Nebula Graph

Tim α : Mihael, Zvonimir

Što je Nebula Graph?

- Open-source grafovska baza podataka
- Distribuirana, skalabilna, iznimno brza
- Namijenjena za analizu velikog broja podataka
- Koristi različite algoritme analize grafova
- Ima zaseban jezik za upite i manipulaciju podatcima
- Podržava OpenCypher, upitni jezik za grafovske baze

NebulaGraph

🚀 Connecting with NebulaGraph Community 😊

NebulaGraph

Products Learn Resources Community Contact Us Star 9,139 Get Started

Open Source, Distributed, Scalable, Lightning Fast

The graph database built for super large-scale graphs with milliseconds of latency.

Get Started Live Demo Deploy in Cloud

LATEST

user-review

Benchmark: Neo4j vs NebulaGraph vs JanusGraph

use-case

7 Use Cases in Social Networks with NebulaGraph Database

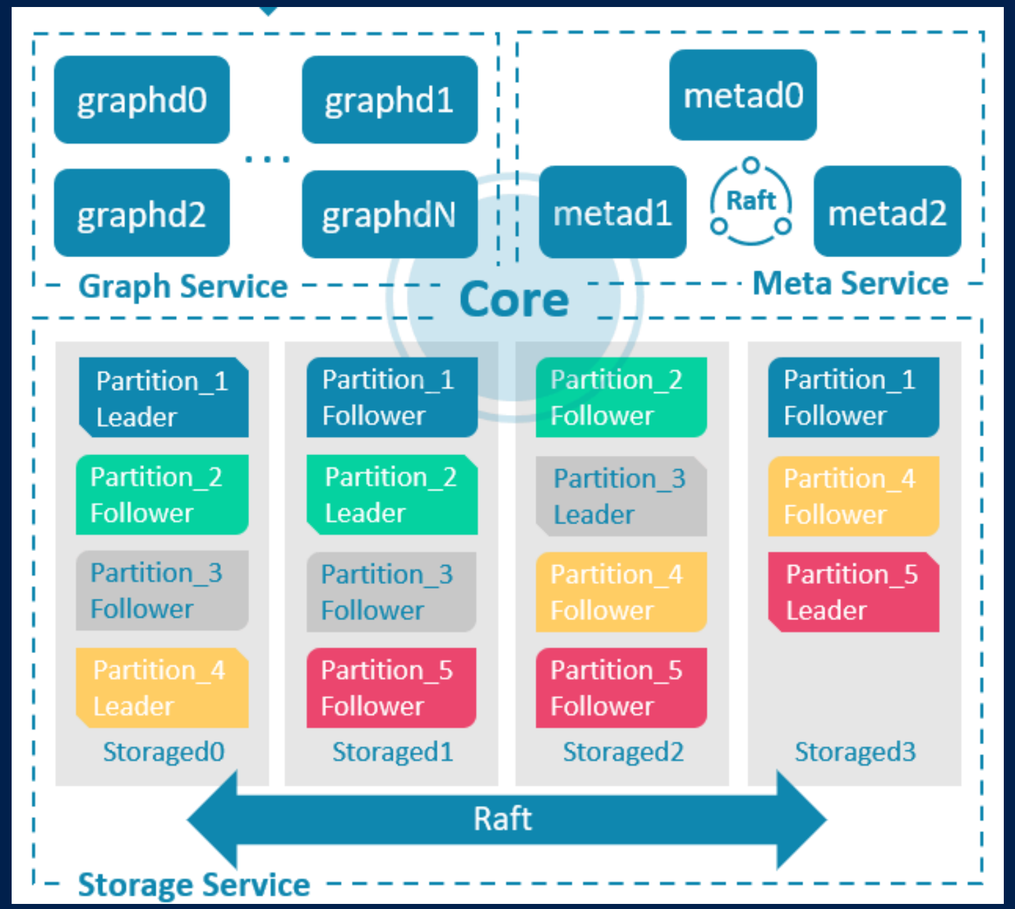
graph-database

Graph Database vs Relational Database: What to Choose?

Arhitektura

Grafovski servis

- domena grafova
- Parser -> Validator -> Planer -> Executor



Meta servis

- računi i privilegije
- sheme
- sl.

- zasebni strojni kodovi -----> DOSTUPNOST

Particije

- *shared nothing*
  - 2 x bridovi
- *static hashing*
- SKALABILNOST
- DOSTUPNOST

Sprem-nički servis

KV Store

- *RocksDB*
- stroga tipizacija
- BRZINA

vrh :

Type (1 byte)	PartID (3 bytes)	VertexID (n bytes)	TagID (4 bytes)	SerializedValue

Primjene

- Detekcija prijevara
  - *mnoge transakcije i uređaji*
  - *uzorci*
- Real-time preporuke
  - *algoritmi preporučivanja*
  - *velika brzina*
- Društvene mreže
  - *velik broj korisnika*
  - *grafovska struktura*

Usporedba performansi

Graph Data Size	Platform	Data Import	One-Hop Query	Two-Hop Query	Shared Friends Query
10 Million Edges	Neo4j	26s	6.618s	6.644s	6.661s
	HugeGraph	89s	16ms	22ms	72ms
	NebulaGraph	32.63s	1.482ms	3.095ms	0.994ms
100 Million Edges	Neo4j	1min21s	42.921s	43.332s	44.072s
	HugeGraph	10min	19ms	20ms	5s
	NebulaGraph	3min52s	1.971ms	4.34ms	4.147ms
1 Billion Edges	Neo4j	8min34s	165.397s	176.272s	168.256s
	HugeGraph	65min	19ms	651ms	3.8s
	NebulaGraph	29min35s	2.035ms	22.48ms	1.761ms
8 Billion Edges	Neo4j	1h23min	314.34s	393.18s	608.27s
	HugeGraph	16h	68ms	24s	541ms
	NebulaGraph	~30min	Less than 1s	Less than 5s	Less than 1s

nGQL

- Nebula Graph Query Language
- Deklarativni jezik za postavljanje upita i manipuliranje Nebula Graph bazom
- Po sintaksi sličan SQL-u
- vertex (node), edge (relationship), tag (label)
- vertex identifier: vid

Nebula Console

nbp_trgovina

Run

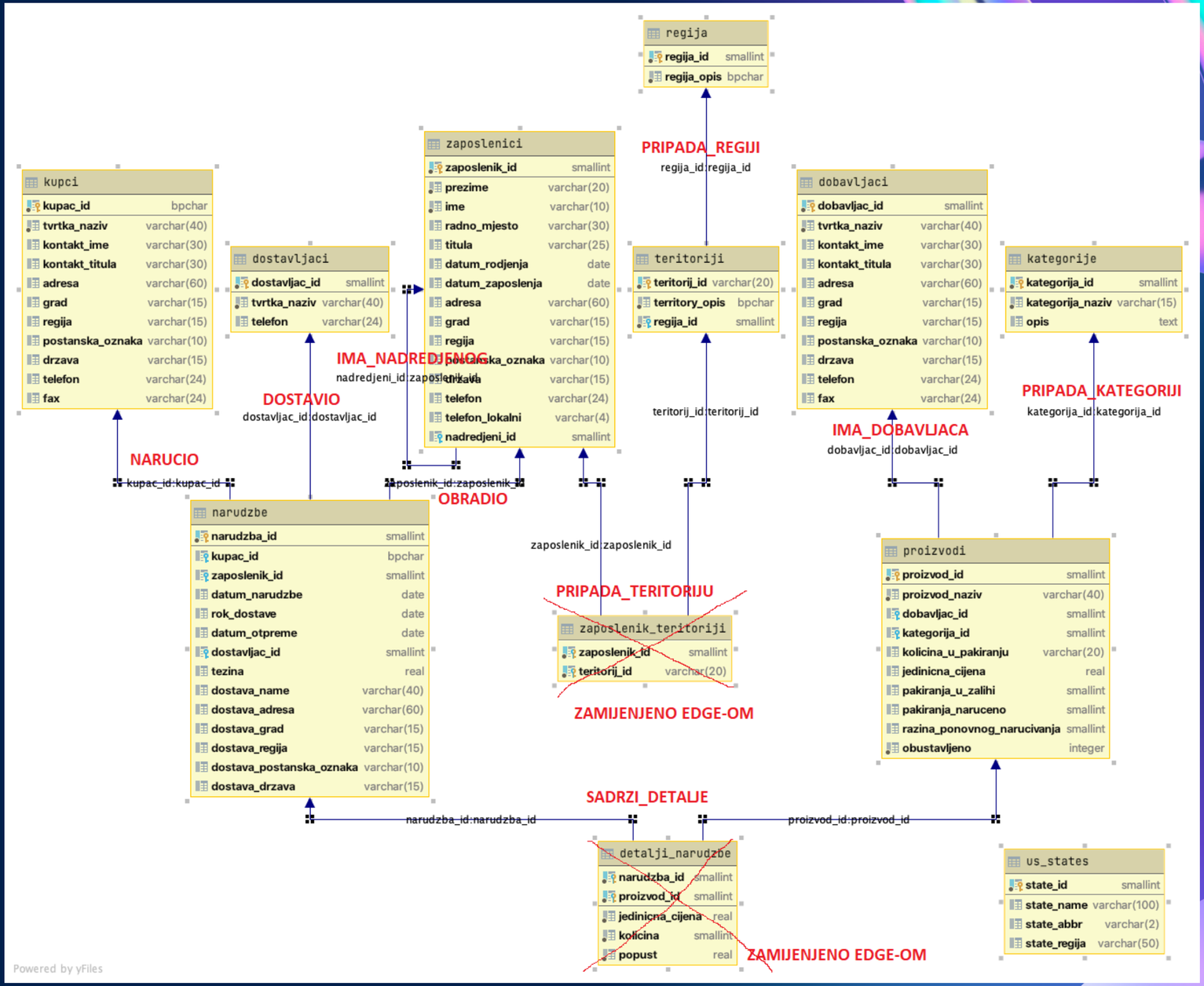
```
INSERT VERTEX proizvodi (proizvod_id, proizvod_naziv, kolicina_u_pakiranju,
                         jedinicna_cijena, pakiranja_u_zalihi, pakiranja_naruceno,
                         razina_ponovnog_narucivanja, obustavljeno)
VALUES "proizvodi_150":("proizvodi_150", "Tikvice", "20 kom.",
                         13.30, 4, 2,
                         0, 0)
```

Migracija baze nbp_trgovina

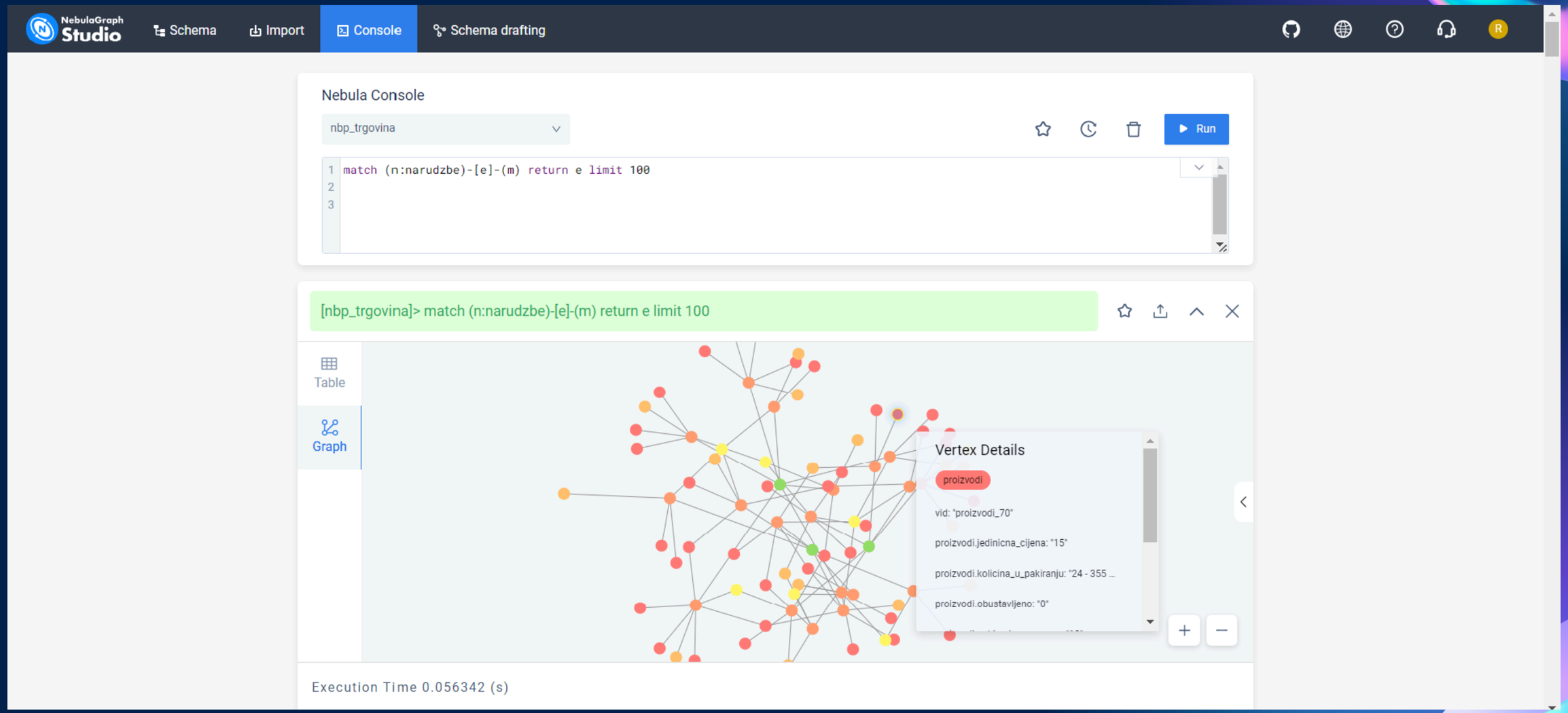
Tri etape migracije

- Načini uvoza u Nebula Graph
  - Nebula Importer
  - Nebula Exchange
  - Nebula Graph Studio
- Načini izvoza iz MongoDB, Neo4j, PostgreSQL (CSV)
- Prilagodba podataka i povezivanje s Nebula Graph strukturama

Izmjene sheme podataka



Nebula Graph Studio



Aplikacija

Instalacija paketa:

```
!pip install nebula3-python
from nebula3 import *
```

Automatizacija spajanja (može i bolje):

```
def povezi_se_i_obavi(upit):
    config = Config()
    config.max_connection_pool_size = 10
    connection_pool = ConnectionPool()

    ok = connection_pool.init([('127.0.0.1', 9669)], config)

    with connection_pool.session_context('root', 'root') as session:
        session.execute('USE nbp_trgovina')

        result = session.execute(upit)

        # convert ResultSet to dict
        columns = result.keys()

        d = {}
        for col_num in range(result.col_size()):
            col_name = columns[col_num]
            col_list = result.column_values(col_name)
            d[col_name] = [str(x)[1:-1] for x in col_list]


    connection_pool.close()
    return d
```

Primjer korištenja:

```
def dohvati_dostavljace(prozor) :
    # ORDER BY koristi pipe, nakon kojeg referenciramo povratnu strukturu s $-.<property_name>
    upit = "LOOKUP ON dostavljaci YIELD properties(vertex).tvrtka_naziv AS naziv | ORDER BY $-.naziv"

    rezultat = povezi_se_i_obavi(upit)

    return rezultat
```

```
dostavljaci = dohvati_dostavljace()["naziv"]
```

Ne funkcionira!? Javlja (nejasnu) grešku *Syntax error near INSERT*.

```
obradio = <prezime_zaposlenika>
ID = <id_narudzbe_koju_unosimo>
```

```
upit = upit + f'''
                LOOKUP ON zaposlenici
                WHERE zaposlenici.prezime == "{obradio}"
                YIELD id(vertex) as id |
                INSERT EDGE obradio () VALUES "{ID}" -> $-.id : ();
            '''
```

Zaobilazak:

```
    obradio_id = povezi_se_i_obavi(f'''

                LOOKUP ON zaposlenici
                WHERE zaposlenici.prezime == "{obradio}"
                YIELD id(vertex) as id;

            ''')["id"][0]

    upit = upit + f'''
                INSERT EDGE obradio () VALUES "{ID}" -> "{obradio_id}" : ();
            '''
```

Prozor *Unos_narudzbe*

```
unos = Unos_narudzba()
unos.draw()
```

id: narudzbe_test

Zaprimljeno: 06/04/2023

Rok: 06/13/2023

Otprema: 06/12/2023

Težina(kg): 100

Obradio: Dodsworth

Naručio: Consolidated Holdings

Dostavljač: DHL

Proizvod	Kolicina	Popust
Aromatični sir	4	0.03
Bakin namaz od jagoda	3	0
Bijela čokolada	2	0.07
Bijeli sir	0	0
Crna alga	0	0
Dalmatinski papar	0	0
Dalmatinski sir	0	0
Dimljeni sir	0	0
Domaći stout	0	0
Dugotrajni tofu	0	0
Francuski kruh	0	0

Unesi

Unesi

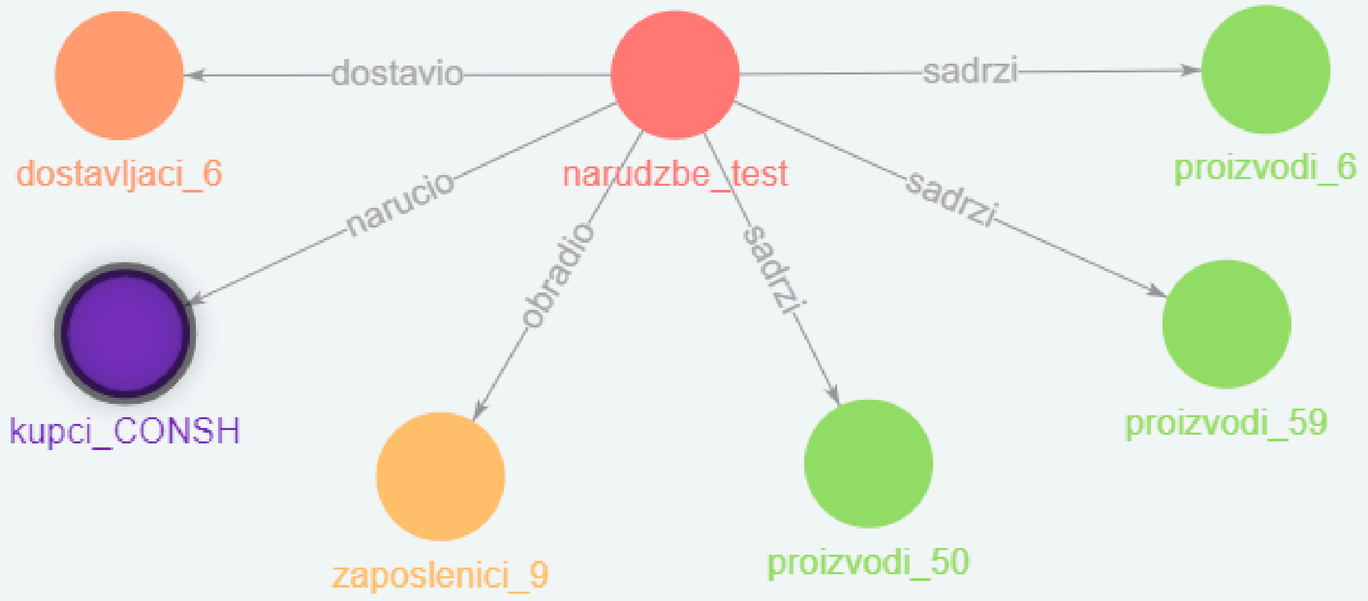
generiran je upit:

```
INSERT VERTEX narudzbe
    (narudzba_id, datum_narudzbe,
    rok_dostave, datum_otpreme,
    tezina)
VALUES
    "narudzbe_test" : ("narudzbe_test","2023-06-04","2023-06-13","2023-06-12",100);


INSERT EDGE obradio () VALUES "narudzbe_test" -> "zaposlenici_9" : ();


INSERT EDGE narucio () VALUES "narudzbe_test" -> "kupci_CONSH" : ();


INSERT EDGE dostavio () VALUES "narudzbe_test" -> "dostavljaci_6" : ();


INSERT EDGE sadrzi (kolicina, popust)
VALUES "narudzbe_test" -> "proizvodi_59" : (4, 0.03);


INSERT EDGE sadrzi (kolicina, popust)
VALUES "narudzbe_test" -> "proizvodi_6" : (3, 0.0);


INSERT EDGE sadrzi (kolicina, popust)
VALUES "narudzbe_test" -> "proizvodi_50" : (2, 0.07);
```

[nbp_trgovina]> match (n:narudzbe {narudzba_id : "narudzbe_test"})-[s]-(p) return n, s, p;

Table

Graph



Execution Time 0.014837 (s)

Update_kupca()

```
update = Update_kupac()
update.draw()
```

```
Unesite ID kupca i popunite one podatke koje je potrebno ažurirati.
Nakon toga pritisnite gumb Ažuriraj na ispod podataka.
```

ID kupca:	kupci_ANATR
Naziv tvrtke	Bjelovar d.o.o.
Kontakt ime:	
Kontakt titula:	
Adresa:	
Grad:	
Regija:	
Poštanska ...	
Država:	
Telefon:	
Fax:	

Ažuriraj

Nebula Console

nbp_trgovina

```
1 MATCH (k:kupci{kupac_id:"kupci_ANATR"}) RETURN k
```

[nbp_trgovina]> MATCH (k:kupci{kupac_id:"kupci_ANATR"}) RETURN k

Table

Graph

kupci_ANAT

kupci.kontakt_titula: "Vlasnik"

kupci.kupac_id: "kupci_ANATR"

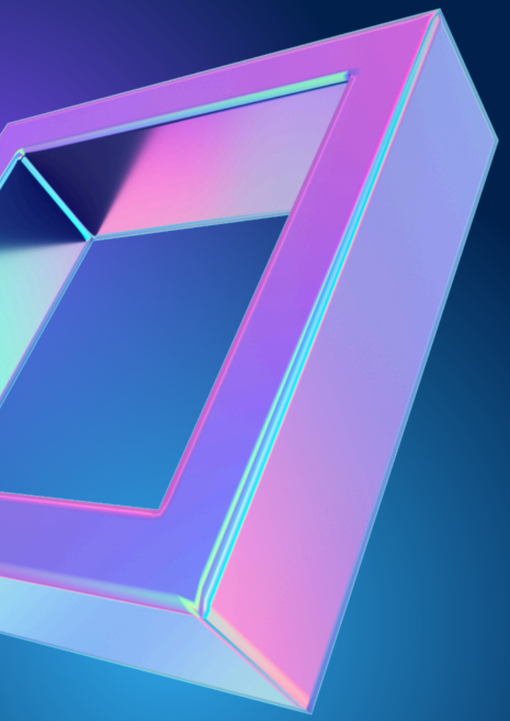
kupci.postanska_oznaka: "5021"

kupci.regija: ""

kupci.telefon: "(5) 555-4729"

kupci.tvrtka_naziv: "Bjelovar d.o.o."

Execution Time 0.004039 (s)

Hvala na slušanju!

Pitanja?