

Analiza Indeksa potrošačkih cijena (CPI) korištenjem MariaDB baze podataka: Teorijski i praktični pristup

Škriljevečki, Velimir

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:088344>

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

Download date / Datum preuzimanja: **2025-03-01**



Repository / Repozitorij:

[University North Digital Repository](#)





**Sveučilište
Sjever**

Završni rad br. 544/EL/2024

**Analiza Indeksa potrošačkih cijena (CPI) korištenjem
MariaDB baze podataka: Teorijski i praktični pristup**

Velimir Škriljevečki, 0514/601

Varaždin, rujan 2024. godine



Sveučilište Sjever

Odjel za elektrotehniku

Završni rad br. 544/EL/2024

Analiza Indeksa potrošačkih cijena (CPI) korištenjem MariaDB baze podataka: Teorijski i praktični pristup

Student

Velimir Škriljevečki, 0514/601

Mentor

Izv. prof. dr. sc. Ladislav Havaš, dipl. ing. el.

Varaždin, rujan 2024. godine

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL	Odjel za elektrotehniku		
STUDIJ	Stručni prijediplomski studij Elektrotehnika		
PRISTUPNIK	Velimir Škriļjevečki	MATIČNI BROJ	0514/601
DATUM	05.09.2024.	KOLEGIJ	Baze podataka i SQL
NASLOV RADA	Analiza Indeksa potrošačkih cijena (CPI) korištenjem MariaDB baze podataka: Teorijski i praktični pristup		
NASLOV RADA NA ENGL. JEZIKU	Analysis of the Consumer Price Index (CPI) using the MariaDB database: A theoretical and practical approach		
MENTOR	Ladislav Havaš	ZVANJE	Izvanredni profesor
ČLANOVI POVJERENSTVA	<ol style="list-style-type: none">doc. dr. sc. Tomislav Horvat - predsjednik povjerenstvadoc. dr. sc. Dunja Srpak - članizv. prof. dr. sc. Ladislav Havaš - član / mentormr. sc. Ivan Šumiga, viši predavač - zamjenski član		

Zadatak završnog rada

BROJ	544/EL/2024
OPIS	<p>Indeks potrošačkih cijena (CPI – engl. Consumer Price Index) predstavlja jedan od ključnih ekonomskih pokazatelja koji prati promjene cijena dobara i usluga koje domaćinstva kupuju za osobnu potrošnju. Kao mjera inflacije i troškova života, CPI ima značajan utjecaj na donošenje ekonomskih odluka, planiranje politika i analize tržišta.</p> <p>Cilj ovog završnog rada je primijeniti metode analize podataka kako bi se istražile promjene u CPI-u, identificirali ključni trendovi, te razumjeli utjecaj različitih faktora na kretanje cijena. Za analizu promjena CPI_a, koristiti će se relacijski model podataka. Za potrebe ovog rada odabrana je MariaDB, moderna relacijska platforma otvorenog koda. U radu je potrebno usporediti relacijske i hibridne (NoSQL) modele podataka, objasniti zašto se u ovom radu koristi relacijski model u analizi potrošačkih cijena. Potrebno je kreirati konceptualni model (ER) i stvoriti prototip informacijskog sustava za analizu CPI_a.</p> <p>Sukladno navedenom, u tekstualnom dijelu završnog rada treba kreirati odgovarajući sadržaj te napisati cjeloviti tekst završnog rada.</p>

ZADATAK URUČEN

06.09.2024.



OPIS MENTORA

[Signature]

Sažetak

Ovaj završni rad se bavio analizom indeksa potrošačkih cijena (CPI) koristeći relacijski model podataka. Cilj rada je bio primijeniti metode analize podataka kako bi se istražile promjene u CPI-u, identificirali ključni trendovi i razumjeli utjecaj različitih faktora na kretanje cijena.

Za pohranu i obradu podataka, korištena je relacijska baza podataka zbog svoje otvorenosti, fleksibilnosti i kompatibilnosti s raznim analitičkim alatima. Podaci o CPI-u prikupljeni su iz relevantnih statističkih izvora, a zatim uvezeni u MariaDB bazu gdje su organizirani u relacijske tablice. Upotrebom SQL upita, provedena je analiza vremenskih serija, komparativna analiza između različitih kategorija proizvoda, te evaluacija sezonskih utjecaja na cijene.

MariaDB se pokazala kao pouzdana i efikasna platforma za ovakve analize, omogućujući jednostavan pristup velikim skupovima podataka i fleksibilne mogućnosti manipulacije i analize podataka. Rad je naglasio važnost upotrebe otvorenih tehnologija u ekonomskoj analizi, što može doprinijeti većoj dostupnosti i transparentnosti podataka u istraživačkoj zajednici.

Ključne riječi:

- CPI
- ER model
- MariaDB
- MySQL
- Relacijski model podataka
- SQL

Abstract

This thesis dealt with the analysis of the Consumer Price Index (CPI) using a relational data model. The aim of the paper was to apply data analysis methods to investigate changes in the CPI, identify key trends and understand the impact of various factors on price movements.

For data storage and processing, a relational database was used due to its openness, flexibility and compatibility with various analytical tools. Data on the CPI were collected from relevant statistical sources, and then imported into the MariaDB database, where they were organized into relational tables. Using SQL queries, time series analysis, comparative analysis between different product categories, and evaluation of seasonal effects on prices were performed.

MariaDB has proven to be a reliable and efficient platform for such analyses, enabling easy access to large data sets and flexible options for data manipulation and analysis. The paper emphasized the importance of using open technologies in economic analysis, which can contribute to greater availability and transparency of data in the research community.

Keywords:

- CPI
- ER model
- MariaDB
- MySQL
- Relational data model
- SQL

Popis korištenih kratica

ACID	Atomicity, Consistency, Isolation, Durability (Atomičnost, Konzistentnost, Izolacija, Durabilnost)
API	Application Programming interface (aplikacijsko programsko sučelje)
CMS	Content Management System (sustav za upravljanje sadržajem)
CPI	Consumer Price Indeks (Indeks potrošačkih cijena)
ER	Entity-Relationship
NOSQL	Not Only Structured Query Language (ne samo strukturni upitni jezik)
RDBMS	Relational Database Management System (sustav upravljanja relacijskom bazom)
SQL	Structured Query Language (strukturni upitni jezik)

Sadržaj

1.	Uvod.....	1
2.	Indeks potrošačkih cijena.....	2
2.1.	Metoda izračuna CPI-a.....	2
2.2.	Primjena CPI-a u ekonomskim analizama.....	2
3.	Modeli podataka.....	3
4.	Relacijske i nerelacijske (NoSQL) baze podataka.....	6
4.1.	Prednosti i nedostaci relacijskih i nerelacijskih baza podataka.....	7
4.2.	Vrste relacijskih baza podataka.....	8
4.3.	Vrste nerelacijskih baza podataka.....	9
5.	Usporedba MariaDB i MySQL-a.....	13
5.1.	MariaDB.....	14
5.2.	MySQL.....	15
6.	Odabir MariaDB-a za analizu cijena.....	18
6.1.	Razlozi odabira MariaDB-a.....	18
6.2.	Primjeri uspješnih primjena MariaDB-a.....	19
7.	Primjena MariaDB-a.....	22
7.1.	Način preuzimanja i instaliranja MariaDB baze podataka.....	24
7.2.	Priprema podataka.....	27
7.3.	Izvori podataka.....	29
7.4.	Izrada konceptualnog modela.....	31
7.5.	Unos podataka u MariaDB.....	33
7.6.	Analiza cijena pomoću MariaDB-a.....	35
7.7.	Prikaz i interpretacija rezultata.....	38
7.8.	Rezultati i diskusija.....	47
8.	Zaključak.....	48
9.	Literatura.....	49

1. Uvod

Indeks potrošačkih cijena (CPI – engl. *Consumer Price Index*) predstavlja jedan od ključnih ekonomskih pokazatelja koji prati promjene cijena dobara i usluga koje domaćinstva kupuju za osobnu potrošnju. Kao mjera inflacije i troškova života, CPI ima značajan utjecaj na donošenje ekonomskih odluka, planiranje politika i analize tržišta. S obzirom na rastuću važnost baza podataka u suvremenom poslovanju i istraživanju, ovaj rad će se usredotočiti na teorijsku i praktičnu primjenu MariaDB baze podataka u analizi cijena.

Cilj rada je pružiti teorijski pregled CPI indeksa, razjasniti razlike između MariaDB i MySQL-a, te istaknuti prednosti korištenja MariaDB-a, posebno u kontekstu analize cijena. MariaDB, kao relativno nova i inovativna baza podataka, nudi brojne prednosti, uključujući poboljšane performanse i skalabilnost.

U drugom poglavlju bit će objašnjeno što je indeks potrošačkih cijena te metoda njegova izračuna a u trećem poglavlju će biti obrađeni modeli podataka. Relacijske i nerelacijske baze podataka te njihove prednosti i nedostaci bit će objašnjeni u četvrtom poglavlju. U petom poglavlju napraviti će se usporedba Maria DB-a i MySQL-a dok će se u praktičnom dijelu rada, odn. u šestom i sedmom poglavlju, provesti analiza cijena pomoću MariaDB-a demonstrirajući njenu efikasnost i praktičnost. Metodološki, rad će kombinirati teorijska istraživanja s praktičnim primjerima, čime će se dobiti cjelovit uvid u mogućnosti MariaDB baze podataka za analizu ekonomskih podataka kao što je CPI. Cijene koje će biti prikazane u ovom radu bazirane su na trendovima koji su bili aktualni u prvoj polovici 2024. godine u Hrvatskoj. Ovaj pristup omogućit će da se dobije sveobuhvatan pregled teme, istovremeno naglašavajući relevantnost i primjenjivost MariaDB-a u stvarnom svijetu.

2. Indeks potrošačkih cijena

Indeks potrošačkih cijena (CPI) je mjera koja prati promjene u cijenama košarice dobara i usluga koje domaćinstva kupuju za osobnu potrošnju. Kao jedan od najvažnijih ekonomskih pokazatelja, CPI se koristi za procjenu inflacije i promjena u troškovima života.

CPI mjeri prosječne promjene u cijenama koje potrošači plaćaju za određeni skup proizvoda i usluga kroz određeni vremenski period. Ova mjera je ključna za razumijevanje inflacijskih trendova, jer pomaže u određivanju koliko se cijene povećavaju ili smanjuju tijekom vremena. Također, CPI ima značajan utjecaj na ekonomsku politiku, jer centralne banke i vlade koriste ovaj indeks za donošenje odluka o kamatnim stopama, socijalnim programima i porezima.

2.1. Metoda izračuna CPI-a

Izračunavanje CPI-a uključuje nekoliko koraka. Prvo, definira se košarica dobara i usluga koja odražava prosječnu potrošnju domaćinstava. Zatim se prate cijene tih dobara i usluga u različitim vremenskim razdobljima. Promjene cijena se ponderiraju prema važnosti svakog proizvoda ili usluge u ukupnoj potrošnji. Konačno, CPI se izračunava kao indeksni broj koji pokazuje relativne promjene u cijeni košarice u odnosu na osnovnu godinu.

2.2. Primjena CPI-a u ekonomskim analizama

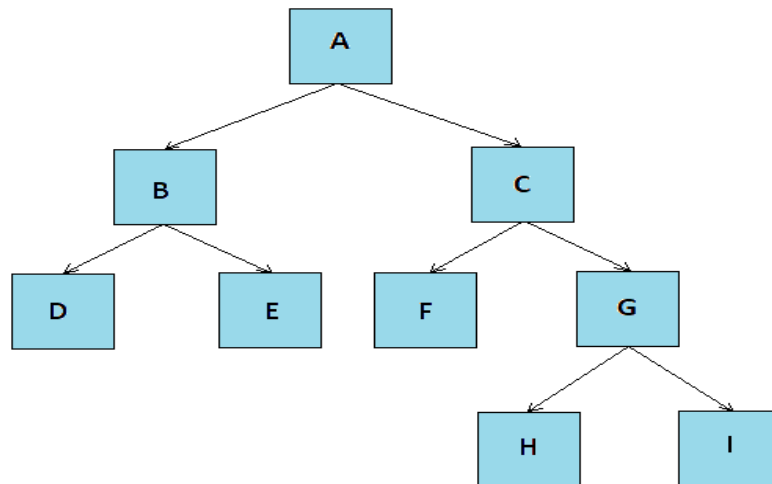
CPI se koristi u brojnim ekonomskim analizama i istraživanjima. Ekonomski analitičari koriste CPI za praćenje inflacijskih trendova i predviđanje budućih kretanja cijena. Također, CPI je ključan za prilagođavanje plaća, mirovina i socijalnih naknada inflaciji, osiguravajući da se stvarna vrijednost tih isplata održi tijekom vremena. Poslovni subjekti koriste CPI za donošenje odluka o cijenama proizvoda i usluga, dok investitori koriste ovaj indeks za ocjenu realnog povrata ulaganja.[1]

3. Modeli podataka

Modeli podataka su strukture koje određuju kako će podaci biti pohranjeni, organizirani, upravljani i pristupani u sustavima za upravljanje bazama podataka. Oni predstavljaju različite pristupe za strukturiranje i organizaciju podataka.[2]

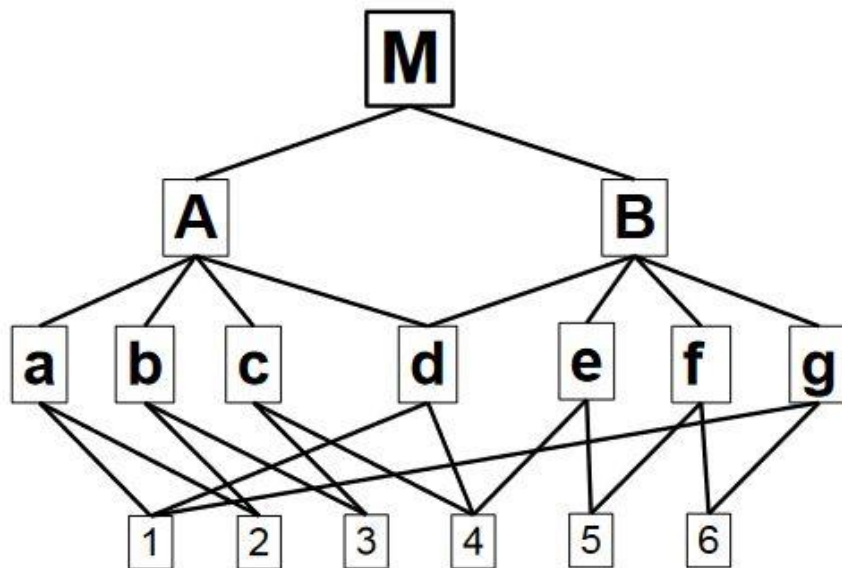
Glavni modeli podataka uključuju:

1. **Hijerarhijski model:** organizira podatke u strukturu stabla gdje svaki čvor (entitet) ima jedan nadređeni čvor, osim korijenskog čvora kao što je prikazano na slici 3.1.
 - Podaci su organizirani u hijerarhijsku strukturu – struktura stabla.
 - Svaki čvor ima jedan nadređeni čvor i može imati više podređenih čvorova – odnos roditelj-dijete.



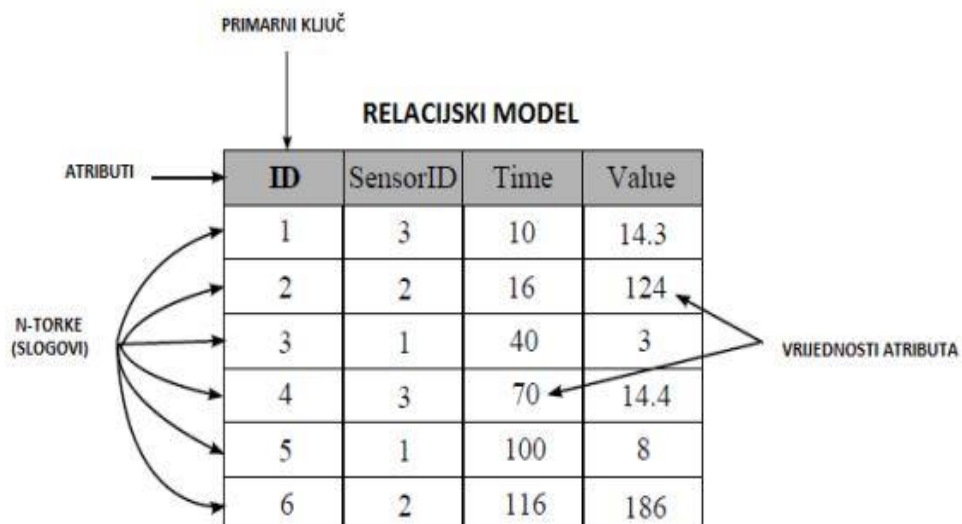
Slika 3.1-Primjer hijerarhijskog modela baze podataka

2. **Mrežni model:** proširuje hijerarhijski model dopuštajući čvorovima da imaju više roditelja, što omogućuje složenije odnose među podacima kao što je prikazano na slici 3.2.
 - Podaci su organizirani u obliku grafa – graf struktura.
 - Čvorovi mogu imati više roditeljskih čvorova – mnogostruki roditelj.



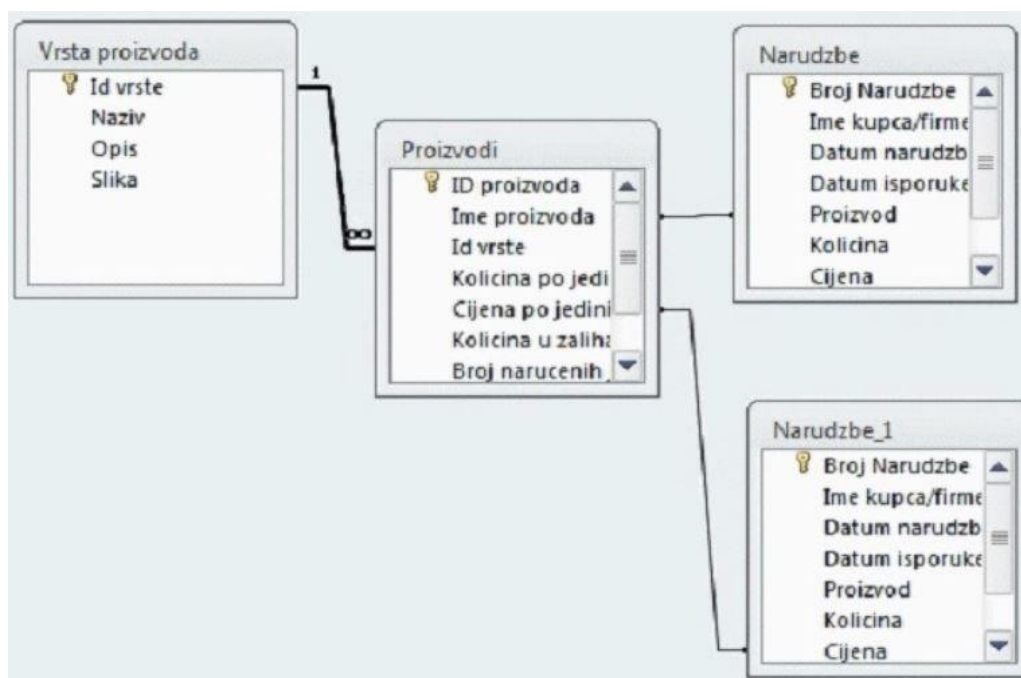
Slika 3.2 -Primjer mrežnog modela baze podataka [3]

3. **Relacijski model:** organizira podatke u tablice (relacije) koje se sastoje od redaka (zapisa) i stupaca (atributa). Svaka tablica ima jedinstveni ključ (primarni ključ) koji identificira svaki redak. Primjer takvog modela je prikazan na slici 3.3.
- Podaci su pohranjeni u tablicama.
 - Jedinstveni identifikator za svaki redak u tablici – primarni ključ.
 - Atribut koji se koristi za povezivanje tablica – strani ključ.



Slika 3.3-Primjer relacijskog modela baze podataka [4]

4. **Objektno orijentirani model:** organizira podatke u obliku objekata. Svaki objekt može sadržavati podatke i metode za manipulaciju tim podacima. Primjer takvog modela je prikazan na slici 3.4.
- Podaci su pohranjeni kao objekti.
 - Podaci su organizirani u klase, a nasljeđivanje omogućuje stvaranje hijerarhije klasa.



Slika 3.4-Primjer objektnog modela baze podataka [5]

Različiti modeli podataka odgovaraju različitim vrstama aplikacija i potreba. Relacijski model je najčešće korišten zbog svoje jednostavnosti i standardizacije, dok novi modeli kao što su dokumentni, ključ-vrijednost i grafički model dobivaju na popularnosti zbog svoje fleksibilnosti i sposobnosti za rad s kompleksnim podacima što će biti detaljnije objašnjeno u poglavlju 4.3.

ER dijagram (engl. *Entity-Relationship dijagram*) je grafički prikaz strukture baze podataka koji prikazuje odnose između različitih entiteta. To je ključni alat u fazi dizajniranja jer pomaže u vizualizaciji i razumijevanju načina na koji će podaci biti organizirani, povezani i integrirani u bazi podataka. Izbor odgovarajućeg modela ovisit će o specifičnim zahtjevima aplikacije, prirodi podataka i performansama koje su potrebne.[6]

4. Relacijske i nerelacijske (NoSQL) baze podataka

Baze podataka su osnovni alati za skladištenje, organizaciju i pristup velikim količinama podataka, omogućujući učinkovito upravljanje informacijama. Postoje dvije glavne kategorije baza podataka: relacijske i nerelacijske (NoSQL). Razlikovanje između ovih dviju vrsta temelji se na načinu pohrane i pristupa podacima, te na vrstama aplikacija za koje su optimizirane.

Relacijske baze podataka koriste model temeljen na tablicama za organizaciju podataka. Svaka tablica sastoji se od redova i stupaca, gdje svaki red predstavlja jedan zapis, a svaki stupac predstavlja atribut tog zapisa. Relacijski model omogućava uspostavljanje veza između tablica pomoću primarnih i stranih ključeva, čime se osigurava referencijalni integritet podataka. Ove baze podataka podržavaju standardizirani jezik SQL (engl. *Structured Query Language*), koji omogućava složene upite, transakcije i manipulaciju podacima. Neki od najpoznatijih RDBMS-a (engl. *Relational Database Management System*) su MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server, i MariaDB. MariaDB posebno je nastala kao fork MySQL-a i nudi dodatne funkcionalnosti i poboljšanja, čime je postala popularna zbog svoje stabilnosti i visokih performansi.

Nerelacijske baze podataka, poznate i kao NoSQL baze, dizajnirane su za rad s velikim količinama raznovrsnih podataka koje se često ne uklapaju u tablični model. To su ne relacijski sustavi za brzu organizaciju i analizu različitih tipova podataka. One koriste različite modele za pohranu podataka, uključujući dokument-orijentirane, ključ-vrijednost, graf i stupac orijentirane baze podataka. Ove baze podataka nude veću fleksibilnost u strukturi podataka i često su optimizirane za specifične vrste upita i aplikacija. Na primjer, MongoDB koristi dokument-orijentirani model koji pohranjuje podatke u formatu sličnom JSON-u, dok Redis koristi model ključ-vrijednost za brz pristup podacima. Nerelacijske baze podataka su poznate po svojoj skalabilnosti i performansama, posebno u distribuiranim sustavima i aplikacijama koje zahtijevaju brzu obradu podataka.

Obje vrste baza podataka imaju svoje specifične prednosti i nedostatke, ovisno o specifičnim potrebama aplikacije i načinu na koji se podaci koriste. U ovom radu, poseban naglasak će biti na MariaDB, jednoj od popularnih relacijskih baza podataka, i njezinoj primjeni u analizi cijena, čime ću detaljno istražiti njezine karakteristike, prednosti i usporediti je s drugim bazama podataka poput tradicionalnog SQL-a.[6]

4.1. Prednosti i nedostaci relacijskih i nerelacijskih baza podataka

Relacijske baze podataka (RDBMS) su temeljni sustavi za upravljanje podacima koji koriste model tablica za organizaciju podataka. Njihove prednosti i nedostaci igraju ključnu ulogu u odabiru i implementaciji baze podataka, ovisno o specifičnim zahtjevima aplikacije.

- Prednosti:
 - Struktura i organizacija: Relacijske baze podataka koriste jasnu tabličnu strukturu (relacijski model) koja olakšava organizaciju podataka i njihovo razumijevanje.
 - Integritet podataka: Korištenjem ključeva (primarnih i stranih) relacijske baze podataka osiguravaju integritet podataka, sprječavajući neželjene ili nevaljane veze između podataka.
 - Standardizacija upita: SQL je standardizirani jezik za upite i manipulaciju podacima u relacijskim bazama, što olakšava rad s podacima i omogućava jednostavno pisanje složenih upita.
 - Transakcijska podrška: Relacijske baze podataka podržavaju ACID (engl. *Atomicity, Consistency, Isolation, Durability*) transakcijska svojstva, što osigurava da su transakcije pouzdane i sigurne.
 - Fleksibilnost i skalabilnost: Moderni RDBMS-i nude različite mogućnosti skalabilnosti, uključujući vertikalno i horizontalno skaliranje, što omogućava prilagodbu bazi podataka rastućim potrebama aplikacija.
- Nedostaci:
 - Složenost skaliranja: Horizontalno skaliranje relacijskih baza podataka može biti složeno i zahtijevati dodatne tehnike kao što su particioniranje i replikacija podataka.
 - Performanse: Za vrlo složene upite i visok broj transakcija, performanse relacijskih baza podataka mogu biti ograničene u usporedbi s nerelacijskim bazama podataka koje su optimizirane za specifične vrste upita.
 - Žestoka konkurencija za resurse: U višekorisničkim okruženjima, relacijske baze podataka mogu doći do konkurencije za resurse poput blokada i čekanja, što može usporiti performanse sustava.
 - Troškovi licenci: Komercijalne relacijske baze podataka često dolaze s visokim troškovima licenci i podrške, što može biti financijski izazovno za male organizacije ili startupove.

- Potrebno vrijeme za upravljanje: Administracija relacijskih baza podataka zahtijeva vještine i znanje, što može zahtijevati dodatno vrijeme i resurse za održavanje sustava.
- Razumijevanje prednosti i nedostataka relacijskih baza podataka ključno je za donošenje informiranih odluka o odabiru tehnologije baze podataka za specifične aplikacijske zahtjeve.[7]

4.2. Vrste relacijskih baza podataka

Relacijske baze podataka (RDBMS) koriste model tablica za organizaciju podataka te podržavaju upotrebu SQL-a za manipulaciju podacima. Postoje različite vrste relacijskih baza podataka, svaka s svojim karakteristikama i primjenama.

1. MySQL:

- Opis: MySQL je jedna od najpopularnijih open-source relacijskih baza podataka. Razvijen od strane Oracle Corporation, MySQL je poznat po svojoj jednostavnosti, brzini i fleksibilnosti.
- Primjena: Široko se koristi u web aplikacijama, CMS sustavima (Content Management System) te kao backend za različite vrste aplikacija.

2. PostgreSQL:

- Opis: PostgreSQL je napredni open-source objektno-relacijski sustav baza podataka. On podržava složene upite, transakcije i ima širok spektar proširenja.
- Primjena: Često se koristi za aplikacije koje zahtijevaju naprednu funkcionalnost baze podataka poput geografskih informacijskih sustava, analitike podataka i poslovnih aplikacija.

3. Oracle Database:

- Opis: Oracle Database je jedna od najpoznatijih komercijalnih relacijskih baza podataka. Poznata je po svojoj sigurnosti, pouzdanosti i naprednim mogućnostima upravljanja podacima.
- Primjena: Često se koristi u velikim korporacijama i organizacijama koje zahtijevaju visoke performanse, sigurnost i napredne funkcionalnosti baze podataka.

4. Microsoft SQL Server:

- Opis: Microsoft SQL Server je komercijalni relacijski sustav baza podataka razvijen od strane Microsofta. Integriran je s drugim Microsoftovim proizvodima kao što su Windows Server, Azure i Microsoft Office.
- Primjena: Često se koristi u korporativnom okruženju, posebno za aplikacije koje rade na Microsoftovom okruženju, kao i za poslovnu inteligenciju i analitiku.

5. MariaDB:

- Opis: MariaDB je open-source relacijski sustav baza podataka koji je nastao kao fork MySQL-a. Razvija ga MariaDB Corporation AB.
- Primjena: MariaDB je popularan izbor za web aplikacije, e-trgovinu, forume i druge aplikacije koje zahtijevaju brze i pouzdane baze podataka. Također je popularan izbor zbog svoje kompatibilnosti s MySQL-om, ali nudi i dodatne funkcionalnosti i poboljšanja o kojima se detaljnije govori u poglavlju 5.1.

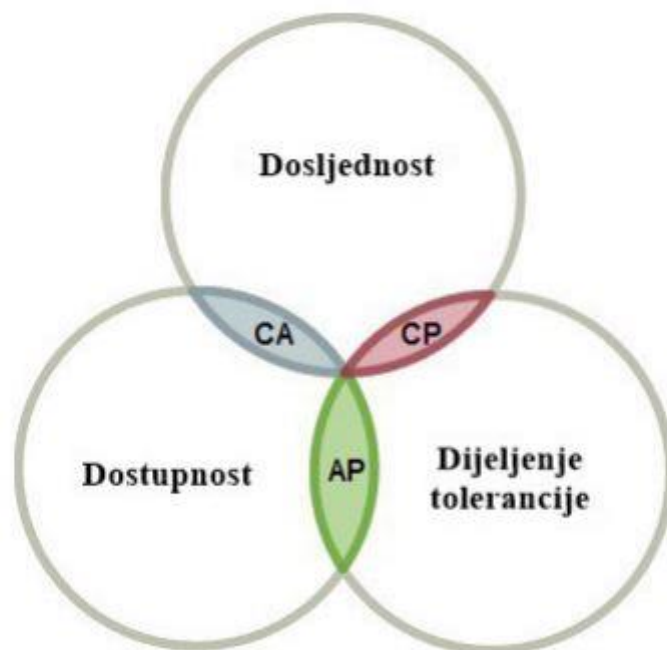
Svaka od ovih vrsta relacijskih baza podataka ima svoje specifične prednosti i primjene, ovisno o zahtjevima i potrebama aplikacija. Odabir prave vrste baze podataka ključan je za postizanje optimalnih performansi i efikasnosti u radu s podacima.[8]

4.3. Vrste nerelacijskih baza podataka

Nerelacijske baze podataka, poznate i kao NoSQL baze podataka, pružaju alternativne modele za pohranu podataka koji se ne temelje na tradicionalnom relacijskom modelu tablica. Ne koriste standardizirani SQL jezik nego se upiti vrše u raznim drugim jezicima. Rade na principu CAP teorema (engl. *Consistency, Availability, Partition tolerance*) koji je prikazan na slici 4.3.1.

- Dosljednost (engl. *Consistency*) – isti pogled nad podacima za svakog klijenta,
- Dostupnost (engl. *Availability*) – pohraniti ili pristupiti podacima može svaki klijent,
- Otpornost particija (engl. *Partition Tolerance*) – sustav se nalazi na više umreženih particija ali usprkos tome radi normalno.

Ove baze podataka su posebno dizajnirane za specifične potrebe aplikacija koje zahtijevaju fleksibilnost, skalabilnost i brze performanse [9].



Slika 4.3.1-CAP teorem [10]

1. Dokument-orientirane baze podataka:

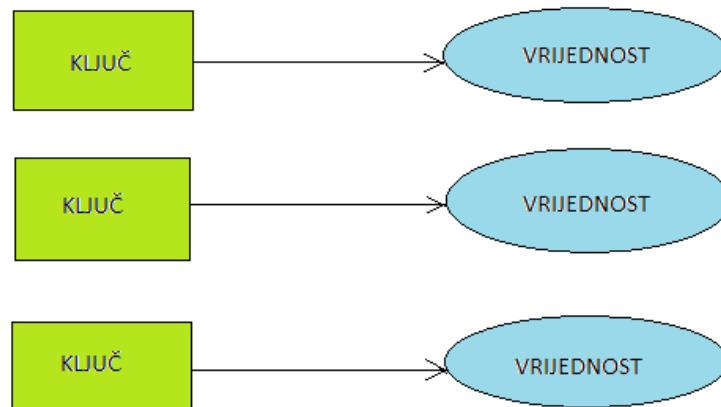
- Opis: Ove baze podataka pohranjuju podatke u dokumentima, koji su obično u formatu JSON-a (engl. *JavaScript Object Notation*), BSON-a ili XML-a. Svaki dokument može imati različitu strukturu, što omogućava veliku fleksibilnost u pohrani podataka. Primjer takvog modela je prikazan na slici 4.3.2.
- Primjeri: MongoDB, CouchDB, RavenDB.

```
{
  object_id:23141
  {
    ime: "Pero",
    prezime: "Perić",
    grad: "Varaždin",
    lokacija: [45.123, 47.232],
    auto [{
      model: "Opel Astra",
      godina: 2013,
      vrijednost: 86000,
    }
  ]
}
```

Slika 4.3.2-Dokument-orientirani model baze podataka (JSON)[11]

2. Baze podataka tipa ključ-vrijednost (eng. *Key Value store*):

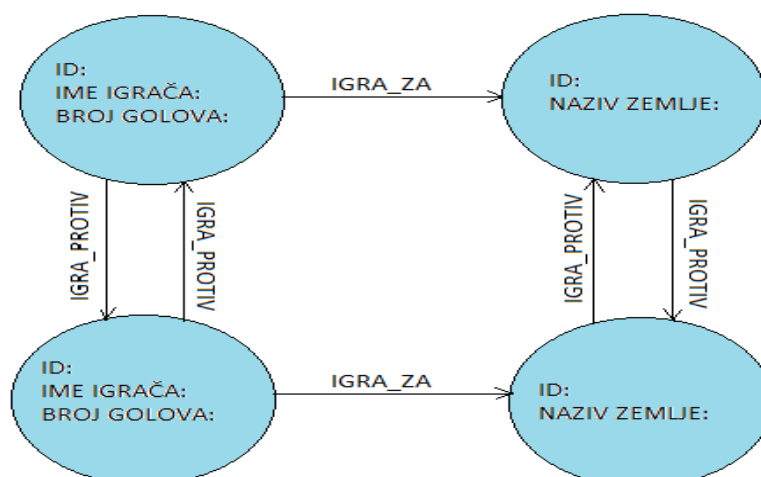
- Opis: Ključ-vrijednost baze podataka pohranjuju podatke kao jednostavne parove ključeva i vrijednosti. Ove baze podataka su vrlo brze za upite koji se temelje na ključevima, ali imaju ograničenu mogućnost složenih upita. Primjer takvog modela je prikazan na slici 4.3.3.
- Primjeri: Redis, DynamoDB, Riak.



Slika 4.3.3-Ključ-vrijednost model baze podataka

3. Graf baze podataka:

- Opis: Graf baze podataka modeliraju podatke kao grafove, gdje su entiteti (čvorovi) povezani usmjerenim ili neusmjerenim vezama (bridovima) kao što je prikazano na slici 4.3.4. Ovo je idealno za aplikacije koje zahtijevaju analizu veza između podataka.
- Primjeri: Neo4j, OrientDB, ArangoDB.



Slika 4.3.4-Graf orijentirani model baze podataka

4. Stupac orijentirane baze podataka:

- Opis: Stupac orijentirane baze podataka organiziraju podatke u stupce umjesto redova, što je posebno učinkovito za analizu i agregaciju podataka kao što je prikazano na slici 4.3.5.
- Primjeri: Apache Cassandra, HBase, Vertica.

row_id	Država	Grupa	Broj utakmica	Pobjeđene	Izgubljene	Neriješene	Bodovi
1	Švedska	A	3	2	1	0	4
2	Hrvatska	A	3	2	1	0	4
3	Srbija	A	3	1	2	0	2
4	Island	A	3	1	2	0	2
5	Francuska	B	3	3	0	0	6
6	Norveška	B	3	2	1	0	4
7	Bjelorusija	B	3	1	2	0	2
8	Austrija	B	3	0	3	0	0

Slika 4.3.5-Stupac orijentirani model baze podataka [11]

Nerelacijske baze podataka nude različite modele za pohranu podataka i pristup koji se mogu prilagoditi specifičnim zahtjevima aplikacija. Ove baze podataka često su odabir za moderna distribuirana i skalabilna rješenja gdje tradicionalni relacijski modeli mogu biti ograničavajući. Važno je pažljivo odabrati vrstu baze podataka koja najbolje odgovara potrebama vaše aplikacije kako bi se postigle optimalne performanse i efikasnost u radu s podacima.[4]

5. Usporedba MariaDB i MySQL-a

U ovom dijelu rada napravit će se usporedba MariaDB-a i MySQL-a, dva ključna sustava za upravljanje bazama podataka, kako bi se bolje razumjele njihove karakteristike, prednosti i primjene u kontekstu analize cijena.

MariaDB: MariaDB je open-source relacijski sustav baza podataka koji je stvoren 2009. godine od strane originalnih tvoraca MySQL-a, kao odgovor na zabrinutosti zbog preuzimanja MySQL-a od strane Oraclea. Razvija ga MariaDB Corporation AB, a poznat je po svojoj brzini, skalabilnosti i naprednim sigurnosnim značajkama. MariaDB je popularan izbor zbog svoje kompatibilnosti s MySQL-om, ali nudi i dodatne funkcionalnosti kao što su bolje performanse i fleksibilnost baze podataka.

MySQL: MySQL je razvijen sredinom 90-ih godina od strane švedske tvrtke MySQL AB. Postao je jedan od najpopularnijih open-source RDBMS-ova na svijetu. Godine 2008., Sun Microsystems je kupio MySQL AB, a potom je Oracle Corporation kupio Sun Microsystems 2010. godine, čime je MySQL postao dio Oraclea.[7]

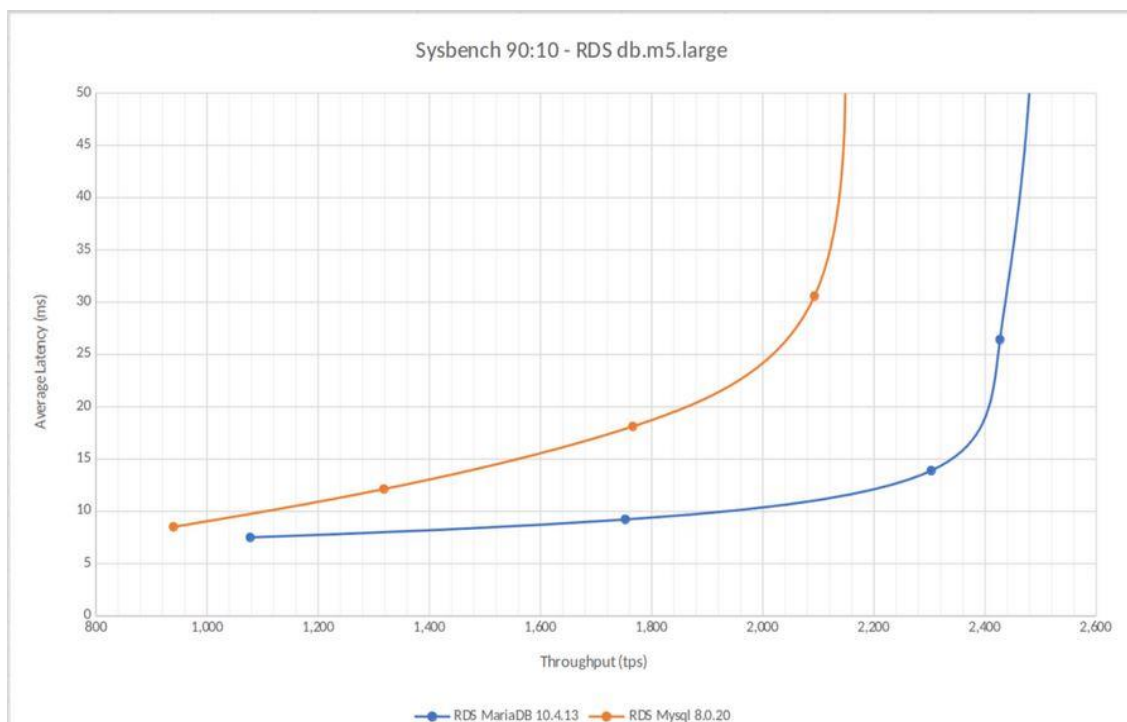
Usporedba MariaDB-a i MySQL-a:

- Sličnosti:
- Oboje koriste SQL kao jezik za upravljanje podacima, što olakšava prijelaz između različitih RDBMS-ova.
- Podržavaju relacijski model podataka, što omogućuje organizaciju podataka u tablicama s povezanim podacima.
- Razlike:

MariaDB je specifičan RDBMS s otvorenim pristupom (engl. *open-source*) i fokusom na brzinu i skalabilnost koji koristi isključivo GPL licencu, što znači da je uvijek slobodan i otvoren izvorni kod. Ovaj model pruža više slobode korisnicima i osigurava da će softver ostati otvoren i besplatan za sve.

- MariaDB nudi dodatne funkcionalnosti poput poboljšanih sigurnosnih značajki i naprednih performansi u usporedbi s tradicionalnim MySQL-om.
- MySQL ima bogat skup značajki i podržava različite skladišne motore kao što su InnoDB i MyISAM

Razumijevanje razlika i sličnosti između MariaDB-a i MySQL-a ključno je za odabir pravog alata za analizu cijena. MariaDB se često preferira zbog svoje otvorenosti i dodatnih mogućnosti, dok MySQL nudi standardizirani pristup koji je uobičajen u industriji. Ako je otvorenost i brži razvoj ključan, MariaDB može biti bolji izbor. Ako je potrebna komercijalna podrška i određene značajke dostupne samo u MySQL-u, tada MySQL može biti prikladniji. U idućim koracima istraživanja, analizirat će se kako se ovi alati mogu primijeniti u konkretne svrhe analize cijena, istražujući njihove performanse, funkcionalnosti i primjene u praksi. Na slici 5.1. je prikazana usporedba MariaDB-a i MySQL-a.



Slika 5.1-Usporedba MariaDB-a i MySQL-a [12]

5.1. MariaDB

MariaDB je *open-source* relacijski sustav baza podataka koji se razvio kao *fork* MySQL-a. Razvija ga MariaDB Corporation AB s ciljem pružanja brzine, skalabilnosti i naprednih značajki koje unaprjeđuju upravljanje podacima. Ovaj dio rada detaljnije će istražiti povijest, glavne karakteristike i prednosti MariaDB-a, fokusirajući se na razloge zašto je MariaDB odabrana za analizu cijena u ovom istraživanju.[5]

Glavne karakteristike i prednosti:

- Open-source: MariaDB je dostupan kao besplatan softver, što ga čini pristupačnim i fleksibilnim za različite vrste organizacija.
- Kompatibilnost s MySQL-om: MariaDB zadržava visoku kompatibilnost s MySQL-om, što olakšava migraciju i korištenje već postojećih vještina.
- Napredne sigurnosne značajke: Uključuje sigurnosne mehanizme poput transparentnog šifriranja podataka i kontrole pristupa. MariaDB također ima snažan fokus na sigurnost s redovitim ažuriranjima i zakrpama.
- Visoke performanse: Optimiziran je za brzinu i skalabilnost, podržavajući velike terete podataka i visok broj upita. MariaDB koristi unaprijeđene skladišne motore kao što su Aria i XtraDB (fork Percona XtraDB), koji mogu pružiti bolje performanse za specifične scenarije.

Povijest i razvoj:

- MariaDB je nastao 2009. godine kao fork MySQL-a nakon što je Oracle Corporation stekla Sun Microsystems, vlasnika MySQL-a.
- Razvoj MariaDB-a vodi se prema otvorenom modelu zajednice, što potiče brži razvoj i inovacije.

Primjena u praksi:

- MariaDB se često koristi za web aplikacije, e-trgovinu, forume te aplikacije koje zahtijevaju visoke performanse i skalabilnost.
- Prednosti MariaDB-a uključuju fleksibilnost u implementaciji i podršku za različite operativne sustave.

5.2. MySQL

MySQL je sustav za upravljanje relacijskim bazama podataka (RDBMS) otvorenog koda koji koristi SQL (engl. *Structured Query Language*) za upravljanje podacima. Razvijen je od strane švedske tvrtke MySQL AB, a kasnije je postao dio Sun Microsystems, koji je potom preuzeo Oracle Corporation. MySQL je jedan od najpopularnijih RDBMS-ova na svijetu i široko se koristi za razne web aplikacije, uključujući Facebook, Twitter i YouTube. Nastao je sredinom 1970-ih godina u IBM-u, temeljen na prethodnim radovima na jezicima za

upravljanje podacima te je ubrzo postao industrijski standard i danas je široko prihvaćen u industriji informacijskih tehnologija.[9]

Glavne karakteristike:

- MySQL je otvorenog koda, što znači da korisnici mogu slobodno koristiti, mijenjati i distribuirati softver. Dostupan je pod GNU General Public License (GPL), ali također nudi komercijalne licence za korisnike koji trebaju dodatne značajke ili podršku.
- MySQL koristi SQL, standardizirani jezik za upravljanje bazama podataka, što omogućuje korisnicima lako postavljanje, održavanje i pretraživanje podataka.
- MySQL je dostupan za različite operacijske sustave, uključujući Windows, Linux, macOS, i Unix, što ga čini fleksibilnim za razne okoline.

Prednosti:

- Jedna od najvećih prednosti MySQL-a je podrška koju pruža Oracle Corporation kroz razne komercijalne pakete podrške, uključujući hitne ispravke i profesionalne usluge koje mogu biti ključne za velika poduzeća.
- MySQL je široko testiran i koristi se u mnogim velikim proizvodnim okruženjima, što može pružiti dodatnu razinu povjerenja u njegovu stabilnost i pouzdanost.

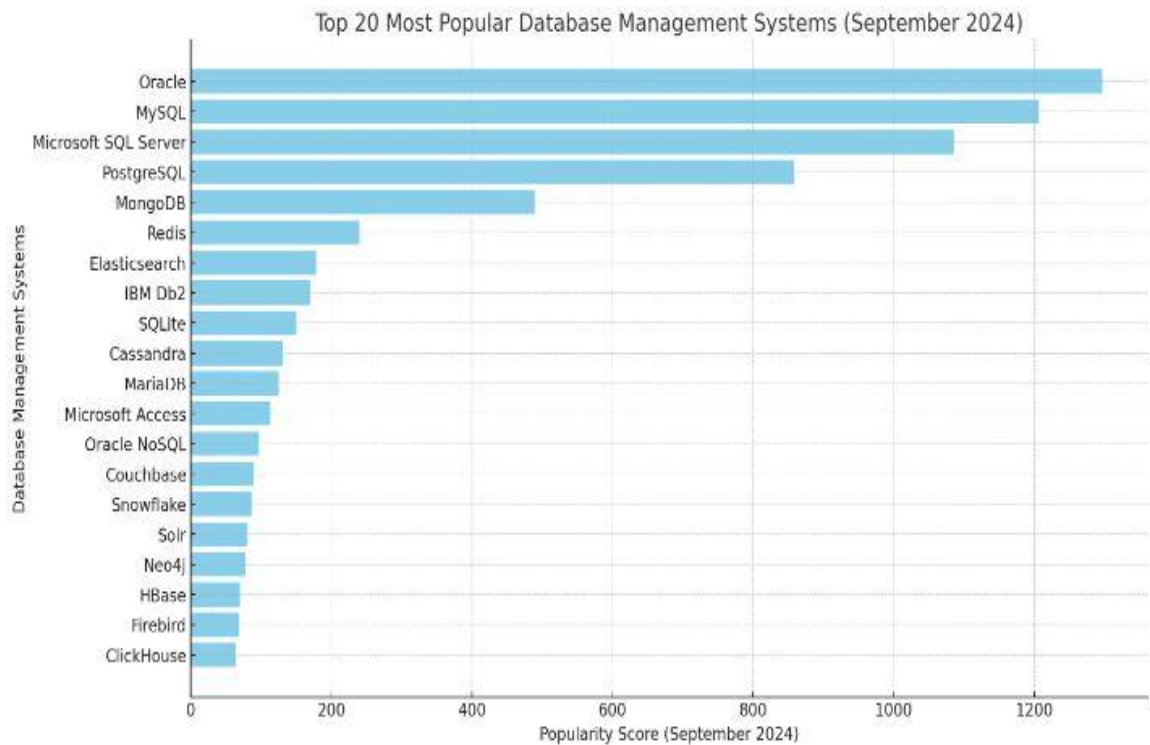
Nedostaci:

- MySQL koristi dualni licencni model, što znači da korisnici mogu koristiti MySQL besplatno pod General Public License (GPL), ali komercijalni korisnici koji ne žele otvoriti svoj izvorni kod moraju kupiti komercijalnu licencu od Oraclea. Ovo može biti skupo za poduzeća koja trebaju dodatne značajke ili podršku.
- U nekim složenim scenarijima može biti potrebno dodatno vrijeme za optimizaciju upita ili rješavanje performansi.
- Neke napredne značajke dostupne su samo u komercijalnim verzijama MySQL-a, što može ograničiti funkcionalnost za korisnike besplatne verzije.

Iako MySQL ostaje vrlo popularan i koristan sustav za upravljanje bazama podataka, važno je biti svjestan njegovih ograničenja i nedostataka. Za korisnike koji trebaju napredne SQL značajke, veću fleksibilnost u razvoju, ili potpunu otvorenost u razvoju i licenciranju, alternativa kao što je MariaDB može biti bolji izbor. Usporedbom MySQL-a s MariaDB-om i drugim sustavima za upravljanje bazama podataka može se lakše razumjeti kako odabrati

najprikladniji alat za analizu cijena, uzimajući u obzir specifične zahtjeve projekta i karakteristike podataka.[13]

Najpopularnije i najkorištenije baze podataka za rujan 2024. prikazane su na slici 5.2.1.[14]



Slika 5.2.1- Najpopularnije baze podataka [14]

Metoda izračunavanja rezultata DB-Engines rangiranja (engl. *DB-Engines Ranking*) je popis sustava za upravljanje bazama podataka poredanih prema njihovoj trenutnoj popularnosti. Popularnost sustava mjeri se pomoću sljedećih parametara:

- Broj spominjanja sustava na web stranicama, mjereno kao broj rezultata u upitima tražilica.
- Opći interes za sustav. Za ovo mjerenje koristi se učestalost pretraživanja u Google trendovima.
- Učestalost tehničkih rasprava o sustavu. Koristi se broj povezanih pitanja i broj zainteresiranih korisnika na poznatim stranicama za pitanja i odgovore.
- Broj ponuda poslova u kojima se spominje sustav. Koristi se broj ponuda na vodećim tražilicama poslova.
- Broj profila na profesionalnim mrežama u kojima se spominje sustav. Koristi se međunarodno najpopularnija profesionalna mreža LinkedIn.
- Relevantnost na društvenim mrežama. Broji se broj Twitter (X) *tweetova* u kojima se spominje sustav. [14]

6. Odabir MariaDB-a za analizu cijena

Odabir odgovarajuće baze podataka ključan je za uspješnu analizu cijena, koja zahtijeva pouzdanost, visoke performanse i fleksibilnost u rukovanju velikim količinama podataka. MariaDB, kao sustav za upravljanje relacijskim bazama podataka, nudi niz prednosti koje je čine idealnim izborom za ovu vrstu analize. U nastavku će se detaljno istražiti razlozi odabira MariaDB-a, njene performanse te primjere uspješnih primjena.

6.1. Razlozi odabira MariaDB-a

Relativna novost i inovativnost: MariaDB je relativno nova baza podataka koja je brzo stekla popularnost zahvaljujući kontinuiranim inovacijama i naprednoj tehnologiji. Njena sposobnost da se brzo prilagodi novim trendovima i tehnologijama čini je idealnim alatom za analizu cijena u dinamičnim tržišnim uvjetima. Inovacije u MariaDB-u omogućuju poboljšanja performansi, sigurnosti i skalabilnosti, što su ključni faktori za učinkovitu analizu podataka.

Kompatibilnost s MySQL-om: MariaDB je fork MySQL-a, što znači da zadržava visoku kompatibilnost s MySQL-om. Ova kompatibilnost olakšava migraciju postojećih MySQL aplikacija i baza podataka na MariaDB bez potrebe za značajnim promjenama. To je posebno važno za organizacije koje već koriste MySQL i žele iskoristiti prednosti MariaDB-a bez složenih migracijskih procesa.

Open-source priroda: MariaDB je open-source softver, što znači da je besplatan za korištenje i može se prilagoditi specifičnim potrebama korisnika. Ova otvorenost potiče zajednicu korisnika na doprinos razvoju i unapređenju sustava, što rezultira bržim i inovativnijim rješenjima. Organizacije svih veličina mogu koristiti napredne značajke MariaDB-a bez visokih troškova licenciranja, čime se smanjuju ukupni troškovi vlasništva.

Napredne sigurnosne značajke: Sigurnost podataka je ključna u bilo kojoj analizi cijena, posebno kada se radi o osjetljivim financijskim podacima. MariaDB nudi napredne sigurnosne značajke kao što su transparentno šifriranje podataka, kontrola pristupa i revizijski zapisi. Ove značajke osiguravaju da su podaci zaštićeni od neovlaštenog pristupa i manipulacije, čime se povećava povjerenje korisnika u sustav.

Visoke performanse MariaDB-a čine ga izuzetno pogodnim za analizu cijena, gdje je ključna brzina obrade i analiza velikih količina podataka. MariaDB koristi različite tehnike optimizacije koje omogućuju efikasno rukovanje velikim opterećenjima i složenim upitima:

- Napredni skladišni motori: MariaDB podržava različite skladišne motore poput InnoDB i Aria, koji su optimizirani za različite vrste upita i opterećenja. InnoDB, na primjer, nudi visoku pouzdanost i performanse za transakcijske aplikacije, dok Aria pruža brzu obradu za analitičke upite.
- Pametna replikacija: MariaDB podržava replikaciju na razini zapisa, što omogućuje brže i efikasnije sinkroniziranje podataka između različitih instanci baze podataka. Ovo je posebno korisno za distribuciju opterećenja i povećanje dostupnosti podataka.
- Optimizirani indeksi: MariaDB koristi optimizirane indekse koji omogućuju brz pristup podacima, značajno poboljšavajući performanse upita. Ova značajka je ključna za analizu cijena, gdje je potrebno brzo pretraživanje i agregacija podataka.

6.2. Primjeri uspješnih primjena MariaDB-a

Mnoge organizacije diljem svijeta prepoznale su vrijednost MariaDB-a i implementirale ga u svoje poslovne procese zbog njegovih brojnih prednosti. U ovom dijelu istražiti će se nekoliko primjera uspješnih primjena MariaDB-a, pokazujući kako je ova baza podataka omogućila učinkovito rukovanje velikim količinama podataka, poboljšala performanse i osigurala visoku razinu sigurnosti i fleksibilnosti.

Booking.com

Booking.com, jedna od najvećih svjetskih internetskih turističkih agencija, koristi MariaDB za upravljanje ogromnim količinama podataka o rezervacijama, korisnicima i smještajnim objektima. S obzirom na to da Booking.com svakodnevno obrađuje milijune transakcija i upita korisnika, potreban im je sustav baze podataka koji može pružiti visoke performanse i pouzdanost.

MariaDB omogućuje Booking.com-u brzo i učinkovito rukovanje podacima, što rezultira bržim vremenima odziva i poboljšanom korisničkom iskustvu. Fleksibilnost MariaDB-a također omogućava jednostavnu prilagodbu i skaliranje baze podataka kako bi se zadovoljili rastući zahtjevi poslovanja. Zahvaljujući MariaDB-u, Booking.com može učinkovito analizirati podatke o rezervacijama i korisnicima, omogućujući im donošenje informiranih poslovnih odluka i poboljšanje svojih usluga.

Wikipedia

Wikipedia, jedna od najposjećenijih web stranica na svijetu, koristi MariaDB za pohranu i upravljanje svojim ogromnim spremištem sadržaja. Wikipedia se oslanja na MariaDB kako bi osigurala brzu i pouzdanu dostupnost informacija korisnicima diljem svijeta.

MariaDB pomaže Wikipediji da učinkovito upravlja svojim globalnim opterećenjem, pružajući visoku razinu usluge korisnicima. Zahvaljujući naprednim značajkama replikacije i optimiziranim indeksima, Wikipedia može brzo pristupiti i pretraživati veliku količinu podataka. Ovo omogućava korisnicima brzo pronalaženje informacija koje traže, poboljšavajući korisničko iskustvo i osiguravajući da Wikipedia ostane pouzdan izvor informacija.

Deutsche Börse

Deutsche Börse, jedna od vodećih svjetskih burzi, koristi MariaDB za analizu financijskih podataka i upravljanje velikim količinama transakcija. Financijska industrija zahtijeva brz i pouzdan pristup podacima kako bi se omogućilo donošenje pravovremenih i informiranih odluka na tržištu kapitala.

MariaDB omogućuje Deutsche Börse-u da brzo analizira financijske podatke, osiguravajući visoke performanse i pouzdanost sustava. Napredne sigurnosne značajke MariaDB-a osiguravaju da su osjetljivi financijski podaci zaštićeni od neovlaštenog pristupa i manipulacije. Ovo je ključno za održavanje integriteta podataka i povjerenja investitora.

ServiceNow

ServiceNow, tvrtka koja nudi platformu za upravljanje digitalnim radnim tokovima, koristi MariaDB za upravljanje podacima svojih korisnika i pružanje visoko dostupnih usluga. MariaDB omogućuje ServiceNow-u da učinkovito rukuje velikim količinama podataka, osiguravajući visoku dostupnost i pouzdanost njihovih usluga.

Zahvaljujući MariaDB-u, ServiceNow može brzo i pouzdano obrađivati podatke korisnika, omogućujući im pružanje visokokvalitetnih usluga. Fleksibilnost i skalabilnost MariaDB-a također omogućavaju ServiceNow-u prilagodbu i proširenje njihovih sustava kako bi zadovoljili rastuće zahtjeve svojih korisnika.

Tencent

Tencent, jedan od najvećih kineskih tehnoloških divova, koristi MariaDB za upravljanje podacima svojih brojnih aplikacija i usluga. Tencent se oslanja na MariaDB kako bi osigurao visoke performanse, skalabilnost i pouzdanost svojih sustava.

MariaDB omogućuje Tencentu brzo i učinkovito rukovanje velikim količinama podataka, što je ključno za njihove aplikacije s visokim prometom. Osim toga, napredne sigurnosne značajke MariaDB-a pomažu u zaštiti osjetljivih korisničkih podataka, osiguravajući visoku razinu sigurnosti i povjerenja korisnika.

Ovi primjeri jasno pokazuju kako MariaDB može pružiti visoke performanse, pouzdanost i fleksibilnost potrebne za učinkovito upravljanje i analizu velikih količina podataka u različitim industrijama. Njegova sposobnost prilagodbe specifičnim potrebama korisnika i kontinuirane inovacije čini ga izvanrednim izborom za analizu cijena i mnoge druge aplikacije.

7. Primjena MariaDB-a

Primjena MariaDB-a, kao i svakog sustava za upravljanje bazama podataka (DBMS), može obuhvaćati različite scenarije ovisno o potrebama korisnika i vrsti aplikacije koja se razvija ili održava. Evo nekoliko tipičnih primjera primjene MariaDB-a:

1. Web Aplikacije

MariaDB se često koristi kao backend za web aplikacije zbog svoje pouzdanosti, performansi i široko podržanih tehnologija za povezivanje i integraciju.

Evo kako se može primijeniti u web okruženju:

- **Skladištenje korisničkih podataka:** MariaDB se koristi za pohranu podataka o korisnicima, njihovim profilima, autentifikaciji i autorizaciji.
- **Skladištenje sadržaja:** Slike, videozapisi, članci i drugi sadržaji pohranjuju se u MariaDB-u ili se referenciraju putem veza.
- **Transakcijske operacije:** Za obradu transakcija kao što su kupnje, rezervacije i narudžbe, MariaDB pruža ACID (engl. *Atomicity, Consistency, Isolation, Durability*) svojstva.

2. Analitika i izvještavanje

- **Data warehousing:** Za analiziranje velikih skupova podataka, MariaDB se može koristiti kao skladište podataka (engl. *data warehouse*). Podaci se mogu učitavati iz različitih izvora, transformirati i analizirati pomoću SQL upita.
- **Business intelligence:** BI alati mogu se integrirati s MariaDB-om radi izrade naprednih izvještaja, grafikona i vizualizacija na temelju podataka pohranjenih u bazi.

3. IoT (engl. *Internet of Things*)

- **Pohrana senzorskih podataka:** MariaDB se može koristiti za pohranu podataka iz senzora, IoT uređaja ili strojeva. Podaci se mogu analizirati u stvarnom vremenu ili se koristiti za kasniju analizu i izvještavanje.
- **Upravljanje uređajima:** Informacije o uređajima, njihovom statusu i upravljačkim akcijama mogu se pohraniti u MariaDB-u.

4. CMS (engl. *Content Management System*)

- **Skladištenje sadržaja:** MariaDB se može koristiti kao backend za CMS sustave za upravljanje sadržajem kao što su WordPress, Drupal ili Joomla. Pohranjuje se tekstualni sadržaj, korisnički komentari, konfiguracijske postavke i druge informacije.
- **Praćenje aktivnosti korisnika:** Informacije o posjetiteljima, njihovim interakcijama s web stranicom ili platformom također se mogu pohraniti i analizirati.

5. E-commerce

- **Pohrana proizvoda i narudžbi:** MariaDB se često koristi za pohranu kataloga proizvoda, inventara, informacija o narudžbama, kupcima i transakcijama.
- **Preporuke i personalizacija:** Na temelju podataka pohranjenih u MariaDB-u, može se implementirati funkcionalnost preporuka proizvoda, personaliziranih ponuda i prilagođenog marketinga.

6. Sigurnosne kopije i oporavak

- **Redovito sigurnosno kopiranje:** MariaDB omogućuje automatsko ili ručno sigurnosno kopiranje podataka radi zaštite od gubitka podataka.
- **Oporavak podataka:** U slučaju neželjenog brisanja ili kvara, MariaDB nudi mehanizme oporavka podataka koji omogućuju vraćanje baze podataka u prethodno stanje.

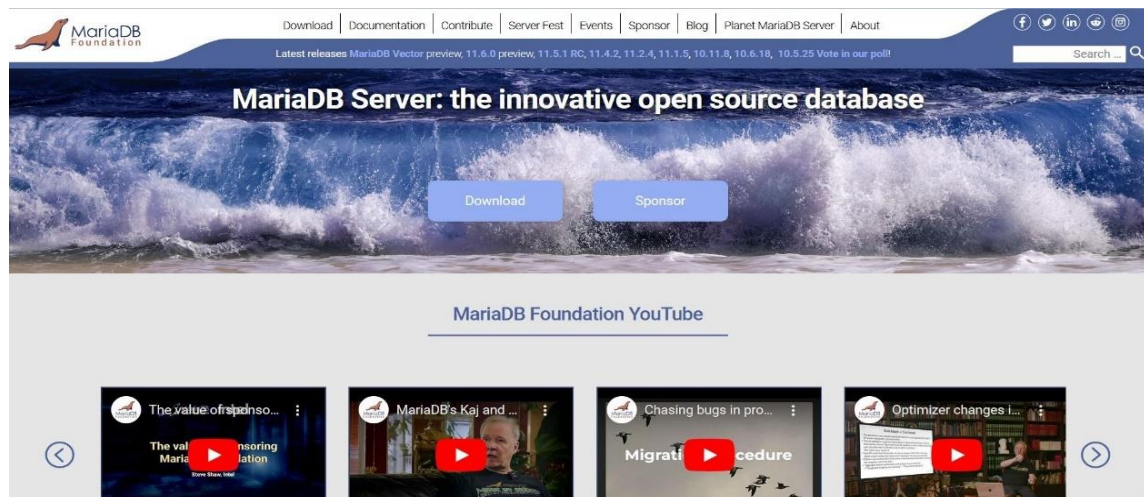
MariaDB je snažan DBMS koji se može prilagoditi različitim scenarijima primjene od jednostavnih aplikacija do složenih sustava za analitiku i upravljanje velikim količinama podataka. Njegove prednosti uključuju otvorenost, brzinu, skalabilnost i podršku za transakcije. Integracija s različitim platformama i alatima čini ga popularnim izborom među razvojnim zajednicama diljem svijeta.

7.1. Način preuzimanja i instaliranja MariaDB baze podataka

Koraci:

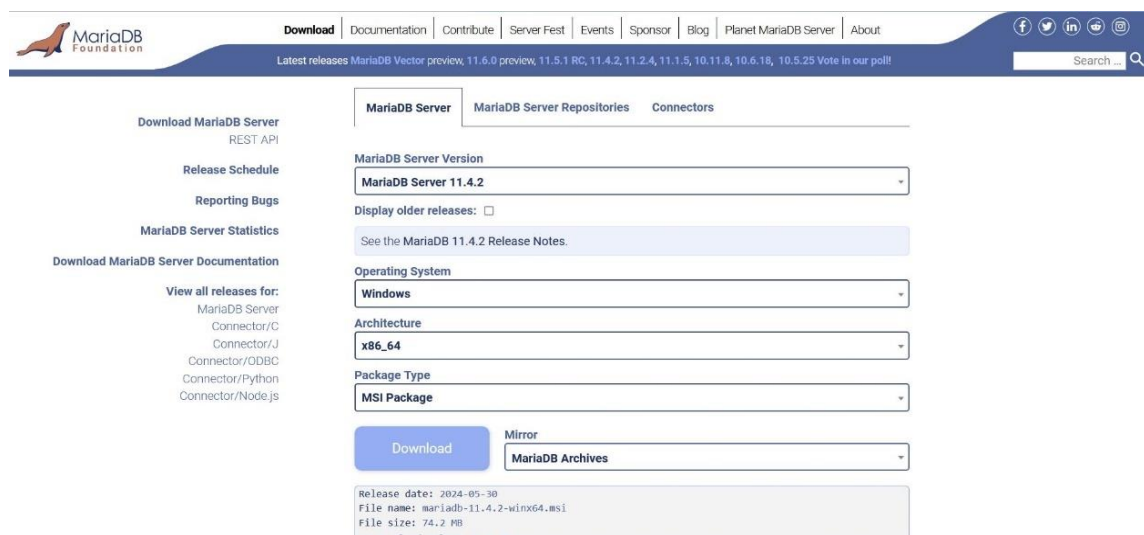
1. Preuzimanje MariaDB Windows instalacijskog programa (engl. *Windows Installer program*)

- Otvaranjem web stranice mariadb.org prikazat će se tipka za preuzimanje (engl. *Download*) što je prikazano na slici 7.1.1.



Slika 7.1.1-Početna stranica za preuzimanje MariaDB poslužitelja [15]

- Klikom na tipku preuzimanje (engl. *Download*) otvorila se stranica na kojoj se odabire za koji operativni sustav se preuzima MariaDB poslužitelj. U ovom slučaju to je sustav Windows kao što je prikazano na slici 7.1.2.



Slika 7.1.2-Stranica za odabir operativnog sustava [15]

- Još jedan klik na tipku *download* i krenut će preuzimanje *.msi* instalacijskog programa koji je prikazan na slici 7.1.3.



Slika 7.1.3-Mariadb.msi instalacijski program [15]

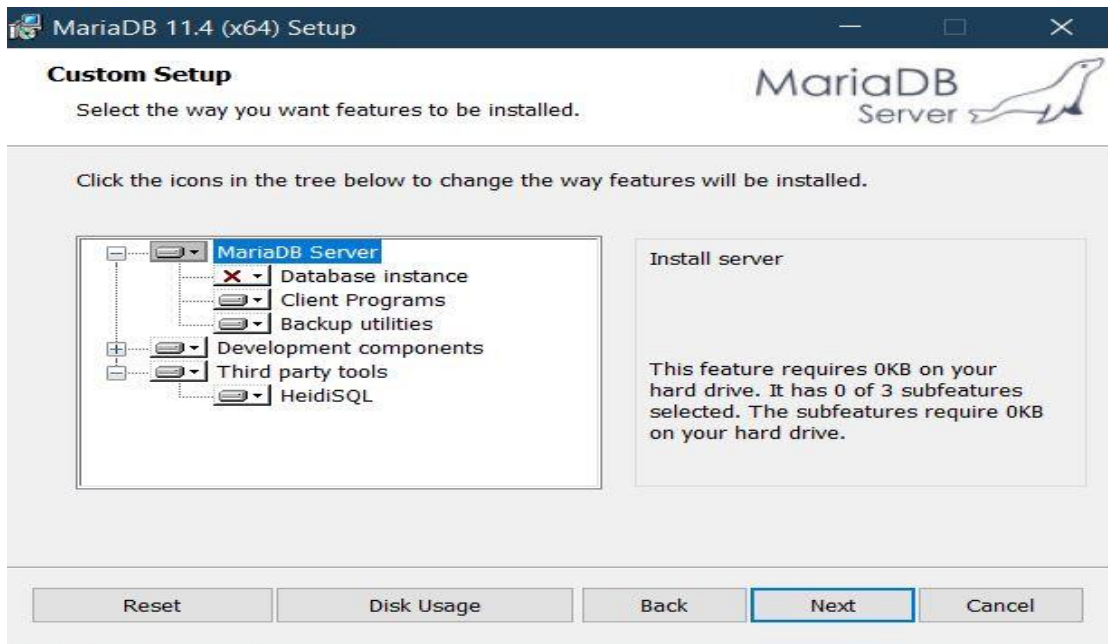
2. Instaliranje MariaDB servera

- Dvostrukim klikom na preuzeti *.msi* instalacijski program otvorit će se prozor dobrodošlice za instalaciju kao što je prikazano na slici 7.1.4.



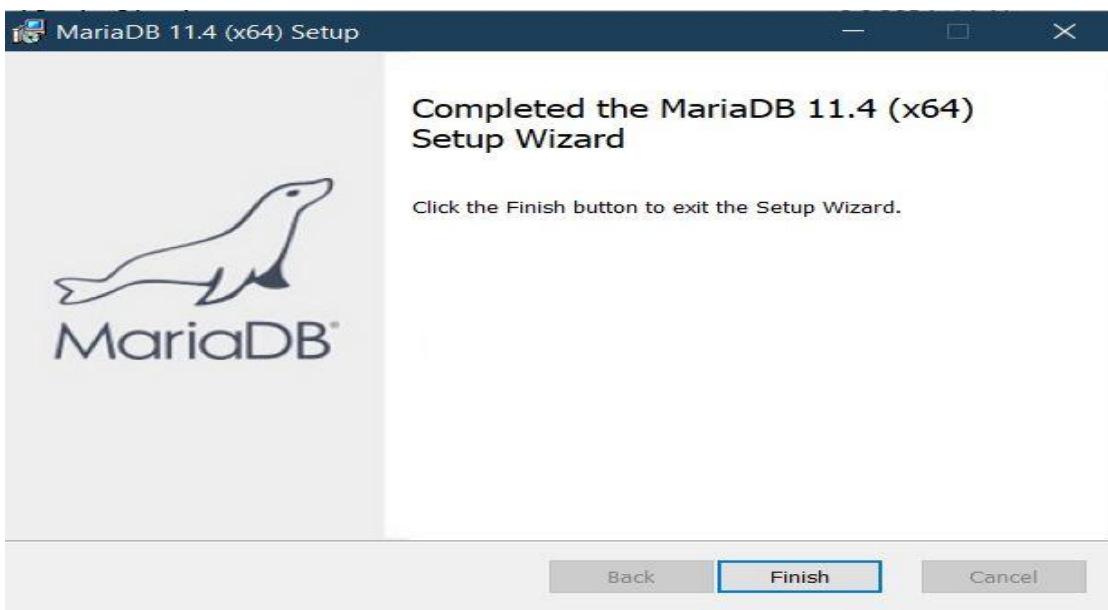
Slika 7.1.4-Početni prozor za instalaciju [15]

- Korištenjem čarobnjaka za postavljanje aplikacije (engl. *Setup Wizard*), počeo je postupak instalacije koji je u jednom trenutku ponudio opciju mogućih komponenata za instalaciju. Klikom na tipku nastavi (engl. *next*) odabiru se već unaprijed određene opcije koje u pravilu ne treba mijenjati kao što je prikazano na slici 7.1.5.



Slika 7.1.5-Opcije za instalaciju [15]

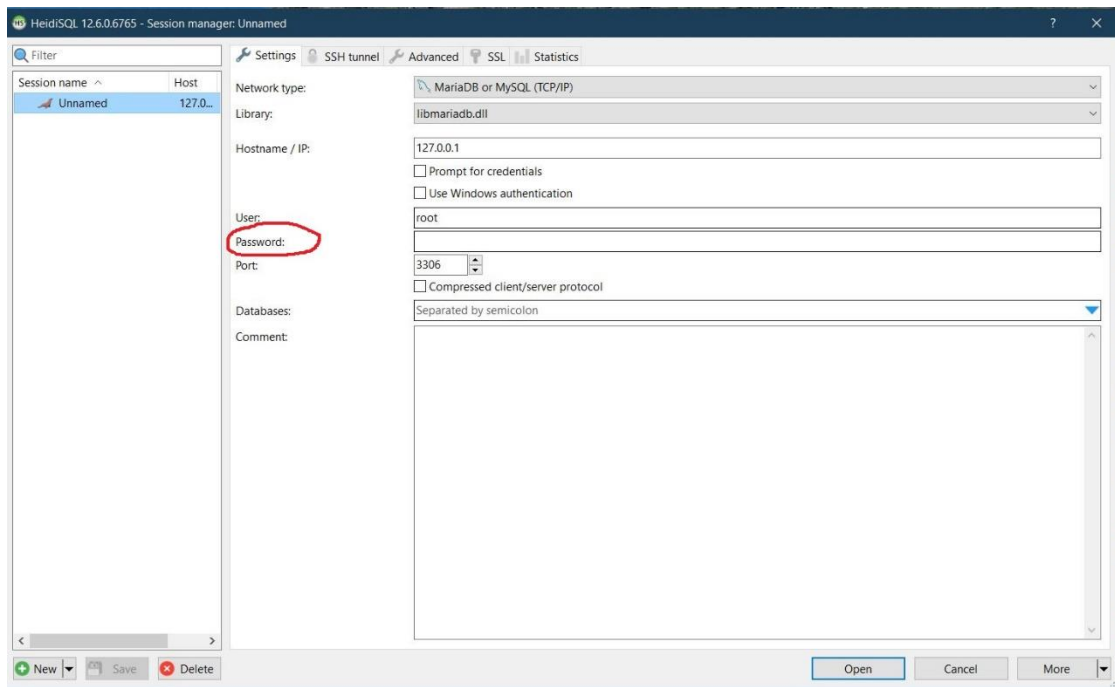
- Klikom na tipku nastavi (engl. *next*) pokreće se instalacija onih značajki koje su prethodno odabrane. Na kraju se prikazao prozor gdje se klikom na tipku završi (engl. *finish*) privodi kraju instalacija kao što je prikazano na slici 7.1.6.



Slika 7.1.6-Završni prozor instalacije [15]

3. Postavljanje MariaDB poslužitelja

- Posljednji korak je postavljanje *root* lozinke korisnika koji je administrator MariaDB baze podataka te postoji i opcija omogućiti ili onemogućiti daljinski pristup *root* korisnika kao što je prikazano na slici 7.1.7.



Slika 7.1.7.-Prozor za postavljanje root lozinke [15]

7.2. Priprema podataka

Priprema podataka (engl. *data preparation*) u kontekstu MariaDB-a odnosi se na proces pripreme i organizacije podataka unutar baze kako bi se omogućila efikasna analiza, upravljanje ili korištenje podataka u aplikacijama ili sustavima. Ovaj proces uključuje nekoliko ključnih koraka kako bi se osigurala kvaliteta, dosljednost i korisnost podataka. Evo nekoliko važnih aspekata pripreme podataka u MariaDB-u:

1. Učitavanje Podataka

Učitavanje podataka je početni korak u pripremi podataka. Može uključivati:

- **Uvoz podataka:** Korištenjem SQL `LOAD DATA` naredbe ili alata poput `mysqldump` za uvoz podataka iz vanjskih izvora kao što su CSV datoteke, Excel tablice ili druge baze podataka.
- **Sinkronizaciju s aplikacijama:** Automatizirano učitavanje podataka putem aplikacija koje generiraju podatke u realnom vremenu, kao što su IoT uređaji ili web aplikacije.

2. Čišćenje Podataka

Čišćenje podataka je ključni korak za osiguravanje kvalitete podataka. Ovaj proces uključuje:

- **Identifikaciju i uklanjanje duplikata:** Prepoznavanje i uklanjanje dvostrukih zapisa iz baze podataka.

- **Ispravljanje grešaka:** Ispravljanje grešaka u podacima kao što su pogrešni formati, nedostajuće vrijednosti ili nekonzistentni zapisi.
- **Normalizaciju podataka:** Organiziranje podataka u strukturu koja minimizira redundanciju i omogućuje efikasnu upotrebu.

3. Integraciju podataka

Integracija podataka je proces spajanja različitih izvora podataka u jedinstvenu strukturu ili bazu podataka. Ovaj korak uključuje:

- **Spajanje podataka:** Kombiniranje podataka iz više tablica ili izvora u jedan skup podataka koji je koristan za analizu ili upravljanje.
- **Ujednačavanje formata:** Prijenos podataka u zajednički format ili standard koji olakšava analizu i upravljanje.

4. Indeksiranje Podataka

Indeksiranje je važan korak za optimizaciju performansi upita nad bazom podataka.

To uključuje:

- **Stvaranje indeksa:** Definiranje indeksa na često korištenim stupcima kako bi se ubrzali upiti za pretraživanje, sortiranje ili grupiranje podataka.
- **Optimizaciju performansi:** Analiza i optimizacija indeksa kako bi se poboljšale performanse baze podataka, posebno u slučajevima s velikim skupovima podataka.

5. Sigurnosne Kopije

Redovito stvaranje sigurnosnih kopija podataka ključno je za zaštitu podataka od gubitaka ili neželjenih događaja. Ovaj proces uključuje:

- **Automatizirano sigurnosno kopiranje:** Postavljanje automatskih zadatka za redovito sigurnosno kopiranje baze podataka.
- **Pohranu sigurnosnih kopija:** Čuvanje sigurnosnih kopija na sigurnom mjestu ili u oblaku kako bi se osigurala otpornost na fizičke ili tehničke neuspjehe.

6. Dokumentacija i metapodaci

Dokumentiranje strukture baze podataka i metapodataka pomaže u razumijevanju i upravljanju podacima. Ovaj korak uključuje:

- **Opis tablica i veza:** Dokumentiranje svih tablica, stupaca, veza i ograničenja unutar baze podataka.
- **Korištenje metapodataka:** Upotreba metapodataka za praćenje verzija i promjena u strukturi baze podataka.

Priprema podataka u MariaDB-u nije samo administrativni proces već ključan korak koji omogućuje efikasnu upotrebu podataka u aplikacijama, analizi ili drugim poslovnim procesima. Svaki od navedenih koraka igra važnu ulogu u osiguravanju kvalitete, dosljednosti i performansi podataka unutar baze.

7.3. Izvori podataka

Izvori podataka (engl. *data sources*) u kontekstu MariaDB-a predstavljaju različite lokacije ili sustave iz kojih se podaci mogu učitati ili integrirati u MariaDB bazu radi daljnje obrade, analize ili upotrebe u aplikacijama. Važno je identificirati i razumjeti različite vrste izvora podataka kako bi se osigurala potpunost i kvaliteta podataka unutar baze. Evo nekoliko uobičajenih izvora podataka koji se mogu integrirati s MariaDB-om:

1. Vanjske Datoteke

- **CSV datoteke:** Česte su datoteke koje sadrže strukturirane podatke odvojene zarezima, koje je lako učitati u MariaDB korištenjem `LOAD DATA INFILE` naredbe. Za potrebe ove analize korišten je spomenuti način učitavanja datoteka.
- **Excel datoteke:** Podaci iz Excel tablica mogu se prebaciti u CSV format ili učitati direktno u MariaDB, često korištenjem alata za uvoz podataka.
- **JSON datoteke:** Učitavanje JSON (engl. *JavaScript Object Notation*) formata podataka može biti korisno za polu-strukturirane ili nestrukturirane podatke koji se mogu transformirati i pohraniti u MariaDB.

2. Ostale Baze Podataka

- **MySQL:** Ako se prelazi s MySQL na MariaDB, podaci mogu jednostavno migrirati korištenjem SQL skripti ili alata za repliciranje podataka.
- **PostgreSQL, SQL Server, Oracle:** Podaci iz drugih relacijskih baza podataka mogu migrirati u MariaDB koristeći SQL skripte ili specijalizirane alate za migraciju podataka.

3. Web Servisi i API-ji

- **RESTful API-ji:** Podaci dostupni putem RESTful API-ja mogu se integrirati u MariaDB korištenjem HTTP zahtjeva i JSON formatiranja.
- **SOAP Web servisi:** Podaci dostupni putem SOAP web servisa mogu se učitati u MariaDB, obično korištenjem posebnih biblioteka ili alata za komunikaciju s web servisima.

4. IoT Uređaji

- **Senzori i uređaji za praćenje:** Podaci generirani od strane IoT uređaja, kao što su senzori temperature ili vlage, mogu se prikupiti i pohraniti u MariaDB za analizu i obradu.

5. Cloud Servisi

- **S3 (Amazon Simple Storage Service):** Podaci pohranjeni na Amazon S3 mogu se učitati u MariaDB korištenjem specijaliziranih alata ili SQL skripti za preuzimanje podataka.
- **Google Cloud Storage, Microsoft Azure Storage:** Slično kao i s Amazon S3, podaci pohranjeni u oblaku mogu se integrirati u MariaDB za daljnju analizu ili pohranu.

6. Lokalni Sustavi i Aplikacije

- **Interni sustavi za praćenje:** Podaci iz internih aplikacija ili sustava za praćenje, poput CRM-a ili ERP-a, mogu se integrirati u MariaDB kako bi se stvorio centralizirani izvor podataka.

Raznolikost izvora podataka omogućava MariaDB-u da služi kao centralna točka za pohranu i analizu različitih vrsta podataka. Integracija podataka iz raznih izvora omogućava organizacijama da optimiziraju svoje operacije, poboljšaju donošenje odluka i pruže bolju podršku korisnicima. Važno je odabrati odgovarajući pristup i alate za integraciju podataka kako bi se osigurala kvaliteta i dosljednost podataka unutar baze podataka.

7.4. Izrada konceptualnog modela

Izrada konceptualnog modela za indeks potrošačkih cijena (CPI) predstavlja prvi važan korak za analizu te osiguravan točno i dosljedno praćenje promjena cijena potrošačkih dobara i usluga. U ovom modelu najvažnija je identifikacija ključnih entiteta, atributa i njihovih veza. Ovaj primjer je orijentiran na osnovne entitete i njihove veze.

Osnovni entiteti i njihovi atributi:

1. Proizvod (engl. *Product*)

- ProductID (Primarni ključ)
- ProductName
- CategoryID (Strani ključ prema entitetu Category)
- UnitOfMeasure

2. Prodavaonica (engl. *Store*)

- StoreID (Primarni ključ)
- StoreName
- Location
- StoreType
- RegionID (Strani ključ prema entitetu Region)

3. Cijena (engl. *Price*)

- PriceID (Primarni ključ)
- ProductID (Strani ključ prema entitetu Product)
- StoreID (Strani ključ prema entitetu Store)
- Date
- Amount

4. Kategorija proizvoda (engl. *Category*)

- CategoryID (Primarni ključ)
- CategoryName

5. Geografska regija (engl. *Region*)

- RegionID (Primarni ključ)
- RegionName

Veze između entiteta:

1. Proizvod-Kategorija (engl. *Product-Category*)

- Jedan proizvod pripada jednoj kategoriji, dok jedna kategorija može sadržavati više proizvoda (odnos jedan-prema-više).

2. Prodavaonica-Regija (engl. *Store-Region*)

- Jedna prodavaonica se nalazi u jednoj geografskoj regiji, dok jedna regija može imati više prodavaonica (odnos jedan-prema-više).

3. Proizvod-Cijena (engl. *Product-Price*)

- Jedan proizvod može imati više cijena u različitim prodavaonicama i/ili različitim vremenskim periodima (odnos jedan-prema-više).

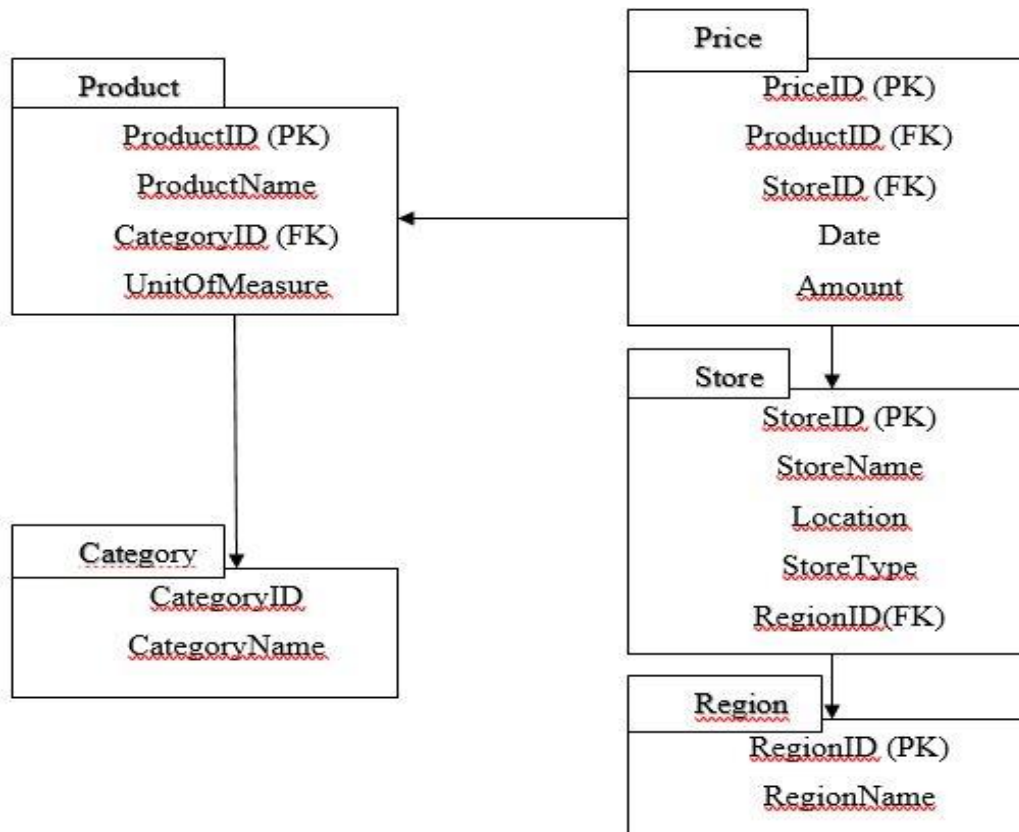
4. Prodavaonica-Cijena (engl. *Store-Price*)

- Jedna prodavaonica može imati više cijena za različite proizvode i/ili različitim vremenskim periodima (odnos jedan-prema-više).

Veze među atributima:

- Product je povezan sa Category putem atributa CategoryID.
- Price je povezan sa Product putem atributa ProductID.
- Price je povezan sa Store putem atributa StoreID.
- Store je povezan sa Region putem atributa RegionID.

Na slici 7.4.1. je prikazana relacijsko konceptualna shema informacijskog sustava za CPI.



Slika 7.4.1-Relacijsko-konceptualna shema informacijskog sustava za CPI

Objašnjenje:

Svaka strelica na dijagramu prikazuje vezu između entiteta. Strelica od *Product* do *Category* prikazuje da svaki proizvod pripada jednoj kategoriji. Strelica od *Price* do *Product* i *Store* pokazuje da je svaka cijena povezana sa jednim proizvodom i jednom prodavaonicom. Kroz ovaj model je jasno vidljiva struktura baze podataka i olakšava razumijevanje na koji način su podaci međusobno povezani.

7.5. Unos podataka u MariaDB

Unos podataka u MariaDB može se obaviti na nekoliko načina, ovisno o izvoru podataka i specifičnim zahtjevima sustava.

Evo nekoliko uobičajenih metoda za unos podataka:

1. Ručni unos podataka je prikazan na slici 7.5.1.

Ručni unos podataka obično se koristi za manje količine podataka ili za unos pojedinačnih zapisa. To se može obaviti korištenjem SQL naredbi unutar MariaDB CLI alata ili putem grafičkog korisničkog sučelja kao što je phpMyAdmin ili HeidiSQL.

Na primjer, za unos podataka u tablicu `cijene` u MariaDB koristeći SQL naredbu:

```
sql

INSERT INTO cijene (proizvod, mjesečna_stopa)
VALUES ('Proizvod A', '100');
```

Slika 7.5.1-SQL naredba za unos podataka u tablicu cijene

2. Uvoz podataka iz datoteka je prikazan na slici 7.5.2.

Ako postoje veće količine strukturiranih podataka spremljenih u CSV, Excel ili drugim formatima datoteka, mogu se uvesti u MariaDB korištenjem naredbe `LOAD DATA INFILE`.

Na primjer, za uvoz CSV datoteke `podaci.csv` u tablicu `cijene` u MariaDB:

```
sql

LOAD DATA INFILE 'putanja/do/podaci.csv'
INTO TABLE cijene
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

Slika 7.5.2-SQL naredbe za uvoz .CSV datoteke u MariaDB

3. Integracija s vanjskim izvorima putem API-ja

Ako se podaci redovito ažuriraju ili dolaze iz vanjskih izvora poput RESTful API-ja, mogu se automatski učitavati podatke u MariaDB koristeći skripte za uvoz podataka ili programski jezik poput Pythona ili JavaScripta.

4. Repliciranje podataka iz drugih baza podataka

Ako već postoje podaci u drugim bazama podataka poput MySQL-a ili PostgreSQL-a, mogu se replicirati u MariaDB korištenjem alata za replikaciju ili jednostavno kopiranjem SQL skripti.

5. Korištenje alata za ETL (engl. *Extract, Transform, Load*)

Za kompleksne integracije podataka mogu se koristiti alati za ETL poput Apache NiFi, Talend ili Pentaho. Ovi alati omogućuju ekstrakciju podataka iz različitih izvora, transformaciju podataka prema potrebama vašeg sustava, te učitavanje podataka u MariaDB.

6. Korištenje MariaDB Connectora za različite aplikacije

MariaDB nudi razne konektore koji olakšavaju integraciju s raznim aplikacijama i alatima. Na primjer, MariaDB Connector/Python ili MariaDB Connector/J za Java aplikacije omogućuju jednostavnu interakciju s bazom podataka.

Važnost sigurnosti i performansi

Pri unosu podataka u MariaDB važno je voditi računa o sigurnosnim aspektima kao što su autentifikacija i autorizacija, kao i optimizacija performansi za brzi i učinkoviti unos podataka. Redovito ažuriranje i održavanje baze podataka također su ključni za osiguranje stabilnosti i integriteta podataka.

Korištenje odgovarajućih metoda i alata za unos podataka u MariaDB omogućuje se efikasno upravljanje podacima i da se iskoristi potencijal baze podataka u aplikacijama i analitičkim procesima.

7.6. Analiza cijena pomoću MariaDB-a

Analiza cijena pomoću MariaDB-a može biti vrlo korisna za razumijevanje trendova, uspoređivanje podataka i donošenje informiranih poslovnih odluka.

Primjer kako se može provesti analiza cijena koristeći MariaDB:

1. SQL Upiti za Analizu

SQL je moćan jezik koji omogućuje izvlačenje, filtriranje, grupiranje i računanje podataka iz baze.

Evo nekoliko primjera SQL upita koji se mogu koristiti za analizu cijena:

Naredba za prikaz cijena po kategorijama proizvoda je prikazana na slici 7.6.1.

```
sql

SELECT kategorija_proizvoda, AVG(cijena) AS prosjecna_cijena
FROM tablica_cijena
GROUP BY kategorija_proizvoda;
```

Slika 7.6.1-SQL naredbe za grupiranje podataka o cijenama prema kategorijama

Ovaj upit grupira podatke o cijenama prema kategorijama proizvoda i izračunava prosječnu cijenu za svaku kategoriju.

Naredba za usporedbu cijena između različitih razdoblja je prikazana na slici 7.6.2.

```
sql

SELECT proizvod,
       MIN(cijena) AS minimalna_cijena,
       MAX(cijena) AS maksimalna_cijena
FROM cijene
WHERE datum BETWEEN '2024-01-01' AND '2024-06-30'
GROUP BY proizvod;
```

Slika 7.6.2-SQL naredbe za minimalne i maksimalne cijene unutar određenog razdoblja

Ovaj upit prikazuje minimalne i maksimalne cijene za svaki proizvod unutar određenog razdoblja, što omogućuje usporedbu promjena cijena tijekom vremena.

Pronalaženje proizvoda s najvećim porastom cijene prikazano je na slici 7.6.3.

```
sql

SELECT proizvod, MAX(cijena) - MIN(cijena) AS porast_cijene
FROM tablica_cijena
GROUP BY proizvod
ORDER BY porast_cijene DESC
LIMIT 10;
```

Slika 7.6.3-SQL naredbe za proizvode s najvećim porastom cijena

Ovaj upit identificira proizvode s najvećim porastom cijene, što može biti korisno za praćenje fluktuacija na tržištu.

2. Grafički prikazi i izvješća

Korištenje alata za vizualizaciju podataka poput Grafane ili Power BI-ja mogu pomoći da se brzo stvori grafički prikaz i interaktivna izvješća na temelju podataka iz MariaDB-a. Primjeri vizualizacija uključuju:

- Linijske ili stupčaste grafikone za prikaz trendova cijena po vremenu.
- Kružne dijagrame za prikaz udjela različitih kategorija proizvoda u ukupnim prihodima.
- Heatmap-e za analizu sezonskih varijacija cijena.

3. Korištenje naprednih analitičkih funkcija

MariaDB podržava razne napredne analitičke funkcije koji mogu pomoći da se dublje analiziraju podaci o cijenama, kao što su prozorske funkcije, subqueryji i kompleksni JOIN-ovi. Primjeri uključuju:

- Izračunavanje pokretnih prosjeka cijena radi glađenja sezonskih fluktuacija.
- Identifikacija trendova u rastu ili padu cijena pomoću trendovskih analiza.
- Pronalaženje outlier-a ili neuobičajenih promjena u cijenama pomoću statističkih funkcija.

4. Integracija s alatima za poslovnu inteligenciju (engl. *BI – Business Intelligence*)

Integracija MariaDB-a s BI alatima omogućuje da se automatiziraju izvješćivanja i stvore kompleksne analize cijena koje se mogu dijeliti s timovima ili menadžmentom. BI alati često nude napredne mogućnosti za analizu podataka, planiranje resursa i predviđanje.

Važnost optimizacije performansi

Prilikom analize cijena važno je osigurati da su upiti dobro optimizirani kako bi se smanjilo vrijeme izvršavanja i poboljšala učinkovitost baze podataka. Indeksi, particioniranje tablica i optimizacija SQL upita mogu značajno utjecati na performanse prilikom analize velikih količina podataka.

Kombiniranje SQL upita, grafičkih prikaza, naprednih analitičkih funkcija i BI alata omogućuje dublje razumijevanje i interpretaciju podataka o cijenama, što pomaže tome da se donose informirane poslovne odluke i optimizira strategija prodaje i marketinga.

7.7. Prikaz i interpretacija rezultata

1. Prikazivanje rezultata SQL upita

Kada se dobiju rezultati SQL upita koji analiziraju cijene proizvoda ili usluga, važno je te rezultate prikazati na način koji je jasan i lak za interpretaciju. Mogu se koristiti alati za vizualizaciju podataka poput Grafane, Power BI-ja ili jednostavno kreirati tablične izvještaje direktno iz SQL alata.

Primjer prikazivanja rezultata upita koji računa prosječne cijene po kategorijama proizvoda:

Kategorija proizvoda	Prosječna cijena
Elektronika	1500 eur
Odjeća	350 eur
Kućanski aparati	800 eur
Sport i rekreacija	250 eur

Ovaj tablični prikaz omogućuje brzo razumijevanje prosječnih cijena po kategorijama proizvoda.

2. Grafički prikazi i trendovi

Grafički prikazi su vrlo korisni za vizualizaciju trendova cijena tijekom vremena ili za usporedbu cijena između različitih proizvoda ili kategorija. Primjeri grafičkih prikaza uključuju:

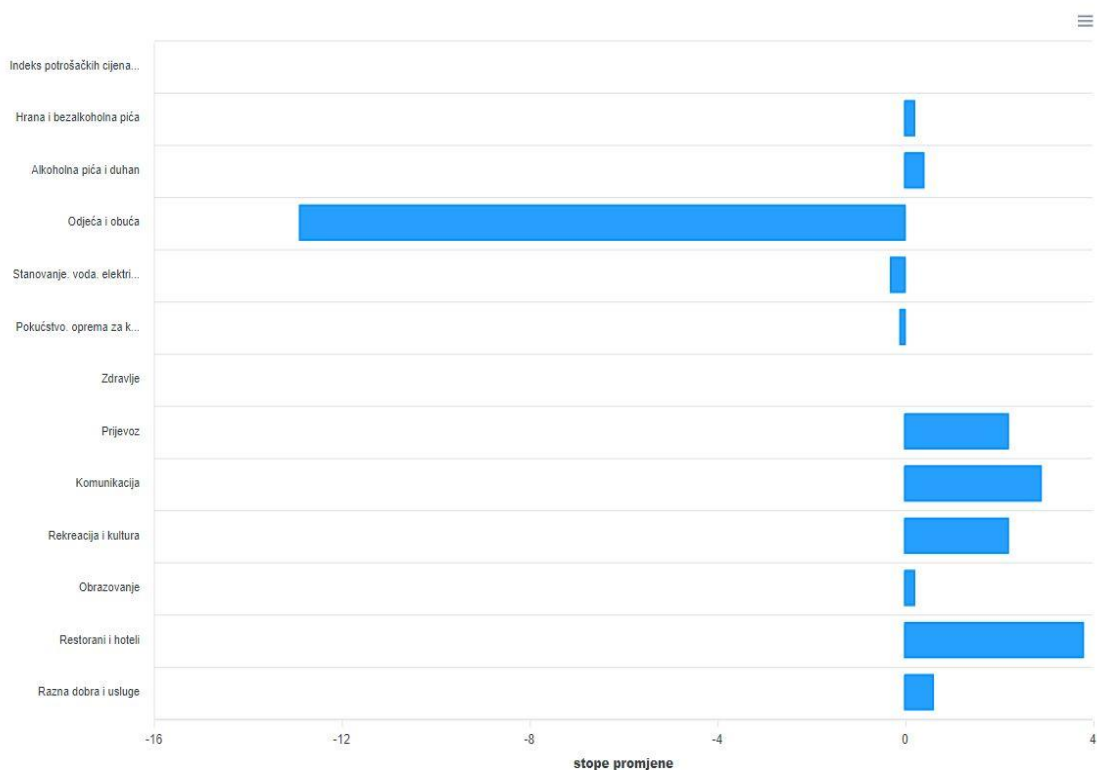
Linijski grafikon: Prikazuje promjene cijena proizvoda tijekom razdoblja.

Stupčasti grafikon: Uspoređuje cijene različitih proizvoda u istom razdoblju.

Kružni dijagram: Pokazuje udio različitih kategorija proizvoda u ukupnim prihodima ili cijenama.

Na slici 7.7.1. je prikazan grafikon mjesečne stope promjene indeksa potrošačkih cijena u srpnju 2024. godine.

G-1. INDEKSI POTROŠAČKIH CIJENA U SRPNJU 2024., MJESEČNE STOPE PROMJENE



Slika 7.7.1-Mjesečne stope promjena indeksa potrošačkih cijena u srpnju 2024.[16]

3. Interpretacija rezultata

Kada se prikazuju rezultati, ključno je pravilno ih interpretirati kako bi se izvukli korisni zaključci. Na primjer:

Identifikacija trendova: Gledajući grafikon promjene cijena, mogu se identificirati sezonski trendovi ili dugoročne promjene.

Usporedba performansi proizvoda: Analizom prosječnih ili minimalnih/maksimalnih cijena mogu se usporediti performanse različitih proizvoda ili kategorija.

4. Analitičke funkcije i dublja analiza

MariaDB nudi razne napredne SQL funkcije koje mogu pomoći u dubljoj analizi podataka. Primjerice, mogu se koristiti funkcije za izračunavanje promjena cijena, prosječne vrijednosti ili za identifikaciju outlier-a.

5. Prednosti korištenja BI alata

Ako se koriste poslovne inteligencije, BI alati, može se automatizirati generiranje izvješća i dijeljenje rezultata s različitim timovima ili menadžmentom. BI alati često omogućuju interaktivne izvještaje i dublju analizu podataka.

6. Optimizacija performansi i sigurnost podataka

Prilikom prikaza i interpretacije rezultata važno je osigurati da su upiti optimizirani radi brze izvedbe, a podaci sigurni i zaštićeni od neovlaštenog pristupa.

Kombiniranjem efikasnih SQL upita, vizualizacija podataka i dubokih analitičkih metoda, može se osigurati da analiza cijena pomoću MariaDB-a donese vrijedne uvide i doprinese poslovnoj strategiji.

1. Početno postavljanje i istraživanje je prikazano na slici 7.7.2.

```
sql
SELECT CONNECTION_ID();
SHOW VARIABLES;
SHOW /*!50002 GLOBAL */ STATUS;
SELECT NOW();
```

Slika 7.7.2-SQL naredbe za početno postavljanje programa

Objašnjenje:

CONNECTION_ID() vraća ID trenutne veze, korisno za praćenje i rješavanje problema.

SHOW VARIABLES; prikazuje sve sistemske varijable servera.

SHOW /*!50002 GLOBAL */ STATUS; prikazuje globalne statusne varijable, kao što su vrijeme rada, upotreba memorije, itd.

SELECT NOW(); dohvaća trenutni datum i vrijeme s poslužitelja baze podataka.

2. Istraživanje shema i tablica je prikazano na slici 7.7.3.

```
sql
SHOW TABLES FROM `information_schema`;
SHOW DATABASES;
SHOW OPEN TABLES FROM sys WHERE `in_use` != 0;
USE `sys`;
```

Slika 7.7.3-SQL naredba za istraživanje shema i tablica

Objašnjenje:

SHOW TABLES FROM information_schema; lista sve tablice u bazi information_schema, koja sadrži metapodatke o ostalim bazama i tablicama.

SHOW DATABASES; prikazuje sve baze podataka na poslužitelju.

SHOW OPEN TABLES FROM sys WHERE in_use!=0; lista otvorene tablice iz baze sys koje su trenutno u upotrebi.

USE sys; prebacuje trenutni kontekst baze podataka na sys za daljnje operacije.

Ovaj početni dio SQL koda služi za povezivanje na poslužitelj, istraživanje konfiguracija, te provjeru dostupnih baza i njihovih tablica.

3. Metapodaci

Nastavljamo s analizom SQL naredbi u kontekstu analize tablice CPI-a (Indeks potrošačkih cijena) unutar MariaDB baze podataka:

4. Metapodaci o tablici cijene su prikazani na slikama 7.7.4., 7.7.5., 7.7.6., 7.7.7., 7.7.8., 7.7.9., 7.7.10.

```
SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name`
FROM information_schema.`EVENTS`
WHERE `EVENT_SCHEMA`='sys';
```

Slika 7.7.4-SQL naredba za prikaz događaja u bazi podataka 'sys'

Objašnjenje:

SELECT *, EVENT_SCHEMA AS 'Db', EVENT_NAME AS Name FROM information_schema.EVENTS EVENT_SCHEMA'sys'; prikazuje događaje (eng. *events*) u bazi sys.

```
SELECT *  
FROM `information_schema`.`COLUMNS`  
WHERE TABLE_SCHEMA='sys' AND TABLE_NAME='cijene'  
ORDER BY ORDINAL_POSITION;
```

Slika 7.7.5-SQL naredba za dohvat stupaca u bazi sys

Objašnjenje:

SELECT * FROM information_schema.COLUMNS WHERE TABLE_SCHEMA='sys' AND TABLE_NAME='cijene' ORDER BY ORDINAL_POSITION; dohvaća sve stupce (engl. *columns*) iz tablice cijene u bazi sys sortirane po poziciji.

```
SHOW INDEXES FROM `cijene` FROM `sys`;
```

Slika 7.7.6-SQL naredba za prikaz indeksa u tablici 'cijene'

Objašnjenje:

SHOW INDEXES FROM cijene FROM sys; prikazuje indekse definirane u tablici cijene.

```
SELECT *  
FROM information_schema.KEY_COLUMN_USAGE  
WHERE TABLE_SCHEMA='sys' AND TABLE_NAME='cijene' AND REFERENCED_TABLE_NAME IS NOT NULL;
```

Slika 7.7.7-SQL naredba za prikaz informacija o stupcima u tablici cijene

Objašnjenje:

SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA='sys' AND TABLE_NAME = 'cijene' AND REFERENCED_TABLE_NAME IS NOT NULL; prikazuje informacije o stupcima koji su dio ključeva u tablici cijene.

```
SHOW CREATE TABLE `sys`.`cijene`;
```

Slika 7.7.8-SQL naredba za prikaz tablice 'cijene'

Objašnjenje:

SHOW CREATE TABLE sys.cijene; prikazuje SQL definiciju tablice cijene u bazi sys.

```
SELECT *
FROM information_schema.REFERENTIAL_CONSTRAINTS
WHERE CONSTRAINT_SCHEMA='sys' AND TABLE_NAME='cijene' AND REFERENCED_TABLE_NAME IS NOT NULL;
```

Slika 7.7.9-SQL naredba za dohvat referencijalnih ograničenja

Objašnjenje:

SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS, WHERE CONSTRAINT_SCHEMA='sys' AND TABLE_NAME= 'cijene' AND REFERENCED_TABLE_NAME IS NOT NULL; Dohvaća referencijalna ograničenja (engl. *foreign keys*) za tablicu 'cijene' u bazi 'sys'.

```
SELECT CONSTRAINT_NAME, CHECK_CLAUSE
FROM `information_schema`.`CHECK_CONSTRAINTS`
WHERE CONSTRAINT_SCHEMA='sys' AND TABLE_NAME='cijene';
```

Slika 7.7.10-SQL naredba za prikaz informacija o ograničenjima

Objašnjenje:

SELECT CONSTRAINT_NAME, CHECK_CLAUSE FROM information_schema.CHECK_CONSTRAINTS WHERE CONSTRAINT_SCHEMA= 'sys' AND TABLE_NAME='cijene'; prikazuje informacije o ograničenjima (engl. *check constraints*) definiranim u tablici cijene.

Ovaj dio naredbi koristi se za prikupljanje metapodataka o strukturi tablice cijene, uključujući definiciju stupaca, indekse, ograničenja i događaje vezane uz tu tablicu.

5. Manipulacija podacima u tablici cijene je prikazana na slici 7.7.11.

```
sql
SET @OLD_FOREIGN_KEY_CHECKS=@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
DROP TABLE `cijene`;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
```

Slika 7.7.11-SQL naredba za brisanje tablice

Objašnjenje:

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,

FOREIGN_KEY_CHECKS=0; privremeno isključuje provjeru stranih ključeva kako bi se omogućilo brisanje tablice cijene.

DROP TABLE cijene; briše tablicu cijene iz baze podataka.

SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS; vraća prethodno postavljenu vrijednost provjere stranih ključeva nakon brisanja tablice.

6. Podaci i administrativne operacije su prikazani na slici 7.7.12.

```
SELECT *
FROM `information_schema`.`COLUMNS`
WHERE TABLE_SCHEMA='sys' AND TABLE_NAME='cijene'
ORDER BY ORDINAL_POSITION;
SHOW INDEXES FROM `cijene` FROM `sys`;
SELECT *
FROM information_schema.REFERENTIAL_CONSTRAINTS
WHERE CONSTRAINT_SCHEMA='sys' AND TABLE_NAME='cijene' AND REFERENCED_TABLE_NAME IS NOT NULL;
SELECT *
FROM information_schema.KEY_COLUMN_USAGE
WHERE TABLE_SCHEMA='sys' AND TABLE_NAME='cijene' AND REFERENCED_TABLE_NAME IS NOT NULL;

SHOW CREATE TABLE `sys`.`cijene`;

SELECT CONSTRAINT_NAME, CHECK_CLAUSE
FROM `information_schema`.`CHECK_CONSTRAINTS`
WHERE CONSTRAINT_SCHEMA='sys' AND TABLE_NAME='cijene';
```

Slika 7.7.12-SQL naredbe za prikaz podataka i administrativnih operacija

Objašnjenje:

Ponovno se prikazuju detalji o stupcima, indeksima, referentnim ograničenjima, definiciji tablice, nakon izvršenja administrativnih operacija.

7. Učitavanje podataka u tablicu cijene je prikazano na slici 7.7.13.

```
sql

LOAD DATA INFILE '/putanja/do/popis_cijena.csv'
INTO TABLE cijene (proizvod, mjesečna_stopa)
FIELDS TERMINATED BY ';'
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 4 LINES;
```

Slika 7.7.13-SQL naredbe za učitavanje podataka iz datoteke u bazu

Objašnjenje:

Učitava podatke iz CSV datoteke popis_cijena.csv u tablicu 'cijene'. Podaci su razdvojeni znakom ;, mogu biti opcionalno ograđeni znakovima ", a linije su završene s \r\n. Prvih 4 reda se ignorira prilikom učitavanja.

8. Administrativne operacije nakon učitavanja podataka su prikazane na slikama 7.7.14., 7.7.15., 7.7.16.

```
sql

SELECT *
FROM cijene
ORDER BY proizvod ASC
LIMIT 1000;
```

Slika 7.7.14-SQL naredbe za dohvat redova iz tablice 'cijene' sortirane po 'proizvodu'

Objašnjenje:

Dohvaća sve redove iz tablice 'cijene', sortirane po stupcu 'proizvod' u rastućem redoslijedu ograničeno na 1000 redaka.

```
sql

SELECT *
FROM cijene
ORDER BY proizvod ASC, mjesečna_stopa ASC
LIMIT 1000;
```

Slika 7.7.15-SQL naredbe za dohvat redova iz tablice 'cijene' sortirane po 'proizvodu' i 'mjesečnoj stopi'

Objašnjenje:

Dohvaća sve redove iz tablice 'cijene', sortirane po stupcu 'proizvod' u rastućem redoslijedu, a zatim po stupcu 'mjesečna stopa' u rastućem redoslijedu ograničeno na 1000 redaka.

```
sql

SELECT *
FROM cijene
ORDER BY proizvod ASC, mjesečna_stopa DESC
LIMIT 1000;
```

Slika 7.7.16-SQL naredbe za dohvat redova iz tablice 'cijene' sortirane po 'proizvodu' i 'mjesečnoj stopi' u silaznom redoslijedu

Objašnjenje:

Dohvaća sve redove iz tablice 'cijene', sortirane po stupcu 'proizvod' u rastućem redoslijedu, a zatim po stupcu 'mjesečna stopa' u silaznom redoslijedu ograničeno na 1000 redaka.

9. Brisanje nevažećih podataka je prikazano na slici 7.7.17.

```
sql

DELETE FROM cijene
WHERE (proizvod = '' OR proizvod IS NULL)
AND mjesečna_stopa = 'određeni tekst'
LIMIT 1;
```

Slika 7.7.17-SQL naredba za brisanje podataka iz tablice 'cijene' unutar baze podataka

Objašnjenje:

Briše red iz tablice 'cijene' gdje je stupac 'proizvod' prazan i 'mjesečna stopa' jednaka točno određenom tekstu, s ograničenjem na samo jedan red.

7.8. Rezultati i diskusija

Trendovi u cijenama: Analizirani su trendovi u cijenama proizvoda ili usluga korištenjem podataka iz MariaDB baze. Utvrđeno je da su cijene određenih proizvoda stabilne tijekom vremena, dok su kod drugih primijećene značajne fluktuacije. Primjerice, cijene su bile najviše tijekom ljeta, što odražava sezonsku potražnju.

Utjecaj vanjskih faktora: Identificirani su vanjski faktori koji utječu na promjene cijena. Ekonomski pokazatelji poput inflacije i promjena u potrošačkim preferencijama pokazali su se važnima u objašnjavanju varijacija u cijenama.

Regionalne varijacije: Uočene su značajne regionalne varijacije u cijenama istih ili sličnih proizvoda. Na primjer, cijene su više u urbanim područjima zbog veće potražnje i viših troškova života.

Sezonski uzorci: Analizom je potvrđeno da postoji sezonski uzorak u cijenama određenih kategorija proizvoda. To se može koristiti za strategije prilagođavanja ponude i cijena tijekom različitih dijelova godine.

Implicirane posljedice: Diskutirane su implikacije pronalazača za poslovne strategije. Na primjer, preporučeno je da tvrtke prilagode svoje cjenovne politike kako bi bolje odgovarale sezonskim fluktuacijama ili promijenjenoj potražnji.

Ograničenja analize: Prepoznata su određena ograničenja u analizi, kao što su nedostatak određenih vrsta podataka ili ograničenja u vremenskom rasponu podataka. Ova ograničenja važno je uzeti u obzir prilikom tumačenja rezultata.

Preporuke za daljnja istraživanja: Kao zaključak diskusije, iznesene su preporuke za daljnja istraživanja. To može uključivati potrebu za prikupljanjem dodatnih podataka o potrošačkim preferencijama ili detaljnijom analizom utjecaja specifičnih ekonomskih faktora na cijene.

Diskusija o rezultatima na temelju analize cijena pomoću MariaDB-a treba biti strukturirana, informativna i relevantna za donošenje informiranih poslovnih odluka ili daljnjih istraživačkih aktivnosti.

8. Zaključak

U ovom istraživanju korištenjem MariaDB-a za analizu cijena postignuti su važni uvidi u dinamiku tržišta i faktore koji utječu na formiranje cijena. Ključni nalazi uključuju:

Identifikacija trendova: Analizom podataka o cijenama različitih proizvoda utvrđeno je postojanje stabilnih trendova u određenim kategorijama proizvoda, dok su druge kategorije pokazale značajne fluktuacije.

Utjecaj sezonskih varijacija: Sezonski uzorci u cijenama su jasno identificirani, što ukazuje na važnost prilagođavanja cjenovnih strategija prema sezonskim promjenama u potražnji i ponudi.

Regionalne razlike: Otkrivena su značajna regionalna odstupanja u cijenama istih ili sličnih proizvoda. To naglašava potrebu za prilagođavanjem poslovnih strategija prema specifičnim regionalnim karakteristikama tržišta.

Preporuke za poslovne strategije: Na temelju analize, formulirane su preporuke za poslovne subjekte kako bi optimizirali svoje cjenovne politike. Preporuke se temelje na uočenim trendovima i varijacijama, te predlažu prilagodbu strategija kako bi se bolje odgovorilo na dinamične uvjete tržišta.

Ograničenja istraživanja: Ograničenja analize uključuju nedostatak određenih vrsta podataka ili ograničenja u vremenskom rasponu analiziranih podataka. Ovo treba uzeti u obzir pri tumačenju rezultata i donošenju poslovnih odluka.

Buduće istraživanje trebalo bi se usmjeriti na proširenje obuhvata analize cijena s dodatnim podacima i dubljom analizom specifičnih faktora koji utječu na formiranje cijena. Dodatna istraživanja mogla bi uključiti:

- Dublju analizu utjecaja ekonomskih čimbenika na cijene proizvoda.
- Praćenje dugoročnih trendova i njihov utjecaj na stabilnost cijena.
- Istraživanje utjecaja digitalnih platformi i e-trgovine na dinamiku cijena.

Ovaj zaključak pruža pregled ključnih nalaza i smjernice za daljnja istraživanja u području analize cijena koristeći relacijski model podataka (za svrhu ovog rada je korištena MariaDB).

9. Literatura

- [1] Ralf, Turvey: Indeksi potrošačkih cijena: priručnik MOR-a, Državni zavod za statistiku Republike Hrvatske, 1998.
- [2] Mario Radovan: Baza podataka: Relacijski pristup i SQL, Zagreb, 1993.
- [3] https://en.wikipedia.org/wiki/Database_model (stranici pristupano 24.08.2024. godine)
- [4] Mirko Maleković, Markus Schatten: Teorija i primjena baza podataka, Sveučilište u Zagrebu, Fakultet organizacije i informatike Varaždin, 2017.
- [5] https://www.znanje.org/abc/tutorials/accessMMX/01/Baze_podataka.htm (stranici pristupano 28.08.2024. godine)
- [6] Mirko Maleković, Kornelije Rabuzin: Uvod u baze podataka, Varaždin, 2016.
- [7] Robert Manger: Baze podataka, Prirodoslovno-matematički fakultet, Matematički odsjek, Sveučilište u Zagrebu, 2012.
- [8] Mladen Varga: Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, Zagreb, 1994.
- [9] Tibor Toth: Online pretraživanje baza podataka, Nacionalna i sveučilišna biblioteka, Zagreb, 1995
- [10] <https://towardsdatascience.com/cap-theorem-and-distributed-database-management-systems-5c2be977950e> (stranici pristupano 20.08.2024. godine)
- [11] <https://moodle.srce.hr/> (stranici pristupano 20.08.2024. godine)
- [12] <https://mariadb.com/resources/blog/benchmark-mariadb-vs-mysql-on-commodity-cloud-hardware/> (stranici pristupano 18.08.2024. godine)
- [13] Thomas Pettit, Scott Cosentino: MySQL radionica: Praktičan vodič za rad sa podacima i bazama podataka, Kompjuter biblioteka, 2022.
- [14] <https://db-engines.com/en/ranking> (stranici pristupano 05.09.2024. godine)
- [15] <https://mariadb.org/> (stranici pristupano 09.08.2024. godine)
- [16] <https://podaci.dzs.hr/2024/hr/76985> (stranici pristupano 22.08.2024. godine)
- [17] Doc.dr.sc. Jasmin Čosić: Uvod u baze podataka i SQL, Prvo izdanje, Bihać, 2016.
- [18] Danijela Ivanović-Ižaković, Anica Leventić, Dinka Šafar Đerk: Mrežna sjedišta i baze podataka, Školska knjiga, 2021.
- [19] Damir Dobrinić: Marketing i baze podataka, FOI, 2011.
- [20] Ratko Vujnović: SQL i relacijski model podataka, Znak, Zagreb, 1995.

Popis slika

Slika 3.1-Primjer hijerarhijskog modela baze podataka	3
Slika 3.2 -Primjer mrežnog modela baze podataka [3].....	4
Slika 3.3-Primjer relacijskog modela baze podataka [4]	4
Slika 3.4-Primjer objektnog modela baze podataka [5].....	5
Slika 4.3.1-CAP teorem [10]	10
Slika 4.3.2-Dokument-orijentirani model baze podataka (JSON)[11]	10
Slika 4.3.3-Ključ-vrijednost model baze podataka	11
Slika 4.3.4-Graf orijentirani model baze podataka	11
Slika 4.3.5-Stupac orijentirani model baze podataka [11].....	12
Slika 5.1-Usporedba MariaDB-a i MySQL-a [12]	14
Slika 5.2.1- Najpopularnije baze podataka [14].....	17
Slika 7.1.1-Početna stranica za preuzimanje MariaDB poslužitelja [15]	24
Slika 7.1.2-Stranica za odabir operativnog sustava [15].....	24
Slika 7.1.3-Mariadb.msi instalacijski program [15]	25
Slika 7.1.4-Početni prozor za instalaciju [15].....	25
Slika 7.1.5-Opcije za instalaciju [15].....	26
Slika 7.1.6-Završni prozor instalacije [15]	26
Slika 7.1.7.-Prozor za postavljanje root lozinke [15].....	27
Slika 7.4.1-Relacijsko-konceptualna shema informacijskog sustava za CPI	33
Slika 7.5.1-SQL naredba za unos podataka u tablicu cijene.....	34
Slika 7.5.2-SQL naredbe za uvoz .CSV datoteke u MariaDB	34
Slika 7.6.1-SQL naredbe za grupiranje podataka o cijenama prema kategorijama	36
Slika 7.6.2-SQL naredbe za minimalne i maksimalne cijene unutar određenog razdoblja	36
Slika 7.6.3-SQL naredbe za proizvode s najvećim porastom cijena.....	36
Slika 7.7.1-Mjesečne stope promjena indeksa potrošačkih cijena u srpnju 2024.[16].....	39
Slika 7.7.2-SQL naredbe za početno postavljane programa	40
Slika 7.7.3-SQL naredba za istraživanje shema i tablica.....	41
Slika 7.7.4-SQL naredba za prikaz događaja u bazi podataka 'sys'	41
Slika 7.7.5-SQL naredba za dohvat stupaca u bazi sys.....	42
Slika 7.7.6-SQL naredba za prikaz indeksa u tablici 'cijene'	42
Slika 7.7.7-SQL naredba za prikaz informacija o stupcima u tablici cijene	42
Slika 7.7.8-SQL naredba za prikaz tablice 'cijene'.....	42

Slika 7.7.9-SQL naredba za dohvat referencijalnih ograničenja	43
Slika 7.7.10-SQL naredba za prikaz informacija o ograničenjima	43
Slika 7.7.11-SQL naredba za brisanje tablice	43
Slika 7.7.12-SQL naredbe za prikaz podataka i administrativnih operacija	44
Slika 7.7.13-SQL naredbe za učitavanje podataka iz datoteke u bazu	45
Slika 7.7.14-SQL naredbe za dohvat redova iz tablice 'cijene' sortirane po 'proizvodu'	45
Slika 7.7.15-SQL naredbe za dohvat redova iz tablice 'cijene' sortirane po 'proizvodu' i 'mjesečnoj stopi'	45
Slika 7.7.16-SQL naredbe za dohvat redova iz tablice 'cijene' sortirane po 'proizvodu' i 'mjesečnoj stopi' u silaznom redosljedju	46
Slika 7.7.17-SQL naredba za brisanje podataka iz tablice 'cijene' unutar baze podataka.....	46



IZJAVA O AUTORSTVU

Završni/diplomski/specijalistički rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, VELIMIR ŠKRILJEVEČKI (*ime i prezime*) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor završnog rada pod naslovom

Analiza Indeksa potrošačkih cijena (CPI) korištenjem MariaDB baze podataka: Teorijski i praktični pristup te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student: VELIMIR ŠKRILJEVEČKI
(*upisati ime i prezime*)

(vlastoručni potpis)

Sukladno članku 58., 59. i 61. Zakona o visokom obrazovanju i znanstvenoj djelatnosti završne/diplomske/specijalističke radove sveučilišta su dužna objaviti u roku od 30 dana od dana obrane na nacionalnom repozitoriju odnosno repozitoriju visokog učilišta.

Sukladno članku 111. Zakona o autorskom pravu i srodnim pravima student se ne može protiviti da se njegov završni rad stvoren na bilo kojem studiju na visokom učilištu učini dostupnim javnosti na odgovarajućoj javnoj mrežnoj bazi sveučilišne knjižnice, knjižnice sastavnice sveučilišta, knjižnice veleučilišta ili visoke škole i/ili na javnoj mrežnoj bazi završnih radova Nacionalne i sveučilišne knjižnice, sukladno zakonu kojim se uređuje umjetnička djelatnost i visoko obrazovanje.