Realizacija relacijskog modela podataka u Oracle okruženju

Vučić, Petra

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, University of Split, Faculty of science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:166:389590

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-04**

Repository / Repozitorij:

Repository of Faculty of Science

zir.nsk.hr

UNIVERSITY OF SPLIT

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

ZAVRŠNI RAD

REALIZACIJA RELACIJSKOG MODELA PODATAKA U ORACLE OKRUŽENJU

Petra Vučić

Split, rujan 2022.

Temeljna dokumentacijska kartica

Završni rad

Sveučilište u Splitu
Prirodoslovno-matematički fakultet
Odjel za informatiku
Ruđera Boškovića 33, 21000 Split, Hrvatska

REALIZACIJA RELACIJSKOG MODELA PODATAKA U ORACLE OKRUŽENJU

Petra Vučić

SAŽETAK

Cilj ovog završnog rada bio je osmisliti, te zatim realizirati relacijsku bazu podataka koja sadrži sve bitne informacije koje se vežu za sajam brodova. Realizacija i kreiranje ove baze podataka omogućeno je uz Oracle alat, a osim kreiranja logičkog i relacijskog modela baze u Oracle SQL Data Modeleru, u Oracle SQL Developeru je kreiran korisnik, kao i tablice, okidači i sekvence, te naposljetku je kreirana i web aplikacija pomoću alata Oracle Application Express. Svrha web aplikacije jest prikaz manipuliranja podacima, odnosno kreiranja, ažuriranja, brisanja i pretraživanja podataka.

Rad je pohranjen u knjižnici Prirodoslovno-matematičkog fakulteta u Splitu

Ključne riječi: relacijski model, baza podataka, Oracle, entitet, Apex

Rad sadrži: 48 stranica, 39 grafičkih prikaza i 9 literaturnih navoda.

Izvornik je na hrvatskom jeziku.

Mentor: **Doc. Dr. sc. Monika Mladenović**, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

Ocjenjivači: **Doc. Dr. sc. Monika Mladenović**, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

Doc. Dr. sc. Goran Zaharija, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

Doc. Dr. sc. Divna Krpan, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

Rad prihvaćen: rujan 2022.

Basic documentation card

Thesis

University of Split
Faculty of Science
Deparment of informatics
Ruđera Boškovića 33, 21000 Split, Croatia

IMPLEMENTATION OF THE RELATIONAL DATA MODEL IN THE ORACLE ENVIRONMENT

Petra Vučić

ABSTRACT

The goal of this final work was to design and realize a relational database containing all important informations related to the boat fair. Realization and creation of this database was made possible with Oracle tool and in addition to creating a logical and relational model of the database in Oracle SQL Data Modeler. User was created in Oracle SQL Developer as well as tables, triggers and sequences and finally a web application was created using the Oracle Application Expess tool. The purpose of the web application is to display data manipulation, i.e. creating, updating, deleting and searching data.

Thesis deposited in library of Faculty of science, University of Split

Keywords: relational model, database, Oracle, entity, Apex.

Thesis consists of: 48 pages, 39 figures i 9 references

Original language: Croatian

Mentor: **Monika Mladenović**, , **Ph.D.** *Assistant Professor of Faculty of Science, University of Split*

Reviewers: **Monika Mladenović**, , **Ph.D.** *Assistant Professor of Faculty of Science, University of Split*

Goran Zaharija, **Ph.D.** *Assistant Professor of Faculty of Science, University of Split*

Divna Krpan, **Ph.D.** *Assistant Professor of Faculty of Science, University of Split*

Thesis accepted: September 2022.

IZJAVA

kojom izjavljujem s punom materijalnom i moralnom odgovornošću da sam završni rad s naslovom „Realizacija relacijskog modela podataka u Oracle okruženju“ izradila samostalno pod voditeljstvom Doc. Dr. sc. Monike Mladenović. U radu sam primijenila metodologiju znanstveno-istraživačkog rada i koristila literaturu koja je navedena na kraju završnog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući navela u završnom radu na uobičajen, standardan način citirala sam i povezivala s fusnotama s korištenim bibliografskim jedinicama. Rad je pisan u duhu hrvatskog jezika.

Studentica

Petra Vučić

ZAHVALA

Ovim putem želim se zahvaliti svojoj mentorici Doc. Dr. sc. Moniki Mladenović na prenesenom znanju i savjetima tijekom školovanja i tijekom izrade završnog rada.

Hvala mojim prijateljima i kolegama koji su uljepšali moje studiranje, popravili svaki loš dan smijehom i bili moja potpora i motivacija.

Hvala svim profesorima na prenesenom znanju i životnim savjetima.

Najveće hvala mojim roditeljima, sestri i Petru na potpori, savjetima, strpljenju, povjerenju i razumijevanju protekle tri godine.

Sadržaj

1. Uvod 1
2. Relacijske baze podataka 3
2.1. Povijest Oracle platforme 4
2.2. Oracle 5
2.3. Razvojni ciklus baze podataka 5
2.4. Objekti baze podataka 7
2.4.1. Tablica (table) 7
2.4.2. Stupac (Column) 7
2.4.3. Ograničenja (Constraints) 8
2.4.4. Sekvenca (Sequence) 8
2.4.5. Okidač (Trigger) 8
2.4.6. Pogled (View i materialized view) 8
3. Izrada baze podataka 9
3.1. Entiteti 9
3.2. Relacije 13
3.2.1. Relacija jedan na jedan 13
3.2.2. Relacija jedan na više 14
3.2.3. Relacija više na više 14
3.3. Logički model 16
3.4. Relacijski model 17
3.5. Izrada baze podataka prema modelu 18
3.5.1. Ograničenja (eng. Constraints) 19
3.6. Upiti na bazu podataka 25
3.6.1. Unos podataka 25
3.6.2. Ažuriranje podataka 26

3.6.3. Brisanje podataka 26
3.6.4. Odabir podataka 27
3.7. APEX 28
3.7.1. Application builder 30
3.7.2. Izvještaji i forme 36
3.7.3. Povezani podaci 38
Zaključak 40
Literatura 41
Popis slika 42
Skraćenice 44

1. Uvod

Računala i baze podataka usko su povezani, naime prvi programski jezik bio je „Cobol“ i samim tim programiranje je postalo temelj stvaranja poslovnih sustava. Sustavi su trebali negdje pohraniti svoje podatke pa su programeri kako bi olakšali svakodnevne radnje za tu svrhu osmislili baze podataka. Kasnih 1960-ih kada je uporaba računala postala isplativija opcija krenule su se koristiti baze podataka na računalu. Sustavi koji su se tada koristili razvili su se iz datotečnih sustava, te je 1960-ih godina održavanje baza podataka je bilo skupo iz razloga što su se iste podatke spremali različiti programi čime se gubio prostor i resursi. 1970-te godine ujedno nazivamo i „Erom nerelacijskih baza podataka“. U tom periodu baza podataka pruža integriranu i strukturiranu zbirku pohranjenih podataka koje mogu koristiti ili dijeliti aplikacijski sustavi.

Baze podataka u računarstvu definiramo kao organiziranu zbirku strukturiranih podataka/informacija koje su obično pohranjene u računalnom sustavu. Podaci koji se nalaze unutar baza podataka obično su modelirani u stupcima i redovima kako bi obrada i upiti bili učinkoviti, a sami pristup podacima lakši, te da bi se podatke moglo lakše kontrolirati, korigirati, ažurirati i organizirati. Glavna svrha baze podataka, pogotovo u današnjim vremenima kad postoji jako puno podataka i informacija jest pomoći ljudima da prate stvari – kako one različite, tako i povezane stvari. Baze podataka su danas ključ svakog poslovanja, te nam olakšavaju svakodnevnicu samim time što mnoge usluge koje koristimo kao što su web stranice, znanstveni radovi i ostali mogući su zahvaljujući upravo bazama podataka.

Sami podaci su sirove činjenice koje čine sastavne dijelove informacija, dok je baza podataka skup informacija i sredstvo za manipuliranje podacima na koristan način koji mora omogućiti pravilno pohranjivanje velikih količina podataka. Desetljećima se razvijao softver koji je danas zaslužan za izvanredno funkcioniranje baza podataka. Trud, znanje i tehnologija su nam pružili softver koji nazivamo sustav za upravljanjem bazom podataka (DBMS), te koji nam pruža kvalitetno upravljanje velikim brojem podataka kao i samo trajanje tih podataka dugi vremenski period. Sustav za upravljanje bazom podataka (DBMS) zadužen je za sve operacije poput kreiranja strukture, administraciju sustava, brisanje, promjena i dohvat podataka i ostale operacije. Sustavi za upravljanje bazama

podataka kategoriziraju se većinom prema samom modelu podataka pa tako razlikujemo relacijski model koji se najčešće pojavljuje, stariji sustavi u koje spada hijerarhijski model i mrežni model, dok su noviji sustavi semantički model i objektno orijentirani model. Relacijski model najčešće se koristi jer prikazuje informacije smisleno povezivanjem tablica.

Podaci su krvotok mnogih aplikacija koje pokreću naš svijet, međutim stalno rastuće brdo podataka možda stvori problem kako sve te iste podatke prikupiti, pohraniti i razvrstati.

Cilj rada je upoznavanje sa relacijskom bazom podataka, Oracle platformom i izrada web aplikacije na Oracle platformi.

Baze podataka su žila kucavica svake današnje aplikacije ili platforme koju koristite. Od svakodnevnih primjera kao što je odlazak u školu i korištenje E-dnevnika koji povlači podatke iz baze podataka pa sve do kompliciranih bankovnih sustava. Zbog važnosti baza podataka u svakodnevnom životu odlučila sam iskoristiti dosad stečeno znanje, te napraviti bazu podataka za sajam brodova. Baza podataka pružiti će podatke o sajmu brodova i svim informacijama tog sajma kao što su: koji brodovi će biti izloženi na sajmu, tko je organizator, tko je sponzor, mjesto i vrijeme održavanja sajma, tvrtke koje će predstaviti svoje brodove na sajmu, broj posjetitelja, broj kupaca, osnovni podaci izlagača brodova, računi kupnje karata, te računi kupnje brodova.

Sajmovi predstavljaju veoma važan element prodaje i plasmana proizvoda jer su temelj za predstavljanje proizvoda, ali i mogućnost ispitivanja samog tržišta, kao i novih tehnologija. Sajmovi su također važni zbog usporedbe sa konkurencijom, te rezultat tome unaprjeđenju proizvoda naspram konkurencije. Sajmovi predstavljaju važan oblik promidžbe, a smatra se da su se prvi sajmovi pojavili u antičkom Rimu pri čemu se razvija svojevrsna industrija sajmova koja je slična kasnijim srednjovjekovnim sajmovima.

2. Relacijske baze podataka

Ted Codd 1970. objavio je svoj poznati rad koji je bitno utjecao i promijenio dotadašnje poimanje baza podataka. Do tada programeri su bili zaduženi i za strukturu skladištenja podataka dok programeri relacijskih baza podataka ne bi bili zaduženi za strukturu skladištenja podataka. Codd je predložio da sustavi baza podataka korisniku trebaju biti prikazani na način da podaci budu organizirani u tablice koje su povezane relacijama, a u pozadini bi postojala složena struktura koja će omogućiti brz odgovor na razne upite. Tvrtka koja je prva usvojila pojam relacijski model podataka bila je upravo Oracle platforma, te koja je danas sinonim bazama podataka. Relacijski model najrašireniji je model, a najpoznatiji sustavi koji služe za upravljanje relacijskim bazama su:

- Oracle
- SQL Server
- Microsoft Access i drugi.

U ovom radu pobliže će se opisati svojstva i funkcionalnosti platforme Oracle.

Sustav upravljanja baza podataka mora poštivati barem šest od trinaest pravila koje je Edgar Frank Codd objavio kako bi baza bila relacijska, a u nastavku će biti navedena pravila koja trebaju biti zadovoljena kako bi baza bila relacijska.

1. Predstavljanje informacija – podaci trebaju biti jednostavni i dosljedni.
2. Pravilo pristupa – podaci u tablicama trebaju biti logički dostupni.
3. Tretiranje nepoznatih vrijednosti – nepoznata vrijednost tretira se kao *NULL* neovisno o tipu.
4. Dinamički online katalog – rječnik baze gdje su pohranjene informacije o relacijskoj shemi tablica trebaju biti pohranjene kao i svi ostali podaci u tablici.
5. Pravilo sveobuhvatnog jezika – jezik za komuniciranje sa bazom treba podržavati relacijske operatore.
6. Pravilo pogleda – Sustav treba moći ažurirati sve poglede, a sastoje se od *SELECT* naredbe koja može dohvatiti podatke iz jedne ili više tablica.

7. Pravila ažuriranja skupova – Podaci iz relacijske baze podataka mogu biti preuzeti u skupovima podataka, a također operacije unosa, ažuriranja, brisanja trebaju biti podržane za sve skupove podataka.
8. Nezavisnost fizičkih podataka – Aplikacije koje pristupaju podacima u relacijskoj bazi ne smiju biti ovisne o promjenama u fizičkom načinu spremanja podataka.
9. Nezavisnost logičkih podataka – Odnosi između tablica trebaju imati mogućnost ažuriranja i promjene podataka, a da to istovremeno ne utječe na funkcije aplikacija koje su spojene na tablice.
10. Nezavisnost integriteta podataka – Sustav treba brinuti o integritetu baze podataka, ne treba brinuti o vanjskom izgledu aplikacije.
11. Distribuirana nezavisnost – Ako i kada se uvede distribuirana verzija sustava za upravljanjem baza podataka aplikacija treba nastaviti raditi.
12. Pravilo o nenarušavanju – Integritet podataka ne smije biti narušen zbog zaobilaženja pravila.
13. Nulto pravilo – Da bi sustav bio relacijski mora koristiti isključivo relacijske mogućnosti baze podataka kod upravljanja.

Relacijske baze podataka strukturirane od više tablica koje se sastoje od redaka i stupaca u kojima su organizirani podaci. Podaci su obično strukturirani u više tablica koji se obično spajaju primarnim ili stranim ključevima.

Prednosti relacijske baze podataka je mogućnost stvaranja smislenih informacija spajanjem tablica. Referencijalni integritet osigurava da u tablici relacijskog modela ne može postojati vrijednost stranog ključa, a da ne postoji ista vrijednost primarnog ključa u osnovnoj tablici. Referencijalni integritet određuje pravila unosa, brisanja i ažuriranja podataka, a sve u svrhu koherentnosti podataka.

2.1. Povijest Oracle platforme

Nakon nastanka relacijske modela nastala je tvrtka Oracle 1977. godine pod nazivom *Software Development Laboratories*. Ime samog Oracle-a dolazi od projekta za koji je SDL bio zadužen i radio na njemu za američku obavještajnu agenciju CIA, te kojoj je ime bilo „*Project Oracle*“.

2.2. Oracle

Verzija koja je korištena pri realizaciji ovog rada radu jest „*Oracle Database 11g Express Edition*“. Osim već spomenute verzije koja je korištena osim nje prilikom izrade baze podataka korištena su dva alata: *Oracle SQL Developer* i *Oracle SQL Data Modeler*.

Oracle SQL Developer je grafičko korisničko sučelje za Oracle baze podataka, te je automatski dostupan nakon same instalacije *Oracle Database*. Primarna svrha mu je pružanje grafičkog sučelja za upravljanje objektima baza podataka, a osim toga služi i za migracije raznih baza na Oracle, dizajniranje modela podataka.

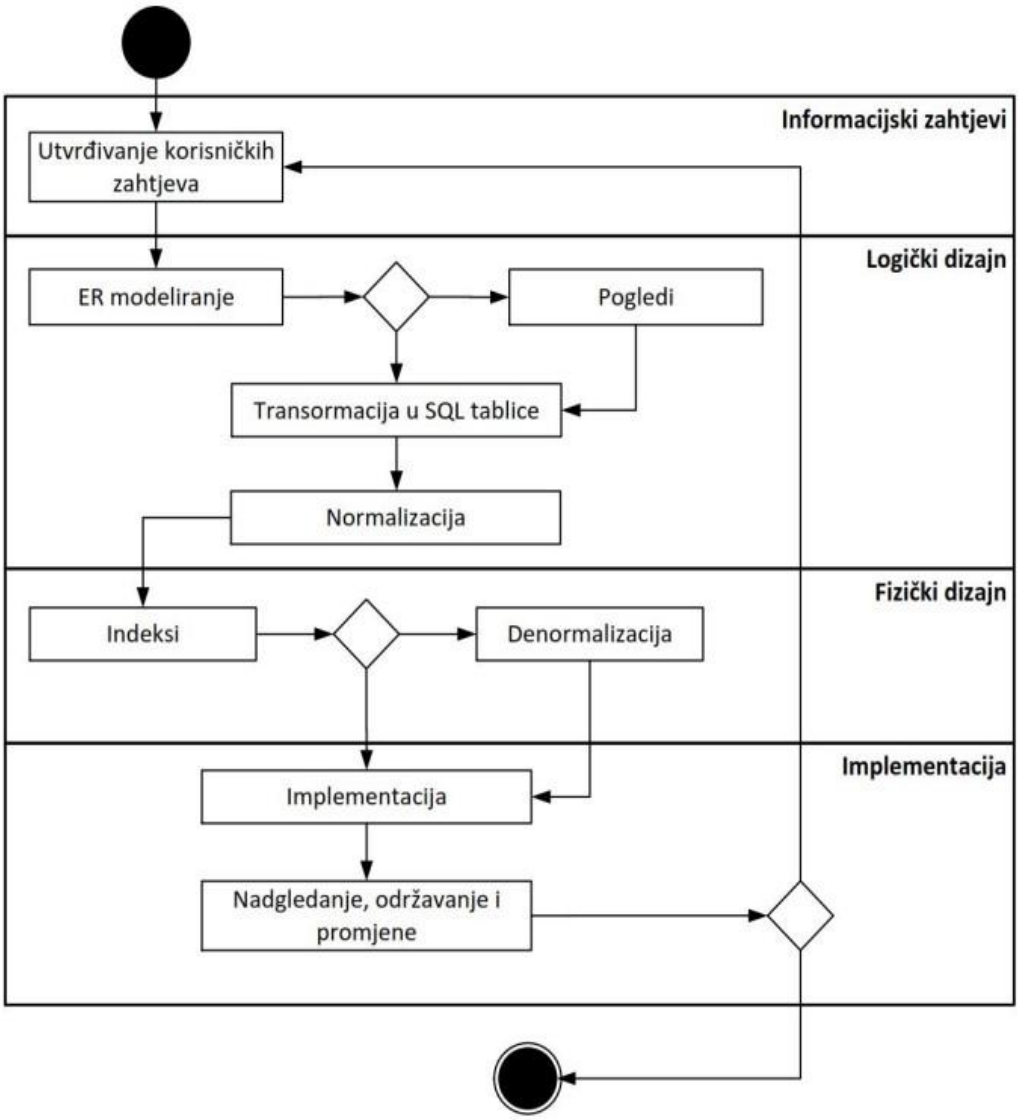
Oracle SQL Data Modeler je alat koji je integriran u *SQL Developer*, te pomoću njega korisnici kreiraju baze podataka, a ujedno je i besplatno grafičko sučelje. Pomoću njega korisnici također mogu kreirati, crtati, otvarati, analizirati relacijske i fizičke modele baza podataka.

2.3. Razvojni ciklus baze podataka

Razvojni ciklus svake baze podataka sastoji se od tri faze:

- projektiranje na konceptualnoj razini,
- projektiranje na logičkoj razini,
- projektiranje na fizičkoj razini.

Samo projektiranje baze podataka predstavlja ključan dio njezinog razvoja zbog toga da se oblikuje pogodna građa baze na osnovi utvrđenih potreba za podacima. Ciklus baze podataka nakon dizajna nastavlja se implementacijom i samim održavanjem baze podataka. Razvojni ciklus baze podataka sastoji se od: analize i utvrđivanja korisničkih zahtjeva, zatim projektiranja na konceptualnoj, logičkoj i fizičkoj razini, zatim implementacije, nadgledanja, održavanja i promjene baze podataka. Sve korake razvojnog ciklusa treba obuhvatiti kako bi baza podataka bila kompletna i kako bi obuhvatila sve potrebno. Slika 2-1 prikazuje schemu razvojnog ciklusa baze.



Slika 2-1 Schema razvojnog ciklusa baze podataka

Konceptualni model sastavljen je od entiteta, atributa i veza. Entitet je skup objekata o kojemu želimo spremati podatke iz stvarnog svijeta. Atributima su opisani entiteti, dok se vezama spajaju tablice koje imaju zajedničke informacije. Odnosno, veze između dvaju ili više entiteta ukazuju da su ti entiteti u nekom odnosu. Veze mogu biti jednostavne i složene, a svaki entitet je u vezi sa entitetima iz svog okruženja. Pobliže o vezama i koje sve veze postoje priloženo je niže na slici 2-2.

OZNAKA	NAZIV	OPIS
1:1	Jedan-naprama-jedan	Jedan primjerak od E_1 može biti povezan najviše s jednim primjerkom od E_2. Također, jedan primjerak od E_2 može biti povezan najviše s jednim primjerkom od E_1.
1:M	Jedan-naprama-mnogo	Jedan primjerak od E_1 može biti povezan s više primjeraka od E_2. Istovremeno, jedan primjerak od E_2 može biti povezan najviše s jednim primjerkom od E_1.
M:M	Mnogo-naprama-mnogo	Jedan primjerak od E_1 može biti povezan s više primjeraka od E_2. Također, jedan primjerak od E_2 može biti povezan s više primjeraka od E_1.

Slika 2-2 Veze između entiteta

Nakon projektiranja na konceptualnoj razini slijedi projektiranje na logičkoj razini – normalizacija. Projektiranje na logičkoj razini još nazivamo drugom fazom jer u tom dijelu se stvara logička shema sastavljena od tablica, odnosno relacija. Tijekom normalizacije logička struktura relacija se popravlja tako da više dođu do izražaja same osobine podataka. Treća faza je projektiranje na fizičkoj razini.

2.4. Objekti baze podataka

Kao što smo već prethodno definirali bazu podataka i kako je ona zapravo sustav pomoću kojeg se prikupljaju i organiziraju podaci, ono što nije spomenuto jest da je baza spremnik objekata. Detaljno o objektima niže u nastavku.

2.4.1. Tablica (table)

Tablica je temeljni objekt baze podataka. Baza podataka organizirana je pomoću tablica, sastoji se od stupaca i redaka u kojima su pohranjeni svi podaci kojima korisnici mogu pristupiti. Polja tablica su najčešće povezana primarnim ključem koji sadrži posebne vrijednosti za svaki zapis u tablici.

2.4.2. Stupac (Column)

Svaka tablica je sastavljena od stupaca i predstavljaju polja zapisa, dok redci od kojih su stupci sastavljeni predstavljaju zapise podataka.

2.4.3. Ograničenja (Constraints)

Relacije entiteta mogu sadržavati više vrsta ograničenja koja nisu kompleksna, ovdje ćemo nabrojiti samo neka od njih koja ne mogu biti modelirana nego ih se često treba riješiti odvojeno:

- **Uvjetna domena** – domena atributa ovisi o vrijednosti jednog ili više atributa istog entiteta.
- **Promjena vrijednosti stanja** – skup vrijednosti na koje se atribut može promijeniti ovisi o trenutnoj vrijednosti tog atributa.
- **Provjera raspona** – numerički atribut mora biti između vrijednosti atributa sa kojim je povezan.

2.4.4. Sekvenca (Sequence)

Sekvence se koriste kako bi se svi objekti u bazi razlikovali na način da se generira popis jedinstvenih brojeva za stupce tablica. Jamče generiranje jedinstvenih brojeva u rastućem ili padajućem redoslijedu.

2.4.5. Okidač (Trigger)

Okidači mogu ograničiti i zabraniti pristup pojedinim podacima u bazi podataka, te samostalno mogu izvršiti SQL naredbe i napraviti upise u bazu podataka. Baza automatski pokreće okidače kada se određeni događaji pokrenu u nekoj od tablica.

2.4.6. Pogled (View i materialized view)

Pogled je pregled baze podataka. Pomoću naredbe „*Select*“ dolazi se do uvida u podatke. Pogled nema vlastite podatke nego on povlači podatke iz željenih tablica, te ih prenosi u obliku informacija korisniku.

3. Izrada baze podataka

Nakon upoznavanja sa teorijskim dijelom baza podataka, vrijeme je da se stečeno znanje pretoči u praktični dio. Za ovaj projekt bilo je potrebno osmisliti i realizirati bazu podataka koja će poslužiti za organizaciju i jednostavnije održavanje sajma brodova. Potrebno je omogućiti pregled brodova, model i marku brodova, pregled izlagača brodova, tvrtki koje posjeduju brodove i njihovo sjedište (mjesto), posjetitelja koji su ujedno potencijalni kupci, mjesto održavanja sajma, datum održavanja sajma, sponzore sajma, račune posjetitelja koji sadrže cijenu i datum kupnje, organizatore sajma, rezervacije za kupnju broda od strane posjetitelja, imena prodavača koji su izdali račun posjetiteljima. Potrebno je omogućiti uvid u osobne podatke izlagača brodova, uvid u status broda tj. ako je brod već kupljen, rezerviran ili prodan. Računi za karte i računi za brodove izdaju se odvojeno, jer netko tko je posjetitelj ne mora nužno biti i kupac – stoga kupac i posjetitelj su odvojeni i imaju zasebne tablice.

Prvenstveno je bitno omogućiti svim radnicima i sudionicima ovog sajma lagan i jednostavan pristup svim podacima i informacijama koji su na jedinstven i smislen način prikazani s bazom podataka.

3.1. Entiteti

1. Sajam
2. Brod
3. Sponzor
4. Organizator
5. Država
6. Županija
7. Mjesto
8. Tvrtka
9. Izlagač broda

10. Marka
11. Model
12. Posjetitelj
13. Prodavač
14. Račun (karte)
15. Rezervacija
16. Status broda
17. Kupac
18. Račun (brod)
19. Sponzor_Sajam

Sajam

Entitet Sajam glavni je dio ove baze namijenjen je za pohranu podataka o sajmu. On nam na uvid daje **Naziv sajma**, **Vrijeme održavanja**, **Adresa**. Preko stranih ključeva dobivamo podatke o **Organizatorima**, **Mjestu održavanja**, **Brodovima**, **Računima.**

Brod

Ukoliko korisnika zanimaju podaci o brodu u entitetu Brod može pronaći sve podatke kao što su **Ime broda**, **Godina Proizvodnje**. Entitet Brod povezan je s entitetom **Izlagač broda**, također je povezan s entitetom **Tvrtka**, s entitetom **Model**, te **Rezervacija**. Preko entiteta Rezervacija imamo uvid u status broda, račun broda i kupca.

Izlagač broda

Preko entiteta Izlagač broda dobivamo podatke o **imenu** i **prezimenu** obučenog izlagača koji potencijalnim kupcima detaljno i smisleno izlaže brod.

Država

Entitet Država nam daje uvid u kojoj državi se sajam održava, tj. sadrži atribut **Ime države**.

Županija

Entitet Županija nam daje uvid u kojoj županiji se održava sajam odnosno **Ime županije**, te je povezana s entitetom **Država**.

Mjesto

Entitet Mjesto nam daje na uvid **Ime mjesta** u kojem se sajam održava, te također u kojem se mjestu neka tvrtka nalazi. Povezan je s entitetom **Županija**.

Tvrtka

Entitet Tvrtka ima atribute **Ime tvrtke**, **Adresa** tvrtke iz kojih možemo saznati osnovne informacije neke tvrtke brodova. Povezana je i s entitetom **Mjesto** koje nam daje uvid u kojem se gradu neka tvrtka nalazi.

Sponzor

Entitet Sponzor sadrži atribut **Ime sponzora** koji sponzoriraju sajam brodova.

Organizator

Entitet Organizator mora sadržavati sve informacije o organizatorima sajma, a tu spada **Ime** organizatora. Oni su zaduženi za cjelokupnu organizaciju sajma.

Posjetitelj

Entitet Posjetitelj nam daje informaciju **Rednog broja posjetitelja**, te iz toga možemo pratiti evidenciju i koliko ljudi je posjetilo sajam.

Prodavač

Iz entiteta Prodavač možemo saznati **Ime** i **Prezime** prodavača.

Račun (karte)

Entitet Račun (karte) nam služi za evidenciju posjeta samom sajmu, kada je najviše ljudi cirkuliralo kroz sajam, u slučaju neke pogreške preko šifre računa može se doći do određenog računa.

Prodavač

Entitet Prodavač ima atribute **Ime** i **Prezime** prodavača, te je povezan s entitetom **Račun** zbog toga što na računu mora biti vidljivo ime prodavača koji je izdao račun.

Status broda

Entitet Status broda nam govori je li određeni brod u neko vrijeme rezerviran za razgledavanje ili je prodan pa ga se mora naručiti, te na taj način olakšava rezervaciju i prodaju brodova jer automatski pokazuje koji je brod slobodan, kupljen ili rezerviran.

Model

Entitet Mode*l* nam daje na uvid **Ime modela** i koji svi modeli brodova postoje.

Marka

Entitet Marka nam daje na uvid **Ime marke** i koje sve marke na sajmu postoje.

Kupac

Informacije o Kupcima koje želimo dohvatiti radi cjelokupnog pregleda posjeta i kupnji, entitet Kupac nam daje informacije kao što su **Ime** i **Prezime** kupca, te **OIB**. Kupac radi **Rezervaciju** za razgledavanje broda, te nakon razgledavanja i vožnje brodom može i ne treba kupiti brod.

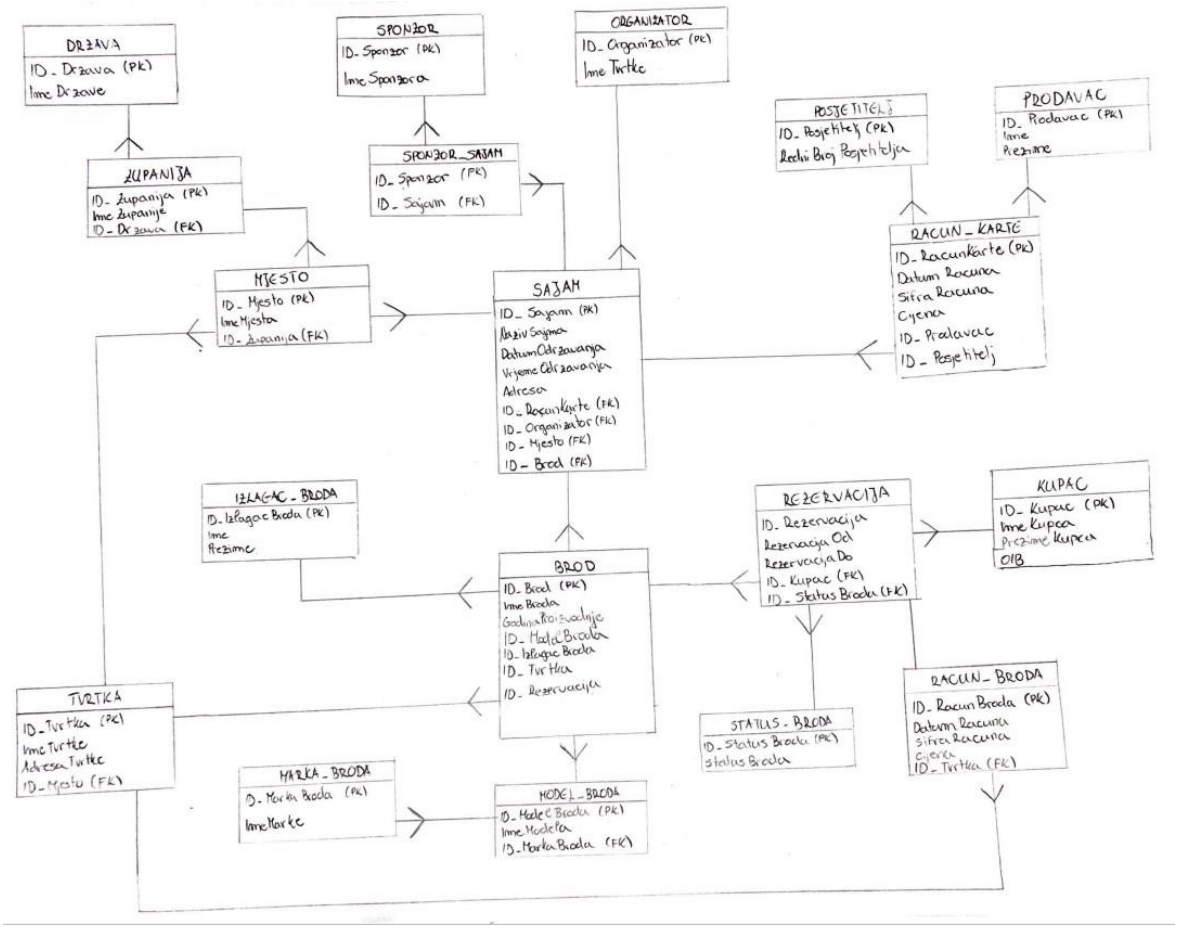
Račun (brod)

U slučaju oštećenja broda prilikom transporta, kupac mora posjedovati Račun u slučaju zamjene broda. Na računu mora biti vidljiv **Datum** i **Šifra računa**.

Sponzor_sajam

Entitet Sponzor_sajam nam govori između kojeg sponzora i sajma je potpisan ugovor o sponzorstvu.

Prvi korak pri izradi baze podataka je napraviti konceptualni model na papiru koji je prikazan slikom 3-1 niže u nastavku.



Slika 3-1 Konceptualni model na papiru

3.2. Relacije

Relacije odnosno veze su odnosi između dvije tablice koji pokazuju na koji način su entiteti međusobno povezani. Postoje tri vrste relacija:

- Jedan na jedan 1:1
- Jedan na više 1:n
- Više na više m:n

3.2.1. Relacija jedan na jedan

Svaka n-torka jednog entiteta može biti povezana samo s jednom n-torkom drugog entiteta.

U izradi ove baze relacija jedan na jedan je korištena između entiteta **račun** i **rezervacija** zbog toga što samo jedna rezervacija može imati samo jedan račun. Također, između entiteta **rezervacija** i **brod** je korištena relacija jedan na jedan zbog toga što na jednoj

rezervaciji može biti samo jedan brod, te jedan brod u određeno vrijeme može biti samo na jednoj rezervaciji.

3.2.2. Relacija jedan na više

Relacija jedan na više je relacija koja se najviše pojavljuje, te nam govori da jedan element iz nekog X entiteta može biti povezan s više elemenata iz nekog Y entiteta, a element iz Y može biti povezan sa samo jednim elementom iz X entiteta.

- U jednoj državi može biti više županija/pokrajina, dok jedna županija/pokrajina može biti u samo jednoj državi. Tako u jednoj županiji može biti više mjesta/gradova, dok jedno mjesto/grad može biti u samo jednoj županiji.
- Jedna tvrtka se može nalaziti u samo jednom mjestu/gradu, dok se u jednom mjestu/gradu može nalaziti više tvrtki. Jedan brod može biti pod vlasništvom samo jedne tvrtke, dok jedna tvrtka u vlasništvu može imati više brodova.
- Jedan izlagač broda može izlagati više brodova, dok jedan brod može izlagati samo jedan izlagač.
- Jedan organizator može organizirati više sajmova, dok jedan sajam može organizirati jedan organizator.
- Na jednom sajmu može biti izloženo više brodova, ali jedan brod može biti na samo jednom sajmu. Jedna marka može imati više modela, dok jedan model može biti samo jedne marke.
- Jedan prodavač može izdati više računa, dok jedan račun može izdati samo jedan prodavač.
- Jedan brod može imati samo jedan model, dok jedan model može imati više brodova. Također, brod može imati samo jedan status (rezerviran/slobodan/prodan), dok jedan status može imati više brodova.
- Jedan posjetitelj može napraviti više rezervacija, dok jednu rezervaciju može napraviti samo jedan posjetitelj. Također, jedan posjetitelj može posjetiti jedan sajam, dok na jednom sajmu može biti više posjetitelja.

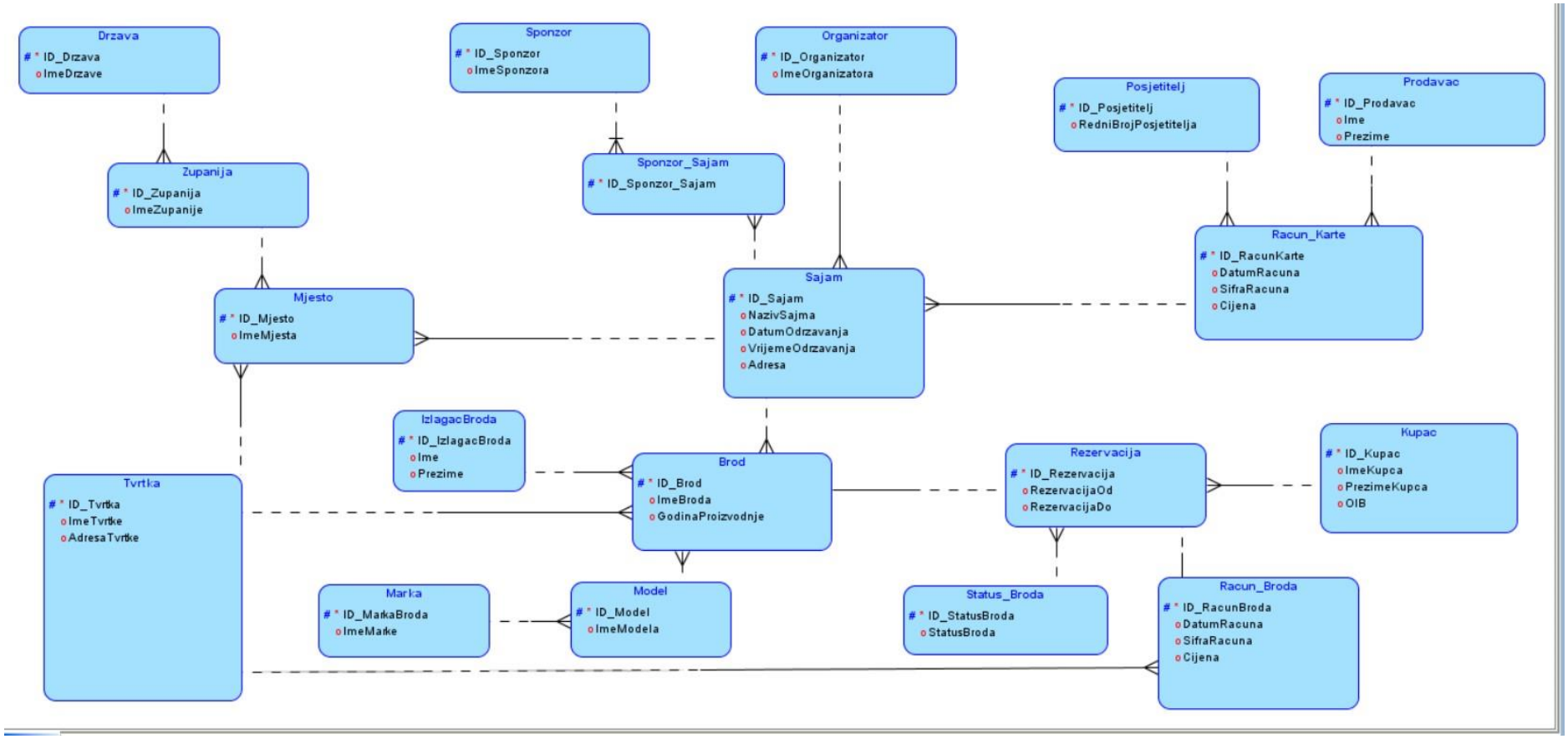
3.2.3. Relacija više na više

Jedan entitet može imati više elemenata iz drugog entiteta i obratno. Zbog nepoželjnosti ove relacije u bazama podataka taj problem se rješava na način da se uvodi novi entitet odnosno posljedični entitet čiji će primarni ključ u većini slučajeva biti sastavljen od primarnih ključeva osnovnih entiteta, te će imati naziv **kompozitni primarni ključ**.

U izradi ove baze jednom mi se pojavila veza više na više kod sponzora i sajma. Odnosno, na jednom sajmu se može pojaviti više sponzora, dok isto tako jedan sponzor može biti pokrovitelj na više sajmova.

3.3. Logički model

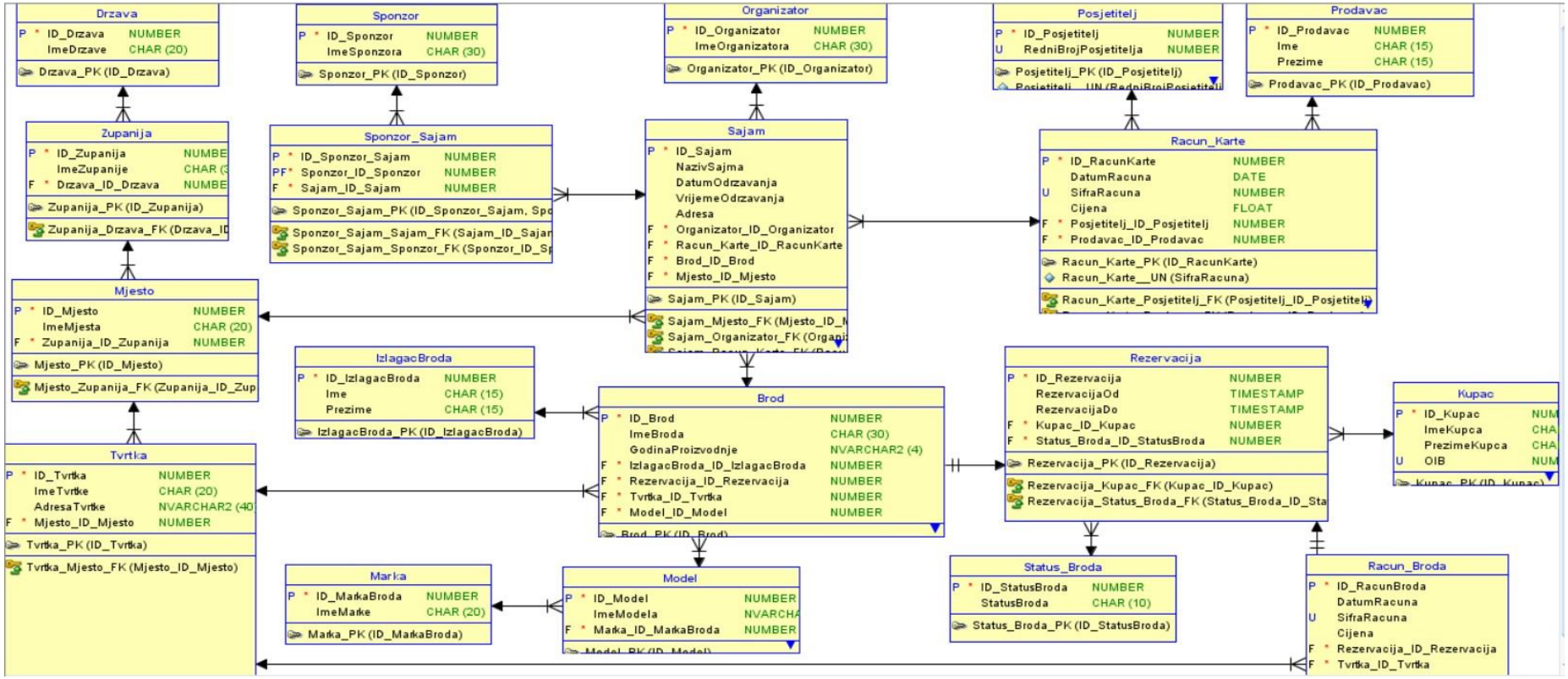
Logički model - ERM (Entity Relationship Model) uključuje sve entitete i odnose između njih. Prikazani su svi atributi za sve entitete te se određuje njihova optimalnost. Definira se opcionalnost i kardinalitet veza gdje su vidljivi primarni i/ili kompozitni ključevi, no ne i strani.



Slika 3-2 Logički model

3.4. Relacijski model

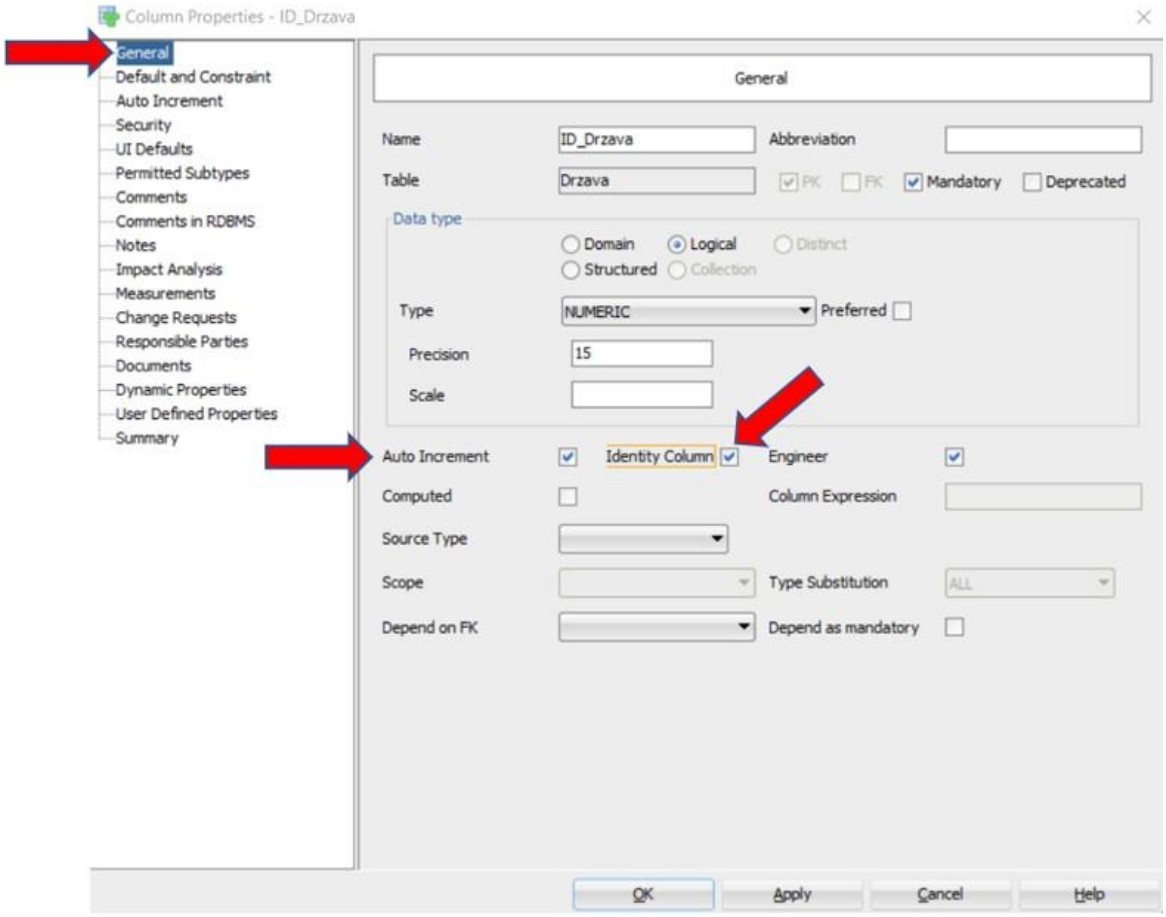
Nakon završenog logičkog modela izrađuje se relacijski model baze podataka. Iz ovog modela osim entiteta, relacija i atributa, možemo vidjeti i tipove podataka kao i strane ključeve u svim tablicama.



Slika 3-3 Relacijski model

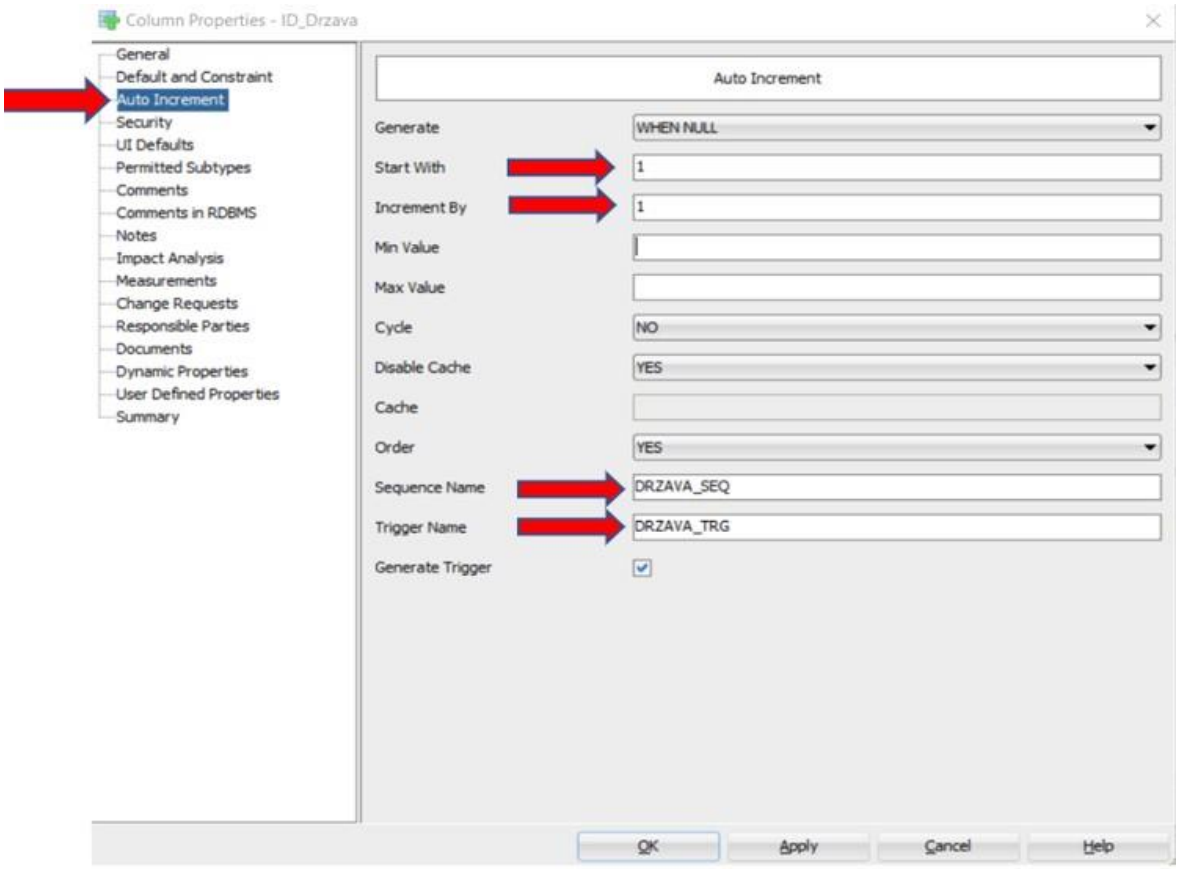
3.5. Izrada baze podataka prema modelu

Nakon kreiranja logičkog i relacijskog modela, sljedeće što trebamo učiniti je kreirati DDL da bih omogućili unos podataka i izvršavanje upita u *SQL Developeru*. Naime, za Oracle bazu ne postoji ugrađeni tip podataka za primarni ključ pa stoga prije izrade DDL-a je potrebno svim primarnim ključevima izraditi sekvencu - *sequence* i okidač – *trigger* za koji će uvećati ID za svaki sljedeći unos. Slika 3-4 jasno prikazuje pomoću crvenih strelica što je sve potrebno označiti.



Slika 3-4 Omogućeni "*Auto Increment*" i "*Identity Column*" za dodavanje sekvenci i okidača

Svaka sekvenca i okidač imaju svoje ime, a u ovom slučaju imena sekvencama i okidačima postavljena su na način da je prvi dio imena ime entiteta i dodatak „_SEQ“ ili „_TRG“ zavisno o tome je li u pitanju sekvenca ili okidač.

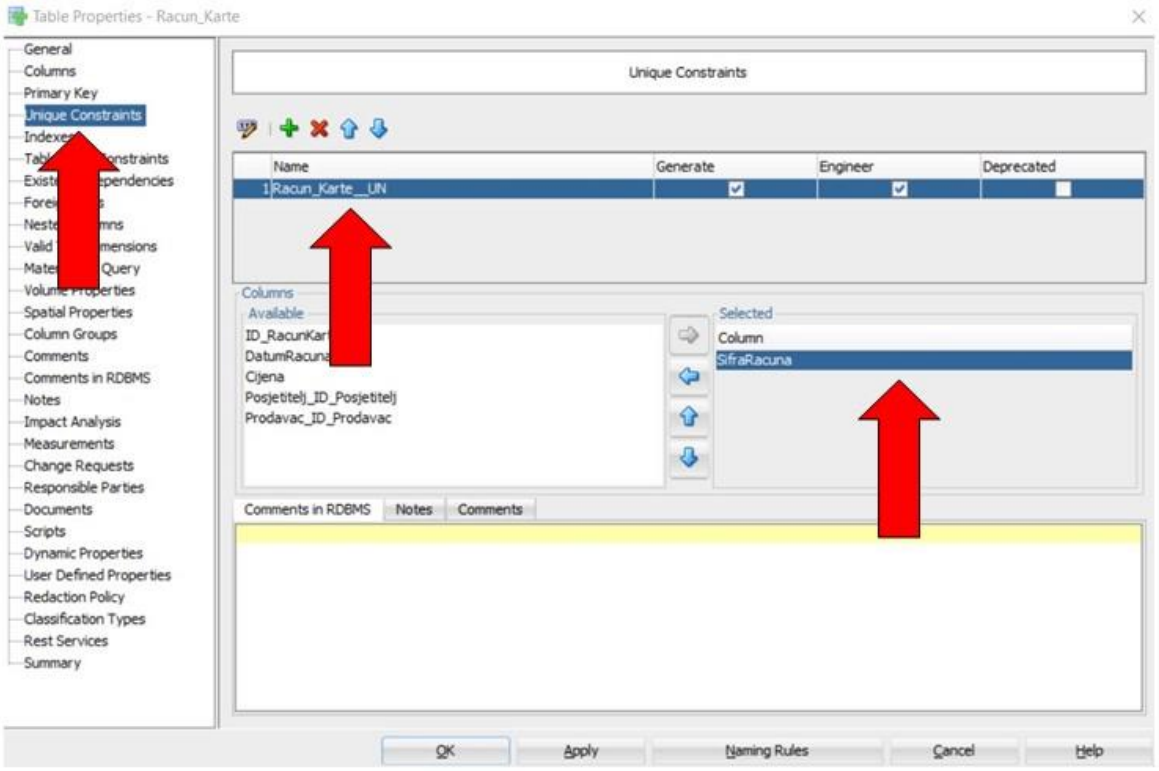


Slika 3-5 Dodavanje imena sekvencama i okidačima

Sekvenca će omogućiti povećavanje ID-a za svaki sljedeći unos. Naravno, sekvenca u jednom entitetu neovisna je o onoj u drugom entitetu. Okidač - *trigger* će nam pri unosu podataka omogućiti izvlačenje prvog slobodnog broja iz sekvence. Ovdje je također moguće upisati i minimalnu vrijednost od koje započinjemo.

3.5.1. Ograničenja (eng. Constraints)

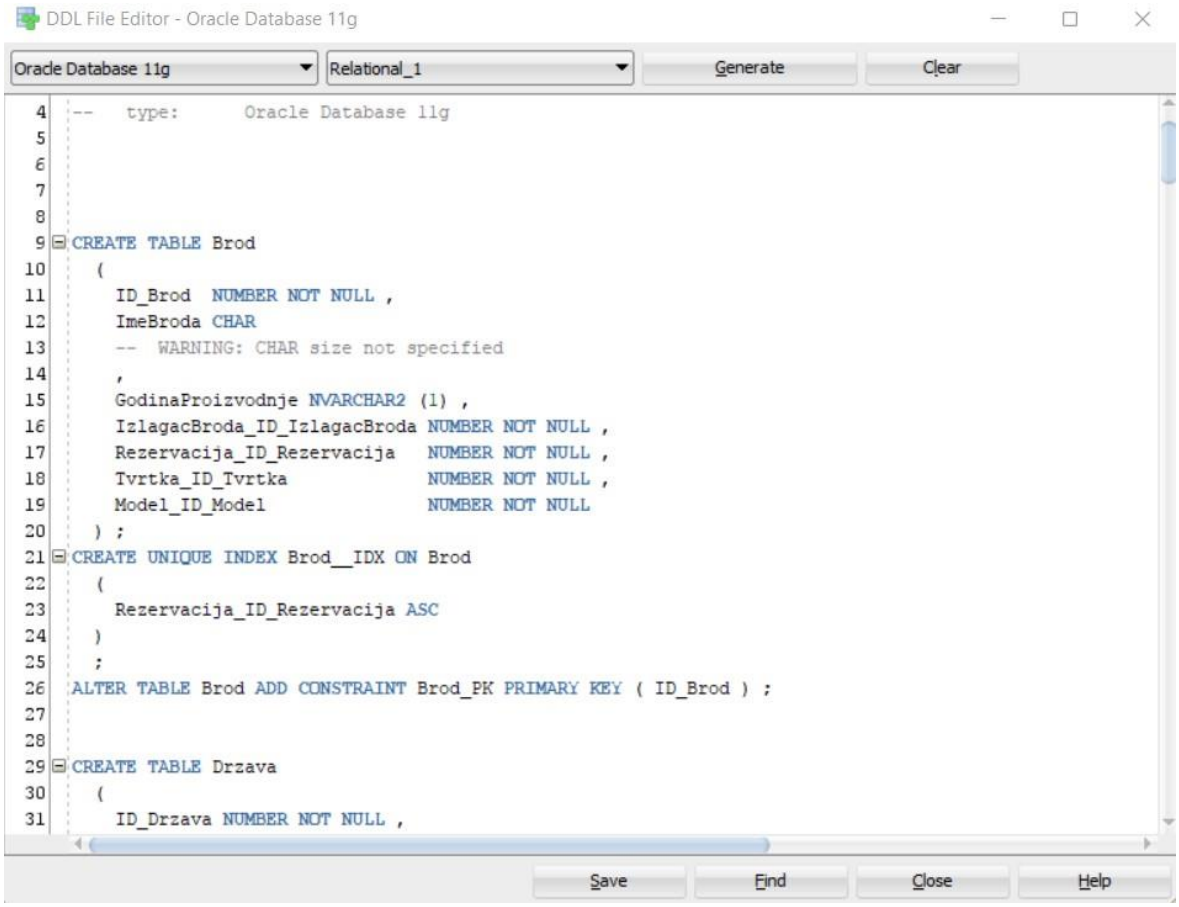
SQL izraz za postavljanje ograničenja nad podacima je „*Constraints*“, a kada želimo osigurati jedinstvenost nekog atributa koji nije primarni ključ koristimo „*Unique Constraint“*. Slika 3-6 Dodavanje "*Unique Constraints*" za šifru računa prikazuje dodavanje „*Unique Constraint*“.



Slika 3-7 Dodavanje "*Unique Constraints*" za šifru računa

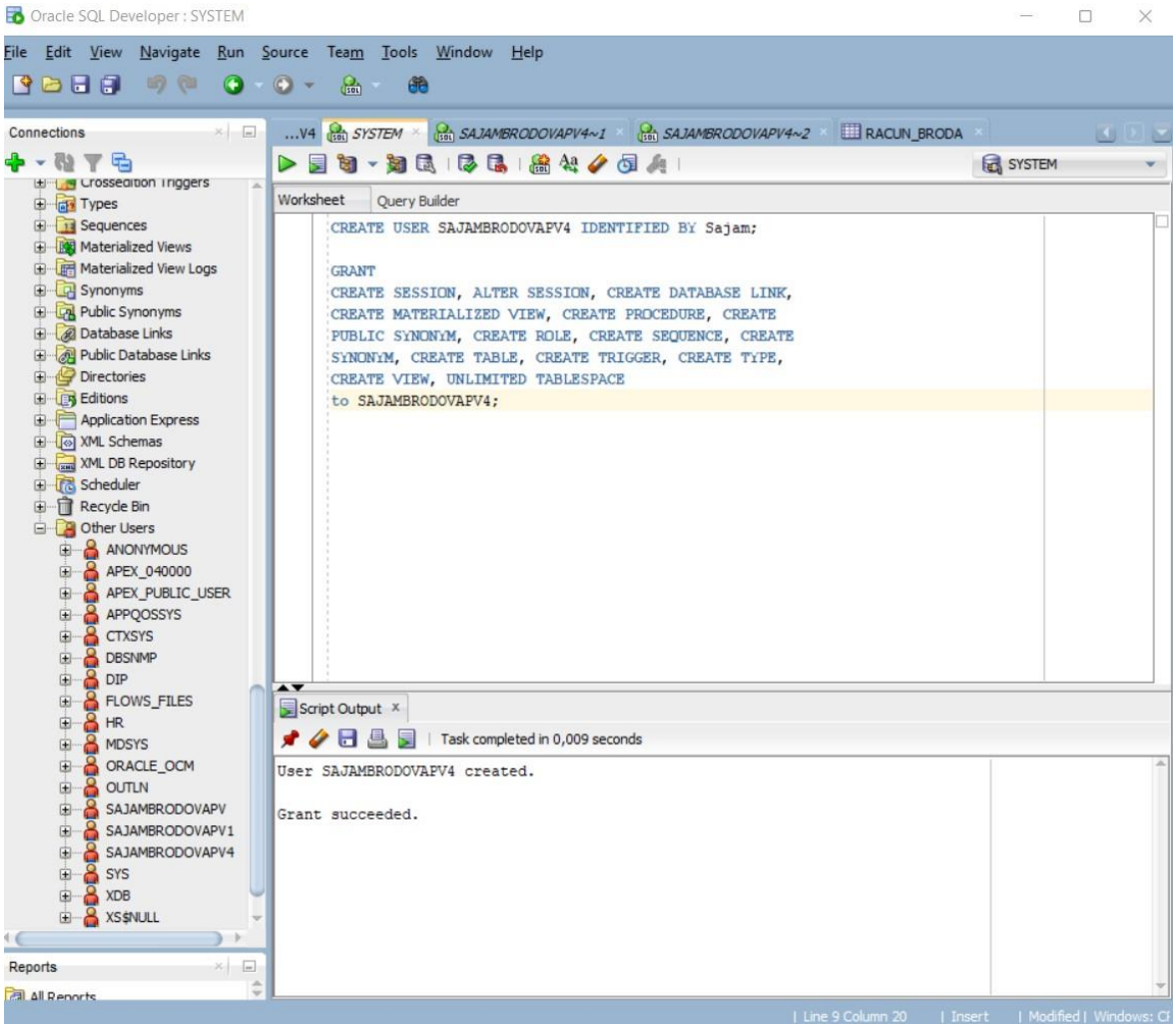
Jedinstveno ograničenje ima oznaku „U“ (eng. „Unique“) u logičkom i relacijskom modelu. *Unique Constraint* koristi se kod atributa koji su jedinstveni poput šifra računa, ni jedna šifra računa ne može biti ista jer se onda računi ne bi mogli razlikovati.

Nakon izrade sekvenci i okidača sljedeće što radimo je generiranje DDL-a u *SQL Modeleru* koji predstavlja kod za kreiranje svih atributa, entiteta, sekvenci i okidača određene baze podataka.



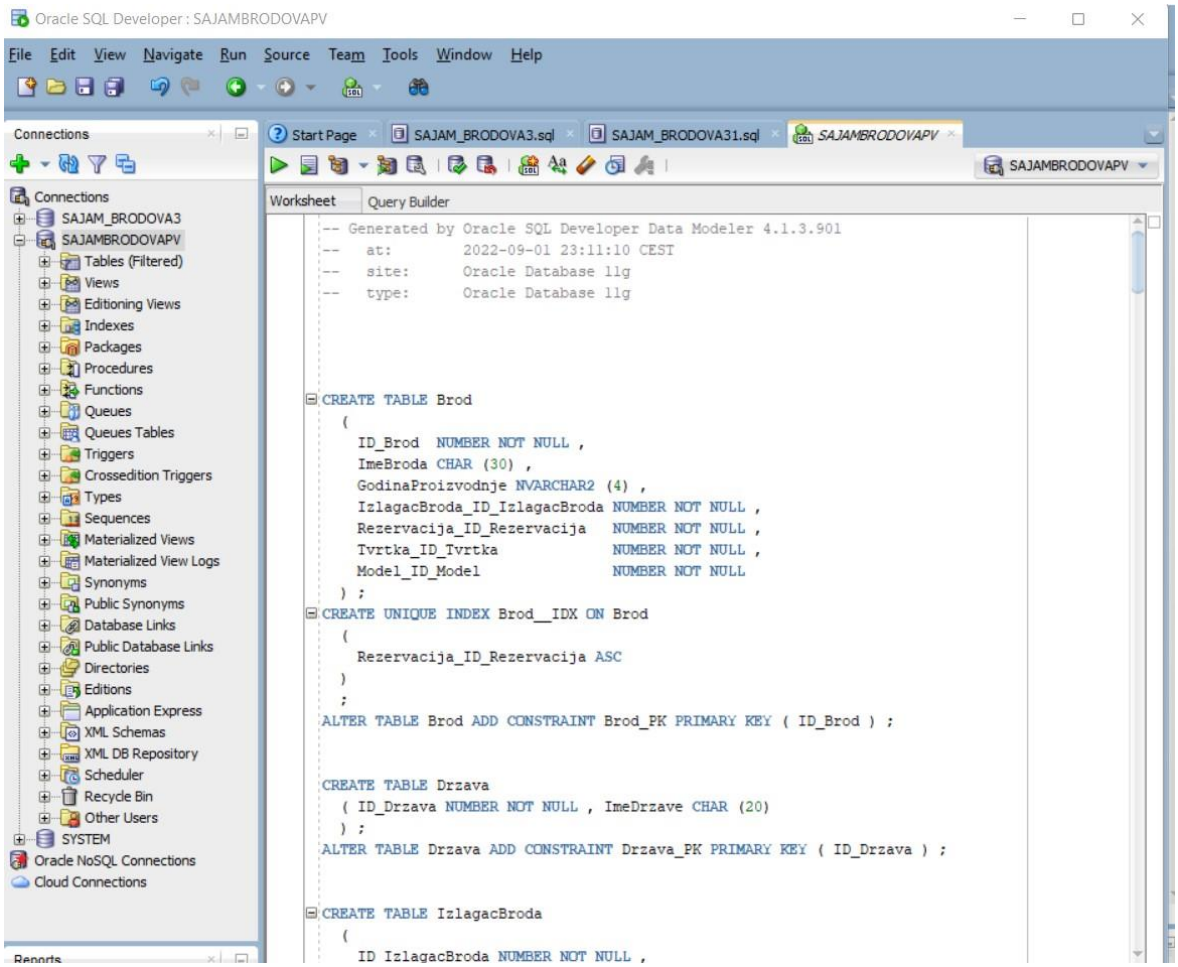
Slika 3-8 Generiranje DDL datoteke

Kako bi se mogla kreirati baza u *SQL Developeru* potrebno je imati instaliran Oracle, te kreirati novog korisnika i tom korisniku omogućiti ovlasti. Korisnik se zove SAJAMBRODOVAPV4, te su mu dodijeljene ovlasti.



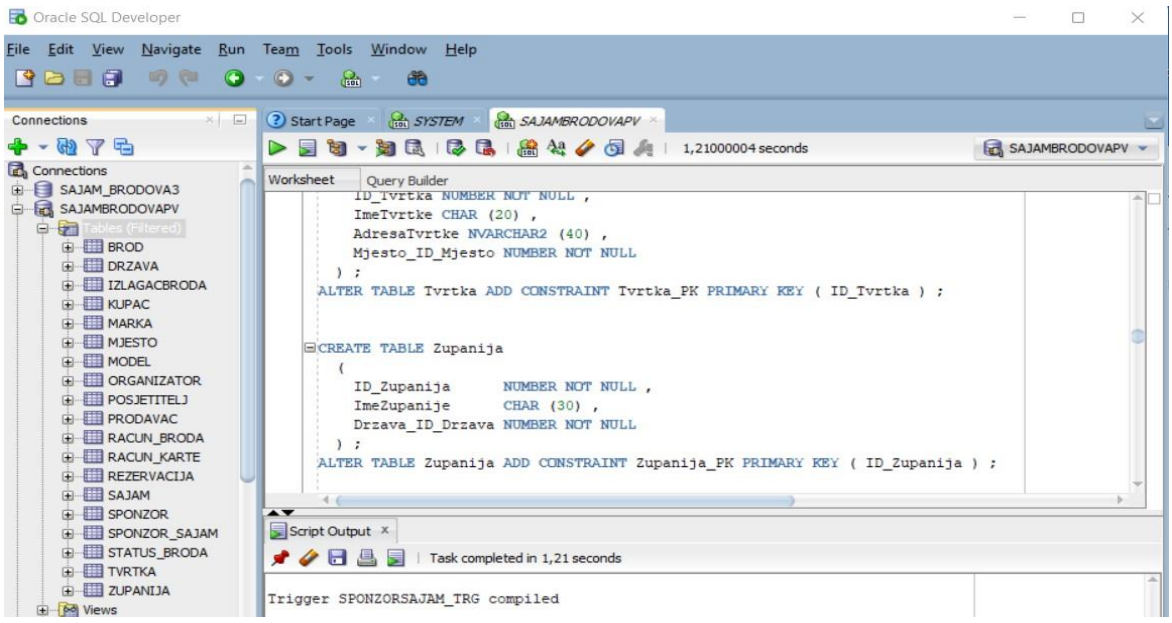
Slika 3-9 Kreiranje korisnika i dodavanje ovlasti

Nakon kreiranja novog korisnika, kod iz generirane datoteke DDL-a kopiramo se, a potom zalijepi u novog korisnika, koji se u ovom slučaju naziva SAJAMBRODOVAPV4. Nakon pokretanja koda uspješno su kreirane sve tablice, sekvence i okidače što je prikazano na slici 3-9.



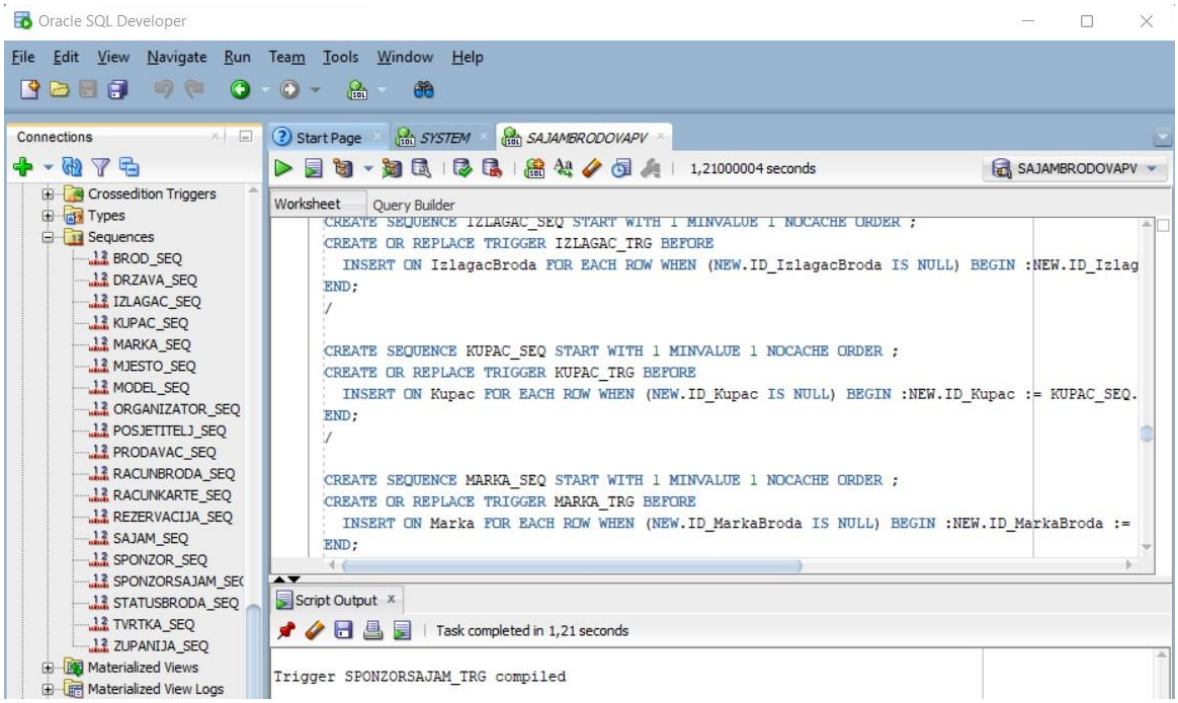
Slika 3-10 Kod iz DDL-a zalijepljen u Oracle SQL

Na slici 3-10 prikazan je primjer SQL izraza za kreiranje tablica.



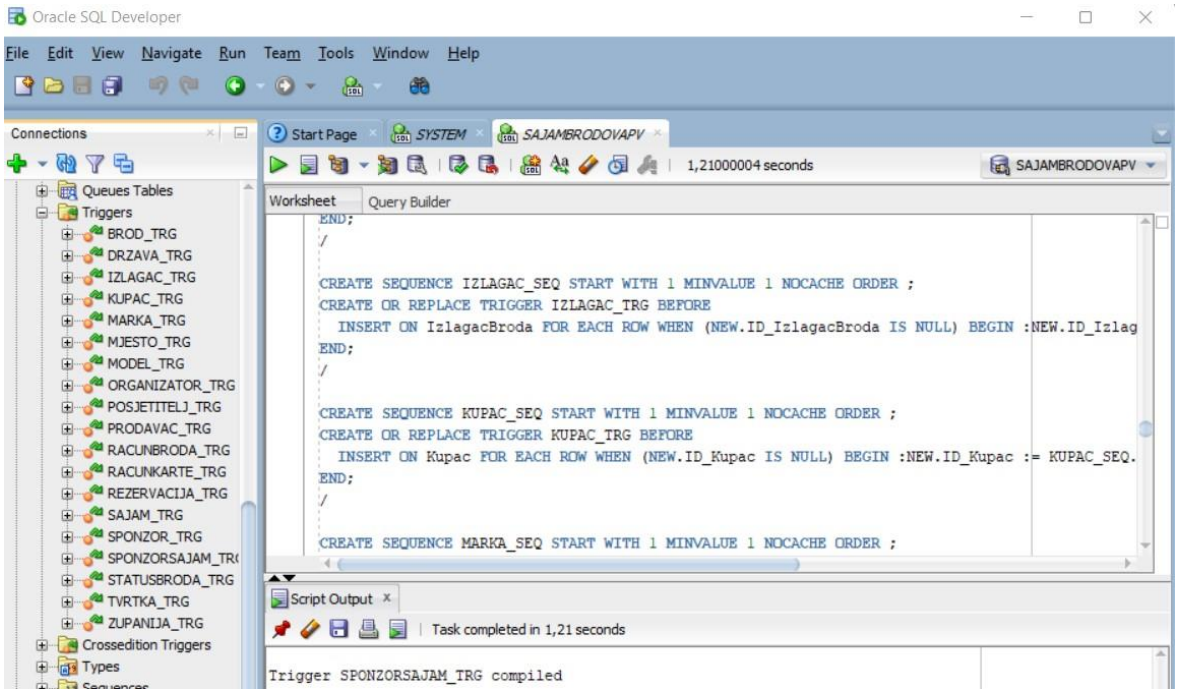
Slika 3-11 Kreiranje tablica i lijevo na slici prikaz tablica

Slika 3-11 jasno prikazuje kreiranje sekvenci.



Slika 3-12 Kreiranje sekvenci i lijevo na slici prikaz sekvenci

Slika 3-12 prikazuje kreiranje okidača.



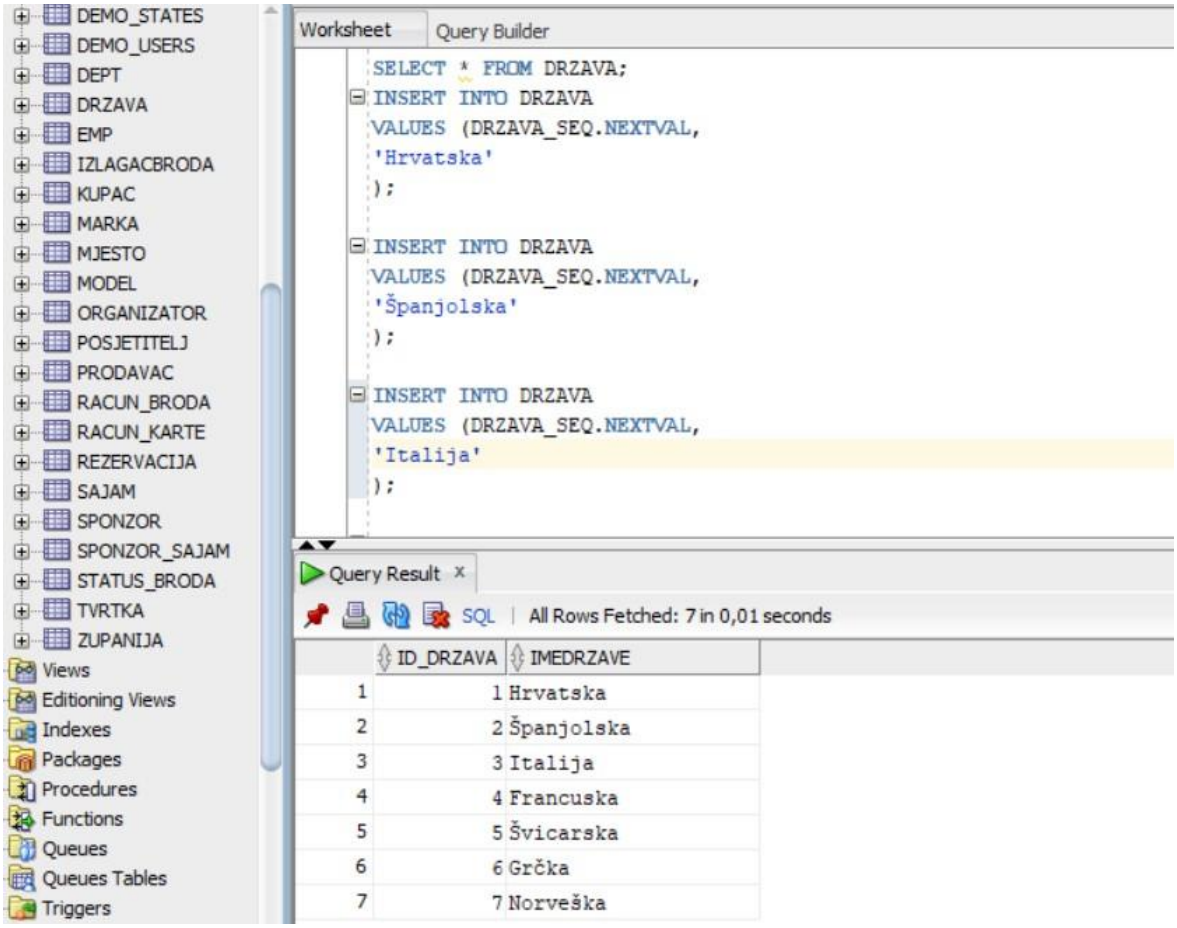
Slika 3-13 Kreiranje okidača i lijevo na slici prikaz okidača

3.6. Upiti na bazu podataka

Upiti na bazu podataka služe nam kako bi prikaz, dodavanje, brisanje ili ažuriranje podataka bili jednostavniji. Kombinacijom više upita možemo lakše doći do informacija koje su nam potrebne.

3.6.1. Unos podataka

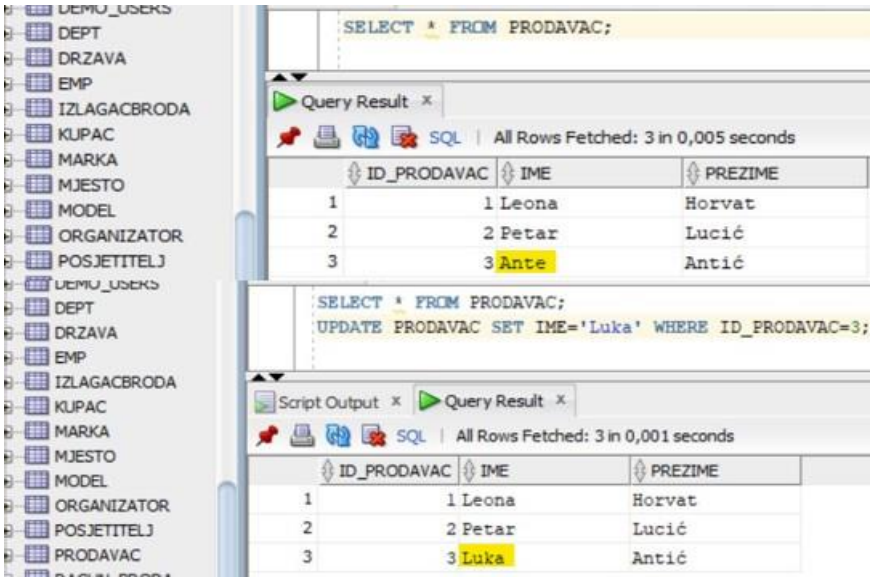
SQL izraz za unos podataka je „*Insert*“, te koristi se kako bi se unijeli podaci u tablicu. Da bih sve funkcioniralo, tablice je potrebno popunjavati „po redu“ kako smo odredili atribute.



Slika 3-14 Naredba Insert

3.6.2. Ažuriranje podataka

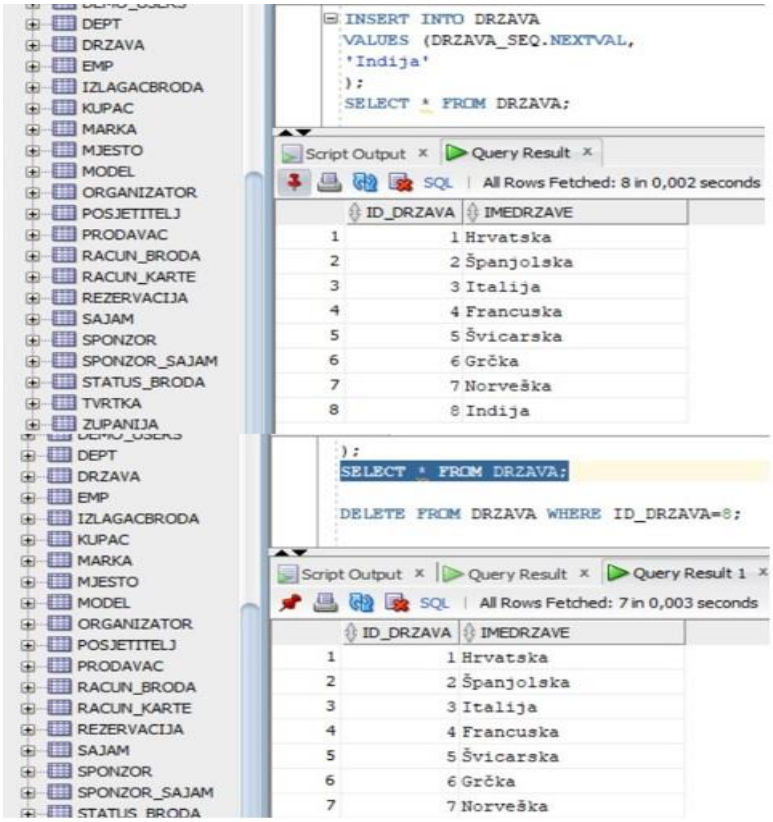
SQL izraz za ažuriranje podataka je *„Update“*.



Slika 3-15 Naredba *Update*

3.6.3. Brisanje podataka

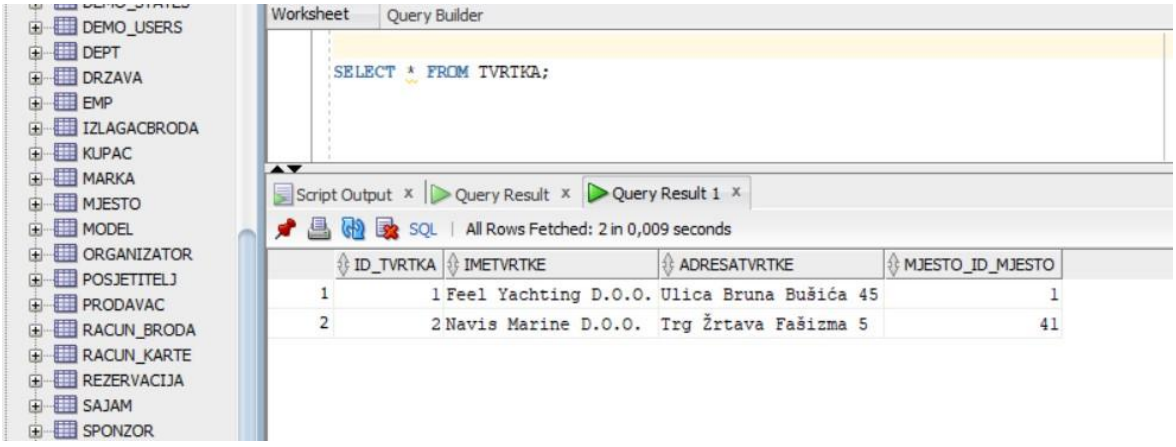
SQL izraz za brisanje podataka je „*Delete*“. Također se može koristiti za brisanje retka u tablici na način da se definira ID retka kojeg se želi izbrisati. Također se može koristiti za brisanje svih podataka u tablici.



Slika 3-16 Naredba *Delete*

3.6.4. Odabir podataka

SQL izraz za odabir podataka je „*Select*“, te dohvaća podatke iz jedne ili više tablica. Ako želimo dohvatiti sve podatke iz neke tablice koristimo znak *.



Slika 3-17 Naredba *Select*

3.7. APEX

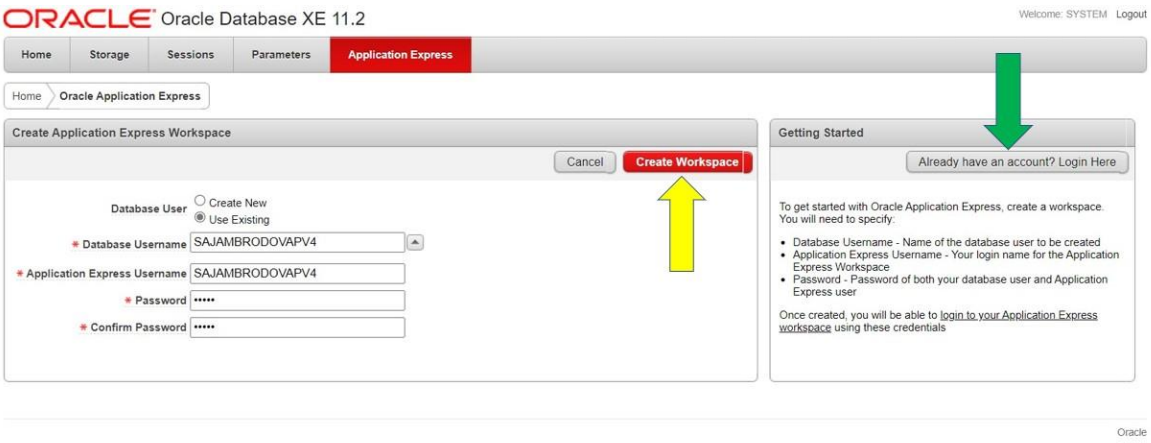
Oracle Application Express (APEX) je platforma za razvoj aplikacija baziranih na webu čiji rad je usko povezan sa serverom Oracle baze podataka. Od verzije 11g dolazi u sklopu same instalacije baze i automatski se instalira zajedno s njom, ali se naknadno mora izvršiti konfiguracija što je potkrijepljeno slikama ispod.

Oracle APEX omogućava:

- Razvoj profesionalnih, brzih i sigurnih web aplikacija
- Pokretanje i rad u okviru Oracle baze podataka, pri čemu se APEX resursi i podaci pohranjuju u tablice baze podataka
- Licenca nije potrebna za rad s ovim alatom
- Distribucija aplikacije je jednostavna nezavisno od okruženja i platformi
- Fleksibilna je u prilagođavanju izgleda i funkcionalnosti aplikacije
- Moguć je pristup velikog broja korisnika

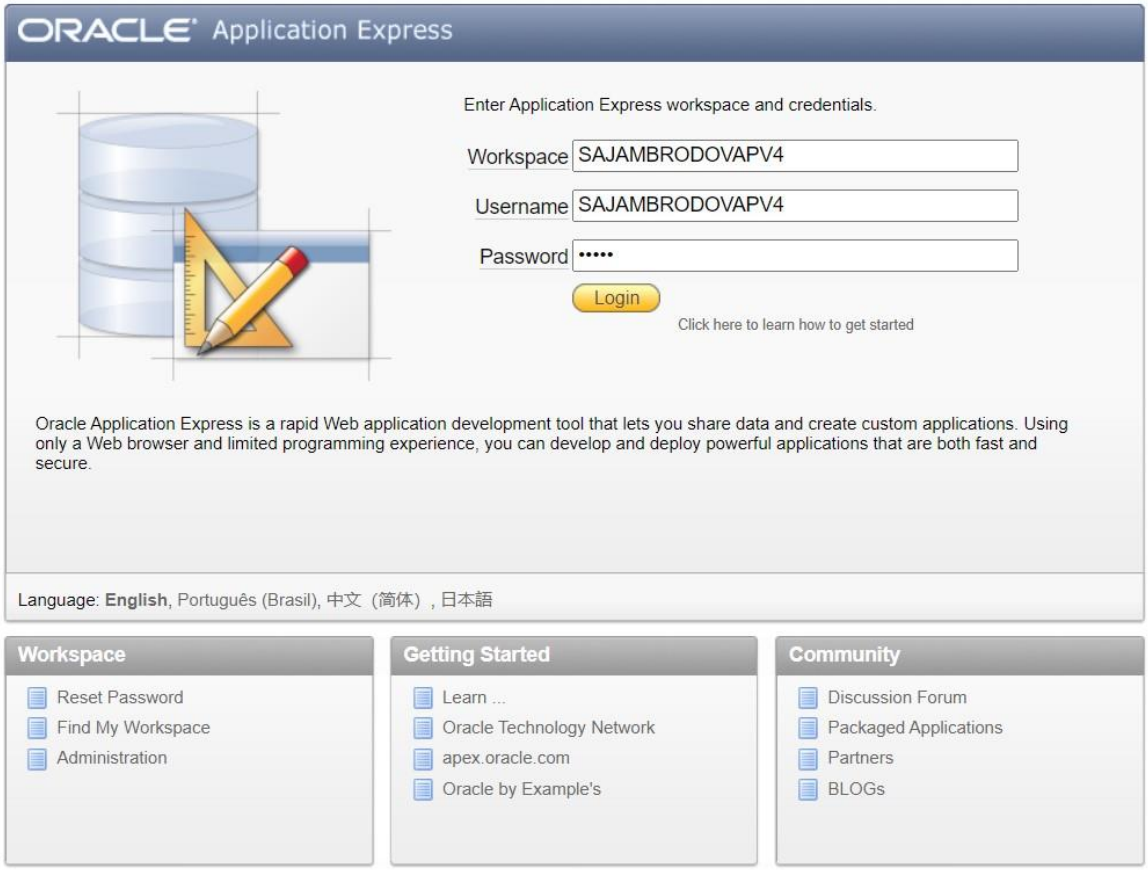
Koristeći APEX napravit ćemo aplikaciju za bazu podataka „Sajam brodova“. Preko aplikacije moći će se vidjeti podaci o sajmu, organizatorima, brodovima, izlagačima brodova, tvrtkama, rezervacijama, računima, posjetiteljima.
Kako bi se povezali na APEX na naslovnoj stranici koja nam se otvori trebamo kliknuti „*Application Express*“. Nakon toga potrebno je prijaviti se na SYSTEM korisnika s lozinkom koju je postavljena pri instalaciji (Slika 3-17). Korak nakon toga je napraviti novog korisnika za shemu (žuta strelica), a u slučaju ponovnog prijavljivanja potrebno je kliknuti na „Login“ (zelena strelica), jer u suprotnom izbaci grešku da korisnik već postoji (Slika 3-18).

Slika 3-18 Apex prijava



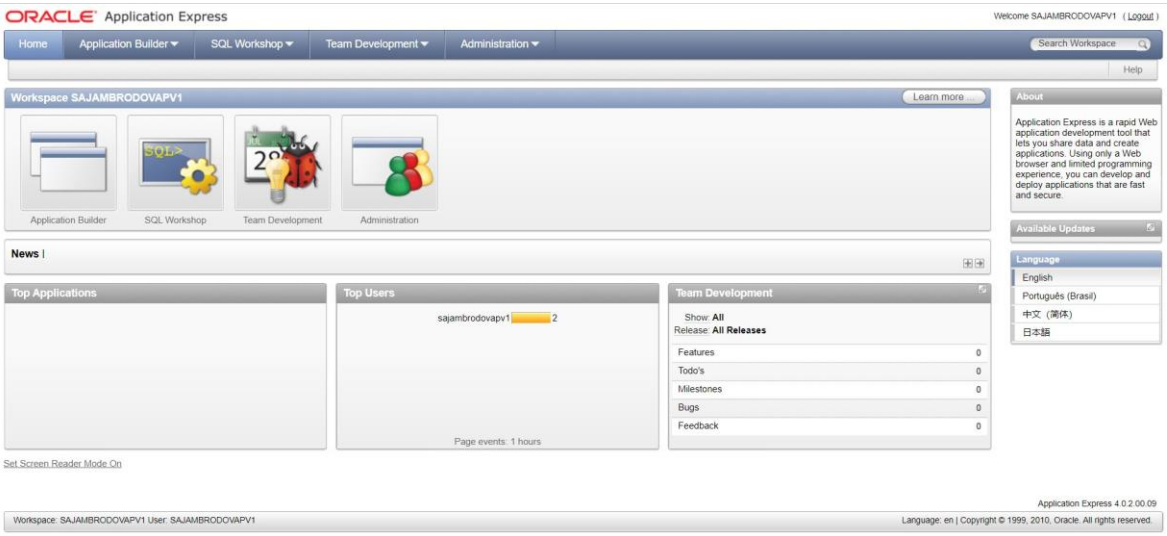
Slika 3-19 Apex kreiranje baze

Sljedeće što je potrebno je unijeti lozinku i spojiti se na kreiranog korisnika, nakon spajanja na kreiranog korisnika automatski se otvara sučelje pogodno za izradu aplikacije. Kada se otvori sučelje potrebno je kliknuti na „*Application Builder*“.



Slika 3-20 Apex spajanje na korisnika

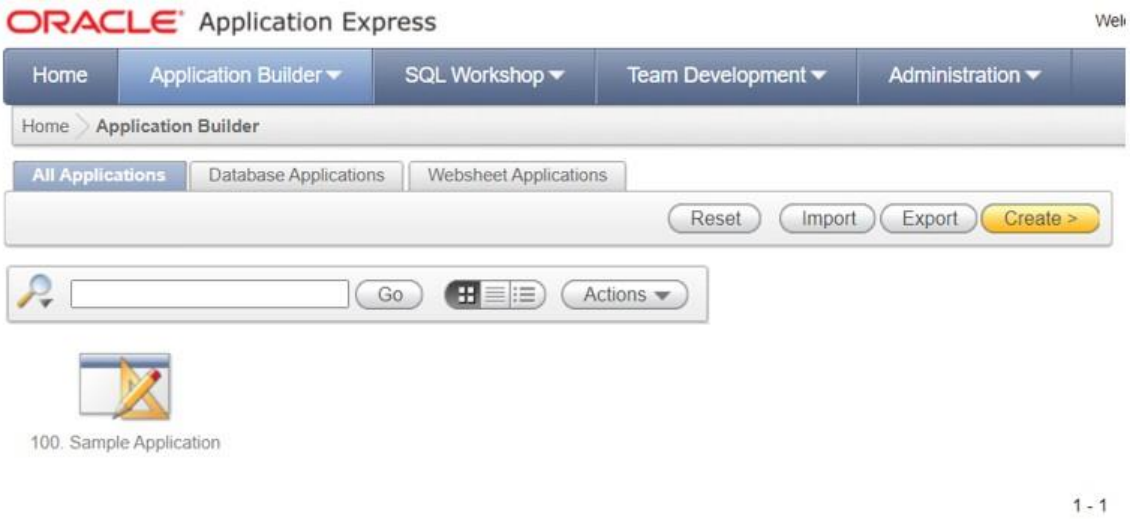
Slika 3-20 prikazuje naslovnu stranicu *Application Express*-a.



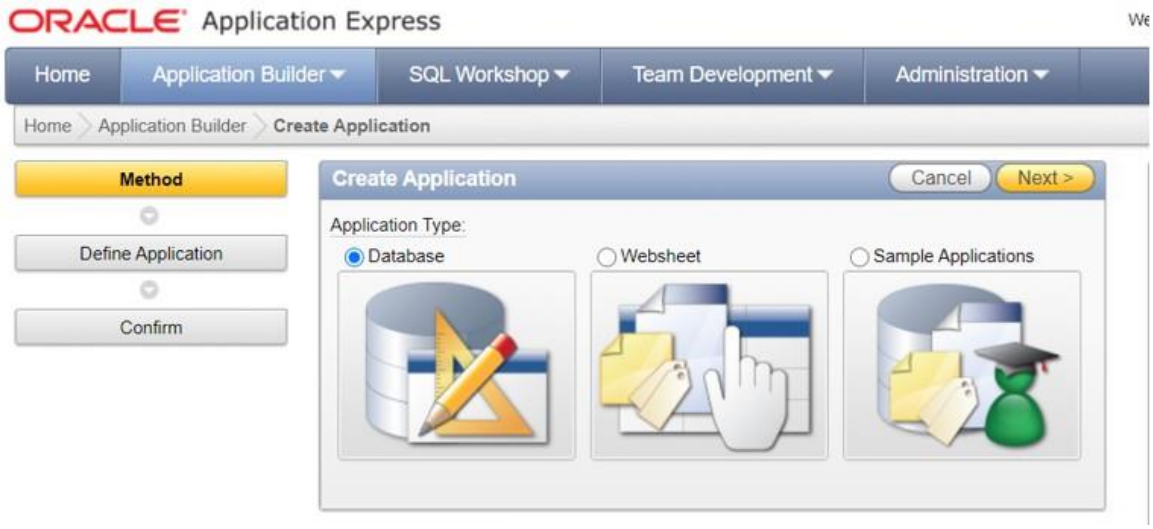
Slika 3-21 Naslovna stranica Application Express-a

3.7.1. Application builder

Kako bi se kreirala baza prvotno treba kliknuti „*Create*“, zatim „*Database*“ jer je potrebno kreirati aplikaciju baze podataka (Slika 3-21).



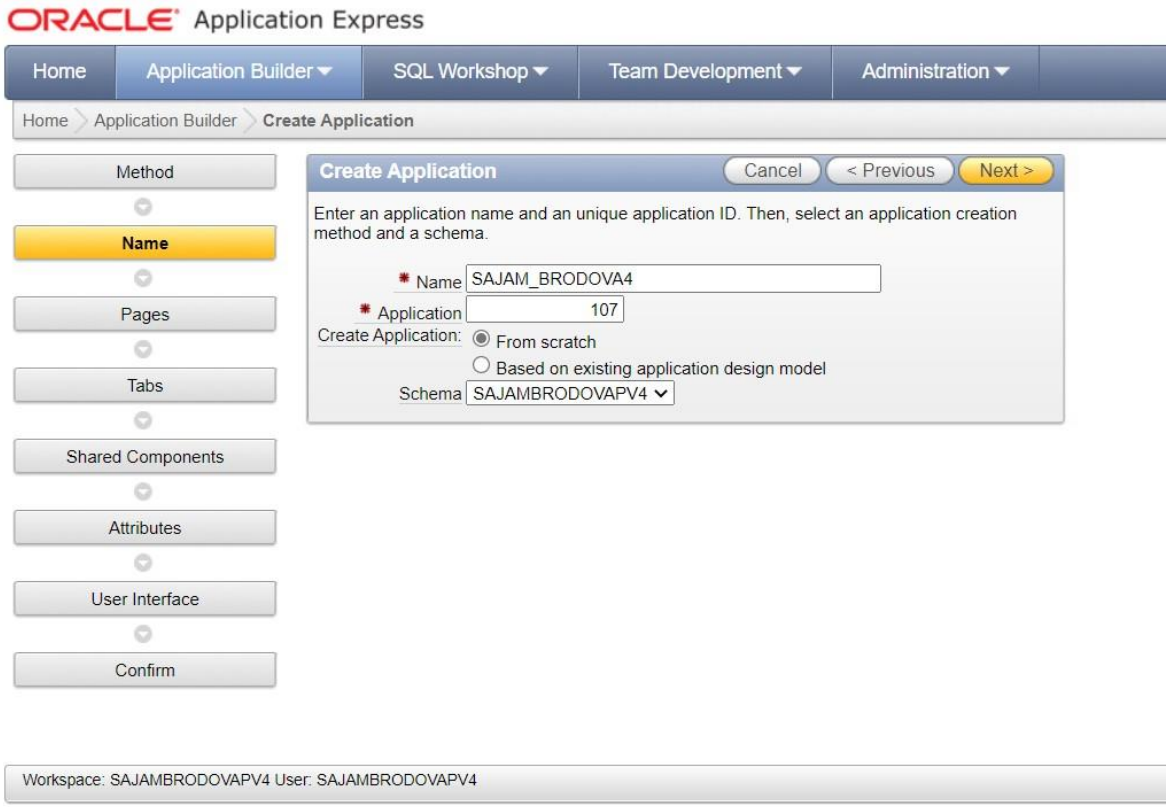
Slika 3-22 Konfiguracija za web aplikaciju



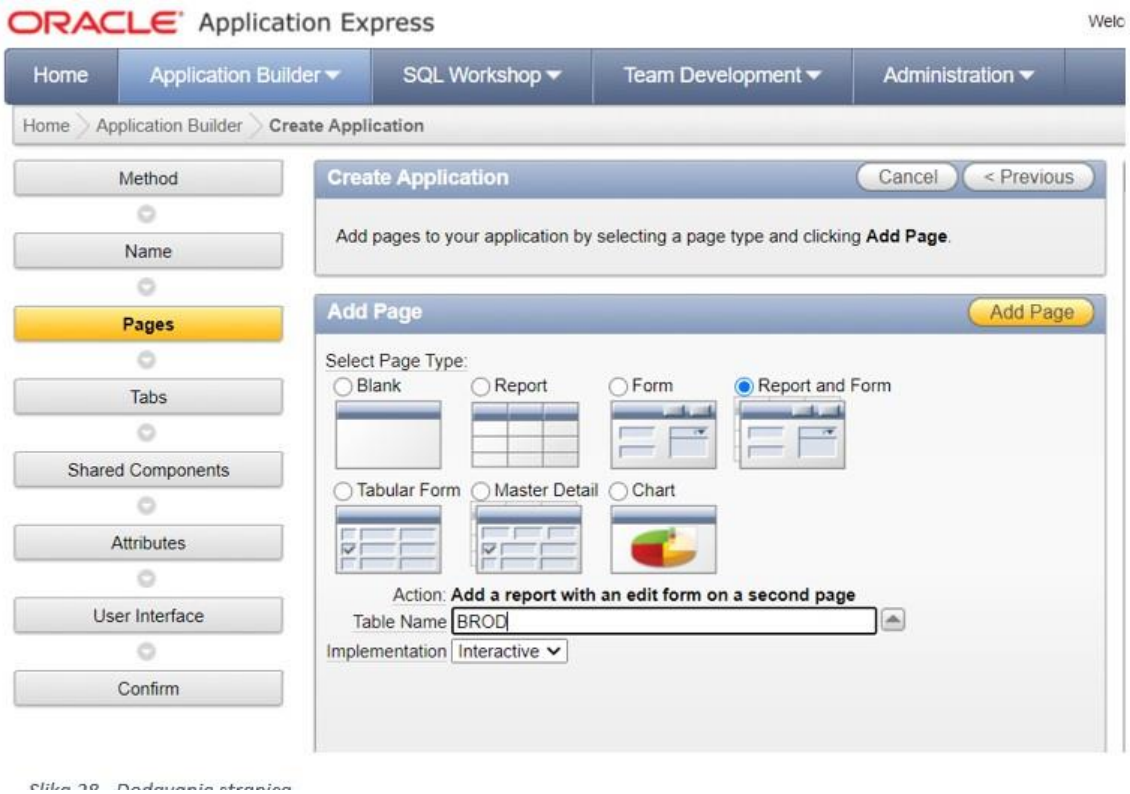
Slika 3-23 Konfiguracija za web stranicu

Potrebno je kliknuti „*From Scratch*“ zbog izrade iz temelja. Nakon toga novo ime dodjeljuje se aplikaciji (SAJAM_BRODOVA4). Kada se ime dodijeli, otvara se nova stranica na kojoj je omogućeno dodavanje funkcija koje željena aplikacija može predstavljati. Stranica može predstavljat ili skup iz kojega je moguć prikaz podatka ili pak unos podataka ili mijenjanje postojećih. Tip stranice „*Report*“ služi samo za pregled podataka. *„Form*“ služi za unos podataka, dok *„Report and Form*“ je kombinacija već spomenutih dviju stranica. *„Master Detail“* služi za uređivanje baze podataka. *Za „Report and Form“* stranice odabrani su entiteti: Brod, Izlagač broda, Tvrtka, Marka broda.

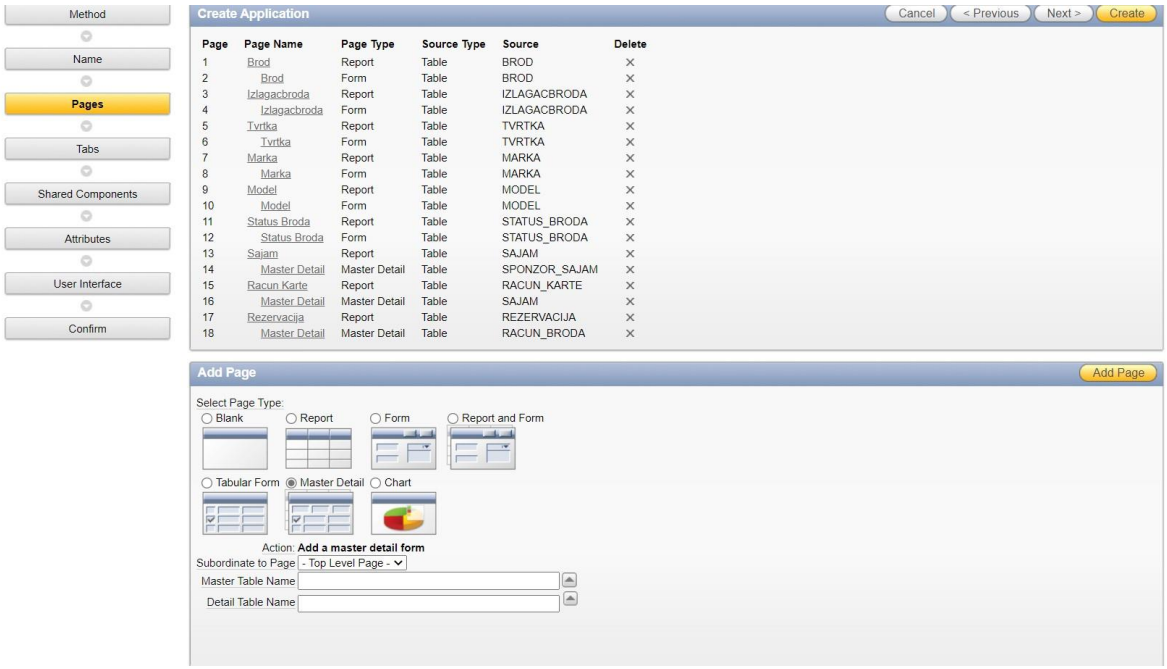
Slika 3-24 Konfiguracija za web aplikaciju



Slika 3-25 Dodavanje imena aplikaciji

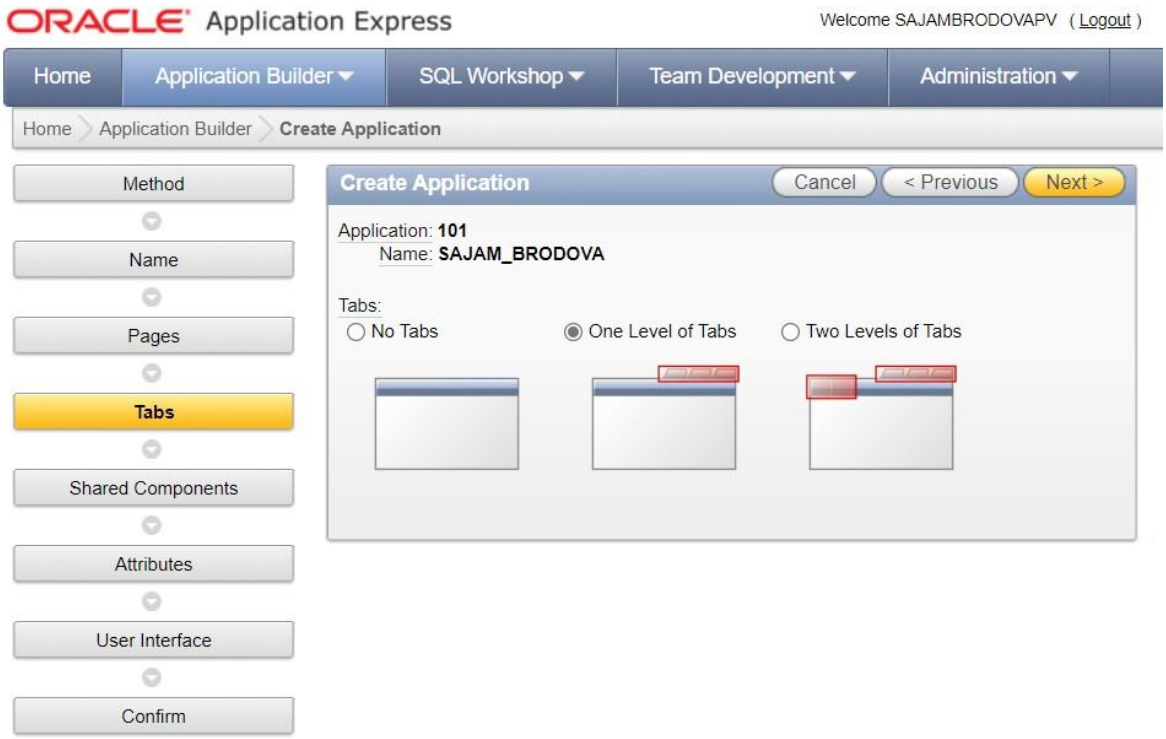


Slika 3-26 Dodavanje stranica - Report and Form



Slika 3-27 Pregled dodanih stranica

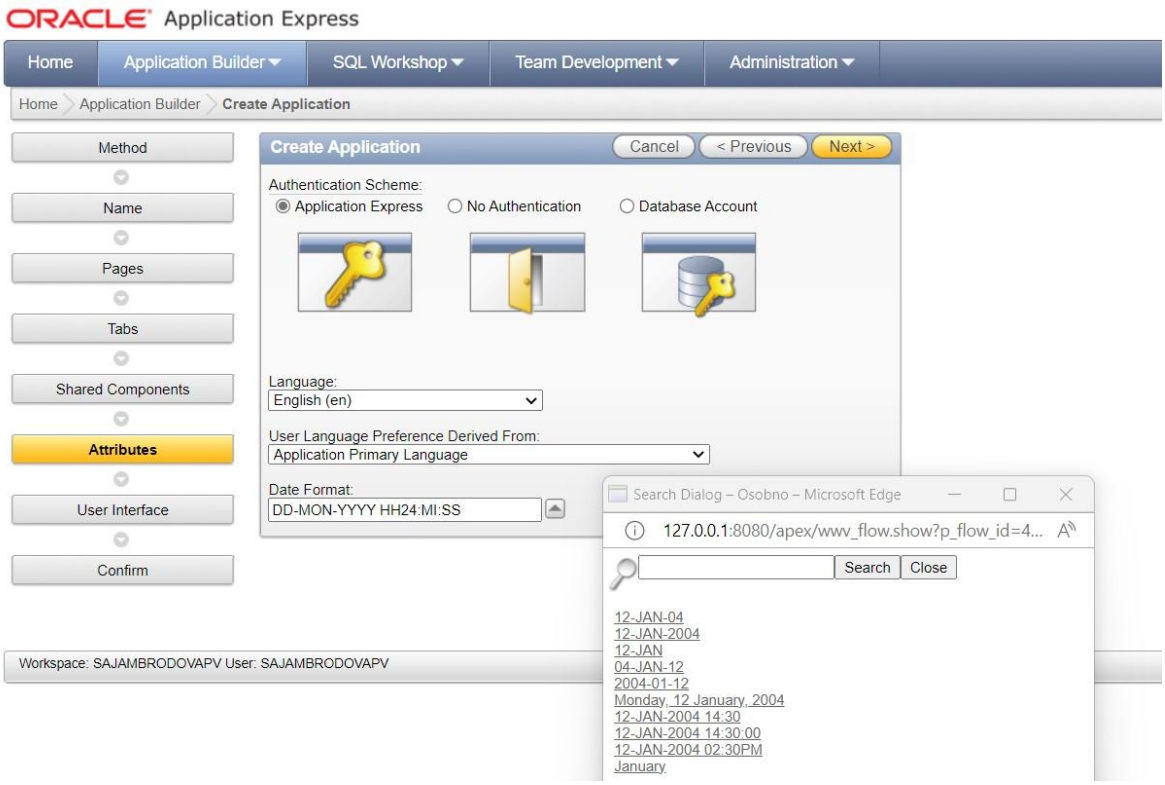
Nakon izrade svega što je potrebno za web aplikaciju kliknite *„Next“*. Na sljedećoj stranici prikazan je odabir za razinu tabulatora.



Slika 3-28 Odabir razine tabulatora

Sljedeći korak koji je potrebno napraviti jest izabrati format datuma, te kliknuti „*Next*“.

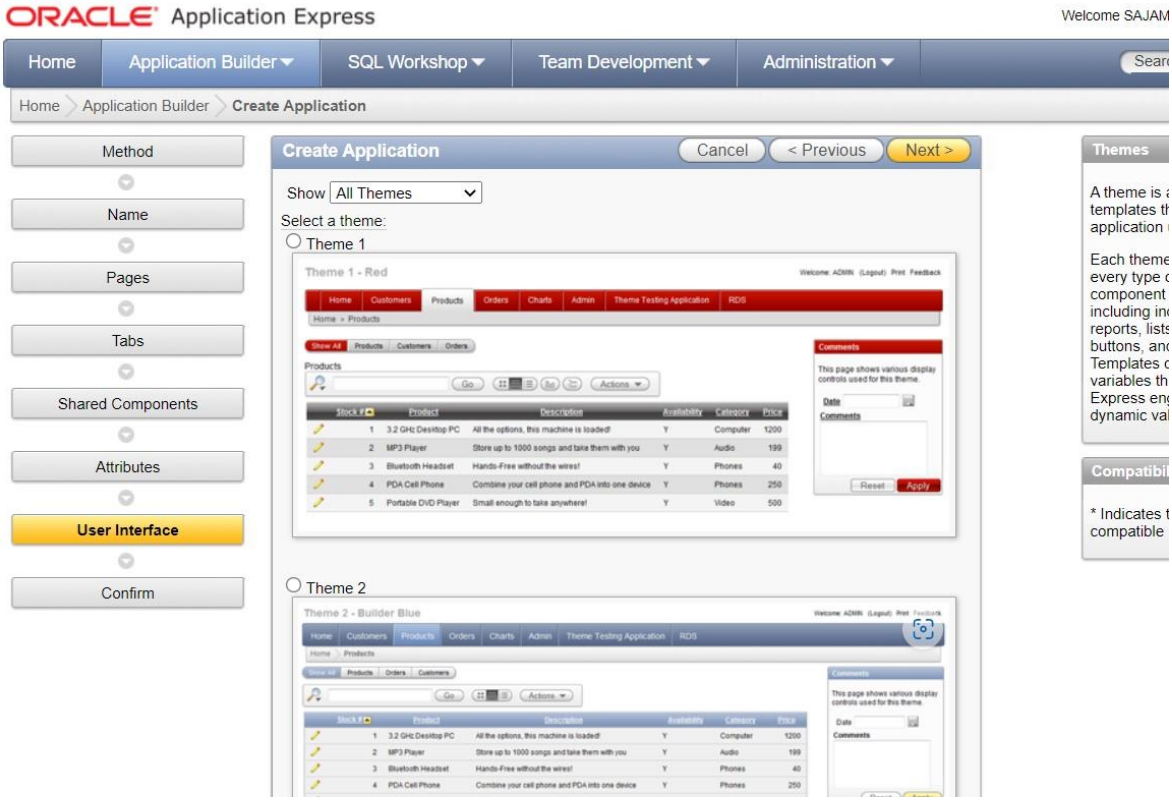
Slika 3-29 Konfiguracija



Slika 3-30 Odabir formata datuma

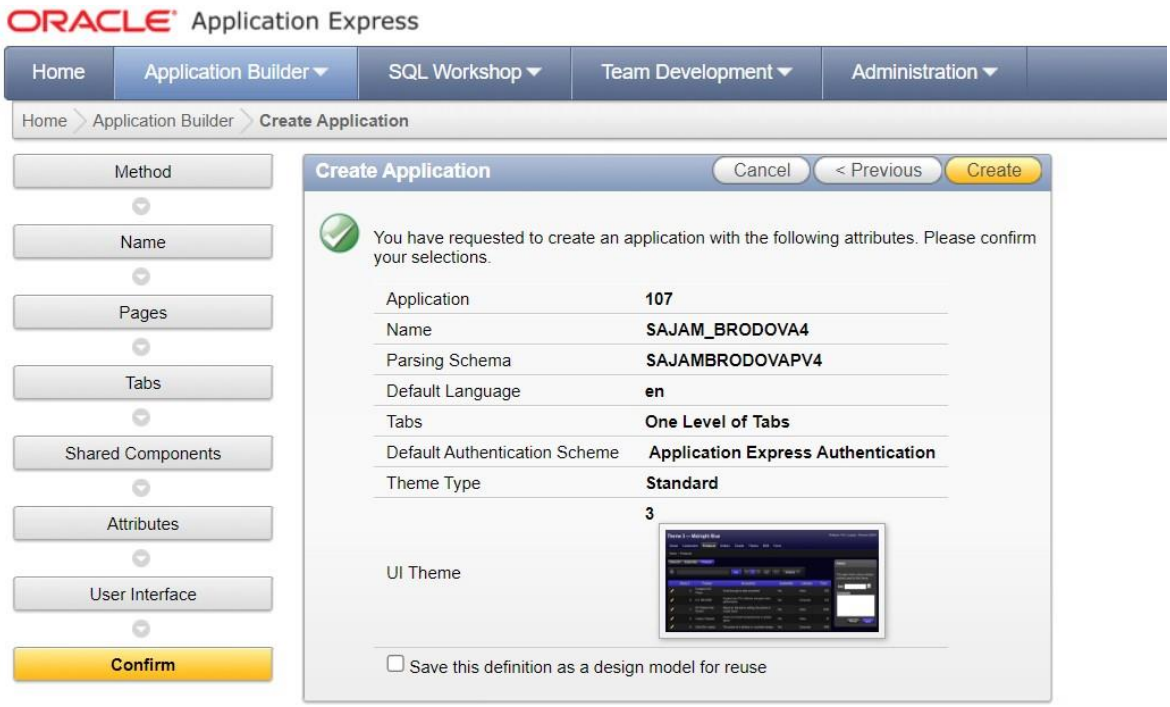
Nakon što kliknete *„Next“* otvara se stranica za odabir sučelja. Nakon odabira željenog sučelja potrebno je kliknuti *„Create“*.

Slika 3-30 prikazuje koje sve naslovne teme je moguće odabrati.



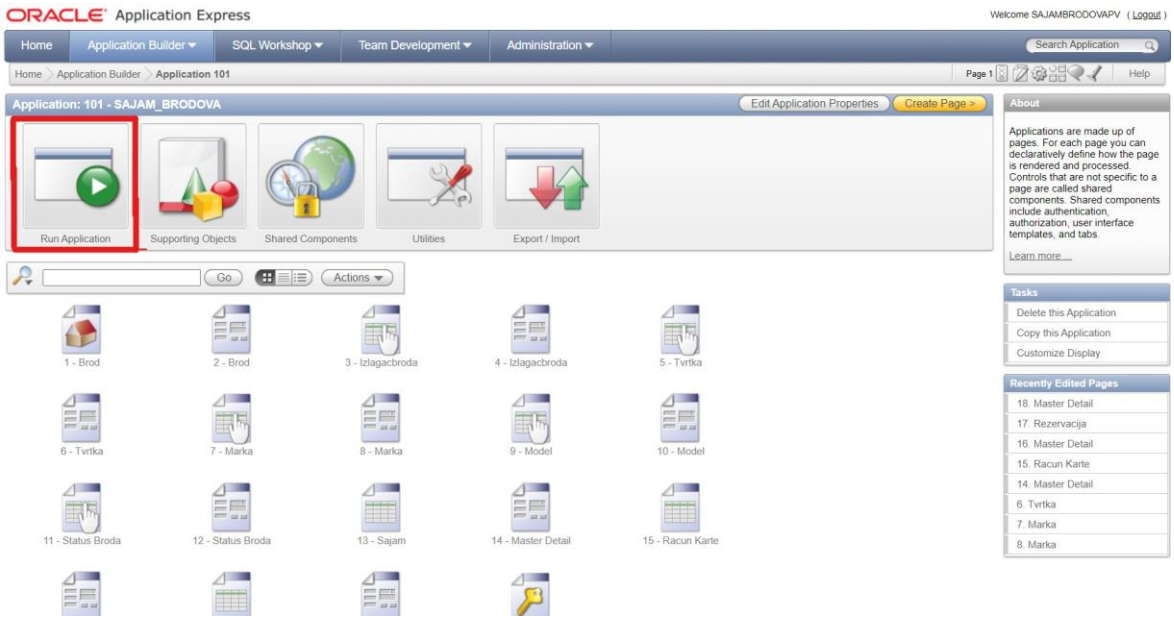
Slika 3-31 Prikaz sučelja za odabir

Nakon odabira teme na slici 3-31 prikazani su detalji cjelokupne konfiguracije.



Slika 3-32 Prikaz cjelokupne konfiguracije

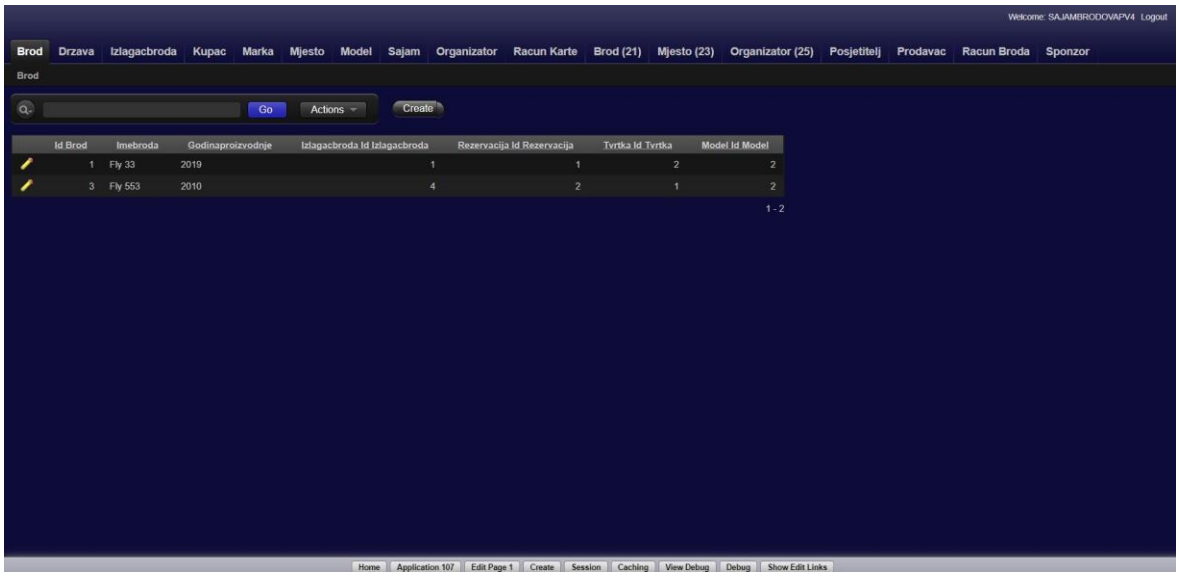
Nakon kreiranja potrebno je kliknuti *„Run Aplication“*, nakon čega je potrebno prijaviti se sa imenom i lozinkom koju je postavljena. Kada je lozinka unesena prvotno se otvara *„Report and Form*“ dio.



Slika 3-33 Naslovna stranica nakon konfiguracije

3.7.2. Izvještaji i forme

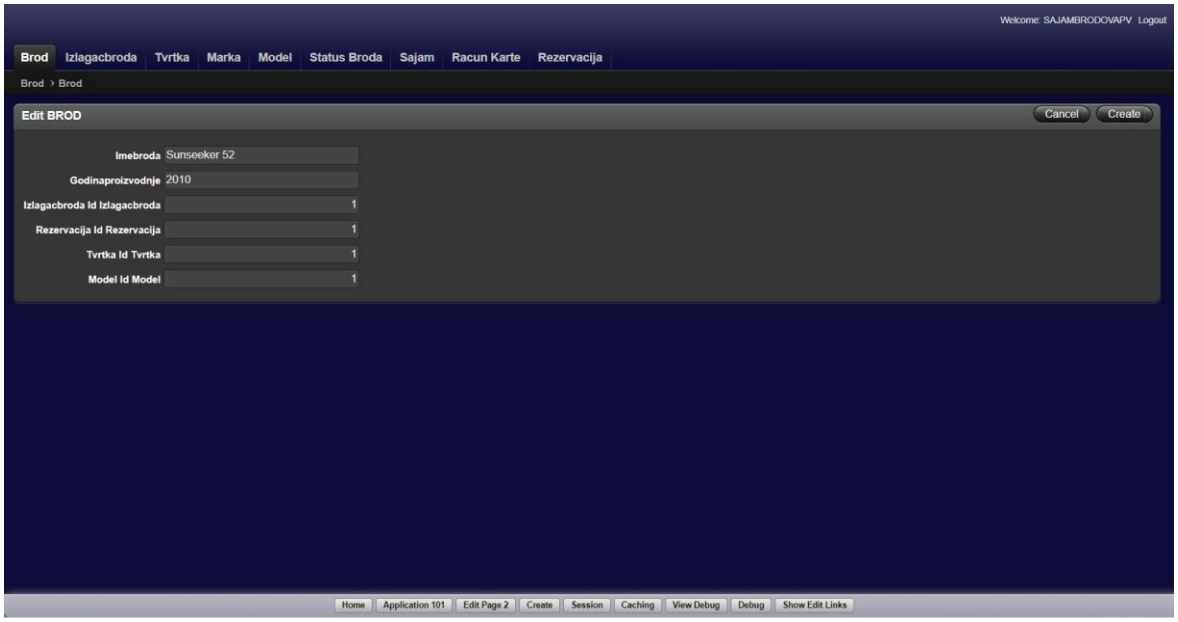
U APEX-u se mogu kreirati izvještaji i forme („Reports and Forms“) što je prikazano na sljedećim slikama.



Slika 3-34 Naslovna stranica "Report and Form"

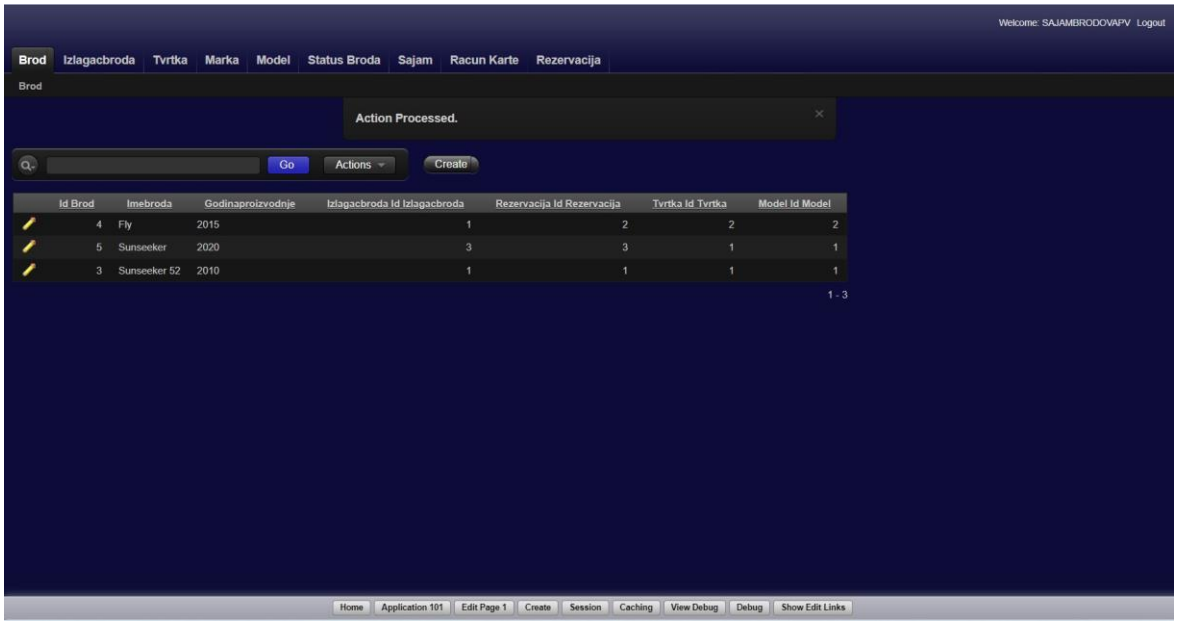
Za ažuriranje podataka potrebno je kliknuti na ikonu olovke, upisati željeno i kliknuti na *„Apply Changes“*.

Moguće je i kreirati novi podatak. U ovom slučaju bilo je potrebno i definirati novi podatak u entitetu Model jer nam je u entitetu Brod potreban ID_Model. Nakon unosa novih podataka potrebno je kliknuti *„Create“*.



Slika 3-35 Dodavanje novog podatka

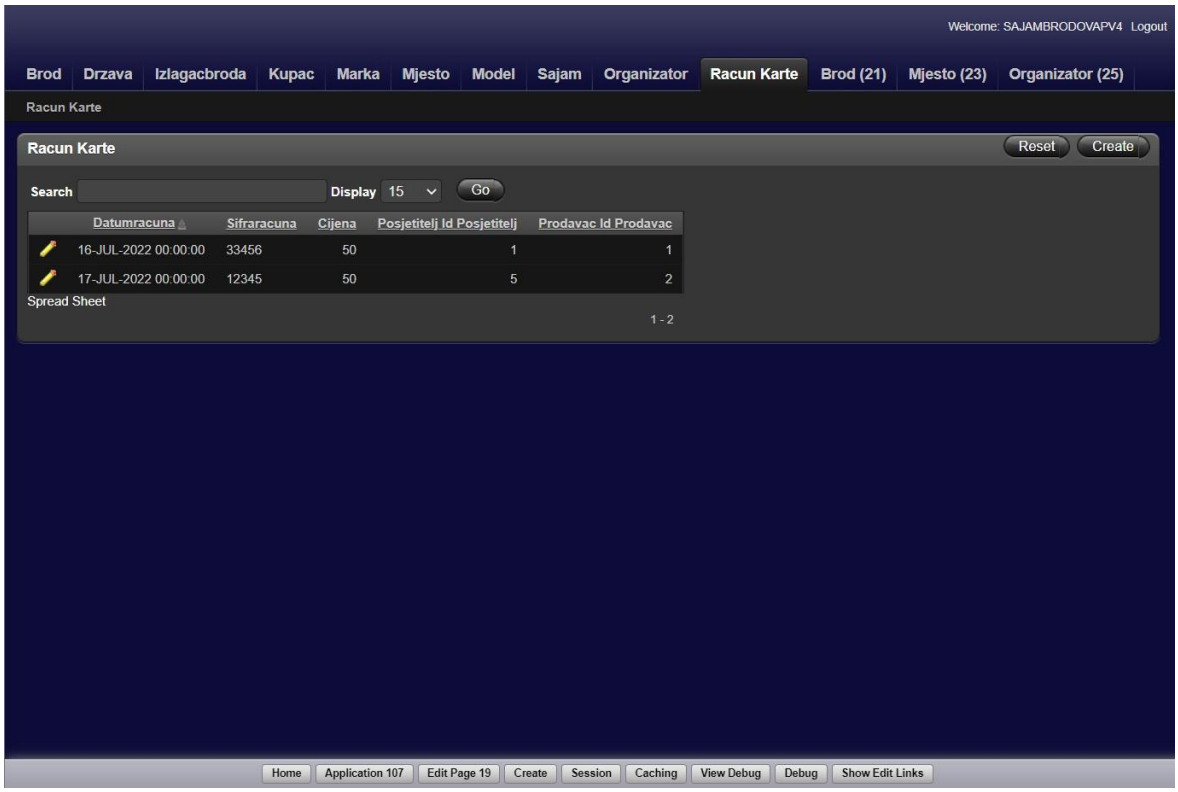
Slika 3-35 prikazuje tablicu sa novim podatkom.



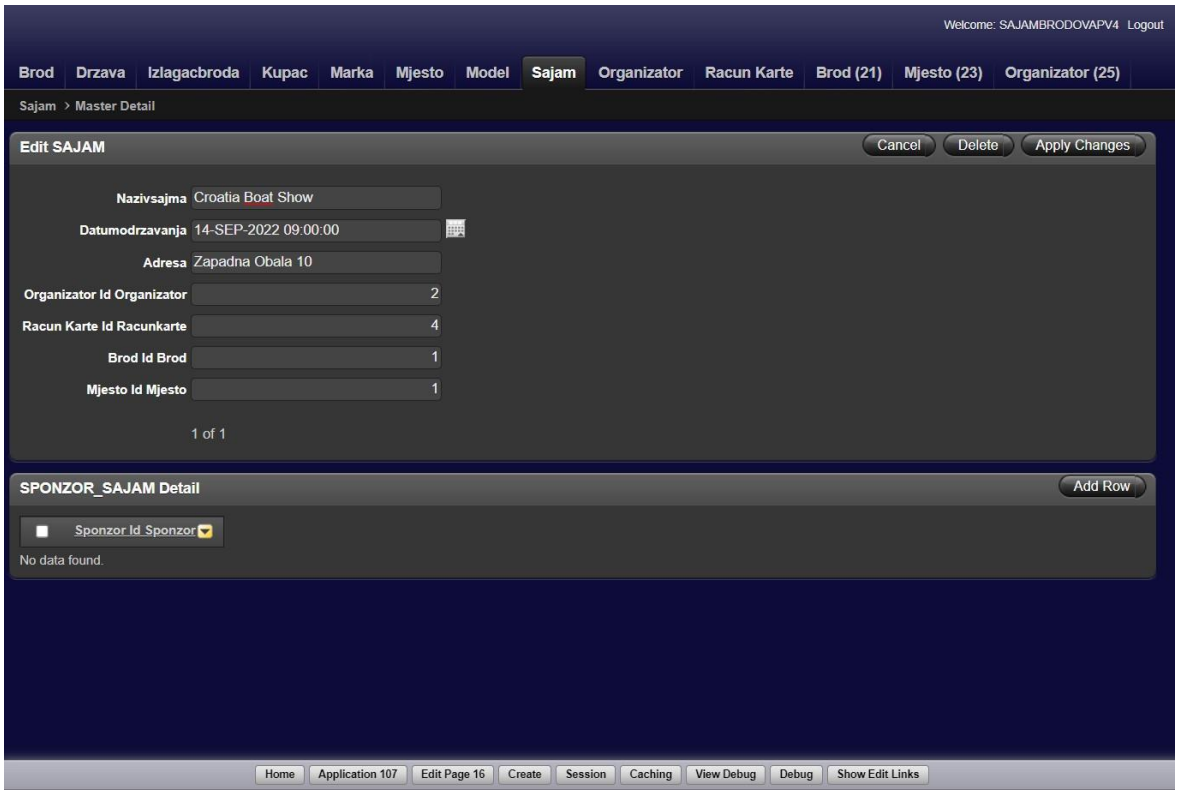
Slika 3-36 Tablica sa novim podatkom

3.7.3. Povezani podaci

U relacijskim bazama podataka česti odnos tablica jedan na prema više što se u APEX-u može jednostavno prikazati pomoću *Master Detail* tipa stranice. *Master* predstavlja „glavnu“ tablicu, a *Detail* „povezane“ podatke. Master/Detail stranica omogućuje promjenu podataka, te uvid u detalje. Detalji se odnose na tablice s kojima je određena tablica direktno vezana.

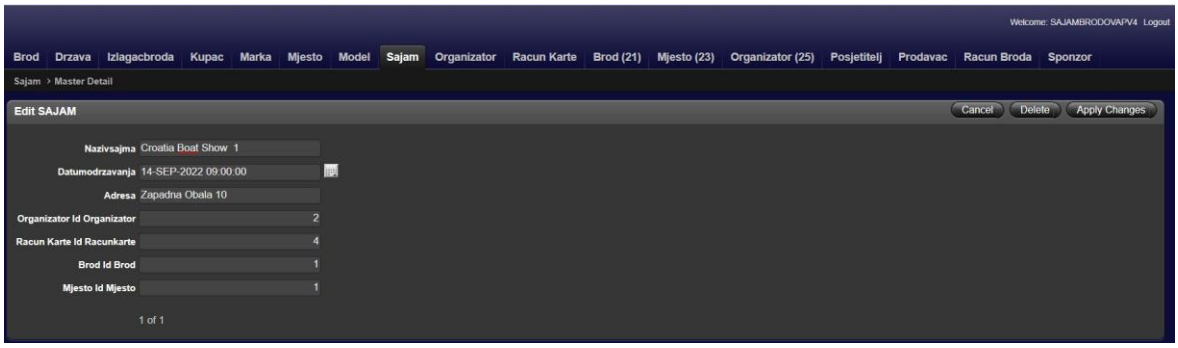


Slika 3-37 Naslovna stranica Master Detail



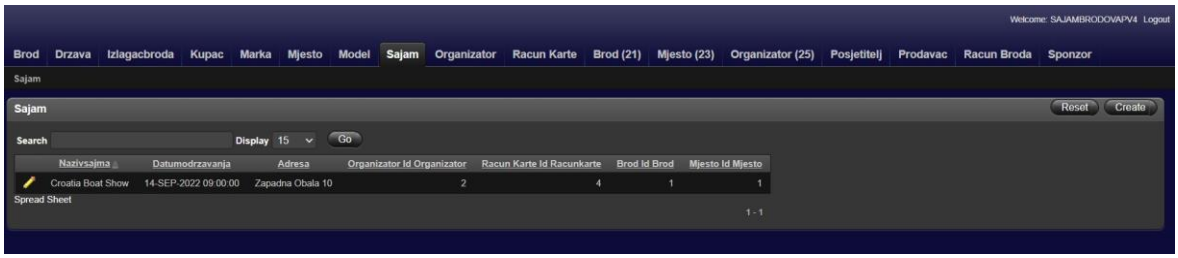
Slika 3-38 Uređivanje u Master Detail-u

Ažuriranje „*Master Detail"*. Prikazano je kako se može promijeniti npr. godina održavanja, tj. naziv iz „*Croatia Boat Show 1*" promijenila u „*Croatia Boat Show*".



Slika 3-39 Ažuriranje podataka

Slika 3-39 prikazuje novi ažurirani podatak.



Slika 3-40 Novi naziv

Zaključak

Baze podataka ključ su svake aplikacije koja postoji. Jako je važno bazu podataka kreirati smisleno i razumljivo kako bi ostalima bila jednostavna za korištenje ili pri implementiranju nekog sustava. Jako je važno tko sve ima pristup bazi podataka i koje uloge ima jer ipak je riječ o nečijim podacima. Na primjer da se provali u bazu podataka neke banke dogodio bi se kolaps cijelog sustava, podaci bi završili u nesigurnim rukama i slično. Upravo zbog takvih stvari potrebno je bazu podataka jako dobro zaštiti na primjer kompliciranim lozinkama.

U ovom završnom radu opisan je Oracle alat i pomoću njega je realizirana izrada ove relacijske baze. Kako bi manipuliranje podacima bilo pobliže prikazano ova relacijska baza implementirana je pomoću *Oracle Application Express* alata.

Sajmovi su jako bitna stavka za unaprjeđenje proizvoda, borbe sa konkurencijom i na kraju krajeva same promocije gradova u kojima se sajam održava. S obzirom na jako puno informacije koje sam sajam brodova sadržava kroz ovaj završni rad odlučila sam realizirati relacijsku bazu podataka o sajmu brodova koja se može implementirati u sustav koji bi sadržavao sve podatke o sajmovima, brodovima koji su bili na sajmu, posjetiteljima, kupcima, organizatorima, sponzorima, tvrtkama i radnicima sajma.

Bazu podataka možemo još nazvati i „kamenom temeljcem“ svakog sustava i svake aplikacije. S obzirom da su podaci zlato 21. stoljeća, te da se gomilaju u golemim serverima tehnoloških giganata kao što je na primjer Facebook koji je do sad imao nekolicinu skandala u vezi curenja podataka, shvaćamo da su baze podataka osnova svakog sustava, te da sa bazama treba znati dobro upravljati.

Literatura

[1] Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, *DATABASE SYSTEMS The Complete Book*, Upper Saddle River, New Jersey, 2002.

[2] Jan Speelpenning, Patrice Daux, Jeff Gallus, *Data Modeling and Relational Database Design*, 2001.

[3] Tonči Carić, Mario Buntić, *Uvod u relacijske baze podataka*, Zagreb, 2015.

[4] Robert Manger, *Osnove projektiranja baza podataka*, 2010.

[5] Dadić, T. (22. Srpanj 2012.). BAZE PODATAKA. Dohvaćeno iz http://mapmf.pmfst.unist.hr/~tdadic/Dadic_BazePodataka.pdf

[6] Wikipedia. *Baza podataka*. Dohvaćeno iz https://hr.wikipedia.org/wiki/Baza_podataka

[7] Quickbase. A Timeline of Database History & Database Management. Dohvaćeno iz https://www.quickbase.com/articles/timeline-of-database-history

[8] Learning Computing History. *Database Systems*. Dohvaćeno iz http://www.comphist.org/computing_history/new_page_9.htm

[9] IBM. Relational Databases. Dohvaćeno iz https://www.ibm.com/cloud/learn/relational-databases

Popis slika

Slika 2-1 Schema razvojnog ciklusa baze podataka 6

Slika 2-2 Veze između entiteta 7

Slika 3-3 Relacijski model 17

Slika 3-4 Omogućeni "Auto Increment" i "Identity Column" za dodavanje sekvenci i okidača 18

Slika 3-5 Dodavanje imena sekvencama i okidačima 19

Slika 3-6 Dodavanje "Unique Constraints" za šifru računa 20

Slika 3-7 Generiranje DDL datoteke 21

Slika 3-8 Kreiranje korisnika i dodavanje ovlasti 22

Slika 3-9 Kod iz DDL-a zalijepljen u Oracle SQL 23

Slika 3-10 Kreiranje tablica i lijevo na slici prikaz tablica 23

Slika 3-11 Kreiranje sekvenci i lijevo na slici prikaz sekvenci 24

Slika 3-12 Kreiranje okidača i lijevo na slici prikaz okidača 24

Slika 3-13 Naredba Insert 25

Slika 3-14 Naredba Update 26

Slika 3-15 Naredba Delete 27

Slika 3-16 Naredba Select 27

Slika 3-17 Apex prijava 28

Slika 3-18 Apex kreiranje baze 29

Slika 3-19 Apex spajanje na korisnika 29

Slika 3-20 Naslovna stranica Application Express-a 30

Slika 3-21 Konfiguracija za web aplikaciju 30

Slika 3-22 Konfiguracija za web stranicu 31

Slika 3-23 Konfiguracija za web aplikaciju 31
Slika 3-24 Dodavanje imena aplikaciji 32
Slika 3-25 Dodavanje stranica - Report and Form 32
Slika 3-26 Pregled dodanih stranica 33
Slika 3-27 Odabir razine tabulatora 33
Slika 3-28 Konfiguracija 34
Slika 3-29 Odabir formata datuma 34
Slika 3-30 Prikaz sučelja za odabir 35
Slika 3-31 Prikaz cjelokupne konfiguracije 35
Slika 3-32 Naslovna stranica nakon konfiguracije 36
Slika 3-33 Naslovna stranica "Report and Form" 36
Slika 3-34 Dodavanje novog podatka 37
Slika 3-35 Tablica sa novim podatkom 37
Slika 3-36 Naslovna stranica Master Detail 38
Slika 3-37 Uređivanje u Master Detail-u 39
Slika 3-38 Ažuriranje podataka 39
Slika 3-39 Novi naziv 39

Skraćenice

DBMS	Database Management System	sustav upravljanja bazom podataka
CIA	*Central Intelligence Agency*	središnja obavještajna agencija
SDL	*Software Development Laboratories*	laboratoriji za razvoj softvera
ERM	*Entity Relationship Model*	logički model
APEX	*Oracle Application Express*	alat u Oraclu (web aplikacija)