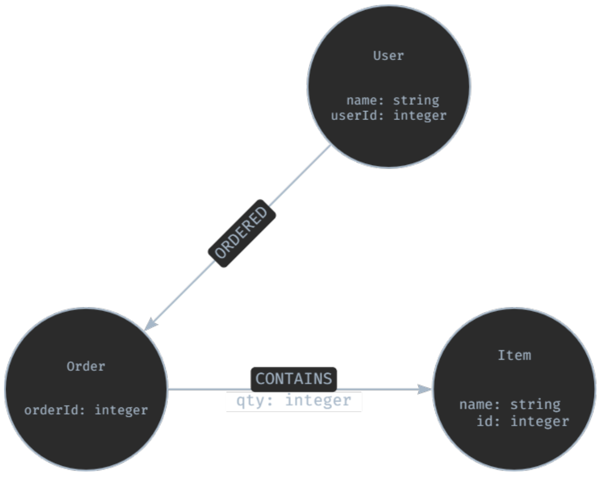
CS343 Graph Data Science
Spring 2024

Quiz 2
Model Solution

ID & Name :

Imagine you are managing a food delivery service database that tracks users, orders, and food items. Each user can place orders consisting of various food items, each with its own name and price. Users can order multiple quantities of items in a single order. We are storing this data in a graph database, where the Graph Property Model for this database is given below:



1. Write Cypher queries to answer the following questions. You can assume that the graph database is already populated with the data.

   (a) (5 points) Create a user named "Steve".

   Solution:
   ```
   Create (m:User{name:"Steve"})
       OR
   CREATE (m:User) SET m.name = "Steve"
       OR
   MREGE (m:User{name:"Steve"})
       OR
   MREGE (m:User) SET m.name = "Steve"
   ```

   (b) (5 points) Create an item named "Dosa" if it does not already exist.

   Solution: MERGE (m:Itemname:"Dosa")

(c) (5 points) Retrieve all users who have ordered the item ”Pizza”.

Solution:

```
MATCH (u:User)-[:ORDERED]->(o:Order)-[:CONTAINS]->(i:Item{name:"Pizza"})
RETURN u
```

(d) (5 points) Retrieve all items ordered by the user named ”Asim”.

Solution:

```
MATCH (u:User{name:"Asim"})-[:ORDERED]->(o:Order)-[:CONTAINS]->(i:Item)
RETURN i
```

(e) (10 points) Retrieve all users who ordered pizza in more than one quantity in an order.

Solution:

```
MATCH (u:User)-[:ORDERED]->(o:Order)-[c:CONTAINS]->(i:Item{name:"Pizza"})
WHERE c.qty > 1
return u
```

(f) (10 points) Retrieve all users who ordered both Pizza and Pasta together.

Solution:

```
match (i:Item{name:"Pizza"})
match (j:Item{name:"Pasta"})
match (u:User)-[:ORDERED]->(o:Order)-[:CONTAINS]-(i)
match (u)-[:ORDERED]->(o)-[:CONTAINS]-(j)
return *
```

(g) (10 points) Write a query to increase the price of all items by 10%.

Solution: MATCH (i:Item) SET i.price = i.price * 1.1

(h) (10 points) Create a new order by the user ”Steve” with two items, ”Pizza” and ”Pasta”, each with a quantity of one.

Solution:

```
MATCH (u:User{name:"Steve"})
CREATE (u)-[:ORDERED]->(o:Order)
CREATE (o)-[:CONTAINS{qty:1}]->(i:Item{name:"Pizza"})
CREATE (o)-[:CONTAINS{qty:1}]->(j:Item{name:"Pasta"})
```

(i) (10 points) Delete the user ”Asim” along with all of their orders.

Solution:

```
MATCH (u:User{name:"Asim"})
```

```
DETACH DELETE u
```

(j) (25 points) We want to recommend items for a user who is buying Pizza. To find items to recommend, we need to find items that are bought with pizza by any user. Write a query to find such items.

Solution:

```
MATCH (i:Item{name:"Pizza"})<-[:CONTAINS]-(o:Order)-[:CONTAINS]->(j:Item)
WHERE j <> i
RETURN j.name, count(j) as count
ORDER by count desc
```