20/06/2023

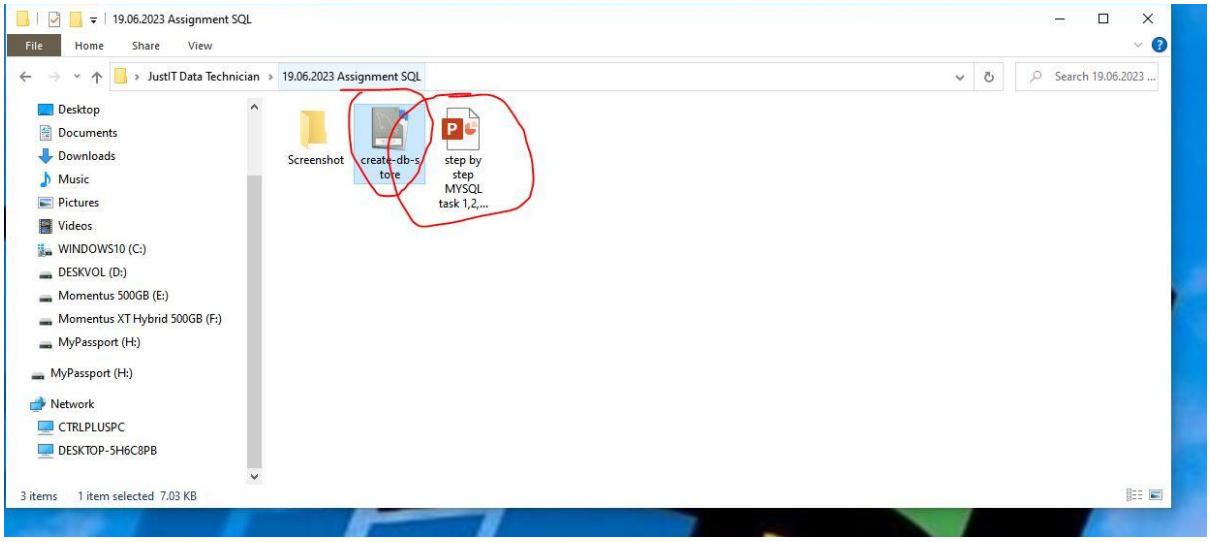
SQL Assignment

Atikur Rahaman

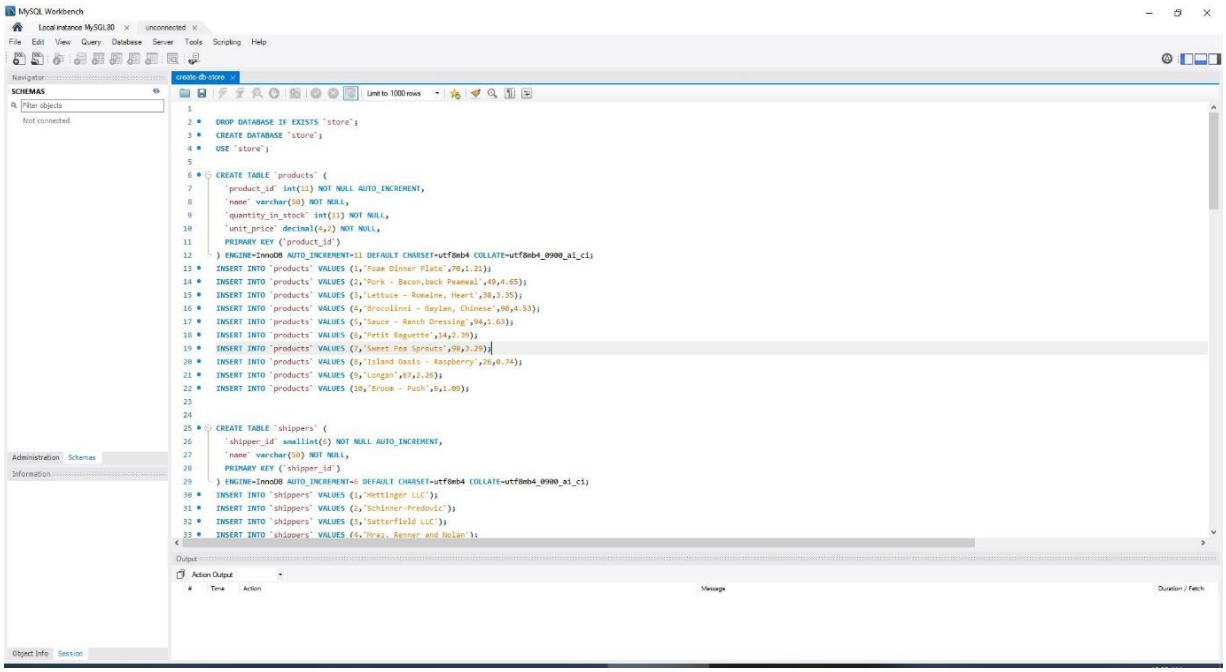
JUSTIT
DATA TECHNICIAN

Table of Contents

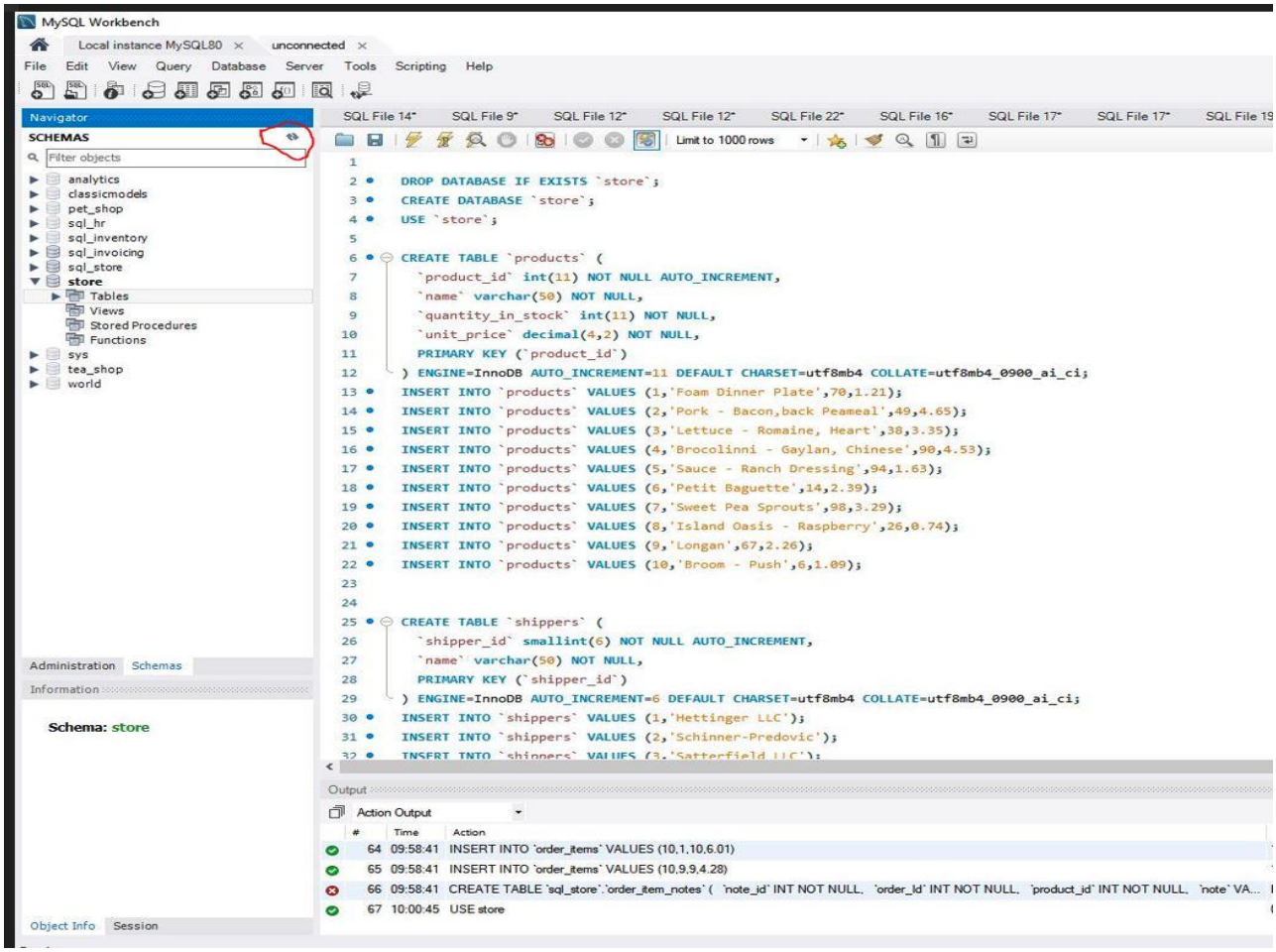
1. Downloaded the database and Instruction file.



2. I've run the database by double clicking on file called create-db-store and then copied all the queries by pressing Ctrl +a from the keyboard and pasted it by pressing Ctrl+v in a new query tab.



Finally refreshed the database by clicking the refresh button from the navigation panel to make sure that ‘store’ database has been created.

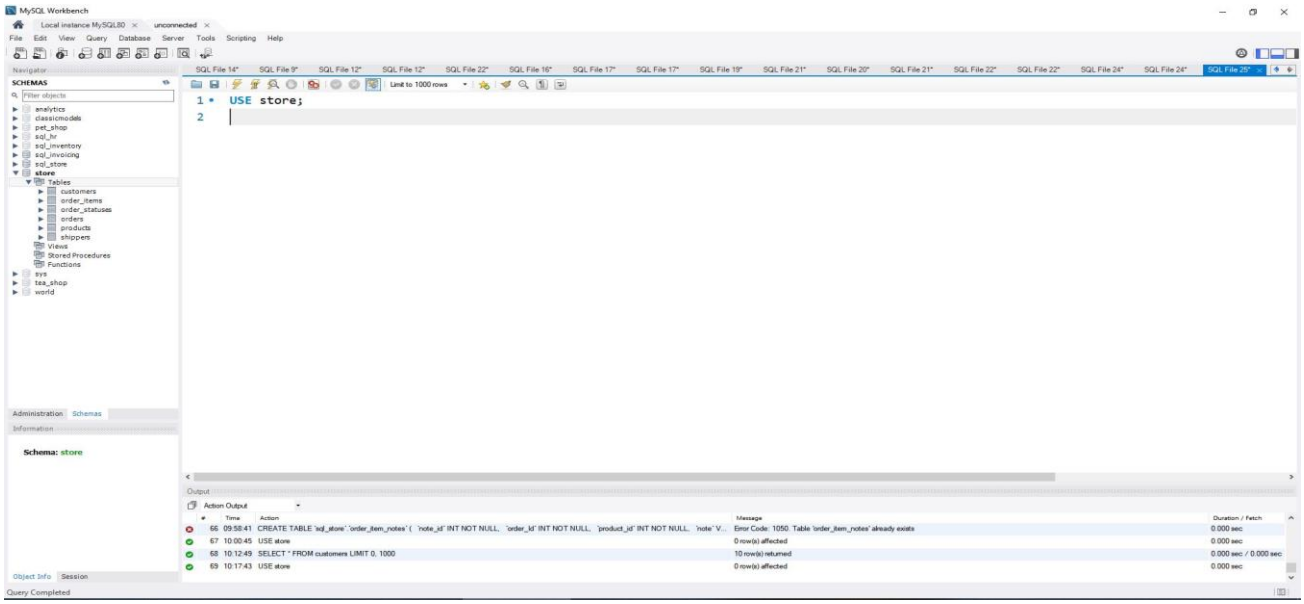


Query 1

Connect to the database and view the customer table.

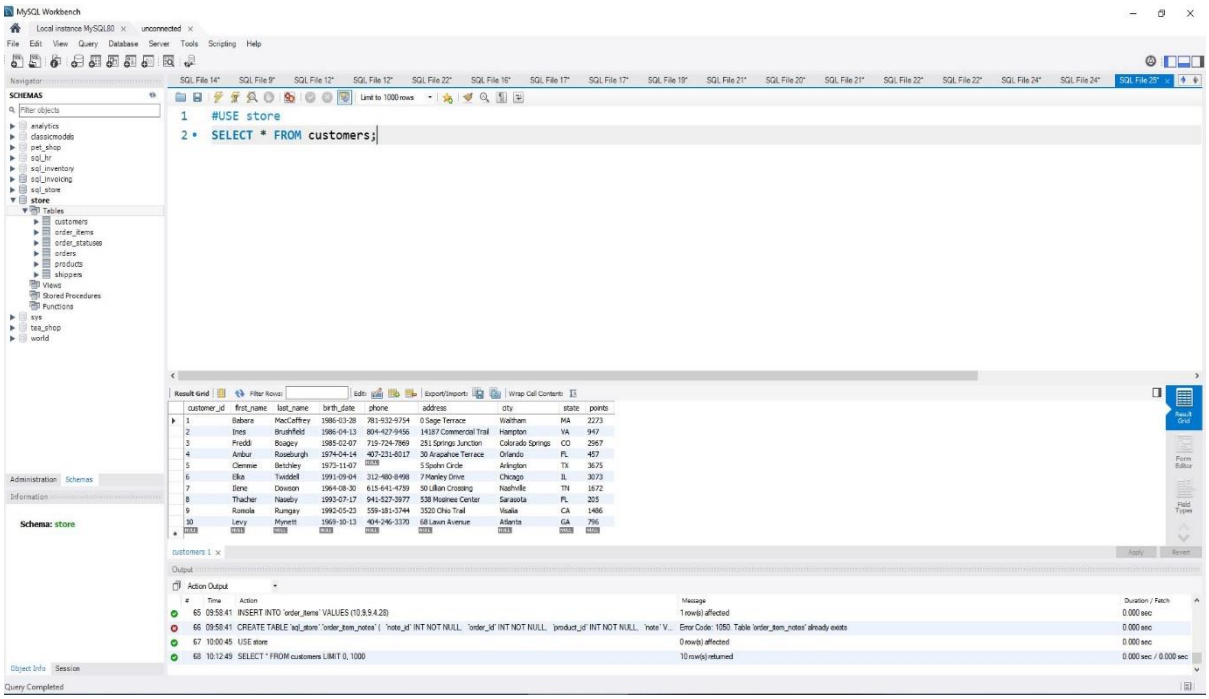
USE store;

(I have used USE statement to connect the database to follow up the tasks)



SELECT * FROM customers;

I have used SELECT statement to see all the columns within customer table for me to get familiar with columns and data.

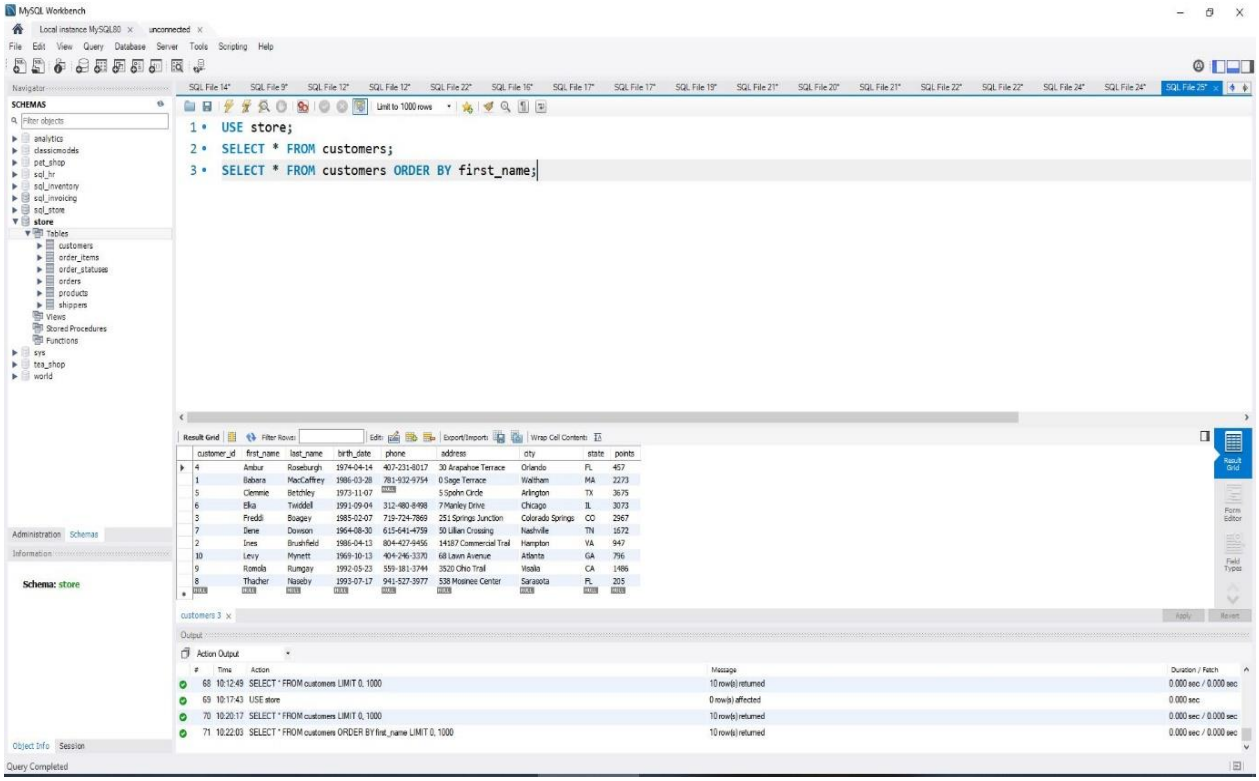


Query 1 Continued…

Arrange first_name in ascending order.

Ans: SELECT * FROM customers ORDER BY first_name;

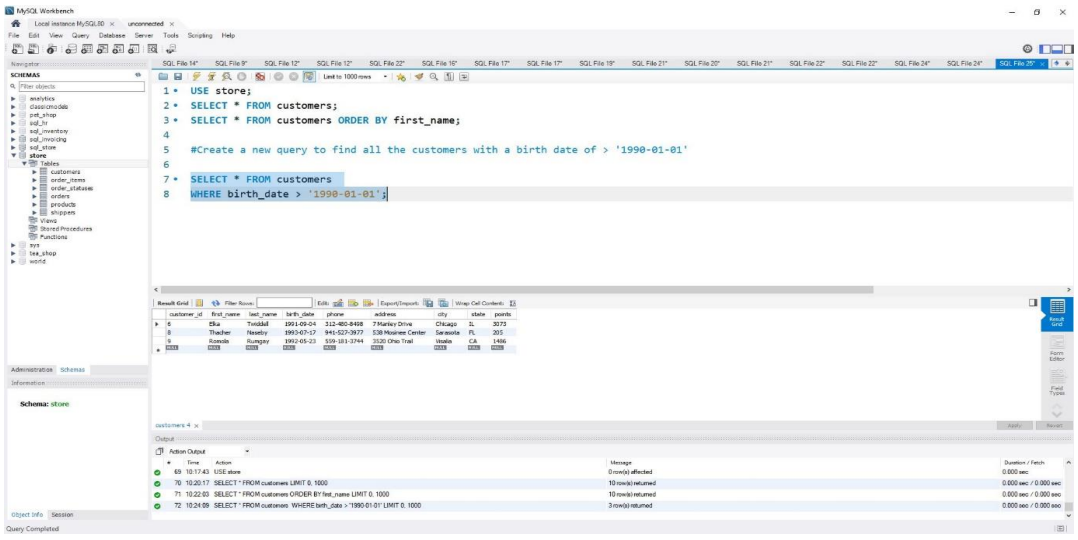
By default, MySQL sort the result in Ascending order. That's why I didn't include ASC at the end of the statement.



Query 2: Create a new query to find all the customers with a birth date of > '1990-01-01'.

Ans: SELECT * FROM customers WHERE birth_date > '1990-01-01'

I had to put date inside quotation mark to make it readable to SQL. Otherwise, MySQL may think it is for calculation.

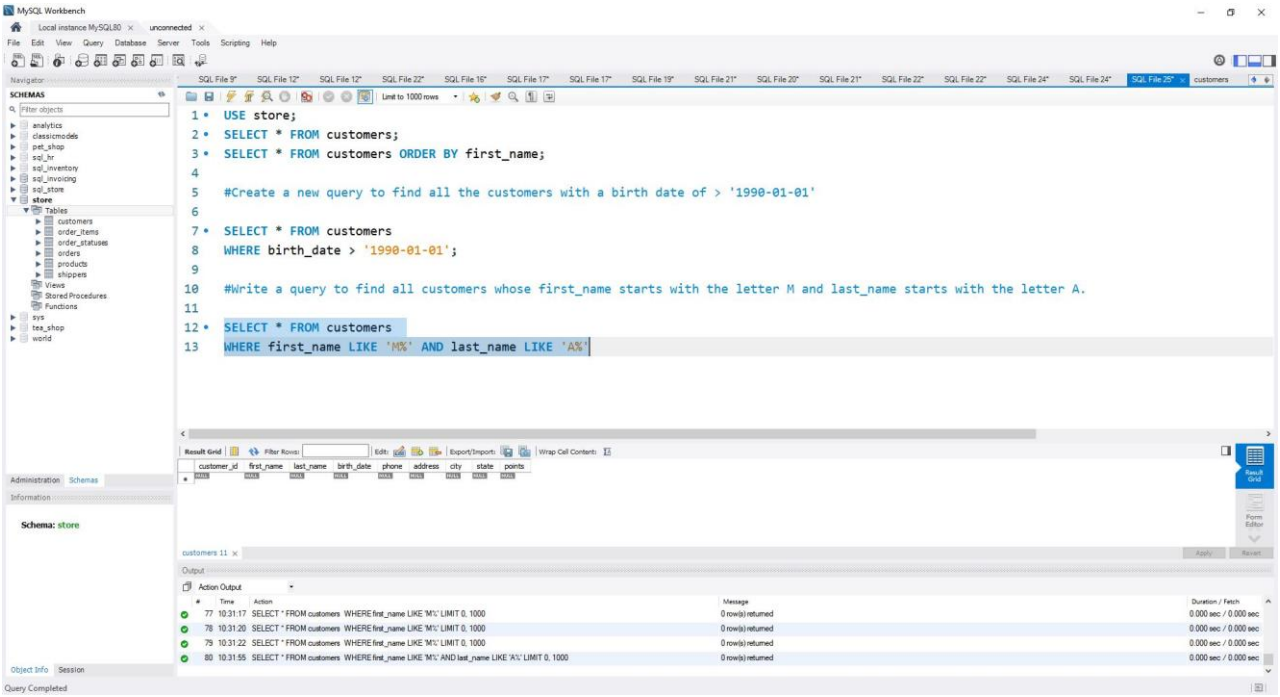


Query 3.1

Write a query to find all customers whose first_name starts with the letter M and last_name starts with the letter A.

Ans: SELECT * FROM customers WHERE first_name LIKE ‘M%’ AND last_name LIKE ‘A%’;

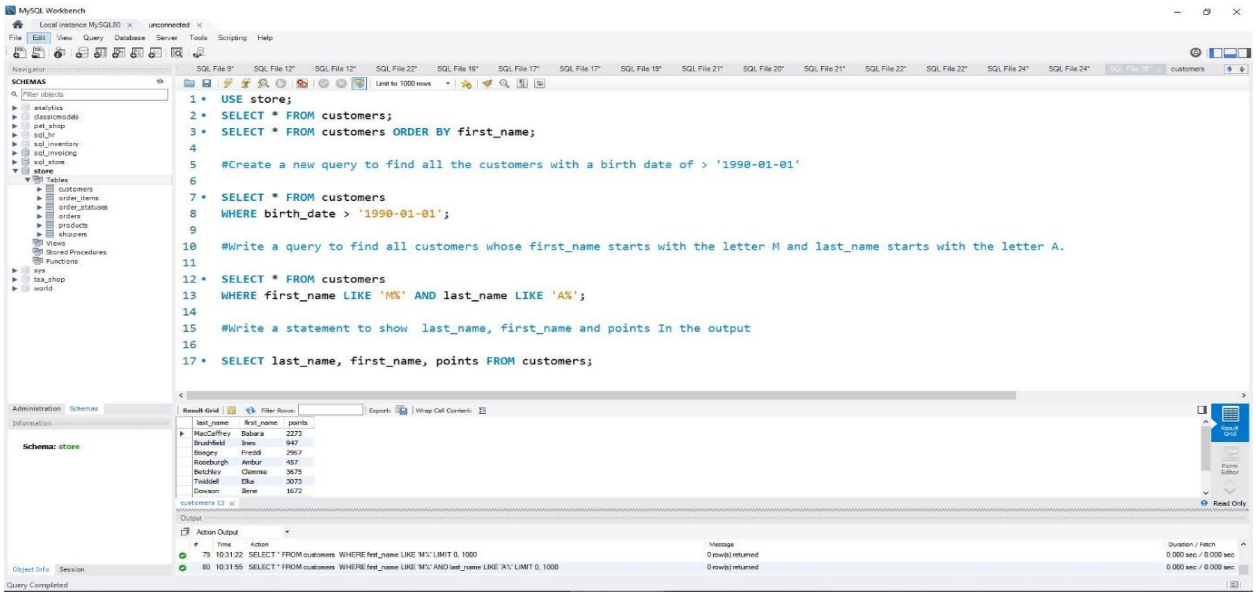
There was NO Customers with that name conditions. So, it pulls NULL value. But syntax wise it is correct. I have tested it with other characters.



Query 3.2: Write a statement to show last_name, first_name and points in the output

Ans: SELECT last_name, first_name, points FROM customers;

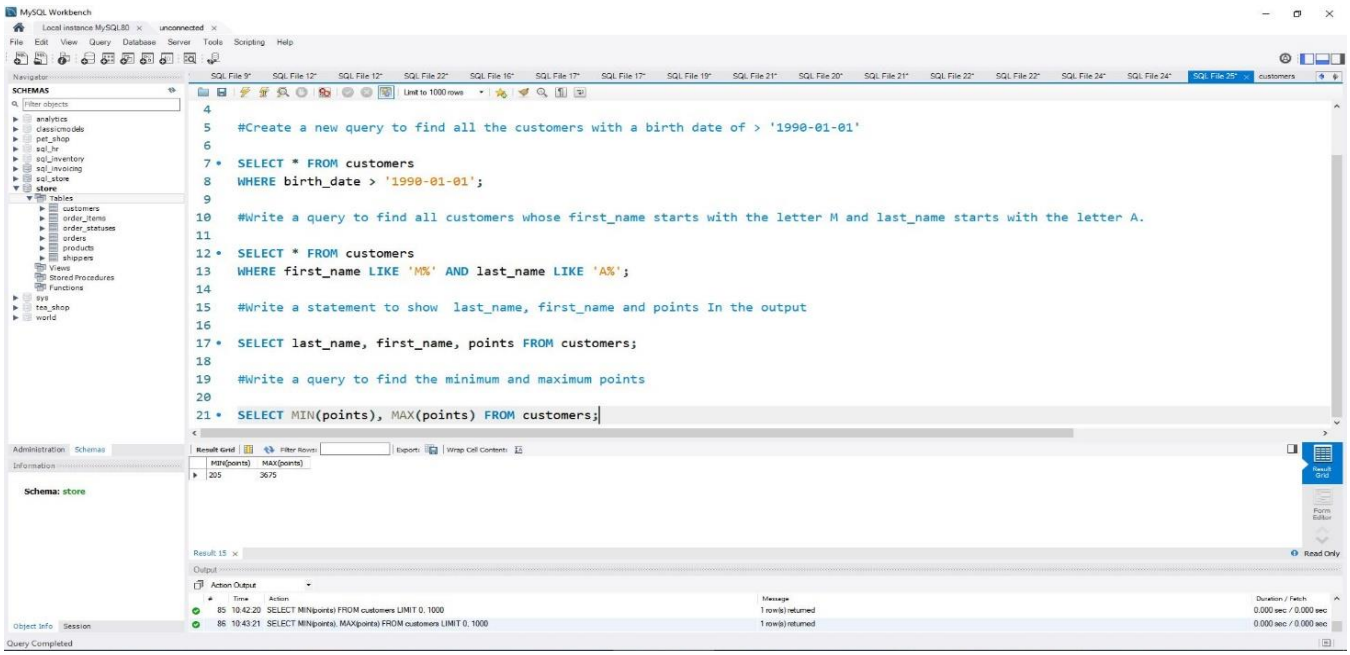
I have used SELECT statement and specify the name of the columns that I was asked to retrieve from the ‘customers’ table.



Task 3: Write a query to find the minimum and maximum points

Ans: SELECT MIN(points), MAX(points) FROM customers;

I've used MIN() and MAX() Functions to perform this query.

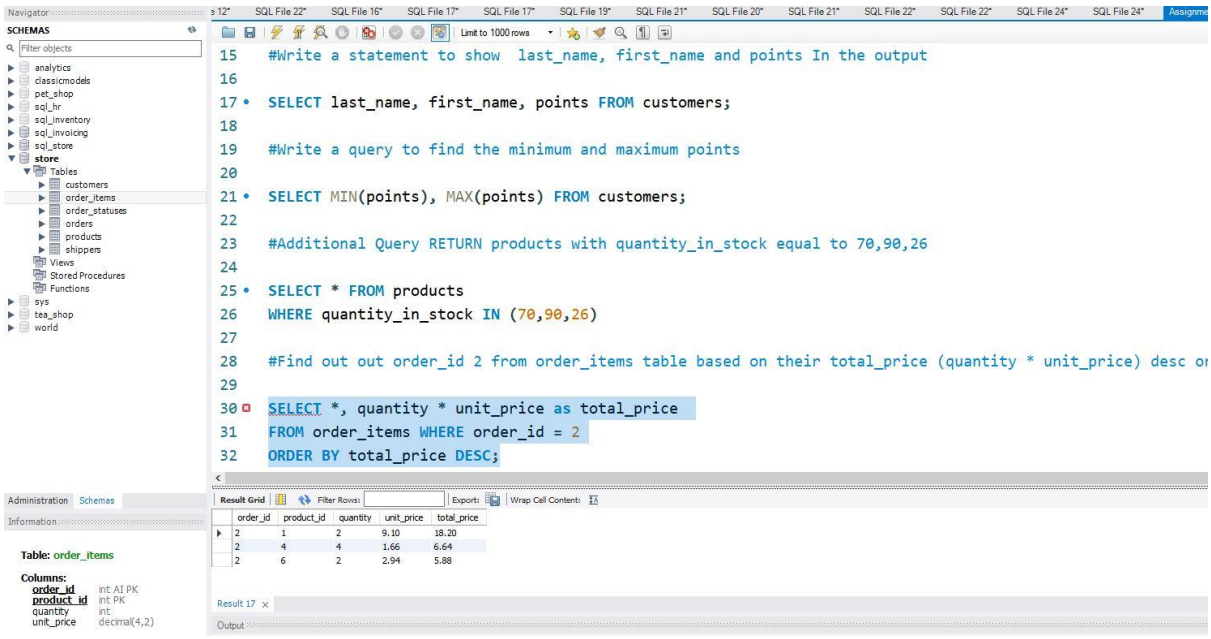


Additional Queries 1:

RETURN products with quantity_in_stock equal to 70,90,26

Ans: SELECT * FROM products WHERE quantity_in_stock IN (70,90,26);

I used IN operator to perform this query.



Additional queries 2:

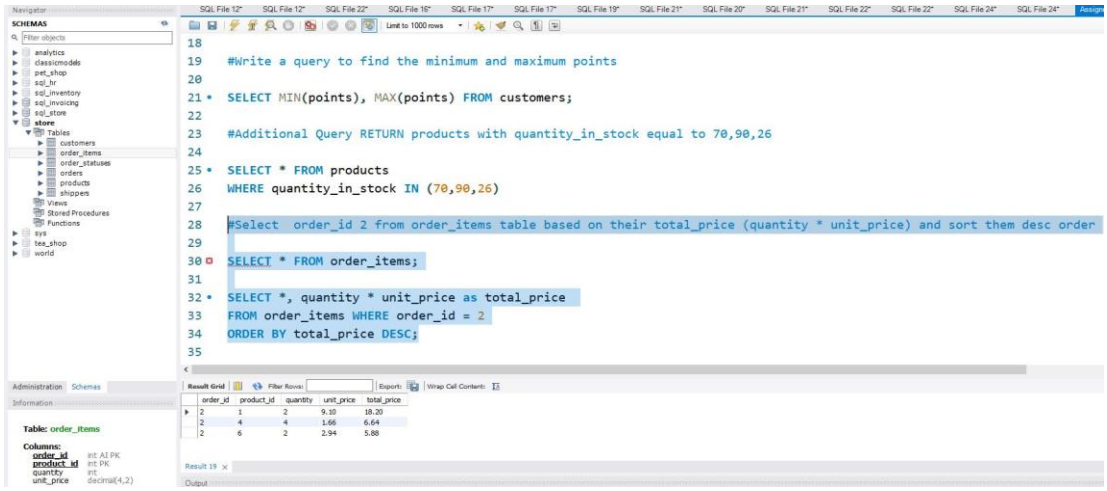
Select order_id '2' from order_items table based on their total_price (quantity * unit_price) and sort them desc order.

Ans: SELECT *, quantity * unit_price AS total_price

FROM order_items WHERE order_id = 2

ORDER BY total_price DESC;

There was no total_price colum in the table. So I used total_price as an ALIAS



Additional Query 3:

Write a query to produce a report with customer_id, first_name, last_name, points and type.

Type column doesn't exist in customer's table. So, we have to calculate this column with the values by following the below conditions

Points less than 2,000 'Bronze'

Points between 2,000 and 3,000 'Silver' and

Points more than 3,000 'Gold'

Ans: SELECT customer_id, first_name, last_name, points, 'Bronze' as type

FROM customers

WHERE points > 2000

UNION

SELECT customer_id, first_name, last_name, points, 'Silver' as type

FROM customers

WHERE points between 2000 and 3000

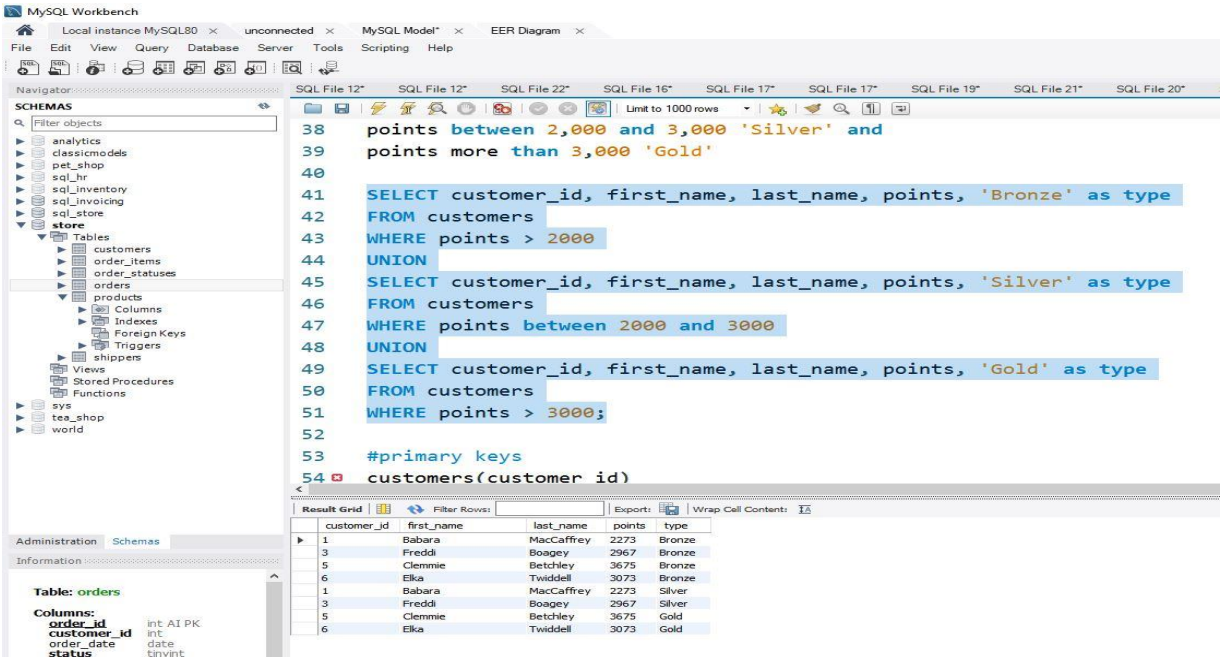
UNION

SELECT customer_id, first_name, last_name, points, 'Gold' as type

FROM customers

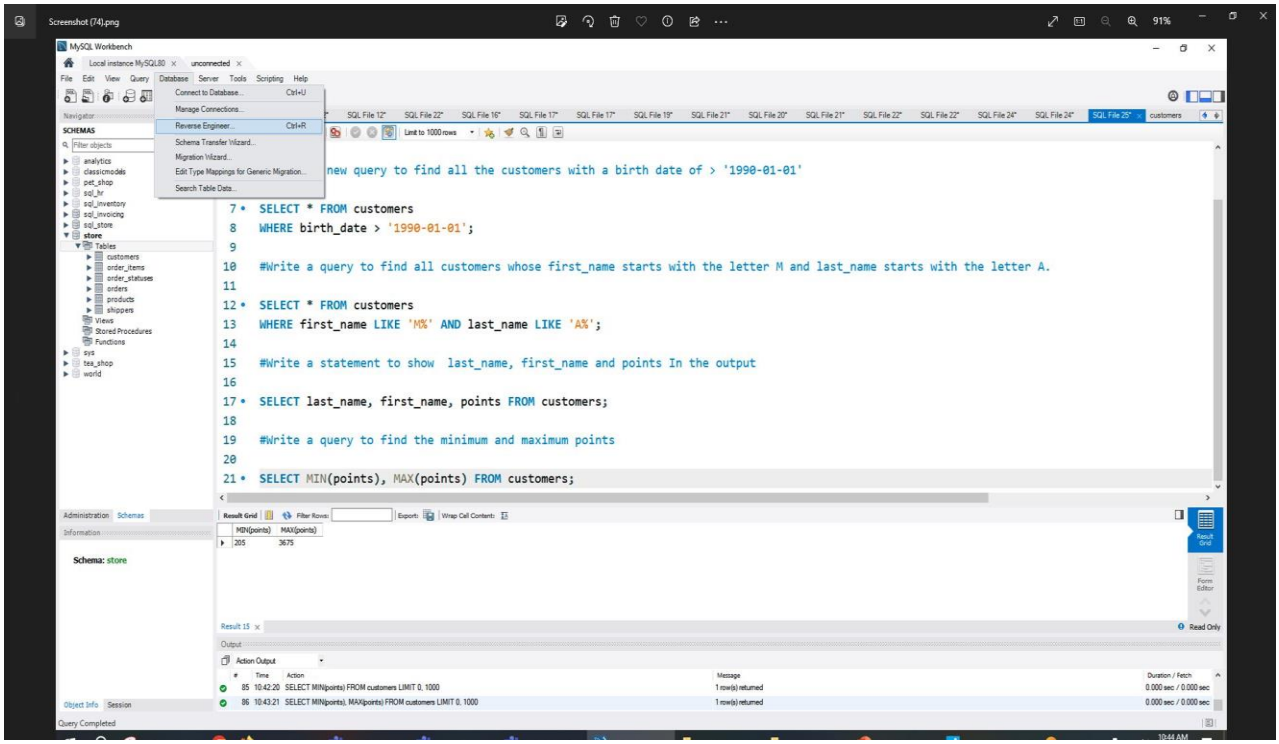
WHERE points > 3000;

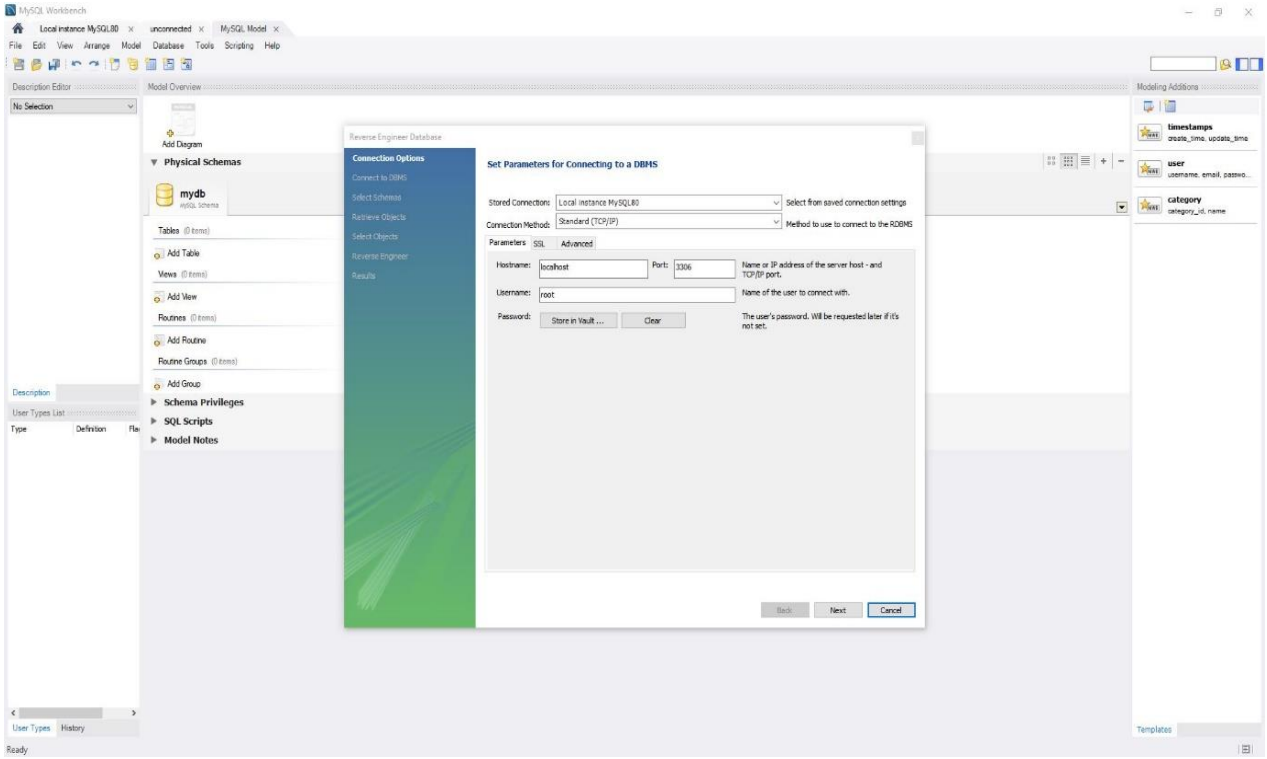
I have used **UNION** clause to produce and combine this report.

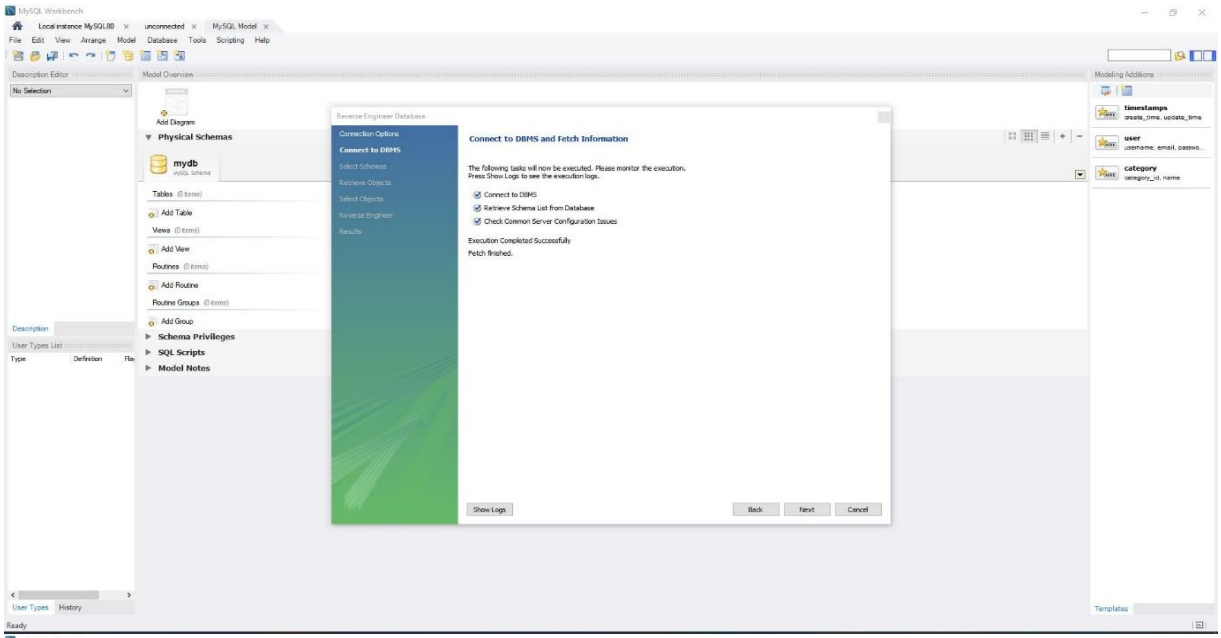


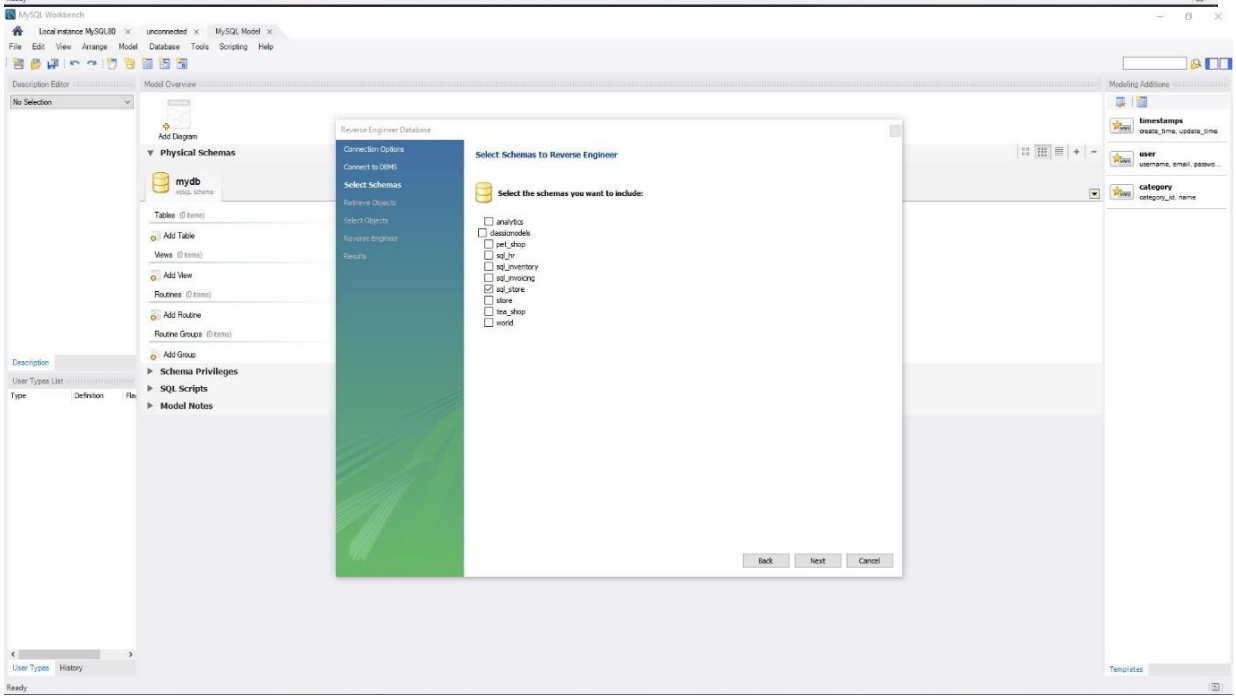
Creating an EER Diagram

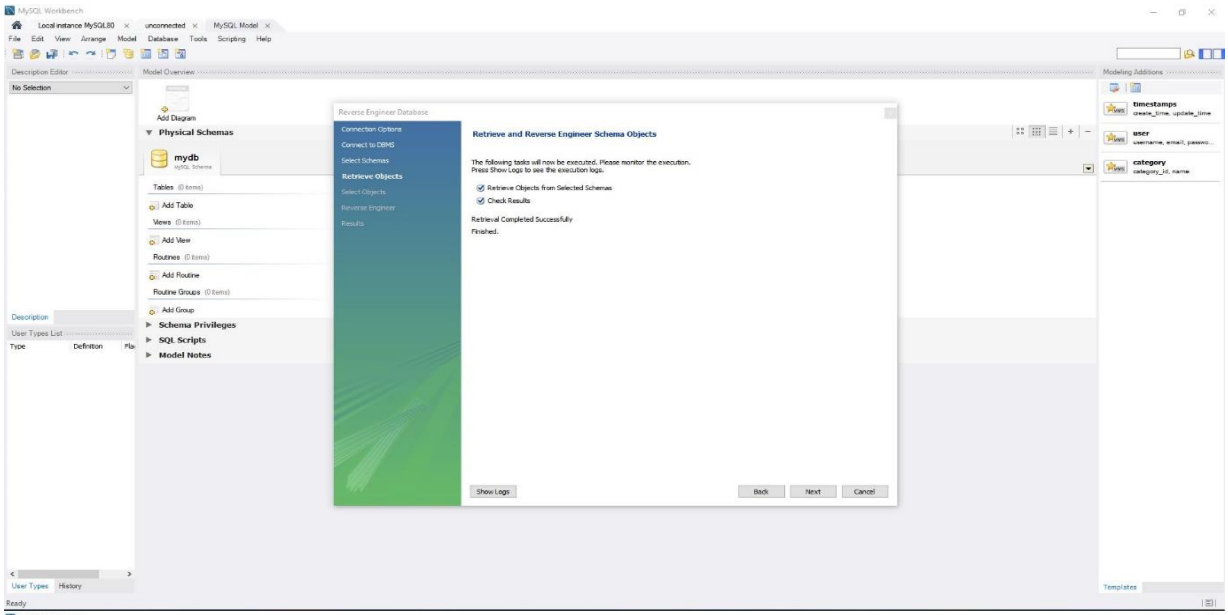
Followings are the Screenshots of all the steps of creating an EER Diagram

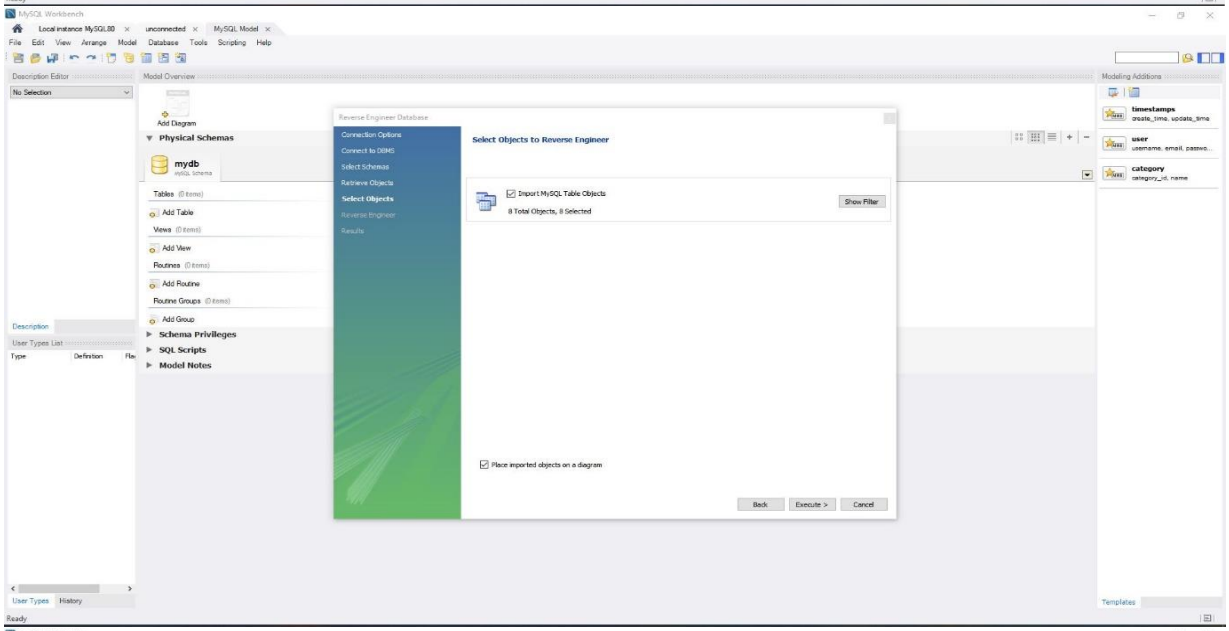


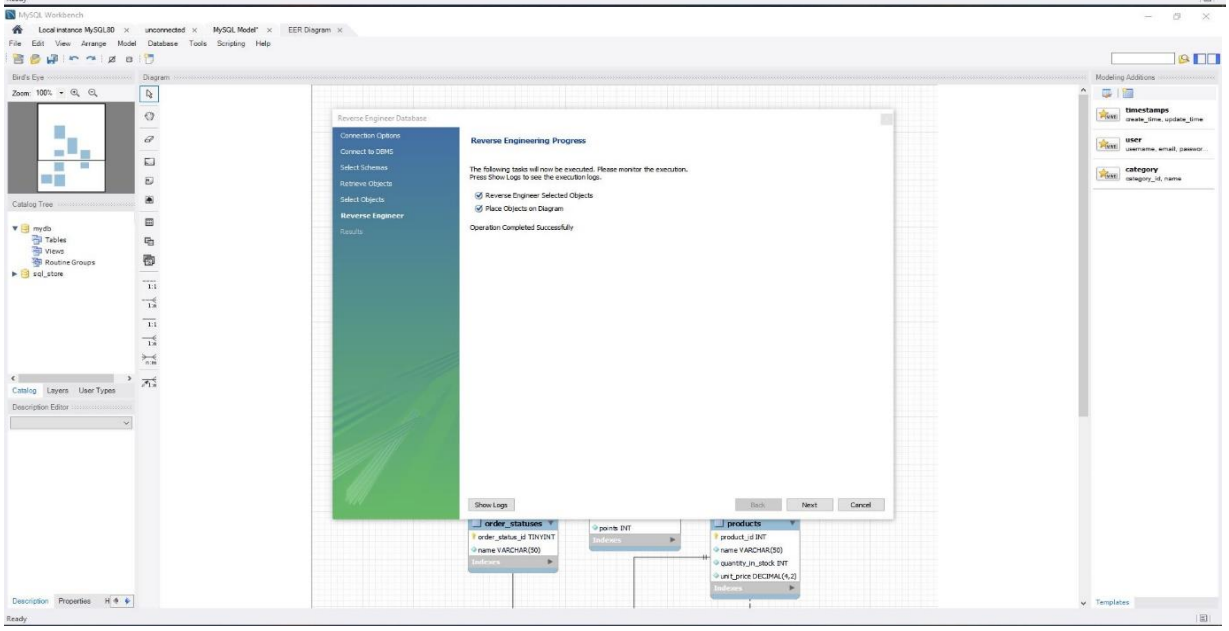


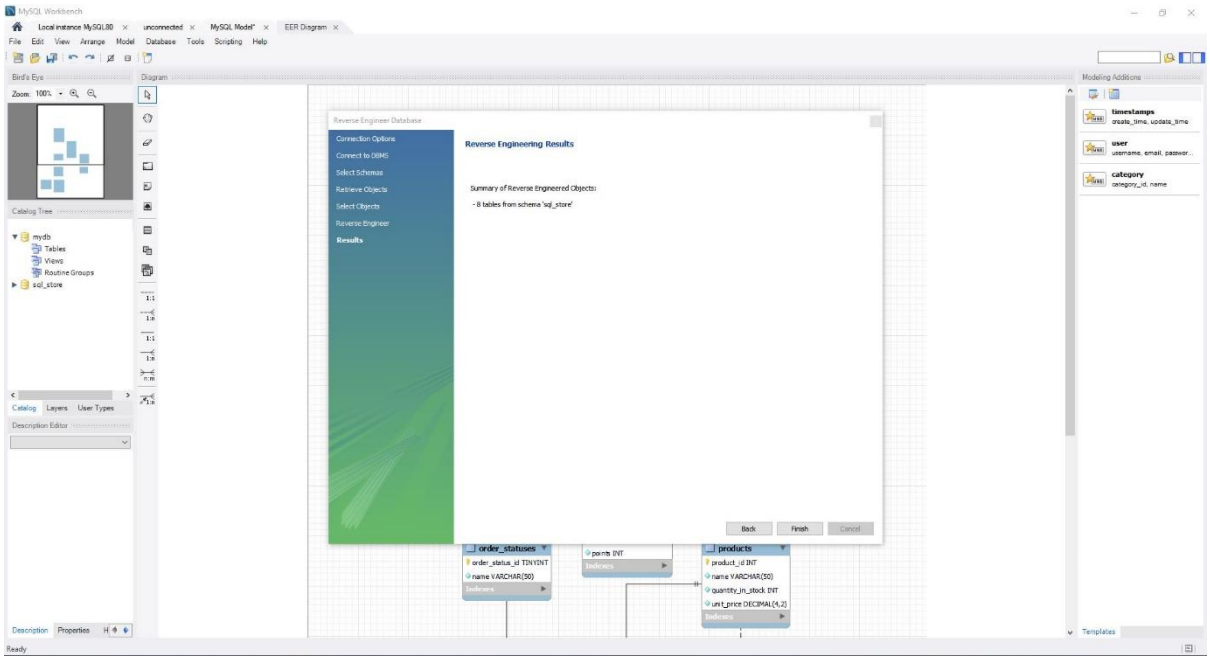

MySQL Workbench
Local instance MySQL80
unconnected
MySQL Model
File Edit View Arrange Model Database Tools Scripting Help
Description Editor
No Selection
Model Overview
Add Diagram
Physical Schemas
mydb
MySQL Schema
Tables (0 items)
Add Table
Views (0 items)
Add View
Routines (0 items)
Add Routine
Routine Groups (0 items)
Add Group
Schema Privileges
SQL Scripts
Model Notes
Description
User Types List
Type
Definition
User Types
History
Ready
Modeling Additions
timestamps
create_time, update_time
user
username, email, passwo...
category
category_id, name
Templates
Reverse Engineer Database
Connection Options
Connect to DBMS
Select Schemas
Retrieve Objects
Select Objects
Reverse Engineer
Results
Connect to DBMS and Fetch Information
The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.
Connect to DBMS
Retrieve Schema List from Database
Check Common Server Configuration Issues
Execution Completed Successfully
Fetch finished.
Show Logs
Back
Next
Cancel


MySQL Workbench
Local instance MySQL80
unconnected
MySQL Model
File Edit View Arrange Model Database Tools Scripting Help
Description Editor
No Selection
Model Overview
Add Diagram
Physical Schemas
mydb
MySQL Schema
Tables (0 items)
Add Table
Views (0 items)
Add View
Routines (0 items)
Add Routine
Routine Groups (0 items)
Add Group
Schema Privileges
SQL Scripts
Model Notes
Description
User Types List
Type
Definition
User Types
History
Ready
Modeling Additions
timestamps
create_time, update_time
user
username, email, passwo...
category
category_id, name
Templates
Reverse Engineer Database
Connection Options
Connect to DBMS
Select Schemas
Retrieve Objects
Select Objects
Reverse Engineer
Results
Select Schemas to Reverse Engineer
Select the schemas you want to include:
analytics
classicmodels
pet_shop
sql_hr
sql_inventory
sql_invoicing
sql_store
store
tine_shop
world
Back
Next
Cancel

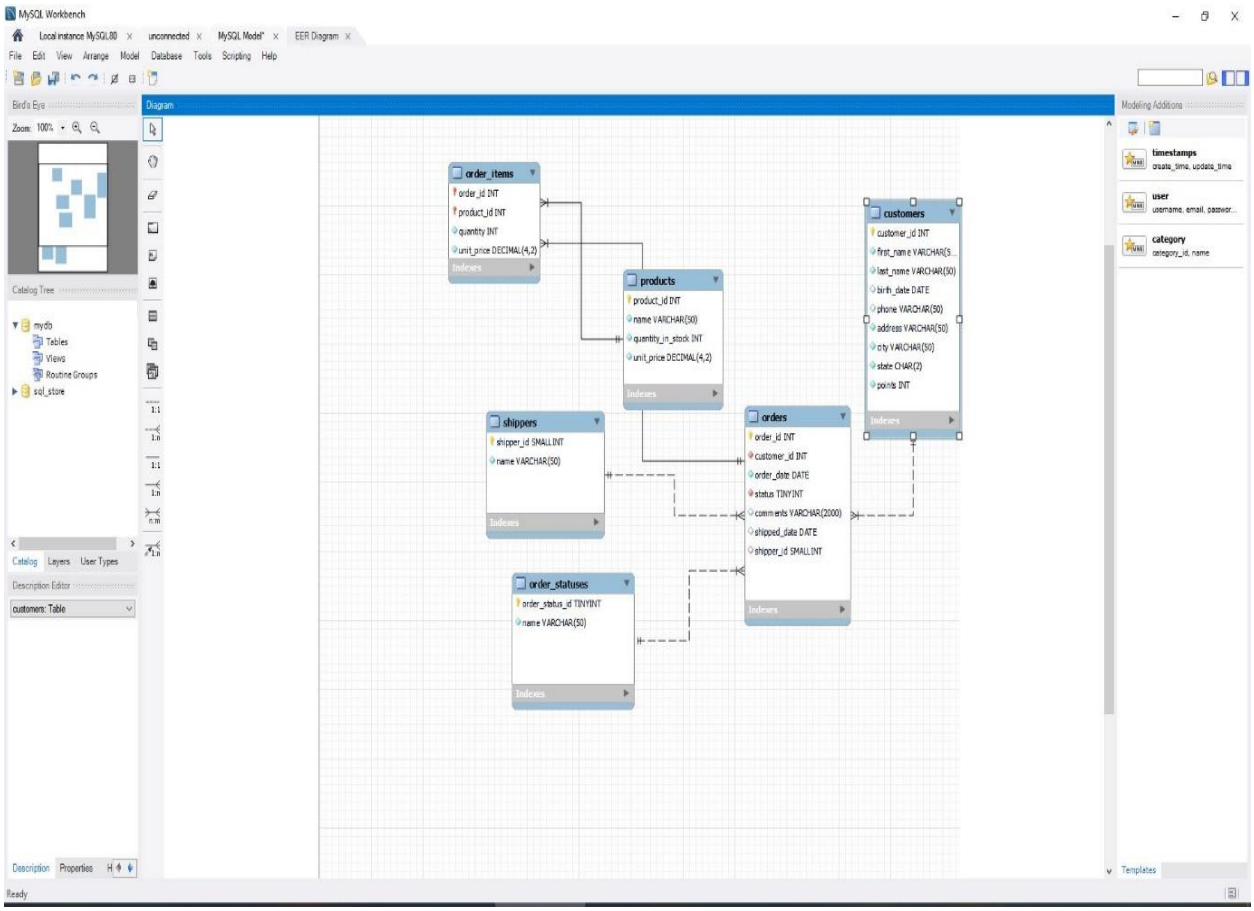

Retrieve and Reverse Engineer Schema Objects
Retrieve Objects from Selected Schemas
Check Results
Retrieval Completed Successfully
Finished.
Show Logs
Back
Next
Cancel


Select Objects to Reverse Engineer
Import MySQL Table Objects
8 Total Objects, 8 Selected
Show Filter
Place imported objects on a diagram
Back
Execute >
Cancel


Reverse Engineering Progress
Reverse Engineer Selected Objects
Place Objects on Diagram
Operation Completed Successfully
Show Logs
Back
Next
Cancel


MySQL Workbench
Reverse Engineer Database
Connection Options
Connect to DBMS
Select Schemas
Retrieve Objects
Select Objects
Reverse Engineer
Results
Reverse Engineering Results
Summary of Reverse Engineered Objects:
- 8 tables from schema 'sql_store'
Back
Finish
Cancel

EER Diagram


MySQL Workbench
Diagram
order_items
order_id INT
product_id INT
quantity INT
unit_price DECIMAL(4,2)
products
product_id INT
name VARCHAR(50)
quantity_in_stock INT
unit_price DECIMAL(4,2)
customers
customer_id INT
first_name VARCHAR(5...
last_name VARCHAR(50)
birth_date DATE
phone VARCHAR(50)
address VARCHAR(50)
city VARCHAR(50)
state CHAR(2)
points INT
shippers
shipper_id SMALLINT
name VARCHAR(50)
orders
order_id INT
customer_id INT
order_date DATE
status TINYINT
comments VARCHAR(2000)
shipped_date DATE
shipper_id SMALLINT
order_statuses
order_status_id TINYINT
name VARCHAR(50)