

Neo4j Sample Question

1. Model the following Clothing Brand information as a graph. Consider a Mall for clothing. This mall will include different sections for males, females and kids. Each section contains different types of apparels from different brands. There are many apparels with different designs, of each type. An apparel may be available in one or more standard sizes (S/M/L/XL/XXL)
 - a) Create the nodes as Mall, Apparel, Size, Section and relationships as Has_Section, Contains, Available_in and so on.
Create the relationship by following the given structure.
(:Mall)-[:HAS_SECTION]->(:Section) (A mall has multiple sections)
(:Section)-[:CONTAINS]->(:ApparelType) (A section contains various types of apparels)
(:Apparel)-[:AVAILABLE_IN]->(:Size) (An apparel is available in multiple sizes of apparels)
 - b) Write a Cypher Query Language to list the different apparels type in the female section.

Solution:

- a) 1. Create a mall node:

```
CREATE (mall:Mall {name: 'Main Mall'})
```

2. create section nodes:

```
CREATE (femaleSection:Section {name: 'Female'})  
CREATE (maleSection:Section {name: 'Male'})  
CREATE (kidsSection:Section {name: 'Kids'})
```

3. Create apparel-type nodes

```
CREATE (tshirt:ApparelType {name: 'T-shirt'})  
CREATE (jeans:ApparelType {name: 'Jeans'})  
CREATE (dress:ApparelType {name: 'Dress'})  
CREATE (shirt:ApparelType {name: 'Shirt'})  
CREATE (jacket:ApparelType {name: 'Jacket'})
```

4. Create Apparel nodes:

```
CREATE (tshirtApparel:Apparel {design: 'Round Neck', brand: 'Brand A'})  
CREATE (jeansApparel:Apparel {design: 'Slim Fit', brand: 'Brand B'})  
CREATE (dressApparel:Apparel {design: 'Floral', brand: 'Brand C'})  
CREATE (shirtApparel:Apparel {design: 'Checked', brand: 'Brand D'})  
CREATE (jacketApparel:Apparel {design: 'Bomber', brand: 'Brand E'})
```

5. Create size nodes:

```
CREATE (sizeS:Size {size: 'S'})  
CREATE (sizeM:Size {size: 'M'})  
CREATE (sizeL:Size {size: 'L'})  
CREATE (sizeXL:Size {size: 'XL'})  
CREATE (sizeXXL:Size {size: 'XXL'})
```

Create the relationships:

1. Create Relationships Between Mall and Sections

```
MATCH (mall:Mall {name: 'Main Mall'}), (femaleSection:Section {name: 'Female'}),  
(maleSection:Section {name: 'Male'}), (kidsSection:Section {name: 'Kids'})  
CREATE (mall)-[:HAS_SECTION]->(femaleSection)  
CREATE (mall)-[:HAS_SECTION]->(maleSection)  
CREATE (mall)-[:HAS_SECTION]->(kidsSection)
```

2. Create Relationships Between Sections and Apparel Types

```
MATCH (femaleSection:Section {name: 'Female'}), (tshirt:ApparelType {name:  
'T-shirt'}), (jeans:ApparelType {name: 'Jeans'}), (dress:ApparelType {name:  
'Dress'})  
CREATE (femaleSection)-[:CONTAINS]->(tshirt)  
CREATE (femaleSection)-[:CONTAINS]->(jeans)  
CREATE (femaleSection)-[:CONTAINS]->(dress)
```

3. Create Relationships Between Apparel and Sizes

```
MATCH (tshirtApparel:Apparel {design: 'Round Neck'}), (sizeS:Size {size: 'S'}),  
(sizeM:Size {size: 'M'}), (sizeL:Size {size: 'L'})  
CREATE (tshirtApparel)-[:AVAILABLE_IN]->(sizeS)  
CREATE (tshirtApparel)-[:AVAILABLE_IN]->(sizeL)  
  
MATCH (jeansApparel:Apparel {design: 'Slim Fit'}), (sizeM:Size {size: 'M'}),  
(sizeL:Size {size: 'L'}), (sizeXL:Size {size: 'XL'})  
CREATE (jeansApparel)-[:AVAILABLE_IN]->(sizeM)  
CREATE (jeansApparel)-[:AVAILABLE_IN]->(sizeXL)
```

b) Write a Cypher Query Language to list the different apparels type in the female section.

```
MATCH  
(mall:Mall)-[:HAS_SECTION]->(section:Section)-[:CONTAINS]->(apparelType:ApparelType)  
  
WHERE section.name = 'Female'  
  
RETURN DISTINCT apparelType.name AS ApparelType
```

Mongodb Sample Question

Model the Dairy Brand Information system by considering the given below collections.

I. Create the below collections.

- a) Dairy_brand - Insert five documents in the Dairy_brand collection with the attributes like brand_id, name, locations, product.
- b) Product - Insert five documents in the Product collection with the attributes like product_name, category, popularity.

II. Write a query to display the popular cheese brands in Gujarat using a cursor.

Solution

I Creating Dairy_brandcollection

```
db.Dairy_brand.insertMany([
  {
    brand_id: 1,
    name: "Amul",
    locations: ["Gujarat", "Maharashtra", "Delhi"],
    product: ["Cheese", "Butter", "Milk"]
  },
  {
    brand_id: 2,
    name: "Mother Dairy",
    locations: ["Gujarat", "Rajasthan", "Delhi"],
    product: ["Cheese", "Curd", "Milk"]
  },
  {
    brand_id: 3,
    name: "Britannia",
    locations: ["Gujarat", "Maharashtra", "Tamil Nadu"],
    product: ["Cheese", "Biscuits", "Milk"]
  },
  {
    brand_id: 4,
    name: "Nestle",
    locations: ["Gujarat", "Karnataka", "Maharashtra"],
    product: ["Cheese", "Milk", "Ice Cream"]
  },
  {
    brand_id: 5,
    name: "Govind Milk",
    locations: ["Gujarat", "Madhya Pradesh", "Rajasthan"],
    product: ["Cheese", "Butter", "Curd"]
  }
]);
```

Creating product collection

```
db.Product.insertMany([
  {
    product_name: "Cheese",
    category: "Cheese",
    popularity: 5
  },
  {
    product_name: "Butter",
    category: "Butter",
    popularity: 3
  },
  {
    product_name: "Milk",
    category: "Milk",
    popularity: 4
  },
  {
    product_name: "Ice Cream",
    category: "Ice Cream",
    popularity: 2
  },
  {
    product_name: "Curd",
    category: "Curd",
    popularity: 3
  }
]);
```

II Write a query to display the popular cheese brands in Gujarat using a cursor.

```
db.Dairy_brand.aggregate([
  {
    $match: {
      locations: "Gujarat"
    }
  },
  {
    $lookup: {
      from: "Product",
      localField: "product",
      foreignField: "product_name",
      as: "product_details"
    }
  },
  {
    $unwind: "$product_details"
```

```
    },
    {
      $match: {
        "product_details.category": "Cheese"
      }
    },
    {
      $sort: { "product_details.popularity": -1 }
    },
    {
      $project: {
        _id: 0,
        brand_name: "$name",
        cheese_product: "$product_details.product_name",
        popularity: "$product_details.popularity"
      }
    }
  ]).forEach(function(doc) {
    printjson(doc);
  });
```

