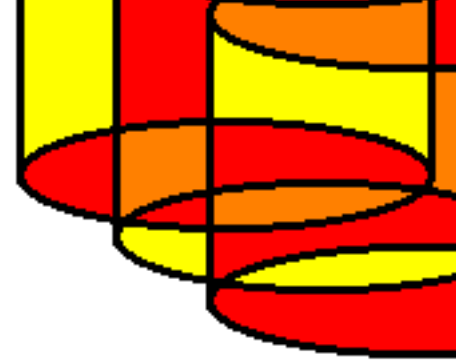
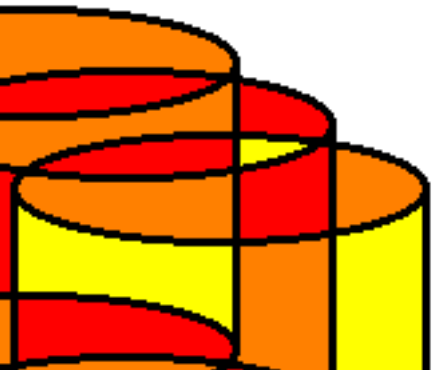


Polustrukturirane graf baze podataka



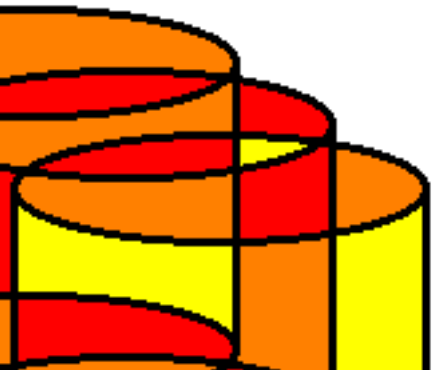
- Osim dva najpoznatija formata vezana uz polustrukturirane podatke i NoSQL (engl. Not only SQL) – JSON i XML – u posljednje se vrijeme razvijaju i novi formati poput graf baza podataka koje se izravno temelje na podatkovnim grafovima
- Najpoznatiji primjer takvog sustava je Neo4j



Neo4j



- Neo4j je sustav za upravljanjem graf bazom podataka u kojoj se podaci pohranjuju u obliku čvorova (engl. nodes) i veza (engl. relations) između njih
- Neo4j se temelji na upitnom jeziku Cypher koji omogućuje raspoznavanje uzoraka u grafovima



Priprema

- Ako na računalu nije instaliran Neo4j moguće ga je instalirati skriptom za instalaciju (<http://tinyurl.com/instalacija2019> – oprez instalirava i sve ostale tehnologije za vježbe) ili putem naredbi:

```
wget --no-check-certificate -O - https://debian.neo4j.org/neotechnology.gpg.key | sudo apt-key add -  
sudo touch /etc/apt/sources.list.d/neo4j.list  
sudo chmod 663 /etc/apt/sources.list.d/neo4j.list  
sudo echo "deb http://debian.neo4j.org/repo stable/" > /etc/apt/sources.list.d/neo4j.list  
sudo apt update  
sudo apt -y install neo4j  
sudo neo4j start
```

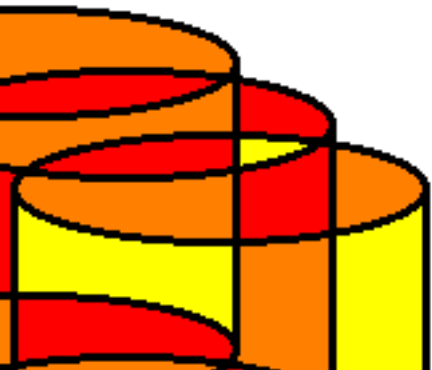
Web sučelje



- Pokretanjem neo4j servisa (zadnja naredba na prethodnom slajdu) pokreće se i web sučelje dostupno na:

`http://localhost:7474/`

- Pretpostavljeno korisničko ime i lozinka su:
neo4j i neo4j



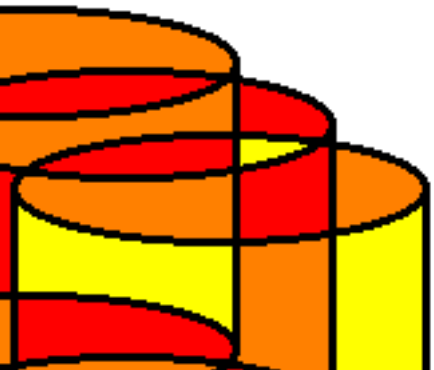
Upute

- Upute postavljamo u tekstualnom polju pri vrhu Web sučelja
- Sljedeće upite postavite u Web sučelju, a upite i dobivene rezultate (dobivaju se odmah ispod tekstualnog polja) kopirajte u datoteku ime_prezime.txt (koristite pogled *Text*)
- Također, provjerite (gdje je to moguće) i pogled *Graph* koji vizualno prikazuje odgovor na upit

Kreiranje čvorova i veza



- Za kreiranje čvorova i veza koristimo klauzulu **CREATE**
- Čvorovi se pišu unutar običnih zagrada, pa tako kreiranje običnog čvora izgleda ovako:
CREATE (cvor)



Kreiranje čvorova

- Po potrebi možemo kreirati više čvorova, npr.:
CREATE (cvor1), (cvor2)

Kreiranje čvorova

- Na čvorove možemo dodavati oznake (engl. Label koje su slične tipovima odnosno klasama), npr.

CREATE (ivek:Osoba)

Kreiranje čvorova

- Čvor može imati više oznaka, npr.:

CREATE (barica:Osoba:Zaposlenik)

Kreiranje čvorova

- Čvorovi mogu imati i atribute (engl. properties),
npr.

```
CREATE ( joza:Osoba { ime:'Josip', prezime:'Presvetli' } )
```

Kreiranje čvorova

- Prilikom kreiranja čvora možemo i vratiti kreirani čvor, npr.:

```
CREATE ( stef:Osoba { ime:"Štef" } )  
RETURN stef.ime
```

Kreiranje veza

- Cypher koristi grafičku sintaksu (gotovo ASCII art) za reprezentaciju veza, npr.

```
CREATE (c1) - [ :VRSTA_VEZE ] -> (c2)
```

Kreiranje veza

- Na način kao što je to učinjeno na prethodnom slajdu, kreirali su se i novi čvorovi. Kako bismo iskoristili postojeće čvorove, moramo koristiti klauzulu MATCH kako bi ih označili (o toj će klauzuli još biti riječi), npr.
MATCH (a:Osoba), (b:Osoba)
WHERE a.ime = 'Josip' AND b.ime = 'Štef'
CREATE (a)-[r:PRIJATELJ_OD]-(b)
RETURN a, type(r), b

Kreiranje veza

- Veze također mogu imati atribute, npr.

```
MATCH (a:Osoba), (b:Osoba)
WHERE a.ime = 'Štef' AND b.ime = 'Josip'
CREATE (a)-[ r:POSLOVNI_PARTNER { firma: a.ime + ' OPG' } ]-(b)
RETURN type(r), r.firma
```

Kreiranje putanje u grafu

- Po potrebi moguće je kreirati čitavu putanju u grafu (kreirajući sve veze odnosno čvorove na njoj).

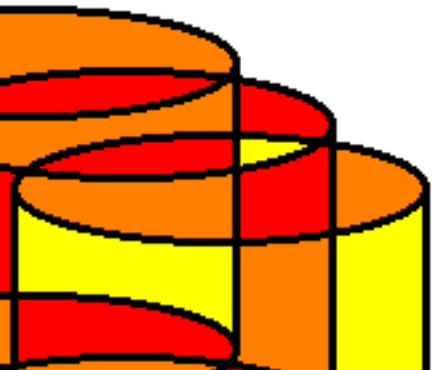
```
CREATE p = ( stefica { ime:'Štefanija' } )-[:RADI_U]->(vindija)<-[:RADI_U]-(dragec {ime:'Dragutin'})  
RETURN p
```

Postavljanje upita



- Za postavljanje upita koristimo klauzulu **MATCH** najčešće uz klauzulu **RETURN**
- Primjerice, da bismo dobili sve čvorove koji su u bazi podataka koristit ćemo naredbu:

```
MATCH (cvor)  
RETURN cvor
```



Primjeri

- Kako bismo dobili samo one čvorove koji imaju odgovarajuću oznaku, koristimo uzorak, npr.

MATCH (cvor:Osoba)

RETURN cvor.ime

Primjeri

- Kako bismo dobili čvorove koji su na bilo koji način povezani koristimo uzorak – (označava bilo kakvu vezu), npr.

MATCH (o1) -- (o2)

RETURN o1, o2

Primjeri

- Naravno, možemo postavljati i uvjete nad atributima, npr.

```
MATCH (osoba:Osoba { prezime:'Presvetli' } )  
RETURN osoba.ime
```

Primjeri

- S uzorcima `-->` i `<--` možemo odrediti smjer veze, npr.

MATCH (o1)-->(o2)

RETURN o1.ime, o2.ime

Primjeri

- Također, možemo postaviti i upit koji nam vraća vrstu veze između neka dva čvora, npr.

```
MATCH (o1) - [veza] -> (o2)
```

```
RETURN o1.ime, type( veza ), o2.ime
```

Primjeri

- Ako poznajemo tip veze, možemo postaviti upit tako da nam vraća samo taj tip veze, npr.

```
MATCH (o1) - [veza:RADI_U] -> (org)  
RETURN o1.ime, type( veza ), org
```

Primjeri

- Možemo istovremeno testirati i bilo koju od više vrsta veza pomoću operatora alternacije |, npr.

```
MATCH (o1) - [:PRIJATELJ_OD | :POSLOVNI_PARTNER] - (o2)  
RETURN o1, o2
```

Primjeri

- Nad atributima čvorova možemo postavljati uvjete pomoću klauzule **WHERE**, npr.

```
MATCH (o1:Osoba)
```

```
WHERE o1.ime = 'Josip' OR o1.ime = 'Štef'
```

```
RETURN o1
```


Primjeri

- Moguće je pretraživati i putanje (više uzastopnih veza) u grafu
- Kreirajmo sljedeću putanju:

```
CREATE (o1)-[:RODBINA {vrsta:'majka'}]->(o2)-[:RODBINA  
{vrsta:'brat'}]->(o3)-[:RODBINA {vrsta:'otac'} ]->(o4)
```

- Zatim postavimo upite:

```
MATCH p=()-[:RODBINA*2]->()  
RETURN relationships( p )
```

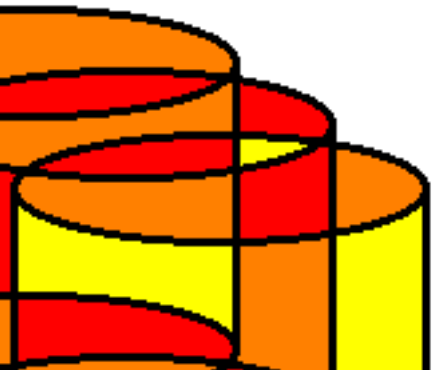
```
MATCH p=()-[:RODBINA*0..3]->()  
RETURN relationships( p )
```

```
MATCH p=()-[:RODBINA*3]->()  
RETURN relationships( p )
```

Pojašnjenje



- Funkcija relationships vraća listu svih veza unutar nekog grafa
- U prvom upitu tražili smo sve putanje za vrstu veze :RODBINA duljine 2
- U drugom upitu tražili smo sve putanje za vrstu veze :RODBINA duljine minimalno 0 maksimalno 2 (uočite da su ušle i sve ostale veze zbog opcionalnosti)
- U trećem upitu tražili smo sve putanje za vrstu veze :RODBINA duljine 3



Sortirnje izlaza

- Kako bismo dobili sortirani izlaz iz upita prema nekom ili više kriterija, koristi se ORDER BY klauzula koja je ekvivalentna istoimenoj klauzuli u SQL-u. Primjer:

```
MATCH (o:Osoba)
```

```
RETURN o
```

```
ORDER BY o.prezime, o.ime DESC
```

- **Napomena:** null vrijednost se uvijek sortira kao zadnja

Unije upita

- Pomoću klauzule UNION [ALL] moguće je raditi unije upita. Opcionalni ALL uključuje duplikate, dok bez njega se duplikati eliminiraju. Primjer:

```
MATCH (n:Osoba)  
RETURN n AS osoba  
UNION  
MATCH (o) - [:RADI_U] ->()  
RETURN o AS osoba
```

Brisanje čvorova

- Čvorove brišemo klauzulom DELETE, pri čemu se čvorovi koje se želi obrisati trebaju prethodno označiti s MATCH, npr.

```
MATCH (o:Osoba:Zaposlenik)  
DELETE o
```

Brisanje čvorova

- Neo4j ne dopušta brisanje ako neki čvor još ima veza. Kako bismo omogućili brisanje i samih veza čvora, koristimo klauzulu DETACH

```
MATCH () - [:RODBINA] -> (o)  
DETACH DELETE o
```

Brisanje veza

- Veze se brišu na ekvivalentan način kao i čvorovi, npr.

```
MATCH () - [r:PRIJATELJ_OD] -> ()  
DELETE r
```

Brisanje atributa

- Atributi se brišu klauzulom REMOVE, npr.

```
MATCH (a:Osoba)  
REMOVE a.prezime  
RETURN a
```


Ažuriranje čvorova

- Kako bismo ažurirali vrijednosti atributa koristimo klauzulu SET također uz označavanje čvora klauzulom MATCH, npr.

```
MATCH (o:Osoba {ime: 'Štef'})  
SET o.ime='Stjepan'  
RETURN o
```

Zadatak



- Obrišite sve čvorove u trenutnoj bazi podataka te kreirajte novu bazu podataka na temu sustava za preporuke filmova:
 - Kreirajte barem 10 čvorova koji su korisnici i imaju barem 6 atributa, među kojima e-mail adresa, grad stanovanja i godina rođenja
 - Importirajte bazu filmova koja je primjer korištenja Neo4j (možete koristiti meta naredbu **:play movies** i zatim na drugom slajdu rezultata koji se pojavi kliknuti na gumb play pored prve CREATE naredbe)
 - Kreirajte barem 20 veza tipa recenzija koje će spajati korisnike s filmovima i sadržavati atribut ocjena (broj od 1 – 5)
 - Zamislite aplikaciju za rezensiranje filmova te postavite barem 5 korisnih upita
 - Neovisno o prethodnom zadatku postavite i sljedeće upite:
 - Upit koji će vraćati čvorove korisnika i čvorove glumaca koji su rođeni iste godine, sortirano prema godini
 - Upit preporuke filmova: treba vraćati čvorove korisnika i čvorove onih filmova u kojima glume glumci koji također glume u filmovima koje je dotični korisnik ocijenio ocjenom 4 ili više, a da te filmove korisnik još nije pogledao (pretpostavka je da ako je korisnik napisao recenziju za neki film da ga je i pogledao)
- Sve korištene naredbe pohranite u datoteku ime_prezime.cypher

