SQL Project – Supermarket Company

BUAN6320.32

Sindhu Parthasardi Reddy
Shraddha Gururaj Kodancha

Activity	Sindhu Parthasardi Reddy	Shraddha Gururaj Kodancha
Prepared Data Model and Created Physical DB	x	x
Loaded Data into Database	x	x
Wrote SQL Queries	x	x
Prepared Report	x	x
Reviewed Report	x	x

Contents

Relational Data Model 4
Assumptions/Notes About Data Entities and Relationships 4
Entity-Relationship Diagram 5
Physical MySQL Database 6
Assumptions/Notes About Data Set 6
Screen shot of Physical Database objects 6
Data in the Database 9
SQL Queries 13
SQL Query 1 13
Question: 13
Notes/Comments About SQL Query and Results (Include # of Rows in Result) 13
Translation 13
Screen Shot of SQL Query and Results 13
SQL Query 2 15
Question 15
Notes/Comments About SQL Query and Results (Include # of Rows in Result) 15
Translation 15
Screen Shot of SQL Query and Results 15
SQL Query 4 17
Question 17
Notes/Comments About SQL Query and Results (Include # of Rows in Result) 17
Translation 17
Screen Shot of SQL Query and Results 17
SQL Query 5 19
Question 19
Notes/Comments About SQL Query and Results (Include # of Rows in Result) 19
Translation 19
Screen Shot of SQL Query and Results 19
SQL Query 7 21
Question 21
Notes/Comments About SQL Query and Results (Include # of Rows in Result) 21
Translation 21

Screen Shot of SQL Query and Results.....21

Relational Data Model

Assumptions/Notes About Data Entities and Relationships

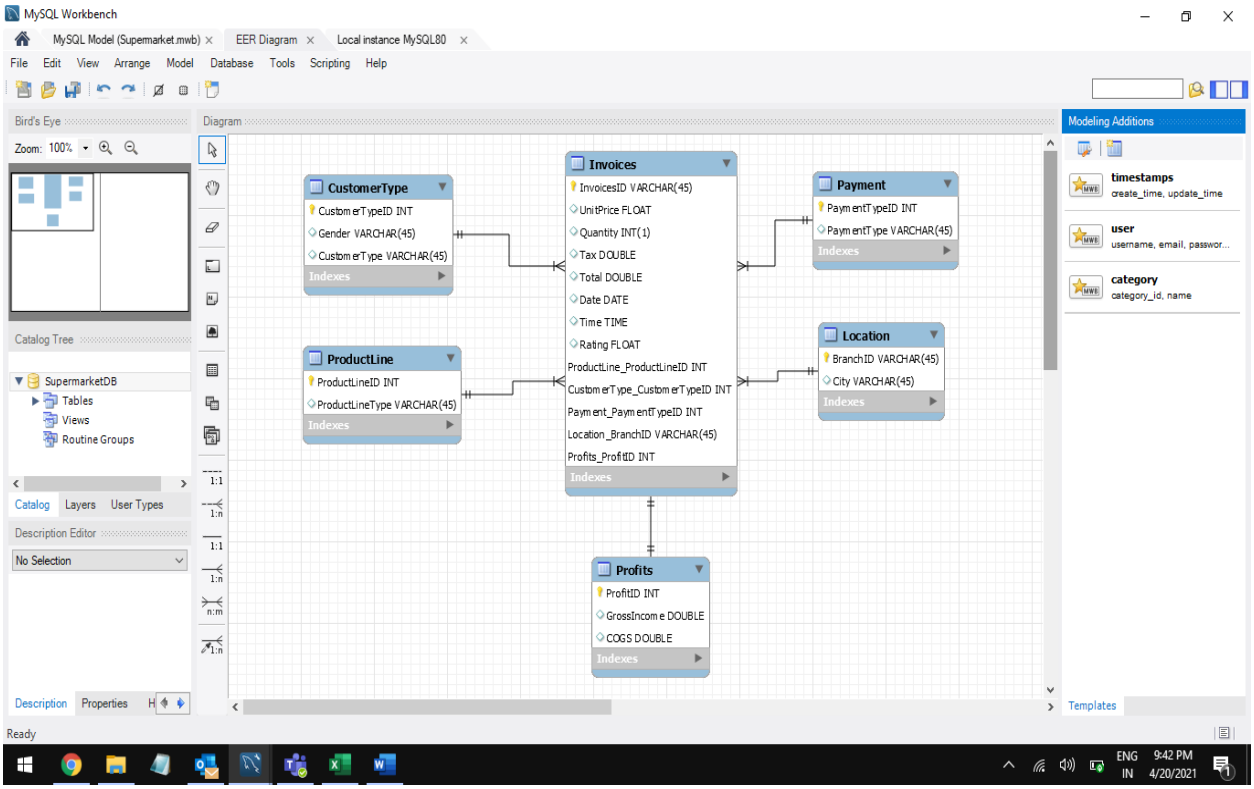
Project assumption:

1. One invoices ID can be associated with only one payment type ID and one payment type ID can be associated with multiple invoice ID. Hence, it is a one-to-many relationship(1:N)
2. One invoices ID can be associated with only one Branch ID and one Branch ID can be associated with multiple invoices ID. Hence, it is a one-to-many relationship(1:N)
3. One invoices ID can be associated with only one product line ID and one product line ID can be associated with multiple invoices ID. Hence, it is a one-to-many relationship(1:N)
4. One invoices ID can be associated with only one customer type ID and one customer type ID can be associated with multiple invoices ID. Hence, it is a one-to-many relationship(1:N)
5. Gross margin percentage column is not included in the data model as it is a constant.

Reasons why the data model is in 3NF:

1. All the rows in the tables have primary keys with unique and non-null values.
2. Every attribute in every table has atomic values (no multi value, no multi part values).
3. Every attribute that is not a part of primary key is functionally dependent on complete primary key.
4. Every non-key column is independent of each other and is only functionally dependent on the entire primary key and not on each other.

Entity-Relationship Diagram

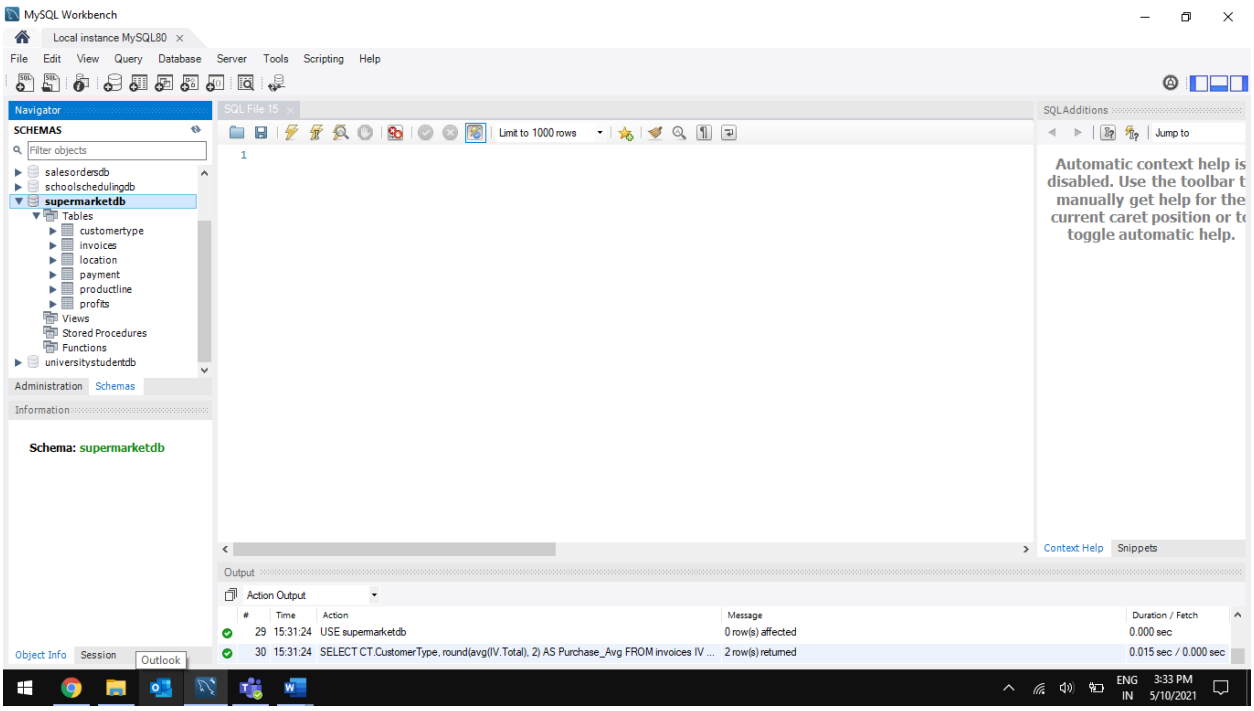


Physical MySQL Database

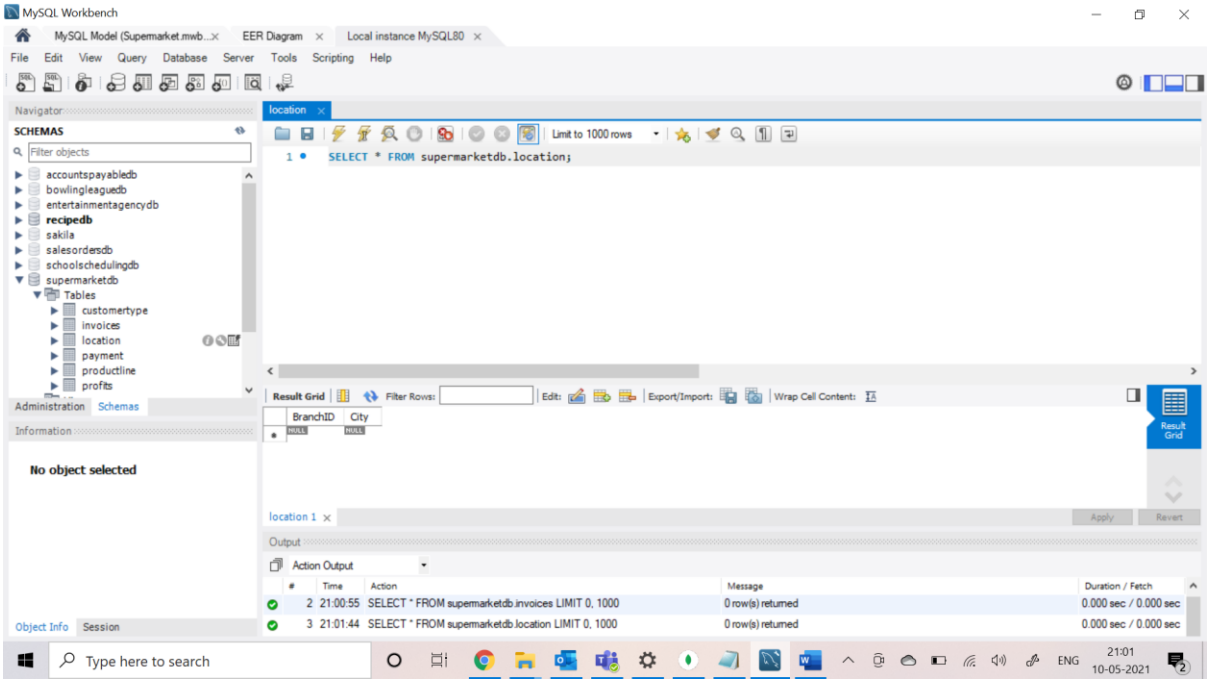
Assumptions/Notes About Data Set

1. The customer Type table has four combinations of customer type with their primary keys defined as below:
    - Member, female = 1001
    - Normal, female = 1002
    - Member, male = 1003
    - Normal, male = 1004
2. The payment Type table has four combinations of payment type with their primary keys defined as below:
    - Ewallet = 101
    - Cash = 102
    - Credit card = 103
3. The product line Type table has four combinations of product line type with their primary keys defined as below:
    - Health and beauty = 1
    - Electronic accessories = 2
    - Home and lifestyle = 3
    - Sports and travel = 4
    - Food and beverages = 5
    - Fashion accessories = 6

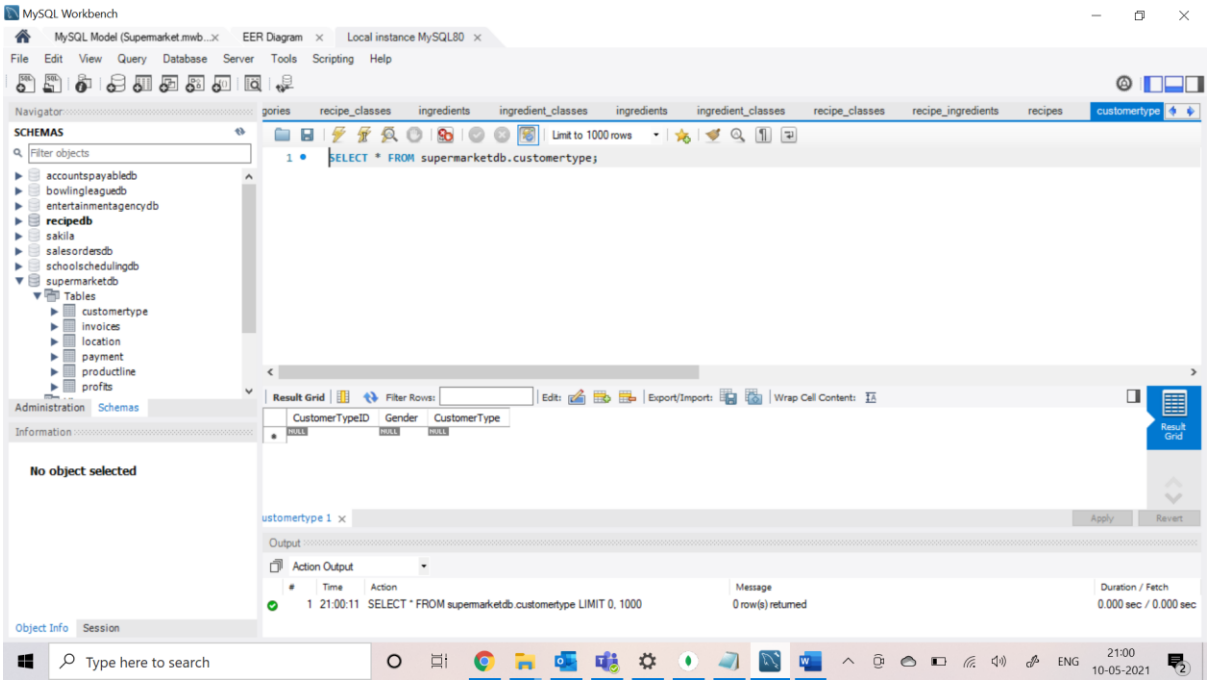
Screen shot of Physical Database objects



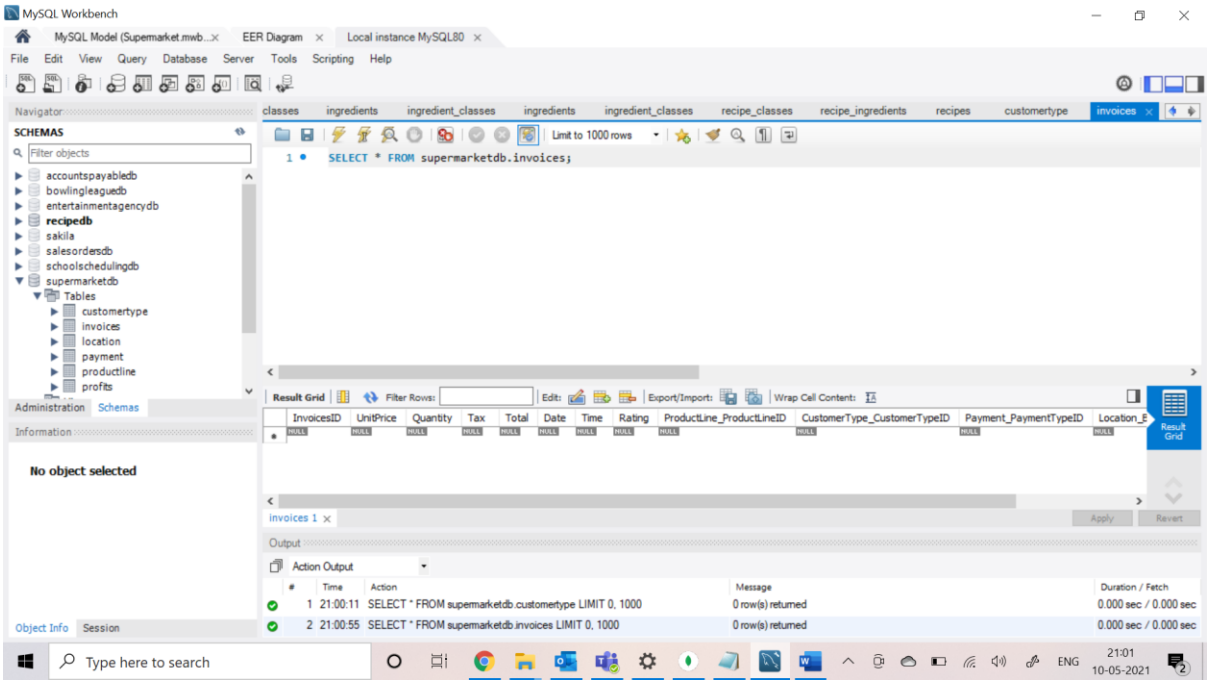
Location Table



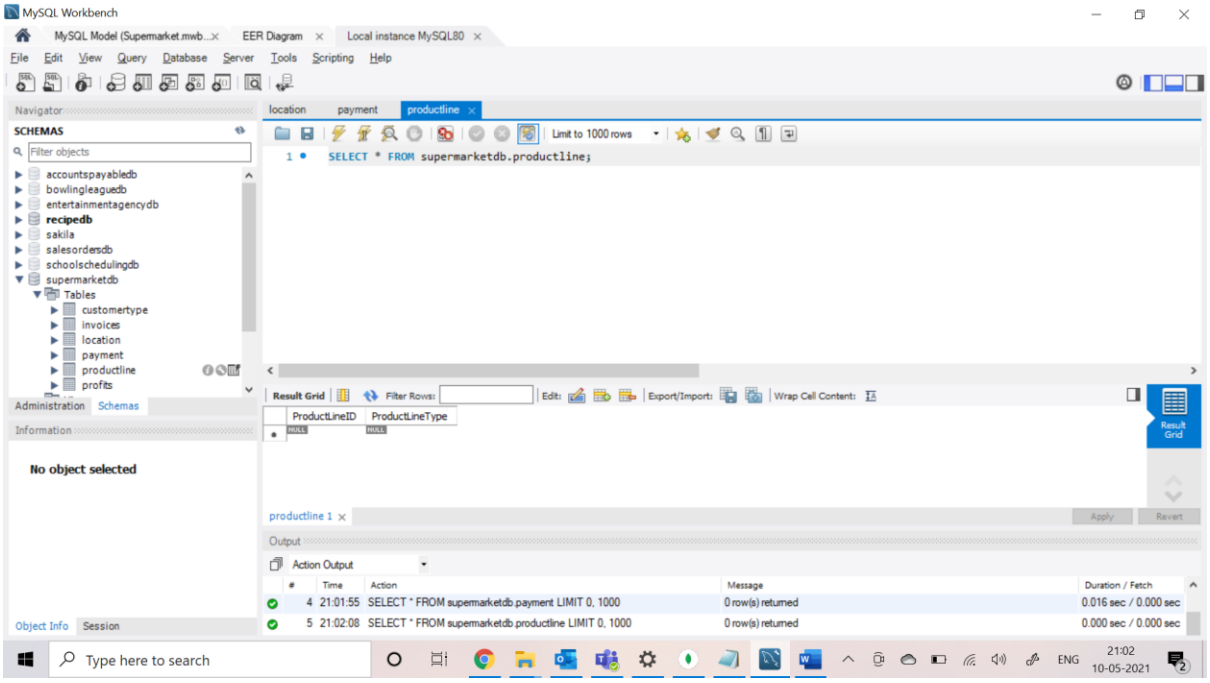
Customer Type Table



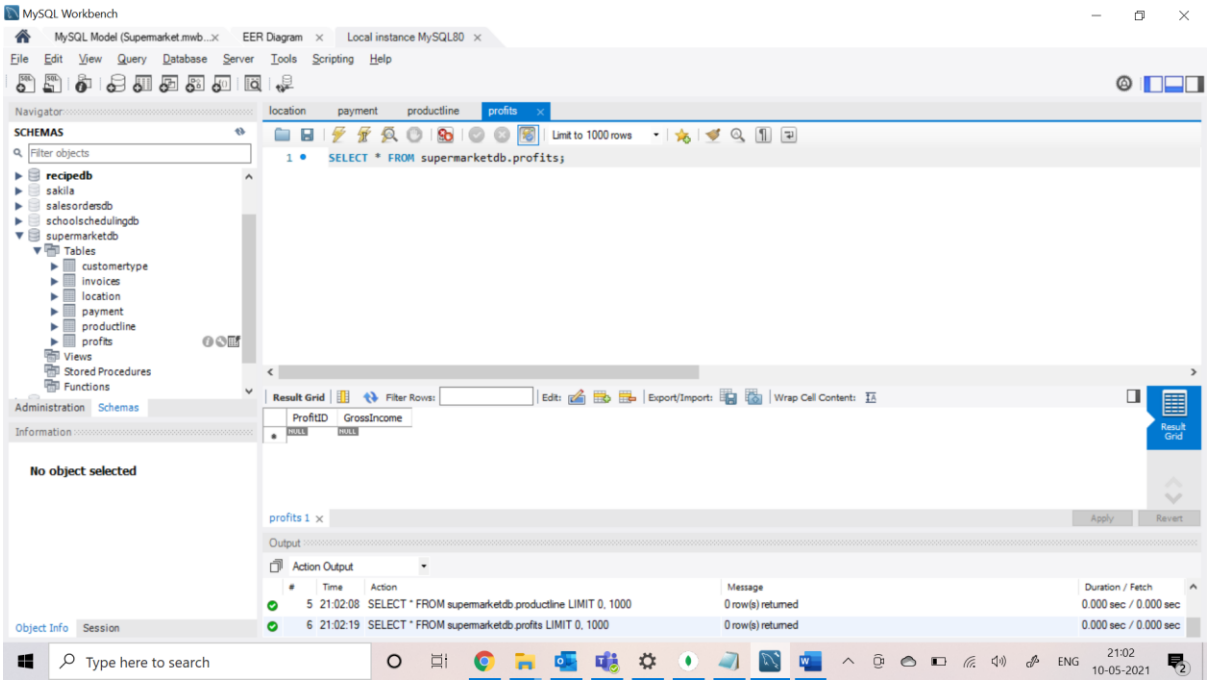
Invoices Table



Product Line Details Table



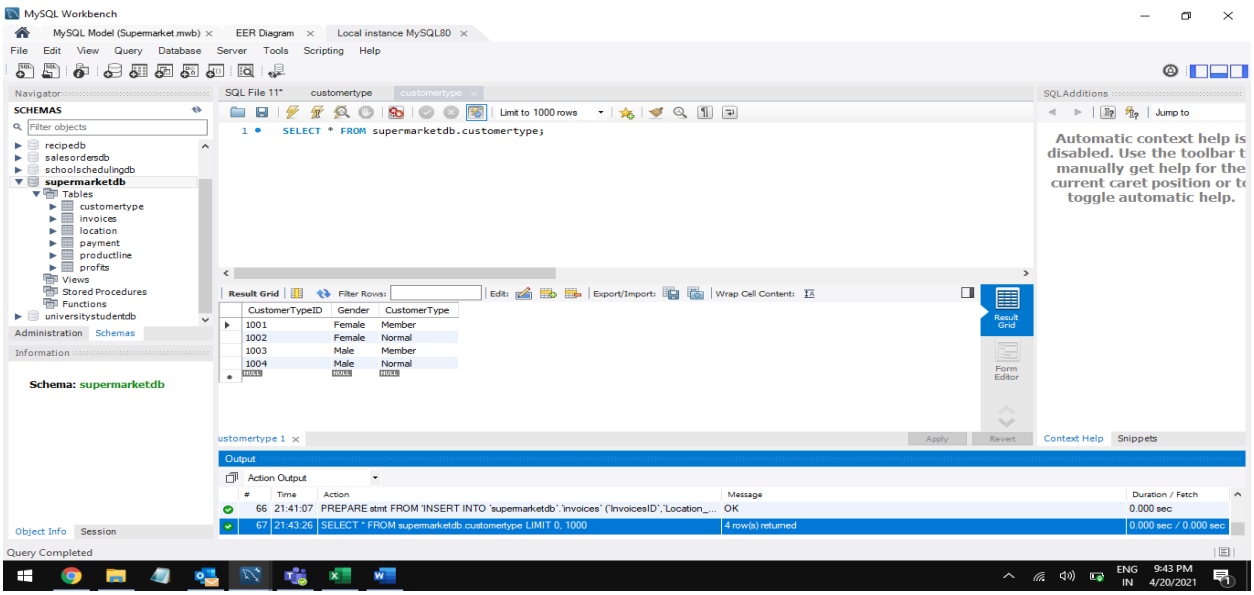
Profits Table



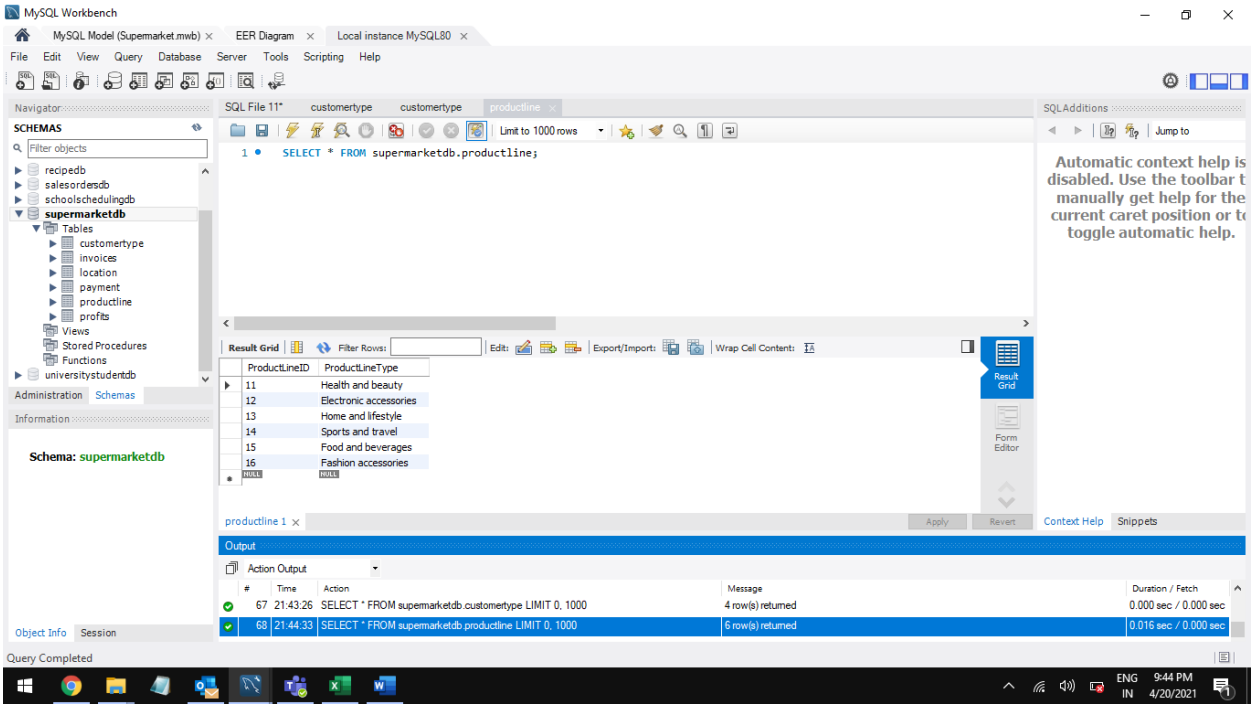
Data in the Database

Table Name	Primary Key	Foreign Key	# of Rows in Table
CustomerType	CustomerTypeID	-	4
ProductLine	ProductLineID	-	6
Invoices	InvoicesID	ProductLine_ProductLineID CustomerType_CustomerTypeID Payment_PaymentTypeID Location_BranchID Profits_ProfitID	1000
Payment	PaymentTypeID	-	3
Profits	ProfitID	-	1000
Location	BranchID	-	3

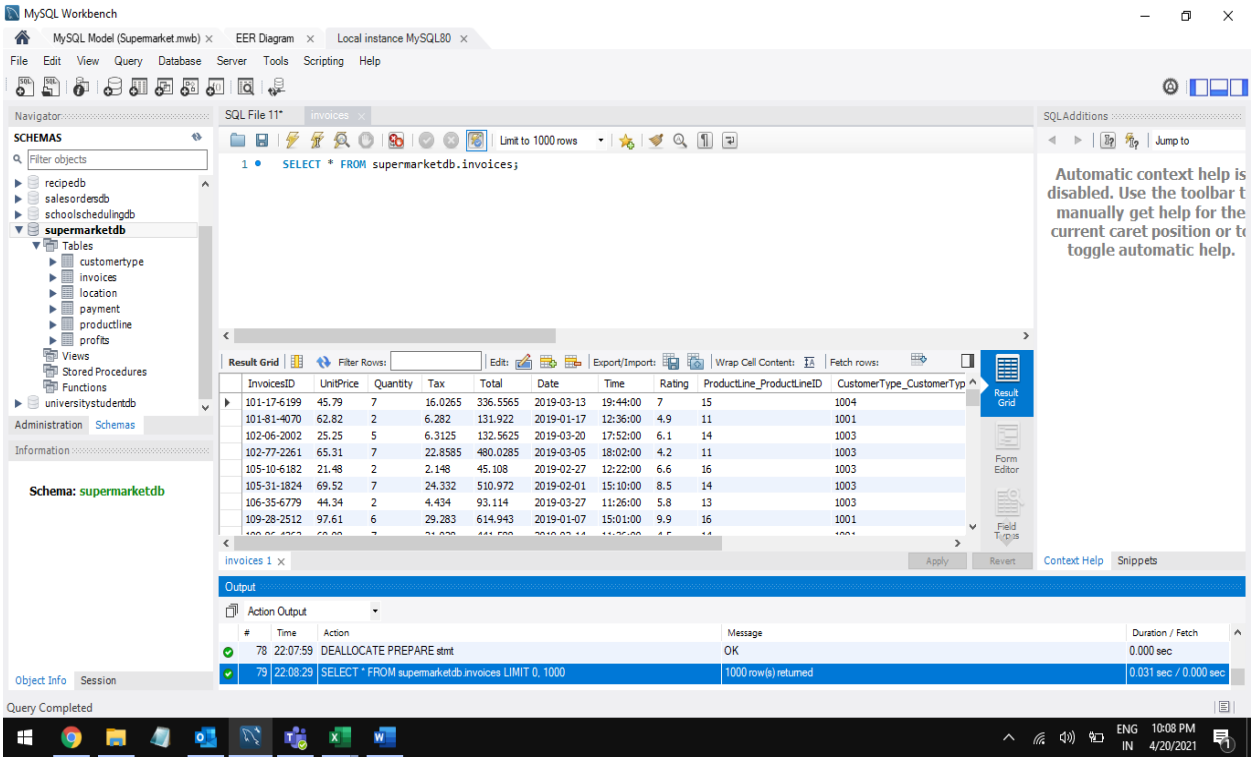
CustomerType table loaded in the database:



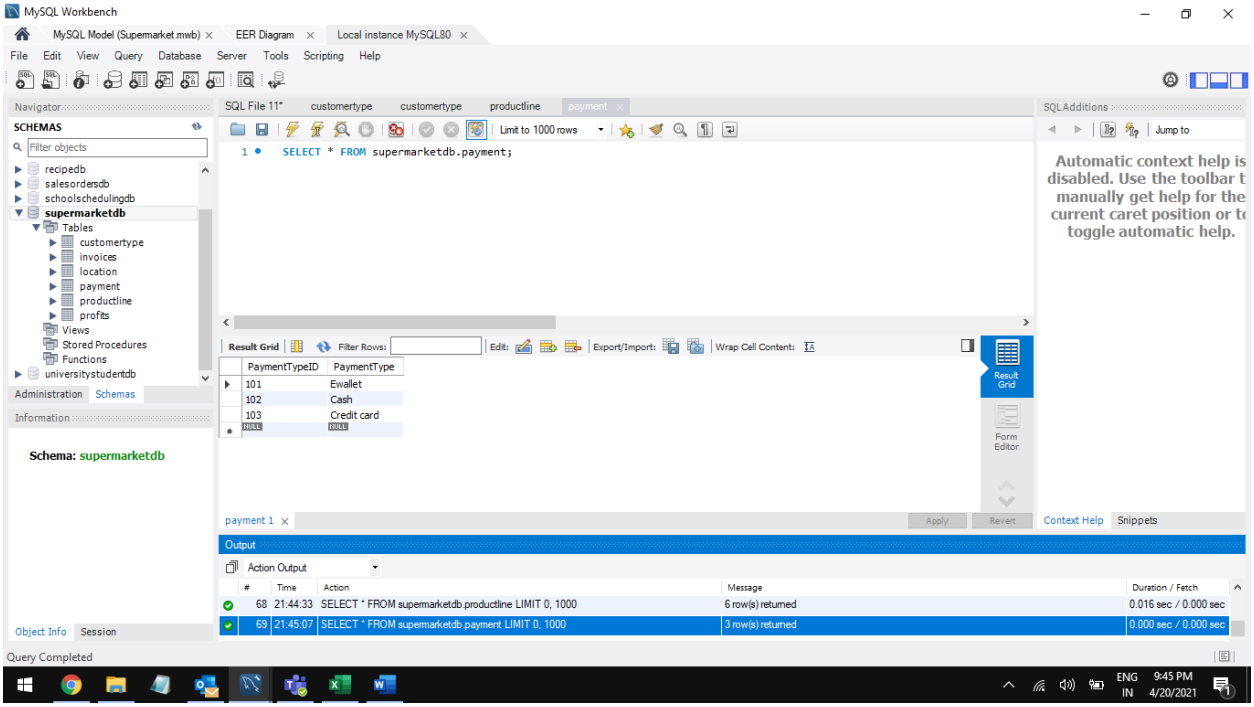
ProductLine table loaded in the database:



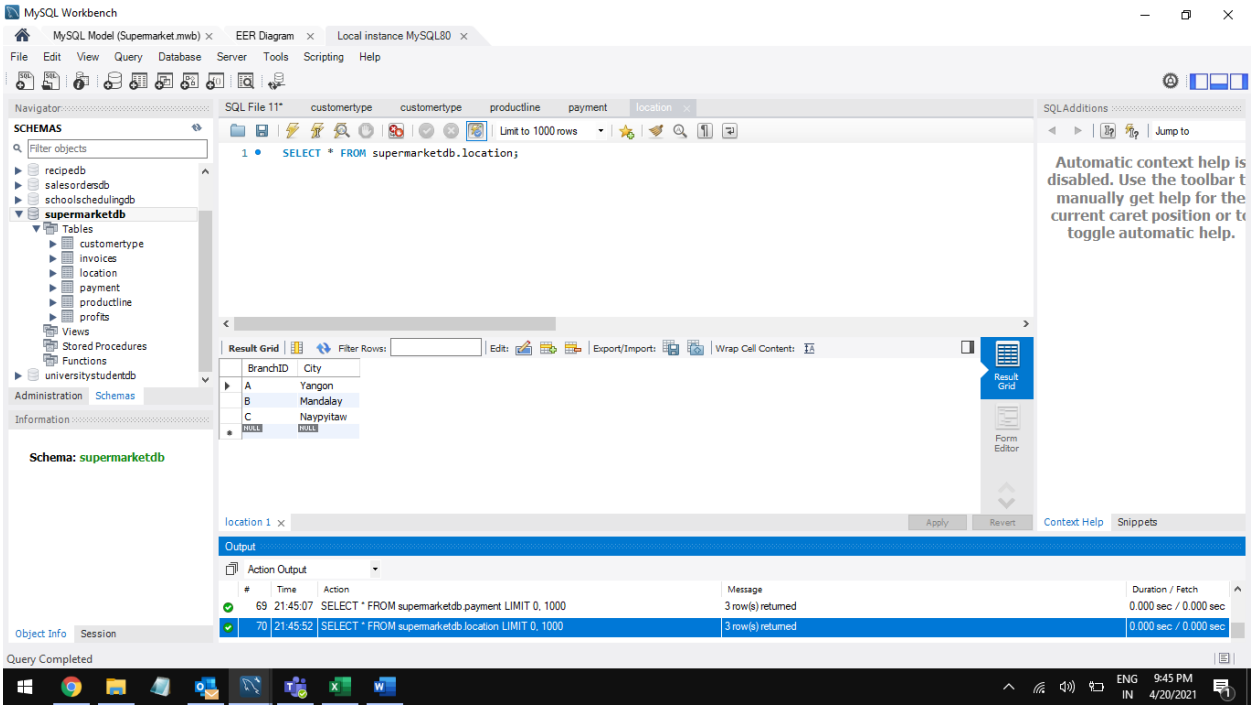
Invoices table loaded in the database:



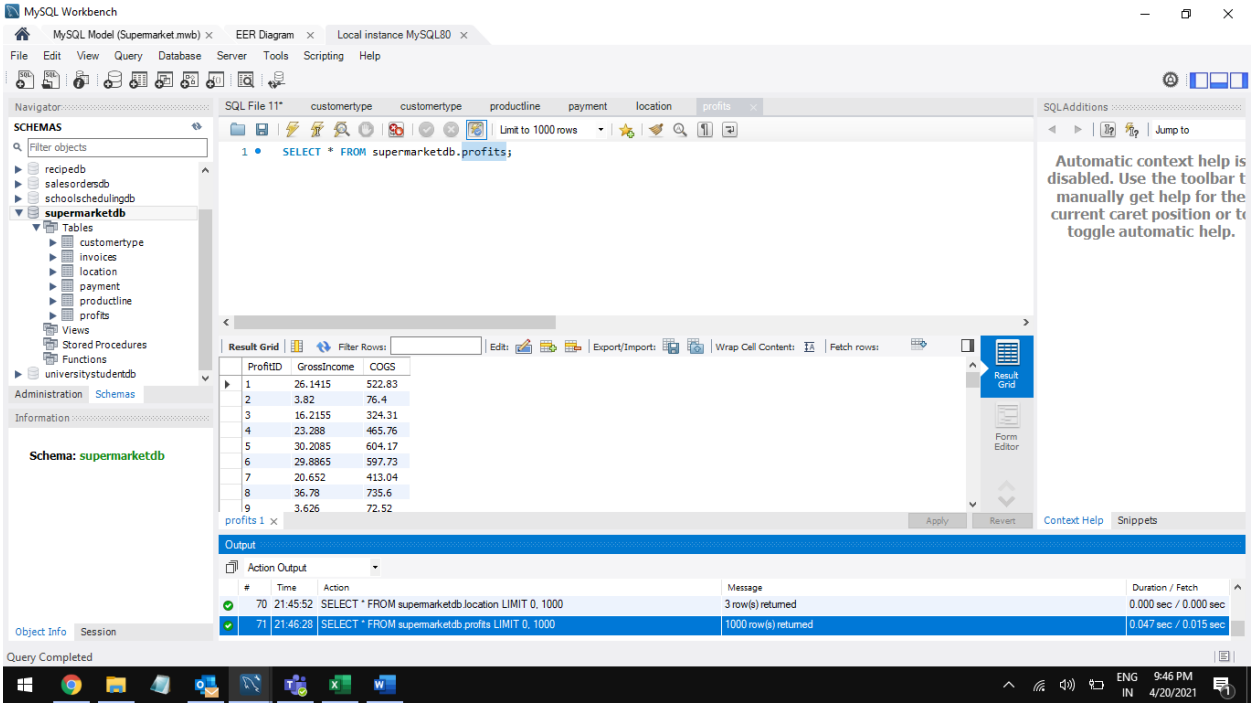
Payment table loaded in the database:



Location table loaded in the database:



Profits table loaded in the database:



SQL Queries

SQL Query 1

Question:

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 SQL Query and Results (Include # of Rows in Result)

2. As per the requirement we need three tables to be joined, that is, the invoice table, product line and customer type table.

3. The SQL query returns two rows consisting of total that is made by “male” in “fashion and accessories” and “female” in “sports and travel”

4. From the output, it is clearly seen that total for male in fashion and accessories is 23868.49 and the total for female in sports and travel is 28574.72.

5. Therefore, we conclude that whatever the retailers believe is true and that there is more money to be made from male in fashion and accessories than female in sports and travel.

Translation

Translation: Select the sum of total, gender and production line type from invoices table joining the production line table joining customer type table where gender is Male and product line type is Fashion accessories union with Select the purchase total, gender and production line type from invoices table joining the production line table joining customer type table where gender is female and product line type is Sports and travel

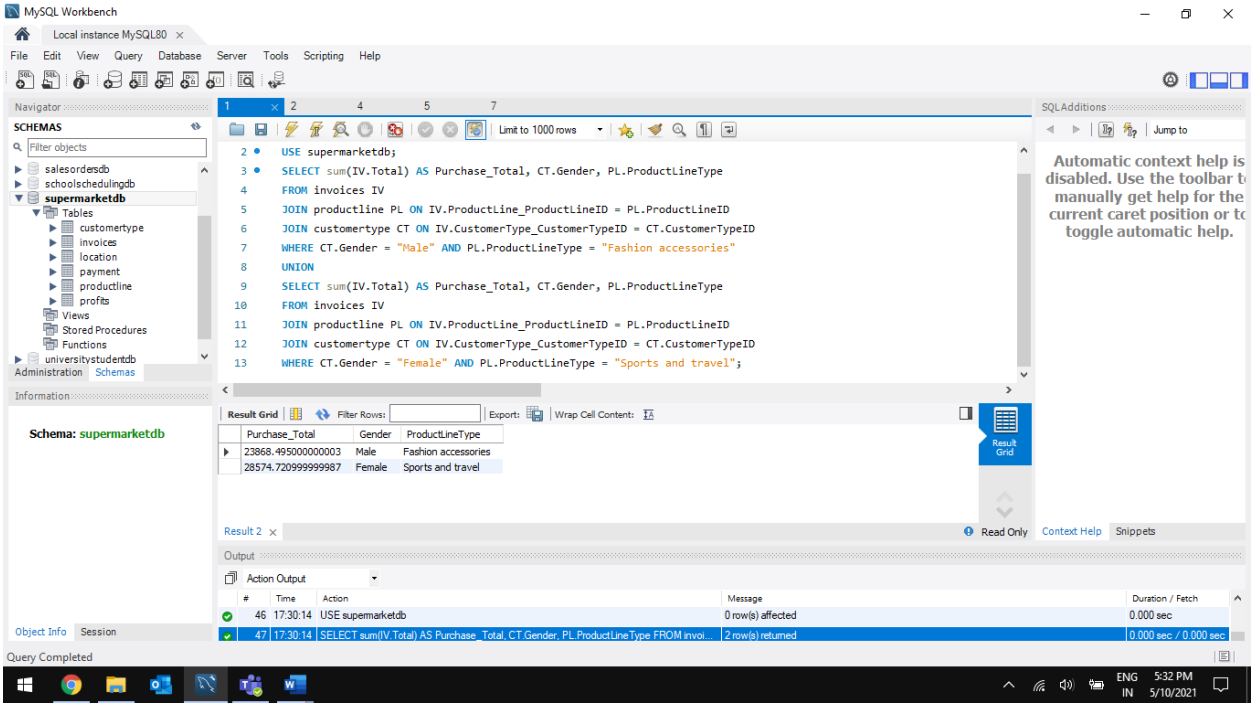
Clean up: Select sum of Total as Purchase_Total, Gender and ProductLineType from invoices join productline on ProductLine_ProductLineID = ProductLineID join customertype on CustomerType_CustomerTypeID = CustomerTypeID where Gender = Male and ProductLineType = Fashion accessories union Select sum of Total as Purchase_Total, Gender and ProductLineType from invoices join productline on ProductLine_ProductLineID = ProductLineID join customertype on CustomerType_CustomerTypeID = CustomerTypeID where Gender = Female and ProductLineType = Sports and travel

Screen Shot of SQL Query and Results

SQL Query:

```
USE supermarketdb;

SELECT sum(IV.Total) AS Purchase_Total, CT.Gender, PL.ProductLineType

FROM invoices IV

JOIN productline PL ON IV.ProductLine_ProductLineID = PL.ProductLineID

JOIN customertype CT ON IV.CustomerType_CustomerTypeID = CT.CustomerTypeID

WHERE CT.Gender = "Male" AND PL.ProductLineType = "Fashion accessories"

UNION
```

```
SELECT sum(IV.Total) AS Purchase_Total, CT.Gender, PL.ProductLineType

FROM invoices IV

JOIN productline PL ON IV.ProductLine_ProductLineID = PL.ProductLineID

JOIN customertype CT ON IV.CustomerType_CustomerTypeID = CT.CustomerTypeID

WHERE CT.Gender = "Female" AND PL.ProductLineType = "Sports and travel";
```



SQL Query 2

Question

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)

1. As per the requirement we need four tables to be joined, that is, the invoice table, product line type, payment and customer type table.
2. The SQL query returns two rows consisting of total that is made by “female” on “food and beverages” using “Ewallet” payment type and “male” in “Electronic accessories” using “Ewallet” payment type.
3. From the output, it is clearly seen that total for male in electronic accessories using ewallet is 11364.6435 and the total for female in food and beverages using ewallet is 9335.0549.
4. Therefore, we conclude that Ewallets are more popular with men buying electronic accessories than female buying food and beverages.

Translation

Translation: Select the sum of total, gender, payment type and production line type from invoices table joining the production line table joining customer type table joining payment table where gender is Male and product line type is Electronic accessories and payment type is EWallet union with Select the purchase total, gender, payment type and production line type from invoices table joining the production line table joining customer type table joining the payments table where gender is female and product line type is Food and beverages and payment type is EWallet

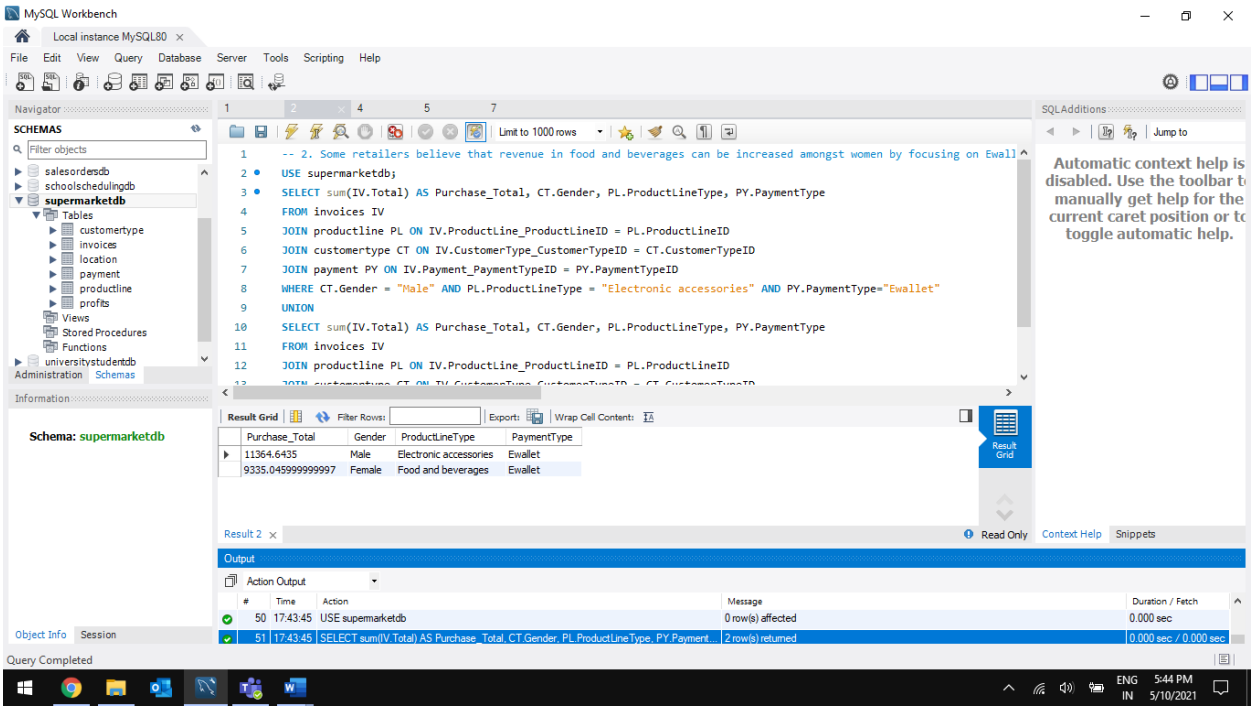
Clean up: Select sum(IV.Total) as Purchase_Total, Gender, ProductLineType and PaymentType from invoices join productline on ProductLine_ProductLineID = ProductLineID customertype on CustomerType_CustomerTypeID = CustomerTypeID join payment on payment_PaymentTypeID = PaymentTypeID where Gender = Male AND ProductLineType = Electronic accessories AND PaymentType = Ewallet union select sum(IV.Total) as Purchase_Total, Gender, ProductLineType, PaymentType from invoices join productline on ProductLine_ProductLineID = ProductLineID join customertype on CustomerType_CustomerTypeID = CustomerTypeID join payment on Payment_PaymentTypeID = PaymentTypeID where Gender = Female and ProductLineType = Food and beverages and PaymentType = Ewallet

Screen Shot of SQL Query and Results

SQL Query:

```
USE supermarketdb;

SELECT sum(IV.Total) AS Purchase_Total, CT.Gender, PL.ProductLineType, PY.PaymentType

FROM invoices IV

JOIN productline PL ON IV.ProductLine_ProductLineID = PL.ProductLineID
```

```
JOIN customertype CT ON IV.CustomerType_CustomerTypeID = CT.CustomerTypeID
JOIN payment PY ON IV.Payment_PaymentTypeID = PY.PaymentTypeID
WHERE CT.Gender = "Male" AND PL.ProductLineType = "Electronic accessories" AND PY.PaymentType="Ewallet"
UNION
SELECT sum(IV.Total) AS Purchase_Total, CT.Gender, PL.ProductLineType, PY.PaymentType
FROM invoices IV
JOIN productline PL ON IV.ProductLine_ProductLineID = PL.ProductLineID
JOIN customertype CT ON IV.CustomerType_CustomerTypeID = CT.CustomerTypeID
JOIN payment PY ON IV.Payment_PaymentTypeID = PY.PaymentTypeID
WHERE CT.Gender = "Female" AND PL.ProductLineType = "Food and beverages" AND PY.PaymentType = "Ewallet";
```



SQL Query 4

Question

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)

1. As per the requirement we need two tables to be joined, that is, the invoice table, customer type table.
2. The SQL query returns two rows consisting of Member and Normal customer type with their respective purchase average.
3. From the output, it is clearly seen that purchase average for member customer type is 327.79 and purchase average for normal customer type is 318.12
4. Therefore, we conclude that retailers who believe that members are spending more per purchase are right in making the statement.

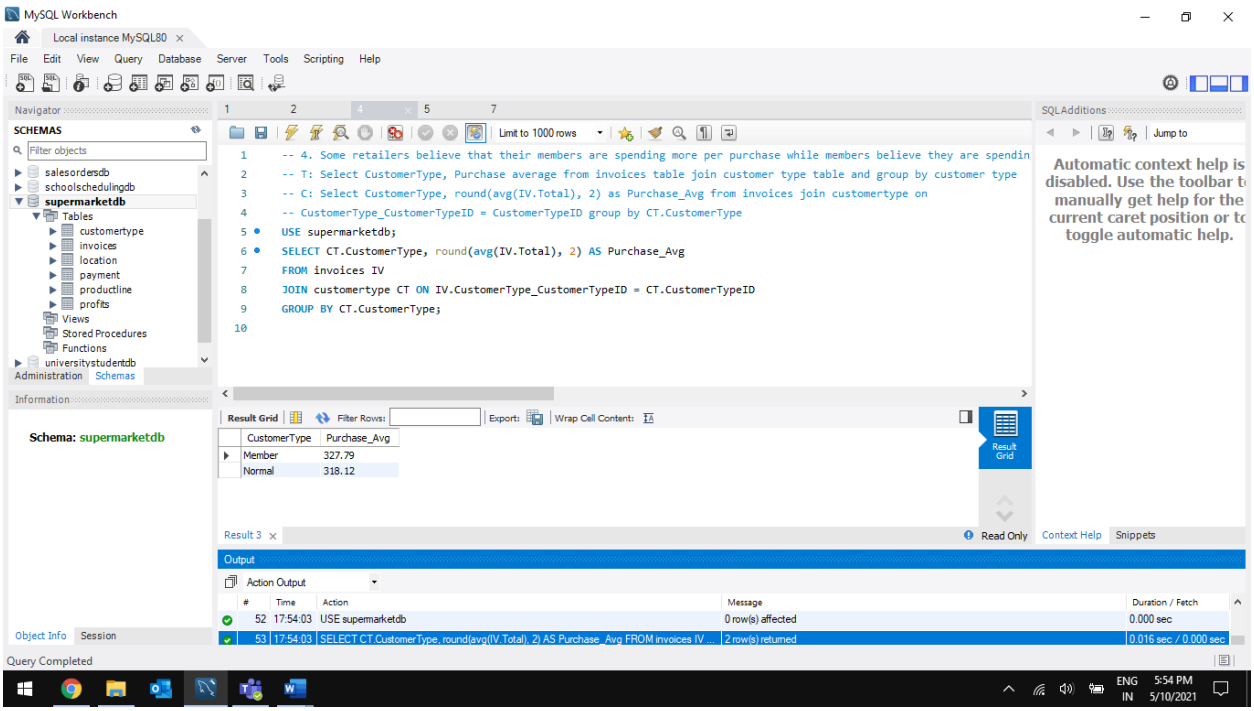
Translation

Translation: Select CustomerType, Purchase average from invoices table join customer type table and group by customer type

Cleanup: Select CustomerType, round(avg(IV.Total), 2) as Purchase_Avg from invoices join customertype on CustomerType_CustomerTypeID = CustomerTypeID group by CT.CustomerType

Screen Shot of SQL Query and Results

```
USE supermarketdb;
SELECT CT.CustomerType, round(avg(IV.Total), 2) AS Purchase_Avg
FROM invoices IV
JOIN customertype CT ON IV.CustomerType_CustomerTypeID = CT.CustomerTypeID
GROUP BY CT.CustomerType;
```


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
salesordersdb
schoolschedulingdb
supermarketdb
Tables
customertype
invoices
location
payment
productline
profits
Views
Stored Procedures
Functions
universitystudentdb
Administration Schemas
Information
Schema: supermarketdb
Object Info Session
Limit to 1000 rows
1 -- 4. Some retailers believe that their members are spending more per purchase while members believe they are spendin
2 -- T: Select CustomerType, Purchase average from invoices table join customer type table and group by customer type
3 -- C: Select CustomerType, round(avg(IV.Total), 2) as Purchase_Avg from invoices join customertype on
4 -- CustomerType_CustomerTypeID = CustomerTypeID group by CT.CustomerType
5 USE supermarketdb;
6 SELECT CT.CustomerType, round(avg(IV.Total), 2) AS Purchase_Avg
7 FROM invoices IV
8 JOIN customertype CT ON IV.CustomerType_CustomerTypeID = CT.CustomerTypeID
9 GROUP BY CT.CustomerType;
10
Result Grid Filter Rows: Export: Wrap Cell Content:
CustomerType Purchase_Avg
Member 327.79
Normal 318.12
Result 3
Read Only
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or to toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
52 17:54:03 USE supermarketdb 0 row(s) affected 0.000 sec
53 17:54:03 SELECT CT.CustomerType, round(avg(IV.Total), 2) AS Purchase_Avg FROM invoices IV ... 2 row(s) returned 0.016 sec / 0.000 sec
Query Completed
ENG IN 5:54 PM 5/10/2021

SQL Query 5

Question

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)

1. As per the requirement we need three tables to be joined, that is, the invoice table, customer type and product line table.
2. The SQL query returns two rows consisting of Member male on fashion accessories and Normal female on fashion accessories with their respective purchase average.
3. From the output, it is clearly seen that purchase average for member male in fashion accessories is 287.21296 and purchase average for normal female in fashion accessories is 312.5457
4. Therefore, we conclude that female non-members are bringing in more revenue per purchase of fashion accessories than male members in fashion accessories. Hence, retailers who believe female non-members are bringing in more revenue per purchase of fashion accessories is right.

Translation

Translation: Select average of total, gender, product line, customer type from invoices table join product line table join customer type table where customer type id is 1003 and product line type is Fashion accessories union select purchase average, gender, product line, customer type from invoices table join product line table join customer type table where customer type id is 1002 and product line type is Fashion accessories.

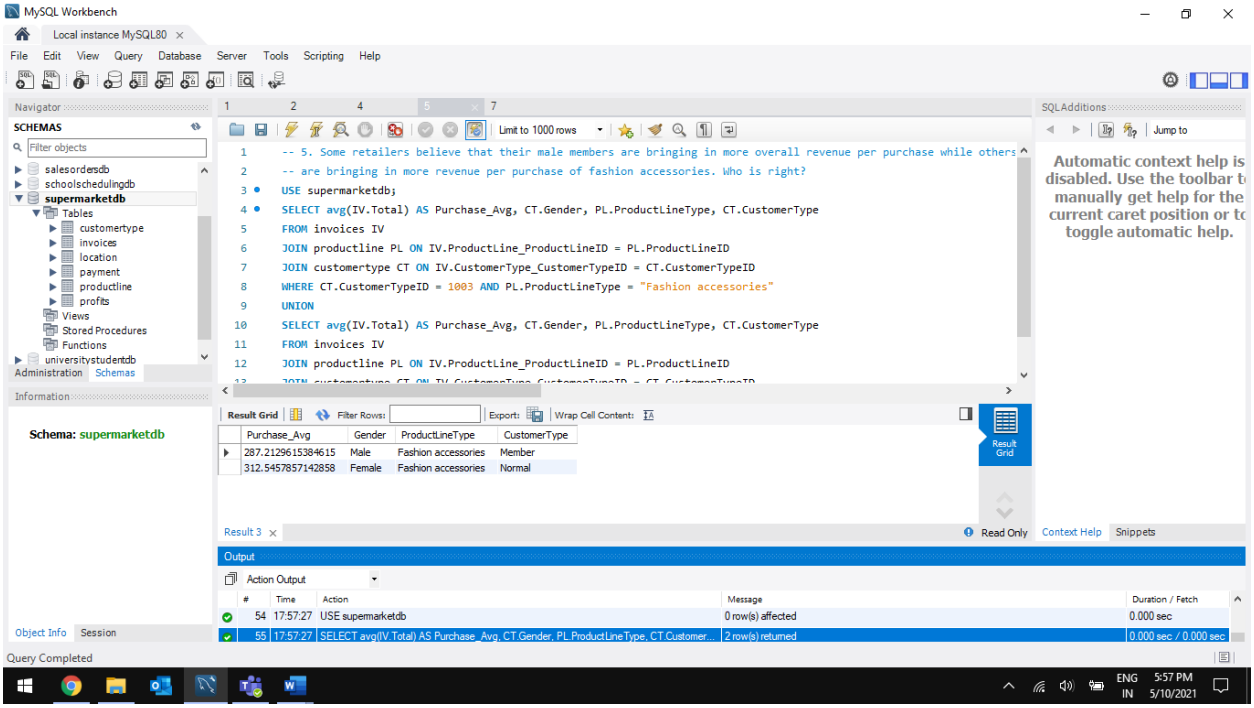
Cleanup: Select avg(IV.Total) AS Purchase_Avg, Gender, ProductLineType, CustomerType from invoices IV Join productline on ProductLine_ProductLineID = ProductLineID join customertype on CustomerType_CustomerTypeID = CT.CustomerTypeID where CustomerTypeID = 1003 and ProductLineType = "Fashion accessories" union select avg(IV.Total) AS Purchase_Avg, Gender, ProductLineType, CustomerType from invoices join productline on ProductLine_ProductLineID = ProductLineID join customertype CustomerType_CustomerTypeID = CustomerTypeID where CustomerTypeID = 1002 and ProductLineType = "Fashion accessories";

Screen Shot of SQL Query and Results

SQL Query:

```
USE supermarketdb;

SELECT avg(IV.Total) AS Purchase_Avg, CT.Gender, PL.ProductLineType, CT.CustomerType

FROM invoices IV

JOIN productline PL ON IV.ProductLine_ProductLineID = PL.ProductLineID

JOIN customertype CT ON IV.CustomerType_CustomerTypeID = CT.CustomerTypeID

WHERE CT.CustomerTypeID = 1003 AND PL.ProductLineType = "Fashion accessories"

UNION

SELECT avg(IV.Total) AS Purchase_Avg, CT.Gender, PL.ProductLineType, CT.CustomerType
```

```
FROM invoices IV

JOIN productline PL ON IV.ProductLine_ProductLineID = PL.ProductLineID

JOIN customertype CT ON IV.CustomerType_CustomerTypeID = CT.CustomerTypeID

WHERE CT.CustomerTypeID = 1002 AND PL.ProductLineType = "Fashion accessories";
```



SQL Query 7

Question

Yangon calls itself the most eWallet-friendly city for health and beauty while Mandalay calls itself a haven for cash spending. Does the data support their claims?

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

1. As per the requirement we need three tables to be joined, that is, the invoice table, location, payment and product line table.
2. The SQL query returns six rows consisting of Purchase total using Ewallet payment type in Yangon, Mandalay, Naypyitaw city and Purchase total using Cash payment type in Yangon Yangon, Mandalay, Naypyitaw city.
3. Therefore, we conclude that Yangon having a purchase total of 5155.4684 is not the most ewallet friendly city for health and beauty as Mandalay has a higher purchase total of 6054.7725. We also conclude that Mandalay with a purchase total of 35339.4614 is not a haven for cash spending as Naypyitaw has a higher purchase total of 43085.8574.

Translation

Translation: Select city, sum of total, payment type from invoices join product line join location join payment where ProductLineType = Health and beauty and PaymentType=Ewallet group by City union select city, sum of total, payment type from invoices join product line join location join payment where PaymentType=Cash group by City.

Cleanup: Select sum(IV.Total) as Purchase_Total, PaymentType, City from invoices join productline on ProductLine_ProductLineID = ProductLineID join location on Location_BranchID = BranchID join payment on Payment_PaymentTypeID = PaymentTypeID where ProductLineType = Health and beauty" and PaymentType=Ewallet group by City union Select sum(IV.Total) as Purchase_Total, PaymentType, City from invoices join productline on ProductLine_ProductLineID = ProductLineID join location on Location_BranchID = BranchID join payment on Payment_PaymentTypeID = PaymentTypeID where PaymentType=Cash group by City.

Screen Shot of SQL Query and Results

SQL Query:

```
USE supermarketdb;

SELECT sum(IV.Total) AS Purchase_Total, PY.PaymentType, LT.City

FROM invoices IV

JOIN productline PL ON IV.ProductLine_ProductLineID = PL.ProductLineID

JOIN location LT ON IV.Location_BranchID = LT.BranchID

JOIN payment PY ON IV.Payment_PaymentTypeID = PY.PaymentTypeID

WHERE PL.ProductLineType = "Health and beauty" AND PY.PaymentType="Ewallet"

GROUP BY LT.City

UNION
```

```
SELECT sum(IV.Total) AS Purchase_Total, PY.PaymentType, LT.City

FROM invoices IV

JOIN productline PL ON IV.ProductLine_ProductLineID = PL.ProductLineID

JOIN location LT ON IV.Location_BranchID = LT.BranchID

JOIN payment PY ON IV.Payment_PaymentTypeID = PY.PaymentTypeID

WHERE PY.PaymentType = "Cash"

GROUP BY LT.City;
```