

Tutorial 9b - Solution

Neo4j Basic

In this tutorial, you are going to implement a graph database for a simple online electronic shop. The entities that need to be managed consist of products and customer details. Table 1 shows the list of products and its related category, while Table 2 lists the current customers. You are required to store this information as nodes in the graph database.

Other than recording the entities, we also need to maintain the activities of the customers. The activities are mainly on keeping track of how many times a product has been viewed by a user, the wish list of each customer, and the list of products that have been purchased by a certain customer. Table 3 shows the list of products that each customer has viewed and the number of times each product has been viewed by the corresponding customer. Table 4 shows the list of products that have been added to the customer's wish list. Table 5 shows the products that the customers have purchased. The data listed in Table 3-5 should be stored as relationships in the graph database that you are going to build.

Table 1. Products

Name	Availability	Shippability	Price	Category
Sony Experia Z22	true	true	765.00	Smartphones
Samsung Galaxy S8	true	true	784.00	Smartphones
Sony Xperia XA1 Dual G3112	false	true	229.50	Smartphones
Apple iPhone 8 Plus 64GB	false	true	874.20	Smartphones
Xiaomi Mi Mix 2	true	true	420.87	Smartphones
Huawei P8 Lite	true	true	191.00	Smartphones
Acer Swift 3 SF314-51-34TX	false	true	595.00	Notebooks
HP ProBook 440 G4	true	true	771.30	Notebooks
Dell Inspiron 15 7577	true	true	1477.50	Notebooks
Apple MacBook A1534 12' Rose Gold	true	false	1293.00	Notebooks
Canon EOS 6D Mark II Body	false	true	1794.00	Cameras
Nikon D7500 Kit 18-105mm VR	true	true	1612.35	Cameras

Table 2. Customers

Name	Email	Age
Joe Baxton	joe_baxton@example.com	25
Daniel Johnston	dan_j@example.com	31
Alex McGyver	mcgalex@example.com	22
Allison York	ally_york1@example.com	24

Table 3. Products viewed by customer

Customer	Product Viewed	View counts
Joe	Sony Experia Z22	10
Daniel	Sony Experia Z22	10
Daniel	Dell Inspiron 15 7577	20

Alex	Canon EOS 6D Mark II Body	20
------	---------------------------	----

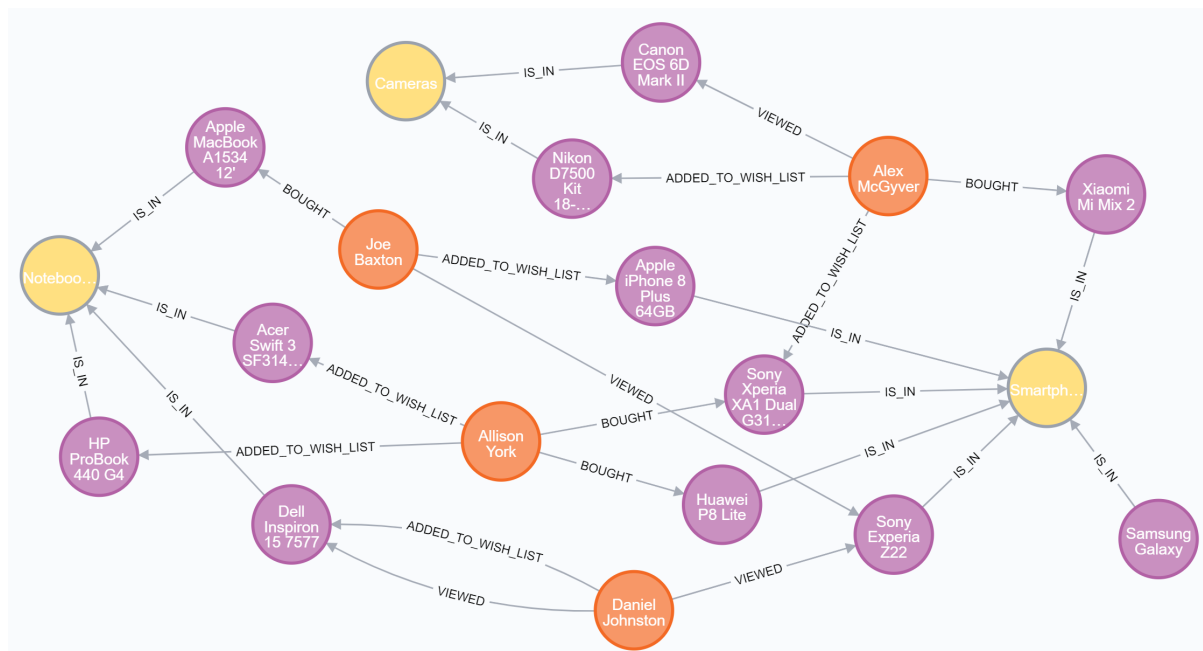
Table 4. Wishlist

Customer	Product
Joe	Apple iPhone 8 Plus 64GB
Daniel	Dell Inspiron 15 7577
Alex	Sony Xperia XA1 Dual G3112
Alex	Nikon D7500 Kit 18-105mm VR
Allison	Acer Swift 3 SF314-51-34TX
Allison	HP ProBook 440 G4

Table 5. Purchase

Customer	Product
Joe	Apple MacBook A1534 12' Rose Gold
Alex	Xiaomi Mi Mix 2
Allison	Huawei P8 Lite
Allison	Sony Xperia XA1 Dual G3112

The figure below displays the graph database you will be building in this tutorial.



Tasks:

1. Create a new project in Neo4j and name it as Tutorial9b.
2. Add a new graph. Name the graph to eShopGraph and set the password to graph.
3. After your graph has been created, start the graph then open Neo4j Browser.

4. Create the nodes for Categories. For this exercise, you need to create a separate node for each product category for the purpose of maintaining the relationship between each product and category.

```
CREATE (smartphones:Category {title: 'Smartphones'}),
(notebooks:Category {title: 'Notebooks'}),
(cameras:Category {title: 'Cameras'})
```

5. Retrieve the nodes you recently created.

```
MATCH (n) RETURN n
```

6. Delete all nodes in the database.

```
MATCH (n)
DETACH DELETE n
```

7. Recreate the nodes for Categorise and also create the nodes for products. Include the relationships between products and categories when you create the nodes.

```
// Categories
CREATE (smartphones:Category {title: 'Smartphones'}),
(notebooks:Category {title: 'Notebooks'}),
(cameras:Category {title: 'Cameras'})
```

```
// Smartphones
CREATE (sony_xperia_z22:Product {title: 'Sony Xperia Z22',
price: 765.00, shippability: true, availability: true})-
[:IS_IN]->(smartphones)
```

```
CREATE (samsung_galaxy_s8:Product {title: 'Samsung Galaxy S8',
price: 784.00, shippability: true, availability: true})-
[:IS_IN]->(smartphones)
```

```
CREATE (sony_xperia_xa1:Product {title: 'Sony Xperia XA1 Dual
G3112', price: 229.50, shippability: true, availability:
false})-[:IS_IN]->(smartphones)
```

```
CREATE (iphone_8:Product {title: 'Apple iPhone 8 Plus 64GB',
price: 874.20, shippability: true, availability: false})-
[:IS_IN]->(smartphones)
```

```
CREATE (xiaomi_mi_mix_2:Product {title: 'Xiaomi Mi Mix 2',
price: 420.87, shippability: true, availability: true})-
[:IS_IN]->(smartphones)
```

```
CREATE (huawei_p8:Product {title: 'Huawei P8 Lite', price:
191.00, shippability: true, availability: true})-[:IS_IN]-
>(smartphones)
```

```
// Notebooks
CREATE (acer_swift_3:Product {title: 'Acer Swift 3 SF314-51-34TX', price: 595.00, shippability: true, availability: false})-[:IS_IN]->(notebooks)
```

```
CREATE (hp_pro_book:Product {title: 'HP ProBook 440 G4', price: 771.30, shippability: true, availability: true})-[:IS_IN]->(notebooks)
```

```
CREATE (dell_inspiron_15:Product {title: 'Dell Inspiron 15 7577', price: 1477.50, shippability: true, availability: true})-[:IS_IN]->(notebooks)
```

```
CREATE (apple_macbook:Product {title: 'Apple MacBook A1534 12" Rose Gold', price: 1293.00, shippability: false, availability: true})-[:IS_IN]->(notebooks)
```

```
// Cameras
```

```
CREATE (canon_eos_6d:Product {title: 'Canon EOS 6D Mark II Body', price: 1794.00, shippability: true, availability: false})-[:IS_IN]->(cameras)
```

```
CREATE (nikon_d7500:Product {title: 'Nikon D7500 Kit 18-105mm VR', price: 1612.35, shippability: true, availability: true})-[:IS_IN]->(cameras)
```

8. Search and retrieve the node you have recently created.

```
MATCH (n) RETURN n
```

9. Create customer nodes.

```
CREATE (joe:Customer {name: 'Joe Baxton', email: 'joeeee_baxton@example.com', age: 25})
```

```
CREATE (daniel:Customer {name: 'Daniel Johnston', email: 'dan_j@example.com', age: 31})
```

```
CREATE (alex:Customer {name: 'Alex McGyver', email: 'mcgalex@example.com', age: 22})
```

```
CREATE (alisson:Customer {name: 'Allison York', email: 'ally_york1@example.com', age: 24})
```

10. Create the relationship for views (hint: you need to use match to find the nodes before creating the relationship).

```
MATCH (joe:Customer {name: 'Joe Baxton'})
MATCH (sony_xperia_z22:Product {title: 'Sony Xperia Z22'})
CREATE (joe)-[:VIEWED {views_count: 10}]->(sony_xperia_z22)
RETURN joe, sony_xperia_z22
```

```

MATCH (daniel:Customer {name: 'Daniel Johnston'})
MATCH (sony_xperia_z22:Product {title: 'Sony Experia Z22'})
MATCH (dell_inspiron_15:Product {title: 'Dell Inspiron 15 7577'})
CREATE (daniel)-[:VIEWED {views_count: 10}]->(sony_xperia_z22)
CREATE (daniel)-[:VIEWED {views_count: 20}]->(dell_inspiron_15)
RETURN daniel, sony_xperia_z22, dell_inspiron_15

MATCH (alex:Customer {name: 'Alex McGyver'})
MATCH (canon_eos_6d:Product {title: 'Canon EOS 6D Mark II Body'})
CREATE (alex)-[:VIEWED {views_count: 20}]->(canon_eos_6d)
RETURN alex, canon_eos_6d

```

11. Create the relationships for Wishlist.

```

MATCH (joe:Customer {name: 'Joe Baxton'})
MATCH (iphone_8:Product {title: 'Apple iPhone 8 Plus 64GB'})
CREATE (joe)-[:ADDED_TO_WISH_LIST]->(iphone_8)
RETURN joe, iphone_8

MATCH (daniel:Customer {name: 'Daniel Johnston'})
MATCH (dell_inspiron_15:Product {title: 'Dell Inspiron 15 7577'})
CREATE (daniel)-[:ADDED_TO_WISH_LIST]->(dell_inspiron_15)
RETURN daniel, dell_inspiron_15

MATCH (alex:Customer {name: 'Alex McGyver'})
MATCH (sony_xperia_xa1:Product {title: 'Sony Xperia XA1 Dual G3112'})
MATCH (nikon_d7500:Product {title: 'Nikon D7500 Kit 18-105mm VR'})
CREATE (alex)-[:ADDED_TO_WISH_LIST]->(sony_xperia_xa1)
CREATE (alex)-[:ADDED_TO_WISH_LIST]->(nikon_d7500)
RETURN alex, sony_xperia_xa1, nikon_d7500

MATCH (alisson:Customer {name: 'Allison York'})
MATCH (acer_swift_3:Product {title: 'Acer Swift 3 SF314-51-34TX'})
MATCH (hp_pro_book:Product {title: 'HP ProBook 440 G4'})
CREATE (alisson)-[:ADDED_TO_WISH_LIST]->(acer_swift_3)
CREATE (alisson)-[:ADDED_TO_WISH_LIST]->(hp_pro_book)
RETURN alisson, acer_swift_3, hp_pro_book

```

12. Create the relationships for purchase.

```

MATCH (joe:Customer {name: 'Joe Baxton'})
MATCH (apple_macbook:Product {title: "Apple MacBook A1534 12' Rose Gold"})
CREATE (joe)-[:BOUGHT]->(apple_macbook)
RETURN joe, apple_macbook

MATCH (alex:Customer {name: 'Alex McGyver'})
MATCH (xiaomi_mi_mix_2:Product {title: 'Xiaomi Mi Mix 2'})
CREATE (alex)-[:BOUGHT]->(xiaomi_mi_mix_2)

```

```

RETURN alex, xiaomi_mi_mix_2

MATCH (alisson:Customer {name: 'Allison York'})
MATCH (huawei_p8:Product {title: 'Huawei P8 Lite'})
MATCH (sony_xperia_xa1:Product {title: 'Sony Xperia XA1 Dual G3112'})
CREATE (alisson)-[:BOUGHT]->(huawei_p8)
CREATE (alisson)-[:BOUGHT]->(sony_xperia_xa1)
RETURN alisson, huawei_p8, sony_xperia_xa1

```

As you have created your database, answer the following queries:

1. Display all the customer's email.

```

MATCH (n:Customer)
RETURN n.email

```

2. Find customer whose name is Allison York.

```

MATCH (n:Customer {name:"Allison York"})
RETURN n

```

OR

```

MATCH (n:Customer)
WHERE n.name="Allison York"
RETURN n

```

3. Show Alex McGyver's age.

```

MATCH (n:Customer)
WHERE n.name="Alex McGyver"
RETURN n.age

```

4. Find all customers whose age range is in between 25-30.

```

MATCH (n:Customer)
WHERE n.age>=25 and n.age<=30
RETURN n

```

5. Display all products' name and price.

```

MATCH (n:Product)
RETURN n.title, n.price

```

6. Show all products with price over 1000.

```

MATCH (n:Product)
WHERE n.price>1000

```

```
RETURN n
```

7. Show products that are not available.

```
MATCH (n:Product)
WHERE n.availability=false
RETURN n
```

OR

```
MATCH (n:Product)
WHERE NOT n.availability=true
RETURN n
```

8. Show products that are available but not shippable.

```
MATCH (n:Product)
WHERE n.availability=true and n.shippability=false
RETURN n
```

9. Update Daniel Johnston's node to include his nickname "Dan".

```
MATCH (n:Customer {name:"Daniel Johnston"})
SET n.nickname="Dan"
RETURN n
```

10. Update Samsung Galaxy S8's price to 650.

```
MATCH (n:Product {title:"Samsung Galaxy S8"})
SET n.price=650
RETURN n
```

The End