



ИНТЕГРАЦИЈА РЕЛАЦИОНЕ И ГРАФОВСКЕ БАЗЕ ПОДАТАКА ЗА ПОДРШКУ РАДА ИНФОРМАЦИОНОГ СИСТЕМА ДРУШТВЕНЕ МРЕЖЕ

INTEGRATION OF RELATIONAL AND GRAPH DATABASES TO SUPPORT THE INFORMATION SYSTEM OF THE SOCIAL NETWORK

Тамара Говорчин, Факултет техничких наука, Нови Сад

Област – ПРИМЕЊЕНЕ РАЧУНАРСКЕ НАУКЕ И ИНФОРМАТИКА

Кратак садржај: У овом раду описане су релациона и графовска база података, као и проблеми на које се може наићи приликом коришћења само релационе базе података. Такође су наведени начини на које графовска база може да превазиђе поједине недостатке релационе базе података. На основу свих прикупљених информација, истражен је начин за интеграцију релационе и графовске базе података. У оквиру рада, такође је описан информациони систем друштвене мреже, који представља приказ случаја коришћења решења које је претходно изведено. Циљ изабраног дела информационог система је да олакша процес повезивања корисника у виду изражавања пријатељства у оквиру друштвене мреже, као и да омогући додавање и прегледање фотографија са додатним информацијама. Концептуална шема базе података пројектована је помоћу алата *lucid-app*, а од база података су коришћене *MySQL* као релациона база података и *Neo4j* као графовска база података.

Кључне речи: релациона база података, графовска база података, тип ентитета, друштвена мрежа.

Abstract: This paper describes the relational and graph database, as well as the problems that can be encountered when using only the relational database. The ways in which the graph database can overcome some of the issues of the relational database are also listed. Based on all the collected information, a way to integrate the relational and graph database was investigated. In this paper, the information system of the social network is also described, which represents a use case of the solution previously performed. The goal of this information system is to facilitate the process of connecting users in the form of expressing friendship within the network, as well as to enable adding and viewing photos with additional information. The conceptual database scheme was designed using the *lucid-app* tool, and the databases used were *MySQL* as a relational database and *Neo4j* as a graph database.

Keywords: relational database, graph database, entity type, social network.

НАПОМЕНА:

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

1. УВОД

Развојем информационе технологије уочени су недостаци информационог система заснованих на принципу да свака апликација, односно скуп програма за једну област пословања, има своју колекцију података. Систем база података значајно се разликује од овакве концепције, јер подразумева колекцију података која је заједничка за више апликација. Различити начини коришћења информација довели су до експанзије и креирања разних типова база података. Поред релационих база података, све су више заступљене *NoSQL*, *Document*, *Key-Value*, хијерархијске, графовски оријентисане и друге базе података.

Релационе базе података су најчешће коришћене базе података због погодности које пружају. Међутим, могу да се јаве проблеми уколико се користи само релациона база података, што може да утиче и на саме перформансе система. Неке од слабости релационог модела су ограничена способност да експлицитно искаже семантику захтева и скупа операција рекурзивног повезивања табела (енгл. *JOIN*). Додатно, релациона база података има фиксну шему базе података, што доводи до линеарне зависности броја чворова и веза које се складиште у бази.

Истражен је начин за решавање претходно наведених проблема са којима се може сусрести у току развоја система коришћењем само релационе базе података. Са друге стране, рад се бави истраживањем нерелационе, графовске базе података која може да превазиђе поједине недостатке релационе базе података, с обзиром на то да се структура графовске базе података састоји од чворова и веза, и самим тим је употреба овакве базе података ефикасно решење када је акценат на вези и својствима између чворова. Графовска база података има флексибилну структуру, што може да смањи цену складиштења при великим количинама података.

На основу свих наведених информација, истражен је начин за интеграцију релационе и графовске базе података. Такође је описан информациони систем друштвене мреже, који представља приказ случаја коришћења решења које је претходно изведено. Употребљени су велики скупови података, и за релациону и за графовску базу података. Примарни циљ описаног информационог система, је да пружи могућност корисницима да остварују различите везе са другим корисницима, као и да пружи подршку у процесу додавања нових слика, прегледања профила

корисника, прегледања и измене веза између корисника.

Поред уводног поглавља и закључка, овај рад садржи још шест поглавља, а то су: Преглед актуелног стања у области, Анализа проблема интеграције релационе и графовске базе података, Опис реалног система, Опис шема базе података, Опис архитектуре система и коришћених технологија и Опис апликативног решења.

2. ПРЕГЛЕД АКТУЕЛНОГ СТАЊА У ОБЛАСТИ

У овом поглављу наведени су циљеви коришћења база података, након чега су класификоване и описане базе података. Осим тога представљен је преглед литературе у области којом се бави овај рад. Овај преглед извршен је у циљу бољег разумевања графовске и релационе базе података, као и самог начина функционисања друштвене мреже. Након истраживања јавно доступних радова, детаљније је истражено три рада. Сваки од радова је на свој начин имао значајан утицај у разумевању и дубљем размшљању о начину решавања проблема са којима смо се суочили током израде рада.

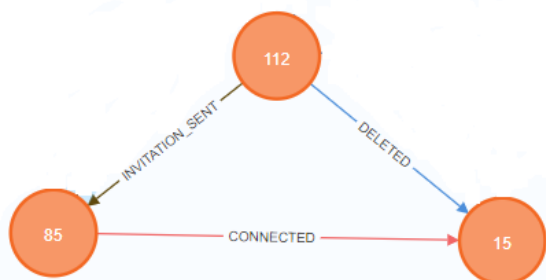
2.1 Класификација база података

База података је систематска колекција података, која подржава електронско складиштење и манипулацију подацима. Базе података олакшавају приступ, управљање и ажурирање подацима. Најчешће коришћене базе података су хијерархијска, мрежна, објектно-оријентисана, релациона и нерелациона (*NoSQL*) база података.

У релационој бази података, подаци се организују у облику дводимензионалних табела, са врстама и колонама. Свака табела би требала да има примарни кључ (енгл. *primary key*). Примарни кључ означава колону, или више колона, која омогућава сваком реду у табели да се јединствено идентификује.

Графовска база података је *NoSQL* база података која се користи за моделовање података у облику графа. Чворови графа приказују ентитете, док релације описују повезаност ових чворова.

Пример графа може се видети на слици 1. У оквиру овог рада, релације могу бити различитог типа, као што су обрисано (енгл. *DELETED*), послат захтев (енгл. *INVITATION_SENT*) и повезано (енгл. *CONNECTED*). Свака релација има смер који је означен стрелицом, а тиме се означава од ког чвора је веза усмерена ка којем чвору. Чвор је представљен са идентификационом ознаком корисника.



Слика 1 – Пример графа у *Neo4j* бази

Уколико би се упоредили релациона база података и графовска база података, однос појмова би изгледао као на слици 2.

Релациона база података	Графовска база података
Табела	Граф
Ред	Чвор
Колона и подаци	Својства са вредностима
Ограничења	Релације
Спајање табела (енг. <i>JOIN</i>)	Специфицирање жељених кретања кроз граф (енг. <i>Traversal</i>)

Слика 2 – Однос појмова између релационе и графовске базе података

2.2 Преглед литературе

У раду [7] обрађени су концепти графовске базе података користећи *Neo4j*. Такође су образложени случајеви у којима је потребно користити графовску базу података.

У раду [8] спроведено је истраживање које пружа упоређивање перформанси и могућности релационе и графовске базе података у складиштењу и обради огромних друштвених података. При повећању количине социјалних података, приказан је однос величина складишта, времена додавања података, као и времена претраге података између *MySQL* и *Neo4j* база података.

У раду [9] описан је начин за постизање флексибилног и ефикасног механизма за складиштење података о метапотацима књиге, у циљу креирања система за препоручивање књига.

3. АНАЛИЗА ПРОБЛЕМА ИНТЕГРАЦИЈЕ РЕЛАЦИОНЕ И ГРАФОВСКЕ БАЗЕ ПОДАТАКА

У оквиру овог поглавља описане су релациона и графовска база података. Након описа ових база података, следи опис конкретних система за управљање базама података. Као систем за управљање релационе базе података описана је *MySQL* [4], док је као систем за управљање графовске базе података описана *Neo4j* [5] база података. Такође је наведен начин на који су *MySQL* [4] и *Neo4j* [5] базе података интегрисане у циљу превазилажења недостатака који постоје када се ове базе података користе самостално, а не у интегрисаној форми.

3.1 Релациона база података

Релациона база података је колекција информација која организује податке у унапред дефинисаним односима где се подаци чувају у једној или више табела. Систем за управљање релационим базама података који је коришћен у оквиру овог рада је *MySQL* [4].

MySQL [4] је брз, скалабилан и једноставан систем за управљање релационим базама података који, између осталог, подржава:

- имплементирање операција базе података на табелама, редовима, колонама и индексима;
- дефинисање базе података у облику табела, што представља колекцију редова и колона.
- декларативну имплементацију ограничења базе података;
- аутоматско ажурирање индекса табела; и
- коришћење SQL упита и комбиновање информације из више табела за крајње кориснике.

3.2 Графовска база података

Графовска база података је нерелациона база података која се користи за моделовање података у облику графа. Чворови графа приказују ентитете, док релације описују повезаност ових чворова. Систем за управљање графовском базом података која је коришћена у оквиру овог рада је *Neo4j* [5].

Предности *Neo4j* су:

- Пружа флексибилан и једноставан модел података.
- Пружа резултате засноване на подацима у реалном времену.
- Доступан је за велике пословне апликације у реалном времену са трансакцијским гаранцијама.
- Могуће је лако представљање повезаних и полуструктурираних података.
- Пружа декларативни језик упита за визуелно представљање графа под називом *Cypher*. Команде овог језика су у читљивом формату и врло једноставне за учење.
- Не захтева сложена спајања за преузимање повезаних података.

У оквиру информационог система који је описан у овом раду, сваки корисник система може да добије на увид листу препоручених корисника. Препоручен корисник се добија тако што се посматрају сви корисници који имају везу типа *CONNECTED* са тренутним корисником, који има идентификациону ознаку 1. Након тога се посматрају корисници који имају везу *CONNECTED* са претходно пронађеним корисницима који имају везу *CONNECTED* са тренутним корисником. Мора се узети у обзир да тренутни корисник не сме да има везу типа *INVITATION_SENT* са пријатељем његовог пријатеља. Овакав упит може се на видети на слици 3.

```
MATCH (userA: User)-[:CONNECTED]-(userB: User)-[:CONNECTED]-(userC: User)
      WHERE userA.id = 1
      AND NOT (userA: User)-[:CONNECTED]-(userC: User)
      AND NOT (userA: User {id: 1})-[:INVITATION_SENT]-(userC: User)
      AND userC.id <> 1
WITH userC, count(*) as score return userC.id as id ORDER BY score DESC LIMIT 20
```

Слика 3 – Упит који враћа листу препоручених пријатеља

3.3 Анализа интеграције MySQL и Neo4j база података

С обзиром на то да је слабост релационе базе података ограничена способност да експлицитно искаже семантику, као и у скупој операцији рекурзивног повезивања табела, неопходно је предложити решење за интеграцију *MySQL* базе са *Neo4j* базом података, како би графовска база података могла да надомести те слабости у појединим моментима. *Neo4j* је систем за управљање графовском базом података, што би значило да је сваки чвор представљен на одређен начин, а такође се на брз начин може креирати велик број веза различитог типа између корисника. Да би ове две базе података могле да се користе у исто време са синхронизованим подацима, неопходно је издвојити поље у односу на који ће се вршити

интеграција ове две базе података. Као такав податак, предлаже се коришћење поља које је дефинисано у *MySQL* бази података, а такође представља и идентификациону ознаку изабране табеле. Изабрано је поље које означава идентификациону ознаку из разлога што такво поље никада неће мењати своју вредност.

Овакав случај коришћења представљен је кроз пример информационог система друштвене мреже. На овом примеру, поље у односу на који ће се вршити интеграција представља идентификациону ознаку типа ентитета *User*.

Сваки корисник ће имати дефинисану идентификациону ознаку у оквиру *MySQL* базе података, док ће при креирању веза између корисника, бити креиран чвор у *Neo4j* графовској бази података који ће означавати корисника, а биће представљен са том идентификационом ознаком.

4. ОПИС РЕАЛНОГ СИСТЕМА

У претходним поглављима анализирани су базе података и наведени су проблеми који се могу јавити приликом коришћења само релационе базе података. Описане су погодности које графовска база података нуди и на који начин може да надомести недостатке релационе базе података.

Након тога је приказано решење за интеграцију релационе и графовске базе података. У даљем тексту биће описан пример употребе решења за интеграцију релационе и графовске базе података на конкретном случају реалног система друштвене мреже.

Опис реалног система односи се на опис класа корисника, њихових карактеристика и надлежности у оквиру информационог система друштвене мреже. У оквиру система друштвене мреже постоје два типа корисника, обичан корисник и администратор система.

Основне корисничке функционалности које могу да извршавају сви корисници система су регистрација корисника, пријава и одјава на систем и ажурирање профила. Сви корисници такође могу да извршавају и сложеније функционалности, које су приказане на дијаграму случајева коришћења на слици 4. На дијаграму се такође може приметити да администратор система наслеђује обичног корисника.



Слика 4 - Дијаграм случајева коришћења за сложене радње сваког корисника

5. ОПИС ШЕМА БАЗЕ ПОДАТАКА

У овом поглављу представљена је прво концептуална шема базе података, а након тога су описани сви типови ентитета и типови повезника у оквиру модела.

5.1. Модел концептуалне шеме базе података

Концептуални модел шеме базе података заснован је на анализи правилника за делатност система за који се пројектује информациони систем, као и на основу постојећих информационих система.

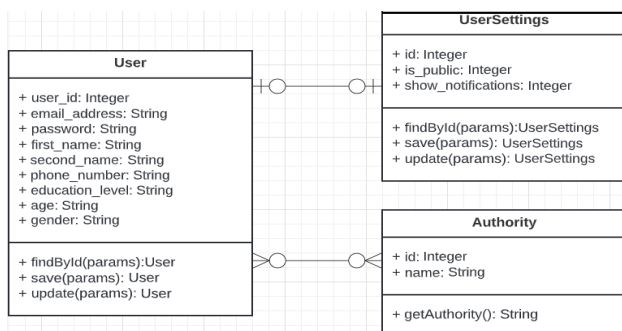
Модел је направљен помоћу алата *lucid-app* [6] и представљен је кроз више делова:

1. део концептуалне шеме базе података за смештање података о корисницима,
2. део концептуалне шеме базе података за смештање података о фотографијама,
3. део концептуалне шеме базе података за смештање података о коментарима,
4. део концептуалне шеме базе за смештање података о реакцијама на фотографијама и
5. део концептуалне шеме базе за смештање корисникових обавештења.

У релациону базу података се смештају подаци који се односе на информације о корисницима, фотографијама, коментарима и подешавања профила, док се у графовску вазу података смештају релације које постоје између корисника.

На слици 5 може се видети део дијаграма концептуалне шеме базе података за смештање података о корисницима. Корисник је представљен са типом ентитета *User* и сваки тип ентитета *User* се везује са нула или више улога корисника представљених типом ентитета *Authority*.

Сваки тип ентитета *User* се везује са тачно једним типом ентитета *UserSettings*, што означава подешавања профила корисника.



Слика 5 - Део дијаграма концептуалне шеме базе података за смештање података о корисницима

На слици 6 приказане су асоцијације типа ентитета *User*.

За сваку асоцијацију наведен је назив асоцијације, референтни ентитет, кардиналитет као и опис саме асоцијације.

Назив асоцијације	Референцирани ентитет	Кардиналитет	Опис асоцијације
Authorities	Authority	*..*	Улоге додељене кориснику
User_settings	UserSettings	1-1	Подешавања налога корисника

Слика 6 - Асоцијације типа ентитета *User*

5.2 Опис типова ентитета и типова повезника у моделу

Након представљања модела, дати су описи обележја у оквиру типова ентитета и типова повезника. Описани су следећи типови ентитета: тип ентитета *Authority*, тип ентитета *UserSettings*, тип ентитета *User*, тип ентитета *Location*, тип ентитета *Image*, тип ентитета *Comment*, тип ентитета *ImageComment*, тип ентитета *Notification* и тип ентитета *Follower*.

На слици 7 се могу видети обележја типа ентитета *User*. За свако обележје се наводи назив, тип податка, да ли сме да буде *null*, да ли је јединствено у оквиру система, као и опис самог обележја.

Редни број	Назив обележја	Тип података	Nullable	Јединствен	Опис обележја
1.	User_id	Integer	Не	Да	Идентификациона ознака
2.	First_name	String	Не	Не	Име корисника
3.	Second_name	String	Не	Не	Презиме корисника
4.	Email_address	String	Не	Да	Е-маил адреса корисника
5.	Password	String	Не	Не	Шифра корисника
6.	Phone_number	String	Да	Не	Број телефона корисника
7.	Age	String	Да	Не	Број година корисника
8.	Gender	String	Да	Не	Пол корисника
9.	Education_level	String	Да	Не	Ниво образовања

Слика 7 - Обележја у типу ентитета *User*

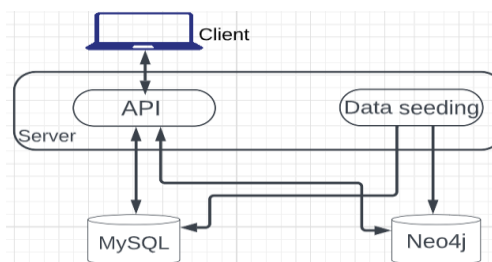
6. ОПИС АРХИТЕКТУРЕ СИСТЕМА И КОРИШЋЕНИХ ТЕХНОЛОГИЈА

С обзиром на то да је апликативно решење замишљено као веб апликација, неопходно је да постоји архитектура система као и технологије помоћу којих је могуће имплементирати овакво решење. У овом поглављу описана је и сликовно приказана архитектура система, а након тога су описане све технологије које су коришћене за израду информационог система кроз апликативно решење.

На слици 8 може се видети приказ архитектуре система апликативног решења за информациони систем друштвене мреже.

Архитектура система се састоји од три дела, а то су: *Client*, *Server* и *Databases*. Клијентска страна, односно кориснички интерфејс, је развијена помоћу *React-a* [3]. На серверској страни постоје два сервиса. Први је *API*, који се односи на инфраструктуру апликативног решења која је развијена помоћу *Spring Boot-a* [2]. Други сервис под називом *Data seeding* представља сервис у оквиру којег се врши рашчлањивање скупова података а након тога и чување података у базе података, а то се врши у оквиру програмског језика *Python* [1].

Апликација користи *MySQL* објектно-релациони систем за управљање базама података, као и *Neo4j* графовску базу података.



Слика 8 – Приказ архитектуре система

7. ОПИС АПЛИКАТИВНОГ РЕШЕЊА

У овом поглављу описано је апликативно решење које обухвата све функционалности описане у претходним поглављима, користећи делове шеме базе података намењене за чување података о корисницима, подешавањима налога, везама између корисника, фотографијама и коментарима. Апликативно решење је замишљено као веб апликација и имплементирано је помоћу технологија које су претходно наведене. Свака функционалност је детаљно описана, наводећи улогу корисника за коју је намењена као и поступак који је неопходно урадити за извршавање исте. Приликом попуњавања форми, наведени су кораци које је неопходно испратити, тако што је наглашено које информације треба унети, као и које дугме кликнути за потврђивање жељене радње.

8. ЗАКЉУЧАК

У овом раду детаљно су истражене и описане употребе релационе и графовски оријентисане базе података, као и начини за њихову интеграцију. Дат је преглед литературе у оквиру које се пореде перформансе релационе и графовске базе података, као и детаљније изучава сама графовска база са системом препоручивања. Такође је описан информациони систем друштвене мреже, који представља приказ случаја коришћења решења које је претходно изведено.

Софтверски пакет за подршку рада друштвене мреже подржава кориснике и администраторе система да манипулишу везама са другим корисницима, да прегледају фотографије, коментаре на фотографијама као и друге информације. Информациони систем доприноси прегледности и доступности података и убрзава процес додавања и брисања фотографија, као и додавања коментара на фотографије. Проналажење нових пријатеља је омогућено помоћу система за препоручивање корисника.

Могуће проширење рада представља увођење додатних типова података и типова релација, чиме би се додатно увидео значај интеграције база података. Додавање подсистема за продају артикала у виду веб продавнице је потенцијално проширење.

9. ЛИТЕРАТУРА

- [1] Документација везана за Python - <https://www.python.org/>
- [2] Документација везана за Spring Framework - https://en.wikipedia.org/wiki/Spring_Framework#Spring_Boot
- [3] Документација везана за React.js - <https://reactjs.org/>
- [4] Документација везана за MySQL базу података - <https://www.mysql.com/>
- [5] Документација везана за Neo4j базу података - <https://neo4j.com/>
- [6] Веб сајт за креирање дијаграма - <https://lucid.app/documents#/dashboard>
- [7] Justin J. Miller - "Graph Database Applications and Concepts with Neo4j", 2013 - <https://asset-pdf.scinapse.io/prod/776871969/776871969.pdf>
- [8] Chen, Yaowen - "Comparison of Graph Databases and Relational Databases When Handling Large-Scale Social Data", 1989 - <https://harvest.usask.ca/handle/10388/7434>
- [9] D. Naga Jyothi - "Book Recommendation System using Neo4j Graph Database", 2020 - <http://www.ijaema.com/gallery/55-ijaema-june-4051.pdf>

Кратка биографија:



Тамара Говорчин рођена је 1999. године у Новом Саду. Факултет техничких наука, смер Рачунарство и аутоматика, уписала је 2017. године. Дипломирала је 2021. године и исте године уписала мастер студије на Факултету техничких наука, смер Рачунарство и аутоматика.