

# Grafovske baze podataka – pregled istraživanja i budućih trendova

Kornelije Rabuzin\*, Martina Šestak\*

\*Sveučilište u Zagrebu. Fakultet organizacije i informatike, 42000 Varaždin, Republika Hrvatska  
kornelije.rabuzin@foi.hr, msestak2@foi.hr

**Sažetak** - Koncept NoSQL baza podataka obuhvaća niz tehnologija za alternativni (nerelacijski) pristup upravljanju podacima. Kao jedna od kategorija NoSQL baza podataka, grafovske baze podataka specifične su po prikazivanju podataka u obliku čvorova i veza u grafu nad kojima se prilikom izvršavanja upita izvršavaju različiti algoritmi iz teorije grafova. Društvene mreže najčešći su scenarij korištenja grafovskih baza podataka, budući da je osobe i veze između njih „najprirodnije“ prikazati kao čvorove i veze u grafovskim bazama podataka. U ovom ćemo radu napraviti pregled dosadašnjih istraživanja vezanih uz grafovske baze podataka općenito, sustave za upravljanje grafovskim bazama podataka i upitne jezike za interakciju s grafovskim bazama podataka, te ćemo posebnu pozornost obratiti na buduće trendove razvoja ove zanimljive i sve popularnije kategorije NoSQL baza podataka.

**Gljučne riječi** - grafovske baze podataka; NoSQL; Neo4j; upitni jezici

## I. UVOD

Grafovske baze podataka predstavljaju kategoriju NoSQL baza podataka čiji je koncept predstavljen 2009. godine. Koncept grafova u bazama podataka spomenut je i u ranijim istraživanjima tijekom 1970-ih godina u mrežnim bazama podataka, ali se u tom razdoblju relacijski model ipak pokazao kao bolje rješenje za pohranu podataka. Posljednjih nekoliko godina ideja je ponovno zaživjela kroz koncept NoSQL grafovskih baza podataka koje, zahvaljujući mogućnosti efikasne pohrane i pretraživanja međusobno povezanih podataka (s naglaskom na pohranu informacija o vezama između entiteta) bilježe stalni porast broja korisnika. Danas je upotreba grafovskih baza podataka dosta široka; najčešći scenariji korištenja grafovskih baza podataka su društvene mreže [1]–[4], sustavi za preporuke (primjerice, filmova) [5], detekcija financijskih prijevара ili računalnih virusa [6], itd. Međutim, uzimajući u obzir suvremene trendove u znanosti i praksi, objavljena su istraživanja u kojima je područje korištenja grafovskih baza podataka prošireno na Internet stvari (engl. Internet of Things) [7], rješavanje transportnih problema i javni transport [8]–[11], sustave za opskrbu električnom energijom [12], [13], biokemiju [14], medicinu (posebice genetiku) [15], itd. Grafovske baze podataka primjenjuju se i turističkim aplikacijama [16], te za potrebe popisa stanovništva [17] i *crowdsourcing*-a [18].

S obzirom na navedeno, može se zaključiti da područje grafovskih baza podataka budi sve veći interes kod šireg broja korisnika (istraživača, programera i sl.). Cilj ovog rada je u preglednom obliku prezentirati dosadašnja istraživanja, te glavne ideje i buduće smjerove istraživanja u području grafovskih baza podataka. Inicijalnim

pretraživanjem literature utvrđeno je da se istraživanja u području grafovskih baza podataka provode u više različitih smjerova (primjerice, optimizacija grafovskih baza podataka općenito, područja primjene grafovskih baza podataka, dubinska analiza grafovskih baza podataka i strojno učenje, upitni jezici i sučelja za rad s grafovskim bazama podataka, itd.), pa će u narednim poglavljima biti napravljen pregled istraživanja u tim smjerovima.

Struktura ovog rada je sljedeća: u Poglavlju 2 su nakon objašnjenja karakteristika grafovskih baza podataka prikazana istraživanja o grafovskim bazama podataka i pristupima poboljšanju njihovih performansi. U Poglavlju 3 napravljen je pregled grafovskih modelaa podataka razvijenih i korištenih u prošlosti, te suvremenih grafovskih modelaa podataka koji se danas koriste u grafovskim bazama podataka. Poglavlje 4 sadrži pregled istraživanja vezanih uz upitne jezike i sučelja za rad s grafovskim bazama podataka, dok su u Poglavlju 5 navedeni budući trendovi razvoja grafovskih baza podataka.

## II. GRAFOVSKE BAZE PODATAKA

Grafovska baza podataka sastoji se od niza čvorova međusobno povezanih usmjerenih vezama. Kao što je već spomenuto, grafovske baze podataka su najprikladnije za pohranu međusobno povezanih podataka jer omogućuju i pohranu informacija o vezama između entiteta.

Osim nativnog kreiranja grafovske baze podataka, nekoliko je skupina autora predložilo pristup transformacije relacijske u grafovsku bazu podataka [19] ili kreiranja grafovske baze podataka na temelju ER (engl. Entity Relationship) dijagrama [20]. U usporedbi s relacijskim bazama podataka, grafovske baze podataka smatraju se boljim načinom prikazivanja međusobno povezanih podataka, što su pokazala i brojna istraživanja. Primjerice, u [21] autori uspoređuju različite sustave, uključujući relacijske i grafovske baze, a za potrebe izrade web aplikacija. Autori navode da je grafovski model podataka najtočniji s aspekta modeliranja stvarnosti, dok PostgreSQL ima najbolje performanse.

Jedan od najčešćih načina optimiziranja performansi grafovskih baza podataka (i baza podataka općenito) je korištenje indeksa. Postoji niz istraživanja u kojima su predloženi novi indeksi za grafovske baze podataka; u [22], autori predlažu GIRAS, dok u [23] autori predlažu indeks naziva Lindex, koji indeksira podgrafove sadržane u grafovskoj bazi. Za pohranu većeg broja čvorova i veza, u [24] autori predlažu korištenje kompresije kako bi nastala komprimirana baza bez gubitka informacija.

### III. MODELI GRAFOVSKE BAZE PODATAKA

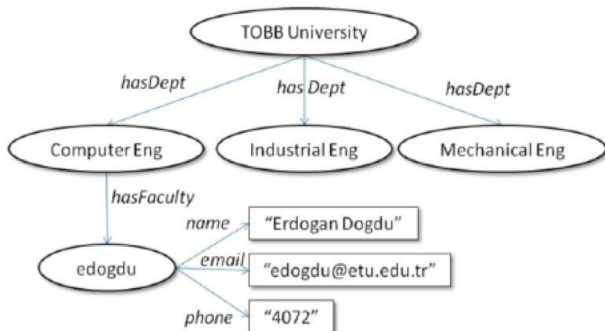
U bazi podataka podaci se pohranjuju kao odgovarajuće strukture u modelu podataka nad kojima se izvršavaju određene operacije koje taj model podataka podržava. Koncept i elementi modela podataka zajednički su svim vrstama baza podataka; svaki model podataka sadrži strukturalni, operativni i integritetni dio [25]. Kao što se za implementaciju podataka u relacijskim bazama podataka koristi relacijski model, u grafovskim bazama podataka se za to koristi grafovski model podataka. Podaci iz grafovskog modela podataka se zatim fizički mogu pohraniti nativno (u obliku grafa) ili u obliku drugih podatkovnih struktura (npr. tablica, parova ključ-vrijednost, i sl.). Iako je tijekom godina razvijeno nekoliko grafovskih modela podataka, općenito se komponente grafovskog modela podataka sastoje od sljedećih elemenata [26]:

- strukturalnu komponentu čine elementi grafa (čvorovi i veze)
- operativnu komponentu čine operatori algebre grafova za transformaciju grafova (traženje uzorka u grafu, podgrafovi, putanje, itd.)
- integritetnu komponentu čine integritetna ograničenja nad shemom ili podacima (primjerice, jedinstveni nazivi oznaka čvorova)

Prema tome, model podataka u pozadini grafovskih baza podataka se može predstavljati u obliku usmjerenog grafa koji se sastoji od čvorova (entiteti iz stvarnog svijeta) i veza (povezuju čvorove, gdje jedan predstavlja izvorišni, a drugi odredišni čvor). U usporedbi s relacijskim bazama podataka, grafovske baze podataka su fleksibilnije s obzirom na shemu (unos novih i promjena postojećih čvorova i veza ne zahtijeva promjenu postojećih čvorova i veza). Osim toga, čvorovi u grafovskom modelu podataka sadrže fizičke pokazivače na sebi susjedne čvorove s kojima povezani (svojstvo indeksa bez susjedstva), što značajno utječe na brzinu izvršavanja upita i pretraživanje baze podataka.

Detaljan pregled grafovskih modela podataka razvijenih tijekom niza godina napravili su R. Angles i C. Gutierrez [26], a u kasnijem radu Angles je napravio usporedbu suvremenih grafovskih modela podataka [27]. Neki od značajnijih grafovskih modela podataka razvijenih u prošlosti su:

- GRAM [28] model je bio prvenstveno namijenjen pretraživanju hipertekstova



Slika 1. Primjer RDF modela podataka [56]

(hipertekst predstavlja graf u kojem su čvorovi dokumenti, a veze poveznice između dokumenata),

- GOOD [29] model je razvijen kao odgovor na pojavu objektno-orijentiranih programskih jezika poput Smalltalka radi efikasnog prikaza podataka u objektno-orijentiranim bazama podataka
- GROOVY [30] grafički model čija se osnovna struktura hipergraf koristila za prezentaciju složenih objekata i objektno-orijentiranih mehanizama

Danas se u grafovskim bazama podataka najčešće koriste sljedeći modeli podataka:

1. RDF (engl. Resource Description Framework) model podataka
2. grafovski model podataka sa svojstvima (engl. property graph data model)

#### A. RDF model podataka

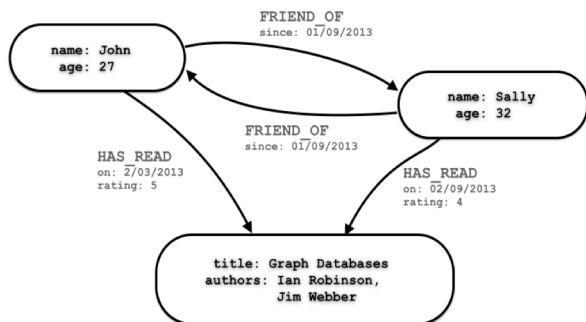
RDF je standardizirani format za prikaz podataka na Webu u obliku usmjerenog grafa s oznakama [31]. RDF tretira entitete kao resurse koji su na modelu podataka prikazani kao trojke *subjekt-predikat-objekt*, gdje subjekt označava entitet koji se opisuje, predikat označava svojstvo subjekta, a objekt vrijednost svojstva subjekta [26]. Niz međusobno povezanih RDF trojki čini RDF izraz (engl. expression), a niz RDF izraza se prikazuje kao RDF graf. Na slici 1 prikazan je jednostavan primjer RDF grafa. Prikazani graf sastoji se, primjerice, od RDF izraza koji označavaju koje odjele ima određeno sveučilište “TOBB University”. U navedenom primjeru “TOBB University” predstavlja RDF trojku, dok traženi RDF izraz ima oblik “<university><hasDept><department>”.

Za pretraživanje RDF modela podataka razvijen je i koristi se SPARQL upitni jezik čija je sintaksa slična sintaksi SQL-a. SPARQL upit se temelji na definiranju uzoraka za pretraživanje (engl. pattern matching) te varijabli koje se tijekom procesiranja upita zamjenjuju pronađenim podacima [26].

#### B. Grafovski model podataka sa svojstvima

Model podataka sa svojstvima (engl. property graph data model) prikazuje podatke u obliku čvorova i veza između čvorova. Danas je ovaj model službeni model Neo4j SUGBP-a, najpopularnijeg sustava na tržištu. Za razliku od RDF modela podataka u kojem se svojstvo čvora prikazuje kao dodatna veza i čvor u modelu, svojstva čvorova, ali i veza, se u modelu podataka sa svojstvima prikazuju izravno kao svojstva čvorova u obliku parova ključ-vrijednost (ključ predstavlja naziv svojstva, dok je vrijednost stvarna vrijednost određenog svojstva).

Ovakav način prikaza smanjuje suvišnu kompleksnost modela podataka, ali i pojednostavnjuje definiranje upita i uzoraka za pretraživanje. Za dodatnu klasifikaciju čvorova i veza moguće je definirati i jednu ili više oznaka (engl. label) čvorova i veza.



Slika 2. Primjer modela podataka sa svojstvima [55]

Na slici 2 prikazan je jednostavan primjer modela podataka sa svojstvima koji se sastoji od 3 čvora; dva čvora predstavljaju osobe sa svojstvima “name” i “age”, dok jedan čvor predstavlja knjigu sa svojstvima “title” i “authors”. Čvorovi su spojeni vezama s oznakama “HAS\_READ” i “FRIEND\_OF” koje također imaju svoja svojstva.

#### IV. UPITNI JEZICI ZA GRAFOVSKE BAZE PODATAKA

Prije pojave grafovskih baza podataka razvijen je niz upitnih jezika za grafove u kojima su implementirani koncepti objektno-orijentirane paradigme; G-Log [34] je koristio karakteristike deduktivnih i objektno-orijentiranih baza podataka, GraphDB [35] integrira koncepte objektno-orijentiranog modeliranja i modeliranja graf struktura, te GOQL [36] za prikaz i pretraživanje multimedijских grafova.

S pojavom prvih grafovskih baza podataka nastavljena su istraživanja vezana uz efikasno pretraživanje grafovskih baza podataka. He i Singh su razvili GraphQL upitni jezik namijenjen pretraživanju heterogenih atributa i struktura u grafovskim bazama podataka [37]. Osnovna jedinica obrade u GraphQL upitnom jeziku je uzorak u grafu, dok graf predstavlja osnovnu jedinicu informacije.

GraQL [38] je upitni jezik za pretraživanje grafova s atributima (engl. attributed graphs) u grafovskim bazama podataka s visokim performansama u klasterima. Zbog problema particioniranja grafova, autori ističu efikasno mapiranje sustava za upravljanje grafovskim bazama podataka i klastera te efikasno izvršavanje kompleksnih upita nad graf bazom podataka kao još uvijek prisutne izazove koje je potrebno svladati.

Za pretraživanje modela podataka sa svojstvima koriste se upitni jezici Cypher i Gremlin. Gremlin je jezik za obilazak grafa i danas se rjeđe koristi jer je za definiranje upita potrebno poznavati sintaksu jezika koja je nešto kompleksnija, dok je Cypher zbog svoje jednostavne sintakse slične SQL-u popularniji među korisnicima i službeni upitni jezik za Neo4j bazu podataka. Što se tiče performansi navedenih upitnih jezika, one su slične u slučaju jednostavnijih upita, dok Gremlin ostvaruje bolje performanse u slučaju obilaska grafa u više razina [32], [33].

Osim SPARQL-a, Cyphera i Gremlina, tijekom posljednjih godina razvijen je i PGQL, upitni jezik namijenjen pretraživanju grafovskog modela sa svojstvima (engl. property graph model). Kao i Cypher, PGQL je

deklarativni upitni jezik i temelji se na paradigmi pronalaženja uzoraka u grafu (engl. graph pattern matching) [39], no, u usporedbi sa Cypherom, PGQL nudi neke dodatne mogućnosti i operacije (primjerice, formulaciju upita za pretraživanje regularnih putanja).

U novije vrijeme pojavio se novi pristup pretraživanju podataka u grafovskim bazama podataka čiji je cilj olakšati pretraživanje grafovske baze podataka korisnicima koji nisu upoznati s domenom ni sintaksom upitnih jezika, a jedan od načina je formulacija upita zadavanjem primjera u tzv. egzemplarnim upitnim jezicima (engl. exemplar queries). Formalizacija navedenih upitnih jezika dana je u [40], gdje su autori prikazali fundamentalnu razliku u odnosu na klasične tzv. query-by-example (QBE) upitne jezike. QBE upitni jezici omogućuju korisniku da na sebi razumljiviji način formuliра upit, koji ujedno predstavlja i primjer, te se izravno evaluira od strane mehanizma za procesiranje upita (engl. query processing engine). Suprotno tome, egzemplarni upitni jezici [40] služe za specifikaciju tipa elemenata koje korisnik očekuje u rezultatu, odnosno uzorka iz željenog skupa vrijednosti u bazi podataka, što se u algebri grafova svodi na traženje odgovarajućih izomorfnih grafova. Dosad je razvijeno nekoliko vizualnih upitnih jezika za grafovske baze podataka (primjerice, DaVinci [41], GBLENDER [42], Gremlin By Example [43], itd.) .

Kao jedan od većih nedostataka grafovskih baza podataka (i NoSQL baza podataka općenito) ističe se nepostojanje standardiziranog i jedinstvenog upitnog jezika ekvivalentnog SQL-u u relacijskim bazama podataka. Jedna od posljedica navedenog problema je otežana migracija podataka između sustava za upravljanje grafovskim bazama podataka (SUGBP) koji koriste različite upitne jezike i modele podataka [44].

Prema tome, definiranje standardnog upitnog jezika za grafovske baze podataka predstavlja značajan izazov u sadašnjim i budućim istraživanjima. Neo4j korporacija je 2016. godine pokrenula projekt nazvan “openCypher” čiji je cilj standardizacija najčešće korištenog Cypher upitnog jezika [45] na čemu će se raditi kroz sljedećih nekoliko godina. U [44], autori su prikazali formalnu specifikaciju openCypher upitnog jezika korištenjem relacijske algebre koju su proširili operatorima nad grafovima (npr. dohvaćanje čvorova, veza i putanja).

#### V. DISKUSIJA I SISTEMATIZACIJA ISTRAŽIVANJA

Kao što je već spomenuto, istraživanja u području grafovskih baza podataka bave se različitim problematikama, a u ovom radu su navedena istraživanja sistematizirana unutar sljedećih kategorija:

- grafovske baze podataka i optimizacija njihovih performansi
- modeli grafovskih baza podataka
- upitni jezici za grafovske baze podataka
- dubinska analiza grafovskih baza podataka
- primjena grafovskih baza podataka u strojnom učenju

Identificirana istraživanja u navedenim područjima prikazana su u Tablici 1 (TABLICA 1).

TABLICA 1. PREGLED ISTRAŽIVANJA U PODRUČJU GRAFOVSKIH BAZA PODATAKA

Područje istraživanja	Istraživanja
Grafovske baze podataka i optimizacija performansi	[19], [20], [21], [22], [23], [24]
Modeli grafovskih baza podataka	[26], [27], [28], [29], [30], [32], [33]
Upitni jezici za grafovske baze podataka	[32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45]
Dubinska analiza grafovskih baza podataka	[46], [47], [48], [49]
Primjena grafovskih baza podataka u strojnom učenju	[50], [51], [52]

## VI. BUDUĆI TRENDVI

Uz već navedena istraživanja u kojima se područje primjene grafovskih baza podataka sve više proširuje i prilagođava suvremenim trendovima u razvoju tehnologije, pregledom literature identificirano je još nekoliko značajnijih trendova razvoja ove NoSQL tehnologije, poput rudarenja grafovskim bazama podataka, primjene u strojnom učenju, te implementacije grafovskih baza podataka u oblaku i distribuiranom okruženju.

Dubinska analiza grafovskih baza podataka (engl. graph mining) se odnosi na ekstrakciju novog znanja iz podataka pohranjenih u grafovskoj bazi podataka [46] i obuhvaća velik broj različitih tehnika koje se u literaturi kategoriziraju kao tehnike klasteriranja i klasifikacije grafova, te dubinske analize podgrafova [47]. Tijekom godina objavljen je niz znanstvenih radova u kojima su autori objavili nove pristupe, algoritme ili tehnike iz svake od prethodno navedenih tehnika.

U [48], autori su predstavili GRAMI okvir za otkrivanje učestalih podgrafova unutar jednog velikog grafa. X. Yan i J. Han su u svom istraživanju pristupa za pronalaženje čestih uzoraka u grafovskim bazama podataka predstavili novi algoritam gSpan [49] koji, u usporedbi sa sličnim pristupima, na efikasniji način pronalazi česte podgrafove.

U području strojnog učenja tehnike poput klasifikacije i klasteriranja grafova koriste se u detekciji objekata ili uzoraka [50], [51]. Osim toga, H. Bunke i K. Rispan su u svom radu [52] analizirali napredak u području klasifikacije grafova i primjene navedene tehnike u analizi dokumenata.

Zbog suvremenog porasta količine podataka koje nije moguće procesirati samo na jednom računalu, javlja se potreba za distribuiranim bazama podataka koje će riješiti taj problem na efikasan način. Kad se radi o distribuiranim grafovskim bazama podataka, izazov je još veći budući da je potrebno particionirati graf. B. Shao, H. Wang i Y. Li su predstavili Trinity mehanizam za distribuiranu grafovsku bazu podataka u oblaku [53] koja, zahvaljujući optimiziranom upravljanju memorijom, omogućuje efikasno dohvaćanje distribuiranih podataka.

Grafovske baze podataka mogu se primijeniti i u području poslovne inteligencije (engl. business intelligence) u kontekstu analize povezanih podataka. U [54], autori su predstavili sustav za integraciju i analizu podataka temeljenu na grafovima, odnosno vezama između čvorova u grafu. U razvijenom sustavu, integracijom podataka iz heterogenih izvora dobivaju se graf koji sadrži stvarne podatke iz izvora i graf s metapodacima koji se u sljedećim koracima mogu dodatno analizirati.

## VII. ZAKLJUČAK

Grafovske baze podataka predstavljaju područje NoSQL tehnologija koja posljednjih godina budi zanimanje kod različitih vrsta korisnika. U ovom radu napravljen je pregled literature vezane uz grafovske baze podataka kako bi se identificirale dosadašnje, kao i buduće, teme istraživanja. Budući da se područje grafovskih baza podataka pokazalo prilično širokim i pronađen je velik broj istraživanja, u ovom radu analizirana su istraživanja vezana uz grafovske baze podataka općenito i njihove performanse, dostupne grafovske modele podataka, upitne jezike za rad s grafovskim bazama podataka, te buduće trendove razvoja grafovskih baza podataka.

## ZAHVALA

Ovaj rad nastao je u sklopu projekta “Advanced Regional Civil Emergency Coordination Pilot (ARCECP)” (oznaka NATO SPS G4968) pod sponzorstvom NATO-Science for Peace and Security (SPS), US Department of Homeland Security Science & Technology Directorate, MIT Lincoln Laboratory (MIT LL) i Državne uprave za zaštitu i spašavanje (DUZS).

## LITERATURA

- [1] I. Hoque and I. Gupta, “Disk Layout Techniques for Online Social Network Data,” *IEEE Internet Comput.*, vol. 16, no. 3, pp. 24–36, 2012.
- [2] G. Drakopoulos, A. Kanavos, and A. Tsakalidis, “Evaluating Twitter Influence Ranking with System Theory,” in *Proceedings of the 12th International Conference on Web Information Systems and Technologies, Vol 1 (WEBIST)*, 2016, pp. 113–120.
- [3] C. Constantinov, C. M. Poteras, and M. L. Mocanu, “Performing real-time social recommendations on a highly-available graph database cluster,” in *Carpathian Control Conference (ICCC), 2016 17th International*, 2016, pp. 116–121.
- [4] E. Olshannikova, T. Olsson, J. Huhtamäki, and H. Kärkkäinen, “Conceptualizing Big Social Data,” *J. Big Data*, vol. 4, no. 1, p. 3, 2017.
- [5] A. Sharma and S. Batra, “Enhancing the Accuracy of Movie Recommendation System Based on Probabilistic Data structure and Graph Database,” in *2015 Fifth International Conference on Advances in Computing and Communications (ICACC)*, 2015, pp. 41–45.
- [6] X. Hu, T. Chiueh, and K. G. Shin, “Large-Scale Malware Indexing Using Function-Call Graphs,” in *CCS’09: Proceedings of the 16th ACM Conference on Computer and Communications Security*, 2009,

- pp. 611–620.
- [7] G. M. D'silva, S. Thakare, and V. A. Bharadi, "Real-time Processing of IoT Events using a Software as a Service (SaaS) Architecture with Graph Database," in *2016 International Conference on Computing Communication Control and Automation (ICCCUBEA)*, 2016.
- [8] M. Miler, D. Medak, and D. Odobasic, "The Shortest Path Algorithm Performance Comparison in Graph and Relational Databases on a Transportation Network," *PROMET-Traffic Transp.*, vol. 26, no. 1, pp. 75–82, 2014.
- [9] E. Eser, F. Kocayusufoglu, B. Eravci, H. Ferhatosmanoglu, and J. L. Larriba-Pey, "Generating Time-Varying Road Network Data Using Sparse Trajectories," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 2016, pp. 1118–1124.
- [10] I. Benenson, E. Ben-Elia, Y. Rofe, and D. Geyzersky, "The benefits of a high-resolution analysis of transit accessibility," *Int. J. Geogr. Inf. Sci.*, vol. 31, no. 2, pp. 213–236, 2017.
- [11] A. Czerepicki, "Application of graph databases for transport purposes," *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 64, no. 3, Jan. 2016.
- [12] G. Ravikumar and S. A. Khaparde, "A Common Information Model Oriented Graph Database Framework for Power Systems," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 2560–2569, Jul. 2017.
- [13] M. Yisong, W. Zhigang, G. Lin, Z. Baorong, and L. Rongrong, "Study on the relationship between transmission line failure rate and lightning information based on Neo4j," in *2014 International Conference on Power System Technology (POWERCON)*, 2014.
- [14] S. Sheikhezadeh, M. E. Schranz, M. Akdel, D. de Ridder, and S. Smit, "PanTools: representation, storage and exploration of pan-genomic data," *Bioinformatics*, vol. 32, no. 17, pp. 487–493, Sep. 2016.
- [15] C. J. Saunders, M. J. S. Dashti, and J. Gamielidien, "Semantic interrogation of a multi knowledge domain ontological model of tendinopathy identifies four strong candidate risk genes," *Sci. Rep.*, vol. 6, Jan. 2016.
- [16] A. Maccioni and M. Collina, "Graph databases in the browser: using LevelGraph to explore New Delhi," *Proc. VLDB Endow.*, vol. 9, no. 13, pp. 1469–1472, 2016.
- [17] K. T. Yar and K. M. L. Tun, "Searching Personnel Relationship from Myanmar Census Data using Graph Database and Deductive Reasoning Prolog Rules," in *2016 International Conference on Computer Communication and Informatics (ICCCI)*, 2016.
- [18] V. Liptchinsky, B. Satzger, S. Schulte, and S. Dustdar, "Crowdstore: A crowdsourcing graph database," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 2016, vol. 163, pp. 72–81.
- [19] O. Orel, S. Zakosek, and M. Baranovic, "Property Oriented Relational-To-Graph Database Conversion," *Automatika*, vol. 57, no. 3, pp. 836–845, 2016.
- [20] N. Roy-Hubara, L. Rokach, B. Shapira, and P. Shoval, "Modeling Graph Database Schema," *IT Prof.*, no. 6, pp. 34–43, 2017.
- [21] M. Plechawska-Wojcik and D. Rykowski, "Comparison of Relational, Document and Graph Databases in the Context of the Web Application Development," in *INFORMATION SYSTEMS ARCHITECTURE AND TECHNOLOGY, ISAT 2015, PT II*, 2016, vol. 430, pp. 3–13.
- [22] M. Azaouzi and L. Ben Romdhane, "A Minimal Rare Substructures-Based Model for Graph Database Indexing," in *Intelligent Systems Design and Applications (ISDA 2016)*, 2017, vol. 557, pp. 250–259.
- [23] D. Yuan and P. Mitra, "Lindex: a lattice-based index for graph databases," *VLDB J.*, vol. 22, no. 2, pp. 229–252, Apr. 2013.
- [24] G. B. A. Sutrisna, K. R. W. Saleh, and A. A. Gozali, "Implementation of GRAC Algorithm (Graph Algorithm Clustering) in Graph Database Compression," in *2015 3rd International Conference on Information and Communication Technology (ICoICT)*, 2015, pp. 391–395.
- [25] M. Maleković and K. Rabuzin, *Uvod u baze podataka*. Varaždin, Croatia: Fakultet organizacije i informatike, Varaždin, 2016.
- [26] R. Angles and C. Gutierrez, "Survey of graph database models," *ACM Comput. Surv.*, vol. 40, no. 1, pp. 1–39, 2008.
- [27] R. Angles, "A comparison of current graph database models," in *Data Engineering Workshops (ICDEW)*, *2012 IEEE 28th International Conference on*, 2012, pp. 171–177.
- [28] B. Amann and M. Scholl, "Gram: a graph data model and query languages," in *Proceedings of the ACM conference on Hypertext*, 1992, pp. 201–211.
- [29] M. Gyssens, J. Paredaens, J. den Bussche, and D. Van Gucht, "A graph-oriented object database model," *IEEE Trans. Knowl. Data Eng.*, vol. 6, no. 4, pp. 572–586, 1994.
- [30] M. Levene and A. Poulouvasilis, "An object-oriented data model formalised through hypergraphs," *Data Knowl. Eng.*, vol. 6, no. 3, pp. 205–224, 1991.
- [31] W3C, "SPARQL Query Language for RDF," 2008. [Online]. Available: <https://www.w3.org/TR/rdf-sparql-query/>. [Accessed: 18-Feb-2018].
- [32] F. Holzschuher and R. Peinl, "Querying a graph database – language selection and performance considerations," *J. Comput. Syst. Sci.*, vol. 82, no. 1, pp. 45–68, Feb. 2016.
- [33] M. Šestak, "Usporedba jezika za graf baze podataka," University of Zagreb. Faculty of Organization and Informatics Varaždin. Department of Theoretical and Applied Foundations of Information Sciences., 2016.
- [34] J. Paredaens, P. Peelman, and L. Tanca, "G-Log: a

- graph-based query language,” *IEEE Trans. Knowl. Data Eng.*, vol. 7, no. 3, pp. 436–453, 1995.
- [35] R. H. Güting, *GraphDB: a data model and query language for graphs in databases*. Citeseer, 1994.
- [36] L. Sheng, Z. m. Ozsoyoglu, and G. Ozsoyoglu, “A Graph Query Language and Its Query Processing,” *Proc. 15th Int. Conf. Data Eng. (Cat. No.99CB36337)*, 1999.
- [37] H. He and A. K. Singh, “Graphs-at-a-time: query language and access methods for graph databases,” in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 405–418.
- [38] D. Chavarria-Miranda, V. G. Castellana, A. Morari, D. Haglin, and J. Feo, “GraQL: A Query Language for High-Performance Attributed Graph Databases,” in *2016 IEEE 30th International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2016, pp. 1453–1462.
- [39] O. Van Rest, S. Hong, J. Kim, X. Meng, and H. Chafi, “PGQL: a Property Graph Query Language,” *Proc. Fourth Int. Work. Graph Data Manag. Exp. Syst. - GRADES 16*, 2016.
- [40] D. Mottin, M. Lissandrini, Y. Velegrakis, and T. Palpanas, “Exemplar Queries: Give me an Example of What You Need,” *Proc. VLDB Endow.*, vol. 7, no. 5, pp. 365–376, Jan. 2014.
- [41] J. Zhang, S. S. Bhowmick, H. H. Nguyen, B. Choi, and F. Zhu, “DAVINCI: Data-driven Visual Interface Construction for Subgraph Search in Graph Databases,” in *2015 IEEE 31st International Conference on Data Engineering (ICDE)*, 2015, pp. 1500–1503.
- [42] C. Jin, S. S. Bhowmick, X. Xiao, B. Choi, and S. Zhou, “GBLENDER: visual subgraph query formulation meets query processing,” in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, 2011, pp. 1327–1330.
- [43] K. Rabuzin, M. Maleković, and M. Šestak, “Gremlin By Example,” in *International Conference on Advances in Big Data Analytics*, 2016.
- [44] J. Marton, G. Szárnyas, and D. Varró, “Formalising openCypher Graph Queries in Relational Algebra,” *Adv. Databases Inf. Syst. Lect. Notes Comput. Sci.*, pp. 182–196, 2017.
- [45] Neo4j Inc., “openCypher.” [Online]. Available: <https://www.opencypher.org/>. [Accessed: 05-Feb-2018].
- [46] D. J. Cook and L. B. Holder, *Mining graph data*. John Wiley & Sons, 2006.
- [47] S. U. Rehman, A. U. Khan, and S. Fong, “Graph mining: A survey of graph mining techniques,” in *Digital Information Management (ICDIM), 2012 Seventh International Conference on*, 2012, pp. 88–92.
- [48] M. Elseidy, E. Abdelhamid, and S. Skiadopoulos, “GRAMI: Frequent Subgraph and Pattern Mining in a Single Large Graph,” *Proc. VLDB Endow.*, 2014.
- [49] X. Yan and J. Han, “gSpan: Graph-Based Substructure Pattern Mining,” in *Proceedings of IEEE International Conference on Data Mining*, 2002, vol. 1, no. d, pp. 721–724.
- [50] R. Raveaux, S. Adam, P. Heroux, and E. Trupin, “Learning graph prototypes for shape recognition,” *Comput. Vis. IMAGE Underst.*, vol. 115, no. 7, SI, pp. 905–918, Jul. 2011.
- [51] R. Kaspar and B. Horst, *Graph classification and clustering based on vector space embedding*, vol. 77. World Scientific, 2010.
- [52] H. Bunke and K. Riesen, “Recent advances in graph-based pattern recognition with applications in document analysis,” *Pattern Recognit.*, vol. 44, no. 5, pp. 1057–1067, 2011.
- [53] B. Shao, H. Wang, and Y. Li, “Trinity- A Distributed Graph Engine on a Memory Cloud,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2013.
- [54] H. Chen, R. H. L. Chiang, and V. C. Storey, “Business intelligence and analytics: from big data to big impact,” *MIS Q.*, pp. 1165–1188, 2012.
- [55] M. Lal, *Neo4j Graph Data Modeling*. Packt Publishing Ltd, 2015.
- [56] E. Dogdu, S. Hakimov, and S. Yumusak, “A data-model driven web application development framework,” in *Proceedings of the 2014 ACM Southeast Regional Conference*, 2014, p. 47.