

Usporedba relacijskih i NoSQL baza podataka

Gašparović, Anna Marie

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Humanities and Social Sciences / Sveučilište Josipa Jurja Strossmayera u Osijeku, Filozofski fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:142:346418>

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

Download date / Datum preuzimanja: **2025-02-17**



FILOZOFSKI FAKULTET
SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

Repository / Repozitorij:

[FFOS-repository - Repository of the Faculty of Humanities and Social Sciences Osijek](#)



Sveučilište J.J. Strossmayera u Osijeku

Filozofski fakultet Osijek

Jednopredmetni sveučilišni preddiplomski studij informatologije

Anna-Marie Gašparović

Usporedba relacijskih i NoSQL baza podataka

Završni rad

Mentor: doc. dr. sc. Tomislav Jakopec

Osijek, 2023.

Sveučilište J.J. Strossmayera u Osijeku

Filozofski fakultet Osijek

Odsjek za informacijske znanosti

Jednopedmetni sveučilišni preddiplomski studij informatologije

Anna-Marie Gašparović

Usporedba relacijskih i NoSQL baza podataka

Završni rad

Društvene znanosti, Informacijske i komunikacijske znanosti, Informacijski sustavi
i informatologija

Mentor: doc. dr. sc. Tomislav Jakopec

Osijek, 2023.

Prilog: Izjava o akademskoj čestitosti i o suglasnosti za javno objavljivanje

Obveza je studenta da donju Izjavu vlastoručno potpiše i umetne kao treću stranicu završnoga, odnosno diplomskog rada.

IZJAVA

Izjavljujem s punom materijalnom i moralnom odgovornošću da sam ovaj rad samostalno napisao/napisala te da u njemu nema kopiranih ili prepisanih dijelova teksta tuđih radova, a da nisu označeni kao citati s navođenjem izvora odakle su preneseni.

Svojim vlastoručnim potpisom potvrđujem da sam suglasan/suglasna da Filozofski fakultet u Osijeku trajno pohrani i javno objavi ovaj moj rad u internetskoj bazi završnih i diplomskih radova knjižnice Filozofskog fakulteta u Osijeku, knjižnice Sveučilišta Josipa Jurja Strossmayera u Osijeku i Nacionalne i sveučilišne knjižnice u Zagrebu.

U Osijeku 12.09.2023.

Anna-Marie Gasparović, 0122238044

Ime i prezime studenta, JMBAG

Sažetak

Ovaj završni rad uspoređuje relacijske baze podataka (RDBMS) i NoSQL (Not Only SQL) baze podataka kako bi istaknuo njihove ključne karakteristike, prednosti i nedostatke. U završnom radu razjašnjen je pojam baze podataka, uz detaljna objašnjenja relacijskih i NoSQL baza podataka. Obje vrste imaju specifične prednosti i nedostatke, a korisniku je prepuštena odluka o izboru temeljem vlastitih potreba. Relacijski sustavi upravljanja bazama podataka (RDBMS) tradicionalno su dominirali pejzažem podataka, pružajući pouzdanost i konzistentnost putem strukturiranih tablica i SQL upita. Relacijske baze podataka koriste strukturu tablica s odnosima između njih, dok NoSQL baze podataka koriste fleksibilne i nestrukturirane modele podataka. NoSQL baze popunit će segment tržišta izazovima koje relacijske baze ne mogu riješiti. Kvalitetnije su za spremanje, obradu i pohranjivanje iznimno velikih količina podataka te samim tim brže su i učinkovitije. No, relacijske baze podataka bolje su po pitanju sigurnosti te imaju jasno definiranu shemu i upitni jezik koji vrijedi za sve relacijske baze, olakšavajući upotrebu. Usporedba performansi između ta dva tipa baza podataka pokazuje da NoSQL baze podataka imaju prednost kada je potrebno rukovati velikim volumenima podataka i postići visoku skalabilnost. S druge strane, relacijske baze podataka obično pružaju bolju podršku za transakcije, konzistentnost podataka i složenije upite. Kroz usporedbu ove dvije vrste baza podataka zaključuje se da nema univerzalnog "najboljeg" izbora. Analizom prednosti i nedostataka istih, ovaj rad pomaže donijeti informirane odluke pri odabiru baze podataka za svoje projekte. Kroz dublju analizu arhitekture, modela konzistentnosti, primjera upita te stvarnih slučajeva uporabe, ovaj rad ima za cilj pomoći čitateljima stvoriti bolje razumijevanje i koji pristup, na kraju krajeva, odabrati za dizajniranje baze podataka za svoje projektne potrebe.

Ključne riječi: baze podataka, relacijska baza podataka, nerelacijska baza podataka, SQL, NoSQL

Sadržaj

1. Uvod	1
2. Baze podataka	2
3. Relacijske baze podataka (RDBMS)	4
3.1 ERA model.....	5
3.2 Entiteti i atributi	6
3.3 Domena	7
3.4 Veze među entitetima.....	7
3.5 Primarni i vanjski ključevi	8
3.6 SQL	9
3.8 Najpoznatiji sustavi za upravljanje relacijskim bazama podataka	10
4. NoSQL (Not Only SQL) baze podataka.....	12
4.1 Vrste nerelacijskih baza podataka	12
4.1.1 Ključ-vrijednost baze podataka.....	12
4.1.3 Dokumentne baze podataka	15
4.1.4 Stupčane baze podataka	16
5. Izrada relacijske i NoSQL baze podataka	18
5.1 Relacijska baza podataka „Knjižnica“	18
5.2 Izrada NoSQL baze podataka	20
5.2.1 NoSQL baza podataka „Knjižnica“	20
6. Rasprava	23
6.1 Prednosti i nedostaci SQL i NoSQL baza podataka	23
6.1.1 SQL.....	23
6.1.2 NoSQL.....	24
7. Zaključak.....	26
Literatura	28

1. Uvod

U današnjem digitalnom dobu, upravljanje velikim količinama podataka postalo je neizbježan i složen zadatak. Sve veći broj organizacija suočava se s izazovom skladištenja, upravljanja i analiziranja raznolikih podataka kako bi donijele informirane odluke i ostvarile konkurentne prednosti. U tom kontekstu baze podataka igraju ključnu ulogu u organizaciji, pohrani i obradi podataka.

Tradicijski, relacijske baze podataka (RDBMS) dominirale su kao glavni oblik skladištenja podataka. Njihova struktura tablica i temeljni SQL (eng. *Structured Query Language*) omogućili su organizirano i strukturirano pohranjivanje podataka uz podršku transakcija i složenih upita. Međutim, razvojem interneta, društvenih mreža i mobilnih aplikacija, pojavila se potreba za drugačijim pristupom skladištenju podataka.¹

Uzlet NoSQL (eng. *Not Only SQL*) baza podataka označio je početak paradigme koja je pružila alternativu relacijskim bazama podataka. NoSQL baze podataka odlikuju se fleksibilnijim i nestrukturiranim modelima podataka, što ih čini prikladnima za obradu velikih količina nestrukturiranih podataka i visoku razinu skalabilnosti. Ova nova generacija baza podataka donijela je novu dimenziju u upravljanju podacima i pružila mogućnosti za rješavanje problema koji nisu bili adekvatno pokriveni tradicionalnim relacijskim pristupom.²

S ciljem razumijevanja razlika i prednosti između relacijskih i NoSQL baza podataka, ovaj završni rad provodi usporedbu između ove dvije vrste sustava. Analizirat će se njihove karakteristike, prednosti i nedostaci, kao i njihova primjena u različitim scenarijima. Ova usporedba pomoći će organizacijama i razvojnim timovima donijeti informirane odluke pri odabiru prikladne baze podataka za svoje specifične potrebe.

Konačno, cilj ovog završnog rada pružiti je pregled relacijskih i NoSQL baza podataka.

¹ Usp. Carić, Tonči; Buntić, Mario. Uvod u relacijske baze podataka. Zagreb: 2015., str. 2-16. URL: <http://files.fpz.hr/Djelatnici/tcaric/Tonci-Caric-Baze-podataka.pdf> (2020-08-13)

² Usp. Sadalage, Pramod J. & Fowler, Martin, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley Professional, 2012., str. 18-25. URL: <https://bigdata-ir.com/wp-content/uploads/2017/04/NoSQL-Distilled.pdf> (2023-08-03)

2. Baze podataka

Mogućnost trajnog pohranjivanja podataka u računalima postoji gotovo jednako dugo kao i sama računala te je podržano u svim programskim jezicima. Internetska revolucija kasnih 1990-ih dramatično je proširila izravan korisnički pristup bazama podataka. Organizacije su transformirale mnoga telefonska sučelja u web sučelja i omogućile pristup različitim uslugama i informacijama putem interneta. Primjerice, pri posjeti internetskoj knjižari i pretraživanju knjiga ili glazbene kolekcije, zapravo pristupate podacima koji su pohranjeni u njihovoj bazi podataka. Tijekom izvršavanja online narudžbe, ista narudžba evidentira se u bazi podataka. Isto tako, prilikom posjeta web stranici, informacije o osobi koja joj je pristupila mogu se povući iz baze podataka kako bi se odabralo koje reklame treba vidjeti.³

Definicija "Baza podataka je kolekcija podataka, ograničenja i operacija koja reprezentira neke aspekte stvarnog svijeta"⁴ označava kako se ti podaci čuvaju u vanjskoj memoriji računala te su dostupni raznim korisnicima i aplikacijama. Pohranjivanje, promjena, brisanje i čitanje podataka obavlja se putem sustava za upravljanje bazom podataka (DBMS).

Sustav za upravljanje bazom podataka (SUBP, eng. DBMS – *Database Management System*) specijalizirano je programsko rješenje koji posreduje između aplikacija i podataka te se brine za zajedničku kolekciju podataka, odnosno bazu podataka. DBMS poslužitelj je baze podataka odgovoran za oblikovanje fizičkog prikaza baze i izvođenje operacija s podacima. Podržava različite baze s različitim logičkim strukturama, brine se o sigurnosti podataka i automatizira administrativne zadatke.⁵

Strukturalni opis objekata unutar baze podataka poznat je kao shema, koja detaljno opisuje objekte i njihove međusobne odnose u bazi podataka. Modeli baza podataka služe za organizaciju ove sheme i modeliranje strukture baze podataka. Među različitim modelima, najčešće se koristi relacijski model podataka, koji predstavlja podatke putem tablica. Svaka od ovih tablica sastoji se od redaka i stupaca, a relacijski model koristi zajedničke vrijednosti kako

³ Usp. Silberschatz, Abraham, Korth, Henry, F. & Sudarshan, S. Database system concepts. URL: https://www.octawian.ro/fisiere/situri/asor/build/html/_downloads/1fcab53a6d916e39c715fc20a9a9c2a8/Silberschatz_A_databases_6th_ed.pdf

⁴ Usp. Manger, Robert. Baze podataka, Zagreb: Element, 2011., str. 9-13. URL: <https://dokumen.tips/documents/robert-manger-baze-podatakapdf.html?page=1> (2023-07-11)

⁵ *Ibid.*

bi prikazao odnose između različitih tablica.⁶ S druge strane postoji mrežni model među kojima se baza prikazuje mrežom čvorova i usmjerenih veza, gdje čvorovi predstavljaju tipove zapisa, a veze definiraju poveznice između njih. Potom, hijerarhijski model koji je specifičan oblik mrežnog modela gdje je baza organizirana u stablo s čvorovima koji predstavljaju tipove zapisa i vezama "nadređeni-podređeni" između njih. Naposljetku objektni model, inspiriran objektno-orijentiranim programskim jezicima, gdje su podaci predstavljeni kao trajno pohranjeni objekti s atributima i metodama za rukovanje njima.⁷

Sustav upravljanja bazom podataka (SUBP) temelji se na odabranom modelu podataka. Njegovi zadaci uključuju: Zaštitu objekata baze podataka od neovlaštenog pristupa, očuvanje integriteta podataka u bazi, mogućnost oporavka podataka u slučaju gubitka, osiguravanje višekorisničkog pristupa podacima u bazi istovremeno, kako bi se osigurala konkurentnost, prikazivanje opisa podataka kroz metapodatke, identificiranje optimalne strukture za učinkovito upravljanje, opis i rukovanje podacima unutar baze podataka.⁸

⁶ Carić, Tonči; Buntić, Mario. Str. 2. Nav. dj.

⁷ Manger, Robert. Nav. dj.

⁸ Carić, Tonči; Buntić, Mario. Str. 3. Nav. dj.

3. Relacijske baze podataka (RDBMS)

Relacijski podatkovni model kreiran je od strane engleskog matematičara Edgara Franka Codd-a i predstavljen 1970. godine u njegovom članku "A Relational Model of Data for Large Shared Data Banks".⁹ Tada je Codd bio zaposlenik IBM-a, međutim usprkos njegovim očekivanjima, IBM nije prvi implementirao ovaj model, nego su se držali zastarjelog hijerarhijskog modela u sustavu IMS. Prva tvrtka koja je uspješno implementirala relacijski model podataka bila je Oracle, koja i danas predstavlja sinonim za baze podataka.

Codd-ov relacijski upitni jezik Alpha nije bio prihvaćen, umjesto njega prihvaćen je nerelacijski jezik SEQUEL (eng. *Structured English Query Language*). Prvobitno ime SEQUEL moralo je biti promijenjeno u SQL radi zaštite naziva SEQUEL od strane jedne zrakoplovne kompanije.

Edgar se tijekom godina suočavao s različitim proizvođačima sustava upravljanja bazama podataka (SUBP-ova) koji nisu ispravno ili potpuno implementirali relacijski podatkovni model, iako su ga nazivali relacijskim. Kao rezultat njegove borbe, objavio je dvanaest pravila koja SUBP moraju poštivati kako bi bili smatrani relacijskim:

1. Pravilo predstavljanja informacija - podaci se reprezentiraju jedinstveno kao vrijednosti u relacijama (tablicama).
2. Pravilo pristupa - svaki podatak mora biti dostupan preko kombinacije imena tablica, vrijednosti primarnog ključa i imena atributa.
3. Tretiranje nepoznatih vrijednosti - vrijednost NULL tretira se kao nepoznata vrijednost neovisno o tipu podatka.
4. Dinamički online katalog - rječnik baze podataka mora biti pohranjen kao i ostali podaci u bazi.
5. Pravilo sveobuhvatnog jezika - mora postojati jezik za komunikaciju s bazom podataka koji podržava relacijske operatore za modifikaciju, definiciju i administraciju podataka.
6. Pravilo pogleda - svi pogledi moraju biti ažurirani radi SELECT naredbe koja dohvaća podatke.
7. Pravila ažuriranja skupova - operacije umetanja, ažuriranja i brisanja moraju biti podržane za skupove podataka.

⁹ Usp. IBM's 100 Icons of Progress. URL: <https://www.ibm.com/ibm/history/ibm100/us/en/icons/reldb/> (2023-07-19)

8. Nezavisnost fizičkih podataka - aplikacije ne smiju ovisiti o promjenama u fizičkom načinu spremanja podataka.
9. Nezavisnost logičkih podataka - promjene u odnosima između tablica ne smiju utjecati na funkcionalnost aplikacija.
10. Nezavisnost integriteta podataka - SUBP se mora brinuti o integritetu baze podataka, a ne druge aplikacije.
11. Distribuirana nezavisnost - aplikacija mora nastaviti raditi kada se uvede distribuirana verzija SUBP-a ili se ta verzija centralizira.
12. Pravilo o nenarušavanju - integritet podataka ne smije biti narušen obilaženjem pravila integriteta i ograničenja.

Danas se SUBP smatra relacijskim ako poštuje šest od ovih trinaest pravila. Relacijski model, iako nije najbrži, smatra se najfleksibilnijim i najraširenijim modelom za upravljanje bazama podataka u suvremenom svijetu.¹⁰

Relacijska baza podataka temelji se na organizaciji podataka u povezane relacije, osiguravajući ažurnost, kontrolu pristupa, sigurnosno arhiviranje, centraliziranu administraciju i složene upite za analizu podataka. Osim toga, održava minimalnu redundanciju, postojanost, točnost i integritet pohranjenih podataka. Organizira podatke u tablice (relacije) koje sadrže stupce i retke s jedinstvenim ključem. Svaka tablica predstavlja tip entiteta, redovi su instance tih entiteta, a stupci predstavljaju vrijednosti dodijeljene tim instancama, odnosno njihovim atributima.¹¹

3.1 ERA model

Model podataka entitet-veza (E-R) koristi skup osnovnih objekata, nazvanih entiteti i veza između tih objekata. Entitet predstavlja "stvar" ili "objekt" u stvarnom svijetu koji se razlikuje od drugih objekata.

Ukupna logička struktura (shema) baze podataka može se grafički prikazati pomoću dijagrama entitet-veza (E-R). Pri izradi skupovi entiteta prikazuju se pravokutnim okvirom s nazivom skupa entiteta iznad okvira, dok se atributi nabrajaju ispod okvira. Skupovi veza

¹⁰ Carić, Tonči; Buntić, Mario. Str. 8-10. Nav. dj.

¹¹ Usp. Harrington, Jan, L. Relational Database Design and Implementation, Burlington, MA, USA: Morgan Kauffman. 2009.

prikazuju se dijamantom koji povezuje par povezanih skupova entiteta, a naziv veze nalazi se unutar romba.

Osim entiteta i veza, E-R model također definira različita ograničenja koja moraju biti zadovoljena u bazi podataka. Jedno od važnih ograničenja je ograničenje kardinalnosti koje određuje koliko entiteta u jednom skupu može biti povezano s drugim skupom putem odnosa.¹² Primjerice, E-R model može izraziti ograničenje da svaki autor mora biti povezan samo s jednom knjigom.

Model entitet-veza široko se koristi u procesu dizajniranja baze podataka.¹³

3.2 Entiteti i atributi

Entiteti su osnovne jedinice podataka u RDBMS-u. Svaki entitet predstavlja određeni koncept, objekt ili događaj o kojem želimo prikupljati podatke.¹⁴ Primjeri entiteta mogu uključivati "autore" u bazi podataka za upravljanje knjižnicom. Svaki entitet obično ima jedinstven identifikator poznat kao primarni ključ koji ga jednoznačno identificira.

Atributi su karakteristike ili svojstva entiteta koja opisuju njegovu prirodu ili svojstva. Služe za pohranu detaljnih informacija o njima. Mogu imati određene tipove podataka kao što su tekst, brojevi, datumi i drugi što osigurava precizno definiranje formata podataka.¹⁵ Primjerice, za entitet "autor" atributi mogu uključivati ime, prezime, adresu, broj telefona i e-poštu.

Entiteti u bazi podataka opisuju se skupom atributa. Primjerice, atributi kao što su naziv odjela, zaposlenik i posudba mogu opisati određeni odjel u knjižnici, čineći attribute skupa entiteta odjela. Slično tome, atributi poput ID-a, imena, prezimena i datuma rođenja mogu opisati entitet zaposlenika. Dodatni atribut ID koristi se za jedinstvenu identifikaciju zaposlenika, iz razloga što je moguće postojanje dva zaposlenika s istim imenom i prezimenom. Jedinstveni identifikator zaposlenika mora biti dodijeljen svakom zaposleniku.¹⁶

Tablica 1: Zaposlenik

¹² Usp. Silberschatz, Abraham, Korth, Henry, F. & Sudarshan, S. Database system concepts. URL: https://www.octawian.ro/fisiere/situri/asor/build/html/_downloads/1fcab53a6d916e39c715fc20a9a9c2a8/Silberschatz_A_databases_6th_ed.pdf

¹³ *Ibid.*

¹⁴ Silberschatz, Abraham, Korth, Henry, F. & Sudarshan, S. Nav. dj.

¹⁵ Harrington, Jan, L. Nav. dj.

¹⁶ Silberschatz, Abraham, Korth, Henry, F. & Sudarshan, S. Nav. dj.

ID	Ime	Prezime	Datum_Rodenja
1	Dragica	Likić	17.09.1997.
2	Hana	Bijelić	23.01.1995.
3	Marisa	Draginić	10.12.1997.

Veza između entiteta i atributa ostvaruje se putem relacija ili tablica u relacijskoj bazi podataka. Svaki entitet obično ima svoju vlastitu tablicu, gdje su redci u tablici zapisi ili instance tog entiteta, a stupci u tablici predstavljaju njegove attribute. Ovaj organizacijski pristup omogućava učinkovito pohranjivanje, pretraživanje i upravljanje podacima te olakšava izradu upita za dohvaćanje informacija o entitetima i njihovim atributima.

Entiteti i atributi temeljni su koncepti u relacijskim bazama podataka koji omogućavaju strukturiranje i organizaciju podataka na jasan i dosljedan način. Definiranje entiteta i njihovih atributa ključno je za uspješno upravljanje informacijama i omogućava efikasno korištenje RDBMS-a za različite aplikacije i poslovne potrebe.¹⁷

3.3 Domena

Za svaki atribut relacije postoji skup dopuštenih vrijednosti koji se naziva domena tog atributa. Osim toga što predstavlja domena, važno je i kako koristimo elemente domene u našoj bazi podataka. Većina sustava, koji koriste SQL kao upitni jezik, pruža različite tipove podataka za definiranje domene atributa, kao što su CHAR (fiksni broj znakova), VARCHAR (varijabilni broj znakova), INT (cijeli broj pohranjen u 4 byte-a), DATE (datum) te druge tipove kao što su DECIMAL, BOOLEAN, TIME, DATETIME, SMALLINT, BIT, FLOAT, NUMERIC itd.¹⁸

3.4 Veze među entitetima

Veze među entitetima ključna su komponenta relacijskih baza podataka (RDBMS) iz razloga što omogućuju organizaciju podataka i povezivanje informacija iz različitih tablica. Ove veze često nazivamo "relacijama" ili "stranim ključevima" i igraju ključnu ulogu u modeliranju složenih stvarnosti i relacija iz stvarnog svijeta.

RDBMS pruža različite vrste veza između entiteta, a to su:

veza jedan na jedan (eng. *one-to-one*) 1:1,

¹⁷ Harrington, Jan, L. Nav. dj.

¹⁸ Usp. Livaja, Ivan. Uvod u baze podataka, Šibenik, 2016.

veza jedan na više (eng. *one-to-many*) 1:M i

veza više na više (eng. *many-to-many*) M:N.

Jedan-na-jedan veze označavaju kako svaki entitet u jednoj tablici ima točno jedan povezani entitet u drugoj tablici, jedan-na-više odnosi se na to kako svaki entitet može imati više povezanih entiteta u drugoj tablici, dok više-na-više veze predstavljaju kako entiteti iz jedne tablice mogu biti povezani s više entiteta iz druge tablice i obratno.

Korištenjem stranih ključeva i ograničenja referencijalnog integriteta, RDBMS osigurava dosljednost podataka, odnosno kako se ne može izbrisati entitet koji je referenciran iz druge tablice, osim ako se ne uklone povezane veze ili se primijene određene akcije (npr. brisanje tablica).

Veze među entitetima služe za izvođenje složenih upita koji povezuju podatke iz različitih tablica i osiguravaju modeliranje složenih stvarnih relacija. Ne mora nužno svaka instanca svakog entiteta biti povezana s drugom instancom, osim ako eksplicitno specificiramo kako je veza obavezna.¹⁹ Primjerice, u sustavu knjižnice, korisnik knjižnice ne mora nužno posuđivati knjigu u svakom trenutku.

3.5 Primarni i vanjski ključevi

Primarni i vanjski ključevi ključni su koncepti u relacijskim bazama podataka (RDBMS) koji omogućuju povezivanje podataka iz različitih tablica i održavanje integriteta podataka. Ovi ključevi igraju ključnu ulogu u modeliranju i upravljanju složenim bazama podataka.

Primarni ključ (eng. *Primary Key*) jedinstveni je identifikator za svaki zapis u tablici. Primjerice, u relaciji "zaposlenik," atribut "ID" dovoljan je kako bi se jedinstveno razlikovao jedan zaposlenik od drugog stoga je "ID" primarni ključ. S druge strane, atribut "ime zaposlenika" nije primarni ključ iz razloga što više zaposlenika može imati isto ime. Primarni ključevi biraju se pažljivo, a trebaju biti takvi da se njihove vrijednosti rijetko ili nikada ne mijenjaju kako bi osigurali stabilnost identifikacije. Između ostalog, primarni ključ ne smije biti prazan (null).²⁰

¹⁹ *Ibid.*

²⁰ Usp. Maleković Mirko i Rabuzin Kornelije. Uvod u baze podataka. Varaždin: Fakultet organizacije i informatike, 2016.

Vanjski ključ (eng. Foreign Key) atribut je ili skup atributa u tablici relacijske baze podataka koji upućuje na primarni ključ druge tablice. Ovaj mehanizam omogućava uspostavljanje veza među tablicama.²¹ Primjerice, ukoliko imamo tablicu "Autor" i tablicu "Knjiga", vanjski ključ u tablici "Autor" može se odnositi na primarni ključ tablice "Knjiga" kako bi se povezala knjiga s njezinim autorom.

```
knjiga (id, naslov, ISBN)
```

```
autor (id, id_knjiga, ime, prezime, datum_rodenja)
```

Primarni i vanjski ključevi omogućuju složene upite i analize koji pretražuju, povezuju i izvlače informacije iz različitih tablica, što je ključno kada želimo dohvatiti podatke iz relacijske baze podataka.²²

3.6 SQL

SQL (eng. *Structured Query Language*) standardizirani je jezik za upravljanje i manipulaciju podacima u relacijskim bazama podataka (RDBMS). Ključni je alat za komunikaciju s bazom podataka i omogućava izradu, ažuriranje, oblikovanje upita i upravljanje podacima unutar RDBMS-a.²³

SQL služi za upravljanje i dijeljenje podataka, posebno u relacijskim bazama podataka koji uključuju podatke organizirane u tablicama. Pomoću SQL-a mogu se postavljati upiti, ažurirati i reorganizirati podaci, kao i stvarati i mijenjati struktura sustava baze podataka i kontrolirati pristup njegovim podacima. Neizostavan je alat za razvoj, upravljanje i analizu podataka u relacijskim bazama podataka. Njegova standardizirana sintaksa olakšava interakciju s bazama podataka na različitim platformama i čini ga neophodnim alatom za programere, analitičare podataka i administratore baza podataka.²⁴

Sastoji se od nekoliko podjezika koji se koriste za različite svrhe ili u različitim okruženjima:

²¹ *Ibid.*

²² Usp. SQL Notes for Professionals. URL: <https://goalkicker.com/SQLBook>

²³ Usp. Rabuzin, Kornelije. Uvod u SQL. Varaždin: Fakultet organizacije i informatike, 2011.

²⁴ Usp. Pickett, Patricia. What Is SQL?, 2020. URL: <https://www.thebalancecareers.com/what-is-sql-and-uses-2071909> (2023-08-20)

1. Jezik za opis podataka (eng. *Data Description Language* - DDL) koristi se za definiranje strukture baze podataka. Ovdje pripadaju naredbe za stvaranje i brisanje tablica, definiranje ključeva, indeksa i ograničenja integriteta. Primjeri DDL naredbi uključuju CREATE (TABLE), ALTER (TABLE) i DROP (TABLE).
2. Jezik za manipuliranje podacima (eng. *Data Manipulation Language* - DML) koristi se za manipulaciju podacima unutar baze podataka. Naredbe ovog jezika jesu naredbe za unos, izmjenu i brisanje podataka u tablicama. Najpoznatija DML naredba je SELECT za dohvaćanje podataka, dok se INSERT, UPDATE i DELETE koriste za unos, izmjenu i brisanje podataka.
3. Jezik za postavljanje upita (eng. *Data Control Language*) poznat kao *Query Language* (QL), specijaliziran za upite nad podacima. Glavna naredba ovog podjezika je SELECT.²⁵

3.8 Najpoznatiji sustavi za upravljanje relacijskim bazama podataka

Baze podataka obično zahtijevaju upravljački softver koji se zove sustav za upravljanje bazom podataka (eng. *Database Management System* - DBMS). DBMS djeluje kao posrednik između baze podataka i korisnika ili programa, omogućujući korisnicima dohvaćanje, ažuriranje i manipulaciju podacima. Razlikujemo nekoliko najpoznatijih sustava za upravljanje relacijskim bazama podataka.²⁶

MariaDB MySQL RDBMS otvorenog koda ističe se brzinom i jednostavnošću. Nedvojbeno najčešća je baza podataka koja se koristi u web aplikacijama zbog svojih brzih performansi i integracije s različitim jezicima.²⁷ Izvorno razvijen od strane MySQL AB, potom preuzet od Oracle Corporation dostupan je na više od 20 platformi. Veliki igrači poput Facebooka, Twittera i YouTubea služe se MySQL-om, što svjedoči o njegovoj popularnosti.²⁸ MySQL temelji se na klijent-server modelu, odnosno server izvršava upute i izvodi naredbe unutar baze podataka. Njegove mogućnosti uključuju podršku za ogromne baze podataka, replikaciju podataka i defragmentirane tablice za poboljšane performanse i izdržljivost.²⁹

²⁵ Livaja, Ivan. Nav. dj.

²⁶ Manger, Robert. Nav. dj.

²⁷ Usp. What is MySQL? URL: <https://www.atlantic.net/dedicated-server-hosting/what-is-mysql/> (2023-08-02)

²⁸ Usp. Moore, Lindsay. MySQL. URL: <https://searchoracle.techtarget.com/definition/MySQL> (2023-08-02)

²⁹ *Ibid.*

Oracle Database bez daljnijega vođa je sustava za upravljanje relacijskim bazama podataka. Široko je upotrebljavan i popularan zbog svojih visokih kompetencija te vrlo malo nedostataka.³⁰

PostgreSQL relativno nov u području RDBMS-a, vrlo je brzo postigao visoku reputaciju. Besplatan je i otvorenog koda, podržava različite vrste podataka, dostupan na različitim platformama te podržava složene upite i punu tekstnu pretragu. Zbog navedenih prednosti i još mnogih drugih, čest je izbor za organizacije koje traže pouzdan DBMS.³¹

SQLite poseban je zbog ugrađenog sustava za upravljanje bazama podataka gdje je baza podataka pohranjena u jednoj datoteci, što ga čini brzim i praktičnim za lokalnu pohranu podataka u aplikacijama.³²

Microsoft SQL Server moćan je i široko korišten sustav za upravljanje bazama podataka (DBMS) koji nudi niz naprednih mogućnosti i alati za organizacije i razvojne timove.³³

Jedna od ključnih prednosti Microsoft SQL Servera njegova je integracija s ekosustavom Microsoft proizvoda, što olakšava njegovo korištenje i upravljanje organizacijama koje već koriste Microsoft alate. Nudi napredne mogućnosti za upravljanje pristupom podacima i sigurnošću, uključujući mogućnosti za autentifikaciju, autorizaciju i enkripciju podataka. Također podržava transakcije i ACID svojstva, čime se osigurava integritet podataka. Osim toga, nudi niz alati za upravljanje i razvoj baza podataka, između ostalih i SQL Server Management Studio (SSMS) za upravljanje i upite baza podataka te Visual Studio za razvoj aplikacija koje koriste SQL Server kao backend.³⁴

³⁰ Usp. Suehring, Steve. *MySQL Bible*, Wiley Publishing, Inc. URL: https://box.cs.istu.ru/public/docs/other/_New/Books/Data/DB/MySQL/MySQL%20Bible.pdf (2023-08-30)

³¹ *Ibid.*

³² Usp. What is SQLite? URL: <https://www.codecademy.com/article/what-is-sqlite> (2023-08-02)

³³ Usp. What is SQL Server. URL: <https://www.sqlservertutorial.net/getting-started/what-is-sql-server/> (2023-08-02)

³⁴ Usp. MS SQL Server – Overview. URL:

https://www.tutorialspoint.com/ms_sql_server/ms_sql_server_overview.htm (2023-08-02)

4. NoSQL (Not Only SQL) baze podataka

NoSQL (Not Only SQL) baze podataka predstavljaju revolucionarnu promjenu u svijetu pohrane i upravljanja podacima. Za razliku od tradicionalnih relacijskih baza podataka, NoSQL pristup osigurava veću fleksibilnost i skalabilnost u obradi različitih vrsta podataka. Ove baze često se koriste za pohranu nestrukturiranih ili polustrukturiranih podataka poput JSON ili XML dokumenata, senzorskih podataka ili podataka s društvenih mreža. Prikladne su za sustave koji zahtijevaju brzu obradu velikih količina podataka i visoku dostupnost. Pojam NoSQL potječe od Carla Strozzi 1998. godine koji je upotrijebio navedeni pojam kako bi opisao svoju Open Source, Light Weight bazu podataka koja nije imala SQL sučelje.³⁵

NoSQL baze podataka omogućuju prilagodljiv pristup pohrani podataka odnosno mogu se efikasno koristiti u različitim scenarijima, od velikih web aplikacija do analize velikih skupova podataka, nudeći alternativu tradicionalnim relacijskim bazama podataka i prilagođavajući se potrebama modernih informacijskih sistema.³⁶

Nude fleksibilne modele pohrane, kao što su dokumenti, grafikoni ili ključevi-vrijednosti, pružajući programerima odabir najprikladnijeg modela za specifične potrebe njihovih aplikacija. Ova raznolikost modela olakšava rješavanje različitih izazova u pohrani i obradi podataka.³⁷

4.1 Vrste nerelacijskih baza podataka

Iako postoje različite vrste NoSQL baza podataka, u ovom radu opisane su četiri glavne tehnologije:

1. Ključ-vrijednost baze podataka (eng. *Key-value databases*)
2. Grafovske baze podataka (eng. *Graph databases*)
3. Dokumentne baze podataka (eng. *Document databases*)
4. Stupčaste baze podataka (eng. *Columnar databases*)³⁸

4.1.1 Ključ-vrijednost baze podataka

Ključ-vrijednost baze podataka tip su NoSQL baza koje čuvaju podatke u jednostavnom formatu parova ključ-vrijednost. Svaki zapis sastoji se od jedinstvenog ključa koji identificira podatak i

³⁵ Usp. NoSQL (26.2.2020.). URL: <https://www.w3resource.com/mongodb/nosql.php> (2023-08-05)

³⁶ Usp. Stojanović, Aleksandar. Osvrt na NoSQL baze podataka - Četiri osnovne tehnologije, Polytechnic & Design, 2016. URL: <https://hrcak.srce.hr/clanak/283391> (2023-08-09)

³⁷ NoSQL. Nav. dj.

³⁸ Stojanović, Aleksandar. Nav. dj.

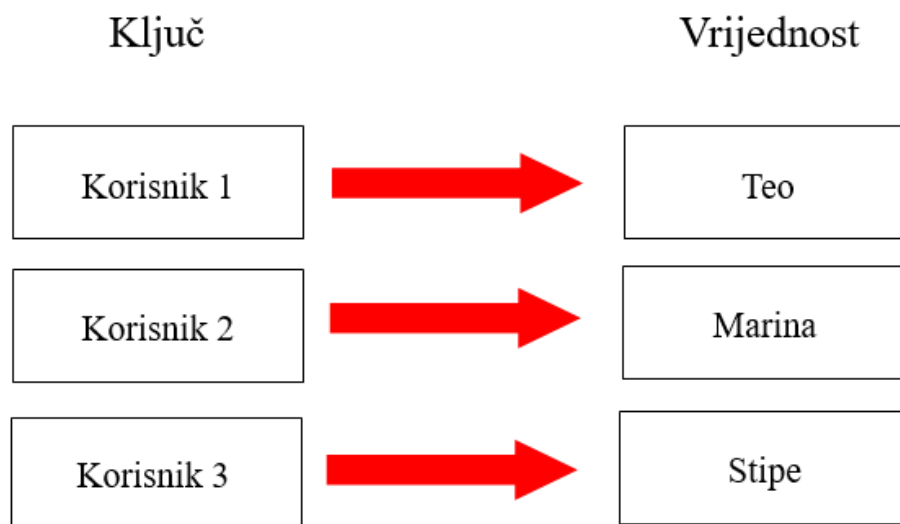
odgovarajuće vrijednosti koja može biti bilo koja informacija ili objekt. Ovaj koncept daje brz pristup podacima zato što se traženi podatak lako nalazi putem ključa što je posebno korisno za česte operacije čitanja i upisivanja. Iako ovi sustavi pružaju visoke performanse, nisu idealni za složene modele podataka.

Podržavaju tri osnovne operacije:

GET (ključ) – naredba koja vraća pridruženu vrijednost zadanom ključu

PUT (ključ, vrijednost) – naredba koja dodaje novi par ili pridružuje novu vrijednost postojećem ključu

DELETE (ključ) – naredba koja uklanja par iz baze.³⁹



Slika 5. Primjer ključ-vrijednost sustava

Navedene baze podataka, osnovane na modelu podataka s parovima oblika (ključ, vrijednost), osiguravaju povezivanje bilo kojeg tipa vrijednosti s tekstualnim ključem. Bez prisutnog upitnog jezika, nude rad temeljen na ključu, dozvoljavajući pronalaženje, dodavanje i uklanjanje parova.⁴⁰

Primjeri popularnih ključ-vrijednost baza podataka uključuju Redis i Amazon DynamoDB.

³⁹ *Ibid.*

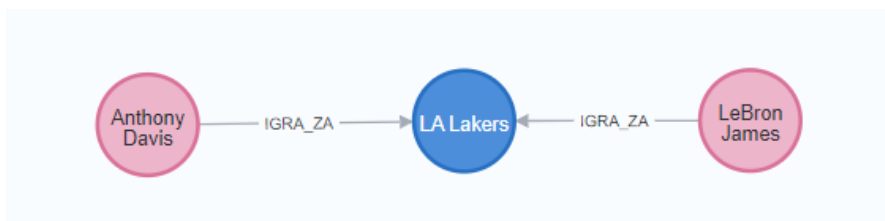
⁴⁰ Usp. What is a Key-Value Database? URL: <https://database.guide/what-is-a-key-value-database/>

Redis čuva cijeli set podataka u memoriji s povremenim pisanjem na disk. Moguće je konfigurirati ga spremati podatke nakon određenog broja sekundi, ali u slučaju prekida sustava, nedavne promjene mogu biti izgubljene. Redis se ubraja među najbrže sustave ovog tipa, a posebno je koristan kada manji gubitak podataka nije ključan. Izvrstan je izbor za pohranu podataka u stvarnom vremenu koji se često ažuriraju.⁴¹

Amazon DynamoDB visoko je dostupna i skalabilna ključ-vrijednost i dokument-orijentirana baza podataka koju pruža Amazon Web Services (AWS). DynamoDB namijenjen je rukovanju velikim količinama podataka uz brze i predvidive performanse. Raspoređen je na globalnom nivou i nudi visoku dostupnost i otpornost na smetnje.⁴²

4.1.2 Grafovske baze podataka

Graf-sustavi temelje se na grafičkom modelu i manje su uobičajeni, ali izuzetno korisni za rad s podacima koji su bogato povezani.⁴³



Slika 6. Primjer grafovske baze podataka

Podaci u ovim bazama podataka organizirani su kao čvorovi (entiteti) i lukovi (veze) između tih čvorova, stvarajući graf koji olakšava prikazivanje kompleksnih odnosa između entiteta. Spomenuti model podataka čini grafovske baze posebno korisnim za scenarije kao što su društvene mreže, preporuke proizvoda, analizu mreža i usmjeravanje u telekomunikacijama. Primjer popularne grafovske baze podataka predstavlja Neo4j.

Ono što izdvaja grafovske baze podataka brz je i efikasan način za traženje i analizu podataka koji uključuju kompleksne veze. Pružaju dubinske i široke upite koji se lako izvode, što olakšava otkrivanje skrivenih uzoraka i korisnih informacija unutar podataka grafa. Osim toga, grafovske baze podataka često podržavaju svoj jezik upita (npr. Cypher za Neo4j), posebno prilagođen za rad s graf podacima, što olakšava razvoj aplikacija koje zahtijevaju kompleksne

⁴¹ Stojanović, Aleksandar. Nav. dj.

⁴² Usp. What is AWS? URL: https://aws.amazon.com/what-is-aws/?nc1=f_cc (2023-09-01)

⁴³ Stojanović, Aleksandar. Nav. dj.

upite i analizu međusobnih veza između entiteta. Grafovske baze podataka tako su postale ključne aplikacijama koje se oslanjaju na razumijevanje i analizu složenih odnosa između podataka.⁴⁴

4.1.3 Dokumentne baze podataka

Dokument-baze podataka trenutno su najpopularniji tip NoSQL baza podataka, a jedan od razloga za to predstavlja intuitivan model podataka koji nude. Osnovni element ovog modela podataka je dokument, koji predstavlja uređeni skup ključeva i njima pridruženih vrijednosti. Dokumenti se grupiraju u kolekcije, što može biti analogno tablicama u relacijskim bazama podataka, gdje svaki dokument odgovara jednom redu u tablici.

Za razliku od relacijskih baza podataka, dokument-sustavi sadrže dinamičke sheme, što znači da svaki dokument u istoj kolekciji može imati različitu strukturu, što pruža fleksibilnost. Primjerice, jedna kolekcija može sadržavati informacije o knjigama, druga o odjelima knjižnice, treća o autorima i tako dalje. Isto tako, ne moraju se pohranjivati NULL vrijednosti ukoliko nisu potrebne. Na taj način, dokument-sustavi omogućuju bolje performanse i optimizaciju resursa.

Dokument-sustavi često koriste JSON format za pohranu podataka radi njegove jednostavnosti i lakšeg prevođenja u JSON iz različitih programskih jezika, bez potrebe za ORM-om (eng. *object-relational mapper*).

Mogu imati hijerarhijsku strukturu, gdje jedan dokument može biti ugniježđen unutar drugog, pružajući time fleksibilnost korisnicima, posebno onima koji dolaze iz okoline objektno-orijentiranog programiranja.⁴⁵

JavaScript je jezik koji se koristi za rad s podacima pa upit kojim izlistavamo sve podatke koji su uneseni u tablici „korisnik“ izgleda ovako:

```
db.korisnik.find()
```

⁴⁴ Usp. What is a Graph Database? URL: <https://neo4j.com/developer/graph-database/>

⁴⁵ Stojanović, Aleksandar. Nav. dj.

```

Knjiznica> db.korisnik.find()
[
  {
    _id: ObjectId("65003f5c53198703b6145e01"),
    ime: 'Karlina',
    prezime: 'Topčić',
    godina_rodenja: 1999
  },
  {
    _id: ObjectId("65004b4853198703b6145e07"),
    ime: 'Lota',
    prezime: 'Vučko',
    godina_rodenja: 1975
  },
  {
    _id: ObjectId("65004b4853198703b6145e08"),
    ime: 'Sandra',
    prezime: 'Honda',
    godina_rodenja: 2000
  },
  {
    _id: ObjectId("65004b4853198703b6145e09"),
    ime: 'Marko',
    prezime: 'Ćoro',
    godina_rodenja: 1969
  }
]
Knjiznica> █

```

Slika 7. Primjer dokumentne baze podataka

4.1.4 Stupčane baze podataka

Stupčani sustavi (eng. *Column-Oriented System - COS*), kao što njihovo ime sugerira, stavljaju naglasak na pojedinačne stupce podataka, slično kao stupci u relacijskim tablicama. Međutim, u usporedbi s relacijskim bazama podataka, stupčane baze podataka omogućuju pristup i manipulaciju pojedinačnim stupcima neovisno o drugim stupcima, što je ključna prednost u smislu performansi.⁴⁶

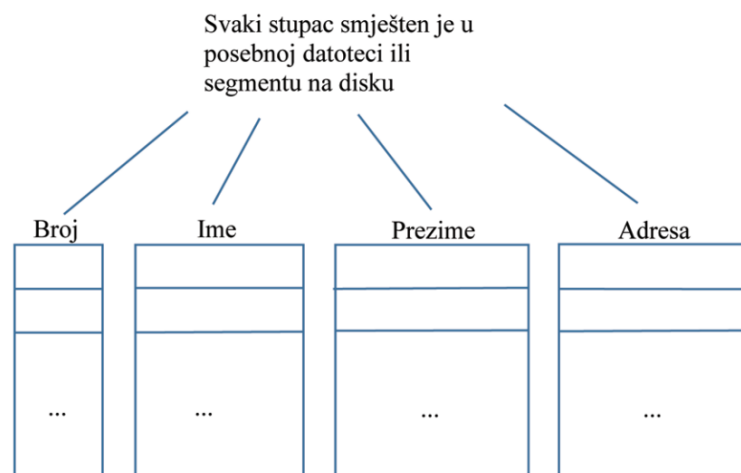
Vrijednosti za svaki stupac pohranjuju se na disku u zasebnim prostorima, omogućujući efikasno učitavanje pojedinog stupca u memoriju. Ova karakteristika izuzetno je korisna kada upiti zahtijevaju samo određene stupce, jer ne zahtijeva učitavanje cijele tablice. Primjerice, kod

⁴⁶ Maleković Mirko i Rabuzin Kornelije. Nav. dj.

tablice koja zauzima 15 GB, gdje je potreban samo 1 GB podataka iz jednog stupca, samo tih 1 GB će biti učitano, dok bi za tradicionalnu tablicu bilo potrebno učitati svih 15 GB.

Navedene baze podataka često nude mogućnosti poput particioniranja (raspodjele podataka na više servera), agregacije (kombiniranje i transformacija podataka) i posebne vrste kolekcija radi bolje prilagodljivosti. Obično ne podržavaju operacije pridruživanja (join), što ih čini pogodnima za distribuirane sustave i skalabilnost.⁴⁷

HBase tipičan je primjer sustava orijentiranog prema stupcima, iako se može činiti sličan relacijskim bazama podataka. Međutim, tablice u HBase-u ne djeluju kao relacije, redovi nisu uobičajeni zapisi i struktura stupaca fleksibilna je i nije unaprijed definirana shemom. Mnoge ugledne tvrtke koriste HBase, uključujući eBay i Twitter.⁴⁸



Slika 8. Primjer stupčane baze podataka

(Izvor: Stojanović, 2016.)

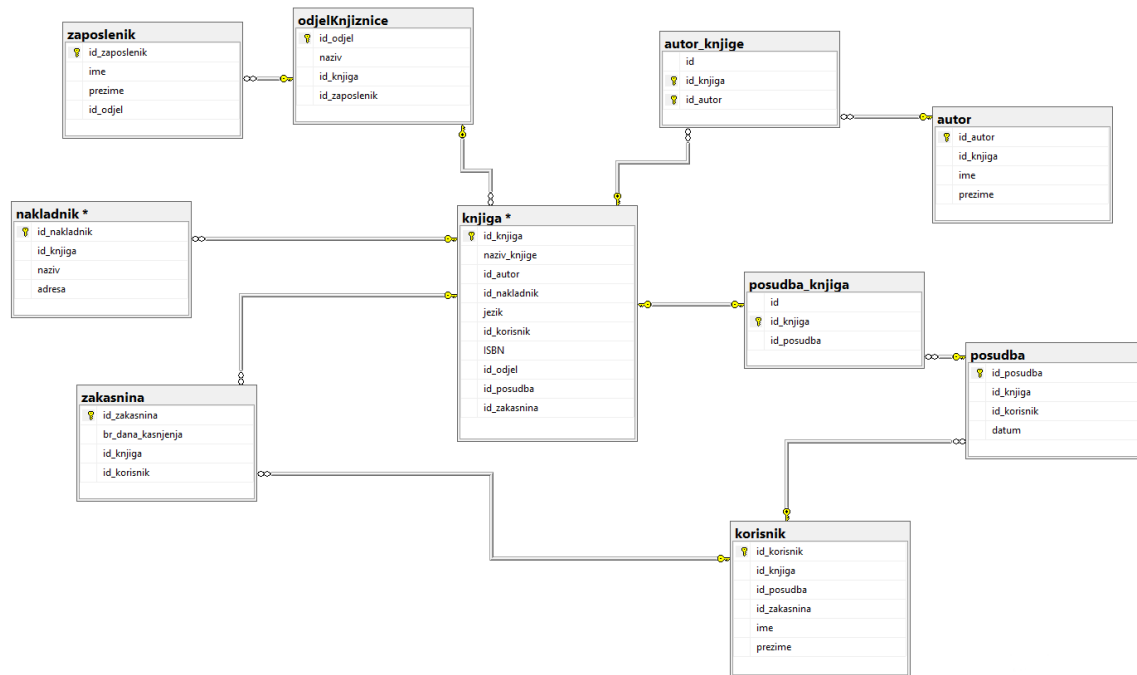
⁴⁷ Stojanović, Aleksandar. Nav. dj.

⁴⁸ Maleković Mirko i Rabuzin Kornelije. Nav. dj.

5. Izrada relacijske i NoSQL baze podataka

5.1 Relacijska baza podataka „Knjižnica“

Slika 9. prikazuje grafički prikaz ER modela koji je izrađen u MySQL bazi podataka koristeći alat MySQL server. Baza podataka pod nazivom „Knjižnica” sastoji se od sveukupno 10 tablica s odgovarajućim atributima koji su u nastavku objašnjeni. Također, svaki atribut ima označen tip podataka koje određeno polje može primiti.



Slika 9. ER model baze podataka „Knjižnica“

Osnova svake baze podataka su tablice, kao što je prikazano na slici 9. U ovom kontekstu postoje tablice za autore, knjige, korisnike, posudbe, zakasnine, nakladnike, zaposlenike, odjele knjižnice i veze između knjiga i autora te posudbe i knjige. Ove tablice su organizirane kako bi se postigla međusobna neovisnost podataka unutar njih. To se postiže uspostavljanjem relacijskih veza između tablica, što olakšava povezivanje i poboljšava preglednost. Ovaj pristup pomaže definirati jasne ciljeve izrade cijele baze podataka.

Tablica „knjiga“

Tablica knjiga ima svrhu za unos novih knjiga ili uređivanje postojećih. Primarni ključ ove tablice je „id_knjiga“, dok su vanjski ključevi „id_odjel“ koji ju povezuje s tablicom „odjelKnjiznice“.

Tablica „autor_knjige“

Kao i tablica „posudba_knjiga“, ova tablica također predstavlja „tablicu sjecište“ i koristi se za povezivanje dviju tablica koje imaju više-naprema-više vezu. Konkretno, povezuje tablice „knjiga“ i „autor“. Primarni ključ ove tablice je „id“, dok su vanjski ključevi „id_knjiga“ koji se povezuje s tablicom „knjiga“ i „id_autor“ koji se povezuje s tablicom „autor“.

Tablica „autor“

Tablica „autor“ sadrži osnovne informacije o autorima koji su napisali određene knjige i služi za unos novih autora. Njen primarni ključ je „id_autor“, koji se povezuje s vanjskim ključem „id_autor“ u tablici „autor_knjige“.

Tablica „posudba_knjiga“

Tablicu „posudba_knjiga“ često nazivamo "tablicom sjecište" jer služi kao poveznica između tablica "knjiga" i "posudba". Svaka knjiga može biti posuđena više puta, dok svaka posudba može uključivati više knjiga istovremeno. Kako većina sustava za upravljanje bazama podataka ne podržava direktnu više-naprema-više vezu između tablica, potrebno je stvoriti treću tablicu koja uspostavlja ovu vezu. Primarni ključ tablice "posudba_knjiga" je "id", dok su vanjski ključevi "id_knjiga" koji se povezuje s tablicom "knjiga" i "id_posudba" koji se povezuje s tablicom "posudba".

Tablica „posudba“

Tablica „posudba“ sadrži informacije o posudbi knjiga i omogućuje unos novih posudbi. Njen primarni ključ je „id_posudba“, koji se povezuje s vanjskim ključem „id_posudba“ u tablici „posudba_knjiga“. Također, vanjski ključ ove tablice je „id_korisnik“, koji se povezuje s primarnim ključem tablice „korisnik“.

Tablica „korisnik“

Tablica „korisnik“ sadrži informacije o korisnicima koji su članovi knjižnice i pruža mogućnost unosa novih korisnika. Njen primarni ključ je „id_korisnik“.

Tablica „zakasnina“

Tablica „zakasnina“ sadrži informacije o posuđenim knjigama i jamči unos podataka o korisnicima koji su posudili knjige, kao i vremenu kada su knjige posuđene. Njen primarni ključ je „id_zakasnina“, dok su vanjski ključevi „id_knjiga“ koji se povezuje s primarnim ključem tablice „knjiga“ i „id_korisnik“ koji se povezuje s primarnim ključem tablice „korisnik“.

Tablica „nakladnik“

Tablica „nakladnik“ sadrži osnovne informacije o nakladnicima i omogućuje unos novih nakladnika. Njen primarni ključ je „id_nakladnik“, dok je vanjski ključ „id_knjiga“ koji se koristi za povezivanje s primarnim ključem „id_knjiga“ u tablici „knjiga“.

Tablica „zaposlenik“

Tablica „zaposlenik“ sadrži informacije o zaposlenicima koji rade u knjižnici. Njen primarni ključ je „id_zaposlenik“, a vanjski ključ tablice je „id_odjel“ koji se povezuje s primarnim ključem tablice „odjelKnjiznice“.

Tablica „odjelKnjiznice“

Tablica „odjelKnjiznice“ sadrži informacije o odjelu u knjižnici gdje se određena knjiga nalazi i omogućuje unos novih knjiga u odgovarajuće odjele. Njen primarni ključ je „id_odjel“.

5.2 Izrada NoSQL baze podataka

MongoDB besplatan je program otvorenog koda, što znači kako je riječ o bazi podataka usmjerenoj prema dokumentima. Za izradu NoSQL baze podataka prije svega potrebno je instalirati MongoDB Community Server i MongoDB Compass, što je grafičko korisničko sučelje. Bazu podataka i kolekciju moguće je izraditi kroz grafičko sučelje te kroz MongoDB naredbeni redak odnosno MongoDB Shell. MongoDB Shell najbrži je način za povezivanje (i rad s) MongoDB.⁴⁹ Kako bismo provjerili uspješnost instalacije MongoDB Shell-a, potrebno je otvoriti naredbeni redak (engl. *command prompt*) i upisati naredbu:

```
mongosh
```

koja će završiti instalaciju i otvoriti MongoDB Shell gdje možemo započeti s kreiranjem kolekcija.

5.2.1 NoSQL baza podataka „Knjižnica“

Za stvaranje baze podataka u Naredbenom retku MongoDB-a koristimo naredbu "use ime_baze". U kontekstu ovog završnog rada, nazvat ćemo bazu podataka "Knjižnica", stoga će naredba za stvaranje baze podataka izgledati ovako: "use Knjižnica". NoSQL baza podataka Knjižnica (Slika 10.) sastoji se od osam kolekcija koje se mogu gledati kao ekvivalenti tablicama u relacijskim bazama podataka:

⁴⁹ Usp. MongoDB Shell. URL: <https://www.mongodb.com/try/download/shell> (2023-09-02)

Kolekcija "knjiga": Pruža informacije o glavnim atributima knjige u knjižnici, uključujući njezin pripadajući naslov, autora i nakladnika knjige, jezik na kojemu je knjiga napisana i standardni broj (ISBN) te odjel na kojemu je određena knjiga smještena.

Kolekcija "autor": Sadrži glavne atribute autora, uključujući ime i prezime autora te autorovu starost.

Kolekcija "korisnik": Ova kolekcija sadrži ključne atribute o korisniku knjižnice, uključujući njegovo ime, prezime i godinu rođenja.

Kolekcija "odjelKnjiznice": Pruža informacije o glavnim atributima odjela knjižnice uključujući naziv i broj odjela na kojemu se određena knjiga nalazi, naziv knjige te ime i prezime zaposlenika koji radi na tom određenom odjelu.

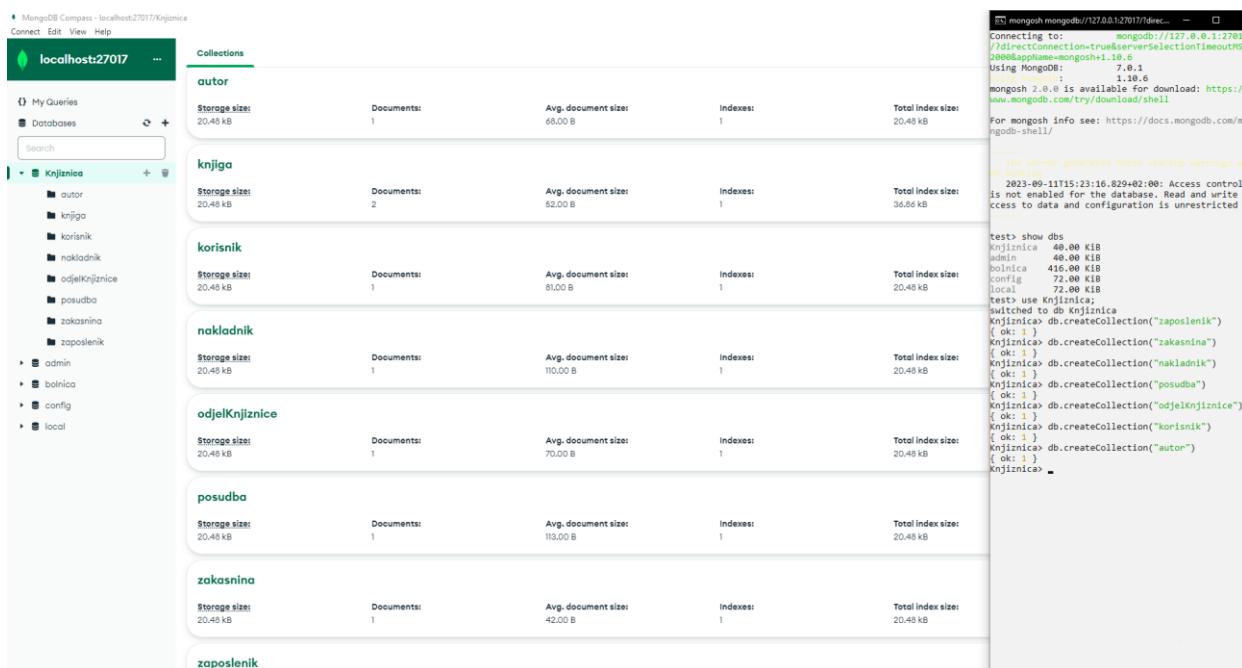
Kolekcija "nakladnik": Sadrži informacije o nakladniku, uključujući naziv nakladničke kuće, adresu i naziv knjige.

Kolekcija "posudba": Ova kolekcija obuhvaća atribute kao što su datum kada je knjiga posuđena, naslov knjige te ime i prezime korisnika koji je posudio knjigu.

Kolekcija "zaposlenik": Omogućuje uvid u glavne atribute zaposlenika kao što su zaposlenikovo ime i prezime te naziv i broj odjela na kojemu radi.

Kolekcija "zakasnina": Sastoji se atributa kao što su dani kašnjenja, naslov knjige na kojoj je zakasnina te ime i prezime korisnika koji nije vratio knjigu na vrijeme.

Ovaj strukturirani pristup bazi podataka omogućuje učinkovito upravljanje informacijama o knjižnici, knjigama i njihovim autorima, zaposlenicima, nakladnicima, zakasninama, odjelima knjižnice te posudbi.



Slika 10. Prikaz izrade baze podataka „Knjiznica“ i njezinih kolekcija

6. Rasprava

Nakon završetka teorijskog dijela ovog rada krenula je izrada relacijske i NoSQL baze podataka radi usporedbe. Prvo, za relacijsku bazu, bilo je potrebno definirati tablice i osmisliti urednu logičku povezanost istih. Koristeći alate poput MySQL Server krenula je izrada baze te se ista odvijala brzo i s minimalnim problemima. Prije svega, izrađene su tablice s odgovarajućim atributima i tipovima podataka. Nakon toga, slijedilo je povezivanje tablica i njihovih relacija. Problem koji je uslijedio bio je nedostatak jedne od relacija odnosno relacija između tablica „posudba“ i „posudba_knjiga“. Ponovnom izradom problematične tablice odnosno tablice „posudba_knjiga“ problem je riješen. Iako su se pojavili problemi s relacijama između tablica, riješili smo ih brisanjem i ponovnim stvaranjem tablice "zaposlenik". Cijeli proces izrade relacijske baze trajao je dva dana. Izrada relacijske baze podataka „Knjižnica“ trajala je sveukupno dva dana. Nakon izrade ER dijagrama, dobiven je najbolji mogući prikaz baze podataka sa svim njezinim relacijama.

Nakon izrade relacijske baze bilo je potrebno izraditi NoSQL bazu podataka. Za potrebe ovog završnog rada, baza podataka izrađena je koristeći MongoDB. Prije svega, istraženi su potrebni programi i alati te instalirani na računalo. To su: MongoDB Community Server, MongoDB Compass i MongoDB Shell. Nakon besprijeorne instalacije i povezivanja na server započela je izrada baze. Uslijedili su izazovi, kako tijekom kreiranja NoSQL baze tako i tijekom postavljanja upita, obzirom na različiti upitni jezik u odnosu na strukturirani upitni jezik (SQL). Nakon kreiranja baze bilo je potrebno kreirati sve potrebne kolekcije te unijeti podatke u iste. Osim sintaksnih grešaka, koje su se javljale u slučaju nedostataka zagradi i dvotočaka, nije bilo ozbiljnijih problema. Cijeli proces izrade NoSQL baze trajao je jedan dan.

Unatoč manjim poteškoćama, izrada obje baze podataka završila je uspješno.

6.1 Prednosti i nedostaci SQL i NoSQL baza podataka

6.1.1 SQL

SQL, odnosno relacijske baze podataka, nude nekoliko prednosti. Prvo, ističe se njihova jednostavnost, kako u razvoju baze podataka tako i u korištenju iste. Relacijske baze podataka su poznate po svojoj intuitivnosti i jednostavnosti u razvoju. Definiranje tablica, veza između njih i unos podataka često su intuitivni procesi, što olakšava stvaranje baze. Drugo, prednost je strukturirani upitni jezik (SQL), koji olakšava filtriranje podataka i omogućava precizno dohvaćanje željenih podataka. Također, relacijske baze podataka su skalabilne, što znači kako je lako dodavati, uklanjati ili mijenjati podatke bez većih utjecaja na cijelu bazu. Osim toga, zbog

dobro definiranih standarda, relacijske baze podataka pružaju sigurnost i stabilnost. Konačno, unaprijed definirana shema olakšava upravljanje podacima.⁵⁰

Sve ove prednosti čine relacijske baze podataka popularnim izborom za mnoge aplikacije, osobito one gdje je potrebna precizna i sigurna obrada podataka, poput poslovnih sustava ili aplikacija za upravljanje inventarom.

S druge strane, unaprijed definirana shema, iako se često smatra prednošću zbog jednostavnijeg upravljanja podacima, može također predstavljati ozbiljan nedostatak tijekom izrade same baze. Primjerice, ukoliko odlučimo dodati novu tablicu, moramo mijenjati cjelokupnu strukturu baze kako bismo uskladili njene relacije. Osim toga, strogo definirana struktura relacijskih baza podataka, koja prati ERA model, zahtijeva pažljivo upravljanje tipovima, veličinama i vrstama podataka prilikom unošenja podataka u tablice. To se razlikuje od NoSQL sustava gdje se mogu unositi raznoliki podaci bez obzira na njihovu strukturu.⁵¹

6.1.2 NoSQL

NoSQL baze podataka nedvojbeno su naprednije od relacijskih baza podataka, s naglaskom na mogućnostima upravljanja velikim količinama podataka, bez obzira na njihovu strukturu, bilo da su strukturirani, polu-strukturirani ili ne-strukturirani.⁵² Zbog nedostatka strogo definirane sheme, NoSQL baze podataka nude veću fleksibilnost, što olakšava razvoj i uporabu, bolje performanse u odnosu na relacijske baze podataka, podršku za agilni pristup razvoju s naglaskom na agilne sprintove, brze iteracije te česte promjene kôda. Osim toga, NoSQL baze automatski stvaraju kolekcije (ekvivalent tablicama) kad se koriste, ako već ne postoje.⁵³

Uz sve navedene prednosti tu su i određeni nedostaci u usporedbi s relacijskim bazama podataka. NoSQL sustavi predstavljaju novonastalu, mladu tehnologiju koja svojom zrelošću i stabilnošću nije ni približna relacijskim sustavima za upravljanje bazama podataka. Također, česta pojava u vidu nedostatka kod NoSQL baza podataka bio bi nedostatak standardizacije iz razloga što velika raznolikost i raskošnost u dizajnu i među upitnim jezicima predstavljaju problem za programere iz razloga što se najčešće znanje rada s jednim NoSQL sustavom ne

⁵⁰ Usp. Schaefer, Lauren. MongoDB. URL: <https://www.mongodb.com/nosql-explained> (2023-08-30)

⁵¹ Usp. Hussain, Sadequl. (28.03.2019) SQL and NoSQL Databases Features and Differences. URL: <https://www.mssqltips.com/sqlservertip/5980/sql-and-nosql-database-features-and-differences/> (2023-08-30)

⁵² Usp. Sullivan, Dan. NoSQL for Mere Mortals. Addison-Wesley, 2015.

⁵³ Usp. Advantages Of NoSQL. URL: <https://www.mongodb.com/scale/advantages-of-nosq> (2023-08-10)

može primijeniti prilikom rada s nekim drugim.⁵⁴ Kompleksnost unosa podataka može biti izazovna, posebno pri pisanju specifičnih naredbi. Na primjer, u kolekciji "knjiga" s 6 atributa, uneseni podaci zahtijevaju više truda nego u relacijskim bazama gdje je često dovoljno koristiti identifikacijski broj (ID). NoSQL baze nemaju primarne ili vanjske ključeve kao što ih imaju relacijske baze podataka, što može otežati organizaciju podataka i referenciranje.

⁵⁴ Usp. Tozzi, Christopher. (31.05.2016) The Limitations of NoSQL Database Storage: Why NoSQL's Not Perfect. URL: <https://www.channelfutures.com/cloud-2/the-limitations-of-nosql-database-storage-why-nosqls-not-perfect> (2023-08-20)

7. Zaključak

Ovaj rad istražuje dva različita pristupa upravljanju podacima: relacijske baze podataka i tehnologije NoSQL. Istraživanje i analiza relacijskih i NoSQL baza podataka ukazuju na njihove različite karakteristike, prednosti i nedostatke. Relacijske baze podataka pružaju čvrstu strukturu, ACID transakcije i mogućnost složenih upita, što ih čini idealnim za transakcijske sustave i aplikacije gdje je konzistencija podataka ključna. S druge strane, NoSQL baze podataka nude skalabilnost, fleksibilnost i bolju sposobnost rada s nestrukturiranim podacima, što ih čini odgovarajućim za brze, distribuirane sustave i Big Data aplikacije.

Kroz istraživanje smo također uočili da ne postoji univerzalni odabir između ova dva tipa baza podataka, već ovisi o konkretnim zahtjevima i potrebama projekta. U današnjem tehnološkom okruženju, gdje se aplikacije često razvijaju agilno i brzo, hibridni pristup koji kombinira oba tipa baza podataka može biti najbolje rješenje za postizanje optimalne izvedbe i skalabilnosti. Ovo može značiti korištenje relacijskih baza za dijelove sustava koji zahtijevaju konzistenciju podataka, dok se NoSQL baze koriste za pohranu i analizu velikih količina podataka ili za podršku brzim operacijama. Osim toga, brzi razvoj tehnologije baza podataka i konstantne inovacije čine ovu usporedbu dinamičnom i zahtjevnom. Za buduće projekte važno je pratiti nove trendove i tehnološke promjene kako bi se osigurala najbolja moguća infrastruktura za pohranu i upravljanje podacima.

Nadalje, razvoj tehnologije baza podataka nikada ne stoji mirno. Nove inovacije i tehnologije poput NewSQL baza podataka, grafovskih baza, i različitih vrsta NoSQL sustava konstantno se pojavljuju na tržištu. Stoga je važno da organizacije ostanu otvorene prema promjenama i budu spremne prilagoditi svoje strategije baza podataka kako bi ostale konkurentske i iskoristile najnovije tehnološke prednosti.

Iako modeli baza podataka u ovom radu nisu bili dovoljno složeni da bi se pokazale detaljne razlike u performansama, bilo je moguće identificirati osnovne teorijske razlike u temeljnim funkcijama i karakteristikama tih dviju vrsta baza. Rad je istraživao način pristupa i osnovne principe rada s alatima za SQL i MongoDB, uključujući i razlike u sintaksi upita.

Naglasak je stavljen na razlike u strukturi podataka: relacijske baze koriste strogo definiranu shemu, dok NoSQL baze podataka omogućavaju fleksibilniji pristup podacima i podržavaju različite modele podataka. Relacijske baze široko su prihvaćene zbog svoje stroge strukture, što pruža veću sigurnost podataka.

NoSQL baze pokazale su se korisnima za rad s polu-strukturiranim i ne-strukturiranim podacima, što ih čini prikladnijima u određenim situacijama gdje se očekuje brz rast korisnika i podataka.

Zaključno, odabir između relacijskih i NoSQL baza podataka nije jednostavna odluka i nema univerzalno ispravnog odgovora. Relacijske baze često zadovoljavaju potrebe organizacija i aplikacija s manjim obimom podataka, dok će NoSQL baze biti prikladnije za projekte koji očekuju brzi rast i rukovanje velikim količinama nestrukturiranih podataka. Zahtijeva pažljivu analizu specifičnih potreba projekta i ciljeva, i razmatranju prednosti i nedostataka svake tehnologije uz kontinuirano praćenje razvoja tehnologije kako bi se održala konkurentna prednost i osigurala optimalna podrška aplikacijama.

Literatura

1. Advantages Of NoSQL. URL: <https://www.mongodb.com/scale/advantages-of-nosq> (2023-08-10)
2. Carić, Tonči; Buntić, Mario. Uvod u relacijske baze podataka. Zagreb: 2015., str. 2-16. URL: <http://files.fpz.hr/Djelatnici/tcaric/Tonci-Caric-Baze-podataka.pdf> (2020-08-13)
3. Harrington, Jan, L. Relational Database Design and Implementation, Burlington, MA, USA: Morgan Kauffman. 2009. URL: https://booksite.elsevier.com/samplechapters/9780123747303/01~Front_Matter.pdf (2023-07-11)
4. Hussain, Sadequl. SQL and NoSQL Databases Features and Differences. 28. ožujka 2019. URL: <https://www.mssqltips.com/sqlservertip/5980/sql-and-nosql-database-features-and-differences/> (2023-08-30)
5. IBM's 100 Icons of Progress. URL: <https://www.ibm.com/ibm/history/ibm100/us/en/icons/reldb/> (2023-07-19)
6. Livaja, Ivan. Uvod u baze podataka, Šibenik, 2016.
7. Maleković Mirko i Rabuzin Kornelije. Uvod u baze podataka. Varaždin: Fakultet organizacije i informatike, 2016.
8. Manger, Robert. Baze podataka, Zagreb: Element, 2011., str. 9-33. URL: <https://dokumen.tips/documents/robert-manger-baze-podatakapdf.html?page=1> (2023-07-11)
9. MongoDB Shell. URL: <https://www.mongodb.com/try/download/shell> (2023-09-02)
10. Moore, Lindsay. MySQL. URL: <https://searchoracle.techtarget.com/definition/MySQL> (2023-08-02)
11. MS SQL Server – Overview. URL: https://www.tutorialspoint.com/ms_sql_server/ms_sql_server_overview.htm (2023-08-02)
12. NoSQL (26.2.2020.). URL: <https://www.w3resource.com/mongodb/nosql.php> (2023-08-05)
13. Pickett, Patricia. What Is SQL?, 2020. URL: <https://www.thebalancecareers.com/what-is-sql-and-uses-2071909> (2023-08-20)
14. Rabuzin, Kornelije. Uvod u SQL. Varaždin: Fakultet organizacije i informatike, 2011.

15. Sadalage, Pramod J. & Fowler, Martin. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley Professional, 2012., str. 18-25. URL: <https://bigdata-ir.com/wp-content/uploads/2017/04/NoSQL-Distilled.pdf> (2023-08-03)
16. Schaefer, Lauren. MongoDB. URL: <https://www.mongodb.com/nosql-explained> (2023-08-30)
17. Silberschatz, Abraham, Korth, Henry, F. & Sudarshan, S. Database system concepts. URL: https://www.octawian.ro/fisiere/situri/asor/build/html/_downloads/1fcab53a6d916e39c715fc20a9a9c2a8/Silberschatz_A_databases_6th_ed.pdf
18. SQL Notes for Professionals. URL: <https://books.goalkicker.com/SQLBook/> (2023-08-20)
19. Stojanović, Aleksandar. Osvrt na NoSQL baze podataka - Četiri osnovne tehnologije, Polytechnic & Design, 2016. URL: <https://hrcak.srce.hr/clanak/283391> (2023-08-09)
20. Suehring, Steve. MySQL Bible, Wiley Publishing, Inc. URL: https://box.cs.istu.ru/public/docs/other/_New/Books/Data/DB/MySQL/MySQL%20Bible.pdf (2023-08-30)
21. Sullivan, Dan. NoSQL for Mere Mortals. Addison-Wesley, 2015.
22. Three different data structures. URL: <https://www.bigdataframework.org/data-types-structured-vs-unstructured-data/>
23. Tozzi, Christopher (31.05.2016) The Limitations of NoSQL Database Storage: Why NoSQL's Not Perfect. URL: <https://www.channelfutures.com/cloud-2/the-limitations-of-nosql-database-storage-why-nosqls-not-perfect> (2023-08-20)
24. What is a Graph Database? URL: <https://neo4j.com/developer/graph-database/>
25. What is a Key-Value Database? URL: <https://database.guide/what-is-a-key-value-database/>
26. What is AWS? URL: https://aws.amazon.com/what-is-aws/?nc1=f_cc (2023-09-01)
27. What is MySQL? URL: <https://www.atlantic.net/dedicated-server-hosting/what-is-mysql/> (2023-08-02)
28. What is SQLite? URL: <https://www.codecademy.com/article/what-is-sqlite> (2023-08-02)
29. What is SQL Server. URL: <https://www.sqlservertutorial.net/getting-started/what-is-sql-server/> (2023-08-02)