Maloprodaja sportske opreme - poslovna aplikacija nad relacijskom bazom podataka (Oracle APEX)

Zelenika, Antonio

Sveučilište u Rijeci – Fakultet informatike i digitalnih tehnologija

Antonio Zelenika

Maloprodaja sportske opreme - poslovna aplikacija nad relacijskom bazom podataka (Oracle APEX)

Završni rad

Mentor: doc. dr. sc. Danijela Jakšić

Rijeka, 19.9.2022.

FAKULTET INFORMATIKE I DIGITALNIH TEHNOLOGIJA
SVEUČILIŠTE U RIJECI

Rijeka, 14.6.2022.

Zadatak za završni rad

Pristupnik: Antonio Zelenika

Naziv završnog rada: Maloprodaja sportske opreme - poslovna aplikacija nad relacijskom bazom podataka (Oracle APEX)

Naziv završnog rada na eng. jeziku: Sports Retail Shop - 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 procesa maloprodaje sportske opreme. Aplikacija će biti izgrađena nad relacijskom BP pomoću alata Oracle APEX. Na početku će biti specificirani zahtjevi i potrebne značajke aplikacije. Sljedeći korak bit će dizajn baze podataka, odnosno izrada modela entiteti i veze. Transformacijom modela entiteti i veze konstruirat će se relacijski model. Na temelju relacijskog modela izgradit će se baza podataka. Nakon toga kreće izrada same aplikacije. Aplikacija će raditi nad relacijskom BP u navedenoj aplikacijskoj domeni (maloprodaja sportske opreme).

Mentor

doc. dr. sc. Danijela Jakšić

Voditelj za završne radove

doc. dr. sc. Miran Pobar

Zadatak preuzet: 14.6.2022.

(potpis pristupnika)

Sadržaj

1. Uvod

Bez obzira čime se bavili, danas gotovo da ne postoji posao koji se ne može poboljšati uz neku vrstu poslovne aplikacije. Glavna zadaća poslovnih aplikacija je da olakša posao pružatelju usluga, a samim time utječe na to da korisnik tih usluga jednostavnije dobije ono što želi. Kako bi bilo koja usluga što bolje funkcionirala potrebno je organizirati podatke koje je potrebno koristiti.

Izrada aplikacije može trajati mjesecima, a osim što može trajati dugo, izrada može biti i skupa. Jedna od opcija za smanjivanje vremena izrade aplikacije je da koristimo alat za brzi razvoj aplikacija, odnosno RAD alat (Rapid Application Development). Prednosti pristupa kreiranja aplikacija pomoću RAD alata ovise o potrebama aplikacije. Kreiranje poslovne aplikacije maloprodaje sportske opreme pristup izrade putem RAD alata je dobar odabir jer nam osigurava kvalitetu i kontrolu rizika, a aplikacija je dovoljno jednostavna da se ne moramo brinuti o skalabilnosti.

U izradi ove poslovne aplikacije koristit ćemo relacijsku bazu podataka. Kako bi izradili aplikaciju nad relacijskom bazom podataka proći ćemo kroz korake dizajniranja dijagrama entiteti – veze, a zatim i relacijskog modela. Proći ćemo i kroz izradu same baze podataka, na više načina koje nam nudi Oracle APEX. Vidjet ćemo koje funkcije nam Oracle APEX nudi i kako nam pojednostavljuje izradu baze podataka, a i izradu same aplikacije.

U ovom radu proći ćemo kroz proces izrade aplikacije korištenjem Oracle APEX platforme. Radi se o maloprodaji sportske opreme, kupac će moći pretraživati po kategorijama, nazivu ili odabranim kombinacijama filtera, stavljati odabrane predmete u „košaricu“, cijenu pojedinih proizvoda i ukupnu cijenu košarice. Administrator će imati uvid u sve kupnje, iznose narudžbe, u statistike koje se kategorije najbolje prodaju i slično. Također, administrator može dodavati i izmjenjivati podatke. U radu će biti detaljno opisane relacije između tablica koje će se koristiti u izradi.

2. Baza podataka

Kako bi mogli pohranjivati podatke potrebne da bi maloprodaja sportske opreme funkcionirala moramo izgraditi bazu podataka. Baza podataka je skupina ustrojenih, logički povezanih zapisa ili datoteka; ili skupina datoteka koje sadrže zapise s podacima što su međusobno u nekoj vezi, a korisnici ih mogu rabiti u različite svrhe; može se sastojati i od pomoćnih datoteka (na primjer datoteke s indeksima) (Pavlić, 2011.).

Relacijske baze podataka su jedne od najpoznatijih vrsta baza podataka. Koristeći relacijske baze podataka, podatke razvrstavamo u relacije i njene atribute. Svaka relacija je jedna tablica, a atributi su njeni stupci. Veze između relacija ovise su njihovim atributima. Relacijska baza podataka (engl. relational database) je skup u vremenu promjenljivih relacija opisanih u shemi baze podataka. (Pavlić, 2011.).

Za potrebe procesa maloprodaje sportske opreme potrebno je osmisliti bazu podataka koja će sadržavati podatke o proizvodima koji se prodaju i podatke vezane za samu prodaju odabranog proizvoda. Kako bi izgradili aplikaciju nad relacijskom bazom podataka, prvo moramo izraditi model entiteti i veze.

2.1 Model entiteti – veze

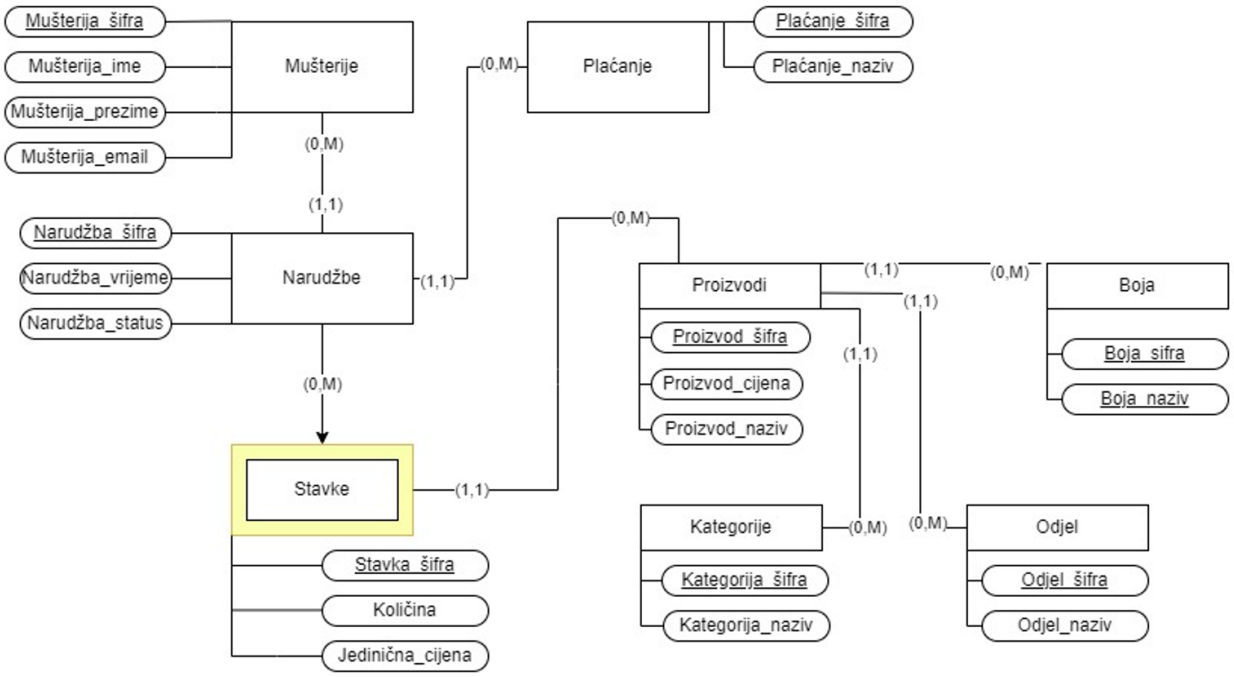
Model entiteti – veze je grafički prikaz međusobno povezanih grupa podataka promatranog sustava (Pavlić, 2011.). Dijagram strukture takvog modela naziva se dijagram entiteta i veza (DEV).

Model entiteti – veze sastoji se od entiteta, veza i atributa. Entiteti se razvrstavaju u tipove entiteta na temelju zajedničkih svojstava ili uvjeta definiranih za potrebe određenog sustava (Pavlić, 2011.). U kontekstu tablice relacijske baze podataka to bi značilo da je jedan red tablice jedan entitet, a cijela tablica je tip entiteta. Tip entiteta označavamo s pravokutnikom (Pavlić, 2011.)(slika 2).

Karakteristike entiteta ih detaljnije opisuju vrijednostima. Sve vrijednosti jedne karakteristike nekog tipa entiteta moraju biti istog tipa vrijednosti, a imenovanu karakteristiku zovemo atribut. Atributi mogu zadovoljavati uvjete jedinstvenosti i neredundantnosti, a takav skup atributa nazivamo ključ tipa entiteta. Uvjet jedinstvenosti je zadovoljen ako ne postoje dva pojavljivanja iste vrijednosti atributa u jednom tipu entiteta i ne postoje dva tipa entiteta koji dijele skup atributa za ključ. Uvjet neredundantnosti je zadovoljen ako izostavljanjem bilo kojeg atributa iz skupa atributa za ključ gubimo uvjet jedinstvenosti. U modelu entiteti – veze uvodimo samo jedan ključ po tipu entiteta. Taj ključ zovemo primarni ključ i označava se podcrtavanjem atributa koji je ključ. Atribute tipa entiteta označavamo kao poveznicu na tip entiteta. (Pavlić, 2011.) (slika 1).

Veza predstavlja odnos među dva tipa entiteta, odnosno opisuje nam što povezuje dva tipa entiteta i koja je njihova brojnost. Tip veza je naziv za skup veza između dva ista tipa entiteta. Brojnost nam govori koliko entiteta iz tipa entiteta A sudjeluje u tipu veze V s entitetom iz tipa entiteta B. Brojnost označavamo s gornjom i donjom granicom kao: (1,1), (0,M) i sl. Jedna linija bez strelica označava dvije veze, po jedna u svakom smjeru, a brojnost se označava se na liniji spoja između tipa entiteta B i veze V za jedan smjer, a za drugi smjer između tipa entiteta A i veze V (Pavlić, 2011.). (slika 1).

2.2 Dijagram entiteti – veze



Slika 1 DEV KUPOVINA

Na slici 1 je prikazan DEV Kupovina. Tipovi entiteta označeni su pravokutnicima, a slabi tip entiteta (Stavke) dodatno je označen dodatnim pravokutnikom. Slabi tip entiteta znači da postojanje tog tipa entiteta ovisi o nekom drugom tipu entiteta. Na primjeru sa slike vidimo da je tip entiteta „Stavke“ slabi tip entiteta. Tip entiteta „Stavke“ egzistencijalno i identifikacijski ovisi o tipu entiteta „Narudžbe“ i zato njihovu vezu označavamo sa strelicom i brojnošću (0,M) jer se podrazumijeva da svaka stavka mora pripadati narudžbi radi koje je stvorena. Tipovi entiteta „Mušterije“ i „Stavke“ povezani su s tipom entiteta „Narudžbe“ s dvije veze: (1,1), (0,M). Brojnost (1,1) označava da jedna mušterija može izvršiti narudžba iz tipa entiteta „Narudžbe“, a uz to u toj istoj narudžbi iz tipa entiteta „Narudžbe“ imati samo jednu vrstu plaćanja iz tipa entiteta „Plaćanje“. Brojnost (0,0) označava da jedna mušterija iz tipa entiteta „Mušterije“ može izvršiti više narudžbi, a isto tako i jedna vrsta plaćanja iz tipa entiteta „Plaćanje“ može se pojaviti na više narudžbi iz tipa entiteta „Narudžbe“. Isto tako možemo vidjeti da jedan proizvod iz tipa entiteta „Proizvodi“ može biti jedne boje iz tipa entiteta „Boje“, pripadati jednom odjelu iz tipa entiteta „Odjel“ i jednoj kategoriji iz tipa entiteta „Kategorije“. Više proizvoda iz tipa entiteta „Proizvodi“ mogu biti iste boje iz tipa entiteta „Boja“ i pripadati istom odjelu i/ili kategoriji iz tipova entiteta „Odjel“ i „Kategorija“. Na narudžbi iz tipa entiteta „Narudžbe“ nalaze se stavke iz tipa entiteta „Stavke“, a na samo jednoj po narudžbi se može pojavljivati proizvod iz tipa entiteta „Proizvod“.

Transformacijom modela entiteti – veze konstruira se relacijski model. Svaki tip entiteta postaje relacija u relacijskom modelu, atributi tipa entiteta postaju atributi relacije, a primarni ključ tipa entiteta postaje primarni ključ relacije. Tipovi veze također mogu postati relacije ako imaju atribute.

U „DEV Kupovina“ imamo sljedeće tipove entiteta: Mušterije, Narudžbe, Stavke (slabi tip entiteta), Proizvod, Kategorija, Odjel, Boja, Plaćanje. Tip entiteta „Mušterije“ postaje relacija „Mušterije“, a atributi tipa entiteta postaju atributi relacije „Mušterija_ime“, „Mušterija_prezime“, „Mušterija_email“. Primarni ključ relacije je isti kao i primarni ključ tipa entiteta- „Mušterija_šifra“, a označava se podcrtavanjem. Dobivena relacija:

- **Mušterije** (Mušterija_šifra, Mušterija_ime, Mušterija_prezime, Mušterija_email)

Koristeći istu metodu dolazimo i do sljedećih relacija:

- **Kategorije** (Kategorija_šifra, Kategorija_naziv)
- **Odjel**(Odjel_šifra, Odjel_naziv)
- **Boje**(Boja_šifra, Boja_naziv)
- **Plaćanje** (Plaćanje_šifra , Plaćanje_naziv)

Ako su dva tipa entiteta povezana s tipom veze kojemu je brojnost (1,1)(0,M), tada se tip veze ne prevodi u relaciju već se ključ tipa entiteta koji ulazi u vezu sa strane s brojnošću M umeće kao vanjski ključ u relaciju dobivenu prevođenjem tipa entiteta s brojnošću (1,1). Tipovi entiteta „Narudžbe“, „Proizvodi“ i „Stavke“ pretvoreni u relacije sadrže vanjske ključeve. U relaciju „Narudžbe“, uz prethodne korake prevođenja, umećemo vanjske ključeve koji su jednaki primarnim ključevima u relacijama „Mušterije“ (primarni ključ: Mušterija_šifra) i „Plaćanje“ (primarni ključ: Plaćanje_šifra). Na isti način iz relacija „Proizvodi“ dobiva vanjske ključeve iz relacija „Odjel“, „Kategorije“ i „Boje“. Vanjski ključevi u relacijskom modelu označavaju se kurzivnim slovima. Dobivene relacije su:

- **Narudžbe** (Narudžba_šifra, Narudžba_vrijeme, Narudžba_status, *Mušterija_šifra, Plaćanje_šifra)*
- **Proizvodi** (Proizvod_šifra, Proizvod_cijena, Proizvod_naziv, Proizvod_slika, Proizvod_detalji, *Kategorija_šifra, Odjel_šifra, Boja_šifra*)

Preostala je relacija dobivena prevođenjem tipa entiteta „Stavke“. Budući da je tip entiteta „Stavke“ slabi tip entiteta koji egzistencijalno i identifikacijski ovisi o tipu entiteta „Narudžbe“, da bi ga preveli u relaciju moramo uz sva prethodno navedena pravila složiti ključ relacije koji se sastoji od ključa slabog tipa entiteta (Stavke) i jakog tipa entiteta (Narudžbe). Vanjske ključeve prima od relacija „Narudžbe“ i „Proizvodi“. Dobivena relacija:

- **Stavke** (Stavke_šifra,Narudžba_šifra, Količina, Jedinična_cijena, *narudžba_šifra proizvod_šifra*)

Kada smo preveli sve tipove entiteta u relacije možemo vidjeti naš relacijski model na temelju kojega se gradi baza podataka za izradu aplikacije:

Mušterije (Mušterija_šifra, Mušterija_ime, Mušterija_prezime, Mušterija_email)

Narudžbe (Narudžba_šifra, Narudžba_vrijeme, Narudžba_status, *Mušterija_šifra, Plaćanje_šifra)*

Stavke (Stavka_šifra,Narudžba_šifra, Količina, Jedinična_cijena, *proizvod_šifra*)

Proizvodi (Proizvod_šifra, Proizvod_cijena, Proizvod_naziv, Proizvod_slika, Proizvod_detalji, *Kategorija_šifra, Odjel_šifra, Boja_šifra*)

Kategorije (Kategorija_šifra, Kategorija_naziv)

Odjel(Odjel_šifra, Odjel_naziv)

Boje(Boja_šifra, Boja_naziv)

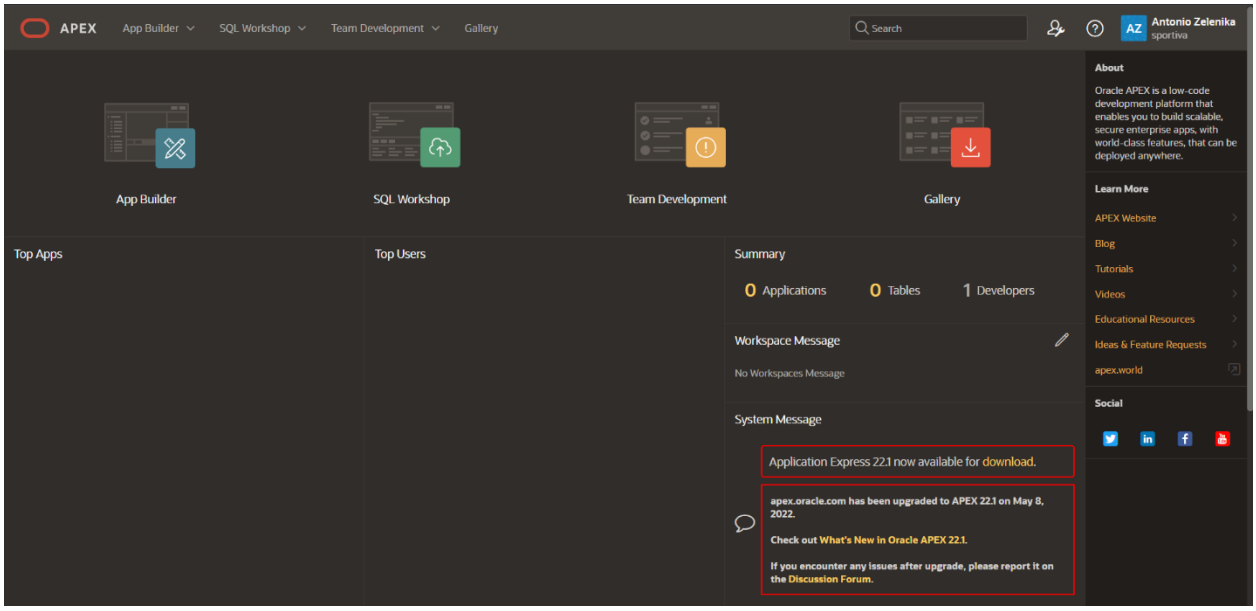
Plaćanje (Plaćanje_šifra , Plaćanje_naziv)

3 Oracle

3.1 Oracle APEX

Oracle APEX je platforma za brzi i relativno jednostavan razvoj aplikacija koja nam pruža mogućnost da izradimo aplikacije uz vrlo malo pisanja koda. Za izradu same baze podataka na kojoj se temelji aplikacija pruža se mogućnost korištenja jezika SQL i Quick SQL. Također, možemo ručno izraditi tablice pomoću intuitivnog sučelja ili ih učitati ako ih već imamo spremne u nekoj datoteci. Ručna izrada tablica je spora, ali primjerena za početnike. Iz odabranih opcija kreira se SQL kod koji se može naknadno uređivati. Prilikom izrade ove aplikacije koristit ćemo Oracle APEX verziju 22.1.3.

Nakon prve prijave u Oracle APEX korisnik vidi prikaz (slika 2):

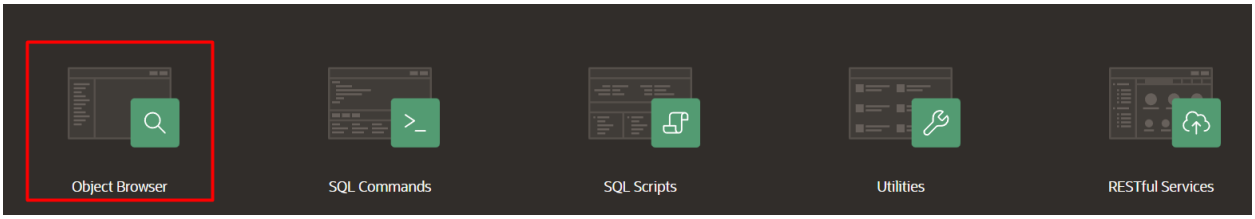


Slika 2 – prva prijava

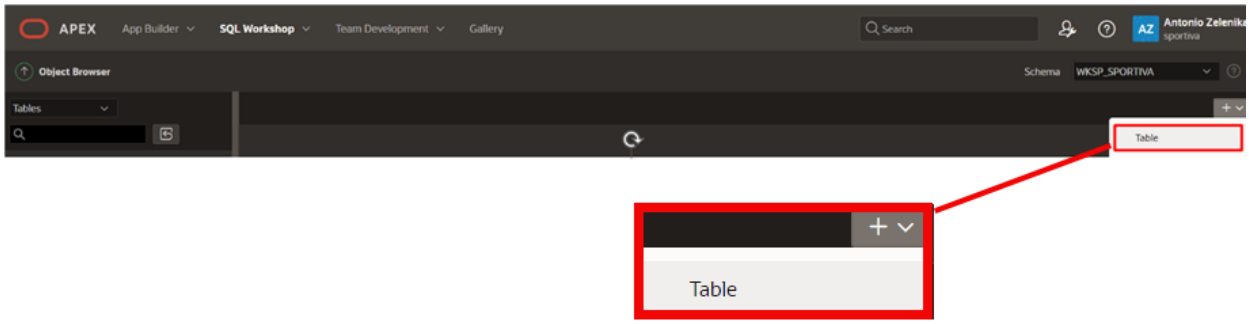
Kako bismo izgradili bazu podataka moramo kliknuti na *SQL Workshop*.

3.2 Baza podataka

Kao što smo već spomenuli, platforma Oracle APEX pruža više načina na koji možemo izgraditi bazu podataka. Najjednostavniji način za kreirati tablicu je tako da u SQL Workshop-u kliknemo na *Object browser* i na padajućem izborniku za dodavanje novog objekta odaberemo novu tablicu.

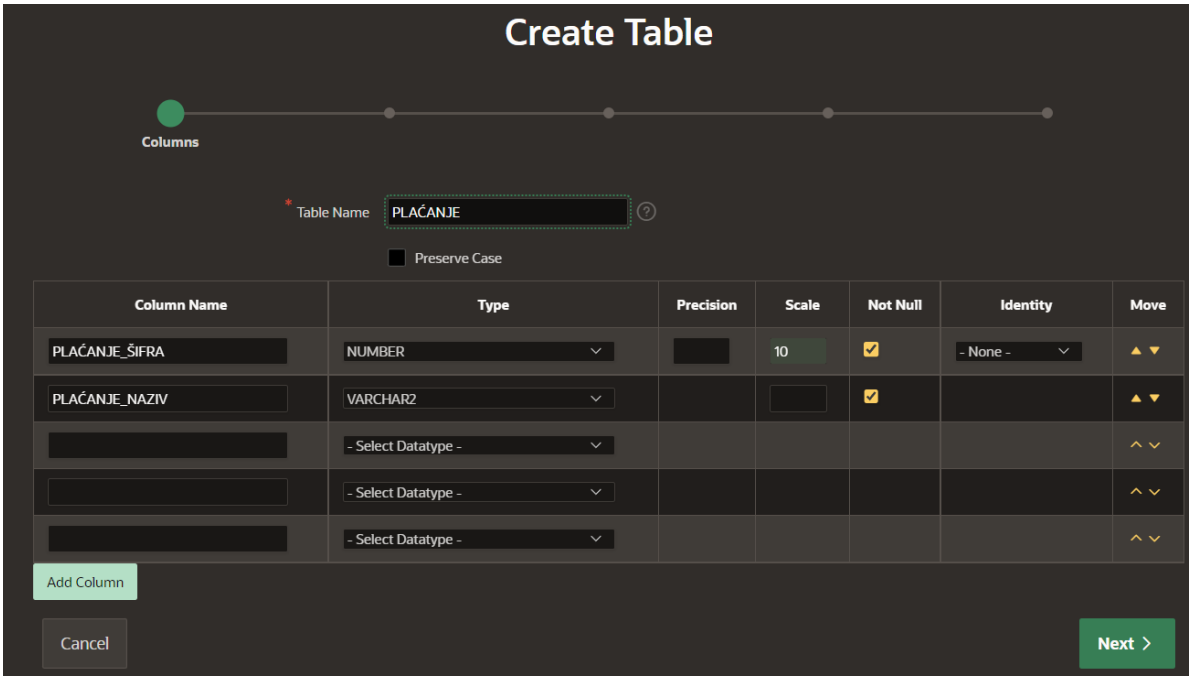


Slika 3 – Object browser

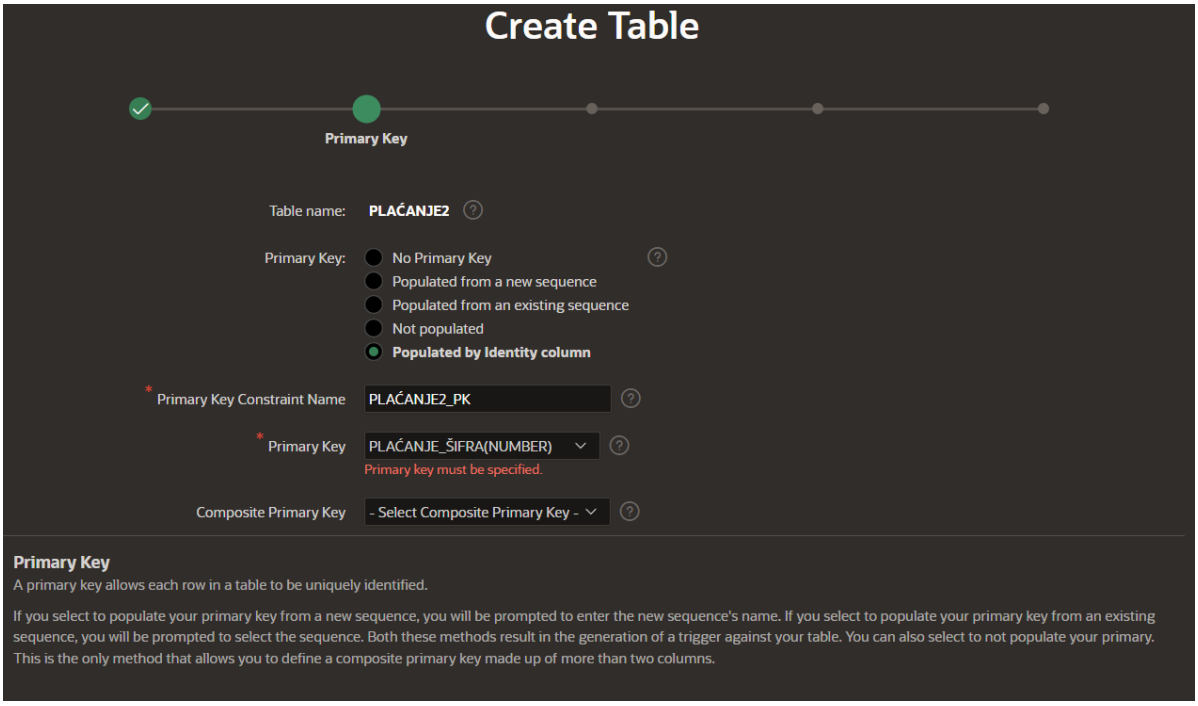


Slika 4 – dodavanje tablice

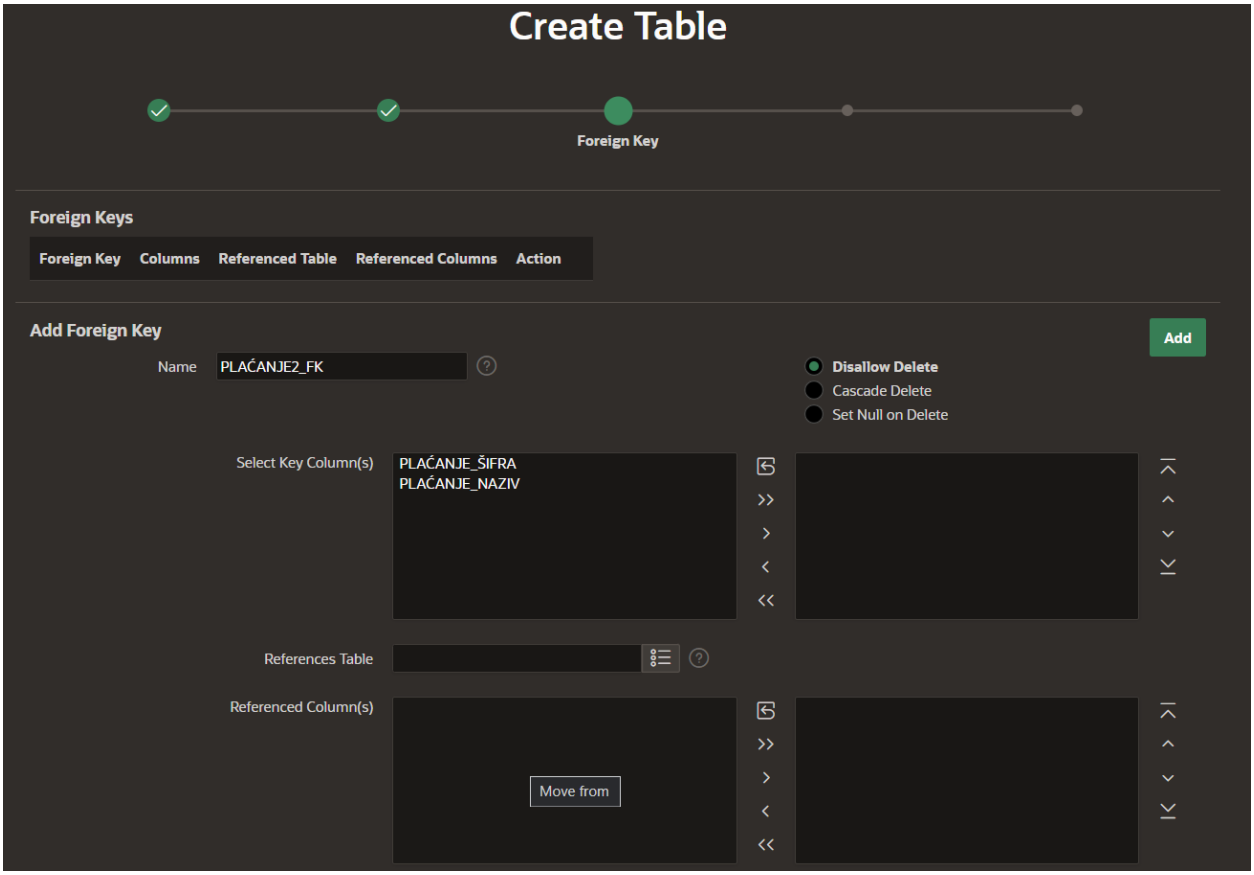
Takav pristup izrade baze podataka, iako je najjednostavniji, zahtjeva previše vremena kao što možemo vidjeti na primjeru izrade tablice *Plaćanje* (slike 5-8).



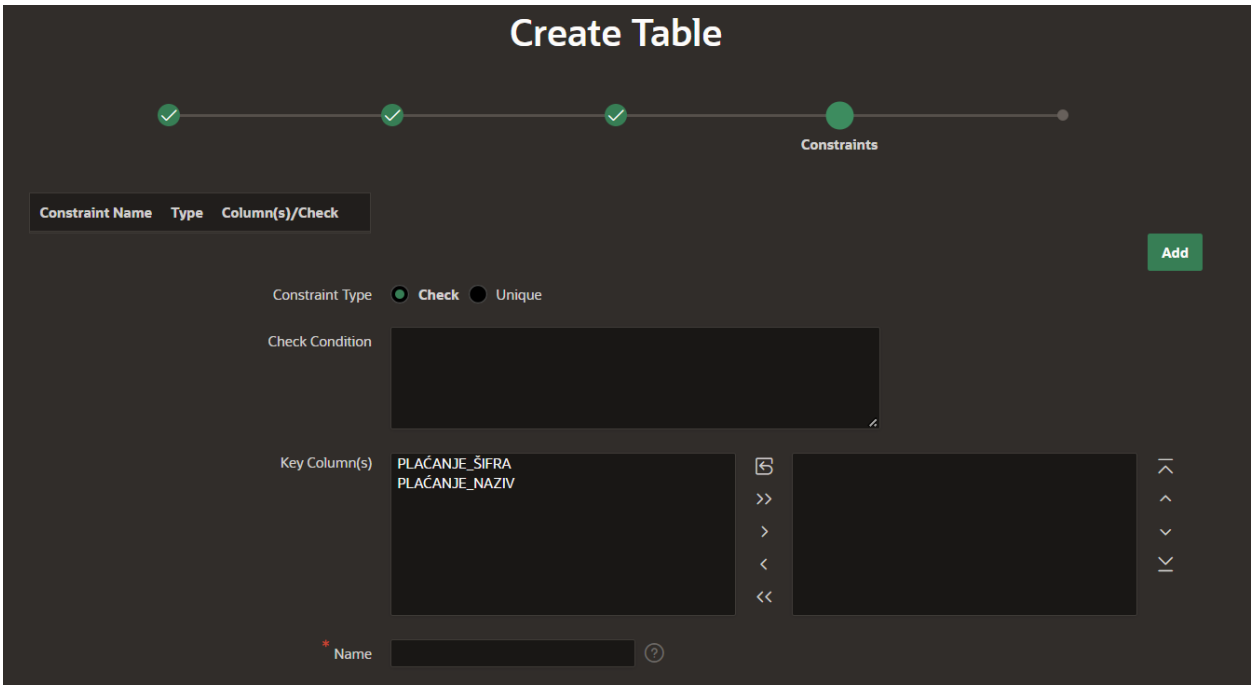
Slika 5 – kreiranje tablice, korak 1



Slika 6 – kreiranje tablice, korak 2



Slika 7 – kreiranje tablice, korak 3



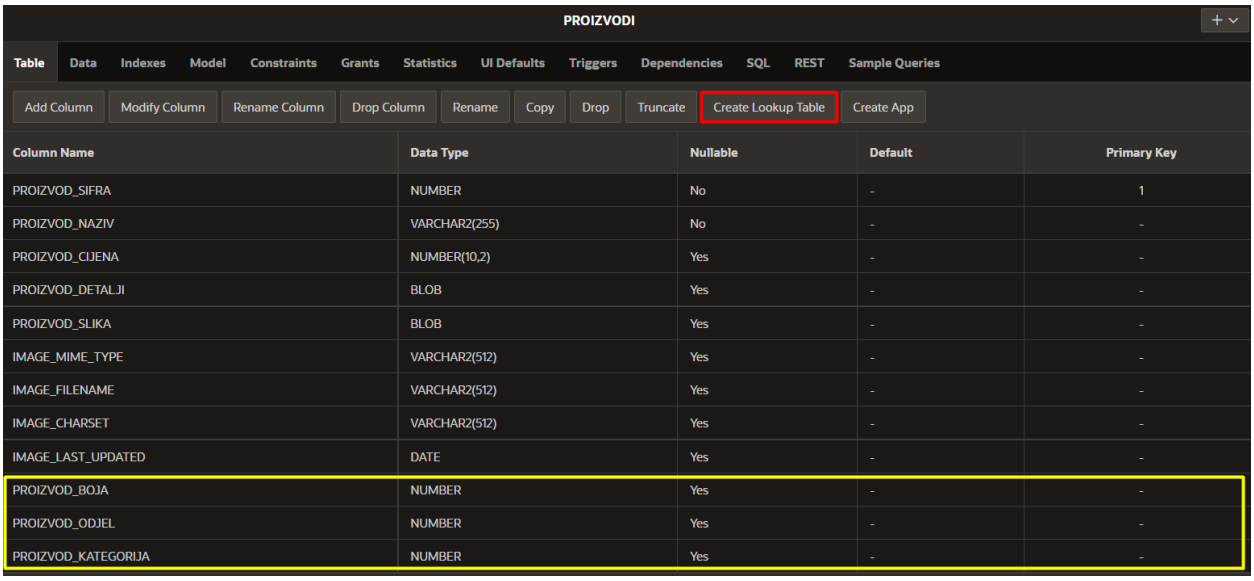
Slika 8 - – kreiranje tablice, korak 4

Da bi napravili tablicu moramo proći kroz četiri koraka odabira (peti korak je samo potvrda). Kada potvrdimo sve odabire možemo vidjeti SQL kod koji je Oracle APEX kreirao iz odabira korisnika. U ovom slučaju SQL kod je:

```
CREATE TABLE  "PLACANJE"
(       "PLACANJE_SIFRA" NUMBER NOT NULL ENABLE,
        "PLACANJE_NAZIV" VARCHAR2(255) NOT NULL ENABLE,
        CONSTRAINT "PLACANJE_PK" PRIMARY KEY ("PLACANJE_SIFRA")
 USING INDEX  ENABLE,
        CONSTRAINT "PLACANJE_NAZIV_U" UNIQUE ("PLACANJE_NAZIV")
 USING INDEX  ENABLE
  )
/
```

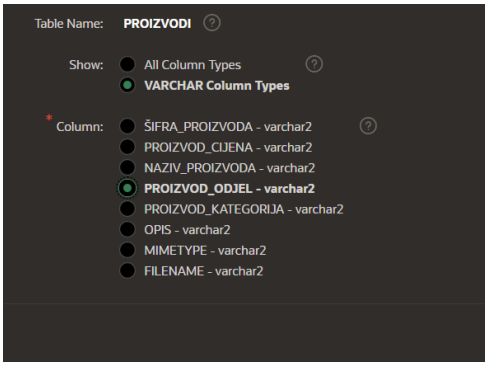
Iz prikazanog koda možemo vidjeti da je učinkovitije pisati kod nego stvarati tablice pomoću odabira, ali neke tablice se isplate kreirati pomoću odabira. Primjer takve tablice bile bi tablice za pretraživanje.

Tablice za pretraživanje poboljšavaju izvođenje upita, a u našoj aplikaciji koristit ćemo ih za stvaranje padajućeg izbornika pri unosu podataka u tablicu. Tako osiguravamo konzistentnost podataka pri izmjene tablica od strane administratora. Npr. pri dodavanju novog artikla u tablicu *Proizvodi* administrator će odabrati kategoriju/boju/odjel umjesto da ih upisuje. Tako se sprječava greška pri pisanju u slučaju da dođe do krivog unosa ili komunikacijska greška ako dva administratora imaju drugačije nazive za istu vrijednost atributa, npr. odjel „muški“ ili „muškarci“. Da bi napravili tablice za pretraživanje odaberemo tablicu čije atribute želimo premjestiti u odvojene tablice. U našem slučaju je to tablica *Proizvodi*. Prvo kreiramo tablicu tako da vanjske ključeve stvorimo kao obične atribute jer tablice u kojima su ti atributi primarni ključevi još ne postoje.

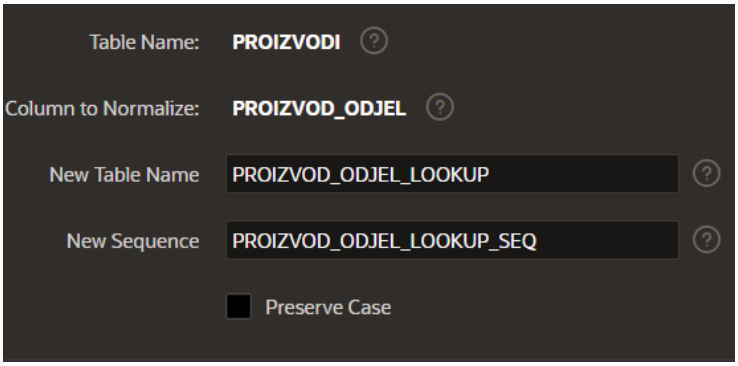


Slika 9 – kreiranje lookup tablice

Crvenom bojom označen je gumb za stvaranje tablice za pretraživanje (slika 9). Žutom bojom označeni su atributi koje ćemo iskoristiti za stvaranje tih tablica. Svaki atribut bit će iskorišten za jednu za tablicu za pretraživanje.

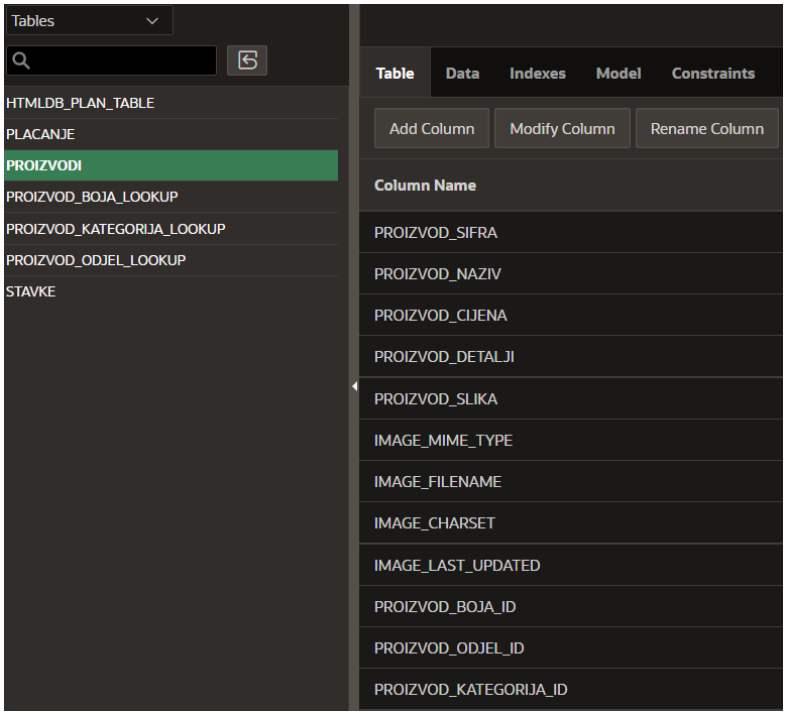


Slika 10 – kreiranje lookup tablice, korak 1



Slika 11 – kreiranje lookup tablice, korak 2

Nakon što potvrdimo odabir, atribut PROIZVOD_ODJEL smo zamijenili s vanjskim ključem PROIZVOD_ODJEL_ID koji je primarni ključ novostvorene tablice za pretraživanje. Kada ponovimo to i s ostala dva atributa dobijemo sljedeće (slika 12):



Slika 12 – dobivene lookup tablice

Vidimo da su stvorene tri nove tablice (slika 12), a atributi su postali vanjski ključevi. Uz to, Oracle APEX nam je stvorio i okidač (trigger) za svaku od novih tablica. To vidimo u SQL kodu nove tablice:

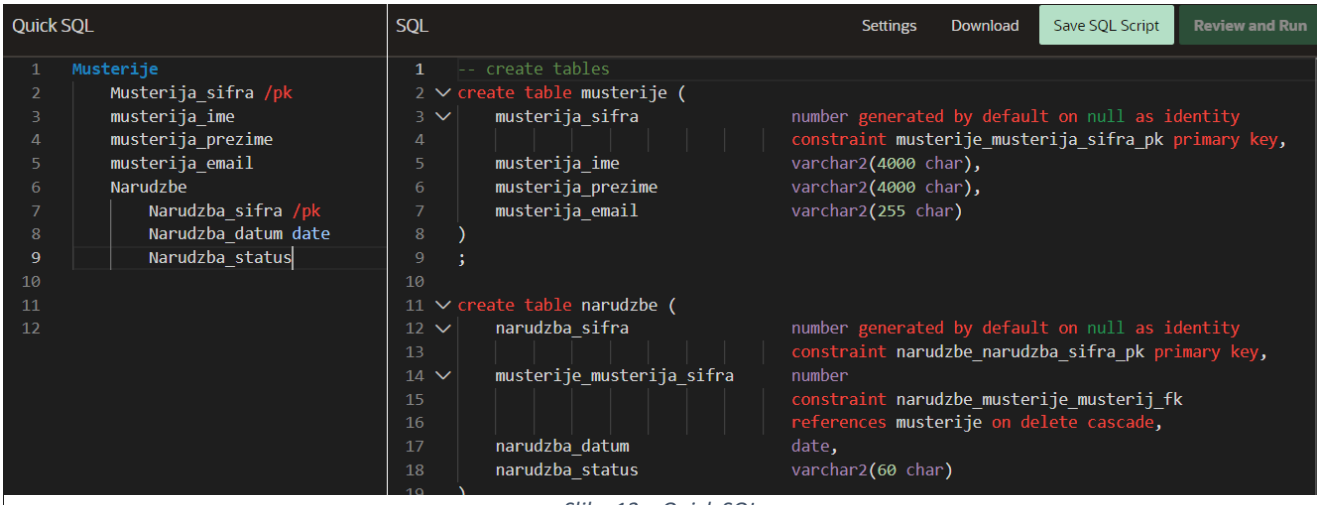
```
CREATE TABLE  "PROIZVOD_BOJA_LOOKUP"
   (      "PROIZVOD_BOJA_ID" NUMBER NOT NULL ENABLE,
  "PROIZVOD_BOJA" VARCHAR2(4000) NOT NULL ENABLE,
   PRIMARY KEY ("PROIZVOD_BOJA_ID")
  USING INDEX  ENABLE
   )
/

CREATE OR REPLACE EDITIONABLE TRIGGER  "T_PROIZVOD_BOJA_LOOKUP" before inser
or update on "PROIZVOD_BOJA_LOOKUP" for each row begin if inserting and
:new."PROIZVOD_BOJA_ID" is null then
  for c1 in (select "PROIZVOD_BOJA_LOOKUP_SEQ".nextval nv from dual) loop
     :new."PROIZVOD_BOJA_ID" := c1.nv;   end loop; end if; end;

/
ALTER TRIGGER  "T_PROIZVOD_BOJA_LOOKUP" ENABLE
/
```

Okidač služi za automatsko generiranje primarnog ključa pri dodavanju nove boje u bazu podataka. Isto vrijedi i za tablice odjel_lookup i kategorija_lookup.

3.2.1. Quick SQL

Quick SQL je način da brzo kreiramo ili uređujemo tablice koristeći jednostavniju sintaksu pomoću kojih sustav generira SQL. Dovoljno je unijeti samo imena tablice i stupaca, a sustav će automatski popuniti tipove podataka (ako ih korisnik sam ne definira). Na slici 13 je prikazan Quick SQL kod za kreiranje dvije tablice: *Musterije* i *Narudzbe*. Tablica *Narudzbe* sadrži vanjski ključ koji se automatski generirao jer je tablica definirana u ravnini sa stupcima tablice *Musterije*.

Sada kada smo izgradili cijelu bazu podataka možemo prijeći na izradu paketa za upravljanje narudžbama, a onda i na samu izradu aplikacije.

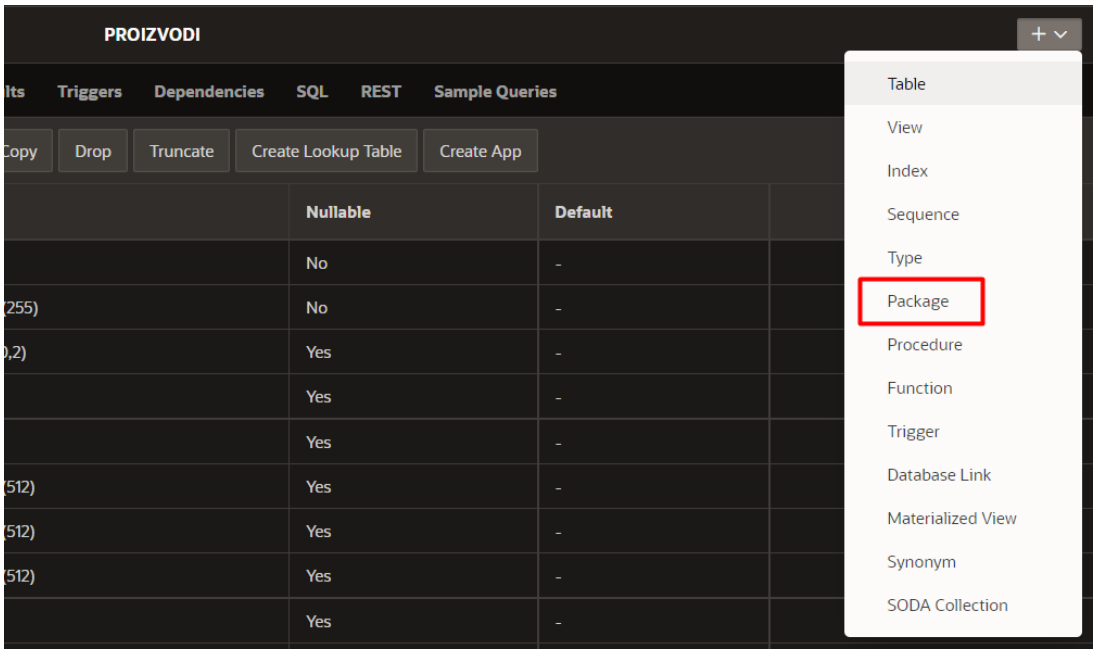


Slika 13 – Quick SQL

4 Aplikacija

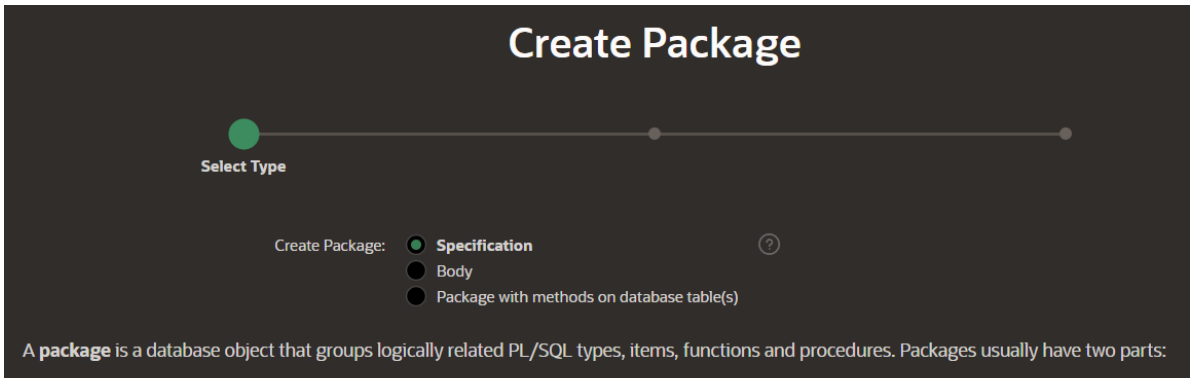
4.1 Upravljanje narudžbama

Da bi korisnik mogao uspješno obaviti kupnju koristit ćemo mogućnost Oracle APEX-a da

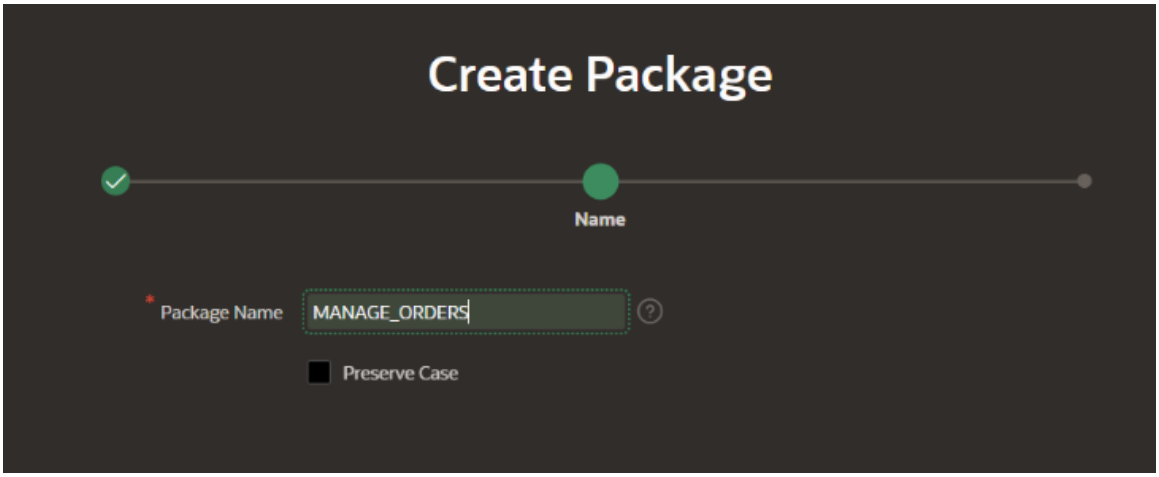


Slika 14 – odabir za kreiranje paketa

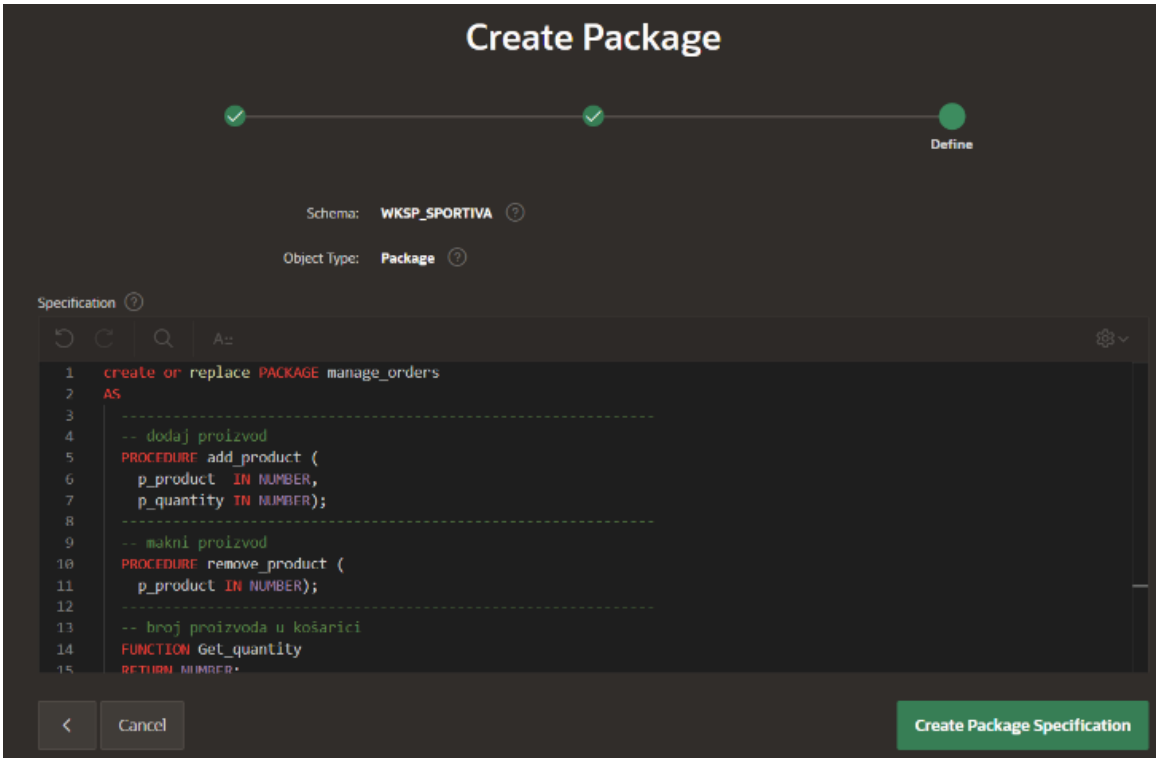
privremeno pohranjuje podatke korisnika te ih tako može procesirati prema zahtjevima aplikacije. U našem slučaju koristit ćemo pohranjene podatke da izvršimo narudžbu, te da nakon narudžbe pohranimo podatke u bazu podataka. Nakon svake kupnje generirat će se nova narudžba koja će se pohraniti u bazu podataka u tablicu *Narudžbe*. Koristeći Oracle APEX funkcije i procedure za izradu narudžbe kreiramo paket na sljedeći način (slika 14-17):



Slika 15 – kreiranje paketa, korak 1



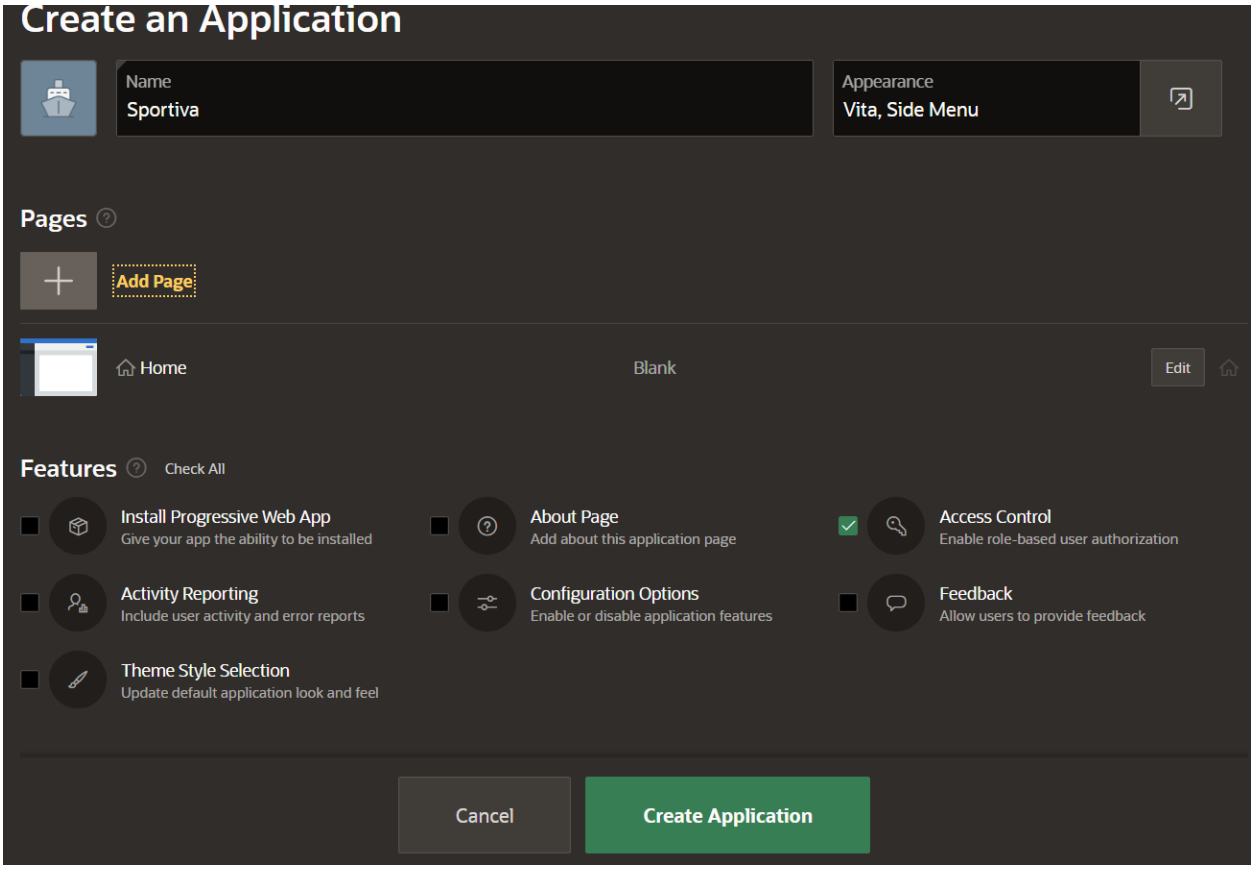
Slika 16 – kreiranje paketa, korak 2



Slika 17 – kreiranje pakera. korak 3

4.2 Kreiranje Aplikacije

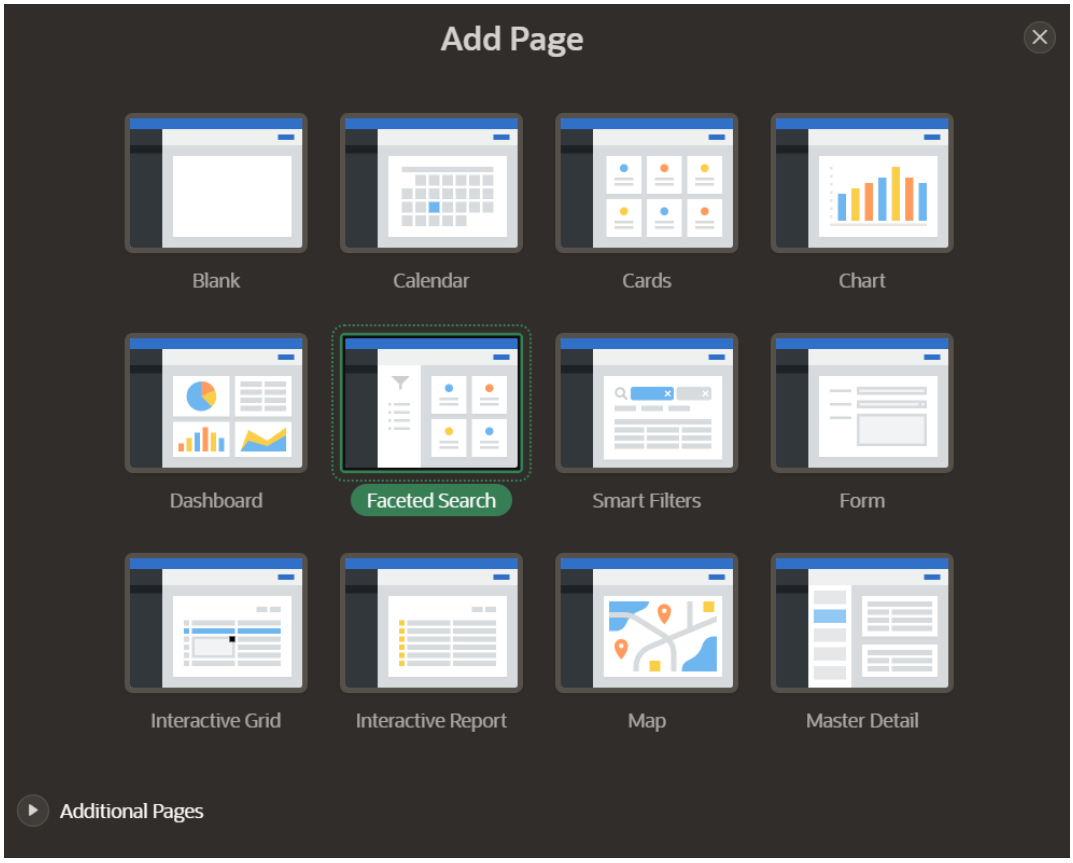
Budući da nam je baza podataka spremna, možemo preći na kreiranje aplikacije. Na početnoj stranici klikom na *App Builder* dolazimo do izbornika na kojemu je potrebno (u našem slučaju) odabrati *Create* > *New Application*. Dolazimo do izbornika na kojemu se biraju osnovne značajke aplikacije. Sve od navedenog u izborniku može se naknadno izmijeniti.



Slika 18 – kreiranje aplikacije

Access Control nam omogućuje da razdvojimo korisnike na administratore i goste, tako možemo administratorima dopustiti da upravljaju trgovinom, a gostima ograničiti mogućnosti samo na kupovanje.

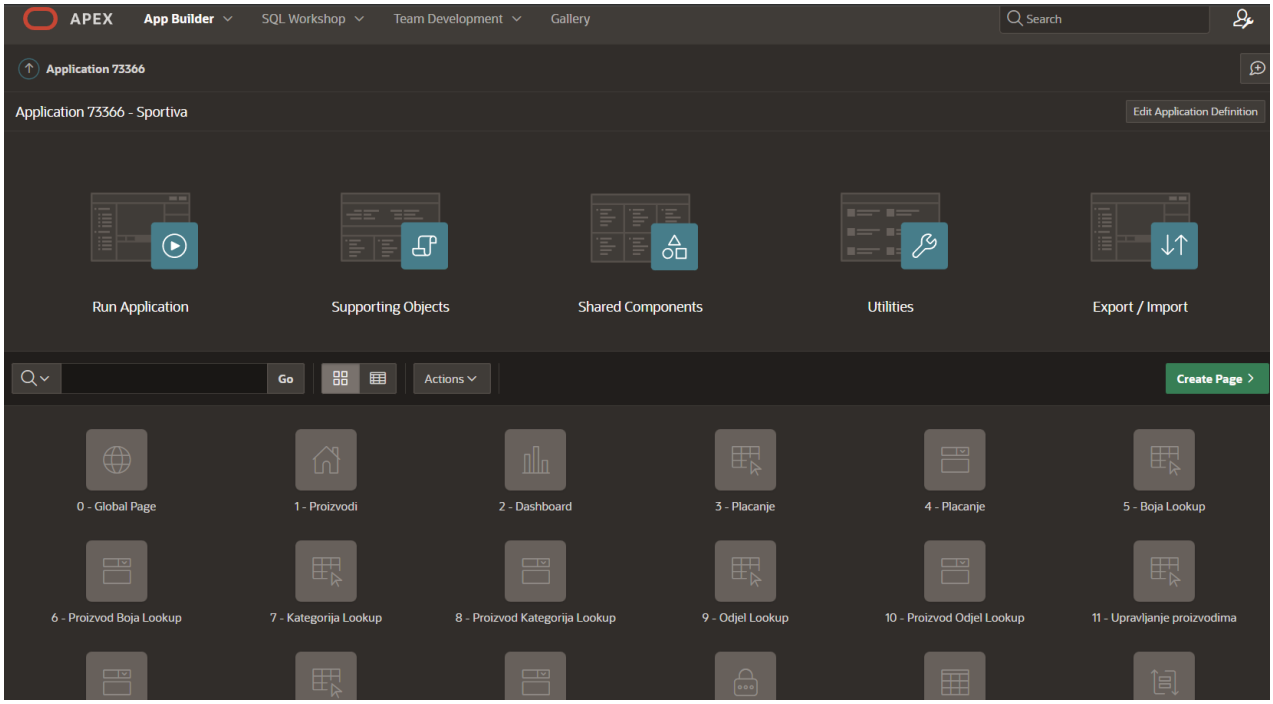
Za izradu maloprodaje dodat ćemo stranicu Proizvodi na kojoj će biti izlistane vrijednosti tablice, a zatim ćemo urediti stranicu da korisnik može dodavati proizvode u košaricu.



Slika 19 – dodavanje nove stranice

Oracle APEX nudi nam mnogo šablona pomoću kojih aplikacije možemo kreirati brzo i jednostavno. U kontekstu maloprodaje, neke šablone su savršene za kupce, a neke za administratore. *Faceted Search* šablona pomoći će nam pri filtriranju proizvoda pa ćemo ju odabrati za stranicu kupovine. Prilikom kreiranja ove stranice potrebno je odabrati atribute tablice po kojima želimo da korisnici filtriraju proizvode.

Za stranice administracije koristit ćemo šablonu *Interactive Report*. Kako bi stranica bila samo za administrativne svrhe potrebno je odabrati opciju *Advanced>Set as Administration Page*. Kada smo odabrali sve tablice koje za sad želimo možemo kliknuti na *Create Application*.

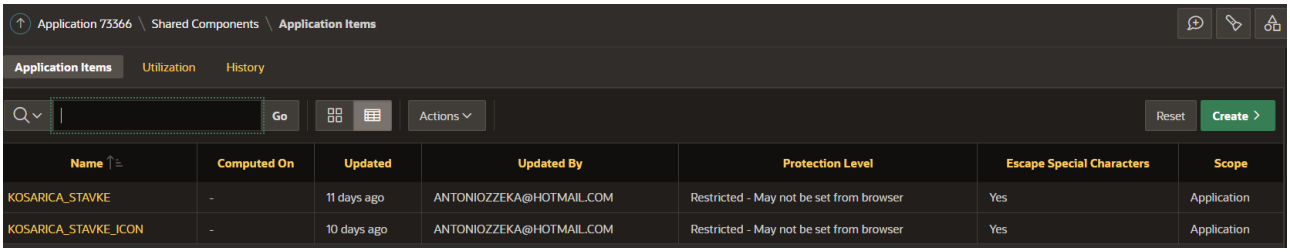


Slika 20 – korisničko sučelje za kreiranje i pregled stranica aplikacije

4.3 Ostale stranice

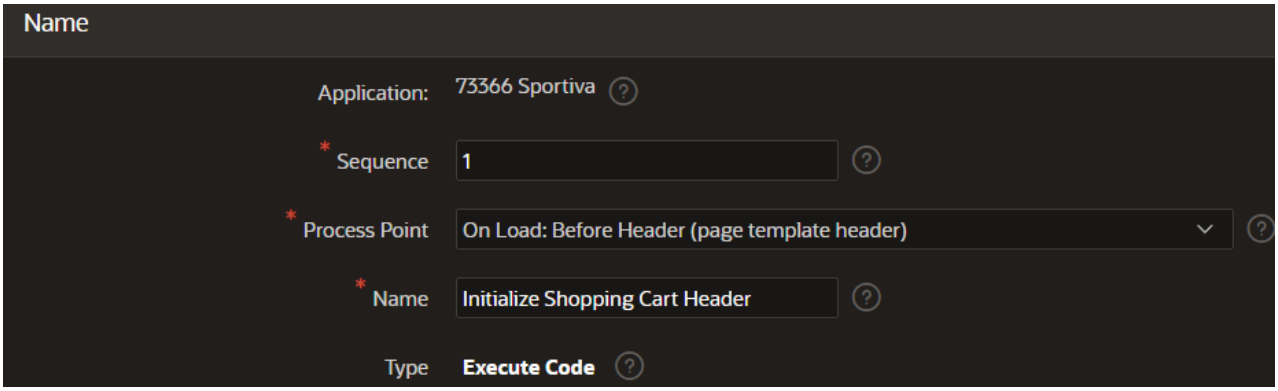
Kako bi aplikacija radila kao što je zamišljeno, morat ćemo kreirati još neke stranice koje nemaju svoje tablice u bazi podataka. To su: stranica za informacije o narudžbi, košarica i stranica za dodavanje proizvoda u košaricu. Stranice kreiramo klikom na *Create Page*, a sve stranice možemo odabrati da za početak budu prazne. Za stranicu za dodavanje proizvoda u košaricu, prilikom kreiranja stranice, odabiremo opciju *Modal Dialog*. Opcija *Modal Dialog* omogućuje nam da se stranica prikaže kao dodatan prozor na drugoj stranici. Tako ćemo omogućiti da prilikom odabira proizvoda otvorimo novi prozor s detaljima odabranog proizvoda gdje ćemo imati opciju da predmet dodamo u košaricu.

Košarica će nam se nalaziti na zaglavlju stranice i želimo ju ažurirati svaki put kada korisnik doda novi proizvod u košaricu. Isto tako, svaki puta kada je proizvod obrisan iz košarice želimo da je to vidljivo na ikonici košarice. U svrhu ažuriranja stanja košarice napravit ćemo dva predmeta (slika) koja će primati vrijednost stanja košarice. Količinu predmeta u košarici pratimo pomoću ranije stvorenog paketa MANAGE_ORDERS, a predmete koristimo kako bi korisnik mogao vidjeti stanje košarice kao ikonicu u zaglavlju.



Slika 21 – predmeti zaglavlja

Nakon što smo kreirali predmete moramo kreirati i proces koji će predmetima ažurirati vrijednosti:



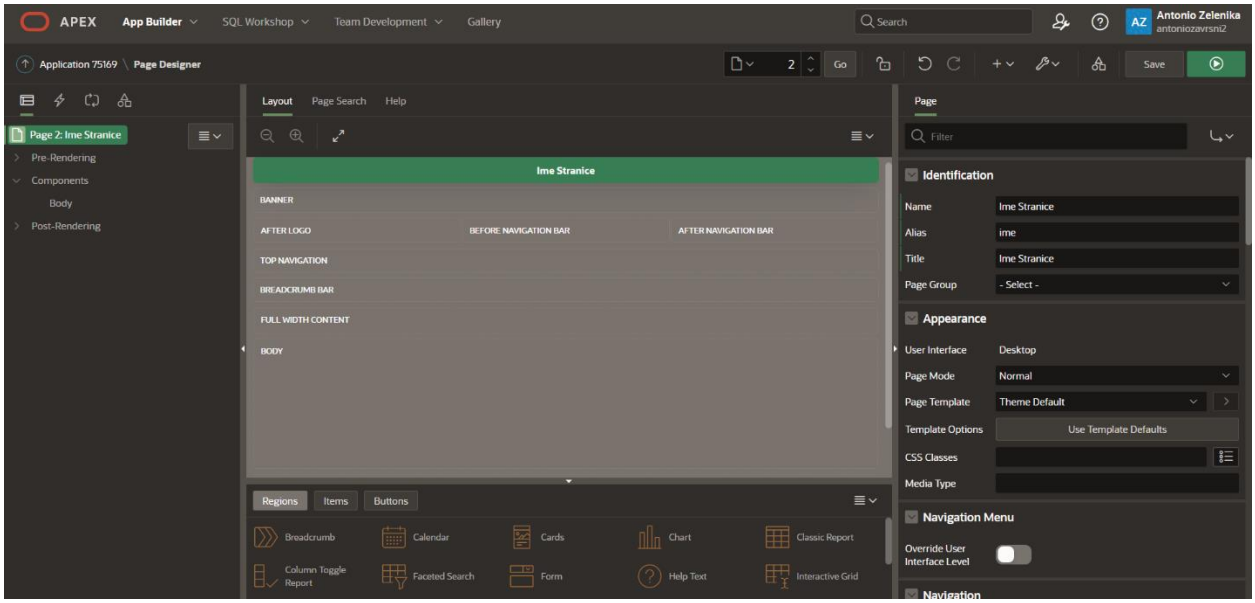
Slika 22 – ažuriranje ikone za košaricu

Navedeni proces izvršit će sljedeći kod:

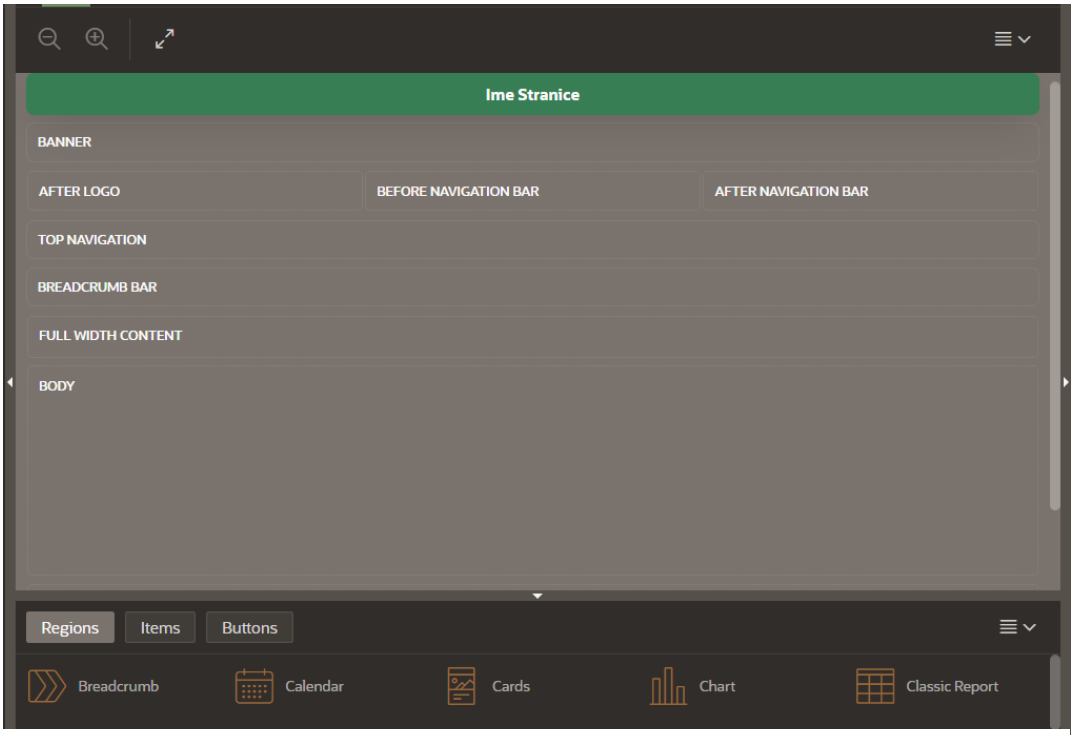
```
DECLARE
  l_cnt NUMBER := manage_orders.get_quantity;
BEGIN
  IF l_cnt > 0 THEN
    :KOSARICA_STAVKE := l_cnt;
    :KOSARICA_STAVKE_ICON := 'fa-cart-full';
  ELSE
    :KOSARICA_STAVKE := '';
    :KOSARICA_STAVKE_ICON := 'fa-cart-empty';
  END IF;
END;
```

4.4 Uređivanje stranica

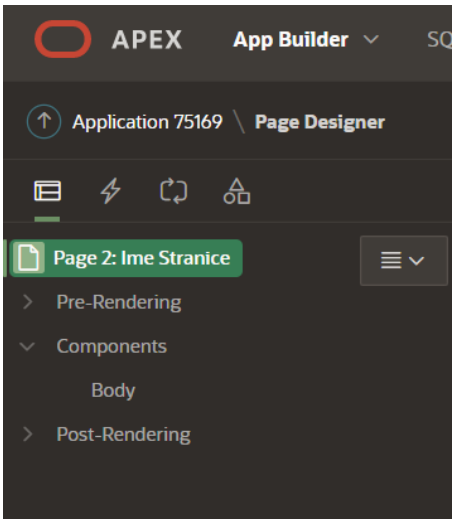
Budući da su nove kreirane stranice prazne, a stranice kreirane prilikom kreiranja aplikacije neuređeno, moramo se upoznati s korisničkim sučeljem za uređivanje stranica i izmjenjivanja njihovog sadržaja (slika 23-26).



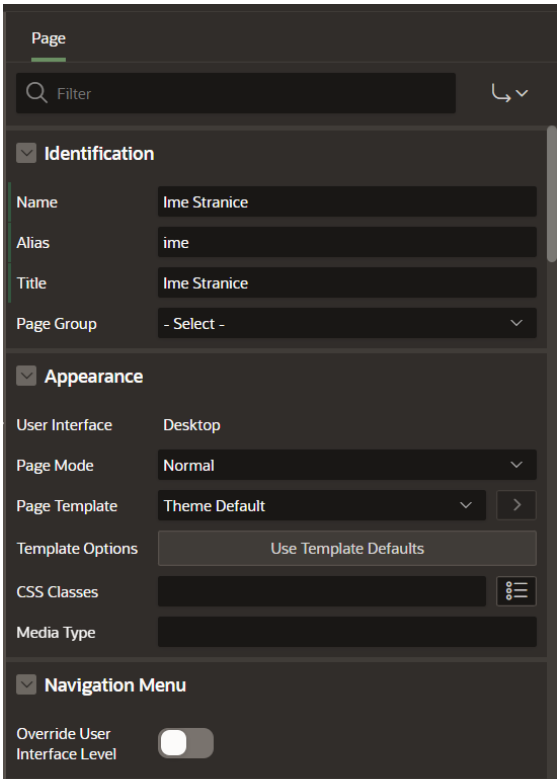
Slika 23 – korisničko sučelje za uređivanje stranica



Slika 24 - korisničko sučelje za uređivanje stranica, središnji dio



Slika 25 - korisničko sučelje za uređivanje stranica, lijevi dio

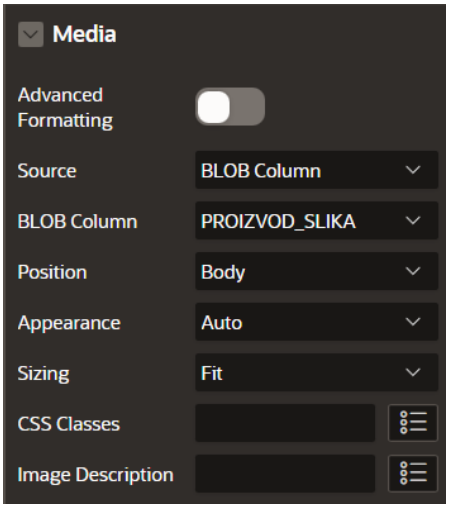


Slika 26 - korisničko sučelje za uređivanje stranica, desni dio

Na slici 23 možemo vidjeti korisničko sučelje za uređivanje stranica. Lijevo (slika 25) se nalaze komponente na koje je stranica podijeljena, a u sredini (slika 24) je grafički prikaz tih komponenti koji odgovara rasporedu sadržaja na stranici koju korisnik vidi. Desno (slika 26) se nalaze postavke i atributi tih komponenti, a dolje se nalaze opcije za prikazivanje željenog sadržaja, interakciju sa sadržajem i gumbi kojima možemo dodijeliti funkcije kao što su npr. dodavanje predmeta u košaricu.

4.4.1 Proizvodi

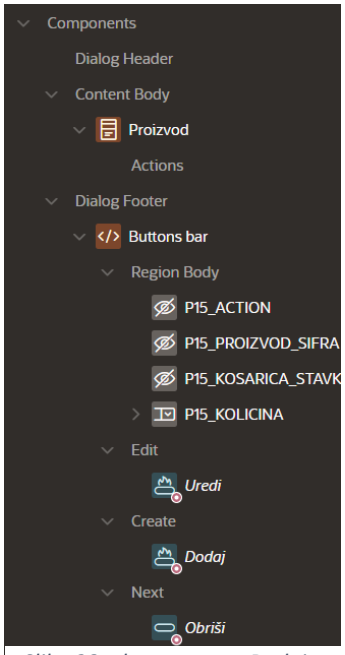
Na stranici *Proizvodi* želimo prikazati sve proizvode iz tablice kojoj pripadaju. Prilikom kreiranja stranice *Proizvodi* odabrali smo šablonu *Faceted Search* koja ima ugrađene opcije pretraživanja u stranicu. Uz automatski kreirane opcije pretraživanja, možemo i urediti upit pomoću kojega se pretraživanje obavlja. Kako bi korisnik mogao vidjeti slike proizvoda moramo odabrati izvor medija. Slike u bazu podataka pohranjujemo kao tip BLOB i zato odabiremo postavke prikazane na slici 27.



Slika 27 – postavke slika

4.4.2 Dodaj u košaricu

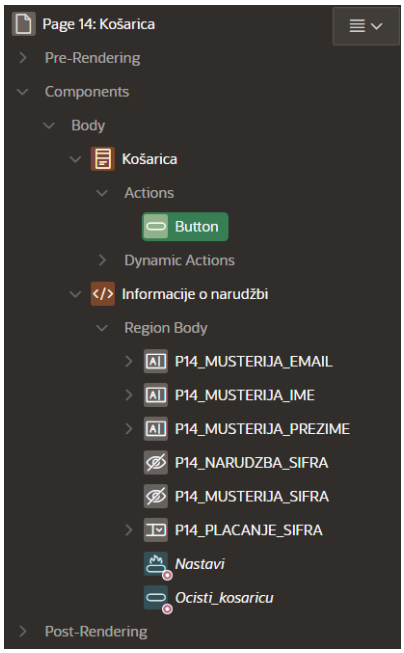
Kako bismo mogli dodati proizvode u košaricu moramo pohraniti podatke koje korisnik unosi. Ti podaci su: količina odabranog proizvoda, šifra odabranog proizvoda, broj stavki u košarici i radnja. Radnja se odnosi na radnju koju korisnik odabere gumbom. Korisnik može proizvod dodati u košaricu, promijeniti količinu koju želi dodati u košaricu (kada je predmet već u košarici) i obrisati predmet iz košarice. Da bismo to sve mogli omogućiti potrebna su nam četiri odjeljka, od koja su tri skrivena. Samo je količina vidljiva i to kao tip *select list* tako da korisnik odabere količinu iz padajućeg izbornika. Također, potrebna su nam i tri gumba: Uredi, Obriši i Dodaj. Slika 28 prikazuje kako te komponente izgledaju.



Slika 28 – komponente Dodaj u košaricu

4.4.3 Košarica

Korisnik u košarici ima uvid u odabrane proizvode i polja koja mora popuniti kako bi izvršio narudžbu. Kupac mora unijeti email, ime i prezime, te odabrati način plaćanja kako bi mogao dovršiti kupnju. Način plaćanja je padajući izbornik, a uz to imamo i dva elementa koja korisnik ne vidi ali su nam potrebni za bazu podataka. Ti elementi su: šifra mušterije i šifra narudžbe. Također, korisnik ima opciju uređivanja narudžbe, odnosno promijeniti količinu proizvoda ili obrisati proizvod. Za to nam je potreban gumb koji otvara stranicu *Dodaj u košaricu* koja se otvara preko stranice na kojoj se korisnik nalazi. Još su nam potrebna dva gumba; jedan za brisanje svih predmeta iz košarice i jedan za potvrdu kupnje. Slika 29 prikazuje kako te komponente izgledaju.



Slika 29 – komponente Košarica

4.4.4 Informacije o narudžbi

Sve informacije o narudžbi koje korisnik vidi nalaze se u komponentama Detalji narudžbe i Stavke, a za to nam služe SQL upiti. Upite upisujemo pod *Source* regije u koju želimo upisati podatke iz tih upita.

Budući da imamo dvije regije u koje želimo upisati podatke stvaramo dva upita:

Detalji narudžbe

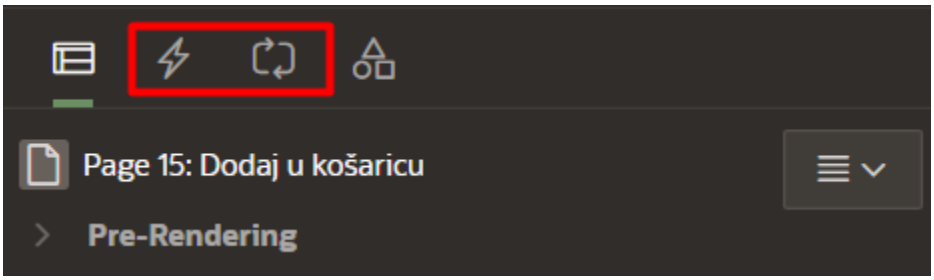
```
SELECT n.narudzba_sifra,
   n.narudzba_vrijeme,
   n.musterija_sifra,
   n.narudzba_status,
   n.placanje_sifra,
   (SELECT Sum(jedinicna_cijena * kolicina)
      FROM   stavke s
      WHERE  s.narudzba_sifra = n.narudzba_sifra)
ukupno
FROM   narudzbe n
WHERE  narudzba_sifra = :P13_NARUDZBA
```

Stavke

```
SELECT  s.stavka_sifra Stavka,
       p.proizvod_naziv Proizvod,
       s.jedinicna_cijena,
       s.kolicina,
       ( s.jedinicna_cijena * s.kolicina ) Ukupno,
       p.proizvod_slika
FROM   stavke s,
   proizvodi p
WHERE  p.proizvod_sifra = s.proizvod_sifra
```

4.5 Povezivanje stranica

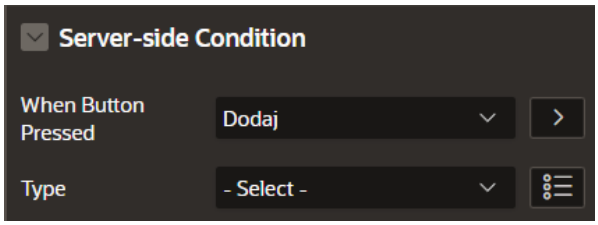
Za sada smo napravili stranice koje su nam potrebne da bi korisnik mogao obaviti kupnju. Sada te stranice moramo povezati procesima kako bi se podaci mogli prenositi iz stranice u stranicu da bi korisnik mogao dobivati povratnu informaciju o svojim odabirima. Obavljanje procesa definirano je zadanim događajima, npr. kada kupac doda predmet u košaricu.



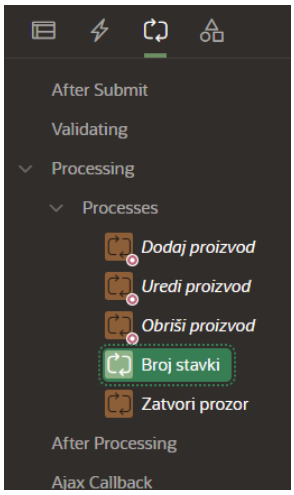
Slika 30 – Procesi i dinamičke radnje

4.5.1 Procesi

Na stranici *Dodaj u košaricu* potrebno je napraviti po jedan proces za svaki gumb, jedan proces da prati broj stavki dodanih u košaricu i jedan proces za zatvaranje prozora *Dodaj u košaricu*, odnosno potrebno nam je 5 procesa. Za procese *Dodaj proizvod*, *Uredi proizvod* i *Obriši proizvod* važno je da ih povežemo s odgovarajućim gumbom. Na slici 31 je prikazan uvjet za proces dodavanja proizvoda.



Slika 31 – Uvjet od strane servera



Slika 32 - procesi

BROJ STAVKI

```
BEGIN
    :P15_KOSARICA_STAVKE := manage_orders.get_quantity;
END;
```

UREDI PROIZVOD

```
BEGIN
    IF manage_orders.product_exists(p_product => :P15_PROIZVOD_SIFRA) > 0 THEN
        manage_orders.remove_product(p_product => :P15_PROIZVOD_SIFRA);
        manage_orders.add_product (p_product  => :P15_PROIZVOD_SIFRA,
                                   p_quantity => :P15_KOLICINA);
    END IF;
    :P15_ACTION := 'UREDI';
END;
```

DODAJ PROIZVOD

```
BEGIN
    IF manage_orders.product_exists(p_product => :P15_PROIZVOD_SIFRA) = 0 THEN
        manage_orders.add_product (p_product  => :P15_PROIZVOD_SIFRA,
                                   p_quantity => :P15_KOLICINA);
    END IF;
    :P15_ACTION := 'DODAJ';
END;
```

OBRISI PROIZVOD

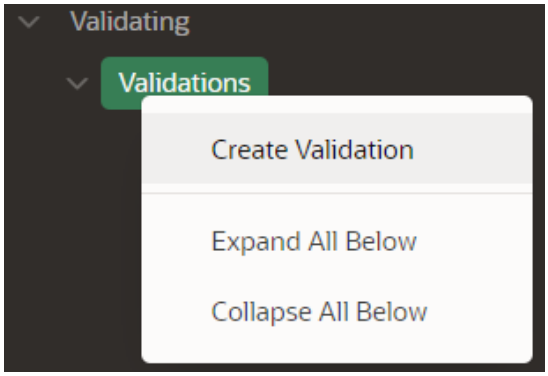
```
BEGIN
    IF manage_orders.product_exists(p_product => :P15_PROIZVOD_SIFRA) > 0 THEN
        manage_orders.remove_product(p_product => :P15_PROIZVOD_SIFRA);
    END IF;
    :P15_ACTION := 'OBRIŠI';
```

Slika 33- PL/SQL Dodaj u košaricu

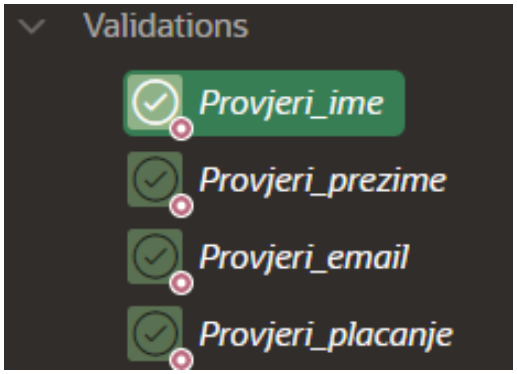
Tip procesa za zatvaranje prozora je *Close Dialog*, a svi ostali procesi na ovoj stranici su tipa *Execute Code*. Kreirani procesi prikazani su na slici. Na slici 33 vidimo kodove za procese stranice *Dodaj u košaricu*.

Osim za prosljeđivanje vrijednosti, procesi se mogu koristiti i za provjeru podataka, npr. jesu li ispunjena sva potrebna polja kako bi se mogla izvršiti narudžba. Na stranici *Košarica* imamo polja u koja kupac mora unijeti ime, prezime, email i odabrati način plaćanja. Kako bi bili sigurni da će kupac unijeti potrebne podatke unosimo provjere. Za osnovnu provjeru unesenih podataka dovoljno je staviti uvjet da se polje razlikuje od nule. Kreiramo provjeru i dajemo joj naziv Provjeri_ime

Zatim pod tip provjere odabiremo opciju *item is NOT NULL* i odabiremo na kolji se predmet odnosi ta provjera, u našem slučaju to je predmet P14_MUSTERIJA_IME. Iste korake ponovimo za prezime, email i način plaćanja. Na slici 35 su prikazane sve provjere.



Slika 34 – kreiranje provjera



Slika 35 - provjere

Na stranici *Košarica* također imamo i dva procesa: proces za završetak narudžbe i proces za brisanje svih proizvoda iz košarice. Kao i procese stranice *Dodaj u košaricu*, ove ćemo zapisati u PL/SQL-u. Prikazani su na slici 36.

Zavrsi_narudzbu

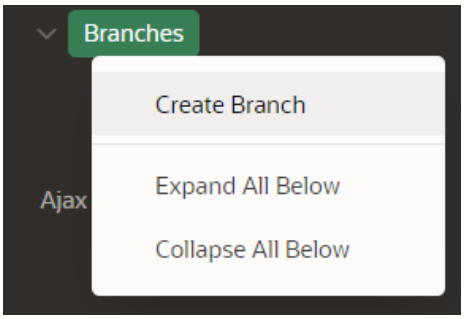
```
1   MANAGE_ORDERS.create_order (
2                               p_customer_firstname => :P14_MUSTERIJA_IME,
3                               p_customer_lastname  => :P14_MUSTERIJA_PREZIME,
4                               p_customer_email     => :P14_MUSTERIJA_EMAIL,
5                               p_payment            => :P14_PLACANJE_SIFRA,
6                               p_order_id           => :P14_NARUDZBA_SIFRA,
7                               p_customer_id        => :P14_MUSTERIJA_SIFRA);
```

isprazni_kosaricu

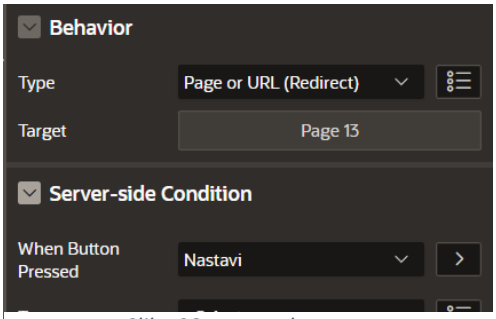
```
1  BEGIN
2      manage_orders.clear_cart;
3  END;
```

Slika 36 – PL/SQL Košarica

Sve što nam je preostalo je preusmjeriti korisnika iz košarice kada stisne određeni gumb. Ako pritisne *Nastavi* i ako je ispunio sva potrebna polja i košarica nije prazna, preusmjerit će se na stranicu Informacije o narudžbi. Ako pritisne na *Isprazni košaricu*, preusmjerit će se nazad na stranicu *Proizvodi*.



Slika 37 – kreiranje grana



Slika 38 – postavke grana

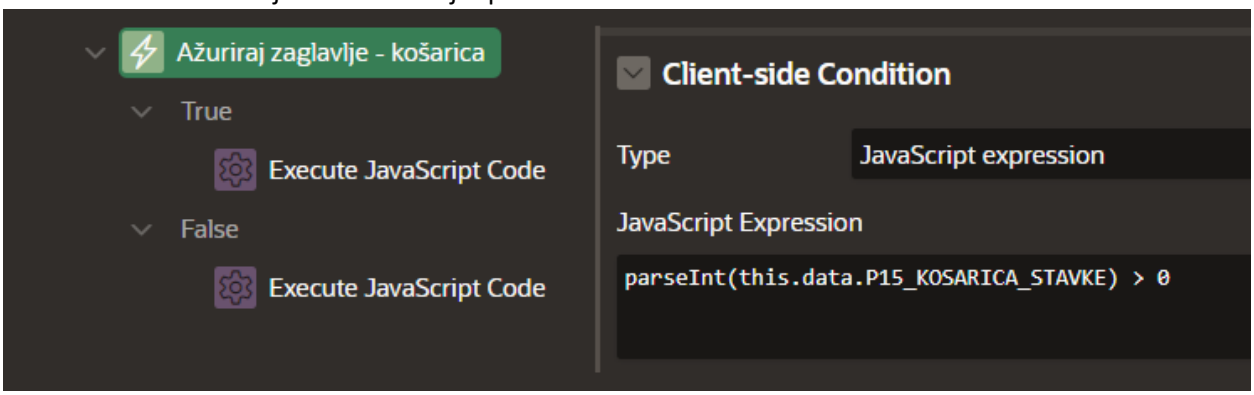
To ćemo postići grananjem. Postavke grananja su vrlo jednostavne, samo odaberemo na koju nas stranicu vodi, a za uvjet postavimo pritisak na gumb. Postavke grananja za nastavak do informacija o narudžbi vidimo na slici 38.

4.5.2 Dinamičke radnje

Da bi kupac dobio povratnu informaciju kada pritisne gumb za određenu radnju koristimo dinamičke radnje. Pomoću dinamičke radnje pokrećemo odabranu radnju tijekom određenih promjena. Za stranicu *Proizvodi* pokrećemo JavasScript kod koji korisniku prikazuje poruku kada doda predmet u košaricu, obriše predmet iz košarice ili kada uredi košaricu. Za slanje povratne informacije koristimo sljedeći kod:

```
var productAction   = this.data.P15_ACTION,
   productQuantity = this.data.P15_KOLICINA,
   productCard$  = apex.jQuery("#message_" + this.data.P15_PROIZVOD_SIFRA);

if (productAction === 'DODAJ') {
   productCard$.text("Proizvoda dodano u košaricu: " + productQuantity + "!" );
} else if (productAction === 'UREDI') {
   productCard$.text("Količina ažurirana u " + productQuantity + "!");
} else if (productAction === 'OBRIŠI') {
   productCard$.text("Obrisano iz košarice!");
}
```

Na isti način ćemo zadati uvjet da se promijeni ikonica košarice te da prikazuje broj proizvoda u košarici. Za ovu dinamičku radnju koristimo uvjet prikazan na slici 39.



Slika 39 – uvjet za košaricu

Ovisno o tome je li dobiveni broj veći od nule (odnosno je li broj proizvoda u košarici veći od nule), pokreće se pripadajući True ili False JavaScript kod. Oba koda prikazana su na slici 40.

Ako je True

```
// Update Badge Text
apex.jQuery(".js-shopping-cart-item .t-Button-badge").text(this.data.P15_KOSARICA_STAVKE);

// Update Icon
apex.jQuery(".js-shopping-cart-item .t-Icon").removeClass('fa-cart-empty').addClass('fa-cart-full');
```

Ako je False

```
// Update Badge Text
apex.jQuery(".js-shopping-cart-item .t-Button-badge").text('');

// Update Icon
apex.jQuery(".js-shopping-cart-item .t-Icon").removeClass('fa-cart-full').addClass('fa-cart-empty');
```

Slika 40 – True/False JS

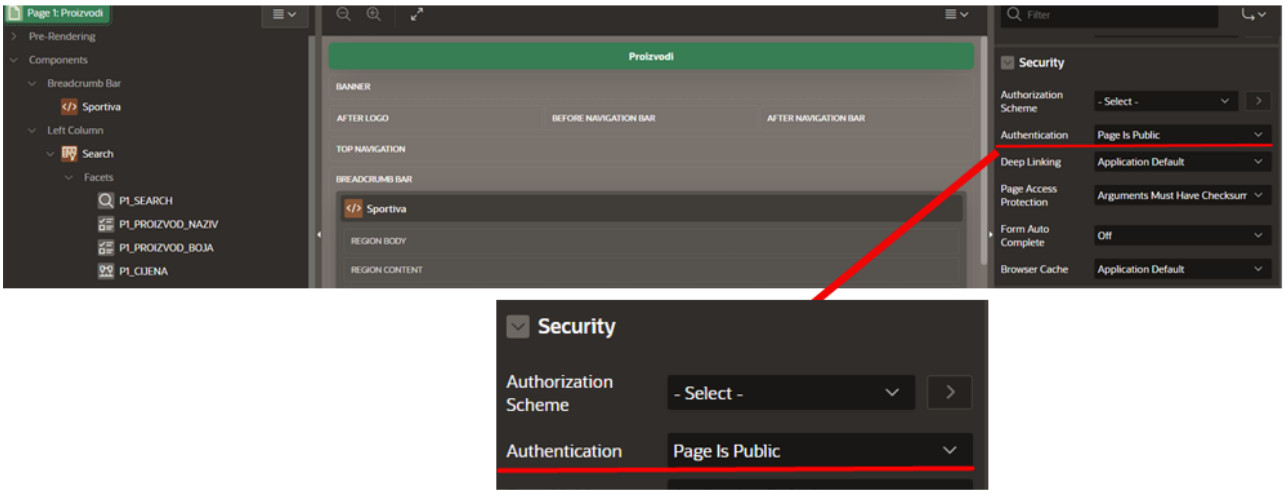
Isti taj uvjet i kod dodan je i na stranicu košarice zato što se i tamo može uređivati i brisati sadržaj košarice.

Sada je sve spremno za kupovinu, odnosno imamo sve potrebne procese, a kupac ima sve potrebne povratne informacije.

4.6 Administracija

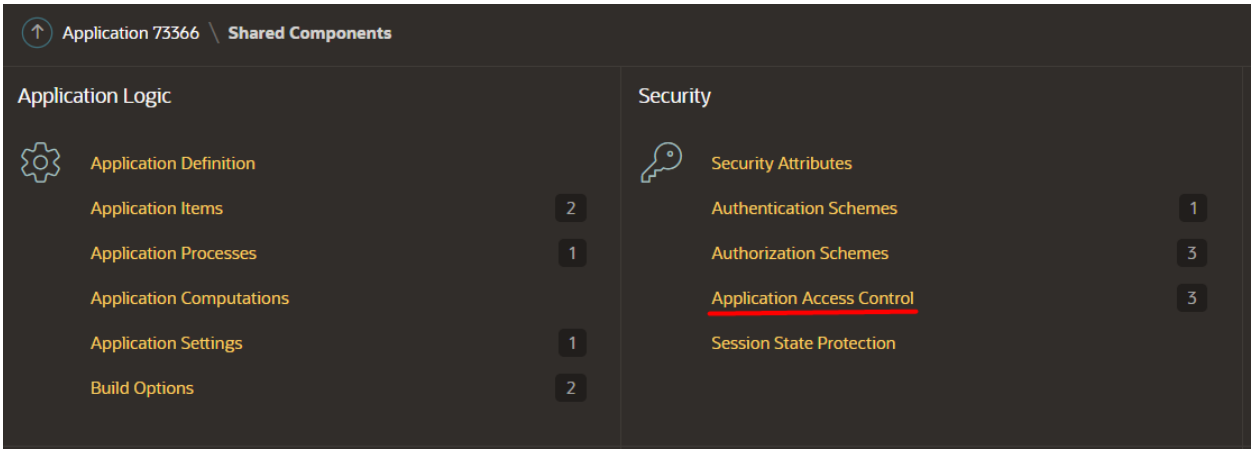
Kako bi mogli razlikovati administratore aplikacije od svih ostalih korisnika sve stranice za kupovinu pretvorit ćemo u javne stranice, a za pristup administraciji bit će potrebna autentifikacija. Ako se korisnik ne prijavi, moći će samo pregledavati i koristiti stranice koje ćemo postaviti kao javne odnosno *Public*. Stranice koje ćemo postaviti da kao javne su: Proizvodi, Dodaj u košaricu, Košarica i Informacije o narudžbi. Na postavkama svake od navedenih stranica pod djelom *Security*, koji se nalazi s

desne strane korisničkog sučelja za uređivanje stranica, pod *Authentication* odabiremo opciju *Page is Public* kao što je prikazano na slici 41.



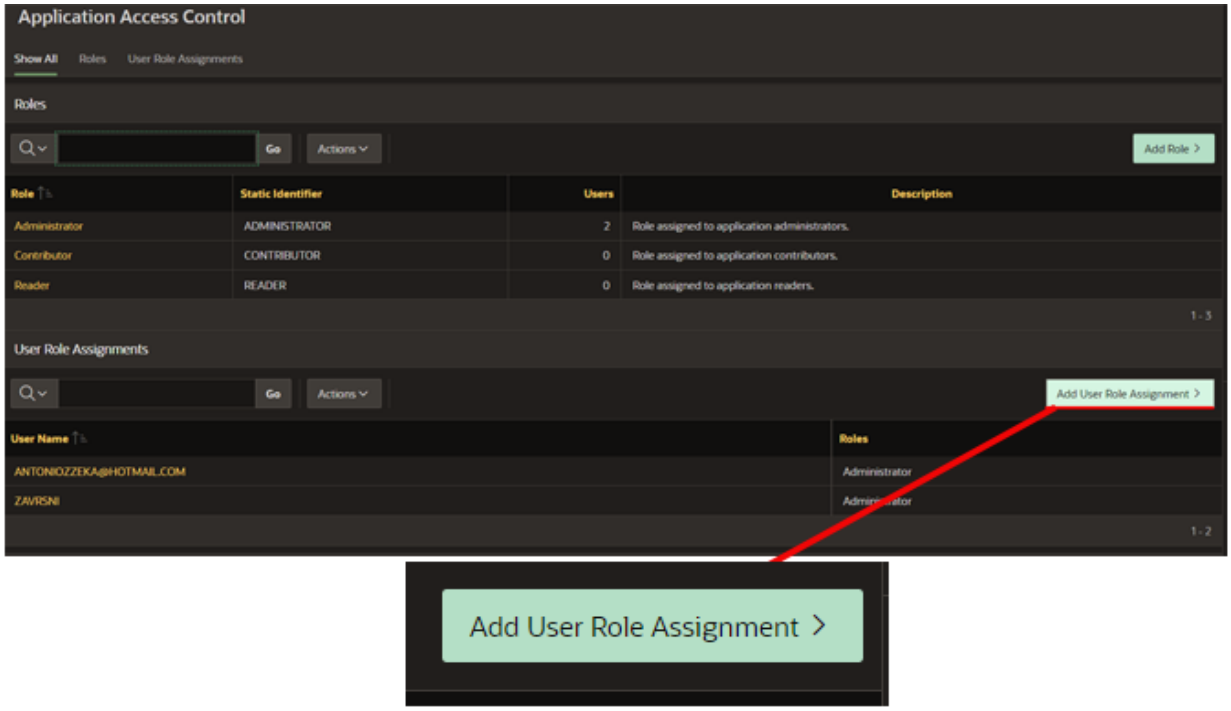
Slika 41 – postavljanje javne stranice

Korisnički račun administratora kreiramo tako da se uputimo na *Application>Shared Components* i odaberemo *Application Access Control* (slika 42)



Slika 42 - Application Access Control

Nakon toga u donjem desnom kutu odaberemo gumb *Add User Role Assignment* (slika 43) i unesemo željene podatke s kojima bi se administrator prijavio u aplikaciju.

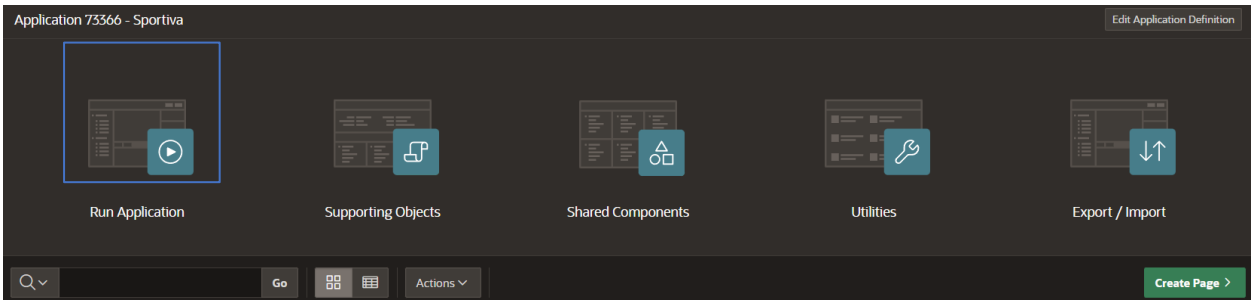


Slika 43 – dodavanje novog korisnika

4.7. Korisničko sučelje i prijava

Aplikaciji pristupamo pomoću linka klikom na *Run Application* (slika 44.) ili putem linka :

https://apex.oracle.com/pls/apex/r/sportiva/sportiva/home?session=112714807737108



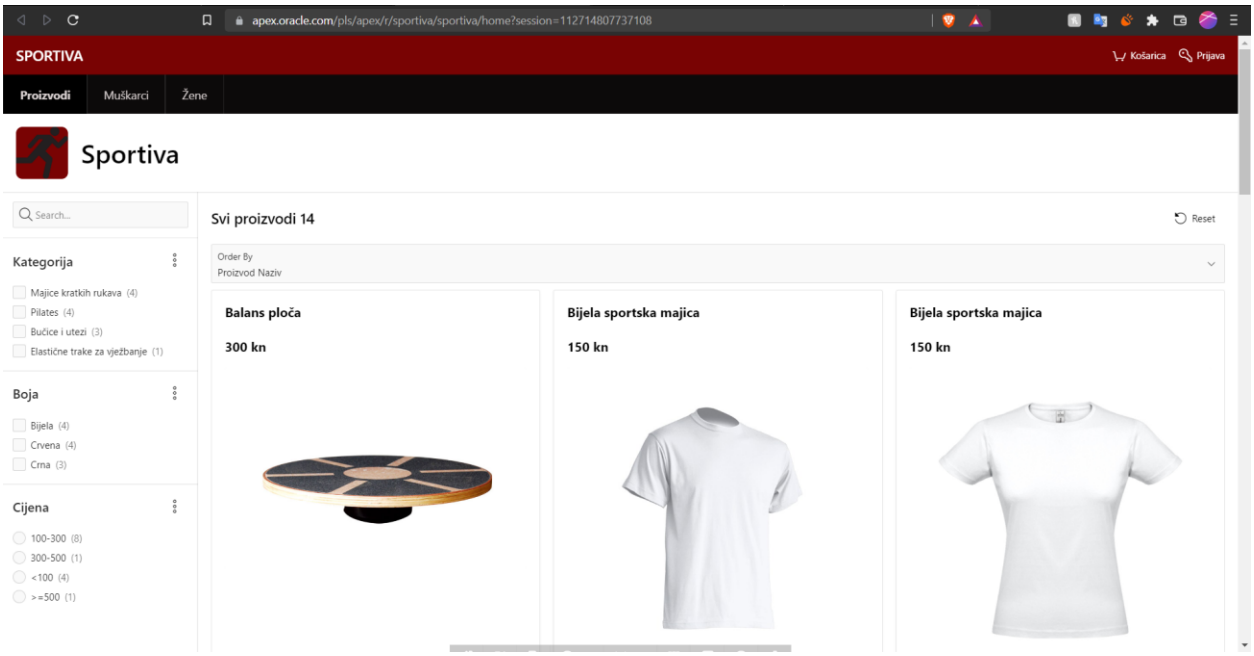
Slika 44 – pokretanje aplikacije

Kada otvorimo aplikaciju koristimo ju kao gost, što znači da možemo dodavati predmete u košaricu i dovršiti narudžbu, dočeka nas početna stranica (slika 50) na kojoj možemo vidjeti sve proizvode. Na lijevoj strani nalaze se filteri prema kojima korisnik može lakše pronaći što ga zanima, a na naslovnoj traci možemo odabrati hoćemo li gledati sve proizvode ili samo muške ili samo ženske. Kako bi mogli vidjeti detalje o narudžbama, proizvodima, kupcima itd. moramo se prijaviti kao administrator. Gumb za prijavu nalazi se u gornjem desnom kutu naslovne trake.

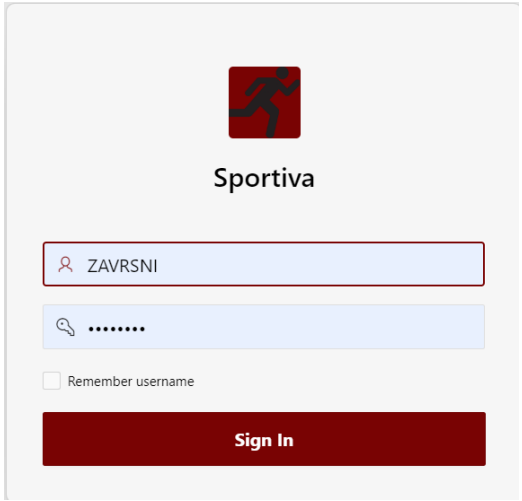
Za prijavu ćemo koristiti slijedeće podatke:

- Korisničko ime: ZAVRSNI
- Lozinka : zavrsni1

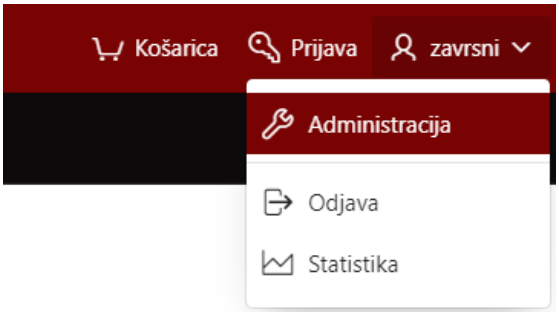
Nakon što smo se prijavili, u gornjem desnom kutu gdje je bio natpis za prijavu sada se nalazi naše korisničko ime. Klikom na korisničko ime imamo opciju za administraciju (slika 47), odnosno dodavanje novih predmeta, uklanjanje predmeta iz prodaje, pregled kupaca i slično. Uz administraciju, ponuđena nam je odjava i statistika.



Slika 45 – početna stranica



Slika 46 – prijava u aplikaciju



Slika 47 – dodatne opcije za administratora

5 Zaključak

Razvojna platforma s niskim udjelom koda pruža mogućnost za brzi razvoj aplikacija. Prednosti „low-code“ platformi su što pojednostavljuje izradu poslovnih aplikacije, time smanjuje potrebu za većom investicijom pri izrade poslovne aplikacije, lakše je educirati potencijalne zaposlenike za održavanje aplikacije i možda najbitnije od svega, omogućuje brzu izradu poslovnih aplikacija.

Tijekom izrade aplikacije odvojeno smo radili dio po dio aplikacije, odnosno stranicu po stranicu, a zatim smo sve povezali procesima i dinamičkim radnjama. Pri svakom koraku izrade aplikacije mogli smo stati i provjeriti da li sve radi kako je zamišljeno, odnosno mogli smo paralelno isprobavati aplikaciju i uređivati ju, a ako bi se pojavila neka greška Oracle APEX bi ju detektirao i prikazao gdje je problem. Brzim isprobavanjem novog sadržaja i jednostavnim rješavanjem grešaka izrada aplikacije napreduje brzo, a sve se temelji na dobro izgrađenoj bazi podataka koju smo izgradili na temelju relacijskog modela dobivenog transformacijom modela entiteti – veze. Sama izrada baze podataka jednostavna je za osobu koja nije upoznata sa SQL-om radi Oracle APEX alata za izradu tablica, iako poznavanje SQL-a pomaže da se to brže odradi.

Kontrola pristupa koristan je alat koji nam je omogućio da podijelimo uloge korisnika na administratora, ako se osoba prijavi kao administrator, i na običnu mušteriju, ako se korisnik ne prijavi i nastavi kao gost.

Stranice smo uređivali na vrlo jednostavan način pomoću korisničkog sučelja za uređivanje stranica. Umetanjem podataka iz tablica, odabiranjem gumba i njihovih funkcija i pozicija bez koda smo uredili stranicu. Omogućili smo korisniku da pretražuje ponudu uz pomoć filtera, da odabere, da dodaje predmete u košaricu, da ukloni neke ili sve predmete iz košarice i da dovrši narudžbu. Administratoru smo omogućili da izmjenjuje predmete u ponudi koje želi, da dodaje nove kategorije, boje i slično, a uz to ima i uvod u statistiku prodaje, odnosno može vidjeti koji su predmeti najprodavaniji, a koji su u najmanjoj potražnji.

Oracle APEX ili Oracle Application Express zasigurno je zaslužio svoje ime jer kreiranje aplikacije čini vrlo jednostavnim i brzim, a uz to, sa svojom opširnom dokumentacijom platforma je pristupačna početnicima. Za izradu aplikacije za potrebe procesa maloprodaje sportske opreme koristio sam relativno osnovne alate Oracle APEX-a, ali sam i dalje zadovoljan funkcionalnošću, a i izgledom stranice.

Sažetak

Izradu aplikacije započinjemo izradom baze podataka. Najprije izradimo model entiteti – veze, zatim taj model prevedemo u relacijski model prema kojemu ćemo izgraditi bazu podataka. Bazu podataka radimo unutar Oracle APEX-a uz pomoć alata za kreiranje baze podataka.

Nakon što smo izgradili bazu podataka još moramo kreirati pakete za upravljanje s narudžbama kako bi korisnik mogao koristiti košaricu. Nakon kreiranja paketa kreće izrada same aplikacije. Prvo kreiramo prazne stranice i svakoj dodijelimo ime i podatke koje će prikazivati. Podatke odabiremo iz baze podataka, ali u ovom koraku korisnik još ne može imati nikakvu interakciju s podacima. Zato stranice međusobno povezujemo s procesima i dinamičkim funkcijama kako bi se informacije, odnosno odabiri korisnika mogli prenositi s jedne stranice na drugu (npr. kada doda predmet u košaricu, taj predmet sada vidimo i na stranici *Košarica* ,

Korisnike dijelimo na goste i administratore i zato za administratore kreiramo korisnički račun. Prijavom na korisnički račun omogućujemo administratoru pristup stranicama koje gost ne može vidjeti.

Ključne riječi: Oracle APEX, relacijska baza podataka, model entiteti – veze, aplikacija

Popis slika

Literatura

[1] Pavlić, Mile, Oblikovanje baza podataka, Odjel za informatiku, Sveučilište u Rijeci, Rijeka, 2011.

[2] https://docs.oracle.com/en/database/oracle/application-express/21.2/index.html (posjećeno: 10. kolovoza 2022.)

[3] https://www.foxinfotech.in/2019/10/how-to-create-a-pl-sql-code-process-in-oracle-apex.html (posjećeno: 10. kolovoza 2022.)

[4] https://oracle.github.io/learning-library/developer-library/ (posjećeno: 10. kolovoza 2022.)

[5] https://www.tutorialspoint.com/dbms/er_diagram_representation.htm (posjećeno: 10. kolovoza 2022.)