

Rent a car - izrada poslovne aplikacije nad relacijskom bazom podataka u Oracle APEX-u

Mikić, Karlo

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:159836>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-17**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



Sveučilište u Rijeci – Fakultet informatike i digitalnih tehnologija

Sveučilišni preddiplomski studij informatike

Karlo Mikić

Rent a car – izrada poslovne aplikacije
nad relacijskom bazom podataka u
Oracle APEX-u

Završni rad

Mentor: prof. dr. sc. Patrizia Pošćić

Rijeka, rujan 2022.

Rijeka, 30. lipnja 2022.

Zadatak za završni rad

Pristupnik/ca: Karlo Mikić

Naziv završnog rada: Rent-a-car - poslovna aplikacija nad relacijskom bazom podataka (Oracle APEX)

Naziv završnog rada na eng. jeziku: Rent-a-car - business application with relational database (Oracle APEX)

Sadržaj zadatka: Baza podataka (BP) predstavlja kolekciju podataka, ograničenja i operacija koji reprezentiraju neke aspekte realnoga svijeta. Dakle, BP je model neke aplikacijske domene. Cilj rada je izgraditi aplikaciju za potrebe poslovanja nekog rent-a-car servisa. Aplikacija će biti izgrađena nad relacijskom BP u alatu Oracle APEX. Od studenta se očekuje da kreira model podataka i odgovarajuću poslovnu aplikaciju

Mentor

Prof. dr. sc. Patrizia Pošćić



Voditelj za završne radove

Doc. dr. sc. Miran Pobar



Zadatak preuzet: 30. lipnja 2022.



(potpis pristupnika)

Sažetak

Ovim radom opisan je proces izrade poslovne aplikacije za djelatnost *rent a car*-a, na Oracle APEX platformi za razvoj i implementaciju aplikacija. Prije početka korištenja Oracle APEX-a, na temelju informacija, podataka i procesa koje je trebalo prikupiti, izrađuje se dijagram entiteti-veze, a zatim relacijski model. Izrađuju se tablice te s time i baza podataka sa testnim podacima koji služe za testiranje, a potom i za korištenje aplikacije. Detaljno se opisuje postupak izrade aplikacije u Oracle APEX-u te nedostatke i prednosti platforma za razvoj s niskim kodom. Prikazano je korištenje Javascript-a, HTML-a, jQuery-ja, PL/SQL-a, SQL-a te izrade formi, izvješća, stranica, poveznica, ograničenja i izračuna pomoću Oracle APEX-a.

Ključne riječi

Oracle APEX, baza podataka, modeliranje podataka, SQL, Javascript, HTML, aplikacija, WEB development

Sadržaj

1. Uvod.....	1
2. Oracle APEX	2
3. Modeliranje podataka.....	3
3.1 Konceptualni EV.....	3
3.1.1 Dijagram entiteti-veze.....	4
3.2 Logički model	6
3.3 Fizički model	7
4. Izrada aplikacije u Oracle APEX-u.....	8
4.1 Zahtjev za radno okruženje	8
4.2 Izrada tablica u Oracle APEX-u	9
4.3 Kreiranje aplikacije	14
4.3.1 Prazna stranica	16
4.3.2 Pogledi („View“)	18
4.3.3 Interaktivno izvješće s dinamičnim akcijama	20
4.3.4 Forma aktivnosti vozila.....	27
4.3.5 Master Detail.....	29
5. Dodavanje novih korisnika u aplikaciju	32
6. Zaključak.....	34
7. Literatura.....	35
9. Popis priloga	37

1. Uvod

Usljed porasta informatičke pismenosti, te velike i relativno financijski pristupačne dostupnosti interneta, *cloud* servisa te *cloud* spremišta široj populaciji, dolazi do izuzetne potrebe za aplikacijama i internetskim stranicama. Paralelno s istim, a uzevši u obzir kako je programiranje kao djelatnost opsežna i zahtjevna, javlja se potrebna za jednostavnijim i bržim rješenjima za izradu aplikacija.

Kako bi se zadovoljila potreba mase za aplikacijama, razvijaju se platforme za razvoj s niskim kodom i integriranim razvojnim okruženjem (Integrated development environment, dalje u tekstu: IDE). Spomenute platforme omogućavaju izradu aplikacija sa malim predznanjem programskog jezika (i programiranja općenito). Najveća prednost IDE-a je mogućnost izrade aplikacije u vizualnom obliku. Radi toga moguće je izraditi aplikacije 20 puta brže i s 100 puta manje koda. [1]

Primjena jednostavnih aplikacija osobito je isplativa i u poslovne svrhe za manje poslovne objekte kojima one pomažu u optimizaciji redovnog poslovanja, uz niske financijske izdatke. Pritom, nije isključeno da se istim rješenjima mogu koristiti i veći poslovni subjekti. Obrti, te mali i mikro poduzetnici sve češće pribjegavaju programskim rješenjima ove vrste kako bi lakše analizirali te kontrolirali svoje poslovanje. Ovim radom prikazati će se funkcioniranje jednog takvog rješenja na primjeru *Aplikacije „Rent a car“*.

Sadržaj rada prati tijek izrade zadatka te je raspoređen u više dijelova. Najprije se detaljnije opisuju Oracle APEX, relacijske baze podataka te strukturno upitni jezik SQL. Potom slijedi izrada dijagrama entiteti-veze, relacijskog modela te vizualna izrada baze podataka. Naposljetku je opisana priprema radnog okruženja u Oracle APEX-u, izrada tablica i odabir potrebnih ograničenja.

Po završetku pripremne faze spremno je radno okruženje i moguće je započeti razvoj aplikacije u grafičko korisničkom sučelju. Postupak razvoja aplikacije je u radu dokumentiran u obliku slika te detaljno objašnjen postupak svakog koraka. Proces razvoja aplikacije obuhvaća izradu interaktivnih izvješća, formi za unos i izmjenu podataka, funkcije za brisanje, prosljeđivanje, uređivanje i automatski unos podataka pomoću SQL-a, Javascript-a i jQuery-ja.

Po završetku razvoja prikazano je korištenje aplikacije i navedene su prednosti i nedostaci razvoja aplikacije u Oracle APEX-u.

2. Oracle APEX

Oracle APEX je razvojna platforma niskog koda koja omogućuje izradu skalabilnih, sigurnih i poduzetnih aplikacija koje mogu biti dostupne cijelom svijetu. Korištenjem APEX-a, *developeri* mogu brzo razvijati konkurentne aplikacije kojima se rješavaju konkretni problemi i pružaju neposredni rezultati, bez znanja širokog broja tehnologija i programskih jezika. [1]

Platforma je besplatna za korištenje, uz uvjet da se posjeduje i Oracle baza podataka. Besplatno je moguće izrađivati manje ili testne aplikacije, dok se veće aplikacije i aplikacije sa svrhom korištenja u poduzetničke i poslovne svrhe plaćaju. [1]

APEX je platforma niskog koda, ali omogućuje korištenje Javascript-a, SQL-a, PL/SQL-a, HTML-a, CSS-a, jQuery-ja te još mnogo drugih ugrađenih opcija. Koristeći navedene programske jezike, moguće je proširiti aplikaciju do razine platforma koje nisu bazirane na niskom kodu.

Za izradu i razvoj aplikacija nije potrebna lokalna instalacija, već je APEX dostupan na internetskom pregledniku, a baza podataka na *cloud*-u. Istim se uvelike olakšavaju uvjeti rada jer je potreban samo pristup internetu. Uz navedeno, postoji i lokalna verzija aplikacije koja je također besplatna za preuzimanje i korištenje kao i na *cloud*-u.

Korisničko sučelje aplikacije je od početka izrade potpuno responzivno i automatski se prilagođava veličini ekrana. Također, vrlo je jednostavno dodati i uzorke za navigaciju, izgled stranice i ostalih komponenti, koji bi pomagali pri pristupanju mobilnim uređajima. Uređenju stranice se može jednostavno pristupiti preko korisničkog sučelja te je prilagođavati ili stvoriti novu temu koja se koristi za aplikaciju. [1]

3. Modeliranje podataka

Dobrim planiranjem te modeliranjem podataka, osiguravamo lakši razvoj i logiku baze podataka te i samu aplikaciju. Prvo je potrebno istražiti koje sve informacije, procese i podatke treba prikupiti i izraditi, nakon čega se kreira dijagram entiteti-veze, te potom relacijski model, te model u bazi podataka.

3.1 Konceptualni EV

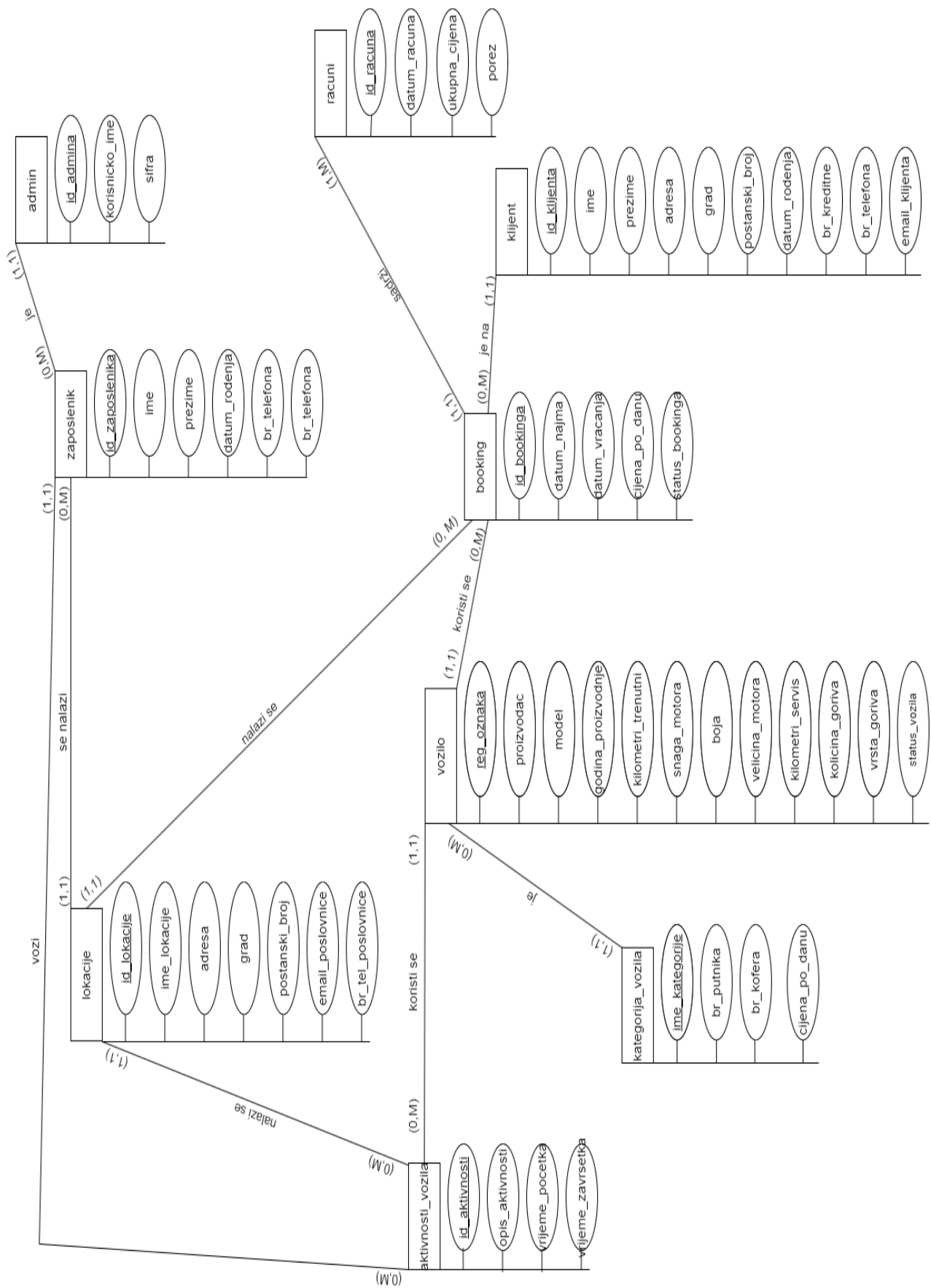
Model EV je dijagram koji ilustrira kako su entiteti u relaciji s drugim entitetima u jednom sistemu. Koriste se za dizajniranje i uočavanje grešaka relacijskih baza podataka u područjima softverskog inženjeringa, edukacije i istraživanja. Zbog jednostavnosti i vrlo lakog razumijevanja dijagrama, koristi se kao glavni dijagram između klijenta i programera. Dijagram se sastoji od pravokutnika, kvadrata, ovala i linija koje prikazuju međusobnu povezanost između entiteta, veza i njihovih atributa. [2]

Osnovni koncepti modela su:

- Entitet – objekt iz stvarnoga svijeta koji se može lagano identificirati. Primjer: baza podataka – skladište; entitet – radnik.
- Tip entiteta – kolekcija sličnih vrsta entiteta. Primjer: svi radnici iz skladišta.
- Slabi tip entiteta – entitet koji egzistencijalno ili identifikacijski ovisi od nekog drugog entiteta. Primjer: slabi tip entiteta bi bio stavka računa koja ovisi o računu. Primarni ključ je složeni ključ koji sadrži primarni ključ jakog entiteta i djelomični ključ slabog entiteta.
- Atributi – Entiteti su definirani atributima, svaki atribut ima vrijednost. Radnik može imati attribute ime i prezime.
- Veze – povezanost među entitetima.
- Agregacija – kada veza ima atribut, nastaje agregacija te se dobije veza *many-to-many*. Nema svoj primarni ključ, već sadrži složeni primarni ključ koji se sastoji od ključeva tipova entiteta između te veze.
- Kardinalnost – broj entiteta iz tipa entiteta koji se može povezati s brojem entiteta u drugom tipu entiteta. Postoje veze *one-to-one*, *one-to-many*, *many-to-many*.

3.1.1 Dijagram entiteta-veze

Dijagram sadrži 9 jakih tipova entiteta: KATEGORIJA_VOZILA, VOZILO, AKTIVNOSTI_VOZILA, BOOKING, ZAPOSLENIK, KLIJENT, ADMIN, RAČUNI, LOKACIJE. Svaki tip entiteta sadrži primarni ključ koji jedinstveno identificira taj tip entiteta. U nastavku je prikazan dijagram pojednostavljenog poduzeća *rent a car*-a koji olakšava vizualizaciju cijelog koncepta baze podataka i same aplikacije.



Slika 1 Dijagram EV rent-a-car

3.2 Logički model

Relacijski model se izrađuje na temelju već izrađenog dijagrama EV preko pravila prevođenja . Primarni ključevi su podcrtani punom crtom, a vanjski ključevi isprekidanom. Vanjski ključ je primarni ključ neke druge tablice i koristi se za povezivanje tih dviju tablica. Kada je brojnost veze između tablica *one-to-many*, vanjski ključ se nalazi u tablici na *many* strani. [3]

ADMIN (id_admina, korisnicko_ime, sifra)

KATEGORIJA_VOZILA (ime_kategorije, br_putnika, br_kofera)

KLIJENT (id_klijenta, ime, prezime, adresa, grad, postanski_broj, datum_rodenja, br_kreditne_kartice, br_telefona, email_klijenta)

LOKACIJE (id_lokacije, ime_lokacije, adresa, grad, postanski_broj, email_poslovnice, br_telefona_poslovnice)

VOZILO (reg_oznaka, proizvođač, model, godina_proizvodnje, kilometri_trenutni, snaga_motora, boja, velicina_motora, kilometri_servis. Kolicina_goriva, vrsta_goriva, status_vozila, kat_vozila)

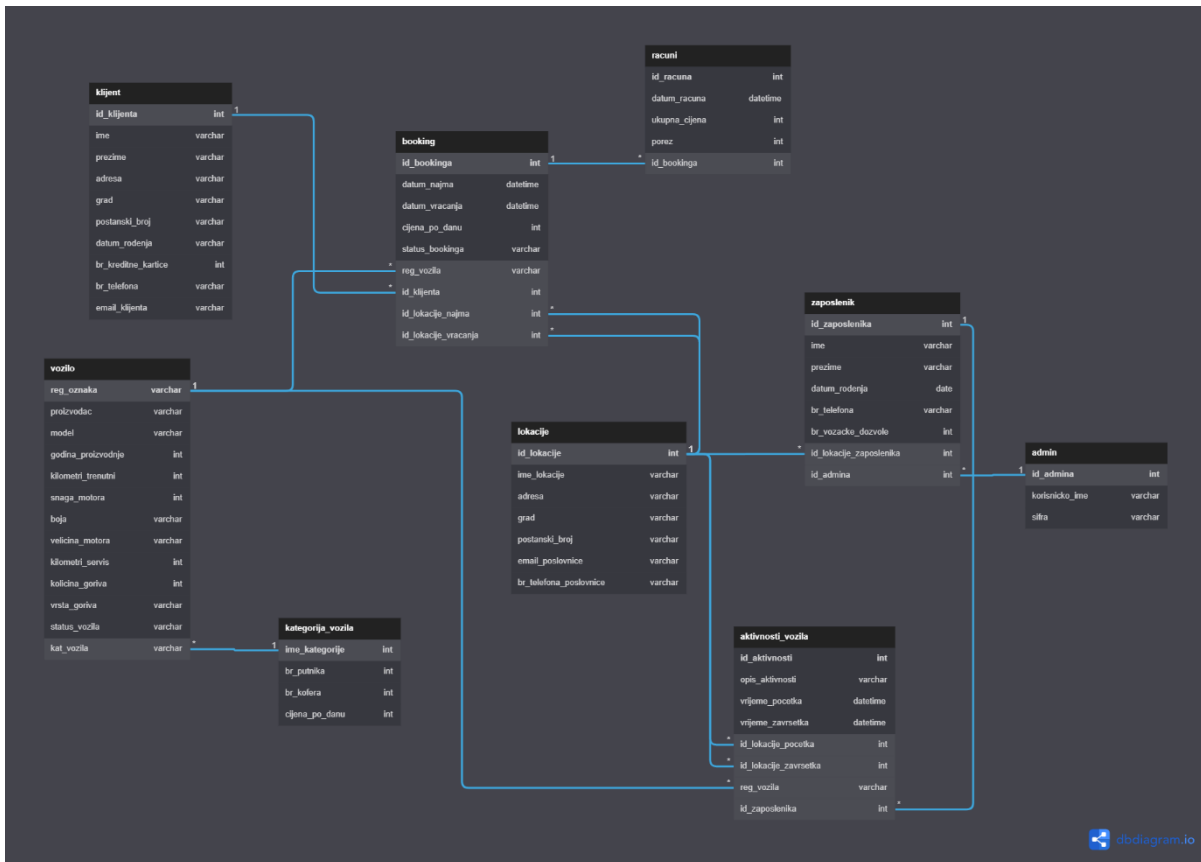
ZAPOSLENIK (id_zaposlenika, ime, prezime, datum_rodenja, br_telefona, br_vozacke_dozvole, id_lokacije_zaposlenika, id_admina)

AKTIVNOSTI_VOZILA (id_aktivnosti, opis_aktivnosti, vrijeme_pocetka, id_lokacije_pocetka, id_lokacije_zavrsetka, reg_vozila, id_zaposlenika)

BOOKING (id_bookinga, datum_najma, datum_vracanja, cijena_po_danu, status_bookinga, reg_vozila, id_klijenta, id_lokacije_najma, id_lokacije_vracanja)

RACUNI (id_racuna, datum_racuna, ukupna_cijena, porez, id_bookinga)

3.3 Fizički model



Slika 2 Vizualni model

Ovaj model (vidi: Slika 2) znatno olakšava samu izradu fizičke baze podataka. Sadrži sva imena tablica i atributa, vrstu svakog atributa, sve primarne, vanjske, složene ključeve i sve veze između tablica.

4. Izrada aplikacije u Oracle APEX-u

4.1 Zahtjev za radno okruženje

Prije početka korištenja platforme, potrebno je zatražiti radno okruženje od Oracle-a (vidi: Slika 3). Potrebno je ispuniti podatke, šifru koja će se koristiti i napisati razlog zahtjeva za radno okruženje. Zahtjev je u svrhu završnog rada za fakultet. Nakon kratkog vremena na *e-mail* dolazi obavijest o odobrenju kojim je omogućeno korištenje okruženja.

Request a Workspace

Identification

First Name
Karlo
First Name is required.

Last Name
Mikić
Last Name is required.

Email
karlo.mikic@student.uniri.hr
A link to activate your workspace will be sent to this email address.
Email is required.

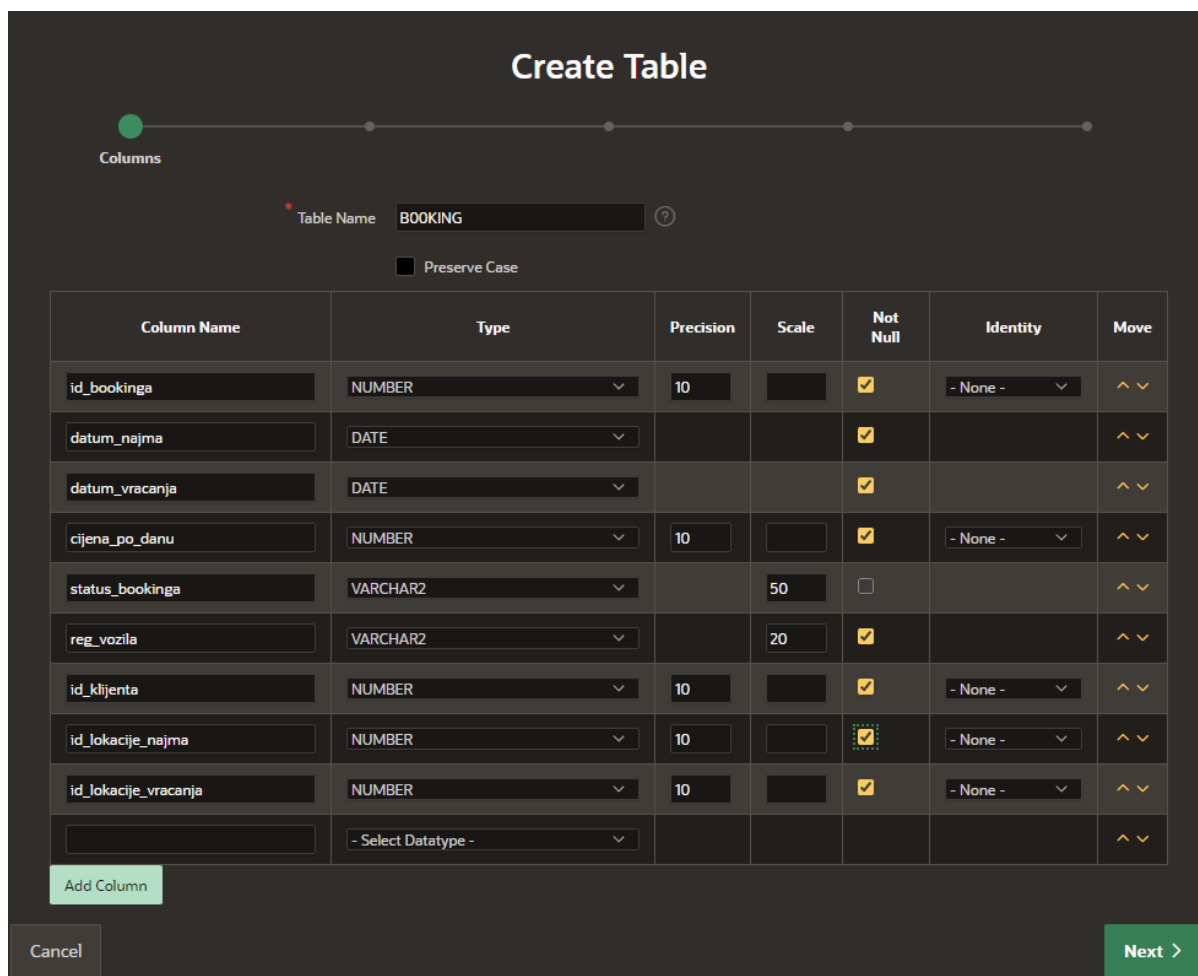
Workspace
ZAVRSNI
This workspace name uniquely identifies your development environment and is used when signing in.
Workspace is required.

Country
Croatia
Country is required.

Slika 3 Zahtjev za Radno okruženje (engl. Workspace)

4.2 Izrada tablica u Oracle APEX-u

Izrada tablica je moguća i u Oracle Cloud-u, ali postoji i mogućnost izrade u samoj platformi koja se koristi unutar „*SQL workshop*“. Oracle APEX nudi i SQL komande i skripte te pomoću njih se isto mogu kreirati potrebne tablice i ograničenja. U primjeru će se koristiti grafičko sučelje platforme (vidi: Slika 4).



The screenshot shows the 'Create Table' wizard in Oracle APEX. The table name is 'BOOKING'. The 'Columns' step is active. The table has 10 columns with various data types and constraints.

Column Name	Type	Precision	Scale	Not Null	Identity	Move
id_bookinga	NUMBER	10		<input checked="" type="checkbox"/>	- None -	^ v
datum_najma	DATE			<input checked="" type="checkbox"/>		^ v
datum_vracanja	DATE			<input checked="" type="checkbox"/>		^ v
cijena_po_danu	NUMBER	10		<input checked="" type="checkbox"/>	- None -	^ v
status_bookinga	VARCHAR2		50	<input type="checkbox"/>		^ v
reg_vozila	VARCHAR2		20	<input checked="" type="checkbox"/>		^ v
id_klijenta	NUMBER	10		<input checked="" type="checkbox"/>	- None -	^ v
id_lokacije_najma	NUMBER	10		<input checked="" type="checkbox"/>	- None -	^ v
id_lokacije_vracanja	NUMBER	10		<input checked="" type="checkbox"/>	- None -	^ v
	- Select Datatype -					^ v

Buttons: Add Column, Cancel, Next >

Slika 4 Izrada tablice *BOOKING*

Prilikom kreiranja tablica vizualni model uvelike olakšava unos potrebnih podataka i vrsta podataka. Određuje se vrsta stupaca, veličina, a za brojeve i preciznost decimalnih brojeva te obavezan upis podataka „Not Null“ ako je potrebno (vidi: Slika 4).

↑ Create Table

Create Table

✓ — ● — ● — ● — ●

Primary Key

Table name: **BOOKING** ?

Primary Key: No Primary Key ?
 Populated from a new sequence ?
 Populated from an existing sequence ?
 Not populated ?
 Populated by Identity column ?

* Primary Key Constraint Name: ?

* Primary Key: ?

Composite Primary Key: ?

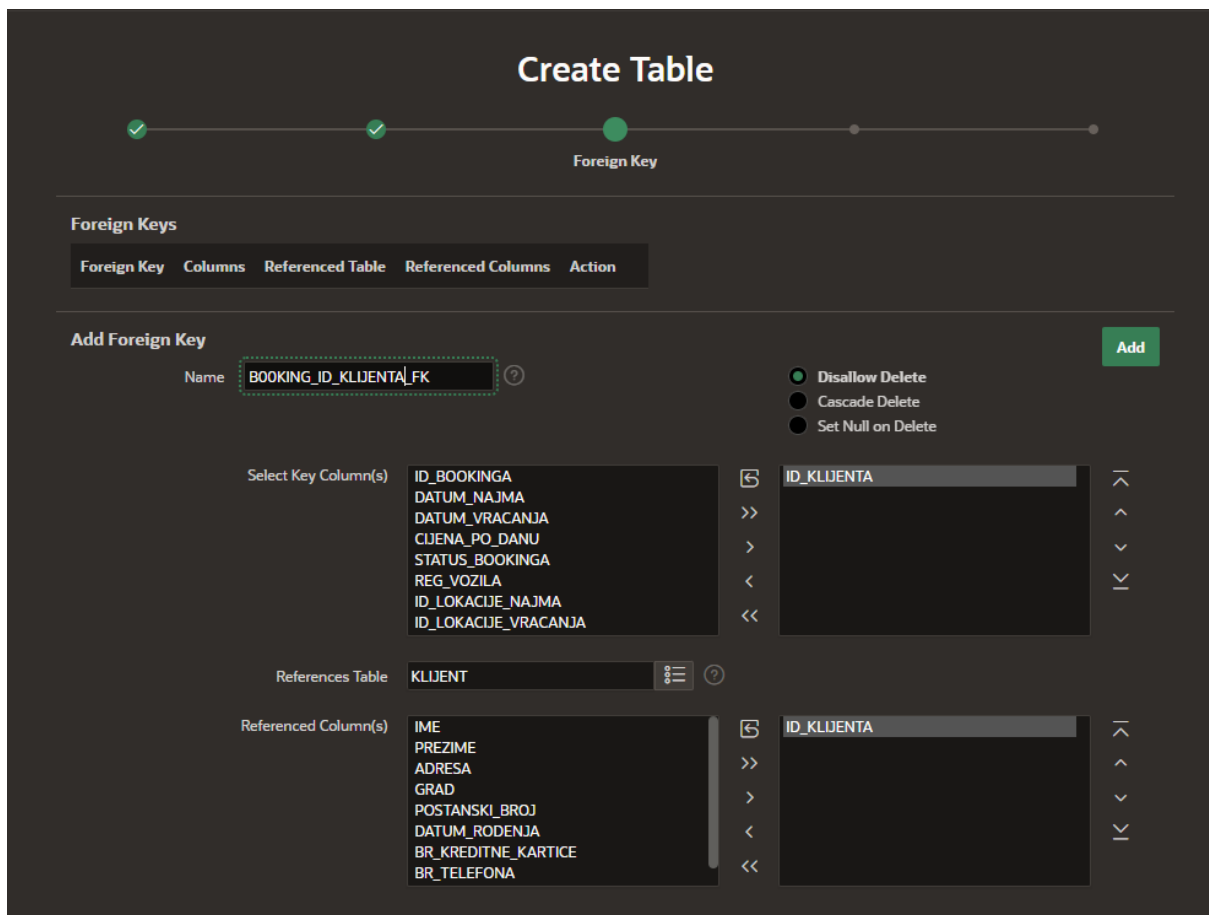
Primary Key
A primary key allows each row in a table to be uniquely identified.

If you select to populate your primary key from a new sequence, you will be prompted to enter the new sequence's name. If you select to populate your primary key from an existing sequence, you will be prompted to select the sequence. Both these methods result in the generation of a trigger against your table. You can also select to not populate your primary. This is the only method that allows you to define a composite primary key made up of more than two columns.

< Cancel
Next >

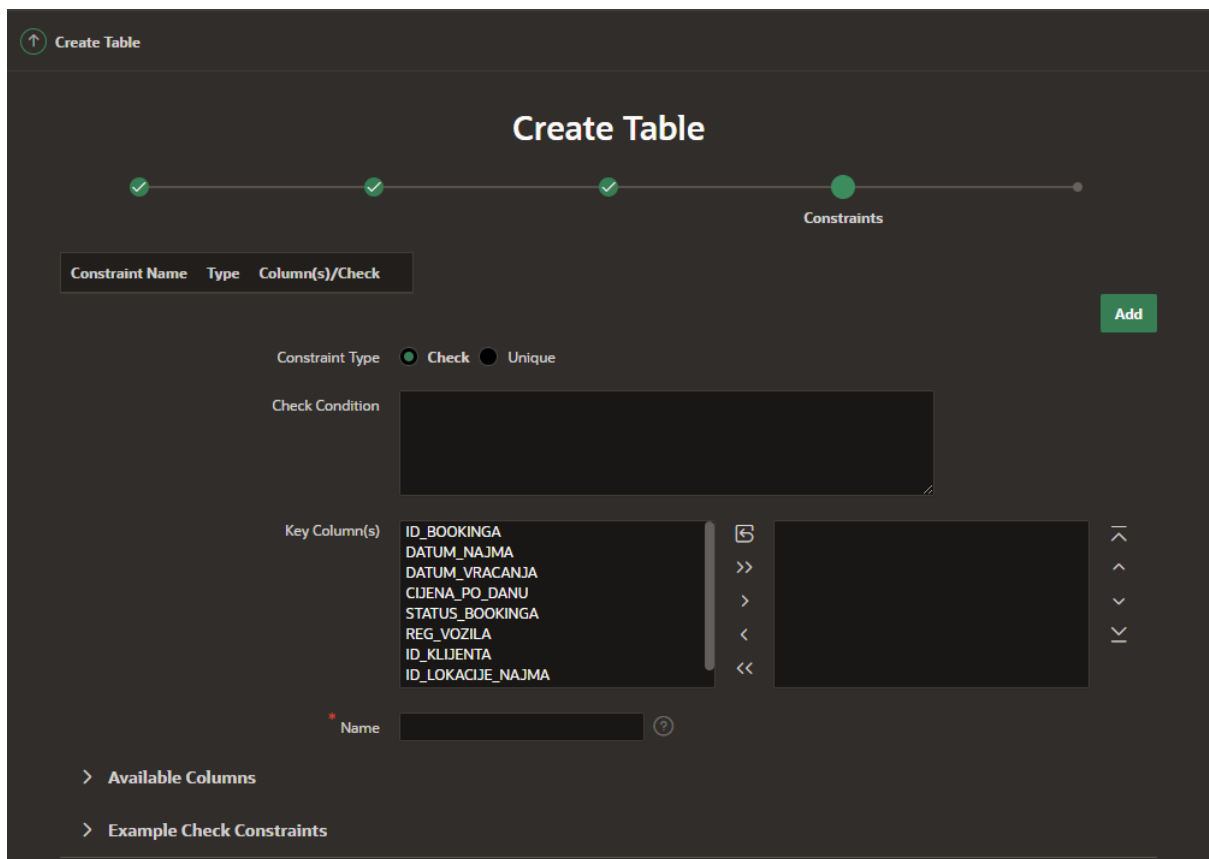
Slika 5 Postavljanje primarnog ključa

Idući korak je odabir primarnog ključa i način njegova upisivanja. Za primarne ključeve, koji se sami generiraju pomoću sekvence ili nasumično izabranih brojeva ili su složeni ključevi, odabire se „*Populated by Identity column*“, a ako je potrebno ručno upisivanje onda se odabire „*Not populated*“. (vidi: Slika 5).



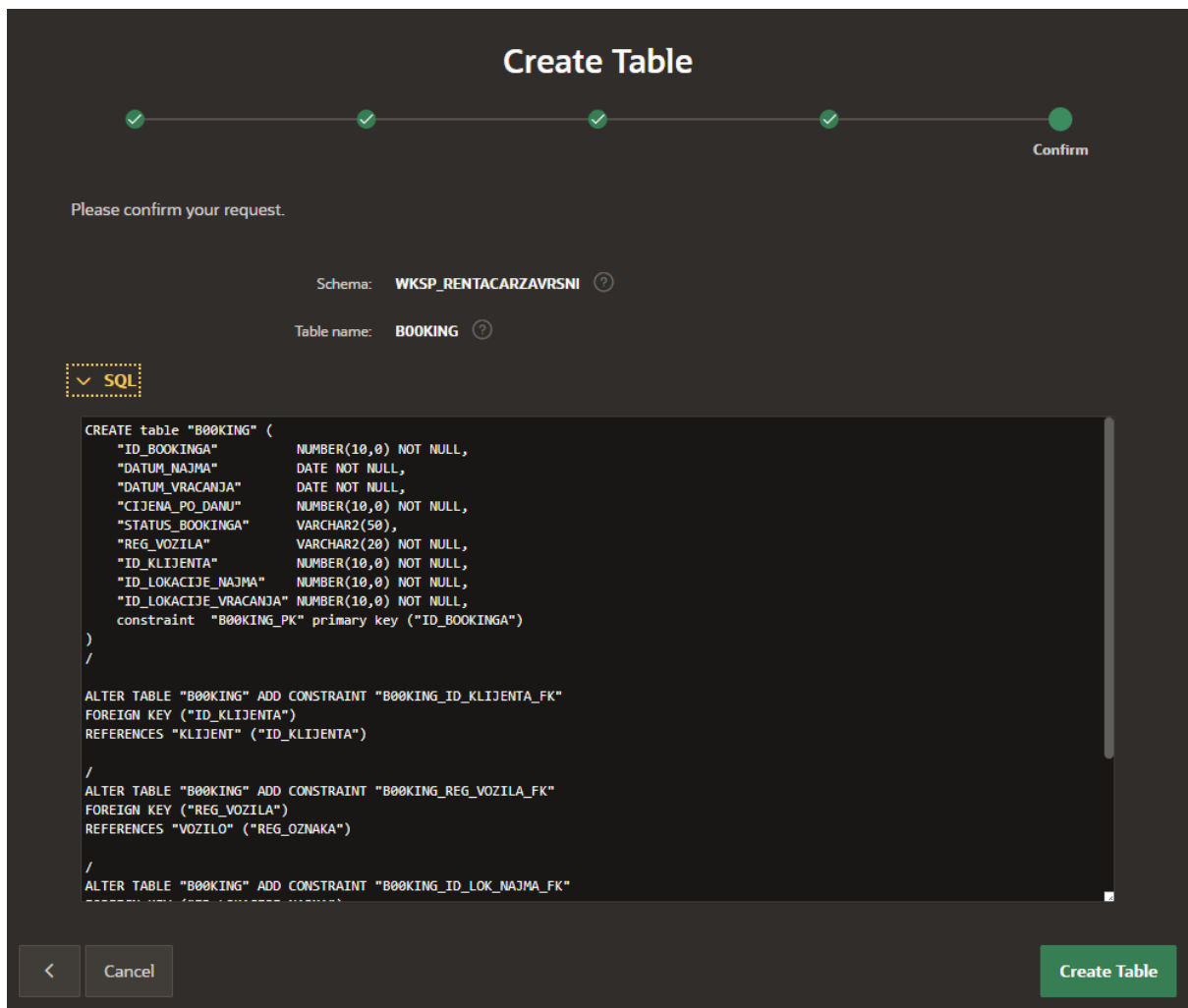
Slika 6 Dodavanje vanjskih ključeva

Dodavanjem vanjskih ključeva stvaramo relacije između tih tablica. Moraju sadržavati iste vrste polja radi sprječavanja grešaka pri upisivanju u bazu podataka. Dodaje se ime vanjskom ključu jer će ih biti više u toj tablici. Odabire se stupac koji će biti vanjski ključ, a zatim tablica u kojoj je traženi primarni ključ te i sam primarni ključ. U tablici (vidi: Slika 6) vanjske ključeve dodajemo stupcima ID_KLIJENTA, REG_VOZILA, ID_LOKACIJE_NAJMA, ID_LOKACIJE_VRACANJA. Potrebno je odabrati i vrstu brisanja. Zbog potrebe BOOKING tablice odabire se „Disallow Delete“ koji onemogućuje brisanje entiteta iz referentne tablice kada postoje zavisni redci u BOOKING tablici.



Slika 7 Dodavanje ograničenja

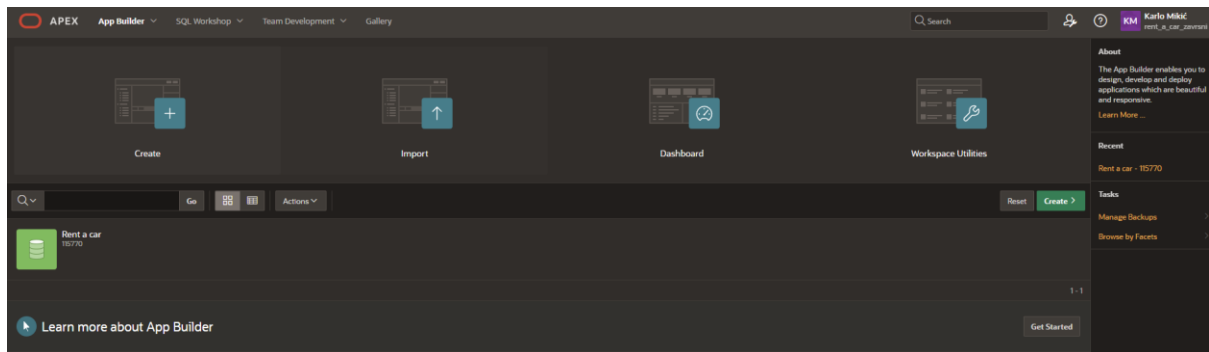
Dodavanjem ograničenja (vidi: Slika 7), mogu se ograničiti ulazi podataka u tablicu. To može biti provjera je li osoba starija od 18 godina ili samo provjeravati ako je jedinstven podatak. U ovoj tablici nisu potrebna ograničenja, već će se ograničavati preko aplikacije. Primarni ključ, vanjski ključevi i provjere „Not Null“ su već dodani.



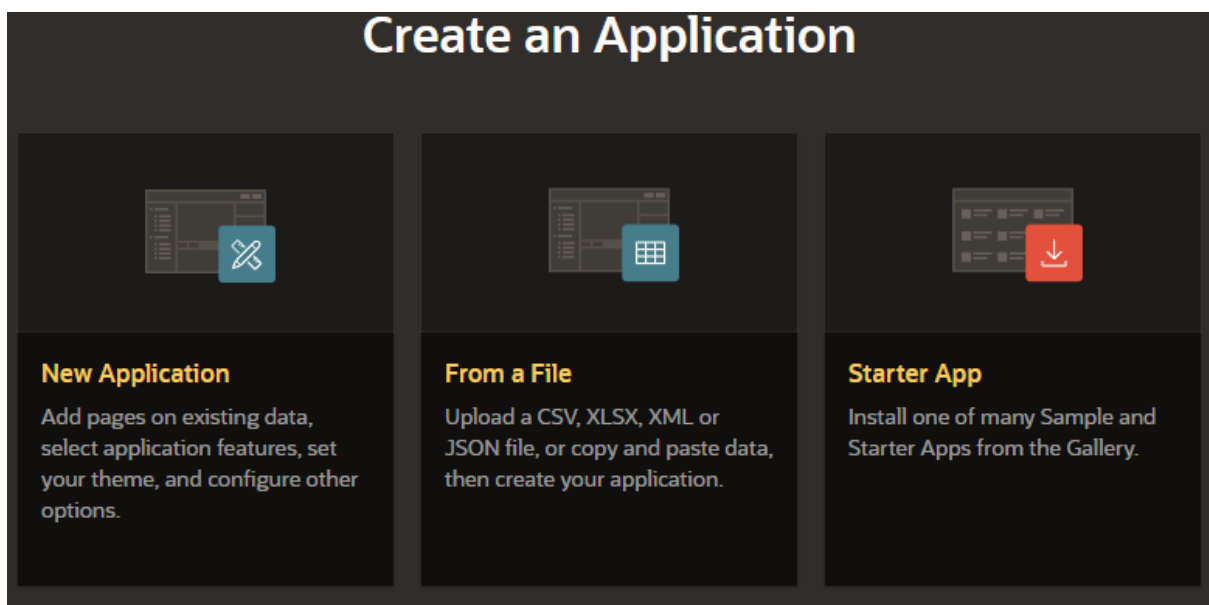
Slika 8 Potvrda izrade tablice

Na kraju se generira SQL te se može provjeriti prije potvrđivanja je li sve ispravno uneseno i deklarirano (vidi: Slika 8).

4.3 Kreiranje aplikacije

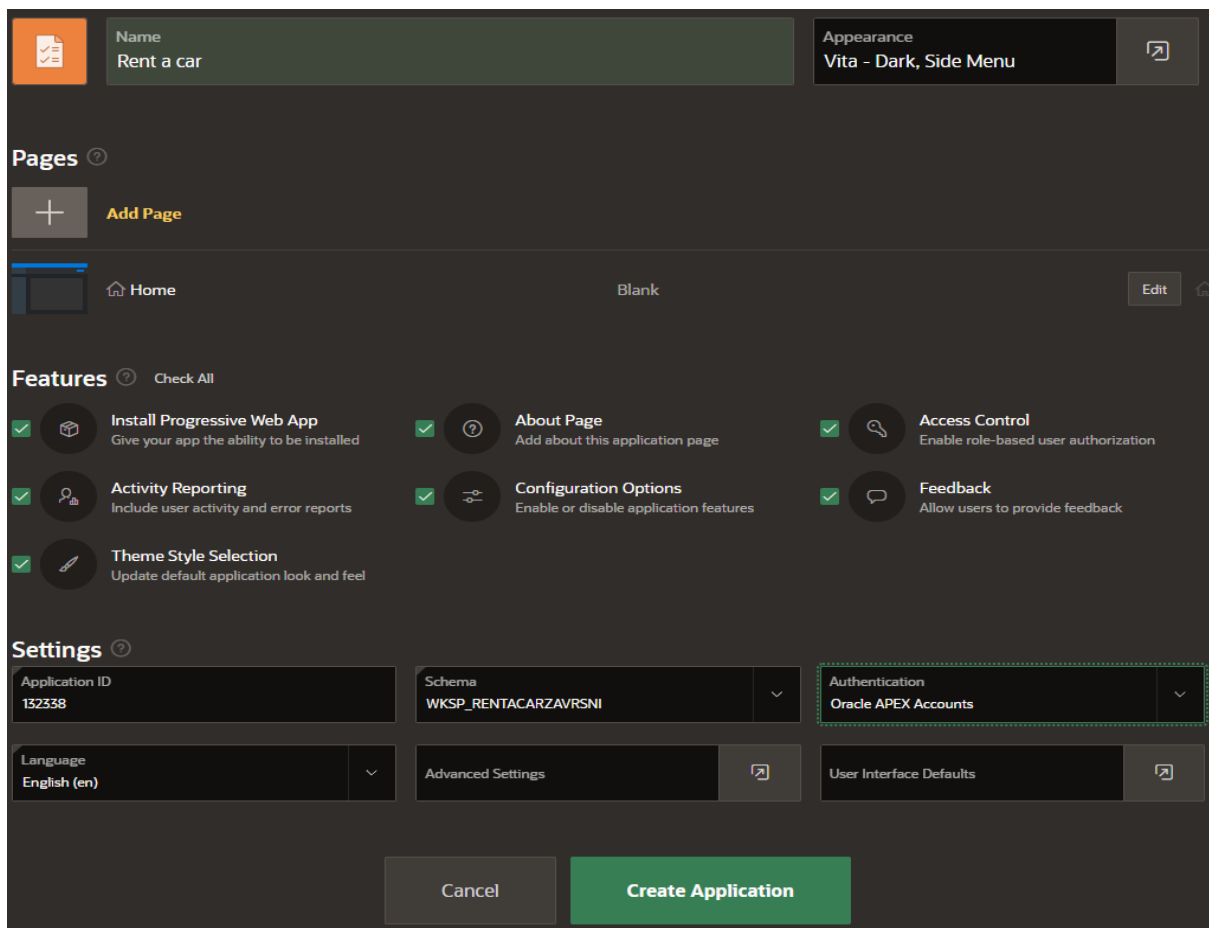


Slika 9 Grafičko sučelje "App Builder"



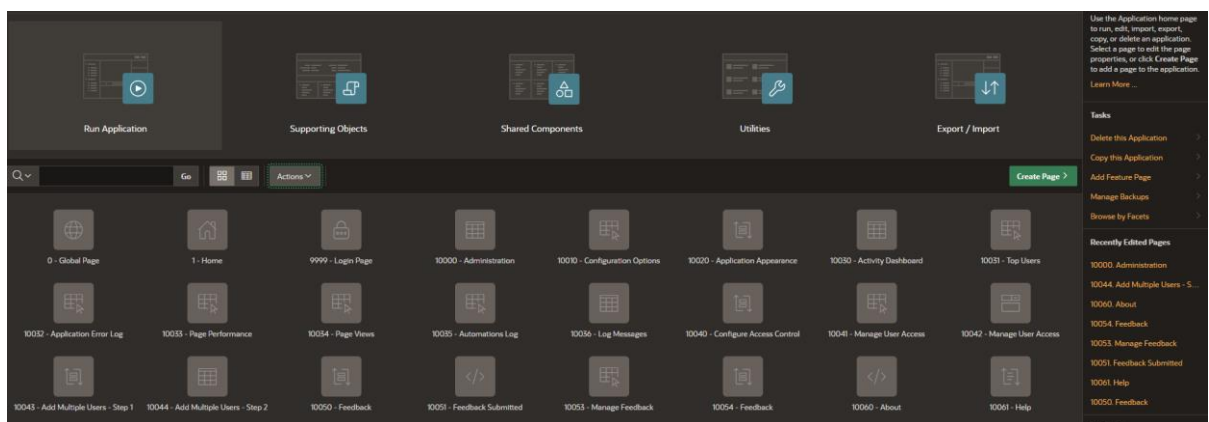
Slika 10 Opcije kreiranja aplikacije

Odabirom „Create“ (vidi: Slika 9) ulazi se u novi izbornik (vidi: Slika 10) koji nudi opcije izrade aplikacije bez postojećih podataka, sa postojećim podacima te neke već gotove testne aplikacije koje nudi Oracle APEX.



Slika 11 Izbornik za kreiranje nove aplikacije

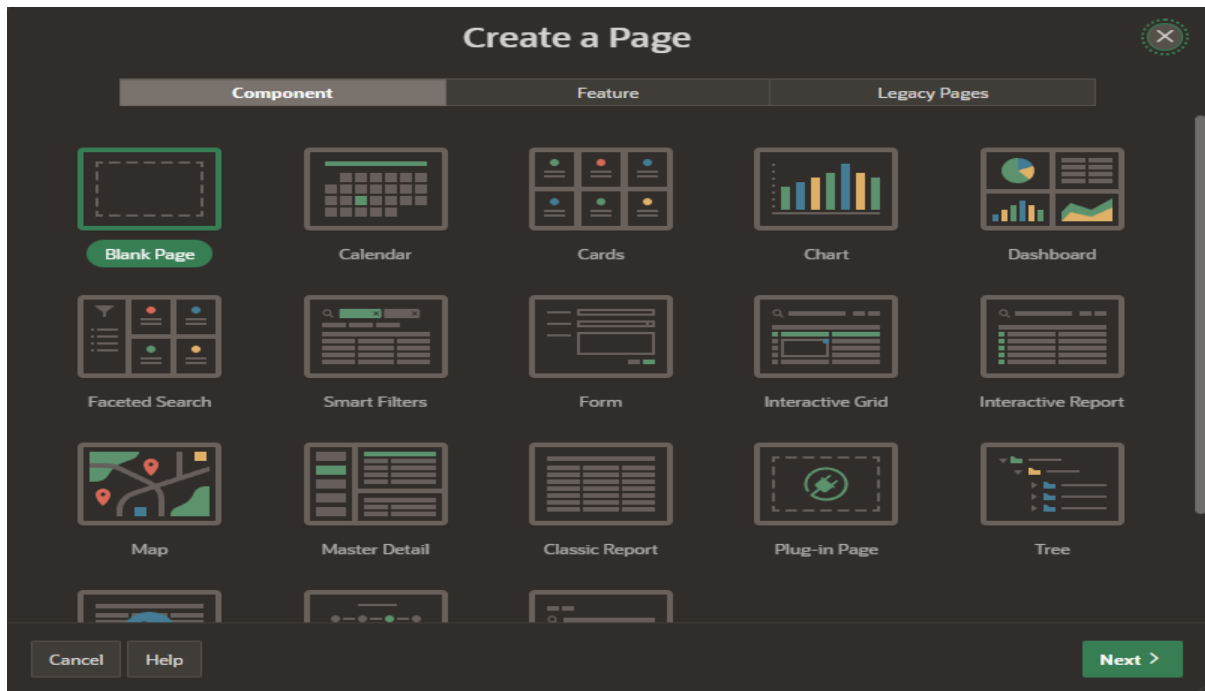
Odabirom na „New Application“ otvara se izbornik (vidi: Slika 11) gdje se određuje ime aplikacije, izgled same aplikacije pomoću već definiranih tema, raznih korisnih Oracle APEX usluga za praćenje korisnika i stranice te u kojem radnom okruženju i jeziku će se nalaziti aplikacija. Teme za izgled se mogu dodavati i nisu ograničene na ponuđenim Oracle temama.



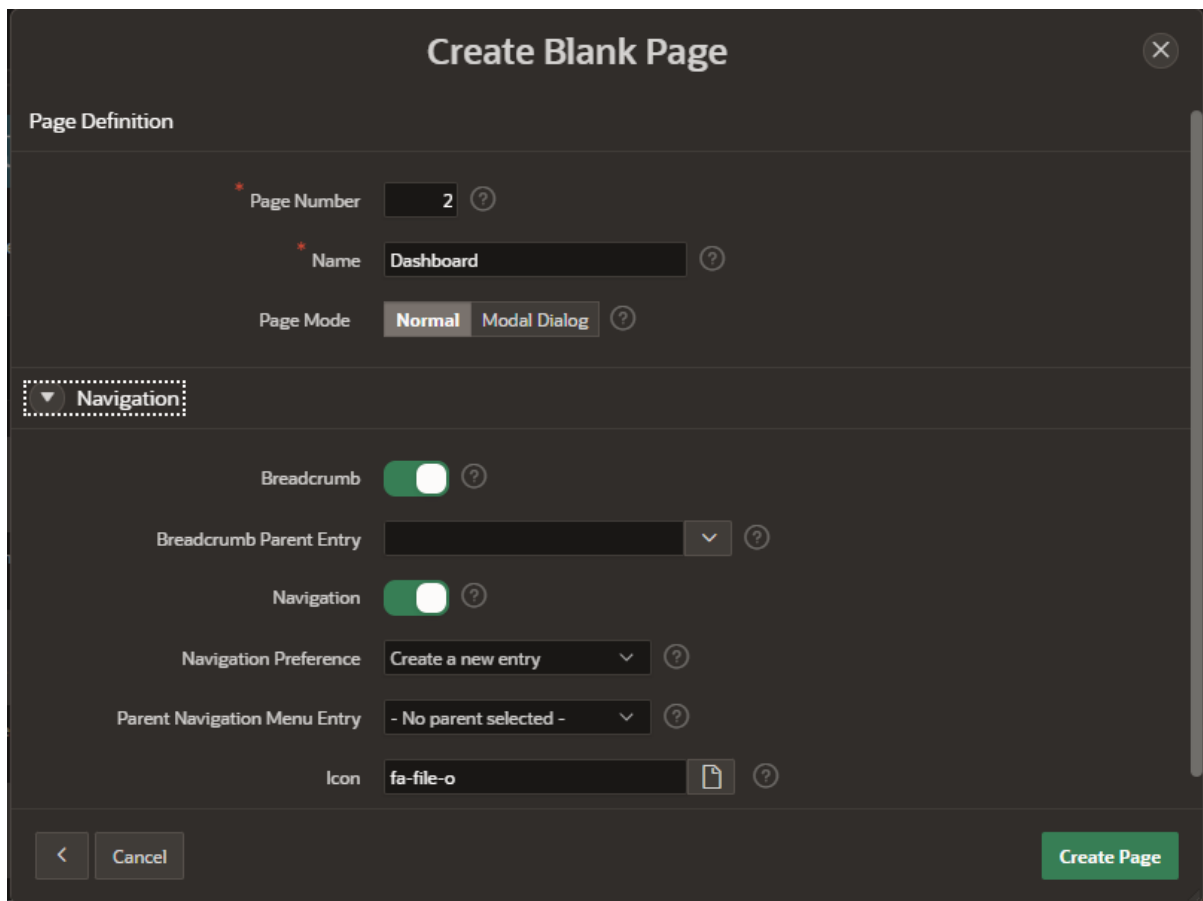
Slika 12 Izgled nove aplikacije u „App Builder“

Sa „*Run Application*“ pokreće se aplikaciju, ali aplikacija ne sadrži nikakve podatke osim nekih Oracle APEX usluga (vidi: Slika 12). Na istom izborniku moguće je izraditi nove stranice (vidi: Slika 13) u samoj aplikaciji.

4.3.1 Prazna stranica

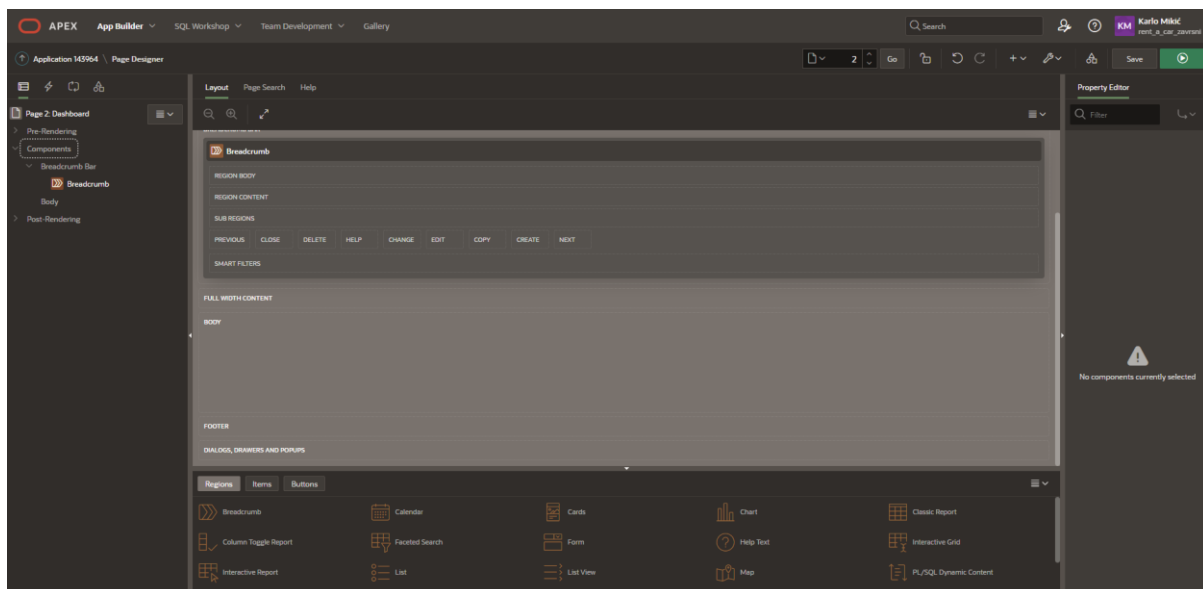


Slika 13 Popis ponuđenih stranica za izradu



Slika 14 Izrada prazne stranice

Prazna stranica ne zahtjeva unos podataka ni tablica, ali možemo odabrati broj stranice, ime, hoće li se nalaziti u navigaciji i koju će ikonu sadržavati ako se nalazi (vidi: Slika 14).



Slika 15 Izgled prazne stranice u „App Builder“

S lijeve strane, nalazi se ime stranice te njene komponente (vidi: Slika 15). Za sad je sve prazno, osim „*Breadcrumb*“ navigacije. Moguće je dodavanje novih regija i polja. Regije su skup polja i gumba koje se mogu automatski generirati pri izradi nove stranice te odmah sadrži sve što je potrebno za neku tablicu ili pogled. U ovom slučaju regije i gumbi će se dodavati ručno, a kasnije preko opcija koje Oracle APEX ponuđuje (vidi: Slika 13).

4.3.2 Pogledi („*View*“)

Radi jednostavnosti aplikacije izrađuju se pogledi u „SQL Workshop“ sa SQL *query-ima*. Moguće je u aplikaciji posebno pisati *query*, ali radi lakšeg korištenja napraviti će se pogledi koji će biti spremljeni odvojeno. Pogledi će se koristiti za izvješća u aplikaciji.

U svrhu aplikacije, potrebna su tri pogleda:

Prvi pogled predstavlja sve aktivnosti koje zaposlenik otvori, te se sadrži od odabranih atributa iz 4 tablica. S ovim pogledom će se moći prikazati opis aktivnosti, vrijeme početka i završetka, lokaciju, registraciju i kategoriju vozila, te ime i prezime klijenta kao jedna kolumna.

```
CREATE OR REPLACE FORCE EDITIONABLE VIEW "AKTIVNOSTI_SVE"  
("ID_AKTIVNOSTI", "OPIS_AKTIVNOSTI", "VRIJEME_POCETKA",  
"VRIJEME_ZAVRSETKA", "REG_VOZILA", "KAT_VOZILA", "POCETNA",  
"ZAVRSNA", "VOZAC") AS
```

```
SELECT b.id_aktivnosti, b.opis_aktivnosti, b.vrijeme_pocetka, b.vrijeme_zavrsetka,  
b.reg_vozila, v.kat_vozila, l.ime_lokacije as Pocetna, l2.ime_lokacije as Zavrсна, (z.ime || ' ' ||  
z.prezime) as vozac FROM AKTIVNOSTI_VOZILA b
```

```
JOIN VOZILO v ON b.reg_vozila=v.reg_oznaka
```

```
JOIN LOKACIJA 1 ON b.ID_LOKACIJE_POCETKA=l.id_lokacije
```

```
JOIN LOKACIJA 12 ON b.ID_LOKACIJE_ZAVRSETKA=l2.id_lokacije
```

```
JOIN ZAPOSLENIK z on b.ID_ZAPOSLENIKA=z.ID_ZAPOSLENIKA
```

Drugi pogled predstavlja predviđena vraćanja u kojoj zaposlenik može provjeriti kada se klijenti vraćaju i u koju poslovnici. Pogled sadrži i podatke o vozilu i cijenu po danu.

```
CREATE OR REPLACE FORCE EDITIONABLE VIEW "PREDVIDENA_VRACANJA"  
("ID_BOOKINGA", "DATUM_NAJMA", "DATUM_VRACANJA",
```

```

"CIJENA_PO_DANU", "REG_VOZILA", "KAT_VOZILA", "POCETNA", "ZAVRSNA",
"VOZAC", "STATUS_BOOKINGA") AS
SELECT b.id_bookinga, b.datum_najma, b.datum_vracanja, b.cijena_po_danu, b.reg_vozila,
v.kat_vozila, l.ime_lokacije as Pocetna, l2.ime_lokacije as Zavrсна, (k.ime || ' ' || k.prezime)
as vozac, b.STATUS_BOOKINGA FROM booking b

JOIN VOZILO v ON b.reg_vozila=v.reg_oznaka
JOIN LOKACIJA l ON b.ID_LOKACIJE_NAJMA=l.id_lokacije
JOIN LOKACIJA l2 ON b.ID_LOKACIJE_VRACANJA=l2.id_lokacije
JOIN KLIJENT k on b.ID_KLIJENTA=k.ID_KLIJENTA

```

Treći pogled predstavlja izvješće računa. S ovim pogledom zaposlenik može provjeriti kome je račun izdan, kojeg datuma i iznos računa u kunama.

```

CREATE OR REPLACE FORCE EDITIONABLE VIEW "RACUNI_REPORT"
("DATUM_NAJMA", "DATUM_VRACANJA", "CIJENA_PO_DANU", "ID_RACUNA",
"DATUM_RACUNA", "UKUPNA_CIJENA", "POREZ", "ID_BOOKINGA", "KLIJENT")
AS
SELECT b.datum_najma, b.datum_vracanja, b.cijena_po_danu,
r."ID_RACUNA",r."DATUM_RACUNA",r."UKUPNA_CIJENA",r."POREZ",r."ID_BOOK
INGA", (k.ime || ' ' || k.prezime) as klijent FROM booking b

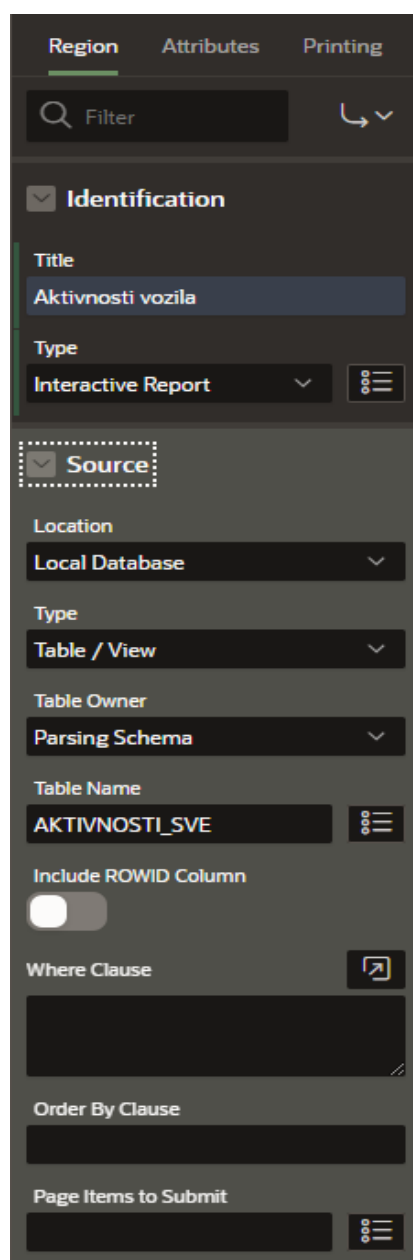
JOIN racuni r ON r.id_bookinga=b.id_bookinga
JOIN klijent k ON b.id_klijenta=k.id_klijenta

```


4.3.3 Interaktivno izvješće s dinamičnim akcijama

Velika prednost interaktivnih izvješća je mala ili nikakva potreba za pisanjem kompliciranih i velikih SQL *query*-ja. Omogućuje korisnicima i administratorima da izrađuju svoja izvješća i manipuliraju podacima bez pisanja *query*-ja. Korisnik može spremati više izvješća odjednom te ih koristiti u bilo kojem trenutku.

Za dodati interaktivno izvješće u praznu stranicu potrebno je odabrati regiju *Interactive Report* (vidi: Slika 15) i postaviti je u komponente aplikacije, u ovom slučaju u *body*.

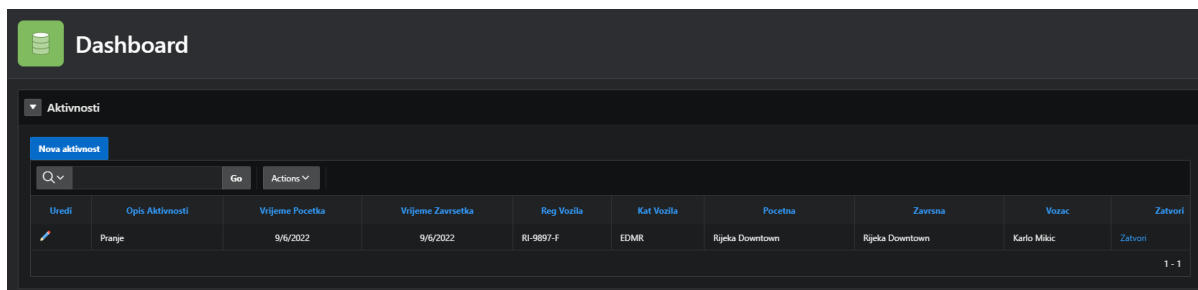


The screenshot displays a configuration panel for an interactive report. At the top, there are tabs for 'Region', 'Attributes', and 'Printing'. Below these is a search bar labeled 'Filter' and a refresh icon. The main configuration is divided into two sections: 'Identification' and 'Source'. The 'Identification' section includes a checked checkbox, a title field containing 'Aktivnosti vozila', and a type dropdown menu set to 'Interactive Report'. The 'Source' section includes a checked checkbox, a location dropdown set to 'Local Database', a type dropdown set to 'Table / View', a table owner dropdown set to 'Parsing Schema', and a table name field containing 'AKTIVNOSTI_SVE'. Below these are a toggle switch for 'Include ROWID Column' (currently off), a 'Where Clause' field with a refresh icon, an 'Order By Clause' field, and a 'Page Items to Submit' field.

Slika 16 Spajanje izvješća s pogledom

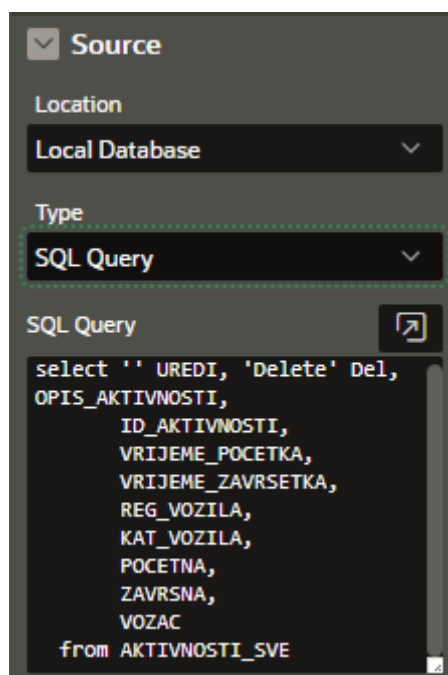
U postavkama regije (vidi: Slika 16) potrebno je odabrati tablicu ili pogled koji je potreban. Potom će se prikazati svi stupci tog pogleda koji su prethodno kreirani.

Nakon koje je potrebno dodati gumbе za dinamično uređivanje pojedinog retka, dodavanje nove aktivnosti te brisanje aktivnosti (vidi: Slika 17).



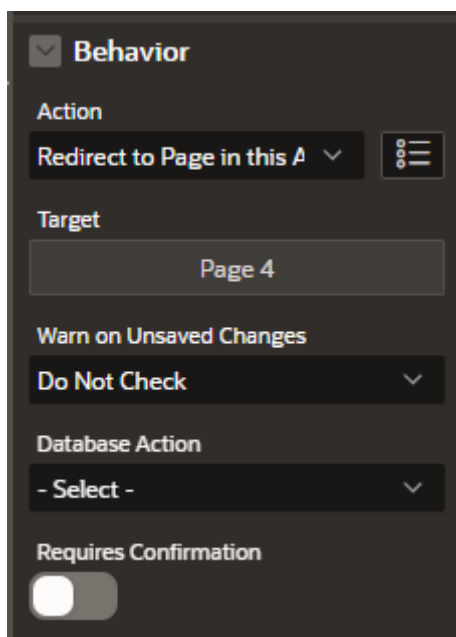
The screenshot shows a dashboard titled "Dashboard" with a sub-section "Aktivnosti". There is a search bar and a "Go" button. Below the search bar is a table with columns: "Uredi", "Opis Aktivnosti", "Vrijeme Pocetka", "Vrijeme Zavrsetka", "Reg Vozila", "Kat Vozila", "Pocetna", "Zavrsna", "Vozac", and "Zatvori". The table contains one row of data: "Pranje", "9/6/2022", "9/6/2022", "RI-9897-F", "EDMR", "Rijeka Downtown", "Rijeka Downtown", "Karlo Mikic", and "Zatvori". There are also "Nova aktivnost" and "Actions" buttons.

Slika 17 Izvješće s dodanim gumbima



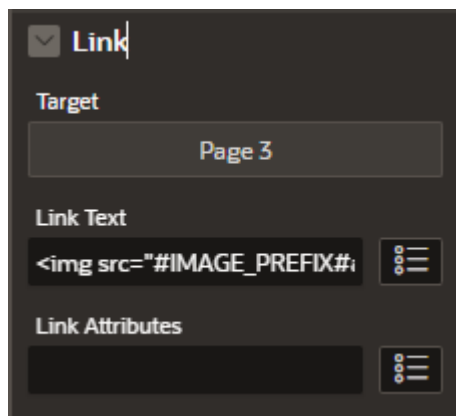
Slika 18 SQL Query izvješća

Na slici 18 prikazano je kako se može iskoristiti prednost pogleda i umjesto odabrane tablice, označuje se „SQL Query“ i ostaje ista tablica samo u SQL Query formatu koji se može uređivati. Dodaju se 2 nova pomoćna stupca koji će biti linkovi i s njima će se uređivati i brisati aktivnosti (vidi: Slika 18). Gumb za novu aktivnost se nalazi pod „Items“ (vidi: Slika 14) te ga se smješta na pripadajuće mjesto unutar regije.



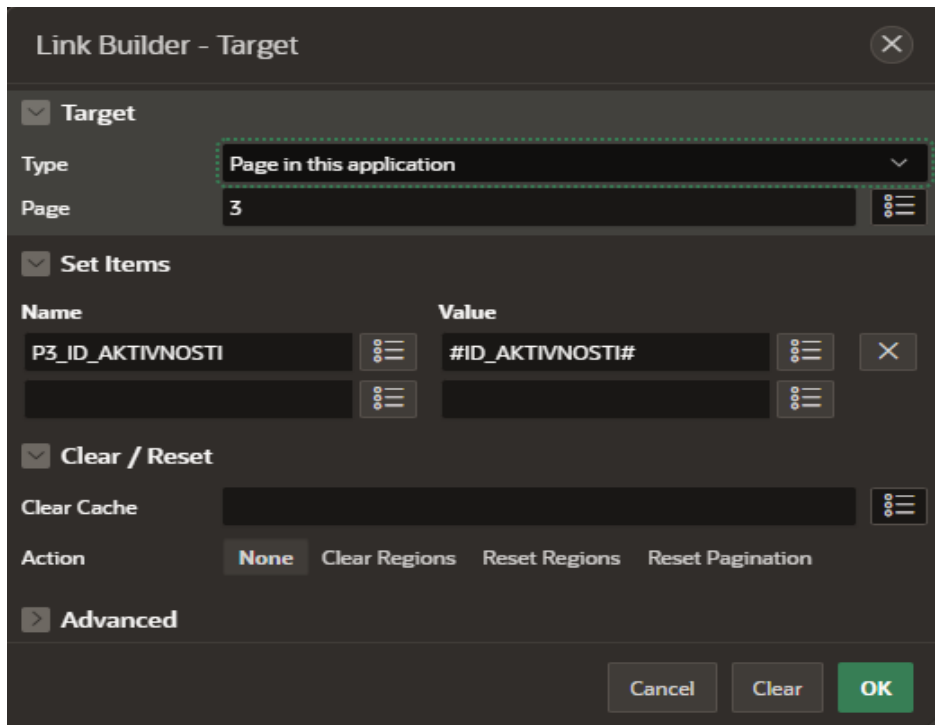
Slika 19 Ponašanje gumba Nova aktivnost

Kada se klikne na gumb Nova aktivnost (vidi: Slika 17), otvara se forma za unos nove aktivnosti koja je zaseban prozor. To se postiže preusmjerenjem gumba u njegovim postavkama (vidi: Slika 19).



Slika 20 Dodavanje linka na gumb

Za dodavanje uređivanja svakog retka posebno, potrebno je pomoćnim stupcima promijeniti vrstu u link (vidi: Slika 20).



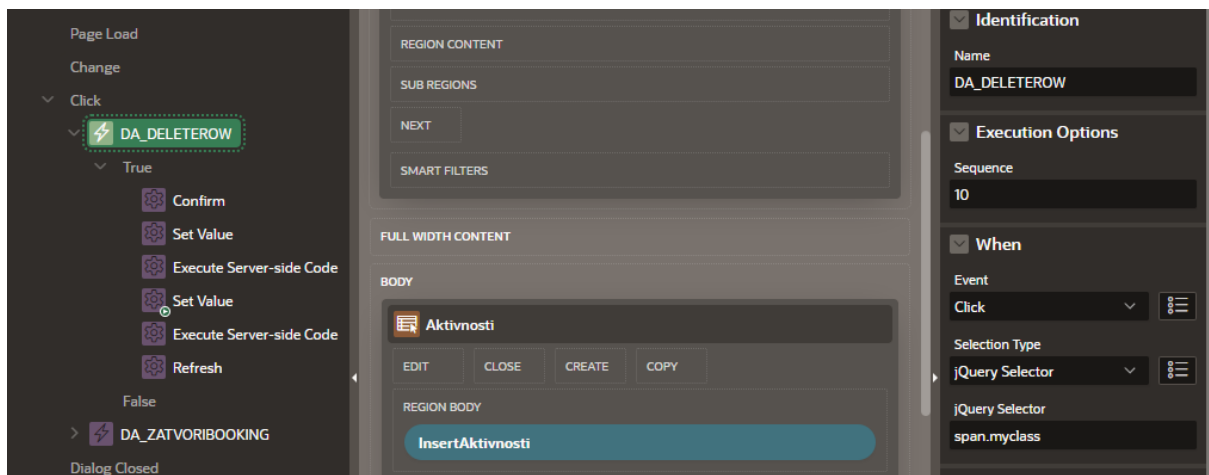
Slika 21 Odabir stranice i prosljeđivanje vrijednosti

Zatim se prosljeđuje ID_AKTIVNOSTI iz tog retka u formu, čime je forma popunjena s vrijednostima cijelog retka te mogu se uređivati i spremiti link (vidi: Slika 21).

Za brisanje svake aktivnosti potrebno je drugi pomoćni stupac također pretvoriti u link (vidi: Slika 20), ali ovaj put link će preusmjeravati u „javascript:void(null);“ i tekst linka će se sastojati od:

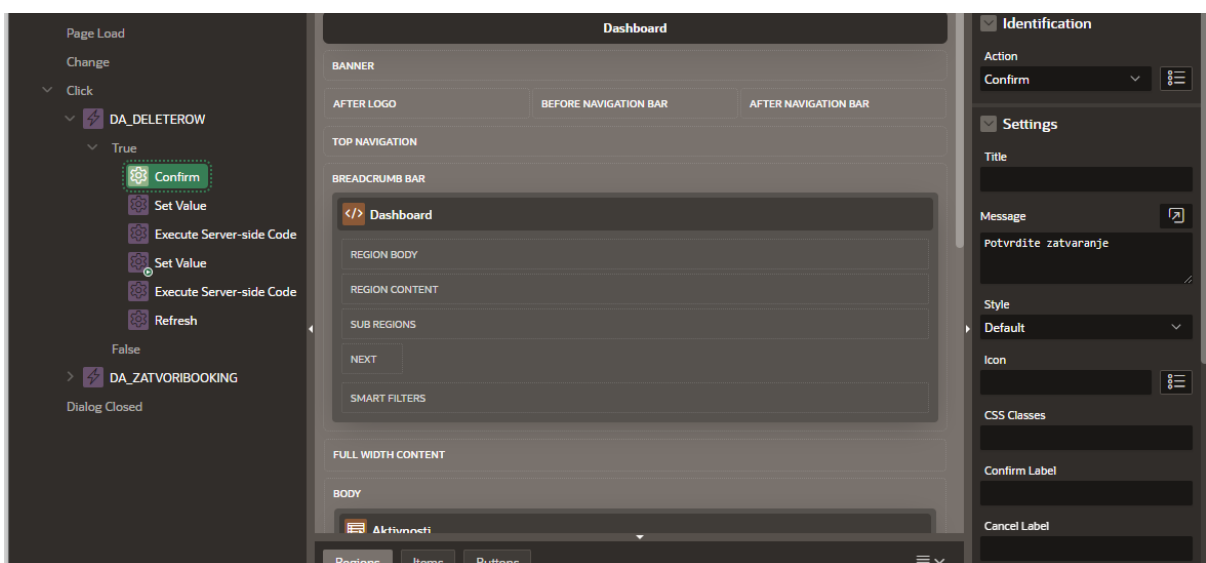
```
<span class="myclass" data-id='#ID_AKTIVNOSTI#' dana
reg='#REG_VOZILA#>Zatvori</span>
```

Taj link nam prosljeđuje ID_AKTIVNOSTI i REG_VOZILA kada se odabere „jQuery Selector“ (vidi: Slika 22).

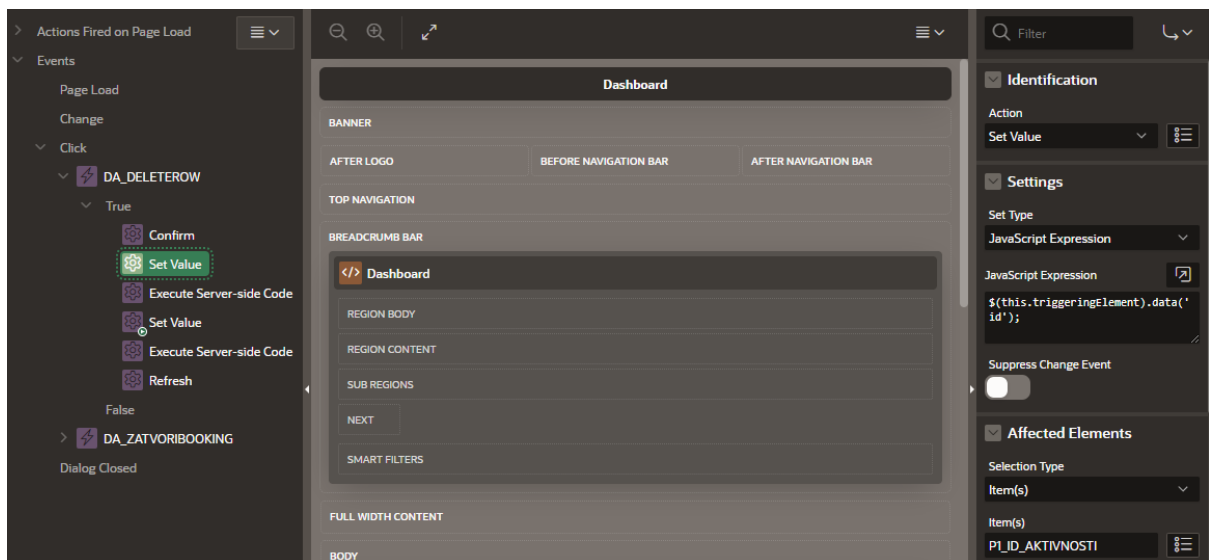


Slika 22 „jQuery Selector“ sa dinamičnim eventom

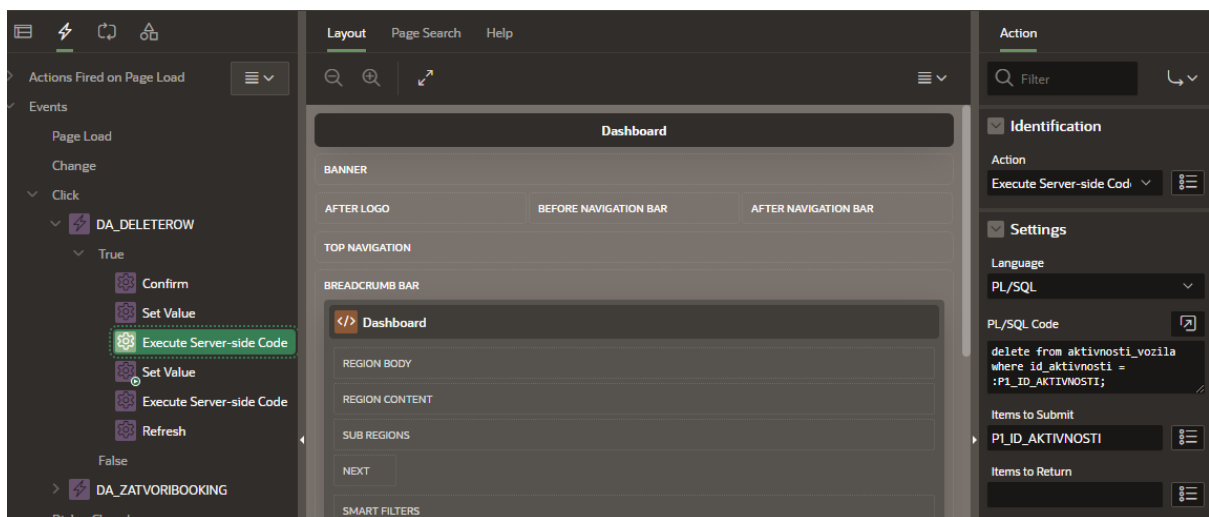
Dinamični event je alat Oracle APEX-a koji provjerava neke događaje na stranici. Primjeri, klikovi na određene gumbе ili linkove, učitavanje stranice, promjena vrijednosti regija i *item*-a. Nakon tog događaja izvršavaju se jedna ili više akcija, cijeli tijek akcija se može pratiti na lijevoj strani na slici 22. U ovom slučaju dinamični event DA_DELETEROW se pokreće kada se klikne na link koji sadrži `` i koji ima klasu „myclass“ što je postavljeno u postavkama u donjem desnom kutu (vidi: Slika 22). Kada se klikne na taj link, pokreće se akcija „Confirm“ (vidi: Slika 23), u slučaju potvrde, P1_ID_AKTIVNOSTI poprima vrijednost ID_AKTIVNOSTI retka s tim linkom sa „Set Value“ (vidi: Slika 24) i pokreće se „Execute Server-side Code“ koji briše redak koji je sadržavao taj ID_AKTIVNOSTI pomoću PL/SQL koda (vidi : Slika 25).



Slika 23 Postavke akcije „Confirm“

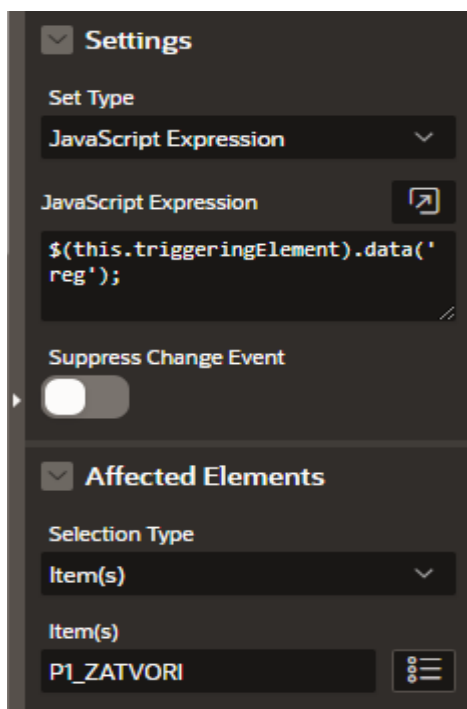


Slika 24 „Set Value“ postavke

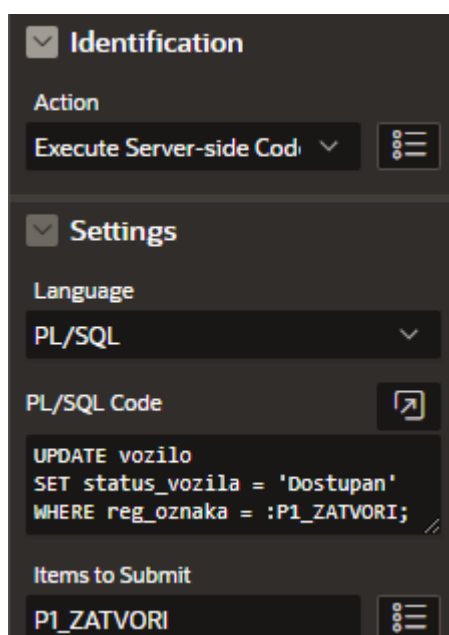


Slika 25 „Execute Server-side Code“ postavke

Nakon tih akcija, pokreće se drugi „Set Value“, ali u ovom slučaju „Set Value“ postavlja *item* P1_ZATVORI na vrijednost REG_VOZILA (vidi: Slika 26). Nakon toga pokreće se druga „Execute Server-side Code“ (vidi: Slika 27).



Slika 26 Drugi „Set Value“



Slika 27 Drugi PL/SQL kod

Vozilo na koje se otvara aktivnost, odmah poprima STATUS_VOZILA „Nedostupno“, radi lakšeg kontroliranja voznog parka. Kada se pobriše aktivnost, sprema se REG_OZNAKA te se STATUS_VOZILA u tablici VOZILO mijenja na „Dostupan“, zatim je vozilo opet dostupno za izdavanje ili novu aktivnost, što je omogućeno PL/SQL kodom (vidi: Slika 27).

4.3.4 Forma aktivnosti vozila

Da bi se dodala nova aktivnost potreban je unos novih podataka u tablicu. Oracle APEX to znatno olakšava tako da automatski kreira sve što smatra potrebnim za unos podataka u tablicu. Kasnije developer može po svojoj želji uređivati, ali glavne funkcionalnosti su odmah gotove. Za izradu forme kreira se nova stranica i od ponuđenih opcija bira se „Form“ (vidi: Slika 13).

The screenshot shows the 'Create Form' dialog in Oracle APEX. The 'Page Definition' section includes: Page Number (4), Name (AktivnostiVozilaModal), and Page Mode (Normal and Modal Dialog). The 'Data Source' section includes: Data Source (Local Database), Source Type (Table), Table / View Owner (WKSP_RENTACARZAVRSNI), and Table / View Name (AKTIVNOSTI_VOZILA). Navigation options are visible at the bottom, including a 'Next >' button.

Slika 28 Kreiranje forme u obliku „Modal Dialog“

Zatim se upisuje broj stranice, ime i određuje se hoće li forma biti nova stranica ili dijalog. U ovom slučaju se bira dijalog te se ne ide u navigaciju već se poziva samo klikom na gumb, prilikom čega se otvara prozor unutar prozora. Potrebno je još označiti tablicu u koju se dodaju podatci te odabrati primarni ključ tablice (vidi: Slika 28).

Opis Aktivnosti

Vrijeme Pocetka

Vrijeme Zavrsetka

Id Lokacije Pocetka

Id Lokacije Zavrsetka

Reg Vozila

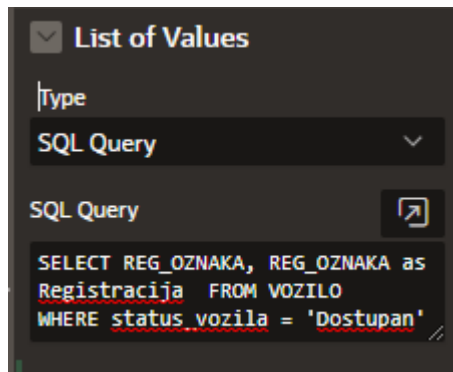
- Alfa Romeo
- Alfa Romeo
- Seat
- Vw
- Škoda

Cancel Create

Slika 29 Rezultat izrade forme

Oracle APEX ne pogodi uvijek vrijednosti koje *developer* želi, ali to se lako izmjeni sa listama vrijednosti koje se kreiraju u aplikaciji.

U postavkama tog unutar regije forme (vidi: Slika 30) postavlja se lista vrijednosti prema potrebama korisnika. Registracija vozila je primarni ključ te tablice. Generalno gledajući, primarni ključevi se ne prikazuju u formama već se prikazuje konkretna vrijednost atributa u ovom slučaju marka vozila (vidi: Slika 29). Ali, u ovom slučaju potrebna je REG_OZNAKA te se taj problem rješava kao što je prikazano na *slici 30*.



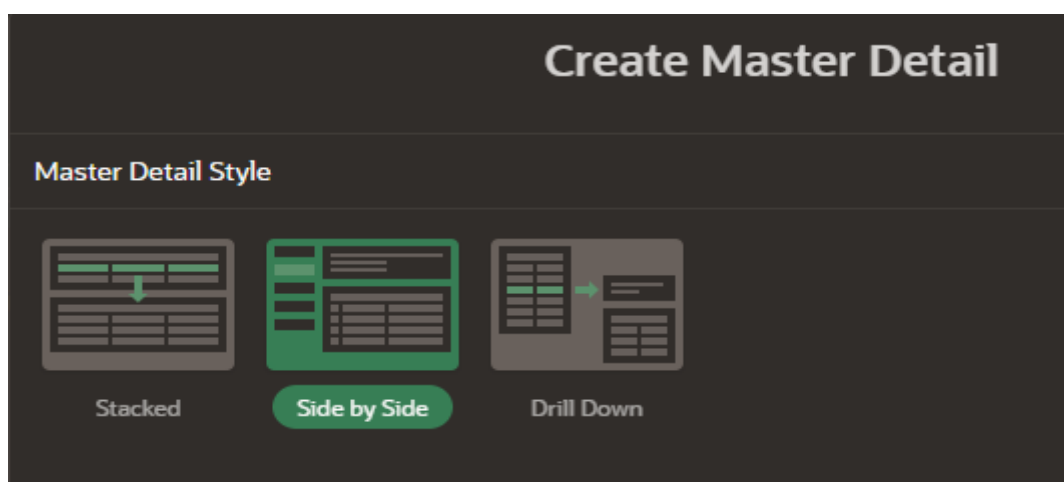
Slika 30 Lista vrijednosti za formu

Pritom se dodaje i uvjet koji ograničava vidljive registrarske oznake, na one automobile, kojima je STATUS_VOZILA „Dostupan“ (vidi: Slika 30).

4.3.5 Master Detail

Master Detail je izvješće koje reflektira *one-to-many* relaciju između dvije tablice. Generalno gledajući ta izvješća prikazuju red glavne tablice i više ili jedan red tablice detalja. Nakon što se označi jedan redak glavne tablice, u drugom izvješću, na istoj stranici, se ispisuju redci koji su povezani s glavnim izvješćem preko vanjskog ključa.

U ovoj aplikaciji *Master Detail* služi za prikazivanje zatvorenih *bookinga* i njihovih računa, ali ne i njihovo dodavanje, brisanje ni uređivanje.



Slika 31 Stilovi Master Detail-a

Potrebno je kreirati novu stranicu i odabrati *Master Detail* iz ponuđenih opcija (vidi: Slika 13). Bira se „*Side by Side*“ radi najbolje preglednosti za ovaj slučaj (vidi: Slika 31).

The screenshot shows a 'Create Master Detail' dialog box. It is divided into two main sections: 'Page Definition' and 'Master Data Source'. In the 'Page Definition' section, there are two fields: 'Page Number' with the value '17' and 'Name' with the value 'Arhiva bookinga'. In the 'Master Data Source' section, there are three tabs: 'Local Database', 'REST Enabled SQL Service', and 'REST Data Source'. The 'Local Database' tab is selected. Below the tabs, there are two fields: 'Table / View Owner' with a dropdown menu showing 'WKSP_RENTACARZAVRSNI' and 'Table / View Name' with the value 'BOOKING'. At the bottom left, there is a 'Navigation' section with a right-pointing arrow. At the bottom of the dialog, there are three buttons: a back arrow, 'Cancel', and 'Next >'.

Slika 32 Izrada Master Detail izvješća

Nakon odabira imena i broja stranice, za glavnu tablicu postavlja se *BOOKING* (vidi: Slika 32), zbog toga što se po svakom *bookingu* treba tražiti račun.

Create Master Detail

Primary Key

* Primary Key Column 1: ID_BOOKINGA (Number) ?

Primary Key Column 2: - Select - ?

* Primary Display Column: ID_BOOKINGA (Number) ?

Secondary Display Column: REG_VOZILA (Varchar2) ?

< Cancel Next >

Slika 33 Izrada Master Detail izvješća 2

Radi preglednosti, vrijednosti koje se odabiru za prikaz su ID_BOOKINGA i REG_VOZILA, i te vrijednosti će se prikazivati na glavnoj listi *Master Detail*-a (vidi: Slika 33).

Create Master Detail

Detail Page Definition

* Page Number: 18 ?

* Name: Racuni ?

Detail Data Source

Data Source: Local Database REST Enabled SQL Service REST Data Source ?

Show Only Related Tables: ?

* Table / View Owner: WKSP_RENTACARZAVRSNI ?

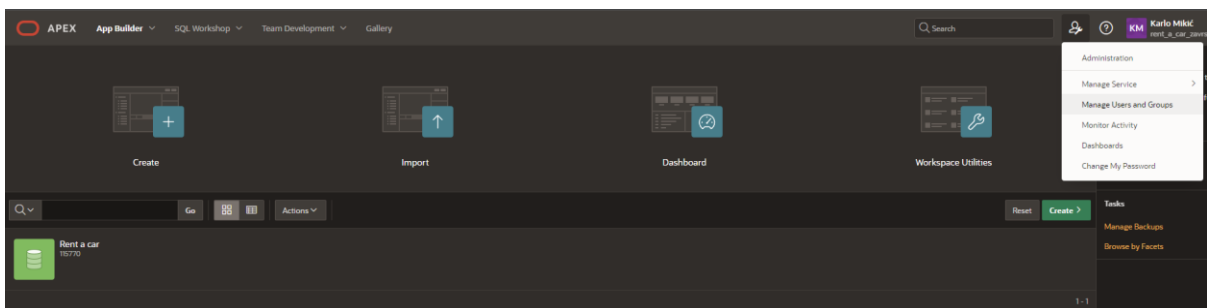
* Table / View Name: RACUNI_REPORT ?

< Cancel Next >

Slika 34 Izrada Master Detail izvješća 3

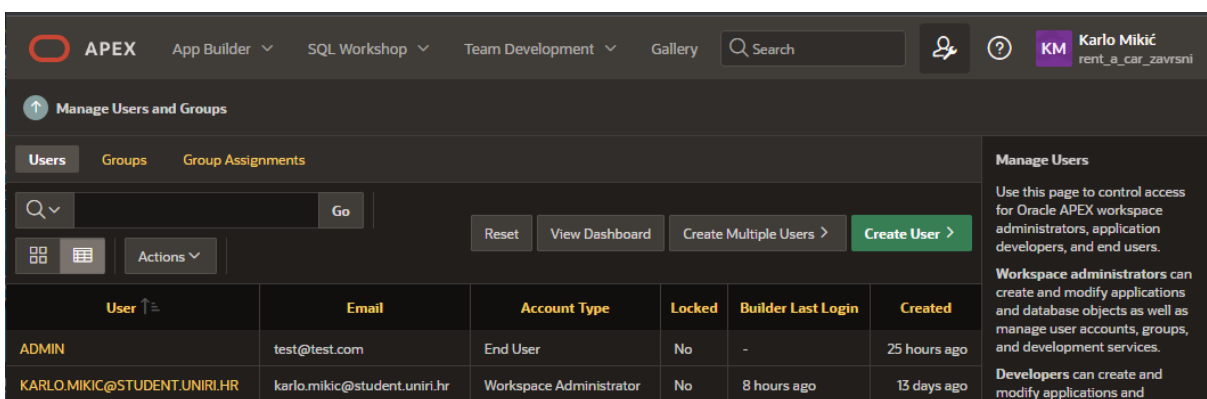
Zatim se upisuje tablica detalja, koja se spaja sa glavnom tablicom preko vanjskog ključa te se za svaki *booking* prikazuje njegov pripadajući račun. U ovom slučaju potreban je pogled RACUNI_REPORT koji sadrži detalje računa koji su potrebni (vidi: Slika 34).

5. Dodavanje novih korisnika u aplikaciju



Slika 35 Dodavanje novih korisnika

U gornjem desnom kutu Oracle APEX-a je opcija „*Manage Users and Groups*“. Klikom na taj link se dolazi do izbornika u kojem se dodaju novi korisnici (vidi: Slika 35).



Slika 36 Izbornik „*Manage Users and Groups*“

U izborniku (vidi: Slika 36) se vidi popis svih korisnika i administratora, te interaktivno izvješće s kojim se može filtrirati ili tražiti po određenim vrijednostima. Klikom na „*Create User*“ otvara se forma za upis podataka korisnika.

Create User

Cancel
Create and Create Another
Create User

Show All
User Identification
Account Privileges
Password
Group Assignments

*** Username**

*** Email Address**

First Name

Last Name

Description

Default Date Format

Account Privileges

Default Schema

Accessible Schemas (null for all)

User is a workspace administrator:

User is a developer:

App Builder Access

SQL Workshop Access

Team Development Access

Set Account Availability

Password

*** Password** Passwords are case sensitive

*** Confirm Password**

Require Change of Password on First Use

Use this page to control access for Oracle APEX application developers, workspace administrators and end users.

Developers can create and edit applications as well as create and modify database objects.

Workspace administrators can additionally create and edit user accounts, manage groups, manage development services.

End users have no development privileges and are defined to provide access control to applications that do not use an external authentication scheme.

Slika 37 Forma za upis novog korisnika

U formu (vidi: Slika 37) se upisuju detalji za prijavu, podaci o korisniku te privilegija tog korisnika. Uzevši u obzir da je aplikacija samo za zaposlenike, nije potrebno dodavati ni oduzimati nikakve posebne privilegije, već će se samo moći prijaviti zaposlenici.

6. Zaključak

Izrađena aplikacija je pojednostavljeni primjer aplikacija koje se koriste u poduzećima koja se bave iznajmljivanjem vozila. Zaposlenici mogu pratiti stanje vozila, stvarati nove aktivnosti koje se prate unutar aplikacije te ih i zatvarati, izrađivati nove *bookinge*, unositi nove klijente i vozila, zatvarati *bookinge* i pratiti kada se vozila trebaju vratiti. Vrlo bitna implementacija je dostupnost vozila, s time aplikacija osigurava da se ne iznajmljuje nedostupno vozilo te dovodi tvrtku u neugodan položaj spram klijenta.

Prije same izrade aplikacije, na temelju prikupljenih informacija, bilo je potrebno izraditi model EV i relacijski model koji su znatno olakšali planiranje i izradu. Lošim modelom, izrada aplikacije, bila bi teška, čak u nekim slučajevima i neizvediva.

Oracle APEX ima mnogo prednosti, a glavna prednost i osobina cijele platforme je niski kod (engl. Low code). S *low code* okruženjem vrlo je jednostavno vizualno „slagati“ aplikaciju bez ikakvog ručno napisanog koda. U isto vrijeme nudi se i opcije pisanja koda i proširivanja aplikacije, što ga čini usporedivim s ozbiljnim developerskim aplikacijama. To omogućuje vrlo brzu izradu aplikacije, jednostavnost korištenja cijele platforme te veliku fleksibilnost.

Uz prednosti Oracle APEX ima i svoje nedostatke. Mogu se koristiti samo Oracle servisi, Oracle baza podataka te s time i isključivo PL/SQL. To znači da je developer ograničen na upotrebu Oracle proizvoda, što nije poželjno. Još jedan bitan nedostatak je kontrola verzija (engl. *Version control*) koja je neophodna u razvijanju aplikacije u kojoj sudjeluje cijeli tim *developer*a. Pošto nema kontrole verzija, *developer*i se moraju dogovarati pri izradi aplikacije, kako ne bi radili na predmetima koji se preklapaju.

Može se zaključiti da je Oracle APEX snažan alat za izradu i većih i manjih aplikacija temeljenih na bazama podataka. Koristi se i u većim i u manjim poduzećima za gotovo sve vrste aplikacija. Oracle APEX nije toliko poznat kao konkurentni alati. Radeći u njemu smatram kako će njegova popularnost rasti ukoliko se nastavi razvijati u skladu s potrebama tržišta.

7. Literatura

[1] Oracle APEX preuzeto 31. Kolovoz 2022. :

<https://apex.oracle.com/en/>

[2] „What is an ER diagram?“ preuzeto 01. Rujan 2022. :

<https://www.lucidchart.com/pages/er-diagrams>

[3] Pavlić, M. Oblikovanje baza podataka. Rijeka: Odjel za informatiku Sveučilišta u Rijeci, 2011.

8. Popis slika

Slika 1 Dijagram EV rent-a-car	5
Slika 2 Vizualni model.....	7
Slika 3 Zahtjev za Radno okruženje (engl. Workspace).....	8
Slika 4 Izrada tablice BOOKING	9
Slika 5 Postavljanje primarnog ključa	10
Slika 6 Dodavanje vanjskih ključeva	11
Slika 7 Dodavanje ograničenja	12
Slika 8 Potvrda izrade tablice	13
Slika 9 Grafičko sučelje "App Builder"	14
Slika 10 Opcije kreiranja aplikacije	14
Slika 11 Izbornik za kreiranje nove aplikacije.....	15
Slika 12 Izgled nove aplikacije u „App Builder“	15
Slika 13 Popis ponuđenih stranica za izradu.....	16
Slika 14 Izrada prazne stranice	17
Slika 15 Izgled prazne stranice u „App Builder“	17
Slika 16 Spajanje izvješća s pogledom	20
Slika 17 Izvješće s dodanim gumbima.....	21
Slika 18 SQL Query izvješća	21
Slika 19 Ponašanje gumba Nova aktivnost.....	22
Slika 20 Dodavanje linka na gumb	22
Slika 21 Odabir stranice i prosljeđivanje vrijednosti	23
Slika 22 „jQuery Selector“ sa dinamičnim eventom	24
Slika 23 Postavke akcije „Confirm“	24
Slika 24 „Set Value“ postavke	25
Slika 25 „Execute Server-side Code“ postavke	25
Slika 26 Drugi „Set Value“	26
Slika 27 Drugi PL/SQL kod.....	26
Slika 28 Kreiranje forme u obliku „Modal Dialog“	27
Slika 29 Rezultat izrade forme.....	28
Slika 30 Lista vrijednosti za formu	29
Slika 31 Stilovi Master Detail-a.....	29
Slika 32 Izrada Master Detail izvješća.....	30
Slika 33 Izrada Master Detail izvješća 2.....	31
Slika 34 Izrada Master Detail izvješća 3.....	31
Slika 35 Dodavanje novih korisnika	32
Slika 36 Izbornik „Manage Users and Groups“	32
Slika 37 Forma za upis novog korisnika.....	33

9. Popis priloga

Oracle APEX link za pristup aplikaciji:

https://apex.oracle.com/pls/apex/r/rent_a_car_zavrzni/rent-a-car/

Username: admin

Password: admin123