

Usporedba SQL i NoSQL baza podataka

Batur, Branimir

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Maritime Studies, Rijeka / Sveučilište u Rijeci, Pomorski fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:187:773580>

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

Download date / Datum preuzimanja: **2024-06-24**



Sveučilište u Rijeci, Pomorski fakultet
University of Rijeka, Faculty of Maritime Studies

Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Maritime Studies - FMSRI Repository](#)



SVEUČILIŠTE U RIJECI
POMORSKI FAKULTET

BRANIMIR BATUR

USPOREDBA SQL I NOSQL BAZA PODATAKA

DIPLOMSKI RAD

Rijeka, 2023.

SVEUČILIŠTE U RIJECI
POMORSKI FAKULTET

USPOREDBA SQL I NOSQL BAZA PODATAKA
COMPARISON OF SQL AND NOSQL DATABASES
DIPLOMSKI RAD

Kolegij: Analiza velikih podataka

Mentor: doc. dr. sc. Dario Ogrizović

Komentor: dr. sc. Ivan Lorencin

Student/studentica: Branimir Batur

Studijski smjer: Logistika i menadžment u pomorstvu i prometu

JMBAG: 0112076604

Rijeka, rujan 2023.

Student/studentica: Branimir Batur

Studijski program: Logistika i menadžment u pomorstvu i prometu

JMBAG: 0112076604

IZJAVA O SAMOSTALNOJ IZRADI ZAVRŠNOG RADA

Kojom izjavljujem da sam završni rad s naslovom

Usporedba SQL i NoSQL baza podataka

(naslov završnog rada)

izradio/la samostalno pod mentorstvom

(prof. dr. sc. / izv. prof. dr. sc. / doc dr. sc Ime i Prezime) doc. dr. sc. Dario Ogrizović

te komentorstvom dr. sc. Ivan Lorencin

stručnjaka/stručjakinje iz tvrtke

(naziv tvrtke).

U radu sam primijenio/la metodologiju izrade stručnog/znanstvenog rada i koristio/la literaturu koja je navedena na kraju završnog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući naveo/la u završnom radu na uobičajen, standardan način citirao/la sam i povezao/la s fusnotama i korištenim bibliografskim jedinicama, te nijedan dio rada ne krši bilo čija autorska prava. Rad je pisan u duhu hrvatskoga jezika.

Student/studentica

B Batur

(potpis)

Branimir Batur

Student/studentica: Branimir Batur

Studijski program: Logistika i menadžment u pomorstvu i prometu

JMBAG: 0112076604

IZJAVA STUDENTA – AUTORA
O JAVNOJ OBJAVI OBRANJENOG ZAVRŠNOG RADA

Izjavljujem da kao student – autor završnog rada dozvoljavam Pomorskom fakultetu Sveučilišta u Rijeci da ga trajno javno objavi i besplatno učini dostupnim javnosti u cjelovitom tekstu u mrežnom digitalnom repozitoriju Pomorskog fakulteta.

U svrhu podržavanja otvorenog pristupa završnim radovima trajno objavljenim u javno dostupnom digitalnom repozitoriju Pomorskog fakulteta, ovom izjavom dajem neisključivo imovinsko pravo iskorištavanja bez sadržajnog, vremenskog i prostornog ograničenja mog završnog rada kao autorskog djela pod uvjetima *Creative Commons* licencije CC BY Imenovanje, prema opisu dostupnom na <http://creativecommons.org/licenses/>

Student/studentica - autor

B Batur

(potpis)

SAŽETAK

Usporedba SQL i NoSQL baza podataka otkriva jedinstvene kvalitete, prednosti i nedostatke ovih tipova baze podataka. SQL baze podataka idealne su za aplikacije koje zahtijevaju integritet podataka i jasno definirane sheme jer su izvrsne u upravljanju strukturiranim podacima, kompliciranim odnosima i visokoj dosljednosti. NoSQL baze podataka najbolji su izbor za obradu velikih podataka, platforme društvenih medija te suvremene web i mobilne aplikacije jer pružaju fleksibilnost, skalabilnost i agilnost u upravljanju nestrukturiranim podacima. U ovome radu pokušati će se približiti na koji način odabrati najbolji alat prema zahtjevima aplikacije, prirodi podataka i željenoj razini skalabilnosti.

Ključne riječi: SQL, NoSQL, baze podataka, podatci.

SUMMARY

A comparison of SQL and NoSQL databases reveals the unique qualities, advantages, and disadvantages of these database types. SQL databases are ideal for applications that require data integrity and clearly defined schemas because they excel at managing structured data, complicated relationships, and high consistency. NoSQL databases are the best choice for big data processing, social media platforms, and modern web and mobile applications because they provide flexibility, scalability, and agility in managing unstructured data. This paper will try to approach how to choose the best tool according to the requirements of the application, the nature of the data and the desired level of scalability.

Keywords: SQL , NoSQL, databases, data.

SADRŽAJ

| | |
|--|-----------|
| SAŽETAK | I |
| SUMMARY | I |
| SADRŽAJ | II |
| 1. UVOD | 1 |
| 1.1. PROBLEM, PREDMET I OBJEKTI ISTRAŽIVANJA | 3 |
| 1.2. RADNA HIPOTEZA | 3 |
| 1.3. SVRHA I CILJEVI ISTRAŽIVANJA..... | 4 |
| 1.4. ZNANSTVENE METODE..... | 4 |
| 1.5. STRUKTURA RADA | 4 |
| 2. OPĆENITO O BAZAMA PODATAKA | 6 |
| 2.1. Relacijske baze podataka | 9 |
| 2.2. Objektno orijentirane baze podataka (OODB)..... | 10 |
| 2.3. Dokumentno orijentirane baze podataka..... | 13 |
| 2.4. Ključ – vrijednost baze podataka | 15 |
| 3. SQL BAZE PODATAKA | 17 |
| 3.1. MySQL | 19 |
| 3.2. Oracle | 21 |
| 3.3. Postgre SQL..... | 23 |
| 3.4. Microsoft SQL Server | 24 |

| | |
|--|-----------|
| 3.5. Primjena SQL sustava | 25 |
| 4. NoSQL BAZE PODATAKA | 28 |
| 4.1. MongoDB | 30 |
| 4.2. Apache Cassandra | 31 |
| 4.3. Redis | 32 |
| 4.4. Primjena NoSQL sustava | 35 |
| 5. USPOREDBA SQL I NOSQL BAZA PODATAKA | 37 |
| 5.1. Usporedba performansi | 37 |
| 5.2. Usporedba skalabilnosti | 39 |
| 5.3. Usporedba dosljednosti | 41 |
| 5.4. Usporedba modeliranja podataka | 44 |
| 5.5. Usporedba jednostavnosti korištenja | 46 |
| 5.6. Implementacija SQL i NoSQL baza podataka | 48 |
| 6. NOVI TRENDOMI U SUSTAVIMA ZA UPRAVLJANJE BAZAMA PODATAKA | 51 |
| 6.1. NewSQL | 52 |
| 6.2. Višemodelne baze podataka | 54 |
| 6.3. Baze podataka bez poslužitelja | 55 |
| 6.4. In-Memory baze podataka | 57 |
| 7. ZAKLJUČAK | 59 |
| LITERATURA | 61 |
| INTERNET IZVORI | 62 |
| POPIS TABLICA | 65 |
| POPIS SLIKA | 65 |
| POPIS GRAFIKONA | 65 |
| POPIS KRATICA | 66 |

1. UVOD

Odluka između SQL (Structured Query Language) i NoSQL (Not Only SQL) baza podataka u kontekstu sustava upravljanja bazom podataka postala je sve važnija za tvrtke koje traže učinkovita i skalabilna rješenja za pohranu i dohvaćanje podataka. Uobičajena metoda bila je korištenje SQL baza podataka jer one pružaju uspostavljenu i isprobanu osnovu za upravljanje i organiziranje strukturiranih podataka. S druge strane, uvođenje NoSQL baza podataka dovelo je do nove paradigme koja cijeni prilagodljivost, skalabilnost i agilnost pri upravljanju različitim nestrukturiranim podacima. Prilikom odabira najboljeg rješenja baze podataka za određeni slučaj upotrebe, važno je napraviti distinkciju razlika između ove dvije vrste baza podataka, kao i njihovih relativnih prednosti i nedostataka.

Podaci u SQL bazama podataka organizirani su u retke i stupce prema utvrđenim shemama u strukturiranom, tabličnom formatu. Koristeći transakcije, podržavaju načela ACID (Atomičnost, Dosljednost, Izolacija, Trajnost), jamčeći integritet podataka i čvrstu dosljednost. SQL baze podataka idealne su za aplikacije koje trebaju stroga jamstva dosljednosti, kao što su financijski sustavi, sustavi za planiranje resursa poduzeća (Enterprise resource planning) i konvencionalne transakcijske aplikacije, budući da dobro funkcioniraju u situacijama kada su odnosi podataka i komplicirani upiti ključni. Za dohvaćanje, ažuriranje i mijenjanje podataka, SQL baze podataka nude standardizirani upitni jezik (Structured Query Language), koji programerima omogućuje korištenje razvijenog sustava alata i okvira.¹

Nasuprot tome, NoSQL baze podataka usvajaju fleksibilniju metodologiju bez shema i uklanjaju nefleksibilnu strukturu SQL baza podataka. Moderne web i mobilne aplikacije, platforme društvenih mreža i obrada velikih podataka imaju koristi od sposobnosti NoSQL baza podataka da obrađuju nestrukturirane, polustrukturirane podatke i podatke koji se brzo mijenjaju. Strukturirani podaci - obično kategorizirani kao kvantitativni podaci - visoko su organizirani i lako ih je dešifrirati pomoću algoritama strojnog učenja. Razvio ga je IBM 1974., strukturirani upitni jezik (SQL) je programski jezik koji se koristi za upravljanje strukturiranim podacima.² Polustrukturirani podaci (npr. JSON, CSV, XML) su "most" između strukturiranih

¹ Garcia-Molina, H., Ullman, J. D., & Widom, J. Database Systems: The Complete Book. Prentice Hall, 2009.

² <https://www.ibm.com/blog/structured-vs-unstructured-data/> (11.08.2023)

i nestrukturiranih podataka.³ Nema unaprijed definirani podatkovni model i složeniji je od strukturiranih podataka, ali ga je lakše pohraniti od nestrukturiranih podataka. Polustrukturirani podaci koriste "metapodatke" (npr. oznake i semantičke oznake) za identifikaciju specifičnih karakteristika podataka i skaliranje podataka u zapise i unaprijed postavljena polja. Metapodaci u konačnici omogućuju bolje katalogiziranje, pretraživanje i analizu polustrukturiranih podataka od nestrukturiranih podataka.⁴ Korištenjem relacijske (SQL) baze podataka poslovni korisnici mogu brzo unositi, pretraživati i manipulirati strukturiranim podacima. NoSQL baze podataka koriste metode kao što su dijeljenje, replikacija i eventualna dosljednost za pružanje horizontalne skalabilnosti, distribuirane arhitekture i sposobnosti upravljanja golemim količinama podataka. Programeri mogu pohranjivati i dohvaćati podatke u različitim formatima, uključujući naprimjer parove ključ-vrijednost, strukture orijentirane na dokumente, modele temeljene na grafikonima i pohrane širokih stupaca, jer se ne pridržavaju ograničenja utvrđenih shema. Zbog mogućnosti mijenjanja shema podataka bez potrebe za ulaganjem u skupe migracije baze podataka, ova fleksibilnost omogućuje brz i agilan razvoj.

Postoji mnogo različitih čimbenika koje treba uzeti u obzir kada se uspoređuju SQL i NoSQL baze podataka, uključujući performanse, skalabilnost, dosljednost, modeliranje podataka, jednostavnost upotrebe i potrebe za određenim slučajevima upotrebe. Najbolja opcija oslanja se na posebne zahtjeve aplikacije i svojstva podataka koji se čuvaju, jer svaka vrsta baze podataka ima jedinstvene prednosti i nedostatke. Za mudar izbor potrebno je temeljito istraživanje i razumijevanje SQL i NoSQL baza podataka, njihovih značajki, specifičnosti implementacije i praktičnih slučajeva upotrebe. Ovaj diplomski rad će detaljno ispitati karakteristike, prednosti i nedostatke SQL i NoSQL baza podataka, uzimajući u obzir niz čimbenika. Kako bi se pružilo sveobuhvatno razumijevanje relativnih prednosti i nedostataka SQL i NoSQL baza podataka, ispitati će se referentne vrijednosti performansi, pristupe skalabilnosti, pitanja modeliranja podataka, značajke jednostavnosti korištenja i slučajeve korištenja u stvarnom svijetu.

³ <https://www.mongodb.com/nosql-explained> (19.08.2023)

⁴ <https://www.ibm.com/blog/structured-vs-unstructured-data/> (11.08.2023)

1.1. PROBLEM, PREDMET I OBJEKTI ISTRAŽIVANJA

Problem kojim se bavi ovaj rad je usporedba i procjena SQL i NoSQL baza podataka kako bi se što bolje razumjele njihove prednosti i mane u različitim scenarijima primjene. Uz sve veću složenost i raznolikost zahtjeva za podacima, brojne organizacije se suočavaju s izazovima kod odabiru odgovarajuće tehnologije baze podataka koja može učinkovito pohranjivati, upravljati i dohvaćati podatke. Nedostatak sveobuhvatnog i sustavnog istraživanja koje uspoređuje SQL i NoSQL baze podataka ometa procese donošenja odluka i može rezultirati neoptimalnim izborom baze podataka za specifične slučajeve upotrebe.

Cilj ovog istraživanja je provesti temeljitu usporedbu između SQL i NoSQL baza podataka, analizirajući njihove karakteristike, performanse, skalabilnost, dosljednost, pristupe modeliranju podataka i jednostavnost korištenja. Provođenjem sveobuhvatne studije, cilj je pružiti uvid kako bi se jasnije razumjele razlike između dvije baze te olakšalo donošenje informiranih odluka pri odabiru najprikladnijeg rješenja baze podataka na temelju njihovih specifičnih zahtjeva za aplikacijom.

Istraživanje će identificirati i definirati skup metrika i kriterija za usporedbu SQL i NoSQL baza podataka. Ove metrike mogu uključivati referentne vrijednosti performansi, mjere skalabilnosti, razine dosljednosti, fleksibilnost modeliranja podataka, faktore jednostavnosti upotrebe i slučajeve korištenja u stvarnom svijetu. Cilj je uspostaviti sveobuhvatan okvir za procjenu prednosti i slabosti svake vrste baze podataka na temelju objektivnih kriterija.

1.2. RADNA HIPOTEZA

Radna hipoteza ovog istraživanja temeljen je ne tome da bi izbor između SQL i NoSQL baza podataka trebao biti vođen specifičnim zahtjevima aplikacije, karakteristikama podataka i ciljevima skalabilnosti, pri čemu SQL baze podataka nude prednosti u upravljanju strukturiranim podacima i složenim odnosima, dok NoSQL baze podataka ističu u rukovanju nestrukturirani podaci, horizontalna skalabilnost i brza obrada podataka.

1.3. SVRHA I CILJEVI ISTRAŽIVANJA

Cilj ovog istraživanja je napraviti sveobuhvatnu usporedbu i procjenu SQL i NoSQL baza podataka, kako bi se najlakše ustvrdio odabir najprikladnijeg rješenja baze podataka za njihove specifične zahtjeve aplikacije, karakteristike podataka i ciljeve skalabilnosti.

1.4. ZNANSTVENE METODE

Kod obrade teme upotrebljavati će se sljedeće znanstvene metode : metoda indukcije i dedukcije kod iznošenja činjenica o SQL i NoSQL sustavima, metoda deskripcije, metoda komparacije, povijesna metoda te metoda klasifikacije.

1.5. STRUKTURA RADA

Diplomski rad govori o SQL i NoSQL bazama podataka. Svrha ovoga rada je na što bolji i kvalitetniji način usporediti dva sustava, njihove prednosti i mane te prikazati u kojem slučaju je najbolje koristiti iste.

Rad je podijeljen u 7 cjelina:

1. Uvod
2. Općenito o bazama podataka
3. SQL baze podataka
4. NoSQL baze podataka
5. Usporedba SQL I NoSQL baza podataka
6. Novi trendovi u sustavima za upravljanje bazama podataka
7. Zaključak

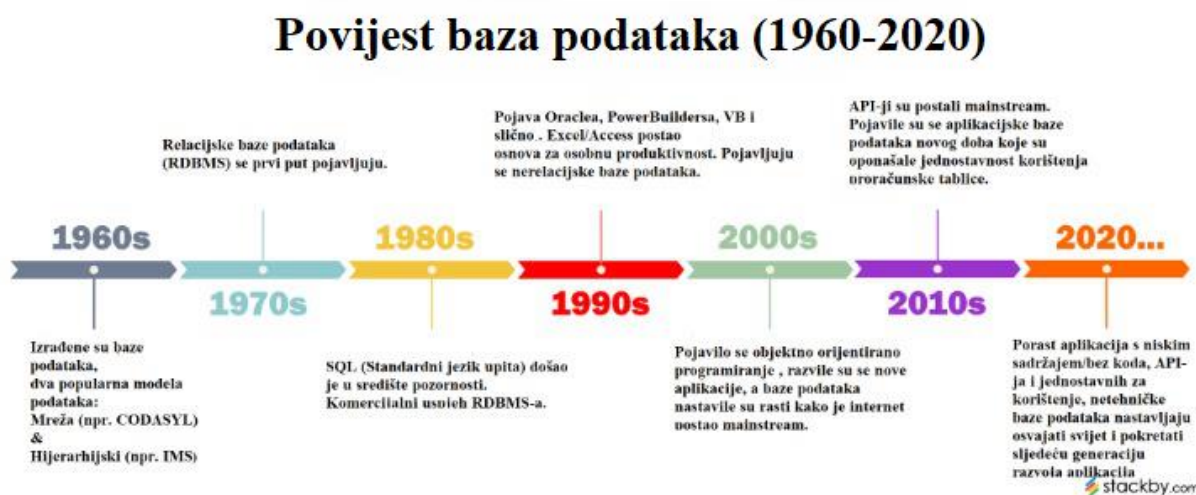
U uvodu opisani su radna hipoteza, ciljevi istraživanja, znanstveni postupci nakon čega su obrađeni i tema, problem, predmet i objekt istraživanja. U drugom poglavlju definirano je značenje baze podataka, kratka povijest te osnovni pojmovi vezani uz temu.

Treće poglavlje detaljnije opisuje SQL baze podataka, njene prednosti i mane te najpoznatiji sustavi koji se koriste. U četvrtom poglavlju obrađuju se karakteristike

NoSQL baze podataka te najkorišteniji sustavi, ukazuje se na prednosti i mane takvog sustava te gdje se najčešće koristi. Peto poglavlje donosi usporedbu dvaju sustava, prikazuju se primjeri implementacije u raznim tvrtkama, usporedba prednosti te njihove razlike i mogućnosti korištenja. U šestom poglavlju tema su novi trendovi u sustavima baza podataka, nove alternative klasičnim sustavima koje mogu predstavljati nova rješenja u budućnosti.

2. OPĆENITO O BAZAMA PODATAKA

„Baza podataka je organizirana zbirka strukturiranih informacija ili podataka, obično elektronički pohranjenih u računalnom sustavu. Bazom podataka obično upravlja sustav za upravljanje bazom podataka (Database Management Systems). Zajedno, podaci i DBMS, zajedno s aplikacijama koje su s njima povezane, nazivaju se sustavom baze podataka, često skraćeno na samo baza podataka.“⁵ Kada su se računala isprva počela koristiti za poslovne i znanstvene primjene u ranim danima računarstva, prvo su razvijene baze podataka. Prvi sustavi za upravljanje bazama podataka (DBMS) razvijeni su 1960-ih, omogućujući organizacijama učinkovitije pohranjivanje, organiziranje i upravljanje podacima. Za misiju Apollo, koja je dopremila ljude na Mjesec 1963., General Electric je stvorio prvi sustav za upravljanje bazom podataka (DBMS), poznat kao IDS.⁶ Relacijski model baza podataka izradio je IBM-ov informatičar Edgar Codd 1970-ih godina, a poslužio je kao osnova za sve naredne iteracije relacijskih baza podataka. Prema Coddovu modelu, podaci bi trebali biti raspoređeni u tablice od kojih svaka ima retke podataka i stupce koji odgovaraju svojstvima ili poljima podataka. Važan izum u administraciji baze podataka, relacijski model doveo je do SQL-a, koji je danas najpopularniji jezik za pristup i modificiranje podataka u relacijskim bazama podataka. Povijesni pregled razvitka baza podataka prikazan je shematskim prikazom na Slici 1.



Slika 1. Povijest razvitka baza podataka

Izvor: <https://stackby.com/blog/what-is-a-database/> (12.05.2023)

⁵ <https://www.oracle.com/database/what-is-database/> 12.05.2023

⁶ <https://www.ibm.com/ibm/history/ibm100/us/en/icons/ibmims/> 12.05.2023

Baze podataka napredovale su u snazi i sofisticiranosti tijekom 1980-ih i 1990-ih, a stvorene su i nove kategorije baza podataka, poput objektno orijentiranih baza podataka i objektno-relacijskih baza podataka. Ove nove vrste baza podataka bile su prikladnije za rad s kompliciranim strukturama podataka i golemim količinama nestrukturiranih podataka jer su bile prilagodljivije i skalabilnije od konvencionalnih relacijskih baza podataka. Baze podataka dodatno su dobile na važnosti kada se internet proširio tijekom 1990-ih budući da su ih pokretale mnoge od najranijih web stranica i platformi za e-trgovinu u svijetu.⁷ Nove vrste baza podataka, poput distribuiranih baza podataka i NoSQL baza podataka, stvorene su kao rezultat potražnje za bazama podataka koje bi mogle podnijeti goleme količine podataka i prometa. Danas postoje razne vrste baza podataka, svaka s različitim prednostima i nedostacima i izgrađena za upravljanje različitim vrstama radnih opterećenja i podataka. Najpopularniji primjeri su relacijske baze podataka, objektno orijentirane baze podataka, dokumentno orijentirane baze podataka te baze podataka bazirane na principu ključ-vrijednost.

Razvoj distribuiranih baza podataka potaknut je pojavom interneta i potražnjom za web aplikacijama. Distribuirane baze podataka nude visoku dostupnost i skalabilnost dok upravljaju podacima na različitim mjestima. Koristili su tehnike replikacije i particioniranja kako bi rasporedili radno opterećenje i podatke preko mreže međusobno povezanih sustava. Ideja NoSQL baza podataka pojavila se otprilike u isto vrijeme kako bi se riješili nedostaci konvencionalnih relacijskih baza podataka u obradi velikih količina nestrukturiranih i polustrukturiranih podataka. Replikacija je proces kopiranja podataka iz izvorne baze podataka u jednu ili više ciljnih baza podataka. Uključuje često kopiranje ili strujanje podataka iz baze podataka ili poslužitelja u drugu bazu podataka tako da svi korisnici imaju pristup sinkroniziranim podacima, neovisno o tome koji sustav koriste za pristup.⁸ Particioniranje baze podataka znači uzimanje različitih dijelova podataka pohranjenih u bazi podataka i njihovo odvajanje u različite particije ili dijelove. To se često radi kako bi se prilagodilo balansiranje opterećenja ili kako bi se pomoglo osigurati manje skupove baza podataka na kojima mogu raditi nezavisni poslužiteljski sustavi.⁹ Kako bi podržali fleksibilno modeliranje podataka i horizontalnu skalabilnost, NoSQL baze podataka ponudile su pristup bez shema. Bili su

⁷ <https://www.quickbase.com/articles/timeline-of-database-history> (20.08.2023)

⁸ <https://www.arcion.io/learn/database-replication> (11.08.2023.)

⁹ <https://www.techopedia.com/7/31992/storage/what-does-partitioning-mean-in-regards-to-a-database/> (11.08.2023.)

posebno prikladni za programe koji su trebali brzo unos podataka, neposrednu analitiku i brzu obradu podataka.

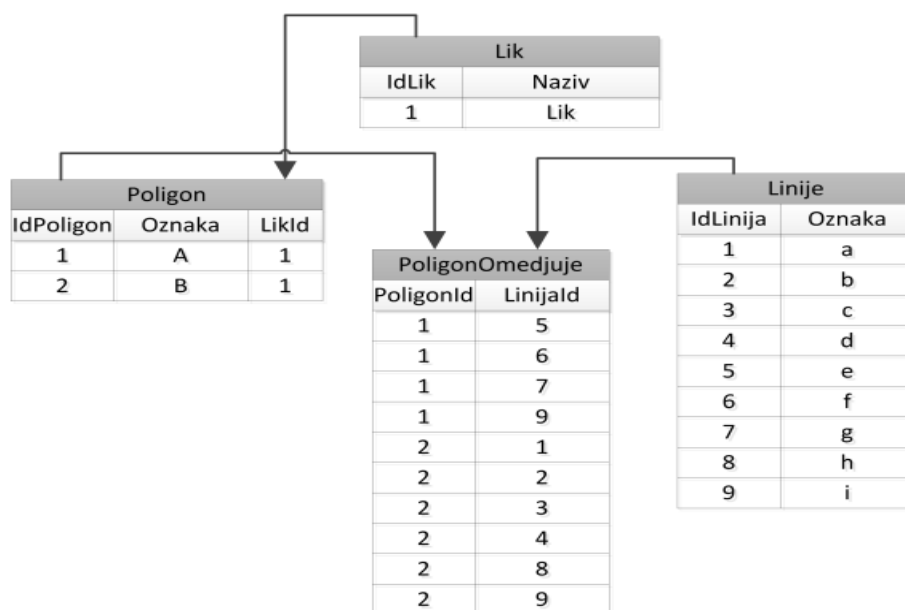
Korištenje baza podataka temeljenih na oblaku, gdje usluge baze podataka pružaju pružatelji usluga oblaka, u zadnjih par godina postalo je iznimno popularno. Uz skalabilnost, visoku dostupnost i trajnost podataka koje nude baze podataka u oblaku, tvrtke više nisu obavezne rukovati temeljnom infrastrukturom. Ova je paradigma postala sve popularnija zbog svoje pristupačnosti, skalabilnosti i jednostavnosti korištenja.

NewSQL baze podataka, čiji je cilj kombinirati skalabilnost NoSQL-a s transakcijskom dosljednošću konvencionalnih SQL baza podataka, također su rastući trendovi u sustavima za upravljanje bazama podataka. Za rukovanje suvremenim velikim transakcijskim opterećenjima, NewSQL baze podataka nude distribuirane arhitekture, horizontalnu skalabilnost i usklađenost s ACID-om (Atomicity, Consistency, Isolation, Durability).

Općenito, promjenjivi zahtjevi industrije, tehnički razvoj i pojava novih sektora primjene utjecali su na povijest baza podataka. Područje administracije baze podataka razvilo se od ranih dana strukturiranih relacijskih baza podataka do trenutnog stanja mnogih modela baza podataka i rješenja temeljenih na oblaku kako bi zadovoljila zahtjeve aplikacija s velikim brojem podataka i digitalnog sustava koji se neprestano širi.

2.1. Relacijske baze podataka

Najpopularnija vrsta baze podataka, relacijske baze podataka, prisutne su već neko vrijeme. Podaci se čuvaju u tablicama od kojih svaka ima retke podataka i stupce koji odražavaju različite aspekte ili polja podataka. Relacijske baze podataka koriste SQL za pristup i upravljanje podacima i temelje se na relacijskom modelu koji je prvi iznio Edgar Codd 1970. godine. Brojne prednosti relacijskih baza podataka uključuju integritet podataka, skalabilnost, fleksibilnost, sigurnost i dobre performanse. Na slici 2. prikazan je primjer relacijske baze podataka.



Slika 2. Primjer relacijske baze podataka

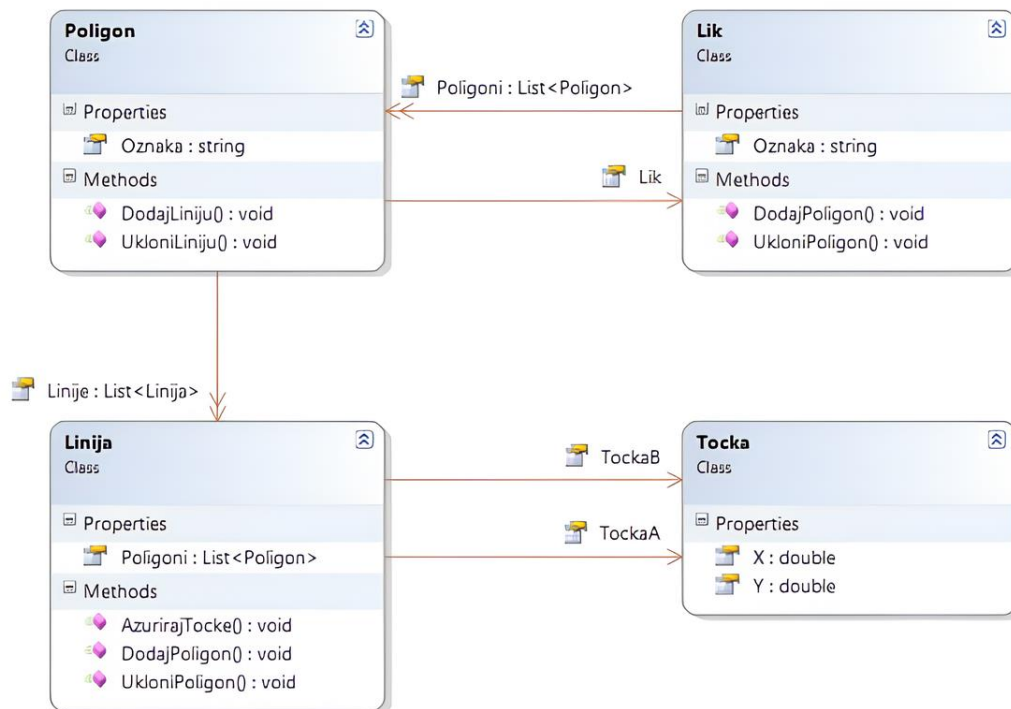
Izvor: Carić T., Buntić M., Uvod u relacijske baze podataka, Zagreb 2015.

Tablice relacijske baze podataka sastoje se od redaka i stupaca, gdje svaki redak odgovara određenoj instanci ili zapisu podataka, a svaki stupac jednom od svojih atributa. SQL je strukturirani jezik koji koristi relacijske baze podataka za dobivanje i upravljanje podacima. Budući da su učinkovite u rukovanju strukturiranim podacima i pružaju snažan integritet podataka i sigurnosne značajke, relacijske baze podataka bile su najpopularnija vrsta baza podataka dugi niz godina. Koriste se u raznim aplikacijama, uključujući e-trgovinu, bankarstvo i financije.¹⁰

¹⁰ <https://www.ibm.com/topics/relational-databases> (12.08.2023.)

2.2. Objektno orijentirane baze podataka (OODB)

Paradigma objektno orijentiranog programiranja služi kao temelj za objektno orijentirane baze podataka (OODB), posebnu vrstu baze podataka. Objektno orijentirane baze podataka pohranjuju podatke kao objekte, koji mogu sadržavati i podatke i ponašanje, za razliku od relacijskih baza podataka, koje raspoređuju podatke u tablice. Glavna prednost OODB-ova je da nude besprijekorno preslikavanje između baze podataka koja se koristi za pohranu podataka za aplikaciju i programskog jezika koji se koristi za njezino stvaranje. Budući da se objekti korišteni u aplikaciji mogu izravno preslikati na objekte pohranjene u bazi podataka, to može olakšati stvaranje aplikacija koje trebaju komplicirane strukture podataka. Slika 3. prikazuje primjer objektnog modela podataka.



Slika 3. Primjer objektnog modela podataka

Izvor: Carić T., Buntić M., Uvod u relacijske baze podataka, Zagreb 2015.

Osnovna svojstva bez kojih se model ne može proglasiti objektno orijentiranim su:

Apstrakcija - predstavlja pojednostavljivanje složenih objekata iz realnog svijeta na način da se izdvoje bitne karakteristike svakog objekta i njegovo ponašanje.

Enkapsulacija - predstavlja implementaciju koja će dovesti do željenog ponašanja objekta. Potrebno je odvojiti sučelje objekta od same implementacije njegovog ponašanja. Enkapsulacijom je moguće promijeniti implementaciju ponašanja objekta bez da se mijenja sučelje. Bitno je da drugi objekti mogu koristiti samo sučelje bez poznavanja implementacije ponašanja objekta.

Modularnost – modularnost ima za cilj poboljšati složenost organizacije složenih sustava, mogućnost održavanja i prilagodljivost. To podrazumijeva segmentiranje sustava u samostalne module kojima se lakše upravlja i koji mogu raditi neovisno i međusobno komunicirati putem jasno definiranih sučelja. Usredotočujući se na ponovno korištenje postojećih komponenti i izdvajanje određenih funkcionalnosti, ova strategija olakšava jednostavan razvoj, testiranje i održavanje. Ideja izgradnje modula koji obuhvaćaju objekte sa zajedničkim atributima i funkcionalnostima jedan je primjer modularnosti. Primjerice softverski program koji ima niz značajki povezanih s upravljanjem korisnicima, kao što su registracija korisnika, prijava i upravljanje profilom. Ove mogućnosti mogu se modulariti u neovisne, samostalne module umjesto da se kodiraju izravno u glavni program.

Nasljeđivanje - je svojstvo definiranja objekta na temelju objekata koji su već definirani. Novi objekt “nasljeđuje” sve metode i atribute već definiranog objekta te se može i ne mora dodati neke nove atribute ili izmijeniti ponašanje nasljeđenih metoda. Npr. objekt Učenik bi mogao naslijediti objekt Osoba te njegove atribute ime, prezime i datum rođenja. Bitno je zapamtiti da objekt koji nasljeđuje može izmijeniti ponašanje objekta koji nasljeđuje, ali ne i njegovu strukturu.

Polimorfizam - je svojstvo promjenjivosti oblika. Ovo se smatra najvažnijim svojstvom objektno orijentiranog koncepta gdje se objekt može predstavljati više od jednog tipa podatka.¹¹

ObjectStore, GemStone i Zope neki su primjeri OODB-ova. Zbog niza problema, kao što je nedostatak standardizacije i poteškoća integracije OODB-ova s drugim vrstama sustava, OODB-ovi se općenito ne koriste u industriji. Stvaranje okvira object-relational mapping (ORM), koji nude mehanizam za preslikavanje između objekata i relacijskih baza podataka, jedno je područje istraživanja predmeta OODB-ova. ORM okviri mogu olakšati izradu aplikacija koje trebaju komplicirane podatkovne strukture premošćivanjem problema između objektno orijentiranog programiranja i dizajna relacijske baze podataka.

¹¹ Carić T., Buntić M., Uvod u relacijske baze podataka, Zagreb 2015.

2.3. Dokumentno orijentirane baze podataka

Nestrukturirani ili djelomično strukturirani podaci pohranjuju se i njima se upravlja kao dokumentima pomoću baza podataka orijentiranih na dokumente, vrste NoSQL baze podataka. Za neke su aplikacije baze podataka orijentirane na dokumente fleksibilnije i skalabilnije budući da im nije potrebna unaprijed definirana shema, za razliku od relacijskih baza podataka, a svaki dokument može imati različitu strukturu. MongoDB je jedna od najomiljenijih baza podataka orijentiranih na dokumente. MongoDB je jednostavan za korištenje za programere koji su upoznati s JSON (JavaScript Object Notation), sintaksom jer koristi model dokumenta koji je sličan JSON-u za pohranu podataka. "JSON je format podataka čitljiv ljudima koji je postao prilično popularan posljednjih godina. Dok se mnogi različiti formati mogu koristiti za predstavljanje podataka unutar baze podataka dokumenata, kao što su XML ili YAML, JSON je jedan od najčešćih izbora. Na primjer, MongoDB je prihvatio JSON kao primarni format podataka za definiranje i upravljanje podacima."¹²

```
{
  "_id": "IveIvanov",
  "Ime": "Ive",
  "Prezime": "Ivanov",
  "Email": "ive.ivanov@digitalocean.com",
  "Odjel": "Financije"
}
```

Primjer 1.

Jednostavnije je opisati složene strukture podataka jer je svaki dokument samostalan i dopušta uključivanje već poznatih dokumenata i nizova. Za slučajeve kao što su naprimjer analitika društvenih mreža, sustavi za upravljanje sadržajem i aplikacije za e-trgovinu koje trebaju pohranjivati i postavljati upite ogromnim količinama nestrukturiranih ili polustrukturiranih podataka baze podataka orijentirane na dokumente posebno su prikladne.

Skalabilnost je jedna od glavnih prednosti baza podataka orijentiranih na dokumente. Svaki dokument može se distribuirati na nekoliko poslužitelja jer je samostalan, što omogućuje

¹² <https://www.digitalocean.com/community/conceptual-articles/an-introduction-to-document-oriented-databases> (19.08.2023)

horizontalno skaliranje kako se količina podataka povećava. Zbog toga su baze podataka orijentirane na dokumente mudar izbor za aplikacije koje moraju upravljati golemim količinama podataka ili se brzo skalirati. Fleksibilnost baza podataka orijentiranih na dokumente još je jedna prednost. Ne postoji postavljena struktura, tako da se u istoj bazi podataka može pohraniti niz različitih vrsta podataka. Ovo može biti korisno za aplikacije koje trebaju rukovati različitim vrstama podataka, kao što su multimedijски sadržaj ili podaci senzora. Međutim, korištenje baza podataka orijentiranih na dokumente također ima značajne nedostatke. Može biti izazovno izvršiti sofisticirana spajanja ili pretraživanja koja pokrivaju brojne dokumente jer je svaki od njih neovisan. Nadalje, budući da ne postoji postavljena shema, održavanje dosljednosti podataka između dokumenata može biti veći izazov.¹³ Baze podataka orijentirane na dokumente vrijedan su alat za mnoge programere unatoč ovim nedostacima i koriste se u raznim aplikacijama. Druge poznate baze podataka orijentirane na dokumente uključuju Couchbase, Apache CouchDB i Amazon DocumentDB uz MongoDB.

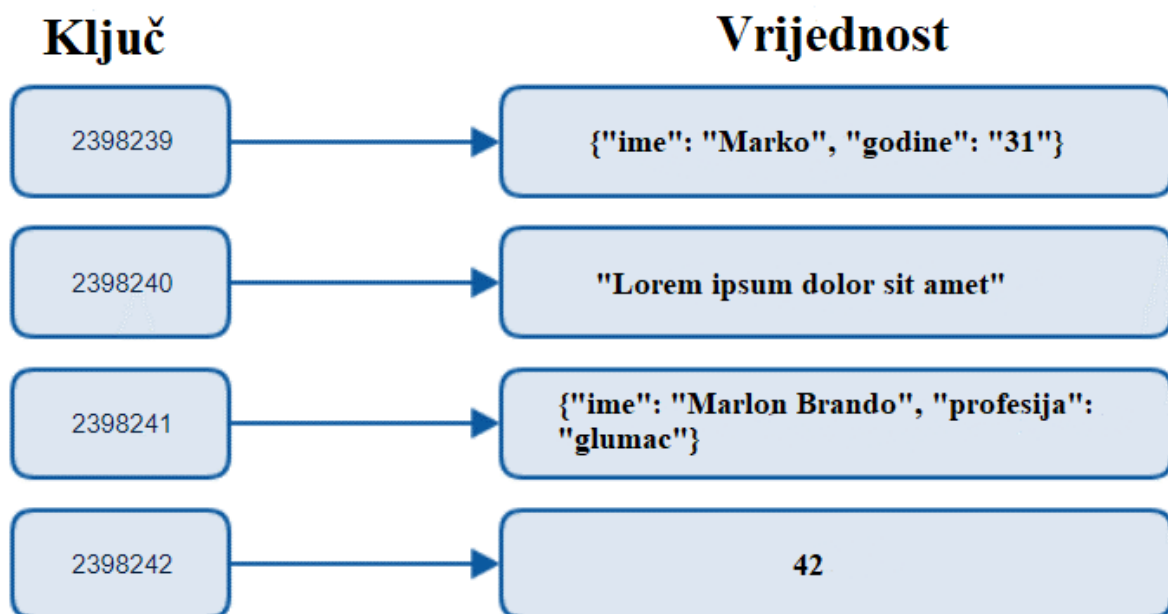
¹³ <https://www.digitalocean.com/community/conceptual-articles/an-introduction-to-document-oriented-databases> (19.08.2023)

2.4. Ključ – vrijednost baze podataka

Baza podataka ključ-vrijednost definirana je činjenicom da dopušta programima ili korisnicima programa dohvaćanje podataka pomoću ključeva, koji su u biti imena ili identifikatori koji upućuju na neku pohranjenu vrijednost. Budući da su baze podataka ključ-vrijednost definirane tako jednostavno, ali se mogu proširiti i optimizirati na brojne načine, ne postoji globalni popis značajki, ali postoji nekoliko uobičajenih:

- Dohvaćanje vrijednosti (ako postoji) pohranjene i povezane s danim ključem
- Brisanje vrijednosti (ako postoji) pohranjene i povezane s danim ključem
- Postavljanje, ažuriranje i zamjena vrijednosti (ako postoji) povezane s danim ključem¹⁴

Slika 4. prikazuje primjer baze podataka temeljene na principu ključ – vrijednost.



Slika 4. Primjer baze podataka temeljene na principu ključ – vrijednost

Izvor: <https://dev.to/aws-builders/intro-to-4-types-of-nosql-databases-45nh>

Predmemorija, pohrana sesije i druge aplikacije u kojima je brz pristup podacima ključan često koriste baze podataka ključ-vrijednost. Redis je jedna od najpopularnijih baza

¹⁴ <https://www.mongodb.com/databases/key-value-database> 13.05.2023

podataka ključ-vrijednost. Redis je besplatna pohrana strukture podataka u memoriji koja se može koristiti kao broker poruka, predmemorija ili baza podataka ključ-vrijednost. Uz izbor niza korisnih mogućnosti poput transakcija, Lua skriptiranja i pub/sub poruka, Redis nudi širok izbor struktura podataka, uključujući nizove, hashove, skupove i sortirane skupove. Hash je zbirka parova ključ-vrijednost. U Redisu, hashovi su mape između polja niza i vrijednosti niza. Dakle, koriste se za predstavljanje objekata. Niz je skup bajtova. U Redis bazi podataka, nizovi su binarno sigurni. To znači da imaju poznatu duljinu i nisu određeni posebnim završnim znakovima. Tako je moguće pohraniti do 512 megabajta u jednom nizu. Skupovi predstavljaju neuređenu zbirku nizova dok su složeni skupovi kolekcija nizova koji se ne ponavljaju. Svaki član sortiranog skupa pridružen je rezultatu koji se koristi da bi se poredao sortirani skup, od najmanjeg do najvećeg rezultata. Iako su članovi jedinstveni, rezultati se mogu ponavljati.¹⁵ Baze podataka ključ-vrijednost posebno su prikladne za slučajeve korištenja gdje je potreban brz, pouzdan pristup podacima. Čak i za vrlo velike skupove podataka, baze podataka ključ-vrijednost mogu ponuditi nevjerovatno brze operacije čitanja i pisanja jer se svaka kombinacija ključ-vrijednosti smatra jednim entitetom. Baze podataka ključ-vrijednost također mogu biti vrlo svestrane i prilagodljive različitim slučajevima upotrebe jer ne nameću postavljenu strukturu.

Međutim, korištenje baza podataka ključ-vrijednost ima i značajne nedostatke. Održavanje konzistentnosti podataka u više kombinacija ključeva i vrijednosti može biti izazovno jer one ne nameću postavljenu strukturu. Nadalje, može biti izazovno izvršiti sofisticirane upite koji obuhvaćaju brojne parove ključ-vrijednost jer ne podržavaju složene jezike upita. Unatoč ovim nedostacima, baze podataka ključ-vrijednost ostaju koristan alat za programere i koriste se u brojnim aplikacijama. Osim Redisa, popularne baze podataka ključ-vrijednost uključuju Apache Cassandra, Amazon DynamoDB i Riak.

¹⁵ <https://www.javatpoint.com/redis-data-types> (17.08.2023)

3. SQL BAZE PODATAKA

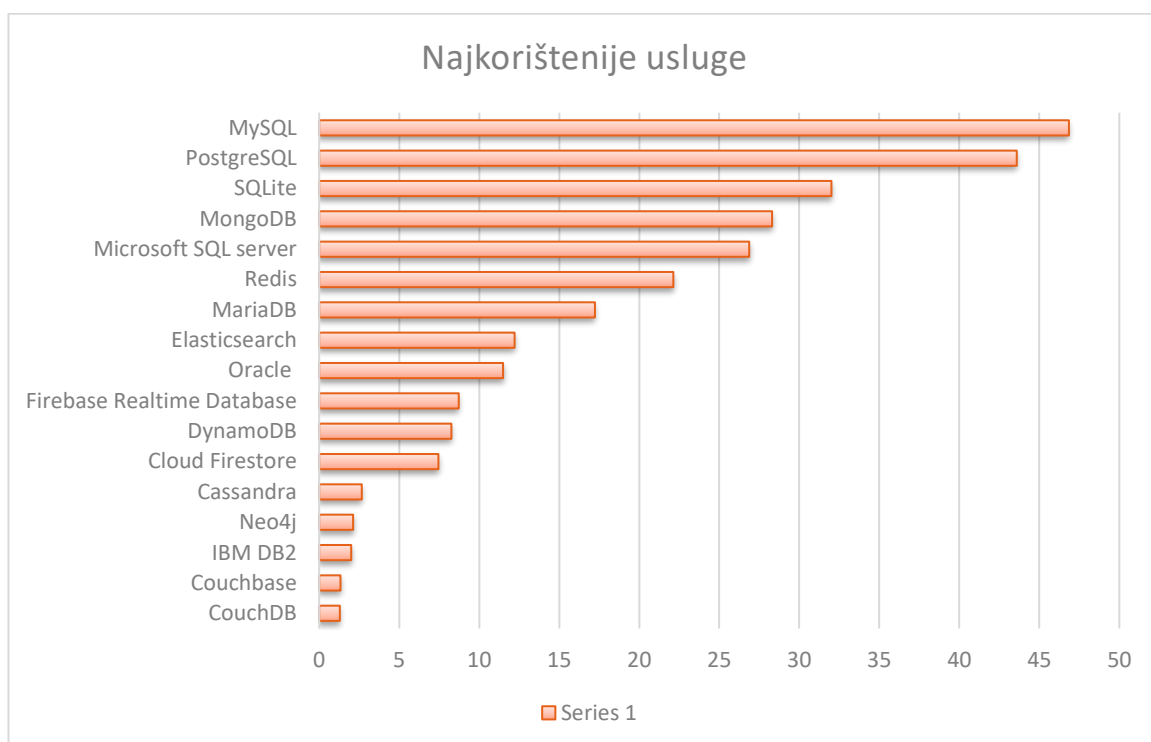
Structured Query Language (SQL) je standardizirani programski jezik koji se koristi za upravljanje relacijskim bazama podataka i izvođenje različitih operacija na podacima u njima.¹⁶ Porijeklo SQL-a seže još od tima IBM-ovih istraživača koji su ranih 1970-ih stvorili System R, sustav za pohranu i dohvaćanje podataka. Jedan od prvih sustava za upravljanje bazom podataka koji je koristio SQL kao upitni jezik bio je System R. SEQUEL (Structured English QUery Language), što je bio izvorni naziv SQL-a, modificiran je zbog zabrinutosti zbog zaštitnih znakova. Mnogi komercijalni sustavi za upravljanje relacijskim bazama podataka (RDBMS) stvoreni su u godinama koje su uslijedile, uključujući Oracle, IBM DB2 i Microsoft SQL Server. Interakcija relacijske baze podataka omogućena je uz pomoć SQL-a, koji se i danas naširoko koristi. Brzi rast popularnosti SQL-a tijekom 1980-ih i 1990-ih doveo je do njegovog usvajanja kao industrijskog standarda za RDBMS. S vremenom su u jezik uvedene dodatne sintakse i značajke kako se nastavio razvijati. Iako su NoSQL baze podataka postale popularnije u 2000-ima, SQL je i dalje ključna vještina za programere i stručnjake za podatke. Budući da je SQL deklarativni jezik, korisnici mogu definirati željene rezultate upita bez definiranja preciznih radnji koje bi sustav upravljanja bazom podataka trebao poduzeti da proizvede te rezultate. Nudi mogućnost umetanja, ažuriranja, brisanja i dohvaćanja podataka iz baze podataka pomoću SQL naredbi, među ostalim operacijama. Dodatno, SQL ima niz mogućnosti koje korisnicima olakšavaju izvođenje sofisticiranih podatkovnih operacija.

Jedna od najvažnijih karakteristika SQL-a je njegova učinkovitost u rukovanju velikim skupovima podataka. SQL je dizajniran za obradu ogromnih količina podataka i može učinkovito rukovati složenim upitima. Osim toga, napravljen je da funkcionira u višekorisničkim scenarijima, dopuštajući velikom broju ljudi da istovremeno pregledavaju i mijenjaju istu bazu podataka bez ugrožavanja integriteta podataka. Korisnici imaju kontrolu pristupa i visoku razinu sigurnosti zahvaljujući SQL-u. Omogućuje niz sigurnosnih značajki koje korisnicima omogućuju ograničavanje pristupa određenim aktivnostima baze podataka, reguliranje pristupa određenim podacima i odobravanje različitih razina pristupa različitim pojedincima.

Upravljanje cjelovitošću podataka još je jedna ključna funkcija SQL-a. Različite metode provjere valjanosti podataka koje nudi SQL omogućuju garanciju točnosti, dosljednosti

¹⁶ <https://www.techtarget.com/searchdatamanagement/definition/SQL> (23.08.2023)

i pouzdanosti podataka. Ovi alati uključuju okidače, jedinstvena ograničenja, ograničenja provjere, strane ključeve, primarne ključeve i strane ključeve. Osim toga, SQL korisnicima pruža raznovrsnost. Može se koristiti za različite poslove, od jednostavnog pronalaženja podataka do zamršene analize i manipulacije podacima. Budući da je SQL neovisan o platformi, može se koristiti s različitim operativnim sustavima i sustavima za upravljanje bazom podataka. Na slici 5. možemo vidjeti najkorištenije programe prema istraživanju stranice Stack Overflow koja prikazuje kako su MySQL i PostgreSQL najkorištenije usluge.

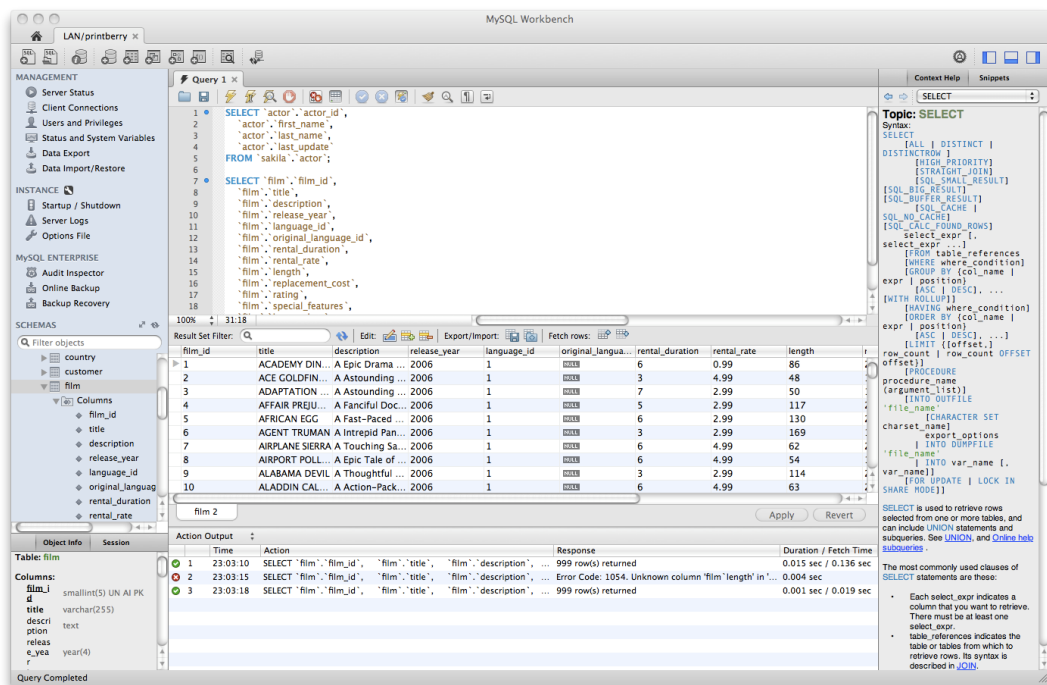


Grafikon 1. Najkorištenije usluge prema ispitanicima stranice Stack Overflow

Izvor: <https://survey.stackoverflow.co/2022/#most-popular-technologies-database> 14.05.2023.

3.1. MySQL

Sustav za upravljanje relacijskom bazom podataka otvorenog koda (open source) pod nazivom MySQL prvi je put postao dostupan 1995. Dobro je poznat po svojoj brzini, skalabilnosti i upotrebljivosti te se često koristi u web aplikacijama. Podržava širok raspon tipova podataka, uključujući nizove, datume i cijele brojeve, kao i širok raspon struktura podataka, uključujući tablice, prikaze i pohranjene procedure. Brzina MySQL-a jedna je od njegovih prednosti. Može učinkovito upravljati velikim brojem podataka i optimiziran je za brzu obradu podataka. Osim toga, MySQL podržava niz sofisticiranih mogućnosti koje omogućuju skalabilnost i visoku dostupnost, uključujući replikaciju, klasteriranje i particioniranje. Podaci se mogu automatski sinkronizirati na nekoliko poslužitelja replikacijom, što može poboljšati performanse i ponuditi redundantnost. Distribucija podataka preko brojnih poslužitelja omogućena je klasteriranjem i particioniranjem, što može povećati skalabilnost i otpornost na pogreške. Velika i aktivna zajednica programera i korisnika MySQL sustava olakšava dobivanje informacija i pomoći. Osim toga, različiti programski jezici i okviri kompatibilni su s MySQL-om, što ga čini jednostavnim za kombiniranje s drugim aplikacijama. Na slici 6. prikazan je „workbench“ popularnog programa MySQL.



Slika 5. Prikaz MySQL "Workbencha"

Izvor: <https://www.mysql.com/products/workbench/> (13.05.2023)

Podrška za niz uređaja za pohranu jedna je od temeljnih karakteristika MySQL-a. Mehanizam za pohranjivanje baze podataka odgovoran je za upravljanje pohranjivanjem i dohvaćanjem podataka. MySQL podržava višestruke mehanizme za pohranu, uključujući InnoDB, MyISAM i Memory. Ovisno o posebnim potrebama aplikacije, može se odabrati mehanizam za pohranu podataka. Činjenica da MySQL podržava transakcije i da je usklađen s ACID-om još je jedan ključni aspekt. Niz operacija baze podataka koje se nazivaju transakcije mogu se predati ili vratiti kao cjelina i njima se rukuje kao s jednom jedinicom rada. Karakteristike sustava baze podataka koje jamče integritet podataka, dosljednost i pouzdanost poznate su kao usklađenost s ACID-om. Za aplikacije koje trebaju transakcije i usklađenost s ACID-om, MySQL InnoDB mehanizam za pohranu je pouzdana opcija jer podržava ove karakteristike.¹⁷

Dodatno, MySQL nudi niz sigurnosnih značajki kao što su provjera autentičnosti korisnika, enkripcija i revizija. Dok enkripcija dodaje dodatnu razinu sigurnosti za osjetljive podatke, provjera autentičnosti korisnika osigurava da samo ovlaštene osobe mogu pristupiti bazi podataka. Administratori mogu nadzirati i pratiti aktivnosti baze podataka putem revizije, što je korisno zbog sigurnosti i usklađenosti. Sposobnost MySQL-a može se proširiti korištenjem raznih dodataka i proširenja uz njegove osnovne funkcije. To uključuje konektore za različite programske jezike i okvire, dodatke za pretraživanje cijelog teksta, dodatke za prostorne podatke i dodatke za skladištenje podataka.¹⁸

Svaki mehanizam za pohranu podataka ima vlastite prednosti i nedostatke. Kada je riječ o značajkama kao što su transakcije i usklađenost s ACID-, MySQL ipak ima određena ograničenja. U nekim okolnostima ova ograničenja mogu utjecati na točnost i cjelovitost podataka. Osim toga, za aplikacije na razini poduzeća kojima su potrebne sofisticiranije značajke, kao što su skladištenje podataka, rudarenje podataka i napredna analitika, MySQL možda nije uvijek idealna opcija.

¹⁷ <https://dev.mysql.com/doc/refman/8.0/en/storage-engines.html> (16.08.2023)

¹⁸ <https://dev.mysql.com/doc/refman/8.0/en/mysql-nutshell.html> (16.08.2023)

3.2. Oracle

Od svog predstavljanja 1979. godine, Oracle Corporation odgovorna je za razvoj i održavanje Oracle Database, dobro poznatog i često korištenog sustava za upravljanje SQL bazama podataka. Često se koristi u aplikacijama na razini poduzeća i poznat je po svojim izvrsnim performansama, skalabilnosti i robusnosti.¹⁹ Relacijski podaci, JSON, XML i geografski podaci samo su neki od mnogih vrsta podataka i formata koje podržava Oracle Database, koja je izgrađena za rukovanje ogromnim količinama podataka. Visoka dostupnost jedna je od prepoznatljivih karakteristika Oracle Database, a postiže se različitim metodama, uključujući klasteriranje, replikaciju, sigurnosno kopiranje i oporavak. Osim toga, nudi vrhunske sigurnosne mogućnosti za zaštitu podataka od neovlaštenog pristupa ili modifikacije, uključujući enkripciju, autentifikaciju i autorizaciju. Ova 3 pojma mogu se činiti slična, ali postoje razlike. Autentifikacija je proces potvrđivanja identiteta korisnika. Počinje u trenutku kada korisnik pokuša pristupiti informacijama. S druge strane, autorizacija samo odobrava pristup resursima nakon utvrđivanja mogućnosti pristupa sustavu i u kojoj mjeri je to dopušteno. Autorizacija je malo složenija od autentifikacije jer se može primijeniti na detaljnijoj razini. Identitet bi mogao biti uključen u grupu identiteta koji dijele zajedničku politiku autorizacije. Na kraju, tu je i enkripcija. Ako autentifikacija otvara vrata, a autorizacija određuje koja vrata smijete otvoriti, enkripcija se odnosi na zaštitu podataka. U svom najjednostavnijem obliku, enkripcija je proces kodiranja podataka kako bi bili kodirani i nerazumljivi. To bi podatke učinilo beskorisnim za svakoga tko im nije dopušten pristup.²⁰ Postoje različita izdanja Oracle Database, od kojih svako ima jedinstvenu kolekciju značajki i funkcionalnosti. Najpotpunije izdanje Oracle Database, Enterprise Edition dolazi sa svim svojim značajkama i funkcijama. Dok je Express Edition Oracle Database besplatna verzija koja je prikladna za testiranje ili male aplikacije, Standard Edition je skuplji izbor s manje funkcionalnosti.

Oracle Database uključuje niz dodatnih dodataka i alata uz svoje osnovne funkcije, kao što su geografska i graf analitika, pretraživanje teksta i dodatne sigurnosne mjere. Osim toga, podržava integraciju Oracle Application Servera, Oracle E-Business Suitea i Oracle Fusion Middlewarea, između ostalih Oracle softverskih proizvoda.

¹⁹ <https://www.techtarget.com/searchoracle/definition/Oracle> (13.05.2023)

²⁰ <https://www.secureauth.com/blog/authentication-vs-authorization-vs-encryption-whats-the-difference/> (16.08.2023)

Za rukovanje i pohranu ogromnih količina podataka u aplikacijama na razini poduzeća, Oracle Database snažan je i prilagodljiv sustav za upravljanje SQL bazom podataka koji nudi niz značajki i mogućnosti. Uvelike ga koriste i male i velike tvrtke u nizu sektora, a ima dobru reputaciju zbog pouzdanosti, sigurnosti i performansi.

3.3. Postgre SQL

Sustav za upravljanje relacijskom bazom podataka otvorenog koda (RDBMS) nazvan PostgreSQL poznat je po svojoj pouzdanosti, stabilnosti i pridržavanju SQL standarda. Kao sastavni dio projekta POSTGRES, prvobitno je nastao 1980-ih na Kalifornijskom sveučilištu Berkeley.²¹ Zbog svojih vrhunskih značajki i snažne podrške zajednice, PostgreSQL postaje sve češći izbor brojnih programera i organizacija.

Sposobnost PostgreSQL-a da upravlja masivnim i kompliciranim skupovima podataka jedna je od njegovih ključnih značajki. Prikladan je i za radna opterećenja mrežne obrade transakcija (online transactional processing) i mrežne analitičke obrade (online analytical processing) jer podržava širok raspon vrsta podataka i nudi brojne sofisticirane mogućnosti, uključujući optimizaciju upita, upravljanje transakcijama i indeksiranje. Kao rezultat toga, PostgreSQL se često koristi za aplikacije koje zahtijevaju visoku razinu dosljednosti i integriteta podataka. Mogućnost proširenja PostgreSQL-a još je jedan važan aspekt. PostgreSQL olakšava proširenje funkcionalnosti baze podataka kako bi se zadovoljili određeni poslovni zahtjevi dopuštajući korisnicima da konstruiraju vlastite jedinstvene tipove podataka, operatore i funkcije. Dodatno, PostgreSQL ima jake sigurnosne značajke kao što su enkripcija, kontrola pristupa temeljena na ulogama te provjera autentičnosti korisnika i dopuštenje. Stoga se može koristiti u aplikacijama koje rade s osjetljivim podacima. Može se zaključiti da je PostgreSQL je robustan i prilagodljiv sustav upravljanja relacijskom bazom podataka (RDBMS) s velikom i aktivnom zajednicom korisnika. Često je prvi izbor za mnoge velike aplikacije zahvaljujući svojoj izdržljivosti i pouzdanosti, uključujući one koje koriste tvrtke poput Applea, Fujitsua i Cisca.

²¹ <https://www.postgresql.org/docs/current/history.html> 14.05.2023

3.4. Microsoft SQL Server

Dobro poznati i naširoko korišteni sustav za upravljanje relacijskim bazama podataka, Microsoft SQL Server, kreirao je Microsoft. Poznat po svojoj pouzdanosti, skalabilnosti i opsežnom skupu značajki, SQL Server je stekao ime kao vodeća opcija za tvrtke svih veličina. Korisnici imaju pristup širokom rasponu mogućnosti sa SQL Serverom koje se mogu koristiti za sve, od malih aplikacija do velikih poslovnih instalacija. Snažan mehanizam za optimizaciju i izvođenje upita koji se može pohvaliti sofisticiranim optimizatorom upita koji pomno analizira upite i proizvodi učinkovite planove za izvođenje nalazi se u srcu Microsoft SQL Servera. Ova mogućnost optimizacije osigurava optimalne performanse za poslove koji uključuju obradu upita. SQL, koji je u potpunosti podržan od strane SQL Servera, pruža standardiziran i ekspresivan način za interakciju korisnika s bazom podataka i jednostavno dobivanje, manipuliranje i analizu podataka.

Kako bi se poboljšala pohrana i dohvaćanje podataka, SQL Server korisnicima daje slobodu izbora između raznih opcija indeksiranja i mehanizama za pohranu. SQL Server Database Engine, koji je njegov zadani mehanizam za pohranu podataka, nudi transakcijsku podršku, zaštitu integriteta podataka i pouzdane mehanizme oporavka. Dodatno, SQL Server nudi razne metode indeksiranja, kao što su klasterirani i neklasterirani indeksi, koji su dizajnirani za poboljšanje izvedbe upita ubrzavanjem postupaka dohvaćanja podataka. Značajke Extract, Transform, and Load (ETL) SQL Server Integration Services (SSIS) omogućuju Microsoft SQL Serveru da radi iznimno dobro u scenarijima integracije podataka. Korisnici mogu lako izvući podatke iz različitih izvora, izvršiti transformacije podataka i učitati podatke u baze podataka SQL Servera zahvaljujući SSIS-u. Izgradnja postupaka integracije podataka, sustava skladištenja podataka i aplikacija poslovne inteligencije imaju velike koristi od ove funkcionalnosti.²²

U SQL Serveru sigurnost se još uvijek pridaje vrhunskoj važnosti, a podaci u prijenosu i mirovanju zaštićeni su širokim rasponom sigurnosnih značajki. Sustav uključuje upravljanje korisnicima i ulogama za finu kontrolu pristupa, procese provjere autentičnosti i autorizacije te transparentnu enkripciju podataka za jačanje zaštite podataka. Kako bi se osigurala usklađenost sa zakonskim obvezama, SQL Server također uključuje funkcije revizije i usklađenosti.

²² : Kellyn Gorman, Allan Hirt, Dave Noderer, James Rowland-Jones, Arun Sirpal, Dustin Ryan, and Buck Woody: *Introducing Microsoft SQL Server 2019*, 2019.

Korisnici mogu imati koristi od potpuno upravljanog rješenja baze podataka temeljenog na oblaku s Azure SQL bazom podataka. Skalabilnost, visoka dostupnost, automatizirano sigurnosno kopiranje i interakcija s drugim Azure uslugama značajke su Azure SQL baze podataka. To omogućuje tvrtkama da se koncentriraju na razvoj aplikacija dok upravljanje bazom podataka ostavljaju u rukama pružatelja usluga u oblaku.

Microsoft SQL Server pruža široku paletu alata, razvojnih okvira i konektora trećih strana podržanih robusnim sustavom. Sa SQL Serverom kao pozadinskom bazom podataka, sustav se besprijekorno povezuje s različitim programskim jezicima, uključujući .NET, Java, Python, između ostalih.

3.5. Primjena SQL sustava

Moderni sustavi za upravljanje podacima izgrađeni su oko SQL baza podataka, koje pružaju pouzdan i učinkovit način pohranjivanja, organiziranja i dohvaćanja strukturiranih podataka. Budući da su pouzdane, imaju definiran jezik upita i imaju razvijen sustav, SQL baze podataka naširoko se koriste u raznim sektorima i aplikacijama. Budući da mogu jamčiti integritet podataka, prihvatiti komplicirane upite i ponuditi transakcijske mogućnosti, SQL baze podataka su se etablirale kao ključan alat mnogih poduzeća.

Poduzetničke aplikacije - Upravljanje odnosima s kupcima (CRM), planiranje resursa poduzeća (ERP), upravljanje ljudskim resursima (HRMS) i rješenja za upravljanje lancem opskrbe oslanjaju se na SQL baze podataka kao svoj temelj. SQL baze podataka najbolja su opcija za ove aplikacije jer podržavaju organiziranu pohranu podataka, transakcijski integritet i brzu obradu upita.²³

E-trgovina i maloprodaja - Za rukovanje golemim brojem informacija o proizvodu, podataka o kupcima i evidencije o transakcijama, platforme za online kupnju, sustavi za upravljanje zalihama i sustavi za obradu narudžbi značajno se oslanjaju na SQL baze podataka. Za preciznu

²³ <https://www.oracle.com/erp/erp-vs-crm/> (19.08.2023)

i pouzdanu pohranu i dohvaćanje podataka, SQL baze podataka nude potrebne značajke ACID.²⁴

Financijske usluge - Financijske aplikacije kao što su bankarski sustavi, platforme za trgovanje, obrada plaćanja i upravljanje rizikom često koriste SQL baze podataka. Budući da ti sustavi trebaju točno i dosljedno rukovanje podacima za transakcije, stanja računa, revizijske tragove i zahtjeve usklađenosti, SQL baze podataka su pouzdana opcija.²⁵

Zdravstveni sustav - SQL baze podataka neophodne su za pohranjivanje i upravljanje medicinskih podataka, zapisima pacijenata, povijesti bolesti, dijagnostičkim izvješćima i zakazivanjem termina u elektroničkim medicinskim zapisima (electronic medical records), sustavima za upravljanje pacijentima i zdravstvenim informacijskim sustavima. Budući da su SQL baze podataka strukturirane, moguće je brzo dobiti podatke za korištenje u kliničkim istraživanjima i donošenju odluka.²⁶

Vlada i javni sektor - Aplikacije vladinog i javnog sektora, uključujući baze podataka građana, sustave porezne uprave, registraciju imovine, sustave javne sigurnosti i sustave upravljanja dokumentima, sve se uvelike oslanjaju na SQL baze podataka. Ove bitne aplikacije mogu imati koristi od integriteta podataka, sigurnosti i mogućnosti standardiziranih upita koje pružaju SQL baze podataka.²⁷

Web aplikacije - Razvoj web aplikacija i sustavi za upravljanje sadržajem i značajno koriste SQL baze podataka. Korisnički profili, informacije o sesijama, metapodaci o sadržaju i sadržaj koji su generirali korisnici mogu se u njima pohraniti pouzdano i učinkovito. Složene upite također podržavaju SQL baze podataka, omogućujući sofisticiraniju analizu podataka i mogućnosti pretraživanja.²⁸

Skladištenje podataka i poslovna inteligencija - Aplikacije za skladištenje podataka i poslovnu analitiku najčešće koriste SQL baze podataka. Za obradu ogromnih količina

²⁴ <https://anywhere.epam.com/business/big-data-analytics-in-retail> (19.08.2023)

²⁵ Connolly, T. M., & Begg, C. E. (2014). Database Systems: A Practical Approach to Design, Implementation, and Management

²⁶ <https://www.sciencedirect.com/science/article/pii/S1532046420302987> (19.08.2023)

²⁷ <https://www.oecd-ilibrary.org/sites/1ab27217-en/index.html?itemId=/content/component/1ab27217-en> (19.08.2023)

²⁸ <https://www.ramotion.com/blog/database-in-web-app-development/> (19.08.2023)

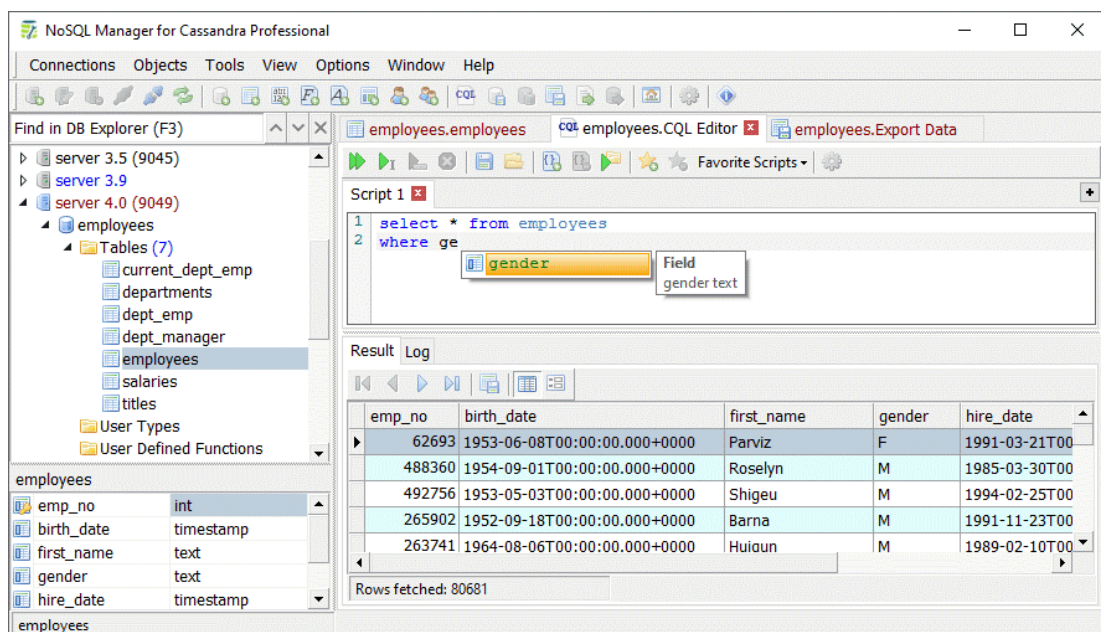
strukturiranih podataka, izradu izvješća i stjecanje uvida, oni uključuju snažne mogućnosti modeliranja podataka, ad-hoc upite i snažne analitičke alate.²⁹

Ovo je samo nekoliko primjera koliko su široko korištene SQL baze podataka. Prikladne su za različite primjene u raznim industrijama gdje se koriste za upravljanje strukturiranim podacima, integritet podataka i složene upite zahvaljujući njihovoj prilagodljivosti, skalabilnosti i zrelosti.

²⁹ Connolly, T. M., & Begg, C. E. (2014). Database Systems: A Practical Approach to Design, Implementation, and Management

4. NoSQL BAZE PODATAKA

NoSQL baze podataka postale su popularna zamjena za konvencionalne SQL baze podataka tijekom posljednjih deset godina. NoSQL baze podataka, koje koriste nerelacijske modele podataka, stvorene su kako bi zadovoljile zahtjeve suvremene obrade i pohrane podataka. NoSQL baze podataka prvotno su stvorene kako bi se prevladali nedostaci SQL baza podataka pri rukovanju ogromnim količinama podataka, tokovima podataka koji se brzo kreću i visoko raspršenim arhitekturama podataka. Ta ograničenja uključuju nedostatak fleksibilnosti u dizajnu sheme, poteškoće s horizontalnim širenjem i ograničenu podršku za nestrukturirane i polustrukturirane podatke. Nekoliko modela podataka, uključujući one orijentirane na dokumente, modele “key-value”, “column family” i grafikone, postoje u NoSQL bazama podataka za razliku od SQL baza podataka. Ovi su modeli dizajnirani za rukovanje različitim vrstama podataka i scenarijima korištenja. Osim toga, NoSQL baze podataka moraju uzeti u obzir druge čimbenike dizajna osim SQL baza podataka, kao što su konzistencija, visoka dostupnost i distribuirano postavljanje upita. Slika 7. daje uvid u sučelje programa Cassandra.



Slika 6. Prikaz NoSQL baze podataka “Cassandra”

Izvor: <https://www.mongoddbmanager.com/cassandra/screenshots/> (25.06.2023.)

Kao što implicira naziv, baze podataka ključ-vrijednost pohranjuju podatke kao parove ključ-vrijednost. Korisni su za slučajeve upotrebe kao što su predmemorija i upravljanje sesijama jer su optimizirani za jednostavan, brz i skalabilan pristup podacima. NoSQL baze podataka porasle su u popularnosti posljednjih godina kao rezultat njihove svestranosti u obradi različitih vrsta struktura podataka, sposobnosti upravljanja velikim količinama podataka i sposobnosti horizontalnog skaliranja. Dobro poznate NoSQL baze podataka MongoDB, Cassandra, Redis i Neo4j samo su neki od primjera.

Redis, Riak i Amazon DynamoDB nekoliko su dobro poznatih baza podataka ključ-vrijednost. Velike količine strukturiranih podataka mogu se pohraniti te upravljati u bazama podataka „column family“ zahvaljujući njihovoj fleksibilnoj shemi. Oni kategoriziraju podatke u „column family“, koje su zbirke povezanih stupaca. Slučajevi korištenja analitike i skladištenja podataka dobro odgovaraju bazama podataka obitelji stupaca. Amazon SimpleDB, HBase i Apache Cassandra nekoliko su primjera baza podataka obitelji stupaca.³⁰

³⁰ <https://cloudacademy.com/blog/amazon-dynamodb-ten-things/> (16.08.2023)

4.1. MongoDB

MongoDB je moćna, fleksibilna i skalabilna baza podataka opće namjene. Kombinira se mogućnost skaliranja sa značajkama kao što su sekundarni indeksi, upitni raspon, sortiranje, agregacije i geoprostorni indeksi.³¹ Sa svojom velikom skalabilnošću, fleksibilnošću i performansama za upravljanje nestrukturiranim i polustrukturiranim podacima, MongoDB je dobro poznata NoSQL baza podataka orijentirana na dokumente. Stvorio ga je MongoDB Inc. i prvi put je postao dostupan 2009. Moderne web i mobilne aplikacije imaju koristi od performansi koje nudi MongoDB-a – pohranjivanje i upravljanje ogromnim količinama podataka na distribuiran način. Prilagodljivi podatkovni model MongoDB-a jedna je od njegovih istaknutih kvaliteta. MongoDB pohranjuje podatke kao fleksibilne, samostalne dokumente koristeći BSON format sličan JSON-u, za razliku od konvencionalnih relacijskih baza podataka. Također omogućuje razvojnim programerima pohranjivanje podataka i postavljanje upita bez potrebe za određenim shemama.

Skalabilnost je još jedan važan aspekt MongoDB-a. Koristi se distribuiranim dizajnom koji omogućuje horizontalno skaliranje na više poslužitelja ili klastera, što rezultira velikom dostupnošću i performansama. Distribucija i replikacija podataka pojednostavljena je MongoDB-ovim značajkama dijeljenja i replikacije, koje također pružaju balansiranje opterećenja, toleranciju grešaka i pouzdane podatke. Opsežan sustav i integracija alata dostupnih za MongoDB također poboljšava njegovu funkcionalnost. aMongoDB je dobra opcija za stvaranje vrhunskih, skalabilnih aplikacija jer se također integrira s dobro poznatim okvirima kao što su Node.js i Spring.

Uzimajući sve u obzir, MongoDB pruža skalabilno i prilagodljivo rješenje za upravljanje nestrukturiranim i polustrukturiranim podacima. Prikladan je za različite primjene, uključujući sustave za upravljanje sadržajem, analitiku u stvarnom vremenu i mobilne aplikacije, zbog svog pristupa usmjerenog na dokumente, distribuiranog dizajna i robusnih mogućnosti postavljanja upita.

³¹ Chodorow K., MongoDB: The Definitive Guide, 2013.

4.2. Apache Cassandra

NoSQL baza podataka s velikom skalabilnošću i distribucijom, Apache Cassandra prvotno je kreirao Facebook, a kasnije je postao dostupan kao projekt otvorenog koda. Visoka dostupnost, otpornost na pogreške i linearna skalabilnost značajke su koje Cassandra nudi dok rukuje ogromnim količinama podataka na nekoliko robnih poslužitelja. Distribuirana arhitektura Cassandre jedna je od njezinih primarnih karakteristika. „Distribuirani sustav skup je računalnih programa koji koriste računalne resurse preko višestrukih, odvojenih računalnih čvorova za postizanje zajedničkog zajedničkog cilja. Distribuirani sustavi imaju za cilj ukloniti uska grla ili središnje točke kvara iz sustava.“³² Podaci se šire među brojnim čvorovima klastera koristeći peer-to-peer pristup distribuiranog sustava. Zahvaljujući ovoj decentraliziranoj metodi, Cassandra može obraditi ogromne količine podataka i visoku propusnost čitanja i pisanja uz nisku latenciju. Osim toga, nudi toleranciju na greške i automatsku replikaciju podataka, jamčeći postojanost podataka čak i u slučaju kvarova čvorova. „Dok u mnogočemu Cassandra sličići bazi podataka i s njom dijeli mnoge strategije dizajna i implementacije, Cassandra ne podržava puni relacijski model podataka; umjesto toga, klijentima pruža jednostavan podatkovni model koji podržava dinamičku kontrolu nad rasporedom i formatom podataka.“³³

Cassandra je poznata po svojoj sposobnosti horizontalnog skaliranja. „Horizontalno skaliranje odnosi se na dodavanje dodatnih čvorova ili strojeva infrastrukturi kako bi se mogli nositi s novim zahtjevima. Ako se aplikacija uslužuje na poslužitelju i ustanovi se da više nema kapacitet ili mogućnosti za rukovanje prometom, dodavanje poslužitelja može biti rješenje.“³⁴ Dodatni čvorovi mogu se dodati u klaster bez zastoja ili smetnji s povećanjem opterećenja podataka. Zbog toga može podnijeti velika radna opterećenja podataka i velik promet i širenje podataka. Još jedna značajna značajka Cassandre je njezin podatkovni model. Slijedi podatkovni model stupaca gdje su podaci organizirani u tablice sastavljene od redaka i stupaca. Podržava fleksibilan dizajn sheme, dopuštajući svakom redu da ima različite skupove stupaca. Ova fleksibilnost omogućuje brz razvoj i jednostavnu prilagodbu promjenjivim zahtjevima podataka.

³² <https://www.atlassian.com/microservices/microservices-architecture/distributed-architecture/> (15.08.2023)

³³ Lakshman, A., Malik, P., Cassandra: A decentralized structured storage system. ACM SIGOPS Operating Systems Review, 2009.

³⁴ <https://www.cloudzero.com/blog/horizontal-vs-vertical-scaling> (17.08.2023)

Cassandra također nudi podesive razine konzistentnosti, omogućujući razvojnim programerima da uravnoteže konzistentnost i dostupnost podataka na temelju zahtjeva svojih aplikacija. Podržava konačnu dosljednost i pruža podesive postavke dosljednosti za kontrolu replikacije podataka i operacija čitanja i pisanja u cijelom klasteru. Nadalje, Cassandra nudi pouzdanu podršku za replikaciju i toleranciju na greške. Podaci se mogu kopirati između podatkovnih centara i zemljopisnih lokacija zahvaljujući nizu algoritama replikacije, koji uključuje osnovne i one koji se temelje na topologiji mreže. To jamči dostupnost podataka i otpornost na kvarove.

Kompresija podataka, predmemorija i sažimanje također su izvorno podržani od strane Cassandre, poboljšavajući učinkovitost pohrane i performanse upita. Prikladan je za razne scenarije upotrebe, uključujući analitiku u stvarnom vremenu, podatke vremenskih serija i unos podataka velikom brzinom, zahvaljujući značajkama kao što su sekundarni indeksi, skeniranje raspona i podrška za komplicirane vrste podataka. Uzevši sve u obzir, Cassandra je snažna i skalabilna NoSQL baza podataka koja se ističe u rukovanju velikim, raspršenim i brzim radnim opterećenjima podataka. Aplikacije koje zahtijevaju visoku dostupnost, skalabilnost i izvedbu dobro su mu prilagođene zahvaljujući svojoj distribuiranoj arhitekturi, toleranciji na greške i fleksibilnom modelu podataka.

4.3. Redis

„Redis je pohrana strukture podataka otvorenog koda (licenciran za BSD) u memoriji koja se koristi kao baza podataka, predmemorija, broker poruka i mehanizam za strujanje. Redis pruža strukture podataka kao što su nizovi, hashovi, popisi, skupovi, sortirani skupovi s upitima raspona, bitmape, hiperlogovi, geoprostorni indeksi i tokovi. Nizovi su nizovi znakova koji se često koriste za spremanje tekstualnih podataka, kao što su imena, adrese ili poruke, a mogu se koristiti i za predstavljanje binarnih podataka. Hashevi su podatkovne strukture koje pohranjuju parove ključ-vrijednost, omogućujući učinkovito dohvaćanje i pohranu podataka. Obično se koriste za predstavljanje objekata ili zapisa s višestrukim atributima. Popisi predstavljaju uređene zbirke elemenata, koji mogu biti bilo koje vrste podataka, dopuštajući operacije poput dodavanja, uklanjanja i dohvaćanja elemenata s oba kraja popisa. Skupovi su zbirke jedinstvenih elemenata, čime se osigurava da se svaki element pojavljuje samo jednom. Korisni su za pohranjivanje i rukovanje podacima gdje su važne različite vrijednosti. Sortirani skupovi

slični su skupovima, ali s pridruženim rezultatom za svaki element. Omogućuju učinkovite upite temeljene na rasponu, što ih čini prikladnima za aplikacije koje uključuju rangiranje, rezultate ili poredane podatke. Bitmape su kompaktne podatkovne strukture koje se koriste za predstavljanje skupova cijelih brojeva ili binarnih vrijednosti pomoću bitova. Obično se koriste za rješavanje problema koji uključuju članstvo u skupu ili brojanje. Geoprostorni indeksi koriste se za pohranu i upite podataka temeljenih na lokaciji, omogućujući aplikacijama upravljanje i traženje objekata na temelju njihovih zemljopisnih koordinata. Redis ima ugrađenu replikaciju, Lua skriptiranje, izbacivanje LRU-a, transakcije i različite razine postojanosti na disku te pruža visoku dostupnost putem Redis Sentinela i automatskog particioniranja s Redis Clusterom.³⁵ Prvobitno ga je razvio Salvatore Sanfilippo i izdao ga je 2009. Redis je kratica za Remote Dictionary Server odnosno za pohranu u memoriji otvorenog koda, koja se koristi kao distribuirana baza podataka s ključevima i vrijednostima u memoriji, predmemoriji i broker porukama te je poznat po svojim visokim performansama, jednostavnosti i svestranosti. Redisova sposobnost čuvanja podataka u memoriji, omogućavajući nevjerojatno brz pristup podacima i dohvaćanje, jedna je od njegovih velikih prednosti. Podaci se pohranjuju kao parovi ključ-vrijednost pomoću paradigme pohrane ključ-vrijednost u memoriji. Zbog toga je Redis najbolja opcija za aplikacije s niskom latencijom pristupa podacima kao što su analitika u stvarnom vremenu, predmemorija sesija i web aplikacije s velikim prometom. Latencija označava vrijeme koje je potrebno da paket podataka prijeđe s jednog mjesta na drugo. Smanjenje latencije važan je dio izgradnje dobrog korisničkog iskustva.

Nizovi, hashovi, popisi, skupovi, sortirani skupovi i više samo su neki od nekoliko tipova podataka koje Redis nudi. Ove vrste podataka omogućuju različite operacije manipulacije podacima i nude svestranost. Snažne naredbe i operacije za manipulaciju podacima uključujući atomsko povećanje i smanjenje, guranje i iskakanje popisa, presjecišta skupova i unije te operacije sortiranih skupova također su uključeni u Redis. Redisova podrška za postojanost podataka još je jedna ključna značajka. Kako bi se spriječio gubitak podataka u slučaju kvara ili ponovnog pokretanja sustava, korisnicima omogućuje uspostavljanje postojanosti podataka na disku. Redis korisnicima nudi niz izbora postojanosti, uključujući snimanje i postojanost datoteke samo za dodavanje Append Only File (AOF), dopuštajući im da odaberu rješenje koje najbolje odgovara njihovim potrebama.³⁶

³⁵ <https://redis.io/docs/about/> (15.06.2023)

³⁶ <https://redis.io/docs/about/> (19.06.2023)

Napredna skalabilnost i karakteristike visoke dostupnosti također su dostupne u Redisu. Nudi master-slave replikaciju, koja osigurava redundanciju podataka i toleranciju na pogreške repliciranjem podataka na brojne slave čvorove. Još jedna značajka koja omogućuje dijeljenje podataka i distribuciju kroz brojne čvorove, a istovremeno nudi horizontalnu skalabilnost i balansiranje opterećenja je Redis Cluster. Redis također ima pub/sub mogućnosti, koje programima omogućuju objavljivanje i pretplatu na događaje ili poruke. Zbog toga je Redis dobra opcija za stvaranje arhitektura vođenih događajima, aplikacija za chat i sustava za razmjenu poruka u stvarnom vremenu. Redis je poznat po tome što je jednostavan za korištenje. Nudi klijentske biblioteke za mnoge programske jezike i jednostavno, intuitivno sučelje naredbenog retka (CLI), što olakšava povezivanje s brojnim aplikacijama i okvirima.³⁷ Zaključno, Redis predstavlja fleksibilnu i učinkovitu pohranu podataka u memoriji koja pruža brz pristup podacima, veliki izbor struktura podataka, postojanost podataka, skalabilnost i pub/sub sposobnost. To je predstavlja vrhunsku opciju za predmemoriju, aplikacije u stvarnom vremenu i razne druge slučajeve upotrebe zbog svoje jednostavnosti, brzine i svestranosti.

³⁷ <https://redis.io/docs/management/scaling/> (22.06.2023)

4.4. Primjena NoSQL sustava

NoSQL (ne samo SQL) baze podataka postale su dobro poznata zamjena za konvencionalne SQL baze podataka, pružajući fleksibilnu i skalabilnu metodu administracije podataka. NoSQL baze podataka imaju arhitekturu bez sheme ili fleksibilnu shemu, koja omogućuje učinkovitu obradu nestrukturiranih, polustrukturiranih podataka i podataka koji se brzo razvijaju, za razliku od strukturirane i krute prirode SQL baza podataka. Popularnost NoSQL baza podataka značajno je porasla u posljednjih nekoliko godina kao rezultat brzog porasta količine podataka, raznolikosti vrsta podataka i potražnje za obradom u stvarnom vremenu.

„Big data“ i analitika - Upravljanje velikim podacima i osnaživanje analitičke platforme dva su područja u kojima NoSQL baze podataka postižu sjajne rezultate. Oni mogu učinkovito pohraniti i obraditi velike količine nestrukturiranih ili polustrukturiranih podataka, uključujući feedove društvenih medija, log datoteke, podatke senzora i podatke o pregledima. NoSQL baze podataka pružaju distribuiranu obradu, unos podataka u stvarnom vremenu i horizontalno skaliranje, omogućujući stvaranje analitike i uvida u stvarnom vremenu.³⁸

Web i mobilne aplikacije - Izrada web stranica i mobilnih aplikacija često koristi NoSQL baze podataka. Oni nude fleksibilnost i skalabilnost potrebnu za upravljanje radnim opterećenjima s velikim prometom, sadržajem koji generiraju korisnici i modelima podataka koji se brzo mijenjaju. NoSQL baze podataka omogućuju brzo i fleksibilno pohranjivanje i dohvaćanje podataka za mobilnu sinkronizaciju, aplikacije u stvarnom vremenu i sustave za upravljanje sadržajem.³⁹

Internet stvari (IoT) - Ogromne količine podataka moraju se apsorbirati, pohraniti i trenutno obraditi zbog rasta IoT uređaja. NoSQL baze podataka mogu rukovati vremenskim serijama podataka, horizontalno se širiti i nuditi brz unos podataka, što ih čini idealnim za obradu IoT podataka. Oni su ključni za IoT platforme jer omogućuju upravljanje podacima senzora, nadzor uređaja i analitiku u stvarnom vremenu.⁴⁰

³⁸ <https://www.datastax.com/guides/nosql-use-cases> (17.07.2023)

³⁹ Ibidem

⁴⁰ Ibidem

Isporuca sadržaja i predmemorija - NoSQL baze podataka, posebno pohrane ključeva i vrijednosti, često se koriste u slojevima predmemoriranja i mrežama za isporuku sadržaja (CDN). Savršeni su za predmemoriju podataka kojima se često pristupa i distribuciju materijala na svjetskoj razini jer pružaju brzi pristup podacima, vrijeme odziva niske latencije i horizontalnu skalabilnost.⁴¹

Sustavi za personalizaciju i preporuke - Izgradnja sustava za personalizaciju i preporuke u stvarnom vremenu zahtijeva korištenje NoSQL baza podataka. Omogućuju učinkovito pohranjivanje i dohvaćanje korisničkih profila, postavki i podataka o ponašanju, omogućujući aplikacijama da ponude preporuke u stvarnom vremenu i personalizirana iskustva.⁴²

NoSQL baze podataka nude fleksibilno, skalabilno i agilno rješenje za upravljanje podacima u vrijeme kada se tvrtke suočavaju sa sve većim poteškoćama s podacima. Oni su atraktivna opcija za suvremene aplikacije koje se temelje na podacima zbog svoje sposobnosti rukovanja različitim vrstama podataka, horizontalnog skaliranja i podrške za obradu u stvarnom vremenu.

⁴¹ Sadalage, P. J., & Fowler, M.: NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, 2012

⁴² Ibidem

5. USPOREDBA SQL I NOSQL BAZA PODATAKA

5.1. Usporedba performansi

Kada se uspoređuju sustavi baza podataka, performanse su ključno polje koje se treba promotriti jer imaju izravan utjecaj na učinkovitost i odziv aplikacija. U ovom ću dijelu usporediti operacije čitanja i pisanja, skalabilnost i značajke izvedbe obrade upita SQL (Structured Query Language) i NoSQL (Not Only SQL) baza podataka. Tri ključne stvari o kojima ću pričati su: operacije čitanja i pisanja, skalabilnost te obrada upita.

Složene upite za čitanje, osobito one koji uključuju nekoliko tablica i zamršene spojeve, izuzetno dobro obrađuju SQL baze podataka. SQL baze podataka nude snažne i optimizirane mehanizme izvršenja upita zahvaljujući svom relacijskom dizajnu i podršci za jezik upita SQL, što ih čini učinkovitima za sofisticirano dohvaćanje podataka. Međutim, zahtjev da se zadovolje zahtjevi referentnog integriteta i očuvaju ACID kvalitete mogu imati utjecaj na to koliko se dobro izvode operacije pisanja kao što su umetanja, ažuriranja i brisanja. Kada radite s velikim količinama podataka, obrada nekih operacija može potrajati dulje.⁴³ Atomičnost jamči da se sve naredbe koje čine transakciju tretiraju kao jedna cjelina i da zajedno uspijevaju ili ne uspijevaju. Ovo je važno jer u slučaju neželjenog događaja, poput pada ili nestanka struje, možemo biti sigurni u stanje baze podataka. Transakcija bi ili bila uspješno dovršena ili bi bila vraćena ako bilo koji dio transakcije nije uspio. Dosljednost jamči da su promjene napravljene unutar transakcije u skladu s ograničenjima baze podataka. To uključuje sva pravila, ograničenja i okidače. Ako podaci dospiju u nedopušteno stanje, cijela transakcija propada. Izolacija osigurava da se sve transakcije odvijaju u izoliranom okruženju. To omogućuje istovremeno izvođenje transakcija jer transakcije ne ometaju jedna drugu. Trajnost jamči da će se zadržati nakon što se transakcija završi i promjene upišu u bazu podataka. Ovo osigurava da će podaci unutar sustava postojati čak i u slučaju kvarova sustava poput pada ili nestanka struje.⁴⁴ S druge strane, NoSQL baze podataka pružaju izvrsnu izvedbu pisanja i nisku latenciju za jednostavne operacije čitanja i pisanja. NoSQL baze podataka mogu se učinkovito nositi s radnim opterećenjima koja zahtijevaju intenzivno čitanje zahvaljujući svojim prilagodljivim modelima podataka jer često koriste denormalizirane strukture podataka koje eliminiraju

⁴³ Hellerstein, J. M., Stonebraker, M., & Hamilton, J.: Architecture of a Database System, 2007.

⁴⁴ <https://www.mongodb.com/basics/acid-transactions/> (10.08.2023)

potrebu za zamršenim spajanjima. NoSQL baze podataka mogu postići visoku izvedbu pisanja zbog nedostatka stroge provedbe sheme i mogućnosti distribucije podataka preko nekoliko čvorova, što ih čini prikladnima za aplikacije koje zahtijevaju brzi unos podataka i ažuriranja u stvarnom vremenu.

Opcije skalabilnosti dostupne su u SQL i NoSQL bazama podataka, iako imaju različite pristupe. Proširivanjem hardverskih mogućnosti jednog poslužitelja, SQL baze podataka često mogu prihvatiti povećana radna opterećenja. Ova opcija ima svoja ograničenja jer postoji točka u kojoj je dodatno skaliranje ili neizvedivo ili skupo.

S druge strane, NoSQL baze podataka izgrađene su za horizontalnu skalabilnost. One mogu distribuirati podatke kroz brojne čvorove klastera, omogućujući glatko širenje uvođenjem dodatnih čvorova za rukovanje rastućim količinama podataka i prometom. NoSQL baze podataka su visoko skalabilne i prikladne za aplikacije s potrebom za velikom propusnošću i brzim rastom podataka budući da se ističu u rukovanju velikim radnim opterećenjima podataka i postizanju visoke dostupnosti korištenjem distribuiranih arhitektura.

Napredne metode indeksiranja i optimizacije upita dostupne su u SQL bazama podataka, što omogućuje brzu obradu upita. Budući da su SQL baze podataka strukturirane, moguće je koristiti sofisticirane upite koji kombiniraju mnoge tablice, agregacije i sortiranje. Za procjenu planova upita i odabir najučinkovitije tehnike izvršenja na temelju indeksa i statistike, SQL baze podataka koriste planere i optimizatore upita.

S druge strane, NoSQL baze podataka često se odriču određene fleksibilnosti upita i sofisticiranih značajki postavljanja upita radi skalabilnosti i performansi. NoSQL baze podataka često nude jednostavne modele upita koji su specijalizirani za određene podatkovne modele (kao što su ključ-vrijednost, dokument ili grafikon) i ne mogu omogućiti sofisticirana spajanja ili proizvoljne upite. Međutim, izvrsni su u ponudi brzih pretraga i dohvaćanja podataka putem indeksiranih polja ili glavnih ključeva.

Ključno je zapamtiti da brojne varijable, uključujući količinu podataka, hardversku arhitekturu, algoritme indeksiranja i uzorke upita, mogu utjecati na to koliko dobro rade SQL i NoSQL baze podataka. Prilikom procjene karakteristika izvedbe važno je pažljivo uzeti u obzir potrebe za radnim opterećenjem i određeni slučaj upotrebe.

5.2. Usporedba skalabilnosti

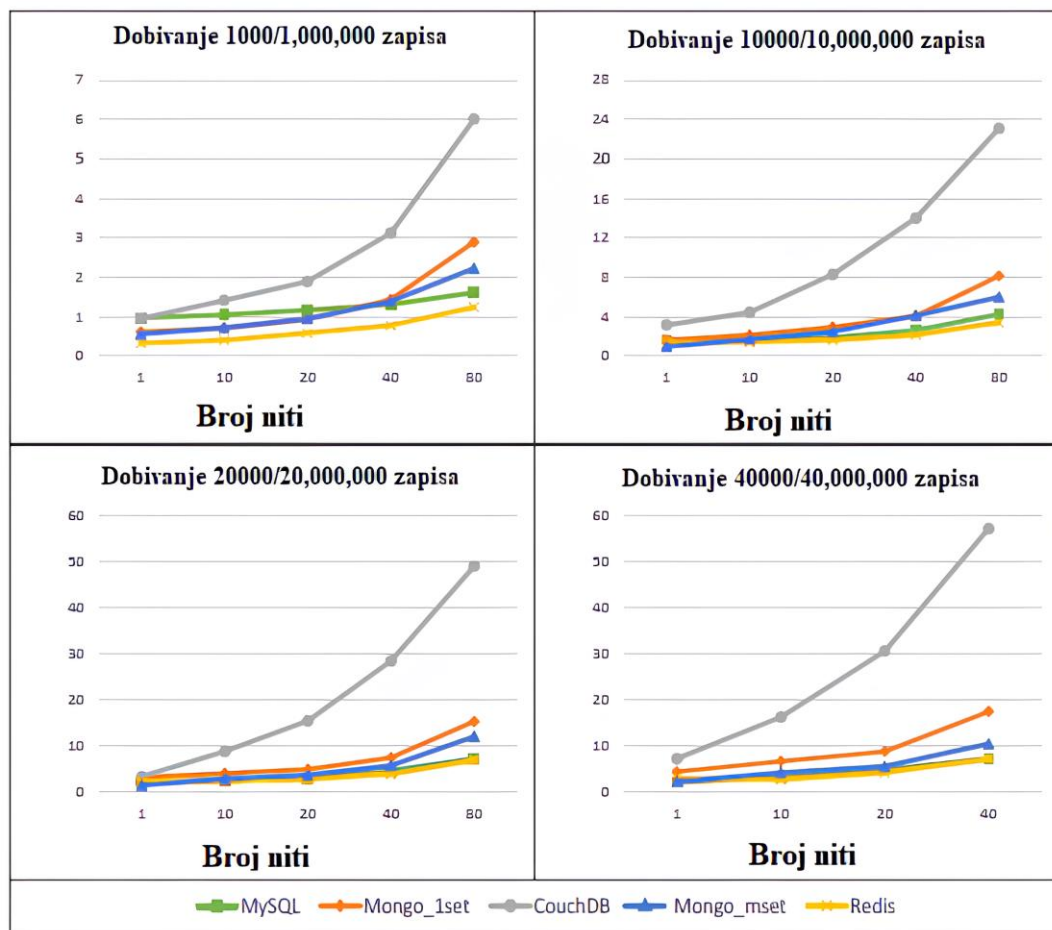
Prilikom razvoja i postavljanja sustava baza podataka, skalabilnost je važan faktor koji treba uzeti u obzir jer utječe na sposobnost sustava da se nosi s rastućim količinama podataka, konkurentnošću korisnika i potrebama radnog opterećenja. Svojstva skalabilnosti baza podataka SQL i NoSQL uspoređivat će se u ovom dijelu, s naglaskom na njihove strategije za horizontalno i vertikalno skaliranje, distribuciju podataka i rukovanje velikim radnim opterećenjem. NoSQL baze podataka dobro su poznate po svojoj sposobnosti horizontalnog skaliranja ili "scale-out".⁴⁵ Kako radno opterećenje i količina podataka rastu, više čvorova može se dodati u klaster baze podataka kako bi se radno opterećenje i podaci rasporedili na više poslužitelja. Povećana snaga procesora, kapacitet pohrane i performanse sustava omogućeni su horizontalnim skaliranjem. Dodatno, budući da baza podataka može nastaviti raditi čak i ako jedan ili više čvorova zataji, poboljšava se tolerancija na greške. Usvajanjem strategija poput dijeljenja, u kojem se podaci dijele i šire među nekoliko čvorova, i replikacije, u kojoj se kopije podataka čuvaju na različitim poslužiteljima kako bi se zajamčila dostupnost podataka, NoSQL baze podataka mogu se horizontalno skalirati. Povijesno gledano, SQL baze podataka oslanjale su se na vertikalno skaliranje, koje se obično naziva "scale-up". U vertikalnom skaliranju, hardver jednog poslužitelja se nadograđuje kako bi se proširili resursi sustava, kao što su CPU, memorija i pohrana. Iako okomito skaliranje može poboljšati performanse, ono ima ograničenja na količinu hardvera koji se može koristiti odjednom i popratne troškove. Kako bi se nosilo sa zahtjevnijim radnim opterećenjima, povećanje SQL baze podataka moglo bi zahtijevati skupe hardverske nadogradnje ili specijalizirane postavke. Međutim, nedavni napredak u SQL bazama podataka dodao je podršku za distribuirane sustave i horizontalno skaliranje, omogućujući povećanu skalabilnost ekvivalentnu NoSQL bazama podataka.

Distribuirane arhitekture se uzimaju u obzir prilikom izrade NoSQL baza podataka. Omogućuju visoku dostupnost, toleranciju grešaka i uravnoteženje opterećenja učinkovitom distribucijom i repliciranjem podataka na nekoliko čvorova. Particioniranje podataka i distribuirana obrada podataka omogućeni su činjenicom da svaki čvor u NoSQL klasteru može pohraniti dio podataka. Opterećenje pojedinačnih čvorova je smanjeno kao rezultat

⁴⁵ Stonebraker, M.: SQL databases v. NoSQL databases, 2010.

diseminacije podataka, što poboljšava performanse. Kako bi bili sigurni da su podaci uspješno diseminirani i dostupni, NoSQL baze podataka koriste razne tehnike distribucije podataka, kao što su dosljedno raspršivanje, particioniranje raspona i procedure replikacije podataka.

Tradicionalne SQL baze podataka koriste centraliziranu arhitekturu, gdje podatke čuva i kontrolira jedan poslužitelj. Oni su, međutim, prilagodili strategije distribuirane obrade podataka kao rezultat popularnosti distribuiranih SQL baza podataka. Kako bi se postigla horizontalna skalabilnost i diseminacija podataka kroz brojne čvorove, distribuirane SQL baze podataka koriste tehnologije replikacije, dijeljenja i distribucije podataka. Ove baze podataka kombiniraju snagu mogućnosti SQL upita s distribuiranom pohranom i obradom podataka kako bi ponudile prednosti i SQL i NoSQL baza podataka. Na slici broj 8 možemo vidjeti kako različiti programi funkcioniraju pri dohvatanju određenog broja zapisa.



Grafikon 2. Prikaz sposobnosti dohvatanja podataka raznih sustava (latencija po sekundi)

Izvor: Cloud Databases for Internet-of-Things Data, Thi Anh Mai Phan, Kongens Lyngby 2013,

Technical University of Denmark Informatics and Mathematical Modelling

„Slika 8 prikazuje prosječnu latenciju upita (uključujući vrijeme povezivanja) s obzirom na različite brojeve niti u četiri testna paketa, od kojih svaki ima različitu veličinu baze podataka. Već na prvi pogled može se vidjeti da je CouchDB značajno nadmašio sve ostale baze podataka, dok su razlike među ostalima bile manje značajne. Na primjer, skup podataka od ukupno 20 milijuna zapisa, s 80 istodobnih klijenata, za dohvaćanje 20 000 zapisa o čvoru, CouchDB-u je trebalo gotovo 50 sekundi, ali samo manje od 7 sekundi za MySQL i Redis, s Redisom malo bržim. Ova latencija Mongo_1seta i Mongo_mseta bila je 15 odnosno 12 sekundi. Općenito, Redis je dobio najbolje rezultate zahvaljujući svojoj pohrani u memoriji i prirodi upita po ključu u ovom slučaju. Istodobno, u slučajevima kada je veličina baze podataka bila velika (10.000.000 zapisa ili više), MySQL je pomno pratio Redis i vrlo brzo postavljao upite.“⁴⁶

Nadalje, postignut je ogroman napredak u sposobnosti SQL baza podataka, posebno distribuiranih SQL baza podataka, da upravljaju velikim radnim opterećenjem. SQL baze podataka mogu upravljati većim količinama podataka i istodobnim pristupom koristeći poboljšane metode optimizacije upita, distribuiranu obradu upita i algoritme za particioniranje podataka. Distribuirane SQL baze podataka često kombiniraju skalabilnost i toleranciju na pogreške NoSQL baza podataka s prednostima klasičnih SQL baza podataka, kao što su snažna jamstva dosljednosti i sveobuhvatne mogućnosti upita. Odabir između SQL i NoSQL baza podataka za skalabilnost je ključan pri odabiru između SQL i NoSQL sustava jer ovisi o nizu varijabli, uključujući određeni slučaj upotrebe, model podataka i potrebe za izvedbom. Za situacije koje uključuju visoku propusnost pisanja, unos podataka u stvarnom vremenu i veliku skalabilnost, NoSQL baze podataka češće će se koristiti. Prednost SQL baza podataka, posebno distribuiranih SQL baza podataka, je u tome što kombiniraju poznavanje SQL upita s mogućnošću horizontalnog širenja i prilagođavanja kompliciranim potrebama obrade podataka.

5.3. Usporedba dosljednosti

Stupanj do kojeg baza podataka podržava integritet podataka i osigurava da svi klijenti dobiju isti prikaz podataka u bilo kojem trenutku naziva se dosljednošću. O snažnoj dosljednosti, eventualnoj dosljednosti i kompromisima između dosljednosti i performansi raspravljat ćemo u ovom dijelu dok uspoređujemo značajke dosljednosti SQL i NoSQL baza podataka. SQL baze podataka prema zadanim postavkama daju snažna sigurnost kada je u

⁴⁶ Thi Anh Mai Phan, Kongens Lyngby: Cloud Databases for Internet-of-Things Data, Technical University of Denmark Informatics and Mathematical Modelling, 2013.

pitanju dosljednost. Svaka operacija čitanja uvijek će primiti najnoviju verziju podataka zahvaljujući snažnoj dosljednosti. SQL baze podataka to čine pomoću karakteristika ACID-a. U SQL bazi podataka, predaja transakcije osigurava da će se promjene napravljene transakcijom odraziti na sva posljedična čitanja. Jaka dosljednost prikladna je za aplikacije koje trebaju osigurati snažan integritet podataka i da svi klijenti uvijek vide istu verziju podataka. Međutim, stroga ograničenja dosljednosti često su ublažena u NoSQL bazama podataka kako bi se poboljšala izvedba i skalabilnost. Oni podržavaju eventualnu dosljednost, koja kaže da iako će se promjene u bazi podataka s vremenom proširiti na sve replike, neće sve operacije čitanja nužno odražavati te promjene odmah. Operacije visoke dostupnosti i niske latencije omogućene su konačnom dosljednošću jer svaka replika može funkcionirati neovisno i odgovarati na zahtjeve bez konzultiranja s drugim replikama. Replikacija baze podataka odnosi se na proces kopiranja podataka iz primarne baze podataka u jednu ili više replika baza podataka kako bi se poboljšala dostupnost podataka i otpornost na pogreške i pouzdanost sustava.⁴⁷ Jaka dosljednost žrtvovana je u korist bolje skalabilnosti i tolerancije na greške, ali po cijenu jake dosljednosti.

Eventualna dosljednost je kompromis koji su usvojile NoSQL baze podataka kako bi ponudile poboljšanu skalabilnost i dostupnost. Istodobna ažuriranja moguća su zahvaljujući konačnoj dosljednosti, koja eliminira zahtjev za trenutnom sinkronizacijom između svih replika. Sustav osigurava da sve replike na kraju konvergiraju u dosljedno stanje iako pojedinačne replike mogu imati različite verzije podataka u bilo kojem trenutku. „Konvergencija je spajanje dvaju različitih subjekata, a u kontekstu računalstva i tehnologije integracija je dviju ili više različitih tehnologija u jedan uređaj ili sustav.“⁴⁸ „Dosljednost podataka je točnost, potpunost i ispravnost podataka pohranjenih u bazi podataka. Isti podaci u svim povezanim sustavima, aplikacijama i bazama podataka su kada kažemo da su podaci dosljedni.“⁴⁹ Algoritmi za izradu verzija, rješavanje sukoba i usklađivanje koriste se za postizanje ove konvergencije. Kroz distribuirane dizajne i tehnike replikacije, SQL baze podataka također mogu pružiti konačnu dosljednost. Distribuirane SQL baze podataka mogu postići konačnu dosljednost uz očuvanje fleksibilnosti i snage SQL upita uključivanjem distribuiranih transakcija i tehnika rješavanja sukoba. Iako se eventualna dosljednost uvodi kao

⁴⁷ <https://www.qlik.com/us/data-replication/database-replication> (15.08.2023)

⁴⁸ <https://www.techopedia.com/definition/769/convergence> (19.08.2023)

⁴⁹ <https://www.decube.io/post/what-is-data-consistency-definition-examples-and-best-practice> (19.08.2023)

izborna značajka za specifične slučajeve upotrebe koji mogu prihvatiti eventualnu konvergenciju podataka, jaka dosljednost često je zadani način rada u SQL bazama podataka.

Odluka između SQL i NoSQL baza podataka ovisi o potrebama i prioritetima aplikacije jer su dosljednost i izvedba često u suprotnosti. Konzistentnost podataka u različitim istodobnim operacijama ima prioritet u SQL bazama podataka, koje također nude transakcijska jamstva. Međutim, u distribuiranim kontekstima gdje su potrebna koordinacija i sinkronizacija između čvorova, ova visoka dosljednost može doći nauštrb smanjene izvedbe.⁵⁰ Kako je već spomenuto, snažna dosljednost često se žrtvuje u korist performansi i skalabilnosti kada se koriste NoSQL baze podataka. NoSQL baze podataka mogu pružiti visoku propusnost pisanja, operacije niske latencije i besprijekornu skalabilnost labavljenjem jamstava dosljednosti. Eventualna dosljednost eliminira zahtjev za koordinacijom i promiče veću konkurentnost dopuštajući neovisne aktivnosti na mnogim replikama. Iako to ima pozitivan učinak na izvedbu, također povećava mogućnost čitanja zastarjelih ili netočnih podataka prije nego što se konačno postigne konvergencija.⁵¹

⁵⁰ <https://www.ibm.com/blog/sql-vs-nosql/> (05.08.2023)

⁵¹ Stonebraker, M.: SQL databases v. NoSQL databases, 2010.

5.4. Usporedba modeliranja podataka

Budući da određuje kako su podaci raspoređeni, strukturirani i povezani unutar sustava baze podataka, modeliranje podataka bitno je za arhitekturu baze podataka. S fokusom na fleksibilnost sheme, veze podataka i rukovanje kompliciranim strukturama podataka, u ovom ćemo odjeljku ispitati svojstva modeliranja podataka SQL i NoSQL baza podataka. Strogi zahtjevi za shemu u SQL bazama podataka dobro su poznati. Oni se pridržavaju unaprijed utvrđene sheme koja navodi strukturu baze podataka, uključujući njezine tablice, stupce, tipove podataka i odnose. Promjena strukture tablice i promjena postojećih podataka često su potrebni promjenama sheme, što može oduzimati vrijeme i utjecati na dostupnost baze podataka. Kroz ograničenja i odnose koje pruža shema, SQL baze podataka podržavaju kvalitete ACID-a i osiguravaju integritet podataka. S druge strane, opcije fleksibilne sheme dostupne su u NoSQL bazama podataka. Često su poznate kao baze podataka bez sheme ili fleksibilne sheme. NoSQL baze podataka omogućuju dinamičke strukture podataka u razvoju s mogućnošću da svaki zapis ili dokument ima jedinstvenu shemu. Programeri mogu dodavati ili mijenjati polja unutar podataka zahvaljujući ovoj fleksibilnosti bez mijenjanja cjelokupne sheme baze podataka.⁵² Budući da baza podataka može podržati promjenu zahtjeva podataka bez potrebe za prilagodbama sheme ili zastoja, čini razvojni proces jednostavnijim. Budući da NoSQL baze podataka koriste "shemu pri čitanju" umjesto "sheme pri pisanju", fleksibilnije su i agilnije.

SQL baze podataka fantastično obrađuju složene veze podataka. Kroz stvaranje veza između tablica pomoću primarnih i stranih ključeva, oni daju ideju o odnosima. Referentni integritet nametnut je SQL bazama podataka, čuvajući podatkovne veze i poštujući pravila integriteta podataka. Zbog toga su SQL baze podataka prikladne za aplikacije sa zamršenim modelima podataka i visoko strukturiranim, međusobno povezanim podacima. Tradicionalne operacije spajanja koje se nalaze u SQL bazama podataka često nisu podržane od strane NoSQL baza podataka, osobito baza podataka orijentiranih na dokumente. Umjesto toga, promoviraju denormalizirane modele podataka u kojima su informacije organizirane hijerarhijski ili ugniježdene unutar dokumenta. Kako bi izrazili odnose između dijelova podataka, NoSQL baze podataka često koriste ugrađene dokumente ili nizove. Iako su neke od kompliciranih mogućnosti spajanja žrtvovane u ovoj metodi, ona pruža vrhunsku izvedbu za aktivnosti intenzivnog čitanja i omogućuje brže dohvaćanje podataka.

⁵²
(18.08.2023)

<https://dev.to/harperdb/strict-schema-enforcement-vs-schemaless-vs-dynamic-schema-333a>

NoSQL baze podataka mogu učinkovitije rukovati složenim i nestrukturiranim tipovima podataka. Imaju izvornu podršku za niz formata podataka, uključujući binarne formate, XML i JSON, omogućujući pohranu i dohvaćanje različitih struktura podataka. NoSQL baze podataka posebno su prikladne za situacije u kojima podaci moraju biti fleksibilni u svojoj shemi ili kada je potrebno pohraniti polustrukturirane i polimorfne podatke. Iako su SQL baze podataka uglavnom usmjerene na organizirane podatke s određenom shemom, one obrađuju različite vrste podataka.⁵³ U SQL bazama podataka, rukovanje kompliciranim strukturama podataka često zahtijeva dodatni rad, kao što je stvaranje različitih tablica, definiranje asocijacija ili korištenje specijaliziranih tipova podataka. Kada se radi s iznimno raznolikim i nestrukturiranim podacima, SQL baza podataka možda neće ponuditi istu razinu fleksibilnosti i upotrebljivosti kao NoSQL baza podataka, iako se mogu rukovati kompliciranim podacima kada se koriste odgovarajući pristupi modeliranja i normalizacije.

Zaključno, SQL baze podataka imaju strogu i definiranu shemu koja jamči integritet podataka i prilagođava zamršene interakcije između tablica. Nasuprot tome, NoSQL baze podataka nude prilagodljive opcije sheme koje programerima omogućuju da zadovolje promjenjive potrebe podataka bez potrebe za promjenom modela. NoSQL baze podataka nude veću agilnost pri radu s kompliciranim strukturama podataka i izvrsne su u upravljanju nestrukturiranim ili polustrukturiranim podacima. Vrsta podataka, potrebe aplikacije, potreba za fleksibilnošću sheme i potreba za povezivanjem podataka utječu na odluku između SQL i NoSQL baza podataka za modeliranje podataka.

⁵³ <https://www.ibm.com/blog/sql-vs-nosql/> (05.08.2023)

5.5. Usporedba jednostavnosti korištenja

Jednostavnost korištenja, vjerojatno je jedan od najvažnijih stavki kada su u pitanju baze podataka. Značajke jednostavne upotrebe SQL i NoSQL baza podataka uspoređivat će se u ovom dijelu s naglaskom na modeliranje podataka, jezik upita i jednostavnost razvoja. Većina SQL baza podataka ima čvrste sheme i zahtijeva modeliranje podataka unaprijed. Tablice, stupci i odnosi moraju se definirati prije umetanja podataka jer su SQL baze podataka strukturirane. Ovo preliminarno modeliranje moglo bi oduzimati puno vremena i zahtijevati vještine dizajniranja baze podataka. SQL baze podataka, međutim, pružaju nedvosmisleni organizaciju podataka i provode integritet podataka kroz ograničenja nakon što je shema navedena, što ih čini prikladnima za aplikacije sa strogim zahtjevima za podacima i zamršenim vezama. Modeliranje podataka je fleksibilnije s NoSQL bazama podataka, posebice bazama podataka orijentiranim na dokumente.⁵⁴ Programeri mogu pohraniti i dohvatiti podatke bez potrebe za unaprijed postavljenim shemama zahvaljujući upotrebi tehnika bez shema ili shema-fleksibilnih tehnika. Budući da se promjene u strukturi podataka mogu prilagoditi bez ponovnog pisanja cijele arhitekture baze podataka, ova fleksibilnost čini razvoj jednostavnijim. Međutim, kako bi se očuvala dosljednost podataka, nepostojanje unaprijed određene sheme moglo bi zahtijevati daljnju provjeru valjanosti na razini aplikacije. U sljedećoj tablici prikazan je sažet prikaz jednostavnosti korištenja dva sustava.

| Prednosti SQL-a | Prednosti NoSQL-a |
|----------------------------------|-------------------------|
| Dobro dokumentirano | Fleksibilna shema |
| Dosljednost podataka | Brzina |
| Jednostavna sintaksa i struktura | Horizontalno skaliranje |
| Jednostavna agregacija | Razni modeli podataka |

Tablica 1. Prikaz prednosti kod jednostavnosti korištenja između dva sustava

Izradio autor

SQL sintaksa, koju koriste SQL baze podataka i koji ima razvijen sustav alata i biblioteka, već je dugo vremena naširoko prihvaćen. Programeri mogu izraziti svoje potrebe za dohvaćanjem podataka i manipulacijom koristeći SQL-ov deklarativni pristup bez brige o

⁵⁴ <https://www.techtarget.com/searchdatamanagement/definition/NoSQL-Not-Only-SQL> (17.08.2023)

specifičnostima implementacije. Programeri mogu lakše razumjeti, pisati i optimizirati upite jer jezik SQL upita ima standardiziranu sintaksu. NoSQL baze podataka koriste brojne jezike za stvaranje upita, svaki za određenu vrstu baze podataka. Na primjer, upitni jezici temeljeni na JavaScriptu ili sintaksi sličnoj JSON-u često se koriste s bazama podataka orijentiranim na dokumente. Ovim upitnim jezicima nedostaje univerzalnost i standardiziranost SQL-a, unatoč činjenici da su programeri koji se bave suvremenim online tehnologijama možda više navikli na njih. Možda postoji krivulja učenja za programere koji se prebacuju sa SQL baza podataka jer se NoSQL jezici upita mogu razlikovati u smislu složenosti i podrške alatima.

Zbog svog naslijeđa i postojeće tehnologije, SQL baze podataka dostupne su široj masi programera. Sustavi za upravljanje bazama podataka (DBMS) dolaze u širokoj paleti, svaki s korisničkim sučeljima prilagođenim korisniku, optimizacijom upita i administrativnim mogućnostima. Većina SQL baza podataka dolazi s opsežnom dokumentacijom, snažnom pomoći zajednice i obiljem izvora znanja. Daljnja prednost SQL baza podataka je ta što često nude integrirano upravljanje transakcijama i usklađenost s ACID-om, što olakšava stvaranje aplikacija za slučajeve upotrebe koji zahtijevaju strogu dosljednost podataka. Zbog raznolikosti vrsta baza podataka i jezika upita, NoSQL baze podataka, koje su praktički još u razvitku, mogu imati težu krivulju učenja. Međutim, NoSQL baze znaju se isticati dobrim i jednostavnim sučeljima prilagođenim programerima. Brojne NoSQL baze podataka nude korisničke API-je i podršku za dobro poznate programske jezike, pojednostavljujući interakciju između programera i baze podataka. Dodatno, NoSQL baze podataka podržavaju horizontalnu skalabilnost, koja usmjerava postupke skaliranja i može zadovoljiti sve veću potražnju aplikacija.⁵⁵

Za kraj, SQL baze podataka pružaju razvijeni sustav alata i biblioteka, standardizirani SQL jezik za upite i dobro uspostavljene metodologije modeliranja podataka. Prikladni su za aplikacije koje trebaju organizirane podatke i imaju složene podatkovne interakcije. Iako žrtvuju standardizirani jezik upita i neke od naprednijih mogućnosti, NoSQL baze podataka nude više slobode u modeliranju podataka. Odabir između SQL i NoSQL baza podataka radi jednostavnosti upotrebe ovisi o nizu elemenata, uključujući iskustvo programera sa SQL-om, vrstu podataka koji se pohranjuju i posebne potrebe aplikacije.

⁵⁵ <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems> (19.08.2023)

U tablici 2. prikazane su prednosti i nedostatci SQL i NoSQL sustava.

| Prednosti SQL baze podataka | Prednosti NoSQL baze podataka |
|--|---|
| Široko prihvaćen | Fleksibilna shema omogućuje dinamičke i evoluirajuće strukture podataka |
| Zreli sustav alata, biblioteka i okvira | Skalabilna i vrlo učinkovita za radna opterećenja koja zahtijevaju veliko čitanje |
| Snažan integritet podataka putem unaprijed definiranih shema i ograničenja | Pogodniji za rukovanje nestrukturiranim i raznolikim tipovima podataka |
| Podržava složene podatkovne odnose i spajanja | Pojednostavljen razvoj s intuitivnim API-jima i sučeljima prilagođenim programerima |
| Standardizirani jezik upita | Horizontalna skalabilnost za rukovanje velikom količinom i distribuiranim podacima |
| Usklađenost s ACID-om za upravljanje transakcijama | |

Tablica 2. Usporedba prednosti SQL i NoSQL baza podataka

Izvor: Izradio autor

5.6. Implementacija SQL i NoSQL baza podataka

SQL baze podataka često koriste mehanizme za pohranu podataka za kontrolu načina na koji se podaci čuvaju i dohvaćaju. SQL baze podataka često koriste različite mehanizme za pohranu podataka, uključujući:

InnoDB: Popularni mehanizam za pohranu podataka InnoDB je poznat po svojoj pouzdanosti i podršci za transakcije. Zaključavanje na razini retka, oporavak od pada i podrška za ograničenja stranog ključa neke su od značajki koje nudi. InnoDB koristi klasterirani indeks za fizičku organizaciju podataka, poboljšavajući izvedbu upita.

MyISAM: Drugi mehanizam za pohranu koji se često koristi u SQL bazama podataka je MyISAM. Iako mu nedostaje transakcijska sposobnost, pruža brzu izvedbu za radna

opterećenja koja uključuju puno čitanja. MyISAM koristi neklasterirani indeks, što čini čitanje bržim, ali pisanje sporijim od InnoDB-a.

SQL baze podataka također koriste strategije indeksiranja za poboljšanje izvedbe upita. Indeksi B-stabla često se koriste za olakšavanje učinkovitog pronalaženja podataka ovisno o određenim stupcima. Ovi su indeksi uravnotežene strukture stabla koje olakšavaju brzo umetanje i brisanje podataka, kao i pretraživanje. Sharding je metoda koja se koristi u SQL bazama podataka za horizontalnu distribuciju podataka preko brojnih poslužitelja baze podataka. To podrazumijeva razbijanje ogromnog skupa podataka u dijelove kojima se lakše upravlja, poznate kao krhotine. Zbog činjenice da se svaki shard čuva na drugom poslužitelju, moguća je paralelna obrada i povećana skalabilnost. Sharding se često koristi za upravljanje velikim količinama podataka i raspoređivanje tereta na nekoliko poslužitelja, poboljšavajući performanse čitanja i pisanja.

Zbog svojih raznolikih struktura podataka, NoSQL baze podataka zahtijevaju različite detalje implementacije od SQL baza podataka. Međutim, oni imaju neke slične ideje:

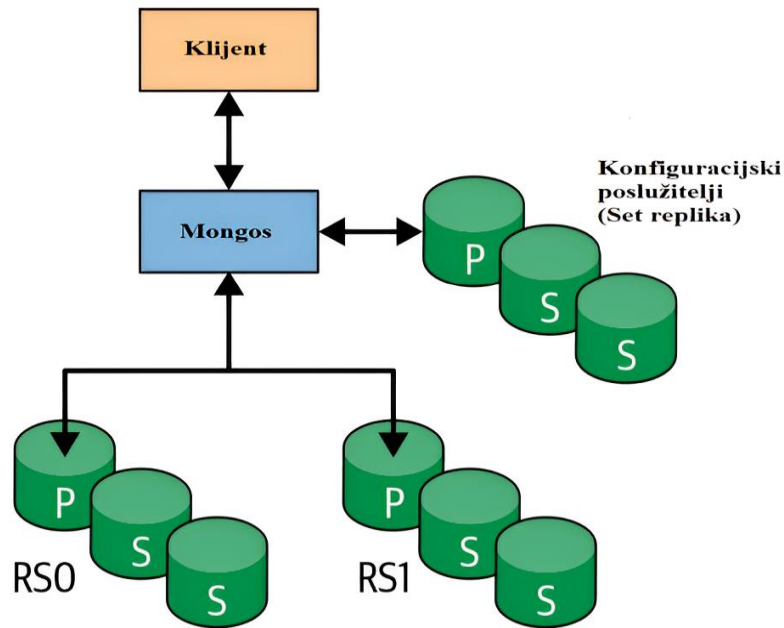
Motori za pohranu podataka: NoSQL baze podataka koriste različite mehanizme za pohranu podataka od kojih je svaki prilagođen određenom modelu podataka. Na primjer, mehanizme za pohranu kao što su WiredTiger ili RocksDB, koji pohranjuju podatke u fleksibilnom formatu dokumenta, koriste baze podataka orijentirane na dokumente kao što je MongoDB.

Indeksiranje: Ovisno o modelu podataka, NoSQL baze podataka dopuštaju različite mehanizme indeksiranja. Za učinkovito dohvaćanje podataka na temelju ključeva ili polja dokumenta, baze podataka orijentirane na dokumente, na primjer, često koriste indekse dokumenata ili indekse B-stabla. Za učinkovito prelaženje i postavljanje upita o podacima grafa, baze podataka grafova zahtijevaju specijalizirane strukture indeksiranja kao što su indeksi svojstava ili invertirani indeksi.

Sharding: U NoSQL bazama podataka, sharding je bitan pristup skaliranju. Podaci se u ovom procesu raspodjeljuju na određeni broj čvorova. Svaki shard može se čuvati na drugom

poslužitelju i sadrži dio podataka. Sharding omogućuje horizontalno skaliranje i povećanje performansi raspodjelom radnog opterećenja čitanja i pisanja.

Slika 9. prikazuje proces „shardinga“.



Slika 7. Prikaz „shardinga“ u MongoDB-u

Izvor: <https://nosqlmasterdb.wordpress.com/2020/08/25/overview-of-sharding-in-mongodb/>

Denormalizacija: Za poboljšanje performansi čitanja, NoSQL baze podataka često koriste denormaliziranu strategiju modeliranja podataka. Denormalizacija je grupiranje relevantnih podataka kako bi se smanjila potreba za zamršenim spajanjima i poboljšala učinkovitost upita. NoSQL baze podataka mogu ubrzati dohvaćanje podataka denormalizacijom podataka, posebno za radna opterećenja koja zahtijevaju puno čitanja.

6. NOVI TRENDVI U SUSTAVIMA ZA UPRAVLJANJE BAZAMA PODATAKA

Okolina pohrane, obrade i upravljanja podacima mijenja se kao rezultat novih razvoja sustava za upravljanje bazama podataka, otvarajući nove mogućnosti za digitalnu eru. Organizacije traže vrhunska rješenja za rješavanje širokih i stalno rastućih zahtjeva za podacima kako tehnologija napreduje i količine podataka rastu. Ovi novi trendovi pokazuju koliko su poboljšane performanse, skalabilnost, agilnost i isplativost potrebni u suvremenim sustavima vođenim podacima. Ove inovacije, koje sežu od najsuvremenijih modela baza podataka do arhitektura u oblaku, mijenjaju način na koji se podaci pohranjuju, kako im se pristupa i kako se obrađuju, omogućujući tvrtkama dobivanje pronicljivih informacija i donošenje odluka na temelju podataka s razinom brzine i učinkovitosti nikad prije moguće.

Rast baza podataka u memoriji, koje pohranjuju podatke uglavnom u memoriju sustava za razliku od konvencionalne pohrane na disku, jedna je od najnovijih rastućih tehnologija. Smanjivanjem latencije čitanja i pisanja, povećanjem analitike u stvarnom vremenu i omogućavanjem brže obrade transakcija, baze podataka u memoriji značajno povećavaju performanse. Baze podataka u memoriji pružaju uvid u podatke u stvarnom vremenu i omogućuju brzo izvršavanje složenih upita korištenjem brzine pristupa memoriji.

Potražnja za skalabilnim rješenjima, učinkovitim rukovanjem stalno rastućim količinama podataka i izvlačenjem uvida u stvarnom vremenu pokreću ove nove trendove u sustavima upravljanja bazama podataka. Organizacije mogu u potpunosti iskoristiti svoje podatke, steći konkurentsku prednost i poticati kreativnost u današnjem brzom digitalnom okruženju usvajanjem novih modela baza podataka, arhitekture izvorne u oblaku i novih pristupa upravljanju podacima.

6.1. NewSQL

Novi razvoj u sustavima za upravljanje bazama podataka, poznat kao NewSQL, ima za cilj zatvoriti jaz između NoSQL baza podataka i klasičnih SQL baza podataka kombinirajući najveće značajke obje vrste baza podataka. Nedostaci konvencionalnih SQL baza podataka su posebno riješeni newSQL bazama podataka, koje su u stanju nositi se s velikim količinama podataka i visokim stopama transakcija uz očuvanje pouzdanih jamstava dosljednosti. Ove baze podataka kombiniraju ACID karakteristike tradicionalnih SQL baza podataka s prednostima skalabilnosti i performansi NoSQL baza podataka. Kako bi razvili svoje sustave baza podataka na nekoliko poslužitelja i ispunili zahtjeve trenutnih aplikacija, poduzeća to mogu učiniti koristeći distribuirane arhitekture NewSQL baza podataka, koje pružaju horizontalnu skalabilnost. Koriste tehnike paralelne obrade, particioniranja podataka i dijeljenja kako bi zajamčili izvrsne performanse i podržali velika radna opterećenja.

Razlike između klasičnog SQL-a , NoSQL-a i NewSQL-a možemo vidjeti u tablici 3.

| | „Klasični“ SQL | NoSQL | NewSQL |
|----------------------------------|----------------|-------|--------|
| Relacijske baze | DA | NE | DA |
| SQL | DA | NE | DA |
| ACID transakcije | DA | NE | DA |
| Horizontalna skalabilnost | NE | DA | DA |
| Preformanse | NE | DA | DA |
| Uporaba sheme | NE | DA | NE |

Tablica 3. Usporedba NewSQL-a sa klasičnim SQL-om i NoSQL-om

Izvor: Izradio autor u uzoru na <https://labs.sogeti.com/newsq1-whats/> (29.06.2023.)

Dodatno, NewSQL baze podataka koriste vrhunske tehnike za optimizaciju upita i pohranu podataka. Kako bi poboljšali izvedbu upita i smanjili kašnjenje, mogli bi koristiti obradu u memoriji, tehnike predmemoriranja i pohranu u stupcima. Kako bi se omogućila brza obrada i dohvaćanje podataka, ove baze podataka koriste suvremene hardverske inovacije kao što su višejezgreni procesori i solid-state diskovi (SSD).

Kapacitet NewSQL baza podataka da upravljaju sofisticiranim i transakcijskim radnim opterećenjima, dok istovremeno pružaju horizontalnu skalabilnost, jedna je od njihovih glavnih prednosti. NewSQL baze podataka osiguravaju da se transakcije provode dosljedno i pouzdano preko brojnih čvorova, za razliku od NoSQL baza podataka, koje često mijenjaju jaku dosljednost za skalabilnost. Kao rezultat toga, aplikacije uključujući financijske sustave, platforme za e-trgovinu i poslovne aplikacije kritične za misiju koje zahtijevaju integritet podataka mogu koristiti NewSQL baze podataka.⁵⁶ NewSQL baze podataka CockroachDB, TiDB i VoltDB nekoliko su poznatih primjera. Visoka dostupnost, automatizirana replikacija podataka i distribuirane ACID transakcije značajke su distribuirane SQL baze podataka CockroachDB. TiDB je distribuirana baza podataka otvorenog koda koja spaja NoSQL-ovu skalabilnost sa SQL sučeljem kako bi tvrtkama omogućila horizontalni razvoj baza podataka i učinkovito rukovanje ogromnim količinama podataka. VoltDB je NewSQL baza podataka u memoriji koja se ističe u rukovanju podacima velike brzine s malim kašnjenjem, što je čini prikladnom za strujanje i analitiku u stvarnom vremenu.

Potreba za upravljanjem suvremenim zahtjevima za podacima, kao što su masivna obrada podataka, analitika u stvarnom vremenu i enormne količine transakcija, bez ugrožavanja robusnih jamstava dosljednosti, dovela je do rođenja NewSQL baza podataka. Za tvrtke koje traže prednosti skalabilnosti i performansi NoSQL baza podataka s pouzdanošću i transakcijskim mogućnostima konvencionalnih SQL baza podataka, ove baze podataka predstavljaju zanimljivu opciju.

⁵⁶ Kuno, H., Linskey, D., & Reed, B.: What's Really New with NewSQL?, 2012.

6.2. Višemodelne baze podataka

Trend u nastajanju u sustavima upravljanja bazama podataka koji pruža mogućnost rukovanja mnogim tipovima podataka i modelima unutar jednog sustava baze podataka su baze podataka s više modela. Baze podataka s više modela omogućuju različite modele podataka, kao što su ključ-vrijednost, grafikon, dokument i stupac, unutar kohezivnog okvira, za razliku od konvencionalnih baza podataka, koje se pridržavaju jednog modela podataka, poput relacijskih ili dokumentno orijentiranih. Umjesto da bude ograničena na jedan model, ova fleksibilnost omogućuje razvojnim programerima odabir podatkovnog modela koji je najprikladniji za njihove pojedinačne slučajeve upotrebe. Integracija podataka je jednostavnija i više nema potrebe za održavanjem zasebnih sustava baza podataka za različite vrste podataka zahvaljujući bazama podataka s više modela. Programeri mogu raditi s podacima na način koji najbolje odgovara potrebama njihove aplikacije zahvaljujući bazama podataka s više modela. Na primjer, mogu učinkovito rukovati podacima u stupcima za analitiku i pohranjivati visoko strukturirane podatke pomoću relacijskog modela, rukovati polustrukturiranim podacima ili podacima temeljenim na dokumentima pomoću modela dokumenta, upravljati kompliciranim odnosima pomoću modela grafikona, itd. Sposobnost rješavanja raznolikost slučajeve upotrebe, od transakcijskih sustava do analitike podataka i aplikacija za strojno učenje, sve unutar jedinstvenog okruženja baze podataka omogućeno je ovom fleksibilnošću.⁵⁷

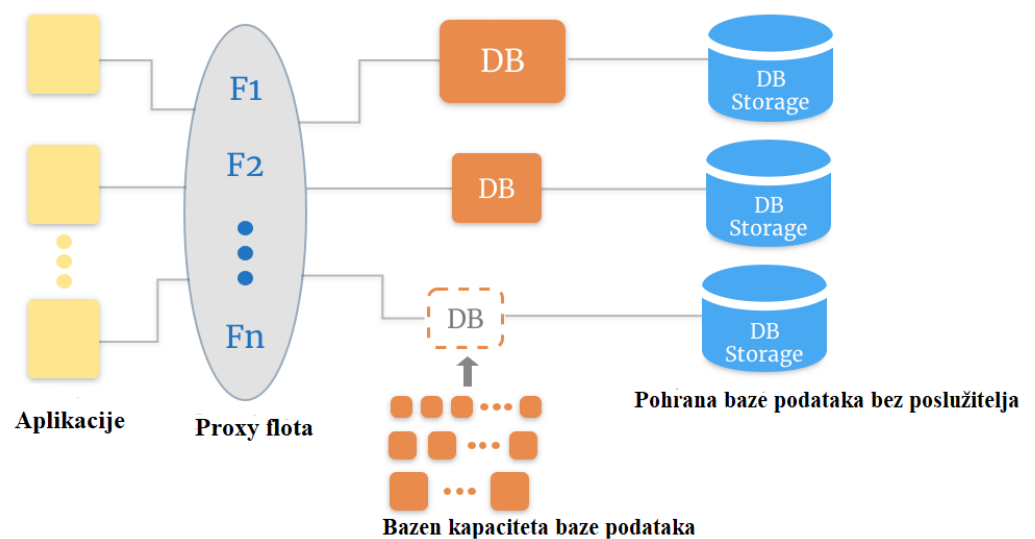
Baze podataka s više modela ArangoDB, OrientDB i Cosmos DB neki su dobro poznati primjeri. Programeri mogu jednostavno kreirati sofisticirane aplikacije uz pomoć ArangoDB-a, baze podataka s više modela otvorenog koda koja kombinira modele podataka ključ-vrijednost, dokument i grafikon u jednoj jedinici baze podataka. Grafikon, dokument, ključ-vrijednost i objektno orijentirani modeli integrirani su u višemodelnu OrientDB bazu podataka otvorenog koda, koja također ima jake mogućnosti indeksiranja i postavljanja upita. Pružajući skalabilnost i pristup niske latencije u svim regijama, Microsoft Azure Cosmos DB je globalno distribuirana usluga baze podataka s više modela koja podržava dokumente, grafikone, ključ-vrijednost i modele podataka u stupcima. Želja da se pojednostavi upravljanje podacima i otklone poteškoće povezivanja mnogih baza podataka za različite vrste podataka ono je što je dovelo do pojave baza podataka s više modela. Organizacije mogu poboljšati koherentnost podataka, minimizirati troškove održavanja i pojednostaviti razvojne napore

⁵⁷ Jovanovic, P., & Romero, O.: Multi-model Databases: A Paradigm Shift for Big Data Systems, 2014.

kombiniranjem nekoliko modela podataka u jedan sustav. Uz pomoć baza podataka s više modela, programeri mogu odabrati model podataka koji je najprikladniji za njihove aplikacije, koristeći prednosti svakog modela, a istovremeno čuvati integritet cijelog sustava.

6.3. Baze podataka bez poslužitelja

Trend u nastajanju u sustavima za upravljanje bazama podataka su baze podataka bez poslužitelja, koje nude potpuno upravljani pristup operacijama baze podataka bez poslužitelja. Bez potrebe za postavljanjem, upravljanjem ili skaliranjem poslužitelja baze podataka, programeri se mogu usredotočiti samo na stvaranje aplikacija i korištenje podataka pomoću baza podataka bez poslužitelja. Ovaj revolucionarni pristup, također poznat kao baza podataka kao usluga (DBaaS), uklanja infrastrukturni sloj i omogućuje programerima interakciju s bazama podataka bez poslužitelja. Na slici 10. prikazan je grafički prikaz baze podataka bez poslužitelja.



Slika 8. Grafički prikaz baze podataka bez poslužitelja

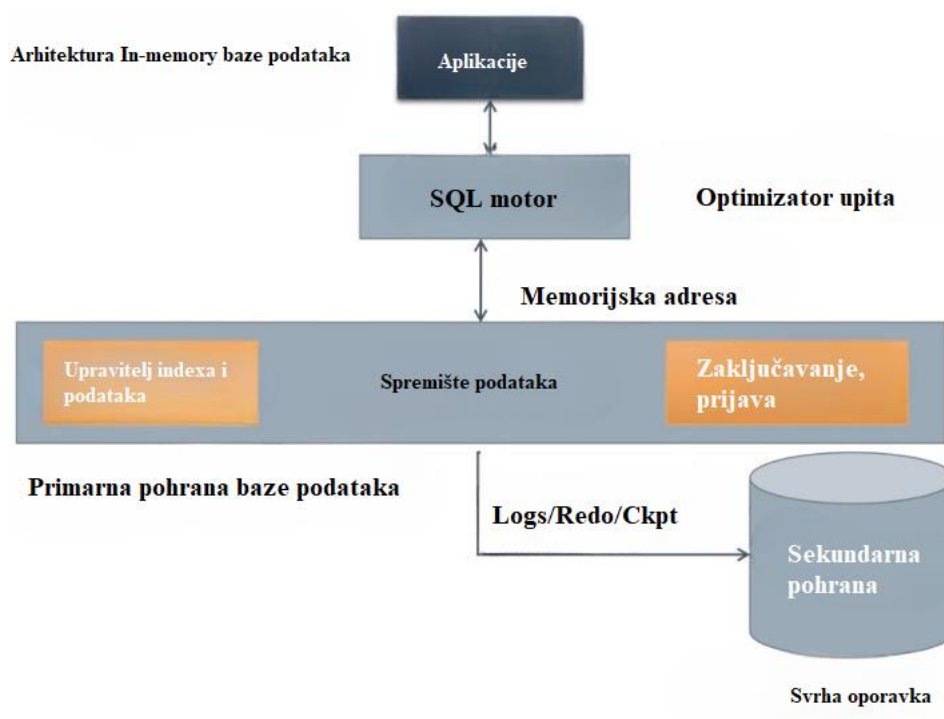
Izvor: <https://www.simform.com/blog/serverless-databases/>

Usporedba baza podataka bez poslužitelja s konvencionalnim samoupravljivim sustavima baza podataka otkriva niz prednosti. Kao prvo, oni oslobađaju programere dužnosti povezanih s infrastrukturom ukidanjem zahtjeva za unaprijed postavljenim i nadgledanjem poslužitelja baze podataka. Programeri se mogu usredotočiti na razvoj aplikacija, povećati produktivnost i kao rezultat toga skratiti vrijeme izlaska na tržište. Dodatno, baze podataka bez poslužitelja nude mogućnost automatskog skaliranja, koje dinamički modificira resurse kako bi

odgovaralo zahtjevima poslovanja. S obzirom da se resursi raspoređuju u skladu sa stvarnom potrošnjom, ova skalabilnost jamči vrhunske performanse i isplativost. Struktura cijene bez poslužitelja jedna je od njegovih glavnih prednosti. Umjesto plaćanja unaprijed određenog kapaciteta, organizacijama se naplaćuje prema stvarnoj potrošnji resursa. Zbog ove fleksibilnosti, tvrtke mogu bolje uskladiti troškove sa zahtjevima aplikacije i riješiti se administrativnog tereta održavanja i poboljšanja infrastrukture baze podataka. Uz visoku dostupnost i toleranciju na pogreške, baze podataka bez poslužitelja također imaju ugrađenu replikaciju i sigurnosne kopije kojima transparentno upravlja pružatelj usluga. Baze podataka bez poslužitelja oslobađaju programere brige o administraciji infrastrukture, skaliranju ili dostupnosti kako bi se mogli usredotočiti na svoju temeljnu logiku aplikacije. Posebno su povoljni za mala i srednja poduzeća, novoosnovana poduzeća i aplikacije s nepredvidivim ili fluktuirajućim radnim opterećenjima budući da pružaju isplativ, visoko skalabilan pristup operacijama baze podataka koji zahtijevaju malo održavanja.

6.4. In-Memory baze podataka

Rješenja za upravljanje bazom podataka koja koriste brzinu i učinkovitost glavne memorije (RAM) za obradu i pohranu podataka poznata su kao baze podataka u memoriji. Baze podataka u memoriji održavaju puni skup podataka u memoriji radi brzog pristupa i izmjene, za razliku od tipičnih baza podataka na disku koje pohranjuju podatke prvenstveno na fizičkim uređajima za pohranu.⁵⁸ Ova metoda iskorištava veliku brzinu memorije i uklanja I/O odgodu diska kako bi pružila značajne prednosti performansi. Brže vrijeme reakcije i povećana propusnost u usporedbi s bazama podataka na disku dvije su glavne prednosti baza podataka u memoriji. Pohrana u stupcima, kompresija podataka i učinkovito indeksiranje podataka uobičajeni su aspekti baza podataka u memoriji, koje su dizajnirane za brzu obradu podataka i dodatno ubrzavaju obradu upita. Slika 11. grafički prikazuje arhitekturu in-memory baze podataka.



Slika 9. Prikaz arhitekture In-memory baze podataka

Izvor: <https://devopedia.org/in-memory-database>

⁵⁸ Plattner, H.: A Common Database Approach for OLTP and OLAP Using an In-Memory Column Database, 2011.

S velikim skupovima podataka lako se rukuje bazama podataka u memoriji, koje često koriste tehnike paralelne obrade i particioniranja podataka kako bi što bolje iskoristile moderna višejezrena računala. Oni također omogućuju metode za osiguranje dostupnosti i trajnosti podataka u slučaju kvarova, kao što su replikacija podataka i sustavi postojanosti. Baze podataka u memoriji SAP HANA, Oracle TimesTen i MemSQL značajni su primjeri. Pružajući mogućnosti transakcijske i analitičke obrade, SAP HANA je in-memory platforma koja omogućuje uvide u goleme količine podataka u stvarnom vremenu. Relacijska baza podataka u memoriji nazvana Oracle TimesTen pruža aplikacijama u stvarnom vremenu obradu transakcija visokih performansi. S unosom podataka u stvarnom vremenu i skalabilnim upitima, MemSQL je distribuirana baza podataka u memoriji koja spaja transakcijska i analitička radna opterećenja.⁵⁹

Organizacije sve više koriste baze podataka u memoriji kao sredstvo za ubrzavanje obrade podataka, omogućavanje analitike u stvarnom vremenu i poboljšanje izvedbe aplikacija. In-memory baze podataka nude snažno rješenje za aplikacije koje zahtijevaju brz pristup podacima i mogućnosti donošenja odluka u stvarnom vremenu budući da mogu upravljati velikim skupovima podataka u memoriji i minimizirati I/O kašnjenje diska.

⁵⁹ <https://www.oracle.com/technetwork/database/options/dbim-vs-sap-hana-2215625.pdf> (29.07.2023)

7. ZAKLJUČAK

Usporedba SQL i NoSQL baza podataka prikazala je prednosti, nedostatke i karakteristične značajke svakog sustava baze podataka. SQL baze podataka i dalje su pouzdana opcija za aplikacije koje trebaju strogu dosljednost, složene podatkovne veze i jasno definirane sheme zbog svog organiziranog dizajna, usklađenosti s ACID-om i zrelog sustava. Izvršni su u situacijama u kojima su kvaliteta i pouzdanost podataka ključni, kao što su financijski sustavi, ERP sustavi i konvencionalne transakcijske aplikacije. S druge strane, NoSQL baze podataka postale su snažna alternativa koja omogućuje prilagodljivost, skalabilnost i agilnost u obradi različitih i nestrukturiranih podataka. NoSQL baze podataka nude distribuirani dizajn, horizontalnu skalabilnost i sposobnost upravljanja golemim količinama podataka. Otkrili su tržište u obradi velikih podataka, platformama društvenih medija i suvremenim online i mobilnim aplikacijama gdje je sposobnost brze prilagodbe promjenjivim strukturama podataka i upravljanje teškim radnim opterećenjima čitanja i pisanja ključna.

Posebni zahtjevi aplikacije u konačnici određuju koju vrstu baze podataka koristiti—SQL ili NoSQL. Prilikom donošenja ovog izbora, organizacije moraju pažljivo uzeti u obzir aspekte uključujući potrebe za podacima, skalabilnost, performanse, jamstva dosljednosti i razvojnu agilnost. Ključno je odvagati kompromise i uskladiti tehnologiju baze podataka sa karakteristikama podataka kao i funkcionalnim i nefunkcionalnim zahtjevima aplikacije. Važno je napomenuti da su hibridni sustavi, koji kombiniraju prednosti NoSQL i SQL baza podataka, također sve popularniji. Ova strategija, također poznata kao poliglotna upornost, uključuje korištenje različitih sustava baza podataka za različite komponente aplikacije prema njihovoj prikladnosti. Maksimiziranjem pohrane i dohvaćanja podataka za određene slučajeve upotrebe unutar jednog sustava, omogućuje tvrtkama da iskoriste prednosti najboljih aspekata i SQL i NoSQL baza podataka.

Razlika između SQL i NoSQL baza podataka može postati puno manja kako tehnologija napreduje. Dok neke NoSQL baze podataka nude transakcijske mogućnosti usklađene s ACID-om za ispunjavanje strožih zahtjeva dosljednosti, neke SQL baze podataka uključuju značajke slične NoSQL-u za povećanje skalabilnosti. Ovi razvoji pokazuju da se

okolina baze podataka uvijek mijenja zbog potražnje za učinkovitim i skalabilnim rješenjima za upravljanje podacima koja se mogu nositi sa kompliciranim podatkovnim scenarijima.

Konačno, usporedba SQL i NoSQL baza podataka naglašava koliko je važno razumjeti jedinstvene zahtjeve aplikacije i svojstva podataka koji se pohranjuju. Organizacije mogu odabrati najbolji sustav baze podataka koji podržava njihove poslovne ciljeve pažljivom procjenom performansi, skalabilnosti, dosljednosti, modeliranja podataka i jednostavnosti upotrebe. Odluku treba donijeti na temelju posebnih zahtjeva aplikacije i podataka koje ona obrađuje jer su i SQL i NoSQL baze podataka vrijedni alati za administraciju baze podataka. Organizacije mogu koristiti snagu ovih tehnologija baze podataka kako bi otkrile puni potencijal svojih podataka, potaknule inovacije i isporučile dojmljive aplikacije u digitalnom svijetu koji se neprestano mijenja razvijanjem temeljitog razumijevanja sličnosti, razlika, prednosti i ograničenja SQL-a i NoSQL-a baze podataka.

LITERATURA

1. Carić T., Buntić M., Uvod u relacijske baze podataka, Zagreb 2015.
2. Groff J., Wienberg P., SQL: The Complete Reference, 1999.
3. Connolly, T. M., & Begg, C. E. (2014). Database Systems: A Practical Approach to Design, Implementation, and Management
4. Chodorow K., MongoDB: The Definitive Guide, 2013.
5. Lakshman, A., Malik, P., Cassandra: A decentralized structured storage system. ACM SIGOPS Operating Systems Review, 2009.
6. Sadalage, P. J., & Fowler, M.: NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, 2012.
7. Hellerstein, J. M., Stonebraker, M., & Hamilton, J.: Architecture of a Database System, 2007.
8. Stonebraker, M.: SQL databases v. NoSQL databases, 2010.
9. Kuno, H., Linskey, D., & Reed, B.: What's Really New with NewSQL? ,2012.
10. Jovanovic, P., & Romero, O.: Multi-model Databases: A Paradigm Shift for Big Data Systems, 2014.
11. Plattner, H.: A Common Database Approach for OLTP and OLAP Using an In-Memory Column Database, 2011.
12. Thi Anh Mai Phan, Kongens Lyngby: Cloud Databases for Internet-of-Things Data, Technical University of Denmark Informatics and Mathematical Modelling, 2013.

INTERNET IZVORI

1. IBM
<https://www.ibm.com/blog/structured-vs-unstructured-data/> Pristupljeno -11.08.2023.
2. Oracle
<https://www.oracle.com/database/what-is-database/> Pristupljeno – 12.05.2023.
3. IBM
<https://www.ibm.com/ibm/history/ibm100/us/en/icons/ibmims/> Pristupljeno – 12.05.2023.
4. IBM
<https://www.ibm.com/topics/relational-databases/> Pristupljeno – 12.08.2023.
5. Arcion
<https://www.arcion.io/learn/database-replication/> Pristupljeno - 11.08.2023.
6. MongoDB
<https://www.mongodb.com/databases/key-value-database/> Pristupljeno – 13.05.2023
7. MongoDB
<https://www.mongodb.com/basics/acid-transactions/> Pristupljeno - 10.08.2023
8. Technopedia
<https://www.techopedia.com/7/31992/storage/what-does-partitioning-mean-in-regards-to-a-database/> Pristupljeno - 11.08.2023.
9. Techtarget
<https://www.techtarget.com/searchoracle/definition/Oracle> Pristupljeno – 13.05.2023
10. Postgre
<https://www.postgresql.org/docs/current/history.html> Pristupljeno – 14.05.2023
11. Redis
<https://redis.io/docs/about/> Pristupljeno – 15.06.2023
12. Atlassian
<https://www.atlassian.com/microservices/microservices-architecture/distributed-architecture/> Pristupljeno - 15.08.2023
13. MySQL
<https://dev.mysql.com/doc/refman/8.0/en/storage-engines.html> Pristupljeno - 16.08.2023

14. MySQL

<https://dev.mysql.com/doc/refman/8.0/en/mysql-nutshell.html> Pristupljeno -
16.08.2023

15. Cloudacademy

<https://cloudacademy.com/blog/amazon-dynamodb-ten-things/> Pristupljeno -
16.06.2023

16. Qlik

<https://www.qlik.com/us/data-replication/database-replication> Pristupljeno -
16.06.2023

17. Quickbase

<https://www.quickbase.com/articles/timeline-of-database-history> Pristupljeno -
20.08.2023

18. DigitalOcean

<https://www.digitalocean.com/community/conceptual-articles/an-introduction-to-document-oriented-databases> Pristupljeno - 19.08.2023

19. TechTarget

<https://www.techtarget.com/searchdatamanagement/definition/SQL> Pristupljeno -
23.08.2023

20. Secureauth

<https://www.secureauth.com/blog/authentication-vs-authorization-vs-encryption-whats-the-difference/> Pristupljeno - 16.08.2023

21. Datastax

<https://www.datastax.com/guides/nosql-use-cases> Pristupljeno - 17.07.2023

22. Oracle

<https://www.oracle.com/erp/erp-vs-crm/> Pristupljeno – 19.08.2023

23. Epam

<https://anywhere.epam.com/business/big-data-analytics-in-retail> Pristupljeno - 19.08.2023

24. Sciencedirect

<https://www.sciencedirect.com/science/article/pii/S1532046420302987>

Pristupljeno - 19.08.2023

25. Ramotion

<https://www.ramotion.com/blog/database-in-web-app-development/> Pristupljeno –

19.08.2023

POPIS TABLICA

| | |
|---|----|
| Tablica 1. Prikaz prednosti kod jednostavnosti korištenja između dva sustava..... | 47 |
| Tablica 2. Usporedba prednosti SQL i NoSQL baza podataka..... | 49 |
| Tablica 3. Usporedba NewSQL-a sa klasičnim SQL-om i NoSQL-om..... | 53 |

POPIS SLIKA

| | |
|--|----|
| Slika 1. Povijest razvitka baza podataka..... | 6 |
| Slika 2. Primjer relacijske baze podataka..... | 9 |
| Slika 3. Primjer objektnog modela podataka..... | 10 |
| Slika 4. Primjer baze podataka temeljene na principu ključ – vrijednost..... | 15 |
| Slika 5. Prikaz MySQL “Workbench”..... | 19 |
| Slika 6. Prikaz NoSQL baze podataka “Cassandra”..... | 28 |
| Slika 7. Prikaz „shardinga“ u MongoDB-u..... | 51 |
| Slika 8. Grafički prikaz baze podataka bez poslužitelja..... | 56 |
| Slika 9. Prikaz arhitekture In-memory baze podataka..... | 58 |

POPIS GRAFIKONA

| | |
|--|----|
| Grafikon 1. Najkorištenije usluge prema ispitanicima stranice Stack Overflow..... | 18 |
| Grafikon 2. Prikaz sposobnosti dohvaćanja podataka raznih sustava (latencija po sekundi).. | 41 |

POPIS KRATICA

| Kratika | Puni naziv na stranom jeziku | Tumačenje na hrvatskom jeziku |
|----------------|---|--|
| ACID | Atomicity, Consistency, Isolation, Durability | Atomičnost, Dosljednost, Izolacija, Trajnost |
| CRM | Customer relationship management | Upravljanje odnosima s klijentima |
| CSV | Comma-Separated Values | Vrijednosti razdvojene zarezom |
| DBMS | Database management system | Sustav upravljanja bazama podataka |
| ERS | Enterprise resource planning | Planiranje resursa poduzeća |
| HRMS | Human resource management | Upravljanje ljudskim resursima |
| IBM | International Business Machines | Međunarodni poslovni strojevi |
| IDS | Intrusion detection system | Sustav za detekciju provala |
| IoT | Internet of Things | Internet stvari |
| JSON | JavaScript Object Notation | JavaScript objektna notacija |
| NoSQL | Not only SQL | Ne samo SQL |
| OODB | Object-oriented databases | Objektno orijentirane baze podataka |

| | | |
|------------|-----------------------------|-----------------------------------|
| ORM | Objectal-relational mapping | Objektno-relacijsko preslikavanje |
| SQL | Structured Query Language | Strukturirani jezik upita |
| XML | Extensible Markup Language | Proširivi jezik za označavanje |