SQL-Mongo Project – Supermarket Sales

Contents

Relational Data Model

Assumptions/Notes About Data Entities and Relationships

Include assumptions about data entities and their relationships with each other.

Data Requirements:

1. Database needs to store details about each invoice being generated through transactions.
2. Database needs to store details about their branches.
3. Database needs to store every Invoice-ID and is assigned to one Unique Customer-ID.
4. In Every Product Line, unique unit price represents one unique Product –ID.
5. Database needs to store details about the customer (Customer ID, Gender)
6. Identify the type of the customer (Normal, Member) for each order.
7. Database needs to store details about the product line and identify each type of product line the product belongs to.
8. Database needs to identify the tax amount for each invoice total and store it with Invoice ID
9. Database needs to store date and time for each unique transaction.
10. Company needs to identity the payment type been used for each unique transaction and store it in the database.
11. Database needs to store the cost of goods sold for per invoice.
12. Company needs to recognize the gross margin of each transaction and stored in database.
13. Database needs to store satisfaction rating for each customer.

Data entities and their relationships:

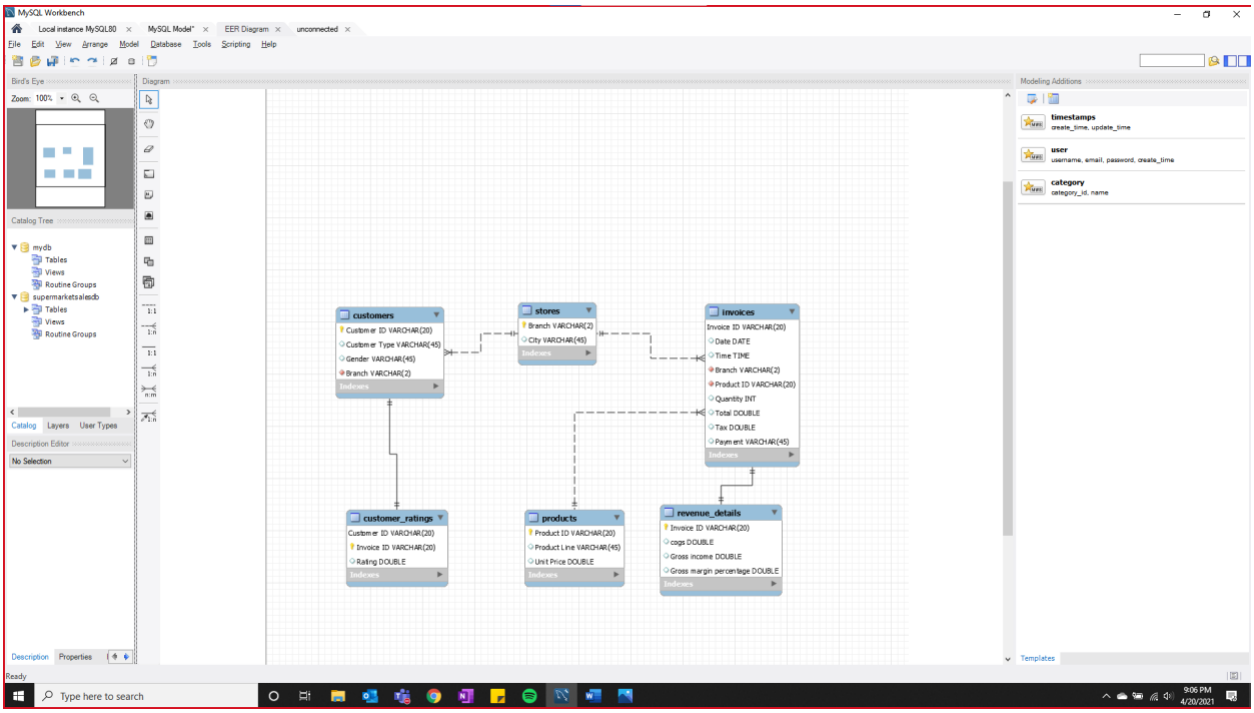
1. Stores have a 1:M non-identifying relationship with customers.
2. Stores have a 1:M non-identifying relationship with invoices.
3. Customers have 1:1 identifying relationship with customer_ratings.
4. Products have a 1:M non-identifying relationship with invoices.
5. Revenue_details have 1:1 identifying relationship with invoices.

Include reasons why the data model is in 3NF:

1. Every entity has unique and clear name.
2. Every attribute is directly related to the entity.

3. Every attribute has clearly understood name and is unique inside the entity.
4. Every attribute only holds one value that is indivisible.
5. Every entity has unique identifier (primary key/composite primary key).
6. All non-key columns have functional dependency on the entire primary key and only on the entire primary key.

Entity-Relationship Diagram



Physical MySQL Database

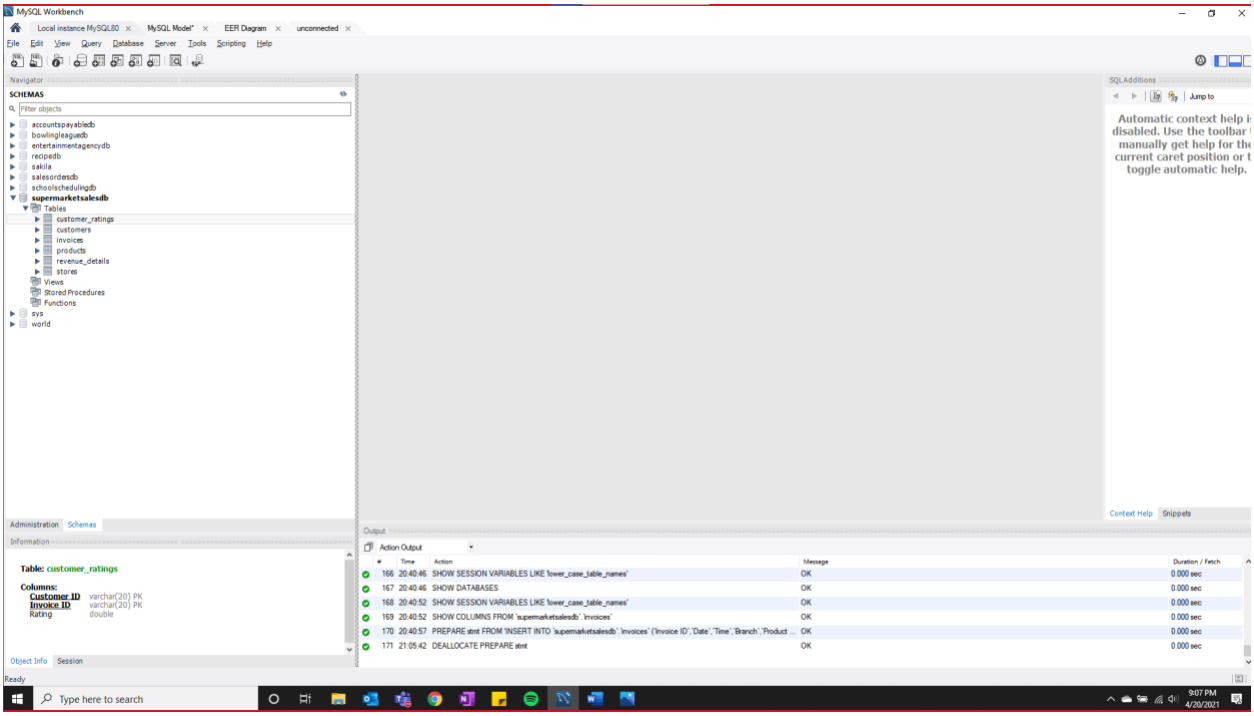
Assumptions/Notes About Data Set

Include any assumptions made about data such as empty fields, sparse data, bad data, etc.

Assumptions:

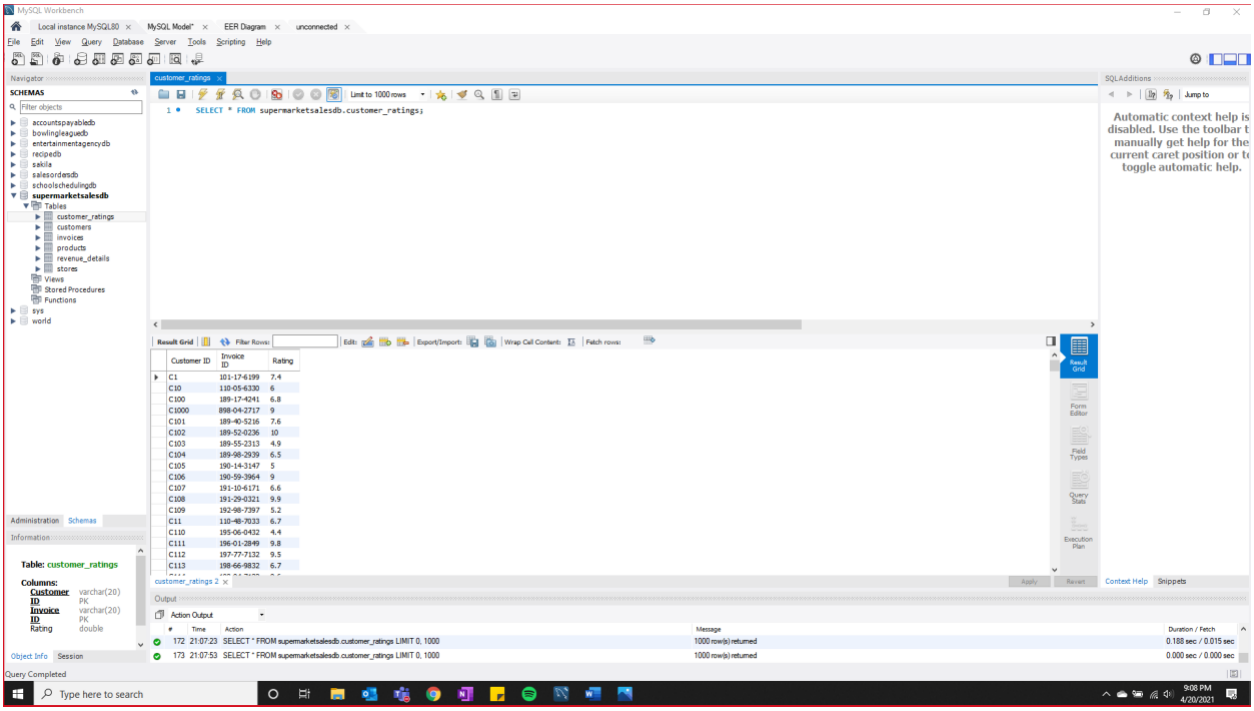
1. There is a unique customer for every invoice.
2. There is a unique invoice for every transaction.
3. A store can have many customers.
4. A store can have many invoices.
5. A product can have many invoices.
6. An invoice has one revenue detail.
7. A customer has one customer rating.
8. The value of tax is unrelated to total.
9. The value of total is unrelated to quantity and unit price.
10. The value of cogs is independent.
11. Gross margin percentage is an independent attribute.
12. Gross income is an independent attribute.

Screen shot of Physical Database objects

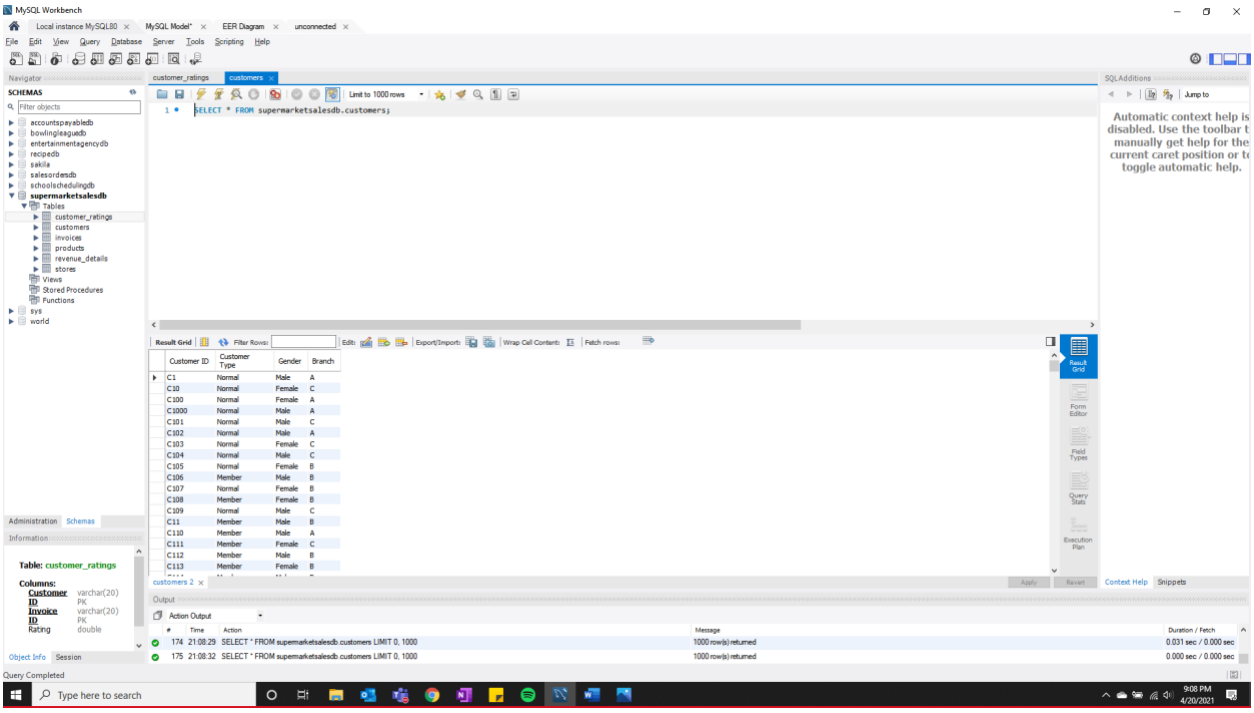


To check row counts of data loaded into each table:

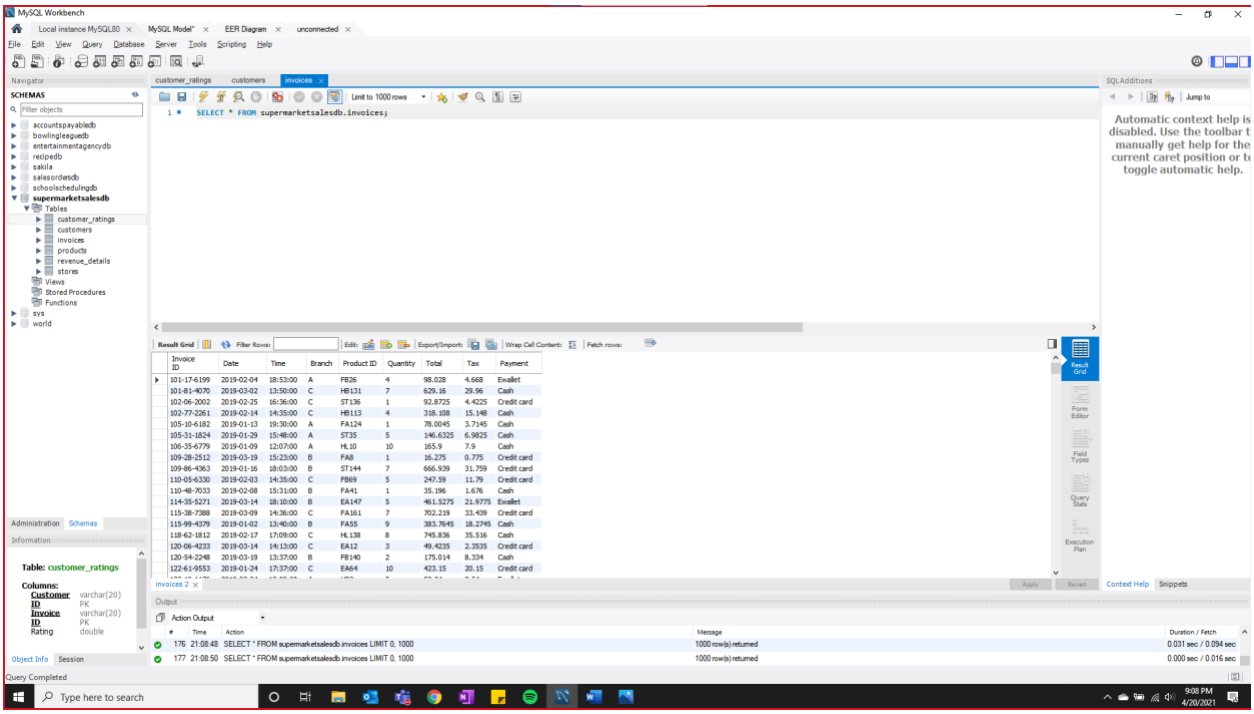
customer ratings:



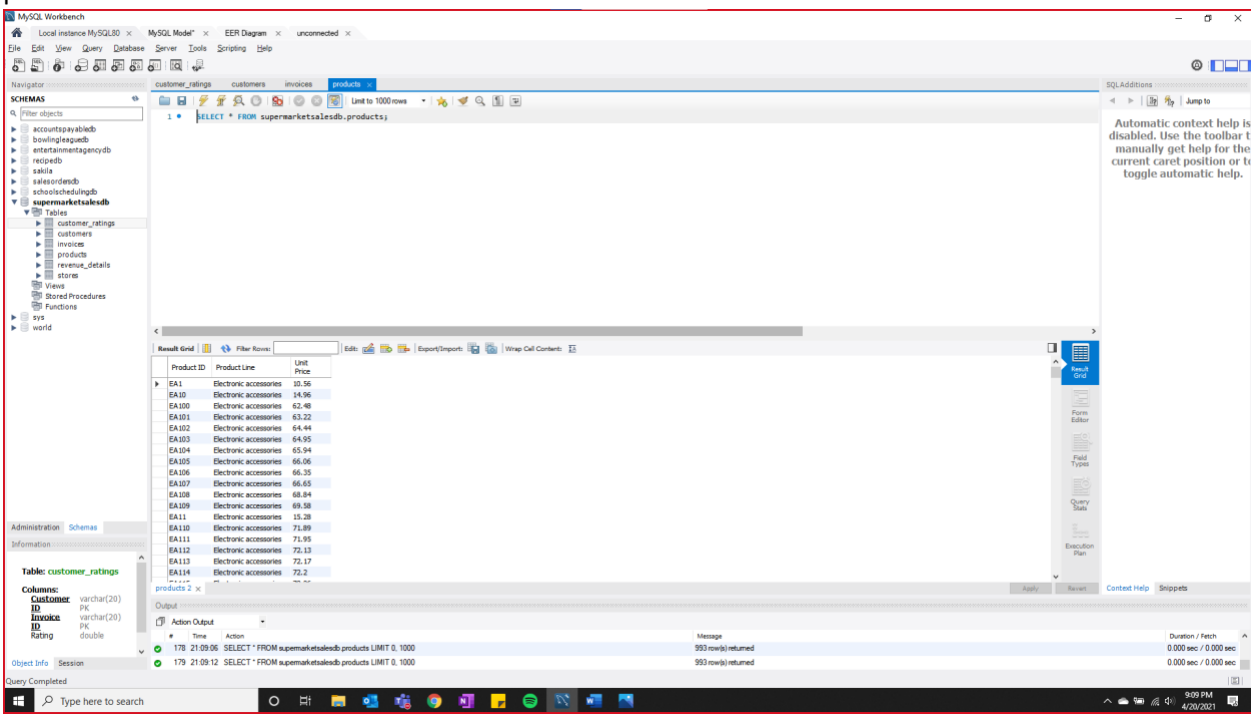
customers:



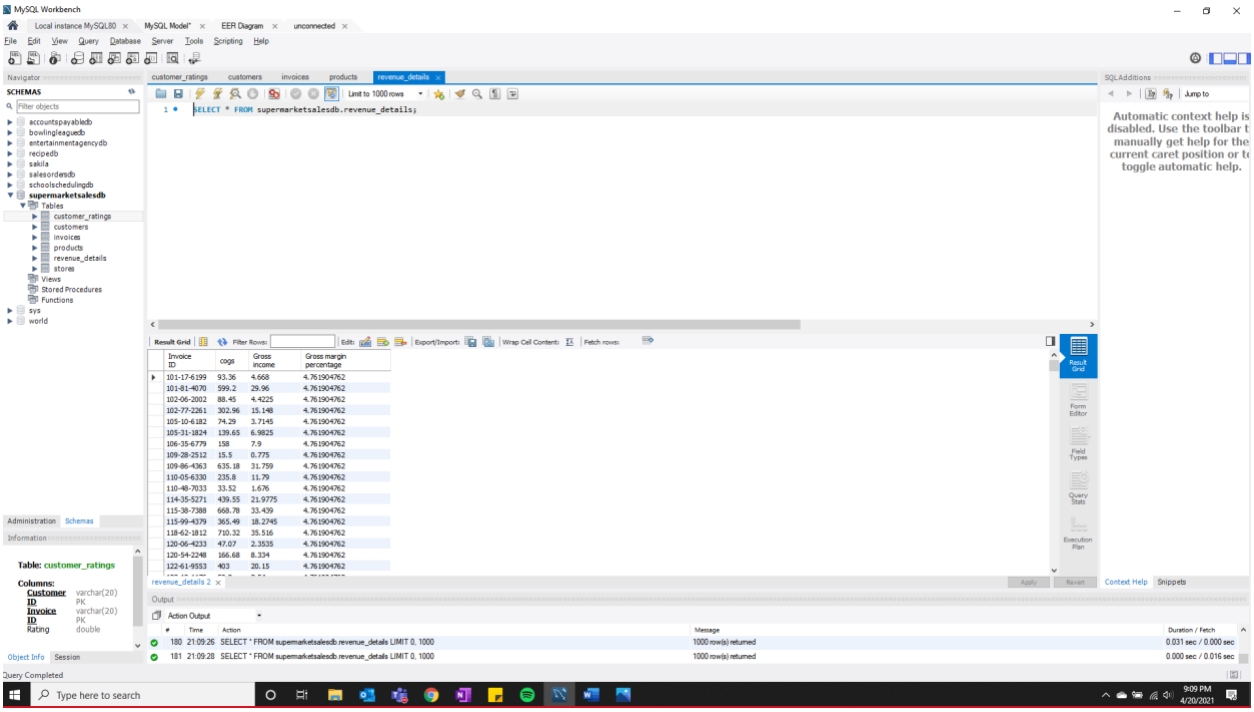
invoices:



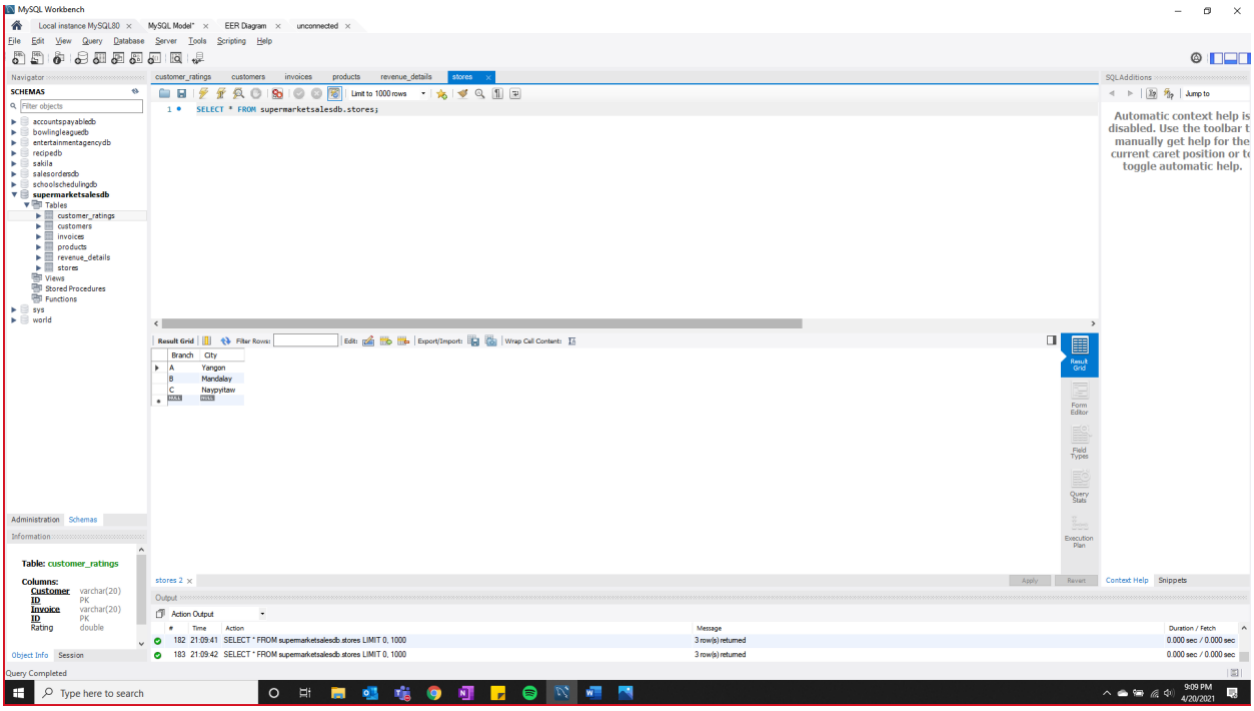
products:



revenue details:



stores:



Data in the Database

Table Name	Primary Key	Foreign Key	# of Rows in Table
customer_ratings	Customer ID, Invoice ID		1000
customers	Customer ID	Branch	1000
invoices	Invoice ID	Branch, Product ID	1000
products	Product ID		993
revenue_details	Invoice ID		1000
stores	Branch		3

SQL Queries

SQL Query 1

Question (Q1)

Some retailers believe that there is more money to be made in selling fashion accessories to men than sports and travel to women. Is this true?

Notes/Comments About SQL Query and Results (Include # of Rows in Result)

Explanation:

First, we find the sum of Gross income for 2 different categories, i.e., 'fashion accessories sold to men' and 'sports and travel sold to women'. We then compare its values to see which is greater.

Assumptions:

We assume the value of 'gross income' denotes the money made on a purchase. Hence, summing the value of gross income for the 2 categories 'fashion accessories sold to men' and 'sports and travel sold to women' can be used to compare which category makes the most money.

# of Rows in Result: 1

Conclusion:

From the result of the query mentioned below, we can see that the sum of gross income for the category 'fashion accessories sold to men' is $1136.595 compared 'sports and travel sold to women' which is $1360.70. Hence the belief of some retailers that there is more money to be made in selling fashion accessories to men than sports and travel to women is FALSE.

Translation

Translation:

Select sum of Gross Income for the case when gender is male and Product line is Fashion Accessories and for the case when gender is female and Product line is Sports and Travel from Customer table joined with Customer Ratings table where Customer ID in Customer table matches Customer ID in Customer Ratings table joined with revenue details table where Invoice ID in revenue details table match with invoice in customer ratings table joined with invoices table where invoice ID in invoices table matches with invoice id in revenue details table joined with products table where product id in products table matches with product id in invoices table.

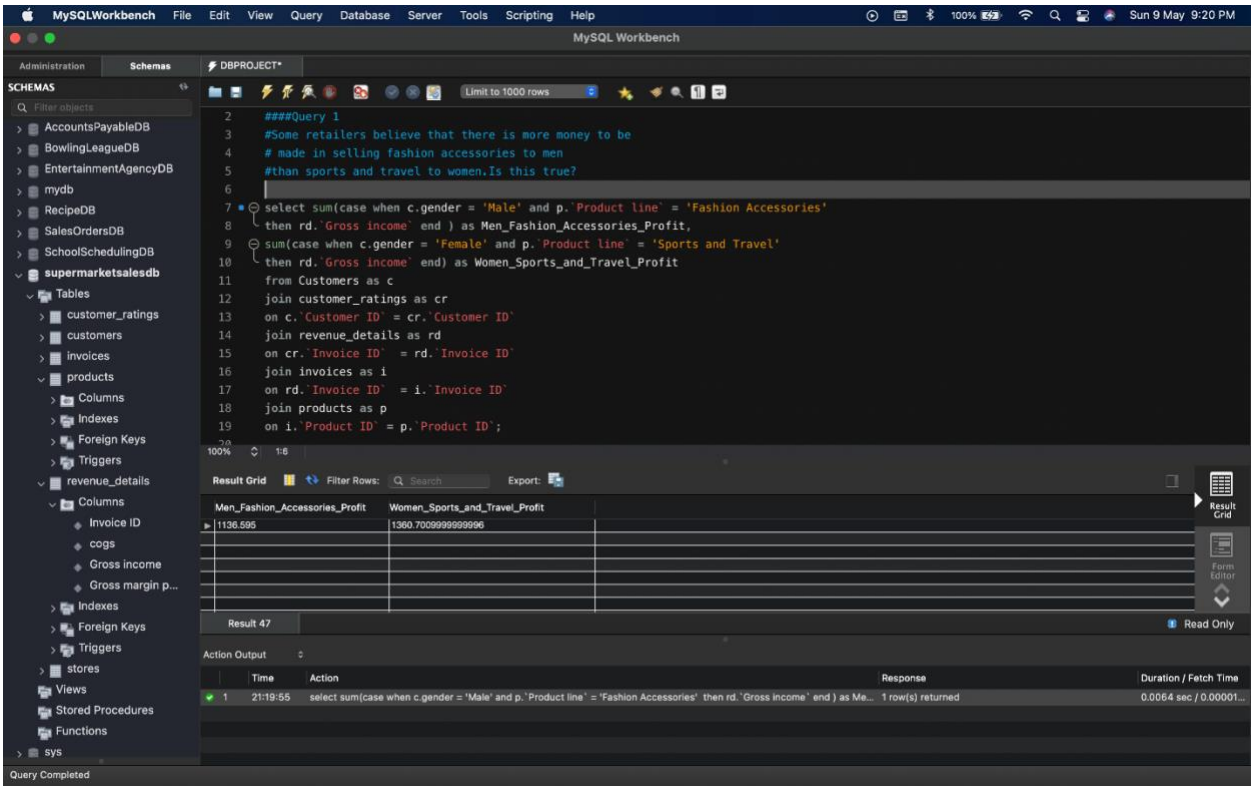
Cleanup:

select sum(case when c.gender = 'Male' and p.'Product line' = 'Fashion Accessories' then rd.'Gross income' end) as Men_Fashion_Accessories_Profit, sum(case when c.gender = 'Female' and p.'Product line' = 'Sports and Travel' then rd.'Gross income' end) as Women_Sports_and_Travel_Profit from Customers as c join customer_ratings as cr on c.'Customer ID' = cr.'Customer ID' join revenue_details as rd on cr.'Invoice ID' = rd.'Invoice ID' join invoices as i on rd.'Invoice ID' = i.'Invoice ID' join products as p on i.'Product ID' = p.'Product ID'

Query:

```
select sum(case when c.gender = 'Male' and p.’Product line’ = 'Fashion Accessories'
then rd.’Gross income’ end ) as Men_Fashion_Accessories_Profit,
sum(case when c.gender = 'Female' and p.’Product line’ = 'Sports and Travel'
then rd.’Gross income’ end) as Women_Sports_and_Travel_Profit
from Customers as c
join customer_ratings as cr
on c.’Customer ID’ = cr.’Customer ID’
join revenue_details as rd
on cr.’Invoice ID’ = rd.’Invoice ID’
join invoices as i
on rd.’Invoice ID’ = i.’Invoice ID’
join products as p
on i.’Product ID’ = p.’Product ID’;
```

Screen Shot of SQL Query and Results



SQL Query 2

Question (Q.2)

Some retailers believe that revenue in food and beverages can be increased amongst women by focusing on Ewallets, while others believe eWallets are more popular with men buying electronic accessories. Who is right?

Notes/Comments About SQL Query and Results (Include # of Rows in Result)

Explanation:

First, we find the sum of 'Total' for the 2 categories compared i.e., 'food and beverages sold to women' and 'electronic accessories sold to men' grouped by the different payment types such as Ewallet, Credit card and cash. We can use this result to compare the revenue generated for 'food and beverages sold to women' and 'electronic accessories sold to men' amongst different payment types and also see if Ewallet is a popular choice.

Assumptions:

Revenue is the total amount of income generated by the sale of goods or services related to the company's primary operations. We assume the value of 'total' denotes the revenue of a purchase. Hence, summing the value of total for the 2 categories 'food and beverages sold to women' and 'electronic accessories sold to men' can be used to compare revenue generated by different payment types.

# of Rows in Result: 3

Conclusion:

From the results of the query mentioned below, we can see that the sum of total for 'food and beverages sold to women' is least for payments made using Ewallets compared to credit cards and cash.

Next, we can see that sum of total for 'electronic accessories sold to men' is maximum for payments made using Ewallets.

Hence, the claim made by some retailers that revenue in food and beverages can be increased amongst women by focusing on Ewallets is FALSE, while the claim that eWallets are more popular with men buying electronic accessories is TRUE.

Translation

Translation:

select payment, sum of total for the case when Product line is Food and Beverages and gender is Female as Food_and_Beverages_Women_Revenue, sum of total for the case when Product line is Electronic Accessories and gender is Male as Electronic_Accessories_Men_Revenue from Customers table joined with customer ratings table when Customer ID from customers matches Customer ID from customer ratings table joined with revenue details table where Invoice ID from customer ratings matches Invoice ID from revenue details joined with invoices table where Invoice ID from revenue details matches Invoice ID from invoices table joined with products table where Product ID from invoices matches Product ID from products grouped by Payment

Cleanup:

select i.payment Payment_Method, sum(case when p.’Product line’ = 'Food and Beverages' and c.gender = 'Female'then i.Total end) as Women_Food_and_Beverages_Revenue, sum(case when p.’Product line’ = 'Electronic Accessories' and c.gender = 'Male' then i.Total end) as Men_Electronic_Accessories_Revenue from Customers as c join customer_ratings as cr on c.’Customer ID’ = cr.’Customer ID’ join revenue_details as rd on cr.’Invoice ID’ = rd.’Invoice ID’ join invoices as i on rd.’Invoice ID’ = i.’Invoice ID’ join products as p on i.’Product ID’ = p.’Product ID’ group by i.Payment

Query:

select i.payment Payment_Method, sum(case when p.’Product line’ = 'Food and Beverages'

and c.gender = 'Female'then i.Total end) as Women_Food_and_Beverages_Revenue,

sum(case when p.’Product line’ = 'Electronic Accessories'

and c.gender = 'Male' then i.Total end) as Men_Electronic_Accessories_Revenue

from Customers as c

join customer_ratings as cr

on c.’Customer ID’ = cr.’Customer ID’

join revenue_details as rd

on cr.’Invoice ID’ = rd.’Invoice ID’

join invoices as i

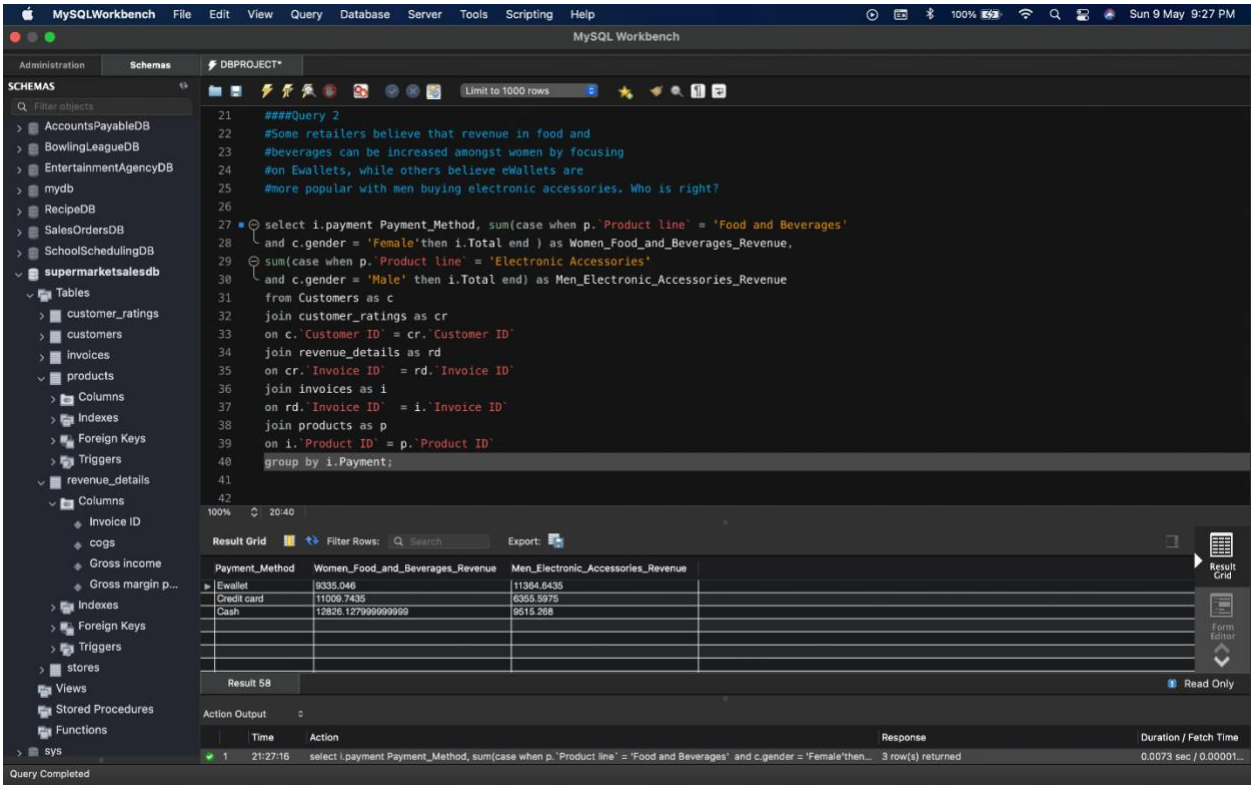
on rd.’Invoice ID’ = i.’Invoice ID’

join products as p

on i.’Product ID’ = p.’Product ID’

group by i.Payment

Screen Shot of SQL Query and Results



SQL Query 3

Question (Q.3)

Some retailers believe payment method is a bigger indicator of health and beauty purchases while other retailers believe gender is a bigger factor. Who is right?

Notes/Comments About SQL Query and Results (Include # of Rows in Result)

Explanation:

For the Product line 'Health and Beauty', first we find the count of purchases made for different payment methods. Next, we find the count of purchases made for different gender. We can use this result to see which of the two (payment method, gender) play a bigger factor for health and beauty purchases.

Assumptions:

'Count' of purchases for different methods or gender can be used to determine which of the two plays a bigger factor in the purchase of health and beauty products.

# of Rows in Result: 6

Conclusion:

From the result of the query mentioned below, for the product line 'health and beauty' we can see that when we consider the factor as payment method 34.9% (53 of 152) of purchases are made with Ewallets, 32.2% (49 of 152) of purchases are made with cash and 32.9% (50 of 152) of purchases are made with credit cards.

When we consider the factor as gender 42.1% (64 of 152) of purchases are made by women, 57.9% (88 of 152) of purchases are by men.

We can conclude that GENDER is a bigger indicator of health and beauty purchases as we can clearly see that men make majority of purchases in this product line compared to women. We cannot say the same about payment types as all the payment methods have approximately the same share of purchases.

Translation

Translation:

select distinct payments as Factor, count of payments as Health_and_Beauty_Count grouped by different payment types from invoices joined with products on Product ID from invoices matching Product ID from products table where Product Line is Health and Beauty union with select "--", "--" union with select distinct Gender as Factor, count of payments as Health_and_Beauty_Count grouped by Gender from customers table joined with customer_ratings table on Customer ID from customer matching Customer ID from customer ratings joined with invoices table on Invoice ID from customer ratings Invoice ID from invoices table joined with products table on Product ID from invoices table matching Product ID from products table where Product Line is Health and Beauty

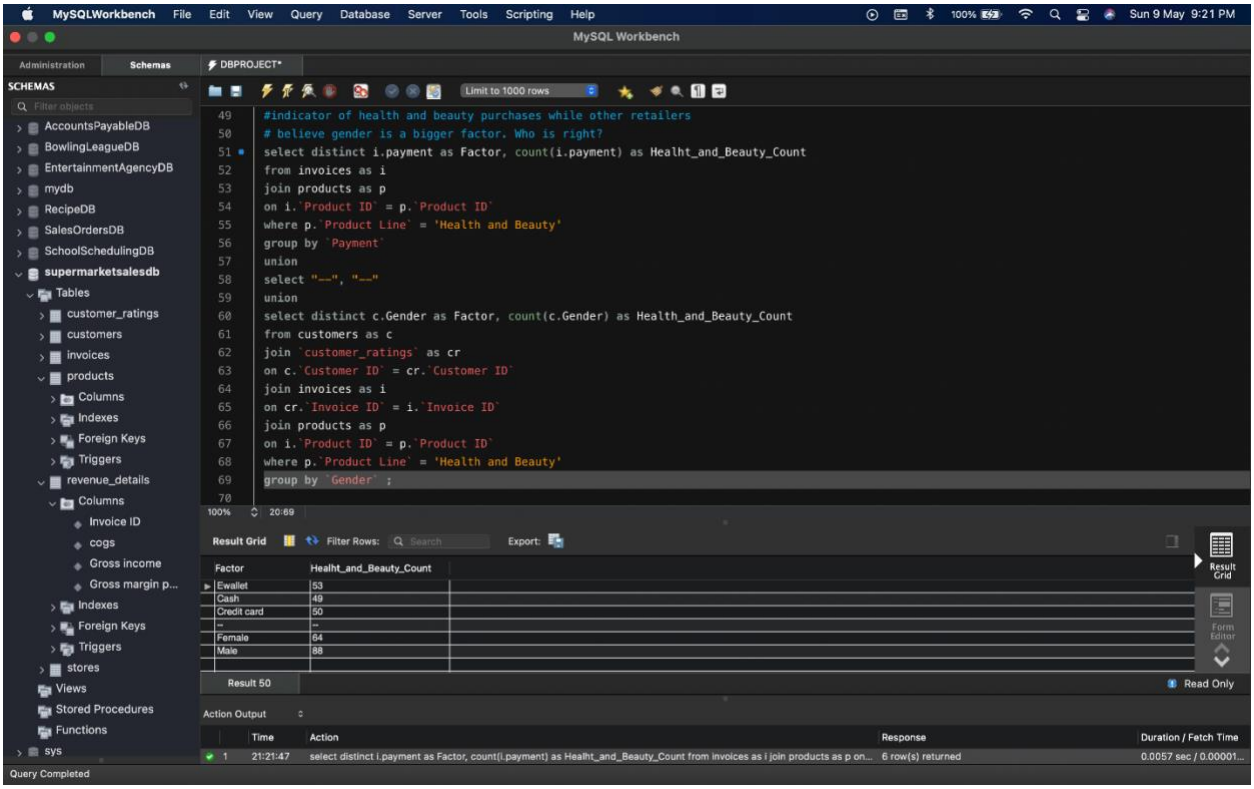
Cleanup:

select distinct i.payment as Factor, count(i.payment) as Healht_and_Beauty_Count from invoices as I join products as p on i.’Product ID’ = p.’Product ID’ where p.’Product Line’ = 'Health and Beauty' group by ‘Payment’ union select "--", "--" union select distinct c.Gender as Factor, count(c.Gender) as Healht_and_Beauty_Count from customers as c join ‘customer_ratings’ as cr on c.’Customer ID’ = cr.’Customer ID’ join invoices as i on cr.’Invoice ID’ = i.’Invoice ID’ join products as p on i.’Product ID’ = p.’Product ID’ where p.’Product Line’ = 'Health and Beauty' group by ‘Gender’

Query:

```
select distinct i.payment as Factor, count(i.payment) as Healht_and_Beauty_Count

from invoices as i

join products as p

on i.’Product ID’ = p.’Product ID’

where p.’Product Line’ = 'Health and Beauty'

group by ‘Payment’

union

select "--", "--"

union

select distinct c.Gender as Factor, count(c.Gender) as Health_and_Beauty_Count

from customers as c

join ‘customer_ratings’ as cr

on c.’Customer ID’ = cr.’Customer ID’

join invoices as i

on cr.’Invoice ID’ = i.’Invoice ID’

join products as p

on i.’Product ID’ = p.’Product ID’

where p.’Product Line’ = 'Health and Beauty'

group by ‘Gender’
```

Screen Shot of SQL Query and Results



SQL Query 4

Question (Q.4)

Some retailers believe that their members are spending more per purchase while members believe they are spending less per purchase. Who is right?

Notes/Comments About SQL Query and Results (Include # of Rows in Result)

Explanation:

First, we find the average of total where customer type is Member. Next, we find the average of total where customer type is Normal. We then compare the averages to find if members or non-members are spending more per purchase

Assumptions:

Revenue is the total amount of income generated by the sale of goods or services related to the company's primary operations. We assume the value of 'total' denotes spending of a purchase. Hence, we find the average of total for members and non-members to compare their average spending per purchase.

# of Rows in Result: 1

Conclusion:

The result of the query shown below clearly shows that members spent on average $327.8 per purchase and non-members spent on average $318.12 per purchase. Hence, some retailers who believe that their members are spending more per purchase are right and members who believe they are spending less per purchase are wrong.

Translation

Translation:

select average of total for the case when Customer Type is Member as Avg_Members_Spending, average of total for the case when Customer Type is Normal as Avg_Non_Members_Spending from Customers table joined with customer_ratings on Customer ID from customer matching Customer ID from customer ratings joined with revenue_details on Invoice ID from credit ratings matching Invoice ID from revenue details joining invoices on matching Invoice ID from revenue details matching Invoice ID from invoices joining products table on Product ID from invoices matching Product ID in products table.

Cleanup:

select avg(case when c.'Customer Type' = 'Member' then i.Total end) as Avg_Members_Spending, avg(case when c.'Customer Type' = 'Normal' then i.Total end) as Avg_Non_Members_Spending from Customers as c join customer_ratings as cr on c.'Customer ID' = cr.'Customer ID' join revenue_details as rd on cr.'Invoice ID' = rd.'Invoice ID' join invoices as i on rd.'Invoice ID' = i.'Invoice ID' join products as p on i.'Product ID' = p.'Product ID'

Query:

select avg(case when c.’Customer Type’ = 'Member'

then i.Total end) as Avg_Members_Spending,

avg(case when c.’Customer Type’ = 'Normal'

then i.Total end) as Avg_Non_Members_Spending

from Customers as c

join customer_ratings as cr

on c.’Customer ID’ = cr.’Customer ID’

join revenue_details as rd

on cr.’Invoice ID’ = rd.’Invoice ID’

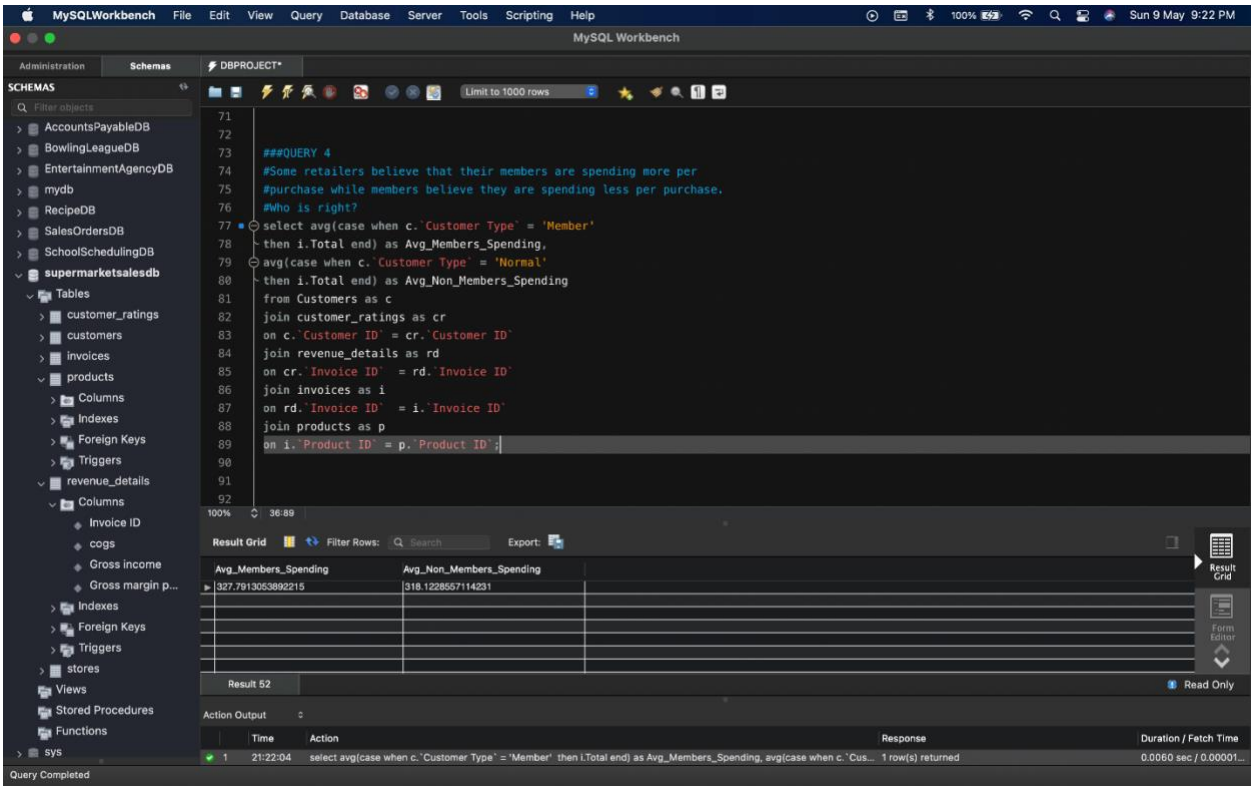
join invoices as i

on rd.’Invoice ID’ = i.’Invoice ID’

join products as p

on i.’Product ID’ = p.’Product ID’

Screen Shot of SQL Query and Results



SQL Query 5

Question (Q.5)

Some retailers believe that their male members are bringing in more overall revenue per purchase while others believe female non-members are bringing in more revenue per purchase of fashion accessories. Who is right?

Notes/Comments About SQL Query and Results (Include # of Rows in Result)

Explanation:

First, we find the average of 'total' for the category 'fashion accessories sold to male members'.

Next, we find the average of 'total' for the category 'fashion accessories sold to female non-members'.

Then we compare these averages to find if male members are bringing in more overall revenue per purchase or female non-members are bringing in more revenue per purchase of fashion accessories.

Assumptions:

We assume that we are comparing the overall revenue per purchase for the fashion accessories sold to male members and fashion accessories sold to female non-members.

Revenue is the total amount of income generated by the sale of goods or services related to the company's primary operations. We assume the value of 'total' denotes the revenue of a purchase.

Hence, we find the average of total for the 2 categories 'fashion accessories sold to male members' and 'fashion accessories sold to female non-members' to find which group brings in more revenue per purchase of fashion accessories.

# of Rows in Result: 1

Conclusion:

The result of the query shown below clearly shows that the average of 'total' for the category 'fashion accessories sold to male members' is $287.21 as compared to 'total' for the category 'fashion accessories sold to female non-members' which is $312.55.

Hence, we can conclude that the claim made by some retailers that their male members are bringing in more overall revenue per purchase is FALSE and the claim that female non-members are bringing in more revenue per purchase of fashion accessories is TRUE.

Translation

Translation:

select average of total for the case where Product line is Fashion Accessories and Customer Type is Member and gender is Male as Avg_Male_Member_Fashion_Accessories_Revenue, average of total for the case where Product line is Fashion Accessories and Customer Type is Normal and gender is Female as Avg_Female_Non_Member_Fashion_Accessories_Revenue from Customers table joined with customer ratings table on Customer ID from customers matching Customer ID from customer ratings joined with revenue details table on Invoice ID from customer ratings matching Invoice ID from revenue details joined

with invoices table on Invoice ID from revenue details matching Invoice ID from invoices joined with products on Product ID from invoices matching Product ID from products

Cleanup:

select avg(case when p.’Product line’ = 'Fashion Accessories' and c.’Customer Type’ = 'Member' and c.gender = 'Male' then i.Total end) as Avg_Male_Member_Fashion_Accessories_Revenue, avg(case when p.’Product line’ = 'Fashion Accessories' and c.’Customer Type’ = 'Normal' and c.gender = 'Female' then i.Total end) as Avg_Female_Non_Member_Fashion_Accessories_Revenue from Customers as c join customer_ratings as cr on c.’Customer ID’ = cr.’Customer ID’ join revenue_details as rd on cr.’Invoice ID’ = rd.’Invoice ID’ join invoices as i on rd.’Invoice ID’ = i.’Invoice ID’ join products as p on i.’Product ID’ = p.’Product ID’

Query:

select

avg(case when p.’Product line’ = 'Fashion Accessories'

and c.’Customer Type’ = 'Member' and c.gender = 'Male' then i.Total end) as Avg_Male_Member_Fashion_Accessories_Revenue,

avg(case when p.’Product line’ = 'Fashion Accessories'

and c.’Customer Type’ = 'Normal' and c.gender = 'Female' then i.Total end) as Avg_Female_Non_Member_Fashion_Accessories_Revenue

from Customers as c

join customer_ratings as cr

on c.’Customer ID’ = cr.’Customer ID’

join revenue_details as rd

on cr.’Invoice ID’ = rd.’Invoice ID’

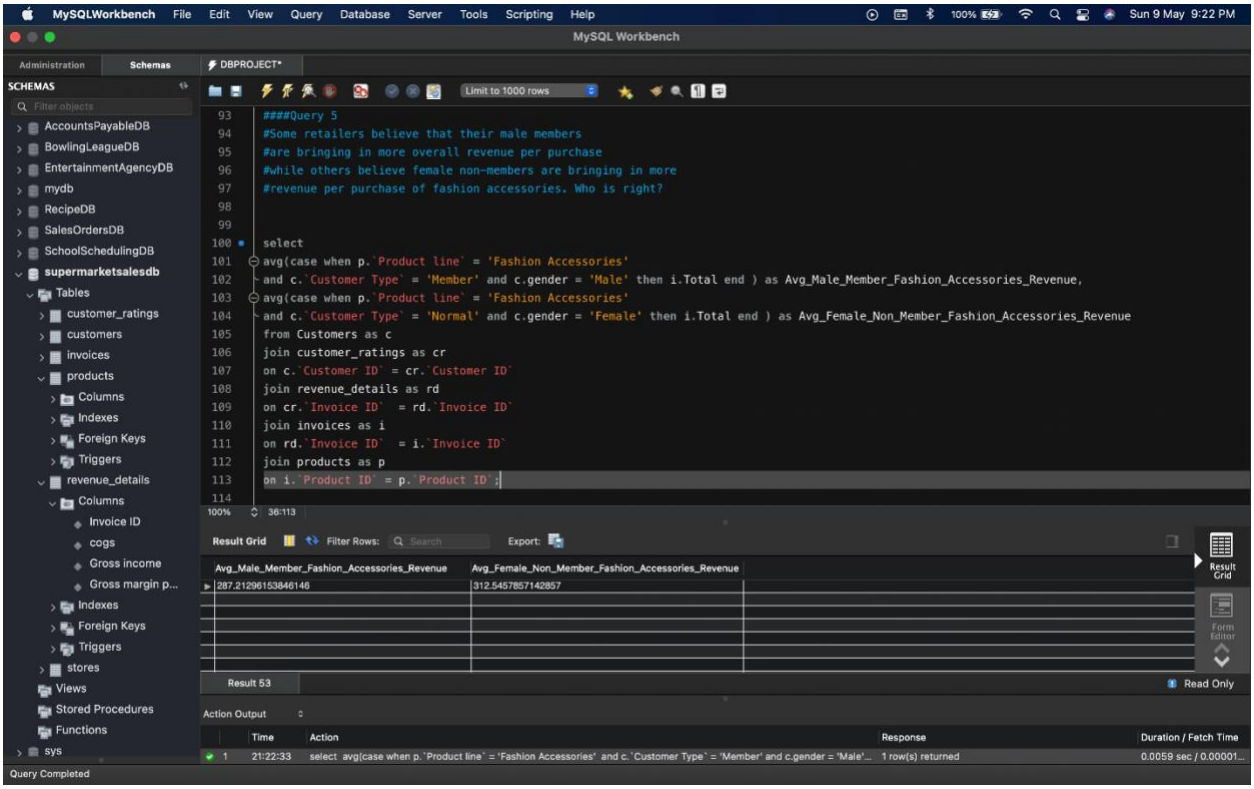
join invoices as i

on rd.’Invoice ID’ = i.’Invoice ID’

join products as p

on i.’Product ID’ = p.’Product ID’

Screen Shot of SQL Query and Results



Data Review for MongoDB

Assumptions/Notes About Data Collections, Attributes and Relationships between Collections

Supermarketsales database has 1 collection supermarketsales which has 17 attributes.

List of attributes:

Invoice ID, Branch, City, Customer type, Gender, Product Line, Unit price, Quantity, Tax 5%, Total, Date, Time, Payment, cogs, gross margin percentage, gross income, Rating

Number of documents in supermarketsales collection: 1000

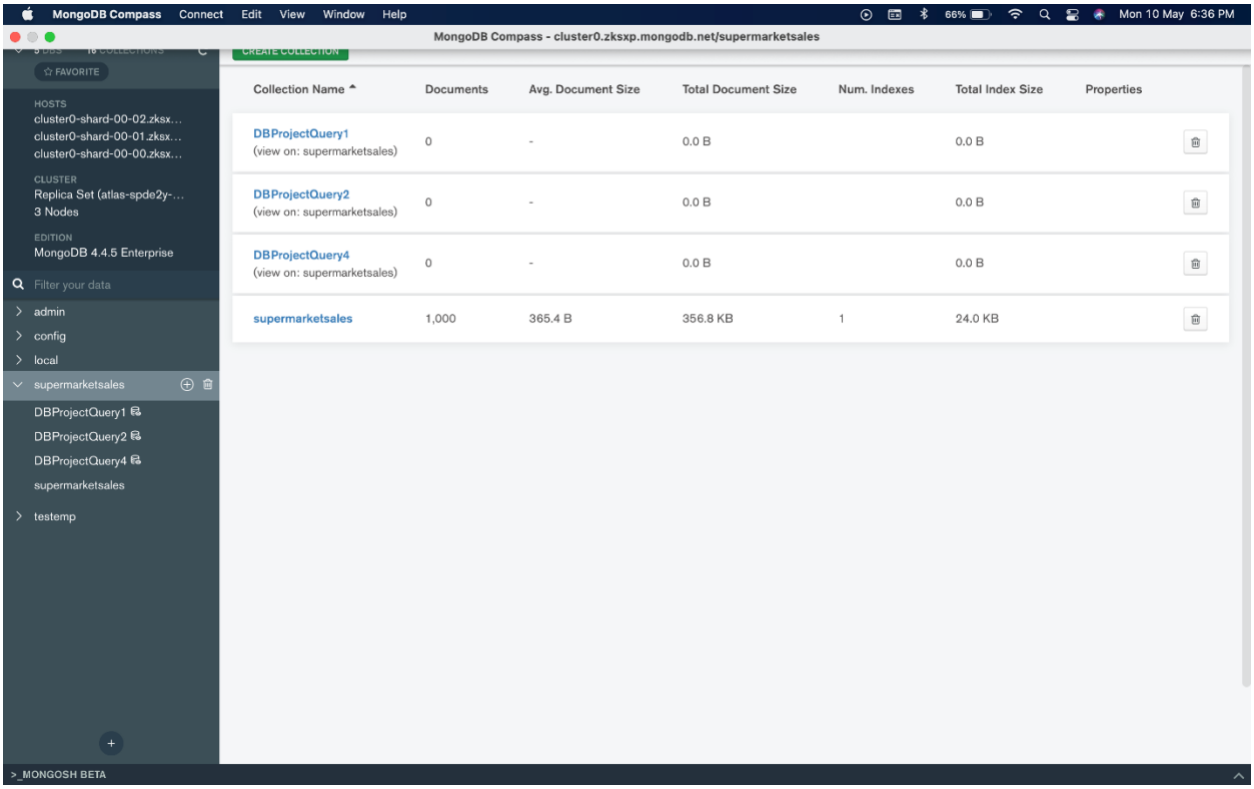
Since there is only one collection, there are no relationships.

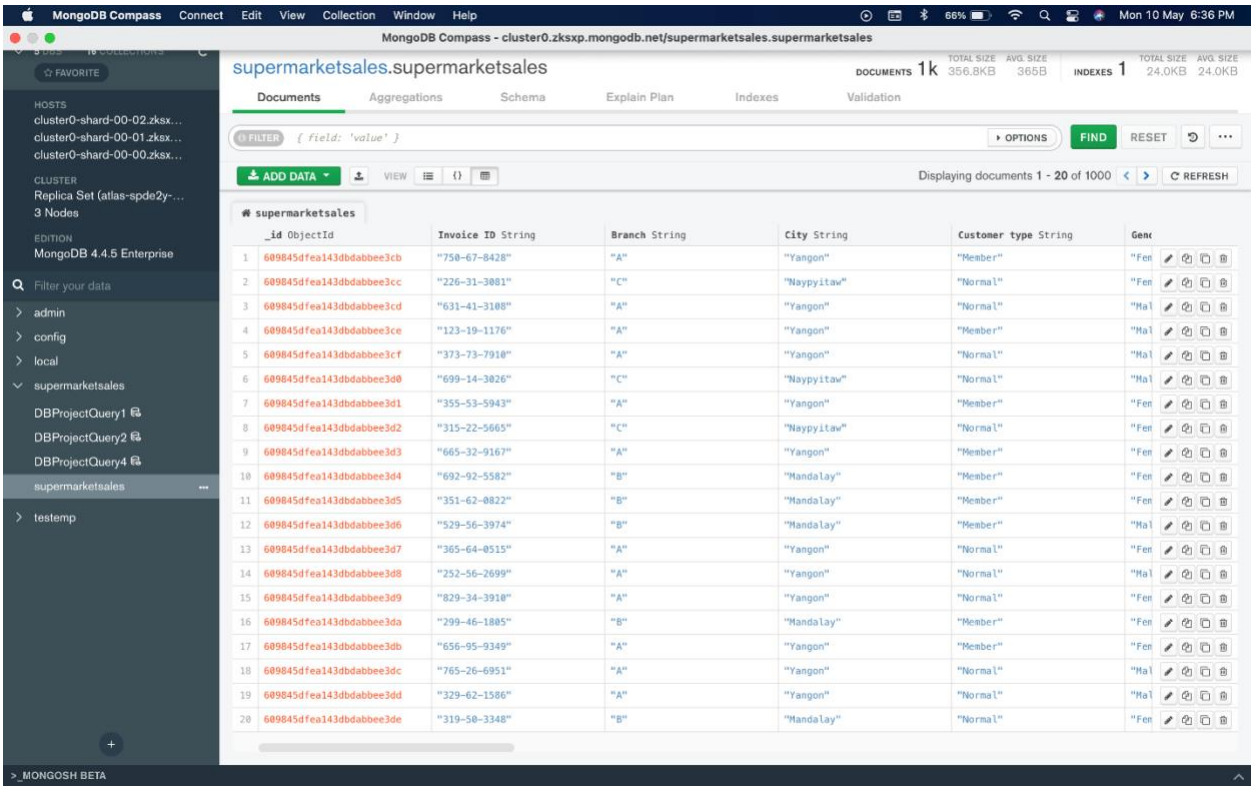
Physical Mongo Database

Assumptions/Notes About Data Set

1. Database needs to store details about each invoice being generated through transactions.
2. Database needs to store details about their branches.
3. Database needs to store every Invoice-ID and is assigned to one Unique customer.
4. Database needs to store details about the customer (Customer ID, Gender)
5. Identify the type of the customer (Normal, Member) for each order.
6. Database needs to store details about the product line and identify each type of product line the product belongs to.
7. Database needs to identify the tax amount for each invoice total
8. Database needs to store date and time for each unique transaction.
9. Company needs to identity the payment type been used for each unique transaction and store it in the database.
10. Database needs to store the cost of goods sold for per invoice.
11. Company needs to recognize the gross margin of each transaction and stored in database.
12. Database needs to store satisfaction rating for each customer.

Screen shot of Physical Database objects (Database, Collections and Attributes)





MongoDB Compass Connect Edit View Collection Window Help 100% Mon 10 May 8:16 PM

MongoDB Compass - cluster0.zksxp.mongodb.net/supermarketsales.supermarketsales

FAVORITE

HOSTS
cluster0-shard-00-02.zksx...
cluster0-shard-00-01.zksx...
cluster0-shard-00-00.zksx...

CLUSTER
Replica Set (atlas-spde2y-...
3 Nodes

EDITION
MongoDB 4.4.5 Enterprise

Filter your data
admin
config
local
supermarketsales
DBProjectQuery1
DBProjectQuery2
DBProjectQuery4
supermarketsales
testemp

supermarketsales.supermarketsales

DOCUMENTS 1k TOTAL SIZE 356.8KB AVG. SIZE 365B INDEXES 1 TOTAL SIZE 24.0KB AVG. SIZE 24.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET

ADD DATA VIEW Displaying documents 1 - 20 of 1000 REFRESH

supermarketsales

	Gender String	Product line String	Unit price Double	Quantity Int32	Tax 5% Double	Tota
1	"Female"	"Health and beauty"	74.69	7	26.1415	548.
2	"Female"	"Electronic accessories"	15.28	5	3.82	80.2
3	"Male"	"Home and lifestyle"	46.33	7	16.2155	340.
4	"Male"	"Health and beauty"	58.22	8	23.288	489.
5	"Male"	"Sports and travel"	86.31	7	30.2085	634.
6	"Male"	"Electronic accessories"	85.39	7	29.8865	627.
7	"Female"	"Electronic accessories"	68.84	6	20.652	433.
8	"Female"	"Home and lifestyle"	73.56	10	36.78	772.
9	"Female"	"Health and beauty"	36.26	2	3.626	76.1
10	"Female"	"Food and beverages"	54.84	3	8.226	172.
11	"Female"	"Fashion accessories"	14.48	4	2.896	60.8
12	"Male"	"Electronic accessories"	25.51	4	5.102	107.
13	"Female"	"Electronic accessories"	46.95	5	11.7375	246.
14	"Male"	"Food and beverages"	43.19	10	21.595	453.
15	"Female"	"Health and beauty"	71.38	10	35.69	749.
16	"Female"	"Sports and travel"	93.72	6	28.116	590.
17	"Female"	"Health and beauty"	68.93	7	24.1255	506.
18	"Male"	"Sports and travel"	72.61	6	21.783	457.
19	"Male"	"Food and beverages"	54.67	3	8.2005	172.
20	"Female"	"Home and lifestyle"	40.3	2	4.03	84.6

>_MONGOSH BETA

MongoDB Compass Connect Edit View Collection Window Help 100% Mon 10 May 8:16 PM

MongoDB Compass - cluster0.zksxp.mongodb.net/supermarketsales.supermarketsales

FAVORITE

HOSTS
cluster0-shard-00-02.zksx...
cluster0-shard-00-01.zksx...
cluster0-shard-00-00.zksx...

CLUSTER
Replica Set (atlas-spde2y-...
3 Nodes

EDITION
MongoDB 4.4.5 Enterprise

Filter your data
admin
config
local
supermarketsales
DBProjectQuery1
DBProjectQuery2
DBProjectQuery4
supermarketsales
testemp

supermarketsales.supermarketsales

DOCUMENTS 1k TOTAL SIZE 356.8KB AVG. SIZE 365B INDEXES 1 TOTAL SIZE 24.0KB AVG. SIZE 24.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

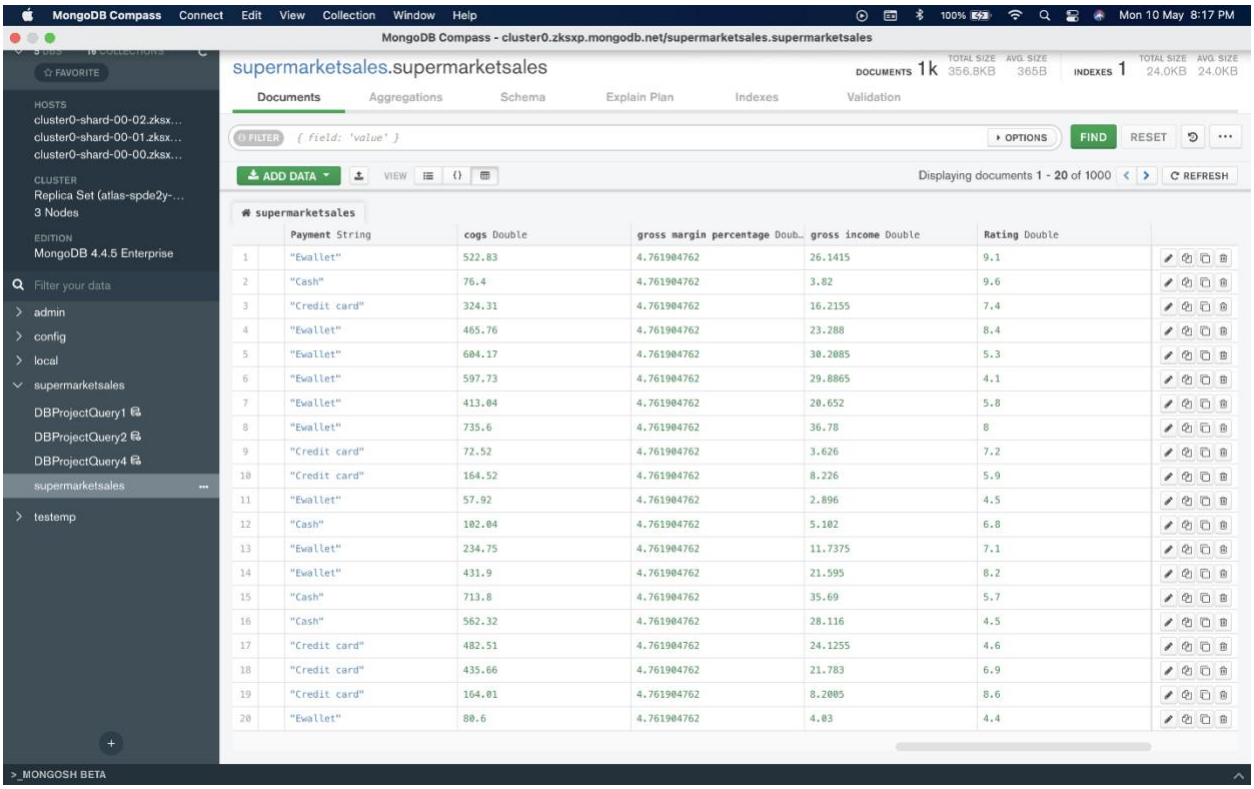
FILTER { field: 'value' } OPTIONS FIND RESET

ADD DATA VIEW Displaying documents 1 - 20 of 1000 REFRESH

supermarketsales

	Total Double	Date Date	Time String	Payment String	cogs Double	gros
1	548.9715	2019-01-05T06:00:00.000+00:0	"13:08"	"Ewallet"	522.83	4.76
2	80.22	2019-03-08T06:00:00.000+00:0	"10:29"	"Cash"	76.4	4.76
3	340.5255	2019-03-03T06:00:00.000+00:0	"13:23"	"Credit card"	324.31	4.76
4	489.048	2019-01-27T06:00:00.000+00:0	"20:33"	"Ewallet"	465.76	4.76
5	634.3785	2019-02-08T06:00:00.000+00:0	"10:37"	"Ewallet"	604.17	4.76
6	627.6165	2019-03-25T05:00:00.000+00:0	"18:30"	"Ewallet"	597.73	4.76
7	433.692	2019-02-25T06:00:00.000+00:0	"14:36"	"Ewallet"	413.04	4.76
8	772.38	2019-02-24T06:00:00.000+00:0	"11:38"	"Ewallet"	735.6	4.76
9	76.146	2019-01-10T06:00:00.000+00:0	"17:15"	"Credit card"	72.52	4.76
10	172.746	2019-02-20T06:00:00.000+00:0	"13:27"	"Credit card"	164.52	4.76
11	60.816	2019-02-06T06:00:00.000+00:0	"18:07"	"Ewallet"	57.92	4.76
12	107.142	2019-03-09T06:00:00.000+00:0	"17:03"	"Cash"	102.04	4.76
13	246.4875	2019-02-12T06:00:00.000+00:0	"10:25"	"Ewallet"	234.75	4.76
14	453.495	2019-02-07T06:00:00.000+00:0	"16:48"	"Ewallet"	431.9	4.76
15	749.49	2019-03-29T05:00:00.000+00:0	"19:21"	"Cash"	713.8	4.76
16	590.436	2019-01-15T06:00:00.000+00:0	"16:19"	"Cash"	562.32	4.76
17	506.6355	2019-03-11T05:00:00.000+00:0	"11:03"	"Credit card"	482.51	4.76
18	457.443	2019-01-01T06:00:00.000+00:0	"10:39"	"Credit card"	435.66	4.76
19	172.2105	2019-01-21T06:00:00.000+00:0	"18:00"	"Credit card"	164.01	4.76
20	84.63	2019-03-11T05:00:00.000+00:0	"15:30"	"Ewallet"	80.6	4.76

>_MONGOSH BETA



	Payment String	cogs Double	gross margin percentage Doub...	gross income Double	Rating Double
1	"Ewallet"	522.83	4.761904762	26.1415	9.1
2	"Cash"	76.4	4.761904762	3.82	9.6
3	"Credit card"	324.31	4.761904762	16.2155	7.4
4	"Ewallet"	465.76	4.761904762	23.288	8.4
5	"Ewallet"	604.17	4.761904762	30.2085	5.3
6	"Ewallet"	597.73	4.761904762	29.8865	4.1
7	"Ewallet"	413.04	4.761904762	20.652	5.8
8	"Ewallet"	735.6	4.761904762	36.78	8
9	"Credit card"	72.52	4.761904762	3.626	7.2
10	"Credit card"	164.52	4.761904762	8.226	5.9
11	"Ewallet"	57.92	4.761904762	2.896	4.5
12	"Cash"	102.04	4.761904762	5.102	6.8
13	"Ewallet"	234.75	4.761904762	11.7375	7.1
14	"Ewallet"	431.9	4.761904762	21.595	8.2
15	"Cash"	713.8	4.761904762	35.69	5.7
16	"Cash"	562.32	4.761904762	28.116	4.5
17	"Credit card"	482.51	4.761904762	24.1255	4.6
18	"Credit card"	435.66	4.761904762	21.783	6.9
19	"Credit card"	164.01	4.761904762	8.2005	8.6
20	"Ewallet"	80.6	4.761904762	4.03	4.4

Data in the Database

Collection Name	**Relationships With Other Collections (if any)**	**# of Documents in Collection**
supermarketsales	No relationship	1000

MongoDB Queries/Code

Pick 3 SQL queries and write them in MongoDB

Mongo Query 1

Question (Q.1)

Some retailers believe that there is more money to be made in selling fashion accessories to men than sports and travel to women. Is this true?

Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)

Explanation:

First, we filter the documents which have Product Line= fashion accessories, gender= Male and Product Line= sports and travel, gender= Female.

Next, we find the sum of Gross income for 2 different categories, i.e., 'fashion accessories sold to men' and 'sports and travel sold to women'. We then compare its values to see which is greater.

Assumptions:

We assume the value of 'gross income' denotes the money made on a purchase. Hence, summing the value of gross income for the 2 categories 'fashion accessories sold to men' and 'sports and travel sold to women' can be used to compare which category makes the most money.

# of Documents in Result: 2

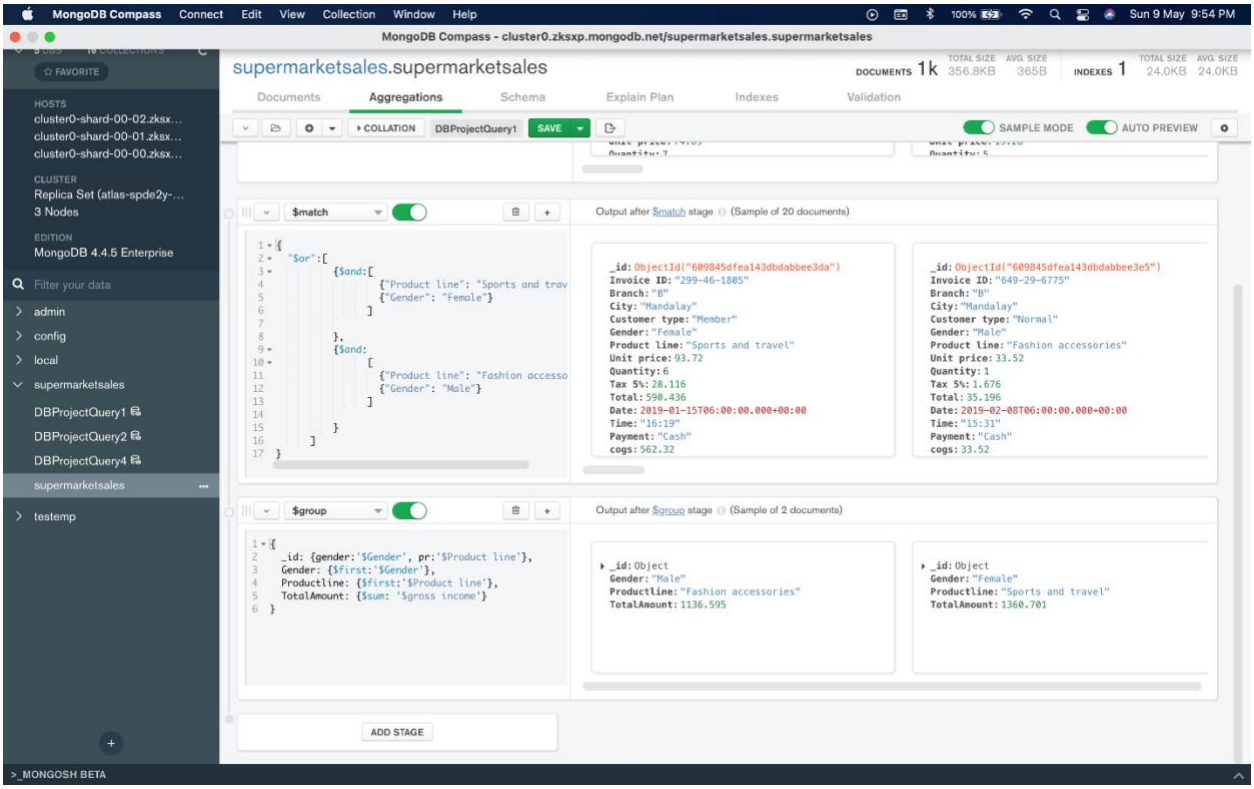
Conclusion:

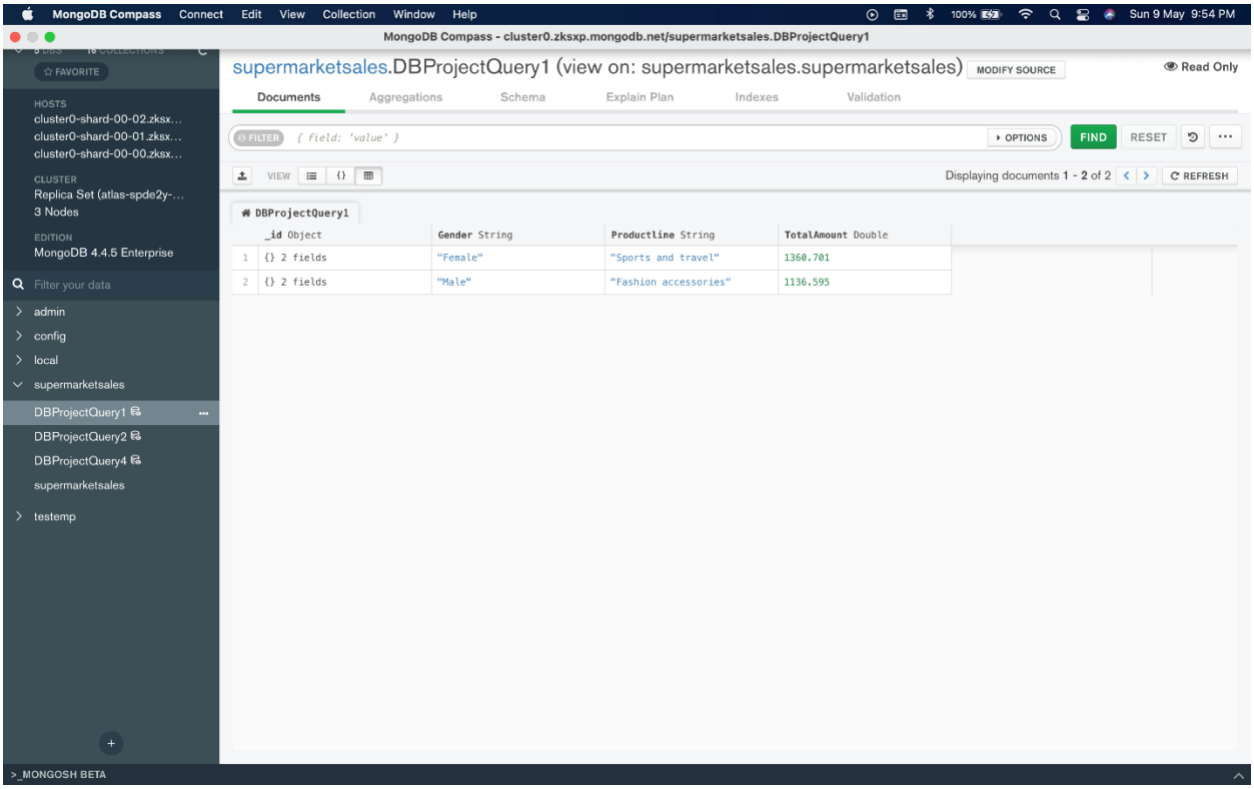
From the result of the query shown below, we can see that the sum of gross income for the category 'fashion accessories sold to men' is $1136.595 compared 'sports and travel sold to women' which is $1360.70. Hence the belief of some retailers that there is more money to be made in selling fashion accessories to men than sports and travel to women is FALSE.

Translation

Using supermarketsales collection, we use $match to filter documents for Product Line= fashion accessories, gender= Male and Product Line= sports and travel, gender= Female. $group is used in the next stage to get sum of gross income grouped by product line and gender.

Screen Shot of MongoDB Query/Code and Results





Mongo Query 2

Question (Q.2)

Some retailers believe that revenue in food and beverages can be increased amongst women by focusing on Ewallets, while others believe eWallets are more popular with men buying electronic accessories. Who is right?

Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)

Explanation:

First, we filter the documents which have Product Line= food and beverages, gender= Female and Product Line= electronic accessories, gender= Male.

Next, we find the sum of 'Total' for the 2 categories compared i.e., 'food and beverages sold to women' and 'electronic accessories sold to men' grouped by the different payment types such as Ewallet, Credit card and cash. We can use this result to compare the revenue generated for 'food and beverages sold to women' and 'electronic accessories sold to men' amongst different payment types and also see if Ewallet is a popular choice.

Assumptions:

Revenue is the total amount of income generated by the sale of goods or services related to the company's primary operations. We assume the value of 'total' denotes the revenue of a purchase. Hence, summing the value of total for the 2 categories 'food and beverages sold to women' and 'electronic accessories sold to men' can be used to compare revenue generated by different payment types.

# of Documents in Result: 6

Conclusion:

From the results of the query shown below, we can see that the sum of total for 'food and beverages sold to women' is least for payments made using Ewallets compared to credit cards and cash.

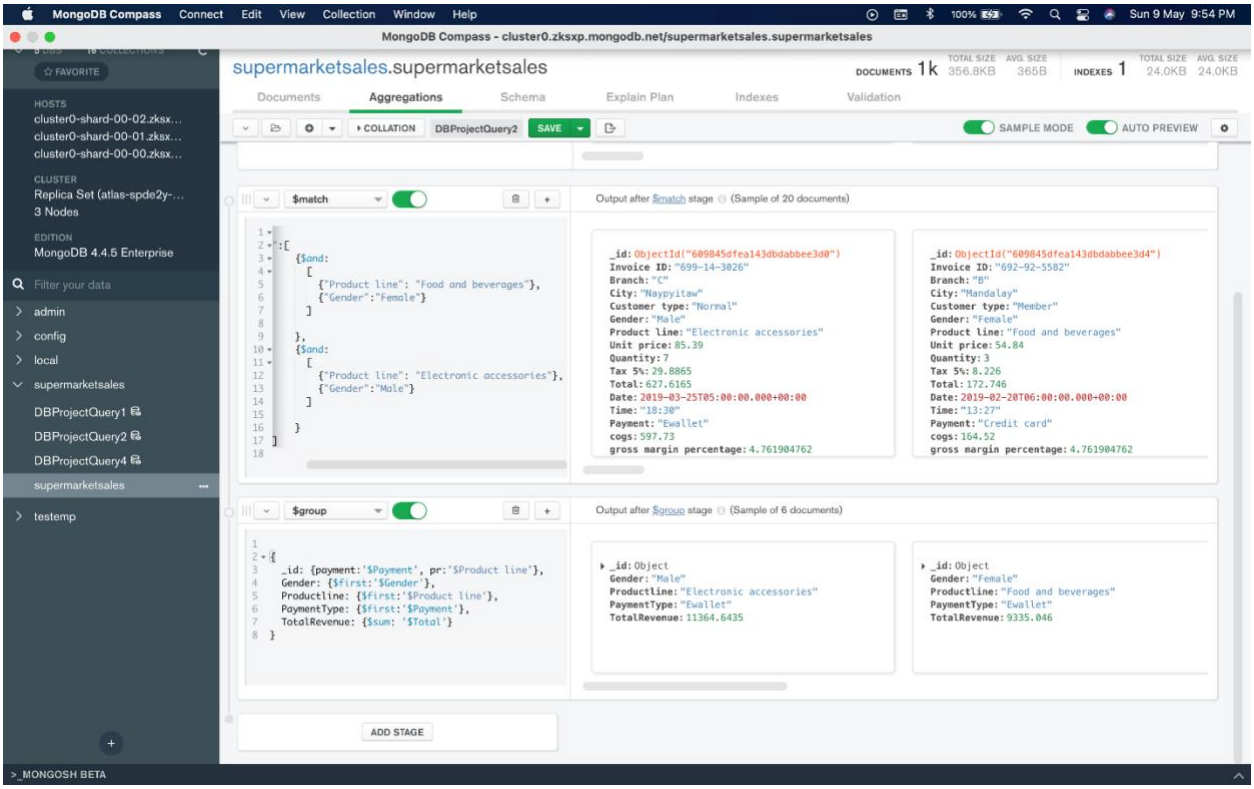
Next, we can see that sum of total for 'electronic accessories sold to men' is maximum for payments made using Ewallets.

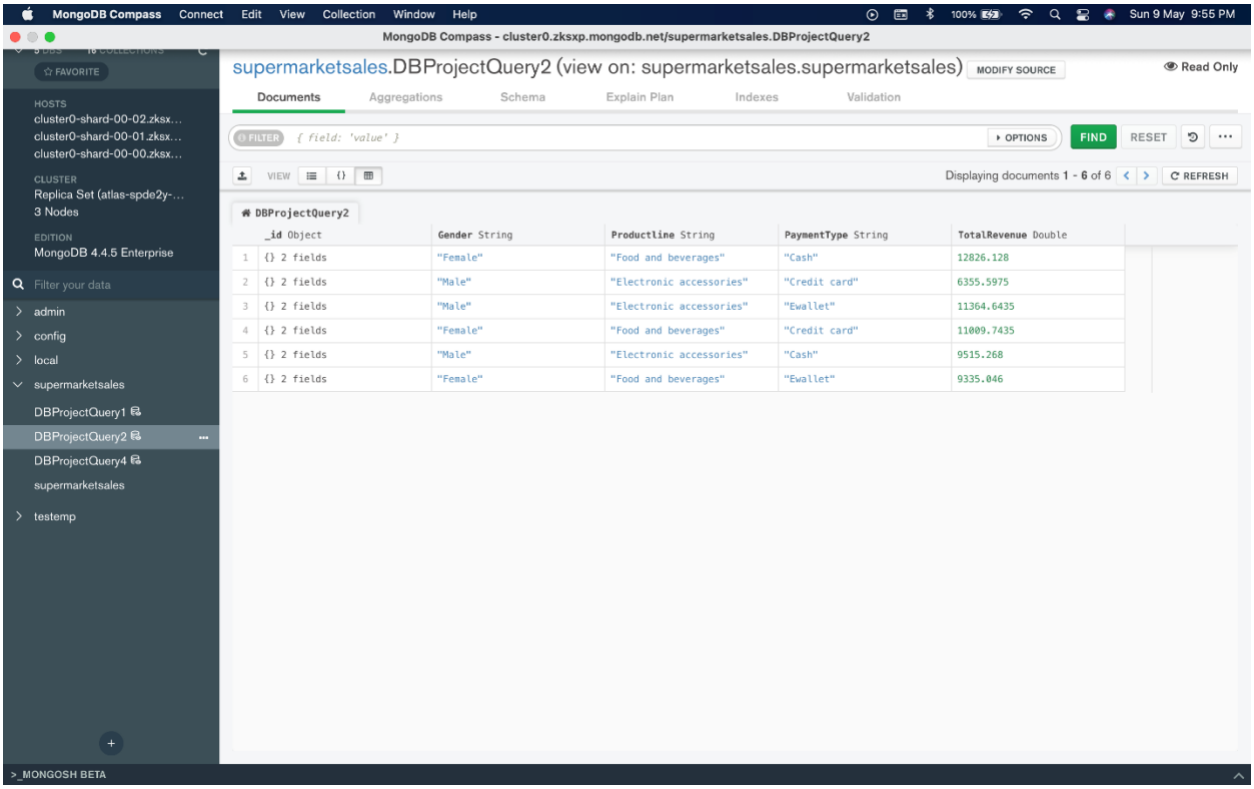
Hence, the claim made by some retailers that revenue in food and beverages can be increased amongst women by focusing on Ewallets is FALSE, while the claim that eWallets are more popular with men buying electronic accessories is TRUE.

Translation

Using supermarketsales collection, we use $match to filter documents for Product Line= food and beverages, gender= Female and Product Line= electronic accessories, gender= Male. $group is used in the next stage to get sum of total grouped by product line and payment.

Screen Shot of MongoDB Query/Code and Results





Mongo Query 3

Question (Q.4)

Some retailers believe that their members are spending more per purchase while members believe they are spending less per purchase. Who is right?

Notes/Comments About MongoDB Query/Code and Results (Include # of Documents in Result)

Explanation:

First, we find the average of total where customer type is Member. Next, we find the average of total where customer type is Normal. We then compare the averages to find if members or non-members are spending more per purchase

Assumptions:

Revenue is the total amount of income generated by the sale of goods or services related to the company's primary operations. We assume the value of 'total' denotes spending of a purchase. Hence, we find the average of total for members and non-members to compare their average spending per purchase.

# of Documents in Result: 2

Conclusion:

The result of the query shown below clearly shows that members spent on average $327.8 per purchase and non-members spent on average $318.12 per purchase. Hence, some retailers who believe that their members are spending more per purchase are right and members who believe they are spending less per purchase are wrong.

Translation

Using supermarketsales collection, we use $group to get the average of total grouped by customer type to compare the averages to find if members or non-members are spending more per purchase.

Screen Shot of MongoDB Query/Code and Results