SVEUČILIŠTE U SPLITU

**PRIRODOSLOVNO MATEMATIČKI FAKULTET**

PROJEKT

**USPOREDBA MSSQL RELACIJSKE BAZE PODATAKA SA MONGODB NERELACIJSKOM BAZOM PODATAKA**

Marko Bonovil

Split, lipanj 2023

Sadržaj

[1. Uvod 1](#_Toc138072618)

[2. Izrada modela baze podataka u MSSQL-u 2](#_Toc138072619)

[2.1. Korisnički zahtjevi 2](#_Toc138072620)

[2.2. Model baze podataka 2](#_Toc138072621)

[2.3. Entiteti 3](#_Toc138072622)

[2.4. Veze 6](#_Toc138072623)

[2.5. Kreiranje baze 7](#_Toc138072624)

[2.5.1. Stvaranje baze podataka, tablica i povezivanje 7](#_Toc138072625)

[2.5.2. SQL dijagram baze podataka 9](#_Toc138072626)

[2.5.3. Generiranje podataka i unos u tablice 9](#_Toc138072627)

[3. MongoDB 11](#_Toc138072628)

[3.1. Razlika između SQL i MongoDB baze podataka 11](#_Toc138072629)

[3.2. Kolekcija 11](#_Toc138072630)

[3.3. Dokument 12](#_Toc138072631)

[3.4. Model podataka za MongoDB 12](#_Toc138072632)

[4. Izrada modela baze podataka u MongoDB 13](#_Toc138072633)

[4.2. Unos podataka u kolekcije 13](#_Toc138072634)

[5. Usporedba brzine izvršavanja zahtjeva SQL i MongoDB baze podataka 15](#_Toc138072635)

[5.1. Izvršavanje upita 15](#_Toc138072636)

[5.1.1. Dohvaćanje svih zapisa 15](#_Toc138072637)

[5.1.2. Dodavanje zapisa u baze 16](#_Toc138072638)

[5.1.3. Uklanjanje zapisa iz baze 16](#_Toc138072639)

[5.1.4. Sortiranje podataka 17](#_Toc138072640)

[5.1.5. Operacije agregacije 18](#_Toc138072641)

[5.1.6. Ostala testiranja 18](#_Toc138072642)

[6. Zaključak 21](#_Toc138072643)

# Uvod

Razvoj tehnologije svakodnevno sve više omogućava visoku razinu povezanosti i dostupnostiinformacijama na globalnoj razini te istovremeno predstavlja problem pohranjivanja i pristupanja istim informacijama. Zbog toga se pojavljuje potreba za implementacijom baza podataka, koje su kolekcije informacija i atributa koji ih opisuju. Baze podataka su sveprisutna tehnologija koja se koristi za organizaciju podataka u velikim i malim institucijama, obradama kreditnih kartica, pohranu elektroničke pošte i na kraju krajeva, u čuvanju više-manje svih podataka na Internetu. Najčešća podjela baza podataka se svodi na dva glavna tipa; relacijske baze podataka i ne-relacijske ili baze podataka bez slijeda.

U ovom radu će se govoriti o prednostima i nedostacima jedne on ne-relacijskih baza podataka, to jest dokumentno orijentiranih baza podataka, naspram relacijskih baza podataka. Rad je podijeljen na šest poglavlja: uvod, izrada modela baze podataka u MSSQL-u, MongoDB, Izrada modela baze podataka u MongoDB-u, usporedba brzine izvršavanja zahtjeva SQL i MongoDB baze podataka i zaključak.

U drugom poglavlju opisuje se postupak izrade baze podataka u SQL-u na koji način je kreirana baza koji su entiteti te sa kakvim podacima je popunjena. U trećem poglavlju opisuju se glavne razlike između MongoDB dokumentne baze podataka i relacijskih baza podataka. U četvrtom poglavlju opisuje se princip izrade baze podataka u MongoDB dokumentnoj bazi, a u petom poglavlju opisani su zahtjevi koji su korišteni za mjerenje brzine izvršavanja upita te je prikazano potrebno vrijeme.

# Izrada modela baze podataka u MSSQL-u

## Korisnički zahtjevi

Kao zadatak uzeta je izrada modela baze podataka za umjetničku galeriju. Potrebno je voditi evidenciju o umjetninama koje su izložene u galeriji kao i kreatorima tih umjetničkih dijela. Svako umjetničko djelo ima motive te se može interpretirati na različite načine. Pošto su izložbe javne potrebno je voditi evidenciju o posjetiteljima koji prisustvuju izložbi, te ujedno imaju mogućnost kupnje ili najma umjetničkog dijela. Svaki oblik kupnje ili najma treba biti zabilježen u bazi, stoga se iz postojeće evidencije očekuje da se mogu dobiti sljedeći izvještaji:

* Popis svih umjetnika koji sudjeluju na izložbi.
* Popis svih umjetnina koje su izložene na izložbi.
* Popis svih posjetitelja koji prisustvuju izložbi.
* Popis svih motiva koje karakteriziraju umjetničko djelo.
* Pregled svih dodijeljenih oznaka pojedinom umjetničkom dijelu na izložbi.
* Popis svih izložbi koje su se odvijale u galeriji te otkada do kada su trajale.
* Popis svih oblika zakupa pojedine umjetnine.
* Pregled svih obavljenih transakcija za umjetnička dijela.

## Model baze podataka

U nastavku je prikazan ER dijagram i konačni izgled zamišljene baze. Dijagram je kreirano pomoću internetske stranice *dbdiagram.io* ( <https://dbdiagram.io/home> )

Slika na kojoj se prikazuje tekst, snimka zaslona, dijagram, broj

Opis je automatski generiran

Slika 1 Konačan izgled baze podataka

Slika na kojoj se prikazuje tekst, dijagram, paralelno, dizajn

Opis je automatski generiran

Slika 2 ER-dijagram baze podataka

Baza je rađena na način da zadovolji sve korisničke zahtjeve, a u nastavku su detaljno opisani entiteti korišteni u realizaciji baze te opis veze između njih.

## Entiteti

PostanskiBroj

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUT | KLJUČ | TIP | PRIMJER UPISA | OGRANIČENJA | PRIMJEDBA |
| zip\_id | PK | int NOT NULL | 1 |  |  |
| grad |  | varchar(100) | Split |  |  |
| drzava |  | varchar(100) | Hrvatska |  |  |

Tablica 1 Entitet PostanskiBroj

Tip

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUT | KLJUČ | TIP | PRIMJER UPISA | OGRANIČENJA | PRIMJEDBA |
| tip\_id | PK | int NOT NULL | 1 |  |  |
| naziv |  | varchar(100) | Kupnja |  |  |

Tablica 2 Entitet Tip

Umjetnik

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUT | KLJUČ | TIP | PRIMJER UPISA | OGRANIČENJA | PRIMJEDBA |
| umjetnik\_id | PK | int NOT NULL | 1 |  |  |
| ime |  | varchar(100) | Ante |  |  |
| prezime |  | varchar(100) | Antić |  |  |
| datum\_rodenja |  | date | 1976-11-26 |  |  |
| adresa |  | varchar(100) | Matice hrvatske 13 |  |  |
| zip | FK | Int NOT NULL | 21000 |  |  |
| broj\_mobitela |  | varchar(100) | 09123456 |  |  |

Tablica 3 Entitet Umjetnik

Umjetnina

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUT | KLJUČ | TIP | PRIMJER UPISA | OGRANIČENJA | PRIMJEDBE |
| umjetnina\_id | PK | int NOT NULL | 1 |  |  |
| godina |  | int | 2023 |  |  |
| naslov |  | varchar(100) | Slika grada |  |  |
| cijena |  | int | 2000 |  |  |
| u\_id | FK | int NOT NULL | 2 |  |  |

Tablica 4 Entitet Umjetnina

Motiv

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUT | KLJUČ | TIP | PRIMJER UPISA | OGRANIČENJA | PRIMJEDBA |
| Motiv\_id | PK | int NOT NULL | 1 |  |  |
| Naziv |  | varchar(100) | Akt |  |  |

Tablica 5 Entitet Motiv

Izlozba

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUT | KLJUČ | TIP | PRIMJER UPISA | OGRANIČENJA | PRIMJEDBA |
| izlozba\_id | PK | int NOT NULL | 1 |  |  |
| ime\_galerije |  | varchar(100) | Galerija Meštrović |  |  |
| datum\_početka |  | date | 2023-06-23 |  |  |
| datum\_završetka |  | date | 2023-06-29 |  |  |
| adresa |  | varchar(100) | Šetalište Ivana Meštrovića |  |  |
| zip | FK | int NOT NULL | 21000 |  |  |

Tablica 6 Entitet Izlozba

Posjetitelj

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUT | KLJUČ | TIP | PRIMJER UPISA | OGRANIČENJA | PRIMJEDBA |
| posjetitelj\_id | PK | int NOT NULL | 1 |  |  |
| ime |  | varhcar(100) | Ante |  |  |
| prezime |  | varchar(100) | Antić |  |  |
| broj\_mobitela |  | varchar(100) | 09123456 |  |  |
| adresa |  | varchar(100) | Matice hrvatske 13 |  |  |
| zip | FK | int NOT NULL | 21000 |  |  |

Tablica 7 Entitet Posjetitelj

Umjetnina\_Motiv

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUT | KLJUČ | TIP | PRIMJER UPISA | OGRANIČENJA | PRIMJEDBA |
| Um\_id | FK | int NOT NULL | 1 |  |  |
| m\_id | FK | int NOT NULL | 1 |  |  |
| interpretacija |  | varchar(255) | Opis prirode na specifičan način |  |  |

Tablica 8 Entitet Umjetnina\_Motiv

Stavka

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ATRIBUT | KLJUČ | TIP | PRIMJER UPISA | OGRANIČENJA | PRIMJEDBA |
| stavka\_id | PK | int NOT NULL | 1 |  |  |
| t\_id | FK | int NOT NULL | 1 |  |  |
| datum\_dostave |  | date | 2023-10-05 |  |  |
| datum\_kraja |  | date | 2024-10-05 |  |  |
| cijena |  | int | 2000 |  |  |
| adresa\_dostave |  | varchar(100) | Matice hrvatske 13 |  |  |
| zip | FK | int NOT NULL | 21000 |  |  |
| u\_id | FK | int NOT NULL | 1 |  |  |
| p\_id | FK | int NOT NULL | 1 |  |  |

Tablica 9 Entitet Stavka

## Veze

ER-dijagram izveden je prema sljedećem principu:

* Jedan umjetnik može imati jedan poštanski broj, dok jedan poštanski broj može imati više umjetnika.
* Jedan umjetnik može imati jednu ili više umjetničkih djela, dok jedno umjetničko djelo može imati jednog i samo jednog umjetnika.
* Jedna umjetnina može imati jednu ili više izložbu, dok jedna izložba može imati jednu ili više umjetninu.
* Jedna umjetnina može imati jedan ili više motiv, dok jedan motiv može imati jedna ili više umjetnina.
* Jedan posjetitelj ima jedan poštanski broj, dok jedan poštanski broj može imati više posjetitelja.
* Jedna izložba ima jedan poštanski broj, dok jedan poštanski broj može imati više izložbi.
* Jedan tip (odabir između kupnje ili najma) može biti u više stavki, dok jedna stavka ima samo jedan tip.
* Jedan posjetitelj može imati više stavki, dok jedna stavka se odnosi samo na jednog posjetitelja.
* Jedna umjetnina može imati više stavki, dok jedna stavka se odnosi na samo jednu umjetninu.
* Jedna stavka može imati jedan poštanski broj, dok jedan poštanski broj može imati jednu ili više stavki.

S obzirom da su definirani entiteti za kreiranje baze podatak kao i način povezivanja entiteta sljedeći korak predstavlja stvaranje SQL baze podataka.

## Kreiranje baze

### Stvaranje baze podataka, tablica i povezivanje

CREATE TABLE Umjetnik(

umjetnik\_id int NOT NULL PRIMARY KEY,

ime varchar(100),

prezime varchar(100),

datum\_rodenja date,

adresa varchar(100),

zip int NOT NULL,

broj\_mobitela varchar(100),

);

CREATE TABLE PostanskiBroj(

zip\_id int NOT NULL PRIMARY KEY,

grad varchar(100),

drzava varchar(100)

);

CREATE TABLE Posjetitelj(

posjetitelj\_id int NOT NULL PRIMARY KEY,

ime varchar(100),

prezime varchar(100),

broj\_mobitela varchar(100),

adresa varchar(100),

zip int NOT NULL

);

CREATE TABLE Izlozba(

izlozba\_id int NOT NULL PRIMARY KEY,

ime\_galerije varchar(100),

datum\_pocetka date,

datum\_zavrsetka date,

adresa varchar(100),

zip int NOT NULL

);

CREATE TABLE Umjetnina\_Izlozba(

iz\_id int NOT NULL,

um\_id int NOT NULL,

broj\_umjetnine int

);

CREATE TABLE Tip(

tip\_id int NOT NULL PRIMARY KEY,

naziv varchar(100)

);

CREATE TABLE Stavka(

stavka\_id int NOT NULL PRIMARY KEY,

t\_id int NOT NULL,

datum\_dostave date,

datum\_kraja date,

cijena int,

adresa\_dostave varchar(100),

zip int NOT NULL,

u\_id int,

p\_id int

);

CREATE TABLE Umjetnina(

umjetnina\_id int NOT NULL PRIMARY KEY,

godina int,

naslov varchar(100),

cijena int,

u\_id int NOT NULL

);

CREATE TABLE Motiv(

motiv\_id int NOT NULL PRIMARY KEY,

naziv varchar(100)

);

CREATE TABLE Umjetnina\_Motiv(

um\_id int NOT NULL,

m\_id int NOT NULL,

interpretacija varchar(255)

);

ALTER TABLE Umjetnina\_Izlozba

ADD FOREIGN KEY (iz\_id) REFERENCES Izlozba(izlozba\_id);

ALTER TABLE Umjetnina\_Izlozba

ADD FOREIGN KEY (um\_id) REFERENCES Umjetnina(umjetnina\_id);

ALTER TABLE Umjetnik

ADD FOREIGN KEY (zip) REFERENCES PostanskiBroj(zip\_id);

ALTER TABLE Umjetnina\_Motiv

ADD FOREIGN KEY (um\_id) REFERENCES Umjetnina(umjetnina\_id);

ALTER TABLE Umjetnina\_Motiv

ADD FOREIGN KEY (m\_id) REFERENCES Motiv(motiv\_id);

ALTER TABLE Izlozba

ADD FOREIGN KEY (zip) REFERENCES PostanskiBroj(zip\_id);

ALTER TABLE Umjetnina

ADD FOREIGN KEY (u\_id) REFERENCES Umjetnik(umjetnik\_id);

ALTER TABLE Posjetitelj

ADD FOREIGN KEY (zip) REFERENCES PostanskiBroj(zip\_id);

ALTER TABLE Stavka

ADD FOREIGN KEY (p\_id) REFERENCES Posjetitelj(posjetitelj\_id);

ALTER TABLE Stavka

ADD FOREIGN KEY (u\_id) REFERENCES Umjetnina(umjetnina\_id);

ALTER TABLE Stavka

ADD FOREIGN KEY (zip) REFERENCES PostanskiBroj(zip\_id);

ALTER TABLE Stavka

ADD FOREIGN KEY (t\_id) REFERENCES Tip(tip\_id);

### SQL dijagram baze podataka

Nakon stvaranja entiteta i uređivanje veza između njih kreiran je dijagram baze podataka Umjetnička galerija pomoću SQL-a koji je prikazan na slici 3.

Slika na kojoj se prikazuje tekst, dijagram, Plan, tehničko crtanje

Opis je automatski generiran

Slika 3 SQL dijagram

### Generiranje podataka i unos u tablice

Obzirom da je cilj projekta usporedba vremena izvršavanja zahtjeva između relacijskih i ne relacijskih baza podataka, sukladno tome potrebno je baze napuniti sa velikom količinom podataka kako bi dobili jasnu predodžbu o tome koliko je vremena zapravo potrebno za dobivanje rezultate upita. U tu svrhu korišten je internetski stranica *ExtendsClass* koja omogućava generiranje velike količine podataka. Postoje i brojne druge stranice sa sličnom primjenom kao što su *Generate data*, *mockaroo* i sl. međutim, imaju ograničenu veličinu podataka koje generiraju. Tako *Generate data* može maksimalno generirati 500 linija, dok *mockaroo* može 1000 linija. Pomoću *ExtendsClass* (<https://extendsclass.com/csv-generator.html>) internetske stranice generirano je 100 000 linija u SQL kodu za sve tablice u bazi podataka, osim tablica Tip(2) i Motiv(50), a sučelje stranice *ExtendsClass* prikazano je na slici 4 dok je na slici 5 prikazan oblik u kojem su podaci generirani.

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, Font

Opis je automatski generiran

Slika 4 Sučelje internetske stranice ExtendsClass

Slika na kojoj se prikazuje tekst, Font, snimka zaslona

Opis je automatski generiran

Slika 5 Primjer generiranih podataka

Generirani podaci uneseni se u bazu podataka, a u sljedećoj tablici prikazano je vrijeme koje je bilo potrebno da se podaci unesu u predodređene tablice.

|  |  |
| --- | --- |
| NAZIV | VRIJEME |
| POŠTANSKI BROJ | **[00:01:10]** |
| UMJETNIK | **[00:01:12]** |
| UMJETNINA | **[00:01:57]** |
| IZLOŽBA | **[00:01:39]** |
| POSJETITELJ | **[00:01:25]** |
| STAVKA | **[00:01:40]** |
| TIP | **[00:00:01]** |
| UMJETNINA IZLOŽBA | **[00:01:13]** |
| MOTIV | **[00:00:02]** |
| UMJETNINA MOTIV | **[00:01:15]** |

Tablica 10 Vrijeme potrebno za unos podataka u SQL bazu

# MongoDB

Izrada baze podataka u MongoDB-u i SQL-u ima svoje specifičnosti i pristupe. U MongoDB-u, podaci se pohranjuju kao dokumenti u kolekcijama, dok se u SQL-u koriste tablice s relacijama između njih. Oba pristupa imaju svoje prednosti i nedostatke, a izbor između njih ovisi o specifičnim zahtjevima projekta i prirodi podataka koje je potrebno pohraniti. U nastavku su detaljnije opisane glavne razlike između SQL baze podataka i MongoDB baze.

## Razlika između SQL i MongoDB baze podataka

Mongo spada u dokument orijentirane NoSQL baze podataka, otvorenog je koda, a jedan od najvažnijih razloga popularnosti MongoDB-a je taj što je JSON-friendly baza podataka. Sljedeća tablice prikazuje neke od osnovnih razlika između MongoDB i relacijskih baza podataka.

|  |  |
| --- | --- |
| RELACIJSKE BAZE PODATAKA | MONGODB |
| Baza | **Baza** |
| Tablica | **Kolekcija** |
| Red | **Dokument** |
| Stupac | **Polje (ključ)** |
| Join tablice | **Ugniježđeni dokumenti** |
| Primarni ključ | **Primarni ključ (zadani ključ \_id koji osigurava sam MongoDB)** |

Tablica 11 Osnovne razlike između Relacijskih baza podataka i MongoDB baze

## Kolekcija

Kolekcija u osnovi predstavlja grupu dokumenata, ona je analogna tablica iz relacijskih baza podataka. Dokumenti unutar kolekcije mogu imati različita polja , te su tipično svi dokumenti unutar kolekcije slične ili srodne namjene. Sudeći po pravilima dinamične sheme, sve dokumente možemo svrstati u jednu kolekciju, međutim iz više razloga bi bilo bolje da postoji više kolekcija, a neki od razloga su:

* Svrstavanje različitih dokumenata u kolekciju može napraviti pomutnju programerima zbog toga što je potrebno da svaki upit nad bazom vraća dokumente istog tipa jer bi se posebno trebalo odvojiti vrijeme za implementaciju dodatne logike što je nepotrebno.
* Prilikom upita jedne velike kolekcije, MongoDB-u bi trebalo više vremena da vrati jedan ili više podatak.
* Indeksi se kreiraju po kolekciji te ukoliko se svi dokumenti istog tipa svrstaju u jednu kolekciju, može se efikasnije indeksirati kolekcija.

## Dokument

MongoDB sadrži zapise podataka kroz BSON dokumente, a BSON predstavlja binarni prikaz JSON dokumenta. Dokument se odnosi na uređenu listu ključeva sa pripadajućim vrijednostima, a struktura dokumenta u različitim programskim jezicima varira. Pri generiranju dokumenta u MongoDB-u, kao polje dokumentu dodaje se primarni ključ \_id, svaki dokument pohranjen u kolekciji zahtjeva jedinstveno polje \_id koje djeluje kao primarni ključ, a u slučaju izostavljanja polja \_id pri pokretanju baze Mongo sam generira ObjectID za polje \_id.

## Model podataka za MongoDB

Podaci u MongoDB-u imaju fleksibilnu shemu za razliku od SQL baze podataka, gdje mora biti određena i deklarirana shema prije umetanja podataka, kolekcije MongoDB-a ne provode strukturu dokumenta. S MongoDB-om može se ugraditi povezane podatke u jednu strukturu ili dokument, a te su sheme općenito poznate kao de normalizirani modeli, kao što je prikazano na slici 6, ugniježđeni modeli podataka omogućuju aplikacijama pohranu srodnih podataka u isti zapis baze podataka.

Slika na kojoj se prikazuje tekst, Font, snimka zaslona, crta

Opis je automatski generiran

Slika 6 Ugniježđeni dokument

Dok u drugu ruku normalizirani model podataka, prikazan na slici 7, opisuje odnos pomoću referenci između dokumenata. Koristi se u slučajevima kada bi ugnježđivanje rezultiralo dupliciranjem podataka koje ne bi pružalo veće performanse prilikom pretraživanja baze.

Slika na kojoj se prikazuje tekst, Font, snimka zaslona, crta

Opis je automatski generiran

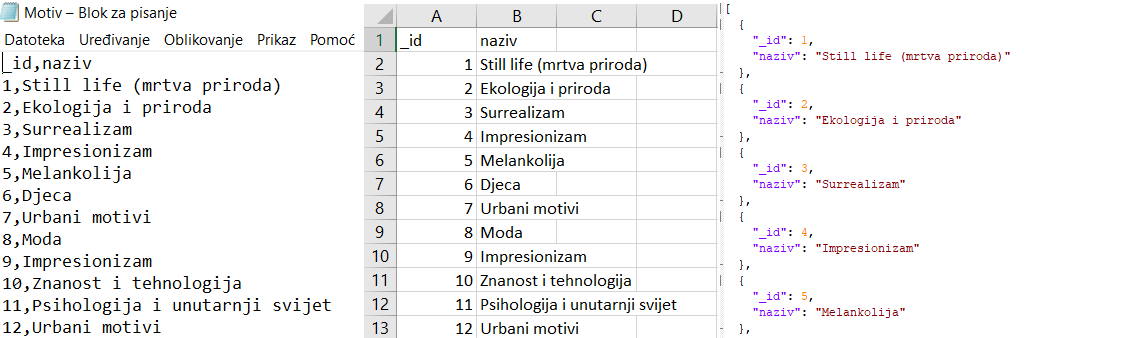
Slika 7 Normalizirani dokument

# Izrada modela baze podataka u MongoDB

Izrada baze podataka u MongoDB-u obavljena je po principu repliciranja baze podataka iz SQL-a tj. kreirani broj kolekcija ekvivalentan je broju tablica u SQL-u, a kao jedna od značajnijih promjena ističe primarni ključ koji je prilagođen MongoDB-u. U nastavku detaljnije je opisan postupak stvaranja baze podataka.

## Unos podataka u kolekcije

Obzirom da su podaci veći izgenerirani, za potrebe baze podataka u SQL-u, potrebno je prilagoditi podatke unosu u MongoDB bazu podataka. Na slici 5 prikazan je primjer generiranih podataka, stoga je uklonjen određeni dio zapisa u datotekama, kako bi se podaci mogli preoblikovati u JSON oblik prikladan za unos u MongoDB. Sljedeća slika prikazuje postupak preoblikovanja podataka u oblik prikladan za unos u MongoDB.



Slika 8 Oblikovanje podataka

Prema slici 6 uklonjeni su *Insert Into* **naziv\_tablice** ( **atributi** ) *Values* ( ); za sve unose u tablicu te su preostali samo podaci koji se unose u tablice, osim toga dodani su nazivi atributa u prvi red zapisa. Nakon toga txt datoteka prenamijenjena je u CSV datoteku, kako MongoDB sadrži mogućnost unosa podataka pomoću CSV datoteke testiran je i taj postupak unosa podataka. Međutim došlo je do pojave određenih problema prilikom unosa zadnjeg atributa u CSV datotekama, nakon toga kao rješenje nametnuo se postupak preoblikovanja CSV datoteke u JSON format. U tu svrhu korištena je internetska stranica *csvjson* ( <https://csvjson.com/> ) koja omogućava učitavanje CSV datoteke te ju preoblikuje u JSON oblik. Na slici 7 nalazi se sučelje internetske stranice csvjson te je prikazano pretvaranje CSV datoteke *Motiv* u JSON format.

Slika na kojoj se prikazuje tekst, snimka zaslona

Opis je automatski generiran

Slika 9 Internetska stranica csvjson

Prenamijenjeni podaci uneseni su u MongoDB bazu podataka, a sljedeća tablica prikazuje vrijeme potrebno da se ti podaci pohrane u bazu podataka.

|  |  |
| --- | --- |
| NAZIV | VRIJEME |
| POŠTANSKI BROJ | **[00:00:11]** |
| UMJETNIK | **[00:00:15]** |
| UMJETNINA | **[00:00:12]** |
| IZLOŽBA | **[00:00:18]** |
| POSJETITELJ | **[00:00:14]** |
| STAVKA | **[00:00:26]** |
| TIP | **[00:00:01]** |
| UMJETNINA IZLOŽBA | **[00:00:12]** |
| MOTIV | **[00:00:02]** |
| UMJETNINA MOTIV | **[00:00:17]** |

Tablica 12 Vrijeme potrebno za unos podataka u MongoDB bazu podataka

Slika na kojoj se prikazuje tekst, Font, broj, snimka zaslona

Opis je automatski generiran

Slika 10 Prikaz kolekcija u MongoDB bazi podataka

# Usporedba brzine izvršavanja zahtjeva SQL i MongoDB baze podataka

Prva jasno uočljiva razlike između SQL i MongoDB baze podataka je vrijeme potrebno za unos podataka u tablicu kako je prikazano u tablici 10 za SQL bazu i 12 za MongoDB bazu, a jasniji prikaz o tome kolika je razlika u vremenu potrebnom za unos podataka prikazano je na sljedećem grafu.

Slika 11 Usporedba vremena potrebnom za unos podataka u baze

## Izvršavanje upita

U ovom poglavlju će se testirati neki osnovne i često korišteni upiti nad bazom podataka u napravljenim kolekcijama te će se mjeriti potrebno vrijeme da se isti izvrše. Isti upiti će se napraviti u relacijskoj i ne relacijskoj bazi podataka te će se na posljetku usporediti dobivene vrijednosti.

### Dohvaćanje svih zapisa

Naredbom *'Select \* from* ***Naziv\_Tablice***;' dohvaćamo sve podatke iz SQL baze koji su uneseni u navedenu tablicu, a dohvaćanje podataka provedeno je na svim tablicama u bazi. Vrijeme trajanja navedene operacije za sve tablice iznosi 4878 ms. Naredba istog tipa za MongoDB provodi se pomoći *'db.****Naziv\_Kolekcije****.find({ });*' a potrebno vrijeme za njeno izvršavanje iznosi 1101ms. Na slici 12 prikazan je postupak mjerenja vremena za tablicu tj. kolekciju PostanskiBroj u SQL-u i MongoDB-u.

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, zaslon

Opis je automatski generiran

Slika 12 Mjerenje vremena učitavanja podataka za PostanskiBroj

### Dodavanje zapisa u baze

Na sljedećoj slici, slika 13, prikazane su naredbe za unos novog zapisa u bazu. Kao primjer korištena je tablica tj. kolekcija Umjetnik a vrijeme potrebno za unos podatka u SQL iznosi 10 ms dok vrijeme potrebno za unos u MongoDB iznosi 6 ms. Dodavanje se u SQL-u obavlja pomoću ' *Insert Into* ***Naziv\_Tablice*** *Values();*', dok u MongoDB-u pomoću naredbe ' *db.****Naziv\_Kolekcije****.insertOne({})* '.

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, softver

Opis je automatski generiran

Slika 13 Unos podataka u tablicu/kolekciju Umjetnik

### Uklanjanje zapisa iz baze

Uklanjanje podatka iz baze kod SQL-a i MongoDB obavljeno je preko atributa id ,a na slici 14, prikazan je postupak uklanjanja Umjetnika iz baze. Vrijeme potrebno za izvršavanje navedenog upita u SQL-u iznosilo je 21 ms dok za MongoDB istoj naredbi bilo je potrebno 9 ms.

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, crta

Opis je automatski generiran

Slika 14 Uklanjanje podatka iz tablice/kolekcije Umjetnik

### Sortiranje podataka

U tablici / kolekciji Umjetnina sortirani su podaci prema cijeni silazno s tim da su uključene samo umjetnine napravljene nakon 1960-te godine. U SQL-u upit je izvržen pomoću naredbe ' *select \* from Umjetnina where godina>1960 order by cijena ASC* ', a potrebno vrijeme za izvršavanje upita iznosi 336 ms, dok za MongoDB bazu upit se izvršava pomoću naredbe ' *db.Umjetnina.find({godina : {$gt : 1960}}).sort({"cijena":1})* ', a potrebno vrijeme za izvršavanje iznosi 119 ms. Na sljedećoj slici, slika 15, prikazan je upit za sortiranje kao i iznos potrebnog vremena i broj podataka koji je vraćen na upit.

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, Font

Opis je automatski generiran

Slika 15 Sortiranje umjetnina po cijeni

Dio podataka koji su dobiveni nakon upita za sortiranje i primjenom ograničenja o godini nastanka umjetničkog djela prikazani su na slici 16.

Slika na kojoj se prikazuje tekst, snimka zaslona, izbornik, Font

Opis je automatski generiran

Slika 16 Podaci dobiveni primjenom upita za sortiranje

### Operacije agregacije

Kao zadatak zadano je bazama da ispišu minimalnu cijenu kupljenih umjetničkih dijela. Zanimljivo je uočiti da je SQL bazi podataka bilo potrebno manje vremena za dohvaćanje rezultata upita tj. 51 ms dok je za isti upit MongoDB-u bilo potrebno 324 ms .Sljedeća slika, slika 17, prikazuje upit koji je proslijeđen bazama te vrijeme potrebno za dobivanje rezultata.

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, Font

Opis je automatski generiran

Slika 17 Ispis imena, prezimena, grada i države za pojedinog umjetnika

Dok sljedeća slika, slika 18, prikazuje podatke koji su dobiveni kao rezultat upita prikazanih na prethodnoj slici.

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, crta

Opis je automatski generiran

Slika 18 Rezultati upita

### Ostala testiranja

Uspoređivanje će se praviti na temelju rezultata dobivenih korištenjem MongoDB baze podataka kao ne relacijske baze i MsSQL kao relacijske baze podataka. Na sljedećoj slici, slika 19, prikazani su upiti i vrijeme potrebno za izvršavanje upita kod MongoDB i SQL baze podataka.

Slika na kojoj se prikazuje tekst, snimka zaslona, broj, paralelno

Opis je automatski generiran

Slika 19 Popis upita i vremena potrebnih za njihovo izvršavanje

Slika 20 Grafički prikaz potrebnog vremena za osnovne upite

Slika 21 Grafički prikaz potrebnog vremena za sortiranje podataka

Slika 22Grafički prikaz potrebnog vremena za ostale operacije agregacije

Nakon izvedenog testiranja nad 800 052 podatka u svim tablicama tj. kolekcijama može se primijetiti da MongoDB ima prednost pri izvođenju osnovnih upita odnosno: učitavanje svih podataka, dodavanje zapisa, uklanjanje zapisa i sl. Prednost je značajna u pogledu učitavanja podataka gdje je MongoDB učitao sve podatke u pripadajuće kolekcije za 1101 ms dok je SQL-u trebalo 4878 ms, a grafički prikaz ostalih razlika prikazan je na slici 20. Što se tiče postupka sortiranja i tu je Mongo ima značajnu prednost. Od 10 upita vezanih za sortiranje u svih 10 je MongoDB bio brži u odnosu na SQL, grafički prikaz vremena potrebnog za sortiranje podataka prikazan je na slici 21. Na posljetku provedene su ostale operacije agregacije gdje je SQL pokazao značajno bolje rezultate, a provedeni upiti odnosili su se na određivanje minimuma, brojanje određenih tipova podataka, određivanje prosječne vrijednosti i sl. Grafički prikaz vremena potrebnog za određivanje vrijednosti ostalih operacija agregacije prikazan je na slici 22.

# Zaključak

U ovom radu napravljena je usporedba relacijske baze podataka SQL i ne relacijske baze podataka MongoDB. Provedeno je testiranje performansi jednostavnih i često korištenih upita i promatrano je vrijeme potrebno za njihovo izvršavanje. Može se zamijetiti da MongoDB i MSSQL nude različita rješenja te posjeduju različite prednosti i nedostatke pri rješavanju problema i zahtjeva korisnika. Uspoređene su dvije baze popunjene istim podacima u različitom obliku.

Koristeći MongoDB napravljena je dokumentno orijentirana baza nad kojoj su se provodili isti testovi kao u relacijskoj bazi konfiguriranoj pomoću MsSQL-a. Rad je pokazao da dokumentno orijentirana baza podataka ima prednosti pri izvođenju osnovnih upita tj. unosa, brisanja, ažuriranja i sl. te sortiranja podataka, dok relacijska baza podataka pokazala je prednost pri izvođenju operacija agregacije kao što su izračun prosječnih vrijednosti, brojanje određenih tipova podataka, grupiranje i sl.

Može se doći do zaključka da takozvane NoSQL baze podataka više pogoduju radu sa podacima bez strukture te omogućuju veću fleksibilnost. S druge strane relacijske baze podataka sa njihovim definiranom strukturom omogućuju višu razinu sigurnosti pri održavanju i zaštiti podataka.