

Sveučilište Jurja Dobrile u Puli

Fakultet informatike u Puli



Projektni zadatak

Sustav za upravljanje Game Shopom

Tim 5

Pula, siječanj 2020.

Sadržaj:

| | |
|--|----|
| 1. Uvod | 2 |
| 2. Instalacija potrebnih programa i postavljanje baze | 3 |
| 2.1.Instalacija MySQL Servera za postavljanje baze unutar Oracle SQL Developera | 3 |
| 2.2 Postavljanje MySQL Servera i kreiranje MySQL baze unutar Oracle SQL Developera | 4 |
| 2.3 Spajanje ostalih korisnika na bazu putem Hamachija | 6 |
| 3. Kreiranje tablica i ER dijagram | 8 |
| 3.1. Popis tablica i njihov sadržaj..... | 8 |
| 3.2. Bitne napomene vezane uz tablice..... | 14 |
| 3.3. ER Dijagram | 16 |
| 4. Pogledi..... | 17 |
| 6. Procedure..... | 23 |
| 7. Funkcije | 26 |
| 8. Transakcije | 28 |
| 9. Aplikacija..... | 32 |
| 9.1. Potrebni programi i postupak za uspješno postavljanje aplikacije | 32 |
| 9.2. Izgled aplikacije i testiranje rada | 35 |
| 10. QA – ispitivanje kvalitete i točnosti dokumentacije, baze i aplikacije..... | 51 |
| 11. Zaključak i zahvale | 53 |
| 12. Korisne poveznice..... | 54 |

1. Uvod

Kako je navedeno u prethodnom sažetku projekta, tema našeg projektnog zadatka bila je osmisliti potpuno funkcionalnu bazu podataka za svojevrsni „Game Shop“, te kreirati aplikaciju za lakše upravljanje raznim vrstama upita nad tom bazom.

Iako je naziv našeg poduzeća „Game Shop“, to ipak nije primarno trgovina već smo si kao primarnu zadaću našeg poslovanja odredili iznajmljivanje video igara za sve platforme posljednje generacije, odnosno posudbu nekog željenog naslova na određeni vremenski period (kao što su to nekada radile videoteke za filmove i serije). Naravno, ne bi imalo smisla zvati se Game Shopom ukoliko nešto i ne prodajemo. Stoga, iako nam je primarni cilj iznajmljivanje naslova, nudimo također i otkup starijih naslova po znatno nižim cijenama.

S obzirom na opširnost naše teme cilj nam je otpočetak bio napraviti što je moguće „savršeniju“ bazu u smislu poštivanja ACID pravila i pravila normalnih formi kako bi u kasnijem radu s bazom nailazili na što manje problema ili poteškoća prilikom unosa novih, te izmjene i brisanja postojećih podataka unutar baze.

Za potrebe upravljanja bazom kako je već navedeno izradili smo i prilagođenu C++ aplikaciju, odnosno jednostavno sučelje koje smo povezali s našom MySQL bazom podataka u svrhu lakšeg izvršavanja željenih operacija nad njom, te olakšanja pristupa samim podacima unutar baze.

U postupku izrade naše baze kreirano je ukupno 15 tablica u koje je unesena velika količina podataka. Iz tog razloga kreirane su i određene procedure, funkcije, transakcije, pogledi te okidači kako bi nam što lakše bilo unositi podatke u bazu, te im kasnije pristupati, pregledavati i ukoliko je to potrebno - mijenjati ih.

Svi detalji o pripremi i izradi cjelokupne baze i aplikacije te o radu sa njima biti će više govora u nastavku ovoga dokumenta.

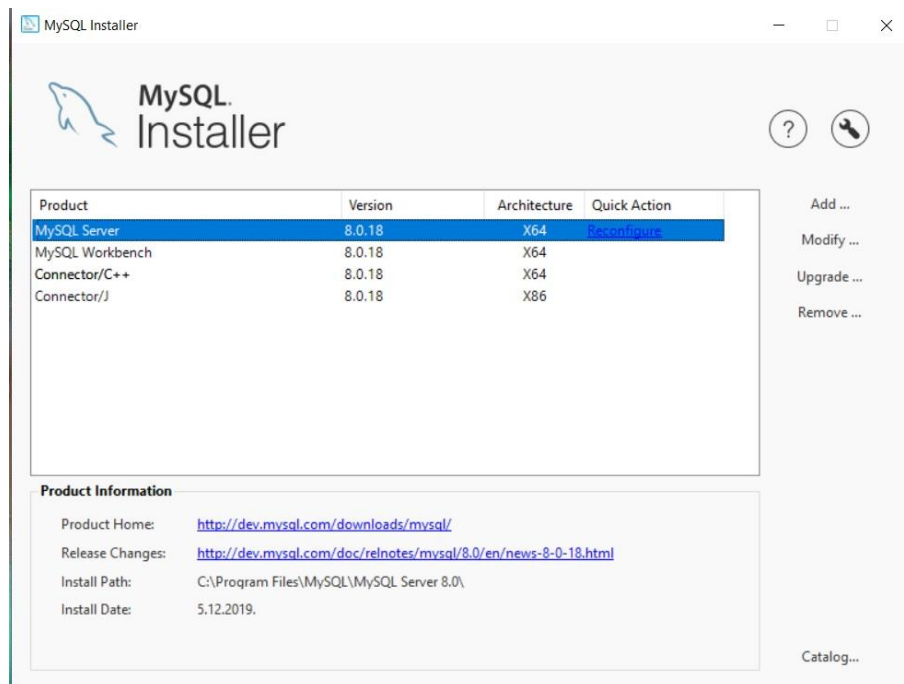
2. Instalacija potrebnih programa i postavljanje baze

2.1. Instalacija MySQL Servera za postavljanje baze unutar Oracle SQL Developera

Za kreiranje naše baze podataka koristili smo program tvrtke Oracle - „Oracle SQL Developer“. Za njega smo se odlučili iz razloga što s njime radimo i na vježbama iz ovoga kolegija, te smo stoga naviknuti na njegovo sučelje i mogućnosti.

Iako smo se odlučili za Oracleov program koji po zadanim vrijednostima koristi i njihovu izvedbu SQL jezika (koja je vjerojatno i najnaprednija izvedba tog jezika), ipak odlučili smo koristiti MySQL izvedbu SQL programskog jezika s obzirom da je ista puno češće korištena na predavanjima iz ovoga kolegija, te je ujedno i navedena kao pogodna za izradu baze unutar dokumenta „Pravila projekta“.

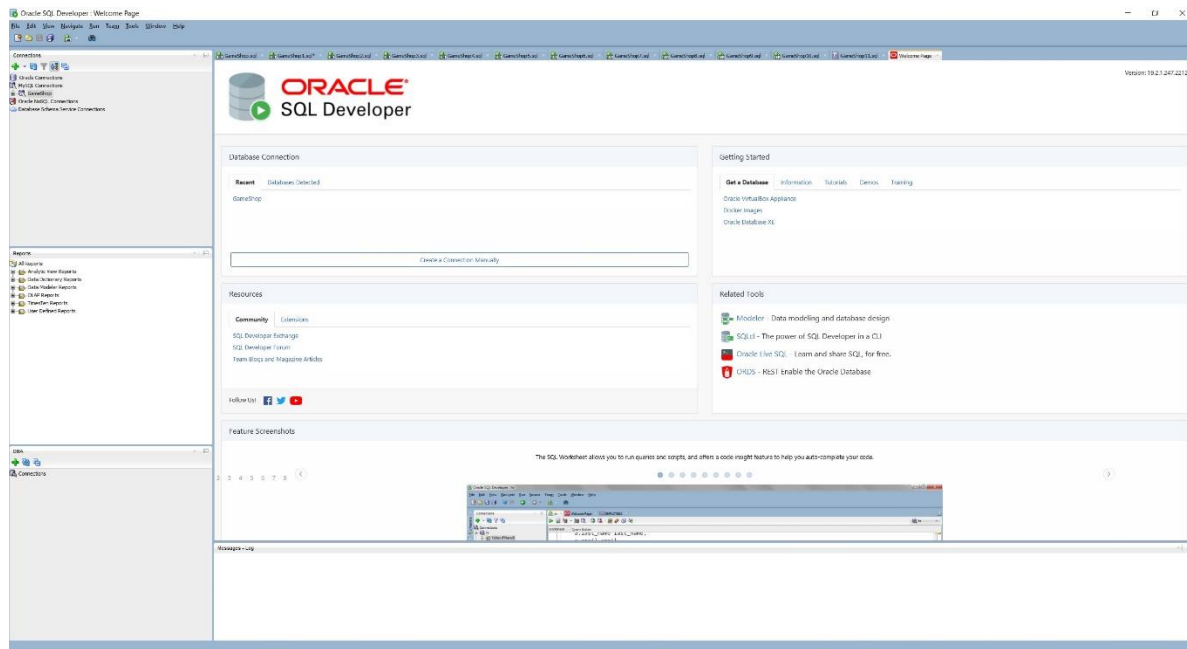
S obzirom da SQL Developer nakon instalacije radi na Oracle serveru tj. s Oracle bazom, trebalo je napraviti određene prilagodbe unutar programa (i naših računala) kako bi on mogao podržavati MySQL bazu. Iz tog razloga morali smo naknadno preuzeti i instalirati program „MySQL Installer“ (Slika 1) preko kojega smo potom preuzeli i instalirali posljednju verziju servera za rad sa MySQL bazama (MySQL Server 8.0), te prikladan „Java konektor“ za povezivanje MySQL Servera sa SQL Developerom.



Slika 1 – Prikaz instaliranih MySQL programa i dodataka unutar MySQL Installera

2.2 Postavljanje MySQL Servera i kreiranje MySQL baze unutar Oracle SQL Developera

Po otvaranju SQL Developera prikazuje nam se njegovo početno sučelje (Slika 2).



Slika 2 – Početni izbornik Oracle SQL Developera

Iako mi već imamo kreiranu našu bazu, probati ćemo za potrebu ove dokumentacije kreirati neku drugu bazu sa drugačijim imenom kako bi prikazali postupak postavljanja baze unutar Oracle SQL Developera. Pri kreiranju pokušali smo kreirati što sličniju bazu našoj bazi „Game Shop“ kako bi približno prikazali na koji način se i naša baza kreirala.

Ipak, prije ikakvoga kreiranja baza unutar SQL Developera, morali smo odrediti računalo koje će nam služiti kao svojevrsni server i na kojemu ćemo potom postaviti sve potrebne podatke i postavke unutar MySQL Servera kao što su npr. dodjele administratorskih ili ograničenih prava korisnicima, korisnička imena i lozinke za pristup serveru tj. bazi, te ostale manje bitne stavke.

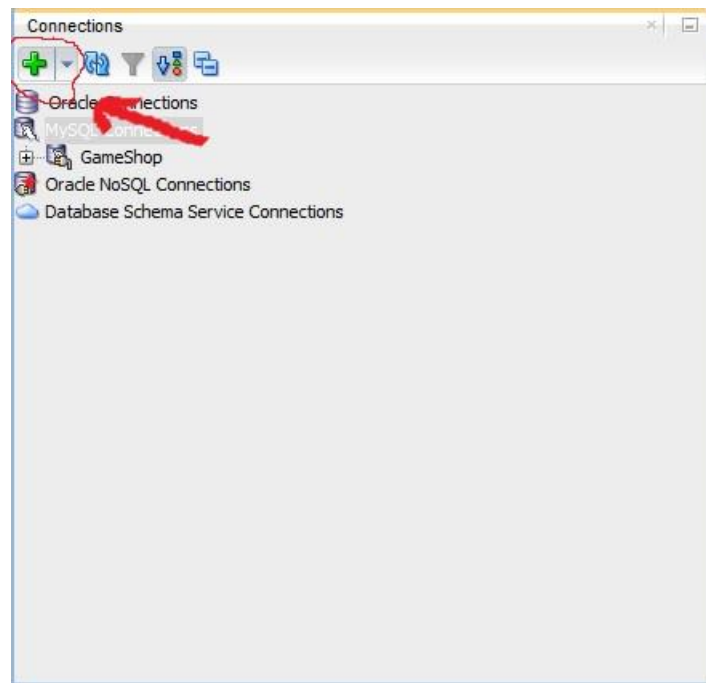
Dogovorili smo da će server-računalo biti računalo kolege Đurašinovića s obzirom da je on jedini iz programerskog dijela tima konstantno u Puli, te posjeduje novo stolno računalo koje će stoga moći i najbolje „podnijeti“ bazu i sve nadolazeće konekcije. To nam je savršeno odgovaralo pošto smo imali u planu spajati se s udaljenih i različitih mreža na mrežu tog jednog „server računala“. Drugim riječima, učinili smo to kako bi izbjegli potrebu stalnoga sastajanja za rad na bazi, već smo na ovaj način ostvarili mogućnost simultanog (paralelnog) rada na bazi sa više različitih računala i s više različitih udaljenih mreža. Taj dio samoga spajanja sa vanjskih mreža na mrežu našeg servera nam je u početku zadao određene glavobolje i probleme, ali o tome više u nastavku teksta.

Nažalost slike postavljanja MySQL Servera ne možemo priložiti s obzirom da MySQL Server dopušta postavljanje bitnih postavki (kao npr. postavljanje korisnika i lozinki)

isključivo prilikom prvog postavljanja servera, odnosno ne dopušta kasnije modifikacije po tom pitanju, osim u vidu potpunog brisanja i postavljanja novoga servera, što ipak za potrebe slika nećemo uraditi kako ne bi ugrozili stabilnost i sigurnost podataka, te korisničkih profila unutar naše baze.

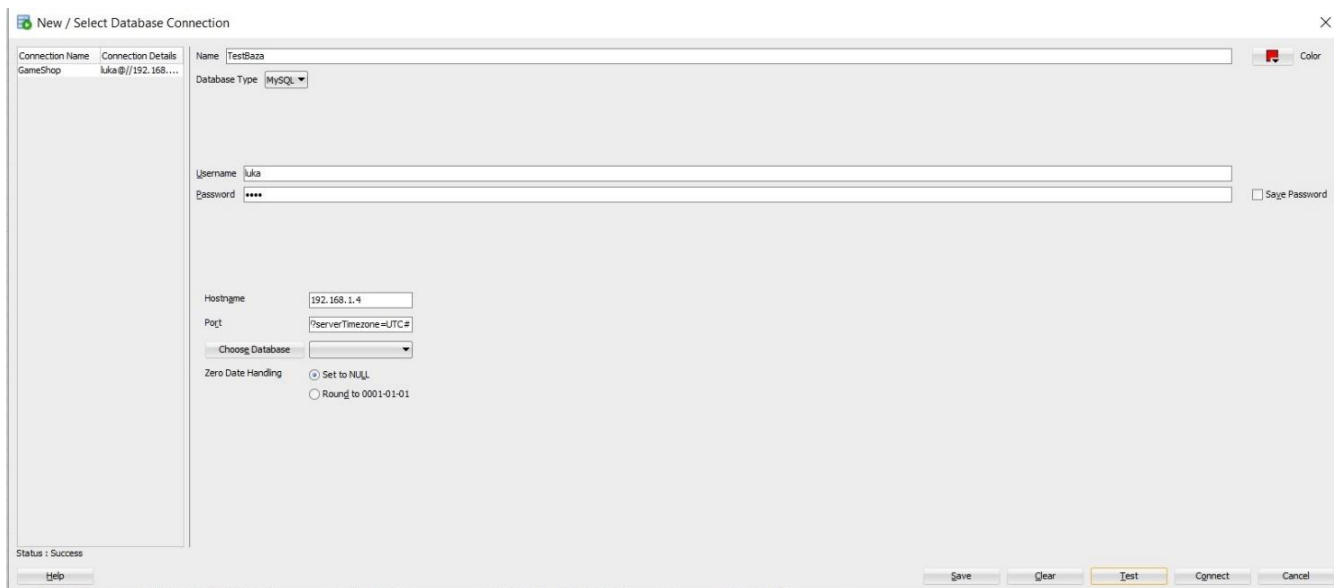
Nakon postavljanja svih željenih profila i postavki unutar MySQL Servera, možemo konačno kreirati i postaviti bazu u SQL Developeru.

Na početnom izborniku SQL Developera u gornjem lijevom kutu vidimo zeleni znak plusa (Slika 3), klikom na njega otvaramo opciju za dodavanje nove baze.



Slika 3 – Gumb za otvaranje izbornika za kreiranje nove baze podataka

Nakon što kliknemo na gumb za dodavanje nove konekcije tj. baze, dobivamo izbornik u kome postavljamo ime baze podataka (konekcije), tip baze podataka (u ovom slučaju mijenjamo vrijednost iz Oracle u MySQL), unosimo korisničko ime i lozinku (naše korisničko ime i lozinku koje smo prethodno kreirali u MySQL Serveru), te u hostname unosimo IP adresu server računala (na koje se želimo spajati) i port koji ostavljamo najčešće na „default“ vrijednosti 3306 (Slika 4).



Slika 4 – Izgled izbornika za kreiranje nove baze unutar SQL Developera

Nakon što unesemo sve potrebne vrijednosti klikom na „Test“ gumb u donjem desnom uglu izbornika testiramo može li se baza spojiti sa serverom. Ukoliko nam nakon klika na „Test“ program vrati „Status: Success“, možemo slobodno kliknuti na gumb „Connect“ i na taj način kreirati bazu.

2.3 Spajanje ostalih korisnika na bazu putem Hamachija

Kao što je prije navedeno cilj nam je bio povezivati se na bazu s vanjskih mreža, odnosno spajanje bilo kojeg računala iz bilo kojeg kraja svijeta na našu bazu podataka, naravno uz prethodnu točnu autentifikaciju, odnosno uz unos valjanog korisničkog imena i lozinke za pristup našoj bazi.

Takav pristup bazi postavili smo ponajviše iz razloga što smo znali da će za vrijeme zimskih praznika mnogi članovi tima oputovati svojim kućama (pošto nisu iz Pule), te neće biti dostupni za sastajanje i zajednički rad na bazi. Ipak, takav pristup postavljanju baze nam je uzrokovao i određene probleme, ponajviše u vidu spajanja vanjskih računala (računala ostalih članova tima) na IP adresu server-računala.

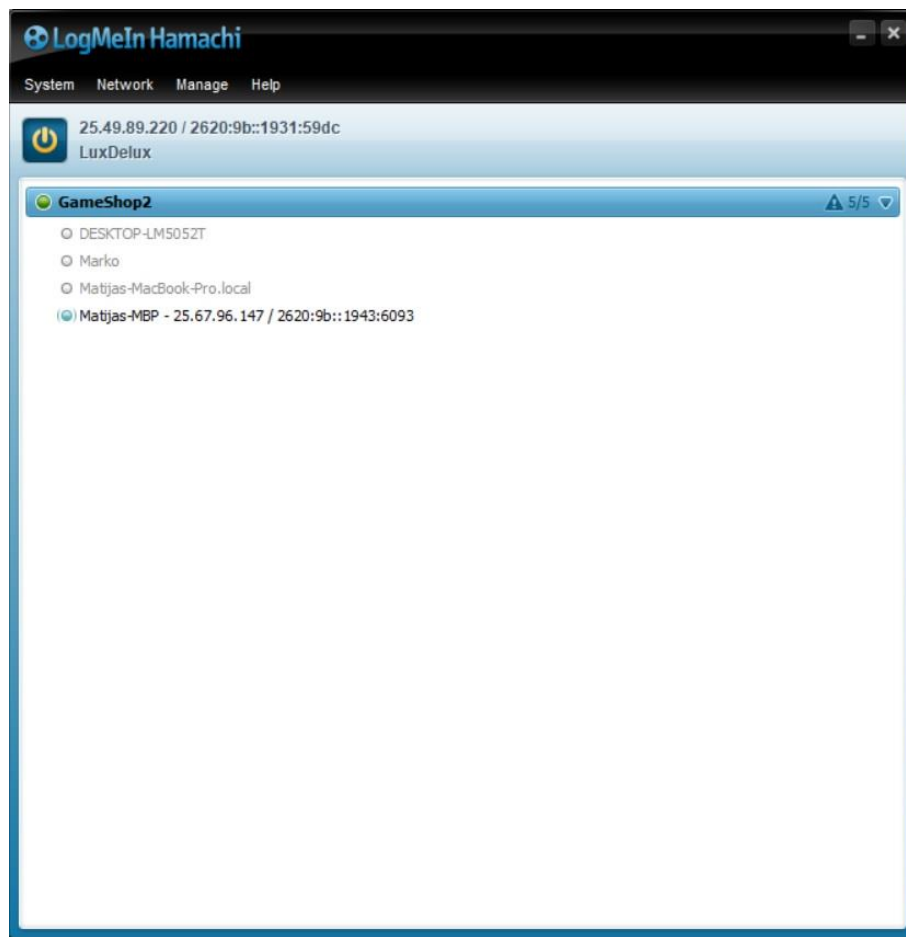
Naime, pribojavali smo se da bi nam se takvi problemi mogli dogoditi, a oni su se naposljetku i dogodilo – nijedno računalo sa mreže različite od mreže server-računala se nije moglo spojiti direktno na IP adresu server-računala.

Pokušavali smo to riješiti na više načina – otvaranjem određenih portova, blokiranjem Firewalla na računalima, korištenjem „vanjskih“ IP adresa računala ... Ipak, iako smo pokušavali štošta pokušavali – ništa nije urodilo plodom. Pretpostavljamo, zbog nekakvog CGNAT-a nad mrežom server-računala, ali to u ovoj priči i nije toliko bitno.

Već smo mislili odustati od takvog načina rada na bazi i naći neku drugu soluciju (npr. postavljanja baze na neki online server i rada preko njega), ali u zadnji tren se kolega Čunj sjetio da možemo koristiti neku vrstu VPN aplikacije ili svojevrsnog „remote servera“. Tu smo se svi složili da bi mogli isprobati hoćemo li se uspjeti spojiti na bazu preko „Hamachija“.

Hamachi je vrsta aplikacije koja služi kao neka vrsta „virtualnog servera“, odnosno ona premošćuje sve Firewall i NAT blokade pri direktnom povezivanju računala iz različitih mreža dodijeljujući svakom računalu njegovu specifičnu Hamachi IP adresu, odnosno IP adresu za spajanje putem same aplikacije.

Tako smo svi instalirali na svoja računala Hamachi, te postavili računalo kolege Đurašinovića kao server računalo na Hamachiju (Slika 5).



Slika 5 – Izgled Hamachi sučelja i popis računala dodijeljenih željenoj mreži

Nakon što je Hamachi postavljen kolege su se uspješno spajale na bazu odakle god su željele, uz uvjet da je server-računalo bilo upaljeno i spojeno na Hamachi.

To je značilo da kreiranje samih tablica i punjenje baze podacima može početi.

3. Kreiranje tablica i ER dijagram

3.1. Popis tablica i njihov sadržaj

Prije svake nove “seanse”, odnosno prilikom svakog novog paljenja SQL Developera i spajanja u bazu prva naredba koju uvijek moramo unijeti je:

USE gameshop;

Sa tom naredbom javljamo programu da želimo koristiti baš tu bazu podataka (u slučaju da ih imamo više) i nad njom vršiti željene operacije.

Za potrebu kreiranja baze kako bi si olakšali početno kreiranje tablica i njihovo međusobno povezivanje naši programeri su izradili “privremeni” ER dijagram po kojemu su kreirali početni izgled baze, odnosno par početnih najbitnijih tablica.

Naravno, postepenom izradom baze, odnosno kreiranjem i punjenjem tablica uviđali su njihove eventualne mane i nedostatke. Tako je osnovni koncept zamišljene baze tijekom izrade i punjenja baze podacima naišao na brojne promjene i prilagodbe. Na kraju se od početno zamišljenih oko 10 tablica došlo na konačni broj od 15 tablica unutar naše baze.

Popis, kao i osnovna funkcija te sadržaj tablica unutar baze je slijedeći:

- **Kupac** – tablica od 7 stupaca (Slika 6), govori nam osnovne i najbitnije informacije o našim kupcima, te od strane kojeg su zaposlenika posluženi/registrirani

| COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|-----------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 KupacID | 1 (null) | NO | int | 10 | 0 (null) | | |
| 2 ZaposlenikID | 2 (null) | NO | int | 10 | 0 (null) | | |
| 3 Ime | 3 (null) | NO | varchar | (null) | (null) | (null) | |
| 4 Prezime | 4 (null) | NO | varchar | (null) | (null) | (null) | |
| 5 Email | 5 (null) | NO | varchar | (null) | (null) | (null) | |
| 6 Aktivan | 6 (null) | NO | char | (null) | (null) | (null) | |
| 7 Datum Dolaska | 7 (null) | NO | date | (null) | (null) | (null) | |

Slika 6 – Tablica „Kupac“ kako je pikazana unutar SQL Developera, sa pripadnim stupcima i dodijeljenim tipovima podataka

- **Adresa** – tablica od 5 stupaca (Slika 7), svojevrsna „podtablica“ tablice kupac, govori nam neke za poslovanje „manje“ bitne podatke o našim kupcima

| COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|-----------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 KupacID | 1 (null) | NO | int | 10 | 0 (null) | | |
| 2 Adresa | 2 (null) | NO | varchar | (null) | (null) | (null) | |
| 3 Grad | 3 (null) | NO | varchar | (null) | (null) | (null) | |
| 4 Drzava | 4 (null) | NO | varchar | (null) | (null) | (null) | |
| 5 Broj_telefona | 5 (null) | NO | varchar | (null) | (null) | (null) | |

Slika 7- Tablica „Adresa“

- **Poslovnica** – tablica od 5 stupaca (Slika 8), govori nam osnovne informacije o naših pet poslovnica

| COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|------------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 ID | 1 (null) | NO | int | 10 | 0 (null) | | |
| 2 Grad | 2 (null) | NO | varchar | (null) | (null) | (null) | |
| 3 Adresa | 3 (null) | NO | varchar | (null) | (null) | (null) | |
| 4 Zupanija | 4 (null) | NO | varchar | (null) | (null) | (null) | |
| 5 Postanski_broj | 5 (null) | NO | varchar | (null) | (null) | (null) | |

Slika 8 – Tablica „Poslovnica“

- **Režije** – tablica od 10 stupaca (Slika 9), govori nam troškove svih mjesečnih režija i cijene najma po poslovnica

| COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|----------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 ID | 1 (null) | NO | int | 10 | 0 (null) | | |
| 2 PoslovnicaID | 2 (null) | NO | int | 10 | 0 (null) | | |
| 3 Godina | 3 (null) | NO | varchar | (null) | (null) | (null) | |
| 4 Mjesec | 4 (null) | NO | varchar | (null) | (null) | (null) | |
| 5 Najamnina | 5 (null) | NO | int | 10 | 0 (null) | | |
| 6 Struja | 6 (null) | NO | int | 10 | 0 (null) | | |
| 7 Voda | 7 (null) | NO | int | 10 | 0 (null) | | |
| 8 Plin | 8 (null) | NO | int | 10 | 0 (null) | | |
| 9 Internet | 9 (null) | NO | int | 10 | 0 (null) | | |
| 10 Telefon | 10 (null) | NO | int | 10 | 0 (null) | | |

Slika 9 – Tablica „Režije“

- **Poslovođa** - tablica od 9 stupaca (Slika 10), govori nam sve važne informacije za sve poslovođe naših poslovnica

| COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|------------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 ID | 1 (null) | NO | int | 10 | 0 (null) | | |
| 2 sifra | 2 (null) | NO | int | 10 | 0 (null) | | |
| 3 PoslovnicaID | 3 (null) | NO | int | 10 | 0 (null) | | |
| 4 Adresa | 4 (null) | NO | varchar | (null) | (null) | (null) | |
| 5 Ime | 5 (null) | NO | varchar | (null) | (null) | (null) | |
| 6 Prezime | 6 (null) | NO | varchar | (null) | (null) | (null) | |
| 7 Placa | 7 (null) | NO | varchar | (null) | (null) | (null) | |
| 8 Email | 8 (null) | NO | varchar | (null) | (null) | (null) | |
| 9 Korisnicko_ime | 9 (null) | NO | varchar | (null) | (null) | (null) | |

Slika 10 – Tablica „Poslovođa“

- **Zaposlenik** – tablica od 11 stupaca (Slika 11), govori nam sve važne informacije o našim zaposlenicima, uključujući i tko im je poslovođa, u kojoj poslovnicu rade, kao i njihovo korisničko ime i šifru za pristup blagajni

| COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|-------------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 ID | 1 (null) | NO | int | 10 | 0 (null) | | |
| 2 PoslovodaSifra | 2 (null) | NO | int | 10 | 0 (null) | | |
| 3 PoslovnicaID | 3 (null) | NO | int | 10 | 0 (null) | | |
| 4 Grad_Stanovanja | 4 (null) | NO | varchar | (null) | (null) | (null) | |
| 5 Adresa | 5 (null) | NO | varchar | (null) | (null) | (null) | |
| 6 Ime | 6 (null) | NO | varchar | (null) | (null) | (null) | |
| 7 Prezime | 7 (null) | NO | varchar | (null) | (null) | (null) | |
| 8 Placa | 8 (null) | NO | varchar | (null) | (null) | (null) | |
| 9 Email | 9 (null) | NO | varchar | (null) | (null) | (null) | |
| 10 Korisnicko_ime | 10 (null) | NO | varchar | (null) | (null) | (null) | |
| 11 Sifra | 11 (null) | NO | varchar | (null) | (null) | (null) | |

Slika 11 – Tablica „Zaposlenik“

- **Proizvod** – tablica od 8 stupaca (Slika 12), govori nam sve najvažnije informacije o našem asortimanu proizvoda (igre koje imamo u ponudi)

| COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|--------------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 ID | 1 (null) | NO | int | 10 | 0 (null) | | |
| 2 Sifra_Proizvoda | 2 (null) | NO | varchar | (null) | (null) | (null) | |
| 3 Naziv | 3 (null) | NO | varchar | (null) | (null) | (null) | |
| 4 Zanzr | 4 (null) | NO | varchar | (null) | (null) | (null) | |
| 5 Izdavac | 5 (null) | NO | varchar | (null) | (null) | (null) | |
| 6 Platforma | 6 (null) | NO | varchar | (null) | (null) | (null) | |
| 7 Godina_izlaska | 7 (null) | NO | int | 10 | 0 (null) | | |
| 8 Cijena_za_kupnju | 8 (null) | NO | varchar | (null) | (null) | (null) | |

Slika 12- Tablica „Proizvod“

- **Posuđeno** – tablica od 8 stupaca (Slika 13), govori nam sve bitne i potrebne informacije o posuđenim proizvodima, kao i datum kada su posuđeni te rok za vraćanje proizvoda bez plaćanja zakasnine

| COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|------------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 ID | 1 (null) | NO | int | 10 | 0 (null) | | |
| 2 ProizvodID | 2 (null) | NO | int | 10 | 0 (null) | | |
| 3 RacunID | 3 (null) | NO | int | 10 | 0 (null) | | |
| 4 datum_posudbe | 4 (null) | NO | date | (null) | (null) | (null) | |
| 5 datum_roka | 5 (null) | NO | date | (null) | (null) | (null) | |
| 6 cijena_posudbe | 6 (null) | NO | varchar | (null) | (null) | (null) | |
| 7 poslovnicaid | 7 (null) | NO | int | 10 | 0 (null) | | |
| 8 Platforma | 8 (null) | NO | varchar | (null) | (null) | (null) | |

Slika 13 – Tablica „Posuđeno“

- **Prodano** – tablica od 7 stupaca (Slika 14), govori nam sve bitne informacije o prodanim proizvodima, kao i cijenu svakog prodanog proizvoda

| COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|----------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 ID | 1 (null) | NO | int | 10 | 0 (null) | | |
| 2 ProizvodID | 2 (null) | NO | int | 10 | 0 (null) | | |
| 3 RacunID | 3 (null) | NO | int | 10 | 0 (null) | | |
| 4 Kolicina | 4 (null) | NO | int | 10 | 0 (null) | | |
| 5 cijena | 5 (null) | NO | varchar | (null) | (null) | (null) | |
| 6 Platforma | 6 (null) | NO | varchar | (null) | (null) | (null) | |
| 7 PoslovnicaID | 7 (null) | YES | int | 10 | 0 (null) | | |

Slika 14 – Tablica „Prodano“

- **Račun** – tablica od 10 stupaca (Slika 15), govori nam sve bitne informacije što se tiču obavljene transakcije, kao i to je li određeni proizvod posuđen ili prodan

| COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|--------------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 ID | 1 (null) | NO | int | 10 | 0 (null) | | |
| 2 KupacID | 2 0 | NO | int | 10 | 0 (null) | | |
| 3 ZaposlenikID | 3 0 | NO | int | 10 | 0 (null) | | |
| 4 ProizvodID | 4 0 | NO | int | 10 | 0 (null) | | |
| 5 PoslovnicaID | 5 0 | NO | int | 10 | 0 (null) | | |
| 6 Platforma | 6 (null) | NO | varchar | (null) | (null) | (null) | |
| 7 Cijena | 7 (null) | NO | varchar | (null) | (null) | (null) | |
| 8 Kolicina | 8 (null) | NO | int | 10 | 0 (null) | | |
| 9 Posudeno_Prodano | 9 0 | NO | varchar | (null) | (null) | (null) | |
| 10 Datum_Placanja | 10 (null) | NO | date | (null) | (null) | (null) | |

Slika 15 – Tablica „Račun“

- **Vraćene igre** – tablica od 4 stupca (Slika 16), puni se putem procedure (više govora o tome kasnije), sadrži informacije o igrama koje su vraćene u određenu poslovnicu

| COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|----------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 ProizvodID | 1 (null) | NO | int | 10 | 0 (null) | | |
| 2 RacunID | 2 (null) | NO | int | 10 | 0 (null) | | |
| 3 PoslovnicaID | 3 (null) | NO | int | 10 | 0 (null) | | |
| 4 Platforma | 4 (null) | NO | varchar | (null) | (null) | (null) | |

Slika 16 – Tablica „Vraćene igre“

- **Stanje poslovnica Nintendo** – tablica od 5 stupaca (Slika 17), djelomično punjena putem „privremenog“ okidača, govori nam stanje na skladištu svake od poslovnica za listu naslova za Nintendo platformu

| COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|---------------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 ProizvodID | 1 (null) | NO | int | 10 | 0 (null) | | |
| 2 Naziv | 2 (null) | NO | varchar | (null) | (null) | (null) | |
| 3 siFraPoslNintendo | 3 (null) | NO | int | 10 | 0 (null) | | |
| 4 Dostupnost | 4 (null) | NO | varchar | (null) | (null) | (null) | |
| 5 Kolicina | 5 (null) | NO | int | 10 | 0 (null) | | |

Slika 17 – Tablica „Stanje poslovnica Nintendo“

- **Stanje poslovnica PS4** – tablica od 5 stupaca (Slika 18), također kao tablica za stanje poslovnica za Nintendo platformu punjena je putem „privremenog“ okidača, te nam govori stanje na skladištu svih proizvoda u svakoj poslovnici za Playstation 4 platformu

| | COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|---|--------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 | ProizvodID | 1 | (null) | NO | int | 10 | 0 | (null) |
| 2 | Naziv | 2 | (null) | NO | varchar | (null) | (null) | (null) |
| 3 | SifraPoslPS4 | 3 | (null) | NO | int | 10 | 0 | (null) |
| 4 | Dostupnost | 4 | (null) | NO | varchar | (null) | (null) | (null) |
| 5 | Kolicina | 5 | (null) | NO | int | 10 | 0 | (null) |

Slika 18 – Tablica „Stanje poslovnica PS4“

- **Stanje poslovnica Xbox** – tablica od 5 stupaca (Slika 19), na isti način popunjena kao i prethodne dvije tablice, govori nam stanja naslova na skladištima za Xbox One konzolu

| | COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|---|---------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 | ProizvodID | 1 | (null) | NO | int | 10 | 0 | (null) |
| 2 | Naziv | 2 | (null) | NO | varchar | (null) | (null) | (null) |
| 3 | SifraPoslXBOX | 3 | (null) | NO | int | 10 | 0 | (null) |
| 4 | Dostupnost | 4 | (null) | NO | varchar | (null) | (null) | (null) |
| 5 | Kolicina | 5 | (null) | NO | int | 10 | 0 | (null) |

Slika 19 – Tablica „Stanje poslovnica Xbox“

- **Narudžbenice** – tablica od 4 stupca (Slika 20), tablica u kojoj su pohranjene sve narudžbenice koje se automatski kreiraju kada stanje nekog proizvoda na nekom skladištu padne na nulu

| | COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE | NUMERIC_PRECISION | NUMERIC_SCALE | COLUMN_COMMENT |
|---|--------------|------------------|----------------|-------------|-----------|-------------------|---------------|----------------|
| 1 | ID | 1 | (null) | NO | int | 10 | 0 | (null) |
| 2 | ProizvodID | 2 | (null) | NO | int | 10 | 0 | (null) |
| 3 | PoslovnicaID | 3 | (null) | NO | int | 10 | 0 | (null) |
| 4 | Platforma | 4 | (null) | NO | varchar | (null) | (null) | (null) |

Slika 20 – Tablica „Narudžbenice“

Ovo je bila potpuna lista sa svim shemama naših tablica u bazi. Same podatke unutar ovih tablica nećemo trenutno prikazivati, odnosno više prikaza samog sadržaja tj. unosa unutar tablica vidjeti ćemo kasnije u odlomku gdje smo testirali funkcionalnost naše aplikacije kreirane za upravljanje bazom. Ipak, kako bi vidjeli kako prikaz sadržaja unutar tablica izgleda u samom SQL Developeru, na donjoj slici možemo vidjeti dio unesenoga sadržaja u tablici “Proizvod” (Slika 20).

| ID | Sifra_Proizvoda | Naziv | Zavr | Izdavac | Platforma | Godina_dolaska | Ocjena_za_kupcu |
|----|-----------------|---|--------------------|-----------------------------|---------------------------------|----------------|-----------------|
| 25 | 25 BATTLEFIELD | Battlefield V | FPS | Electronic Arts | PS 4, Xbox One | 2018 x | |
| 26 | 26 BIOSHOCK | Bioshock: The Collection | FPS | 2K Games | PS 4, Xbox One | 2016 90 km | |
| 27 | 27 BORDER | Borderlands: Game of the Year Edition | Akcijski RPG, FPS | 2K Games | PS 4, Xbox One | 2019 x | |
| 28 | 28 BORDER | Borderlands: The Handsome Collection | Akcijski RPG, FPS | 2K Games | PS 4, Xbox One | 2015 60 km | |
| 29 | 29 BURNOUT | Burnout Paradise Remastered | Utrkivanje | Electronic Arts | PS 4, Xbox One | 2018 x | |
| 30 | 30 COD | Call of Duty: Black Ops III | FPS | Activision | PS 4, Xbox One | 2015 60 km | |
| 31 | 31 COD | Call of Duty: Infinite Warfare | FPS | Activision | PS 4, Xbox One | 2016 60 km | |
| 32 | 32 COD | Call of Duty: Modern Warfare | FPS | Activision | PS 4, Xbox One | 2019 x | |
| 33 | 33 COD | Call of Duty: Modern Warfare Remastered | FPS | Activision | PS 4, Xbox One | 2016 60 km | |
| 34 | 34 COD | Call of Duty: WWII | FPS | Activision | PS 4, Xbox One | 2017 60 km | |
| 35 | 35 CAT | Cat Quest | Akcijski RPG | FQube | PS 4, Nintendo Switch | 2017 65 km | |
| 36 | 36 CAT | Cat Quest 2 | Akcijski RPG | FQube | PS 4, Nintendo Switch | 2019 x | |
| 37 | 37 CEL | Celeste | Platformer | Matt Makes Games | PS 4, Xbox One, Nintendo Switch | 2018 x | |
| 38 | 38 BEYOSOU | Beyond: Two Souls | Interaktivna drama | Sony Computer Entertainment | PS 4 | 2013 50 km | |
| 39 | 39 BLOOD | Bloodborne | Akcijski RPG | Sony Computer Entertainment | PS 4 | 2015 60 km | |
| 40 | 40 BORDER | Borderlands 3: Game of the Year Edition | Akcijski RPG, FPS | 2K Games | PS 4, Xbox One | 2019 x | |
| 41 | 41 CATHERINE | Catherine: Full Body | Puzzle Platformer | Monami | PS 4 | 2019 x | |

Slika 21 – Prikaz dijela sadržaja tablice “Proizvod”

Svi kodovi koji su korišteni za kreiranje gore navedenih tablica su priloženi uz ovu dokumentaciju, ali čisto primjera radi na donjoj slici možemo vidjeti kod koji je korišten za kreiranje tablice “Kupac” (Slika 21). Sam kod izvučen je iz “Backup Filea” naše baze, te stoga može sadržavati neke minimalne “dodatke” dodane od strane samoga programa koji se razlikuju od našeg originalnoga koda, ali u principu on je skoro pa identičan.

```
-- Table `gameshop`.`kupac`
--
CREATE TABLE IF NOT EXISTS `gameshop`.`kupac` (
  `KupacID` INT(11) NOT NULL AUTO_INCREMENT,
  `ZaposlenikID` INT(11) NOT NULL,
  `Ime` VARCHAR(255) NOT NULL,
  `Prezime` VARCHAR(255) NOT NULL,
  `Email` VARCHAR(50) NOT NULL,
  `Aktivan` CHAR(1) NOT NULL,
  `Datum_Dolaska` DATE NOT NULL,
  PRIMARY KEY (`KupacID`),
  INDEX `str_ZaposlenikID_Kupac` (`ZaposlenikID` ASC) VISIBLE,
  CONSTRAINT `str_ZaposlenikID_Kupac`
    FOREIGN KEY (`ZaposlenikID`)
    REFERENCES `gameshop`.`zaposlenik` (`ID`))
```

Slika 21 – Programski kod za kreiranje tablice “Kupac”

3.2. Bitne napomene vezane uz tablice

Iako je više manje sve jasno vidljivo o tablicama naše baze iz gore priloženih slika, neke pojedinosti ipak smatramo da treba dodatno objasniti i napomenuti.

To je sljedeće:

- Tablice (kao i kasnije procedure, funkcije, okidači, pogledi i transakcije) u originalnom nazivlju unutar baze nemaju znakove „č“, „ž“, „š“, kao ni klasične razmaknice u imenu, već se za razmaknice koristi „_“, npr. tablica „Vraćene igre“ se u našoj bazi originalno zove „Vracene_igre“, diakritički znakovi i razmaci su dodani isključivo za potrebu ovoga dokumenta kako bi ostali što više „u duhu hrvatskog jezika“
- Tablica „Vraćene igre“ može se puniti na klasičan „INSERT INTO“ način, ali mi smo to djelomično olakšali ili barem „uljepšali“ u svrhu veće smislenost pri korištenju aplikacije - pozivanjem procedure koja izvršava punjenje te tablice podacima
- Stupac „Posuđeno_Prodano“ unutar tablice „Račun“ nam je iznimno koristan jer je za njega postavljen i okidač koji nam ovisno o unesenoj vrijednosti u taj stupac naslov „prebacuje“ ili u tablicu posuđenih igara ili u tablicu prodanih igara
- Tablica „Račun“ se također kao i tablica „Vraćene_igre“ puni putem za to kreirane procedure
- Tablica „Narudžbenica“ se puni automatski putem za to postavljenog okidača na način da kada stanje nekog proizvoda u nekoj od tri tablice sa „količinom proizvoda unutar poslovnica“ padne na nulu, automatski se stvara narudžbenica za taj proizvod
- U tablicama „Zaposlenik“ i „Poslovođa“ namjerno je pod stupcima „Plaća“ postavljen tip podatka za unos kao „varchar“ a ne kao „integer“, razlog je taj što smo htjeli napisati ne samo iznos kao brojku, već uz to i valutu – npr. „5250kn“
- U tablicama „Zaposlenik“ i „Poslovođa“ postavljeni su „Check constraints“ pod stupcima „Plaća“, gdje baza provjerava i blokira bilo koji unos ispod „3500kn“ u tablici „Zaposlenik“, odnosno „5500kn“ u tablici „Poslovođa“
- Tri tablice sa stanjima poslovnica su punjene „privremenim okidačima“, odnosno Triggerima koje smo napravili samo kako bi jednostavno prekopirali naslove iz tablice „Proizvod“ u te tri tablice, tj. kako ne bi ručno morali prepisivati naslove u svaku od tih tablica. Prilikom kopiranja kopiran je samo ID proizvoda i njegov naziv u za to predviđenu tablicu stanja (ovisno za koju je platformu dostupan), sve količine i dostupnosti odnosno ostali stupci unutar tablica su kasnije ručno punjeni (UPDATE i INSERT INTO naredbama). Trigger je napravljen isključivo s tim ciljem (kopiranja naziva i ID-a), te je kasnije uklonjen jer nam je po popunjavanju te tri tablice postao pomalo problematičan (ponajviše zbog NULL constrainta)

Iako je još u uvodu rečeno da smo se trudili što je više moguće pridržavati se ACID pravila i pravila normalizacije, to smo ipak u određenim tablicama **svjesno** prekršili.

Pravila normalizacije su tako prekršena u 5 od 15 tablica, odnosno u 5 od 15 tablica nemamo niti UNIQUE constraint na nekom stupcu (ključ kandidat), a niti primarni ključ.

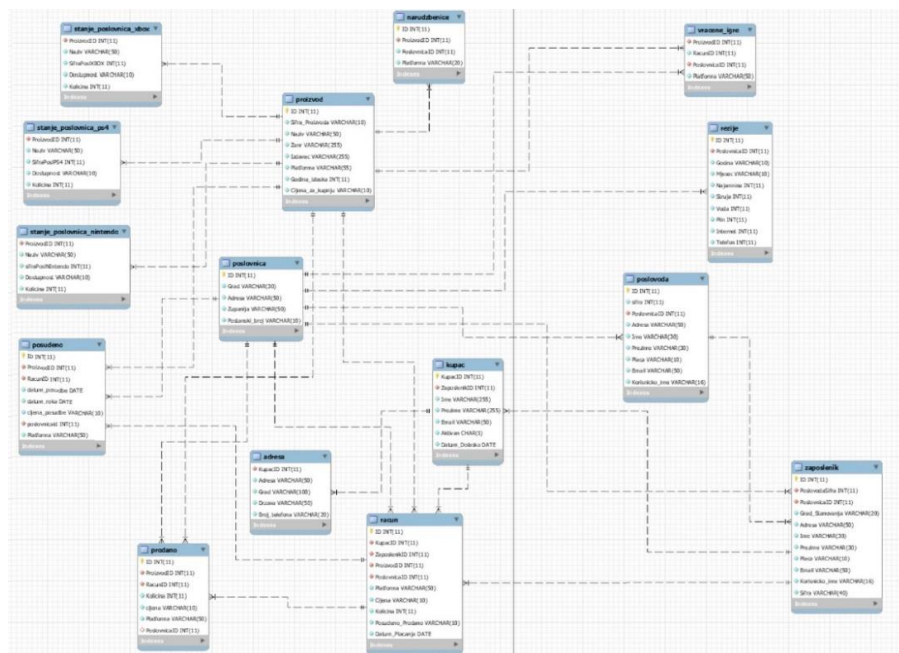
Ipak, smatramo kako smo to uradili iz dobrih i opravdanih razloga, a to su slijedeći:

- U tablicama za stanja poslovnica (Stanje poslovnica Nintendo, Stanje poslovnica PS4, Stanje poslovnica Xbox) smo mogli poštovati pravila normalizacije i stvoriti primarne ključeve na način da za svaku pojedinu poslovnicu stvorimo po tri tablice (za svaku platformu po jedna), ali na taj način bi umjesto 3 tablice dobili ukupno 15 tablica za stanja poslovnica. S obzirom na već poveliku količinu podataka u našoj bazi i veliki broj tablica htjeli smo to izbjeći tj. olakšati si posao koliko je to moguće, pa smo stanja poslovnica podijelili samo prema platformama igara, a stanje proizvoda za određenu platformu svake poslovnice je uneseno u samo jednu prikladnu tablicu za tu platformu. Drugim riječima, kao što možemo vidjeti u sadržaju tih tablica svaki proizvod nam je unesen po pet puta u tablicu stanja gdje je svaki pojedini unos stanje u nekoj od poslovnica (prema šiframa poslovnica gdje je 1 – Zagreb, 2 – Osijek, 3 – Split, 4 – Pula, 5 – Rijeka). Kako nemamo nijednu UNIQUE vrijednost u tim tablicama, tj vrijednost svakog od stupaca unutar tablice se ponavlja više puta, izostavili smo primarni ključ, ali smo pri tome dobro pazili pri osmišljavanju i punjenju tih tablica da kasnije ne bi imali problema u rukovanju sa podacima zbog toga.
- Tablica „Adresa“ nam je svojevrsna „podtablica“ tablice „Kupac“, kao takva ona je povezana sa tablicom „Kupac“ stranim ključem „KupaclD“ (stupac koji je primarni ključ u tablici „Kupac“). Taj stupac nam je mogao eventualno poslužiti i kao „Primary Key“ unutar tablice adresa jer je UNIQUE, tj. ne ponavlja se vrijednost „KupaclD“ unutar tablice „Adresa“, ali pošto to nije baš „pravilna“ i česta praksa u bazama podataka, odlučili smo se da u ovoj tablici izostavimo normalizaciju, odnosno dodijeljivanje primarnoga ključa svjesni da zbog toga nećemo imati nikakvih kasnijih problema niti posljedica u slučaju ove tablice.
- Tablica „Vraćene igre“ je tablica koja se popunjava putem procedure kao što smo prije naveli, te je vrlo jednostavna tablica od samo 4 stupca pri čemu nijedan od tih stupaca nema UNIQUE svojstvo i kao takav ne može biti primarni ključ. U ovoj tablici nam nije ni cilj da imamo ikakvu UNIQUE vrijednost pošto se sav sadržaj ovih tablica može više puta ponavljati, a pritom svaki proces vraćanja proizvoda možemo najbolje razlikovati pogledom na „RačunID“ stupac, te nam stoga nekakva jedinstvenost i svojstvo normalizacije i nije toliko potrebno, odnosno ne narušava nam funkcionalnost te tablice, niti nam je stvaralo ikakve probleme

3.3. ER Dijagram

ER (Entity Relationship) dijagram je slikovni model entiteta, veza i atributa unutar neke baze podataka. On je izrazito koristan prilikom kreiranja samih tablica i veza među tablicama za programere baza podataka kako bi što lakše mogli uočiti određene nedostatke u tablicama, ili eventualne probleme koji bi mogli nastati zbog loših veza među istima.

U konačnici naš ER dijagram prikazuje svih 15 tablica unutar naše baze podataka, te njihov međusobni odnos kao i attribute svake od tablica (Slika 22).



Slika 22 – ER dijagram baze podataka “GameShop”

S obzirom na ograničenje straničnog prostora unutar ovog Word dokumenta, odnosno sitnoga prikaza našega dijagrama, isti je priložen kao slika unutar cjelokupne dokumentacije.

Na našem ER dijagramu simbol ključa unutar tablica označava primarni ključ te tablice, crveni rombovi označavaju strane ključeve unutar tablica, dok su plavi rombovi stupci kreirani unutar tih tablica.

Ono što treba naglasiti, a nije vidljivo na ER dijagramu je da svi stupci u svim tablicama imaju postavljen “NOT NULL” Constraint, a za stupce “Plaća” unutar tablica “Zaposlenik” i “Poslovođa” je postavljen i Check Constraint kako je već prije navedeno, dok je u tablici “Proizvod” za stupac “Naziv” postavljen “UNIQUE” Constraint kako ne bi imali duple unose naziva proizvoda.

4. Pogledi

Za potrebe naše baze kreirali smo i 10 raznovrsnih View-ova, odnosno pogleda.

Kreirani pogledi osmišljeni su za razne ciljeve i potrebe. Neke smo kreirali kako bi postavili dodatnu razinu sigurnosti pri pristupanju određenim podacima unutar baze (npr. plaće zaposlenika i njihova korisnička imena te lozinke), a neke iz potrebe za što jednostavnijim pristupom određenim podacima iz baze (bez potrebe za traženjem podataka putem SELECT naredbe svaki put).

Lista i sadržaj naših pogleda je sljedeći:

- **Lista kupaca** – pogled koji nam javlja trenutnu listu kupaca sa svim bitnim podacima kao što su ime i prezime, adresa, grad, poštanski broj, broj telefona, državu, broj njegovog posljednjeg računa, te ime i prezime i ID zaposlenika koji ga je uslužio
- **Lista zaposlenika** – pogled koji nam govori najbitnije podatke o našim zaposlenicima kao što su ime i prezime, adresa, grad, te u kojoj poslovnici radi i koja je šifra njegovog poslovođa
- **Plaće zaposlenika** – jednostavan pogled koji nam vraća imena i prezimena svih zaposlenika te iznose njihovih plaća
- **Plaće poslovođa** – funkcionira identično kao i pogled „Plaće zaposlenika“, samo što nam vraća imena i prezimena poslovođa i iznose njihovih plaća
- **Korisničko ime i lozinka zaposlenici** – pogled koji nam izlistava imena i prezimena svih zaposlenika, te njihova korisnička imena i lozinke za pristup blagajni/sustavu
- **Korisničko ime i lozinka poslovođe** – pogled koji nam izlistava imena i prezimena svih poslovođa, te njihova korisnička imena i šifre za pristup blagajni/sustavu
- **Posuđeno po poslovnicama** – pogled koji nam vraća vrijednosti ID-a svih poslovnica, gradova u kojima se te poslovnice nalaze, ID-eve poslovođa tih poslovnica i njihova imena i prezimena, te naposljetku ukupan zbroj posuđenih proizvoda u svakoj od tih poslovnica
- **Prodaja po poslovnicama** - pogled koji nam vraća vrijednosti ID-a svih poslovnica, gradova u kojima se te poslovnice nalaze, ID-eve poslovođa tih poslovnica i njihova imena i prezimena, te naposljetku ukupan zbroj (njihov ukupni iznos u kunama) svih prodanih proizvoda u svakoj od poslovnica

- **Lista proizvoda kupnja** – izlistava sve proizvode koji su dostupni za kupovinu unutar poslovnica, odnosno one koji imaju postavljenu cijenu za kupnju
- **Lista proizvoda posudba** – prikazuje sve proizvode koji su dostupni za posudbu unutar poslovnica (lista proizvoda)

Kodovi koji su korišteni pri kreiranju svih ovih pogleda priloženi su u sklopu dokumentacije projekta kao .txt file. Ipak, kako bi prikazali način kreiranja određenog pogleda na donjoj slici kao primjer imamo kod koji je korišten pri kreiranju pogleda “Lista proizvoda kupnja” (Slika 23).

```
CREATE VIEW Lista_Proizvoda_Kupnja
AS
SELECT
    Proizvod.ID AS ProizvodID,
    Proizvod.Naziv AS Naziv,
    Proizvod.Zanr AS Zanr,
    Proizvod.Izdavac AS Izdavac,
    Proizvod.Platforma AS Platforma,
    Proizvod.Cijena_Za_Kupnju AS Cijena
FROM proizvod
WHERE proizvod.cijena_za_kupnju NOT LIKE '%x%';
```

Slika 23 – Kod za kreiranje pogleda “Lista_Proizvoda_Kupnja”

Samim sadržajem (prikazom) pogleda smo se više bavili prilikom postavljanja i testiranja aplikacije, te će tamo to biti i detaljnije prikazano.

Bitno je naglasiti da smo pristup određenim pogledima dodatno osigurali kroz aplikaciju, odnosno ukoliko želimo pristupiti pogledima koji sadrže listu plaća ili listu imena i lozinki od zaposlenika, aplikacija će nas prvo pitati da unesemo našu lozinku kako bi mogli vidjeti te poglede (više o tome u poglavlju o korištenju aplikacije).

5. Okidači

Za potrebe lakšeg popunjavanja naše baze podacima i “automatiziranja” unosa kreirali smo šest Triggera, odnosno okidača.

Osnovni princip rada okidača je da automatski pokreću svoju funkciju po izvršavanju neke postavljene radnje (unosa) u bazu. Tako smo i mi postavili naše okidače da nam pomognu pri automatskom ažuriranju stanja određenih tablica nakon određenih događaja tj. promjena u bazi.

Lista i funkcija naših okidača je slijedeća:

- **Narudžbenica Nintendo** – ukoliko stanje nekog proizvoda u tablici stanja proizvoda za Nintendo Switch konzolu padne na nulu, odnosno ukoliko ostanemo bez nekog proizvoda na skladištu, automatski se kreira narudžbenica za taj proizvod sa ID-em proizvoda, ID-em poslovnice u kojoj je ponestalo proizvoda i platformom za koju nam je narudžba potrebna, te se ta vrijednost automatski pohranjuje u tablicu „Narudžbenice“
- **Narudžbenica PS4** - ukoliko stanje nekog proizvoda u tablici stanja proizvoda za Playstation 4 konzolu padne na nulu, odnosno ukoliko ostanemo bez nekog proizvoda na skladištu, automatski se kreira narudžbenica za taj proizvod sa ID-em proizvoda, ID-em poslovnice u kojoj je ponestalo proizvoda i platformom za koju nam je narudžba potrebna, te se ta vrijednost automatski pohranjuje u tablicu „Narudžbenice“
- **Narudžbenica Xbox** - ukoliko stanje nekog proizvoda u tablici stanja proizvoda za Xbox One konzolu padne na nulu, odnosno ukoliko ostanemo bez nekog proizvoda na skladištu, automatski se kreira narudžbenica za taj proizvod sa ID-em proizvoda, ID-em poslovnice u kojoj je ponestalo proizvoda i platformom za koju nam je narudžba potrebna, te se ta vrijednost automatski pohranjuje u tablicu „Narudžbenice“
- **Posuđeno_Prodano** – ukoliko je u tablici „Račun“ pod stupcem „Posuđeno_Prodano“ unesena vrijednost „Posuđeno“ – proizvod se automatski sa svojim ID-em, brojem računa, cijenom posudbe i datumom plaćanja dodaje u tablicu „Posuđeno“, a ukoliko je unos u tom stupcu „Prodano“ taj proizvod se sa sličnim vrijednostima dodaje u tablicu „Prodano“
- **Update_Stanja_Prodano** – ukoliko se neki proizvod proda u nekoj od poslovnica stanje tog proizvoda se automatski smanjuje za 1 u za to prikladnoj tablici stanja poslovnica (ovisno je li prodani proizvod bio za PS 4, Nintendo ili Xbox)

- **Update_Stanja_Vraćeno** – ukoliko se neki posuđeni proizvod vrati u poslovnici stanje tog proizvoda se povećava za 1 u za to prikladnoj tablici stanja poslovnica naravno i u za to prikladnoj poslovnici

Svi kodovi za napravljene okidače su priloženi kao .txt file unutar dokumentacije projekta, ipak kako bi prikazali način kreiranja određenog pogleda uzeli smo za primjer kod “Posuđeno_Prodano” okidača koji možemo vidjeti na donjoj slici (Slika 24).

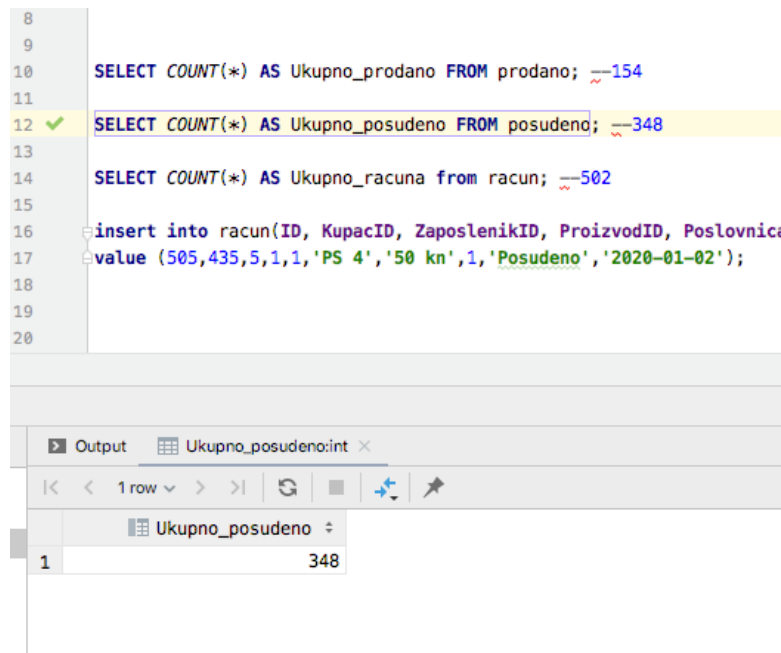
```
USE `gameshop`
CREATE
TRIGGER `gameshop`.`Posuđeno_Prodano`
AFTER INSERT ON `gameshop`.`racun`
FOR EACH ROW
BEGIN
    IF (new.Posuđeno_Prodano = 'Posuđeno') THEN
        insert into Posuđeno(ProizvodID, RacunID, Datum_Posudbe, Datum_roka, Cijena_Posudbe, Platforma, PoslovnicaID)
        values (new.ProizvodID, new.ID, new.datum_placanja, (DATE_ADD(new.datum_placanja, INTERVAL 1 MONTH)), '25kn', new.Platforma, new.PoslovnicaID);
    ELSE
        insert into Prodano(ProizvodID, RacunID, Kolicina, Cijena, Platforma, PoslovnicaID)
        values(new.ProizvodID, new.ID, new.Kolicina, (select Cijena_Za_Kupnju from Proizvod where new.ProizvodID=ID), new.Platforma, new.PoslovnicaID);
    END IF;
END
```

Slika 24 – Kod za kreiranje okidača “Posuđeno_Prodano”

S obzirom da se okidači sami izvršavaju nismo radili nikakva posebna dodavanja opcija u aplikaciji za njih, ali smo izvršili testiranje unutar SQL Developera nad okidačem “Posuđeno_Prodano” kako bi prikazali kako taj okidač u stvarnosti funkcionira.

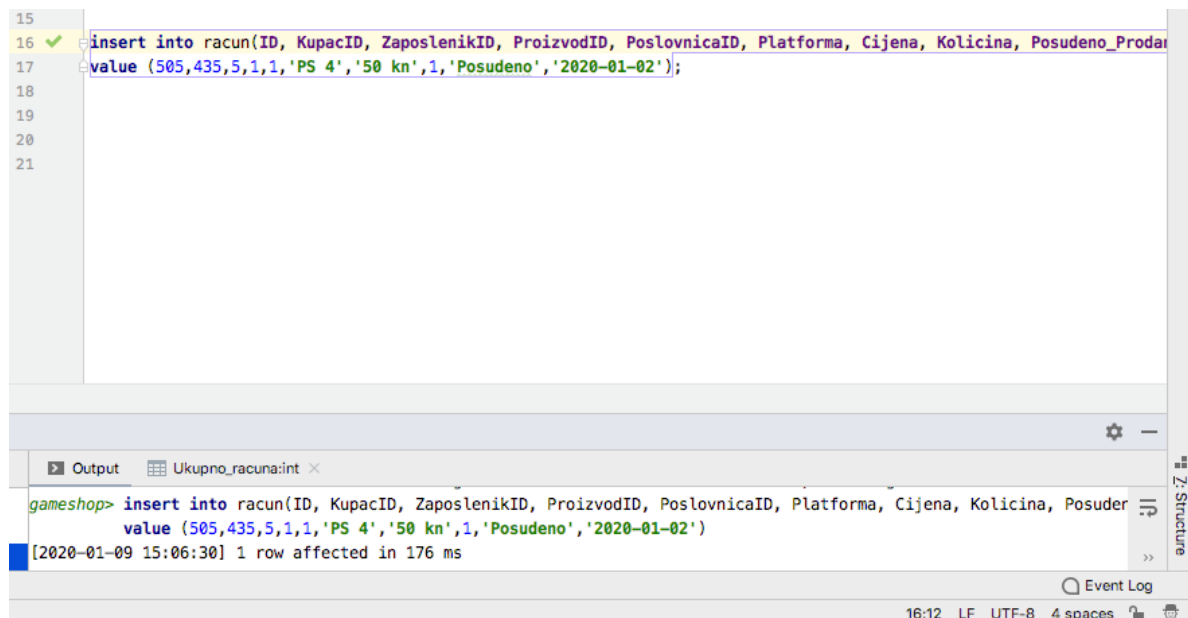
Tako smo se odlučili da ćemo u tablicu “Račun” dodati račun za određeni posuđeni proizvod, te smo provjerili broj posuđenih proizvoda u tablici “Posuđeno” prije unosa navedenog proizvoda, a potom i nakon “posuđivanja” tog nekog proizvoda kako bi dokazali da se stanje proizvoda u tablici “Posuđeno” povećalo za 1.

Na donjoj slici vidimo da je broj posuđenih proizvoda prije umetanja nove posudbe 348 (Slika 25).



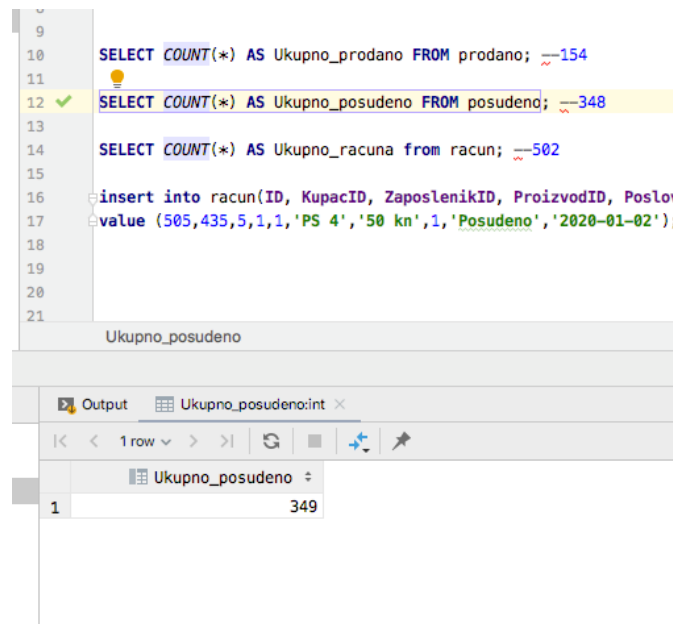
Slika 25 – Stanje broja posuđenih proizvoda prije nove posudbe

Potom smo unijeli sljedeći račun u tablicu “Račun” gdje je uneseno da je traženi proizvod “Posuđen” (Slika 26).



Slika 26 – Unošenje novog računa sa posuđenim proizvodom

Nakon unošenja novog računa ponovo provjeravamo stanje broja posuđenih proizvoda u tablici “Posuđeno”, te vidimo da se on uvećao za 1, odnosno broj proizvoda se povećao sa 348 na 349 (Slika 27).



```
9
10 SELECT COUNT(*) AS Ukupno_prodano FROM prodano; --154
11
12 ✓ SELECT COUNT(*) AS Ukupno_posudeno FROM posudeno; --348
13
14 SELECT COUNT(*) AS Ukupno_racuna from racun; --502
15
16 insert into racun(ID, KupacID, ZaposlenikID, ProizvodID, Poslovi
17 value (505,435,5,1,1,'PS 4','50 kn',1,'Posudeno','2020-01-02');
18
19
20
21
```

Ukupno_posudeno

| Ukupno_posudeno:int | |
|---------------------|-----|
| 1 | 349 |

Slika 27 – Stanje broja posuđenih proizvoda nakon izvršene posudbe

Primjere za ostale okidače nismo uzimali iz razloga što svi funkcioniraju na istom ili vrlo sličnom principu, a i iz njihovih dodijeljenih tablica možemo uvijek jasno vidjeti kako oni uspješno rade.

6. Procedure

Pohranjene procedure su objekti unutar SQL programskog jezika koji nam služe da nam vrate (ne nužno) određenu vrijednost iz neke tablice za koju su definirane. Razlikuju se od pogleda po tome što za izvršavanje procedura najčešće moramo unijeti neke vrijednosti kako bi dobili željeni rezultat. Služe često za olakšavanje unosa podataka u određenu tablicu.

Pri kreiranju naše baze definirali smo 12 procedura sa sljedećim funkcionalnostima:

- **Igre koje nisu vraćene** – pozivom ove procedure dobivamo sve igre iz tablice „Posuđeno“ koje se ne nalaze u tablici „Vraćene igre“, odnosno koje još nisu vraćene u poslovnici
- **Mjesečne režije** – pozivom ove procedure i unosom ID -a poslovnice, mjeseca i godine dobijemo ukupan zbroj režija za taj mjesec za traženu poslovnici
- **Poslovođa_Podređeni** – unošenjem ID-a poslovođe dobijemo broj ljudi kojima je on direktni šef
- **Prodane igre u mjesecu** – unošenjem mjeseca dobijemo listu igara koje su prodane u tom mjesecu
- **Promet zaposlenika** – procedura koja nam daje izvješće o ukupnom prometu (iznosu) koji je određeni zaposlenik priskrbio za poslovnici
- **Provjera stanja** – procedura koja unosom ID-a, poslovnice i željene platforme provjerava kolika je količina tog proizvoda u skladištu
- **Ubacivanje kupca** – jednostavno ubacivanje kupaca u bazu (umjesto korištenja INSERT INTO)
- **Ubacivanje računa** – jednostavno ubacivanje računa u bazu
- **Vraćanje igre** – procedura čijim pozivom na jednostavan način unosimo igru koja se vraća u bazu
- **Zaliha proizvoda** – pozivom ove procedure i unošenjem željene tablice stanja tj. odabirom platforme dobivamo listu svih proizvoda kojima je stanje na skladištu manje od 1, odnosno svih proizvoda koji su nedostupni

- **Zaposlenik_informacije** – procedura koja nam po unosu ID-a zaposlenika vraća sve ostale važne informacije o njemu
- **Zaposlenik_Rent** – vraća nam koliko je ukupno proizvoda određeni zaposlenik iznajmio

Svi kodovi korišteni za izradu gore navedenih procedura su priloženi u dokumentaciji. Za primjer izgleda koda unutar ovog dokumenta uzeli smo kod prvonavedene procedure “Igre koje nisu vraćene” (Slika 28).

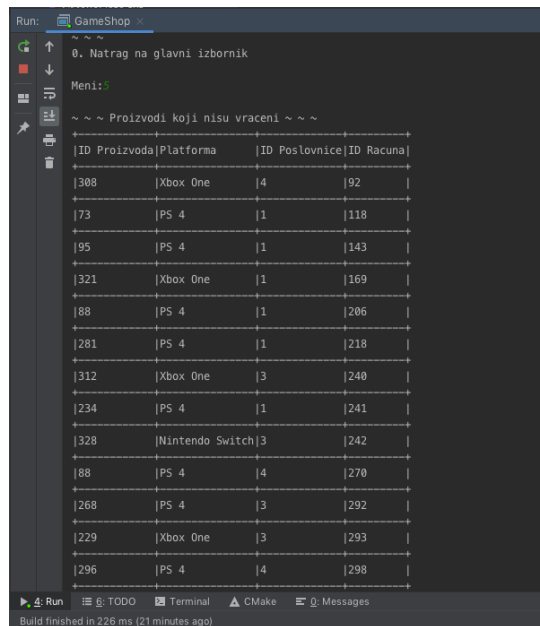
```
CREATE PROCEDURE Igre_Koje_Nisu_Vracene (
    OUT ProizvodID int,
    OUT Platforma varchar(50),
    OUT PoslovnicaID int,
    OUT RacunID int)
BEGIN
    SELECT posudeno.proizvodid, posudeno.platforma, posudeno.poslovnicaid, posudeno.racunid
    FROM posudeno
    NATURAL LEFT JOIN vracene_igre
    WHERE vracene_igre.racunid IS NULL;
END
```

Slika 28 – Primjer koda za izradu procedure “Igre koje nisu vraćene”

Za pozive procedura unutar SQL-a koristimo naredbu **CALL ime_procedure**.

Procedure nismo pretjerano testirali unutar SQL Developera, već kasnije unutar naše aplikacije. Unutar aplikacije procedure su ubačene, odnosno isprogramirane u C++ kod, te kao takve se jednostavno pozivaju odabirom prikladne procedure na izborniku našeg programa (više o tome u kasnijim poglavljima).

Primjer rezultata ispisa naše procedure “Igre koje nisu vraćene” iz aplikacije možemo vidjeti na donjoj slici (Slika 29).



The screenshot shows a C++ application window titled "GameShop". The main menu displays "0. Natrag na glavni izbornik" and "Meni:". Below the menu, there is a section titled "~~~ Proizvodi koji nisu vraćeni ~~~". This section contains a table with four columns: "ID Proizvoda", "Platforma", "ID Poslovnice", and "ID Racuna". The table lists 15 products with their respective IDs, platforms, store IDs, and account IDs.

| ID Proizvoda | Platforma | ID Poslovnice | ID Racuna |
|--------------|-----------------|---------------|-----------|
| 308 | Xbox One | 4 | 92 |
| 73 | PS 4 | 1 | 118 |
| 95 | PS 4 | 1 | 143 |
| 321 | Xbox One | 1 | 169 |
| 88 | PS 4 | 1 | 206 |
| 281 | PS 4 | 1 | 218 |
| 312 | Xbox One | 3 | 240 |
| 234 | PS 4 | 1 | 241 |
| 328 | Nintendo Switch | 3 | 242 |
| 88 | PS 4 | 4 | 270 |
| 268 | PS 4 | 3 | 292 |
| 229 | Xbox One | 3 | 293 |
| 296 | PS 4 | 4 | 298 |

Slika 29 – Rezultat ispisa nakon poziva procedure “Igre koje nisu vraćene”

7. Funkcije

Funkcije unutar SQL-a su po strukturi dosta slične procedurama, ali ipak se razlikuju u nekim pravilima. Glavne razlike su te da funkcije moraju obavezno uvijek vratiti vrijednost, te da se pomoću njih ne mogu mijenjati podaci u tablicama.

Za potrebe naše baze kreirali smo 11 funkcija sa sljedećim zadaćama:

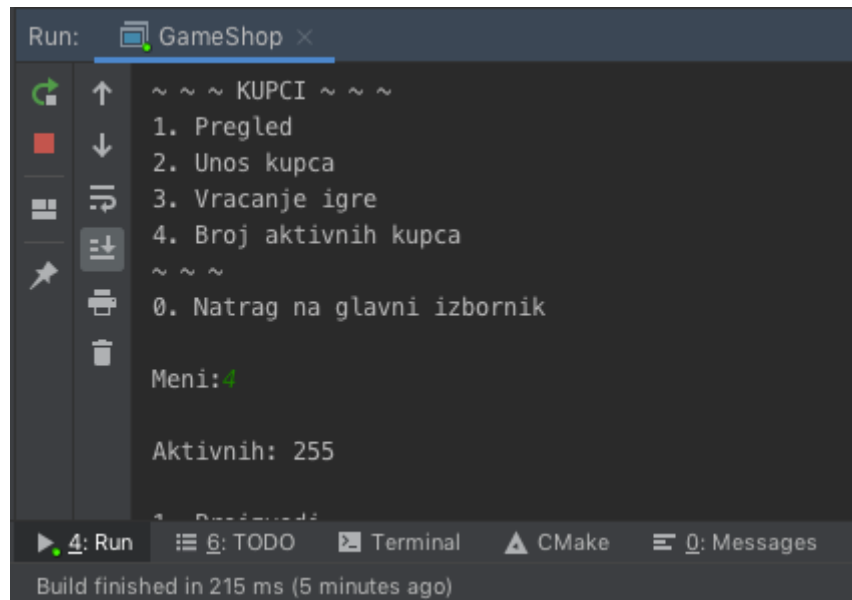
- **Broj aktivnih kupaca** – vraća nam ukupni broj kupaca koji su aktivni
- **Broj dostupnih proizvoda Nintendo** – vraća nam ukupni broj dostupnih proizvoda iz tablice „Stanje poslovnica Nintendo“
- **Broj dostupnih proizvoda PS4** – vraća nam ukupni broj dostupnih proizvoda iz tablice „Stanje poslovnica PS4“
- **Broj dostupnih proizvoda Xbox** – vraća nam ukupni broj dostupnih proizvoda iz tablice „Stanje poslovnica Xbox“
- **Broj proizvoda Nintendo** – vraća ukupnu količinu svih dostupnih proizvoda za Nintendo Switch konzolu (zbraja količine svih proizvoda iz svih poslovnica)
- **Broj proizvoda PS4** - vraća ukupnu količinu svih dostupnih proizvoda za Playstation 4 konzolu
- **Broj proizvoda Xbox** – vraća ukupnu količinu svih dostupnih proizvoda za Xbox One konzolu
- **Broj višaka proizvoda Nintendo** – vraća koliki je broj proizvoda u tablici Stanje_Poslovnica_Nintendo gdje je količina pojedinog proizvoda u nekoj poslovnici veća od 10
- **Broj višaka proizvoda PS4** – identična uloga kao i kod prethodne funkcije za višak proizvoda, samo što je ovdje u pitanju Playstation 4 konzola
- **Broj višaka proizvoda Xbox** – identično kao i za višak proizvoda kod PS4 i Nintendo, samo je u ovom slučaju funkcija napravljena za Xbox One konzolu
- **Ukupan promet** – funkcija nam vraća ukupnu zaradu od svih prodanih proizvoda u tablici „Prodano“

Svi kodovi za kreiranje funkcija su priloženi u sklopu dokumentacije. Kao primjer koda kojeg ćemo prikazati u ovom dokumentu uzeli smo kod funkcije “Broj aktivnih kupaca” koji možemo vidjeti na donjoj slici (Slika 30).

```
CREATE FUNCTION broj_aktivnih_kupaca(  
    aktivan CHAR  
) RETURNS INTEGER DETERMINISTIC  
begin  
    declare k_zbroj INTEGER;  
    SELECT count(*) INTO k_zbroj FROM kupac AS k  
    WHERE k.aktivan = 'D';  
    RETURN k_zbroj;  
END
```

Slika 30 – Kod za kreiranje funkcije “Broj aktivnih kupaca”

Funkcije smo ponajviše isprobavali kroz sučelje aplikacije, odnosno tamo smo ih unijeli kao izbore na podizbornicima prikladnih tablica, pa tako možemo vidjeti prikaz izvršavanja funkcije za “Broj aktivnih kupaca” na sljedećoj slici (Slika 31).



Slika 31 – Primjer uspješnog izvršavanja funkcije “Broj aktivnih kupaca” unutar aplikacije

8. Transakcije

Transakcija je specifičan način izvršavanja slijeda naredbi unutar SQL programskog jezika. Transakcije se izvode neovisno od ostalog koda, te njihovo potpuno izvršavanje uvijek zahtijeva potvrdu od strane korisnika u vidu “Commit” naredbe na kraju transakcije. Naravno transakcije (većinom) imaju i “Rollback” opciju kojom se vraćamo na stanje baze prije izvršavanja određene naredbe u transakciji. Te osobitosti ih čine vrlo bitnim i korisnim načinima unosa podataka u SQL baze, pogotovo u svrhu očuvanja ACID pravila.

S obzirom da SQL Developer ne sprema transakcije trajno u svoju memoriju kao što čini za sve ostale prethodno obrađene stavke (pogled, procedure, okidače i funkcije), sa transakcijama se nismo pretjerano bavili prilikom kreiranja i testiranja naše baze.

Za potrebe naše baze isprobali smo funkcioniranje sljedeće dvije transakcije:

- **Transakcija za podizanje cijene posudbe u zagrebačkoj poslovnici na 30kn**
- **Transakcija za podizanje plaće radnicima u riječkoj poslovnici za 500kn**

Na sljedećoj slici (Slika 32) vidimo kod za prvu transakciju za podizanje cijene posudbe.

```
START TRANSACTION ;  
UPDATE posudeno SET cijena_posudbe = '30kn' WHERE poslovnicaid = 1;  
COMMIT;
```

Slika 32 – Kod transakcije za podizanje cijene posudbe

Prije izvršavanja transakcije vidimo da je cijena posudbe u zagrebačkoj poslovnici bila 25kn (Slika 33).

| ID | ProizvodID | RacunID | datum_posudbe | datum_roka | cijena_posudbe | poslovni... |
|----|------------|---------|---------------|------------|----------------|-------------|
| 1 | 2 | 48 | 2019-02-27 | 2019-03-27 | 25kn | |
| 2 | 3 | 241 | 2019-03-13 | 2019-04-13 | 25kn | |
| 3 | 5 | 128 | 2019-04-02 | 2019-05-02 | 25kn | |
| 4 | 10 | 19 | 2019-05-31 | 2019-06-30 | 25kn | |
| 5 | 12 | 310 | 2019-07-16 | 2019-08-16 | 25kn | |
| 6 | 24 | 194 | 2019-03-14 | 2019-04-14 | 25kn | |
| 7 | 26 | 94 | 2019-04-25 | 2019-05-25 | 25kn | |
| 8 | 31 | 27 | 2019-07-12 | 2019-08-12 | 25kn | |
| 9 | 32 | 89 | 2019-08-05 | 2019-09-05 | 25kn | |
| 10 | 36 | 247 | 2019-07-01 | 2019-08-01 | 25kn | |
| 11 | 42 | 261 | 2019-09-28 | 2019-10-28 | 25kn | |
| 12 | 44 | 184 | 2019-12-05 | 2020-01-05 | 25kn | |
| 13 | 48 | 341 | 2019-03-04 | 2019-04-04 | 25kn | |
| 14 | 49 | 215 | 2019-03-11 | 2019-04-11 | 25kn | |
| 15 | 53 | 33 | 2019-05-25 | 2019-06-25 | 25kn | |
| 16 | 55 | 98 | 2019-07-23 | 2019-08-23 | 25kn | |
| 17 | 56 | 11 | 2019-08-10 | 2019-09-10 | 25kn | |
| 18 | 58 | 334 | 2019-09-09 | 2019-10-09 | 25kn | |
| 19 | 62 | 227 | 2019-01-20 | 2019-02-20 | 25kn | |
| 20 | 64 | 338 | 2019-03-19 | 2019-04-19 | 25kn | |
| 21 | 67 | 191 | 2019-07-23 | 2019-08-23 | 25kn | |
| 22 | 70 | 58 | 2019-09-23 | 2019-10-23 | 25kn | |
| 23 | 72 | 76 | 2019-10-07 | 2019-11-07 | 25kn | |
| 24 | 74 | 73 | 2019-11-03 | 2019-12-03 | 25kn | |
| 25 | 77 | 89 | 2019-02-03 | 2019-03-03 | 25kn | |
| 26 | 82 | 32 | 2019-08-07 | 2019-09-07 | 25kn | |
| 27 | 83 | 253 | 2019-08-10 | 2019-09-10 | 25kn | |

Slika 33 – Cijena posudbe prije izvršenja transakcije

Nakon izvršavanja koda transakcije vidimo da je cijena posudbe u poslovnici skočila sa 25 na 30kn (Slika 34).

| ID | ProizvodID | RacunID | datum_posudbe | datum_roka | cijena_posudbe | poslovni... |
|----|------------|---------|---------------|------------|----------------|-------------|
| 1 | 2 | 48 | 2019-02-27 | 2019-03-27 | 30kn | |
| 2 | 3 | 241 | 2019-03-13 | 2019-04-13 | 30kn | |
| 3 | 5 | 128 | 2019-04-02 | 2019-05-02 | 30kn | |
| 4 | 10 | 19 | 2019-05-31 | 2019-06-30 | 30kn | |
| 5 | 12 | 310 | 2019-07-16 | 2019-08-16 | 30kn | |
| 6 | 24 | 194 | 2019-03-14 | 2019-04-14 | 30kn | |
| 7 | 26 | 94 | 2019-04-25 | 2019-05-25 | 30kn | |
| 8 | 31 | 27 | 2019-07-12 | 2019-08-12 | 30kn | |
| 9 | 32 | 89 | 2019-08-05 | 2019-09-05 | 30kn | |
| 10 | 36 | 247 | 2019-07-01 | 2019-08-01 | 30kn | |
| 11 | 42 | 261 | 2019-09-28 | 2019-10-28 | 30kn | |
| 12 | 44 | 184 | 2019-12-05 | 2020-01-05 | 30kn | |
| 13 | 48 | 341 | 2019-03-04 | 2019-04-04 | 30kn | |
| 14 | 49 | 215 | 2019-03-11 | 2019-04-11 | 30kn | |
| 15 | 53 | 33 | 2019-05-25 | 2019-06-25 | 30kn | |
| 16 | 55 | 98 | 2019-07-23 | 2019-08-23 | 30kn | |
| 17 | 56 | 11 | 2019-08-10 | 2019-09-10 | 30kn | |
| 18 | 58 | 334 | 2019-09-09 | 2019-10-09 | 30kn | |
| 19 | 62 | 227 | 2019-01-20 | 2019-02-20 | 30kn | |
| 20 | 64 | 338 | 2019-03-19 | 2019-04-19 | 30kn | |
| 21 | 67 | 191 | 2019-07-23 | 2019-08-23 | 30kn | |
| 22 | 70 | 58 | 2019-09-23 | 2019-10-23 | 30kn | |
| 23 | 72 | 76 | 2019-10-07 | 2019-11-07 | 30kn | |
| 24 | 74 | 73 | 2019-11-03 | 2019-12-03 | 30kn | |
| 25 | 77 | 89 | 2019-02-03 | 2019-03-03 | 30kn | |
| 26 | 82 | 32 | 2019-08-07 | 2019-09-07 | 30kn | |
| 27 | 83 | 253 | 2019-08-10 | 2019-09-10 | 30kn | |

Slika 34 – Promjena cijene posudbe nakon izvršenja transakcije

Kod druge transakcije za podizanje plaće radnicima vidimo u nastavku (Slika 35).

```
START TRANSACTION;  
UPDATE zaposlenik SET Placa = Placa + 500 WHERE PoslovodaSifra = 6812;  
COMMIT;
```

Slika 35 – Kod transakcije za podizanje plaće radnicima u Rijeci

Prije izvršavanja gornje transakcije na sljedećoj slici vidimo koliko su iznosile cijene svakog radnika u Rijeci (Slika 36).

| | GradStanovanja | Adresa | Ime | Prezime | Placa | Email |
|---|----------------|----------------------------|---------|----------|-------|---------------------------|
| 1 | rijeka | Ulica Ivana Mažuranića 65 | Mario | Nikolić | 4586 | mario.nikolic@gmail.com |
| 2 | trikvenica | Ulica Petra Preradovića 89 | Danijel | Grgić | 5018 | danigrbic@hotmail.com |
| 3 | ovran | Ulica Branka Gavella 91 | Patrik | Ivanović | 5586 | patrik.ivanovic@gmail.com |
| 4 | orec | Kastelska_12 | Marko | Kacic | 5520 | mkacic@gmail.com |

Services

| Tx | Service | Duration | Details |
|----|---------------------|----------|---|
| ✓ | gameshop.zaposlenik | 747 ms | WHERE PoslovodaSifra = 6812 LIMIT 501 |
| ✓ | gameshop.zaposlenik | 747 ms | [2020-01-09 16:08:46] 4 rows retrieved starting from 1 in 249 ms (execution: 230 ms, fetching: 19 ms) |
| ✓ | console | 531 ms | |

Slika 36 – Plaće riječkih radnika prije izvršenja transakcije

Nakon izvršenja transakcije vidimo da se plaća svih gore navedenih radnika podigla za 500kn (Slika 37).

The screenshot shows a database management interface with the following components:

- Database Explorer:** Shows the database structure for '@GameShop', including schemas, tables (15), and views (7). The 'zaposlenik' table is highlighted under the 'tables' folder.
- Query Results:** Displays the results of a query: `WHERE PoslovodaSifra = 6812 LIMIT 501`. The results show 4 rows retrieved starting from 1 in 200 ms (execution: 184 ms, fetching: 16 ms).
- Table Data:** The 'zaposlenik' table data is shown with columns: Grad Stanovanja, Adresa, Ime, Prezime, Placa, and Email. The data is as follows:

| Grad Stanovanja | Adresa | Ime | Prezime | Placa | Email | |
|-----------------|-----------|----------------------------|---------|----------|-------|---------------------------|
| 1 | tijeka | Ulica Ivana Mažuranića 65 | Mario | Nikolić | 5086 | mario.nikolic@gmail.com |
| 2 | rikvenica | Ulica Petra Preradovića 89 | Danijel | Grgić | 5518 | danigrgic@hotmail.com |
| 3 | ovran | Ulica Branka Gavella 91 | Patrik | Ivanović | 6086 | patrik.ivanovic@gmail.com |
| 4 | orec | Kastelska_12 | Marko | Kacic | 6020 | mkacic@gmail.com |
- Services:** Shows the execution of the query, with a status of '4 rows retrieved starting from 1 in 200 ms (execution: 184 ms, fetching: 16 ms)'. The 'console' service is also visible.

Slika 37 – Plaće riječkih radnika nakon izvršenja transakcije

9. Aplikacija

9.1. Potrebni programi i postupak za uspješno postavljanje aplikacije

Sa prethodnim poglavljem obradili smo zadnju bitnu stavku što se tiče kreiranja i punjenja same baze podacima. Nakon kompletiranja baze trebalo je još kreirati i funkcionalnu aplikaciju za lakše pristupanje, pregledavanje i upravljanje sa podacima u bazi.

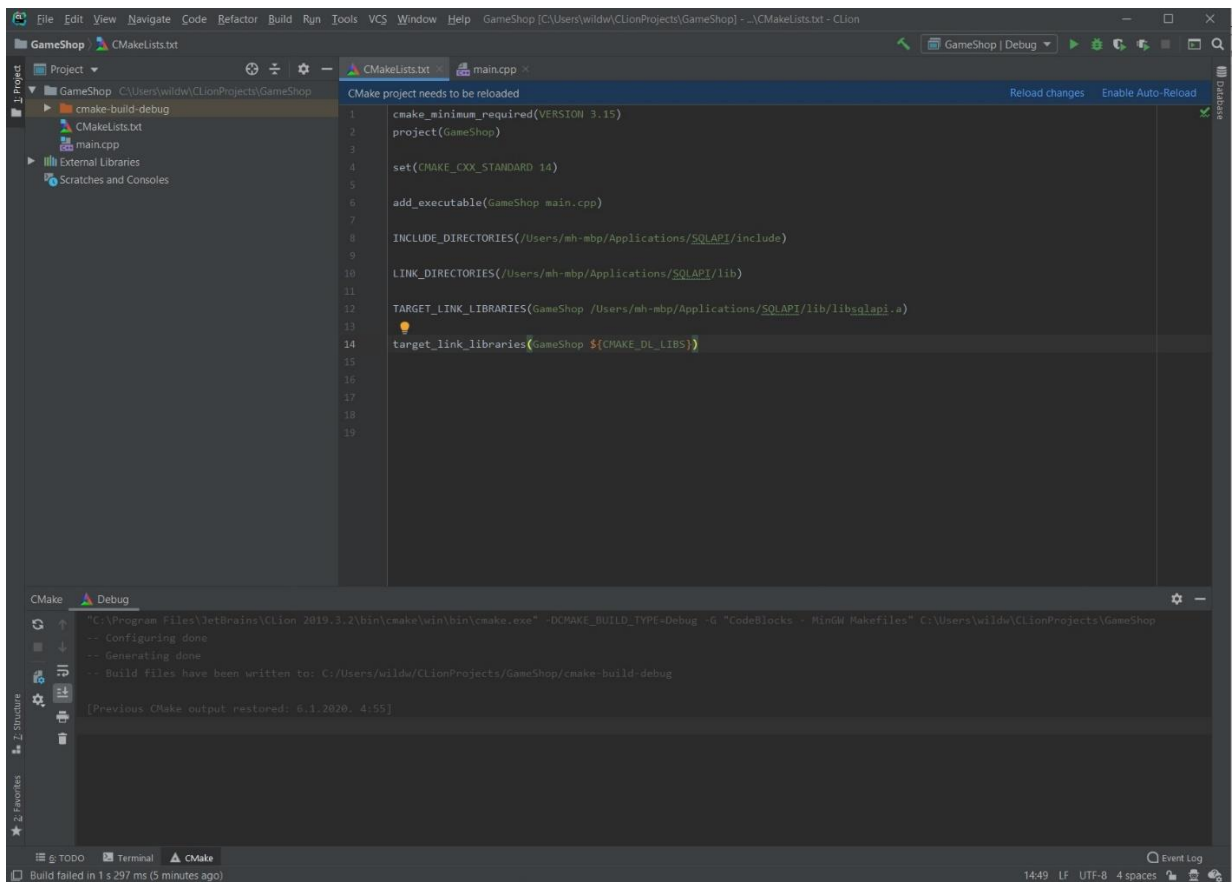
U prethodnom dokumentu „Sažetak projekta“ definirali smo da ćemo pokušati napraviti aplikaciju za povezivanje s bazom u C++ programskom jeziku. Kada je krenuo proces kreiranja baze shvatili smo da C++ možda i nije bio najlakši odabir za kreiranje aplikacije za rad s bazom (iako smo s tim programskim jezikom najviše upoznati). Naime, u razgovorima s kolegama iz ostalih timova shvatili smo da bi možda puno lakše postavili aplikaciju u nekom drugom programskom jeziku (npr. Python). Ipak, nismo odustajali te smo na kraju stvorili savršeno funkcionalnu konzolnu aplikaciju za upravljanje našom bazom unutar C++ okruženja.

Prije početka rada na programskom kodu kolege programeri su se morali dobro informirati na internetu oko toga šta im je potrebno za povezivanje C++ aplikacije sa SQL bazom. Shvatili su da moraju preuzeti još dva dodatna programa s interneta, a to su programi CLion za pisanje programskog koda (jasno tu se mogao koristiti i neki drugi program, ali preko njega je bilo najlakše) i knjižnicu SQLApi++ koja je pisana u svrhu povezivanja C++ aplikacija sa SQL bazama.

Nakon instaliranja svih potrebnih programa mogao je početi proces kreiranja aplikacije. Za postavljanje potrebnog okružja unutar CLiona potrebno je odabrati da želimo kreirati novi „C++ Executable“ projekt, te je ostalo sve dovoljno pustiti na već zadanim postavkama. Projekt nazivamo po želji, nama je to naravno bio „GameShop“.

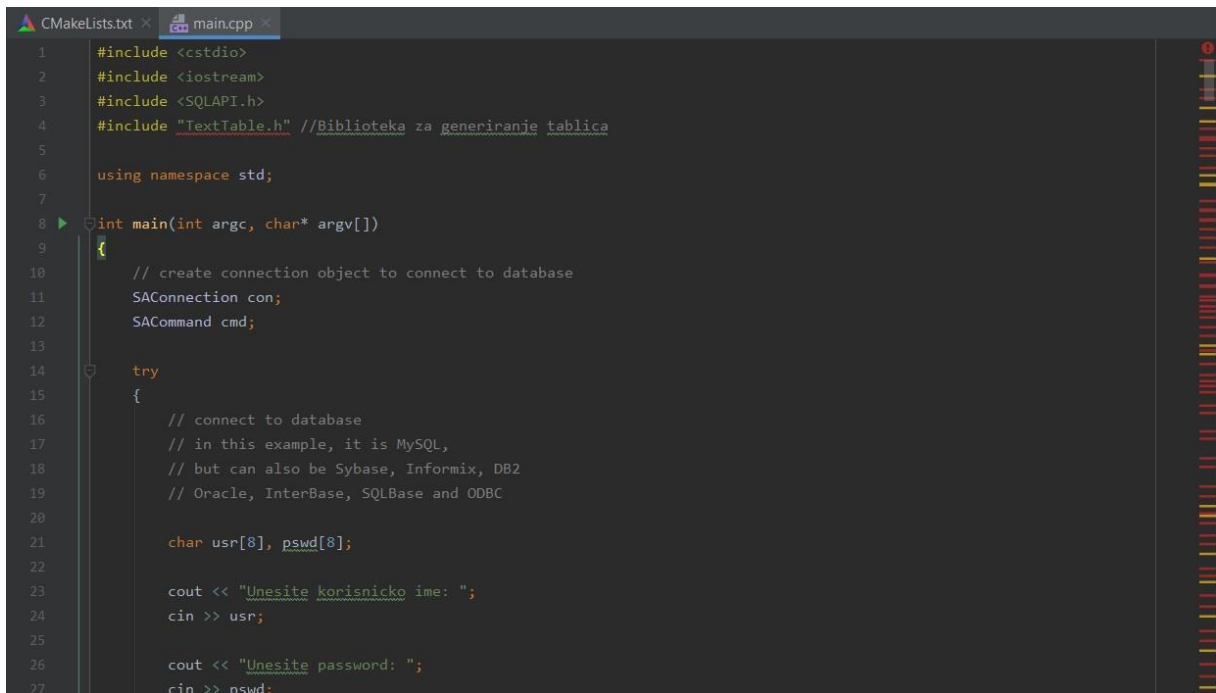
Po kreiranju projekta otvoriti će nam se dva prozora za pisanje koda unutar CLiona, jedan je „CmakeLists“, a drugi je „main.cpp“.

Prvonavedeni prozor „CmakeLists“ (Slika 38) služi za pisanje koda za povezivanje potrebnih datoteka i programa za rad s našim programskim kodom. Drugim riječima tu postavljamo sve „datotečne puteve“ do potrebnih libraryja (SQLApi++) koji nam koriste za kasnije uspješno spajanje programa na neku SQL bazu.



Slika 38 – Izgled Clion sučelja i CmakeLists prozor

U drugom prozoru „main.cpp“ (Slika 39) unosimo kod našeg programa sa svim potrebnim i željenim podacima iz naše baze. Kolege programeri su za pisanje koda prvotno na internetu našli neku vrstu „bazičnog koda“, odnosno univerzalni kod za funkcioniranje SQL-a unutar C++ programa, te su taj kod prilagodili našoj bazi i s vremenom ga nadopunjavali sa potrebnim stavkama naše baze.



```
1 #include <cstdio>
2 #include <iostream>
3 #include <SQLAPI.h>
4 #include "TextTable.h" //Biblioteka za generiranje tablica
5
6 using namespace std;
7
8 int main(int argc, char* argv[])
9 {
10     // create connection object to connect to database
11     SAConnection con;
12     SACommand cmd;
13
14     try
15     {
16         // connect to database
17         // in this example, it is MySQL,
18         // but can also be Sybase, Informix, DB2
19         // Oracle, InterBase, SQLBase and ODBC
20
21         char usr[8], pswd[8];
22
23         cout << "Unesite korisnicko ime: ";
24         cin >> usr;
25
26         cout << "Unesite password: ";
27         cin >> pswd;
```

Slika 39 – Izgled „main.cpp“ prozora unutar Cliona za već gotovu aplikaciju

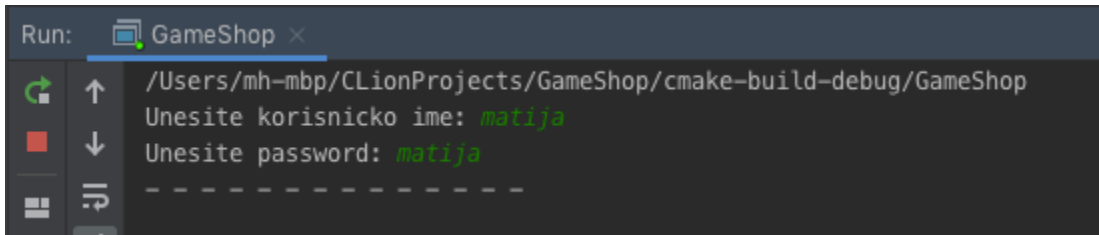
Bitno je naglasiti da je aplikacija pisana na „macOS-u“, odnosno na MacBook Pro računalu kolege Hamera. Kao takva ona je primarno prilagođena za funkcioniranje na „macOS-u“. Kako bi aplikacija radila i na ostalim operacijskim sustavima kao što su Windowsi ili razne Linux distribucije, potrebno je prilagoditi kod u „CmakeLists“ dokumentu. Naime „main.cpp“ je napravljen „univerzalno“ tj. na njemu se ne moraju raditi nikakve promjene ni prilagodbe prilikom promjene OS-a, već sve promjene za rad programa na drugom OS-u treba postaviti u „CmakeLists“ dokumentu.

Ovisno od računala do računala te promjene su negdje manje, a negdje veće, u principu najčešće treba podesiti nanovo „datotečne puteve“ do traženih libraryja, a ponekad i instalirati vanjski compiler program.

Aplikacija je uspješno osposobljena za rad na sve tri vrste OS-a, ali ispitivanja i rad na aplikaciji se ipak najviše odvijao na sustavu na kojem je i kreirana (jer se tu događalo najmanje komplikacija), odnosno na MacBook Pro računalu kolege Hamera.

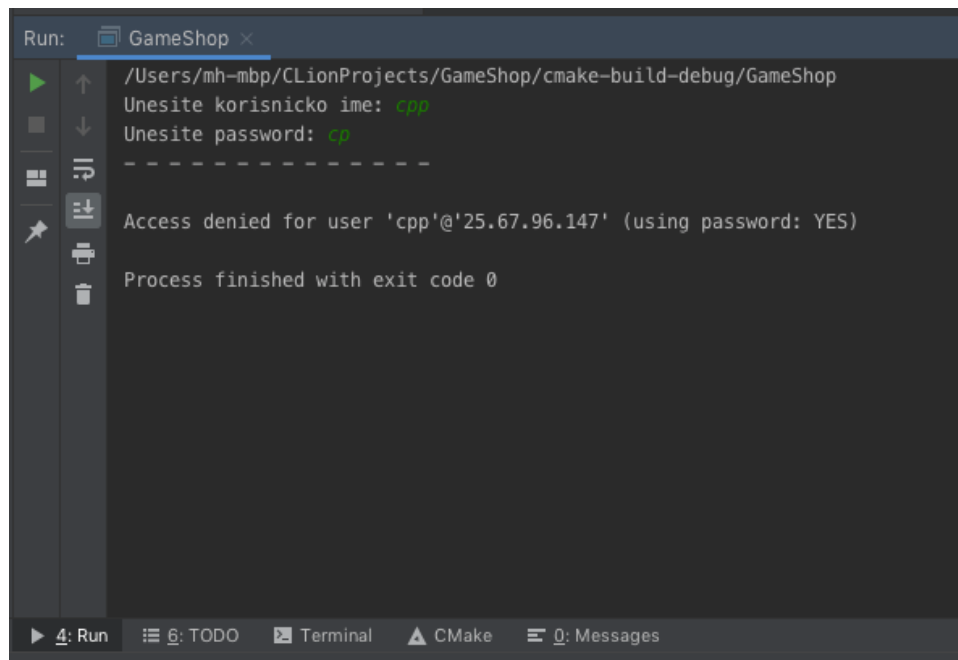
9.2. Izgled aplikacije i testiranje rada

Prilikom uspješnog pokretanja naše aplikacije odmah nam se pojavljuje „login screen“ (Slika 40), odnosno aplikacija nas traži da unesemo naše korisničko ime i lozinku koje koristimo i za pristup bazi preko SQL Developera (koje smo prvotno postavili na MySQL Serveru).



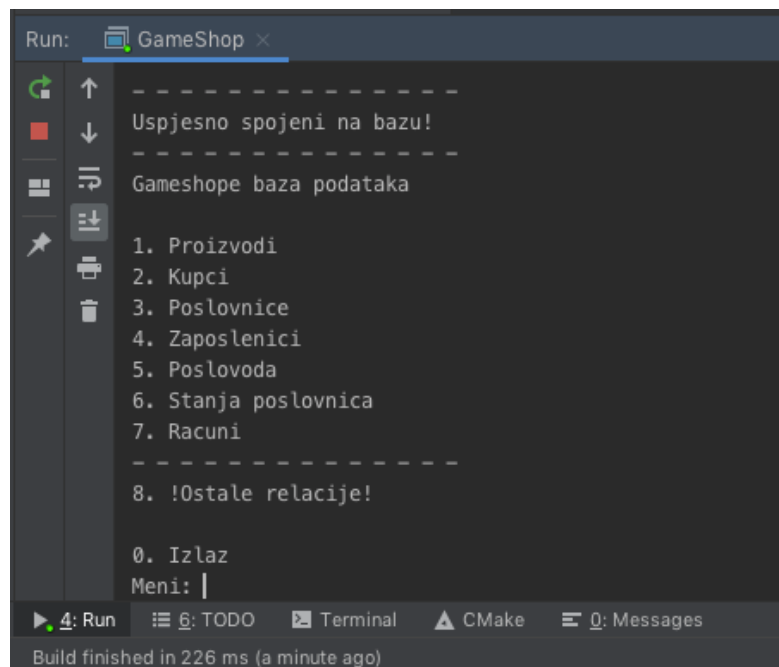
Slika 40 – Login screen unutar aplikacije

Ukoliko unesemo pogrešno korisničko ime ili lozinku aplikacija nam javlja grešku i ne dopušta nam pristup bazi (Slika 41).



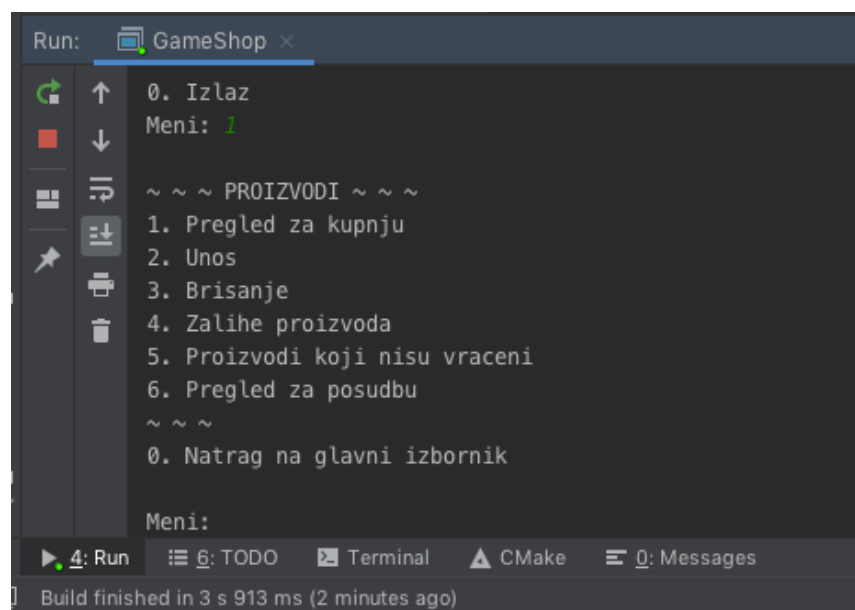
Slika 41 – Pogrešno uneseni korisnički podaci i blokiranje pristupa bazi

Ukoliko se uspješno spojimo u bazu s našim korisničkim podacima otvara nam se glavni izbornik naše aplikacije (Slika 42).



Slika 42 – Izgled glavnog izbornika unutar aplikacije

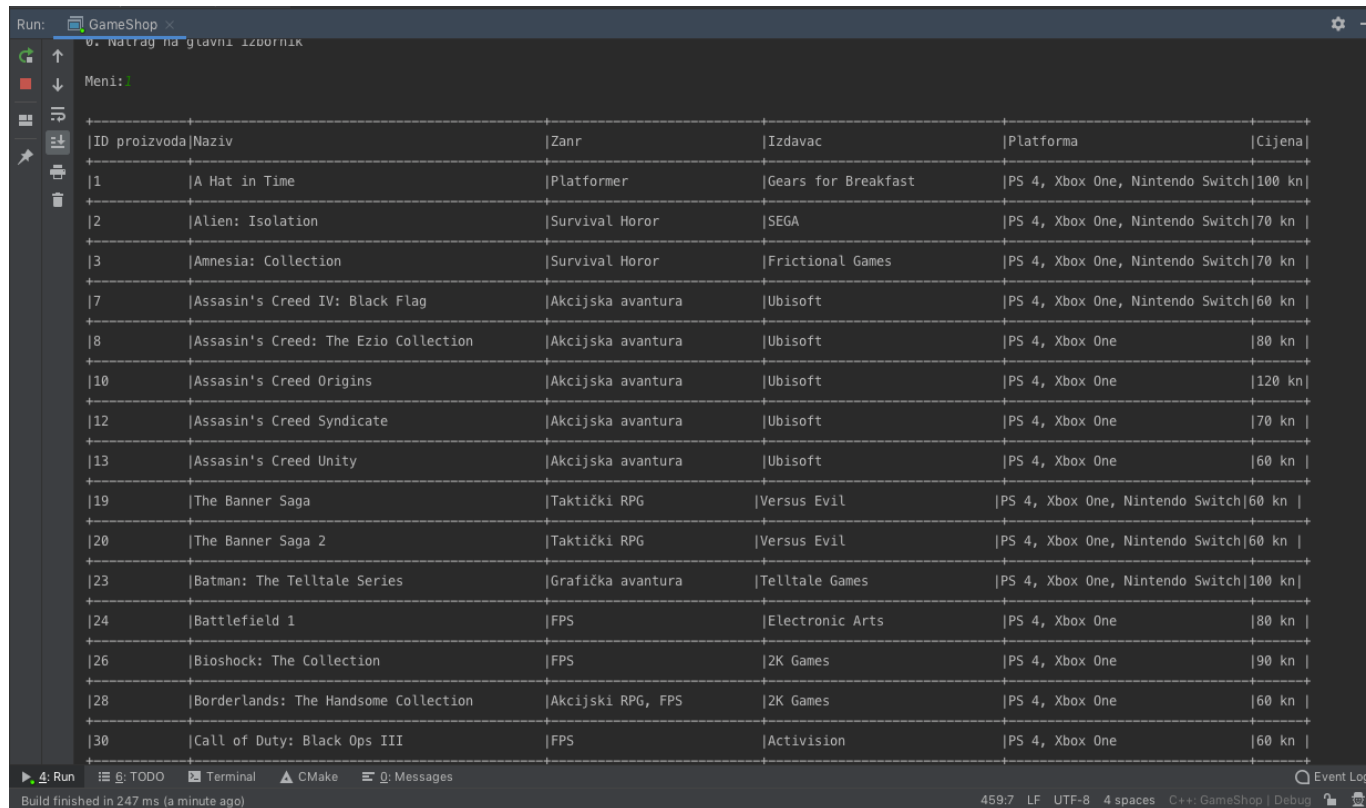
Unosom nekog od brojeva unutar izbornika otvaramo „podizbornik“ sa svim potrebnim operacijama za željenu tablicu. Tako odabirom opcije „1“ otvaramo podizbornik za tablicu „Proizvodi“ (Slika 43).



Slika 43 – Podizbornik tablice Proizvodi i sve dostupne opcije

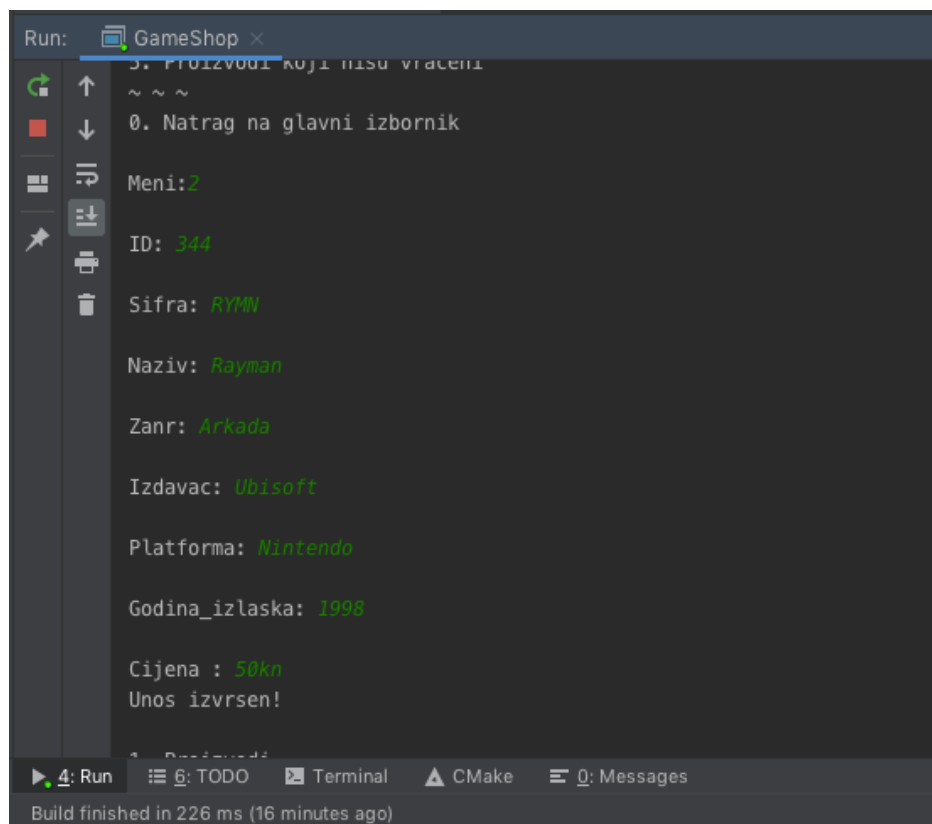
Vidimo tako da za tablicu „Proizvodi“ imamo još 6 podizbora: pregled proizvoda dostupnih za kupnju (jedan od prethodno definiranih pogleda u našoj bazi), unos novih podataka u tablicu, brisanje podataka iz tablice, zaliha proizvoda (procedura), popis proizvoda koji nisu vraćeni (procedura) i pregled proizvoda dostupnih za posudbu (pogled).

Kako smo većinu ovih unosa objasnili kroz prethodna poglavlja, prikazati ćemo samo prva tri izbora da vidimo kako nam to aplikacija prikazuje podatke iz baze, te da li funkcionira dodavanje/brisanje proizvoda iz tablice (Slike 44, 45, 46).

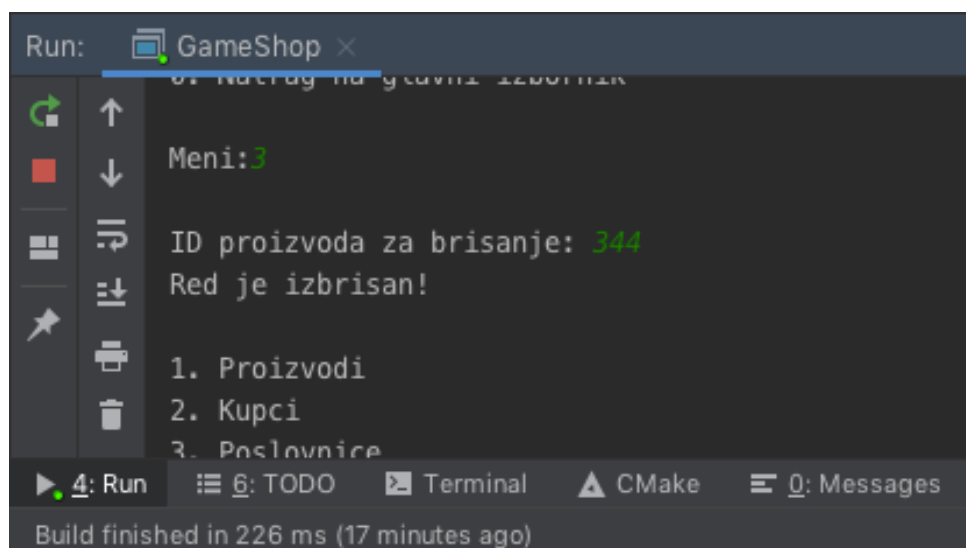


| ID proizvoda | Naziv | Zanr | Izdavac | Platforma | Cijena |
|--------------|---------------------------------------|-------------------|---------------------|---------------------------------|--------|
| 1 | A Hat in Time | Platformer | Gears for Breakfast | PS 4, Xbox One, Nintendo Switch | 100 kn |
| 2 | Alien: Isolation | Survival Horror | SEGA | PS 4, Xbox One, Nintendo Switch | 70 kn |
| 3 | Amnesia: Collection | Survival Horror | Frictional Games | PS 4, Xbox One, Nintendo Switch | 70 kn |
| 7 | Assassin's Creed IV: Black Flag | Akcijska avantura | Ubisoft | PS 4, Xbox One, Nintendo Switch | 60 kn |
| 8 | Assassin's Creed: The Ezio Collection | Akcijska avantura | Ubisoft | PS 4, Xbox One | 80 kn |
| 10 | Assassin's Creed Origins | Akcijska avantura | Ubisoft | PS 4, Xbox One | 120 kn |
| 12 | Assassin's Creed Syndicate | Akcijska avantura | Ubisoft | PS 4, Xbox One | 70 kn |
| 13 | Assassin's Creed Unity | Akcijska avantura | Ubisoft | PS 4, Xbox One | 60 kn |
| 19 | The Banner Saga | Taktički RPG | Versus Evil | PS 4, Xbox One, Nintendo Switch | 60 kn |
| 20 | The Banner Saga 2 | Taktički RPG | Versus Evil | PS 4, Xbox One, Nintendo Switch | 60 kn |
| 23 | Batman: The Telltale Series | Grafička avantura | Telltale Games | PS 4, Xbox One, Nintendo Switch | 100 kn |
| 24 | Battlefield 1 | FPS | Electronic Arts | PS 4, Xbox One | 80 kn |
| 26 | Bioshock: The Collection | FPS | 2K Games | PS 4, Xbox One | 90 kn |
| 28 | Borderlands: The Handsome Collection | Akcijski RPG, FPS | 2K Games | PS 4, Xbox One | 60 kn |
| 30 | Call of Duty: Black Ops III | FPS | Activision | PS 4, Xbox One | 60 kn |

Slika 44 – Lista proizvoda dostupnih za kupnju

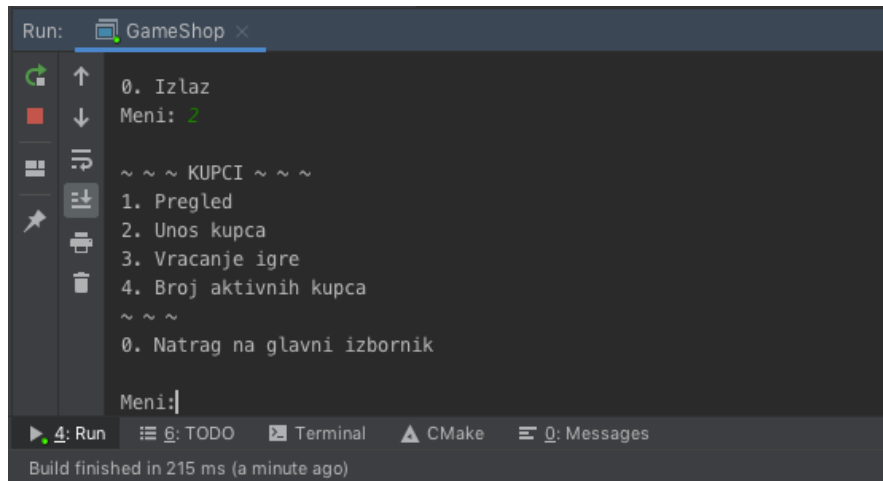


Slika 45 – Unos novog proizvoda u tablicu



Slika 46 – Brisanje prethodno unesenog proizvoda

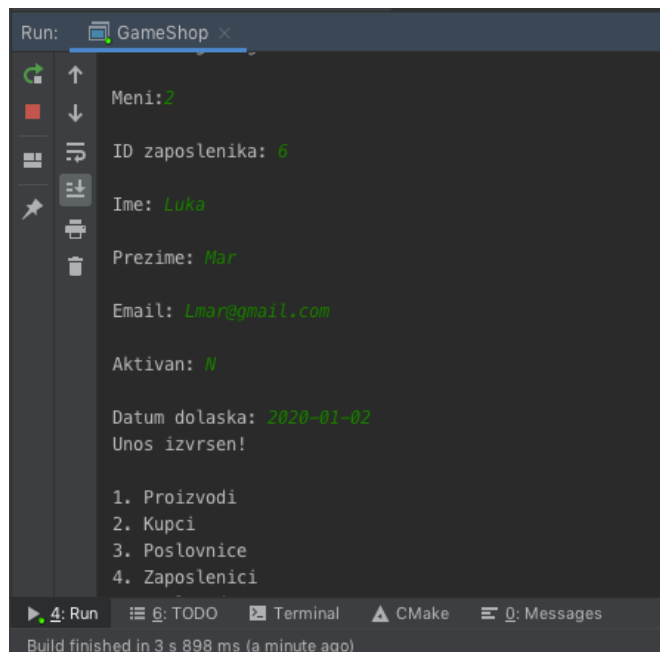
Vidjeli smo kako to funkcionira za tablicu proizvodi, pa povratkom na glavni izbornik bismo sljedeću tablicu, odnosno unosimo broj „2“ kako bi pristupili svim stavkama tablice „Kupci“ (Slika 47).



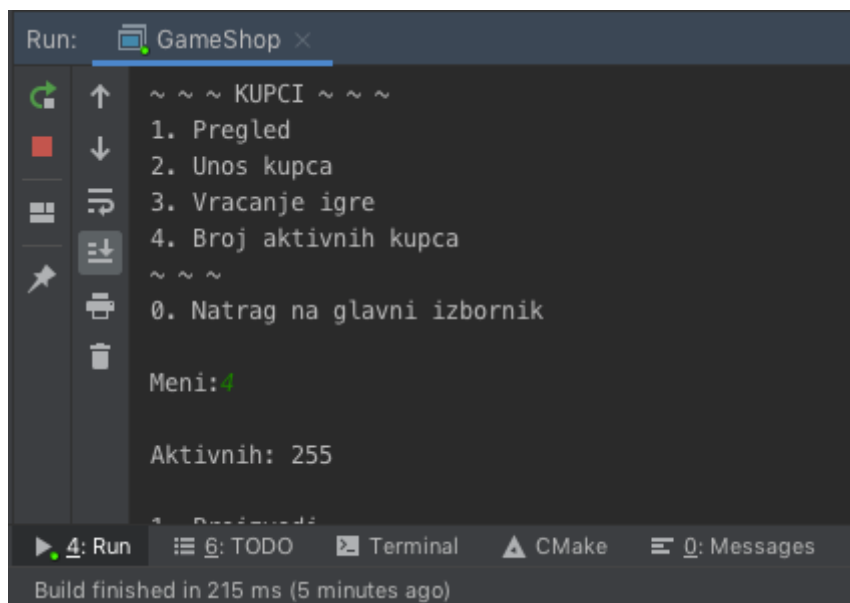
Slika 47 – Podizbornik tablice „Kupci“

Vidimo da za ovu tablicu imamo nešto manje izbora nego za tablicu „Proizvod“, a izbori koji su nam dostupni su sljedeći: pregled kompletne tablice sa svim stupcima, unos novog kupca (procedura), vraćanje igre (procedura), te broj aktivnih kupaca (funkcija).

Kako bi vidjeli funkcioniraju li odabiri za ovu tablicu, odabrati ćemo opciju unosa kupca, te funkciju za povrat broja aktivnih kupaca (Slike 48 i 49).



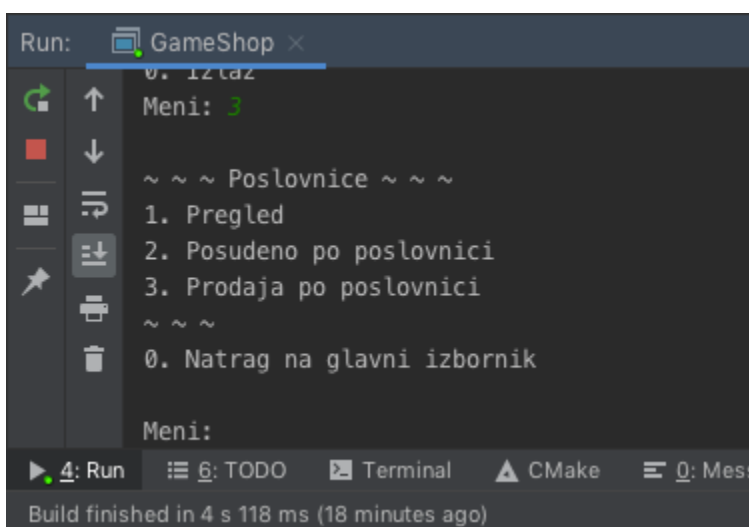
Slika 48 – Procedura za unos novog kupca



Slika 49 – Broj aktivnih kupaca

Vidimo da i kod tablice „Kupci“ sve funkcionira kako bi trebalo (uključujući i dva preostala izbora), te se stoga opet vraćamo na glavni izbornik kako bi odabrali sljedeću tablicu.

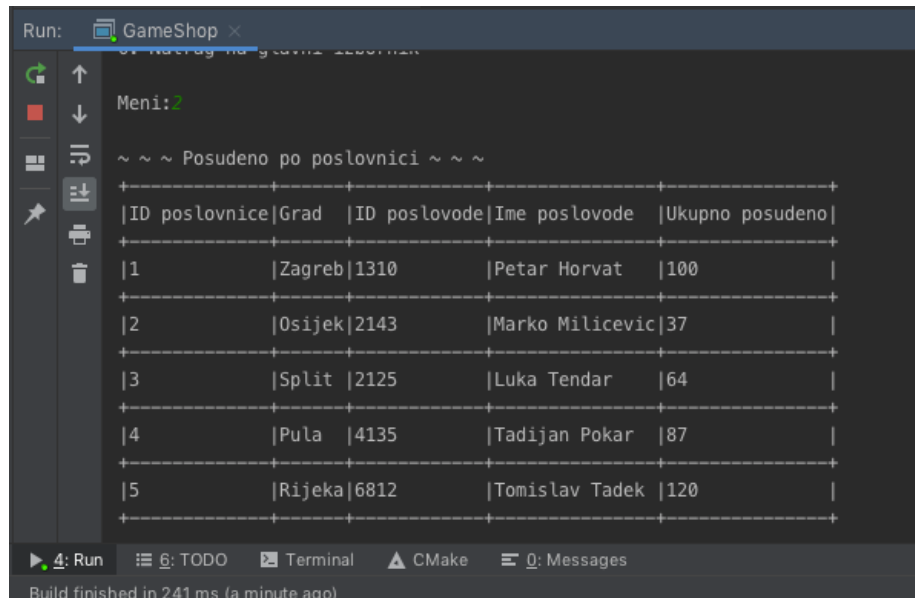
Idući izbor na glavnom izborniku nam je pristup stavkama tablice „Poslovnice“, pa smo to i odabrali, te nam se otvorio podizbornik navedene tablice (Slika 50).



Slika 50 – Podizbornik tablice „Poslovnice“

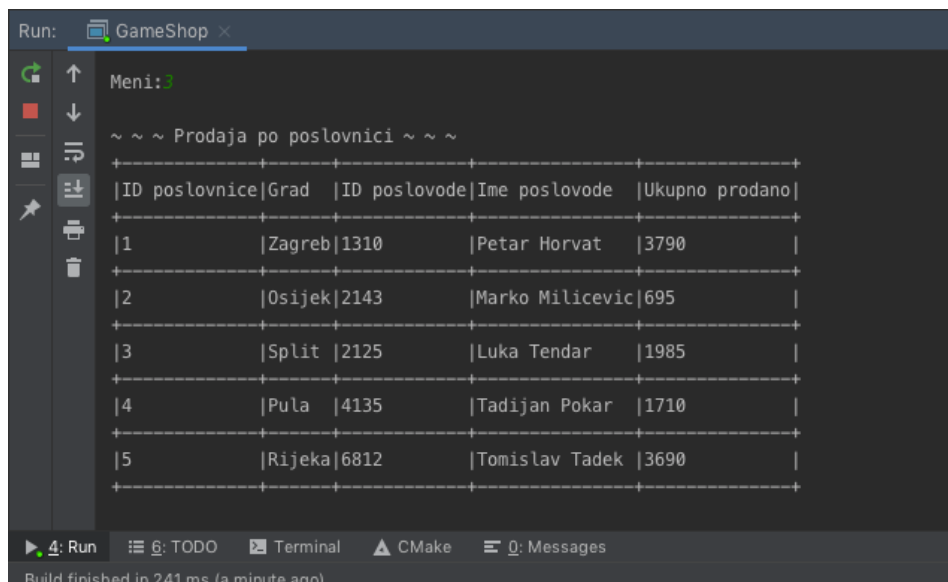
U ovoj tablici imamo samo tri izbora vezana uz nju, a to su: pregled sadržaja tablice, te dva prethodno kreirana pogleda – jedan za ukupnu vrijednost svih prodanih proizvoda u poslovnici, a drugi je za ukupan broj posuđenih proizvoda u poslovnici.

Za prikaz ćemo ovdje odabrati oba pogleda da vidimo hoće li nam aplikacija ispravno prikazati definirane poglede (Slike 51 i 52).



| ID poslovnice | Grad | ID poslovođe | Ime poslovođe | Ukupno posuđeno |
|---------------|--------|--------------|-----------------|-----------------|
| 1 | Zagreb | 1310 | Petar Horvat | 100 |
| 2 | Osijek | 2143 | Marko Milicevic | 37 |
| 3 | Split | 2125 | Luka Tendar | 64 |
| 4 | Pula | 4135 | Tadijan Pokar | 87 |
| 5 | Rijeka | 6812 | Tomislav Tadek | 120 |

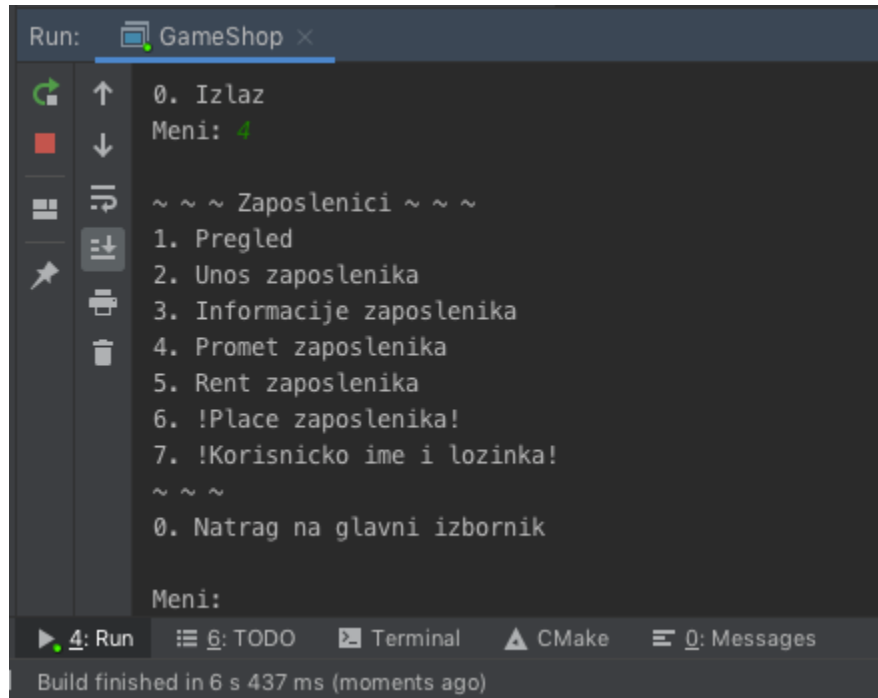
Slika 51 – Ukupan broj posuđenih proizvoda po poslovnici



| ID poslovnice | Grad | ID poslovođe | Ime poslovođe | Ukupno prodano |
|---------------|--------|--------------|-----------------|----------------|
| 1 | Zagreb | 1310 | Petar Horvat | 3790 |
| 2 | Osijek | 2143 | Marko Milicevic | 695 |
| 3 | Split | 2125 | Luka Tendar | 1985 |
| 4 | Pula | 4135 | Tadijan Pokar | 1710 |
| 5 | Rijeka | 6812 | Tomislav Tadek | 3690 |

Slika 52 – Ukupan iznos (vrijednost) svih prodanih proizvoda po poslovnici

Po povratku u glavni izbornik odabiremo sljedeću tablicu po redu, a to je tablica „Zaposlenici“ (Slika 53).



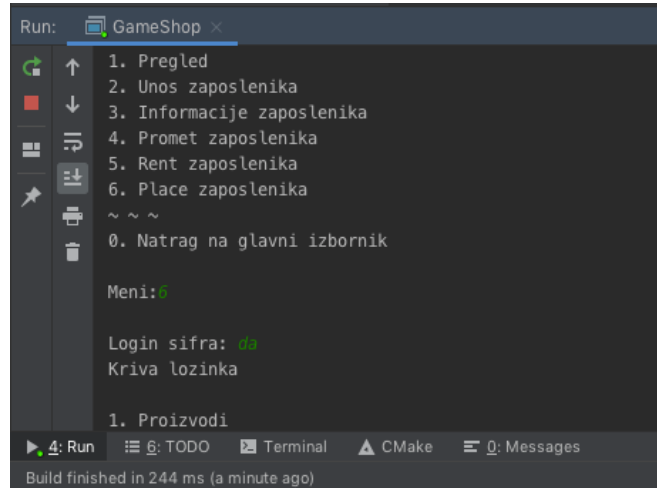
Slika 53 – Podizbornik tablice „Zaposlenici“

Vidimo da za ovu tablicu imamo velik broj podizbora unutar izbornika, a to su sljedeći izbori: pregled cijele tablice, unos novog zaposlenika, informacije o zaposleniku (procedura), promet zaposlenika (procedura), rent zaposlenika (procedura), plaće zaposlenika (pogled), te njihova korisnička imena i lozinke (pogled).

Vidimo da su dva posljednja izbora markirana sa „!“, ali što nam to znači ?

Naime, ta dva pogleda kreirali smo s ciljem da im ne može svatko „tek tako“ pristupiti, odnosno da im mogu pristupiti samo administratori same baze ili eventualno njihove poslovođe. Stoga smo te poglede zaštitili sa šifrom (šifra koju koristimo i prilikom prijave u bazu/aplikaciju).

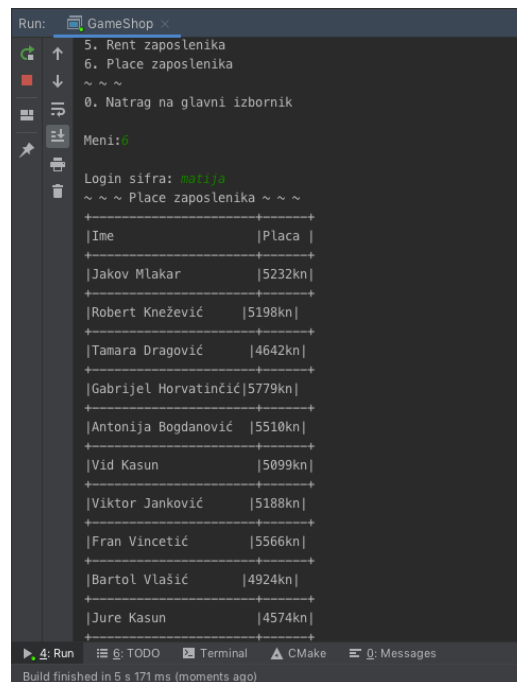
Na sljedećoj slici vidimo kako taj pristup pogledu sa šifrom funkcioniра, te što će se dogoditi ukoliko unesemo krivu šifru (Slika 54).



Slika 54 – Pokušaj pristupanja pogledu sa krivo unesenom šifrom

Vidimo da ukoliko unesemo krivu šifru aplikacija nas o tome obavještava, te nam ne dopušta pristup željenom pogledu.

Na iduće dvije slike vidimo i kako izgledaju ti pogledi kada se uspješno otvore uz valjanu šifru (Slike 55 i 56).



Slika 55 – Lista zaposlenika i njihovih plaća

```
Run: GameShop x
~ ~ ~
0. Natrag na glavni izbornik
Meni: 7
Login sifra: marija
~ ~ ~ Korisnicko ime i lozinka ~ ~ ~
+-----+-----+
|Username|lozinka|
+-----+-----+
|Jakov Mlakar|Jmlakar2|
+-----+-----+
|Robert Knežević|Rknez12|
+-----+-----+
|Tamara Dragović|tamdrag8|
+-----+-----+
|Gabrijel Horvatinčić|gabiho|
+-----+-----+
|Antonija Bogdanović|AntonijaB|
+-----+-----+
|Vid Kasun|vidk9|
+-----+-----+
|Viktor Janković|vikjanko93|
+-----+-----+
|Fran Vincetić|franvi|
+-----+-----+
4: Run 6: TODO Terminal CMake 0: Messages
Build finished in 371 ms (a minute ago)
```

Slika 56 – Lista zaposlenika s njihovim imenima i lozinkama

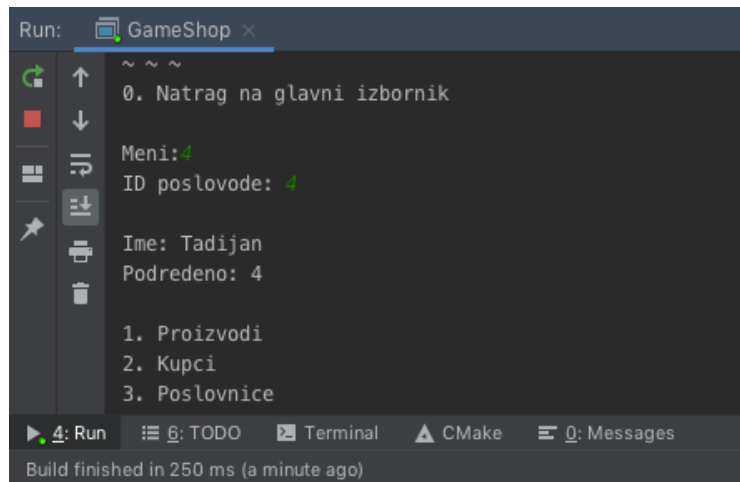
Vidimo da s unosom valjane lozinke sve funkcioniра kako smo željeli, te se stoga vraćamo ponovo na glavni izbornik i odabiremo sljedeću tablicu „Poslovođa“ (Slika 57).

```
Run: GameShop x
7. Racuni
-----
8. !Ostale relacije!
0. Izlaz
Meni: 5
~ ~ ~ Poslovoda ~ ~ ~
1. Pregled
2. Unos
3. Mjesecne rezije
4. Podređeni
5. !Korisnicko ime i lozinka!
6. !Place poslovoda!
~ ~ ~
0. Natrag na glavni izbornik
Meni:
```

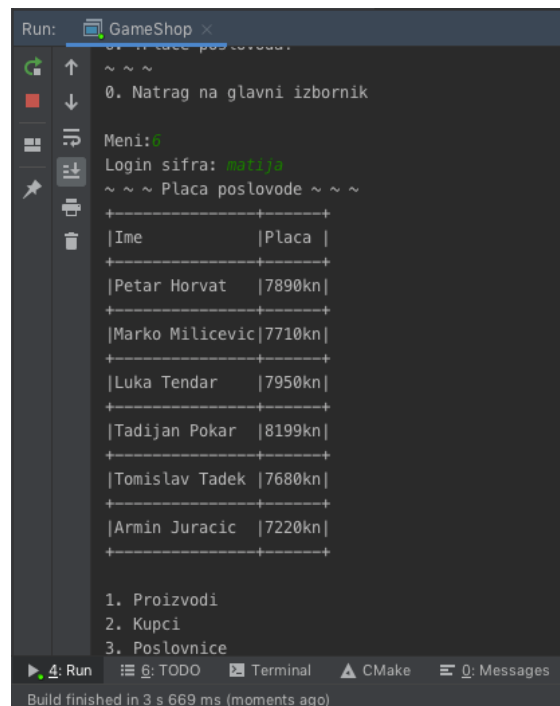
Slika 57 – Podizbornik tablice „Poslovođa“

U ovom podizborniku također vidimo nešto veći broj izbora za tablicu, a to su slijedeći: pregled cijele tablice, unos novog poslovođa, mjesečne režije za određenu poslovnicu pod nekim od poslovođa (procedura), podređene zaposlenike (procedura), te zaključane poglede na korisnička imena i lozinke poslovođa i njihove plaće.

S obzirom da smo u prošloj tablici provjerili funkcionalnost zaključanih pogleda kod ove tablice ćemo za primjer uzeti samo pogled za plaće poslovođa, te proceduru za podređene zaposlenike (Slike 58 i 59).

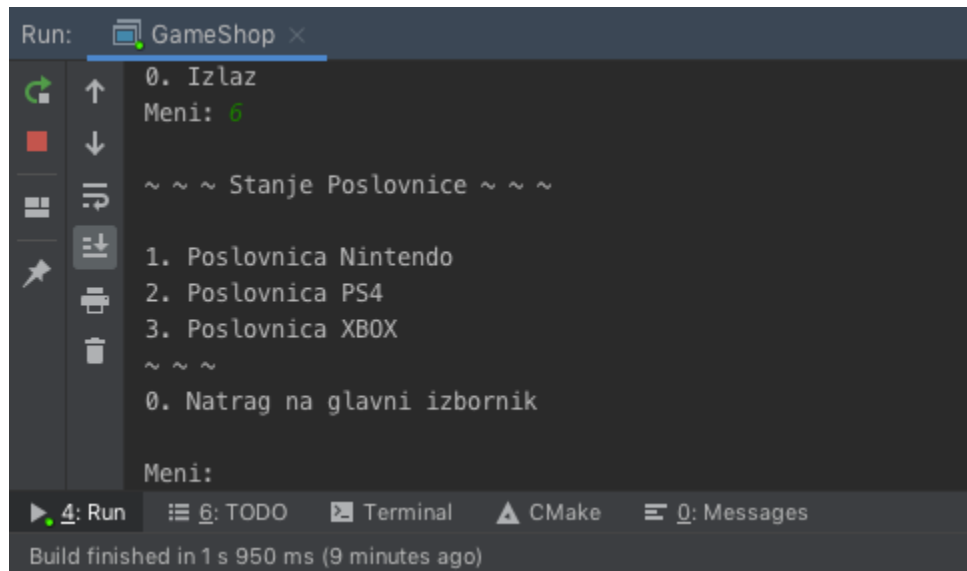


Slika 58 – Procedura koja vraća broj podređenih radnika određenog poslovođe



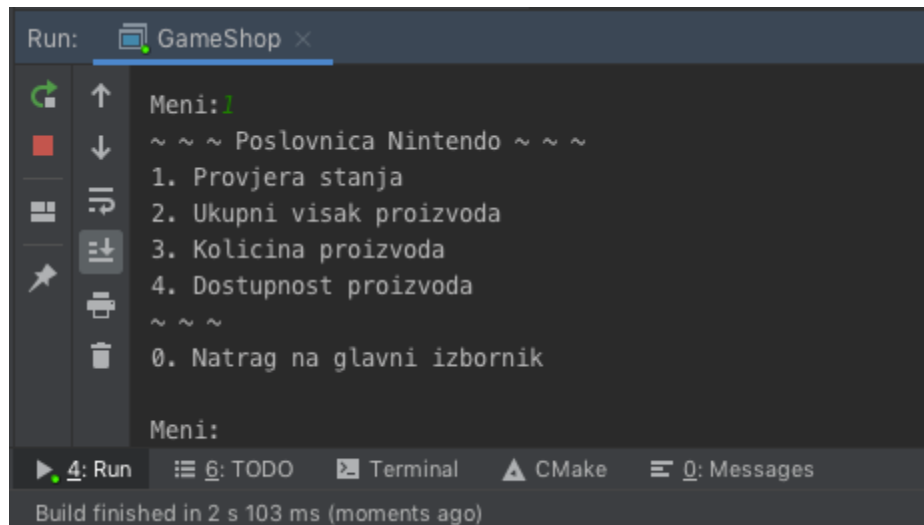
Slika 59 – Lista imena svih poslovođa i njihovih plaće

Povratkom na glavni izbornik odabiremo sljedeću tablicu s glavnog izbornika, a to su tablice „Stanje Poslovnice“ (Slika 60).



Slika 60 – Podizbornik tablica „Stanje Poslovnice“

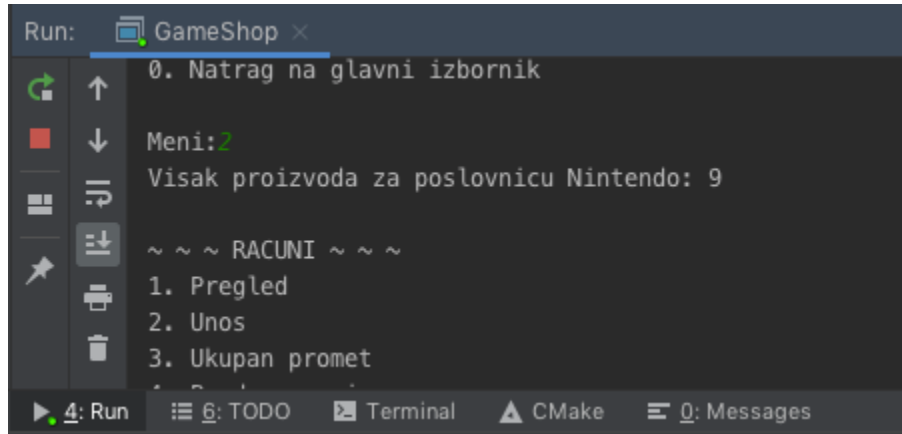
Pošto otprije znamo da imamo **tri** tablice sa stanjima poslovnica, odnosno za svaku konzolu po jedna, vidimo da i u aplikaciji nam ovaj odabir ustvari otvara klasičan podizbornik, ali u kojem potom biramo željeni „pod-podizbornik“ za svaku od željenih tablica. Odlučili smo se vidjeti kako izgleda konačni izbornik za tablicu „Stanje_poslovnica_Nintendo“ (Slika 61).



Slika 61 – Konačni izbornik za tablicu „Stanje_poslovnica_Nintendo“

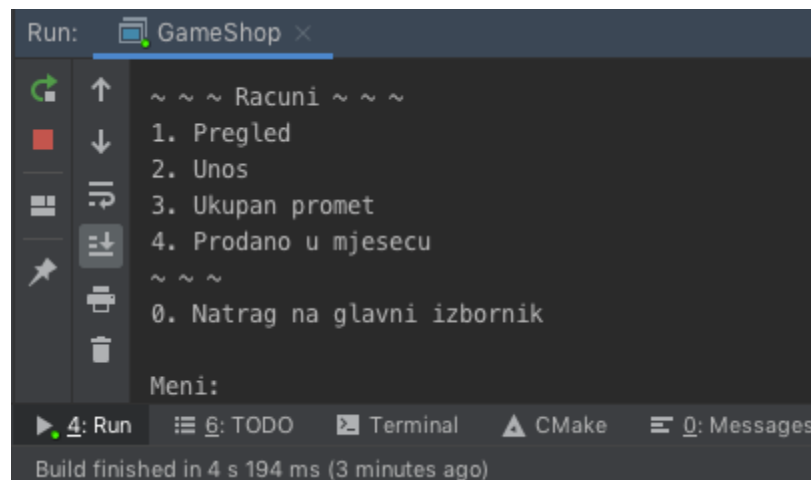
Vidimo da za tablicu stanja poslovnica imamo 4 konačna izbora, a to su: provjera proizvoda koji su nedostupni (procedura), ukupan višak proizvoda (funkcija), količina proizvoda (funkcija) i ukupan broj dostupnih proizvoda (funkcija).

Pošto je ovo relativno jednostavna tablica, uzeti ćemo samo jedan unos za provjeru, a neka to bude funkcija za višak proizvoda u tablici, tj. broj proizvoda kojima količina u nekoj od poslovnica prelazi 10 primjeraka (Slika 62).



Slika 62 – Broj proizvoda koji imaju višak primjeraka na skladištu

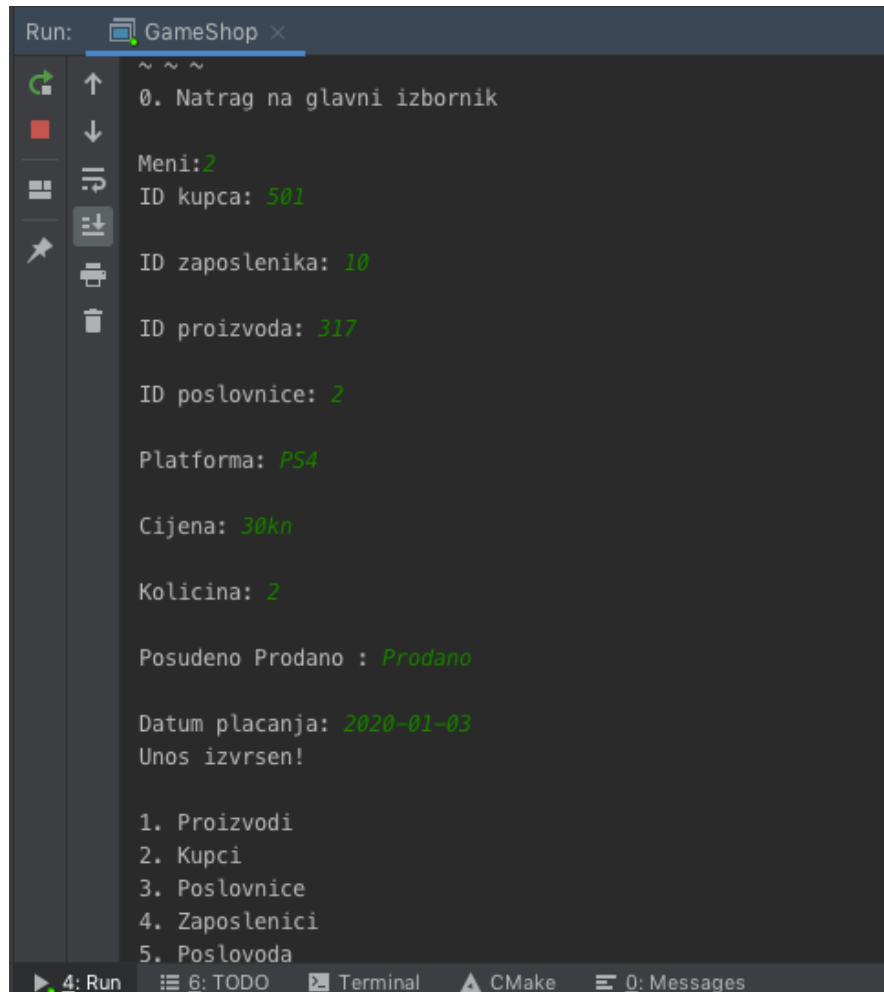
S obzirom da su izbori za stanja poslovnica za PS4 i Xbox potpuno identični kao i u tablici za stanje poslovnica Nintendo konzole, vraćamo se na glavni izbornik i odabiremo sljedeću tablicu, a to je tablica „Računi“ (Slika 63).



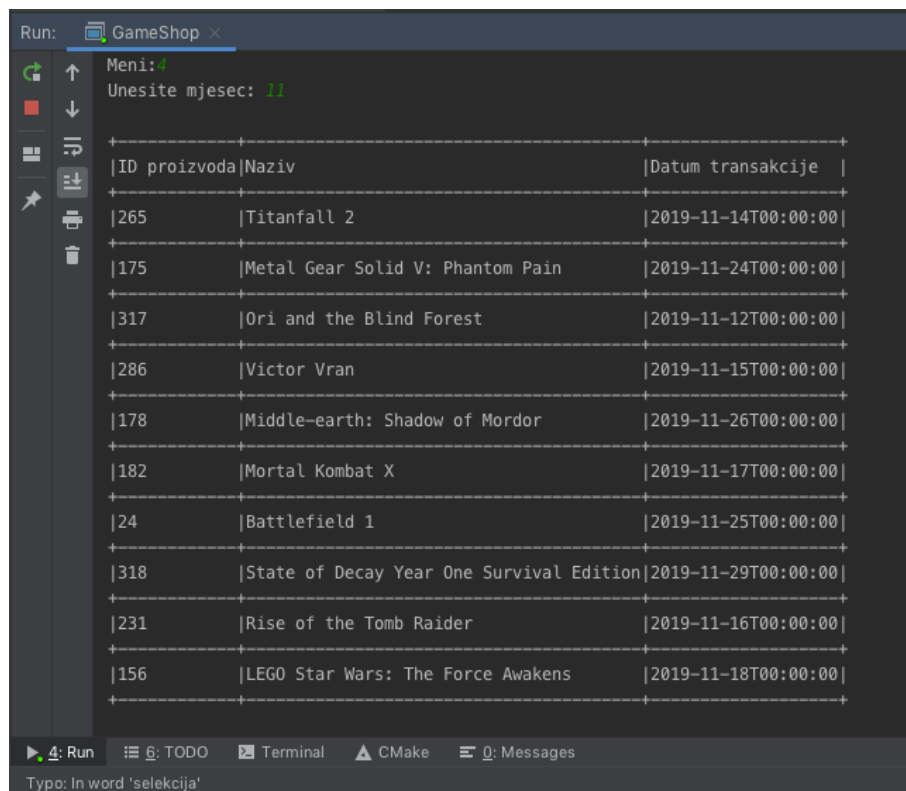
Slika 63 – Podizbornik tablice „Računi“

Vidimo da kao i kod skoro svake prethodne tablice imamo za odabir klasičan pregled cijele tablice, te unos novog računa (u ovom slučaju je to procedura), kao i funkciju za ukupan promet, te proceduru za popis prodanih proizvoda u nekom mjesecu.

U slučaju ove tablice probati ćemo izvršiti unos jednog novog računa, te ćemo provjeriti koje su to igre prodane u studenom (Slike 64 i 65).

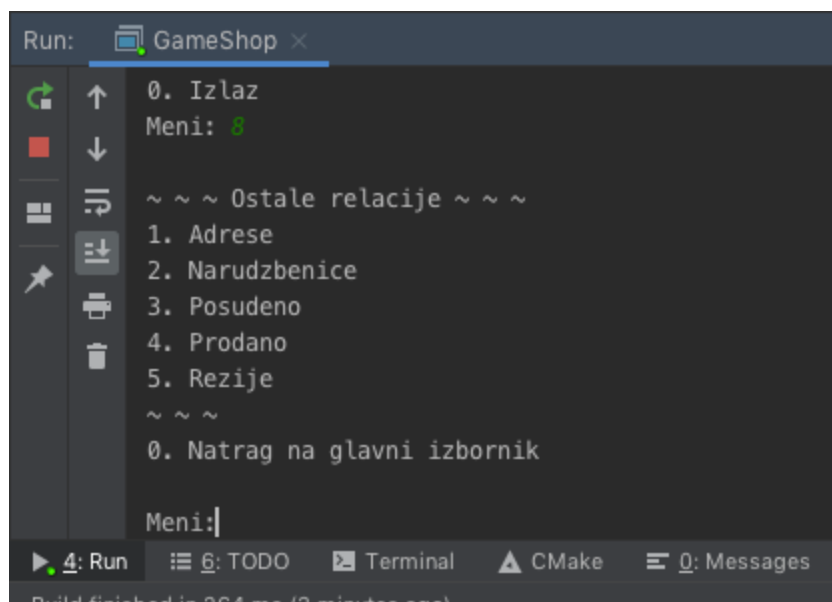


Slika 64 – Unos novog računa



Slika 65 – Lista prodanih igara u studenom

Vraćamo se ponovno u glavni izbornik te odabiremo posljednji izbor na njemu, a to je izbor pod brojem 8 – „Ostale relacije“ (Slika 66).



Slika 66 – Podizbornik „Ostale relacije“

Vidimo da u ovom podizborniku imamo ustvari listu svih preostalih tablica. One su svrstane u ovaj podizbornik pošto nam nisu toliko „ključne“, odnosno podaci iz njih nam nisu prečesto potrebni. Za njih nećemo nikakve detaljne obrade raditi s obzirom da svi nude identične „pod-podizbornike“ u kojima možemo samo izvršavati pregled tih tablica, unos novih vrijednosti ili brisanje postojećih.

Važno je za napomenuti da UPDATE naredbu nismo dodavali kao izbor ni u jednu tablicu unutar aplikacije, a to je iz razloga što je izrazito kompleksna za implementaciju unutar ovakve aplikacije, odnosno stvarala nam je previše problema i glavobolja prilikom pokušaja implementiranja, te smo u konačnici odustali od iste. Update određenih redaka unutar tablice se ipak i dalje može normalno odrađivati unutar sučelja SQL Developera.

10. QA – ispitivanje kvalitete i točnosti dokumentacije, baze i aplikacije

Kolega Cukon je u našem timu bio zadužen za pregled sve napisane dokumentacije, provjeru točnosti podataka u samoj bazi, te dodatnu provjeru funkcionalnosti i točnosti rada aplikacije.

Prilikom provjere kvalitete i ispravnosti dokumentacije kolega je naišao na sljedeće nepravilnosti unutar iste:

- Kontradiktornosti u pojedinim rečenicama
- Nelogičnosti u nekim rečenicama (krivi raspored riječi ili gubljenje smisla samih rečenica)
- Gramatičke i pravopisne pogreške
- Nenamjerne neusklađenosti fontova i njihovih veličina u pojedinim dijelovima teksta
- Pogrešni brojevi slika na pojedinim mjestima

Iako djeluje da je grešaka u prvoj verziji dokumentacije bilo podosta, one su bile na svega par mjesta unutar cijelog dokumenta što je i razumljivo s obzirom na opseg i veličinu ovoga izvješća. Navedene greške su naravno na kraju ispravljene.

Prilikom provjere kvalitete i ispravnosti baze podataka unutar SQL Developera kolega je ukazao na sljedeće probleme:

- Nepoštivanje normalnih formi unutar svih tablica
- Sitne greške u nazivima tablica ili stupaca

Što se tiče prvonavedenog problema kolegi je naknadno objašnjeno da je to namjerno urađeno iz razloga koje smo prethodno objasnili i u ovom dokumentu. Greške prilikom imenovanja tablica i stupaca su kasnije ispravljene uz pomoć UPDATE naredbi.

Prilikom provjere kvalitete i ispravnosti aplikacije za upravljanje bazom kolega je ukazao na sljedeće greške:

- U pogledu „Korisničko ime i lozinka“ u tablicama „Zaposlenici“ i „Poslovođa“ umjesto stupca sa korisničkim imenima korištena su prava imena zaposlenika i poslovođa
- U tablici „Poslovođa“ kao podizbor je dodana procedura „Mjesečne režije“, iako ta procedura i nema pretjerane veze s tom tablicom, već bi trebala biti dodana kao podizbor u tablicu „Režije“

Obje greške na koje je kolega ukazao unutar aplikacije su prije predaje konačne dokumentacije otklonjene, odnosno ispravljene.

11. Zaključak i zahvale

Na kraju cijele ove “avanture” da je tako nazovemo možemo svi reći da smo izrazito zadovoljni i sretni s obujmom i kvalitetom naše baze i aplikacije.

U konačnici naša baza se sastoji od :

- 15 tablica sa ukupno 103 stupca
- približno 6800 unosa
- 10 pogleda
- 12 procedura
- 11 funkcija
- 6 okidača
- 2 transakcije
- potpuno funkcionalne aplikacije od približno 1700 linija koda

Smatramo da je ovaj projektni zadatak bio izrazito poučno iskustvo za svakoga od nas. Produbili smo svoja znanja o programiranju u SQL-u i C++-u, te smo naučili koliko je važan timski rad i međusobno surađivanje za uspjeh ovakvog projekta.

Mogli smo na ovaj način bar približno osjetiti kako bi to moglo izgledati u realnom poslovnom okruženju, te s kojim bi se problemima i stresovima, ali i “nagradama” mogli susresti jednoga dana kao nadamo se uspješni poslovni ljudi unutar informatičke struke.

Također, s obzirom na opseg i detaljnost našeg projekta odnosno baze, smatramo da bi ona uz minimalne dorade i dopune (normalizacija svih tablica, dodavanje još nekih funkcionalnosti, naprednije sučelje...) mogla bez problema poslužiti za pokretanje nekog sličnog poduzeća odnosno Game Shopa.

U konačnici, u ime naše voditeljice Mie Rovis od srca se zahvaljujem svim kolegama u timu, a pogotovo programerskom dijelu tima koji je hrabro iznio najveći i najteži dio posla.

Ja osobno, kao pisac ove dokumentacije bi se posebno htio zahvaliti kolegi Đurašinoviću, koji iako je imao primarnu funkciju programera, znatno je pomogao i u stvaranju ove kompletne dokumentacije u vidu upućivanja mene u sve etape kreiranja baze podataka i aplikacije, te objašnjavanja svih njihovih funkcionalnosti, kao i za pribavljanje većine potrebnih slika za potrebe ovoga dokumenta i prezentacije.

12. Korisne poveznice

1. <https://dev.mysql.com/downloads/installer/> → **MySQL Installer**
2. <https://www.oracle.com/tools/downloads/sqldev-v192-downloads.html> → **SQL Developer**
3. <https://www.jetbrains.com/clion/> → **Clion**
4. <https://www.sqlapi.com/> → **SQLApi++**