

Usporedba relacijske i nerelacijske baze podataka

Hudinec, Lorena

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Polytechnic of Međimurje in Čakovec / Međimursko veleučilište u Čakovcu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:110:420287>

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

Download date / Datum preuzimanja: **2024-09-02**



Repository / Repozitorij:

[Polytechnic of Međimurje in Čakovec Repository - Polytechnic of Međimurje Undergraduate and Graduate Theses Repository](#)



MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

Lorena Hudinec

**USPOREDBA RELACIJSKE I NERELACIJSKE BAZE
PODATAKA**

ZAVRŠNI RAD

Čakovec, rujan 2023.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

Lorena Hudinec

**USPOREDBA RELACIJSKE I NERELACIJSKE BAZE
PODATAKA
COMPARISON OF RELATIONAL AND NON-
RELATIONAL DATABASES**

ZAVRŠNI RAD

Mentor:
Mr. sc. Željko Knok

Čakovec, rujan 2023.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
ODBOR ZA ZAVRŠNI RAD

Čakovec, 31. siječnja 2023.

država: **Republika Hrvatska**
Predmet: **Baze podataka II-izborni**

ZAVRŠNI ZADATAK br. 2022-RAČ-I-4

Pristupnik: **Lorena Hudinec (0313023157)**
Studij: **Izvanredni preddiplomski stručni studij Računarstvo**
Smjer: **Inženjerstvo računalnih sustava i mreža**

Zadatak: **Usporedba relacijske i nerelacijske baze podataka**

Opis zadatka:

Zadatak je izraditi bazu podataka na jednom istovijetnom primjeru u relacijskom modelu i ne relacijskom modelu baze podataka. Usporedbu izraditi od nivoa instalacije potrebnih alata do svih sastavnih dijelova same baze podataka. Za primjer koristiti bazu Mysql i MongoDB. Za potrebe ovog zadatka izraditi sučelje koje će poslužiti za paralelnu usporedbu navedenih baza podataka. Posebnu pozornost posvetiti mogućnostima primjene navedenih baza.

Alati: Mysql, MongoDB, potrebni alati za izradu sučelja po osobnom izboru

SAŽETAK

U današnje vrijeme, kada je tehnologija od iznimno značajne važnosti, bitno je iskoristiti njene prednosti. Odgojno-obrazovne ustanove većim su dijelom opremljene računalima za kvalitetno izvođenje nastave, praćenje rada i interaktivniji pristup. Dakle, vidljivo je da se ustanove digitaliziraju, pa tako i vrtići. Tehnologijom se pojednostavljuje komunikacija između roditelja i odgojitelja, pruža se bolji uvid u informacije, stvaraju se uspomene koje je bitno sačuvati. U takvom smislu, važno je spremiti informacije i podatke, a to nam omogućava baza podataka.

Ovaj rad sadrži usporedbu relacijske i nerelacijske baze podataka na istovjetnom primjeru podataka dječjeg vrtića. Prikazan je postupak izrade baze podataka, unos i izmjena te funkcionalnost baze podataka na web-sučelju. Za vizualni prikaz relacijske baze podataka korišten je ER model stvoren alatom MySQL Workbench.

Nerelacijska baza podataka izrađena je programom MongoDB, dok je vizualni prikaz napravljen Lucidchartom, web-aplikacijom za stvaranje dijagrama.

Web-sučelje izrađeno je pomoću otvorenog alata WordPress koji omogućava jednostavnu izradu web-stranice.

KLJUČNE RIJEČI: *relacijska baza podataka, nerelacijska baza podataka, MySQL Workbench, MongoDB Compass*

SADRŽAJ

1. Uvod	6
2. Alati i metode	7
2.1 MySQL Workbench instalacija	7
2.2 MongoDB Compass instalacija.....	9
2.3 XAMPP server instalacija	10
2.4 WordPress instalacija	11
3. Baza podataka.....	12
4. Relacijska baza podataka.....	13
5. Nerelacijska baza podataka	16
6. Usporedba relacijske i nerelacijske baze podataka.....	17
6.1 Izrada baze podataka.....	20
6.2 Izrada entiteta/kolekcija	21
6.3 Unos podataka.....	22
6.4 Brisanje podataka.....	23
6.5 Ažuriranje podataka	23
6.6 ER model/Dokumentni model	24
7. Aplikativni dio na istovjetnom primjeru	27
7.1 Ovlasti administratora.....	28
7.2 Ovlasti korisnika	29
8. Zaključak	30
Literatura.....	31
Prilozi.....	32
Popis tablica	33

1. UVOD

U današnjem digitalnom dobu, baze podataka igraju ključnu ulogu u pohrani, organizaciji i upravljanju podacima. S obzirom na raznolike zahtjeve i složenost informacijskih sustava, postoji mnogo različitih pristupa za pohranu podataka. Ove baze omogućuju učinkovito praćenje i upravljanje informacijama, što je od iznimne važnosti u različitim sektorima, uključujući i odgojno-obrazovne ustanove. Među najvažnijim odlukama koje organizacije moraju donijeti jest izbor između relacijskih i nerelacijskih baza podataka kao osnove za svoje informacijske sustave. Takva odluka ima duboke posljedice na performanse, skalabilnost i fleksibilnost sustava.

Relacijske baze podataka bile su temeljni standard za organizaciju i upravljanje strukturiranim podacima dugi niz godina, dok su se nerelacijske baze podataka pojavile kao odgovor na nove izazove koje su postavili moderni aplikacijski zahtjevi, uključujući visoku skalabilnost i brzu dostupnost podataka. Svaki od ovih modela ima svoje prednosti i nedostatke, a izbor između njih ovisi o specifičnim potrebama i zahtjevima aplikacije.

Cilj ovoga rada je analizirati i usporediti relacijske i nerelacijske baze podataka kako bi se pružio temeljni uvid u prednosti, mane i primjenu spomenutih modela. Osim usporedbe tehničkih aspekata, rad se također bavi izradom relacijske i nerelacijske baze podataka s odgovarajućim alatima te izradom web-sučelja za upravljanje podacima iz baze 'vrtic'.

2. ALATI I METODE

Za izradu rada korištena je dostupna literatura navedena u popisu literature te alati otvorenog koda. Relacijska baza podataka izrađena je MySQL Workbench grafičkim alatom koji omogućava modeliranje podataka. Inženjering prema naprijed i unazad (*eng. forward and reverse engineering*) pruža mogućnost ispisivanje SQL koda pomoću vizualnog modela i obratno. [1] Za izradu nerelacijske baze podataka korišten je program MongoDB Compass, dokument baza koja omogućava jednostavan rad s podacima. Osigurava skalabilnost, to jest mogućnost proširivanja baze sa što više podataka. [2]

Za provjeru sadržaja, prije objavljivanja na web poslužitelj, korišten je XAMPP, softverski server otvorenog koda koji omogućava testiranje MySQL, PHP, Pearl i Apache projekata jednostavnim pokretanjem lokalnog poslužitelja.

WordPress je jedan od najpopularnijih, besplatnih alata za stvaranje web-aplikacija. Sadržava predloške, dodatke (*eng. plugin*) i razne opcije za objavljivanje aplikacije na poslužitelja. Kod instalacije važno je da se alat nalazi u direktoriju XAMPP-a kako bi se vidio sadržaj prije objavljivanja. Potrebno je stvoriti bazu podataka gdje će se spremite tablice WordPressa. [3]

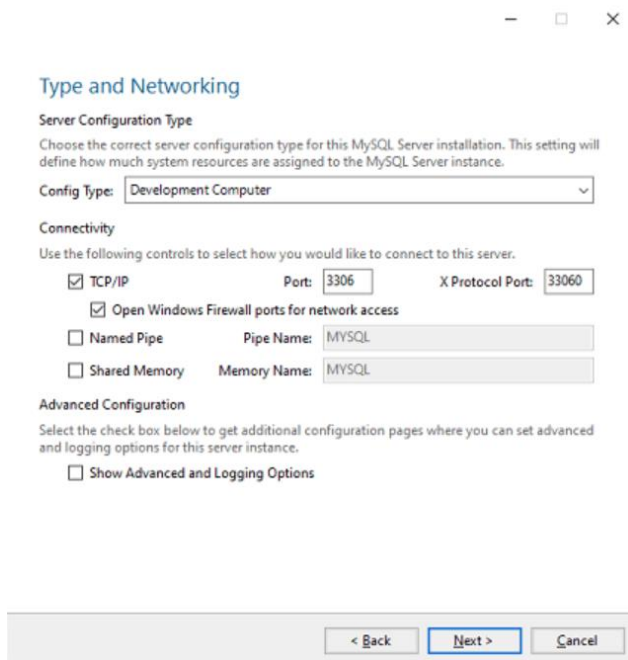
ER dijagram relacijske baze podataka izrađen je alatom MySQL Workbench. Unesen je SQL kod koji je, pomoću inženjeringa prema nazad, stvorio vizualni prikaz baze podataka.

Vizualni prikaz nerelacijske baze podataka izrađen je web-aplikacijom Lucidchart. Ručno su stvorene tablice u svrhu jednostavnijeg razumijevanja podataka u bazi.

2.1 MySQL Workbench instalacija

Za potrebe ovog rada instalirana je verzija 8.0.34. Sama instalacija je vrlo jednostavna. Na službenoj stranici nalaze se alati za preuzimanje gdje je potrebno odabrati MySQL Workbench. Potrebno je prilagoditi operativnom sistemu računala na koji se preuzima te se može započeti preuzimanje.

Nakon preuzimanja paketa, pojavljuje se sučelje za instalaciju u kojem se odabire mjesto instalacije te željene komponente.



Slika 1. Konfiguracija MySQL Workbench
IZVOR: autor

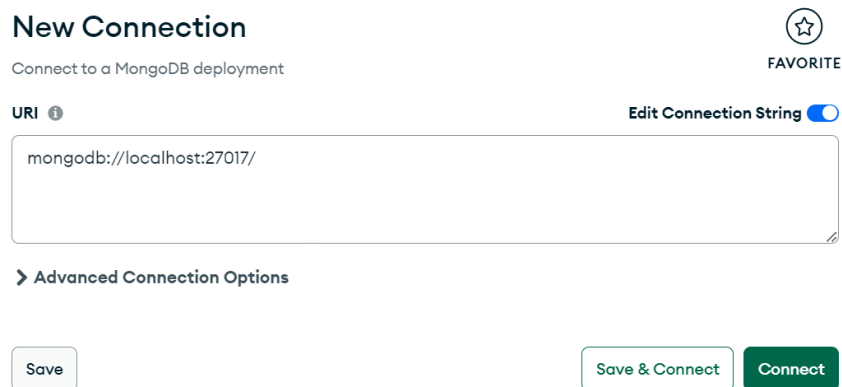
Prateći sve upute instalacije, otvara se sučelje sa stvorenom konekcijom. Nova konekcija dodaje se pritiskom na gumb plus ili odabirom *Database – New Connection*.



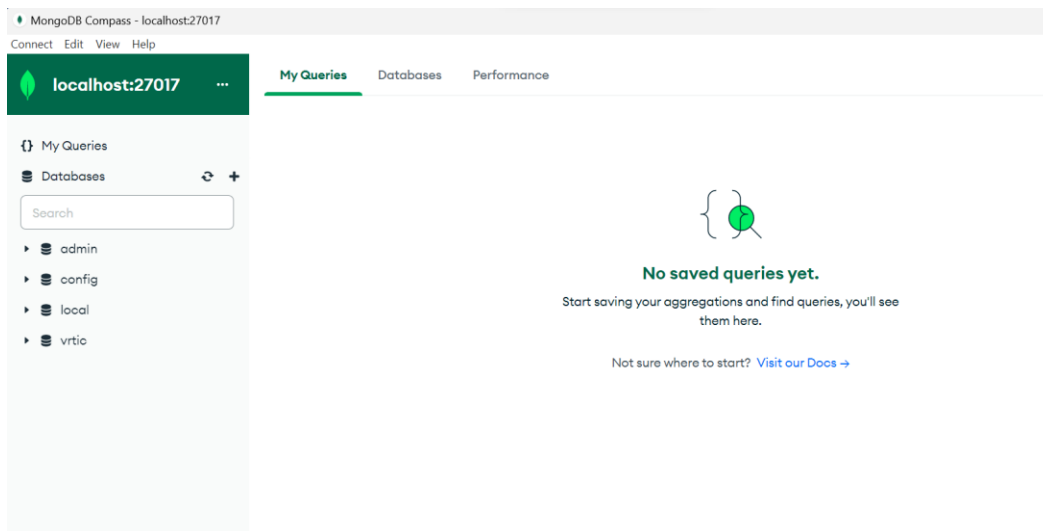
Slika 2. Sučelje MySQL Workbench
IZVOR: autor

2.2 MongoDB Compass instalacija

Na službenoj stranici odabire se opcija preuzimanja programa koja se prilagođava računalu. Za potrebe ovog rada instalirana je verzija 1.39.4. Nakon instalacije otvara se grafičko sučelje koje traži konekciju. U ovom radu stvorena je konekcija putem lokalnog poslužitelja (*eng. local host*).



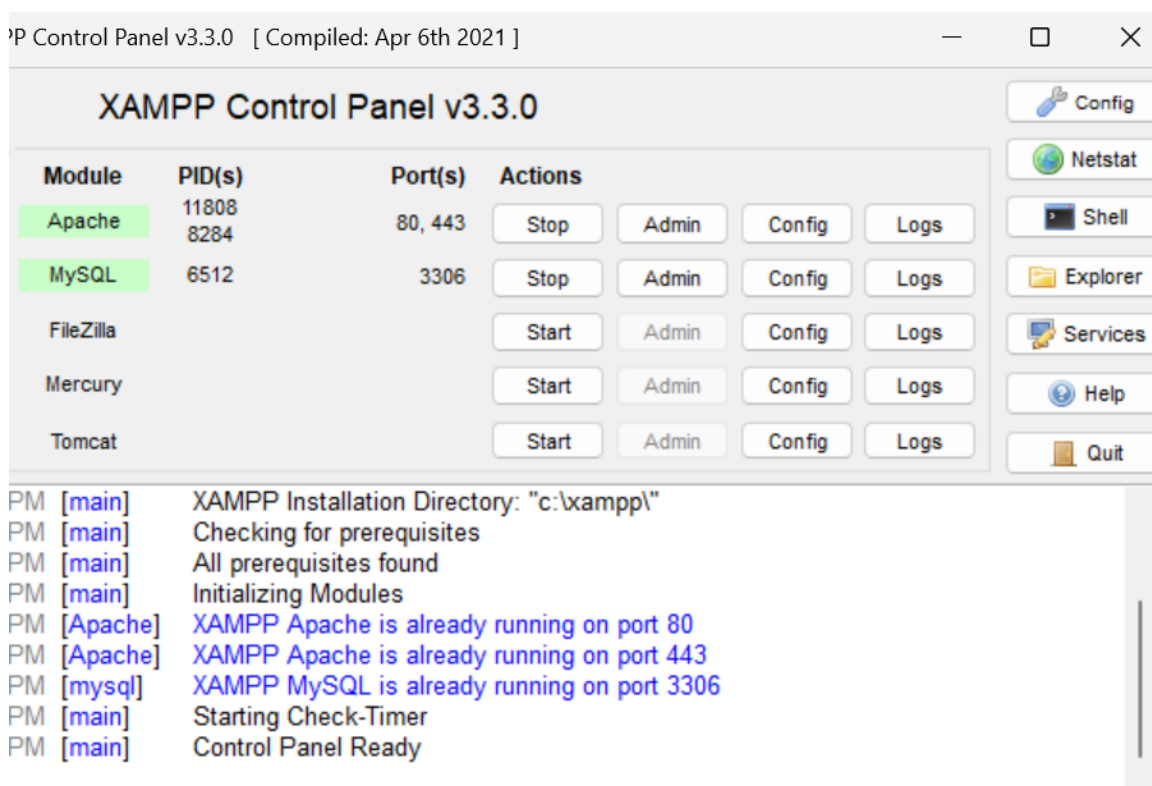
Slika 3. Stvaranje konekcije MongoDB Compass
IZVOR: autor



Slika 4. Sučelje MongoDB Compass
IZVOR: autor

2.3 XAMPP server instalacija

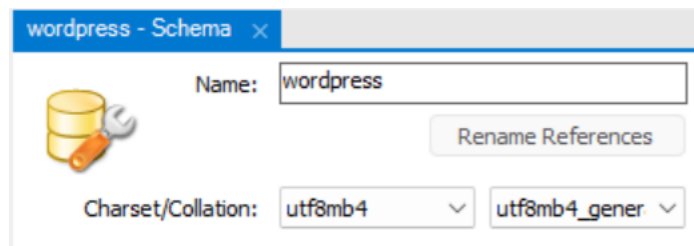
Za testiranje baze podataka i web-aplikacije važno je instalirati alat koji to omogućuje kako bi se mogao koristiti lokalni poslužitelj. U ovom radu instaliran je XAMPP server verzije 3.3.0. Preuzimanje se također vrši preko službene stranice. Nakon instalacije otvara se sučelje u kojem se, za potrebe ovog rada, pokreće *Apache* i *MySQL*. Apache server zaslužan je za pregled HTTP (eng. *The Hypertext Transfer Protocol*) web-stranica, a MySQL za pokretanje MySQL baze podataka na lokalnom poslužitelju.



Slika 5. Sučelje XAMPP-a
IZVOR: autor

2.4 WordPress instalacija

Kako bi se WordPress stranica testirala preko lokalnog poslužitelja, instalacija se ručno postavlja. Važno je prije instalacije pokrenuti XAMPP server te preuzetu komprimiranu datoteku WordPressa spremati u direktorij XAMPP-a i izdvojiti datoteku u podmapu *htdocs*. Nakon toga potrebno je u MySQL Workbenchu dodati novu shemu koja će spremati podatke WordPressa.



Slika 6. Shema za WordPress
IZVOR: autor

Otvora se lokalni poslužitelj za WordPress gdje se nastavlja instalacija. Potrebno je upisati podatke koji su stvoreni s bazom podataka kako bi se povezali. Nakon uspješno unesenih podataka, otvara se sučelje za WordPress spremno za rad.

Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	<input type="text" value="wordpress"/>	The name of the database you want to use with WordPress.
Username	<input type="text" value="root"/>	Your database username.
Password	<input type="password"/>	Your database password.
Database Host	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if <code>localhost</code> doesn't work.
Table Prefix	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.

Slika 7. Spajanje baze i WordPressa
IZVOR: autor

3. BAZA PODATAKA

„Baza podataka je u osnovi skup podataka organiziran u skladu s određenim kriterijama.“ [4]

Skup programa, koji omogućava manipulaciju podataka u bazi, naziva se Elektronički sustav za upravljanje bazama podataka (*eng. Electronic Database Management System – DBMS*), a on sadržava:

- funkcije za definiranje baze podataka (*eng. Data Definition*) – one se ostvaruju upotrebom standardnog jezika za rad s bazom podataka
- funkcije za manipulaciju podacima u bazi podataka (*eng. Data Manipulation Language*) – dodavanje naredbi za manipuliranje podataka
- upravljačke funkcije – funkcije sigurnosti baze podataka/zaštita od neovlaštenog korištenja, funkcije očuvanja integriteta u bazi podataka (*eng. backup*), funkcije statističkog praćenja rada baze podataka. [5]

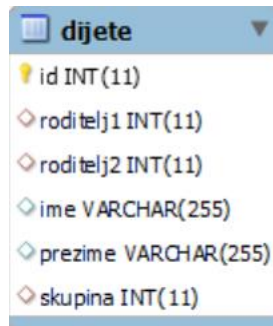
Postoje razni modeli DBMS-a, neki od čestih su:

1. Hijerarhijski model – podaci su organizirani prema odnosima nadređeni – podređeni
2. Mrežni model – podaci su organizirani na usmjerenim grafovima
3. Relacijski model – podaci su organizirani u tablice koje sadrže stupce i retke
4. Dokument model – podaci su organizirani u jednom dokumentu

4. RELACIJSKA BAZA PODATAKA

Relacijske baze podataka predstavljaju ključni element u modernom svijetu informacijske tehnologije. One omogućuju organizaciju, pohranu, upravljanje i pristup podacima na učinkovit i strukturiran način. Sustav za upravljanje relacijskim bazama podataka (*eng. Relation Database Management System – RDBMS*) je alat otvorenog koda koji pohranjuje podatke u relacijama, to jest tablicama. Za pristup bazi podataka koristi se SQL (*eng. Structured Query Language*).

Nazivi relacija, odnosno tablica, su entiteti. Oni su važni za jednostavniju organizaciju baze podataka. Mogu biti realan objekt, apstraktni sadržaj, događaj ili odnos. [6] U bazi podataka 'vrtić' entiteti su: dijete, roditelj, odgojitelj, skupina, korisnik, admin.



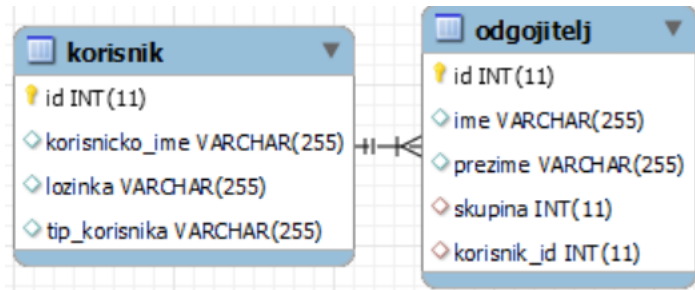
Slika 8. Entitet s atributima

IZVOR: autor

U tablici, kao stupci, nalaze se atributi koji opisuju entitete, a oni mogu biti ključni i neključni atributi. Ključni atribut ili identifikator u entitetu 'dijete' je *id*, dok neključni atribut je *ime*, *prezime*. Također, atribut *id* je primarni ključ koji sadrži jedinstvenu vrijednost podataka u tablici te ne smije imati nultu vrijednost.

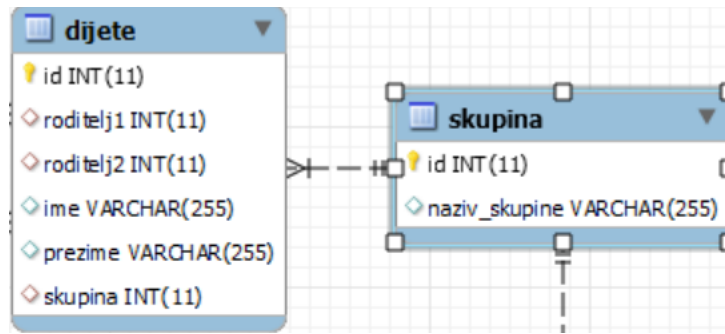
Veza između dva ili više entiteta omogućava njihovu međusobnu komunikaciju. Neke od najčešćih veza su:

1. Jedan prema jedan 1:1
2. Jedan prema mnogo 1:N
3. Mnogo prema mnogo M:N



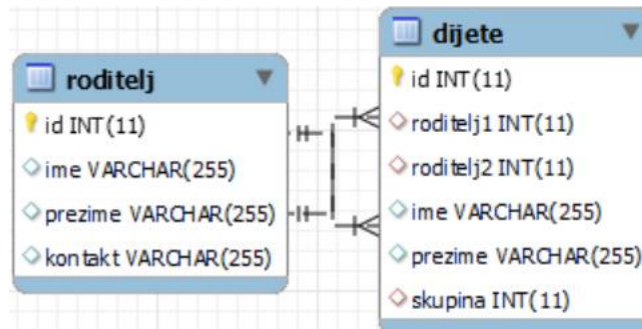
Slika 9. Veza 1:1
IZVOR: autor

Vezu 1:1 opisuju entiteti korisnik i odgojitelj, jednom odgojitelju se dodjeljuje korisničko ime kojim on postaje jedan korisnik.



Slika 10. Veza 1:N
IZVOR: autor

Entiteti dijete i skupina označavaju vezu 1:N, u jednoj skupini nalazi se više djece.



Slika 11. Veza M:N
IZVOR: autor

Posljednja veza u ovome radu je M:N gdje više roditelja može imati više djece.

Relacijske baze podataka imaju široku primjenu u različitim sektorima, uključujući poslovne aplikacije, e-trgovinu, zdravstvo i obrazovanje. U poslovnom okruženju, koriste se za praćenje inventara, narudžbi, financijskih izvještaja i klijentskih podataka. E-trgovine ih koriste za upravljanje proizvodima, narudžbama i kupcima. U zdravstvu se koriste za pohranu pacijentovih zapisa i medicinskog inventara dok ih odgojno-obrazovne institucije primjenjuju za praćenje podataka o djeci, roditeljima, odgojno-obrazovnim zaposlenicima i slično.

Prednosti relacijskih baza podataka su brojne. Struktura i organizacija podataka omogućuju jednostavan pristup i upravljanje. Integritet podataka održava se putem mehanizama za kontrolu valjanosti, a sposobnost povezivanja podataka iz različitih tablica olakšava analizu i donošenje odluka. Također, relacijske baze podataka nude mehanizme za kontrolu pristupa podacima i zaštitu osjetljivih informacija. Njihova struktura, organizacija i sposobnost povezivanja podataka čine ih nezamjenjivima za upravljanje informacijama.

5. NERELACIJSKA BAZA PODATAKA

Nerelacijske baze podataka, također poznate kao NoSQL (*Not Only SQL*) baze podataka, predstavljaju paradigmu pohrane i upravljanja podacima koja se temelji na principima koji se razlikuju od klasičnih relacijskih baza podataka. Ključna karakteristika ovih baza je njihova sposobnost da rukuju s različitim vrstama podataka, od tekstualnih do grafičkih te omogućavaju horizontalno skaliranje kako bi se nosili s rastućim volumenima podataka. [7]

Jedna od najvažnijih prednosti nerelacijskih baza podataka je njihova skalabilnost. Tradicionalne relacijske baze podataka često se suočavaju s problemima skaliranja jer zahtijevaju strogo definiranu shemu i strukturu podataka. Nasuprot tome, nerelacijske baze podataka omogućavaju dodavanje novih čvorova u sustav kako bi se povećala njihova sposobnost za obradu podataka. Ova karakteristika čini ih idealnim rješenjem za aplikacije koje zahtijevaju brzu i enigmatičku ekspanziju, poput web-stranica s visokom posjećenošću ili mobilnih aplikacija s milijunima korisnika.

Još jedna ključna prednost nerelacijskih baza podataka je njihova sposobnost za rukovanje nestrukturiranim podacima. Dok su relacijske baze podataka optimalne za tablične podatke s čvrstim shemama, nerelacijske baze podataka mogu se nositi s podacima poput JSON-a, XML-a i drugih nestrukturiranih formata. Ovo je posebno korisno u kontekstu obrade podataka s društvenih mreža gdje je struktura podataka često fluidna i promjenjiva.

Različite vrste nerelacijskih baza podataka pružaju različite pristupe za pohranu i upravljanje podacima. Dok grafičke baze podataka, optimizirane za pohranu i analizu grafova, omogućavaju brzo pretraživanje i analizu povezanih podataka, dokumentne baze podataka fokusiraju se na pohranu i upravljanje dokumentima, kao što su JSON ili XML datoteke. Važno je naglasiti da se odabir određene nerelacijske baze podataka često temelji na specifičnim zahtjevima aplikacije.

Unatoč svim prednostima, nerelacijske baze podataka nisu bez nedostataka. Jedan od glavnih izazova je složenost upita i nedostatak standardizacije u jezicima upita, što može otežati razvoj aplikacija i analizu podataka. Također, promjena iz relacijskog modela na nerelacijski zahtijeva promjenu razmišljanja i pristupa projektiranju baze podataka.

6. USPOREDBA RELACIJSKE I NERELACIJSKE BAZE PODATAKA

Relacijske i nerelacijske baze podataka imaju svoje nedostatke i prednosti stoga je vrlo važno prije odabira znati o kakvoj obradi podataka se radi. Ključne razlike mogu se definirati kroz nekoliko stavki:

1. Model podataka – ukoliko su podaci definirani i imaju jasno naznačenu strukturu, relacijska baza podataka je izvrsna za obradu tih podataka. S druge strane, nerelacijska baza podataka koristi različite modele koji omogućavaju veću fleksibilnost obrade.
2. Fleksibilnost sheme – relacijska baza podataka zahtijeva unaprijed čvrsto definiranu shemu gdje svaki podatak mora odgovarati prema njoj, dok nerelacijska baza podataka koristi shemu za čitanje (*eng. schema-on-read*) koja omogućava dodavanje, mijenjanje atributa i polja bez promjene sheme.
3. Tipovi podataka – za strukturirane podatke s definiranim vezama između entiteta najbolje se pokazuju relacijske baze podataka, a za nestrukturirane, polustrukturirane ili s promjenjivim podacima bolje odgovara nerelacijska baza podataka.
4. Transakcije – relacijska baza podataka pruža podršku za ACID transakcije (*eng. Atomicity, Consistency, Isolation, Durability*) odnosno, osigurava konzistentnost i pouzdanost dok se nerelacijska baza podataka fokusira na skalabilnost i brzinu što može dovesti do manje konzistentnosti.
5. Upiti i performanse – za kompleksne upite i analitičke funkcionalnosti koristi se SQL, a za brze upite NoSQL.
6. Primjena – relacijska baza podataka često se koristi u tradicionalnim aplikacijama gdje su podaci strukturirani, dok se nerelacijska baza podataka češće koristi za analizu velikih podataka, mobilnih aplikacija, društvenih mreža i slično. [8] [9]

Relacijska baza podataka, sa svojom čvrstom i strukturiranom shemom, posebno se ističe u situacijama gdje je potrebna dosljednost i preciznost podataka. To ju čini idealnim izborom za financijske sustave, zdravstvene zapise, online trgovine i sve aplikacije gdje je važno osigurati da se podaci ne izgube i da se održi njihova konzistentnost. Korištenje ACID transakcija omogućava aplikacijama da ostanu pouzdane čak i u slučaju neočekivanih događaja poput prekida mreže.

S druge strane, nerelacijska baza podataka pruža fleksibilnost koja je često potrebna u modernim aplikacijama. Ova vrsta baza podataka omogućuje brzu iteraciju i prilagodbu na promjene u zahtjevima za podacima, što je često slučaj u aplikacijama koje se brzo razvijaju. Također, nerelacijske baze su idealne za rješavanje problema povezanih s velikim količinama nestrukturiranih podataka, poput informacija s društvenih mreža.

Važno je napomenuti da se u stvarnom svijetu često koristi kombinacija oba tipa baza podataka kako bi se postigli najbolji rezultati. Ova hibridna strategija omogućava iskorištavanje prednosti oba pristupa u različitim dijelovima aplikacije. Naprimjer, podaci koji zahtijevaju strogu konzistenciju mogu se pohraniti u relacijsku bazu, dok se nestrukturirani podaci mogu čuvati u nerelacijskim bazama. Izbor između relacijske i nerelacijske baze podataka ovisi o zahtjevima projekata, prirodi podataka, ali i budućim potrebama za promjenama.

Zaključno, u nastavku se nalazi tablični prikaz usporedbe relacijske i nerelacijske baze podataka.

ASPEKT	SQL baza podataka	NoSQL baza podataka
Tipovi podataka	Struktura definirana pomoću sheme s definiranim tipovima podataka za svaku tablicu.	Fleksibilni, nema stroge sheme. Može pohranjivati nestrukturirane i polustrukturirane podatke.
Shema	Stroga, unaprijed definirana s fiksnim stupcima i tipovima podataka.	Dinamička shema. Promjene u strukturi podataka su fleksibilne.
Skalabilnost	Vertikalna skalabilnost (nadogradnja resursa)	Horizontalna skalabilnost (dodavanje čvorova)
Konzistencija	Stroga, podaci će uvijek biti dosljedni, ali to može dovesti do veće latencije.	Eventualna, podaci će se uskladiti, ali ne nužno odmah.
Transakcije	Podrška za transakcije s ACID svojstvima	Nema podrške za složene transakcije u svim NoSQL sustavima.
Pristup podacima	Jednostavno, zbog strukturirane sheme	Složenije, zbog nestrukturirane sheme
Prilagodljivost	Bolje se prilagođava složenim transakcijskim sustavima i strogim shemama.	Bolje se prilagođava potrebama za brzim skaliranjem i velikim volumenima podataka.
Upitni jezik	SQL je standardni jezik za upite i manipulaciju podacima.	Koristi se poseban jezik ili upitne metode specifične za svaki sustav.
Primjena	Često se koristi za aplikacije gdje je brzina važna (društvene mreže)	Često se koristi za aplikacije gdje su transakcije i dosljednost ključni (financijske aplikacije)

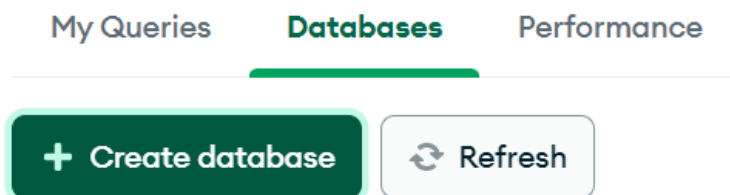
Tablica 1. Usporedba SQL i NoSQL baze podataka

6.1 Izrada baze podataka

U ovom radu relacijska baza podataka je izrađena naredbom

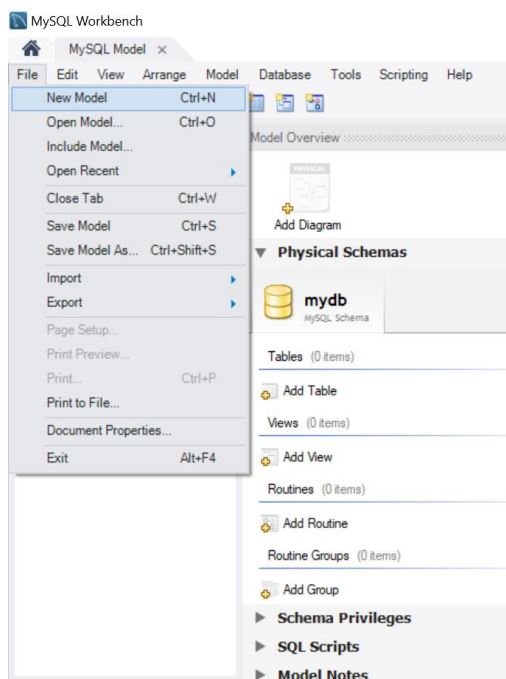
```
CREATE DATABASE 'vrtic';
```

dok je nerelacijska baza podataka napravljena preko sučelja.



Slika 12. Izrada NoSQL baze podataka
IZVOR: autor

Također, baza podataka u MySQL Workbench alatu može se izraditi odabirom *File – New Model* gdje se otvaraju opcije dodavanja tablice, dijagrama i slično.



Slika 13. Izrada SQL baze podataka
IZVOR: autor

6.2 Izrada entiteta/kolekcija

Baza podataka nosi naziv 'vrtic' te ima iste entitete u relacijskoj bazi podataka odnosno kolekcije u nerelacijskoj bazi podataka.

```
CREATE TABLE Dijete (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    roditelj1 INT,  
    roditelj2 INT,  
    ime VARCHAR(255),  
    prezime VARCHAR(255),  
    skupina INT,  
    FOREIGN KEY (roditelj1) REFERENCES Roditelj(id),  
    FOREIGN KEY (roditelj2) REFERENCES Roditelj(id),  
    FOREIGN KEY (skupina) REFERENCES Skupina(id)  
);
```

Slika 14. Izrada tablice 'Dijete'
IZVOR: autor

U MongoDB Compass odabirom na *Create Collection* stvara se prazna kolekcija.

Create Collection

Collection Name

Dijete

Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

› Additional preferences (e.g. Custom collation, Capped, Clustered collections)

Cancel

Create Collection

Slika 15. Izrada kolekcije 'Dijete'
IZVOR: autor

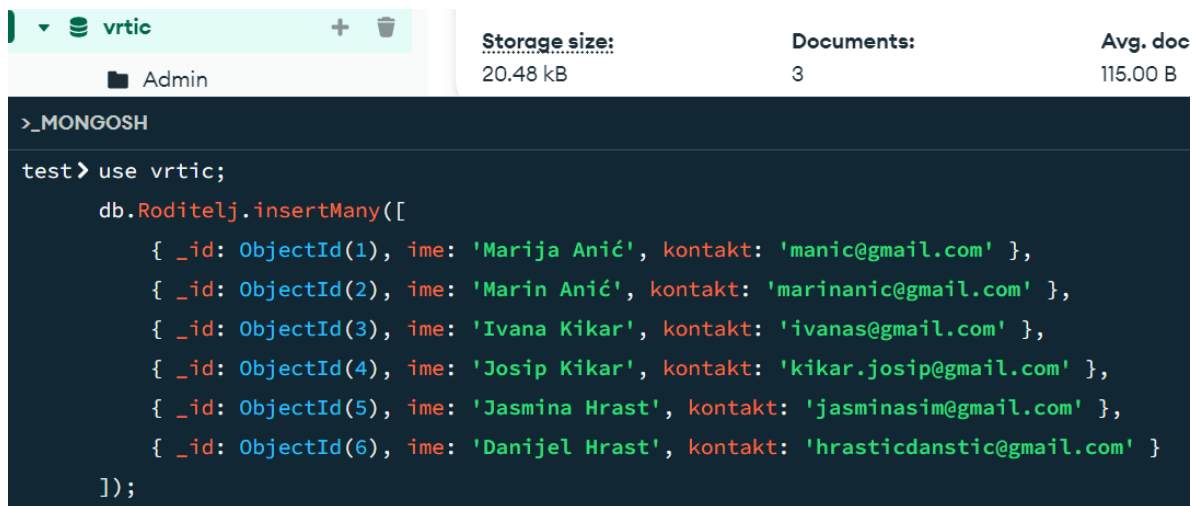
6.3 Unos podataka

Unos podataka u relacijsku bazu podataka izvršen je pomoću SQL naredbe INSERT INTO.

```
INSERT INTO Roditelj (ime, kontakt) VALUES
('Marija Anić', 'manic@gmail.com'),
('Marin Anić', 'marinanic@gmail.com'),
('Ivana Kikar', 'ivanas@gmail.com'),
('Josip Kikar', 'kikar.josip@gmail.com'),
('Jasmina Hrast', 'jasminasim@gmail.com'),
('Danijel Hrast', 'hrasticdanstic@gmail.com');
```

Slika 16. Unos podataka u relacijsku bazu podataka
IZVOR: autor

Podaci u nerelacijsku bazu podataka uneseni su preko interaktivne ljuške (eng. *shell*) koja služi za izvršavanje MongoDB naredbi.



The screenshot shows the MongoDB Shell interface. At the top, there is a navigation bar with a folder icon and the name 'vrtic'. Below it, a table displays database statistics:

Storage size:	Documents:	Avg. doc
20.48 kB	3	115.00 B

Below the table, the shell prompt is shown as `>_MONGOSH`. The user has entered the following commands:

```
test> use vrtic;
db.Roditelj.insertMany([
  { _id: ObjectId(1), ime: 'Marija Anić', kontakt: 'manic@gmail.com' },
  { _id: ObjectId(2), ime: 'Marin Anić', kontakt: 'marinanic@gmail.com' },
  { _id: ObjectId(3), ime: 'Ivana Kikar', kontakt: 'ivanas@gmail.com' },
  { _id: ObjectId(4), ime: 'Josip Kikar', kontakt: 'kikar.josip@gmail.com' },
  { _id: ObjectId(5), ime: 'Jasmina Hrast', kontakt: 'jasminasim@gmail.com' },
  { _id: ObjectId(6), ime: 'Danijel Hrast', kontakt: 'hrasticdanstic@gmail.com' }
]);
```

Slika 17. Unos podataka u nerelacijsku bazu podataka
IZVOR: autor

Za manipulaciju podataka korištena je naredba `findOne()` koja traži prvi dokument koji zadovoljava uvjet te dodjeljuje `_id` roditelja u varijablu `roditeljObjectId`.

```
const roditelj1ObjectId = db.Roditelj.findOne({ ime: 'Marija Anić' })._id;
const roditelj2ObjectId = db.Roditelj.findOne({ ime: 'Marin Anić' })._id;
const roditelj3ObjectId = db.Roditelj.findOne({ ime: 'Ivana Kikar' })._id;
const roditelj4ObjectId = db.Roditelj.findOne({ ime: 'Josip Kikar' })._id;
const roditelj5ObjectId = db.Roditelj.findOne({ ime: 'Jasmina Hrast' })._id;
const roditelj6ObjectId = db.Roditelj.findOne({ ime: 'Danijel Hrast' })._id;
```

Slika 18. NoSQL naredba `findOne()`
IZVOR: autor

6.4 Brisanje podataka

U relacijskoj bazi podataka SQL upit izbrisat će redak koji odgovara osobi s imenom "Nikolina" i prezimenom "Zmajic" iz tablice "Odgojitelj". Nakon izvođenja ovog upita, taj redak više neće postojati u tablici.

```
DELETE FROM Odgojitelj WHERE ime = 'Nikolina' AND prezime = 'Zmajic';
```

Slika 19. Brisanje podataka SQL
IZVOR: autor

U nerelacijskoj bazi podataka potrebno je definirati brisanje dokumenta. U ovom slučaju briše se prvi podatak koji odgovara uvjetu. `DeleteMany` briše sve dokumente koji odgovaraju uvjetu.

```
db.Odgojitelj.deleteOne({ ime: 'Nikolina', prezime: 'Zmajic' });
```

Slika 20. Brisanje podataka NoSQL
IZVOR: autor

6.5 Ažuriranje podataka

Za sigurnije ažuriranje podataka postavljen je uvjet `id=1`, kako ne bi došlo do nepoželjne promjene u slučaju istog imena i prezimena.


```
UPDATE Dijete
SET ime = 'Maja'
WHERE ime = 'Ana' AND prezime = 'Anić' AND id =1;
```

Slika 21. Ažuriranje podataka SQL
IZVOR: autor

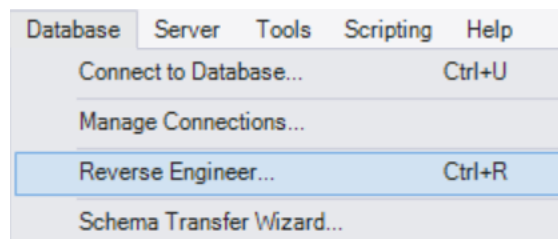
Za razliku od SQL upita, NoSQL prvo pretražuje dokument s uvjetima, u ovom slučaju prvi koji zadovoljava kriterije, zatim postavlja novo ime pomoću operatora *\$set*.

```
db.Dijete.updateOne(
  { ime: 'Ana', prezime: 'Anić' },
  { $set: { ime: 'Maja' } }
);
```

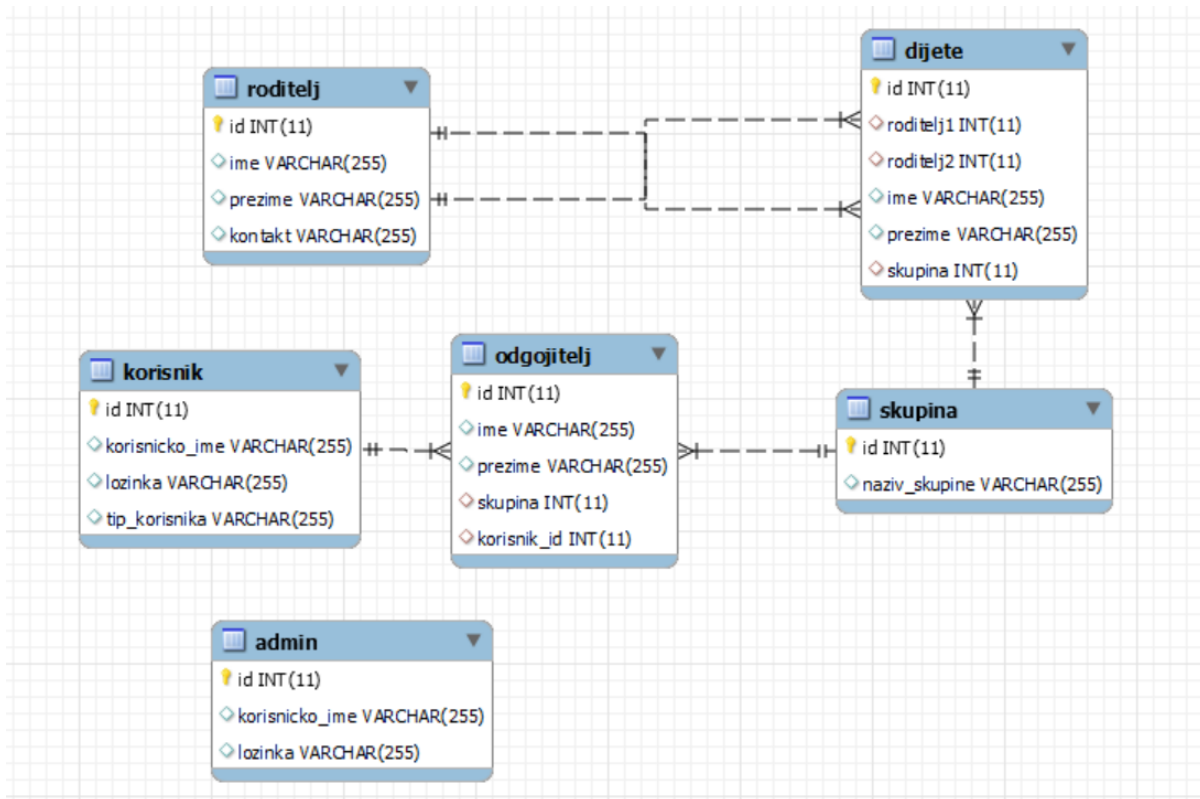
Slika 22. Ažuriranje podataka NoSQL
IZVOR: autor

6.6 ER model/Dokumentni model

ER model (*eng. Entity-Relationship*) je grafički model za prikazivanje strukture baze podataka. On pomaže u vizualizaciji entiteta, njihovih atributa, veze između entiteta i primarne ključeve. U ovom radu model je stvoren pomoću inženjeringa prema nazad.

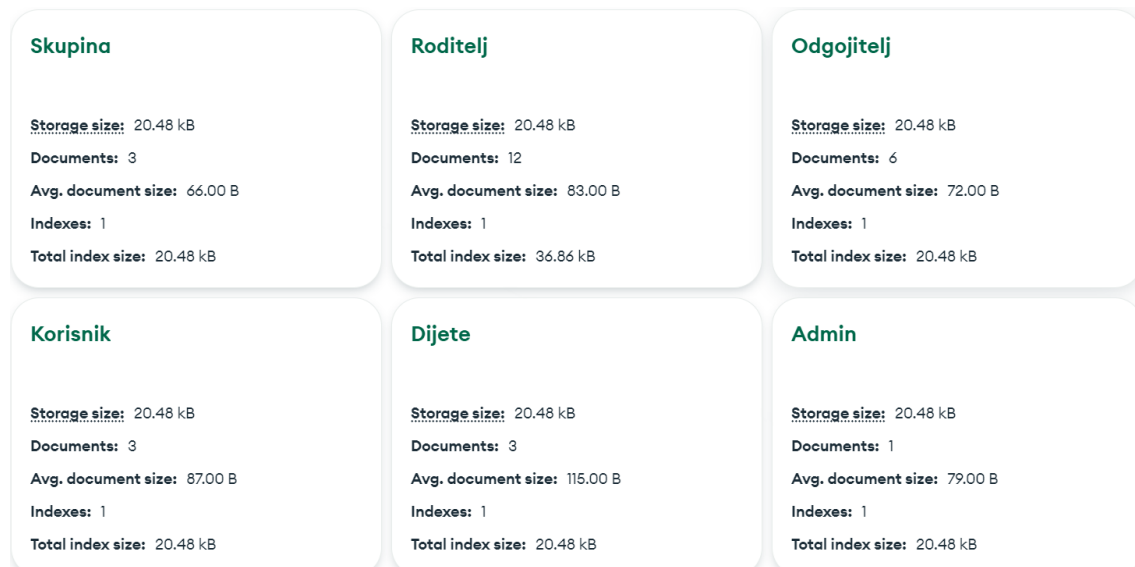


Slika 23. Inženjering prema nazad
IZVOR: autor



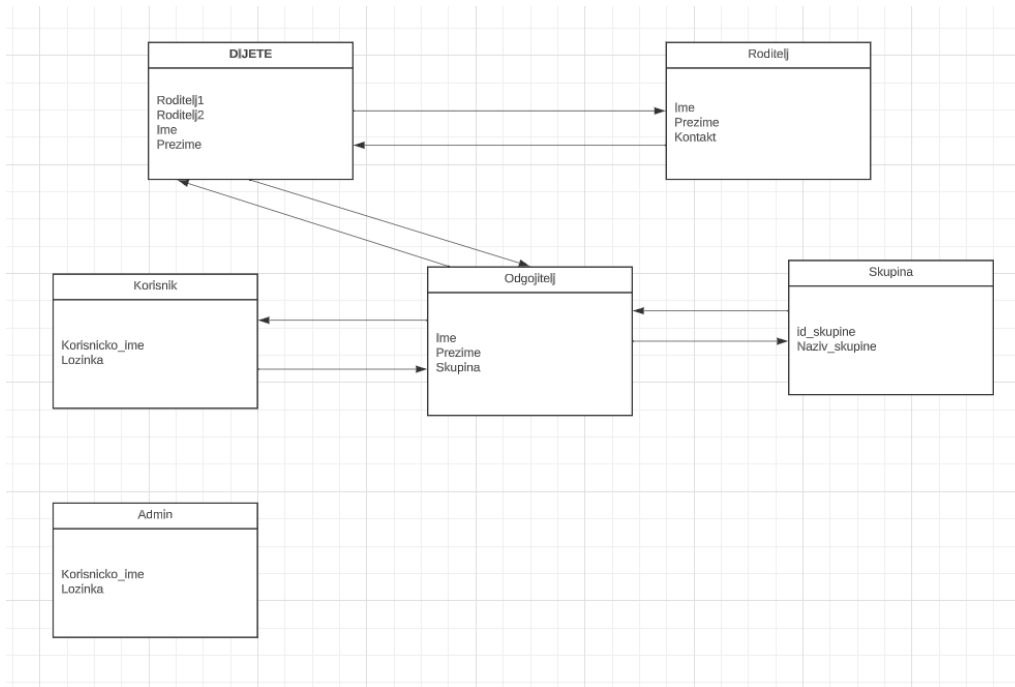
Slika 24. ER model
IZVOR: autor

U nerelacijskoj bazi podataka dokumentni model prikazuje dokumente, kolekcije i veze između njih. To može uključivati reference na druge dokumente.



Slika 25. Dokumentni model
IZVOR: autor

Za potrebe ovog rada, ručno je stvoren dijagram nerelacijske baze podataka. Korištena je web-aplikacija Lucidchart za bolju vizualizaciju dokumenata, kolekcija. Prema slici 25. može se primijetiti kako su sheme relacijske i nerelacijske baze podataka iste, bez obzira što nerelacijska ne sadrži entitete, atribute, veze ili primarne ključeve.

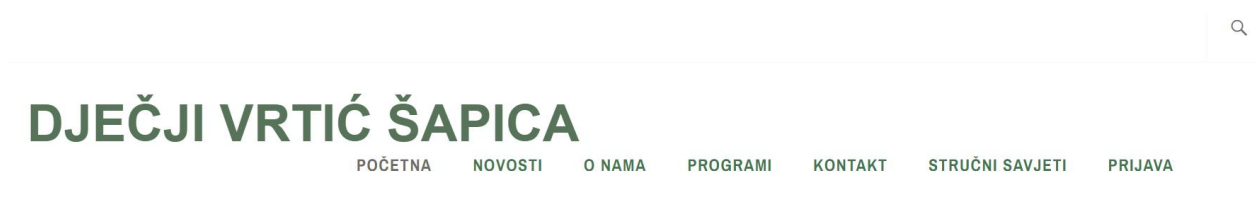


Slika 26. Dijagram NoSQL baze podataka

7. APLIKATIVNI DIO NA ISTOVJETNOM PRIMJERU

Za bolje razumijevanje izrađene baze podataka istovjetnog primjera, izrađeno je web-sučelje koje koristi bazu podataka 'vrtic'. Pomoću aplikacije moguće je izvršavati funkcije:

1. Pregled web-stranice
2. Prijava – admin, korisnik
3. Dodavanje i ažuriranje korisnika
4. Dodavanje i uređivanje objava



*Slika 27. Navigacijska traka stranice
IZVOR: autor*

Početna stranica je statična, odnosno samo administrator može mijenjati njen sadržaj. Objave se objavljuju na stranici Novosti, a njih mogu stvarati, uz administratora i korisnici, to jest odgojitelji. Prijava je omogućena samo za admina i korisnike preko WordPress prijave.

7.1 Ovlasti administratora

WordPress nudi mnoge mogućnosti administratoru za manipulaciju stranice. Prijavljuje se putem vlastitog korisničkog imena i lozinka te ima pristup u kompletno WordPress sučelje. Ono nudi sljedeće funkcije:

1. Dodavanje i uređivanje korisnika
2. Dodavanje i uređivanje strana
3. Dodavanje i uređivanje objava
4. Dodavanje i uređivanje zbirke medija

Korisnici [Dodaj novi](#)

Svih (9) | Administrator (1) | Urednik (1) | Autor (6) | Pretplatnik (1)

Grupne radnje [Primijeni](#) Promijeni ulogu na... [Promijeni](#)

<input type="checkbox"/>	Korisničko ime ↕	Ime	E-pošta ↕	Uloga
<input type="checkbox"/>	Hudi	Lorena Hudinec	lorena.hudinec@mev.hr	Pretplatnik
<input type="checkbox"/>	LorenaH	Lorena Hudinec	hudineclorena@07gmail.com	Administrator
<input type="checkbox"/>	MarinaLR	Marina Lauš Ruganec	marinalaus93@gmail.com	Urednik
<input type="checkbox"/>	odgojitelj_bubamare	Skupina Bubamare	odgojitelj_bubamare@gmail.com	Autor

Slika 28. Prikaz korisnika
IZVOR: autor

Stranice [Dodaj novu](#)

Svih (7) | Objavljenih (7)

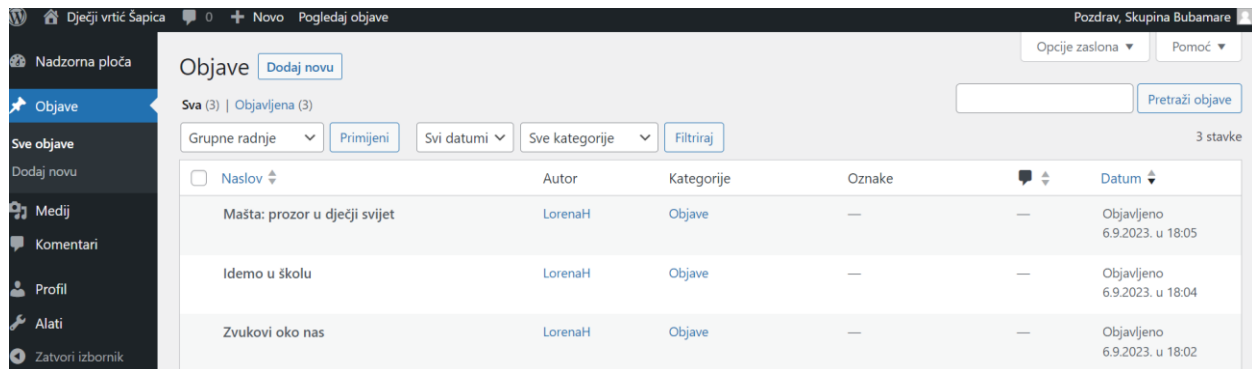
Grupne radnje [Primijeni](#) Svi datumi [Filtriraj](#) [Pretraži strani](#) 7 st

<input type="checkbox"/>	Naslov ↕	Autor	Datum ↕
<input type="checkbox"/>	Kontakt	LorenaH	Objavljeno 6.9.2023. u 17:18
<input type="checkbox"/>	Novosti — Stranica objava Uredi Brzo uredi Smeće Pregledaj	LorenaH	Objavljeno 6.9.2023. u 23:54
<input type="checkbox"/>	O nama	LorenaH	Objavljeno 6.9.2023. u 17:09
<input type="checkbox"/>	Početna — Početna stranica	LorenaH	Objavljeno 6.9.2023. u 17:09

Slika 29. Prikaz stranica

7.2 Ovlasti korisnika

Korisnik, odnosno odgojitelj, prijavljuje se pomoću korisničkog imena i lozinke koju mu je dodijelio administrator. Na stranici ima samo mogućnost dodavanja i uređivanja objava i medijskih zapisa.



Slika 30. Nadzorna ploča korisnika
IZVOR: autor

8. ZAKLJUČAK

Baze podataka su ključan alat za organizaciju i pristup podacima u današnjem digitalnom dobu. Razumijevanje njihove važnosti i primjene pomaže organizacijama i ustanovama u učinkovitijem upravljanju podacima i optimizaciji informacijskih sustava za različite potrebe.

Izbor između relacijskih i nerelacijskih baza podataka ovisi o specifičnim potrebama i zahtjevima aplikacije. Ako se naglašava konzistentnost i integritet podataka, relacijske baze često su bolji izbor. S druge strane, ako su brza dostupnost i skalabilnost prioriteti, nerelacijske baze mogu pružiti bolje rezultate.

Vrtići moraju pratiti različite informacije o djeci, roditeljima, odgojiteljima, skupinama, a relacijske baze podataka omogućuju strukturiranje tih podataka u jasno definirane tablice s atributima, što olakšava organizaciju i održavanje podataka. Također, često se takvi podaci moraju pretraživati, a relacijska baza podataka omogućava složene SQL upite koji olakšavaju dohvaćanje i analizu podataka.

S druge strane, dječji vrtići postaju sve veći, a s time dolazi veći broj djece, odgojitelja, skupina, tj. više podataka za unos u bazu podataka. Nerelacijska baza podataka omogućuje znatno veću fleksibilnost u oblikovanju i promjeni sheme podataka. Pohranjivanje nestrukturiranih podataka, poput slika i videa, je jednostavnija i kvalitetnija u odnosu na relacijsku bazu podataka. Također, brže se pristupa podacima, ali i jednostavnije.

Zaključujemo da nema univerzalnog rješenja koje bi odgovaralo svim situacijama te da je važno pažljivo razmotriti specifične zahtjeve aplikacije pri donošenju odluke između relacijskih i nerelacijskih baza podataka. S obzirom na dinamičnost digitalnog okruženja, sposobnost prilagodbe i razumijevanja prednosti, oba pristupa postaju ključni faktori za uspješno upravljanje podacima i informacijskim sustavima u suvremenom svijetu.

LITERATURA

1. MySQL Workbench <https://www.mysql.com/products/workbench/design/> 8.9.2023.
2. MongoDB <https://www.mongodb.com/what-is-mongodb> 8.9.2023.
3. Williams B., Damstra D., Stern H. Professional WordPress: Design and Development, 2015
https://www.google.hr/books/edition/Professional_WordPress/YhDTBQAAQBAJ?hl=hr&gbpv=0 8.9.2023.
4. Vikram V. PHP i MySQL – Zagreb, Mikro knjiga; 2005
5. Varga M. Upravljanje podacima, 2021
https://www.google.hr/books/edition/Upravljanje_podacima/3HcpEAAAQBAJ?hl=hr&gbpv=0 9.9.2023.
6. Entiteti i atributi <https://arhiva-2021.loomen.carnet.hr/mod/book/view.php?id=1207037&chapterid=172772> 9.9.2023.
7. MongoDB NoSQL <https://www.mongodb.com/nosql-explained> 9.9.2023.
8. Meier A., Kaufmann M. SQL & NoSQL Databases - Models, Languages, Consistency Options and Architectures for Big Data Management, 2019
https://www.google.hr/books/edition/SQL_NoSQL_Databases/XOCgDwAAQBAJ?hl=hr&gbpv=0 12.9.2023.
9. NoSQL <https://www.mongodb.com/nosql-explained> 9.9.2023.
10. NoSQL vs SQL <https://www.mongodb.com/nosql-explained/nosql-vs-sql> 9.9.2023.
11. SQL vs NoSQL <https://www.ibm.com/blog/sql-vs-nosql/> 9.9.2023.

PRILOZI

Slika 1. Konfiguracija MySQL Workbench IZVOR: autor	8
Slika 2. Sučelje MySQL Workbench IZVOR: autor	8
Slika 3. Stvaranje konekcije MongoDB Compass IZVOR: autor	9
Slika 4. Sučelje MongoDB Compass IZVOR: autor	9
Slika 5. Sučelje XAMPP-a IZVOR: autor	10
Slika 6. Shema za WordPress IZVOR: autor	11
Slika 7. Spajanje baze i WordPressa IZVOR: autor.....	11
Slika 8. Entitet s atributima.....	13
Slika 9. Veza 1:1 IZVOR: autor	14
Slika 10. Veza 1:N IZVOR: autor	14
Slika 11. Veza M:N IZVOR: autor.....	14
Slika 12. Izrada NoSQL baze podataka IZVOR: autor.....	20
Slika 13. Izrada SQL baze podataka IZVOR: autor.....	20
Slika 14. Izrada tablice 'Dijete' IZVOR: autor	21
Slika 15. Izrada kolekcije 'Dijete' IZVOR: autor	21
Slika 16. Unos podataka u relacijsku bazu podataka IZVOR: autor	22
Slika 17. Unos podataka u nerelacijsku bazu podataka IZVOR: autor.....	22
Slika 18. NoSQL naredba findOne() IZVOR: autor	23
Slika 19. Brisanje podataka SQL IZVOR: autor.....	23
Slika 20. Brisanje podataka NoSQL IZVOR: autor.....	23
Slika 21. Ažuriranje podataka SQL IZVOR: autor	24
Slika 22. Ažuriranje podataka NoSQL IZVOR: autor	24
Slika 23. Inženjering prema nazad IZVOR: autor	24
Slika 24. ER model IZVOR: autor.....	25
Slika 25. Dokumentni model IZVOR: autor.....	25
Slika 26. Dijagram NoSQL baze podataka	26
Slika 27. Navigacijska traka stranice IZVOR: autor	27
Slika 28. Prikaz korisnika IZVOR: autor.....	28
Slika 29. Prikaz stranica.....	28
Slika 30. Nadzorna ploča korisnika IZVOR: autor.....	29

POPIS TABLICA

Tablica 1. Usporedba SQL i NoSQL baze podataka..... 19