

Geoanaliza koristeći graf ili slične baze podataka

Mitrović, Goran

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:024046>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-14**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

Stručni prijediplomski studij Računarstvo

**GEOANALIZA KORISTEĆI GRAF ILI SLIČNE BAZE
PODATAKA**

Završni rad

Goran Mitrović

Osijek, 2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1S: Obrazac za ocjenu završnog rada na stručnom prijediplomskom studiju****Ocjena završnog rada na stručnom prijediplomskom studiju**

Ime i prezime pristupnika:	Goran Mitrović
Studij, smjer:	Stručni prijediplomski studij Elektrotehnika, smjer Informatika
Mat. br. pristupnika, god.	A3566, 03.09.2008.
JMBAG:	0165038807
Mentor:	izv. prof. dr. sc. Alfonzo Baumgartner
Sumentor:	doc. dr. sc. Tomislav Galba
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	prof. dr. sc. Tomislav Keser
Član Povjerenstva 1:	izv. prof. dr. sc. Alfonzo Baumgartner
Član Povjerenstva 2:	izv. prof. dr. sc. Časlav Livada
Naslov završnog rada:	Geoanaliza koristeći graf ili slične baze podataka
Znanstvena grana završnog rada:	Obradba informacija (zn. polje računarstvo)
Zadatak završnog rada:	Potrebno je proučiti graf baze podataka i njihovu podršku za tip podataka koji sadrži geolokacijske informacije. U upitnom jeziku takve baze podataka napraviti primjere koji pokazuju načine korištenja ovakvih podataka i osnovne operacije koje su podržane s takvim podacima. (sumentor Tomislav Galba)
Datum ocjene pismenog dijela završnog rada od strane mentora:	20.09.2024.
Ocjena pismenog dijela završnog rada od strane mentora:	Izvrstan (5)
Datum obrane završnog rada:	30.09.2024.
Ocjena usmenog dijela završnog rada (obrane):	Izvrstan (5)
Ukupna ocjena završnog rada:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio stručni prijediplomski studij:	30.09.2024.



FERIT

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O IZVORNOSTI RADA

Osijek, 30.09.2024.

Ime i prezime Pristupnika:	Goran Mitrović
Studij:	Stručni prijediplomski studij Elektrotehnika, smjer Informatika
Mat. br. Pristupnika, godina upisa:	A3566, 03.09.2008.
Turnitin podudaranje [%]:	4

Ovom izjavom izjavljujem da je rad pod nazivom: **Geoanaliza koristeći graf ili slične baze podataka**

izrađen pod vodstvom mentora izv. prof. dr. sc. Alfonzo Baumgartner

i sumentora doc. dr. sc. Tomislav Galba

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

Sadržaj

1. UVOD.....	1
1.1. Zadatak završnog rada	1
2. PREGLED PODRUČJA TEME	2
2.1. Primjena graf baza podataka	3
2.2. Topološka analiza.....	4
2.3. Alati.....	7
2.4. Izazovi i ograničenja.....	10
2.5. Budući trendovi i mogućnost poboljšanja	11
3. KORIŠTENI ALATI I TEHNOLOGIJE	13
3.1. Neo4j.....	13
3.2. Upitni jezik Cypher	14
3.3. Neo4j desktop	17
4. EKSPERIMENTALNI REZULTATI	20
4.1. Izračun najkraćih puteva	20
4.2. Prikaz podataka u ovisnosti o zadanim parametrima	22
5. ZAKLJUČAK	36
LITERATURA.....	37
SAŽETAK	38
ABSTRACT.....	39
PRILOG (on-line).....	40

1.UVOD

U ovom radu opisan je spoj graf baza podataka i geoanalize koji postaje snažan alat za obradu i analizu geolokacijskih podataka. Takve analize koristeći graf baza uvelike pomažu u različitim granama i iz tog razloga se koriste sve više, a naročito su od velikog značaja prilikom rada s velikom količinom podataka. Također, da bi se analiza mogla izvršiti potrebno je ispuniti uvjete kao što su prikupljanje i priprema prostornih podataka, izgradnja graf modela, primjena algoritama za analizu mreža (npr. najkraći put, centralnost, klasterizacija) te vizualizacija rezultata.

U drugom poglavlju opisani su geoanaliza i graf baza podataka te njihovo zajedničko korištenje u brojnim područjima. Prikazane su popularne graf baze podataka i navedena područja primjene. Nadalje, spomenute su prednosti kojih ne manjka, ali navedeni su i pojašnjeni izazovi i nedostaci. U ovom poglavlju opisana je trenutna primjena geoanalize u graf bazama podataka i razmatranje budućih trendova i mogućih poboljšanja u tehnologiji i metodologiji.

U trećem poglavlju nalazi se opis Neo4j graf baze podataka gdje je naveden povijesni razvitak, način i područja primjene kao i prednosti. U ovom dijelu nalaze se i Cypher, deklarativni upitni jezik Neo4j, te Neo4j Desktop u kojem su izvršavani upiti na bazu.

Poglavlje četiri sadrži dobivene rezultate raznih upita na bazi podataka Neo4j na projektu *São Paulo Public Transport*.

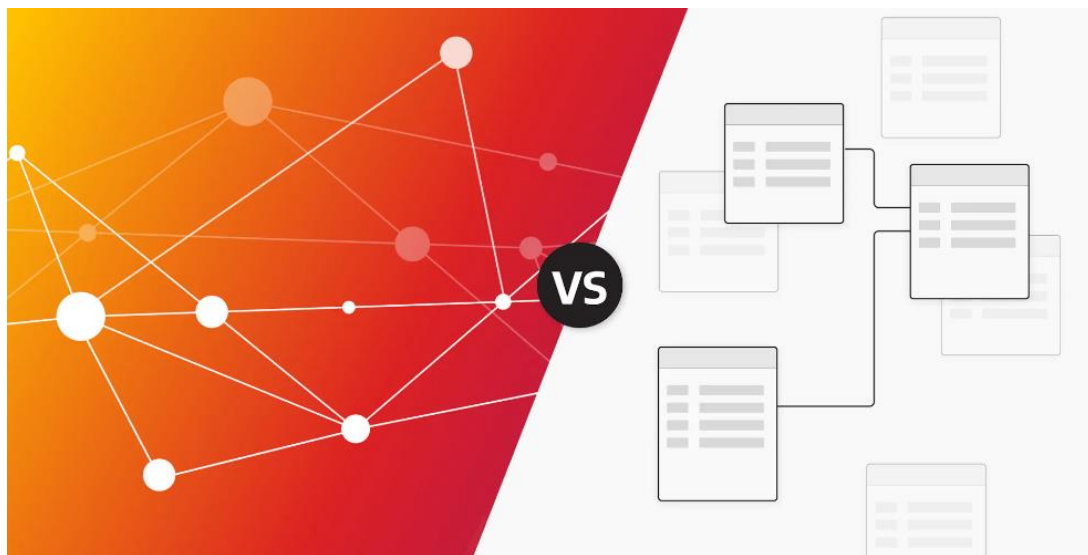
1.1. Zadatak završnog rada

Potrebno je proučiti graf baze podataka i njihovu podršku za tip podataka koji sadrži geolokacijske informacije. U upitnom jeziku takve baze podataka napraviti primjere koji pokazuju načine korištenja ovakvih podataka i osnovne operacije koje su podržane s takvim podacima.

2. PREGLED PODRUČJA TEME

Geoanaliza je multidisciplinarno područje koje se bavi prikupljanjem, analizom i interpretacijom geolokacijskih podataka čija se količina značajno povećala razvojem tehnologije što dovodi do potrebe za naprednijim pristupima za njihovu obradu i analizu. Ti podaci mogu uključivati informacije o fizičkim prostorima, objektima koji se nalaze u njima te odnosima između tih objekata. Geoanaliza omogućuje razumijevanje i vizualizaciju složenih prostornih odnosa [1].

Geoanaliza se tradicionalno oslanjala na relacijske baze podataka koje omogućuju pohranu i obradu podataka u strukturiranom obliku. Rastućim potrebama za analizom složenih i velikih mreža pojavljuju se novi pristupi koji bolje odgovaraju tim izazovima kao što su graf baze podataka koje nude znatno efikasnija rješenja. Na slici 2.1. prikazan je način zapisa podataka između graf baza podataka i relacijskih baza.



Slika 2.1. Grafički prikaz graf baza podataka i relacijskih baza

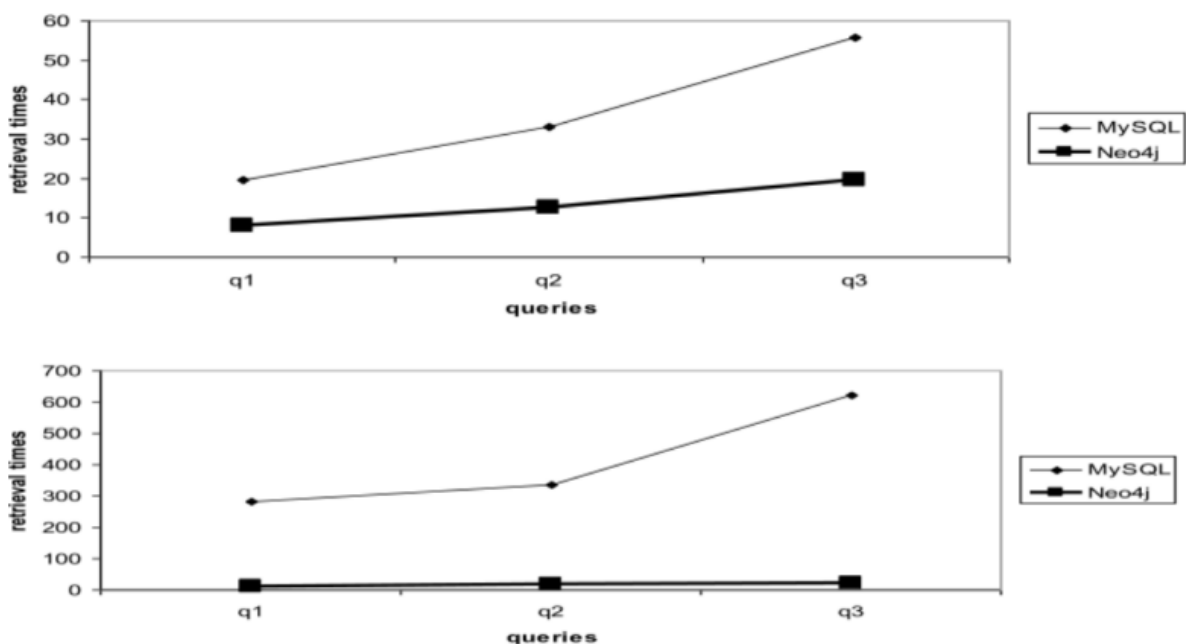
Graf baze podataka su specijalizirane baze podataka dizajnirane za pohranu, pretraživanje i analizu podataka strukturiranih u obliku grafa koji se sastoji od čvorova (koji predstavljaju entitete) i veza (koje predstavljaju odnose između entiteta). Ovaj pristup omogućuje jednostavnije modeliranje i analizu složenih odnosa u podacima na prirodni i učinkovitiji način, posebno kada su ti odnosi međusobno povezani na puno načina što ih čini sve popularnijima.

Za razliku od relacijskih baza podataka gdje su podaci pohranjeni u tablicama, graf baze podataka omogućuju izravno modeliranje odnosa među entitetima što rezultira većom fleksibilnošću i bržim

performansama pri izvršavanju složenih upita. Primjerice, u tradicionalnim relacijskim bazama podataka, pretraga koja uključuje više razina povezanosti može biti vrlo spora i neefikasna dok graf baze podataka mogu takve operacije izvoditi puno brže.

Upotrebom graf baza podataka u geoanalizi omogućuje se brža i efikasnija analiza velikih i složenih skupova podataka, što rezultira preciznijim i bržim donošenjem odluka u područjima koja ovise o prostornoj analizi. S obzirom na to, graf baze podataka predstavljaju važan korak naprijed u razvoju alata za geoanalizu, omogućujući inovativne pristupe i rješenja za složene probleme u ovom području [2].

Na slici 2.2. prikazan je odnos vremena izvršenja relacijskih i graf baza podataka koji je zabilježen prilikom istraživanja. Na gornjem grafu prikazani su rezultati mjerenja vremena izraženim u milisekundama za upite koji su provedeni na uzorku od 100 objekata, a donji graf prikazuje rezultate mjerenja vremena za upite nad 500 objekata u bazi podataka.



Slika 2.2. Rezultati usporedbe prema [3]

2.1. Primjena graf baza podataka

Primjena graf baza podataka s geolokacijskom podrškom značajno je utjecala na različita područja. Prema [4], u urbanističkom planiranju, ove baze podataka koriste se za analizu infrastrukturnih mreža kao što su ceste i vodovod kako bi se optimizirale rute ili identificirala kritična područja

koja zahtijevaju intervenciju. Graf baza podataka je vrlo učinkovita u modeliranju i optimizaciji mreža transporta, bilo da se radi o gradskom javnom prijevozu ili međunarodnoj logistici.

Mreže transporta, koje uključuju međusobno povezane rute, stanice i putnike, mogu biti vrlo složene. Graf pomaže u modeliranju tih mreža kako bi se optimizirale rute, pronašle najučinkovitije putanje i analizirali tokovi putnika. To omogućuje optimizaciju resursa i smanjenje troškova u transportnim sustavima.

U financijskoj industriji koristi se za praćenje složenih mreža transakcija, otkrivanje pranja novca i analizu rizika. Korištenjem grafova za modeliranje transakcija, financijske institucije mogu brzo prepoznati nepravilnosti i otkriti potencijalne prijetnje ili sumnjive aktivnosti. Posebno je korisno za praćenje transakcijskih putova kroz složene mreže povezanih entiteta, gdje bi relacijske baze podataka bile neučinkovite.

Preporučiteljski sustavi poput onih koji se koriste u e-trgovini, mogu koristiti graf baze podataka za analizu korisničkih preferencija i ponašanja te povezivanje tih podataka s proizvodima ili uslugama. Na primjer, graf baza podataka može analizirati uzorke kupnje i ponašanja korisnika kako bi se predložili novi proizvodi na temelju sličnosti s prethodnim kupcima ili njihovim interesima. Amazon i Netflix koriste slične modele za preporučivanje proizvoda i sadržaja.

Društvene mreže također koriste geolokacijske podatke u graf bazama podataka za analizu korisničkih interakcija u fizičkom prostoru što omogućuje ciljanje sadržaja i reklama na temelju lokacije korisnika. Društvene mreže, kao što su Facebook, LinkedIn i Twitter, temelje se na analizi veza između korisnika. S obzirom na sve veći broj aplikacija koje ovise o prostornim podacima, graf baze podataka s geolokacijskom podrškom postaju nezamjenjive u analizi i donošenju odluka [5-6].

2.2. Topološka analiza

Graf baze podataka podržavaju različite tipove podataka za rad s geolokacijskim podacima. Geometrijski podaci su podijeljeni na točke koje predstavljaju specifične lokacije na geografskoj karti, definirane parom koordinata (geografska širina i dužina) te se u graf bazama podataka točke obično pohranjuju kao atributi čvorova. Linije predstavljaju linearne oblike, poput cesta ili rijeka, a iste se koriste za modeliranje povezanosti između različitih točaka te su definirane nizom točaka koje opisuju njihovu geometriju. Definicija zatvorenih geometrijskih oblika kao što su granice zemalja, gradova ili zona nazivaju se poligoni koji su niz točaka te zajedno čine oblik na geografskoj karti.

Prostorni atributi dijele se na koordinate (npr. latituda i longituda) i pohranjuju se kao svojstva čvorova ili veza u grafu, omogućujući precizno lociranje entiteta na mapi. Također, u prostorne attribute pripada *bounding box* (ograničavajući pravokutnik) koji definira minimalnu i maksimalnu latitudu i longitudu, a često je korišten za pretraživanje unutar određenog geografskog područja. Analiza putanja omogućuje pronalaženje najkraće rute između dvije ili više točaka na temelju geolokacijskih podataka i veza u grafu, uzimajući pri tome u obzir udaljenosti ili druge kriterije. Također u analizu putanja pripada i analiza mreža koja se koristi za analizu složenih mreža, poput prometnih sustava, kako bi se identificirale optimalne rute, prometni čepovi ili kritične točke unutar mreže [1,5-6].

Na slici 2.3. prikazan je primjer prostornog upita koji pronalazi sve lokacije unutar 1000 metara jedne od druge koristeći pohranjene geolokacijske podatke.

```
MATCH (a:Location), (b:Location)
WHERE distance(a.point, b.point) < 1000
RETURN a, b;
```

Slika 2.3. Primjer prostornog upita u Neo4j

Topološka analiza u geoanalizi pruža ključne alate za razumijevanje i upravljanje prostornim podacima u geoanalizi. Njena primjena omogućuje dublje razumijevanje složenih prostorno-vremenskih odnosa, olakšava donošenje informiranih odluka u planiranju i upravljanju prostorom, te poboljšava točnost i konzistentnost geoinformacijskih sustava. Ključne koncepte topološke analize u geoanalizi prepoznajemo prema niže spomenutom.

Identifikacija susjedstva je funkcionalnost koja omogućuje identifikaciju čvorova koji dijele određene prostorne odnose, kao što su susjedstvo ili povezanost putem specifičnih geometrijskih oblika. Npr, u GIS-u, analiza susjedstva može pomoći u identifikaciji koje zemlje ili regije graniče jedna s drugom.

Zatim, povezanost koja se odnosi na sposobnost putovanja ili prijenosa između različitih dijelova prostorne mreže. Primjer topološke povezanosti je mreža cesta gdje povezanost ceste s drugom cestom omogućuje kretanje vozila između točaka.

Zatvorenost je činjenica da određena površina može biti potpuno okružena granicom. U kartografiji i GIS-u, zatvorenost je važna za identifikaciju zatvorenih poligona, poput jezera, parkova ili parcela zemljišta dok se kontinuitet odnosi na neprekinutost geometrijskih ili prostornih objekata. Također, u topološkoj analizi kontinuitet može pomoći u određivanju hoće li se put ili rijeka nastaviti bez prekida.

Unutrašnjost, vanjština i granica su svojstva koja definiraju prostorne odnose gdje se svaki objekt može podijeliti na unutarnje područje (unutar granica), vanjsko područje (izvan granica) i granicu koja ih razdvaja. U geoanalizi, ovo je ključno za analizu prostorne raspodjele objekata, a postoji još i koncept klasterizacije koja omogućuje grupiranje čvorova na temelju njihovih geolokacijskih atributa što je korisno za identificiranje zona sličnih karakteristika ili za analizu prostorne distribucije.

Primjena topološke analize u geoanalizi dijeli se na planiranje urbanih područja gdje topološka analiza može pomoći u razumijevanju odnosa između različitih urbanih elemenata kao što su zgrade, ceste, parkovi i infrastruktura. Može se koristiti za analizu povezanosti različitih dijelova grada ili povezanosti različitih zona unutar grada s prometnim mrežama dok u ekološkim studijama topološka analiza pomaže u razumijevanju povezanosti ekosustava, staništa i migracijskih puteva životinja. Analiza povezanosti šumskih područja može pomoći u očuvanju staništa i planiranju koridora za divlje životinje [7].

Graf baza podataka omogućuje izvršavanje različitih operacija nad geolokacijskim podacima kao što su dodavanje i ažuriranje geolokacijskih Promjene u geografskim informacijama kao što su promjene lokacija ili oblikovanja poligona također se mogu ažurirati izravno u grafu.

Prostorno pridruživanje (*spatial join*) omogućuje kombiniranje geolokacijskih podataka iz različitih izvora na temelju njihove prostorne blizine ili relacije što je korisno za složene analize poput pridruživanja demografskih podataka geografskim zonama i sl. Graf baze podataka često se koriste u kombinaciji s alatima za vizualizaciju kako bi se prostorni podaci prikazali na karti time omogućujući intuitivniju analizu i donošenje odluka.

2.3. Alati

Neke od poznatih graf baza podataka, osim Neo4j koji je korišten u ovom radu, niže su opisane ključnim značajkama i njihovim primjenama [2-6].

Amazon Neptune



Slika 2.4. Amazon Neptune logo

Potpuno upravljana graf baza podataka u oblaku koja podržava nekoliko modela grafova, uključujući *RDF (Resource Description Framework)* i *Property Graph*. Neptune podržava geolokacijske podatke putem svojih ugrađenih funkcionalnosti i integracija s drugim Amazonovim alatima. Na slici 2.4. prikazan je logo Amazon Neptune.

Ključne značajke ove graf baze podataka su:

- Podrška za standarde - podržava *SPARQL* za *RDF* modele i *Gremlin* za *Property Graph* modele.
- Integracija s *AWS* alatima - laka integracija s *Amazon Location Service* i drugim *AWS* servisima za geolokaciju.
- Skalabilnost - namijenjen je za rad s velikim grafovima i prostornim podacima.

Primjenjuje se u sljedećim granama:

- *E-commerce* - analiza podataka o korisnicima u kontekstu njihove lokacije za ciljani marketing.
- *IoT* - praćenje uređaja i njihovih interakcija unutar određene geografske zone.

ArangoDB



Slika 2.5. ArangoDB logo

ArangoDB je multi-model baza podataka koja podržava dokumente, ključ-vrijednost i graf modele. Nudi prostornu podršku i omogućuje kompleksne prostorne upite unutar graf baze. Na slici 2.5. prikazan je ArangoDB logo.

Ključne značajke:

- Prostorni upiti - podrška za GeoJSON i prostorne indekse za brže pretrage.
- Multi-model - fleksibilnost u korištenju različitih tipova podataka unutar iste baze.
- *AQL (ArangoDB Query Language)* - moćan upitni jezik koji podržava prostorne upite.

Primjene:

- Geolokacijski servisi - pronalazak i prikaz podataka temeljenih na korisničkoj lokaciji.
- Pametna infrastruktura - praćenje i analiza podataka u stvarnom vremenu iz pametnih gradova i uređaja.

TigerGraph



Slika 2.6. TigerGraph logo

TigerGraph je graf baza podataka dizajnirana za velike količine podataka i visoko performansne analize. Podržava napredne analitičke upite, uključujući one koji se odnose na geolokacijske podatke. Slika 2.6. prikazuje logo TigerGraph baze podataka.

Ključne značajke:

- Distribuirana arhitektura - omogućuje rad s vrlo velikim grafovima koji uključuju prostorne podatke.
- *GSQL* - upitni jezik specifičan za TigerGraph, koji podržava složene graf operacije.
- Ugrađeni algoritmi - algoritmi za analizu mreža, poput najkraćeg puta, *PageRank*, itd.

Primjene:

- Financijska industrija - praćenje i analiza transakcija s obzirom na geolokaciju za otkrivanje prijevara.
- Telekomunikacije - analiza mreža i performansi s obzirom na geografske zone.

OrientDB



Slika 2.7. OrientDB logo

OrientDB je *open-source multi-model* baza podataka koja podržava grafove, dokumente, ključ-vrijednost i objekte. Podržava rad s geolokacijskim podacima kroz prostorne funkcionalnosti. Logo OrientDB je prikazan na slici 2.7.

Ključne značajke:

- Podrška za *SQL-like* upite - uključuje prostorne funkcije i indeksiranje.
- Visoka performansa - optimiziran za brze pretrage i rad s velikim podacima.
- Podrška za *GeoSpatial* - mogućnost izvođenja prostornih upita i analiza.

Primjene:

- Pametni gradovi - praćenje i upravljanje resursima unutar određenih geografskih područja.
- *Retail* - optimizacija lanca opskrbe na temelju geografskih podataka.

JanusGraph



Slika 2.8. JanusGraph logo

JanusGraph je *open-source* distribuirana graf baza podataka koja se integrira s različitim sustavima za pohranu podataka kao što su *HBase*, *Cassandra*, i *Elasticsearch*. Podržava prostorne podatke i omogućuje složene analize. Na slici 2.8. prikazan je JanusGraph logo.

Njezine ključne značajke koje možemo izdvojiti su:

- Distribuiranost - omogućuje rad s velikim grafovima koji su prostorno raspoređeni.
- *Gremlin* upitni jezik - podržava prostorne upite kroz *Gremlin DSL (Domain-Specific Language)*.
- Integracija - relativno laka s *Elasticsearchom* za poboljšane prostorne pretrage.

Primjene:

- Geoinformacijski sustavi (GIS) - slojevite analize podataka na temelju geografske lokacije.
- Energetika - praćenje i analiza infrastrukture na temelju geografskih podataka.

Navedeni alati omogućuju različite pristupe radu s geolokacijskim podacima unutar graf baza podataka pružajući podršku za složene prostorne analize i vizualizacije. Izbor odgovarajuće tehnologije ovisi o specifičnim potrebama aplikacije, skalabilnosti, performansama, i kompleksnosti upita koje treba izvršavati.

2.4. Izazovi i ograničenja

Iako geoanaliza i graf baze podataka imaju puno prednosti potrebno je spomenuti i određene izazove te ograničenja kao što su [2-6]:

- Performanse i skalabilnost - iako su graf baze podataka vrlo učinkovite za rad sa složenim zahtjevima, performanse mogu igrati važnu ulogu kada se radi s izuzetno velikim grafovima

(npr. milijardama čvorova i veza) što može dovesti do problema s vremenom odziva i zahtijevati optimizaciju baze podataka.

- Složenost upita - pisanje složenih upita u grafičkim bazama može biti izazovno, posebno za korisnike koji su naviknuti na SQL i relacijske baze podataka. Ovakvi upiti često zahtijevaju drugačiji način razmišljanja i poznavanje specifičnih alata ili jezika (npr. Cypher za Neo4j).
- Nedostatak standardizacije - relacijske baze podataka imaju dobro razvijene standarde (npr. SQL), dok graf baze još uvijek nemaju univerzalno prihvaćene standarde, što može otežati prelazak između različitih grafičkih sustava ili njihovu integraciju s drugim tehnologijama.
- Ograničena podrška za kompleksne transakcije - u usporedbi s relacijskim bazama, graf baze podataka mogu imati ograničenu podršku za složene transakcije, što može predstavljati problem u aplikacijama gdje su potrebne konzistentnost i integritet podataka.
- Trošak i resursi - implementacija i održavanje graf baze podataka može biti skupa i zahtijevati specijalizirane resurse, uključujući znanje i alate. Također, neke graf baze podataka mogu biti dosta zahtjevne u pogledu memorije i prostora na disku, a osobito kod velikih grafova.

Unatoč navedenim izazovima, kontinuirani razvoj tehnologije i poboljšanja u graf bazama podataka otvaraju nove mogućnosti za njihovu primjenu u geolokacijskoj analizi čineći ih ključnim alatom u mnogim suvremenim industrijama i znanostima.

2.5. Budući trendovi i mogućnost poboljšanja

- Dubinska integracija s *AI* i *ML* tehnologijama - budućnost geoanalize u graf bazama podataka leži u integraciji s umjetnom inteligencijom (*AI*) i strojnim učenjem (*ML*). *AI* može poboljšati prediktivne analize, omogućiti automatsko otkrivanje obrazaca u prostornim podacima i olakšati donošenje odluka u stvarnom vremenu.
- Povećanje skalabilnosti i performansi - kako količina prostornih podataka eksponencijalno raste, graf baze podataka će morati ponuditi još bolju skalabilnost i performanse. Ovo će zahtijevati napredne distribucijske arhitekture, bolje algoritme za paralelnu obradu, i učinkovitije indeksiranje prostornih podataka.
- Razvoj specifičnih algoritama za geoanalizu - postoji potreba za razvojem novih, specijaliziranih algoritama koji su optimizirani za rad s geolokacijskim podacima unutar grafova. Ovi algoritmi će omogućiti učinkovitiju analizu velikih prostornih grafova, poput

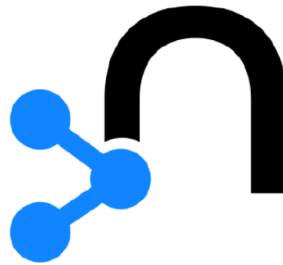
analiza klasterizacije, najkraćih puteva u složenim mrežama i pretraga unutar geografskih regija.

- Povećanje interoperabilnosti - kako bi graf baze podataka postale još korisnije u geoanalizi, povećanje interoperabilnosti s drugim GIS alatima i tehnologijama, kao što su *QGIS* ili *ArcGIS*, bit će ključno jer će omogućiti lakšu razmjenu podataka i kombiniranje analiza iz različitih izvora.
- Razvoj naprednih vizualizacijskih alata - vizualizacija prostornih podataka u graf bazama podataka trenutno je osnovna ali budućnost će donijeti napredne vizualizacijske alate koji će omogućiti interaktivne, *real-time* prikaze podataka, pomažući korisnicima da intuitivnije razumiju prostorne odnose i obrasce.
- Povećanje sigurnosti i privatnosti - s obzirom na sve veću količinu prikupljenih geolokacijskih podataka, sigurnost i privatnost postat će ključni aspekti. U budućnosti će graf baze podataka morati ponuditi napredna rješenja za zaštitu podataka, uključujući anonimizaciju podataka, kontrolu pristupa i enkripciju [8].

3. KORIŠTENI ALATI I TEHNOLOGIJE

U ovom dijelu navedeni su i opisani alati i tehnologije korišteni za potrebe ovog rada kako bi se napravili upiti na bazu naziva *Public-Transport-SP-Graph-Database* [9-10].

3.1. Neo4j



Slika 3.1. Neo4j logo

Razvoj Neo4j-a započeo je sredinom 2000-ih, kada je suosnivač Emil Eifrem zajedno sa svojim timom, prepoznao potrebu za bazom podataka koja bi omogućila jednostavnije i brže praćenje te analizu odnosa među podacima. Njihov glavni cilj bio je riješiti izazove s kojima su se suočili u poslovnim aplikacijama gdje su tradicionalne relacijske baze podataka bile neučinkovite u upravljanju povezanim podacima. Relacijski modeli prisiljavali su programere da stalno rade složene upite (*JOIN*-ove) kako bi povezali podatke što je postajalo problematično s obzirom na skaliranje i složenost podataka. Slika 3.1. prikazuje logo Neo4j.

Jedan od glavnih izazova s kojima se razvojni tim suočio u ranijim fazama razvoja bila je skalabilnost. Kako su se baze počele povećavati, potreba za horizontalnom skalabilnošću postala je ključna što je dovelo do uvođenja distribuiranih tehnologija. To je bio važan korak zbog velikih korporacija koje se oslanjaju na Neo4j za rad s velikim skupovima podataka [9,10-11].

Kroz svoju povijest Neo4j je prošao nekoliko faza razvoja, a neke od ključnih su:

- Prva komercijalna verzija - Neo4j baza podataka objavljena je 2007. godine. Time je Neo4j postao prva industrijski prepoznata graf baza podataka koja je pružala stabilno, skalabilno rješenje za složene odnose u podacima.
- Proširenje funkcionalnosti - dogodilo se tijekom godina kada je Neo4j proširio svoje mogućnosti uvođenjem novih funkcionalnosti, kao što su napredni algoritmi za analizu grafova (npr. pronalaženje najkraćeg puta, središnjosti u mreži), proširena podrška za

distribuirane sustave i integracija s alatima za analizu podataka kao što su Apache Spark i Elasticsearch.

- *Open-source* pristup - Neo4j zajednica i dalje raste zahvaljujući open-source modelu, omogućujući istraživačima i programerima diljem svijeta da koriste, prilagođavaju i doprinose razvoju baze podataka. Verzija *Neo4j Community* omogućuje slobodan pristup bazi podataka, dok komercijalna verzija nudi napredne funkcionalnosti za poduzeća s većim potrebama za skalabilnošću i podrškom.

Neo4j je implementiran kao nativna graf baza podataka što znači da je njegova arhitektura optimizirana za rad s grafovima od početka. Koristi algoritme specifične za grafove, kao što su Dijkstra-ov algoritam za pronalaženje najkraćih putova te podržava jednostavno proširenje kroz dodatne funkcionalnosti. Nativni mehanizam za pohranu podataka omogućuje brzo dohvaćanje i izvršavanje upita što ga čini bržim i efikasnijim u usporedbi s ostalim bazama koje simuliraju grafove preko relacijskih struktura.

Također nudi bogatu ekosustavsku podršku kroz različite *API*-je i razvojne alate, što ga čini pristupačnim za različite vrste korisnika, od istraživača podataka do programera aplikacija [12].

3.2. Upitni jezik Cypher

Cypher je deklarativni jezik za upite u Neo4j-u, specijaliziran za rad s graf bazama podataka. Dizajniran je kako bi olakšao interakciju s grafovima koristeći sintaksu koja omogućuje intuitivno modeliranje čvorova (*nodes*), veza (relacija) i njihovih atributa. Kao ključni aspekt Neo4j baze podataka, Cypher omogućuje korisnicima da efikasno pretražuju i manipuliraju podacima, čineći ga lakim za učenje i upotrebu čak i za korisnike koji dolaze iz okruženja relacijskih baza podataka.

Cypher sintaksa temelji se na jednostavnim i čitljivim izrazima koji omogućuju brzo modeliranje i pretragu grafova [9-10].

Osnovni elementi Cypher jezika su:

- Čvorovi (*nodes*) - predstavljaju entitete u grafu kao što su osobe, proizvodi ili mjesta.
- Veze (*relationships*) - spajaju čvorove i definiraju odnose među entitetima kao što su prijateljstva, transakcije, vlasništvo ili udaljenosti.
- Svojstva (*properties*) - oba čvora i veze mogu imati pridružena svojstva koja pružaju dodatne informacije kao što su imena, datumi ili vrijednosti.

Cypher je jedan od najpopularnijih jezika za upite u graf bazama podataka zbog svoje jednostavnosti, a glavne prednosti Cypher jezika uključuju:

- Intuitivna sintaksa - koristi obrasce koji odražavaju prirodni način razmišljanja o podacima i njihovim odnosima, što ga čini jednostavnim za učenje i korištenje.
- Deklarativni stil - omogućuje korisnicima da definiraju željene rezultate, dok sustav sam izvršava potrebne korake u pozadini, olakšavajući manipulaciju kompleksnim grafovima.
- Visoka efikasnost - upiti su optimizirani za rad s grafovima, omogućujući brzo dohvaćanje podataka čak i u velikim i složenim mrežama.

U narednim primjerima prikazane su osnovne operacije u Cypher jeziku.

Na slici 3.2. prikazan je jednostavan Cypher upit koji pronalazi sve osobe u bazi podataka.

```
MATCH (p:Person)
RETURN p.name;
```

Slika 3.2. Jednostavan Cypher upit

Ovdje se koristi ključna riječ **MATCH** kako bi se specificirao obrazac pretrage grafova.

Oznaka (*p*) označava čvor pod oznakom *Person*, a **RETURN** vraća ime svake osobe iz baze.

Jedna od glavnih prednosti Cypher jezika je njegova sposobnost pretraživanja odnosa između entiteta. U tradicionalnim relacijskim bazama podataka, pretraga veza između entiteta često zahtijeva složene SQL upite sa spajanjima (*JOIN*) dok Cypher koristi jednostavne obrasce za definiranje tih veza.

Na primjer, kako bi se pronašli svi prijatelji određene osobe, Cypher upit može izgledati kao na slici 3.3.:

```
MATCH (p:Person)-[:FRIEND_OF]->(friend:Person)
WHERE p.name = "John"
RETURN friend.name;
```

Slika 3.3. Cypher upit za pronalazak prijatelja

Ovdje se koristi obrazac $(p)-[]->(friend)$ koji prikazuje čvor osobe p povezan vezom $FRIEND_OF$ s čvorom drugog entiteta, označenog kao $friend$. Ovaj način specifikacije odnosa omogućuje intuitivno i brzo dohvaćanje podataka iz kompleksnih mreža.

Umetanje podataka u Neo4j putem Cypher jezika također je jednostavno i intuitivno. Umjesto rada s tablicama, korisnici definiraju čvorove i veze koristeći čitljivu i jasnu sintaksu. Slika 3.4. prikazuje umetanje novog čvora koji predstavlja osobu i dodavanje veze.

```
CREATE (p1:Person {name: "Alice", age: 30}),
      (p2:Person {name: "Bob", age: 35}),
      (p1)-[:FRIEND_OF]->(p2);
```

Slika 3.4. Umetanje podataka koristeći Cypher

Ovaj upit kreira dva čvora označena kao *Person* s odgovarajućim svojstvima (ime i dob) te stvara vezu $FRIEND_OF$ između njih. Cypher upotrebljava deklarativni stil gdje korisnici definiraju što žele postići dok sustav sam izvršava potrebne operacije u pozadini.

Ključna riječ *SET* koristi se za mijenjanje svojstava čvorova ili veza, odnosno, ažuriranje podataka. Da bi se promijenila dob osobe *Alice*, Cypher upit bi izgledao kao na slici 3.5.:

```
MATCH (p:Person {name: "Alice"})
SET p.age = 31;
```

Slika 3.5. Ažuriranje podataka u Cypher jeziku

Upit prvo pronalazi čvor koji predstavlja osobu s imenom *Alice*, a zatim ažurira svojstvo *age* na novu vrijednost.

Za brisanje podataka, Cypher nudi naredbu *DELETE* koja se može koristiti za uklanjanje čvorova, veza ili oboje. Upit sa slike 3.6. pronalazi i uklanja vezu $FRIEND_OF$ između osobe *Alice* i osobe *Bob*.

```
MATCH (p1:Person {name: "Alice"})-[r:FRIEND_OF]->(p2:Person {name: "Bob"})
DELETE r;
```

Slika 3.6. Brisanje podataka u Cypher jeziku

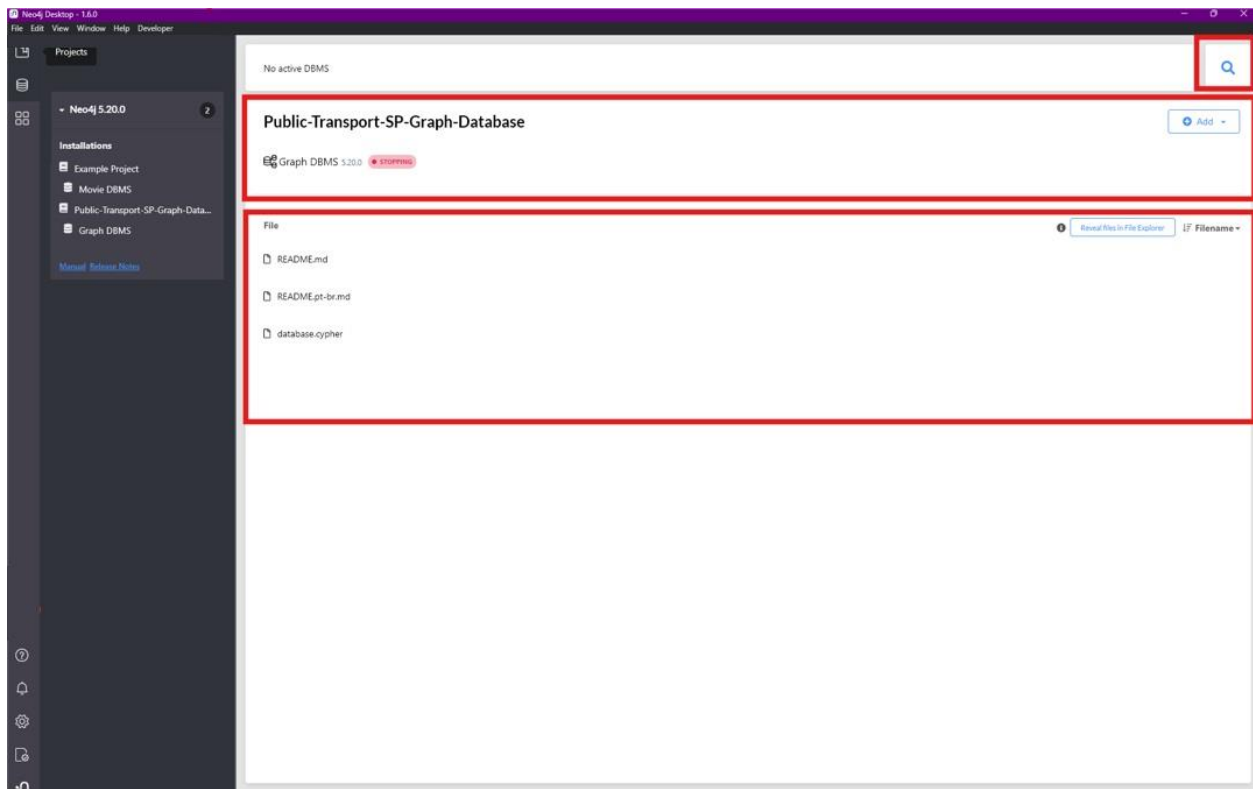
Cypher također podržava napredne funkcionalnosti kao što su agregacija podataka, sortiranje i grupiranje rezultata. Na primjer, kako bi se dobio prosjek starosti svih osoba u bazi podataka, korisnik može izvršiti sljedeći upit prikazan na slici 3.7.:

```
MATCH (p:Person)
RETURN avg(p.age);
```

Slika 3.7. Prosjek starosti osoba u Cypher jeziku

Upit sa slike iznad koristi agregacijsku funkciju `avg()` kako bi vratio prosječnu dob svih osoba u bazi podataka. Također, podržava i druge agregacijske funkcije poput `COUNT`, `SUM`, `MAX`, `MIN`, a koje omogućuju napredne statističke upite i analize.

3.3. Neo4j desktop



Slika 3.8. Prikaz sučelja Neo4j

Na slici 3.8. prikazano je sučelje Neo4j-a u kojemu je označena odabrana baza *Public-Transport-SP-Graph-Database* što označava aktivni projekt korisnika. Ovaj panel omogućuje korisniku pregled i odabir različitih instalacija i projekata s kojima želi raditi [9-13].

Na slici su navedene i datoteke:

- *README.md* - obično sadrži osnovne informacije o projektu i upute.
- *database.cypher* - sadrži Cypher upite, specifične za graf baze podataka Neo4j.

U Neo4j desktopu postoje tri različite opcije za dodavanje, odnosno, za kreiranje projekata koje su označene crvenim okvirom na slici. Ove opcije nalaze se u izborniku *File*, a omogućuju korisnicima uvoz projekata iz različitih izvora ili dodavanje datoteka unutar postojećih projekata:

- *Add file(s) to Project* - dodavanje datoteka u projekt je opcija koja omogućava korisnicima da dodaju pojedinačne datoteke u već postojeći projekt. Korisna je kada korisnik ima specifične datoteke (kao što su *.cypher* datoteke s upitima, konfiguracijske datoteke ili druge relevantne dokumente) koje treba uključiti u projekt.

Tipična upotreba obuhvaća dodavanje skripti, *README* datoteka ili drugih resursa koji su potrebni za rad na projektu.

- *Create project from directory* - kreiranje projekta iz direktorija je opcija koja omogućava korisnicima da kreiraju novi projekt na temelju cjelokupnog direktorija na svom računalu. Umjesto da dodaju pojedinačne datoteke, korisnici mogu odabrati cijelu mapu koja sadrži sve potrebne datoteke i strukture za projekt.

Tipična upotreba obuhvaća uvoz postojećih projekata s lokalnog diska koji su već organizirani u strukturu mapa.

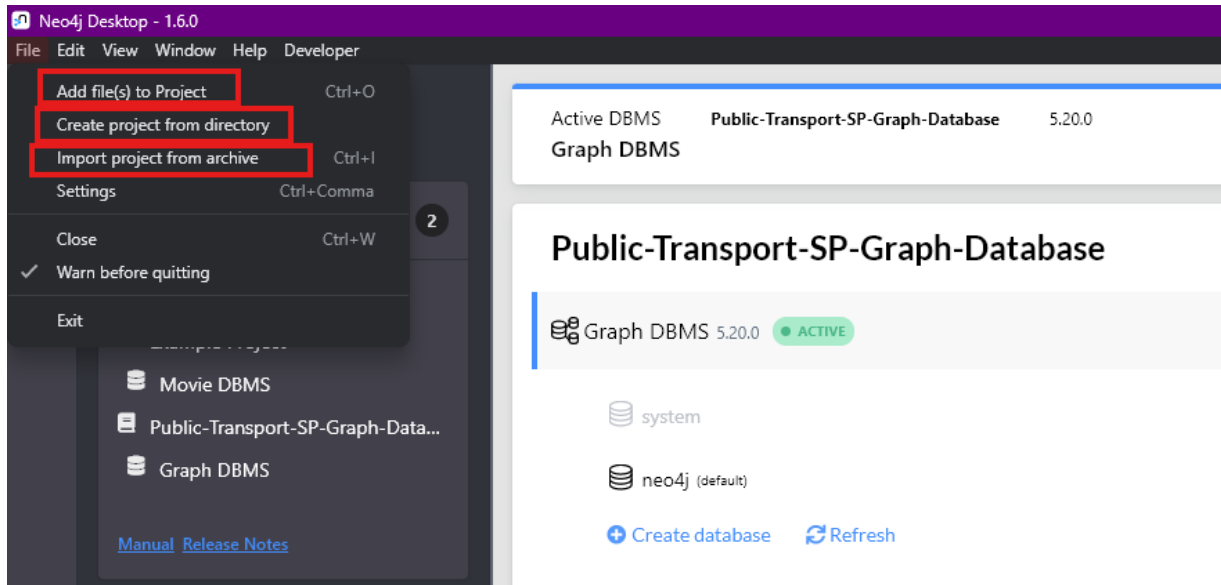
- *Import project from archive* - uvoz projekta iz arhive je opcija koja omogućuje korisnicima da uvezu projekt iz arhivske datoteke, obično u formatu *.zip* ili sličnom. Ako je projekt prethodno bio arhiviran, može se koristiti ova opcija kako bi se cijeli projekt učitao iz arhive.

Tipična upotreba obuhvaća uvoz projekata preuzetih s interneta ili onih koji su ranije arhivirani radi sigurnosne kopije ili dijeljenja s drugim korisnicima.

Ove tri opcije omogućuju fleksibilan pristup uvozu i kreiranju projekata u Neo4j Desktopu. Ovisno o korisničkim potrebama mogu se dodavati pojedinačne datoteke, kreirati projekti iz direktorija ili uvoziti cijeli projekt iz arhiva. Ove funkcionalnosti omogućuju jednostavno upravljanje različitim

projektima u Neo4j okruženju i olakšavaju korisnicima rad s podacima u grafičkim bazama podataka.

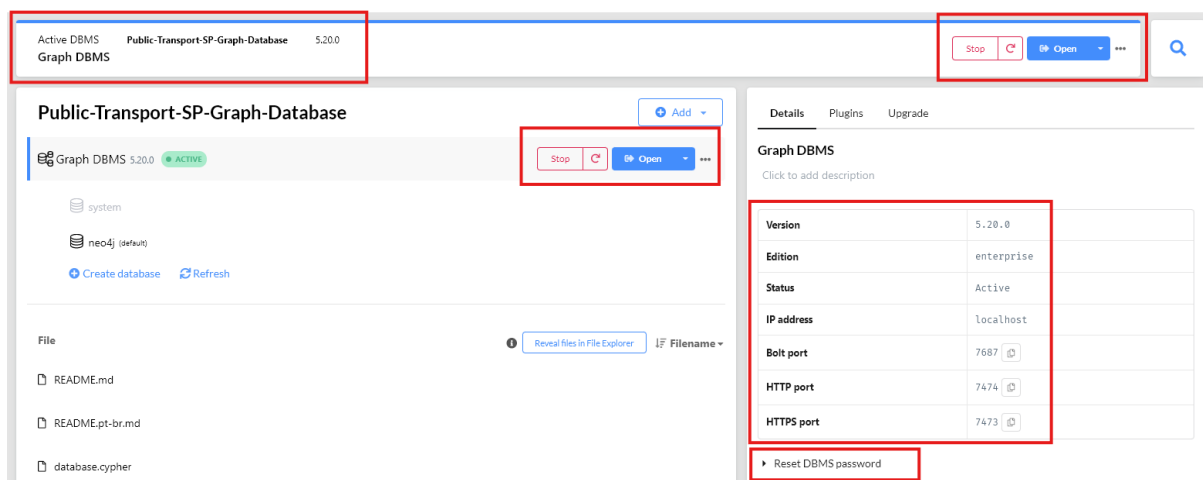
Slika 3.9. prikazuje kreiranje i uvoz projekta.



Slika 3.9. Kreiranje/uvoz projekta u Neo4j

Iz razloga što je rađeno na postojećem projektu, opcija "Create project from directory" je idealna. Omogućuje brz uvoz cijelog projekta u Neo4j Desktop uključujući sve potrebne datoteke i strukture iz postojećeg direktorija.

Na ovaj način mogu se brzo učitati sve potrebne komponente projekta te se odmah fokusirati na analizu podataka i rad bez dodatnog ručnog prebacivanja pojedinačnih datoteka. Na slici 3.10. prikazani su detalji uvoza projekta.



Slika 3.10. Uvoz projekta u Neo4j

4. EKSPERIMENTALNI REZULTATI

U ovom dijelu prikazani su različiti upiti na bazi podataka kako bi dobili što detaljnije informacije te prikazali iste na grafu.

4.1. Izračun najkraćih puteva

```
MATCH (start:MetroStation {name: "Butantã"}, (end:MetroStation {name: "Paulista"}))
MATCH path = shortestPath((start)-[:CONNECT*]-(end))
RETURN path;
```

Slika 4.1. Upit za najkraći put

Na slici 4.1. prikazan je upit za najkraći put između dvije stanice, a slijedi opis samog upita:

MATCH (start{name: "Butantã"}, (end{name: "Paulista"})):

- Ovdje tražimo dva čvora (stanice) u bazi podataka.
 - start označava stanicu Butantã, koja je tipa MetroStation.
 - end označava stanicu Paulista, također tipa MetroStation.
- Ova naredba traži čvorove u bazi koji imaju oznaku (labelu) MetroStation i svojstvo name jednako zadanim vrijednostima ("Butantã" i "Paulista").

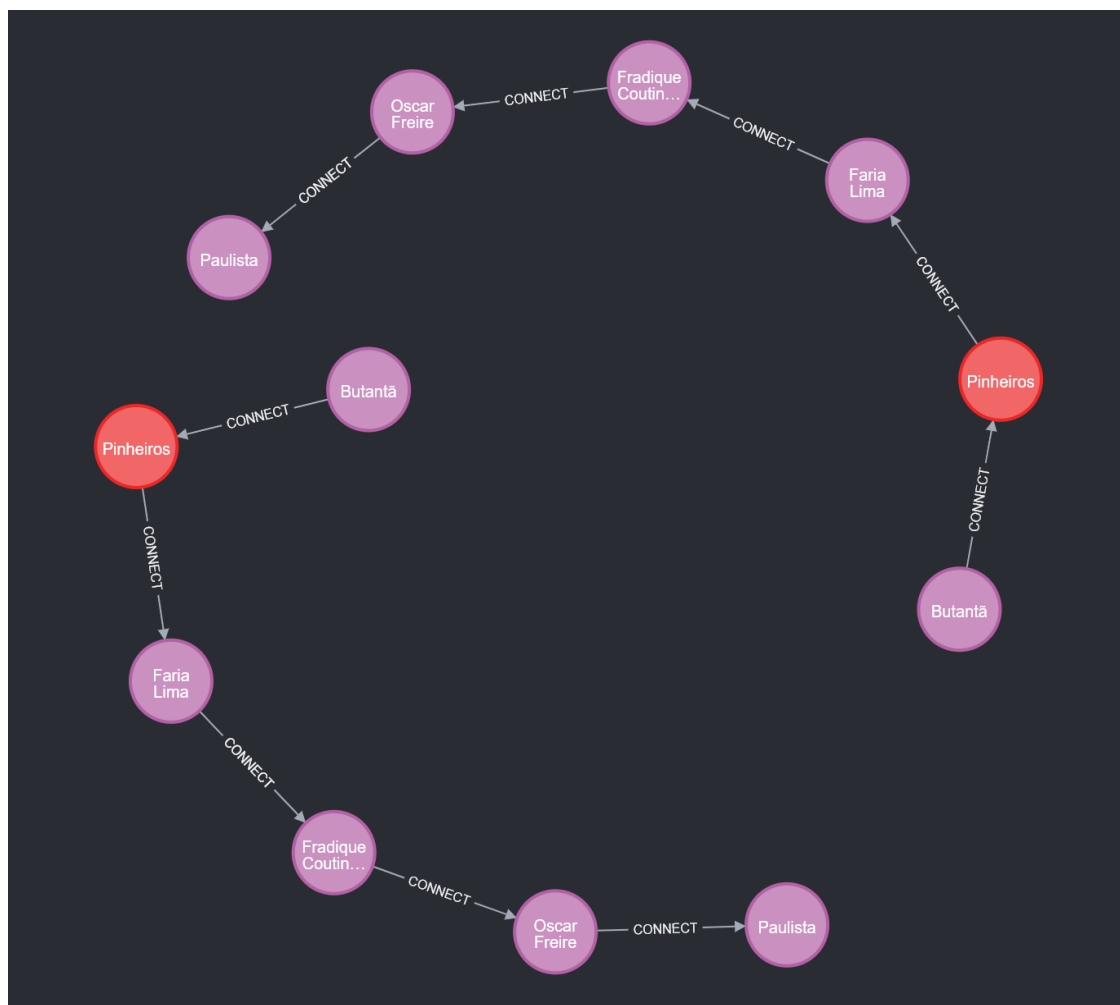
**MATCH path = shortestPath((start)-[]-(end))*:*

- Ovdje tražimo najkraći put između dvije stanice, *start* i *end*.
- *shortestPath* je Cypher funkcija koja vraća najkraći put između dva čvora u grafu.
 - *(start)-[]-(end)** označava sve moguće rute između *start* i *end* koristeći relaciju *CONNECT*, koja povezuje stanice u grafu.
 - Zvezdica (*), koja dolazi nakon [], označava da može biti više uzastopnih veza između tih stanica. Dakle, pretražuju se svi putevi koji povezuju te stanice koristeći *CONNECT* relacije bez obzira na broj veza (može biti jedan ili više koraka).
- Funkcija *shortestPath* automatski pronalazi put s najmanjim brojem veza (koraka) između stanica.

RETURN path:

- Ova naredba vraća varijablu *path*, koja sadrži najkraći put između dvije stanice.
- *path* je cijela ruta koja uključuje čvorove (stanice) i veze (*CONNECT* relacije) koje su uključene u najkraći put.

Ovaj upit traži najkraći put između stanica Butantã i Paulista. Funkcija *shortestPath* pretražuje mrežu transportnih veza između stanica i vraća put s najmanjim brojem koraka (veza). Ovaj upit će vam dati pregled stanica kroz koje prolazi najkraća ruta između dviju metro stanica kako je prikazano na slici 4.2.



Slika 4.2. Prikaz najkraćeg puta između dvije stanice

4.2. Prikaz podataka u ovisnosti o zadanim parametrima

```
MATCH
  (s1 {name: "Pinheiros"}),
  (s2 {name: "Grajaú"}),
  p = shortestPath((s1)-[:CONNECT*]-(s2))
WHERE ALL (x IN RELATIONSHIPS(p) WHERE x.transport = 'train' OR x.transport = 'metro' OR x.transport = 'bus')
RETURN p;
```

Slika 4.3. Prikaz najkraćeg puta između dvije stanice koristeći bus, metro ili vlak

Na slici 4.3. prikazan je najkraći put između dvije stanice koristeći bus, metro ili vlak, a obrazloženje upita slijedi.

MATCH (*start*{name: "Butantã"}, (*end*{name: "Paulista"}):

- Ovdje tražimo dva čvora (stanice) u bazi podataka.
 - *start* označava stanicu Butantã, koja je tipa *MetroStation*.
 - *end* označava stanicu Paulista, također tipa *MetroStation*.
- Ova naredba traži čvorove u bazi koji imaju oznaku (labelu) *MetroStation* i svojstvo *name* jednako zadanim vrijednostima ("Butantã" i "Paulista").

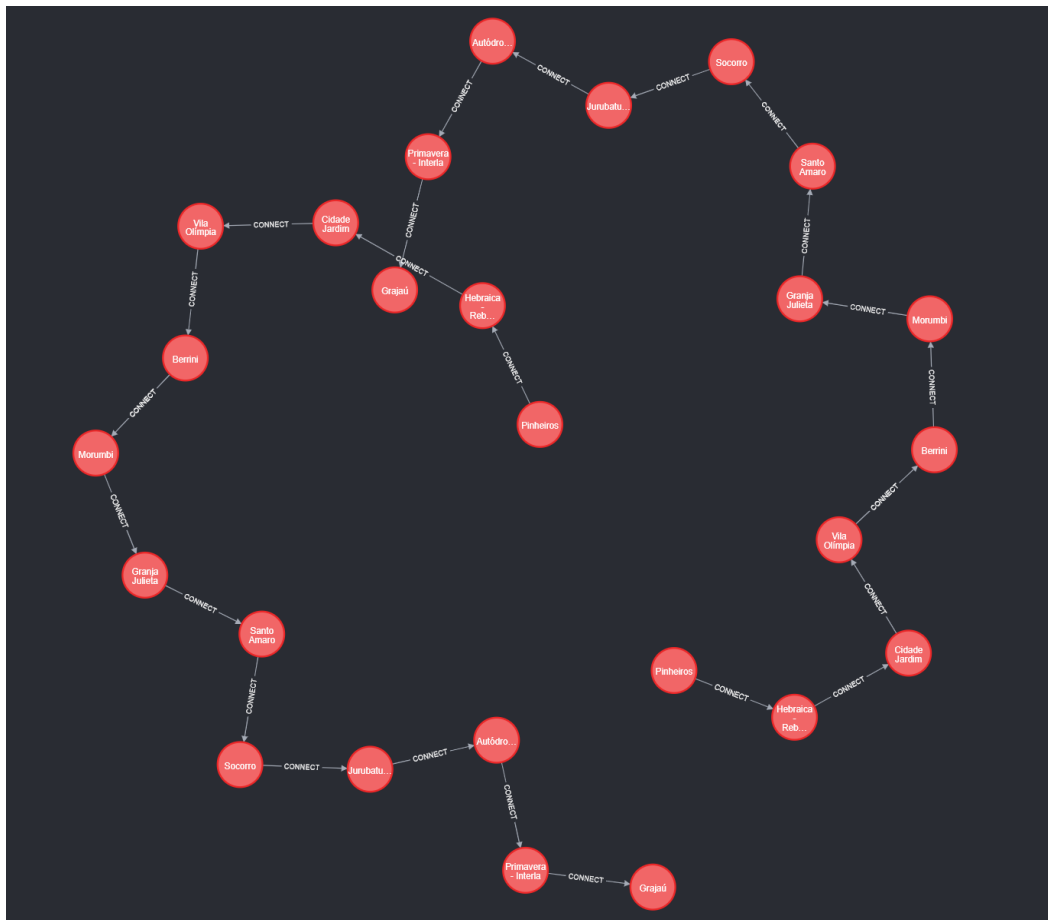
MATCH path = shortestPath((start)-[]-(end)):

- Ovdje tražimo najkraći put između dvije stanice, *start* i *end*.
- *shortestPath* je Cypher funkcija koja vraća najkraći put između dva čvora u grafu.
 - (*start*)-[]-(*end*)* označava sve moguće rute između *start* i *end* koristeći relaciju *CONNECT*, koja povezuje stanice u grafu.
 - Zvezdica (*), koja dolazi nakon [], označava da može biti više uzastopnih veza između tih stanica. Dakle, pretražuju se svi putevi koji povezuju te stanice koristeći *CONNECT* relacije bez obzira na broj veza (može biti jedan ili više koraka).
- Funkcija *shortestPath* automatski pronalazi put s najmanjim brojem veza (koraka) između stanica.

RETURN path:

- Ova naredba vraća varijablu *path*, koja sadrži najkraći put između dvije stanice.
- *path* je cijela ruta koja uključuje čvorove (stanice) i veze (*CONNECT* relacije) koje su uključene u najkraći put.

Na slici 4.4. prikazan je najkraći put između stanica koristeći bus, metro ili vlak.



Slika 4.4. Prikaz puta između stanica koristeći bus, metro ili vlak

- Svi besplatni terminali

```
MATCH (station)
WHERE station.isFreeAccess = true
RETURN station.name, labels(station);
```

Slika 4.5. Prikaz svih besplatnih terminala

Slika 4.5. prikazuje sve besplatne terminale, a niže je opisan upit.

MATCH (station):

- Ova naredba traži sve čvorove (stanice) u bazi podataka, bez obzira na njihov tip (metro stanice, autobusni terminali, željezničke stanice itd.).

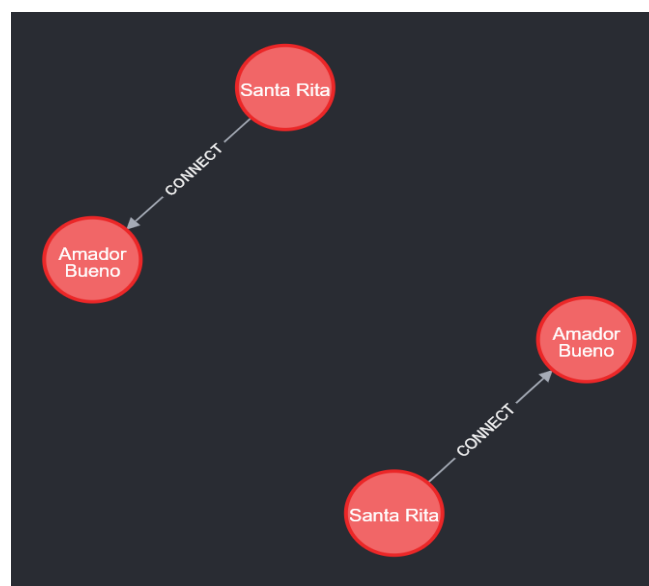
WHERE station.isFreeAccess = true:

- Ovaj uvjet filtrira stanice i vraća samo one koje imaju svojstvo *isFreeAccess* postavljeno na *true*. To znači da će biti vraćene samo one stanice koje imaju besplatan pristup.

RETURN station.name, labels(station):

- *station.name* vraća naziv stanice.
- *labels(station)* vraća oznake (label) stanice

Na slici 4.6. je prikaz svih besplatnih terminala.



Slika 4.6. Svi besplatni terminali

- **Sve stanice koje imaju parking za bicikle**

```
MATCH (station)
WHERE station.hasBikeParkingTerminal = true
RETURN station;
```

Slika 4.7. Sve stanice koje imaju parking za bicikle

Na slici 4.7. prikazan je upit za sve stanice koje imaju parking za bicikle dok je niže i opis samog upita.

MATCH (station):

- Ova naredba traži sve čvorove (stanice) u bazi podataka, bez obzira na njihov tip (metro stanice, autobusni terminali, željezničke stanice itd.).

WHERE station.hasBikeParkingTerminal = true:

- Ovaj uvjet filtrira stanice i vraća samo one koje imaju svojstvo *hasBikeParkingTerminal* postavljeno na *true*. To znači da će biti vraćene samo one stanice koje nude mogućnost parkiranja za bicikle.

RETURN station.name, labels(station):

- *station.name* vraća naziv stanice.
- *labels(station)* vraća oznake (labele) stanice, čime se prikazuje tip stanice, poput *MetroStation*, *BusTerminal*, *TrainStation* itd.

Slika 4.8. prikazuje jedan dio dobivenih rezultata:



Slika 4.8. Prikaz dijela svih stanica koje imaju parking za bicikle

- Sve stanice koje imaju parking za bicikle, a povezane su sa nekom od drugih stanica

```
MATCH (station1)-[:CONNECT]->(station2)
WHERE station1.hasBikeParkingTerminal = true
      AND station2.hasBikeParkingTerminal = true
RETURN station1, station2;
```

Slika 4.9. Upit u Cypher jeziku

Na slici 4.9. prikazan je upit za sve stanice koje imaju parking za bicikle, a povezane su sa nekom od drugih stanica.

Obrazloženje upita:

MATCH (station1)-[]->(station2):

- Ova naredba pronalazi sve parove povezanih stanica koje su povezane relacijom *CONNECT*. *station1* i *station2* su čvorovi koji predstavljaju dvije povezane stanice.

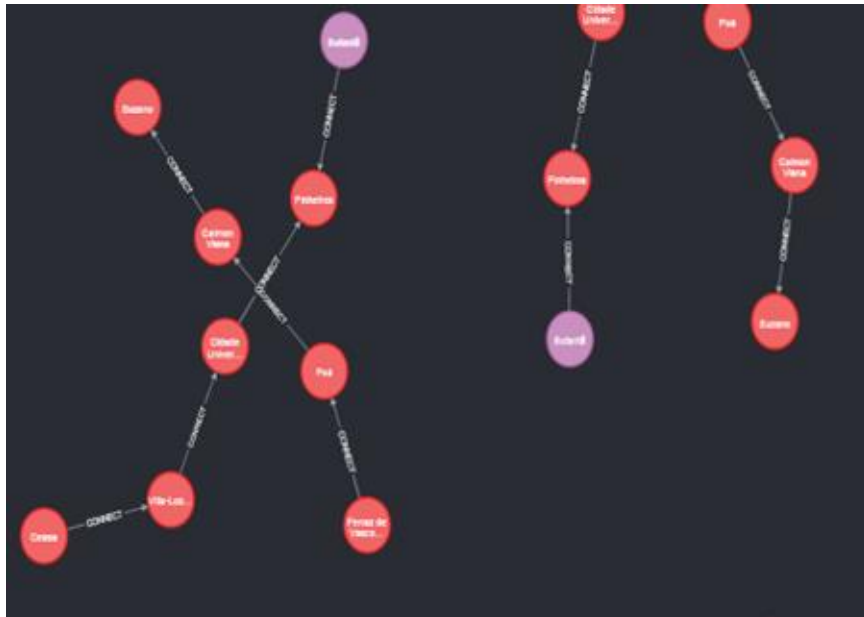
WHERE station1.hasBikeParkingTerminal = true AND station2.hasBikeParkingTerminal = true:

- Ovaj uvjet osigurava da obje stanice u vezi imaju svojstvo *hasBikeParkingTerminal* postavljeno na *true*, što znači da obje stanice moraju imati parking za bicikle.

RETURN station1.name, station2.name:

- Vraća nazive obje stanice koje su povezane i koje imaju parking za bicikle.

Slika 4.10. prikazuje dio svih stanica koje imaju parking za bicikle, a povezane su sa nekom od drugih stanica.



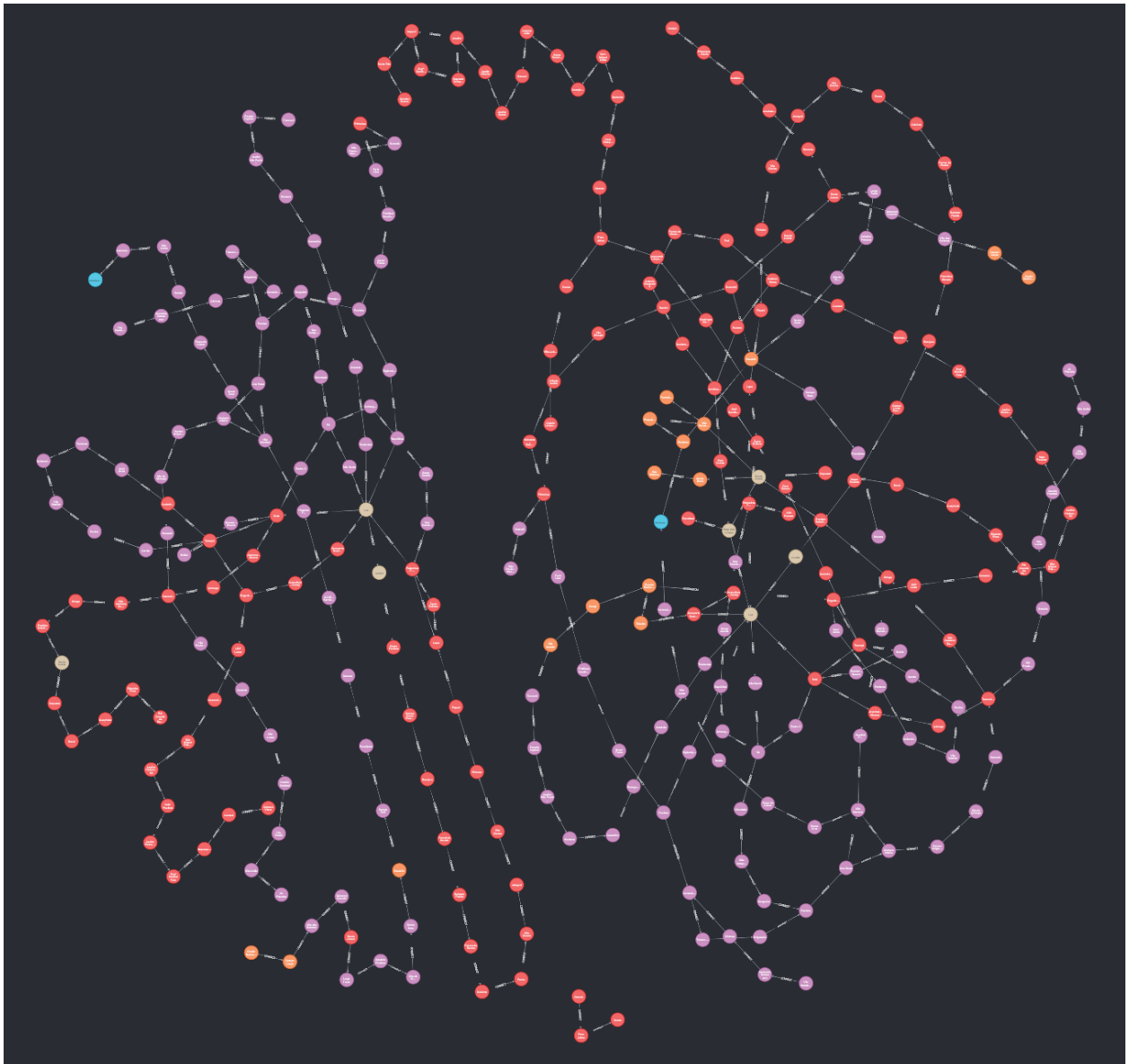
Slika 4.10. Prikaz dijela svih stanica koje imaju parking za bicikle, a povezane su sa nekom od drugih stanica

- Sve stanice

```
MATCH (station)
WHERE station:MetroStation OR station:BusTerminal OR station:TrainStation
RETURN station;
```

Slika 4.11. Upit za prikaz svih stanica

Na slici 4.11. prikazan je upit u Cypher jeziku za dohvat svih stanica iz baze, dok je na slici 4.12. prikaz dobivenih rezultata s uvidom u sve baze.



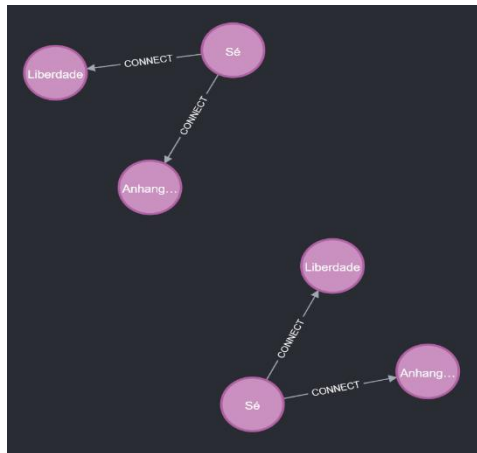
Slika 4.12. Prikaz svih stanica

- Sve stanice povezane sa zadanom stanicom

```
MATCH (m:MetroStation {name: "Sé"})-[:CONNECT]->(connectedStations)
RETURN m, connectedStations;
```

Slika 4.13. Upit za sve stanice povezane sa zadanom stanicom

Slika 4.13. prikazuje upit za sve stanice povezane sa zadanom stanicom, dok se na slici 4.14. prikazuje dobiveni rezultat pretrage.



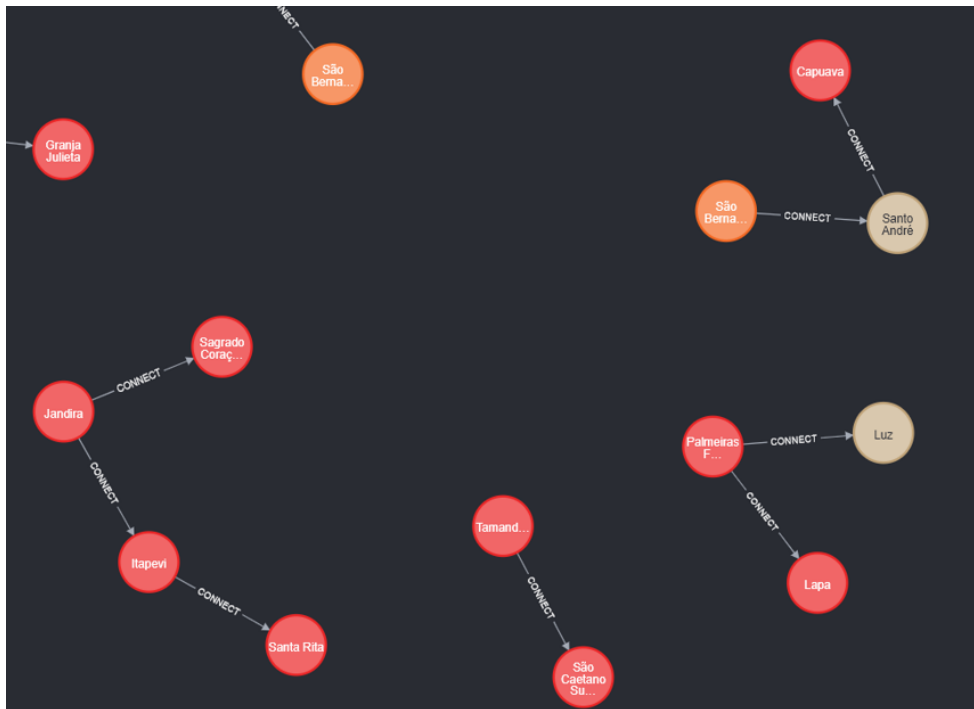
Slika 4.14. Prikaz svih stanica povezanih sa zadanom stanicom

- **Autobusni terminali povezani sa željezničkim stanicama**

Na slici 4.15. nalazi se upit za autobusne terminale koji su povezani željezničkim stanicama.

```
MATCH (bus:BusTerminal)-[:CONNECT]->(train:TrainStation)
RETURN bus, train;
```

Slika 4.15. Prikaz autobusnih terminala povezanih sa željezničkim stanicama



Slika 4.16. Prikaz autobusnih terminala povezanih sa željezničkim stanicama

Slika 4.16. prikazuje autobusne terminale povezane sa željezničkim stanicama.

- **Broj stanica po vrsti (metro, autobus, željeznica)**

Na slici 4.17. nalazi se upit kojim pretražujemo broj stanica po vrsti, te na slici 4.18. prikaz dobivenih.

```

MATCH (station)
WHERE station:MetroStation OR station:BusTerminal OR station:TrainStation
RETURN labels(station) AS StationType, COUNT(station) AS NumberOfStations
ORDER BY NumberOfStations DESC;
  
```

Slika 4.17. Prikaz broja stanica po vrsti

StationType	NumberOfStations
["TrainStation"]	152
["MetroStation"]	138
["BusTerminal"]	20
["TrainStation", "BusTerminal"]	12
["MetroStation", "TrainStation"]	10
["MetroStation", "BusTerminal"]	6
["MetroStation", "TrainStation", "BusTerminal"]	4
["TrainStation", "TouristicTerminal"]	4
["MetroStation", "TrainStation", "TouristicTerminal"]	2
["MetroStation", "BusTerminal", "OrcaShuttleTerminal"]	2
["TrainStation", "TouristicTerminal", "BusTerminal"]	2

Slika 4.18. Broj stanica po vrsti

- **Prikaz svih relacija između čvorova**

```
MATCH (n)-[r]->(m)
RETURN n, type(r), m
LIMIT 100;
```

Slika 4.19. Upit za sve relacije između čvorova

Slika 4.19. prikazuje upit za prikaz svih relacija između čvorova, koji su prikazani na slici 4.20.



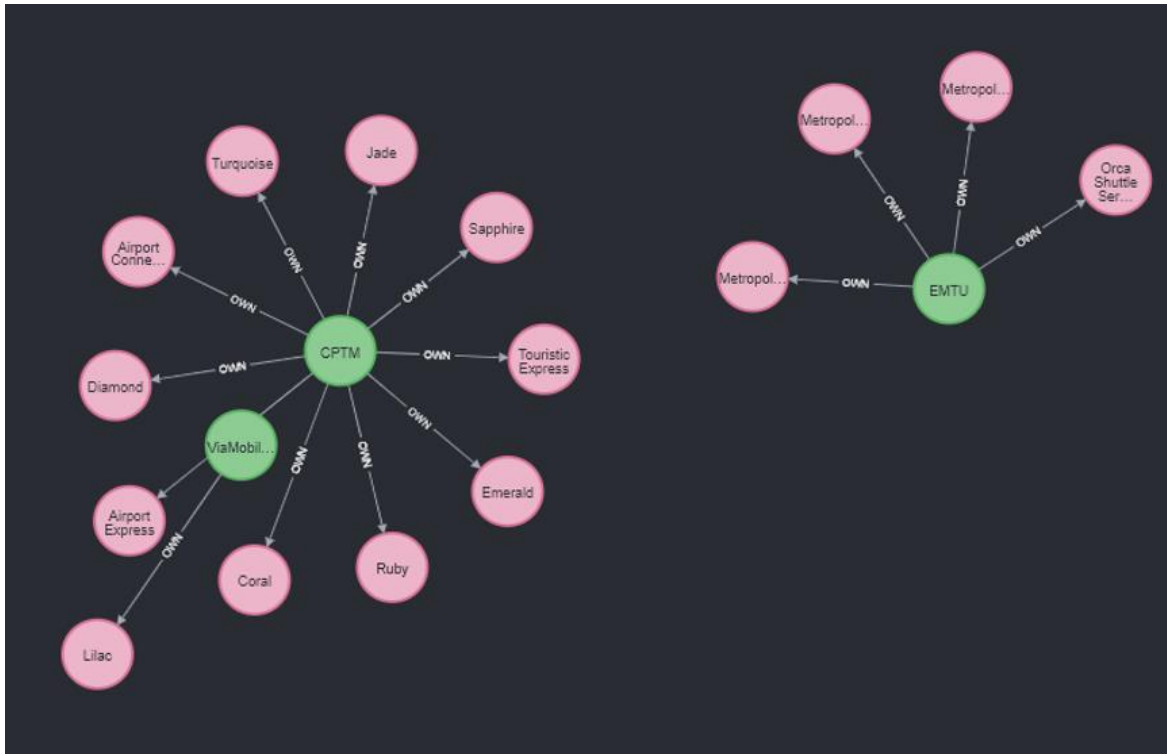
Slika 4.22. Prikaz svih stanica povezanih s turističkim terminalima

- Sve tvrtke koje upravljaju određenim linijama

Upitom sa slike 4.23. prikazane su sve tvrtke koje upravljaju određenim linijama, što je i prikazano na slici 4.24.

```
MATCH (company:Company)-[:OWN]->(line:Line)
RETURN company, line;
```

Slika 4.23. Upit za sve tvrtke koje upravljaju određenim linijama

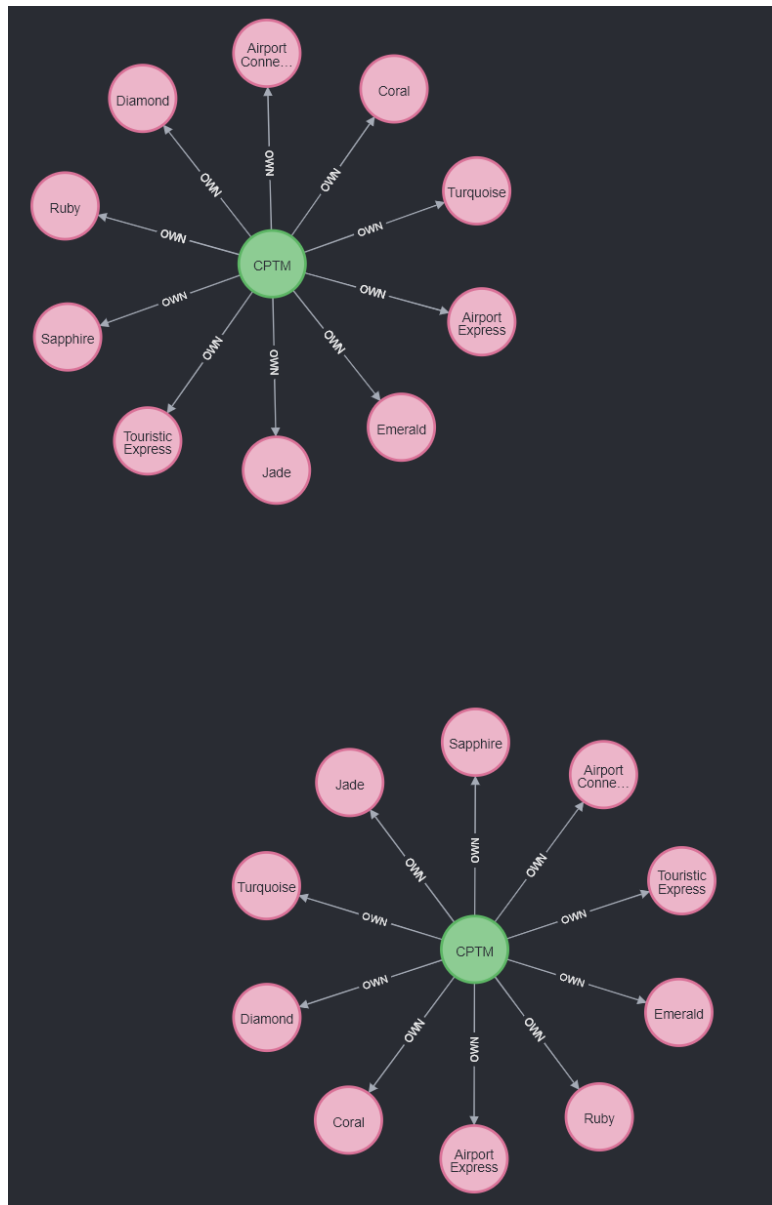


Slika 4.24. Prikaz dijela tvrtki koje upravljaju određenim linijama

- Sve linije kojima upravlja zadana tvrtka

```
MATCH (company:Company)-[:OWN]->(line:Line)
WHERE company.name = 'CPTM'
RETURN company, line;
```

Slika 4.25. Upit za sve linije kojima upravlja zadana tvrtka



Slika 4.26. Prikaz svih linija kojima upravlja zadana tvrtka

Upitom sa slike 4.25. pretražile su se sve linije kojima upravlja zadana tvrtka, te su prikazane na slici 4.26.

5. ZAKLJUČAK

Geoanaliza u graf bazama podataka je već sada iznimno moćan alat s brojnim aplikacijama u različitim industrijama. Ranije opisano i prema primjerima, zaključujemo da su graf baze podataka iznimno funkcionalne i brže od relacijskih baza podataka što ih čini adekvatnijim izborom u današnje vrijeme kada je podataka sve više. Geoanaliza se oslanjala na relacijske baze podataka pohranjujući i obrađujući podatke u strukturiranom obliku, ali s rastućim potrebama za analizom složenih i velikih mreža podataka, javljaju se novi pristupi koji su dosljedniji tim izazovima kao što je upravo graf baze podataka koji nudi efikasna rješenja. Budući trendovi usmjereni su prema dubljoj integraciji s AI tehnologijama, poboljšanju performansi i skalabilnosti, razvoju specijaliziranih algoritama te poboljšanju sigurnosti i privatnosti podataka. S razvojem tehnologije očekuje se da će geoanaliza u graf bazama postati još važnija za donošenje strateških odluka i optimizaciju poslovanja.

LITERATURA

- [1] M. Lapaine, & N. Frančula, *Kartografija i geoinformacije*. Hrvatska akademija znanosti i umjetnosti, 2011.
- [2] B. Mihaljević, *NoSQL i graf baze podataka: Pregled tehnologija i primjena*. Algebra, Zagreb. 2018.
- [3] S. Batra, C. Tyagi, Comparative Analysis of Relational And Graph Databases, *International Journal of Soft Computing and Engineering*, Volume 2, str. 509 – 512, svibanj 2012.
- [4] K. Podolšak, *Primjena geoinformacijskih sustava u urbanizmu i prostornom planiranju*. Sveučilište u Splitu, Građevinsko-arhitektonski fakultet, 2019.
- [5] M. Pavlić, *Primjena graf baza podataka u analizi složenih mreža*. Zbornik radova međunarodnog simpozija INFOTEH-JAHORINA, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva. 2019.
- [6] M. Varga, *Graf baze podataka u poslovnim sustavima*. Zbornik radova sa znanstveno-stručnog skupa "Informacijske tehnologije u poslovanju". Sveučilište u Zagrebu, Fakultet organizacije i informatike. 2015.
- [7] T. Bašić, & B. Bašić, *Prostorna analiza u GIS-u*. Sveučilište u Zagrebu, Geodetski fakultet, 2012.
- [8] Needham M., E. Hodler A., *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*, 2019.
- [9] S. Đurić, *Primjena Neo4j u geoanalizi i prostornom modeliranju*. Geodetski fakultet, Sveučilište u Zagrebu. 2019.
- [10] M. Varga, & V. Vašiček, *Baze podataka* (5. izdanje). TIVA, Varaždin, 2017.
- [11] S. Šestanović, *Geoinformatika: prostorna analiza podataka*. Zagreb: Sveučilište u Zagrebu, 2013.
- [12] <https://neo4j.com/developer-blog/neo4j-desktop-graph-app-restrictions/>
[30.08.2024.]
- [13] <https://www.slideshare.net/slideshow/public-transport-so-paulo-in-a-graph-database-180731153/180731153#1> [01.09.2024.]

SAŽETAK

Razvoj informacijskih tehnologija svakodnevno olakšava poslovanjima diljem svijeta i nudi znatne mogućnosti upravo upravljanjem podacima iz baza podataka. Velike količine podataka su zapravo rudnik ako se nad njima rade različite analize koje se mogu upotrijebiti za daljnje iskorake prema naprijed u vidu poboljšanja ili kako bi se dobio uvid u novije poglede na budućnost i promjene. Tradicionalna primjena relacijskih baza podataka donedavno je bila standard, koje se temelje na relacijskom modelu, ali u zadnjih nekoliko godina prevladavaju nerelacijske baze podataka među koje pripada i graf baza podataka. U radu je opisana primjena geoanalize s graf bazom podataka kao i grane gdje se sve koristi. U današnjici kad se razne infrastrukture, analize, internetske i druge mreže koriste sve više i bez kojih pogled na današnjicu ne bi bio isti, ovakve baze podataka su od velikog značaja.

Ključne riječi: Geoanaliza, graf baze podataka, Neo4j, RDBMS

ABSTRACT

The development of information technologies is continuously facilitating businesses worldwide and offering significant opportunities through data management from databases. Large amounts of data are essentially a gold mine if various analyses are conducted on them, which can be used for further advancements in terms of improvements or to gain new insights into future trends and changes. The traditional use of relational databases, based on the relational model, was a standard until recently, but in recent years, non-relational databases, including graph databases, have prevailed. This paper describes the application of geoanalysis with graph databases, as well as the fields where they are used. In today's world, where various infrastructures, analyses, internet, and other networks are increasingly utilized and indispensable for understanding the present, such databases are of great importance.

Keywords: Geoanalysis, graph databases, Neo4j, RDBMS

PRILOG (on-line)

Prilog 1. Završni rad s korištenim datasetom

https://bitbucket.org/goranmit/zavrsni_rad/downloads/