOB (Binary Large Object). Objekty BLOB môžu uchovávať binárne údaje, ako sú obrázky a videá, až do veľkosti 4 GB. Tento typ údajov je ideálny na ukladanie veľkých objektov, ako sú multimediálne súbory, pretože umožňuje efektívne ukladanie a získavanie

binárnych údajov.



15. GreenDi - Autorizácia a správa používateľov

VJERAN STRAHONJA, DARKO ANDROCEC, ANA KUTNJAK, LARISA HRUSTEK

15.1 Biznis pohľad prípadu

Stručný biznis prehľad platformy GreenDi je opísaný v Kapitole 14. Zameriava sa na otvorenú časť platformy GreenDi. Pokročilé funkcie sú možné len pre registrovaných používateľov (členov a správcov). Je potrebné umožniť funkcionalitu registrácie používateľov, prihlásenie registrovaných používateľov do systému s autentifikáciou a autorizáciou na vykonávanie určitých činností. Medzi potrebné funkcionality patrí pozastavenie používateľa za určitých podmienok a sledovanie histórie údajov o registrovanom používateľovi.

15.2 Definícia problému

Nasledujúcich niekoľko viet opisuje problematiku autorizácie a správy používateľov:

- Pre hlbší pohľad na informácie o rastlinách a ďalšie pokročilé funkcie je potrebné prihlásiť sa do platformy, t. j. je to možné len pre registrovaných používateľov (členov).
- Prihlasovanie je spoločné pre všetky funkcie a v zásade musí ísť o jednotné prihlásenie.
- Základným mechanizmom overovania používateľov je meno používateľa a heslo.
- Informácie o aktuálnom hesle nie sú viditeľné pre správcu ani pre žiadneho iného používateľa.
- Proces registrácie zahŕňa zadanie základných požadovaných informácií o používateľovi, ako je meno, adresa, e-mail, telefón, používateľské meno a heslo.
- Nepovinné údaje sú záujmy používateľa, dátum narodenia atď.
- Po úspešnej registrácii dostane nový používateľ automaticky predvolený typ Member (člen) a stav Active (aktívny). Zmenu stavu na Administrator (správca) a naopak, ako aj typu môže vykonať len iný aktívny správca.
- Pri registrácii člen súhlasí s podmienkami používania. Ak ich poruší, administrátori pozastavia členstvo, čo je tiež popísané v podmienkach používania.
- Bežná registrácia a vymazanie členstva je dobrovoľné a vykonáva ho používateľ.
- Každá zmena akýchkoľvek údajov používateľa sa zaznamenáva vo forme logu. Správca si môže prezrieť históriu zmien každého používateľa s výnimkou hesla.
- Na žiadosť používateľa systém umožní obnovenie hesla tak, že správca pošle používateľovi dočasné heslo prostredníctvom e-mailu alebo SMS.
- Voliteľné funkcie: Okrem prihlasovania pomocou používateľského mena a hesla je voliteľne

možné overenie pomocou služieb Fb a Google Sign-In alebo iných služieb. Napríklad webová aplikácia, ktorá používa OAuth 2.0 na prístup k rozhraniam API spoločnosti Google, musí mať autorizačné poverenia, ktoré identifikujú aplikáciu na serveri OAuth 2.0 spoločnosti Google.

15.3 Prípady použitia

15.3.1 Príbehový popis prípadu použitia

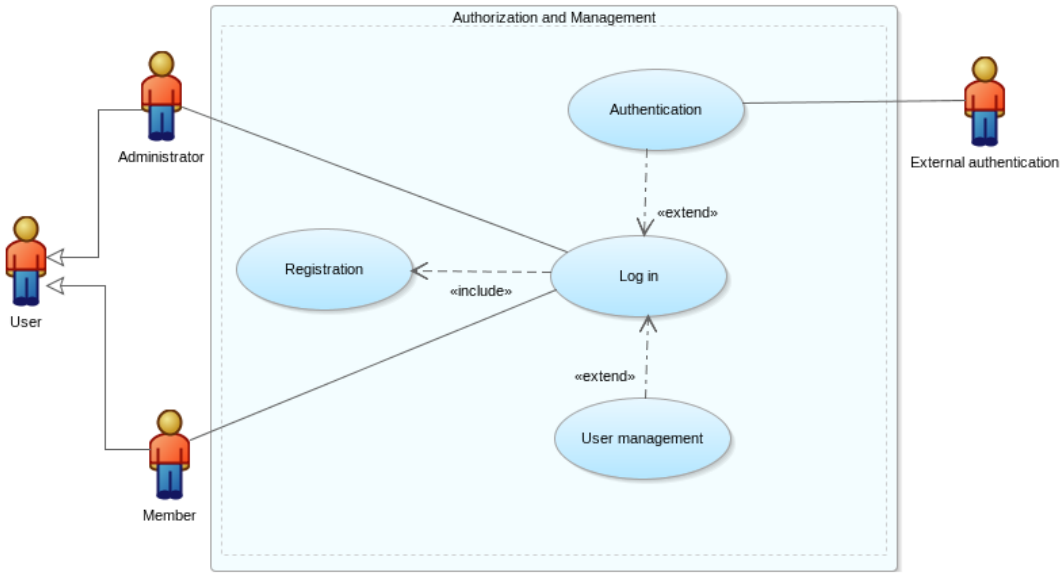
Use case diagram popisuje komunikáciu aktora v prostredí vyvíjaného systému. Use case reprezentuje funkcionalitu autorizácie a manažmentu z pohľadu používateľa. Je zobrazený na Figure 15.1.

15.3.2 Pološtruktúrovaný popis

Tabuľka 15.1 predstavuje UC.

15.3.3 Diagram prípadov použitia

Uvedený príbeh je znázornený na diagrame prípadov použitia 15.1.



Obr. 15.1: Diagram prípadov použitia - Autorizácia a správa používateľov GreenDi

15.4 Dátový model

15.4.1 Príbehový popis prípadu použitia

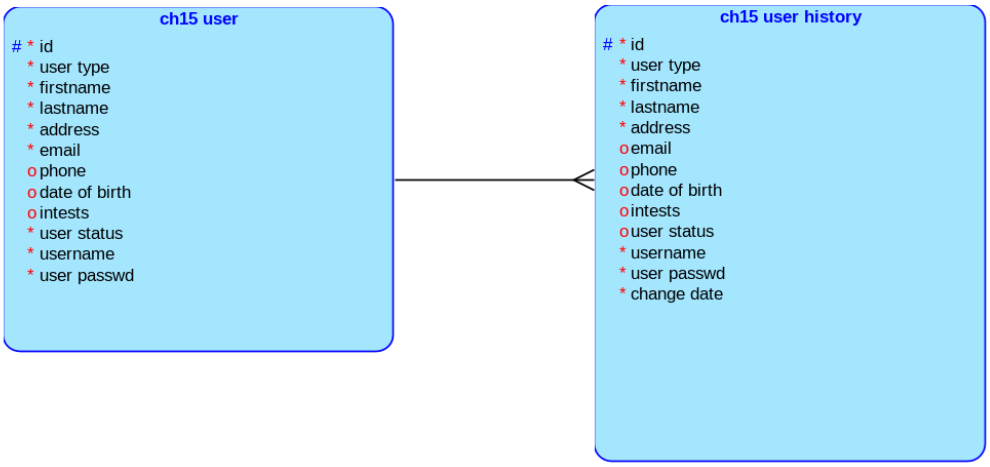
Logický dátový model (Entity-Relationship model) obsahuje 2 typy entít: User a User History. User má tieto atribúty: ID používateľa (identifikačný atribút); Typ (typ používateľa: člen, správca); Meno (meno a priezvisko používateľa; Adresa (adresa používateľa); Elektronická pošta; Telefón; Dátum narodenia (voliteľný); Záujmy (voliteľný); Stav (aktívny, pozastavený, neaktívny); Meno používateľa (neviditeľné pre správcu, zašifrované); Heslo (neviditeľné pre správcu, zašifrované) Každý používateľ má 1 alebo viac záznamov histórie používateľov. História používateľa má rovnaké atribúty ako Používateľ, je to kópia záznamu používateľa po zmene s tým, že má systémovo generovaný čas zmeny a ID osoby, ktorá zmenu vykonala (samotný člen alebo administrátor). To znamená, že zmenené údaje sa do tohto záznamu zaznamenávajú po každej zmene.

Tabuľka 15.1: Popis prípadu použitia: Autorizácia a správa používateľov.

Kľúčové slovo	**Hodnota**
ID:	*ch15-01*
Názov:	*Autorizácia a správa používateľov*
Popis:	*Autorizácia a správa používateľov.*
Hlavný aktér:	*Ktorákoľvek osoba, člen, správca*
Predpoklady:	*Prihlásenie sa zvyčajne spustí automaticky z iných prípadov použitia, keď je používateľ nastavený ako člen alebo správca.*
Postpodmienky:	*Výsledná situácia je, že používateľ nie je autorizovaný, alebo že je autorizovaný, ale je známy jeho stav (čakajúci, aktívny...) a typ (člen, administrátor).*
Hlavné	*Scenáre*
Úspešný scenár:	1. Pokročilé funkcie platformy GreenDi vyžadujú prihlásenie do systému Prihlásenie Formulár prihlásenia Prípad použitia. 2. Pri prvom prihlásení sa nový používateľ musí zaregistrovať prostredníctvom formulára Registrácia používateľa prípadu použitia Registrácia používateľa, ktorý je súčasťou prípadu použitia Prihlásenie. Proces registrácie zahŕňa zadanie povinných informácií o používateľovi, ako je meno, adresa, e-mail a telefón, ako aj nepovinných informácií, ako sú záujmy používateľa a dátum narodenia. b.Používateľ nastaví používateľské meno a heslo (sú zaznamenané v špeciálnej tabuľke v databáze, oddelené od ostatných údajov používateľa alebo zašifrované). c.Používateľ súhlasí s podmienkami používania a potvrdzuje zadané údaje. d.Predvolený typ používateľa je Člen a predvolený stav je Aktívny. 3. Údaje používateľa môže zmeniť sám používateľ pri každom ďalšom prihlásení do systému prostredníctvom registračného formulára používateľa. 4. Obnovenie hesla iniciuje používateľ z prihlasovacieho formulára, takže na e-mail používateľa sa odošle správa s dočasným heslom, ktoré trvá 15 minút. 5. Každá zmena akýchkoľvek údajov používateľa sa zaznamenáva vo forme protokolu v dátovej entite História používateľa. Správca môže prezerať históriu zmien každého používateľa s výnimkou hesla. 6. Funkcionalita prípadu použitia Správa používateľov umožňuje používateľovi typu Správca a v stave Aktívny zmeniť typ a stav iného používateľa (Správca na Člen a naopak, ako aj stav Aktívny - Pozastavený - Neaktívny). 7. K dispozícii je tabuľkové zobrazenie používateľov s možnosťou vyhľadávania (formulár Tabuľka používateľov). Na tomto formulári môže správca vybrať údaje o ľubovoľnom používateľovi na podobnom formulári ako na formulári Registrácia používateľa (Formulár s údajmi o používateľovi), zmeniť stav a typ používateľa, resetovať heslo a otvoriť formulár História používateľov, na ktorom sa zobrazujú zmeny údajov o vybranom používateľovi v tabuľkovej forme. 8. Každé ďalšie prihlásenie registrovaného používateľa iniciuje kontrolu oprávnenia používateľa, ktorá kontroluje používateľské meno a heslo, stav používateľa a to, či je správcom alebo členom.
Rozšírenia:	• *Prihlasovací formulár* • *Registračný formulár používateľa* • *Formulár tabuľky používateľov* • *Formulár s podrobnosťami o používateľovi* • *Formulár histórie používateľov*
Frekvencia používania:	*Približný maximálny počet je 100 za deň.*
Stav:	*Stav: návrh*

15.4.2 Logický dátový model

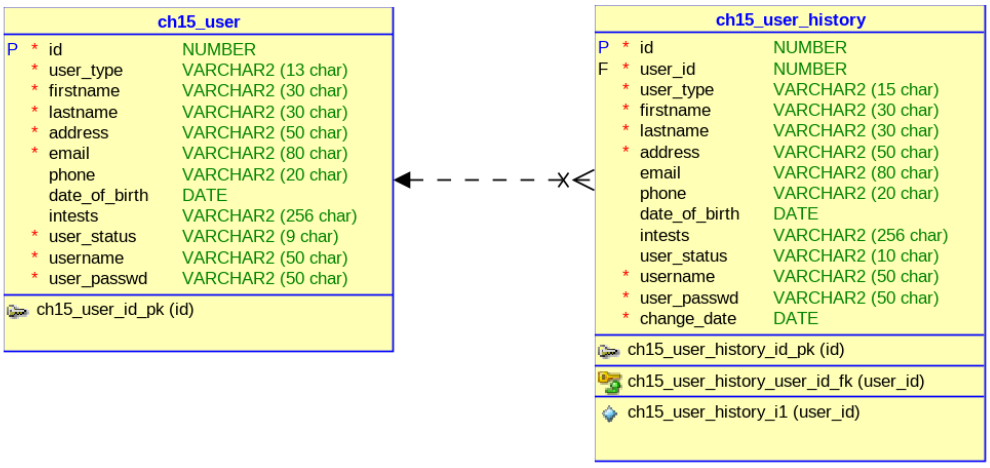
Logický dátový model je uvedený na Obrázku 15.2.



Obr. 15.2: Logický dátový model.

15.4.3 Relačný dátový model

Automatickú transformáciu z logického dátového modelu na relačný dátový model v programe Oracle SQL Data Modeler zabezpečuje funkcia *Inžinierstvo na relačný*. Výsledok je zobrazený na Obrázku 15.3.

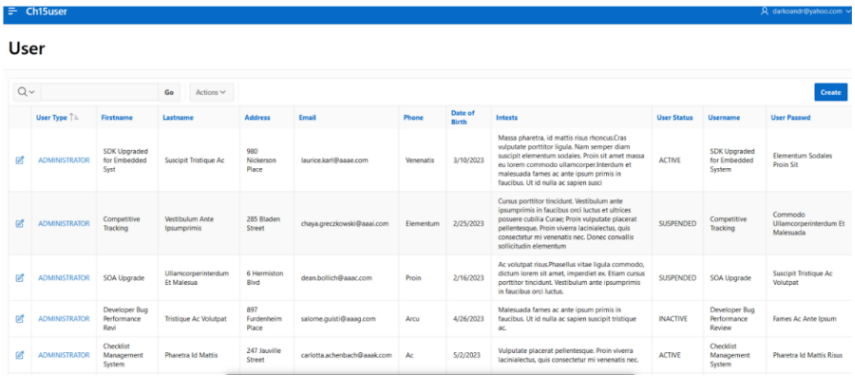


Obr. 15.3: Relačný dátový model.

Teraz nechajme Oracle SQL Data Modeler vygenerovať SQL skript. Vyberieme všetky tabuľky relačného modelu a pomocou funkcie *Súbor* > *Export* > *DDL súbor* získajte skript. Vygenerovaný skript môžeme importovať do systému APEX a spustiť ho.

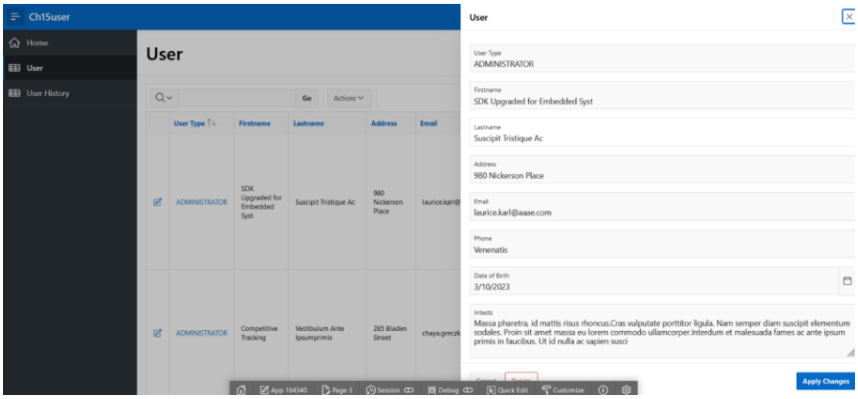
15.5 Rozhrania aplikácie

Formulár pouívateľa je uvedený na Obrázku 15.4.



Obr. 15.4: Formulár používateľa.

Vloženie údajov do formulára História používateľa je uvedené na Obrázku 15.5.



Obr. 15.5: Formulár histórie používateľov.

Funkcie, ktoré vyžadujú autorizáciu používateľa, sú opísané v Kapitole 18 Webový obchod - GreenDi.

15.6 Doplnkový učebný materiál

Nájdete tu tieto doplnkové učebné materiály:

- exportovaná aplikácia
- skripty na vytváranie, odstraňovanie a vkladanie
- video sprievodca

Všetky doplnkové vzdelávacie materiály sú k dispozícii na verejnej stránke projektu BeeAPEX. Prihláste sa ako hosťujúci používateľ (nevyžaduje sa heslo). Učebnicu nájdete v časti Knihy, skripty v priečinku Časť 2 > Kapitola15 v časti Skripty a videonávody v časti Zbierka videonávodov. Materiály pre krátke kurzy sa nachádzajú v časti Krátke kurzy.

15.6.1 Exportované aplikácie

Exportovaná aplikácia je zabalená. Inštalácia vytvorí tabuľky a naplní ich údajmi. Odinštalovanie odstráni všetky objekty databázy údajov použité v tejto aplikácii.

15.6.2 Video sprievodcovia

Video sprievodca ukazuje každý krok pri vývoji aplikácie.

15.7 Otázky

1. Aké sú zabudované funkcie overovania a autorizácie na úrovni používateľa a na úrovni aplikácie pri vývoji aplikácií pomocou Oracle APEX?
2. Ako môžeme v aplikácii alebo webovej lokalite, ktorú sami vyvíjame, používať prihlasovanie cez Facebook, Google a iné platformy?
3. Aké sú výhody pre používateľov, keď sa prihlásia do aplikácie pomocou svojich účtov na Facebooku, Googli a podobne?

15.8 Odpovede

1. a) Zabudované overovanie používateľa: APEX poskytuje vstavanú funkciu overovania používateľov, ktorá umožňuje definovať a spravovať používateľské účty priamo v prostredí APEX. Používatelia môžu mať rôzne roly a oprávnenia a na základe rolí používateľov môžete riadiť prístup ku konkrétnym stránkam aplikácie alebo komponentom. b) Autorizácia na úrovni aplikácie: APEX umožňuje definovať jemné autorizačné pravidlá na úrovni aplikácie. Môžete určiť, ktorí používatelia alebo roly majú prístup ku konkrétnym stránkam, oblastiam, tlačidlám alebo iným komponentom v rámci aplikácie.
2. V aplikácii alebo webovej lokalite, ktorú si sami vytvoríte, môžeme použiť možnosti prihlásenia, ako je Facebook, Google a iné platformy. Mnohé populárne platformy poskytujú vývojárske rozhrania API (Application Programming Interfaces), ktoré umožňujú integrovať ich funkcie prihlasovania do vlastných aplikácií alebo webových stránok. Napríklad Facebook poskytuje rozhranie API Facebook Login, ktoré umožňuje používateľom prihlásiť sa do vašej aplikácie alebo webovej lokality pomocou ich prihlasovacích údajov na Facebooku. Podobne spoločnosť Google poskytuje rozhranie Google Sign-In API, ktoré umožňuje používateľom prihlásiť sa pomocou ich účtov Google. Ak chcete implementovať tieto možnosti prihlásenia, musíte zvyčajne zaregistrovať svoju aplikáciu alebo webovú lokalitu na portáli pre vývojárov príslušnej platformy. Tento registračný proces zvyčajne zahŕňa získanie kľúčov API alebo ID klienta, ktoré potom použijete vo svojom kóde na overovanie používateľov a riadenie procesu prihlásenia.
3. Začlenením možností prihlásenia z týchto platforiem môžete používateľom ponúknuť pohodlný a známy spôsob prihlásenia do aplikácie alebo webovej lokality bez toho, aby museli vytvárať nové kontá alebo si pamätať ďalšie prihlasovacie údaje. Nie je potrebné sa registrovať a zadávať meno, adresu a podobné údaje, nevytvárajú sa nové používateľské mená a heslá, ktoré by si bolo potrebné pamätať. Okrem toho je prihlasovanie oveľa rýchlejšie.



16. Malý inovačný systém

R. Leskovar, U. Rajkovič, A.Baggia

16.1 Biznis pohľad prípadu

V súčasnom podniku sú inovatívne nápady kľúčom k úspechu a zlepšeniu výkonnosti. Zamestnanci sú vďaka svojim znalostiam o podnikových procesoch pravdepodobne najlepším zdrojom inovačných nápadov. Tieto nápady zahŕňajú rôzne pohľady, ako sú návrhy na optimalizáciu procesov, zlepšenia v oblasti riadenia vzťahov so zákazníkmi, modernizácia úloh zabezpečenia kvality atď. Preto sa vedenie v tomto konkrétnom prípade rozhodlo zbierať od zamestnancov drobné inovatívne nápady a odmeňovať ich.

16.2 Definícia problému

Inovatívne nápady v podniku sa predtým zhromažďovali prostredníctvom e-mailov a papierových formulárov. Prehľad nápadov v podniku bol časovo náročný. Kritériá hodnotenia nápadov neboli definované. Vzhľadom na vysoký počet potenciálnych nápadov, ktoré by mohli byť pre podnik prínosom, je potrebné mať dobrý prehľad. Platforma umožní zhromažďovať inovatívne nápady v malom podniku. Zamestnanci sa budú môcť prihlásiť do platformy a vkladať svoje nápady s jednoduchým popisom alebo prílohami. Keďže nápady nemajú rovnakú hodnotu (pre manažment a prevádzku), budú sa preskúmavať a hodnotiť. Keď hodnotitelia ukončia proces, nápad sa vyhodnotí ako prijatý, zamietnutý alebo je potrebná jeho revízia. Najinovatívnejšie nápady budú ocenené v závislosti od získaného hodnotenia. Za navrhnuté nápady možno získať tieto ocenenia: ZLATÝ (100 EUR), STRIEBORNÝ (80 EUR), BRONZOVÝ (60 EUR), PRÍNOSNÝ (40 EUR) alebo POĎAKOVANIE. Možnosť hodnotiť nápady budú mať len manažéri. Manažéri budú pravidelne iniciovať hodnotenie a odmeňovanie (t. j. raz za mesiac). Všetci prihlásení používatelia si budú môcť nápady po vyhodnotení prezrieť.

16.3 Prípady použitia

16.3.1 Naratívny popis

Každý zamestnanec má možnosť navrhnúť inovatívne nápady prostredníctvom prihlášky. Všetky nápady sú zhromažďované a vyhodnocované vedením. Nápady sú zoradené podľa užitočnosti a možnosti realizácie a dobré nápady sú ocenené.

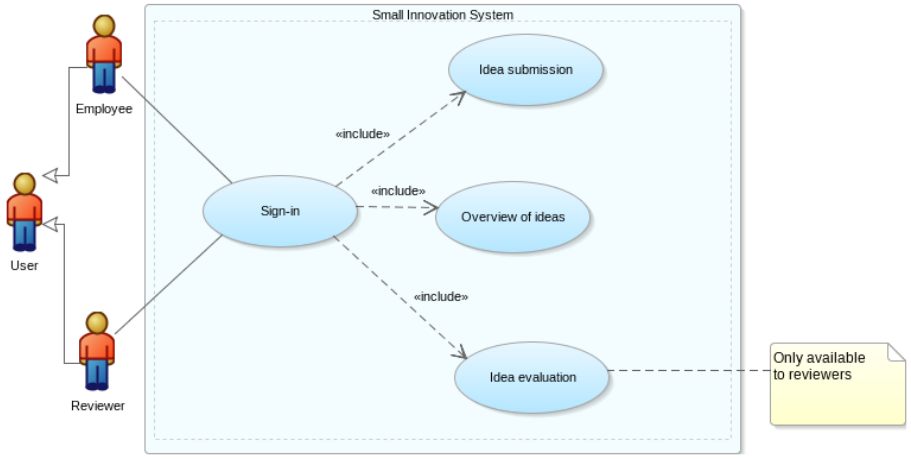
16.3.2 Pološtruktúrovaný popis

Môžeme zhrnúť tri rôzne používateľské príbehy alebo prípady použitia: predkladanie nápadov, prehľad nápadov a hodnotenie nápadov. Posledný krok - vyplácanie odmeny - nie je v tomto biznis prípade zahrnutý.

Pridanie prílohy k nápadu je vlastne rozšírením vyššie opísaného UC a mohli by sme ho opísať ako nový prípad použitia. Kvôli zrozumiteľnosti budeme pokračovať v ďalších dvoch hlavných prípadoch použitia, avšak aplikácia bude mať funkciu na pridávanie príloh k nápadu.

16.3.3 Diagram prípadov použitia

Uvedený príbeh je znázornený na diagrame prípadov použitia (pozri Obrázok 16.1).



Obr. 16.1: Diagram prípadov použitia.

Opäť platí, že pre zjednodušenie sa prípad použitia rozšírenia (pridanie prílohy k nápadu) na Obrázku 16.1 vynecháva.

16.4 Dátový model

16.4.1 Príbehový popis dátového modelu

V logickom dátovom modeli sú tri entity. Entita CH16 IDEA má osem atribútov: ID, dátum predloženia, názov, popis nápadu, názov súboru nápadu, mime typ, rozhodnutie a dátum rozhodnutia. Každý nápad navrhuje jeden zamestnanec. Základné údaje o zamestnancoch sú uložené v entite CH16 EMPLOYEE: ID, meno a priezvisko. Zamestnanec môže byť manažérom iných zamestnancov. Každý zamestnanec má len jedného manažéra, pričom manažér môže riadiť viacerých zamestnancov. Manažéri hodnotia nápady. Jeden manažér môže hodnotiť veľa nápadov, zatiaľ čo jeden nápad môže hodnotiť len jeden manažér. Každý nápad môže získať jedno ocenenie. Entita CH16 AWARD má tri atribúty: ID, popis ocenenia a sumu v eurách. Jeden typ ocenenia môže byť udelený viacerým nápadom. Pravidlá hodnotenia nie sú zahrnuté v dátovom modeli, pretože by to predstavovalo riziko zmeny aplikácie pri každej zmene pravidiel. Navrhovaným dátovým modelom sme zabezpečili: robustnosť aplikácie, rýchly vývoj a nasadenie, nízke náklady na vývoj a údržbu a flexibilitu.

16.4.2 Logický dátový model

Uvedený príbeh je prezentovaný na logickom dátovom modeli (pozri Obrázok 16.2).

Tabuľka 16.1: Popis prípadu použitia: Predloženie nápadu.

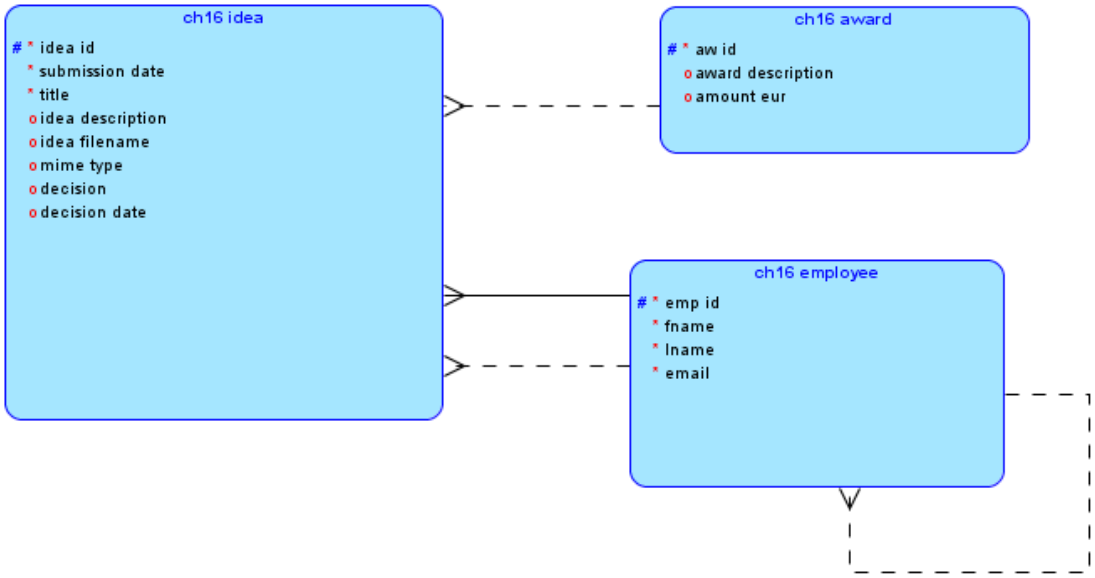
Kľúčové slovo	**Hodnota**
ID:	*Ch16-01*
Názov:	*Predloženie nápadu*
Popis:	*Zamestnanec sa prihlási do platformy a zadá nápad. Každý nápad má názov a krátky popis pridaný ako príloha (súbor dokumentu).*
Hlavný aktér:	*Zamestnanec*
Predpoklady:	*Zamestnanec musí mať používateľské konto Oracle APEX pre aplikáciu Malý Inovačný System. Prístup k webovej aplikácii musí byť povolený.*
Podmienky po skončení:	*Po úspešnom zadaní nápadu je nápad a jeho podrobný popis v prílohe k dispozícii ostatným používateľom portálu SIS.*
Hlavné	*Scenáre*
Úspešný scenár:	1. Otvorte webový prehliadač a prihláste sa do aplikácie SIS 2. V ponuke vyberte položku Nápady 3. Vyberte položku Pridať nový nápad 4. Zadajte názov nápadu a pridajte prílohu 5. Zadajte názov súboru a typ prílohy 6. Potvrďte vloženie kliknutím na tlačidlo Vytvoriť 7. Skontrolujte vložené údaje
Rozšírenia:	• *1a. Prihlásenie zlyhá* • *1a* Rozšírenie:* • *1a1. Zobrazenie chybovej správy* • *1a2. Otvorenie prihlasovacieho okna* • *4a. Chyba dátového typu* • *4a* Rozšírenie:* • *4a1. Zobrazenie chybovej správy* • *7a. Chyba dátového typu* • *7a* Rozšírenie:* • *7a1. Zobrazenie chybovej správy*
Frekvencia použitia:	*Spoločnosť dostáva v priemere 3 inovatívne nápady týždenne.*
Stav:	*Dokončené*
Vlastník:	*Zamestnanec s prístupom k aplikácii*
Priorita:	*Stredná*

Tabuľka 16.2: Popis prípadu použitia: Prehľad nápadov.

Kľúčové slovo	**Hodnota**
ID:	*Ch16-02*
Názov:	*Prehľad nápadov*
Popis:	*Zamestnanec sa prihlási do platformy a skontroluje stav nápadu alebo iných nápadov.*
Hlavný aktér:	*Zamestnanec*
Predpoklady:	*Zamestnanec musí mať používateľské konto Oracle APEX pre aplikáciu Malý Inovatívny System. Prístup k webovej aplikácii musí byť povolený.*
Podmienky po skončení:	-
Hlavné	*Scenáre*
Úspešný scenár:	1. Otvorte webový prehliadač a prihláste sa do aplikácie SIS 2. V ponuke vyberte položku Nápady 3. Prehľadávať uvedené nápady a ich stav 4. V ponuke vyberte položku Prehľad 5. Zobraziť grafy údajov o predkladaní nápadov
Rozšírenia:	• *1a. Prihlásenie zlyhá* • *1a* Rozšírenie:* • *1a1. Zobrazenie chybovej správy*
Frekvencia použitia:	*Možnosť prezerať nápady majú zamestnanci, hodnotitelia a správcovia. Celkovo pristupujú k platforme približne 5-krát denne.*
Stav:	*Dokončené*
Vlastník:	*Zamestnanec s možnosťou prihlásenia*
Priorita:	*Nízka*

Tabuľka 16.3: Popis prípadu použitia: Hodnotenie nápadu.

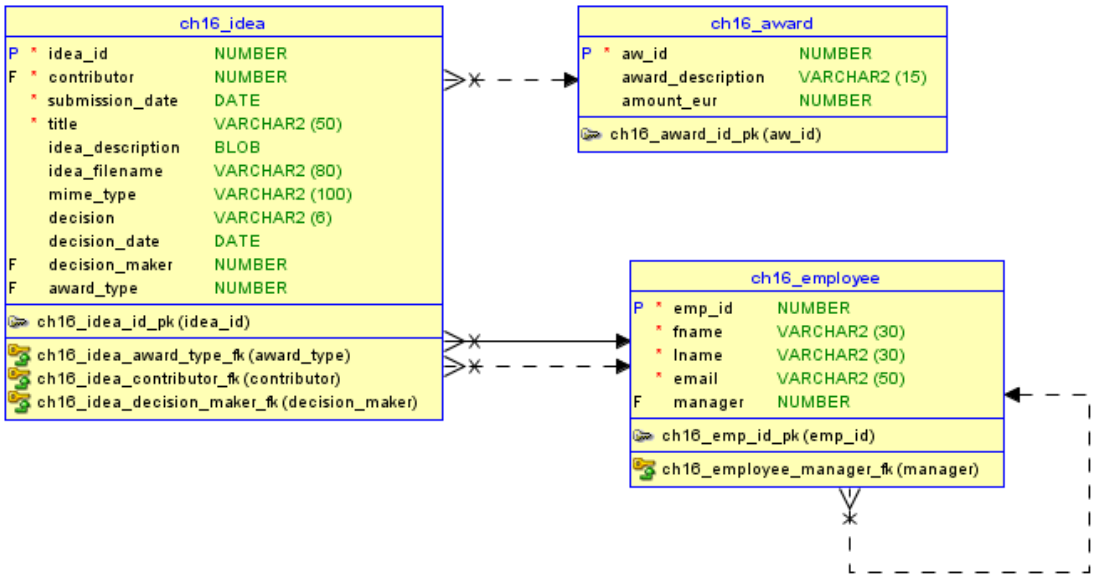
Kľúčové slovo	**Hodnota**
ID:	*Ch16-03*
Názov:	*Hodnotenie nápadov*
Popis:	*Recenzent sa prihlási do platformy a zhodnotí nápad. V tomto prípade použitia je definované aj ocenenie.*
Hlavný aktér:	*Recenzent*
Predpoklady:	*Používateľ zamestnanec musí byť pridaný do role CH16_Reviewer. Prístup k webovej aplikácii musí byť povolený.*
Podmienky po skončení:	*Vybraný nápad sa vyhodnotí a určí sa navrhované ocenenie.*
Hlavné	*Scenáre*
Úspešný scenár:	1. Otvorte webový prehliadač a prihláste sa do aplikácie SIS 2. V ponuke vyberte položku Nápady na preskúmanie 3. Identifikujte nápad a pridajte podrobnosti pomocou tlačidla Upraviť 4. Vyberte rozhodnutie zo zoznamu hodnôt (Prijaté, Odmietnuté, Revízia) 5. Vyberte typ ocenenia zo zoznamu hodnôt (Zlaté, Strieborné, Bronzové, Prispievateľ, Poďakovanie). 6. Potvrdenie aktualizácie
Rozšírenia:	• *1a. Prihlásenie zlyhá* • *1a* Rozšírenie:* • *1a1. Zobrazenie chybovej správy* • *1a2. Otvorenie prihlasovacieho okna* • *2a. Chyba dátového typu* • *2a* Rozšírenie:* • *2a1. Zobrazenie chybovej správy* • *3a. Chyba dátového typu* • *3a* Rozšírenie:* • *3a1. Zobrazenie chybovej správy* • *4a. Chyba dátového typu* • *4a* Rozšírenie:* • *4a1. Zobrazenie chybovej správy* • *5a. Chyba dátového typu* • *5a* Rozšírenie:* • *5a1. Zobrazenie chybovej správy*
Frekvencia použitia:	*Manažéri preskúmajú navrhované nápady raz za mesiac.*
Stav:	*Dokončené*
Vlastník:	*Recenzent*
Priorita:	*Vysoká*



Obr. 16.2: Logický dátový model.

16.4.3 Relačný dátový model

Automatickú transformáciu z logického dátového modelu na relačný dátový model v Oracle SQL Data Modeler zabezpečuje funkcia *Inžinierstvo na relačný*. Výsledok je zobrazený na Obrázku 16.3.



Obr. 16.3: Relačný dátový model.

Nástroj Oracle SQL Data Modeler tiež generuje skript SQL na vytváranie tabuliek, sekvencií a triggrov. Vyberte všetky tabuľky v relačnom modeli a pomocou funkcie *Súbor* > *Export* > *Súbor DDL* získame takýto skript:

```
CREATE TABLE CH16_AWARD (
    AW_ID NUMBER NOT NULL
```

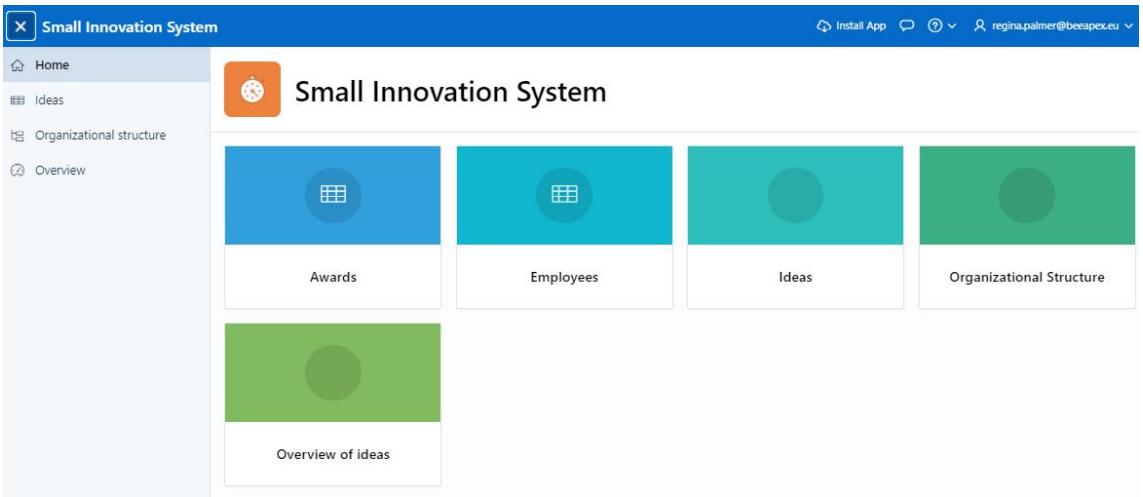
```
...
);

CREATE TABLE CH16_EMPLOYEE (
    EMP_ID NUMBER NOT NULL
...
);
```

Teraz je čas vygenerovať tabuľky v databáze Oracle. Vygenerovaný skript môžeme importovať a spustiť v APEX.

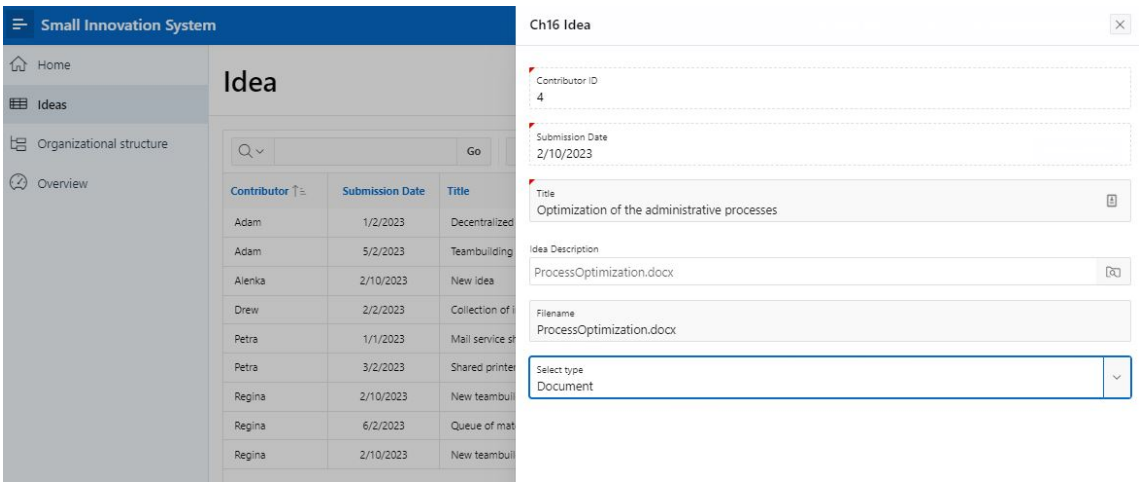
16.5 Rozhrania aplikácie

Na nasledujúcich obrázkoch sú uvedené rozhrania aplikácie pre obe základné roly (zamestnanca a kontrolóra). Najprv je na Obrázku 16.4 uvedený ovládací panel pre zamestnanca.



Obr. 16.4: Ovládací panel pre zamestnancov.

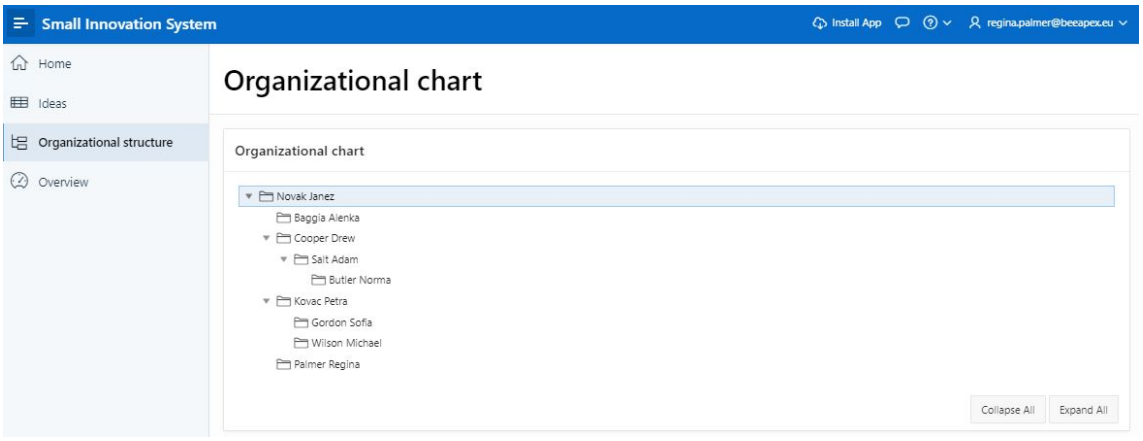
Stránka na predkladanie nápadov s prílohami a jej podrobnosti sú uvedené na Obrázku 16.5.



Obr. 16.5: Odoslanie nápadu s prílohami.

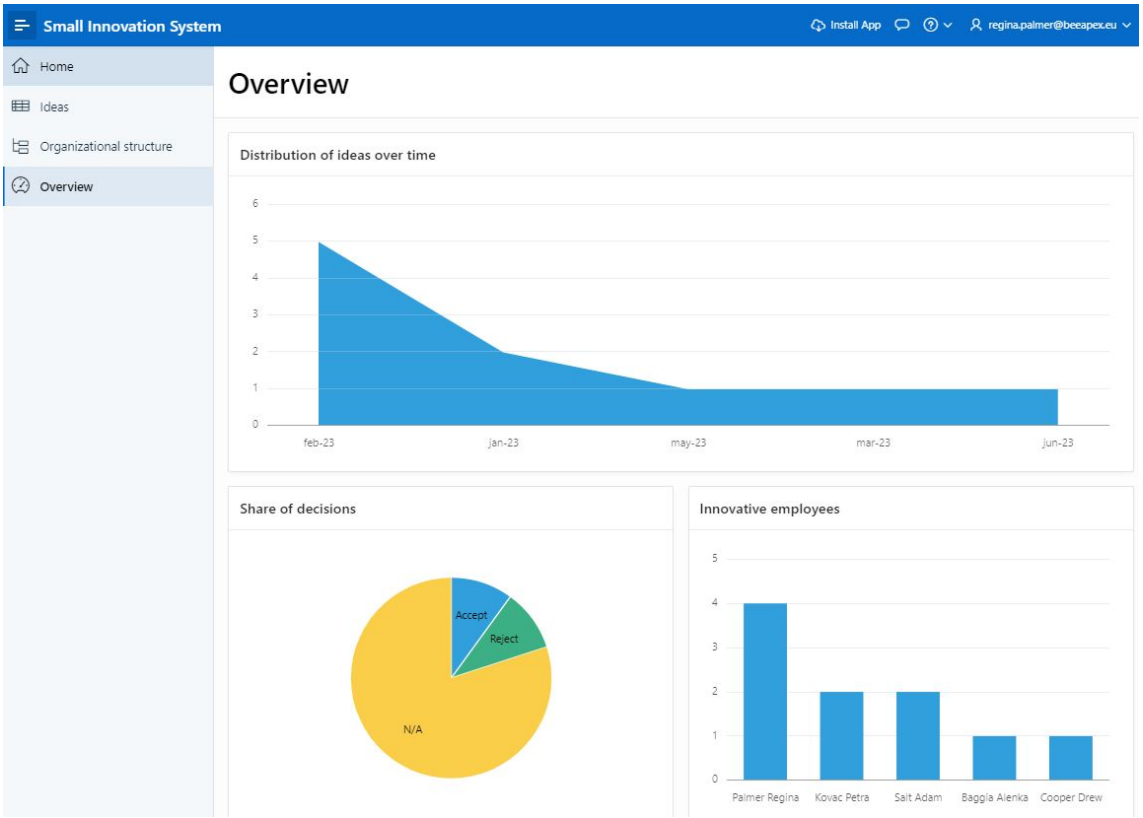
Každý zamestnanec má možnosť zobraziť organizačnú štruktúru v spoločnosti, ako je zobra-

zená na Obrázku 16.6.



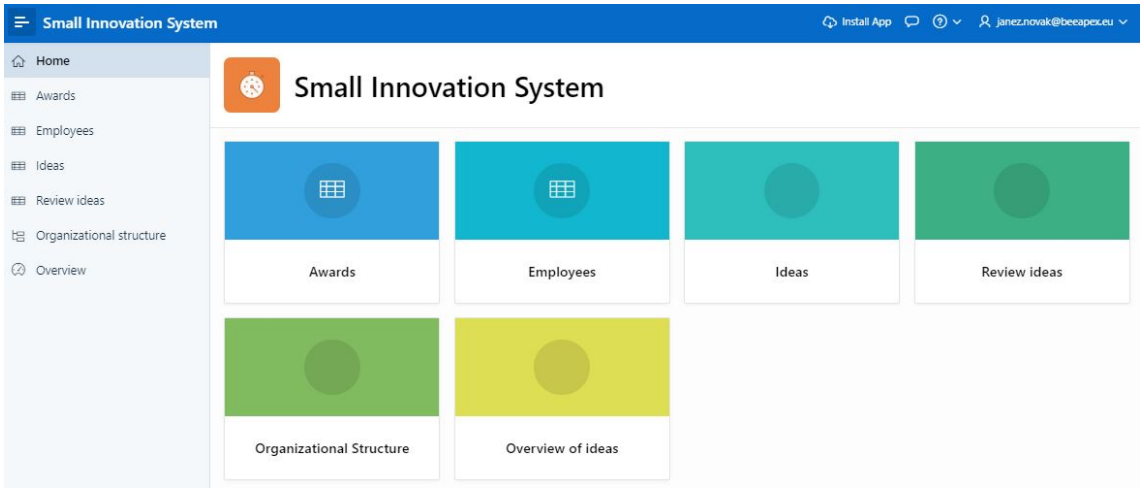
Obr. 16.6: Organizačná štruktúra spoločnosti.

Každý zamestnanec si tiež môže pozrieť panel grafov prezentujúcich rozloženie nápadov v čase, podiel prijatých, zamietnutých a revidovaných nápadov a zamestnancov s najvyšším počtom inovatívnych nápadov v spoločnosti (pozri Obrázok 16.7).



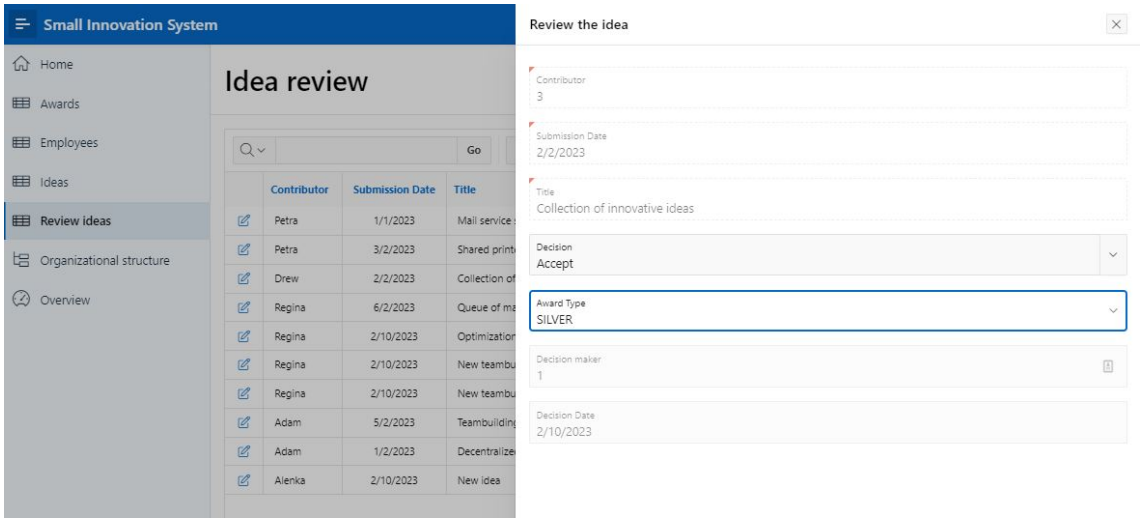
Obr. 16.7: Prehľad Malého Inovačného Systému.

Okrem opísaných činností má recenzent možnosť aktualizovať údaje o oceneniach a zamestnancoch, ako je znázornené na Obrázku 16.8. Stránka administrácie je dostupná len pre administrátorov, nie pre recenzentov.



Obr. 16.8: Ovládací panel recenzenta.

Stránka na hodnotenie nápadov je uvedená na Obrázku 16.9. Táto stránka je prístupná len recenzentovi. Recenzent vyberá rozhodnutie a prípadne typ ocenenia.



Obr. 16.9: Stránka na hodnotenie nápadov.

16.6 Doplnkový učebný materiál

Nájdete tu tieto doplnkové učebné materiály:

- skript na vytvorenie
- skript na naplnenie tabuliek
- skript na zrušenie tabuliek
- exportovaná zabalená aplikácia
- video, ktoré ukazuje, ako generovať aplikáciu zo skriptu

Všetky doplnkové vzdelávacie materiály sú k dispozícii na verejnej stránke projektu BeeAPEX. Prihláste sa ako hosťujúci používateľ (nevyžaduje sa heslo). Učebnicu nájdete v časti Knihy, skripty v priečinku Časť 2 > Kapitola16 v časti Skripty a videonávody v časti Zbierka videonávodov. Materiály pre krátke kurzy sa nachádzajú v časti Krátke kurzy.

16.6.1 Exportovaná aplikácia

Exportovaná aplikácia je zabalená. Inštalácia vytvorí tabuľky a vyplní údaje. Odinštalovanie odstráni všetky objekty databázy údajov použité v tejto aplikácii.

Zabalená aplikácia sa otestuje a spustí sa v novom pracovnom priestore, ak sú splnené nasledujúce požiadavky:

- používateľ APEX je pridaný pred spustením aplikácie.

Ak používateľ nemá pridelenú príslušnú rolu, importovaná aplikácia sa zrúti. Po páde aplikácie z dôvodu nesplnených požiadaviek je potrebné vymazať súbory cookie webového prehliadača (napr. Firefox: Nastavenia > Súbory cookie a údaje o stránkach > Správa údajov).

16.6.2 Video sprievodca

Video sprievodca ukazuje všetky kroky pri vývoji aplikácie.

16.7 Otázky

1. Kde sú definované roly používateľov v Kapitole 16?
2. Ktorý vzťah v logickom dátovom modeli umožňuje prezentáciu organizačnej štruktúry v Kapitole 16?
3. Prečo sú niektoré polia na stránke pre hodnotenie nápadov v Kapitole 16 vypnuté?

16.8 Odpovede

1. V dátovom modeli Kapitoly 16 nie sú definované žiadne roly. Používajú sa len v prostredí APEX.
2. Vzťah spájajúci tabuľku Employee so sebou samou (rekurzívny vzťah) umožňuje prezentáciu organizačnej štruktúry v Kapitole 16.
3. V Kapitole 16 sú niektoré polia vypnuté, aby sa vytvorila lepšia používateľská skúsenosť a zabezpečila konzistentnosť údajov, pretože používateľ nemusí pridávať údaje, ktoré už sú v databáze.



17. Manažment podnikových procesov

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA

17.1 Biznis pohľad prípadu

V tejto knihe je pri vývoji aplikácií so službou APEX zdôrazňovaná dôležitosť údajov a prístupu založenom na nich. Prístup založený na údajoch má však svojho súrodenca: procesne riadené aplikácie. Ako meno naznačuje, procesne riadené aplikácie sa zameriavajú na procesy. Žiadna dobrá, na údajoch založená aplikácia nemôže zanedbať procesný aspekt a naopak, ani žiadna procesne riadená aplikácia nemôže zanedbať ten údajový. Implicitné alebo explicitné zahrnutie oboch aspektov umožňuje, aby každá aplikácia vyhovovala svojmu účelu. Je veľmi dôležité, aby vývojári dôkladne poznali podnikové procesy. Akýkoľvek typ organizácie, bez ohľadu na veľkosť, ročné príjmy, vlastníctvo, odvetvie a poslanie, riadi procesy v podniku. Bežné triedy podnikových procesov, ktoré sú charakteristické pre väčšinu organizácií je možné zaradiť do nasledujúcich kategórií:

- predaj a marketing, zahŕňajú čiastkové procesy ako je získavanie a udržanie si zákazníkov, oslovovanie potenciálnych zákazníkov, riadenie vzťahom so zákazníkmi a marketingové kampane.
- riadenie dodávateľského reťazca zahrňujúce čiastkové procesy ako je koordinácia a riadenie toku tovarov a služieb, ich obstarávanie, výroba a distribúcia.
- finančný manažment zahrňujúci riadenie finančných zdrojov organizácie, účtovníctvo, rozpočtovanie a finančné výkazníctvo.
- riadenie ľudských zdrojov zahrňujúce procesy ako je riadenie pracovnej sily organizácie, nábor, školenia a rozvoj zamestnancov.
- zákaznícky servis, ktorý zahŕňa poskytovanie podpory a pomoci zákazníkom, riešenie sťažností a riešenie problémov.
- operačný manažment zahrňuúci čiastkové procesy týkajúce sa riadenia každodenných podnikových operácií, výroby, logistiky a riadenia zásob.
- informačné technológie zahrňujúce riadenie podnikových informačno-technologických zdrojov a systémov, hardvéru, softvéru a riadenie údajov.

Tieto procesy sú považované za kritické, pretože podporujú základné funkcie organizácie a umožňujú dosahovanie jej cieľov a zámerov. Zlepšenie efektívnosti a účinnosti týchto procesov môže mať významný vplyv na celkový výkon a úspech organizácie. V snahe o jasné pochopenie a definíciu typického podnikania, OASIS OPEN vyvinul Universal Business Language, ktorý popisuje podnikové procesy a dokumenty (viď. [5]). Softvérový priemysel začal v 90. rokoch 20.

storočia s integráciou procesov vynájdením Business Process Execution Language (BPEL) a neskôr Business Process Modeling Language (BPML). Veľkí hráči ako Microsoft, IBM, SAP, Oracle, ako aj nespočetné množstvo menších spoločností prispeli k softvérovým platformám, ktorých cieľom bolo zabezpečiť bezproblémovú integráciu podnikových funkcií. Okrem softvérového sveta sa aj iné odvetvia sústreďovali na procesy cez objektív špecifických podnikov z 80. rokov s iniciatívami a štandardmi riadenia kvality.

Od 90. rokov 20. storočia zohrávajú disciplíny manažment podnikových procesov (Business Process Management - BPM) a reinžinierstvo podnikových procesov (Business Process Reengineering - BPR) dôležitú úlohu pri zlepšovaní výkonnosti podnikov. BPM sa zameriava na prezentáciu a dokumentáciu podnikových postupov, operácií a procesov využitím grafov, vývojových diagramov alebo iných vizuálnych pomôcok. Cieľom je pochopiť a zlepšiť existujúce procesy, prípadne navrhnúť nové, zvýšiť efektivitu a znížiť ich neefektívnosť. BPR je využívané za účelom zásadného prehodnotenia a radikálneho prepracovania podnikových procesov s cieľom dosiahnuť výrazné zlepšenie výkonu a produktivity podniku. Zahŕňa analýzu a návrh tokov práce a procesov v rámci podniku s cieľom byť efektívnejší a schopný prispôsobiť sa meniacemu podnikateľskému prostrediu. BPR môže zahŕňať aj využitie technológie a informačných systémov za účelom automatizácie a zlepšenia rozhodovania.

V BPM a BPR je možné využiť niekoľko typov diagramov:

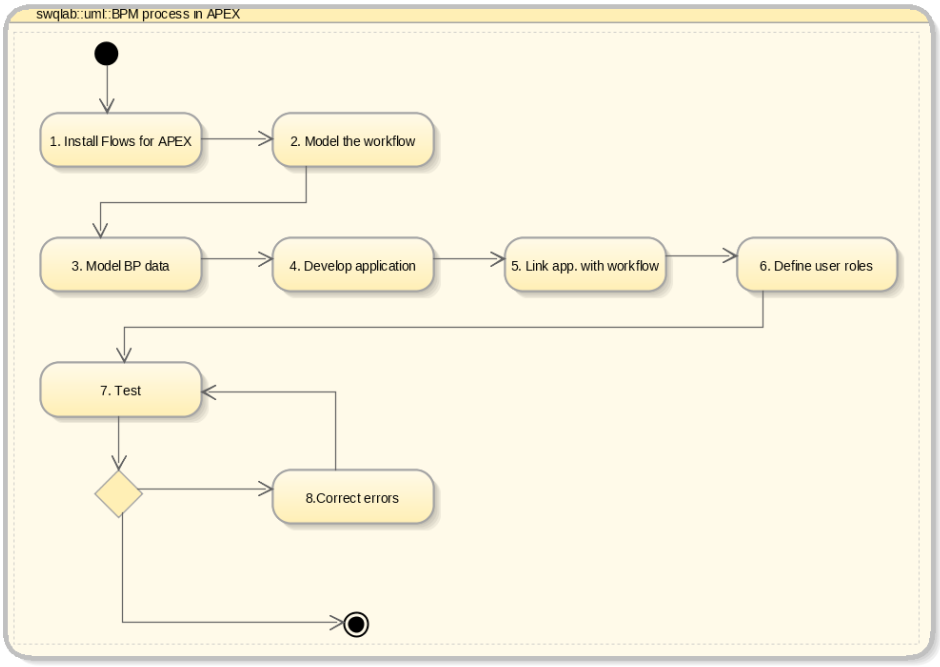
- Vývojové diagramy: Jednoduchý a intuitívny spôsob znázornenia procesu ako série krokov, rozhodnutí a slučiek.
- Diagramy plaveckých dráh: Typ vývojového diagramu, ktorý pridáva vizuálnu reprezentáciu toho, kto alebo čo je zodpovedný za každý krok v procese.
- Procesné mapy: Poskytujúce základný pohľad na kroky a činnosti zapojené do procesu, často vrátane vstupov, výstupov a rozhodovacích bodov.
- BPMN (Business Process Model and Notation [4]): Notácia pre modelovanie podnikových procesov, štandardizovaná konzorciom Object Management Group (OMG).
- EPC (Event-driven Process Chain): Typ vývojového diagramu, ktorý namiesto aktivít používa udalosti ako hnaciu silu procesu.
- IDEF (Integration Definition for Function Modeling): Metóda používaná v softvérovom inžinierstve na modelovanie a analýzu procesov a systémov.

Výber typu diagramu pre prezentáciu pracovných tokov závisí od úrovne potrebných detailov, účelu modelu a cieľového publika. V tomto bode sa zameriame len na BPMN. Hlavnými dôvodmi sú, že: a) BPMN má štandardnú notáciu a širokú akceptovateľnosť v rôznych odvetviach a b) BPMN je implementovaný ako typ modelovacieho diagramu vo Oracle Application Express (APEX) funkcii (niekedy označovanej ako rozšírenie) s názvom Flows for APEX. Flows for APEX je open source, licencovaný pod veľmi permisívnou MIT licenciou. Flows for APEX umožňuje vývojárom vytvárať a nasadzovať webové aplikácie pomocou vizuálneho rozhrania typu drag-and-drop. Flows for APEX poskytuje spôsob ako automatizovať a zjednodušiť zložité podnikové procesy ich rozdelením na sériu krokov a úloh a následne nimi sprevádzať používateľov.

Flows for APEX môžu obsahovať rôzne prvky, ako sú stránky, formuláre, zostavy a dialógy, a možno ich nakonfigurovať tak, aby obsahovali logiku vetvenia, podmienené vetvenie, overovanie údajov a spracovanie chýb. Vizuálne rozhranie Flows for APEX uľahčuje vývojárom navrhovať, zostavovať a testovať podnikové procesy bez potreby rozsiahleho programovania. Dá sa integrovať s ďalšími Oracle APEX komponentmi, ako sú SQL, PL/SQL, a REST službami, za účelom vytvárania kompletných webových aplikácií ako aj na vytváranie vlastných riešení pre špecifické podnikové potreby, ako je napríklad riadenie pracovného toku, proces vzdelávania nových zákazníkov o hodnote produktu alebo služby (customer on-boarding) a správu dát. Flows for APEX poskytuje organizáciám spôsob ako zefektívniť a automatizovať zložité podnikové procesy, zvýšiť produktivitu a znížiť chyby a neefektívnosť. Výborné návody na Flows for APEX sú dostupné na tejto stránke. Ako teda je možné prepojiť APEX ako programovacie prostredie nenáročné na

programovanie a Flows for APEX so zabudovanou štandardnou BPML prezentáciou?

Diagram aktivity (viď. Obrázok 17.1) zobrazuje zjednodušenú prezentáciu integrácie aplikácie Flows for APEX a APEX.



Obr. 17.1: Integrácia Flows for APEX s aplikáciou APEX.

Kroky *3. modelovanie dát, 4. vývoj aplikácie, 6. definovanie užívateľských rolí, 7. testovanie* a *8. odstránenie chýb z aplikácie* sú spoločné pre všetky APEX aplikácie. Krok *1. inštalácia Flows for APEX* je podobná inštalácii akejkoľvek inej zbalenej aplikácii. Novinkami sú *2. modelovanie pracovného toku* a *5. prepojenie aplikácie APEX s pracovným tokom*.

Čitateľom, ktorí nie sú oboznámení s manažmentom podnikových procesov (BPM) a BPMN (business process modeling notation) diagramami odporúčame vyhľadať populárne študijné materiály a naštudovať si už spomínanú špecifikáciu BPMN [4]. Ako krátke pripomenutie pre všetkých čitateľov spomenieme najčastejšie používané prvky BPMN diagramov:

- Počiatočná udalosť: Označuje začiatok procesu.
- Úloha: Predstavuje jednu jednotku práce, napríklad aktivitu alebo krok v procese.
- Brána: Predstavuje rozhodovací bod v procese, ako je napríklad rozvetvenie toku alebo podmienená vetva.
- Sekvenčný tok: Spája prvky v diagrame a predstavuje tok riadenia z jedného prvku do druhého.
- Konečná udalosť: Indikuje koniec procesu.
- Bazén/Dráha: Predstavuje zoskupenie súvisiacich úloh používané na definovanie rolí a zodpovedností rôznych účastníkov procesu.
- Tok správ: Predstavuje tok správ medzi účastníkmi procesu, ako napríklad komunikácia medzi dvomi systémami.
- Dátové objekty:Predstavuje časť údajov, ktorá sa používa alebo vytvára v procese, ako je napríklad faktúra alebo záznam o zákazníkovi.
- Úložisko dát: Predstavuje kontajner na ukladanie dát, ktoré sa používajú alebo vytvárajú v procese, napr. databáza alebo súborový systém.

Tieto prvky je možné kombinovať a tak vytvárať podrobné a presné modely podnikových procesov umožňujúce organizáciám identifikovať oblasti pre zlepšenie a optimalizovanie ich pracovných

tokov.

17.1.1 Inštalácia Flows for APEX

Stiahnite si zip súbor z tohto URL a rozbaľte všetky súbory do vášho počítača. Postupujte podľa pokynov poskytnutých na tomto odkaze. Pred inštaláciou Flows for APEX doplnku, skontrolujte požiadavky. Požiadavky sú splnené ak používa svoj bezplatný APEX pracovný priestor, bezplatný/-platený OCI alebo Oracle Academy účet. Dve ďalšie úlohy, ktoré je potrebné spraviť po inštalácií sú:

- udeliť oprávnenie "create job"v pracovnom priestore. Vy alebo administrátor inštancie APEX musíte vytvoriť pripojenie k databáze (s rozhraním OCI, SQL Developer, SQL Plus, TOAD) a zadať príkaz. Toto oprávnenie nie je možné udeliť v rámci APEX.
- nakonfigurovať pracovný priestor, predvolenú aplikáciu a informácie v rámci aplikácie "Flows for APEX". Tento krok je možné vykonať aj neskôr, napr. po vytvorení vlastnej procesnej aplikácie.

Aj keď sa niekedy nazýva funkcia alebo doplnok, inštalácia vytvorí nové dátové objekty (tabuľky, views) a aplikáciu s menom "Flows for APEX". Táto aplikácia je administratívne rozhranie, ktoré umožňuje:

- monitorovanie inštancií pracovných tokov na dashboarde (Dashboard),
- vizuálne vytváranie nových tokov práce (Flow Management) a
- vykonávanie pracovných tokov vytváraním inštancií, dokončením úloh, ktoré patria inštanciám a testovaním pracovných tokov (Flow Monitor).

17.1.2 Inštalovanie Sample Process Flow aplikácie

Sample process Flow aplikácia sa nachádza v rovnakom zip súbore ako Flows for APEX. Inštalácia je jednoduchá. Zaoberá sa nárokmi zamestnancov na vrátenie výdavkov spojených s prácou. Zamestnanec pripraví žiadosť o vrátenie výdavkov, ktorá je schvaľovaná manažérom. Ak je nárok zamestnanca na vrátenie výdavkov zamietnutý, zamestnanec je o tom informovaný. Pri schválených žiadostiach, účtovnícke oddelenie pripraví platbu, vykoná bankovú transakciu a nastaví stav platby. Ak je suma výdavku vrátená zamestnancovi, tak je inštancia tohto postupu prác ukončená. Aplikácia je veľmi podobná skutočnému procesu v akejkoľvek organizácii. Vďaka dobrému zdrokumentovaniu procesu predstavuje skvelý študijný materiál.

17.1.3 Cvičenia zamerané na čítanie a prax

Existujú veľmi dobré návody na Flows for APEX. Začnite s touto stránkou. Na konci tohto cvičenia nájdete BPMN návod a APEX integračný návod.

Čítanie a precvičovanie vám umožní vyvinúť si vlastnú procesne riadenú aplikáciu. Precvičenie všetkých cvičení bude čitateľovi pravdepodobne trvať niekoľko hodín. Odporúčame, aby si čiteľ vyskúšal tvz. learning-by-doing. Zvyšok tejto kapitoly poskytuje rovnakú šablónu ako ostatné kapitoly v tejto učebnici (od definície problému po aplikáciu) s dodatočným zahrnutím BPML a integráciou aplikácie Flows for APEX. Rozsah aplikácie vyvinutej v Kapitole 17 je však zúžený na základnú funkčnosť a nevenuje sa detailom autorizácie. Poskytuje návody ako:

- definovať pracovné toky v aplikácii Flows for APEX
- spustiť inštanciu definovaného procesu v rámci vyvíjanej aplikácie
- monitorovať tok inštancií v rámci vytvorenej aplikácie
- dokončiť každú úlohu konkrétnej inštancie pracovného toku v rámci vyvíjanej aplikácie

Aj aplikácia so základnými funkciami však môže predstavovať dobrú výzvu pre nadšeného študenta, no nie?

17.2 Definovanie problému

Stredne veľká výrobná spoločnosť dostáva od zákazníkov dopyty na produkty vyrobené na mieru. Každá požiadavka môže mať viac súvisiacich dokumentov so špecifikáciami, požadovanými normami, schémami, náčrtmi a podobne. Zákazník môže mať aj viac dopytov. Predajca pripravuje a klasifikuje dokumenty týkajúce sa dopytu do troch kategórií: výrobné, finančné a obchodné. Na vyhodnotenie jedného dopytu je potrebné pripraviť viacero dokumentov pre osoby s rozhodovacou právomocou. Výrobný manažér a jeho tím vyhodnotia výrobný aspekt dopytu. Poskytujú odborný posudok o schopnosti spoločnosti vyrobiť požadovaný produkt. Ak je možné produkt vyrobiť, tím poskytne odhadované trvanie výrobného procesu v dňoch. Finančný manažér a jeho tím vyhodnotia finančnú stránku dopytu. Poskytujú odborný posudok o schopnosti spoločnosti zabezpečiť financovanie a odhadnú očakávaný zisk. Konečné hodnotenie a rozhodnutie prijíma generálny riaditeľ a jeho tím. Zvážia realizovateľnosť (výrobnú, finančnú), odhadnú dôležitosť zákazníka a rozhodnú. V závislosti od schopnosti spoločnosti vyrábať a financovať obchodnú príležitosť, ako aj obchodné vyhliadky, generálny riaditeľ rozhodne, ako zákazníkovi odpovedať: a) poslať obchodný návrh s cenou a očakávanou dobou dodania alebo b) správou informovať zákazníka, že spoločnosť neakceptuje dopyt.

Zlé obchodné rozhodnutia (akceptovanie obchodného návrhu alebo jeho odmietnutie) môžu mať katastrofálne dopady na výrobnú spoločnosť, či už v podobe ušlých tržieb, nízkeho zisku, zlej povesti v dôsledku prekročených dodacích termínov alebo aj bankrotu. Uvedená situácia spôsobuje "bolesti hlavy" všetkým zainteresovaným: predajcovi, manažérovi výroby, finančnému manažérov a riaditeľovi. IT oddelenie je zodpovedné za poskytnutie užívateľsky prívetivej, spoľahlivej a bezpečnej softvérovej podpory pre popísaný postup. Ich úlohou je teda pripraviť takéto riešenie s využitím Flows for APEX.

Na vývoj aplikácie je potrebné si nainštalovať Flows for APEX. V tejto kapitole je analyzovaný tento obchodný prípad (popis, pološtruktúrovaný popis, diagram prípadu použitia (use case), procesný model) a vývoj aplikácie. Naučíte sa ako:

- ako definovať procesný model (workflow) pre tento prípad,
- ako vytvoriť dátový model a vyvinúť APEX aplikáciu,
- ako prepojiť vyvinutú aplikáciu s definovaným pracovným tokom.

Majte na pamäti, že autorizácia používateľa nebude implementovaná, pretože by to rozšírilo túto kapitolu nad rámec primeraného úsilia študentov.

17.3 Prípady použitia

17.3.1 Naratívny popis

Štyria aktéri sú zapojení do štyroch prípadov použitia. Každý aktér je zodpovedný za jeden prípad použitia: predajca pripravuje dokumentáciu týkajúcu sa dopytu, vedúci výroby poskytuje hodnotenie výrobných aspektov, finančný manažér poskytuje hodnotenie finančných aspektov a generálny riaditeľ prijíma rozhodnutie. Use case diagram (see Figure 17.2) predpokladá, že dizajnér pracovného toku rozhodne, či majú výrobný a finančný manažér pracovať paralelne alebo sekvenčne. Ak pracujú paralelne a jeden alebo obaja sa rozhodnú, že požiadavky na dopyt nie je možné splniť, bola síce vynaložená zbytočná námaha (náklady), ale proces je možné dokončiť rýchlejšie. Ak sa dizajnér pracovného toku rozhodne pre sekvenčný prístup, pracovný tok zaberie viac času, ale je vynaloženého menej zbytočného úsilia.

17.3.2 Pološtruktúrovaný popis

UML navrhuje pre každý prípad použitia aj pološtruktúrovaný popis. V tabuľkách 17.1, 17.2, 17.3 a 17.4 sú uvedené štyri rôzne pološtrukturované popisy.

Tabuľka 17.1: Popis prípadu použitia: príprava dokumentácie k dopytu.

Kľúčové slovo	**Hodnota**
ID:	*ch17-01*
Názov:	*Príprava dokumentácie k dopytu*
Popis:	*Predajca vytvorí nový dopyt a pridá súvisiace dokumenty*
Hlavný aktér:	*Predajca*
Predpoklady:	*Zákazník už poslal všetky dokumenty súvisiace s dopytom*
Podmienky po skončení:	-
Hlavné	*Scenáre*
Úspešný scenár:	1. Pridajte nový dotaz 2. Pridajte súvisiace dokumenty ku konkrétnemu dopytu 3. Potvrďte tlačidlom “Confirm” alebo zamietnite tlačidlom “Cancel”
Rozšírenia:	-
Frekvencia použitia:	*Približne 250 ročne, v priemere 5 týždenne.*
Stav:	*[Vo vývoji]*
Vlastník:	*Predajca*
Priorita:	*Vysoká*

Tabuľka 17.2: Popis prípadu použitia: zhodnotenie výrobného aspektu dopytu.

Kľúčové slovo	**Hodnota**
ID:	*ch17-02*
Nadpis:	*Zhodnotenie výrobného aspektu dopytu*
Popis:	*Vedúci výroby a jeho tím poskytnú odborné stanovisko, či je spoločnosť schopná vyrobiť výrobok, a ak áno, ako dlho (v dňoch) by trvala výroba daného množstva. Odôvodnenie stanoviska je povinné.*
Hlavný aktér:	*Vedúci výroby*
Predpoklady:	Pripravená dokumentácia z dopytu
Podmienky po skončení:	-
Hlavné	*Scenáre*
Úspešný scenár:	1. Vložte stanovisko 2. Potvrďte tlačidlom “Confirm” alebo zamietnite tlačidlom “Cancel”
Rozšírenia:	-
Frekvencia použitia:	*Približne 250 ročne, v priemere 5 týždenne.*
Stav:	*[Vo vývoji]*
Vlastník:	*Vedúci výroby*
Priorita:	*Vysoká*

Tabuľka 17.3: Popis prípadu použitia: zhodnotenie finančného aspektu dopytu.

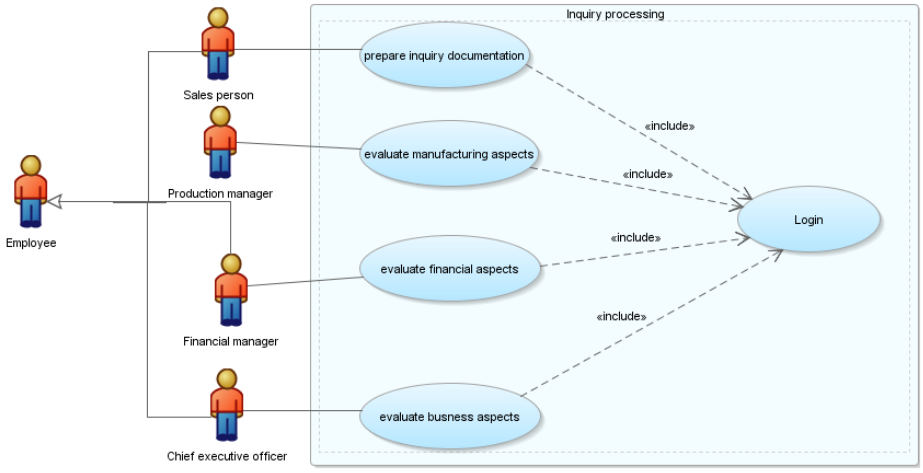
Kľúčové slovo	**Hodnota**
ID:	*ch17-03*
Nadpis:	*Zhodnotenie finančného aspektu dopytu*
Popis:	*Finančný manažér a jeho tím poskytnú odborné stanovisko, či je spoločnosť schopná financovať položku výrobku, a ak áno, aký by bol očakávaný zisk. Odôvodnenie stanoviska je povinné.*
Hlavný aktér:	*Finančný manažér*
Predpoklady:	Pripravená dokumentácia k dopytu
Podmienky po skončení:	-
Hlavné	*Scenáre*
Úspešný scenár:	1. Vložte stanovisko 2. Potvrďte tlačidlom “Confirm” alebo zamietnite tlačidlom “Cancel”
Rozšírenia:	-
Frekvencia použitia:	*Približne 250 ročne, v priemere 5 týždenne.*
Stav:	*[Vo vývoji]*
Vlastník:	*Finančný manažér*
Priorita:	*Vysoká*

Tabuľka 17.4: Popis prípadu použitia: zhodnotenie dopytu z hľadiska celého podniku.

Kľúčové slovo	**Hodnota**
ID:	*ch17-04*
Nadpis:	*Zhodnotenie dopytu z hľadiska celého podniku*
Popis:	*Generálny riaditeľ a jeho tím rozhodnú, či zákazníkovi poskytnú obchodnú ponuku, alebo ju odmietnu. Odôvodnenie stanoviska je povinné a vyžaduje ho predstavenstvo.*
Hlavný aktér:	*Generálny riaditeľ*
Predpoklady:	Prípady použitia 17.1, 17.2 a 17.3 sú ukončené.
Podmienky po skončení:	-
Hlavné	*Scenáre*
Úspešný scenár:	1. Vložte stanovisko 2. Potvrďte tlačidlom “Confirm” alebo zamietnite tlačidlom “Cancel”
Rozšírenia:	-
Frekvencia použitia:	*Približne 250 ročne, v priemere 5 týždenne.*
Stav:	*[Vo vývoji]*
Vlastník:	*Generálny riaditeľ*
Priorita:	*Vysoká*

17.3.3 Diagram prípadu použitia

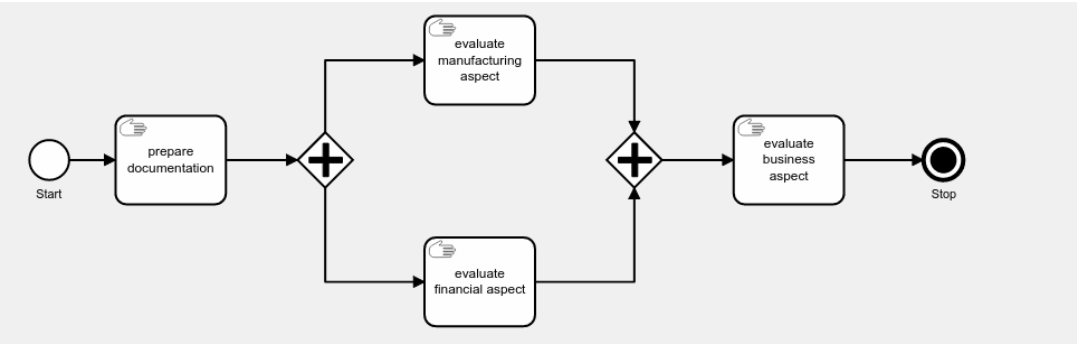
Use case diagram môže znázorňovať aj zovšeobecnenie prípadu. V danom kontexte má zamestnanec štyri špeciálne inštancie (správcov polí) a každému manažérovi je priradený jeden prípad použitia. Všetky štyri prípady použitia sú znázornené na Obrázku 17.2.



Obr. 17.2: Use case diagram.

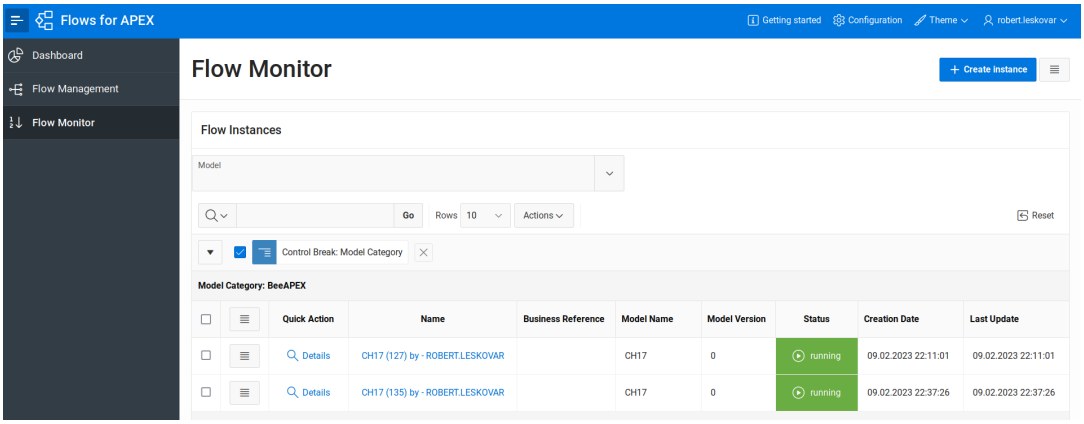
17.4 Model pracovného toku (Workflow model)

Počiatočný model pracovného toku zobrazuje prípravnú úlohu, ktorá sa potom rozdeľuje na dve paralelné úlohy (výrobné a finančné hodnotenie). Po tom ako sú predchádzajúce úlohy dokončené, pokračuje pracovný tok poslednou úlohou (hodnotenie z hľadiska podniku) (viď. 17.3).



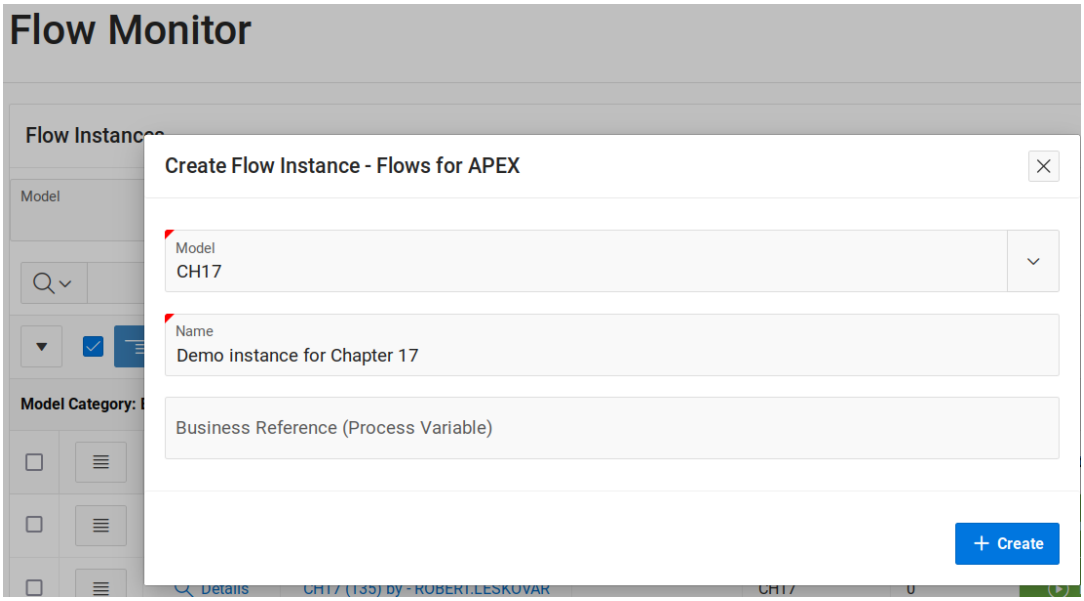
Obr. 17.3: Workflow - processing inquiry - BPMN diagram.

Všetky štyri úlohy sú definované ako “Manual task”, konečná udalosť (Stop) je definovaná ako “Terminate end event”. Pracovný tok je možné okamžite testovať vo Flow Monitor (vo Flows for APEX). Uložte tok ako CH17. Potom prejdite na Flow Monitor a kliknite “Create instance” tlačidlo (viď Obrázok 17.4).



Obr. 17.4: Vytváranie inštancií.

Vyberte model (CH17) a zadajte odlišný názov inštancie toku CH17 (viď. Obrázok 17.5).



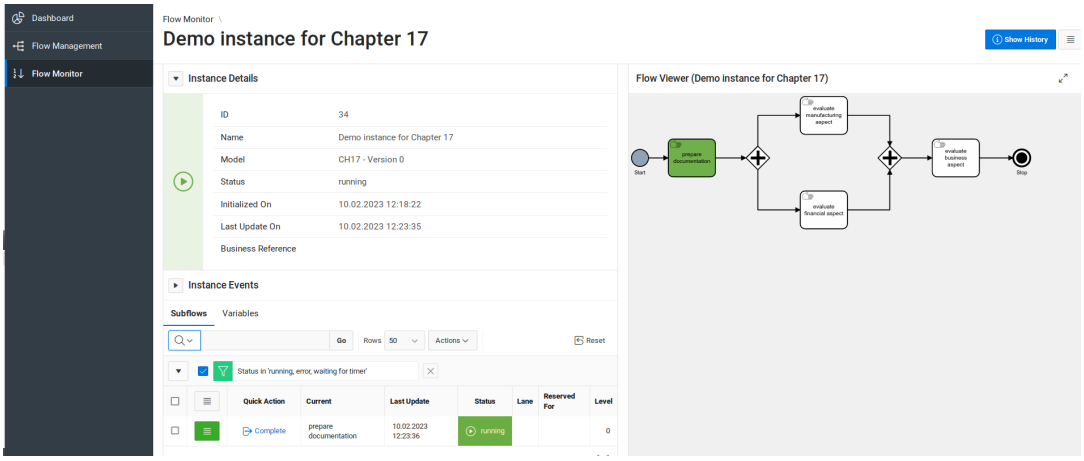
Obr. 17.5: Vytvorenie inštancie toku CH17.

Objaví sa nová inštancia. Kliknite na označenú ikonu na Obrázku 17.6 a vyberte “Start”.



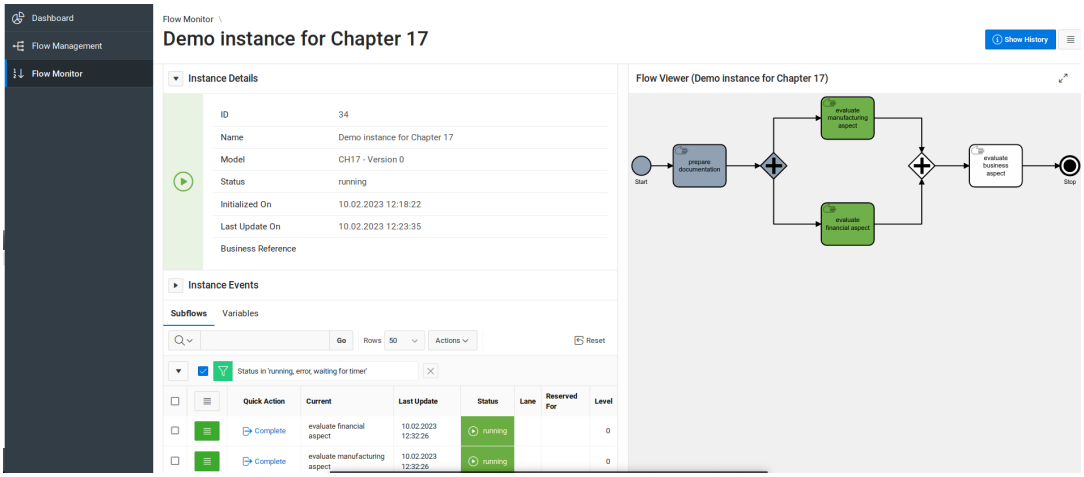
Obr. 17.6: Spustenie demo inštancie.

Stav inštancie sa zmení na “running”. Kliknite na “Details” pri demo inštancii. Nastavte vedľa seba (side-by-side) zobrazenie a kliknite “Complete” v stĺpci “Quick Action” (viď. obrázok 17.7).



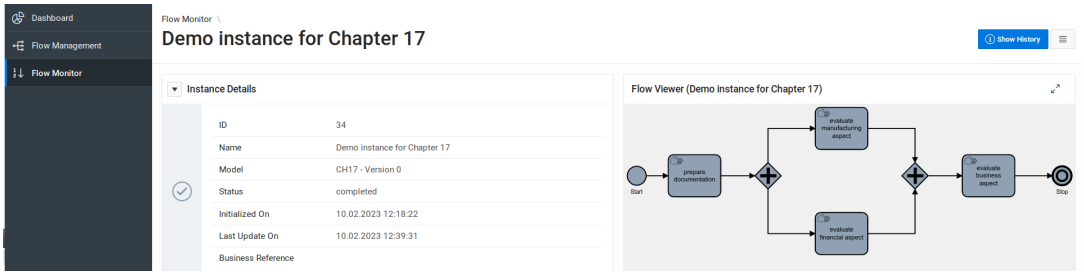
Obr. 17.7: Dokončenie prvej úlohy v demo inštancii.

Teraz tok pokračuje s paralelnou bránou a dvomi úlohami (viď. Obrázok 17.8).



Obr. 17.8: Vykonávanie úloh v paralelnej bráne.

Kliknite na “Complete” v stĺpci “Quick Action” pre obe úlohy a pokračujte poslednou úlohou. Zopakujte dokončenie pre poslednú úlohu a inštancia je dokončená (viď. Obrázok 17.9).



Obr. 17.9: Dokončená demo inštancia.

Celá história inštancie sa zobrazí po kliknutí na “Show history”. Týmto sme dokázali, že pracovný tok CH17 je spustiteľný a vieme modelovať dátové a aplikačné rozhranie.

17.5 Dátový model

17.5.1 Naratívny popis dátového modelu

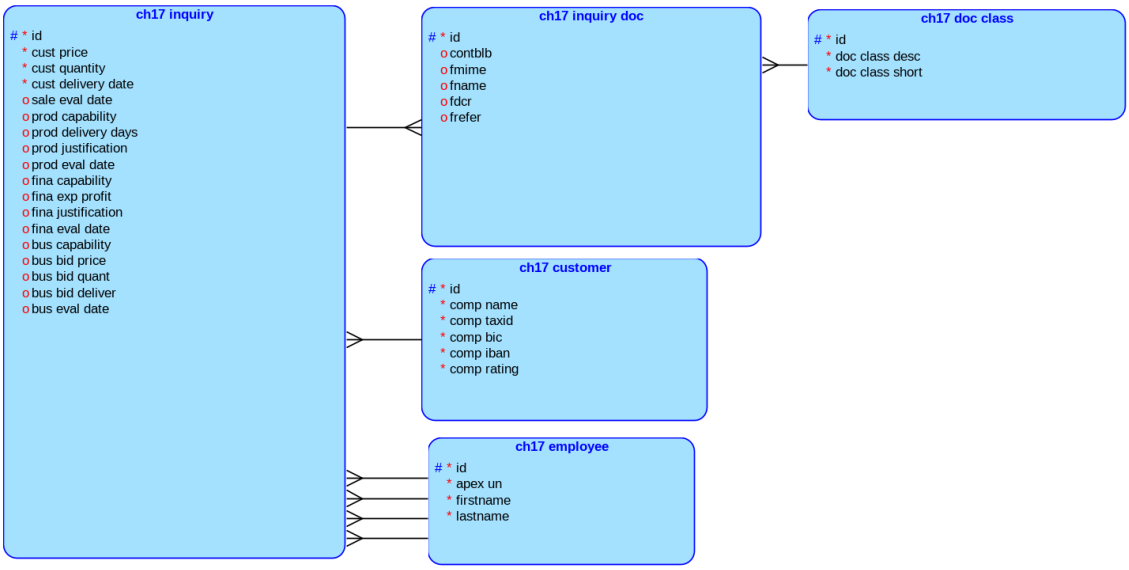
Najdôležitejšou entitou je **ch17 inquiry**. Inštancia **ch17 inquiry** súvisí s jednou inštanciou **ch17 custimer** zatiaľ čo zákazník môže mať viacero dopytov. Každá inštancia **ch17 inquiry** musí mať aspoň tri ďalšie inštancie **ch17 document**. Každá inštancia **ch17 document** patrí jednej inštancii **ch17 document class**.

Atribúty entity **ch17 inquiry** sú: ID, požiadavky zákazníka (cena objednávky, množstvo, dátum dodania), výrobné aspekty (schopnosť výroby, dodacia lehota pre požadované množstvo, odôvodnenie), finančné aspekty (finančná spôsobilosť, očakávaný zisk z požadovaného množstva) a podnikové aspekty (rozhodnutie využiť alebo nevyužiť príležitosť, odôvodnenie).

Pre ostatné entity zadefinujeme len obmedzenú množinu atribútov, aby bol tento prípad kompaktnejší.

17.5.2 Logický dátový model

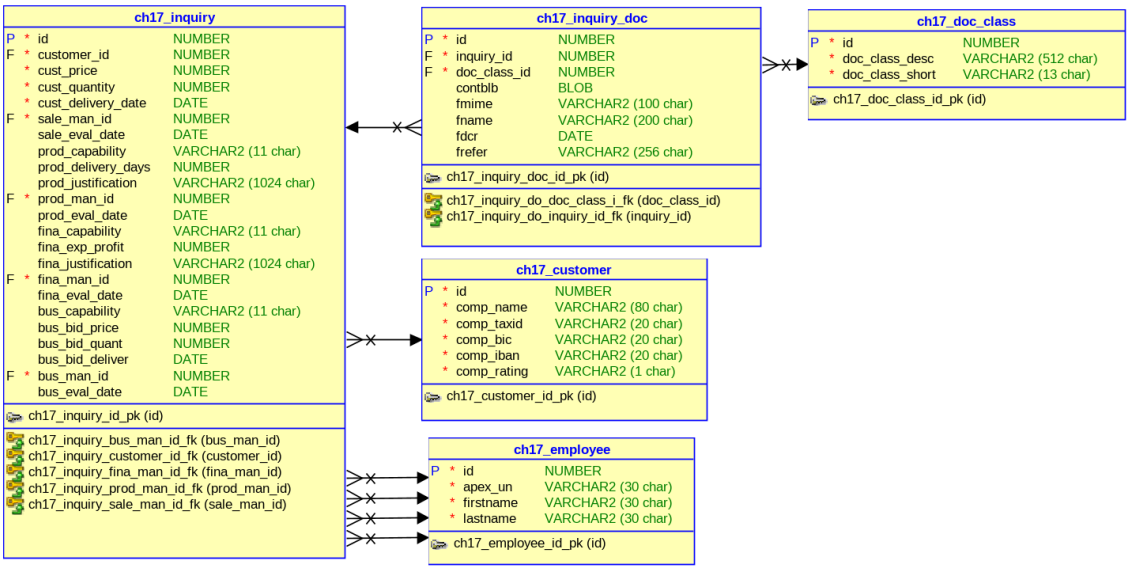
Logický dátový model je zobrazený na Obrázku 17.10.



Obr. 17.10: Logický dátový model.

17.5.3 Relačný dátový model

Relačný dátový model je zobrazený na Obrázku 17.11.



Obr. 17.11: Relačný dátový model.

17.5.4 Quick SQL for generating SQL script

Quick SQL je praktický nástroj na urýchlenie vývoja dátového modelu. Dá sa použiť na prototypovanie vygenerovaním SQL skriptu v rámci APEX (generovať aplikáciu zo skriptu) alebo použiť reverzné inžinierstvo na logický dátový model importovaním SQL skriptu do SQL Developer Data Modeler a transformovať relačný na logický model. Dátový model uvedený v tejto kapitole možno opísať pomocou nasledujúceho 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
```

Jednou z výhod Quick SQL je jeho kompaktnosť a jednoduchosť. Počet riadkov v Quick SQL je polovicou SQL skriptu a počet znakov je menší ako jedna tretina SQL skriptu. Vynaložený čas a úsilie na naučenie sa Quick SQL je dobrou investíciou. APEX tiež poskytuje užívateľsky prívetivý editor s automatickým dopĺňaním, avšak výhody Quick SQL sú viac využiteľné používateľmi ovládajúcimi SQL jazyk.

17.5.5 Sekvencia a dve uložené funkcie

V tomto prípade nás aplikačné požiadavky nútia použiť niektoré ľahko pochopiteľné koncepty databázy Oracle:

- mechanizmus sekvencie - generuje celé čísla definovaním počiatočnej hodnoty, maximálnej hodnoty, prírastku atď. Pomocou tohto mechanizmu naša aplikácia vygeneruje jedinečné číslo na identifikáciu dopytu a zostavenie názvu inštancie toku.
- uložené funkcie v jazyku PL/SQL prevezmú vstupné parametre a vrátia jednu hodnotu. Naša aplikácia bude využívať dve užívateľom definované funkcie: jednu na získanie veľkosti dokumentu uloženého v stĺpci BLOB a druhú na počítanie počtu dokumentov spojených s konkrétnym dopytom.

Sekvenciu a uložené funkcie môžeme definovať v SQL Workshop > SQL Command. Sekvencia má názov, minimálnu hodnotu, maximálnu hodnotu, prírastok, počiatočnú hodnotu a niektoré ďalšie vlastnosti, ktoré pre nás momentálne nie sú dôležité:

```
CREATE SEQUENCE   "CH17_SEQ_INQUIRY"  MINVALUE 1 MAXVALUE 999999999999
INCREMENT BY 1 START WITH 100 CACHE 1
NOORDER  NOCYCLE  NOKEEP  NOSCALE  GLOBAL;
```

Zadefinujeme aj dve uložené funkcie, ktoré sú v kóde vysvetlené ako komentáre - dve pomlčky na začiatku riadkov sú komentáre. Prvá funkcia sa nazýva 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;
```

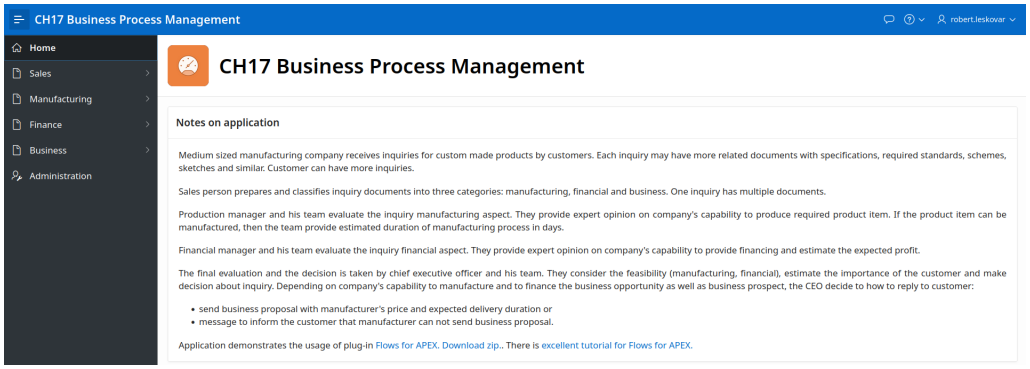
Druhá funkcia sa nazýva ch17_count_docs:

```
create or replace function ch17_count_docs(xinqid in number) return number
-- Function ch17_count_docs returns the number of documents associated with
-- particular inquiry. Function parameter 1 is the ID of inquiry.
-- If the length of the BLOB in the table ch17_inquiry_doc is greater than 0,
-- then we assume that the content is stored. So, function adds 1 to variable
-- wnumdocs, which is used as a counter. When all matching rows in the
-- table ch17_inquiry_doc are examined, than counter function returns counter.
is
wnumdocs number := 0;
CURSOR c_blob is
select contblb from ch17_inquiry_doc where inquiry_id = xinqid;
Begin
FOR r_blobs IN c_blob
LOOP
if dbms_lob.getlength(r_blobs.contblb) > 0 then
wnumdocs := wnumdocs + 1;
end if;
END LOOP;
RETURN  wnumdocs;
END ch17_count_docs;
```

17.6 Rozhrania aplikácie

Odporúčame čitateľovi, aby si pred návrhom aplikácie nainštaloval zabalenú aplikáciu pre túto kapitolu. Odkaz nájdete v doplnkovom učebnom materiáli k tejto kapitole. Uvedené čísla strán sa vzťahujú na pripravenú zabalenú aplikáciu. Čitateľ môže samozrejme vychádzať aj z iných strán, ktoré popisujú rovnakú funkčnosť ako je popísaná v tejto sekcii.

Rozhranie aplikácie poskytuje funkcie pre štyroch aktérov: obchodného manažéra, výrobného manažéra, finančného manažéra a výkonného manažéra. Prvýkrát je vygenerované so sprievodcom a všetkými funkciami (postupne: webová aplikácia, o stránke, riadenie prístupu, hlásenie aktivity, možnosti konfigurácie, spätná väzba a výber štýlu, spolu 21 strán). Štandardne sa vygeneruje aj prihlasovacia stránka. Domovská stránka obsahuje HTML na popis podnikovej situácie. Poskytuje odkazy na požadovaný plug-in a návod (viď. Obrázok 17.12, Strana 1 v zbalenej aplikácii).



Obr. 17.12: Domovská stránka “CH17 Business Process Management” aplikácie.

Niektoré LOV a Plug-iny boli nastavené alebo zahrnuté v predchádzajúcom vývoji aplikácií v užšom zmysle.

17.6.1 Zoznam hodnôt v Shared Components

V tejto aplikácii sú definované nasledujúce zoznamy hodnôt (LOV):

- CH17_LOV_CAPABILITY: typ je “static”, zobrazované hodnoty sú (YES, NO, CONDITIONAL, Not Available), návratové hodnoty sú (YES, NO, CONDITIONAL, N.A.). Tento LOV sa používa na naplnenie troch odlišných schopností (výrobná, finančná a podniková).
- CH17_LOV_CRATING: typ je “static”, zobrazované hodnoty sú (Excelent rating, Good rating, Edge acceptable rating, Not acceptable rating), návratové hodnoty sú (A, B, C, D). Tento LOV slúži na zadanie hodnotenia zákazníkov.
- CH17_LOV_CUSTOMER: typ je “dynamic”, na základe tabuľky CH17_CUSTOMER, zobrazovaná hodnota je DOC_CLASS_DESC a návratová hodnota je ID. Tento LOV sa používa pri formulároch a zostavách, aby prispel k čitateľnosti a zabránil zadávaniu nesprávneho ID zákazníka.
- CH17_LOV_DOC_CLASS: typ je “dynamic”, na základe tabuľky CH17_DOC_CLASS, zobrazovaná hodnota je COMP_NAME a návratová hodnota je ID. Tento LOV sa používa na formulároch a zostavách, aby prispel k čitateľnosti a zabránil zadávaniu nesprávnej triedy dokumentu súvisiaceho s dopytom.
- CH17_LOV_EMPLOYEE: typ je “dynamic”, na základe SQL query v tabuľke CH17_EMPLOYEE:

  ```
  SELECT FIRSTNAME || ' ' || LASTNAME as d,
  id as r
  FROM CH17_EMPLOYEE order by LASTNAME, FIRSTNAME
  ```

  Tento LOV sa používa pri formulároch a správach, aby prispel k čitateľnosti a zabránil zadávaniu nesprávnej osoby, ktorá pripravuje dokumentáciu a vyhodnocuje dopyt. Operátor zreťazenia sa používa na vytvorenie zobrazovaného reťazca s krstným menom zamestnanca, medzerou a priezviskom.
- CH17_LOV_INQUIRY: typ je “dynamic”, na základe SQL query v tabuľke CH17_INQUIRY:

  ```
  SELECT
  'Instance:' || ch17_inquiry.id || ' (Customer: ' || comp_name || ',
  product:' || cust_product_item  as d,
  ch17_inquiry.id as r
  FROM ch17_inquiry, ch17_customer
  WHERE ch17_inquiry.customer_id = ch17_customer.id;
  ```

Tento LOV sa používa pri formulároch a správach, aby prispel k čitateľnosti a zabránil zadávaniu nesprávneho ID dopytu. Operátor zreťazenia sa používa na vytvorenie dlhého reťazca zobrazenia s množstvom informácií.

17.6.2 Plug-iny v Shared Components

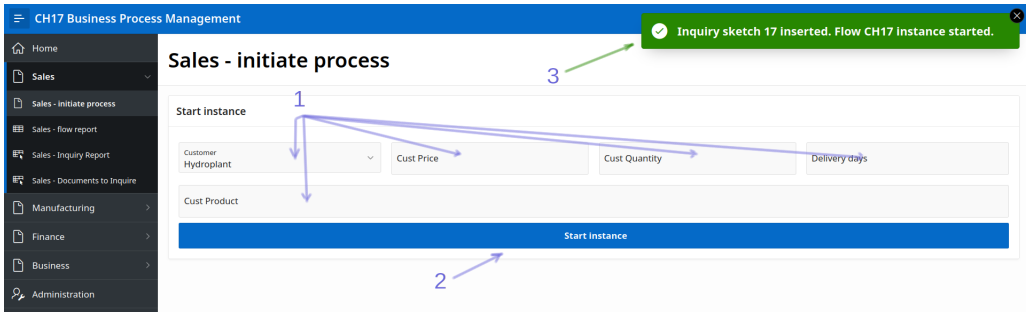
Nasledovné plug-iny sú importované do Flows for APEX: Manage Flow Instance, Manage Flow Instance Step, Manage Flow Instance Variables and Viewer. Tieto plug-iny sú skopírované z nainštalovaného Flows for APEX plug-in.

17.6.3 Manažér predaja (Sales manager)

Rozhrania aplikácie umožňujú manažérovi predaja:

- spustiť inštanciu pracovného toku (viď. Obrázok 17.13, strana 2 v aplikácii)
- monitorovať pracovný tok (viď. Obrázok 17.14, strana 5 v aplikácii)
- potvrdiť, že všetky dokumenty pre ďalšie hodnotenia sú uložené v databáze (viď. Obrázok 17.15, strana 4 v aplikácii)
- získať prehľad o všetkých dopytoch (viď. Obrázok 17.17, strana 6 v aplikácii)
- získať prehľad o všetkých dokumentoch týkajúcich sa dopytov (viď. Obrázok 17.17, strana 8 v aplikácii)
- nahrať dokument do súvisiaceho stĺpca dopytu (viď. Obrázok 17.18 a 17.20, strana 8 a 9 v aplikácii)

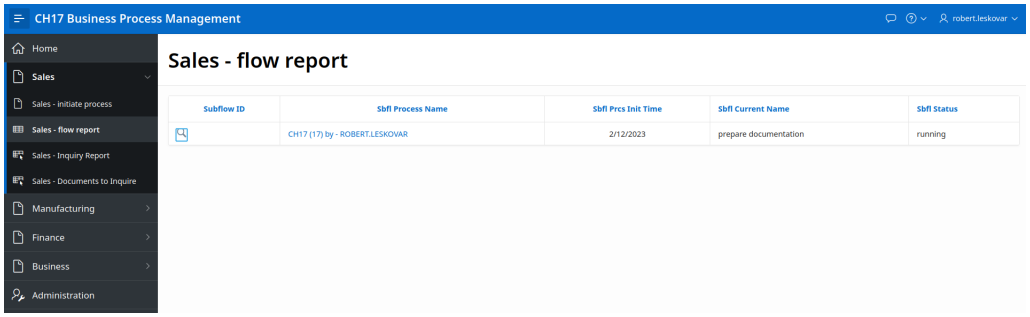
Komentáre k Obrázku 17.13 (čísla a šípky) predstavujú postupnosť akcií: 1 (používateľ zadá údaje), 2 (používateľ potvrdí zadané údaje) and 3 (zobrazí sa hlásenie o úspechu alebo zlyhaní, dátové polia sú vymazané pre nový vstup).



Obr. 17.13: Predaj - iniciovaný proces.

Okrem viditeľných komponentov je tu jedna skrytá položka stránky (P2_WID) ktorá získa ďalšiu číselnú hodnotu zo sekvencie CH17_SEQ_INQUIRY. To bol dôvod, prečo sme pred návrhom stránky zadefinovali sekvenciu CH17_SEQ_INQUIRY . Tlačidlo "Start" vykoná dva procesy: "Add inquiry" pridá jeden riadok do tabuľky CH17_INQUIRY a štyri riadky do tabuľky CH17_INQUIRY_DOC, zatiaľ čo "Create and start" proces vykoná akciu Flows for APEX doplnku (viď. podrobnosti v zbalenej aplikácii, strana 2, Procesy). Následne je zobrazená správa o úspešnom ukončení, viď. Obrázok 17.13.

Prehľad toku predaja je znázornený na Obrázku 17.14.



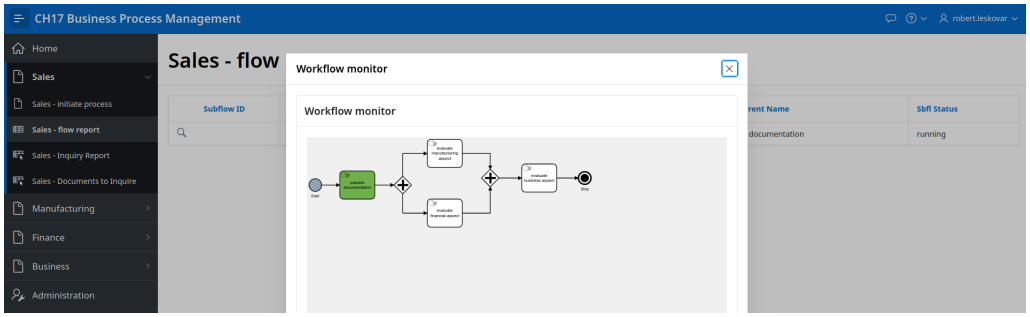
Obr. 17.14: Sales - flow report.

Prehľad toku pre predaj je klasický prehľad založený na SQL query:

```
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'
```

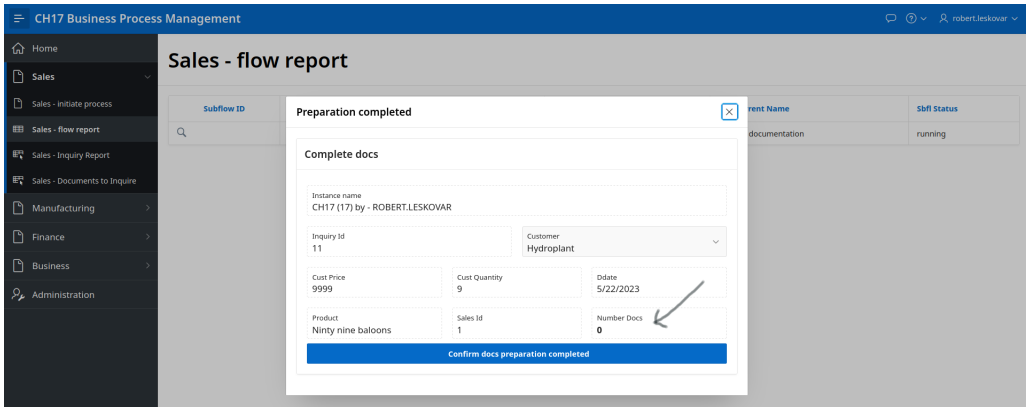
Zobrazuje iba prípady procesu "CH17", ktoré majú aktuálny názov "prepare documentation". Dva odkazy sú uvedené na strane 4 v aplikácii (viď. Obrázok 17.14):

1. ikona šošovky v stĺpci "Subflow ID" zobrazuje stav inštancie v diagrame pracovného postupu (viď. Obrázok 17.15, strana 5 v aplikácii).



Obr. 17.15: Predaj - prehľad toku pre vybranú inštanciu.

2. "Sbfl Process Name" stĺpec zobrazuje modálny dialóg v ktorom môže oddelenie predaja potvrdiť, že je dokumentácia pripravená (viď. Obrázok 17.16, strana 7 v aplikácii).



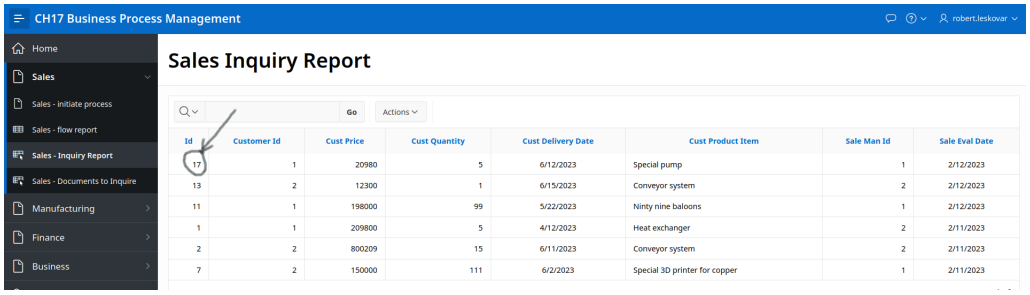
Obr. 17.16: Predaj - prehľad toku pre vybranú inštanciu.

V tomto momente ešte nie je pripravená dokumentácia (0 nahraných dokumentov). Počítadlo je výsledkom bloku PL/SQL ako zdroja:

```
BEGIN
RETURN CH17_COUNT_DOCS(:P7_INQUIRY_ID);
END;
```

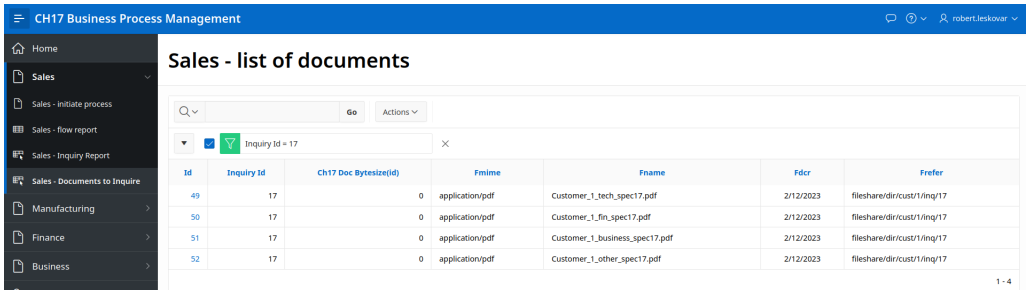
Funkcia CH17_COUNT_DOCS (podrobnosti nájdete v aplikácii Supporting objects alebo po inštalácii v Object Browser > Functions) počíta všetky veľké binárne objekty (BLOB) uložené v tabuľke CH17_INQUIRY_DOC , ktoré sú spojené s konkrétnym dopytom. Ak je veľkosť objektu BLOB väčšia ako nula, obsah sa uloží, v opačnom prípade existuje len prázdny BLOB. Podľa podnikových pravidiel musia byť pripravené aspoň tri dokumenty (pre výrobu, financie a vedenie podniku).

Na získanie prehľadu o všetkých dopytoch je pripravená interaktívna správa (viď. Obrázok 17.17, strana 6 v aplikácii).



Obr. 17.17: Predaj - prehľad dopytov.

Prehľad dopytov ponúka vyhľadávanie a rôzne druhy akcií (výber stĺpcov, usporiadanie, filtrovanie, formátovanie, vytváranie grafov, reportovanie, export, atď.). Cieľom tohto prehľadu je poskytnúť manažérovi predaja ID dopytu, ktoré možno použiť v procese prípravy dokumentácie (t.j. 17 ako je znázornené na Obrázku 17.17). Ďalším krokom pre manažéra predaja by bola príprava technickej, finančnej, obchodnej a inej špecifikácie. (viď. Obrázok 17.18, strana 8 v aplikácii).



Obr. 17.18: Predaj - zoznam dokumentov.

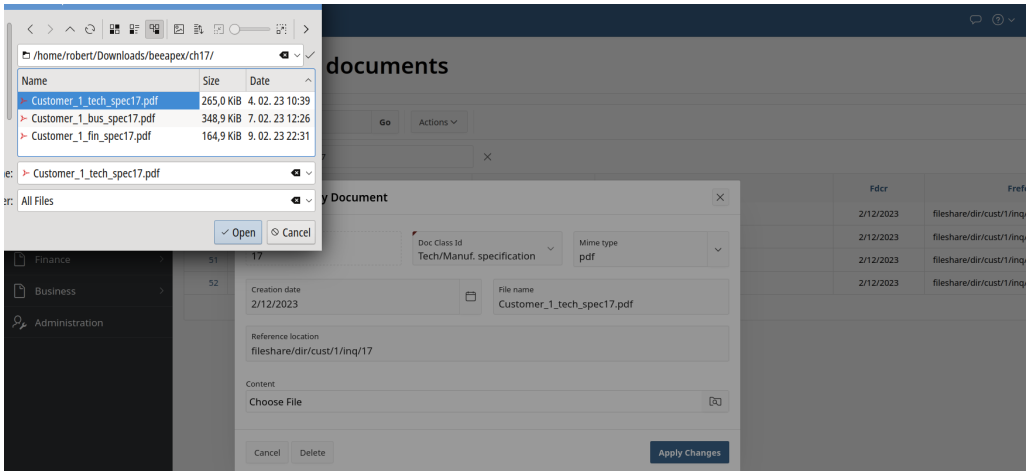
Obrázok 17.18 zobrazuje filtrované výsledky (ID dopytu je 17). Nie je nahratý žiadny BLOB - stĺpec “Ch17 Doc Bytesize(id)” zobrazuje nuly. Strana 8 je interaktívna zostava založená na nasledujúcom SQL query:

```
select ID, INQUIRY_ID,DOC_CLASS_ID, CH17_DOC_BYTESIZE(ID), CONTBLB,
FMIME,FNAME, FDCR, FREFER
from CH17_INQUIRY_DOC
```

PL/SQL funkcia CH17_DOC_BYTESIZE vracia veľkosť BLOB. Funkcia Function berie ID CH17_INQUIRY_DOC ako vstupný parameter a je definovaná takto:

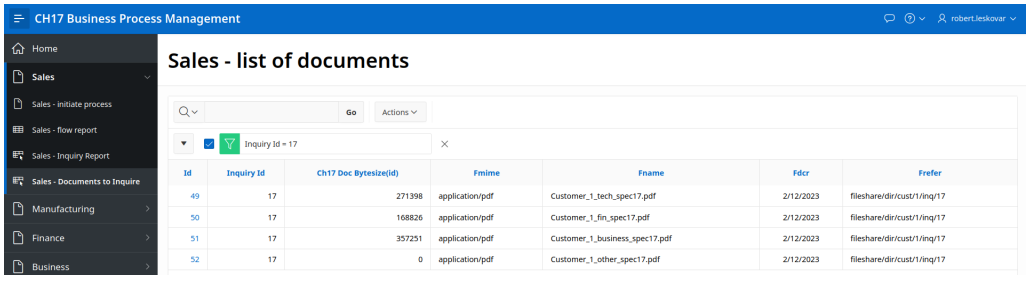
```
create or replace function CH17_DOC_BYTESIZE(xdocid in number) return number
is
wdoc_size number := 0;
wblob blob;
Begin
select contblb into wblob from ch17_inquiry_doc where id = xdocid;
wdoc_size := dbms_lob.getlength(wblob);
RETURN  wdoc_size;
END;
```

ID tabuľky CH17_INQUIRY_DOC sa používa ako odkaz na otvorenie modálnej stránky v aplikácii (viď Obrázok 17.19).



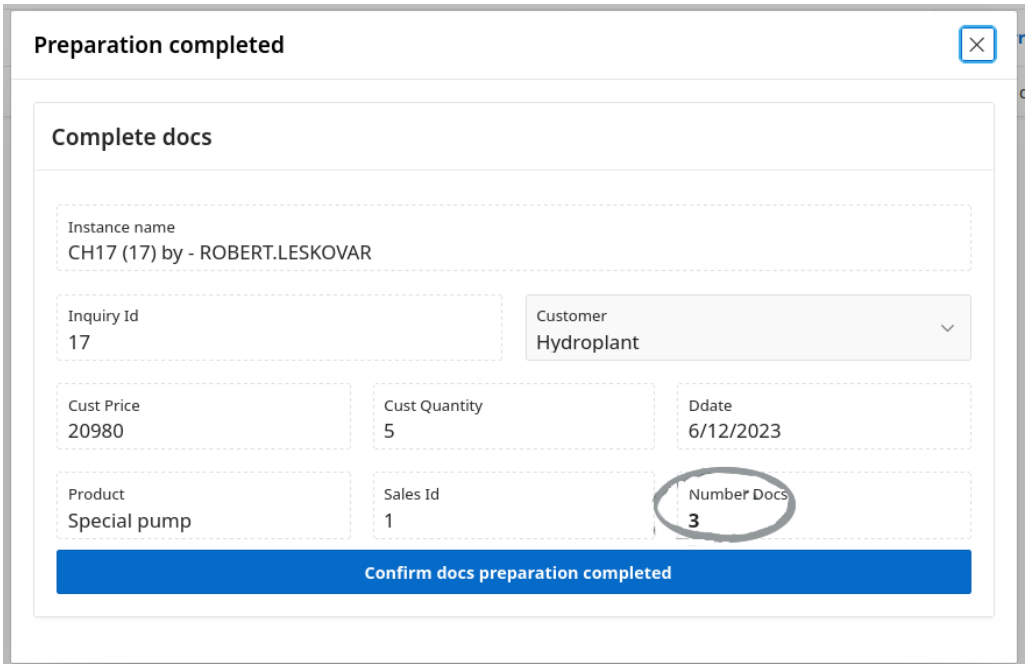
Obr. 17.19: Predaj - nahratie dokumenty pre dopyt 17.

Obnovením strany 8 si môže byť používateľ istý, že dokumenty sú nahrané (viď. Obrázok 17.20, strana 8 v aplikácii).



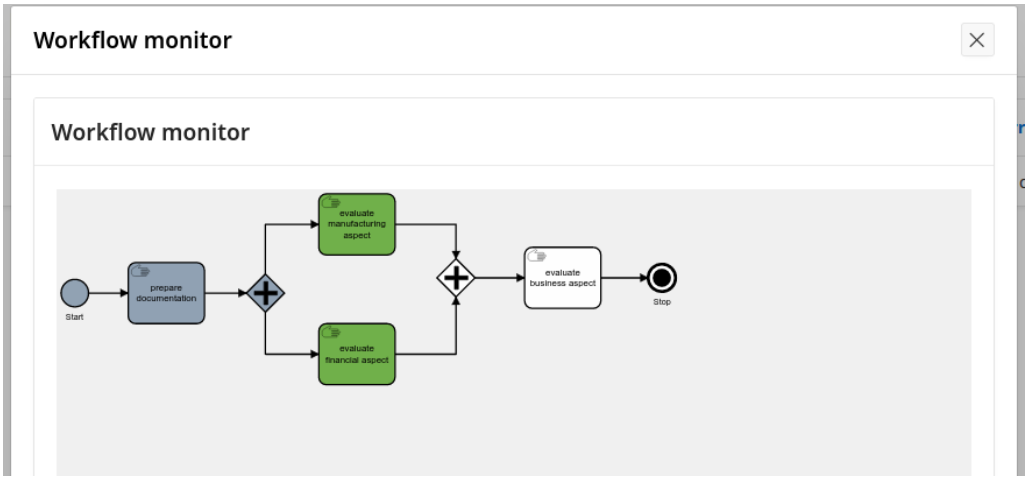
Obr. 17.20: Predaj - zoznam dokumentov po nahraní a obnovení.

Oddelenie predaja by teraz potvrdilo, že prípravná úloha je dokončená. Otvorením prehľadu toku predaja a kliknutím na odkaz na názov procesu sa otvorí modálna stránka (viď. Obrázok 17.21, strana 8 v aplikácii).



Obr. 17.21: Predaj - prehľad toku pre vybratú inštanciu po nahratí troch dokumentov.

Kliknutím na “Subflow ID”, dostane manažér predaja vizuálne potvrdenie, že príprava dokumentu je dokončená (viď. Obrázok 17.22, strana 5 v aplikácii).



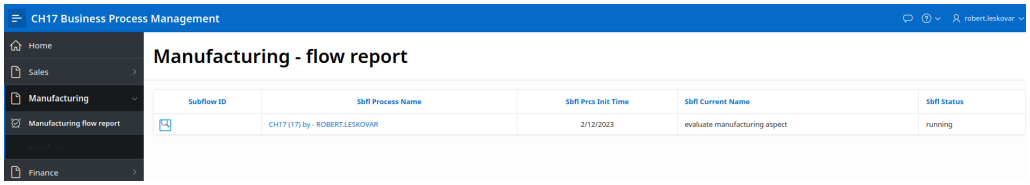
Obr. 17.22: Predaj - prehľad toku pre vybratú inštanciu po nahratí troch dokumentov.

17.6.4 Manažér výroby

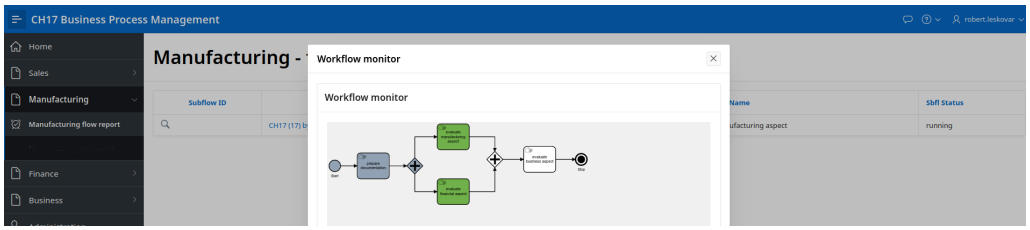
Manažér výroby môže:

- monitorovať pracovný tok (prehľad na Obrázku 17.23, strana 20 v aplikácii a diagram na Obrázku 17.24, strana 5 v aplikácii)
- enter manufacturing evaluation (Figure 17.25, Page 21 in application)

Kliknutím na “Manufacturing flow report” môže manažér výroby sledovať nevyhodnotené dopyty (t.j. prehľad na Obrázku 17.23, strana 20 v aplikácii).

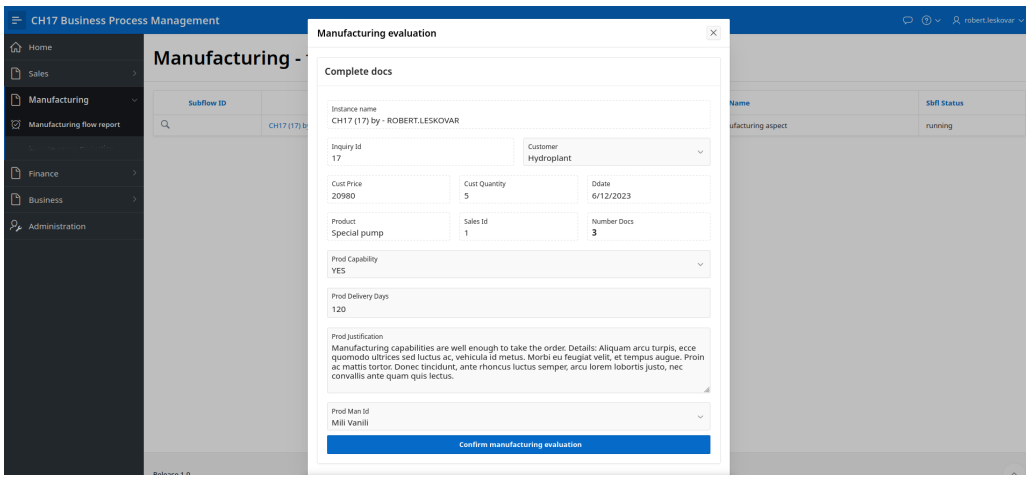


Obr. 17.23: Výroba - prehľad toku.



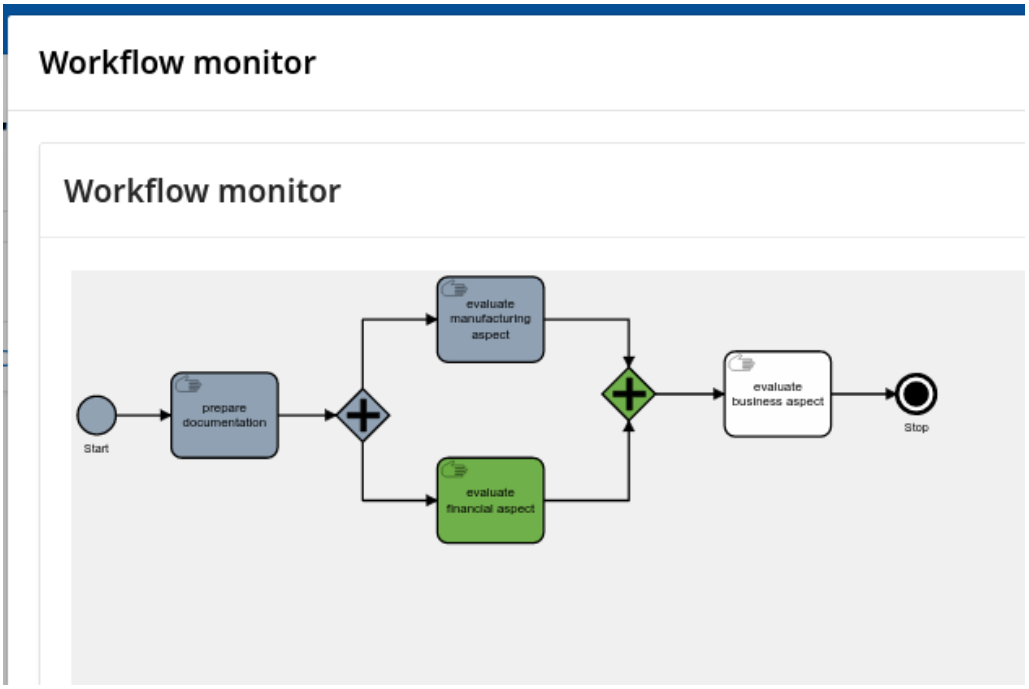
Obr. 17.24: Výroba - BPMN diagram - stav inštancie.

Výberom ikony šošovky pod “Subflow ID” sa otvorí strana 5 (viď. Obrázok 17.24), ktorá zobrazuje inštanciu toku CH17. Inštancia je v paralelnej bráne, čo znamená, že obe úlohy (výrobné a finančné hodnotenie) musia byť ukončené pred spustením ďalšej úlohy (hodnotenia na najvyššej úrovni riadenia). Hodnotenie výroby môže byť zadané kliknutím na odkaz “Sbfl Process Name” na strane 20 (viď. Obrázok 17.25, strana 21 v aplikácii).



Obr. 17.25: Hodnotenie z výrobného hľadiska.

Stlačením tlačidla “Confirm manufacturing evaluation” sa tok inštancie presunie do ďalšieho stavu znázorneného na obrázku reffig:ch17ENguiman04.



Obr. 17.26: Výrobné hodnotenie ukončené, inštancia čaká na finančné hodnotenie.

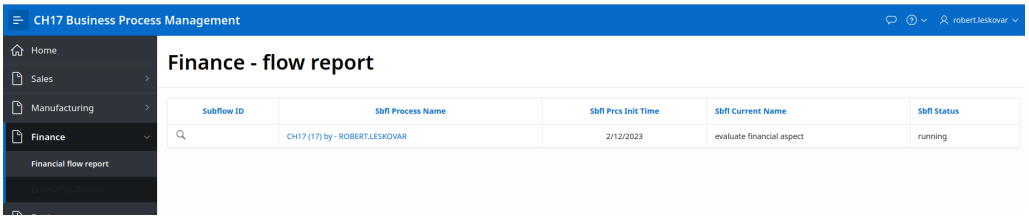
Po obnovení strany 20, sa v zostave zobrazí “Manufacturing has no flow instances”.

17.6.5 Finančný manažér

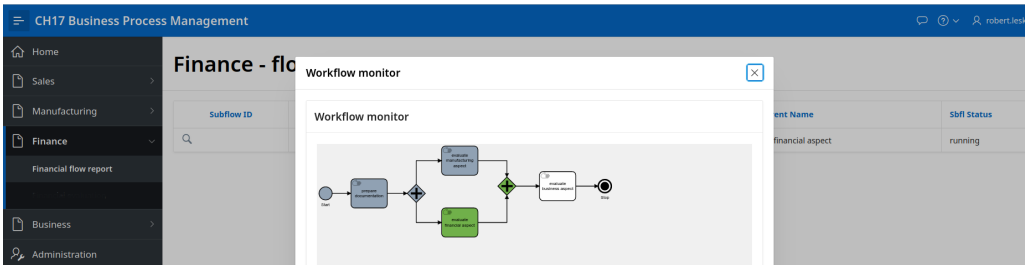
Finančný manažér môže:

- monitorovať pracovný tok (prehľad na Obrázku 17.27, strana 30 v aplikácii a diagram na Obrázku 17.28, strana 5 v aplikácii)
- zadať finančné hodnotenie (Obrázok 17.29, strana 31 v aplikácii)

Kliknutím na “Financial flow report” môže finančný manažér sledovať nevyhodnotené požiadavky (t.j. prehľad na Obrázku 17.27, strana 30 v aplikácii).

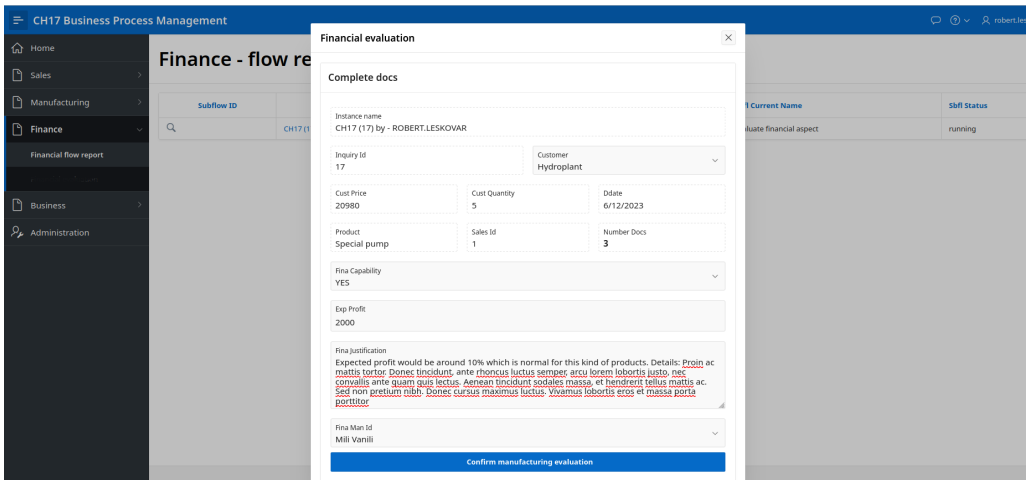


Obr. 17.27: Financie - prehľad toku.



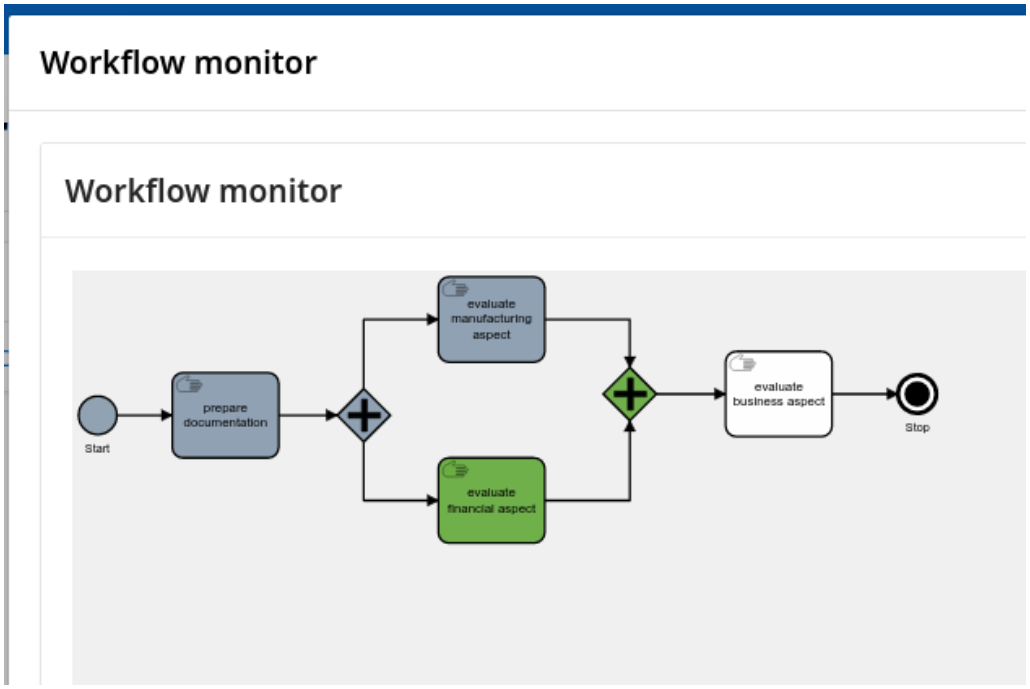
Obr. 17.28: Financie - BPMN diagram - stav inštancie.

Kliknutím na ikonu šošovky pod “Subflow ID” sa otvorí aplikácia na strane 5 (viď. Obrázok 17.28), ktorá zobrazuje inštanciu toku CH17. Finančné hodnotenie je možné zadať kliknutím na odkaz “Sbfl Process Name” na strane 30 (viď. Obrázok 17.29, strana 31 v aplikácii).



Obr. 17.29: Finančné hodnotenie.

Stlačením tlačidla “Confirm financial evaluation” sa tok inštancie posunie do ďalšieho stavu znázorneného na Obrázku 17.30.



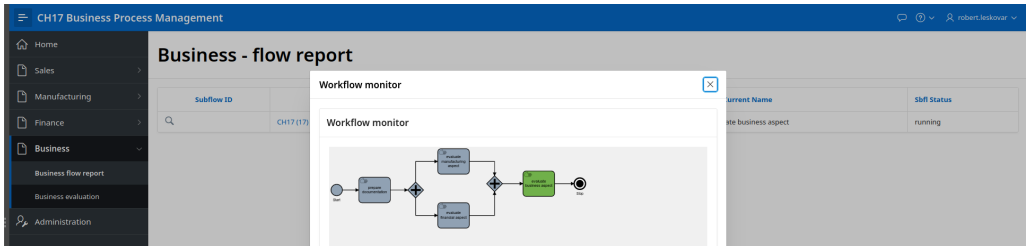
Obr. 17.30: Finančné hodnotenie ukončené, inštancia čaká na hodnotenie z podnikateľského hľadiska.

Po obnovení stránky 30 sa v zostave zobrazí “Finance has no flow instances”.

17.6.6 Generálny riaditeľ - manažér podniku

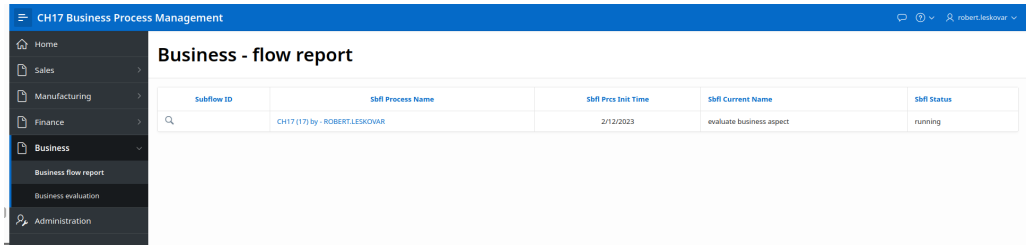
Manažér podniku môže:

- monitorovať pracovný tok (prehľad na Obrázku 17.32, strana 40 v aplikácii a diagram na Obrázku 17.31, strana 5 v aplikácii)
- zadať hodnotenie z podnikového hľadiska (Obrázok 17.33, strana 41 v aplikácii)



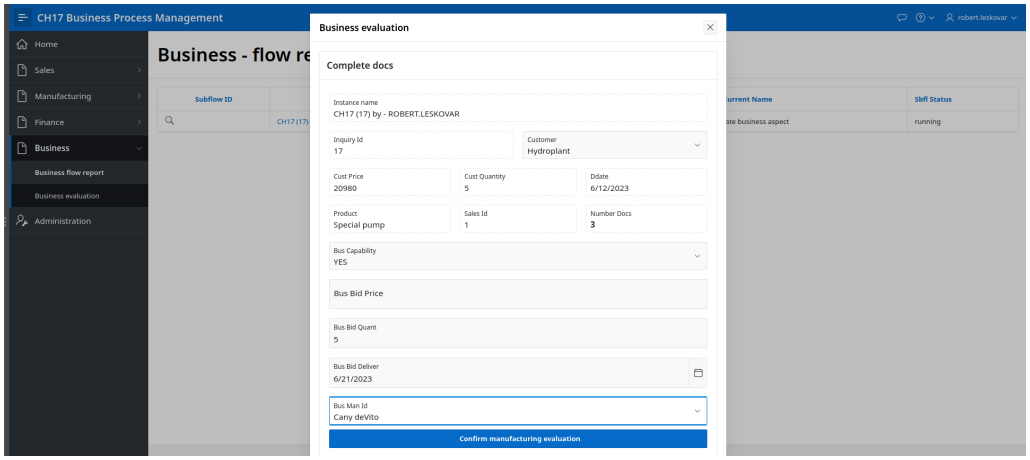
Obr. 17.31: Podnikové hodnotenie - BPMN diagram - stav inštancie.

Kliknutím na "Business flow report" môže generálny riaditeľ sledovať nevyhodnotené dopyty (t.j. prehľad na Obrázku 17.32, strana 40 v aplikácii).



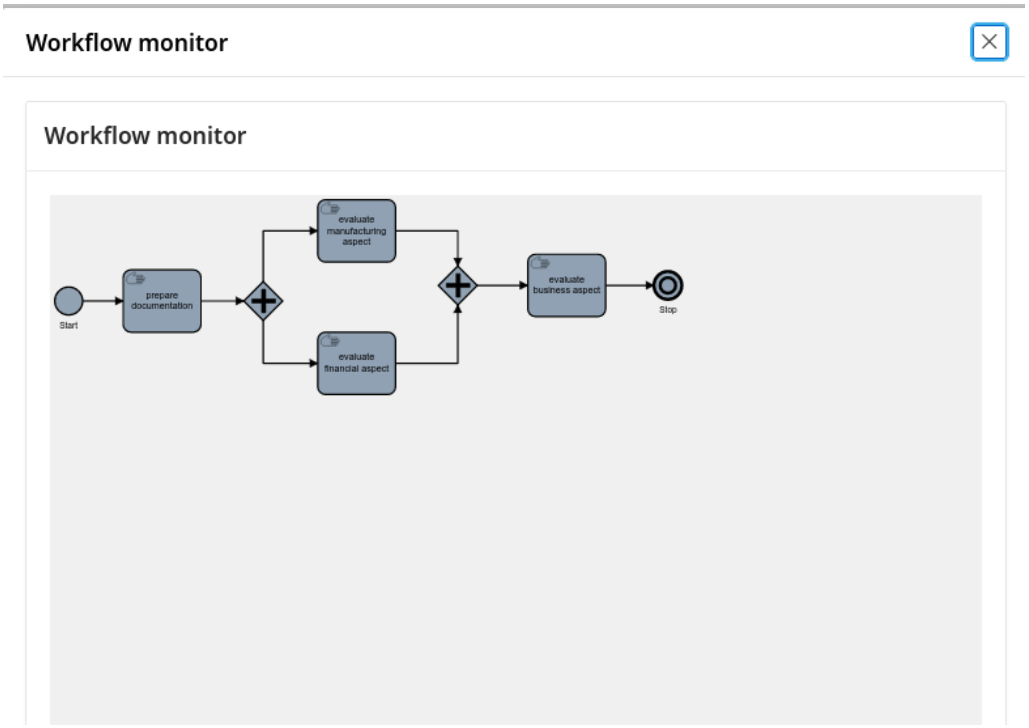
Obr. 17.32: Podnikové hodnotenie - prehľad toku.

Kliknutím na ikonu šošovky pod "Subflow ID" sa otvorí stránka 5 (viď. Obrázok 17.31), ktorá zobrazuje inštanciu toku CH17. Hodnotenie na najvyššej úrovni je možné zadať kliknutím na odkaz "Sbfl Process Name" na strane 40 (viď. Obrázok 17.33, Page 41 in application).



Obr. 17.33: Hodnotenie na najvyššej úrovni.

Stlačením tlačidla "Confirm business evaluation" sa inštancia posunie na koniec pracovného toku, čo je znázornené na Obrázku 17.34.



Obr. 17.34: Hodnotenie na najvyššej úrovni skončilo, inštancia ukončená.

Obnovením strany 40 sa v zostave zobrazí “Business has no flow instances”.

17.7 Prepojenie aplikácie s Flows for APEX

Aplikácia vyvinutá v Kapitole 17 aplikovala dve funkcie poskytované Flows for APEX:

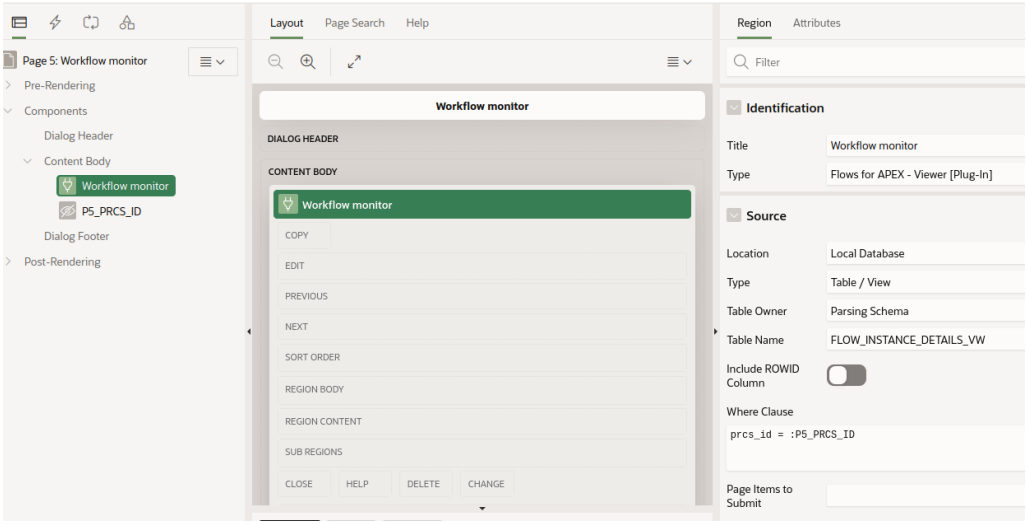
- spustiť inštanciu na strane 2,
- zobraziť stav inštancie na strane 5
- posúvať inštanciu dopredu na ďalšie úlohy na stranách 7, 21, 31, 41

Spustenie inštancie si vyžaduje existenciu procesu na stránke v dentifikačnej oblasti typu procesu: “Flows for APEX - Managing Flow Instance [Plug-In]”. Akcia by mala byť “Create and Start”. Názov toku inštancie sme nastavili na:

```
CH17 (&P2_WID.) by - &APP_USER.
```

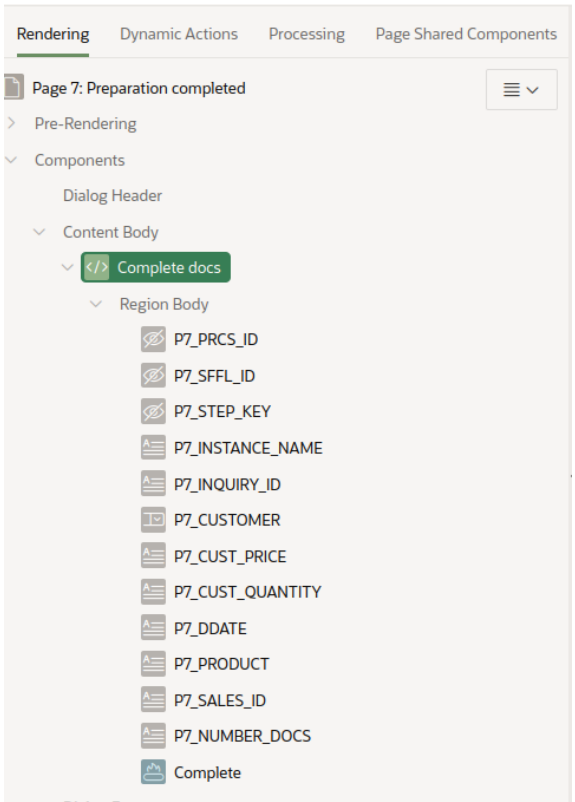
Pamätajte, že položka stránky P2_WID obsahuje jedinečné číslo poskytnuté sekvenciou. Názov je teda zreťazený reťazec zložený z fixného reťazca “CH17”, medzery, ľavej zátvorky, jedinečného poradového čísla, pravej zátvorky, medzery fixného reťazca “by - ” a aktuálneho užívateľa. Select Flow používa “Static text” a statický text je “CH17”. Potvrdzujúca správa is “Flow CH17 instance started.” a chybová správa je “Flow CH17 instance NOT started.”

Na zobrazenie stavu inštancie na strane 5 sa používa sprievodca pre plug-in. Stránka obsahuje plug-in prehliadača a skrytú položku - P5_PRCS_ID. Viď nastavenia na Obrázku 17.35.



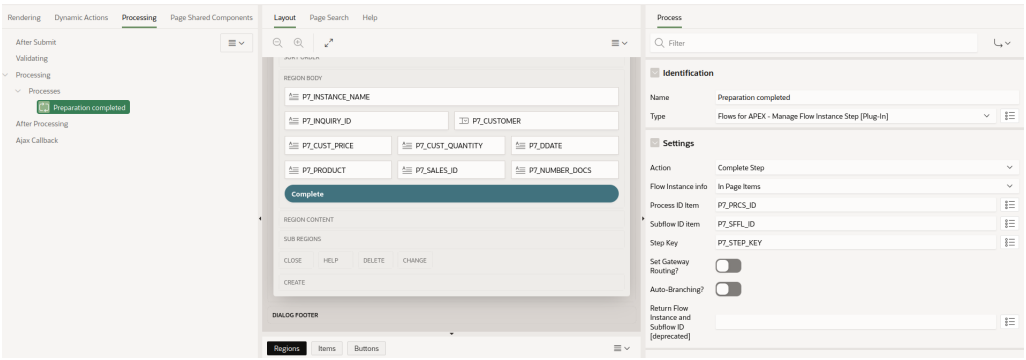
Obr. 17.35: Zobrazenie stavu inštancie na diagrame BPMN.

Ak chcete posunúť inštanciu dopredu (t.j. stránky 7, 21, 31, 41), definujte položky stránky ako je znázornené na Obrázku 17.36.



Obr. 17.36: Nastavenie položiek stránky.

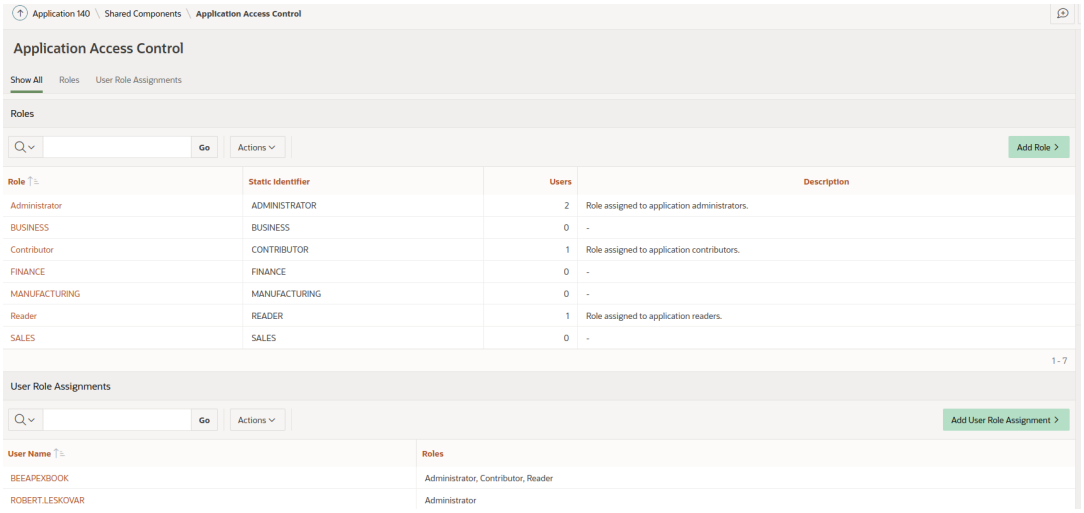
Iba prvé štyri položky stránky (from P7_PRCS_ID to P7_INSTANCE_NAME) sú relevantné pre integráciu Flows for APEX. Vytvorte nový proces, ktorý sa spustí tlačidlom. Proces je definovaný ako doplnok “Manage Flow Instance Step”. Nastavte Action, Process ID Item, Subflow ID item a Step Key ako je znázornené na Obrázku 17.37.



Obr. 17.37: Dokončenie kroku vo Flows for Apex.

17.8 Definovanie užívateľských rolí

Používateľské roly je možné definovať v Application Access Control v Shared Components (Obrázok 17.38).



Obr. 17.38: Definujte roly a užívateľské roly v Application Access Control menu.

V tomto bode preskočíme podrobnosti a implementáciu pre tento konkrétny prípad. Pre rýchly prehľad si môžete pozrieť Kapitolu 13 alebo pre hlbšie pochopenie problematiky dokumentáciu k APEX autorizácii.

17.9 Testovanie a oprava chýb

Pre začiatočníkov by bol proces testovania veľmi zjednodušený. Ak položíte ruky na klávesnicu a myš za účelom vývoja pomocou APEX, je takmer isté, že spravíte chybu ale aj to, že ju opravíte. Nebojte sa robiť chyby. Vezmite si ponaučenie z každej úlohy, ktorú stse sami vyriešili. Podeľte sa o svoje znalosti v otvorenej APEX komunite. Testovanie softvéru je síce obrovskou disciplínou, ale je možné sa ho naučiť tak, že ho realizujete. Neexistuje žiaden iný spôsob. Vývojári sú prvou líniou v obrane proti nedostatkom softvéru, chybám a zlyhaniam.

Prekvapivo neúspešné projekty vývoja softvéru vo väčšine prípadov zlyhali skôr, ako bol napísaný jediný riadok kódu. Aplikácia vyvinutá pre túto kapitolu má tiež softvérové chyby. Niektoré z nich šikovní čitatelia objavia a odladia. Niektoré chyby sú aj hlboko v samotnom APEX. My, autori a vývojári tejto kapitoly dúfame, že testovanie a ladenie bude pre čitateľov príjemnou výzvou.

Odstraňovanie chýb zo softvéru je niekedy zábava a niekedy prekliatie. Len sa netreba vzdávať príliš skoro!

17.10 Doplnkový učebný materiál

Medzi doplnkovým učebným materiálov môžete nájsť:

- exportované zabalené aplikácie a skripty
- model pracovného toku s názvom CH17
- video návody

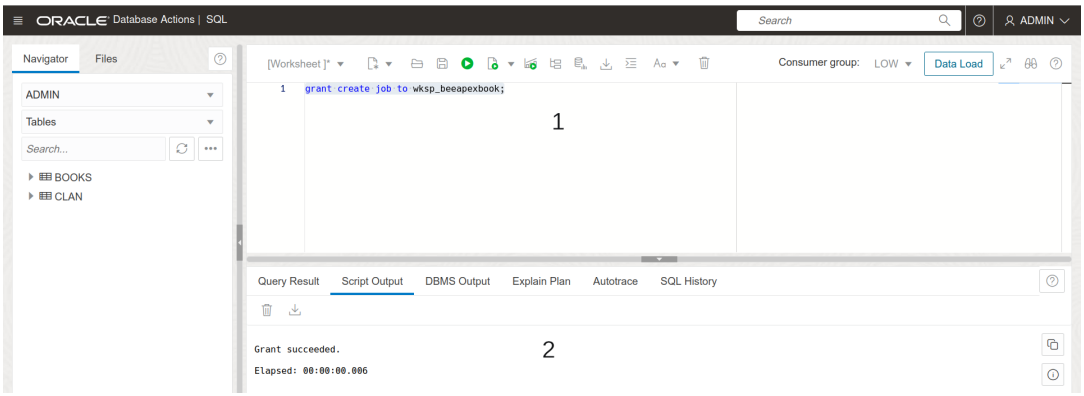
Všetky doplnkové učebné materiály sú dostupné na verejná BeeAPEX projektová stránka. Prihláste sa ako hosť (nevyžaduje sa žiadne heslo). Učebnicu nájdete v sekcii Knihy (Books), skriptá v priečinku Časť 2 (Part 2) > Kapitola17 (Chapter17) v sekcii Skriptá (Scripts) a video návody v Zbierke video návodov (Collection of video guides). Materiál pre krátke kurzy je v sekcii Krátke kurzy (Short courses).

17.10.1 Exportovaná aplikácia

Exportovaná aplikácia je zabalená. Inštalácia vytvára tabuľky, sekvencie, funkcie a tiež napĺňa dáta. Odinštalovaním sa odstránia všetky objekty databázy používané v tejto aplikácii.

Zbalená aplikácia je testovaná a bude spustená v novom pracovnom priestore (workspace), ak budú splnené nasledujúce požiadavky:

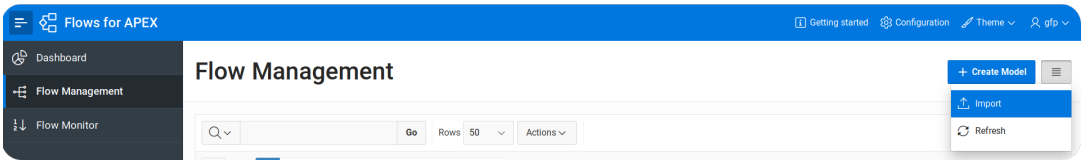
- oprávnenie "create job" je udelené schéme, ktorá obsahuje nový workspace. Administrátor databázy môže udeliť toto oprávnenie príkazom "grant create job to <schema>". APEX workspace je spojený s databázovým objektom s názvom schéma. Ak používate OCI (bezplatné, platené alebo poskytované Oracle Academy) tak použite webové OCI rozhranie (Autonomous Database > specific instance > Database action) alebo vstúpte priamo do webového rozhrania správcu databázy. Tak môžete vykonávať požiadavky a skripty. Obrázok 17.39 zobrazuje pohľad na rozhranie so zadaným príkazom (1) an odpoveďou databázy (2).



Obr. 17.39: Udelenie "create job privilege" workspace-u.

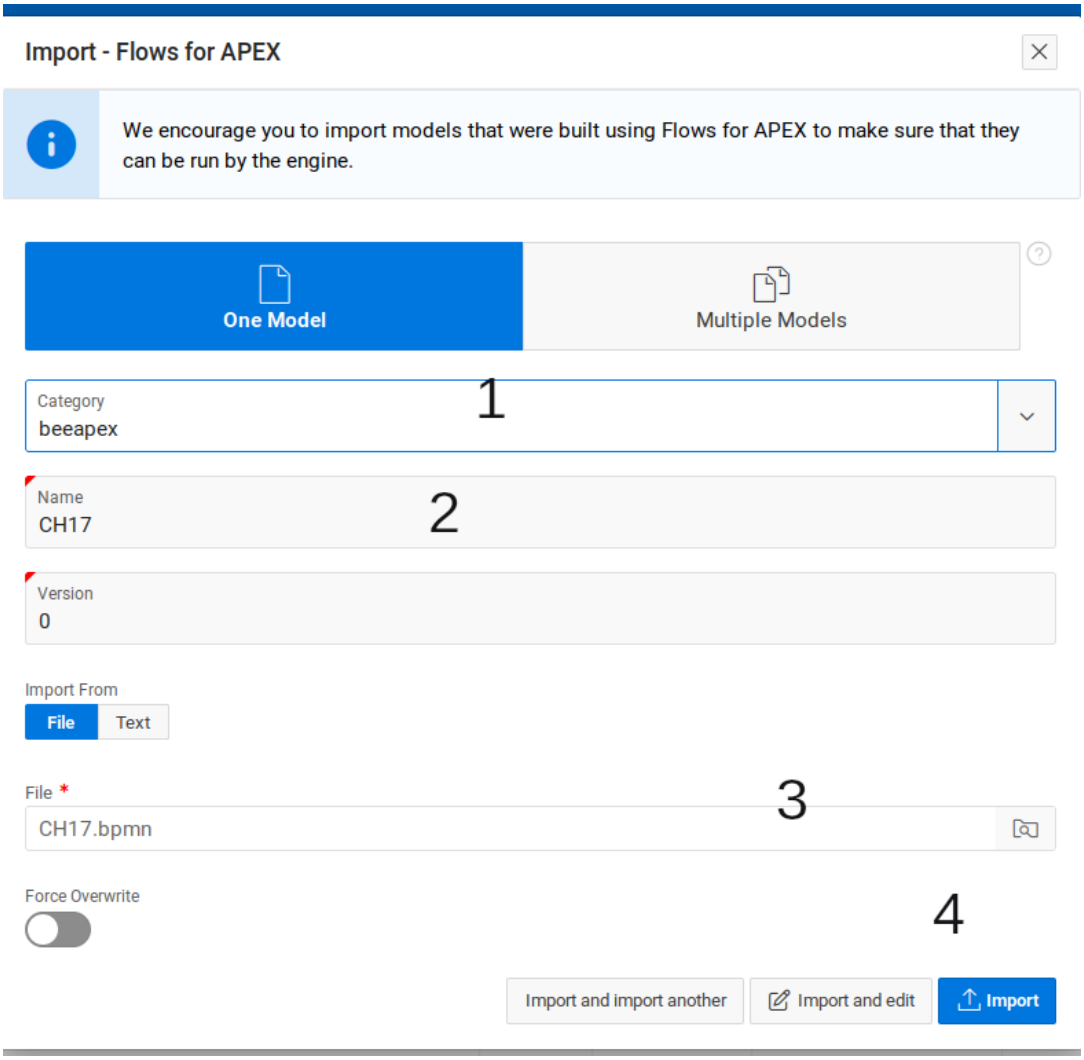
Ak používate iné nástroje (SQL Developer, TOAD), postup je takmer rovnaký, s výnimkou vzhľadu rozhrania.

- Flows for APEX musí byť nainštalovaný v novom workspace (viď. predchádzajúce pokyny).
- Workflow model s názvom CH17 je importovaný do Flows for APEX. Súbor *CH17.bpmn* je uložený v adresári Flows4APEX v zbalenej aplikácii ako statický súbor. Prejdite do časti Components > Static Application Files a stiahnite si ich do lokálneho počítača. Rovnaký súbor je dostupný aj v skriptoch učebných materiálov. Potom importujte súbor *CH17.bpmn* pomocou Flows for APEX. Prihláste sa do aplikácie Flows for APEX. Vyberte Flow management a importujte súbor (viď. obrázky 17.40 - krok 1 a Obrázok 17.41 - krok 2).



Obr. 17.40: Import pracovného toku s názvom CH17 do Flows for APEX - krok 1.

Význam čísel na Obrázku 17.41 je: 1 (ľubovoľná kategória), 2 (názov modelu musí byť presne CH17), 3 (prejdite k prieskumníkovi súborov na CH17.bpmn) a 4 (potvrďte stlačením tlačidla Import)



Obr. 17.41: Import pracovného toku s názvom CH17 do Flows for APEX - krok 2.

- Po inštalácii zbalenej aplikácie pridajte priradenie používateľskej role (Shared components > Application Access Control > Add User Role Assignment)

Ak niektorá z vyššie uvedených požiadaviek nie je splnená, importovaná aplikácia zlyhá. Po zlyhaní aplikácie z dôvodu nesplnených požiadaviek je potrebné vymazať súbor cookie webového prehliadača (i.e. Firefox: Settings > Cookies and Site Data > Manage Data) po zlyhaní aplikácie z dôvodu nesplnených požiadaviek.

17.10.2 Video návody

Video návod ukazuje počiatočné kroky pri vývoji aplikácie.

17.11 Otázky

1. Vysvetlitl, čo je Flows for APEX?
2. Aké sú artefakty modelu podnikových procesov prezentované pomocou BPMN diagramu?
3. Ktoré funkcie od Flows for APEX boli použité v tejto aplikácii?

17.12 Odpovede

1. Flows for APEX je považovaná za funkciu, doplnok a aplikáciu. Všetky tri významy sú správne. Funkcia je myslená vtedy, keď diskutujeme v kontexte procesného modelovania. Plug-in sa týka vývoja vlastnej aplikácie. Aplikácia "Flows for APEX" je zameraná na návrh pracovných postupov a správu inštancií v prípade problémov.
2. Artefakty modelu podnikového procesu prezentované s BPMN diagramom sú:
   - Počiatočná udalosť: Označuje začiatok procesu.
   - Úloha: Predstavuje jednu jednotku práce, napríklad aktivitu alebo krok v procese.
   - Brána: Predstavuje rozhodovací bod v procese, ako je napríklad rozvetvenie toku alebo podmienená vetva.
   - Sekvenčný tok: Spája prvky v diagrame a predstavuje tok riadenia z jedného prvku do druhého.
   - Konečná udalosť: Indikuje koniec procesu.
   - Bazén/Dráha: Predstavuje zoskupenie súvisiacich úloh používané na definovanie rolí a zodpovedností rôznych účastníkov procesu.
   - Tok správ: Predstavuje tok správ medzi účastníkmi procesu, ako napríklad komunikácia medzi dvomi systémami.
   - Dátové objekty:Predstavuje časť údajov, ktorá sa používa alebo vytvára v procese, ako je napríklad faktúra alebo záznam o zákazníkovi.
   - Úložisko dát: Predstavuje kontajner na ukladanie dát, ktoré sa používajú alebo vytvárajú v procese, napr. databáza alebo súborový systém.
3. Na spustenie inštancie bola použitá akcia "Create and Start". Na prezentáciu inštancie na diagrame BPMB bol použitý "Viewer component". Na posun vpred k ďalšiemu kroku bol v aplikácii použitý komponent "manage Flow Instance Step".



18. GreenDi – Výmena rastlín a semien

VJERAN STRAHONJA

18.1 Biznis pohľad prípadu

Stručný biznis prehľad GreenDi platformy je popísaný v Kapitole 15. Kapitola 18 sa týka časti platformy venovanej výmene rastlín a semien. Všetci registrovaní používatelia platformy sa môžu zúčastniť tejto výmeny a predkladať svoje vlastné ponuky alebo reagovať na ponuky iných ľudí. Systém by mal zároveň umožniť vyhľadávanie ponúk podľa rôznych filtrov, triedenie stĺpcov a podobných funkcionalít. Okrem toho by mala byť umožnená komunikácia medzi uchádzačom a ostatnými používateľmi súvisiacimi s konkrétnou ponukou. Komunikácia prebieha vo forme četu.

18.2 Definícia problému

Nasledujúce vety popisujú problém:

- Základné funkcie výmeny semien a rastlín sú predkladanie vlastných ponúk a reagovanie na ponuky iných ľudí, čo je vyhradené pre členov.
- Členovia môžu vyhľadávať v katalógu podľa rôznych kritérií a otvoriť si tabuľku s otvorenými ponukami pre každý druh rastliny z katalógu.
- Platforma nie je komerčná, t. j. rastliny sa na platforme nepredávajú ani nekupujú, ale vymieňajú sa a darujú. Preto obchody s rastlinami nepodliehajú zdaneniu.
- Ponuka obsahuje rastlinu, ktorá sa ponúka alebo žiada, množstvo, miesto, záujemcu, komentáre, dátum ponuky, dátum platnosti ponuky, pokyny pre vyzdvihnutie alebo odoslanie, stav atď.
- Ponuky sú prezentované vo forme tabuľky, s možnosťou triedenia podľa stĺpcov, vyhľadávania a filtrovania podľa rôznych kritérií (typ rastliny, dátum ponuky, člen, miesto, stav atď.).
- Každý člen môže reagovať na akúkoľvek otvorenú ponuku.
- Konverzácia k akejkoľvek ponuke má formu verejného alebo súkromného četu.
- Základnou jednotkou pre konverzáciu (čet) je správa.
- Správa obsahuje informácie o členovi, ktorý ju posiela (vygeneruje sa automaticky), text správy a viditeľnosť (verejná, súkromná).
- Systém by mal podporovať všetky fázy správy životného cyklu ponúk (vytvárať, meniť, uzatvárať, odstraňovať a archivovať). Ponuku vytvára, upravuje, uzatvára, odstraňuje a archivuje vlastník ponuky. Administrátor môže odstrániť ponuku, ak porušuje niektoré

pravidlá a môže po určitom čase archivovať neaktívne ponuky.

18.3 Prípady použitia

18.3.1 Naratívny popis

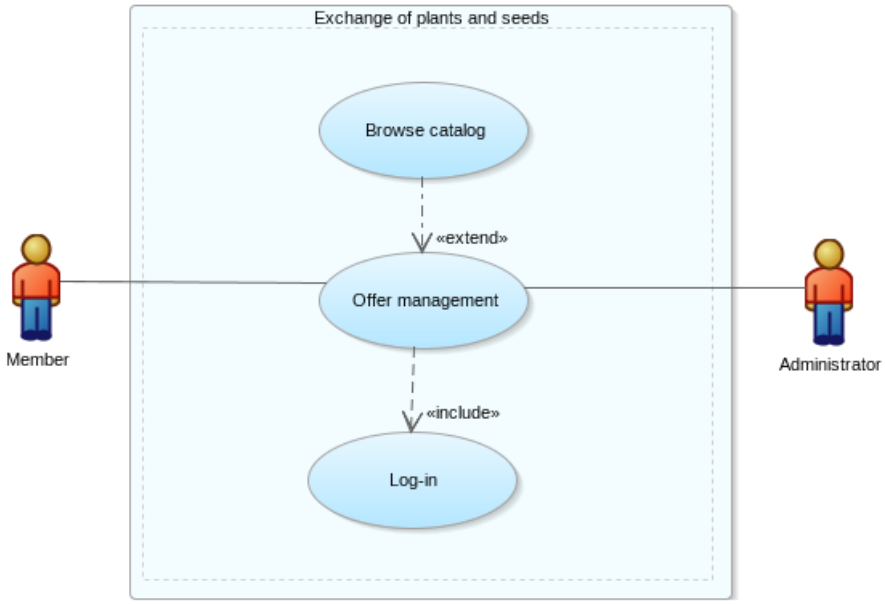
Komunikácia s prípadom použitia Offer management (správa ponúk) je dostupná každému registrovanému používateľovi alebo členovi platformy GreenDi. Predpokladom pre komunikáciu je úspešné prihlásenie (prípad použitia Log). Funkcionalita prípadu použitia Browse catalog of plant (prehliadania katalógu rastlín) je rozšírením prípadu použitia Offer management. Administrátor je tiež externým používateľom prípadu použitia Offer management. Zároveň si môže prezerať všetky ponuky a čety bez obmedzení a môže meniť stav každej ponuky.

18.3.2 Pološtruktúrovaný popis

Tabuľka 18.1 predstavuje prípady použitia.

18.3.3 Diagram prípadov použitia

Vyššie uvedený príbeh je zobrazený na diagrame prípadov použitia 18.1.



Obr. 18.1: Diagram prípadov použitia - Výmena rastlín a semien

18.4 Dátový model

Táto časť vysvetľuje dátový model.

18.4.1 Naratívny popis dátového modelu

Logický dátový model (entitno-relačný model) je v skutočnosti vylepšením dátového modelu z Kapitol 14 a 15. Entitné typy Plant (Rastlina) a Thematic unit (Tematická jednotka) boli prevzaté z Kapitoly 14, a User (Používateľ) z Kapitoly 15. Základné entity pre výmenu rastlín a semien sú Offer (Ponuka) a Message (Správa). Entita Offer je popísaná nasledujúcimi atribútmi: ID offer (identifikačný atribút), Bid or ask (ponuka alebo dopyt - určuje, či sa rastlina alebo semeno ponúka alebo žiada), Quantity (množstvo rastliny alebo semena), Location (miesto, kde je možné

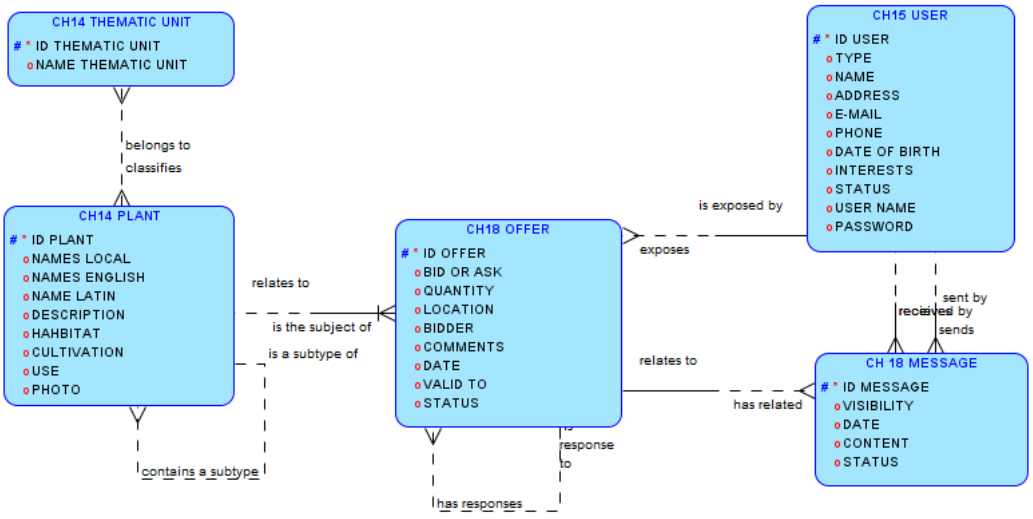
Tabuľka 18.1: Popis prípadu použitia: Výmena rastlín a semien.

Kľúčové slovo	**Hodnota**
ID:	*ch18-01*
Názov:	*Výmena rastlín a semien*
Popis:	*Výmena rastlín.*
Hlavný aktér:	*Akákoľvek osoba*
Predpoklady:	*Základné informácie o rastline sú verejne prístupné pre vyhľadávanie a prezeranie bez registrácie.*
Podmienky po skončení:	*Ak je vyvolané prehliadanie katalógu rastlín z iného prípadu použitia (extend), prenesie sa ID vybranej rastliny.*
Hlavné	*Scenáre*
Úspešný scenár:	1. používateľ prechádza hore a dolu v katalógu, kým nenájde rastlinu alebo nepoužije vyhľadávanie pomocou filtra 2. používateľ otvorí zoznam ponúk pre vybranú rastlinu alebo všetky rastliny 3. používateľ prechádza hore a dolu v zozname ponúk alebo použije vyhľadávanie pomocou filtra 4. používateľ otvorí novú ponuku pre vybranú rastlinu, zadá údaje a zverejní novú ponuku, 5. alebo klikne na vybratú ponuku, otvorí sa vyskakovacie okno s detailom vybranej ponuky. 6. používateľ môže odpovedať na akúkoľvek aktívnu ponuku protiponukou alebo prijatím, alebo môže začať čet. 7. čet je vo forme správ a odpovedí na správy 8. spojenie je viditeľné pre ponúkajúceho a inú osobu zúčastňujúcu sa na konverzácii
Rozšírenia:	• *Zoznam ponúk (vo forme tabuľky)* • *Detaily ponuky (nová alebo existujúca)* • *Čet (zoznam správ)*
Frekvencia použitia:	*Približný maximálny počet za deň je 100.*
Status:	*Vo vývoji*
Vlastník:	*verejnosť, anonymný používateľ*
Priorita:	*Vysoká*

rastlinu alebo semeno prevziať alebo doručiť), Bidder (uchádzač - používateľ, ktorý ponúka alebo žiada), Comments (komentáre - pripomienky ku kvalite, inštrukcie pre doručenie atď.), Date (dátum ponuky), Valid to (platnosť do - dátum platnosti ponuky), Status (stav - aktívna, stiahnutá, zrealizovaná). Ponuka môže byť pôvodnou ponukou alebo odpoveďou na pôvodnú ponuku. To znamená, že každá ponuka je odpoveďou na 0 alebo jednu ponuku a SÚČASNE má 0, 1 alebo viac odpovedí. Každá ponuka (Offer) je ponúkaná jedným a len jedným používateľom (User). K ponuke (Offer) je priradených 0, 1 alebo viac správ (Messages). Každá správa (Message) je priradená k 1 a len 1 ponuke (Offer), odoslaná 1 a len 1 používateľom (User) - odosielateľom, a určená 1 a len 1 používateľovi (User) - prijímateľovi.

18.4.2 Logický dátový model

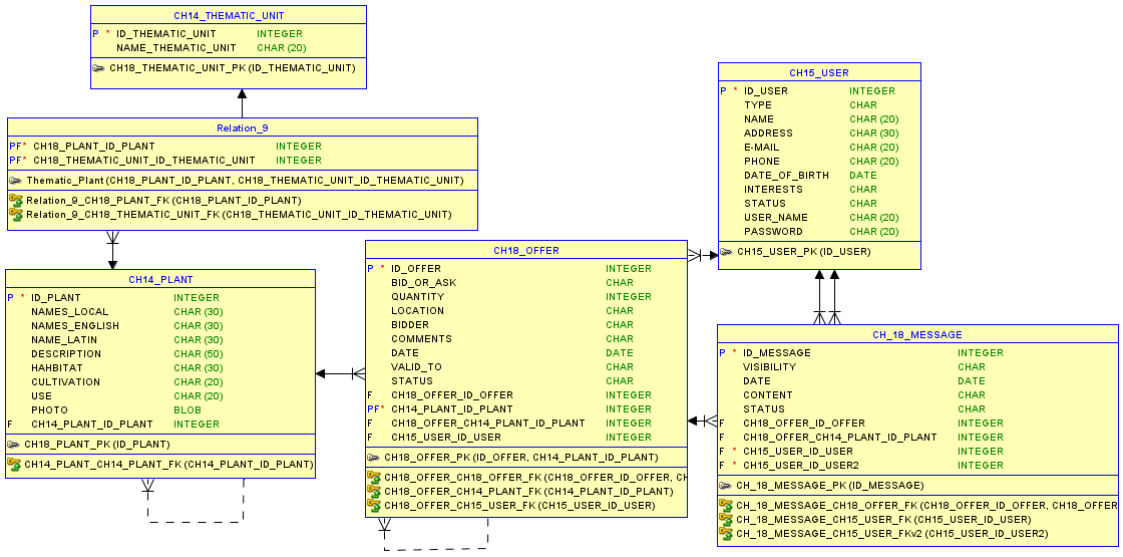
Logický dátový model je zobrazený na Obrázku 18.2.



Obr. 18.2: Logický dátový model.

18.4.3 Relačný dátový model

Automatickú transformáciu z logického dátového modelu na relačný dátový model v Oracle SQL Data Modeler zabezpečuje funkcia *Engineering to relational*. Výsledok je zobrazený na Obrázku 18.3.

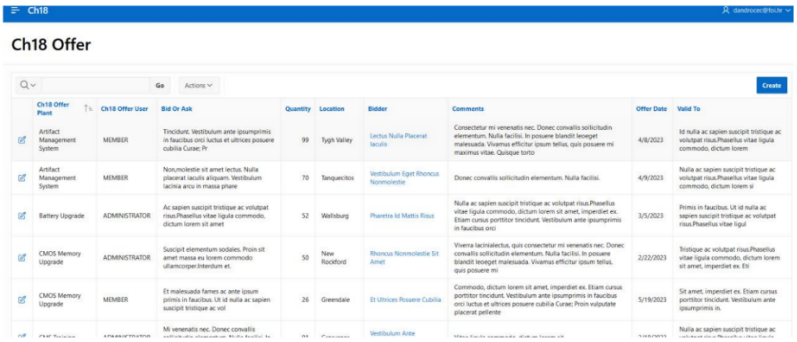


Obr. 18.3: Relačný dátový model.

Nechajme Oracle SQL Data Modeler vygenerovať SQL skript. Vyberte všetky tabuľky relačného modelu a použite funkciu *File* > *Export* > *DDL File* na získanie skriptu. Vygenerovaný skript môžeme importovať do APEX-u a spustiť ho.

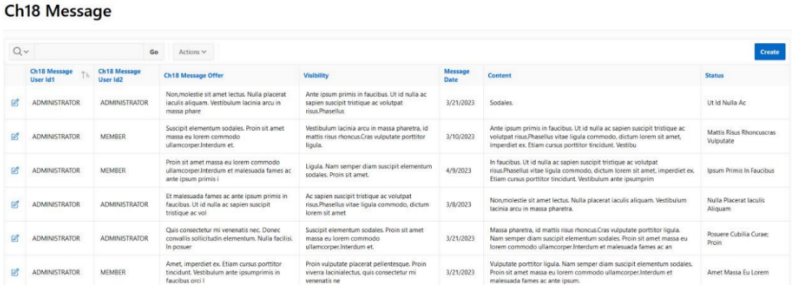
18.5 Rozhrania aplikácie

Návrh formulára pre ponuku (Offer) je zobrazený na Obrázku 18.4.



Obr. 18.4: Ponuka (Offer).

Návrh formulára pre správu (Message) je zobrazený na Obrázku 18.5.



Obr. 18.5: Správa (Message).

18.6 Doplnkový učebný materiál

Môžete nájsť nasledujúci doplnkový učebný materiál:

- exportovaná aplikácia,
- skripty skript na vytvorenie, zrušenie a naplnenie tabuliek,
- video sprievodcu.

Všetky doplnkové učebné materiály sú dostupné na verejnej stránke projektu BeeAPEX. Prihláste sa ako hosť (nevyžaduje sa žiadne heslo). Učebnicu nájdete v sekcii Books (Knihy), skripty v priečinku Part 2 > Chapter18 v sekcii Scripts, a videopríručky v Collection of video guides. Materiál pre krátke kurzy je v sekcii Short courses.

18.6.1 Exportovaná aplikácia

Exportovaná aplikácia je zabalená. Inštalácia vytvorí tabuľky a naplní ich údajmi. Odinštalovanie odstráni všetky objekty databázy, ktoré boli použité v tejto aplikácii.

18.6.2 Video sprievodca

Video návod ukazuje všetky kroky pri vývoji aplikácie.

18.7 Otázky

1. Ako by sme mohli zviditeľniť ponuku s oneskorením, od určitého momentu, keď ju uchádzač chce?
2. Môže sa táto aplikácia použiť na četovanie o inej téme než je výmena rastlín a semien? Ako?
3. V relačnom dátovom modeli v CH18_OFFER je v atribúte VALID TO definovaný formát CHAR. Aké sú možné problémy, ak je dátum zapísaný vo formáte znakov v systéme Oracle?

18.8 Odpovede

1. Aby bola ponuka viditeľná s oneskorením od určitého momentu, keď ju uchádzač chce, museli by sme pridať atribút VISIBLE_FROM do CH_18_OFFER a skontrolovať, či je táto podmienka splnená, keď sa zobrazujú ponuky.
2. Teoreticky je to možné. Moderátor četu by mal otvoriť "falošnú" (dummy) ponuku a všetci používatelia, ktorí prídu so svojou falošnou protiponukou, by sa mohli zúčastniť četu.
3. Ukladanie dátumov ako CHAR v Oracle môže spôsobiť problémy s integritou údajov, problémy s triedením, obmedzené operácie s dátumami, problémy s formátovaním a niektoré iné problémy. Na presné a efektívne ukladanie, a manipuláciu s dátumom sa vo všeobecnosti odporúča používať vhodné typy, ktoré poskytuje Oracle, ako napríklad DATE alebo TIMESTAMP.



19. Systém správy knižných recenzií

A. Kutnjak, L. Hrustek, A. Baggia and R.Leskovar

19.1 Biznis pohľad prípadu

Systém na správu recenzií kníh bol navrhnutý s cieľom spravovať recenzie kníh, ktoré sú k dispozícii na predaj na ľubovoľnej on-line platforme. Spoločnosť (Amazon) uvažuje o správe katalógu dostupných kníh tak, že na základe recenzií používateľov rozhoduje o ponuke kníh ich pridaním alebo odobratím. Pomocou tohto systému spoločnosť zhromažďuje recenzie používateľov, vytvára jedinečnú databázu a na základe zhromaždených údajov rozhoduje o ďalšom postupe. Základnými účastníkmi databázy sú správca a používateľ, čitateľ knihy.

Prípad použitia a myšlienka sú inšpirované a ponúkané platformou Kaggle - online komunitnou platformou pre dátových vedcov a nadšencov strojového učenia. Pôvodné údaje sú k dispozícii na tejto stránke. Prípady použitia a dátový model sa zásadne upravujú a dopĺňajú. Cieľom tohto prípadu je demonštrovať vývojové možnosti systému APEX, ktorý sa spolieha len na sprievodcov. S výnimkou Quick SQL (celkovo 31 krátkych riadkov) nebude vývojár písať žiadny iný kód.

19.2 Definícia problému

Systém správy recenzií má tieto funkcie:

- Vzhľadom na širokú škálu kníh v globálnej ponuke zhromažďuje údaje o ich recenziách a hodnotení.
- Bežné zobrazenie platformy obsahuje zoznam kníh s informáciami o názve, autoroch, popise obsahu, vydavateľovi, dátume vydania, kategóriách a recenziách.
- Možnosti správcu zahŕňajú pridávanie kníh do databázy platformy, ako aj ich úpravu a výber podľa kategórií.
- Keďže databáza umožňuje používateľom pridávať recenzie, správca môže prezerať recenzie a komentáre. Na základe toho správca rozhoduje o aktivitách týkajúcich sa jednotlivých kníh (pridanie alebo odstránenie).
- Používateľ má možnosť zaregistrovať sa na platforme, čo mu umožní prezerať si podrobnosti o knihách, ale aj prezerať si ich a komentovať recenzie.
- Recenzie na strane používateľa majú dve výhody. Po prvé, na základe analýzy recenzií sa potenciálny používateľ rozhoduje o kúpe knihy. Po druhé, na základe existujúcich recenzií spoločnosť vykonáva kroky týkajúce sa kníh, t. j. pridáva alebo odstraňuje ich z katalógu.

19.3 Prípady použitia

19.3.1 Naratívny popis

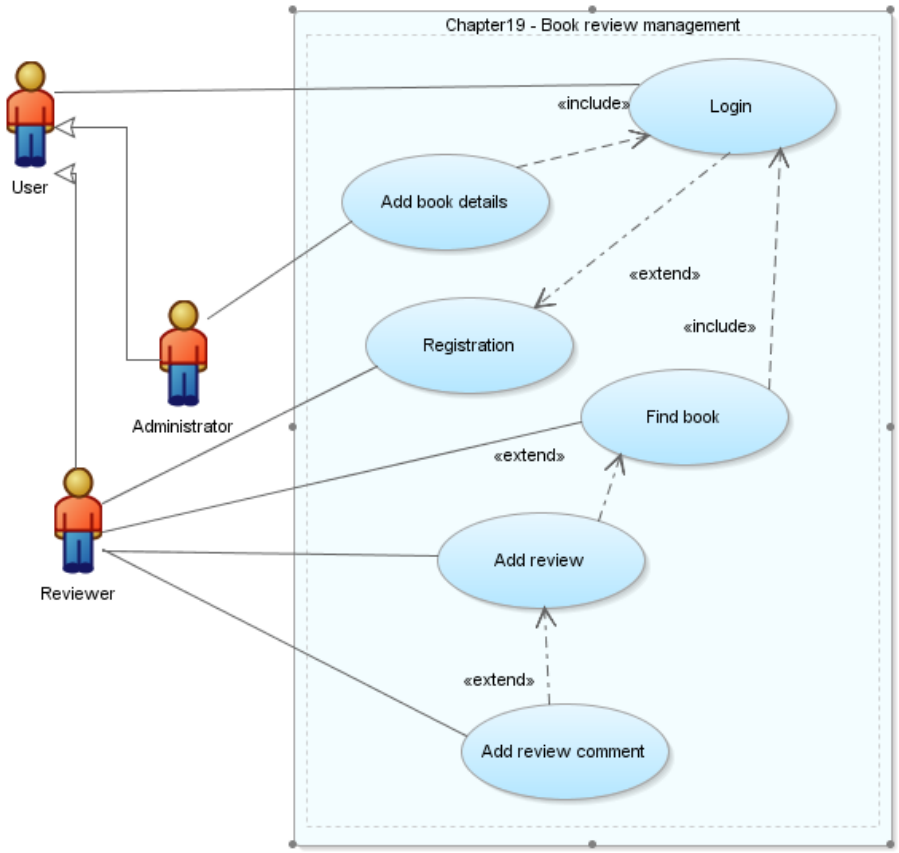
Správca má na starosti zadávanie alebo zmenu údajov o knihách v rámci systému správy recenzií a má možnosť prezerať si rôzne štatistické ukazovatele týkajúce sa kníh. Systém správy recenzií Amazon je založený na registrácii používateľa na platforme. Po registrácii si používatelia môžu prezerať katalóg kníh dostupných na platforme. Katalóg kníh obsahuje podrobné informácie o dostupných knihách a je základom pre ďalšie činnosti v rámci platformy. Knihy sú rozdelené do kategórií, čo uľahčuje vyhľadávanie a umožňuje využívať funkciu osobných preferencií používateľa. Registrovaný používateľ môže pridať recenziu a komentár k recenzii.

19.3.2 Pološtruktúrovaný popis

Činnosti na zadávanie nových údajov o knihách sú pridelené správcovi. Ten môže doplniť aktuálnu databázu o nové knihy, čo má za následok doplnenie ponuky v knižnom katalógu. Frekvencia používania platformy správcom je na dennej báze. Používateľ vykoná registráciu na platforme systému na správu knižných recenzií. Po úspešnej registrácii si používateľ prezrie katalóg kníh, ich recenzie a nakoniec pridá vlastnú recenziu a komentáre k recenziám. Frekvencia používania platformy používateľmi je niekoľkokrát mesačne.

Opis prípadov použitia je uvedený v tabuľke 19.1.

Vyššie uvedený príbeh je znázornený na diagrame prípadov použitia (viď Obrázok 19.1).



Obr. 19.1: Use case diagram: administrátor pridáva nový komentár.

Tabuľka 19.1: Popis prípadu použitia: analýza recenzií kníh

Kľúčové slovo	**Hodnota**
ID:	*ch19-10*
Názov:	*Analýza recenzií kníh*
Popis:	*Systém manažovania recenzií kníh je dostupný na základe registrácie do platformy. Platforma umožňuje prezeranie katalógov, ich filtrovanie a ich klasifikáciu vzhľadom na kategórie.*
Hlavný aktér:	*User*
Predpoklady:	Povinná registrácia do platformy umožní pozeranie údajov o knihe.
Podmienky po skončení:	Vykonávanie ďalších akcií v rámci platformy: pridávanie a odstraňovanie kníh (administrátor) a prezeranie a komentovanie alebo kupovanie kníh (používateľ).
Hlavné	*Scenáre*
Úspešný scenár:	Prvý scenár (administrátor, pridávanie novej knihy): 1. Administrátor sa prihlási do platformy. 2. Administrátor vyberie vstupný formulár pre tvorbu novej knihy. 3. Vstupný formulár zahŕňa nadpis, autorov, popis, vydavateľa, dátum vydania a výber kategórie. 4. Administrátor dokončí procedúru vytvorením nového záznamu. 5. Administrátor sa odhlási z platformy. Druhý scenár (používateľ, recenzia knihy): 1. Používateľ sa registruje do platformy. 2. Používateľ prezerá katalóg knih a vyberie knihu, ktorú chce recenzovať. 3. Recenzný formulár zahŕňa nadpis, cenu, profilové meno, užitočnosť recenzie, skóre recenzie, sumár recenzie a text recenzie. 4. Používateľ dokončí procedúru zadaním novej recenzie. 5. Používateľ sa odhlási z platformy. Tretí scenár (používateľ, komentovanie recenzie knihy): 1. Používateľ sa registruje do platformy. 2. Používateľ prezerá recenzie kníh. 3. Recenzia zahŕňa nadpis, cenu, profilové meno, užitočnosť recenzie, skóre recenzie, sumár recenzie a text recenzie. 4. Používateľ pridáva nový komentár recenzie knihy. 5. Používateľ sa odhlási z platformy.
Rozšírenia:	• *Vloženie knihy* • *Interaktívny report pre knihu* • *Recenzie knihy* • *Komentáre na recenziu*
Frekvencia použitia:	*Na dennej báze, niekoľko krát za mesiac.*
Stav:	*[Vo vývoji]*
Vlastník:	*Potencionálny používatelia na základe registrácie*
Priorita:	*Vysoká*

19.4 Dátový model

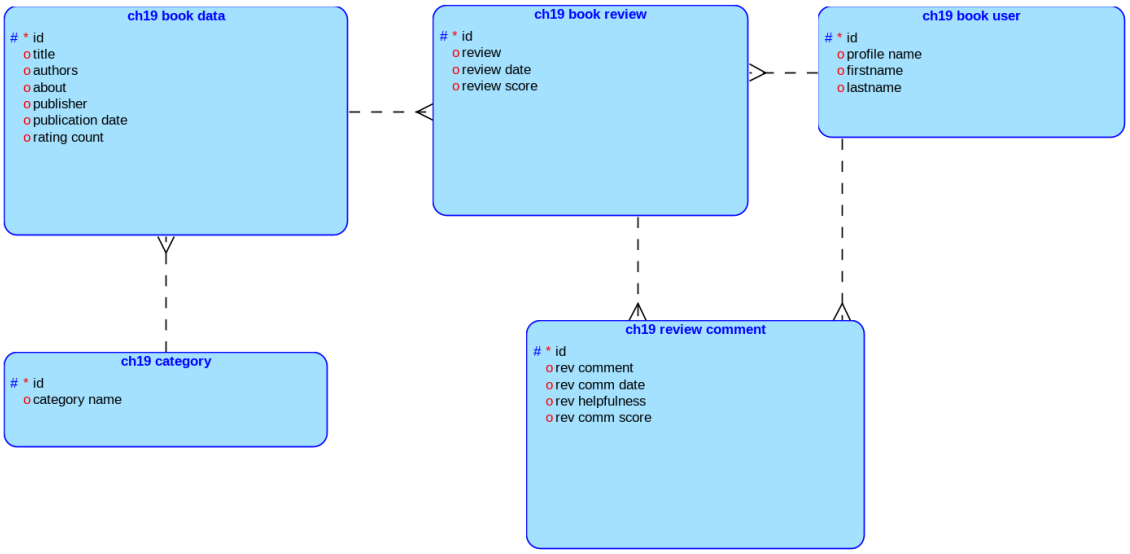
19.4.1 Naratívny popis dátového modelu

Logický dátový model (Entity-Relationship model) pozostáva z niekoľkých entít: category, book_data, book_user, book_review a book_review_comment. Entita category obsahuje atribút category_name. Entita book_data obsahuje niekoľko atribútov: title, authors, about, publisher, publication_date a rating_count. Entita book_user obsahuje profile_name, firstname a lastname. Entita book_review obsahuje atribúty: review, review_date a review_score. Entita review_comment obsahuje odpovede na recenzie: rev_comment, rev_helpfulness a rev_comm_score. Každá kniha patrí len do jednej kategórie. Kategória ale môže mať viacero kníh. K jednej knihe sa môže viazať viac recenzií. Používateľ môže komentovať viac kníh a viac recenzií od iných používateľov. Do každej entity pridáme primárne UID nazývané ID. Tieto atribúty sa stanú primárnymi kľúčmi v príslušných tabuľkách. Po vygenerovaní SQL skriptu na vytvorenie tabuľky pridáme dve kontrolné obmedzenia - povolíme, aby hodnotenie recenzie a komentár k hodnoteniu recenzie mali len číselné hodnoty od 1 do 5.

Vývojár má niekoľko možností vytvárania tabuliek. Jednou z nich je použitie programu SQL Developer Data Modeler a navrhnutie logického modelu od začiatku, transformácia na relačný model a vygenerovanie skriptu SQL. Ďalšou možnosťou je vytvoriť tabuľky v APEX, exportovať skripty, ktoré vytvárajú tabuľky, použiť spätné inžinierstvo v SQL Develoepr Data Modeler na vygenerovanie relačného modelu a potom ho transformovať na logický model. Použitie iba funkcie APEX Quick SQL je najjednoduchší prístup. Po napísaní Quick SQL ho možno transformovať do skriptu SQL pomocou vstavaného generátora APEX.

19.4.2 Logický dátový model

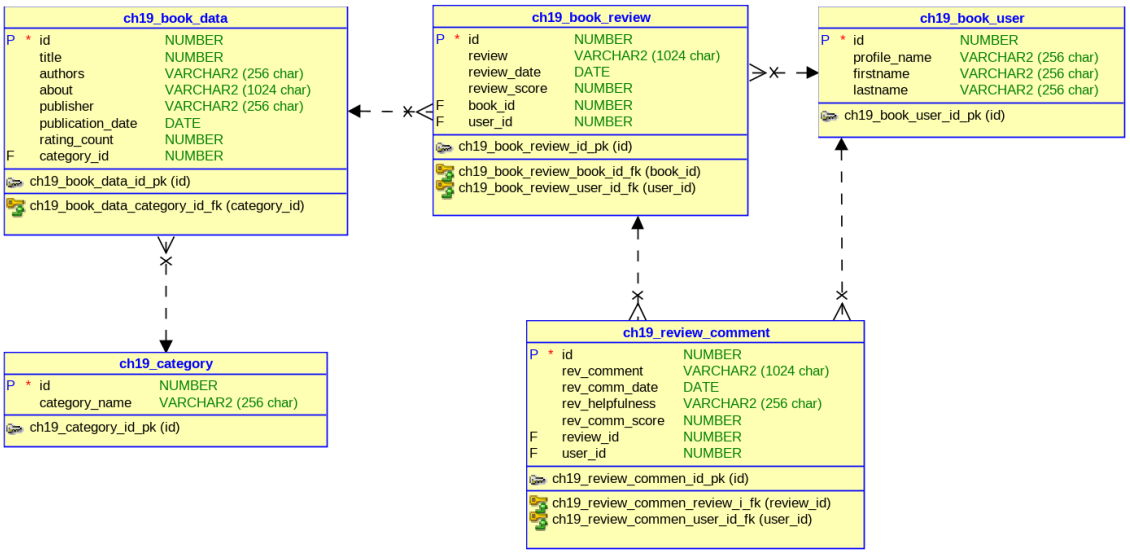
Logický dátový model je zobrazený na Obrázku 19.2.



Obr. 19.2: Logický dátový model.

19.4.3 Relačný dátový model

Automatickú transformáciu z druhého logického dátového modelu na relačný dátový model v programe Oracle SQL Data Modeler zabezpečuje funkcia Engineer to Relational Model. Výsledok je znázornený na Obrázku 19.3.



Obr. 19.3: Relačný dátový model.

Pomocou Oracle SQL Data Modeler môžete vygenerovať SQL skript. Vyberte všetky tabuľky v relačnom modeli a pomocou funkcie File > Export > DDL File získajte skript. Vygenerovaný skript môžeme importovať do systému APEX a spustiť ho.

19.4.4 SQL skript

Skript, ktorý zodpovedá relačnému modelu, obsahuje nasledujúce príkazy SQL:

```
1   create table ch19_category (
2   id number generated by default on null as identity
3   constraint ch19_category_id_pk primary key,
4   category_name varchar2(256 char));
5
6   create table ch19_book_user (
7   id number generated by default on null as identity
8   constraint ch19_book_user_id_pk primary key,
9   profile_name varchar2(256 char),
10  firstname varchar2(256 char),
11  lastname varchar2(256 char));
12
13  create table ch19_book_data (
14  id number generated by default on null as identity
15  constraint ch19_book_data_id_pk primary key,
16  title varchar2(256 char),
17  authors varchar2(256 char),
18  about varchar2(1024 char),
19  publisher varchar2(256 char),
20  publication_date date,
21  rating_count number,
22  category_id      number
23  constraint ch19_book_data_category_id_fk
24  references ch19_category on delete cascade);
25
```

```
create table ch19_book_review (
id number generated by default on null as identity
constraint ch19_book_review_id_pk primary key,
review varchar2(1024 char),
review_date date,
review_score  number
constraint ch19_book_revie_review_scor_ck
check (review_score in (1,2,3,4,5)),
book_id   number
constraint ch19_book_review_book_id_fk
references ch19_book_data on delete cascade,
user_id   number
constraint ch19_book_review_user_id_fk
references ch19_book_user on delete cascade);

create table ch19_review_comment (
id  number generated by default on null as identity
constraint ch19_review_commen_id_pk primary key,
rev_comment varchar2(1024 char),
rev_comm_date date,
rev_helpfulness varchar2(256 char),
rev_comm_score number
constraint ch19_review_c_rev_comm_scor_ck
check (rev_comm_score in (1,2,3,4,5)),
review_id number
constraint ch19_review_commen_review_i_fk
references ch19_book_review on delete cascade,
user_id   number
constraint ch19_review_commen_user_id_fk
references ch19_book_user on delete cascade);
```

19.4.5 Quick SQL

Najjednoduchším postupom na generovanie skriptu SQL je použitie funkcie Quick SQL v systéme APEX. Nasledujúce riadky funkcie Quick SQL umožňujú vygenerovať vyššie uvedený skript SQL:

```
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
```

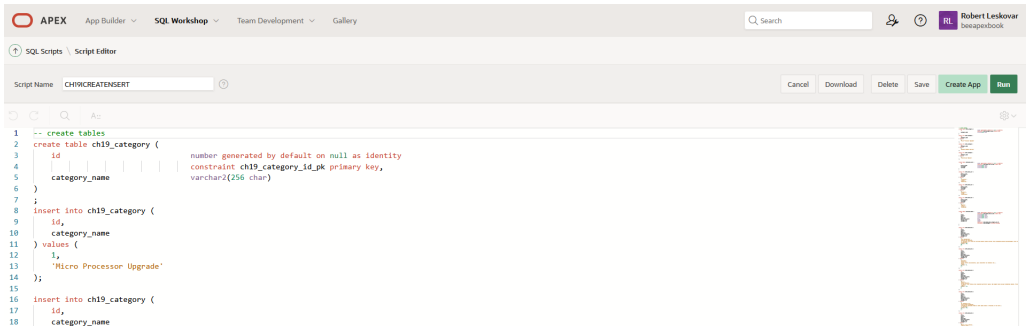
```
17
18          ch19_book_review
19          review vc1024
20          review_date date
21          review_score  num /check 1,2,3,4,5
22          book_id /fk ch19_book_data
23          user_id /fk ch19_book_user
24
25          ch19_review_comment
26          rev_comment vc1024
27          rev_comm_date date
28          rev_helpfulness vc256
29          rev_comm_score  num /check 1,2,3,4,5
30          review_id /fk ch19_book_review
31          user_id /fk ch19_book_user
```

Kód je veľmi "obsiahly". Vyžaduje však len základné znalosti dátového modelovania a syntaxe Quick SQL.

19.5 Rozhrania aplikácie

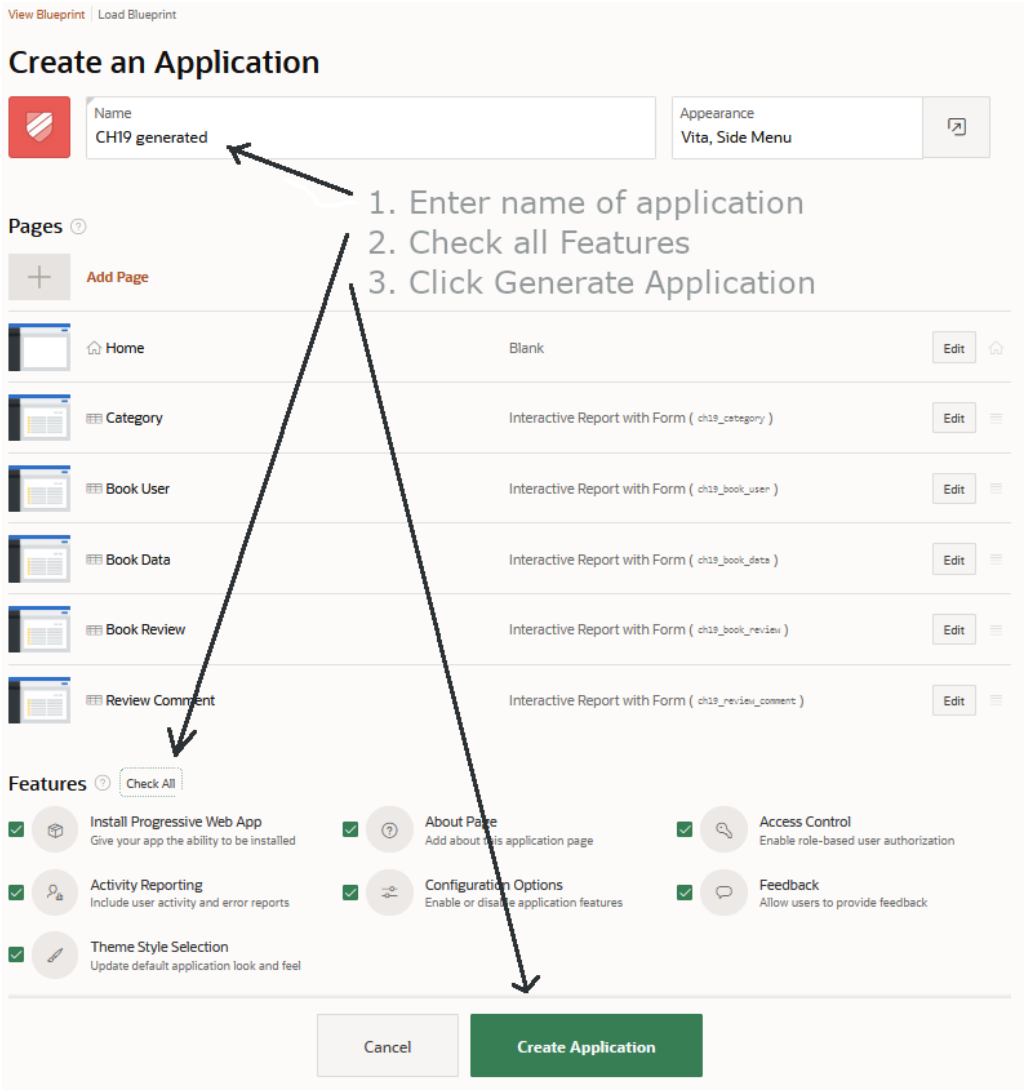
19.5.1 Administrátor

V tomto prípade budeme generovať aplikáciu zo skriptu SQL, ktorý vytvorí tabuľky a naplní vzorové údaje. Nájdite súbor CH19CREATEINSERT.sql v učebných materiáloch a importujte ho do workspace. Po importovaní ho otvorte a kliknite na Create App (viď Obrázok 19.4).



Obr. 19.4: Generovanie aplikácie zo skriptu CH19CREATEINSERT - časť 1.

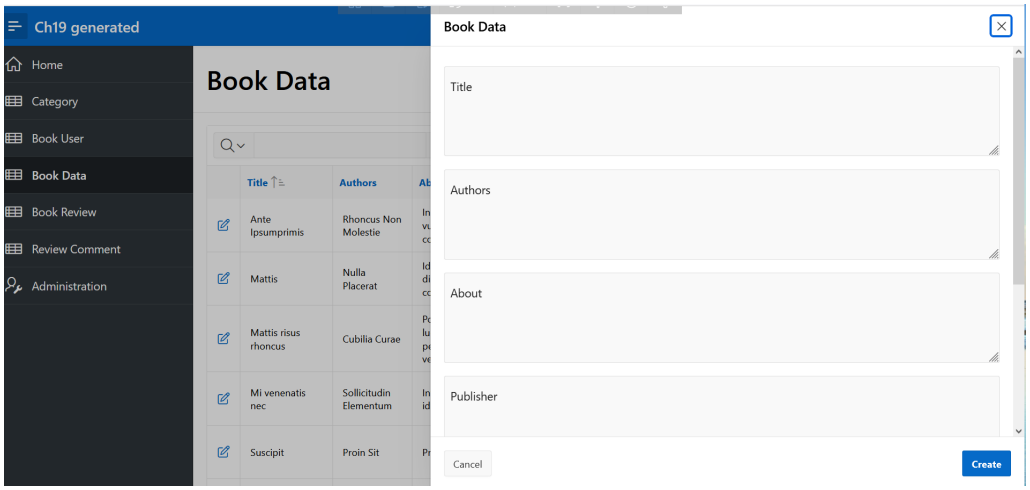
Pomenujte aplikáciu, vyberte všetky funkcie a kliknite na Generate Application (viď Obrázok 19.5).



Obr. 19.5: Generovanie aplikácie zo skriptu CH19CREATEINSERT - časť 2.

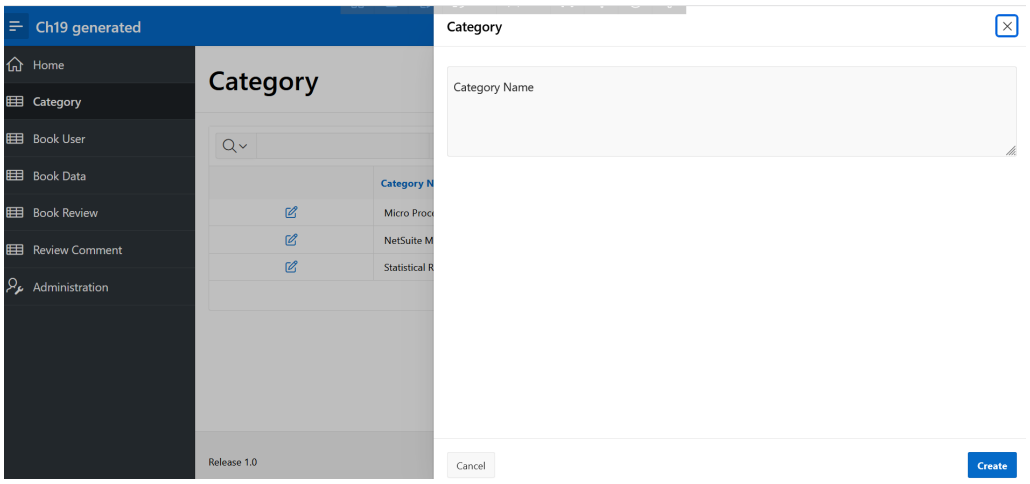
Pracovný prototyp aplikácie je v tomto momente pripravený. Odteraz sú dostupné funkcie pre správcu a koncového používateľa. Vývojár zatiaľ nenapísal ani jeden riadok kódu (s výnimkou Quick SQL) a neupravil žiadne rozhranie aplikácie. Skutočný kód za aplikáciou bol kompletne zostavený sprievodcami APEX.

Obrázok 19.6 zobrazuje formulár na zadanie novej knihy správcom.



Obr. 19.6: Pridanie knihy správcom.

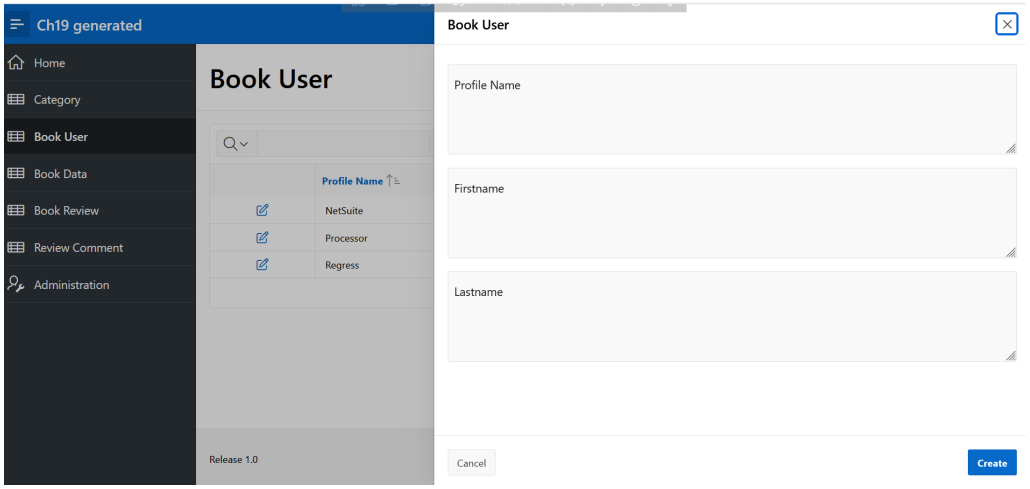
Obrázok 19.7 zobrazuje formulár na vloženie kategórie knihy správcom.



Obr. 19.7: Pridanie kategórie správcom.

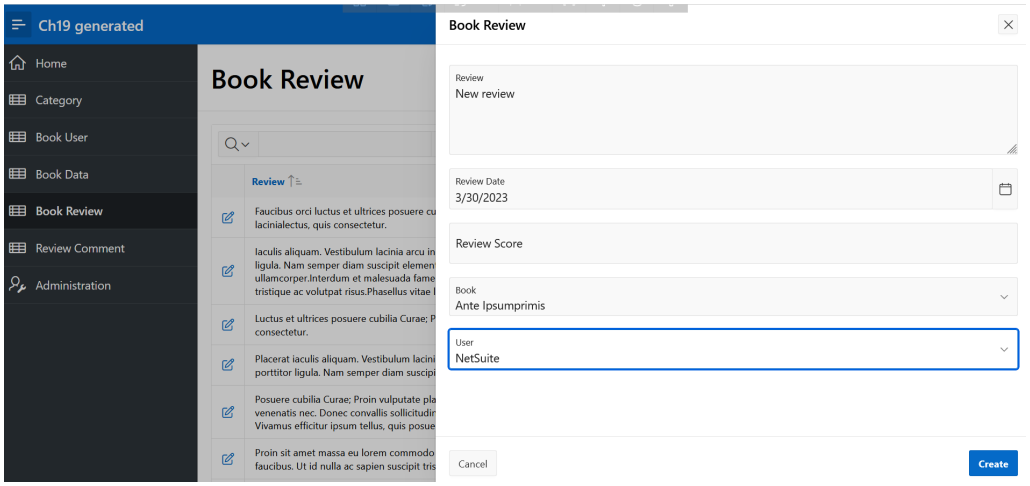
19.5.2 Používateľ

Obrázok 19.8 zobrazuje registračný formulár používateľa.



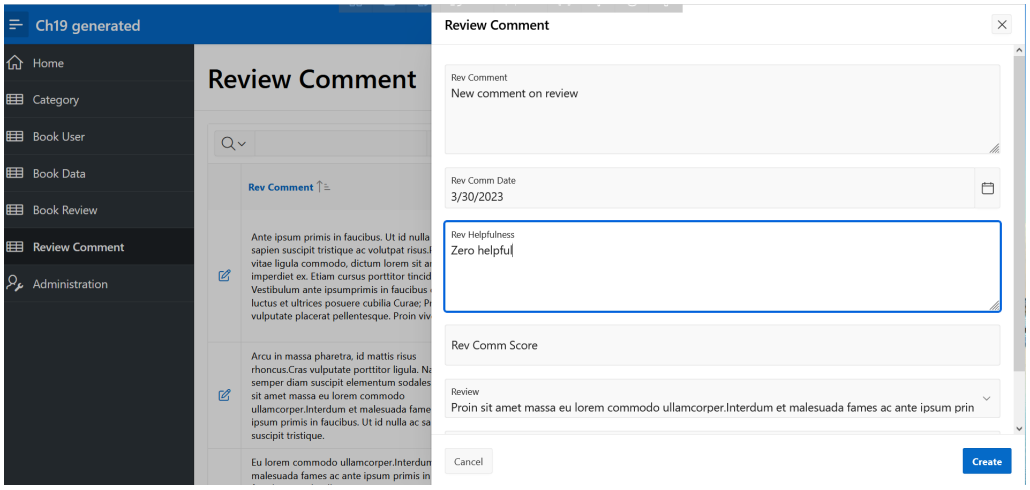
Obr. 19.8: Registrácia používateľa

Obrázok 19.9 zobrazuje prehliadanie a pridávanie recenzií.



Obr. 19.9: Prehliadanie a pridávateni recenzií.

Obrázok 19.10 zobrazuje formulár na komentovanie recenzie.



Obr. 19.10: Formulár na komentovanie recenzie

Prehľady sú štandardne tabuľkové. Koncový používateľ môže generovať grafy bez programovania, len pomocou vstavaného sprievodcu. Informácie o transformácii zostáv na grafy nájdete v Kapitole 8.

19.6 Definovanie užívateľských rolí

Používateľské roly možno definovať v časti Application Access Control v Shared Components. V tejto chvíli preskočíme podrobnosti a implementáciu pre tento konkrétny prípad. Rýchly prehľad nájdete v Kapitole 13 alebo pre dôkladný prehľad si preštudujte dokumentáciu APEX o autorizácii.

19.7 Doplnkový učebný materiál

Nájdete tu tieto doplnkové učebné materiály:

- skript na vytváranie a napĺňanie tabuliek
- skript na zrušenie tabuliek
- exportované zbalené aplikácie
- video zobrazujúce generovanie aplikácie zo skriptu

Všetky doplnkové vzdelávacie materiály sú k dispozícii na verejnej BeeAPEX projektovej stránke. Prihláste sa ako hosťujúci používateľ (nevyžaduje sa heslo). Učebnicu nájdete v časti Knihy, skripty v priečinku Časť 2 > Kapitola 19 v časti Skripty a videonávody v časti Zbierka videonávodov. Materiály pre krátke kurzy sa nachádzajú v časti Krátke kurzy.

19.7.1 Exportovaná aplikácia

Exportovaná aplikácia je zabalená. Inštalácia vytvorí tabuľky a vyplní údaje. Odinštalovanie odstráni všetky objekty databázy údajov použité v tejto aplikácii.

Zabalená aplikácia sa otestuje a spustí v novom pracovnom priestore, ak sú splnené nasledujúce požiadavky:

- je pridaný APEX používateľ pred spustením apolikácie. V pracovnom priestore pre vývoj a testovanie prejdite do Shared Components > Application Access Control > Add User Role Assignment; vložte APEX používateľa a nastavte tomuto používateľovi rolu Správcu (Administrator), Prispievateľa (Contributor) alebo Čitateľa (Reader). V produkcii je nevyhnutná konzultácia s kvalifikovaným pracovníkom pred nasadením.

Ak používateľ nemá pridelenú príslušnú rolu, spustenie importovanej aplikácie zlyhá. Po takomto zlyhaní aplikácie z dôvodu nesplnených požiadaviek je potrebné vymazať súbory cookie webového prehliadača (napr. Firefox: Nastavenia > Súbory cookie a údaje o stránkach > Správa

údajov).

19.7.2 Video návody

Video návod ukazuje všetky kroky pri vývoji aplikácie.

19.8 Otázky

1. Prečo bol v tomto prípade použitý Quick SQL skript?
2. Ako je možné vygenerovať viacero formulárov a zostáv naraz?
3. Ako je možné autorizovať používateľa aplikácie APEX v rámci samotnej aplikácie?

19.9 Odpovede

1. Na urýchlenie vývoja dátového modelu bol zavedený skript Quick SQL. Skript Quick SQL generuje skript SQL na požiadanie. SQL skript je spustiteľný, t. j. vytvára databázové objekty, ako sú tabuľky a pohľady.
2. Napísaním skriptu, ktorý obsahuje definíciu viacerých tabuliek, je možné vygenerovať viacero formulárov a zostáv naraz.
3. Ak chcete autorizovať používateľa aplikácie APEX v rámci samotnej aplikácie, aplikácia musí byť vygenerovaná s funkciou kontroly prístupu. Potom môže oprávnený používateľ (Administrator) pridávať ďalších používateľov s rolami Administrator (správca), Contributor (prispievateľ) a Reader (čitateľ).



20. Kusovník a kalkulácia nákladov

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA

20.1 Biznis pohľad prípadu

Malá spoločnosť s názvom **MS** (10-20 zamestnancov) vyrába káblové súpravy, káblové zväzky, vodiče a signálne lampy. Ponúka tiež montáž elektromechanických dielov a mechanických polotovarov. Ich zákazníkmi sú väčšie spoločnosti, ktoré dodávajú domáce spotrebiče, meracie zariadenia, riadiace systémy, elektromery, zdravotnícke zariadenia a podobne. Objednávky zákazníkov prijímajú telefonicky a e-mailom. V prípade jednoduchých výrobkov a operácií, ako aj opakovaných objednávok, **MS** zasiela potvrdenie zákazníkovi veľmi rýchlo. V poslednom čase pribúdajú objednávky na viacúrovňové montované výrobky a zložitejšie polotovary. V jednej objednávke sú aj desiatky položiek a každá položka môže obsahovať výrobok zložený z niekoľkých polotovarov a materiálov. Polotovary sa môžu skladať z polotovarov a materiálov. Zákazníci chcú dostať ponuku rýchlo. Dodanie kalkulácie ceny materiálu a polotovarov pre viacúrovňové montované výrobky a polotovary (kusovník alebo skrátená kalkulácia materiálu) je náročná úloha, či už procesne alebo časovo. Odpoveď zákazníkovi s ponukou môže trvať niekoľko dní. Ďalšou kalkuláciou, ktorá sa pre stručnosť kapitoly vynecháva, je kalkulácia nákladov na prácu (operácie). Rozsah kapitoly sa preto bude týkať len výpočtu materiálu. Majiteľ spoločnosti vyjadril nasledujúce potreby týkajúce sa aplikácie pre kalkuláciu materiálu. Mala by obsahovať:

- prehľad o výrobkoch, polotovaroch, materiáloch a prípadne o dopytoch zákazníkov,
- prehľad o všetkých dieloch a materiáloch (kusovník),
- formulár na úpravu výrobkov, polotovarov, materiálu a prípadne zákazníckych dopytov,
- formulár na úpravu kusovníka,
- kalkuláciu a stromový graf prezentujúci materiálové položky.

Aplikácia by mala byť použiteľná na telefóne, tablete alebo stolnom počítači.

20.2 Definícia problému

Manažér spoločnosti čelí niekoľkým problémom súvisiacim s agilnou reakciou:

- výpočet je procesne a časovo náročný, chybový a musí sa pripraviť veľmi rýchlo.
- nadmerné náklady a chyby vo výpočtoch môžu ovplyvniť podnikanie viacerými spôsobmi, napríklad stratou potenciálnych zákazníkov, nižšími finančnými výsledkami, znížením reputácie podniku, znížením predaja a podobne.

Uvedené riziká by mala účinne zmierniť webová aplikácia.

Tabuľka 20.1: Popis prípadu použitia: vykazovanie a údržba základných údajov.

Kľúčové slovo	**Hodnota**
ID:	*Ch20-01*
Názov:	*Reportovanie a údržba základných dát*
Popis:	*Manažér využíva APEX aplikáciu na reportovanie a údržbu (selektovanie, aktualizovanie) základných dát, či je to produkt, polotovar, materiál alebo dopyt. Materiály a agregáty sú popísané rovnakou množinou atribútov. Aktualizovanie cien materiálov nespustí rekalkuláciu cien agregátov, v ktorých je materiál použitý. Dopyt zákazníka môže pozostávať z viacerých produktov a polotovarov. Merná jednotka bude v tomto prípade kus.*
Hlavný aktér:	*Manažér*
Predpoklady:	*Manažér má účet pre inštanciu APEX.*
Podmienky po skončení:	*Po reportovaní alebo údržbe dát môže byť vykonaná akcia zahŕňajúca výpočet kusovníka.*
Hlavné	*Scenáre*
Úspešný scenár:	1. Otvorte webový prehliadač a prihláste sa do aplikácie. 2. Vyberte položku z menu alebo z navigácie stránky “Prod./Semi/Material”. 3. Na editovanie prejdite na stránke nižšie alebo filtrujte dáta tak, aby ste našli konkrétny produkt, polotovar alebo materiál. 4. Kliknite na ikonku ceruzky pri konkrétnom produkte, polotovare alebo materiáli a aktualizujte dáta. 5. Kliknite na “Apply changes”.
Rozšírenia:	*Žiadne*
Frekvencia použitia:	*Priemerne 5 krát za deň.*
Stav:	*Dokončené*
Vlastník:	*Manažér*
Priorita:	*Vysoká*

20.3 Prípady použitia

20.3.1 Naratívny popis

Základné úlohy, ktoré treba vykonať sú:

- reportovať a uchovávať údaje o výrobkoch, polotovaroch a materiáloch. Spotrebu chápeme ako vykazovanie v tabuľkovej a stromovej forme a údržbu ako pridávanie, aktualizáciu a mazanie.
- vypočítať kusovník (BOM) pre konkrétny dopyt, výrobok alebo polotovar.

Názvy prípadov použitia sú preto “vykazovanie a údržba základných dát”, “vykazovanie a údržba dát o štruktúre” a “vytvorenie kusovníka”.

20.3.2 Pološtruktúrovaný popis

Uvedený príbeh je znázornený na diagrame prípadov použitia.

20.3.3 Diagram prípadov použitia

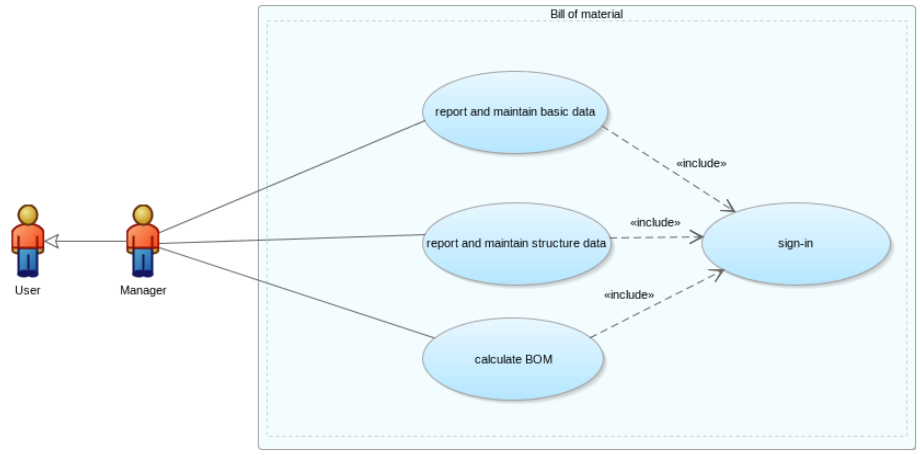
Vyššie uvedený príbeh je znázornený na diagrame prípadov použitia (viď. Obrázok 20.1).

Tabuľka 20.2: Popis prípadu použitia: vykazovanie a údržba dát o štruktúre.

Kľúčové slovo	**Hodnota**
ID:	*Ch20-02*
Názov:	*Reportovanie a údržba štruktúry dát*
Popis:	*Manažér využíva APEX aplikáciu aby reportoval (zobrazil vybrané dáta) a udržoval (vkladanie a mazanie) štruktúru produktov, polotovarov a materiálov. Dopyt zákazníka môže pozostávať z niekoľkých produktov a polotovarov. Vkladanie nových položiek alebo mazanie existujúcich položiek nespustí rekalkuláciu cien agregátov.*
Hlavný aktér:	*Manažér*
Predpoklady:	*Manažér má účet pre inštanciu APEX.*
Podmienky po skončení:	*Po reportovaní alebo údržbe dát sa môže vykonať iná akcia, zahŕňajúca výpočet kusovníka.*
Hlavné	*Scenáre*
Úspešný scenár:	1. Otvorte webový prehliadač a prihláste sa do aplikácie. 2. Vyberte položku z menu alebo z navigácie stránky “Bill-of-material”. 3. Na zvolenie komponentu kliknite na tlačidlo “Create”. 4. Na odstránenie položky v štruktúre kliknite na “Delete”.
Rozšírenia:	*Žiadne*
Frekvencia použitia:	*Priemerne 30 krát za mesiac.*
Stav:	*Dokončené*
Vlastník:	*Manažér*
Priorita:	*Vysoká*

Tabuľka 20.3: Popis prípadu použitia: vytvorenie kusovníka.

Kľúčové slovo	**Hodnota**
ID:	*Ch20-03*
Názov:	*Výpočet kusovníka*
Popis:	*Manažér využíva APEX aplikáciu na výpočet ceny štruktúry popísanej pomocou kusovníka pre špecifickú položku, či je to produkt polotovar alebo dopyt. Výsledné ceny sa aktualizujú pre všetky položky, agregované v štruktúre, bez ohľadu na ich zložitosť a hĺbku, vrátane najvyššieho agregátu.*
Hlavný aktér:	*Manažér*
Predpoklady:	*Manažér má účet pre APEX inštanciu.*
Podmienky po skončení:	*Po výpočte môže byť vykonaná ľubovoľná akcia vrátane ďalšieho výpočtu kusovníka.*
Hlavné	*Scenáre*
Úspešný scenár:	1. Otvorte webový prehliadač a prihláste sa do aplikácie. 2. Vyberte položku z menu alebo z navigácie stránky “Tree view and calculation”. 3. Vyberte agregát na výpočet. 4. Kliknite na tlačidlo “Calculate”. Objaví sa správa zobrazujúca stav výpočtu a vypočítanú cenu. 5. Rozbaľte alebo zbaľte stromovú štruktúru vybranej položky, aby ste zobrazili údaje podčastí..
Rozšírenia:	*Žiadne*
Frekvencia použitia:	*Priemerne 30 krát za mesiac.*
Stav:	*Dokončené*
Vlastník:	*Manažér*
Priorita:	*Vysoká*



Obr. 20.1: Diagram prípadu použitia.

20.4 Dátový model

20.4.1 Naratívny popis

V dátovom modeli sú pre tento prípad len dve entity. Entita **ch20 psm** (neskôr označovaná ako tabuľka) opisuje výrobky, polotovary, materiály a dopyty zákazníkov len s tromi spoločnými atribútmi: ID, popis, merná jednotka a cena za mernú jednotku. Hodnoty všetkých atribútov sú povinné. Prečo? Nemá zmysel ukladať inštanciu bez toho, aby sme vedeli, ako sa volá, ako sa meria a aká je cena za jednotku. ID je jedinečné číslo bez desatinnej časti. Popis musí obsahovať aspoň jeden znak, ktorý sa dá vytlačiť. Oblasť merných jednotiek je obmedzená na milimeter, centimeter, meter a kus. Vzhľadom na malý počet odlišných hodnôt merných jednotiek sa nezavádza žiadna ďalšia entita, ktorá by prezentovala skutočnosť, že jedna merná jednotka má mnoho výskytov v **ch20 psm**, zatiaľ čo jedna inštancia **ch20 psm** sa vzťahuje len na jeden prípad mernej jednotky.

Na prezentáciu ľubovoľnej hierarchickej štruktúry (kusovník s ľubovoľnou šírkou a hĺbkou) má entita **ch20 bom** len tri atribúty: identifikáciu čiastkovej časti, identifikáciu agregátu a množstvo čiastkovej časti obsiahnutej v agregáte. Hodnoty všetkých atribútov musia byť prítomné. Dvojica (**identification of sub part, identification of aggregate**) je jedinečná, avšak pridáme primárny kľúč vytvorený ako postupnosť celočíselných hodnôt, aby sme využili výhody prostredia APEX s nízkym počtom kódov. Jedna inštancia **ch20 bom** môže mať dvojitý vzťah s **ch20 psm**: čiastková časť sa vzťahuje na jednu inštanciu **ch20 psm** a agregát sa vzťahuje na jednu inštanciu **ch20 psm**. V opačnom smere máme: jedna inštancia **ch20 psm** môže odkazovať na mnoho čiastkových častí a mnoho agregátov v entite **ch20 bom**. Preto máme medzi **ch20 psm** a **ch20 bom** dva vzťahy 1-to-many. Je potrebné poukázať ešte na jednu vec: jedinečnosť dvojice **(identification of sub part, identification of aggregate)**. Logický model v Oracle Data Modeler nemá "syntax" na vyjadrenie tejto jedinečnosti ako špeciálny objekt v databáze nazývaný index. V relačnom dátovom modeli je ale možné zaviesť jedinečnosť určenej dvojice hodnôt ako index. V relačnom dátovom modeli sa jedinečný index skladá z dvoch polí v tabuľke. Tento jedinečný index zaručuje, že v databáze nebude uložená žiadna duplicita dvojice hodnôt - databázový engine bude chrániť vývojára a koncového používateľa pred narušením integrity údajov.

20.4.2 Implementácia podnikových pravidiel v databáze

Logický a relačný dátový model je jednoduchý. V matematike je súpis materiálu špeciálnym typom grafu, kde je každý uzol spojený len s jedným nadradeným uzlom okrem koreňového uzla, ktorý nemá žiadny nadradený uzol. Hierarchiu je veľmi jednoduché narušiť zavedením ďalšieho spojenia

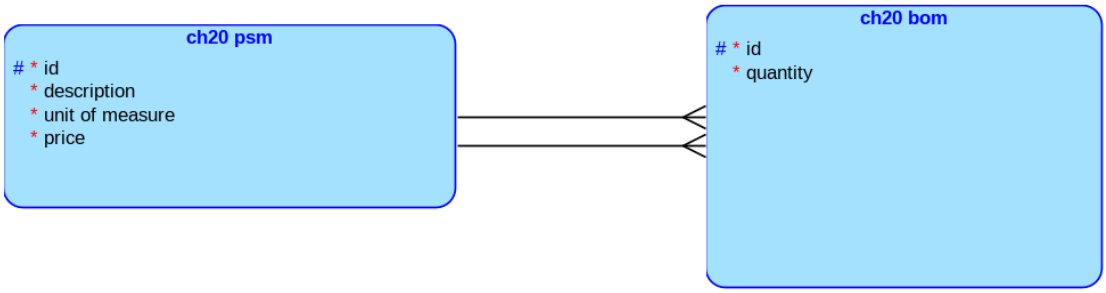
medzi uzlami alebo odstránením spojenia. V kontexte databázy údajov zmeny spojení ovplyvňujú integritu údajov. Uloženie cyklickej štruktúry v **ch20 bom** by spôsobilo, že výpočet kusovníka by sa nikdy neskončil. V kontexte podnikových pravidiel je nemožné a zakázané, aby sa auto (výrobok) skladalo z kolies (polotovar) a zároveň aby sa toto koleso skladalo z áut. Takisto auto nemôže byť zložené samo zo seba. Aby sme zabránili porušovaniu obchodných pravidiel, zavedieme mechanizmus spúšťania (iný objekt databázy), ktorý je zvyčajne reprezentovaný niekoľkými riadkami kódu v jazyku PL/SQL. Spustenie sa vykoná pred zápisom alebo aktualizáciou **ch20 bom**. Takýto kód by zabránil novému vloženiu do **ch20 bom**, ak už existuje dvojica v opačnom poradí alebo ak má nová časť aj agregát rovnakú hodnotu. Implementácia takéhoto druhu pravidiel sa nedá prezentovať na logickom alebo relačnom dátovom modeli.

Takisto nie je možné prezentovať, ako bude vytvorenie kusovníka implementované v dátových modeloch. Táto kapitola poskytne náhľad do kódu, ktorý obráti hierarchiu hore nohami (zdola nahor) a začne sčítavať celkové náklady zdola - t. j. najprv vypočíta náklady na materiál v polotovaroch a tie náklady polotovarov vo výrobku. Nezabudnite, že dátový model s dvoma entitami alebo dvoma tabuľkami je schopný prezentovať akúkoľvek zložitosť (šírku a hĺbku hierarchie) štruktúry kusovníka.

Začnite teda najprv s dátovými modelmi.

20.4.3 Logický dátový model

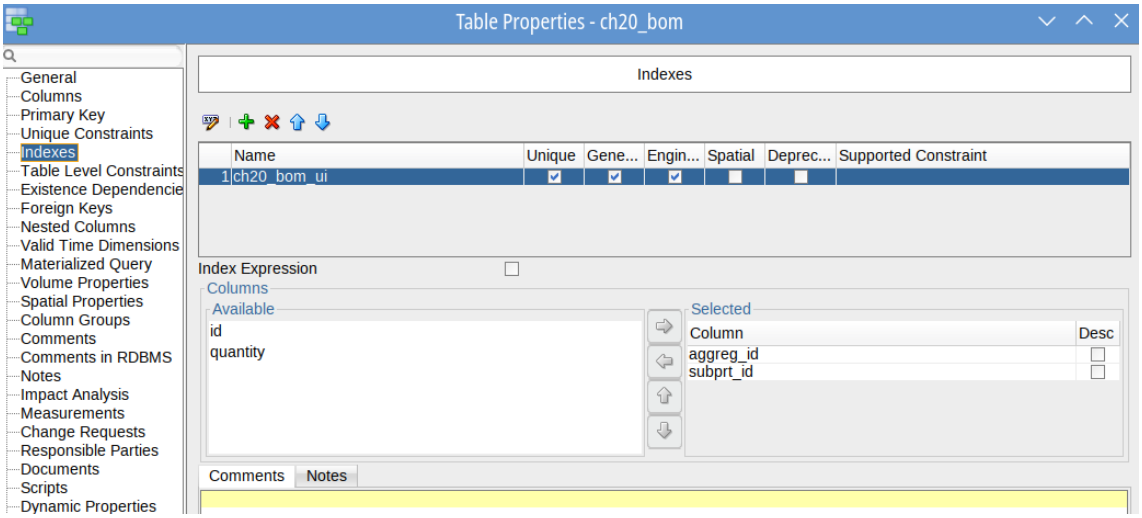
Logický dátový model je uvedený na Obrázku 20.2.



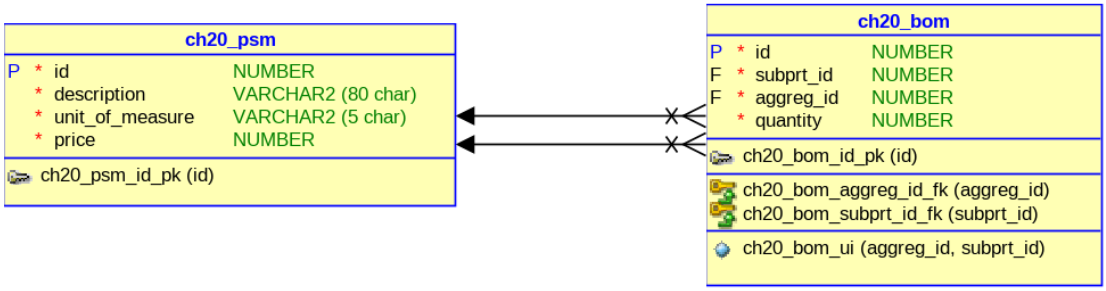
Obr. 20.2: Logický dátový model.

20.4.4 Relačný dátový model

Automatickú transformáciu z logického dátového modelu na relačný dátový model v Oracle SQL Data Modeler zabezpečuje funkcia *Engineering to relational*. Potom sa uvedie unikátny index *ch20_bom_ui* pozostávajúci z dvoch polí *aggreg_id* a *subprt_id* (viď. Obrázok 20.3). Výsledný relačný dátový model pripravený na export ako SQL skript je znázornený na Obrázku 20.4.



Obr. 20.3: Definícia jedinečného indexu v programe Oracle SQL Data Modeler.



Obr. 20.4: Relačný dátový model.

Oracle SQL Data Modeler tiež generuje skript SQL na vytváranie tabuliek, sekvencií a spúšťačov. Vyberte všetky tabuľky v relačnom modeli a pomocou funkcie *File* > *Export* > *DDL File* získajte skript. Uložte skript a skontrolujte poradie definícií tabuliek. Najprv musí byť definovaná tabuľka *ch20_psm* a potom *ch20_bom*. Nasledujúci kód definuje tabuľku *ch20_psm* a nastavuje obmedzenia:

```
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' ) );
```

Nasledujúci kód definuje tabuľku *ch20_bom*, nastavuje obmedzenia a definuje jedinečný index:

```
CREATE TABLE ch20_bom (
id        NUMBER GENERATED BY DEFAULT ON NULL
```

```
AS IDENTITY ( START WITH 1 NOCACHE ) NOT NULL,
subprt_id NUMBER NOT NULL,
aggreg_id NUMBER NOT NULL,
quantity  NUMBER NOT NULL
) LOGGING;

ALTER TABLE ch20_bom ADD CONSTRAINT ch20_bom_id_pk PRIMARY KEY ( id );

ALTER TABLE ch20_bom ADD CONSTRAINT
ch20_bom_subprt_id_fk FOREIGN KEY ( subprt_id )
REFERENCES ch20_psm ( id ) ON DELETE CASCADE NOT DEFERRABLE;

ALTER TABLE ch20_bom ADD CONSTRAINT
ch20_bom_aggreg_id_fk FOREIGN KEY ( aggreg_id )
REFERENCES ch20_psm ( id ) ON DELETE CASCADE NOT DEFERRABLE;

CREATE UNIQUE INDEX ch20_bom_ui ON ch20_bom (aggreg_id, subprt_id );
```

Uložte skript s pokynmi, ktoré vytvoria tabuľky s primárnymi a cudzími kľúčmi, kontrolnými obmedzeniami a jedinečným indexom. Teraz je čas vygenerovať tabuľky v databáze Oracle. Vygenerovaný skript môžeme importovať a spustiť v systéme APEX.

20.4.5 Objekty v APEX

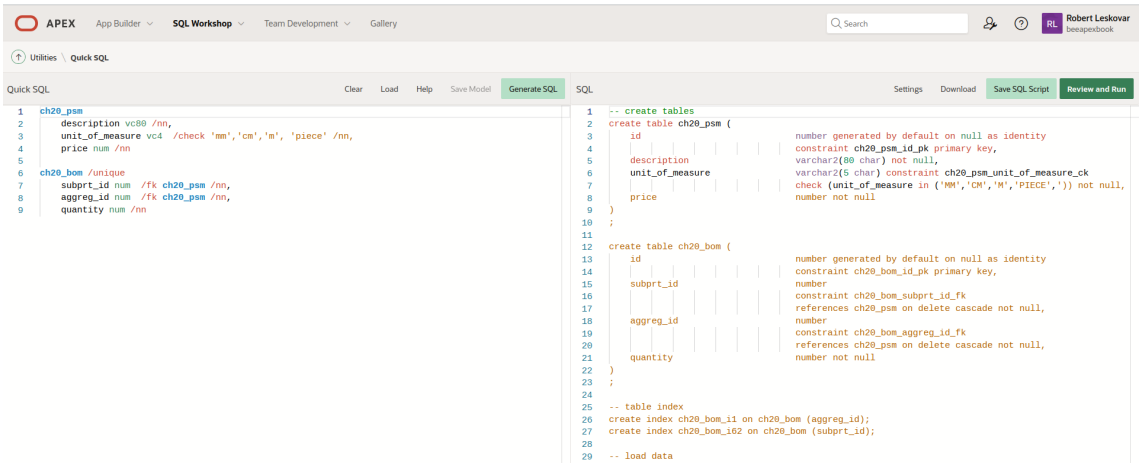
Prejdite v prehliadači do APEX workspace. V tejto časti sa budeme venovať:

- vytvoreniu tabuliek a indexov,
- vytvoreniu funkcie a procedúr,
- vytvoreniu spúšťača.

Existujú dve možnosti vytvorenia tabuliek a indexu. Ak ste na vytvorenie skriptu použili nástroj Oracle Data Modeler, najprv ho importujte (SQL Workshop > SQL Scripts > Upload) a potom ho spustite. Ďalšou možnosťou je vygenerovať uvedený skript pomocou Quick SQL v prípade, že chcete na vygenerovanie takéhoto skriptu použiť iba APEX. Prejdite do SQL Workshop > Utilities > Quick SQL. Do textovej oblasti Quick SQL (ľavá časť okna) vložte nasledujúci text:

```
1          ch20_psm
2          description vc80 /nn,
3          unit_of_measure vc4  /check 'mm','cm','m', 'piece' /nn,
4          price num /nn
5
6          ch20_bom /unique
7          subprt_id num  /fk ch20_psm /nn,
8          aggreg_id num  /fk ch20_psm /nn,
9          quantity num /nn
```

Kliknite na tlačidlo vygenerovať SQL. Tabuľky majú rovnaké vlastnosti ako tabuľky vygenerované v programe Oracle Data Modeler (viď 20.5).



Obr. 20.5: Generovanie skriptu SQL pomocou nástroja Quick SQL.

Pozrite sa na riadky 26 a 27. Tieto dva riadky, hoci sú správne, budú neskôr upravené. Kliknite na tlačidlo Save SQL Script a na Review and Run. Nahraďte riadky 26 a 27 (vytvorte dva indexy) a nahraďte ich jedným príkazom:

```
create unique index ch20_bom_ui on ch20_bom (aggreg_id, subprt_id);
```

Teraz spustite skript. Vytvoria sa dve tabuľky a jeden jedinečný index. Upozorňujeme, že spustenie tohto skriptu nebude úspešné, ak ste už spustili skript vygenerovaný pomocou programu Oracle Data Modeler. Dôvodom je, že objekty databázy údajov už existujú.

Ak chcete vytvoriť funkciu, zadajte nasledujúci text do 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;
```

Táto funkcia vráti cenu (výrobok, polotovar, materiál) pre zadané ID. Ak sa s databázou údajov stane niečo neočakávané (strata spojenia, výpadok servera, hardvérová chyba, žiadne údaje pre zadané ID), potom sa vráti záporná hodnota -505. Vyberte celý text a spustite. Zobrazí sa spätná väzba, že funkcia je vytvorená.

Ak chcete vytvoriť procedúru na výpočet nákladov na vyúčtovanie materiálu, zadajte nasledujúci text do 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
12        --              given product and all underlaying
13        --              levels
14        -- exceptions: system and user defined
15        -- date of testing: 20.1.2023
16        -- author of test: swqlab
17        -- status: APPROVED
18        is
19        v_rezult   NUMBER(38, 2);
20        v_psm        NUMBER;
21        v_price    NUMBER(38, 2);
22        e_negative_price EXCEPTION;
23        CURSOR c_ds IS
24        WITH STRUK(hier, subprt_id, aggreg_id, quantity) AS
25        (SELECT 1 AS hier, subprt_id, aggreg_id, quantity
26        FROM CH20_BOM JOIN CH20_PSM ON subprt_id = CH20_PSM.ID
27        WHERE aggreg_id = xpsm
28        UNION ALL
29        SELECT hier+1, C1.subprt_id, C1.aggreg_id, C1.quantity
30        FROM CH20_BOM C1 JOIN CH20_PSM P ON C1.subprt_id = P.
             ID
31        JOIN STRUK C2 ON C1.aggreg_id = C2.subprt_id)
32        SELECT hier, subprt_id, aggreg_id, quantity
33        FROM STRUK ORDER BY hier DESC, aggreg_id, subprt_id;
34        BEGIN
35        ystatus := 0;
36        v_rezult := 0.0;
37        v_psm := 0;
38        SAVEPOINT old_state;
39        FOR r_ds IN c_ds LOOP
40        v_price:= CH20_GET_PRICE(r_ds.subprt_id);
41        IF ( v_price <= 0 ) THEN
42        RAISE e_negative_price;
43        END IF;
44        IF ( r_ds.aggreg_id = v_psm ) THEN
45        v_rezult := v_rezult + v_price * r_ds.quantity;
46        UPDATE ch20_psm SET price = v_rezult WHERE ID = r_ds.
             aggreg_id;
47        ELSE
48        v_psm := r_ds.aggreg_id;
49        v_rezult := v_price * r_ds.quantity;
50        UPDATE ch20_psm SET price = v_rezult WHERE ID = r_ds.
             aggreg_id;
51        END IF;
52        END LOOP;
```

```
53          COMMIT;
54          ystatus := 1;
55          yprice := v_rezult;
56          EXCEPTION
57          WHEN e_negative_price THEN
58          ROLLBACK TO old_state;
59          ystatus := 0;
60          yprice := -505;
61          WHEN OTHERS THEN
62          ROLLBACK TO old_state;
63          ystatus := 0;
64          yprice := -100;
65          end CH20_CALCULATE_BOM;
```

Tento postup vráti stav výpočtu a cenu (výrobok, polotovar, materiál) pre dané ID. Ak sa s databázou údajov stane niečo neočakávané (strata spojenia, výpadok servera, hardvérová chyba, žiadne údaje pre zadané ID), potom sa vráti záporná hodnota. Ak čitateľ nepozná PL/SQL, stačí si prečítať komentáre v procedúre. Inak sa vytvára pomerne zložitý kurzor pomocou bežného tabuľkového výrazu (príkaz WITH). Tento kurzor vracia zjednotenie dvoch množín. Poskytuje kompletnú štruktúru agregátu, ktorý je usporiadaný zdola nahor. Ceny priebežných agregátov sa ukladajú do príslušných riadkov a kumulujú sa, kým sa nespracujú všetky riadky v kurzoroch.

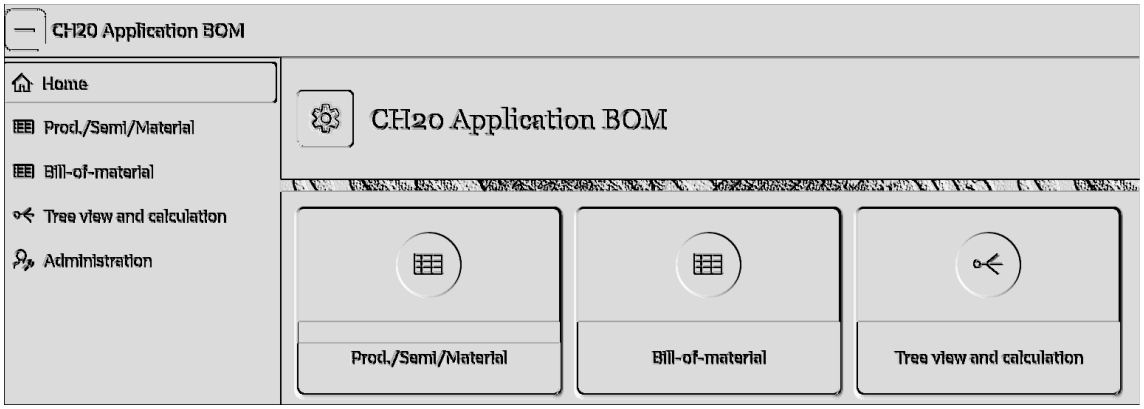
Posledným objektom, ktorý sa má vytvoriť, je spúšťač. Spúšťač zabezpečí, aby žiadny agregát a čiastková časť v kusovníku nemohli mať rovnakú hodnotu (auto je zložené z auta) a vytvorí sa vzťah prechodu (ak je auto zložené z kolies, tak koleso nemôže byť zložené z auta). Ak chcete vytvoriť spúšťač, ktorý implementuje uvedené pravidlá, zadajte v SQL Workshop > SQL Commands nasledujúci text:

```
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;
```

Pomocou tohto spúšťača zabránime vkladaniu tak, že vynútime hodnotu NULL v poli tabuľky, ktoré musí mať hodnotu. Prednosť definície NOT NULL v stĺpci tabuľky pred mechanizmom spúšťača spôsobí, že v prípade porušenia tohto obchodného pravidla sa nevložia žiadne údaje. Chránili sme však hierarchiu, aby sa stala grafom s cyklom. Ak vynecháme vytvorenie tohto spúšťača, mali by sme koncového používateľa preškoliť, aby sa nedostal do pasce cyklu. Je oveľa lepšie predchádzať ako opravovať, však?

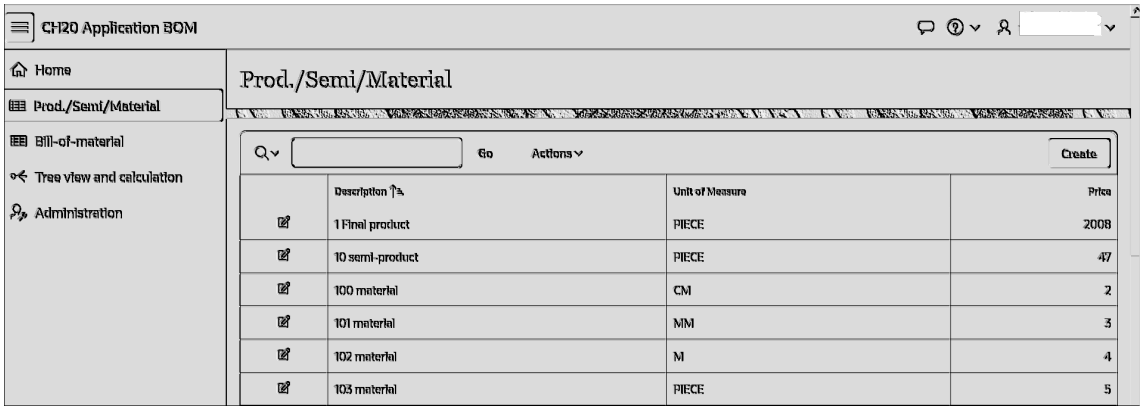
20.5 Rozhrania aplikácie

Na nasledujúcich obrázkoch sú uvedené rozhrania aplikácie. Najprv uvádzame náčrt vstupného okna. Používateľ prejde na správu základných údajov, správu údajov o štruktúre a výpočet buď pomocou ľavého menu, alebo pomocou navigácie po stránke (viď Obrázok 20.6).

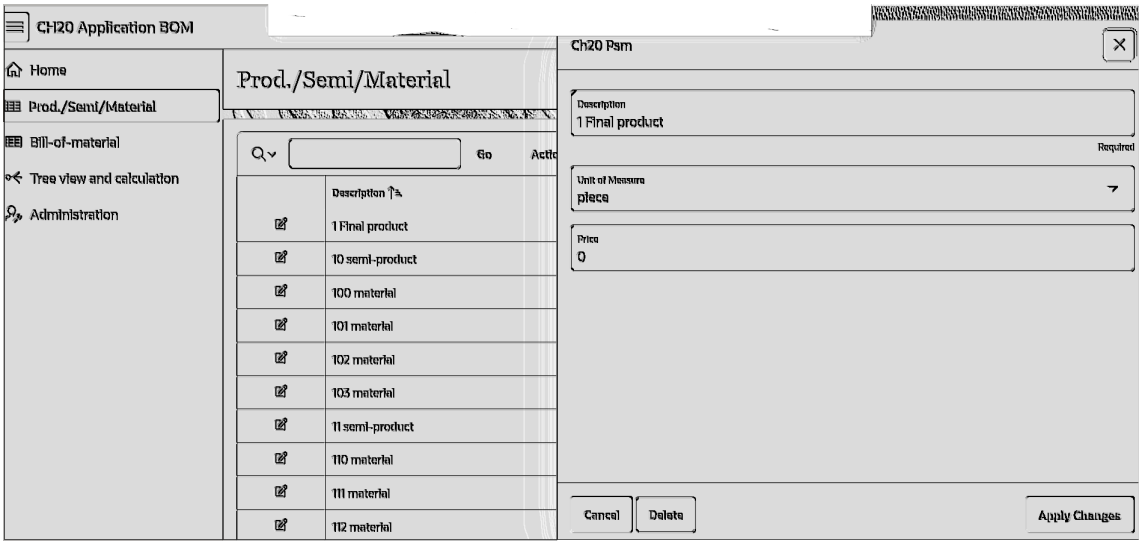


Obr. 20.6: Domovská stránka aplikácie.

Ďalšie dva obrázky, (20.7 a 20.8), predstavujú náčrty správy a formulára pre výrobky, polotovary, materiály a otázky. K dispozícii je vkladanie, aktualizácia a mazanie.

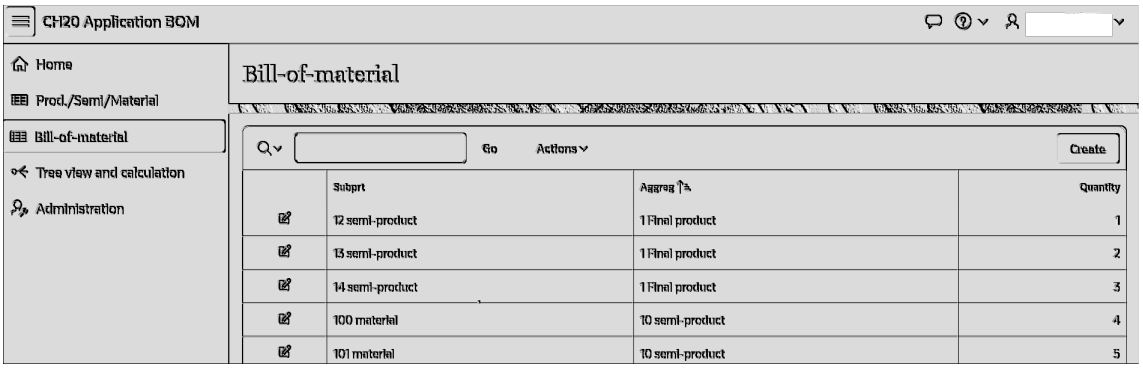


Obr. 20.7: Správa základných údajov - report.

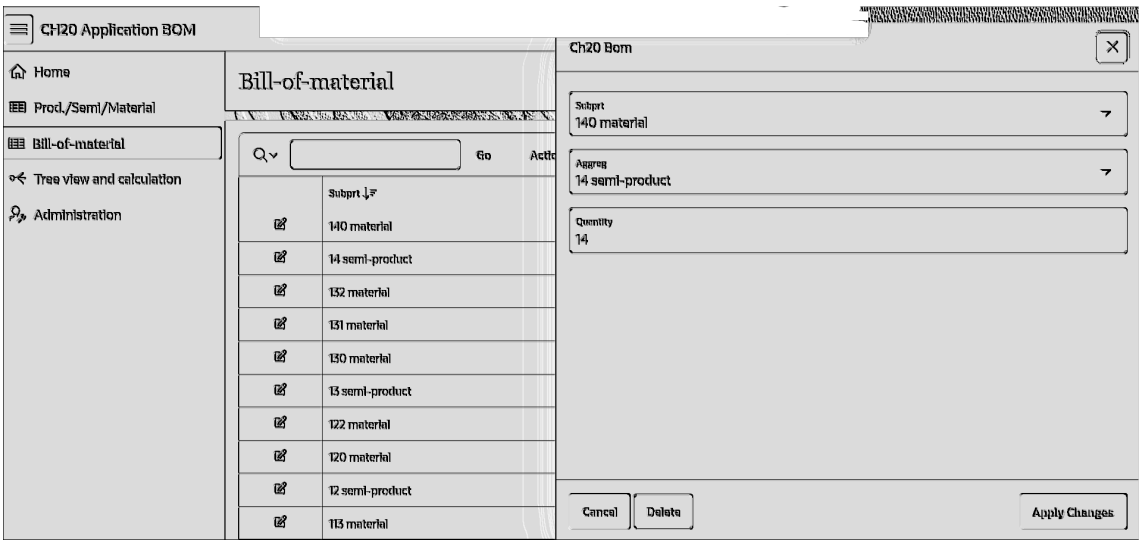


Obr. 20.8: Správa základných údajov - formulár.

Ďalšie dva obrázky, (20.9 a 20.10), predstavujú náčrty správy a formulára pre štruktúru materiálu. K dispozícii je vkladanie, aktualizácia a mazanie.

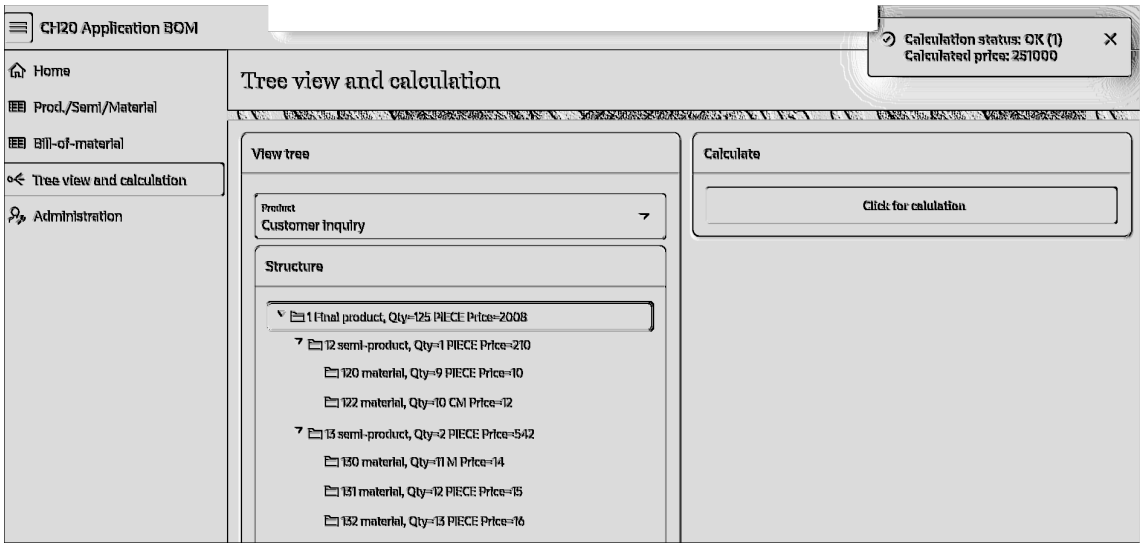


Obr. 20.9: Správa štruktúry - report.



Obr. 20.10: Správa štruktúry dát - form.

Obrázok 20.11 predstavuje rozhranie na vykonanie výpočtu ľubovoľného agregátu (polotovaru, produktu). Používateľ najprv vyberie položku a potom stlačí tlačidlo Calculate. Stav výpočtu a vypočítaná cena sa zobrazí ako vyskakovacie okno a zobrazí sa stromová zložka. Používateľ môže hierarchiu rozbaľovať a zbaľovať.



Obr. 20.11: Strana na výpočet kusovníka.

20.6 Doplnkové účebné materiály

Doplnkové účebné materiály k dispozícii:

- exportovaná aplikácia
- videonávody

Všetky doplnkové učebné materiály sú k dispozícii na verejnej stránke projektu BeeAPEX. Prihláste sa ako hosťujúci používateľ (nevyžaduje sa heslo). Učebnicu nájdete v časti Books, skripty v priečinku Part 2 > Chapter20 v časti Scripts a videonávody v časti Collection of video guides. Materiály pre krátke kurzy sa nachádzajú v časti Short courses.

20.6.1 Exportované aplikácie

Exportovaná aplikácia je zabalená. Inštalácia vytvorí tabuľky, indexy, funkcie, procedúry a spúšťače, ako aj naplní údaje. Odinštalovanie odstráni všetky objekty databázy použité v tejto aplikácii.

20.6.2 Video návody

Video návod zobrazuje každý krok pri vývoji aplikácie.

20.7 Otázky

1. Ako by ste zmenili logický dátový model, aby ste implementovali novú entitu pre merné jednotky, pretože chceme zahrnúť desiatky jednotiek?
2. Ako by ste špecifikovali Quick SQL, aby odrážal vyššie uvedenú zmenu?
3. Aké dôsledky bude mať zrušenie jedinečného indexu ch20_bom_ui?

20.8 Odpovede

1. Použijeme novú entitu *ch20_uom* a nastavíme aspoň dva atribúty (ID a a popis mernej jednotky). Potom nastavíme vzťah 1-to-many medzi *ch20_psm* a *ch20_uom*.

2. V Quick SQL najskôr definujeme tabuľku *ch20_uom* ako:

```
ch20_uom
description vc80
```

a zmeníme definíciu *ch20_psm*:

```
ch20_psm
description vc80 /nn,
unit_of_measure num /fk ch20_uom
price num
```

3. Odstránenie jedinečného indexu ch20_bom_ui by umožnilo ukladať viacero párov (častí, agregátov) s rovnakými hodnotami. Tabuľka by spotrebovala viac miesta na disku. Funkčnosť výpočtu by nebola ovplyvnená, spracovanie by však trvalo dlhšie a spotrebovalo by sa viac pamäte RAM.



21. Manažment výživy a stravovania

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

21.1 Biznis pohľad prípadu

Naším príkladom je najpopulárnejšia značka v Grécku, v oblasti médií zameriavajúcich sa na jedlo, ktorá má viac ako 500 000 užívateľov. Táto spoločnosť je hrdým vlastníkom popredného gréckeho časopisu o jedle a taktiež webovej stránky. Pýši sa aj rozširujúcim sa počtom živých podujatí, vrátane gastronomickej akcie – Greek Food Show – a mimoriadnou úspešnou sériou kuchárskych kníh.

Spoločnosť sa rozhodla vytvoriť a propagovať novým užívateľom webový nástroj, pomocou ktorého má užívateľ cez svoju webovú stránku prístup k tisíckam receptov. Navyše používateľ môže personalizovať svoje obľúbené recepty pridávaním alebo odoberaním ingrediencií. Služba ďalej poskytuje informácie o nutričnej hodnote nových receptov, pričom používateľovi umožňuje vytvoriť si svoju vlastnú personalizovanú kuchársku knihu s ich obľúbenými receptami..

21.2 Definícia problému

Ľudia, ktorí si užívajú varenie často čelia obmedzeniam keď príde na to, čo môžu jesť či už kvôli zdravotným problémom ako je cukrovka alebo vysoký cholesterol, alebo z osobných stravovacích preferencií akými sú vegánstvo alebo bezlepková diéta. Jedným z problémov v súčastnosti existujúcich webovo založených zdrojov receptov je to, že často neposkytujú komplexné alternatívy na prispôsobenie sa týmto obmedzeniam. Výsledkom je, že používateľ môže byť frustrovaný z obmedzeného množstva poskytnutých informácií a celkovo zavrhne vyhľadávanie receptov, čo vedie k jeho nedostatočnému zapojeniu.

21.3 Prípady použitia

21.3.1 Naratívny popis

Vytvoríme Oracle Apex aplikáciu, ktorá umožní používateľom prezerať si a pridávať rozličné zdravé recepty. Aplikácia poskytne detailný popis každého receptu, zahŕňajú zoznam ingrediencií spolu s nutričnými informáciami pre každú z nich.

Používatelia môžu taktiež pridávať svoje vlastné recepty a k nim potrebné ingrediencie prostredníctvom formuláru. Nutričné hodnoty každého receptu, ako je počet kalórií, celkové množstvo bielkovín, počet porcií a čas prípravy budú pre jednoduchú čitateľnosť zobrazené v diagramoch,

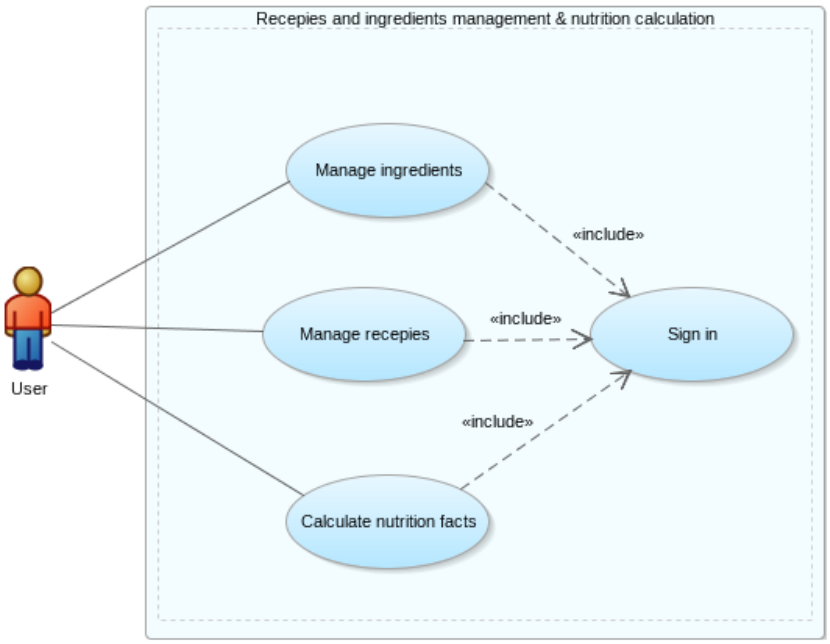
namiesto jednoduchých tabuliek údajov.

21.3.2 Položtruktúrovaný popis

Tento text načrtáva vývoj Oracle Apex aplikácie, ktorá poskytne používateľom prístup k rozličným zdravým receptom. Aplikácia zahŕňa detailný popis receptu a nutričných informácií pre všetky ingrediencie, rovnako tiež umožňuje používateľom pridávať ich vlastné recepty a ingrediencie. Položtruktúrovaný popis je uvedený v tabuľke 21.1.

21.3.3 Diagram prípadov použitia

Hore uvedený príbeh je zobrazený pomocou diagramu prípadov použitia (pozri Obrázok 21.1).



Obr. 21.1: Diagram prípadov použitia.

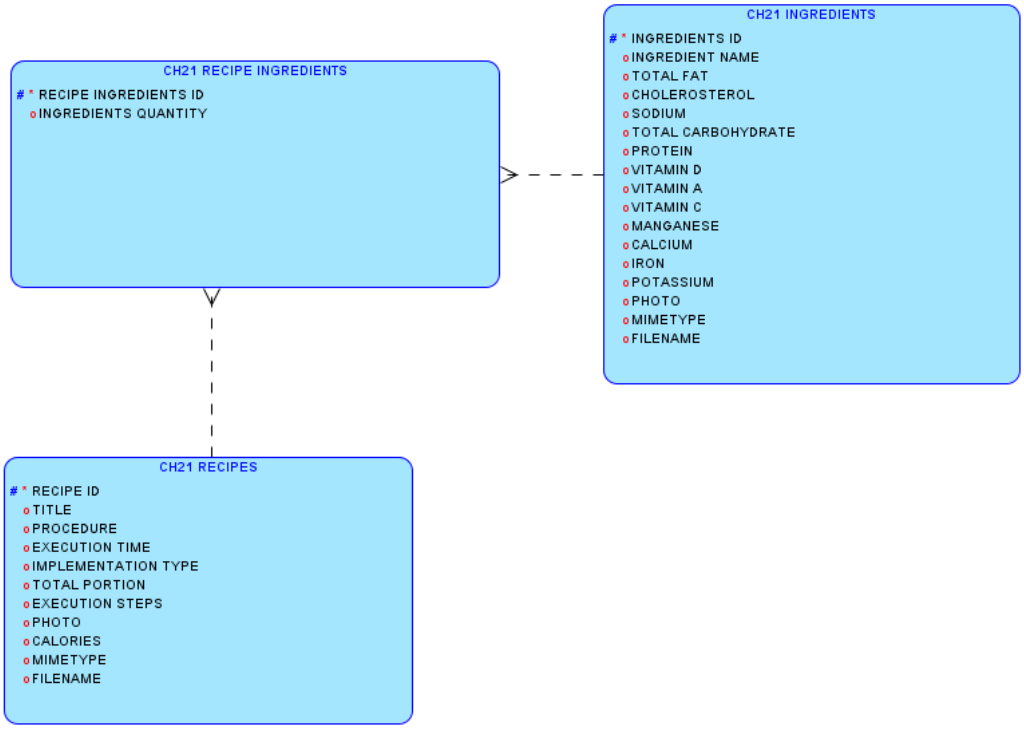
21.4 Dátový model

21.4.1 Logický dátový model

Logický dátový model je zobrazený na Obrázku 21.2.

Tabuľka 21.1: Popis prípadu použitia: Prístup k receptom, ingredienciám a k výpočtu nutričných hodnôt

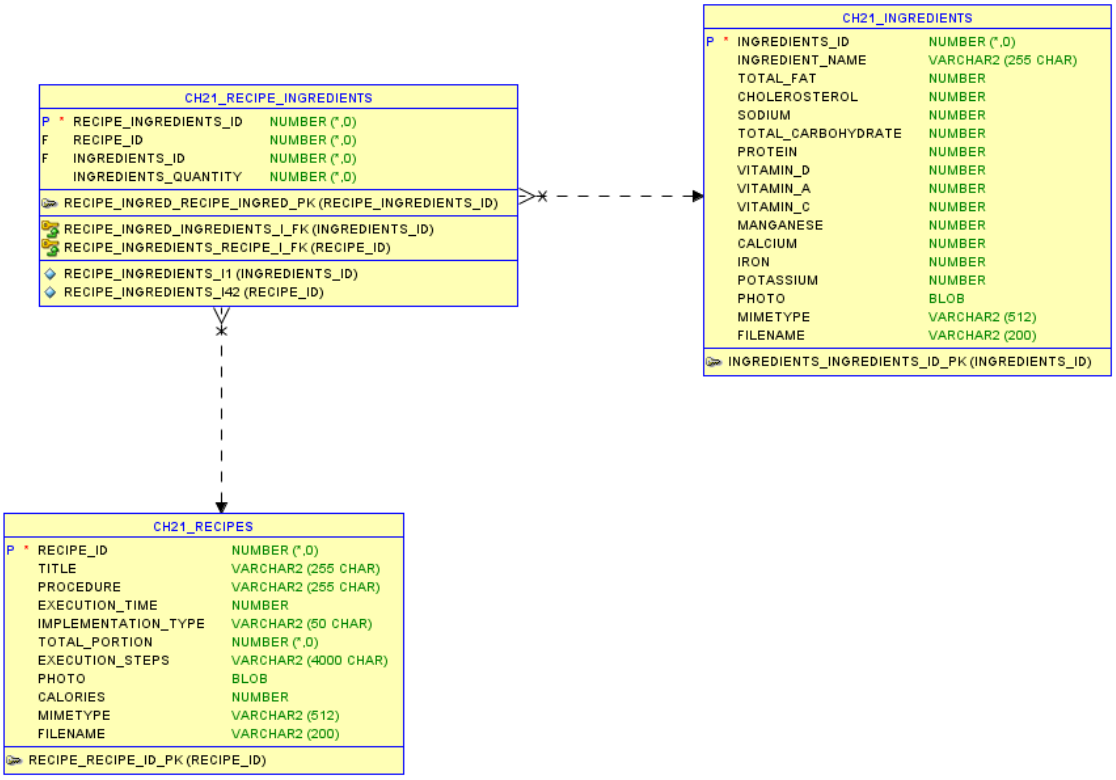
Kľúčové slovo	**Hodnota**
ID:	*Ch21-01*
Názov:	*Platforma poskytuje prístup k rozličným predpripraveným receptom a ingredienciám a taktiež poskytuje možnosť pridávať alebo odstraňovať ingrediencie, za účelom prispôsobenia si receptu podľa individuálnych preferencií. Používatelia môžu vytvárať nové personalizované recepty, vyberaním ich preferovaných ingrediencií. Služba potom vypočíta nutričné hodnoty pre daný recept.*
Popis:	*Pri používaní Apex aplikácie môže používateľ prezerať a vyberať recepty a taktiež konfigurovať údaje o receptoch ako je napríklad vytváranie, aktualizovanie alebo mazanie ingrediencií. Recepty aj ingrediencie sú definované množinou atribútov. Používatelia môžu vytvárať nové recepty pridávaním alebo odstraňovaním ingrediencií a vypočítať aktualizované nutričné hodnoty. Výsledky sú zobrazené v diagramoch, ktoré sú aktualizované v reálnom čase na základe vykonaných zmien v recepte.*
Hlavný aktér:	*Používateľ*
Predpoklady:	Používateľ má účet pre inštanciu APEX.
Podmienky po skončení:	Po aktualizácii a výbere receptu môže používateľ vypočítať nutričné hodnoty. Po vypočítaní nutričných hodnôt pre recept, môžu používatelia opakovať proces pre iné recepty pridaním alebo odobraním ingrediencií, tak aby vytvorili nový personalizovaný recept.
Hlavné	*Scenáre*
Úspešný scenár:	1. Otvorte webový prehliadač a prihláste sa do aplikácie. 2. Vyberte buď to **Recipes** alebo **Ingredients** z možností v menu alebo na stránke s navigáciou. 3. Na pridanie receptu kliknite na tlačidlo **Add a recipe**. Na pridanie ingrediencie kliknite na tlačidlo *Add ingredient*. 4. Na úpravu receptu alebo ingrediencie kliknite na symbol ceruzky pri konkrétnom zázname. 5. Po vykonaní zmien kliknite na tlačidlo **Apply Changes** na to, aby ste ich uložili. Na vymazanie údajov kliknite na tlačidlo **Delete**. 6. Vyberte buď to **Recipes** alebo **Ingredients** z možností v menu alebo na navigačnej stránke. 7. Na pridanie receptu kliknite na tlačidlo **recipe** a vyberte dostupné hotové recepty. 8. Na pridanie ingrediencií do receptu kliknite na tlačidlo **Ingredients** a vyberte zo zoznamu. 9. Na skontrolovanie nutričných hodnôt receptu choďte do menu alebo na stránku s navigáciou a zvoľte **Recipe total nutrition facts**. 10. Poskytnuté nutričné detaily zahŕňajú veľkosť porcie jedla, ktorá môže byť pripravená na základe konkrétneho množstva ingrediencií, celkové množstvo bielkovín, počet kalórií pre daný recept a celkový čas potrebný na prípravu.
Rozšírenia:	*Žiadne*
Frekvencia použitia:	*Stránka publikuje približne 5 receptov za deň alebo približne 1000 receptov za rok, pričom skutočný počet závisí na návštevnosti stránky.*
Status:	*Ukončená*
Vlastník:	*Používateľ*
Priorita:	*Stredná*



Obr. 21.2: Logický diagram.

21.4.2 Relačný dátový model

Relačný dátový diagram je zobrazený na Obrázku 21.3.



Obr. 21.3: Relačný diagram.

21.4.3 QuickSQL a Script SQL Script

Poskytnutý kód slúži ako súbor inštrukcií pre vygenerovanie vzorových dát s využitím Quick SQL nástroja. Tabuľky, ktoré potrebujeme vytvoriť pre našu aplikáciu sú CH21_RECIPES, CH21_RECIPE_INGREDIENTS a CH21_INGREDIENTS. Polia každej tabuľky by mali byť štruktúrované následovne:

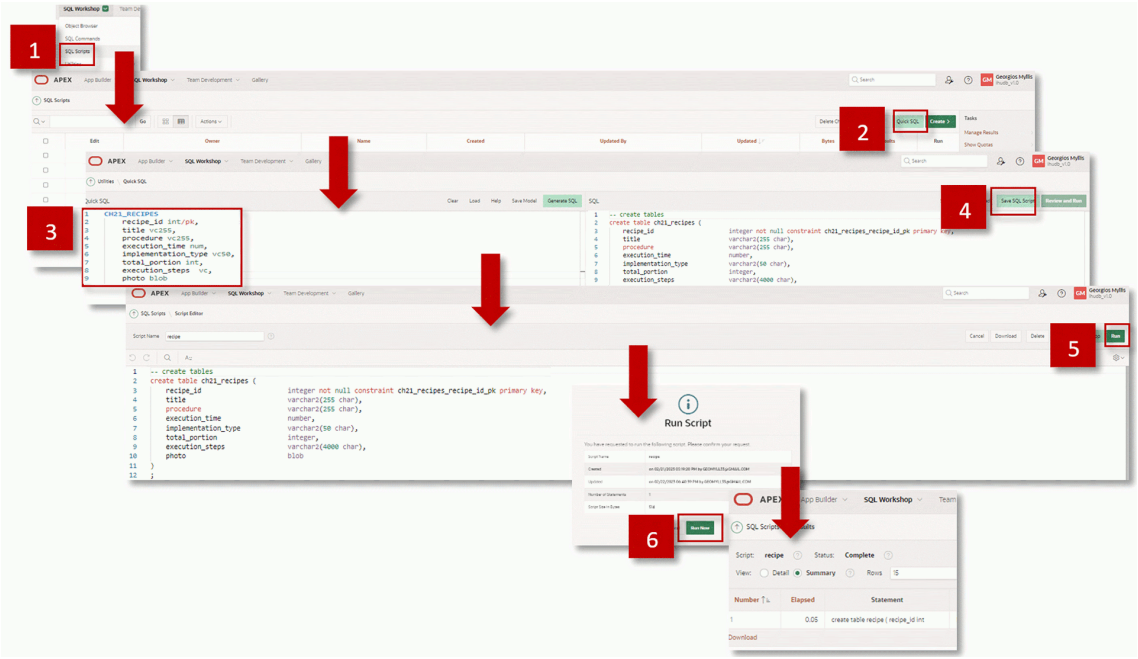
```
CH21_RECIPES
RECIPE_ID int/pk,
TITLE VC255,
PROCEDURE VC255,
EXECUTION_TIME NUM,
IMPLEMENTATION_TYPE VC50,
TOTAL_PORTION int,
EXECUTION_STEPS  VC,
PHOTO BLOB

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
```

Na spustenie Quick SQL kódu je potrebné najskôr sprístupniť **SQL Workshop** a vybrať **SQL Scripts**. Odtiaľto si môžeme vybrať **Quick SQL** na vytvorenie tabuliek. Proces je jednoduchý, potrebuje len špecifikovať názov tabuľky, jej polia a taktiež aj prípadné dcérske tabuľky súvisiace s našou tabuľkou. Obrázok 21.4 zobrazuje ako rýchlo vytvoriť tabuľky využitím tejto funkcionality.



Obr. 21.4: Quick SQL procedure to create tables.

Našim nasledujúcim krokom pre správne fungovanie našich tabuliek je uistiť sa, že každé pridávanie do tabuľky bude unikátne a že primárny kľúč každej z tabuliek je automaticky inkrementovaný. Na to aby sme to dosiahli potrebujeme vytvoriť tri sekvencie.

Zvýšenie ID receptu

```
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;
```

Zvýšenie ID ingrediencie

```
CREATE SEQUENCE seq_INGREDIENTS
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;

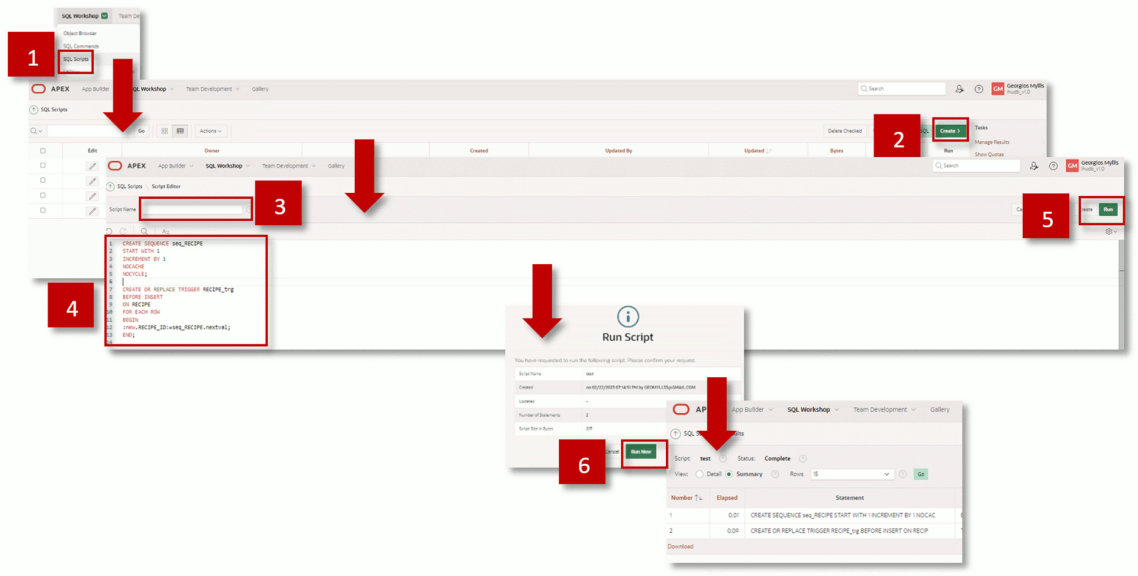
CREATE OR REPLACE TRIGGER INGREDIENTS_trg
BEFORE INSERT
ON INGREDIENTS
FOR EACH ROW
BEGIN
:new.INGREDIENTS_ID:=seq_INGREDIENTS.nextval;
END;
```

Zvýšenie ID ingrediencie receptu

```
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;
```

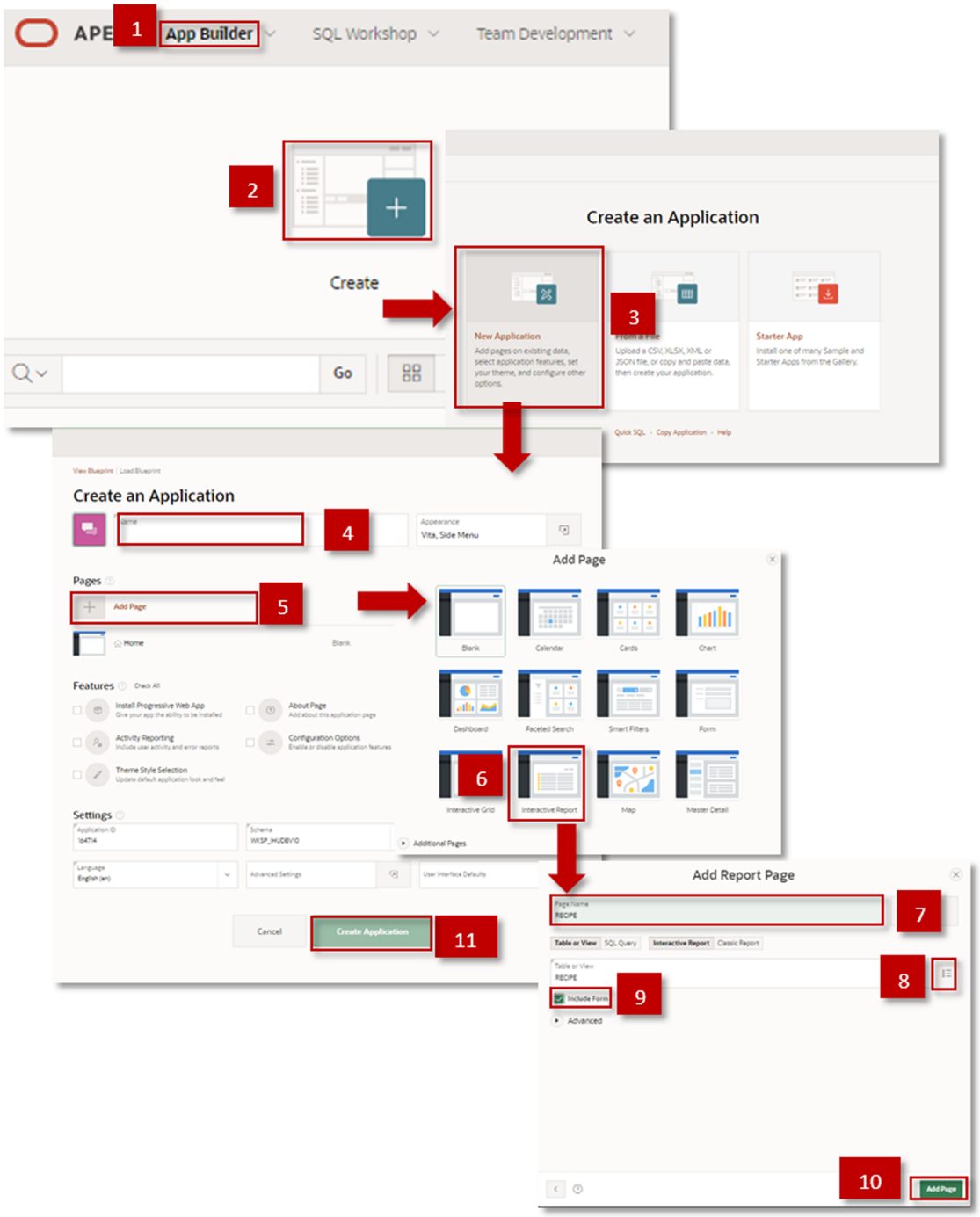
Pre spustenie sekvencie (SEQUENCE) je potrebné ísť do **SQL Workshop** a kliknúť na **SQL Scripts**. Potom je môžeme kliknúť na tlačidlo **Create** a skopírovať a vložiť vyššie uvedený kód pre každú sekvenciu samostatne. Po pomenovaní skriptu môžeme kliknúť na tlačidlo **Run**, na spustenie vykonávania skriptu. Toto zaistí, že každý záznam v tabuľkách bude jedinečné a eliminuje potrebue manuálne sledovať posledne pridaný záznam. Všetko sa bude diať automaticky. Príklad toho ako spustiť sekvenciu v APEX prostredí zobrazuje Obrázok 21.5.



Obr. 21.5: Procedúra s SQL skriptom na vytvorenie SEQUENCE.

21.5 Rozhrania aplikácie

Za účelom vytvorenia aplikácie ako je na Obrázku 21.6, kliknite na **App Builder** a následne kliknite na **Create**. Potom sa objaví nové okno, v ktorom musíme zadať názov našej aplikácie a vybrať funkcionality, ktoré v našej aplikácii chceme. Tabuľky, ktoré budú v našej aplikácii použité sú: RECIPE, RECIPE INGREDIENTS, INGREDIENTS ktoré boli vysvetlené skôr.



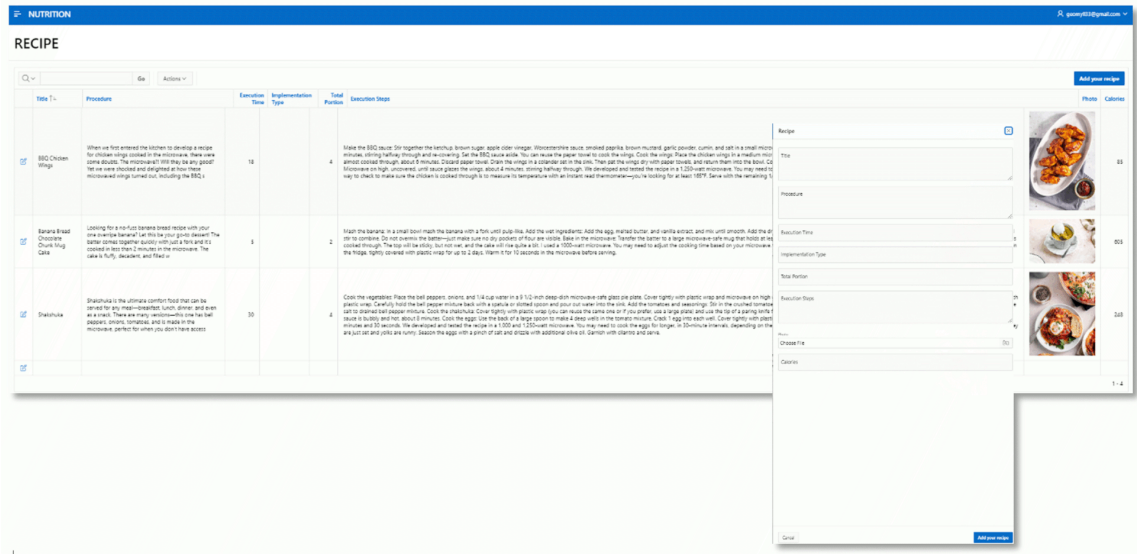
Obr. 21.6: Vytvorenie novej aplikácie.

Na Obrázku 21.7 je vytvorené nové App Builder prostredie pre vašu novú aplikáciu. Na prístup k vašej aplikácii kliknite a zadajte administrátorské prihlasovacie údaje, ktoré by mali byť rovnaké pre workspace. Keď ste prihlásení, môžete sa navigovať vo vašej aplikácii s využitím mega menu, ktoré sa nachádza na vrchu.



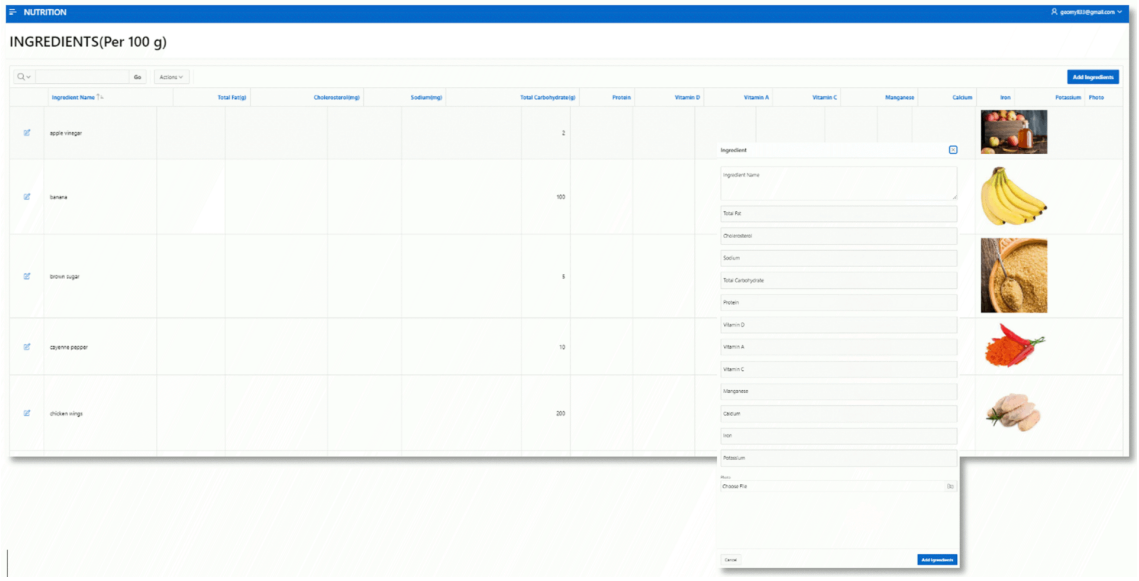
Obr. 21.7: Nutričná aplikácia.

Na Obrázku 21.8 je zobrazená stránka s receptami. Stránka poskytuje používateľovi všetky potrebné informácie týkajúce sa prípravy receptu ako je postup, čas varenia, celková porcia, kroky varenia a fotografiu požadovaného výsledku. Na základe užívateľských oprávnení môže používatelia môžu taktiež pridávať nové recepty alebo aktualizovať existujúce tak, aby sedeli jeho preferenciám.



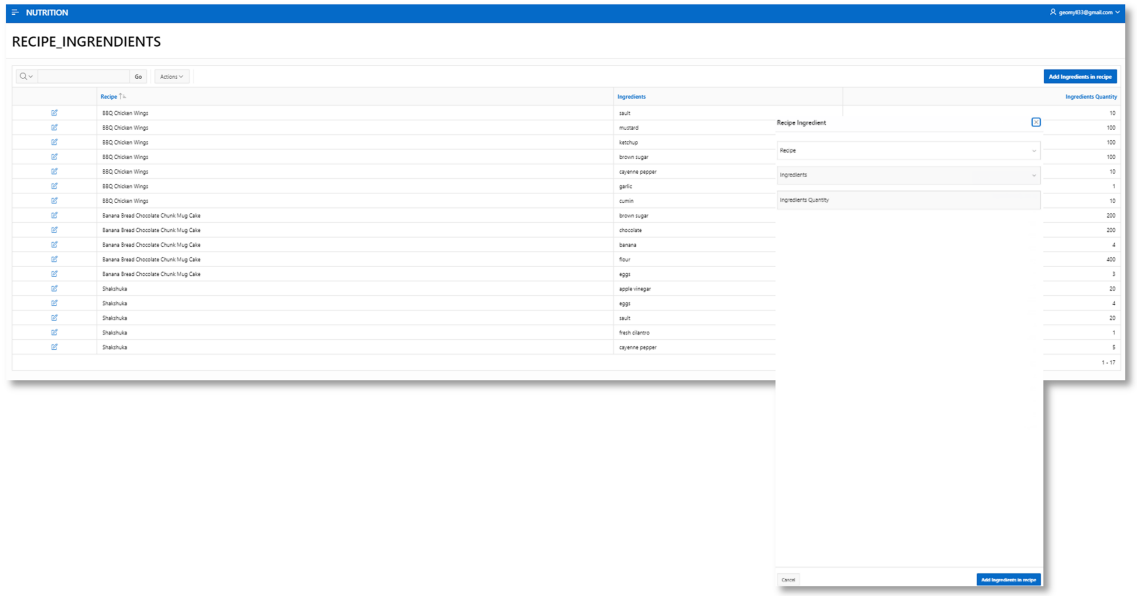
Obr. 21.8: Nutričná aplikácia - stránka receptu.

Na Obrázku 21.9 sú znázornené atribúty ingrediencií ako celkový tuk, cholesterol, sodík, uhľohydráty, bielkoviny, atď. Tieto atribúty sú použité pri výpočte nutričných hodnôt jedál v receptoch. Používateľ s potrebnými právami môže udržiavať stránku pridávaním nových ingrediencií alebo aktualizáciou existujúcich, ak je to potrebné.



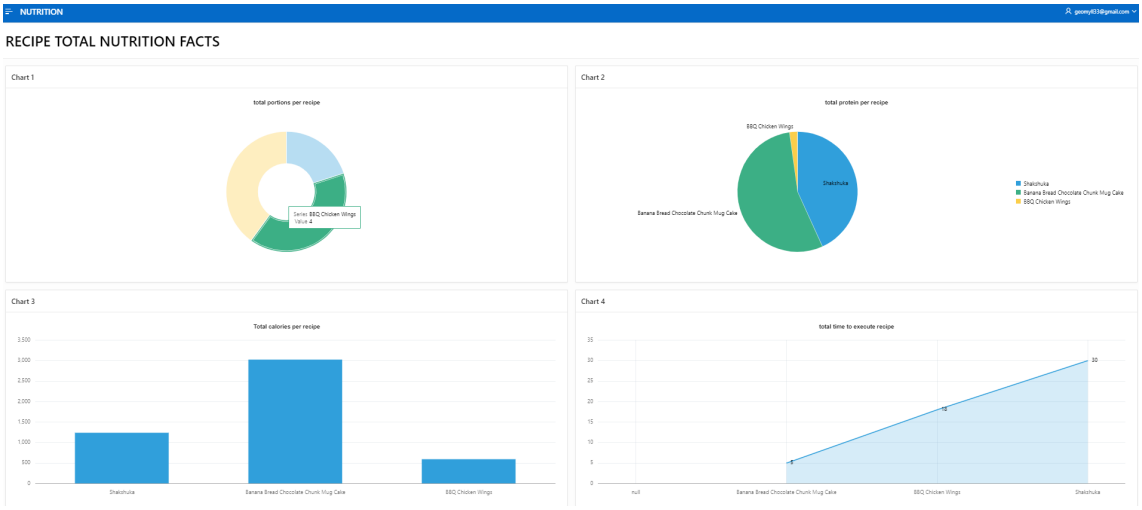
Obr. 21.9: Nutričná aplikácia - stránka ingrediencií.

Obrázok 21.10 zobrazuje prínos tejto webovej služby, nakoľko tu si môže užívateľ vytvárať recepty podľa jeho potrieb, prípadne meniť už existujúce recepty a informovať sa o ich výživovej hodnote. Ak má užívateľ potrebné oprávnenia, môže taktiež pridávať alebo odstraňovať ingrediencie z receptov.



Obr. 21.10: Nutričná aplikácia - stránka s ingredienciami receptov.

Na Obrázku 21.11 je zobrazená stránka s prehľadom nutričných hodnôt, ktorá poskytuje užívateľovi všetky potrebné informácie týkajúce sa receptu, ktorý zostavil, t.j. celkový obsah bielkovín, kalórií, porcií a času potrebného pre uvarenie.



Obr. 21.11: Nutričná aplikácia - stránka s informáciami o výživovej hodnote.

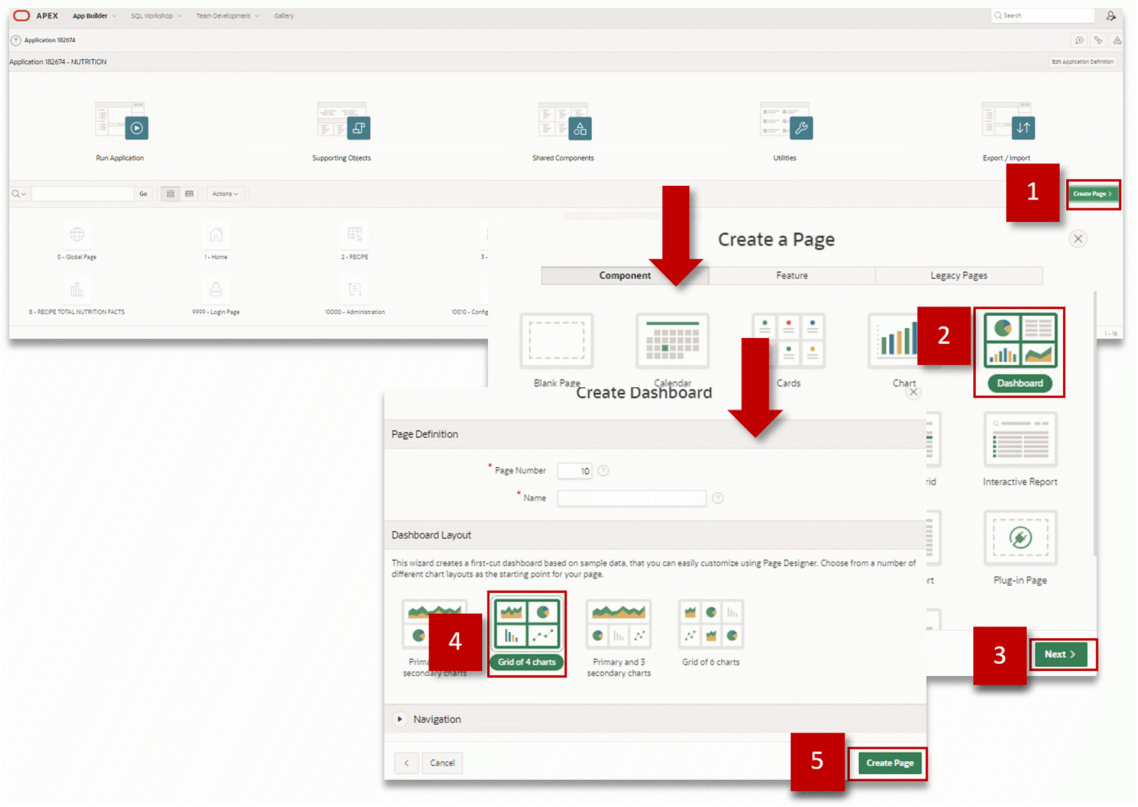
21.5.1 Ako vytvoriť report

Na vytvorenie dashboard stránky pre vašu aplikáciu postupujte podľa krokov zobrazených na 21.12.

- Kliknite na vaše aplikačné prostredie.
- Kliknite na tlačidlo **Create Page**.
- Vyberte **Dashboard** zo zoznamu dostupných typov stránok.
- Kliknite na tlačidlo **Next**.
- Zvoľte preferované rozloženie dashboardu.
- Kliknite na tlačidlo **Create Page** aby ste vytvorili dashboard stránku.

Tieto kroky vás prevedú cez proces vytvárania dashboard stránky pre vašu aplikáciu. Keď

ju vytvoríte, môžete dashboard upravovať pridávaní widgetov a konfigurovaním rozloženia na zobrazovanie dát, ktoré potrebujete k monitorovaniu a riadeniu vašej aplikácie.

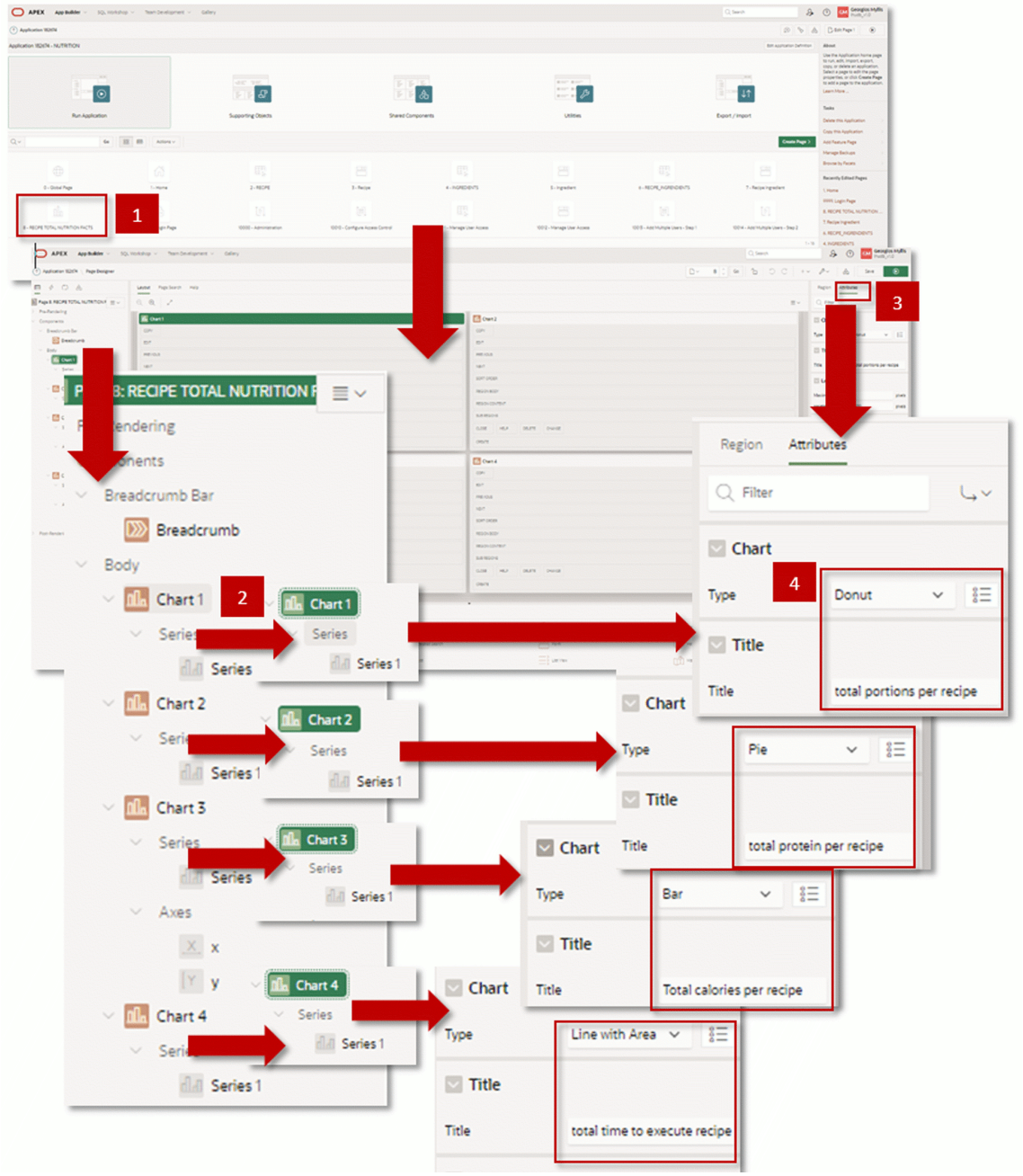


Obr. 21.12: Vytvorenie dashboardu pre aplikáciu.

Teraz sa navigujte do vášho dashboard reportu, ktorý môžete nájsť v aplikačnom prostredí a vstúpte do dizajnéra stránky. Ako je zobrazené na Obrázku 21.12, vybrali sme štyri grafy a do každého z nich pridáme rozličné údaje pomocou SQL dotazov.

Na začiatok vyberieme Chart 1 a premenujeme to na “Total Portion per Recipe”. Potom vyberieme **Attribute** a vyberieme Donutový typ grafu. Ďalej vyberieme Chart 2 a premenujeme ho na “Total Protein per Recipe”, ktorý zobrazuje recepty s najviac bielkovinami. Pre tento graf vyberieme typ **Pie Chart**.

Prejdime na Chart 3, premenujte to na “Total Calories per Recipe”. Tento graf bude zobrazovať celkový počet kalórií pre každý recept a v tomto prípade použijeme typ **Bar Chart**. Na záver premenujeme Chart 4 na “Total Time to Execute Recipe”. Vyberieme **Attributes** a potom zvolíme **Line with Area** ako typ grafu (pozri Obrázok 21.13).



Obr. 21.13: Konfigurácia názvu a typu grafu v dashboard reportoch.

Následne kliknite na **Series 1** v **chart 1** v “Source” zadajte nasledujúce nastavenia:

```
SELECT DISTINCT RECIPE.TITLE,
RECIPE.TOTAL PORTION
FROM RECIPE
```

V “Series 1” v “chart 2” zadajte do “Source” nasledujúce nastavenia:

```
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
```

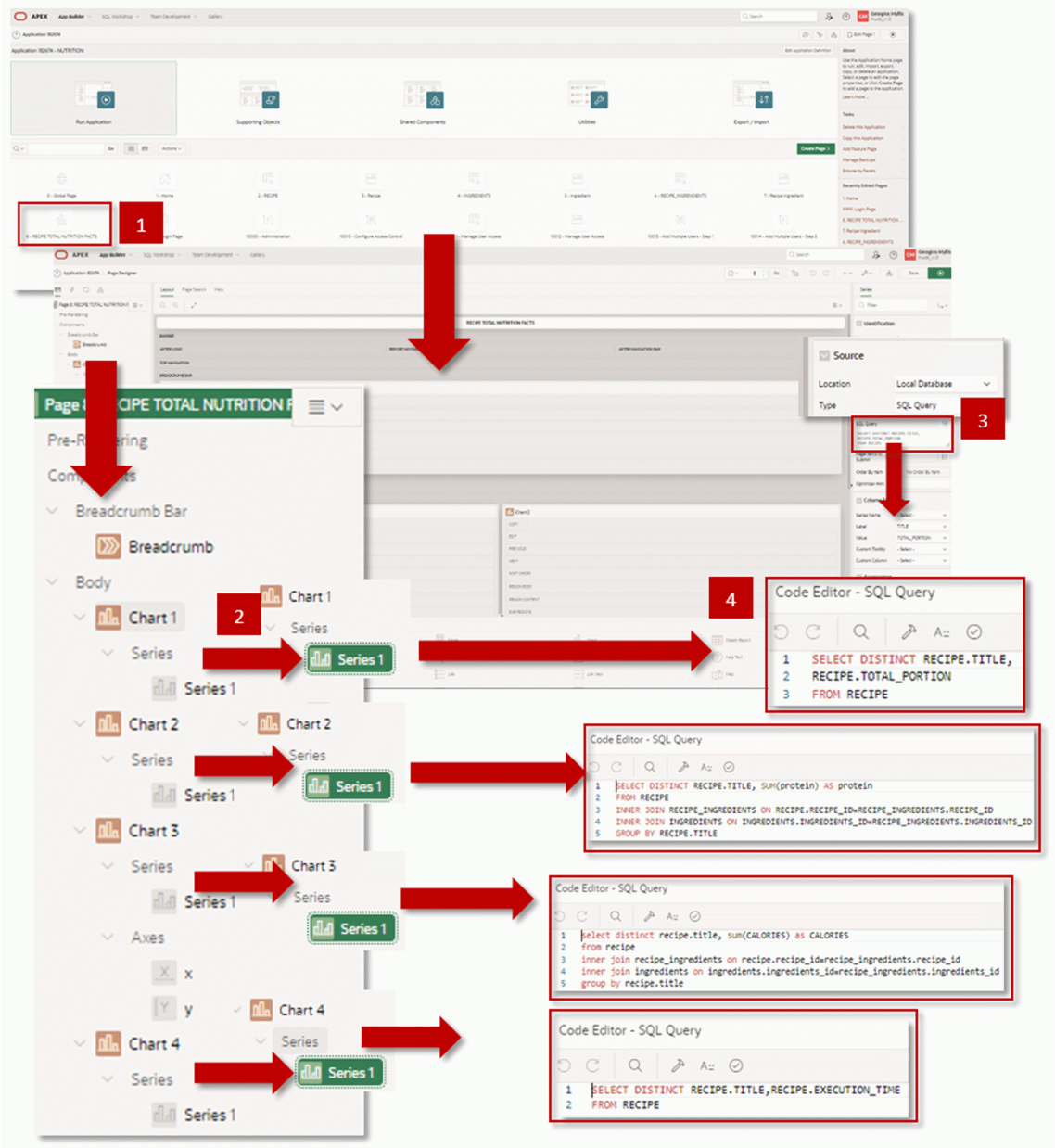
V “Series 1” v “chart 3” zadajte do “Source” nasledujúce nastavenia

```
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
```

a do “Series 1” v “chart 4” zadajte do “Source”:

```
SELECT DISTINCT RECIPE.TITLE,RECIPE.EXECUTION_TIME
FROM RECIPE
```

Obrázok 21.14 zobrazuje kroky konfigurácie.

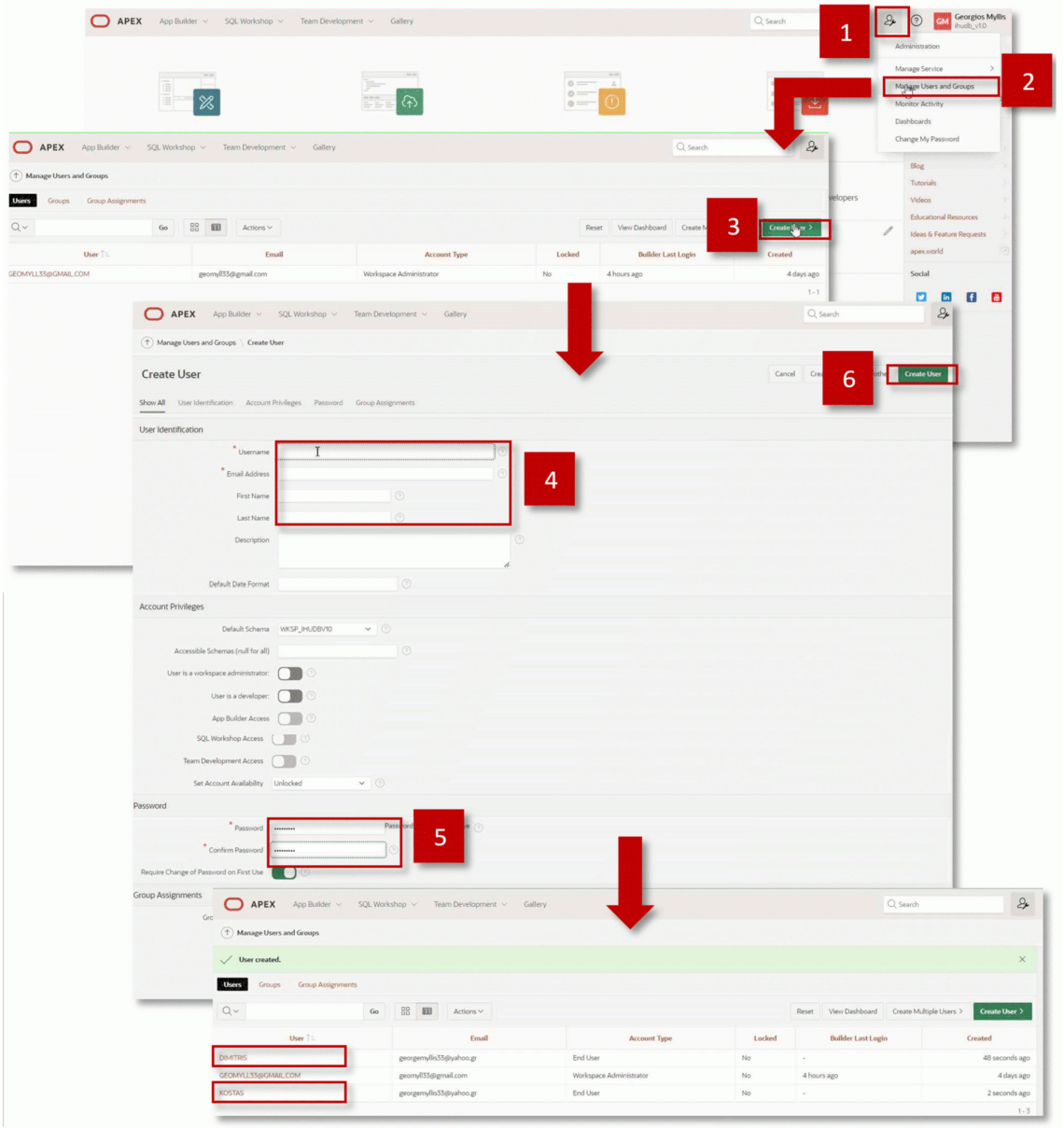


Obr. 21.14: Konfigurácia SOURCE s SQL QUERY.

21.5.2 Vytvorenie prístupu pre používateľa

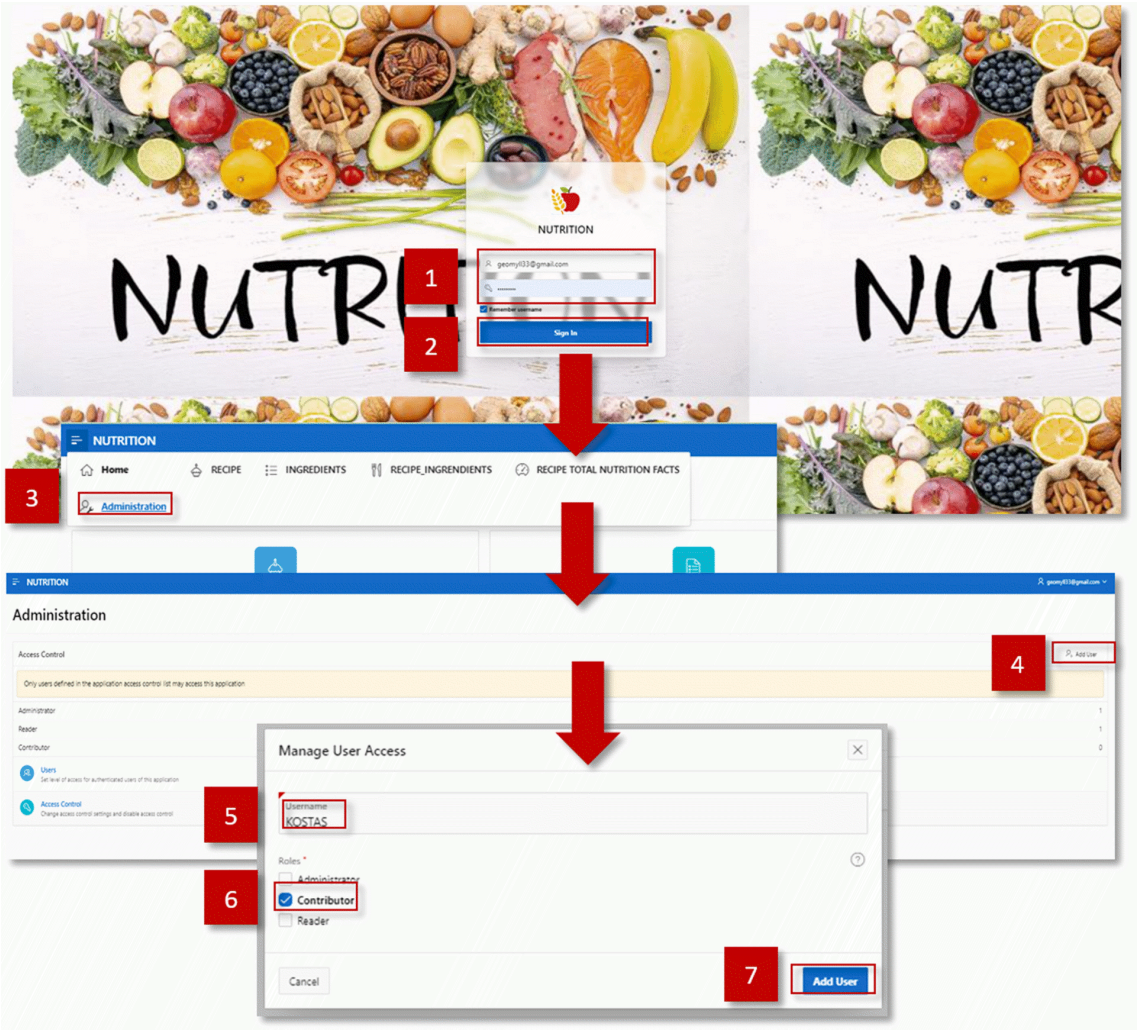
Oracle APEX poskytuje 3 oprávnenia, pomocou ktorých je možné riadiť prístup do aplikácie alebo jej komponentov. Tieto oprávnenia sú administrácia, úprava a zobrazenie. Každé z týchto oprávnení súvisí s rolou prístupu. Administrácia súvisí s administrátorskou rolou, úprava s rolou prispievateľa a zobrazenie s rolou čitateľa.

Pre pridelenie prístupových práv v aplikácii, musíte najskôr vytvoriť používateľský účet. Aby ste to spravili, choďte na “Administration” ikonu na vrchnej lište, ako je zobrazené na Obrázku 21.15, reprezentovaná postavou a symbolom kľúča a vyberte **Manage Users and Groups** z dropdown menu. Potom vyberte **Create User**, zadajte používateľskú identifikáciu a heslo a potom kliknite na **Create User** aby ste dokončili proces.



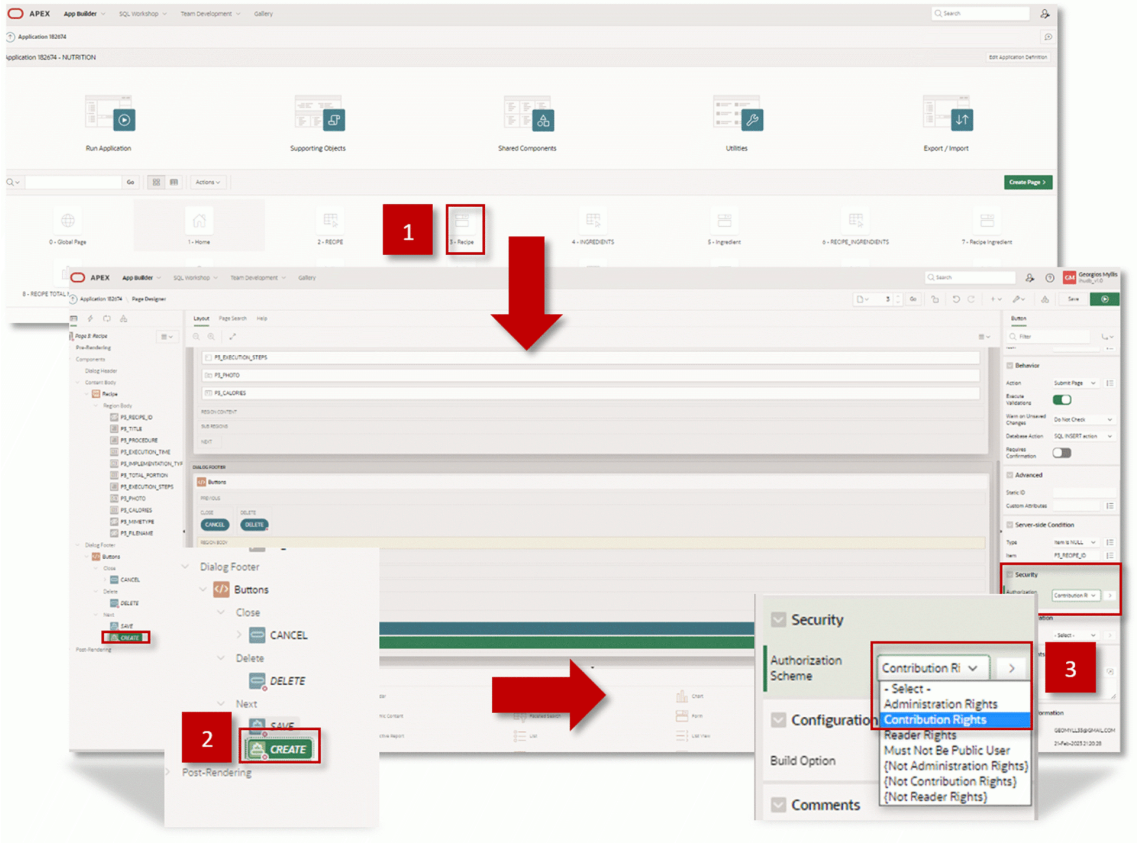
Obr. 21.15: Vytvorenie užívateľského účtu.

Jedným zo spôsobov, ako definovať roly užívateľov, je urobiť to prostredníctvom aplikácie. Prihláste sa do aplikácie s oprávneniami správcu, choďte do menu a kliknite na **Administration**. Odtiaľ kliknite na **Add User**, vložte užívateľské identifikátor a rolu a kliknite na **Add User**. Ak potrebujete podrobnejší návod, môžte postupovať podľa Obrázku 21.16.



Obr. 21.16: Vytvorenie užívateľského účtu.

Následne, tak ako je zobrazené na Obrázku 21.17, choďte do prostredia aplikácie a kliknite na **RECIPE** formulár. V návrhárovi stránky kliknite na **Create** a v možnosti zabezpečenia vyberte **Contribution Rights** ako autorizačnú schému. Následne kliknite na **Save**. Toto zaistí, že iba používatelia s rolou prispievateľa budú môcť vytvárať a pridávať recepty.



Obr. 21.17: Konfigurácia stránok s užívateľskými oprávneniami.

Všetky doplnkové výučbové materiály sú dostupné na public BeeAPEX project page. Prihláste sa ako hosťovský používateľ (nevyžaduje heslo). Nájdete učebnicu v sekcii Books, skripty v priečinku Part 2 > Chapter24 v sekcii Scripts a video návod v Collection of video guides. Materiály pre krátke kurzy sa nachádzajú v sekcii Short courses.

21.5.3 Exportovaná aplikácia

Exportovaná aplikácia je zabalená. Inštalácia vytvorí tabuľky, indexy, funkcie, procedúry a triggre a taktiež ich naplní dátami. Odinštalovanie odstráni všetky databázové objekty použité v tejto aplikácii.

21.5.4 Video návody

Video návod ukazuje každý krok tvorby aplikácie.

21.6 Otázky

1. Ako je možné zmeniť pozadie na prihlasovacej stránke?
2. Kde je možné nájsť správu rolí v prostredí Oracle Apex, ak nie je povolená v aplikácii?
3. Ako je možné nakonfigurovať názov tlačidla zostavy?
4. Ako je možné zmeniť názov stránky po jej vytvorení?

21.7 Odpovede

1. V prostredí aplikácie vyberte **Shared Components**, následne **Static Application Files**, potom kliknite na **Create File** a vo vyskakovacom okne pridajte obrázok podľa svojich

predstáv. Potom skopírujte referenčný názov obrázka, vráťte sa do prostredia aplikácie a vyberte **9999-Login Page**, prejdite do poľa css a zadajte

```
body {
background-image:
url("#APP_FILES#\textit{the reference name of the image}");
background-size:1700px 900 px:
}
```

2. V prostredí aplikácie zvoľte **Shared Components**, potom kliknite na **Application Access control**.
3. Aby ste zmenili názov tlačidla **Create**, musíte spustiť návrhára stránok (page designer) pre reporty, lokalizovať **Create** tlačidlo a následne zmeniť označenie v identifikačnom poli.
4. Pre zmenu názvu vytvorenej stránky je potrebné sa prihlásiť do aplikácie, kliknúť na **Quick edit** a názov stránky. Následne kliknite navigačnú šípku v zdrojovej oblasti. Potom kliknite na **Edit Component**, v **Short Name** políčku aktualizujte na požadované meno. Na záver kliknite na **Apply Changes** a nový názov sa zobrazí v názve stránky.



22. Plánovanie konzultačných hodín

J. Mańko, M. Sońta, R. Leskovar

22.1 Biznis pohľad prípadu

Univerzitnú aplikáciu na rezerváciu konzultačných hodín možno z podnikového hľadiska považovať za nástroj na zefektívnenie a optimalizáciu procesu plánovania pre zamestnancov univerzity. Môže pomôcť zlepšiť efektivitu plánovania, znížiť administratívnu záťaž, zvýšiť spokojnosť študentov a poskytovať informácie o dostupnosti v reálnom čase. Aplikácia môže tiež generovať údaje a poznatky o vzoroch a využívaní schôdzok, ktoré môžu byť podkladom pre rozhodovanie a prideľovanie zdrojov. Okrem toho môže aplikácia ponúknuť pohodlnú a dostupnú rezervačnú platformu pre študentov, čo pomôže podporiť aktívnejšiu a angažovanejšiu študentskú komunitu. Každý týždeň sú akademickí pracovníci povinní poskytnúť študentom 60 minút svojho času, ktorý sa interne nazýva "úradné/konzultačné hodiny". Okrem toho sa to stáva pomerne často, najmä s blížiacim sa koncom semestra, keď sa zintenzívňujú otázky od študentov čo do počtu aj rozsahu. V takomto prípade 60 minút týždenne nestačí, pretože študenti sú odhodlaní čo najskôr získať od akademických pracovníkov naliehavo požadovaný výsledok. Aj keď existujú štandardizované témy na diskusiu, ako napríklad: diplomové práce, skúšky, záverečné zadania, niektoré prípady sa však týkajú nepodstatných tém, ktoré by sa možno mali prediskutovať, ale nie nevyhnutne počas konzultačných hodín s učiteľom (napríklad: administratívne požiadavky alebo problémy, ktoré sa dajú ľahko zvládnuť bez zásahu učiteľa).

V posledných rokoch, najmä v dôsledku pandémie COVID-19, prešli univerzity radikálnou technologickou transformáciou, ktorá umožnila rôzne formy výučby na diaľku s využitím dostupných platforiem a komunikátorov, ktoré uľahčili online vyučovací proces. V skutočnosti sa prakticky všetky akademické aktivity preniesli do online prostredia, prirodzene vrátane konzultačných hodín. Aj keď sa akademická obec zhoduje, že dištančná výučba nikdy plne nenahradí výučbu na pracovisku, vo svete sa naďalej v určitej miere realizuje výučba a dokonca aj práca na diaľku. Ešte dôležitejšie je, že niektoré univerzitné činnosti a procesy už prešli alebo v blízkej budúcnosti úplne prejdú do online režimu, napríklad s cieľom zabezpečiť efektívny bezpapierový obeh dokumentov. Tieto okolnosti ešte viac posilňujú naliehavú potrebu životaschopného digitálneho nástroja, ktorý by umožnil online rezerváciu konzultačných hodín, ktorá by mohla prebiehať buď online, alebo offline. Je zrejmé, že celá akademická komunita by z takejto aplikácie, ktorá by sa čoskoro mohla stať každodenným nástrojom, ktorý by používali všetci, ako napríklad e-mailová schránka, a nie nejaký digitálny výstrelok, mala veľký úžitok.

22.2 Definícia problému

V súčasnosti nie je možné vopred rezervovať (zapísať) časový interval a naplánovať obsah stretnutia. V skutočnosti sú študenti, ak sa dostavia v akomkoľvek počte, vybavení na základe zásady "kto prv príde, ten prv melie". Nielenže to zvyšuje frustráciu študentov, ktorí musia čakať v rade bez akejkoľvek istoty, že sa ich stretnutie vôbec uskutoční, ale rozčuľuje to aj akademických pracovníkov, ktorí sa musia rozhodnúť, či nechajú niektorých študentov čakať zbytočne alebo preložia svoj plán na daný deň. V dôsledku toho dochádza k situáciám, keď akademickí pracovníci fakticky poskytujú študentom poradenstvo pro bono alebo dokonca na úkor svojho súkromného života, len aby ich študenti nevnímali ako nedostupných alebo nepriateľských. Okrem toho tento stav znamená neefektívne rozdelenie času pre študentov aj akademických pracovníkov, ako aj ich zbytočné zaťaženie pri plánovaní práce.

Táto administratívna systémová neefektívnosť núti akademických pracovníkov k ďalším, často neplateným povinnostiam, pričom študenti si nie sú istí výsledkami a s najväčšou pravdepodobnosťou sú s celým procesom len málo spokojní. Ide o ukážkový príklad prenášania dôsledkov nefunkčného systému stretnutí na skupiny, ktoré sa nachádzajú nižšie v akademickej hierarchii (študenti a akademickí pracovníci), namiesto toho, aby sa našlo riešenie zhora nadol iniciované vedením univerzity. Technológia tu preto ponúka životaschopné, posilňujúce a realizovateľné riešenie, ktoré by výrazne odbremenilo akademických pracovníkov a zároveň zabezpečilo vyššiu spokojnosť a efektívnosť využívania pre študentov.

Študenti sa často obracajú na študijné oddelenie s otázkami o termínoch a miestach konzultačných hodín. Keďže neexistuje žiadny spoľahlivý zdroj, spoliehajú sa na skúsenosti a často študentov nasmerujú na nesprávne miesta v nesprávnom čase. Podľa pravidiel akademickej inštitúcie by každú zmenu konzultačných hodín mal schváliť prodekan. Schválenia sa udeľujú v priamej komunikácii medzi učiteľom a prodekanom a študijné oddelenie o tom nemá žiadne informácie. Prodekan si preto želá, aby študijné oddelenie zapisovalo zmenené konzultačné hodiny.

Vedenie má málo informácií o celkovom obraze konzultačných hodín v akademickej inštitúcii. Fámy a iné nespoľahlivé informácie nie sú dobrým základom pre racionálne rozhodnutia o zvýšení alebo znížení počtu týchto hodín. Môže to súvisieť aj s odmeňovaním učiteľov za ich mimoriadne úsilie.

22.3 Prípady použitia

Aplikácia na rezerváciu konzultačných hodín umožňuje prístup vedeniu, zamestnancom študijného oddelenia, študentom a akademickým pracovníkom (označovaným aj ako učitelia), pričom každý z nich má iné oprávnenia:

- Vedenie získa súhrnné informácie zhromaždené na dashboard-e (počet konzultačných hodín podľa rokov a mesiacov, prvých 10 učiteľov podľa počtu študentov, prvých 5 učiteľov s nadmerným počtom študentov a prvých 5 učiteľov bez študentov počas konzultačných hodín).
- Zamestnanci študijného oddelenia majú prístup ku konzultačným hodinám pomocou interaktívneho reportu, pričom môžu meniť termín konzultačných hodín.
- Študenti majú prístup ku kalendáru svojich stretnutí, kalendáru všetkých konzultačných hodín, ktoré ponúkajú učitelia vrámci študijného programu študenta, môžu sa zapísať na konzultácie prostredníctvom kalendára a majú k dispozícii aj podrobný interaktívny prehľad o svojich konzultačných hodinách.
- Akademickí zamestnanci/učitelia majú k dispozícii podrobný interaktívny prehľad o konzultačných hodinách učiteľa, kalendár s konzultačnými hodinami učiteľa a môžu zmeniť termín konzultačných hodín, na ktoré ešte nie sú zapísaní študenti.

Tabuľka 22.1: Popis prípadu použitia: preloženie konzultačných hodín učiteľom.

Kľúčové slovo	**Hodnota**
ID:	*Ch22-01*
Nadpis:	*Preplánovanie konzultačných hodín učiteľom*
Popis:	*Učiteľ využíva APEX aplikáciu na preplánovanie konzultačných hodín, tam kde nie je zapísaný žiadny študent.*
Hlavný aktér:	*Učiteľ*
Predpoklady:	Prístup k webovému prehliadaču na mobilnom zariadení alebo osobnom počítači, používateľ ma učiteľské prihlasovacie údaje a prístupové práva, stránka aplikácie je dostupná.
Podmienky po skončení:	dáta uložené v databáze v tabuľke ch22_duty_hours
Hlavné	*Scenáre*
Úspešný scenár:	1. Učiteľ otvorí webový prehliadač a prihlási sa do aplikácie. 2. Vyberte preplánovanie konzultačných hodín tam, kde nie je nikto zapísaný. 3. Kliknite na slot označujúci, že nikto nie je zapísaný. 4. Do formulára zadajte preplánované údaje (lokácia, začiatočný a koncový čas) 5. Skontrolujte a potvrďte preplánovanie kliknutím na vhodné tlačidlo.
Rozšírenia:	-
Frekvencia použitia:	*Približne 4 z 30 povinných dátumov v jednom akademickom roku.*
Stav:	*Dokončené*
Vlastník:	*Používateľ s prístupovými právami učiteľa*
Priorita:	*Stredná*

22.3.1 Naratívny popis

Pre stručnosť uvedieme len dva prípady použitia:

- Preloženie konzultačných hodín učiteľom, na ktorých nie je zapísaný žiadny študent, si vyžaduje, aby sa učiteľ prihlásil, prezrel naplánované konzultačné hodiny, na ktorých nie je zapísaný žiadny študent, a "presunul" ich po jednom na iné termíny.
- Zápis študenta na konzultačné hodiny pozostáva z prehľadu kalendára (všetky konzultačné hodiny, ktoré ponúkajú všetci učitelia zapojení do študijného programu študenta), výberu požadovaných konzultačných hodín, zadania účelu a potvrdenia zápisu. Nie je povolené zapísať sa viac ako jedenkrát na plánované konzultačné hodiny.

22.3.2 Pološtruktúrovaný popis

Prvým prípadom použitia je preloženie konzultačných hodín učiteľom, keď sa nezapíše žiaden študent (viď tabuľka 22.1). Preloženie na iný termín obmedzujeme na tento typ, pretože sa netýka žiadneho študenta. Druhý prípad použitia je zameraný na zápis študentov na konzultačné hodiny (viď tabuľka 22.2). Aplikácia by tiež mala zabrániť viacnásobnému zápisu na jednu konzultačnú hodinu.

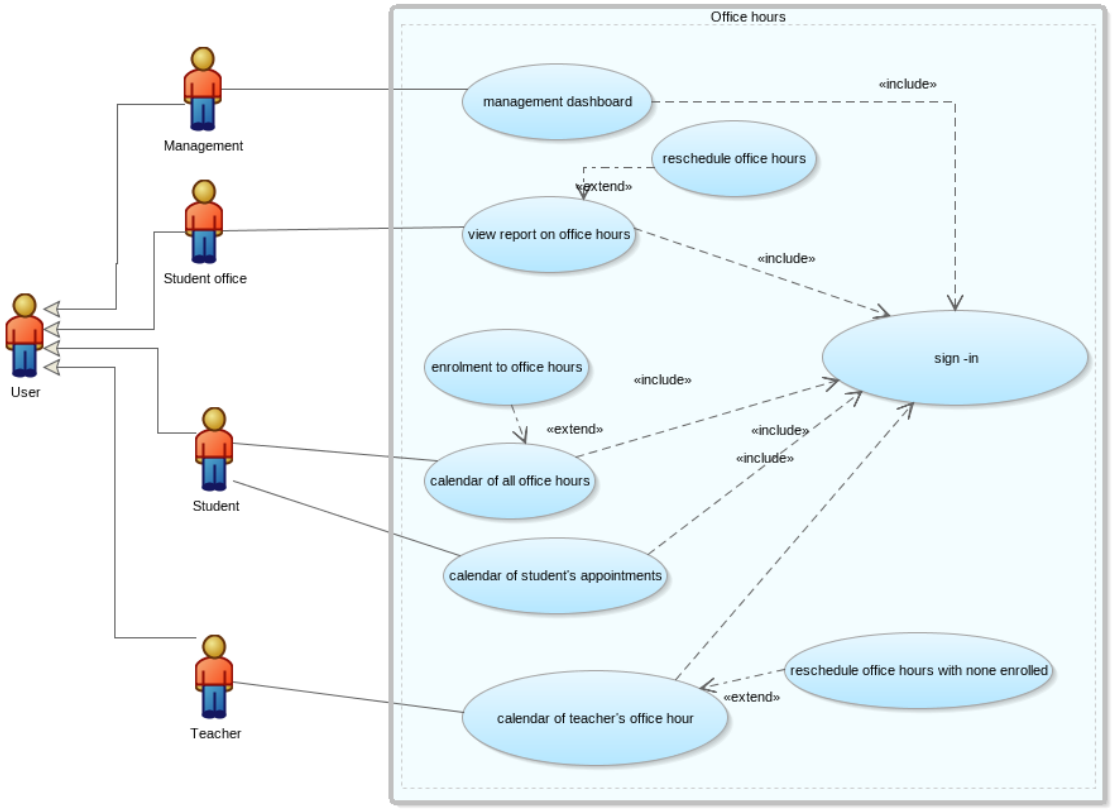
22.3.3 Diagram prípadu použitia

Obrázok 22.1 zobrazuje identifikované prípady použitia. Všimnite si, že prípad použitia "sign-in (prihlásenie)" sa vykonáva len raz za reláciu používateľa a že žiadny z ostatných prípadov použitia sa nemôže vykonať pred úspešným dokončením "sign-in". Tri prípady použitia, a to "reschedule

Tabuľka 22.2: Popis prípadu použitia: zápis študenta na konzultačné hodiny.

Kľúčové slovo	**Hodnota**
ID:	*Ch22-02*
Nadpis:	*Zápis študenta na konzultačné hodiny*
Popis:	*Študent využíva aplikáciu na zapísanie konzultačných hodín. Kalendár má rozvrhnuté konzultačné hodiny pre všetkých učiteľov, ktorí sa zapájajú do štúdijných programov študenta (katedry). Zvyčajne je jeden študent zaradený do jedného študijného programu ale existujú výnimočný jedinci, ktorým je umožnené študovať dva alebo viac. Aplikácia zabráni viacnásobným zaradeniam jedného študenta na špecifické konzultačné hodiny.*
Hlavný aktér:	*Študent*
Predpoklady:	Prístup k webovému prehliadaču na mobilnom zariadení alebo osobnom počítači, používateľ ma učiteľské prihlasovacie údaje a prístupové práva, stránka aplikácie je dostupná.
Podmienky po skončení:	dáta uložené v databáze v tabuľke ch22_dh_slot
Hlavné	*Scenáre*
Úspešný scenár:	1. Študent otvorí webový prehliadač a prihlási sa do aplikácie. 2. Zapíšte sa cez kalendár. 3. Kliknite na požadovaný slot (sloty sú vyfarbené aby indikovali stavy). 4. Na novom formuláre študent klikne na tlačidlo a potvrdí zapísanie.
Rozšírenia:	-
Frekvencia použitia:	*Cez 1000 študentov využíva konzultačné hodiny v priemere 2 krát za semester u 10 až 15 učiteľov.*
Stav:	*Dokončené*
Vlastník:	*Študent*
Priorita:	*Vysoká*

office hours", "enrollment to office hours" a "reschedule office hours with none enrolled", sú závislé a označené asociáciou "extend". Tieto prípady použitia sa vykonajú na požiadanie po dokončení "rodičovského" prípadu použitia.



Obr. 22.1: Diagram prípadu použitia.

22.4 Dátový model

22.4.1 Naratívny popis dátového modelu

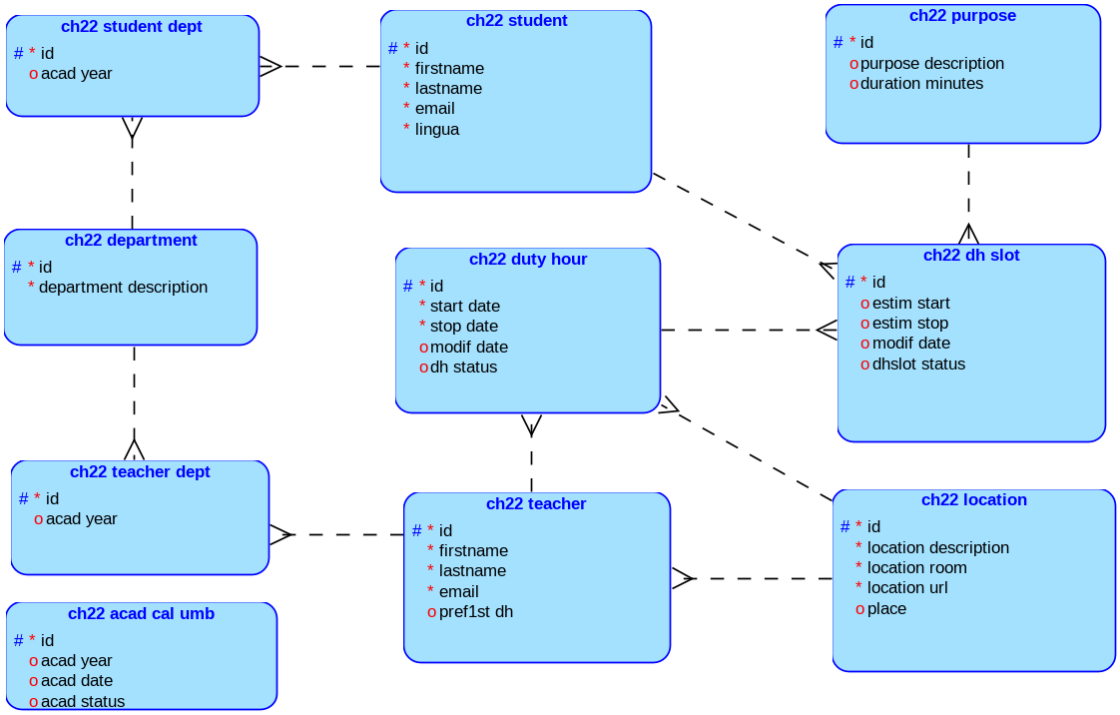
V logickom dátovom modeli je desať entít:

- **ch22_dh_slot** predstavuje jedno stretnutie študenta s učiteľom,
- **ch22_duty_hour** predstavuje vopred pridelený časový úsek učiteľa (konzultačné hodiny), ktorý poukazuje na niekoľko časových úsekov alias stretnutí študenta s učiteľom,
- **ch22_purpose** ukladá dôvody a priemernú dĺžku trvania stretnutí študentov s učiteľmi,
- **ch22_student_dept** poukazuje na jeden študijný program jedného študenta, Výnimoční študenti môžu byť zapísaní na viac ako jeden program (synonymum pre odbor) v jednom akademickom roku,
- **ch22_student** je osoba, ktorá študuje na univerzite a je zapojená do študijného programu (synonymum pre katedru),
- **ch22_teacher_dept** poukazuje na jeden študijný program (tzv. katedru) jedného učiteľa. Jeden učiteľ sa zvyčajne venuje viac ako jednému študijnému programu,
- **ch22_teacher** je osoba, ktorá vedie akademické aktivity (vyučovanie, laboratórne cvičenia atď.),
- **ch22_location** je buď fyzická (R), alebo virtuálna (I), alebo oboje (B) - je to miesto, kde sa konajú konzultačné hodiny,
- **ch22_department** je synonymom pre študijný program,

- **ch22_acad_cal_umb** určuje pracovné/nepracovné dni. Táto entita súvisí so všetkými entitami pomocou niektorých dátumových atribútov, avšak tieto vzťahy nebudeme modelovať explicitne, pretože implicitný vzťah je dostatočný pre úplnosť modelu.

22.4.2 Logický dátový model

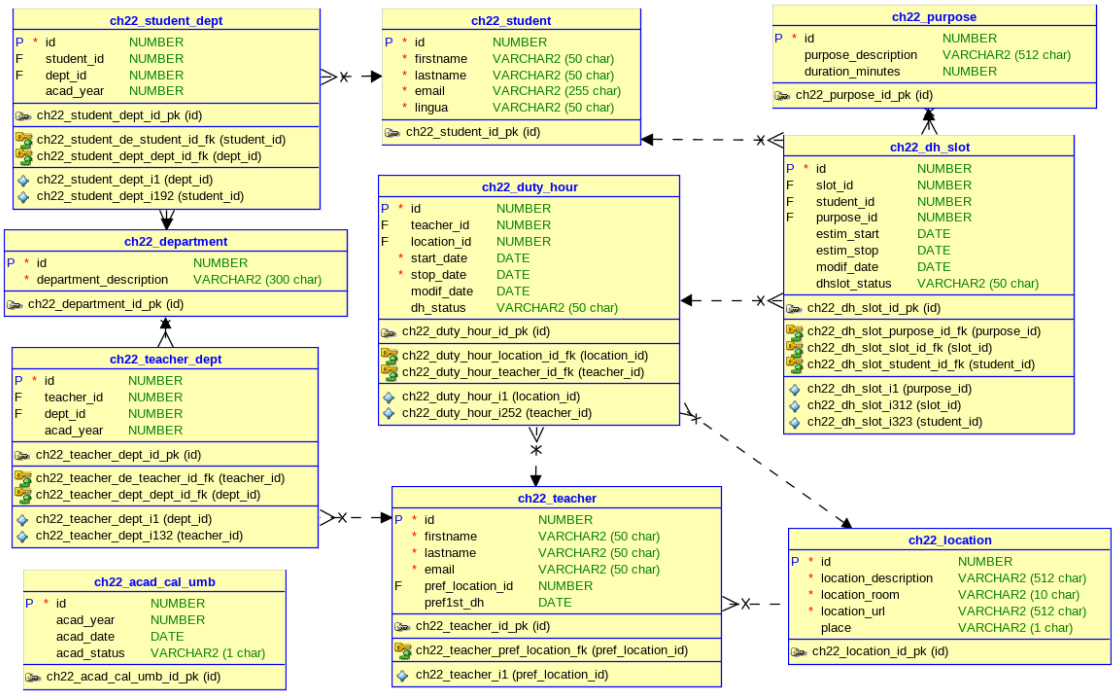
Logický dátový model je uvedený na Obrázku 22.2.



Obr. 22.2: Logický dátový model.

22.4.3 Relačný dátový model

Automatickú transformáciu z logického dátového modelu na relačný dátový model v Oracle SQL Data Modeler zabezpečuje funkcia *Engineering to relational*. Výsledok, relačný dátový model pripravený na export ako skript SQL, je znázornený na Obrázku 22.3.



Obr. 22.3: Relačný dátový model.

Relačný dátový model je možné vytvoriť aj bez aplikácie SQL Data Developer Data Modeler. V APEX môžeme použiť Quick SQL.

22.4.4 Quick SQL

Zobrazený Quick SQL (Obrázok 22.4) obsahuje všetky potrebné fakty na generovanie SQL skriptu.

```
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
```

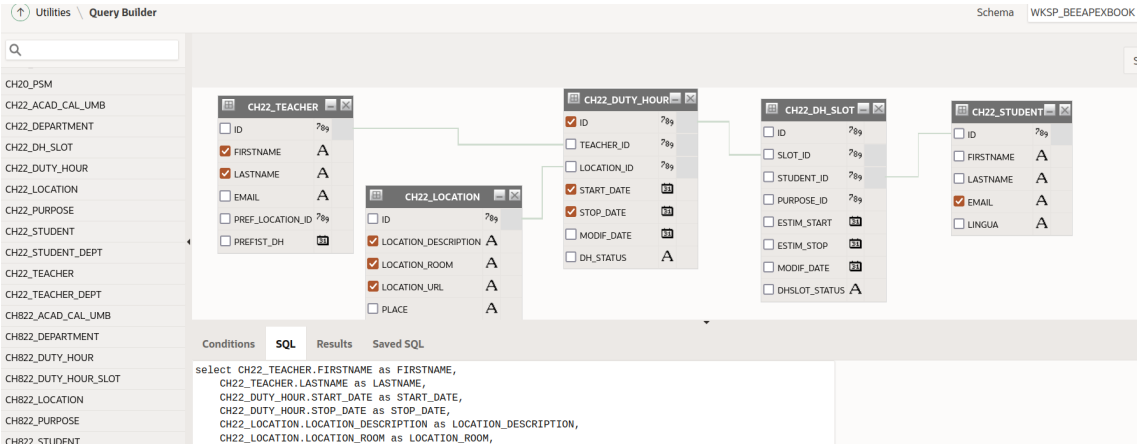
Obr. 22.4: Dátový model vytvorený pomocou Quick SQL.

22.4.5 SQL Skript na vytváranie tabuliek

SQL skript na vytvorenie tabuliek je uvedený v učebných materiáloch ako súbor CH22CREATE.sql. Skript na vytváranie tabuliek je k dispozícii aj v balenej aplikácii v učebných materiáloch.

22.4.6 Query builder v APEX

Táto konkrétna aplikácia bude používať pomerne dlhé queries, ktoré môžu začiatočníkov v systéme APEX vystrašiť. Existuje však riešenie - nástroj Query Builder, ktorý je zabudovaný v systéme APEX (SQL Workshop > Utilities > Query Builder). Tento nástroj pomôže každému zostaviť správne queries jednoduchým kliknutím a potiahnutím. Obrázok 22.5 uvádza príklad query na kalendár zapísaných študentov.



Obr. 22.5: Príklad použitia Query Builder.

Vygenerovaný príkaz SQL má nasledujúci tvar:

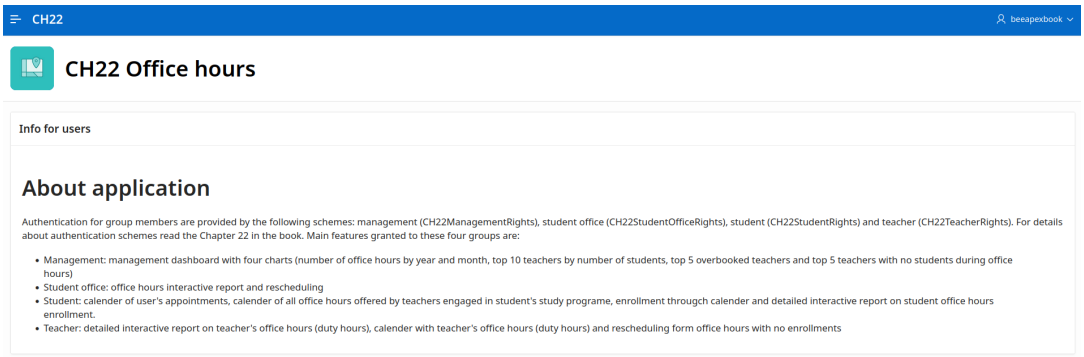
```
select CH22_TEACHER.FIRSTNAME as FIRSTNAME,
CH22_TEACHER.LASTNAME as LASTNAME,
CH22_DUTY_HOUR.START_DATE as START_DATE,
CH22_DUTY_HOUR.STOP_DATE as STOP_DATE,
CH22_LOCATION.LOCATION_DESCRIPTION as
LOCATION_DESCRIPTION,
CH22_LOCATION.LOCATION_ROOM as LOCATION_ROOM,
CH22_LOCATION.LOCATION_URL as LOCATION_URL,
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
```

Pridaním ďalšej podmienky na koniec “and CH22_TEACHER.EMAIL = :APP_USER” query vráti zapísaného študenta pre daného učiteľa. Nástroj Query Builder je potrebný pre úplných začiatočníkov, ale jeho používanie môžu využiť aj skúsení používatelia.

Queries použité v tejto aplikácii využívajú operátor spájania (||), funkciu EXTRACT, bežné tabuľkové výrazy, načítanie len prvých 10 alebo 5 riadkov, vlastnú funkciu zostavenia CH22_dh_utilization. Čitatelia tejto knihy môžu využiť, ak si najprv nainštalujú balenú aplikáciu a preskúmajú dotazy používané v kalendároch, na informačnom paneli a v zostavách.

22.5 Rozhrania aplikácie

Aplikácia má jednu bežnú stránku - Domovskú stránku. Táto stránka poskytuje základné informácie o aplikácii (viď Obrázok 22.6).



Obr. 22.6: Domovská stránka aplikácie.

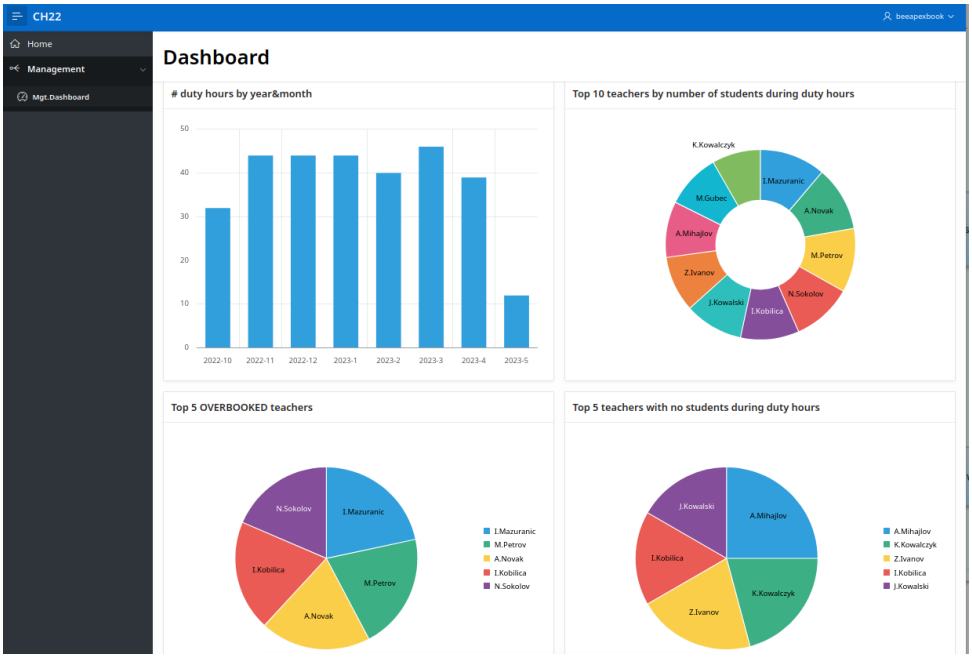
Manažment, zamestnanci študijného oddelenia, študenti a učitelia budú mať k dispozícii iba prispôsobené rozhrania.

22.5.1 Rozhrania aplikácie pre manažment

Používatelia s manažérskymi oprávneniami majú prístup k manažérskemu dashboard-u (viď Obrázok 22.7). Obsahuje grafy na podporu rozhodovania týkajúceho sa konzultačných hodín a učiteľov:

- počet konzultačných hodín podľa rokov a mesiacov,
- top 10 učiteľov podľa počtu študentov,
- top 5 najvyťaženejších učiteľov,
- top 5 učiteľov s najnižším počtom študentov na konzultačných hodinách.

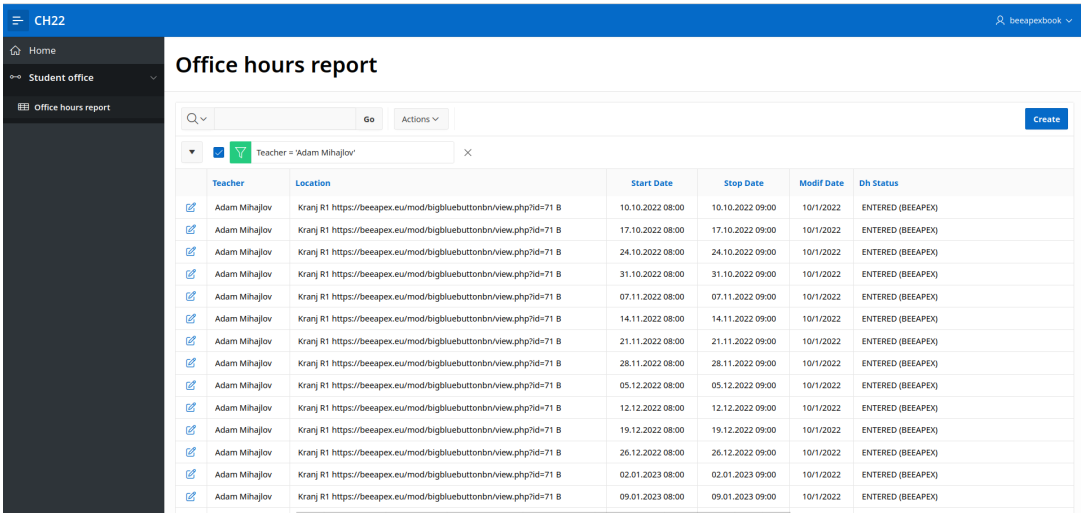
V súčasnosti nie sú definované žiadne ďalšie požiadavky z manažérskeho hľadiska.



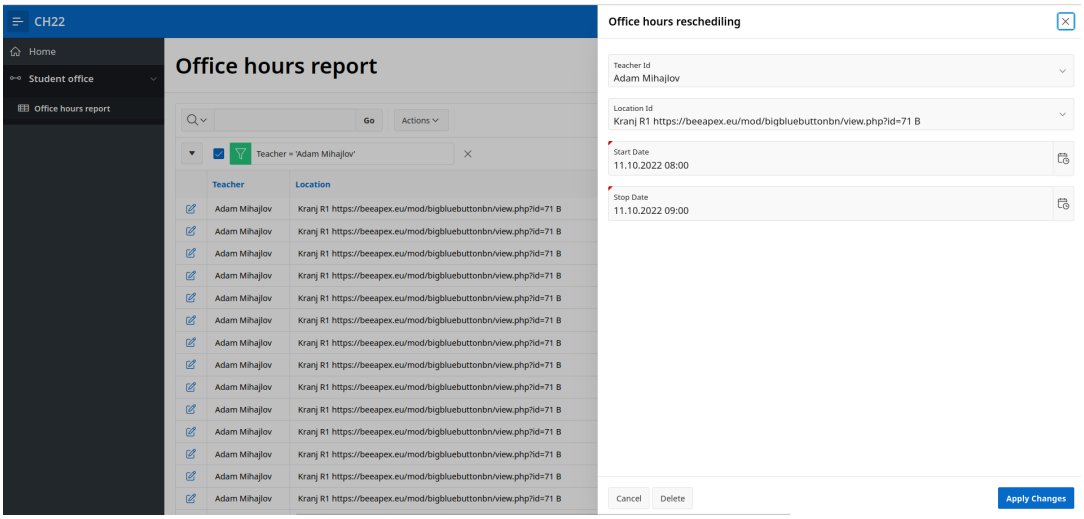
Obr. 22.7: Manažment dashboard.

22.5.2 Rozhrania aplikácie pre študijné oddelenie

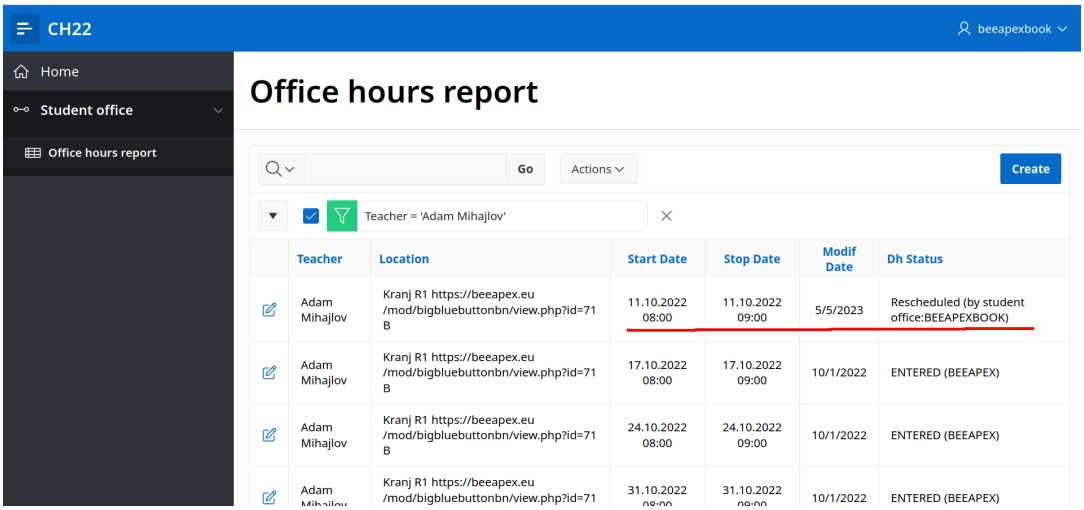
Používatelia s právami študijného oddelenia majú prístup k interaktívnemu hláseniu o konzultačných hodinách a môžu meniť plán konzultačných hodín. Obrázok 22.8 zobrazuje interaktívnu správu s aktivovaným filtrom (zobrazí sa len jeden učiteľ). Ďalej Obrázok 22.9 zobrazuje formulár, ktorý umožňuje zmenu plánu. Používateľ môže zmeniť učiteľa, miesto, dátum začiatku a konca vybraných konzultačných hodín. Obrázok 22.10 zobrazuje prehľad po zmene rozvrhu. Automaticky sa aktualizujú dve polia: dátum zmeny a stav konzultačných hodín. Neskôr je to spojený reťazec, ktorý obsahuje aj používateľské meno, ktoré vykonalo zmenu.



Obr. 22.8: Interaktívny prehľad o konzultačných hodinách pre študijné oddelenie.



Obr. 22.9: Formulár na zmenu termínu pre študijné oddelenie.

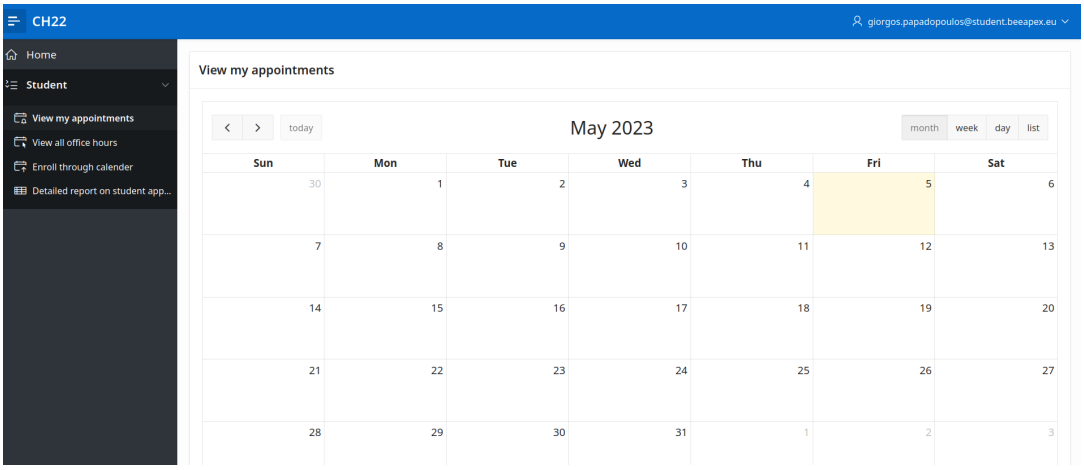


Obr. 22.10: Interaktívny prehľad o konzultačných hodinách po zmene termínu.

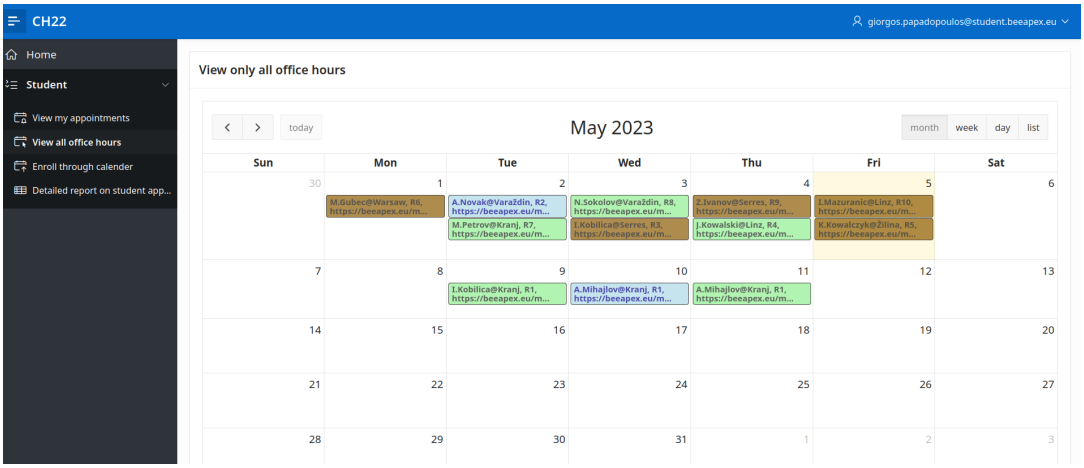
22.5.3 Rozhrania študentskej aplikácie

Položky kalendára sú podfarbené v závislosti od využitia. NONE enrolled (nikto neprihlásený) je zobrazený zelenou farbou, AVAILABLE (dostupný) je zobrazený modrou farbou a OVEBOOKED (obsadený) je zobrazený hnedou farbou. V tomto okamihu sa študent môže zapísať, aj keď je určený stav OVERBOOKED. Používatelia s právami študenta majú prístup ku:

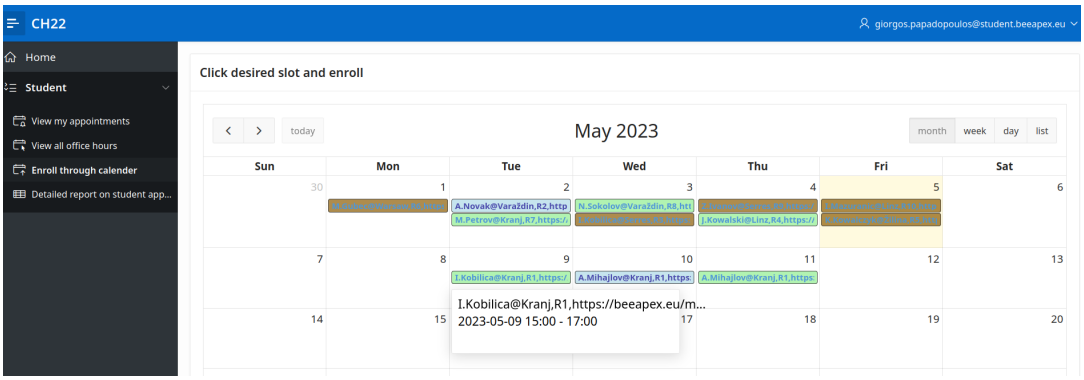
- kalendáru konzultácii študenta (viď Obrázok 22.11).
- kalendáru všetkých konzultačných hodín učiteľov zapojených do študijného programu študenta (viď Obrázok 22.12).
- zápisu na konzultačné hodiny prostredníctvom kalendára. Používatelia si najprv vyberú požadované konzultačné hodiny. (viď Obrázok 22.13). Na formulári si študent vyberie iba účel (viď Obrázok 22.14) a klikne na "Enroll" tlačidlo. Ak je zápis úspešný, študent dostane spätnú väzbu (viď Obrázok 22.15). Kontrola zápisu prostredníctvom položky "View my appointments" potvrdzuje úspešný zápis (viď Obrázok 22.16).
- Podrobný interaktívny prehľad prezentuje všetky konzultácie študentov v tabuľkovej forme (viď Obrázok 22.17).



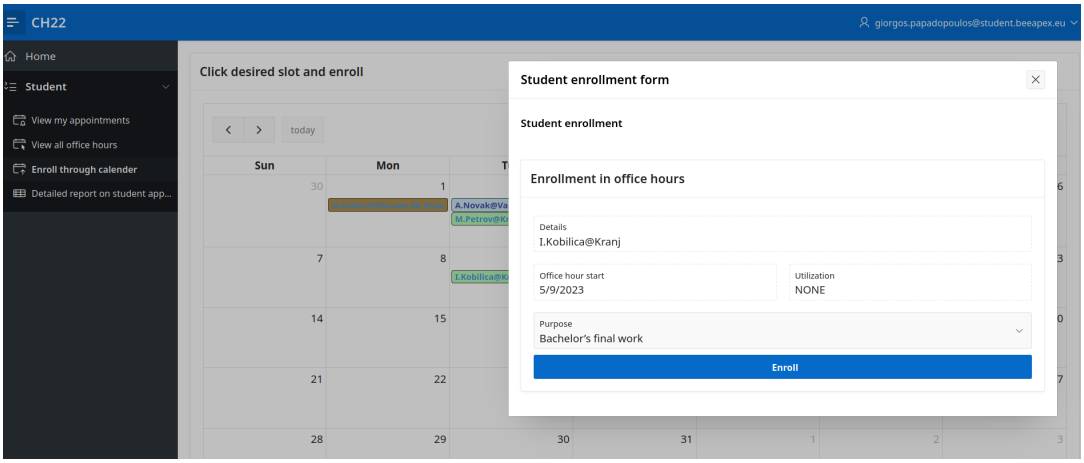
Obr. 22.11: Kalendár konzultácií študentov.



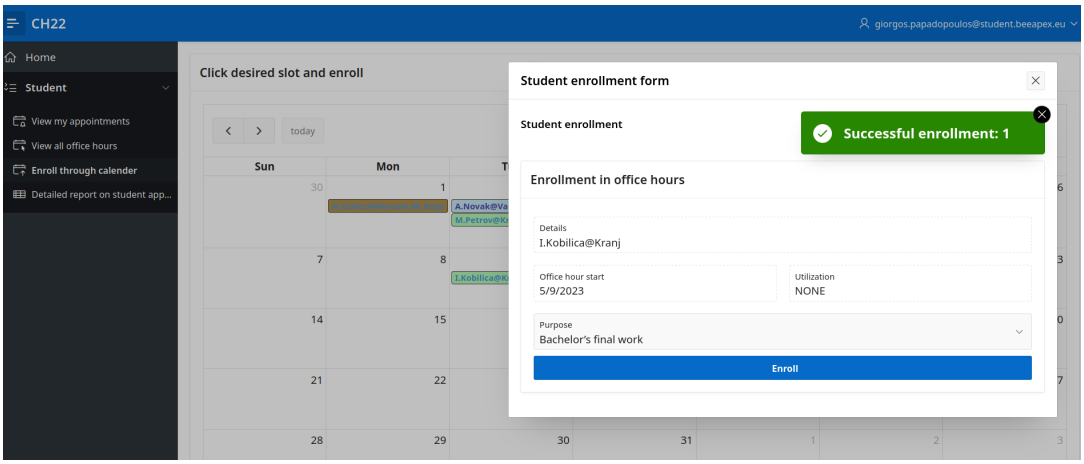
Obr. 22.12: Kalendár všetkých konzultačných hodín učiteľov zapojených do študijného programu študenta.



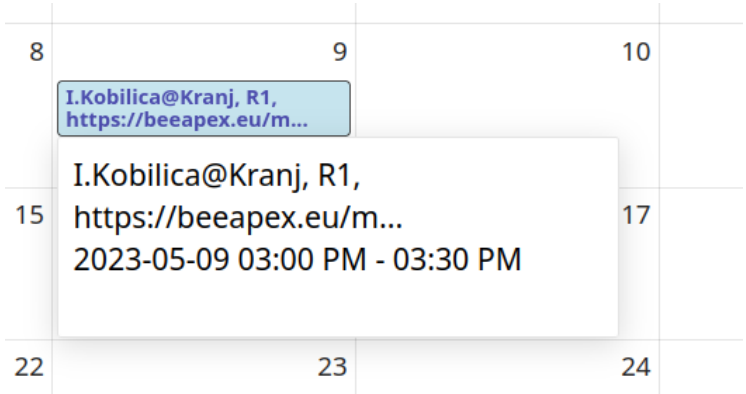
Obr. 22.13: Zápis na konzultačné hodiny prostredníctvom kalendára - výber okienka v kalendári.



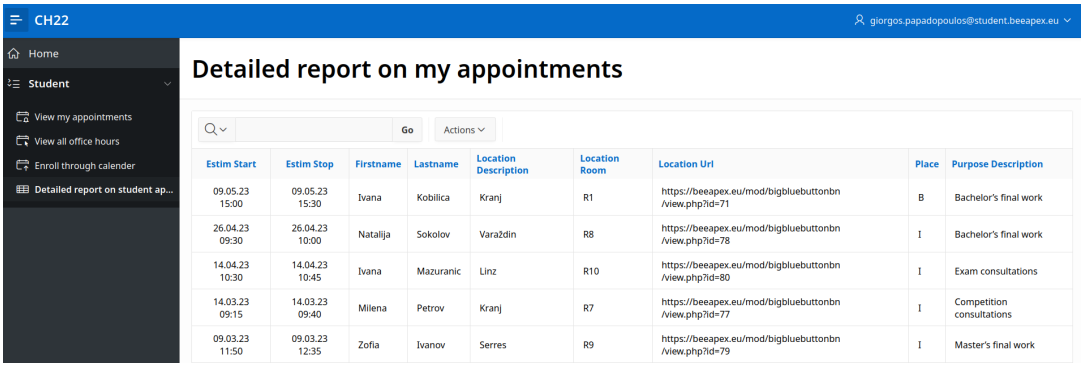
Obr. 22.14: Zápis na konzultačné hodiny prostredníctvom kalendára - výber účelu.



Obr. 22.15: Zápis do konzultačných hodín prostredníctvom kalendára - úspešný zápis.



Obr. 22.16: Kontrola zápisu prostredníctvom položky “View my meetings”.

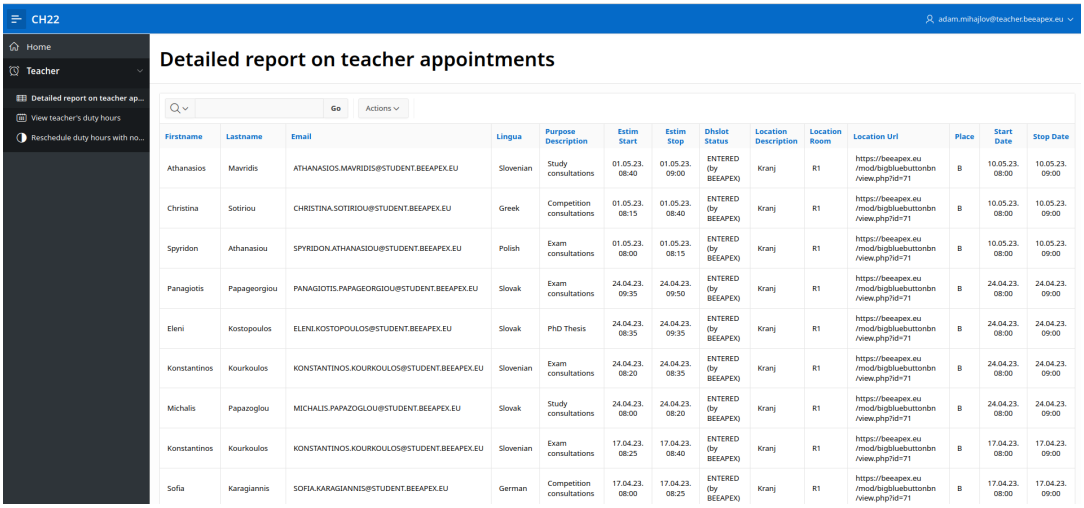


Obr. 22.17: Podrobný interaktívny prehľad o všetkých konzultáciách študentov.

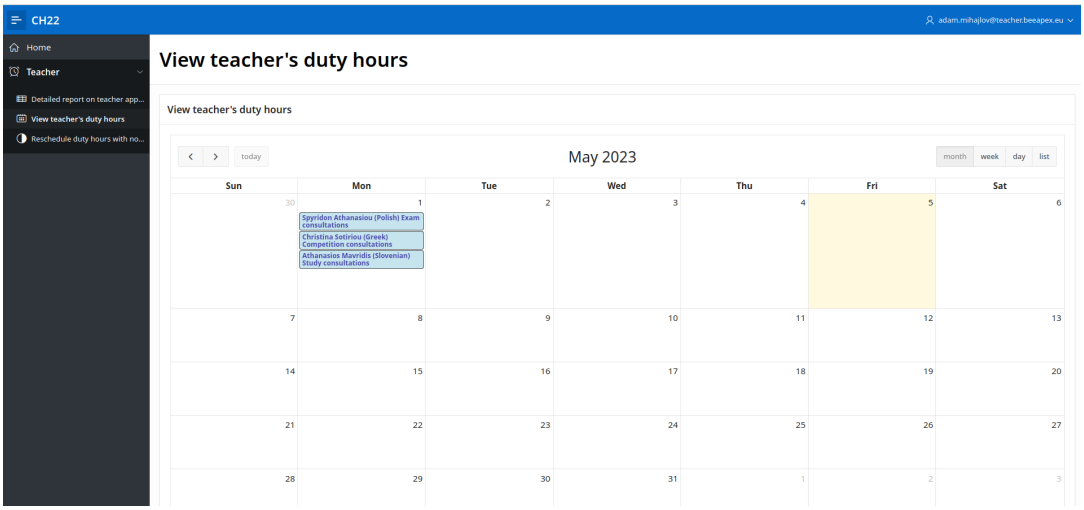
22.5.4 Rozhrania aplikácie pre učiteľov

Položky kalendára sú podfarbené v závislosti od využitia. Farby stavov sú rovnaké ako v rozhraní pre študentov (NONE je zelená, AVAILABLE je modrá a OVEBOOKED je hnedá). Používatelia s právami učiteľa môžu:

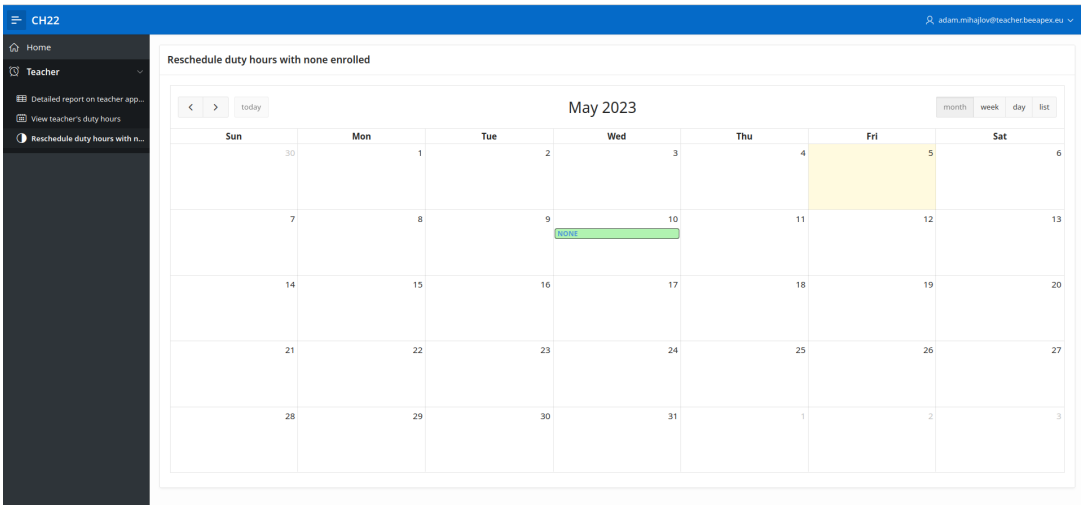
- zobraziť podrobný prehľad o konzultáciách učiteľov (viď Obrázok 22.18).
- zobraziť kalendár učiteľa. Meno študenta, jazyk a účel sa zobrazujú pre každého zapísané študenta. (viď Obrázok 22.19).
- zmeniť konzultačné hodiny, na ktoré nie je nikto zapísaný. Najskôr sa zobrazí kalendár konzultačných hodín so stavom NONE (viď Obrázok 22.20). Kliknutím na položku v kalendári sa otvorí modálny formulár, na ktorom sa zadávajú dátumy začiatku a konca (tiež hodiny a minúty) (viď Obrázok 22.21). Kliknutím na tlačidlo "Apply changes" sa konzultačné hodiny presunú na požadovaný nový dátum (viď Obrázok 22.22).



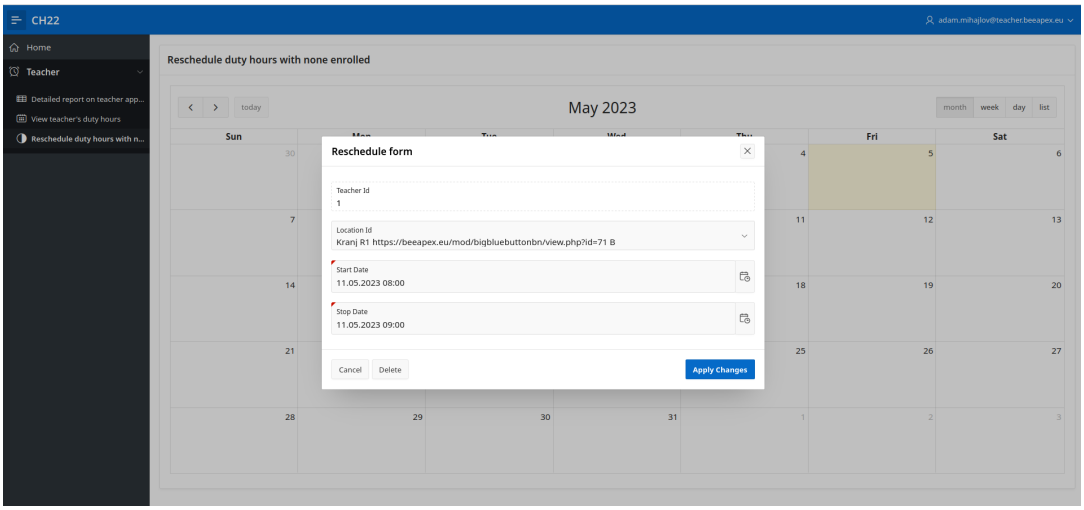
Obr. 22.18: Detailný prehľad o konzultáciách učiteľov



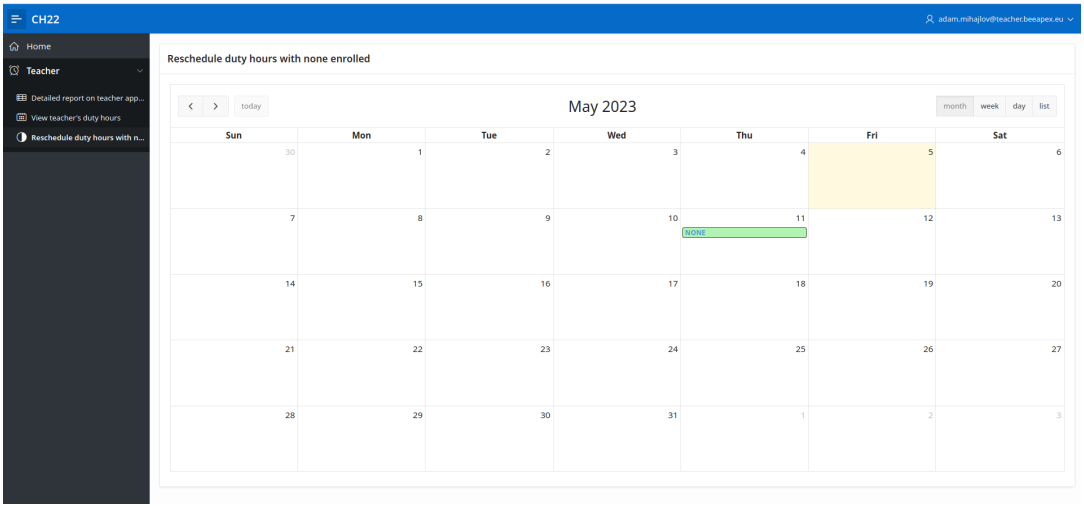
Obr. 22.19: Kalendár učiteľa s viditeľnými menami žiakov, ich jazykmi a cieľmi.



Obr. 22.20: Zmena konzultačných hodín učiteľa, na ktoré nie je niekto zapísaný - zobrazenie kalendára.



Obr. 22.21: Zmena termínu konzultačných hodín učiteľa, na ktorý nie je niekto zapísaný - zadaný nový termín.



Obr. 22.22: Výsledky preloženia konzultačných hodín učiteľa, na ktoré nebol nik zapísaný.

22.6 Doplnkový učebný materiál

Doplnkové učebné materiály k dispozícii:

- exportovaná aplikácia, skripty na vytváranie tabuliek, vkladanie údajov, vytváranie funkcie PL/SQL a odstraňovanie tabuliek.
- videonávody: a) inštalácia zabalenej aplikácie a vytvorenie používateľov (študentov a učiteľov) a b) vývoj aplikácie od začiatku.

All supplementary learning material is available on public BeeAPEX project page. Login as a guest user (no password is required). Find textbook in Books section, scripts in folder Part 2 > Chapter22 in the Scripts section and video guides in Collection of video guides. Material for short courses is in Short courses section.

Všetky doplnkové vzdelávacie materiály sú k dispozícii na verejnej stránke projektu BeeAPEX. Prihláste sa ako hosťujúci používateľ (nevyžaduje sa heslo). Učebnicu nájdete v časti Books, skripty v priečinku Part 2 > Chapter22 v Scripts časti a videonávody v časti Collection of video guides. Materiály pre krátke kurzy sa nachádzajú v časti Short courses.

22.6.1 Exportovaná aplikácia

Exportovaná aplikácia je zabalená. Inštalácia vytvorí tabuľky, indexy, funkcie, ako aj naplní údaje. Odinštalovaním sa odstránia všetky objekty databázy údajov použité v tejto aplikácii. Zabalená aplikácia má vložený súbor CSS (ch22_cal.css), na ktorý odkazuje väčšina stránok kalendára ako #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;
}
```

Manažment dashboard používa nasledujúce queries:

```
1          SELECT
2          EXTRACT(YEAR FROM start_date) || '-' ||
3          EXTRACT(MONTH FROM  start_date) AS "Year and Month",
4          COUNT(id) AS count
5          FROM ch22_duty_hour
6          GROUP BY EXTRACT(YEAR FROM start_date),
7          EXTRACT(MONTH FROM start_date)
8          order by 1;
9
10         with teacher_appointmets (appointment, teacher) as
11         (select CH22_DH_SLOT.ID,
12         substr(CH22_TEACHER.FIRSTNAME,1,1) || '.' ||
13         CH22_TEACHER.LASTNAME
14         from CH22_DH_SLOT CH22_DH_SLOT,
15         CH22_DUTY_HOUR CH22_DUTY_HOUR,
16         CH22_TEACHER CH22_TEACHER
17         where CH22_DUTY_HOUR.ID=CH22_DH_SLOT.SLOT_ID
18         and CH22_TEACHER.ID=CH22_DUTY_HOUR.TEACHER_ID)
19         select count(appointment), teacher from
              teacher_appointmets
20         group by teacher
21         order by 1 desc
22         fetch  first 10 rows only;
23
24         with teacher_dh_statuses (duty_hour_id, teacher,
              dh_utilization) as
25         (select CH22_DUTY_HOUR.ID,
26         substr(CH22_TEACHER.FIRSTNAME,1,1) || '.' ||
27         CH22_TEACHER.LASTNAME,
28         CH22_dh_utilization(CH22_DUTY_HOUR.ID)
29         from CH22_DUTY_HOUR CH22_DUTY_HOUR,
30         CH22_TEACHER CH22_TEACHER
31         where CH22_DUTY_HOUR.TEACHER_ID=CH22_TEACHER.ID and
32         CH22_dh_utilization(CH22_DUTY_HOUR.ID) = 'OVERBOOKED')
33         select count(duty_hour_id), teacher from
              teacher_dh_statuses
34         group by teacher
35         order by 1 DESC
```

```
36          fetch  first 5 rows only;
37
38          with teacher_dh_statuses (duty_hour_id, teacher,
               dh_utilization) as
39          (select CH22_DUTY_HOUR.ID,
40          substr(CH22_TEACHER.FIRSTNAME,1,1) || '.' ||
               CH22_TEACHER.LASTNAME,
41          CH22_dh_utilization(CH22_DUTY_HOUR.ID)
42          from CH22_DUTY_HOUR CH22_DUTY_HOUR,
43          CH22_TEACHER CH22_TEACHER
44          where CH22_DUTY_HOUR.TEACHER_ID=CH22_TEACHER.ID and
45          CH22_dh_utilization(CH22_DUTY_HOUR.ID) = 'NONE')
46          select count(duty_hour_id), teacher from
               teacher_dh_statuses
47          group by teacher
48          order by 1 DESC
49          fetch  first 5 rows only;
```

Pozrite si ďalšie dotazy na stránky, ktoré sú založené na prvku kalendára v balenej aplikácii.

22.6.2 Videonávody

Videonávody ukazujú inštaláciu zbalenej aplikácie a vývoj aplikácie od začiatku.

22.7 Otázky

1. Ako pridať používateľa s právami učiteľa?
2. Ako by ste vylepšili aplikáciu tým, že by ste študentovi umožnili robiť si poznámky pre konkrétne stretnutie?
3. Ako by ste obmedzili výsledky query na riadky, ktoré majú stĺpce dátumu rovné alebo väčšie ako aktuálny čas

22.8 Odpovede

1. Použite Manage Users and Group. Pridajte nového používateľa. Choďte do SQL Workshop > Object Browser a vyberte tabuľku CH22_TEACHER. Vyberte kartu "Data" a "Insert row". E-mail vo vloženom riadku sa musí zhodovať s vytvoreným používateľom APEX.
2. Vytvorte novú tabuľku, t.j. CH22_NOTES s poliami: ID, note a DH_SLOT_ID ako cudzí kľúč, ktorý odkazuje na tabuľku CH22_DH_SLOT. Vytvorte novú správu a formulár pre študenta a autorizujte stránku pre študenta. V závislosti od inštitucionálnych pravidiel môžete učiteľom udeliť právo zobrazovať alebo upravovať poznámky študentov.
3. Ak chcete obmedziť výsledky query na riadky, ktoré majú stĺpce dátumu rovné alebo väčšie ako aktuálny čas, je potrebné pridať podmienku WHERE date_column >= SYSDATE.

23. Prípad Telco

Veronika Šalgová, Jozef Kostolný, Michal Mrena, Michal Kvet, Miroslav Potočár

23.1 VEYOMI aplikácia

Na demonštráciu tvorby prototypu aplikácie si ako príklad vyberieme fiktívnu spoločnosť VEYOMI. VEYOMI je malá spoločnosť, ktorá sa zameriava na predaj telekomunikačných služieb svojim klientom. Vytvorenie a údržba aplikácie na správu produktov, klientov a automatickú fakturáciu je nevyhnutná pre rastúci rozsah produktov. S rastúcou konkurenciou na trhu je tiež nevyhnutné poskytovať klientom ľahko dostupné informácie. Pre podnikanie je kľúčové evidovať klientov a produkty so službami a aby mal zákazník jednoduchý prístup k aktuálnemu stavu svojich služieb.

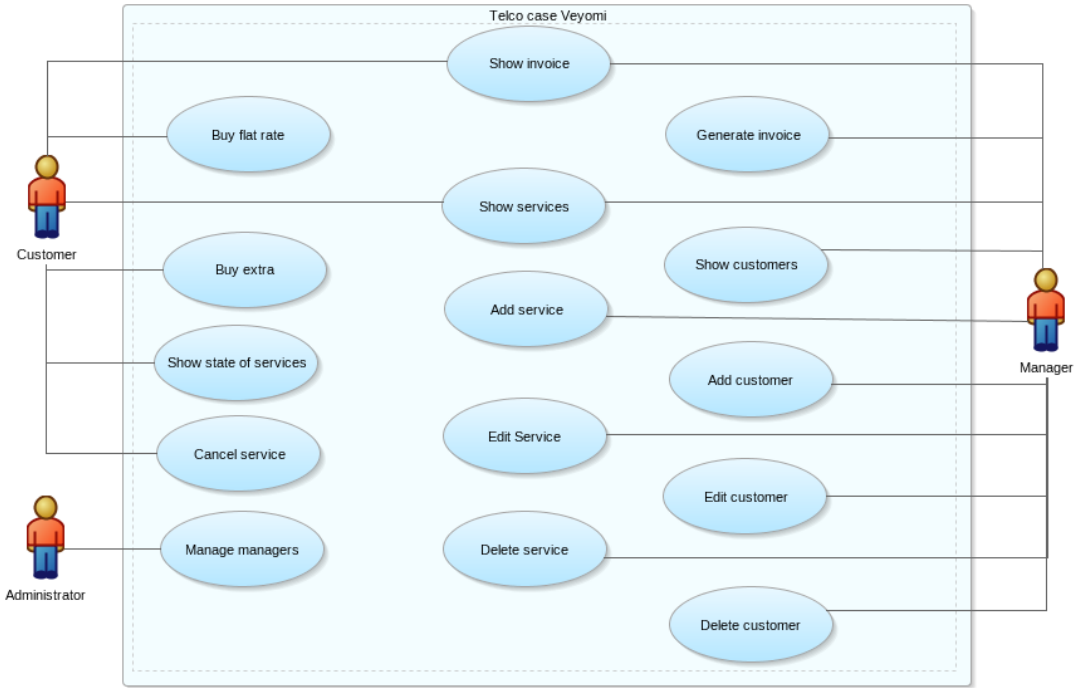
23.2 Diagram prípadov použitia

Pri tvorbe aplikácie je potrebné začať špecifikáciou služieb, ktoré by mala aplikácia poskytovať a čo by mala pokrývať. Preto je príprava zoznamu jednotlivých funkcionalít a služieb kľúčová. Najprv je však potrebné zamerať sa na definovanie typov používateľov. Pre naše použitie môžeme zvážiť tri základné role – zákazníka, manažéra a správcu. Každá rola má špecifické služby na zabezpečenie celkového pokrytia funkčnosti aplikácie.

Zákazník si potrebuje zabezpečiť nákup paušálnych a nadštandardných služieb so zobrazením stavu služieb, ktoré by mal mať možnosť aj zrušiť. Ďalšou podstatnou funkcionalitou na pokrytie je zobrazenie prehľadu faktúr.

Rola manažéra má prístup k prehľadu zákazníkov a ich pridávaniu, úprave a odstraňovaniu. Okrem toho môže manažér generovať aj faktúry.

Poslednou užívateľskou rolou je administrátor, ktorý má jednu základnú funkcionalitu: riadenie manažérov. Reprezentácia týchto rolí je znázornená na Obrázku 23.1.



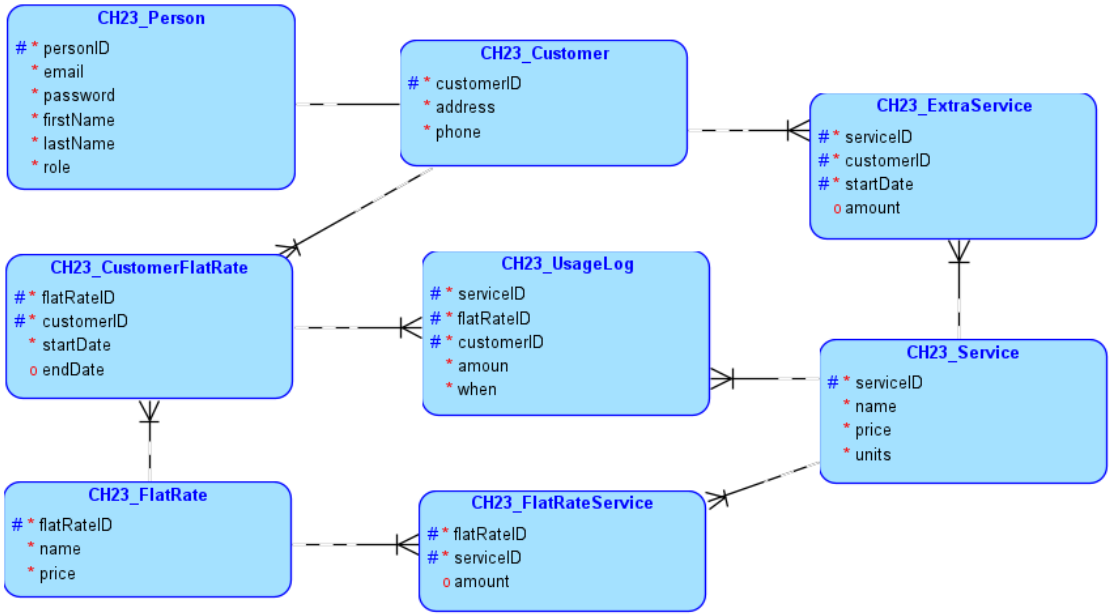
Obr. 23.1: Diagram prípadov použitia.

23.3 Dátový model

23.3.1 Naratívny popis

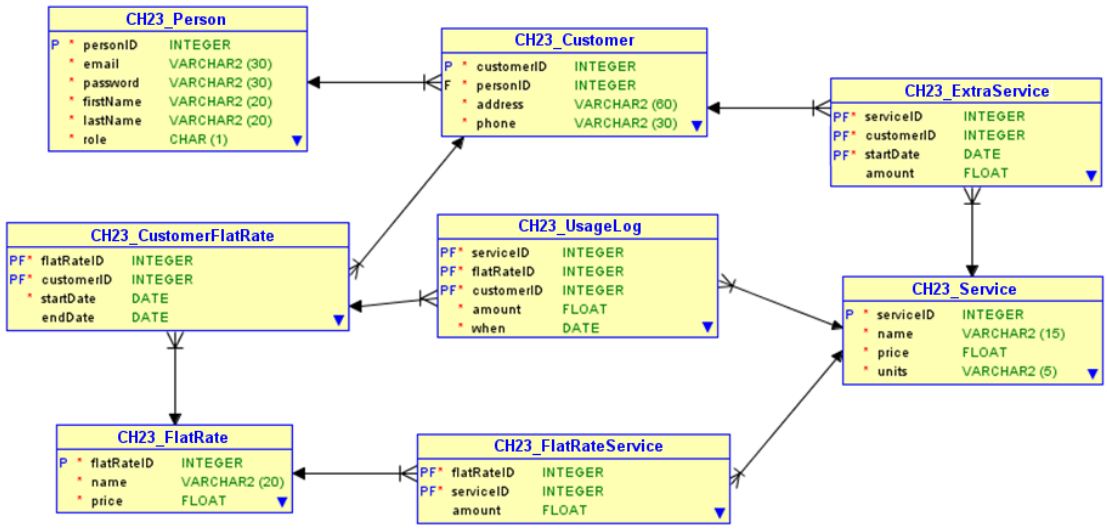
V tabuľke **CH23_Person** sú uložené primárne osobné údaje a typ roly používateľa. Keď je človek zákazníkom, má vytvorený záznam v tabuľke **CH23_Customer**, kde je okrem kľúčov uložená aj jeho adresa a telefónne číslo. V tabuľke **CH23_FlatRate** sú uložené informácie o paušále, konkrétne názov a cena. Údaje o jednotlivých službách, ich názve, cene a ponúkaných jednotkách sú uložené v tabuľke **CH23_Service**. Jednotlivé služby priradené k paušálom sú uložené v tabuľke **CH23_FlatRateService**, kde okrem kľúčov poznáme aj pridelené množstvo. Ak si zákazník zakúpi paušálnu sadzbu, informácie o tom sa uložia do tabuľky **CH23_CustomerFlatRate** vrátane dátumu začiatku a konca. Spotrebované jednotky z paušálu sa evidujú v tabuľke **CH23_UsageLog**, kde vieme, ktorý zákazník využil ktorú službu v rámci ktorej paušálnej sadzby, spolu s dátumom a množstvom. K paušálu si zákazník môže dokúpiť extra služby zaznamenané v tabuľke **CH23_ExtraService** a ich výšku.

23.3.2 Logický dátový model



Obr. 23.2: Logický dátový model.

23.3.3 Relačný dátový model



Obr. 23.3: Dátový model.

23.4 Autentifikácia používateľa a používateľské roly

Vo vyššie uvedenom popise sme identifikovali tri používateľské roly. Aplikácia po prihlásení sprístupní používateľovi obsah podľa jeho roly. Samotný proces prihlásenia je však pre všetkých používateľov rovnaký. Apex nám ponúka niekoľko spôsobov, ako spravovať používateľov a overovať ich. V nasledujúcom zozname uvádzame niekoľko vybraných metód:

- Databázový účet,

- LDAP server,
- konto Oracle Apex,
- prihlásenie do sociálnych sietí,
- vlastné (Custom).

V nástroji Oracle Apex zvolíme metódu autentifikácie - **Authentication Scheme** - na podstránke **Shared Components \ Authentication Schemes**. Na tejto podstránke nájdeme zoznam existujúcich schém prihlasovania a môžeme si tu vytvoriť aj novú schému prihlasovania. Pri vytváraní novej schémy si môžeme zvoliť jednu z vyššie uvedených metód autentifikácie. Pre potreby našej aplikácie sme zvolili najvšeobecnejšiu metódu **Custom**.

Hlavnou časťou **Custom** autentifikácie je takzvaná autentifikačná funkcia. Vstupom do tejto funkcie je používateľské meno a heslo, ktoré používateľ zadá na prihlasovacej stránke, a výstupom je hodnota **boolean**. Ako sa očakávalo, funkcia vráti **true**, ak sa zadané meno a heslo zhodujú s existujúcim používateľom a **false** v opačnom prípade. Na Obrázku 23.1 môžeme vidieť hlavičku takejto funkcie. Preto pri vytváraní schémy prihlásenia **Custom** musíme zadať názov takejto funkcie. To znamená, že takáto funkcia už musí existovať.

```
CREATE OR REPLACE FUNCTION authenticate_user
(p_username IN varchar2,
p_password IN varchar2)
RETURN boolean;
```

Listing 23.1: Hlavička autentifikačnej funkcie.

Telo autentizačnej funkcie implementujeme pomocou jazyka PL/SQL, t.j. je len na nás, akým spôsobom používateľa overíme. To je dôvod, prečo je metóda **Custom** najvšeobecnejšia. Môžeme napríklad kontaktovať externú autentifikačnú službu pomocou REST API, vyhľadať používateľa v lokálnej databáze alebo nechať autentifikáciu na iný lokálny proces. V našej aplikácii sme zvolili bežne zaužívaný postup vyhľadávania a overovania používateľa v lokálnej databáze.

Tabuľku **CH23_Person** môžeme vidieť v dátovom modeli, ktorý sme predstavili vyššie. Stĺpce **email** a **password** sú relevantné pre autentifikáciu. Používateľov **email** tiež slúži ako jedinečné prihlasovacie meno. Stĺpec **password** obsahuje používateľské heslá uložené v zabezpečenej hašovanej forme. Popis bezpečného ukladania hesiel je nad rámec tejto kapitoly. Viac sa o tom môže čitateľ dozvedieť v literatúre. Pred zadaním údajov alebo úpravou stĺpca **password** v tabuľke **CH23_Person** je však potrebné heslo zahašovať. Na tento účel sme vytvorili pomocnú funkciu **hash_password**, ktorej hlavičku vidíme na Obrázku 23.2. Táto funkcia prevezme používateľské meno a heslo vo forme otvoreného textu a vygeneruje heslo v zabezpečenom hashovanom formáte, ktoré je možné uložiť do databázy.

```
CREATE OR REPLACE FUNCTION hash_password
(p_username IN varchar2,
p_password IN varchar2)
RETURN varchar2;
```

Listing 23.2: Hlavička pomocnej funkcie hash_password.

Automatické hashovanie hesiel pri práci s tabuľkou **CH23_Person** sme vyriešili vytvorením triggrov pre tabuľku **CH23_Person**. Pred vložením alebo aktualizáciou stĺpca **password** triggre hašujú heslo s otvoreným textom pomocou vyššie opísanej funkcie **hash_password**. Zabezpečí, že všetky heslá v tabuľke **CH23_Person** budú bezpečne uložené – k aktuálnemu heslu nemá prístup ani správca databázy. V tabuľke 23.1 môžeme vidieť vzorku údajov uložených v tabuľke **CH23_Person**. Zaujímavé je, že dvaja používatelia uvedení v tabuľke majú rovnaké heslo. Túto skutočnosť však nevieme určiť na základe bezpečne uložených hesiel.

Implementácia autentifikačnej funkcie je teraz jednoduchá. V tejto funkcii najskôr hľadáme v tabuľke **CH23_Person** heslo používateľa s daným používateľským menom. Potom zahašujeme

PERSONID	EMAIL	PASSWORD	ROLE
5	DUIS.A@OUTLOOK .CA	$2a$12$EfSjb2zSiKjqWCoZ47m QeOY6IPoHP/6LvECZxGEWjFZ IZcXTF3UgGasd	'a'
11	LOBORTIS@OUTLO OK.EDU	$2a$12$IUqALTj3reLAoTdtC5SN de2.C5A.DOm..7pFw9UBnIsaf7gS NcZyapql	'c'
14	AC.FERMENTUM@ AOL.NET	$2a$12$GuaO7Tjp.65d7NYLiEjS zeO6VWQfDz8eXja7dWubcjy77a ZDAKRTutyq	'm'

Tabuľka 23.1: Vzorové údaje uložené v tabuľke CH23_Person.

heslo, ktoré používateľ zadal pomocou funkcie **hash_password**. Ak sa oba hashe zhodujú, overenie používateľa je úspešné a funkcia vráti **true**. Na druhej strane, ak sa hodnoty hash nezhodujú alebo v tabuľke **CH23_Person** nie je žiadny používateľ s daným menom, vrátime hodnotu false.

Nevýhodou nami zvolenej metódy **Custom** je väčšia počiatočná náročnosť pri vytváraní databázy, v rámci ktorej je potrebné vyvíjať pomocné funkcie a triggre. Na druhej strane výhodou je, že celý proces autentifikácie kontrolujeme my, vývojári aplikácií. Ďalšou výhodou je, že použitá implementácia je ľahko škálovateľná. Ak by sme napríklad v budúcnosti chceli na autentifikáciu využívať externú službu, potrebovali by sme zmeniť implementáciu autentifikačnej funkcie tak, aby namiesto lokálnej databázy kontaktovala externú službu.

Druhou dôležitou úlohou súvisiacou so správou používateľov je správa používateľských rolí. Aj keď proces prihlásenia bol pre všetkých používateľov rovnaký, pri správe rolí sme potrebovali urobiť prístup k rôznym častiam aplikácie na základe roly používateľa. Správa rolí úzko súvisí so správou používateľov. Po prvé, všetky možné užívateľské roly musia byť špecifikované v prostredí Oracle Apex. Správu rolí nájdeme na podstránke **Shared Components \ Application Access Control**. Na tejto podstránke vidíme zoznam existujúcich rolí a môžeme tu vytvoriť aj novú rolu. Pre každú novú rolu je potrebné špecifikovať jej statický identifikátor, ktorý neskôr použijeme pri prideľovaní roly jednotlivým užívateľom. V našej aplikácii používame roly s nasledujúcimi statickými identifikátormi:

- ADMIN (administrátor),
- CUSTOMER (zákazník),
- MANAGER (manažér).

Používateľovi pridelíme rolu pomocou PL/SQL kódu. Na Výpise 23.3 môžeme vidieť vzorový kód na priradenie roly **CUSTOMER** (statický identifikátor roly) užívateľovi **LOBORTIS@OUTLOOK.EDU**. Na automatizáciu tohto procesu sme použili inú skupinu triggrov pre tabuľku Person. Stĺpec **Role** v tejto tabuľke môže nadobúdať hodnoty z množiny {'a', 'm', 'c'}, ktoré sa zhodujú so začiatočnými písmenami vyššie definovaných rolí. V tele triggra pred vložením alebo aktualizáciou údajov nastavíme pre daného používateľa príslušnú rolu podľa hodnoty stĺpca Role pomocou kódu podobného kódu na Výpise 23.3.

```
APEX_ACL.REPLACE_USER_ROLES (
 p_application_id    => 151905,
p_user_name          => 'LOBORTIS@OUTLOOK.EDU',
 p_role_static_ids   => wwv_flow_t_varchar2('CUSTOMER') );
RETURN boolean;
```

Listing 23.3: PL/SQL kód, ktorý priraďuje daného užívateľa k danej roly.

Posledným krokom implementácie užívateľských rolí v aplikácii je zobrazenie rôznych podstránok na základe roly prihláseného užívateľa. Pre každú podstránku, ktorá je určená len pre

konkrétnu rolu používateľa, je potrebné vybrať rolu používateľa, pre ktorú je daná podstránka určená v nastaveniach **Page Designer** stránky v sekcii **Security \ Authorization Scheme**. Tento krok zabezpečí, že používatelia s inými rolami nebudú mať prístup k tejto podstránke. Napríklad Domovská stránka našej aplikácie je špecifická tým, že na jej zobrazenie nie je potrebné prihlásenie. Toto správanie sme nastavili aj v sekcii **Security** položkou **Authentication**, kde vyberieme možnosť **Page Is Public**.

Vyššie popísané nastavenia zabezpečia, že používateľ bude mať prístup len na podstránky svojej roly. Všetci používatelia však stále uvidia všetky podstránky v navigačnom menu – aj tie, ku ktorým nemajú prístup. Takéto správanie určite nie je žiaduce. Aby aplikácia používateľom zobrazovala iba podstránky, ku ktorým majú prístup, je potrebné upraviť nastavenia navigačnej ponuky. Môžeme to nastaviť na podstránke navigačnej ponuky **Shared Components \ Navigation and Search \ Navigation Menu**. Tu v stĺpci **Authorization Scheme** pre každú položku v menu je potrebné nastaviť rolu používateľa, ktorej sa bude daná položka zobrazovať.

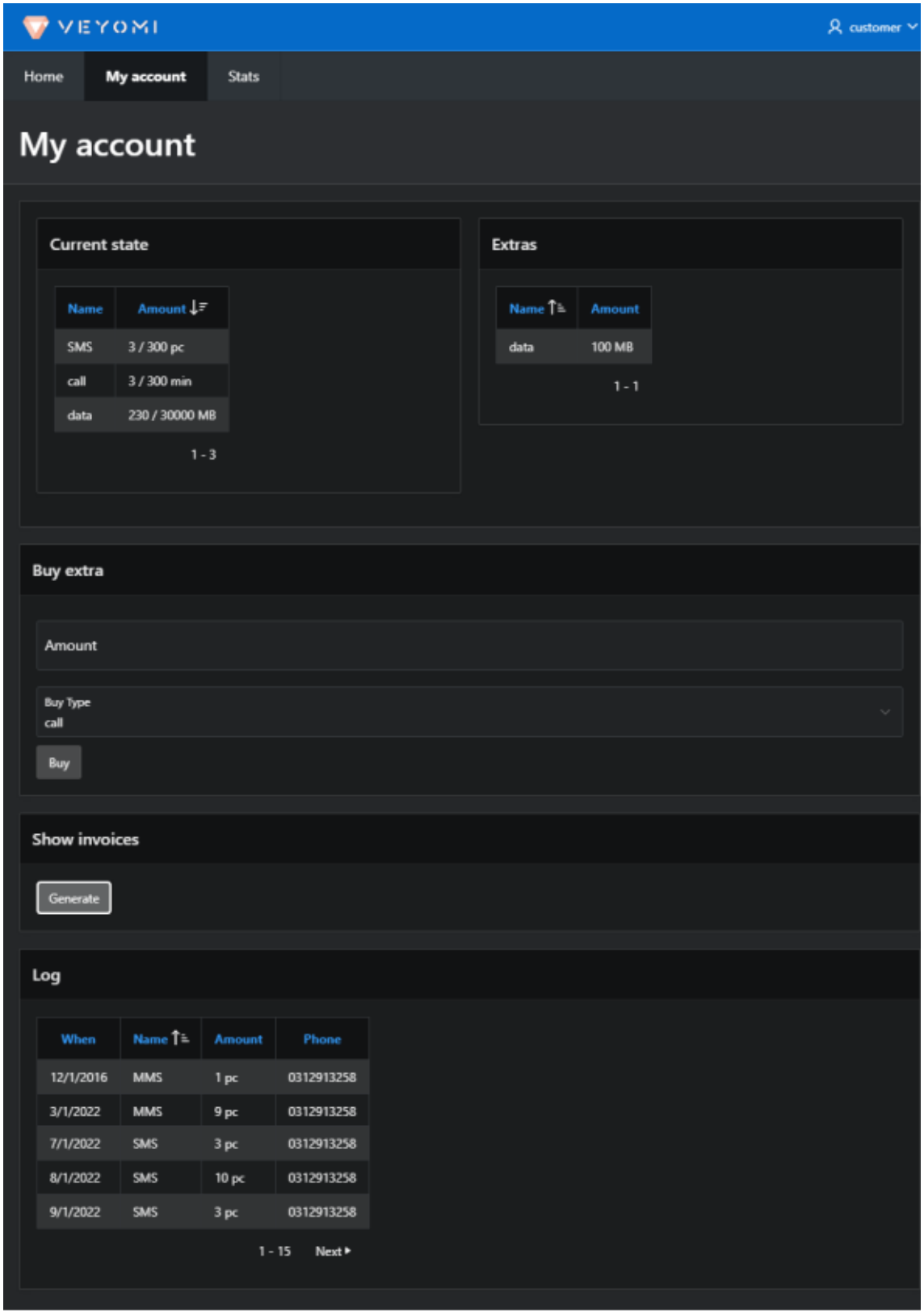
23.5 Rozhrania aplikácie

Dizajnom aplikácie je vytvorenie jednotlivých podstránok, ktoré budú poskytovať navrhované služby. Primárnou stránkou je úvodná stránka, ktorá poskytuje prehľad ponúkaných služieb a portfólia paušálnych služieb. Túto stránku je možné využiť na prezentáciu spoločnosti a poskytovaných služieb, ktoré si zákazník môže zakúpiť. Táto stránka je prístupná bez prihlásenia všetkým návštevníkom. Hlavná stránka ponúka ešte jednu dôležitú funkcionalitu: prístup do administrácie po prihlásení. Administráciou rozumieme súbor podstránok rozdelených podľa typu účtu, ku ktorému sa prihlasujete - zákazník, manažér, alebo admin. Pre každú z nich je teda potrebné vytvoriť samostatné podstránky, ktoré pokrývajú navrhovanú funkcionalitu služieb jednotlivej role.

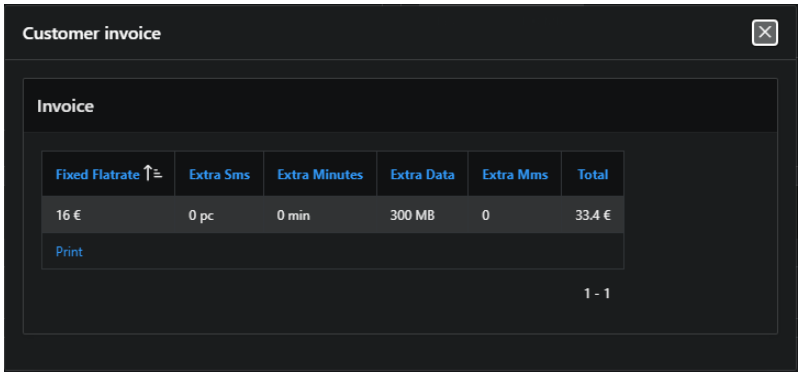
Zákaznícka stránka poskytuje prehľad o aktuálnom stave objednaných služieb a doplnkových služieb v jednotlivých oblastiach. Ďalšia extra záložka umožňuje nákup jednotlivých služieb výberom typu služby z rozbaľovacieho zoznamu. Ďalšou funkcionalitou je zobrazovanie a generovanie faktúr, čo je možné riešiť vyskakovacím oknom, v ktorom sa po otvorení zobrazí faktúra vo forme zostavy, ktorá umožňuje konfiguráciu ukladania a tlače vo formáte PDF. Výkaz k faktúre vyvoláva konkrétny skript sumarizujúci údaje za poskytnutú službu v danom mesiaci. Ďalšou kritickou časťou zákazníckej stránky je prehľad využívania služieb s podrobnosťami ako množstvo, typ služby a čas. Táto tabuľka umožňuje túto recenziu filtrovať a triediť pomocou špecifikácie filtra a triedenia podľa jednotlivých stĺpcov. Takáto funkcionalita môže byť pre klienta výhodná, keď chce získať napríklad prehľad o počte odoslaných SMS za dané obdobie. Nakoniec je zákazníkovi poskytnutá funkcionalita zobrazovania štatistík v grafoch v konkrétnom priečinku. Táto funkcionalita je riešená na samostatnej stránke, ktorá je ku stránke zákazníka pripojená záložkou a medzi jednotlivými funkcionalitami je možné prepínať. Štatistická stránka obsahuje ukážku grafického znázornenia log. Takáto reprezentácia je vhodnou reprezentáciou na znázornenie zmeny v čase. V objednanom programe môžeme napríklad jednoducho zobraziť prevolané minúty počas jednotlivých mesiacov, počet odoslaných SMS, či množstvo prenesených dát.

23.6 Dizajn aplikácie

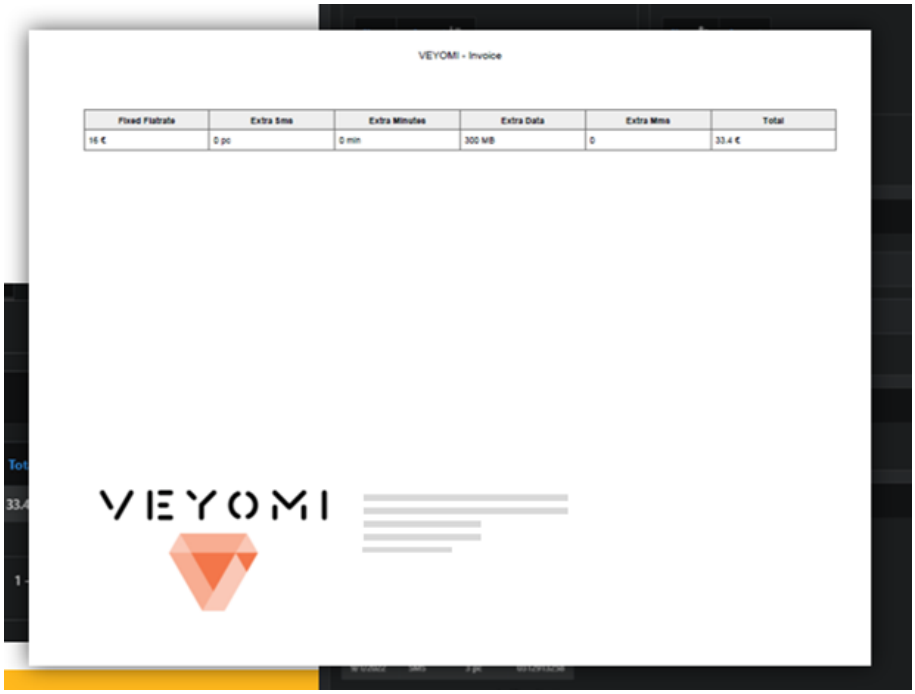
Táto kapitola zobrazuje snímky obrazoviek aplikačných formulárov rôznych rolí používateľov.



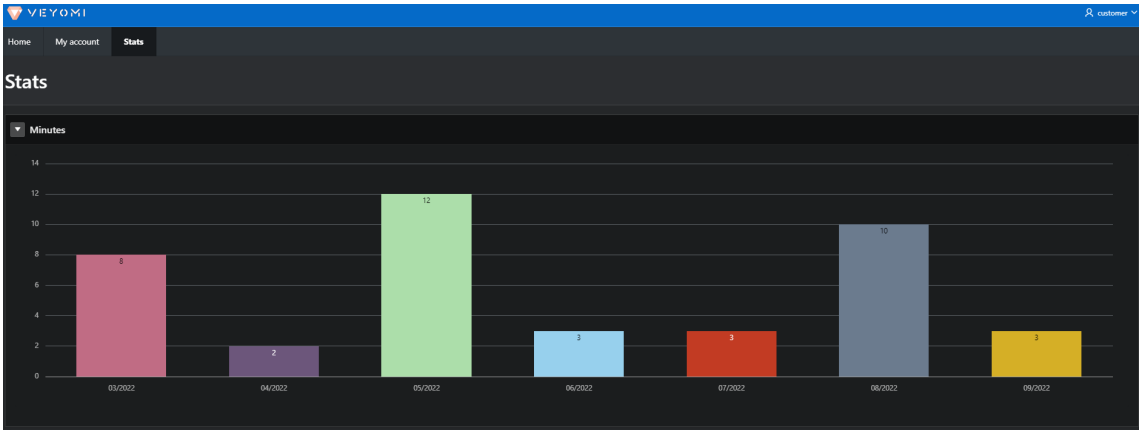
Obr. 23.4: Zákaznícky dashboard.



Obr. 23.5: Zákaznícky dashboard – Modálne okno faktúry.



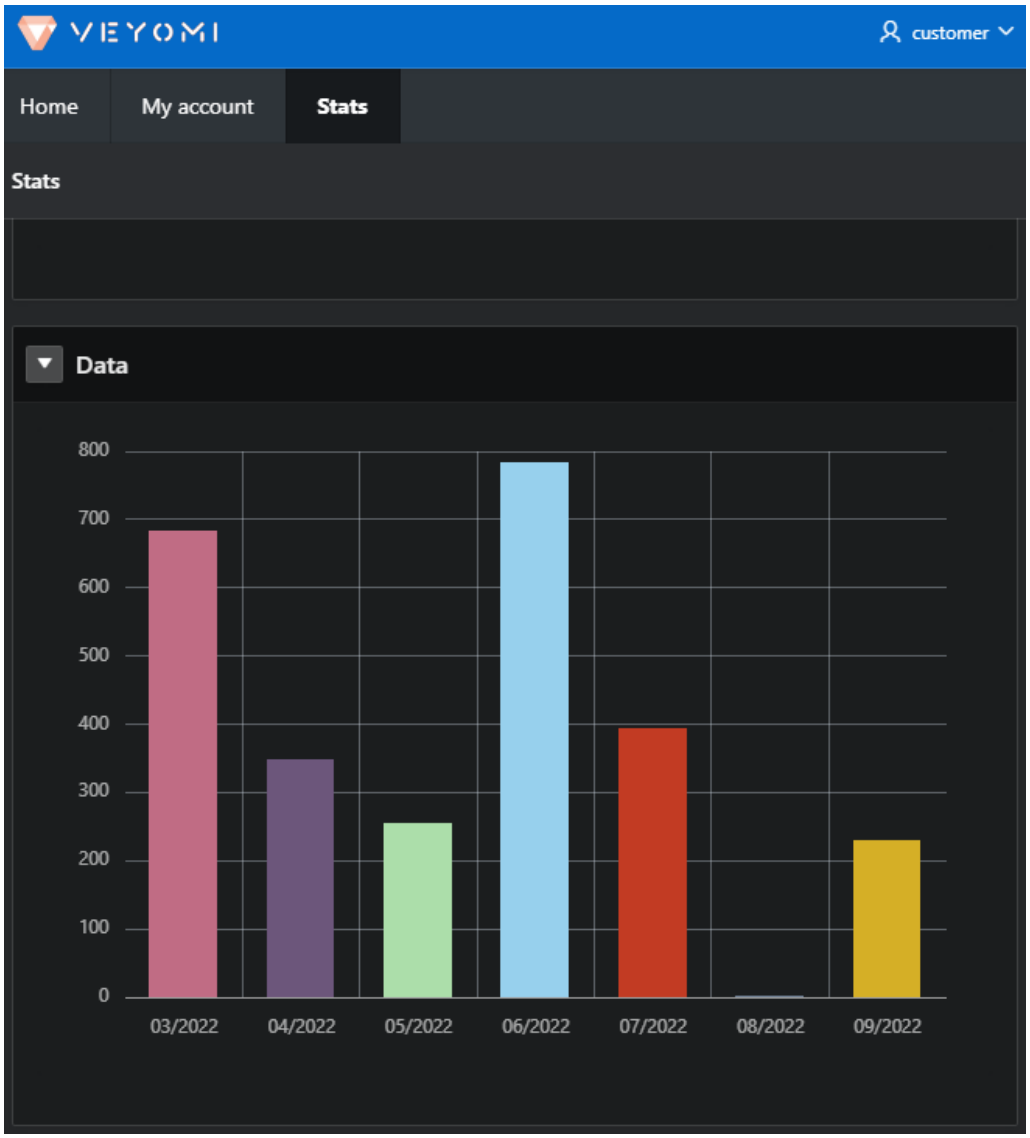
Obr. 23.6: Zákaznícky dashboard – Faktúra v PDF.



Obr. 23.7: Zákaznícky dashboard – Štatistika minút.



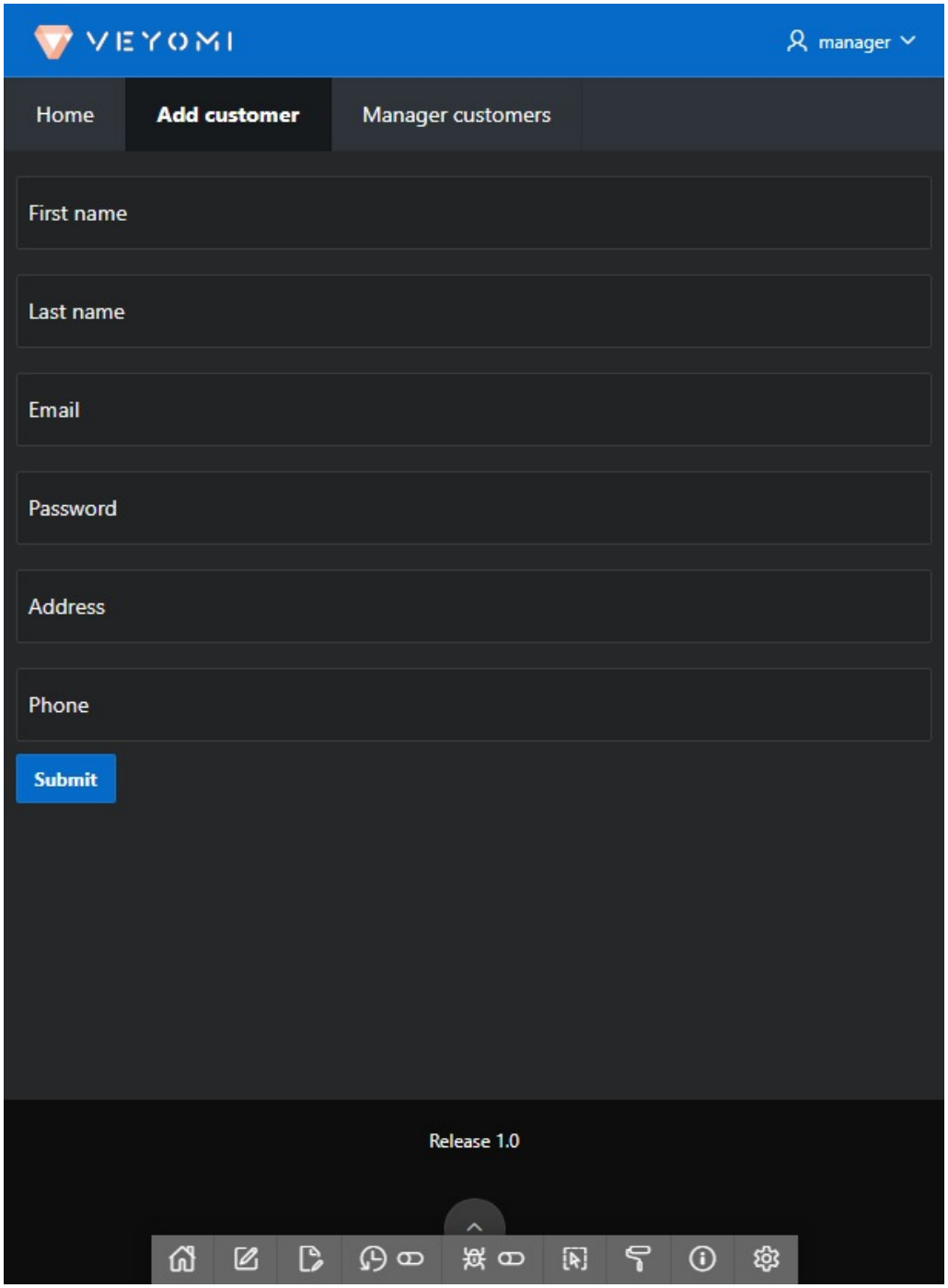
Obr. 23.8: Zákaznícky dashboard – Štatistika SMS.



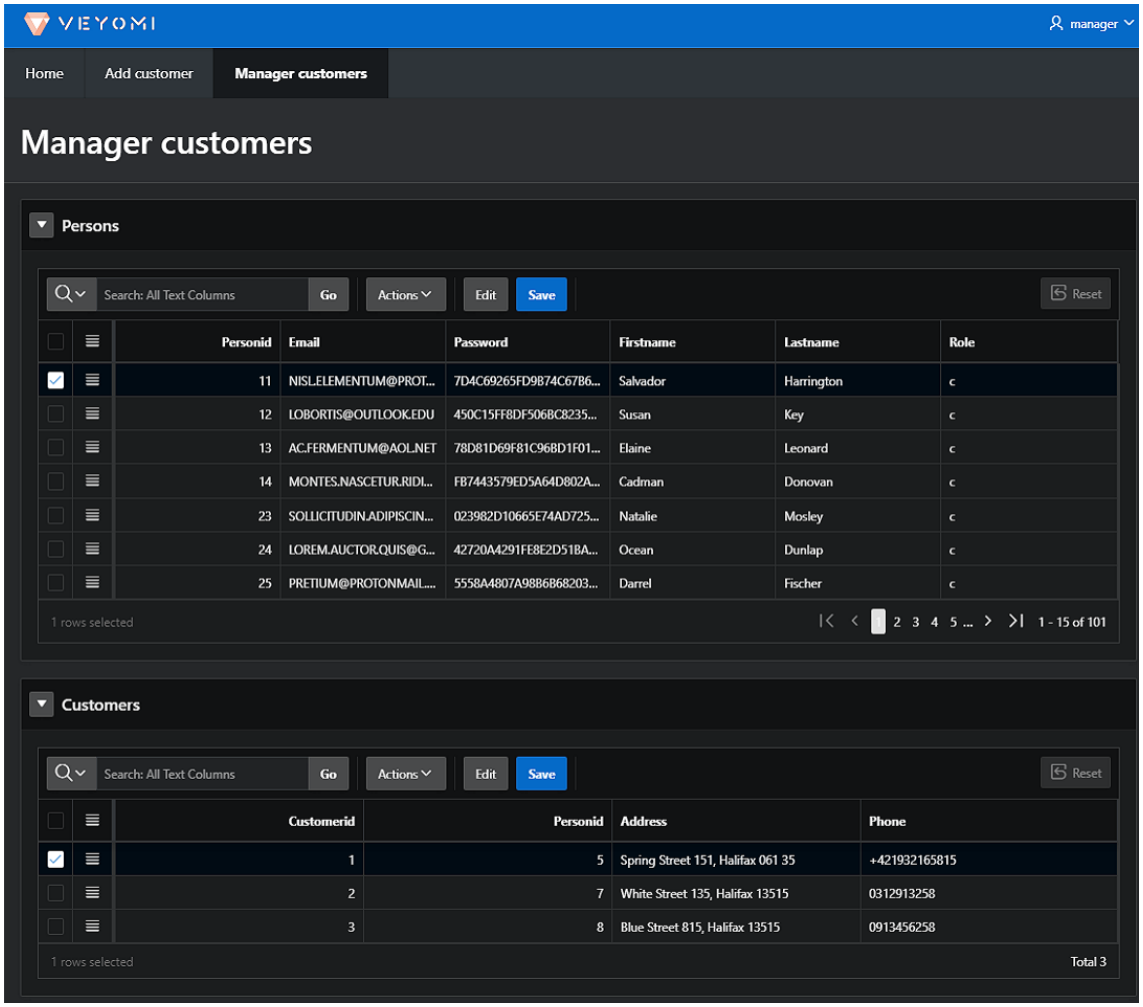
Obr. 23.9: Zákaznícky dashboard – Štatistika dát.

Po prihlásení manažéra do aplikácie je funkcionalita správy zákazníkov poskytovaná na samostatných záložkách, ktoré sú súčasťou stránky manažéra. Prvou funkcionalitou je pridanie nového

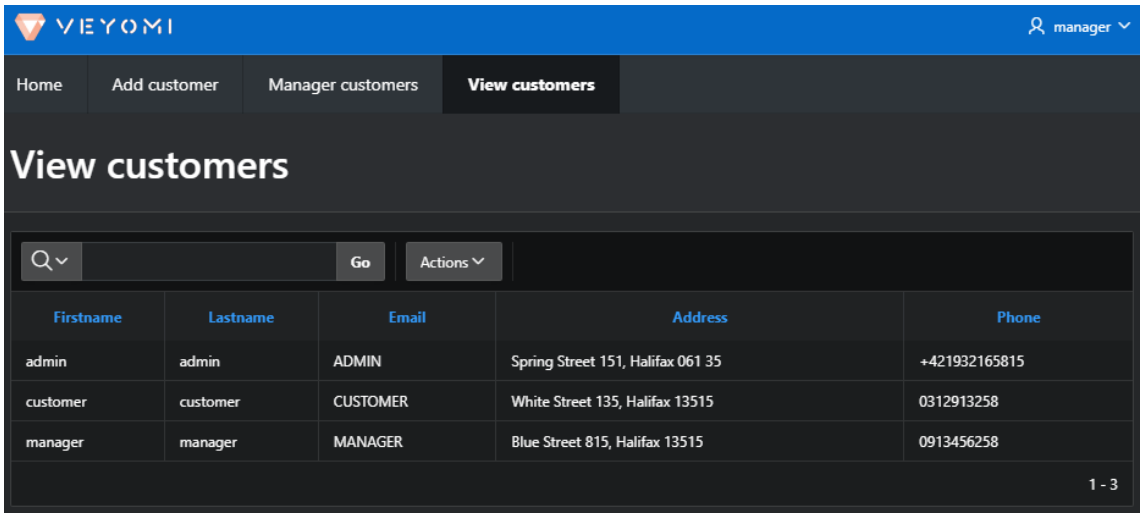
zákazníka, kde manažér môže pridať a vytvoriť nový zákaznícky účet. Ďalej používame komponent správy zákazníkov na úpravu informácií o zákazníkoch, ktoré sú navrhnuté v data grid-e. Tento komponent taktiež poskytuje funkcionalitu pre jednoduché reportovanie alebo exportovanie informácií o zákazníkovi iným aplikáciam tretích strán, ak je to potrebné. Na záver záložka Zobrazenie zákazníka demonštruje zjednodušenú formu zobrazenia prehľadu informácií o zákazníkovi. Táto mriežka je nakonfigurovaná tak, že neumožňuje editáciu jednotlivých položiek, ale má možnosť tvorby reportov a exportov z tohto zoznamu.



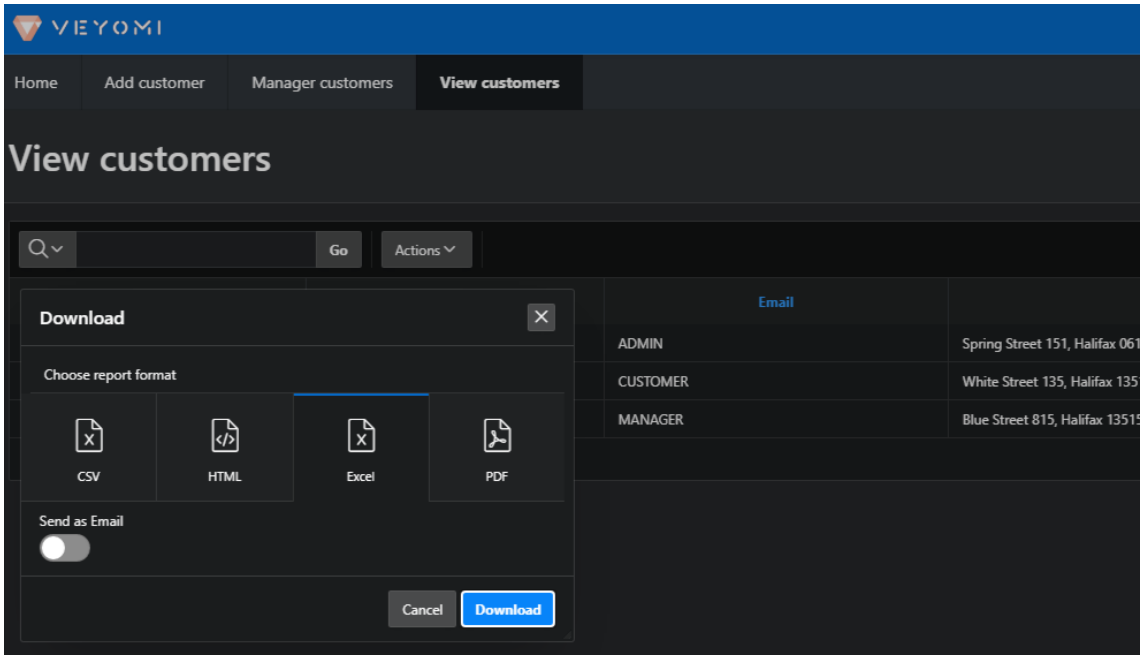
Obr. 23.10: Manažérsky dashboard – Pridanie zákazníka.



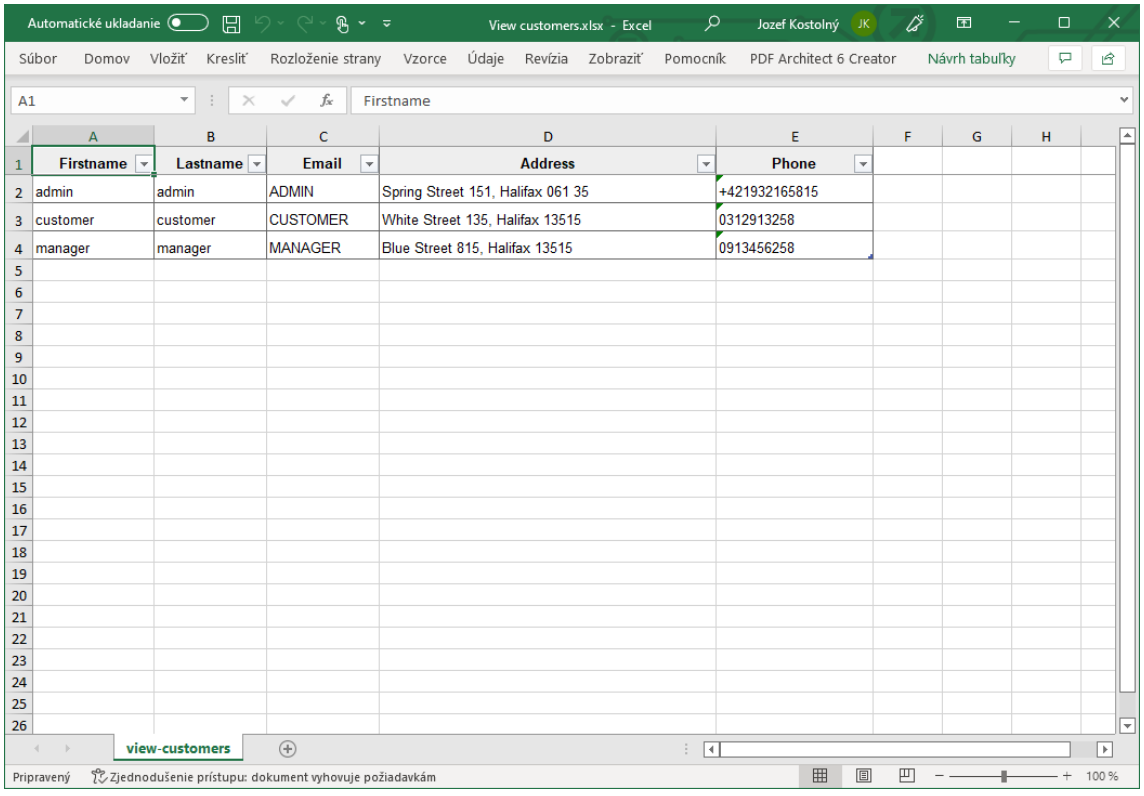
Obr. 23.11: Manažérsky dashboard – Pridanie zákazníka.



Obr. 23.12: Manažérsky dashboard – Zobrazenie zákazníkov.

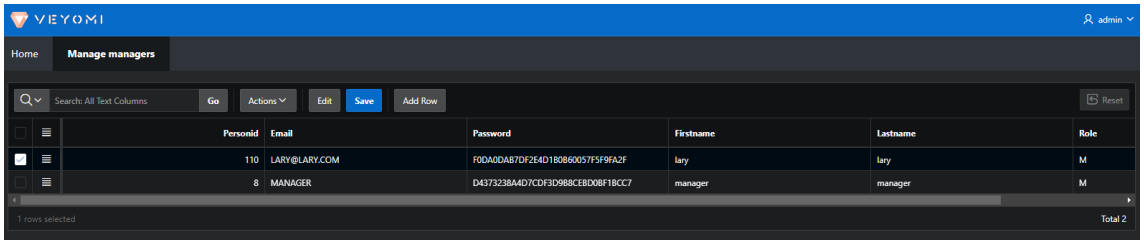


Obr. 23.13: Manažérsky dashboard – Export zákazníkov.



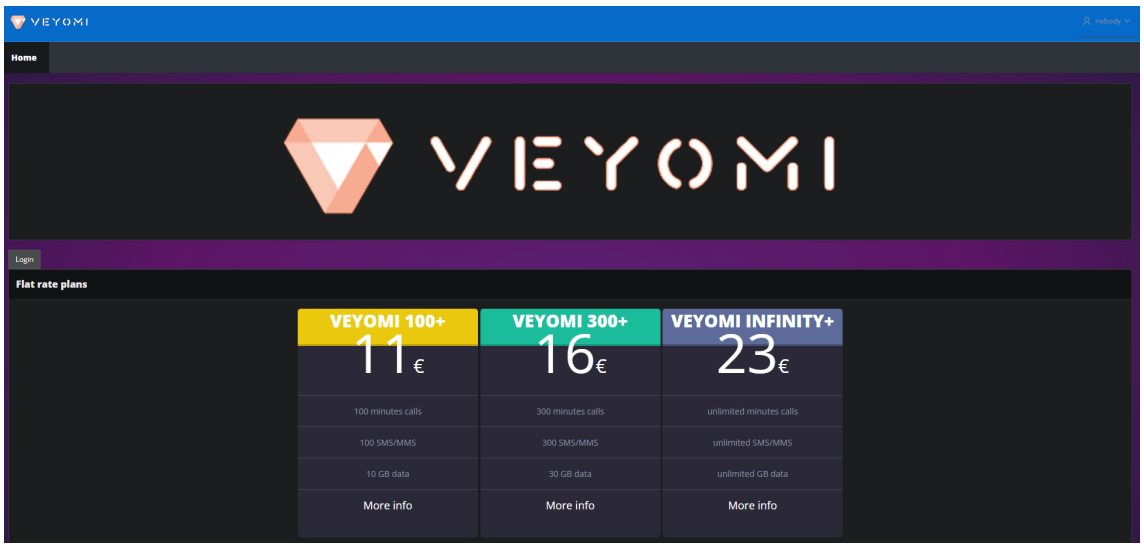
Obr. 23.14: Manažérsky dashboard – Export zákazníkov do XLS súboru.

Posledná stránka, ktorú je potrebné vytvoriť, je stránka pre administrátora aplikácie. Na tejto stránke poskytujeme službu správy manažérov v editovateľnom data grid-e, ktorý umožňuje exportovať reporty a funkcie filtrovania.

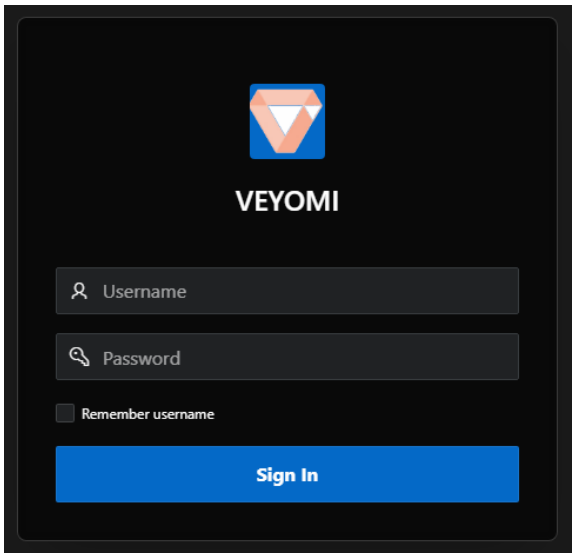


Obr. 23.15: Dashboard administrátora.

Dôležitou súčasťou je vytvorenie prihlasovacieho formulára. V našej aplikácii uvažujeme o umiestnení tohto formulára na konkrétnu stránku, ktorá sa volá vždy pri vstupe na zabezpečené stránky, kde je potrebné čo najskôr sa autentifikovať.



Obr. 23.16: Úvodná stránka s prihlásením.



Obr. 23.17: Prihlasovacia stránka.

23.7 Skripty

```
CREATE SEQUENCE "CUSTOMER_SEQ"
MINVALUE 4 MAXVALUE 9999999999999999999999999999
INCREMENT BY 1
START WITH 4
NOCACHE NOORDER NOCYCLE NOKEEP NOSCALE GLOBAL
/

CREATE OR REPLACE EDITIONABLE TRIGGER "BI_CUSTOMER"
BEFORE INSERT ON CH23_CUSTOMER
FOR EACH ROW
BEGIN
SELECT CUSTOMER_SEQ.nextval INTO :new.customerid
FROM dual;
END;
/

ALTER TRIGGER "BI_CUSTOMER" ENABLE;
/
```

Listing 23.4: Trigger nad zákaznickou sekvenciou.

```
CREATE SEQUENCE "PERSON_SEQ"
MINVALUE 1 MAXVALUE 9999999999999999999999999999
INCREMENT BY 1
START WITH 1
NOCACHE NOORDER NOCYCLE NOKEEP NOSCALE GLOBAL
/
CREATE OR REPLACE EDITIONABLE TRIGGER "BI_PERSON"
BEFORE INSERT ON CH23_PERSON
FOR EACH ROW
BEGIN
SELECT PERSON_SEQ.nextval INTO :new.personid
FROM dual;
:new.email := UPPER(:new.email);
:new.password := hash_password(UPPER(:new.email), :new.password);
END;
/

ALTER TRIGGER "BI_PERSON" ENABLE;
/

CREATE OR REPLACE EDITIONABLE TRIGGER "BU_PERSON"
BEFORE UPDATE ON CH23_PERSON
FOR EACH ROW
BEGIN
:new.email := UPPER(:new.email);
IF :new.password IS NOT NULL THEN
:new.password := hash_password(UPPER(:new.email), :new.password);
ELSE
```

```
:new.password := :old.password;
END IF;
END;
/
ALTER TRIGGER "BU_PERSON" ENABLE;
/
```

Listing 23.5: Trigger nad vkladaním osoby.

```
CREATE OR REPLACE FUNCTION AUTHENTICATE_USER
(p_username in varchar2,
p_password in varchar2)
 return boolean
IS
l_user_name         CH23_PERSON.email%type := UPPER(p_username);
l_password          CH23_PERSON.password%type;
l_hashed_password varchar2(1000);
l_count             number;
BEGIN
SELECT COUNT(*) INTO l_count
FROM CH23_PERSON
WHERE email = l_user_name;

IF l_count > 0 THEN
l_hashed_password := hash_password(l_user_name, p_password);
SELECT password INTO l_password
FROM CH23_PERSON
WHERE email = l_user_name;

IF l_hashed_password = l_password THEN
APEX_UTIL.SET_AUTHENTICATION_RESULT(0);
 return true;
ELSE
APEX_UTIL.SET_AUTHENTICATION_RESULT(4);
 return false;
END IF;

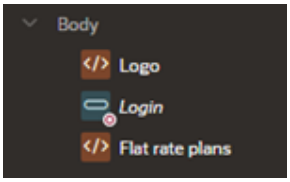
ELSE
APEX_UTIL.SET_AUTHENTICATION_RESULT(1);
 return false;
END IF;
APEX_UTIL.SET_AUTHENTICATION_RESULT(7);
 return false;
EXCEPTION
WHEN OTHERS THEN
APEX_UTIL.SET_AUTHENTICATION_RESULT(7);
APEX_UTIL.SET_CUSTOM_AUTH_STATUS(sqlerrm);
 return false;
END AUTHENTICATE_USER;
/
```

Listing 23.6: Funkcia AUTHENTICATE_USER.

23.8 Vytvorenie domovskej stránky

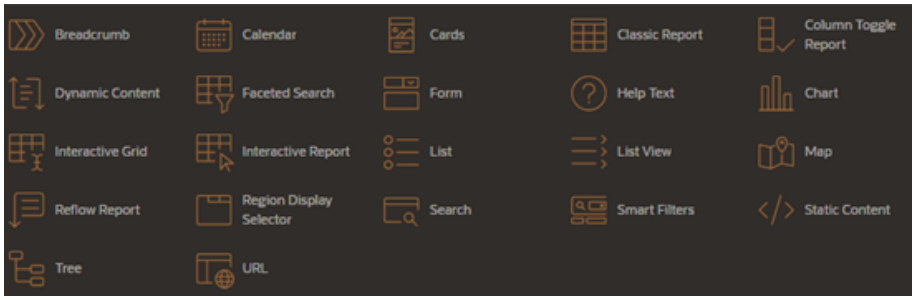
V sekcii vytvorenia stránky (Create a Page), si môžete vybrať z rôznych šablón, prázdnej stránky alebo stránok funkcií s preddefinovanými funkciami, ako sú napríklad Login, stránka About a možnosti konfigurácie. Najprv musíte vyplniť základné informácie, ako je číslo a názov strany a vybrať si model strany, ako napríklad Normal, Modal Dialog alebo Drawer. Môžete tiež vyplniť ďalšie informácie, ako je nastavenie položky navigácie pre stránku (či už to bude horná položka alebo vnorená do existujúcej položky) a výber ikony zobrazenej v ponuke ovládacieho panela.

Po kliknuti na možnosť Create Page sa zobrazí panel na úpravu informácií a obsahu stránky. Hlavný obsah stránky (Body) je rozdelený na regióny.



Obr. 23.18: Regióny časti body.

Každý región môže obsahovať HTML alebo rôzny obsah, ako je statický HTML alebo preddefinovaný obsah.



Obr. 23.19: Zoznam obsahu.

Náš **Login** región obsahuje tlačidlo na presmerovanie stránky do prihlasovacieho formulára používateľa. Vo vlastnosti správania tlačidla sme definovali akciu, ktorá bola nastavená na presmerovanie na stránku tejto aplikácie (Redirect to Page in this Application). Potom nastavíme cieľ (Target) na vopred definovanú prihlasovaciu stránku. Keďže toto tlačidlo nie je potrebné zobrazovať, keď je používateľ už prihlásený, musíme ho skryť nastavením typu, že používateľ je verejný používateľ (User is the Public User) - používateľ sa neautentizoval. Nakoniec sme vytvorili **Flat rate plans** ako región statického obsahu obsahujúci kód HTML, pre ktorý môžeme definovať CSS.



Obr. 23.20: HTML kód statického regiónu.

23.9 Vytvorenie stránky zákazníka

Túto stránku sme vytvorili podobným postupom ako pri vytváraní domovskej stránky. Po vytvorení stránky pokračujeme vo vývoji layoutu zákazníckej stránky, ktorá je rozdelená do niekoľkých oblastí, napríklad pre zobrazenie aktuálneho stavu služieb, nákup doplnkov, zobrazenie faktúr a zobrazenie logov.

Región **Current state** pozostáva z dvoch podregiónov, ktoré zobrazujú číselné údaje – aktuálnu cenu spotrebovaných jednotiek paušálu a príplatkov. Na získanie týchto údajov sa používa SQL dotaz.

```
SELECT name, sum(u.amount) || ' ' || ' / ' || fs.amount || ' ' || s.units AS
    amount
FROM CH23_USAGELOG u
JOIN CH23_CUSTOMERFLATRATE cf ON (u.customerid = cf.customerid AND u.
    flatrateid = cf.flatrateid)
JOIN CH23_CUSTOMER c ON (cf.customerid = c.customerid)
JOIN CH23_PERSON p ON (c.personid = p.personid)
JOIN CH23_SERVICE s ON (u.serviceid = s.serviceid)
JOIN CH23_FLATRATESERVICE fs ON (fs.flatrateid = cf.flatrateid AND fs.serviceid = s
    .serviceid )
WHERE email = :APP_USER
AND EXTRACT(month from when) = EXTRACT(month from sysdate)
GROUP BY s.serviceid, s.name, s.units, fs.amount;
```

Listing 23.7: SQL dotaz - Spotrebované jednotky paušálnej sadzby.

```
SELECT name, SUM(amount) || ' ' || units AS amount
FROM CH23_EXTRASERVICE JOIN CH23_CUSTOMER USING(customerid)
JOIN CH23_PERSON USING (personid)
JOIN CH23_SERVICE USING (serviceid)
WHERE email = :APP_USER
AND EXTRACT(month from startDate) = EXTRACT(month from sysdate)
GROUP BY serviceid, name, units;
```

Listing 23.8: SQL dotaz - Spotrebované jednotky extra služieb.

Región **Buy extra** obsahuje číselné pole, výberový zoznam a tlačidlo. Je možné nastaviť sumu a vybrať si, ktorú extra službu si chceme dokúpiť. Ovláda sa pomocou tlačidla, ktoré spúšťa proces s kódom PL/SQL:

```
INSERT INTO CH23_EXTRASERVICE VALUES (
:BUY_TYPE,
(SELECT customerid
FROM CH23_CUSTOMER JOIN CH23_PERSON USING (personid)
WHERE email = :APP_USER),
sysdate ,
:BUY_AMOUNT
);
```

Listing 23.9: PL/SQL kód pre vloženie extra služby.

Región **Show invoices** pozostáva z tlačidla používaného na generovanie faktúr. Vo vlastnosti *Behaviour* tlačidla sme definovali akciu (*Action*), ktorá bola nastavená na presmerovanie na stránku tejto aplikácie - *Redirect to Page in this Application*. Potom nastavíme cieľ - *Target* na preddefinovanú stránku faktúry zákazníka, ktorá bola nastavená ako modálna. Na tejto stránke sa pomocou SQL dotazu zobrazí klasický report, ktorý je možné aj vytlačiť:

```
SELECT f.price || ' €' as "Fixed Flatrate ",
SUM(CASE e.serviceid WHEN 1 THEN NVL(e.amount, 0) ELSE 0 END)
|| ' ' || listagg (CASE s.serviceid WHEN 1 THEN s.units ELSE '' END, '') AS "Extra
    SMS",
SUM(CASE e.serviceid WHEN 2 THEN NVL(e.amount, 0) ELSE 0 END)
|| ' ' || listagg (CASE s.serviceid WHEN 2 THEN s.units ELSE '' END, '') AS "Extra
    Minutes",
SUM(CASE e.serviceid WHEN 3 THEN NVL(e.amount, 0) ELSE 0 END)
|| ' ' || listagg (CASE s.serviceid WHEN 3 THEN s.units ELSE '' END, '') AS "Extra
    Data",
SUM(CASE e.serviceid WHEN 4 THEN NVL(e.amount, 0) ELSE 0 END)
|| ' ' || listagg (CASE s.serviceid WHEN 4 THEN s.units ELSE '' END, '') AS "Extra
    MMS",
(f.price
+ SUM(CASE e.serviceid WHEN 1 THEN NVL(e.amount, 0) * s.price ELSE 0 END)
+ SUM(CASE e.serviceid WHEN 2 THEN NVL(e.amount, 0) * s.price ELSE 0 END)
+ SUM(CASE e.serviceid WHEN 3 THEN NVL(e.amount, 0) * s.price ELSE 0 END)
+ SUM(CASE e.serviceid WHEN 4 THEN NVL(e.amount, 0) * s.price ELSE 0 END)
|| ' €') AS "Total"
FROM CH23_CUSTOMER c
JOIN CH23_PERSON p ON (p.personid = c.personid)
JOIN CH23_CUSTOMERFLATRATE cf ON (cf.customerid = c.customerid)
JOIN CH23_FLATRATE f ON (f.flatrateid = cf. flatrateid )
JOIN CH23_FLATRATESERVICE fs ON (fs.flatrateid = f.flatrateid)
JOIN CH23_SERVICE s ON (s.serviceid = fs.serviceid)
LEFT JOIN CH23_EXTRASERVICE e ON (e.customerid = cf.customerid)
WHERE p.email = :APP_USER
AND (e. startdate IS NULL OR
(EXTRACT(month from e.startdate) = EXTRACT(month from sysdate)
AND EXTRACT(year from e.startdate) = EXTRACT(year from sysdate)))
GROUP BY f.price;
```

Listing 23.10: SQL dotaz pre report faktúry.

Región **Log** obsahuje interaktívny report na zobrazenie logu používania pomocou SQL dotazu:

```
SELECT name, phone, amount || ' ' || units AS amount, when
FROM CH23_USAGELOG JOIN CH23_CUSTOMER USING (customerid)
JOIN CH23_PERSON USING (personid)
JOIN CH23_SERVICE USING (serviceid)
WHERE email = :APP_USER
ORDER BY when DESC;
```

Listing 23.11: SQL dotaz – Log používania.

Záložka **Stats** je nová stránka pozostávajúca z troch oblastí, ako sú minúty, SMS a dáta. Typ oblasti je nastavený na statický obsah - *Static Content*, vrátane komponentov *Chart* zobrazujúcich grafy zo série. Jeho zdroj je definovaný pomocou SQL dotazu. Štandardne je farba pruhov grafu nastavená na jednotnú. Dá sa upraviť v SQL dotaze:

```
SELECT TO_CHAR(when, 'MM/YYYY') AS "Month",
SUM(amount) AS Amount,
CASE WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '01'
THEN '#F8B195'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '02'
THEN '#F67280'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '03'
THEN '#C06C84'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '04'
THEN '#6C567B'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '05'
THEN '#ACDEAA'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '06'
THEN '#97D0ED'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '07'
THEN '#C23B23'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '08'
THEN '#6B7B8E'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '09'
THEN '#D5AF26'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '10'
THEN '#9AB5C1'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '11'
THEN '#74698C'
WHEN SUBSTR(TO_CHAR(when, 'MM/YYYY'), 0, 2) = '12'
THEN '#C1867B'
ELSE 'black'
END AS MY_COLOR
FROM CH23_USAGELOG
JOIN CH23_CUSTOMER USING (customerid)
JOIN CH23_PERSON USING (personid)
JOIN CH23_SERVICE USING (serviceid)
WHERE email = :APP_USER AND serviceid = 2
GROUP BY TO_CHAR(when, 'MM/YYYY')
ORDER BY 1;
```

Listing 23.12: SQL dotaz - Štatistika.

23.10 Vytvorenie stránky manažéra

V manažérskej stránke sme na vytváranie podstránok použili podobné metódy a komponenty. Každá podstránka sa vytvorí ako nová stránka, napríklad pridanie zákazníka (Add customer), spravovanie zákazníkov (Manage customers) a zobrazenie zákazníkov (View customers).

Stránka *Add customer* obsahuje jednoduchý formulár, ktorý je vytvorený z komponentov. Proces pridania nového zákazníka sa vykonáva pomocou PL/SQL kódu:

```
INSERT INTO CH23_PERSON (EMAIL, PASSWORD, FIRSTNAME, LASTNAME,
    ROLE)
VALUES (:CUSTOMER_EMAIL, :CUSTOMER_PASSWORD, :
    CUSTOMER_FIRSTNAME,
:CUSTOMER_LASTNAME, 'C');
```

Listing 23.13: PL/SQL kód pre vkladanie osoby.

Stránka *Manage customers* je vytvorená s dvomi interaktívnymi gridmi: Správa osôb (Manage persons) a správa aktuálnych zákazníkov (Manage current customers). Interaktívne gridy sú prepojené s procesom editovania a ukladania zmien. Typ procesu je nastavený na automatické spravovanie riadkov - *Interactive Grid – Automatic Row Processing (DML)*.

Stránka *View customers* obsahuje interaktívny report na zobrazenie údajov bez ich úpravy. Tento report je vytvorený pomocou SQL dotazu:

```
SELECT firstname, lastname, email, address, phone
FROM CH23_PERSON
JOIN CH23_CUSTOMER USING (personid);
```

Listing 23.14: SQL dotaz - Zákazníci.

23.11 Doplnkové výučbové materiály

Môžete nájsť nasledujúce doplnkové výučbové materiály:

- skript pre vytvorenie a naplnenie tabuliek
- skript pre zrušenie tabuliek
- exportovanú zabalenú aplikáciu
- video, ktoré ukazuje, ako vytvoriť aplikáciu

Všetky doplnkové výučbové materiály sú dostupné na public BeeAPEX project page. Prihláste sa ako hosťovský používateľ (nevyžaduje heslo). Nájdete učebnicu v sekcii Books, skripty v priečinku Part 2 > Chapter23 v sekcii Scripts a video návod v Collection of video guides. Materiály pre krátke kurzy sa nachádzajú v sekcii Short courses.

23.11.1 Exportovaná aplikácia

Exportovaná aplikácia je zabalená. Inštalácia vytvorí tabuľky, indexy, funkcie, procedúry a triggre a taktiež ich naplní dátami. Odinštalovanie odstráni všetky databázové objekty použité v tejto aplikácii.

23.11.2 Video návody

Video návod ukazuje každý krok tvorby aplikácie.

23.12 Otázky

1. Ktorá podstránka sa používa na vytvorenie novej prihlasovacej schémy?
2. Uveďte aspoň dve metódy autentifikácie pre správu používateľov.

3. Počas implementácie sa v príkazoch Select odkazovalo na rôzne položky. Ako tieto položky môžete identifikovať?

23.13 Odpovede

1. V nástroji Oracle Apex si na podstránke Shared Components => Authentication Schemes zvolíme spôsob autentifikácie - Authentication Scheme.
2. Existuje viacero metód, ktoré možno použiť na autentifikáciu používateľa, ako napríklad server LDAP, databázový účet, účet Oracle APEX atď.
3. Na položky v príkazoch Select sa odkazuje pomocou dvojbodky na začiatku názvu položky.



24. Prípad prenájmu automobilov

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

24.1 Biznis pohľad prípadu

Prenájom automobilu otvára svet cestovných možností, ale existujú aj určité úskalia, ktorým sa treba vyhnúť. Prenájom automobilov je skôr ako rezervácia hotelovej izby než rezervácia letenky. Existujú dva typy poplatkov: ceny, ktoré platíte pri prebratí automobilu a predplatené ceny.

Pre tých, ktorí si automobil prenajímajú prvý krát, je zvyčajne najlepšou možnosťou zajednať si platbu neskôr (“Pay Later”). Tá umožňuje väčšiu flexibilitu v prípade ak potrebujete zmeniť svoje plány. Tento typ prenájmu zvyčajne nevyžaduje žiadne pokuty za zrušenie, čo môže byť veľmi užitočné ak sa objavia neočakávané okolnosti.

Na to aby sme sa vyhli dodatočným poplatkom, je dôležité pozorne si preštudovať nájomnú zmluvu a všetky dodatočné poplatky, ktoré sa môžu objaviť ako sú napríklad poistenie a dodatočné poplatky za vodiča. Okrem toho pred opustením požičovne dôkladne skontrolujte požičiavané vozidlo a akékoľvek poškodenie nahláste ihneď požičovni, aby ste sa vyhli neskoršej zodpovednosti.

Pochopením procesu prenájmu vozidla a prípadných poplatkov si môžete zabezpečiť hladký a bezstarostný zážitok z prenájmu a naplno si užiť slobodu a flexibilitu, ktorú takéto požičanie vozidla môže počas cestovania ponúknuť.

24.2 Definícia problému

Aj keď sa môže zdať, že pri prenájme automobilu existuje veľa pravidiel a podmienok, je to z dobrého dôvodu. Nové autá stoja v priemere okolo 36 eur. Keď si požičiate auto, platíte len zlomok jeho hodnoty za používanie.

Aj keď je dôležité mať vhodné poistenie pre prípad nehody alebo nepredvídateľných okolností, väčšina skúseností s prenájmom bude bez incidentov. Avšak, stále je dôležité dôkladne si prečítať nájomnú zmluvu, pochopiť obchodným podmienkam a skontrolovať vozidlo ešte pred podpísaním zmluvy, či sa na ňom nenachádza nejaké poškodenie, aby sme sa neskôr vyhli potencionálnym problémom.

Ak sa budete riadiť týmito odporúčaniami, budete mať bezpečnú a príjemnú skúsenosť s prenájmom vozidla, bez nečakaných prekvapení. A pamätajte si, pravidlá a podmienky sú na to, aby chránili obe strany, vás a aj spoločnosť, ktorá vozidlá prenajíma. Preto je najlepšie tieto podmienky dodržiavať, aby bol prenájom bezproblémový.

24.3 Prípady použitia

24.3.1 Naratívny popis

Cieľom tohto projektu je vytvoriť funkčnú aplikáciu, v ktorej budeme simulovať prevádzku autopožičovne. Táto aplikácia zahŕňa správu zákazníkov, áut, prenájmov, ako aj podrobné informácie o týchto entitách pomocou platformy Oracle Apex.

Aplikácia je navrhnutá tak, aby bola užívateľsky prívetivá pre vlastníka autopožičovne. Umožňuje mu manažovať všetky potrebné informácie, zahŕňajúc prenájmy, dostupné vozidlá na prenájom a údaje o zákazníkovi. Používateľ bude mať možnosť prenajať si vozdilo tak, že si vyberie auto na základe rozličných kritérií ako je výrobca, model, rok, typ paliva a taktiež budú zobrazené všetky ďalšie dôležité detaily, ktoré sú potrebné pre dokončenie prenájmu.

Okrem toho aplikácia zobrazí detailne údaje o súčasných prenájmoch, dostupných vozidlách na prenájom a zákazníckych údajoch. Aktuálne prenájmy sa zobrazia používateľovi v tabuľkách, v závislosti od toho, v akej karte sa nachádza.

Celkovo aplikácia poskytne efektívny a organizovaný systém na manažovanie autopožičovne spolu s detailnými informáciami dostupnými po kliknuté na tlačidlo. Užívateľsky prívetivé rozhranie a obsiahle nástroje na manažovanie dát, uľahčia a zefektívnia prevádzku autopožičovne.

24.3.2 Pološtruktúrovaný popis

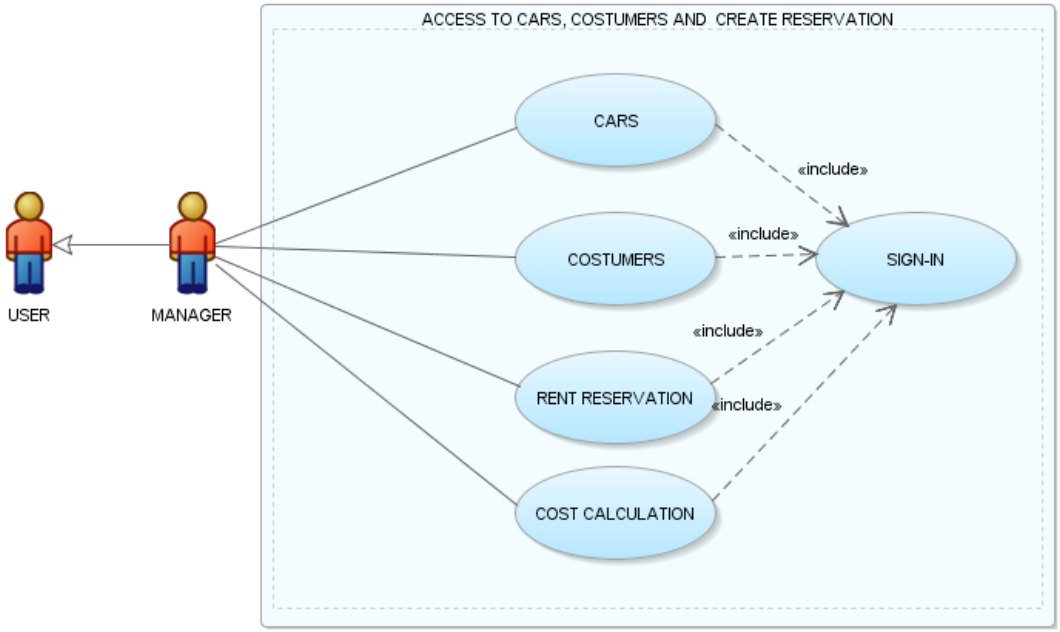
Tento projekt je zameraný na vývoj užívateľsky prívetivej aplikácie, ktorá využíva Oracle Apex na riadenie detailných informácií o zákazníkoch, autách a prenájmoch pre autopožičovňu. Táto aplikácia umožní používateľom robiť prenájmy na základe rozličných kritérií, zobrazovať informácie o prenájme a poskytovať obsiahlu paletu nástrojov na riadenie dát určenú k zobrazovaniu súčasných prenájmov, dostupných vozidiel na prenájom a zákaznických údajov. Poskytnutím efektívneho a organizovaného systému na manažovanie autopožičovne, aplikácia uľahčí a zefektívni prevádzku autopožičovne. Pološtruktúrovaný popi je poskytnutý v tabuľke 24.1.

Tabuľka 24.1: Popis prípadu použitia: Sprístupňovanie vozidiel, zákazníkov a rezervácií prenájmov áut

Kľúčové slovo	**Hodnota**
ID:	*Ch24-01*
Názov:	*Sprístupnenie zoznamu áut a zákazníkov, schopnosť pridávať a odstraňovať dáta a vytvárať nové rezervácie na prenájom vozidla výberom preferovaných dátumov a výpočet nákladov.*
Popis:	*Používateľ využije Apex aplikáciu na prístup a prezeranie si vozidiel, ktoré patria spoločnosti, rovnako ako aj všetkých zákazníkov, ktorí predtým vykonali rezerváciu. Autá rovnako ako aj zákazníci sú popísané množinou atribútov. Okrem toho, používateľ má možnosť vytvárať, modifikovať a mazať dáta z oboch kategórií, čo umožňuje efektívny manažment systému autopožičovne. Naviac môže používateľ vytvoriť novú rezerváciu na prenájom vozidla, pridaním vozidla, zákazníka a dátumov. Systém vypočíta celkové náklady rezervácie na základe zvolených dátumov a modifikuje podľa toho výsledky. Záverečný report zobrazuje podrobnosti o rezervácii, zahŕňajúc vozidlo, zákazníka, dátumy prenájmu a celkové náklady.*
Hlavný aktér:	*Správca používateľov*
Predpoklady:	Používateľ má správcovské konto pre APEX inštanciu.
Podmienky po skončení:	Po modifikácii a výbere auta a zákazníka, používateľ môže vytvoriť rezerváciu na prenájom vozidla vo vnútri APEX inštancie. Po dokončení rezervácie prenájmu a výpočte celkových nákladov.
Hlavné	*Scenáre*
Úspešný scenár:	1. Otvorte webový prehliadač a prihláste sa do aplikácie autopožičovne. 2. Vyberte položku ponuky alebo navigáciu stránky **Cars** (autá) alebo **Customers** (zákazníci). 3. Na pridanie auta kliknite na tlačidlo **"Create"** (vytvoriť), rovnako tak môžete spraviť aj pre zákazníkov. 4. Na úpravu auta alebo zákazníka, kliknite na symbol ceruzky v konkrétnom zázname. 5. Na pridanie alebo zachovanie údajov, kliknite na tlačidlo **"Apply changes"** (použiť zmeny) alebo na odstránenie stlačte tlačidlo **"Delete"** (zmazať). 6. Vyberte položku z menu alebo navigácie stránky **"Rentcar"**. 7. Na vytvorenie nového prenájmu kliknite **"Create"**. 8. Na pridanie auta do rezervácie, kliknite na **carid** a vyberte si zo zoznamu dostupných áut. 9. Na pridanie zákazníka k rezervácii vozidla kliknite na **custid** a vyberte zo zoznamu. 10. Na nastavenie dátumov rezervácie kliknite na **rent date** a vyberte preferovaný čas začiatku a rezervácie. 11. Potom kliknite na pole **return date** a vyberte dátum, kedy prenájom vozidla skončí. 12. Keď je auto vrátené, celkové náklady rezervácie sú vypočítane na základe počtu dní prenájmu a dodatočných poplatkov. Ak je auto poškodené, bude zaúčtovaný extra doplatok na pokrytie nákladov na opravu. 13. Finálna cena je zobrazená zákazníkovi spolu s detailným rozdelením poplatkov, v reporte o prenájme.
Rozšírenia:	*Žiadne*
Frekvencia použitia:	*Počet rezervácií za deň sa líši v závislosti od turistickej sezóny, ale v priemere môžeme očakávať, že obdržíme približne 5 rezervácií za deň.*
Stav:	*Dokončené*

24.3.3 Diagram prípadov použitia

Hore uvedený priebeh je zobrazený na diagrame prípadov použitia (pozri Obrázok 24.1).



Obr. 24.1: Diagram prípadu použitia.

24.4 Dátový model

24.4.1 Textový popis dátového modelu

Výučbový projekt je implementovaný za použitia troch tabuliek, ktoré reprezentujú entity zahŕňajúc: tabuľku áut, tabuľku používateľov a tabuľku prenájmu vozidiel. Tieto tabuľky obsahujú všetky potrebné atribúty a údaje, potrebné k riadeniu procesu prenájmu vozidla prostredníctvom APEX aplikácie. Vzdelávací projekt sa skladá z troch tabuliek, ktoré popisujú jednotlivé entity.

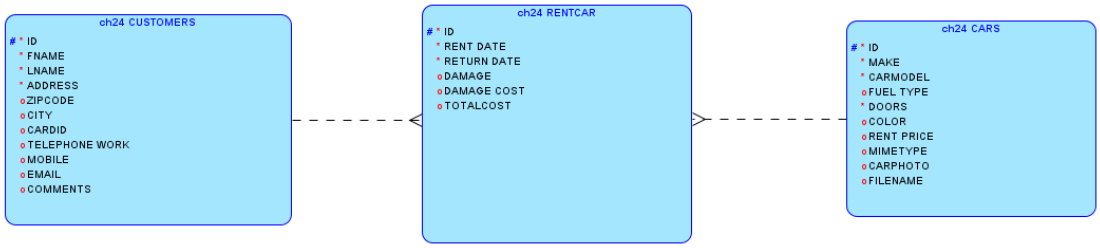
1. Tabuľka Cars (Autá)
   - **id** identifikačné číslo a primárny kľúč,
   - **make** výrobca auta,
   - **carmodel** model auta,
   - **fuel** typ paliva: benzín, nafta, LNG alebo elektrina,
   - **doors** počet dverí auta,
   - **color** farba auta,
   - **rent** cena za jednodňový prenájom konkrétneho auta,
   - **carphoto** fotografia auta,
2. Tabuľka Customers (Zákazníci)
   - **id** identifikačné číslo a primárny kľúč zákazníka,
   - **fname** meno zákazníka,
   - **lname** priezvisko zákazníka,
   - **address** adresa zákazníka,
   - **zipcode** poštové smerové číslo (PSČ) adresy,
   - **city** mesto zákazníka,
   - **cardid** číslo vodičského alebo občianskeho preukazu zákazníka,
   - **telephone-work** telefónne číslo zákazníka do práce,

- **mobile** mobilné telefóne číslo zákazníka,
- **email** e-mail zákazníka,
- **comments** komentáre súvisiace so správaním osoby ako zákazníka.

3. Tabuľka CarRental (Prenájom áut)
   - **id** identifikačné číslo a primárny kľúč tabuľky prenájmu,
   - **carid id** prepojenie medzi tabuľkou áut a tabuľkou prenájmu,
   - **custid id** prepojenie medzi tabuľkou zákazníkov a tabuľkou prenájmu,
   - **rent date** dátum začiatku prenájmu,
   - **return date** dátum vrátenia auta,
   - **damage** popis poškodenia auta, ak k nemu došlo,
   - **damage cost** náklady na opravu poškodenia,
   - **total cost** celkové náklady za prenájom.

24.4.2 Logický dátový model

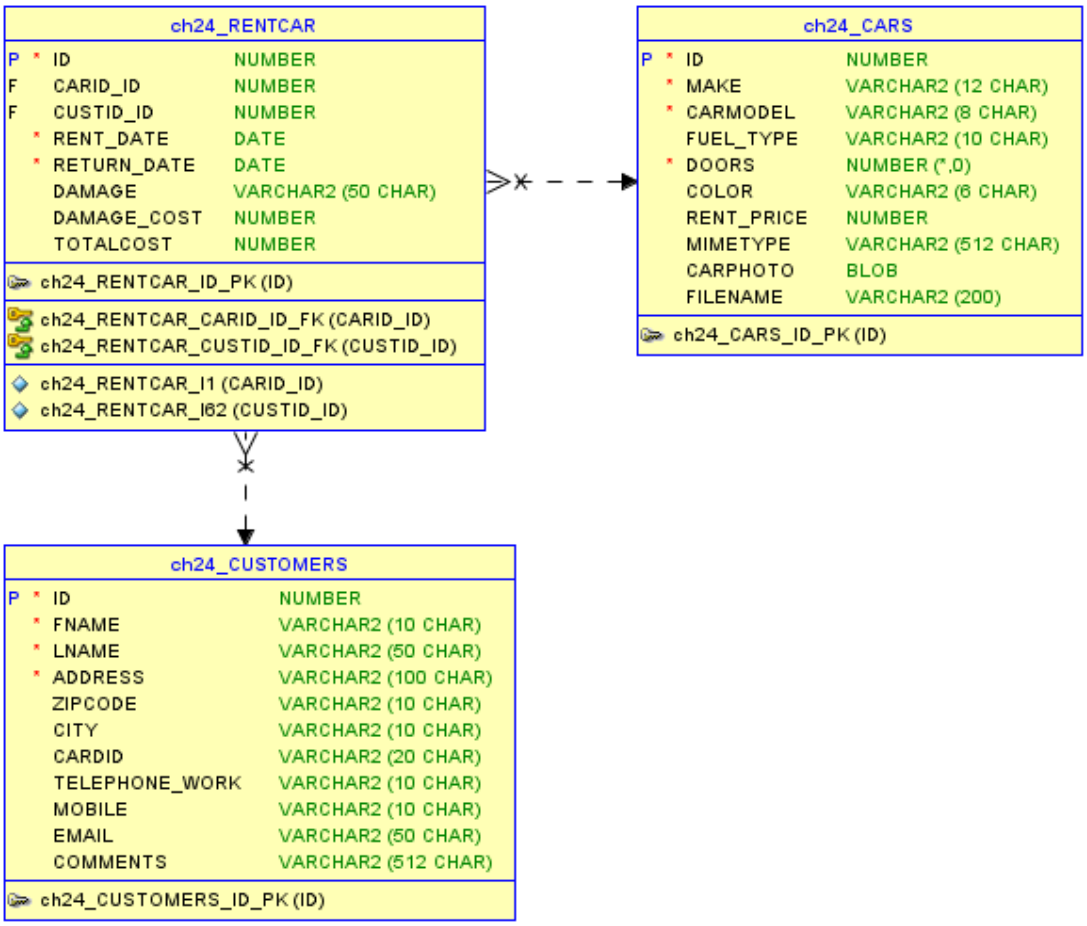
Logický dátový model na Obrázku 24.2.



Obr. 24.2: Logický model projektu autopožičovne.

24.4.3 Relačný dátový model

Relačný dátový model na Obrázku 24.3.



Obr. 24.3: Relačný model projektu autopožičovne.

24.4.4 SQL skript

Tabuľky, ktoré vytvárame pre našu aplikáciu, sú CH24_CARS, CH24_CUSTOMERS a CH24_RENTCAR. Polia každej tabuľky by sú popísané nižšie:

```
drop table  ch24_cars;
drop table  ch24_customers;
drop table  ch24_rentcar;

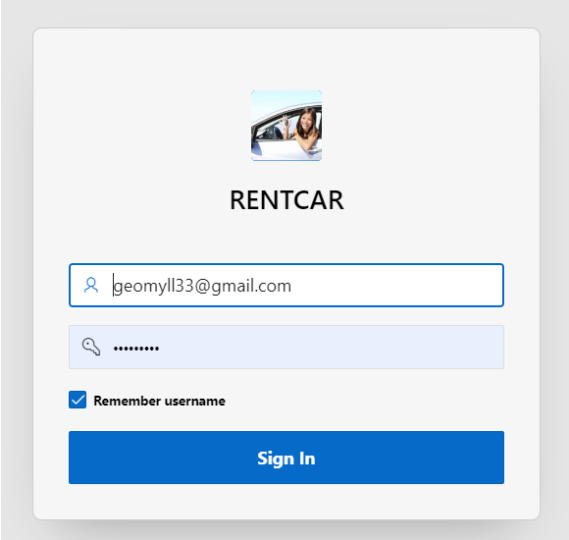
-- create tables
create table ch24_cars (
id                    number generated by default on null as identity
constraint ch24_cars_id_pk primary key,
make                  varchar2(12 char) not null,
carmodel              varchar2(8 char) not null,
fuel_type             varchar2(10 char),
doors                 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);
```

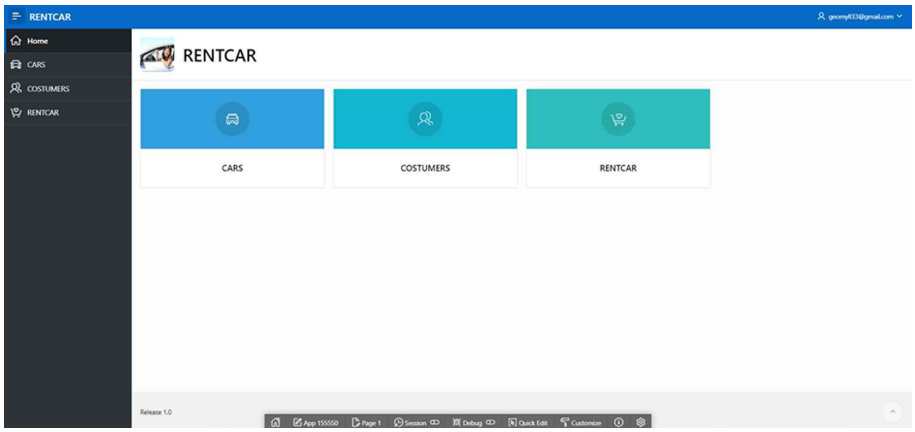
24.5 Rozhrania aplikácie

Používateľ, ktorý je majiteľom spoločnosti na prenájom vozidiel, potrebuje poskytovať svoje prihlasovacie údaje na to, aby mal prístup k aplikácii. Aplikácia využíva Application Express Accounts autentifikáciu. Prihlasovacie údaje vlastníka sú rovnaké ako prihlasovacie údaje pre WORKSPACE, zobrazené na 24.4.



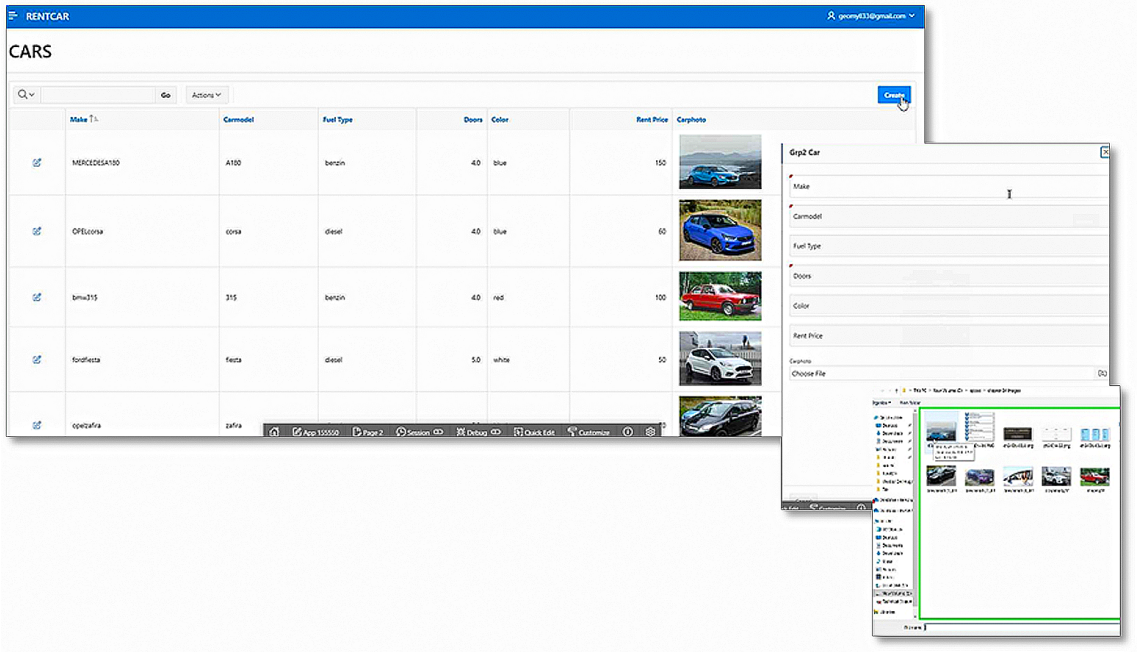
Obr. 24.4: Prihlásenie do aplikácie.

Aplikácia simuluje prevádzku autopožičovne. Zahrňuje správu zákazníkov, áut a prenájmov, ako aj podrobné informácie o týchto entitách pomocou používateľsky prívetivej webovej aplikácie založenej na platforme Oracle Apex, ako je vyobrazené na Obrázku 24.5.



Obr. 24.5: Domovská stránka aplikácie.

Šablóna áut na Obrázku 24.6 je navrhnutá na manažovanie informácií súvisiacich s vozidlami v autopožičovni. Používateľ, ktorý je majiteľom autopožičovne, môže spravovať a ukladať všetky potrebné informácie o autách, ako je značka, model, rok, typ paliva, ako aj fotografiu auta, ktorá zobrazuje jeho stav.

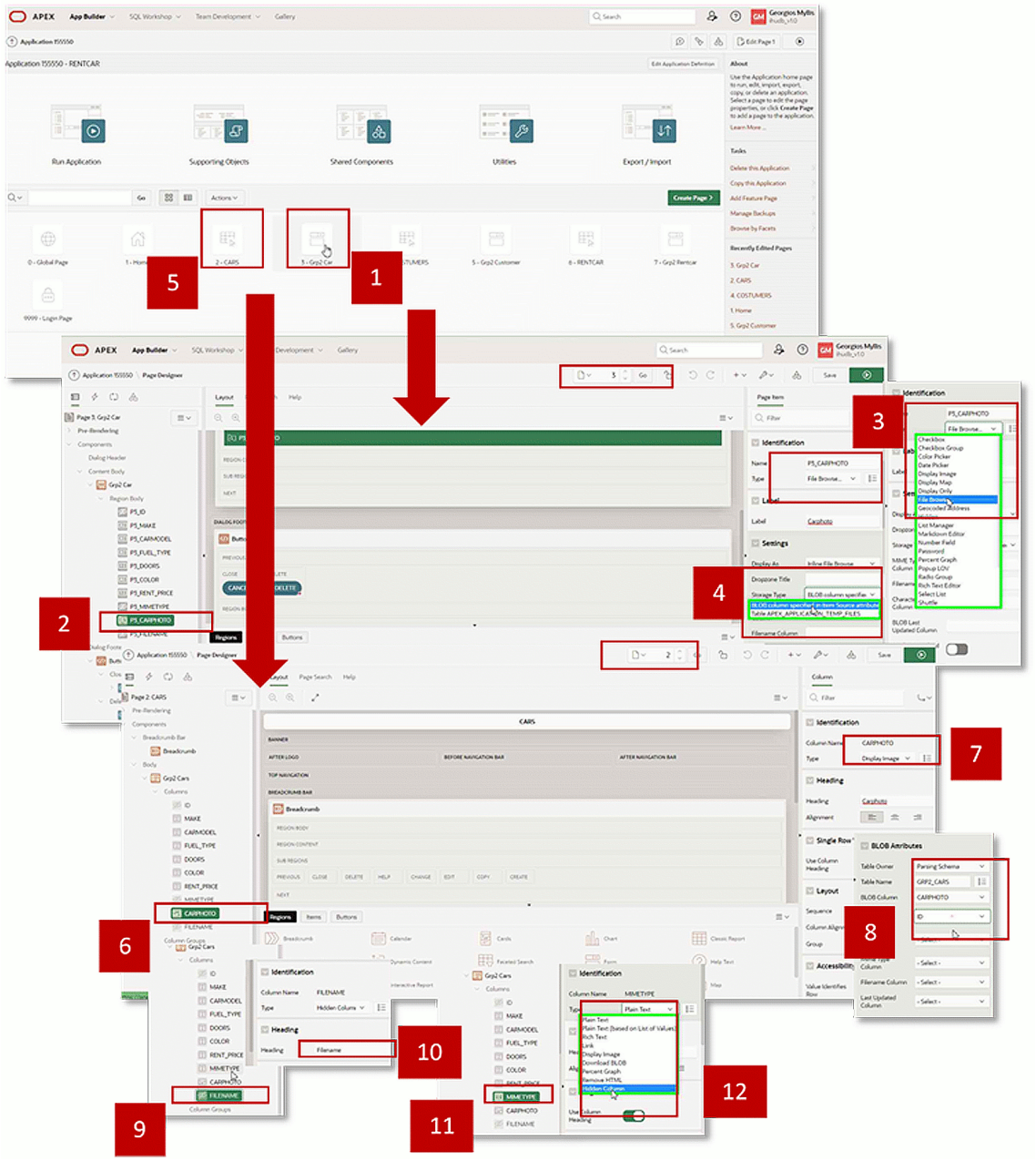


Obr. 24.6: Šablóna Cars (Autá).

Na nasledujúcom Obrázku 24.7 je vysvetlené, ako je možné vytvoriť stĺpec vo vašom formulári, ktorý bude môcť zobrazovať fotografie. Aby ste to dosiahli, potrebujete mať ešte dva ďalšie pomocné stĺpce, ktoré uložia informácie do tabuľky, jeden pre mime a jeden pre názov súboru. Stĺpec, ktorý bude ukladať fotografiu, musí byť nastavený na typ úložiska BLOB, v našom prípade stĺpec **CARPHOTO**.

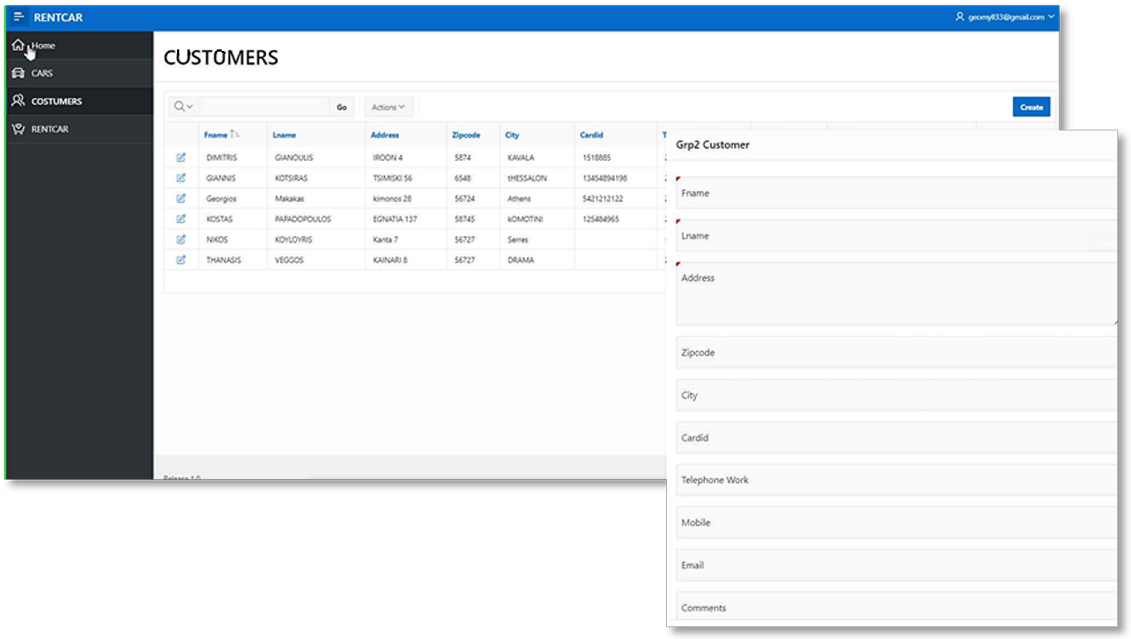
V návrhárovi stránky aplikácie je potrebné prejsť na report pre šablónu CAR a nakonfigurovať stĺpec **CARPHOTO** ako file browser (prehliadač súborov), potom zmeniť nastavenia typu úložiska na **“LOB column specifier in item Source attributes”** (Špecifikátor stĺpca BLOB v položke Zdroj atribútov).

Následne je potrebné prejsť na formulár pre šablónu CAR a nakonfigurovať stĺpec **CARPHOTO** ako (Display image), potom nakonfigurovať BLOB atribúty spájajúce tabuľku **CH14_CARS**, stĺpca **CARPHOTO** a primárneho kľúča **ID**. Nakoniec, stĺpce **MIMETYPE** a **FILENAME** nastavte na skryté.



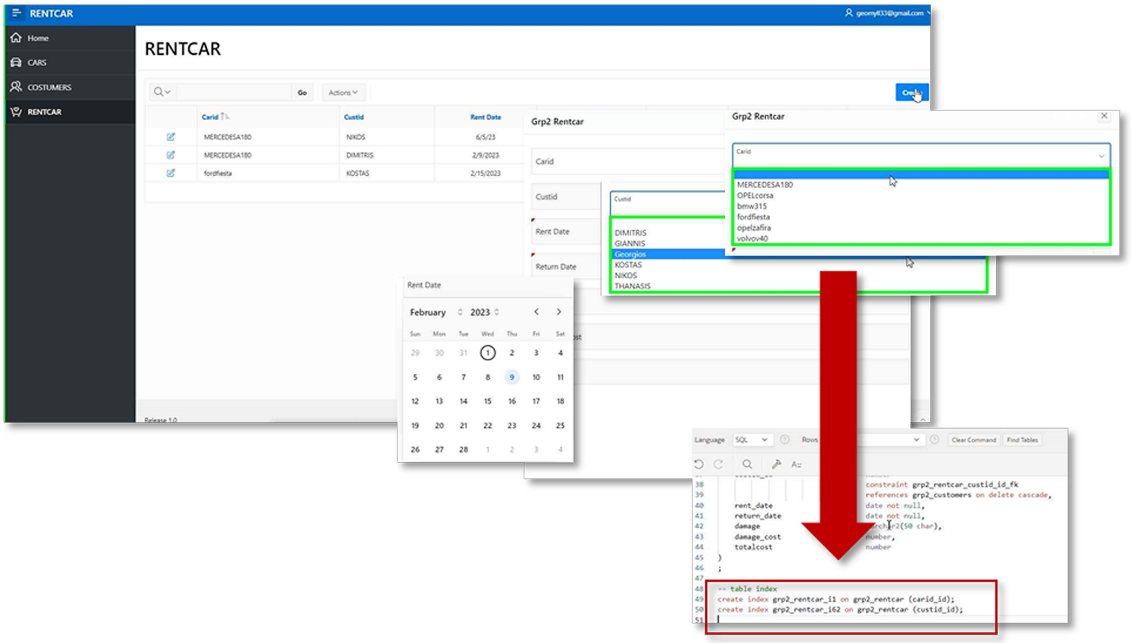
Obr. 24.7: Ako vytvoriť stĺpec s fotografiami.

Podobné údaje, ako pre šablónu zákazníkov, ktoré je možné uložiť, sú zobrazené na Obrázku 24.8.



Obr. 24.8: Šablóna Customers (Zákazníci).

V šablóne prenájmu, ako je to vyobrazené na Obrázku 24.9, môže používateľ vybrať auto na základe kritérií ako je výrobca, model a rok. Prenájom je potom prepojený so zákazníkom a všetky potrebné podrobnosti, potrebné k takému vytvoreniu prenájmu vozidla, ako je prenajímané vozidlo, zákazník, dátum začiatku a dátum konca prenájmu a celkové náklady budú zobrazené. Používateľ môže taktiež pridať dodatočné náklady, ak malo vozidlo nejaké poškodenie po tom, čo bolo vrátené. Keď je prenájom dokončený, celkové náklady sú vypočítané a zobrazené vo výslednom reporte.



Obr. 24.9: Šablóna Rentcar (Prenájom áut).

24.6 Doplnkové výučbové materiály

Všetky doplnkové výučbové materiály sú dostupné na public BeeAPEX project page. Prihláste sa ako hosťovský používateľ (nevyžaduje heslo). Nájdete učebnicu v sekcii Books, skripty v priečinku Part 2 > Chapter24 v sekcii Scripts a video návod v Collection of video guides. Materiály pre krátke kurzy sa nachádzajú v sekcii Short courses.

24.6.1 Exportovaná aplikácia

Exportovaná aplikácia je zabalená. Inštalácia vytvorí tabuľky, indexy, funkcie, procedúry a triggre a taktiež ich naplní dátami. Odinštalovanie odstráni všetky databázové objekt použité v tejto aplikácii.

24.6.2 Video návody

Video návod ukazuje každý krok tvorby aplikácie.

24.7 Otázky

1. Ako môžete pridať ikonu do loga vašej aplikácie, keď ste ju už vytvorili?
2. Ako môžete pridať obrázky na stránky vašich reportov?

24.8 Odpovede

1. Vo vašom prostredí aplikácie vyberte **Shared Components** (Zdieľané komponenty), potom vyberte **User Interface Attributes** (Atribúty používateľského rozhrania), v kontextovom okne kliknite na **Edit** (Upraviť) a pridajte obrázok podľa vášho výberu.
2. Najskôr nastavte dátový typ vášho stĺpca na BLOB, potom prejdite do návrhára stránky aplikácie šablóny reportu a nastavte stĺpec ako prehliadač súborov, potom zmeňte nastavenia typu úložiska na **BLOB column specifier in item Source Attributes** (Špecifikátor stĺpca BLOB v položke Zdroj atribútov). Následne je potrebné prejsť do formulára tejto šablóny, ktorý sa nachádza opäť v prostredí aplikácie, kde nakonfigurujte stĺpec ako **Display image** (Zobraziť obrázok), potom nakonfigurujte atribúty BLOB spájajúce tabuľku, stĺpca a primárneho kľúča ID, ktorý sa týka tohto stĺpca.

Literatúra

Články

[1] Hill Kim. „Altruistic cooperation during foraging by the Ache, and the evolved human predisposition to cooperate“. In: *Human Nature* 13 (mar. 2002), strany 105–128. DOI: 10.1007/s12110-002-1016-3 (citované na strane 187).

Knihy

[2] Roy Fielding a Richard N. Taylor. *Principled design of the modern Web architecture*. Editované Mehdi Jazayeri andAlexander L. Wolf Carlo Ghezzi. Association for Computing Machinery, 2010. ISBN: 978-1-58113-206-9. DOI: 10.1145/337180.337228 (citované na stranách 104, 105).

[3] Dariusz Jemielniak a Aleksandra Przegalinska. *Collaborative society*. MIT Press, 2020 (citované na strane 187).

[4] OMG. *Business Process Model and Notation (BPMN), Version 2.0*. 2011. URL: http://www.omg.org/spec/BPMN/2.0 (citované na stranách 264, 265).

[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 (citované na strane 263).

[6] A. Osterwalder a Y. Pigneur. *Business model generation: a handbook for visionaries, game changers, and challengers*. Zväzok 1. John Wiley a Sons, 2010 (citované na stranách 37, 38).

Register

access control, 86
ako pripraviť databázu, 49
Ako sa pohybovať v nástroji APEX, 81
ako si vymieňať dáta v APEX-e, 93
ako vygenerovať prvý návrh aplikácie, 113
Ako vytvoriť report, 338
ako využiť galériu štartovacích/vzorových aplikácií a pluginov?, 199
Answers, 345
APEX docker, 44
application programming interface, API, 105
atribút, 51
AutoREST, 109
autorizácia používateľov, 247

bezstavový REST, 105
Bill-of-material, 311
Biznis pohľad prípadu, 227, 237, 253, 263, 293, 299, 311, 327, 347, 389
biznis pohľad prípadu, 247
Business process management, 263

computation, 87
create application, 87
create application wizard, 84
create page wizard, 86
CRUD operácie, 105, 108
cudzí kľúč, 56

data manipulation, 68
Data model, 232, 273, 302, 351
Data Modeler, 61
data workshop, 94, 100
databáza, 50
databázový systém, 49
DB, 50
DB schéma, 54
DB vrstva, 78
DB-vrstva, 50
DBMS, 50
DBS, 49
DDL, 60
Definícia problému, 227, 237, 253, 293, 299, 311, 327, 348, 389
definícia problému, 247
Diagram prípadov použitia, 238, 312, 328, 392
diagram prípadov použitia, 248
DML, 68
doména, 51
DQL, 70
druhá normálna forma, 2NF, 59
dátová štruktúra, 50
Dátový model, 240, 254, 294, 315, 328, 392
dátový model, 50, 248
dátový typ, 51

entita, 51
entitno-relačný diagram, ERD, 50
entity relationship diagram, ERD, 51
ER, 50
ER diagram, 50, 51
export wizard, 100
export údajov, 100

generovanie RDB schémy, 55
generovanie relačného modelu, 56
graf, 87

How to, 35
how to collaborate in team, 187
HTTP request, 105, 108
HTTP response, 106
https://apex.oracle.com, 43

import údajov, 94
install Flows for APEX, 266
interaktívna zostava, 87
inštancia entity, 54
item, 87

kalendár, 87
kardinalita, 51

list of values, LOV, 212
load wizard, 96
Logický dátový model, 240, 328, 393
logický dátový model, 50, 250
logický model, 50

logika aplikácie, 87

multiple languages, 219

Naratívny popis, 228, 253, 300, 315, 327, 349, 368, 390
normalizácia, 59

object browser, 67, 101
obslužný program zdroja, 108
Odpovede, 400
Office Hours Scheduling, 347
On-premise, 43
Oracle Academy, 47
ORACLE APEX, 81
ORACLE application builder, 83
Oracle Cloud Infrastructure, 46
ORACLE object browser, 69
Oracle REST data service, ORDS, 105, 106
Otázky, 345, 400

packaged application, 207
page designer, 89
PL/SQL, 208
Pološtruktúrovaný popis, 228, 254, 300, 312, 328, 390
Problem definition, 267
proces vývoja webových aplikácií, 81
prvá normálna forma, 1NF, 59
Prípady použitia, 228, 238, 253, 294, 300, 312, 327, 390
prípady použitia, 248

query builder, 75
Quick SQL, 63, 69, 119, 207
QuickSQL a Script SQL Script, 331

RDB, 54
RDB schéma, 54, 58
region, 87
Relačný dátový model, 241, 330, 393
relačný dátový model, 250
relačný model, 50, 54
report aplikácie, 103
representational state transfer, REST, 104
resource handler, 108
resource modul, 108
resource template, 108
REST, 104
RESTful prístup, 106
RESTful servis, 104
Rozhrania aplikácie, 233, 242, 251, 259, 276, 297, 305, 322, 334, 355, 372, 395

schéma relačnej DB., 54
Set-up Apex environment, 113
Skill level, 42
správa RDBS schemy, 60
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 GROUP BY, 74
SQL INSERT, 68
SQL JOIN, 73
SQL ORDER BY, 71, 74
SQL Script, 121
SQL SELECT, 71, 102
SQL skript, 394
SQL SUM, 74
SQL TO_CHAR(), 72
SQL UPDATE, 68
SQL WHERE klauzula, 68, 72
SQL workshop, 67
SQL workshop example development, 117
SQL-DDL, 60, 68, 207
SQL-DQL, 70
stránka, 87, 90
stĺpec, 54
Systém správy databáz, 50

tabuľka, 54
Textový popis dátového modelu, 392
tretia normálna forma, 3NF, 60
typ stránky, 87

use case model, UC, 39
Use cases, 267, 348
user feedback, 86

validácie, 87
Virtual Box Appliance, 44
Vytvorenie prístupu pre používateľa, 342
vytvorenie stránky, 86
vzorové a štartovacie aplikácie, 199
vzťah, 51, 54
výmena údajov, 93

web application, 89

webová aplikácia, 49

Workflow model, 270

zoznam, 87

úroveň abstrakcie, 50

štruktúrovaný dopytovací jazyk, SQL, 60