Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/01BE22Savic

GRAF BAZE PODATAKA

GRAPH DATABASE

Ivan Savić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U radu su opisane razlike između graf baze podataka i ostalih tipova baza podataka. Rad je ispratila implementacija softvera u programskom jeziku Java. Zadatak softvera jeste da uspostavi konekciju sa Neo4j graf bazom podataka.*

Ključne reči: *Graf, baza podataka, Java, Neo4j, softver*.

Abstract – *The thesis describe differences between graph database and other types of database. With thesis, there is an implementation of software in Java programming language. Task of the software is to establish a connection with Neo4j graph database.*

Keywords: *Graph, database, Java, Neo4j, software*.

1. UVOD

Baze podataka oduvek su se smatrale jednim od najvažnijih delova sistema, odnosno aplikacija za različite namene. Osobina koja je ključna jeste trajno skladištenje podataka i dobavljanje istih kada je to potrebno.

Danas, najveći udeo na tržištu čine relacione baze podataka. Relaciona baza podataka je poseban tip baze podataka kod kojeg se organizacija podataka zasniva na relacionom modelu.

Podaci se u ovakvim bazama organizuju u skup relacija između kojih se definišu određene veze. Relacija se definiše kao skup n-torki sa istim atributima, definisanih nad istim domenima iz kojih mogu da uzimaju vrednosti. U relacionim bazama podataka, svaka relacija mora da ima definisan primarni ključ, koji predstavlja atribut pomoću kojeg se jedinstveno identifikuje svaka n-torka. Relacija opciono može da poseduje i spoljni ključ, preko kojeg se ostvaruje veza sa drugim relacijama.

Upravljanje ovakvim bazama podataka se realizuje preko sistema za upravljanje relacionim bazama podataka. O karakteristikama relacionih baza podataka, njihovim prednostima i manama, biće reći u narednim poglavljima ovog rada.

Pored pomenutih - relacionih baza podataka, svoje mesto na tržištu pronašle su i tzv. NoSQL (*Not only SQL*) baze podataka koje svojim osobinama i načinom upravljanja podacima nadoknađuju pojedine nedostatke relacionih baza podataka.
U ovom će radu naglasak biti na proučavanju osobina i načinu izvršavanja upita nad spomenutom kategorijom baza podataka, kao i na prednostima i nedostacima prilikom njihovog izvršavanja. Detaljnije će se analizirati načini izvršavanja upita, pre svega, u novije vreme sve popularniji, Cypher Query Language (u nastavku Cypher).
Spomenuti upitni jezici će se uporediti sa upitnim jezikom za relacione baze podataka (u natavku SQL). Tematika rada će biti usredsređena na graf baze podataka. Graf baze podataka predstavljaju poseban oblik NoSQL baza podataka.
U novije vreme ovaj tip baza podataka je sve korišćeniji, i kao takav nama postaje sve zanimljiviji za korišćenje i na nekin način testiranje.

2. RELACIONE BAZE PODATAKA

Relaciona baza podataka se može definisati kao organizovana kolekcija podataka u kojoj su podaci organizovani u skup relacija [1]. Za uvođenje koncepta relacionih baza podataka najvećim delom je zaslužan Edward Codd, koji je ga je prvi spomenuo u svom radu 1970. god. U navedenom radu Edward Codd pojašnjava osobine i mogućnosti relacionog modela koji čini osnovu ove najčešće korišćene vrste baze podataka.

Kao jednu od važnijih osobina tog modela Edward Codd izdvaja činjenicu da su sve informacije u relacionim bazama podataka dostupne i predstavljene kao vrednosti u tabelama. Ova osobina omogućava programerima, ali i krajnim korisnicima, da pristupe podacima preko vrednosti podataka, a ne preko njihove pozicije u sistemu (kao što je bila ranija praksa u domenu baza podataka). Osnovna karakterisitka relacionih baza podataka jeste reprezentacija podataka u obliku dvodimenzionalnih nizova, tj. tabela određena sa n brojem redova i m brojem kolona.

Zbog načina njihovog predstavljanja u praksi mnogi koriste pojam tabela kao zamena za pojam relacija, kolona tabele kao zamena za atribut i red tabele kao zamenu za n-torka.

Dakle, relacione baze podataka svoju dugogodišnju popularnost duguju svojoj jednostavnosti i razumljivosti za osobe s različitim nivoima zanja i sposobnosti.

3. GRAF BAZE PODATAKA

S povećanjem složenosti i povezanosti podataka modeliranje podataka u obliku međusobno povezanih relacija više nije zahvalan zadatak. Zbog toga se danas kao prirodniji način modeliranja podataka sve više nameće modeliranje podataka u obliku grafova koji se sastoje od međusobno povezanih čvorova.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Branko Milosavljević.

U tim grafovima čvorovi (*Nodes*) predstavljaju objekte u stvarnom svetu i okolini, dok veze (*Edges*) predstavljaju veze između tih objekata.

Tehnologija graf baza podataka omogućuje potpuno iskorišćavanje potencijala savremenih podataka u jednostavnijim, ali i složenijim situacijama. Poslednjih godina širok spektar mogućnosti, korisnost i performanse vodećih graf baza podataka su došle na zavidan nivo zrelosti, pa graf baze podataka već i danas zauzimaju značajan udeo na tržištu, a koji konstantno raste. Osim toga, graf baze podataka na važnosti dobijaju i zbog razvoja koncepta Interneta stvari (*Internet of Things*, u nastavku teksta IoT) jer on u svojoj u definiciji uključuje povezivanje uređaja, a time i podataka, što čini osnovu graf baza podataka.

Sistem za upravljanje graf bazama podataka je sistem sa Create, Read, Update i Delete, tzv. CRUD operacijama koje razotkrivaju i deluju nad graf modelom podataka [2]. Prema poslednjim podacima u avgustu 2018. godine sistemi za upravljanje graf bazama podataka zauzimaju 1.3%. Još uvek veliki udeo imaju relacioni DBMS sistemi (77%), pa se informacija o korišćenju graf baza podataka možda čini neznatnom.

Jedan od razloga velikog porasta popularnosti graf baza podataka na tržištu sigurno je činjenica da vrlo uspešno utiču na složene i dinamične veze između čvrsto povezanih podataka [2]. Dakle, graf baze podataka su sve popularnija kategorija NoSQL baza podataka koju je najbolje koristiti za reprezentaciju i pretraživanje povezanih podataka značajnih veličina.

3.1. Definicija grafa

Graf G je uređeni par (V, E). Elementi skupa V se zovu čvorovi (*Vertex*), a elementi skupa E grane (*Edge*) grafa G. Za dati graf G, skup čvorova se označava sa V(G), a skup grana sa E(G). Svaki čvor grafa ima jedinstveni identifikator (ekvivalent primarnom ključu kod relacionih baza podataka) i oznaku (eng. label) koja može biti svojstvena većem broju čvorova u grafu.
U praksi ovako jednostavno definisana struktura grafa omogućava modeliranje podataka iz bilo kojeg domena problema.
Grafovi se mogu iskoristiti za lakše razumevanje različitih oblasti poput nauke, poslovnih aktivnosti, vlade, društvenih mreža, medicine, itd.

3.2. Algebra grafova

Algebra grafova na kojoj se temelji graf model podataka predstavlja proširenje već spomenute relacione algebre na kojoj se temelji istoimeni model podataka. Međutim, postoje značajne razlike između navedenih algebri. Nekim operatorima relacione algebre pridružen je određeni operator u algebri grafova (s razlikama ili bez), ali su u algebri grafova definisani i novi operatori.

3.3. Graf model podataka sa atributima

Graf model podataka sa atributima (*Property graph data model*) čini osnovu graf baza podataka, a sadrži povezane entitete sa atributima navedenih u samoj definiciji grafa:
Čvorovi - predstavljaju entitete u stvarnom svetu, svaki ima svoj identifikator i atriute u obliku parova ključ-vrednost, dok više čvorova može imati istu oznaku (jednu ili više);
Veze - usmerene, imenovane i semantički relevantne veze između dva čvora (entiteta u stvarnosti), svaka ima identifikator, smer, tip, kao i početni i završni čvor (izvorište i odredište), a može imati i atribute (najčešće su kvantitativni poput težine, cene i sl.)

3.4. Obrada podataka u grafu

Procesiranje podataka u graf bazama podataka jedan je od važnijih faktora te tehnologije na koju treba obratiti pažnju. Naime, graf baze podataka koriste tzv. susednost bez indeksa (*Index-free adjacency*) za povezivanje čvorova u grafu.
To znači da je svaki čvor u grafu „fizički“ povezan sa svojim susednim čvorovima, odnosno u sebi sadrži vezu na druge čvorove.

3.5. Pronalaženje uzorka u grafu

Kao što je već spomenuto, popularnost graf baza podataka i algoritama zasnovanih na grafu u novije vreme sve više raste, posebno u naučnom okruženju. Zahvaljujući tome i visokom nivou ekspresivnosti u modeliranju složenih struktura podataka, broj mogućih primena za modeliranje podataka u obliku grafa raste.
Relativno brzo pretraživanje grafa moguće je zahvaljujući algoritmu pronalaženja uzoraka u grafu koji se koristi u izvođenju upita nad graf bazama podataka. Uzorkom u grafu smatra se izomorfna slika tog grafa, što predstavlja podgraf ciljnog grafa, pa se pronalaženje uzoraka u grafu (*graph pattern matching*) često naziva i tzv. problem izomorfizma podgrafa.

3.6. Poređenje graf baza podataka sa drugim bazama podataka

Graf baze podataka moguće je uporediti s relacionim, ali i drugim kategorijama NoSQL baza podataka s obzirom na različite karakteristike. Poređenjem relacionih i svih kategorija NoSQL baza podataka s obzirom na dubinu pretraživanja i veličinu podataka može se uočiti da su graf baze podataka jedini predstavnik NoSQL baza podataka koji može postići veću dubinu pretraživanja podataka (više nivoa) u odnosu na relacione baze podataka.

Međutim, kada se radi o sposobnosti skladištenja, skladišta tipa ključ-vrednost postižu bolje performanse i mogu skladištiti više podataka u odnosu na relacione i graf baze podataka.

4. SISTEMI ZA UPRAVLJANJE GRAF BAZAMA PODATAKA

Prilikom razvoja aplikacije korišćene su različite komponente sistema kako bi se poboljšale neke od funkcionalnosti, kao i omogućio brži i lakši razvoj. Sistemi za upravljanje graf bazama podataka imaju sličnu ulogu kao i relacioni DBMS sistemi opisani u prethodnim poglavljima.

Zastupljenost takvih sistema na tržištu u novije vreme sve više raste. Najčešće korišćeni graf DBMS sistem je Neo4j, čija popularnost konstantno raste. Slede ga OrientDB i Titan koji, zajedno sa Neo4j, ostvaruju najveći porast u korišćenju i zastupljenosti na tržištu. O Neo4j graf DBMS sistemu će biti više reči u nastavku. Osim njega, detaljnije će biti objašnjen i Rexster server korišćen u implementaciji aplikacije.

4.1. Neo4j

Neo4j je sistem otvorenog koda koji skladišti podatke u graf, čime ih je kasnije lakše dobavljati. Zasniva se na mrežno-orjentisanom modelu podataka sa osobinama u kojem su veze najvažniji objekti [4]. Implementiran je pomoću Java i Scala programskih jezika i pripada grupi starijih NoSQL baza podataka.

Neo4j pruža podršku za nativni pristup podacima, ali je podacima moguće pristupiti i putem Cypher i Gremlin upitnog jezika. Standardni jezik za upravljanje podacima u Neo4j bazi podataka je Cypher, pa je Cypher upite moguće definisati i izvršavati putem Neo4j desktop ili web aplikacije.

Zbog svoje robustnosti, skalabilnosti, opcione šeme i visokih performansi, Neo4j baze podataka moguće je koristiti i u velikim korporacijama [5].

4.2. Rexster

Rexster je server za graf baze podataka koji omogućava pristup graf bazi podataka putem HTTP/REST protokola i binarnog protokola RexPro koji omogućava slanje Gremlin skripti na udaljenu instancu Rexster servera [6].

Korišćenje HTTP protokola pruža podršku za svoje metode GET, POST, PUT i DELETE, fleksibilan model podataka koji je moguće proširiti brojnim ekstenzijama te procedure skladištene na serveru za brže izvršavanje Gremlin upita.

5. UPITNI JEZICI ZA GRAF BAZE PODATAKA

Povećanje mogućih oblasti primene graf baza podataka tokom nekoliko poslednjih decenija dovelo je do razvoja nekoliko različitih upitnih jezika za graf baze podataka. Ti upitni jezici su nastajali pod različitim uticajima: hipertekstualnih sistema u 80-im godinama prošlog veka, polustruktuiranih podataka i objektnih baza podataka u 90-im godinama prošlog veka, pa sve do najnovijeg uticaja semantičkog web-a i društvenih mreža [3].

Kao što je već spomenuto u prethodnim poglavljima, prilikom izvođenja upita koriste se različiti algoritmi obilaska grafa koji traže zadani uzorak.

Opšta šema obilaska u svakom od tih algoritama je da obilazak počinje u jednom čvoru iz kojeg se on nastavlja preko veza tog čvora prema ostalim čvorovima grafa. Putem posećenih veza obilazak se nastavlja prema posećenim čvorovima koji imaju neposećene veze i tako dugo dok se ne obiđu sve veze grafa.

5.1. Osnovni algoritmi grafa

Iako postoji puno algoritama obilazaka grafa, u ovom radu će se spomenuti i ukratko objasniti jedni od najpoznatijih i najkorišćenijih algoritama: Pretraga grafa po dubini (*Depth-first search*, u nastvku teksta DFS); Pretraga grafa po širini (*Breadth-first search*, u nastvku teksta BFS); Dijkstra algoritam

5.2. Upiti nad graf bazama podataka

Zbog širenja oblasti primene graf baza podataka tokom vremena, razvijeni su brojni upitni jezici za pretraživanje i upravljanje grafovima s različitim atributima i strukturama.

Povećana heterogenost i veličina podataka skladištenih u graf bazama podataka stvorila je potrebu za nativnim pristupom bazi podataka.

Savremeni koncept upita koji upravljaju grafom na globalnom nivou označava da se ograničenja i pravila nad čvorovima i vezama u grafu mogu definisati istovremeno nad skupom čvorova i veza koji čine traženi objekat, odnosno podgraf posmatranog grafa, za razliku od dosadašnjeg načina definisanja nad svakim pojedinačnim čvorom ili vezom u više iteracija.

6. ZAKLJUČAK

Graf baze podataka dobijaju sve veću pažnju kod korisnika, pre svega zahvaljujući svojim karakteristikama pojašnjenim u prethodnim poglavljima ovog rada.

Njihovim korišćenjem u različitim područjima moguće je skladištiti kompleksne podatke nad kojima se kasnije mogu izvršavati upiti na više nivoa, koji će vraćati rezultate u konačnom vremenu, što za relacione baze podataka nije uvek slučaj.

Njihovim poređenjem može se zaključiti kako je u slučaju povećanja složenosti i količine podataka u bazi podataka standardna devijacija vremena izvođenja upita nad tim podacima kod graf baza podataka mnogo manja u odnosu na relacione baze podataka.

Osim toga, često je skladištenje podataka u obliku grafa mnogo prirodnija nego njihovo skladištenje u redove tabela, jer je graf model podataka fleksibilniji. Neki autori smatraju graf baze podataka relacionim bazama podataka sledeće generacije, budući da su uspele preuzeti i proširiti neke prednosti relacionih baza poput matematičke algebre i razumljivosti modela podataka, ali i popraviti i zameniti njihove uočene nedostatke.

U radu su prikazani osnovni koncepti grafova koji čine model podataka graf baza podataka, ali i pojmovi poput pronalaženja uzoraka u grafu i obilaska grafa koje je potrebno razlikovati. Detaljnije su pojašnjeni najčešće korišćeni algoritmi xpronalaženja uzoraka i obilaska grafa. Ti su algoritmi samo jedan od faktora koji utiču na efikasno izvođenje upita na graf bazama podataka. Upiti nad graf bazama podataka mogu se izvoditi putem nekoliko upitnih jezika, od kojih su danas najzastupljeniji Cypher i Gremlin.

Poređenjem njihovih performansi u radu je pokazano kako Cypher ostvaruje bolje performanse u dobavljanju podataka, ali se ta razlika smanjuje s povećanjem složenosti podataka koje je potrebno dobaviti, jer ih Cypher nakon dobavljanja mora konvertovati u traženi oblik.

Osim teorijskog, ovaj radi sadrži i praktični dio u kojem je napravljena aplikacija za rad s Neo4j bazom podataka te izvođenje Cypher, upitima nad njom. Svrha aplikacije bila je pokazati da je moguće napraviti jednostavnu web aplikaciju, koja će, pomoću određenih bibliotekam, na elegantan način biti povezana sa Neo4j bazom podataka. Ovim je pokazano da nije potrebno mnogo truda oko učenja Cypher jezika i njegove sintakse, ali isto tako je bitno poznavanje njegovog izvršavanja.

Na osnovu analiziranih osobina graf baza podataka može se zaključiti kako će njihova zastupljenost na tržištu i u budućnosti nastaviti svoj rast zahvaljujući stalnom ulaganju u njhov razvoj, ali i povećanoj potrebi upravljanja složenim podacima.

7. LITERATURA

[1] Darwen H. (2012) An Introduction to Relational Database Theory. Peterlee, UK. Ventus Publishing ApS

[2] Robinson I., Webber J., Eifrem E. (2015) Graph Databases (2.izd) Neo Technology Inc. Sebastopol: O'Reilly Media Inc.

[3] Wood P.T. (2012) Query Languages for Graph Databases (1.izd) . London: Department of Computer Science and Information Systems. http:// users.dcc.uchile.cl/~pbarcelo/wood.pdf

[4] Angles R. (2012) A Comparison of Current Graph Database Models. http:// campuscurico.utalca.cl/~rangles/files/gdm2012.pdf

[5] Neo4j - Relational Databases vs Graph Databases: A Comparison - http:// neo4j.com/developer/graph-db-vs-rdbms/

[6] Rodriguez M.A. (2015) The Benefits of the Gremlin Graph Traversal Machine. http://www.datastax.com/dev/blog/the-benefits-of-the-gremlin-graph-traversal- machine

Kratka biografija

Ivan Savić je rođen 07.07.1994. godine u Novom Sadu. Osnovnu školu „Branko Radičević“ završio je 2009. godine u Šidu. Tehničku školu „Nikola Tesla“ u Šidu završio je 2013. godine kao Đak generacije. Iste godine upisao se na Fakultet tehničkih nauka u Novom Sadu, na odsek Računarstvo i automatika. Nakon uspešno završenih osnovnih akademskih studija na smeru Računarske nauke i informatika postaje diplomirani inženjer 2017. godine. Iste godine upisuje master akademske studije na istoimenom fakultetu i smeru i polaže sve ispite predviđene planom i programom 2018. godine.