

Analiza izvršavanja SQL upita na različitim vrstama relacijskih baza podataka

Vidak, Josip

Undergraduate thesis / Završni rad

2023

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:166:667420>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2025-02-04**

Repository / Repozitorij:

[Repository of Faculty of Science](#)



SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET

ZAVRŠNI RAD

**ANALIZA IZVRŠAVANJA SQL UPITA NA
RAZLIČITIM VRSTAMA RELACIJSKIH BAZA
PODATAKA**

Josip Vidak

Split, rujan 2023

Temeljna dokumentacijska kartica

Završni rad

Sveučilište u Splitu

Prirodoslovno-matematički fakultet

Odjel za informatiku

Ruđera Boškovića 33, 21000 Split, Hrvatska

ANALIZA IZVRŠAVANJA SQL UPITA NA RAZLIČITIM VRSTAMA RELACIJSKIH BAZA PODATAKA

Josip Vidak

SAŽETAK

U ovom radu opisane su glavne karakteristike relacijskih baza podataka, prikazan je postupak stvaranja, te usporedba izvršavanja upita na različitim vrstama relacijskih baza podataka. Također je prikazan razvoj testne aplikacije, te konačni rezultati mjerenja opterećenja sustava i vremena izvršavanja kod obje baze podataka u tabličnom i grafičkom obliku.

Ključne riječi: baze podataka, MS SQL Server, MySQL, SQL upiti

Rad je pohranjen u knjižnici Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Rad sadrži: 27 stranica, 18 grafičkih prikaza i 9 literaturnih navoda. Izvornik je na hrvatskom jeziku.

Mentor: **Doc. dr. sc. Divna Krpan**, *docent Prirodoslovno-matematičkog fakulteta u Splitu, Sveučilišta u Splitu*

Neposredni voditelj: **Dino Nejašmić, mag. educ. math. et inf.**, *predavač Prirodoslovno-matematičkog fakulteta u Splitu, Sveučilišta u Splitu*

Ocjenjivači: **Doc. dr. sc. Divna Krpan**, *docent Prirodoslovno-matematičkog fakulteta u Splitu, Sveučilišta u Splitu*

Dino Nejašmić, mag. educ. math. et inf., *predavač Prirodoslovno-matematičkog fakulteta u Splitu, Sveučilišta u Splitu*

Antonela Prnjak, mag. educ. inf., *asistent Prirodoslovno-matematičkog fakulteta u Splitu, Sveučilišta u Splitu*

Rad prihvaćen: **rujan, 2023**

Basic documentation card

Thesis

University of Split

Faculty of Science and Mathematics

Department of computer science

Ruđera Boškovića 33, 21000 Split, Croatia

ANALYSIS OF SQL QUERIES EXECUTION ON DIFFERENT TYPES OF RELATIONAL DATABASES

Josip Vidak

ABSTRACT

In this thesis, the main characteristics of relational databases are described, the creation process is presented, and a comparison of query execution on different types of relational databases is presented. The thesis also presents development of the test application and the final results of measuring system usage and database execution time in tabular and graphical form.

Key words: databases, MS SQL Server, MySQL, SQL queries

Thesis deposited in library of Faculty of Science and Mathematics, University of Split

Thesis consists of: 27 pages, 18 figures and 9 references. Original language: Croatian.

Mentor: **Divna Krpan, Ph. D.** *assistant professor at the Faculty of Science and Mathematics in Split, University of Split*

Supervisor: **Dino Nejašmić, mag. educ. math. et inf.,** *lecturer at the Faculty of Science and Mathematics in Split, University of Split*

Reviewers: **Divna Krpan, Ph. D.** *assistant professor at the Faculty of Science and Mathematics in Split, University of Split*

Dino Nejašmić, mag. educ. math. et inf., *lecturer at the Faculty of Science and Mathematics in Split, University of Split*

Antonela Prnjak, mag. educ. inf., *assistant at the Faculty of Science and Mathematics in Split, University of Split*

Thesis accepted: **September, 2023**

IZJAVA

kojom izjavljujem s punom materijalnom i moralnom odgovornošću da sam završni rad s naslovom „ANALIZA IZVRŠAVANJA SQL UPITA NA RAZLIČITIM VRSTAMA RELACIJSKIH BAZA PODATAKA“ izradio samostalno pod voditeljstvom Dina Nejašmića, mag. educ. math. et inf., pred. U radu sam primijenio metodologiju znanstvenoistraživačkog rada i koristio literaturu koja je navedena na kraju završnog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući naveo u diplomskom radu na uobičajen, standardan način citirao sam i povezoao s fusnotama s korištenim bibliografskim jedinicama. Rad je pisan u duhu hrvatskog jezika.

Student:

Josip Vidak

Sadržaj

UVOD	1
1 BAZE PODATAKA	2
1.1 Relacijske baze podataka	3
1.2 Programska podrška	4
1.3 Upiti na bazu podataka	5
1.4 Razrada modela	6
2 PRAKTIČAN RAD	9
2.1 Izrade baza podataka	9
2.2 Generiranje podataka	12
2.3 Test aplikacija	15
2.4 Upiti kroz aplikaciju	18
2.5 Usporedba rezultata	20
ZAKLJUČAK	24
LITERATURA	25
POPIS SLIKA	26
Skraćenice	27

UVOD

U današnjem svakodnevnom životu imamo potrebu za različitim oblicima evidencije podataka, bilo da se radi o potrebama za evidencijom troškova, popisa telefonskih pretplatnika, do nadziranja radnika i radnih mjesta i slično. Lakšu i organiziraniju evidenciju nam omogućuju baze podataka.

Baza podataka je oblik evidencije, organiziranja i prikupljanja podataka. Podatci su ono što baza pohranjuje, mogu biti u obliku slova, riječi, brojeva, slika i slično, a obrađen podatak jest informacija.

U ovom radu su obrađene dvije vrste relacijskih baza podataka, napravljene u različitim sustavima za upravljanje relacijskim bazama podataka, opisani su načini na koje su one izrađene, analizirane su njihove prednosti i nedostaci, te su uspoređene njihove izvedbe SQL upita u vidu odabira, grupiranja, sortiranja, pretrage i slično. Upiti su analizirani kroz jednostavno sučelje koje je izrađeno u Windows Forms aplikaciji.

Cilj ovog rada je ukazati na vrstu relacijske baze koja bolje obrađuje veliki skup podataka, pravi manje opterećenje sustava, te joj je potrebno manje vremena za izvršiti određeni zadatak. Također su prikazana dva različita programa za izradu baza podataka, te su analizirane njihove prednosti i mane.

Rad je podijeljen u četiri poglavlja: Uvod, Baze podataka, Praktičan rad i Zaključak. U poglavlju Baze podataka prikazane su definicije baza podataka i pojmova povezanih s njima, koji su bitni za ovaj rad. U poglavlju Praktičan rad prikazane su izrade samih baza podataka i njihovih tablica, razvoj sučelja za testiranje, generiranje podataka za baze, te je na samom kraju odrađena usporedba dobivenih rezultata iz obje baze.

1 BAZE PODATAKA

Baze podataka nam služe za prikupljanje, evidenciju, te organizaciju podataka. U bazu podataka možemo pohraniti bilo kakav oblik podataka iz svakodnevnoga života, počevši od osoba i podataka o njima, do podataka koji služe za vođenje poduzeća i slično. Također, u baze podataka možemo pohraniti različite vrste podataka, kao što su brojevi, slova, datumi i slično. (Microsoft, Osnove baza podataka)

Mnoge baze podataka iz prva krenu kao neki obični tekstualni podaci, u nekim osnovnim programima za obradu teksta, međutim problem nastaje kada broj zapisa postane dovoljno velik, tako da ga ne možemo više letimično pretraživati. Rješenje tog problema leži u programima, koji su namijenjeni upravo za to da bolje organiziraju skupove podataka. (Microsoft, Osnove baza podataka)

Programi za upravljanje bazama podataka omogućuju:

- Unos novih podataka
- Uređivanje postojećih podataka
- Brisanje podataka
- Sortiranje
- I sl. (Microsoft, Osnove baza podataka)

Jedan od primjera programa za izradu baza podataka jest program tvrtke Microsoft, kojemu je puni naziv Microsoft SQL Server Management Studio, u kojemu je izrađena jedna baza podataka korištena za ovaj rad.

Računalna baza podataka jest zapravo nekakav spremnik objekata, koji može sadržavati jednu ili više tablica. Bitno je znati razliku između tablice i baze, primjerice, neko poduzeće prati svoje zaposlenike, te za to koristi tri tablice, ali to nisu tri baze podataka, nego je to jedna baza, koja sadrži tri tablice. (Microsoft, Osnove baza podataka)

Postoji pet vrsta baza podataka, a to su:

- Plošne - npr. Excel tablice
- Hijerarhijske – zasnivaju se na hijerarhijskim oblicima, poput stabla
- Mrežne – zasnivaju se na mreži povezanih podataka
- Objektne – zasnivaju se na objektno orijentiranim programskim jezicima, u njima su podaci objekti
- Relacijske – podaci su organizirani u tablicama, koje se povezuju primarnim i stranim ključevima (Loomen, Baze podataka)

Neki od najpopularnijih programa koji služe za upravljanje bazama podataka su:

- Oracle Express
- Mongo DB
- Microsoft Access
- Microsoft SQL Server
- MySQL

1.1 Relacijske baze podataka

Ova vrsta baza podataka je izdvojena u posebno pod poglavlje, jer se sam rad odnosi na usporedbu dviju vrsta relacijskih baza podataka, te upita na njih.

Relacijske baze podataka su baze podataka koje svoje podatke raspoređuju u tablice, te između tih tablica postoje veze (relacije). Relacijska baza podataka može sadržavati samo jednu tablicu, ali to je rijetkost. Svaka relacijska baza se sastoji od entiteta i atributa. (Microsoft, Početak rada s bazama podataka)

Entitet(eng. *entity*) je izraz za predmet koji tablica opisuje, kao što su u slučaju ove baze podataka kupci, artikli, radnici i slično. Atribut je pojam koji se veže za ono što opisuje entitet, kao na primjer: ime, prezime, plaća... Zapisi se u tablice raspoređuju u retke i stupce. (Microsoft, Početak rada s bazama podataka)

Redci pohranjuju vrijednosti atributa podataka, kao što bi bilo za ime „Josip“, a stupci označavaju same nazive atributa kao što je „Ime“. Svaka tablica ima svoj primarni ključ(eng. *primary key*), a to je vrijednost koja garantira da je svaki podatak jedinstven. Primjer toga je da se u bazi podataka mogu naći dvije osobe s istim imenom i prezimenom , na primjer „Josip Vidak“ i „Josip Vidak“, ali oba imaju svoj primaran ključ ili ID po kojemu se razlikuju, u jednog može biti 10, a u drugog recimo 20.

Relacijske baze podataka omogućavaju izvršavanje upita, pomoću kojih podaci iz dvije tablice izgledaju kao podaci iz jedne tablice, drugim riječima, tablice se povezuju pomoću primarnih i stranih ključeva(eng. *foreign key*). Također, treba spomenuti sigurnost, jer sustavi za upravljanje relacijskim bazama podataka, omogućuju pristup samo ovlaštenim korisnicima. Najveći problem ove vrste baza podataka jest održavanje, jer, kad baza poprimi previše podataka, njeno održavanje oduzima programerima previše vremena. (DatabaseTown, Relacijske baze podataka prednosti i nedostaci)

Kako bi se poboljšao integritet podataka radi se normalizacija, a to je oblikovanje relacijske baze podataka, kako bi se smanjila redundancija podataka. Normalizacijom organiziramo podatke u tablice, na način da su tablice povezane međusobno primarnim i stranim ključevima, odnosno, primarni ključ jedne tablice može biti strani ključ, odnosno veza na drugu tablicu.. (Wikipedia, Normalizacija baza podataka)

1.2 Programska podrška

Za način izrade projekta odabrani su Microsoft SQL Server i MySQL Workbench, iz razloga jer oba alata spadaju među najpopularnije sustave za upravljanje relacijskim bazama podataka. Također, napravljena je i aplikacija koja služi kao sučelje za izvođenje upita i usporedbu baza . Napravljena u programu Visual Studio 2022, programskim jezikom C#, u obliku Windows Forms aplikacije.

MS SQL Server je program koji je razvila tvrtka Microsoft, a služi za izradu i upravljanje bazama podataka. Program služi kao poslužitelj za baze podataka, koji ima uloge pohranjivanja i dohvaćanja podataka. Zahtjeve pohranjivanje i dohvat podataka mogu vršiti i drugi programi, koji su kompatibilni sa poslužiteljem, na istom, ili s drugog računala, preko interneta. (Wikipedia, Microsoft SQL server)

MySQL je programski alat, koji je namijenjen za administratore baza podataka i programere. Omogućuje modeliranje, upravljanje i razvoj relacijskih baza podataka. Program sadrži sve ono što je potrebno za pravilnu izradu i oblikovanje baza podataka. Također, kao i prethodni program, može komunicirati s kompatibilnim programima preko poslužitelja. (MySQL, MySQL Workbench)

Visual Studio je proizvod Microsoft tvrtke, te služi kao razvojno okruženje za web stranice, web aplikacije, mobilne aplikacije, aplikacije konzole i slično. Podržava 36 različitih programskih jezika, a za ovaj rad korišten je programski jezik C#, te je aplikacija napravljena u obliku Windows Forme. (Wikipedia, Visual Studio)

1.3 Upiti na bazu podataka

Upiti(eng. *queries*) u bazama podataka služe za izvođenje raznih operacija s podacima. Najčešće se koriste za dohvat podataka. Često nailazimo na slučaj kada se podatci nalaze u više tablica (relacijske baze podataka), a mi ih možemo sve dohvatiti u jednoj tablici upravo pomoću SQL upita. Osim što imaju ulogu dohvata podataka, upiti imaju uloge i brisanja, uređivanja i unošenja podataka. Postoje dvije vrste upita, a to su akcijski i upiti odabira. Akcijskim upitima se izvršava određeni zadatak nad podacima. Pomoću njih se podatci unose u postojeće tablice, brišu, uređuju i slično. Kod upita odabira podatci se dohvaćaju, te ispisuju na zaslone. Takve upite je moguće ispitivati, odnosno sortirati, grupirati... (Microsoft, Osnove baza podataka)

Ključne riječi za pisanje SQL upita su:

- *Select*-dohvat podataka
- *Insert*-služi za unos podataka
- *Delete*-brisanje podataka
- *Create*-za stvaranje tablica
- *Update*-uređivanje podataka

Također, postoje ključne riječi koje služe za oblikovanje SQL upita, a to su:

- *Where*-određuje uvjet gdje će se upit izvršiti (npr. where id=1)
- *Order by*-sortiranje prema zadanoj vrijednosti
- *Group by*-grupiranje podataka

Unutar SQL-a postoje i „*join*“ operacije koje služe za obradu podataka iz više tablica (*left*, *right* i *inner*). SQL koristi i neke agregatne funkcije:

- AVG()-vraća prosječnu vrijednost
- SUM()-vraća zbroj
- COUNT()-broji redove podataka
- MAX() i MIN()-vraćaju najvišu/najnižu vrijednost

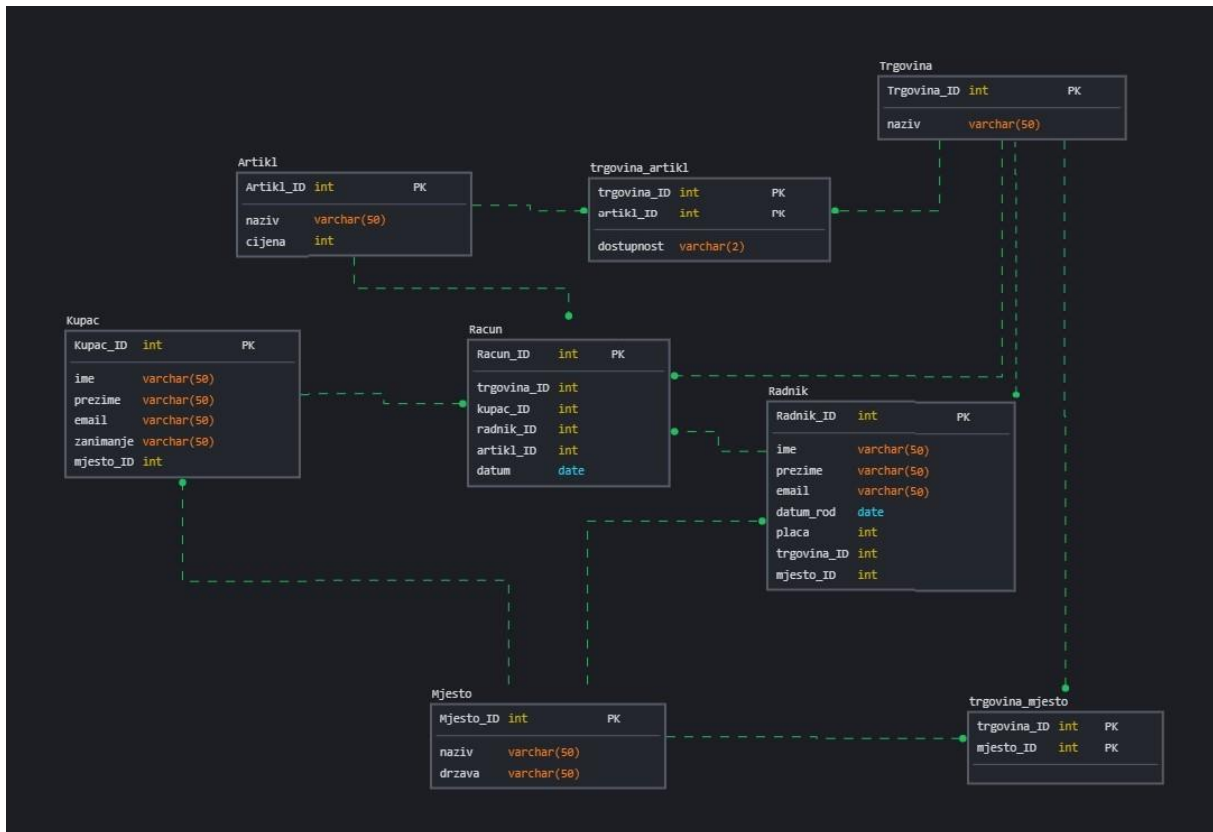
Optimizacijom upita se poboljšava brzina izvođenja, učinkovitost, te se smanjuje opterećenje sustava, a neke od tehnika optimizacije SQL upita su:

- smanjivanje „*join*“ operacija, odnosno, što manje povezanih tablica s upitom koji se izvršava
- korištenje „*select*“ opcije s nazivom atributa koje dohvaćamo, umjesto * (zvjezdica predstavlja odabir svih atributa). (DbForgeTeam, SQL optimizacija upita)

1.4 Razrada modela

Baza podataka, koja je korištena u ovom radu sadrži podatke o kupo-prodaji opreme za IT sektor . Radi se o bazi podataka pruža jednostavan nadzor o cijeni, dostupnosti artikla, informacije o radnicima i kupcima, kao i informacije o samim trgovinama.

Prije implementacije same baze podataka, prvo je napravljena skica, koja predstavlja model baza podataka. Skica za ovaj model baze podataka je izrađena u online aplikaciji za izrade modela baza podataka *SQLDbm* .Na slici 1 je prikazana skica modela baze podataka.



Slika 1 Skica modela baze podataka

Baza podataka je podijeljena u osam tablica, koje su sve međusobno povezane vezom jedan prema više. Problem povezivanja tablica ja bio prisutan u slučaju tablica Trgovine i Mjesta, te Trgovine i Artikla.

Entiteti u bazi podataka su :

- Trgovina
- Kupac
- Radnik
- Artikl
- Mjesto
- Račun

Tablica Kupac ima strani ključ mjesto_ID, jer je tablica kupac povezana primarnim ključem tablice Mjesto preko njega. Isto vrijedi i za tablicu Radnik, koja uz vezu s tablicom Mjesto, sadrži i vezu prema tablici Trgovina. Najviše stranih ključeva se nalazi u tablici Račun, jer je ona središte baze, te svaki račun mora imati svoga kupca, radnika koji ga je izdao, artikl koji je kupljen, te trgovinu u kojoj je kupljen artikl, stoga je bilo potrebno povezati primarne ključeve tih tablica, sa stranim ključevima u tablici račun..

U bazi podataka je dva puta bio prisutan problem veze više prema više, između trgovine i artikla, jer jedan artikl može biti dostupan u više trgovina, a jedna trgovina sadrži više artikala. Problem je riješen među-tablicom trgovina-artikl, gdje imamo kompozitni primaran ključ, koji čine trgovina_ID, te artikl_ID. Tablica također sadrži atribut dostupnost, kojim se određuje je li zadani artikl dostupan u određenoj trgovini. Slična situacija je u vezi tablica trgovine i mjesta, samo ovdje nema dodatnih atributa. Ostale su tablice povezane vezom jedan na više.

Obrazloženje veza:

- Jedan radnik može biti iz jednog mjesta, dok jedno mjesto može imati više radnika.
- Jedan kupac može biti iz jednog mjesta, dok jedno mjesto može imati više kupaca.
- Jedan račun može imati jednu trgovinu, kupca, radnika i artikl, a jedan kupac može imati više računa, jedan radnik ih može izdati više, jedna vrsta artikla se može naći na više računa, jedna trgovina može izdati više računa.
- Jedna vrsta artikla može biti dostupna u više trgovina, a jedna trgovina sadrži više artikala.
- Jedna trgovina može biti u više mjesta, a jedno mjesto može imati više trgovina.
- Jedan radnik može raditi u jednoj trgovini, a trgovina može imati više radnika.

2 PRAKTIČAN RAD

U ovom poglavlju je prikazan način na koji su izrađene same baze podataka, izgled tablica i dijagrama baza, generiranje podataka, izradu aplikacije, te je prikazana sama analiza izvršavanja upita na bazu podataka, kroz aplikaciju.

2.1 Izrade baza podataka

U ovom pod poglavlju je prikazan postupak kojim su MSSQL i MySQL baze podataka izrađene, s pripadajućim dijagramima i tablicama.

Nakon izrade skice modela baze podataka, bilo je potrebno izraditi tablice. Prilikom izrade tablica baze podataka, bilo im je potrebno unijeti odgovarajuće atribute, te odrediti primarne i strane ključeve. Za svaki podatak koji se nalazi u tablici, određen je njegov tip (int-cijeli broj, varchar(50)-tekstni zapis duljine 50, date-datum itd.). Također, trebalo je odrediti sadrže li atributi nulte vrijednosti. Primjer izgleda tablica Kupac u obje baze je prikazan slikom 2 i slikom 3.

Kod 1 Kod za stvaranje tablice Kupac

```
CREATE TABLE Kupac (  
    kupac_ID int NOT NULL PRIMARY KEY,  
    ime varchar(50) NOT NULL,  
    prezime varchar(50) NOT NULL,  
    email varchar(50) NOT NULL,  
    zanimanje varchar(50) NOT NULL,  
    mjesto_ID int NOT NULL,  
)
```


Kupac			
	Column Name	Data Type	Allow Nulls
🔑	Kupac_ID	int	<input type="checkbox"/>
	ime	varchar(50)	<input type="checkbox"/>
	prezime	varchar(50)	<input type="checkbox"/>
	email	varchar(50)	<input type="checkbox"/>
	zanimanje	varchar(50)	<input type="checkbox"/>
	mjesto_ID	int	<input type="checkbox"/>
			<input type="checkbox"/>

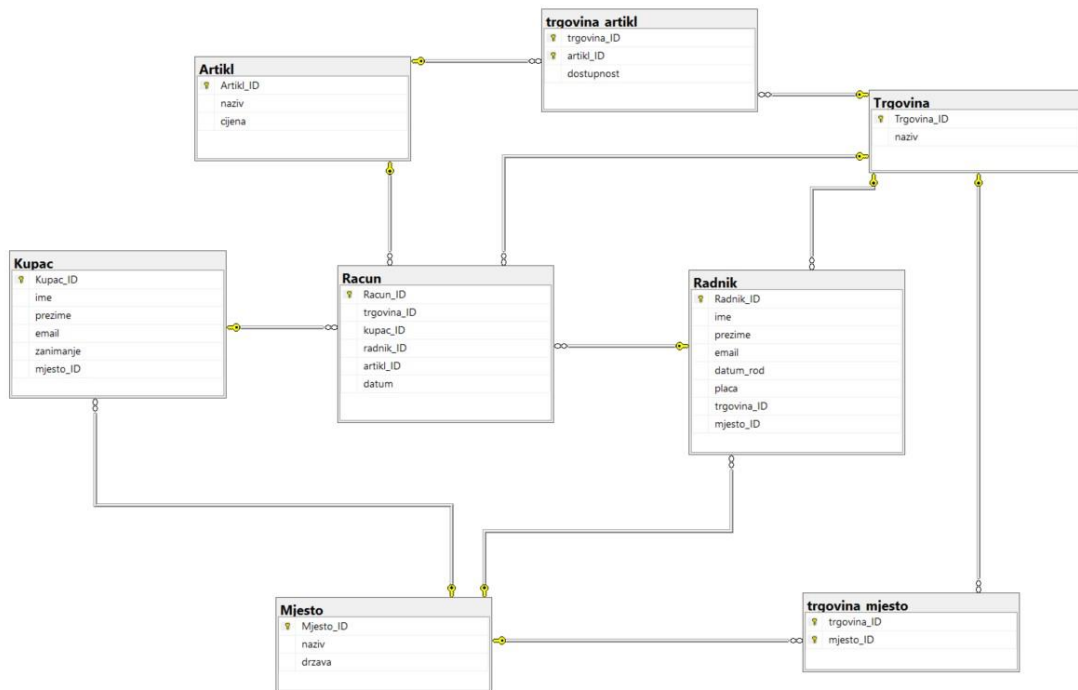
Slika 2 Izgled tablice Kupac MSSQL

kupac	
🔑	kupac_id INT(11)
◇	ime VARCHAR(45)
◇	prezime VARCHAR(45)
◇	email VARCHAR(45)
◇	zanimanje VARCHAR(45)
◇	mjesto_id INT(11)
Indexes ▶	

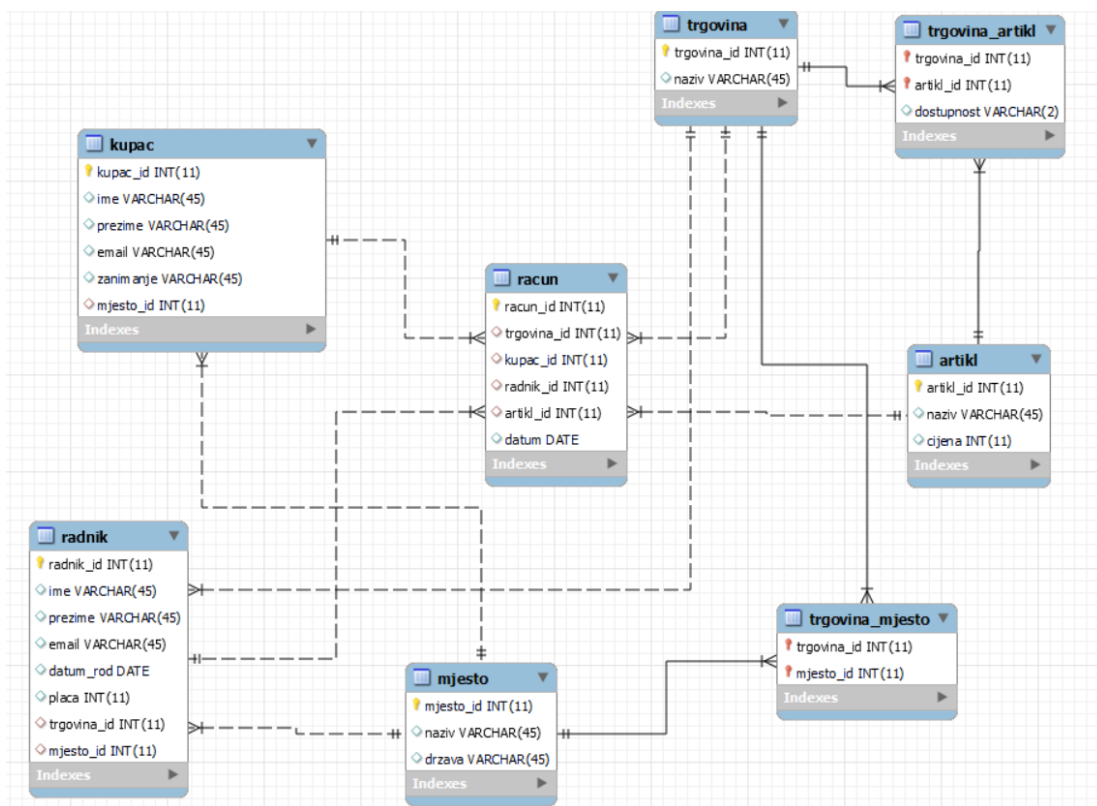
Slika 3 Izgled tablice Kupac MySQL

Na prethodnim slikama i primjeru koda je vidljivo da je način stvaranja tablica isti, ali izgled samih tablica nije, MSSQL sadržava podatke o tome jesu li dopuštene nulte vrijednosti, te je sama tablica dosta preglednija.

Na sljedećim slikama su prikazani dijagrami baza podataka za MSSQL (Slika 4) i MySQL (Slika 5), s pripadajućim tablicama i relacijama kojima su povezane.



Slika 4 Dijagram MSSQL



Slika 5 Dijagram baze podataka MySQL

Dijagrami služe za prikazivanje odnosa među samim tablicama, kako bismo lakše zapisivali upite.

Na prethodnim slikama su vidljivi izgledi dijagrama baza podataka, te je isto tako vidljivo da je MSSQL dijagram pregledniji od MySQL, jer su MySQL dijagramu podatci o tablici i atributima previše nagomilani, dok je u MSSQL dijagramu imamo opciju da to uključimo, odnosno isključimo, te je sve raspoređeno u svoje stupce.

2.2 Generiranje podataka

Što se tiče unosa podataka u bazu, bilo je potrebno unijeti veliku količinu podataka, da bi se baze uopće mogle uspoređivati. Ručni unos velikog broja podataka bi oduzeo previše vremena, stoga je za generiranje podataka korišten generator SQL datoteka pod nazivom *ExtendClass CSV Generator*. Generator nudi razne vrste podataka kao što su ime, prezime, grad, mjesto. Moguće je i izraditi vrstu podataka kombiniranjem dostupnih vrsta, primjer email. Na slici 6 je prikazano sučelje generatora podataka koji se unose u bazu.

Field name	Data type	Setting
<input type="text" value="Radnik_ID"/>	Index	From <input type="text" value="1"/>
<input type="text" value="ime"/>	First name	
<input type="text" value="prezime"/>	Last name	
<input type="text" value="email"/>	Expression	Expression <input type="text" value="{{ime}}.{{prezime}}@yopmail.com"/>
<input type="text" value="datum_rod"/>	Random (Date)	From <input type="text" value="1970-01-01"/> To <input type="text" value="2000-01-01"/> Format <input type="text" value="Y-m-d"/>
<input type="text" value="trgovina_ID"/>	Random (integer)	From <input type="text" value="1"/> To <input type="text" value="20"/>
<input type="text" value="mjesto_ID"/>	Random (integer)	From <input type="text" value="1"/> To <input type="text" value="1000"/>

Format:	<input type="text" value="SQL"/>	# Rows:	<input type="text" value="100000"/>	# Files:	<input type="text" value="1"/>	Filename	<input type="text" value="radnik.sql"/>
Table name	<input type="text" value="Radnik"/>	Include create table statement	<input type="text" value="No"/>				

Slika 6 Generiranje podatka tablice Radnik

Ukupno je u bazu uneseno oko 650 000 podataka. U tablicu Kupac uneseno je 150 000 podataka, Radnik 100 000, Mjesto 100 000, Trgovina-mjesto 97 500, Trgovina-artikl 1235, Artikl 100, Trgovina 20, te Račun 200 000. Nisu obje baze jednakim vremenom unosile podatke, pa je ovdje prikazan prvi dio usporedbe, odnosno usporedba upita unosa.

Na slici 7 je prikazan oblik u kojem su generirani podatke dobiveni, a na slici 8 grafički je prikazano vrijeme potrebno za unos podataka u obje relacijske baze.

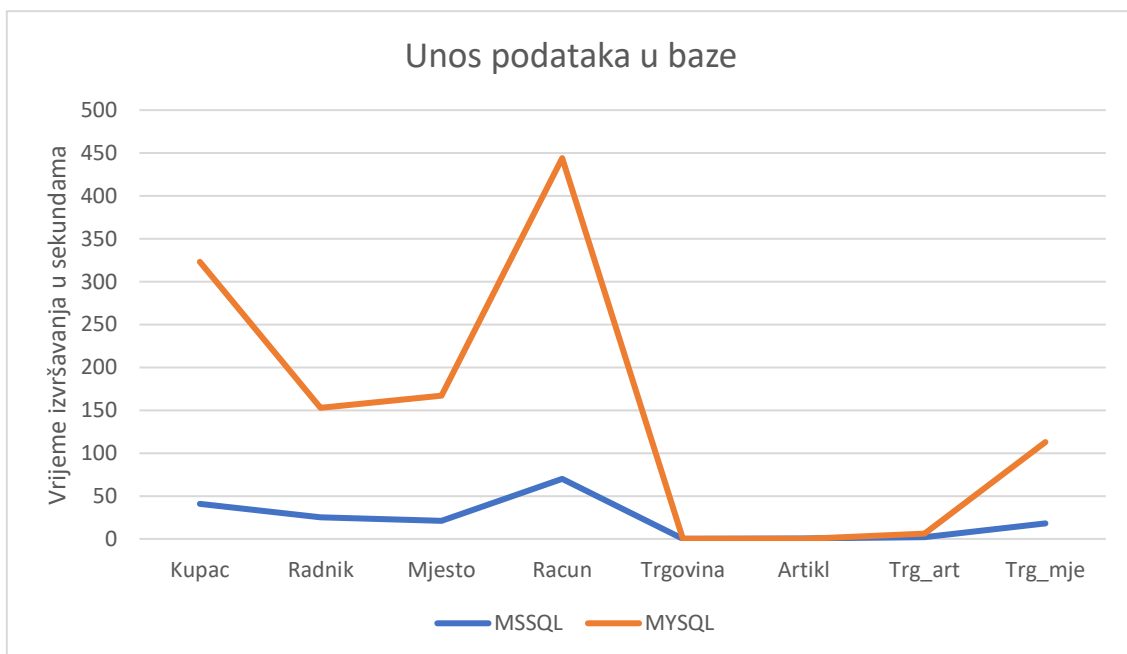
```

Kupac - Blok za pisanje
Datoteka Uređivanje Oblikovanje Prikaz Pomoć
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 1, 'Kaia', 'Tayib', 'Kaia.Tayib@yopmail.com', 'developer', 528
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 2, 'Zia', 'Dash', 'Zia.Dash@yopmail.com', 'firefighter', 93896
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 3, 'Roberta', 'Stefa', 'Roberta.Stefa@yopmail.com', 'worker',
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 4, 'Aili', 'Niccolo', 'Aili.Niccolo@yopmail.com', 'developer'
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 5, 'Shauna', 'Lacomb', 'Shauna.Lacomb@yopmail.com', 'develo
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 6, 'Aubrie', 'Hermes', 'Aubrie.Hermes@yopmail.com', 'police o
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 7, 'Selia', 'Forrer', 'Selia.Forrer@yopmail.com', 'worker', 59
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 8, 'Carly', 'Krystle', 'Carly.Krystle@yopmail.com', 'doctor',
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 9, 'Nyssa', 'Joni', 'Nyssa.Joni@yopmail.com', 'firefighter', 8
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 10, 'Viki', 'Krystle', 'Viki.Krystle@yopmail.com', 'developer
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 11, 'Alisha', 'Shields', 'Alisha.Shields@yopmail.com', 'worke
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 12, 'Vanessa', 'Buckler', 'Vanessa.Buckler@yopmail.com', 'fir
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 13, 'Petronia', 'Standing', 'Petronia.Standing@yopmail.com',
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 14, 'Jany', 'Judy', 'Jany.Judy@yopmail.com', 'police office
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 15, 'Frank', 'Jaylene', 'Frank.Jaylene@yopmail.com', 'firefig
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 16, 'Aaren', 'Edee', 'Aaren.Edee@yopmail.com', 'developer', 91
Insert Into Kupac ( Kupac_ID, ime, prezime, email, zanimanje, mjesto_ID) Values ( 17, 'Emilia', 'Naor', 'Emilia.Naor@yopmail.com', 'developer',

```

Slika 7 Generirani podaci za unos

Generirani podatci dolaze u obliku onoliko upita za unos, koliko smo i zatražili, te se u bazu podataka unose kroz sučelja baza podataka za upite.

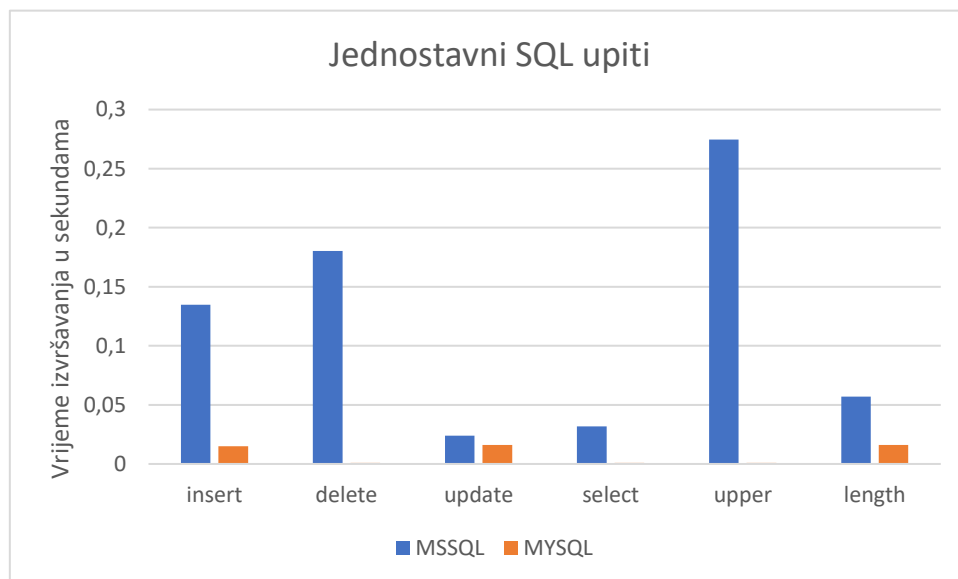


Slika 8 Usporedba upita unosa u sekundama

Na slici 8, je vidljivo da je za tablice koje sadrže veliki broj podataka, bazi podataka izrađenoj u MySQL-u bilo potrebno znatno više vremena, nego bazi izrađenoj u MSSQL-u. Konkretni primjer je vidljiv u tablici Račun, koja sadrži 200 000 podataka. MSSQL baza je obradila unos za 70 sekundi, dok je bazi MySQL bilo potrebno 444 sekunde, što je gotovo sedam puta više vremena. Kod unošenja manjeg broja podataka, razlika nije toliko značajna. Primjer je tablica trgovina-artikl, koja sadrži 1 235 podataka, te su MSSQL-u bile potrebne dvije sekunde, a MySQL-u šest.

Prilikom unošenja podataka u bazu, zanimljivo za uočiti je bilo, da MySQL baza ne može unijeti specijalne znakove, dok MSSQL baza može. Primjer je naziv grada „Gdańsk“, MySQL baza nije mogla unijeti znak „Ń“, dok MSSQL baza je.

Za potrebe usporedbe, izvedeni su i neki jednostavniji upiti, poput unosa, brisanja, uređivanja i odabira samo jednog podatka. Ovi upiti su izvedeni u programima za baze podataka.



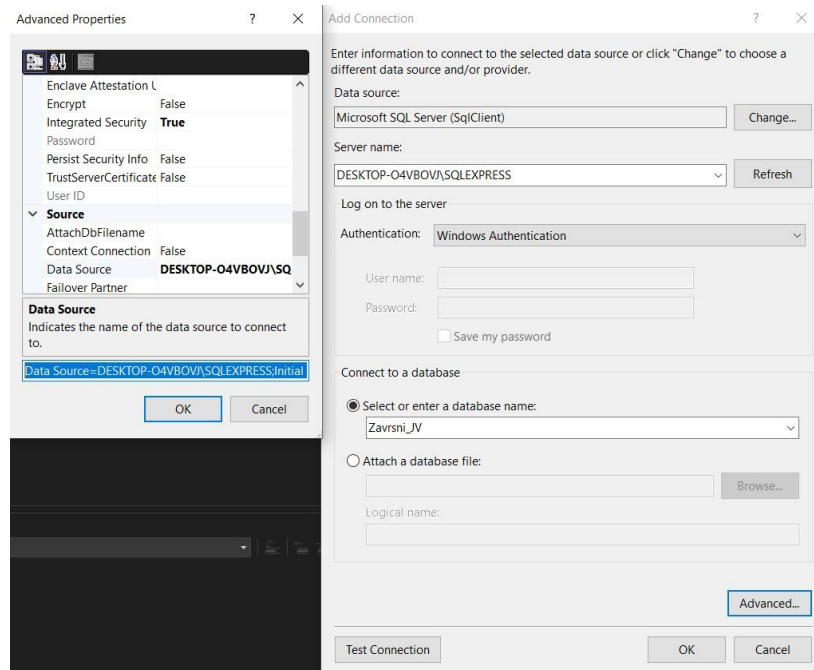
Slika 9 Prikaz vremena izvršavanja osnovnih upita

Na slici 9 je vidljivo da MSSQL baza koristi značajno više vremena za jednostavne upite, od MySQL baze. Razlika se najbolje vidi na primjeru upita odabira jednog podatka, sa svim velikim slovima, gdje je MySQL obradio upit za 0.001 sekundu, a MSSQL za 0.2746 sekundi. Sličnost se pojavljuje kod upita za uređivanje, ali i tu je MSSQL baza sporija. Svaki od ovih upita je ponovljen pet puta, te se ponekad javilo odstupanje do otprilike 0.04 sekunde za MSSQL, te do 0.01 sekunde za MySQL.

Vrijeme izvršavanja ovih jednostavnijih upita i upita unosa mjereno je pomoću programa za izradu baze podataka, koji u sebi imaju štoperice, koje mjere vrijeme u sekundama od trenutka pokretanja izvršavanja upita, sve dok upit ne bude izvršen.

2.3 Test aplikacija

Prilikom izrade testne aplikacije za izvršavanje SQL upita, prvo je bilo potrebno spojiti bazu podataka na Windows Formu pomoću odgovarajuće poveznice(eng. *link*) za spoj. Na slici 10 je prikazano dobivanje poveznice za povezivanje MSSQL baze i aplikacije.



Slika 10 Poveznica za MSSQL

Sljedeći kod prikazuje poveznicu MSSQL baze i aplikacije.

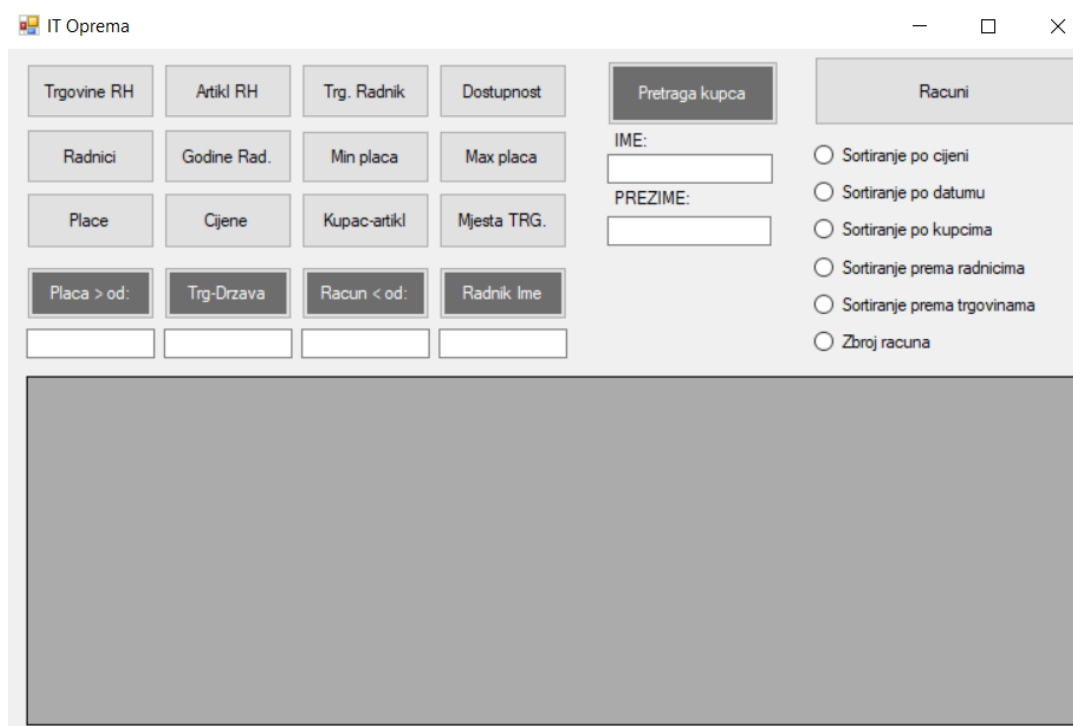
```
@ "Data Source=DESKTOP-O4VBOVJ\SQLEXPRESS;Initial  
Catalog=zavjv;Integrated Security=True"
```

Sljedeći kod prikazuje poveznicu MySQL baze i aplikacije.

```
"SERVER=localhost ;DATABASE=zavjv ;UID=root  
;PASSWORD=lozinka123 ;"
```

Što se poveznica za povezivanje aplikacije s bazom tiče, MySQL baza ima jednostavniji oblik. Jasno se može vidjeti gdje treba unijeti koji podatak, te za njen dohvat ne trebamo otvarati nikakve dijaloške okvire, kao što je bio slučaj sa MSSQL.

Nakon uspješnog povezivanja baze podataka i aplikacije, bilo je potrebno započeti izradom sučelja. Cilj je bio izraditi sučelje koje neće biti prezahtjevno za korisnika, već da odmah pri pokretanju same aplikacije, jasno zna što za što služi. Na svakom gumbu napisan je kratak opis radnje za koju služi, te svaka vrsta gumbova ima svoju boju pozadine, kao što je prikazano na slici 11.



Slika 11 Sučelje aplikacije

Na sučelju je prisutno osamnaest gumbova, od kojih svaki služi za izvršavanje određenog upita, izuzev gumba „Računi“, koji ima dodatne opcije, te ukupno može odraditi šest SQL upita. Pet gumbova dolazi s pripadajućim tekstnim okvirima, u koje se upisuju određene vrijednosti, prema kojima se upiti izvršavaju. U slučaju neke pogreške, javlja se okvir poruke, koji ukazuje zbog čega se pogreška dogodila, te što je potrebno ispraviti. Također je prisutna i štoperica u aplikaciji, koja mjeri vrijeme od početka, do kraja izvršavanja upita u milisekundama. Primjer upita pretrage po imenu radnika je prikazan na slici 12.

IT Oprema

Trgovine RH Artikli RH Trg. Radnik Dostupnost

Radnici Godine Rad. Min placa Max placa

Place Cijene Kupac-artikl Mjesta TRG.

Placa > od: Trg-Drzava Racun < od: Radnik Ime

IME:

PREZIME:

Pretraga kupca Racuni

- Sortiranje po cijeni
- Sortiranje po datumu
- Sortiranje po kupcima
- Sortiranje prema radnicima
- Sortiranje prema trgovinama
- Zbroj racuna

	IME	Prezime	Trgovina
▶	Di	Cressida	Sony
	Di	Tamsky	Sony
	Di	Donell	Sony
	Di	Wildemuth	Sony
	Di	Hurley	Sony
	Di	Dorothy	Samsung
	Di	Couture	Samsung
	Di	Modie	Samsung
	Di	Noelyn	Samsung

30 ms

U redu

Slika 12 Primjer upita pretrage Radnika

Na slici 12 je vidljiva obrada upita, koji od 100 000 radnika izabire samo one koji imaju ime jednako unesenom, u ovom slučaju „Di“. Kao rezultat upita u tablici su prikazana imena i prezimena radnika, te je prikazan naziv pripadajuće trgovine za svakog radnika. Na slici je prisutno i vrijeme izvršavanja upita, u ovom slučaju 30 mili-sekundi.

2.4 Upiti kroz aplikaciju

Kao što je navedeno u prethodnom poglavlju, upiti u aplikaciji se izvode pritiskom na željene gumbе, te oni u okviru baze podataka prikazuju podatke traženog upita, te se vrše mjerenje od početka izvršavanja upita, do filtriranja i prikaza podataka u milisekundama.

SQL upiti se u nekim slučajevima razlikuju kod njihove implementacije . Za primjer imamo upit.

```
"Select ime, prezime, YEAR(GETDATE())-YEAR(datum_rod) as Godine
from Radnik order by Godine" .
```

Ovim upitom se dohvaća ime i prezime radnika, te njihove godine. Pomoću metode „GETDATE“ se dohvaća trenutni datum, te mu oduzimamo datum rođenja, međutim MySQL baza ne podržava tu metodu, pa je umjesto te metode, upisana metoda „Now“, koju MySQL podržava.

Upiti agregatnih funkcija također se razlikuju, ali ovaj put u obliku podataka koji ispisuju. Primjer je agregatna funkcija koja računa prosjek.

```
"Select t.naziv as Trgovina,AVG(r.placa) as Prosjecna_Placa
from Radnik r inner join Trgovina t on
t.Trgovina_ID=r.trgovina_ID GROUP by t.naziv" .
```

Ovaj upit nam daje podatke o trgovinama i prosječnim plaćama radnika koji rade u njima. MSSQL baza ispisuje plaće u obliku cijelih brojeva, dok MySQL baza ispisuje decimalne brojeve s četiri decimale.

Upiti ,koji se izvršavaju kroz aplikaciju, se uglavnom tiču potreba odabira, sortiranja i grupiranja podataka. Primjeri za upite grupiranje, sortiranja i odabira, kao i neke agregatne funkcije.

Kod 2 Primjeri testiranih upita

```
"Select * from Radnik r inner join Trgovina t on  
t.Trgovina_ID=r.trgovina_ID inner join Mjesto m on  
m.Mjesto_ID=r.mjesto_ID where r.placa>@placa"
```

```
"Select t.naziv as Prodavaonica,k.ime+' '+k.prezime as  
Kupac,ra.ime+' '+ra.prezime as Radnik,a.cijena as  
Cijena,a.naziv as Proizvod,r.datum from Racun r inner join  
Kupac k on k.Kupac_ID=r.kupac_ID inner join Artikl a on  
a.Artikl_ID=r.artikl_ID inner join Radnik ra on  
ra.Radnik_ID=r.radnik_ID inner join Trgovina t on  
t.Trgovina_ID=r.trgovina_ID where a.cijena<@racuni order by  
a.cijena desc"
```

```
"Select MAX(placa) from Radnik"
```

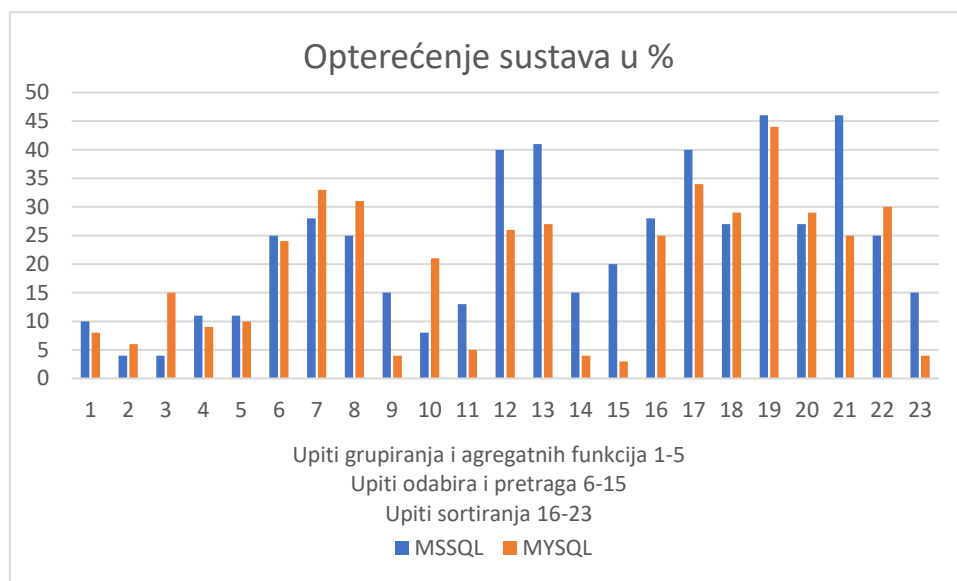
```
"Select t.naziv as Trgovina,AVG(r.placa) as Prosjecna_Placa  
from Radnik r inner join Trgovina t on  
t.Trgovina_ID=r.trgovina_ID GROUP by t.naziv"
```

Prvi upit se odnosi na isključivo upit odabira, u njemu je prikazan dohvat podataka iz tablice Radnik, koja se „*inner join*“ operacijom povezivanja tablica, povezuje s tablicama Trgovina i Mjesto, te se prikazuju podatci samo za one radnike koji imaju isključivo veću plaću od one koja je unesena u pripadajući tekstni okvir. Drugi upit je malo složeniji, a on se odnosi na upit u kojem se odabiru podatci iz tablice Račun, te se povezuju s tablicama Kupac, Radnik, Trgovina i Artikl, a na samom kraju podatci se sortiraju prema cijeni silazno. Treći upit je upit agregatne funkcije, u kojem se odabire najveća plaća radnika. Konačno, četvrti primjer je upit grupiranja, gdje se trgovine grupiraju sa prosječnim plaćama radnika u njima, prema nazivu trgovine.

2.5 Usporedba rezultata

U ovom poglavlju su prikazani rezultati usporedbe izvršavanja upita na ovih dviju vrsta relacijskih baza podataka. Način analize rezultata bilo je uspoređivanje vremena izvršavanja upita u aplikaciji, te mjerenja opterećenja sustava, u vidu postotka korištenja središnje upravljačke jedinice (eng. *Central Processing Unit*).

Mjerenja za opterećenje sustava su izvršena u Visual Studiu pomoću *debugger*-a, alata koji služi za analizu opterećenja sustava prilikom izvođenja aplikacije. Mjerenja vremena su odrađena pomoću ugrađene metode „Stopwatch“ u Visual Studio, za mjerenje vremena.



Slika 13 Usporedba opterećenja sustava

Na slici 13 stupčastim grafikonom je prikazano opterećenje sustava kod izvršavanja upita na obje baze. Na slici je vidljivo, da je od ukupno dvadeset tri izvršena upita, MSSQL baza napravila veće opterećenje procesora u 15 slučajeva, što je skoro dva puta više od slučajeva, gdje MySQL pravi veće opterećenje.

	Trgovina	Artikl	dostupnost
▶	Sony	Skener	DA
	Sony	Vanjski tvrdi disk	DA
	Sony	USB	DA
	Sony	Zvučnici	DA
	Sony	Procesor (CPU)	DA
	Sony	RAM memorija	DA

	Trgovina	Artikl	dostupnost
▶	Samsung	Graficki procesor	DA
	Huawei	Racunalna svjetla	DA
	Instar	Bezicni mis i tipko...	DA
	Intel	Mis	DA
	Canon	3D skener	DA
	Links	Slusalice	DA
	Links	Extrema zvučna k...	DA

Slika 14 Usporedba upita sortiranja

Na slici 14 prikazana je usporedba sortiranja trgovine, artikla i dostupnosti, prema dostupnosti artikala, a kod 3 prikazuje sam kod upita.

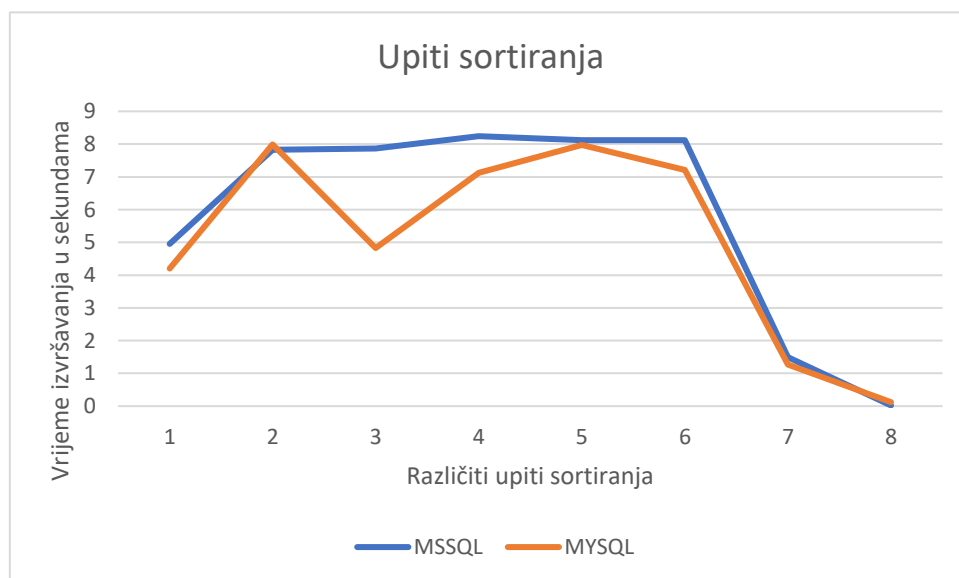
Kod 3 Upit sortiranja trgovina i artikala po dostupnosti

```

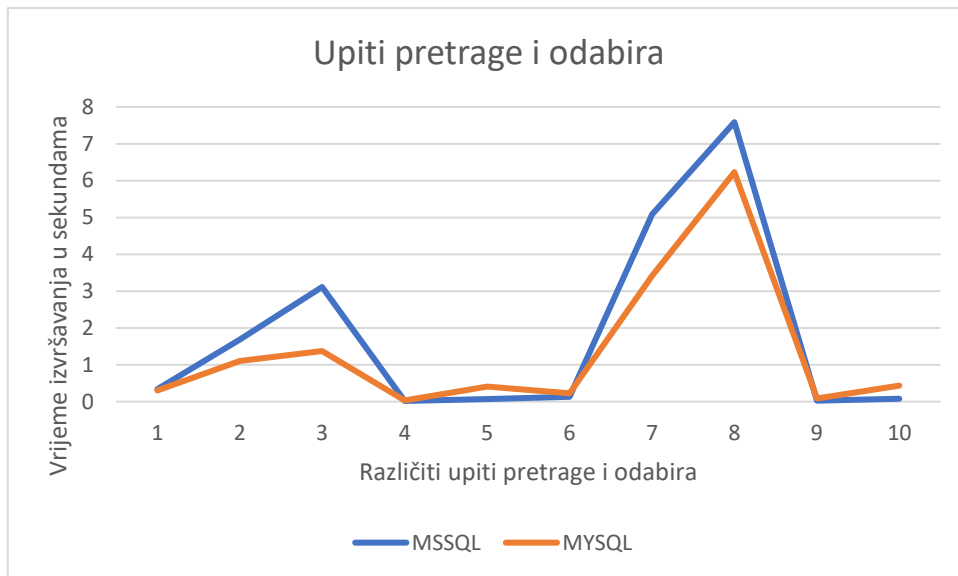
Select t.naziv as Trgovina,a.naziv as Artikl,ta.dostupnost from
trgovina_artikl ta inner join Trgovina t on
t.Trgovina_ID=ta.trgovina_ID inner join Artikl a on
a.Artikl_ID=ta.artikl_ID order by dostupnost

```

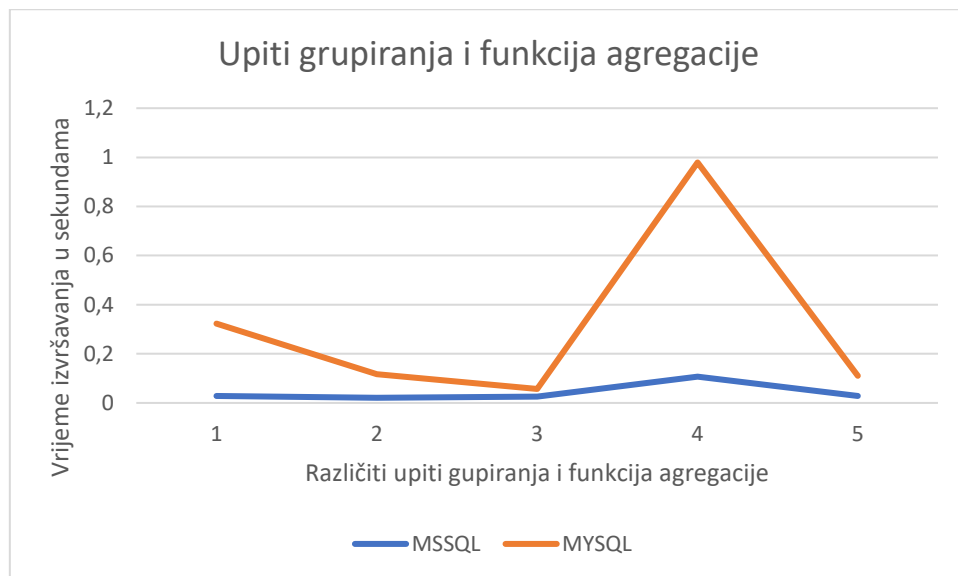
Prva tablica na slici 14 se odnosi na MSSQL, te vidljivo da podaci ne dolaze u istom redoslijedu, kao kod MySQL. Na slikama 15, 16 i 17 grafički je prikazano vrijeme utrošeno za izvršavanje složenijih SQL upita.



Slika 15 Upiti sortiranja u sekundama



Slika 16 Upiti pretrage i odabira u sekundama



Slika 17 Upiti grupiranja i agregatnih funkcija u sekundama

Analiziravši rezultate iz grafikona, dobivene mjerenjem vremena izvršavanja upita u aplikaciji, vidljivo je da MySQL baza podataka utroši manje vremena za upite koji se tiču sortiranja, te pretrage i odabira podataka. Kada je u pitanju manji broj podataka koje je potrebno filtrirati, rezultati su slični, te je više-manje razlika neprimjetna, dok je kod izvršavanja upita s većom količinom podataka, razlika veća. Baza podataka izrađena u MSSQL-u ima prednost u brzini izvršavanja kada su u pitanju upiti grupiranja i upiti koji sadrže agregatne funkcije.

Razlog sporijeg izvršavanja upita sortiranja, pretrage i odabira, te većeg opterećenja procesora, proizlazi iz toga što dolazi do previše povezivanja različitih tablica, odnosno prevelikog broja podataka koje je potrebno filtrirati, te MSSQL baza sporije obrađuje takve skupove podataka, koji proizlaze iz više tablica. S druge strane, kada se radilo o upitima grupiranja i agregatnih funkcija, gdje je bilo prisutno najviše povezivanje dviju tablica, MSSQL baza je bila brža od MySQL baze.

			Opterećenje sustava u %		Vrijeme izvršavanja u sekundama	
			MSSQL	MYSQL	MSSQL	MYSQL
1	Grupiranje i agregatne funkcije	1. Izračun broja radnika grupirano prema trgovinama	10	8	0,029	0,323
2		2. Izračun maksimalne plaće	4	6	0,021	0,117
3		3. Izračun minimalne plaće	4	15	0,026	0,057
4		4. Izračun prosječne plaće grupirano po trgovinama	11	9	0,107	0,979
5		5. Izračun ukupnog broja svih racuna	11	10	0,028	0,11
6	Upiti odabira i pretrage	1. Odabir artikla U RH	25	24	0,344	0,305
7		2. Odabir mjesta i trgovina	28	33	1,688	1,108
8		3. Odabir svih radnika	25	31	3,114	1,373
9		4. Odabir trgovina i artikla te njihovih cijena	15	4	0,02	0,034
10		5. Odabir trgovina U RH	8	21	0,069	0,414
11		6. Pronalazak kupca po imenu i prezimenu	13	5	0,13	0,227
12		7. Pronalazak podataka gdje je placa veća od unesene	40	26	5,088	3,415
13		8. Pronalazak podataka gdje je racun manji od unesenog	41	27	7,586	6,231
14		9. Pronalazak radnika po imenu	15	4	0,024	0,091
15		10. Pronalazak trgovina u drzavi	20	3	0,08	0,44
16	Upiti sortiranja	1. Sortiranje kupaca i artikala po imenu kupca	28	25	4,954	4,198
17		2. Sortiranje racuna po cijeni	40	34	7,827	7,99
18		3. Sortiranje racuna po datumu	27	29	7,862	4,823
19		4. Sortiranje racuna po kupcima	46	44	8,243	7,124
20		5. Sortiranje racuna prema radnicima	27	29	8,12	7,974
21		6. Sortiranje racuna prema trgovinama	46	25	8,118	7,212
22		7. Sortiranje radnika po godinama starosti	25	30	1,497	1,271
23		8. Sortiranje trgovine i artikla uzlazno po dostupnosti	15	4	0,021	0,12

Slika 18 Tablični prikaz opterećenja sustava i vremena izvršavanja

Na slici 18 tablično je prikazan postotak opterećenja sustava, te vrijeme izvršavanja upita za obje baze u sekundama.

ZAKLJUČAK

Analizirajući ove dvije baze, te njihovo izvođenje upita na više od 650 tisuća podataka, dolazimo do zaključka da svaka baza u određenom aspektu, ima svoje prednosti i mane.

Primjerice, MSSQL baza podataka je utrošila ukupno 177,6 sekundi vremena za unos svih 650000 podataka, dok je bazi izrađenoj u MySQL-u bilo potrebno čak 1206,3 sekundi, što je ogromna razlika.

Izrada baza podataka u oba programa je slična, ali , MSSQL baza nudi nešto jednostavniji i pregledniji prikaz, od baze u MySQL. Povezivanje obje baze s aplikacijom je vrlo jednostavno.

Što se tiče jednostavnih upita unosa, brisanja, ažuriranja i slično, MySQL baza ima značajnu prednost u vremenu izvršavanja, te isto tako pravi manje opterećenje sustava nego baza izrađena u MSSQL, prilikom izvođenja složenijih upita. Također, MySQL koristi kraće vrijeme izvođenja složenijih upita, kada se radi o velikoj količini podataka, dok kod manje količine, vrijeme je slično.

MSSQL baza ispisuje podatke drugačijim, odnosno preglednijim redoslijedom, nego MySQL.

Upiti koje MySQL baza brže obrađuje, su upiti sortiranja, te upiti pretrage i odabira, gdje je od osamnaest mogućih upita, MySQL baza bila brža u trinaest. Kada su u pitanju upiti agregatnih funkcija i grupiranja, baza MSSQL je u svim slučajevima utrošila manje vremena od MySQL baze.

Za kraj možemo reći da učinkovitost i brzina određene relacijske baze podataka ovisi o korisničkim zahtjevima i potrebama, te o broju podataka koji se obrađuje.

LITERATURA

- [1] Microsoft SQL Server, Wikipedia, 23. 08. 2023.
https://en.wikipedia.org/wiki/Microsoft_SQL_Server
- [2] MySQL Workbench, MySQL, 03. 09. 2023.
<https://www.mysql.com/products/workbench/>
- [3] Visual Studio, Wikipedia, 23. 08. 2023. https://en.wikipedia.org/wiki/Visual_Studio
- [4] Normalizacija baza podataka, Wikipedia, 01. 09. 2023
[.https://en.wikipedia.org/wiki/Database_normalization](https://en.wikipedia.org/wiki/Database_normalization)
- [5] Osnove baza podataka, Microsoft, 26. 08. 2023. <https://support.microsoft.com/hr-hr/office/osnove-baza-podataka-a849ac16-07c7-4a31-9948-3c8c94a7c204>
- [6] Početak rada s bazama podataka, Microsoft , 23. 08. 2023.
<https://support.microsoft.com/hr-hr/office/videozapis-po%C4%8Detak-rada-s-bazama-podataka-457013e7-f75d-48a9-bc8a-4b816436a5a0>
- [7] Baze podataka, Loomen, 24. 08. 2023. <https://arhiva-2021.loomen.carnet.hr/mod/book/view.php?id=1207037&chapterid=173094>
- [8] Relacijske baze podataka prednosti i nedostatci, Database Town, 13. 09. 2023.
<https://databasetown.com/relational-database-benefits-and-limitations/>
- [9] SQL optimizacija upita, DbForgeTeam, Devart, 12. 09. 2023.
<https://blog.devart.com/how-to-optimize-sql-query.html>

POPIS SLIKA

Slika 1 Skica modela baze podataka	7
Slika 2 Izgled tablice Kupac MSSQL	10
Slika 3 Izgled tablice Kupac MySQL	10
Slika 4 Dijagram MSSQL	11
Slika 5 Dijagram baze podataka MySQL	11
Slika 6 Generiranje podatka tablice Radnik	12
Slika 7 Generirani podaci za unos	13
Slika 8 Usporedba upita unosa u sekundama	13
Slika 9 Prikaz vremena izvršavanja osnovnih upita	14
Slika 10 Poveznica za MSSQL	15
Slika 11 Sučelje aplikacije	16
Slika 12 Primjer upita pretrage Radnika	17
Slika 13 Usporedba opterećenja sustava	20
Slika 14 Usporedba upita sortiranja	21
Slika 15 Upiti sortiranja u sekundama	21
Slika 16 Upiti pretrage i odabira u sekundama	22
Slika 17 Upiti grupiranja i agregatnih funkcija u sekundama	22
Slika 18 Tablični prikaz opterećenja sustava i vremena izvršavanja	23

Skraćenice

MSSQL	<i>Microsoft SQL Server Managment Studio</i>	Ime programa za izradu baza podataka
MySQL	<i>My SQL Workbench</i>	Ime programa za izradu baza podataka
SQL	<i>Structured query language</i>	Ime programskog jezika za baze podataka
C#	<i>C Sharp</i>	Naziv programskog jezika
AVG	<i>Average</i>	Prosjek
MIN	<i>Minimum</i>	Najmanje
MAX	<i>Maximum</i>	Najviše
CSV	<i>Comma-separated values</i>	Vrijednosti odvojene zarezom
INT	<i>Integer</i>	Cijeli broj
ID	<i>Identifier</i>	Jedinstvena oznaka
VARCHAR	<i>variable character</i>	Tekstni zapis
IT	<i>Information technology</i>	Informacijska tehnologija