Napredne baze podataka

Vježbe

Domaća zadaća br. 3 – rješenja i komentari

U ovom dokumentu…

- Zadaci koje je trebalo proći u 3. domaćoj zadaći i moguća rješenja
- Komentari na neka od vaših rješenja i usputne opaske

- Koristimo Neo4j
  - Docker container (prema uputama za vježbe)
  - Alternativno:
    - Instalaciju neo4j community na svoje računalo
    - Instalaciju neo4j desktopa na svoje računalo

Priprema baze podataka

- Prema uputama, trebalo je izvršiti naredbe koje su već pripremljene u sklopu neo4j tutoriala, pokretanjem naredbe u neo4j okruženju:

  :play northwind-graph

- Tijekom prolaska kroz demo, to rezultira s izvođenjem naredbi na sljedećim slajdovima

```
LOAD CSV WITH HEADERS FROM "http://data.neo4j.com/northwind/products.csv" AS row CREATE
(n:Product)
SET n = row,
n.unitPrice = toFloat(row.unitPrice),
n.unitsInStock = toInteger(row.unitsInStock),
n.unitsOnOrder = toInteger(row.unitsOnOrder),
n.reorderLevel = toInteger(row.reorderLevel),
n.discontinued = (row.discontinued <> "0");

LOAD CSV WITH HEADERS FROM "http://data.neo4j.com/northwind/categories.csv" AS row CREATE
(n:Category) SET n = row;

LOAD CSV WITH HEADERS FROM "http://data.neo4j.com/northwind/suppliers.csv" AS row CREATE
(n:Supplier) SET n = row;

CREATE INDEX ON :Product(productID);
CREATE INDEX ON :Category(categoryID);
CREATE INDEX ON :Supplier(supplierID);

MATCH (p:Product),(c:Category) WHERE p.categoryID = c.categoryID
CREATE (p)-[:PART_OF]->(c);
MATCH (p:Product),(s:Supplier) WHERE p.supplierID = s.supplierID
CREATE (s)-[:SUPPLIES]->(p);
```

Učitavaju se podaci s neo4j servera.
Stvaraju se tri vrste vrhova – Product, Category, Supplier.

Stvaraju se indeksi za brži dohvat.

Stvaraju se bridovi između Product i Category (Product PART_OF Category).
Stvaraju se bridovi između Supplier i Product (Supplier SUPPLIES Product).
Koriste se strani ključevi između tablica.

```
LOAD CSV WITH HEADERS FROM "http://data.neo4j.com/northwind/customers.csv" AS row
CREATE (n:Customer)
SET n = row;

LOAD CSV WITH HEADERS FROM "http://data.neo4j.com/northwind/orders.csv" AS row
CREATE (n:Order)
SET n = row;

CREATE INDEX ON :Customer(customerID);
CREATE INDEX ON :Order(orderID);

MATCH (c:Customer),(o:Order)
WHERE c.customerID = o.customerID
CREATE (c)-[:PURCHASED]->(o);

LOAD CSV WITH HEADERS FROM "http://data.neo4j.com/northwind/order-details.csv" AS row
MATCH (p:Product), (o:Order)
WHERE p.productID = row.productID
AND o.orderID = row.orderID
CREATE (o)-[details:ORDERS]->(p)
SET details = row, details.quantity = toInteger(row.quantity);
```

Stvaraju se dvije vrste vrhova – Customer i Order.

Stvaraju se indeksi za brži dohvat.

Stvaraju se bridovi između Customer i Product (Customer PURCHASED Product). Koriste se strani ključevi između tablica kako bi se moglo spojiti vrhove.

U ovom učitavanju se odmah formiraju bridovi između već stvorenih vrhova Product i Order (Order ORDERS Product), povezuju se pomoću stranih ključeva iz *order_details* na *orders* i *products* tablice.

Zadatak 1

- Nakon ovakvog načina uvoza podataka, u grafu su neki podaci postali višak. Napisati naredbe kojima će se obrisati ono što je višak u grafu.

Višak su prvenstveno oni atributi koji predstavljaju strane ključeve – zbog same strukture grafa, tu informaciju sada imamo u bridovima.

```
MATCH (order:Order)-[orders:ORDERS]->(product:Product)
REMOVE order.customerID,
       product.supplierID,
       product.categoryID,
       orders.orderID,
       orders.productID;

Set 5294 properties, completed after 60 ms.
```

Moglo je i u 3 naredbe, jedna za Order, druga za Product, a treća za ORDERS

Zadatak 1

- Nakon ovakvog načina uvoza podataka, u grafu su neki podaci postali višak. Napisati naredbe kojima će se obrisati ono što je višak u grafu.

Je li trebalo brisati primarne ključeve? Npr:

```
MATCH (c:Customer) REMOVE c.customerID;
```

U principu ne, iz dva razloga:
- oni možda nose neku informaciju koju ne želimo izgubiti
- nad njima su izgrađeni indeksi

Je li trebalo brisati „prazne” atribute? Npr.

```
MATCH (s:Supplier)
WHERE s.fax = 'NULL'
REMOVE s.fax;
```

Lijepo je ako ste to napravili, ali nije strašno ako niste.

Glavna ideja ovog zadatka je bila shvatiti da su relacije među podacima sada u bridovima i da su nam strani ključevi višak. Ako ste napravili ono s prethodnog slajda, dobili ste 2 boda.

Zadatak 2.

- Napisati upit koji će vratiti popis najprodavanijih proizvoda (najprodavaniji = suma količine prodanih proizvoda u narudžbama). Popis treba sadržavati ime proizvoda, ime kategorije kojoj pripada i broj prodanih proizvoda. Ograničiti ispis na prvih 20 proizvoda.

Trebaju nam kategorije, proizvodi i detalji narudžbi. Dakle tražimo podgrafove (kategorija)-(proizvod)-[detalji]-().

```
MATCH (c:Category)--(p:Product)-[o:ORDERS]-()
RETURN p.productName, c.categoryName, SUM(o.quantity) as brojProdanih
ORDER BY brojProdanih LIMIT 20;
```

Radi jednostavnosti i čitljivosti:
- ne navodimo brid između Category i Product
- ne označavamo vrh Order koji je na kraju uzorka

Nijedno ni drugo ne nosi nikakvu informaciju koja nam treba.

Opaska

- Neki su bespotrebno koristili WITH kad nije imalo smisla, npr.:

```
MATCH (c:Category)--(p:Product)-[o:ORDERS]-()
WITH p.productName, c.categoryName, SUM(o.quantity) as brojProdanih
RETURN p.productName, c.categoryName, brojProdanih
ORDER BY brojProdanih LIMIT 20;
```

- Neki su bespotrebno komplicirali s uzorcima, rastavljajući ih (također vrijedi i za sve ostale zadatke), npr.:

```
MATCH (c:Category)-[]-(p:Product), (p)-[o:ORDERS]-(:Order)
...
```

Nije utjecalo na bodove, ali stavljao sam komentare.

Zadatak 2

```
MATCH (c:Category)--(p:Product)-[o:ORDERS]-()
RETURN p.productName, c.categoryName, SUM(o.quantity) as brojProdanih
ORDER BY brojProdanih LIMIT 20;
```

"p.productName"	"c.categoryName"	"broj"
"Camembert Pierrot"	"Dairy Products"	1577
"Raclette Courdavault"	"Dairy Products"	1496
...		
"Steeleye Stout"	"Beverages"	883
"Chai"	"Beverages"	828

Zadatak 3

- Napisati upit za preporučivanje (recommendation engine, „kupci koji su kupili ovo su kupili i...“). Upit treba dohvatiti 8 proizvoda koji se najčešće kupuju s „Mozzarella di Giovanni“ i poredati ih po tome koliko su često kupljeni zajedno.

Nije bilo potrebe za bilo kakvim kompliciranjem. Tražimo samo proizvode iz iste narudžbe: (mozzarella)—(narudžba)—(drugi proivod) i brojimo ih.

```
MATCH (:Product {productName:'Mozzarella di Giovanni'})--(:Order)--(p2:Product)
RETURN p2.productName, COUNT(*) AS broj
ORDER BY broj DESC LIMIT 8;
```

-- ili –

```
MATCH (mozzarella:Product)--(:Order)--(p2:Product)
WHERE mozzarella.productName = 'Mozzarella di Giovanni'
RETURN p2.productName, COUNT(*) AS broj
ORDER BY broj DESC LIMIT 8;
```

Jednostavni uzorak, bez više uzoraka u upitu, bez prenošenja s WITH i dodatnih varijabli

Zadatak 3

```
MATCH (:Product {productName:'Mozzarella di Giovanni'})--(:Order)--(p2:Product)
RETURN p2.productName, COUNT(*) AS broj
ORDER BY broj DESC LIMIT 8;
```

"p2.productName"	"broj"
"Gorgonzola Telino"	6
"Uncle Bob's Organic Dried Pears"	4
"Gumbär Gummibärchen"	3
"Queso Cabrales"	3
"Tarte au sucre"	3
"Sir Rodney's Marmalade"	3
"Camembert Pierrot"	3
"Geitost"	3

Opaska

- Možemo li umjesto:

```
MATCH (:Product {productName:'Mozzarella di Giovanni'})--(:Order)--(p2:Product)
...
```

koristiti:

```
MATCH (:Product {productName:'Mozzarella di Giovanni’})-[* 2]-(p2:Product)
...
```

?

Ne, jer će onda u rezultate ući i veze između proizvoda preko vrha :Category, a ne samo preko vrha :Order.
Moramo specificirati vrstu vrha.

Zadatak 4.

- Napisati upit kojim će se pronaći koji su kupci (Customer.contactName) prema svome izboru proizvoda najsličniji kupcu kojeg predstavlja Paula Wilson. Ispisati prvih 5 osoba.

Najzanimljiviji zadatak, jer ste mogli biti kreativni i sami odrediti što znači „najsličniji”.

To znači da je većina za njega dobila 2 boda, osim ako niste zeznuli između ideje što želite dobiti i realizacije toga. Dakle, nije se ocjenjivala ideja, nego realizacija.

U nastavku slijede neke ideje (i vaše i moje) te moja naivna ideja kako bih procijenio tko je najsličniji Pauli.

Zadatak 4

- Dva najčešća rješenja:
  - Brojimo sve proizvode koje su kupili i Paula i drugi
  - Brojimo distinct proizvode koje su kupili i Paula i drugi
- cca 80% rješenja spada u kategoriju, ne brojeći one koji su imali jednu od ove dvije ideje ali ju nisu uspješno realizirali
- Treba naglasiti da ste imali velike varijacije u realizaciji ovih ideja, s višestrukim upitima, uzorcima u kojima ste nabrajali nepotrebne elemente (bridove, postavljali varijable koje ne koristite), kreirali kolekcije pa onda mjerili njihovu veličinu (umjesto koristiti count), ….

Zadatak 4

- Najjednostavniji način - brojimo proizvode preko kojih su povezani Paula i drugi. Ovo ne uzima u obzir činjenicu da su drugi možda naručili isti proizvod puno više puta nego Paula.

```
MATCH (c1:Customer)--(:Order)--(p:Product)--(:Order)--(c2:Customer)
WHERE c1.contactName = 'Paula Wilson'
AND c1 <> c2
RETURN c2.contactName, COUNT(p.productID) AS broj
ORDER BY broj DESC LIMIT 5;
```

"c2.contactName"	"broj"
"Jose Pavarotti"	135
"Roland Mendel"	116
"Horst Kloss"	85
"Patricia McKenna"	63
"Maria Larsson"	58

Ovaj uvjet c1<>c2 ste napravili na najrazličitije moguće načine… uglavnom usporedbom imena, ali samo jednom usporedbom samih vrhova.

Za ovakvo rješenje nije trebalo dodatne uzorke, nije trebalo prenositi rezultate u druge upite s WITH niti brojati nešto drugo.

Zadatak 4

- Ako hoćemo uspoređivati samo po broju različitih proizvoda, onda jedino u prethodni upit treba dodati distinct, i dobiju se drugačiji rezultati

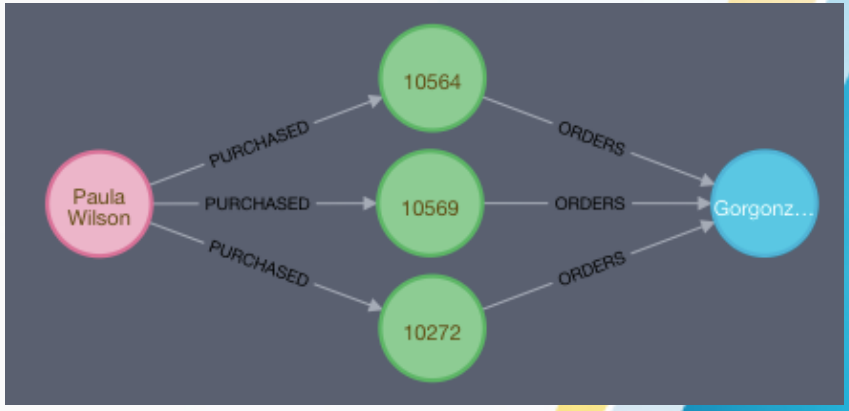
```
MATCH (c1:Customer)--(:Order)--(p:Product)--(:Order)--(c2:Customer)
WHERE c1.contactName = 'Paula Wilson'
AND c1 <> c2
RETURN c2.contactName, COUNT(DISTINCT p.productID) AS broj
ORDER BY broj DESC LIMIT 5;
```

"c2.contactName"	"broj"
"Roland Mendel”	33
"Jose Pavarotti”	29
"Horst Kloss"	29
"Christina Berglund”	26
"Patricia McKenna"	25

Funkcija size(*pattern*)

- Neki od vas su zadatak riješili uz pomoć funkcije size(*pattern*)
- Ova funkcija, kad joj damo uzorak kao argument, vraća broj podgrafova grafa koji odgovaraju tom uzorku
- Npr., Paula je tri puta naručila gorgonzolu, što možemo vidjeti upitom:

```
MATCH (c:Customer {contactName: 'Paula Wilson’})
--(o:Order)
--(p:Product {productName: 'Gorgonzola Telino’})
 RETURN c,o,p;
```




```
MATCH (c:Customer {contactName: 'Paula Wilson'})--(o:Order)--
(p:Product {productName: 'Gorgonzola Telino'}) RETURN COUNT(*);
```

vraća broj 3 kao rezultat

Funkcija size(*pattern*)

- Ako izvedemo samo:

```
RETURN size (
  (:Customer {contactName: 'Paula Wilson’})
  --(:Order)
  --(:Product {productName: 'Gorgonzola Telino’})
);
```

- također dobivamo broj 3 kao rezultat
- Obratite pažnju da ovdje ne koristimo varijable jer nas samo zanima koliko ovakvih uzoraka ima.
- Iako su Paula i Gorgonzola po jednom u bazi, ovakvih podgrafova zapravo ima tri, jer su povezani preko tri vrha tipa :Order

A kako riješiti zadatak s ovim??

Zadatak 4

- Koliko je puta tko naručio proizvod možemo vidjeti i na ovaj način:

```
MATCH (c:Customer)-[]->()-[]->(p:Product)
RETURN c.contactName,
       p.productName,
       size((p)<-[]-()<-[]-(c)) AS brojNarudzbiProizvoda;
```

- S time onda možemo kombinirati koliko je puta netko drugi naručio proizvod i koliko ga je puta Paula naručila i vratiti usporedbu

Zadatak 4

- Slijedom prethodnih slajdova, neka od vaših rješenja bila su ovog tipa:

```
MATCH (c:Customer)-[]->()-[]->(p:Product)
WITH c, p,
     size((c)-[]->()-[]->(p)) AS brNarudzbiDrugi,
     size((:Customer {contactName:'Paula Wilson'})-[]->()-[]->(p)) AS
                                                        brNarudzbiPaula
WHERE c.contactName <> 'Paula Wilson'
  AND brNarudzbiPaula = brNarudzbiDrugi
RETURN c.contactName, COUNT(brNarudzbiDrugi) AS brNarudzbiProizvoda
ORDER BY brNarudzbiProizvoda DESC LIMIT 5;
```

"c.contactName"	"brNarudzbiProizvoda"
"Christina Berglund"	20
"Jose Pavarotti"	19
"Roland Mendel"	18
"Carlos Hernández"	16
"Horst Kloss"	15

Zašto je ovo ***krivo***?

Jer se u MATCH nalaze svi ovakvi uzorci, i tamo gdje ih više od jedan (npr. Paula 4 puta naručivala neki proizvod), će se rezultati toliko puta i vratiti, pa ćemo imati npr. 4 puta isti c, p, brNaruzbi (=4) i onda neki rezultati isplivaju previše (npr Jose Pavarotti).

Zadatak 4

- Idemo analizirati što upit zbilja vrati, tako da malo promijenimo RETURN i ograničimo se samo na jednog kupca (npr. Jose)
- Boldano su promjene u odnosu na upit s prethodnog slajda

```
MATCH (c:Customer)-[]->()-[]->(p:Product)
WITH c, p,
     size((c)-[]->()-[]->(p)) AS brNarudzbiDrugi,
     size((:Customer {contactName:'Paula Wilson'})-[]->()-[]->(p)) AS
                                                        brNarudzbiPaula
WHERE c.contactName = 'Jose Pavarotti'
  AND brNarudzbiPaula = brNarudzbiDrugi
RETURN c.contactName, p.productName, brNarudzbiDrugi, brNarudzbiPaula
ORDER BY brNarudzbiDrugi DESC, c.contactName, p.productName
```

Rezultati upita na sljedećem slajdu

Zadatak 4

"c.contactName"	"p.productName"	"brNarudzbiDrugi"	"brNarudzbiPaula"
"Jose Pavarotti"	"Gnocchi di nonna Alice"	4	4
"Jose Pavarotti"	"Gnocchi di nonna Alice"	4	4
"Jose Pavarotti"	"Gnocchi di nonna Alice"	4	4
"Jose Pavarotti"	"Gnocchi di nonna Alice"	4	4
"Jose Pavarotti"	"Gorgonzola Telino"	3	3
"Jose Pavarotti"	"Gorgonzola Telino"	3	3
"Jose Pavarotti"	"Gorgonzola Telino"	3	3
"Jose Pavarotti"	"Rhönbräu Klosterbier"	3	3
"Jose Pavarotti"	"Rhönbräu Klosterbier"	3	3
"Jose Pavarotti"	"Rhönbräu Klosterbier"	3	3
"Jose Pavarotti"	"Wimmers gute Semmelknödel"	2	2
"Jose Pavarotti"	"Wimmers gute Semmelknödel"	2	2
"Jose Pavarotti"	"Chef Anton's Gumbo Mix"	1	1
"Jose Pavarotti"	"Escargots de Bourgogne"	1	
"Jose Pavarotti"	"Lakkalikööri"	1	
"Jose Pavarotti"	"NuNuCa Nuß-Nougat-Creme"	1	
"Jose Pavarotti"	"Queso Manchego La Pastora"	1	
"Jose Pavarotti"	"Röd Kaviar"	1	1
"Jose Pavarotti"	"Tunnbröd"	1	1

Vidimo multipliciranje rezultata iako je Jose zapravo kupio 4 puta njoke kao i Paula ☺, dakle koncept upita je krivi… (ali nije posebno sankcionirano u bodovima, radi dobre ideje)

Zadatak 4

- Kako najjednostavnije popraviti prethodni upit?
- Brojati distinct proizvode (dobivaju se dosta različiti rezultati)
- Razlike u odnosu na upit su boldane

```
MATCH (c:Customer)-[]->()-[]->(p:Product)
WITH c, p,
     size((c)-[]->()-[]->(p)) AS brNarudzbiDrugi,
     size((:Customer {contactName:'Paula Wilson'})-[]->()-[]->(p)) AS
                                                          brNarudzbiPaula
WHERE c.contactName <> 'Paula Wilson'
  AND brNarudzbiPaula = brNarudzbiDrugi
RETURN c.contactName, COUNT(DISTINCT p) AS brNarudzbiProizvoda
ORDER BY brNarudzbiProizvoda DESC LIMIT 5;
```

"c.contactName"	"brNarudzbiProizvoda"
"Christina Berglund"	15
"Roland Mendel"	12
"Laurence Lebihan"	12
"Patricia McKenna"	12
"Carlos Hernández"	12

Zadatak 4

- Moja ideja je ovog upita je bila slična prethodnoj, ali bez korištenja size(*pattern*) jer to nismo ni prošli na predavanju
- Ideja upita: vidjeti koliko puta je Paula naručila neki proizvod, koliko puta su taj isti proizvod naručili drugi, usporediti te vidjeti u koliko takvih narudžbi se preklapaju
- Koristimo samo count i s with ulančavamo upite

Zadatak 4

```
MATCH
(paula:Customer {contactName: 'Paula Wilson'})--(:Order)--(proizvod:Product)
WITH paula, proizvod, COUNT(*) AS brojPaula

MATCH (drugi:Customer)--(:Order)--(proizvod)
WHERE drugi<>paula
WITH drugi, proizvod, COUNT(*) AS brojDrugi, brojPaula

WHERE brojDrugi = brojPaula
RETURN drugi.contactName, COUNT(*) AS brojPreklapanja
ORDER BY brojPreklapanja DESC LIMIT 5;
```

"drugi.contactName"	"brojPreklapanja"
"Christina Berglund"	15
"Roland Mendel"	12
"Michael Holz"	12
"Laurence Lebihan"	12
"Philip Cramer"	12

Vrh *paula* prenosimo u drugi upit samo da bismo razlikovali druge kupce, nakon toga nam više ne treba

Filtriramo samo one parove kupac-proizvod gdje je isti broj narudžbi kao i kod Paule.

Rezultati su zapravo isti kao u prethodnom upitu, ali je poredak ovih s 12 preklapanja različit jer je neo4j došao do rezultata na različite načine

Zadatak 4 - zaključno

- Bilo je i rješenja gdje se pokušalo bodovati sličnosti, pohvala za kreativnost

- Sličnost je u ovakvim slučajevima teško definirati, npr. je li sličniji netko tko je kupio točno ono što i Paula, a uz to i puno više toga, ili netko tko je kupio skoro sve kao i Paula i ništa drugo…

- Zato se ovakve mjere određuju prema primjeni, i to je sasvim u redu. Ali treba znati kako se mjera računa u kodu, i to imati dokumentirano!

Zadatak 5

- Napisati upit koji će svakoj narudžbi dodati svojstvo total koje predstavlja ukupnu plaćenu cijenu te narudžbe (uzeti u obzir količine i jedinične cijene kupljenih proizvoda). Napomena: unit_price je importiran kao string, pa za njega treba koristiti funkciju toFloat().

```
MATCH (order:Order)-[details:ORDERS]-()
WITH order,
     SUM(toFloat(details.unitPrice) * details.quantity) AS ukupno
SET order.total = ukupno;

Set 830 properties, completed after 391 ms.
```

Prilično straight-forward.

Ali bilo je tu i znatno kompliciranijih rješenja…

Stay tuned…

Još samo jedna zadaća!