**MySQL Assignment Part-1**

**ASSIGNMENT 2**



Table of Contents

[INTRODUCTION TO MYSQL 2](#_Toc163939677)

[WHAT IS SQL? 2](#_Toc163939678)

[SQL QUERIES EXAMPLES 2](#_Toc163939679)

[MYSQL WORKBENCH 3](#_Toc163939680)

[DATABASE MIGRATION 4](#_Toc163939681)

[QUERIES EXECUTION 6](#_Toc163939682)

[QUERY 1 6](#_Toc163939683)

[QUERY 2 7](#_Toc163939684)

[TASK 1 8](#_Toc163939685)

[TASK 2 9](#_Toc163939686)

[TASK 3 10](#_Toc163939687)

[Task4 11](#_Toc163939688)

[Task5 11](#_Toc163939689)

[Task6 12](#_Toc163939690)

[EER DIAGRAM 13](#_Toc163939691)

# INTRODUCTION TO MYSQL

MySQL is an open-source relational database management system used to store, manage, and retrieve data. It uses a SQL (Structured Query Language) syntax. To interact with a MySQL database users can write SQL queries to create, modify, and delete tables, as well as insert, update, and select data. Each table is identified by a unique name and consists of fields (columns) and records (rows) that contain the actual data. MySQL processes these queries and returns the requested data or performs the requested operation.

When data is stored in a MySQL database, it is organized in a way that ensures data integrity and consistency. MySQL enforces data integrity by defining rules for each table that specify what types of data can be stored in each column, and by ensuring that data is entered in a valid format.

MySQL can be used as a standalone database or can be integrated with web applications through various programming languages and APIs. This allows developers to create dynamic web applications that can store and retrieve data from a MySQL database. MySQL is commonly used in web applications and can be integrated with programming languages such as PHP, Python, and Java.

# WHAT IS SQL?

SQL (Structured Query Language) is a programming language used to manage and manipulate relational databases. It is used to create, modify, and retrieve data from databases, as well as to perform operations on the data, such as sorting, filtering, and grouping.

SQL is commonly used in web development, data analytics, business intelligence, and many other fields where data needs to be stored, organized, and analysed. SQL is a standardized language, meaning that it can be used across different platforms and database management systems, such as MySQL, Oracle, SQL Server, and PostgreSQL.

SQL allows users to create tables, insert, update, and delete data, as well as perform complex queries that join multiple tables together. SQL also supports the creation of views, indexes, and stored procedures, which can improve performance and simplify complex database operations.

# SQL QUERIES EXAMPLES

1. SELECT statement:

This is used to retrieve data from a table.

SELECT \* FROM table\_name;

This query will retrieve all the columns and rows from the table called "table\_name".

1. WHERE clause:

This is used to filter data based on certain conditions.

SELECT \* FROM table\_name WHERE column\_name = 'value';

This query will retrieve all the columns and rows from the table called "table\_name" where the value in the column "column\_name" is equal to 'value'.

1. UPDATE statement:

This is used to update existing data in a table.

UPDATE table\_name SET column\_name = 'new\_value' WHERE column\_name = 'old\_value';

This query will update the column "column\_name" in the table called "table\_name" from 'old\_value' to 'new\_value'.

1. INSERT statement:

This is used to insert new data into a table.

INSERT INTO table\_name (column1, column2, column3) VALUES (value1, value2, value3);

This query will insert new data into the table called "table\_name" with values for columns 1, 2, and 3.

1. DELETE statement:

This is used to delete data from a table.

DELETE FROM table\_name WHERE column\_name = 'value';

This query will delete all the rows from the table called "table\_name" where the value in the column "column\_name" is equal to 'value’.

# MYSQL WORKBENCH

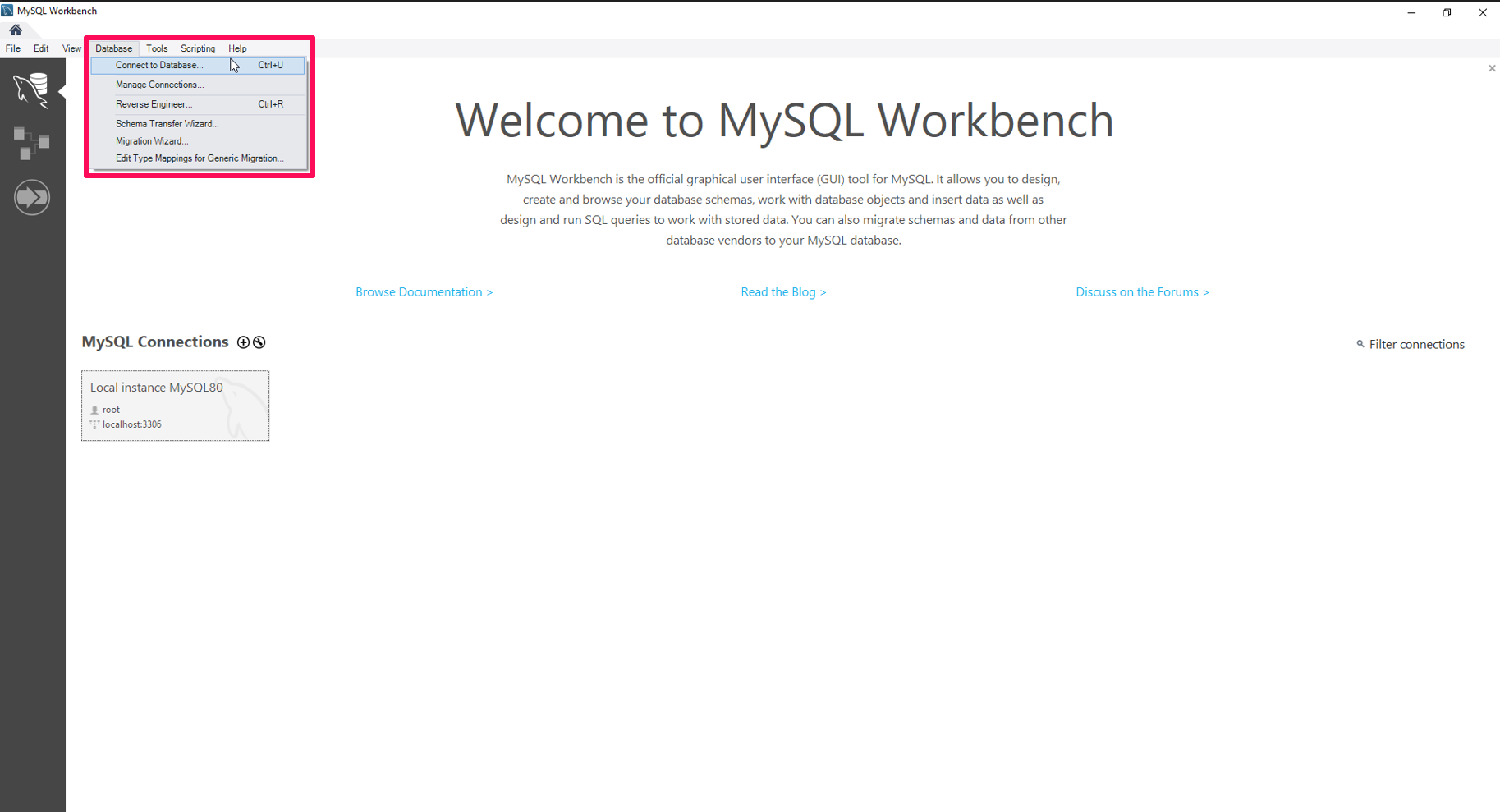
MySQL Workbench is a visual tool used to design, develop, and administer MySQL databases. It is a unified visual tool for database architects, developers, and DBAs. The tool provides a comprehensive graphical interface for designing, modeling, and managing databases.

MySQL Workbench includes features such as data modeling, SQL development, database administration, and migration tools. Users can create, edit, and manage database schemas, tables, indexes, and relations through an intuitive user interface. It also provides a query editor and debugger to help developers write and debug SQL code.

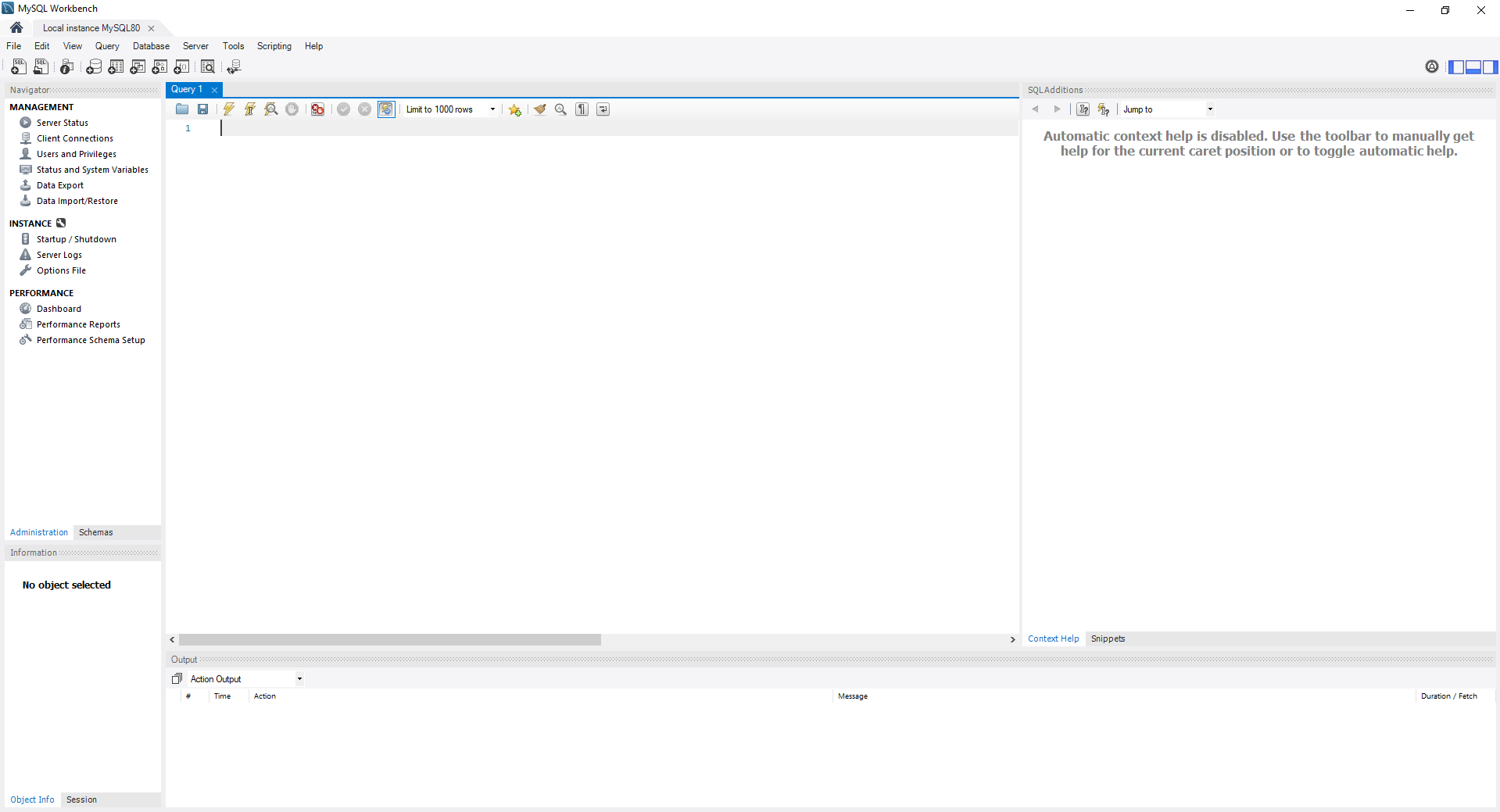
MySQL Workbench allows users to perform common administrative tasks such as backup and restore, user management, and server monitoring. It supports multiple database connections and allows users to create and manage multiple server instances.

## DATABASE MIGRATION

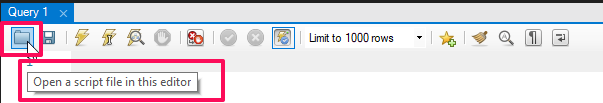
1. Navigate to MySQL Workbench and connect to the Local Database.

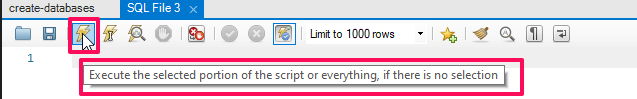


1. Overview of the MySQL Database after setup of the root user password.

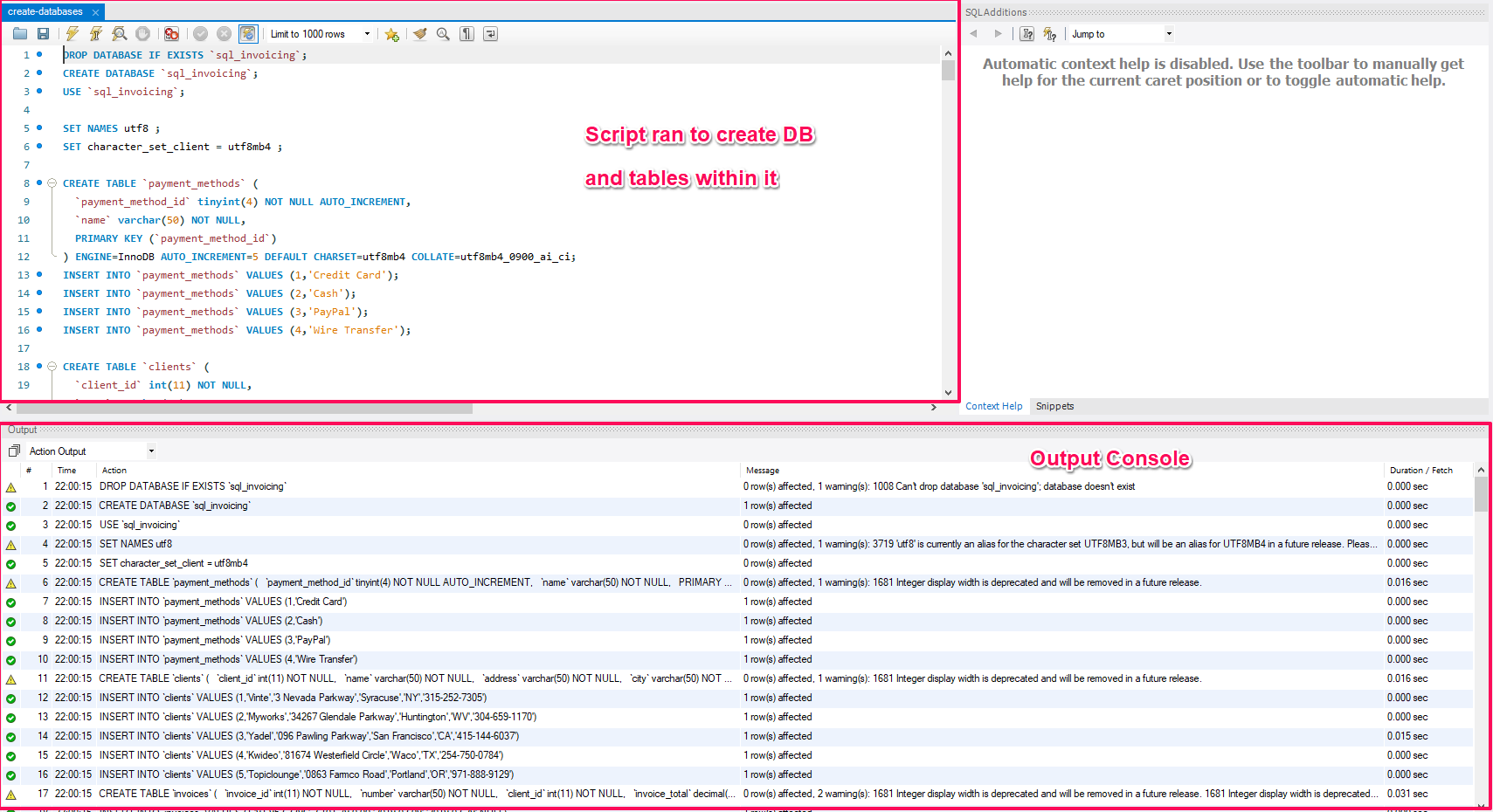


1. Open Script File provided in the query editor then execute.

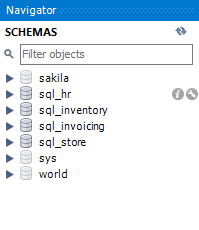




1. Output created after the file has been executed.



1. Navigator view with the Data and Tables created.



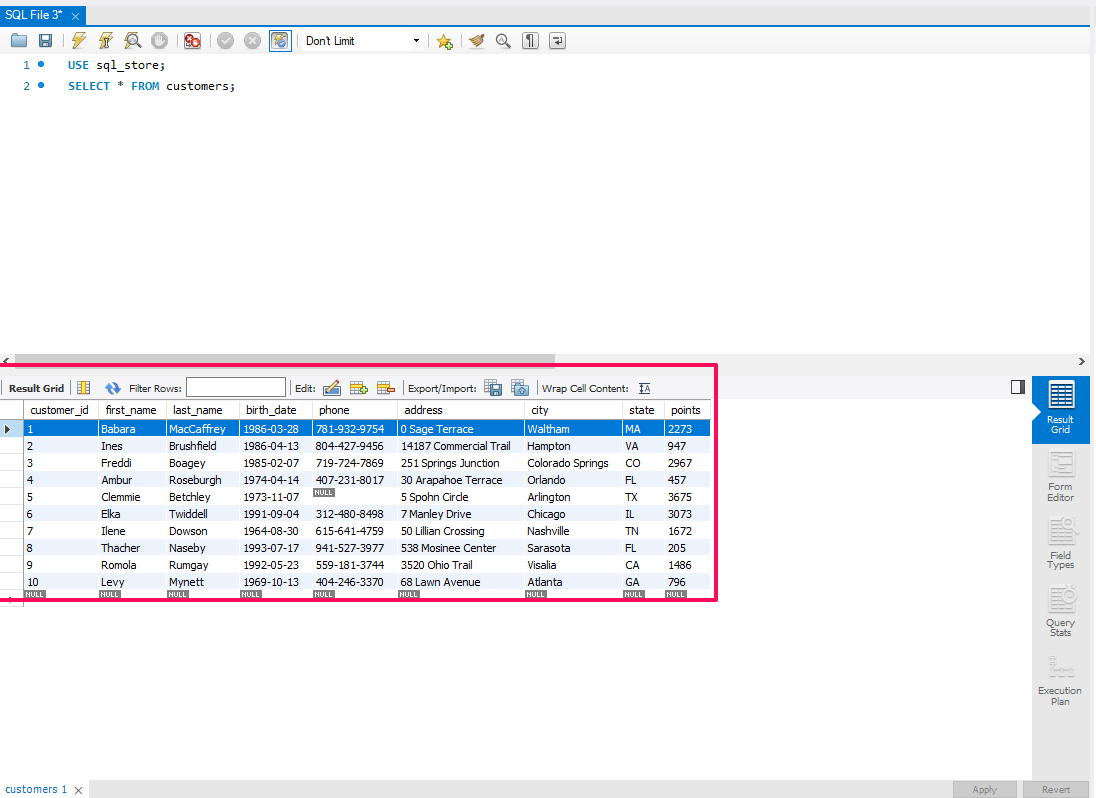
# QUERIES EXECUTION

## QUERY 1

1. Select all the data from the customers’ table contained in the store database:  
   **Query:** USE sql\_store;

SELECT \* FROM customers;

Result:



1. Add the following into the file Query 1

Order customers alphabetically

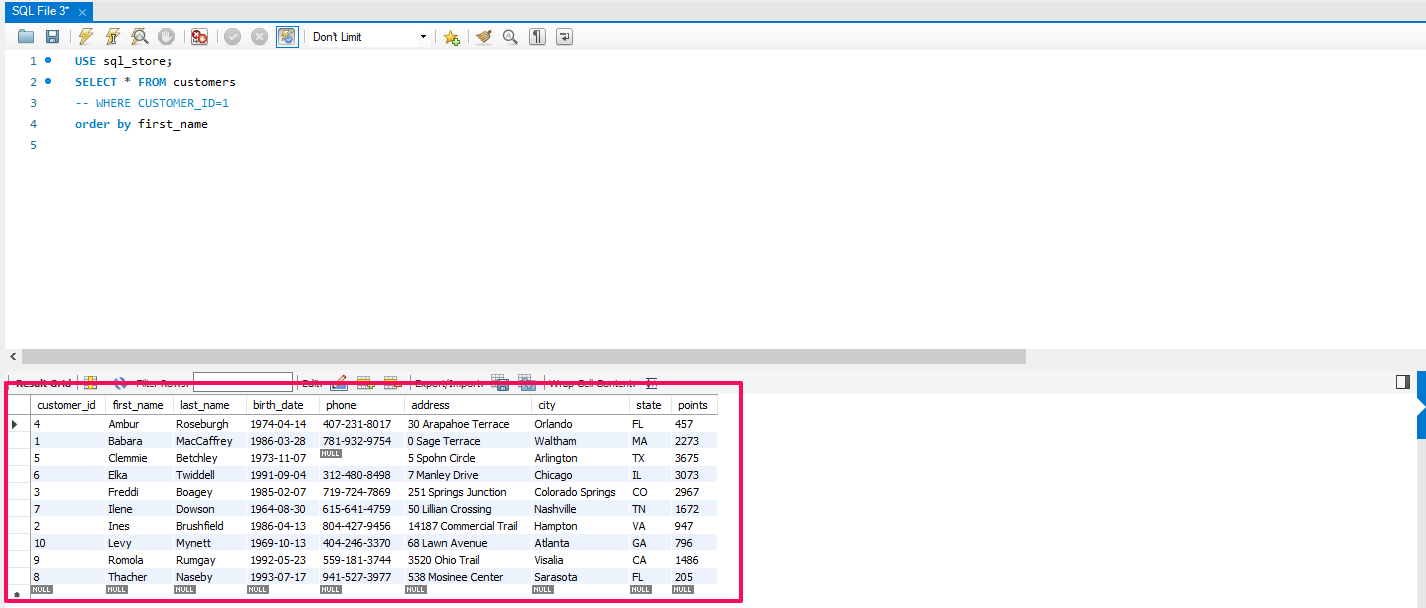
ORDER BY first\_name

**Query:** USE sql\_store;

SELECT \* FROM customers

ORDER BY first\_name;

Result:

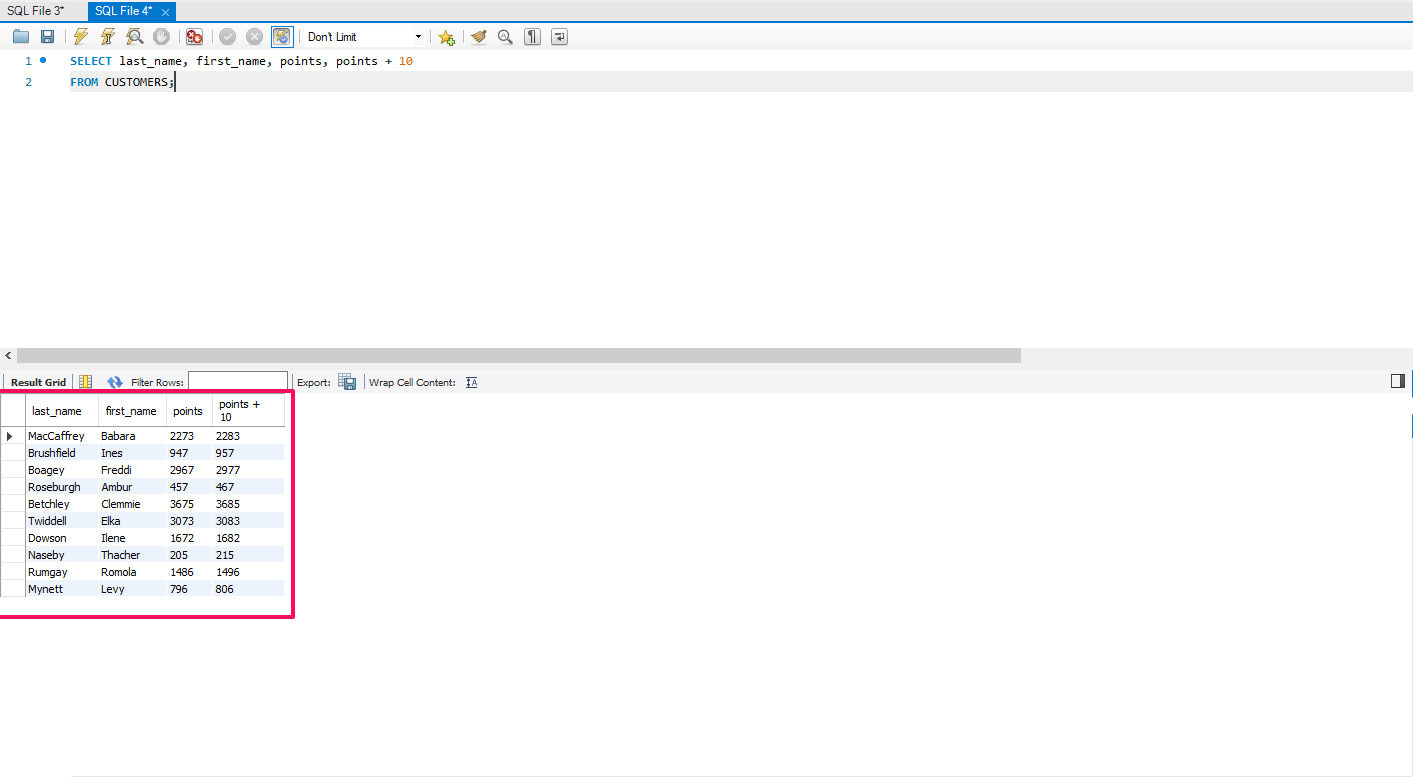


## QUERY 2

1. Extract last\_name, first\_name, points and compute points +10 column

**Query:** SELECT last\_name, first\_name, points, points + 10

FROM customers;



# TASK 1

**Task**: Using the Query 2 we created change the points to read times by 10 and plus 100 points.

**Solution:** calculates the adjusted points by first multiplying the points by 10 and then adding 100 to the result.

**Query:** SELECT last\_name, first\_name, points, points \* 10 + 100

FROM customers;

Result:

A screenshot of a computer

Description automatically generated

**Task:** Change the Query 2 code to create a discount factor so the table now shows a discount header and changing the (point + 10) \*100.

**Solution:** It calculates a new value based on the points column by first adding 10 and then multiplying the result by 100. The result is aliased as 'discount\_factor' in the query result.

**Query:** SELECT last\_name, first\_name, points, (points + 10) \* 100 as discount\_factor

FROM customers;

Result:

A screenshot of a computer

Description automatically generated

# TASK 2

**Task:** Write a SQL query to return all the products in our database in the result set. I want to make three new columns, name, unit price, and new column called new price, which is based on this expression, (unit price \* 1.1).

So, what you are doing is increasing the product price of each by 10%.

So, with the query we want to retrieve all the products showing the original unit price and increased price.

**Solution:** In this Query, the new\_price column is calculated by multiplying the unit\_price by 1.1, which increases the price.

**Query:** SELECT

name,

unit\_price,

unit\_price \* 1.1 AS new\_price

FROM products;

A screenshot of a computer

Description automatically generated

# TASK 3

**Task:** In this task create a new query to find all the customers with a birth date of > '1990-01-01'

**Solution:** Created a query which extracts data of customers which were born after 1st of Jan 1990

**Query:** SELECT \* FROM customers

WHERE birth\_date > '1990-01-01';

Result:

A screenshot of a computer

Description automatically generated

# Task4

**Task:** Select sql\_inventory.

Write a query to find out the name of the product with most amount in stock.

**Solution:** This query selects the name of the product with the highest quantity in stock from the Products table in sql\_inventory schema. It uses a subquery to find the maximum quantity in stock across all products and then selects the product(s) with that maximum quantity.

**Query:** SELECT name FROM products

WHERE

quantity\_in\_stock = (SELECT MAX (quantity\_in\_stock) FROM products);

Result:

A screenshot of a computer

Description automatically generated

# Task5

**Task:** Select sql\_inventory.

Write a query to find out the name of the most expensive product.

**Solution:** SELECT name, unit\_price : This selects the column name of the product and unit\_price from the table.

FROM products : This specifies the table products from which we are selecting data.

WHERE unit\_price = (SELECT MAX(unit\_price) FROM products) : This filters the rows where the unit\_price is equal to the maximum unit\_price found in the products table. This subquery (SELECT MAX (unit\_price) FROM products) finds the maximum amount in stock.

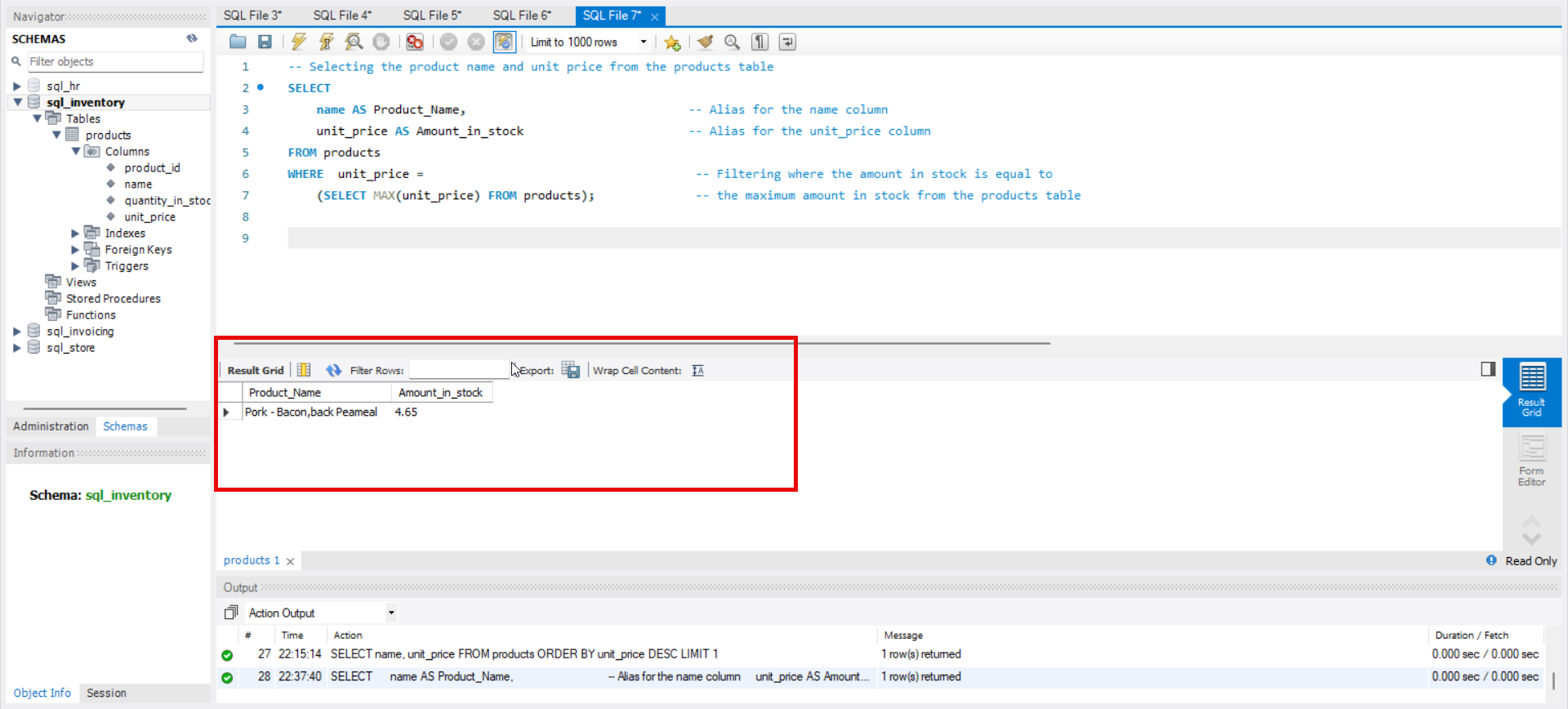
Finally, we're selecting only the name from the rows that meet this condition.

**Query:** SELECT name AS Product\_Name, unit\_price AS Amount\_in\_stock

FROM products

WHERE unit\_price = (SELECT MAX(unit\_price) FROM products);

Result:



# Task6

**Task:** Select sql\_store. Write a query to find out the first name, last name, address, and the birthdate of the oldest customer.

**Solution:** This query retrieves the first name, last name, address, and birthdate of the oldest customer from the sql\_store table. It uses a subquery to find the minimum birthdate among all customers and then selects the customer(s) with that minimum birthdate.

**Query:** SELECT first\_name,

last\_name,

address,

birth\_date

FROM customers

WHERE birth\_date =

(SELECT MIN(birth\_date) FROM customers);

A screenshot of a computer

Description automatically generated

# EER DIAGRAM

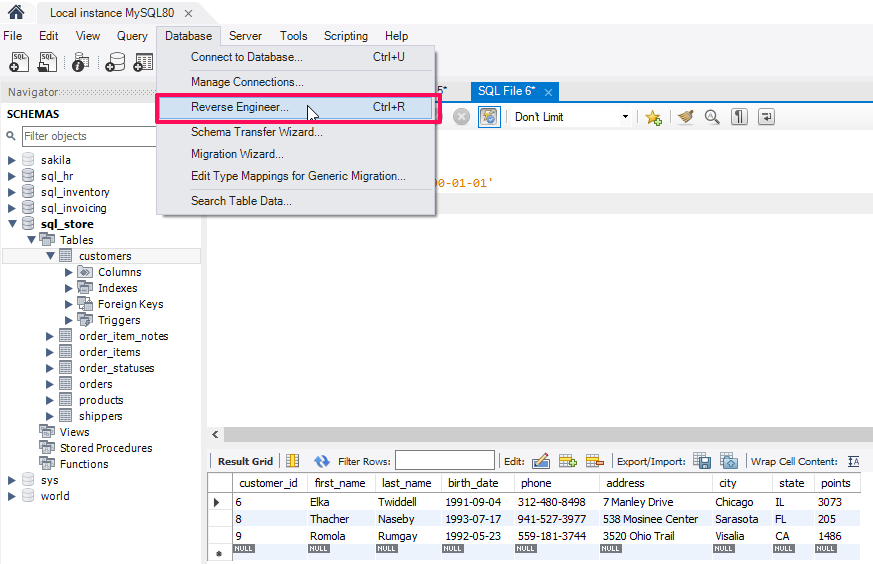
EER (Enhanced Entity-Relationship) diagrams are a type of data modeling tool used in software engineering to design and document complex databases. EER diagrams are an extension of the traditional Entity-Relationship (ER) model, which is used to define the relationships between entities (such as people, places, and things) in a database.

EER diagrams add additional features to the ER model, such as subtypes and supertypes, to better represent real-world scenarios where entities may have multiple subtypes or belong to multiple supertypes. EER diagrams also allow for the representation of relationships between relationships, which can be useful in more complex database designs.

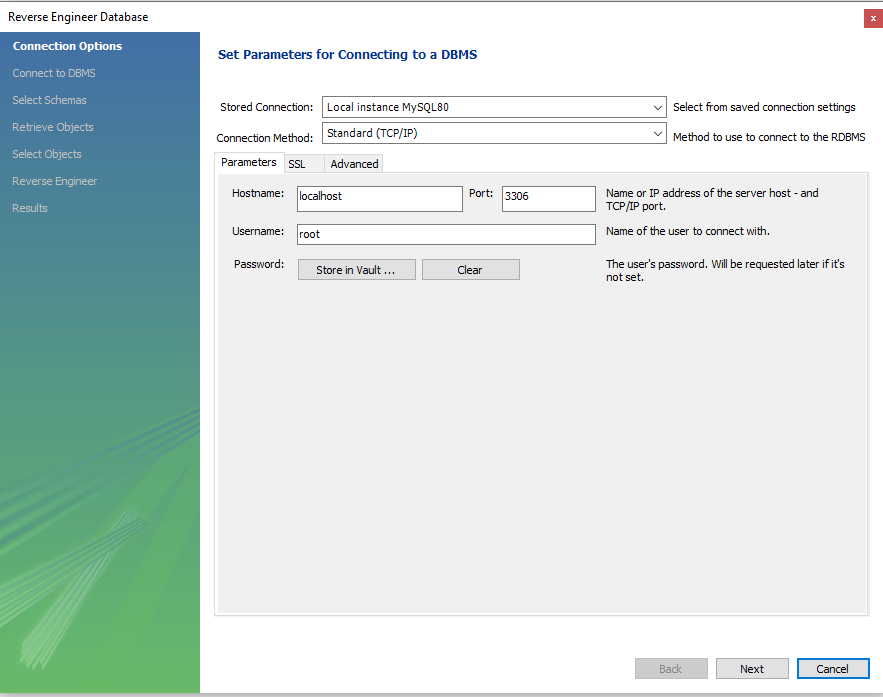
EER diagrams typically consist of boxes representing entities, with lines connecting the boxes to represent relationships between entities. The boxes may contain attributes of the entities, such as names, IDs, or other descriptive information. The lines connecting the boxes may include symbols to indicate the nature of the relationship between entities, such as one-to-many or many-to-many relationships.

Overall, EER diagrams are an important tool for database designers and developers to visualize and plan complex databases, and to communicate their designs with stakeholders and other team members.

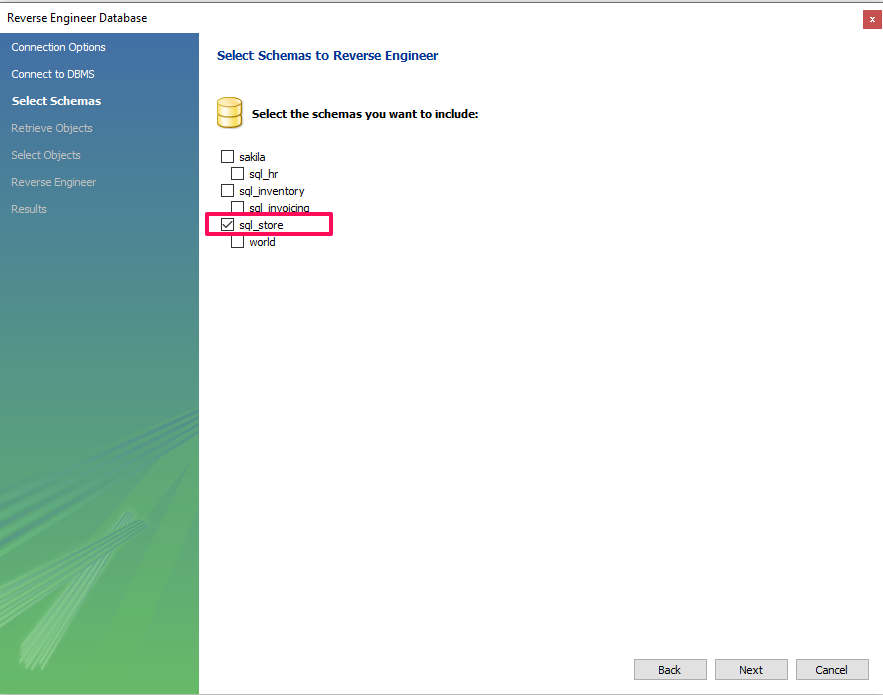
1. Create EER Diagram – navigate to Database tab, choose Reverse Engineer option or use ctrl + R shortcut on Windows devices or cmd + R on Mac OS.



1. Navigate through setup wizard



1. We want to include the store schema



1. EER Diagram Result:

