NoSQL graf baza podataka: od domena do modela preko upita

Olivera Janković
"ORAO" a.d.
Bijeljina, BiH
janolja@yahoo.com

Sadržaj — **Izazovi modelovanja aktuelnih karakteristika visoko povezanih podataka, adresirani su od strane sve popularnijih NoSQL graf baza podataka. Intuitivan pristup problemu modelovanja podataka, potkrepljen kroz određene ilustrativne primjere, korištenjem Neo4j NoSQL graf baze podataka, biće prikazan u ovome radu.**

Ključne riječi –NoSQL graf baze podataka; Neo4j;

I. UVOD

NoSQL (Not only SQL) baze su svojevrstan vid odgovora na važne i permanentne izazove modernog vremena u kontekstu izraženog volumena i složenosti podataka [1], [2]. One su tu da ponude vrijedna rješenja, kroz posebne modele podataka u svjetlu rješenja pomenutih dimenzija.

Graf baze podataka su jedan od četiri elementa NoSQL prostora čija specifična namjena je skladištenje graf orijentisanih struktura podataka [3], [4]. Dakle, ove baze podataka su dobra forma istraživanja podataka koji su graf struktuirani (npr. stablo) posebno kada su izraženi i važni odnosi između pojedinih elemenata. Samo modeliranje podataka korištenjem grafova predstavlja prirodan način izražavanja i u osnovi je čitljiv i onima koji se ne bave tehnikom. U prostoru baza podataka graf baze podataka najčešće adresiraju neku od sledećih mogućnosti:

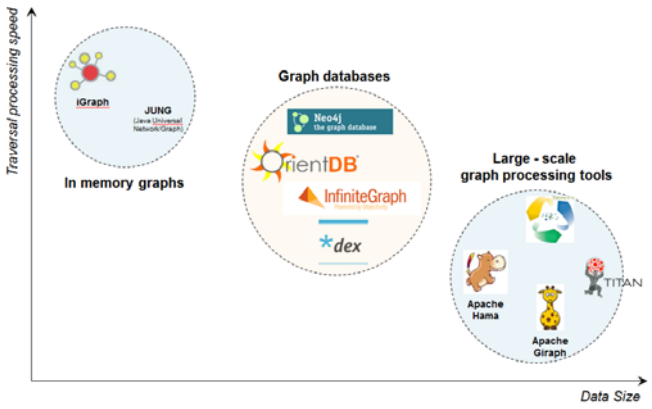
- procesiranje visoko povezanih podataka,
- omogućavaju lako upravljanje složenim i fleksibilnim modelom podataka,
- nude izuzetne performanse koristeći prelaženje grafa[1] (*traversal*).

Generalno ne postoji jedinstven odgovor na pitanje koju bazu podataka izabrati. Ono što može biti jedna od smjernica na putu izbora je i odgovor na pitanje kako će se podaci primarno koristiti. Ako sistem zahtijeva da pretražuje podatke kao hijerarhijsku strukturu, stablo ili graf, odnosno ako je u pitanju složen i visoko povezan model, tada je koncept grafa vjerovatno najadekvatniji izbor. Sledeća važna smjernica je i način obrade podataka. Ako je u pitanju potreba pristupa podacima u realnom vremenu i volumen nije previsok[2] graf baze podataka bi takođe mogle biti dobar kandidat.

Grafovi su dakle vrlo atraktivni kada je u pitanju modeliranje realnog svijeta podataka, jer su intuitivni i zbog toga što iza njih stoji teorija koja je sazrijevala vjekovima. Kao posljedica toga, na raspolaganju su sve popularnije NoSQL graf baze podataka, pri čemu je Neo4j graf baza jedna od najpopularnijih [5]. U tom kontekstu u ovom radu biće prikazan intuitivan način procesa modelovanja podataka korištenjem grafova iz ugla Neo4j graf baze podataka.

II. NOSQL GRAF BAZE PODATAKA

Graf baze podataka su NoSQL sistemi baza podataka koji koriste graf model podataka za skladištenje i obradu podataka. Napore modeliranja podataka prilikom upotrebe graf baza podataka prate različite paradigme od onih koje obično imaju modeli podataka memorisani u relacionim ili drugim NoSQL bazama podataka (kao što su: dokument orijentisane baze podataka, ključ-vrijednost baze podataka ili kolona bazirane baze podataka). Model graf podataka [6] može da se koristi za kreiranje bogatih i visoko povezanih podataka da predstavljaju reprezente korištenja i aplikacije realnog svijeta.



Slika 1. Pozicija graf baza (u kontekstu veličine podataka i brzine procesa prelaženja grafa) [4]

[1] Prelaženje spojenih zapisa kroz veze, eng. traversal. Prelaženje grafa znači posjetu njegovih čvorova, sledeći veze u skladu sa određenim pravilima.

[2] Problemi nastaju prilikom obrade veoma velikih grafova, prilikom posjete milijarde visoko povezanih tjemena.

Na Sl.1 je prikazana pozicija graf baza podataka u odnosu na veličinu i brzinu procesa prelaženja grafa i u tom kontekstu i mjesto Neo4j baze podataka, koja pored ostalog ima implementiran i graf upitni jezik Cypher.

III. GRAF UPITNI JEZIK CHYPER

Cypher je deklarativni graf upitni jezik koji omogućava ekspresivne i efikasne upite i ažuriranje graf skladišta. U biti Cypher je relativno jednostavan, ali ipak vrlo moćan jezik, dizajniran sa ciljem da postane upitni jezik korišten kako od strane razvojnih programera tako i operativnih stručnjaka, tako da jednostavne stvari učini lakim a složene mogućim. Ovaj upitni jezik je temeljen na ustaljenoj praksi za izražajne, ekspresivne upite (*expressive query*). Mnoge od ključnih riječi, kao što su WHERE i ORDERED BY su inspirisane sintaksom SQL upitnog jezika, te se može reći da je Cypher preuzeo svoju strukturu iz SQL-a - upiti su izgrađeni korištenjem različitih klauzula koje su lančano vezane i koje koriste međusobne rezultate. Najčešći način dohvaćanja podataka nekog grafa podrazumjeva upotrebu klauzule MATCH (pronalazi čvorove) i klauzule RETURN koja specificira šta treba da se vrati tim upitom. Upotreba Chyper upitnog jezika nalazi se u svim ilustrovanim primjerima koji su sastavni dio sledećeg poglavlja.

IV. NEO4J

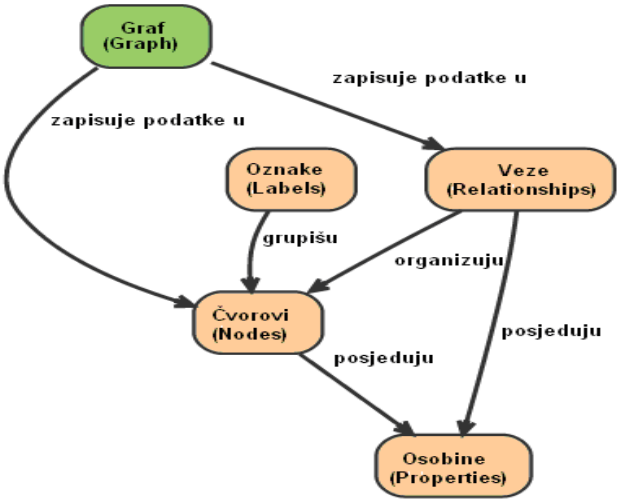
Neo4j je graf baza podataka otvorenog koda, implementirana u Javi, razvijena od strane Neo Technology, koja se zahvaljujući svojim karakteristikama (stabilnost, izdržljivost i brzina, izuzetna skalabilnost i visoka dostupnost, ekspresivnost,...) svrstava u najpopularnije graf baze podataka [7], [8].

A. *Bazni elementi*

Neo4j graf baza podatka može da skladišti podatke koristeći koncept nekoliko jednostavnih pojmova:

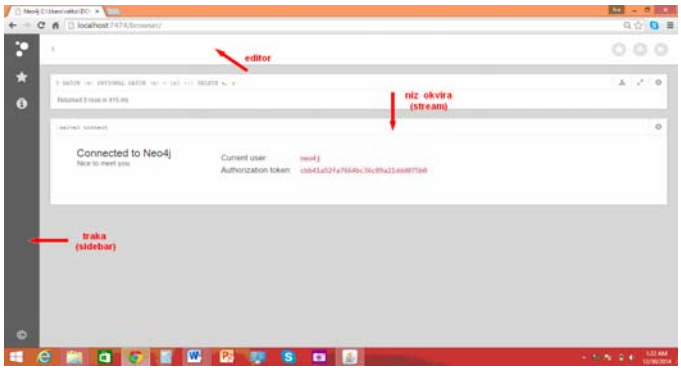
- Čvorovi (*Nodes*) – zapisi graf podataka,
- Veze (*Relationships*) – povezuju čvorove,
- Osobine (*Properties*) – imenovane vrijednosti podataka.

Na Sl. 2 se može vidjeti „imovina“ grafa. Fundamentalne jedinke koje čine graf su svakako čvorovi i veze. Čvorovi su sinonim za zapise (slogove) podataka u grafu pri čemu se podaci skladište kao osobine čvora koje se predstavljaju korištenjem jednostavnih parova ime i vrijednost. Često se koriste da predstave entitete, a zavisno od domena i veze (*relationships*) se mogu koristiti u tu svrhu. Čvorovi mogu da se grupišu zajedno korištenjem oznaka, tj. labela (labele nemaju svojstva) koje asociraju na skup čvorova Labela je imenovana graf konstrukcija koja se dakle koristi za grupisanje čvorova u skupove. Svi čvorovi označeni istom labelom pripadaju istom skupu. Mnogi upiti nad bazom podataka, umjesto nad cijelim grafom, mogu se raditi nad ovim setovima čineći upit efikasnijim ali i jednostavnijim za kreiranje. Čvor može biti označen sa nijednom ili više proizvoljnih labela, čineći tako labele opcionom karakteristikom. Prava snaga Neo4j je u povezanim podacima. U kontekstu grafa veze su zapisi podataka koji mogu imati vlastita svojstva pri čemu veze između čvorova uvijek imaju smjer i tip.



Slika 2. Neo4j graf koncept [8]

Neo4j Browser je komandama vođen klijent, slično kao veb bazirano šel (*shell*) okruženje. Koristan je za izvršavanje ad-hoc upita nad grafovima sa mogućnošću izrade prototipa Neo4j bazirane aplikacije.



Slika 3. Neo4j Browser - osnovni elementi

Osnovni elementi Neo4j (korištena je Neo4j server aplikacija, verzija 2.2.0-M02) programerskog interfejsa dostupnog korištenjem veb čitača (http://localhost:7474/browser/ - prethodno je potrebno startati server pokretanjem Neo4j.bat fajla) dati su na Sl.3. Editor je primarni interfejs za unos i izvršavanje komandi. U njemu se unose Cypher upiti za rad sa graf podacima. Okvir rezultata se kreira za svaku izvršenu komandu (unesenu putem editora) koji se dodaje na vrh niza okvira (*stream*) čineći skrolabilnu kolekciju rezultata upita u suprotnom hronološkom redoslijedu. Lijevo se nalazi proširiva traka (*sidebar*) koja omogućava različite funkcionalne panele za upite i osnovne informacije (metapodaci baze podataka).

B. *Od domena do modela preko upita*

Simboličan slogan koji predstavlja ovaj podnaslov, opisuje na pojednostavljen način proces dizajna graf modela Neo4j

baze podataka, koji u osnovi može biti odgovor na širok rang poslovnih pitanja koja se protežu kroz razne domene.

Modelovanje podataka je proces u kome Neo4j korisnik opisuje proizvoljan domen kao povezan graf sastavljen od čvorova i veza. Iz pomenutog opisa domena graf model podataka je dizajniran sa ciljem da da odgovore na određena pitanja, u formi Cypher upita.

Recimo da je opis domena interakcije određenih osoba i knjiga, za početak dat sa sledećom rečenicom: „Tri osobe, Olivera, Milan i Ratko su čitale knjige: Kuća sećanja, Kainov ožiljak i Martinina velika zagonetna avantura."

Ova rečenica je dovoljna da se na osnovu njenog konteksta mogu identifikovati već pomenute komponente, kao što su labele, čvorovi i veze i na osnovu toga izgradi adekvatan model. Čvorovi se mogu prepoznati kao entiteti sa jedinstvenim konceptualnim identitetom, tako da bi čvorovi u ovom slučaju bili:

- Olivera, Milan i Ratko;
- Kuća sećanja, Kainov ožiljak i Martinina velika zagonetna avantura.

Nakon identifikacije čvorova, može se odrediti koju labelu (ako ih ima) dodijeliti odabranim čvorovima. Identifikovanjem uloge objekata (Olivera,..., Kuća sećanja,...) pomenutih u opisnoj rečenici mogu se identifikovati dva tipa objekata, osobe i knjige, čime se mogu izdvojiti i dvije labele:

- Osoba
- Knjiga

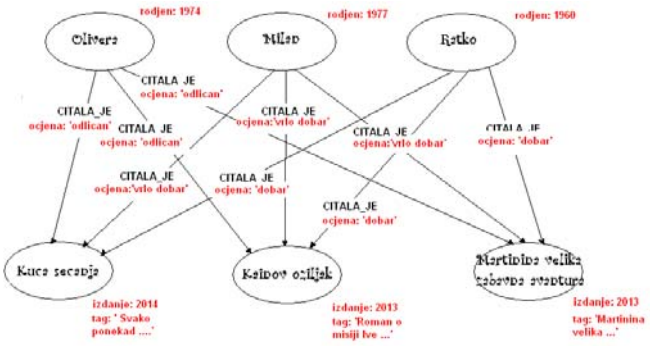
i nakon toga navedene labele pridružiti čvorovima. Naravno čvorovi (Olivera, Milan i Ratko) imaju labelu Osoba a čvorovi Kuća sećanja, Kainov ožiljak i Martinina velika zagonetna avantura) imaju labelu Knjiga (vidljivo u kodu na Sl.5).

Nadalje se može prepoznati interakcija, međusobno djelovanje između objekata (npr. Olivera je čitala knjigu...), tako da nakon što su opisane vrste stvari datog domena, kao čvorovi sa oznakama, možemo povezati čvorove zajedno i opisati njihove interakcije. Za početak, iz date rečenice jasno je da parovi čvorova sa oznakom osoba i knjiga mogu biti međusobno povezani CITALA JE vezom, čime se ujedno prošlo kroz proces stvaranja osnovnog modela podataka za navedenu interakciju između osoba i knjiga. Ovim su ispunjeni potrebni uslovi da se za početak skicira model (dio označen crnom bojom) u formi prikazanoj na Sl.4.

Početni model podataka može se koristiti dalje, definišući osobine entiteta kao ključ-vrijednost osobine. U određivanju novih osobina može poslužiti navođenje pitanja podacima, na koja se u biti očekuje odgovor, kao što su:

- Koliko je star čitalac knjige, osoba sa imenom Milan?
- Kako se dopala knjiga osobi imena Ratko?
- O čemu nam govori knjiga „Kainov ožiljak"?

Ovo su samo neka od pitanja, dovoljna da se identifikuju novi atributi (na Sl. 4 označeni crvenom bojom), nove osobine koje treba da pripadaju entitetima modela podataka.



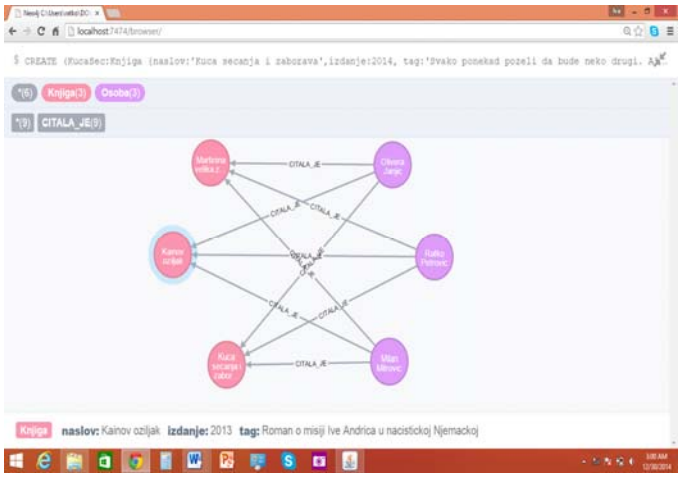
Slika 4. Skica modela domena interakcije osoba-knjiga

Nakon što se postigne određeni model podataka za opisani domen, koji u traženoj mjeri odgovara na postavljena pitanja, može se kreirati skup uzoraka podataka pomoću Cipher programskog koda prikazanog na Sl. 5.

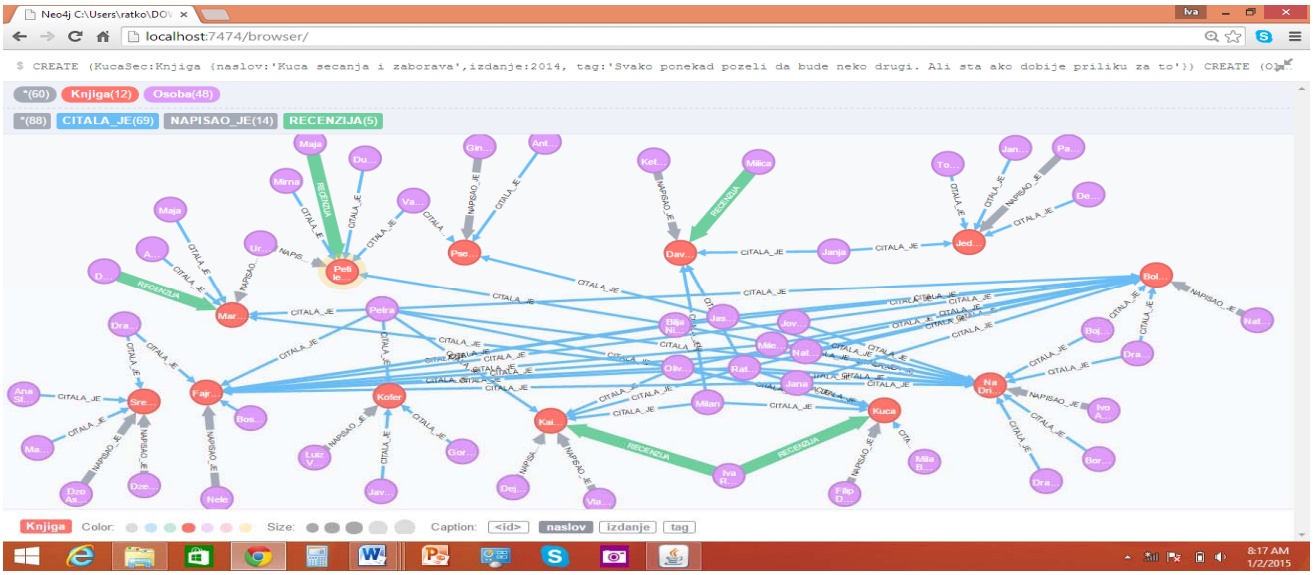
```
CREATE (KucaSec:Knjiga {naslov:'Kuca secanja i
zaborava',izdanje:2014, tag:'Svako ponekad pozeli da bude neko
drugi. Ali sta ako dobije priliku za to'})
CREATE (KainovO:Knjiga {naslov:'Kainov oziljak',izdanje:2013,
tag:'Roman o misiji Ive Andrica u nacistickoj Njemackoj'})
CREATE (MartininaV:Knjiga {naslov:'Martinina velika zagonetna
avantura',izdanje:2013, tag:'Pridruzite se Marti u velikoj
zagonetnoj avanturi'})
CREATE (Olivera:Osoba {ime:'Olivera Janjic', rodjen:1974})
CREATE (Milan:Osoba {ime:'Milan Mitrovic', rodjen:1977})
CREATE (Ratko:Osoba {ime:'Ratko Petrovic', rodjen:1961})
CREATE
(Olivera)-[:CITALA_JE {ocjena:['odlican']}]->(KucaSec),
(Milan)-[:CITALA_JE {ocjena:['vrlo dobar']}]->(KucaSec),
(Ratko)-[:CITALA_JE {ocjena:['dobar']}]->(KucaSec),
(Olivera)-[:CITALA_JE {ocjena:['odlican']}]->( KainovO),
(Milan)-[:CITALA_JE {ocjena:['vrlo dobar']}]->( KainovO),
(Ratko)-[:CITALA_JE {ocjena:['dobar']}]->( KainovO),
  (Olivera)-[:CITALA_JE {ocjena:['odlican']}]->( MartininaV),
  (Milan)-[:CITALA_JE {ocjena:['vrlo dobar']}]->( MartininaV),
  (Ratko)-[:CITALA_JE {ocjena:['dobar']}]->( MartininaV)
WITH *
MATCH (n) RETURN n LIMIT 100
```

Slika 5. Cypher kod za kreiranje grafa

Nakon izvršenog upita slijedi graf model prikazan na Sl.6.



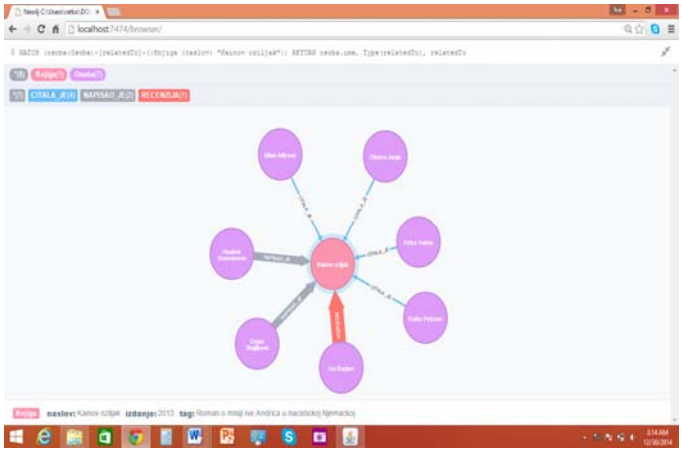
Slika 6. Graf modela domena interakcije osoba-knjiga



Slika 7. Prikaz grafa nakon dorade modela i povećanja uzorka podataka

Osnovni model (sa Sl. 6) se može dalje dorađivati, odgovaranjem na određeni broj dodatnih pitanja koja su naravno u uskoj sprezi sa zahtjevima i odgovorima koje trebamo i očekujemo (Ko je napisao knjigu?, Ko je recenzent?...) od modela podataka, što je rezultiralo i određenim povećanjem uzorka skupa podataka, i tako dobiti graf kao što je ovaj prikazan na Sl.7.

Postojeći uzorak skupa podataka navedenog graf modela, može poslužiti za prikaz pretraživanja grafa, za ilustrativnu realizaciju dobijanja konkretnih rezultata, prevođenjem određenih pitanja u formu upita. U nastavku će biti prikazani neki karakteristični upiti i vizuelizacija dobijenih rezultata.

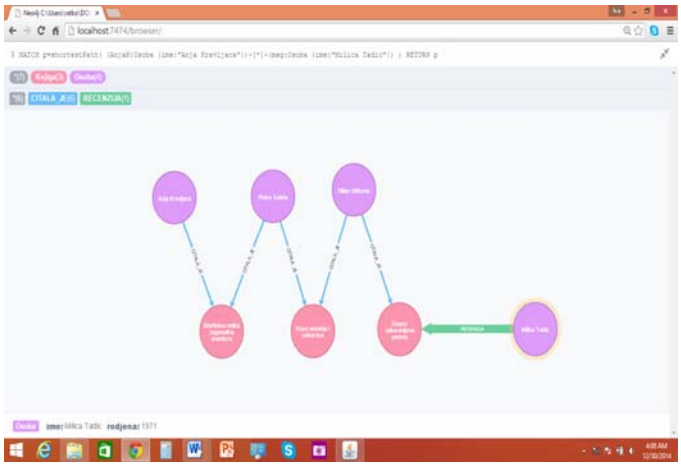


Slika 8. Graf - prikaz relacija sa romanom "Kainov ožiljak"

Na pitanje „Kako se osobe odnose prema romanu „Kainov ožiljak?". odgovarajuća forma upita data je sa:

```
MATCH (osoba:Osoba)-[relatedTo]-(:Knjiga {naslov: "Kainov oziljak"})
RETURN osoba.ime, Type(relatedTo), relatedTo
```

a vizuelni prikaz dat je na Sl. 8, gdje se može jasno vidjeti da za roman „Kainov ožiljak" postoje dva pisca i jedan recenzent, te da su ga pročitale 4 osobe.

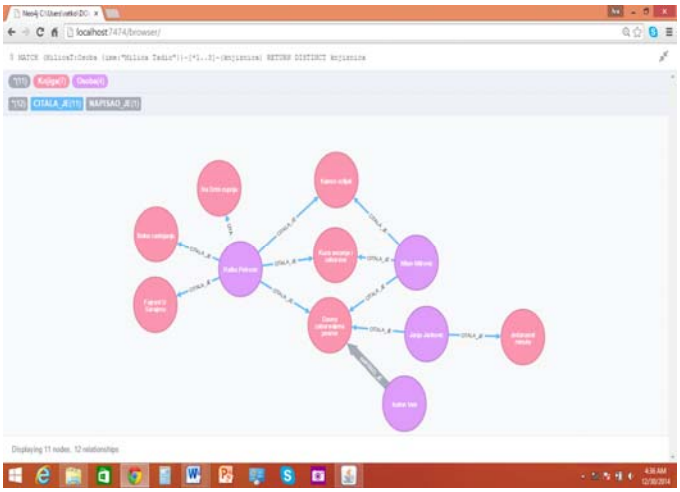


Slika 9. Prikaz najkraćeg puta između osoba Anja Kravljača i Milica Tadić

Primjerima koji slijede pokazan je način pronalaženja zanimljivih konstelacija tzv. puteva[3] (*paths*). Često je potrebno ograničiti dužinu putanje, inače dolazi do kompletnog (iscrpnog ?) pretraživanja grafa. Pronalaženje najkraćeg puta između dva čvora moguće je realizovati korištenjem shortestPath funkcije. Vizuelizacija jednog takvog upita data je na Sl. 9 a sam upit za pronalaženje najkraćeg puta od Anje Kravljaca do Milice Tadić po bilo kojoj vezi glasi:

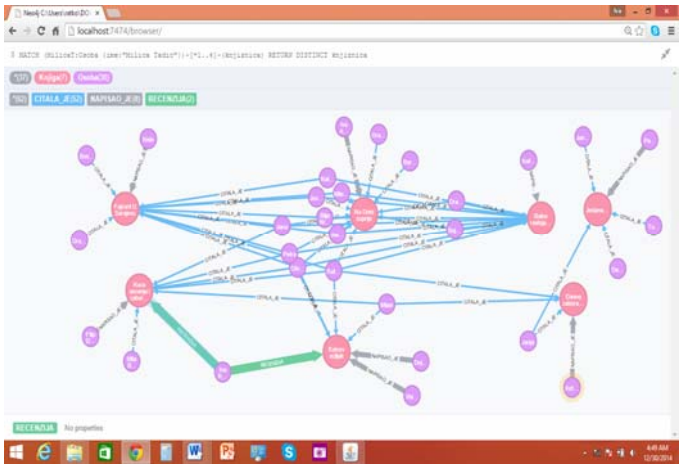
```
MATCH p=shortestPath(
 (AnjaK:Osoba {ime:"Anja Kravljaca"})-[*]-(meg:Osoba {ime:"Milica Tadic"})
)
RETURN p
```

[3] Put je sekvenca čvorova i veza od početnog čvora do krajnjeg čvora.



Slika 10. Osobe i knjige najviše 3 skoka udaljene od zadatog čvora (Milica Tadic)

Moguće je takođe pronalaženja čvorova sa varijabilnim, zadanim brojem skokova (*hops*) u odnosu na određeni čvor, odnosno čvorovi koji su određen broj veza→čvor skoka daleko. Na Sl. 10 i Sl. 11 su prikazani čvorovi koji se nalaze najviše 3 odnosno 4 skoka respektivno (vidljiva je promjena broja rezultujućih čvorova promjenom vrijednosti skoka).



Slika 11. Osobe i knjige najviše 4 skoka udaljene od zadatog čvora (Milica Tadic)

U prethodno ilustrovanim primjerima, korišteni upit koji pronalazi sve osobe i knjige koje se nalaze najviše 4 skoka (1 ili 2 ili 3 ili 4) u odnosu na Milicu Tadić glasi:

```
MATCH (MilicaT:Osoba {ime:"Milica Tadic"}) - [*1..4]-
(knjiznica)

RETURN DISTINCT knjiznica
```

V. ZAKLJUČAK

Izbor baze podataka svakako je jedna od ključnih projektantskih odluka koja se najčešće donosi na početku projekta. Pojava novih modela baza podataka kao što su NoSQL baze podataka može taj izbor učiniti težim ali i potencijalno rješenje svrsishodnijim.

U radu je na intuitivan i ilustrativan način predstavljen proces dizajna graf modela Neo4j baze podataka, koji u osnovi može biti odgovor na širok rang poslovnih pitanja koja se protežu kroz razne domene. Moćan i fleksibilan model omogućava predstavljanje domena realnog svijeta i promjenljivo strukturiranih informacija bez gubitka na njihovoj vjernosti, pri čemu je graf model lako razumjeti i njime rukovati.

LITERATURA

[1] P.J. Saldage, M.Flower, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison-Wesley Professional, 2012.

[2] M. Fowler, "NoSQL Definition", 2012., http://martinfowler.com/bliki/NosqlDefinition.html

[3] I. Robinson, J. Webber, E. Eifrem, "Graph Databases", Neo Technology, O'Reilly Media, 2013.

[4] M. Domenjoud, "Graph databases: an overview", 2012., http://blog.octo.com/en/graph-databases-an-overview/

[5] "DB-Engines Ranking of Graph DBMS", DB-Engines,. Februar 2015,

[6] O. Jankovic, "NoSQL model podataka: Kako modelirati grafovski orijentisano", XXI naučna i biznis konferencija YU INFO 2015, prihvaćen za objavljivanje.

[7] R.V. Bruggen, "Learning Neo4j", Packt Publishing, 2014.

[8] The Neo4j Documentation v2.2.0-M02, 2014 Neo Technology, http://neo4j.com/docs/2.2.0-M02/

ABSTRACT

The challenges of modeling the actual characteristics of highly connected data, are addressed by the increasingly popular NoSQL graph database. An intuitive approach to the problem of modeling data, supported by some illustrative examples, using Neo4j NoSQL graph database, will be shown in this paper.

NoSQL GRAPH DATABASE: FROM DOMAIN TO MODEL OVER QUERY

Olivera Janković