Funded by the European Union

BEE WITH APEX

E-knjiga
Malokodno programiranje sa APEX-om

Praktični vodič za poboljšanje zaposljivosti

Projektna skupina BeeAPEX



Avtorske pravice © 2023 Projektna skupina BeeAPEX

IZDAJATELJ: LABORATORIJ ZA KAKOVOST IN TESTIRANJE PROGRAMSKE OPREME, UM FOV

BEEAPEX.EU

Sadržaj ove knjige je licenciran sa Creative Commons Attribution-NonCommercial 4.0 License (u nastavku "Licenca"). Dozvola za korištenje sadržaja knjige definirana je Licencom. Kopija Licence dostupna je https://creativecommons.org/licenses/by-nc-sa/4.0. Samo u slučaju, da postoji zakonska ili ugovorna obaveza, programska oprema sa tom Licencom distribuirana je po načelu "TAKVA, KAKVA JE", BEZ IKAKVIH JAMSTAVA ILI UVJETA, bilo izraženih implicitno ili eksplicitno. Pročitajte Licencu za određena jezična dopuštenja i ograničenja.

1. izdanje, 7. prosinca 2023.

Sadržaj

Zahvala 27

Predgovor 29

I Kako to učiniti u APEX-u

1 Kako početi s Oracle APEX-om? 35

Vjeran Strahonja, Dijana Oreški

1.1 Što je Oracle APEX i čemu služi? 35
1.1.1 Što je Oracle APEX? 35
1.1.2 Što je aplikacija? 35
1.1.3 Kako Oracle APEX može pomoći u razvoju aplikacija? 35
1.1.4 Ciklus razvoja aplikacije 36
1.2 Kako početi s Oracle APEX-om? 41
1.2.1 Koja je razina Vaših vještina? 41
1.2.2 Korištenje lokalne APEX instance 41
1.2.3 Opcija https://apex.oracle.com 42
1.2.4 Virtualni stroj 42
1.2.5 APEX docker 43
1.2.6 APEX instanca u Oracle Cloud Infrastructure 45
1.2.7 APEX instanca u Oracle Academy 45

2 Kako pripremiti bazu podataka? 47

E. Kapsammer, W. Retschitzegger, W. Schwinger

2.1 Kako modelirati podatke 48
2.1.1 Logički model 48
2.1.2 Relacijski model 52
2.1.3 Normalizacija RDB-sheme 57
2.2 Kako upravljati RDBS-shemom – SQL-DDL 58
2.2.1 Automatsko generiranje tablice pomoću ODM-a 59

2.2.2 Automatsko generiranje tablice pomoću Quick SQL-a . . . 61
2.2.3 Ručno kreiranje tablice pomoću SQL-DDL . . . 63
2.2.4 Ručno upravljanje tablicom pomoću preglednika objekata . . . 65
2.3 Kako manipulirati podacima – SQL-DML . . . 66
2.3.1 Manipulacija podacima pomoću SQL-DML . . . 66
2.3.2 Manipulacija podacima korištenjem ORACLE-ovog preglednika objekata ("OB") 67
2.3.3 Manipulacija podacima pomoću Quick SQL . . . 67
2.4 Kako postaviti upit za podatke – SQL-DQL . . . 68
2.4.1 Upiti podataka pomoću SQL-DQL . . . 68
2.4.2 Kreiranje upita nad podacima pomoću ORACLE-ovog Query Buildera ("QB") 73
2.5 Izgradnja DB-sloja – Velika slika . . . 76
2.6 Pitanja . . . 76
2.7 Odgovori . . . 77

3 Kako koristiti APEX? . . . 79
E. Kapsammer, W. Retschitzegger, W. Schwinger . . .

3.1 Proces razvoja web aplikacije nakon kojeg slijedi APEX . . . 79
3.2 Početni izbornik APEX-a . . . 80
3.3 Pregled App Buildera – Stvorite svoje aplikacije i upravljajte njima . . . 81
3.4 Stvorite aplikaciju – tri slučaja upotrebe . . . 82
3.5 Stvorite aplikaciju – svojstva, stranice, značajke i postavke . . . 83
3.6 Specificirajte stranice . . . 85
3.7 Održavajte i mijenjajte stranicu – Dizajner stranice . . . 87
3.8 Pokrenite aplikaciju ili stranicu . . . 87
3.9 Pitanja . . . 89
3.10 Odgovori . . . 89

4 Kako razmijeniti podatke u APEX-u? . . . 91
E. Kapsammer, W. Retschitzegger, W. Schwinger . . .

4.1 Uvoz i izvoz podataka pomoću "Data Workshopa" . . . 92
4.1.1 Uvoz podataka . . . 92
4.1.2 Koraci uvoza . . . 94
4.1.3 Izvoz podataka . . . 98
4.2 Uvoz i izvoz podataka pomoću "Object Browser-a" . . . 99
4.3 Izvoz rezultata SQL-naredbe . . . 100
4.4 Izvoz podataka iz izvještaja u aplikaciji . . . 101
4.5 Omogućavanje razmjene podataka putem RESTful servisa . . . 102
4.5.1 Arhitektonski obrazac REST . . . 103
4.5.2 Omogućavanje DB sheme za RESTful pristup . . . 104
4.5.3 Moduli resursa | Predlošci | Rukovatelji . . . 105
4.5.4 Korištenje "AutoREST" umjesto ručnog definiranja resursa . . . 107
4.6 Pitanja . . . 109
4.7 Odgovori . . . 109

5 Kako generirati prvi nacrt aplikacije? . . . 111
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis

5.1 Zašto poduzeća trebaju aplikacije? . . . 111
5.2 Postavljanje ORACLE APEX okruženja . . . 111
5.3 Razvoj primjera SQL radionice . . . 115
5.3.1 Quick SQL . . . 119
5.3.2 SQL skripta . . . 120
5.4 Dio aplikacije vođen podacima . . . 122
5.5 Administracija aplikacije . . . 126
5.6 Kontrola pristupa . . . 126
5.7 Dodatni materijali za učenje . . . 132
5.7.1 Izvezene aplikacije . . . 132
5.7.2 Video vodiči . . . 132
5.8 Pitanja . . . 132
5.9 Odgovori . . . 133

6 Kako upravljati izvješćima? . . . 135
Zuzana Žillová, Erik Malina, Matej Grochal, Andrej Staník, Andrea Meleková, Michal Kvet, Miroslav Potočár

6.1 Izvješće . . . 135
6.2 Klasično izvješće . . . 144
6.3 Izvješće o prebacivanju stupaca . . . 146
6.4 Interaktivno izvješće . . . 146
6.5 Pitanja . . . 155
6.6 Odgovori . . . 155

7 Kako upravljati obrascima? . . . 157
Veronika Šalgová, Michal Kvet, Miroslav Potočár

7.1 Kako upravljati obrascima . . . 157
7.2 Interaktivna mreža koja se može uređivati . . . 157
7.3 Obrazac nad tablicom . . . 160
7.4 Master Detail obrazac . . . 160
7.5 Pitanja . . . 163
7.6 Odgovori . . . 163

8 Kako pretvoriti tekstualna izvješća u grafikone? . . . 165
Ivan Pastierik, Michal Kvet, Miroslav Potočár

8.1 Grafikon . . . 165
8.2 Kreiranje stupčastog grafikona . . . 166
8.3 Dodavanje filtriranja na stupčasti grafikon . . . 169
8.4 Dodavanje sortiranja na stupčasti grafikon . . . 174
8.5 Kreiranje različitih tipova grafikona . . . 178

8.6 Pitanja . . . 179
8.7 Odgovori . . . 180

9 Kako upravljati izbornicima? . . . 181
VERONIKA ŠALGOVÁ, MICHAL KVET, MIROSLAV POTOČÁR

9.1 Kako upravljati izbornicima . . . 181
9.2 Izbornik Side Menu . . . 181
9.3 Izbornik Top Menu . . . 182
9.4 Izbornik Mega Menu . . . 182
9.5 Uređivanje lista izbornika . . . 183
9.6 Pitanja . . . 183
9.7 Odgovori . . . 183

10 Kako surađivati u timu? . . . 185
P. STANISZEWSKI, M. SOŃTA, A. KIERZKOWSKI

10.1 Suradnička proizvodnja znanja temelj je razvoja niskog koda . . . 185
10.2 Biti online zajedno . . . 185
10.3 Tehničko znanje kroz suradnju . . . 186
10.4 Opis značajki . . . 187
10.4.1 Zaključavanje stranice . . . 187
10.4.2 Komentari . . . 188
10.4.3 Mogućnosti izgradnje . . . 188
10.4.4 Razvoj u timu . . . 189
10.4.5 Povratne informacije . . . 192
10.4.6 Standardizacija . . . 193
10.5 Zaključak . . . 194
10.6 Pitanja . . . 195
10.7 Odgovori . . . 195

11 Kako iskoristiti galeriju oglednih/početnih aplikacija i dodataka? 197
VJERAN STRAHONJA, DIJANA OREŠKI

11.1 Kako instalirati ogledne i početne aplikacije? . . . 197
11.2 Početne aplikacije . . . 198
11.3 Ogledne aplikacije . . . 199
11.4 Dodaci . . . 201
11.5 Pitanja . . . 204
11.6 Odgovori . . . 204

12 Kako upravljati pakiranim i višejezičnim aplikacijama? . . . 205
R. LESKOVAR, U. RAJKOVIČ, A. BAGGIA

12.1 Aplikacija i pakirana aplikacija . . . 205
12.2 Aplikacija . . . 207
12.2.1 Opseg aplikacije . . . 207

12.2.2 Kreiranje tablica . . . 207
12.2.3 Unos podataka . . . 208
12.2.4 Generiranje aplikacija . . . 208
12.3 Pakirane aplikacije . . . 213
12.4 Višejezične aplikacije . . . 217
12.5 Dodatni materijal za učenje . . . 221
12.5.1 Izvezene aplikacije . . . 221
12.5.2 Video upute . . . 221
12.6 Pitanja . . . 221
12.7 Odgovori . . . 222

II **Razvoj aplikacija u APEX-u**

13 Intranet novosti za zaposlenike . . . 225
R. Leskovar, U. Rajković, A.Baggia

13.1 Poslovni pogled na slučaj . . . 225
13.2 Definicija problema . . . 225
13.3 Slučajevi korištenja . . . 226
13.3.1 Narativni opis slučaja korištenja . . . 226
13.3.2 Polustrukturirani opis . . . 226
13.3.3 Dijagram slučaja korištenja . . . 226
13.4 Podatkovni model . . . 230
13.4.1 Narativni opis podatkovnog modela . . . 230
13.4.2 Logički model podataka . . . 230
13.4.3 Relacijski model podataka . . . 230
13.5 Sučelja aplikacije . . . 231
13.6 Dodatni materijal za učenje . . . 234
13.6.1 Izvezena aplikacija . . . 234
13.6.2 Video upute . . . 234
13.7 Pitanja . . . 234
13.8 Odgovori . . . 234

14 GreenDi - Katalog biljaka . . . 237
Vjeran Strahonja, Dijana Oreški, Darko Andročec, Ana Kutnjak

14.1 Poslovni pogled na slučaj . . . 237
14.2 Definicija problema . . . 237
14.3 Slučajevi korištenja . . . 238
14.3.1 Narativni opis slučaja korištenja . . . 238
14.3.2 Polustrukturirani opis . . . 238
14.3.3 Dijagram slučaja korištenja . . . 238
14.4 Podatkovni model . . . 240
14.4.1 Narativni opis podatkovnog modela . . . 240
14.4.2 Logički model podataka . . . 240
14.4.3 Relacijski model podataka . . . 242

14.5 Sučelja aplikacije . . . 242
14.6 Dodatni materijal za učenje . . . 243
14.6.1 Izvezena aplikacija . . . 243
14.6.2 Video upute . . . 244
14.7 Pitanja . . . 244
14.8 Odgovori . . . 244

15 GreenDi - Autorizacija i upravljanje korisnicima . . . 245
VJERAN STRAHONJA, DARKO ANDROČEC, ANA KUTNJAK, LARISA HRUSTEK . . .

15.1 Poslovni pogled na slučaj . . . 245
15.2 Definicija problema . . . 245
15.3 Slučajevi korištenja . . . 246
15.3.1 Narativni opis slučaja korištenja . . . 246
15.3.2 Polustrukturirani opis . . . 246
15.3.3 Dijagram slučaja korištenja . . . 246
15.4 Podatkovni model . . . 246
15.4.1 Narativni opis podatkovnog modela . . . 246
15.4.2 Logički model podataka . . . 246
15.4.3 Relacijski model podataka . . . 248
15.5 Sučelja aplikacije . . . 248
15.6 Dodatni materijal za učenje . . . 249
15.6.1 Izvezene aplikacije . . . 249
15.6.2 Video upute . . . 249
15.7 Pitanja . . . 249
15.8 Odgovori . . . 250

16 inovativni sustavi . . . 251
R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA . . .

16.1 Poslovni pogled na slučaj . . . 251
16.2 Definicija problema . . . 251
16.3 Slučajevi korištenja . . . 251
16.3.1 Narativni opis slučaja korištenja . . . 251
16.3.2 Polustrukturirani opis . . . 252
16.3.3 Dijagram slučaja korištenja . . . 252
16.4 Podatkovni model . . . 252
16.4.1 Narativni opis podatkovnog modela . . . 252
16.4.2 Logički model podataka . . . 252
16.4.3 Relacijski model podataka . . . 256
16.5 Sučelja aplikacije . . . 257
16.6 Dodatni materijal za učenje . . . 259
16.6.1 Izvezene aplikacije . . . 260
16.6.2 Video upute . . . 260
16.7 Pitanja . . . 260
16.8 Odgovori . . . 260

17 Upravljanje poslovnim procesima . . . 261
R. Leskovar, U. Rajković, A.Baggia

17.1 Poslovni pogled na slučaj . . . 261
17.1.1 Instaliraj Flows for APEX . . . 264
17.1.2 Instalirajte primjer aplikacije za tijek procesa . . . 264
17.1.3 Zadaci za čitanje i vježbanje . . . 264
17.2 Definiranje problema . . . 264
17.3 Slučajevi korištenja . . . 265
17.3.1 Opis riječima . . . 265
17.3.2 Polustrukturirani opis . . . 265
17.3.3 Dijagram slučaja korištenja . . . 265
17.4 Model tijeka rada . . . 268
17.5 Model podataka . . . 271
17.5.1 Opis modela podataka riječima . . . 271
17.5.2 Logički model podataka . . . 271
17.5.3 Relacijski model podataka . . . 271
17.5.4 Quick SQL za generiranje SQL skripte . . . 272
17.5.5 Sekvenca i dvije pohranjene funkcije . . . 273
17.6 Aplikacijska sučelja . . . 274
17.6.1 Popis vrijednosti u dijeljenim komponentama . . . 275
17.6.2 Dodaci u dijeljenim komponentama . . . 276
17.6.3 Voditelj prodaje . . . 276
17.6.4 Voditelj proizvodnje . . . 280
17.6.5 Financijski menadžer . . . 282
17.6.6 Glavni izvršni direktor - voditelj poslovanja . . . 283
17.7 Povezivanje aplikacije s Flows for APEX . . . 285
17.8 Definiranje korisničke uloge . . . 287
17.9 Testiranje i ispravljanje grešaka . . . 287
17.10 Dodatni materijal za učenje . . . 288
17.10.1 Izvezena aplikacija . . . 288
17.10.2 Video vodiči . . . 289
17.11 Pitanja . . . 290
17.12 Odgovori . . . 290

18 GreenDi – Upravljanje katalogom biljaka . . . 291
Vjeran Strahonja, Dijana Oreski, Darko Androcec, Ana Kutnjak

18.1 Poslovni pogled na slučaj . . . 291
18.2 Opis problema . . . 291
18.3 Slučajevi korištenja . . . 292
18.3.1 Opis slučaja korištenja riječima . . . 292
18.3.2 Polustrukturirani opis . . . 292
18.3.3 Dijagram slučajeva korištenja . . . 292
18.4 Model podataka . . . 292
18.4.1 Opis modela podataka riječima . . . 292
18.4.2 Logički model podataka . . . 294
18.4.3 Relacijski model podataka . . . 294

18.5 Aplikacijska sučelja . . . 295
18.6 Dopunski materijal za učenje . . . 296
18.6.1 Izvezene aplikacije . . . 296
18.6.2 Video vodiči . . . 296
18.7 Pitanja . . . 296
18.8 Odgovori . . . 296

19 Sustav za upravljanje recenzijama knjiga . . . 297
A. Kutnjak, L. Hrustek, A. Baggia, R.Leskovar

19.1 Poslovni pogled na slučaj . . . 297
19.2 Definicija problema . . . 297
19.3 Slučajevi korištenja . . . 298
19.3.1 Narativni opis slučaja korištenja . . . 298
19.3.2 Polustrukturirani opis . . . 298
19.4 Podatkovni model . . . 300
19.4.1 Narativni opis podatkovnog modela . . . 300
19.4.2 Logički model podataka . . . 300
19.4.3 Relacijski model podataka . . . 300
19.4.4 SQL script . . . 301
19.4.5 Quick SQL . . . 302
19.5 Sučelja aplikacije . . . 303
19.5.1 Administrator . . . 303
19.5.2 Korisnik . . . 305
19.6 Definiranje korisničkih uloga . . . 307
19.7 Dodatni materijal za učenje . . . 307
19.7.1 Izvezena aplikacija . . . 307
19.7.2 Video upute . . . 308
19.8 Pitanja . . . 308
19.9 Odgovori . . . 308

20 Sastavnica i obračun troškova . . . 309
R. Leskovar, U. Rajković, A.Baggia

20.1 Poslovni pogled na slučaj . . . 309
20.2 Definicija problema . . . 309
20.3 Slučajevi korištenja . . . 310
20.3.1 Narativni opis . . . 310
20.3.2 Polustrukturirani opis . . . 310
20.3.3 Dijagram slučaja korištenja . . . 313
20.4 Podatkovni model . . . 313
20.4.1 Narativni opis podatkovnog modela . . . 313
20.4.2 Implementacija poslovnih pravila u bazi podataka . . . 314
20.4.3 Logički model podataka . . . 314
20.4.4 Relacijski model podataka . . . 314
20.4.5 Objekti u APEX-u . . . 316

20.5 Sučelja aplikacije . . . 319
20.6 Dodatni materijal za učenje . . . 322
20.6.1 Izvezene aplikacije . . . 322
20.6.2 Video upute . . . 322
20.7 Pitanja . . . 322
20.8 Odgovori . . . 322

21 Prehrana i upravljanje dijetom . . . 325
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . . .

21.1 Poslovni pogled na slučaj . . . 325
21.2 Definicija problema . . . 325
21.3 Slučajevi korištenja . . . 325
21.3.1 Narativni opis . . . 325
21.3.2 Polustrukturirani opis . . . 326
21.3.3 Dijagram slučaja korištenja . . . 326
21.4 Podatkovni model . . . 326
21.4.1 Logički model podataka . . . 326
21.4.2 Relacijski model podataka . . . 328
21.4.3 QuickSQL i SQL skripta . . . 329
21.5 Sučelja aplikacije . . . 332
21.5.1 Kako izraditi izvještaj . . . 336
21.5.2 Stvoranje pristupa korisniku . . . 340
21.6 Dodatni materijal za učenje . . . 343
21.6.1 Izvezena aplikacija . . . 343
21.6.2 Video upute . . . 343
21.7 Pitanja . . . 343
21.8 Odgovori . . . 344

22 Raspored konzultacija . . . 345
J. Mańko, M. Sońta, R. Leskovar . . .

22.1 Poslovni pogled na slučaj . . . 345
22.2 Definicija problema . . . 346
22.3 Slučajevi korištenja . . . 346
22.3.1 Narativni opis . . . 346
22.3.2 Polustrukturirani opis . . . 347
22.3.3 Dijagram slučajeva korištenja . . . 347
22.4 Model podataka . . . 349
22.4.1 Opis modela podataka riječima . . . 349
22.4.2 Logički model podataka . . . 349
22.4.3 Relacijski model podataka . . . 350
22.4.4 Quick SQL . . . 351
22.4.5 SQL skripta za kreiranje tablica . . . 351
22.4.6 Query builder u APEX-u . . . 351

22.5 Aplikacijska sučelja . . . 352
22.5.1 Sučelja aplikacija za menadžment . . . 353
22.5.2 Aplikacijska sučelja studentske referade . . . 353
22.5.3 Aplikacijska sučelja namijenjena studentima . . . 355
22.5.4 Aplikacijska sučelja namijenjena nastavnicima . . . 357
22.6 Dodatni materijali za učenje . . . 360
22.6.1 Izvezena aplikacija . . . 360
22.6.2 Video vodiči . . . 362
22.7 Pitanja . . . 362
22.8 Odgovori . . . 363

23 Telco studija slučaja . . . 365
VERONIKA ŠALGOVÁ, JOZEF KOSTOLNÝ, MICHAL MRENA, MICHAL KVET, MIROSLAV POTOČÁR . . .
23.1 Poslovni pogled na slučaj . . . 365
23.2 Definicija problema . . . 365
23.3 Slučajevi korištenja . . . 365
23.3.1 Narativni opis slučaja korištenja . . . 366
23.3.2 Polustrukturirani opis . . . 366
23.3.3 Dijagram slučaja korištenja . . . 366
23.4 Podatkovni model . . . 368
23.4.1 Narativni opis podatkovnog modela . . . 368
23.4.2 Logički model podataka . . . 368
23.4.3 Relacijski model podataka . . . 368
23.5 Autentifikacija korisnika i korisničke uloge . . . 369
23.6 Sučelja aplikacije . . . 371
23.7 Dizajn aplikacije . . . 372
23.8 Skripte . . . 380
23.9 Izrada početne stranice . . . 382
23.10 Izrada korisničke stranice . . . 383
23.11 Kreiranje stranice menadžera . . . 386
23.12 Dodatni materijal za učenje . . . 387
23.12.1 Izvezena aplikacija . . . 387
23.12.2 Video upute . . . 387
23.13 Pitanja . . . 387
23.14 Odgovori . . . 388

24 Studija slučaja iznajmljivanja automobila . . . 389
A. ANGEIOPLASTIS, G. MYLLIS, A. TSIMPIRIS, D. VARSAMIS . . .
24.1 Poslovni pogled na slučaj . . . 389
24.2 Definicija problema . . . 389
24.3 Slučajevi korištenja . . . 390
24.3.1 Narativni opis slučaja korištenja . . . 390
24.3.2 Polustrukturirani opis . . . 390
24.3.3 Dijagram slučaja korištenja . . . 390

24.4 Podatkovni model 392
24.4.1 Narativni opis podatkovnog modela 392
24.4.2 Logički model podataka 393
24.4.3 Relacijski model podataka 393
24.4.4 SQL skripta 394
24.5 Sučelja aplikacije 395
24.6 Dodatni materijal za učenje 399
24.6.1 Izvezena aplikacija 400
24.6.2 Video upute 400
24.7 Pitanja 400
24.8 Odgovori 400

Literatura 401
Članci 401
Knjige 401

Kazalo 403



Popis slika

1.1 Ciklus razvoja aplikacije . . . 36
1.2 Platno poslovnog modela (6) . . . 37
1.3 Osnovni koncepti dijagrama slučajeva korištenja . . . 38
1.4 Generalizacija/specijalizacija Sudionika . . . 39
1.5 Granice sustava i veze "include" i "extend" . . . 39
1.6 Dijagram slučajeva korištenja sustava rezervacije hotela . . . 40
1.7 Searching for the APEX docker. . . . 43

2.1 DBS kao okosnica web aplikacija . . . 47
2.2 Logički model našeg primjera HR-a pomoću ERA dijagrama . . . 48
2.3 Upravljanje logičkim modelom u ODM-u . . . 50
2.4 Upravljanje atributima u ODM-u . . . 51
2.5 Upravljanje vezama u ODM-u . . . 52
2.6 RDB-sheme i sheme tablica . . . 53
2.7 Proces za generiranje RDB-sheme iz logičkog modela . . . 54
2.8 Automatski generirana RDB shema . . . 54
2.9 GUI-elementi za upravljanje stupcima RM-a . . . 56
2.10 GUI-elementi za upravljanje RM-om . . . 56
2.11 Koraci normalizacije – pregled . . . 57
2.12 Kategorije SQL naredbi . . . 59
2.13 Izvoz RDB-sheme u DDL-skriptu . . . 60
2.14 Prijenos i izvođenje SQL-skripte . . . 61
2.15 Pristup Quick SQL-u putem SQL Workshopa . . . 62
2.16 Generirana SQL skripta na temelju Quick SQL-a . . . 63
2.17 SQL Command Editor . . . 64
2.18 SQL naredba za kreiranje tablice `Departments` . . . 64
2.19 SQL naredba za brisanje tablice `Departments` . . . 65
2.20 SQL naredba za promjenu tablice `Departments` . . . 65
2.21 Upravljanje tablicama pomoću ORACLE-ovog Object Browsera . . . 65
2.22 SQL naredba za umetanje novih podataka u tablicu `Departments` . . . 66
2.23 SQL naredba za ažuriranje plaće svih zaposlenika za 10% u tablici `Employees` . . . 66
2.24 SQL naredba za ažuriranje plaće samo zaposlenika "Miller" . . . 66
2.25 SQL izjava za ažuriranje više od jednog atributa u tablici `Employees`. . . . 67

2.26 SQL naredba za brisanje svih podataka iz tablice Employees 67
2.27 SQL naredba za brisanje nekih podataka iz tablice Zaposlenici 67
2.28 Nasumično umetanje podataka pomoću Quick SQL-a 68
2.29 SQL izjava za dohvaćanje svih podataka iz tablice Employees. 68
2.30 Izvadak iz Dohvaćanja svih podataka iz tablice EMPLOYEES. 69
2.31 SQL naredba za dohvaćanje svih podataka iz tablice eMPLOYEES poredanih prema njihovoj plaći. 69
2.32 Izvadak iz dohvaćanja svih podataka iz tablice Employees poredanih prema njihovoj plaći. 69
2.33 SQL upit samo za dohvaćanje određenih podataka iz tablice Employees. 70
2.34 Izvadak iz dohvaćanje određenih podataka samo iz tablice Employees. 70
2.35 SQL upit za dohvaćanje određenih podataka iz tablice Employees samo pomoću TO_CHAR()-funkcije. 70
2.36 Izvadak iz dohvaćanja određenih podataka iz tablice Employees samo pomoću TO_CHAR()-funkcije. 71
2.37 SQL naredba za spajanje podataka iz tablice Employees i tablice Departments. 71
2.38 Izvadak iz dohvaćanja određenih podataka iz tablice Employees i tablice Departments. 72
2.39 SQL naredba za prebrojavanje broja zaposlenika i izračunavanje zbroja njihovih plaća iz tablice Employees. 72
2.40 Izvadak iz dohvaćanja broja zaposlenih i građenja zbroja njihovih plaća iz tablice Employees. 72
2.41 SQL naredba za grupiranje podataka prema njihovoj plaći i sređivanje rezultata prema broju zaposlenika s tom plaćom iz tablice Employees. 73
2.42 Izvadak iz grupiranja podataka prema njihovoj plaći i sređivanja rezultata prema broju zaposlenika s tom plaćom iz tablice Employees 73
2.43 Pregled ORACLE Query Builder-a ("QB") 74
2.44 Korištenje Query Buildera za izradu spoja (join) 75
2.45 Pregled opcija za upravljanje DB-slojem 76

3.1 Sveukupni razvojni proces u APEX-u 80
3.2 Početna stranica Oracle APEX Workspace 81
3.3 Pregled App Buildera 82
3.4 Čarobnjak za izradu aplikacije: Tri slučaja upotrebe pri izradi aplikacije 83
3.5 Svojstva i stranice 84
3.6 Značajke i postavke 84
3.7 Početna stranica aplikacije - pogled razvojnog programera 85
3.8 Izrada stranice 86
3.9 Dizajner stranica 87
3.10 Pokretanje cijele aplikacije 88
3.11 Pokretanje pojedinačnih stranica 89

4.1 Mogućnosti razmjene podataka u APEX-u 92
4.2 Korištenje Data Workshopa za učitavanje podataka - pristup putem SQL Workshopa 93
4.3 Korištenje Data Workshopa za učitavanje podataka - pristup putem App Buildera 94
4.4 Pružanje izvora podataka 95
4.5 Učitavanje podataka u novu tablicu 95
4.6 Učitavanje podataka u postojeću tablicu. 97
4.7 Izvoz ("istovar") podataka 98
4.8 Čarobnjak za uklanjanje podataka 99
4.9 Uvoz/izvoz tabličnih podataka pomoću preglednika objekata 100

4.10 Izvoz podataka kao rezultat SQL-upita 101
4.11 Izvoz podataka iz izvještaja u aplikaciji 102
4.12 RESTful servisi za razmjenu podataka – osnovna arhitektura 103
4.13 Omogućavanje DB sheme za RESTful pristup 105
4.14 Odnosi između različitih komponenti ORDS RESTful usluga 106
4.15 Primjer RESTful usluga za tablicu Employee 107
4.16 Omogućavanje usluge REST u pregledniku objekata 108

5.1 Upišite APEX adresu i prijavite se 112
5.2 Provedite početne korake za dobivanje Oracle APEX Workspace 112
5.3 Zatražite dovršetak Oracle APEX Workspacea 113
5.4 Oracle APEX Workspace odobren 113
5.5 Aktivacijska e-pošta od APEX-a 114
5.6 Postavite novu lozinku za Workspace APEX 114
5.7 Radni prostor APEX okruženje 115
5.8 Logički model našeg primjera vođenja HR-a 116
5.9 Relacijski model našeg primjera Running HR 118
5.10 Umetnite Suick SQL kod u APEX radni prostor 120
5.11 Pokreni SQL skriptu 122
5.12 Stvorite postupak pokretanja aplikacije 122
5.13 Stvorite aplikaciju iz skripte 123
5.14 Izrađena web aplikacija 124
5.15 Okruženje za izradu aplikacija 124
5.16 Stranica za prijavu u aplikaciju 125
5.17 Vaše novo okruženje aplikacije 125
5.18 Upravljajte okruženjem korisnika i grupa 126
5.19 Primjer kontrole pristupa 127
5.20 Dodavanje novih korisnika 128
5.21 Dodajte uloge korisnicima 129
5.22 Stvorite konfiguraciju gumba 130
5.23 Pristup i prava suradnika 131
5.24 Pristup i prava čitatelja 132

6.1 Odabir vrste stranice 136
6.2 Izrada stranice s izvješćem 137
6.3 Renderiranje 137
6.4 Svojstva regije 138
6.5 Razni izvori podataka 138
6.6 Izgled 139
6.7 Izgled 139
6.8 Predlošci 140
6.9 Napredno 140
6.10 Zaglavlje i podnožje 140
6.11 Stanje na strani poslužitelja 141
6.12 Samo za čitanje 141
6.13 Sigurnost, predmemorija poslužitelja i prilagodba 141
6.14 Paginacija 142
6.15 Broj redaka za učitavanje 142
6.16 Preuzimanje i ispis 142

6.17 Središte dizajnera stranica . . . 143
6.18 Vrste stupaca . . . 143
6.19 Postavke stupaca . . . 144
6.20 Maska formata. . . . 144
6.21 Prilagodba . . . 144
6.22 Mogućnost sortiranja po stupcima . . . 145
6.23 Lokacija zelene strelice za prikaz stranice . . . 145
6.24 Primjer klasičnog izvješća . . . 146
6.25 Mogućnost promjene vidljivosti stupca u izvješću o preklapanju stupaca . . . 146
6.26 Interaktivno izvješće . . . 147
6.27 Panel za pretraživanje . . . 147
6.28 Pretraga . . . 148
6.29 Gumb za radnje . . . 148
6.30 Odabir stupaca za prikaz . . . 149
6.31 Filtar stupaca . . . 149
6.32 Filtar redaka . . . 150
6.33 Nakon klika na gumb Data . . . 150
6.34 Sort . . . 151
6.35 Aggregate . . . 151
6.36 Flashback . . . 151
6.37 Control Break . . . 152
6.38 Rezultat prekida kontrole . . . 152
6.39 Highlight . . . 152
6.40 Istakni rezultat . . . 153
6.41 Redak po stranici . . . 153
6.42 Grafikon u interaktivnom izvješću . . . 153
6.43 Group by . . . 154
6.44 Save report . . . 154
6.45 Preuzimanje . . . 155

7.1 Interaktivna mreža s kupcima . . . 157
7.2 Akcije interaktivne mreže . . . 158
7.3 Hide ikona na stupcu Phone . . . 158
7.4 Prikazani stupci . . . 158
7.5 Freeze ikona stupca Address . . . 159
7.6 Sortiranje stupca Address . . . 159
7.7 Aggregate ikona na stupcu Address . . . 159
7.8 Vrste grafikona . . . 160
7.9 Obrazac za umetanje kupaca . . . 160
7.10 Stacked master detail obrazac za Customer Flat Rates . . . 161
7.11 Side by Side master detail obrazac Customer Flat Rates . . . 161
7.12 Drill Down master detail obrazac – prva stranica . . . 162
7.13 Drill Down master detail obrazac – druga stranica . . . 162

8.1 Podatkovni model korišten u aplikaciji . . . 165
8.2 Selektiranje Chart region . . . 166
8.3 Sekcija Layout nakon kreiranja Chart regije. . . . 166
8.4 Pregled stabla prikazuje grešku. . . . 167
8.5 Finalna konfiguracija serije . . . 167

8.6 Podaci vraćeni izvršavanjem SQL upita 168
8.7 Grafikon prikazan prilikom pokretanja aplikacije 168
8.8 Promjena atributa naslova za x os 169
8.9 Grafikon prodaje nakon označavanja osi 169
8.10 Odabir Checkbox Group stavke 170
8.11 Postavljanje temeljnih atributa na Checkbox Group stavku 170
8.12 Postavljanje liste vrijednost na Checkbox Group stavku 171
8.13 Postavljanje inicijalnih vrijednosti Checkbox Group stavke 171
8.14 Povezivanje Sales Chart regije s Checkbox Group stavkom. 172
8.15 Kreiranje Dynamic Action za P1_PRODUCTS_CHOICE stavku 172
8.16 Postavljanje osnovnih atributa promjene dinamičke radnje 173
8.17 Postavljanje atributa na Refresh action 173
8.18 Dynamic Action za osvježavanje Sales Chart u pregledu stabla. 174
8.19 Aplikacija nakon filtriranja proizvoda 174
8.20 Odabir Select List stavke 175
8.21 Sales Chart regija nakon preimenovanja Select List stavke 175
8.22 Postavljanje liste vrijednosti za Select List stavke 175
8.23 Postavljanje statičkih vrijednosti Select List stavke 176
8.24 Dodavanje Dynamic Action na Select List stavci 176
8.25 Postavljanje Order By Item za Sales Chart regiju. 176
8.26 Postavljanje Order By klauzule za Select List stavke korišten za sortiranje 177
8.27 Aplikacija nakon implementacije sortiranja grafikona 177
8.28 Mijenjanje tipa grafa na Line with Area. 178
8.29 List of Values za Select List stavke korišten za filtriranje prodaje proizvoda po godini 178
8.30 Postavljanje izvornog SQL upita i mapiranja stupaca za regiju grafikona mjesečne prodaje 179
8.31 Monthly Sales grafikon 179

9.1 Prošireni Side Navigation Menu 181
9.2 Top Navigation kartice 182
9.3 Top Navigation Menu 182
9.4 Top Navigation Mega Menu 182
9.5 Navigation menu atributi 183

10.1 Page Blocking značajka u APEX-u 187
10.2 Zaključvanje specifične stranice 187
10.3 Dodavanje komentara 188
10.4 Opcije izgradnje 189
10.5 Team Development značajka 190
10.6 Aktivacija Team Development značajke 190
10.7 Prilagodba Team Development značajke 191
10.8 Komunikacija vezana uz problem unutar Team Development značajke 191
10.9 Dopuštanje povratnih informacija u definiciji aplikacije 192
10.10 Predaja povratne informacije u aplikaciji 193
10.11 Utilities izbornik 194

11.1 Oracle APEX radni prostori dolaze s početnim i ogledним aplikacijama 197
11.2 Početne apliacije u galeriji 198
11.3 Instalacija početne aplikacije Team Calender 199
11.4 Ogledne aplikacije 200

12.1 Čarobnjak za uzvoz i izvoz aplikacije 206
12.2 Pretvorba Quick SQL u SQL naredbe 208
12.3 Skripta CH12INSERT unesi podatke u tri tablice i napravi transakcije 208
12.4 Kreiranje aplikacije iz skripte 209
12.5 Definiranje naziva aplikacije i svih značajki 209
12.6 Odabir Shared components 210
12.7 Kreiranje liste vrijednosti od početka. 210
12.8 Definiranje naziva i tipa: CH12_LOV_COMPETENCE_DESCRIPTION je dinamički 211
12.9 Unos SQL SELECT naredbe 211
12.10 Definiranje naziva i tipa: CH12_LOV_COMPETENCE_LEVEL je statično 212
12.11 Definiranje vrijednosti za prikaz i vraćanje: CH12_LOV_COMPETENCE_LEVEL je statično. 212
12.12 Odaberite stranicu 7 (Ch12 Job Competences) 213
12.13 Odaberite stranicu 7 (Ch12 Job Competences) 213
12.14 Odaberite Supporting Objects 214
12.15 Postavljanje preduvjeta 214
12.16 Postavite provjeru postojanja tri tablice 214
12.17 Postavite upit za preimenovanje aplikacije 215
12.18 Postavite instalacijske skripte 215
12.19 Postavite deinstalacijske skripte 215
12.20 Uvezite aplikaciju u drugi radni prostor 216
12.21 Ugovor o licenci 216
12.22 Preimenujte uvezenu aplikaciju 217
12.23 Dodavanje uloge korisniku 217
12.24 Definirani jezici za prijevod 218
12.25 Izvorni tekst koji se može prevesti 218
12.26 Izvoz nizova za određeni jezik i stranicu 219
12.27 Prijevod "target" označeni nizovi u recima 48, 52, 56, 60, 72 i 76 219
12.28 Prijenos XLDIFF datoteka prijevoda 219
12.29 Primjena promjena i objavljivanje. 220
12.30 Konačna objava prijevoda aplikacije 220
12.31 Postavljanje primarnog jezika aplikacije 220
12.32 Prevedena stranica 220

13.1 Dijagram slučaja korištenja 226
13.2 Logički model podataka 230
13.3 Logički model podataka 231
13.4 Sučelje Izdavača 231
13.5 Objavljivanje novosti s prilozima 232
13.6 Sučelje Čitatelja 232
13.7 Čitatelj pristupa novostima 233
13.8 Sučelje Administratora 233
13.9 Uređivanje uloge zaposlenika od strane Administratora 234

14.1 Dijagram slučaja korištenja - Katalog biljaka 240
14.2 Logički model podataka 241
14.3 Relacijski model podataka 242
14.4 Interaktivna mreža (Interactive Grid) obrasca za unos podataka o biljkama (Plant) 243
14.5 Javni dio – otvoreno pregledavanje podataka o biljkama 243

15.1 Dijagram slučaja korištenja - GreenDi Autorizacija i upravljanje korisnicima . . . 246
15.2 Logički model podataka . . . 248
15.3 Relacijski model podataka . . . 248
15.4 User Form . . . 249
15.5 User History obrazac . . . 249

16.1 Dijagram slučaja korištenja . . . 252
16.2 Logički model podataka . . . 256
16.3 Relacijski model podataka . . . 256
16.4 Nadzorna ploča za zaposlenike . . . 257
16.5 Slanje ideje s prilozima . . . 257
16.6 Organizacijska struktura u poduzeću . . . 258
16.7 Small Innovation Idea pregled . . . 258
16.8 Nadzorna ploča recenzenta . . . 259
16.9 Stranica za pregled ideja . . . 259

17.1 Integracija Flows for APEX s APEX aplikacijom . . . 263
17.2 Dijagram slučaja korištenja . . . 268
17.3 Tijek rada - obrada upita - BPMN dijagram. . . . 268
17.4 Kreiranje instanci . . . 269
17.5 Kreiranje instance toka CH17 . . . 269
17.6 Pokretanje demo instance . . . 269
17.7 Dovršite prvi zadatak u demo instanci . . . 270
17.8 Izvršavanje zadataka u paralelnim vratima . . . 270
17.9 Dovršena demo instanca . . . 270
17.10 Logički model podataka . . . 271
17.11 Relacijski model podataka . . . 272
17.12 Početna stranica aplikacije "CH17 Business Process Management - Upravljanje poslovnim procesom".. . . . 275
17.13 Prodaja - započeti proces . . . 276
17.14 Izvješće o tijeku prodaje . . . 277
17.15 Izvješće o tijeku prodaje za odabranu instancu . . . 277
17.16 Izvješće o tijeku prodaje za odabranu instancu. . . . 277
17.17 Prodaja -izvješće o upitima. . . . 278
17.18 Prodaja -popis dokumenata . . . 278
17.19 Prodaja - učitavanje dokumenta na upit . . . 279
17.20 Prodaja - popis dokumenata nakon učitavanja i osvježavanja . . . 279
17.21 Izvješće o tijeku prodaje za odabranu instancu nakon učitavanja tri dokumenta 280
17.22 Izvješće o tijeku prodaje za odabranu instancu nakon učitavanja tri dokumenta 280
17.23 Izvješće o tijeku proizvodnje. . . . 281
17.24 Proizvodnja - BPMN dijagram - stanje instance . . . 281
17.25 Ocjenjivanje proizvodnje . . . 281
17.26 Procjena proizvodnje je završena, primjerak čeka na financijsku procjenu . . . 282
17.27 Financije - izvješće o tijeku . . . 282
17.28 Financije - BPMN dijagram - stanje instance . . . 282
17.29 Financijska procjena . . . 283
17.30 Financijska procjena je završena, instanca čeka na procjenu poslovanja. . . . 283
17.31 Financije - BPMN dijagram - stanje instance. . . . 284
17.32 Izvještaj o tijeku poslovanja . . . 284
17.33 Procjena poslovanja . . . 284

17.34 Poslovna procjena završena, instanca prekinuta 285
17.35 Prikaz stanja instance na BPMN dijagramu 286
17.36 Postavljanje stavki stranice 286
17.37 Dovršavanje koraka u Flows for Apex 287
17.38 Definirajte uloge i korisničke uloge u izborniku Kontrola pristupa aplikaciji . . . 287
17.39 Dodjela "povlasti za stvaranje posla" radnom prostoru. 288
17.40 Uvoz tijeka rada pod nazivom CH17 u Flows for APEX - korak 1 289
17.41 Uvoz tijeka rada pod nazivom CH17 u Flows for APEX - korak 2 289

18.1 Dijagram slučaja korištenja - Razmjena biljaka i sjemena 292
18.2 Logički model podataka 294
18.3 Relacijski model podataka 295
18.4 Ponuda 295
18.5 Poruka 295

19.1 Dijagram slučaja korištenja: administrator dodaje novi komentar 298
19.2 Logički model podataka 300
19.3 Relacijski model podataka 301
19.4 Generiranje aplikacije iz skripte CH19CREATEINSERT - 1.dio 303
19.5 Generiranje aplikacije iz skripte CH19CREATEINSERT - 2. dio. 304
19.6 Unos knjige od strane administratora 305
19.7 Unos kategorije od strane administratora 305
19.8 Registracija korisnika 306
19.9 Pregledavanje i dodavanje recenzija 306
19.10 obrazac za komentiranje recenzije 307

20.1 Dijagram slučaja korištenja 313
20.2 Logički model podataka 314
20.3 Logički model podataka 315
20.4 Logički model podataka. 315
20.5 Logički model podataka. 317
20.6 Početna stranica aplikacije 320
20.7 Upravljanje osnovnim podacima - izvješće 320
20.8 Upravljanje osnovnim podacima - obrazac 320
20.9 Upravljanje strukturnim podacima - izvješće 321
20.10 Upravljanje strukturnim podacima - obrazac 321
20.11 Stranica za izračun sastavnice 322

21.1 Dijagram slučaja korištenja 326
21.2 Logički dijagram 328
21.3 Relacijski dijagram 329
21.4 Brzi SQL postupak za izradu tablica. 330
21.5 Procedura SQL skripte za izgradnju SEQUENCE 332
21.6 Stvaranje nove aplikacije 333
21.7 Aplikacija za prehranu 334
21.8 Aplikacija za prehranu - stranica s receptima 335
21.9 Aplikacija za prehranu - stranica sa sastojcima 335
21.10 Aplikacija za prehranu - stranica sa sastojcima recepta 336
21.11 Aplikacija za prehranu - stranica s podacima o prehrani 336
21.12 Stvorite početnu stranicu nadzorne ploče za svoju aplikaciju 337

21.13 Konfigurirajte naslov i vrstu razgovora na izvješćima nadzorne ploče 338
21.14 Konfigurirajte SOURCE sa SQL QUERY 340
21.15 Kreiranje korisničkog računa 341
21.16 Kreiranje korisničkog računa 342
21.17 Konfiguriranje stranice s korisničkim pravima 343

22.1 Dijagram slučajeva korištenja 349
22.2 Logički model podataka 350
22.3 Relacijski model podataka 350
22.4 Model podataka opisan pomoću Quick SQL-a 351
22.5 Primjer korištenja Query Builder-a. 352
22.6 Početna stranica aplikacije 353
22.7 Upravljačka ploča za menadžment 353
22.8 Interaktivno izvješće o konzultacijama za studentsku referadu 354
22.9 Obrazac za promjenu rasporeda namijenjen studentskoj referadi 354
22.10 Interaktivno izvješće o konzultacijama nakon promjene rasporeda 355
22.11 kalendar studentskih termina 355
22.12 Kalendar svih uredovnih sati nastavnika namijenjenih konzultacijama, na studijskom programu studenta 356
22.13 Upis termina konzultacija putem kalendara - odabir vremensko odsječka u kalendaru 356
22.14 Upis termina konzultacija putem kalendara - odabir svrhe 356
22.15 Upis termina konzultacija putem kalendara - uspješan upis 357
22.16 Provjera upisa kroz točku izbornika "Pregledaj moje termine" 357
22.17 Detaljno interaktivno izvješće o svim studentskim terminima 357
22.18 Detaljno izvješće o zakazanim terminima nastavnika 358
22.19 Kalendar nastavnika s vidljivim imenima studenata, jezikom kojim govore i svrhom konzultacija 358
22.20 Promjena rasporeda termina konzultacija bez upisanih studenata - prikaz kalendara 359
22.21 Promjena rasporeda termina konzultacija bez upisanih studenata - unesen je novi datum 359
22.22 Rezultat promjene rasporeda termina konzultacija bez upisanih studenata . 360

23.1 Dijagram slučaja korištenja 367
23.2 Logički model podataka 368
23.3 Relacijski model podataka 369
23.4 Nadzorna ploča kupca 373
23.5 Nadzorna ploča kupca – modalni prozor računa 374
23.6 Nadzorna ploča kupca – Račun u PDF-u 374
23.7 Nadzorna ploča kupca – Statistika minuta 374
23.8 Nadzorna ploča kupca – Statistika SMSova 375
23.9 Nadzorna ploča kupca – Statistika podataka 375
23.10 Nadzorna ploča menadžera – Dodavanje kupca 376
23.11 Nadzorna ploča menadžera – Upravljanje kupcem 377
23.12 Nadzorna ploča menadžera – Pregled kupca 377
23.13 Nadzorna ploča menadžera – Izvoz kupca 378
23.14 Nadzorna ploča menadžera – Izvoz kupca u XLS datoteku 378
23.15 Nadzorna ploča administratora 379
23.16 Odredišna stranica s prijavom 379

23.17 Stranica za prijavu 379
23.18 Regije glavnog sadržaja 382
23.19 Lista sadržaja 382
23.20 HTML kod statičke regije 383

24.1 Dijagram slučaja korištenja 392
24.2 Logički model podataka studije slučaja iznajmljivanja automobila. 393
24.3 Relacijski model podataka studije slučaja iznajmljivanja automobila 394
24.4 Prijava u aplikaciju 396
24.5 Početna stranica aplikacije 396
24.6 Obrazac prikaza automobila u kreiranoj aplikaciji 397
24.7 Kreiranje stupca s fotografijama 398
24.8 Obrazac prikaza kupaca u kreiranoj aplikaciji 399
24.9 Obrazac za iznajmljivanje automobila u kreiranoj aplikaciji 399



Popis tablica

13.1 Opis slučaja korištenja: Objavljivanje intranet novosti 227
13.2 Opis slučaja korištenja: Čitanje intranet novosti 228
13.3 Opis slučaja korištenja: Upravljanje intranet portalom. 229

14.1 Opis slučaja korištenja: pregled kataloga biljaka 239

15.1 Opis slučaja korištenja: Autorizacija i upravljanje korisnicima 247

16.1 Opis slučaja korištenja: Podnošenje ideje 253
16.2 Opis slučaja korištenja: Pregled ideja. 254
16.3 Opis slučaja korištenja: Evaluacija ideja 255

17.1 Opis slučaja korištnja: priprema dokumentacije za upit 266
17.2 Opis slučaja korištenja: procjena upita s aspekta proizvodnje 266
17.3 Opis slučaja korištenja: procjena financijskih aspekata upita. 267
17.4 Opis slučaja korištenja: procjena poslovnih aspekata upita 267

18.1 Opis slučaja korištenja: Razmjena biljaka i sjemena. 293

19.1 Opis slučaja korištenja: pregled recenzija knjiga 299

20.1 Opis slučaja korištenja: Prijava i održavanje osnovnih podataka 310
20.2 Opis slučaja korištenja: Izvješćivanje i održavanje strukturnih podataka 311
20.3 Opis slučaja korištenja: Izračun sastavnice 312

21.1 Opis slučaja korištenja: Pristup receptima i sastojcima te izračun nutritivnih podataka 327

22.1 Opis slučaja korištenja: preraspoređivanje termina konzultacija od strane nastavnika 347
22.2 Opis slučaja korištenja: upis studenta u termin konzultacija. 348

23.1 Opis slučaja korištenja: dodaj uslugu 366
23.2 Opis slučaja korištenja: prikaži status usluge. 367
23.3 Podaci spremljeni u tablici CH23_Person 370

24.1 Opis slučaja korištenja: pristup automobilima, kupcima i rezervacijama vezanim uz najam automobila 391

Zahvala

Projektni partneri i BEE WITH APEX tim:

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

zahvaljuje za financijsku podršku dobivenu European Commission kroz Action Erasmus + Better Employability for Everyone with APEX (projekt ID 2021-1-SI01-KA220-HED-000032218), sufinanciran Erasmus+ programom Europske Unije.

Podrška Europske komisije za izradu ove publikacije ne znači odobravanje sadržaja koji odražava samo stajališta autora i Komisija se ne može smatrati odgovornom za bilo kakvu upotrebu informacija sadržanih u njoj.

Veliko hvala:

- Oracle Academy *Program menadžeru Mr. Darko Jureković* za kontinuiranu podršku u širenju rezultata projekta i
- THE RIGHT THING SOLUTIONS *CEO Mr. Aljaž Mali* za njegove dragocjene savjete o APEX-u prije i tijekom projekta.

BeeAPEX projektni tim



Predgovor

Dobrodošli u istraživanje Oracle Application Express (APEX) – intuitivne i snažne razvojne platforme s malo koda za stvaranje web aplikacija vođenih podacima. Ovaj je udžbenik osmišljen kako bi vas naučio kompetencijama potrebnim za iskorištavanje punog potencijala Oracle APEX-a i izradu vrhunskih aplikacija za rješavanje poslovnih izazova u stvarnom svijetu.

Prvi dio ovog udžbenika je“Kako’, i posvećen je temeljnim aspektima Oracle APEX-a. U ovih dvanaest poglavlja, krenut ćete na putovanje s ciljem razumijevanja temeljnih koncepata, dobili pristup APEX-u i istražili različite funkcionalnosti za izgradnju robusnih aplikacija. Svako poglavlje obrađuje određenu temu, pruža jasne upute i olakšava praktično učenje.

Poglavlje 1: “Kako početi s Oracle APEX-om?” objašnjava što je, čemu služi i opisuje različite načine pripreme APEX okruženja za praktično iskustvo učenja i razvoj aplikacija.

Poglavlje 2: “Kako pripremiti bazu podataka?” pruža uvod u modeliranje podataka, upravljanje bazom podataka, manipuliranje podacima i postavljanje upita za podatke. Za početnike je neophodna ovakva šira slika koncepata baze podataka.

Poglavlje 3: “Kako koristiti APEX?” prolazi kroz APEX funkcionalnosti koje omogućuju razvoj aplikacija, generiranje i prilagodbu različitih web stranica i njihovih komponenti. Pokretanje i testiranje APEX aplikacije samo je jedna kartica udaljena od razvojnog okruženja.

Poglavlje 4: “Kako razmijeniti podatke u APEX-u?” daje uvid u uvoz i izvoz podataka unutar APEX-a. Poglavlje pokriva razmjenu podataka s datotekama kao što su proračunske tablice i također putem RESTful usluga.

Poglavlje 5: “Kako generirati prvi nacrt aplikacije?” vas poziva da isprobate razvojnu moć APEX-a. Otkrit ćete da nakon što definirati koji su vam podaci, možete odmah generirati privlačnu i funkcionalnu aplikaciju bez programiranja. Također se može odmah generirati osnovna kontrola pristupa za krajnje korisnike s različitim ulogama.

Poglavlje 6: “Kako upravljati izvješćima?” vodi vas kroz načine prikaza podataka putem klasičnih i interaktivnih čarobnjaka za izvješćivanje. APEX izvješća već uključuju funkcionalnosti za prilagodbu izvješća krajnjem korisniku bez ikakvog programiranja.

Poglavlje 7: “Kako upravljati obrascima?” vas upoznaje s tri uobičajene vrste web obrazaca uključujući master-detail obrasce. Prilagoditi ćete i generirati stranice obrazaca bez vještina programiranja.

Poglavlje 8: “Kako pretvoriti tekstualna izvješća u grafikone?” pokazuje put za korištenje APEX mogućnosti za predstavljanje podataka u obliku grafikona neposredno uz tekstualna izvješća.

Poglavlje 9: “Kako upravljati izbornicima?” predstavlja vrste različitih navigacijskih elemenata za APEX aplikaciju.

Poglavlje 10: “Kako surađivati u timu?” nudi uvid u APEX funkcionalnosti koje služe timovima jer je rijetka situacija da razvoj aplikacija uključuje samo jednog programera.

Poglavlje 11: “Kako iskoristiti galeriju oglednih/početnih aplikacija i dodataka?” poziva vas da primijenite moćne APEX mogućnosti kroz ponovnu upotrebu dobrih uzoraka.

Poglavlje 12: “Kako upravljati pakiranim i višejezičnim aplikacijama?” postavlja put za distribuciju aplikacije u druga APEX okruženja za korisnike koji govore različite jezike.

Dio I također pokriva teme koje su ključne za sigurnost aplikacija, strategije implementacije i spremnost za stvarni svijet.

Dio II ovog udžbenika vodi vas dalje od osnova, predstavljajući dvanaest zanimljivih poslovnih slučajeva od opisa problema, preko svakog tehničkog detalja za rješavanje problema. Svaki slučaj je pažljivo dokumentiran kako bi se pružilo holističko razumijevanje razvoja aplikacija iz perspektive poslovanja, podataka i korisničkog sučelja. Ovaj dio uključuje aplikacije za tvrtke:

- intranet novosti za zaposlenike,
- mali inovacijski sustav,
- upravljanje poslovnim procesima s tijekovima rada,
- proračun materijala,
- sustav recenzije knjiga,
- prehrana i upravljanje dijetom,
- raspored radnog vremena,
- naplata telekomunikacijskih tvrtki,
- posao iznajmljivanja automobila

za zajednice:

- katalog biljaka,
- razmjenu biljaka

i opće primjenjiva autorizacija i upravljanje korisnicima. U svakom poslovnom slučaju ćete istraživati:

- Poslovni pogled na slučaj: Pregled poslovne situacije.
- Definicija problema: Potraga za odgovorom tko i zašto nekoga boli glava.
- Slučajevi korištenja: Predstavljene su tri vrste opisa: narativni, polustrukturirani i grafički za pripremu UML dokumentacije o slučaju upotrebe.
- Logički i relacijski model podataka: APEX ima sve značajke za pokretanje novih struktura podataka, korištenje i izmjenu postojećih, kombiniranje s drugim alatima za modeliranje podataka i podršku za unaprijedni ili obrnuti inženjering. Napori programera da osiguraju odgovarajuće dijelove podataka i odnose između i razmatranje poslovnih potreba temelj su za nastavak s korisničkim sučeljima.
- Aplikacijska sučelja: udžbenik nudi HTML stranice, izvješća, obrasce, polja, izbornike, gumbe, hiperveze koji materijaliziraju poslovnu situaciju, rješenje poslovnog problema, slučajeve korištenja i podatke s krajnjim korisnikom na umu.
- Dodatni materijal za učenje: S ciljem poboljšanja, ubrzanja i pomoći na putu razvoja, pronaći ćete poveznice na izvezene aplikacije, skripte, podatke i video upute za svako poglavlje. Ovi resursi pružit će vam praktične uvide, omogućujući vam da ojačate svoje znanje i izravno ga primijenite na projekte iz stvarnog svijeta.

Bilo da ste iskusan programer koji želi proširiti svoje vještine ili početnik željan istražiti svijet APEX-a, ovaj udžbenik je dobar vodič. Nadamo se da će i oni koji ne uče informatičke tehnologije pronaći neprocjenjivog suputnika na putu ka svladavanju Oracle APEX-a i izgradnji inovativnih aplikacija koje imaju pozitivan učinak.

Udžbenik i i dopunski materijali predviđeni su za približno 75 sati studentskog rada (3 ECTS). Nadamo se da se mogu primijeniti različiti načini studiranja:

- nastavnik vodi predavanja i laboratorijske vježbe,
- mješovito učenje te
- samostalno učenje.

Ovisno o osnovnom znanju studenata i raspoloživom vremenu za izvođenje tečaja, nastavnici mogu lako sastaviti skup poglavlja koja odgovaraju situacijama studenata kao što su: izvannastavni tečajevi, ljetne škole, vremenski ograničeni događaji za uvode s malo koda za sve studente (ne samo u IT-u ili računalnim znanostima, te praktičari u raznim granama industrije.

APEX verzije 22 i 23 korištene su za izradu ovog udžbenika i dopunskog nastavnog materijala. Vjerujemo da će koncepti i temeljne tehnologije objašnjeni i primijenjeni u sadržaju biti korisni i učenicima budućih verzija APEX-a.

Pripremite se da se upustite u uzbudljivu avanturu učenja uz Oracle APEX! Uživajte u korištenju čarobnjaka i malo kodiranja!

Profesor Robert Leskovar
voditelj projekta BeeAPEX, Voditelj IT odsjeka, UM, Fakulteta za organizacijske znanosti

Kako to učiniti u APEX-u

1 **Kako početi s Oracle APEX-om?** 35
Vjeran Strahonja, Dijana Oreški . . . . . .

2 **Kako pripremiti bazu podataka?** 47
E. Kapsammer, W. Retschitzegger, W. Schwinger . . . . .

3 **Kako koristiti APEX?** 79
E. Kapsammer, W. Retschitzegger, W. Schwinger . . . . .

4 **Kako razmijeniti podatke u APEX-u?** . . 91
E. Kapsammer, W. Retschitzegger, W. Schwinger . . . . .

5 **Kako generirati prvi nacrt aplikacije?** 111
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . .

6 **Kako upravljati izvješćima?** 135
Zuzana Žillová, Erik Malina, Matej Grochal, Andrej Staník, Andrea Meleková, Michal Kvet, Miroslav Potočár

7 **Kako upravljati obrascima?** 157
Veronika Šalgová, Michal Kvet, Miroslav Potočár . .

8 **Kako pretvoriti tekstualna izvješća u grafikone?** 165
Ivan Pastierik, Michal Kvet, Miroslav Potočár . . . . . .

9 **Kako upravljati izbornicima?** 181
Veronika Šalgová, Michal Kvet, Miroslav Potočár . .

10 **Kako surađivati u timu?** 185
P. Staniszewski, M. Sońta, A. Kierzkowski . . . . . .

11 **Kako iskoristiti galeriju oglednih/početnih aplikacija i dodataka?** 197
Vjeran Strahonja, Dijana Oreški . . . . . .

12 **Kako upravljati pakiranim i višejezičnim aplikacijama?** 205
R. Leskovar, U. Rajković, A. Baggia . . . . . .



1. Kako početi s Oracle APEX-om?

VJERAN STRAHONJA, DIJANA OREŠKI

1.1 Što je Oracle APEX i čemu služi?

1.1.1 Što je Oracle APEX?

Oracle APEX (također poznat kao APEX ili Oracle Application Express) je razvojna platforma s malo koda razvijena od strane Oracle Corporation, a koja korisnicima omogućuje izradu i implementaciju aplikacija pomoću web temeljenog sučelja. To je skup alata za izradu različitih vrsta aplikacija koji se sastoje od programa za rad s podacima u bazi podataka, uključujući unos, pohranu, prikaz, promjenu, izračunavanje i logičke operacije s podacima itd. Korištenje ove platforme omogućuje razvoj vlastitih aplikacija za različite namjene koje uključuju generiranje baze podataka na temelju skupa strukturiranih podataka ili podatkovnih modela, direktan rad s podacima u bazi pomoću SQL jezika, izradu korisničkih sučelja, generiranje raznih tabelarnih i grafičkih izvješća i postavljanje web aplikacija na Oracle baze podataka.

1.1.2 Što je aplikacija?

Aplikacija je općenito skup računalnih programa koji imaju određenu svrhu i rade na skupu podataka u bazi podataka. Posebno su nam zanimljive poslovne aplikacije, odnosno one čiji programi služe za obradu podataka o poslovnim procesima i transakcijama. Razvoj aplikacije, čak i one jednostavne, složen je misaoni i kreativni proces koji zahtijeva određena znanja i vještine. Prije svega, potreban je algoritamski način razmišljanja (algoritamska pismenost). To je sposobnost konceptualizacije, svijest da algoritam postoji u situaciji, sposobnost njegove primjene, znanje o tome kako dizajnirati i koristiti algoritam te sposobnost kritičke procjene algoritama. Potrebna su i znanja i vještine izrade visokokvalitetnih zapisa u programskom jeziku (kodiranje), testiranja i otklanjanja pogrešaka te drugih stvari koje obično nazivamo programiranjem.

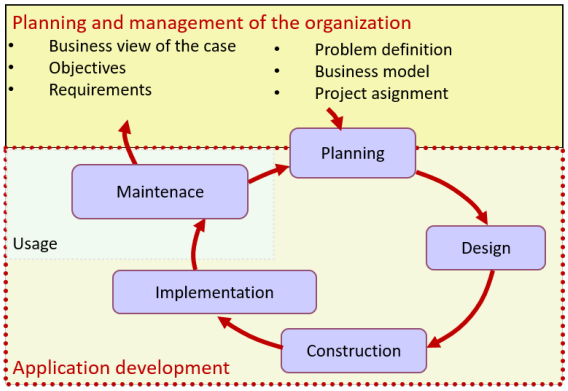
1.1.3 Kako Oracle APEX može pomoći u razvoju aplikacija?

Pri razvoju računalnih programa ne možemo izbjeći algoritamski način razmišljanja. Korištenje platformi i alata za programiranje s niskim kodom/bez koda, kao što je Oracle APEX, omogućuje ne-programerima jednostavno programiranje bez potrebe za poznavanjem pojedinosti programskog jezika i naprednih vještina programiranja. Tako se mogu usredotočiti na problem i algoritam, a ne na programski jezik i tehnologiju. Posebno smo zainteresirani za brz, korisniku prilagođen razvoj aplikacija za pristup, integraciju, analizu i prikaz podataka iz otvorenih izvora podataka i iz

korporativnih baza podataka. Otvoreni podaci su slobodno dostupni podaci kojima svatko može pristupiti, koristiti ih i dijeliti. Mogu se koristiti bez ograničenja (ili s nekima), dijeliti s drugima ili koristiti za stvaranje novih djela. Stavljaju ih na raspolaganje vlade, tvrtke i druge organizacije radi promicanja transparentnosti i suradnje. Danas postoje mnogi lako dostupni izvori meteoroloških, prometnih, geoloških zdravstvenih i sličnih podataka, kao i statistike o obrazovanju, zdravstvu, gospodarstvu, itd. Korporativne baze podataka sadrže podatke o poslovnim transakcijama i druge podatke koji nastaju kao rezultat poslovnih procesa. Vizualizacija, analiza, integracija i različiti pogledi na te podatke mogu poslužiti kao potpora za donošenje poslovnih odluka ili postizanje nekih drugih prednosti. Osim za korisnički razvoj, Oracle APEX se također koristi za razvoj prototipa, koji se koristi za raspravu o tome što program treba raditi i pokazuje IT stručnjaku što korisnik želi. Prototip odražava bitne karakteristike sustava. Kasnije se dodaje u sustav koji će se koristiti ili služi kao osnova za izradu sustava u nekom drugom programskom jeziku, alatu ili platformi.

1.1.4 Ciklus razvoja aplikacije

Bez obzira na to kakvu aplikaciju razvijamo i kakav metodološki okvir, razvojne alate ili platformu koristimo, razvojni ciklus aplikacije prolazi kroz nekoliko faza. Ove su faze različite u različitim metodološkim pristupima. Slika 1.1 prikazuje jedan mogući obrazac ciklusa razvoja aplikacije (vidi Slika 1.1).



Slika 1.1: Ciklus razvoja aplikacije

Kao što je prikazano na slici, razvojni ciklus aplikacije sadrži sljedeće faze ili procese:

1. Planiranje (definiranje problema i projekta):
   - definiranje i analiza poslovnog modela,
   - specifikacija zahtjeva,
   - određivanje osnovnih procesa i podataka,
   - odluke o tehnologiji i arhitekturi,
   - planiranje razvoja.
2. Dizajn aplikacije:
   - analiza i specifikacija zahtjeva,
   - logički dizajn aplikacije (slučajevi korištenja i logički model podataka).
3. Razvoj aplikacije:
   - dizajn baze podataka,
   - razvoj (prototip) aplikacije,
   - testiranje.
4. Implementacija aplikacije.

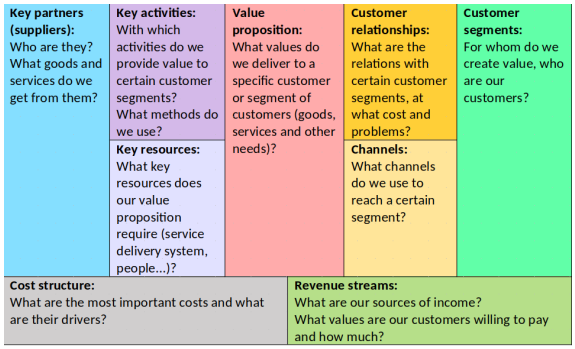
Nakon planiranja slijede projektiranje, razvoj i implementacija. Te se faze ili procesi mogu izvoditi sekvencijalno ili iterativno (evolucijski, inkrementalno), tako da cijeli razvojni ciklus

prolazimo nekoliko puta i u svakoj iteraciji dodajemo dio funkcionalnosti.

Planiranje aplikacije Ciklus razvoja aplikacije počinje planiranjem. Kod jednostavnih aplikacija početna informacija za planiranje aplikacije može biti definicija problema, dokument koji jasno navodi problem. Koga boli glava? Zašto je ovo prava glavobolja? Što je manifestacija problema? To može biti nekoliko rečenica ili stranica teksta. U slučaju složenih poslovnih aplikacija, detaljne specifikacije kao što su:

- Poslovni pogled na slučaj opisuje kako ljudi unutar organizacije percipiraju situacije ili očekivanja od aplikacije izražena prije razvoja aplikacije.
- Ciljevi razvoja aplikacije opisuju što se aplikacijom želi postići u poslovnom, tehnološkom i drugom smislu te pokazatelje uspješnosti.
- Poslovni model opisuje vrijednosti, segmente kupaca, odnose i kanale s kupcima, ključne partnere, aktivnosti i resurse, strukturu i izvore prihoda i troškova itd.
- Specifikacija softverskih zahtjeva definira zahtjeve koji se odnose na funkciju, ponašanje, performanse, tehnologiju itd.
- Projektni plan opisuje opseg i faze budućeg projekta, aktivnosti (zadatke) koji se provode u pojedinim fazama, ulaze i izlaze pojedinih aktivnosti, resurse koji su potrebni, međusobnu povezanost aktivnosti i njihovo trajanje, troškove, rizike, itd.

Za izradu svake od navedenih specifikacija postoje metode koje su opisane u literaturi. Na primjer, Alexander Osterwalder [6] je razvio metodu Platno poslovnog modela (engl. Business Model Canvas) koja se koristi za razvoj, poboljšanje i dokumentiranje novih i postojećih poslovnih modela. Kao što je prikazano na slici 1.2, to je zapravo grafikon od devet blokova koji opisuju ponudu vrijednosti koju pruža organizacija, njezini kupci, odnosi i kanali s njima, ključni partneri, aktivnosti i resursi, kao i struktura troškova i tokovi prihoda.



Slika 1.2: Platno poslovnog modela [6]

.

Dok Platno poslovonog modela služi za poticanje razumijevanja, analize i dizajna poslovnog modela, a posebno onoga što treba podržati aplikacija koju razvijamo, a specifikacija softverskih zahtjeva (SSZ) je dokument koji opisuje što će softver raditi i kako će biti očekuje se izvođenje. Opisuje funkcionalnost, performanse i druge karakteristike koje aplikacija mora imati kako bi podržala poslovni model i potrebe svih korisnika.

Dizajn aplikacije Od prvih računala i računalnih programa do danas razvijene su stotine različitih metodoloških pristupa, metoda i tehnika projektiranja aplikacija. Metodologija izrade programa mijenjala se s tehnološkim razvojem. Razvoj programa s malo koda ili bez koda je pristup koji preferira jednostavne metode, razumljive korisnicima koji nisu profesionalni programeri softvera. Međutim, razvoj aplikacija zahtijeva određeni dizajn prije generiranja programa. Kao što smo već spomenuli, računalna aplikacija je skup programa koji rade s podacima u bazi podataka. Kako bismo razvili aplikaciju, moramo dizajnirati i programe i bazu podataka. Ako baza podataka već postoji, moramo je znati koristiti, a prije svega razumijeti podatkovni model. Za potrebe malo

kodnog razvoja programa uvest ćemo najpotrebnije metode projektiranja. Ujedno, temelj dizajna je izrada modela koji imaju i grafički prikaz. Koristit ćemo sljedeće metode:

- Model slučaja korištenja (UC) koji definira odnos između aplikacije i okoline.
- Model entiteti-veze (ER) koji opisuje strukturu podataka na logičkoj razini.

Dijagram slučajeva korištenja će biti objašnjen u nastavku ovog poglavlja, dok će dijagram entiteti veze (ER) biti objašnjen u Poglavlju 2.

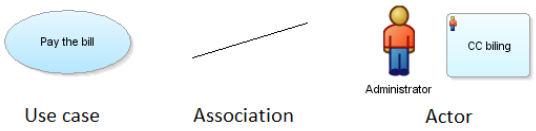
Model slučajeva korištenja Slučajevi korištenja (engl. use case, UC) su jedna od metoda razvoja softvera. Oni opisuju odnos između sustava (aplikacije) i okoline (korisnika i drugih sustava). Metoda je jednostavna, pa je razumiju i korisnici koji nisu profesionalci. Slučajevi korištenja sastavni su dio Unified Modeling Language (UML), danas najkorištenijeg jezika za modeliranje u području programskog inženjerstva, koji pruža standardni način vizualizacije dizajna sustava. Postoji bogata literatura o UML-u, a o njegovom razvoju i standardizaciji brine Object Management Group (OMG). UML sadrži 13 metoda i dijagramskih tehnika za modeliranje strukture, ponašanja i interakcije softvera, a mi ćemo koristiti samo slučajeve korištenja.

Model slučaja korištenja sastoji se od dva dijela:

- specifikacije (narativni opis) slučajeva korištenja,
- dijagram slučaja korištenja.

Prvo ćemo proučiti dijagram slučaja korištenja. Ovaj dijagram opisuje što sustav radi sa stajališta promatrača izvan sustava. Nije važno kako sustav funkcionira iznutra.

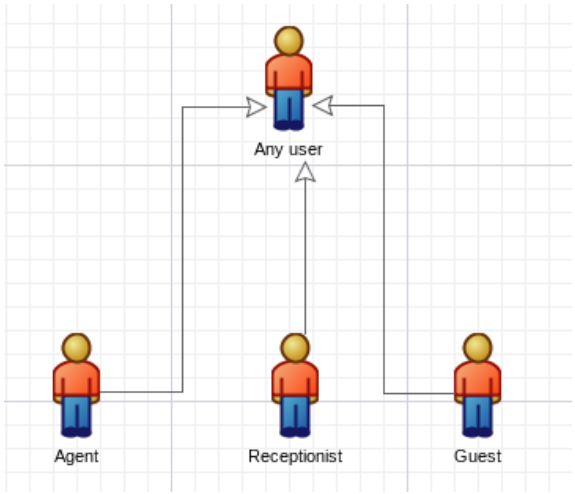
Osnovni koncepti dijagrama slučajeva korištenja su prikazani na slici Slika 1.3.



Slika 1.3: Osnovni koncepti dijagrama slučajeva korištenja

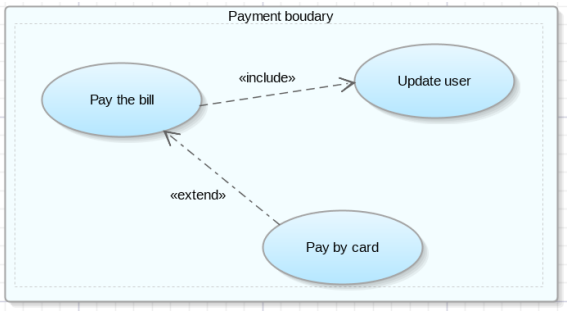
Slučaj korištenja predstavlja skup akcija koje može izvesti sustav (npr. softverska komponenta) u interakciji s vanjskim akterima. To je priča koja opisuje kako akteri koriste sustav za postizanje određenih ciljeva ili obavljanje zadataka. Predstavlja apstraktni zadatak koji ima svrhu. Veza (engl. Association) povezuje sudionike u komunikaciji, npr. sudionika i slučaj korištenja te predstavlja njihovu interakciju i odnos između sustava i okoline (ponašanje). Sudionik (engl. Actor) predstavlja skup uloga koje su u interakciji sa sustavom na isti način, na primjer klasa korisnika, neka vrsta vanjskog sustava, itd. Sudionik je netko izvan sustava o kojem se raspravlja i koji je u interakciji s njim, ali nije njegov dio. To može biti živo biće (Korisnik, Pacijent, Pilot), ili drugi sustav (Billing System, Banka, Carrier).

Dijagram također može prikazivati strukturu klasifikacije (vidi Sliku 1.4), tj., generalizacija/specijalizacija koncepata, npr. Sudionika. Sve potklase (djeca) nasljeđuju svojstva i ponašanje od nadklase (roditelja), što znači da također nasljeđuju veze za slučajeve korištenja od roditelja. Istovremeno, pojedina podklasa može stupiti u komunikaciju u koju ne ulazi niti nadklasa niti druge podklase.



Slika 1.4: Generalizacija/specijalizacija Sudionika

Slika 1.5 prikazuje granice sustava kao pravokutnik s nazivom sustava unutar kojeg se nalaze slučajevi korištenja.

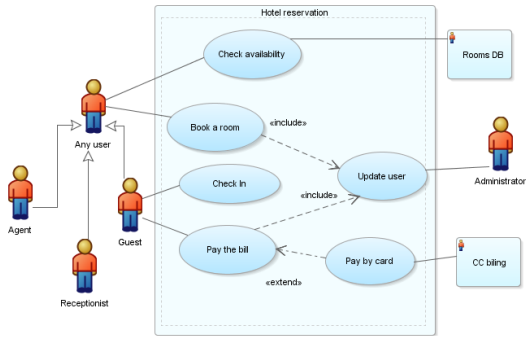


Slika 1.5: Granice sustava i veze "include" i "extend"

UC dijagrami ne prikazuju tijek rada ili redoslijed slučajeva korištenja. Međutim, postoje dvije dopuštene vrste povezivanja između slučajeva korištenja koje su također prikazane na Slici 1.5:

- "include" je veza između osnovnog slučaja korištenja i uključenog slučaja korištenja, što ukazuje da osnovni slučaj korištenja sadrži ponašanje uključenog slučaja korištenja. Na taj način, funkcionalnost i ponašanje koje se često koristi razdvajaju se u slučajeve korištenja koje ćemo po potrebi uključiti u druge slučajeve korištenja. Treba imati na umu da osnovni slučaj korištenja nije potpun bez uključivanja.
- "extend" je veza iz slučaja korištenja koji je proširenje na osnovni slučaj korištenja, što ukazuje da je ponašanje osnovnog slučaja korištenja prošireno ponašanjem proširenja. Istodobno, ekstenzija nije dio osnovnog slučaja korištenja, niti ta komunikacija treba uvijek postojati i funkcionirati besprijekorno.

Korištenje gore navedenih koncepata dijagrama slučajeva korištenja najbolje je objašnjeno jednim primjerom, kao što je sustav hotelskih rezervacija, prikazan na slici 1.6.



Slika 1.6: Dijagram slučajeva korištenja sustava rezervacije hotela

Sustav čije ponašanje promatramo je rezervacija hotela. Cijeli sustav je unutar granica prikazanih pravokutnikom. Okolina sustava je izvan pravokutnika. Unutar granica sustava nalazi se šest slučajeva korištenja, prikazanih elipsama s imenima. Izvan sustava su Sudionici, koji komuniciraju sa sustavom, odnosno njegovim slučajevima korištenja, a ta se komunikacija prikazuje povezujućim linijama, vezama. Bilo koji korisnik je sudionik koji komunicira s Provjeri dostupnost. Ovaj slučaj korištenja komunicira s vanjskim sudionikom, Rooms DB. To je neka vrsta vanjskog rezervacijskog sustava koji u svojoj bazi podataka ima podatke o sobama i rezervacijama. Svaki korisnik može rezervirati sobu. Ovaj UC uključuje još jedan UC, Update User. Klasifikacijska struktura sudionika User prikazana je lijevo. Agent, Recepcionar i Gost su "neka vrsta sudionika Any user". U prikazanom slučaju, samo Gost može komunicirati s UC-om Prijava i plaćanje računa. UC Ažuriranje korisnika uključen je u Pay the bill, što znači da se korisnički podaci također mogu ažurirati unutar funkcije plaćanja računa. UC Pay by card proširuje funkciju Pay the bill, to znači da se plaćanje karticom može pozvati tijekom plaćanja računa. Pay by card komunicira s vanjskom uslugom plaćanja kreditnom karticom CC billing. Administrator može komunicirati s UC-om Ažurirajte korisnika i ažurirajte podatke o korisniku. Kao što je već spomenuto, model slučaja korištenja sastoji se od dva dijela:

- specifikacije (narativnog opisa) slučaja korištenja,
- dijagrama slučaja korištenja.

Specifikacija slučaja korištenja opisuje scenarije i internu logiku slučajeva korištenja, početna stanja (preduvjete) i završna stanja, sučelja, sistemske poruke, specifikaciju obrade grešaka i izuzetaka i slično. Obično se za specifikaciju slučajeva korištenja koristi neka vrsta predloška, stoga ćemo predložak koristiti i u ovoj knjizi.

Izrada, implementacija i održavanje aplikacije Izrada aplikacije podrazumijeva izradu programa i realizaciju fizičke baze podataka. Za to ćemo koristiti Oracle ASPEX platformu, kao što će biti objašnjeno kasnije u knjizi. Prva verzija aplikacije koja radi i može se testirati zapravo je prototip aplikacije. Ovaj prototip se može nadograditi i proširiti. S obzirom na to što će se kasnije dogoditi s prototipom, razlikujemo dvije vrste prototipova: Nadogradivi prototip se poboljšava i nadograđuje u stanje koje je prikladno za upotrebu. Prototip "za bacanje" služi kao ilustracija funkcionalnosti i model na kojem će se graditi aplikacija, najčešće na drugoj platformi. Ako namjeravamo koristiti prototip aplikacije, moramo ga implementirati, odnosno instalirati u produkcijsko okruženje, educirati korisnike, pripremiti bazu i slično.

Nakon faze Implementacije slijedi Korištenje i Održavanje, što uključuje popravljanje i uklanjanje grešaka, poboljšanje performansi i manja proširenja funkcionalnosti. Ukoliko su potrebne veće izmjene, kreće se u novi razvojni ciklus aplikacije.

1.2 Kako početi s Oracle APEX-om?

Razvoj softvera smatra se složenim zbog puno programiranja. Kako bi se prevladala složenost, predlažu se nisko kodna rješenja. Pristup s malo koda omogućuje razvoj sofisticiranih aplikacija koje su i funkcionalne i složene bez pisanja retka koda. Oracle Application Express jedna je od najboljih opcija s malo koda. Postoje različiti načini pokretanja Oracle APEX-a. Oracle APEX može se koristiti bilo gdje gdje se izvodi baza podataka, kako lokalno tako i u privatnom oblaku. To može biti fizički, namjenski poslužitelj, virtualizirani stroj ili docker slika (koja se posebno sviđa programerima Oracle Application Expressa i može se pokrenuti na prijenosnom računalu, u autobusu ili u avionu). Osim toga, može se koristiti Exadata, supersposoban fizički APEX poslužitelj na uslugama u oblaku. Ovdje će se opisati sve opcije. Tema poglavlja je kako pristupiti razvojnom okruženju.

1.2.1 Koja je razina Vaših vještina?

Postoji nekoliko mogućnosti za početak s Oracle Application Express - APEX. Vaša trenutna razina vještina određuje preporučeni način za početak. Pretpostavimo da imate specifične domenske kompetencije u bilo kojem organizacijskom procesu (npr. prodaja, zapošljavanje, proizvodnja, posuđivanje, marketing, pravni poslovi, zdravstvena skrb, izgradnja, obrazovanje) i sljedeće razine digitalnih vještina:

- apsolutni početnik: upoznat radom u web pregledniku, bez ikakvog iskustva u programiranju; preporuka: koristite on-premise instancu ili apex.oracle.com ili Oracle Academy,
- početnik: upoznat radom u web pregledniku, malo programerskog iskustva npr. korištenje proračunskih tablica i postavljanje formula; bilo koji drugi programski jezik; preporuka: koristite on-premise instancu ili apex.oracle.com ili Oracle Academy,
- fresh citizen developer: upoznat radom u web pregledniku, malo programerskog iskustva; razumijevanje što je baza podataka, tablica, stupac tablice, primarni ključ, vanjski ključ; postavljanje upita, umetanje, ažuriranje i brisanje; preporuka: koristite on-premise instancu ili apex.oracle.com ili Oracle Academy ili VBox uređaj,
- skilled citizen developer: upoznat radom u web pregledniku; umjereno programersko iskustvo s procedurama i funkcijama u bilo kojem sustavu za upravljanje bazom podataka; upiti, umetanje, ažuriranje i brisanje; preporuka: koristite on-premise instancu ili apex.oracle.com ili Oracle Academy ili VBox uređaj ili Apex docker,
- on the road to professional developer: razvojne vještine u PL/SQL i Javascriptu; preporuka: koristite on-premise instancu ili apex.oracle.com ili APEX docker ili OCI APEX instancu.

Vaša korisnička uloga i lokacija Oracle APEX-a odredit će kako ćete se prijaviti i koristiti aplikaciju. Oracle APEX može se instalirati lokalno na vašem računalu ili u okruženju kao što je usluga Oracle Cloud. Ovisno o vrsti instalacije, postupak prijave se razlikuje. Korisnici trebaju kreirati radni prostor, dodati Oracle APEX korisnike, zatim se prijaviti u radni prostor prije razvoja ili instaliranja aplikacija. Više korisnika može surađivati na istoj Oracle APEX instalaciji koristeći radni prostor uz zadržavanje privatnosti svojih objekata, podataka i aplikacija. Ako web preglednik podržava JavaScript, korisnici se mogu prijaviti u radni prostor za pristup Oracle APEX početnoj stranici. Svaki radni prostor ima poseban naziv i ID. Unutar Oracle APEX Administration Services, radni prostor može ručno kreirati administrator instance ili korisnici mogu postavljati zahtjeve. Neovisni program pod nazivom Oracle APEX Administration Services koristi se za nadgledanje kompletne Oracle APEX instance.

1.2.2 Korištenje lokalne APEX instance

Ova sekcija opisuje kako instalirati Oracle APEX u lokalnu instalaciju. Oracle APEX instalacija uključuje nekoliko koraka. Ti koraci su:

- **odaberite između punog (engl. Full) ili izvršnog (engl. Runtime) okruženja**. Puno

okruženje daje potpuni pristup razvojnom okruženju App Builder. Za produkcijske implementacije gdje želite izvršavati nepromjenjive aplikacije, izvršno okruženje je dobra opcija. Oracle APEX omogućuje instaliranje samo runtime verzije Oracle APEX-a za testiranje i produkcijske instance. Budući da programeri ne mogu slučajno ažurirati produkcijski program u runtime instanci, ovo runtime okruženje smanjuje instalirani trag i prava te poboljšava sigurnost aplikacije. Korisnici mogu pokretati produkcijske aplikacije u Oracle APEX runtime okruženju, ali ono nema web sučelje za administraciju. Izvršno okruženje je sigurnije jer sadrži samo komponente potrebne za pokretanje programa.

- **Provjerite instalacijske zahtjeve.** Provjerite zadovoljava li vaš sustav minimalne zahtjeve za instalaciju. Postoji pet skupina zahtjeva: a) Zahtjevi Oracle baze podataka (Oracle APEX izdanje 22 zahtijeva izdanje Oracle Database 12.1.0.2 ili novije. Oracle APEX radi na svim izdanjima baze podataka, uključujući Enterprise Edition (EE), Standard Edition (SE) i Express Edition (XE). Oracle APEX može se instalirati u bazi podataka s jednom instancom i u bazi podataka Oracle Real Application Clusters (Oracle RAC).), b) Zahtjevi preglednika (Oracle APEX zahtijeva preglednik s omogućenim JavaScriptom i podržava trenutno i prethodno glavno izdanje Google Chrome, Mozilla Firefox, Apple Safari i Microsoft Edge.), c) Zahtjevi web listener (Oracle APEX zahtijeva Oracle REST Data Services (ORDS) 19.x ili noviji), d) Zahtjevi za prostor na disku (Oracle APEX zahtjevi za prostor na disku 310 MB slobodnog prostora za APEX softverske datoteke na datotečnom sustavu ako koristite preuzimanje samo na engleskom jeziku i 705 MB ako koristite potpuno preuzimanje, 220 MB slobodnog prostora u APEX tabličnom prostoru, 100 MB slobodnog prostora u SYSTEM tabličnom prostoru, 60 MB slobodnog prostora u APEX tabličnom prostoru f ili instaliran svaki dodatni jezik (osim engleskog), e) Zahtjevi za Oracle XML DB (Oracle XML DB mora biti instaliran u Oracle bazi podataka koju želite koristiti ako instalirate potpuno razvojno okruženje. Ako koristite unaprijed konfiguriranu bazu podataka stvorenu ili tijekom instalacije ili pomoću Database Configuration Assistant ili DBCA, Oracle XML DB je već instaliran i konfiguriran.)
- **Instaliranje** softvera - Instalirajte Oracle APEX preuzimanjem ZIP datoteke sa stranice Oracle APEX, a zatim preuzmite i instalirajte Oracle REST Data Services(ORDS).

1.2.3 Opcija https://apex.oracle.com

Idite u pregledniku do this URL. Zahtjev za besplatnim radnim prostorom najbrži je način za početak rada s Oracle APEX-om. Potrebno je samo nekoliko sekundi da se pridružite, a radni prostor je spreman za korisnike da počnu razvijati aplikacije. Ovaj pristup ne zahtijeva račun u oblaku i besplatan je. Za početak s Oracle APEX-om:

- Upišite u svoj web preglednik: apex.oracle.com.
- Kliknite na gumb Start for Free Today.
- Na web stranici koja se pojavi odaberite: Request a free Workspace.
- Unesite svoje ime, adresu e-pošte, dajte naziv svom radnom prostoru i kliknite Next.
- Ispunite anketu i navedite zašto tražite radni prostor. Pročitajte uvjete i odredbe i prihvatite ih, kliknite Next.
- Nakon toga ćete dobiti e-poruku od Oracle APEX-a, kliknite na vezu za registraciju i postavljanje lozinke.

1.2.4 Virtualni stroj

Upišite u svoj web preglednik: this URL. Testiranje, preduvjeti, APEX verzija je iza najnovije. Preuzmite Oracle Virtual Box i Extension Manager. Postavite Oracle Virtual Box i Extension manager. Uvezi .ova datoteku. Pokrenite.

Oracle APEX Development VM je virtualni stroj spreman za korištenje koji možete koristiti jednostavnim uvozom u VirtualBox. Oracle Pre-Built Virtual Machines dostupni su uz Oracle

račune. Sve što trebate učiniti je instalirati Oracle Virtual Box (Free Virtual Machine Client), a zatim uvesti bilo koji uređaj (Pre-Built VM). Oracle ima nekoliko Pre-Built VM dostupnih na URL.

Da biste postavili VM, učinite sljedeće:

- Preuzmite i instalirajte Oracle VM VirtualBox na svoj sustav.
- Preuzmite datoteke (preporučljivo je korištenje upravitelja preuzimanja)
- Uvezite svoj VM: File > Import Appliance to launch Appliance Import Wizard. kliknite Choose... za pregledavanje direktorija u koji ste ponovno sastavili sve datoteke i odaberite OTN_Developer_Day_VM.ova. Zatim kliknite Next za početak uvoza virtualnog stroja. Tijekom uvoza od vas će se tražiti da pristanete na odgovarajuće licence razvojnog programera. Vidjet ćete 'Oracle Developer Days (Powered Off) kada završi s uvozom.
- Testirajte VM: Nakon što je uvoz završen, dvaput kliknite OTN Developer Days VM. Pritisnite OK da biste zatvorili dijaloške okvire s informacijama o Virtualboxu. Kada dođete do zaslona Enterprise Linux 6, sada se možete prijaviti. (Korisničko ime i lozinka su Oracle.) Dopustite da se proces završi; spreman je kada vidite prozor terminala koji možete zatvoriti. Nakon što završite s radom u gostujućem VM-u, možete ga ugasiti preko System > Shut Down; ovo će vratiti gostujući VM u isključeno stanje.

Ključne prednosti korištenja Virtual Machines uključuju:

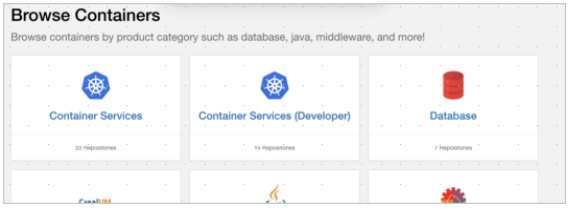
- Korištenje novog Oracle softvera bez potrebe za izvođenjem složenih instalacija ili potrebe za uključivanjem dobavljača.
- Eksperimentirajte bez izlaganja svoje radne stanice.
- Besplatna edukacija koju nudi Oracle u obliku Hands On Labs.

1.2.5 APEX docker

Nabavite službeni APEX docker na Oracle web mjestu. Idite u pregledniku na: this URL.

Oracle APEX može se instalirati kao Docker spremnik. Dockeri su zgodan način za programere jer se mogu instalirati na računala i prijenosna računala. Preporuka je da se APEX docker nabavi samo iz pouzdanih izvora. Postavljanje je obično brzo i jednostavno. Za učenje Oracle Database i APEX njihova verzija možda nije važna, ali za produkcijsku upotrebu obratite pozornost na verzije. Ostatak ove sekcije opisuje jedan mogući način lokalne instalacije APEX dockera na vašem računalu. U ovom procesu kloniramo službenu Oracle docker sliku koja sadrži razvojne alate.

1. Prvi korak je instaliranje dockera na uređaj. Docker se može lako preuzeti i instalirati s početne stranice dockera: https://docs.docker.com/desktop/install/windows-install/
2. Drugi korak je prijava za besplatni račun na Oracle registar kontejneru: https://container-registry.oracle.com/ords/f?p=113:10 Docker slike su izvučene iz ovog registra (vidi Sliku 1.7).



Slika 1.7: Searching for the APEX docker.

3. Treći korak je otvaranje prozora terminala i pristup Oracle registru pomoću prethodno kreiranog korisnika.

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

   Bolje je stvoriti mrežu u docker okruženju kako bi dockeri mogli komunicirati s drugim dockerima koristeći ime glavnog računala.

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

U ovom smo trenutku učinili sve što trebamo i možemo izvršiti naredbu za pokretanje Oracle Database XE u dockeru na računalu.

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

Imajte na umu da su ovdje korišteni naziv parametra, naziv hosta i mreža. Obrazloženje:

- Prva zastavica -d pokrenut će spremnik u odvojenom načinu rada.
- Parametar –name navodi naziv spremnika.
- -p preslikava port 1521 na glavnom računalu u port 1521 u spremniku, tako da se možemo povezati s bazom podataka.
- –network povezuje spremnik s mrežom koju smo stvorili.
- –hostname daje ime DB poslužitelju.
- Posljednji parametar je slika koju želimo koristiti za okretanje spremnika.

Ako se želite spojiti sqlplus-om, možete izvršiti ovu naredbu:

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

S naredbom *show pdbs* će te provjeriti status pdbs-a. Promijenite zadanu lozinku pomoću donje naredbe (u ovom slučaju nova lozinka će biti Welcome1!!). Prije nego što izvršite ovu naredbu, provjerite je li pdb XEPDB1 otvoren za čitanje i pisanje).

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

Da biste postavili Apex za instalaciju Oracle XE u dockeru, izvršite daljnje korake. Iz sesije terminala izvucite sliku naloga.

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

Zatim izvršite naredbu pull za preuzimanje slike naloga.

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

Da biste konfigurirali instalaciju Apexa, stvorite direktorij.

```
mkdir ~/APEX
```

Stavite informacije o nizu unutar datoteke u ovom direktoriju.

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

Imajte na umu da je korišten parametar hostname Oracle Database XE dockera i usluga pdb-a za vezu. Definirajte CONN_STRING varijablu: mora biti u donjem obliku (bez jednostrukih navodnika):

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

Sada možemo pokrenuti 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
```

Ovdje je korištena ista mreža parametara. Za nadzor instalacije, možete otvoriti drugu sesiju terminala i izvršiti ovu naredbu:

```
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
```

Promijenite lozinuku za APEX_PUBLIC_USER korisnik unutar baze. Da biste to učinili, prijavite se u Oracle XE docker pomoću sqlplus-a.

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

Postavite lozinku.

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

Sada se možete povezati s APEX okruženjem na vašem lokalnom računalu.

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

1.2.6 APEX instanca u Oracle Cloud Infrastructure

Oracle Cloud Infrastructure (OCI) nudi APEX nisko kodni razvoj aplikacija na autonomnoj infrastrukturi kao potpuno upravljanu uslugu koja je unaprijed konfigurirana i spremna za korištenje. OCI pruža elastičnu skalabilnost, sigurnost, visoku dostupnost i globalni pristup putem regionalnih podatkovnih centara u oblaku. Kako biste stvorili instancu usluge APEX, u ovoj sekciji opisani su postupci za dobivanje pristupa Oracle Cloud Infrastructure (OCI). Koraci uključeni u pristup OCI-ju su sljedeći:

1. **Dobivanje OCI računa**. Korisnik mora imati OCI račun s pristupom OCI računu kako bi koristio APEX uslugu. Korisnik može koristiti postojeći OCI račun ako organizacija ima ugovor o prodaji s Oracleom. Da biste zatražili pristup, prvo stupite u kontakt s OCI administratorom vaše organizacije. Razmislite o Oracle Cloud Free Tieru ako ste samostalni korisnik koji počinje od nule ili niste sigurni odakle početi. Kao i besplatnu početnu dodjelu Cloud Credits, ovo nudi besplatno OCI zakup bez isteka i račun. Tijekom probnog razdoblja, ti se krediti mogu koristiti za kupnju plaćenih OCI usluga, kao što je APEX usluga. Morate nadograditi račun na status plaćanja i kupiti više kredita ako želite nastaviti koristiti APEX uslugu nakon što probno vrijeme završi ili ponestane besplatnih kredita (što god se prije dogodi). Vaš OCI račun promijenit će se u stanje u kojem može koristiti samo OCI usluge koje imaju dostupan Always Free obrazac ako ne izvršite nadogradnju prije isteka probnog razdoblja. Manji Always Free oblik postoji u APEX usluzi. Oracle savjetuje da svoj račun pretvorite u račun koji se plaća i kupite više kredita tijekom ili nakon probnog razdoblja. Započnite postupak prijave pregledom Oracle Cloud Infrastructure Free Tier. Za početak prijave idite na: https://signup.oraclecloud.com/.
2. **Prijavi se na OCI Console.** Upotrijebite kompatibilni web preglednik kako biste pronašli stranicu za prijavu na OCI konzolu. Unesite svoje korisničko ime i lozinku, a zatim naziv vašeg računa u oblaku. Vaša e-pošta dobrodošlice sadrži i vaše korisničko ime i ime računa u oblaku.

1.2.7 APEX instanca u Oracle Academy

Pridružite se Oracle Academy i dobit ćete APEX radni prostor. Oracle Academy pruža Institucionalnim članovima i njihovim studentima pristup Oracle Application Express (APEX) za praktičnu praksu u oblaku. Idite u pregledniku na https://academy.oracle.com/en/oa-web-overview.html.

2. Kako pripremiti bazu podataka?

E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER

Pretpostavimo da radite za *Walmart* i *baze podataka* (DBS) još nisu izmišljene. Pretpostavimo nadalje, da se od vas traži implementacija *web aplikacije* koja može pohraniti, dohvatiti, vizualizirati i dalje obraditi svaku pojedinačnu prodaju u svakoj od Walmartovih 10.500 trgovina zajedno s informacijama o odgovornim odjelima, zaposlenicima i njihovim poslovima koji mogu biti petabajti podataka. Zadatak koji očito nije izvediv jer se suočavate s brojnim izazovima:

- Koliko datoteka i diskova trebate za pohranu?
- Kako pronaći i dohvatiti podatke?
- Kako osiguravate odgovarajuće vrijeme odziva?
- Kako dopuštate izmjene i istovremeni pristup podacima?
- Kako spriječiti neovlašteni pristup podacima?

Ako biste postavili DBS kao okosnicu svoje web aplikacije (Slika 2.1) ti izazovi postaju savladivi budući da DBS pruža odgovarajuću funkcionalnost za njihovo rješavanje.



Slika 2.1: DBS kao okosnica web aplikacija

DBS se sastoji od:

1. softvera za pomoć u upravljanju i obradi velikih kolekcija podataka koje može koristiti web aplikacija, nazvana Sustavi za upravljanje bazom podataka (DBMS),

2. stvarne pohrane samih podataka, tzv. baze podataka koje pružaju DB-sloj na kojem se nadograđuje web aplikacija.

U nastavku se detaljnije raspravlja o glavnim konceptima potrebnim za izgradnju DB-sloja, tj. *razvijanje DB-a*. Iako se, na samom kraju, sve svodi na "*stvaranje odgovarajućih DB tablica, pohranjivanje podataka, manipuliranje i postavljanje upita njima*", postoje neki važni koncepti koji se moraju uvesti kako bi se postigao ovaj cilj.

Konkretno, počet ćemo u poglavlju 2.1 s postupkom modeliranja podataka koji će biti pohranjeni uvođenjem dvije različite *razine apstrakcije* , koje se sastoje od *logičkog modela* i *fizičkog modela* u smislu *relacijskog modela*[1]. Nadalje, u poglavlju 2.2, mehanizmi i alati specifični za DBS za *stvaranje strukture podataka unutar DBS-a*, kako je propisano procesom modeliranja. Na temelju toga, poglavlje 2.3 bavi se problemima pohrane podataka i manipulacije, a poglavlje 2.4 ima za cilj raspravu o postavljanju upita za pohranjene podatke. Konačno, poglavlje 2.5 daje sažetak svih opcija i odgovarajućih alata koji su dostupni za izgradnju DB-sloja web aplikacije.

Kroz sva ova poglavlja koristi se *isti primjer*, inspiriran prethodno predstavljenim slučajem korištenja Walmarta, naglašavajući njegov aspekt *kadrovskih poslova (HR)* u smislu *zaposlenika*, *odjela* i *poslova*, standardni primjer koji također koristi ORACLE u svrhu demonstracije od prve verzije svog DBS-a 1979.

2.1 Kako modelirati podatke

Prvi korak pri realizaciji DB-sloja web aplikacije je fokusiranje na podatke iz stvarnog svijeta koje treba pohraniti. Budući da ovi podaci mogu biti prilično sveobuhvatni kao što smo vidjeli u našem Walmartovom primjeru iznad, ima smisla prvo se baviti podacima na apstraktnijoj razini, ne uzimajući u obzir sve detalje implementacije niske razine određenog DBS-a. Ovo je fokus logičkog modela kao što je objašnjeno u poglavlju 2.1.1. Nakon toga, na konkretnijoj razini i stoga specifičnoj za DBS, relacijski model predstavljen je u poglavlju 2.1.2.

2.1.1 Logički model

Motivacija. *Logički (podatkovni) model* opisuje "stvari" *stvarnog svijeta* (tj. *problemsku domenu*) o kojima, npr. , organizacija poput Walmarta, želi prikupljati i obrađivati podatke. Logički modeli se vizualno opisuju pomoću dijagrama, u praksi često u terminima takozvanih *ERA - entity-relationship (ER)-dijagrama*, bez razmatranja specifičnosti određenog DBS-a. Prednosti ovog vizualnog prikaza su višestruke, npr. povećanje razumljivosti i olakšavanje komunikacije unutar razvojnog tima i s korisnikom.

Postoje brojni različiti grafički formalizmi i alati dostupni za modeliranje ERA dijagrama. Usredotočit ćemo se samo na *ORACLE-ov SQL Developer Data Modeler (ODM)*. Slika 2.2 prikazuje ERA dijagram našeg HR primjera koji će biti detaljnije objašnjen u nastavku.



Slika 2.2: Logički model našeg primjera HR-a pomoću ERA dijagrama

[1]Napominjemo da se za realizaciju fizičkog modela ne može koristiti samo relacijski model koji je fokus ovog poglavlja, već i drugi dostupni su formati za predstavljanje podataka (aka. "podatkovni modeli"), poput *objektno orijentirani podatkovni model* ili *NoSQL podatkovni modeli*.

Isti primjer prikazan je na slikama 2.3, 2.4 i 2.5, ali ovaj put unutar ORACLE-ovog ODM-a . Svi opisani koncepti su označeni kako bi se pojasnili najvažniji grafički zapisi i simboli.

ERA dijagrami – Entiteti, atributi, veze. *ERA dijagrami* opisuju koncepte od interesa i sastoje se od (i) *entiteta* (npr. Odjeli, Zaposlenici, Poslovi) (ii) *atributa* (npr. "naziv odjela" i "naziv lokacije" odjela ili "prezime" i "plaća" zaposlenika) njihove karakteristike i (iii) *veza* (npr. odjel se "sastoji" od nekoliko zaposlenika) između entiteta .

Entiteti. *Entitet* ima jedinstvenu imenicu dodijeljenu kao ime i vizualno je predstavljen kao zaobljeni pravokutnik. Koriste se oblici za množinu i jedninu, no nedavno se preferira jednina (i standardizirana ISO 11179-5).

Atributi. Atributi su uvijek povezani s entitetima, tj. ne mogu postojati neovisno. Za razliku od entiteta, *atribut* je "*atomski*", što znači da se ne može dalje strukturirati kao entitet, budući da se predstavlja kao jednina imeničke fraze. Za svaki atribut može se dodijeliti određena *domena* (aka. *tip podataka*), označavajući "prirodu" dopuštenih vrijednosti. Najčešći su numerički (`NUMERIC`), znakovni (`VARCHAR`) i datum/vrijeme (`DATE/TIME`).

Osim karakterizacije entiteta, određeni atributi mogu poslužiti za *jedinstvenu identifikaciju svake instance* entiteta (npr. konkretnog zaposlenika ili konkretnog odjela). Takvi identifikacijski atributi djeluju kao takozvani "*Primarni ključ (PK)*". Za odjel, npr., neki umjetni ID može se definirati kao PK (koji bi trebao biti numerički, npr. "departmentID") ili alternativno kombinacijom postojećih atributa kao što su "departmentName" i "location", stvarajući takozvani " *kompozitni PK*". Preporučljivo je definirati PK za svaki entitet jer se inače, npr. zaposlenici ili odjeli koji imaju isti naziv, ne mogu razlikovati prilikom dohvaćanja podataka. Dakle, svaka vrijednost PK atributa mora biti jedinstvena i, naravno, mora postojati.

Konačno, treba spomenuti da osim domena i PK-ova, također se mogu navesti i druga takozvana *ograničenja* na vrijednosti podataka za određene atribute, npr. `NOT NULL`, tj. "obavezno" kako bi se zahtijevalo postojanje vrijednosti (npr. za atribut plaće).

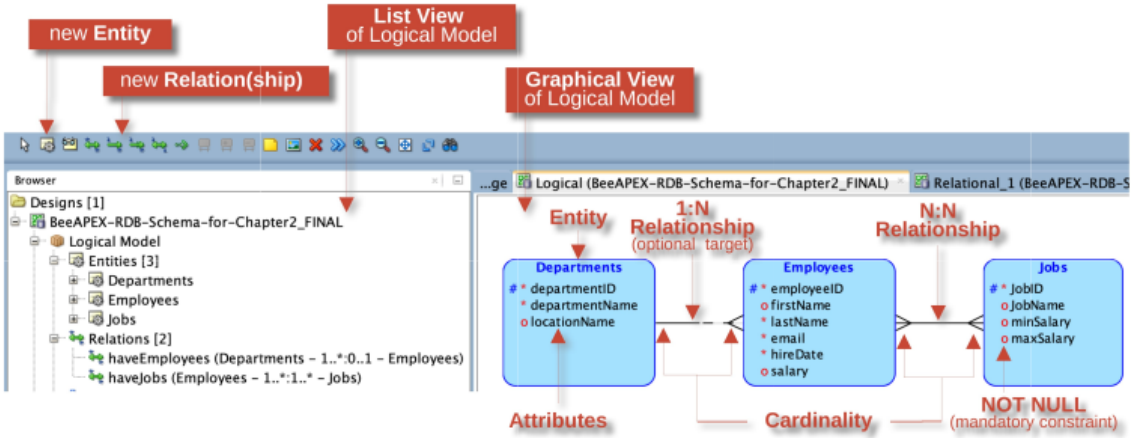
Veza. Konačno, *veza* povezuje (najčešće) dva različita entiteta, vizualno je prikazan linijom između pravokutnika entiteta i imenovan glagolskom frazom iznad crte. Postoje prije svega tri različite vrste veza koje se obično koriste u praksi, a razlikuju se po *broju instanci entiteta* kojima je dopušteno da budu dio veze na svakom kraju koji se naziva njegova "*kardinalnost*" (što predstavlja drugu vrstu ograničenja podataka koji se mogu navesti):

- **Jedan prema više ("1:N"):** Jedna instanca entiteta može biti povezana s mnogo instanci entiteta na strani "više" veze (npr. jedan odjel ima mnogo zaposlenika, tj. odnos "have-Employees") i entitet na strani "više" veze (zaposlenik) mogu biti povezani samo s jednim entitetom na strani "jedan" (odjel). Takav se odnos naziva "*Izvor-Cilj*" ili "*Roditelj-dijete*" *odnos*, izvorni entitet na "jednoj strani" djeluje kao "Roditelj" i proizvoljni broj ciljnih entiteta na strani "više" koji djeluju kao njegova "djeca". U gornjem primjeru, odjel je roditelj, a zaposlenici su entiteti-djeca.
- **Više-više ("M:N"):** Jedna instanca izvornog entiteta može biti povezana s *više instanci ciljnog entiteta*, a jedna instanca ciljnog entiteta može biti povezana s *više izvornih entiteta instance*. Primjer bi bili zaposlenici koji imaju poslove (tj. odnos "imaju poslove"), budući da zaposlenik može imati više od jednog posla tijekom vremena, a određeni posao vjerojatno obavljaju mnogi zaposlenici. Treba napomenuti da će odnos "M:N" uvijek proizvoditi dva odnosa "1:N" kako budemo napredovali u modeliranju baze podataka.
- **Jedan-na-jedan ("1:1"):** Konačno, jedna instanca entiteta također može biti povezana s *jednom instancom entiteta* na drugom kraju odnosa, samo i obrnuto (npr. osoba "živi na" određenoj adresi i ta je adresa adresa samo te određene osobe). Međutim, mora se primijetiti da se ovoj vrsti odnosa često daje prednost kompaktnijem predstavljanju samo unutar jednog entiteta (npr. osobe entiteta također pokrivaju atribute adrese kao što su "ulica" i "grad") i stoga se dalje ne razmatra u ovom poglavlju.
- **Opcionalnost:** Za svaku od ove tri različite vrste odnosa, također je moguće specificirati, na

svakom kraju za izvor i cilj, je li *postojanje najmanje 1 instance entiteta opcionalno ili nije* . Na primjer, ako se osniva novi odjel, moglo bi se dogoditi da još nema zaposlenih zaposlenika za taj odjel, tj. cilj odnosa "imati zaposlenika" mora biti definiran kao neobavezan. To je također prikazano isprekidanom linijom u ODM-u.

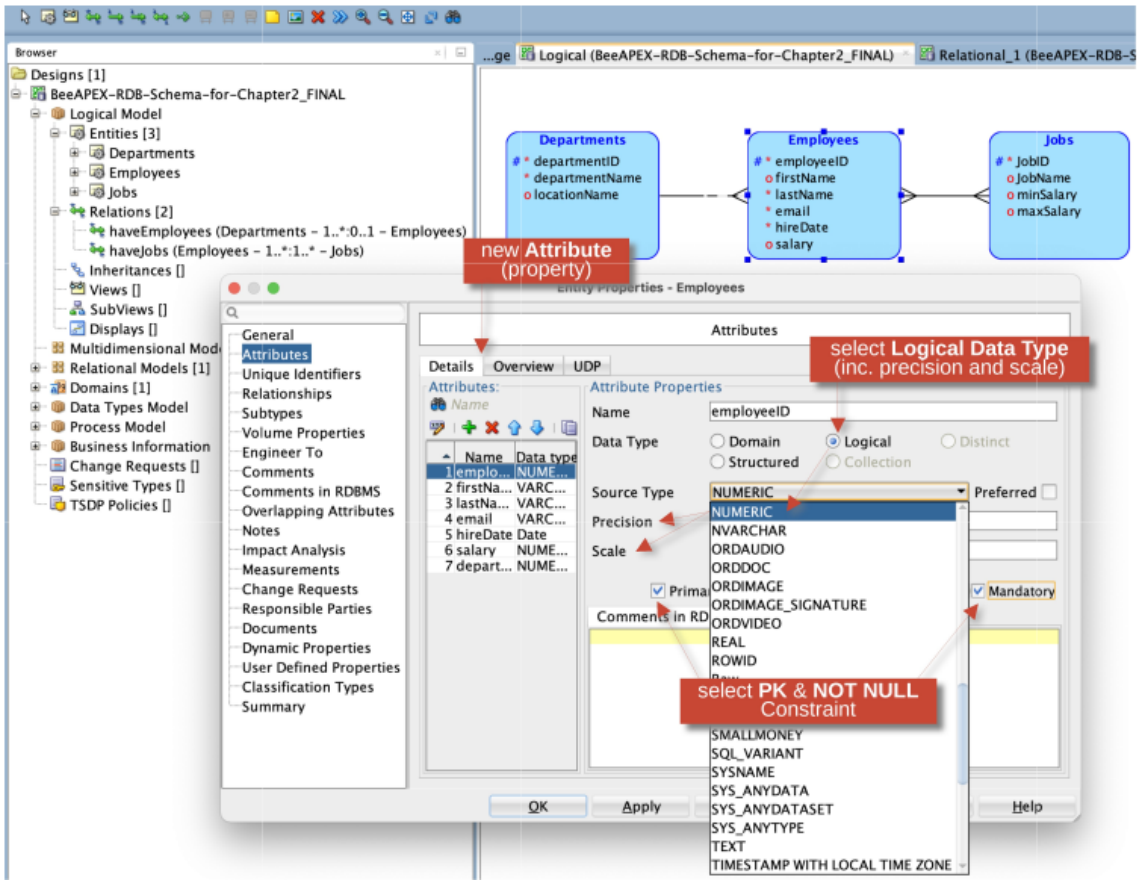
Veze se u ODM-u jednostavno nazivaju "relacije" koje se ne bi trebale brkati s "relacijama" "relacijskog modela" koje su zapravo sinonim za "tablice" (usp. Odjeljak 2.1.2). Stoga se dalje pridržavamo uobičajenijeg pojma "veze".

GUI-elementi ODM-a za upravljanje ER-dijagramima. Sljedeće slike daju prvi dojam o najvažnijim GUI elementima ODM-a. Prvo, slika 2.3 ilustrira kako definirati nove entitete i nove odnose, kao i vizualizaciju logičkog modela u obliku prikaza popisa i grafičkog prikaza. Unutar grafičkog prikaza označeni su najvažniji grafički zapisi i simboli.



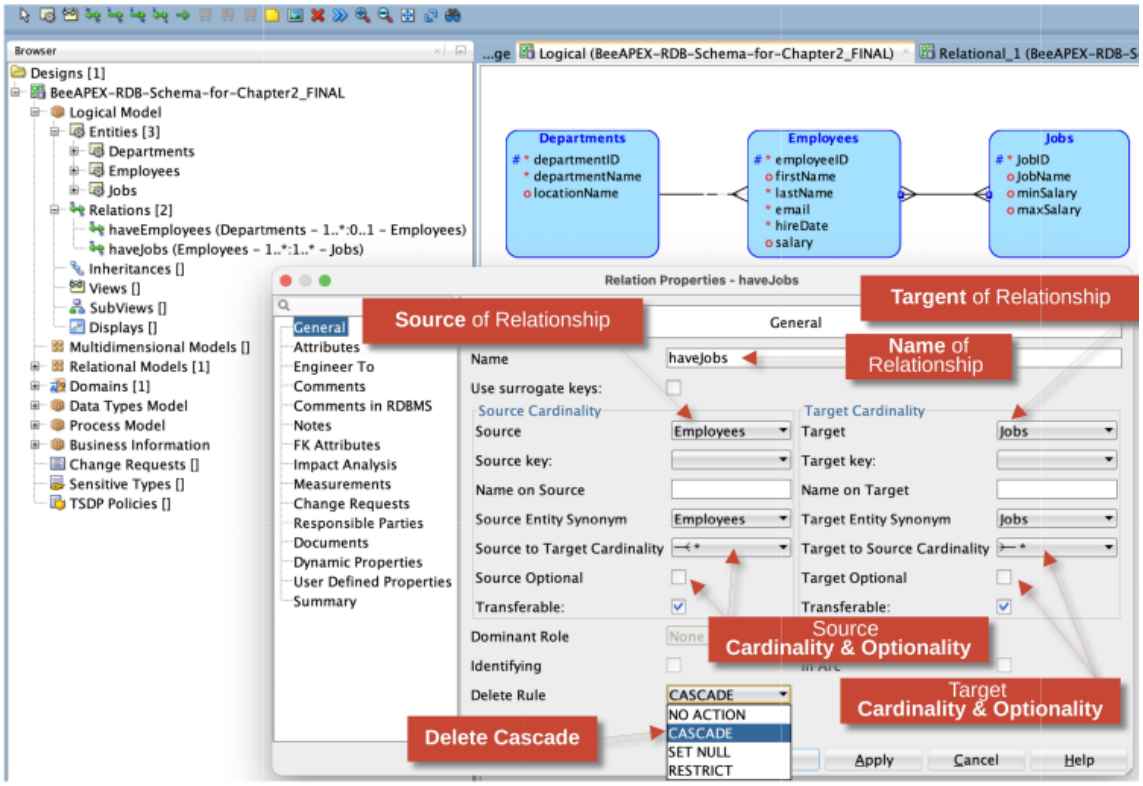
Slika 2.3: Upravljanje logičkim modelom u ODM-u

Slika 2.4 prikazuje dijalog za definiranje atributa koji se otvara jednostavnim dvostrukim klikom na određeni entitet.



Slika 2.4: Upravljanje atributima u ODM-u

Slika 2.5 prikazuje dijalog za definiranje odnosa, koji se može pokrenuti jednostavnim dvostrukim klikom na određeni redak, koji predstavlja vezu.

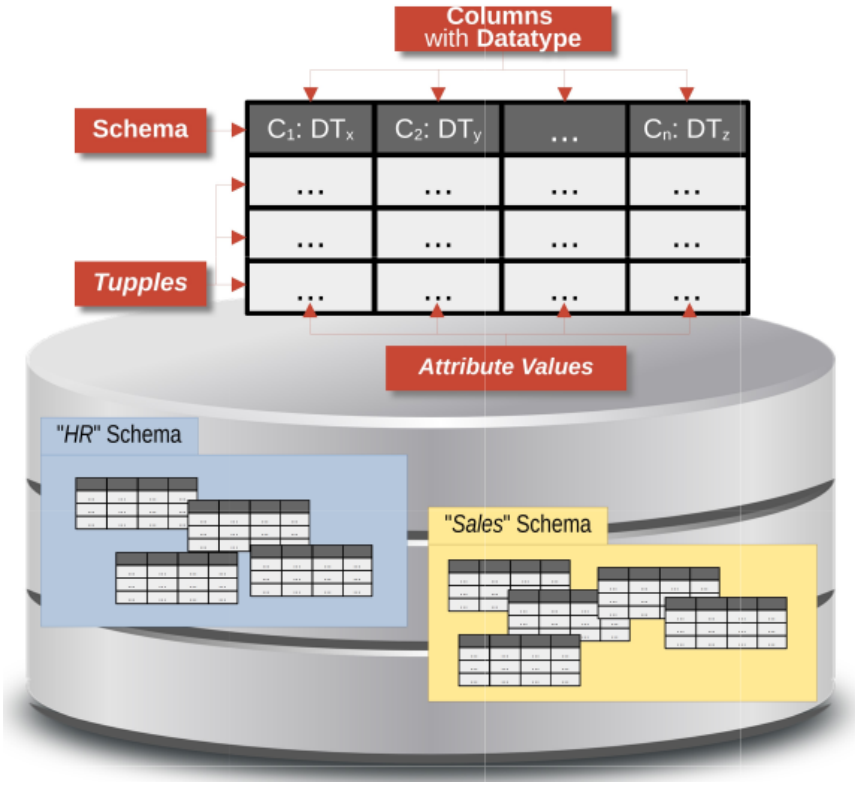


Slika 2.5: Upravljanje vezama u ODM-u

2.1.2 Relacijski model

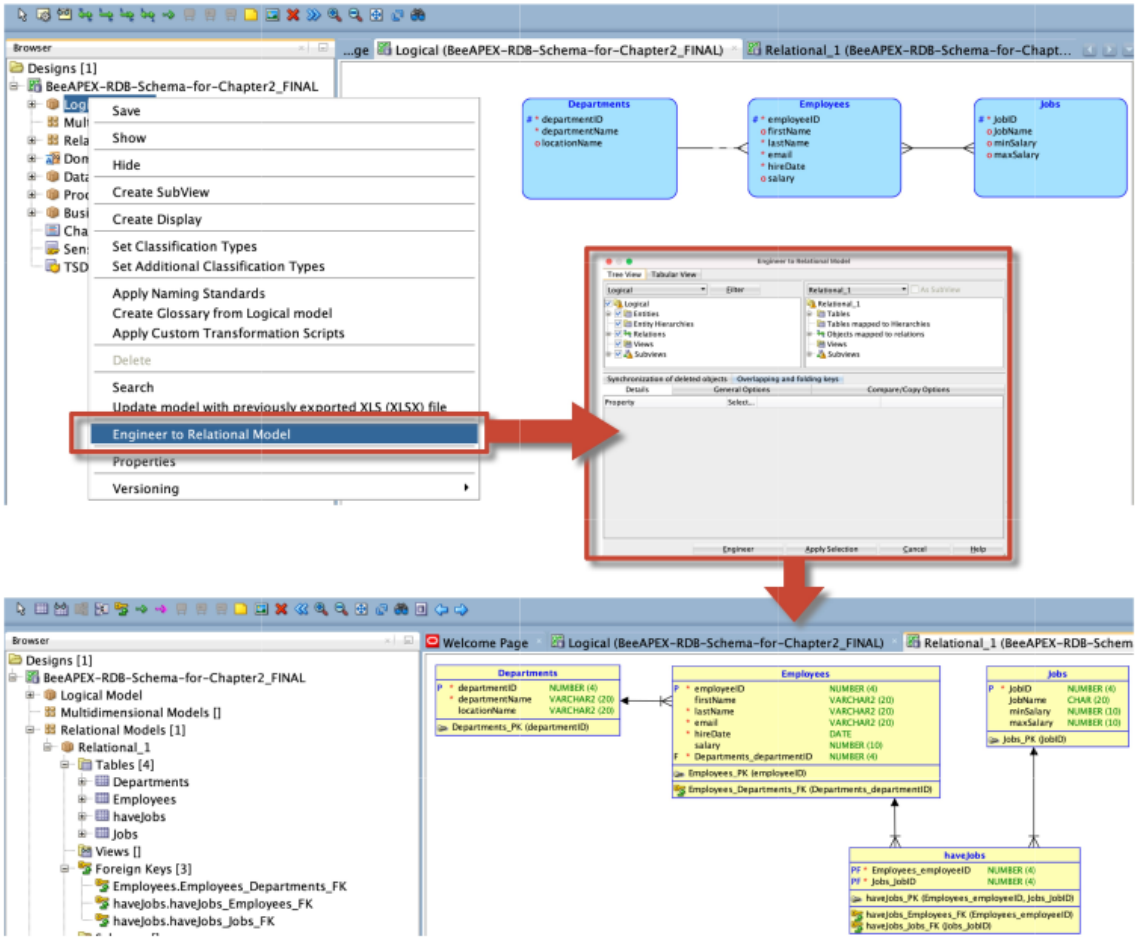
DB shema. Logički model u smislu ER-dijagrama koji je neovisan o određenom DBS-u osnova je za izvođenje DBS-specifične strukture DB-a koja se naziva " *DB-shema*". Unutar određenog DBS-a moguće je izraditi i pohraniti proizvoljan broj različitih shema za različite korisnike i/ili domene, npr. jednu za upravljanje HR podacima i drugu shemu za upravljanje podacima o prodaji proizvoda. Shema djeluje slično mapi u datotečnom sustavu dopuštajući grupiranje podataka za određenu svrhu.

Relacijska DB shema. Vrsta sheme specifična za DB je takozvana *Relacijska DB (RDB) shema* koja koristi *Relacijska Model (RM)* kao osnovni formalizam za opisivanje strukture podataka. RM je prilično jednostavan budući da predstavlja DB shemu kao kolekciju *relations* aka. *tablice*, koje nalikuju konceptu entiteta. Tablice se izvode iz entiteta na jednostavan način. Analogno entitetu, tablica se sastoji od naziva, *columns* (nalik na *atribute*), njihovih tipova podataka i daljnjih (neobaveznih) ograničenja. Sve tablice čine *RDB-shemu* (usp. sliku 2.6). Shema tablice je stalna i služi kako bi se pohranili stvarni podaci, dok se svaki red u tablici naziva "*torka*" koja se sastoji od zbirke povezanih podatkovnih vrijednosti pohranjenih unutar stupaca, čime predstavlja *instancu entiteta* iz stvarnog svijeta u smislu *relacijskih* (*tabularnih*) podataka.



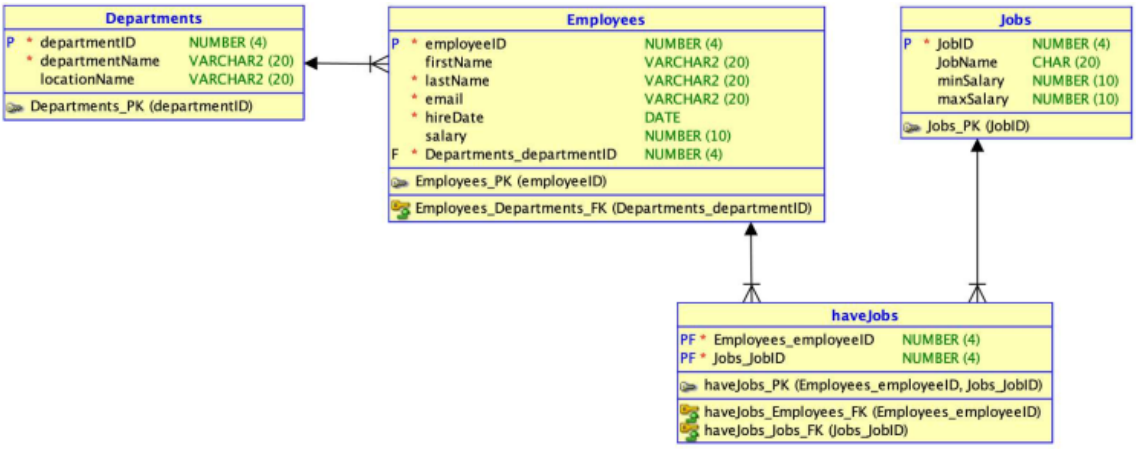
Slika 2.6: RDB-sheme i sheme tablica

Zanimljivo je primijetiti da *ODM* omogućuje automatsko generiranje RDB-sheme iz logičkog modela u smislu tablica, odražavajući tako entitete našeg logičkog modela i sve atribute i ograničenja svaki entitet. To se radi pomoću stavke izbornika *"Engineer to Relational Model"* kao što je prikazano na slici 2.7. Rezultirajuća RDB-shema također je grafički vizualizirana korištenjem neke vrste formalizma ER-dijagrama, temeljenog na malo drugačijim, više "tablično orijentiranim zapisima" od ER-dijagrama logičkog modela, pokazujući sada, npr., i podatke vrste svih atributa.



Slika 2.7: Proces za generiranje RDB-sheme iz logičkog modela

Sada se usredotočimo na naš primjer relacijskog modela prikazanog na slici 2.8.

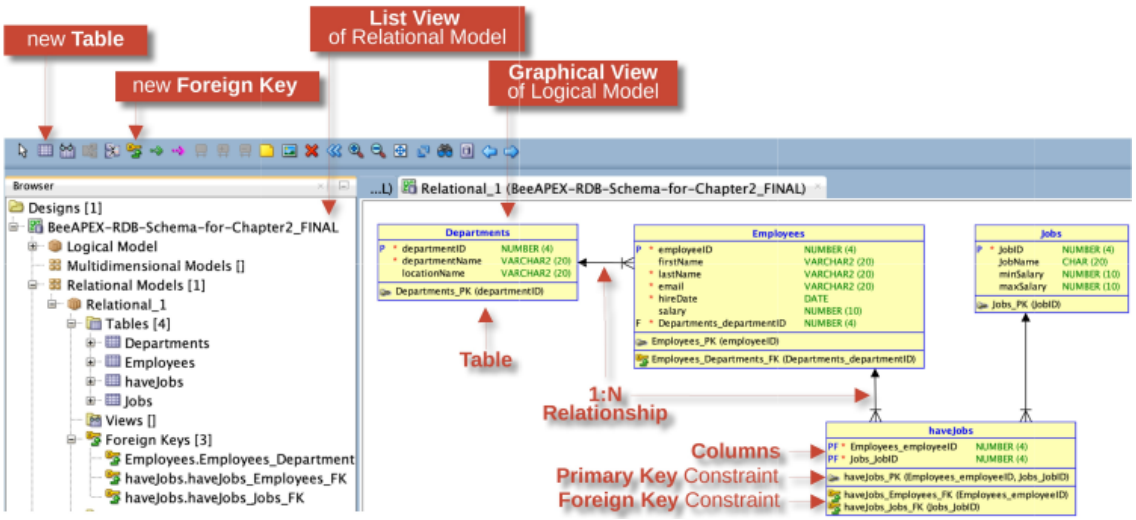


Slika 2.8: Automatski generirana RDB shema

Prije svega, tijekom procesa generiranja relacijskog modela iz logičkog modela, osim pretvaranja postojećih entiteta i atributa u tablice i odgovarajuće stupce, PK-ovi se automatski dodaju svakoj tablici i daju im naziv (usp. ., npr. “Odjeli_PK (ID odjela)”. Osim toga, odnosi između entiteta unutar logičkog modela automatski su predstavljeni unutar tablica, koristeći takozvani “*vanjski ključ (FK)*” *koncept* na sljedeći način:

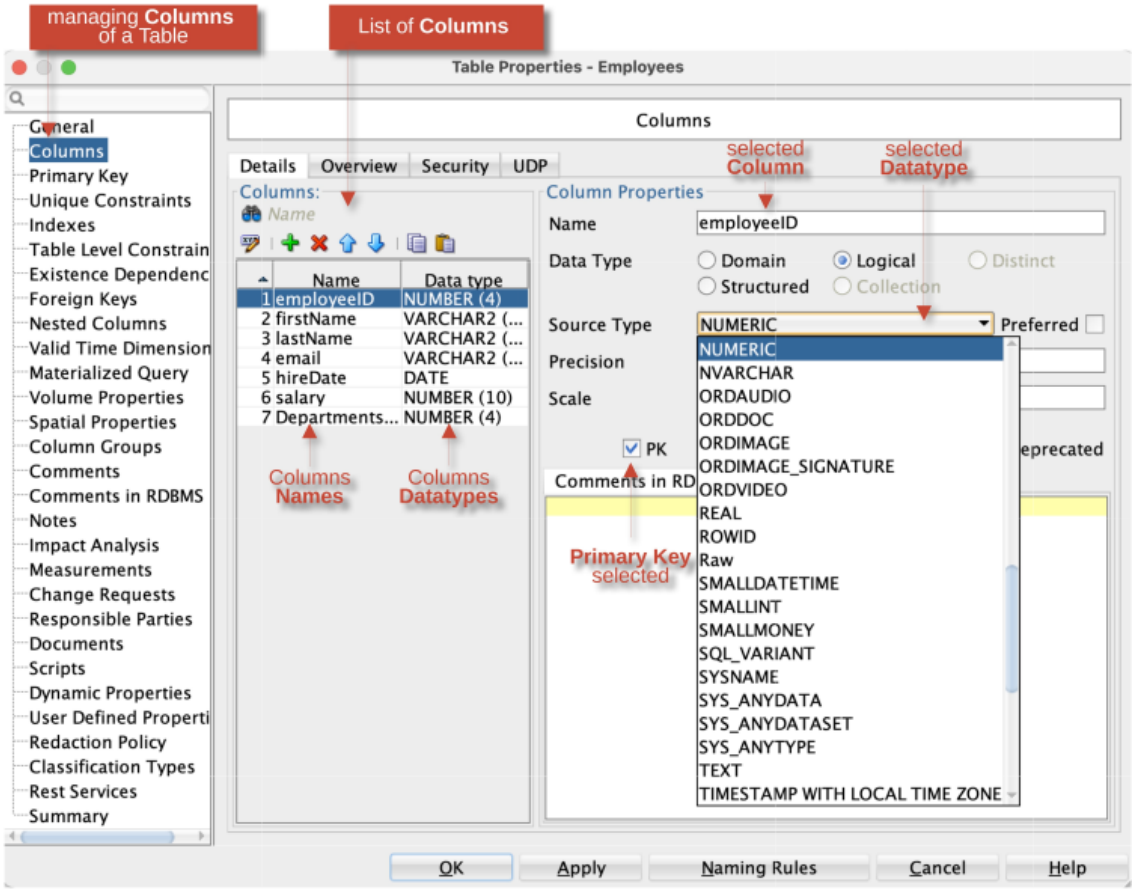
- **Odnos jedan-prema-više (1:N):** Unutar tablice na "više-strani" odnosa (tj. tablica Child), dodatni atribut je uveden i deklariran kao FK, djelujući kao neka vrsta reference na vrijednosti PK relacije na “One-side” (tj. tablica Parent). Ako postoji, npr., zaposlenik s PK ID 4711, tada unutar atributa FK pridruženog odjela mora biti prisutna i vrijednost 4711. Posljedično, FK atribut mora imati isti tip podataka kao PK. U našem primjeru automatski se generira FK "Odjeli_odjel ID", kao i FK pod nazivom "Zaposlenici_Odjeli_FK". Kao posljedica toga, svaki put kada se podacima unutar ova dva odnosa manipulira u DBS-u, FK osigurava *konzistentnost* između ovih odnosa, tj. ispravne podatke odnosa. Npr., ako je novi zaposlenik umetnut u tablicu Zaposlenici, DBS osigurava da je također dodijeljen postojećem odjelu. Ako pokušate izbrisati odjel, tada DBS odbija ovu operaciju prema zadanim postavkama u slučaju da još uvijek postoje povezani zaposlenici. Promjena ovog zadanog ponašanja moguća je korištenjem takozvane “*Cascade-Delete Option*”, kada se navodi FK, što rezultira automatskim brisanjem (možda “otpuštanjem”) svih zaposlenika povezanih s izbrisanim odjelom.
- **Odnos više-prema-više (M:N):** Odnos više-prema-više između dva entiteta ostvaruje se uvođenjem nove, treće tablice, koja djeluje kao neka vrsta “*tablice preslikavanja*” koji uglavnom pohranjuje reference na svaki od PK-ova pridruženih entiteta u smislu dva FK atributa. Što se tiče našeg tekućeg primjera zaposlenika i poslova, automatski se generira treća relacija pod nazivom "haveJobs", koja je sada odgovorna za upravljanje točnim vezama između zaposlenika i njihovih odgovarajućih poslova. PK ove tablice mapiranja je kompozitni, sastoji se od PK-ova dviju povezanih tablica, tj. "haveJobs_PK(Employees_employeeID, Jobs_jobID)". Svaki atribut ovog PK-a također se koristi za definiranje odgovarajućeg FK-a koji se odnosi na PK odgovarajuće povezane tablice (tj. "haveJobs_Employees_FK (Employees_employeeID)" i "haveJobs_Jobs_FK(Poslovi_ID posla)”). Treba napomenuti da ova tablica preslikavanja također može imati proizvoljan broj dodatnih atributa, koji dalje karakteriziraju odnos, npr. "datum početka" i "datum završetka" posla.
- **Opcionalnost:** Prema zadanim postavkama, za svaki od ovih odnosa, postojanje barem jednog izvora i jedne ciljne instance nije obavezno. Ako je obavezno, onda daljnje NOT NULL ograničenje mora biti povezano s odgovarajućim FK-atributom. Uzimajući u obzir naš primjer, sva tri FK trebaju NOT NULL ograničenje jer (i) Zaposlenici moraju biti povezani s određenim odjelom (ii) kao i s najmanje jednim poslom i (iii) posao mora biti povezan s najmanje jednim Zaposlenik.

GUI-elementi ODM-a za upravljanje RM-om. Sljedeće slike daju pregled najvažnijih GUI elemenata *ODM-a* za upravljanje RDB-shemom koji su prilično slični onima za upravljanje logičkim modelom. Prvo, slika 2.9 ilustrira kako definirati nove tablice i nove FK-ove, kao i vizualizaciju RM-a u obliku prikaza liste i grafičkog prikaza. Unutar grafičkog prikaza označeni su najvažniji grafički zapisi i simboli.



Slika 2.9: GUI-elementi za upravljanje stupcima RM-a

Slika 2.10 prikazuje dijalog za definiranje stupaca koji se može pokrenuti jednostavnim dvostrukim klikom na određenu tablicu. Imajte na umu da ovaj dijalog uvelike nalikuje dijalogu za upravljanje atributima kao što je već prikazano na slici 2.4.



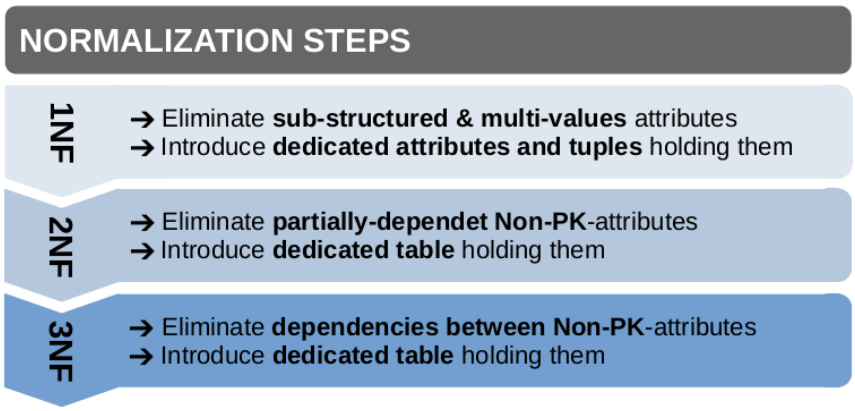
Slika 2.10: GUI-elementi za upravljanje RM-om

RDB-shema neovisna o aspektima fizičke pohrane. Iako je RDB-shema temeljena na relacijskom modelu već konkretnija i specifičnija za DBS od logičkog modela (npr. "implementacija"

odnosa temeljenih na FK-koncept) mora se primijetiti da još uvijek skriva složenost temeljnih mehanizama pohrane DBS-a (aka. "*fizička*" *razina*), npr. koliko se datoteka koristi u redoslijedu za pohranjivanje tablica ili na kojim poslužiteljima će se podaci pohraniti. Još jedna velika prednost ove apstrakcije od aspekata fizičke pohrane je da se nove RDB sheme mogu realizirati ili postojeće promijeniti bez brige o temeljnoj fizičkoj organizaciji podataka.

2.1.3 Normalizacija RDB-sheme

RDB-shema koja je izvedena iz logičke sheme često se može dodatno poboljšati kako bi se bolje postigli određeni *kriteriji kvalitete* koji prevladavaju u DBS-u, s jedne strane najprikladniji za *čitanje i pristup podacima* u smislu DB-upita, ali, s druge strane, pružajući i odgovarajuću osnovu za *manipulaciju podacima* u smislu umetanja, ažuriranja i brisanja podataka. Iako su to, u određenoj mjeri, proturječni ciljevi, postoji dobro poznati koncept dostupan za RDBS u smislu takozvane "*teorije normalizacije*" koja opisuje sustavni proces od tri koraka nazvan "*prvi normalni oblik*", "*drugi normalni oblik*" i "*treći normalni oblik*", što dovodi do poboljšanih RDB-shema (usp. sliku 2.11 za pregled) . U nastavku su ove tri normalne forme detaljnije opisane, koristeći male varijacije našeg tekućeg primjera.



Slika 2.11: Koraci normalizacije – pregled

Prva normalna forma. Prva normalna forma zahtijeva da svaki atribut tablice bude "atomski" što znači da nisu dopuštene ni (i) *podstrukture* (npr. ako bi, umjesto našeg jednostavnog atributa "locationName", alternativni atribut "location" sadržavao vrijednosti o ulici, poštanskom broju, gradu i državi) niti (ii) *multi-values* nisu dopušteni (npr. atribut "location" također može sadržavati nekoliko adresa određenog odjela. Podstrukture unutar vrijednosti mogu se *eliminirati* jednostavnim predstavljanjem podstrukture namjenskim atributima (npr., za svaki dio adrese) i više vrijednosti jednostavnim korištenjem vlastite n-torke za svaku od vrijednosti (tj. stavljanje svake adrese u zaseban red).

Jedna glavna *prednost* prve normalne forme je da upiti podataka mogu biti specifičniji, budući da se sada, npr., svakom dijelu adrese kao i svakoj adresnoj n-torci može zasebno pristupiti.

Druga normalna forma. Druga normalna forma zahtijeva (i) da tablica bude u *prvoj normalnoj formi* i (ii) u slučaju kompozitnog PK, tj., koji se sastoji od dva ili više atributa, sve ostale vrijednosti atributa koji nisu PK su *jedinstveno identificirane samo cijelim složenim PK*. Na primjer, pretpostavimo da tablica Departments sadrži složeni PK s atributima "departmentName" i "locationName". Ovaj PK ispravno određuje vrijednost atributa koji nije PK "annualRevenue", budući da prirodno pretpostavljamo da vrijednost "annualRevenue" ovisi i o odjelu i njegovoj lokaciji. Vrijednosti drugog atributa koji nije PK, "locationAddress", međutim, već bi ovisile o dijelu PK-a, naime o nazivu lokacije. Kako bismo *uklonili* ovu *djelomičnu ovisnost*, moramo *razdvojiti* tablicu Odjeli, izdvajajući informacije o lokaciji u zasebnu tablicu "Lokacije" i povezati

obje tablice preko FK.

Jedna glavna *prednost* druge normalne forme je da sprječava *redundantno pohranjivanje podataka*, budući da se u našem slučaju određena adresa pohranjuje samo jednom i mogu je ponovno koristiti različiti odjeli. Time se smanjuje ne samo prostor za pohranu, već prije svega i sprječava netočne podatke, takozvane "*anomalije*" ako se, npr., određene manipulacije podacima ne obrađuju na bilo kojem dupliciranom podatku (npr. u slučaju da se nekoliko odjela seli na drugu adresu, adresa se mora promijeniti samo jednom). Naposljetku, dobra stvar je da imajući tablice samo s nekompozitnim PK-ovima (npr., numerički ID za svaku tablicu), RDB-shema "automatski" ispunjava drugu normalnu formu, jer djelomične ovisnosti ne mogu postojati.

Treća normalna forma. Treća normalna forma zahtijeva (i) da je tablica u *drugoj normalnoj formi* i (ii) da svaki *atribut koji nije PK atribut jedinstveno identificira samo PK*, a ne bilo koji drugi atribut koji nije PK, što dovodi do *prijelaznih ovisnosti*. Na primjer, ako bi tablica Departments sadržavala osim PK EmploeeID i locationName također locationAddress, tada bi locationAdress već bio jedinstveno identificiran s locationName (ako pretpostavimo da je svaki locationName povezan s različitom adresom) što nije dopušteno trećom normalnom formom. Kako bi se eliminirala ova situacija, informacije o lokaciji moraju se izdvojiti u zasebnu tablicu Locations i povezati preko odgovarajućeg FK-a s tablicom Departments.

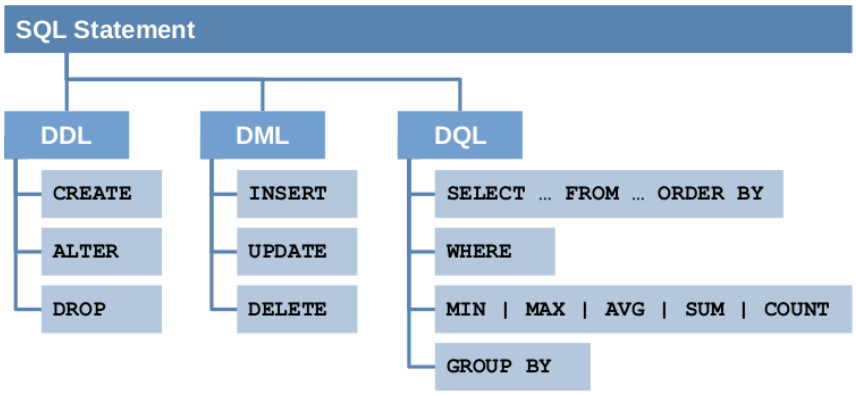
Korist je ista kao što je već spomenuto za drugu normalnu formu – smanjenje redundantnosti, a time i potencijalnih anomalija u slučaju manipulacije podacima.

2.2 Kako upravljati RDBS-shemom – SQL-DDL

U prethodnom odjeljku, RDB-shema je grafički modelirana na različitim razinama apstrakcije na *razvojni način niskog koda*. Sada je vrijeme za raspravu o tome kako se ovi grafički modeli mogu implementirati "programski" unutar RDBS-a tako da se stvarno mogu koristiti za upravljanje stvarnim podacima web aplikacije. Pitanja na koja će biti odgovoreno u nastavku su dakle:

- Kako savjetovati DBS da *kreira potrebne tablice*, idealno iz grafičkih modela i imamo li druge alternative ako takvi grafički modeli ne postoje?
- Kako izgleda *DBS "Programski jezik"* koji može kreirati tablice?

SQL = DDL | DML | DQL. Kako bismo prvo odgovorili na drugo pitanje, jezik koji se koristi za to, ali i za druge svrhe je *standardizirani jezik za strukturirane upite (SQL)*. SQL pruža širok raspon različitih *"DBS programskih upita"* koje se mogu, prema njihovoj namjeni, grupirati u različite kategorije (usp. sliku 2.12), a najvažniji su (i) upiti za upravljanje RDB-shemom (npr. stvaranje, mijenjanje i ispuštanje tablica), sažeto kao *DDL (Jezik za definiranje podataka)*, koji je fokus ovog odjeljka, (ii) upiti za manipuliranje podacima unutar shema (npr. umetanje, ažuriranje i brisanje podataka unutar tablica) nazvana *DML (Data Manipulation Language)* o kojoj će biti riječi u odjeljku 2.3 i na kraju (iii) naredbe za upite podataka nazvane *DQL (Data Query Language)*, o čemu se dalje govori u odjeljku 2.4.



Slika 2.12: Kategorije SQL naredbi

Pet opcija za kreiranje tablica. Prema alatima koje pruža ORACLE, postoji pet različitih opcija za kreiranje tablica, tj. implementacija RDB-sheme (vidi također sliku 2.45 za pregled):

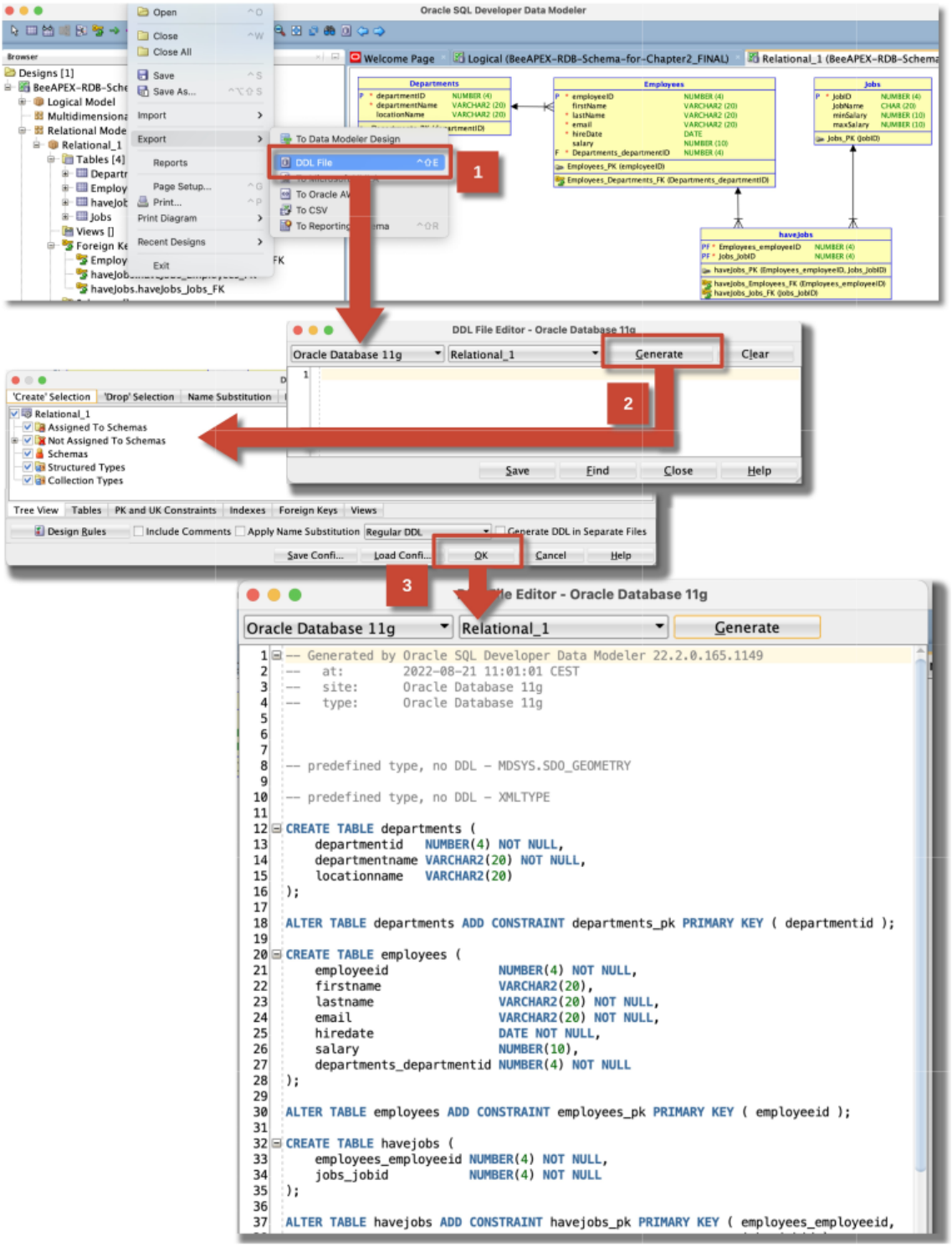
1. **Automatsko generiranje tablice pomoću ODM-a.** Ovo je opcija s najmanje napora, budući da se jednostavno pokreće *automatsko generiranje tablica* nekoliko klikova na gumb *koristeći ODM* bi bilo dovoljno (usp. Odjeljak 2.2.1).
2. **Automatsko generiranje tablice pomoću Quick SQL-a.** Druga bi alternativa bila korištenje neke vrste "skraćenog zapisa" za SQL, koji pruža ORACLE Quick SQL alat. Ovo je savršen izbor ako nema dostupnih grafičkih (logičkih i/ili relacijskih) modela RDB-sheme ili ako je potrebno neko brzo testiranje.
3. **Ručno kreiranje tablice pomoću SQL-DDL-a.** Naravno, postoji i mogućnost programskog specificiranja tablica korištenjem običnog SQL-a, pružajući prednosti da se svaki pojedinačni detalj koji se tiče specifikacije tablice može konfigurirati po potrebi i nije ovisno o, ponekad neoptimalnim procesima automatskog generiranja.
4. **Ručno kreiranje tablice pomoću ORACLE Object Browser (OB).** Za one korisnike koji nemaju SQL znanja, Object Browser omogućuje stvaranje novih tablica i njihovu promjenu ako je potrebno na način temeljen na obrascima.
5. **Automatsko kreiranje tablice pomoću ORACLE Data Workshopa.** Konačno, postoji još jedna mogućnost automatskog kreiranja tablice koja se, međutim, može koristiti samo ako već postoje neki podaci pohranjeni unutar vanjskih datoteka (npr. unutar EXCEL datoteka) korištenjem ORACLE Data Workshop. Ovom se opcijom dalje ne bavimo u ovom poglavlju, već je detaljnije opisana u 4. poglavlju.

U nastavku se detaljnije raspravlja o prve četiri opcije.

2.2.1 Automatsko generiranje tablice pomoću ODM-a

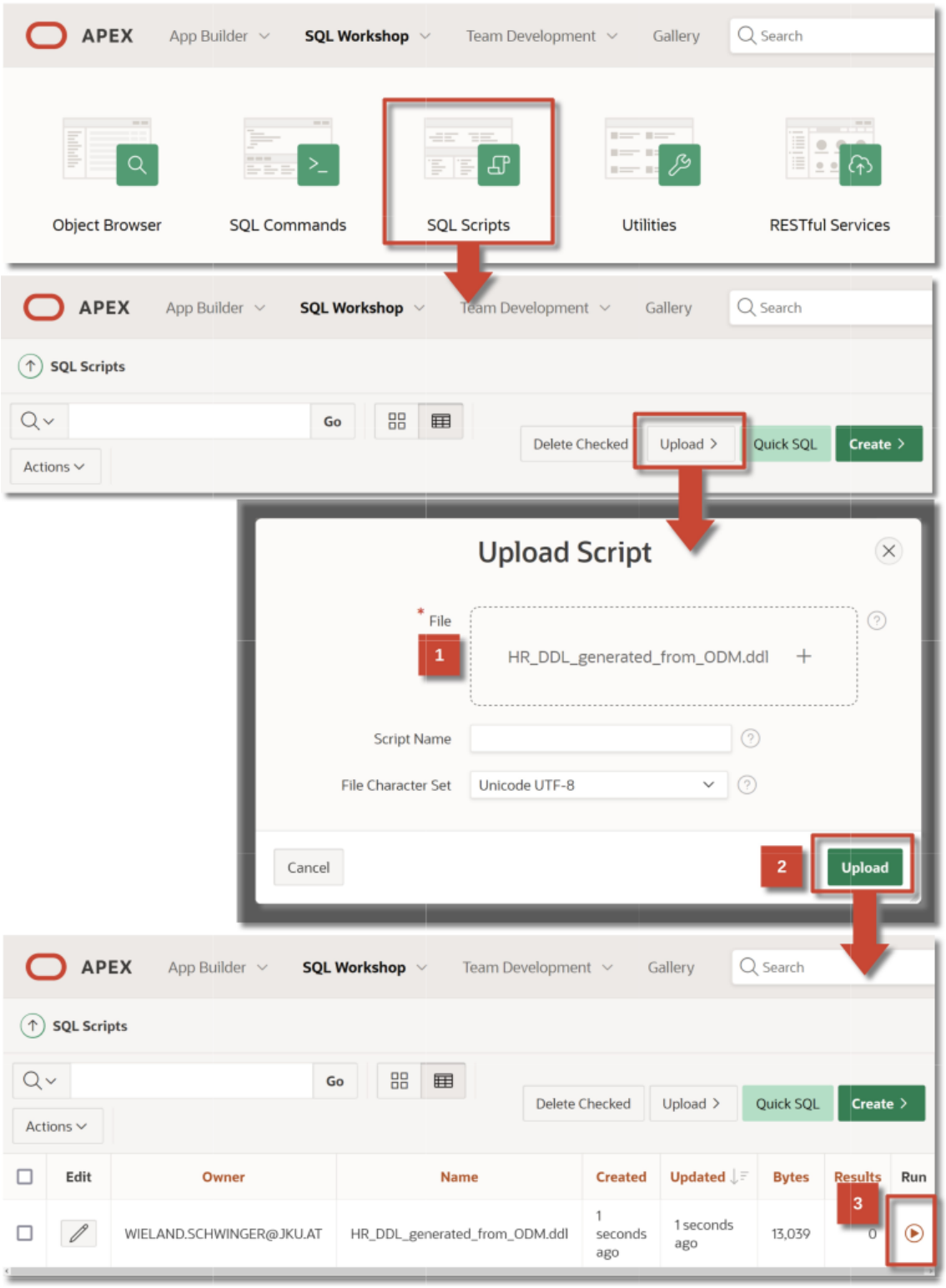
ODM je skraćenica od alata koji se zove Oracle SQL Developer Data Modeler. Budući da je to opcija generiranja tablice s najmanje napora, potrebna su samo dva koraka za stvaranje potrebnih DB tablica u slučaju da je RDB-shema razvijena u *ODM*.

Prvo, modelirana RDB-shema mora biti eksportirana u *SQL-skriptu* (usp. Slika 2.13, koraci 1 do 3), koja je zapravo jednostavna tekstualna datoteka koja sadrži sve SQL-DDL-naredbe potrebne za stvaranje tablica unutar RDBS-a. Imajte na umu da je sintaksa generiranih SQL naredbi detaljnije opisana kasnije.



Slika 2.13: Izvoz RDB-sheme u DDL-skriptu

Drugo, generirana SQL-skripta mora se izvršiti kako bi se uputilo RDBS da zapravo izgradi sve tablice kao prazne unutar svog prostora za pohranu. Za to, kao što je prikazano na slici 2.14, skripta se mora učitati putem ORACLE SQL Workshopa (usp. 1-2) i jednostavno izvršiti (usp. 3).



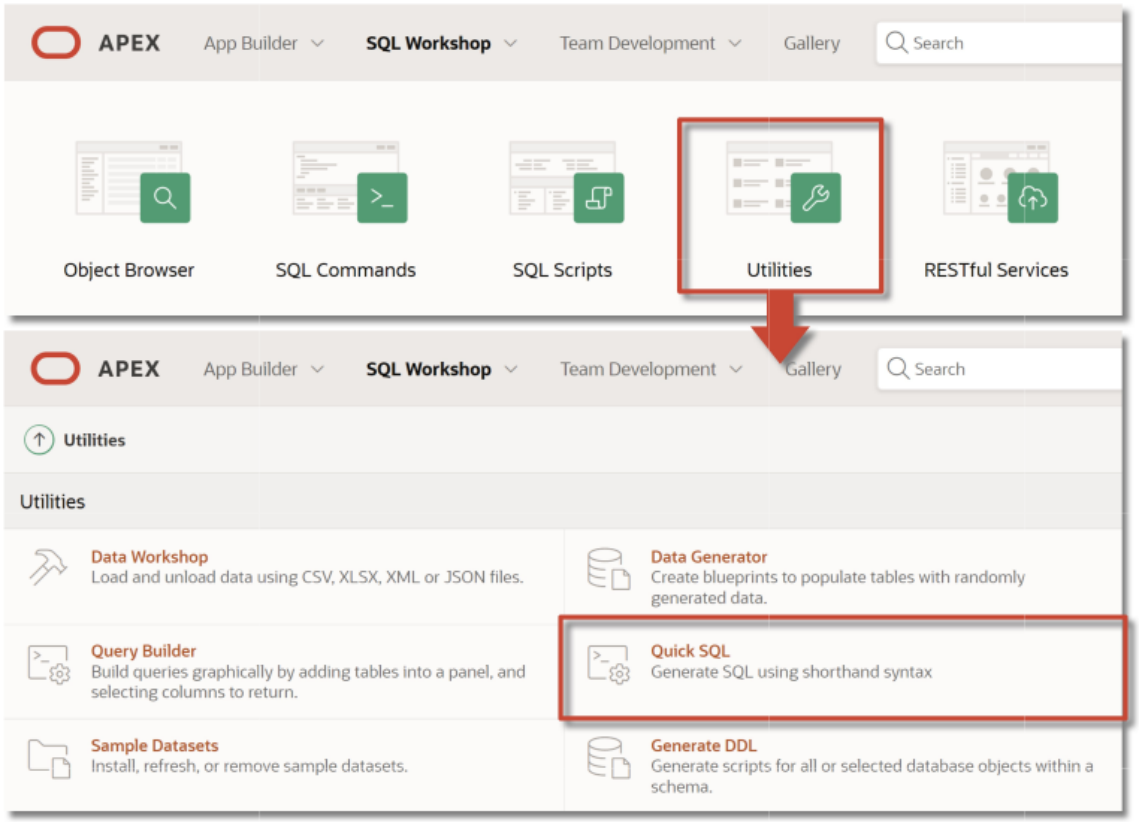
Slika 2.14: Prijenos i izvođenje SQL-skripte

2.2.2 Automatsko generiranje tablice pomoću Quick SQL-a

Prilikom izgradnje DB-sloja web aplikacije, vrlo je preporučljivo slijediti različite razine apstrakcije u smislu logičkih i relacijskih modela. Unatoč tome, *Quick SQL* pruža jednostavnu i intuitivnu alternativu za kreiranje tablica, dosta jednostavniju od SQL-a, npr. za potrebe brzog testiranja. Quick SQL dalje omogućuje automatsko generiranje odgovarajuće SQL-skripte. Nakon što se

generira SQL skripta, može se dotjerivati i proširivati i konačno izvršiti za stvaranje tablica. Ponovno se mora naglasiti da, iako je Quick SQL osmišljen kako bi smanjio napor potreban za izradu tablica, definitivno nije zamišljen da bude zamjena za modeliranje podataka (usp. Odjeljak 2.1).

Uređivaču ORACLE Quick SQL može se pristupiti putem "Utilities" programa SQL Workshop (usp. Slika 2.15).



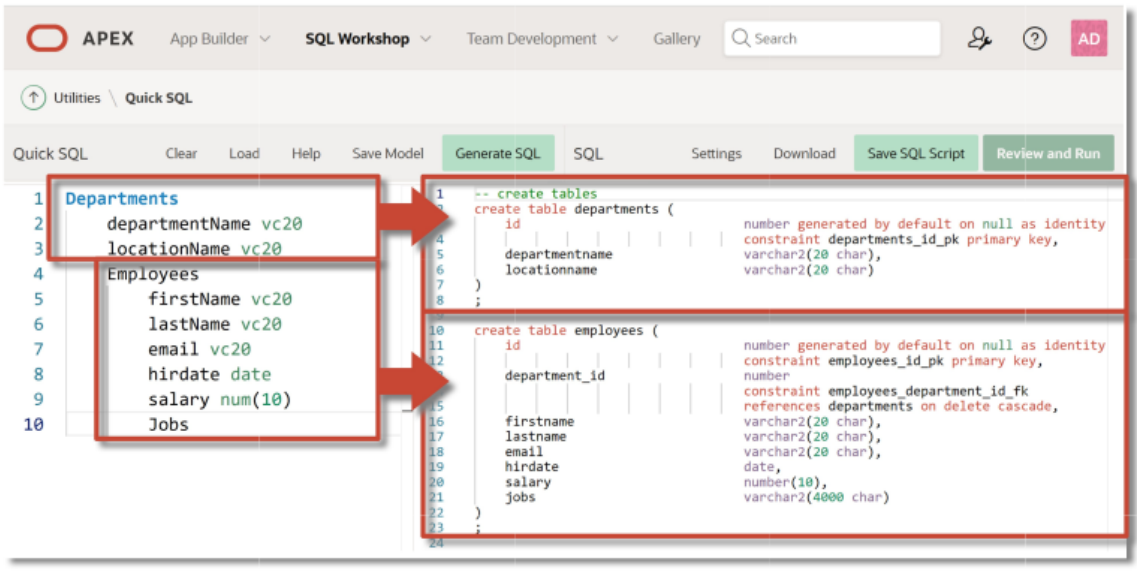
Slika 2.15: Pristup Quick SQL-u putem SQL Workshopa

Držimo se sada jednostavne sintakse koju nudi Quick SQL, koja u osnovi koristi nekoliko principa oblikovanja kako bi odredila potrebne tablica za generiranje:

- **Nadređene tablice:** Imena nadređenih tablica unose se bez ikakve uvlake. Nazivi tablica automatski se formatiraju, zamjenjujući sve razmake podvlakama.
- **Atributi:** Nazivi atributa unose se s ravnomjernim uvlačenjem od dva ili više razmaka. Kao i kod naziva tablica, nazivi atributa automatski se formatiraju, zamjenjujući sve razmake podvlakama.
- **Odnosi roditelj/dijete:** Odnosi roditelja dijete unose se uvlačenjem podređenih tablica ispod nadređenih tablica. Podređene tablice trebaju biti na istoj razini kao i stupci u nadređenoj tablici.
- **Vrste podataka:** Na temelju engleskog teksta sadržanog u nazivu stupca, i u odsutnosti bilo koje specificirane vrste podataka, RDBS automatski izvodi odgovarajuće tipove podataka. Ako stupac treba imati namjenski tip podataka, mogući su sljedeći: `NUM`, `INT`, `VC` (za `VARCHAR2`) ili `DATE`. Tip podataka može se jednostavno navesti na kraju naziva stupca, odvojen razmakom. Ako treba definirati određenu dužinu `VC`, mora se unijeti `VCn`, gdje je `n` duljina `VARCHAR2`. Općenito, dostupna sintaksa prikazana je putem izbornika "Pomoć".

Slika 2.16 prikazuje primjer Quick SQL notacije, prikazujući našu nadređenu tablicu “Odjeli” i odgovarajuću podređenu tablicu “Zaposlenici na lijevoj strani i automatski generiranu SQL skriptu u Okno SQL izlaza. SQL se odmah generira nakon svakog povratka reda. Konačna SQL skripta

može se spremiti za daljnju upotrebu i eventualnu prilagodbu, kao i izvršiti za konačno generiranje praznih tablica.

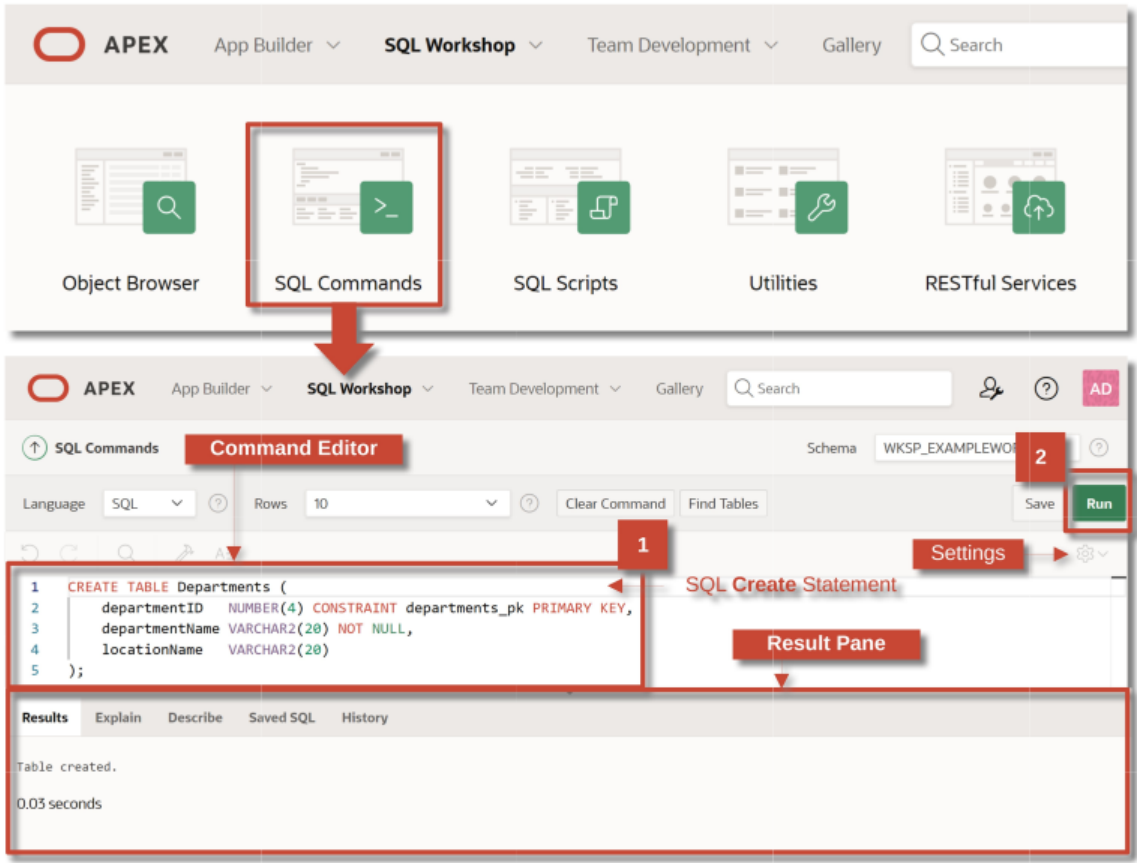


Slika 2.16: Generirana SQL skripta na temelju Quick SQL-a

Quick SQL također ima sintaksu za definiranje ograničenja (vanjski ključ i dopuštene vrijednosti) za određena podatkovna polja kao i za generiranje testnih podataka.

2.2.3 Ručno kreiranje tablice pomoću SQL-DDL

Programski način za stvaranje tablica je odustajanje od bilo kakvog automatskog generiranja SQL skripti "niskog koda" (kao što je objašnjeno u Odjeljku 2.2.1 i Odjeljku 2.2.2) i sami obavite posao. Da budemo konkretniji, možemo naravno i ručno specificirati SQL-skripte koristeći SQL-DDL `CREATE TABLE` naredbu unutar SQL-Command-Editora koji pruža ORACLE *SQL-Workshop* (usp. Slika 2.17). Nakon ručnog definiranja SQL-izjava korištenjem običnog SQL-DDL-a (1), rezultirajuća SQL-skripta može se spremiti i izvršiti (2) kako bi se stvorile stvarne prazne tablice (usp. Slika 2.17).



Slika 2.17: SQL Command Editor

SQL naredbe su manje-više jednostavne poput običnog engleskog, ali sa specifičnom sintaksom. `CREATE TABLE` izjava sastoji se od *SQL specifičnih ključnih riječi* kao što je sama "`CREATE`" kao i od *ograničenja definicija* (tipovi podataka, primarni ključ i vanjski ključ). Na slici 2.18 sve ključne riječi kao i drugi sintaktički simboli specifični za SQL (poput zagrada i zareza) napisani su velikim slovima i obojeni plavom bojom. Važno je napomenuti da su SQL ključne riječi neosjetljive na velika i mala slova, što znači npr. da "`CREATE`" znači isto što i "`create`". Definicija atributa je zatvorena unutar zagrada, atributi su odvojeni zarezima, svaki je idealno definiran unutar zasebnog retka (iako su to samo neformalne konvencije oblikovanja, a ne dio SQL sintakse). Ograničenja se mogu definirati korištenjem eksplicitnog naziva (npr. Departments_pk), koji nije obavezan (usp. npr. `NOT NULL` za DepartmentName), ali služi samo u svrhu bolje čitljivosti. Svaka SQL-naredba je na kraju omeđena točkom i zarezom.

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

Slika 2.18: SQL naredba za kreiranje tablice `Departments`

Uz naredbu `CREATE TABLE`, tu je naravno i SQL naredba za brisanje tablice – radi ispravno, bez obzira je li tablica prazna ili već sadrži neke podatke, koji se zatim također brišu (usp. sliku 2.19).

```
DROP TABLE Departments;
```

Slika 2.19: SQL naredba za brisanje tablice `Departments`

I konačno, definicija već postojećih tablica također se može promijeniti, npr. dodavanjem dodatnih atributa ili promjenom vrste podataka (usp. sliku 2.20):

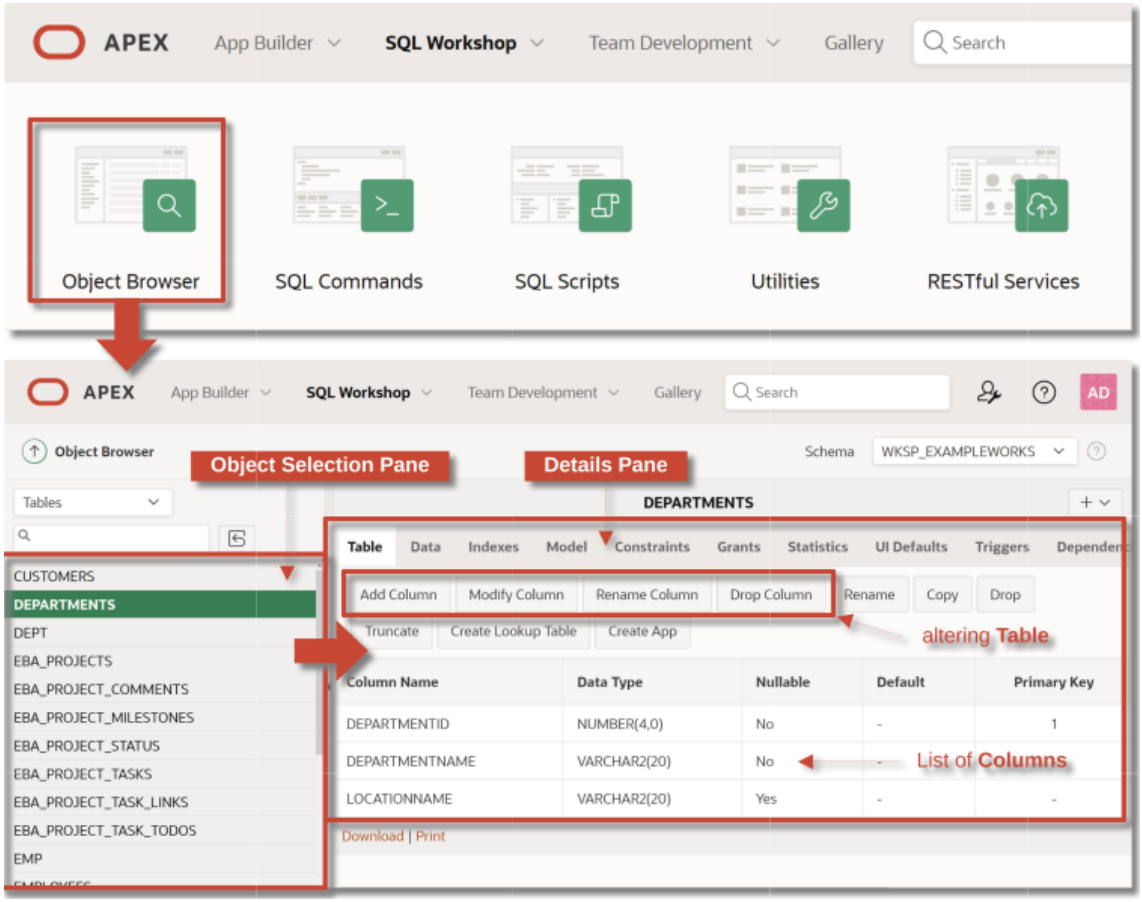
```
ALTER TABLE Departments
      ADD annualRevenue NUMBER(10);
```

Slika 2.20: SQL naredba za promjenu tablice `Departments`

Ako tablica koju treba promijeniti već sadrži podatke, tada su izmjene donekle ograničene, budući da npr. promjena vrste podataka može dovesti do nenamjernih učinaka na te podatke. Ovo pitanje, međutim, nije dalje obrađeno u ovom odjeljku.

2.2.4 Ručno upravljanje tablicom pomoću preglednika objekata

Preglednik objekata u SQL Workshopu omogućuje ne samo pregled svih generiranih tablica, takozvanih *"Objekata"* u ORACLE-ovoj terminologiji, zajedno s njihovim odgovarajućim atributima i ograničenjima, već i njihovo mijenjanje i stvaranje novih objekata, sve temeljene na jednostavnom GUI-čarobnjaku (usp. slika 2.21).



Slika 2.21: Upravljanje tablicama pomoću ORACLE-ovog Object Browsera

2.3 Kako manipulirati podacima – SQL-DML

Prazne tablice kreirane u prethodnom odjeljku, sada "čekaju" da se popune odgovarajućim podacima, koji se mogu dalje modificirati, a eventualno i brisati. To je upravo odgovornost SQL-DML iskaza `INSERT`, `UPDATE` i `DELETE`, koji će biti ukratko predstavljeni u sljedećem odjeljku 2.3.1 preko našeg primjera. Osim toga, postoji i alternativa bez kodiranja dostupna putem ORACLE-ovog preglednika objekata ("OB"), koja omogućuje ručno umetanje podataka u tablice, što je prikladno za korisnike bez ikakvog znanja SQL-a. Konačno, prilično ograničenu alternativu za umetanje podataka pruža Quick SQL (usp. odjeljak 2.3.3). Sve te alternative također su prikazane na slici 2.45.

2.3.1 Manipulacija podacima pomoću SQL-DML

Za programsko rukovanje podacima pomoću SQL-DML-a koristi se SQL Command Editor (vidi sliku 2.17), isti koji je korišten za izradu tablica. U nastavku su ilustrirani neki primjeri kako bi se dobio prvi dojam o sintaksi i funkcionalnosti koju pruža SQL-DML.

`INSERT` koristi se za *dodavanje novih redaka* u tablicu kao što se može vidjeti u sljedećem primjeru (usp. Slika 2.22):

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

Slika 2.22: SQL naredba za umetanje novih podataka u tablicu `Departments`

Treba napomenuti da osim ove programske opcije koju smo spomenuli, u Odjeljku 2.3.3 također postoji mogućnost korištenja Quick SQL-a za umetanje podataka, koji je međutim ograničen samo na umetanje nasumičnih podataka.

`UPDATE` koristi se za ažuriranje jednog ili više redaka podataka unutar tablice. Sljedeći primjer je najjednostavniji oblik izjave `UPDATE`, koja povećava plaću svih zaposlenika za 10% (usp. sliku 2.23):

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

Slika 2.23: SQL naredba za ažuriranje plaće svih zaposlenika za 10% u tablici `Employees`

Malo proširenje prikazano je u sljedećem primjeru, dodavanjem takozvane "`WHERE`-klauzule" kako bi se primijenilo povećanje plaće na zaposlenika koji ima "Miller" kao prezime (usp. sliku 2.24):

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

Slika 2.24: SQL naredba za ažuriranje plaće samo zaposlenika "Miller"

Konačno, također se više od jednog atributa tablice može ažurirati pomoću jedne naredbe kao što se može vidjeti u sljedećem primjeru (usp. Slika 2.25):

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

Slika 2.25: SQL izjava za ažuriranje više od jednog atributa u tablici `Employees`.

`DELETE` se koristi za uklanjanje jednog ili više redaka podataka iz tablice. Sljedeći primjer je najjednostavniji oblik naredbe `DELETE`, kojom se brišu svi podaci pohranjeni u tablici koja pohranjuje zaposlenike (usp. sliku 2.26).

```
DELETE FROM Employees;
```

Slika 2.26: SQL naredba za brisanje svih podataka iz tablice `Employees`

Proširenje prethodne izjave s "`WHERE`-klauzulom" omogućuje selektivno brisanje zaposlenika s ID-om posla "4711" (usp. sliku 2.27).

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

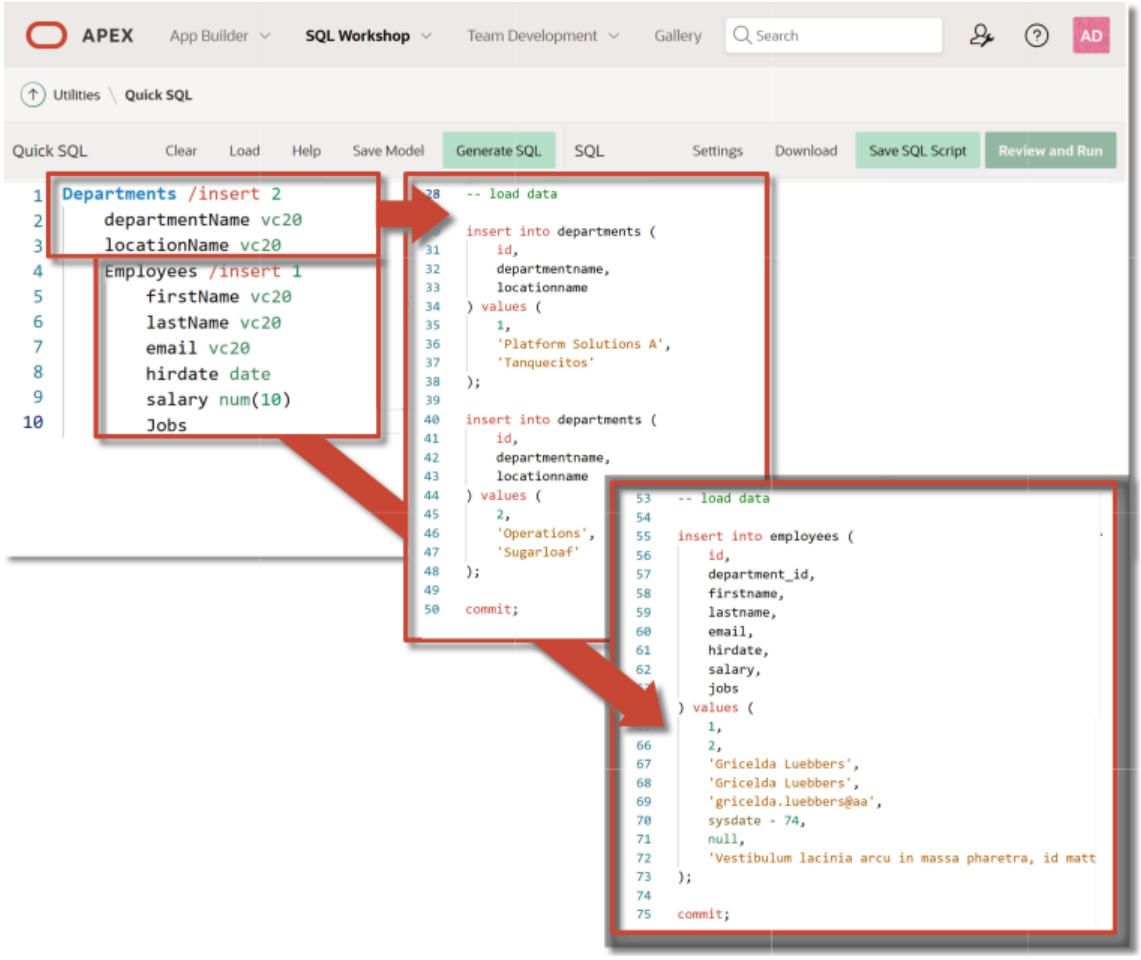
Slika 2.27: SQL naredba za brisanje nekih podataka iz tablice `Zaposlenici`

2.3.2 Manipulacija podacima korištenjem ORACLE-ovog preglednika objekata ("OB")

Osim ove programske opcije za manipuliranje podacima, također bi bilo moguće ručno umetati podatke u tablice, kao i mijenjati ili brisati postojeće podatke korištenjem uređivača koji pruža *Object Browser*, usp. Odjeljak 2.2.4, slika 2.21, korištenjem okna "Podaci". Mora se primijetiti da sve manipulacije podacima koje se rade pomoću OB-a opet automatski prevode ORACLE u odgovarajuće SQL-DML izjave.

2.3.3 Manipulacija podacima pomoću Quick SQL

Pomoću Quick SQL-a moguće je automatski unositi neke podatke u tablice. Primjer je prikazan na slici 2.28 korištenjem izjava "`/insert 2`" i "`/insert 1`". Ovim naredbama se definiraju brojčane n-torke sa slučajnim vrijednostima koje treba umetnuti u tablice.



Slika 2.28: Nasumično umetanje podataka pomoću Quick SQL-a

2.4 Kako postaviti upit za podatke – SQL-DQL

Nakon rasprave o različitim mogućnostima stvaranja praznih tablica u odjeljku 2.2 i opcijama za manipuliranje podacima u odjeljku 2.3, sada ćemo se okrenuti dohvaćanju podataka iz tablica tako što ćete prvo raspravljati o mogućnostima upita SQL-a u smislu izjave "SELECT" (usp. u odjeljku 2.4.1), opet pomoću našeg primjera kadrovske službe, nakon čega slijedi kratki pregled o ORACLE alatu Query Builder ("QB") koji pruža alternativu upitima s "niskim kodom" za korisnike koji imaju malo ili nimalo znanja o SQL-u (usp. Odjeljak 2.4.2). Ove su alternative također prikazane na slici 2.45.

2.4.1 Upiti podataka pomoću SQL-DQL

Prije svega, treba napomenuti da je izražajnost SQL-DQL-a ogromna, tako da ćemo se usredotočiti samo na vrlo mali podskup koncepata. Unatoč tome, cilj nam je dati prvi dojam izražajnosti i široke primjenjivosti SQL-DQL-a, uvođenjem različitih mogućnosti upita putem primjera, poredanih u nastavku prema rastućoj složenosti. Što se tiče alata, opet se koristi SQL uređivač naredbi (usp. sliku 2.17) isti koji je korišten za izradu tablica i manipuliranje podacima.

(1) Počnimo s najjednostavnijim upitom – *"Dohvati sve zaposlenike"* (usp. sliku 2.29).

```
SELECT * FROM Employees;
```

Slika 2.29: SQL izjava za dohvaćanje svih podataka iz tablice `Employees`.

Ciljna tablica upita (`Zaposlenici`) navedena je unutar klauzule `FROM`, traženi rezultat upita definiran je odmah nakon `SELECT`-clause - u ovom primjeru korištenje zvjezdice kao joker znaka – što znači da bismo htjeli dobiti vrijednosti svih stupaca tablice.

Izvadak rezultata ovog upita izgleda ovako (usp. slika 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

Slika 2.30: Izvadak iz Dohvaćanja svih podataka iz tablice `EMPLOYEES`.

Ako želimo poredati rezultat, npr. prema plaćama zaposlenika, moramo jednostavno dodati `ORDER BY`-klauzulu (usp. sliku 2.31):

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

Slika 2.31: SQL naredba za dohvaćanje svih podataka iz tablice `eMPLOYEES` poredanih prema njihovoj plaći.

Rezultat ovog upita izgleda ovako (usp. sliku 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

Slika 2.32: Izvadak iz dohvaćanja svih podataka iz tablice `Employees` poredanih prema njihovoj plaći.

(2) Što je s upitima samo za određene stupce i određene retke – *"Dohvati samo ime, adresu i plaću zaposlenika koji imaju plaću veću od 5.000"*

Prvo, kako bismo vratili vrijednosti samo određenih atributa, samo ih moramo navesti umjesto zvjezdice, odvojene zarezima. Drugo, da bismo selektivno dohvatili samo određene retke, moramo uvesti `WHERE`-klauzulu i specificirati logički uvjet na odgovarajući atribut (plaća u našem slučaju). Imajte na umu da može postojati nekoliko takvih uvjeta na različitim atributima, odvojenih ključnim riječima `AND` / `OR` (usp. Slika 2.33).

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

Slika 2.33: SQL upit samo za dohvaćanje određenih podataka iz tablice `Employees`.

Rezultat ovog upita izgleda ovako (usp. sliku 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

Slika 2.34: Izvadak iz dohvaćanje određenih podataka samo iz tablice `Employees`.

Ako želimo dobiti vrijednosti atributa "`hireDate`" u drugom formatu, može se koristiti`TO_CHAR()`-funkcija" koje pruža ORACLE SQL. Ova funkcija uzima vrijednost `DATUM` kao prvi unos, a željeni format kao drugi unos. U nastavku je prikazana proširena verzija gornjeg primjera (usp. sliku 2.35), sada koristeći `TO_CHAR()`-funkciju za " `hiredate`" (potpuni popis mogućih opcija oblikovanja je, naravno, dostupan u Oracle dokumentaciji):

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

Slika 2.35: SQL upit za dohvaćanje određenih podataka iz tablice `Employees` samo pomoću `TO_CHAR()`-funkcije.

Rezultat ovog upita izgleda ovako (usp. sliku 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

Slika 2.36: Izvadak iz dohvaćanja određenih podataka iz tablice `Employees` samo pomoću `TO_CHAR()`-funkcije.

(3) Usredotočimo se na postavljanje upita za nekoliko tablica – "*Dohvati sve zaposlenike i njihove odjele*" Dohvaćanje podataka dviju ili više tablica unutar jedne `SELECT`-naredbe zahtijeva prilično istaknut SQL-koncept, tj. "*Join*". Spajanje zapravo kombinira retke iz dvije ili više tablica koje moraju biti navedene u klauzuli `FROM`. Dodatno, potrebna nam je klauzula `WHERE` koja igra ulogu *proizvođača* između ove dvije tablice – budući da moramo nekako definirati da DBS treba, za svakog zaposlenika u **T**ablicu `Zaposlenici`, uzeti njenu vrijednost Departments_ departmentID (što je FK), ići dalje do tablice `Departments`, povezati ovu vrijednost s odgovarajućom PK vrijednošću DepartmentID-a i vratiti podatke odjela zajedno s podacima zaposlenika. Formalnije, ovo spajanje određeno je takozvanim "*join-condition*" kao što se može vidjeti u sljedećem primjeru (usp. Slika 2.37):

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

Slika 2.37: SQL naredba za spajanje podataka iz tablice `Employees` i tablice `Departments`.

Rezultat ovog upita izgleda ovako (usp. sliku 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

Slika 2.38: Izvadak iz dohvaćanja određenih podataka iz tablice `Employees` i tablice `Departments`.

Imajte na umu da su odgovarajući atributi pridruživanja također *"kvalificirani"* korištenjem odgovarajućih naziva tablica (odvojenih "točkom"), čime se teži boljoj čitljivosti i - u slučaju da bi ime FK bilo isto kao i naziv PK - osiguranje jedinstvene identifikacije.

(4) Agregacija vrijednosti – "*Dohvati broj zaposlenika i zbroj njihovih plaća*"

Za dohvaćanje broja svih zaposlenika u tablici `Zaposlenici` možemo koristiti SQL `COUNT`-funkciju preko različitih `employeeIDs`. Za izgradnju zbroja svih vrijednosti možemo upotrijebiti `SUM`-funkciju nad atributom, u našem slučaju, jer želimo izračunati zbroj svih plaća, atribut `plaća` (usp. sliku 2.39):

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

Slika 2.39: SQL naredba za prebrojavanje broja zaposlenika i izračunavanje zbroja njihovih plaća iz tablice `Employees`.

Rezultat ovog upita izgleda ovako (usp. sliku 2.40):

COUNT(EMPLOYEEID)	SUM(SALARY)
15	149408

Slika 2.40: Izvadak iz dohvaćanja broja zaposlenih i građenja zbroja njihovih plaća iz tablice `Employees`.

(5) Konačno, što je s grupiranjem određenih redaka – "*Dohvati broj zaposlenika grupiran prema njihovim plaćama silaznim redoslijedom*"

Za grupiranje istih podataka možemo koristiti `GROUP BY`-funkciju u `SELECT`-naredbi. U kombinaciji sa SQL `COUNT`-funkcijom preko različitih `employeeIDs` sada možemo odrediti broj redaka za svaku grupu. Za sortiranje rezultata možemo koristiti `ORDER BY`-funkciju koja navodi koji atribut treba koristiti za poredak. Time `ASC` deklarira uzlazni redoslijed dok `DESC` deklarira silazni redoslijed (usp. sliku 2.41):

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

Slika 2.41: SQL naredba za grupiranje podataka prema njihovoj plaći i sređivanje rezultata prema broju zaposlenika s tom plaćom iz tablice `Employees`.

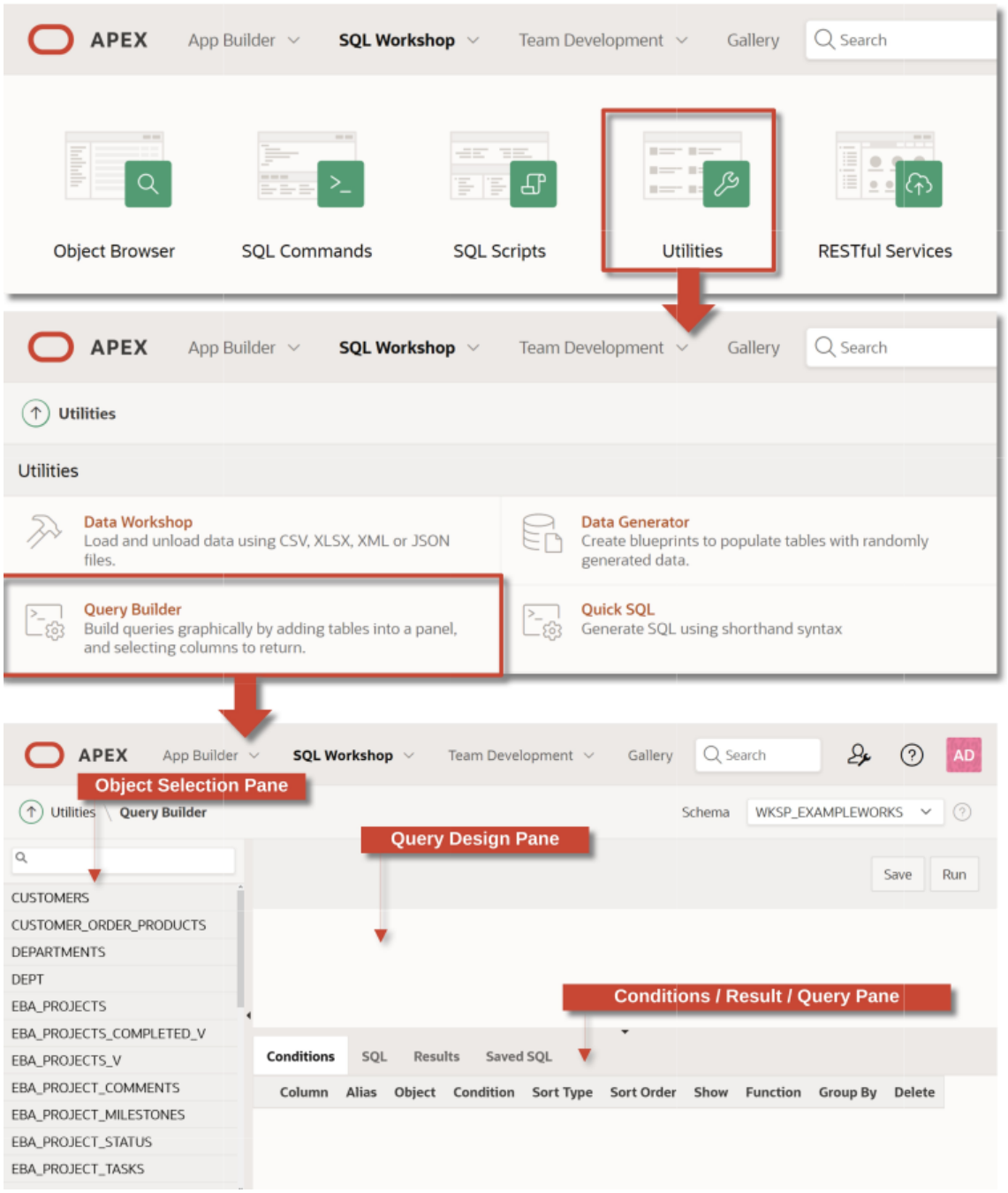
Rezultat ovog upita izgleda ovako (usp. sliku 2.42):

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

Slika 2.42: Izvadak iz grupiranja podataka prema njihovoj plaći i sređivanja rezultata prema broju zaposlenika s tom plaćom iz tablice `Employees`

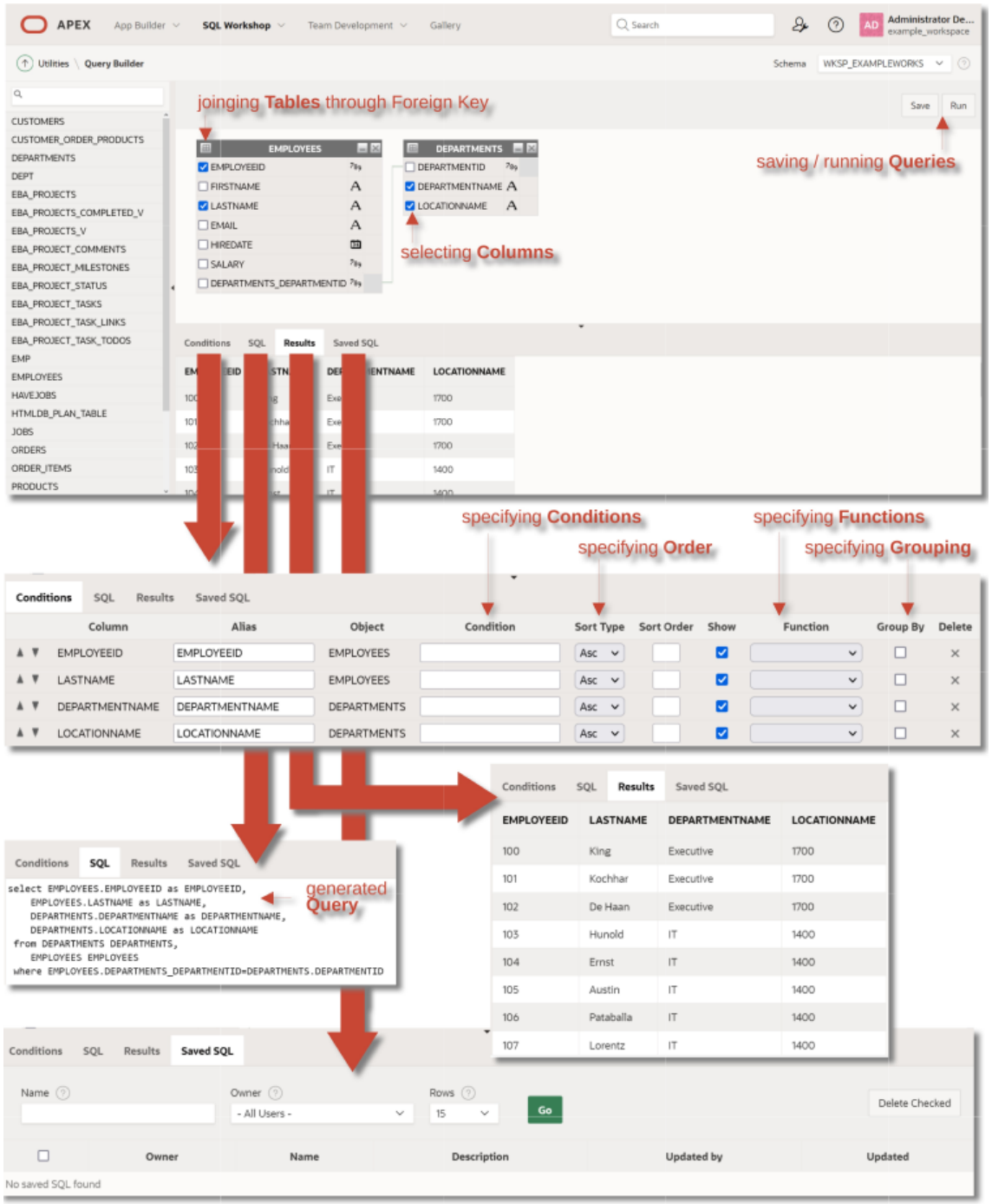
2.4.2 Kreiranje upita nad podacima pomoću ORACLE-ovog Query Buildera ("QB")

Kao što je već spomenuto, ORACLE Utilities u obliku takozvanog Query Buildera ("QB") omogućuje korisnicima s malo ili čak nimalo znanja SQL-a da postavljaju upite nad DB tablicama i da spremaju upite za daljnju upotrebu. QB-u se može pristupiti na sljedeći način (usp. sliku 2.43):



Slika 2.43: Pregled ORACLE Query Builder-a (“QB”)

Općenito, koraci koji su potrebni za izradu upita primjerno su prikazani na slici 2.44, koristeći naš primjer "join" prikazan na slici 2.37 i ukratko opisan u nastavku:



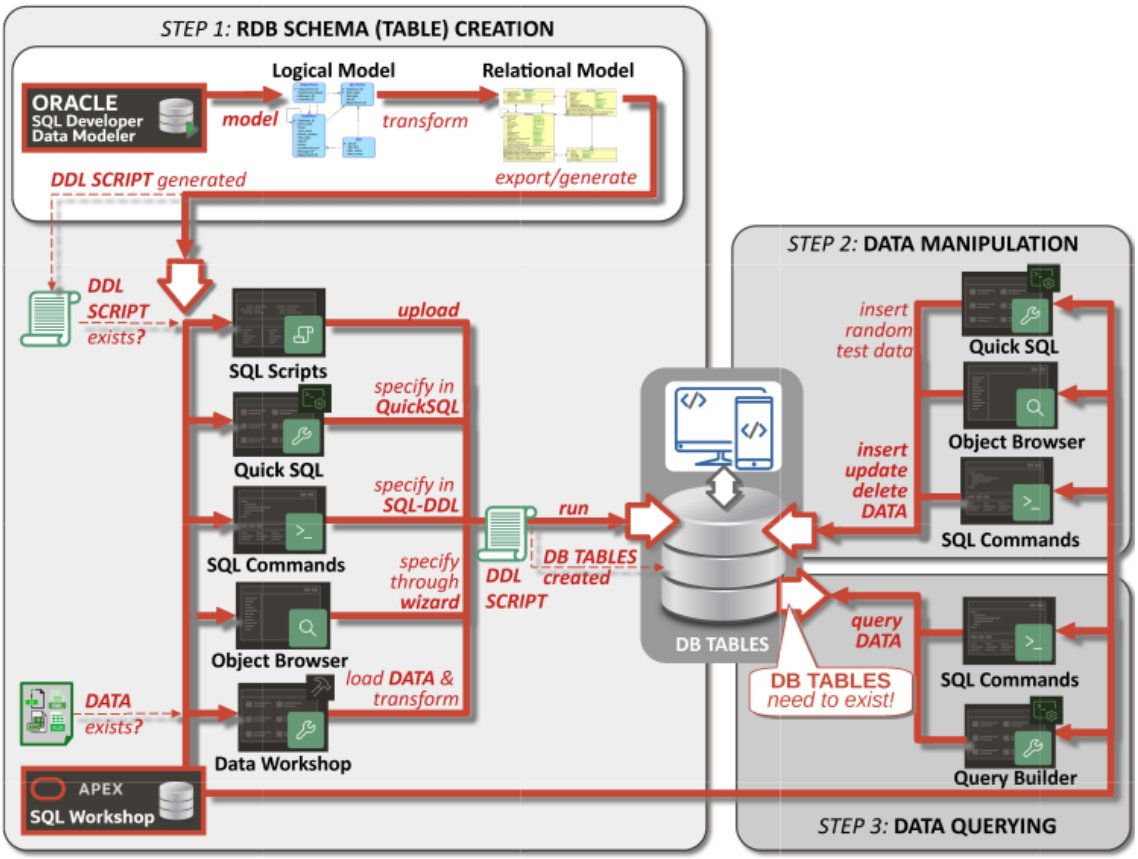
Slika 2.44: Korištenje Query Buildera za izradu spoja (join)

- Odaberite *cilj upita* u smislu jedne ili više DB tablica (nazvanih „objekti“ u QB-u) iz „okna za odabir objekta“ (usp. Slika 2.43). Ovo čini “`FROM`” dio SQL-DQL-a.
- Dodajte odabrane objekte u “Design pane” i dalje odaberite željene stupce. Ovo čini “`SELECT`” dio SQL-DQL-a.
- Izborno: Stvorite uvjete upita, tj. “`WHERE`”-dio SQL-DQL-a.
- Izvršite upit i pogledajte rezultate (kliknite na gumb "Pokreni").
- Neobavezno: Pregledajte SQL-DQL kod koji se automatski generira (kliknite na karticu “SQL”).
- Neobavezno: Spremite upit za daljnju upotrebu (kliknite gumb "Spremi").

Konačno, treba napomenuti da upite definirane pomoću QB-a ponovno automatski prevodi RDBS u odgovarajuće SQL-DQL izjave.

2.5 Izgradnja DB-sloja – Velika slika

Nakon čitanja prethodnih poglavlja, možda ćete biti potpuno zbunjeni svim ovim različitim opcijama i alternativama koje nudi ORACLE-ov krajolik alata za upravljanje DB-slojem web aplikacija. Stoga, kako bismo objasnili ovu džunglu alata, slika 2.45 daje sažeti pregled, ilustrirajući sve opcije i odgovarajući alat koji je dostupan kako bi se konačno upravljalo DB-slojem za web aplikaciju.



Slika 2.45: Pregled opcija za upravljanje DB-slojem

Sve u svemu, s lijeve strane prikazana su dva različita okruženja alata, ODM i APEX SQL Workshop. Krajnji cilj stvaranja RDB sheme u smislu tablica (korak 1) može se postići različitim putevima. Svi oni, međutim, dovode do generiranja SQL skripti koje se konačno izvode za generiranje praznih tablica. Odatle je moguće umetati podatke u tablice, ažurirati, brisati (korak 2) ili postavljati upite za podatke (korak 3) pomoću alata radne površine SQL Workshop kao što je opisano u ovom poglavlju.

Web aplikacija se može generirati na praznim tablicama. Ovaj proces se, međutim, detaljno raspravlja u sljedećem poglavlju 3.

2.6 Pitanja

1. Koja je razlika između logičkog i relacijskog modela? Koji su elementi dvaju modela?
2. Što je normalizacija RDB-sheme?
3. Opišite opcije za kreiranje tablica pomoću Oracle alata.

2.7 Odgovori

1. Logički model je apstraktniji. Elementi logičkog modela su entiteti, atributi i odnosi. Relacijski model predstavlja konkretniju i stoga specifičnu razinu DBS-a. Elementi relacijskog modela su tablice, stupci tablice (podatkovna polja) i relacije.
2. Normalizacija RDB-sheme sustavan je proces s tri koraka koji se nazivaju "prva normalna forma", "druga normalna forma" i "treća normalna forma", što dovodi do poboljšanih RDB-shema.
3. Postoji najmanje pet načina za stvaranje tablica: automatsko generiranje tablice s Oracle Data Modelerom, automatsko generiranje tablice s Quick SQL-om u Oracle APEX-u, ručno stvaranje tablice s SQL-DDL-om u Oracle APEX SQL naredbama, ručno stvaranje tablice s Oracle APEX-om Object Browser i automatsko kreiranje tablice uz Oracle APEX Data Workshop.

3. Kako koristiti APEX?

E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER

Što je Oracle APEX? Oracle Application Express (Oracle APEX) razvojno je okruženje niskog koda temeljeno na web-pregledniku za web-aplikacije koje pokreće Oracle DB. "Nizak kod" znači da je potrebno samo nekoliko programerskih vještina za izgradnju potpune web aplikacije - za prvi, jednostavni prototip web aplikacije, programerske vještine uopće nisu potrebne.

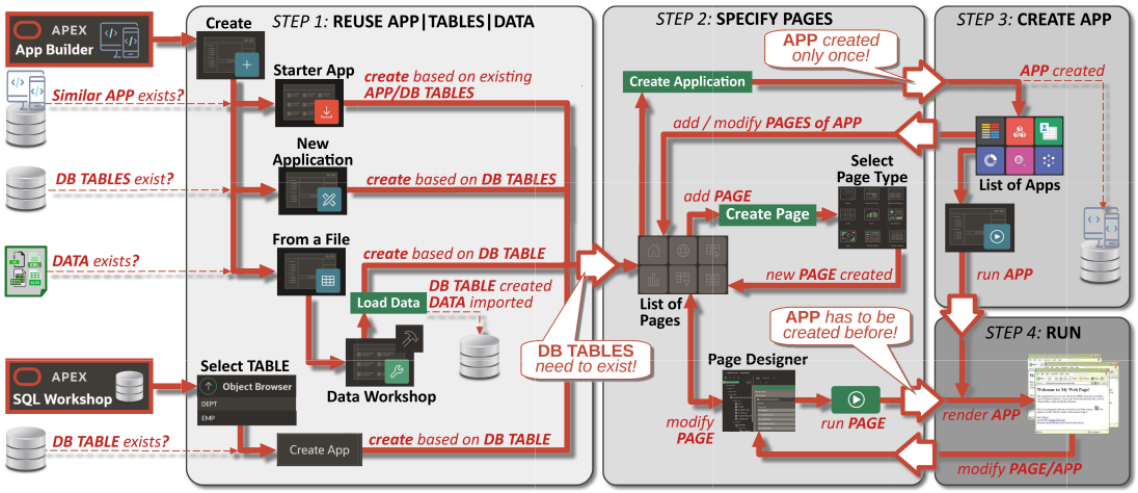
Za koje se svrhe APEX može koristiti? Sve u svemu, APEX omogućuje ne samo izgradnju DB-sloja web aplikacija pomoću APEX-a "*SQL Workshop*", kao što je bilo objašnjeno u 2. poglavlju, ali i za izradu desktop ili mobilnih web aplikacija pomoću APEX-a "*App Builder*". APEX koriste veliki i mali korisnici podjednako, u širokom broju domena primjene, noseći se sa širokim spektrom poslovnih potreba. Te poslovne potrebe mogu varirati od jednostavne transformacije lokalne proračunske tablice u onu temeljenu na webu, do realizacije potpune web aplikacije (usp. primjer koji radi predstavljen u poglavlju 2) koja omogućuje pohranu, dohvaćanje, vizualizaciju i daljnje obrađuju svaku pojedinačnu prodaju u svakoj od Walmartovih 10.500 trgovina zajedno s odgovornim odjelima, zaposlenicima i informacijama o njihovom poslu.

Koje je obrazloženje iza ovog poglavlja? Obrazloženje iza ovog poglavlja je da se najprije raspravlja o cjelokupnom procesu razvoja web aplikacija nakon čega slijedi Oracle APEXs "App Builder". Ovaj proces čini osnovu za mogućnosti navigacije između različitih alata App Builder koji se nazivaju "komponente razvoja" kako bi se izgradila web aplikacija. Drugo, dan je prilično visok pregled ovih različitih razvojnih komponenti, dok će se o detaljima raspravljati u narednim poglavljima.

3.1 Proces razvoja web aplikacije nakon kojeg slijedi APEX

Prije nego što uđemo u detalje u vezi s različitim razvojnim komponentama APEX-ovog "App Buildera", korisno je baciti pogled na proces razvoja web aplikacije nakon kojeg slijedi APEX.

APEX slijedi višestepeni i ciklički razvojni proces. Cjelokupni proces razvoja podržan od strane APEX-a, koji je ilustriran na slici 3.1, organiziran je u *multi-step, partly cyclic manner* dopuštajući *incremental development*, tj. postupno usavršavanje web aplikacije i na kraju DB sloja. Svaki od razvojnih koraka podržan je odgovarajućim čarobnjacima razvojnih komponenti. To znači da se iz opcija izbornika na najvišoj razini vodi dublje u funkcionalnost *App Builder* kako biste specificirali svaki detalj web aplikacije, koračajući naprijed-natrag kako biste izvršili inkrementalna usavršavanja ako je potrebno.



Slika 3.1: Sveukupni razvojni proces u APEX-u

APEX slijedi razvojni proces "DB-Layer First".. Mora se naglasiti da, budući da se APEX fokusira na razvoj web aplikacija vođenih DB-om, potrebno je uvijek prvo započeti razvojem DB-sloja (usp. sliku 3.1 "1. KORAK: PONOVO KORISTI APLIKACIJU | TABLICE | PODACI") . To se može učiniti ili kao što je opisano u Poglavlju 2 ili izravno unutar App Buildera, ako, npr., neke vanjske podatkovne datoteke već postoje, koje se mogu ponovno koristiti za stvaranje DB tablica (usp. Slika 3.1 "From a File"). U slučaju da DB tablice već postoje, razvojni proces se može pokrenuti (i) iz App Buildera, čime se eventualno ponovno koristi (i prilagođava) postojeća aplikacija zajedno s njezinim DB tablicama (usp. Slika 3.1 "Starter App") ili stvaranjem nove aplikacije od nule, opet na temelju postojećih DB tablica (usp. Slika 3.1 "Nova aplikacija") ili (ii) iz SQL Workshopa, jednostavnim odabirom DB tablice unutar Preglednika objekata (vidi sliku 3.1 "Select TABLE") i pokretanje procesa razvoja aplikacije na temelju ove tablice.

APEX slijedi razvojni proces "pokretan stranicom". Čim postoje DB tablice za pohranu podataka web aplikacije, one se mogu koristiti za kreiranje nove web aplikacije, slijedeći *page-driven process* (usp. Slika 3.1 "STEP 2: SPECIFY PAGES"). To znači da se može definirati jedna ili više web stranica koje čine HTML stranice konačne web aplikacije. Svaka od ovih stranica može se temeljiti na jednoj ili više DB tablica, omogućujući vizualizaciju njihovih podataka u smislu različitih *page types* uključujući npr. interaktivna izvješća, obrasce, popise, grafikone ili kalendare, ali i za interaktivno "igranje" s podacima (npr. zumiranje detalja), pa čak i za manipuliranje njima i pohranjivanje promjena natrag u DB. ove *pages can be linked together* pomoću navigacijskih izbornika, kartica, gumba ili hipertekstualnih veza.

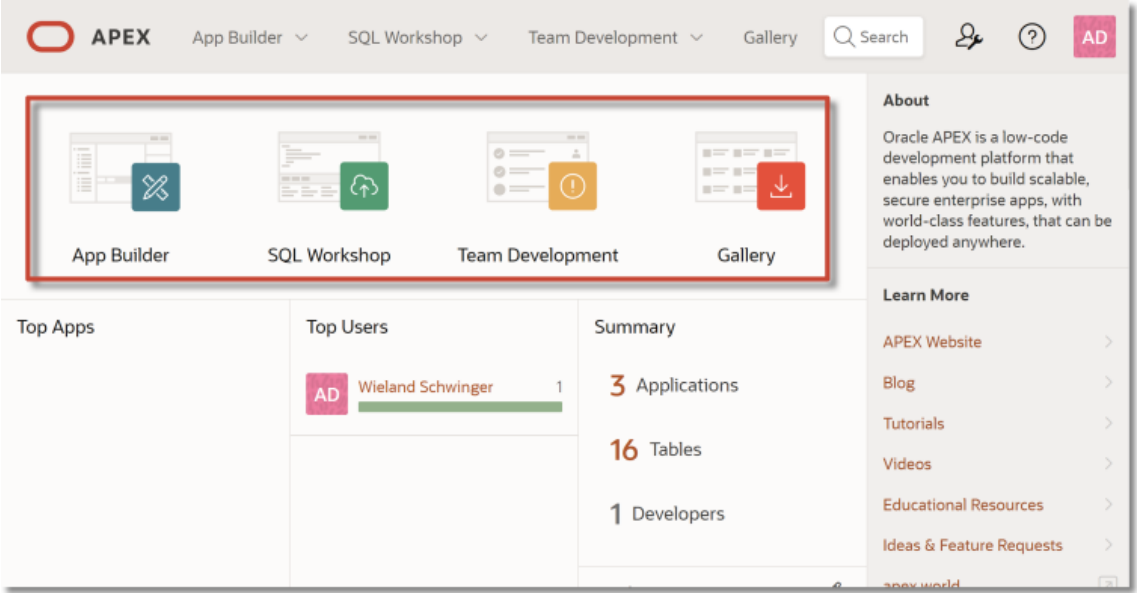
Čim se izradi jedna ili više stranica, potrebno je izraditi cjelokupnu aplikaciju (vidi sliku 3.1 "STEP 3: CREATE APP"), čime se dodatno specificiraju neka opća svojstva cijele web aplikacije (npr. izgled aplikacije) ili se samo koriste zadane postavke. Na kraju, aplikacija i/ili svaka od navedenih stranica mora se prikazati kako bi se generirale konačne HTML stranice (slika 3.1 "STEP: 4 RUN"), koji se dalje može interaktivno testirati, povremeno se povlačeći za daljnje usavršavanje web aplikacije.

3.2 Početni izbornik APEX-a

Kada se prijavite na Oracle APEX, tzv *"Workspace Homepage"* pojavljuje se. Ova početna stranica radnog prostora omogućuje pristup četiri različite razvojne komponente (usp. sliku 3.2):

- *"App Builder"* uglavnom se koristi za izradu web stranica i bit će fokus ovog poglavlja.
- *"SQL Workshop"* koristi se za izgradnju DB-sloja i već je opisan u poglavlju 2.
- *"Team Development"* koristi se, npr., za interakciju s drugim razvojnim programerima putem

sustava izdavanja ulaznica za postavljanje prekretnica itd.
- *"Gallery"* koristi se za jednostavno instaliranje unaprijed izgrađenih demo aplikacija, koje se zatim mogu dalje modificirati i prilagoditi vlastitim potrebama, opet koristeći"*App Builder*".



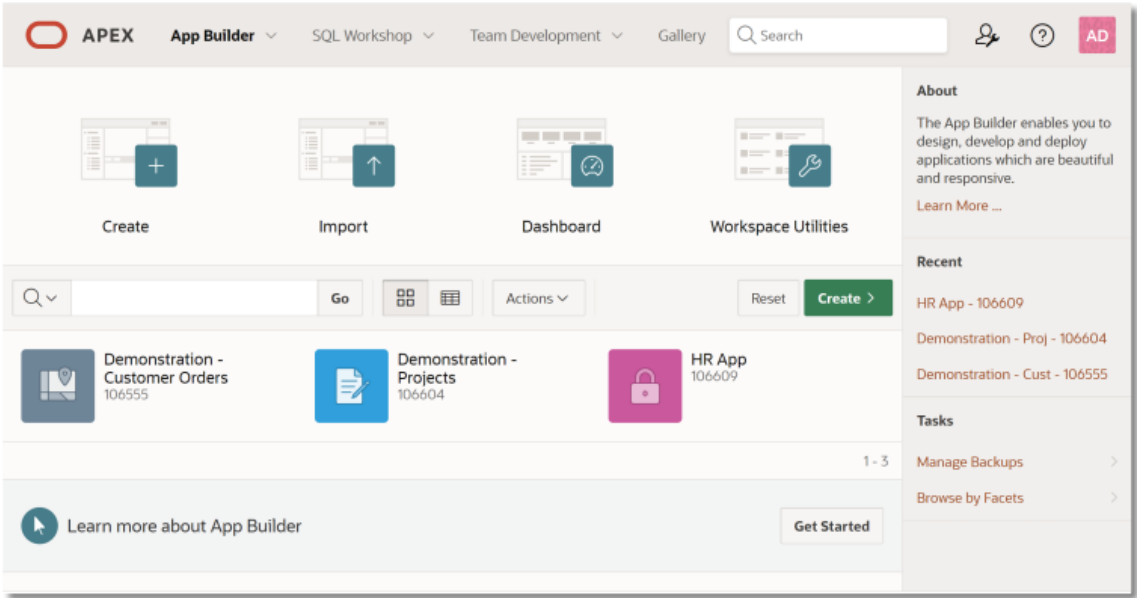
Slika 3.2: Početna stranica Oracle APEX Workspace

Na dnu početne stranice radnog prostora (vidi sliku 3.2),regije "*Top Apps*", "*Top Users*", i "*Summary*" nude informacije u stvarnom vremenu o razvojnim aktivnostima u trenutnom radnom prostoru baze podataka, dok se na desnoj strani navode daljnji APEX resursi za učenje.

3.3 Pregled App Buildera – Stvorite svoje aplikacije i upravljajte njima

Prilikom navigacije do početne stranice App Buildera, imate sljedeće opcije (usp. Slika 3.3):

- *Create* nova web aplikacija.
- *Import* prethodno izvezene web aplikacije.
- View the "Dashboard" pružanje statistike o već razvijenim web aplikacijama.
- *Access "Workspace Utilities"* pružanje širokog spektra usluga poput daljinskog pristupa podacima ili funkcije sigurnosnog kopiranja.



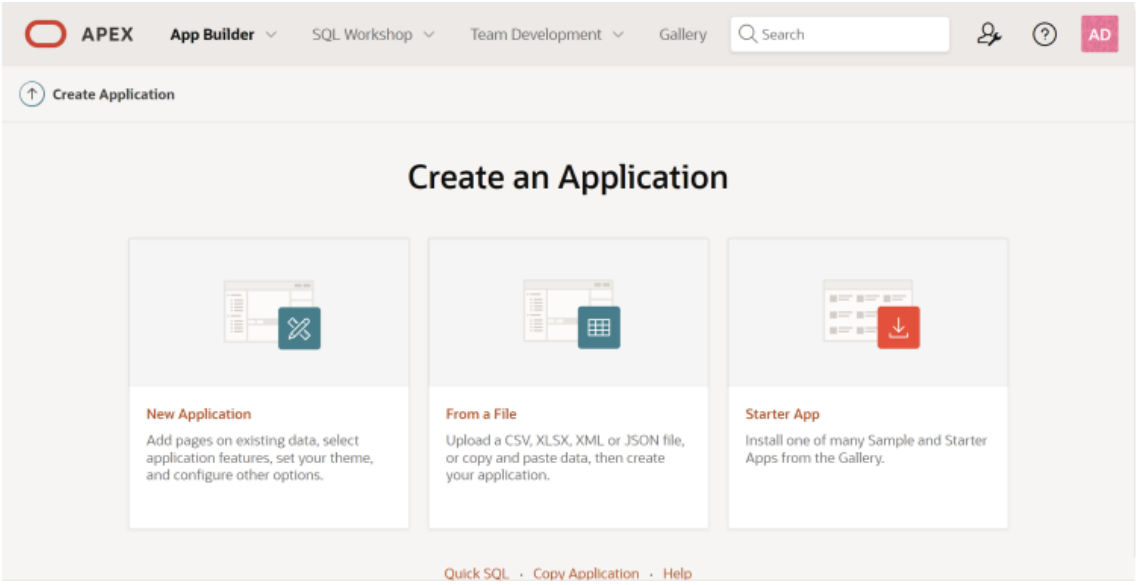
Slika 3.3: Pregled App Buildera

U donjoj polovici prozora (vidi sliku 3.3), navedene su već postojeće web aplikacije koje se mogu odabrati za daljnje uređivanje ili renderiranje (usp. sljedeće odjeljke).

3.4 Stvorite aplikaciju – tri slučaja upotrebe

Prilikom odabira *"Create"* u App Builderu, "*Create Application Wizard*" donosi tri dodatne opcije (usp. sliku 3.4), podržavajući zapravo slučajeve korištenja u stvarnom svijetu koji uzimaju u obzir mogući potencijal ponovne upotrebe u vezi s podacima i samom aplikacijom:

- *"New Application"* treba odabrati ako DB-sloj (ili njegov dio) već postoji ili ako se DB-sloj treba razviti "u hodu" tijekom razvoja aplikacije.
- *"From a File"* je opcija izbora ako se DB-sloj treba automatski generirati iz vanjske datoteke, npr. EXCEL-datoteke.
- *"Starter App"* konačno omogućuje ponovnu upotrebu (i vjerojatno dodatnu izmjenu) već postojeće aplikacije zajedno s njezinim DB-slojem iz "Galerije" aplikacija koje su već spomenute gore.



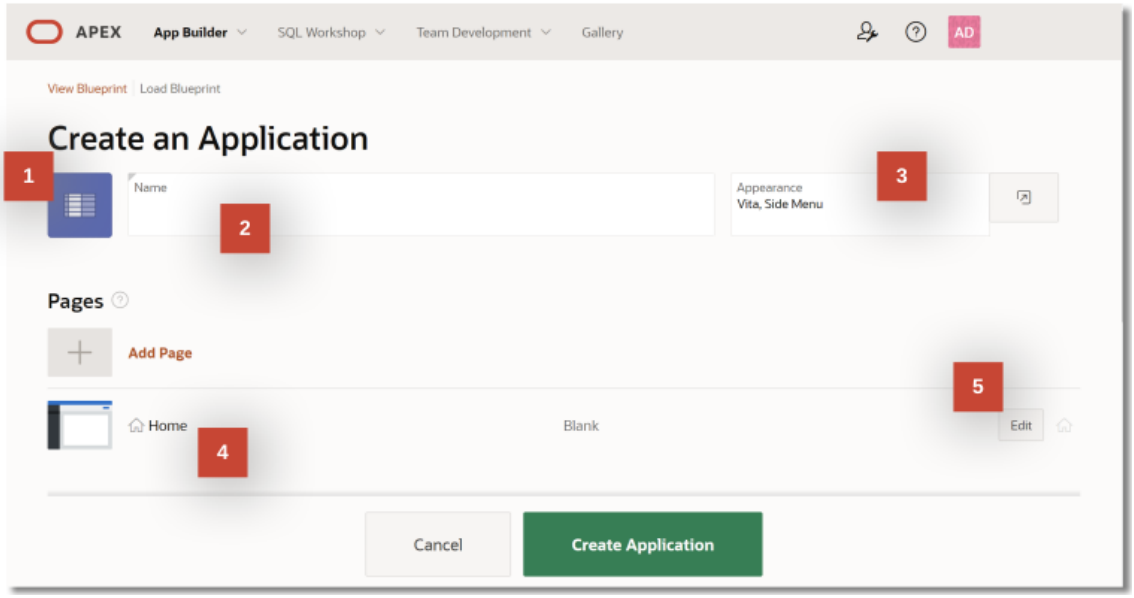
Slika 3.4: Čarobnjak za izradu aplikacije: Tri slučaja upotrebe pri izradi aplikacije

U nastavku će fokus biti na prvoj opciji *"New Application"*, druga opcija *"From a File"* bit će detaljnije opisano u poglavlju 4, a sa trećom opcijom *"Starter App"* se neće baviti.

3.5 Stvorite aplikaciju – svojstva, stranice, značajke i postavke

Prilikom izrade nove web aplikacije, neki *common properties* moraju se definirati, *pages* moraju se stvoriti, a neki dodatni vrlo korisni *features* i *settings* može se izabrati. U nastavku se detaljnije raspravlja o svim ovim definicijama, gdje je slika 3.5) prikazuje gornji dio GUI-a i Slika 3.6) pokazuje njegov donji dio.

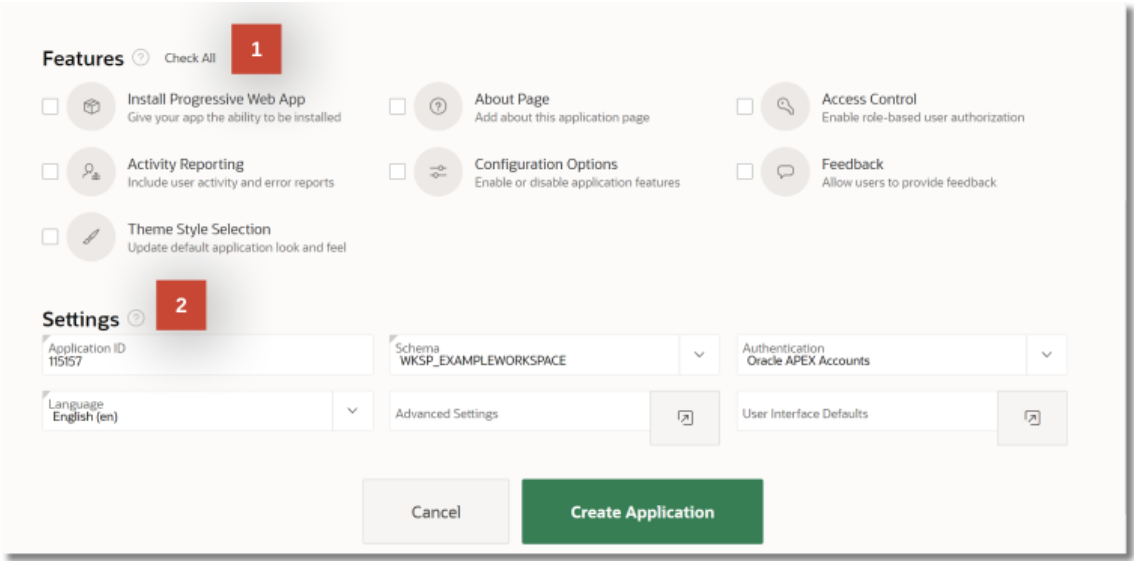
Uobičajena svojstva – ikona, naziv i izgled aplikacije. Što se tiče zajedničkih svojstava, kao što je ilustrirano na slici 3.5, (1) proizvoljan *icon* može se odabrati ili učitati, koristiti kao Favicon i ikona za PWA, Apple touch ili App Builder, (2) a *name* mora se unijeti i (3) *appearance* web aplikacije može se odabrati između nekih unaprijed definiranih opcija, određujući njezin "izgled i dojam" u smislu stila teme (npr. boja, veličina fonta itd.) i *kind of navigation menu* (npr. bočni izbornik ili gornji izbornik), pri čemu je dostupan i zadani izgled.



Slika 3.5: Svojstva i stranice

Izrada stranice za aplikaciju. Glavni zadatak je, međutim, postupno sukcesivno graditi web aplikaciju *adding new pages* koristeći *"Create Page Wizard"* (započeo od *"Add Page"*, usp. (4) na slici 3.5. Nakon što je stranica kreirana, možete, u bilo kojem trenutku tijekom razvoja aplikacije, *edit* the *composition* stranice, *alter* redoslijed stranica i *delete*njih (usp. (5) na slici 3.5). U donjem dijelu slike 3.5, prikazuje se već stvorena stranica "Početna". Dodatne pojedinosti o"*Create Page Wizard*" dati su u odjeljku 3.6.

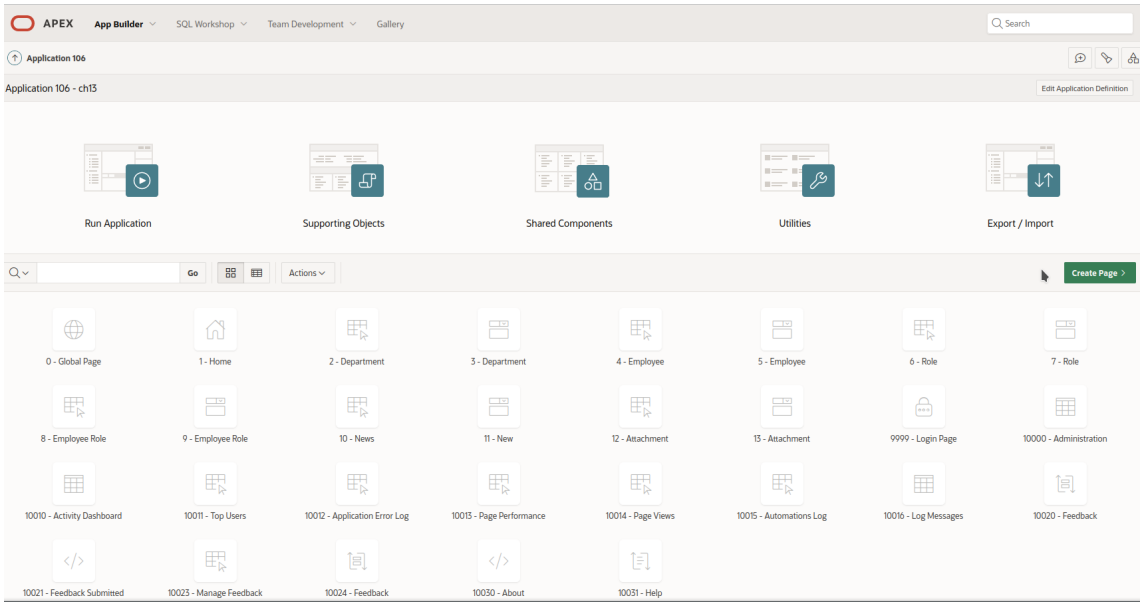
Značajke i postavke aplikacije. Konačno, aplikacija se može poboljšati nekim vrlo korisnim *features* (usp. (1) na slici 3.6) kao što je "About Page", stranica za automatsku "Kontrolu pristupa" ili stranica za "Povratne informacije korisnika" kao i po nekim zajedničkim *settings*, npr. naziv sheme koja sadrži sve DB-tablice i tako tvori DB-sloj za aplikaciju ili korišteni jezik (usp. (2) na slici 3.6).



Slika 3.6: Značajke i postavke

Nakon klika na "*Create Application*" gumb the "*App Builder Homepage*" pojavljuje se prikazu-

jući novostvorenu aplikaciju sa svojim stranicama (vidi sliku 3.7).

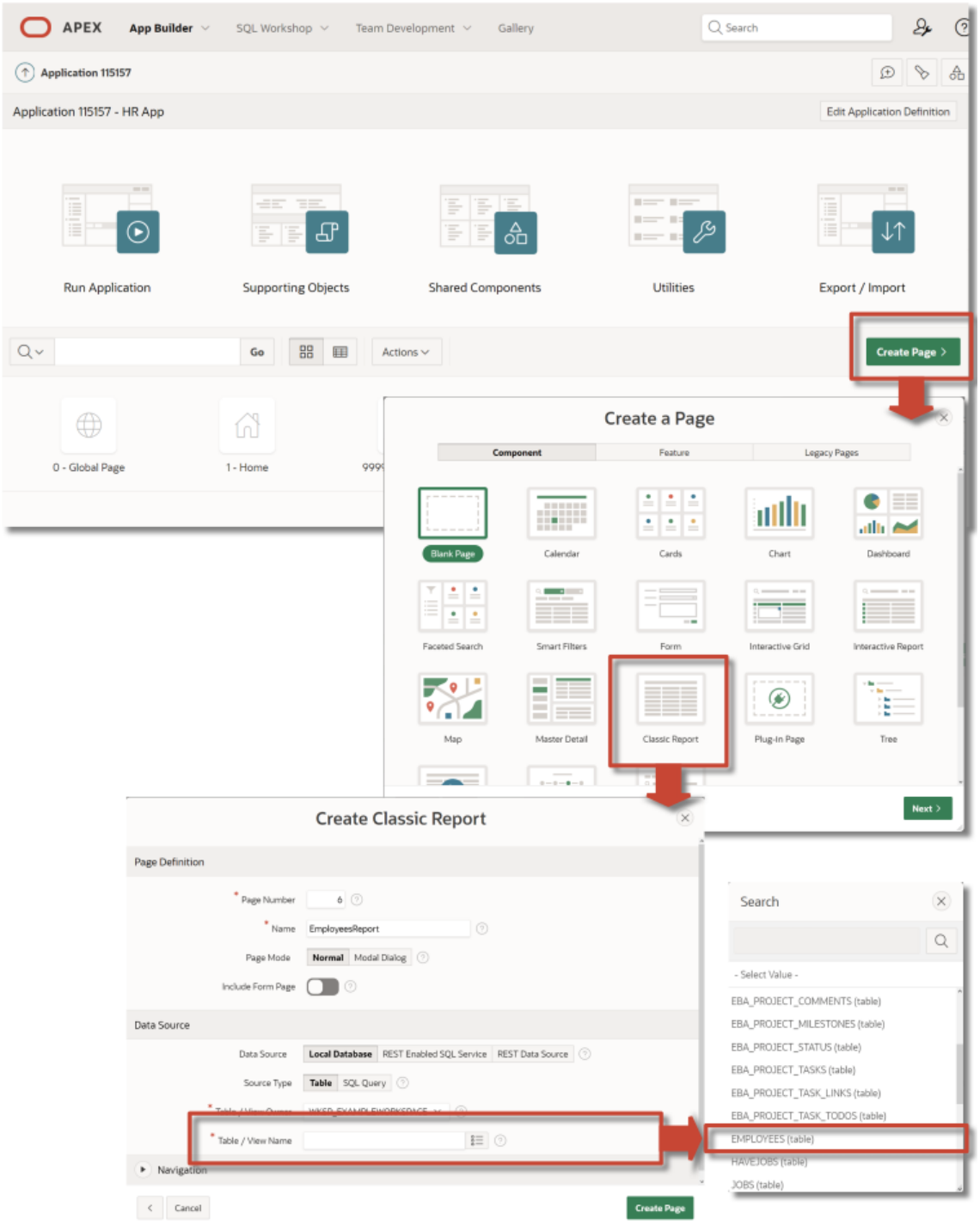


Slika 3.7: Početna stranica aplikacije - pogled razvojnog programera

3.6 Specificirajte stranice

Od čega je sastavljena stranica Kao što je već spomenuto, stranica je osnovni građevni blok web aplikacije, tj. svaka aplikacija se sastoji od najmanje jedne ili više stranica. Svaka stranica može imati gumbe i polja (tzv *"items"*) koji su grupirani u spremnike tzv *"regions"*. Stranice također mogu uključivati *application logic*, npr. izvesti *calculations* (zvano *"computations"*) , i izvoditi *validations* kao što su provjere ispravnosti tijekom manipulacije podacima.

Birajte između 18 različitih vrsta stranica. Stranica može prikazati podatke DB tablica u različitim oblicima određenim takozvanim "vrstama stranica", uključujući npr. interaktivna izvješća , popisi , kalendari ili dijagrami . Vrsta stranice mora biti odabrana, kad god se kreira nova stranica, bilo pomoću gumba "Dodaj stranicu" unutar "Stvori aplikaciju" dijalog (usp. slika 3.5) ili korištenjem gumba "Stvori stranicu" unutar početne stranice aplikacije programera (usp. 3.8). Ukupno je prikazano 18 različitih vrsta stranica kao što je prikazano na slici 3.8. Određeni tip stranice određuje *contents*, *composition*, i *layout* stranice što se može prepoznati ako pogledate ikone vrste stranice na slici 3.8. pojedinosti o svakoj od ovih vrsta stranica dane su u narednim poglavljima.



Slika 3.8: Izrada stranice

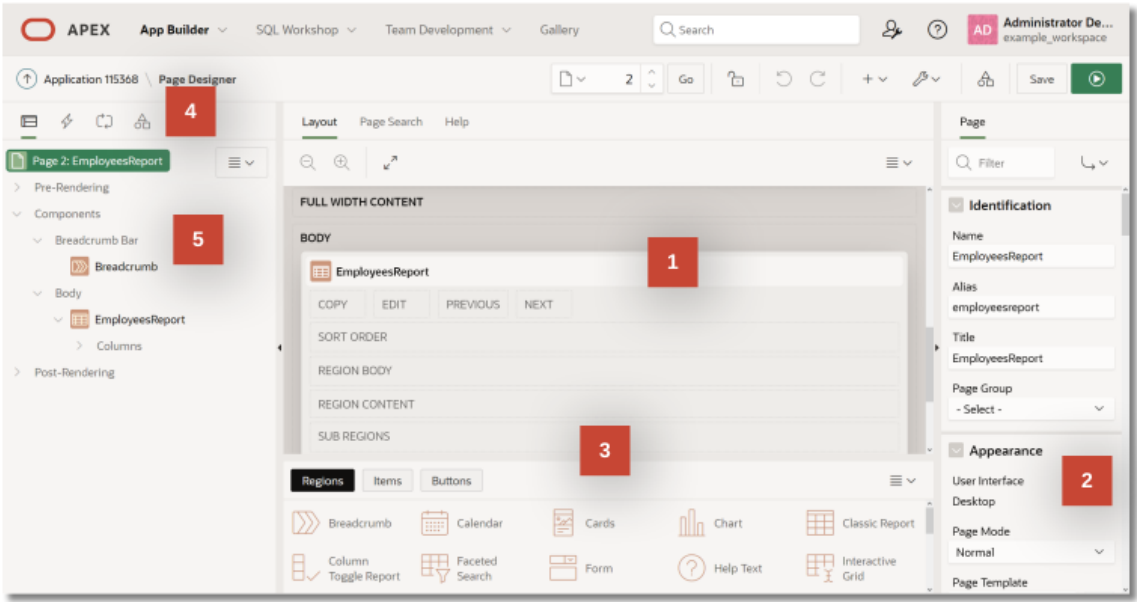
Dijalog specifikacije stranice – mapiranje tablice na stranicu. Prilikom odlučivanja za određenu vrstu stranice, konkretni dijalog za specifikaciju stranice koji prikazuje "Page Creation Wizard" naravno ovisi o odabranoj vrsti stranice. Unatoč tome, budući da se gotovo sve dostupne vrste stranica temelje na DB-tablicama, glavni zadatak je obično preslikati odgovarajuće DB-tablice na stranicu, pružajući izvor za vizualizaciju podataka i cilj za manipulaciju podacima. Slika 3.8 prikazuje primjer dijaloga za prilično uobičajenu vrstu stranice "*Classic Report*". Može se vidjeti da, osim nekih osnovnih informacija kao što je "Naziv", također mora biti specificirano mapiranje na DB-sloj odabirom "Izvora podataka" u smislu DB tablice koja bi trebala biti izvor za ovo izvješće

(na slici 3.8 odabiremo tablicu "Zaposlenici" našeg primjera koji radi). Treba napomenuti da za prikaz podataka iz temeljne DB tablice, APEX inherentno koristi SQL-DQL u pozadini čim se HTML stranica renderira.

Čim se doda nova stranica, "*Page Designer*" komponenta APEX-a (usp. Odjeljak 3.7) automatski se otvara za ovu stranicu, omogućujući pregled i uređivanje specifikacije stranice. Istodobno, stranica se također može odmah i automatski prikazati u HTML-stranicu, npr. za potrebe testiranja i inkrementalnog razvoja, pomoću gumba "Pokreni" (vidi odjeljak 3.8).

3.7 Održavajte i mijenjajte stranicu – Dizajner stranice

Već kreirane stranice moguće je, naravno, i dalje održavati i unapređivati korištenjem tzv "*Page Designer*". Page Designer je pun značajki *Integrated Development Environment (IDE)* koji uključuje alatnu traku i višestruka okna (usp. Slika 3.9). Funkcionalnost koju pruža Page Designer je ogromna, dopuštajući, između ostalog, potpunu izmjenu sastava/izgleda stranice i njezinih komponenti (usp. (1), (2) i (5)), uključujući stavke, gumbe i regije (usp. (3)), ali i za izmjenu prikazivanja stranice (usp. (ili logike aplikacije u smislu izračuna i provjera valjanosti (usp. (4)).



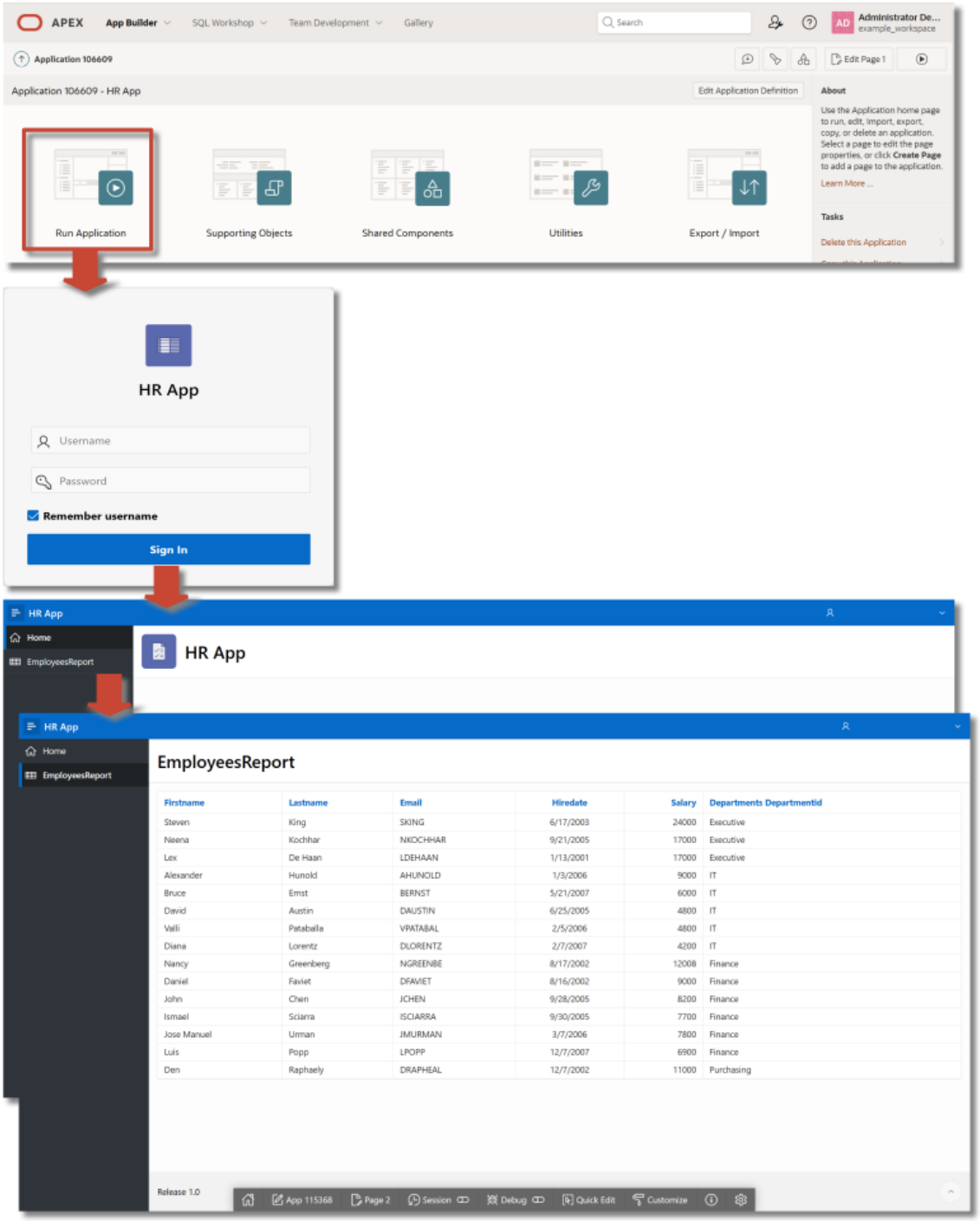
Slika 3.9: Dizajner stranica

Pojedinosti o funkcionalnosti koju nudi Page Designer mogu se pronaći u narednim poglavljima.

3.8 Pokrenite aplikaciju ili stranicu

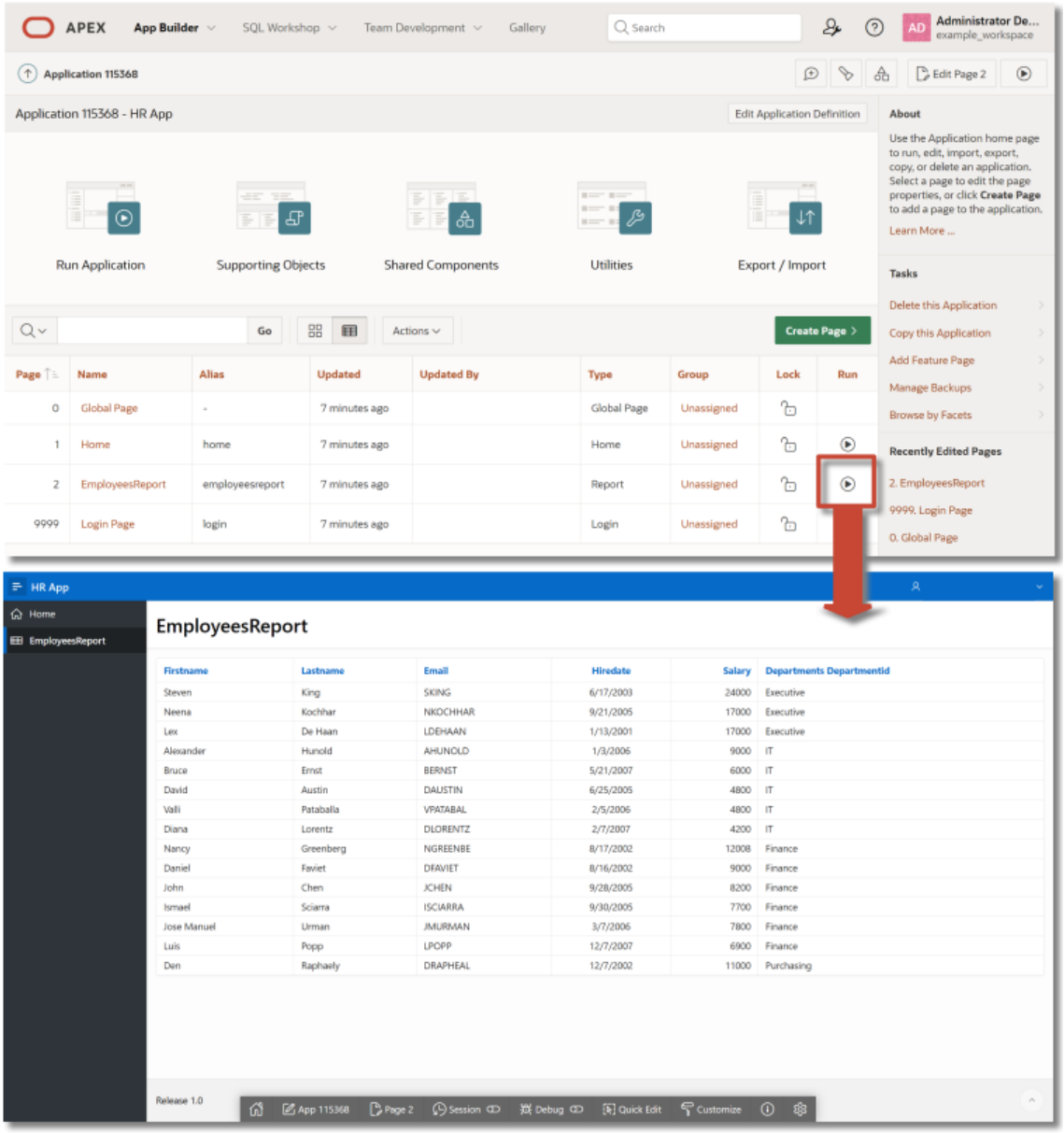
Za pregled renderirane verzije aplikacije ili pojedine stranice potrebno ju je podnijeti na tzv *Oracle APEX engine* klikom na "*Run*" gumb *application home page* (usp. Slika 3.10 i Slika 3.11). APEX motor dinamički renderira i obrađuje stranice u vidljive *HTML pages* na temelju podataka o aplikaciji koji su pohranjeni u "internim" DB tablicama. Postoje dvije različite opcije dostupne na *Application Homepage*, ovisno o tome želite li pokrenuti cijelu aplikaciju, tj. sve stranice ili samo jednu stranicu za npr. potrebe testiranja.

- **Pokrenite cijelu aplikaciju**, tj. sve stranice: Ova je opcija dostupna na početnoj stranici aplikacije kao što se može vidjeti na slici 3.10.



Slika 3.10: Pokretanje cijele aplikacije

- **Run Individual Pages**. Dok stvarate nove stranice, također je moguće pokrenuti svaku stranicu zasebno kako biste dobili trenutni dojam o izgledu i dojmu stvarne stranice. Gumb "Pokreni stranicu" nalikuje maloj crnoj ikoni za reprodukciju i prikazuje se na desnoj strani svake stranice vaše aplikacije kao što se može vidjeti na slici 3.11. Međutim, treba napomenuti da čak i ako pokrenete samo pojedinačnu stranicu, dostupne su i sve ostale stranice.



Slika 3.11: Pokretanje pojedinačnih stranica

Imajte na umu da se druge opcije koje su dostupne na početnoj stranici aplikacije osim "Pokreni aplikaciju" (npr., Podržavajući objekti) obrađuju u narednim poglavljima.

3.9 Pitanja

1. Koje mogućnosti nudi čarobnjak za izradu aplikacije i čemu one služe?
2. Koje značajke nudi čarobnjak "Nova aplikacija"?
3. Kakve stranice može generirati APEX?

3.10 Odgovori

1. postoje tri opcije: a) "Nova aplikacija" omogućuje razvojnom programeru dodavanje stranica za postojeće podatke, odabir značajki aplikacije, postavljanje teme i konfiguraciju, b) "Iz datoteke" omogućuje razvojnom programeru učitavanje datoteka u formatima kao što je CSV , XLSX, XML i JSON ili kopirajte/zalijepite podatke i pokrenite izradu aplikacije i c)

"Početna aplikacija" omogućuje razvojnom programeru da instalira jednu ili više primjera i početnih aplikacija iz Galerije.

2. Postoji nekoliko korisnih značajki kao što je automatsko stvaranje stranice "O stranici", pružanje administratoru aplikacije stranice za dodjeljivanje privilegija korisnicima u značajci "Kontrola pristupa", automatsko stvaranje stranice "Povratne informacije korisnika", izvješćivanje o aktivnostima, odabir stila teme , mogućnosti konfiguracije i izbor instaliranja progresivne web aplikacije.
3. APEX može generirati širok raspon stranica: obrazac, izvješće (interaktivna mreža, interaktivno izvješće, klasično izvješće, glavni detalj), prazna stranica, kalendar, kartice, grafikon, nadzorna ploča, fasetirano pretraživanje, pametni filtar, karta, stranica za pretraživanje, dodatak na stranici, stablu, učitavanju podataka, čarobnjaku i jedinstvenom popisu zadataka.

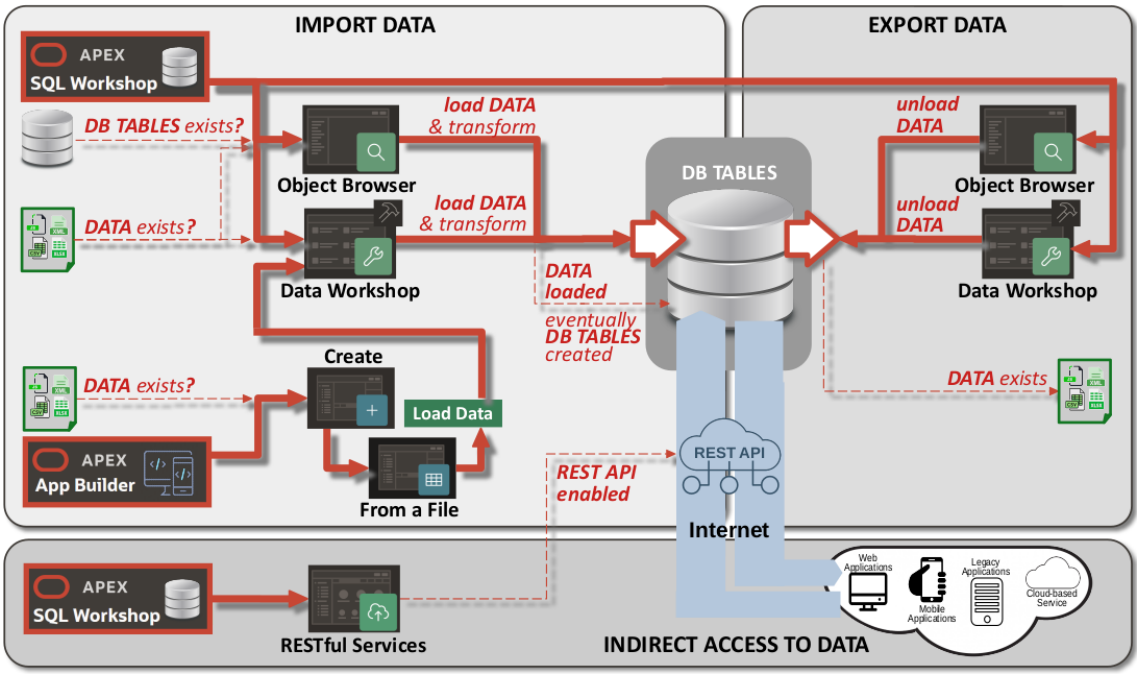


4. Kako razmijeniti podatke u APEX-u?

E. KAPSAMMER, W. RETSCHITZEGGER, W. SCHWINGER

Glavni zadatak pri izgradnji DB-sloja za web aplikaciju je stvaranje DB-tablica i umetanje odgovarajućih podataka (vidi Poglavlje 2). To se, međutim, može učiniti samo ručno ako podaci još ne postoje. U slučaju da već postoje dostupni podaci unutar vanjskih datoteka, oni se jednostavno mogu ponovno upotrijebiti i *uvesti* u Oracle DB. U isto vrijeme, moglo bi biti poželjno *izvoziti* već postojeće podatke u datoteku s vremena na vrijeme, tako da ih neka druga, vanjska aplikacija može ponovno upotrijebiti u drugom koraku. I konačno, u slučaju da nekim vanjskim klijentima (npr. druge web/mobilne/naslijeđene aplikacije ili usluge temeljene na oblaku) treba omogućiti *mrežni pristup* podacima unutar naše baze podataka, umjesto samo razmjene podatkovnih datoteka s vremena na vrijeme postoji i mogućnost pružanja neizravnog online pristupa našim DB podacima.

Ova tri slučaja upotrebe za razmjenu podataka ilustrirana su na slici 4.1, pružajući sažetak različitih opcija koje Oracle APEX pruža za uvoz, izvoz i neizravan pristup DB podacima. Ovo poglavlje ima za cilj dati pregled ovih različitih opcija.



Slika 4.1: Mogućnosti razmjene podataka u APEX-u

4.1 Uvoz i izvoz podataka pomoću “Data Workshopa”

Scenarij aplikacije za Data Workshop. Data Workshop je alat jednostavan za korištenje, koji nudi "čarobnjaka" kako bi se pojednostavio zadatak uvoza/izvoza podataka iz/u vanjske datoteke. Data Workshop posebno je prikladan za scenarije primjene koji se bave podacima umjerene veličine (manje od 10 tablica) koji imaju samo jednostavne standardne tipove podataka (npr. bez polja s više vrijednosti ili ugniježđenih struktura). Treba napomenuti da za ponovnu upotrebu i uvoz *ogromnih i složenih skupova podataka*, Oracle osigurava druge odgovarajuće alate, npr. u smislu takozvanog “*SQL*Loader Utility*”. U nastavku, prije svega, odjeljak 4.1.1 i odjeljak 4.1.2 bave se uvozom podataka, dok odjeljak 4.1.3 opisuje izvoz podataka.

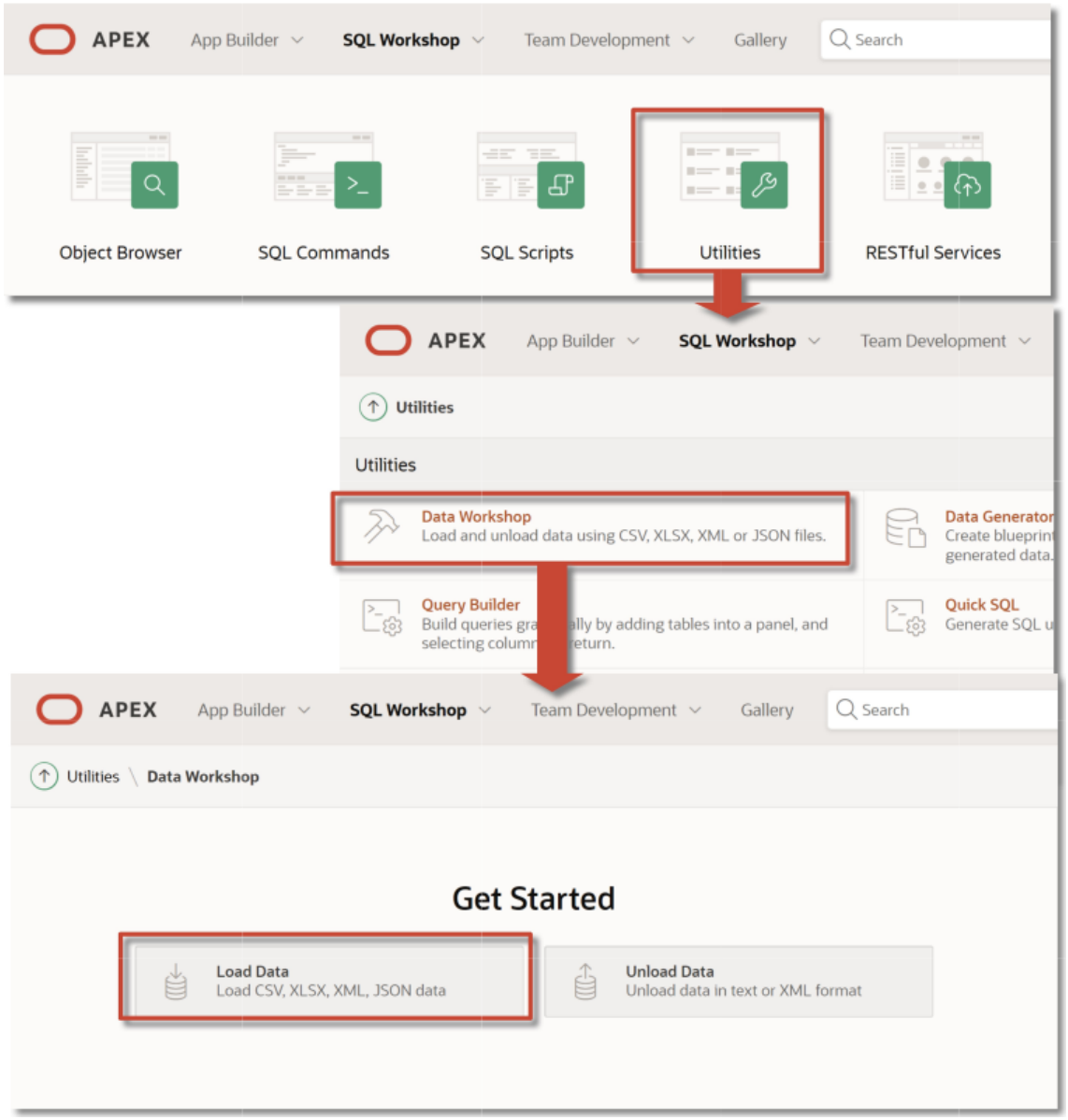
4.1.1 Uvoz podataka

Formati datoteka. Sveukupno, Data Workshop omogućuje učitavanje podataka iz vanjskih datoteka pridržavajući se sljedećih formata:

1. Bilo koji standardni *odvojeni format* koji je *razdvojen tabulatorom* ili *odvojen zarezom* (“*CSV*” – *Vrijednosti odvojene zarezom*).
2. *XLSX datoteke* (tj. *Excel radne knjige*). Imajte na umu da, ako učitana XLSX datoteka sadrži više radnih listova, prema zadanim postavkama bira se prvi list. Da biste učitali drugi list, možete ga odabrati s popisa za odabir "Odaberi list".
3. *JSON datoteke* (*Java Script Object Notation*). Imajte na umu da je podržana samo jedna razina ugniježđivanja.
4. *XML datoteke* (*eXtended Markup Language*). Imajte na umu da je, slično JSON datotekama, podržana samo jedna razina ugniježđivanja.

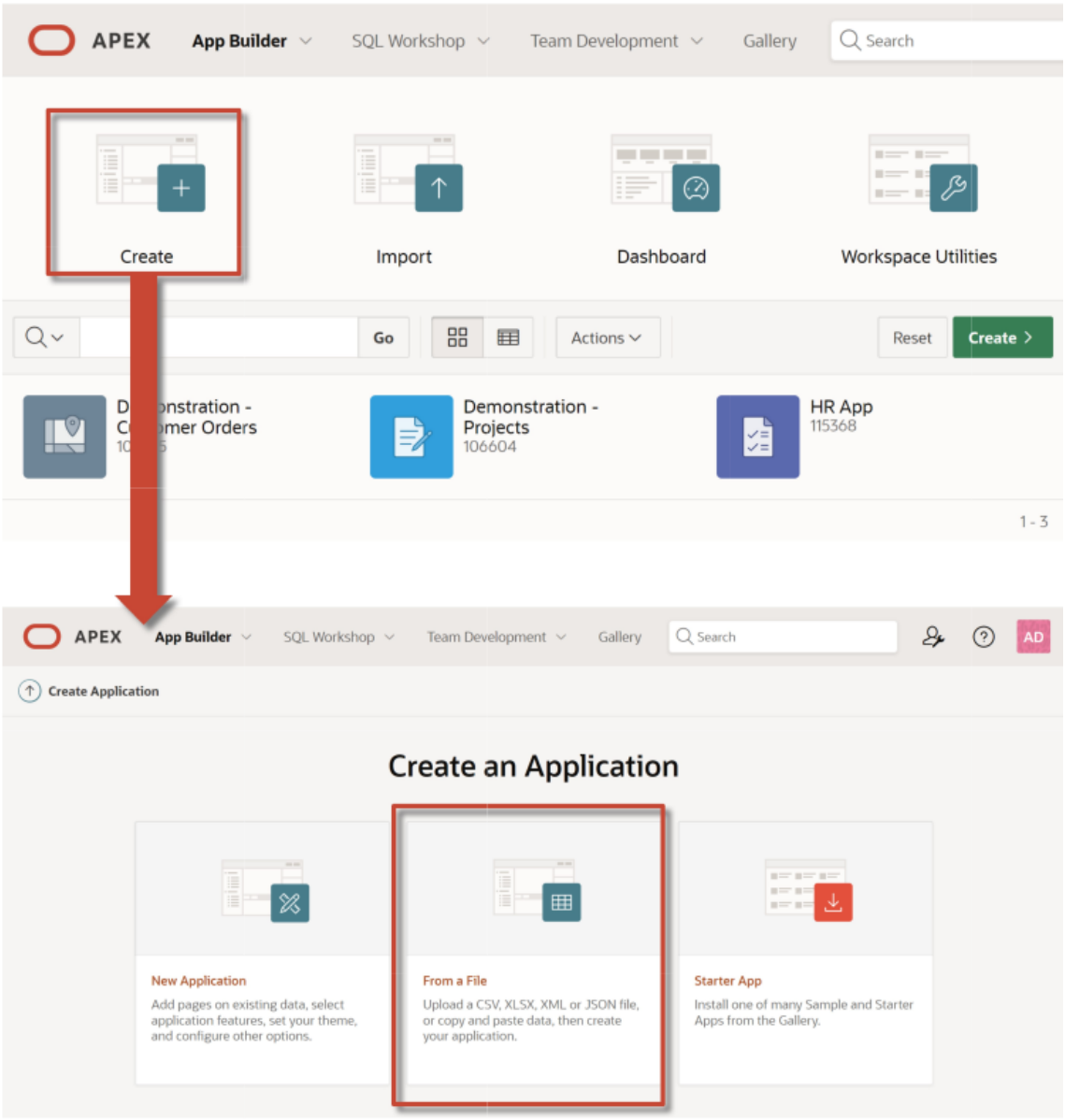
Pristup Data Workshopu za uvoze. Data Workshopu se može pristupiti na dva alternativna načina:

1. Pokrenite Data Workshop na SQL Workshop-u kao što je prikazano na slici 4.2.



Slika 4.2: Korištenje Data Workshopa za učitavanje podataka - pristup putem SQL Workshopa

2. Alternativno, Data Workshopu se također može pristupiti iz APEX App Buildera prilikom izrade aplikacije kao što je već spomenuto u Poglavlju 3 (usp. Slika 4.3), koristeći "Iz datoteke" opciju (usp. sliku 4.1).

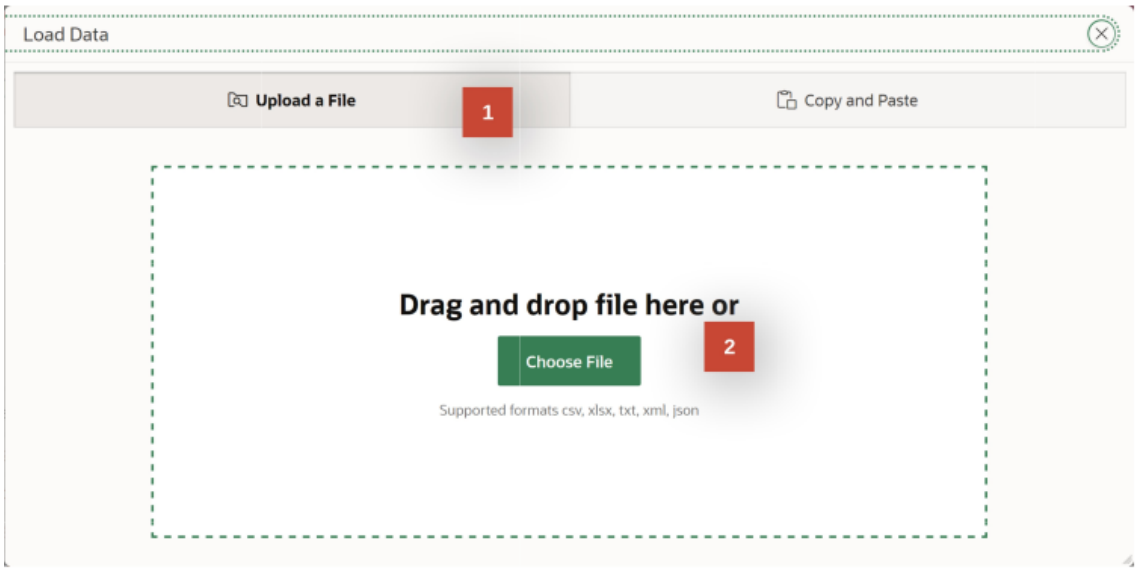


Slika 4.3: Korištenje Data Workshopa za učitavanje podataka - pristup putem App Buildera

4.1.2 Koraci uvoza

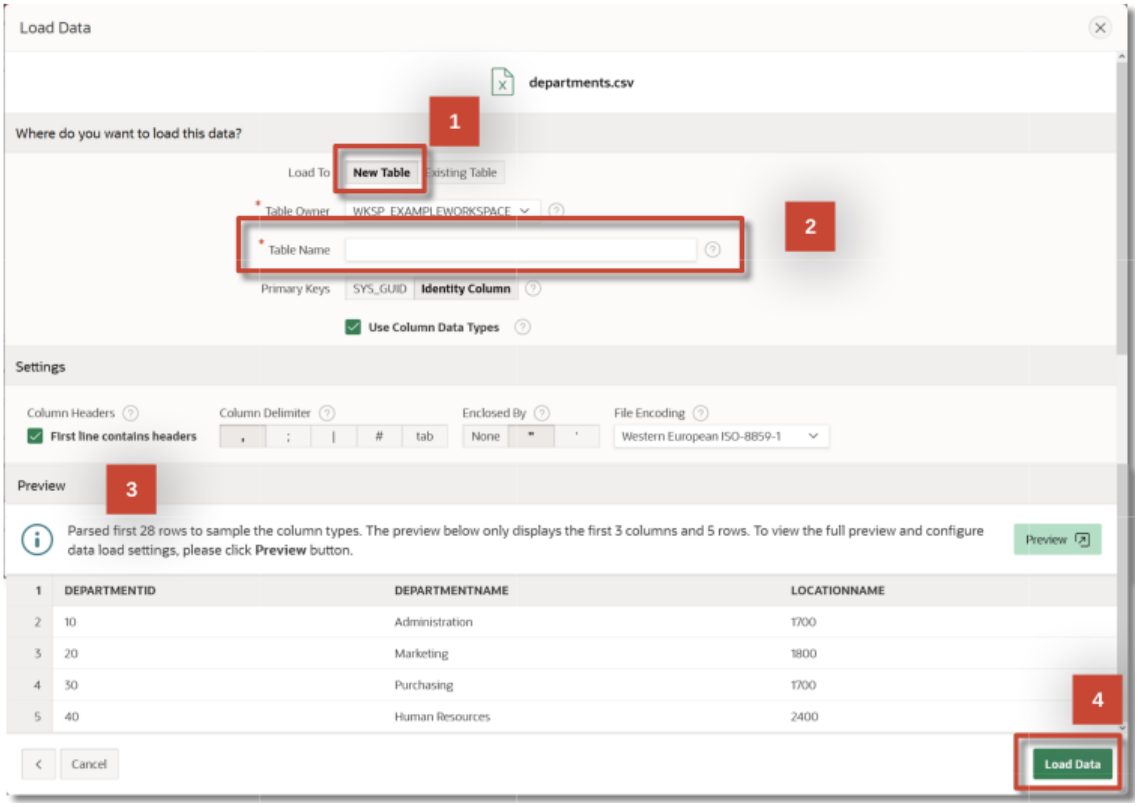
U nastavku se ukratko raspravlja o koracima potrebnim za uvoz podataka, zajedno s nekim važnim pitanjima koja su korisna da biste razumjeli kako koristiti "Load Wizard":

Pružanje izvora podataka. Prvi, sasvim prirodan korak je osigurati izvor podataka što se može učiniti učitavanjem (povuci/ispusti ili odaberi datoteku) CSV datoteka, XLSX, JSON ili XML datoteka ili jednostavnim kopiraj/zalijepi samo razgraničene podatke (usp. sliku 4.4). Imajte na umu da se ovaj dijalog pojavljuje kada kliknete na *"gumb Učitaj podatke"* (usp. sliku 4.2) ili putem *"gumb iz datoteke"* (usp. Slika 4.12).



Slika 4.4: Pružanje izvora podataka

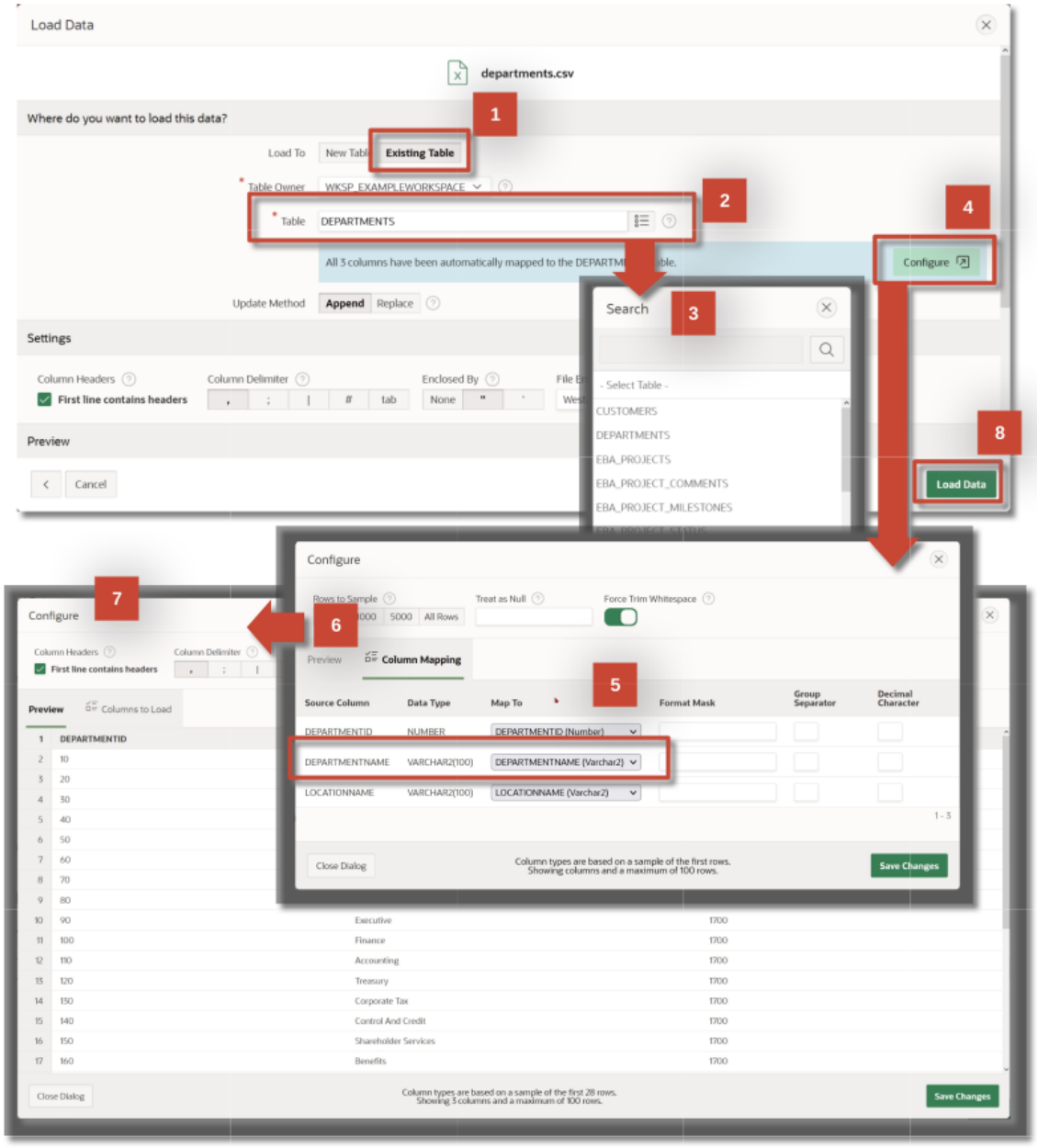
Konfiguracija mapiranja podataka u tablicu. Središnji korak čarobnjaka za učitavanje je konfiguracija mapiranja podataka u tablicu, što je vizualizirano na slici 4.5 i 4.6, gdje su najvažnije opcije konfiguracije označene brojevima (1)-(4) i (1)-(8), a koje su opisane u nastavku.



Slika 4.5: Učitavanje podataka u novu tablicu

1. **Učitaj podatke u postojeću ili novu tablicu.** Prije svega, treba odlučiti hoće li se podaci pohraniti u već postojeću DB tablicu ili treba koristiti novu DB tablicu (usp. Slika 4.5 (1) odnosno slika 4.6 (1)).

2. **Navedite vlasnika i naziv tablice.** Za oba slučaja, vlasnik tablice i naziv tablice moraju biti navedeni (usp. Slika 4.5 (2) i Slika 4.6 (2) redom).
3. **Postavke za CSV datoteke.** *odjeljak postavki* (usp. Slika 4.5 (3)) dopušta tri različite važne konfiguracije specifične za razgraničene ("CSV") datoteke. Konfiguracijska opcija "*Zaglavlja stupaca*" omogućuje određivanje sadrži li prvi redak podataka nazive stupaca. "*Razdjelnik stupca*" može se slobodno odabrati kao što se može vidjeti u odjeljku "Postavke" na slici 4.5 (3) što Oracleu omogućuje *izdvajanje strukture vanjske datoteke* – proces koji se zove "*parsing*", koji je preduvjet za mapiranje podataka u tablicu. Pomoću postavke "*Enclosed by*" početna i završna granica vrijednosti podataka može se ocrtati. Ako navedete znak za razgraničenje, Data Workshop zanemaruje razmake koji se pojavljuju prije početne i završne granice podatkovne vrijednosti. Također možete upotrijebiti ovu opciju za prilaganje podatkovne vrijednosti navedenom znaku razgraničenja.
4. **Prikaz uzorka za raščlanjene podatke.** Oracle automatski analizira podatke, kao što je već spomenuto, kako bi izdvojio strukturu podataka, otkrivajući, na primjer, odgovarajuće tipove podataka. Na dnu slike 4.5 (4), prikazan je mali uzorak (do 10 stupaca i 5 redaka) rezultata ovog procesa parsiranja.
5. **Prošireni pregled i daljnje konfiguracije.** Kada kliknete na gumb "*Pregled*" prikazuje se najviše 100 stupaca i 100 redaka i nude se dodatne mogućnosti konfiguracije, dopuštajući sada i promjenu automatski predloženih tipove podataka i odabrati koji od stupaca treba stvarno učitati. U slučaju da ste u koraku (1) odlučili učitati podatke u postojeću tablicu, tablicu morate odabrati s popisa dostupnih (usp. sliku 4.6 (2) i (3). Nakon toga, klikom na *"Konfiguriraj-gumb"* (usp. sliku 4.6 (4), moguće je *odrediti stupac preslikavanje*, tj. koji od automatski identificiranih stupaca učitanih podataka treba mapirati u koje stupce unutar DB tablice. Kao što se može vidjeti na slici 4.6 (5) i (6), stupac `DEPARTMENTNAME` koji ima tip podataka `VARCHAR2` koji je automatski ekstrahiran iz vanjske podatkovne datoteke tijekom raščlanjivanja sada je mapiran u stupac DB-tablice pod nazivom "DEPARTMENTNAME" koji ima isti tip podataka.

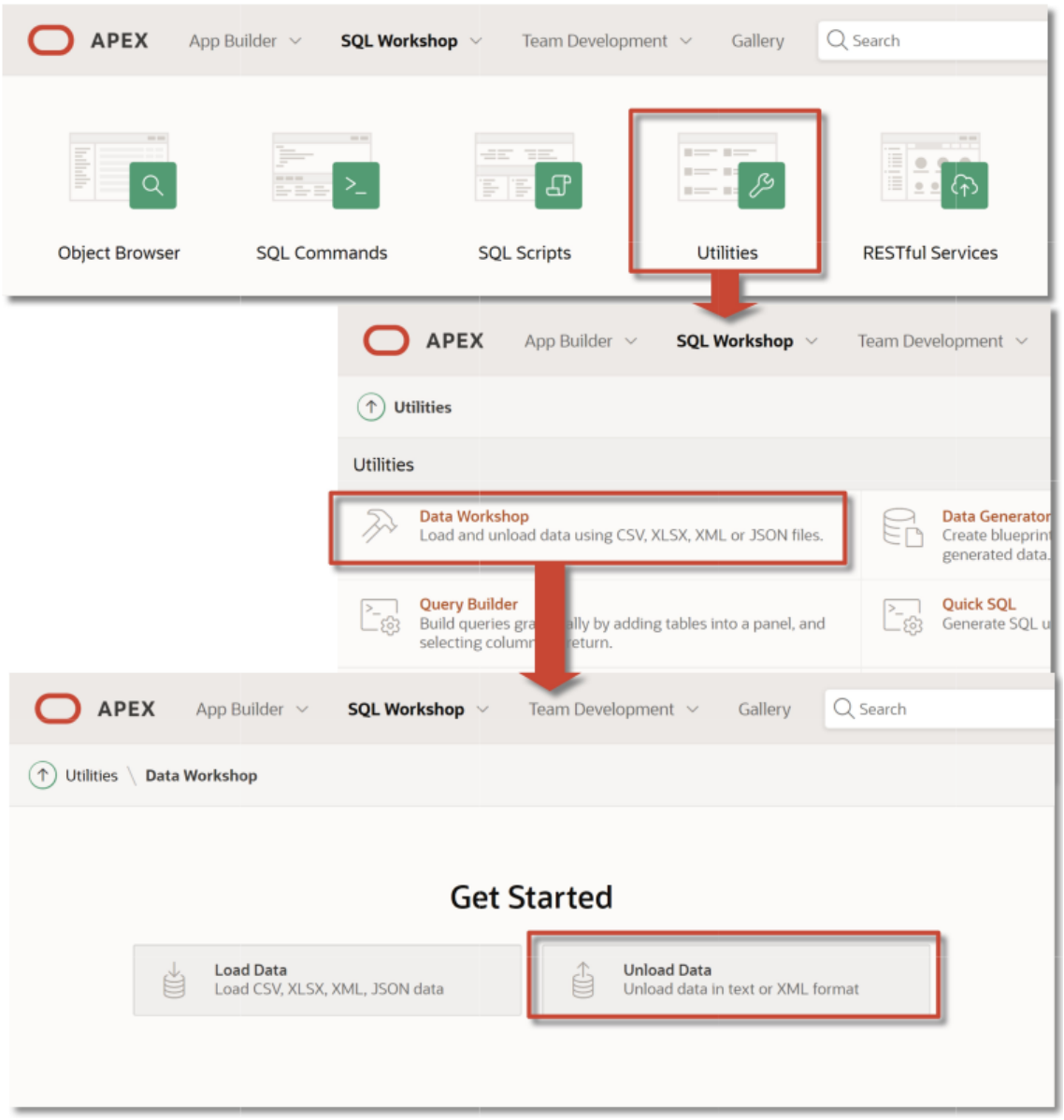


Slika 4.6: Učitavanje podataka u postojeću tablicu.

6. **Učitaj podatke.** Konačno, kada kliknete na *"Učitaj podatke"*-gumb (usp. Slika 4.6 (8)), nova DB-tablica je generira (u slučaju da je ova opcija odabrana u (1)) i podaci se učitavaju u novu ili postojeću tablicu, pri čemu dijaloški okvir za učitavanje obavještava koliko je redaka učitano. Proces učitavanja podataka radi u pozadini, što je korisno ako se učitava veća datoteka, budući da se dijaloški okvir može odbaciti. Tijekom ovog procesa učitavanja također može biti slučaj da se naiđe na neke *pogrešne redove* koji se ne mogu umetnuti u ciljnu tablicu, budući da je, npr., jedna ili druga vrijednost u sukobu s nekim tipom podataka. U ovom slučaju, ovi se pogrešni redovi automatski spremaju u *tablicu pogrešaka* i mogu se naknadno ručno obraditi. Dijaloški okvir za učitavanje također vas obavještava o eventualnim pogrešnim redovima. Rezultirajuća tablica (i eventualna tablica grešaka) sada se može vidjeti navigacijom do "Object Browser" ili se može odmah nastaviti s razvojem web aplikacije.

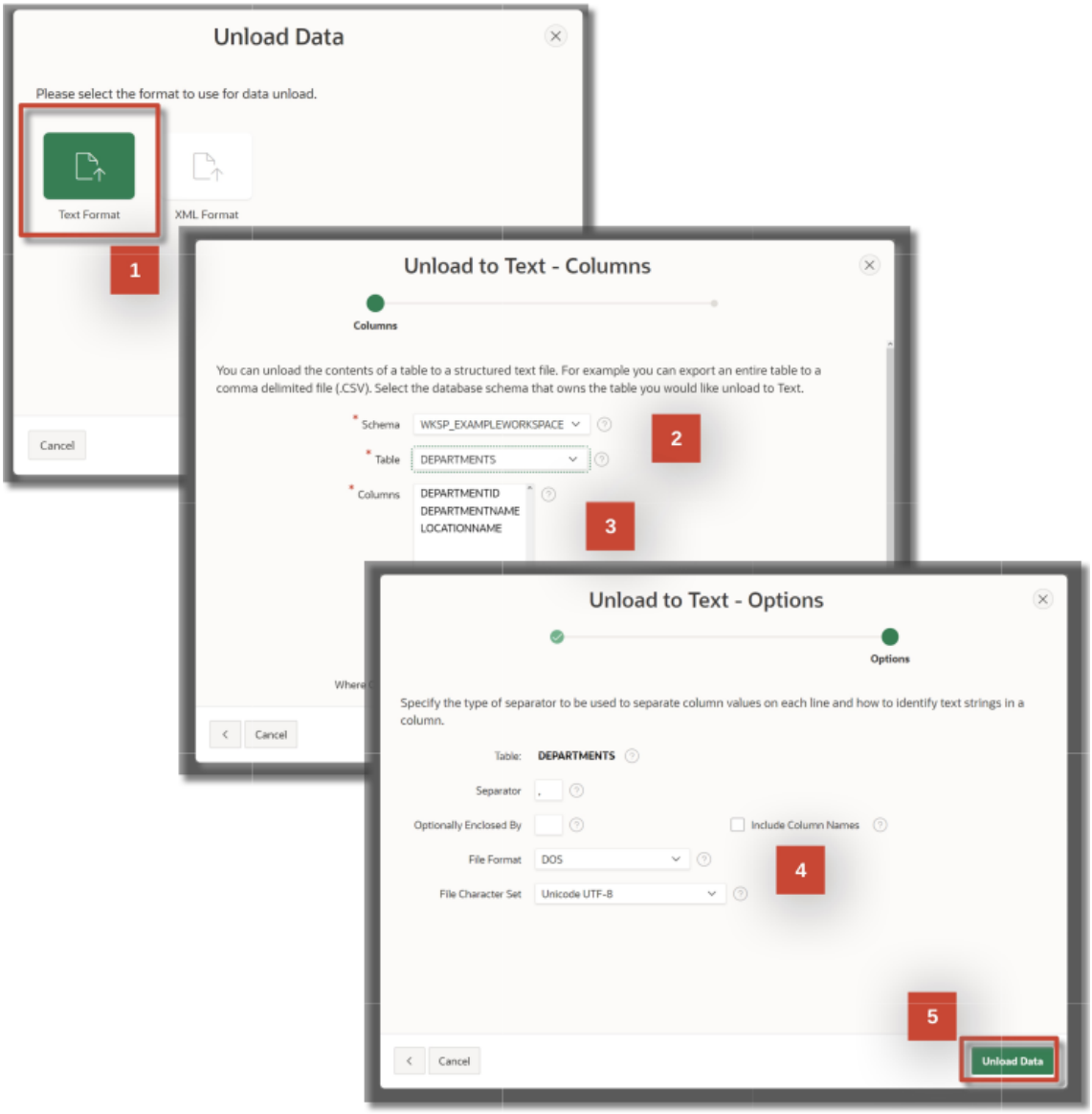
4.1.3 Izvoz podataka

"Export Wizard" Data Workshopa može se pristupiti na isti način kao što je opisano u odjeljku 4.1.1 za "Čarobnjak za učitavanje" (usp. sliku 4.7).



Slika 4.7: Izvoz ("istovar") podataka

U prvom koraku potrebno je odabrati format datoteke, pri čemu Data Workshop omogućuje izvoz podataka u vanjske datoteke koje su formatirane u smislu CSV ili XML (usp. sliku 4.8 (1)). Sljedeći korak je odabir tablice kao i odgovarajućih stupaca čije podatke treba eksportirati (usp. sliku 4.8 (2) odnosno (3). U slučaju da je XML odabran kao izvozni format, sada se može pokrenuti stvarni izvoz. U slučaju CSV-a, mogu se odabrati neke dodatne opcije, određujući razgraničenje između različitih redaka, trebaju li imena redaka biti uključena u izlaznu datoteku i treba li format podataka biti DOS ili UNIX (usp. Slika 4.8 (4)), prije nego što konačno aktivirate "Unload Data" za spremanje izvoza podataka (usp. slika 4.8 (5)).

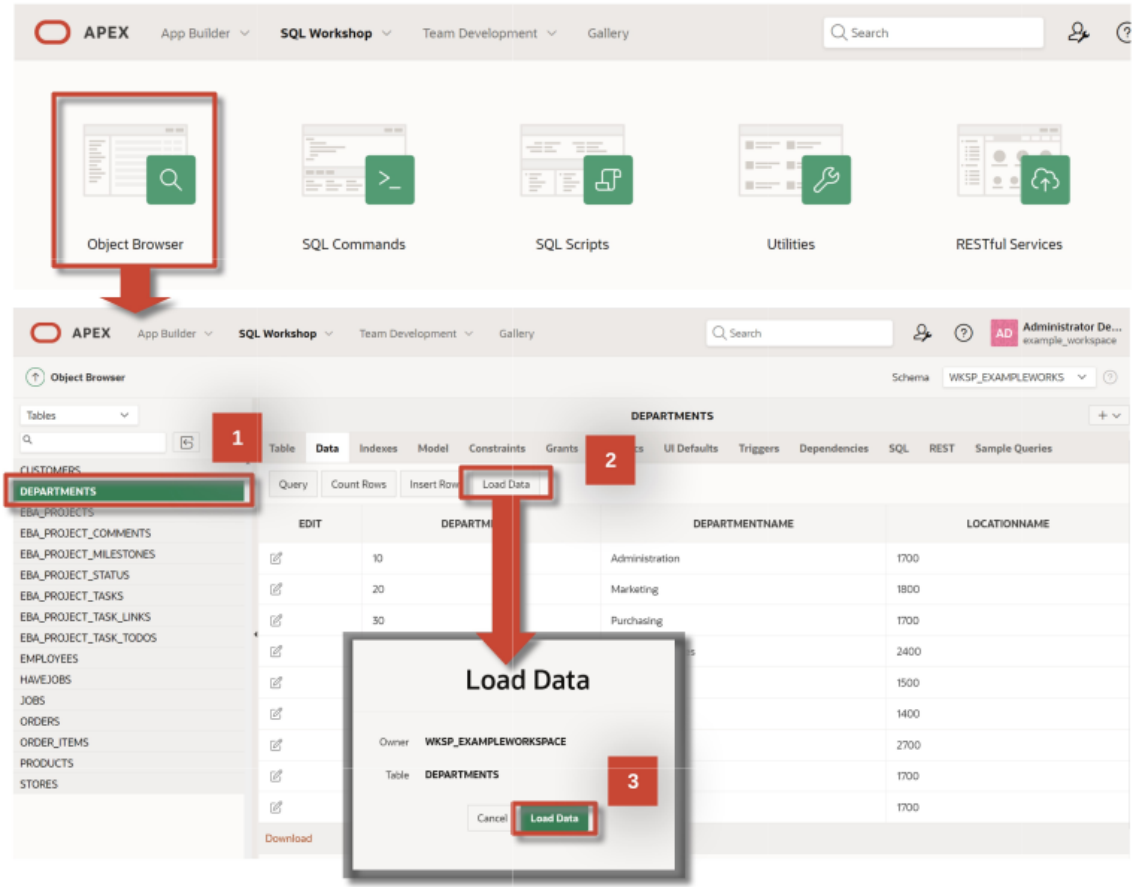


Slika 4.8: Čarobnjak za uklanjanje podataka

Što se tiče DOS-a naspram UNIX-a, imajte na umu da postoji suptilna razlika između ove dvije opcije: DOS datoteke imaju različite završetke redaka od datoteka stvorenih na Unix/Linuxu. DOS koristi početak reda i pomicanje retka ("`\r\n`") kao završetak retka, dok Unix koristi samo pomicanje retka ("`\n`"). Stoga morate biti oprezni kada prenosite datoteke između Windows strojeva i Unix strojeva kako biste bili sigurni da su završeci redaka pravilno postavljeni.

4.2 Uvoz i izvoz podataka pomoću "Object Browser-a"

Druga alternativa je korištenje "Object Browser-a" iz SQL Workshopa za uvoz podataka iz i izvoz podataka u datoteku, pri čemu se to može učiniti na tablični način. Stoga, prije svega, treba odabrati odgovarajuću tablicu (usp. sliku 4.9 (1)).

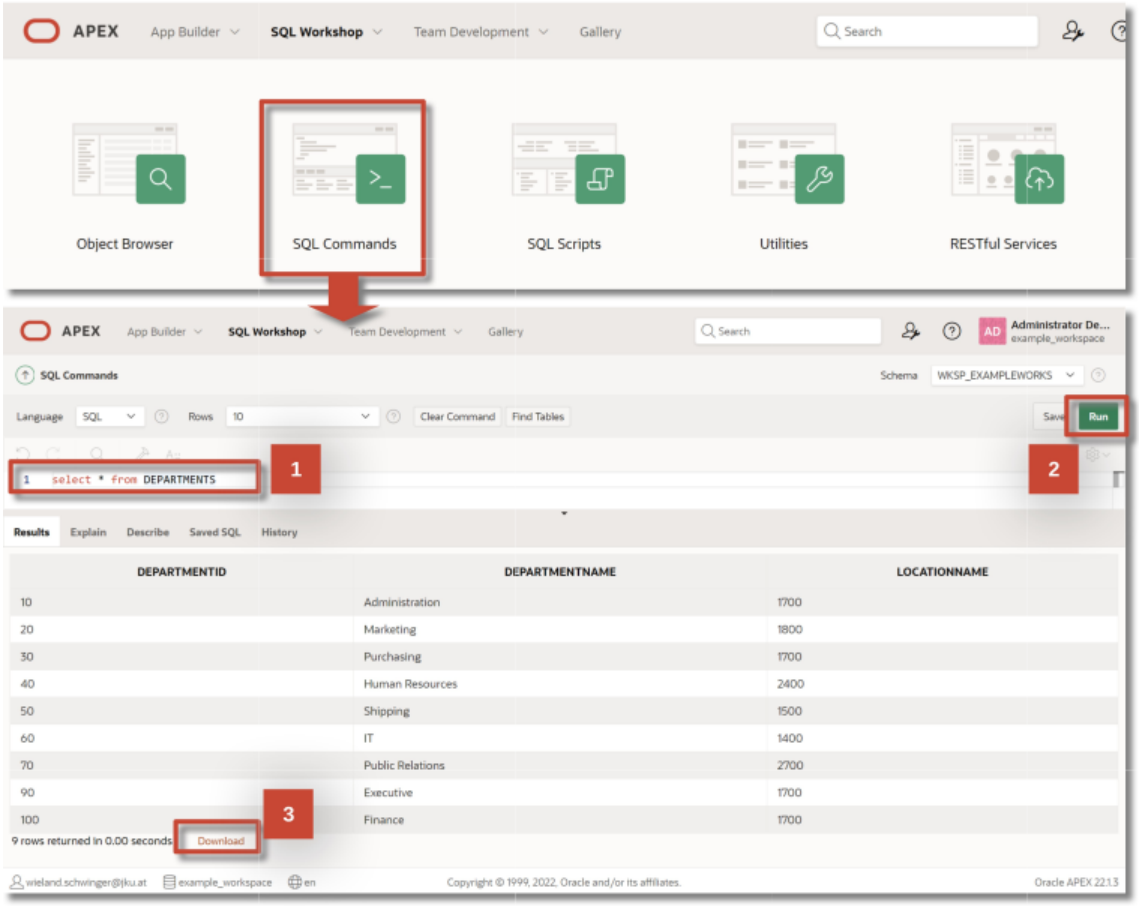


Slika 4.9: Uvoz/izvoz tabličnih podataka pomoću preglednika objekata

Ako sada podatke treba *uvesti*, nakon klika na *"gumb Učitaj podatke"*, pojavit će se isti dijalog kao kod odabira "Učitaj podatke" (usp. sliku 4.9 (2), (3)) kao što je već opisano u odjeljku 4.1.1 - također su podržani isti formati datoteka. S druge strane, *izvoz* podataka može se jednostavno izvršiti klikom na *"Download-button"* na dnu stranice, što dovodi do trenutnog generiranja CSV datoteke koja sadrži podatke DB-tablice.

4.3 Izvoz rezultata SQL-naredbe

Slično izvozu cjelokupnih podataka pohranjenih u tablici, također se određeni podskup tih podataka – ili čak podaci pohranjeni unutar nekoliko tablica – mogu izvesti odjednom. Jednostavno rečeno, skup rezultata proizvoljnog složenog SQL-upita može se izvesti (usp. Slika 4.10 (1))

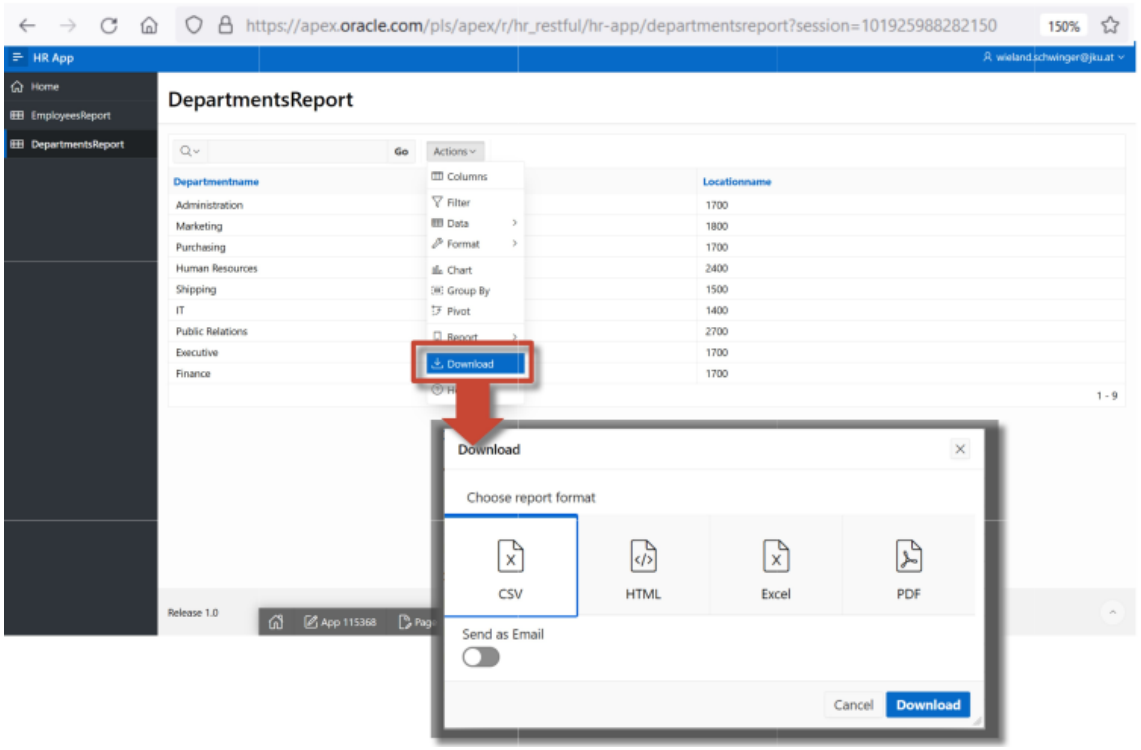


Slika 4.10: Izvoz podataka kao rezultat SQL-upita

Zato, nakon što je upit izvršen unutar alata “SQL command” SQL Workshopa i prikazan je skup rezultata (usp. Slika 4.10 (2)), ovaj skup rezultata možete preuzeti samo klikom na gumb "Preuzmi" (usp. Slika 4.10 (3)).

4.4 Izvoz podataka iz izvještaja u aplikaciji

Kao što smo raspravljali u 3. poglavlju, razvoj web-aplikacije pomoću APEX-ovog App Buildera zahtijeva odabir određene vrste stranice za svaku stranicu koja se dodaje web-aplikaciji. U slučaju da ste odabrali npr. vrstu stranice "Interaktivno izvješće", podaci koje ovo izvješće prikazuje tijekom izvođenja mogu se izvesti. Ovo je ilustrirano na slici 4.11, koja prikazuje oglednu stranicu interaktivnog izvješća "DepartmentsReport".

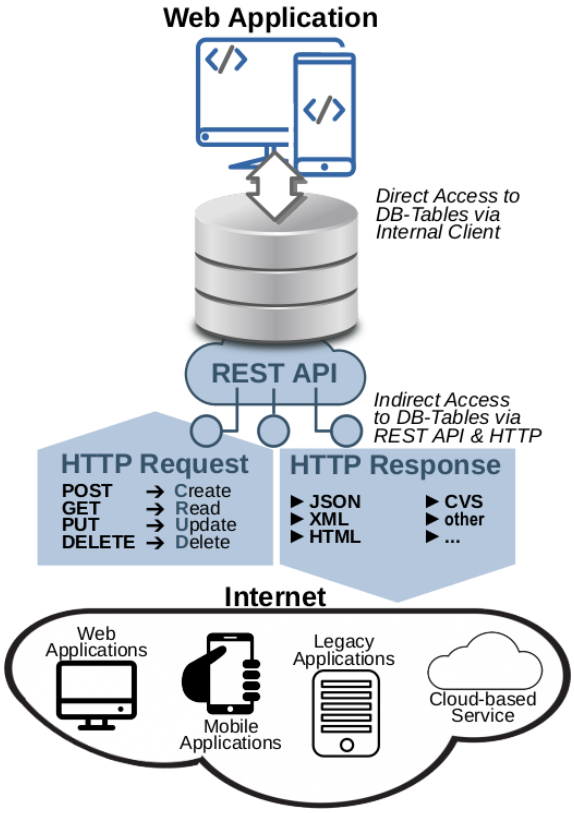


Slika 4.11: Izvoz podataka iz izvještaja u aplikaciji

Mogući formati datoteka su CSV, HTML, PDF i Excel, a opcija je i slanje podataka putem e-pošte.

4.5 Omogućavanje razmjene podataka putem RESTful servisa

Razmjena podataka DB-sloja naše web aplikacije omogućavanjem vanjskih klijenata kao što su druge web aplikacije, mobilne aplikacije, naslijeđene aplikacije ili sustavi u oblaku online pristup našim podacima može se lako ostvariti korištenjem takozvanih *RESTful servisa*. *REST (Representational State Transfer)* je zapravo *arhitektonski obrazac* koji su predložili Roy Fielding i Richard Taylor 2000. [2] za pružanje interoperabilnosti između proizvoljnih sustava preko interneta. Omogućuje postavljanje upita i manipulaciju podacima *bez potrebe za izravnim pristupom* temeljnim DB-tablicama. Slika 4.12 daje pregled na visokoj razini osnovnog arhitektonskog uzorka REST-a.



Slika 4.12: RESTful servisi za razmjenu podataka – osnovna arhitektura

Za realizaciju takvog *indirektnog* pristupa prema REST arhitektonskom obrascu, RESTful servisi moraju biti kreirani u smislu skupa *API-ja (Application Programming Interfaces)* na vrhu vaših DB-tablica. Bez izravnog povezivanja s osnovnom bazom podataka, ovi API-ji omogućuju vanjskim sustavima sigurnu interakciju s podacima postavljanjem upita, umetanjem, ažuriranjem ili brisanjem podataka. Definicije RESTful usluga stvorenih unutar Oracle APEX-a pohranjene su u *Oracle REST Data Services (ORDS)* repozitorij i nazivaju se *REST usluge temeljene na ORDS-u* .

4.5.1 Arhitektonski obrazac REST

Kao što je već spomenuto, usluga se opisuje kao RESTful kada je u skladu s načelima REST-a. Iako je potpuna rasprava o REST-u izvan opsega ovog poglavlja, RESTful servis ima sljedeće karakteristike (imajte na umu da su opće karakteristike REST-a kako je opisano u [2] malo prilagođene našem kontekstu baze podataka):

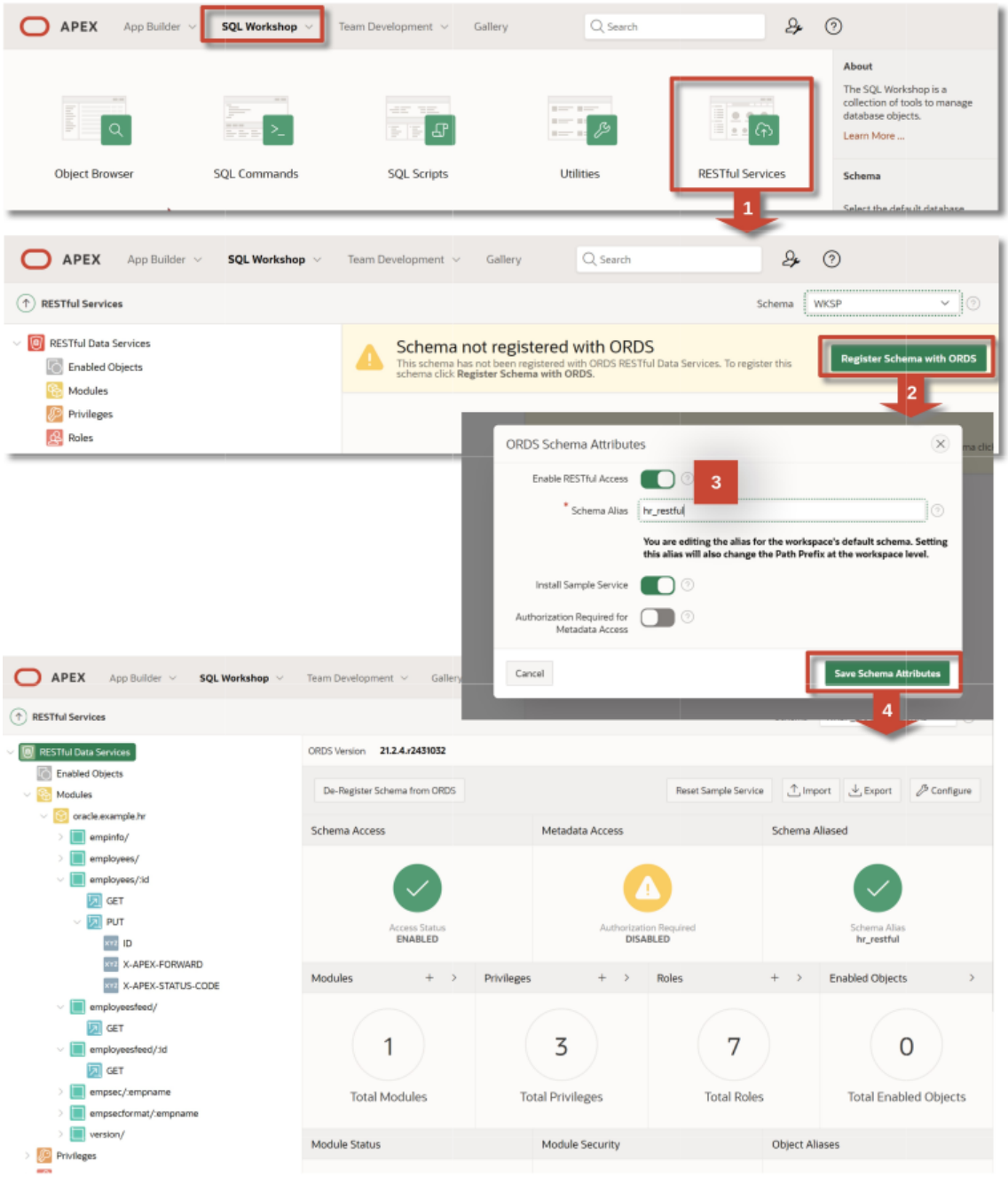
1. **DB-tablice pružene kao resursi za usluge**. RESTful servis je modeliran kao skup resursa, u našem slučaju na vrhu vaših DB-tablica. Ovi resursi su identificirani URL-ovima i pristupa im se preko `HTTP` ili `HTTPS` web protokola.
2. **Operacije poslane putem HTTP-zahtjeva**. Mali skup operacija koristi se za obradu ovih resursa u smislu `HTTP`-zahtjeva koji sadrži `POST`, `GET`, `PUT`, `DELETE`, čime nalikuje CRUD-operacijama (Stvaranje, Čitanje, Ažuriranje, Brisanje).
3. **Servisi su bez stanja**. RESTful servisi su *bez stanja*, tj. klijentski kontekst nije pohranjen na poslužitelju između zahtjeva – stanje sesije u potpunosti se čuva na klijentu. To znači da svaki zahtjev od klijenta do poslužitelja mora sadržavati sve informacije potrebne poslužitelju za razumijevanje zahtjeva i ne može iskoristiti pohranjeni kontekst na poslužitelju. Glavna prednost je da se mogu kreirati višekratno upotrebljive i skalabilne usluge kojima se može upravljati i ažurirati bez utjecaja na sustav, čak i tijekom rada. Nedostatak je taj što može smanjiti izvedbu mreže povećanjem ponavljajućih podataka (preopterećenje po interakciji)

poslanih u nizu zahtjeva, budući da se ti podaci ne mogu ostaviti na poslužitelju u zajedničkom kontekstu.

4. **HTTP-odgovori**. Zahtjevi za RESTful servis uvijek pokreću *odgovor*. Taj je odgovor u obliku XML-a, JSON-a, HTML-a ili nekog drugog definiranog formata. Odgovori daju pojedinosti o nekoj vrsti izmjene temeljnih podataka, poruke o pogrešci i hipertekstualne veze na druge srodne resurse, ovisno o operaciji.

4.5.2 Omogućavanje DB sheme za RESTful pristup

Da biste omogućili DB shemu za RESTful pristup, ona mora biti prvo registrirana u ORDS . To se može jednostavno učiniti kao što je ilustrirano na slici 4.13 pristupom alatu za usluge RESTful unutar SQL Workshopa (1), zatim klikom na gumb "Registriraj shemu s ORDS" (2), unosom proizvoljnog nadimka (alias) korištenog u formiranju URL-a koji upućuje na bilo koju RESTful uslugu unutar sheme (3) i konačno, nakon klika na gumb *"Spremi atribute sheme"* (4) pojavljuje se nadzorna ploča ORDS RESTful usluga, koja se koristi za upravljanje svim RESTful servisima.



Slika 4.13: Omogućavanje DB sheme za RESTful pristup

U slučaju da je u koraku (3) opcija “Instaliraj ogledni servis” bila omogućena, instaliran je i uzorak *module* `oracle.example.hr` koji daje primjere *resource templates* i *rukovatelji resursima* (usp. Odjeljak 4.5.3) koji implementiraju nekoliko različitih operacija za dohvaćanje i prikaz informacija o zaposlenicima iz tablice zaposlenika kao što se može vidjeti na slici 4.15.

4.5.3 Moduli resursa | Predlošci | Rukovatelji

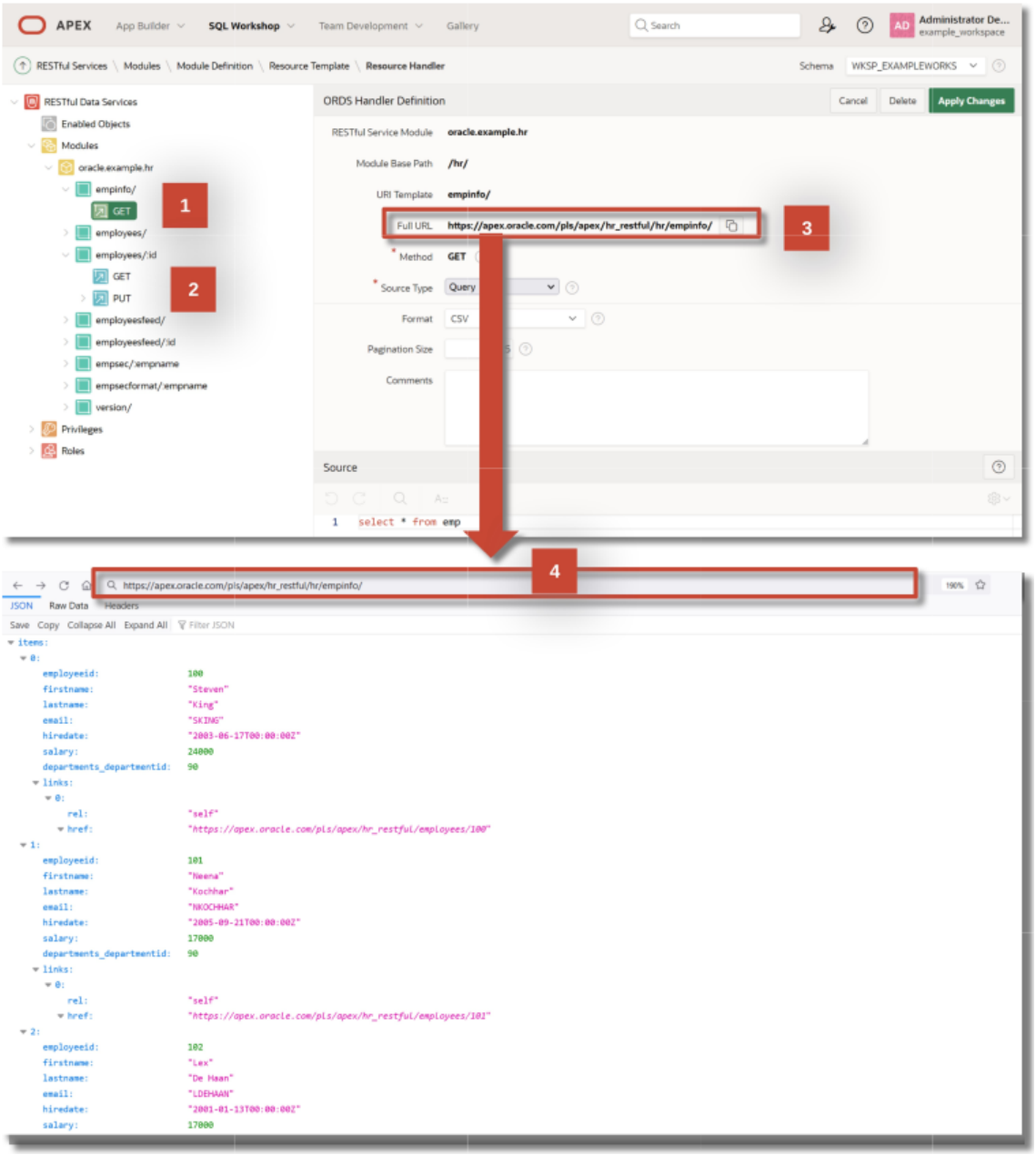
U ORDS-u, RESTful usluge mogu se stvoriti korištenjem tri osnovne komponente, koje se sastoje od modula resursa, predložaka resursa i rukovatelja resursima (usp. Slika 4.15 za neke primjere). Sveukupni odnosi između ovih osnovnih gradivnih blokova ilustrirani su ER-dijagramom na slici 4.14.



Slika 4.14: Odnosi između različitih komponenti ORDS RESTful usluga

- **Moduli resursa.** Modul resursa nije ništa drugo nego spremnik koji grupira skup povezanih RESTful servisa zajedno, na temelju određene *REST-omogućene DB sheme*. DB shema može sadržavati nekoliko modula resursa. Modul resursa ne samo da pruža način za jedinstvenu identifikaciju grupe, već također definira jedinstvenu osnovnu stazu koja se koristi na URI *za pristup skupu servisa* definiranih unutar modula. Na primjer, modul resursa koji vam omogućuje pristup informacijama o zaposlenicima zove se `oracle.example.hr`, a vrijednost osnovne staze za uslugu je `/hr/`, usp. Slika 4.15.
- **Predlošci resursa**. Predložak resursa definira pojedinačni servis koji se može pozvati, npr. `employees/:id`, slika 4.15. Predlošci resursa sadržani su u modulima resursa. Svaki predložak resursa definira *URI obrazac* gdje mu se može pristupiti, npr. "empinfo/" (slika 4.15) i sadrži najmanje jedan rukovatelj resursima (usp. dolje).
- **Rukovatelji resursima.** Svaki rukovatelj resursima implementira jednu (i samo jednu) od četiri različite HTTP operacije, koje nalikuju tradicionalnim CRUD operacijama (Stvaranje, čitanje, ažuriranje, brisanje) .
  - `POST` stvara novi resurs ili dodaje resurs u kolekciju, što je ekvivalentno SQL `INSERT` iskazu.
  - `GET` dohvaća reprezentaciju resursa, ekvivalentnu SQL `SELECT` naredbi.
  - `PUT` ažurira vrijednosti postojećeg resursa, ekvivalentno SQL `UPDATE` naredbi.
  - `DELETE` briše postojeći resurs, ekvivalentno SQL `DELETE` naredbi.

Morate definirati rukovatelja resursima za svaku operaciju povezanu s istim predloškom resursa. Na primjer, da biste pružili *operaciju za vraćanje podataka* i *drugu operaciju za pohranu podataka*, morate definirati rukovatelja resursima za svaku operaciju (usp. sl. 4.15 – za predložak resursa `employees/:id`. definirane su dvije operacije `GET` (1) i `PUT` (2).



Slika 4.15: Primjer RESTful usluga za tablicu Employee

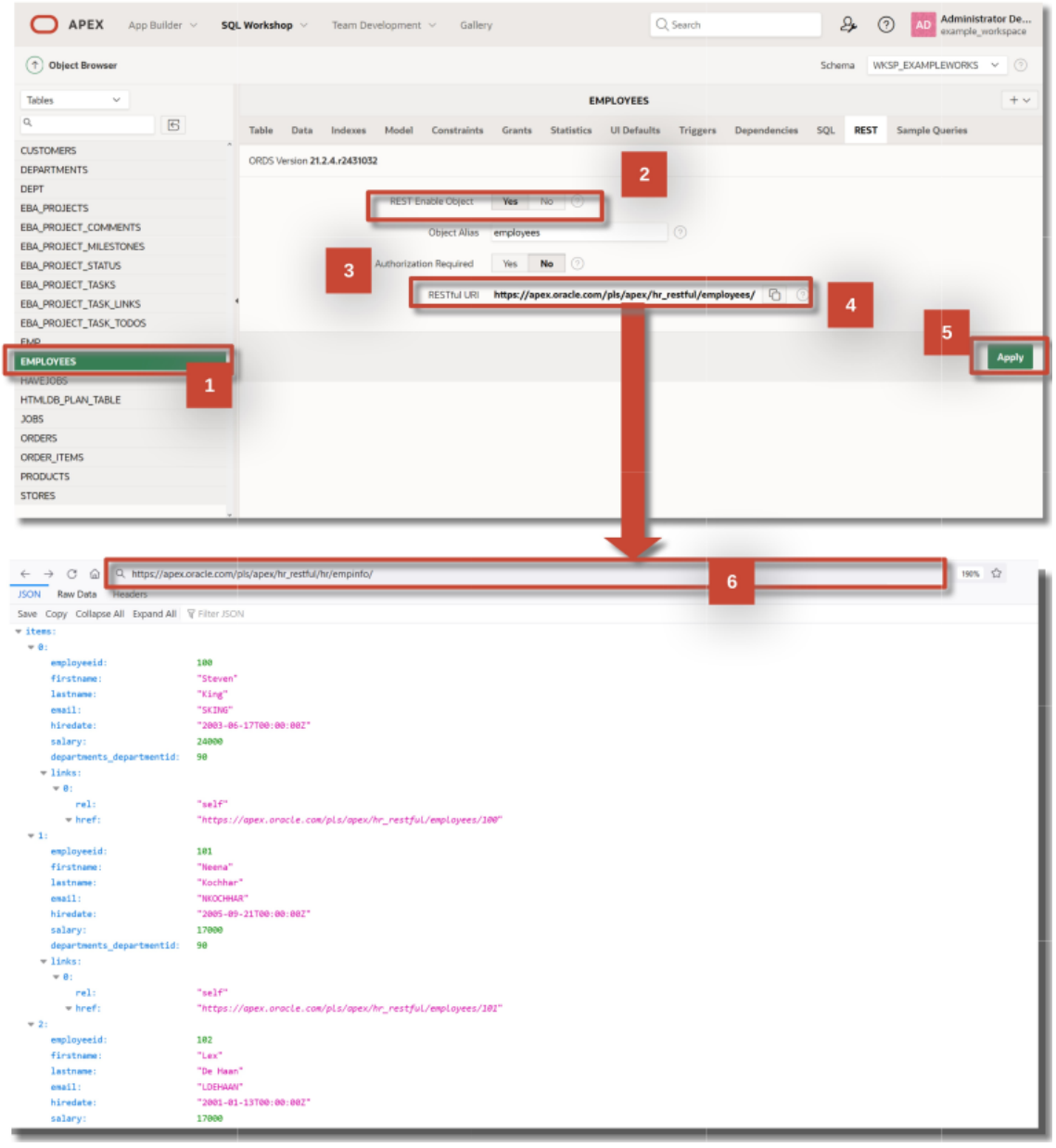
Ukratko, slika 4.15 daje primjeran pregled različitih sastojaka RESTful usluge u smislu modula resursa, predložaka i rukovatelja, uključujući i odabir željenog formata odgovora (CSV) i SQL upit koji implementira vrstu operacije Rukovatelja resursima. Nakon što stvorite RESTful servis, možete je testirati jednostavnim unosom generiranog pristupnog URL-a u vaš preglednik (usp. sl. 4.15 (3) odnosno (4).

4.5.4 Korištenje "AutoREST" umjesto ručnog definiranja resursa

Kao što se moglo vidjeti u prethodnom odjeljku, za svaku vrstu operacije baze podataka koju treba ponuditi vanjskim klijentima kao RESTful uslugu za neizravni pristup našoj bazi podataka, potrebno je nekoliko koraka. Za najjednostavniji oblik upita, tj. skeniranje cijele tablice, postoji, međutim, i neka vrsta "prečaca" koji pruža APEX u smislu takozvane *"AutoREST"-facility*. AutoREST je brz i jednostavan način za izlaganje DB tablica kao REST resursa. Gubite nešto fleksibilnosti i

mogućnosti prilagođavanja ako koristite značajku AutoREST, ali ona u značajnoj mjeri smanjuje vaše vrijeme i trud. AutoRest vam omogućuje brzo izlaganje podataka, ali dopušta samo postavljanje upita cijeloj tablici (tj. skeniranje cijele tablice), uz fiksni izlazni format samo u smislu JSON-a.

AutoREST se može jednostavno koristiti na sljedeći način: Nakon što ste omogućili svoju shemu za RESTful pristup (vidi poglavlje 4.5.2), svaka DB tablica može se pojedinačno definirati kao resurs korištenjem Object Browsera SQL Workshopa kao što se može vidjeti na slici 4.16.



Slika 4.16: Omogućavanje usluge REST u pregledniku objekata

Nakon odabira odgovarajuće DB tablice (1), klikanjem kartice REST (2), odabirom potrebne autorizacije (3) i primjenom svih ovih postavki (5), REST je aktiviran za tablicu i automatski generirani pristupni URL pojavljuje se (4). Opet, ova REST usluga može se jednostavno testirati unosom URL-a u web preglednik (6).

4.6 Pitanja

1. Koji su slučajevi korištenja za uvoz podataka putem SQL Workshopa i App Buildera?
2. Koji su različiti načini izvoza podataka?
3. Čemu služi REST? Koja je njegova svrha / korist?

4.7 Odgovori

1. Data Workshop omogućuje uvoz podataka iz vanjskih datoteka putem "Čarobnjaka" i uvoz podataka u tablicu putem "Preglednika objekata". Uvoz podataka iz Data Workshopa također se može pokrenuti putem App Buildera.
2. Izvoz podataka u datoteku na tablični način podržava SQL Workshopov "Preglednik objekata". Data Workshop nudi "Čarobnjaka za izvoz" koji omogućuje odabir formata datoteke i tablica kao i odgovarajućih stupaca čije podatke treba izvesti.
3. RESTful usluge pružaju interoperabilnost između bilo kojeg sustava preko Interneta dopuštajući upite i manipuliranje podacima.



5. Kako generirati prvi nacrt aplikacije?

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

5.1 Zašto poduzeća trebaju aplikacije?

U današnjem poslovnom okruženju koje se brzo razvija, aplikacije su postale ključni alat za tvrtke da ostanu konkurentne i zadovolje stalno promjenjive zahtjeve svojih kupaca. Izrada aplikacija omogućuje poduzećima da pojednostave svoje poslovanje, povećaju učinkovitost i poboljšaju korisničko iskustvo, što sve u konačnici može dovesti do veće profitabilnosti.

Jedan od primarnih razloga zašto tvrtke izrađuju aplikacije je automatizacija procesa koji se trenutno obavljaju ručno. Na primjer, tvrtka za e-trgovinu može razviti aplikaciju koja automatizira procese upravljanja zalihama i ispunjavanja narudžbi. Na taj način mogu smanjiti vjerojatnost ljudske pogreške i poboljšati brzinu i točnost svojih operacija.

Još jedan ključni razlog zašto tvrtke izrađuju aplikacije je poboljšanje korisničkog iskustva. Aplikacije se mogu razviti kako bi kupcima pružile opcije samoposluživanja, kao što je mogućnost praćenja njihovih narudžbi ili ažuriranja podataka o njihovom računu. To može pomoći smanjiti vrijeme čekanja korisnika i povećati zadovoljstvo korisnika.

Osim toga, aplikacije mogu pomoći tvrtkama prikupiti vrijedne podatke o svojim klijentima, koji se mogu koristiti za poboljšanje njihovih marketinških strategija i razvoj ciljanih reklamnih kampanja. Na primjer, trgovac na malo može razviti aplikaciju koja prati obrasce kupnje kupaca, dopuštajući im slanje personaliziranih preporuka i promocija svojim kupcima.

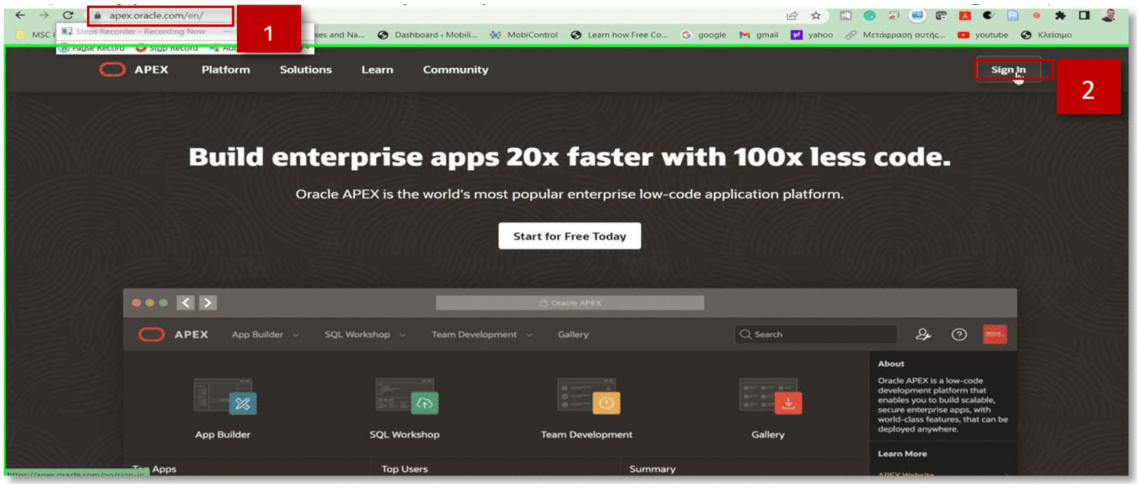
Sve u svemu, izrada aplikacija postala je snažna poslovna potreba jer tvrtkama omogućuje učinkovitije poslovanje, pružanje boljih korisničkih iskustava i stjecanje uvida u ponašanje svojih kupaca. Kako poslovni svijet postaje sve digitalniji, važnost izrade aplikacija samo će rasti.

5.2 Postavljanje ORACLE APEX okruženja

Cilj ovog primjera je upoznati vas s okruženjem ORACLE APEX, vodeći vas kroz postavljanje okruženja APEX, demonstrirajući primjer SQL radionice baze podataka zaposlenici-odjeli-projekti i ilustrirajući kako izgraditi APEX web aplikaciju temeljenu na podatkovnom modelu.

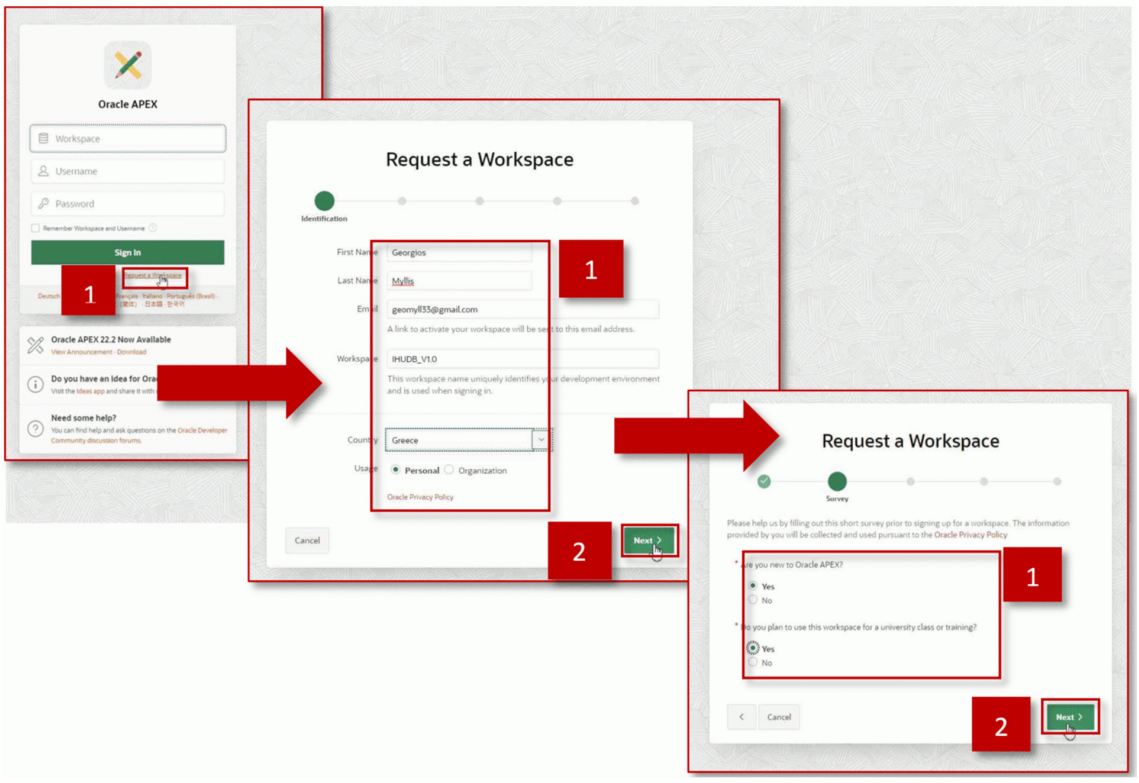
Za implementaciju ovog primjera slijedit će se sljedeći koraci:

Korak 1: Otvorite web preglednik i upišite "https://apex.oracle.com/en/". Klikni na **Sign In** gumb, kao što je prikazano na slici 5.1.



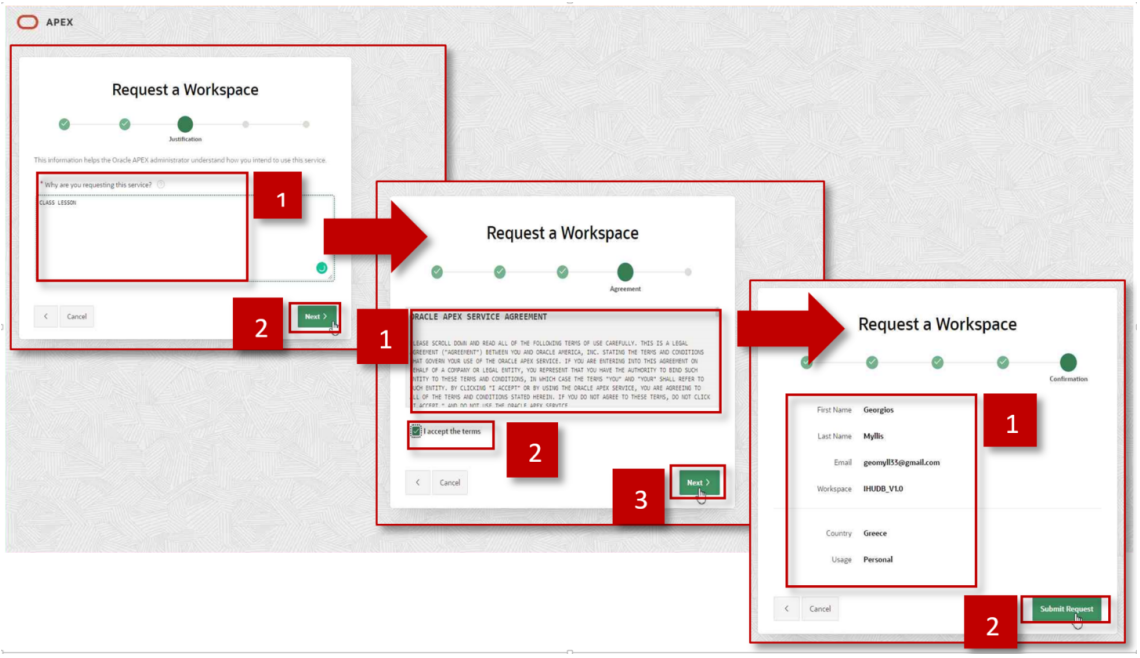
Slika 5.1: Upišite APEX adresu i prijavite se

Korak 2: Kliknite na **Request a Workspace** dugme. U skočnom prozoru koji se pojavi ispunite potrebna polja za identifikaciju i kliknite na **Next**. Zatim odgovorite na anketna pitanja u sljedećem skočnom prozoru i kliknite na **Next**, kako je prikazano na slici 5.2.



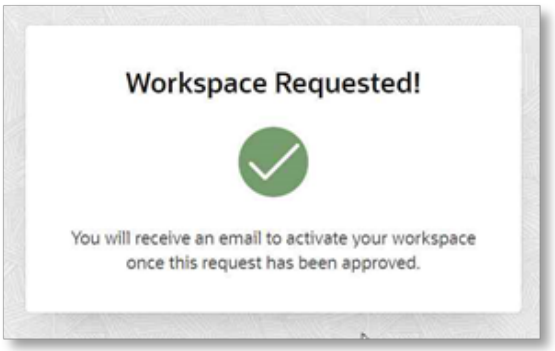
Slika 5.2: Provedite početne korake za dobivanje Oracle APEX Workspace

Korak 3: Odgovorite na tražene podatke za obrazloženje i kliknite na **Next**. U sljedećem skočnom prozoru pročitajte uvjete ugovora, a zatim ih prihvatite klikom na **Next**. U sljedećem skočnom prozoru provjerite ima li pogrešaka u detaljima koje ste unijeli, a zatim kliknite na **Submit request** kako je prikazano na slici 5.3.



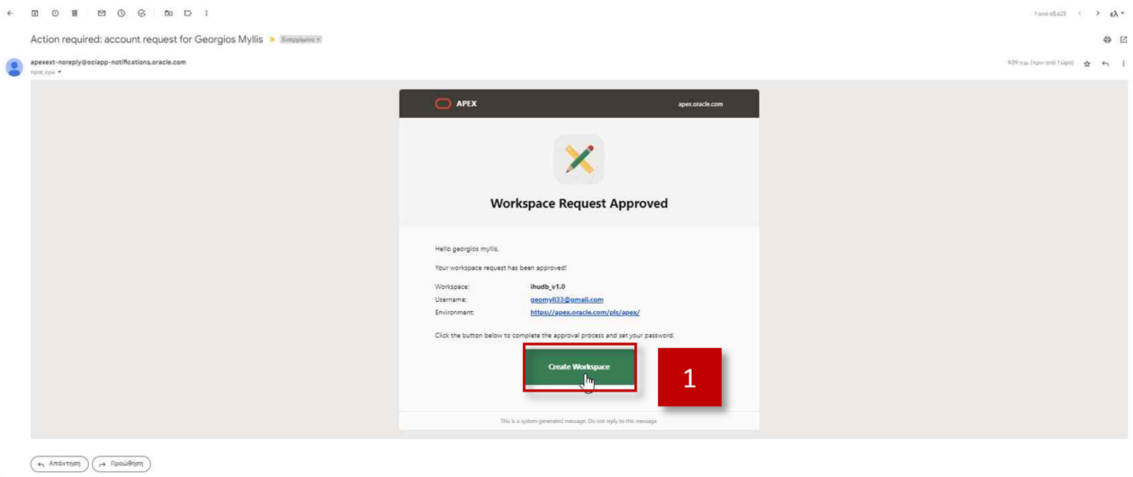
Slika 5.3: Zatražite dovršetak Oracle APEX Workspacea

Korak 4: Nakon što pošaljete zahtjev, pojavit će se skočni prozor koji potvrđuje da je vaš radni prostor zatražen, a aktivacijska e-pošta bit će poslana na adresu e-pošte koju ste unijeli, kao što je prikazano na slici 5.4.



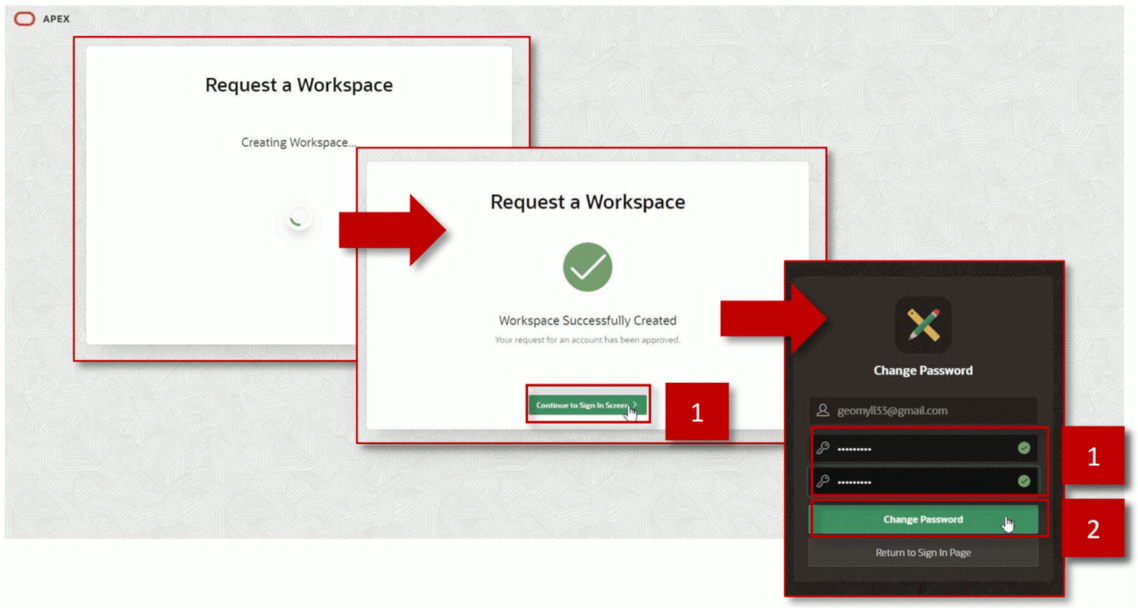
Slika 5.4: Oracle APEX Workspace odobren

Korak 5: Provjerite svoju e-poštu za poruku od APEX-a. Vaš zahtjev mora biti odobren prije nego što možete nastaviti. Nakon odobrenja kliknite na **Stvori radni prostor** kao što je prikazano na slici 5.5.



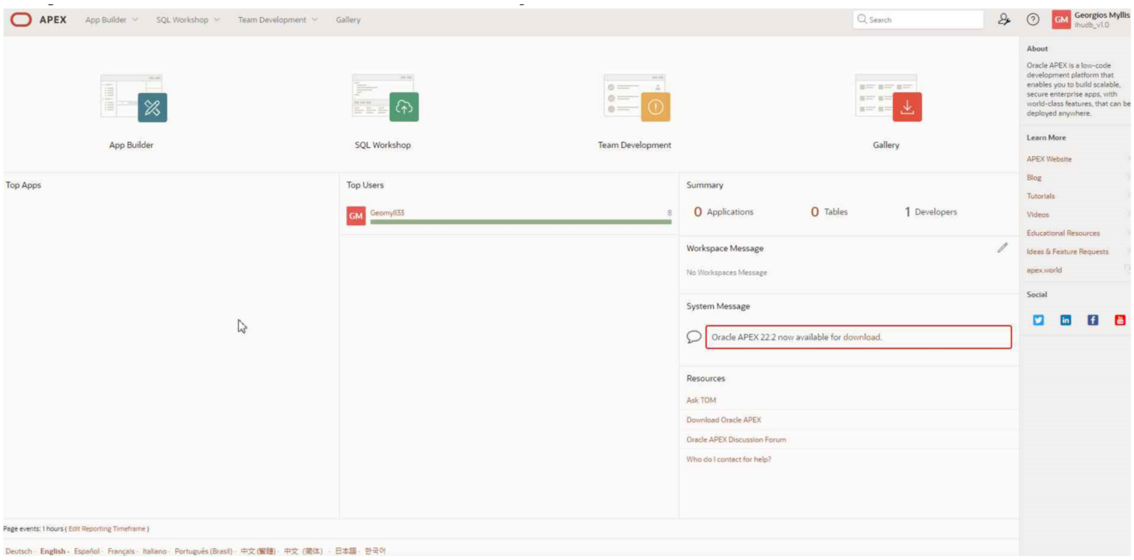
Slika 5.5: Aktivacijska e-pošta od APEX-a

Korak 6: Pričekajte da se završi proces stvaranja radnog prostora. Nakon dovršetka, skočni prozor će vas obavijestiti da je radni prostor uspješno kreiran. Kliknite na **Continue to Sign-in Screen**. Pojavit će se novi skočni prozor koji od vas traži da promijenite lozinku prije prijave. Postavite novu lozinku, potvrdite je i kliknite na **Change Password**, kako je prikazano na slici 5.6.



Slika 5.6: Postavite novu lozinku za Workspace APEX

Korak 7: Oracle APEX okruženje sada je spremno za korištenje, kao što je prikazano na slici 5.7.



Slika 5.7: Radni prostor APEX okruženje

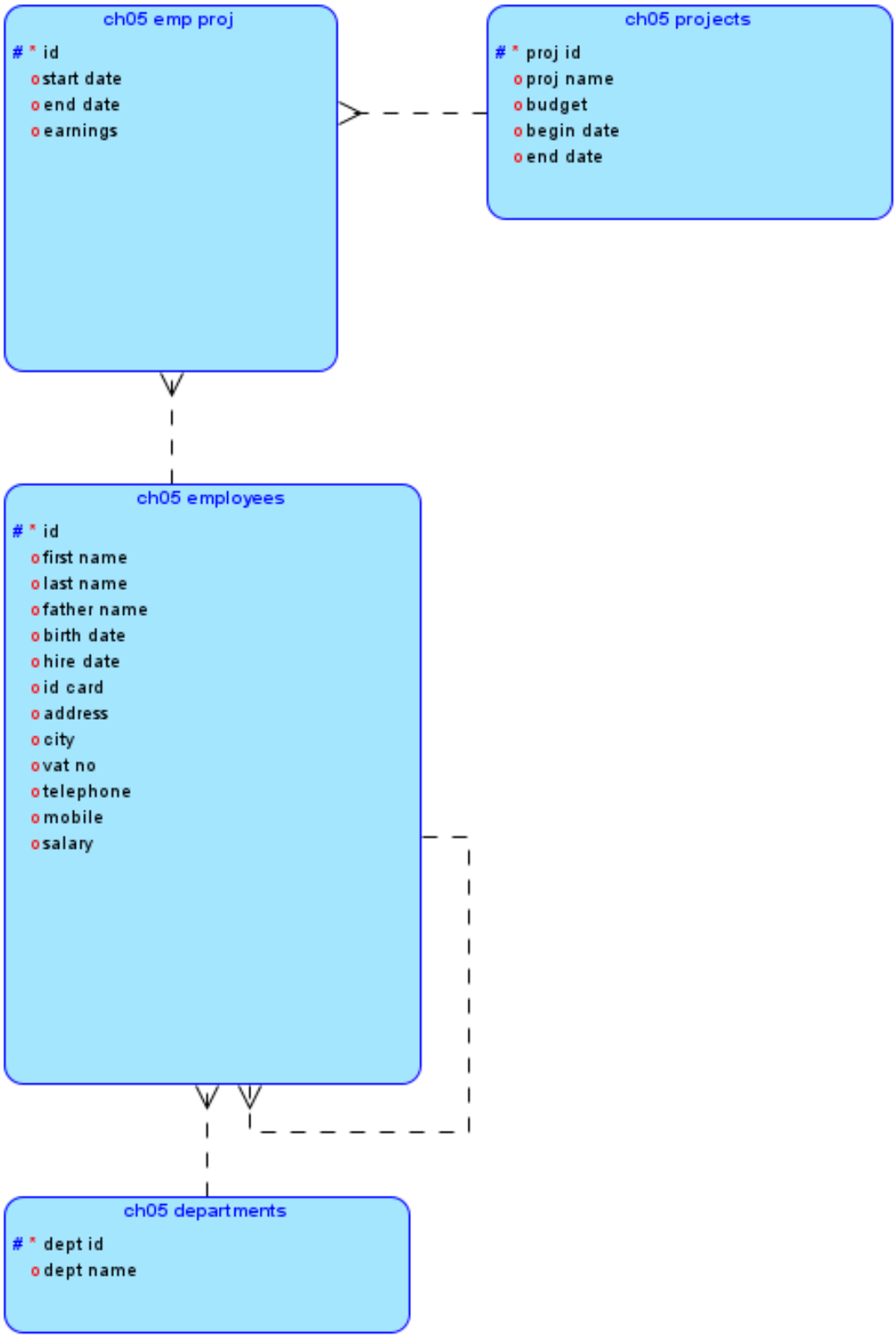
5.3 Razvoj primjera SQL radionice

Za izradu baze podataka u APEX okruženju dostupno je više pristupa. Jedna je mogućnost korištenje Object Browsera, grafičkog alata koji korisnicima omogućuje stvaranje tablica i odgovarajućih polja popunjavanjem unaprijed dizajniranog predloška.

Druga alternativa je korištenje SQL naredbi, koje korisnicima omogućuju stvaranje tablica i drugih objekata baze podataka izvršavanjem SQL naredbi. Ova metoda nudi veću fleksibilnost i kontrolu nad strukturom baze podataka, iako zahtijeva naprednije znanje.

S druge strane, Quick SQL je alat koji pruža Oracle APEX koji olakšava brzo generiranje SQL skripti za stvaranje tablica baze podataka, ograničenja i uzoraka podataka. Quick SQL pojednostavljuje proces stvaranja objekata baze podataka bez potrebe za ručnim pisanjem složenih SQL skripti.

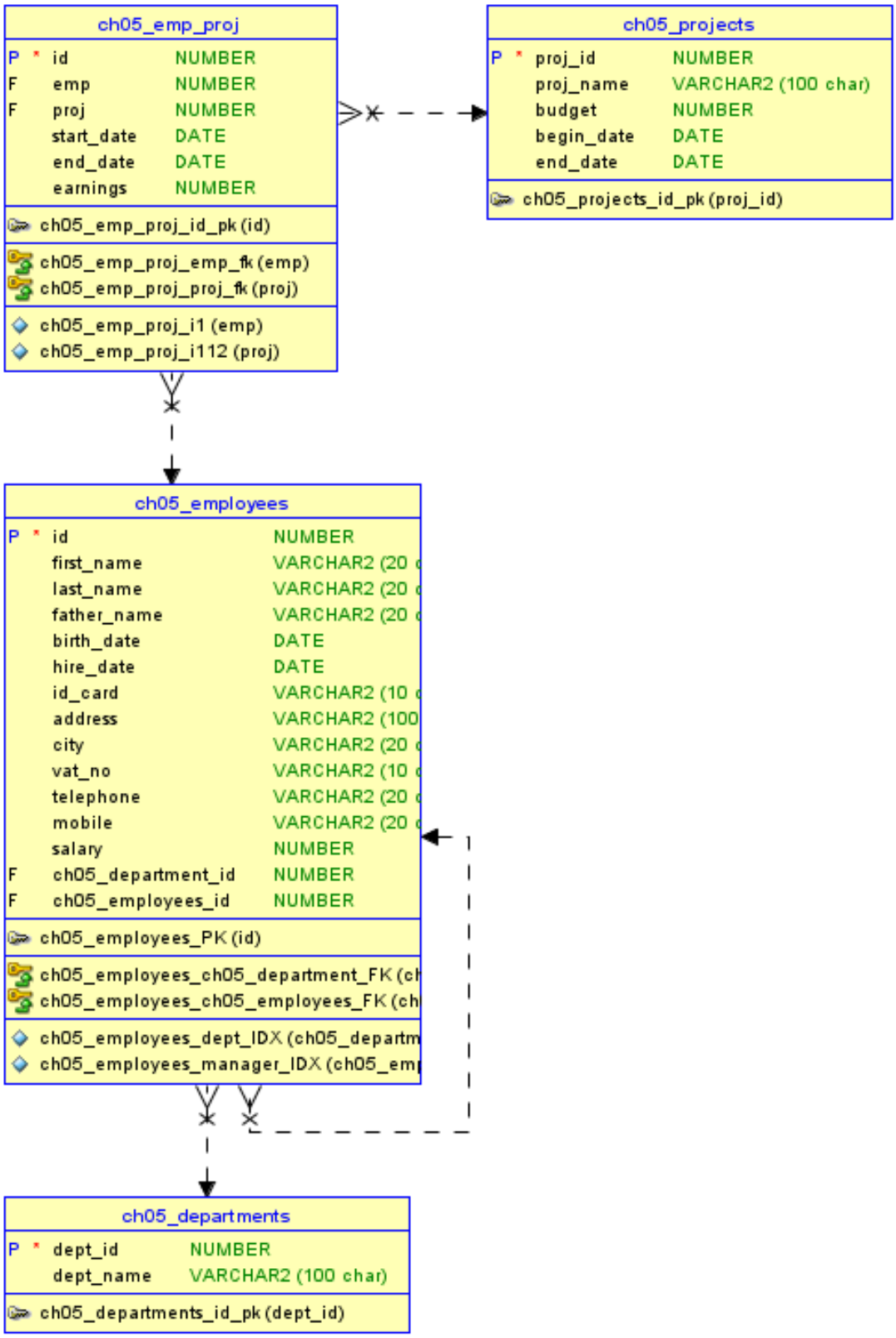
U ovom primjeru koristit ćemo metodu Quick SQL. Skripta je generirana na temelju logičkog modela prikazanog na slici 5.8.



Slika 5.8: Logički model našeg primjera vođenja HR-a

Prema relacijskom modelu izvedenom iz logičkog modela pomoću Oracle Data Modelera

(ODM), kao što je prikazano na slici 5.9, možemo nastaviti s implementacijom sljedeće strukture u bazu podataka.



Slika 5.9: Relacijski model našeg primjera Running HR

5.3.1 Quick SQL

Čini se da je pruženi kod skup uputa za generiranje oglednih podataka pomoću alata Quick SQL. Svaki odjeljak počinje nazivom tablice nakon kojeg slijedi ključna riječ "/insert" i broj koji označava željeni broj redaka koji se umeću u tablicu.

```
ch05_departments /insert 5
    dept_name vc100

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

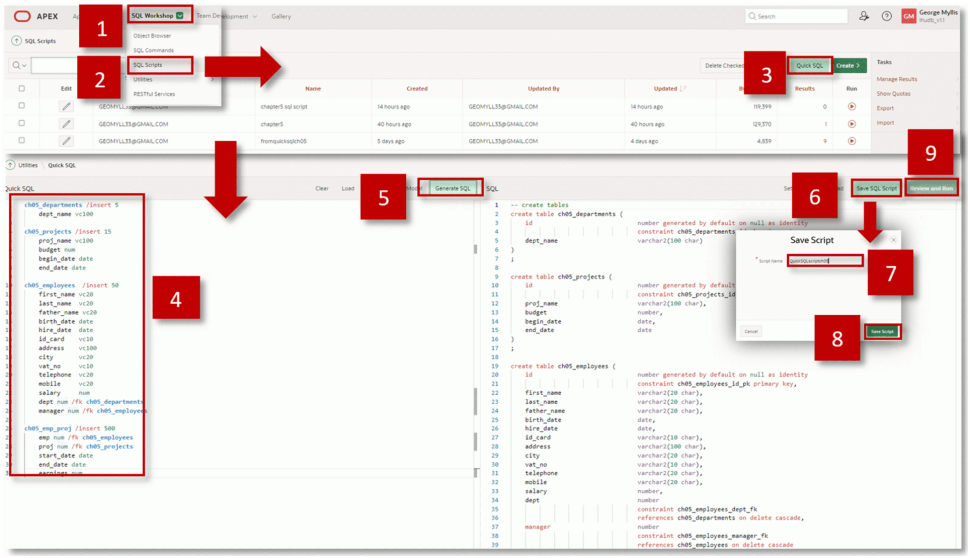
ch05_employees  /insert 50
    first_name vc20
    last_name  vc20
    father_name vc20
    birth_date date
    hire_date  date
    id_card    vc10
    address    vc100
    city       vc20
    vat_no     vc10
    telephone  vc20
    mobile     vc20
    salary     num
    dept num /fk ch05_departments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

Ove upute pružaju shemu i uzorak plana generiranja podataka za bazu podataka, koji se odnose na odjele, projekte, zaposlenike i njihove udruge.

Korak 1: Kao što je prikazano na slici 5.10, slijedite korake u nastavku:

1. Idite do **SQL Workshop** sekcija.
2. Odaberite **SQL Scripts** neku od mogućnosti.
3. Kliknite na **Quick SQL**.
4. Umetnite priloženi Quick SQL kod u predviđeno područje.
5. Kliknite na **Generate SQL**. Generirani SQL bit će prikazan u oknu Oracle SQL Output.
6. Kliknite na **Save SQL Script**.
7. Unesite naziv skripte.
8. Kliknite na **Save script**.
9. Nastavite klikom na **Review and Run**.



Slika 5.10: Umetnite Suick SQL kod u APEX radni prostor

5.3.2 SQL skripta

Kao što je prethodno spomenuto, ovo je SQL skripta generirana iz Quick SQL-a, koja stvara shemu baze podataka za primjer sustava upravljanja ljudskim resursima. Shema obuhvaća tablice za odjele, zaposlenike, projekte i odnose zaposlenika i projekta. Dodatno, uključuje indekse i ograničenja kako bi se zajamčila dosljednost podataka i referentni integritet.

```
-- 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);
```

Tablica ch05_departments pohranjuje informacije o odjelima, uključujući jedinstveni identifikator (id) i naziv (dept_name). Tablica ch05_employees pohranjuje informacije o zaposlenicima, uključujući jedinstveni identifikator (id), ime_ime, prezime_ime, očevo_ime, datum_rođenja, datum zapošljavanja_datum, broj osobne iskaznice, adresu, grad, PDV_broj , broj telefona, broj mobitela, plaću, te ID odjela (odjela) i voditelja (upravitelja). Tablica ch05_projects pohranjuje informacije o projektima, uključujući jedinstveni identifikator (id), naziv, proračun, početni_datum i završni_ datum. Tablica ch05_emp_proj pohranjuje informacije o projektnim odnosima zaposlenika, uključujući jedinstveni identifikator (id), ID-ove zaposlenika (emp) i projekta (proj), početni_datum, završni_datum i zaradu.

Indeksi se stvaraju u tablici ch05_emp_proj za stupce zaposlenika i projekta kako bi se poboljšala izvedba prilikom postavljanja upita tablici na temelju tih stupaca. Indeksi se također stvaraju u

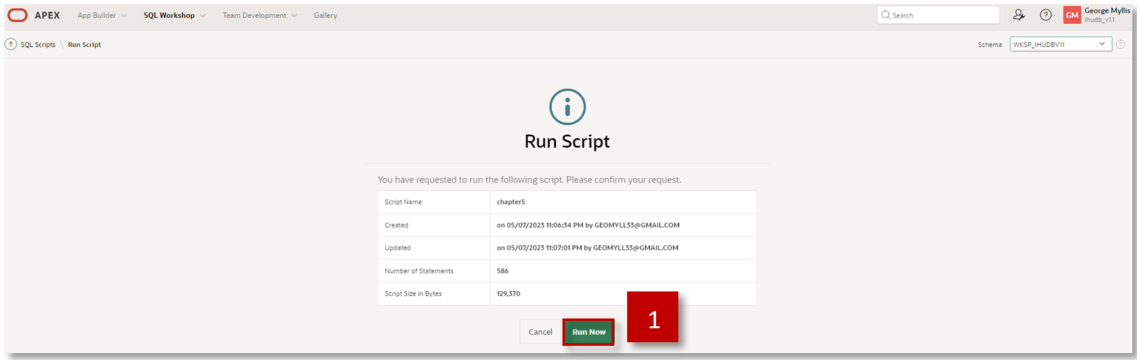
tablici ch05_employee za stupce odjela i upravitelja iz istog razloga.

Konačno, dodaju se ograničenja kako bi se osigurala dosljednost podataka i referentni integritet između tablica. Ograničenja ch05_emp_proj_emp_fk i ch05_emp_proj_proj_fk osiguravaju da stupci zaposlenika i projekta u tablici ch05_emp_proj upućuju na važeće ID vrijednosti u tablicama ch05_employees i ch05_project odnosno.

Ograničenja ch05_employees_dept_fk i ch05_employee_manager_fk osiguravaju da stupci odlaska i upravitelja u tablici ch05_employee upućuju na važeće ID vrijednosti u tablicama ch05_departments odnosno ch05_employees.

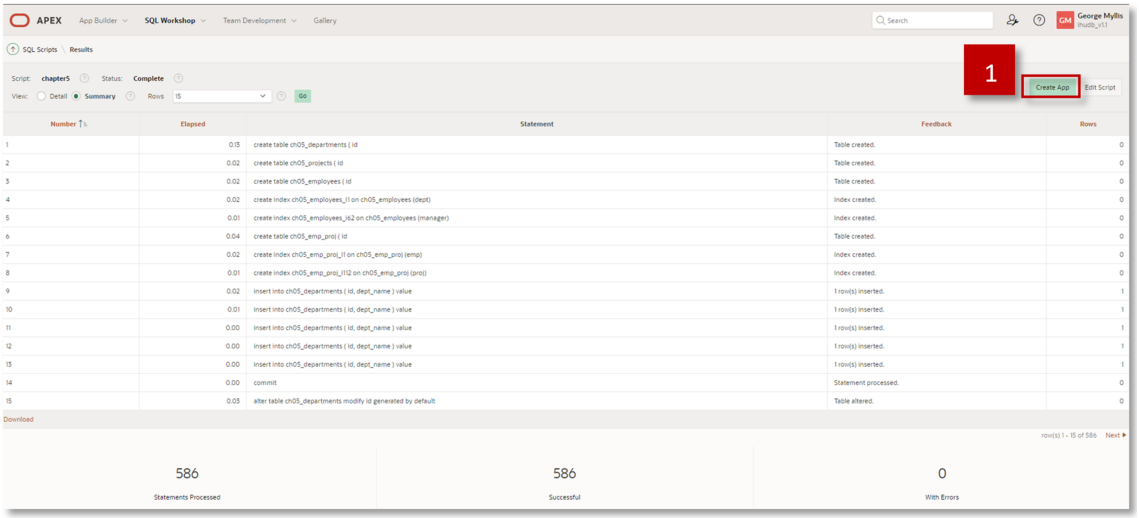
Dodatno, klauzula ON DELETE CASCADE navodi da ako se red u referentnoj tablici izbriše, svi povezani redovi u referentnoj tablici također će biti izbrisani, kako bi se održao referentni integritet.

Korak 2: Za pokretanje željene radnje pogledajte sliku5.11 i kliknite na **Run Now** dugme.



Slika 5.11: Pokreni SQL skriptu

Korak 3: Nakon što su svi iskazi uspješno obrađeni bez ikakvih pogrešaka, prijeđite na sliku 5.12 i kliknite na **Create App** dugme.



Slika 5.12: Stvorite postupak pokretanja aplikacije

5.4 Dio aplikacije voden podacima

Odjeljak aplikacije koji se temelji na podacima fokusiran je na iskorištavanje snage podataka za poticanje uvida, donošenja odluka i interakcije korisnika. U ovom odjeljku podaci zauzimaju

središnje mjesto, omogućujući korisnicima da istražuju, analiziraju i komuniciraju s informacijama na dinamičan i intuitivan način.

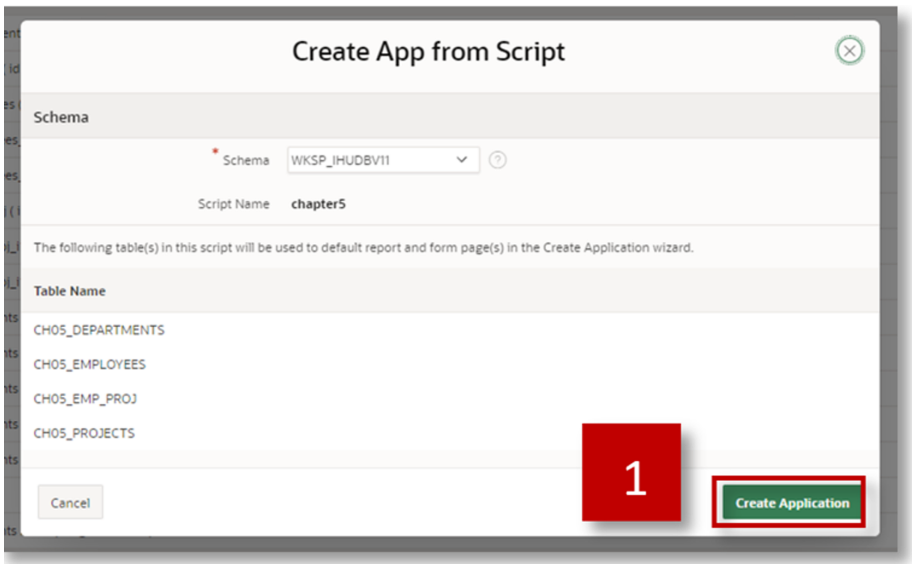
Primarni cilj ovog odjeljka je pružiti korisnicima sveobuhvatan pregled podataka, osnažujući ih da steknu vrijedne uvide i donesu informirane odluke. Kroz različite tehnike vizualizacije podataka, kao što su dijagrami, grafikoni i interaktivne nadzorne ploče, korisnici mogu lako razumjeti složene skupove podataka i identificirati obrasce, trendove i odstupanja.

Osim toga, odjeljak koji se temelji na podacima uključuje snažne mogućnosti upravljanja podacima, omogućujući korisnicima filtriranje, sortiranje i pretraživanje određenih podatkovnih točaka ili segmenata. Napredne značajke kao što su prikupljanje podataka, drill-down i funkcionalnost izvoza podataka dodatno poboljšavaju korisničko iskustvo, omogućujući dublje istraživanje i analizu.

Odjeljak također naglašava važnost integriteta i kvalitete podataka. Uključuje mehanizme za provjeru valjanosti podataka, rukovanje pogreškama i čišćenje podataka kako bi se osiguralo da su prezentirane informacije točne, pouzdane i dosljedne.

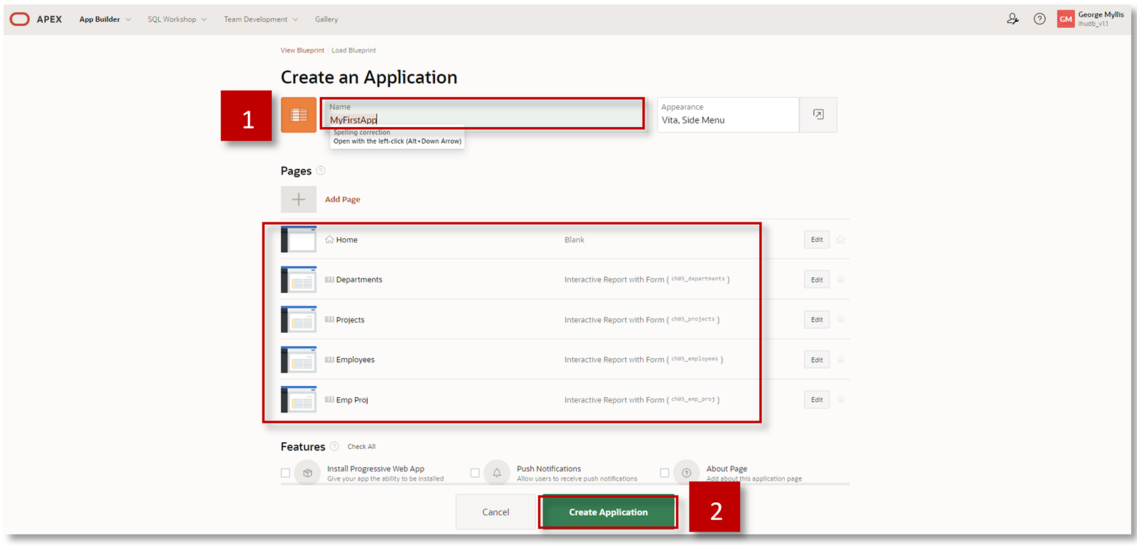
Uz odjeljak koji se temelji na podacima, korisnici mogu istraživati podatke iz više kutova, prilagoditi prikaze na temelju svojih preferencija i iskoristiti snagu podataka za dobivanje smislenih uvida. Služi kao moćan alat za analizu podataka, izvješćivanje i podršku odlučivanju unutar aplikacije, osnažujući korisnike da donose odluke na temelju podataka i postižu uspješne rezultate.

Korak 1: Kada se pojavi skočni prozor s oznakom **Create App from the Script** pojavljuje se u skladu sa slikom 5.13,potvrdite da su tablice stvorene i da će se koristiti u vašoj novoj aplikaciji, kliknite na **Create Application** dugme.



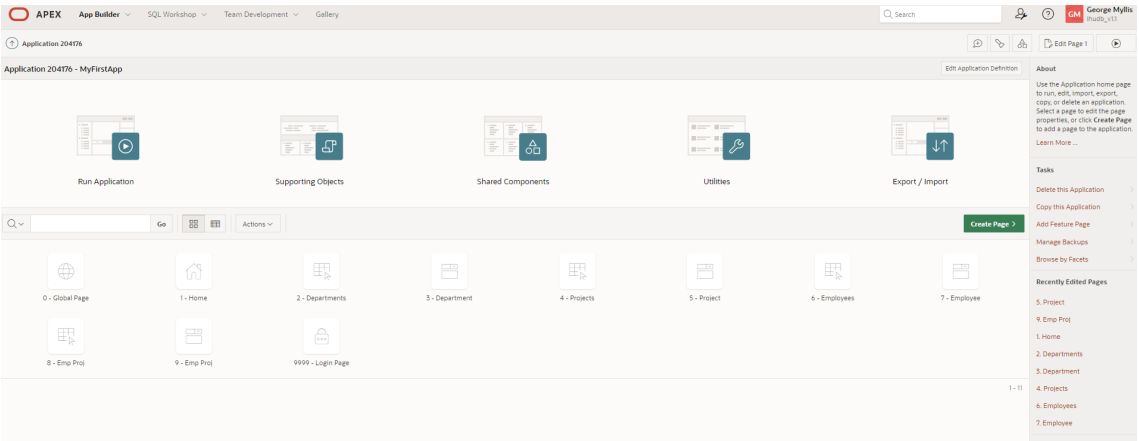
Slika 5.13: Stvorite aplikaciju iz skripte

Korak 2: Unesite naziv u svoju aplikaciju, provjerite jesu li izvješća i stranice obrazaca iz tablica dodani i da će se koristiti u vašoj novoj aplikaciji, a zatim kliknite na **Create Application**, kako je prikazano na slici 5.14



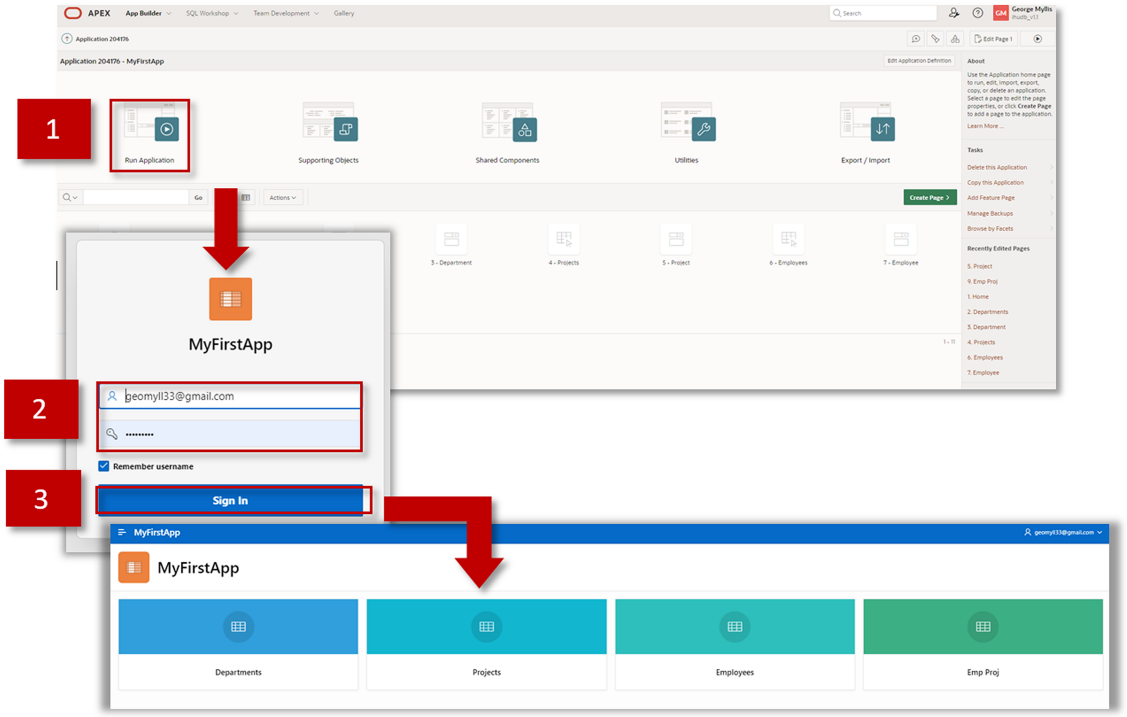
Slika 5.14: Izrađena web aplikacija

Korak 3: Okruženje za izradu aplikacija stvoreno je za vašu aplikaciju, kao što je prikazano na slici 5.15. Važno je napomenuti da unutar ovog okruženja imate mogućnost opsežne prilagodbe svoje aplikacije. Možete dodati, ukloniti ili izmijeniti polja obrasca, prilagoditi izgled, definirati pravila provjere i izvršiti brojne druge radnje kako biste aplikaciju prilagodili svojim specifičnim potrebama.



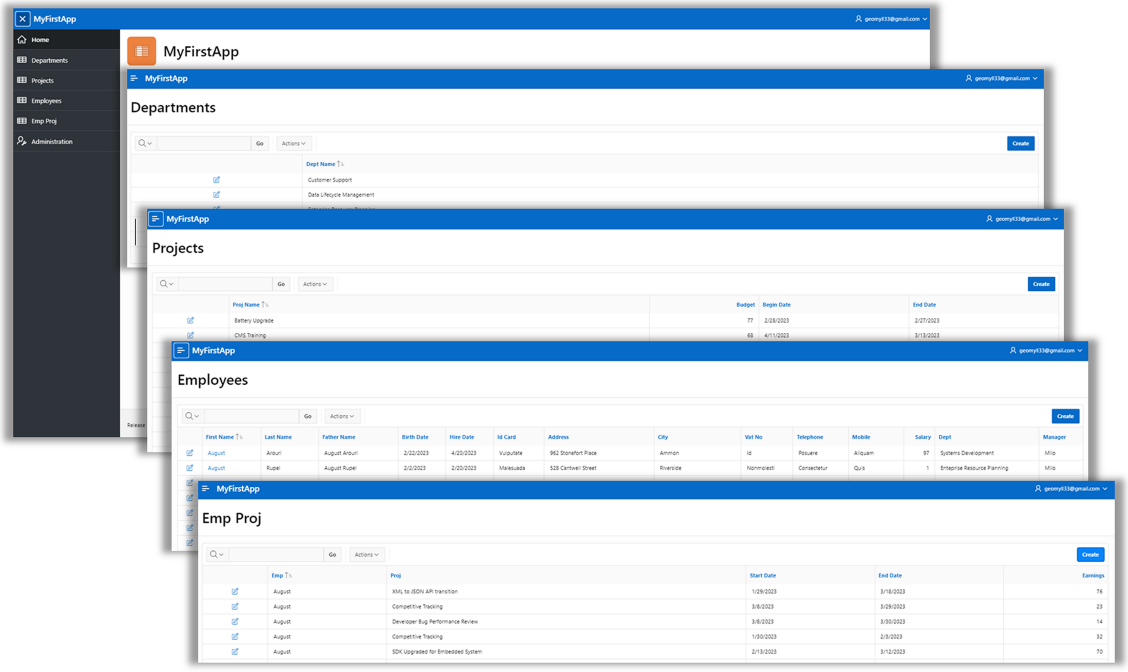
Slika 5.15: Okruženje za izradu aplikacija

Korak 4: Kliknite na **Run Application** ikonu u okruženju App Builder za pokretanje aplikacije kao što je prikazano na slici 5.16. Ovo će otvoriti novu stranicu preglednika s namjenskom vezom iz APEX-a za vašu aplikaciju. Unesite svoje vjerodajnice, koje su iste kao one u WORKSPACE, a zatim kliknite **Sign in**. Bit ćete preusmjereni na početnu stranicu svoje aplikacije.



Slika 5.16: Stranica za prijavu u aplikaciju

Korak 5: Možete se jednostavno kretati kroz svoju aplikaciju pomoću bočnog izbornika, kao što je prikazano na slici 5.17. Aplikacija pruža okruženje prilagođeno korisniku koje vam omogućuje neprimjetno dodavanje, izmjenu i predstavljanje vaših podataka prema vašim specifičnim potrebama i preferencijama.



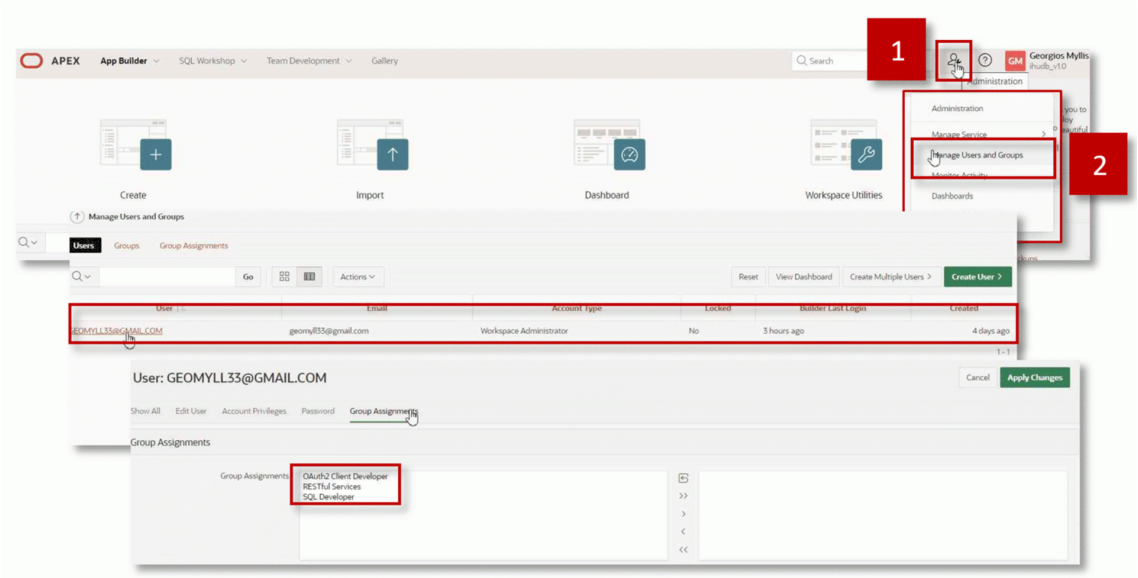
Slika 5.17: Vaše novo okruženje aplikacije

5.5 Administracija aplikacije

Što se tiče provjere autentičnosti aplikacijskih računa, to je ugrađena metoda provjere autentičnosti u Oracle Application Expressu koja omogućuje stvaranje i upravljanje korisničkim računima u korisničkom repozitoriju Application Express. Ova je metoda posebno korisna ako vaša aplikacija koristi provjeru autentičnosti Application Express računa.

U okruženju Workspace aplikacije kliknite na **Administration** i osaberite **Manage Users and Groups**. Početni stvoreni račun može se koristiti kao administrator za ugrađenu aplikaciju. Oracle APEX User održava račune za sheme provjere autentičnosti koje provjeravaju autentičnost pomoću korisničkog imena i lozinke pohranjenih u tim računima.

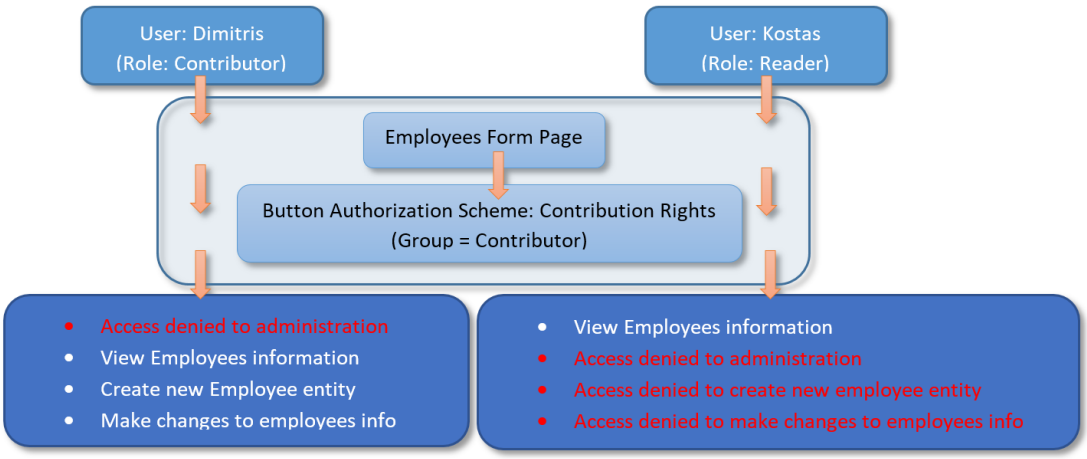
Administratori radnog prostora imaju ovlasti za stvaranje i modificiranje aplikacija i objekata baze podataka, kao i upravljanje korisničkim računima, grupama i razvojnim uslugama. Programeri mogu stvarati i mijenjati aplikacije i objekte baze podataka, dok krajnji korisnici nemaju razvojne privilegije i mogu pristupiti samo aplikacijama koje ne koriste vanjsku shemu provjere autentičnosti, kao što je prikazano na slici 5.18.



Slika 5.18: Upravljajte okruženjem korisnika i grupa

5.6 Kontrola pristupa

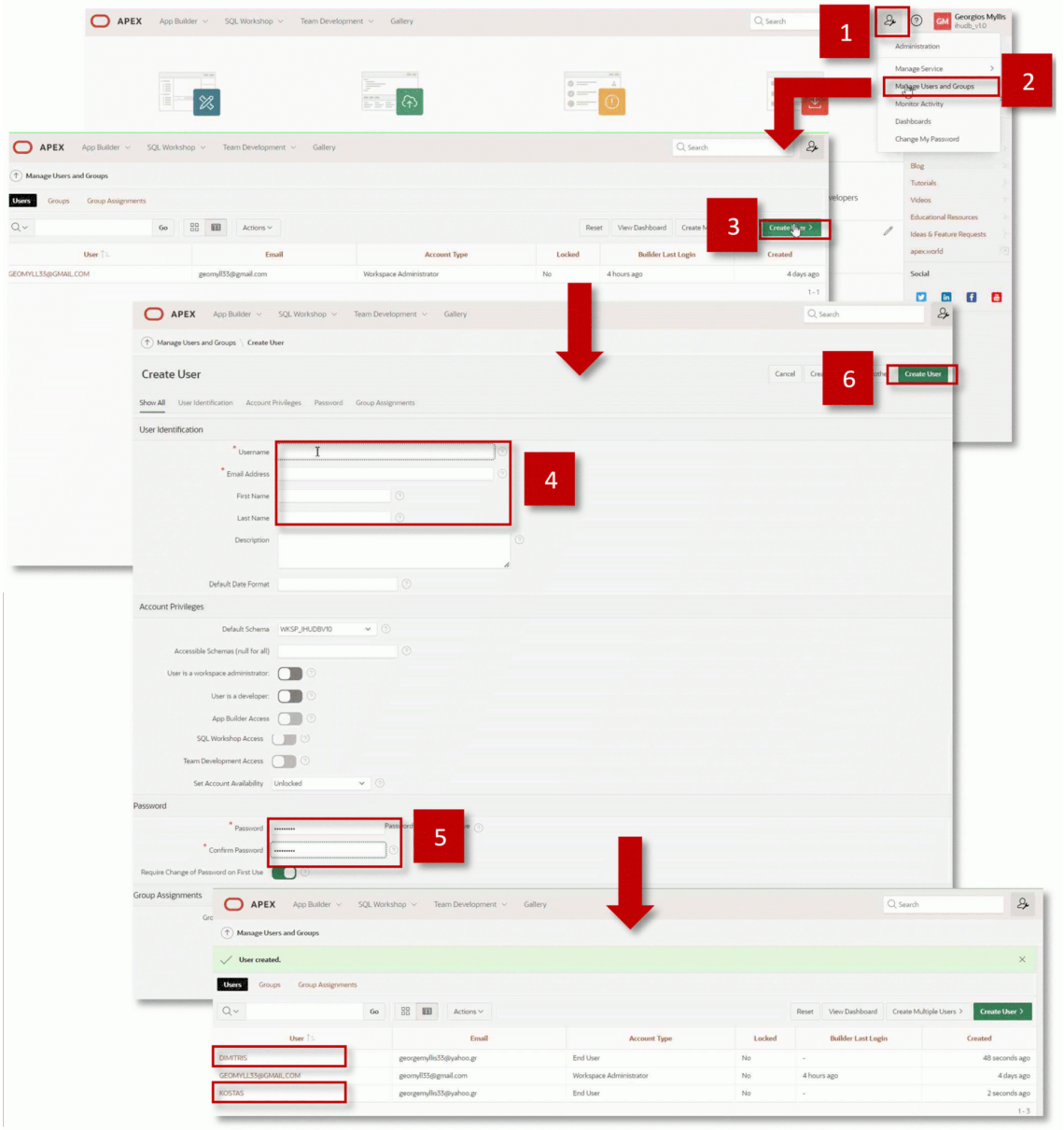
Kao što je prikazano na slici 5.19, pružit ćemo primjer kontrole pristupa za korisnike koji koriste aplikaciju, slijedeći logiku prikazanu na donjoj slici.



Slika 5.19: Primjer kontrole pristupa

Oracle APEX pruža tri ugrađene privilegije koje omogućuju kontrolu pristupa aplikaciji ili njezinim komponentama. Ove privilegije su administracija, uređivanje i pregled, a svaka odgovara ulozi pristupa. Administracija odgovara ulozi administratora, uređivanje odgovara ulozi Contributor, a View odgovara ulozi Reader.

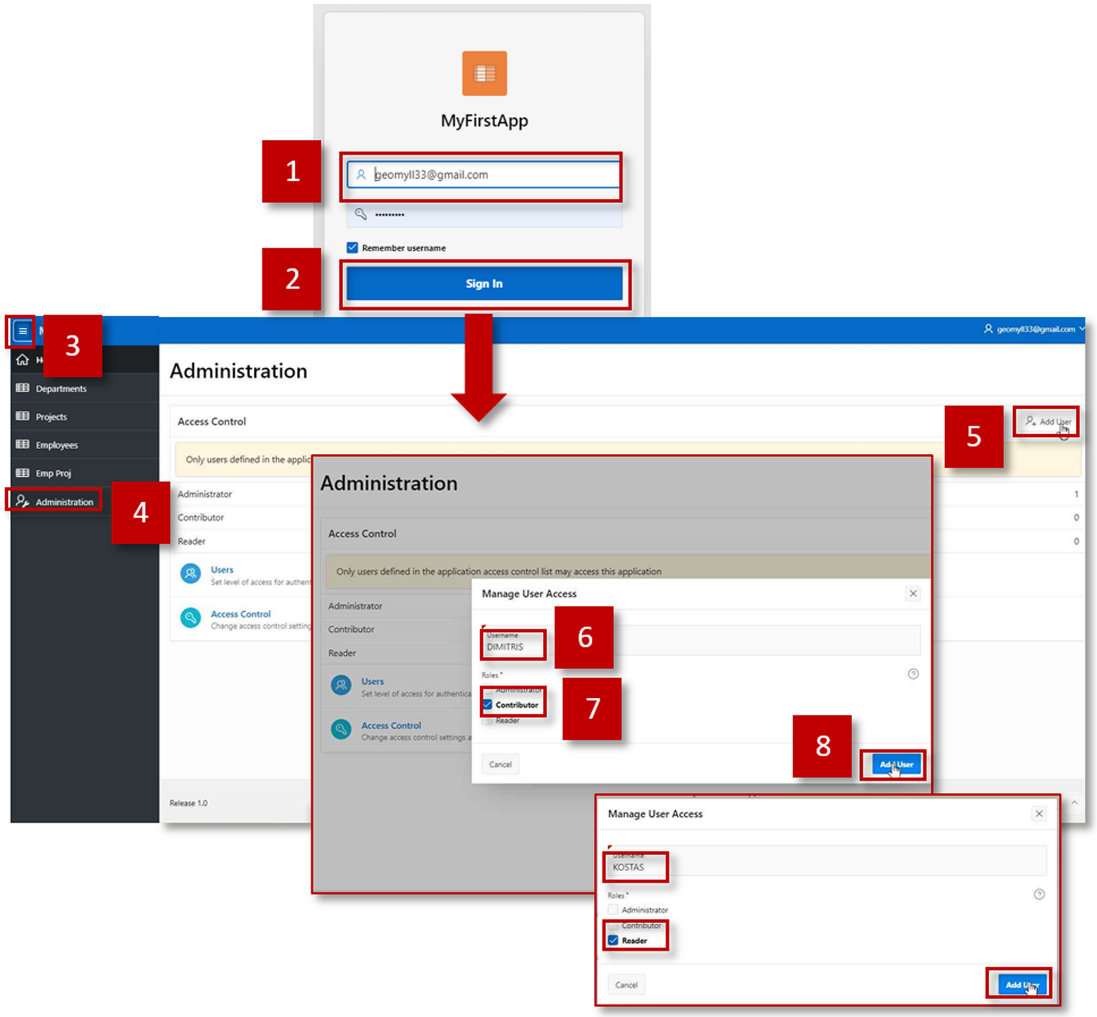
U početku, kako biste dodali nove korisnike i definirali njihove uloge, idite na **Manage Users and Groups** iz okruženja zaslona APEX. Kliknite na **Administration**, i odaberite **Manage Users and Groups**. Potom odaberite **Create User**, unesite korisničku identifikaciju i lozinku te kliknite na **Create User**,kao što je prikazano na slici 5.20.



Slika 5.20: Dodavanje novih korisnika

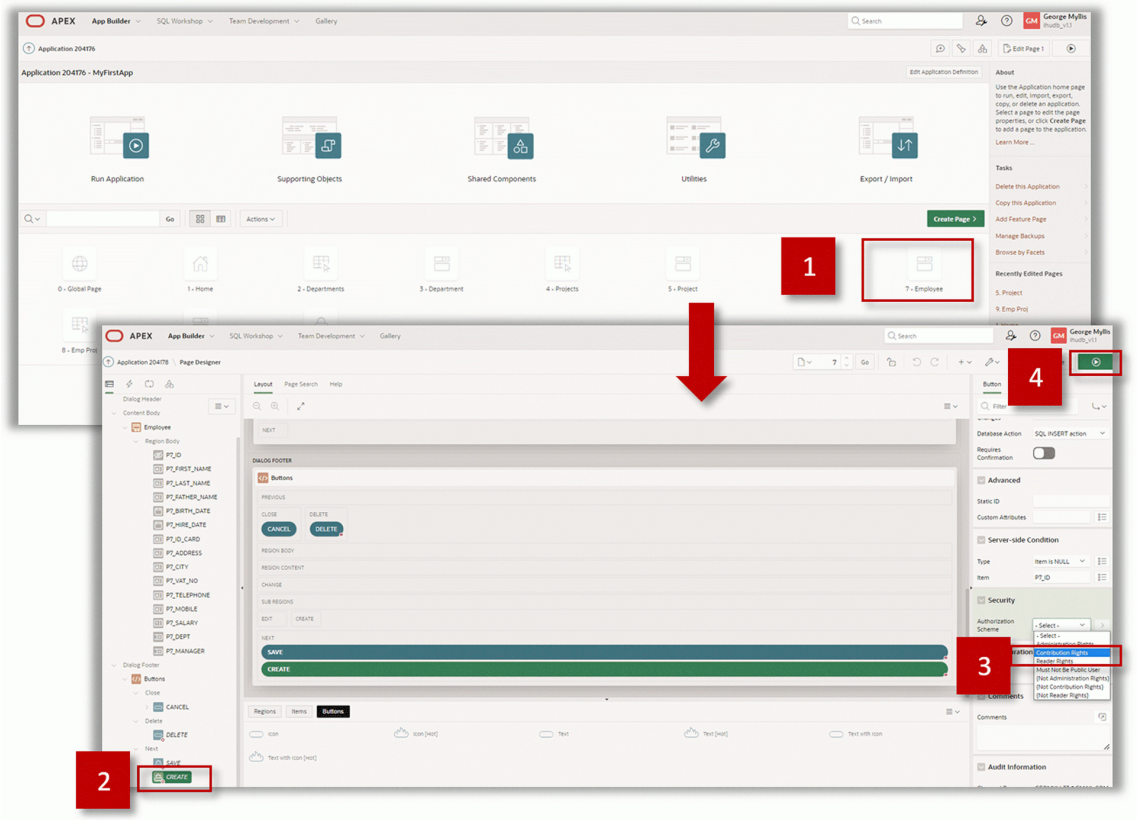
Zatim ćete morati definirati uloge za svoje korisnike. U našem primjeru pokazujemo kako se to može učiniti unutar same aplikacije.

Korak 1: Prijavite se u aplikaciju s administratorskim vjerodajnicama, idite na bočni izbornik i kliknite na **Administration**, kliknite na **Add User**, unesite korisničku identifikaciju i ulogu te kliknite na **Add User** Slici 5.21.



Slika 5.21: Dodajte uloge korisnicima

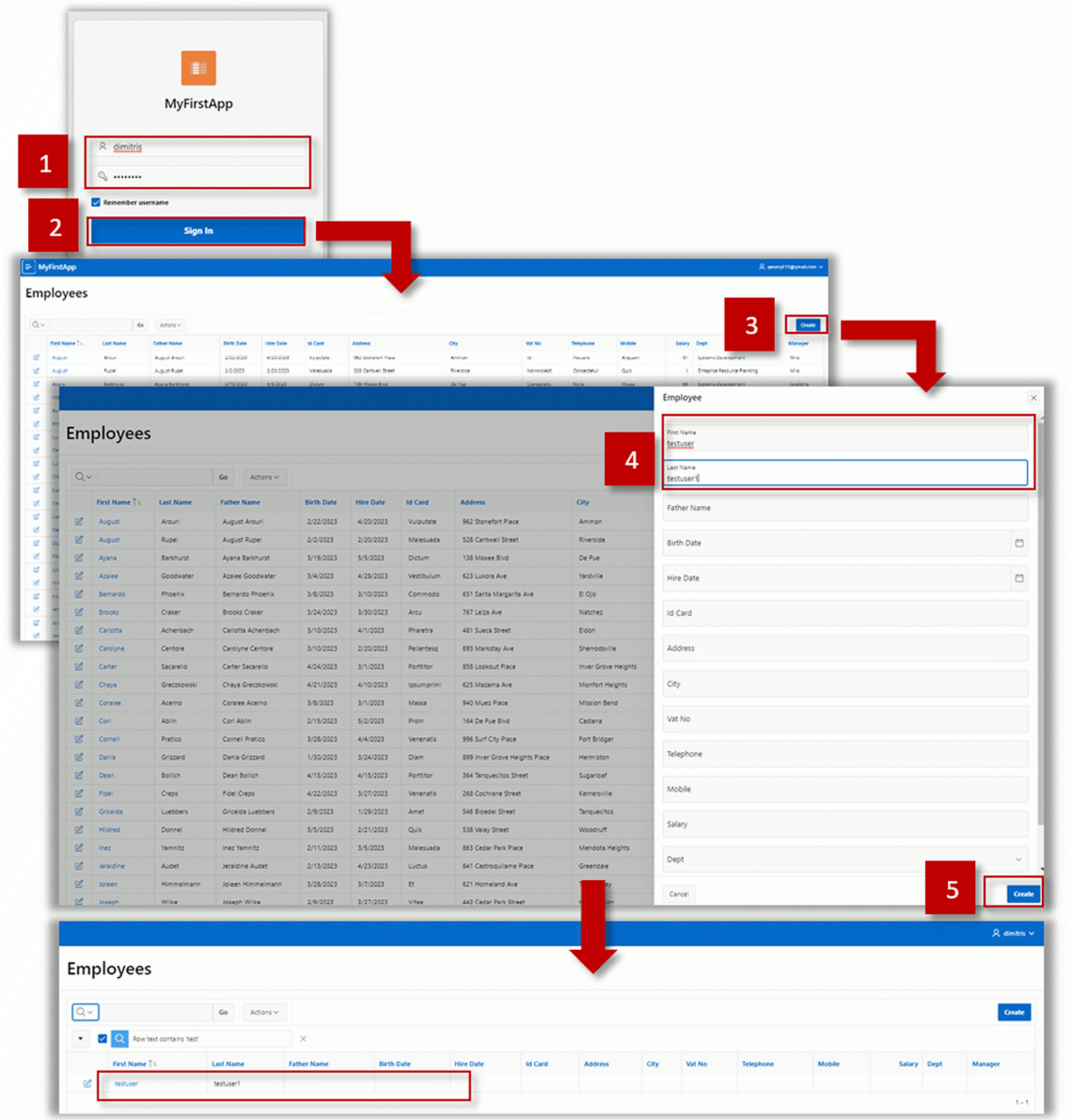
Korak 2: Idite u okruženje radnog prostora aplikacije i kliknite na **employees-Form**.U dizajneru stranice obrasca odaberite **Create** i u opciji Sigurnost odaberite **Contribution Rights** kao temu autorizacije. Kliknite na **Save**. Sada će samo korisnici s ulogom doprinosa moći stvarati i dodavati nove zaposlenike, kao što je prikazano na slici 5.22.



Slika 5.22: Stvorite konfiguraciju gumba

Korak 3: Ovaj korak služi za provjeru konfiguracija kontrole pristupa napravljenih za ulogu Contributor.

- Prijavite se u aplikaciju s vjerodajnicama korisnika Dimitrisa, kojemu je dodijeljena uloga Contributor.
- Otvorite izbornik i provjerite nedostaje li opcija administracije, kao što je očekivano.
- Pokušajte stvoriti i dodati novog zaposlenika kao suradnika, kao što je prikazano na slici 5.23.
- Provjerite možete li vidjeti postojeće podatke o zaposlenicima i možete li unositi promjene.
- Provjerite možete li stvoriti novi entitet zaposlenika.

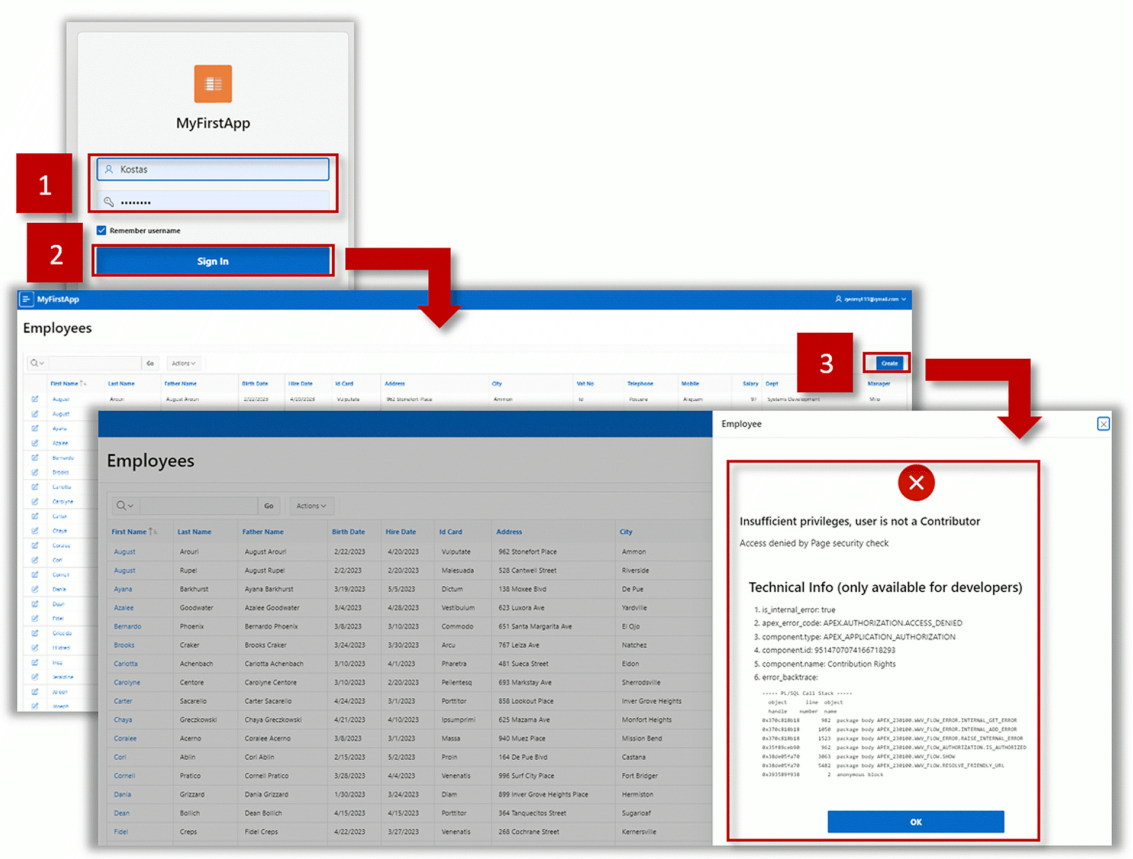


Slika 5.23: Pristup i prava suradnika

Korak 4: Ovaj korak služi za provjeru konfiguracija kontrole pristupa napravljenih za ulogu Reader.

- Prijavite se u aplikaciju s vjerodajnicama korisnika Kostas, kojemu je dodijeljena uloga Reader.
- Otvorite bočni izbornik i provjerite nedostaje li opcija administracije, kao što je očekivano.
- Kliknite na obrazac Zaposlenici u okruženju radnog mjesta aplikacije.
- Pokušajte stvoriti ili dodati novog radnika. Primit ćete poruku "pristup odbijen", kao što je prikazano na slici 5.24.
- Provjerite možete li vidjeti postojeće podatke o zaposlenicima, ali ne možete napraviti nikakve promjene.

Ovi koraci potvrđuju da su konfiguracije kontrole pristupa uspješno primijenjene na korisnike.



Slika 5.24: Pristup i prava čitatelja

5.7 Dodatni materijali za učenje

Ovdje možete pronaći sljedeće dodatne materijale za učenje:

- izvezena aplikacija,
- video materijali.

Sav dodatni materijal za učenje dostupan je na public BeeAPEX project page. Prijavite se kao gost (lozinka nije potrebna). Pronađite udžbenik u odjeljku Knjige, skripte u mapi Dio 2 > Poglavlje 5 u odjeljku Skripte i video vodiče u Zbirci video vodiča. Materijal za kratke tečajeve nalazi se u odjeljku Kratki tečajevi.

5.7.1 Izvezene aplikacije

Izvezena aplikacija je zapakirana. Instalacija stvara tablice, indeks, funkciju, proceduru i okidač te popunjava podatke. Deinstalacija uklanja sve objekte baze podataka koji se koriste u ovoj aplikaciji.

5.7.2 Video vodiči

Video vodič prikazuje svaki korak u razvoju aplikacije.

5.8 Pitanja

1. Koliko načina postoji za stvaranje nove aplikacije u ORACLE APEX Application Builderu?
2. Kako možete dodati korisnike u aplikaciju
3. Koliko ugrađenih korisničkih privilegija postoji u ORACLE APEX za kontrolu pristupa aplikaciji ili njenim komponentama?

5.9 Odgovori

1. Postoje 3 načina
   - Nova aplikacija
   - Iz datoteke
   - Početna aplikacija
2. Dođite do okruženja radnog prostora aplikacije i odaberite Upravljanje korisnicima i grupom
3. Postoje tri
   - Administrator
   - Contributor
   - Reader



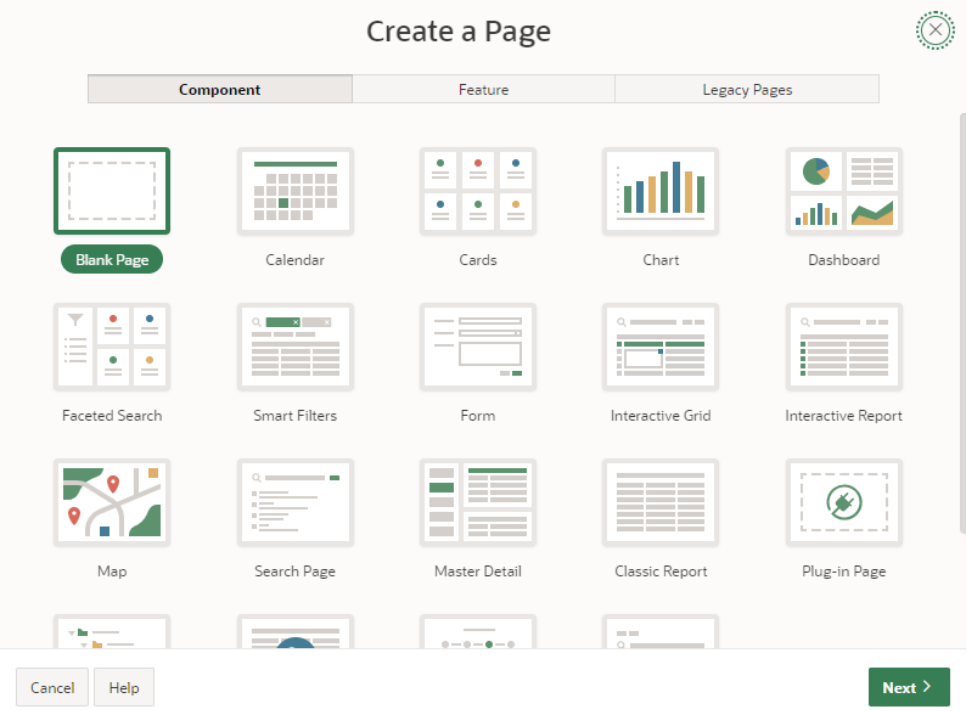
6. Kako upravljati izvješćima?

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

Podaci u bazama podataka često su besmisleni ako iz njih ne dobijemo rezultate koji nam govore više o njihovim stvarnim informacijama koje sadrže. Za predstavljanje rezultata, promatranje i saznavanje koristimo samo izvješća, kojima ćemo se baviti u sljedećem poglavlju. Naučit ćemo kako jednostavno izraditi takvo izvješće u APEX okruženju te kako ga urediti. Bavit ćemo se različitim vrstama izvješća kao što su interaktivna izvješća, klasična izvješća, Column Toggle Report ili izvješća s vizualizacijama grafikona.

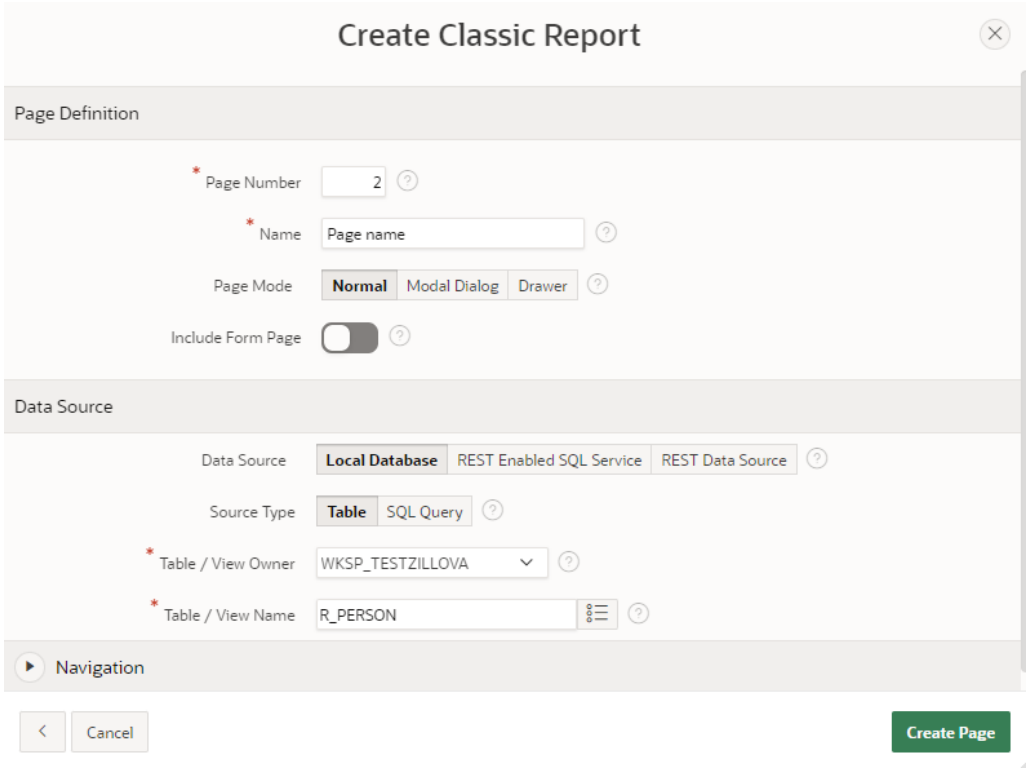
6.1 Izvješće

Report *region* koji pojednostavljuje pregledavanje podataka i prikazuje podatke u redovima. Može biti jedna cjelina ili se može koristiti za podjelu stranica u logičke cjeline. Može biti statičan, samo s naslovom i oblikovanjem, ali bez dodatnih funkcija. Ili to može biti dinamička regija koja ima specifične funkcije poput *calendar, report* or *list*. Ta su područja unaprijed programirana s osnovnom funkcionalnošću koju korisnici također mogu lako uređivati.



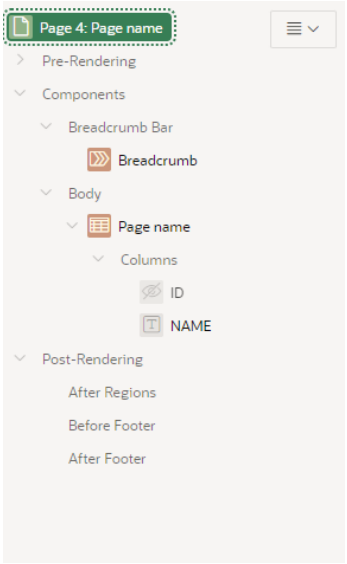
Slika 6.1: Odabir vrste stranice

Kao što možete vidjeti na slici 6.1 prilikom stvaranja **page**, korisnici mogu birati između unaprijed izgrađenih tipova stranica koje već sadrže izvješća, tako da korisnik sam ne mora umetati regiju. Primjer obrasca za izradu izvješća prikazan je na slici 6.2. Prilikom izrade stranice koja sadrži izvješće potrebno je unijeti naziv stranice i izvor podataka stranice. Podaci se mogu učitati iz *local database, REST enabled SQL* or *REST* source. Prilikom odabira opcije lokalne baze podataka, potrebno je odabrati što će se točno koristiti kao izvor izvješća - korisnik ne može samo koristiti bazu podataka *table* sama nego i postojeća *views* ili vlastite *SQL query* izjave.



Slika 6.2: Izrada stranice s izvješćem

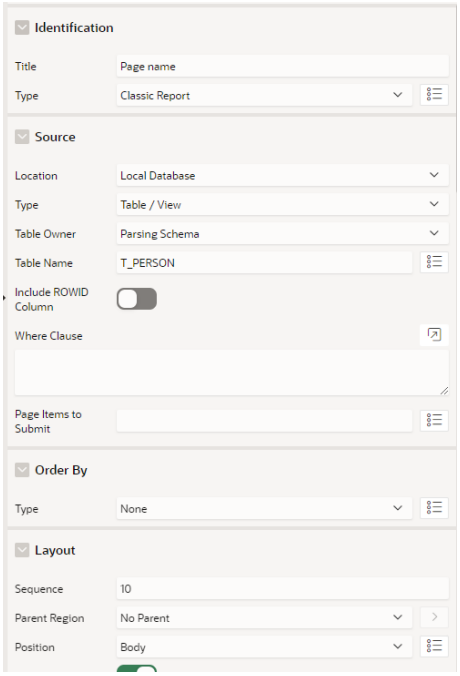
Kada je stranica kreirana, odjeljak sličan onome na slici 6.3 pojavljuje se na lijevoj strani. Na bočnoj navigacijskoj traci možete vidjeti stranicu, njen naziv i sve njene komponente. Nakon klika na jednu od komponenti ona se označava. Nakon toga možete vidjeti stupce koje sadrži, kao i njegove podregije ili stavke. Desnim klikom možete stvoriti novu praznu regiju.



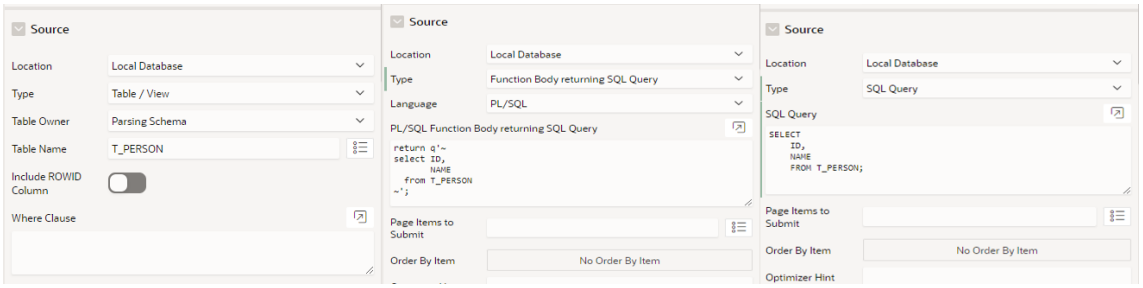
Slika 6.3: Renderiranje

Na slici 6.4 možete vidjeti detalje o regiji i njenim atributima. **Identification** definira naslov stranice i vrstu izvješća (npr. klasično izvješće). Slika 6.5 prikazuje nekoliko primjera odjeljka izvora u kojem je izvor podataka za naše izvješće kao i informacija o podrijetlu podataka (kao što

je gore navedeno: *table, view, function* ili *SQL query*) definirano. **Where Clause** govori o tome koji će se uvjet primijeniti za filtriranje podataka.



Slika 6.4: Svojstva regije



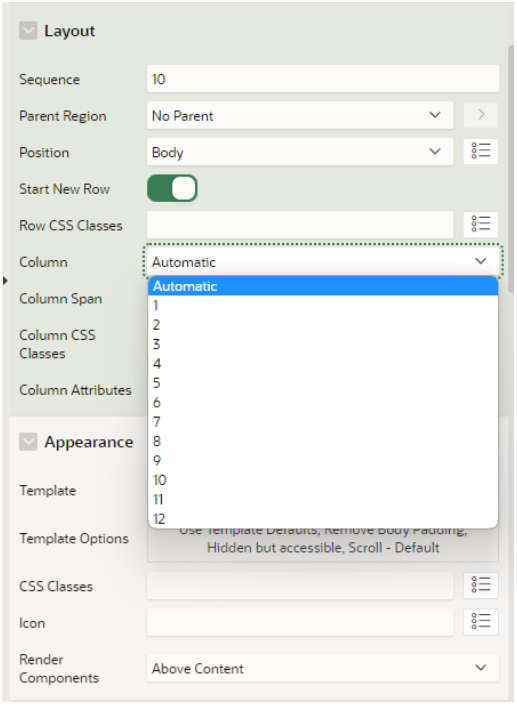
Slika 6.5: Razni izvori podataka

Na slici 6.6 možete vidjeti **Layout** gdje možete odrediti lokaciju regije unutar stranice. **Parent Region** je odrediti koja bi regija sadržavala naše izvješće. To također možete prilagoditi putem **Position**. Položaji su definirani specifičnim *page template*. Nakon izrade stranice, ona se gradi s određenom temom i izgledom ovisno o odabranom predlošku. Zadani predložak za temu **Universal** je **Standard**.

Regije su organizirane u a*grid* - dakle, svaka regija ima svoju lokaciju definiranu retkom i stupcem. Ove regije su organizirane po svojoj **sequence number**. Prvo područje nalazi se u prvom retku i prvom stupcu rešetke. Svaka sljedeća regija tada je organizirana prema jednom od ovih uzoraka: u istom retku i stupcu kao prethodna regija, i prikazana je ispod prethodne regije. Atributi **Start New Row** i **New Column** koriste se za određivanje načina organizacije za te regije.

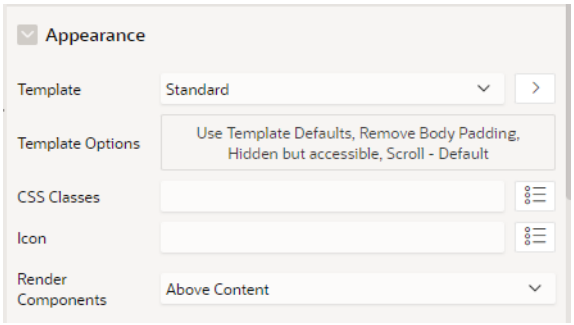
Size i *location* izračunavaju se nakon dodjele retka i stupca. To se postiže izračunavanjem zauzetosti i širine za svaki dio reda prema vrijednostima parametara **Column** i **Column Span**.

Uvlačenje područja u svom redu definirano je atributom **Column**. Vrijednost ovog stupca je u rasponu **1 - 12** koji se odnosi na **1** of **12 points** rešetke koja je podijelila širinu. Vrijednost **1** stoji za najviše *left part* i **7** je za *middle* stranice.

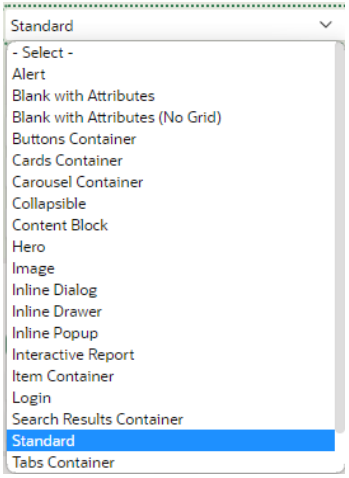


Slika 6.6: Izgled

Na slici 6.7 možete vidjeti **Appearance** odjeljak koji se koristi za uređivanje predloška za određenu regiju. Slika 6.8 prikazuje unaprijed definirane predloške koje korisnik može odabrati. Svaki predložak se može prilagoditi klikom na opcije predloška ili pisanjem vlastitog **CSS** izvještaja. U opcijama predloška možete definirati više pravila poput *body height*, *header visibility* ili čak *distance between items*.



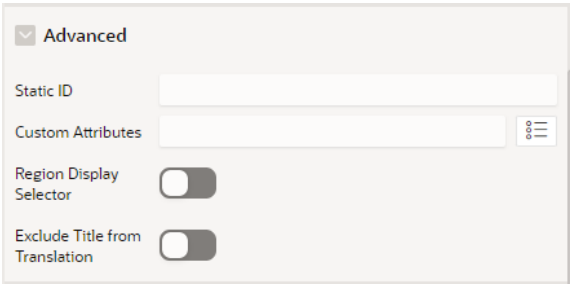
Slika 6.7: Izgled



Slika 6.8: Predlošci

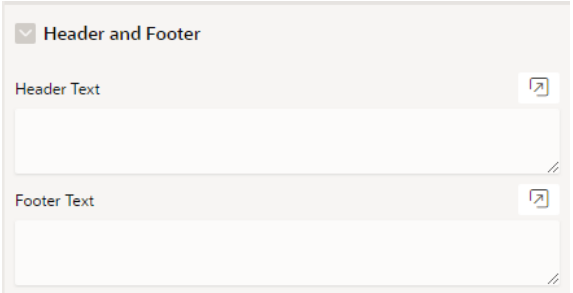
Slika 6.9 prikazuje odjeljak pod nazivom **Advanced**, gdje možete birati između četiri opcije. **Static ID** namijenjen je za postavljanje identifikatora izvješća. Ovaj **Static ID** se koristi kao **ID** za **HTML** element unutar stranice za situacije kada korisnik želi konfigurirati izgled stranice pomoću **CSS**. **Custom Attributes** asu za dodatne informacije za **HTML** element. Prilagođeni atributi pohranjuju se kao naziv/vrijednost, na primjer: *name* = ”*MY_REPORT*”

Region Display Selector je komponenta regije koja pruža navigacijske kontrole za druge regije unutar stranice. **Exclude Title from Translation** izbjegava naslov iz prijevoda u slučaju da ta stranica ima definirane prijevode.



Slika 6.9: Napredno

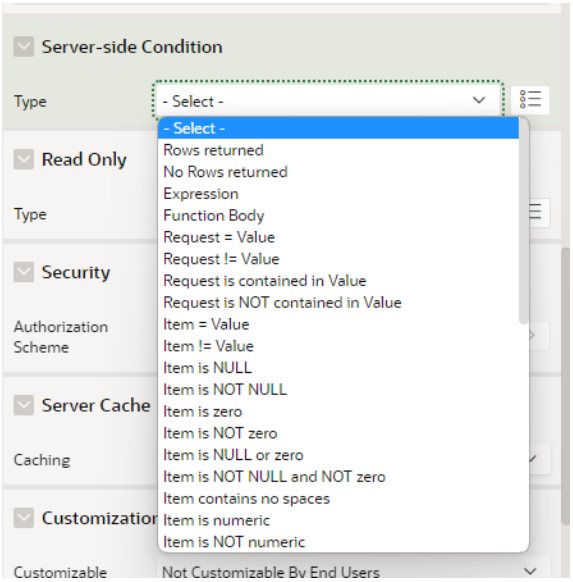
Header and Footer prikazani na Slici 6.10koriste se za uređivanje sadržaja ovih dijelova izvješća. Važno je napomenuti da se podnožje i zaglavlje odnose na izvješće, a ne na cijelu stranicu.



Slika 6.10: Zaglavlje i podnožje

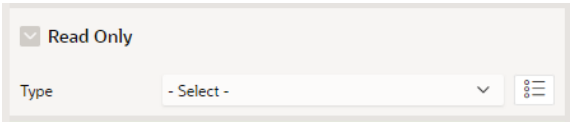
Vrste **Server-side Conditions** su navedeni na Slici 6.11. U aplikaciji koju je razvio Oracle APEX, **Server-side Conditions** rješavaju se u fazi inicijalizacije učitavanja web stranice. Izvješće

se ne prikazuje ako uvjeti na strani poslužitelja nisu ispunjeni. Postoji niz vrsta uvjeta. Kao primjer možete usporediti vrijednosti stavki ili postoje uvjeti za provjeru zahtjeva.



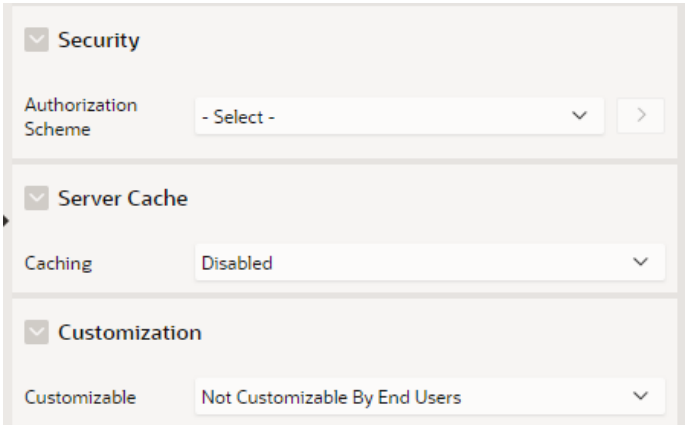
Slika 6.11: Stanje na strani poslužitelja

Read Only stavka prikazana na slici 6.12 has same options as **Server-side Condition** stavka prikazana na slici 6.11. Naravno, nije moguće promijeniti prikaz odabrane regije, dok je ona samo za čitanje. U slučaju osnovnog izvješća, nije moguće sortirati vrijednosti stupaca.



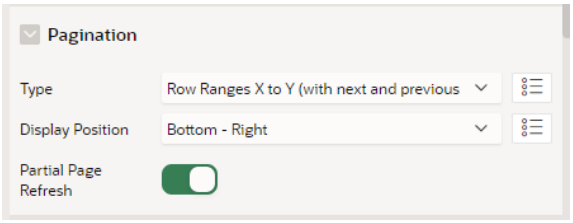
Slika 6.12: Samo za čitanje

Na slici 6.13, možete vidjeti zadnja tri odjeljka koja će biti objašnjena. Ovi su **Security, Server Cache**, i **Customization**. **Security** odjeljak sadrži opcije za ograničenja u pogledu prikazivanja web stranice - npr. *public website, only for authorized users*. **Server Cache** odjeljak uključuje postavke o upravljanju predmemorije. U **Customization** prozoru možete podesiti pristup za izmjenu izvješća.



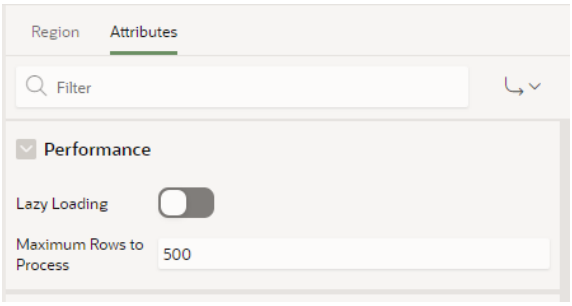
Slika 6.13: Sigurnost, predmemorija poslužitelja i prilagodba

Druga grupa odjeljaka je **Attributes**. Ovdje možete postaviti stvari *like number of displayed rows*, ako je *possible to list over report results* i u *which way you will list* kroz izvještaj.

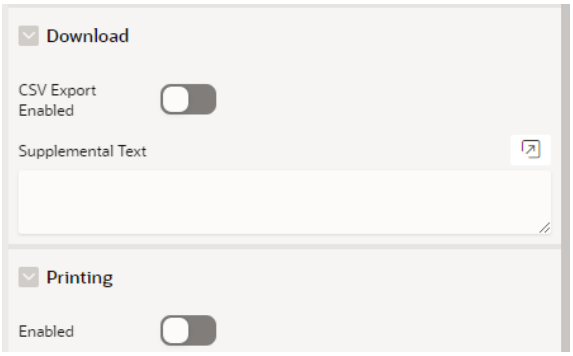


Slika 6.14: Paginacija

Još jedna zanimljiva značajka je postavljanje maksimalnog broja učitanih redaka. Također možete definirati što prikazati kada postoji*no data found.* Moguće je preuzeti svoja izvješća kao **.csv** datoteku ili ispis. Sve navedeno u ovom odlomku može se pronaći u **Attributes**.

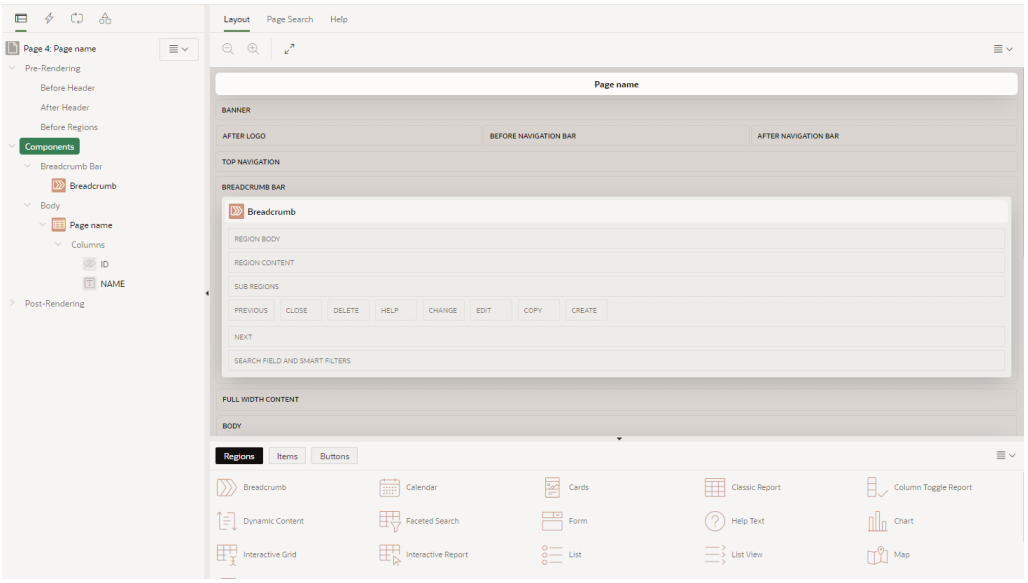


Slika 6.15: Broj redaka za učitavanje



Slika 6.16: Preuzimanje i ispis

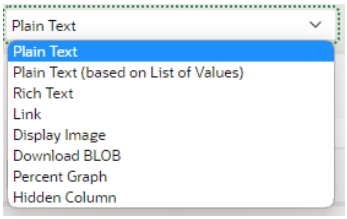
Vrlo važan dio je *center* aplikacije. Na slici 6.17 možete vidjeti **Layout** koji nam pokazuje kako će sve komponente biti organizirane na našoj stranici. Cijela je stranica strukturirana u mrežu, tako da je intuitivnije zamisliti gdje će se komponente smjestiti. Kako biste bili što je moguće više prilagođeni korisniku, možete odabrati regiju/stavku i *drag & drop* na željeno mjesto. Za svaku regiju postoji svoj zadani predložak, koji se dobiva kada ga stavite. Sve potrebne postavke su već postavljene.



Slika 6.17: Središte dizajnera stranica

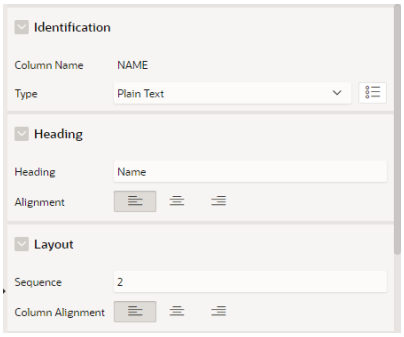
Kao što možete vidjeti na lijevoj strani stranice na slici 6.17, APEX prema zadanim postavkama stvara neke komponente za izvješće. Svaka komponenta predstavlja jedan stupac, a moguće ju je mijenjati na isti način kao i kod izvješća.

Postavke stupaca prikazane su na slici 6.19. U **Identification** možete vidjeti dvije karakteristike - **Type** i **Name of Column**. Nije moguće promijeniti **Name of Column**, jer se njegova vrijednost dobiva iz *SQL query*. **Type of Column** prikazan na Slici 6.18 određuje kako će stupac biti prikazan – istaknuti neke od njih: *Hidden* (ništa nije prikazano), *Plain Text*, *Display Image, Percent Graph* i tako dalje. prema zadanim su postavkama svi stupci označeni kao *Plain Text*.



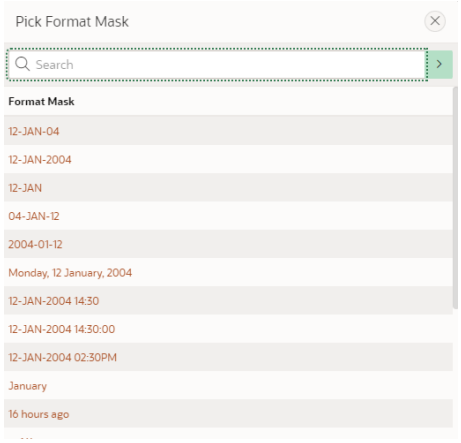
Slika 6.18: Vrste stupaca

Heading sastoji se od dvije opcije o zaglavlju. Za razliku od naziva stupca, **Heading** (naziv) se može urediti. Zaglavlje također može sadržavati **HTML** kod. **Alignment** opcija govori sama o sebi. **Layout** odjeljak nudi **Sequence**, koji određuje kojim će redoslijedom biti dostavljeni stupci i također **Alignment** opciju, isto kao u prethodnom odjeljku.



Slika 6.19: Postavke stupaca

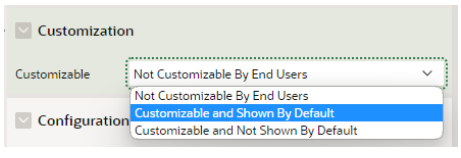
Zadnji ključni dio je **Appearance**, koji bi trebao biti uređen, kada želite tretirati broj kao niz u nekim *special format*. Kao što možete vidjeti na slici 6.20, možete birati između mnoštva osnovnih formata kada kliknete na **Format Mask**. Ove formate nudi sam APEX, ali vi možete izraziti vlastiti format ako želite.



Slika 6.20: Maska formata.

6.2 Klasično izvješće

Classic Report je vrsta izvješća koja omogućuje samo osnovni prikaz podataka, bez mogućnosti da korisnici mijenjaju opcije prikaza. Jedino što je korisniku dostupno je sortiranje podataka po pojedinim stupcima, to također treba postaviti.



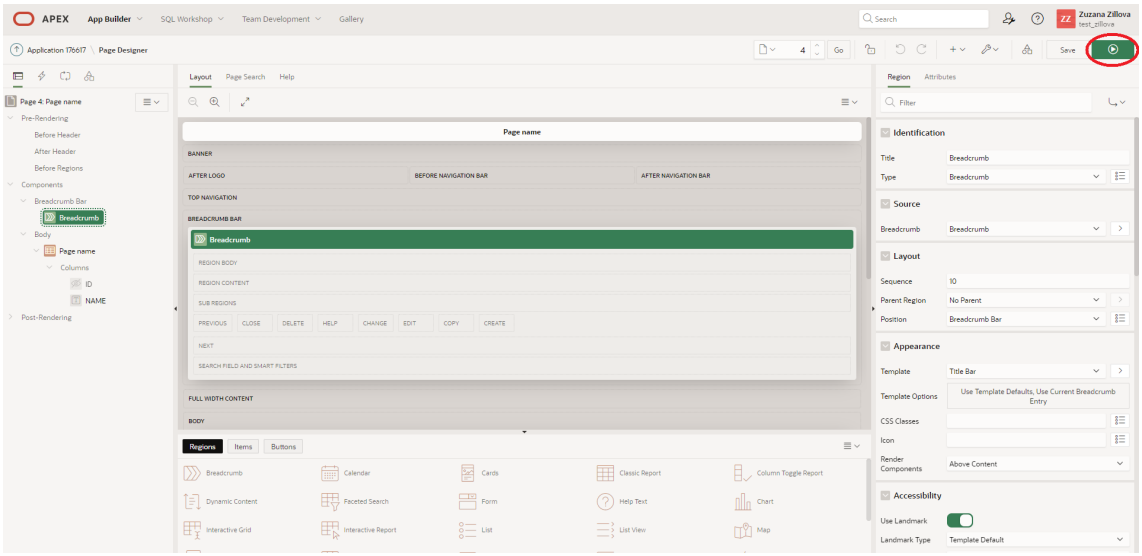
Slika 6.21: Prilagodba

Id Number	First Name	Last Name
000103/7		Goodsite
000103/7702	Vernell	Brancaleone
000104/1072	Peterson	Barree
000104/2678	Athony	Lather
000105/2999	Candler	Frencher
000105/9247	Britt	Drane

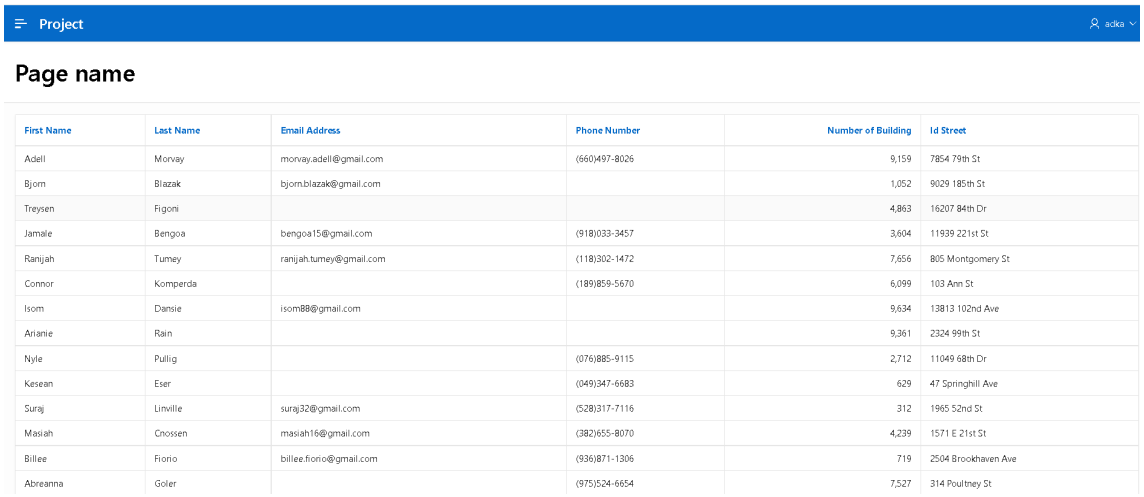
Sort by this column

Slika 6.22: Mogućnost sortiranja po stupcima

Ako želite vidjeti kako će stranica izgledati, možete kliknuti na zelenu strelicu u krugu. Kao što možete vidjeti na slici 6.23, strelica se nalazi u gornjem desnom uglu. Na slici6.24možemo vidjeti kako izgleda klasično izvješće. Ima stupce s imenima i podacima. Korisniku nije dopuštena nikakva interakcija, osim sortiranja po stupcima.



Slika 6.23: Lokacija zelene strelice za prikaz stranice

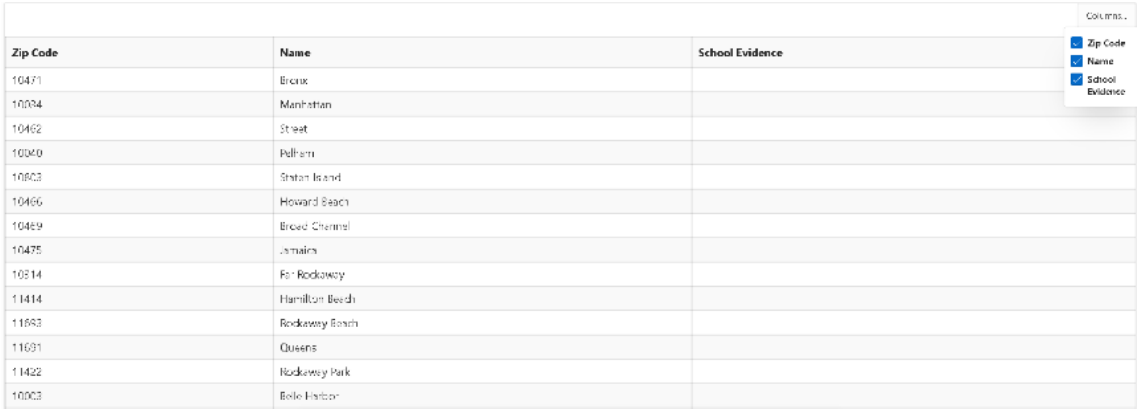


First Name	Last Name	Email Address	Phone Number	Number of Building	Id Street
Adell	Morvay	morvay.adell@gmail.com	(660)497-8026	9,159	7854 79th St
Bjorn	Blazak	bjorn.blazak@gmail.com		1,052	9029 185th St
Treysen	Figoni			4,863	16207 84th Dr
Jamale	Bengoa	bengoa15@gmail.com	(918)033-3457	3,604	11939 221st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

Slika 6.24: Primjer klasičnog izvješća

6.3 Izvješće o prebacivanju stupaca

Column Toggle Report je gotovo isto kao klasično izvješće, samo jedna dodatna značajka je navesti koji stupac treba biti vidljiv, a koji ne. Možemo ga vidjeti u desnom gornjem kutu *Figure 6.25*.

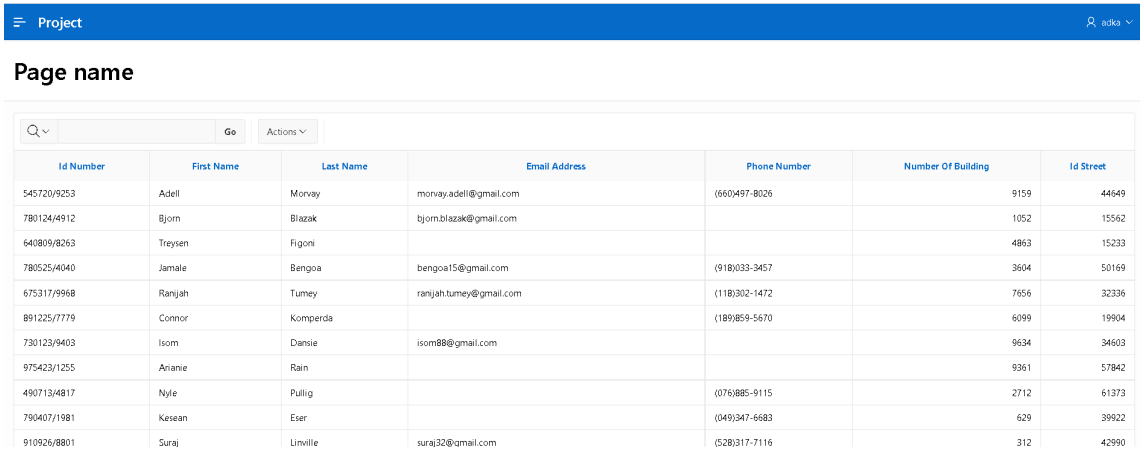


Zip Code	Name	School Evidence
10471	Bronx	
10034	Manhattan	
10462	Street	
10040	Pelham	
10303	Staten Island	
10466	Howard Beach	
10469	Broad Channel	
10475	Jamaica	
10314	Fo- Rockaway	
11414	Hamilton Beach	
11693	Rockaway Beach	
11691	Queens	
11422	Rockaway Park	
10003	Belle Harbor	

Slika 6.25: Mogućnost promjene vidljivosti stupca u izvješću o preklapanju stupaca

6.4 Interaktivno izvješće

Interactive Report prikazano na Slici 6.26 omogućuje korisnicima da naprave neke prilagodbe izvješća dok gledaju stranicu bez pristupa jezgri aplikacije. Interaktivno izvješće izrađuje se kao i svako drugo i ne razlikuje se puno u izgledu. Sada ćemo pobliže pogledati pojedinačne funkcije i prilagodbe koje korisnik može prilagoditi u interaktivnom izvješću.

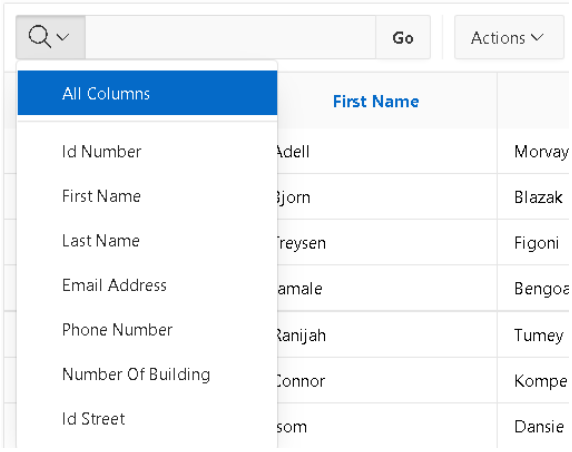


Slika 6.26: Interaktivno izvješće

Panel za pretraživanje prikazan na slici 6.27 sastoji se od četiri stavke:

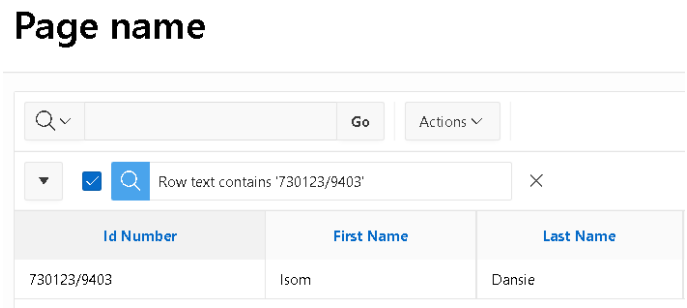
- **Povećalo,**
- **tekstualni okvir,**
- **gumb Idi,**
- **gumb za radnje**.

Prve tri stavke omogućuju korisniku filtriranje redaka izvješća. Unesemo niz za pretraživanje u polje, kliknemo na **Magnifying Glass**, gdje biramo koji stupci trebaju biti prikazani i zatim kliknemo na **Go Button**. Vidjet ćemo one zapise koji sadrže traženi niz.



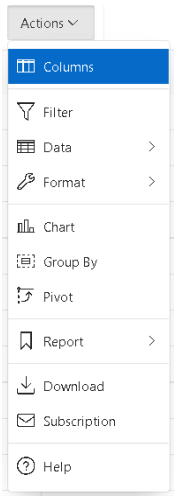
Slika 6.27: Panel za pretraživanje

Na primjer, želimo tražiti zapise koji sadrže broj **"730123/9403"** in any column. Ovaj filtar će se prikazati, a možete ga isključiti klikom na **X Button**. Situacija je prikazana na slici 6.28. Možemo imati više filtara aktivnih u isto vrijeme na različitim stupcima.



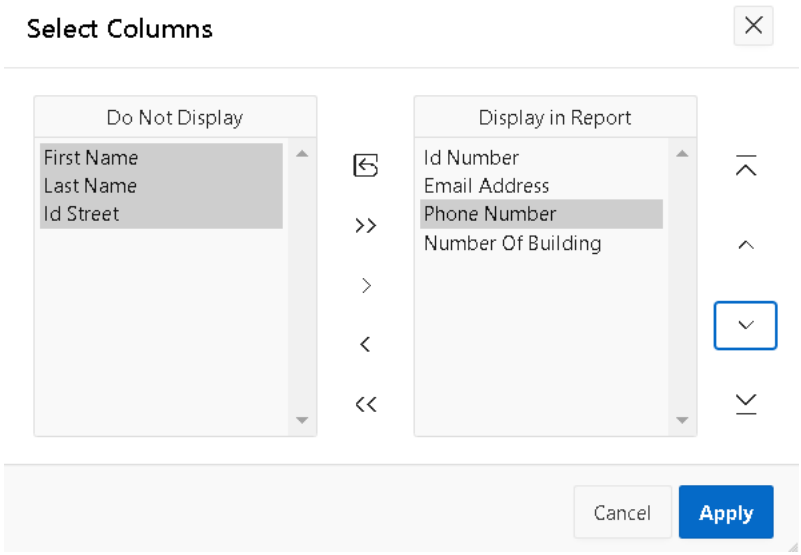
Slika 6.28: Pretraga

The **Actions Button** nudi više mogućnosti za podešavanje prikaza izvješća. Nakon klika pojavit će se podizbornik koji sadrži nekoliko stavki. Navedene stavke prikazane su na slici 6.29.



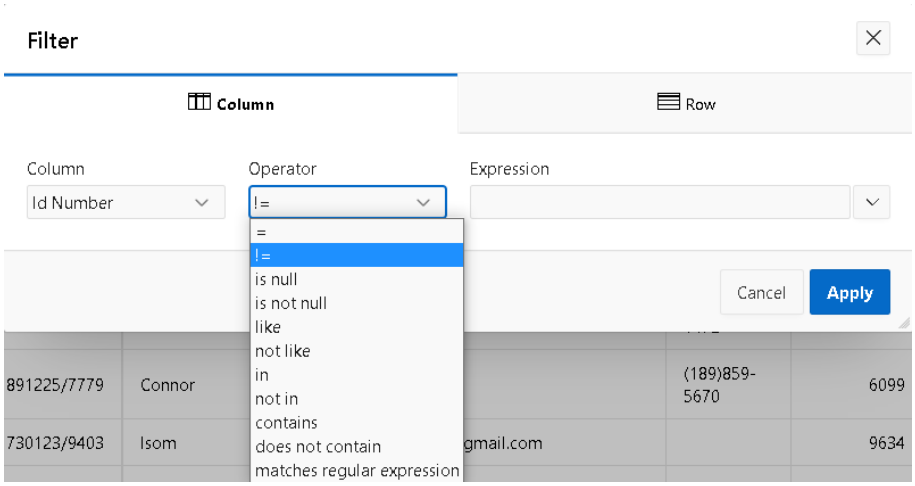
Slika 6.29: Gumb za radnje

Gumb **Columns** će nam pokazati kutiju koju možete vidjeti na slici 6.30. HPrije toga, korisnik može odabrati koji će stupci biti prikazani u izvješću, kao i redoslijed kojim će biti prikazani. Korisnici to mogu postići pomicanjem stupaca između dva odjeljka **Do not Display** i **Display in Report**.Postoje korisne strelice koje olakšavaju rad s pomicanjem stupaca.



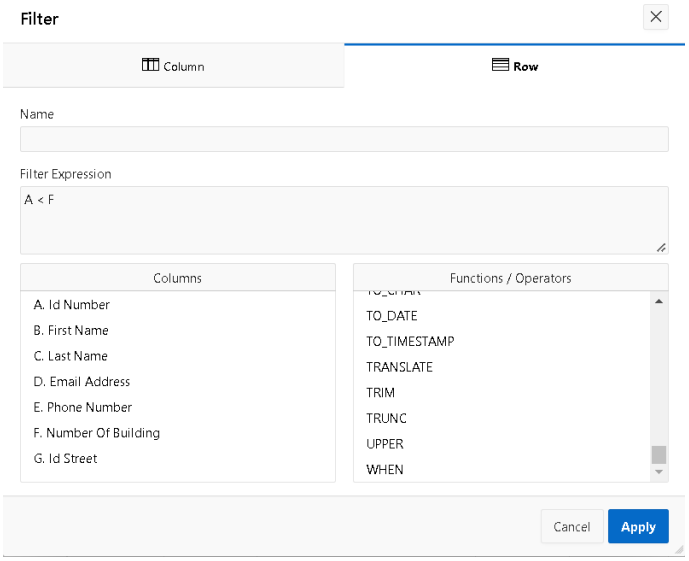
Slika 6.30: Odabir stupaca za prikaz

Kao drugu opciju imamo a **Filter** prikazan na Slici 6.31. Vrijednosti u stupcu možemo filtrirati izrazima poput *to be equal to, not equal, is null*, itd. Izraz možete definirati odabirom iz unaprijed definiranog ili pisanjem vlastitog.



Slika 6.31: Filtar stupaca

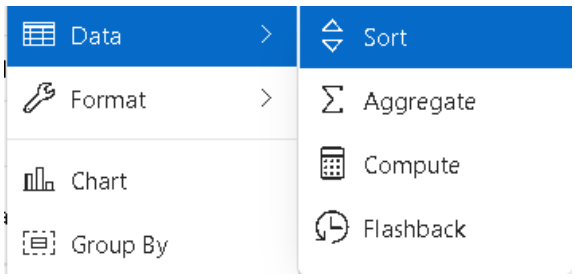
U slučaju filtriranja redaka, imamo izbor bezbrojnih operacija i izraza koje možemo sastaviti. Ovaj se filtar pojavljuje u **Filter Expression** prozor prikazan na slici 6.32.



Slika 6.32: Filtar redaka

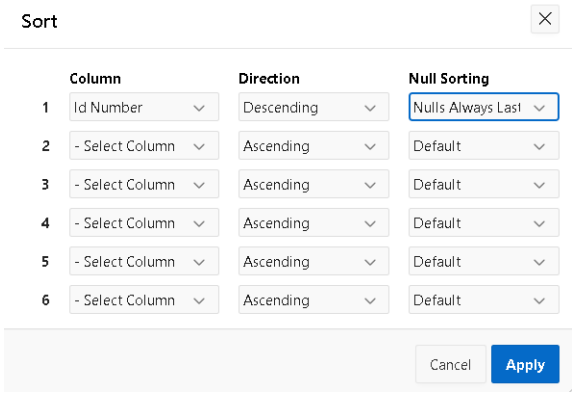
Nakon klikanja na **Data** gumb, izbornik prikazan na slici 6.33 pojavit će se. Izbornik sadrži još četiri mogućnosti:

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



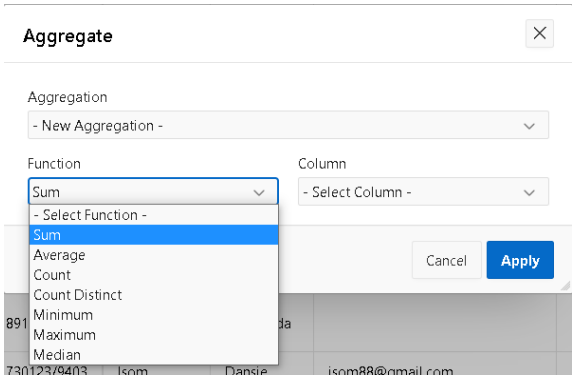
Slika 6.33: Nakon klika na gumb Data

Na Slici 6.34 možete vidjeti **Sort** opciju koja se može primijeniti na svaki stupac, a korisnik može odrediti kako s njom raditi *null values*.



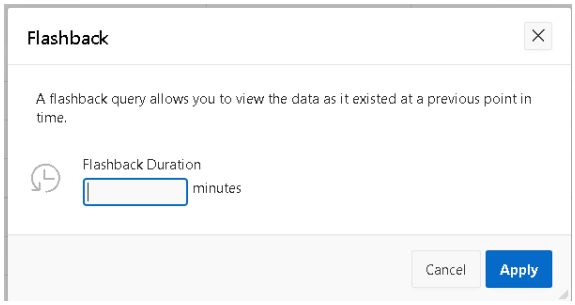
Slika 6.34: Sort

Sa **Aggregate** možemo izvršiti funkciju združivanja iznad jednog od stupaca našeg izvješća. Funkcije agregacije koje možemo izabrati prikazane su na slici 6.35.



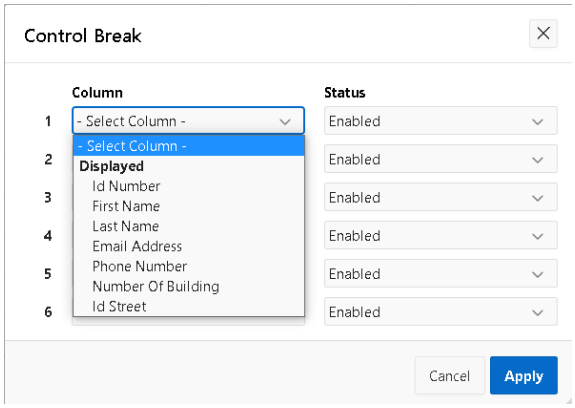
Slika 6.35: Aggregate

Flashback funkcija je prikazana na Slici 6.36 omogućuje nam da se vratimo na stanje prikazanog izvješća prije promjena prije nekoliko minuta.



Slika 6.36: Flashback

Control Break prikazan je na Slici 6.37 omogućuje nam da izvješće podijelimo u nekoliko manjih cjelina, kao da se radi o zasebnim izvješćima. Ove male jedinice podijeljene su ovisno o vrijednostima stupaca. Svaka jedinica dijeli iste podatke u odabranom stupcu.



Slika 6.37: Control Break

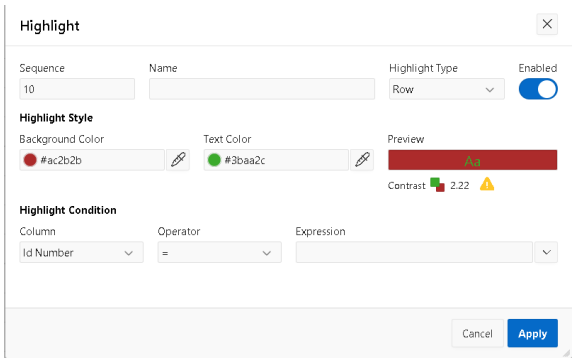
Slika 6.38 pokazuje nam kontrolni prekid primijenjen na *First Name* stupac.

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

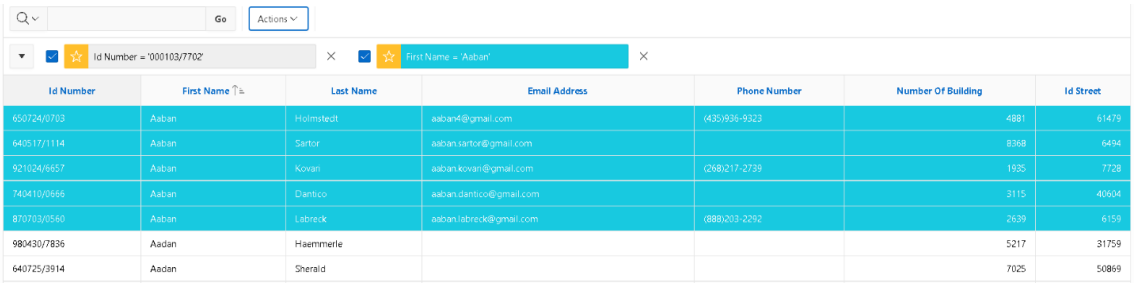
Slika 6.38: Rezultat prekida kontrole

Na Slici 6.39 možete vidjeti **Highlight** funkcija koja lako čini zapise u izvješću transparentnima, bez filtriranja ostalih. Možemo istaknuti jednu ćeliju ili cijeli red bojama koje odaberemo. Usporedba vrijednosti stupca funkcionira isto kao i s **Filter**, osim što nefiltrirani zapisi ne nestaju iz izvješća.



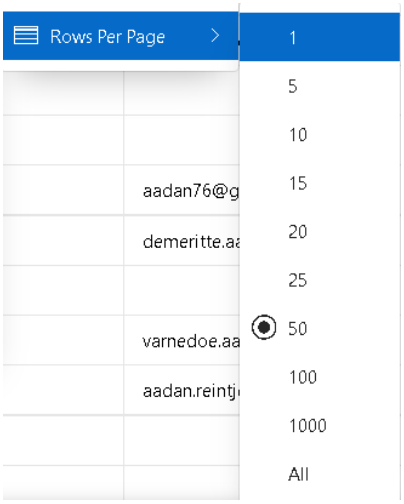
Slika 6.39: Highlight

Na slici 6.40 možemo vidjeti izvješće s označenim redovima, s imenom jednakim **"Aaban"**.



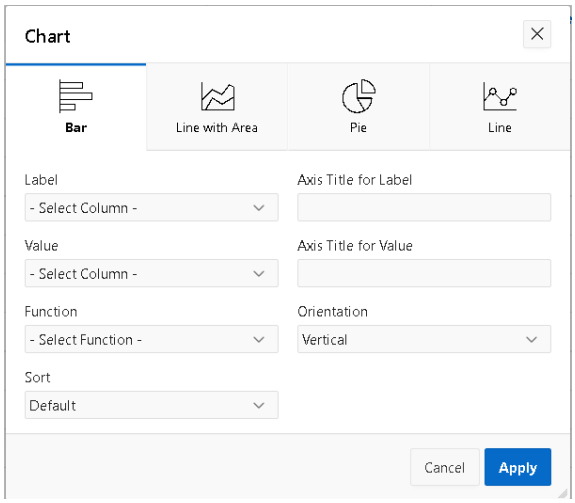
Slika 6.40: Istakni rezultat

Slika 6.41 prikazuje **Rows Per Page** što je atribut, gdje korisnik može izabrati koliko će zapisa biti prikazano na stranici. To je slična postavka **Pagination**.



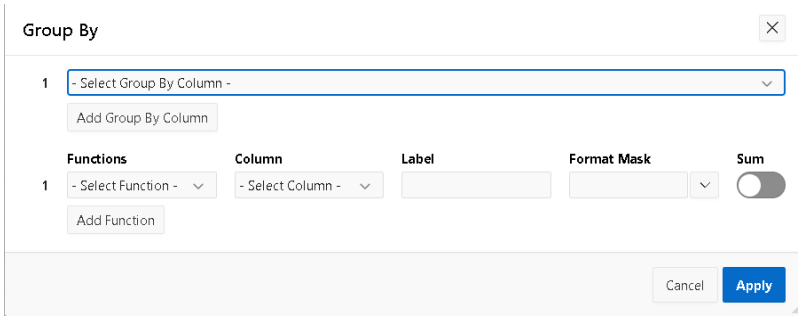
Slika 6.41: Redak po stranici

Ako je prikaz podataka u izvješću nedovoljan, postoji i mogućnost projiciranja tih podataka u grafikon putem **Chart** stavka prikazana na slici 6.42. Postoji izbor od **four** osnovni grafovi. Također možemo odabrati funkciju združivanja koja se primjenjuje na dati stupac.



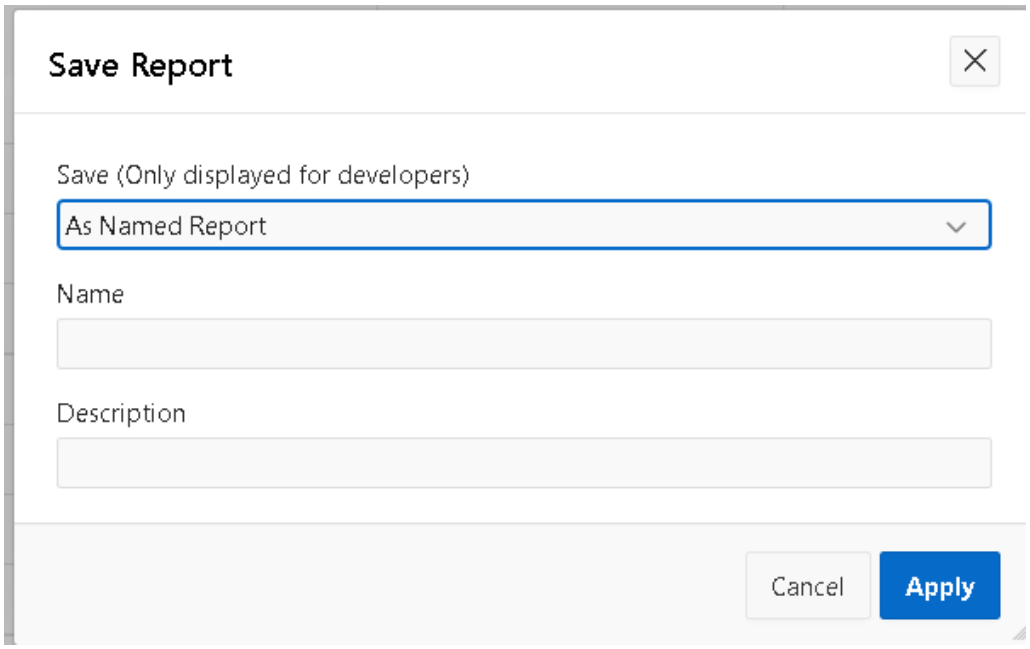
Slika 6.42: Grafikon u interaktivnom izvješću

Također, vrlo važna funkcionalnost interaktivnog izvješća je **Group By** prikazana je na Slici 6.43. Radi na isti način kao u SQL query.



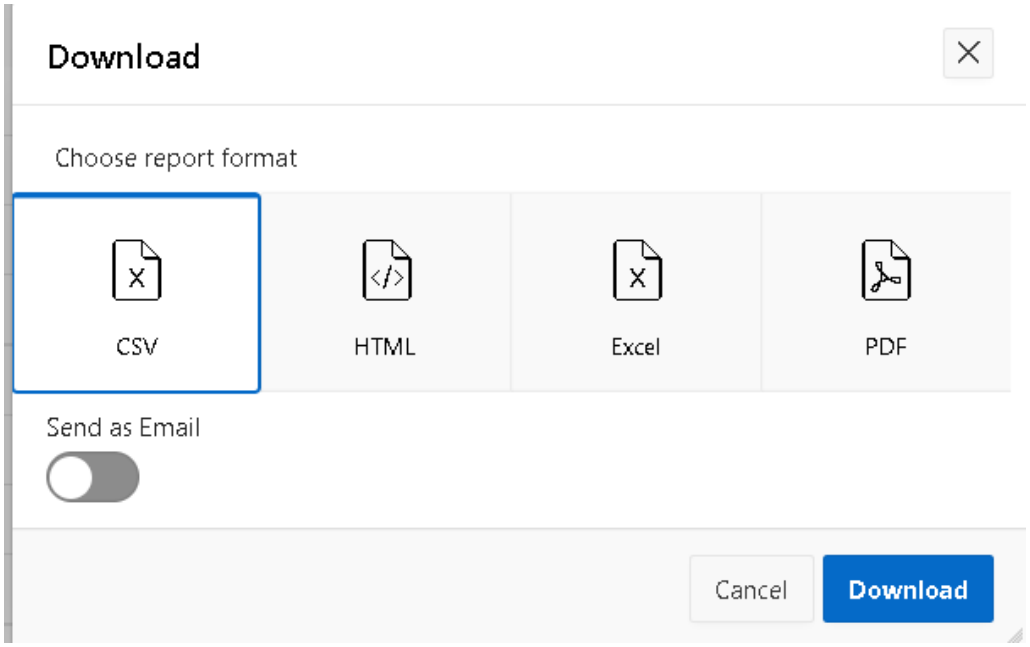
Slika 6.43: Group by

Save Reportprikazano na Slici 6.44 sprema izvješće u trenutnom stanju. Međutim, ovo izvješće može vidjeti samo programer nakon prijave u aplikaciju, a ne i korisnik.



Slika 6.44: Save report

Također možemo resetirati izvješće u izvorno stanje ili ga preuzeti u odabranom formatu. Ovu opciju možete vidjeti na slici 6.45.



Slika 6.45: Preuzimanje

6.5 Pitanja

1. Kojoj kategoriji pripada vrsta izvješća? Regije, stavke ili gumbi?
2. Što može biti izvor podataka za izvješće?
3. Koje se vrste izvora mogu koristiti za specificiranje klasičnog izvješća?
4. Koja je glavna značajka klasičnog izvješća?
5. Koja tehnologija omogućuje korisniku da dobije sliku podataka kakva je postojala u prošlosti pomoću interaktivnog izvješća?

6.6 Odgovori

1. Report je regija koja pojednostavljuje pregledavanje i pretraživanje podataka.
2. Podaci se mogu učitati iz lokalne baze podataka, REST omogućenog SQL-a ili REST izvora.
3. Dostupne vrste izvora su tablica ili SQL query.
4. Glavna značajka klasičnog izvješća je način rada samo za čitanje, koji onemogućuje korisniku da mijenja opcije prikaza ili podatke.
5. Flashback funkcije omogućuju korisniku da konstruira sliku podataka kakva je postojala u definiranoj vremenskoj točki.

7. Kako upravljati obrascima?

VERONIKA ŠALGOVÁ, MICHAL KVET, MIROSLAV POTOČÁR

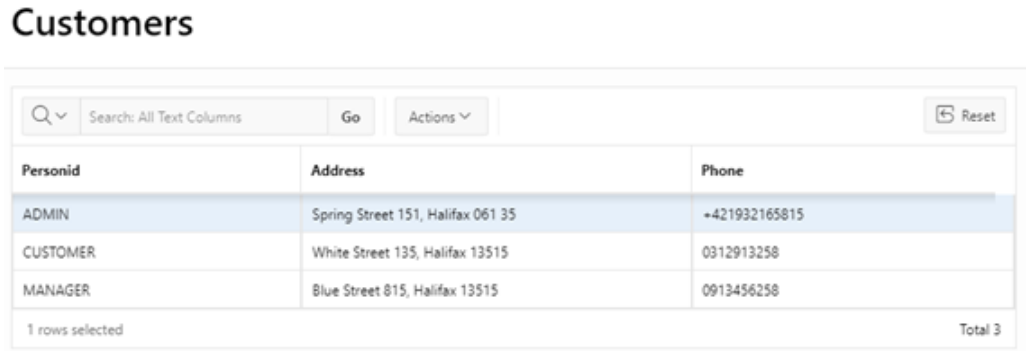
7.1 Kako upravljati obrascima

U Oracle APEX-u, možemo koristiti tri osnovna tipa obrazaca:

- interaktivnu mrežu koja se može uređivati,
- obrazac nad tablicom,
- master-detail obrazac.

7.2 Interaktivna mreža koja se može uređivati

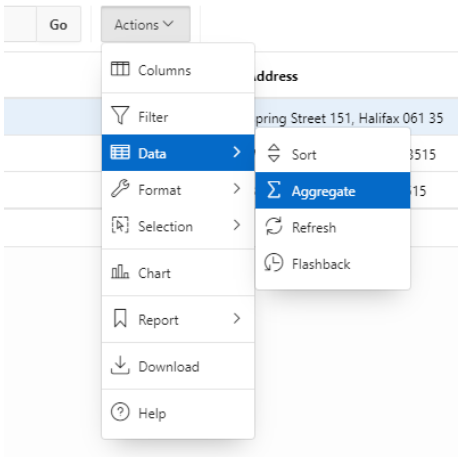
U interaktivnom izvješću koje je moguće uređivati mogu se dodavati, mijenjati, brisati i pretraživati podaci. To je komponenta bogata značajkama s mogućnošću uređivanja podataka dostupnim izravno na stranici. Osim toga, izvješće se može prilagoditi i interaktivno uređivati pomoću miša, izravno ažurirajući strukturu i sadržaj mreže. Moguće je izraditi interaktivnu mreže koja se može uređivati kao konverzija interaktivne mreže samo za čitanje, koja se kreira tako da se odabere *Report* te *Interactive Grid*. Druga opcija je kreirati *Report and Form* te tada odabrati da tip bude *Interactive Grid*, koji će također biti izrađen samo za čitanje, ali se može pretvoriti u oblik koji se može uređivati.



Slika 7.1: Interaktivna mreža s kupcima

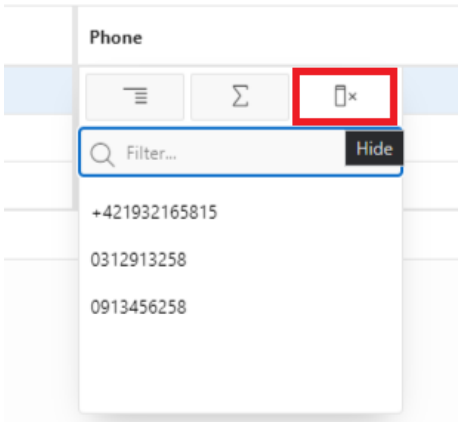
Podaci se mogu pretraživati pomoću teksta u traci za pretraživanje na vrhu stranice. Preostale

potencijalne akcije dostupne na mreži su prikazane na Slici 7.2.



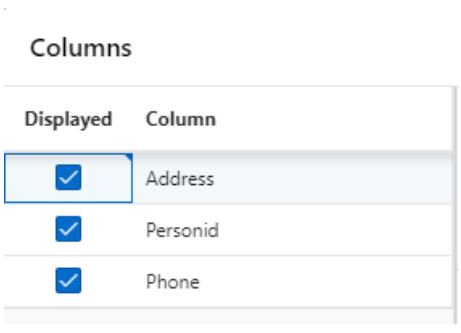
Slika 7.2: Akcije interaktivne mreže

Moguće je promijeniti veličinu stupca mreže klikom mišem i držanjem ruba naslova stupca te podešavanjem pomoću miša. Također se može sakriti stupac odabirom naslova i *Hide* ikone, kao što je prikazano na Slici 7.3.



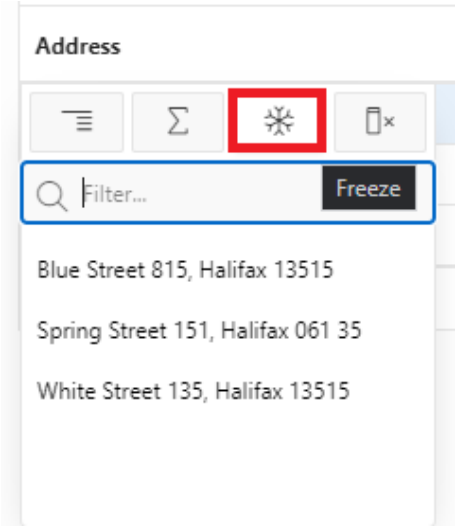
Slika 7.3: Hide ikona na stupcu Phone

Nakon skrivanja stupca, više se ne prikazuje u mreži. Da biste ponovo vidjeli stupac, morate kliknuti na *Actions*, odabrati *Columns* opciju, te kliknuti na stupce kako bi se prikazali.



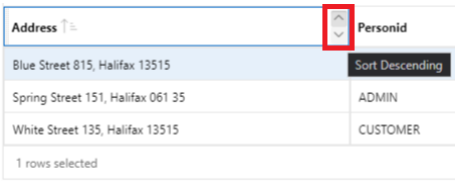
Slika 7.4: Prikazani stupci

Stupac se može zamrznuti odabirom *Freeze* ikone, kako je prikazano na slici 7.5. Zamrzavanje stupca isključuje ga iz područja pomicanja.



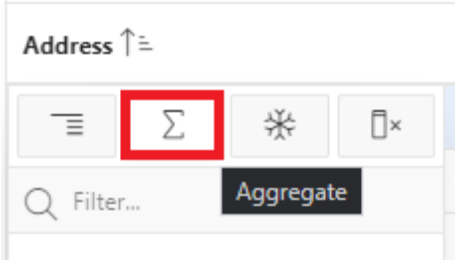
Slika 7.5: Freeze ikona stupca Address

Podaci se mogu sortirati prema stupcu odabirom *Sort Ascending* ili *Sort Descending* ikone, kako je prikazano na Slici 7.6.



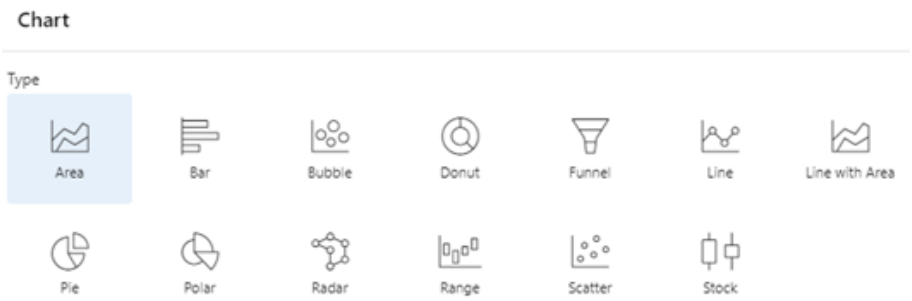
Slika 7.6: Sortiranje stupca Address

Upotrebom *Aggregate* ikone na stupcu, prikazano na Slici 7.7, mogu se odabrati funkcije Count, Count Distinct, Minimum i Maximum te primijeniti nad odabranim stupcem.



Slika 7.7: Aggregate ikona na stupcu Address

Podaci se mogu koristiti za kreiranje grafikona, odabirom *Actions* i *Charts* opcije. Postoje različite vrste grafova, a prikazane su na Slici 7.8. Također se mogu odabrati agregacijske funkcije koje se nalaze uz grafikone.



Slika 7.8: Vrste grafikona

7.3 Obrazac nad tablicom

Oracle APEX daje osnovne obrasce za umetanje retka u tablicu, prikazano na Slici 7.9. Odabirom *Create a Page* gumba, bira se *Form* opcija. Treba odabrati tablicu povezanu s kreiranim obrascem ili napisati svoj SQL upit i stupac primarnog ključa tablice.

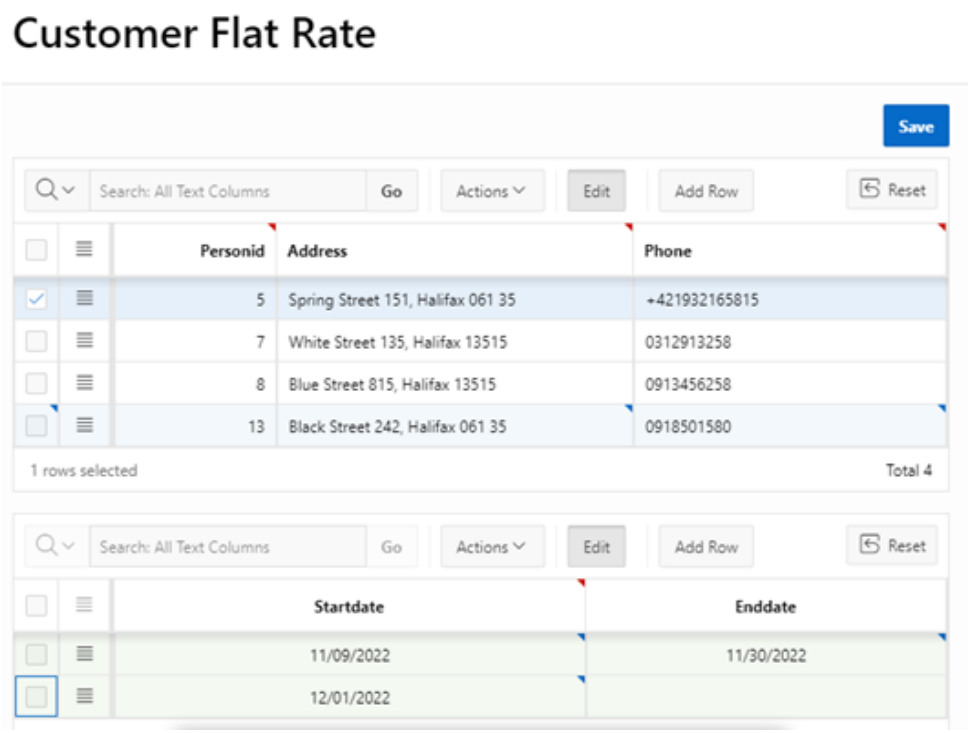


Slika 7.9: Obrazac za umetanje kupaca

7.4 Master Detail obrazac

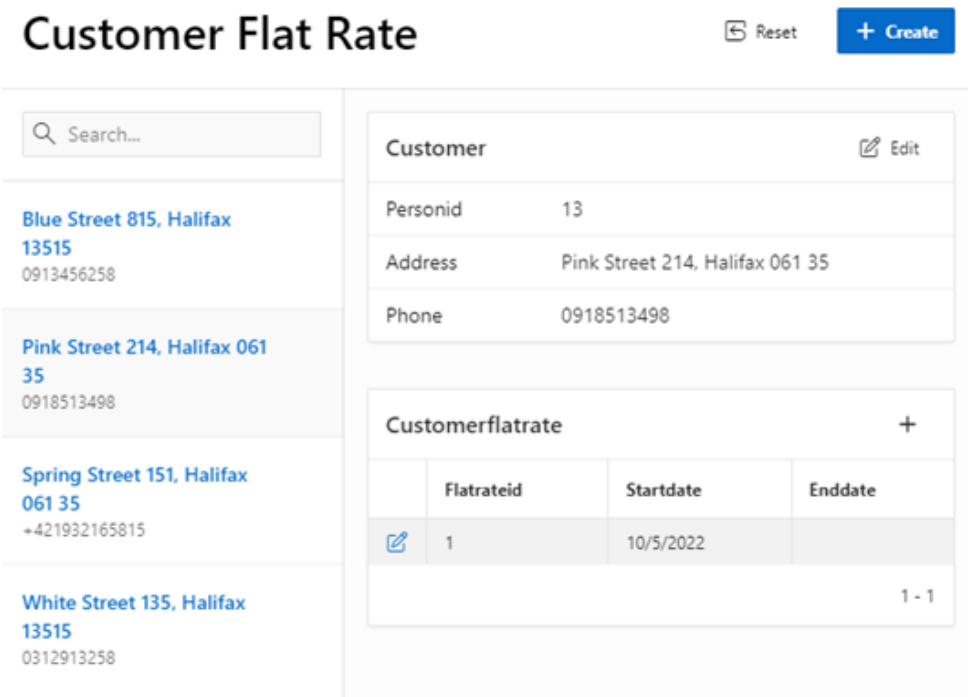
Moguće je postavljati upite, umetati, ažurirati i brisati vrijednosti iz dvije povezane tablice ili pogleda. Nakon odabira *Create a Page* gumba, treba ići na *Master Detail* opciju. Potom se odabire neki od master-detail stilova, kao što su *Stacked*, *Side by Side*, ili *Drill Down*. Obrazac master detail obično prikazuje glavni redak i više redaka detalja. To znači da treba odabrati tablicu ili prikaz koji će biti glavni izvor podataka, a drugi će biti izvor detalja. Nakon odabira primarnog ključa svake tablice, potrebno je odabrati vanjski ključ master detaila-a.

Slika 7.10 prikazuje *Stacked* master detail obrazac. Sadrži interaktivnu mrežu koju je moguće uređivati. Korisnici mogu odabrati red u glavnoj mreži za ažuriranje mreže detalja.



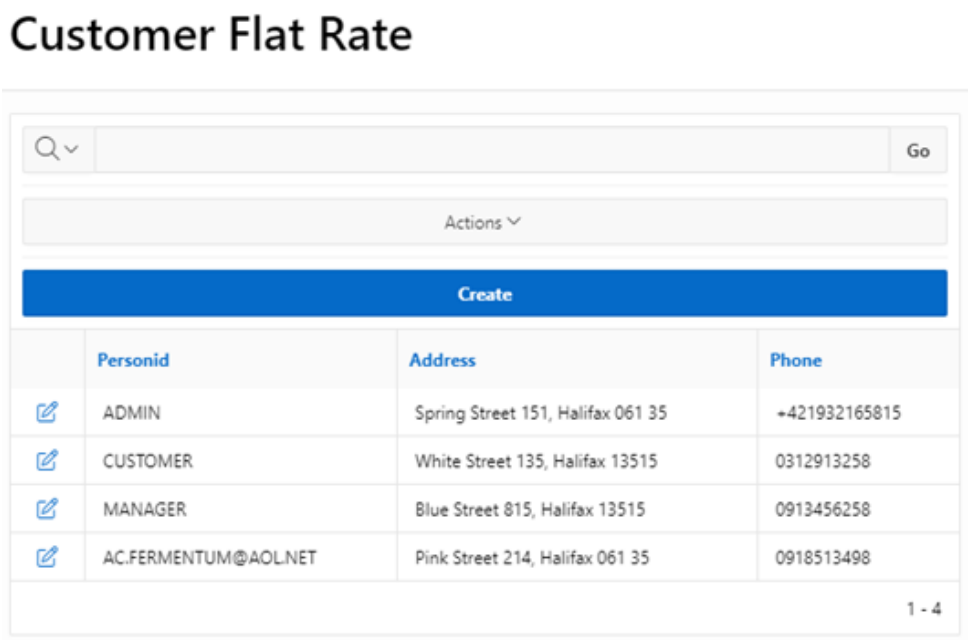
Slika 7.10: Stacked master detail obrazac za Customer Flat Rates

Na slici 7.11, se nalazi *Side by Side* master detail obrazac koji sadrži master detail jedne stranice koristeći prikaz jedan pored drugog i regije izvješća s modalnim prozorima za uređivanje. Na lijevoj strani može se master popisa navigarati do master zapisa. Desna strana sadrži odabrani master zapis i povezana detail izvješća.

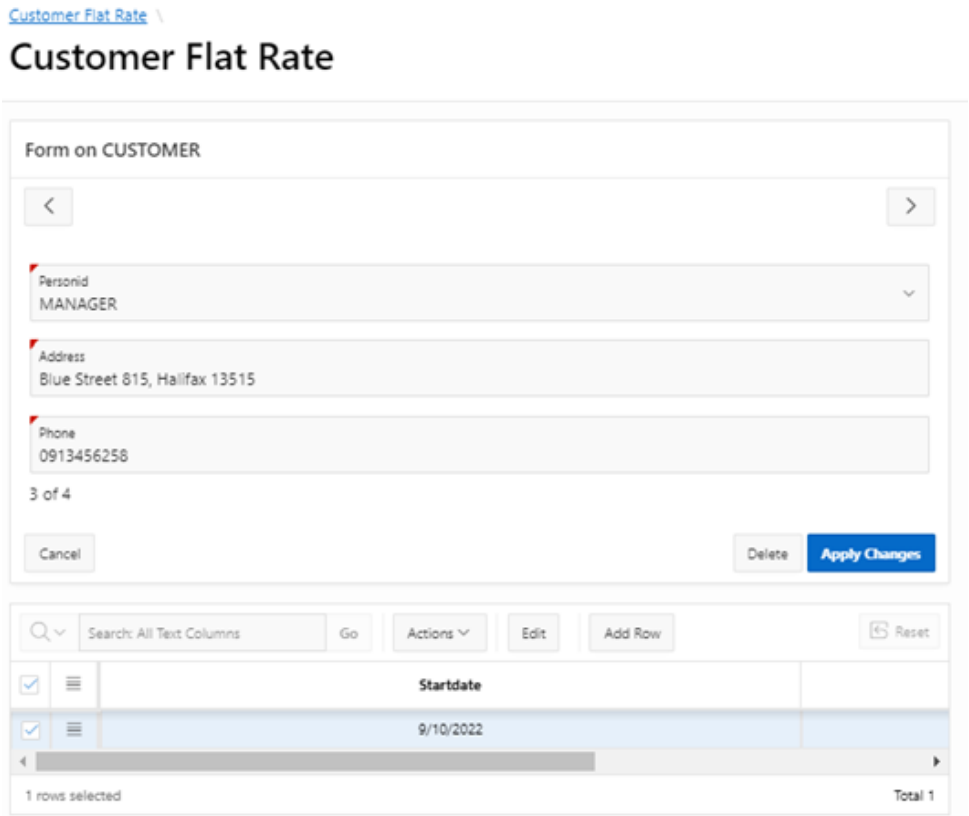


Slika 7.11: Side by Side master detail obrazac Customer Flat Rates

Master detail *Drill Down* sadrži dvije stranice. Prva stranica sadrži interaktivno izvješće za master tablicu, prikazano na Slici 7.12. Na drugoj stranici, prikazano na Slici 7.13, nalazi se standardni obrazac za master i uređivačka interaktivna mreže za detail.



Slika 7.12: Drill Down master detail obrazac – prva stranica



Slika 7.13: Drill Down master detail obrazac – druga stranica

7.5 Pitanja

1. Je li moguće rasporediti podatke pomoću uređivačke interaktivne mreže?
2. Koju definiciju komponente tablice treba odabrati prilikom kreiranja obrasca?
3. Koja opcija isključuje stupac iz područja koje se može pomicati?

7.6 Odgovori

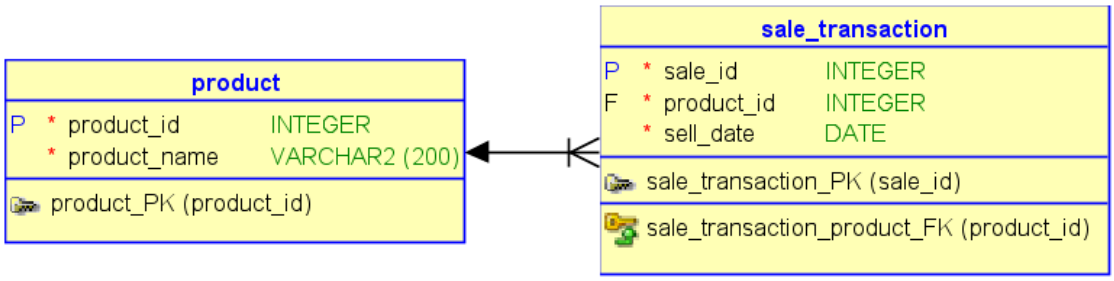
1. Da, izvješća se mogu prilagoditi i interaktivno preuređivati pomoću miša, izravno ažurirajući strukturu i sadržaj mreže.
2. Atributi primarnog ključa moraju biti specificirani.
3. Stupac se može zamrznuti nakon klika na ikonu Zamrzni. Zamrzavanje stupca isključuje ga iz područja pomicanja.

8. Kako pretvoriti tekstualna izvješća u grafikone?

IVAN PASTIERIK, MICHAL KVET, MIROSLAV POTOČÁR

8.1 Grafikon

Regije grafikona u Oracle APEX-u su komponente koje omogućuju programerima i korisnicima da steknu bolji uvid u podatke kroz vizualni prikaz. Ovi dinamički grafikoni, u rasponu od trakastih i linijskih grafikona do grafikona pita, služe kao važna sredstva u poboljšanju korisničkog iskustva i olakšavanju donošenja odluka na temelju podataka. U ovom ćete poglavlju naučiti kako izraditi jednostavan grafikon kroz **filtering** i **sorting** opcije. **Filtering** i **sorting** može uvelike poboljšati interaktivnost i čitljivost grafikona. U tu svrhu izradili smo jednostavan podatkovni model prikazan na Slici 8.1. Podatkovni model sastoji se od entiteta: **"product"** i **"sale_transaction"**. Entitet **"product"** predstavlja proizvod, s atributima kao što su **"product_id"**, te ime proizvoda spremljeno u atributu **"product_name"**. Entitet **"sale_transaction"** pohranjuje informacije o prodanim proizvodima s primarnim ključem **"sale_id"**. Atribut **"product_id"** je vanjski ključ koji se referencira na atribut **"product_id"** u entitetu **"product"**. Atribut **"sell_date"** sadrži datum kada je proizvod prodan.

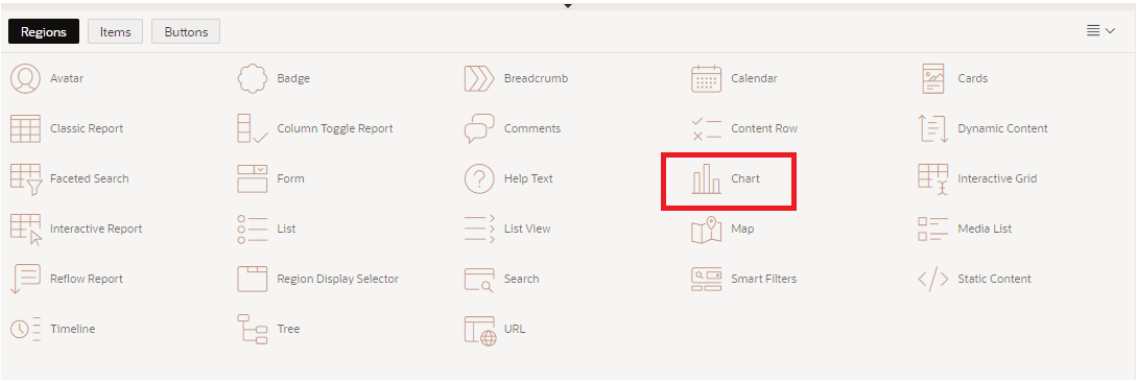


Slika 8.1: Podatkovni model korišten u aplikaciji

Za početak je potrebno izraditi novu aplikaciju koju ćemo nazvati **"Chart Showcase"**. Aplikacija će se sastojati od **"Global Page"**, **"Home"** i **"Login Page"**. Grafikone ćemo kreirati na home page. Detaljne upute za izradu aplikacija opisane su u poglavlju 5. Nakon kreiranja aplikacije, unesite **Page Designer** za **"Home"** stranicu unutar kreirane apliakcije.

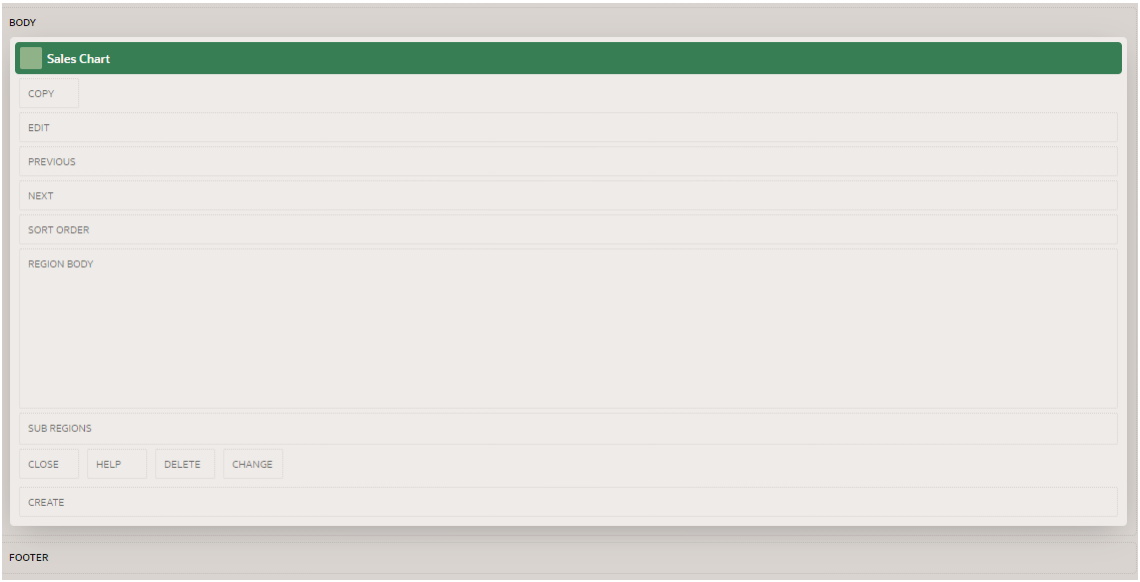
8.2 Kreiranje stupčastog grafikona

Za postavljanje regije grafikona, označene na slici 8.2, u aplikaciju, moramo ga povući u sekciju **Page Designer**.



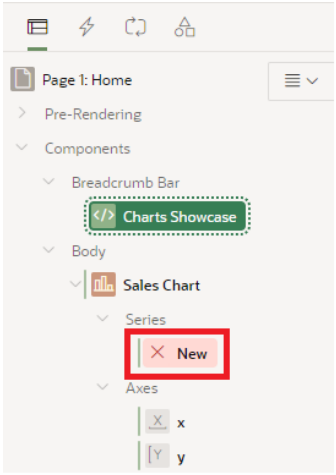
Slika 8.2: Selektiranje Chart region

Smjestit ćemo **Chart** reigiju u **body** sekciju stranice da smjestimo grafikon u središnji sadržaj stranice. Smjestit ćemo **title** na Chart region u **"Sales Chart"**. Na Slici 8.3, možete vidjeti kako izgleda odjeljak rasporeda nakon postavljanja i promjene naslova **Chart** regije.



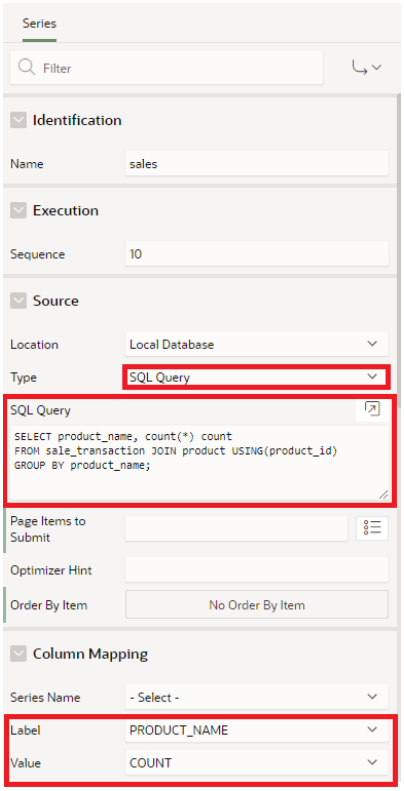
Slika 8.3: Sekcija Layout nakon kreiranja Chart regije.

Na lijevoj strani **Page Designer**, možete vidjeti **tree overview** komponenti. Ako pogledamo **Sales Chart** komponentu, unutar **"Series"**, pokazuje grešku. Ova greška, prikazana na Slici 8.4, nastala je iz razloga jer nismo definirali izvor za podatke koje želimo prikazati u grafikonu.



Slika 8.4: Pregled stabla prikazuje grešku.

Kada kliknete na seriju označenu kao **"New"** iz pregleda stabla otvorit će se izbornik postavki za seriju s desne strane **Page Designer**. Unutar ovog izbornika postavki promijenit ćemo naziv serije u **"sales"** jer želimo prikazati broj prodaje naših pojedinačnih proizvoda u **Sales Chart**. Sljedeće treba definirati serije **source**. Da bismo to učinili, promijenit ćemo vrstu izvora u **"SQL Query"** i unutar SQL Query, smjestit ćemo **SELECT** izjavu prikazanu na slici 8.5. Ova **SELECT** izjava vratit će ukupan broj prodaja grupiran po nazivima proizvoda. Sada kada smo definirali izvor za naše podatke koje želimo prikazati na grafikonu, moramo mapirati stupce koje je odabrano kroz **SELECT** izjavu na osima **Label (X)** i **Value (Y)**. Konačna konfiguracija prikazana je na slici 8.5 s označenim važnim promjenama.



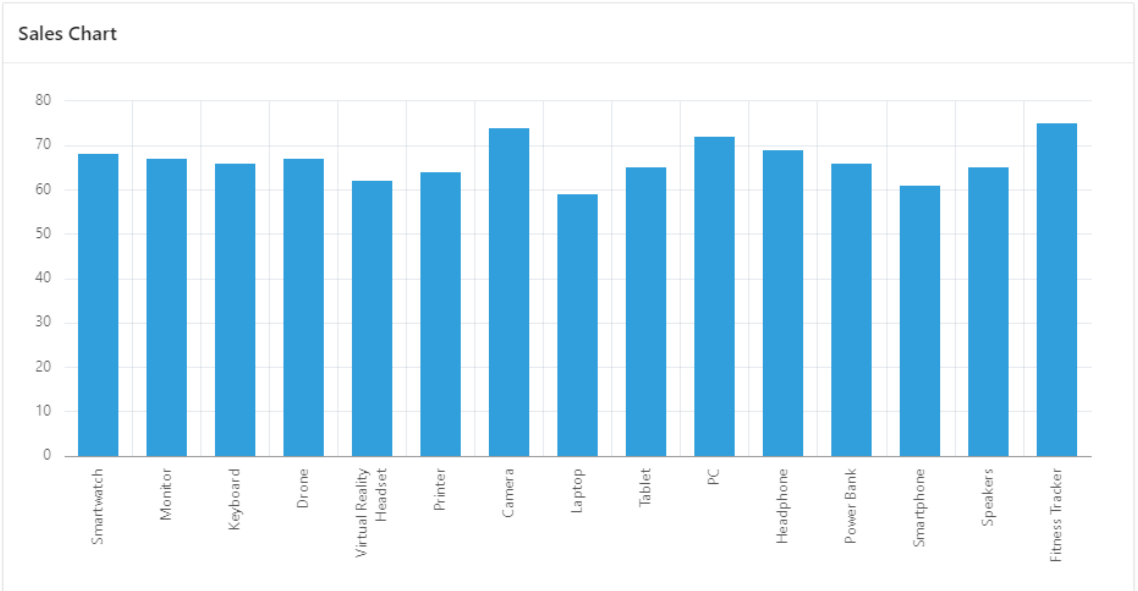
Slika 8.5: Finalna konfiguracija serije

Ako uzmemo SQL upit sa Slike 8.5 i izvšrimo ga, vraća rezultat kao na Slici 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

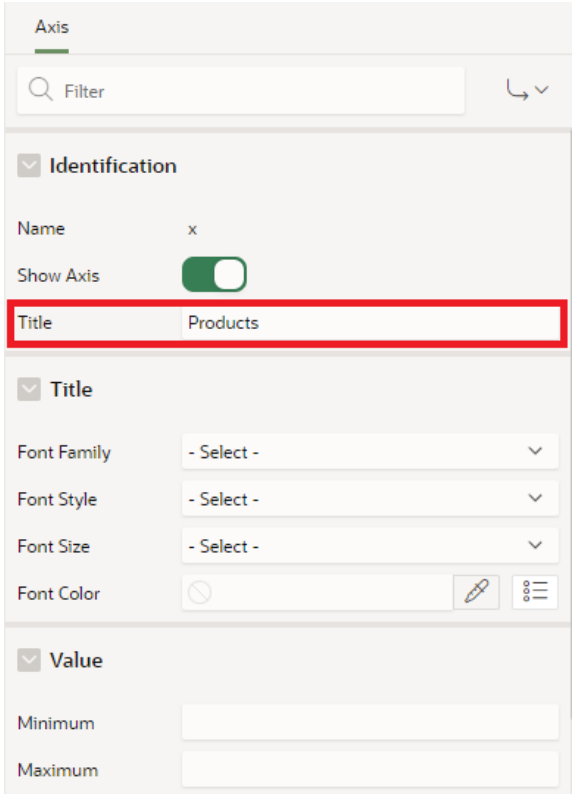
Slika 8.6: Podaci vraćeni izvršavanjem SQL upita

Sada možemo spremiti našu stranicu i pokrenuti je. Ne zaboravite postaviti stranicu **authentication** na **"Page Is Public"** tako da ne zahtijeva prijavu. Kada pokrenete stranicu, trebali biste dobiti rezultate poput onih prikazanih na Dlici 8.7. Međutim, možda ćete primijetiti da naš grafikon nema oznaku **X Y** osi, pa se teže čita.



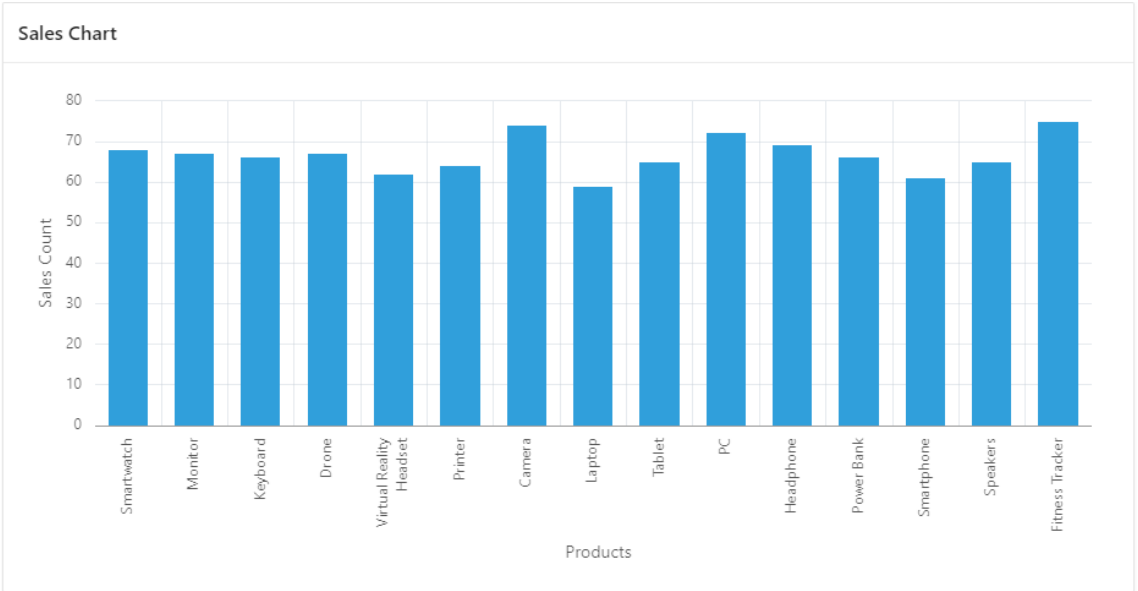
Slika 8.7: Grafikon prikazan prilikom pokretanja aplikacije

Kako bismo to riješili, možemo označiti osi slično kao što smo označili serije. Klikom na pojedine osi s lijevog pregleda stabla i promjenom **Title** atributa, prikazano na Slici 8.8, možemo nazvati **X** os **"Products"** i **Y** os **"Sales Count"**.



Slika 8.8: Promjena atributa naslova za x os

Nakon što označimo obje osi našeg grafikona, rezultat bi trebao nalikovati grafikonu prikazanom na Slici 8.9.

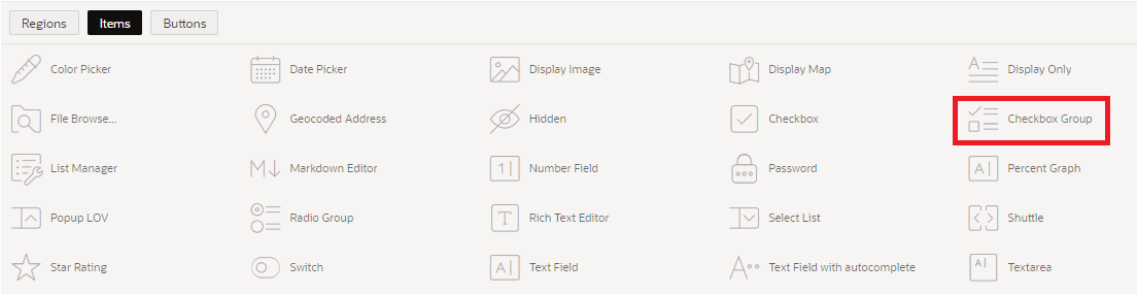


Slika 8.9: Grafikon prodaje nakon označavanja osi

8.3 Dodavanje filtriranja na stupčasti grafikon

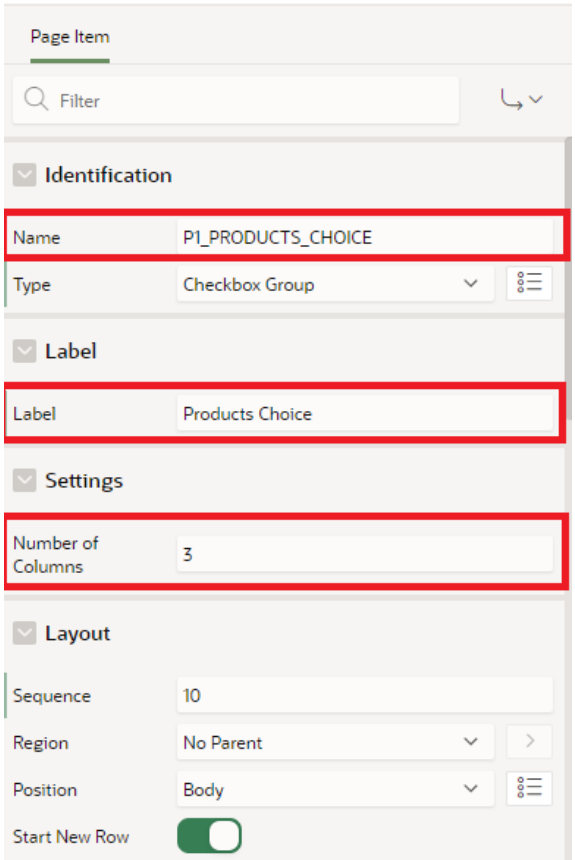
Sad ćemo dodati **filtering** na proizvode na naš grafikon, što nam omogućuje da odaberemo koje ćemo proizvode prikazati. Da bismo to postigli, dodat ćemo **Checkbox Group** stavku s donje

alatne trake **Page Designer**, smještajući iznad **Sales Chart** regije.



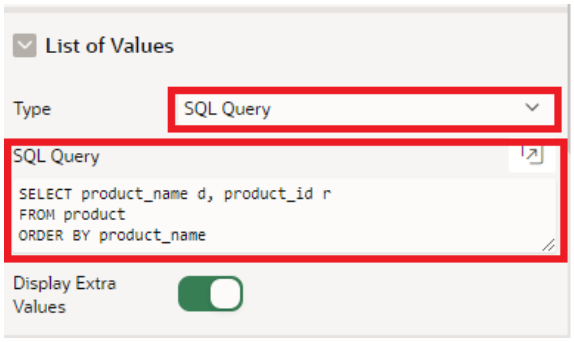
Slika 8.10: Odabir Checkbox Group stavke

Nazvat ćemo **Checkbox Group** stavku **"P1_PRODUCTS_CHOICE"** i oznaku na **"Products Choice"**. Dodatno, možemo prilagoditi izgled tako da ga prikazuje u tri stupca, što je prikazano na Slici 8.11.



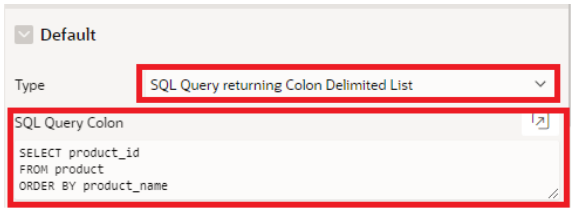
Slika 8.11: Postavljanje temeljnih atributa na Checkbox Group stavku

Popunjavanje **Checkbox Group** s proizvodima, moramo definirati **list of values**. U**"List of Values"** sekciji **Page Item** postavki, promijenit ćemo **"Type"** atribut na **"SQL Query"** i napisati **SELECT** izjavu prikazanu na Slici 8.12. Ova izjava treba odabrati točno dva stupca: prvi stupac koji sadrži vrijednosti prikaza (s aliasom **"d"**), i drugi stupac koji sadrži povratne vrijednosti (s aliasom **"r"**).



Slika 8.12: Postavljanje liste vrijednost na Checkbox Group stavku

Inicijalno, **Checkbox Group** obično nemaju sve potvrdne okvire kada pokrenete aplikaciju. Međutim, radi lakšeg korištenja kada se koristi s grafikonima, poželjno je da korisnici vide sve proizvode prikazane u grafikonu prema zadanim postavkama. Da bismo to postigli, ići ćemo na **"Default"** sekciju **Checkbox Group** postavku sekcije na **"Type"** atribut na **"SQL Query returning Colon Delimited List"**. Ovaj tip SQL Query omogućuje nam da definiramo koje su vrijednosti odabrane prema zadanim postavkama. Regularni SQL Query ne može se upotrijebiti s **Checkbox Group** stavkom jer može definirati samo jednu zadanu vrijednost, a ne više zadanih vrijednosti. Unutar**"SQL Query Colon"**, napisat ćemo **SELECT** izjavu na Slici 8.13. Ova **SELECT** izjava trebala vratiti točno jedan stupac, te će se interno konvertirati u **Colon Delimited List**, koje će izgledati: **"1:5:7:2:3"**.

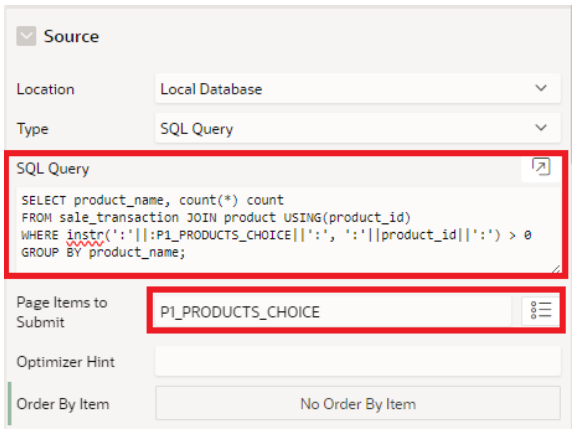


Slika 8.13: Postavljanje inicijalnih vrijednosti Checkbox Group stavke

S **Checkbox Group** stavkom, možemo ga povezati s našom **Sales Chart** regijom. Stoga ćemo dodati naziv **Checkbox Group** stavke **"Page Items to Submit"** polja na **"Source"** sekciji serije **sales** unutar **Sales Chart** regije. To je **zahtjevano** kako bi se dodala stavka SQL Query na **"Page Items to Submit"**. Povratna vrijednost **"P1_PRODUCTS_CHOICE"** je **Colon Delimited List**, koja se sastoji od označenih proizvoda. Stoga treba koristiti funkciju **INSTR**, koji vraća indeks prvog pojavljivanja podniza definiranog u drugom parametru funkcije unutar niza definiranog u prvom parametru funkcije.

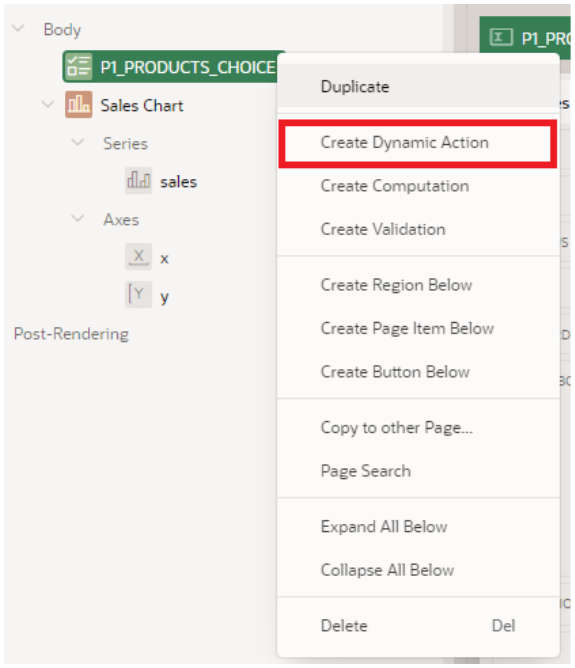
INSTR funkcija vraća **0**, ako nema pojavljivanja u nizu. Koristit ćemo **INSTR** funkciju da se označi proizvod **"product_id"** nalazi se unutar odabranog popisa ID-ova proizvoda u **"P1_PRODUCTS_CHOICE"**. Funkcija **INSTR** pretražuje pojavljivanje **':'||product_id||':'** unutar ulančanog niza **':'||:P1_PRODUCTS_CHOICE||':'**. Ako pronađe podudaranje, to znači da je proizvod među odabranim proizvodima. Moramo se ulančati **':'** na **"product_id"** i **":P1_PRODUCTS_CHOICE"** jer želimo izvršiti precizno podudaranje između pojedinačnih ID-ova proizvoda. Time osiguravamo da npr, **":1: "** je povezan s **":1:2:3:8:9:11: "**. To osigurava da tražimo točna podudaranja, osiguravajući da se svaki odabrani ID proizvoda razmatra neovisno. Ako povezujemo **"1"** s **"1:2:3:8:9:11"**, potencijalno bi prikazao proizvode s oba ID-a **"1"** i **"11"**, kao i svaki drugi ID proizvoda koji sadrži **"1"** kao podniz. To bi moglo dovesti do neželjenih rezultata i netočnog prikaza podataka. Dodavanjem **':'** na obje strane ID-ova, stvaramo dosljedan uzorak razgraničenja, osiguravajući da se odabrani ID-ovi proizvoda precizno uspoređuju i eliminiramo bilo kakvu dvo-

smislenost u procesu podudaranja. To osigurava da grafikon točno prikazuje podatke na temelju odabira korisnika.



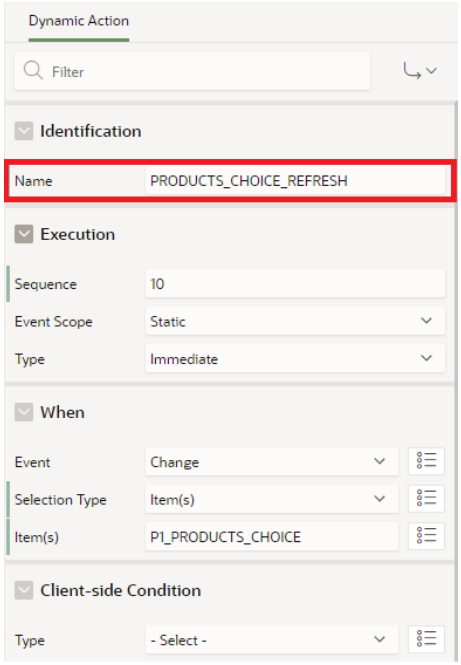
Slika 8.14: Povezivanje Sales Chart regije s Checkbox Group stavkom.

Posljednji korak konfiguracije je postavljanje osvježavanja **Sales Chart** regije kada dođe do promjene u odabiru **Checkbox Group** stavke.Da bi se to postiglo, treba kreirati **Dynamic Action** na**"P1_PRODUCTS_CHOICE"** Checkbox Group stavci,kako je prikazano na Slici 8.15.



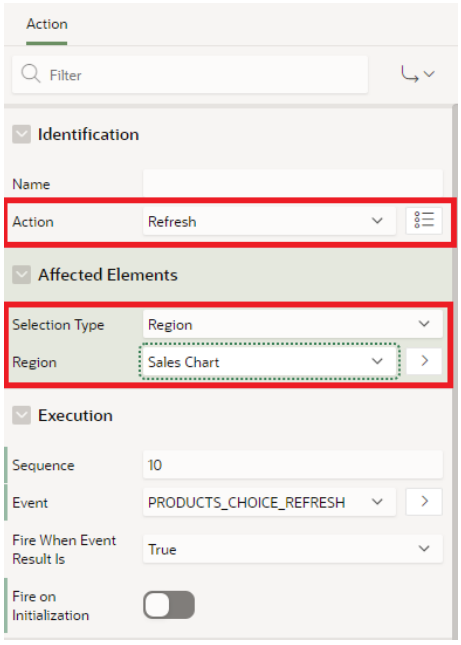
Slika 8.15: Kreiranje Dynamic Action za P1_PRODUCTS_CHOICE stavku

Možemo postaviti **"Name"** atribut **Dynamic Action** na **"PRODUCTS_CHOICE_REFRESH"** prikazano na Slici 8.16. Također moramo osigurati da**"Event"** atribut unutar **"When"** sekcije bude postavlje na **"Change"** i **"Selection Type"** atribut postavljen na **"Item(s)"** i **"Item(s)"** atribut postavljen na trenutno ime stavke, te će se **Dynamic Action** otpustiti kada se vrijednost **"P1_PRODUCTS_CHOICE"** promijeni.



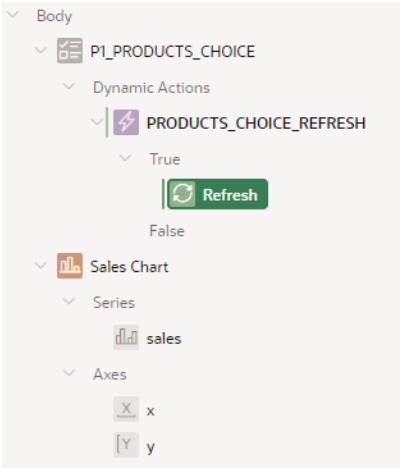
Slika 8.16: Postavljanje osnovnih atributa promjene dinamičke radnje

Unutar pregleda stabla na lijevoj strani **Page Designer**, kliknite radnju koja se izvršava kada je uvjet **Dynamic Action true**, što se događa kada se **Checkbox Group** stavka promjeni. Kada stvorite novi **Dynamic Action**, vidjet će te **"Show"** akcoju unutar greške, te to treba odabrati.U postavkama ove akcije, postavitee **"Action"** atribut unutar **"Identification"** sekcije na **"Refresh"**. Na **"Affected Elements"** sekciji, promijenite **"Selection Type"** na **"Region"** i tada postavite **"Region"** na **"Sales Chart"**, kao što je prikazano na 8.17.



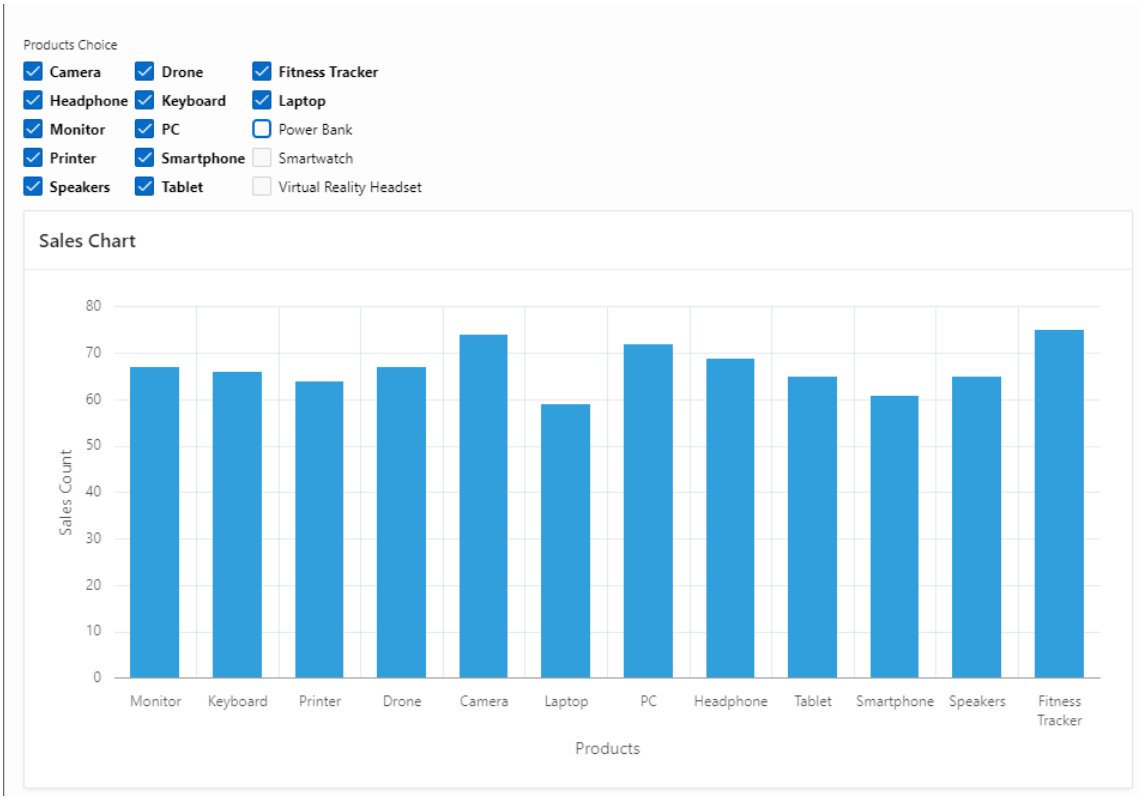
Slika 8.17: Postavljanje atributa na Refresh action

Slika 8.18 prikazuje kako vaš pregled stabla u lijevom dijelu **Page Designer** treba pogledati jeste li ispravno slijedili sve korake.



Slika 8.18: Dynamic Action za osvježavanje Sales Chart u pregledu stabla.

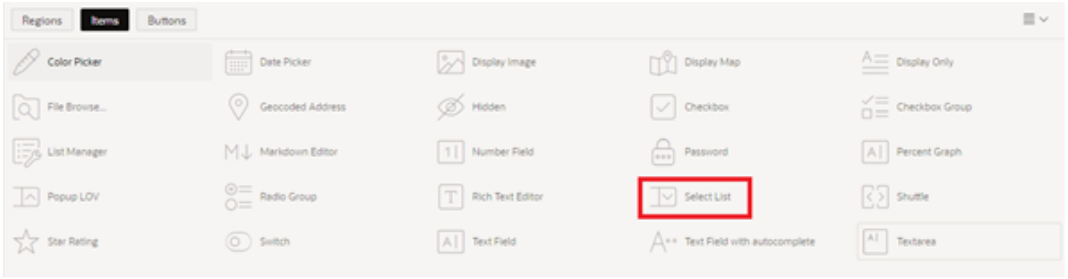
Kada pokrenemo i testiramo aplikaciju, **filtering** i **refreshing** da bi graf bio potpuno funkcionalan, aplikacija bi trebala biti slična onoj prikazanoj na Slici 8.19.



Slika 8.19: Aplikacija nakon filtriranja proizvoda

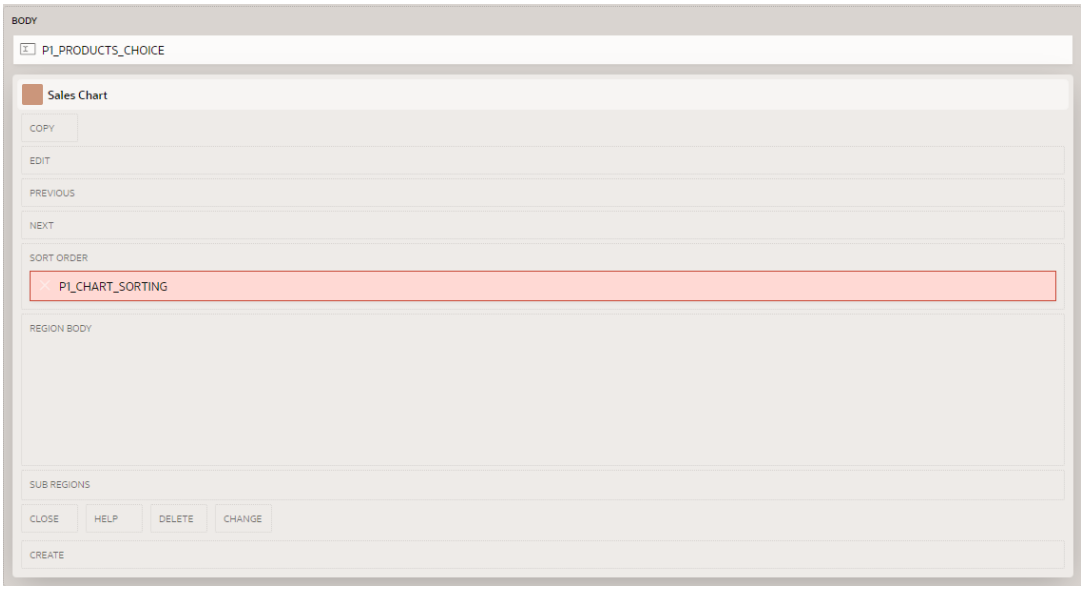
8.4 Dodavanje sortiranja na stupčasti grafikon

Možemo poboljšati korisničko iskustvo dodavanjem mogućnosti za korisnike da sortiraju vrijednosti prikazane u grafikonu prema različitim kriterijima. Za odabir kriterija sortiranja koristit ćemo **Select List** stavku, prikazano na Slici8.20.



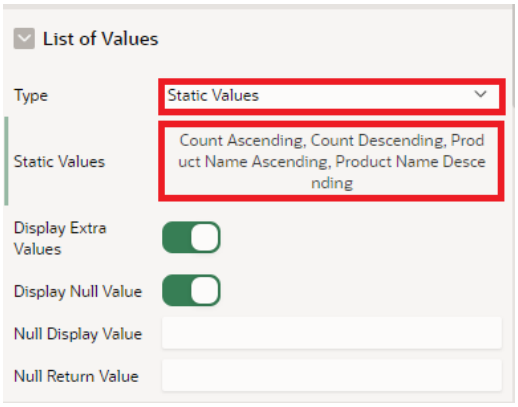
Slika 8.20: Odabir Select List stavke

Smjestit ćemo **Select List** stavku unutar **Sales Chart** regije unutar **"Sort Order"** sekcije i nazvat ćemo je **"P1_CHART_SORTING"**, prikazano na Slici 8.21.



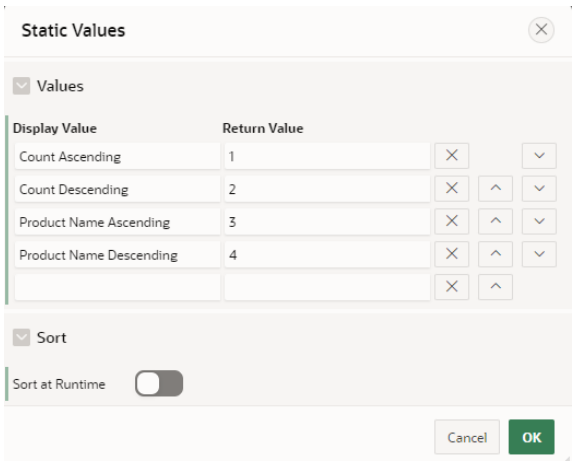
Slika 8.21: Sales Chart regija nakon preimenovanja Select List stavke

Sada moramo definirati popis vrijednosti za Select List stavku. U postavkama **"P1_CHART_SORTING"** stavke, moramo promijeniti **"Type"** atribut unutar **"List of Values"** sekcije na **"Static Values"**. Tada moramo odabrati gumb koji pokazuje inicijalne vrijednosti, **"Display1, Display2"** odmah do **"Static Values"** atributa.



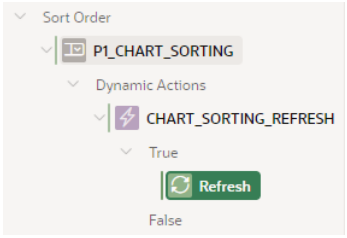
Slika 8.22: Postavljanje liste vrijednosti za Select List stavke

Sada možete vidjeti da se otvorio novi prozor s naslovom **"Static Values."** . Unutar ovog prozora moramo definirati prikaz i povratne vrijednosti. Unutar **display** vrijednosti, napisat ćemo opise sortiranja, i unutar **return** vrijednosti, zapisivat ćemo brojeve koji počinju s **1**, prikazano na Slici 8.23. Ako želimo da ove vrijednosti prikaza budu poredane kako smo ih unijeli od vrha prema dolje, tada moramo onemogućiti **"Sort at Runtime"** opciju.



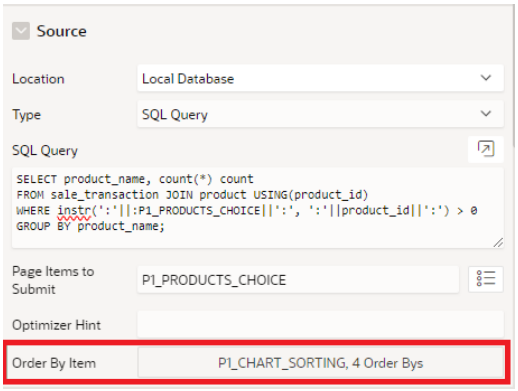
Slika 8.23: Postavljanje statičkih vrijednosti Select List stavke

Moramo dodati **Sales Chart** osvježavanjem **Dynamic Action** na isti način na koji smo ga postavili na **"P1_PRODUCTS_CHOICE"** stavci. Rezultat je na Slici 8.24.



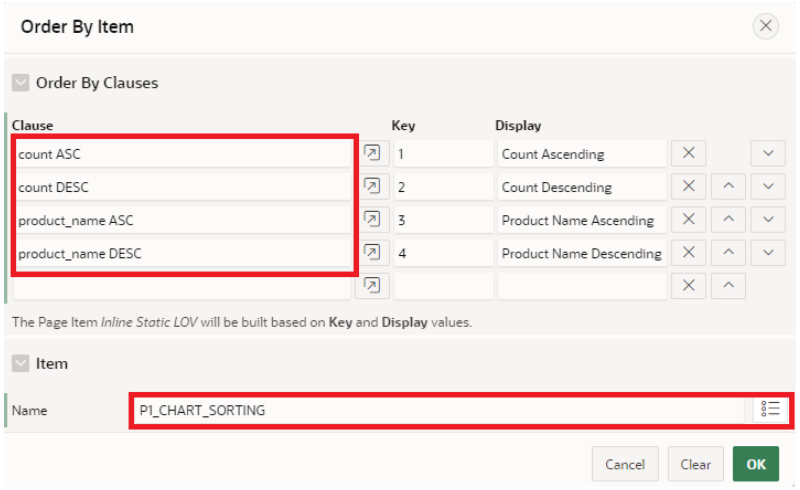
Slika 8.24: Dodavanje Dynamic Action na Select List stavci

Sada kada smo konfigurirali **Select List** stavku, moramo to povezati s **Sales Chart** regijom.To možemo učiniti pritiskom na gumb odmah pored **"Order By Item"** atributa, koji je inicijalno **"No Order By Item"**, naglašeno na Slici 8.25.



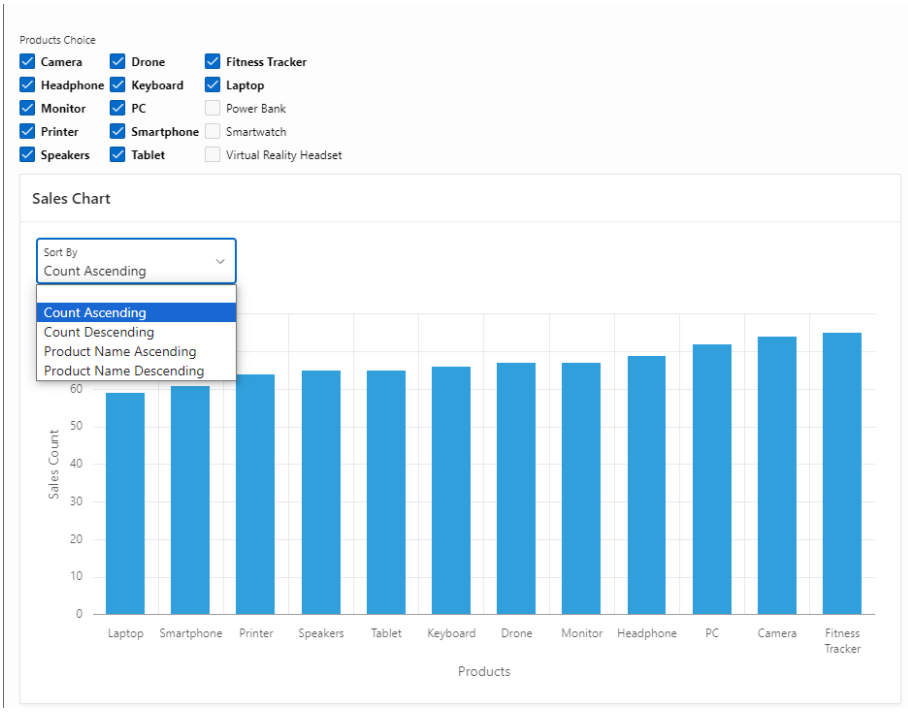
Slika 8.25: Postavljanje Order By Item za Sales Chart regiju.

Nakon pritiska na gumb, trebali biste vidjeti da se otvorio novi prozor s naslovom **"Order By Item."** . Unutar ovog prozora, prvo treba postaviti atribut **"Name"** unutar **"Item"** sekcije na **"P1_CHART_SORTING"**. Tada treba ispuniti "Clause" stupcu kao što se vidi na Slici 8.26. Ove klauzule su praktički **"ORDER BY"** klauzule korištene na **SELECT** izjavama, ali bez korištenja **"ORDER BY"** ključne riječi.



Slika 8.26: Postavljanje Order By klauzule za Select List stavke korišten za sortiranje

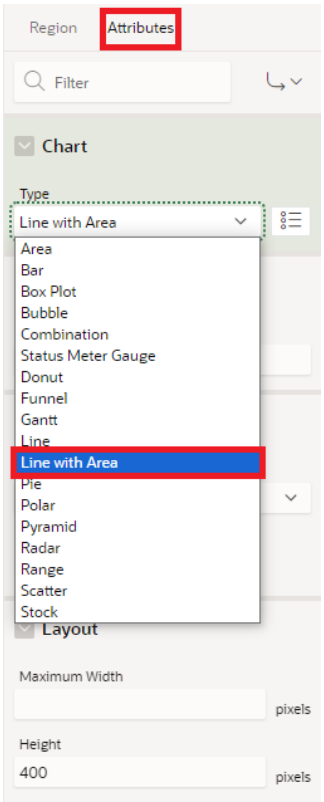
Ako ste pažljivo slijedili sve ove korake, nacrtani grafikoni **sorting** i **refreshing** trebali bi raditi kada spremite stranicu i pokrenete aplikaciju.



Slika 8.27: Aplikacija nakon implementacije sortiranja grafikona

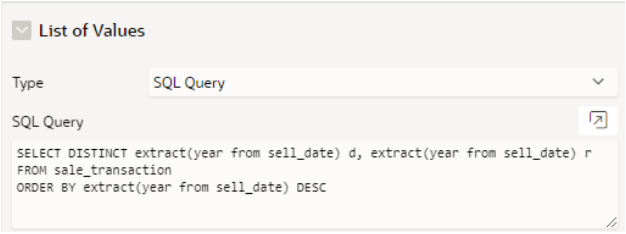
8.5 Kreiranje različitih tipova grafikona

U slučaju da želite implementirati drugu vrstu grafikona, to možete učiniti odabirom regije grafikona i odabirom **"Attributes"** taba na desnoj strani **Page Designer**, i mijenjanje vrijednosti **"Type"** atributa. Kreirat ćemo još jedan **Line with Area** graf prikazuje mjesečni broj prodaje, koji ćemo imenovati **"Monthly Sales Chart"**, i moći ćemo ga filtrirati po godinama.



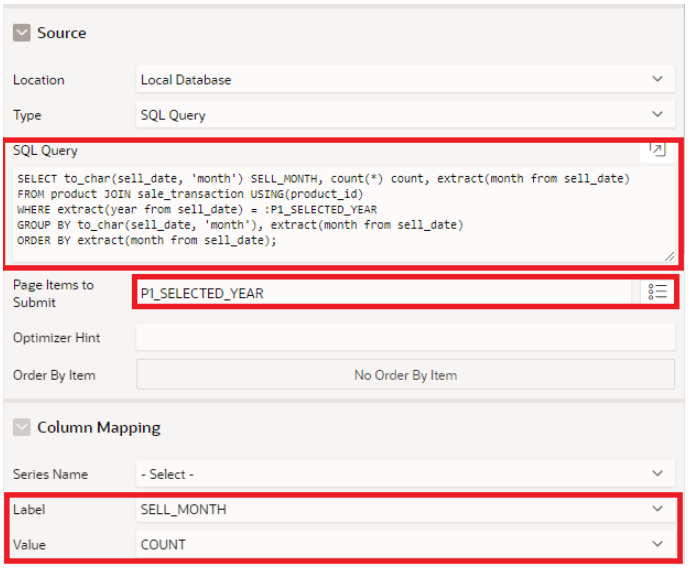
Slika 8.28: Mijenjanje tipa grafa na Line with Area.

Sljedeći korak je dodavanje **Select List** stavke koje će se koristiti na **filter** grafikonu po **godini**. Smjestit ćemo ovu stavku unutar **"Monthly Sales Chart"** regije na **"Edit"** sekciji. SQL Query korišten za dobivanje liste vrijednosti prikazane na Slici 8.29.



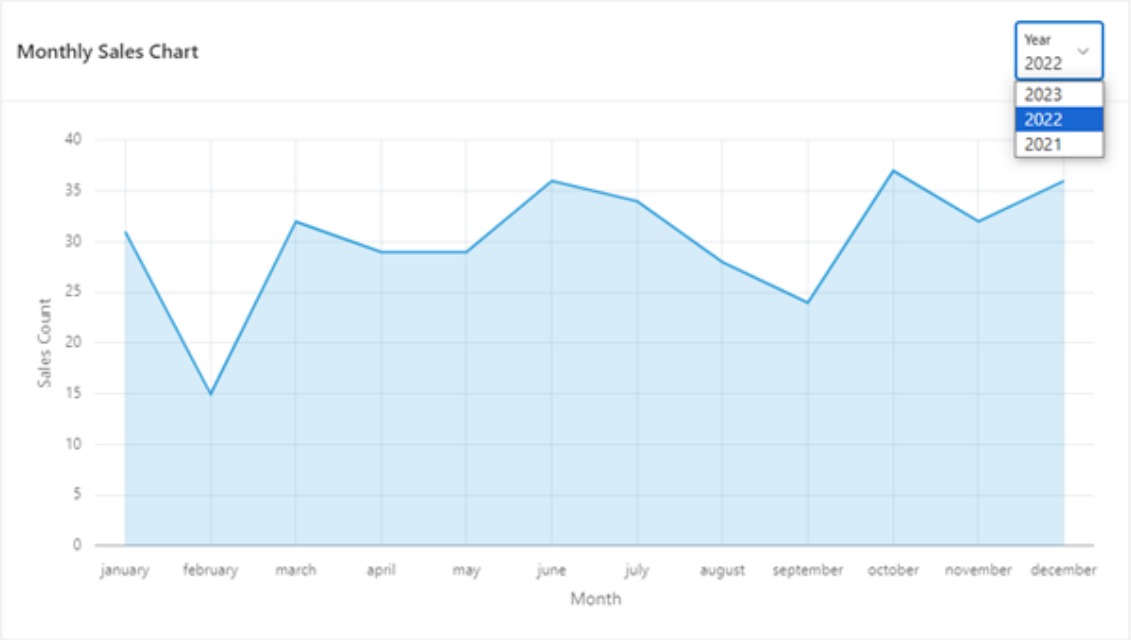
Slika 8.29: List of Values za Select List stavke korišten za filtriranje prodaje proizvoda po godini

Nakon pripremanja **Select List** stavke koju ćemo koristiti za filtriranje, možemo napisati SQL Query za seriju **"Monthly Sales Chart"** regije. SQL Query koji će se koristiti na Slici 8.30. Također ćemo mapirati **"Label"** na **"SELL_MONTH"** i **"Value"** na **"COUNT"**.Također poredamo vrijednosti prema rastućem broju mjeseca i prikazujemo puni naziv mjeseca na **X** osi. Također ćemo postaviti oznaku **X** osi na **"Month"** i **Y** osi na **"Sales Count"**.



Slika 8.30: Postavljanje izvornog SQL upita i mapiranja stupaca za regiju grafikona mjesečne prodaje

Ako ste sve ispravno konfigurirali, vaš bi grafikon trebao sličiti onome prikazanom na Slici 8.31.



Slika 8.31: Monthly Sales grafikon

8.6 Pitanja

1. Koja je primarna svrha Oracle APEX regija grafikona?
2. Koje vrste grafikona možete izraditi pomoću Oracle APEX regija grafikona?
3. Kako možete osvježiti područje grafikona u Oracle APEX-u kada dođe do interakcije korisnika, poput filtriranja ili sortiranja?
4. Što radi INSTR funkcija, i zašto se koristi u Oracle APEX regijama grafikona?

8.7 Odgovori

1. Primarna svrha Oracle APEX regija grafikona je vizualno predstavljanje podataka, omogućujući korisnicima da steknu uvide i donesu odluke temeljene na podacima.
2. Oracle APEX regije grafikona podržavaju različite vrste grafikona, uključujući stupčaste grafikone, linijske grafikone, i još mnogo toga.
3. Možete konfigurirati Dynamic Action za osvježavanje regije grafikona u Oracle APEX-u kada korisničke interakcije promijene podatke, poput filtriranja ili sortiranja.
4. Funkcija INSTR koristi se za pronalaženje indeksa prvog pojavljivanja podniza u nizu. U regijama grafikona Oracle APEX koristi se za provjeru odgovaraju li određene vrijednosti odabranim kriterijima, poput ID-ova proizvoda na popisu.

9. Kako upravljati izbornicima?

VERONIKA ŠALGOVÁ, MICHAL KVET, MIROSLAV POTOČÁR

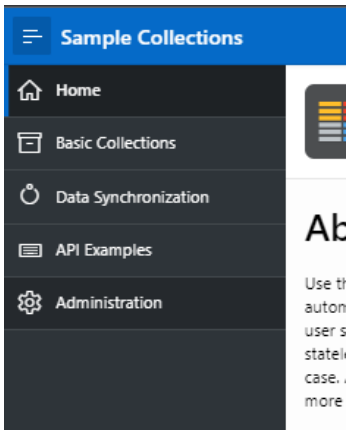
9.1 Kako upravljati izbornicima

Navigacijski izbornik automatski se stvara nakon izrade aplikacije. Sastoji se od unosa popisa koji povezuju na stranice aplikacije. Moguće je promijeniti kako i gdje se navigacijski izbornik prikazuje uređivanjem atributa navigacijskog izbornika na stranici korisničkog sučelja. Vrste navigacijskih izbornika dostupnih u Oracle APEX-u su:

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

9.2 Izbornik Side Menu

Ova vrsta izbornika može se proširiti ili sažeti klikom na ikonu izbornika u zaglavlju. Stavke navigacije prikazuju se pomoću komponente stabla. Podstavke se također mogu proširiti ili sažeti. Možemo vidjeti cijeli izbornik ili ga sažeti u usku traku ikona.



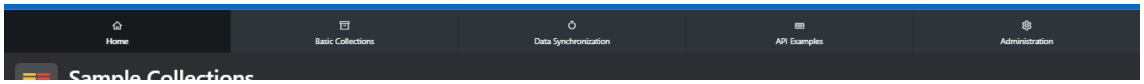
Slika 9.1: Prošireni Side Navigation Menu

Također je moguće konfigurirati navigacijski izbornik da bude potpuno skriven, što je zadana

opcija. Uska traka ikona može se prikazati kada je sažeta pomoću opcija predloška (Template Options).

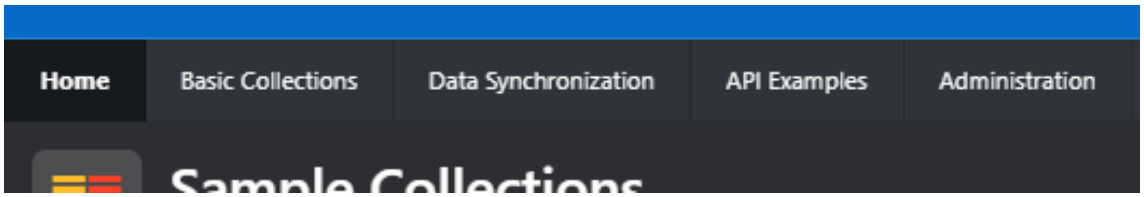
9.3 Izbornik Top Menu

Top Navigation Menu je prikazan na vrhu aplikacije. Oracle APEX nudi dva predloška za prikaz izbornika, to su *Top Navigation Tabs* i *Top Navigation Menu.* Prvi je prikladan za jednostavne aplikacije s nekoliko kartica. Prikaz je na Slici 9.2.



Slika 9.2: Top Navigation kartice

Top Navigation Menu pruža traku izbornika. Idealan je za složenije aplikacije s nekoliko slojeva hijerarhije unutar navigacije. Prikazan je na Slici 9.3.

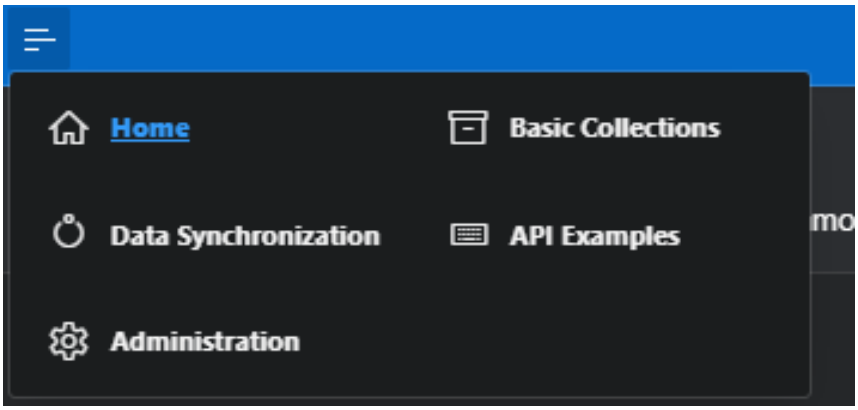


Slika 9.3: Top Navigation Menu

Top Navigation Tabs predložak se automatski postavlja na dno zaslona za male zaslone ili mobilne uređaje. Glavni navigacijski izbornik nalikuje većini aplikacija za stolna računala.

9.4 Izbornik Mega Menu

Ova vrsta izbornika prikazuje se kao skočna ploča koja se može otvoriti ili zatvoriti pomoću ikone iz zaglavlja. Mega izbornici su korisni za prikaz svih stavki navigacije odjednom.

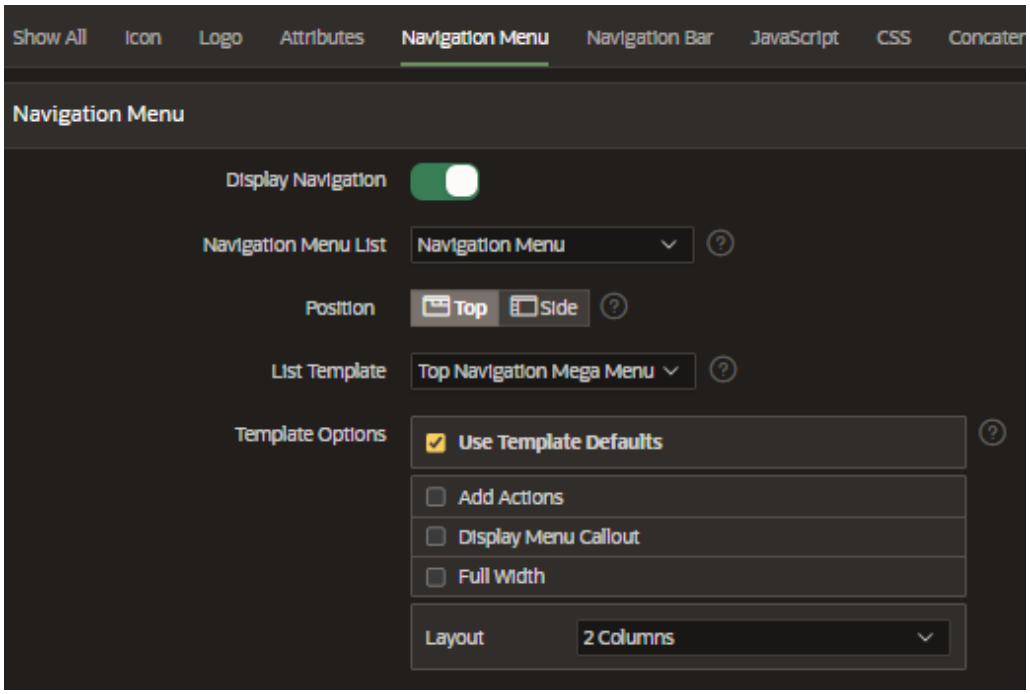


Slika 9.4: Top Navigation Mega Menu

Osim toga, moguće je odabrati broj stupaca u izborniku.

9.5 Uređivanje lista izbornika

Pojedinačne stavke izbornika čine popis koji se može stvarati i uređivati slično kao bilo koji drugi popis. Može im se pristupiti kroz *Shared Components* unutar Navigation regije. Odabire se *Navigation Menu* ili *List* opcija. Moguće je promijeniti položaj i način prikaza izbornika uređivanjem atributa izbornika Navigacija na stranici Korisničko sučelje. Tamo se dolazi odabirom *Shared Components* i *User Interface Attributes.* Na *Navigation menu* kartici, mogu se uređivati atributi, prikazani na Slici 9.5.



Slika 9.5: Navigation menu atributi

- Display Navigation – za uključivanje ili isključivanje navigacije.
- Navigation Menu List – za odabir popisa koji se koristi za navigacijski izbornik.
- Position – za odabir *Top* ili *Side* pozicije navigacijskog izbornika.
- List Template –za odabir različitih predložaka za prikaz navigacijskog izbornika.
- Template Options –za postavljanje opcija predloška koji se koriste za popis navigacijskog izbornika.

Odabirom *Apply changes* gumba pokreće se aplikacija i vidljive su promjene.

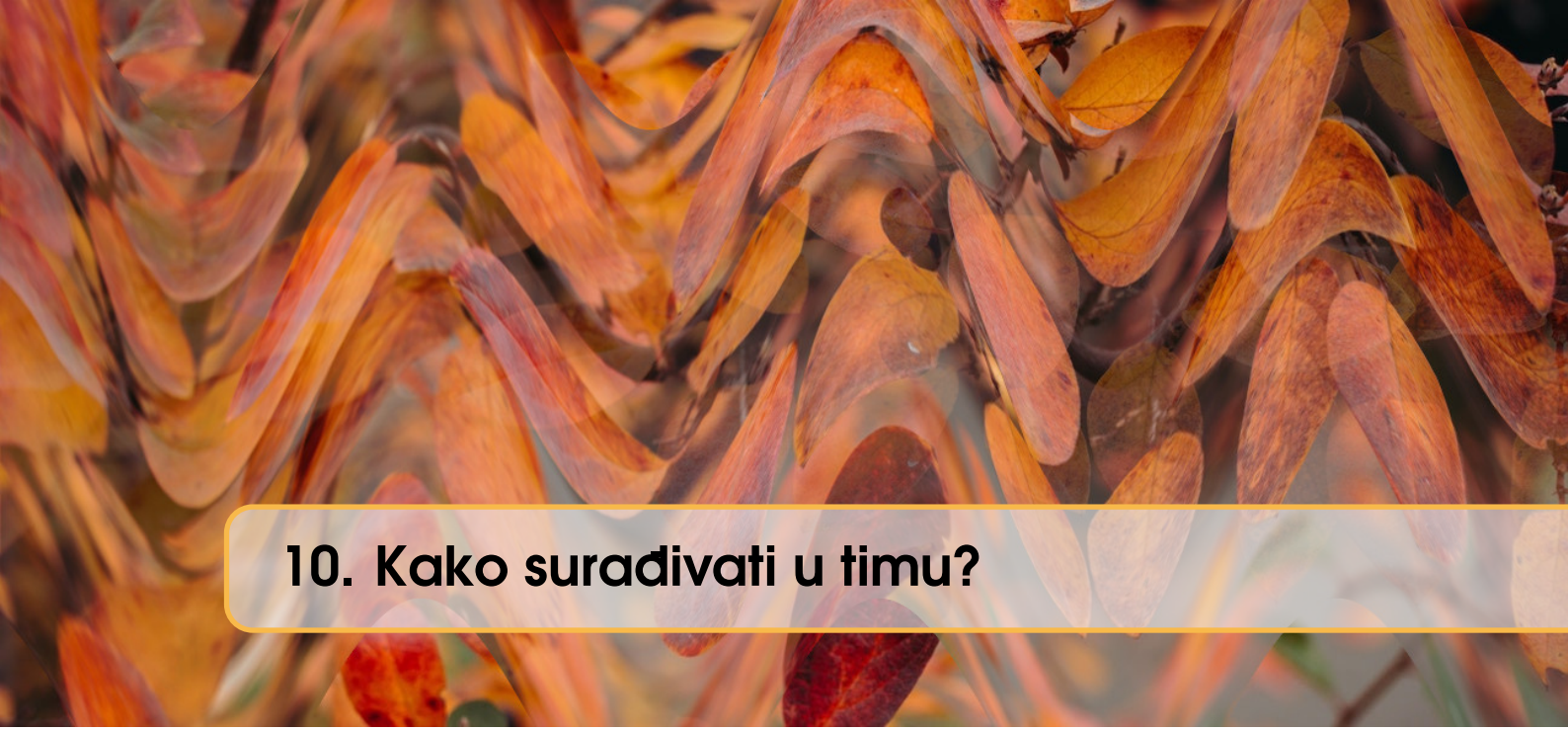
9.6 Pitanja

1. Obično se izbornik koji se koristi u Oracle APEX aplikacijama nalazi u lijevom dijelu ekrana. Na koju se vrstu odnosi?
2. Gdje se može pronaći definicija stavki izbornika koje treba uređivati?
3. Koja vrsta izbornika omogućuje prikaz svih stavki istovremeno bez potrebe za proširivanjem podizbornika?

9.7 Odgovori

1. Side Menu.
2. Pojedinačne stavke izbornika čine popis koji se može stvarati i uređivati slično kao bilo koji drugi popis. Može mu se pristupiti u Shared Components pod Navigation region.

3. Mega menu omogućuje prikaz svih stavki istovremeno samo jednim klikom bez potrebe za proširivanjem podizbornika.



10. Kako surađivati u timu?

P. STANISZEWSKI, M. SOŃTA, A. KIERZKOWSKI

10.1 Suradnička proizvodnja znanja temelj je razvoja niskog koda

"Suradnja nas kao ljude razlikuje čak i više od naših suprotstavljenih palčeva" budući da smo prirodno programirani za suradnju kao preferencijalni pristup (vidi npr. [1]; [3]), a umrežene platforme opremljene su značajkama za poticanje zajedničkog i kolektivnog dijeljenja znanja.

Prednosti pametne suradnje na digitalnim platformama su dvostruke:

- aspekt produktivnosti kada pojedinci koriste kolektivnu inteligenciju za razvoj boljeg proizvoda i
- motivacijski aspekt kada proces aktivira kreativne energije i pristup razvoju znanja vođen zajednicom.

10.2 Biti online zajedno

Oracle APEX je platforma stvorena za umrežavanje i interakciju tijekom rada na aplikacijama, a ovaj pristup ispunjava definiciju platforme suradničkog aktivizma. Ovakav vid suradnje zahtijeva "poseban način zajedničkog života"([3]). Jedan dio suradničkog iskustva je opcija pregledavanja, komentiranja i sukreiranja sadržaja za kolektivnu proizvodnju znanja, a jačanje kreativnosti i duha kroz oblikovanje društvenih odnosa je još jedna dimenzija suradničke upotrebe APEX-ove priče. Oracle APEX zajednica s više od deset regionalnih društvenih događaja godišnje, među mnogim drugima:

- APEX World, Nizozemska
- APEX World, APEX Alpe Adria, Austrija/Hrvatska/Slovenija
- APEX World, APEX Connect, Njemačka
- APEX World, ODTUG Kscope, SAD

je stekao stupanj razvoja za razvoj komercijalne platforme za koju je obećano da će zauvijek ostati besplatna za korisnike na način suradnje otvorenog koda i ambicije da slijedi smjernice energiziranja duha zajednice.

Nizom inicijativa i projekata, način na koji se zajednica razvija postaje još impresivniji, što donosi prilike "[...] načini zajedničkog boravka na mreži, osobito u vrlo imerzivnim okruženjima, mogu imati dodatni kognitivni i perceptivni sloj koji je još neistražen" ([3]) s ambicijom dostizanja zrelosti digitalnog aktivizma i pozivanjem različite publike koja nije informatički iskusna.

10.3 Tehničko znanje kroz suradnju

Pretraživanje akademskih članaka o APEX-u pokazuje da većina radova pokriva slučajeve poslovnog korištenja koji pokazuje primjere napora među različitim odjelima za razvoj aplikacije: ljudski potencijali, korporativna društvena odgovornost, komunikacija s unosom različitih korisnika ne samo u unosu podataka, već i s doprinos u konceptualizaciji dizajna same platforme.

Ovaj pristup, pristupačnost platformi i širok raspon edukativnog sadržaja o APEX-u čine ovaj alat i prilikom za ulaz u svijet niskog kodiranja i jačanja poslovne svijesti o praktičnim primjenama aplikacija. S ovakvim načinom razmišljanja, APEX otvara priliku i osobama koje nisu IT profesionalci da surađuju na IT projektima i omogućuje prostor za podizanje razine tehnološkog znanja. Te kompetencije i stavovi stvaraju bolje okruženje za zapošljavanje za sve dionike uključene u proces razvoja softvera.

Načini suradnje obično su povezani s metodologijom projekta. Čak i ako su platforme niskog koda poput APEX-a dizajnirane za postizanje ciljeva vrlo brzo, obično nam je potrebno više od jednog programera za realizaciju projekta u predviđenom vremenu i proračunu. U takvoj situaciji trebamo pravi alat koji će pomoći programerima da surađuju i rješavaju izazove i prilike tipične za suradnju u IT projektima. Ovisno o razmjeru i našim zahtjevima ili zahtjevima naše organizacije, to može biti povezano s aspektima kao što su:

- Standardizacija – kada želimo osigurati da su svi elementi aplikacije dosljedni. Odnosi se na standarde korisničkog sučelja, konvencije imenovanja, tehnike programiranja, stil kodiranja i sve druge standarde tipične za odabranu tehnologiju. Na primjer, u APEX svijetu možete jednostavno razmišljati o različitim načinima imenovanja stavki, procesa ili provjera valjanosti ili razmotriti standarde pri odabiru broja stranica.
- Obrazovanje – razvoj aplikacije s drugim programerima pomaže u povećanju vještina. Drugi članovi tima mogli bi biti napredniji u aspektima koji nama nisu bliski. Čak i kada koristimo platformu niskog koda kao što je APEX, koja teoretski svakom korisniku omogućuje da bude programer, možemo pronaći specijalizacije. Na primjer, netko se osjeća mnogo ugodnije uz bazu podataka, drugi uz čisti APEX, a treći uz značajke i jezike sučelja. Imati veći tim pomaže ostalima da razviju vještine koje nisu imali i povećaju kvalitetu aplikacije.
- Vrijeme – prirodno je da više programera može brže isporučiti aplikacije, što rezultira boljom izvedbom tima ili odjela odgovornog za određeni sustav ili platformu.
- Kvaliteta – veći tim može implementirati proces pregleda koda i provjeriti kvalitetu elemenata koje isporučuju članovi tima. Povećava kvalitetu konačnog proizvoda i ograničava broj pogrešaka.
- Povratna informacija – suradnja također znači komunikaciju između programera i korisnika. Korisno je imati informacije od korisnika o kvaliteti aplikacije, prijavljenim greškama ili zahtjevima za izmjene.

Oracle APEX, kao zrela platforma, pokriva različite strategije suradnje s ugrađenim značajkama, ali i s opcijama koje se mogu proširiti vanjskim alatima poput paketa za suradnju, aplikacija za upravljanje zadacima ili alata za provjeru koda.

Čak i ako smo tek započeli naše putovanje s APEX-om, bitno je znati kako možemo surađivati s drugim korisnicima s izvornim značajkama kako bismo takav rad učinili produktivnijim i bili sigurni kako riješiti tipične probleme kao što su dodjela zadataka, rad na točnim stranicama/ značajkama ili kontroliranje statusa planiranog rada.

Najvažniji dio na samom početku je komunikacija. Ako ju u timu više osoba, treba definirati načine planiranja rada za sljedeće dane, informiranja o statusu te kontrole vremena i rokova. Ništa nije važnije od redovitih sastanaka i komunikacije ako nešto nije jasno. Dobro je razmišljati o dnevnim sastancima, tjednim planiranjima ili drugim redovnim događajima. Također može se implementirati neku od poznatih metodologija, poput Scruma, kako bi takve navike bile zapisane na jednom mjestu. Ovakvim načinom rada identificirat će se najbolji način suradnje za svoj tim.

Na početku, kada razmišljate o alatima za suradnju, možete razmotriti osnovne proračunske

tablice za praćenje tko što radi. Naravno, na tržištu postoji mnogo namjenskih aplikacija za upravljanje zadacima kao što su Jira, Monday, ClickUp, Redmine itd., i naravno, s timom i razmjerom koji raste, možete razmisliti i o korištenju jedne od njih. Međutim, većina tih radnji može se pokriti značajkama uključenim u platformu Oracle Application Express.

10.4 Opis značajki

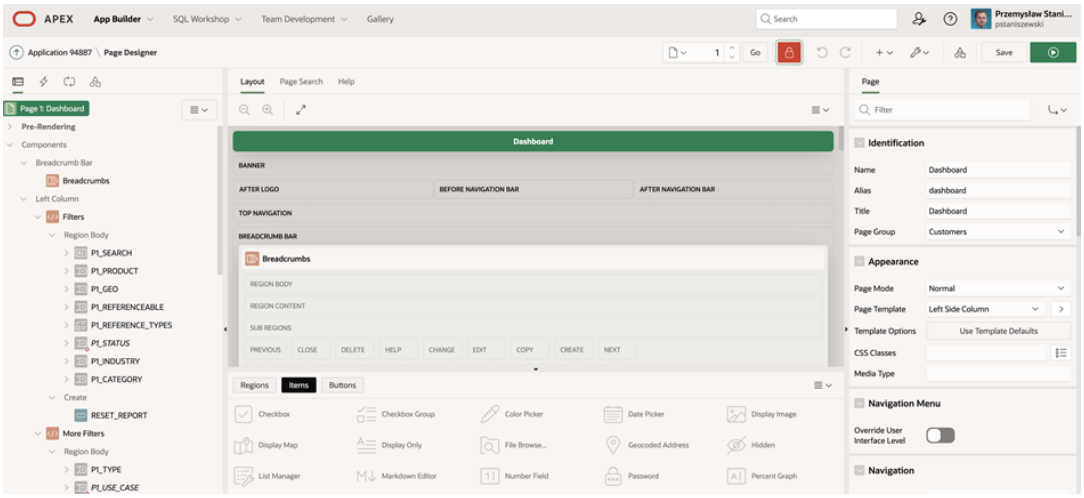
10.4.1 Zaključavanje stranice

Zaključavanje stranice je korisna značajka koja sprječava više programera da uređuju istu stranicu istovremeno. Kada programer uređuje stranicu, značajka blokiranja stranice sprječava druge programere da unesu izmjene na istoj stranici sve dok prvi programer ne završi svoj posao i ne pusti stranicu. To pomaže u sprječavanju sukoba i osigurava da se promjene izvode uredno i kontrolirano (vidi Sliku 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		

Slika 10.1: Page Blocking značajka u APEX-u

You can lock a specific page by clicking the lock icon on the page lists in the Application Builder or do it at the level of the page view.(see Figure 10.2).

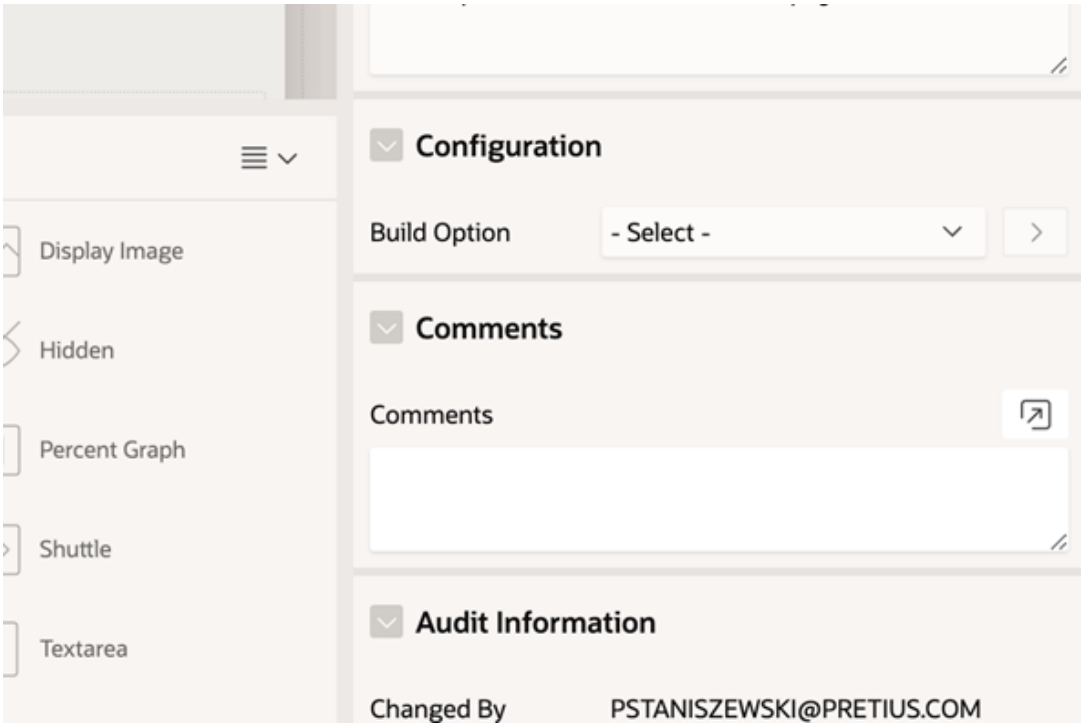


Slika 10.2: Zaključvanje specifične stranice

Dok zaključavate ili otključavate stranicu, možete dodati komentar kako biste vi ili drugi programeri mogli vidjeti više informacija o napretku rada.

10.4.2 Komentari

Oracle APEX omogućuje programerima dodavanje komentara na stranice, regije, stavke i mnoge druge elemente aplikacije. Komentari se mogu koristiti za pružanje konteksta drugim programerima, objašnjenje razloga iza odluke o dizajnu ili jednostavno dokumentiranje promjena napravljenih u aplikaciji. Korištenjem komentara programeri mogu bolje komunicirati jedni s drugima i osigurati da su rade u istom smjeru (vidi Sliku 10.3).

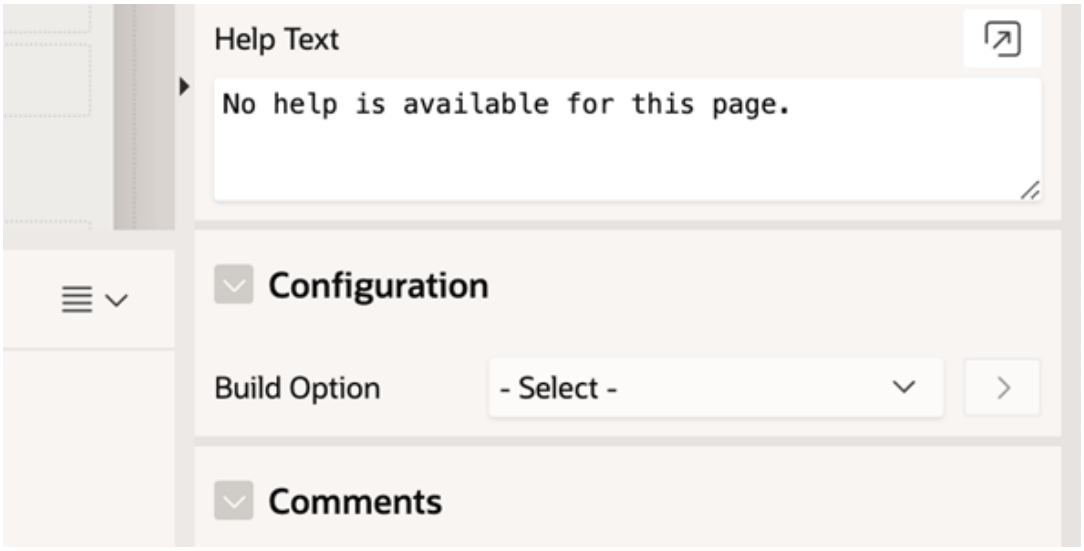


Slika 10.3: Dodavanje komentara

Važno je dokumentirati kod za druge članove tima ili buduće programere koji će se brinuti o razvoju ili održavanju aplikacije. Čak i ako korištenje platforme s niskim kodom može samostalno definirati kako je određena značajka napravljena, dodavanje nekih informacija još uvijek je dobra praksa ako je određeni element stvoren jedinstveno.

10.4.3 Mogućnosti izgradnje

Opcije izgradnje su još jedna od značajki koja može pomoći timovima da učinkovitije surađuju u Oracle APEX-u. Opcije izgradnje omogućuju programerima da specificiraju različite postavke izgradnje za različita okruženja, kao što su razvoj, testiranje i proizvodnja. To čini upravljanje promjenama u različitim okruženjima lakšim i osigurava primjenu ispravnih postavki kada se aplikacija implementira (vidi Sliku 10.4).



Slika 10.4: Opcije izgradnje

Možete ih tretirati kao informaciju treba li određena značajka (stranica, stavka, proces) biti uključena u sljedeće izdanje koje će se izvesti/uvesti u drugo okruženje. Dok izvozite/uvozite svoju aplikaciju, možete odabrati je li određena "izgradnja" uključena. Zamislite samo situaciju kada jedan programer radi na nekoj značajki i ne može to učiniti prije roka. Umjesto brisanja ove stranice iz aplikacije, možete samo izraditi namjensku "izvedbu" za nju i isključiti je tijekom izvoza aplikacije.

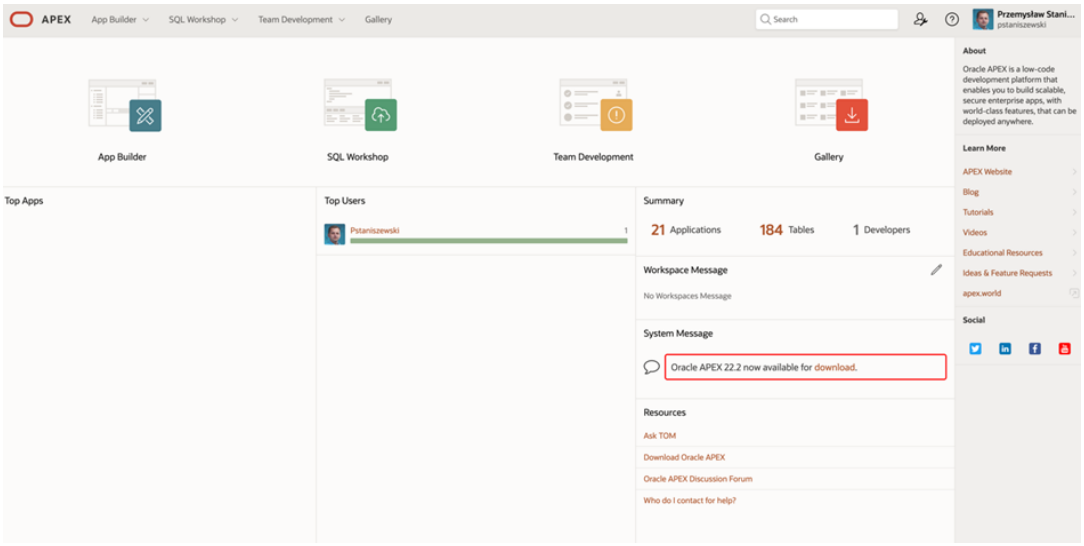
Upotreba ove značajke može biti prvi korak u definiranju CI/CD procesa aplikacije, koji je ključan u profesionalnom razvoju softvera.

10.4.4 Razvoj u timu

Oracle APEX Team Development značajka omogućuje timovima da rade zajedno na istoj aplikaciji u kontroliranom i suradničkom okruženju. Pomoću ove značajke članovi tima mogu posebno upravljati projektima:

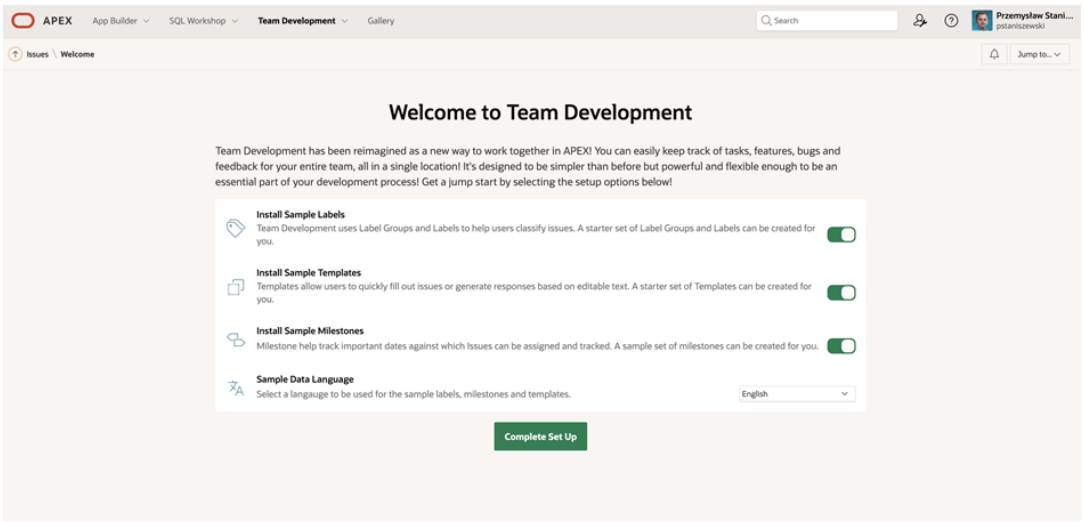
- Definirati kritiče točke i faze projekta.
- Stvaranje, dodjeljivanje i praćenje zadataka/problema.
- Primijenite klasifikacije pomoću grupa oznaka i oznaka.
- Napravite predloške problema kako bi stvaranje zadataka ili izvješćivanje bilo dosljedno.
- Povežite probleme s aplikacijom i stranicom;
- Komentirajte i raspravljajte o postojećim problemima. Promijenite njihov status i pratite napredak faze projekta.
- Pretplatite se za različite vrste problema.
- Pratite povratne informacije korisnika aplikacije i odlučite što učiniti s njima.

Team Development je jedna od četiri glavna odjeljka dostupna u APEX IDE. Njegova dostupnost ovisi o vašoj verziji APEX-a. Predstavljen je u APEX 19.2, ali u najnovijim verzijama nije dostupan odmah, već zahtijeva dodatne radnje da bi postao dostupan (vidi Sliku 10.5).



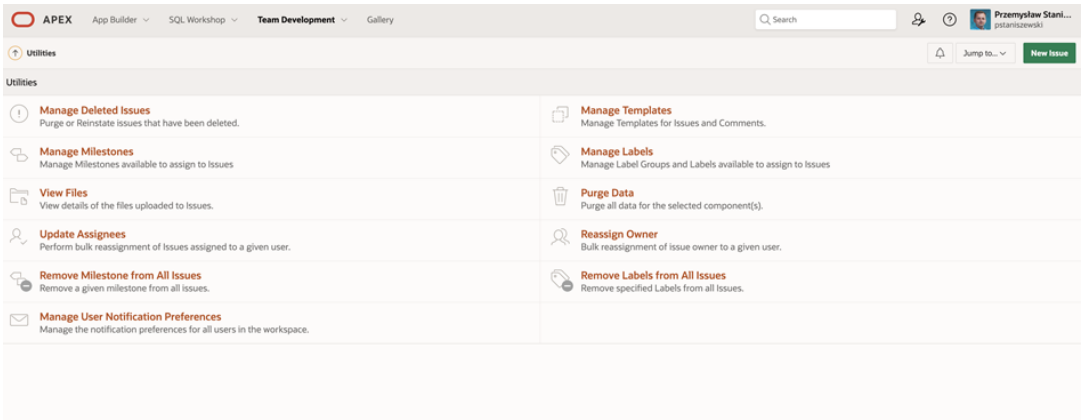
Slika 10.5: Team Development značajka

Da biste ga uključili, potrebno je u glavnom izborniku odabrati opciju "Team Development". Zatim ćete biti premješteni na namjensku instalacijsku stranicu. Ako želite, možete uključiti primjere problema, prekretnica, predložaka i oznaka, ali također možete preskočiti ovaj dio i aktivirati značajku bez njega (vidi Sliku 10.6).



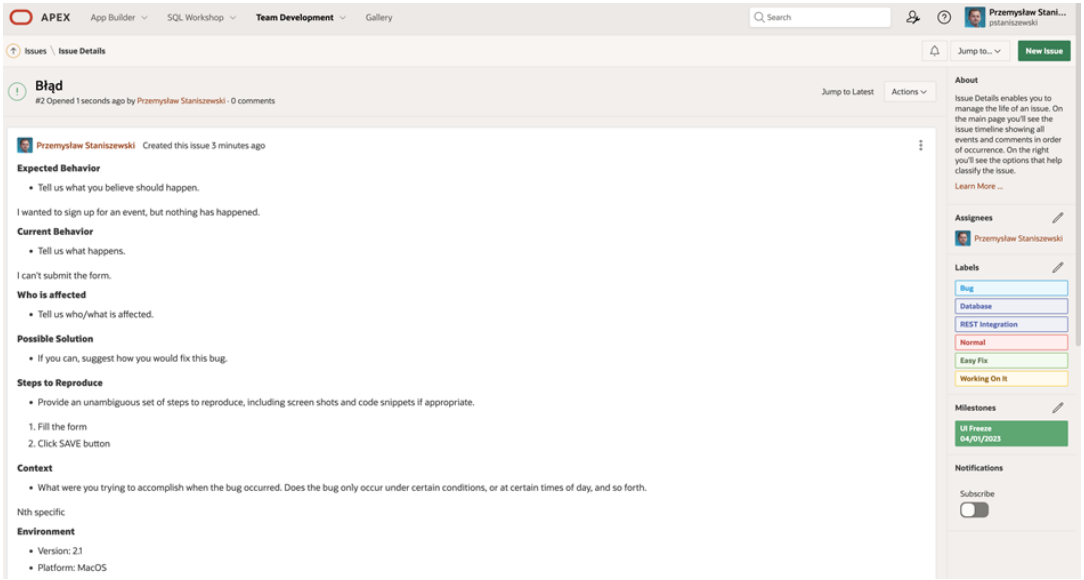
Slika 10.6: Aktivacija Team Development značajke

Ako vaš tim želi početi koristiti "Team Development", prvo biste trebali konfigurirati njegov rječnik za svoj stil rada. Zatim idite na odjeljak Utilities i definirajte vrste svojih problema, faze projekta i prekretnice ili predloške za tipičnu komunikaciju na projektu (vidi Sliku 10.7).



Slika 10.7: Prilagodba Team Development značajke

Najvažniji dio ove značajke je odjeljak Issue, gdje možete pronaći sve prijavljene probleme, kategorizirati ih i dodijeliti određenim osobama. Svaki problem može se postaviti na više od jednog programera, koji će biti dostupni za dodavanje komentara, raspravu o njemu unutar njegovog konteksta i promjenu statusa. Ako je potrebno, korisnici mogu učitati datoteke i u osnovi pokrenuti svu komunikaciju vezanu uz probleme unutar "Team Developmnet". Također je važno da se svi problemi mogu povezati s određenom aplikacijom i stranicom, tako da je jednostavno započeti istragu i pratiti koji su problemi vezani uz koji dio aplikacije. Status problema jednostavna je verzija tijeka problema koji se može pronaći u mnogim sustavima za upravljanje zadacima – korisnici ih mogu prijaviti, označiti kao "u tijeku", vratiti se izvjestitelju i zatvoriti s informacijama o uspjehu ili različitim razlozima zatvaranja (vidi Sliku 10.8).



Slika 10.8: Komunikacija vezana uz problem unutar Team Development značajke

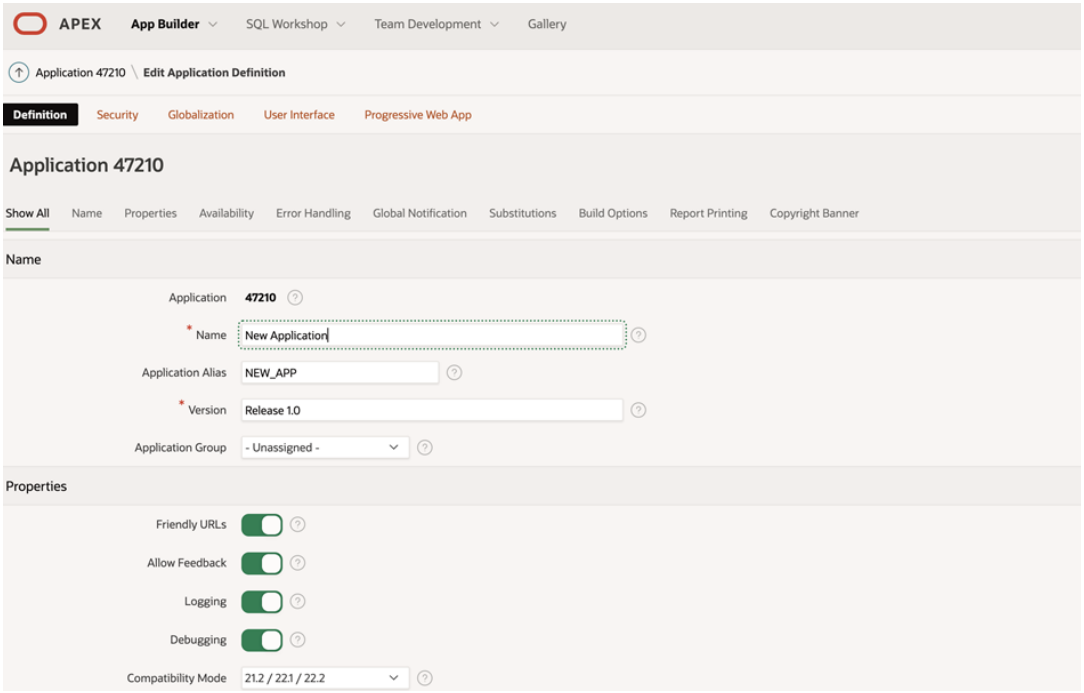
Programeri se također mogu pretplatiti na probleme koji su im dodijeljeni, što pruža informacije e-poštom kada postoji bilo kakva radnja na razini problema. Aplikacija omogućuje korisnicima da ograniče broj e-mailova tako da ih šalju u obliku satnih ili dnevnih sažetaka.

Preporučamo korištenje "Team Development" ako vaš tim nema implementiran drugi sustav upravljanja zadacima ili koristite vrlo osnovne alate poput proračunskih tablica za upravljanje projektom.

10.4.5 Povratne informacije

Feedback Pages su ugrađena značajke u Oracle APEX koja omogućuju krajnjim korisnicima da daju povratne informacije o funkcionalnosti i izvedbi aplikacije. Stranice s povratnim informacijama mogu se prilagoditi potrebama aplikacije i mogu uključivati različite vrste povratnih informacija, kao što su izvješća o greškama, zahtjevi za značajkama i općenite povratne informacije.

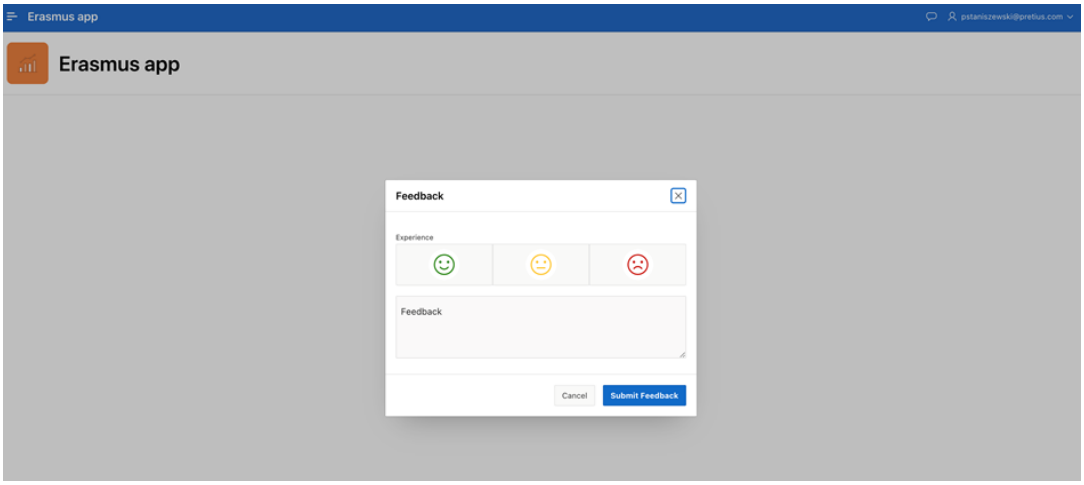
Ključno je dobiti informacije od krajnjih korisnika i tretirati ih kao dio razvojnog tima. Što bliže surađujemo ne samo interno, već i sa stvarnim korisnicima aplikacije, to ćemo kvalitetniju aplikaciju imati. Također omogućuje timu da prilagodi aplikaciju stvarnim potrebama i pomaže u izgradnji povjerenja između razvojnog tima i korisnika (vidi Sliku 10.9).



Slika 10.9: Dopuštanje povratnih informacija u definiciji aplikacije

Da biste ga uključili, trebate odabrati opciju tijekom izrade aplikacije. Ako aplikacija već postoji, idite na "Edit Application Definition" i označite opciju "Allow Feedback" u sekciji Properties.

Da biste ga uključili, trebate odabrati opciju tijekom izrade aplikacije. Ako aplikacija već postoji (vidi Sliku 10.10).



Slika 10.10: Predaja povratne informacije u aplikaciji

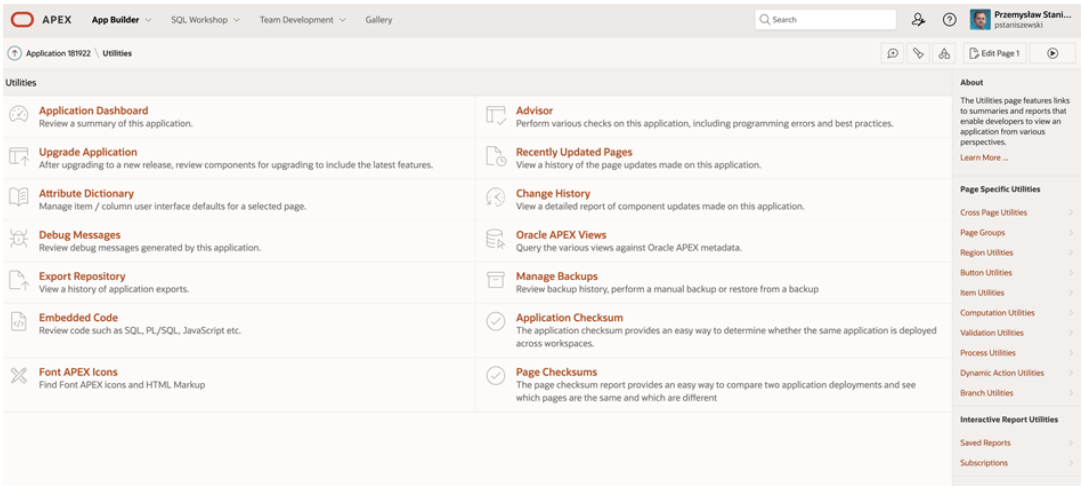
Izgled Feedback stranice može se podesiti jer se radi o standardnoj APEX stranici. Sve informacije koje daje krajnji korisnik dostupne su u sekciji Team Development i tim ih može analizirati (također ih je moguće pratiti na razini pojedine aplikacije u sekciji Administration). Osim toga, razvojni programeri dobivaju detaljne informacije u vezi s prijavljenim povratnim informacijama (ID aplikacije, broj stranice, podaci o korisniku, informacije o njegovom okruženju itd.) i mogu to pretvoriti u problem.

10.4.6 Standardizacija

Dok razvijate aplikacije u timu, također je vrlo važno održavati dosljednost između razvojnih standarda različitih članova tima. Kao i u svakom aspektu suradnje, komunikacija je ključ uspjeha – na početku projekta tim se treba dogovoriti o načinu razvoja i korištenja APEX komponenti. Najvažniji aspekti toga uključuju:

- Konvencija imenovanja (za stavke, procese, provjere valjanosti, dinamičke radnje itd.);
- Numeriranje stranica i imenovanje (alias);
- Dopušteni elementi (npr. za rječnike koristimo samo interaktivne mreže, svaki obrazac mora biti prikazan kao modalna stranica);
- UX/UI standardi (npr. izgled i dojam tipičnih stranica, politika usklađivanja za različito vrijeme podataka, dosljedni formati podataka, preporučene ikone za određene radnje itd.);
- Dosljednost podataka (npr. zajednički rječnici definirani na jednom mjestu i zajedničke komponente).

APEX daje nekoliko rješenja za pomoć timovima u upravljanju standardizacijom. Većina ih se može pronaći u sekciji "Utilities" na razini odabrane aplikacije (vidi Sliku 10.11).



Slika 10.11: Utilities izbornik

Najvažnije na početku razvoja je:

- Cross-page izvješća (Page Specific Utilities) – omogućuju programerima da usporede različite APEX elemente (atribute stranice, izvješća, gumbe, stavke itd.) u aplikaciji kako bi provjerili dosljednost i ažurirali ih odjednom bez ručnog mijenjanja atributa na više mjesta.
- Povijest promjena/Nedavno ažurirane stranice – omogućuje praćenje radnji koje su napravili programeri.
- Definicija zadanih vrijednosti (rječnik atributa) – omogućuje stvaranje zajedničkog rječnika za održavanje dosljednog naziva, oblikovanja, vrste podataka itd. za elemente koji se ponavljaju na mnogim mjestima u aplikaciji. Na primjer, "Datum ažuriranja" može postojati na različitim izvješćima ili stavkama u aplikaciji, ali uvijek želimo imati istu oznaku i format datuma za svako pojavljivanje.

Naravno, APEX daje mnoge dodatne značajke i strategije za održavanje dosljednosti na razini baze podataka i aplikacije. Za one koji žele proširiti svoje znanje, pročitajte APEX dokumentaciju i provjerite naprednije opcije kao što su Global Page, Blueprints, User Interface Defaults, Master Theme Application and Subscribe mechanism, APEX Views, ili Plug-ins.

10.5 Zaključak

Kao zrela razvojna platforma, Oracle APEX pruža mnoge značajke koje pomažu u suradnji. Kao programeri s niskim kodom, možemo ih koristiti i profesionalno stvarati i održavati aplikacije temeljene na APEX-u bez proširenja platforme ili njezine integracije s vanjskim alatima. Ako ste novi u APEX-u i novi ste u strategijama suradnje u programiranju, započnite s najosnovnijim koracima opisanim u ovom poglavlju:

- Komunikacija i redoviti sastanci s Timom (uživo ili na daljinu – koristite alate poput Google Meet, Zoom, ili MS Teamsza video/glas, i Slack, Google Chat, ili MS Teamsza brzu razmjenu poruka);
- Počnite koristiti alate za suradnju, npr. APEX Team Development, planirati razvoj i provjeriti napredak;
- Počnite dokumentirati svoj kod i standarde. I provjerite ih!

Suradnja u razvoju s niskim kodom ponekad može zvučati pretjerano, ali vrijedi je pokušati i donosi jednaku zaradu kao i standardno programiranje. Štoviše, većina tipičnih strategija suradnje poput "pregleda koda", "programiranja u parovima" ili "mob programiranja" može se koristiti i s platformom s niskim kodom, čak i ako ponekad pregledani element nije čisti kod, već proces, stranica ili konfiguracija.

Ovisno o veličini projekta ili standardima implementiranim u organizaciji, u nekom ćete

trenutku možda trebati proširiti svoj proces suradnje. Na primjer, u velikim projektima morat ćete početi koristiti repozitorije koda, alate za automatiziranu izgradnju i implementaciju ili softver posvećen testiranju i provjeri kvalitete vašeg koda. Korištenje elemenata profesionalnih CI/CD procesa kao što su Git, Docker, Flyway, Liquibase, Jenkins, Gradle, SonarQube, Selenium, Cypress i mnogi drugi također je moguće uz aplikaciju temeljenu na APEX-u. Međutim, u mnogim situacijama, elementi uključeni u platformu dovoljni su za stvaranje aplikacije koja daje svrhu s timom kolega programera.

10.6 Pitanja

1. Što je značajka zaključavanja stranice i zašto je važna?
2. Kako programeri mogu komunicirati tijekom razvoja s APEX-om?
3. Što članovi tima mogu iskoristiti korištenjem Team Development značajke?

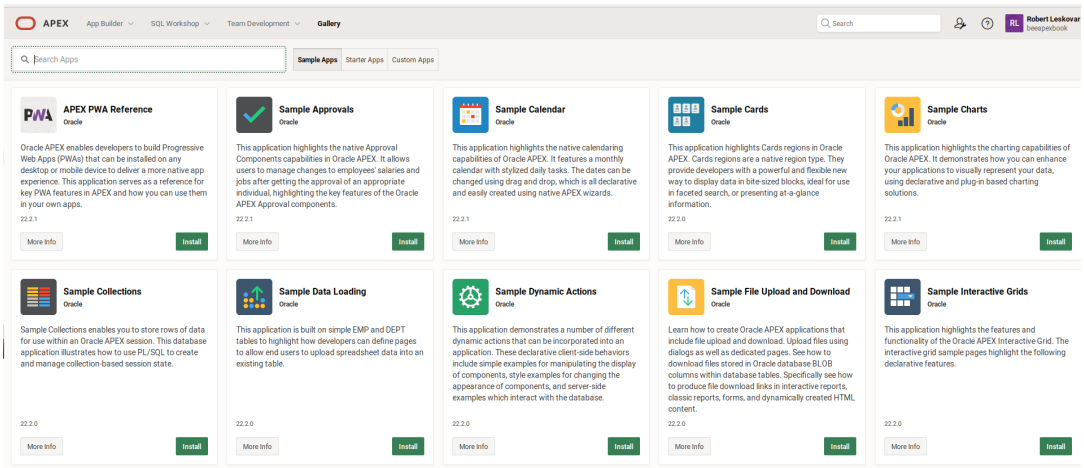
10.7 Odgovori

1. Zaključavanje stranice je korisna značajka koja sprječava više programera da uređuju istu stranicu istovremeno i time sprječava sukobe.
2. Programeri mogu dodavati komentare na stranice, regije, stavke i mnoge druge elemente aplikacije. Komentari se mogu koristiti za pružanje konteksta drugim programerima, objašnjenje razloga koji stoje iza odluke o dizajnu ili jednostavno dokumentiranje promjena unesenih u aplikaciju. Korištenjem komentara programeri mogu bolje komunicirati jedni s drugima i osigurati ukupnu kvalitetu softvera.
3. Team Development značajka omogućuje članovima tima da upravljaju razvojnim projektima, posebno: definiraju prekretnice i faze projekta, stvaraju, dodjeljuju i prate zadatke/probleme, primjenjuju klasifikacije pomoću grupa oznaka i oznaka, stvaraju predloške problema, kako bi stvaranje zadataka ili izvješćivanje bilo dosljedno, povežite probleme s aplikacijom i stranicom, komentirajte i raspravljajte o postojećim problemima, promijenite njihov status i pratite napredak faze projekta i pretplatite se za različite vrste problema

11. Kako iskoristiti galeriju oglednih/početnih aplikacija i dodataka?

VJERAN STRAHONJA, DIJANA OREŠKI

Ovo poglavlje istražuje početne aplikacije, primjere aplikacije i dodatke u Oracle APEX-u. Svi Oracle APEX radni prostori dolaze s oglednim aplikacijama (engl. Sample Apps) i početnim aplikacijama (engl. Starter Apps) (vidi Sliku 11.1) te dodacima (engl. plug-ins).



Slika 11.1: Oracle APEX radni prostori dolaze s početnim i oglednim aplikacijama

Početne aplikacije (engl. Starter apps) su samostalna rješenja koja zadovoljavaju jednostavne zahtjeve i ne zahtijevaju složene pristupe. S druge strane, primjeri aplikacija usmjereni su na specifične funkcije i pružaju vodič za korištenje određene značajke. Dodaci omogućuju korisnicima proširenje aplikacija prilagođenim funkcijama.

11.1 Kako instalirati ogledne i početne aplikacije?

Korisnici mogu preuzeti ogledne i početne aplikacije iz galerije. Ako je to omogućio administrator, aplikacije se mogu instalirati pomoću Gallery.

Za **instalaciju** aplikacija iz Gallery:

1. Odi na Workspace početnu stranicu.

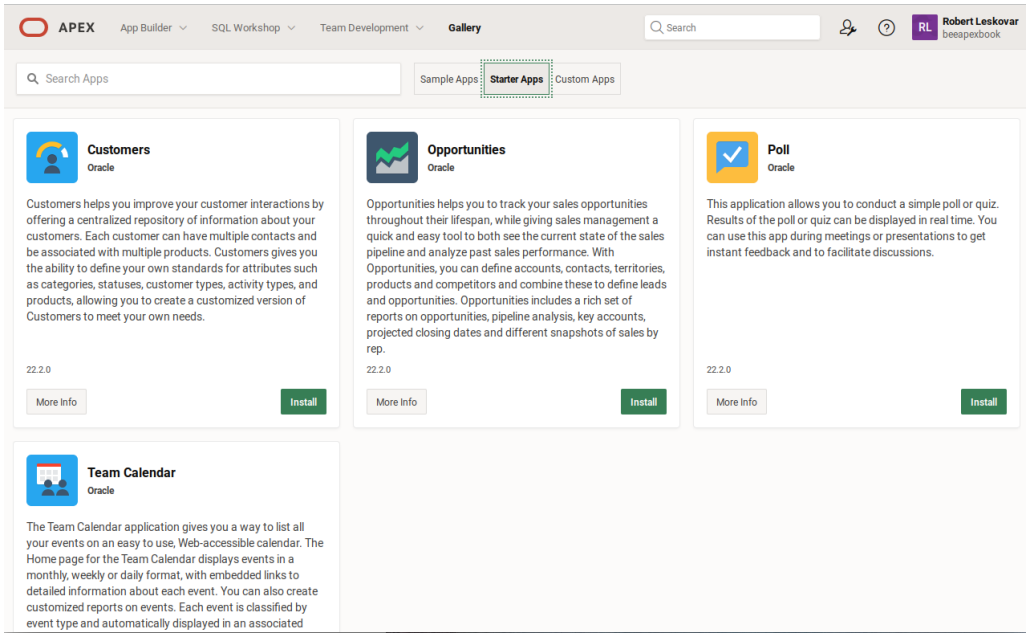
2. Odaberi **Gallery**. Ogledne aplikacije, početne aplikacije i prilagođene aplikacije pomoću kartica prikazuju se kada se **Gallery** stranica učita.
3. Odaberi **Sample Apps** ili **Starter Apps**. Kartice se koriste za prikaz popisa aplikacija. Osim ako program prethodno nije instaliran, svaka kartica sadrži gumb Install. Run i Remove App ikone su prisutne ako je aplikacija već instalirana.
4. Odaberi **Install App** za instalaciju aplikacija
5. Proširie Advanced Settings sekciju. (Nije obvezno)
   a. Definiraj da li će Application ID biti dodan ručno ili automatski (zadano).
   b. Promijeni Parsing Schema.
6. Odaberi Install Application za završetak instalacije.

Odabirom ikone Remove App iz Gallery, korisnici mogu deinstalirati aplikacije koje su odatle instalirane. Aplikacije koje su uvezene s GitHuba i preuzete nemaju ikona za deinstalaciju, pa se moraju izbrisati da bi se deinstalirale iz radnog prostora. Koraci za**deinstaliranje** aplikacija iz Gallery:

1. Odi na Workspace početnu stranicu.
2. Odaberi ikonu Gallery. Sample Apps, Starter Apps, i Custom Apps kartice su prikazane kad se stranica Gallery učita.
3. Odaberi Starter Apps ili Sample Apps. Direktno skinuti programi iz Gallery će imati vidljivu Remove ikonu i i Run ikonu.
4. Za uklanjanje aplikacije, odaberi Remove.

11.2 Početne aplikacije

Početne aplikacije korisne su aplikacije koje daju samostalna rješenja, stvorene da zadovolje jednostavne zahtjeve koji ne zahtijevaju značajan i nepotrebno složen odgovor. Ove se aplikacije mogu koristiti "iz kutije" ili se mogu poboljšati vlastitim jedinstvenim značajkama (vidi Sliku 11.2).



Slika 11.2: Početne apliacije u galeriji

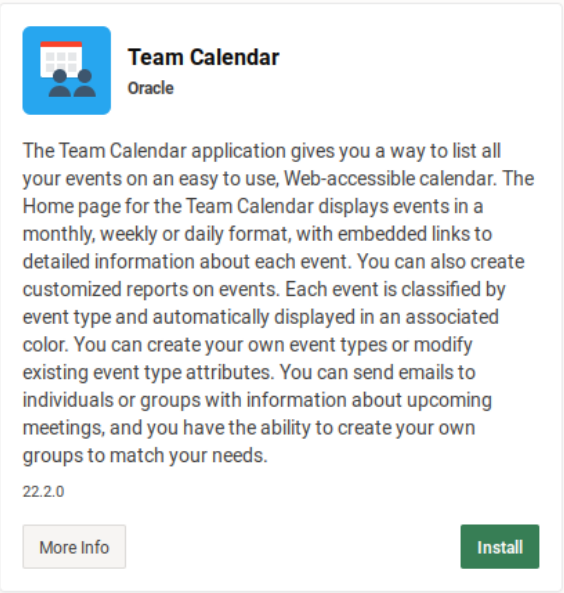
Primjeri početnih aplikacija su:

- **Customers** – Pružajući konsolidiranu bazu podataka o klijentima, aplikacija Customers korisnicima omogućuje poboljšanje interakcije s klijentima. Svaki kupac ima mogućnost povezivanja s nekoliko kontakata i stavki. Kupci vam omogućuju postavljanje vlastitih

kriterija za značajke kao što su kategorije, status, vrste kupaca, vrste aktivnosti i roba tako da možete dizajnirati posebnu verziju kupaca koja će odgovarati vašim zahtjevima.

- **Opportunities** – Ova aplikacija pruža upravljanje prodajom s brzim i jednostavnim rješenjem za oboje, gledajući trenutno stanje prodajnog kanala i analizirajući povijesni učinak prodaje. Pomaže vam pratiti svoje prodajne prilike tijekom njihova životnog ciklusa. Možete definirati potencijalne klijente i prilike kombinirajući račune, kontakte, regije, robu i konkurente. Opportunities nudi sveobuhvatnu zbirku informacija o prilikama, istraživanju planiranja, značajnim računima, predviđenim datumima zatvaranja i drugim snimkama prodajnih predstavnika.
- **Poll** – Korisnici mogu koristiti ovu aplikaciju za izradu brze ankete ili testa. Rezultati ankete ili kviza mogu se odmah vidjeti. Ovaj se alat može koristiti za olakšavanje rasprava i dobivanje trenutnih povratnih informacija tijekom sastanaka ili prezentacija.
- **Team Calendar** - Uz pomoć aplikacije Team Calendar, korisnici mogu jednostavno ispisati sve događaje na kalendaru temeljenom na webu koji je jednostavan za korištenje. Početna stranica Team Calendar navodi događaje u mjesečnom, tjednom ili dnevnom stilu zajedno s integriranim poveznicama na više specifičnih detalja za svaki događaj. Osim toga, mogu se dizajnirati jedinstvena izvješća o događajima. Svaki događaj je kategoriziran prema svojoj vrsti, a odgovarajuća boja se automatski prikazuje. Korisnici mogu razviti nove vrste događaja ili promijeniti atribute već postojećih vrsta događaja. Korisnici imaju opciju izgraditi vlastite grupe kako bi bolje zadovoljile svoje potrebe i mogu slati e-poruke pojedincima ili grupama u kojima ih obavještavaju o nadolazećim sastancima.

Evo primjera instalacije aplikacije za Team Calendar (vidi Sliku 11.3).

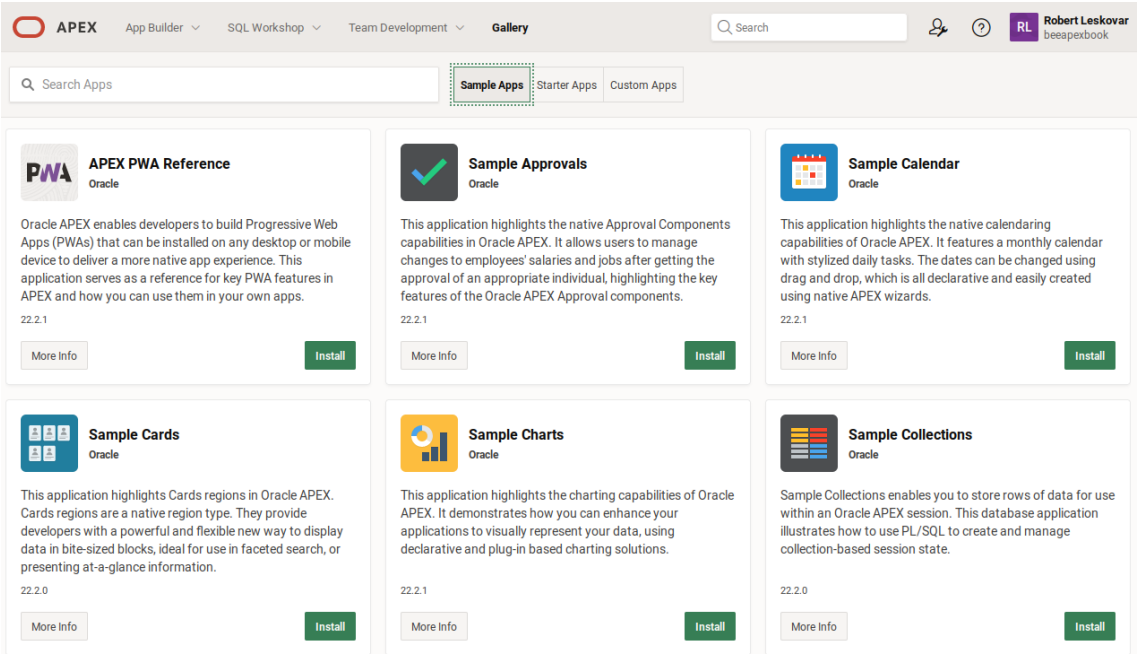


Slika 11.3: Instalacija početne aplikacije Team Calender

Odaberite Install.

11.3 Ogledne aplikacije

Ogledne aplikacije su stvorene kako bi se istaknule određene mogućnosti i služile kao razvojni priručnik za korištenje određene značajke (vidi Sliku 11.4).



Slika 11.4: Ogledne aplikacije

Između ostalog, možemo odabrati:

- **APEX PWA Reference** - Kako bi pružili izvornije iskustvo aplikacije, programeri mogu izraditi progresivne web aplikacije (PWA) pomoću Oracle APEX-a, koje se mogu implementirati na bilo kojem stolnom ili mobilnom uređaju. Glavne PWA mogućnosti u APEX-u i kako ih primijeniti u vlastitim aplikacijama uključene su u ovu aplikaciju.
- **Sample Approvals** - Ova aplikacija prikazuje mogućnosti izvornih komponenti odobrenja Oracle APEX-a. Istaknute su komponente Oracle APEX Consent, omogućujući korisnicima da upravljaju promjenama plaća i poslova zaposlenika nakon što dobiju odobrenje prave osobe.
- **Sample Calendar** - Ova aplikacija prikazuje ugrađene značajke kalendara Oracle APEX. Uključen je mjesečni kalendar sa stiliziranim dnevnim zadacima. Opcijom povuci i ispusti (engl. drag and drop), koja se brzo se generira pomoću izvornih APEX čarobnjaka, može se ažurirati datum.
- **Sample Cards** –U Oracle APEX-u ova aplikacija ističe regije kartica. Regije s karticama su oblik matične regije. Programerima daju snažnu i prilagodljivu metodu za predstavljanje podataka u malim bitovima, što je savršeno za korištenje u fasetiranom pretraživanju ili brzom predstavljanju informacija.
- **Sample Charts** – Ova aplikacija prikazuje značajke grafikona Oracle APEX-a. Objašnjava kako možete koristiti deklarativne alate i alate za grafikone temeljene na dodacima da biste poboljšali svoje aplikacije tako da vizualno prikazuju vaše podatke.
- **Sample Collections** – Ova aplikacija korisnicima omogućuje pohranu podataka u zbirke uzoraka za korištenje u sesiji Oracle APEX. Ova aplikacija baze podataka demonstrira stvaranje i upravljanje stanjem sesije na temelju zbirke pomoću PL/SQL.
- **Sample Data Loading** - Ova je aplikacija izrađena pomoću jednostavnih EMP i DEPT tablica kako bi se pokazalo kako programeri mogu dizajnirati stranice koje korisnicima omogućuju dodavanje podataka proračunske tablice u tablicu koja već postoji.
- **Sample Dynamic Actions** Ova aplikacija prikazuje niz dinamičkih radnji koje se mogu dodati aplikacijama. Deklarativna ponašanja na strani klijenta uključuju jednostavne primjere za modificiranje načina na koji se komponente prikazuju, primjere stilova za prilagodbu

izgleda komponenti i primjere na strani poslužitelja za interakciju s bazama podataka.

- **Sample File Upload and Download** – Naučite kako izraditi Oracle APEX aplikacije s funkcijom učitavanja i preuzimanja datoteka. Za učitavanje datoteka koristite namjenske stranice i dijaloške okvire. Provjerite postupak za preuzimanje datoteka iz BLOB stupaca Oracle baze podataka u tablicama baze podataka. Pogledajte posebno kako napraviti veze za preuzimanje datoteka u obrascima, interaktivnim izvješćima, tradicionalnim izvješćima i dinamičkom HTML sadržaju.
- **Sample Interactive Grids** – Oracle APEX Interactive Grid značajke i funkcionalnosti su naglašene u ovoj aplikaciji. Značajke su naglašene na stranici interaktivne mreže.
- **Sample Maps** – Brojni primjeri pregleda koordinatnih podataka na karti navedeni su u ovoj aplikaciji. Koristite značajku toplinske karte, linije, poligone ili oznake karte. Mogućnosti Oracle Spatial (dostupne u svakoj bazi podataka Oracle) mogu se lako integrirati s regijom karte APEX za izvođenje "Within Distance Search", "Nearest Neighbor Search" ili druge prostorne analize.
- **Sample Master Detail** – Ova aplikacija prikazuje izvorne mogućnosti glavnog detalja Oracle APEX-a. U aplikaciji postoje četiri moguća izgleda stranice s glavnim detaljima. Prva dva izgleda koriste interaktivne mreže koje je moguće uređivati za prikaz glavnih detalja na jednoj stranici. Posljednja dva izgleda predstavljaju glavni detalj na dvije stranice, koristeći kombinaciju interaktivnih mreža koje je moguće uređivati, unosa obrasca, starih izvješća i modalnog skočnog prozora.
- **Sample Reporting** – Ova aplikacija prikazuje značajke izvješćivanja Oracle APEX-a. Koristeći SQL, korisnici mogu kreirati Interactive Reports, Interactive Grids, Faceted Search Reports, Cards Reports, and Classic Reports.
- **Sample REST Services** – Ova aplikacija pokazuje kako koristiti Oracle APEX za povezivanje s vanjskim REST uslugama. Oracle.example.hr, prototip RESTful usluge, je korištena. Primjeri u ovoj aplikaciji pokazuju kako filtrirati, kako dodati stranice i kako proizvesti jednostavno tablično izvješće o podacima usluge REST.
- **Sample Trees** – Saznajte kako koristiti SQL upit za razvoj kontrole stabla. Aplikacija Oracle APEX u ovom primjeru pokazuje nekoliko načina za uključivanje kontrole stabla.
- **Universal Theme Reference** - Dajući jednostavan način za navigaciju kroz brojne predloške, odabire predložaka i stilove tema, ova aplikacija uvodi korisnike u univerzalnu temu. Primjeri pokazuju kako korisnici mogu brzo upravljati izgledom vaše stranice kako bi proizveli odličnu aplikaciju

11.4 Dodaci

Dodaci omogućuju proširenje okvira Application Express s prilagođenim tipovima stavki, tipovima regija, procesima i dinamičkim radnjama. Jednom definirane komponente koje se temelje na dodacima stvaraju se i održavaju vrlo slično standardnim Application Express komponentama. Dodaci omogućuju programerima stvaranje visoko prilagođenih komponenti kako bi poboljšali funkcionalnost, izgled i jednostavnost svojih aplikacija. Dodaci omogućuju korisnicima dodavanje dodatnih funkcija APEX aplikacijama pomoću dodataka koji nisu ugrađeni u platformu. Zajednica APEX razvila je veliki raspon različitih dodataka korisnih za proširenje postojećih značajki ili dodavanje novih značajki. Postoje stotine dodataka za Oracle Apex kategoriziranih u pet vrsta:

- Dynamic action.
- Region.
- Item.
- Process.
- Authentication.

Oracle Application Express nudi različite dodatke unutar svake kategorije. Dodaci daju mogućnost dodavanja novih značajki aplikaciji, čime se poboljšavaju ugrađene vrste. Budući da su dodaci

namijenjeni za ponovnu upotrebu, programeri ih mogu dijeliti sa zajednicom Oracle Application Express pomoću repozitorija dodataka, kao i izvoziti ih i uvoziti u različite radne prostore. Koraci uključeni u implementaciju dodatka su sljedeći:

- Dodajte dodatak radnom prostoru svoje aplikacije ili ga izradite ispočetka.
- Da biste upotrijebili dodatak, uredite ili izradite vrstu dinamičke akcije, stavku, regiju, proces ili shemu autorizacije.
- Pokrenite svoju aplikaciju kako biste provjerili funkcionalnost dodatka.

Središnje mjesto za programere za dijeljenje i preuzimanje dodataka je Plug-in Repository.Repozitorij se može naći na Oracle Technology Network.

- Da biste vidjeli repozitorij za dodatke, idite na stranicu za Shared Components:
  - Odaberi App Builder na glavnoj Workspace stranici.
  - Odlučite se o aplikaciji.
  - Odaberite Shared Components na početnoj stranici aplikacije.
- Pojavljuje se stranica za zajedničke komponente.
- Odaberi Plug-ins s Other Components izbornika.
- Odaberi View Plug-in Repository gumb.

Prikazuje se repozitorij za Oracle Application Express dodatke. Za pristup stranici s dodacima:

- Odi na stranicu Shared Components.
  - Odaberi App Builder na glavnoj Workspace stranici.
  - Odaberi aplikaciju.
  - Odaberi Shared Components na početnoj stranici aplikacije.
- Odaberi Plug-ins s Other Components izbornika. Kartica Plug-ins se automatski odabire kada se učita stranica s dodacima. Prikazuje se cijeli popis dodataka.

Za kreiranje dodatka:

- Odi na Shared Components stranicu.
  - Odaberi App Builder na glavnoj Workspace stranici.
  - Odaberi aplikaciju.
  - Odaberi Shared Components na počenoj stranici aplikacije. Otvara se Shared Components stranica.
- Odaberi Plug-ins na Other Components,
- Odaberi Create. Pojavljuje se čarobnjak Create Plug-in.
- Odaberi Next nakon odaberi metode za kreiranje dodatka unutar Create Plug-in.
- Unutar naziva:
  - Naziv – Unesite naziv dodatka.
  - Interni naziv - Unesite interni naziv dodatka. Ime treba biti jedinstveno unutar aplikacije.
  - Vrsta - odlučite koja vrsta komponente može koristiti ovaj dodatak. Opcije pod Callbacks i Standard Attributes se razlikuju ovisno o vrsti dodatka koji odaberete. Pogledajte pomoć na razini polja za više informacija.
  - Kategorija - pojavljuje se samo ako je navedena vrsta dinamičke radnje. Na korisničkom sučelju odaberite kategoriju u kojoj je dodatak naveden.
- Unutar Subscription reference Master Plug-in Form - Odaberite dodatak s popisa da biste ga koristili kao model za druge dodatke ovog radnog prostora. Ako ne želite da ovo bude glavna kopija ovog dodatka, ostavite polje prazno.
- Unutar Source:
  - PL/SQL kod - Unesite PL/SQL anonimni blok koda koji sadrži korake potrebne za prikazivanje, provjeru valjanosti, pokretanje i izvršavanje povratnih poziva ovog dodatka. Alternativno možete pohraniti ovaj kod u PL/SQL paketu u bazi podataka.
  - Do not validate PL/SQL code - Odaberite ovu opciju samo za parsiranje PL/SQL koda tijekom izvođenja. Ako nije, dodatak se izgrađuje nakon što se kôd analizira.
- Unutar Callbacks, konfigurirajte atribute. Vrsta dodatka određuje atribute koji se prikazuju.

Pogledajte pomoć na razini polja da ispitate uzorke i saznate više o atributu.
- Unutar User Interfaces, odaberite koje prikaze App Builder mora podržavati za ovaj dodatak. Opcije uključuju:
  - Desktop.
  - Mobile.
- Za Standard Attributes, odaberite atribute koji se odnose na ovaj dodatak. tandard Attributes odbacuju određene dodatke.
- Unutar Information:
  - Version -Da biste odredili verziju dodatka, unesite niz.
  - About URL - Unesite URL koji vodi do početne stranice autora dodatka ili do više pojedinosti o dodatku.
- Odabeite Create. Plug-in.

Za uvoz dodatka:
- Odite na stranicu Shared Components:
  - Odaberite App Builder na početnoj Workspace stranici.
  - Odaberite aplikaciju.
  - Odaberite Shared Components na početnoj stranici aplikacije.
- Odaberite Plug-ins unutar Other Components
- Odaberite Import. Otvara se stranica uvoza dodatka.
- Za Specify File:
  - Import file - Unesite ili dođite do naziva datoteke za uvoz.
  - File Type - Odaberite Plug-in.
  - File Character Set - Odaberite kodiranje skupa znakova za datoteke za uvoz.
  - Odaberite Next.
- Za File Import Confirmation, odaberite Next.
- Za Install, odaberite Install Plug-in.

Za izvoz dodatka iz Plug-in stranice:
- Odite na stranicu Shared Components:
  - Odite na Workspace početnu stranicu i odaberite App Builder.
  - Odaberite aplikaciju.
  - Odite na početnu stranicu aplikacije i odaberite Shared Components.
- Unutar Other Components, odaberite Plug-ins.
- Unutar Tasks, odaberite Export Plug-in.
- Na stranici Export Plug-in:
  - Application - Odaberite aplikaciju iz koje želite izvesti dodatak.
  - Plug-in - odaberite dodatak.
  - File Format - odaberite format datoteke za izvoz.
  - Odaberite Export
- Odaberite Export.

U nastavku je dan jedan primjer dodatka. Primjer se odnosi na ažurirani primjer aplikacije kalendara i dodatka za dinamičke radnje. Ovaj primjer dodatka može se koristiti za oblikovanje dana u novoj stavci alata za odabir datuma. Korisnici ga mogu onemogućiti, dodati opise alata ili razrede svakom danu u dnevnoj mreži alata za odabir datuma. Ne postoje preduvjeti za korištenje ovog dodatka. Kako ga koristiti:
- Kreirajte aplikaciju
- Uvezite dodatak u Shared Components - Plug-ins
- Dodaj novi alat za odabir datuma
- Dodajte Dynamic Action na Page Load i kao radnju odaberite uvezeni dodatak
- Postavite URL iCalendar standardnog formata datoteke ili unesite SQL upit koji bi trebao formatirati kalendarske dane

- Odaberite Affected Element željeni birač datuma i pokrenite stranicu

Ovo je poglavlje pokazalo da korištenje dodataka ima nekoliko prednosti. Koliko god jak bio, niti jedan softver ne može pružiti sve funkcije za svakog korisnika. Prazninu između forme i funkcije popunjavaju dodaci. Oni olakšavaju dodavanje određenih funkcija softveru i aplikacijama. Kao takvi, dodaci štede vrijeme. Najveća prednost korištenja dodataka je ta što smanjuju količinu vremena potrebnog za razvoj, što smanjuje ukupne troškove. Nadalje, dodaci omogućuju korisnicima prilagodbu značajki i funkcionalnosti. Većina dodataka omogućuje uključivanje i isključivanje određenih postavki. Jednostavan je postupak uklanjanja dodatka, ako postoji potreba za tim.

11.5 Pitanja

1. Što su ogledne aplikacije i koje su njihove prednosti?
2. Što su početne aplikacije i koje su njihove prednosti?
3. Koje su prednosti dodataka?

11.6 Odgovori

1. Ogledne aplikacije su kreirane kako bi istaknule određene mogućnosti i služile kao razvojni priručnik za korištenje određenih značajki.
2. Starter Apps korisne su aplikacije koje nude samostalna rješenja, stvorene da zadovolje jednostavne zahtjeve koji ne zahtijevaju nepotrebno složen odgovor.
3. Dodaci omogućuju stvaranje visoko prilagođenih komponenti kako bi poboljšali funkcionalnost, izgled i jednostavnost svojih aplikacija. Dodaci omogućuju korisnicima dodavanje dodatnih funkcija APEX aplikacijama pomoću dodataka koji nisu ugrađeni u platformu.



12. Kako upravljati pakiranim i višejezičnim aplikacijama?

R. LESKOVAR, U. RAJKOVIČ, A. BAGGIA

12.1 Aplikacija i pakirana aplikacija

Ovo poglavlje će vas naučiti kako iskoristiti potencijal Quick SQL značajke za generiranje aplikacije bez da je potrebno ikakvo znanje DDL-a i to potpuno unutar Oracle APEX-a. Nakon generiranja aplikacije iz Quick SQL-a pripremit će se zapakirana aplikaciju i migrirat će se u novi radni prostor (tj. "mjesto proizvodnje").

U poglavlju 11 su prikazane aplikacije koje se isporučuju unutar APEX-a. Te se aplikacije mogu svrstati u dvije skupine: a) produktivnost i b) ogledne aplikacije. Aplikacije za produktivnost rješavaju specifične poslovne potrebe. Ugrađene značajke manje-više u potpunosti zadovoljavaju zahtjeve koje nameće poslovna situacija. Primjeri aplikacija pokazuju različite funkcionalnosti. Obje se mogu prilagoditi i u potpunosti ispuniti poslovne potrebe. Ove aplikacije su primjeri pakiranih aplikacija. Kada instalirate (APEX izraz je uvoz) ove zapakirane aplikacije, možete primijetiti da su objekti baze podataka (tablice, nizovi, okidači, funkcije, procedure i paketi) stvoreni i da su podaci popunjeni u tablicama.

Najjednostavnija definicija **packaged application** kaže da je to unaprijed izgrađena aplikacija koja se može instalirati i konfigurirati u Oracle APEX radnom prostoru. APEX koristi dva važna koncepta usko povezana s migracijom aplikacija (izvoz i uvoz): a) prateći objekti and b) dijeljene komponente. Prateći objekti kontroliraju instalaciju, nadogradnju i deinstalaciju definiranjem preduvjeta, nizova zamjene, opcija izgradnje, provjera valjanosti prije instalacije, skripti i poruka. Dijeljene komponente definiraju aplikaciju, logiku aplikacije, sigurnost, navigaciju i pretraživanje, korisničko sučelje, datoteke i izvješća, izvore podataka, tijekove rada (ako su instalirani), globalizaciju i druge komponente. Pakirane aplikacije su one s definiranim pratećim objektima. U ovom trenutku možete se zapitati: "Zašto i kako programeri mogu kreirati prilagođene pakirane aplikacije?".

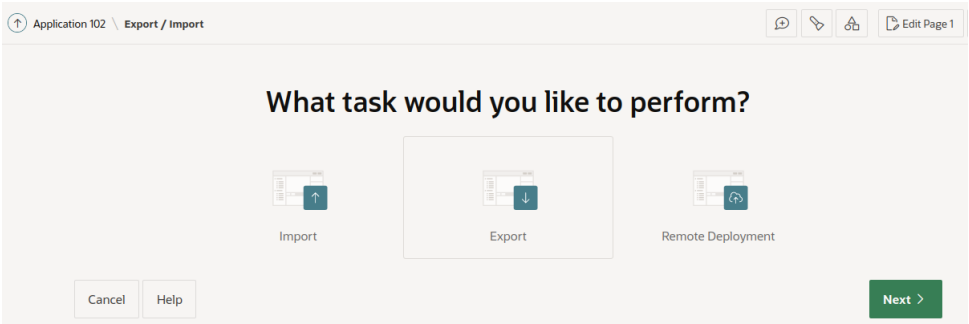
Odgovor na "zašto": programeri bi izradili prilagođenu pakiranu aplikaciju ako postoji zahtjev za migriranje aplikacije u novi radni prostor s pratećim objektima i zajedničkim komponentama. Definicije objekata baze podataka (s DDL izjavama) i operacije na tim objektima (DML izjave) izvršavaju se tijekom instalacije, ažuriranja i deinstalacije. Pojednostavljeno: tijekom instalacije, tablice se mogu kreirati i popuniti podacima, tijekom nadogradnje, tablice se mogu mijenjati, a tijekom deinstalacije, tablice se mogu skratiti ili odbaciti. Slično bi se moglo nametnuti za druge objekte baze podataka kao što su okidači, sekvence, pohranjene procedure, funkcije i paketi. Ključna prednost paketne aplikacije je da je spremna za rad s definicijama podataka i samim

podacima.

Odgovor na "kako" uključuje sljedeće korake:

- definirati opseg aplikacije,
- definirati opseg aplikacije,
- kreirati i izvršiti skriptu za definiranje tablica,
- kreirati i izvršiti skriptu za unos podataka,
- generiranje aplikacije,
- napraviti prilagodbe u aplikaciji,
- pripremiti zapakiranu aplikaciju i testirati ju u novom radnom prostoru.

Prije nego krenemo korak po korak, pogledajmo čarobnjaka za uvoz i izvoz aplikacije u APEX-u. Za premještanje aplikacije u novi radni prostor, aplikaciju je potrebno izvesti, a zatim uvesti. Čarobnjak za izvoz u APEX-u piše tekstualnu datoteku (PL/SQL izjave) na: a) računalu programera ili b) u udaljenom radnom prostoru putem REST usluge (vidi Sliku 12.1).



Slika 12.1: Čarobnjak za uzvoz i izvoz aplikacije

Prilikom izvoza postoji nekoliko opcija poput dijeljenja u više datoteka, odabira skupa znakova datoteke, uključivanja ili isključivanja javnih ili privatnih izvješća i slično. U većini slučajeva, početnik jednostavno pusti APEX da primijeni zadane postavke. Zatim se programer prijavljuje u novi radni prostor i pokreće čarobnjaka za uvoz koji učitava izvezenu tekstualnu datoteku. Definicije tablica, pogleda, okidača, sekvenci, funkcija, procedura i paketa ne migriraju s aplikacijom ako ne pripremimo prateće objekte. Ako ciljni radni prostor već sadrži definicije objekata baze podataka i podatke u tablicama, trebali bismo pripremiti prateće objekte za rukovanje takvim situacijama.

Prvi dio ovog poglavlja će koristiti APEX čarobnjake što je više moguće. Zatim će se prikazati jednostavne SQL INSERT izjave koje čarobnjak ne podržava u APEX-u. Za generiranje INSERT naredbi čitač može uvesti podatke u proračunsku tablicu i zatim izvesti INSERT naredbe s drugim alatima (i.e. SQL Developer ili TOAD). Aplikacija u ovom poglavlju uvest će komponentu stranice koja se naziva lista vrijednosti (engl. list of values, LOV). Lista vrijednosti pripada zajedničkim komponentama. Lista vrijednosti ima dvije svrhe: korisnik će ispuniti polje obrasca na web stranici vrlo jednostavno (kroz padajući izbornik ili radio gumb) i to samo propisanim vrijednostima kako bi se izbjegle pogreške pri tipkanju. Dostupne su dvije vrste liste vrijednosti: dinamičke kojese temelje na SQL upitu dok statičke nude fiksni skup vrijednosti. Prednost dinamičkog popisa vrijednosti je u tome što domena ovisi o tablici ili pogledu. Krajnji korisnik može promijeniti domenu umetanjem, ažuriranjem i brisanjem redaka u tablici. Nedostatak je što se tablica mora kreirati i održavati. Ako se domena ne mijenja, a broj redaka u tablici je mali, može se primijeniti statički popis vrijednosti. U tom slučaju samo programer može promijeniti domenu (ona je "tvrdo kodirana"). Drugi dio će demonstrirati izradu zapakirane aplikacije, njezin izvoz i uvoz u novi radni prostor.

Treći dio poglavlja dat će upute za izradu višejezične aplikacije. Kao dodatni materijal za učenje dane su tri izvezene aplikacije, kao i video isječak koji detaljno prikazuje razvojni proces.

12.2 Aplikacija

12.2.1 Opseg aplikacije

Cilj je aplikacije upravljanje poslovima i kompetencijama za HR odjel. U tvrtki može biti nekoliko stotina radnih mjesta. Jedan posao može uključivati određeni podskup svih poznatih kompetencija. Potrebnu razinu kompetencije propisuje HR služba za svako radno mjesto. Razine kompetencije su deskriptivna sposobnost: pomaganja, korištenja, ovladavanja, definiranja i inoviranja. Dakle, ideja je razviti aplikaciju za pripremu i upravljanje profilom posla. Aplikacija koristi tri tablice: JOB, COMPETENCE i JOB_COMPETENCE s minimalnim brojem polja za podatke (u svrhu jednostavnosti i konciznosti). Svaka tablica ima primarni ključ i tablica JOB_COMPETENCE ima dva vanjska ključa kako bi se svaki red povezao s tablicama JOB i COMPETENCE. Vanjski ključevi osiguravaju konzistentnost podataka: red u JOB_COMPETENCE tablici može postojati samo ako postoje redovi s istim primarnim ključem u tablicama JOB i COMPETENCE. Kako bi se postigla manja strogost, nazivi tablica imaju prefiks "CH12_".

12.2.2 Kreiranje tablica

Tablica u APEX-u se kreira:

- bez znanja jezika opisa podataka, odabirom Object browser. Ilustriranje ovog procesa s mnogo slika bi zauzelo previše prostora u ovoj knjizi,
- sa znanjem DDL-a u SQL Workshop > SQL Commands unosom sintaktički ispravnih naredbi.
- s temeljnim znanjem modeliranja podataka (vidi Poglavlje 2) i Quick SQL.

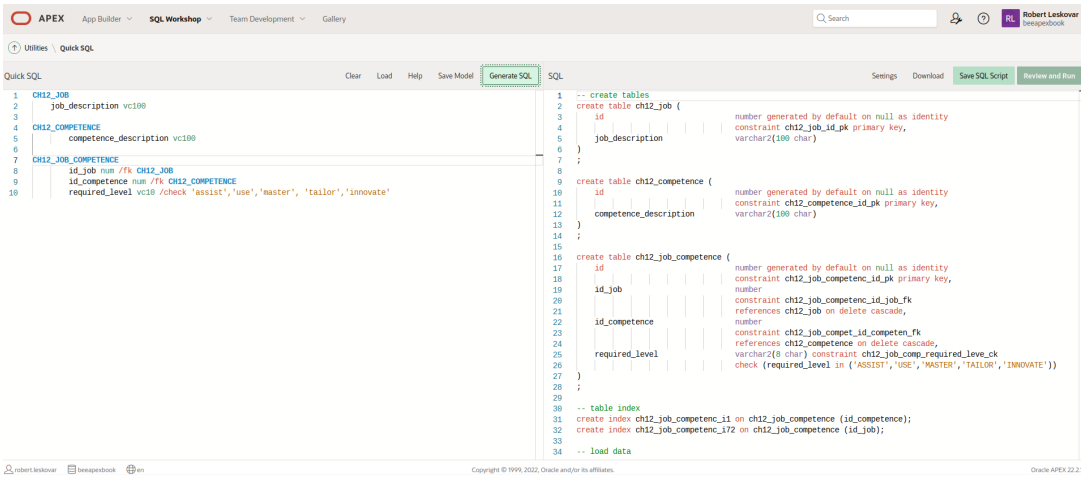
Kako bi se osiguralo široko razumijevanje i konciznost, Quick SQL će se koristiti (nađi Quick SQL unutar SQL Workshop > Utilities > Quick SQL). Unosi se sljedeći tekst:

```
CH12_JOB
  job_description vc100

CH12_COMPETENCE
  competence_description vc100

CH12_JOB_COMPETENCE
  id_job num /fk CH12_JOB
  id_competence num /fk CH12_COMPETENCE
  required_level vc10 /check 'assist','use','master', 'tailor','innovate'
```

Nakon unosa gornjeg teksta i odabira **Generate SQL** gumba, pojavljuju se sljedeći ekran (vidi Sliku 12.2):

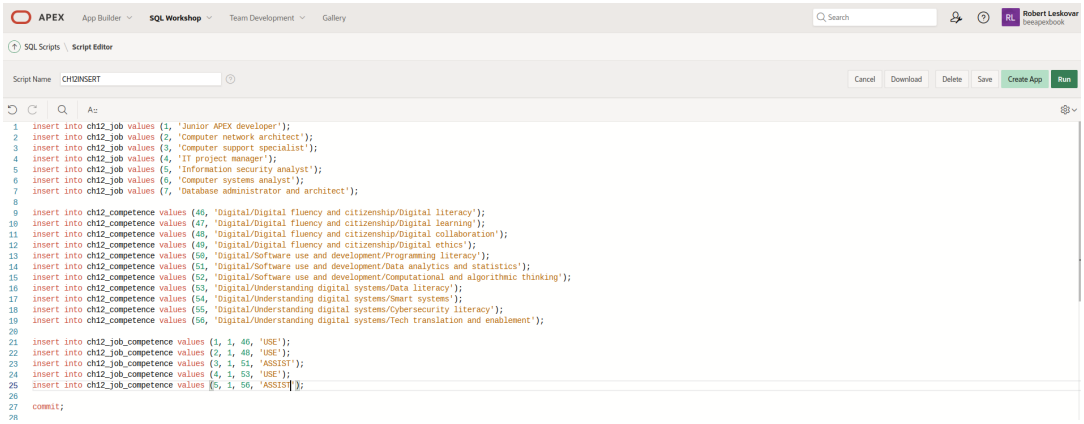


Slika 12.2: Pretvorba Quick SQL u SQL naredbe

Odaberi **Save SQL Script** gumb i nazovi skriptu CH12CREATE. Sada možemo izvršiti skriptu. Vratite se na Scripts, pronađite ikonu Run u retku gdje je skripta CH12CREATE. Odaberite **Run** ikonu. Trebalo bi uspješno kreirati tri tablice i tri indeksa. Možete provjeriti kartice u Object browser.

12.2.3 Unos podataka

U APEX-u odaberite SQL Workshop > SQL Scripts > Create. Sada je vrijeme da napišemo SQL INSERT naredbe kako bi se tablice ispunile podacima. Tablica CH12_JOB će imati sedam redaka od kojih svaki opisuje jedan posao. Tablica CH12_COMPETENCE će imati jedanaest redaka od kojih svaki opisuje jedanaest digitalnih kompetencija definiranih od: McKinsey's DELTAs (različiti elementi talenta). Unijet ćemo i pet kompetencija poslu 'Junior Apex developer'. Nazovimo skriptu CH12INSERT i napišimo skriptu (vidi Sliku 12.3).



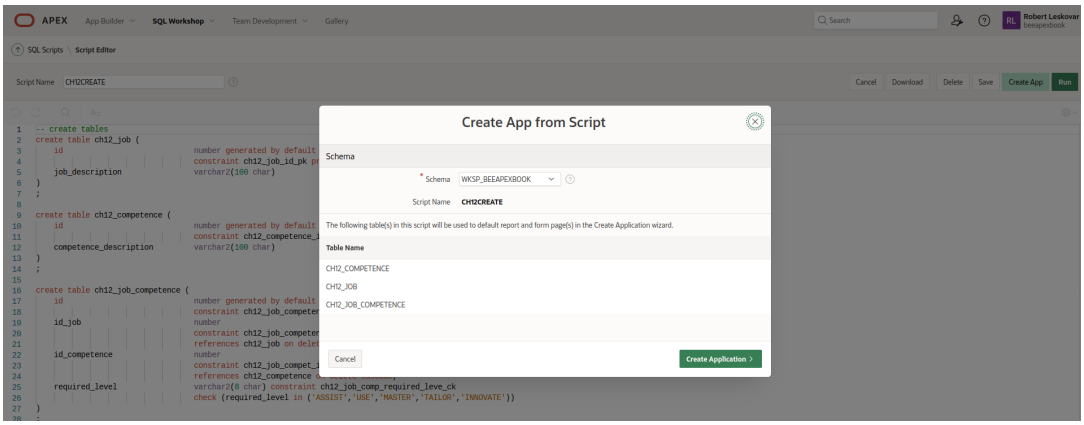
Slika 12.3: Skripta CH12INSERT unesi podatke u tri tablice i napravi transakcije

Odaberite **Create** gumb. Izvedite skriptu CH12INSERT. Vratite se na Scripts, odaberi **Run** ikonu u redu gdje je skripta CH12INSERT prikazana. Odaberite **Run now** gumb. Trebalo bi se uspješno stvoriti dvadeset četiri retka. Možete provjeriti sadržaj tablica u Object browser.

12.2.4 Generiranje aplikacija

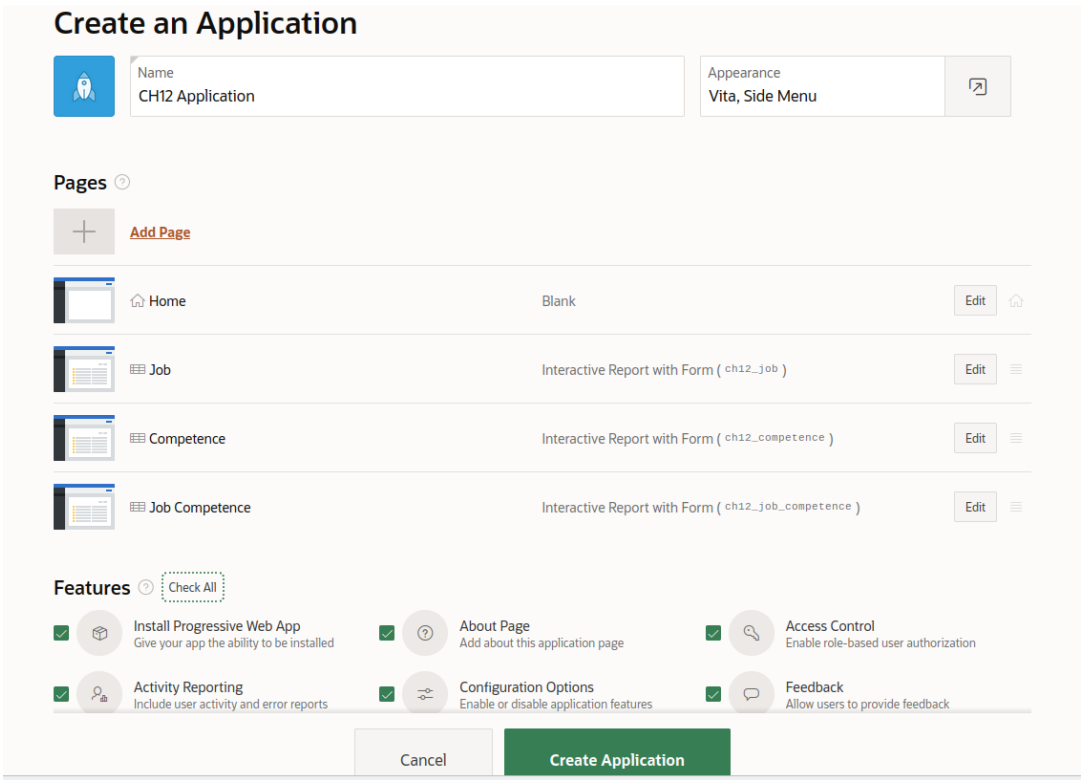
Nakon popunjavanja tablica testnim podacima idite na SQL Workshop > Scripts i uredite skriptu CH12CREATE. Odaberite **Create App** gumb (vidi Sliku 12.4).Možete primijetiti da su navedene

tri tablice.



Slika 12.4: Kreiranje aplikacije iz skripte

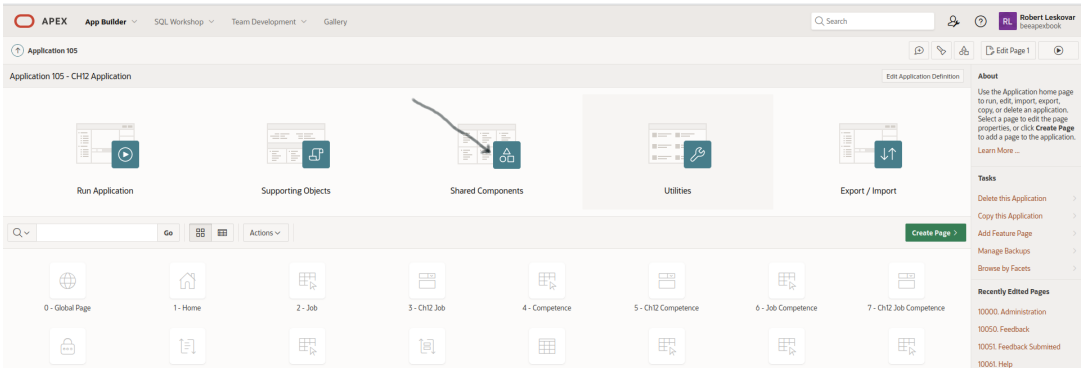
Odaberite Create Application gumb. Unesite naziv aplikacije (npr.CH12 Application), odaberite Check all features i Create Application gumb (vidi Sliku 12.5).



Slika 12.5: Definiranje naziva aplikacije i svih značajki

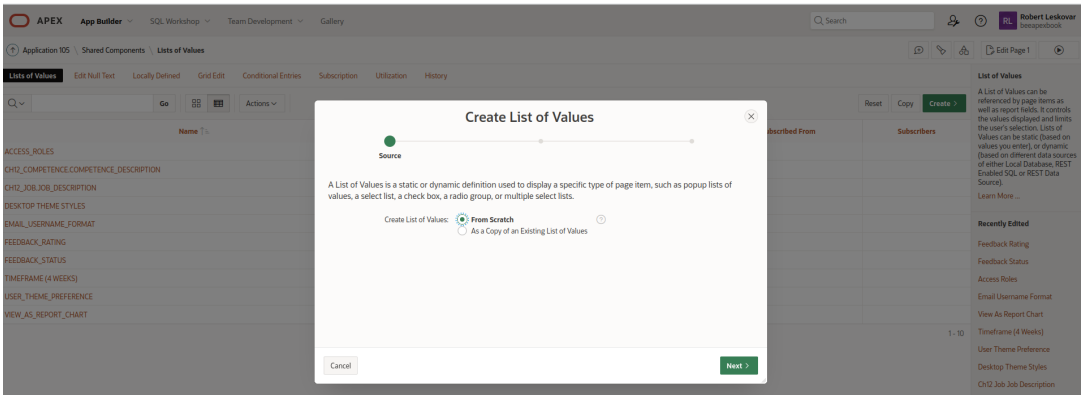
Nakon što je generiranje završeno, mogli bismo pokrenuti aplikaciju. U novijim verzijama APEX-a generator također stvara dvije liste vrijednosti (CH12_COMPETENCE.COMPETENCE_DESCRIPTION i CH12_JOB.JOB_DESCRIPTION), označene kao LOV. Liste vrijednosti su komponente na stranici gdje je jedna vrijednost prikazana i razumljiva korisniku, dok stranica zapravo koristi odgovarajući ključ ili kod za umetanje, ažuriranje ili brisanje. Na primjer: na obrascu stranice prikazuje se prezime zaposlenika i odgovarajući identifikacijski broj zaposlenika se pohranjuje kada se pritisne gumb za potvrdu. Generirani LOV-ovi temelje se na tablicama i nije potrebno prethodno znanje

SQL-a. U terminologiji APEX-a nazivaju se dinamičkim jer bi promjena u tablici proizvela novu listu vrijednosti. Liste vrijednosti se mogu kreirati u Shared Components > List of Values i odrediti kako će se definirati. U ovom poglavlju učimo kako izraditi dinamički popis vrijednosti (na temelju jednostavnog SQL upita) i statički na temelju fiksnog skupa vrijednosti. Upotrebom liste vrijednosti (LOV) na stranicama aplikacije učinit ćemo aplikaciju otpornijom na pogreške i također lakšom za korištenje. To znači da korisnik ne može popuniti proizvoljnu vrijednost u polju obrasca čime ublažavamo pogreške pri unosu. Odite na Shared components (vidi Sliku12.6).



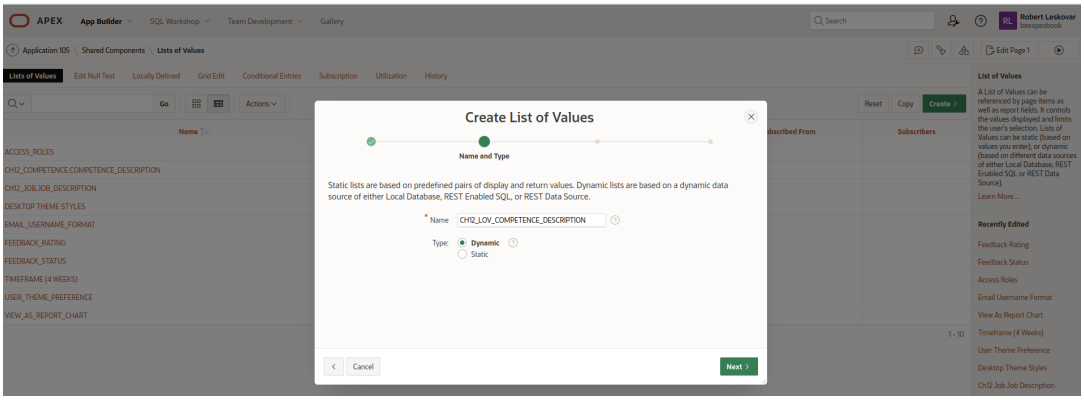
Slika 12.6: Odabir Shared components

U shared components odaberite List of values (u Other Components) te Create. Napravit ćemo listu vrijednosti od početka (vidi Sliku 12.7).



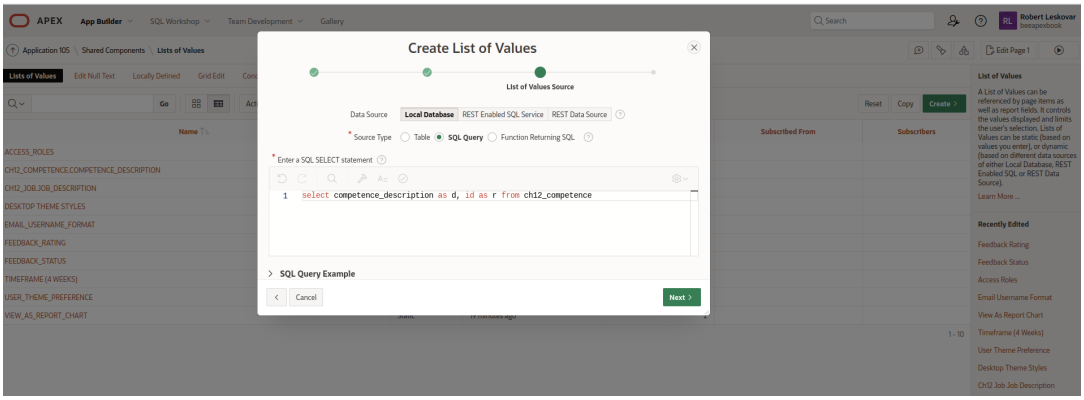
Slika 12.7: Kreiranje liste vrijednosti od početka.

Nazovite LOV kao CH12_LOV_COMPETENCE_DESCRIPTION i odaberite Dynamic (vidi Sliku 12.8)



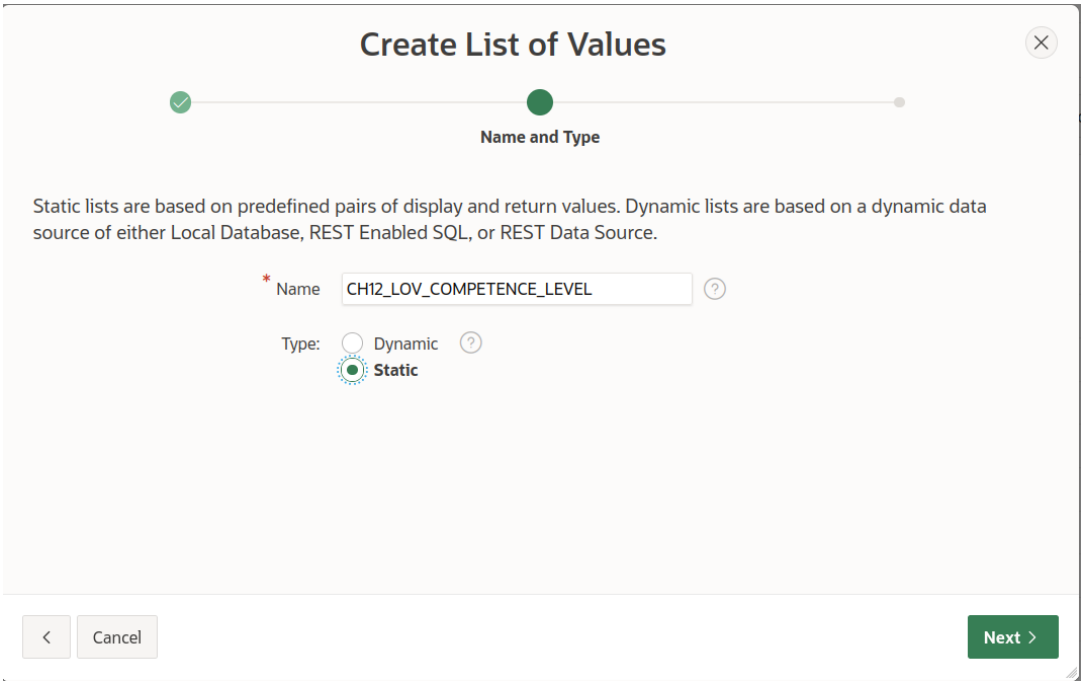
Slika 12.8: Definiranje naziva i tipa: CH12_LOV_COMPETENCE_DESCRIPTION je dinamički

Odaberite SQL Query kao izvor podataka i definirajte "select competence_description as d, id as r from ch12_competence" kao SQL select izjavu (vidi Sliku 12.9).

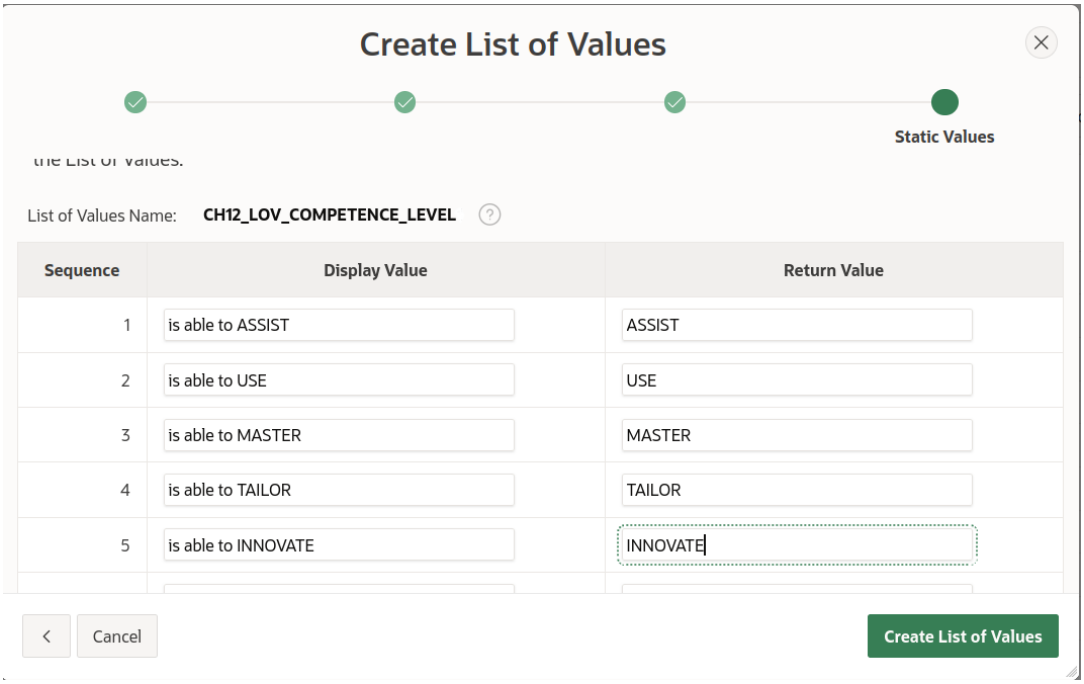


Slika 12.9: Unos SQL SELECT naredbe

Lista vrijednosti prikazuje opis kompetencije ("display" stupac) na stranici obrasca, a vraća ID ("return" stupac). Sljedeća LOV će biti statična, nazvana CH12_LOV_COMPETENCE_LEVEL i kreirana kako je prikazano na Slici 12.10 i Slici 12.11.

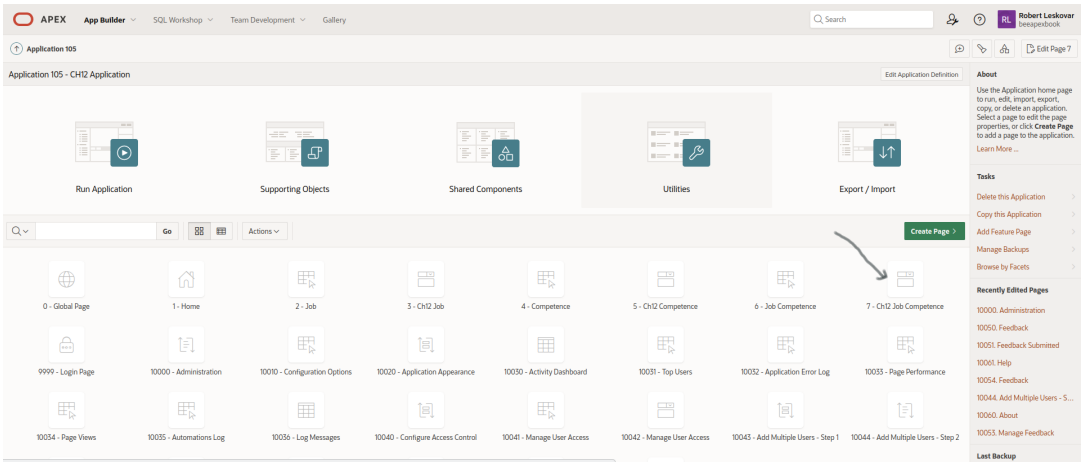


Slika 12.10: Definiranje naziva i tipa: CH12_LOV_COMPETENCE_LEVEL je statično



Slika 12.11: Definiranje vrijednosti za prikaz i vraćanje: CH12_LOV_COMPETENCE_LEVEL je statično.

Idite na stranicu aplikacije (vidi Sliku 12.12). Odaberite stranicu 7 (Ch12 Job Competences).



Slika 12.12: Odaberite stranicu 7 (Ch12 Job Competences)

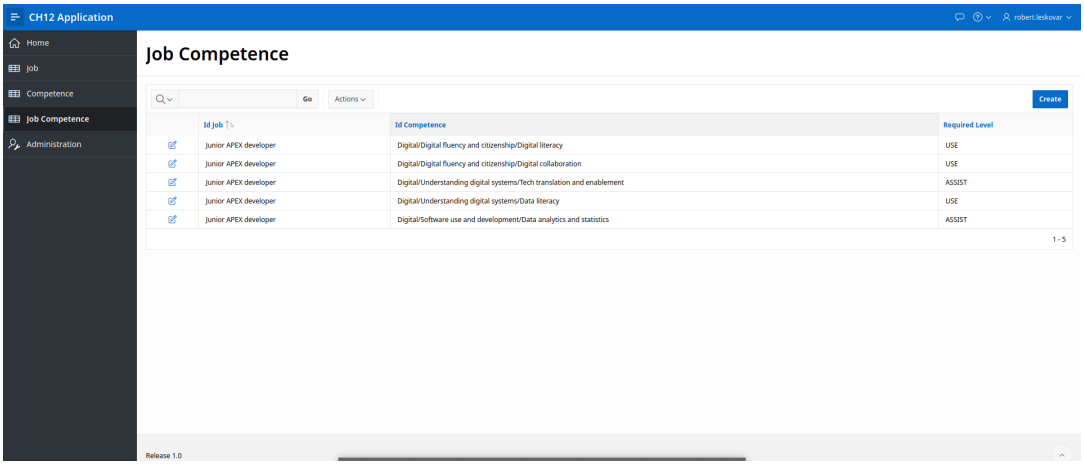
Modificirajte stavku stranice P7_ID_COMPETENCE:

- tip je "Select list" u Identification
- tip je "Shared components" u List of values
- lista vrijednosti je CH12_LOV_COMPETENCE_DESCRIPTION List of values

Modificirajte stavku stranice P7_REQUIRED_LEVEL:

- tip je "Select list" u Identification
- tip je "Shared components" u List of values
- lista vrijednosti je CH12_LOV_COMPETENCE_LEVEL List of values

Spremite stranicu 7, navigirajte stranicu 6 i pokrenite aplikaciju. Trebali bi vidjeti kako je na slici 12.13.

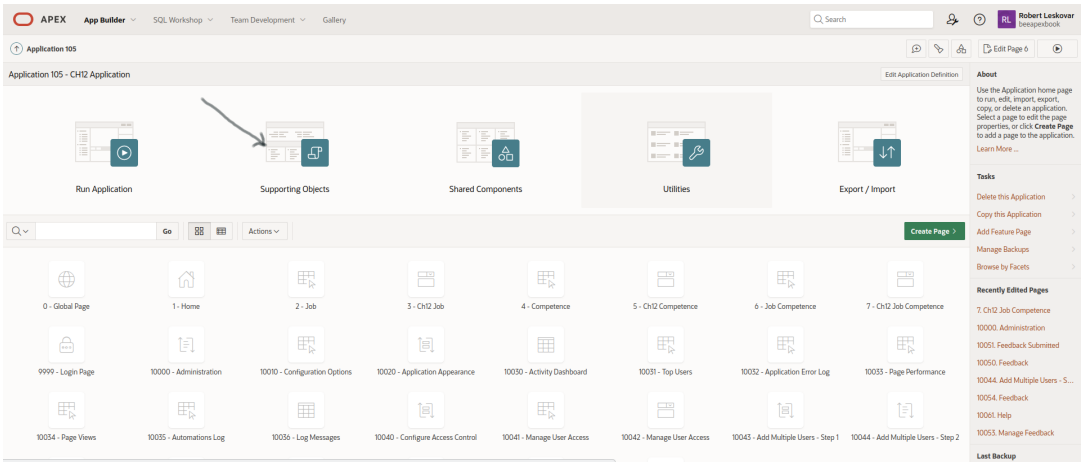


Slika 12.13: Odaberite stranicu 7 (Ch12 Job Competences)

Sada uredite prvi unos. Oba popisa liste daju očekivane podatke.

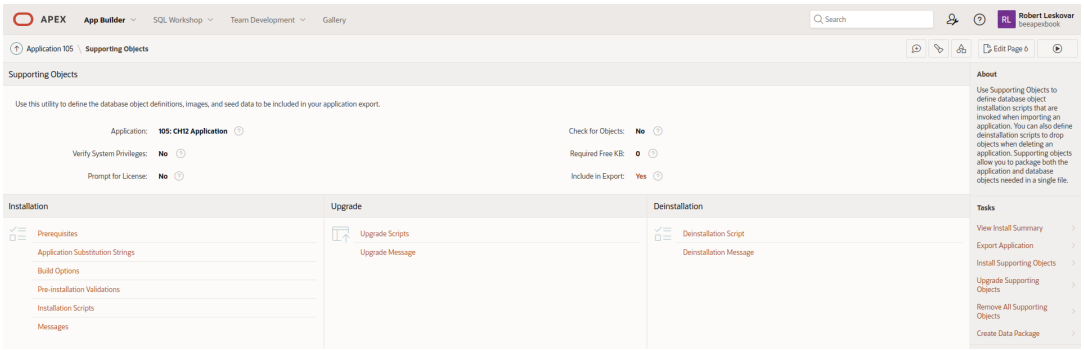
12.3 Pakirane aplikacije

Pošto smo već kreirali skripte (CH12CREATE, CH12INSERT) brzo ćemo kreirati pakirane aplikacije. Dođite do aplikacije i odaberite Supporting Objects (vidi Sliku 12.14).



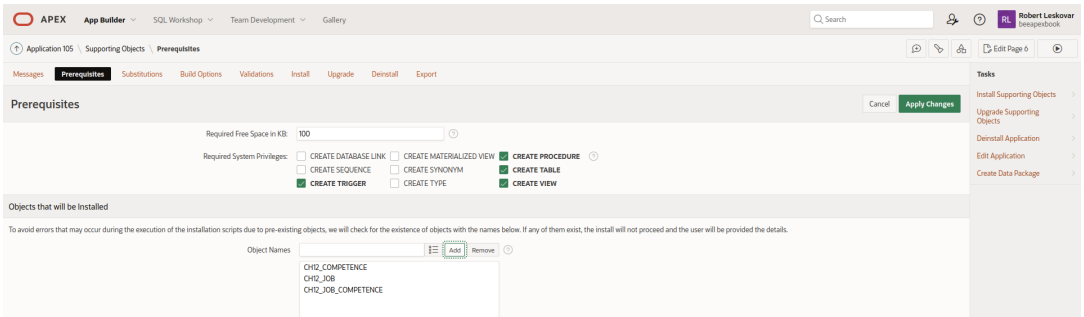
Slika 12.14: Odaberite Supporting Objects

Sada možemo postaviti instalaciju (preduvjeti, nizovi zamjene aplikacije, opcije izrade, provjere valjanosti prije instalacije, instalacijske skripte i poruke), nadogradnju (skripte za nadogradnju, poruka o nadogradnji), deinstalaciju (skripta za deinstalaciju, poruka o deinstalaciji). Vidi Sliku 12.15.



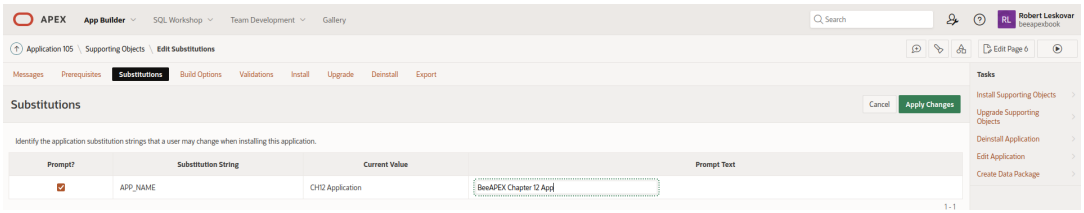
Slika 12.15: Postavljanje preduvjeta

Kako bismo zaštitili korisnika od slučajnog brisanja postojećih podataka, primijenit ćemo provjeru postojanja triju tablica (vidi Sliku 12.16). Odaberite gumb Apply changes.



Slika 12.16: Postavite provjeru postojanja tri tablice

U Application Substitution String pitat ćemo treba li naziv aplikacije biti "BeeAPEX Chapter 12 App" (vidi Sliku 12.17).



Slika 12.17: Postavite upit za preimenovanje aplikacije

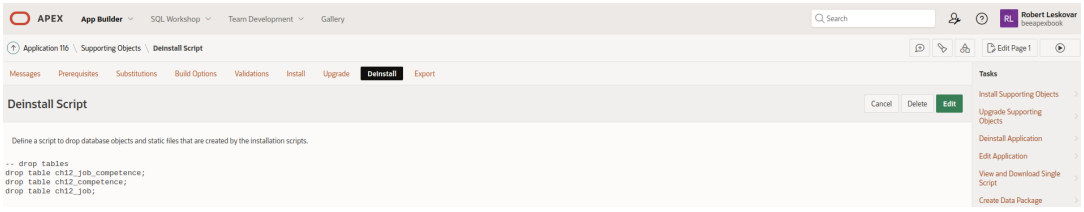
Odaberite gumb Apply changes.

Za ovu pakiranu aplikaciju također ćemo primijeniti sljedeću postavku u Supporting Objects:

- preduvjeti (vidi Sliku 12.16,)
- podnizovi zamjene aplikacije (vidi Sliku 12.17,)
- instalacijske skripte: budući da smo već pripremili skripte CH12CREATE i CH12INSERT koristit ćemo njihov sadržaj (vidi Sliku 12.18,)
- skripte za deinstalaciju (vidi Sliku 12.19,)
- skup poruka: poruka dobrodošlice na "Chapter 12: ...", licencna poruka na "CC BY", poruka o uspješnoj instalaciji na "Success!", poruka o neuspješnoj instalaciji na "Failure", poruka za deinstalaciju na "Application CH12 deinstalled".

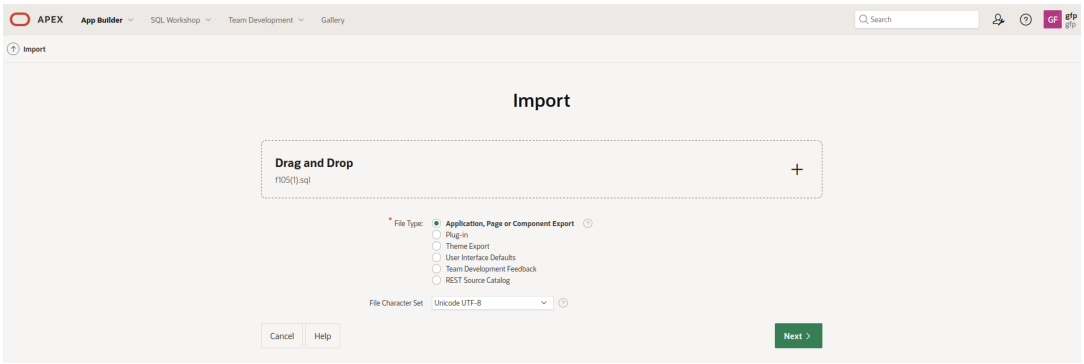


Slika 12.18: Postavite instalacijske skripte



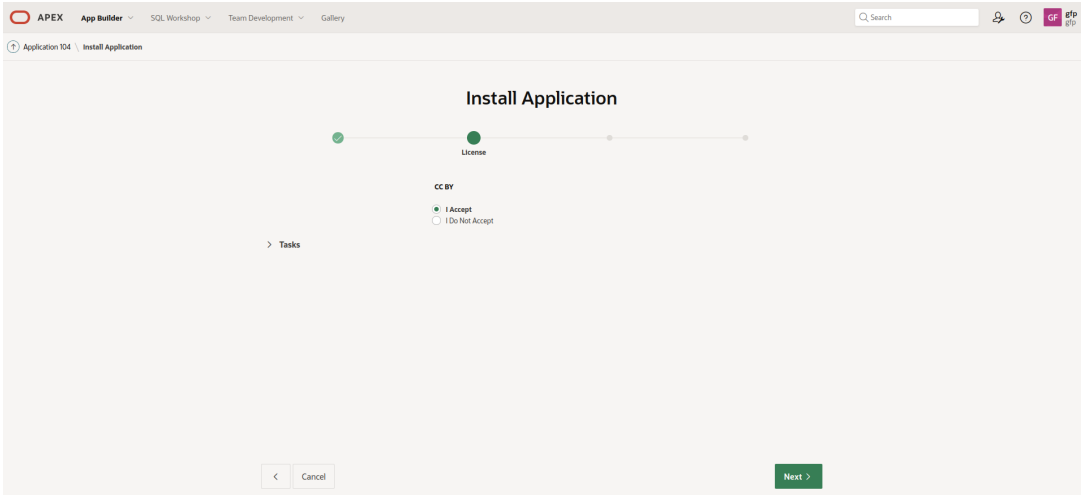
Slika 12.19: Postavite deinstalacijske skripte

Sada izvezite zapakiranu aplikaciju s čarobnjakom (vidi Sliku 12.1). Možete prihvatiti sve postavke izvoza koje je predložio APEX. Spremite izvezenu aplikaciju na svoje računalo i zapamtite naziv datoteke. Prijavite se u novi radni prostor. Pritisnite čarobnjak za uvoz u novom radnom prostoru i povucite naziv datoteke u zonu za povlačenje i ispuštanje (vidi Sliku 12.20).

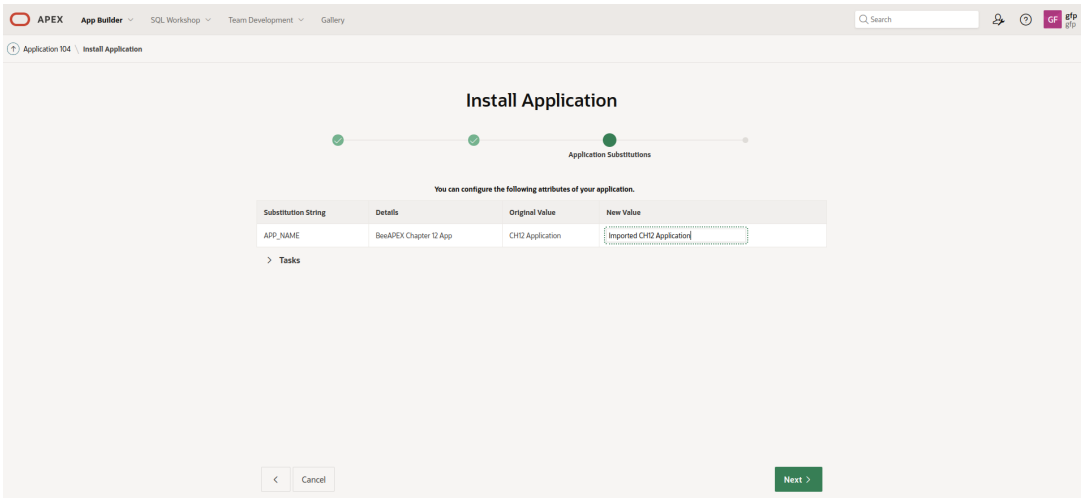


Slika 12.20: Uvezite aplikaciju u drugi radni prostor

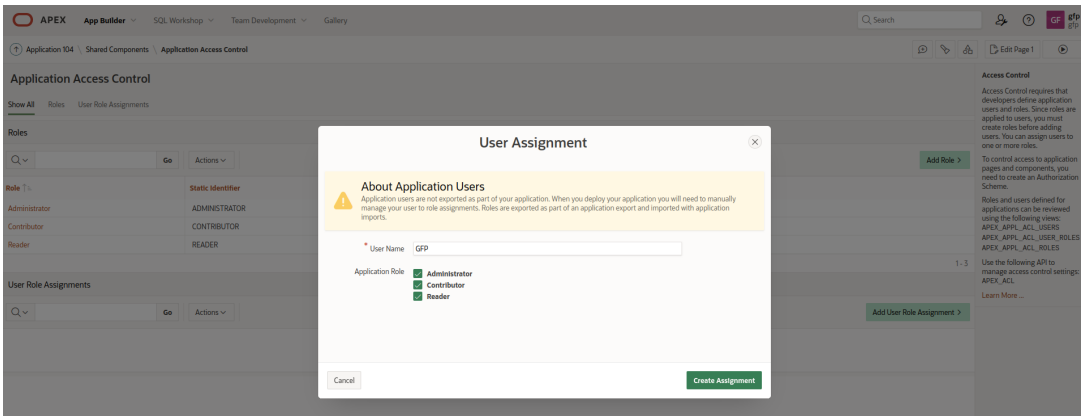
Nastavite odabirom gumba Next (također za potvrdu uvoza datoteke) i gumba Install application. Potvrdite instalaciju pratećih objekata s gumbom Next, prihvatite licencu (vidi Sliku 12.21), preimenujte aplikaciju "Imported CH12 Application" (vidi Sliku 12.22) i odaberite Install gumb. Nemojte još pokretati aplikaciju jer morate zasebno postaviti dopuštenja za novog korisnika. Korisnici i korisnička dopuštenja ne izvoze se iz sigurnosnih razloga. Samo jedna napomena: za migraciju korisnika i dopuštenja mogli bismo duboko zaroniti u APEX API i napisati skripte za instalaciju i deinstalaciju. Idite na dijeljene komponente u uvezenoj aplikaciji. Odaberite kontrolu pristupa aplikaciji i trenutno prijavljenom korisniku dajte odgovarajuća dopuštenja. U našem slučaju ćemo dodati APEX korisničku ulogu GFP administratora, suradnika i čitača (vidi Sliku 12.23). Potvrdite s Add Assignment gumbom.Sada pokrenite aplikaciju kao korisnik s danim dopuštenjima.



Slika 12.21: Ugovor o licenci



Slika 12.22: Preimenujte uvezenu aplikaciju



Slika 12.23: Dodavanje uloge korisniku

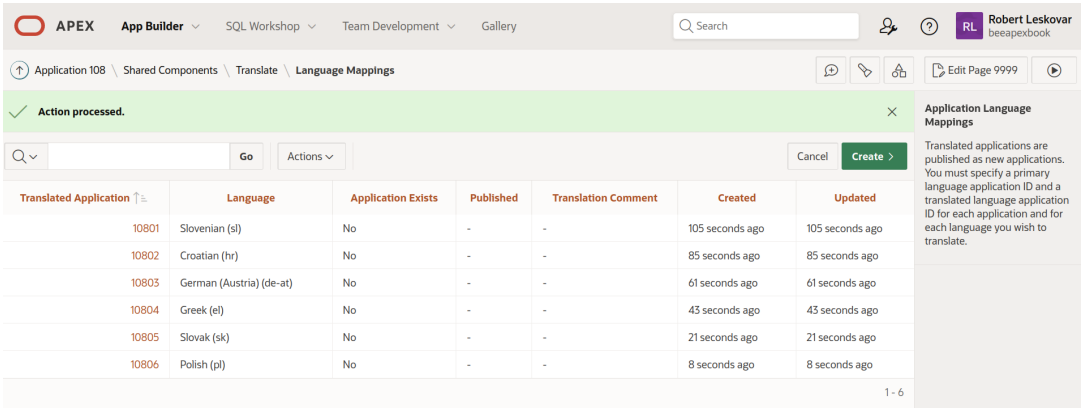
12.4 Višejezične aplikacije

Ova će tema pokriti samo jedan aspekt višejezičnih aplikacija - nizove na stranicama aplikacije kao što su izvješća i obrasci. Primarni razlozi za prilagodbu aplikacije određenom jeziku su: a) korisnici tečno govore samo druge jezike i b) postoje zahtjevi koje nameću organizacije ili države da podržavaju više od jednog jezika. Opseg prilagodbe može biti ograničen na prijevod oznaka stavki stranice, poruka aplikacije, APEX internih poruka, formata brojeva i datuma ili toliko složen kao prijevod teksta pohranjenog u tablicama baze podataka (tj. status, razina, ocjena). Specifični kulturni i jezični kontekst može uzrokovati da prijevod čistog niza bude smiješan, uvredljiv ili nestručan. Dakle, lokalizacija aplikacije je puno više. APEX ima mnogo mogućnosti za primjenu prijevoda (na temelju primarnog jezika aplikacije, preglednika, postavki aplikacije, stavke, sesije). Sljedeći dio pruža najjednostavnije upute za implementaciju više jezika u aplikaciji promjenom primarnog jezika aplikacije. Koraci su:

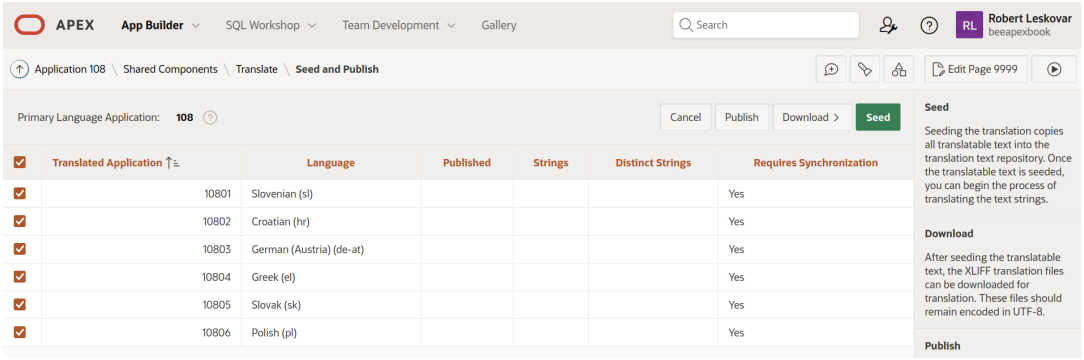
- navigirajte do Shared components > Globalization > Application Translations > Define application languages: odaberite gumv Create; za svaki jezik postavite jedinstveni cijeli broj i vrijednost (npr. dodajte dvije znamenke broju prijave npr 10801 za sl; 10802 za hr; 10803 za de-at; 10804 za el; 10805 za sk; 10806 za pl). Slika 12.24 prikazuje sve jezike.
- navigirajte do Shared components > Globalization > Application translations > Seed translatable text. Odaberite jezike prikazane na Slici 12.25. Odaberite Seed gumb i pričekajte dok

proces ne završi.

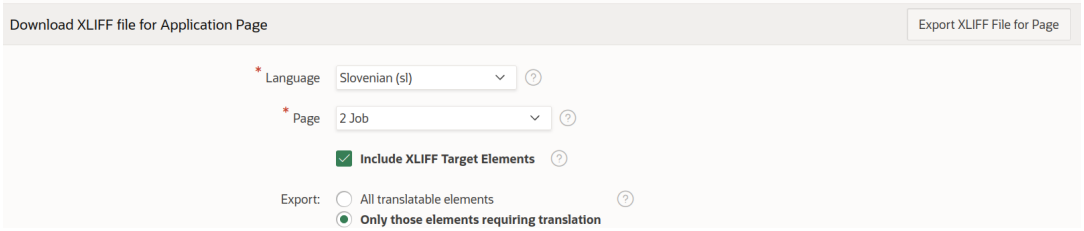
- navigirajte do Shared components > Globalization > Application Translations > Download XLIFF translation files; možemo odabrati preuzimanje svih prevodivih elemenata ili samo onih elemenata koji zahtijevaju prijevod. Primjer na Slici 12.26 pokazuje izvoz stranice 2, Slovenski jezik i elementi koji zahtijevaju prijevod.
- uredite izvezenu datoteku u željenom editoru (Notepad++, Kate, Sublime itd.). U svrhu demonstracije promijenili smo samo nekoliko nizova s oznakom "target": u redovima 48, 52, 56, 60, 72 i 76 (vidi Sliku 12.27). Spremite promjene.
- navigirajte do Shared components > Globalization > Translate application > Apply XLDIFF translation files > Upload Files. Odaberite datoteku (vidi Sliku 12.28) i odaberite Upload. Odaberite jezik u "Apply to Translation" (vidi Sliku 12.29). Odaberite Apply checked. Odaberite Publish.
- navigirajte do Shared components > Globalization > Translate application > Publish translations. Odaberite sve jezike koje ste preveli i idite na Publish (vidi Sliku 12.30). Pričekajte dok ne dobijete poruku da je aplikacija uspješno objavljena.
- navigirajte do Shared components > Globalization > Globalization attributes. Promijeniti Application Primary Language na novi prijevod (vidi Sliku 12.31). Odaberite Apply changes.



Slika 12.24: Definirani jezici za prijevod



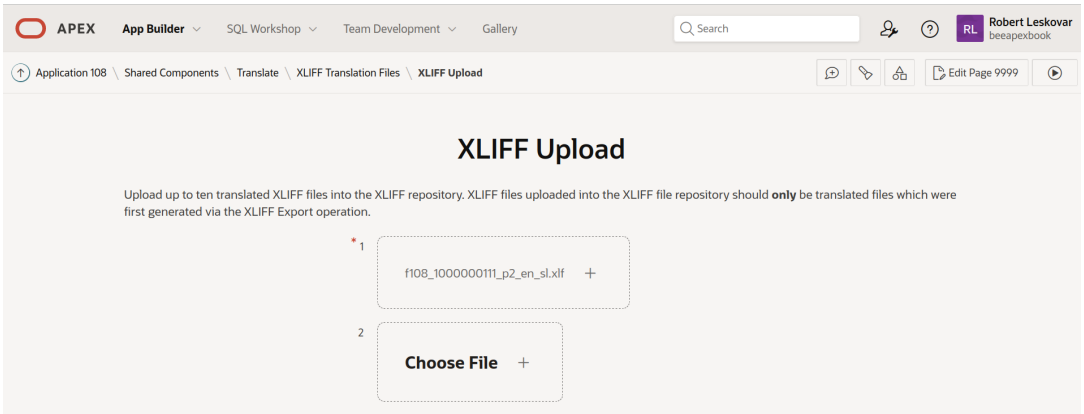
Slika 12.25: Izvorni tekst koji se može prevesti



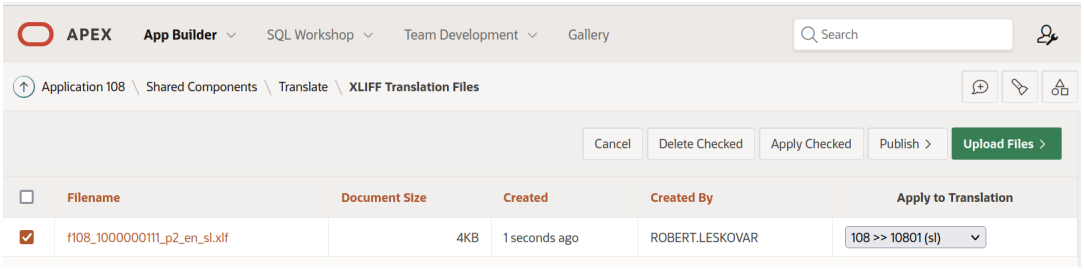
Slika 12.26: Izvoz nizova za određeni jezik i stranicu

```
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>
```

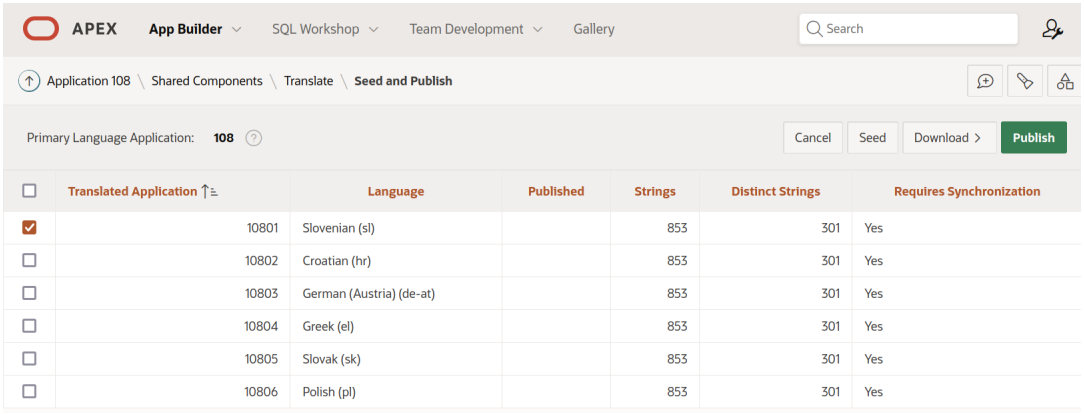
Slika 12.27: Prijevod "target" označeni nizovi u recima 48, 52, 56, 60, 72 i 76



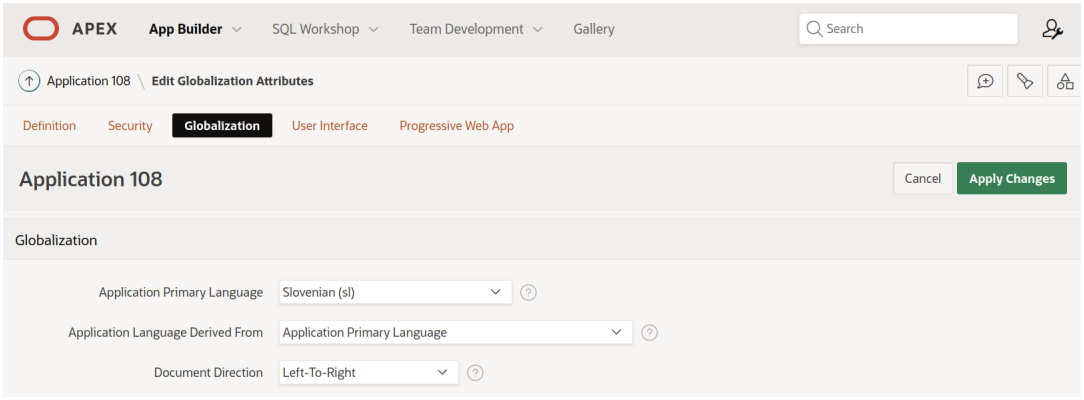
Slika 12.28: Prijenos XLDIFF datoteka prijevoda



Slika 12.29: Primjena promjena i objavljivanje.

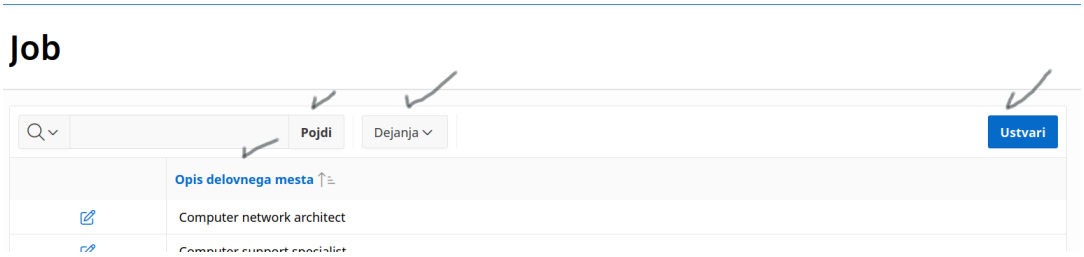


Slika 12.30: Konačna objava prijevoda aplikacije



Slika 12.31: Postavljanje primarnog jezika aplikacije

Sada provjerite stranicu 2 u prevedenoj aplikaciji (vidi Sliku 12.32).



Slika 12.32: Prevedena stranica

Zaključne napomene Razvoj se obično odvija u testnom okruženju. Prilikom migracije aplikacije u novi radni prostor, pakirane aplikacije su vrlo prikladne jer skraćuju vrijeme za migraciju. Iz sigurnosnih razloga se eventualni korisnici i njihove uloge ne migriraju. Autorizacija korisnika u novom radnom prostoru je ozbiljan zadatak. Ovo poglavlje također pruža osnovne upute za višejezične aplikacije. Treba napomenuti da za velike aplikacije proces prevođenja mora biti podržan od strane alata za automatsko prevođenje, kao i stručnjaka za jezik i kulturu.

12.5 Dodatni materijal za učenje

Možete pronaći sljedeći dodatni materijal za učenje:

- eksportirane aplikacije
- video upute

Sav dodatni materijal za učenje dostupan je na public BeeAPEX project page. Prijavite se kao gost (lozinka nije potrebna). Pronađite udžbenik u Books section, skripte u mapi Part 1 > Chapter12 u odjeljku Scripts i video u Collection of video guides. Materijal za kratke tečajeve u sekciji Short courses.

12.5.1 Izvezene aplikacije

Postoje tri povezane aplikacije:

- početna aplikacija. Provjerite postoje li tablice koje počinju s CH12. Ako tablice postoje, možete ih ispustiti pomoću skripte CH12DROP. Zatim uvezite aplikaciju u datoteku CH12_Application_initial.sql. Nakon uvoza aplikacije morate definirati tablice. Izvršite skriptu CH12CREATE i po želji popunite tablice skriptom CH12INSERT. Idite na uvezenu aplikaciju > Shared Components > Application Access Control. Dodajte User Role Assignment za trenutno prijavljenog korisnika. Dopusti sve radnje s ulogom Administrator, odabir, umetanje, ažuriranje i brisanje s ulogom Contributor i samo odabir s ulogom Reader.
- pakirana aplikacija. Provjerite postoje li tablice koje počinju s CH12. Ako tablice postoje, morate ih sa CH12DROP skriptom. Zatim uvezite aplikaciju u datoteku CH12_Application_packaged.sql. Idite na uvezenu aplikaciju > Shared Components > Application Access Control. Dodajte User Role Assignment za trenutno logiranog korisnika. Dopustite sve radnje s ulogom Administrator, odabir, umetanje, ažuriranje i brisanje s ulogom Contributor i samo odabir s ulogom Reader.
- pakirana višejezična aplikacija. Provjerite postoje li tablice koje počinju s CH12. Ako tablice postoje, morate ih ispustiti s CH12DROP skriptom. Zatim uvezite aplikaciju u datoteku CH12_Application_packaged.sql. Navigirajte na uvezenu aplikaciju > Shared Components > Application Access Control. Add User Role Assignment za trenutno logiranog korisnika. Dopustite sve radnje s ulogom Administrator, odabir, umetanje, ažuriranje i brisanje s ulogom Contributor i samo odabir s ulogom Reader.

12.5.2 Video upute

Dostavljene su sljedeće video upute:

- kreirati početnu aplikaciju,
- uvoziti početnu aplikaciju,
- kopirati početnu aplikaciju do zapakirane i kreiranje zapakirane aplikacije,
- uvoziti zapakiranu aplikaciju,
- uvoziti zapakiranu višejezičnu aplikaciju.

12.6 Pitanja

1. Što je pakirana aplikacija i koja joj je prednost?

2. Kako izvesti pakiranu aplikaciju koja uključuje definiciju objekata baze podataka i podatke za tablice?
3. Koja je uloga XLDIFF datoteke u kreiranju višejezične aplikacije?

12.7 Odgovori

1. Pakirana aplikacija unaprijed je izrađena aplikacija koja se može instalirati i konfigurirati u Oracle APEX radnom prostoru. Glavna prednost pakirane aplikacije je laka i glatka migracija, kao i trenutna dostupnost pripremljenih podataka.
2. Pakirana aplikacija se izvozi pomoću čarobnjaka. Mogu se uključiti skripte koje definiraju objekte baze podataka (poput tablica) i upute za umetanje podataka u tablice. Popis skripti i redoslijed izvršavanja tih skripti postavlja se u čarobnjaku. Pakirana aplikacija također može sadržavati skriptu za deinstalaciju.
3. XLDIFF datoteka je označena i omogućuje jednostavan pristup ciljnim vrijednostima za nizove prijevoda. Ove datoteke mogu sadržavati cijelu aplikaciju ili samo određenu stranicu na odabranom jeziku. Datoteka s prevedenim ciljnim nizovima se uvozi u aplikaciju. Repozitorij prijevoda mora biti objavljen kako bi bio dostupan krajnjem korisniku.

Razvoj aplikacija u APEX-u

13 Intranet novosti za zaposlenike 225
R. Leskovar, U. Rajković, A.Baggia . . . . . .

14 GreenDi - Katalog biljaka 237
Vjeran Strahonja, Dijana Oreški, Darko Andročec, Ana Kutnjak . . . . . .

15 GreenDi - Autorizacija i upravljanje korisnicima 245
Vjeran Strahonja, Darko Andročec, Ana Kutnjak, Larisa Hrustek . . . . . .

16 inovativni sustavi 251
R. Leskovar, U. Rajković, A.Baggia . . . . . .

17 Upravljanje poslovnim procesima . . . 261
R. Leskovar, U. Rajković, A.Baggia . . . . . .

18 GreenDi – Upravljanje katalogom biljaka 291
Vjeran Strahonja, Dijana Oreski, Darko Androcec, Ana Kutnjak . . . . . .

19 Sustav za upravljanje recenzijama knjiga 297
A. Kutnjak, L. Hrustek, A. Baggia, R.Leskovar . . . . . .

20 Sastavnica i obračun troškova 309
R. Leskovar, U. Rajković, A.Baggia . . . . . .

21 Prehrana i upravljanje dijetom 325
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . .

22 Raspored konzultacija 345
J. Mańko, M. Soňta, R. Leskovar . . . . . .

23 Telco studija slučaja 365
Veronika Šalgová, Jozef Kostolný, Michal Mrena, Michal Kvet, Miroslav Potočár . . . . . .

24 Studija slučaja iznajmljivanja automobila 389
A. Angeioplastis, G. Myllis, A. Tsimpiris, D. Varsamis . .



13. Intranet novosti za zaposlenike

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA

13.1 Poslovni pogled na slučaj

Tvrtka srednje veličine, sa sedam stotina zaposlenih, etablirani je igrač u globalnoj industriji vrhunske opreme za sport i slobodno vrijeme te naprednih kompozita. Sjedište je u Sloveniji, a podružnice u SAD-u, Kanadi, Njemačkoj i Japanu. Njihov godišnji prihod je oko 90 milijuna eura. Većina zaposlenih nalazi se u jednom mjestu u Sloveniji. Tvrtka je imala četiri divizije i svaka ima nekoliko odjela. U prošlosti su internim polugodišnjim biltenom predstavili razna postignuća, priče o uspjehu, najave novih proizvoda i slično. Zatim, izradili su statičnu (čisti HTML) intranet stranicu za širenje važnih vijesti. No ova prilika za osnaživanje zaposlenika i jačanje njihove predanosti tvrtke bila je kratkog vijeka. Idilični vrt pretvorio se u problem.

13.2 Definicija problema

Proces objavljivanja bio je labavo definiran. Zahtjev za objavu vijesti, informatički djelatnik primio je od izdavatelja – voditelja odjela i odjela. Zahtjev je poslan poštom, telefonom ili osobno, često s oskudnim materijalom (tekst, slike, video). Datum početka i završetka vijesti nije određen. Učestalost zahtjeva za objavom vijesti porasla je na otprilike stotinu vijesti mjesečno, a neki su se zahtjevi preklapali (dva ili više voditelja odjela imali su slične ili iste teme). IT administrator morao je riješiti probleme između menadžera i s menadžerima. Datumi početka i završetka pojavljivanja vijesti često su bili prepušteni IT administratorima što je uzrokovalo loše raspoloženje kod menadžera i IT zaposlenika. Također, upravljanje statičnom stranicom postalo je gubitak vremena i izuzetno podložno pogreškama. Rješenje je zahtijevalo oboje, modifikaciju procesa i novu platformu za objavljivanje.

Modifikacija procesa i nova platforma za objavljivanje Okruženje za razvoj aplikacija s niskim kodom odabrano je za razvoj nove platforme za objavljivanje. Tvrtka već koristi Oracle bazu podataka, stoga je Oracle APEX odabran kao razvojno okruženje. Identificirane su tri vrste korisnika platforme. Prvo administrator, koji ima pristup svim podacima i privilegije za upravljanje korisnicima i ulogama. Osim osnovnih podataka o zaposlenicima, pratimo i njihova zaposlenja u odjelima i njihove lokacije. Svaki zaposlenik ima određenu definiranu ulogu, koja se s vremenom može mijenjati. Druga vrsta korisnika je izdavač, koji nema pravo upravljanja podacima o zaposlenicima i definiranja privilegija, ali ima privilegiju objavljivanja vijesti. Treća vrsta korisnika je gledatelj/čitatelj vijesti, koji ima privilegiju čitati internetske vijesti.

Specifikacija zahtjeva Temeljem modifikacije procesa, platforma će korisnicima omogućiti:

- gledatelj/čitatelj: prijavu, pristup nadzornoj ploči, pregled vijesti i priloga vijesti,
- izdavač: prijava, pristup nadzornoj ploči, pregledavanje i objavu vijesti i dodavanje privitaka kod objave vijesti,
- administrator: prijava, pristup nadzornoj ploči, pregled vijesti i priloga vijesti, pregled i upravljanje odjelima, ulogama, zaposlenicima, ulogama zaposlenika, upravljanje primjenom.

13.3 Slučajevi korištenja

13.3.1 Narativni opis slučaja korištenja

Platforma za objavljivanje omogućuje pristup trima različitim vrstama zaposlenika: administratoru, čitatelju/gledatelju i suradniku/izdavaču, od kojih svaki ima različite privilegije na intranet portalu. Izdavač može objavljivati vijesti, čitatelj može čitati vijesti, a administrator može upravljati platformom. Svaki od ovih slučajeva korištenja zahtijeva da se korisnik prijavi na intranetsku platformu. Pogledajte tablice 13.1, 13.2 i 13.3,

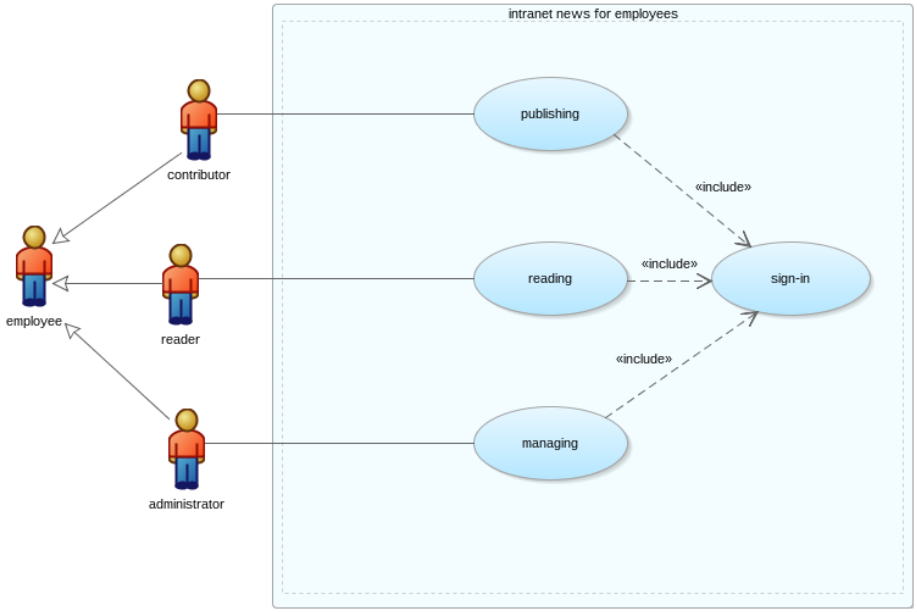
13.3.2 Polustrukturirani opis

Možemo sažeti tri različite korisničke priče ili slučajeva korištenja: objavljivanje, čitanje i upravljanje.

Dodavanje privitka vijesti zapravo je proširenje gore opisanog UC-a i mogli bismo ga opisati kao novi slučaj upotrebe. Radi čitljivosti nastavljamo s druga dva glavna slučaja upotrebe, no aplikacija će imati značajku za dodavanje privitaka vijestima.

13.3.3 Dijagram slučaja korištenja

Ovaj je primjer prikazan na dijagramu slučaja korištenja Figure 13.1.



Slika 13.1: Dijagram slučaja korištenja

Opet radi jednostavnosti proširenja slučaja korištenja (dodavanje privitka vijesti) je preskočen Figure 13.1.

Tablica 13.1: Opis slučaja korištenja: Objavljivanje intranet novosti

Ključna riječ	**Vrijednost**
ID:	*Ch13-01*
Naslov:	*Objavljivanje novosti na intranet portalu*
Opis:	*Odgovorna osoba iz Odjela upravljanja ljudskim resursima, s definiranom ulogom izdavača, koristi intranet APEX portal za objavu vijesti. Svaka vijest ima datum početka i završetka. Zadani datum početka je dan objave.*
Primarna uloga:	*Zaposlenik s ulogom objavljivanja*
Preduvjeti:	*Zaposlenik mora biti naveden u tablici zaposlenika, a korisnik mora biti dodan ulozi izdavača CH13. Pristup web aplikaciji mora biti omogućen.*
Postuvjeti:	*Nakon uspješne objave novosti, vijest i njeni prilozi dostupni su ostalim korisnicima intranet portala.*
Glavni	*Scenariji*
Uspješan scenarij:	1. *Otvoren web preglednik i prijava na intranet aplikaciju.* 2. *Selecija opcije Objava novosti..* 3. *Unos naslova vijesti, opisa i datuma završetka.* 4. *Potvrda kreiranja novosti.* 5. *Dodavanje priloga.* 6. *Unos detalja o prilogu.* 7. *Potvrda dodavanja priloga.* 8. *Revizija objavljene novosti.*
Ekstenzije:	• *1a. Prijava nije uspjela.* • *1a* Prošireno:* • *1a1. Prikazivanje poruke o pogrešci.* • *1a2. Otvaranje prozora za prijavu.* • *4a. Pogreška tipa podataka.* • *4a* Prošireno:* • *4a1. Prikazivanje poruke o pogrešci.* • *7a. Pogreška tipa podataka.* • *7a* Prošireno:* • *7a1. Prikazivanje poruke o pogrešci.*
Frekvencija korištenja:	*Izdavači godišnje objave oko 1000 vijesti, prosječno 5 dnevno*
Status:	*Završeno*
Vlasnik:	*Zaposlenik s ulogom objavljivanja*
Prioritet:	*Umjereno*

Tablica 13.2: Opis slučaja korištenja: Čitanje intranet novosti

Ključna riječ	**Vrijednost**
ID:	*Ch13-02*
Naslov:	*Čitanje novosti na intranet portalu*
Opis:	*Intranet korisnik čita novosti*
Primarna uloga:	*Intranet korisnik*
Preduvjeti:	*Zaposlenik mora biti naveden u tablici zaposlenika, a korisnik mora biti dodan u ulogu CH13 Preglednik. Pristup web aplikaciji mora biti omogućen.*
Postuvjeti:	-
Glavni	*Scenariji*
Uspješan scenarij:	1. *Otvoren web preglednik i prijava na intranet aplikaciju* 2. *Odabir opcije Vidi novosti* 3. *Odabir opcije Preuzimanje priloga za pregled*
Ekstenzije:	• *1a. Prijava nije uspjela* • *1a* Prošireno:* • *1a1. Prikazivanje poruke pogreške*
Frekvencija korištenja:	*Zaposlenici svakodnevno čitaju vijesti.*
Status:	*Završeno*
Vlasnik:	*Intranet korisnik*
Proritet:	*Visok*

Tablica 13.3: Opis slučaja korištenja: Upravljanje intranet portalom.

Ključna riječ	**Vrijednost**
ID:	*Ch13-03*
Naslov:	*Upravljanje intranet portalom*
Opis:	*Upravljanje intranet portalom*
Primarna uloga:	*Intranet portal administrator*
Preduvjeti:	*Zaposlenik mora biti naveden u tablici zaposlenika, a korisnik mora biti dodan u ulogu administratora. Pristup web aplikaciji mora biti omogućen.*
Postouvjeti:	*Korisnici, uloge i zaposlenici su spremni za korištenje intranet portala.*
Glavni	*Scenariji*
Uspješan scenarij:	1. *Otvaranje web preglednika i prijava na intranet aplikaciju* 2. *Dodavanje i upravljanje podacima o odjelima* 3. *Dodavanje i upravljanje podacima o ulogama* 4. *Dodavanje i upravljanje zaposlenicima i njihovim ulogama* 5. *Objava novosti* 6. *Pregled novosti* 7. *Administracija*
Ekstenzije:	• *1a. Prijava nije uspjela* • *1a* Prošireno:* • *1a1. Prikazivanje poruke o pogrešci* • *1a2. Otvaranje prozora za prijavu* • *2a. Podatkovni tip pogreške* • *2a* Prošireno:* • *2a1. Prikazivanje poruke o pogrešci* • *3a. Podatkovni tip pogreške* • *3a* Prošireno:* • *3a1. Prikazivanje poruke o pogrešci* • *4a. Podatkovni tip pogreške* • *4a* Prošireno:* • *4a1. Prikazivanje poruke o pogrešci* • *5a. Podatkovni tip pogreške* • *5a* Prošireno:* • *5a1. Prikazivanje poruke o pogrešci*
Frekvencija korištenja:	*Učestalost korištenja ovisi o novim zaposlenicima i promjenama u shemi povlastica. Otprilike 5 puta tjedno.*
Status:	*Završeno*
Vlasnik:	*Intranet portal administrator*
Prioritet:	*Visok*

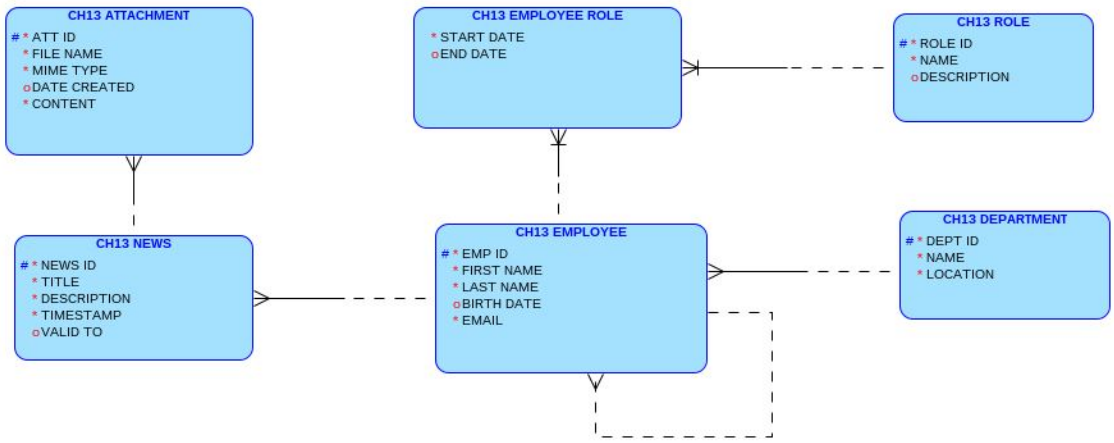
13.4 Podatkovni model

13.4.1 Narativni opis podatkovnog modela

Postoji šest entiteta u logičkom podatkovnom modelu. Entitet CH13 ODJEL ima tri atributa: ID, naziv i lokaciju odjela. Svaki odjel može imati više zaposlenika. U entitetu CH13 ZAPOSLENIK pohranjuju se osnovni podaci o zaposlenicima: ID, ime i prezime, datum rođenja i email. Zaposlenik može biti menadžer. Svaki zaposlenik ima samo jednog voditelja, dok voditelj može voditi više zaposlenika. Svaki zaposlenik može imati svoju ulogu. Uloge su definirane ID-om, imenom i opisom. Za svakog zaposlenika i svaku ulogu definiran je datum početka i završetka. Tijekom vremena, zaposlenik može imati više uloga, također se može dodijeliti više zaposlenika jednoj ulozi. Ipak, jedna definicija uloge zaposlenika ne može se prenijeti na drugog zaposlenika ili ulogu. Zaposlenik može objaviti više vijesti tijekom vremena. Svaka vijest identificirana je ID-om, naslovom, opisom, datumom početka i datumom završetka. Svaka vijest može imati više priloga. Prilog je identificiran ID-om, nazivom datoteke, vrstom, datumom izrade i sadržajem.

13.4.2 Logički model podataka

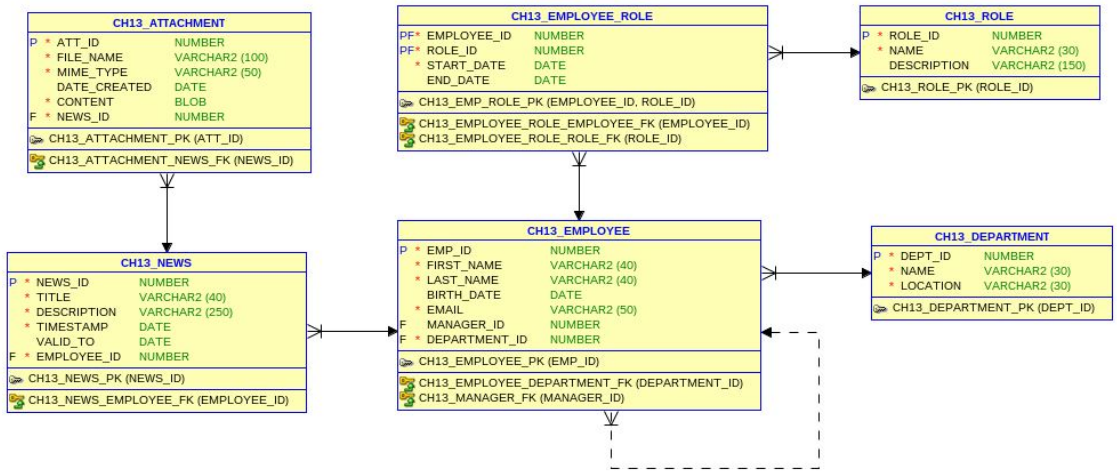
Ovaj je primjer prikazan na logičkom modelu podataka Figure 13.2.



Slika 13.2: Logički model podataka

13.4.3 Relacijski model podataka

Automatska transformacija iz logičkog podatkovnog modela u relacijski podatkovni model u Oracle SQL Data Modeleru omogućuje funkcija *Engineering to relational*. Rezultat je prikazan na Figure 13.3.



Slika 13.3: Logički model podataka

Oracle SQL Data Modeler također generira SQL skriptu za stvaranje tablice, niza i okidača. Odaberite sve tablice na relacijskom modelu i upotrijebite funkciju *File* > *Export* > *DDL File* da biste dobili skriptu poput ove:

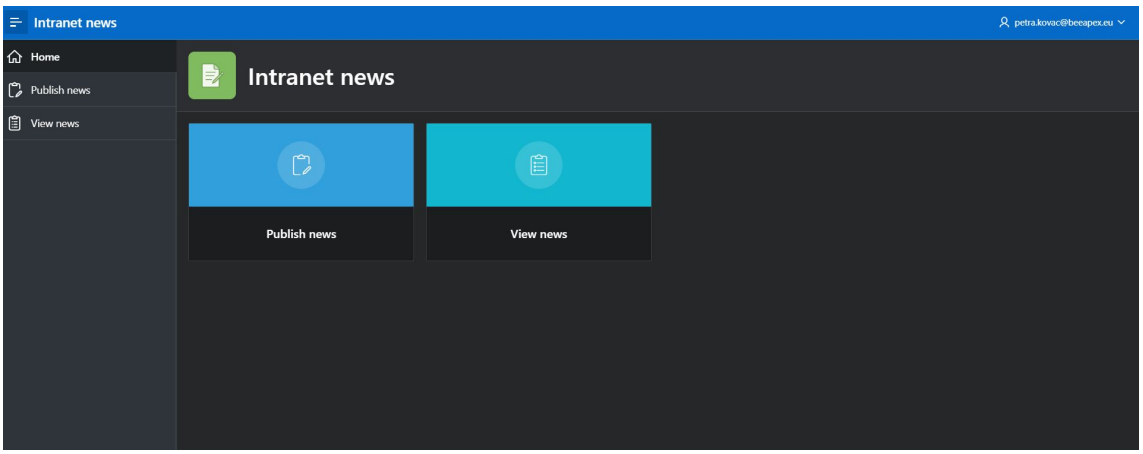
```
CREATE TABLE CH13_NEWS (
NEWS_ID NUMBER NOT NULL
...
);

CREATE TABLE CH13_DEPARTMENT (
DEPT_ID NUMBER NOT NULL
...
);
```

Sada se može generirati tablica u Oracle bazi podataka. Moguće je uvesti generiranu skriptu i izvršiti je u APEX-u.

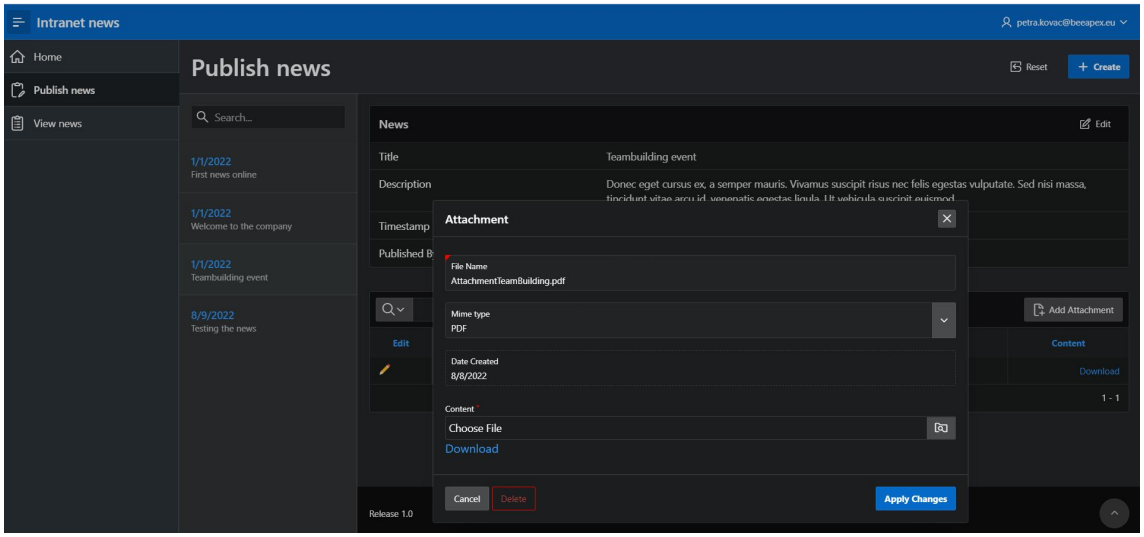
13.5 Sučelja aplikacije

Na sljedećim slikama prikazana su sučelja aplikacije za sve tri uloge. Prvo je prikazano sučelje za Izdavača na sljedećoj sliciFigure 13.4.



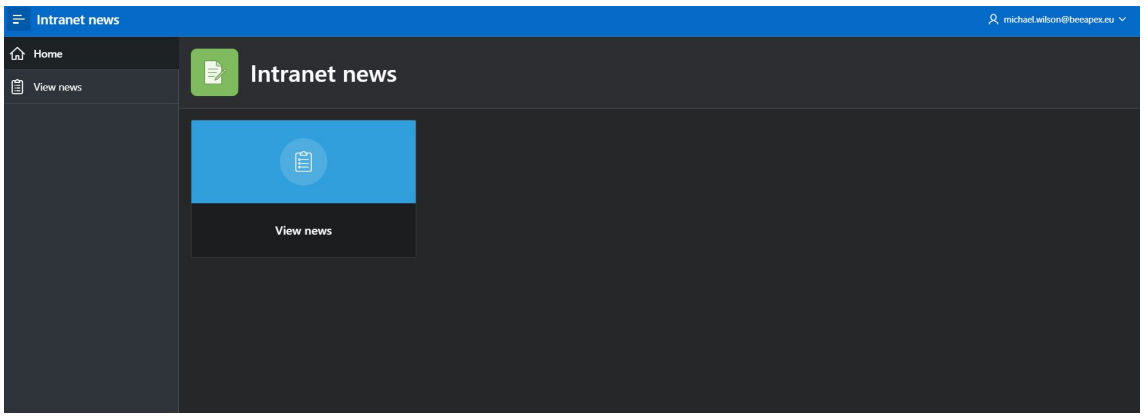
Slika 13.4: Sučelje Izdavača

Stranica za objavu vijesti s prilozima i njihovim detaljima predstavljena je na Figure 13.5.



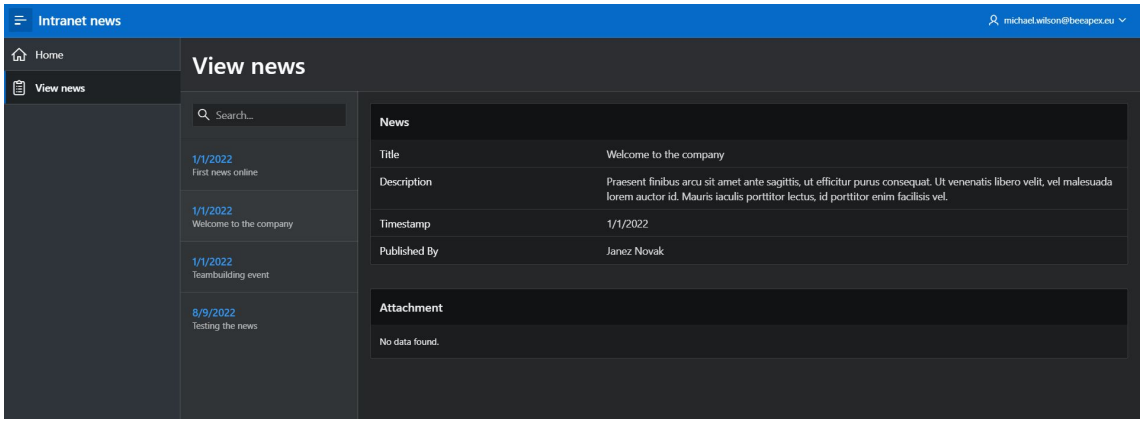
Slika 13.5: Objavljivanje novosti s prilozima

Čitatelj ima ograničenu ulogu na portalu. Za sada su dostupne samo novosti Početna i Prikaz, kao što je prikazano na Figure 13.6.



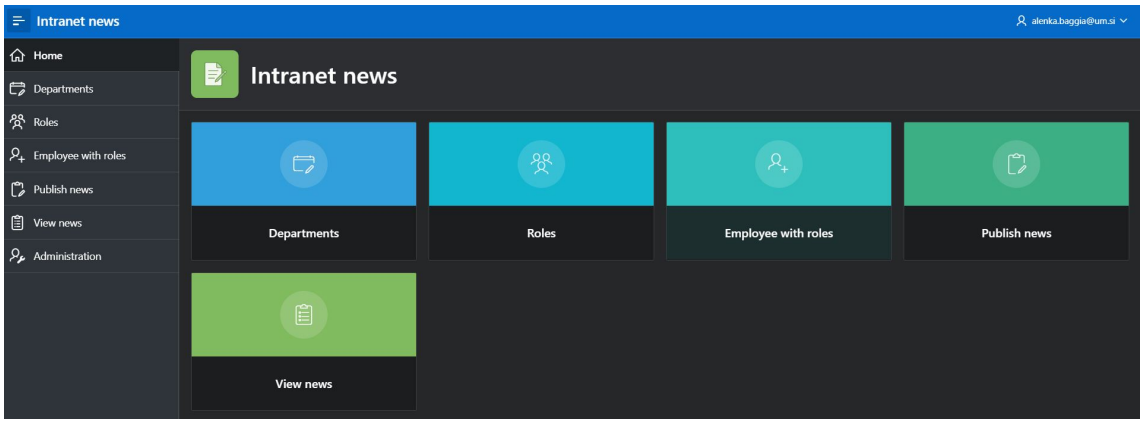
Slika 13.6: Sučelje Čitatelja

Glavna značajka Čitatelja je pristup vijestima (vidi Figure 13.7).



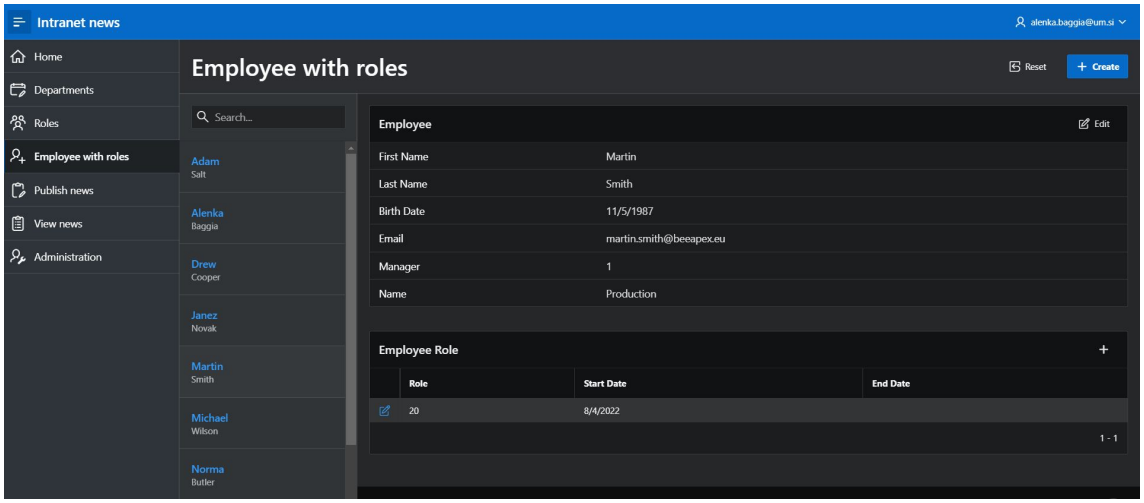
Slika 13.7: Čitatelj pristupa novostima

Uz pregled i objavljivanje vijesti, Administrator može upravljati odjelima, ulogama i zaposlenicima te podacima o ulogama, kao što je prikazano na Figure 13.8. Također, Administratoru je dostupna stranica na koja omogućuje upravljanje korisnicima.



Slika 13.8: Sučelje Administratora

Stranica za uređivanje podataka o ulogama zaposlenika prikazana je naFigure 13.9. Ova stranica je dostupna samo Administratoru intranet portala.



Slika 13.9: Uređivanje uloge zaposlenika od strane Administratora

13.6 Dodatni materijal za učenje

Sav dodatni materijal za učenje dostupan je na:

- izvezena aplikacija,
- skripte za stvaranje, ispuštanje i umetanje,
- video vodič.

Sav dodatni materijal za učenje dostupan je na:public BeeAPEX project page. Prijavite se kao gost (lozinka nije potrebna). Navigirajte do sekcije Books, skripte u mapi Part 2 > Chapter13 u sekciji Scripts i video upute u Collection of video guides. Materijal za kratke tečajeve dostupan je u Short courses section.

13.6.1 Izvezena aplikacija

Izvezena aplikacija je zapakirana. Instalacija kreira tablice i popunjava podatke. Deinstalacija uklanja sve objekte baze podataka koji se koriste u ovoj aplikaciji.

13.6.2 Video upute

Video upute prikazuju svaki korak u razvoju aplikacije.

13.7 Pitanja

1. Kako biste promijenili logički model podataka ako želite implementirati sljedeći zahtjev: jedan privitak može pripadati većem broju vijesti.
2. Kako bi se promijenio relacijski podatkovni model ako bi se implementirao zahtjev (jedan privitak može pripadati više vijesti)?
3. Koje su korisničke uloge implementirane u 13 aplikaciji i zašto?

13.8 Odgovori

1. Veza jedan prema više trebala bi biti povezana u smjeru od privitka do vijesti.
2. Par prilog-vijest u logičkom modelu transformirao bi se u tri relacijske tablice: prilog, vijest i prilog_novosti. Posljednji bi sadržavao najmanje dva polja (ID privitka i vijesti) koja bi služila kao vanjski ključevi.
3. U Poglavlju 13 tImplementirane su sljedeće korisničke uloge: Administrator, Suradnik i Čitatelj. Jesu li ove uloge implementirane kako bi omogućile ovlaštenim korisnicima da rade

svoj posao i spriječile neovlaštene korisnike da izvode neodobrene radnje?



14. GreenDi - Katalog biljaka

VJERAN STRAHONJA, DIJANA OREŠKI, DARKO ANDROČEC, ANA KUTNJAK

14.1 Poslovni pogled na slučaj

Grupa entuzijasta okupljena u neprofitnu organizaciju GreenDi pokreće i provodi razne inicijative vezane uz urbano vrtlarstvo, zdravu prehranu, uzgoj starih domicilnih sorti, suradnju s lokalnim poljoprivrednicima i slično. Do sada su u svom radu koristili društvene mreže. Zato su odlučili pokrenuti vlastitu GreenDi, peer-to-peer (istorazinsku) platformu za dijeljenje sjemena i biljaka, s ciljem promicanja bioraznolikosti i olakšavanja prošlih aktivnosti te omogućavanja novih. Kako bi se razvila i implementirala takva platforma, potrebno je izraditi specifikaciju zahtjeva i osnovnih modela, uključujući podatkovni model i slučajeve korištenja.

14.2 Definicija problema

Otvoreni dio platforme ima sljedeće karakteristike:

- GreenDi korisnici nisu ograničeni na geografski položaj i mogu biti globalni. No, budući da se radi o biljnoj razmjeni, očekuje se formiranje lokalnih mreža vezanih za određeno zemljopisno područje.
- Platforma je podijeljena u nekoliko tematskih cjelina kao što su tradicionalne i autohtone sorte i sjemenje (povrće, voće, ljekovito bilje, sobno i ukrasno bilje), s idejom proširenja na industrijsko i drugo bilje.
- Osnova otvorenog dijela je katalog biljaka, s nazivom / nazivima na jeziku korisnika, latinskom i engleskom, klasifikacijom biljaka (taksonomijom), opisom, staništima, uzgojem i korištenjem, fotografijama i drugim osnovnim podacima.
- Osnovne informacije o biljkama otvorene su za pretraživanje i pregled bez registracije.
- GreenDi katalog biljaka povezan je s nekoliko otvorenih web stranica, bazama podataka i vanjskih servisa koji sadrže informacije vezane uz biljke. Koriste se URL adrese, hiperveze i druge vrste označenih podataka.
- U katalogu se nalaze biljke koje se razmjenjuju na GreenDi platformi, ali i one koje još nisu u razmjeni ili u ponudi. Katalog također ima poveznice na komercijalne dobavljače, s kojima članovi imaju dobra iskustva i koji njeguju organski uzgoj.

Upravljanje katalogom opisano je sljedećim rečenicama:

- Katalogom biljaka upravljaju administratori GreenDi platforme.
- Upravljanje katalogom uključuje unos i promjenu podataka o biljkama.

- Neke funkcionalnosti upravljanja katalogom dostupne su korisnicima ovlaštenim od strane administratora (ovlašteni korisnik).
- Napredne funkcionalnosti, dostupne administratorima, uključuju poveznice na druge stranice koje sadrže informacije o biljkama, videe o uzgoju, korištenju biljaka itd.
- Svi registrirani korisnici (članovi) mogu po želji podijeliti svoje mišljenje o biljkama, napisati recenziju, dodati još neke informacije i sl., ali to nije dio ove funkcionalnosti.

14.3 Slučajevi korištenja

14.3.1 Narativni opis slučaja korištenja

Otvoreni dio GreenDi platforme omogućuje svakom korisniku uvid u podatke kataloga bilja s bilo kojeg mjesta, bez registracije. Istovremeno, biljke su razvrstane u kategorije koje korisniku omogućuju lakšu navigaciju. Za svaku biljku definiran je naziv, taksonomija, opis, stanište, uzgoj i uporaba, fotografije i drugi osnovni podaci, kao i poveznice na vanjske izvore podataka.

14.3.2 Polustrukturirani opis

Tablica 14.1 prikazuje slučaj korištenja.

14.3.3 Dijagram slučaja korištenja

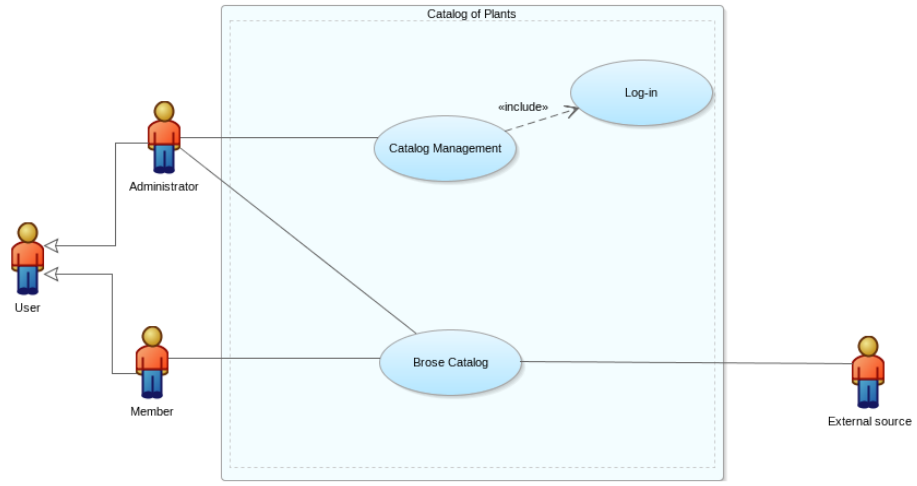
Ovaj je primjer prikazan na dijagramu slučaja korištenja 14.1. Na desnoj strani dijagrama slučaja korištenja nalazi se akter vanjski izvor (External Source). Ovaj akter nije predstavljen stereotipom osobe jer se radi o vanjskom sustavu iz kojeg GreenDi katalog biljaka dohvaća informacije o biljkama, sjemenkama i slično. Mnogo je primjera takvih baza podataka o biljkama koje nude otvorene usluge ili otvorene skupove podataka. To znači da su baza ili podaci slobodno dostupni za korištenje, bez ikakvih ograničenja pristupa ili korištenja. Neki primjeri online baza podataka o biljkama koje pružaju otvorene usluge ili otvorene podatke uključuju:

1. The Global Biodiversity Information Facility (GBIF, https://www.gbif.org) omogućuje slobodan i otvoren pristup podacima o bioraznolikosti, uključujući podatke o biljkama. GBIF omogućuje korisnicima pretraživanje i pristup podacima iz niza izvora, uključujući herbarije, muzeje i istraživačke ustanove. Odnosi se na The International Plant Names Index (IPNI), skup podataka naziva i povezanih osnovnih bibliografskih detalja sjemena, biljaka, paprati i likofita (https://www.gbif.org/dataset/046bbc50-cae2-47ff-aa43-729fbf53f7c5).
2. WFO Plant List (https://wfoplantlist.org/plant-list) pokrenut je u svibnju 2021. godine i pruža jednostavan statički popis svih biljnih vrsta koji se može citirati. Skup podataka WFO Plant List može se preuzeti u cjelosti. Podacima se također može pristupiti putem WFO Plant List API-ja.
3. The Flora of Italy skup podataka može se preuzeti s web stranice Sveučilišta u Rimu i uključuje slike visoke razlučivosti preko 10 000 biljnih vrsta u Italiji.

Ovo je samo nekoliko primjera dostupnih skupova podataka koji uključuju slike biljaka i sjemena u Europi, ali postoje mnogi drugi dostupni skupovi podataka koji pružaju slične informacije. U ovom primjeru nećemo ostvariti komunikaciju s vanjskim izvorom jer zahtijeva naprednije znanje.

Tablica 14.1: Opis slučaja korištenja: pregled kataloga biljaka

Ključna riječ	**Vrijednost**
ID:	*ch14-01*
Naslov:	*Pregled kataloga biljaka*
Opis:	*Katalog biljaka javno je dostupan svim zainteresiranima. Aplikacija omogućuje pretraživanje filterima po svim poljima baze podataka.*
Primarna uloga:	*Bilo koja osoba*
Preduvjeti:	*Osnovne informacije o biljkama otvorene su za pretraživanje i pregled bez registracije.*
Postuvjeti:	*Ako se iz drugog (proširenog) slučaja korištenja pozove pregled kataloga biljaka, prenosi se ID odabrane biljke.*
Glavni	*Katalog biljaka*
Uspješan scenarij:	*Prvi scenarij, katalog biljaka:* 1. *Svaki korisnik na početnoj stranici otvara obrazac za pregled kataloga biljaka, koji je obrazac za pretraživanje, a temelji se na principu upita kroz primjer (QBE).* 2. *Pretraživanje je moguće prema sljedećim podacima: a) tematska cjelina - popis šifri (sve zadane ili specifične za neki naziv, npr. povrće, voće, ljekovito bilje, sobno i ukrasno bilje itd.); b) vrsta postrojenja - hijerarhijski organizirana klasifikacija postrojenja (prema zadanim postavkama ili specifičnom nazivu), c) naziv (ime) na jeziku korisnika, d) latinski naziv (e) engleski naziv.* 3. *Rezultat pretraživanja prikazuje se u obliku tablice.* 4. *Tablični obrazac moguće je sortirati po stupcima.* 5. *Odabirom retka moguće je otvoriti obrazac s detaljima obrasca biljke koji sadrži: vrstu biljke, tematsku cjelinu, lokalne nazive, engleske nazive, latinske nazive, opis, stanište, uzgoj, uporabu, fotografije, hiperveze i status.* *Drugi scenarij, upravljanje katalogom:* 1. *Upravljanje katalogom može obavljati aktivni korisnik u ulozi administratora, pa je prvi korak procedura prijave (UC: Log In) gdje se provjerava ispunjavanje ovih uvjeta.* 2. *Upravljanje katalogom uključuje promjenu ili dodavanje informacija o biljci koja je već u katalogu, brisanje te biljke (zapravo skrivanje jer biljka nije fizički izbrisana) ili dodavanje nove biljke.* 3. *Korisnik potvrđuje ili odbacuje promjenu.*
Ekstenzije:	• *Obrazac za pretraživanje (QBE)* • *Tablični obrazac* • *Pojedinosti obrasca biljaka*
Frekvencija korištenja:	*Cca. maksimalno 1000 pretraga u minuti.*
Status:	*Status razvoja*
Vlasnik:	*javni, anonimni korisnik*
Prioritet:	*visoki*



Slika 14.1: Dijagram slučaja korištenja - Katalog biljaka

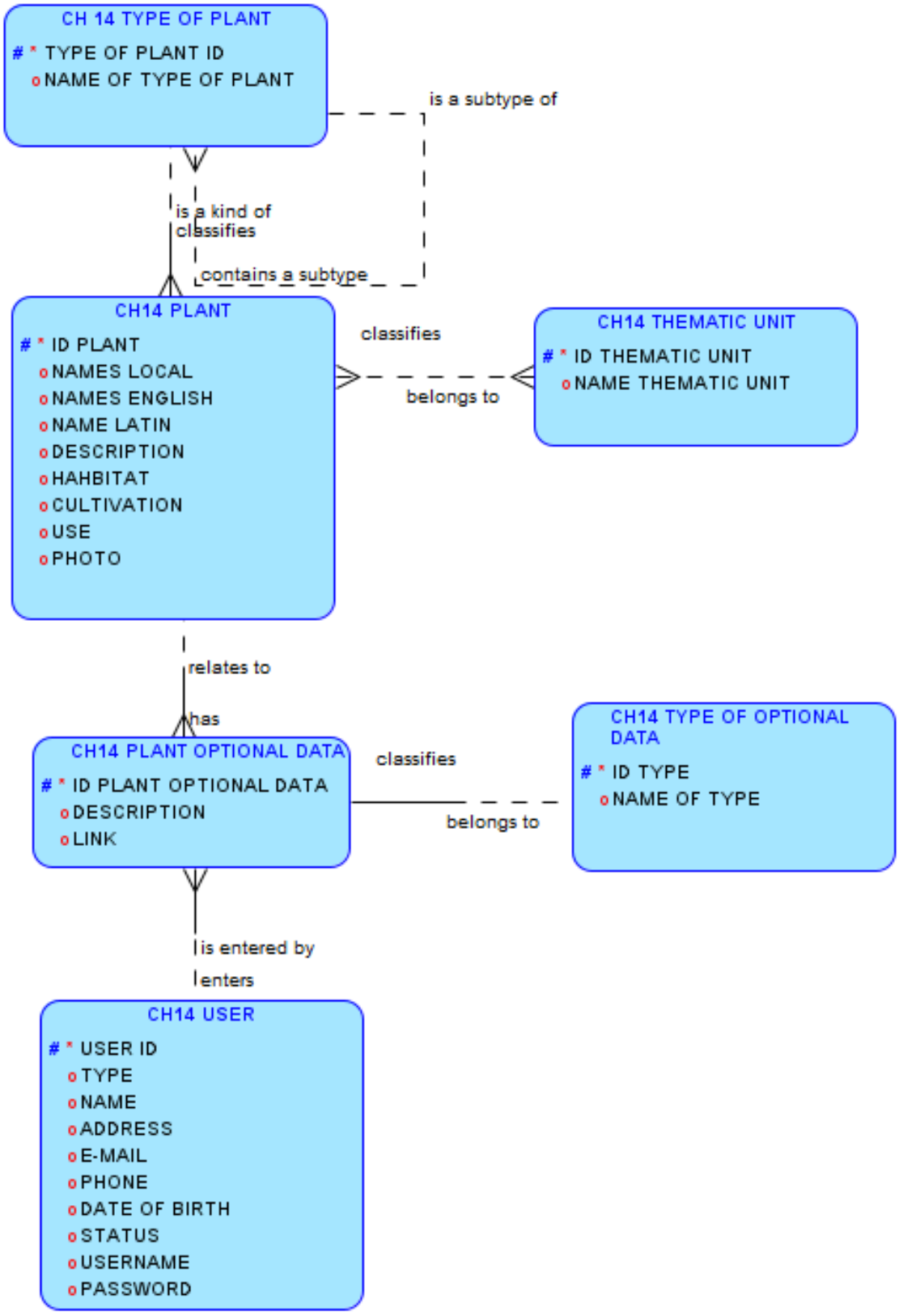
14.4 Podatkovni model

14.4.1 Narativni opis podatkovnog modela

Logički podatkovni model (model Entiteti-Veze) sadrži nekoliko entiteta. Osnovni entitet je biljka (Plant), opisana sljedećim atributima: Plant_ID (identifikacijski atribut biljke); names local (lokalni naziv biljke), names english (naziv biljke na engleskom); lnames latin (naziv biljke na latinskom); description (opis biljke), habitat (opis staništa na kojem se biljka nalazi); cultivation (način uzgoja); use (korištenje biljke); photo (jedna ili više slika biljke); hiperveze (poveznice na vanjske izvore informacija); status (aktivna biljka koja se razmjenjuje, arhivirana biljka...) Biljke predstavljaju hijerarhijski organiziran popis kodova. To znači da svaka vrsta biljke može sadržavati 0, 1 ili više podvrsta biljaka i može biti uključena kao podvrsta u 0 ili 1 super-vrstu biljaka. Tematska cjelina je i šifrarnik koji definira vrstu biljaka i sjemena (povrće, voće, ljekovito bilje, sobno i ukrasno bilje...). Svaka biljka pripada 0, 1 ili više tematskih jedinica. Svaka tematska cjelina klasificira 0, 1 ili više biljaka. Atributi tematske jedinice su: ThematicUnit_ID (šifra, odnosno identifikacijski atribut tematske cjeline; Name Thematic Unit (jedan ili više naziva). Model također uključuje korisnika kao tip entiteta. Nadalje, ovom modelu podataka dodano je nekoliko tipova entiteta: Plant optional data (neobavezni podatak o biljkama), Type of optional data (vrsta neobveznog podatka), Type of plant (vrsta biljke) i Thematic unit (tematska jedinica). Plant optional data (neobavezni podatak) sadrži sljedeće atribute: plant optional data_ID (identifikacijski atribut); description (sadržaj neobveznog podatka, tj. opis poveznice, ako postoji), link (veza; neobavezno). Type of optional data (vrsta neobveznog podatka) sadrži atribute: type_ID (identifikacijski atribut); name of type (naziv vrste: opće informacije, uporaba, uzgoj, osobno mišljenje, recenzija...). Svaka biljka može imati 0, 1 ili više neobaveznih podataka o biljki. Svaki neobavezni podatak o biljci odnosi se na 1 i samo 1 biljku. Svaki neobavezni podatak o biljci pripada 1 i samo 1 vrsti neobaveznih podataka. Svaka vrsta neobaveznog podatka klasificira 0, 1 ili više neobaveznih podataka o biljci. Neobavezne podatke o biljci unosi 1 i samo 1 korisnik.

14.4.2 Logički model podataka

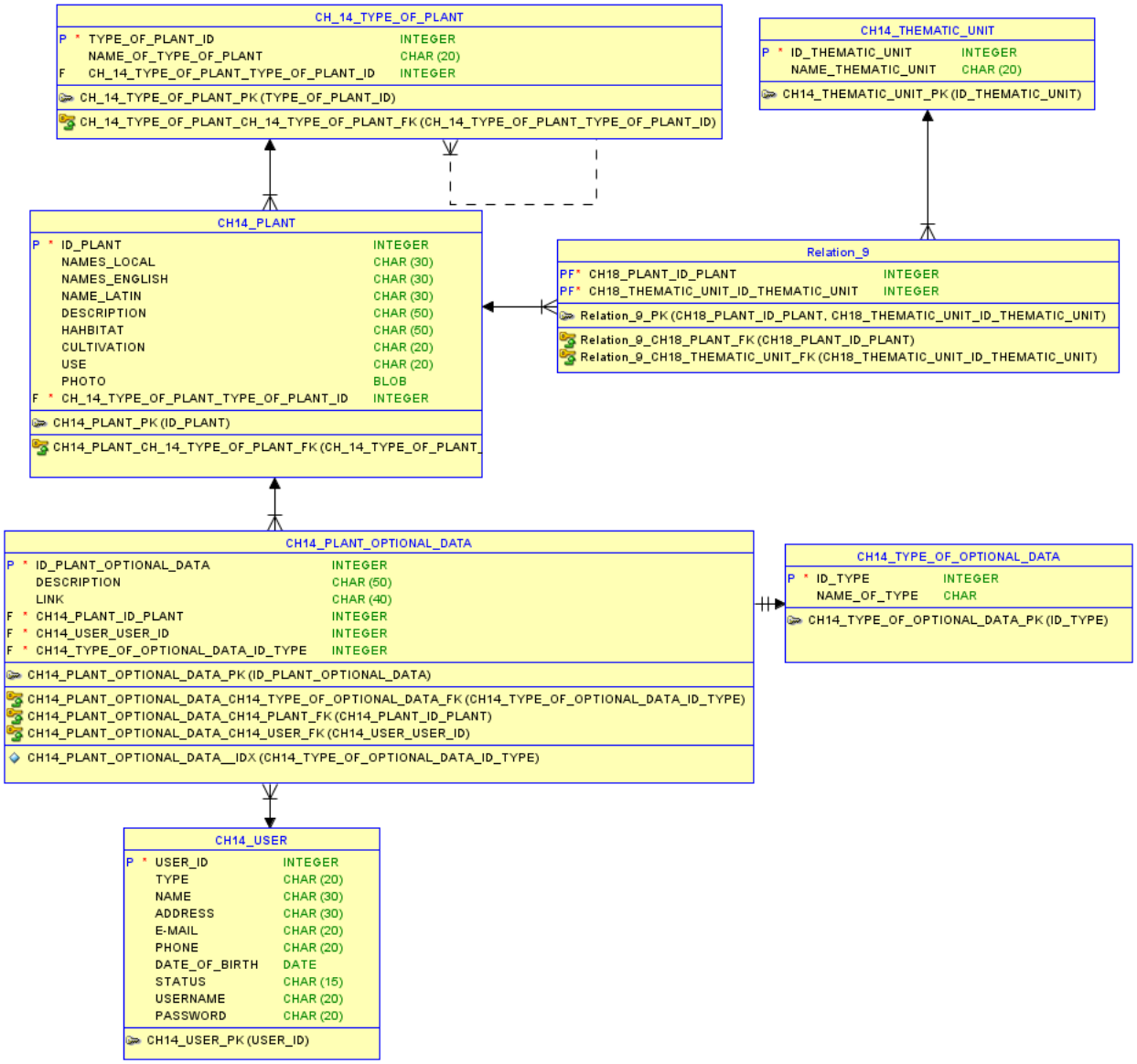
Logički model podataka je prikazan na Figure 14.2.



Slika 14.2: Logički model podataka

14.4.3 Relacijski model podataka

Automatsku transformaciju iz logičkog podatkovnog modela u relacijski podatkovni model u Oracle SQL Data Modeleru omogućuje funkcija *Engineering to relational*. Rezultat je prikazan na Figure 14.3.

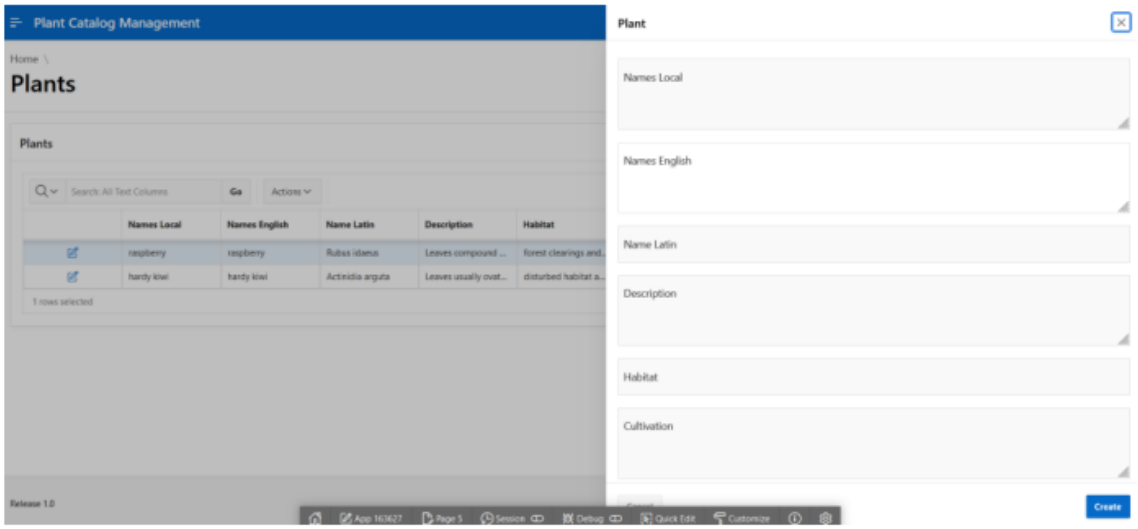


Slika 14.3: Relacijski model podataka

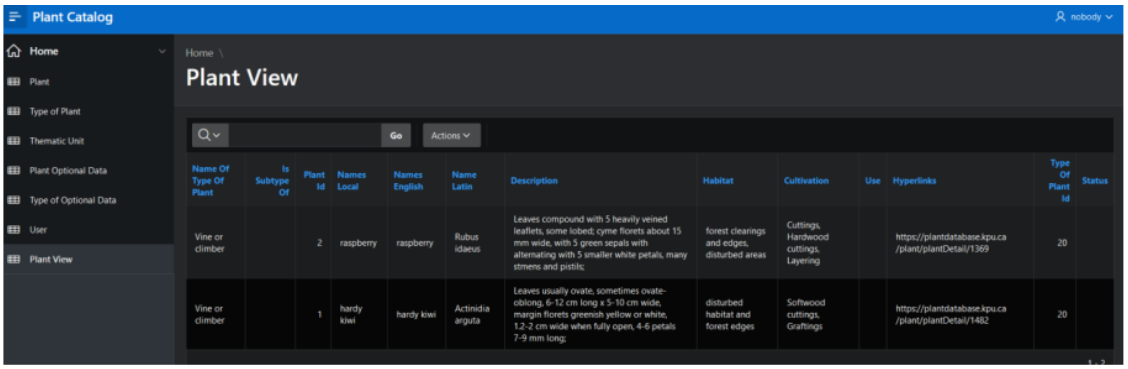
Neka sada Oracle SQL Data Modeler generira SQL skriptu. Odaberite sve tablice na relacijskom modelu i upotrijebite funkciju *File* > *Export* > *DDL File* da bi dobili skriptu. Možete uvesti generiranu skriptu u APEX i izvršiti je.

14.5 Sučelja aplikacije

Pojedinosti o interaktivnoj mreži (Interactive Grid) obrasca za unos podataka o biljkama (Plant) prikazane su na Figure 14.4.



Slika 14.4: Interaktivna mreža (Interactive Grid) obrasca za unos podataka o biljkama (Plant)



Slika 14.5: Javni dio – otvoreno pregledavanje podataka o biljkama

Pregledavanje kataloga lokalnih biljaka provodi se pomoću obrasca za pretraživanje.

14.6 Dodatni materijal za učenje

Dostupni su vam sljedeći dodatni materijali za učenje:

- skripta za izradu i popunjavanje tablica,
- skripta za ispuštanje tablica,
- izvezena aplikacija,
- video koji pokazuje kako generirati aplikaciju.

Sav dodatni materijal za učenje dostupan je na public BeeAPEX project page. Prijavite se kao gost (lozinka nije potrebna). Navigirajte do sekcije Books, skripte u mapi Part 2 > Chapter14 u sekciji Scripts i video upute u Collection of video guides. Materijal za kratke tečajeve dostupan je u Short courses section.

14.6.1 Izvezena aplikacija

Izvezena aplikacija je zapakirana. Instalacija kreira tablice i popunjava podatke. Deinstalacija uklanja sve objekte baze podataka koji se koriste u ovoj aplikaciji.

Zapakirana aplikacija je testirana i pokrenut će se u novom radnom prostoru ako su ispunjeni sljedeći zahtjevi:

- dodajte APEX korisnika prije pokretanja aplikacije. U radnom prostoru za razvoj i testiranje navigirajte do Shared Components > Application Access Control > Add User Role Assignment; unesite APEX korisnika i dodajte mu uloge Administrator, Contributer and Reader. U produkciji prije razvoja potrebno je konzultiranje sa stručnim osobljem.

Ako korisniku nije dodijeljena odgovarajuća uloga, uvezena aplikacija će se srušiti. Potrebno je izbrisati kolačiće web preglednika (npr. Firefox: Settings > Cookies and Site Data > Manage Data) nakon što se aplikacija sruši zbog neispunjenih zahtjeva.

14.6.2 Video upute

Video upute prikazuju svaki korak u razvoju aplikacije.

14.7 Pitanja

1. Istražite kako bi se mogla implementirati komunikacija s vanjskim izvorom, drugom bazom podataka ili sustavom koji sadrži podatke o biljkama.
2. Navedite neke od formata u kojima se pohranjuju slike biljaka i sjemenki.
3. U kojoj se vrsti podataka spremaju slike, video zapisi i slični objekti u Oracle bazi podataka?

14.8 Odgovori

1. U principu, postoje dva načina ostvarivanja komunikacije s takvim vanjskim izvorom. Jedan od načina je preuzimanje seta podataka, odnosno kompletne baze podataka u katalog biljaka. Drugi način je povezivanje pomoću aplikacijskog programskog sučelja (API). To je skup protokola, alata i standarda koji definira način na koji softverske komponente trebaju međusobno djelovati i komunicirati. API-ji omogućuju različitim softverskim sustavima da međusobno komuniciraju i dijele podatke na standardiziran način.
2. Neki od najčešćih formata su:
   a. JPEG (Joint Photographic Experts Group) - ovo je popularan format za digitalne fotografije zbog svojih mogućnosti kompresije, koje omogućuju manje veličine datoteka bez značajnog gubitka kvalitete. Podržava milijune boja i može rukovati slikama visoke razlučivosti.
   b. BMP (Bitmap) - ovaj se format obično koristi u operativnim sustavima Windows i može podržavati slike u sivim tonovima i slike u boji. Ne podržava kompresiju, tako da datoteke mogu biti velike, ali je dobra opcija za pohranjivanje visokokvalitetnih slika.
   c. PNG (Portable Network Graphics) - ovaj je format poznat po svojoj kompresiji bez gubitaka, što znači da kvaliteta slike nije ugrožena kada se veličina datoteke smanji. Podržava transparentnost i često se koristi za web grafiku i digitalnu umjetnost.
   d. TIFF (Tagged Image File Format) - ovaj se format obično koristi za pohranu visokokvalitetnih slika, poput onih koje se koriste u tiskanju. Podržava kompresiju bez gubitaka, više slojeva i može rukovati slikama u sivim tonovima i slikama u boji.
3. Slike, videozapisi i slični objekti obično se pohranjuju u Oracle bazi podataka pomoću tipa podataka BLOB (Binary Large Object). BLOB može pohraniti binarne podatke, poput slika i videa, veličine do 4 gigabajta. Ova vrsta podataka idealna je za pohranjivanje velikih objekata poput multimedijskih datoteka, budući da omogućuje učinkovito pohranjivanje i dohvaćanje binarnih podataka.



15. GreenDi - Autorizacija i upravljanje korisnicima

VJERAN STRAHONJA, DARKO ANDROČEC, ANA KUTNJAK, LARISA HRUSTEK

15.1 Poslovni pogled na slučaj

Kratki pregled poslovanja GreenDi platforme opisan je u poglavlju 14. Taj se dio fokusira na otvoreni dio platforme GreenDi. Napredne funkcionalnosti moguće su samo za registrirane korisnike (članove i administratore). Potrebno je omogućiti funkcionalnost registracije korisnika, prijave registriranih korisnika u sustav uz autentifikaciju i autorizaciju za obavljanje određenih radnji. Potrebne funkcionalnosti uključuju suspendiranje korisnika pod određenim uvjetima i praćenje povijesti podataka o registriranom korisniku.

15.2 Definicija problema

Sljedećih nekoliko rečenica opisuje problem autorizacije i upravljanja korisnicima:

- Za dublji uvid u informacije o biljkama i ostalim naprednim funkcionalnostima potrebno je prijaviti se na platformu, odnosno to je moguće samo registriranim korisnicima (članovima).
- Prijava je zajednička za sve funkcionalnosti i u načelu mora biti jedinstvena prijava.
- Osnovni mehanizam provjere autentičnosti korisnika je korisničko ime i lozinka.
- Informacije o trenutnoj lozinci nisu vidljive administratoru ili bilo kojem drugom korisniku.
- Proces registracije uključuje unos osnovnih potrebnih korisničkih podataka, kao što su ime, adresa, e-mail, telefon, korisničko ime i lozinka.
- Opcionalni podaci su interesi korisnika, datum rođenja itd.
- Nakon uspješne registracije, novi korisnik automatski dobiva zadani tip Člana i status Aktivan. Promjenu statusa u Administrator i obrnuto, kao i vrstu može izvršiti samo drugi aktivni administrator.
- Prilikom registracije član prihvaća uvjete korištenja. Ukoliko ih on ili ona prekrši, administratori suspendiraju članstvo, što je također opisano u uvjetima korištenja.
- Redoviti upis i brisanje članstva je dobrovoljan i obavlja ga sam korisnik.
- Svaka promjena bilo kojeg korisničkog podatka bilježi se u obliku dnevnika. Administrator može pregledati povijest promjena svakog korisnika, osim lozinke.
- Na zahtjev korisnika sustav omogućuje resetiranje lozinke, na način da administrator korisniku šalje privremenu lozinku putem e-maila ili SMS-a.
- Opcionalna funkcionalnost: Osim korisničkog imena i lozinke, autentifikacija je opcionalno moguća putem Facebooka i Google prijave ili drugih servisa. Na primjer, web aplikacija koja

koristi OAuth 2.0 za pristup Google API-jima mora imati vjerodajnice za autorizaciju koje identificiraju aplikaciju Googleovom OAuth 2.0 poslužitelju.

15.3 Slučajevi korištenja

15.3.1 Narativni opis slučaja korištenja

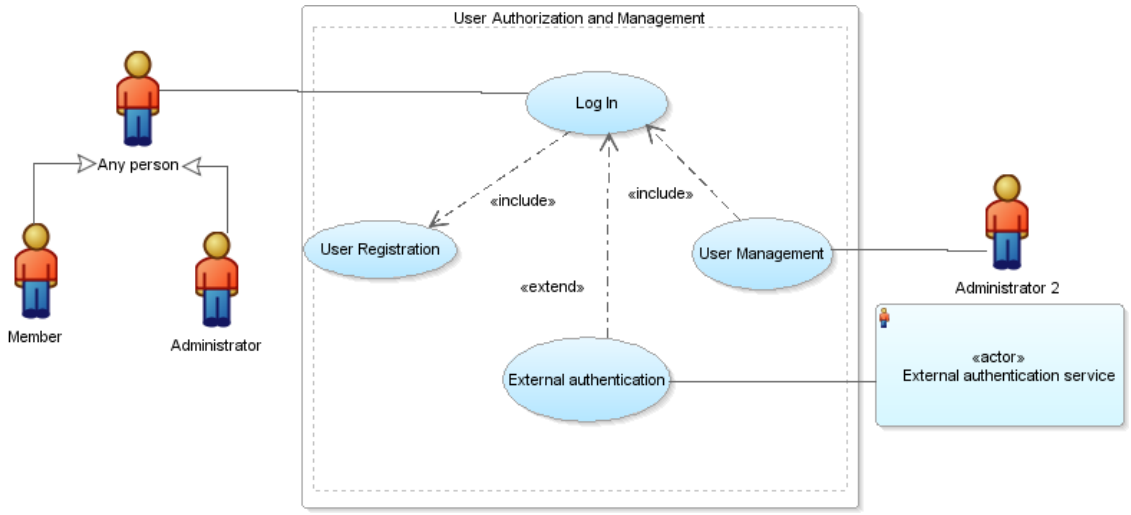
Dijagram slučajeva korištenja opisuje komunikaciju Učesnika i drugih sustava u okolici sa sustavom kojeg razvijamo. Slučajevi korištenja reprezentiraju fukcionalnost autorizacije i upravljanja korisnicima što je opisano na slici 15.1.

15.3.2 Polustrukturirani opis

Tablica 15.1 prikazuje slučaj korištenja.

15.3.3 Dijagram slučaja korištenja

Gornja priča prikazana je na dijagramu slučajeva 15.1.



Slika 15.1: Dijagram slučaja korištenja - GreenDi Autorizacija i upravljanje korisnicima

15.4 Podatkovni model

15.4.1 Narativni opis podatkovnog modela

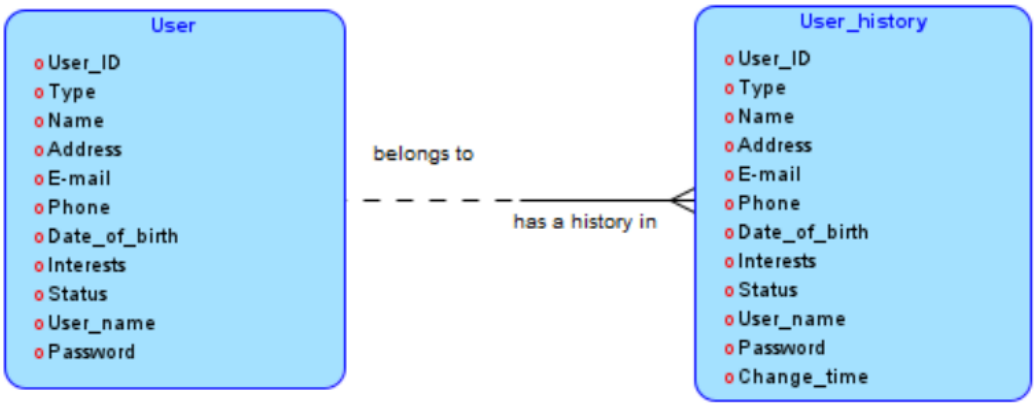
Logički podatkovni model (model Entiteti-Veze) sadrži 2 vrste entiteta: User and User History. User ima sljedeće atribute: User ID (ID attribut); Type (tip korisnika: Member, Administrator); Name (ime i prezime korisnika); Address (adresa korisnika); E- mail; Phone; Date of birth (opt.); Interests (opt.); Status (Active, Suspended, Inactive); User name (nije vidljivo administratoru, šifrirano); Password ((nije vidljivo administratoru, šifrirano) Svaki korisnik ima 1 ili više zapisa korisničke povijesti. Korisnička povijest ima iste atribute kao i Korisnik, to je kopija zapisa User nakon promjene, s tim da ima sistemski generirano vrijeme promjene i ID osobe koja je izvršila promjenu (sam član ili administrator). To znači da se promijenjeni podaci bilježe u ovaj dnevnik nakon svake promjene.

15.4.2 Logički model podataka

Logički model podataka je prikazan na: Figure 15.2.

Tablica 15.1: Opis slučaja korištenja: Autorizacija i upravljanje korisnicima

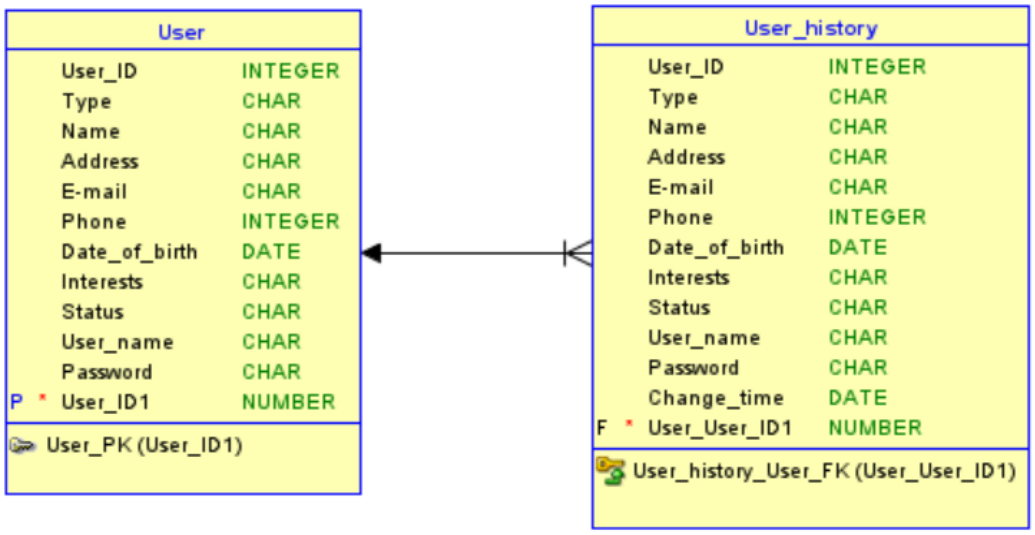
Keyword	**Value**
ID:	*ch15-01*
Title:	*Autorizacija i upravljanje korisnicima*
Description:	*Autorizacija i upravljanje korisnicima.*
Primary Actor:	*Any person, Member, Administrator*
Preduvjeti:	*Prijava obično počinje automatski iz drugih slučajeva upotrebe, gdje je korisnik postavljen kao član ili administrator.*
Postuvjeti:	*Rezultirajuća situacija je da korisnik nije autoriziran, odnosno da je autoriziran, ali je poznat njegov status (na čekanju, aktivan...) i tip (član, administrator).*
Main	-
Uspješan scenarij:	1. Napredne funkcionalnosti GreenDi platforme zahtijevaju prijavu u sustav Log In Form slučaja prijave. 2. Prilikom prve prijave novi se korisnik treba registrirati putem obrasca za registraciju korisnika slučaja registracije korisnika koji je uključen u slučaj korištenja prijave. Proces registracije uključuje unos obaveznih korisničkih podataka, kao što su ime, adresa, e-mail i telefon, kao i izbornih podataka, kao što su interesi korisnika i datum rođenja. b.Korisnik postavlja korisničko ime i lozinku (zapisuju se u posebnu tablicu u bazi podataka, odvojeni su od ostalih korisničkih podataka ili su šifrirani). c. Korisnik prihvaća uvjete korištenja i potvrđuje unesene podatke. d. Zadana vrsta korisnika je Član, a zadani status je Aktivan. 3. Korisničke podatke može mijenjati sam korisnik prilikom svake sljedeće prijave u sustav, putem User Registration Form. 4. Resetiranje lozinke inicira korisnik iz obrasca za prijavu, tako da se na e-mail korisnika šalje poruka s privremenom lozinkom koja traje 15 minuta. 5. Svaka promjena bilo kojeg korisničkog podatka bilježi se u obliku dnevnika u entitetu User History data. Administrator može pregledati povijest promjena svakog korisnika, osim lozinke. 6. Funkcionalnost slučaja korištenja User Management omogućuje korisniku tipa Administrator i u statusu Aktivan da promijeni tip i status drugog (Administrator u člana, i obrnuto, kao i status Aktivan - Suspendiran - Neaktivan). 7. Postoji tablični prikaz korisnika s mogućnošću pretraživanja (Table of Users Form). Na ovom obrascu administrator može odabrati podatke o bilo kojem korisniku, na sličnom obrascu kao i User Registration Form (User Details Form), promijeniti status i tip korisnika, resetirati lozinku i otvoriti obrazac History Form koji prikazuje promjene podataka o odabranom korisniku u tabličnom obliku. 8. Svaka sljedeća prijava registriranog korisnika pokreće provjeru autorizacije korisnika kojom se provjerava korisničko ime i lozinka, status korisnika te je li administrator ili član.
Ekstenzije:	• *Log In obrazac.* • *User Registration obrazac.* • *Table of Users obrazac.* • *User Details obrazac.* • *User History obrazac.*
Frekvencija korištenja:	*Cca. maksimalno je 100 dnevno.*
Status:	*Status razvoja*
Vlasnik:	*javni, anonimni korisnik*
Prioritet:	*visoki*



Slika 15.2: Logički model podataka

15.4.3 Relacijski model podataka

Automatsku transformaciju iz logičkog podatkovnog modela u relacijski podatkovni model u Oracle SQL Data Modeleru omogućuje funkcija *Engineering to relational*. Rezultat je prikazan na SliciFigure 15.3.

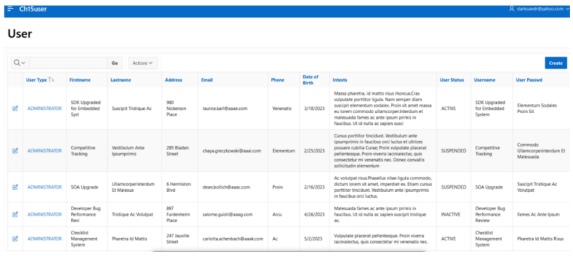


Slika 15.3: Relacijski model podataka

Neka sada Oracle SQL Data Modeler generira SQL skriptu. Odaberite sve tablice na relacijskom modelu i upotrijebite funkciju *File* > *Export* > *DDL File* da bi dobili skriptu. Možete uvesti generiranu skriptu u APEX i izvršiti je.

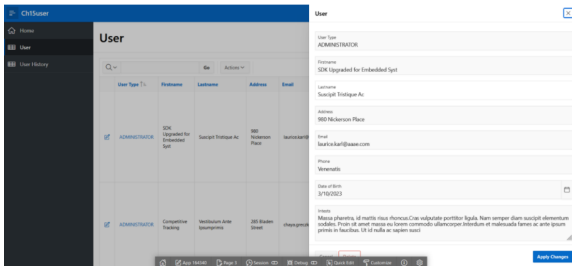
15.5 Sučelja aplikacije

User Form je prikazana na Slici Figure 15.4.



Slika 15.4: User Form

Unos podataka u User History je prikazan na Slici Figure 15.5.



Slika 15.5: User History obrazac

Funkcionalnosti koje zahtijevaju autorizaciju korisnika opisane su u poglavlju 18 Webshop - GreenDi.

15.6 Dodatni materijal za učenje

Možete pronaći sljedeće materijale za učenje:

- izvezena aplikacija,
- skripte za stvaranje, brisanje i umetanje,
- video upute.

Sav dodatni materijal za učenje dostupan je na public BeeAPEX project page. Prijavite se kao gost (lozinka nije potrebna). Navigirajte do sekcije Books, skripte u mapi Part 2 > Chapter15 u sekciji Scripts i video upute u Collection of video guides.Materijal za kratke tečajeve je u Short courses section.

15.6.1 Izvezene aplikacije

Izvezena aplikacija je zapakirana. Instalacija kreira tablice i popunjava podatke. Deinstalacija uklanja sve objekte baze podataka koji se koriste u ovoj aplikaciji.

15.6.2 Video upute

Video upute prikazuju svaki korak u razvoju aplikacije.

15.7 Pitanja

1. Koje su ugrađene funkcije i funkcije provjere autentičnosti korisnika i autorizacije na razini aplikacije pri razvoju aplikacija pomoću Oracle APEX-a?
2. Kako možemo koristiti prijavu putem Facebooka, Googlea i drugih platformi u aplikaciji ili web stranici koju sami razvijamo?
3. Koje su prednosti za korisnike kada se prijave u aplikaciju sa svojim Facebook, Google ili sličnim računima?

15.8 Odgovori

1. a) Ugrađena provjera autentičnosti korisnika: APEX pruža ugrađenu funkciju provjere autentičnosti korisnika, omogućujući definiranje korisničkih računa i upravljanje njima izravno unutar APEX okruženja. Korisnici mogu imati različite uloge i privilegije, a vi možete kontrolirati pristup određenim stranicama aplikacije ili komponentama na temelju korisničkih uloga. b) Autorizacija na razini aplikacije: APEX omogućuje definiranje preciznih pravila autorizacije na razini aplikacije. Možete odrediti koji korisnici ili uloge imaju pristup određenim stranicama, regijama, gumbima ili drugim komponentama unutar vaše aplikacije.
2. Možete koristiti opcije prijave kao što su Facebook, Google i druge platforme u aplikaciji ili web stranici koju sami razvijate. Mnoge popularne platforme nude programerske API-je (Application Programming Interfaces) koji vam omogućuju da integrirate njihovu funkciju prijave u svoje aplikacije ili web stranice. Na primjer, Facebook nudi Facebook API za prijavu, koji korisnicima omogućuje prijavu na vašu aplikaciju ili web stranicu koristeći svoje Facebook vjerodajnice. Slično tome, Google nudi Google Sign-In API, koji korisnicima omogućuje prijavu koristeći svoje Google račune. Da biste implementirali ove opcije prijave, obično morate registrirati svoju aplikaciju ili web mjesto na portalu za razvojne programere odgovarajuće platforme. Ovaj postupak registracije obično uključuje dobivanje API ključeva ili ID-ova klijenata, koje ćete zatim koristiti u svom kodu za autentifikaciju korisnika i upravljanje postupkom prijave.
3. Uključivanjem opcija prijave s ovih platformi, korisnicima se nudi praktičan i poznat način prijave na aplikaciju ili web stranicu bez potrebe da stvaraju nove račune ili pamte dodatne vjerodajnice za prijavu. Nema potrebe za registracijom i unosom imena, adrese i sličnih podataka, ne stvaraju se nova korisnička imena i lozinke koje treba pamtiti. Dodatno, prijava je puno brža.



16. inovativni sustavi

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA

16.1 Poslovni pogled na slučaj

U suvremenom poduzeću, inovativne ideje ključ su uspjeha i poboljšanja poslovnih performansi. Zaposlenici su zbog svog znanja o poslovnim procesima vjerojatno najbolji izvor inovativnih ideja. Te ideje uključuju različita stajališta kao što su prijedlozi za optimizaciju procesa, poboljšanja upravljanja odnosima s klijentima, nadogradnju osiguranja kvalitete itd. Stoga je menadžment u ovom konkretnom slučaju odlučio prikupiti male inovativne ideje od zaposlenika i nagraditi ih.

16.2 Definicija problema

Inovativne ideje u poduzeću prikupljale su se putem e-pošte i papirnatih obrazaca. Pregled ideja u poduzeću oduzimao je puno vremena. Kriteriji vrednovanja za ocjenu ideja nisu bili definirani. Zbog velikog broja potencijalnih ideja koje bi mogle biti od koristi poduzeću, postoji potreba za dobrim pregledom. Platforma će omogućiti prikupljanje inovativnih ideja u poduzeću. Zaposlenici će moći prijaviti se na platformu i unijeti svoje ideje uz jednostavne opise ili privitke. Budući da ideje nemaju jednaku vrijednost (za menadžment i poslovanje), iste će biti pregledane i evaluirane. Kada evaluatori završe proces, ideja se ocjenjuje kao prihvaćena, odbijena ili je potrebna dorada. Najinovativnije ideje bit će nagrađene ovisno o ocjeni koju dobiju. Za predložene ideje mogu se dobiti sljedeće nagrade: ZLATO (100 EUR), SREBRO (80 EUR), BRONCA (60 EUR), DOPRINOS (40 EUR), ili HVALA. Samo menadžeri imat će mogućnost ocjenjivanja ideja. Menadžeri će redovito inicirati ocjenjivanje i nagrađivanje (tj. jednom mjesečno). Svi prijavljeni korisnici moći će pregledati ideje nakon što budu ocijenjene.

16.3 Slučajevi korištenja

16.3.1 Narativni opis slučaja korištenja

Svaki zaposlenik ima priliku predložiti inovativne ideje putem obrasca za prijavu. Sve ideje se prikupljaju i ocjenjuju od strane uprave. Ideje se rangiraju prema korisnosti i mogućnostima realizacije te se dobre ideje nagrađuju.

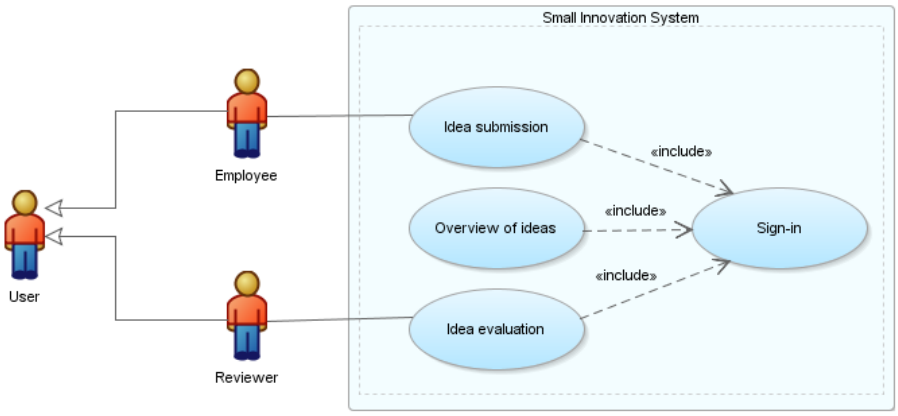
16.3.2 Polustrukturirani opis

Mogu se sažeti tri različite korisničke priče ili slučajeva korištenja: podnošenje ideje, pregled ideja i evaluacija ideja. Posljednji korak - isplata nagrade nije obuhvaćen ovim poslovnim slučajem.

Dodavanje priloga ideji zapravo je proširenje gore opisanog slučaja korištenja i mogli bismo ga opisati kao novi slučaj upotrebe. Radi čitljivosti, nastavljamo s druga dva glavna slučaja upotrebe, međutim, aplikacija će imati značajku za dodavanje privitaka ideji.

16.3.3 Dijagram slučaja korištenja

Prethodno navedeno prikazano je na dijagramu slučaja korištenja (vidi Figure 16.1).



Slika 16.1: Dijagram slučaja korištenja

Također, radi jednostavnosti proširenja slučaja korištenja (dodavanje privitka ideji) preskočen je Figure 16.1.

16.4 Podatkovni model

16.4.1 Narativni opis podatkovnog modela

Postoje tri entiteta u logičkom modelu podataka. Entitet CH16 IDEA ima osam atributa: ID, datum podnošenja, naslov, opis ideje, naziv datoteke ideje, tip, odluku i datum odluke. Svaku ideju predlaže jedan zaposlenik. U entitetu CH16 EMPLOYEE pohranjuju se osnovni podaci o zaposlenicima: ID, ime i prezime. Zaposlenik može biti menadžer drugim zaposlenicima. Svaki zaposlenik ima samo jednog voditelja, dok voditelj može voditi više zaposlenika. Menadžeri ocjenjuju ideje. Jedan menadžer može ocijeniti više ideja, dok jednu ideju može ocijeniti samo jedan menadžer. Svaka ideja može dobiti jednu nagradu. CH16 AWARD entitet ima tri atributa: ID, opis nagrade i iznos u eurima. Jedna vrsta nagrade može se dodijeliti za više ideja. Pravila evaluacije nisu uključena u podatkovni model jer bi to predstavljalo rizik od promjene aplikacije svaki put kada se pravila promijene. Predloženim modelom podataka osigurali smo: robusnost aplikacije, brz razvoj i implementaciju, niske troškove razvoja i održavanja te fleksibilnost.

16.4.2 Logički model podataka

Gornja priča je predstavljena na logičkom modelu podataka (vidiFigure 16.2).

Tablica 16.1: Opis slučaja korištenja: Podnošenje ideje

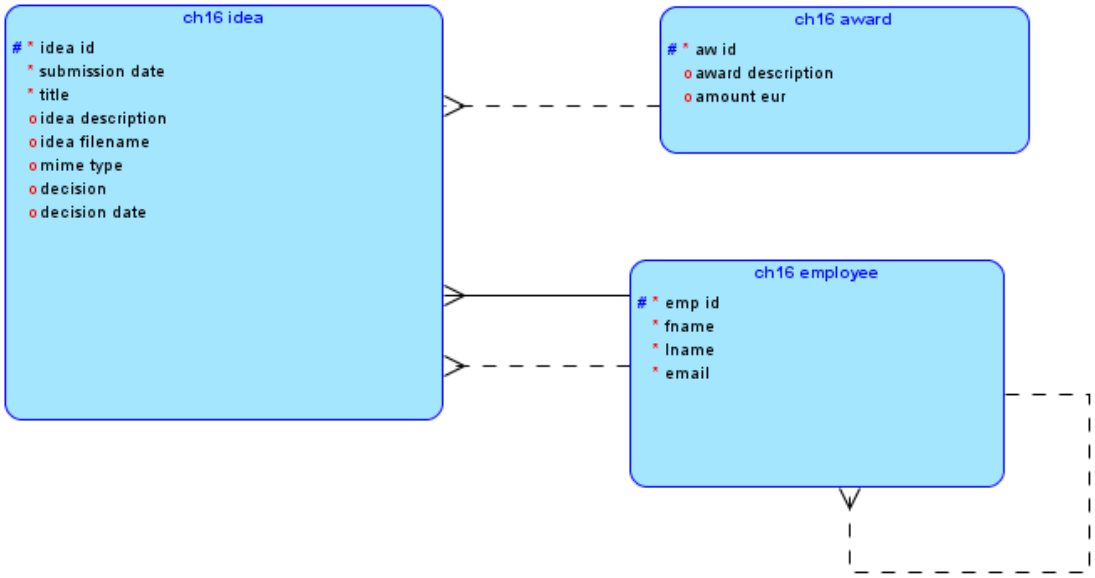
Ključna riječ	**Vrijednost**
ID:	*Ch16-01*
Naslov:	*Podnošenje ideje*
Opis:	*Zaposlenik se prijavljuje na platformu i unosi ideju. Svaka ideja ima naslov i kratki opis koji se dodaje kao privitak (datoteka dokumenta).*
Primarna uloga:	*Zaposlenik*
Preduvjeti:	*Zaposlenik mora imati Oracle APEX korisnički račun za aplikaciju Small Innovation System. Pristup web aplikaciji mora biti omogućen.*
Postuvjeti:	*Nakon uspješnog unosa, ideja i detaljan opis u prilogu dostupni su ostalim korisnicima SIS portala.*
Glavni	*Scenariji*
Uspješan scenarij:	1. *Otvaranje web preglednika i prijava u SIS aplikaciju* 2. *Odabir opcije Ideas u izborniku* 3. *Odabir opcije Add new idea* 4. *Unos naslova ideje i dodavanje priloga* 5. *Unos naziva datoteke i vrste privitka* 6. *Potvrda umetanja klikom na gumb Kreiraj* 7. *Pregled unesenih podataka*
Ekstenzije:	• *1a. Prijava nije uspjela* • *1a* Prošireno:* • *1a1. Prikazivanje poruke o pogrešci* • *1a2. Otvaranje prozora za prijavu* • *4a. Pogreška tipa podataka* • *4a* Prošireno:* • *4a1. Prikazivanje poruke o pogrešci* • *7a. Pogreška tipa podataka* • *7a* Prošireno:* • *7a1. Prikazivanje poruke o pogrešci*
Frekvencija korištenja:	*U prosjeku, tvrtka tjedno dobije 3 inovativne ideje.*
Status:	*Završeno*
Vlasnik:	*Zaposlenik s pristupom aplikaciji*
Prioritet:	*Umjereno*

Tablica 16.2: Opis slučaja korištenja: Pregled ideja.

Ključna riječ	**Vrijednost**
ID:	*Ch16-02*
Naslov:	*Pregled ideja*
Opis:	*Zaposlenik se prijavljuje na platformu i provjerava status ideje ili drugih ideja.*
Primarna uloga:	*Employee*
Preduvjeti:	*Zaposlenik mora imati Oracle APEX korisnički račun za aplikaciju Small Innovation System. Pristup web aplikaciji mora biti omogućen.*
Postuvjeti:	-
Glavni	*Scenariji*
Uspješan scenarij:	1. *Otvoren web preglednik i prijava u SIS aplikaciju* 2. *Odabir opcije Ideas u izborniku* 3. *Pregled navedenih ideja i njihovog statusa* 4. *U izborniku odabir opcije Overview* 5. *Pregled grafikona s podacima o predanim idejama*
Extensions:	• *1a. Prijava nije uspjela* • *1a* Prošireno:* • *1a1. Prikazivanje poruke o pogrešci*
Frekvencija korištenja:	*Mogućnost pregleda ideja dana je zaposlenicima, recenzentima i administratorima. Sveukupno pristupaju platformi otprilike 5 puta dnevno.*
Status:	*Završeno*
Vlasnik:	*Zaposlenik s mogućnošću prijave*
Prioritet:	*Visok*

Tablica 16.3: Opis slučaja korištenja: Evaluacija ideja

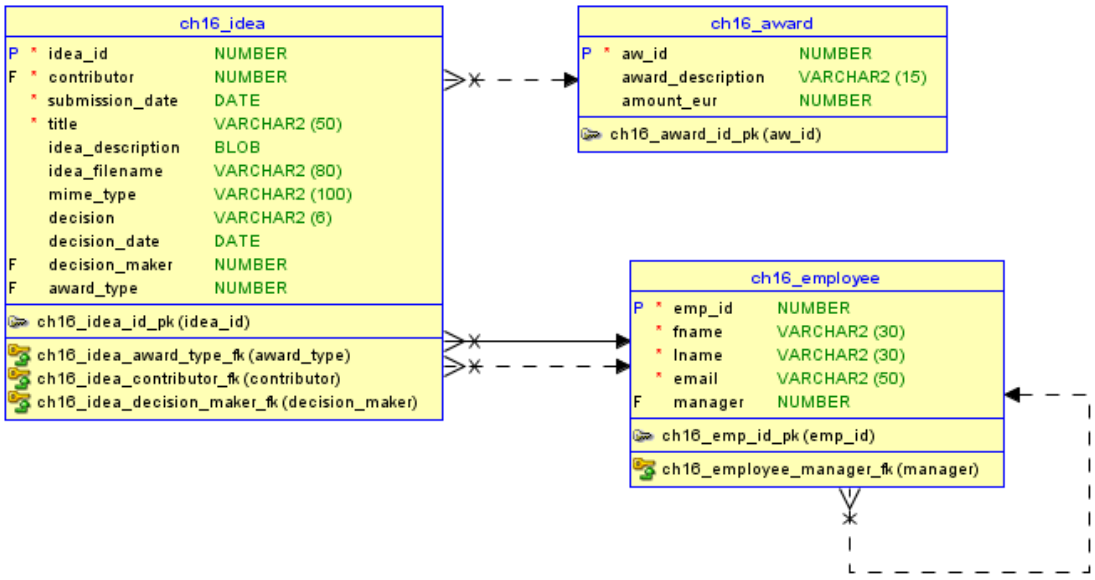
Ključna riječ	**Vrijednost**
ID:	*Ch16-03*
Naslov:	*Evaluacija ideja*
Opis:	*Recenzent se prijavljuje na platformu i pregledava ideju. Nagrada je također definirana u sklopu ovog slučaja korištenja.*
Primarna uloga:	*Reviewer*
Preduvjeti:	*Korisnik zaposlenik mora biti dodan u CH16_Reviewer role. Pristup web aplikaciji mora biti omogućen.*
Postuvjeti:	*Odabrana ideja se ocjenjuje i definira se prijedlog nagrade.*
Glavni	*Scenariji*
Uspješan scenarij:	1. *Otvaranje web preglednika i prijava u SIS aplikaciju* 2. *Odabir opcije Review ideas u izborniku* 3. *Identifikacija ideje i dodavanje detalja pomoću gumba Edit* 4. *Odabir odluke s popisa vrijednosti (Prihvati, Odbaci, Potrebna dorada)* 5. *Odabir vrste nagrade s popisa liste vrijednosti (zlato, srebro, bronca, doprinos, hvala)* 6. *Potvrda ažuriranja*
Ekstenzije:	• *1a. Prijava nije uspjela* • *1a* Prošireno:* • *1a1. Prikazivanje poruke o pogrešci* • *1a2. Otvaranje prozora za prijavu* • *2a. Pogreška tipa podataka* • *2a* Prošireno:* • *2a1. Prikazivanje poruke o pogrešci* • *3a. Pogreška tipa podataka* • *3a* Prošireno:* • *3a1. Prikazivanje poruke o pogrešci* • *4a. Pogreška tipa podataka* • *4a* Prošireno:* • *4a1. Prikazivanje poruke o pogrešci* • *5a. Pogreška tipa podataka* • *5a* Prošireno:* • *5a1. Prikazivanje poruke o pogrešci*
Frekvencija korištenja:	*Menadžeri razmatraju predložene ideje jednom mjesečno.*
Status:	*Završeno*
Vlasnik:	*Reviewer*
Prioritet:	*Visok*



Slika 16.2: Logički model podataka

16.4.3 Relacijski model podataka

Automatsku transformaciju iz logičkog modela podataka u relacijski model podataka u Oracle SQL Data Modeleru omogućuje funkcija*Engineering to relational*. Rezultat je prikazan na Figure 16.3.



Slika 16.3: Relacijski model podataka

Oracle SQL Data Modeler generira SQL skriptu za stvaranje tablice, niza i okidača. Odaberite sve tablice na relacijskom modelu i upotrijebite funkciju *File* > *Export* > *DDL File* da biste dobili skriptu poput ove:

```
CREATE TABLE CH16_AWARD (
    AW_ID NUMBER NOT NULL
...
```

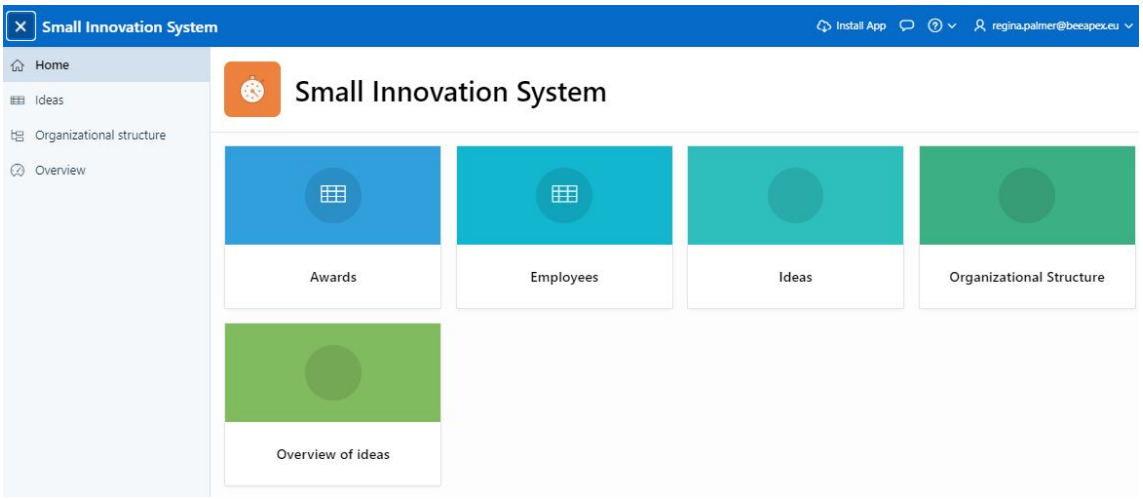
```
);

CREATE TABLE CH16_EMPLOYEE (
    EMP_ID NUMBER NOT NULL
...
);
```

Nakon toga slijedi generiranje tablica u Oracle bazi podataka. Može se uvesti generirana skripta i izvršiti u APEX-u.

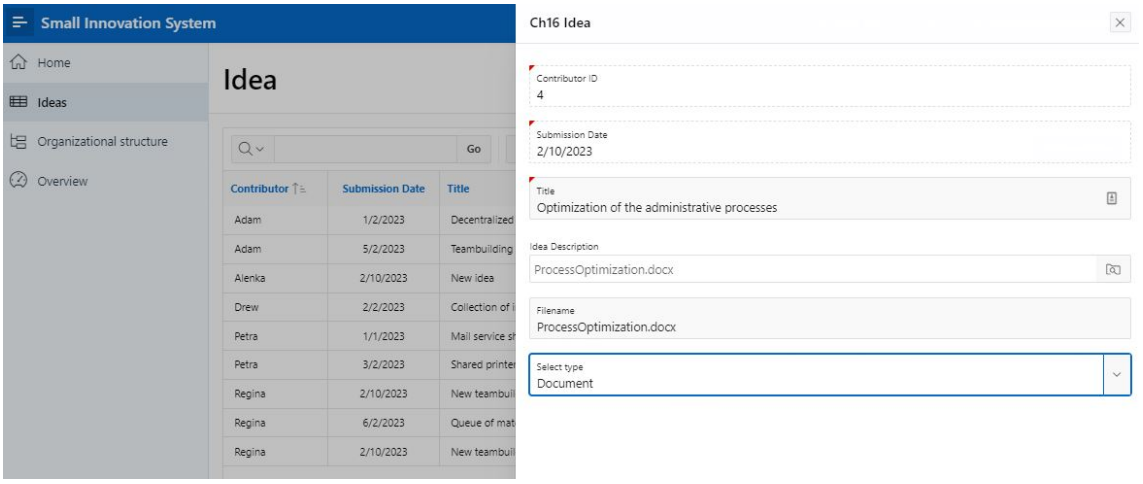
16.5 Sučelja aplikacije

Na sljedećim slikama prikazana su sučelja aplikacije za obje osnovne uloge (zaposlenik i recenzent). Prvo je prikazana nadzorna ploča za zaposlenika Figure 16.4.



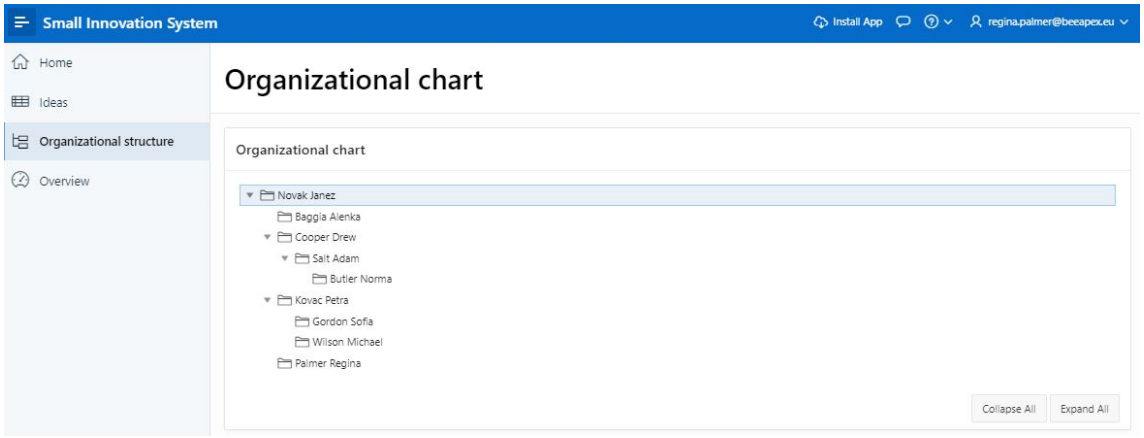
Slika 16.4: Nadzorna ploča za zaposlenike

Stranica za slanje ideja s prilozima i njezinim detaljima predstavljena je na Figure 16.5.



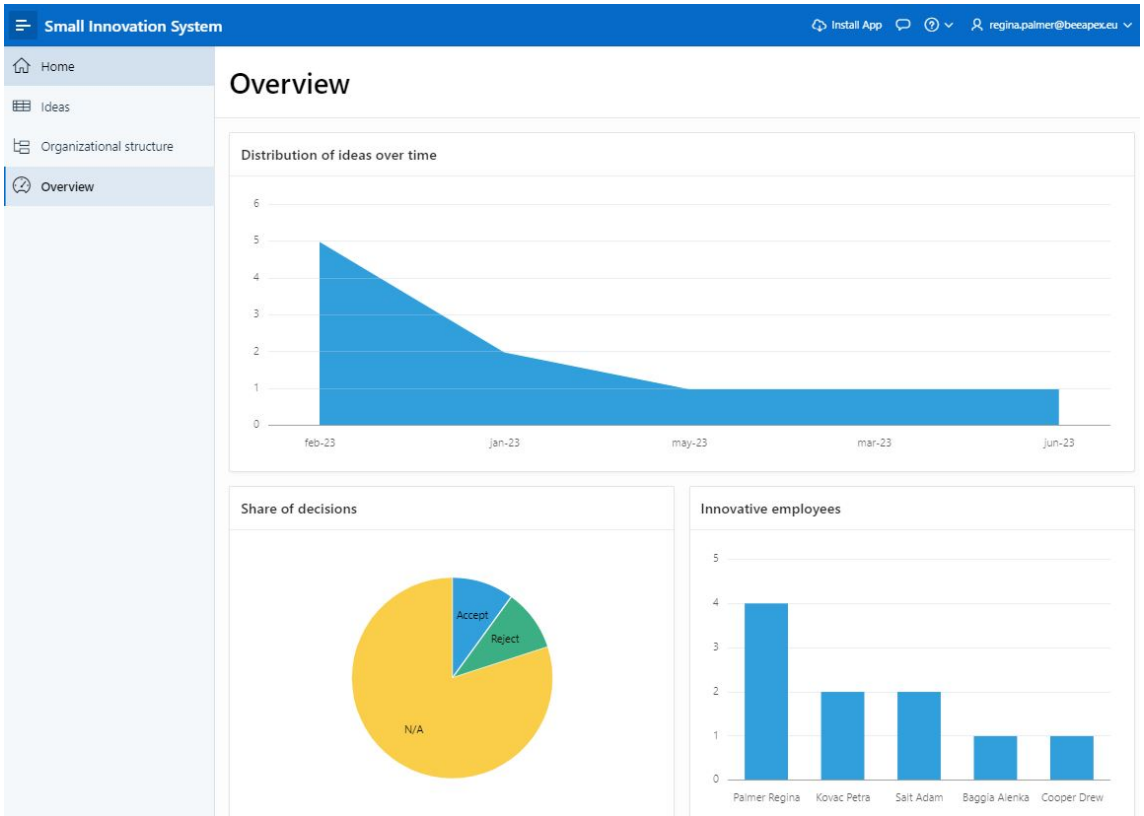
Slika 16.5: Slanje ideje s prilozima

Svaki zaposlenik ima mogućnost pregledati organizacijsku strukturu u poduzeću, kako je prikazano na Figure 16.6.



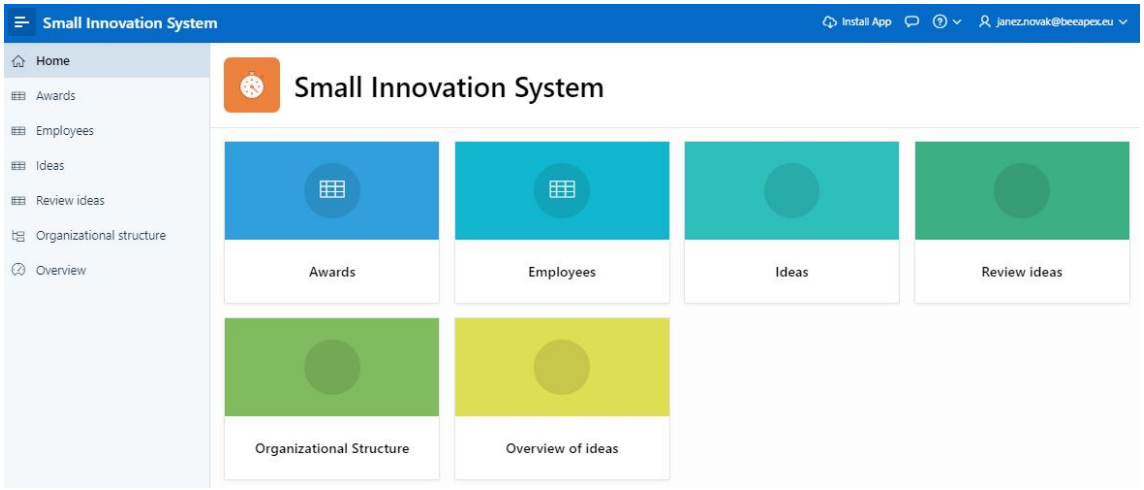
Slika 16.6: Organizacijska struktura u poduzeću

Također, svaki zaposlenik može vidjeti nadzornu ploču grafikona koji prikazuju distribuciju ideja kroz vrijeme, udio prihvaćenih, odbijenih i revidiranih ideja te zaposlenike s najvećim brojem inovativnih ideja u poduzeću (vidi Figure 16.7).



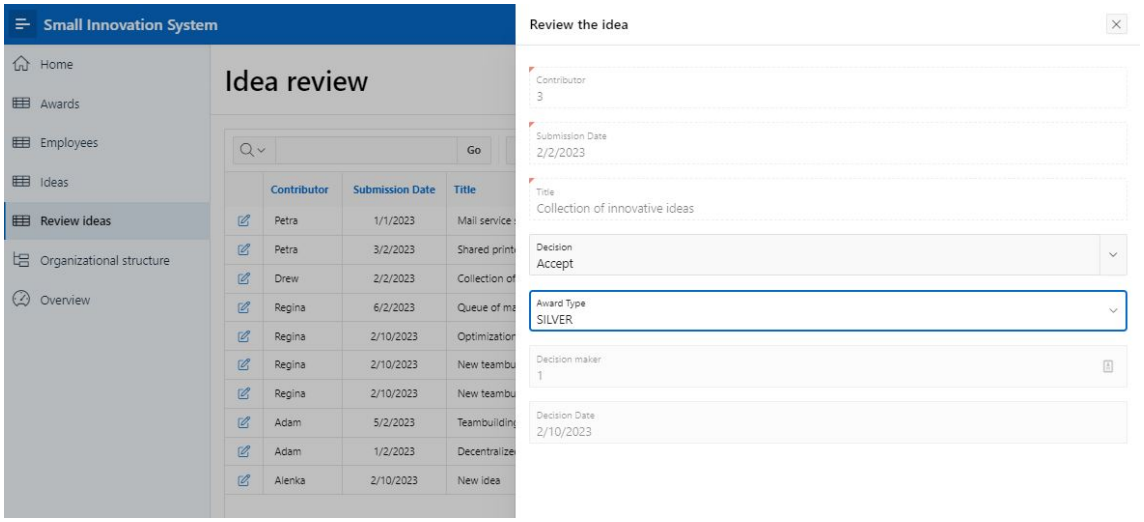
Slika 16.7: Small Innovation Idea pregled

Osim opisanih radnji, recenzent ima mogućnost ažuriranja podataka o nagradama i zaposlenicima, kao što je prikazano na Figure 16.8. Administrativna stranica dostupna je samo administratorima, ne i recenzentima.



Slika 16.8: Nadzorna ploča recenzenta

Stranica za recenziju ideje predstavljena je na Figure 16.9. Ova je stranica dostupna samo recenzentu. Recenzent odabire odluku i vrstu nagrade ako je primjenjivo.



Slika 16.9: Stranica za pregled ideja

16.6 Dodatni materijal za učenje

Možete pronaći sljedeće materijale za učenje:

- skripta za stvaranje,
- skripta za popunjavanje tablica,
- skripta za ispuštanje tablica,
- izvezena aplikacija,
- video koji pokazuje kako generirati aplikaciju iz skripte.

Sav dodatni materijal za učenje dostupan je na public BeeAPEX project page. Prijavite se kao gost (lozinka nije potrebna). Navigirajte do sekcije Books, skripte u mapi Part 2 > Chapter16 u sekciji Scripts i video upute u Collection of video guides. Materijal za kratke tečajeve je u Short courses section.

16.6.1 Izvezene aplikacije

Izvezena aplikacija je zapakirana. Instalacija kreira tablice i popunjava podatke. Deinstalacija uklanja sve objekte baze podataka koji se koriste u ovoj aplikaciji.

Zapakirana aplikacija je testirana i pokrenut će se u novom radnom prostoru ako su ispunjeni sljedeći zahtjevi:

- dodajte APEX korisnika prije pokretanja aplikacije.

Ako korisniku nije dodijeljena odgovarajuća uloga, uvezena aplikacija će se srušiti. Potrebno je izbrisati kolačić web preglednika (i.e. Firefox: Settings > Cookies i Site Data > Manage Data) nakon što se aplikacija sruši zbog neispunjenih zahtjeva.

16.6.2 Video upute

Video upute prikazuju svaki korak u razvoju aplikacije.

16.7 Pitanja

1. Gdje su korisničke uloge u Poglavlju 16 definirane?
2. Koji odnos u logičkom modelu podataka omogućuje prikaz organizacijske strukture u Poglavlju 16?
3. Zašto su neka polja na stranici za pregled ideja onemogućena u Poglavlju 16?

16.8 Odgovori

1. Uloge nisu definirane u Poglavlju 16 podatkovnog modela. Koriste se samo u APEX okruženju.
2. Odnos koji povezuje tablicu zaposlenika sa samim sobom (rekurzivni odnos) omogućuje prikaz organizacijske strukture u Poglavlju 16.
3. U Poglavlju 16 neka su polja onemogućena kako bi se stvorilo bolje korisničko iskustvo i osigurala dosljednost podataka budući da korisnik nije dužan dodavati podatke koji se već nalaze u bazi podataka.



17. Upravljanje poslovnim procesima

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA

17.1 Poslovni pogled na slučaj

Veći dio ove knjige naglašava važnost podataka i pristupa koji se temelji na podacima za razvoj web aplikacija s APEX-om. Ali pristup vođen podacima ima i zrcalnu sliku: aplikacije vođene procesima. Kao što naziv sugerira, aplikacije vođene procesima usredotočuju se na procese. Međutim, niti, dobra aplikacija vođena podacima ne može zanemariti procesni aspekt, niti aplikacija vođena procesima može zanemariti podatkovni aspekt. Implicitno ili eksplicitno uključivanje oba aspekta, čini svaku aplikaciju svrhovitom. Važnoi je da programeri imaju dubok uvid u poslovne procese u organizaciji. Bilo koja vrsta organizacije, neovisno o veličini, prihodu, vlasništvu, djelatnosti i misiji, provodi poslovne procese. Uobičajene klase poslovnih procesa koje se primjenjuju na većinu organizacija spadaju u sljedeće kategorije:

- prodaja i marketing, uključujući potprocese kao što su stjecanje i zadržavanje kupaca, stvaranje potencijalnih kupaca, upravljanje odnosima s kupcima i marketinške kampanje.
- upravljanje opskrbnim lancem uključujući potprocese kao što su koordinacija i upravljanje protokom roba i usluga, nabava, proizvodnja i distribucija.
- financijsko upravljanje, uključujući potprocese kao što su upravljanje financijskim resursima organizacije, računovodstvo, proračun i financijsko izvješćivanje.
- upravljanje ljudskim resursima, uključujući potprocese kao što su upravljanje radnom snagom organizacije, zapošljavanje, osposobljavanje i razvoj zaposlenika.
- služba za korisnike, uključujući potprocese kao što su pružanje podrške i pomoći korisnicima, rješavanje pritužbi i problema.
- upravljanje operacijama, uključujući potprocese kao što su upravljanje svakodnevnim poslovnim operacijama, proizvodnjom, logistikom i upravljanje zalihama.
- informacijska tehnologija, uključujući podprocese kao što su upravljanje resursima i sustavima informacijske tehnologije organizacije, hardverom, softverom i upravljanjem podacima.

Ti se procesi smatraju kritičnima jer podržavaju temeljne funkcije organizacije i omogućuju postizanje njezinih ciljeva. Poboljšanje učinkovitosti i djelotvornosti ovih procesa može imati značajan utjecaj na ukupnu izvedbu i uspjeh organizacije. U pokušaju jasnog razumijevanja i definicije tipičnog poslovanja OASIS OPEN je razvio univerzalni poslovni jezik koji opisuje procese i poslovne dokumente (vidi [5]). Softverska industrija započela je kasnih 1990-ih s integracijom procesa, razvojem Business Process Execution Language (BPEL) i kasnije Business Process Modeling Language (BPML). Veliki igrači poput Microsofta, IBM-a, SAP-a, Oracle-

a kao i mnoštvo manjih, pridonijeli su softverskim platformama koje su imale za cilj pružiti besprijekornu integraciju poslovnih funkcija. Osim svijeta softvera, druge su industrije izoštrile foku,s na procese kroz objektiv specifičnih poduzeća iz 1980-ih s inicijativama za upravljanje kvalitetom i standardima.

Od 1990-ih discipline upravljanja poslovnim procesima (BPM) i reinženjeringa poslovnih procesa (BPR) imaju važnu ulogu kada organizacije žele poboljšati svoje poslovanje. BPM je usredotočen na prikaz i dokumentiranje tijekova rada, operacija i procesa tvrtke pomoću dijagrama procesa, dijagrama tijeka ili drugih vizualnih pomagala. Cilj je razumjeti i poboljšati postojeće procese ili osmisliti nove kako bi se povećala učinkovitost i smanjila rasipanja. BPR se koristi kao temeljno ponovno promišljanje i radikalno redizajniranje poslovnih procesa kako bi se postigla značajna poboljšanja performansi i produktivnosti. Uključuje analizu i dizajn tokova rada i procesa unutar organizacije, s ciljem da bude učinkovitija, djelotvornija i sposobna prilagoditi se promjenjivom poslovnom okruženju. BPR može uključivati tehnologije i informacijske sustave za automatizaciju procesa i poboljšanje donošenja odluka.

U BPM i BPR se mogu koristiti različiti dijagrami:

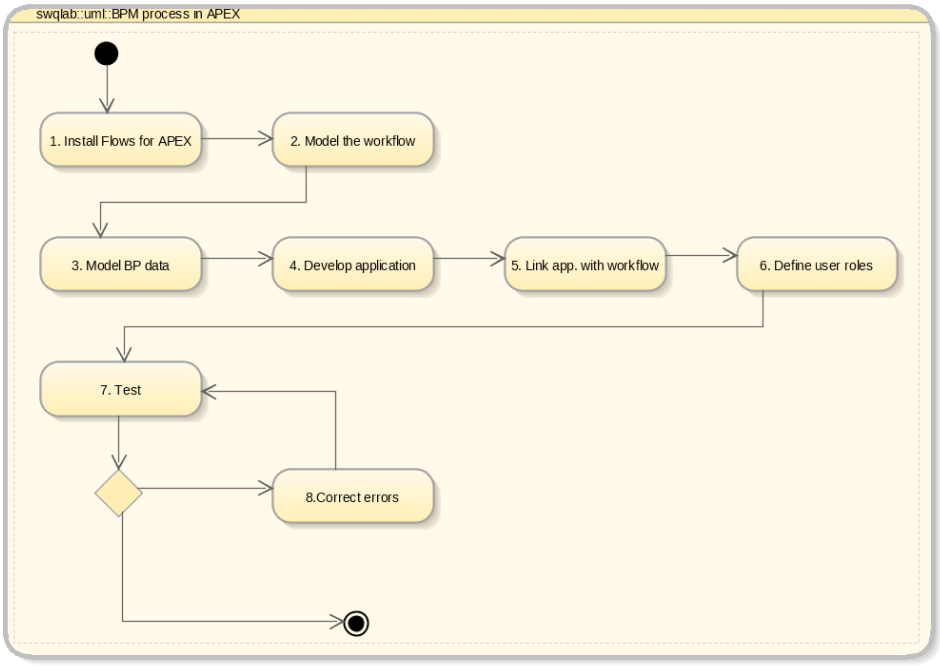
- Dijagram toka (engl. Flowchart): jednostavan i intuitivan način predstavljanja procesa kao niza koraka, odluka i petlji.
- Dijagram s plivačim linijama (engl. Swimlane diagram): Vrsta dijagrama toka koji dodaje vizualni prikaz tko ili što je odgovorno za svaki korak u procesu.
- Mapa procesa (engl. Process map): Pogled na visoku razinu koraka i aktivnosti uključenih u proces, često uključujući ulaze, izlaze i točke odlučivanja.
- BPMN (Business Process Model and Notation [4]): Industrijska standardna notacija koja se koristi za modeliranje poslovnih procesa, standardizirana od strane Object Management Group (OMG).
- EPC (Event-driven Process Chain): Vrsta dijagrama toka koji koristi događaje kao pokretačku snagu procesa, umjesto aktivnosti.
- IDEF (Integration Definition for Function Modeling): Metoda koja se koristi u programskom inženjerstvu za modeliranje i analizu procesa i sustava.

Odabir vrste dijagrama za predstavljanje tijeka rada ovisi o razini potrebnih detalja, namjeni modela i ciljanoj publici. U ovom trenutku ćemo se usredotočiti samo na BPMN. Glavni razlozi su: a) BPMN ima standardnu notaciju i široku prihvaćenost u različitim industrijama i b) BPMN je implementiran kao tip dijagrama modeliranja u značajci Oracle Application Express (APEX) (koja se ponekad naziva i proširenje) pod nazivom Flows for APEX.Flows for APEX je open source, licenciran pod vrlo labavom licencom MIT-a. Flows for APEX omogućuje razvojnim programerima da izgrade i uvedu web-bazirane aplikacije koristeći vizualno, povuci i ispusti sučelje. ovi za APEX pružaju način automatizacije i pojednostavljenja složenih poslovnih procesa razlažući ih na niz koraka i zadataka te vodeći korisnike kroz njih.

Flows for APEX mogu uključivati različite elemente, kao što su stranice, obrasci, izvješća i dijalozi, a mogu se konfigurirati da uključuju logiku grananja, uvjetno grananje, provjeru valjanosti podataka i rukovanje pogreškama. Vizualno sučelje Flows for APEX programerima olakšava dizajn, izgradnju i testiranje poslovnih procesa, bez potrebe za opsežnim kodiranjem. Može se integrirati s drugim Oracle APEX komponentama, kao što su SQL, PL/SQL i REST servisi, za stvaranje cjelovitih aplikacija temeljenih na webu, kao i za izradu prilagođenih rješenja za specifične poslovne potrebe, kao što je upravljanje tijekovima rada, uključivanje novih klijenata i upravljanje podacima. Općenito, Flows for APEX organizacijama omogućuje način da pojednostave i automatiziraju složene poslovne procese, poboljšaju produktivnost i smanje pogreške i neučinkovitost. Izvrsni vodiči o tokovima za APEX dostupni su na this site. Dakle, kako možemo povezati APEX kao programsko okruženje s niskim kodom i Flows for APEX s ugrađenom standardnom BPML prezentacijom?

Dijagram aktivnosti (vidi sliku 17.1) prikazuje pojednostavljenu prezentaciju integriranja

Flowsa i APEX aplikacije.



Slika 17.1: Integracija Flows for APEX s APEX aplikacijom

Koraci *3. modeliranje podataka, 4. razvoj aplikacije, 6. definiranje korisničkih uloga, 7. testiranje* i *8. uklanjanje pogrešaka iz aplikacije* zajednički su za sve APEX aplikacije. Korak *1. instaliranje Flows for APEX* slično je instaliranju bilo koje druge pakirane aplikacije. Pravi noviteti su *2. modeliranje tijeka rada* i *5. povezivanje APEX aplikacije s tijek rada*.

Čitateljima koji nisu upoznati s dijagramima upravljanja poslovnim procesima (BPM) i notacijom modeliranja poslovnih procesa (BPMN) savjetuje se da potraže popularne udžbenike i prouče već spomenutu BPMN specifikaciju [4]. Kao kratki podsjetnik za sve čitatelje, sljedeći elementi BPMN dijagrama se koriste najčešće:

- Početni događaj (engl. Start Event): Označava početnu točku procesa.
- Zadatak engl. Task): Predstavlja jedinicu rada, kao što je aktivnost ili korak u procesu.
- Pristupnik (engl. Gateway): Predstavlja točku odluke u procesu, kao što je račvanje u toku ili uvjetno grananje.
- Slijedni tok (engl. Sequence Flow): Povezuje elemente u dijagramu i predstavlja tijek kontrole od jednog elementa do drugog.
- Završni događaj (engl. End Event): Označava kraj procesa.
- Plivača staza (engl. Pool/Lane): Predstavlja grupiranje povezanih zadataka, koji se koriste za definiranje uloga i odgovornosti različitih sudionika u procesu.
- Tijek poruka (engl. Message Flow): Predstavlja tijek poruka između sudionika u procesu, kao što je komunikacija između dva sustava.
- Podatkovni objekt (engl. Data Object): Predstavlja skup podataka koji se koristi ili proizvodi u procesu, kao što je račun ili evidencija kupca.
- Spremište podataka (engl. Data Store): Predstavlja spremnik za pohranu podataka koji se koriste ili proizvode u procesu, kao što je baza podataka ili datotečni sustav.

Ovi se elementi mogu kombinirati za stvaranje detaljnih i točnih modela poslovnih procesa, omogućujući organizacijama da identificiraju područja za poboljšanje i optimiziraju svoje tijekove rada.

17.1.1 Instaliraj Flows for APEX

Preuzmite zip datoteku s this URL i raspakirajte sve datoteke na svoje računalo. Slijedite upute na this link. Prije instaliranja dodatka Flows for APEX, provjerite zahtjeve. Zahtjevi su ispunjeni ako koristite svoj APEX besplatni radni prostor, besplatni/plativi OCI ili Oracle Academy račun. Dva su dodatna zadatka nakon instalacije:

- dodijeli "kreiraj posao" radnom prostoru. Vi ili administrator APEX instance morate uspostaviti vezu s bazom podataka (s OCI sučeljem, SQL Developer, SQL Plus, TOAD) i izdati naredbu. Ne možete dodijeliti ovu privilegiju unutar APEX-a.
- konfigurirajte radni prostor, zadanu (engl. default) aplikaciju i podatke o korisniku unutar aplikacije "Flows for APEX". Ovaj korak možete učiniti bilo kada kasnije kada budete razvijali vlastitu aplikaciju vođenu procesima.

Iako se ponekad naziva značajkom ili dodatkom, instalacija će stvoriti nove objekte baze podataka (tablice, prikaze) i aplikaciju pod nazivom "Flows for APEX". Ova aplikacija je administrativno sučelje koje vam omogućuje:

- za praćenje instanci tijeka rada na nadzornoj ploči (Dashboard),
- za vizualno stvaranje novih radnih procesa (Flow Management) i
- za izvršavanje tijekova rada stvaranjem instanci, dovršavanjem zadataka koji pripadaju instancama i testiranjem tijekova rada (Flow Monitor).

17.1.2 Instalirajte primjer aplikacije za tijek procesa

Uzorak za aplikaciju tijeka procesa također se nalazi u istoj zip datoteci kao Flows for APEX. Instalacija je jednostavna. Područje aplikacija su potraživanja troškova. Zaposlenik priprema zahtjev za nadoknadu troškova kojeg potvrđuje menadžer. Ako je zahtjev odbijen, zaposlenik je obaviješten. Na kraju, za odobrene zahtjeve, računovodstvo priprema plaćanje, izvršava bankovnu transakciju i postavlja status plaćanja. Ako je zahtjev plaćen, tada je instanca ovog tijeka rada završena. Primjer je vrlo sličan stvarnom procesu u bilo kojoj organizaciji. Budući da je tako dobro dokumentiran, predstavlja odličan materijal za proučavanje.

17.1.3 Zadaci za čitanje i vježbanje

Postoji odličan vodič za Flows for APEX. Počnite s this site. Na kraju ove uvodne vježbe nađite BPMN tutorial i APEX integration tutorial.

Čitanje i vježbanje će vas osnažiti da razvijete vlastitu aplikaciju vođenu procesom. Uvježbavanje svih vježbi čitatelju će vjerojatno trebati nekoliko sati. Čitatelja se snažno potiče da iskusi učenje kroz rad. Ostatak ovog poglavlja pružit će isti predložak kao i ostala poglavlja u ovom udžbeniku (od definiranja problema do primjene) s dodatnim uključivanjem BPML-a i integracije aplikacije s Flows for APEX. Međutim, opseg primjene, razvijen u Poglavlju 17 je sužen na temeljnu funkcionalnost i preskače detalje autorizacije. Uvest će vas u:

- definiranje tijeka rada u Flows for APEX
- pokretanje instance definiranog procesa unutar razvijene aplikacije
- praćenje toka instance unutar razvijene aplikacije
- izvršavanje svakog zadatka određene instance tijeka rada unutar razvijene aplikacije

Jednostavna aplikacija također može predstavljati lijep izazov za entuzijastične učenike, zar ne?

17.2 Definiranje problema

Proizvodno poduzeće srednje veličine prima upite kupaca za proizvode izrađene po narudžbi. Svaki upit može imati više povezanih dokumenata sa specifikacijama, potrebnim standardima, shemama, skicama i sl. Kupac može imati više upita. Prodavač priprema i razvrstava dokumente s upitima u tri kategorije: proizvodni, financijski i poslovni. Za procjenu jednog upita prodaja mora pripremiti više dokumenata za donositelje odluka. Voditelj proizvodnje i njegov tim ocjenjuju upit s aspekta

proizvodnje. Oni daju stručno mišljenje o sposobnosti tvrtke da proizvede traženi proizvod. Ako se traženi proizvod može proizvesti, tim daje procijenjeno trajanje procesa proizvodnje u danima. Financijski menadžer i njegov tim procjenjuju financijski aspekt upita. Daju stručno mišljenje o sposobnosti poduzeća za financiranje i procjenu očekivane dobiti. Konačnu ocjenu i odluku donosi glavni izvršni direktor i njegov tim. Razmatraju izvedivost (proizvodnu, financijsku), procjenjuju važnost kupca i donose odluku o upitu. Ovisno o sposobnosti tvrtke da proizvodi i financira poslovnu priliku kao i perspektivu suradnje, izvršni direktor odlučuje kako će odgovoriti kupcu: a) poslati poslovni prijedlog s cijenom proizvođača i očekivanim trajanjem isporuke ili b) porukom za obavještavanje kupca da proizvođač ne može poslati poslovnu ponudu.

Loše poslovne odluke (poslati poslovnu ponudu ili odbiti posao?) mogu imati katastrofalne posljedice na proizvodnu tvrtku kao što su izgubljena zarada, niska dobit, loša reputacija zbog prekoračenih rokova isporuke pa čak i stečaj. Predstavljena situacija zadaje glavobolju svim uključenima: prodavaču, voditelju proizvodnje, financijskom direktoru i glavnom izvršnom direktoru. IT odjel odgovoran je za pružanje korisničke, pouzdane i sigurne softverske podrške za opisani tijek rada. Njihov problem je razviti takvu podršku s Flows for APEX.

Da biste razvili aplikaciju, morate imati instaliran Flows for APEX. Ovo poglavlje će analizirati poslovni slučaj (opis, polustrukturirani opis, dijagram slučaja korištenja, model procesa, model podataka) i razviti aplikaciju. Naučit ćete:

- kako definirati model procesa (workflow) za ovaj poslovni slučaj,
- kako stvoriti model podataka i razviti APEX aplikaciju,
- kako povezati razvijenu aplikaciju s definiranim radnim procesom.

Imajte na umu da autorizacija korisnika neće biti implementirana jer bi to proširilo ovo poglavlje izvan razumnog napora za studente.

17.3 Slučajevi korištenja

17.3.1 Opis riječima

Četiri sudionika (aktera) uključena su u četiri slučaja korištanja. Svaki sudionik odgovoran je za jedan slučaj korištenja: prodavač priprema dokumentaciju vezanu uz upit, voditelj proizvodnje daje ocjenu proizvodnih aspekata, financijski menadžer daje ocjenu financijskih aspekata, a glavni izvršni direktor donosi odluku. Dijagram slučaja korištenja (vidi sliku17.2) pretpostavlja da će dizajner tijeka rada odlučiti hoće li proizvodnja i financijski upravitelj raditi usporedno ili slijedno (sekvencijalno). Ako rade usporedno i jedan ili oboje odluče da se zahtjevi upita ne mogu ispuniti, tada se pojavljuju neki nepotrebni napori (troškovi), ali se tijek rada može dovršiti brže. Ako se dizajner tijeka rada odluči za sekvencijalni pristup, tijek rada će oduzeti više vremena, ali će se pojaviti manje nepotrebnog napora.

17.3.2 Polustrukturirani opis

UML predlaže da svaki slučaj korištenja ima i polustrukturirani opis. Četiri različita polustrukturirana opisa navedena su u Tablicama 17.1, 17.2, 17.3 i 17.4.

17.3.3 Dijagram slučaja korištenja

Dijagram slučajeva korištenja prikazati i generalizaciju. U zadanom kontekstu zaposlenik ima četiri posebne instance (različiti menadžeri) i svaki je menadžer povezan s jednim slučajem korištenja. Sva četiri slučaja korištenja prikazana su na slici17.2.

Tablica 17.1: Opis slučaja korištnja: priprema dokumentacije za upit

Ključna riječ	**Vrijednost**
ID:	*ch17-01*
Naziv:	*Pripremi dokumentaciju za upit*
Opis:	*Prodavač kreira novi upit i dodaje povezane dokumente*
Primarni učesnik:	*Prodavač*
Preduvjeti:	Kupac je već poslao sve dokumente vezane uz upit
Uvjeti po završetku:	-
Glavni	-
Uspješni scenarij:	1. dodaj novi upit 2. dodaj dokumente vezane uz određeni upit 3. potvrdi s gumbom za potvrdu, ili otkaži s gumbom za poništavanje
Proširenja:	-
Učestalost korištenja:	*Približno 250 godišnje, prosječno 5 tjedno.*
Status:	[Status razvoja]
Vlasnik:	Prodavač
Prioritet:	*visok*

Tablica 17.2: Opis slučaja korištenja: procjena upita s aspekta proizvodnje

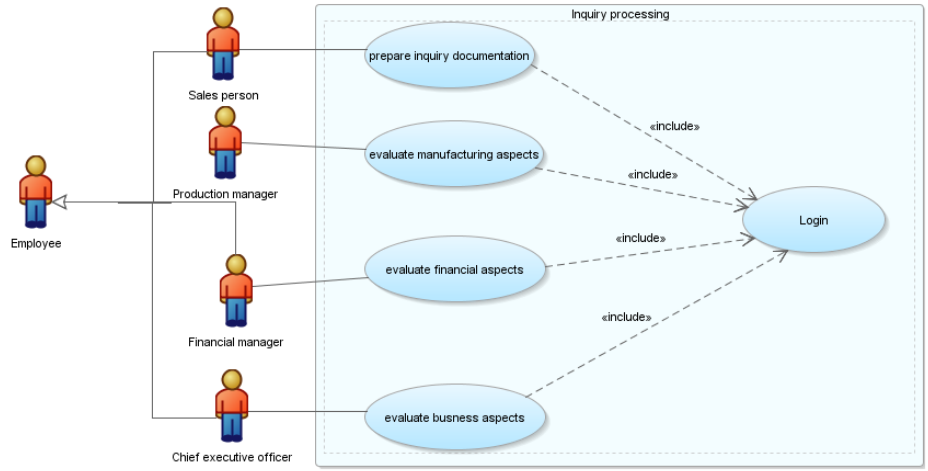
Ključna riječ	**Vrijednost**
ID:	*ch17-02*
Naziv:	*Procijeni upit s aspekta proizvodnje*
Opis:	*Voditelj proizvodnje i njegov/njezin tim daju stručno mišljenje ima li tvrtka sposobnost proizvodnje proizvoda iz upita i ako ima, koliko bi (u danima) trajala proizvodnja određene količine. Obrazloženje mišljenja je obavezno.*
Primarni učesnik:	*Voditelj proizvodnje*
Preduvjeti:	Dokumentacija o upitu je pripremljena
Uvjeti po završetku:	-
Glavni	-
Uspješni scenarij:	1. unesi mišljenje 2. potvrdi s gumbom za potvrdu, ili otkaži s gumbom za poništavanje
Proširenja:	-
Učestalost korištenja:	*Približno 250 godišnje, prosječno 5 tjedno.*
Status:	[Status razvoja]
Vlasnik:	Voditelj proizvodnje
Prioritet:	*visok*

Tablica 17.3: Opis slučaja korištenja: procjena financijskih aspekata upita.

Ključna riječ	**Vrijednost**
ID:	*ch17-03*
Naziv:	*Procijeni financijske aspekte upita*
Opis:	*Financijski menadžer i njegov/njezin tim daju stručno mišljenje je li tvrtka sposobna financirati proizvodnju proizvoda iz upita i ako jest, kolika bi bila očekivana dobit. Obrazloženje mišljenja je obavezno.*
Primarni učesnik:	*Financijski menadžer*
Preduvjeti:	Dokumentacija o upitu je pripremljena
Uvjeti po završetku:	-
Glavni	-
Uspješni scenarij:	1. unesi mišljenje 2. potvrdi s gumbom za potvrdu, ili otkaži s gumbom za poništavanje
Proširenja:	-
Učestalost korištenja:	*Približno 250 godišnje, prosječno 5 tjedno.*
Status:	[Status razvoja]
Vlasnik:	Financijski menadžer
Prioritet:	*visok*

Tablica 17.4: Opis slučaja korištenja: procjena poslovnih aspekata upita

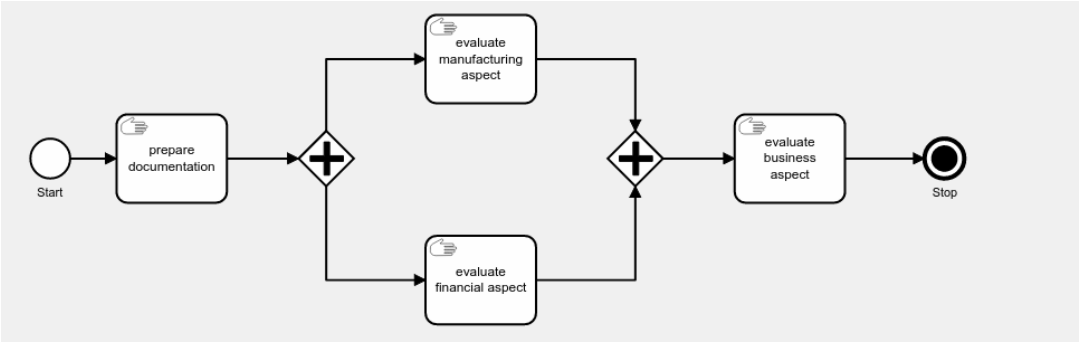
Ključna riječ	**Vrijednost**
ID:	*ch17-04*
Title:	*Procijeni poslovne aspekte upita*
Opis:	*Glavni izvršni direktor i njegov/njezin tim donose odluku o tome hoće li kupcu uputiti poslovnu ponudu, ili ne. Obrazloženje mišljenja je obvezno i zahtijeva ga Upravni odbor.*
Primarni učesnik:	*Glavni izvršni direktor*
Preduvjeti:	Slučajevi korištenja 17.1, 17.2 i 17.3 su uspješno završili.
Uvjeti po završetku:	-
Glavni	-
Uspješni scenarij:	1. unesi mišljenje, 2. potvrdi s gumbom za potvrdu, ili otkaži s gumbom za poništavanje.
Proširenja:	-
Učestalost korištenja:	*Približno 250 godišnje, prosječno 5 tjedno.*
Status:	[Status razvoja]
Vlasnik:	Glavni izvršni direktor
Prioritet:	*visok*



Slika 17.2: Dijagram slučaja korištenja

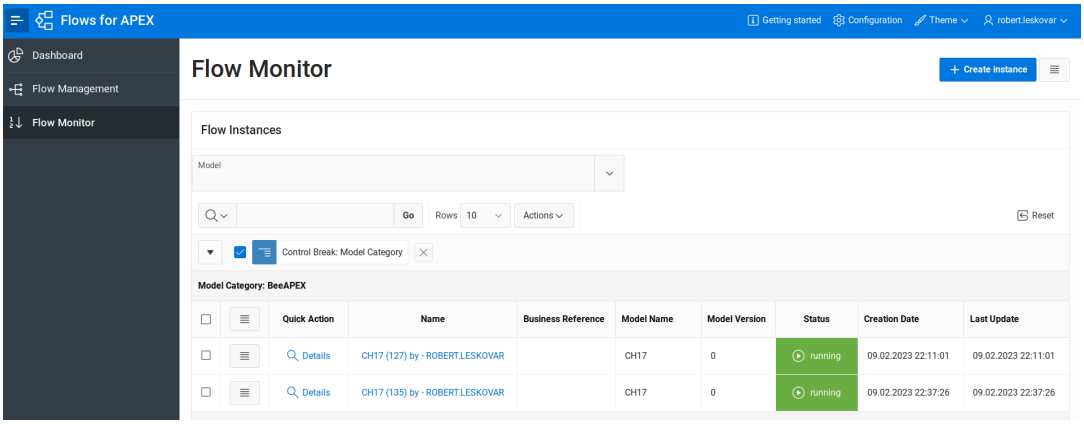
17.4 Model tijeka rada

Početni model tijeka rada prikazuje zadatak pripreme koji zatim usporednim pristupnikom razdvaja dva zadatka (proizvodnja i financijska procjena). Tijek rada nastavlja se na posljednji zadatak (procjena poslovanja) kada su oba prethodna zadatka završena (vidi Figure 17.3).



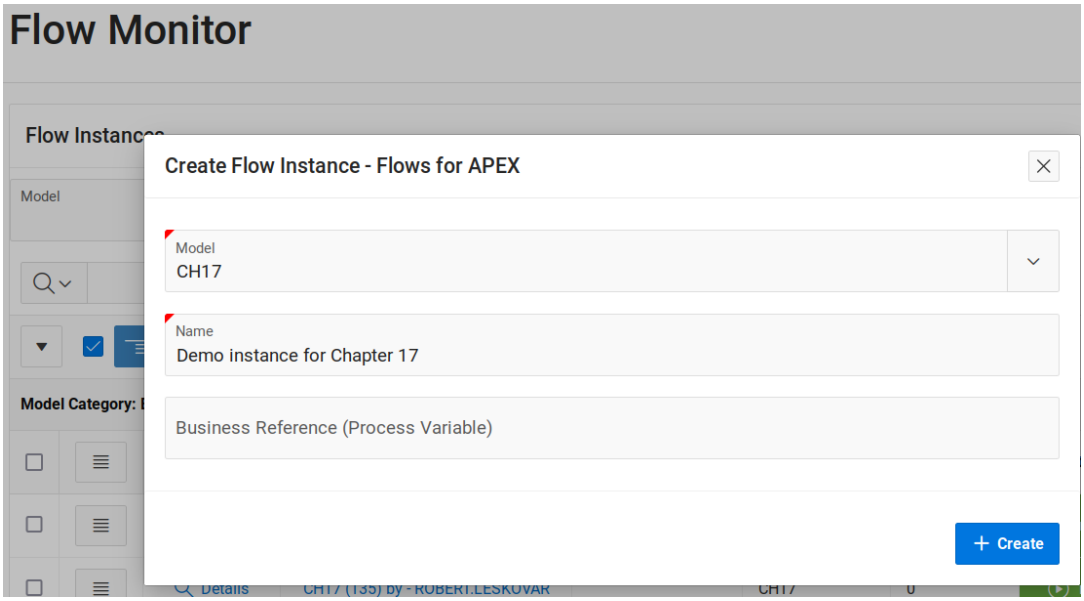
Slika 17.3: Tijek rada - obrada upita - BPMN dijagram.

Sva četiri zadatka definirana su kao "Ručni zadatak", a završni događaj (Stop) definiran je kao "Terminirajući završni događaj". Tijek rada možemo testirati odmah u Flow Monitoru (u aplikaciji "Flows for APEX"). Spremite protok kao CH17. Zatim idite na Monitor protoka. Kliknite gumb "Stvori instancu" (pogledajte sliku 17.4).



Slika 17.4: Kreiranje instanci

Odaberite model (CH17) i unesite različit naziv instance toka CH17 (pogledajte sliku 17.5).



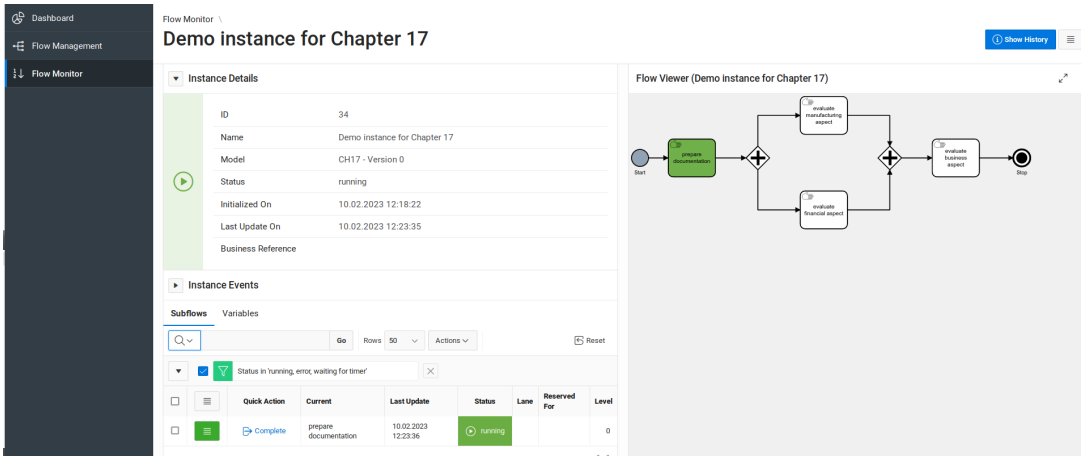
Slika 17.5: Kreiranje instance toka CH17

Pojavljuje se nova instanca. Pritisnite označenu ikonu na Slici 17.6 i odaberite "Start".



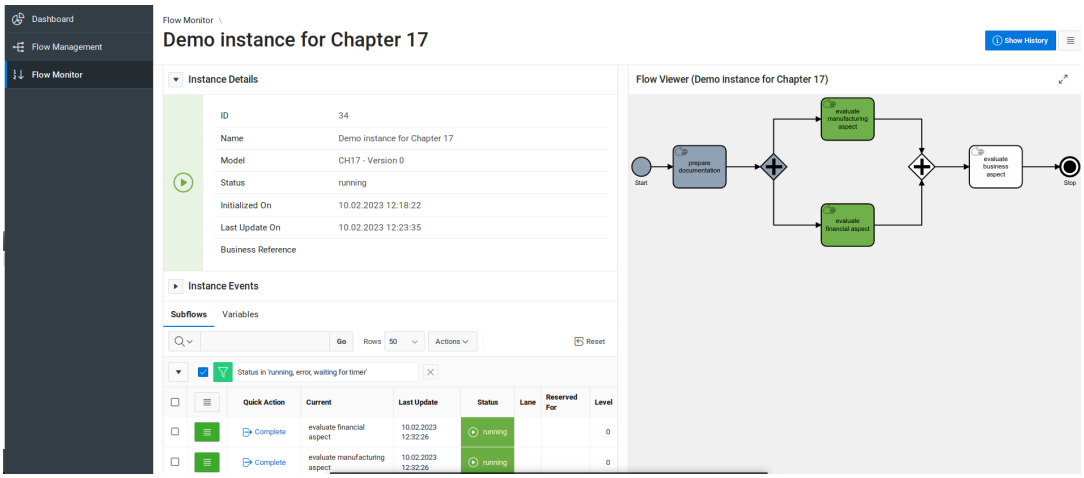
Slika 17.6: Pokretanje demo instance

Status instance promijenit će se u "running". Pritisnite "Details" za demo instancu. Postavite postavke bočnog prikaza na i kliknite "Complete" u stupcu "Quick Action" (pogledajte sliku 17.7).



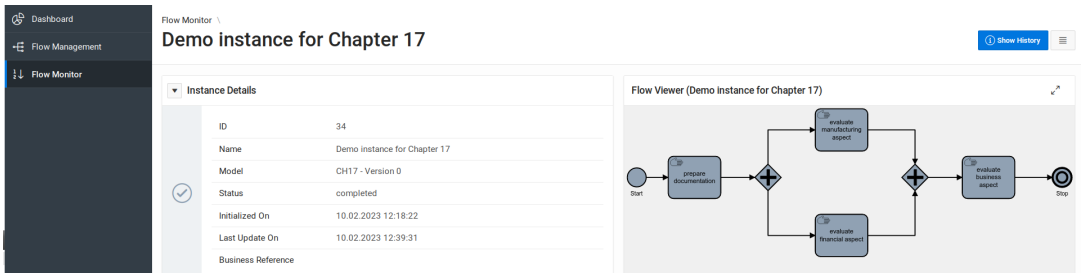
Slika 17.7: Dovršite prvi zadatak u demo instanci

Sada se tok nastavlja s paralelnim vratima i dva zadatka (vidi sliku 17.8).



Slika 17.8: Izvršavanje zadataka u paralelnim vratima

Click "Complete" in column "Quick Action" for both tasks and proceede with the final task. Repeat completion for the last task and the instance is completed (see Figure Figure Pritisnite "Complete" u stupcu "Quick Action" za oba zadatka i nastavite sa završnim zadatkom. Kad ga dovršite, i instanca je dovršena (pogledajte sliku17.9).



Slika 17.9: Dovršena demo instanca

Cjelokupna povijest instance prikazat će se klikom na "Show history". Dokazali smo da je tijek rada CH17 izvršni i da možemo modelirati podatke i sučelje aplikacije.

17.5 Model podataka

17.5.1 Opis modela podataka riječima

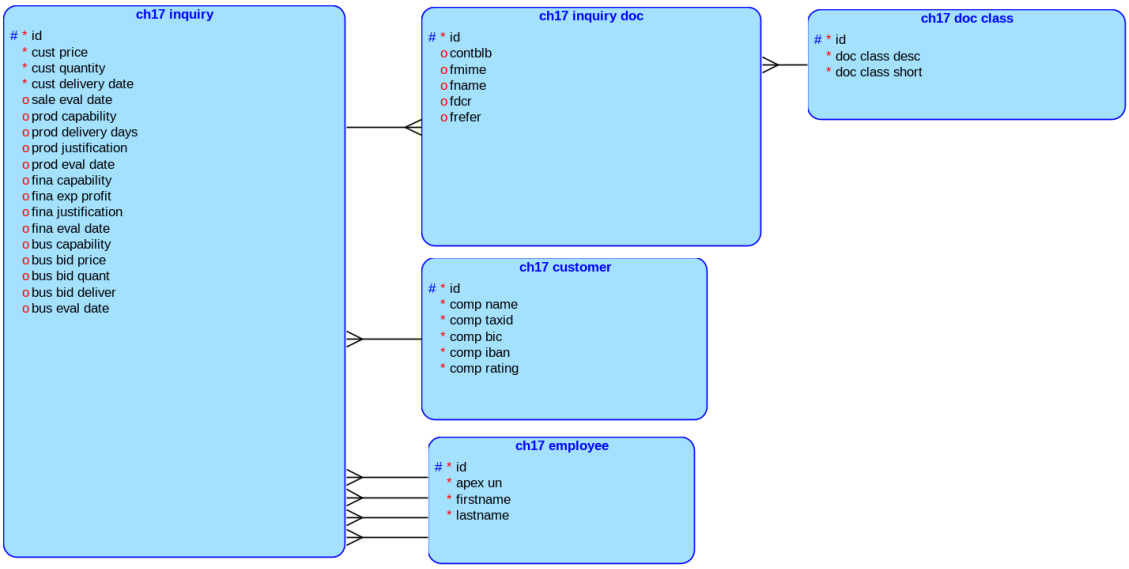
Najvažniji entitet je **ch17 inquiry - upit**. Instanca od **ch17 inquiry - upit** je povezana s jednom instancom **ch17 customer - kupac**, dok kupac može imati više upita. Svaka instanca **ch17 inquiry - upit** mora imati najmanje tri instance **ch17 document - dokument**. Svaka instanca **ch17 document - dokument** pripada jednoj instanci **ch17 document class - tip dokumenta**.

Atributi entiteta **ch17 inquiry - upit** su: ID, customer requirements - zahtjevi kupca (price per order - cijena za narudžbu, quantity - količina, delivery date - datum isporuke), proizvodni aspekti (capability - sposobnost, delivery time for requested quantity - vrijeme isporuke za traženu količinu, justification - utanačenja), financijski aspekti (finacial capability - financijska sposobnost, expected profit for required quantity - očekivani profit za traženu količinu) i poslovni aspekti (decision to take or leave opportunity - odluka da se prihvati ili odbije poslovna prilika, justification - utanačenja).

Definirat ćemo samo ograničeni skup atributa za druge entitete kako bi ovaj slučaj bio kompaktniji.

17.5.2 Logički model podataka

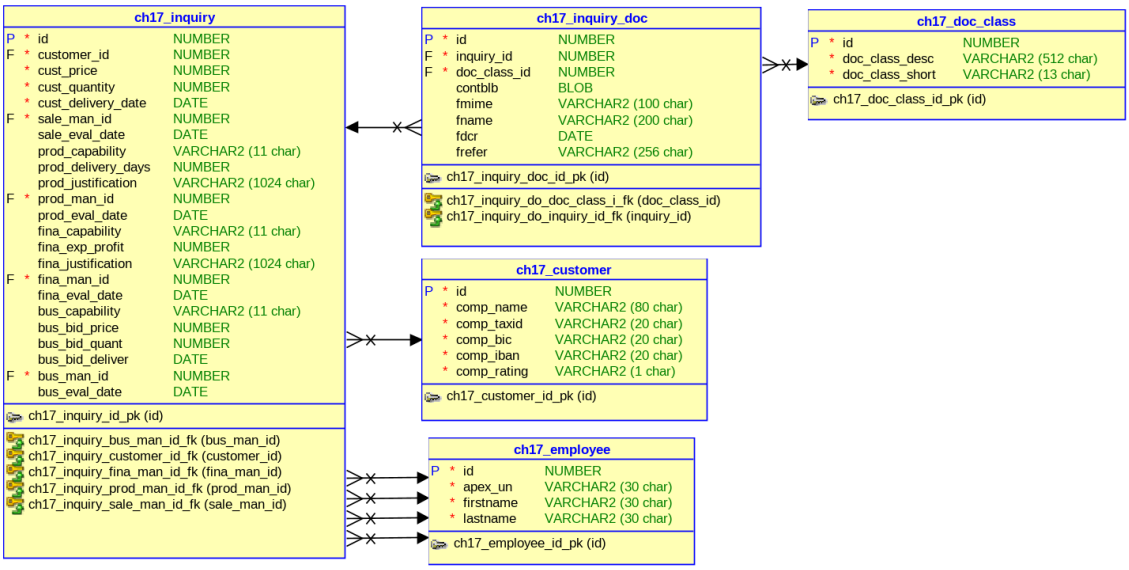
Logički model podataka je prikazan na Slici 17.10.



Slika 17.10: Logički model podataka

17.5.3 Relacijski model podataka

Relacijski model podataka je prikazan na Slici 17.11.



Slika 17.11: Relacijski model podataka

17.5.4 Quick SQL za generiranje SQL skripte

Quick SQL je praktičan alat za ubrzavanje razvoja podatkovnog modela. Može se koristiti za izradu prototipa generiranjem SQL skripte unutar APEX-a (generiranje aplikacije izvan skripte) ili za primjenu obrnutog inženjeringa na logički podatkovni model, uvozom SQL skripte u SQL Developer Data Modeler i transformacijom relacijskog u logički model. Model podataka predstavljen u ovom poglavlju može se opisati sljedećim Quick SQL-om:

```
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
```

Jedna od prednosti Quick SQL-a je njegova kompaktnost i jednostavnost. Broj redaka u Quick SQL-u je polovica SQL skripte, a broj znakova je manji od jedne trećine SQL skripte. Vrijeme i trud za učenje Quick SQL-a dobra su investicija. APEX također nudi uređivač prilagođen korisniku s automatskim dovršavanjem, no prednosti Quick SQL-a više će iskoristiti korisnici koji poznaju SQL.

17.5.5 Sekvenca i dvije pohranjene funkcije

U ovom poslovnom slučaju, zahtjevi aplikacije nas prisiljavaju da koristimo neke lako razumljive koncepte Oracle baze podataka:

- mehanizam sekvence - generira cijele brojeve definiranjem početne vrijednosti, maksimalne vrijednosti, prirasta itd. S ovim mehanizmom naša aplikacija će generirati jedinstveni broj za identifikaciju upita i sastaviti naziv instance protoka.
- pohranjene funkcije u PL/SQL jeziku uzimaju ulazne parametre i vraćaju jednu vrijednost. Naša će aplikacija koristiti dvije korisnički definirane funkcije: jednu za dobivanje veličine dokumenta pohranjenog u BLOB stupcu i jednu za brojanje dokumenata povezanih s određenim upitom.

Možemo definirati sekvencu i pohranjene funkcije u SQL Workshop > SQL Command. Sekvenca ima naziv, minimalnu vrijednost, maksimalnu vrijednost, inkrement, početnu vrijednost i neka druga svojstva koja nam trenutno nisu bitna:

```
CREATE SEQUENCE   "CH17_SEQ_INQUIRY"  MINVALUE 1 MAXVALUE 999999999999
  INCREMENT BY 1 START WITH 100 CACHE 1
  NOORDER  NOCYCLE  NOKEEP  NOSCALE  GLOBAL;
```

Također ćemo definirati dvije pohranjene funkcije koje su objašnjene kao komentari unutar koda - dvije crtice na početku redaka su komentari. Prva funkcija se zove ch17_doc_bytesize:

```
create or replace function ch17_doc_bytesize(xdocid in number) return number
-- Function ch17_doc_bytesize returns the size of the BLOB, stored in contblb
-- column in the table ch17_inquiry_doc for particular row.
-- Function parameter 1 is the ID of of ch17_inquiry_doc table.
-- The length of contblb column is provided by another system level PL/SQL
-- function called dbms_lob.getlength.
is
wdoc_size number := 0;
wblob blob;
Begin
  select contblb into wblob from ch17_inquiry_doc where id = xdocid;
  wdoc_size := dbms_lob.getlength(wblob);
  RETURN  wdoc_size;
END ch17_doc_bytesize;
```

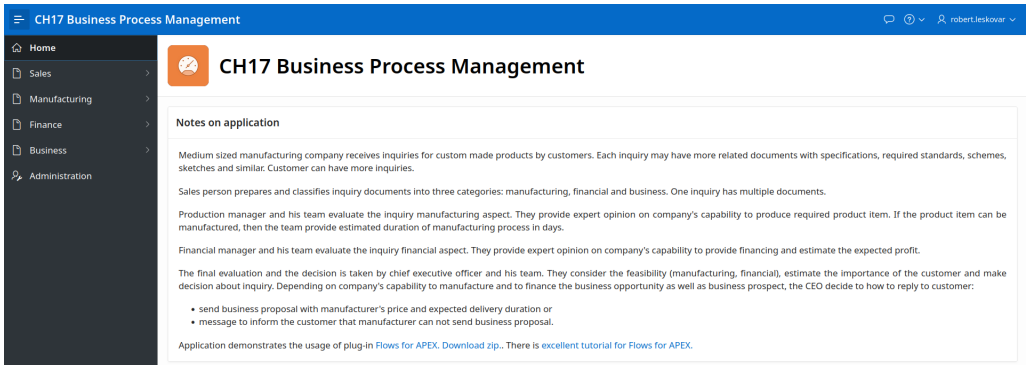
Druga funkcija se zove ch17_count_docs:

```
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 Aplikacijska sučelja

Predlažemo čitatelju da instalira upakiranu aplikaciju za ovo poglavlje prije dizajna aplikacije. Molimo pronađite poveznicu u Dodatnom materijalu za učenje za ovo poglavlje. Navedeni brojevi stranica odnose se na pripremljenu upakiranu aplikaciju. naravno, čitatelj može razviti i koristiti druge brojeve stranica kako bi pružio istu funkcionalnost kao što je opisano u ovom odjeljku.

Aplikacijsko sučelje pruža funkcionalnost za četiri učesnika: voditelja prodaje, voditelja proizvodnje, financijskog menadžera i glavnog izvršnog menadžera. Prvo je generirano pomoću čarobnjaka i uključenih svih značajki (progresivna, web aplikacija, stranica s informacijama, kontrola pristupa, izvješćivanje o aktivnostima, opcije konfiguracije, povratne informacije i odabir stila teme, ukupno 21 stranica). Prema zadanim postavkama također se generira stranica za prijavu. Početna stranica sadrži čisti HTML za opis poslovne situacije. Pruža poveznice na potrebne dodatke i upute (pogledajte sliku 17.12, stranica 1 upakirane aplikacije).



Slika 17.12: Početna stranica aplikacije "CH17 Business Process Management - Upravljanje poslovnim procesom"..

Neke liste vrijednosti (List of Value - LOV i dodaci su postavljeni ili su uključivali prethodni razvoj aplikacije u užem smislu.

17.6.1 Popis vrijednosti u dijeljenim komponentama

U ovoj aplikaciji definirani su sljedeći popisi vrijednosti:

- CH17_LOV_CAPABILITY: tip je statičan, prikazane vrijednosti su (Da, Ne, Uvjetno, Nije dostupno), povratne vrijednosti su (Da, Ne, Uvjetno, Nije dostupno). Ovaj popis vrijednosti se koristi za ispunjavanje tri različite sposobnosti (proizvodne, financijske i poslovne).
- CH17_LOV_CRATING: tip je statičan, prikazane vrijednosti su (Izvrsna ocjena, Dobra ocjena, Rubno prihvatljiva ocjena, Nije prihvatljiva ocjena), povratne vrijednosti su (A, B, C, D). Ovaj se popis vrijednosti koristi za unos ocjene korisnika.
- CH17_LOV_CUSTOMER: tip je dinamički, temeljen na tablici CH17_CUSTOMER, vrijednost za prikaz je DOC_CLASS_DESC, a povratna vrijednost je ID. Ovaj se popis vrijednosti koristi na obrascima i izvješćima kako bi se doprinijelo čitljivosti i spriječilo unošenje pogrešnog korisničkog ID-a.
- CH17_LOV_DOC_CLASS: tip je dinamički, temeljen na tablici CH17_DOC_CLASS, vrijednost za prikaz je COMP_NAME, a povratna vrijednost je ID. Ovaj se popis vrijednosti koristi na obrascima i izvješćima kako bi doprinio čitljivosti i spriječio unos pogrešne vrste dokumenta koji se odnosi na upit.
- CH17_LOV_EMPLOYEE: tip je dinamički, temeljen na SQL upitu na tablici CH17_EMPLOYEE:

```
SELECT FIRSTNAME || ' ' || LASTNAME as d,
id as r
FROM CH17_EMPLOYEE order by LASTNAME, FIRSTNAME
```

  Ovaj se popis vrijednosti koristi na obrascima i izvješćima kako bi doprinio čitljivosti i spriječio unos pogrešne osobe koja priprema dokumentaciju i procjenjuje upit. Operator spajanja koristi se za formiranje niza za prikaz s imenom, razmakom i prezimenom zaposlenika.
- CH17_LOV_INQUIRY: tip je dinamički, temeljen na SQL upitu na tablici 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;
```

  Ovaj se popis vrijednosti koristi na obrascima i izvješćima kako bi doprinio čitljivosti i

spriječio unos pogrešnog ID-a upita. Operator spajanja koristi se za formiranje prikaza dugog niza s puno informacija.

17.6.2 Dodaci u dijeljenim komponentama

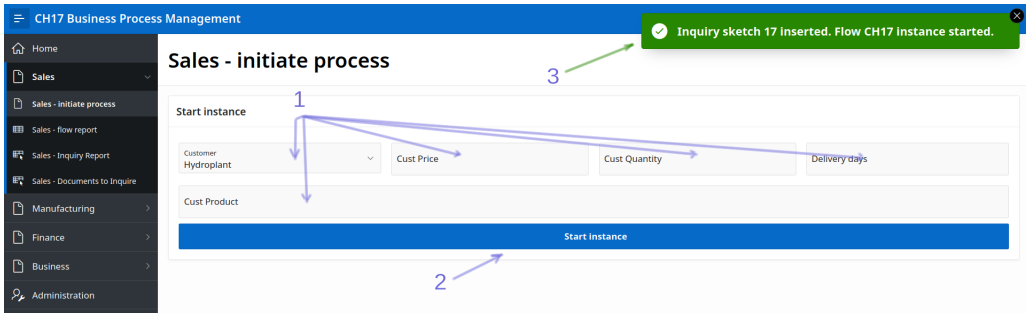
Uvoze se sljedeći dodaci (plug-ina) u Flows for APEX: Upravljanje instancom toka (Manage Flow Instance), Upravljanje korakom instance toka (Manage Flow Instance Step), Upravljanje varijablama instance toka (Manage Flow Instance Variables) i Preglednik (Viewer). Ovi dodaci su kopirani iz instaliranog dodatka Flows for APEX.

17.6.3 Voditelj prodaje

Aplikacijska sučelja omogućuju voditelju prodaje:

- pokrenuti instancu tijeka rada (pogledajte sliku 17.13, stranica 2 u aplikaciji)
- nadzirati tijek rada (pogledajte sliku 17.14, stranica 5 u aplikaciji)
- potvrditi da su svi dokumenti za daljnje procjene pohranjeni u bazi podataka (pogledajte sliku 17.15, stranica 4 u aplikaciji)
- dobiti uvid u sve upite (pogledajte sliku 17.17, stranica 6 u aplikaciji)
- dobiti uvid u sve dokumente vezane uz upite (pogledajte sliku 17.17, stranica 8 u aplikaciji)
- uučitati dokument u povezani stupac dokumenta upita (pogledajte slike17.18 and 17.20, stranica 8 and 9 u aplikaciji)

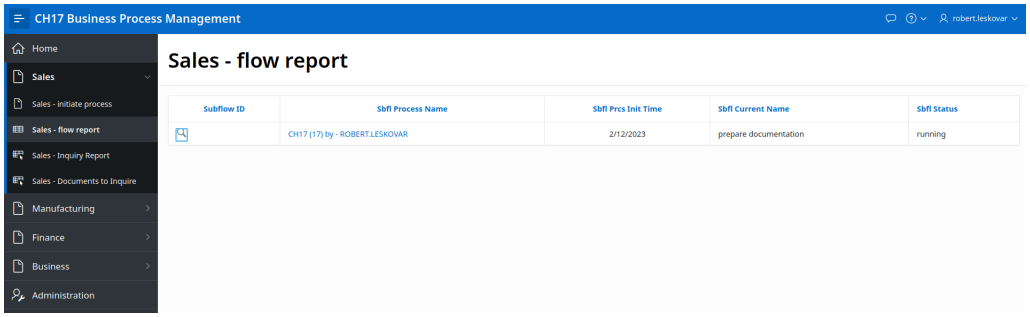
Komentari na Slici 17.13 (brojevi i strelice) prikazuju slijed radnji: 1 (korisnik unosi podatke), 2 (korisnik potvrđuje unesene podatke) i 3 (prikazuje se poruka o uspjehu ili neuspjehu, podatkovna polja se brišu za novi unos).



Slika 17.13: Prodaja - započeti proces

Osim vidljivih komponenti postoji i jedna skrivena stavka stranice (P2_WID) koja jednostavno dobiva sljedeću numeričku vrijednost iz sekvence CH17_SEQ_INQUIRY. To je bio razlog da prije dizajna stranice definiramo sekvencu CH17_SEQ_INQUIRY. Gumb "Start" izvršava dva procesa: "Dodaj upit" dodaje jedan red u tablicu CH17_INQUIRY i četiri retka u tablicu CH17_INQUIRY_DOC, dok proces "Stvori i pokreni" izvršava radnju iz Flows for APEX dodatak (pogledajte detalje u pakiranoj aplikaciji, stranica 2, Procesi). Vraća se poruka uspješnog završetka, kao što se vidi na slici 17.13.

Izvješće o tijeku prodaje prikazano je na Slici 17.14.



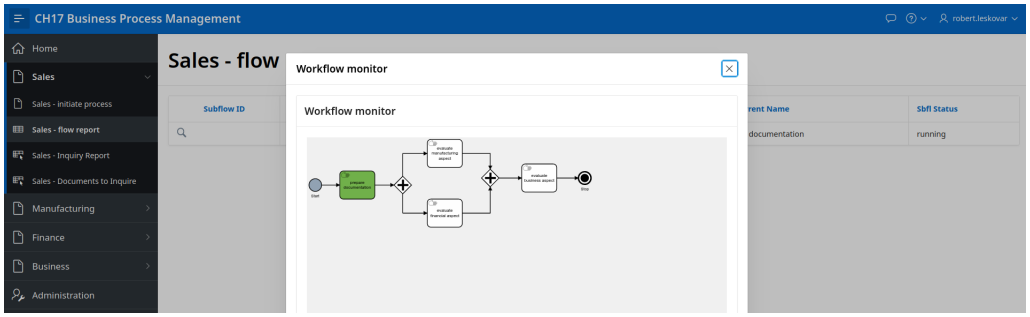
Slika 17.14: Izvješće o tijeku prodaje

Izvješće o tijeku prodaje je klasično izvješće temeljeno na SQL upitu:

```
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'
```

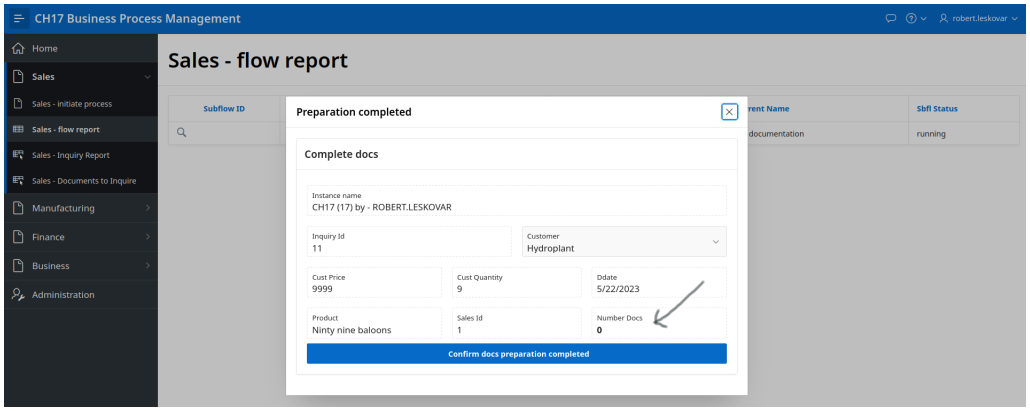
Prikazuje samo instance procesa "CH17" koje imaju trenutni naziv "Pripremi dokumentaciju". Na stranici 4 u aplikaciji nalaze se dvije veze (pogledajte sliku 17.14):

1. ikona leće u stupcu "ID podtoka" prikazuje stanje instance u dijagramu tijeka rada (pogledajte sliku 17.15, stranica 5 u aplikaciji).



Slika 17.15: Izvješće o tijeku prodaje za odabranu instancu

2. "Sbfl Process Name" stupac prikazuje modalni dijalog u kojem prodaja može potvrditi da je dokumentacija spremna (vidi sliku 17.16, stranica 7 u aplikaciji).



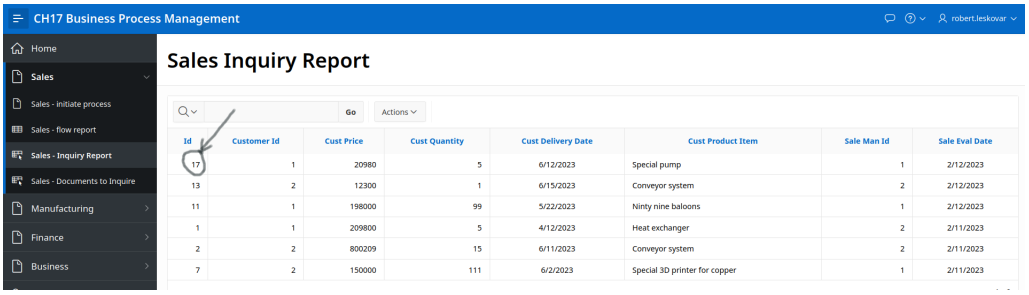
Slika 17.16: Izvješće o tijeku prodaje za odabranu instancu.

U ovom trenutku dokumentacija još nije pripremljena (učitano 0 dokumenata). Brojač je rezultat PL/SQL bloka kao izvora:

```
BEGIN
RETURN CH17_COUNT_DOCS(:P7_INQUIRY_ID);
END;
```

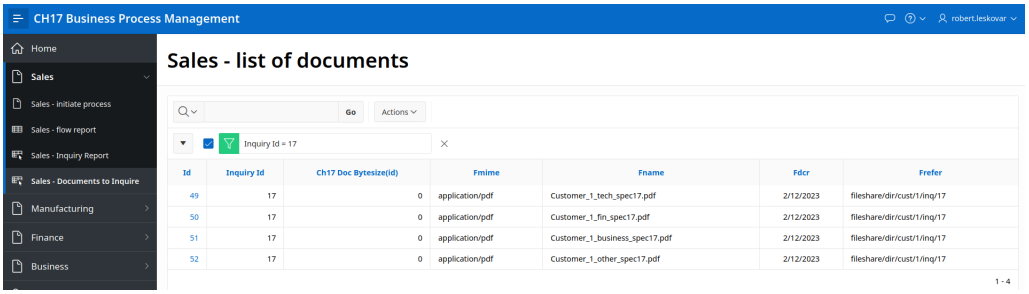
Funkcija CH17_COUNT_DOCS (pogledajte detalje u pakiranoj aplikaciji Potporni objekti ili nakon instalacije u Pregledniku objekata > Funkcije) broji sve velike binarne objekte (BLOB) pohranjene u tablici CH17_INQUIRY_DOC koji su povezani s određenim upitom. Ako je veličina BLOB-a veća od nule, tada je sadržaj pohranjen, inače postoji samo prazan blob. Prema poslovnim pravilima moraju biti pripremljena najmanje tri dokumenta (proizvodni, financijski i poslovni)

Za uvid u sve upite priprema se interaktivno izvješće za prodaju (vidi sliku 17.17, stranica 6 u aplikaciji).



Slika 17.17: Prodaja - izvješće o upitima.

Izvješće o prodajnom upitu nudi pretraživanje i akcije (odabir stupaca, preuređivanje, filtriranje, oblikovanje, grafikon, izvješćivanje, izvoz itd.). Cilj ovog izvješća je pružiti menadžeru prodaje ID upita koji se može koristiti u procesu pripreme dokumentacije (tj. 17 kao što je prikazano na Slici 17.17). Sljedeći korak za voditelja prodaje bila bi priprema tehničke, financijske, poslovne i druge specifikacije (vidi sliku 17.18, stranica 8 u aplikaciji).



Slika 17.18: Prodaja - popis dokumenata

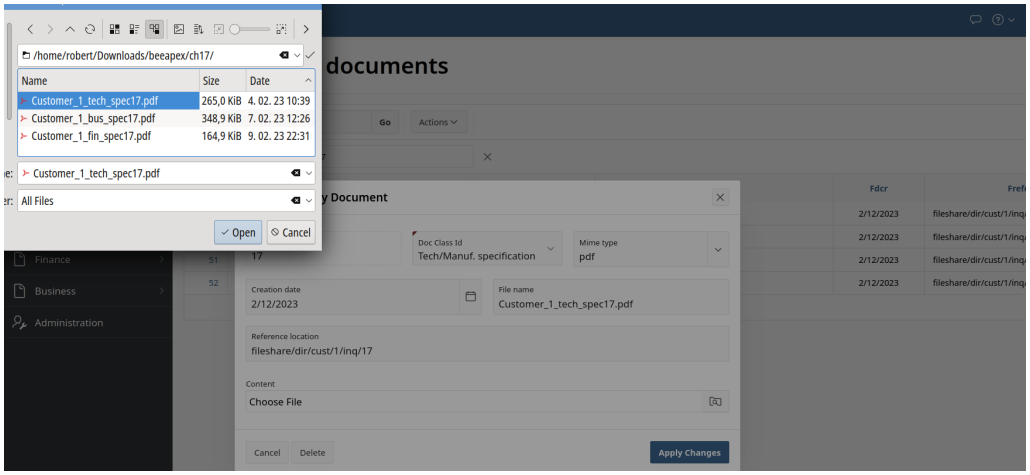
slika 17.18 prikazuje filtrirane rezultate (ID upita je 17). Nije učitan nijedan BLOB - stupac "Ch17 Doc Bytesize(id)" prikazuje sve nule. Stranica 8 je interaktivno izvješće temeljeno na sljedećem SQL upitu:

```
select ID, INQUIRY_ID,DOC_CLASS_ID, CH17_DOC_BYTESIZE(ID), CONTBLB,
       FMIME,FNAME, FDCR, FREFER
from CH17_INQUIRY_DOC
```

PL/SQL funkcija CH17_DOC_BYTESIZE vraća veličinu BLOB-a. Funkcija uzima ID CH17_INQUIRY_DOC kao ulazni parametar i definirana je na sljedeći način:

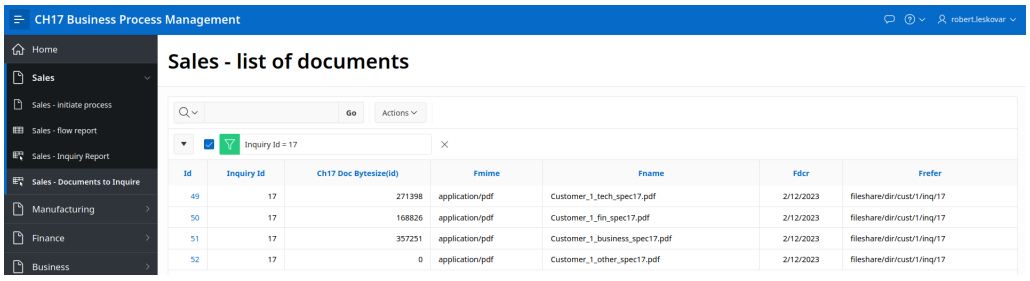
```
create or replace function CH17_DOC_BYTESIZE(xdocid in number) return number
is
  wdoc_size number := 0;
  wblob blob;
Begin
  select contblb into wblob from ch17_inquiry_doc where id = xdocid;
  wdoc_size := dbms_lob.getlength(wblob);
  RETURN  wdoc_size;
END;
```

ID tablice CH17_INQUIRY_DOC koristi se kao veza za otvaranje modalne stranice 9 u aplikaciji (pogledajte sliku 17.19).



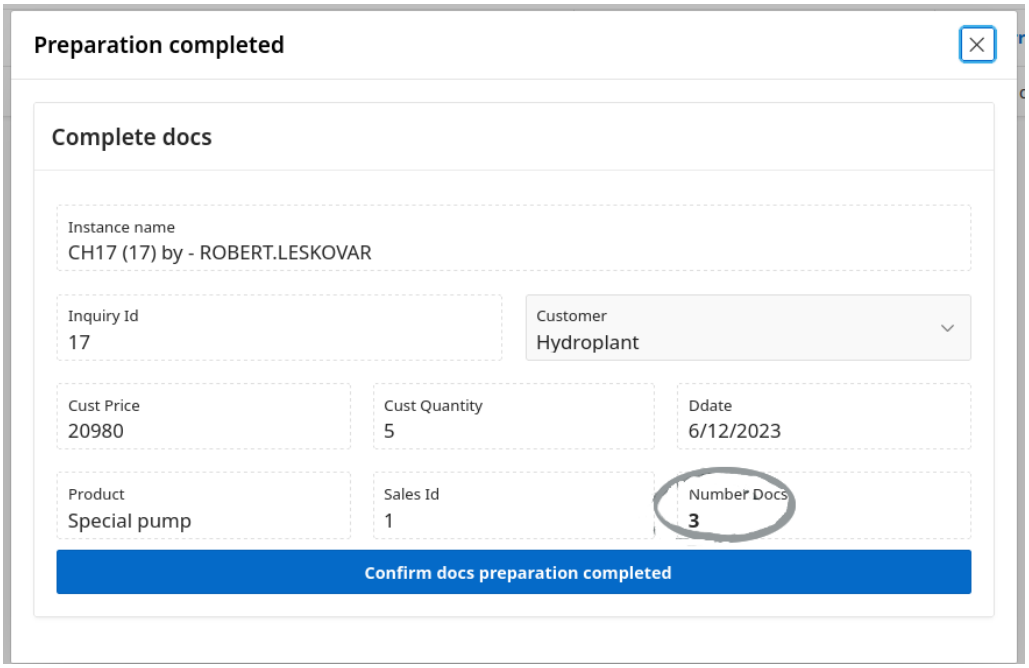
Slika 17.19: Prodaja - učitavanje dokumenta na upit

Osvježavanjem stranice 8 korisnik može biti siguran da su dokumenti učitani (vidi sliku 17.20, strana 8 u aplikaciji).



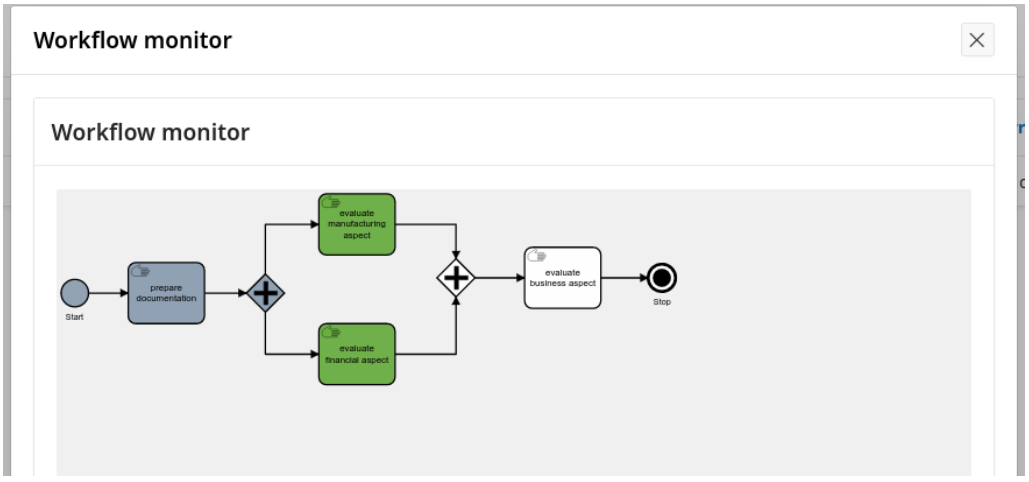
Slika 17.20: Prodaja - popis dokumenata nakon učitavanja i osvježavanja

Prodaja bi sada potvrdila da je zadatak pripreme završen. Otvaranjem izvješća o tijeku prodaje i klikom na poveznicu na naziv procesa, otvara se modalna stranica (pogledajte sliku 17.21, strana 8 u aplikaciji).



Slika 17.21: Izvješće o tijeku prodaje za odabranu instancu nakon učitavanja tri dokumenta

Klikom na "Subflow ID", voditelj prodaje dobiva vizualnu potvrdu da je zadatak pripreme dokumenta završen (vidi sliku 17.22, stranica 5 u aplikaciji).



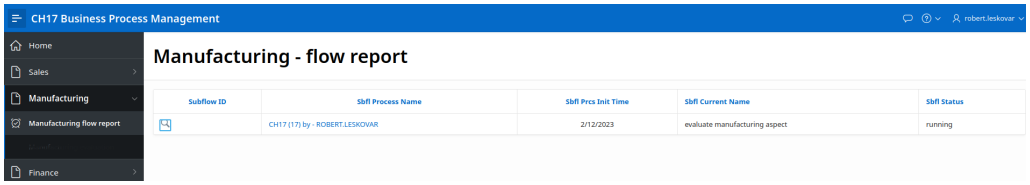
Slika 17.22: Izvješće o tijeku prodaje za odabranu instancu nakon učitavanja tri dokumenta

17.6.4 Voditelj proizvodnje

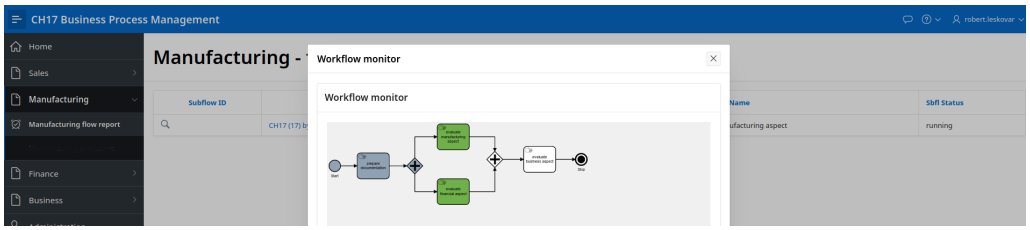
Voditelj proizvodnje može:

- pratiti tijek rada (izvješće na Slici 17.23, stranica 20 u aplikaciji i dijagram na Slici 17.24, stranica 5 u aplikaciji)
- unijeti ocjenu proizvodnje (slika 17.25, stranica 21 u aplikaciji)

Klikom na "Izvješće o tijeku proizvodnje" voditelj proizvodnje može pratiti neprocijenjene upite (tj. izvješće na Slici 17.23, stranica 20 u aplikaciji).

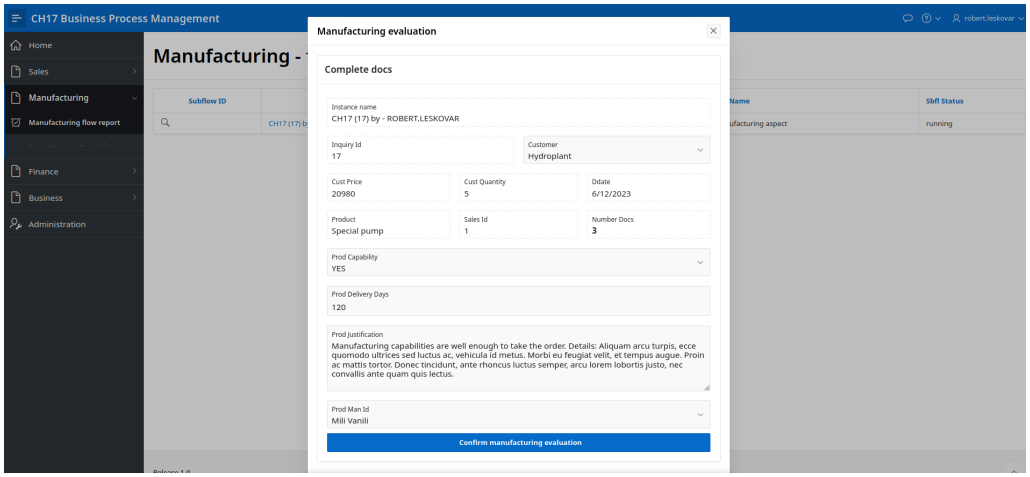


Slika 17.23: Izvješće o tijeku proizvodnje.



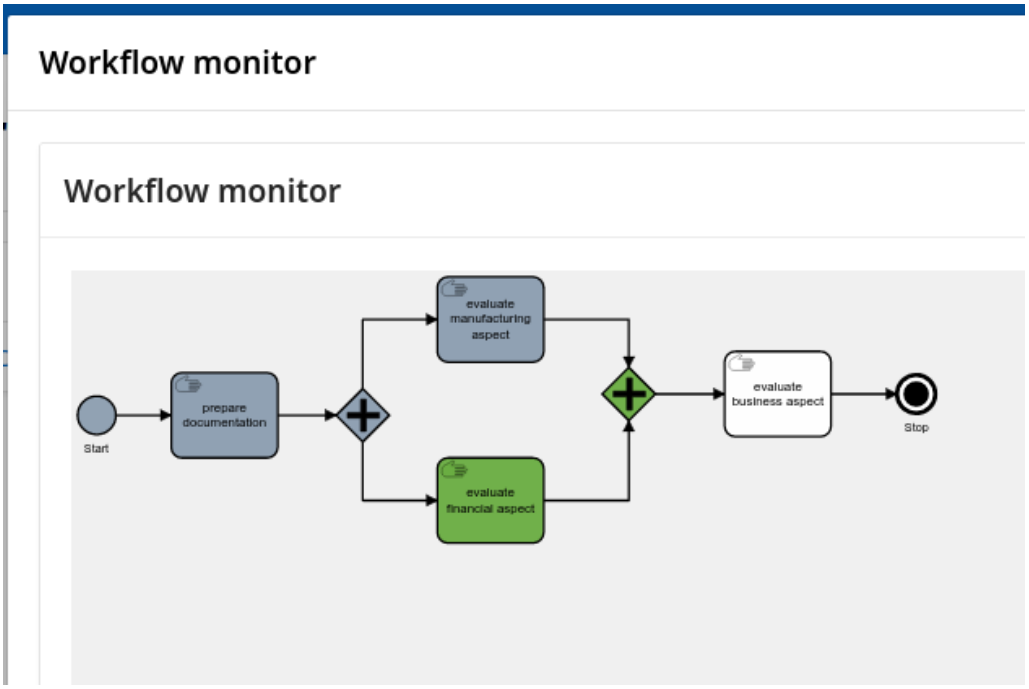
Slika 17.24: Proizvodnja - BPMN dijagram - stanje instance

Odabir ikone pod "Subflow ID" otvara stranicu 5 (pogledajte sliku 17.24) koji prikazuje instancu toka CH17. Instanca je u paralelnim vratima što znači da oba zadatka (proizvodnja i financijska procjena) ne moraju biti završena prije nego što započne sljedeći zadatak (procjena poslovanja). Procjena proizvodnje može se unijeti klikom na vezu ""Sbfl naziv procesa" na stranici 20 (pogledajte sliku 17.25, stranica 21 u aplikaciji).



Slika 17.25: Ocjenjivanje proizvodnje

Pritiskom na gumb "Potvrdi procjenu proizvodnje" premjestit će tijek instance u sljedeće stanje, prikazano na Slici 17.26.



Slika 17.26: Procjena proizvodnje je završena, primjerak čeka na financijsku procjenu

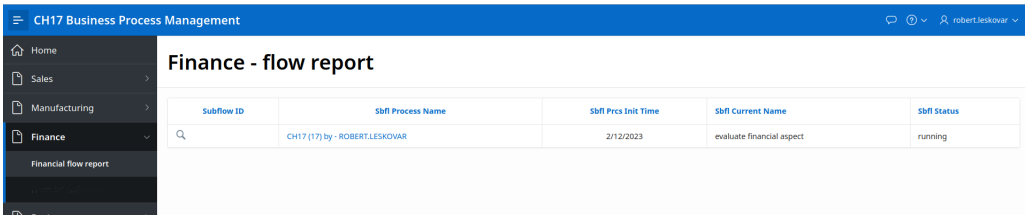
Osvježavanjem stranice 20, izvješće prikazuje "Proizvodnja nema instance tijeka".

17.6.5 Financijski menadžer

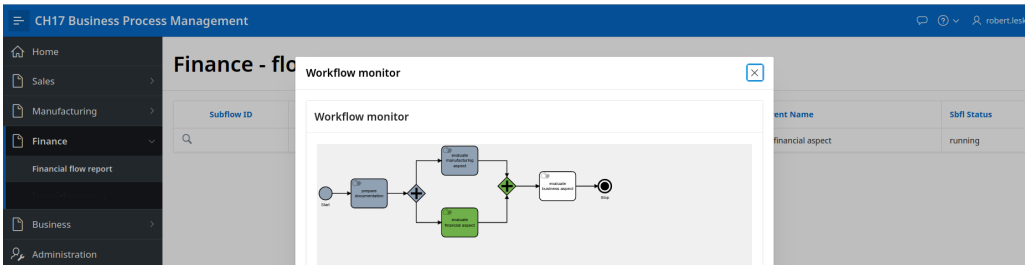
Financijski menadžer može:

- pratiti tijek rada (izvješće na slici 17.27, stranica 30 u aplikaciji i dijagram na Slici 17.28, stranica 5 u aplikaciji)
- unijeti financijsku procjenu (slika 17.29, stranica 31 u aplikaciji)

Klikom na "Izvješće o financijskom tijeku" financijski upravitelj može pratiti neprocijenjene upite (tj. izvješće na Slici 17.27, stranica 30 u aplikaciji).

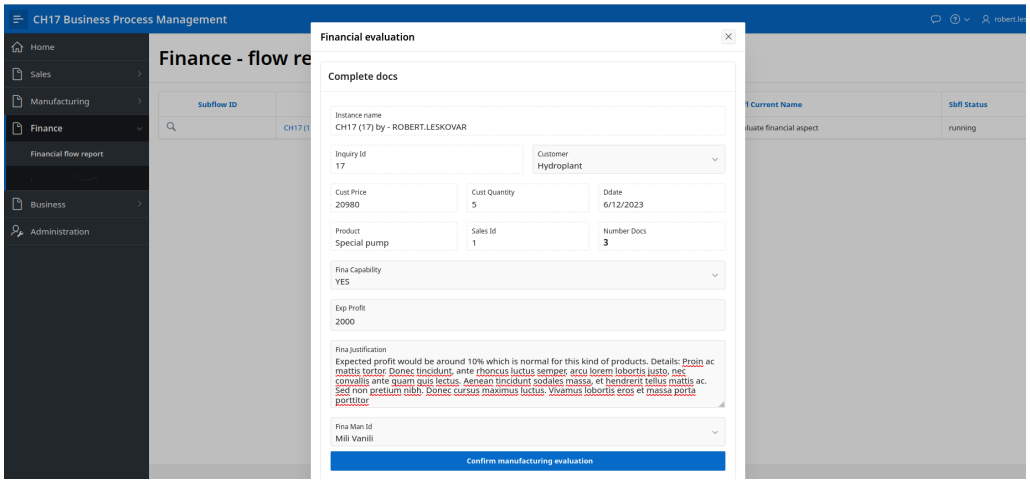


Slika 17.27: Financije - izvješće o tijeku



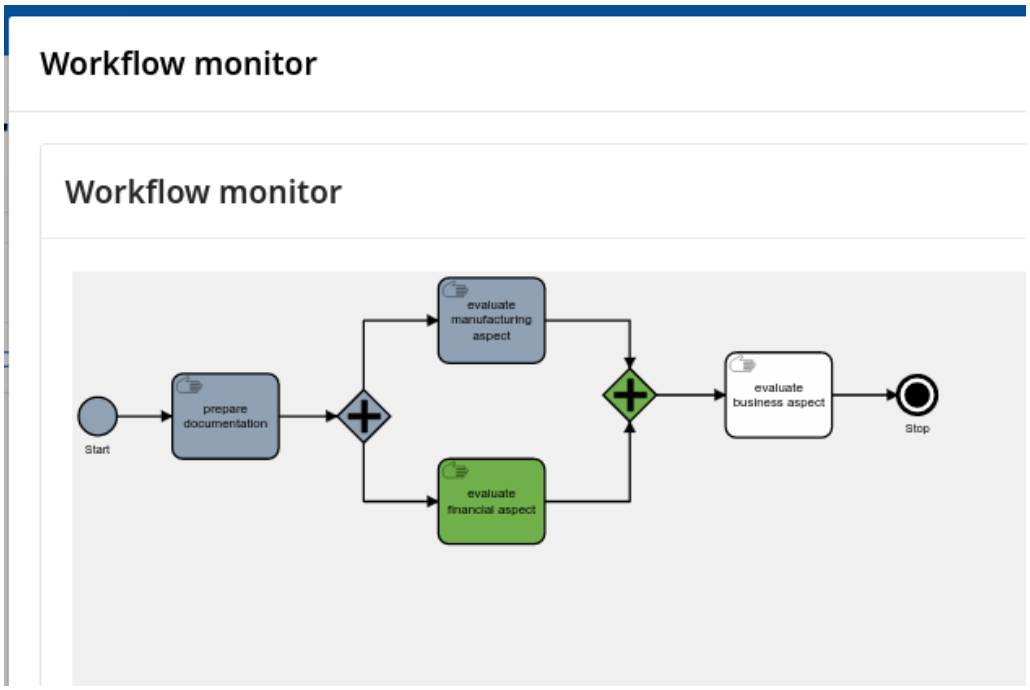
Slika 17.28: Financije - BPMN dijagram - stanje instance

Klikom na ikonu ispod "Subflow ID" aplikacija otvara stranicu 5 (pogledajte sliku 17.28) koja pokazuje instancu toka CH17. Financijsku ocjenu možete unijeti klikom na poveznicu "Sbfl Process Name" na stranici 30 (pogledajte sliku 17.29, stranica 31 u aplikaciji).



Slika 17.29: Financijska procjena

Pritiskom na gumb "Potvrdi financijsku procjenu" pomaknut će se tok instance u sljedeće stanje, prikazano na Slici 17.30.



Slika 17.30: Financijska procjena je završena, instanca čeka na procjenu poslovanja.

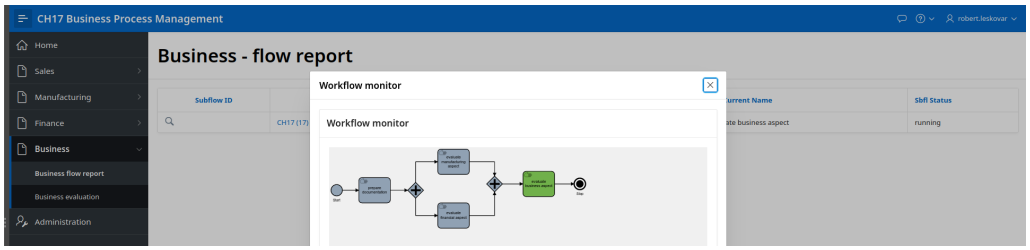
Osvježavanjem stranice 30 izvješće prikazuje "Financije nemaju instance tijeka".

17.6.6 Glavni izvršni direktor - voditelj poslovanja

Voditelj poslovanja može:

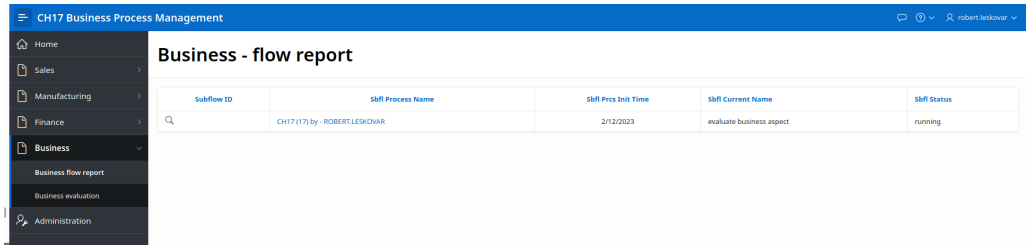
- pratiti tijek rada (izvješće na Slici 17.32, stranica 40 u aplikaciji i dijagram na Slici 17.31, stranica 5 u aplikaciji)

- unijeti ocjenu poslovanja (slika 17.33, stranica 41 u aplikaciji)



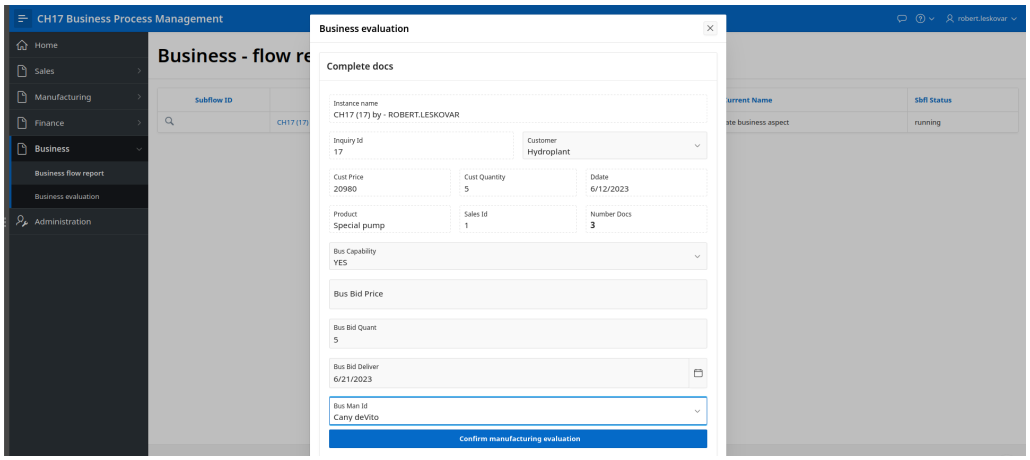
Slika 17.31: Financije - BPMN dijagram - stanje instance.

Klikom na "Izvješće o poslovnom tijeku" CEO može pratiti neprocijenjene upite (tj. izvješće na slici 17.32, stranica 40 u aplikaciji).



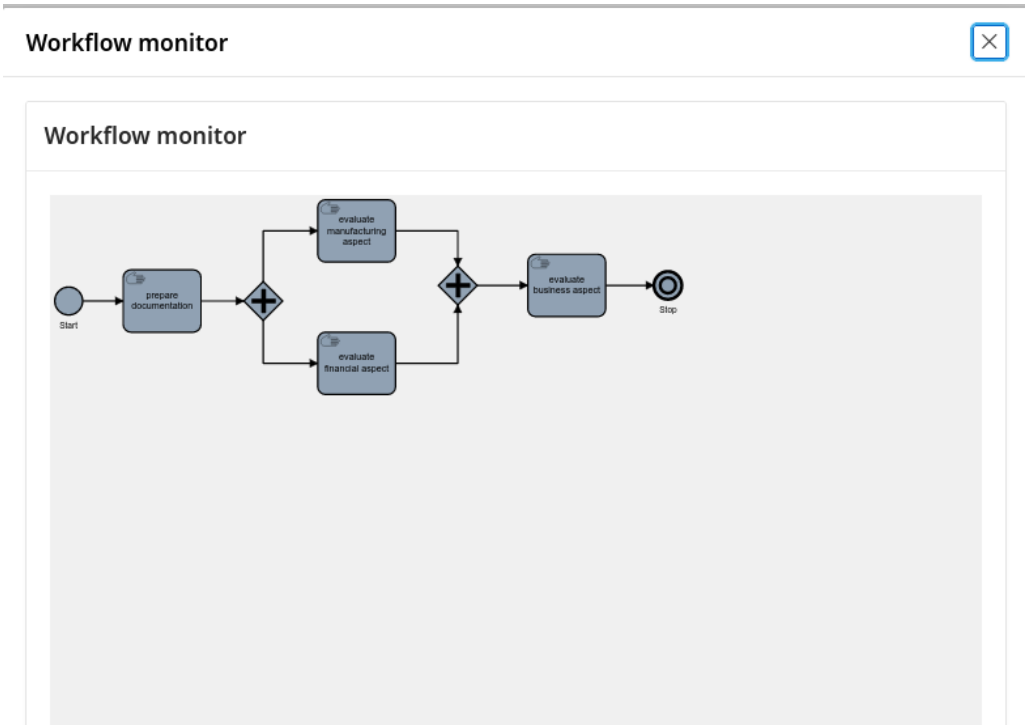
Slika 17.32: Izvještaj o tijeku poslovanja

Klikom na ikonu leće pod "Subflow ID" otvorit će se stranica 5 (pogledajte sliku 17.31) koja prikazuje instancu toka CH17. Financijska procjena može se unijeti klikom na poveznicu ""Sbfl Process Name" na stranici 40 (pogledajte sliku 17.33, stranica 41 u aplikaciji).



Slika 17.33: Procjena poslovanja

Pritiskom na gumb "Potvrdi procjenu poslovanja" pomaknut će se instanca na kraj tijeka rada, prikazan na slici 17.34.



Slika 17.34: Poslovna procjena završena, instanca prekinuta

Osvježavanjem stranice 40, izvješće prikazuje "Poslovanje nema instance toka".

17.7 Povezivanje aplikacije s Flows for APEX

Aplikacija, razvijena u poglavlju 17 primijenjene dvije funkcije koje pruža Flows for APEX:

- pokretanje instancu na stranici 2,
- prikaz stanja instance na stranici 5,
- pomak instance na sljedeći zadatak na stranicama 7, 21, 31, 41.

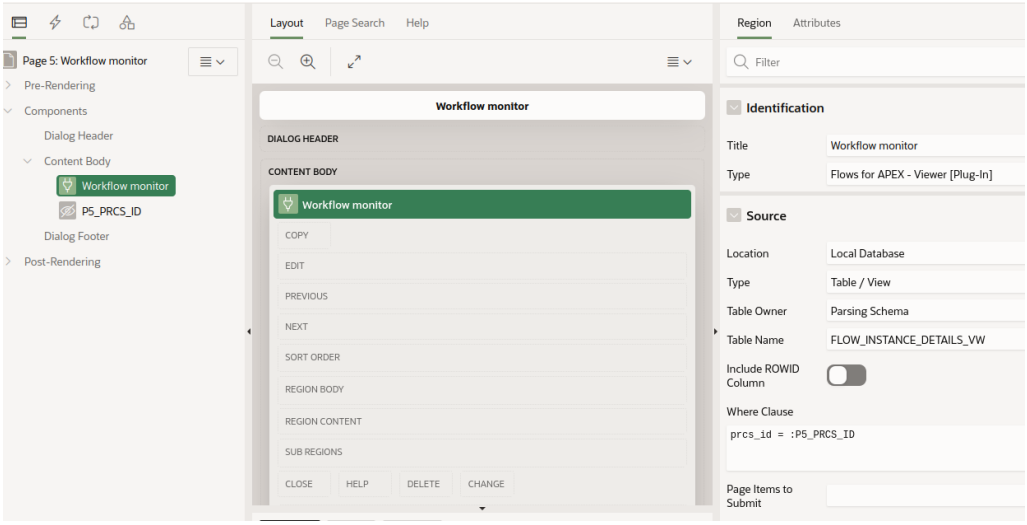
Za pokretanje instance potreban je proces na stranici. U području identifikacije procesa upišite: "Flows for APEX - Managing Flow Instance [Plug-In]". Radnja bi trebala biti "Stvori i pokreni". Naziv instance Flow postavljamo na:

```
CH17 (&P2_WID.) by - &APP_USER.
```

Upamtite da stavka stranice P2_WID sadrži jedinstveni broj, osiguran nizom. Dakle, naziv je spojeni niz, sastavljen od fiksnog niza "CH17", razmaka, otvorene zagrade, jedinstvenog sekvencijskog broja, zatvorene zagrade, razmaka, fiksnog niza "by - " i trenutnog korisnika.

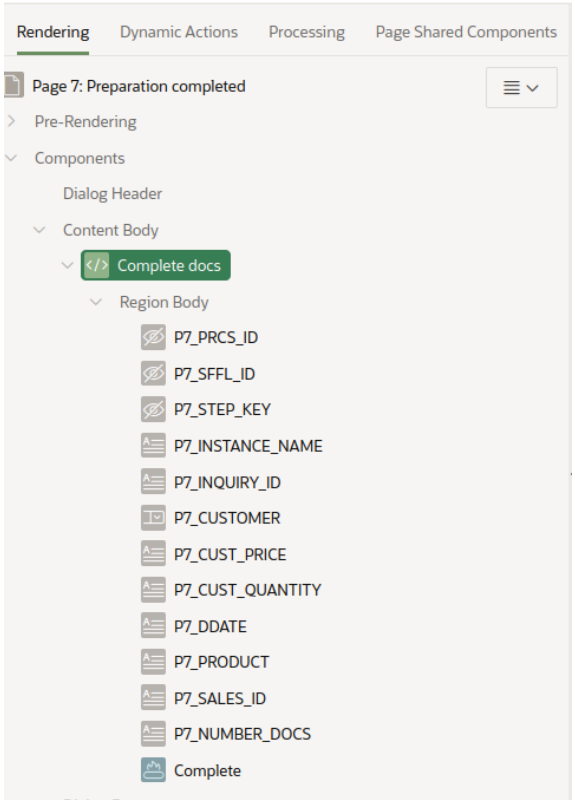
Odabir protoka (Select Flow) koristi "statični tekst", a statični tekst je "CH17". Poruka o uspjehu je "Flow CH17 instance started." a poruka o pogrešci je "Flow CH17 instance NOT started."

Za prikaz stanja instance na stranici 5 koristi se čarobnjak stranice za stranicu s dodacima. Stranica sadrži dodatak za preglednik i skrivenu stavku - P5_PRCS_ID. Pogledajte postavke na slici 17.35.



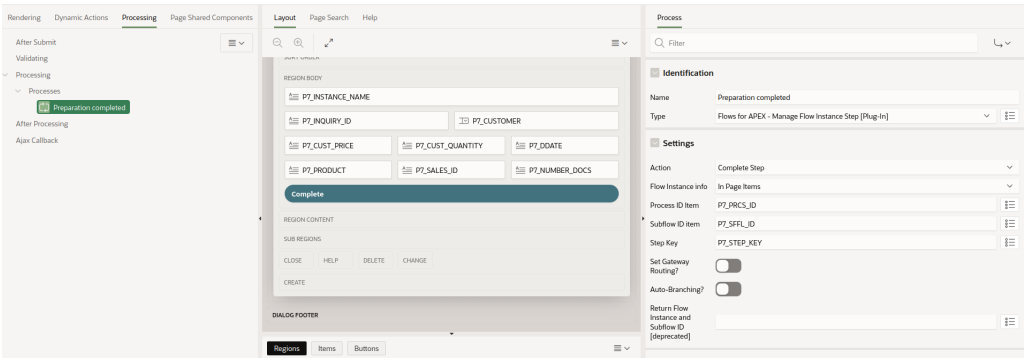
Slika 17.35: Prikaz stanja instance na BPMN dijagramu

Za pomicanje instance naprijed (tj. stranice 7, 21, 31, 41) definirajte stavke stranice kao što je prikazano na slici 17.36.



Slika 17.36: Postavljanje stavki stranice

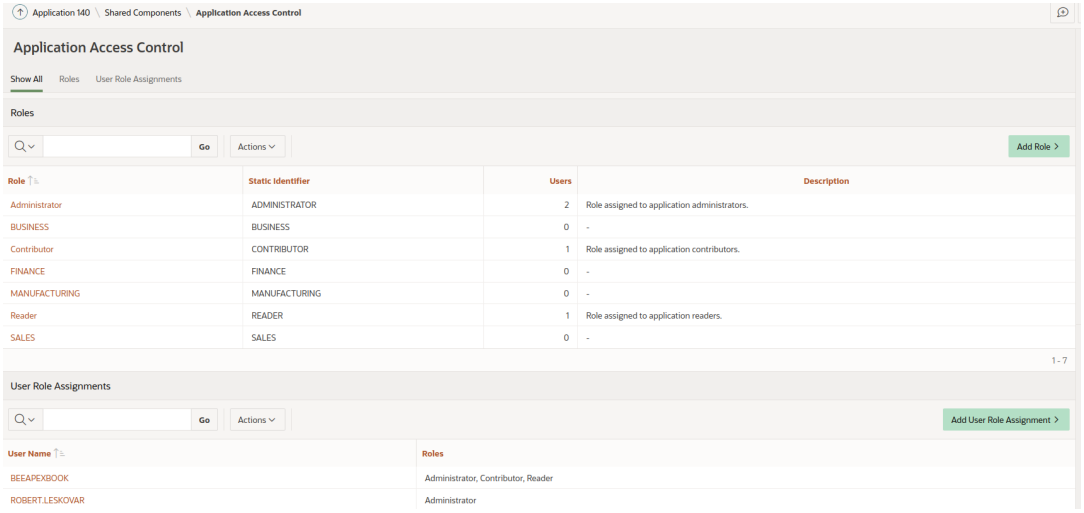
Samo prve četiri stavke stranice (od P7_PRCS_ID do P7_INSTANCE_NAME) relevantne su za integraciju Flows for Apex-a. Stvorite novi proces koji se pokreće gumbom. Proces je definiran kao dodatak "Manage Flow Instance Step". Set Action, Process ID Item, Subflow ID item i Step Key kao što je prikazano na slici 17.37.



Slika 17.37: Dovršavanje koraka u Flows for Apex

17.8 Definiranje korisničke uloge

Korisničke uloge mogu se definirati u Kontroli pristupa aplikaciji (Application Access Control) u Dijeljenim komponentama (Shared Components) (tj. Slika 17.38).



Slika 17.38: Definirajte uloge i korisničke uloge u izborniku Kontrola pristupa aplikaciji

U ovom trenutku ćemo preskočiti pojedinosti i implementaciju za ovaj specifični slučaj. Za brzi pregled pogledajte poglavlje 13 ili za temeljit uvid proučite APEX dokumentaciju o autorizaciji.

17.9 Testiranje i ispravljanje grešaka

Za početnike je primjeren jako pojednostavljen proces testiranja. Čim ste prihvatili tipkovnicu i miša za razvoj s APEX-om, na sigurnom ste putu da napravite grešku i da ćete ju moorati ispraviti. Ne bojte se pogriješiti. Izvucite pouku iz svakoga koji ste sami riješili. Podijelite znanje u otvorenoj APEX zajednici. Disciplina testiranja softvera je ogromna i možete je naučiti radeći to. Nema drugog načina. Programeri su prva linija obrane od softverskih grešaka i kvarova.

Iznenađujuće, neuspjeli projekti razvoja softvera u većini slučajeva krenuli su krivim putem prije nego što je napisan i jedan red koda. Aplikacija, razvijena za ovo poglavlje, također ima softverske pogreške. Neke od njih će pametni čitatelji otkriti i otkloniti pogreške. Neke su pogreške također duboko u samom APEX-u. Mi, pisci i programeri ovog poglavlja nadamo se da će testiranje i otklanjanje pogrešaka predstavljati ugodan izazov čitateljima.

Uklanjanje pogrešaka iz softvera ponekad je zabavno, a ponekad prokletstvo. Samo nemojte odustati prerano!

17.10 Dodatni materijal za učenje

Na raspolaganju vam je sljedeći dodatni materijal za učenje:

- izvezene pakirane aplikacije i skripte
- model tijeka rada pod nazivom CH17
- video vodiči.

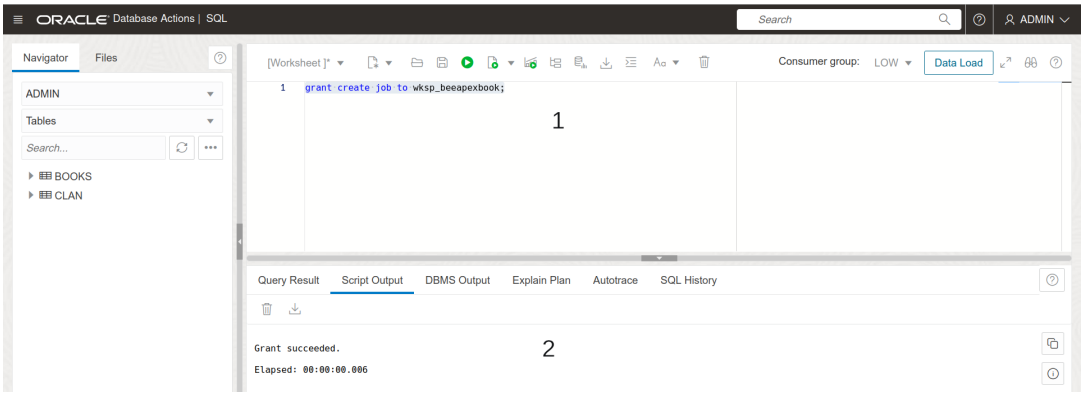
Svi dodatni materijali za učenje dostupni na javna stranica projekta BeeAPEX. Prijavite se kao gost (zaporka nije potrebna). Pronađite udžbenik u odjeljku Books, skripte u mapi Part 2 > Chapter17 u odjeljku Scripts i video vodiče u Collection of video guides. Materijal za kratke tečajeve nalazi se u odjeljku Short courses.

17.10.1 Izvezena aplikacija

Izvezena aplikacija je zapakirana. Instalacija stvara tablice, pojmovnik, funkcije i popunjava podatke. Deinstalacija uklanja sve objekte baze podataka koji se koriste u ovoj aplikaciji.

Zapakirana aplikacija je testirana i pokrenut će se u novom radnom prostoru ako su ispunjeni sljedeći zahtjevi:

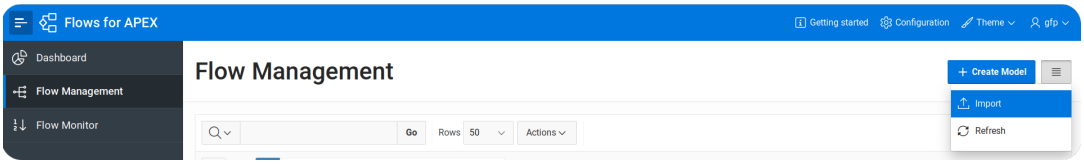
- povlastica "create job" dodijeljena je shemi koja ugošćuje novi radni prostor. Administrator baze podataka može dodijeliti ovu povlasticu naredbom "grant create job to <schema>". APEX radni prostor povezan je s objektom baze podataka koji se zove shema. Ako koristite OCI (besplatno , plaća ili pruža Oracle Academy) nego koristite web OCI sučelje (Autonomous Database > specific instance > Database action) ili unesite izravno administratorsko sučelje web baze podataka. Možete izvršavati upite i skripte. Slika 17.39 prikazuje izgled sučelja s unesenom naredbom (1) i odgovorom baze (2).



Slika 17.39: Dodjela "povlasti za stvaranje posla" radnom prostoru.

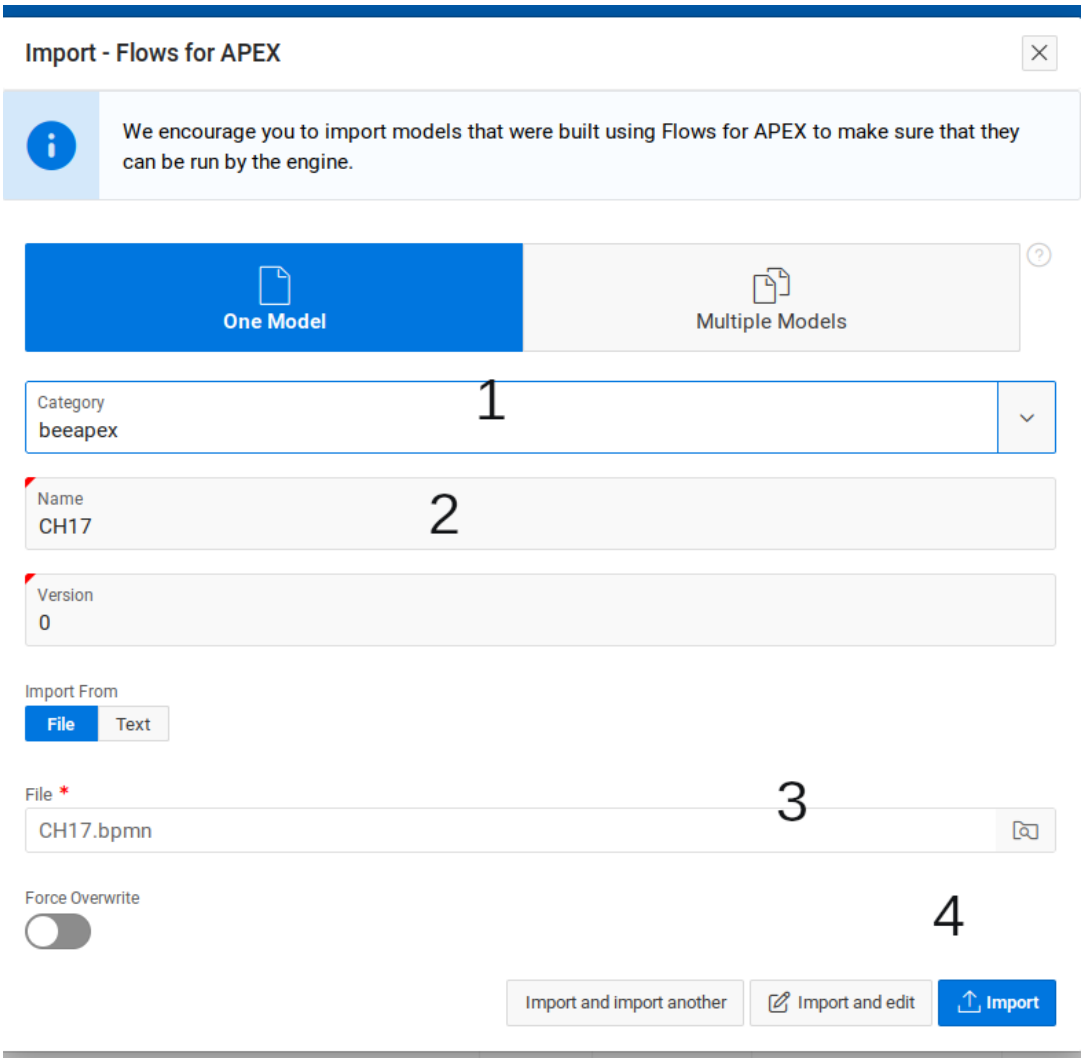
Ako koristite druge alate (SQL Developer, TOAD) postupak je uglavnom isti, osim izgleda sučelja.

- Flows for APEX moraju se instalirati u novi radni prostor (pogledajte prethodne upute).
- Model tijeka rada pod nazivom CH17 uvozi se u Flows for APEX. File *CH17.bpmn* pohranjuje se u direktoriju Flows4APEX u pakiranoj aplikaciji kao statična datoteka. Idite na Shared Components > Static Application Files i preuzmite ga na svoje lokalno računalo. Ista datoteka također je dostupna u skriptama materijala za učenje. Zatim uvezite datoteku *CH17.bpmn* s Flows for APEX. Prijavite se u aplikaciju Flows for APEX. Odaberite Upravljanje protokom i uvezite datoteku (pogledajte slike 17.40 - korak 1 i slika 17.41 - step 2).



Slika 17.40: Uvoz tijeka rada pod nazivom CH17 u Flows for APEX - korak 1

Značenje brojeva na slici 17.41 je: 1 (proizvoljna kategorija), 2 (naziv modela mora biti točno CH17), 3 (idite u svoj file explorer na CH17.bpmn) i 4 (potvrdite pritiskom na Import)



Slika 17.41: Uvoz tijeka rada pod nazivom CH17 u Flows for APEX - korak 2

- Nakon instalacije zapakirane aplikacije dodajte dodjelu korisničke uloge (Shared components > Application Access Control > Add User Role Assignment)

Ako bilo koji od gore navedenih zahtjeva nije ispunjen, uvezena aplikacija će se srušiti. Potrebno je izbrisati kolačić web preglednika (tj. Firefox: Settings > Cookies i Site Data > Manage Data) nakon što se aplikacija sruši zbog nezadovoljenih zahtjeva.

17.10.2 Video vodiči

Video vodič prikazuje početne korake u razvoju aplikacije.

17.11 Pitanja

1. Objasni što je Flows for APEX?
2. Koji se artefakti modela poslovnog procesa prikazuju na BPMN dijagramu?
3. Koje su funkcije Flows for APEX korištene u ovoj aplikaciji?

17.12 Odgovori

1. Flows for APEX je ujedno značajka, dodatak i aplikacija. Sva tri značenja su točna. Značajka podrazumijeva da govorimo u kontekstu modeliranja procesa. Plug-in se odnosi na razvoj prilagođene aplikacije. Aplikacija "Flows for APEX" je namijenjena dizajniranju tijekova rada i administriranju instanci u slučaju problema.
2. Artefakti modela poslovnih procesa prikazani BPMN dijagramom su:
   - Početni događaj (engl. Start Event): Označava početnu točku procesa.
   - Zadatak engl. Task): Predstavlja jedinicu rada, kao što je aktivnost ili korak u procesu.
   - Pristupnik (engl. Gateway): Predstavlja točku odluke u procesu, kao što je račvanje u toku ili uvjetno grananje.
   - Slijedni tok (engl. Sequence Flow): Povezuje elemente u dijagramu i predstavlja tijek kontrole od jednog elementa do drugog.
   - Završni događaj (engl. End Event): Označava kraj procesa.
   - Plivača staza (engl. Pool/Lane): Predstavlja grupiranje povezanih zadataka, koji se koriste za definiranje uloga i odgovornosti različitih sudionika u procesu.
   - Tijek poruka (engl. Message Flow): Predstavlja tijek poruka između sudionika u procesu, kao što je komunikacija između dva sustava.
   - Podatkovni objekt (engl. Data Object): Predstavlja skup podataka koji se koristi ili proizvodi u procesu, kao što je račun ili evidencija kupca.
   - Spremište podataka (engl. Data Store): Predstavlja spremnik za pohranu podataka koji se koriste ili proizvode u procesu, kao što je baza podataka ili datotečni sustav.
3. Za pokretanje instance korištena je radnja "Create and Start". Kako bismo predstavili instancu na BPMB dijagramu, čarobnjak za dodatnu strplug-in stranicu koristi komponentu Preglednik (Viewer). Za pomak naprijed na sljedeći korak u aplikaciji je korištena komponenta "manage Flow Instance Step".



18. GreenDi – Upravljanje katalogom biljaka

VJERAN STRAHONJA, DIJANA ORESKI, DARKO ANDROCEC, ANA KUTNJAK

18.1 Poslovni pogled na slučaj

Kratki pregled poslovanja GreenDi platforme opisan je u poglavlju 15. Poglavlje 18 odnosi se na dio platforme posvećen razmjeni biljaka i sjemena. U ovoj razmjeni mogu sudjelovati svi registrirani korisnici platforme, izlažući vlastitu ponudu ili odgovarajući na tuđu ponudu. Istovremeno, sustav bi trebao omogućiti pretraživanje ponuda različitim filterima, sortiranjem stupaca i sličnim funkcionalnostima. Također treba omogućiti komunikaciju između ponuditelja i ostalih korisnika vezano uz konkretnu ponudu. Komunikacija je u obliku chata.

18.2 Opis problema

Sljedeće rečenice opisuju problem koji je predmet ovog slučaja:

- Osnovne funkcionalnosti razmjene sjemena i biljaka su slanje vlastitih ponuda (bid and ask) i odgovaranje na tuđe ponude (bid and ask), što je rezervirano za članove.
- Članovi mogu pretraživati katalog po različitim kriterijima i otvoriti tablicu s otvorenim ponudama za svaku vrstu biljaka iz kataloga.
- Platforma nije komercijalna, odnosno na platformi se biljke ne prodaju i ne kupuju, već se razmjenjuju i poklanjaju. Stoga transakcije s biljkama nisu predmet oporezivanja.
- Ponuda uključuje biljku koja se nudi ili traži, količinu, mjesto, ponuđača, komentare, datum ponude, rok valjanosti ponude, upute za preuzimanje ili slanje, status itd.
- Ponude (nuženje ili potražnja) prikazane su u tabelarnom obliku, s mogućnošću sortiranja po stupcima te pretraživanja i filtriranja po raznim kriterijima (vrsta pogona, datum ponude, član, mjesto, status, itd...).
- Svaki član može odgovoriti na bilo koju otvorenu ponudu.
- Dogovori o bilo kojoj ponudi imaju oblik javnog ili privatnog razgovora.
- Osnovna jedinica za razgovor je poruka.
- Poruka sadrži podatke o članu koji je šalje (generira se automatski), tekst poruke i vidljivost (javno, privatno).
- Sustav treba podržavati sve faze upravljanja životnim ciklusom ponuda (kreiranje, promjena, zatvaranje, brisanje i arhiviranje). Ponudu kreira, mijenja, zatvara, briše i arhivira vlasnik ponude. Administrator može obrisati ponudu ako krši neka pravila i može arhivirati neaktivne ponude nakon određenog vremena.

18.3 Slučajevi korištenja

18.3.1 Opis slučaja korištenja riječima

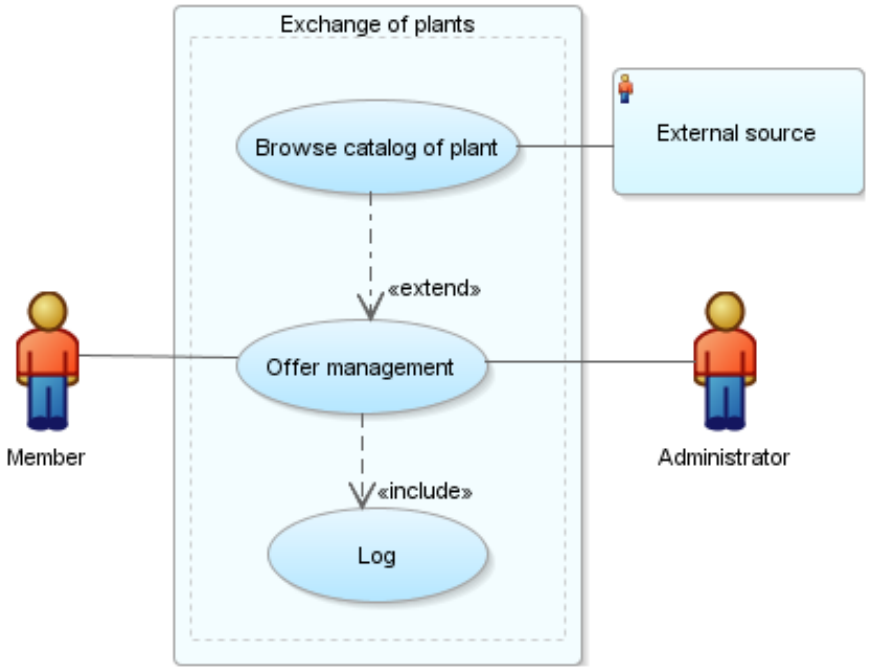
Komunikacija sa slučajem korištenja Upravljanje ponudom (Offer management) dostupna je svakom registriranom korisniku ili članu GreenDi platforme. Preduvjet za komunikaciju je uspješna prijava («include»). Slučaj korištenja Pretraživanje kataloga (Browse caatalog) je proširenje slučaja korištenja Upravljanja ponudom. Administrator je ujedno i vanjski korisnik slučaja korištenja. Istodobno, može pregledavati sve ponude i razgovore bez ograničenja i može promijeniti status svake ponude.

18.3.2 Polustrukturirani opis

Tablica 18.1 prikazuje slučaj korištenja.

18.3.3 Dijagram slučajeva korištenja

Prethodni opis problema prikazan je na dijagramu slučajeva korištenja 18.1.



Slika 18.1: Dijagram slučaja korištenja - Razmjena biljaka i sjemena

18.4 Model podataka

Ovaj odjeljak objašnjava model podataka.

18.4.1 Opis modela podataka riječima

Logički model podataka (model entiteti-veze) zapravo je nadogradnja podatkovnog modela iz poglavlja 14 i 15). Tipovi entiteta Biljka (Plant) i Tematsko područje (Thematic unit) preuzeti su iz poglavlja 14, a Korisnik (User) iz poglavlja 15. Osnovni tipovi entiteta za razmjenu biljaka i sjemena su Ponuda (Offer) i Poruka (Message). Ponuda je opisana sljedećim atributima: identifikacijska oznaka ponude; oznaka je li biljka ili sjeme ponuđeno ili traženo; količina biljke ili sjemena; lokacija gdje se biljka ili sjeme može preuzeti ili dostaviti; korisnik, odnosno ponuditelj koji nudi

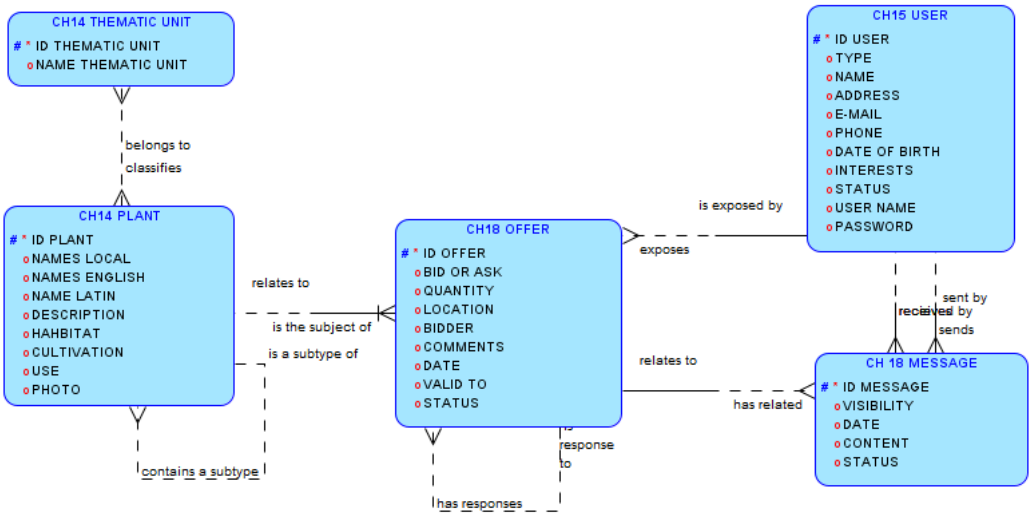
Tablica 18.1: Opis slučaja korištenja: Razmjena biljaka i sjemena.

Keyword	**Value**
ID:	*ch18-01*
Naziv:	*Razmjena biljaka i sjemena*
Opis:	*Razmjena biljaka.*
Primarni učesnik:	*Bilo koja osoba*
Preduvjeti:	*Osnovne informacije o biljkama otvorene su za pretraživanje i pregled bez registracije.*
Uvjeti po završetku:	*Ako se Pretraživanje kataloga biljaka pozove iz drugog slučaja uporabe («extend»), prenosi se ID odabrane biljke.*
Glavni	-
Uspješni scenarij:	1. Korisnik se pomiče gore-dolje po katalogu dok ne pronađe traženu biljku ili ne primijeni filter za pretraživanje. 2. Korisnik otvara popis ponuda za odabranu biljku ili za sve biljke. 3. Korisnik se pomiče gore-dolje po popisu ponuda ili primjenjuje filter za pretraživanje. 4. Korisnik otvara novu ponudu za odabranu biljku, upisuje detalje i izlaže novu ponudu, ili klikom na odabranu ponudu, otvara pop-up prozor s detaljima odabrane ponude. 5. Korisnik može odgovoriti na bilo koju aktivnu ponudu protuponudom ili prihvaćanjem ili započeti razgovor. 6. Razgovor je u obliku poruka i odgovora na poruke. 7. Poruke povezane u razgovoru vidljive su ponuditelju i drugoj osobi koja sudjeluje u razgovoru.
Proširenja:	• *Popis ponuda (tablica).* • *Detalji ponude (nove ili postojeće).* • *Razgovor (lista poruka).*
Učestalost korištenja:	*Približno do 100 dnevno.*
Status:	*Status razvoja*
Vlasništvo:	*javno, anonimni korisnici*
Prioritet:	*visok*

ili traži; komentari , kao što su primjedbe na kvalitetu, upute za isporuku itd.; datum izdavanja ponude; datum valjanosti ponude; status koji pokazuje je li ponuda aktivna, povučena, ili realizirana. Ponuda je izvorna ponuda, odnosno odgovor na izvornu ponudu. To znači da je svaka ponuda odgovor na 0 ili jednu ponudu, te da ima 0, 1 ili više odgovora. Svaku ponudu nudi jedan i samo jedan korisnik. Uz ponudu je vezano nula, jedna ili više poruka. Svaka poruka se odnosi na jednu i samo jednu ponudu, šalje ju jedan i samo jedan korisnik (pošiljatelj) i namijenjena je jednom i samo jednom korisniku (primatelju).

18.4.2 Logički model podataka

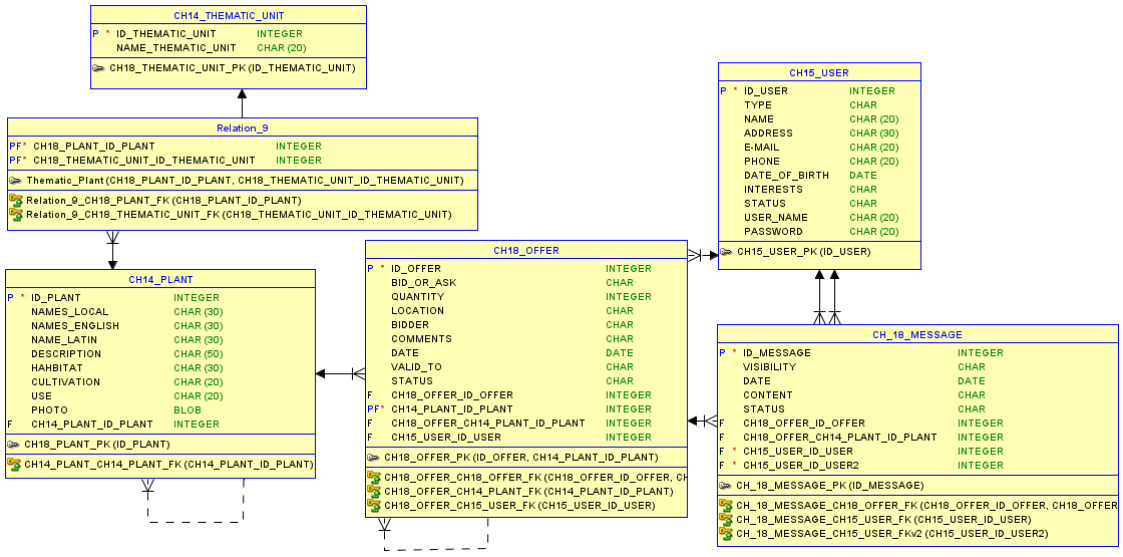
Logički model podataka je prikazan na Figure 18.2.



Slika 18.2: Logički model podataka

18.4.3 Relacijski model podataka

Automatska transformacija iz logičkog modela podataka u relacijski model podataka u Oracle SQL Data Modeleru omogućena je funkcijom *Engineering to relational*. Rezultat je prikazan na Figure 18.3.

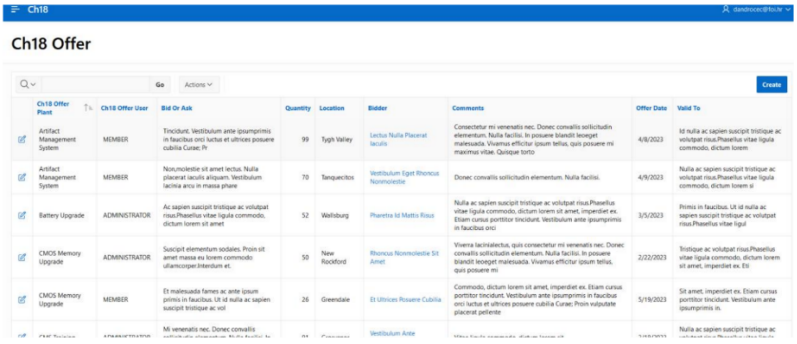


Slika 18.3: Relacijski model podataka

Generirajmo SQL skriptu pomoću Oracle SQL Data Modelera. Odaberite sve tablice na relacijskom modelu i upotrijebite funkciju *File* > *Export* > *DDL File* da dobijete skriptu. Generiranu skriptu možemo uvesti u APEX i izvršiti je.

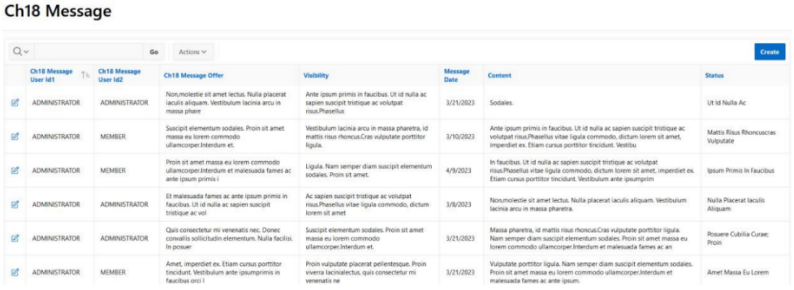
18.5 Aplikacijska sučelja

Ideja obrasca ponude je predstavljena na Figure 18.4.



Slika 18.4: Ponuda

Ideja obrasca poruke predstavljena je na Figure 18.5.



Slika 18.5: Poruka

18.6 Dopunski materijal za učenje

Na raspolaganju vam je sljedeći dodatni materijal za učenje:

- izvezena aplikacija,
- skripte za stvaranje, ispuštanje i umetanje,
- video vodiči.

Sav dodatni materijali za učenje dostupni su na public BeeAPEX project page. Prijavite se kao gost (zaporka nije potrebna). Pronađite udžbenik u odjeljku Knjige, skripte u mapi Part 2 > Chapter18 u odjeljku Skripte i video vodiče u Zbirci video vodiča. Materijal za kratke tečajeve nalazi se u odjeljku Kratki tečajevi.

18.6.1 Izvezene aplikacije

Izvezena aplikacija je zapakirana. Instalacija kreira tablice i popunjava podatke. Deinstalacija uklanja sve objekte baze podataka koji se koriste u ovoj aplikaciji.

18.6.2 Video vodiči

Video vodič prikazuje svaki korak u razvoju aplikacije.

18.7 Pitanja

1. Kako bismo ponudu učinili vidljivom s odgodom, od određenog trenutka, kada to ponuditelj želi?
2. Može li se ova aplikacija koristiti za razgovor o drugim temama osim stvarne razmjene biljaka i sjemenki? Kako?
3. U relacijskom podatkovnom modelu, CHAR format polja VALID TO definiran je u CH18_OFER. Koji su potencijalni problemi ako je datum napisan u char formatu u Oracleu?

18.8 Odgovori

1. Kako bi ponuda bila vidljiva s odgodom, od određenog trenutka kada je ponuditelj želi, morali bismo dodati atribut VISIBLE_FROM u CH_18_PONUDA i provjeriti je li taj uvjet ispunjen kada prikazivanje ponuda.
2. Teoretski, moguće je. Moderator chata treba otvoriti lažnu ponudu, au chatu mogu sudjelovati svi korisnici koji na nju odgovore sa svojom lažnom protuponudom.
3. Pohranjivanje datuma kao CHAR u Oracleu može dovesti do problema s integritetom podataka, problema s sortiranjem, ograničenih operacija datuma, poteškoća s formatiranjem i nekih drugih problema. Općenito se preporučuje korištenje odgovarajućih vrsta podataka datuma koje nudi Oracle, kao što su DATE ili TIMESTAMP, za točnu i učinkovitu pohranu datuma i manipulaciju.



19. Sustav za upravljanje recenzijama knjiga

A. KUTNJAK, L. HRUSTEK, A. BAGGIA, R.LESKOVAR

19.1 Poslovni pogled na slučaj

Sustav za upravljanje recenzijama knjiga osmišljen je s ciljem upravljanja recenzijama knjiga dostupnih u prodaji na proizvoljnoj on-line platformi. Tvrtka (primjerice Amazon) razmišlja o upravljanju katalogom dostupnih knjiga na način da temeljem recenzija korisnika odlučuje o ponudi knjiga, odnosno o dodavanju knjiga u katalog ili njihovom uklanjanju. Sustavom upravljanja recenzijama tvrtka prikuplja recenzije korisnika, kreira jedinstvenu bazu podataka i na temelju prikupljenih podataka odlučuje o daljnjim akcijama. Osnovni dionici baze podataka su administrator i korisnik, čitatelj knjige. Slučaj korištenja i ideja nadahnuti su i ponuđeni od strane Kaggle-a - platforme koja predstavlja online zajednicu namijenjenu podatkovnim znanstvenicima i entuzijastima strojnog učenja. Originalni podaci dostupni su here. Slučajevi korištenja i model podataka su temeljito izmijenjeni i dopunjeni. Cilj ovog poslovnog slučaja je pokazati razvojne sposobnosti APEX-a oslanjajući se samo na čarobnjake. Uz izuzetak Quick SQL-a (ukupno 31 kratki redak), programer neće napisati nijedan drugi kod.

19.2 Definicija problema

Sustav upravljanja recenzijama ima sljedeće funkcije:

- S obzirom na širok skup knjiga u globalnoj ponudi, prikupljanju se podaci o njihovim recenzijama i bodovnim ocjenama.
- Zajednički pogled platforme uključuje listu knjiga s podacima o nazivu, autorima, opisom sadržaja, izdavača, datumu izdavanja, kategorijama i recenzijama.
- Mogućnosti administratora su dodavanje knjiga u bazu platforme te njihovo uređivanje i selekcija po kategorijama.
- Budući da baza omogućava dodavanje recenzija od strane korisnika, administrator može pregledavati recenzije i bodovne ocjene i na temelju toga donositi odluke o akcijama po knjizi (dodavanje ili uklanjanje).
- Korisnik ima mogućnost registracije na platformi, koja mu omogućava detaljan pregled knjiga te recenziranje istih.
- Recenzije na strani korisnika imaju dvojaku korist. Prvo, na temelju pročitanih recenzija korisnik odlučuje o kupnji knjiga. Drugo, na temelju postojećih recenzija tvrtka poduzima radnje u vezi s knjigama, odnosno njihovo dodavanje ili uklanjanje iz kataloga.

19.3 Slučajevi korištenja

19.3.1 Narativni opis slučaja korištenja

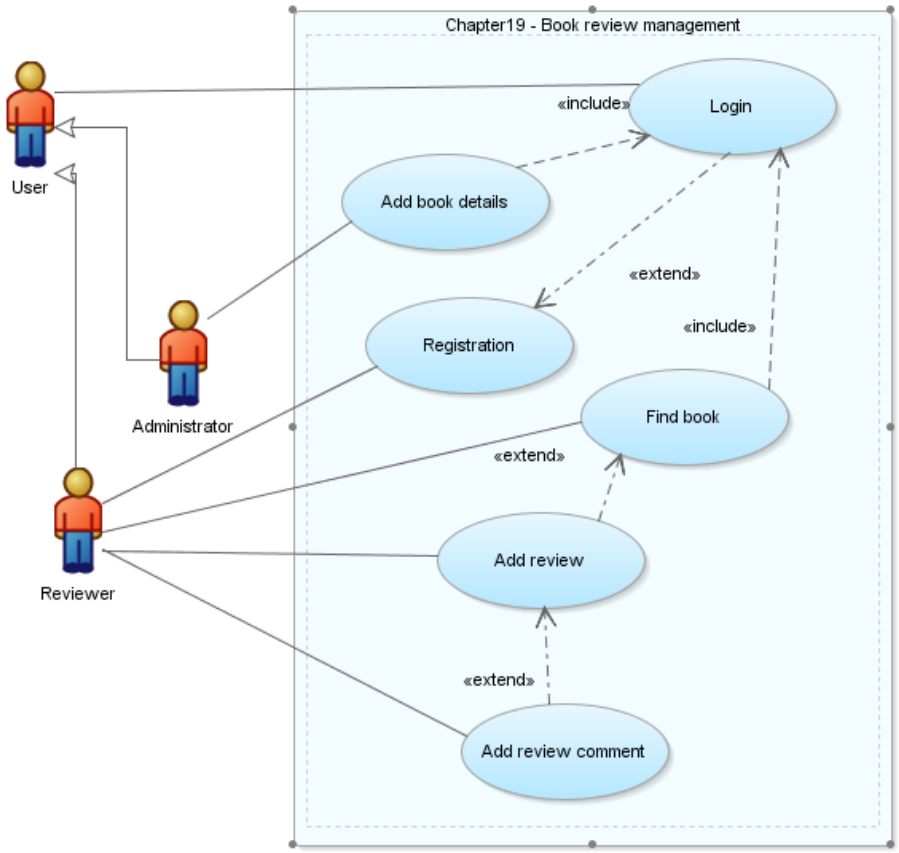
Administrator je zadužen za unos ili promjenu podataka o knjigama u sustavu upravljanja recenzijama te mu je omogućen pregled raznih statističkih pokazatelja vezanih uz knjige. Sustav upravljanja recenzijama baziran je na registraciji korisnika na platformi. Registracijom je korisnicima omogućen uvid u katalog knjiga dostupnih na platformi. Katalog knjiga sadrži detaljne informacije o dostupnim knjigama te predstavlja temelj za provođenje daljnjih akcija unutar platforme. Knjige su svrstane u kategorije što prilikom pretraživanja olakšava odabir prema osobnim preferencijama korisnika. Registrirani korisnik može dodati recenziju i dodati komentar na recenziju.

19.3.2 Polustrukturirani opis

Aktivnosti unosa novih podataka o knjizi dodijeljene su administratoru. Isti nadopunjava trenutnu bazu novim knjigama, što rezultira dopunom ponude u katalogu knjiga. Frekvencija korištenja platforme od strane administratora je na dnevnoj bazi. Korisnik izvršava registraciju na platformi za upravljanje recenzijama knjiga. Nakon uspješne registracije pregledava katalog knjiga, njihove recenzije te u konačnici dodaje vlastitu recenziju i bodovnu ocjenu. Frekvencija korištenja platforme od strane korisnika je nekoliko puta mjesečno.

Opis slučajeva korištenja nalazi se u tablici 19.1.

Prethodno navedeno, prikazano je na dijagramu slučaja korištenja (vidi sliku 19.1).



Slika 19.1: Dijagram slučaja korištenja: administrator dodaje novi komentar

Tablica 19.1: Opis slučaja korištenja: pregled recenzija knjiga

Ključna riječ	**Vrijednost**
ID:	*ch19-10*
Naslov:	*Pregled recenzija knjiga*
Opis:	*Sustav upravljanja recenzijama knjiga dostupan je temeljem registracije na platformi. Platforma omogućuje pregledavanje kataloga, njihovo filtriranje i klasifikaciju prema kategorijama.*
Primarna uloga:	*Korisnik*
Preduvjeti:	Obavezna registracija na platformi koja omogućava pregled podataka o knjigama.
Postuvjeti:	Provođenje daljnjih akcija unutar platforme: dodavanje ili uklanjanje knjiga (administrator) te recenziranje ili kupnja knjiga (korisnik).
Glavni	Amazon sustav upravljanja recenzijama
Uspješan scenarij:	Prvi scenarij (administrator, dodavanje nove knjige): 1. administrator se prijavljuje na platformi, 2. administrator odabire formu za unos nove knjige, 3. forma za unos uključuje naziv, autore, opis sadržaja, izdavača, datumu izdavanja i odabir kategorije, 4. nakon unosa podataka administrator završava postupak kreiranjem novog unosa, 5. nakon uspješne akcije, administrator se odjavljuje s platforme. Drugi scenarij (korisnik, recenziranje knjige): 1. korisnik izvršava registraciju na platformi, 2. korisnik pregledava katalog knjiga te odabire knjigu za koju želi izvršiti recenziju, 3. forma za recenziju uključuje naziv, cijenu, naziv profila, korisnost recenzije, ocjenu recenzije, sažetak recenzije i tekst recenzije, 4. nakon unosa podataka, korisnik završava postupak unosom nove recenzije, 5. korisnik se odjavljuje s platforme. Treći scenarij (korisnik, komentiranje recenzije knjige): 1. korisnik izvršava registraciju na platformi, 2. Korisnik pregledava recenzije knjiga, 3. Forma za recenziju uključuje naziv, cijenu, naziv profila, korisnost recenzije, ocjenu recenzije, sažetak recenzije i tekst recenzije, 4. Korisnik dodaje novi komentar recenzije, 5. Korisnik se odjavljuje s platforme.
Ekstenzije:	• *Unos knjiga.* • *Interaktivno izvješće za knjigu.* • *Recenzije knjiga.* • *Komentari na recenzije.*
Frekvencija korištenja:	*Dnevno, nekoliko puta u mjesecu.*
Status:	[Status razvoja]
Vlasnik:	Potencijalni korisnik, na bazi registracije.
Prioritet:	*visoki*

%endfigure

19.4 Podatkovni model

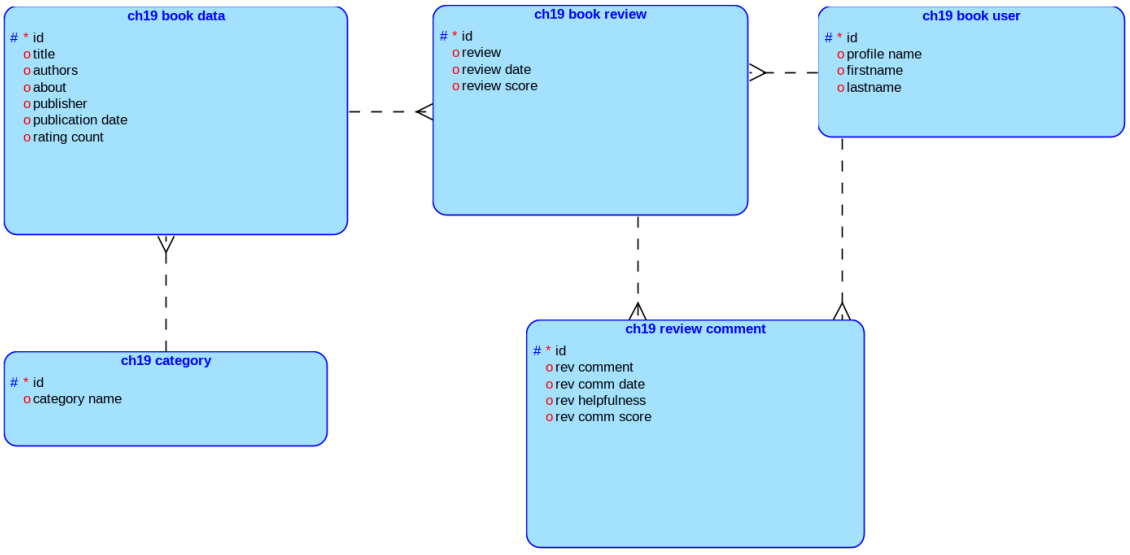
19.4.1 Narativni opis podatkovnog modela

Logički podatkovni model (model Entiteti-Veze) sadrži nekoliko entiteta: kategorija, knjiga_podaci, knjiga_korisnik, knjiga_recenzija i knjiga_recenzija_komentar. Entitet kategorija uključuje atribut kategorija_naziv. Entitet knjiga_podaci sastoji se od nekoliko atributa: naziv, autori, sadržaj, izdavač, datum_izdavanja i bodovna_ocjena. Entitet knjiga_korisnik uključuje naziv_profila, ime i prezime. Entitet knjiga_recenzija sastoji se od atributa: recenzija, datum_recenzije i ocjena_recenzije. Entitet recenzija_komentar sadrži odgovore na recenzije: recenzija_sažetak, recenzija_korisnost i recenzija_tekst_ocjena. Svaka knjiga pripada jednoj i samo jednoj kategoriji. Kategorija može imati više knjiga. Za jednu knjigu može biti zabilježeno više recenzija. Korisnik može komentirati više knjiga i više recenzija drugih korisnika. U svaki entitet dodajemo primarni UID koji se naziva ID. Ovi atributi će postati primarni ključevi u odgovarajućim tablicama.Nakon što generiramo SQL skriptu za izradu tablice, dodajemo dva ograničenja provjere - dopustit ćemo da ocjena recenzije i komentar ocjene recenzije imaju samo numeričke vrijednosti između 1 i 5.

Programer ima nekoliko mogućnosti za izradu tablica. Prva mogućnost odnosi se na korištenje SQL Developer Data Modeler-a s ciljem dizajna logičkog modela od početka, njegove transformacije u relacijski model pri čemu će se generirati SQL skripta. Druga mogućnost odnosi se na kreiranje tablica u APEX-u s ciljem izvoza skripti koje stvaraju tablice pri čemu će se primijeniti obrnuti inženjering u SQL Developer Data Modeleru za generiranje relacijskog modela koji će se zatim transformirati u logički model. Korištenje samo APEX Quick SQL funkcionalnosti najjednostavniji je pristup. Nakon što je Quick SQL napisan, može se transformirati u SQL skriptu pomoću APEX ugrađenog generatora.

19.4.2 Logički model podataka

Logički model podataka je prikazan na slici 19.2.

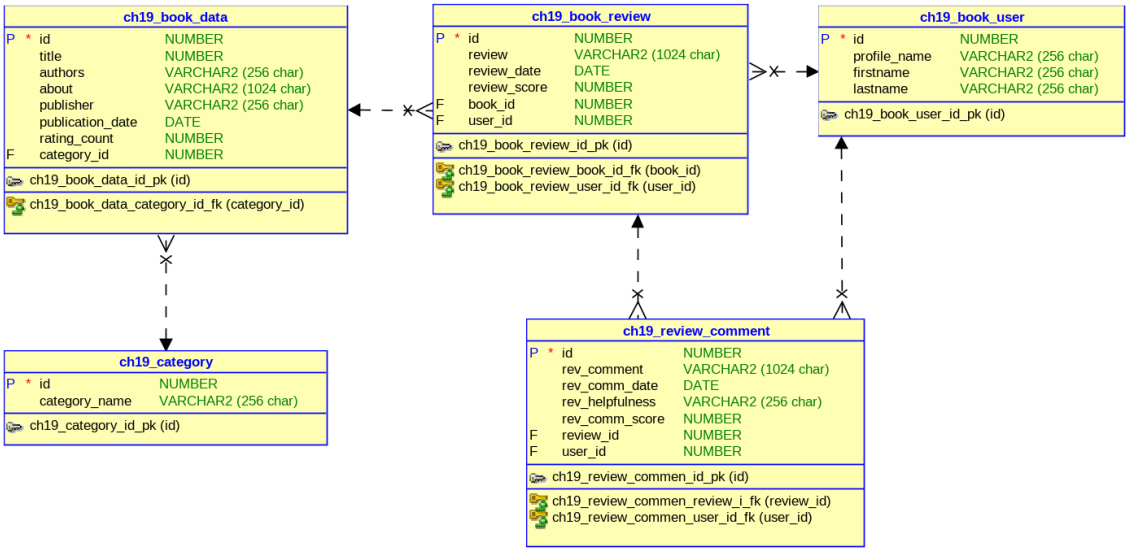


Slika 19.2: Logički model podataka

19.4.3 Relacijski model podataka

Automatsku transformaciju iz logičkog podatkovnog modela u relacijski podatkovni model u Oracle SQL Data Modeleru omogućuje funkcija Engineer to Relational Model. Rezultat je prikazan na

slici 19.3.



Slika 19.3: Relacijski model podataka

Neka sada Oracle SQL Data Modeler generira SQL skriptu. Odaberite sve tablice na relacijskom modelu i upotrijebite funkciju File > Export > DDL da bi dobili skriptu. Možete uvesti generiranu skriptu u APEX i izvršiti je.

19.4.4 SQL script

Skripta koja odgovara relacijskom modelu uključuje sljedeće SQL naredbe:

```
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
26 create table ch19_book_review (
27         id number generated by default on null as identity
28         constraint ch19_book_review_id_pk primary key,
29         review varchar2(1024 char),
30         review_date date,
31         review_score  number
32                 constraint ch19_book_revie_review_scor_ck
33                 check (review_score in (1,2,3,4,5)),
34         book_id   number
35         constraint ch19_book_review_book_id_fk
36         references ch19_book_data on delete cascade,
37         user_id   number
38                 constraint ch19_book_review_user_id_fk
39 references ch19_book_user on delete cascade);
40
41 create table ch19_review_comment (
42         id  number generated by default on null as identity
43                 constraint ch19_review_commen_id_pk primary key,
44         rev_comment varchar2(1024 char),
45         rev_comm_date date,
46         rev_helpfulness varchar2(256 char),
47         rev_comm_score number
48                 constraint ch19_review_c_rev_comm_scor_ck
49                 check (rev_comm_score in (1,2,3,4,5)),
50         review_id number
51                 constraint ch19_review_commen_review_i_fk
52                 references ch19_book_review on delete cascade,
53         user_id   number
54                 constraint ch19_review_commen_user_id_fk
55                 references ch19_book_user on delete cascade);
```

19.4.5 Quick SQL

Najjednostavniji postupak za generiranje SQL skripte je korištenje Quick SQL funkcije u APEX-u. Sljedeći redovi u Quick SQL-u omogućuju generiranje SQL skripte, prikazane ranije:

```
1  ch19_category
2          category_name vc256
3
4  ch19_book_user
5          profile_name vc256
6          firstname vc256
7          lastname vc256
8
9  ch19_book_data
10         title vc256
11         authors vc256
12         about   vc1024
13         publisher  vc256
14         publication_date date
15         rating_count num
```

```
        category_id /fk ch19_category

ch19_book_review
        review vc1024
        review_date date
        review_score  num /check 1,2,3,4,5
        book_id /fk ch19_book_data
        user_id /fk ch19_book_user

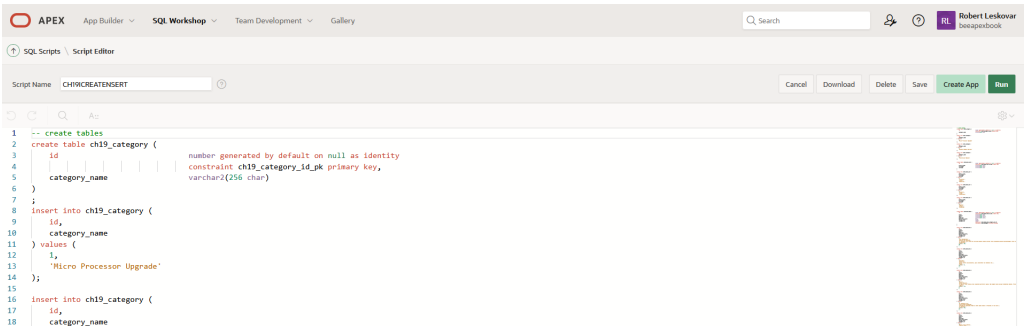
ch19_review_comment
        rev_comment vc1024
        rev_comm_date date
        rev_helpfulness vc256
        rev_comm_score  num /check 1,2,3,4,5
        review_id /fk ch19_book_review
        user_id /fk ch19_book_user
```

Kod je vrlo gust.Međutim, zahtijeva osnovno znanje o modeliranju podataka i sintaksi Quick SQL-a.

19.5 Sučelja aplikacije

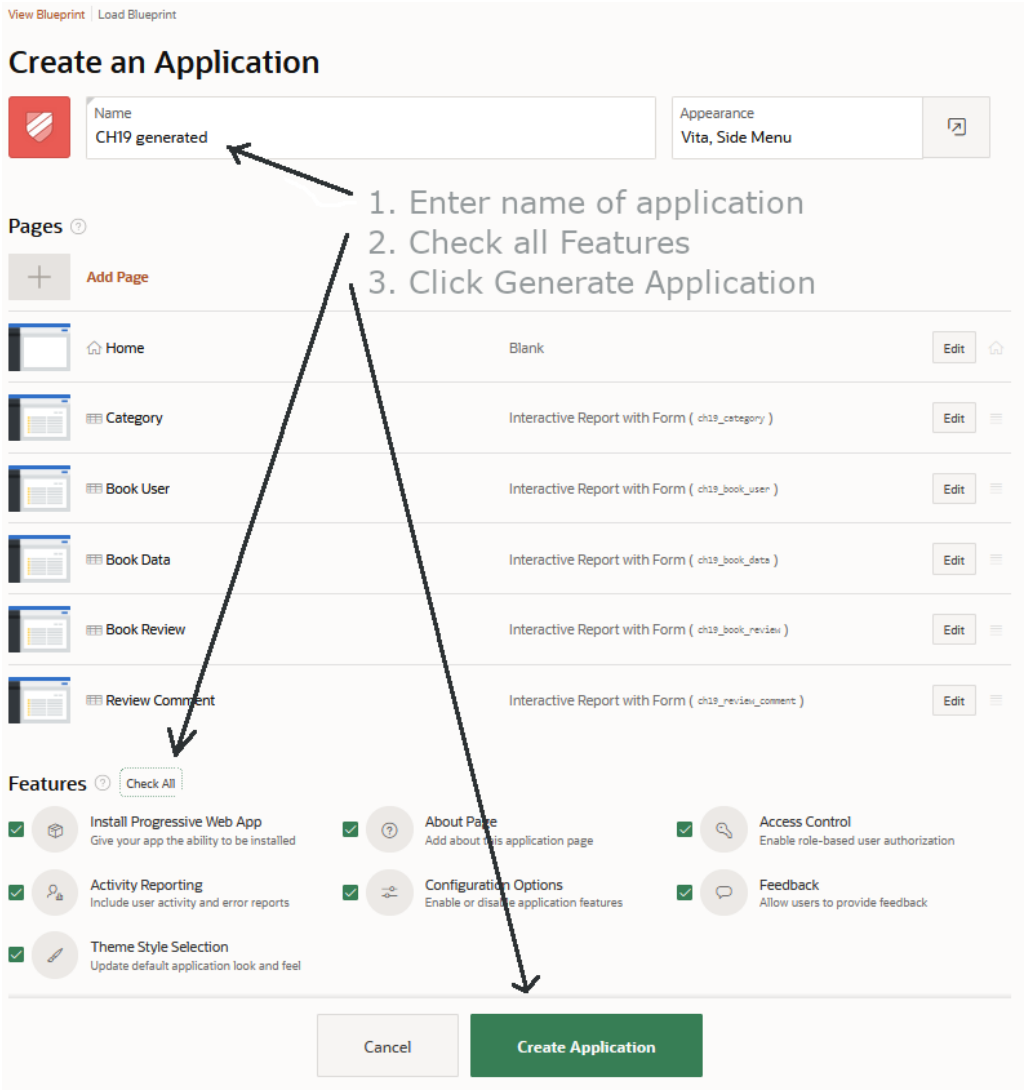
19.5.1 Administrator

U ovom ćemo slučaju generirati aplikaciju iz SQL skripte koja stvara tablice i popunjava ogledne podatke. Pronađite datoteku CH19CREATEINSERT.sql u materijalima za učenje i uvezite je u svoj radni prostor. Nakon uvoza otvorite je i kliknite Create App (pogledajte sliku 19.4).



Slika 19.4: Generiranje aplikacije iz skripte CH19CREATEINSERT - 1.dio

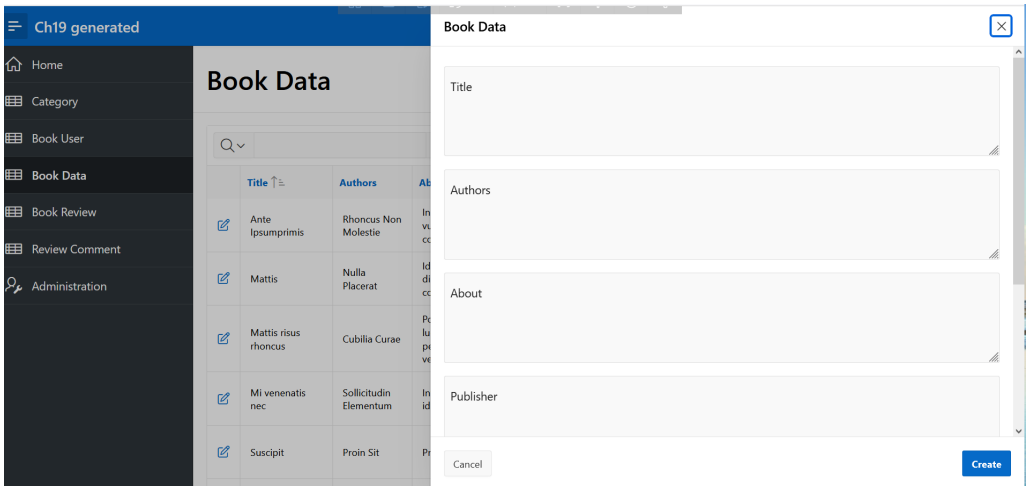
Imenujte aplikaciju, odaberite sve značajke i kliknite Generate Application (pogledajte sliku 19.5).



Slika 19.5: Generiranje aplikacije iz skripte CH19CREATEINSERT - 2. dio.

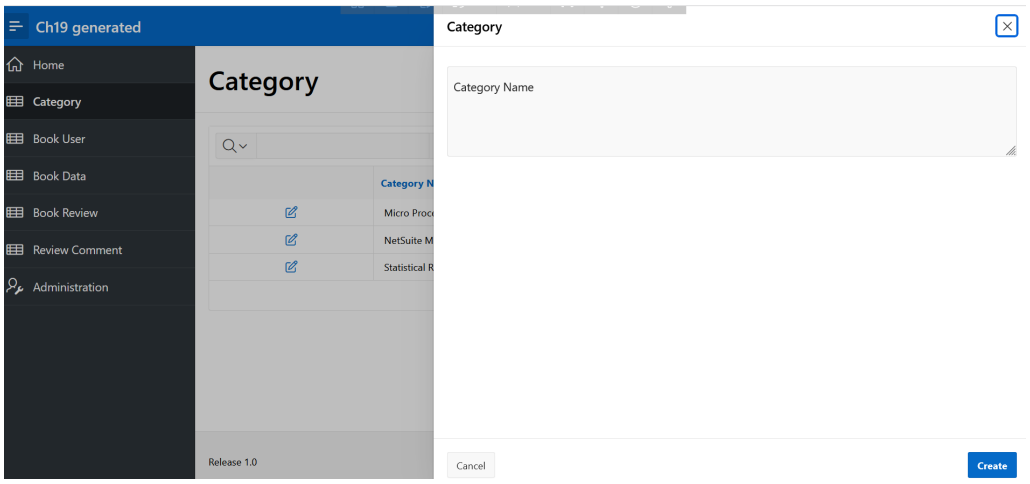
Radni prototip aplikacije je sada pripremljen. U nastavku će biti predstavljene funkcije za administratora i krajnjeg korisnika. Do sada niti jedan redak koda nije napisao programer (s izuzetkom Quick SQL-a) i nijedno sučelje aplikacije nije kreirano od strane programera. Stvarni kod iza aplikacije u potpunosti su sastavili APEX čarobnjaci.

Slika 19.6 prikazuje obrazac za unos nove knjige od strane administratora.



Slika 19.6: Unos knjige od strane administratora

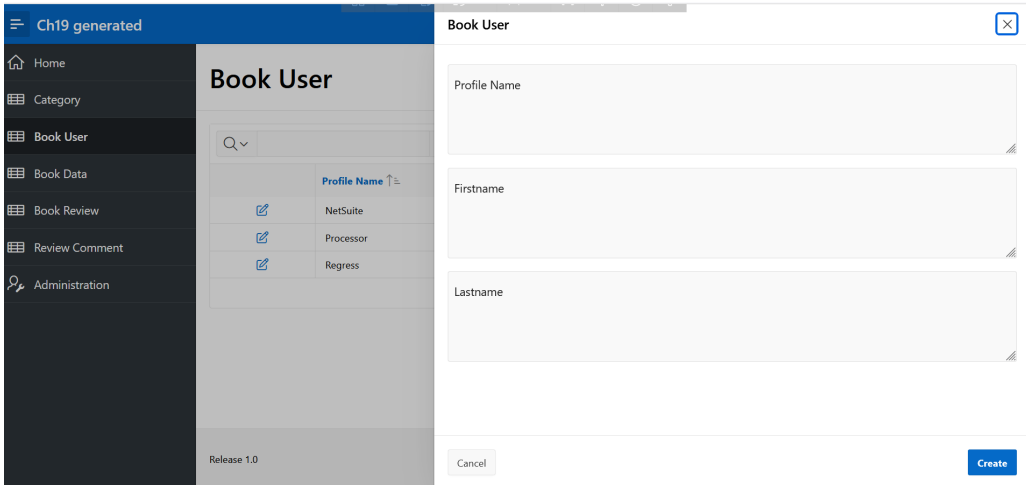
Slika 19.7 prikazuje obrazac za unos kategorije knjiga od strane administratora.



Slika 19.7: Unos kategorije od strane administratora

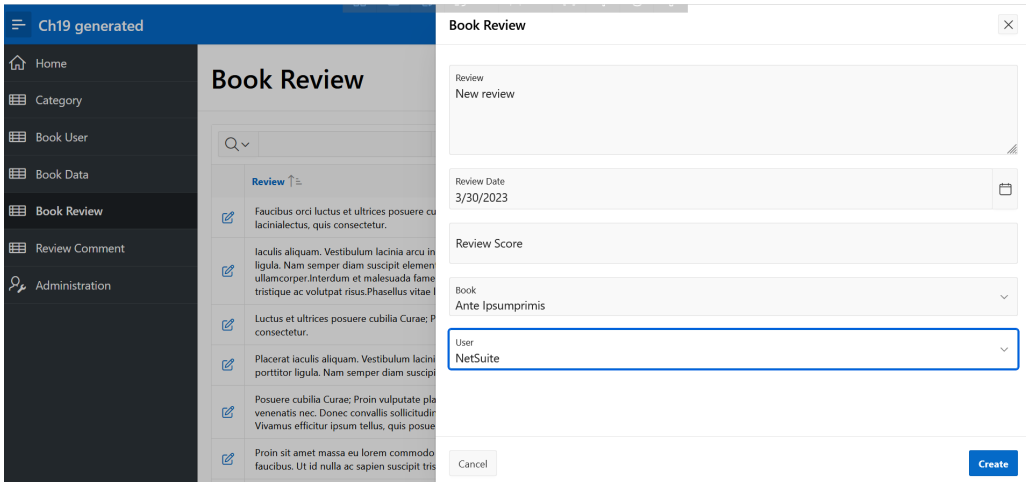
19.5.2 Korisnik

Slika 19.8 prikazuje obrazac za registraciju korisnika.



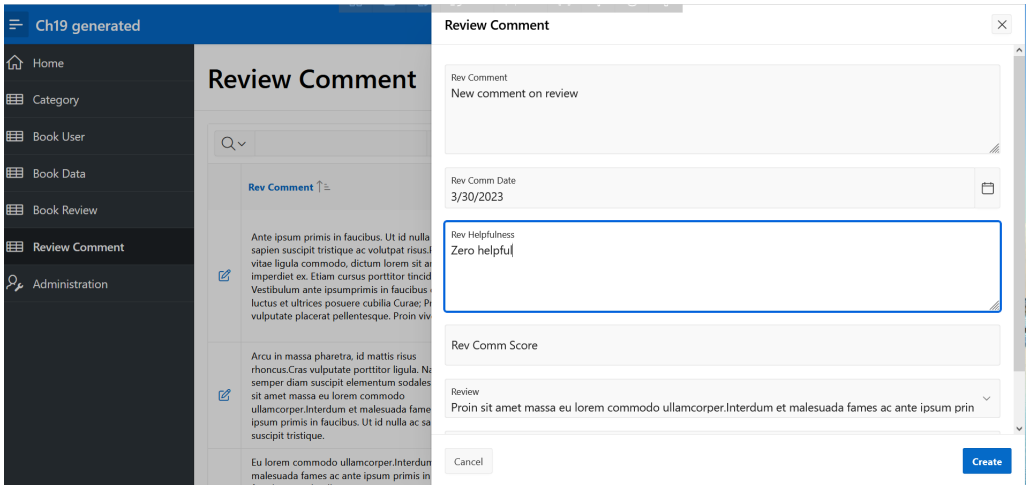
Slika 19.8: Registracija korisnika

Slika 19.9 prikazuje pregledavanje i dodavanje recenzija.



Slika 19.9: Pregledavanje i dodavanje recenzija

Slika 19.10 prikazuje obrazac za komentiranje recenzije.



Slika 19.10: obrazac za komentiranje recenzije

Izvješća su prema zadanim postavkama tablična. Krajnji korisnik može generirati grafikone bez programiranja, koristeći ugrađeni čarobnjak. Vidi poglavlje 8 za pretvaranje izvješća u grafikone.

19.6 Definiranje korisničkih uloga

Korisničke uloge mogu se definirati u Application Access Control u dijelu Shared Components. Za sada ćemo preskočiti pojedinosti i implementaciju za ovaj specifični slučaj. Za brzi pregled pogledajte poglavlje 13, a za temeljit uvid proučite APEX dokumentaciju o autorizaciji.

19.7 Dodatni materijal za učenje

Dostupni su vam sljedeći dodatni materijali za učenje:

- skripta za izradu i popunjavanje tablica,
- skripta za ispuštanje tablica,
- izvezena aplikacija,
- video koji pokazuje kako generirati aplikaciju iz skripte.

Sav dodatni materijal za učenje dostupan je na public BeeAPEX project page. Prijavite se kao gost (lozinka nije potrebna). Navigirajte do sekcije Books, skripte u mapi Part 2 > Chapter19 u sekciji Scripts i video upute u Collection of video guides. Materijal za kratke tečajeve dostupan je u Short courses section.

19.7.1 Izvezena aplikacija

Izvezena aplikacija je zapakirana. Instalacija kreira tablice i popunjava podatke. Deinstalacija uklanja sve objekte baze podataka koji se koriste u ovoj aplikaciji.

Zapakirana aplikacija je testirana i pokrenut će se u novom radnom prostoru ako su ispunjeni sljedeći zahtjevi:

- dodajte APEX korisnika prije pokretanja aplikacije. U radnom prostoru za razvoj i testiranje navigirajte do Shared Components > Application Access Control > Add User Role Assignment; unesite APEX korisnika i dodajte mu uloge Administrator, Contributer and Reader. U produkciji prije razvoja potrebno je konzultiranje sa stručnim osobljem.

Ako korisniku nije dodijeljena odgovarajuća uloga, uvezena aplikacija će se srušiti. Potrebno je izbrisati kolačiće web preglednika (npr. Firefox: Settings > Cookies and Site Data > Manage Data) nakon što se aplikacija sruši zbog neispunjenih zahtjeva.

19.7.2 Video upute

Video upute prikazuju svaki korak u razvoju aplikacije.

19.8 Pitanja

1. Zašto je u ovom slučaju korištena Quick SQL skripta?
2. Kako možete generirati više obrazaca i izvješća odjednom?
3. Kako možete autorizirati korisnika APEX aplikacije unutar same aplikacije?

19.9 Odgovori

1. Quick SQL skripta je uvedena kako bi se ubrzao razvoj podatkovnog modela. Quick SQL skripta generira SQL skriptu na zahtjev. SQL skripta je izvršna, tj. stvara objekte baze podataka poput tablica i pogleda.
2. Moguće je generirati više obrazaca i izvješća odjednom pisanjem skripte koja sadrži definiciju nekoliko tablica.
3. Za autorizaciju korisnika aplikacije APEX unutar same aplikacije, aplikacija mora biti generirana s Access Control značajkom. Nakon toga ovlašteni korisnik (Administrator) može dodati druge korisnike s ulogama Administrator, Contributor i Reader.



20. Sastavnica i obračun troškova

R. LESKOVAR, U. RAJKOVIČ, A.BAGGIA

20.1 Poslovni pogled na slučaj

Promatrano je malo poduzeće, **OSC** (10-20 zaposlenika) koje se bavi proizvodnjom kabelskih sklopova, kabelskih snopova, vodiča i signalnih svjetiljki. Također nude montažu elektromehaničkih dijelova i mehaničkih poluproizvoda. Njihovi kupci su veće tvrtke koje se bave prodajom kućanskih aparata, mjerne opreme, sustava upravljanja, brojila električne energije, medicinskih uređaja i slično. Narudžbe kupaca primaju putem telefona i e-maila. Za jednostavne proizvode i operacije, kao i za narudžbe koje se ponavljaju, **OSC** vrlo brzo šalju potvrdu kupcu. U posljednje vrijeme povećavaju narudžbe za sklopljene proizvode i složenije poluproizvode te stavljaju razinu poslovanja na višu razinu. Također, postoje deseci artikala u jednoj narudžbi, a svaki artikl može uključivati proizvod sastavljen od nekoliko poluproizvoda i materijala. Poluproizvodi se mogu sastojati od poluproizvoda i materijala. Kupci žele brzo dobiti ponudu. Isporuka kalkulacije cijene materijala i poluproizvoda (kalkulacija troškovnika ili kratki izračun sastavnice) zahtjevan je zadatak za otklanjanje grešaka i dugotrajan zadatak za sklopljene proizvode i poluproizvode na više razina. Može proći nekoliko dana da se kupcu odgovori s ponudom. Obračun troškova rada (operacija) još je jedan obračun koji je zbog sažetosti poglavlja izostavljen. Stoga će opseg poglavlja biti samo izračun sastavnice. Vlasnik tvrtke izrazio je sljedeće potrebe:

- izvješće o proizvodima, poluproizvodima, materijalu i upitima kupaca ako je moguće,
- izvješće o sastavnici,
- obrazac za uređivanje proizvoda, poluproizvoda, materijala i upita kupaca ako je moguće,
- obrazac za uređivanje sastavnice,
- izračun sastavnice i njezin prikaz te prezentacija u obliku stabla.

Aplikacija bi trebala biti upotrebljiva na telefonu, tabletu ili stolnom računalu.

20.2 Definicija problema

Voditelj tvrtke suočava se s nekoliko problema povezanih s agilnim odgovorom:

- izračun traje dugo i traži puno vremena, greške je teško i zahtjevno otkloniti, a sve je potrebno vrlo brzo pripremiti,
- dospjele obveze i pogreške u izračunu mogu utjecati na poslovanje na više načina kao što su gubitak potencijalnih kupaca, slabiji financijski rezultati, smanjenje poslovnog ugleda, smanjenje prodaje i popunjavanja kapaciteta i slično.

Tablica 20.1: Opis slučaja korištenja: Prijava i održavanje osnovnih podataka

Ključna riječ	**Vrijednost**
ID:	*Ch20-01*
Naslov:	*Prijava i održavanje osnovnih podataka*
Opis:	*Menadžer koristi aplikaciju APEX za prijavu i održavanje (odabir, ažuriranje) osnovnih podataka bilo da se radi o proizvodu, poluproizvodu, materijalu ili upitu. Materijali i spojevi opisani su istim skupom atributa. Ažuriranje cijena materijala neće izazvati ponovni izračun cijena spojeva u kojima se materijal koristi. Upit kupca može se sastojati od nekoliko proizvoda i poluproizvoda. Jedinica mjere u ovom slučaju bit će komad.*
Primarna uloga:	*Menadžer*
Preduvjeti:	*Menadžer ima račun za APEX aplikaciju.*
Postuvjeti:	*Nakon izvješćivanja ili održavanja podataka može se poduzeti druga radnja uključujući izračun sastavnice.*
Glavni	*Scenariji*
Uspješan scenarij:	1. *Otvaranje web preglednika i prijava u aplikaciju* 2. *Odabir opcije izbornika ili stranice za navigaciju "Prod./Semi/Material"* 3. *Za uređivanje, potrebno se pomaknuti prema dolje po stranici ili filtrirati podatke kako biste pronašli određeni proizvod, poluproizvod ili materijal* 4. *Klik na ikonu olovke za određeni proizvod, poluproizvod ili materijal i potrebno je ažurirati podatke.* 5. *Klik na opciju "Apply changes".*
Ekstenzije:	*nema*
Frekvencija korištenja:	*prosječno 5 puta dnevno*
Status:	*Završeno*
Vlasnik:	*Menadžer*
Prioritet:	*Visok*

Gore navedene rizike potrebno je učinkovito ublažiti web aplikacijom.

20.3 Slučajevi korištenja

20.3.1 Narativni opis

Mogu se odrediti osnovni zadaci koje treba izvršiti:

- izvješćivati i održavati podatke o proizvodima, poluproizvodima i materijalima. Konzumaciju tretiramo kao izvješćivanje u obliku tablice i stabla, a održavanje kao dodavanje, ažuriranje i brisanje (vidi Tablicu 20.1).
- Izračun sastavnice za određeni upit, proizvod ili poluproizvod (vidi Tablicu 20.2).

Naslovi slučajeva korištenja su "prijava i održavanje osnovnih podataka", "izvješćivanje i održavanje strukturnih podataka" i "izračunavanje sastavnice".

20.3.2 Polustrukturirani opis

Gornja priča prikazana je na dijagramu slučaja korištenja.

Tablica 20.2: Opis slučaja korištenja: Izvješćivanje i održavanje strukturnih podataka

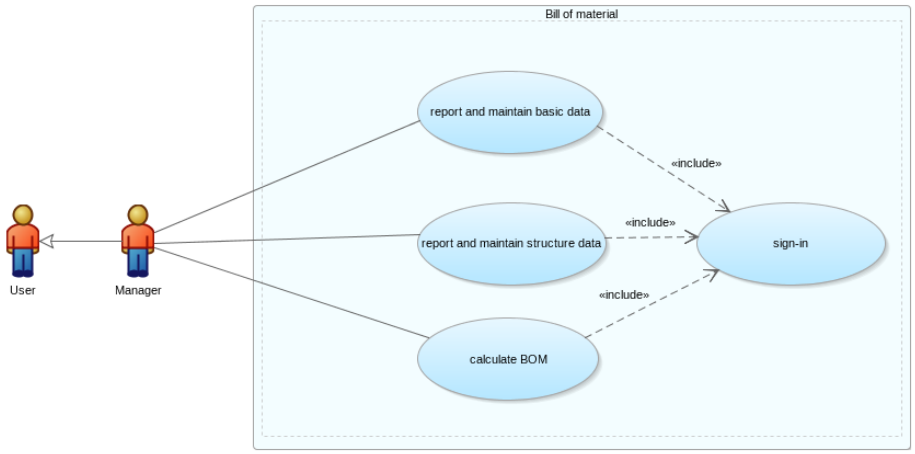
Ključna riječ	**Vrijednost**
ID:	*Ch20-02*
Naslov:	*Izvješćivanje i održavanje strukturnih podataka*
Opis:	*Menadžer koristi APEX aplikaciju za izvješćivanje (pregled odabranih podataka) i održavanje (umetanje i brisanje) strukture proizvoda, poluproizvoda i materijala. Upit kupca može se sastojati od nekoliko proizvoda i poluproizvoda. Umetanje novih stavki ili brisanje postojećih stavki neće pokrenuti (ponovni) izračun cijena spoja.*
Primarna uloga:	*Menadžer*
Preduvjeti:	*Menadžer ima račun za APEX instancu.*
Postuvjeti:	*Nakon izvješćivanja ili održavanja podataka može se poduzeti druga radnja uključujući izračun sastavnice.*
Glavni	*Scenariji*
Uspješan scenarij:	1. *Otvaranje web preglednika i prijava u aplikaciju* 2. *Odabir opcije izbornika ili stranice za navigaciju "Bill-of-material"* 3. *Za dodavanje komponente potrebno je kliknuti gumb "Create".* 4. *Za brisanje stavke iz strukture potrebno je kliknuti "Delete".*
Ekstenzije:	*nema*
Frekvencija korištenja:	*prosječno 30 puta u mjesecu*
Status:	*Završeno*
Vlasnik:	*Menadžer*
Prioritet:	*Visok*

Tablica 20.3: Opis slučaja korištenja: Izračun sastavnice

Ključna riječ	**Vrijednost**
ID:	*Ch20-03*
Naslov:	*Izračun sastavnice*
Opis:	*Menadžer pomoću aplikacije APEX izračunava cijenu strukture opisane sastavnicom za određeni artikl, bilo da se radi o proizvodu, poluproizvodu ili upitu. Finalne cijene ažuriraju se za sve stavke agregirane u strukturi, bez obzira na njihovu složenost i dubinu, uključujući i gornji spoj.*
Primarna uloga:	*Menadžer*
Preduvjeti:	*Menadžer ima račun za APEX instancu.*
Postuvjeti:	*Nakon izračuna, može se poduzeti bilo koja druga radnja uključujući još jedan izračun sastavnice.*
Glavni	*Scenariji*
Uspješan scenarij:	1. *Otvaranje web preglednika i prijava za pristup web pregledniku, mobitelu ili računalu, status nastavnika, pristup sveučilišnom sustavu (interna upotreba), prijavu u aplikaciju.* 2. *Odabir opcije izbornik ili stranice za navigaciju "Tree view and calculation".* 3. *Odabir spoja za izračun* 4. *Klik na gumb "Calculate". Pojavljuje se poruka koja prikazuje status kalkulacije i izračunatu cijenu.* 5. *Mogućnost proširenja ili sažimanja strukture stabla odabrane stavke za prikaz podataka o dijelovima na nižim razinama.*
Ekstenzije:	*nema*
Frekvencija korištenja:	*prosječno 30 puta na mjesec*
Status:	*Završeno*
Vlasnik:	*Menadžer*
Prioritet:	*Visok*

20.3.3 Dijagram slučaja korištenja

Gornja priča prikazana je na dijagramu slučaja korištenja (vidi Sliku 20.1).



Slika 20.1: Dijagram slučaja korištenja

20.4 Podatkovni model

20.4.1 Narativni opis podatkovnog modela

Postoje samo dva entiteta u podatkovnom modelu za ovaj slučaj. Entitet **ch20 psm** (kasnije prikazan Tablicom) opisuje proizvode, poluproizvode, materijale i upite kupaca sa samo tri atributa zajednička svima: ID, opis, jedinica mjere i cijena po jedinici mjere. Potrebne su vrijednosti za sve atribute. Zašto? Besmisleno je pohranjivati primjerak bez znanja kako se zove, kako se mjeri i koja je cijena po jedinici. ID je jedinstveni broj bez decimalnog dijela. Opis mora sadržavati barem jedan znak za ispis. Domena mjernih jedinica ograničena je na milimetar, centimetar, metar i komad. Zbog malog broja različitih vrijednosti mjernih jedinica neće biti uveden nikakav dodatni entitet koji bi predstavio činjenicu da se jedna mjerna jedinica mnogo pojavljuje u **ch20 psm** dok se druga **ch20 psm** instanca odnosi samo na jednu instancu jedinice mjere.

Da bi se prikazala proizvoljna hijerarhijska struktura (spisak materijala bilo koje širine i bilo koje dubine), entitet **ch20 bom** ima samo tri atributa: identifikaciju poddijela, identifikaciju spoja i količinu poddijela sadržanog u spoju. Vrijednosti za sve atribute moraju biti prisutne. Par (**identification of sub part, identification of aggregate**) e jedinstven, no mi ćemo dodati primarni ključ generiran kao niz cjelobrojnih vrijednosti kako bismo iskoristili prednosti APEX okruženja s niskim kodom. Jedna instanca **ch20 bom** može imati dvostruki odnos s **ch20 psm**: poddio se odnosi na jednu instancu **ch20 psm** a spoj se odnosi na jednu instancu **ch20 psm**. U suprotnom smjeru imamo: jedna **ch20 psm** instanca može ukazivati na mnoge poddijelove i mnoge spojeve **ch20 bom** entiteta. Stoga imamo dva odnosa 1-prema-više između **ch20 psm** i **ch20 bom**. Moramo istaknuti još jednu stvar o logičkom modelu podataka: jedinstvenost para **(identification of sub part, identification of aggregate)**. Logički model u Oracle Data Modeleru nema "sintaksu" za izražavanje ove jedinstvenosti kao poseban objekt u bazi podataka koji se zove indeks. Ali na relacijskom modelu podataka moguće je uvesti jedinstvenost navedenog para vrijednosti kao indeks. U relacijskom modelu podataka jedinstveni indeks sastoji se od dva polja u tablici. Ovaj jedinstveni indeks jamči da nijedan duplikat vrijednosti parova neće biti pohranjen u bazi podataka - motor baze podataka zaštitit će programera i krajnjeg korisnika od narušavanja integriteta podataka.

20.4.2 Implementacija poslovnih pravila u bazi podataka

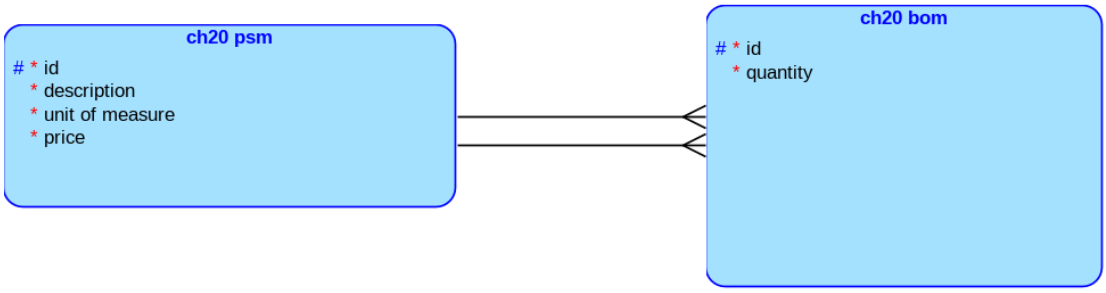
Logički i relacijski model podataka je jednostavan. U matematici popis materijala je posebna vrsta grafa gdje je svaki čvor povezan sa samo jednim roditeljskim čvorom osim s korijenskim čvorom koji nema roditeljski čvor. Vrlo je lako razbiti hijerarhiju uvođenjem dodatne veze između čvorova ili brisanjem veze. U kontekstu baze podataka promjene veza utječu na integritet podataka. Pohranjivanje cikličke strukture u **ch20 bom** uzrokovalo bi da izračun troškovnika nikad ne završi. U kontekstu poslovnih pravila nemoguće je i zabranjeno da automobil (proizvod) bude sastavljen od kotača (poluproizvoda), a da ujedno taj kotač bude sastavljen od automobila. Također, automobil se ne može sastaviti sam od sebe. Kako bismo spriječili kršenje poslovnih pravila, uvest ćemo mehanizam okidača (još jedan objekt baze podataka) koji je obično predstavljen s nekoliko redaka koda u PL/SQL jeziku. Okidač će se izvršiti prije pisanja ili ažuriranja **ch20 bom**. Takav kod bi spriječio novo umetanje u **ch20 bom** ako par obrnutim redoslijedom već postoji ili i novi poddio i spoj imaju istu vrijednost. Implementacija takve vrste pravila ne može se prikazati na logičkom ili relacijskom modelu podataka.

Također je nemoguće predstaviti kako će izračun sastavnice biti implementiran u modele podataka. Ovo poglavlje će pružiti uvid u kod koji hijerarhiju okreće naglavačke (odozdo prema gore) i započinje zbrajanje ukupnih troškova od dna - tj. prvo izračunava trošak materijala u poluproizvodima i njihov trošak poluproizvoda u proizvodu. Upamtite da model podataka s dva entiteta ili dvije tablice može prikazati bilo koju složenost (širinu i dubinu hijerarhije) sastavničke strukture.

Započnimo najprije s podatkovnim modelom.

20.4.3 Logički model podataka

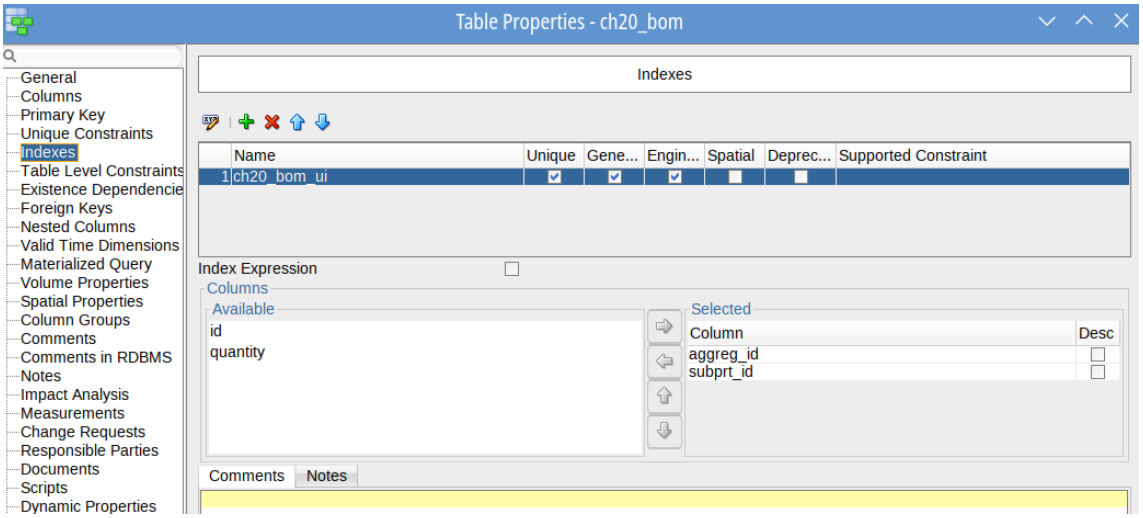
Logički model podataka je predstavljen na Slici 20.2.



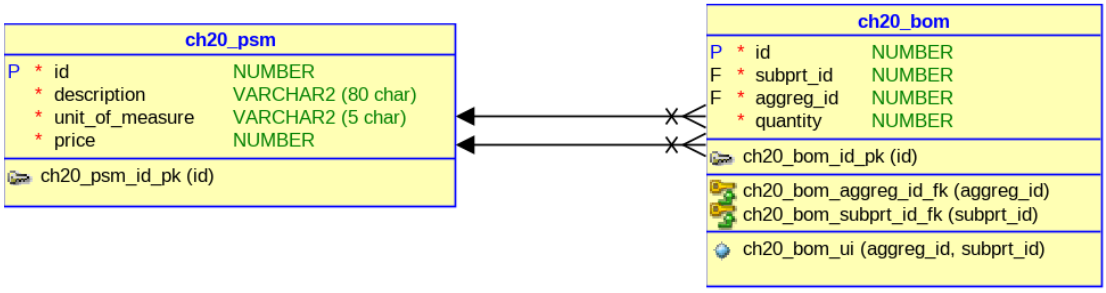
Slika 20.2: Logički model podataka

20.4.4 Relacijski model podataka

Automatska transformacija iz logičkog modela podataka u relacijski model podataka u Oracle SQL Data Modeleru omogućuje funkcija *Engineering to relational*. Zatim se uvodi jedinstveni indeks *ch20_bom_ui* sastavljen od dva polja *aggreg_id* i *subprt_id* (vidi Sliku 20.3). Rezultat, relacijski model podataka spreman za izvoz kao SQL skripta prikazan je na Slici 20.4.



Slika 20.3: Logički model podataka



Slika 20.4: Logički model podataka.

Oracle SQL Data Modeler također generira SQL skriptu za stvaranje tablice, niza i okidača. Odabiru se sve tablice na relacijskom modelu i upotrijebi se funkcija *File* > *Export* > *DDL File* da se dobije skripta. Skriptu je potrebno spremiti i provjeriti tablicu definicija. Prvo se mora definirati Tablica*ch20_psm* i onda *ch20_bom*. Sljedeći kod definira tablicu *ch20_psm* i postavljena ograničenja:

```
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' ) );
```

Sljedeći kod definira tablicu *ch20_bom* postavljenih ograničenja i jedinstveni indeks:

```
CREATE TABLE ch20_bom (
        id         NUMBER GENERATED BY DEFAULT ON NULL
        AS IDENTITY ( START WITH 1 NOCACHE ) NOT NULL,
        subprt_id NUMBER NOT NULL,
        aggreg_id NUMBER NOT NULL,
        quantity  NUMBER NOT NULL
) LOGGING;

ALTER TABLE ch20_bom
        ADD CONSTRAINT ch20_bom_id_pk PRIMARY KEY ( id );

ALTER TABLE ch20_bom ADD CONSTRAINT
ch20_bom_subprt_id_fk FOREIGN KEY ( subprt_id )
REFERENCES ch20_psm ( id )
        ON DELETE CASCADE NOT DEFERRABLE;

ALTER TABLE ch20_bom ADD CONSTRAINT
ch20_bom_aggreg_id_fk FOREIGN KEY ( aggreg_id )
REFERENCES ch20_psm ( id )
        ON DELETE CASCADE NOT DEFERRABLE;

CREATE UNIQUE INDEX ch20_bom_ui
        ON ch20_bom (aggreg_id, subprt_id );
```

Skriptu je potrebno spremiti s uputama koje kreiraju tablice s primarnim i stranim ključevima, provjeravaju ograničenja i jedinstveni indeks. Sada je vrijeme za generiranje tablica u Oracle bazi podataka. Može se uvesti generirana skripta i izvršiti u APEX-u.

20.4.5 Objekti u APEX-u

Svojim preglednikom potrebno je doći do radnog prostora u APEX-u. U ovom odjeljku provest će se sljedeće:

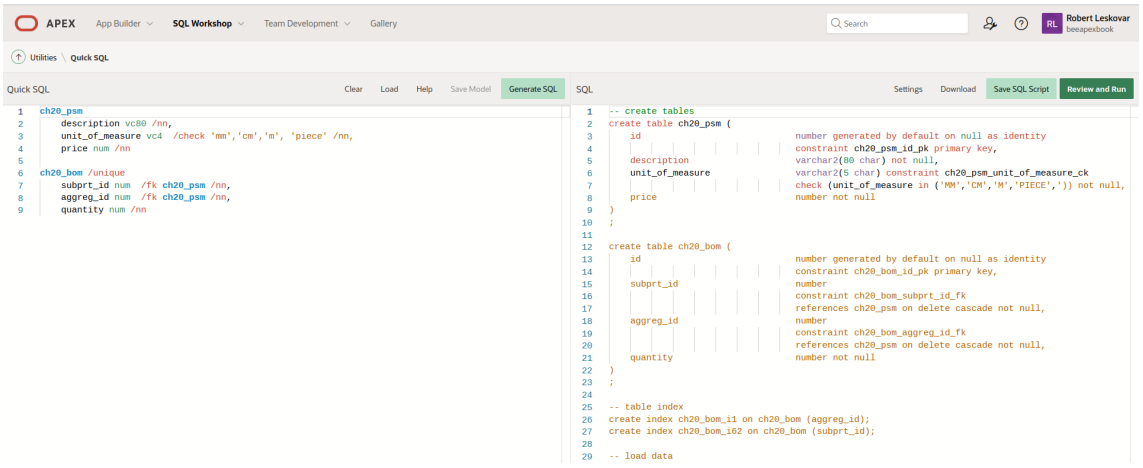
- kreirati tablice i indekse,
- kreirati funkcije i procedure,
- kreirati okidač.

Postoje dvije opcije za izradu tablica i indeksa. Ako se za generiranje skripte koristi Oracle Data Modeler, prvo se uvozi (SQL Workshop > SQL Scripts > Upload), a zatim se pokreće. Druga je mogućnost generiranje gornje skripte pomoću Quick SQL-a u slučaju da želite koristiti samo APEX za generiranje takve skripte. Idite na SQL Workshop > Utilities > Quick SQL. Stavite sljedeći tekst u polje teksta Quick SQL (lijevi dio prozora):

```
ch20_psm
  description vc80 /nn,
  unit_of_measure vc4  /check 'mm','cm','m', 'piece' /nn,
  price num /nn

ch20_bom /unique
  subprt_id num  /fk ch20_psm /nn,
  aggreg_id num  /fk ch20_psm /nn,
  quantity num /nn
```

Kliknite generiraj SQL. Tablice imaju ista svojstva kao i tablice generirane u Oracle Data Modeleru (vidi Figure 20.5).



Slika 20.5: Logički model podataka.

Pogledajte retke 26 i 27. Ova dva retka, iako točna, bit će kasnije uređena. Kliknite Save SQL Script, Review i Run. RZamijenite retke 26 i 27 (kreirajte dva indeksa) i zamijenite jednom izjavom:

```
create unique index ch20_bom_ui on ch20_bom (aggreg_id, subprt_id);
```

Sada pokrenite skriptu. Generirat će se dvije tablice i jedan jedinstveni indeks. Imajte na umu da izvođenje ove skripte neće biti uspješno ako već pokrenete skriptu generiranu pomoću Oracle Data Modelera. Razlog je to što objekti baze podataka već postoje.

Za izradu funkcije unesite sljedeći tekst u SQL Workshop > SQL Commands:

```
create or replace FUNCTION  CH20_GET_PRICE (
  xpsm IN   NUMBER
  ) RETURN NUMBER
IS
  y_price number;
BEGIN
  SELECT price INTO y_price FROM ch20_psm  WHERE ID = xpsm;
  return y_price;
EXCEPTION
  WHEN OTHERS THEN
    return(-505);
END CH20_GET_PRICE;
```

Ova funkcija će vratiti cijenu (proizvod, poluproizvod, materijal, inventar) za dani ID. Ako se nešto neočekivano dogodi s bazom podataka (izgubljena veza, server ne radi, hardverska pogreška, nema podataka za navedeni ID), tada će biti vraćena negativna vrijednost -505. Odaberite sav tekst i pokrenite. Dobit ćete povratnu informaciju da je funkcija stvorena.

Za kreiranje procedure za izračun troškova sastavnice, unesite sljedeći tekst u SQL Workshop > SQL Commands:

```
create or replace PROCEDURE  CH20_CALCULATE_BOM
  (xpsm IN NUMBER,
  ystatus OUT INTEGER,
  yprice OUT NUMBER)
-- creation date: 20.1.2023
-- author: Robert Leskovar
-- input parameter: xpsm, id of product or semi-product
```

```
8  -- output parameters 1: status of calculation:
9                  0=not calculated, 1=succesfully calculated
10 -- output parameters 2: price calculated
11 -- description: calculation of the total costs of given product
12 --                              and all underlaying levels
13 -- exceptions: system and user defined
14 -- date of testing: 20.1.2023
15 -- author of test: swqlab
16 -- status: APPROVED
17 is
18   v_rezult   NUMBER(38, 2);
19   v_psm        NUMBER;
20   v_price    NUMBER(38, 2);
21   e_negative_price EXCEPTION;
22   CURSOR c_ds IS
23    WITH STRUK(hier, subprt_id, aggreg_id, quantity) AS
24     (SELECT 1 AS hier, subprt_id, aggreg_id, quantity
25      FROM CH20_BOM JOIN CH20_PSM ON subprt_id = CH20_PSM.ID
26         WHERE aggreg_id = xpsm
27    UNION ALL
28     SELECT hier+1, C1.subprt_id, C1.aggreg_id, C1.quantity
29      FROM CH20_BOM C1 JOIN CH20_PSM P ON C1.subprt_id = P.ID
30       JOIN STRUK C2 ON C1.aggreg_id = C2.subprt_id)
31    SELECT hier, subprt_id, aggreg_id, quantity
32     FROM STRUK ORDER BY hier DESC, aggreg_id, subprt_id;
33 BEGIN
34   ystatus := 0;
35   v_rezult := 0.0;
36   v_psm := 0;
37   SAVEPOINT old_state;
38   FOR r_ds IN c_ds LOOP
39    v_price:= CH20_GET_PRICE(r_ds.subprt_id);
40    IF ( v_price <= 0 ) THEN
41     RAISE e_negative_price;
42    END IF;
43    IF ( r_ds.aggreg_id = v_psm ) THEN
44     v_rezult := v_rezult + v_price * r_ds.quantity;
45     UPDATE ch20_psm SET price = v_rezult WHERE ID = r_ds.aggreg_id;
46    ELSE
47     v_psm := r_ds.aggreg_id;
48     v_rezult := v_price * r_ds.quantity;
49     UPDATE ch20_psm SET price = v_rezult WHERE ID = r_ds.aggreg_id;
50    END IF;
51   END LOOP;
52   COMMIT;
53   ystatus := 1;
54   yprice := v_rezult;
55 EXCEPTION
56   WHEN e_negative_price THEN
57    ROLLBACK TO old_state;
```

```
58     ystatus := 0;
59     yprice := -505;
60   WHEN OTHERS THEN
61     ROLLBACK TO old_state;
62     ystatus := 0;
63     yprice := -100;
64 end CH20_CALCULATE_BOM;
```

Ova procedura će vratiti status kalkulacije i cijenu (proizvod, poluproizvod, materijal, infra) za dati ID. Ako se nešto neočekivano dogodi s bazom podataka (izgubljena veza, server ne radi, hardverska pogreška, nema podataka za navedeni ID), tada će biti vraćena negativna vrijednost. Ako čitatelj nije upoznat s PL/SQL, dovoljno je pročitati komentare u proceduri. Inače prilično složen kursor se kreira zajedničkim tabličnim izrazom (naredba WITH). Ovaj kursor vraća uniju dva skupa. Osigurava kompletnu strukturu spoja koja je poredana odozdo prema gore. Cijene posrednih spojeva pohranjuju se u odgovarajuće redove i kumuliraju dok se ne obrade svi redovi u pokazivaču.

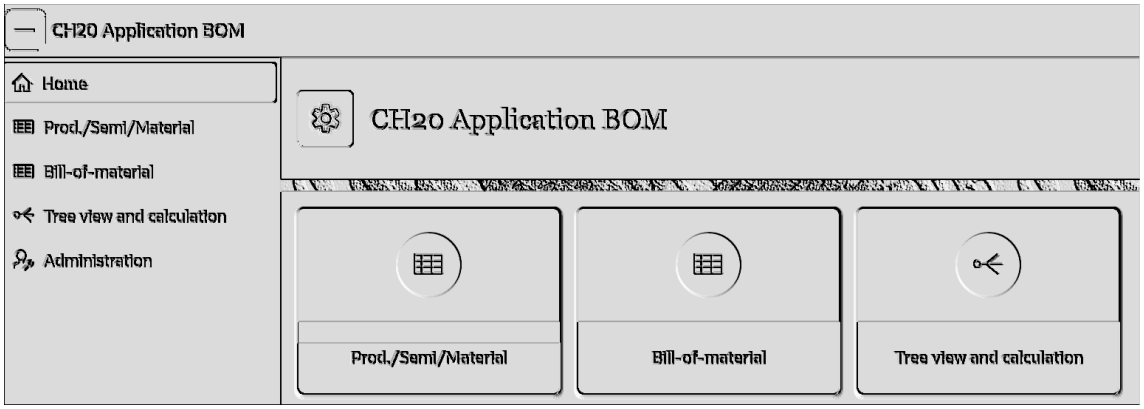
Posljednji objekt koji se stvara je okidač. Okidač će osigurati da niti jedan spoj i poddio u popisu materijala ne mogu imati istu vrijednost (vagon se sastoji od automobila) i stvara se relacija prelaska (ako je automobil sastavljen od kotača onda se kotač ne može sastojati od automobila). Za izradu okidača koji implementira navedena pravila unesite sljedeći tekst u SQL Workshop > SQL Commands:

```
1  create or replace trigger CH20_TRG_BOM_RULES
2    before insert or update on ch20_bom
3    for each row
4
5    declare
6     xsubprt numeric :=:new.subprt_id;
7     xaggreg numeric :=:new.aggreg_id;
8     xcount numeric;
9    begin
10    select count(subprt_id) into xcount from ch20_bom
11     where subprt_id = xaggreg and aggreg_id = xsubprt;
12    if xsubprt = xaggreg or xcount > 0 then
13     :new.subprt_id :=null;
14    end if;
15 end;
```

Ovim okidačem spriječit ćemo umetanje forcing NULL vrijednosti u polje tablice koje mora imati vrijednost. Prednost definicije NOT NULL u stupcu tablice nad mehanizmom okidača uzrokovat će da se podaci ne umetnu ako se prekrši ovo poslovno pravilo. Međutim, zaštitili smo hijerarhiju da postane graf s ciklusom. Ako preskočimo stvaranje ovog okidača, trebali bismo trenirati krajnjeg korisnika da ne upadne u zamku ciklusa. Puno je bolje spriječiti nego liječiti, zar ne?

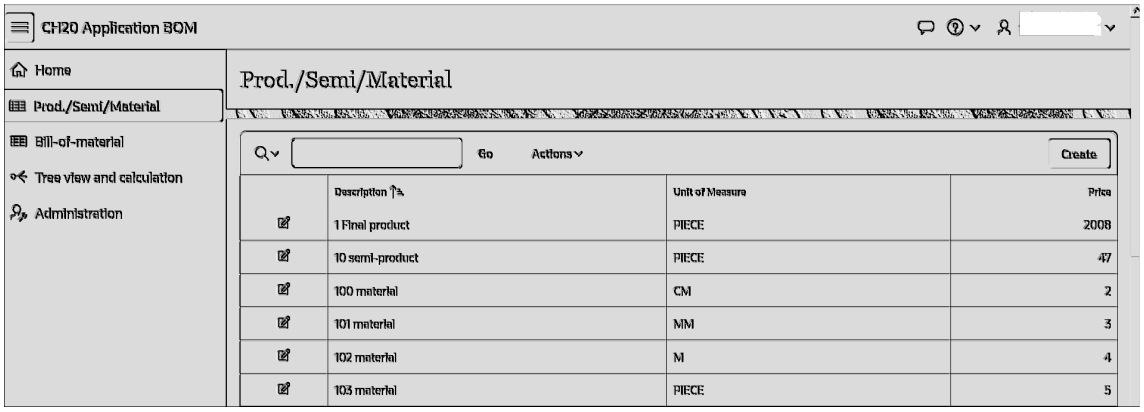
20.5 Sučelja aplikacije

Na sljedećim slikama prikazana su sučelja aplikacije. Prvo predstavljamo skicu ulaznog prozora. Korisnik će pristupiti upravljanju osnovnim podacima, upravljanju strukturnim podacima i izračunima bilo pomoću lijevog bočnog izbornika ili putem navigacije po stranici (vidi Sliku 20.6).

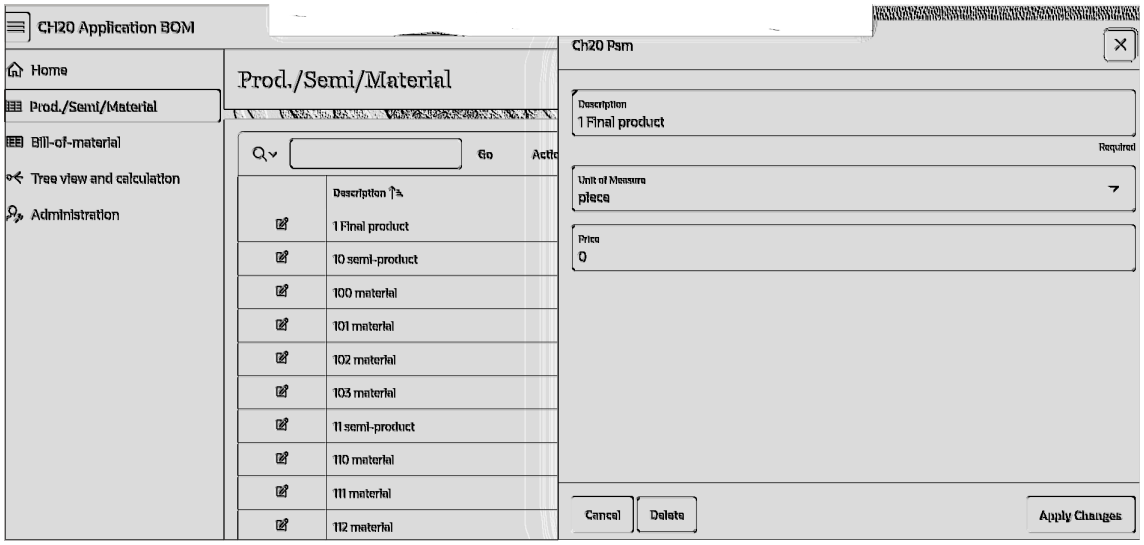


Slika 20.6: Početna stranica aplikacije

Sljedeće dvije slike (20.7 i 20.8) predstavljaju skice izvješća i obrasca za proizvode, poluproizvode, materijale i upite. Dostupno je umetanje, ažuriranje i brisanje.



Slika 20.7: Upravljanje osnovnim podacima - izvješće



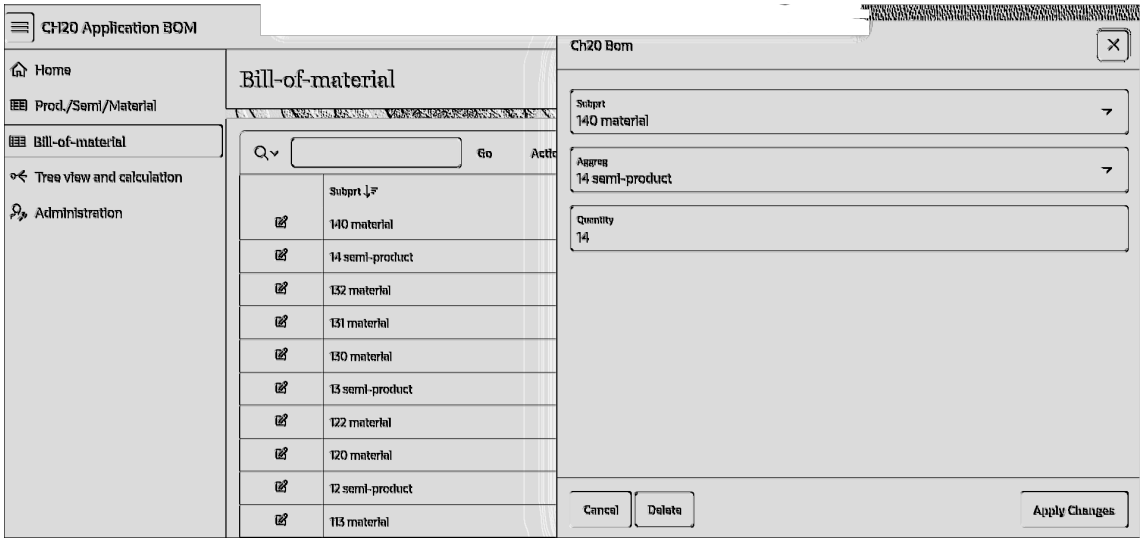
Slika 20.8: Upravljanje osnovnim podacima - obrazac

Sljedeće dvije slike (20.9 i 20.10) predstavljaju skice izvješća i obrazac za strukturu, popis

materijala. Dostupno je umetanje, ažuriranje i brisanje.

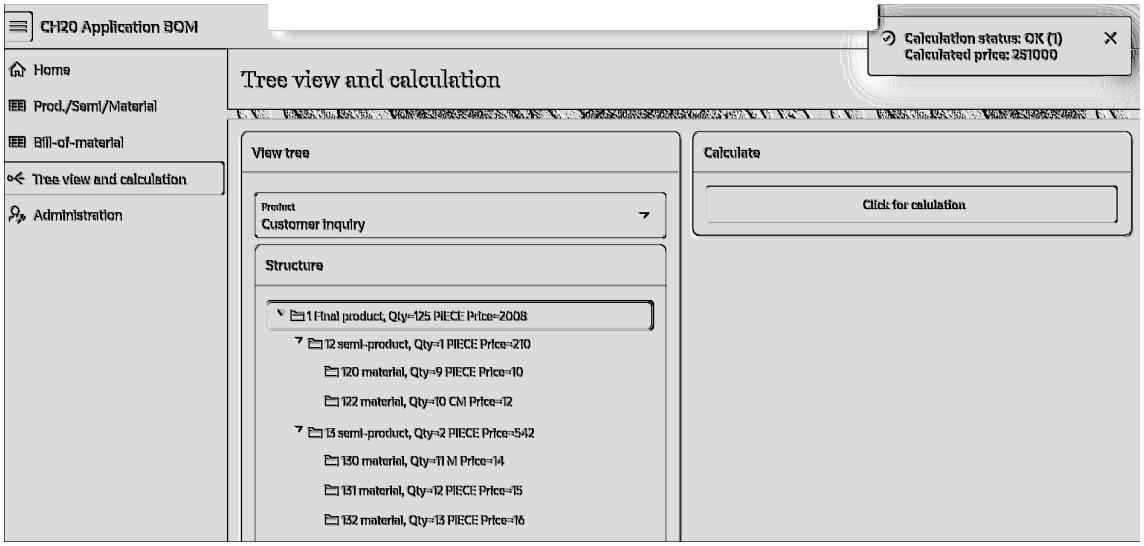


Slika 20.9: Upravljanje strukturnim podacima - izvješće



Slika 20.10: Upravljanje strukturnim podacima - obrazac

Skica (20.11) predstavlja sučelje za izvođenje izračuna bilo kojeg spoja (poluproizvoda, proizvoda, upita). Korisnik najprije odabire stavku, a zatim pritišće gumb Izračunaj. Status kalkulacije i izračunate cijene prikazuju se kao skočni prozor i prikazuje se komponenta stabla. Korisnik može proširiti i sažeti hijerarhiju.



Slika 20.11: Stranica za izračun sastavnice

20.6 Dodatni materijal za učenje

Možete pronaći sljedeće dodatne materijale za učenje:

- izvezena aplikacija,
- video upute.

Sav dodatni materijal za učenje dostupan je na public BeeAPEX project page. Prijavite se kao gost (lozinka nije potrebna). Navigirajte do sekcije Books, skripte u mapi Part 2 > Poglavlje20 u sekciji Scripts i video upute u Collection of video guides. Materijal za kratke tečajeve je u Short courses section.

20.6.1 Izvezene aplikacije

Izvezena aplikacija je zapakirana. Instalacija kreira tablice, indeks, funkcije, procedure i okidače te popunjava podatke. Deinstalacija uklanja sve objekte baze podataka koji se koriste u ovoj aplikaciji.

20.6.2 Video upute

Video upute prikazuju svaki korak u razvoju aplikacije.

20.7 Pitanja

1. Kako biste promijenili logički model podataka da implementirate novi entitet za mjerne jedinice, jer želimo uključiti desetke jedinica?
2. Kako narediti Quick SQL da se napravi gore spomenuta promjena?
3. Koje će biti posljedice pada jedinstvenog indeksa?ch20_bom_ui?

20.8 Odgovori

1. Primjenjujemo novi entitet *ch20_uom* i postavljamo najmanje dva atributa (ID i opis mjerne jedinice). Zatim postavljamo relaciju 1-prema-više između*ch20_psm* i *ch20_uom*.
2. U Quick SQL-u prvo definiramo tablicu *ch20_uom* kao:

```
ch20_uom
  description vc80
```

i promjenu definicije *ch20_psm* kao:

```
ch20_psm
  description vc80 /nn,
  unit_of_measure num /fk ch20_uom
  price num
```

3. Ispuštanje jedinstvenog indeksa ch20_bom_ui omogućilo bi pohranjivanje više parova (poddio, spoj) s istim vrijednostima. Tablica bi zauzela više prostora na disku. To ne bi imalo utjecaja na funkcionalnost izračuna, no obrada bi trajala dulje i trošilo bi se više RAM-a.



21. Prehrana i upravljanje dijetom

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

21.1 Poslovni pogled na slučaj

Ovaj primjer predstavlja najpopularniju grčku medijsku marku hrane, s dosegom od preko 500.000 korisnika. Tvrtka je ponosni vlasnik grčkog vodećeg prehrambenog časopisa na tržištu i digitalnog izdanja, broj jedan prehrambene web stranice u zemlji, sve većeg broja događaja uživo, uključujući najveći događaj hrane u Grčkoj - Greek Food Show - i iznimno uspješne serije kuharice.

Tvrtka je odlučila izraditi i promovirati web alat koji korisnicima omogućuje pristup tisućama recepata putem njihove web stranice. Osim toga, korisnici mogu personalizirati svoje omiljene recepte dodavanjem ili uklanjanjem sastojaka, a usluga će pružiti podatke o nutritivnoj vrijednosti novih recepata, omogućujući korisnicima stvaranje vlastite personalizirane kuharice omiljenih recepata.

21.2 Definicija problema

Korisnici koji uživaju u kuhanju često se suočavaju s ograničenjima kada je riječ o tome što mogu jesti, bilo zbog zdravstvenih problema poput dijabetesa, visokog kolesterola ili osobnih prehrambenih preferencija poput veganske ili bezglutenske prehrane. Jedan problem s postojećim izvorima recepata koji se temelje na webu je taj što često ne pružaju sveobuhvatne alternative za prilagođavanje ovim ograničenjima. Kao rezultat toga, korisnici mogu postati frustrirani ograničenim informacijama i potpuno odustati od pretraživanja recepata, što će rezultirati slabim angažmanom.

21.3 Slučajevi korištenja

21.3.1 Narativni opis

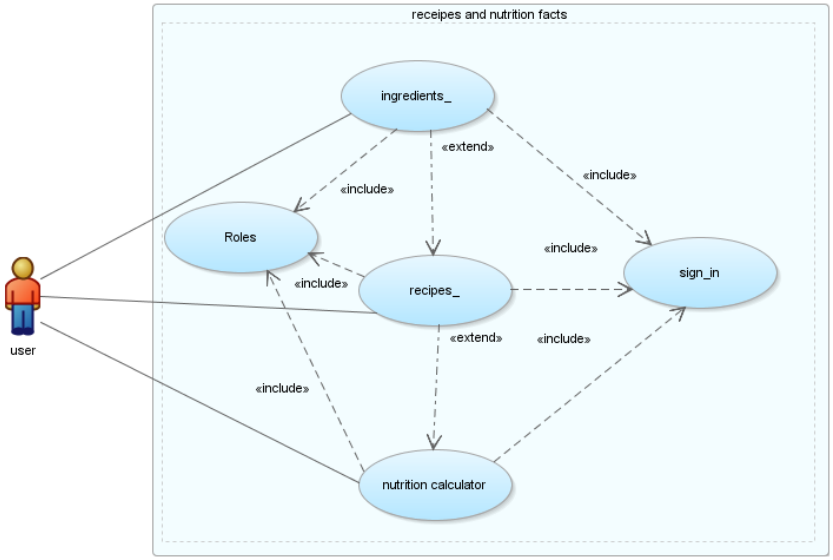
Bit će kreirana Oracle Apex aplikacija koja korisnicima omogućuje pregled i dodavanje raznih zdravih recepata. Aplikacija će pružiti detaljan opis svakog recepta, uključujući popis sastojaka i podatke o nutritivnoj vrijednosti svakog sastojka. Korisnici također mogu dodati vlastite recepte i odgovarajuće sastojke pomoću obrasca. Nutritivne vrijednosti svakog recepta, poput broja kalorija, ukupnih proteina, broja porcija i vremena pripreme, bit će prikazane u dijagramima radi lakšeg čitanja, a ne u jednostavnoj tablici podataka.

21.3.2 Polustrukturirani opis

Ovaj tekst opisuje razvoj Oracle Apex aplikacije koja korisnicima omogućuje pristup raznim zdravim receptima. Aplikacija uključuje detaljne opise recepata i podatke o hranjivim vrijednostima za svaki sastojak, kao i mogućnost za korisnike da dodaju vlastite recepte i sastojke. Opis polustrukture nalazi se u tablici 21.1

21.3.3 Dijagram slučaja korištenja

Gornja priča prikazana je na dijagramu slučaja korištenja (vidi Sliku 21.1).



Slika 21.1: Dijagram slučaja korištenja

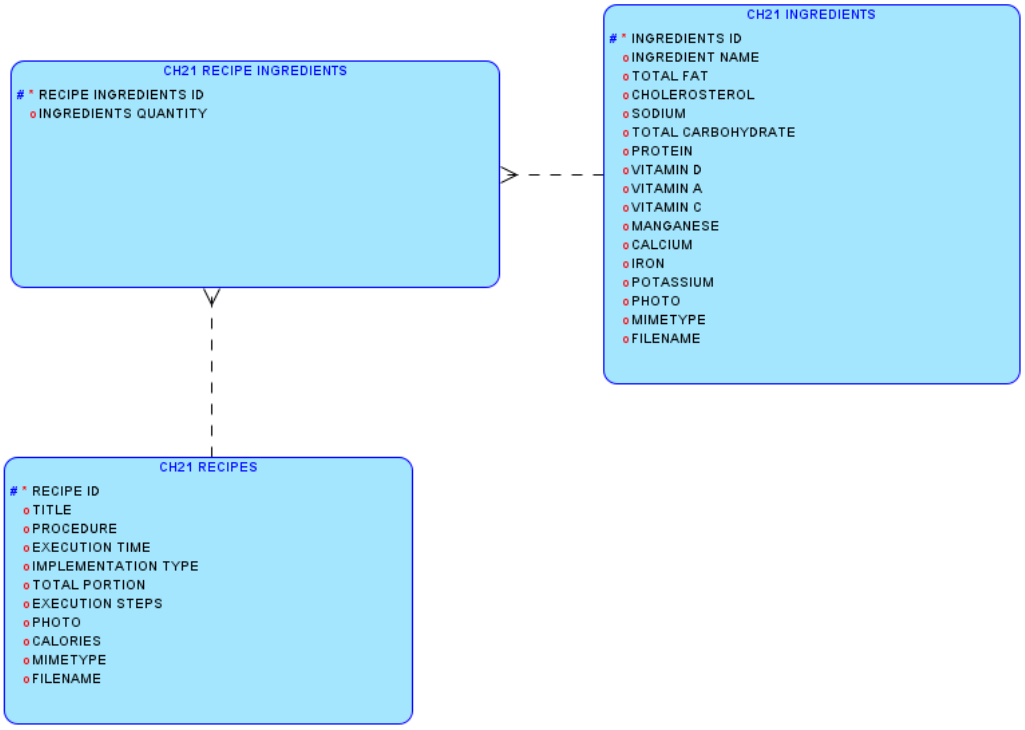
21.4 Podatkovni model

21.4.1 Logički model podataka

Dijagram logičkih podataka prikazan je na Slici 21.2.

Tablica 21.1: Opis slučaja korištenja: Pristup receptima i sastojcima te izračun nutritivnih podataka

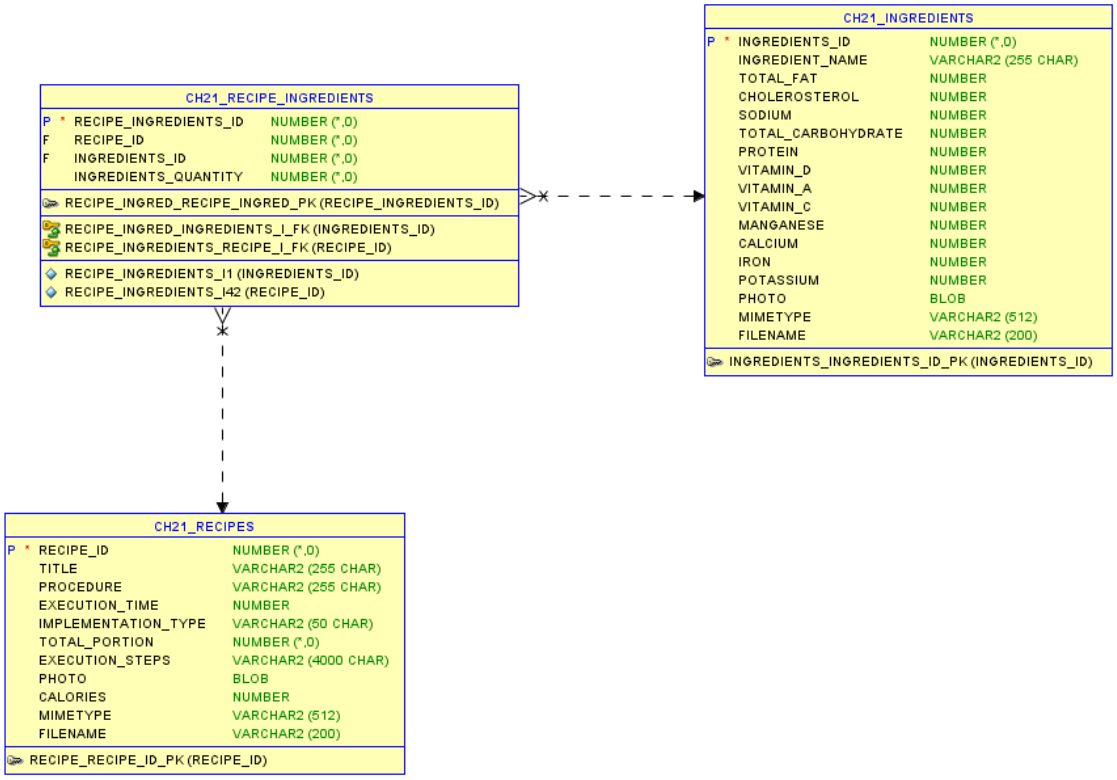
Ključna riječ	**Vrijednost**
ID:	*Ch21-01*
Naslov:	*Platforma nudi pristup raznim unaprijed pripremljenim receptima i sastojcima, kao i opciju dodavanja ili uklanjanja sastojaka kako bi se recepti prilagodili individualnim preferencijama. Korisnici mogu izraditi novi personalizirani recept odabirom željenih sastojaka, a usluga će izračunati hranjive podatke za recept.*
Opis:	*Pomoću aplikacije Apex korisnici mogu pregledavati i birati recepte te također konfigurirati podatke o receptima kao što su stvaranje, ažuriranje i brisanje sastojaka. I recepti i sastojci definirani su skupom atributa. Korisnici mogu stvoriti novi recept dodavanjem ili uklanjanjem sastojaka i izračunati ažurirane podatke o hranjivim vrijednostima. Rezultati se prikazuju u dijagramima koji se ažuriraju u stvarnom vremenu na temelju promjena unesenih u recept.*
Primarna uloga:	*Korisnik*
Preduvjeti:	Korisnik ima račun za APEX instancu.
Postuvjeti:	Nakon ažuriranja i odabira recepta korisnik može izračunati činjenice o sastojcima. Nakon izračuna nutritivnih podataka za recept, korisnici mogu ponoviti postupak za druge recepte dodavanjem ili uklanjanjem sastojaka za izradu novih personaliziranih recepata.
Glavni	Scenarij
Uspješan scenarij:	1. Otvaranje web preglednika i prijava u aplikaciju. 2. Odabir **recipes** ili **ingredients** opcija u izborniku ili stranici za navikaciju 3. Dodavanje recepta, klikom na gumb **add a recipe**. Za dodavanje sastojka potrebno je kliknuti na gumb **Add Ingredient**. 4. Za uređivanje recepta ili sastojka potrebno je kliknuti na simbol olovke na određenom unosu. 5. Nakon bilo kakvih promjena, potrebno je kliknuti gumb **Apply Changes** da se iste spreme. Za brisanje podataka potrebno je klikniti na gumb **Delete**. 6. Odabir opcije **Recipes** ili **Ingredients** iz izbornika ili stranice za navigaciju. 7. Za dodavanje recepta potreban je klik na gumb **recipe** i odabir nekog od dostupnih gotovih recepata. 8. Za dodavanje sastojaka u recept potrebno je kliknuti gumb**Ingredients** i odaberati s popisa. 9. Za provjeru hranjivih podataka u receptu, potrebno je ići na izbornik ili stranicu za navigaciju i odaberati**Recipe total nutrition facts**. 10. Navedeni podaci o nutritivnoj vrijednosti uključuju veličinu porcije hrane koja se može pripremiti na temelju određenih količina sastojaka, ukupan broj proteina i kalorija u receptu te ukupno vrijeme potrebno za pripremu recepta.
Ekstenzije:	*Nema*
Frekvencija korištenja:	*Web stranica u prosjeku objavi 5 recepata dnevno, odnosno oko 1000 recepata godišnje, a stvarni broj ovisi o posjećenosti stranice.*
Status:	Završeno
Owner:	Korisnik
Prioritet:	*Umjeren*



Slika 21.2: Logički dijagram

21.4.2 Relacijski model podataka

Dijagram relacijskih podataka prikazan je na Slici 21.3.



Slika 21.3: Relacijski dijagram

21.4.3 QuickSQL i SQL skripta

Čini se da je pruženi kod skup uputa za generiranje oglednih podataka pomoću alata Quick SQL. Tablice koje trebamo izraditi za našu aplikaciju su CH21_RECIPES,CH21_RECIPE_INGREDIENTS, and CH21_INGREDIENTS. Polja za svaku tablicu trebaju biti strukturirana na sljedeći način:

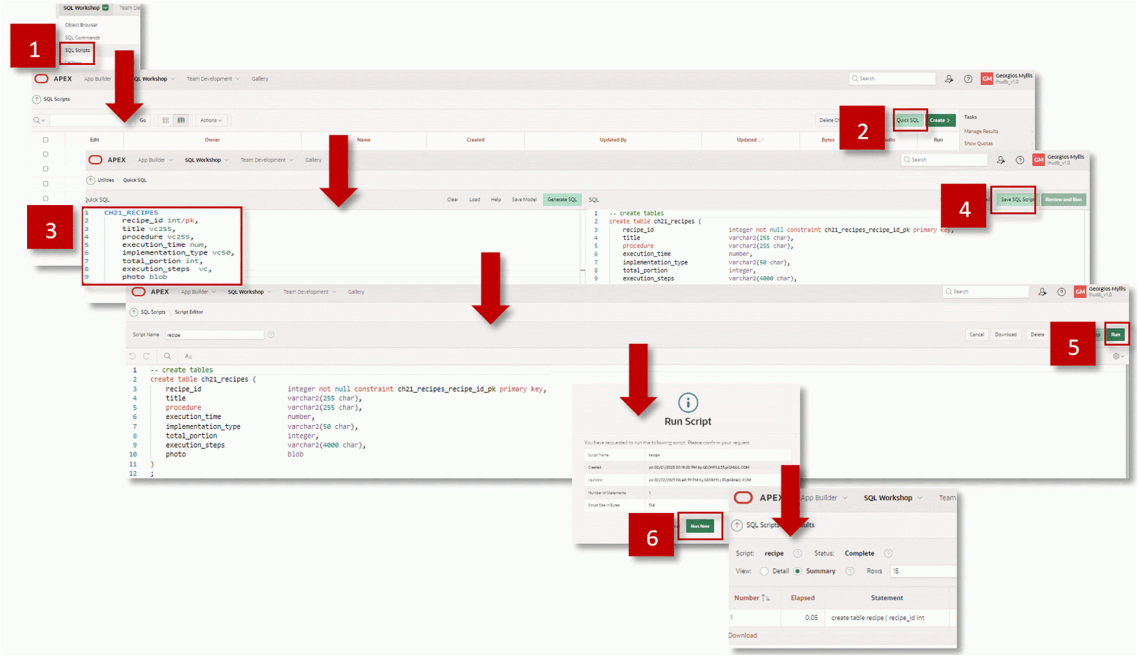
```
CH21_RECIPES
    RECIPE_ID int/pk,
    TITLE VC255,
    PROCEDURE VC255,
    EXECUTION_TIME NUM,
    IMPLEMENTATION_TYPE VC50,
    TOTAL_PORTION int,
    EXECUTION_STEPS  VC,
    PHOTO BLOB

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
```

Za pokretanje Quick SQL koda, prvo treba pristupiti **SQL Workshop** i odabrati **SQL Scripts**. Odatle se može odabrati **Quick SQL** za izradu tablica. Proces je jednostavan; samo treba navesti naziv tablice i njena polja, kao i sve podređene tablice povezane s tablicom. Slika 21.4 prikazuje kako brzo kreirati tablice pomoću ove značajke.



Slika 21.4: Brzi SQL postupak za izradu tablica.

U sljedećem koraku, da bi se osiguralo ispravno funkcioniranje naših tablica, potrebno je postaviti da je svaki dodatak tablici jedinstven i da se primarni ključ svake tablice automatski povećava. Da bismo to postigli, moramo stvoriti tri SEKVENCIJE.

Povećanje ID recepta

```
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;
```

Povećanje ID sastojka

```
CREATE SEQUENCE seq_INGREDIENTS
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;

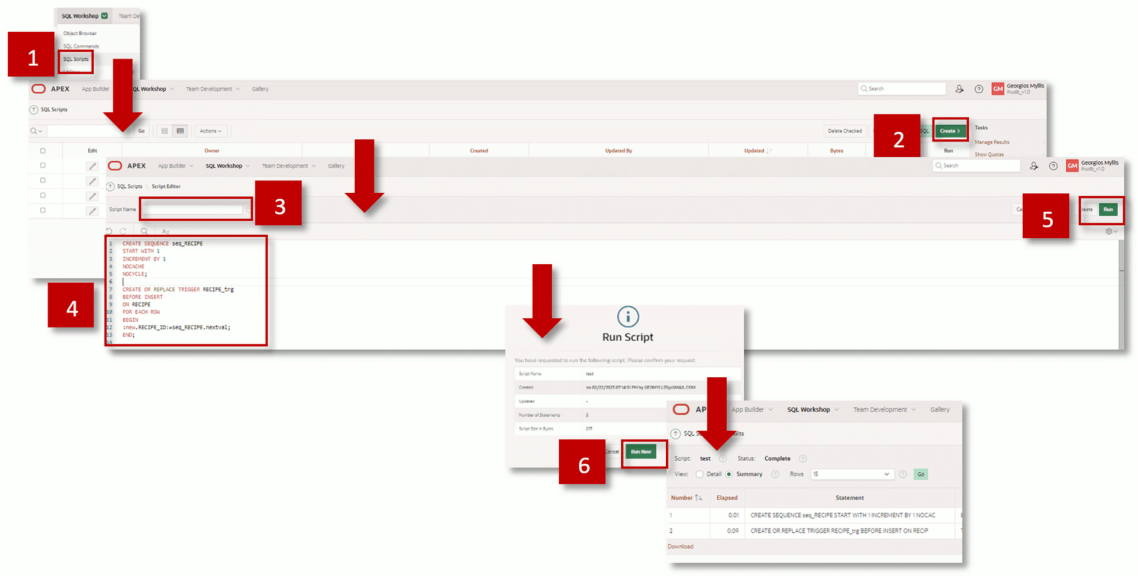
CREATE OR REPLACE TRIGGER INGREDIENTS_trg
BEFORE INSERT
ON INGREDIENTS
FOR EACH ROW
BEGIN
:new.INGREDIENTS_ID:=seq_INGREDIENTS.nextval;
END;
```

Povećanje ID sastojka recepta

```
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;
```

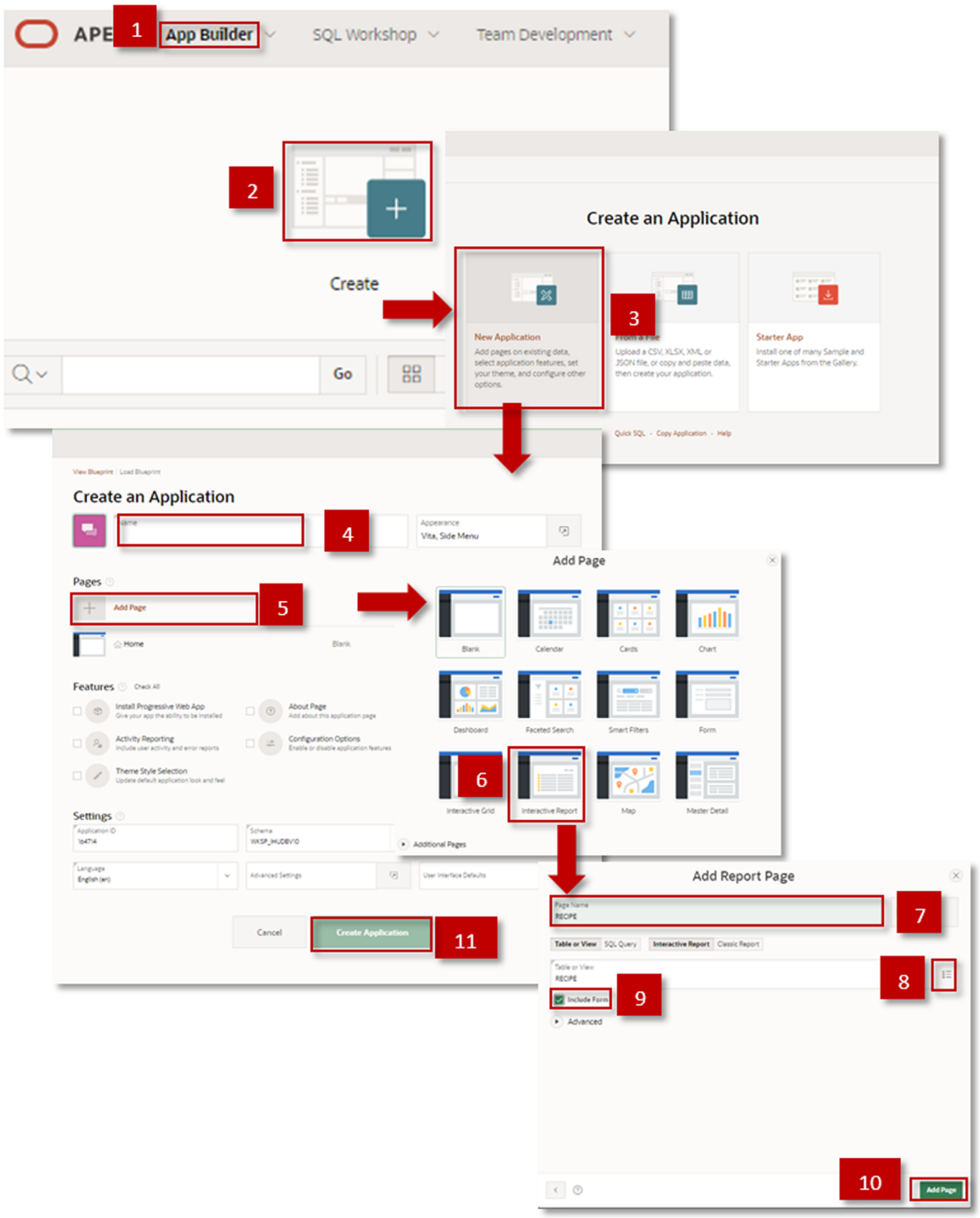
Da bi se pokrenulo SEQUENCE, potrebno je otići do **SQL Workshop** i kliknuti na **SQL Scripts**. Zatim, potrebno je kliknuti na gumb **Create** i kopirati te zalijepiti kod za svaku SEKVENCIJU zasebno. Nakon dodavanja naziva skripti, potrebno je kliknuti na gumb **Run** za izvršavanje skripte. To osigurava da je svaki zapis u tablicama jedinstven i eliminira potrebu za ručnim praćenjem zadnjeg dodanog zapisa. Sve će biti učinjeno automatski. Primjer kako pokrenuti SEQUENCE u APEX okruženju prikazan je na Slici 21.5.



Slika 21.5: Procedura SQL skripte za izgradnju SEQUENCE

21.5 Sučelja aplikacije

Za kreiranje aplikacije kao na slici 21.6, potrebno je kliknuti na **App Builder** te klik na **Create**. Tada će se pojaviti novi prozor u kojem moramo dati naziv aplikacije i odabrati značajke koje želimo da aplikacija ima. Tablica koju će se koristiti za aplikaciju je: RECIPE, RECIPE INGREDIENTS, INGREDIENTS as explained earlier.



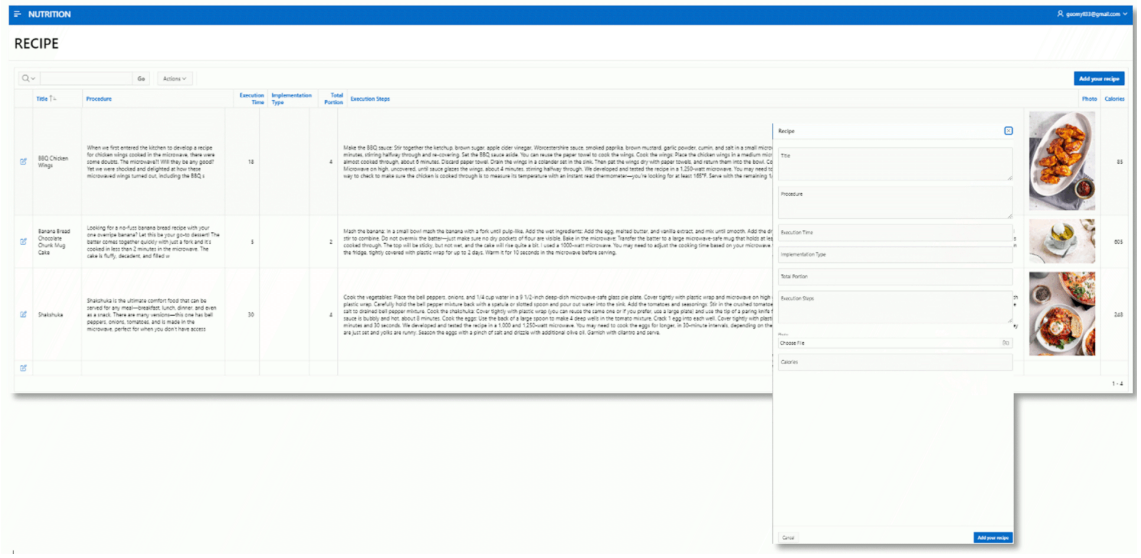
Slika 21.6: Stvaranje nove aplikacije

Na Slici 21.7, stvoreno je novo okruženje App Builder za novu aplikaciju.Za pristup aplikaciji, potrebno je klikniti na nju i unijeti svoje administratorske vjerodajnice koje bi trebale biti iste kao one za radni prostor. Nakon prijave, moguće se kretati svojom aplikacijom pomoću mega izbornika koji se nalazi na vrhu.



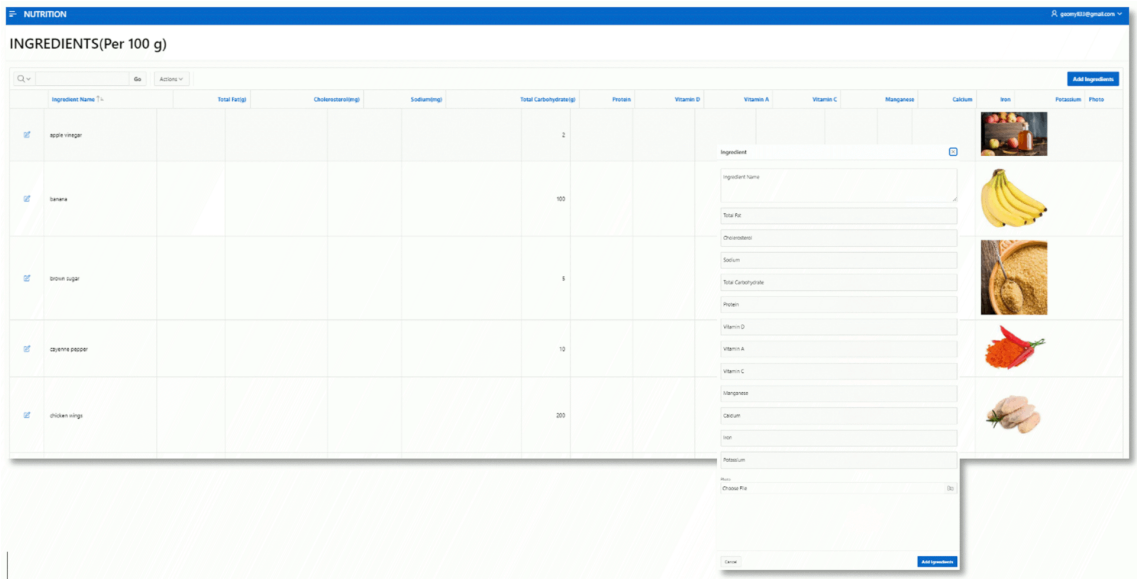
Slika 21.7: Aplikacija za prehranu

Na Slici 21.8 je stranica s receptom te korisnici mogu pristupiti svim potrebnim informacijama za kuhanje recepta, kao što su postupak, vrijeme izvršenja, ukupne porcije, koraci izvršenja i fotografija željenog ishoda. Ovisno o svojim dopuštenjima, korisnici također mogu dodati novi recept ili ažurirati postojeći prema svojim željama.



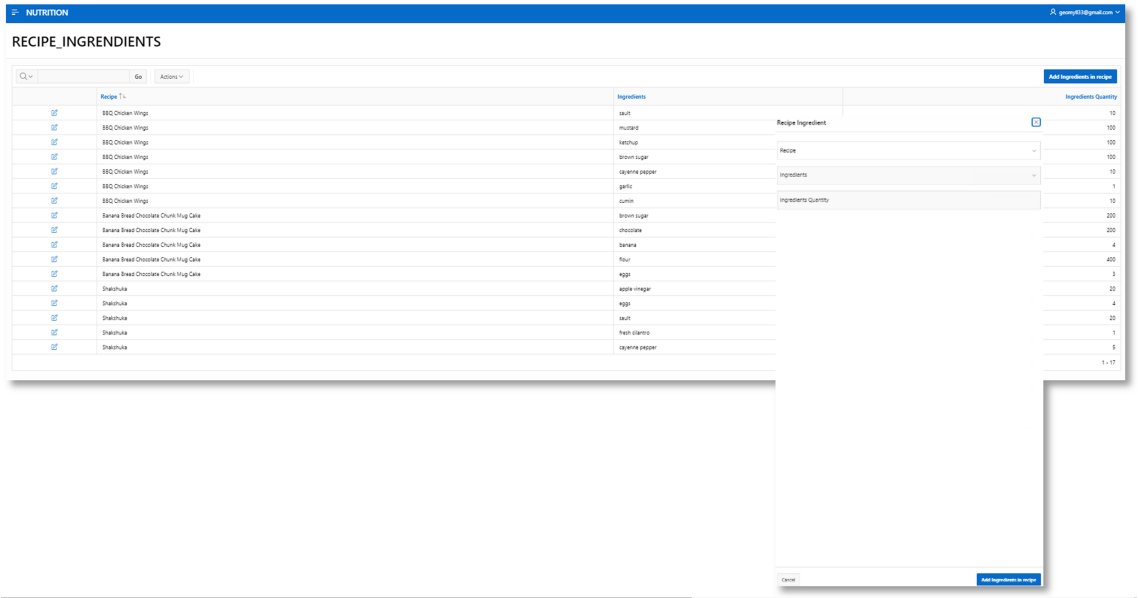
Slika 21.8: Aplikacija za prehranu - stranica s receptima

Na Slici 21.9, stranica sa sastojcima prikazuje različite atribute sastojka kao što su ukupna masnoća, kolesterol, natrij, ukupni ugljikohidrati, proteini itd. Ovi se atributi koriste za izračun nutritivnih činjenica Recepti. Korisnici s odgovarajućim pravima mogu održavati stranicu dodavanjem novih sastojaka ili ažuriranjem postojećih prema potrebi.



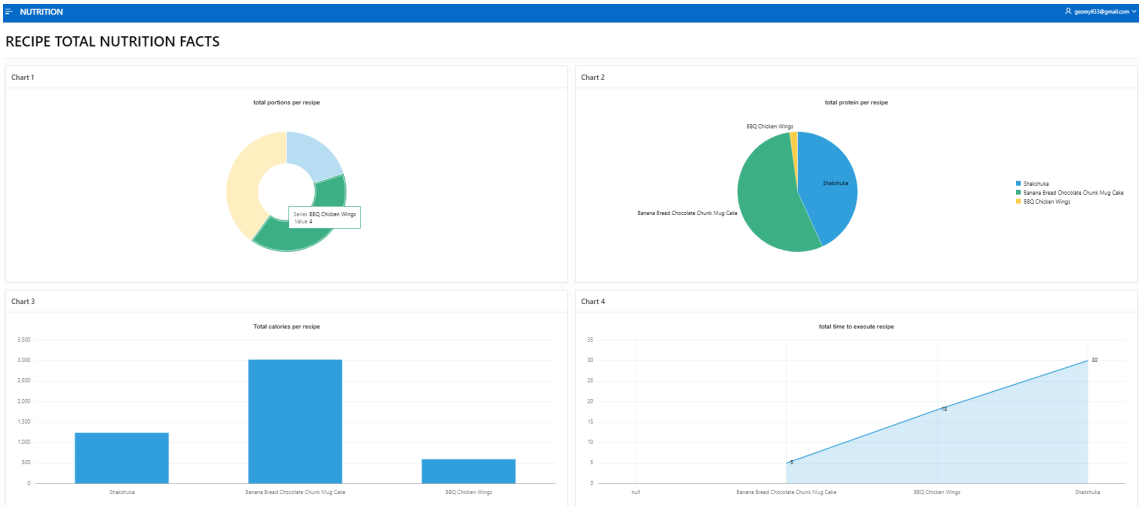
Slika 21.9: Aplikacija za prehranu - stranica sa sastojcima

Na Slici 21.10, stranica sa sastojcima recepata pruža korisnicima prednosti ove web usluge, dopuštajući im da kreiraju recepte prema svojim potrebama ili modificiraju postojeće, a zatim provjeravaju podatke o hranjivim vrijednostima. Korisnici s odgovarajućim dopuštenjima na ovoj stranici mogu dodavati ili uklanjati sastojke iz recepata.



Slika 21.10: Aplikacija za prehranu - stranica sa sastojcima recepta

Na Slici 21.11, stranica s izvješćima o nutritivnim činjenicama pruža korisniku sve potrebne informacije o receptu koji je izradio. Navedene su informacije poput ukupnog broja proteina u receptu, ukupnih kalorija u receptu, broja porcija koje recept poslužuje i koliko je vremena potrebno za kuhanje recepta.



Slika 21.11: Aplikacija za prehranu - stranica s podacima o prehrani

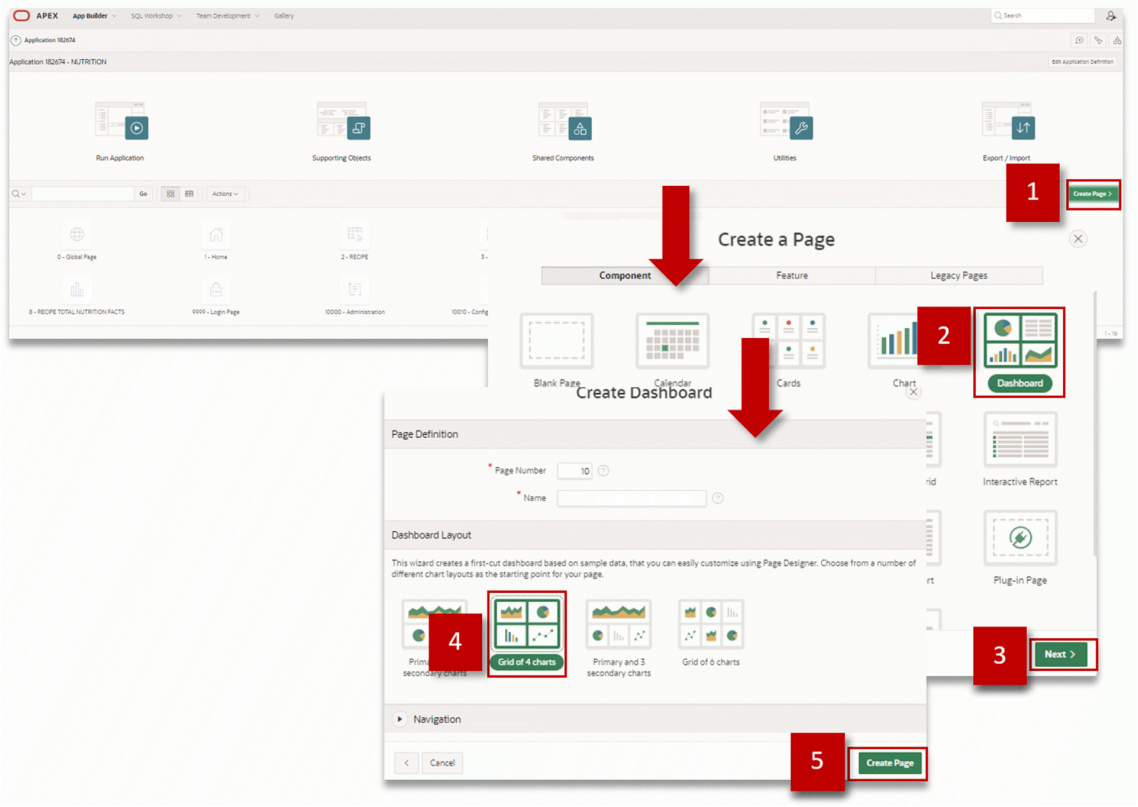
21.5.1 Kako izraditi izvještaj

Za izradu početne stranice aplikacije, potrebno je slijediti korake prikazane na Slici 21.12.

- Klik na okruženje vaše aplikacije.
- Klik na gumb **Create Page**.
- Odabir **Dashboard** s popisa dostupnih vrsta stranica.
- Klik na gumb **Next**.
- Odabir željenog izgleda početne stranice.
- Klik na gumb **Create Page**kako bi se kreirala stranica.

Ovi koraci će vas voditi kroz proces stvaranja početne stranice za vašu aplikaciju. Nakon što je

stvorena, možete prilagoditi početnu stranicu dodavanjem widgeta i konfiguriranjem izgleda za prikaz podataka koji su vam potrebni za praćenje i upravljanje vašom aplikacijom.

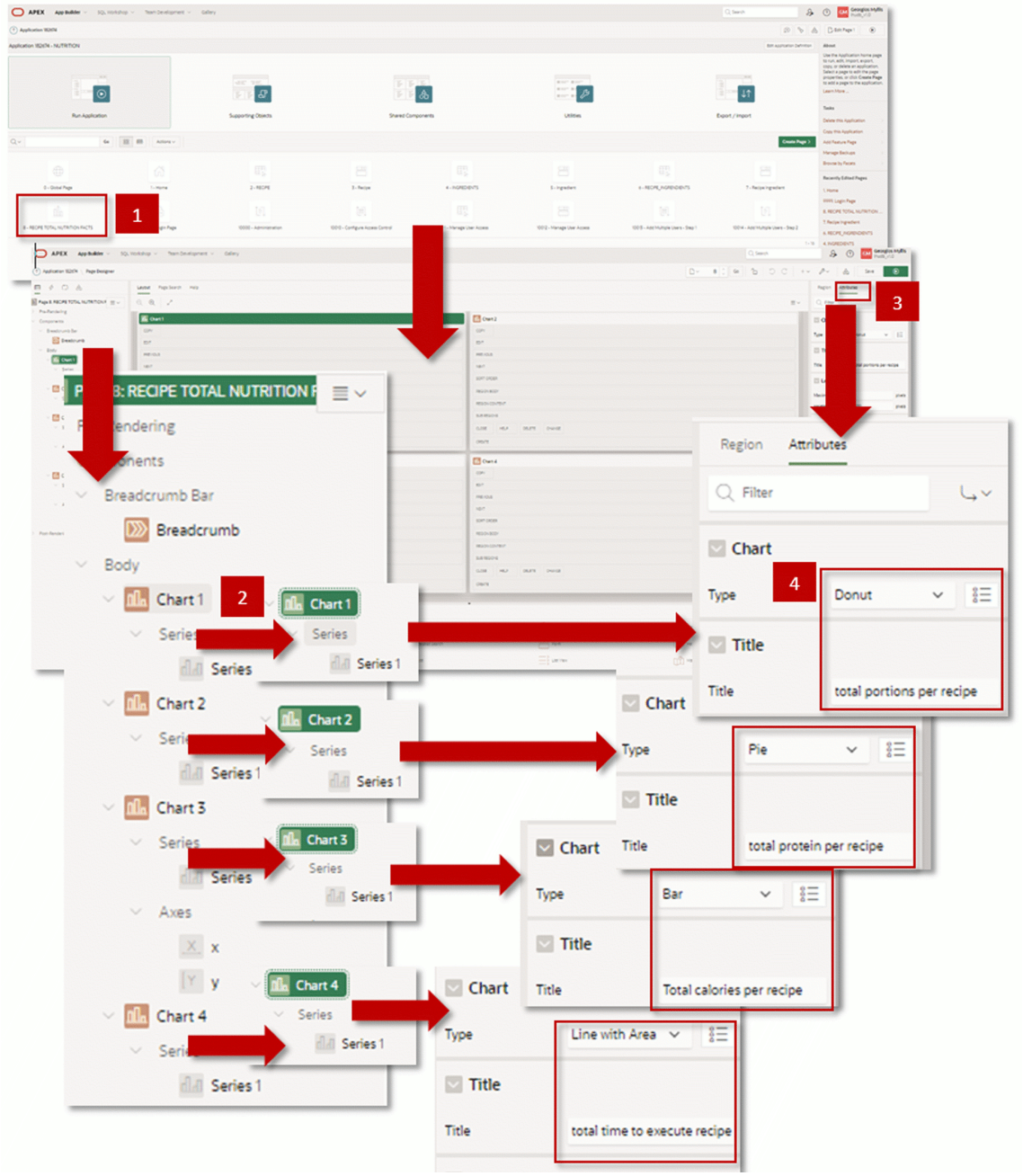


Slika 21.12: Stvorite početnu stranicu nadzorne ploče za svoju aplikaciju

Potrebno je ići na izvješće nadzorne ploče, koje se može pronaći u vašem okruženju aplikacije, i ući u dizajner stranice. Kao što je prikazano na Slici 21.12, odabrana su četiri grafikona i svakom su dodani različiti podaci pomoću SQL upita.

Za početak odaberite Grafikon 1 i preimenujte ga u "Total Portion per Recipe". Zatim odaberite **Attribute** i odaberite vrstu prstenastog grafikona. Zatim odaberite Grafikon 2 i preimenujte ga u "Total Protein per Recipe", što će prikazati recepte s najviše proteina. Za ovaj grafikon odaberite vrstu **Pie Chart**.

Prelazeći na Grafikon 3, preimenujte ga u "Total Calories per Recipe". Ovaj grafikon će prikazati ukupan broj kalorija za svaki recept, a koristit će se **Bar Chart type**. Na kraju, preimenujte Grafikon 4 u "Total Time to Execute Recipe". Odabir **Attributes** a zatim odaberite vrstu grafikona **Line with Area** (vidi Sliku 21.13).



Slika 21.13: Konfigurirajte naslov i vrstu razgovora na izvješćima nadzorne ploče

Zatim klik na **Series 1** Grafikon 1 i u Izvor postavite sljedeće postavke:

```
SELECT DISTINCT RECIPE.TITLE,
RECIPE.TOTAL PORTION
FROM RECIPE
```

Na Seriji 1 Grafikona 2 pod Izvor postavite sljedeće postavke

```
SELECT DISTINCT RECIPE.TITLE, SUM(protein) AS protein
FROM RECIPE
INNER JOIN RECIPE_INGREDIENTS ON
RECIPE.RECIPE_ID=RECIPE_INGREDIENTS.RECIPE_ID
```

```
INNER JOIN INGREDIENTS ON
INGREDIENTS.INGREDIENTS_ID=RECIPE_INGREDIENTS.INGREDIENTS_ID
GROUP BY RECIPE.TITLE
```

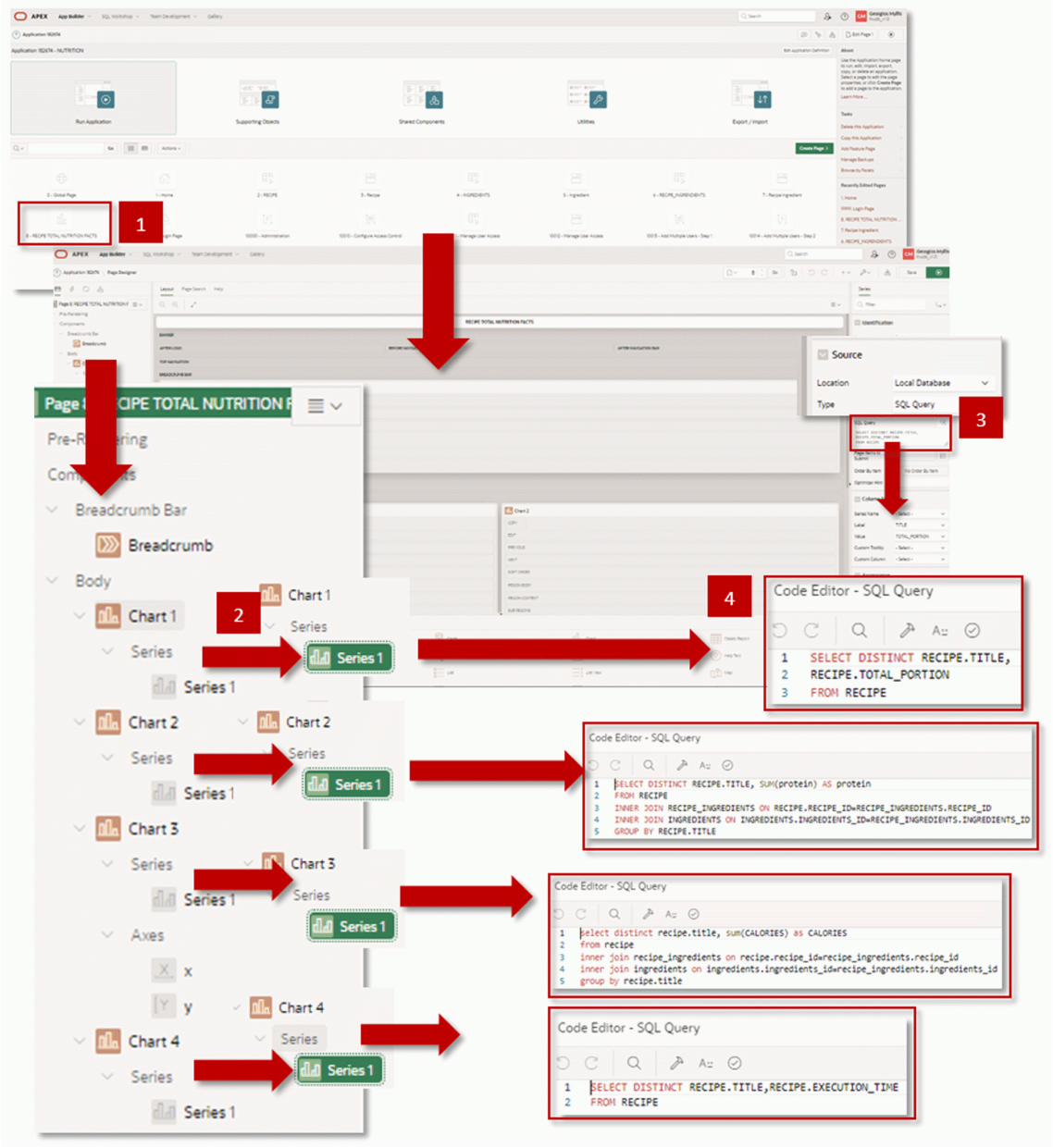
Na Seriji 1 Grafikona 3 pod Izvor postavite sljedeće postavke

```
select distinct recipe.title, sum(CALORIES) as CALORIES
from recipe
inner join recipe_ingredients on
recipe.recipe_id=recipe_ingredients.recipe_id
inner join ingredients on
ingredients.ingredients_id=recipe_ingredients.ingredients_id
group by recipe.title
```

i na Seriji 1 Grafikona 4 pod Izvor postavite sljedeće postavke

```
SELECT DISTINCT RECIPE.TITLE,RECIPE.EXECUTION_TIME
FROM RECIPE
```

Slika 21.14 daje neke smjernice kako ga konfigurirati.

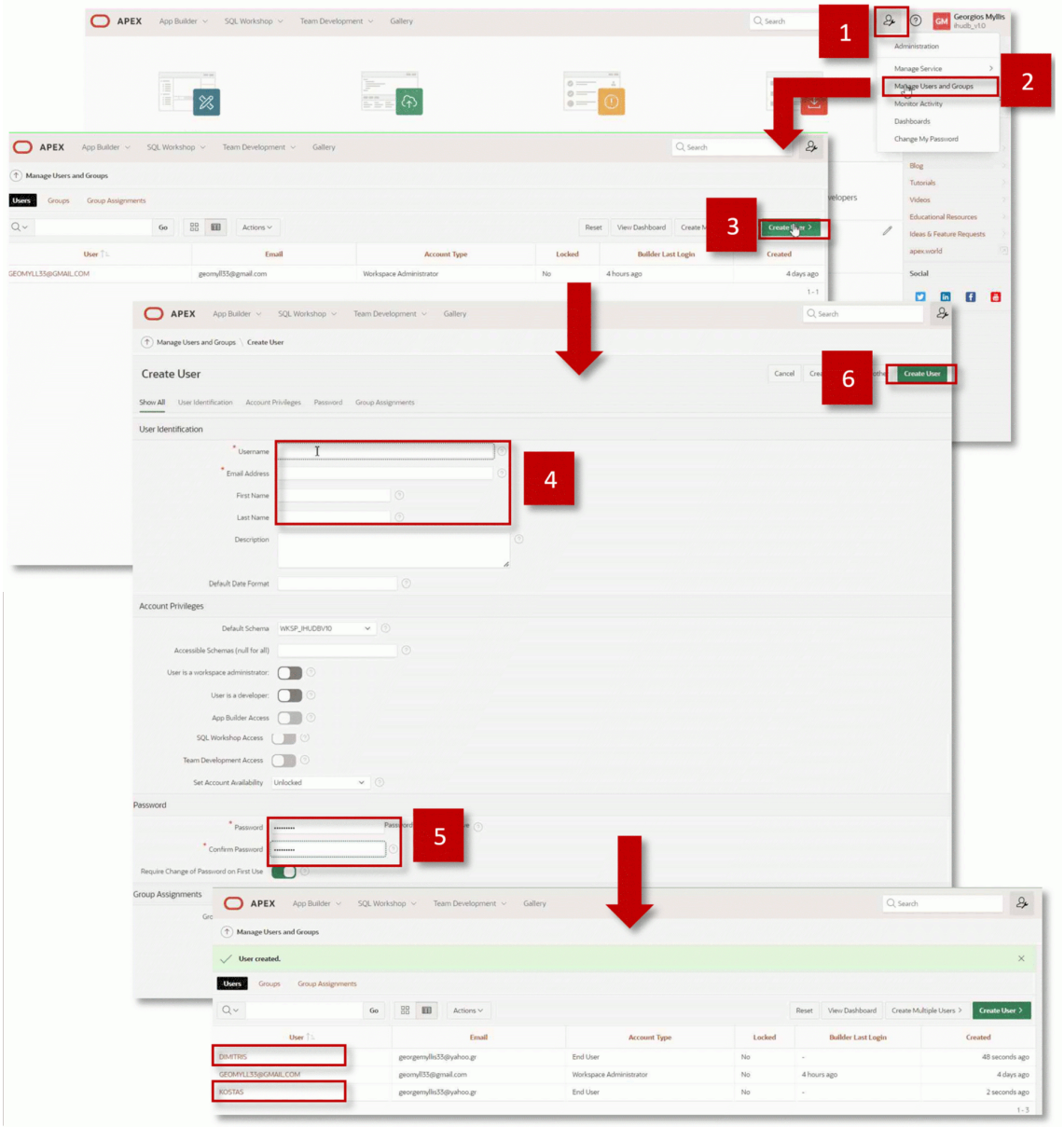


Slika 21.14: Konfigurirajte SOURCE sa SQL QUERY

21.5.2 Stvoranje pristupa korisniku

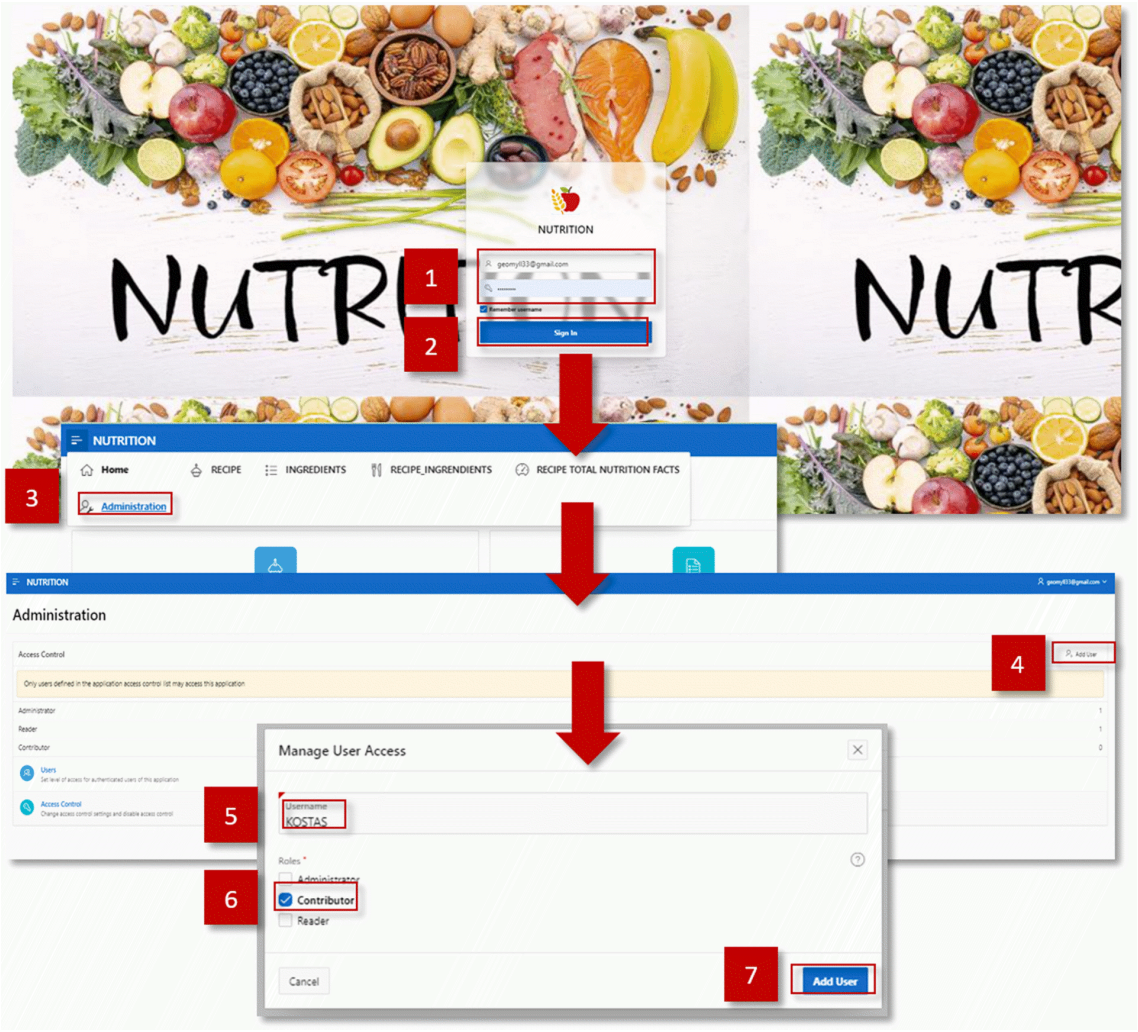
Oracle APEX pruža 3 ugrađene privilegije koje vam omogućuju kontrolu pristupa aplikaciji i njezinim komponentama. Ove povlastice su Administracija, Uređivanje i Pregled, a svaka je povezana s određenom pristupnom ulogom. Administracija je u korelaciji s ulogom administratora, uređivanje je u korelaciji s ulogom suradnika, a prikaz je u korelaciji s ulogom čitača.

Da biste korisniku dodijelili pristup aplikaciji, prvo morate kreirati korisnički račun. Da biste to učinili, idite na ikonu Administracija na gornjoj traci kao što je prikazano na Slici 21.15, predstavljenu simbolom čovjeka i ključa, i odaberite, **Manage Users and Groups** s padajućeg izbornika. Zatim odaberite **Create User**, unesite korisničku identifikaciju i lozinku i kliknite na **Create User** da dovršite postupak.



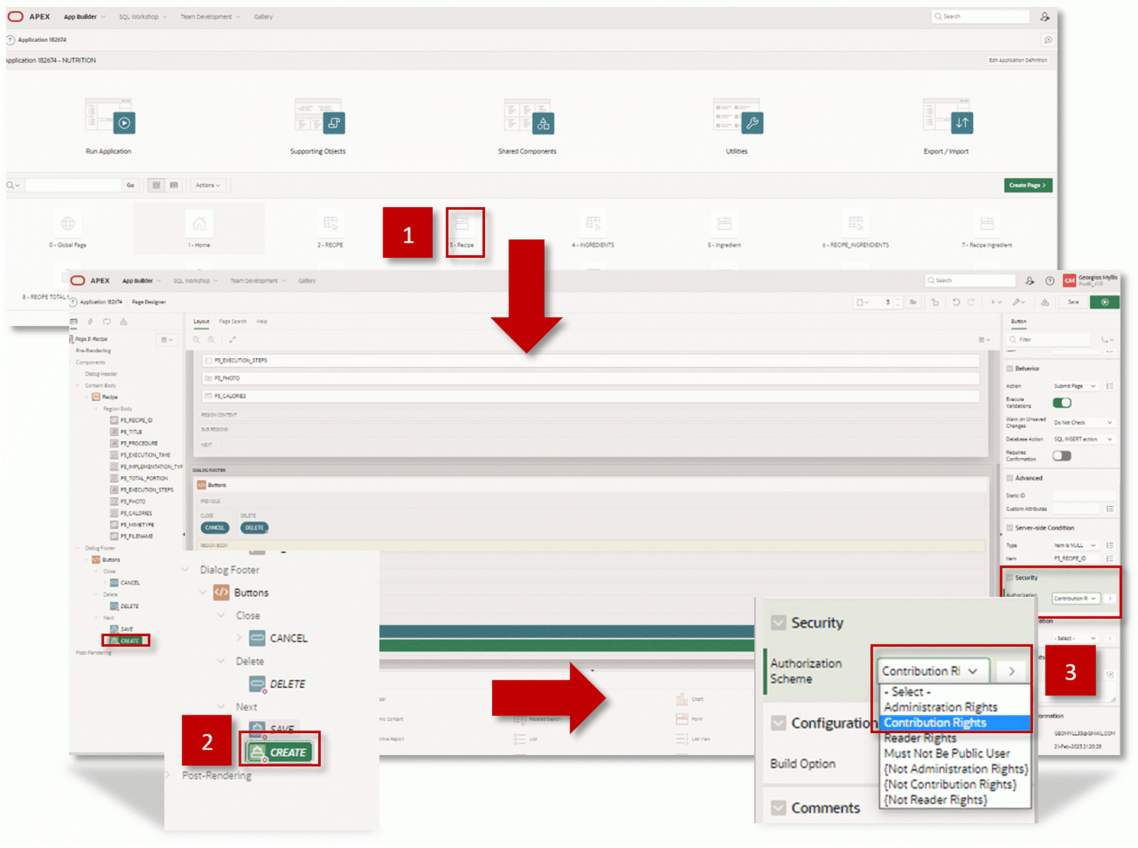
Slika 21.15: Kreiranje korisničkog računa

Jedan od načina za definiranje korisničkih uloga je kroz samu aplikaciju. Najprije se potrebno prijaviti u aplikaciju pomoću administratorskih vjerodajnica. Zatim slijedi pristup mega izborniku i potrebno je kliknuti na **Administration**. Nadalje, potreban je klik na **Add User**, te unos za identifikaciju i ulogu korisnika i na kraju klik na **Add User**. Možete pogledati Sliku 21.16 za više smjernica.



Slika 21.16: Kreiranje korisničkog računa

Na Slici 21.17, potrebno je ići do okruženja radnog prostora aplikacije i kliknuti na obrazac **RECIPE**. U dizajneru stranice za Form, odabire se gumb **Create**, a u sigurnosnim opciSlijedi klik na **Save**. Ovo će osigurati da samo korisnici s ulogom doprinosa mogu stvarati i dodavati recepte.



Slika 21.17: Konfiguriranje stranice s korisničkim pravima

21.6 Dodatni materijal za učenje

Možete pronaći sljedeće dodatne materijale za učenje:

- izvezena aplikacija,
- video upute.

Sav dodatni materijal za učenje dostupan je napublic BeeAPEX project page. Prijavite se kao gost (lozinka nije potrebna). Navigirajte do sekcije Books, skripte u mapi Part 2 > Poglavlje21 u sekciji Scripts i video upute u Collection of video guides. Materijal za kratke tečajeve je u Short courses section.

21.6.1 Izvezena aplikacija

Izvezena aplikacija je zapakirana. Instalacija stvara tablice, indeks, funkciju, proceduru i okidač te popunjava podatke. Deinstalacija uklanja sve objekte baze podataka koji se koriste u ovoj aplikaciji.

21.6.2 Video upute

Video upute prikazuju svaki korak u razvoju aplikacije.

21.7 Pitanja

1. Kako možete promijeniti pozadinu na stranici za prijavu?
2. Gdje još možemo pronaći upravljanje ulogama u Oracle Apex okruženju ako ono nije omogućeno u našoj aplikaciji?
3. Kako možete konfigurirati naziv gumba izvješća?
4. Kako možete promijeniti naziv stranice nakon što ste je izradili?

•

21.8 Odgovori

1. U okruženju svoje aplikacije odaberite **Shared Components** te zatim odaberite **Static Application Files** i kliknite na **Create File** i u skočnom prozoru dodajte sliku po želji.Zatim kopirajte referentni naziv slike , vratite se u okruženje svoje aplikacije i odaberite **9999-Login Page**, idite na css polje i odabrite vrstu inline.

```
body {
background-image:
url("#APP_FILES#\textit{the reference name of the image}");
background-size:1700px 900 px:
}
```

2. U okruženju vaše aplikacije odaberite **Shared Components** zatim odaberite **Application Access control**.
3. Za promjenu oznake gumba **Create** u drugi naziv, trebate ući u Page Designer za izvještaje, locirati gumb **Create**, i promijeniti oznaku u polju Identification.
4. Za promjenu naziva stranice koju ste izradili, prijavite se u svoju aplikaciju i kliknite na **Quick Edit**. Zatim kliknite na naslov stranice i odaberite strelicu u izvornom području puta kroz web stranicu. Zatim kliknite na **Edit Component**, i u polju **Short Name** žurirajte ime na željeno novo ime. Na kraju kliknite na **Apply Changes**, novi naziv će se pojaviti u naslovu stranice.



22. Raspored konzultacija

J. MAŃKO, M. SOŃTA, R. LESKOVAR

22.1 Poslovni pogled na slučaj

Aplikacija za raspored, odnosno zakazivanje vremena konzultacija na sveučilištu može se iz poslovne perspektive promatrati kao alat za pojednostavljenje i optimizaciju procesa raspoređivanja za akademsko osoblje. Može pomoći u poboljšanju učinkovitosti rasporeda, smanjenju administrativnog opterećenja, povećanju zadovoljstva studenata i pružanju informacija o dostupnosti nastavnika i asistenata u stvarnom vremenu. Aplikacija također može generirati podatke i uvid u korištenje termina konzultacija, što može pomoći donošenje odluka i raspodjelu resursa. Osim toga, aplikacija je praktična i pristupačna platformu rasporeda i rezerviranja vremena za studente, pomažući poticanju proaktivnije i angažiranije studentske zajednice. Akademsko osoblje dužno je tjedno studentima staviti na raspolaganje 60 minuta svog vremena, što se interno naziva "konzultacijama" ili "uredovnim satima za studente". Dosta se često događa, posebice kako se bliži kraj semestra, da se upiti studenata intenziviraju brojem i opsegom. U tom slučaju 60 minuta tjedno nije dovoljno, budući da studenti žele što prije dobiti traženi odziv akademskog osoblja. Iako postoje neke standardizirane teme za raspravu kao što su: diplomski rad, ispiti, završni zadaci, neki od slučajeva odnose se na nevezane teme o kojima bi možda valjalo razgovarati, ali ne nužno tijekom konzultacija s nastavnikom (npr., administrativni zahtjevi ili problemi koji se mogu lako riješiti bez intervencije nastavnika). Posljednjih godina, a posebno zbog pandemije COVID-19, sveučilišta su prošla radikalnu tehnološku tranziciju koja je omogućila različite oblike nastave na daljinu, koristeći dostupne platforme i komunikacijske kanale koji olakšavaju proces online nastave. Zapravo, gotovo sve akademske aktivnosti prebačene su na internet uključujući, naravno, komuniciranje na daljinu tijekom konzultacija. Iako postoji suglasje u akademskoj zajednici da nastava na daljinu nikada neće u potpunosti zamijeniti nastavu licem u lice, nastava i drugi oblici rada na daljinu i nadalje se primjenjuju. Za očekivati je da će neke sveučilišne aktivnosti i procesi u bliskoj budućnosti u potpunosti prijeći na mrežni način rada. Jedan od ciljeva je učinkovito uredsko poslovanje bez papira. Sve navedeno dodatno naglašava potrebu za održivim digitalnim alatom koji bi omogućio online rezerviranje uredovnog vremena za konzultacije i aktivnosti koje se odvijaju online ili offline. Jasno su potencijalne koristi akademske zajednice od takve aplikacije, koja bi uskoro mogla postati svakodnevni alat, kao što je elektronička pošta, a ne neki digitalni hir.

22.2 Definicija problema

U sadašnjem sustavu nije moguće unaprijed rezervirati (upisati) termin i planirati sadržaj konzultacija. Zapravo, studenti, bez obzira na to koliko ih se pojavi, usluženi su po principu tko prvi dođe, prvi je poslužen. To ne samo da povećava frustraciju studenata, koji moraju čekati u redu bez ikakve sigurnosti da će se njihove konzultacije uopće održati, nego i za akademsko osoblje koje mora birati između neizvjesnog vremena čekanja studenata, ili pomicanja termina na drugi dan. Kao rezultat toga, dolazi do situacija u kojima akademsko osoblje održava pro bono konzultacije za studente, čak nauštrb svog privatnog života, samo kako bi izbjegli da ih studenti percipiraju kao nedostupne ili neprijateljske. Nadalje, ovaj status quo dovodi do neučinkovite raspodjele vremena i za studente i za akademsko osoblje, kao i do nepotrebnog opterećenja rasporeda.

Studenti se često obraćaju studentskoj referadi s pitanjima o datumima i mjestima održavanja konzultacija. Budući da nema pouzdanog izvora, oslanjaju se na iskustva kolega koji ih često upućuju na pogrešna mjesta u krivo vrijeme. Prema pravilima akademske ustanove, svaku promjenu termina konzultacija treba odobriti prodekan. Suglasnosti se daju u izravnoj komunikaciji između nastavnika i prodekana, a studentska referada o tome nema informacija. Prodekan stoga želi da studentska referada unosi podatke o promjeni termina konzultacija.

Uprava ima loš uvid u širu sliku korištenja radnog vremena u akademskoj instituciji. Glasine i druge nepouzdane informacije nisu dobar temelj za racionalne odluke o povećanju ili smanjenju broja radnih sati. Navedeno bi također trebalo povezano s nagrađivanjem nastavnika za njihove dodatne napore.

22.3 Slučajevi korištenja

Aplikacija za rezervaciju termina konzultacija omogućava pristup menadžmentu, osoblju studentskih referada, studentima i akademskom osoblju (nastavnicima), od kojih svaki ima različite privilegije:

- Menadžment dobiva sažete informacije prikupljene na upravljačkoj ploči (broj sati konzultacija po godini i mjesecu, 10 nastavnika s najviše studenata, 5 nastavnika s najviše rezervacija koje ne mogu odraditi i 5 nastavnika s najmanje ili bez zakazanih konzultacija).
- Osoblje studentske referade pristupa interaktivnom izvješću o terminima konzultacija i mijenja raspored konzultacija.
- Studenti imaju pristup kalendaru vlastitih termina konzultacija, kalendaru svih konzultacija koje nude nastavnici na studijskom programu gdje studiraju, rezervacijama konzultacija putem kalendara i detaljnom interaktivnom izvješću o upisu zahtjeva za konzultacijama.
- Akademsko osoblje (nastavnici) mogu dobiti detaljan interaktivni izvještaj o konzultacijama nastavnika (dežurstva), kalendar s uredovnim satima za konzultacije nastavnika (dežurstva) i mogu promijeniti raspored sati bez upisa.

22.3.1 Narativni opis

Radi sažetosti, od svih slučajevva korištenja, prikazat ćemo samo dva:

- Promjena rasporeda konzultacija za termine gdje nema upisanih zahtjeva studenata, zahtijeva od nastavnika da se prijavi, pregleda zakazane termine bez upisanih studenata i premjesti jedan po jedan termin na drugo vrijeme.
- Upis zahtjeva studenta za konzultacijama sastoji se od pregleda kalendara (svi termini konzultacija koje nude svi nastavnici na studijskom programu studenta), odabira željenih termina konzultacija, unosa svrhe konzultacija i potvrde upisa. Nije dopušteno više od jednog upisa po jednom raspoloživom terminu konzultacija.

Tablica 22.1: Opis slučaja korištenja: preraspoređivanje termina konzultacija od strane nastavnika

Ključna riječ	**Vrijednost**
ID:	*Ch22-01*
Naziv:	*Preraspoređivanje termina konzultacija od strane nastavnika*
Opis:	*Nastavnik koristi APEX aplikaciju za promjenu rasporeda termina konzultacija za koje nema upisanih studenata.*
Primarni učesnik:	*nastavnik*
Preduvjeti:	pristup web pregledniku na mobilnom uređaju ili računalu, korisnik ima vjerodajnice i privilegije nastavnika, web stranica aplikacije je dostupna
Uvjeti po završetku:	podaci pohranjeni u tablici baze podataka ch22_duty_hours
Glavni	-
Uspješni scenarij:	1. nastavnik otvara web preglednik i prijavljuje se u aplikaciju, 2. odaberite promjenu rasporeda termina konzultacija bez upisanih studenata, 3. kliknite termin (vremenski odsječak) konzultacija koji ima oznaku da ga nitko nije rezervirao, 4. na obrascu unesite podatke o dovom terminu u rasporedu (mjesto, vrijeme početka i završetka), 5. pregledajte i potvrdite pomicanje klikom na odgovarajući gumb.
Proširenja:	-
Učestalost korištenja:	*Približno 4 od 30 obaveznih termina konzultacija u jednoj akademskoj godini.*
Status:	[Završen]
Vlasnik:	*Korisnik s vjerodajnicama nastavnika*
Prioritet:	*srednji*

22.3.2 Polustrukturirani opis

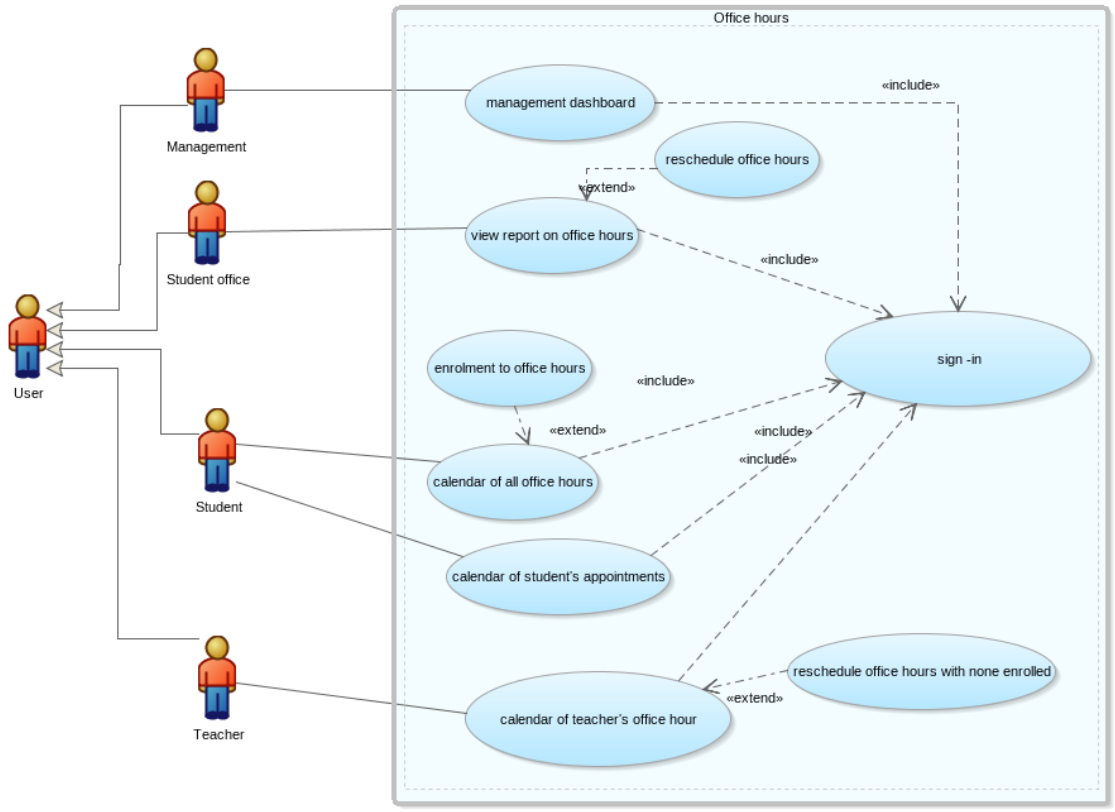
Prvi slučaj korištenja je preraspoređivanje termina konzultacija od strane nastavnika kada se nijedan student ne upiše (pogledajte tablicu 22.1). Ovakvo preraspoređivanje ne utječe na studente. Drugi slučaj korištenja namijenjen je rezervaciji termina konzultacija od strane studenta 22.2). Aplikacija mora spriječiti višestruki upis istog vremenskog razdoblja namijenjenog konzultacijama

22.3.3 Dijagram slučajeva korištenja

Slika 22.1 prikazuje slučajeve korištenja koji su uočeni. Imajte na umu da se slučaj upotrebe "prijava" izvodi samo jednom po korisničkoj sesiji i da se nijedan drugi slučaj upotrebe ne može izvršiti prije uspješnog završetka "prijave". Tri slučaja upotrebe, naime "preraspoređivanje termina konzultacija od strane nastavnika", "upis studenta u termin konzultacija" i "preraspoređivanje termina konzultacija bez upisanih studenata" ovisni su i označeni s "extend". Ovi se slučajevi upotrebe izvršavaju na zahtjev nakon završetka nadređenog slučaja korištenja.

Tablica 22.2: Opis slučaja korištenja: upis studenta u termin konzultacija.

Ključna riječ	**Vrijednost**
ID:	*Ch22-02*
Naziv:	*Upis studenta u termin konzultacija*
Opis:	*Student koristi APEX aplikaciju za upis u termin konzultacija. Kalendar sadrži sva predviđena uredovna vremena konzultacija za sve nastavnike koji su angažirani na studentskim studijskim programima (odsjecima). Obično se jedan student upisuje na jedan studijski program, ali se iznimnim pojedincima može dopustiti da studiraju dva ili više studijska programa. Aplikacija mora spriječiti višestruke upise jednog studenta u određen termin konzultacija.*
Primarni učesnik:	*student*
Preduvjeti:	pristup web pregledniku na mobilnom uređaju ili računalu, korisnik ima vjerodajnice i privilegije nastavnika, web stranica aplikacije je dostupna
Uvjeti po završetku:	podaci pohranjeni u tablicu baze podataka ch22_dh_slot
Glavni	-
Uspješni scenarij:	1. student otvara web preglednik i prijavljuje se u aplikaciju, 2. upišite se za određeni termin u kalendaru, 3. kliknite željeni utor, odnosno vremensko razdoblje konzultacija (boja utora označava statuse), 4. na novom obrascu kliknite gumb i potvrdite upis.
Proširenja:	-
Učestalost korištenja:	*Više od 1000 studenata koristi konzultacije koje drži 10 do 15 nastavnika, prosječno dva puta u semestru.*
Status:	[Završen]
Vlasnik:	*Student*
Prioritet:	*visok*



Slika 22.1: Dijagram slučajeva korištenja

22.4 Model podataka

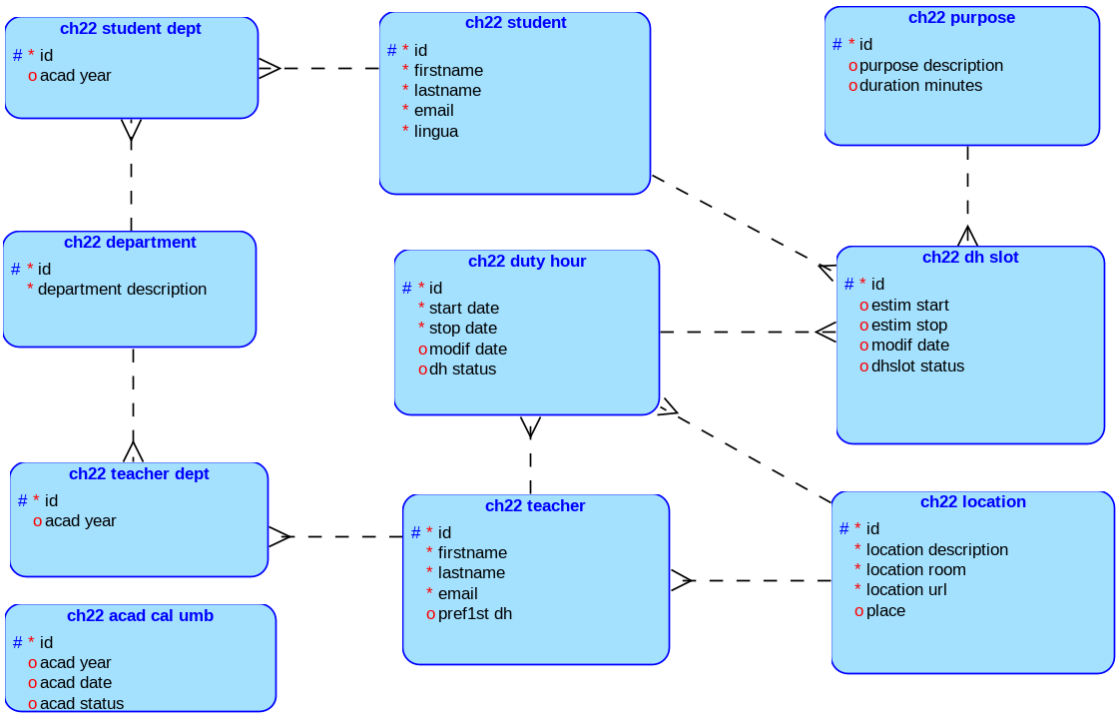
22.4.1 Opis modela podataka riječima

Postoji deset tipova entiteta u logičkom modelu podataka:

- **ch22_dh_slot** predstavlja jedan sastanak (konzultacije) studenta i nastavnika;
- **ch22_duty_hour** predstavlja unaprijed dodijeljeni vremenski odsječak za konzultcije nastavnika i ukazuje na nekoliko vremenskih odsječaka tzv. sastanaka studenta i nastavnika;
- **ch22_purpose** sadrži opis svrhe konzultacija i njihovo trajanje;
- **ch22_student_dept** pokazuje jedan studijski program (sinonim može biti odjel) jednog studenta. Izuzetni studenti mogu biti upisani u više od jednog programa;
- **ch22_student** je osoba koja studira na sveučilištu određeni studijski program;
- **ch22_teacher_dept** pokazuje na jedan studijski program (odsjek) jednog nastavnika. Jedan nastavnik obično je angažiran na više studijskih programa;
- **ch22_teacher** je osoba koja izvodi akademske aktivnosti (nastava, laboratorijske vježbe i sl.);
- **ch22_location** je ili fizička (R) ili virtualna (I) ili oboje (B) - to je prostor gdje se odvijaju konzultacije;
- **ch22_department** je sinonim za studijski program;
- **ch22_acad_cal_umb** sadrži specifikaciju radnih/neradnih dana. Ovaj entitet je povezan sa svim atributima datuma u drugim entitetima, ali nećemo eksplicitno modelirati te relacije jer je implicitna relacija dovoljna za cjelovitost modela .

22.4.2 Logički model podataka

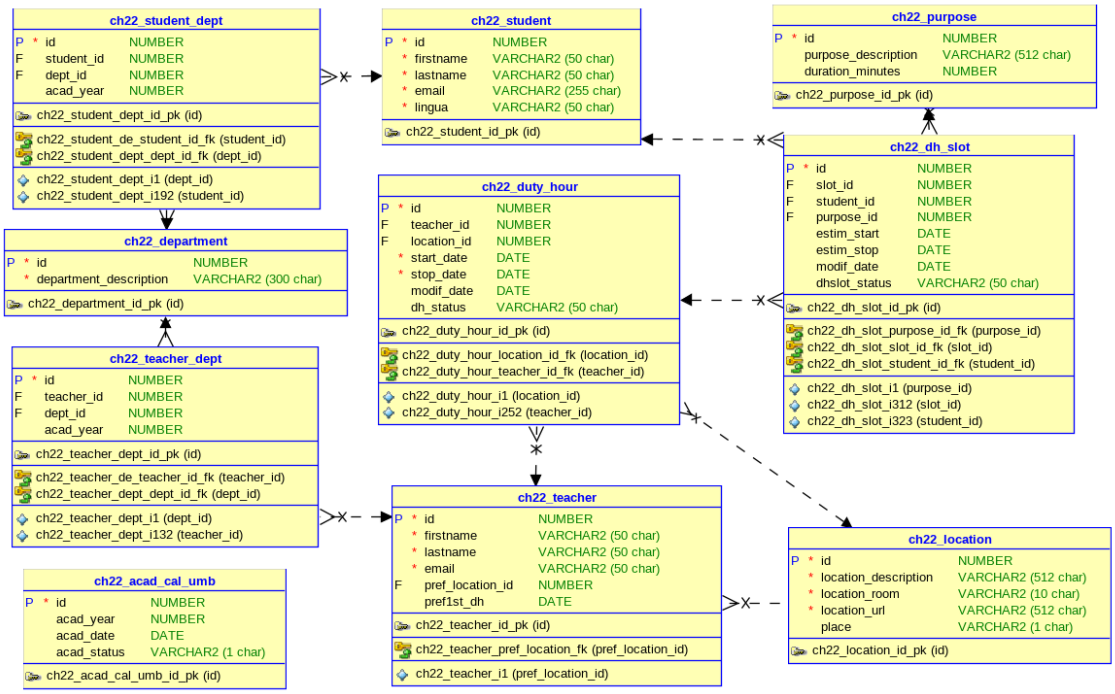
Logički model podataka je prikazan na slici 22.2.



Slika 22.2: Logički model podataka

22.4.3 Relacijski model podataka

Automatsku transformaciju iz logičkog modela podataka u relacijski u Oracle SQL Data Modeleru omogućava funkcija *Engineering to relational*. Rezultat, relacijski model podataka spreman za izvoz kao SQL skripta, prikazan je na slici 22.3.



Slika 22.3: Relacijski model podataka

Relacijski model podataka može se razviti bez SQL Data Developer Data Modelera. Možemo koristiti Quick SQL u APEX-u.

22.4.4 Quick SQL

Prikazani Quick SQL (pogledajte sliku 22.4) sadrži sve činjenice potrebne za generiranje SQL skripte.

```
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
```

Slika 22.4: Model podataka opisan pomoću Quick SQL-a

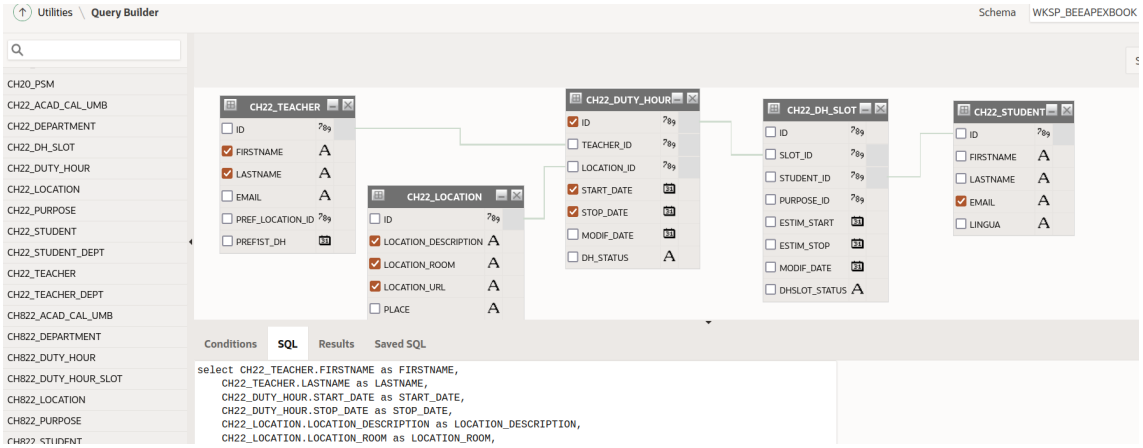
Relacijski model podataka može se razviti bez SQL Data Developer Data Modeler. Možemo koristiti Quick SQL u APEX-u.

22.4.5 SQL skripta za kreiranje tablica

SQL skripta za kreiranje tablica navedena je u materijalima za učenje u datoteci CH22CREATE.sql. Skripta za izradu tablica također je dostupna u pakiranoj aplikaciji u materijalima za učenje.

22.4.6 Query builder u APEX-u

Aplikacija koju razvijamo koristit će prilično duge upite koji bi mogli uplašiti početnike u korištenju APEX-a. Ali postoji lijek, graditelj upita - Query Builder, koji je uključen u APEX (SQL Workshop > Utilities > Query Builder). Ovaj alat može pomoći svakome da konstruira ispravne upite samo klikom i povlačenjem. Figure 22.5 prikazuje primjer upita kojim se dobije kalendar upisanih studenata.



Slika 22.5: Primjer korištenja Query Builder-a.

Generirana SQL naredba ima sljedeći oblik:

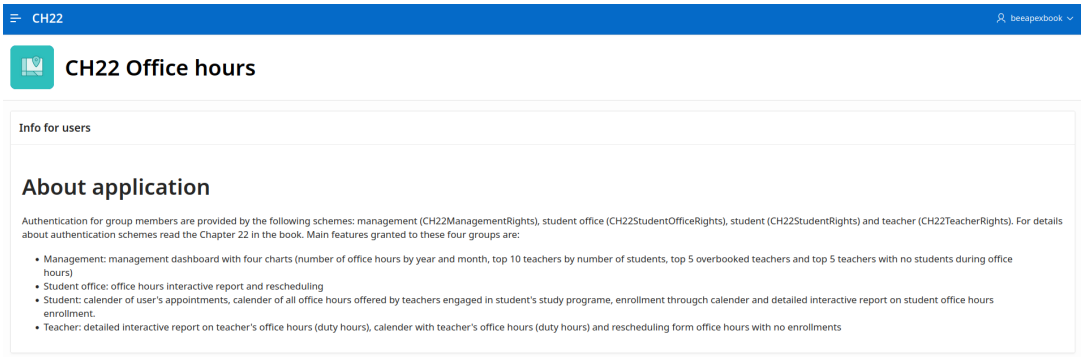
```
select CH22_TEACHER.FIRSTNAME as FIRSTNAME,
       CH22_TEACHER.LASTNAME as LASTNAME,
       CH22_DUTY_HOUR.START_DATE as START_DATE,
       CH22_DUTY_HOUR.STOP_DATE as STOP_DATE,
       CH22_LOCATION.LOCATION_DESCRIPTION as
              LOCATION_DESCRIPTION,
       CH22_LOCATION.LOCATION_ROOM as 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
```

Dodavanjem dodatnog uvjeta na kraju "and CH22_TEACHER.EMAIL = :APP_USER" upit će vratiti upisanog studenta za određenog nastavnika. Query Builder je alat neophodan potpunim početnicima, ali i iskusni korisnici mogu imati koristi od njegove upotrebe.

Upiti koji se koriste u ovoj aplikaciji koriste operator ulančavanja, odnosno konkatenacije (||), funkciju EXTRACT, uobičajene tablične izraze, dohvaćanje samo prvih 10 ili 5 redaka, razvoj koji korisnik prilagođava CH22_dh_utilization. Čitatelji ove knjige mogu imati koristi ako prvo instaliraju zapakiranu aplikaciju i istraže upite koji se koriste na kalendarima, upravljačkoj ploči i izvješćima.

22.5 Aplikacijska sučelja

Aplikacija ima jednu zajedničku stranicu - početnu stranicu. Ova stranica pruža neke osnovne informacije o aplikaciji (vidi Figure 22.6).



Slika 22.6: Početna stranica aplikacije

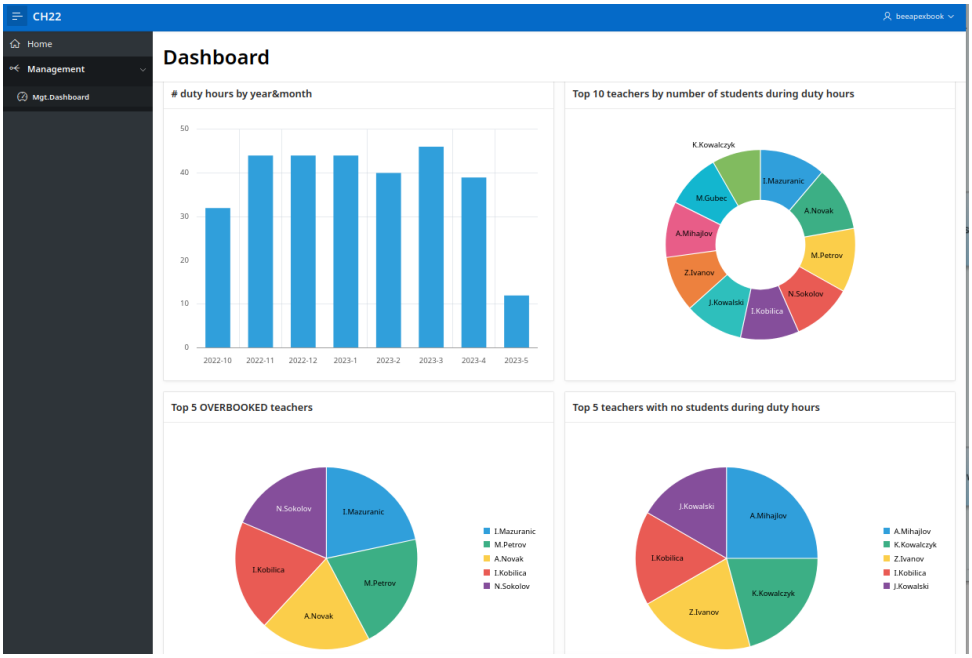
Menadžment, osoblje studentskih referada, studenti i nastavnici koristit će se isključivo prilagođenim sučeljima.

22.5.1 Sučelja aplikacija za menadžment

Korisnici s menadžerskim pravima mogu pristupiti upravljačkoj ploči za menadžment (pogledajte Figure 22.7). Ona sadrži grafikone za potporu donošenja odluka vezanih uz konzultacije i nastavnike:

- broj sati konzultacija u godini i mjesecu,
- 10 nastavnika s najviše studenata,
- 5 nastavnika s najviše preopterećenja,
- 5 nastavnika bez zahtjeva studenata za konzultacijama.

U ovom trenutku nisu definirani drugi zahtjevi menadžmenta.

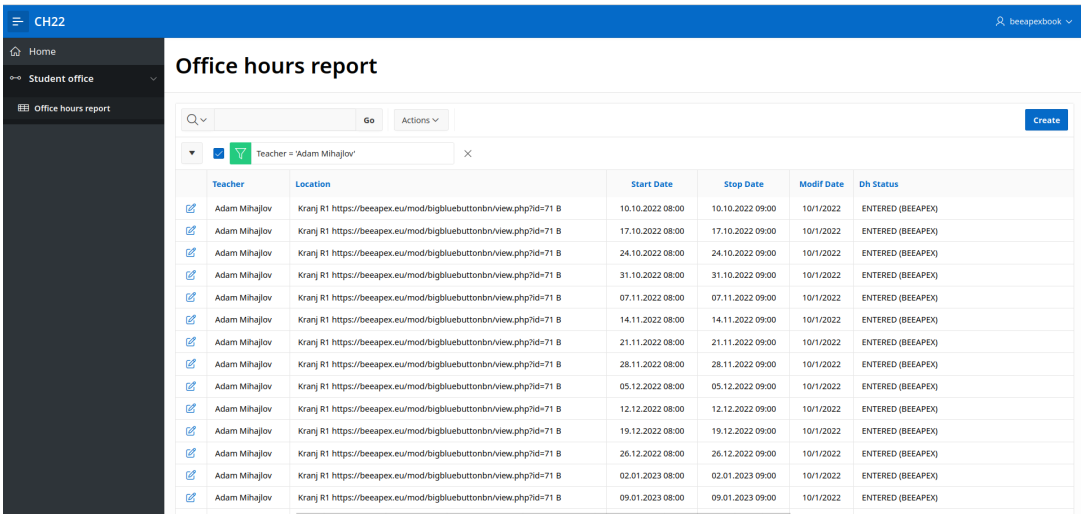


Slika 22.7: Upravljačka ploča za menadžment

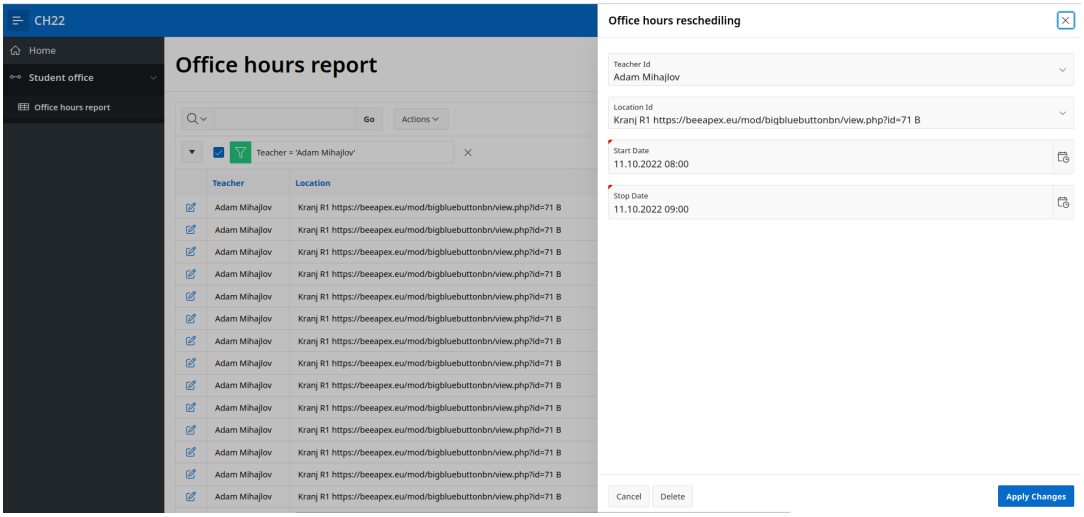
22.5.2 Aplikacijska sučelja studentske referade

Korisnici s pravima studentske referade mogu pristupiti interaktivnom izvješću o konzultacijama i promijeniti raspored bilo kojeg termina konzultacija. Figure 22.8 prikazuje interaktivno izvješće

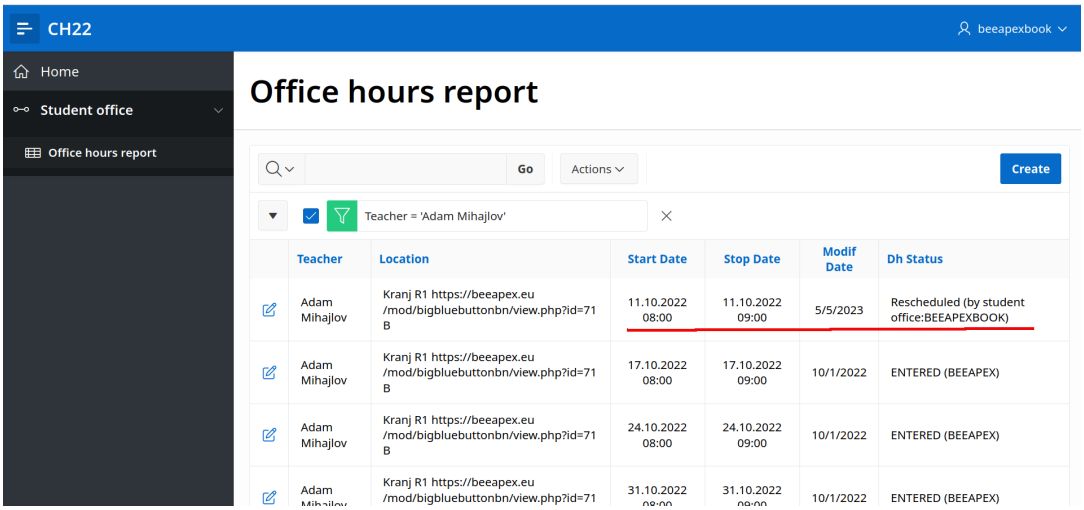
s aktiviranim filtrom (prikazuje se samo jedan učitelj). Nadalje, Figure 22.9 prikazuje obrazac koji omogućuje promjenu rasporeda konzultacija. Korisnik može promijeniti nastavnika, lokaciju, datum početka i završetka odabranih termina konzultacija. Figure 22.10 prikazuje izvješće nakon promjene rasporeda termina konzultacija. Automatski se ažuriraju dva polja: datum izmjene i status termina konzultacija. Potonji je konkatenacija polja koja sadrži i korisničko ime koje je izvršilo promjenu.



Slika 22.8: Interaktivno izvješće o konzultacijama za studentsku referadu



Slika 22.9: Obrazac za promjenu rasporeda namijenjen studentskoj referadi

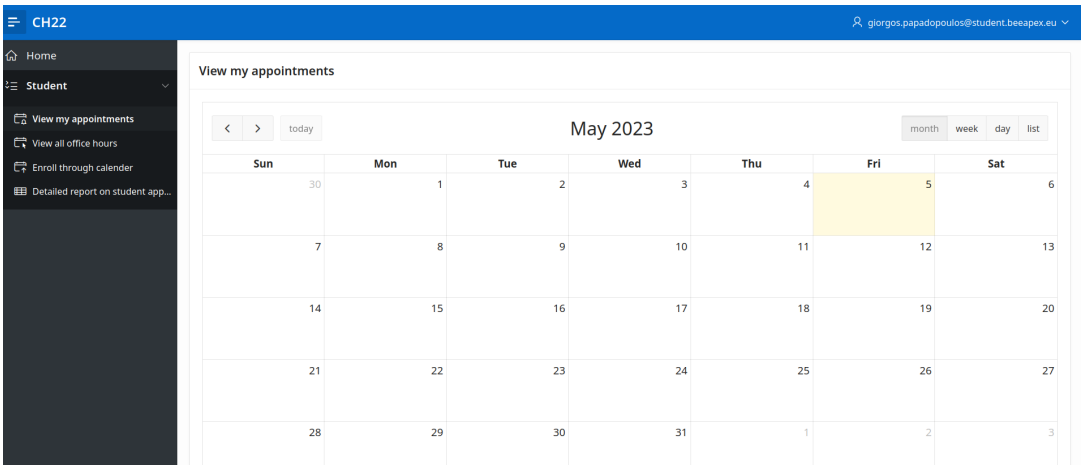


Slika 22.10: Interaktivno izvješće o konzultacijama nakon promjene rasporeda

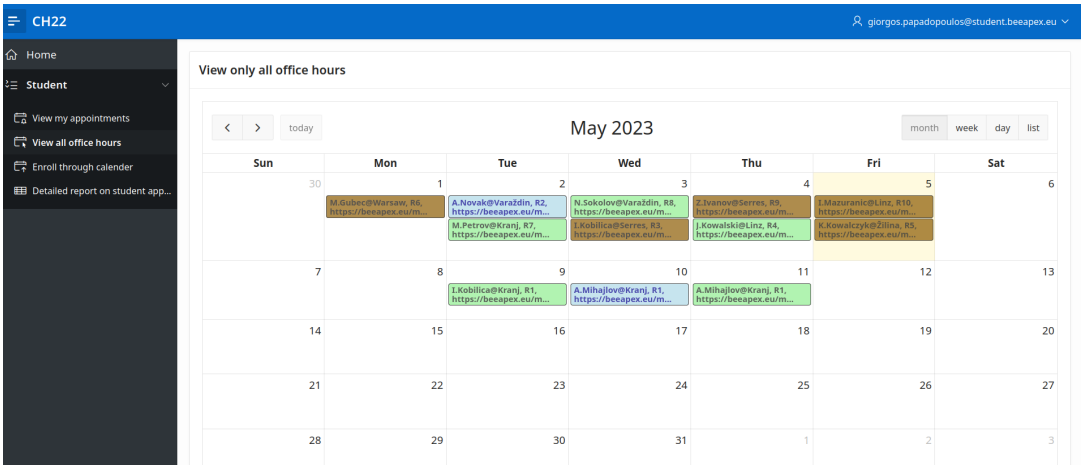
22.5.3 Aplikacijska sučelja namijenjena studentima

stavke kalendara su obojene ovisno o vrijednostima atributa. "Niti jedan upisan" je prikazan zelenom bojom, "raspoloživ" je prikazan plavom bojom, a "preopterećen" je predstavljen smeđom bojom. U ovom trenutku student se može upisati čak i ako se utvrdi status "preopterećen". Korisnici sa studentskim pravima mogu pristupiti:

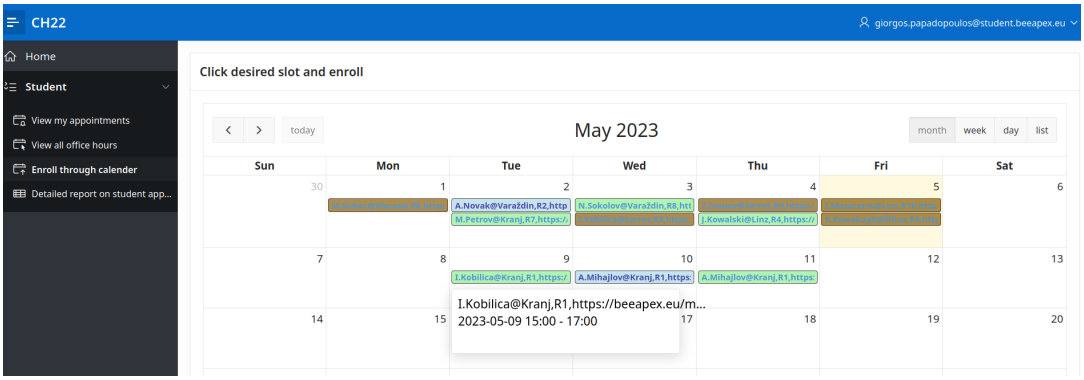
- kalendar njegovih/njezinih sastanaka (vidi Figure 22.11),
- kalendar s tasporedom svih termina konzultacija koje nude nastavnici uključeni u studijski program (vidi Figure 22.12),
- upisi termina konzultacija putem kalendara. Prvo, korisnici odabiru željeni termin konzultacija (pogledajte Figure 22.13). Na obrascu student odabire samo svrhu konzultacija (vidi Figure 22.14) i klikne gumb "Upis". Ako je upis uspješan, studenti dobivaju povratne informacije (pogledajte Figure 22.15). Provjerom upisa kroz stavku izbornika "Prikaži moje termine" potvrđuje se uspješan upis (vidi Figure 22.16),
- Detaljno interaktivno izvješće prikazuje sve studentske termine u tabelarnom obliku (vidi Figure 22.17).



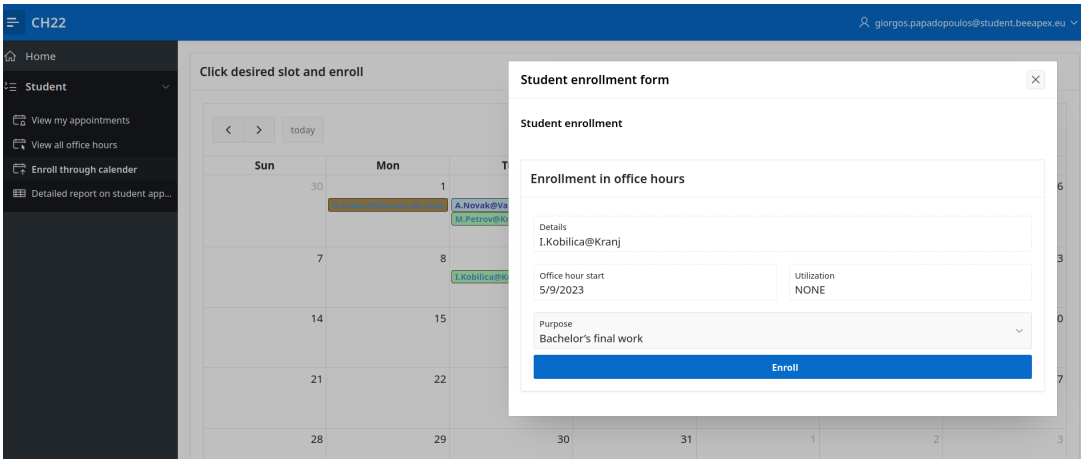
Slika 22.11: kalendar studentskih termina



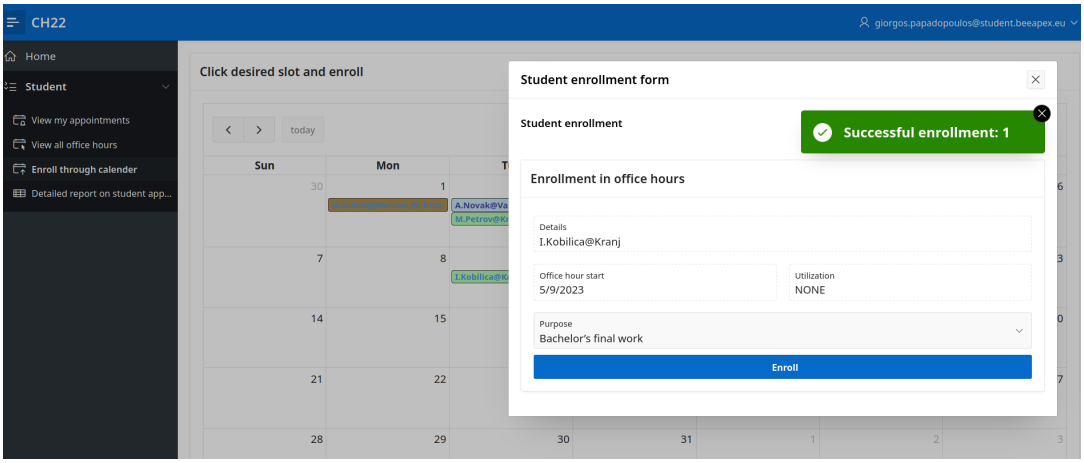
Slika 22.12: Kalendar svih uredovnih sati nastavnika namijenjenih konzultacijama, na studijskom programu studenta



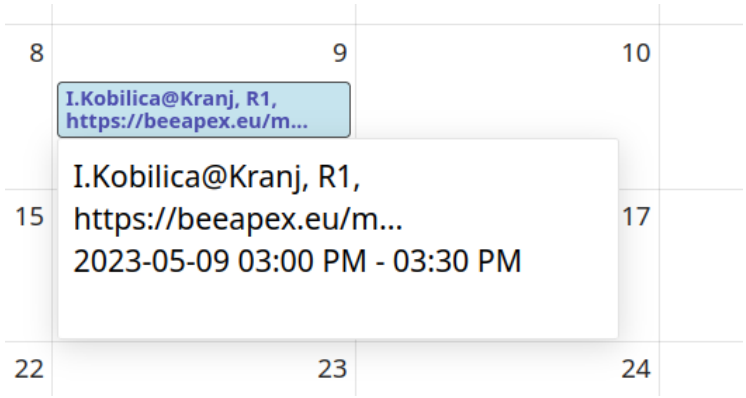
Slika 22.13: Upis termina konzultacija putem kalendara - odabir vremensko odsječka u kalendaru



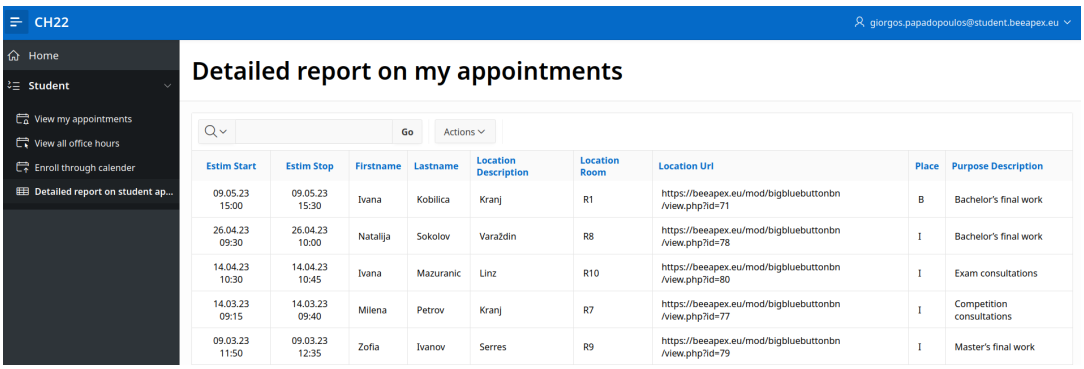
Slika 22.14: Upis termina konzultacija putem kalendara - odabir svrhe



Slika 22.15: Upis termina konzultacija putem kalendara - uspješan upis



Slika 22.16: Provjera upisa kroz točku izbornika "Pregledaj moje termine"



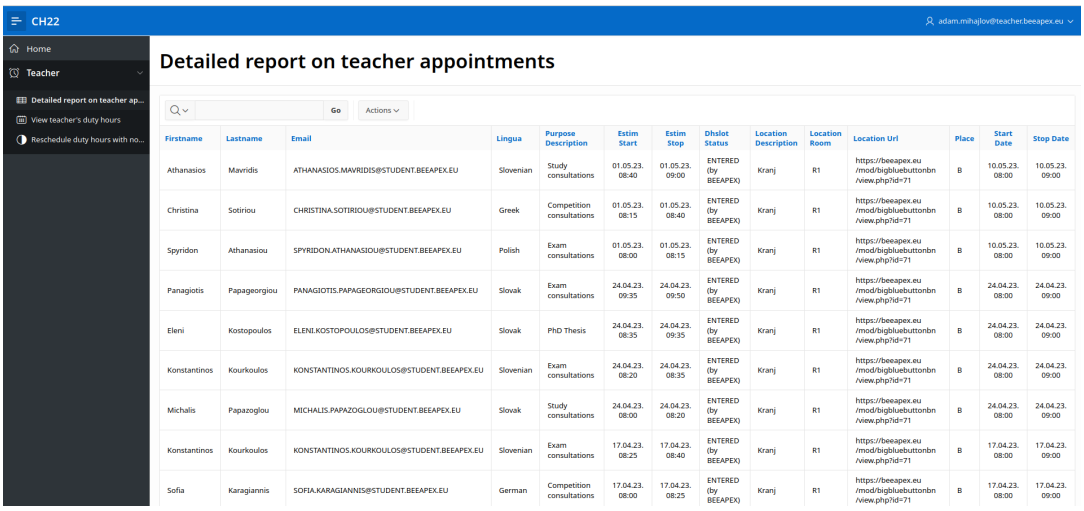
Slika 22.17: Detaljno interaktivno izvješće o svim studentskim terminima

22.5.4 Aplikacijska sučelja namijenjena nastavnicima

stavke kalendara su obojene ovisno o korištenju. Boje statusa su identične onima u studentskom sučelju. "Niti jedan upisan" je prikazan zelenom bojom, "raspoloživ" je prikazan plavom bojom, a "preopterećen" je predstavljen smeđom bojom. Korisnici s pravima nastavnika mogu:

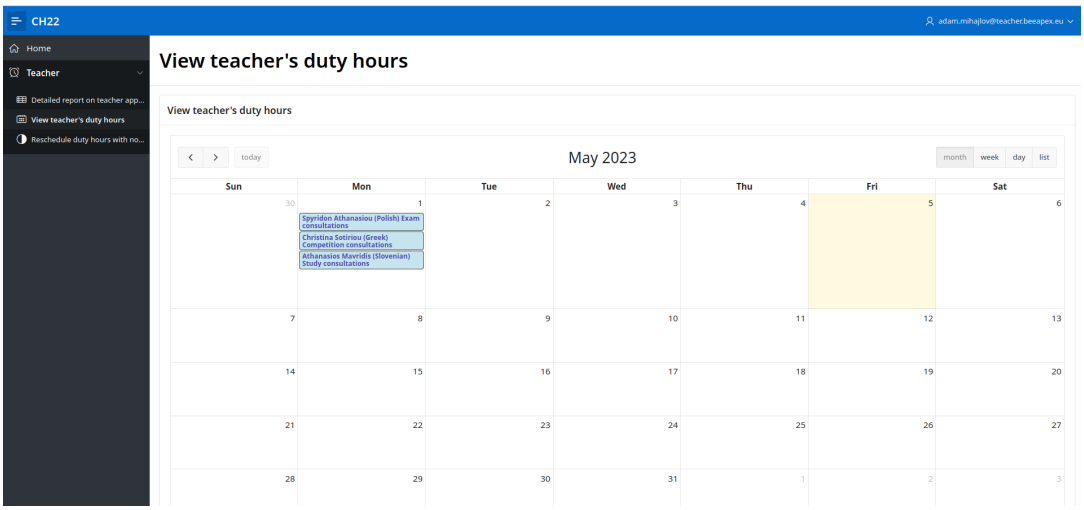
- pogled detaljnog izvješća o zakazanim terminima (pogledajte sliku 22.18),
- pregled kalendara nastavnika. Ime studenta, jezik i svrha prikazani su za svakog upisanog studenta (pogledajte sliku 22.19),

- promjena rasporeda konzultacija bez upisanih studenata. Prvo se prikazuje kalendar konzultacija sa statusom "Niti jedan upisan" (pogledajte sliku 22.20). Klikom na stavku na kalendaru otvara se modalni obrazac u koji se unose datumi početka i završetka (također sati i minute) (pogledajte sliku 22.21). Klikom na "Primijeni promjene" termin konzultacija se pomiče na željeni novi datum (vidi sliku 22.22).

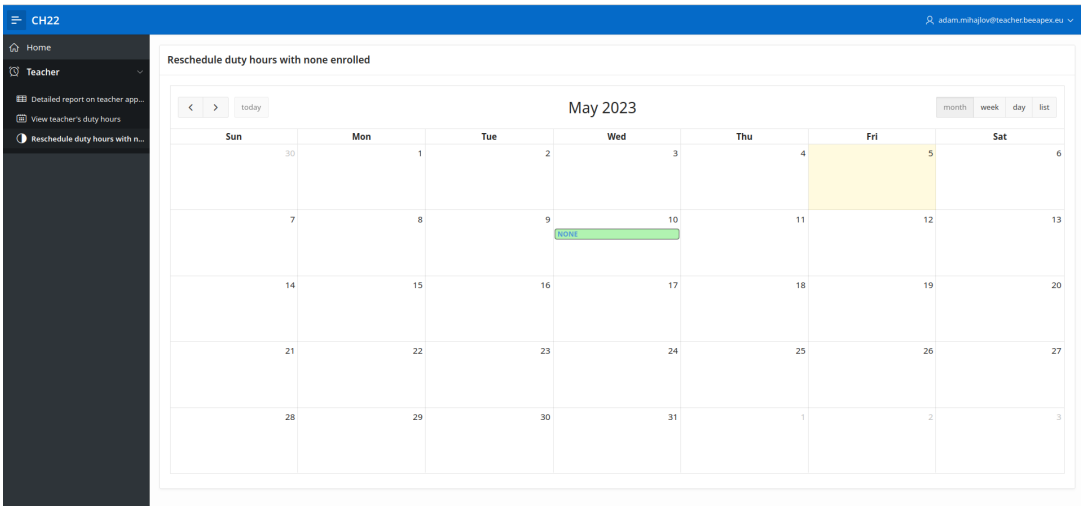


Firstname	Lastname	Email	Lingua	Purpose Description	Estim Start	Estim Stop	Dhslot Status	Location Description	Location Room	Location Url	Place	Start Date	Stop Date
Athanasios	Mavridis	ATHANASIOS.MAVRIDIS@STUDENT.BEEAPEX.EU	Slovenian	Study consultations	01.05.23. 08:40	01.05.23. 09:00	ENTERED (by BEEAPEX)	Kranj	R1	https://beeapex.eu /mod/bigbluebuttonbn /view.php?id=71	B	10.05.23. 08:00	10.05.23. 09:00
Christina	Sotiriou	CHRISTINA.SOTIRIOU@STUDENT.BEEAPEX.EU	Greek	Competition consultations	01.05.23. 08:15	01.05.23. 08:40	ENTERED (by BEEAPEX)	Kranj	R1	https://beeapex.eu /mod/bigbluebuttonbn /view.php?id=71	B	10.05.23. 08:00	10.05.23. 09:00
Spyridon	Athanasiou	SPYRIDON.ATHANASIOU@STUDENT.BEEAPEX.EU	Polish	Exam consultations	01.05.23. 08:00	01.05.23. 08:15	ENTERED (by BEEAPEX)	Kranj	R1	https://beeapex.eu /mod/bigbluebuttonbn /view.php?id=71	B	10.05.23. 08:00	10.05.23. 09:00
Panagiotis	Papageorgiou	PANAGIOTIS.PAPAGEORGIOU@STUDENT.BEEAPEX.EU	Slovak	Exam consultations	24.04.23. 09:35	24.04.23. 09:50	ENTERED (by BEEAPEX)	Kranj	R1	https://beeapex.eu /mod/bigbluebuttonbn /view.php?id=71	B	24.04.23. 08:00	24.04.23. 09:00
Eleni	Kostopoulos	ELENI.KOSTOPOULOS@STUDENT.BEEAPEX.EU	Slovak	PhD Thesis	24.04.23. 08:35	24.04.23. 09:35	ENTERED (by BEEAPEX)	Kranj	R1	https://beeapex.eu /mod/bigbluebuttonbn /view.php?id=71	B	24.04.23. 08:00	24.04.23. 09:00
Konstantinos	Kourkoulos	KONSTANTINOS.KOURKOULOS@STUDENT.BEEAPEX.EU	Slovenian	Exam consultations	24.04.23. 08:20	24.04.23. 08:35	ENTERED (by BEEAPEX)	Kranj	R1	https://beeapex.eu /mod/bigbluebuttonbn /view.php?id=71	B	24.04.23. 08:00	24.04.23. 09:00
Michalis	Papazoglou	MICHALIS.PAPAZOGLOU@STUDENT.BEEAPEX.EU	Slovak	Study consultations	24.04.23. 08:00	24.04.23. 08:20	ENTERED (by BEEAPEX)	Kranj	R1	https://beeapex.eu /mod/bigbluebuttonbn /view.php?id=71	B	24.04.23. 08:00	24.04.23. 09:00
Konstantinos	Kourkoulos	KONSTANTINOS.KOURKOULOS@STUDENT.BEEAPEX.EU	Slovenian	Exam consultations	17.04.23. 08:25	17.04.23. 08:40	ENTERED (by BEEAPEX)	Kranj	R1	https://beeapex.eu /mod/bigbluebuttonbn /view.php?id=71	B	17.04.23. 08:00	17.04.23. 09:00
Sofia	Karagiannis	SOFIA.KARAGIANNIS@STUDENT.BEEAPEX.EU	German	Competition consultations	17.04.23. 08:00	17.04.23. 08:25	ENTERED (by BEEAPEX)	Kranj	R1	https://beeapex.eu /mod/bigbluebuttonbn /view.php?id=71	B	17.04.23. 08:00	17.04.23. 09:00

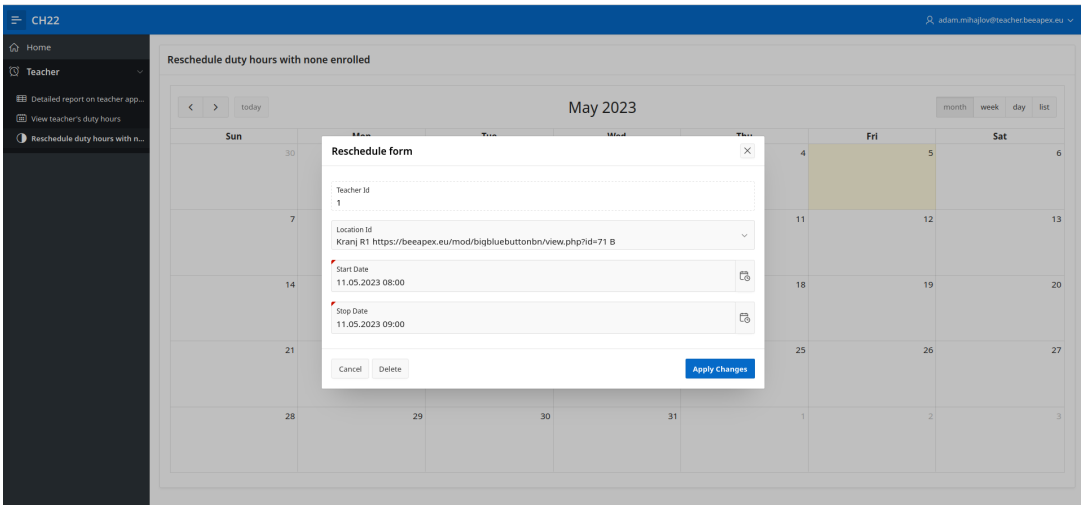
Slika 22.18: Detaljno izvješće o zakazanim terminima nastavnika



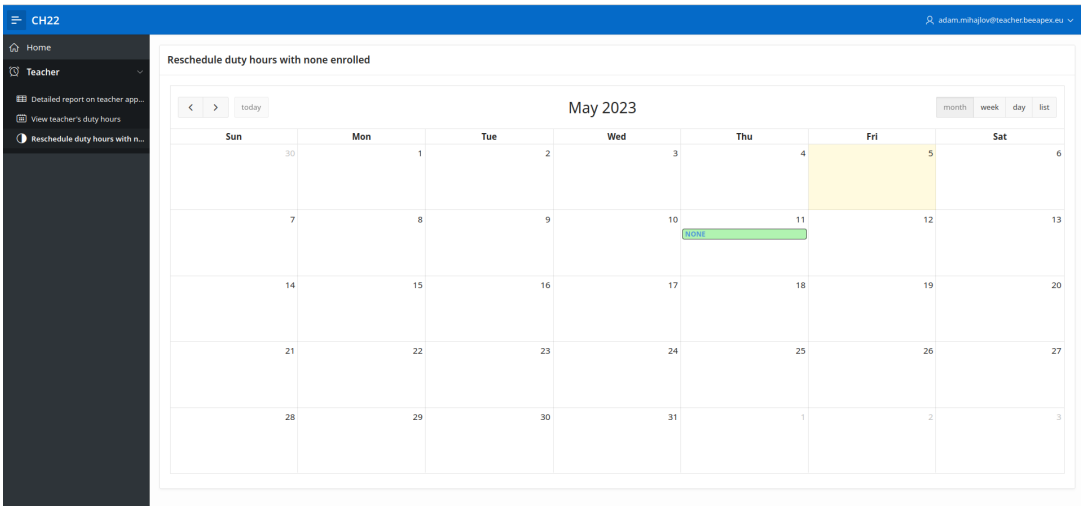
Slika 22.19: Kalendar nastavnika s vidljivim imenima studenata, jezikom kojim govore i svrhom konzultacija



Slika 22.20: Promjena rasporeda termina konzultacija bez upisanih studenata - prikaz kalendara



Slika 22.21: Promjena rasporeda termina konzultacija bez upisanih studenata - unesen je novi datum



Slika 22.22: Rezultat promjene rasporeda termina konzultacija bez upisanih studenata

22.6 Dodatni materijali za učenje

Na raspolaganju vam je sljedeći dodatni materijal za učenje:

- izvezena aplikacija, skripte za kreiranje tablica, umetanje podataka, kreiranje PL/SQL funkcije i ispuštanje tablica,
- video vodiči: a) instaliranje pakirane aplikacije i kreiranje korisnika (studenta i nastavnika) i b) razvoj aplikacije od nule.

Svi dodatni materijali za učenje dostupni na javna stranica projekta BeeAPEX. Prijavite se kao gost (zaporka nije potrebna). Pronađite udžbenik u odjeljku Books, skripte u mapi Part 2 > Chapter22 u odjeljku Scripts i video vodiče u Collection of video guides. Materijal za kratke tečajeve nalazi se u odjeljku Short courses.

22.6.1 Izvezena aplikacija

Izvezena aplikacija je zapakirana. Instalacija stvara tablice, pojmovnik, funkcije i popunjava podatke. Deinstalacija uklanja sve objekte baze podataka koji se koriste u ovoj aplikaciji. Zapakirana aplikacija sadrži uključenu CSS datoteku (ch22_cal.css) koja se referencira iz većine stranica kalendara kao #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;
}
```

Upravljačka nadzorna ploča koristi sljedeće upite:

```
SELECT
        EXTRACT(YEAR FROM start_date) || '-' ||
        EXTRACT(MONTH FROM  start_date) AS "Year and Month",
        COUNT(id) AS count
FROM ch22_duty_hour
        GROUP BY EXTRACT(YEAR FROM start_date),
                EXTRACT(MONTH FROM start_date)
order by 1;

```

```
with teacher_appointmets (appointment, teacher) as
(select CH22_DH_SLOT.ID,
substr(CH22_TEACHER.FIRSTNAME,1,1) || '.' ||
        CH22_TEACHER.LASTNAME
from CH22_DH_SLOT CH22_DH_SLOT,
CH22_DUTY_HOUR CH22_DUTY_HOUR,
CH22_TEACHER CH22_TEACHER
where CH22_DUTY_HOUR.ID=CH22_DH_SLOT.SLOT_ID
and CH22_TEACHER.ID=CH22_DUTY_HOUR.TEACHER_ID)
select count(appointment), teacher from teacher_appointmets
        group by teacher
order by 1 desc
fetch  first 10 rows only;

with teacher_dh_statuses (duty_hour_id, teacher,dh_utilization) as
(select CH22_DUTY_HOUR.ID,
substr(CH22_TEACHER.FIRSTNAME,1,1) || '.' ||
        CH22_TEACHER.LASTNAME,
CH22_dh_utilization(CH22_DUTY_HOUR.ID)
from CH22_DUTY_HOUR CH22_DUTY_HOUR,
CH22_TEACHER CH22_TEACHER
where CH22_DUTY_HOUR.TEACHER_ID=CH22_TEACHER.ID and
CH22_dh_utilization(CH22_DUTY_HOUR.ID) = 'OVERBOOKED')
select count(duty_hour_id), teacher from teacher_dh_statuses
group by teacher
order by 1 DESC
fetch  first 5 rows only;

with teacher_dh_statuses (duty_hour_id, teacher,dh_utilization) as
(select CH22_DUTY_HOUR.ID,
substr(CH22_TEACHER.FIRSTNAME,1,1) || '.' || CH22_TEACHER.LASTNAME,
CH22_dh_utilization(CH22_DUTY_HOUR.ID)
from CH22_DUTY_HOUR CH22_DUTY_HOUR,
CH22_TEACHER CH22_TEACHER
where CH22_DUTY_HOUR.TEACHER_ID=CH22_TEACHER.ID and
CH22_dh_utilization(CH22_DUTY_HOUR.ID) = 'NONE')
select count(duty_hour_id), teacher from teacher_dh_statuses
group by teacher
order by 1 DESC
fetch  first 5 rows only;
```

Pogledajte druge upite za stranice koje se temelje na elementu kalendara u pakiranoj aplikaciji.

22.6.2 Video vodiči

Video vodiči pokazuju instalaciju zapakirane aplikacije i razvoj aplikacije od nule.

22.7 Pitanja

1. Kako biste dodali korisnika s pravima nastavnika?
2. Kako biste poboljšali aplikaciju omogućavajući studentu da vodi bilješke za određeni sastanak?

3. Kako biste ograničili rezultate upita na retke koji imaju datume u stupcu datuma jednake ili veće od trenutnog vremena?

22.8 Odgovori

1. Koristite Upravljanje korisnicima i grupom (Manage Users and Group). Dodajte novog korisnika. Koristite SQL Workshop > Object Browser i odaberite tablicu CH22_TEACHER. Odaberite karticu "Data" i "Insert row". Adresa e-pošte umetnutog retka mora odgovarati kreiranom APEX korisniku.
2. Napravite novu tablicu, tj. CH22_NOTES s poljima: ID, bilješka i DH_SLOT_ID kao strani ključ koji upućuje na tablicu CH22_DH_SLOT. Kreirajte novo izvješće i obrazac za studenta i autorizirajte stranicu za studenta. Ovisno o institucionalnim pravilima, nastavnicima možete dodijeliti pravo pregledavanja ili izmjene studentskih bilješki.
3. Za ograničavanje rezultata upita na retke koji imaju datume u stupcu datuma jednake ili veće od trenutnog vremena mora se dodati uvjet WHERE date_column >= SYSDATE.



23. Telco studija slučaja

VERONIKA ŠALGOVÁ, JOZEF KOSTOLNÝ, MICHAL MRENA, MICHAL KVET, MIROSLAV POTOČÁR

23.1 Poslovni pogled na slučaj

Za demonstraciju izrade prototipa aplikacije, kao primjer, odabrat ćemo fiktivnu tvrtku VEYOMI. VEYOMI je mala tvrtka koja se fokusira na prodaju telekomunikacijskih usluga svojim klijentima. Kako VEYOMI ima za cilj proširiti svoj portfelj proizvoda i zadovoljiti rastuću bazu kupaca, postaje im imperativ uspostaviti i održavati robusnu aplikaciju koja olakšava besprijekorno upravljanje različitim aspektima, uključujući inventar proizvoda, informacije o klijentima i automatizirane sustave naplate. U sve konkurentnijem tržišnom okruženju, VEYOMI prepoznaje važnost pružanja brzog i praktičnog pristupa vitalnim informacijama za svoje klijente. Razvijanjem intuitivne aplikacije, VEYOMI može poboljšati svoj odziv, osiguravajući da relevantne informacije o proizvodima, uslugama, cijenama i naplati budu dostupni na dohvat ruke njegovoj pronicljivoj klijenteli.

23.2 Definicija problema

Za poslovanje je ključno uskladiti potrebe klijenata s nuđenim proizvodima i uslugama, ali i omogućiti da klijenti tvrtkinim uslugama lako ostvaruju svoje željene potrebe. To omogućuje da su sve potrebne informacije nadohvat ruke, što olakšava učinkovito pružanje usluga klijentima. Osim toga, ključno je omogućiti korisnicima jednostavan i pristupačan način provjere statusa njihovih usluga. Nudeći platformu ili aplikaciju jednostavnu za korištenje, klijentima je omogućeno da lako saznaju što se događa s njihovim uslugama i budu u tijeku. Održavanjem točne evidencije i pružanjem jednostavnog pristupa ažuriranjima usluge može se povećati zadovoljstvo korisnika, izgraditi povjerenje i poboljšati cjelokupno poslovanje.

23.3 Slučajevi korištenja

Prilikom izrade aplikacije potrebno je krenuti od specifikacije usluga koje bi aplikacija trebala pružati i definiranja područja koja bi ista trebala pokrivati. Stoga je priprema popisa pojedinačnih funkcionalnosti i usluga ključna. Prvo se potrebno usredotočiti na definiranje tipova korisnika. Za našu upotrebu možemo uzeti u obzir tri temeljne uloge – kupca (Customer), rukovoditelj (Manager) i administratora (Administrator). Svaka uloga ima posebne usluge kako bi se osigurala cjelokupna pokrivenost funkcionalnosti aplikacije.

Tablica 23.1: Opis slučaja korištenja: dodaj uslugu

Ključna riječ	**Vrijednost**
ID:	*Ch23-01*
Naziv:	*Dodaj uslugu*
Opis:	*Korisnik koristi APEX aplikaciju za dodavanje nove usluge među dostupne*
Primari učesnik:	*Kupac*
Preduvjeti:	*Pristup aplikaciji, važeće vjerodajnice, web aplikacija je dostupna.*
Uvjeti po završetku:	*Podaci o novoj usluzi pohranjuju se u bazu podataka.*
Glavni:	*Scenariji*
Uspješan scenarij:	1. *Prijava kupca u aplikaciju.* 2. *Kupac navigira na Buy extra regiju Home stranice.* 3. *Kupac unosi izos.* 4. *S drop-down izbornika kupac odabire novu uslugu.* 5. *Odabirom "Buy" gumba kupac potvrđuje aktivaciju nove usluge.*
Ekstenzije:	-
Frekvencija korištenja	*Kupac mijenja skup usluga otprilike jednom u šest mjeseci.*
Status:	*Završeno*
Vlasnik:	*Kupac*
Prioritet:	*visok*

23.3.1 Narativni opis slučaja korištenja

Radi sažetosti prikazat ćemo samo dva slučaja korištenja, od nekoliko mogućih:

- Dodaj uslugu (Engl. Add Service) - korisnici mogu jednostavno zatražiti dodavanje novih usluga svojoj postojećoj pretplati. Sustav aktivira novu uslugu i u skladu s tim ažurira naplatu korisnika. Kupac dobiva potvrdu i informaciju o dodanoj usluzi.
- Prikaži stanje usluga (engl. Show State of Services) - Moguće je jednostavno vidjeti trenutni status i pojedinosti korisnikovih pretplaćenih usluga. Sustav dohvaća i prikazuje pregled pretplaćenih usluga korisnika. To uključuje informacije poput vrste usluge, statusa aktivacije, preostalog korištenja ili razdoblja valjanosti i svih povezanih značajki ili ograničenja. Unutar pregleda usluge korisnici mogu pristupiti detaljnijim informacijama o korištenju svojih usluga. To može uključivati potrošnju podataka, minute poziva, poslane tekstualne poruke ili bilo koju drugu relevantnu metriku upotrebe.

23.3.2 Polustrukturirani opis

Dva slučaja korištenja su prikazana u tablicama 23.1 i 23.2.

23.3.3 Dijagram slučaja korištenja

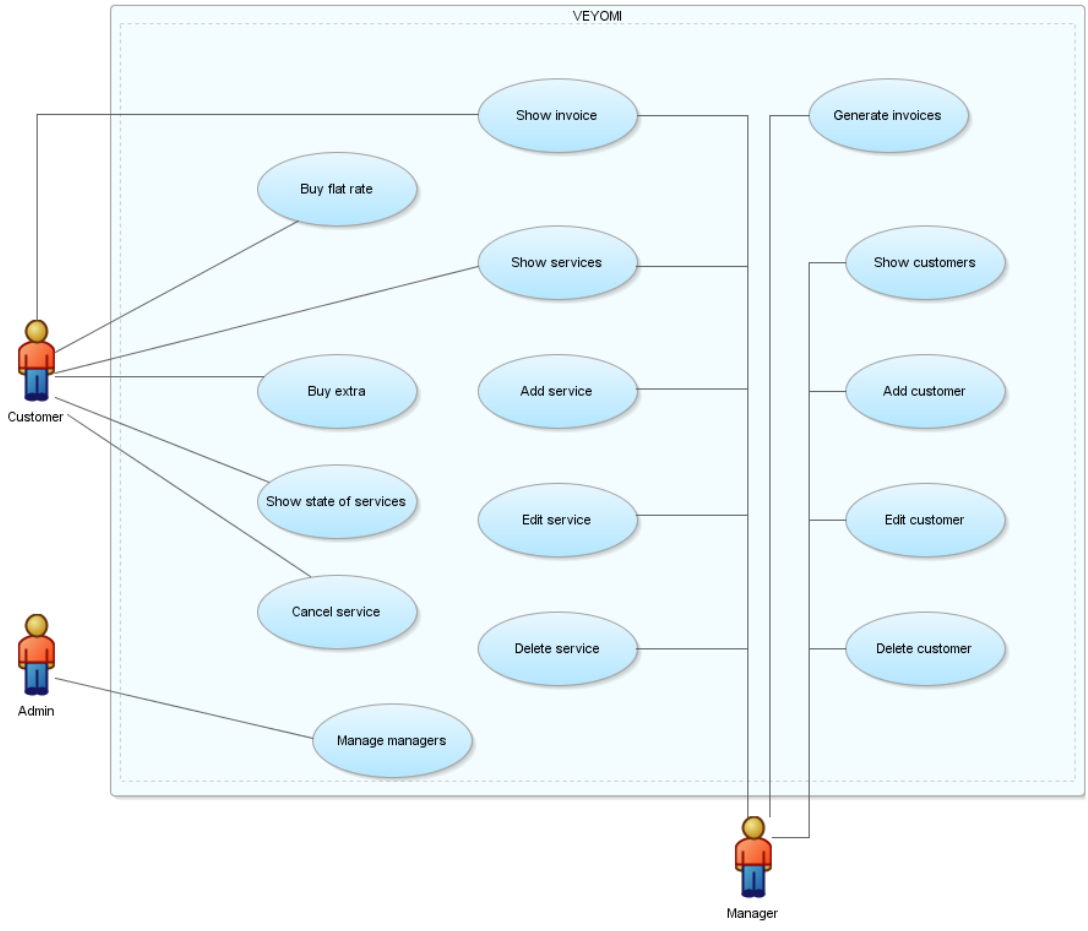
Kupcu treba osigurati kupnju paušalnih i dodatnih usluga, s prikazom statusa usluga koje trebaju imati mogućnost otkazivanja. Druga bitna funkcionalnost koju treba pokriti je prikaz pregleda faktura.

Korisnička uloga rukovoditelja može pristupiti pregledu kupaca te njihovom dodavanju, uređivanju i brisanju. Osim toga, rukovoditelj može generirati i fakture.

Posljednja korisnička uloga je administrator, koji ima jednu osnovnu funkciju: upravljanje rukovoditeljima. Prikaz ovih uloga ilustriran je na slici 23.1.

Tablica 23.2: Opis slučaja korištenja: prikaži status usluge.

Klučna riječ	**Vrijednost**
ID:	*Ch23-02*
Naziv:	*Prikaži korištenje usluge*
Opis:	*Kupac koristi APEX aplikaciju da bi dobio uvid u korištenje usluge*
Primarni učesnik:	*Kupac*
Preduvjeti:	*Pristup aplikaciji, važeće vjerodajnice, web aplikacija je dostupna.*
Uvjeti nakon izvršenja:	-
Glavni:	*Scenariji*
Uspješan scenarij:	1. *Prijava kupca u aplikaciju.* 2. *Kupac odabire "Stats" tab.* 3. *S drop-down izbornika kupac odabire uslugu. Prikazuje se stupčasti graf usluge.*
Ekstenzije:	-
Učestalost korištenja:	*Kupac zahtijeva uvid otprilike jednom mjesečno*
Status:	*Završeno*
Vlsnik:	*Kupac*
Prioritet:	*visok*



Slika 23.1: Dijagram slučaja korištenja

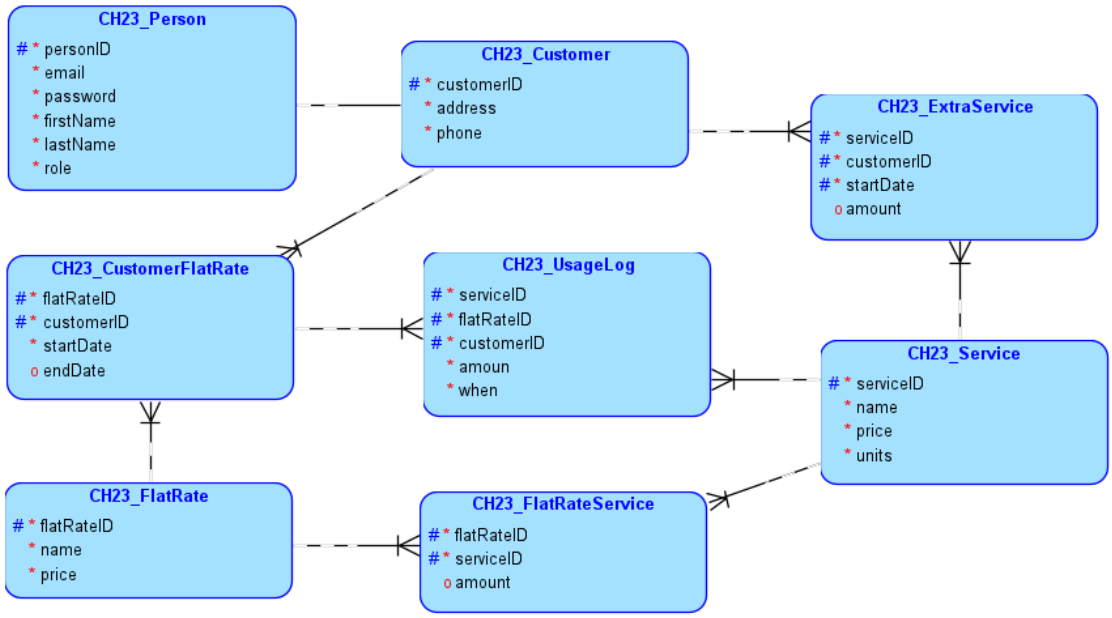
23.4 Podatkovni model

23.4.1 Narativni opis podatkovnog modela

Tablica **CH23_Person** pohranjuje primarne osobne podatke i vrstu uloge korisnika. Kada je osoba kupac, ima kreiran zapis u tablici **CH23_Customer**, gdje su osim ključeva pohranjeni i njegova adresa i broj telefona. Tablica **CH23_FlatRate** pohranjuje podatke o paušalu, odnosno naziv i cijenu. Podaci o pojedinim uslugama, njihov naziv, cijena i ponuđene jedinice pohranjeni su u **CH23_Service** tablici. Pojedinačne usluge dodijeljene paušalima pohranjuju se u tablicu **CH23_FlatRateService**, gdje osim ključeva znamo i dodijeljenu količinu. Ako kupac ugovori paušalnu cijenu, informacije o tome pohranjuju se u tablici **CH23_CustomerFlatRate**, uključujući datum početka i završetka. Potrošene jedinice iz paušala evidentiraju se u tablici **CH23_UsageLog**, gdje je poznato koji je korisnik koristio koju uslugu unutar kojeg paušala, uz datum i količinu. Uz paušal, korisnik može kupiti dodatne usluge evidentirane u tablici **CH23_ExtraService** i njihov iznos.

23.4.2 Logički model podataka

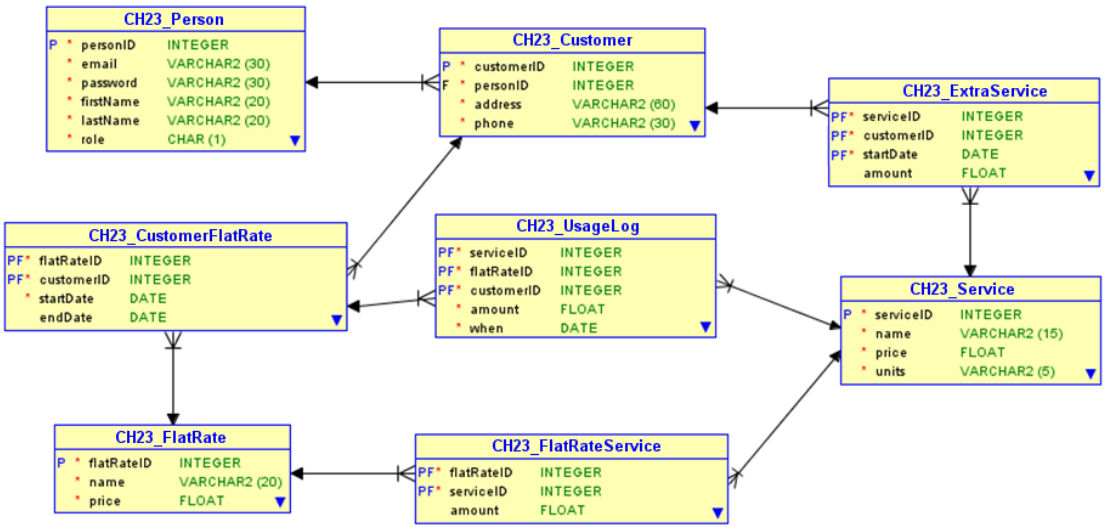
Logički model podataka je prikaza na slici 23.2.



Slika 23.2: Logički model podataka

23.4.3 Relacijski model podataka

Relacijski model podataka je prikazan na Slici 23.3.



Slika 23.3: Relacijski model podataka

23.5 Autentifikacija korisnika i korisničke uloge

U gornjem opisu identificirali smo tri korisničke uloge. Nakon prijave, aplikacija korisniku stavlja na raspolaganje sadržaj prema njegovoj ulozi. Međutim, sam proces prijave je isti za sve korisnike. Apex nam nudi nekoliko načina za upravljanje korisnicima i njihovu autentifikaciju. U sljedećem popisu predstavljamo nekoliko odabranih metoda:

- Račun baze podataka,
- LDAP server,
- Oracle Apex račun,
- prijava putem društvenih mreža,
- prilagođen.

U alatu Oracle Apex odabiremo metodu provjere autentičnosti - **Authentication Scheme** - u **Shared Components \ Authentication Schemes** podstranici. Na ovoj podstranici nalazimo popis postojećih shema za prijavu, a također ovdje možemo kreirati novu shemu za prijavu.Prilikom izrade nove sheme možemo odabrati jednu od gore navedenih metoda provjere autentičnosti.Za potrebe naše aplikacije odabrali smo najopćenitiju metodu, **Custom**.

Glavni dio **Custom** provjere autentičnosti je takozvana funkcija provjere autentičnosti. Ulaz u ovu funkciju je korisničko ime i lozinka koje korisnik unosi na stranici za prijavu, a izlaz je **boolean** vrijednost. Kao što se i očekivalo, funkcija vraća **true** ako dano ime i lozinka odgovaraju postojećem korisniku i **false** ako je situacija suprotna. Na slici 23.1, možemo vidjeti zaglavlje takve funkcije. Stoga, kada stvaramo **Custom** shemu prijave, moramo dati naziv takve funkcije. Odnosno, takva funkcija mora već postojati.

```
CREATE OR REPLACE FUNCTION authenticate_user
(p_username IN varchar2,
p_password IN varchar2)
RETURN boolean;
```

Listing 23.1: Header of an authentication function.

Mi implementiramo autentifikacijsku funkciju koristeći PL/SQL jezik, tj. na nama je da odlučimo kako ćemo autentificirati korisnika. To je razlog zašto je metoda **Custom** najopćenitija. Možemo, na primjer, kontaktirati vanjsku uslugu provjere autentičnosti koristeći REST API, tražiti korisnika u lokalnoj bazi podataka ili prepustiti provjeru autentičnosti drugom lokalnom procesu.

U našoj aplikaciji odabrali smo uobičajeni postupak pretraživanja i provjere korisnika u lokalnoj bazi podataka.

Možemo vidjeti tablicu CH23_Person u modelu podataka koji smo predstavili gore. Stupci **email** i **password** relevantni su za provjeru autentičnosti. Korisnički **email** također služi kao jedinstveno ime za prijavu. Stupac **password** sadrži korisničke lozinke pohranjene u sigurnom raspršenom obliku. Opis sigurne pohrane lozinki je izvan opsega ovog poglavlja. O navedenom čitatelj može više saznati u literaturi. Unatoč tome, prije unosa podataka ili uređivanja kolone **password** u tablici **CH23_Person**, potrebno je raspršiti lozinku. U tu smo svrhu izradili pomoćnu funkciju **hash_password**, čije zaglavlje možemo vidjeti na 23.2. Ova funkcija uzima korisničko ime i lozinku u obliku običnog teksta i generira lozinku u zaštićenom, raspršenom formatu koji se može pohraniti u bazu podataka.

```
CREATE OR REPLACE FUNCTION hash_password
(p_username IN varchar2,
p_password IN varchar2)
RETURN varchar2;
```

Listing 23.2: The header of the auxiliary function hash_password.

Riješili smo automatsko raspršivanje lozinki pri radu s tablicom **CH23_Person** stvaranjem okidača za tablicu osoba CH23_Person table. Okidači raspršuju otvorenu tekstualnu lozinku pomoću gore opisane funkcije **hash_password** prije umetanja ili ažuriranja stupca **password**. Osigurat će da su sve lozinke u tablici **CH23_Person** sigurno pohranjene – čak ni administrator baze podataka nema pristup stvarnoj lozinci. U tablici 23.3, možemo vidjeti uzorak podataka pohranjenih u tablici **CH23_Person**. Zanimljivo je da dva korisnika navedena u tablici imaju istu lozinku. Međutim, ovu činjenicu ne možemo utvrditi na temelju sigurno pohranjenih zaporki.

PERSONID	EMAIL	PASSWORD	ROLE
5	DUIS.A@OUTLOOK.CA	$2a$12$EfSjb2zSiKjqWCoZ47mQeOY6IPoHP/6LvECZxGEWjFZIZcXTF3UgGasd	'a'
11	LOBORTIS@OUTLOOK.EDU	$2a$12$IUqALTj3reLAoTdtC5SNde2.C5A.DOm..7pFw9UBnIsaf7gSNcZyapql	'c'
14	AC.FERMENTUM@AOL.NET	$2a$12$GuaO7Tjp.65d7NYLiEjSzeO6VWQfDz8eXja7dWubcjy77aZDAKRTutyq	'm'

Tablica 23.3: Podaci spremljeni u tablici CH23_Person

Implementacija funkcije autentifikacije sada je jednostavna. U ovoj funkciji prvo tražimo u tablici **CH23_Person** korisničku lozinku s danim korisničkim imenom. Zatim raspršujemo lozinku koju je korisnik unio pomoću funkcije **hash_password**. Ako se oba raspršivanja podudaraju, provjera autentičnosti korisnika je uspjela, a funkcija vraća **true**. S druge strane, ako se raspršivanje ne poklapa ili u tablici ne postoji osoba, odnosno korisnik s tim imenom, funkcija vraća false.

Nedostatak **Custom** metode koju smo odabrali je veća početna problematika izrade baze podataka, unutar koje je potrebno razviti pomoćne funkcije i trigere. S druge strane, prednost je što mi, programeri aplikacija, kontroliramo cijeli proces autentifikacije. Još jedna prednost je to što je korištena implementacija lako skalabilna. Ako, na primjer, u budućnosti želimo koristiti vanjsku uslugu za provjeru autentičnosti, morali bismo promijeniti implementaciju funkcije provjere autentičnosti tako da kontaktira vanjsku uslugu umjesto lokalne baze podataka.

Drugi važan zadatak povezan s upravljanjem korisnicima je upravljanje korisničkim ulogama. Dok je postupak prijave bio isti za sve korisnike, morali smo omogućiti pristup različitim dijelovima

aplikacije na temelju uloge korisnika prilikom upravljanja ulogama. Upravljanje ulogama usko je povezano s upravljanjem korisnicima. Prvo, sve moguće korisničke uloge moraju biti navedene u okruženju Oracle Apex. Upravljanje ulogama možemo pronaći na podstranici **Shared Components \ Application Access Control**. Na ovoj podstranici vidimo popis postojećih uloga, a unutar iste možemo također kreirati i novu ulogu. Za svaku novu ulogu potrebno je navesti njezin statički identifikator koji ćemo kasnije koristiti prilikom dodjele uloge pojedinim korisnicima. U našoj aplikaciji koristimo uloge sa sljedećim statičkim identifikatorima:

- ADMIN,
- CUSTOMER,
- MANAGER.

Ulogu korisniku dodjeljujemo pomoću PL/SQL koda. Na 23.3, možemo vidjeti primjer koda za dodjelu uloge **CUSTOMER** (statički identifikator uloge) korisniku **LOBORTIS@OUTLOOK.EDU**. Koristili smo drugi skup okidača za tablicu osoba (Person) da bismo automatizirali ovaj proces. Stupac **Role** u ovoj tablici može uzeti vrijednosti iz skupa {'a', 'm', 'c'} koje se slažu s početnim slovima gore definiranih uloga. U tijelu okidača, prije umetanja ili ažuriranja podataka, postavljamo odgovarajuću ulogu za određenog korisnika prema vrijednosti stupca uloga (Role) pomoću koda kao na slici 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 Code that associates a given user with a given role.

Posljednji korak implementacije korisničkih uloga u aplikaciji je prikaz različitih podstranica na temelju uloge prijavljenog korisnika. Za svaku podstranicu koja je namijenjena samo određenoj korisničkoj ulozi, potrebno je u **Page Designer** postavkama stranice u **Security \ Authorization Scheme** odabrati korisničku ulogu za koju je određena podstranica namijenjena. Ovaj korak osigurava da korisnici s drugim ulogama ne mogu pristupiti toj podstranici. Primjerice, početna stranica naše aplikacije specifična je po tome što za njezin pregled nije potrebna prijava. Spomenuto definiramo u odjeljku **Security** koristeći stavku **Authentication**, gdje odabiremo opciju **Page Is Public**.

Gore opisane postavke će osigurati da korisnik ima pristup samo podstranicama svoje uloge. Međutim, svi će korisnici i dalje vidjeti sve podstranice u navigacijskom izborniku - čak i one kojima nemaju pristup. Takav prikaz svakako nije poželjan. Kako bi aplikacija korisnicima prikazivala samo podstranice kojima imaju pristup, potrebno je prilagoditi postavke navigacijskog izbornika. To možemo postaviti u podstranici **Shared Components \ Navigation and Search \ Navigation Menu**. Ovdje je u **Authorization Scheme** stupcu za svaku stavku u izborniku potrebno postaviti korisničku ulogu kojoj će se navedena stavka prikazivati.

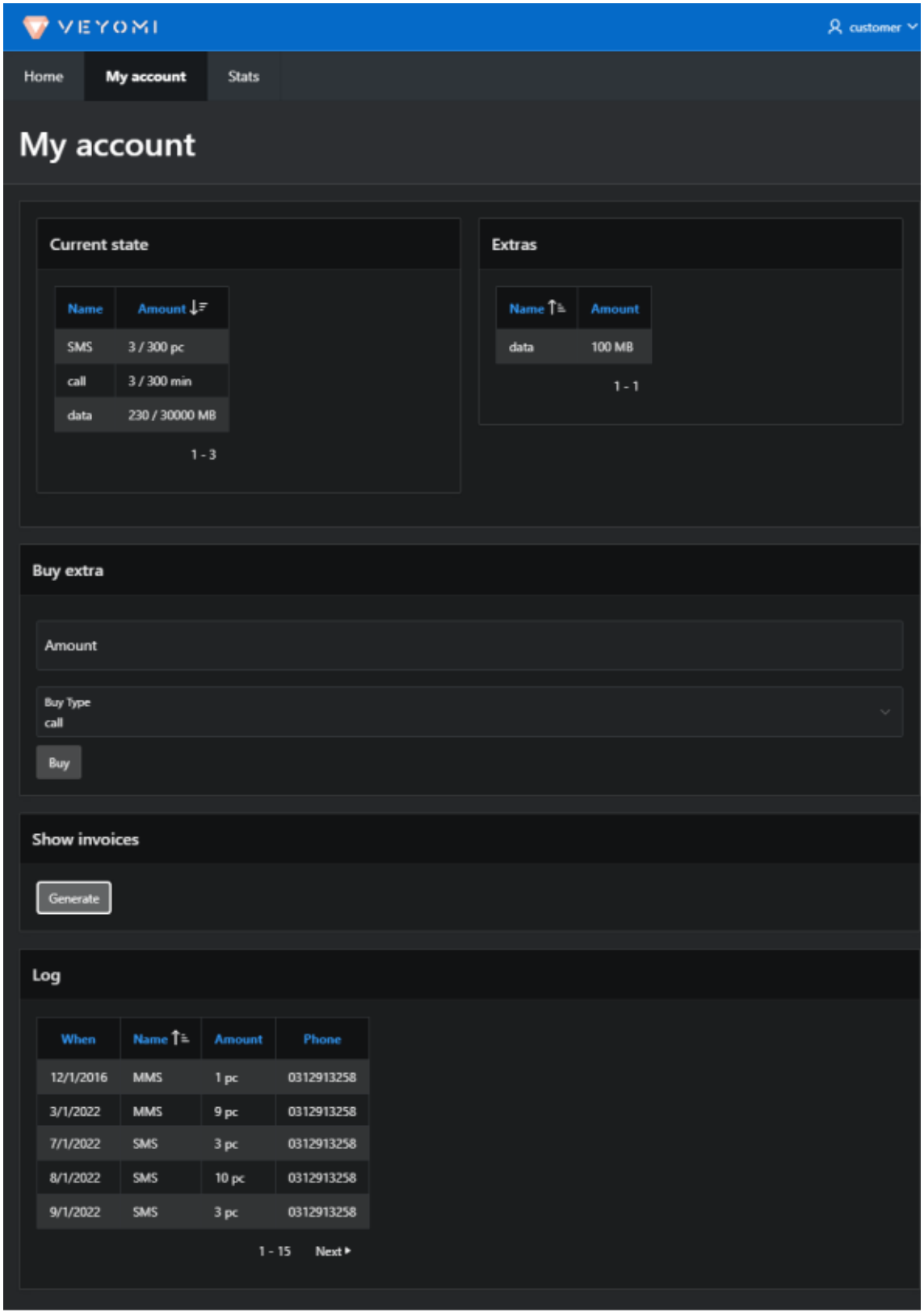
23.6 Sučelja aplikacije

Dizajn aplikacije je stvaranje pojedinačnih podstranica koje će pružati predložene usluge. Primarna stranica je početna na kojoj se nalazi pregled ponuđenih usluga i portfelja paušalnih usluga. Ovu stranicu moguće je koristiti za predstavljanje tvrtke i usluga koje kupac može kupiti. Ova stranica je dostupna svim posjetiteljima bez prijave. Glavna stranica nudi još jednu važnu funkcionalnost: pristup administraciji nakon prijave. Pod administracijom podrazumijevamo skup podstranica podijeljenih prema vrsti računa na koji se prijavljuje - korisnički, menadžerski ili admin. Dakle, za svaku od njih potrebno je izraditi zasebne podstranice koje pokrivaju predloženu funkcionalnost servisa pojedine uloge.

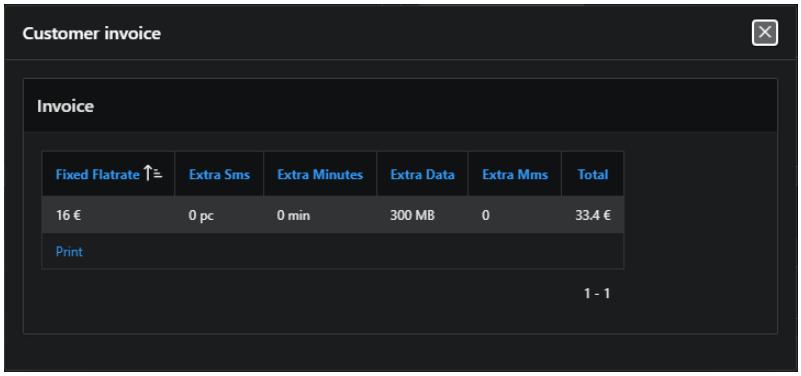
Na korisničkoj stranici nalazi se pregled trenutnog statusa naručenih usluga i dodatnih usluga u pojedinim poljima. Još jedna dodatna kartica omogućuje kupnju određenih usluga odabirom vrste usluge iz kombiniranog okvira. Druga funkcionalnost omogućuje prikaz i generiranje računa, što je moguće riješiti pop-up prozorom, u kojem se nakon otvaranja prikazuje račun u obliku izvješća, što omogućuje konfiguraciju spremanja i ispisa u PDF formatu. Izvještaj za račun poziva posebna skripta koja sumira podatke o pruženoj usluzi u određenom mjesecu. Drugi kritični dio korisničke stranice je pregled korištenja usluge s pojedinostima kao što su količina, vrsta usluge i vrijeme. Ova tablica omogućuje filtriranje i sortiranje ove recenzije pomoću specifikacije filtera i sortiranje po pojedinačnim stupcima. Takva funkcionalnost može biti od koristi klijentu kada želi dobiti npr. pregled broja SMS poruka koje je poslao u određenom razdoblju. Na kraju, kupcu je omogućena funkcionalnost prikaza statistike u grafikonima u pojedinoj mapi. Ova funkcionalnost se obrađuje na posebnoj stranici koja je tabulatorom povezana sa stranicom kupca, a moguće je mijenjati pojedine funkcionalnosti. Stranica sa statistikom sadrži demonstraciju grafičkog prikaza logotipa. Takav prikaz je prikladan prikaz za prikaz promjena tijekom vremena. Na primjer, lako možemo prikazati minute pozvane tijekom pojedinih mjeseci, broj poslanih SMS-ova ili količinu prenesenih podataka u naručenom programu.

23.7 Dizajn aplikacije

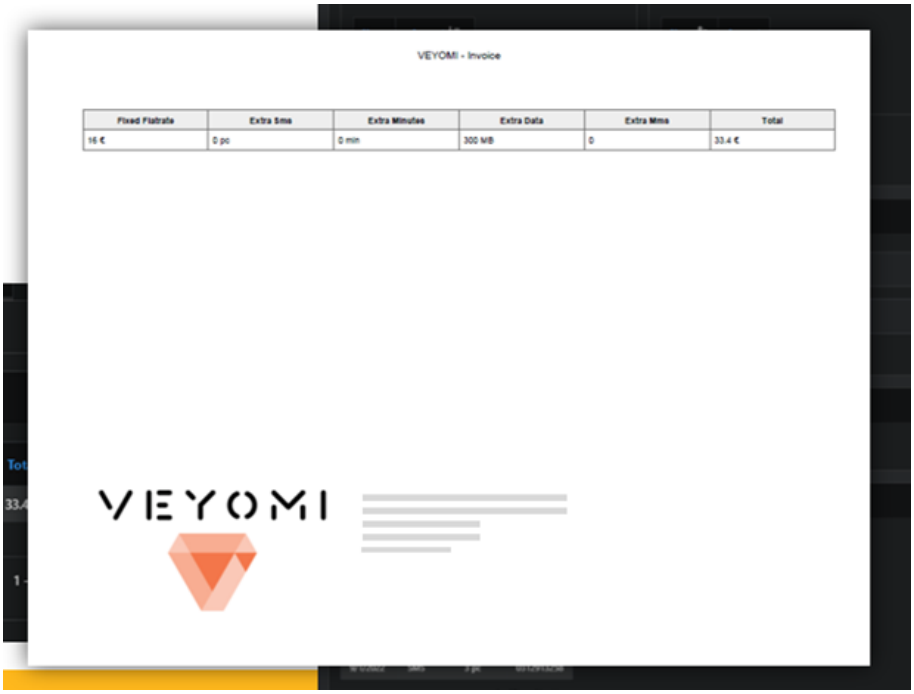
Ovo poglavlje prikazuje snimke zaslona aplikacijskih obrazaca različitih korisničkih uloga.



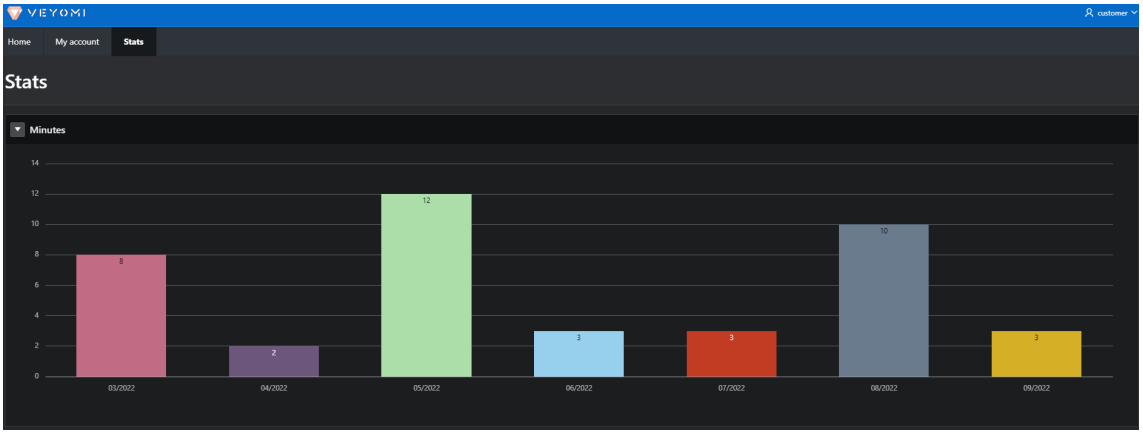
Slika 23.4: Nadzorna ploča kupca



Slika 23.5: Nadzorna ploča kupca – modalni prozor računa



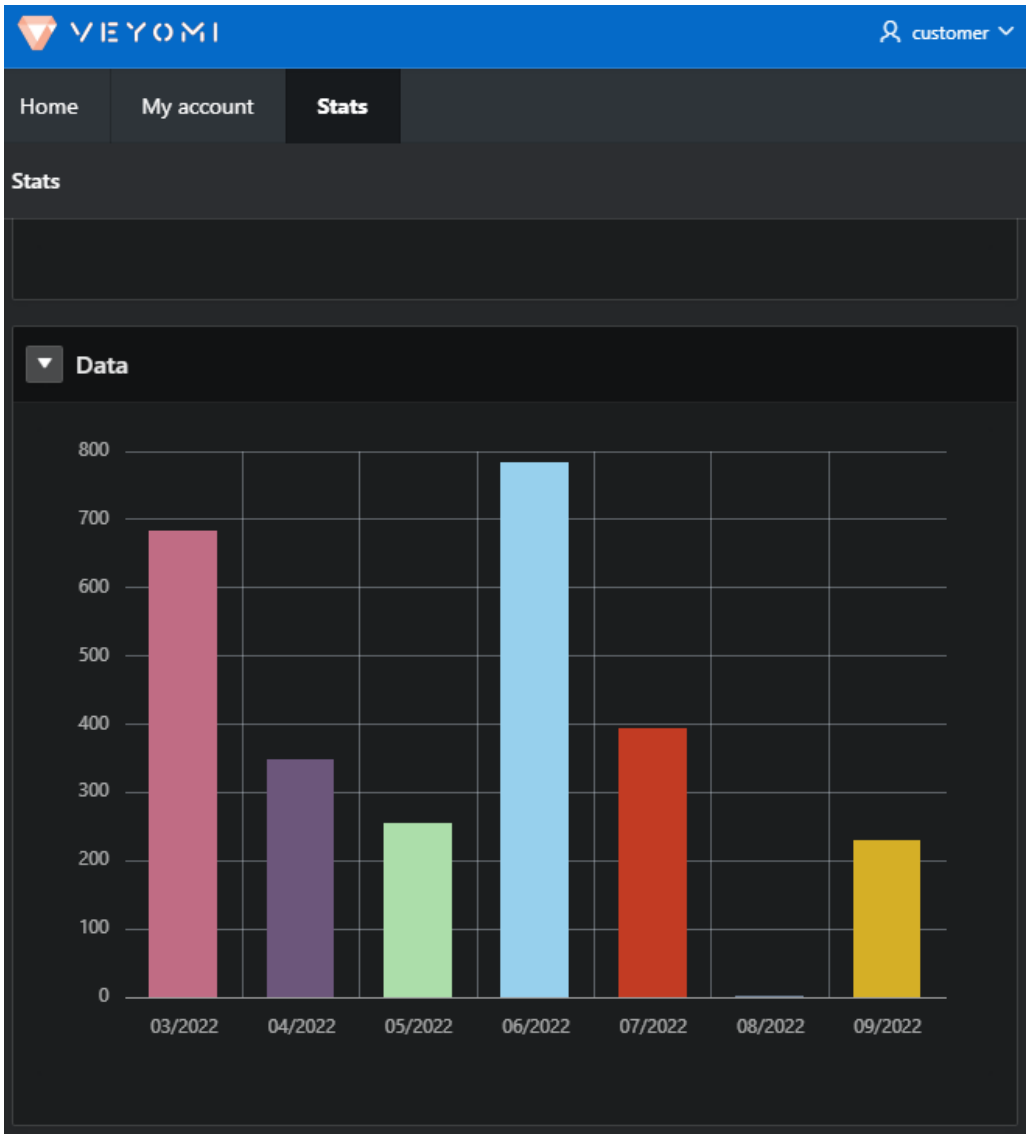
Slika 23.6: Nadzorna ploča kupca – Račun u PDF-u



Slika 23.7: Nadzorna ploča kupca – Statistika minuta



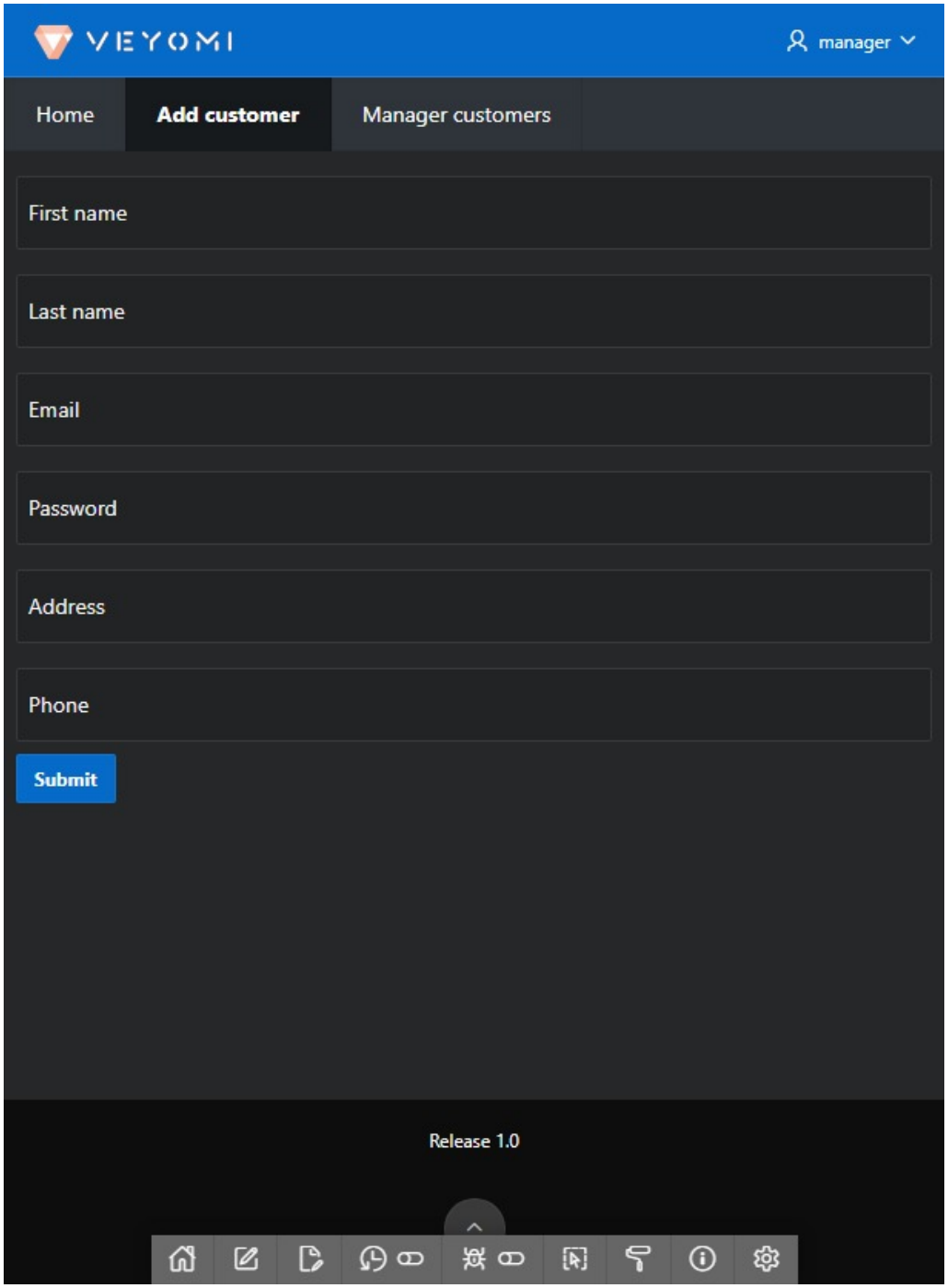
Slika 23.8: Nadzorna ploča kupca – Statistika SMSova



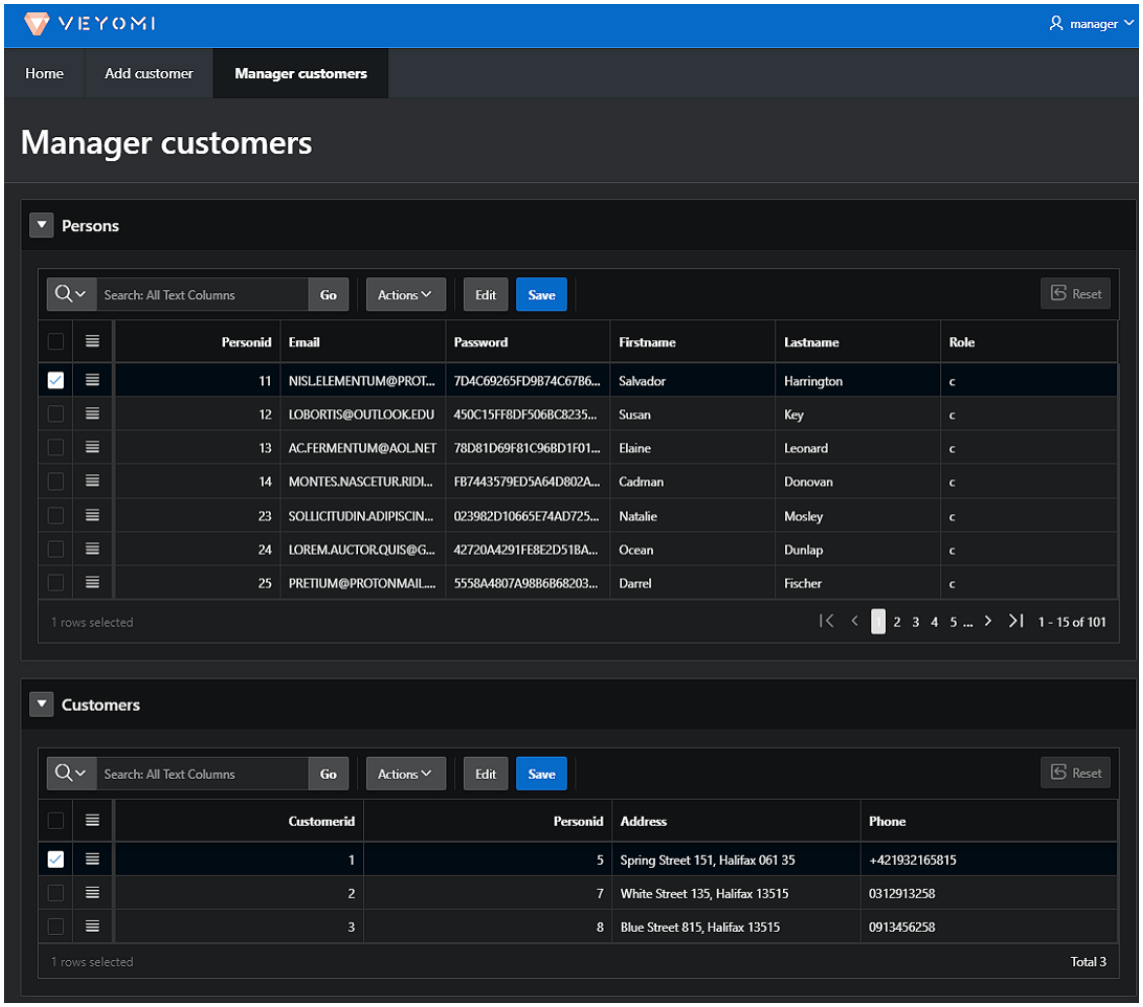
Slika 23.9: Nadzorna ploča kupca – Statistika podataka

Nakon što se menadžer prijavi u aplikaciju, funkcionalnost upravljanja kupcima dostupna je na zasebnim karticama koje su dio stranice menadžera. Prva funkcionalnost je dodavanje novog

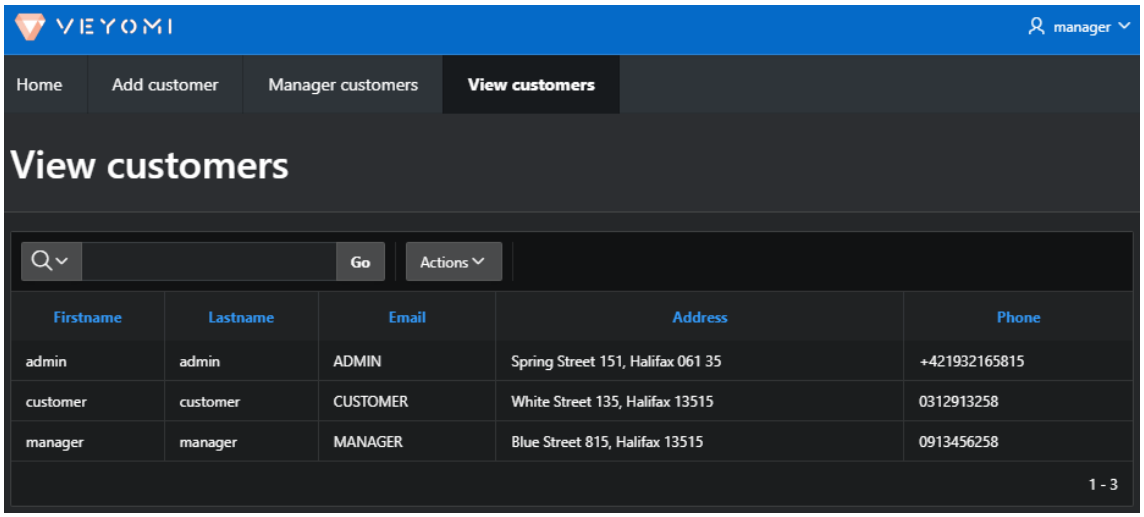
kupca, gdje menadžer može dodati i kreirati novi korisnički račun. Zatim koristimo komponentu upravljanja kupcima za uređivanje podataka o kupcima, koji su kreirani u podatkovnoj mreži. Ova komponenta također pruža funkcionalnost jednostavnog izvješćivanja ili izvoza podataka o kupcima u druge aplikacije trećih strana, ako je potrebno. Na kraju, kartica prikaza kupaca prikazuje pojednostavljeni oblik prikaza pregleda podataka o kupcima. Ova rešetka je konfigurirana tako da ne dopušta uređivanje pojedinačnih stavki, ali ima mogućnost kreiranja izvještaja i izvoza iz ove liste.



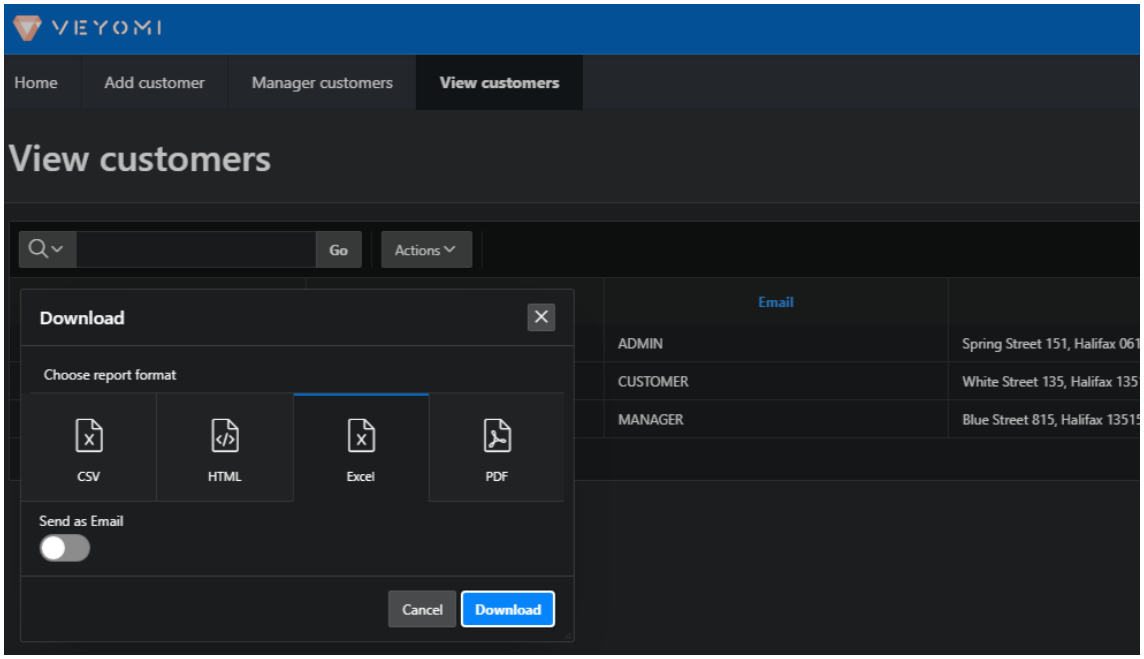
Slika 23.10: Nadzorna ploča menadžera – Dodavanje kupca



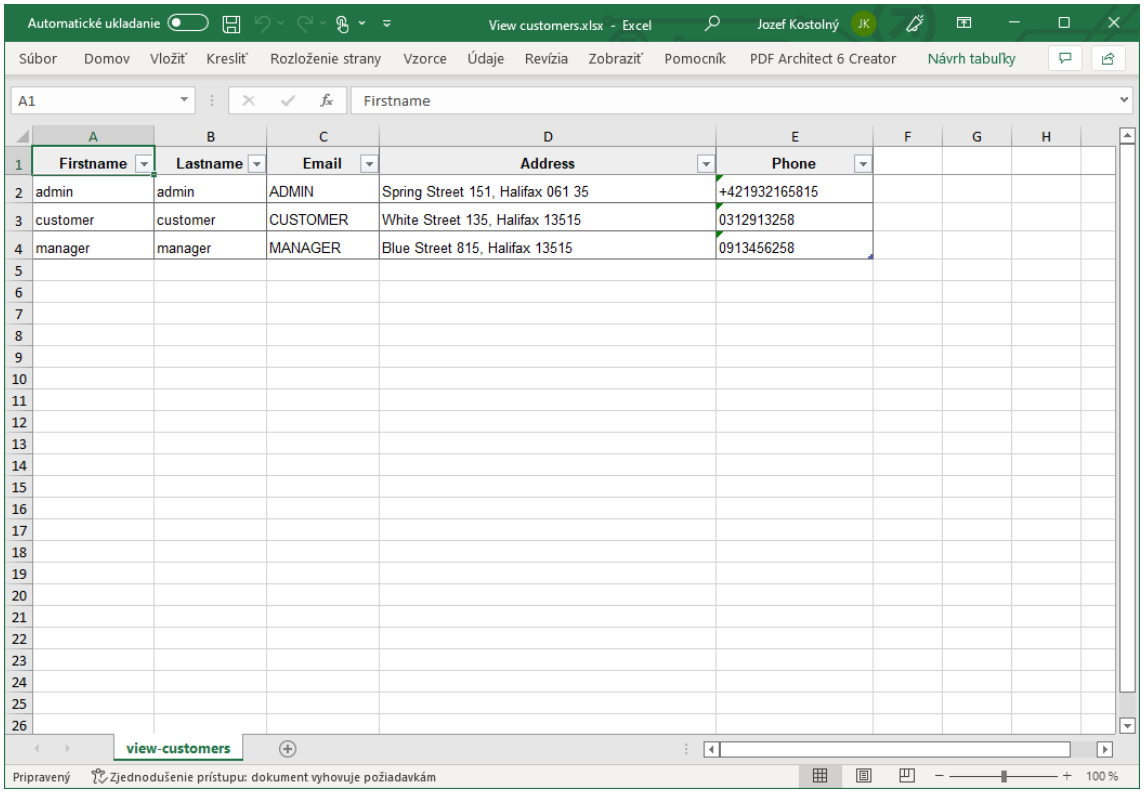
Slika 23.11: Nadzorna ploča menadžera – Upravljanje kupcem



Slika 23.12: Nadzorna ploča menadžera – Pregled kupca

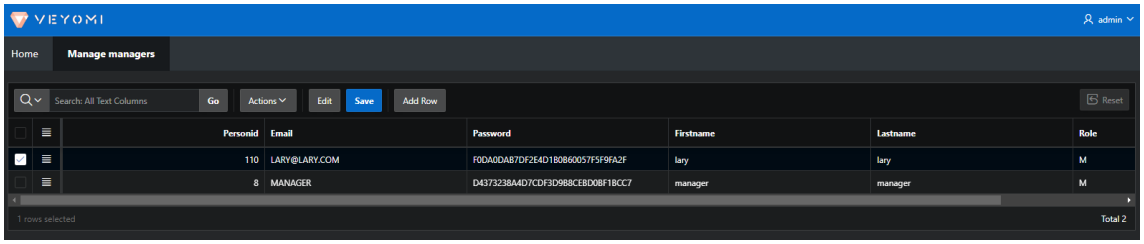


Slika 23.13: Nadzorna ploča menadžera – Izvoz kupca



Slika 23.14: Nadzorna ploča menadžera – Izvoz kupca u XLS datoteku

Zadnja stranica koju je potrebno izraditi je stranica za administratora aplikacije. Na ovoj stranici pružamo uslugu upravljanja meandžerima u mreži podataka koja se može uređivati, omogućujući izvješća o izvozu i funkciju filtriranja.

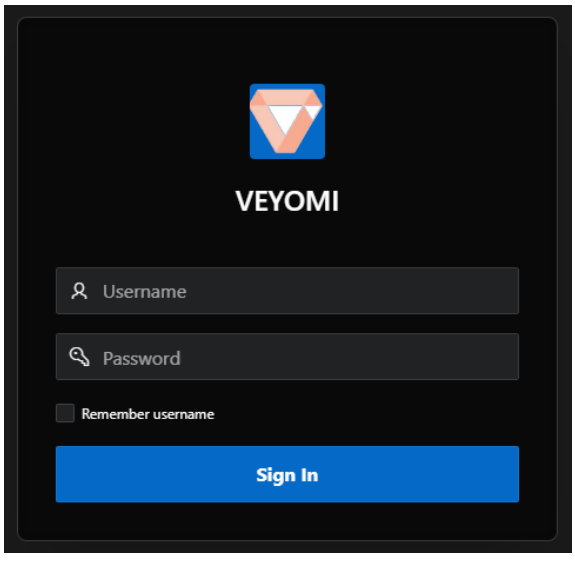


Slika 23.15: Nadzorna ploča administratora

Važan dio je izrada obrasca za prijavu. U našoj aplikaciji razmišljamo o postavljanju ove forme na određenu stranicu, koja se uvijek poziva prilikom ulaska na sigurne stranice, gdje je potrebno što prije izvršiti autentifikaciju.



Slika 23.16: Odredišna stranica s prijavom



Slika 23.17: Stranica za prijavu

23.8 Skripte

```
CREATE SEQUENCE "CUSTOMER_SEQ"
MINVALUE 4 MAXVALUE 9999999999999999999999999999
INCREMENT BY 1
START WITH 4
NOCACHE NOORDER NOCYCLE NOKEEP NOSCALE GLOBAL
/

CREATE OR REPLACE EDITIONABLE TRIGGER "BI_CUSTOMER"
BEFORE INSERT ON CH23_CUSTOMER
FOR EACH ROW
BEGIN
SELECT CUSTOMER_SEQ.nextval INTO :new.customerid
FROM dual;
END;
/

ALTER TRIGGER "BI_CUSTOMER" ENABLE;
/
```

Listing 23.4: Trigger on customer sequence.

```
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 on inserting a person.

```
CREATE OR REPLACE FUNCTION AUTHENTICATE_USER
(p_username in varchar2,
p_password in varchar2)
 return boolean
IS
l_user_name         CH23_PERSON.email%type := UPPER(p_username);
l_password          CH23_PERSON.password%type;
l_hashed_password varchar2(1000);
l_count             number;
BEGIN
SELECT COUNT(*) INTO l_count
FROM CH23_PERSON
WHERE email = l_user_name;

IF l_count > 0 THEN
l_hashed_password := hash_password(l_user_name, p_password);
SELECT password INTO l_password
FROM CH23_PERSON
WHERE email = l_user_name;

IF l_hashed_password = l_password THEN
APEX_UTIL.SET_AUTHENTICATION_RESULT(0);
 return true;
ELSE
APEX_UTIL.SET_AUTHENTICATION_RESULT(4);
 return false;
END IF;

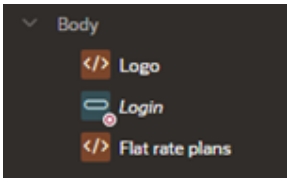
ELSE
APEX_UTIL.SET_AUTHENTICATION_RESULT(1);
 return false;
END IF;
APEX_UTIL.SET_AUTHENTICATION_RESULT(7);
 return false;
EXCEPTION
WHEN OTHERS THEN
APEX_UTIL.SET_AUTHENTICATION_RESULT(7);
APEX_UTIL.SET_CUSTOM_AUTH_STATUS(sqlerrm);
 return false;
END AUTHENTICATE_USER;
/
```

Listing 23.6: Function AUTHENTICATE_USER.

23.9 Izrada početne stranice

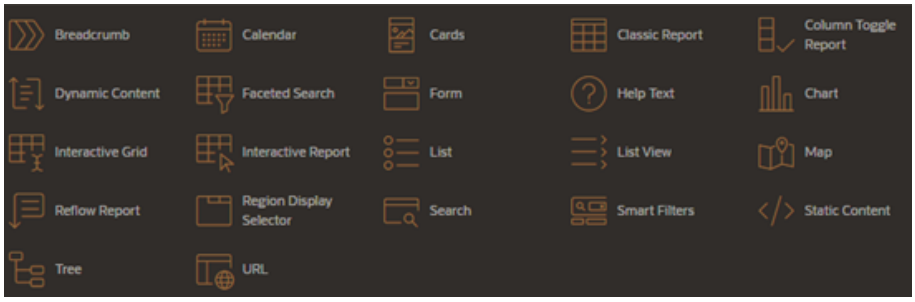
Koristeći Create a Page opciju, možete birati između različitih predložaka, prazne stranice ili stranica značajki s unaprijed definiranim funkcijama, kao što su Login, About page, and Configuration Options. Najprije morate ispuniti osnovne informacije, kao što su broj i naziv stranice i odabrati model stranice, kao što je Normal, Modal Dialog, ili Drawer. Također možete ispuniti dodatne informacije, kao što je postavljanje navigacijskog unosa za stranicu (hoće li to biti gornji unos ili ugniježđen u postojeći unos) i odabir ikone prikazane u izborniku nadzorne ploče.

Nakon klika na Create Page, prikazuje se nadzorna ploča za uređivanje informacija i sadržaja stranice. Glavni sadržaj stranice (Body) podijeljen je na regije.



Slika 23.18: Regije glavnog sadržaja

Svaka regija može sadržavati HTML ili različite sadržaje, poput statičnog HTML-a ili unaprijed definiranog sadržaja.



Slika 23.19: Lista sadržaja

Naša regija **Login** sadrži gumb za preusmjeravanje stranice u obrazac za prijavu korisnika. U dijelu Behaviour property of the button, definirali smo radnju (Action) koja je postavljena kao Redirect to Page in this Application. Zatim postavljamo Target na unaprijed definiranu stranicu za prijavu. Budući da nije potrebno prikazati ovaj gumb kada je korisnik već prijavljen, moramo sakriti gumb postavljanjem Type as User kao Public User (korisnik nije autentificiran). Finalno, stvorili smo **Flat rate plans** kao statično područje sadržaja koje sadrži HTML kod, za koji možemo definirati CSS.



Slika 23.20: HTML kod statičke regije

23.10 Izrada korisničke stranice

Ovu stranicu izradili smo koristeći slične korake kao kod izrade početne stranice. Kada se stranica izradi, nastavljamo s razvojem izgleda korisničke stranice, koja je podijeljena u nekoliko regija, kao što su prikaz trenutnog stanja usluga, kupnja dodataka, prikaz faktura i prikaz zapisa.

Regija **Current state** sastoji se od dvije podregije, koje prikazuju numeričke podatke – trenutnu cijenu potrošenih jedinica paušala i dodataka. Za dobivanje ovih podataka koristi se SQL upit.

```
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 Query - Consumed units of a flat rate.

```
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 Query - Consumed units of extra services.

Regija **Buy extra** sadrži brojčano polje, popis za odabir i gumb. Moguće je odrediti iznos i odabrati koju dodatnu uslugu želimo kupiti. Njime se upravlja pomoću gumba koji izvršava proces koji pokreće PL/SQL kod:

```
INSERT INTO CH23_EXTRASERVICE VALUES (
:BUY_TYPE,
```

(**SELECT** customerid **FROM** CH23_CUSTOMER **JOIN** CH23_PERSON **USING** (personid)
WHERE email = :APP_USER),sysdate,:BUY_AMOUNT);

Listing 23.9: PL/SQL code for inserting an extra service.

Regija **Show invoices** se sastoji od gumbića koji se koriste za kreiranje računa. U *Behaviour* svojstvima gumba, definirali smo *Action*, koji je postavljen kao *Redirect to Page in this Application*. Dalje smo postavili *Target* na unaprijed definiranu stranicu s računom kupca, koja je postavljena kao modalna. Na ovoj stranici se pomoću SQL upita prikazuje klasično izvješće koje se može i ispisati:

```
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 Query for an invoice report.

Regija **Log** sadrži interaktivno izvješće za prikaz dnevnika korištenja upotrebom SQL upita:

```
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 Query – Usage log.

Tab **Stats** je nova stranica koja se sastoji od tri regije: minute, SMS, i pdoaci. Tip regije je postavljen na *Static Content*, uključujući *Chart* komponente koje prikazuju karte iz serije. Njegov izvor je definiran SQL upitom. Prema zadanim postavkama, boja grafikona postavljena je na ujednačenu. Može se prilagoditi u SQL upitu:

```
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 Query - Statistics.

23.11 Kreiranje stranice menadžera

Na stranici menadžera koristili smo slične metode i komponente za izradu podstranica. Svaka podstranica kreirana je kao nova stranica, kao što je Dodaj kupca (engl. Add customer), Upravljaj kupcima (engl. Manage customers) i Pogledaj kupce (engl. View customers).

Stranica *Add customer* sadrži jednostavan oblik koji je stvoren od komponenti. Proces dodavanja novog korisnika izvodi se PL/SQL kodom:

```
INSERT INTO CH23_PERSON (EMAIL, PASSWORD, FIRSTNAME, LASTNAME,
    ROLE)
VALUES (:CUSTOMER_EMAIL, :CUSTOMER_PASSWORD, :
    CUSTOMER_FIRSTNAME,
```

```
:CUSTOMER_LASTNAME, 'C');
```

Listing 23.13: PL/SQL code for inserting a person.

Stranica *Manage customers* kreiran je s dvije interaktivne mreže: Upravljanje osobama (engl. Manage persons) i Upravljanje trenutnim kupcima (engl. Manage current customers). Interaktivne mreže povezane su s procesom uređivanja i spremanja promjena. Vrsta procesa postavljena je na*Interactive Grid – Automatic Row Processing (DML).*

Stranica *View customers* sadrži interaktivno izvješće za prikaz podataka bez uređivanja. Ovo izvješće je kreirano pomoću SQL upita:

```
SELECT firstname, lastname, email, address, phone
FROM CH23_PERSON
JOIN CH23_CUSTOMER USING (personid);
```

Listing 23.14: SQL Query - Customers.

23.12 Dodatni materijal za učenje

Dostupni su vam sljedeći dodatni materijali za učenje:

- skripta za izradu i popunjavanje tablica,
- skripta za ispuštanje tablica,
- izvezena aplikacija,
- video koji pokazuje kako generirati aplikaciju.

Sav dodatni materijal za učenje dostupan je na public BeeAPEX project page. Prijavite se kao gost (lozinka nije potrebna). Navigirajte do sekcije Books, skripte u mapi Part 2 > Chapter23 u sekciji Scripts i video upute u Collection of video guides. Materijal za kratke tečajeve dostupan je u Short courses section.

23.12.1 Izvezena aplikacija

Izvezena aplikacija je zapakirana. Instalacija kreira tablice i popunjava podatke. Deinstalacija uklanja sve objekte baze podataka koji se koriste u ovoj aplikaciji.

Zapakirana aplikacija je testirana i pokrenut će se u novom radnom prostoru ako su ispunjeni sljedeći zahtjevi:

- dodajte APEX korisnika prije pokretanja aplikacije. U radnom prostoru za razvoj i testiranje navigirajte do Shared Components > Application Access Control > Add User Role Assignment; unesite APEX korisnika i dodajte mu uloge Administrator, Contributer and Reader. U produkciji prije razvoja potrebno je konzultiranje sa stručnim osobljem.

Ako korisniku nije dodijeljena odgovarajuća uloga, uvezena aplikacija će se srušiti. Potrebno je izbrisati kolačiće web preglednika (npr. Firefox: Settings > Cookies and Site Data > Manage Data) nakon što se aplikacija sruši zbog neispunjenih zahtjeva.

23.12.2 Video upute

Video upute prikazuju svaki korak u razvoju aplikacije.

23.13 Pitanja

1. Koja se podstranica koristi za izradu nove sheme prijave?
2. Navedite najmanje dvije metode provjere autentičnosti za upravljanje korisnicima.
3. Tijekom implementacije, različite stavke su navedene u Select izjavama. Kako možete identificirati te stavke?

23.14 Odgovori

1. U alatu Oracle Apex odabiremo metodu provjere autentičnosti - Authentication Scheme - u dijelu Shared Components => Authentication Schemes podstranica.
2. Postoji više metoda koje se mogu koristiti za autentifikaciju korisnika, poput LDAP poslužitelja, računa baze podataka, računa Oracle APEX itd.
3. Stavke u izjavama Select navedene su pomoću dvotočke na početku naziva stavke.



24. Studija slučaja iznajmljivanja automobila

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

24.1 Poslovni pogled na slučaj

Iznajmljivanje automobila može biti praktičan način za istraživanje novih odredišta, ali morate biti svjesni nekih potencijalnih zamki. Važno je razumjeti da je najam automobila sličniji rezerviranju hotelske sobe nego rezerviranju zrakoplovne karte. Kod iznajmljivanja automobila postoje dvije vrste naknada: naknada prilikom preuzimanja automobila i unaprijed plaćena naknada.

Za osobe koje prvi puta iznajmljuju automobil, obično je najbolje ugovoriti naknadno plaćanje, što omogućuje veću fleksibilnost u slučaju promijene plana. Ova vrsta najma obično nema kaznu za otkazivanje, što može biti od pomoći ako se pojave nepredviđene okolnosti.

Kako biste izbjegli dodatne troškove, važno je pažljivo pregledati ugovor o najmu i sve dodatne naknade koje mogu nastati, kao što su osiguranje ili dodatne naknade za vozača. Osim toga, svakako pažljivo pregledajte vozilo prije nego što napustite parkiralište i odmah prijavite svaku štetu tvrtki za iznajmljivanje kako biste kasnije izbjegli odgovornost za spomenuto.

Razumijevanjem procesa najma i potencijalnih naknada osiguravate si neometano iskustvo najma bez stresa pri čemu možete u potpunosti uživati u slobodi i fleksibilnosti koje najam automobila može ponuditi tijekom putovanja.

24.2 Definicija problema

Sa svim pravilima i uvjetima, iznajmljivanje automobila može izgledati zastrašujuće, međutim ona postoje s dobrim razlogom. Potpuno novi automobil košta u prosjeku oko 36 000 EUR, tako da kada unajmite automobil, u biti plaćate mali dio njegove ukupne vrijednosti za privremenu upotrebu vozila.

Iako je ključno imati odgovarajuće osiguranje prilikom iznajmljivanja u slučaju nezgode ili nepredviđenih okolnosti, većina iskustava prolazi bez incidenata. Međutim, i dalje je važno pažljivo pročitati ugovor o najmu, razumjeti odredbe i uvjete, kao i pregledati vozilo radi utvrđivanja bilo kakvih oštećenja prije nego što ga prihvatite kako biste izbjegli potencijalne probleme.

Slijedeći ove smjernice, možete imati sigurno i ugodno iskustvo najma bez ikakvih neočekivanih iznenađenja. I zapamtite, pravila i uvjeti postoje kako bi zaštitili i vas i tvrtku za iznajmljivanje automobila, stoga je uvijek najbolje pridržavati ih se za nesmetano iskustvo iznajmljivanja.

24.3 Slučajevi korištenja

24.3.1 Narativni opis slučaja korištenja

Cilj ovog projekta je razviti funkcionalnu aplikaciju koja simulira rad tvrtke za iznajmljivanje automobila. Aplikacija će koristiti platformu Oracle Apex za upravljanje i pohranu detaljnih podataka o kupcima, automobilima i najmu.

Aplikacija je osmišljena tako da za vlasnika tvrtke za iznajmljivanje automobila bude jednostavna za korištenje. Ista omogućuje vlasnicima upravljanje svim potrebnim informacijama, uključujući informacije o najmu, dostupnim automobilima za najam i podacima o kupcima. Korisnici će moći iznajmljivati automobile na temelju kriterija kao što su marka, model, godište i vrsta goriva, a unutar iste će biti prikazani svi potrebni detalji potrebni za završetak najma.

Osim toga, aplikacija će detaljno prikazati podatke kao što su trenutni najam, dostupni automobili za najam i detalji o kupcima. Trenutačni najmovi bit će prikazani korisniku u tablicama, ovisno o kartici u kojoj se korisnik nalazi.

Općenito, ova će aplikacija pružiti učinkovit i organiziran sustav za upravljanje tvrtkom za iznajmljivanje automobila, s detaljnim informacijama dostupnim pritiskom na gumb. Jednostavno sučelje i opsežni alati za upravljanje podacima učinit će vođenje tvrtke za iznajmljivanje automobila jednostavnijim i učinkovitijim.

24.3.2 Polustrukturirani opis

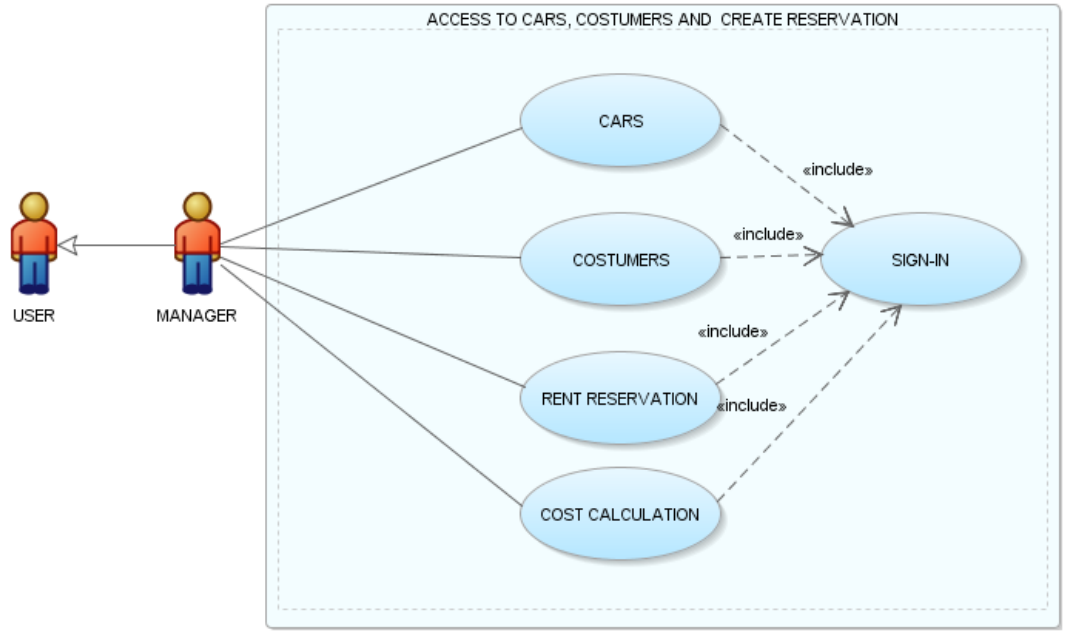
Ovaj projekt ima za cilj razviti aplikaciju u potpunosti prilagođenu korisniku koja koristi Oracle Apex za upravljanje detaljnim informacijama o klijentima, automobilima i iznajmljivanjima u poslovanju tvrtke za iznajmljivanje automobila. Aplikacija će korisnicima omogućiti iznajmljivanje na temelju određenih kriterija, prikaz potrebnih informacija o najmu te će pružiti sveobuhvatne alate za upravljanje podacima pri prikazu trenutnih najmova, dostupnih automobila za najam i detalja o kupcima. Omogućavajući učinkovit i organiziran sustav za upravljanje tvrtkom za iznajmljivanje automobila, ova aplikacija će vođenje tvrtke učiniti jednostavnijim i učinkovitijim. Polustrukturirani opis nalazi se u tablici 24.1.

24.3.3 Dijagram slučaja korištenja

Prethodno spomenuto prikazano je na dijagramu slučaja korištenja (vidi sliku 24.1).

Tablica 24.1: Opis slučaja korištenja: pristup automobilima, kupcima i rezervacijama vezanim uz najam automobila

Ključna riječ	**Vrijednost**
ID:	*Ch24-01*
Naslov:	*Pristup popisu automobila i kupaca, mogućnost dodavanja i uklanjanja podataka i kreiranje nove rezervacije za najam automobila odabirom željenih datuma i izračunom cijene.*
Opis:	*Korisnik će putem Apex aplikacije pristupiti i pregledati automobile koji pripadaju tvrtki, kao i sve kupce koji su prethodno izvršili rezervaciju. I automobili i kupci opisuju se skupom atributa. Dodatno, korisnik će imati mogućnost kreiranja, ažuriranja i brisanja podataka za obje kategorije, omogućavajući učinkovito upravljanje sustavom za najam automobila. Nadalje, korisnik može kreirati novu rezervaciju za najam automobila dodavanjem automobila, kupaca i datuma. Sustav izračunava ukupnu cijenu rezervacije na temelju odabranih datuma i sukladno tome ažurira rezultate. Završno izvješće predstavlja detalje rezervacije, uključujući automobil, kupca, datume najma i ukupne troškove.*
Primarna uloga:	*Korisnički administrator*
Preduvjeti:	Korisnik ima administratorski račun za APEX instancu.
Postuvjeti:	Nakon ažuriranja i odabira automobila i korisnika najma (kupac), korisnik će moći kreirati rezervaciju najma automobila unutar APEX instance. Slijedi završna rezervacija najma automobila i izračun ukupnih troškova.
Glavni	Scenarij
Uspješan scenarij:	1. Otvaranje web preglednika i prijava u aplikaciju za iznajmljivanje automobila. 2. Odabir označene stavke izbornika ili navigacija stranicom **Cars** ili **costumers**. 3. Za dodavanje automobila potrebno je odabrati gumb "Create". Za kupce vrijedi isti postupak. 4. Za uređivanje unosa automobila ili kupca, potrebno je odabati simbol olovke povezan s određenim unosom. 5. Za dodavanje ili ažuriranje podataka potrebno je odabrati gumb "Apply Changes". Za brisanje podataka potrebno je odabrati gumb "Delete". 6. Odabir stavke izbornika ili navigacija stranicom **rentcar**. 7. Za kreiranje novog najma potrebno je odabrati opciju **create**. 8. Za dodavanje automobila u rezervaciju potrebno je odabrati opciju **carid** i izabrati automobil s popisa dostupnih automobila. 9. Za dodavanje kupca u rezervaciju automobila potrebno je odabrati opciju **custid** i izabrati istog s liste. 10. Za postavljanje datuma rezervacije potrebno je odabrati opciju **rent date** i izabrati željeni datum početka rezervacije. 11. Nakon toga potrebno je odabrati polje **return date** i definirati datum završetka najma automobila. 12. Kada se automobil vrati, ukupni trošak rezervacije izračunava se na temelju broja dana najma i svih dodatnih troškova. Ako dođe do oštećenja automobila, dodat će se dodatni trošak za pokrivanje troškova popravka. 13. Konačni trošak predstavljen je kupcu zajedno s detaljnim pregledom troškova u izvješću o najmu.
Ekstenzije:	*Nema*
Frekvencija korištenja:	*Broj rezervacija po danu varira ovisno o turističkoj sezoni, ali u prosjeku tvrtka zaprimi oko pet rezervacija dnevno.*
Status:	Završeno



Slika 24.1: Dijagram slučaja korištenja

24.4 Podatkovni model

24.4.1 Narativni opis podatkovnog modela

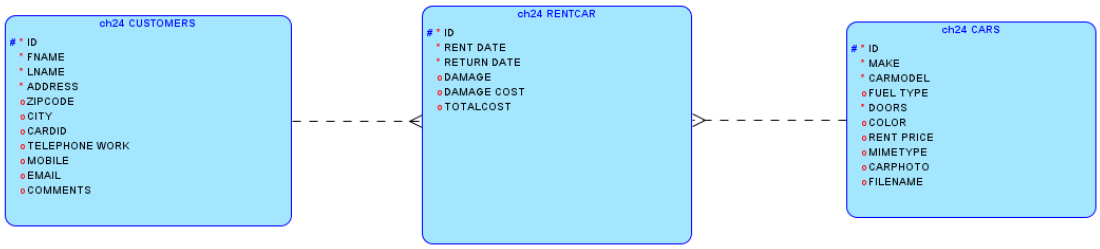
Ovaj edukacijski projekt bazira se na tri tablice koje predstavljaju uključene entitete: tablica automobila, tablica kupaca i tablica iznajmljivanja automobila. Ove tablice sadrže sve relevantne atribute i podatke potrebne za upravljanje procesom najma automobila kroz APEX aplikaciju.

1. Tablica "Automobili"
   - **id** identifikacijski broj i primarni ključ,
   - **make** proizvođač automobila,
   - **carmodel** model automobila,
   - **fuel** benzin, dizel, vozilo na prirodni plin (LNG) ili električno vozilo,
   - **doors** količina vratiju na automobilu,
   - **color** boja automobila,
   - **rent** cijena dnevnog najma specifičnog automobila,
   - **carphoto** fotografija automobila.
2. Tablica "Kupci"
   - **id** identifikacijski broj i primarni ključ kupca,
   - **fname** ime kupca,
   - **lname** prezime kupca,
   - **address** adresa kupca,
   - **zipcode** poštanski broj adrese kupca,
   - **city** grad u kojem živi kupac,
   - **cardid** broj vozačke dozvole ili broj osobne iskaznice kupca,
   - **telephone-work** broj poslovnog telefona kupca,
   - **mobile** broj mobilnog telefona kupca,
   - **email** email adresa kupca,
   - **comments** komentari vezani uz ponašanje kupca.
3. Tablica "Iznajmljivanje automobila"

- **id** identifikacijski broj i primarni ključ tablice iznajmljivanje automobila,
- **carid id** polje povezivanja tablice "Automobili" i tablice "Iznajmljivanje automobila",
- **custid id** polje povezivanja tablice "Kupci" i tablice "Iznajmljivanje automobila",
- **rent date** datum najma,
- **return date** datum povratka,
- **damage** opis oštećenja na automobilu, ako je isto nastalo,
- **damage cost** trošak oštećenja,
- **total cost** ukupni troškovi najma.

24.4.2 Logički model podataka

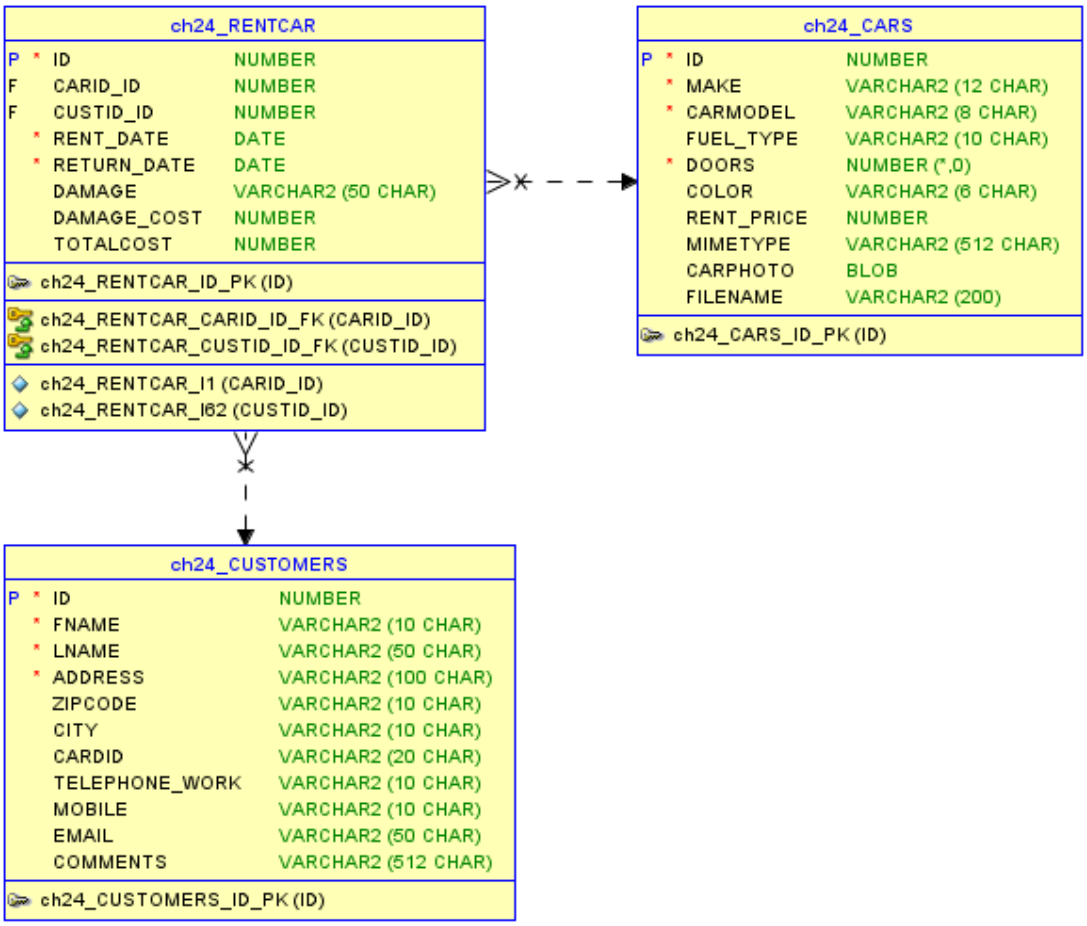
Logički model podataka prikazan je na slici 24.2.



Slika 24.2: Logički model podataka studije slučaja iznajmljivanja automobila.

24.4.3 Relacijski model podataka

Relacijski model podataka prikazan je na slici 24.3.



Slika 24.3: Relacijski model podataka studije slučaja iznajmljivanja automobila

24.4.4 SQL skripta

Tablice potrebne za izradu aplikacije su CH24_CARS, CH24_CUSTOMERS i CH24_RENTCAR. Polja svake tablice opisana su u nastavku:

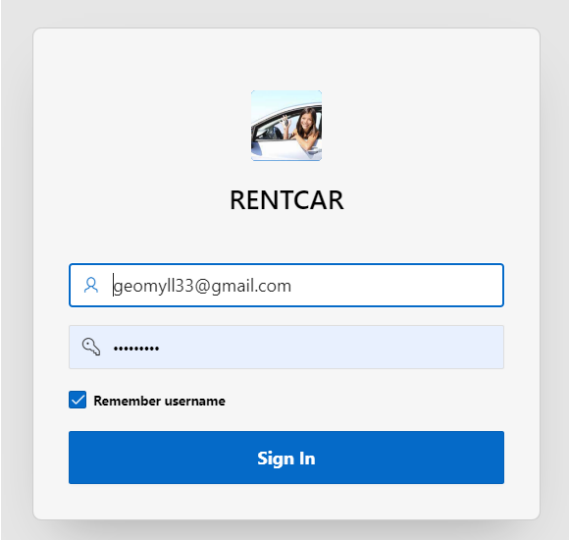
```
drop table  ch24_cars;
drop table  ch24_customers;
drop table  ch24_rentcar;
-- create tables
create table ch24_cars (
    id                  number generated by default on null as identity
                        constraint ch24_cars_id_pk primary key,
    make                varchar2(12 char) 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);
```

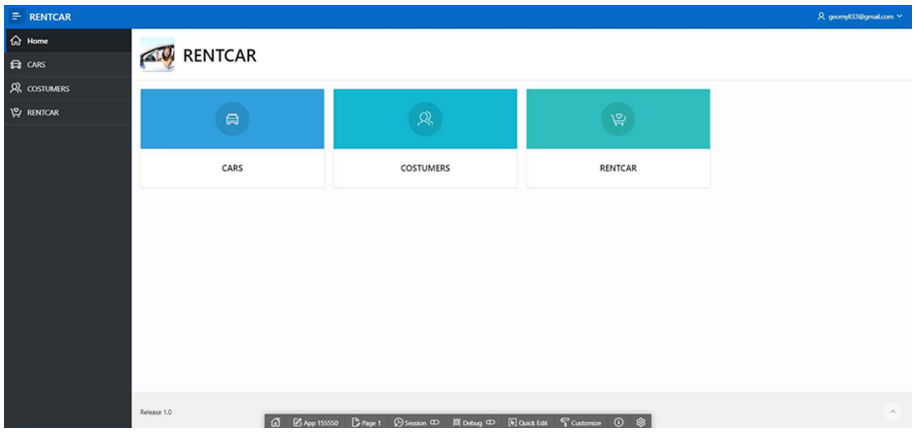
24.5 Sučelja aplikacije

Korisnik, koji je vlasnik tvrtke za iznajmljivanje automobila, mora dati svoje vjerodajnice za pristup aplikaciji. Aplikacija koristi Application Express Accounts authentication za provjeru autentičnosti. Vjerodajnice vlasnika iste su kao i vjerodajnice WORKSPACE prikazane na slici 24.4.



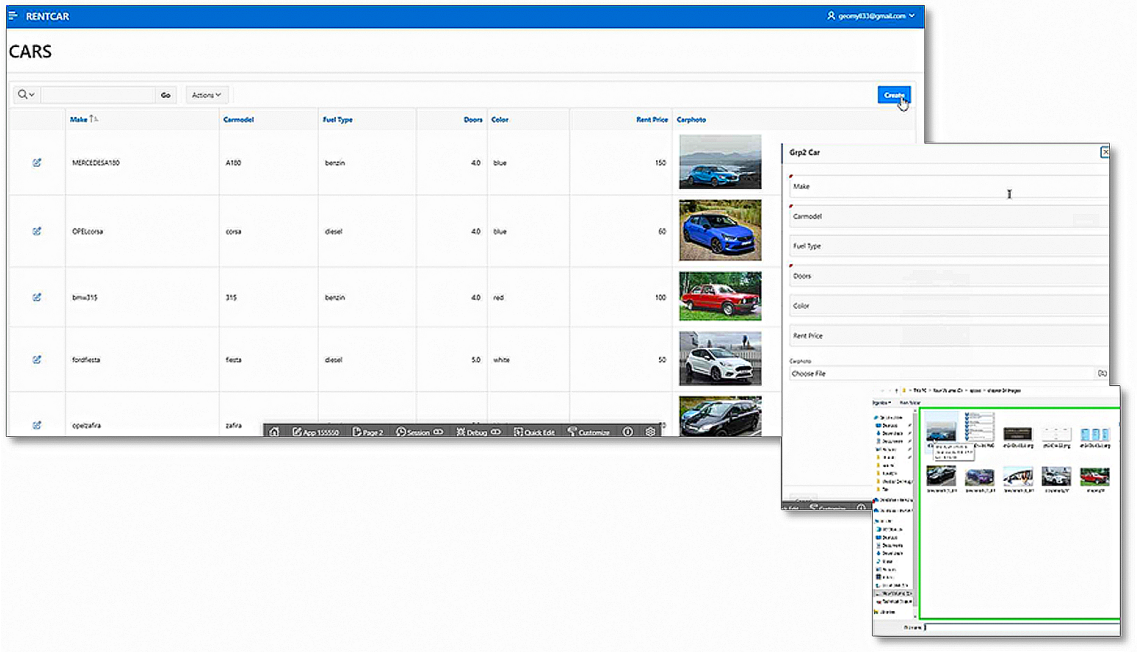
Slika 24.4: Prijava u aplikaciju

Aplikacija simulira rad trgovine za iznajmljivanje automobila i uključuje upravljanje klijentima, automobilima i najmom. Pruža detaljne informacije o tim entitetima pomoću web aplikacije temeljene na platformi Oracle Apex, kao što je prikazano na slici 24.5.



Slika 24.5: Početna stranica aplikacije

Predložak na slici 24.6 dizajniran je za upravljanje informacijama vezanim uz dostupnost automobila u tvrtci za iznajmljivanje automobila. Korisnik, koji je vlasnik trgovine, može pohraniti i upravljati svim potrebnim informacijama o automobilima, kao što su marka, model, godina, vrsta goriva, a može i učitati fotografiju automobila kako bi pokazao njegovo stanje.

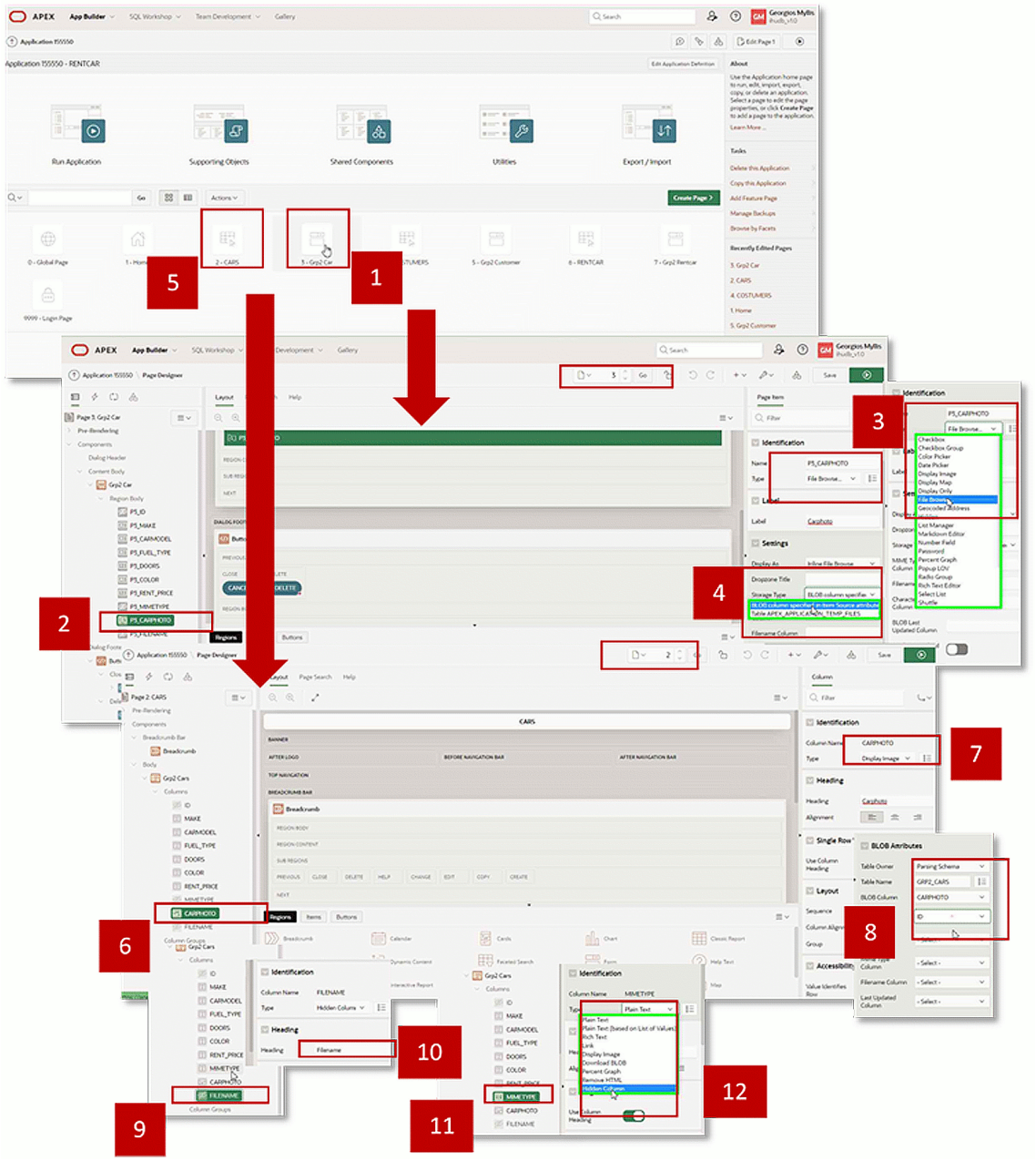


Slika 24.6: Obrazac prikaza automobila u kreiranoj aplikaciji

Na slici 24.7, objašnjeno je kako dodati stupac u obrazac koji se odnosi na automobile, a koji omogućuje prikaz fotografija automobila. Kako bi se isto omogućilo, potrebna su dva pomoćna stupca za spremanje informacija u tablicu - jedan za mime (?) i jedan za naziv datoteke. Stupac koji će pohraniti fotografiju treba postaviti kao tip BLOB. U ovom slučaju naziv stupca je **CARPHOTO**.

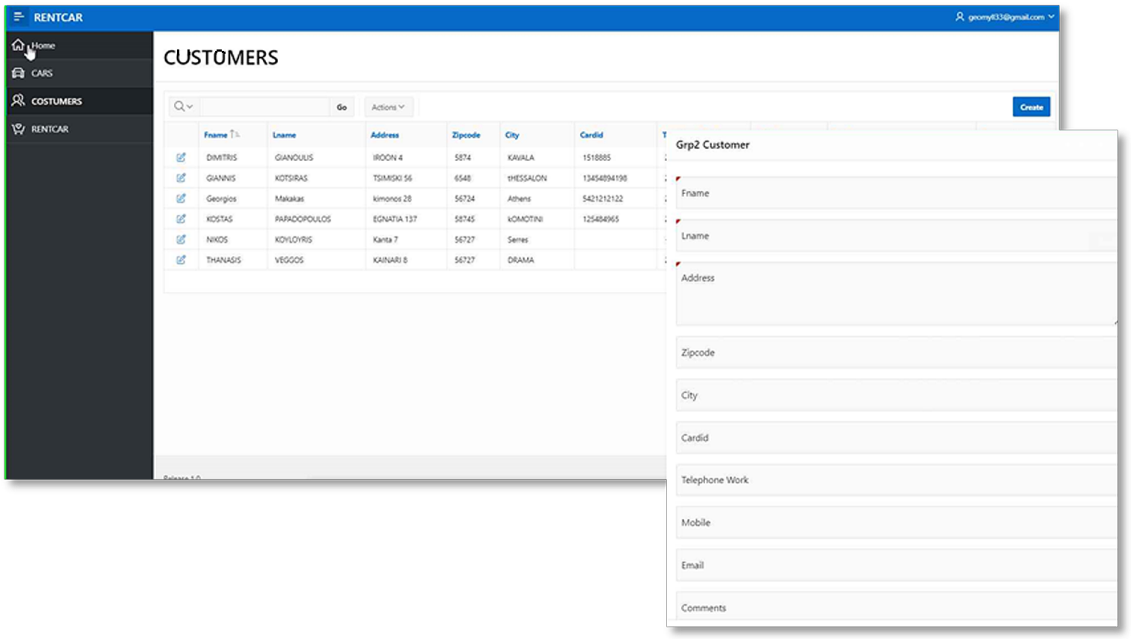
U aplikaciji, pomoću opcije "page designer" potrebno je selektirati izvješće vezano uz CAR obrazac i konfigurirati stupac **CARPHOTO** kao preglednik datoteka. Zatim je potrebno promijeniti postavke vrste pohrane u "BLOB column specifier" u dijelu "item source attributes".

Nadalje, potrebno je odabrati CAR obrazac i konfigurirati **CARPHOTO** stupac kao "Display image". Konfigurirajte BLOB atribute povezivanjem tablice **GRP2-CARS**, stupca **CARPHOTO** i primarnog ključa **ID**. Na kraju, stupci **MIMETYPE** i **FILENAME** trebaju biti skriveni.



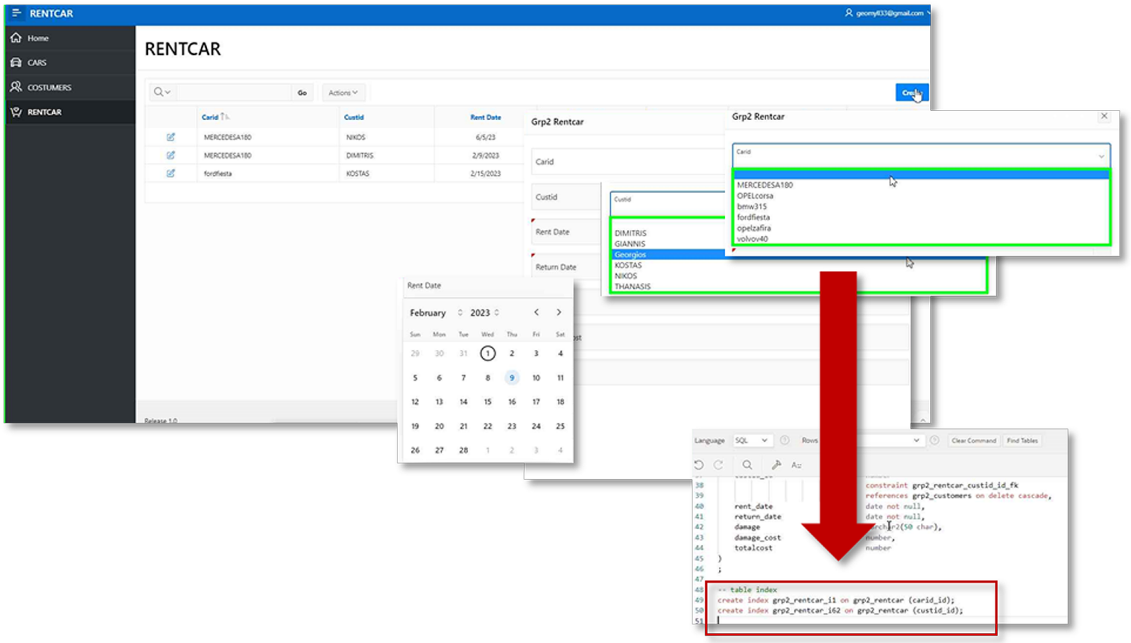
Slika 24.7: Kreiranje stupca s fotografijama

Podaci, primjerice obrasci koji se odnose na kupce, mogu se pohraniti kao što je prikazano na slici 24.8.



Slika 24.8: Obrazac prikaza kupaca u kreiranoj aplikaciji

U obrascu za iznajmljivanje, kao što je prikazano na slici 24.9, korisnik može odabrati automobil na temelju kriterija kao što su marka, model i godina. Najam se zatim povezuje s kupcem i prikazuju se svi potrebni detalji kao što su automobil, kupac, datum početka i završetka najma te ukupni trošak. Korisnik također može dodati dodatni trošak ako je automobil oštećen nakon što je vraćen. Po završetku najma, ukupni trošak će biti izračunat i prikazan u konačnom izvješću.



Slika 24.9: Obrazac za iznajmljivanje automobila u kreiranoj aplikaciji

24.6 Dodatni materijal za učenje

Dostupni su vam sljedeći dodatni materijali za učenje:

- izvezena aplikacija,
- video upute.

Sav dodatni materijal za učenje dostupan je na public BeeAPEX project page. Prijavite se kao gost (lozinka nije potrebna). Navigirajte do sekcije Books, skripte u mapi Part 2 > Chapter24 u sekciji Scripts i video upute u Collection of video guides. Materijal za kratke tečajeve dostupan je u Short courses section.

24.6.1 Izvezena aplikacija

Izvezena aplikacija je zapakirana. Instalacija stvara tablice, indeks, funkciju, proceduru i okidač te popunjava podatke. Deinstalacija uklanja sve objekte baze podataka koji se koriste u ovoj aplikaciji.

24.6.2 Video upute

Video upute prikazuju svaki korak u razvoju aplikacije.

24.7 Pitanja

1. Kako možete dodati ikonu u logotip aplikacije kada ste već izradili aplikaciju?
2. Kako možete dodati slike stranicama tipa izvješće?

24.8 Odgovori

1. U okruženju vaše aplikacije odaberite **Shared Components**, zatim odaberite **User Interface Attributes**, kliknite na **Edit** u padajućem prozoru i dodajte sliku po želji.
2. Prvo morate postaviti vrstu podataka vašeg stupca kao BLOB. Zatim u "Page Designer" za vaš obrazac izvješća konfigurirajte stupac kao preglednik datoteka i promijenite vrstu pohrane u **BLOB column specifier in item Source Attributes**. Zatim navigirajte na obrazac spomenutog predloška u okruženju "App Builder" i konfigurirajte stupac kao **Display image**. Nakon toga konfigurirajte BLOB atribute povezivanjem tablice, stupca i ID-a primarnog ključa koji se odnosi na ovaj stupac.

Literatura

Članci

[1] Hill Kim. „Altruistic cooperation during foraging by the Ache, and the evolved human predisposition to cooperate”. *Human Nature* 13 (ožujak 2002.), stranice 105–128. DOI: 10.1007/s12110-002-1016-3 (citirano na stranci 185).

Knjige

[2] Roy Fielding i Richard N. Taylor. *Principled design of the modern Web architecture*. Obrada Mehdi Jazayeri andAlexander L. Wolf Carlo Ghezzi. Association for Computing Machinery, 2010. ISBN: 978-1-58113-206-9. DOI: 10.1145/337180.337228 (cited on pages 102, 103).

[3] Dariusz Jemielniak i Aleksandra Przegalinska. *Collaborative society*. MIT Press, 2020. (citirano na stranci 185).

[4] OMG. *Business Process Model and Notation (BPMN), Version 2.0*. 2011. URL: http://www.omg.org/spec/BPMN/2.0 (cited on pages 262, 263).

[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 (citirano na stranci 261).

[6] A. Osterwalder i Y. Pigneur. *Business model generation: a handbook for visionaries, game changers, and challengers*. Sv. 1. John Wiley i Sons, 2010. (citirano na stranci 37).

Kazalo

cardinality, 49
SQL-DDL, 66

access control, 84
APEX docker, 43
aplikacijska sučelja, 274, 295
aplikacijsko programsko sučelje, API, 103
application builder, 81
application interfaces, 231, 242, 248, 257, 303, 319, 332, 352, 371, 395
application logic, 85
application report, 101
attribute, 49
AutoREST, 107

bill-of-material, 309
business view of the case, 225, 237, 245, 251, 297, 309, 325, 365, 389

calendar, 85
car rental, 389
chart, 85
column, 52
computation, 85
create application, 85
create application wizard, 82
create page wizard, 84
CRUD operacija, 103, 106

data model, 230, 240, 246, 252, 300, 313, 326, 368, 392
Data Modeler, 59
data workshop, 92, 98
database management system, 47
database system, 47
DB shema, 52
DB sloj, 48
DB sloja, 48
DB-sloj, 76
DBMS, 47
DBS, 47
DDL, 58
definicija problema, 346
definiranje problema, 264
dijagram odnosa entiteta, ERD, 49
dijagram slučaja korištenja, 265
dijagram slučajeva korištenja, 292, 347
DML, 66
domena, 49
DQL, 68
druga normalna forma, 2NF, 57

entitet, 49
entity-relationship dijagram, ERD, 48
ER, 48
ER dijagram, 48
ER-dijagram, 49
export wizard, 98

generacija relacijskog modela, 54
generacije RDB sheme, 53

how to benefit from gallery of sample/starter apps and plug-ins, 197
how to collaborate in team, 185
how to generate first draft of application, 111
how to manage forms, 157
how to manage menus, 181
how to manage packaged and multilingual applications, 205
how to manage reports, 135
how to navigate in APEX, 79
HTTP odgovor, 104
HTTP zahtjev, 103, 106
https://apex.oracle.com, 42

inovativni sustavi, 251
instaliraj Flows for APEX, 264
instancu entiteta, 52
interactive report, 85
intranet, 225
item, 85
Izvoz podataka, 98

jezik za definiranje podataka, SQL-DDL, 58
Jezik za definiranje SQL podataka, 63
jezik za strukturirane upite, SQL, 58

kako početi sa APEX-om, 35
kako pripremiti bazu podataka, 47

kako razmjenjivati podatke u APEX-u, 91

list, 85
list of values, LOV, 210
load wizard, 94
logical data model, 230, 240, 246, 252, 300, 314, 326, 368, 393
logički model, 48
logički model podataka, 48, 271, 294, 349

manipulacija podacima, 66
manipuliranje podacima, 66
model podataka, 271, 292, 349
model slučajeva korištenja, 38
model tijeka rada, 268
modul resursa, 106
multilingual, 217
multiple languages, 217

normalizacija, 57

object browser, 99
odnos, 49
On-premise, 41
opis problema, 291
Oracle Academy, 45
ORACLE APEX, 79
Oracle Cloud Infrastructure, 45
Oracle REST Data Service, ORDS, 103, 104

packaged application, 205
page, 85, 88
page creation, 84
page designer, 87
page type, 85
PL/SQL, 206
podatkovni model, 48
pohranjene funkcije, 273
poslovni pogled na slučaj, 261, 291, 345
predložak resursa, 106
preglednik objekata, 65, 67
problem definition, 225, 237, 245, 251, 297, 309, 325, 365, 389
prva normalna forma, 1NF, 57

Quick SQL, 61, 67, 119, 205
QuickSQL and Script SQL Script, 329

raspored konzultacija, 345
razine apstrakcije, 48
razmjena podataka, 91
RDB, 52
RDB shemu, 52
RDB-shema, 56
region, 85
relacijska DB shema, 52
relacijski model, 48, 52
relacijski model podataka, 271, 294, 350
relation, 52
relational data model, 230, 242, 248, 256, 300, 314, 328, 368, 393
representational State Transfer, REST, 102
REST, 102
REST bez stanja, 103
RESTful access, 104
RESTful service, 102
rukovatelja resursima, 106
rukovatelji resursima, 106

sample and starter apps, 197
sastavljača upita, 73
sekvenca, 273
Skill level, 41
slučajevi korištenja, 265, 292, 346
SQL agregacija, 72
SQL ALTER TABLE, 65
SQL COUNT, 72
SQL CREATE TABLE, 64
SQL DELETE, 67
SQL Developer Data Modeler, 48
SQL DROP TABLE, 64
SQL INSERT, 66
SQL jezik za rukovanje podacima, SQL-DML, 66
SQL JOIN, 71
SQL ORDER BY, 69, 72
SQL Script, 120
SQL SELECT, 69, 100
SQL SUM, 72
SQL TO_CHAR() , 70
SQL UPDATE, 66
SQL upitni jezik za podatke, SQL-DQL, 68
SQL WHERE klauzule, 66
SQL WHERE klauzulu, 70
SQL workshop, 65
SQL-DDL, 58, 205
SQL-DQL, 68
strani ključ, 54
struktura podataka, 48

tablica, 52
telecommunication services, 365
tip podataka, 49
treća normalna forma, 3NF, 58

upravljanje poslovnim procesima, 261
upravljanje RDBS shemom, 58
use case diagram, 226, 238, 246, 252, 313, 326, 366, 390
Use cases, 310
use cases, 226, 238, 246, 251, 298, 325, 365, 390
user authorisation, 245
user feedback, 84
uvoz podataka, 92

validation, 85
Virtual Box Appliance, 42

web application, 47, 87
web application development process, 79