

## КОМПАРАТИВНА АНАЛИЗА ПЕРФОРМАНСИ РЕЛАЦИОНИХ И НЕРЕЛАЦИОНИХ СИСТЕМА БАЗА ПОДАТАКА COMPARATIVE PERFORMANCE ANALYSIS OF RELATIONAL AND NON- RELATIONAL DATABASE SYSTEMS

Миљана Симић, Факултет техничких наука, Нови Сад

**Област – ЕЛЕКТРОТЕХНИЧКО И  
РАЧУНАРСКО ИНЖЕЊЕРСТВО**

**Кратак садржај** – У овом раду описани су основни концепти система релационих и нерелационих база података и перформанси база података. Поређење перформанси база података је извршено у погледу просечног времена извршења упита ка бази података. Описан је коришћени скуп података, изабране релациона (MySQL) и нерелациона (Neo4j) база података су креиране и испуњене подацима. Посебна пажња приликом креирања упита посвећена је класификацији упита у различите категорије ради лакшег поређења перформанси. Упити су анализирани и оптимизовани коришћењем одговарајућих техника. За развој и управљање изабраним базама података коришћени су MySQL и Neo4j алати, а за креирање везе ка базама података и извршавање упита коришћен је програмски језик Python.

**Кључне речи:** системи база података, релациона база података, нерелациона база података, перформансе, упит

**Abstract** – This thesis describes the fundamental concepts of relational and non-relational database systems and their performance. The comparison of database performance was conducted in terms of the average query execution time. The selected dataset was described, and relational (MySQL) and non-relational (Neo4j) databases were created and populated with data. Special attention is given to the classification of queries into different categories for easier performance comparison. Queries are analyzed and optimized using appropriate techniques. MySQL Workbench and Neo4j Desktop tools are used for the development and management of the selected databases. Python is used to establish connections with databases and execute queries.

**Keywords:** database systems, relational database, non-relational database, performance, query

### 1. УВОД

У савременом свету, информационе технологије се константно развијају и постају неизоставне у свакодневном животу и пословању, те је правилно управљање подацима од великог значаја. Базе података представљају темељну инфраструктуру за чување,

организацiju и управљање подацима, а различите врсте база података нуде различите приступе решавању изазова.

У почецима развоја база података, SQL (енгл. *Structured query language*) базе података су доминирале информационим системима и биле су најчешћа опција за чување и управљање подацима. Међутим, у задње време, свет информационих технологија доживео је драстичну трансформацију, што је довело до експлозије великих количина података и појаве полу-структурираних података. NoSQL (енгл. *Not Only SQL*) базе података су стекле огромну популарност и све чешће се примењују.

Циљ истраживања јесте дубље разумевање карактеристика и перформанси SQL и NoSQL база података. У фокусу истраживања јесте анализа и упоређивање перформанси ових двеју база података у смислу времена извршавања упита над истим скупом података. Рад се користи велику количину стварних података прикупљених са Shopify платформе. Циљ је да се пројектним приступом развију релациона (MySQL) и нерелациона (Neo4j) база података, испуне подацима са Shopify платформе, изврше различити упити и упореде перформансе ових база података у одређеним сценаријима.

### 2. ТЕОРИЈСКЕ ОСНОВЕ И КОНЦЕПТИ БАЗА ПОДАТАКА

База података представља темељ сваког информационог система. Без обзира на врсту апликације или област примене, разумевање основних концепата и принципа база података је кључно за ефикасно чување и управљање подацима.

#### 2.1 Релационе и нерелационе базе података

Релациона база података представља врсту базе података која се заснива на релационом моделу података. За разлику од неких комплексних модела, релациони модел је једноставан и интуитиван начин за представљање података. Релационе базе података обезбеђују ACID својства трансакције (атомичност, конзистентност, изолација и трајност), те су трансакције поуздане и обезбеђују конзистентност података [1].

Нерелационе базе података пружају различите опције за организацију података. Нудећи разнолике структуре података, могу примењивати у различите сврхе, као што су анализа података, управљање великим ко-

### НАПОМЕНА:

Овај рад проистекао је из мастер рада чији је ментор био др Милан Челиковић, доцент.

личинама података, друштvene мреже, развој мобилних апликација и друге. У контексту дистрибуиране обраде значајна је CAP теорема у оквиру које се тврди да дистрибуирани системи у датом тренутку времена могу гарантовати само два од следећа три својства: конзистентност, доступност и толеранција на партиционисање. NoSQL базе података често фаворизују доступност и толеранцију на партиционирање у односу на конзистентност у сваком тренутку [2].

Нерелационе базе података најчешће се деле у четири групе: документ, кључ-вредност, графовске и колонске.

Граф база података користи структуре графа за представљање односа између података. Елементи података су дефинисани као чворови, везе и својства [3]. Чворови представљају ентитете. Веза (потег у графу) дефинише однос између чворова, док својства представљају информације које се односе на чворове или везе.

## 2.2 Перформансе базе података

Перформансе базе података се односе на способност базе података да ефикасно обрађује упите и трансакције, као и да пружа висок ниво одзива корисницима.

База података је неизоставан део било ког система који складишти податке и веома је битно да ради коректно, буде доступна, омогућава брзе операције, конзистентност података и буде отпорна на отказе. Чак и релативно мањи проблеми са перформансама базе података могу негативно утицати на функционисање система. и на корисничко искуство.

Управљање и надгледање перформанси осигурава да база података није уско грло (енгл. *bottleneck*) или није та која узрокује проблеме, и да ради што је могуће ефикасније и оптималније [4]. Многи проблеми са перформансама базе података повезани су са перформансама упита. Дobar дизајн упита може смањити оптерећење базе података. Поред упита, индекси су још један кључни елемент базе података. Правилно индексирање може побољшати перформансе базе података јер повећава ефикасност процеса прибављања података. Коришћење кеша, оптимизација шеме базе података и хардвера су још неки од фактора који могу допринети бољим перформансама.

## 3. ПРЕГЛЕД ПОСТОЈЕЋЕ РЕЛЕВАНТНЕ ЛИТЕРАТУРЕ

Истраживањем и анализом постојећих научних радова који се бави сличним проблемима и концептима, издвојена су следећа три као најрелевантнија.

Рад „Поређење перформанси *SQL* и *NoSQL* система за управљање базама података (скраћено СУБП) за конфигурацију 5G базе радио станице“ [5] спроводи истраживање са циљем побољшања постојећег СУБП-а. Циљ је био имплементација и упоређивање две базе података, како би замениле све тренутно сачуване податке у датотекама. Упоређивање укључује време за креирање, читање и ажурирање података у бази података. Као представник база података коришћене

су *PostgreSQL* и *MongoDB* базе података. Креиране базе су упоређиване у погледу времена одговора на уобичајене упите, као и у тестирању оптерећења са очекиваним оптерећењем на базу података. Резултати су показали да обе базе имају своје предности и оба решења испуњавају своју сврху. С обзиром на познавање система и начина на који ће систем бити коришћен, *MongoDB* је изабран као бољи СУБП.

У оквиру истраживачког рада на тему „Перформансе граф и релационе базе података у сложеним упитима“ [6] коришћене су *MariaDB* и две верзије *MySQL* базе података, док је *Neo4j* изабран да буде представник граф СУБП-а. Поред анализе сложених упита, посвећена је пажња факторима који се односе на ефикасност. Резултати граф СУБП-а превазишли су релациони СУБП са структурално једноставним скуповима података и једноставним упитима. Међутим, са сложенијим упитима, нови системи за управљање релационим базама података надмашили су *Neo4j*. Резултати овог истраживања показују како су релационе базе података и даље добра алтернатива када је реч о перформансама. Закључак рада је да, иако је *Neo4j* ефикаснији за једноставне упите, *MariaDB* је доста ефикаснија када се ради о сложеним упитима.

Рад на тему „Поређење перформанси граф и релационе базе података“ [7] је као мотивацију имао поређење релационе и граф базе података и испитивање примене ових база података у различитим случајевима коришћења. Поређење је извршено извршавањем упита над истим подацима у оба шеме базе података. Резултати су показали да граф база података надмашује релациону базу података до 146 пута када су у питању сложени упити и велики скупови података.

## 4. ИМПЛЕМЕНТАЦИЈА ИСТРАЖИВАЊА

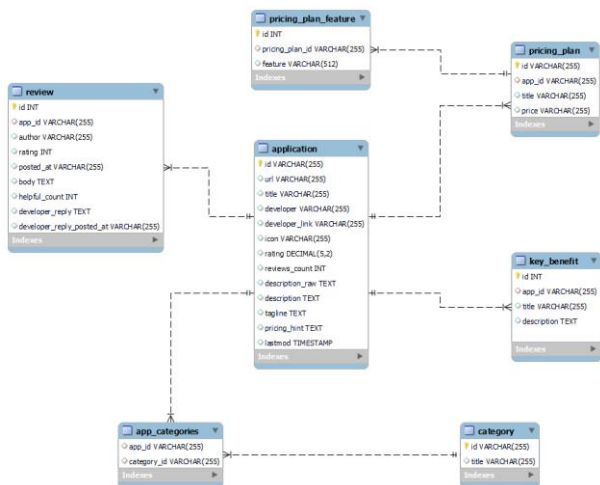
Приступ истраживању је „одоздо према горе“ (енгл. *bottom-up*), што значи да је прво пронађен одговарајући скуп података који ће служити као основа истраживања, а да је потом извршен избор одговарајућих релационих и нерелационих система за управљање базом података.

### 4.1 Скуп података и шеме база података

Извор скупа података је *Shopify App Store* [8], популарна платформа за интернет трговину, а сам скуп података доступан је на *Kaggle* веб страници. Овај скуп података садржи информације о различитим апликацијама које су креирали програмери и које су доступне на платформи, као и о рецензијама које су објавили купци.

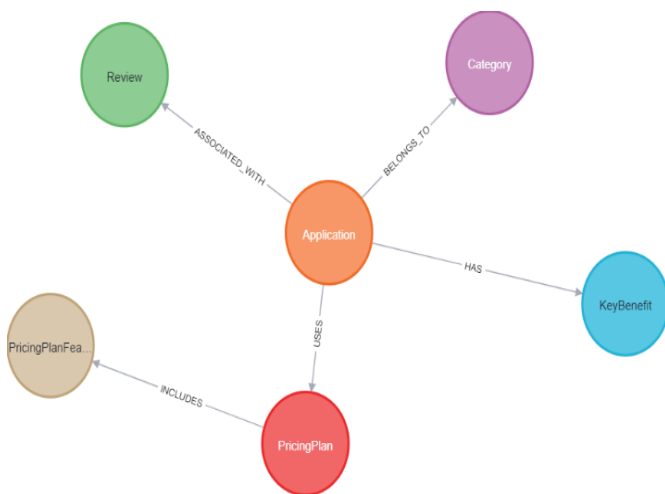
Детаљним истраживањем великог броја скупова података, одабран је овај скуп података у циљу спровођења анализе и поређења перформанси. Одабир скупа података потиче из жеље да се обухвати велики обим података, везе међу подацима и разноврсне информације. У процесу припреме података извршено је пречишћавање података, уклањање неправилности и разлика у кодирању вредности одређених колона како би подаци били конзистентни и употребљиви.

Као релациони СУБП, изабран је *MySQL* сервер базе података је веома брз, поуздан, скалабилан и лак за коришћење. Слика 4.1 приказује визуалну репрезентацију шеме базе података.



Слика 4.1 Шема релационе базе података

Што се тиче нерелационе базе података, изабрана је граф база података, *Neo4j*. *Neo4j* граф база података омогућава детаљно истраживање веза између апликација, категорија, бенефита, рецензија и других ентитета из скупа података. Слика 4.2 приказује визуелну репрезентацију генерисане шеме граф базе података



Слика 4.2 Шема граф базе података

#### 4.2 Извршавање упита

Метрика евалуације перформанси била је просечно време одзива за сваки од креираних упита. Перформансе обе базе података оцењиване су упоређивањем времена одзива. Просечно време је мерено покретањем сваког упита у одређеном броју итерација (50).

За комуникацију са базама података коришћен је *Python* програмски језик. Креиране су *Python* скрипте које се повезују на базу података и шаљу упите ка истој. Прва три захтева ка бази су игнорисана како би се систем за управљање базом података правилно поставио и стабилизовао везу, што значи да је средња

вредност сваког упита израчуната на основу 47 захтева.

Упити су подељени у три категорије:

1. Једноставна селекција: Упити који припадају овој категорији су једноставни упити. Ови упити фокусирани су на прибављање података из једне табеле или типа чвора на основу одређених атрибута.
2. Операције спајања: Упити који припадају овој категорији подразумевају спајање две или више табела, односно чворова када се ради о граф бази података.
3. Агрегација података: Упити који припадају овој категорији обухватају агрегацију података, као што су бројање, сумирање, рачунање просека, максималних и минималних вредности и слично, из више извора података.

Резултати покренутих упита сачувани су у *CSV* датотеке које су касније коришћене за креирање дијаграма и табела са резултатима, ради евалуације и анализе резултата.

#### 4.3. Анализа стратегије упита

Анализа плана извршавања упита доприноси бољем разумевању перформанси упита. *MySQL* пружа веома добру документацију и подршку за оптимизацију и анализу упита. *MySQL* оптимизатор разматра многе технике за ефикасно извршење претрага у *SQL* упиту. Скуп операција које оптимизатор бира како би остварио најефикаснији упит назива се „план извршења упита“, познат као *EXPLAIN* план. Потребно је препознати аспекте *EXPLAIN* плана који указују на то да је упит добро оптимизован, али и искористити технике индексирања ради побољшања плана када се неке од операција не извршавају ефикасно. *EXPLAIN* наредба је извршена над сваким од креираних упита ради побољшања и бољег разумевања перформанси.

*Neo4j* упити се извршавају према одређеном плану извршења. Анализа перформанси упита *Neo4j* базе података изводи се уз помоћ *EXPLAIN* и *PROFILE* команди. Команда *EXPLAIN* се користи за приказивање плана извршења упита, без извршавања упита. Ова команда пружа информације о томе како ће упит бити извршен, укључујући индексе који ће бити коришћени, које операције ће бити извршене и процењен број редова који ће бити обрађен. Команда *PROFILE* извршава упит и пружа детаљније информације од *EXPLAIN* команде.

#### 5. РЕЗУЛТАТИ И ДИСКУСИЈА

Након креираних упита, извршена је анализа и праћење резултата и упити су оптимизовани колико је то могуће.

Резултати упита сачувани су у *CSV* датотекама. Поред графичког приказа резултата извршаваних упита, креирана је и табела са резултатима. Табела је ту ради лакшег и прегледнијег сумирања резултата сваког од упита, као и лакшег поређења просечних времена по категоријама упита.

У наставку се налази табела 5.1 која садржи просечно време извршења упита, по категоријама.

Табела 5.1 Резултати извршења упита

Категорија	Упит	MySQL	Neo4j
Једноставна селекција	Упит 1	26.81ms	38.24ms
	Упит 2	0.58ms	4.14ms
	Упит 3	90.21ms	23.7ms
Операције спајања	Упит 4	0.93ms	4.35ms
	Упит 5	18.32ms	38.5ms
	Упит 6	11.07ms	52.95ms
Агрегација података	Упит 7	20.45ms	68.83ms
	Упит 8	171.49ms	58.63ms
	Упит 9	120.75ms	24.31ms

Кроз евалуацију перформанси, фокус је на три врсте упита – једноставна селекција, операције спајања и агрегација података.

Анализом графика и резултата, простији упити, који селектују податке само једног типа, дали су боље резултате код релационе базе података (и до 7 пута мање просечно време извршења). Са друге стране, упит који садржи под упите у себи се показао боље код граф базе података (3.8 пута брже). Узимајући у обзир резултате, упити спајања дали су убедљиво боље резултате код релационе базе података. На основу резултата агрегационих упита, граф база података се показала доста боље, јер овакви упити захтевају много више времена за извршење над релационим базом података.

## 6. ЗАКЉУЧАК

Циљ рада је био дубље разумевање карактеристика и перформанси релационих и нерелационих база података и издвајање предности и ограничења сваке од ових база података. Истраживање се бавило анализом и упоређивањем перформанси ових двеју база података у смислу времена извршавања упита над истим скупом података. Евалуација је била фокусирана на време одговора, које је приказано просечним временом одговора упита који су покретани у одређеном броју итерација.

Упити који су извршавани над пројектованим базама података класификовани су у три категорије. Резултати извршених упита су показали да релациона података у неким случајевима надмашује граф базу података и до седам пута у одређеним упитима. Ови резултати указују на значај индексирања у релационим базама података. Међутим, у комплекснијим упитима који агрегирају податке, перформансе граф базе података надмашиле су релациону базу података, просечно време извршења упита над граф базом података било је до четири пута мање.

Закључак рада је да се на основу тестирања изабраног скупа података и упита не може опште закључити који од система база података има најбољу ефикасност упита. Приликом избора система база података за одређени домен апликације, потребно је пажљиво и унапред анализирати потребе за будућим упитима, те је често веома корисно комбиновати и интегрисати обе врсте база података у складу са потребама.

Постоји неколико идеја за могућа унапређења и проширења истраживања овог мастер рада. Једна од идеја јесте увести још неке од представника популарних система база података и поредити перформансе између више система. Било би корисно проширити скуп података, користећи још неки од извора података, ради додавања нових ентитета и веза, како би се креирали комплекснији упити. Такође, поређење перформанси је могуће урадити симулацијом стварног оптерећења базе података користећи неки од јавно доступних алата и на тај начин тестирати како ће се понашати сваки од система.

## 7. ЛИТЕРАТУРА

- [1] Oracle, Database topics, <https://www.oracle.com/database/what-is-database/>
- [2] Couchbase, Why NoSQL, <https://www.couchbase.com/resources/why-nosql/>
- [3] Wikipedia, Graph database, [https://en.wikipedia.org/wiki/Graph\\_database](https://en.wikipedia.org/wiki/Graph_database)
- [4] Oracle, Database Performance Tuning Guide
- [5] Alexandra Goltsis, A Performance Comparison of SQL and NoSQL Database Management Systems for 5G Radio Base Station Configuration, Linköping University, 2022, 57 p, <https://www.researchgate.net/>
- [6] Petri Kotiranta, Marko Junkkari, Jyrki Nummenmaa: Performance of Graph and Relational Databases in Complex Queries, Tampere University, 2022, 16 p, <https://www.researchgate.net/>
- [7] Mit Jain, Ashish Khanchandani, Cajetan Rodrigues: Performance Comparison of Graph Database and Relational Database, Computer Science Department San Jose State University, 2023, 19 p, <https://www.researchgate.net/>
- [8] Скуп података Shopify app store, <https://www.kaggle.com/datasets/username3/shopify-app-store>

## Кратка биографија:



**Миљана Симић** рођена је 22. септембра 1999. године у Нишу. Електронски факултет у Нишу, студијски програм електротехника и рачунарство уписала је 2018. године. Дипломирала је 2022. године и исте године уписала мастер академске студије на Факултету техничких наука у Новом Саду.