Neo4j Database [Homework2]
ADT – FALL 2023

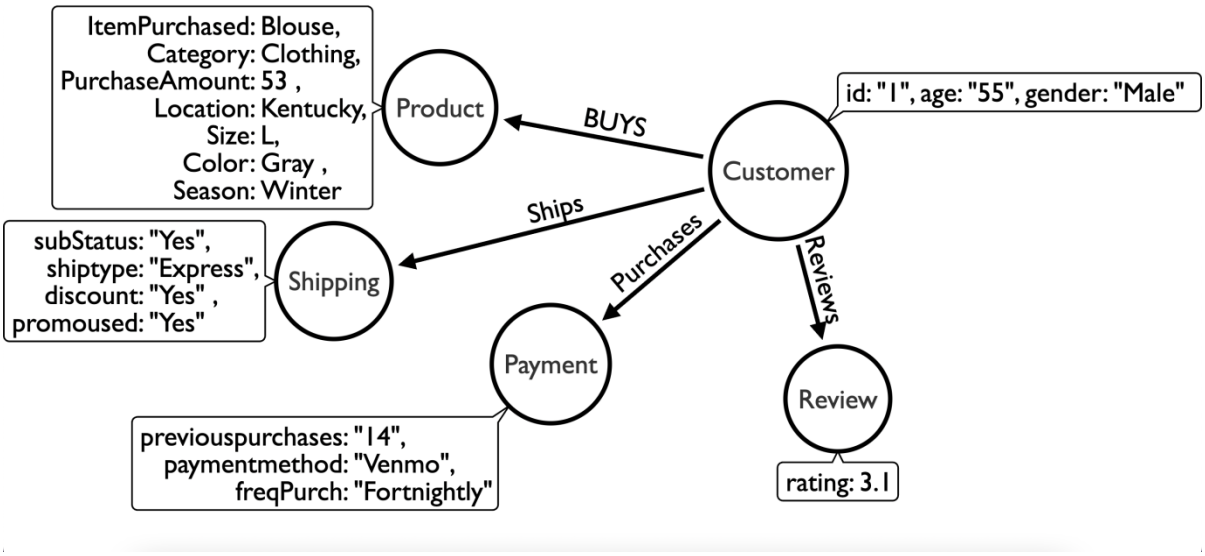
Venkata Srisai Maruthi Revanth Pasupuleti
vpasupu@iu.edu

Part 1 Data Modeling

Data Description:

I have taken the Customer Shopping Trends Dataset from Kaggle. I have used the updated trends file for this homework. The dataset is taken from the Kaggle website – url (https://www.kaggle.com/datasets/iamsouravbanerjee/customer-shopping-trends-dataset). The dataset has 3900 records, and it tells us about different properties related to shopping such as purchaseFrequency, purchase amount, payment method and shipping type and other properties. It helps us to understand different factors related to shopping habits.
There are 18 columns in the dataset namely CustomerID,Age,Gender, ItemPurchased, Category, PurchaseAmount, Location, Size, Color, Season, ReviewRating,SubscriptionStatus,ShippingType,DiscountApplied,PromoCodeUsed, PreviousPurchases, PaymentMethod and FrequencyofPurchases.

Arrow Schema Screenshot:



Part 2 Data Import

There are two ways to load the data we can first create constraints and then load the data or not have any constraints and directly load the data. In this homework I have created few constraints and then loaded the data into nodes and relationships. The node has different properties which are the attributes of the dataset. The relationships are used to establish relation between two nodes.

Constraints used for this dataset:

CREATE CONSTRAINT for (c:Customer) require c.id IS UNIQUE
This is used to check if all the values are unique.

CREATE CONSTRAINT FOR (p:Product) require p.category is not null
This constraint is used to check if the category column is not null.

CREATE CONSTRAINT FOR (p:Product) require p.purchaseAmount is not null
This constraint is used to check purchaseAmount column is null.

Nodes Created:
The nodes created are Customer, Payment, Product, Review, Shipping.

Customer node has the following properties id, gender, and age.

Product node has the following properties ItemPurchased, category, purchaseAmount, Location, Size, Color, Season.

Review node has the following property rating.

Shipping node has the following properties subStatus which is subscriptionStatus, shiptype(shipping type),discount(discountApplied), promoused (promo is used).

Payment node has the following properties previouspurchases, paymentmethod, freqPurch(frequencyofpurchasesbythecustomer).

Relationships Used:

BUYS: This relationship is between customer and product nodes.

Purchases: This relationship is between customer and payment nodes.

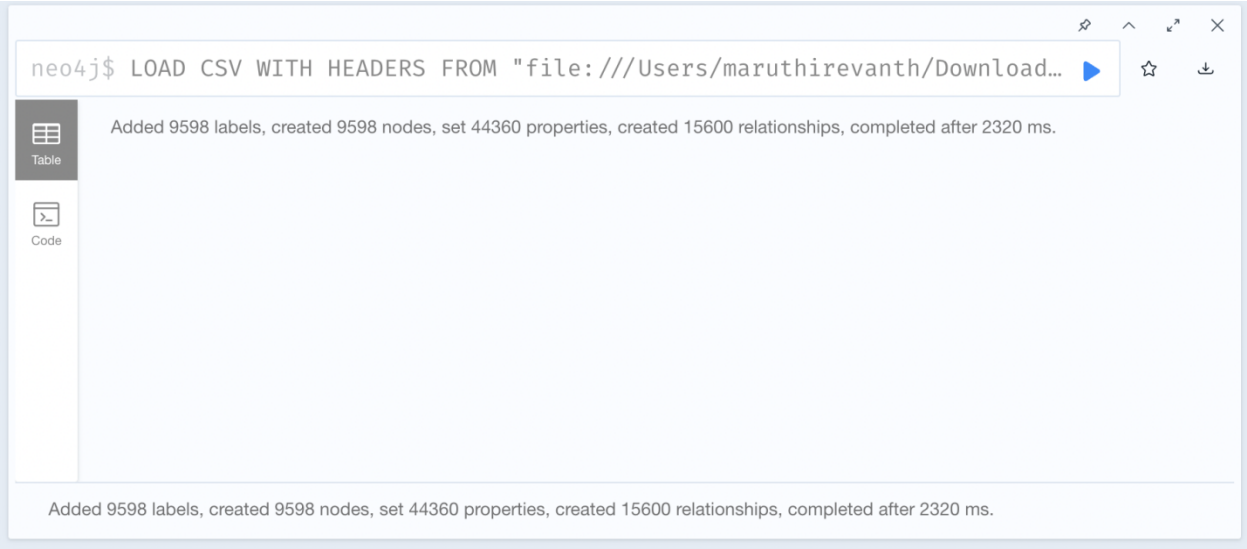
Reviews: This relationship is between customer and review nodes.

Ships: This relationship is between customer and shipping nodes.

Code for loading data and creating relationships:

```
//Loading data from csv file.
LOAD CSV WITH HEADERS FROM
"file:///Users/maruthirevanth/Downloads/dataset/shopping_trends_updated.csv" AS row

//Create customer node.
MERGE (customer:Customer {id: row.CustomerID, age: row.Age, gender: row.Gender})

//Create product node
MERGE (product:Product {
ItemPurchased: row.ItemPurchased,
category: row.Category,
purchaseAmount: row.PurchaseAmount,
Location: row.Location,
Size: row.Size,
```

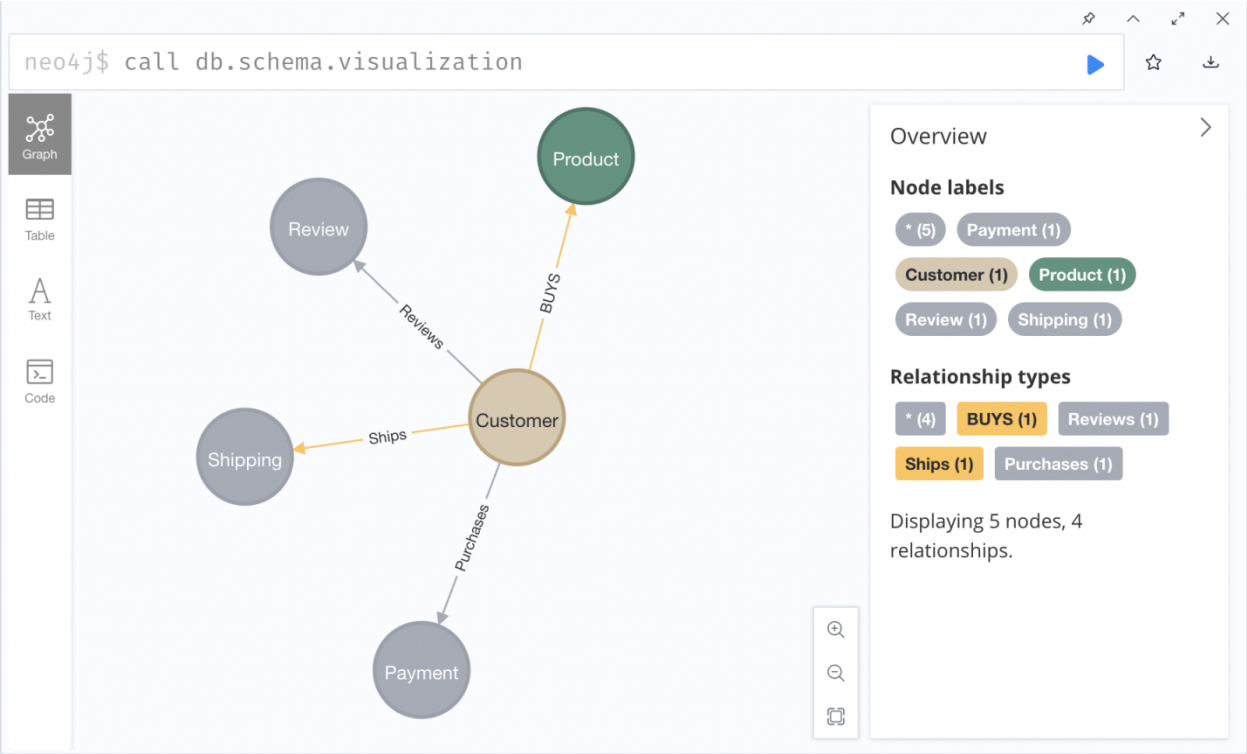
```
Color: row.Color,
Season: row.Season
})
//Create review node
MERGE (review:Review {rating: toFloat(row.ReviewRating)})

//Create Shipping node
MERGE (shipping:Shipping {
subStatus: row.SubscriptionStatus,
shiptype: row.ShippingType,
discount: row.DiscountApplied,
promoused: row.PromoCodeUsed
})


// Create payment node.
MERGE (payment:Payment {
previouspurchases: row.PreviousPurchases,
paymentmethod: row.PaymentMethod,
freqPurch: row.FrequencyofPurchases
})

//Create relationship buys
MERGE (customer)-[:BUYS]->(product)
//Create relationship reviews
MERGE (customer)-[:Reviews]->(review)
//Create relationship ships
MERGE (customer)-[:Ships]->(shipping)
//Create relationship purchases
MERGE (customer)-[:Purchases]->(payment)
```



Part 3 Graph Exploration

Neo4j schema Figure (db.schema.visualization)

call db.schema.visualization



There are totally 5 nodes customer, product, review, shipping, payment and 4 relationships those are buys between customer and product, reviews between customer and review, ships is between customer and shipping, purchases between customer and payment.

Queries and Screenshots for exploration:

1.) Find the number of items purchased in a particular colour

Query) MATCH (p:Product) RETURN p.Color,count(p.ItemPurchased) as itemspurchasedInThatColour

	p.Color	itemspurchasedInThatColour
1	"Gray"	159
2	"Maroon"	158
3	"Turquoise"	145
4	"White"	142
5	"Charcoal"	153
6	"Silver"	173
7		

2.) Find the number of orders by each location and by gender.

Query) MATCH (customer:Customer)-[:BUYS]->(product:Product) RETURN product.Location, customer.gender, COUNT(*) AS countOFNumbers order by product.Location

```
MATCH (customer:Customer)-[:BUYS]→(product:Product)
RETURN  product.Location, customer.gender, COUNT(*) AS
countOFNumbers
order by product.Location
```

	product.Location	customer.gender	countOFNumbers
1	"Alabama"	"Male"	62
2	"Alabama"	"Female"	27
3	"Alaska"	"Male"	48
4	"Alaska"	"Female"	24
5	"Arizona"	"Male"	40
6	"Arizona"	"Female"	25
7			

Started streaming 100 records after 1 ms and completed after 7 ms.

3.) Find the number of customer using each paymentMethod.

Query) MATCH (customer:Customer)-[:Purchases]->(payment:Payment) RETURN payment.paymentmethod as PaymentType, COUNT(DISTINCT customer) AS NumberOfCustomerUsingthepayment

```
MATCH (customer:Customer)-[:Purchases]→(payment:Payment)
RETURN payment.paymentmethod as PaymentType, COUNT(DISTINCT
customer) AS NumberOfCustomerUsingthepayment
```

	PaymentType	NumberOfCustomerUsingthepayment
1	"Venmo"	634
2	"Cash"	670
3	"Credit Card"	671
4	"PayPal"	677
5	"Bank Transfer"	612
6	"Debit Card"	636

Started streaming 6 records after 1 ms and completed after 8 ms.

4.) Find the number of orders of winter clothing(category=clothing and season=winter)

Query) MATCH (customer:Customer)-[:BUYS]->(product:Product) where product.Season="Winter" and product.category="Clothing" RETURN product.Season, product.category, COUNT(customer.id) AS numberOFCustomersPurchasedWinterClothing;

```
MATCH (customer:Customer)-[:BUYS]→(product:Product)
where product.Season="Winter" and product.category="Clothing"
RETURN  product.Season, product.category, COUNT(customer.id) AS
numberOFCustomersPurchasedWinterClothing;
```

	product.Season	product.category	numberOFCustomersPurchasedWinterClothing
1	"Winter"	"Clothing"	448

Started streaming 1 records after 5 ms and completed after 8 ms.

5.) Find the number of customers using the faster shipping types('Express', '2-Day Shipping', 'Next Day Air')

Query) MATCH (customer:Customer)-[:Ships]->(shipping:Shipping) WHERE shipping.shiptype IN ['Express', '2-Day Shipping', 'Next Day Air'] RETURN shipping.shiptype, COUNT(DISTINCT customer.id) AS shippingTypeCount

```
MATCH (customer:Customer)-[:Ships]→(shipping:Shipping)
WHERE shipping.shiptype IN ['Express', '2-Day Shipping', 'Next Day
Air']
RETURN shipping.shiptype, COUNT(DISTINCT customer.id) AS
shippingTypeCount
```

	shipping.shiptype	shippingTypeCount
1	"Express"	646
2	"Next Day Air"	648
3	"2-Day Shipping"	627

Started streaming 3 records after 1 ms and completed after 4 ms.

6.) Display the customer id, color and size of the product together accordingly using concat functionality similar operator.

Query) MATCH (c:Customer)-[:BUYS]->(p:Product) RETURN 'Customer '+c.id+' has purchased ' AS customer,'Item '+p.ItemPurchased +' of color '+ p.Color+' of size ' + p.Size + ' ' AS answer;

```
MATCH (c:Customer)-[:BUYS]→(p:Product)
RETURN 'Customer '+c.id+' has purchased ' AS customer,'Item
'+p.ItemPurchased +' of color '+   p.Color+' of size ' + p.Size + '
' AS answer;
```

Table | Text | Code

	customer	answer
1	"Customer 1 has purchased "	"Item Blouse of color Gray of size L "
2	"Customer 2 has purchased "	"Item Sweater of color Maroon of size L "
3	"Customer 3 has purchased "	"Item Jeans of color Maroon of size S "
4	"Customer 4 has purchased "	"Item Sandals of color Maroon of size M "
5	"Customer 5 has purchased "	"Item Blouse of color Turquoise of size M "
6	"Customer 6 has purchased "	"Item Sneakers of color White of size M "
7		

Started streaming 3900 records after 31 ms and completed after 54 ms, displaying first 1000 rows.