



ASSIGNMENT-2

My SQL Database.

ABSTRACT

Presentation about processing the data using My SQL relational database management with My SQL Workbench.

Parvathavardini Murugan

DATA GLA 1

Contents

INTRODUCTION TO MY SQL	2
What Is SQL?	2
Some of SQL Commands.....	3
Types of SQL Languages.....	3
INTRODUCTION TO MY SQL WORKBENCH	4
Database Migration	4
1. To Execute Query 1	7
2. In the Query 1 shows the customer names in alphabetical order.....	7
3. Query 2 shows a points + 10 customers Data.....	8
Task 1	8
Task 2	9
4. Below data shows the 10% increased price and original price for customers.	9
Task 3	9
EER Diagram.....	10
EER Diagram for this Database	12
Conclusion.....	12

INTRODUCTION TO MY SQL

MySQL is an open-source relational database management system. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network.

MySQL allows you to handle, store, modify and delete data and store data in an organized way.

MySQL is compatible with all primary OS (operating systems). It is a core component of a widely popular open-source web application software stack called LAMP, which stands for Linux, Apache, MySQL, PHP/Perl/Python.

What Is SQL?

SQL is short for Structured Query Language. As per the ANSI (American National Standards Institute) guidelines, SQL is the standard language to maintain and manage a [database](#).

- SQL is a query programming language that manages RDBMS.
- SQL is primarily used to query and operate database systems.
- SQL follows a simple standard format without many or regular updates.
- SQL supports only a single storage engine.

```
mysql> CREATE DATABASE test;
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> USE test;
Database changed
mysql> DROP TABLE IF EXISTS `t_mqtt_msg`;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> CREATE TABLE `t_mqtt_msg` (
  -> `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  -> `msgid` varchar(64) DEFAULT NULL,
  -> `topic` varchar(255) NOT NULL,
  -> `qos` tinyint(1) NOT NULL DEFAULT '0',
  -> `payload` blob,
  -> `arrived` datetime NOT NULL,
  -> PRIMARY KEY (`id`),
  -> INDEX topic_index(`id`, `topic`)
  -> ) ENGINE=InnoDB DEFAULT CHARSET=utf8MB4;
Query OK, 0 rows affected (0.07 sec)

mysql> describe t_mqtt_msg;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(11) unsigned | NO   | PRI | NULL    | auto_increment |
| msgid | varchar(64)      | YES  |     | NULL    |                |
| topic | varchar(255)     | NO   |     | NULL    |                |
| qos   | tinyint(1)       | NO   |     | 0       |                |
| payload | blob             | YES  |     | NULL    |                |
| arrived | datetime         | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)
```

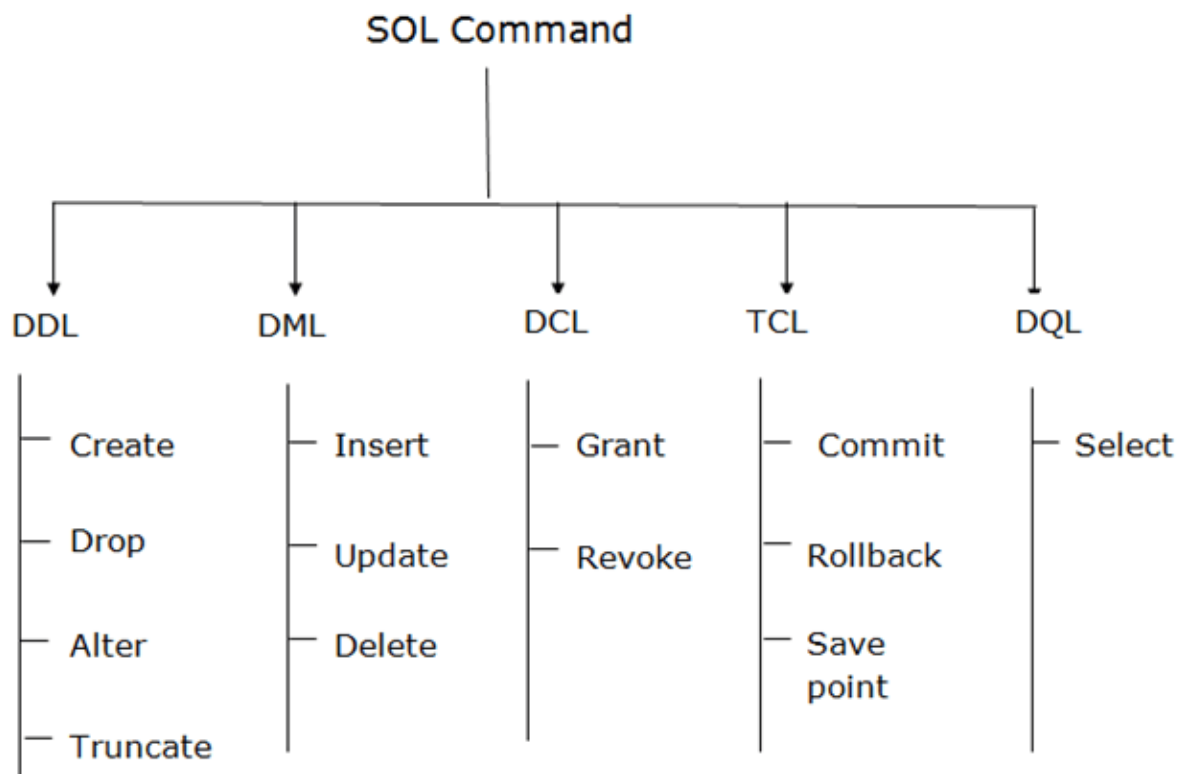
Some of SQL Commands

- **SELECT** - extracts data from a database
- **UPDATE** - updates data in a database
- **DELETE** - deletes data from a database
- **INSERT INTO** - inserts new data into a database
- **CREATE DATABASE** - creates a new database
- **ALTER DATABASE** - modifies a database
- **CREATE TABLE** - creates a new table
- **ALTER TABLE** - modifies a table
- **DROP TABLE** - deletes a table
- **CREATE INDEX** - creates an index (search key)
- **DROP INDEX** - deletes an index

Types of SQL Languages

There are five types of SQL Languages: DDL, DML, DCL, TCL, and DQL.

1. Data Definition Language (DDL),
2. Data Manipulation Language (DML),
3. Data Control Language,
4. Transaction Control Language,
5. Data Query Language.



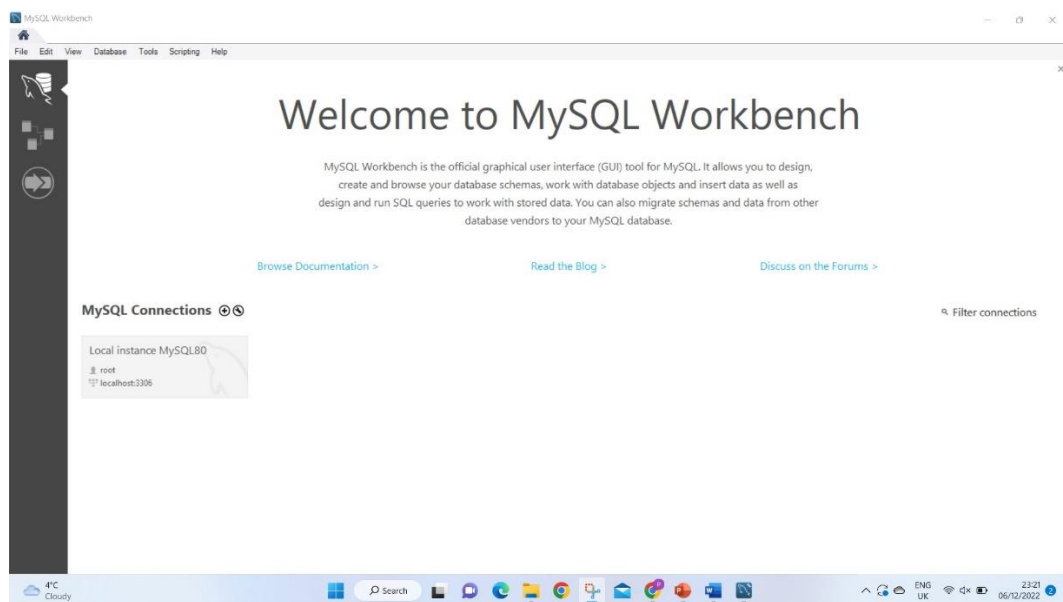
INTRODUCTION TO MY SQL WORKBENCH

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modelling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. MySQL Workbench is available on Windows, Linux and Mac OS X.

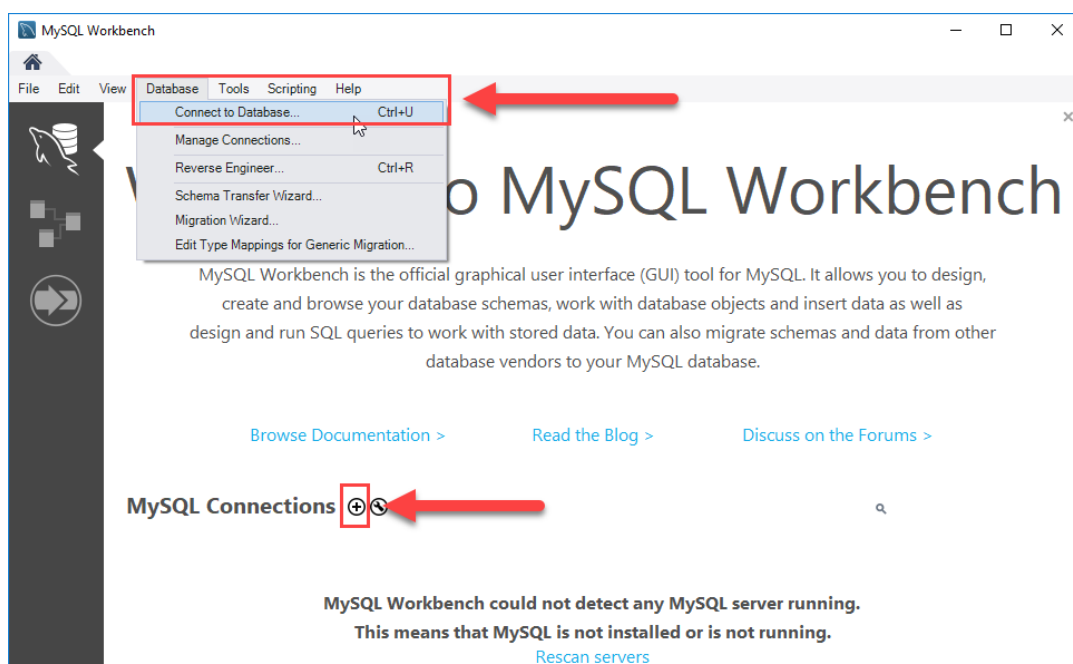
Database Migration

MySQL Workbench now provides a complete, easy to use solution for migrating Microsoft SQL Server, Microsoft Access, Sybase ASE, PostgreSQL, and other RDBMS tables, objects and data to MySQL.

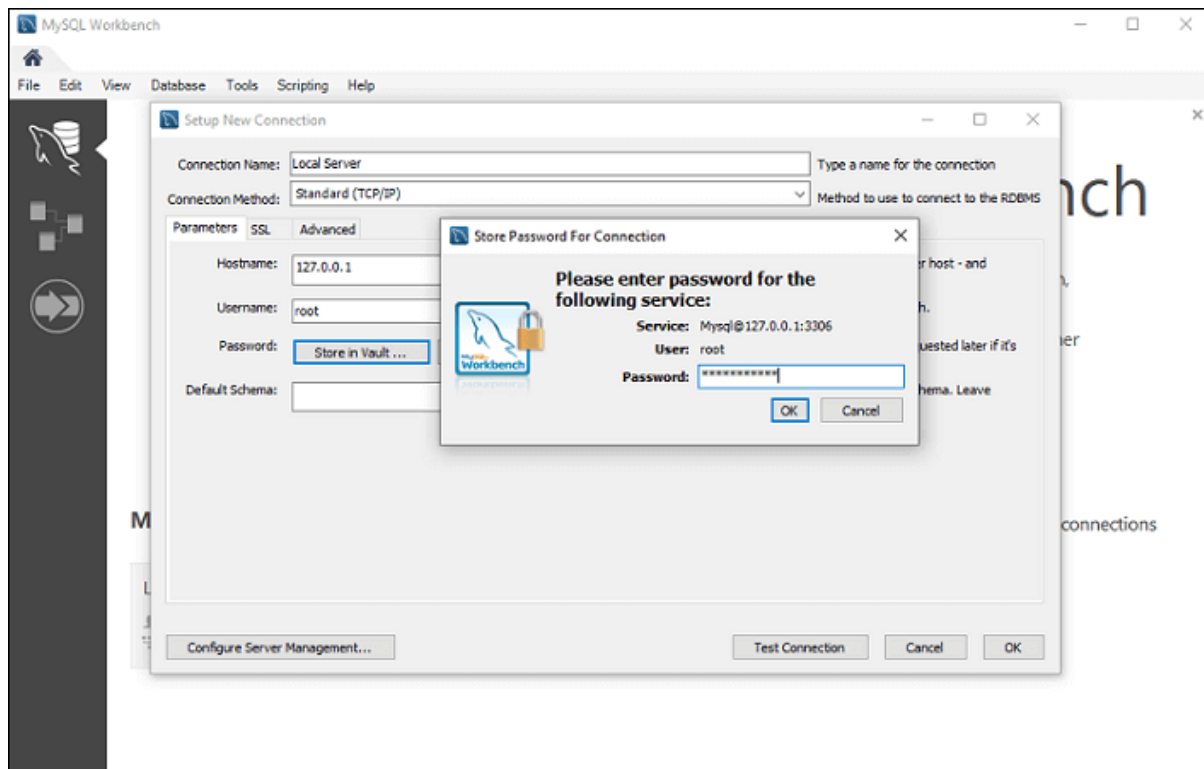
1. Homepage of MYSQL Workbench



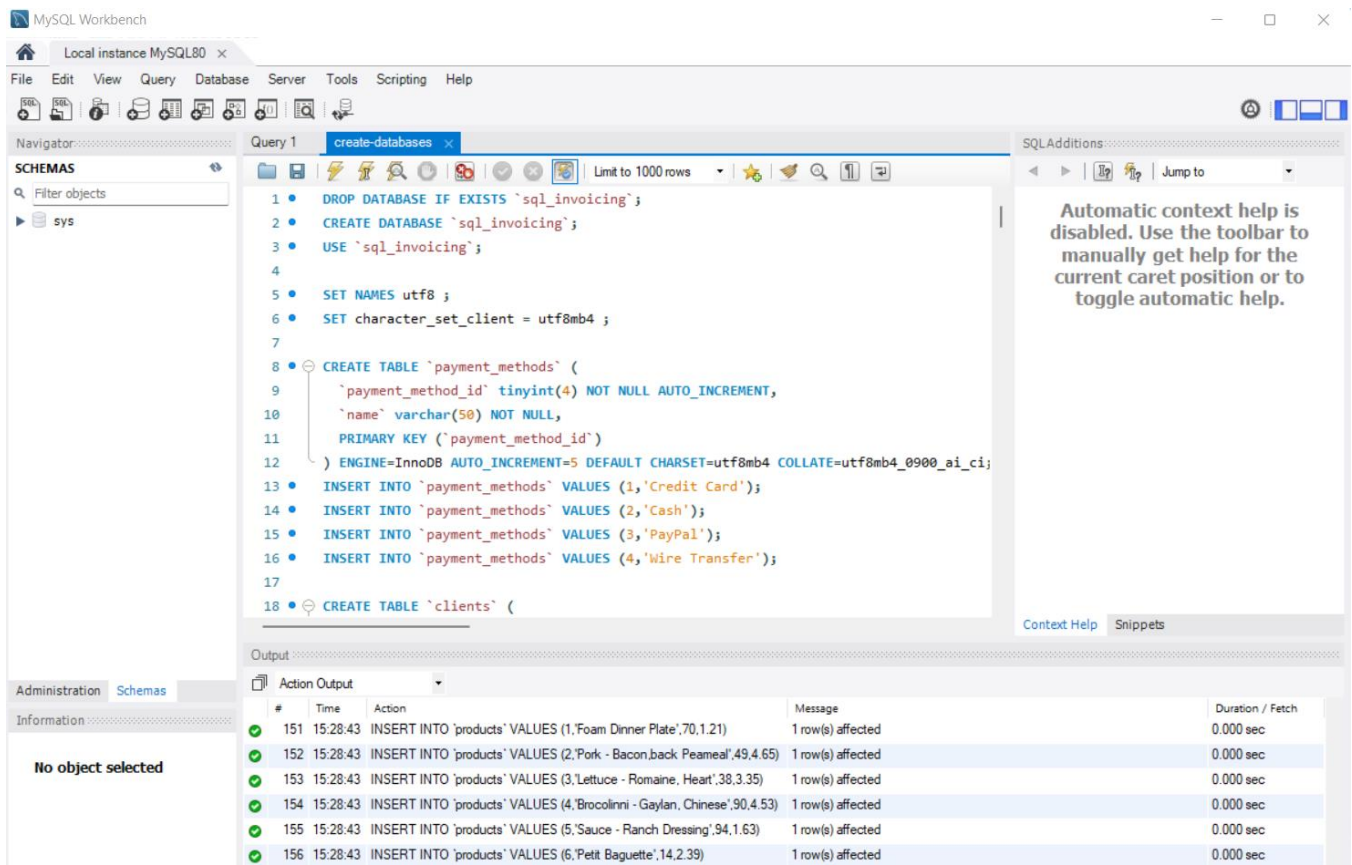
2. Connecting MySQL Workbench with Local Instance



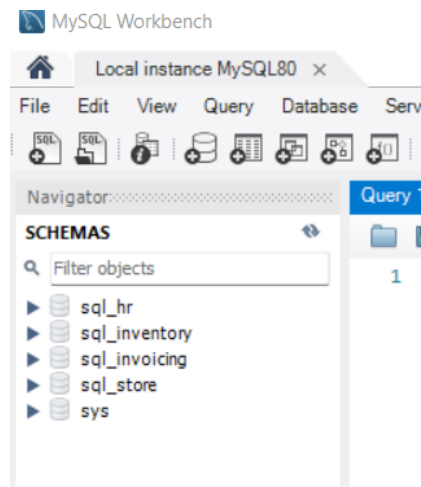
3. Setup for My SQL server to Local Instance



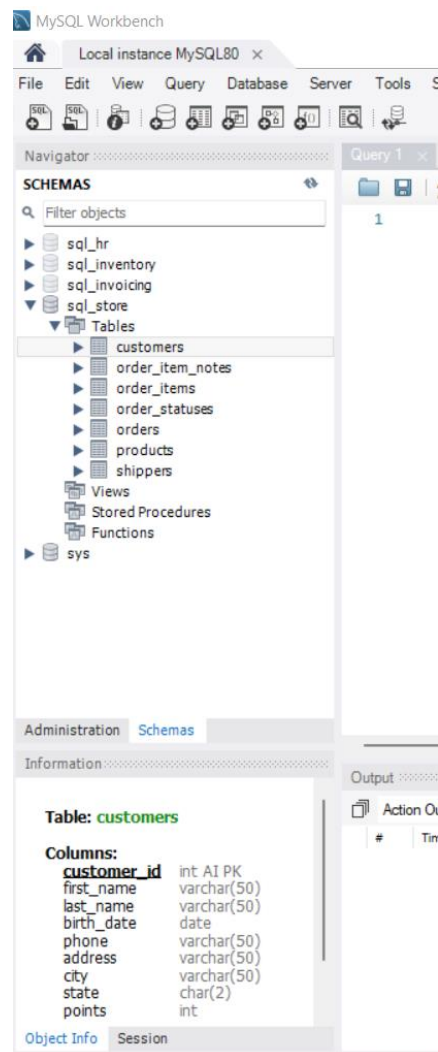
4. Open SQL Script file in a Query tab and execute



5. Schemas to View data



6. To view customer data go to sql_store. Below Schemas it shows table details of customers



1. To Execute Query 1

Query 1 shows the results of customers.

The screenshot shows the MySQL Workbench interface. The Query Editor displays the following SQL query:

```
1 USE sql_store;
2
3 SELECT *
4 FROM CUSTOMERS
```

The Results Grid shows the following data:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
5	Clemmie	Betchley	1973-11-07		5 Spohn Circle	Arlington	TX	3675
6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
7	Ilene	Dowison	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
8	Thatcher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lavin Avenue	Atlanta	GA	796

The left sidebar shows the database schema with the 'customers' table selected. The bottom status bar shows the system clock as 16:59 on 18/11/2022.

2. In the Query 1 shows the customer names in alphabetical order

The screenshot shows the MySQL Workbench interface. The Query Editor displays the following SQL query:

```
1 USE sql_store;
2
3 SELECT *
4 FROM CUSTOMERS
5 -- WHERE CUSTOMER_ID=1
6 order by first_name
```

The Results Grid shows the following data, sorted by first_name:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	MA	2273
5	Clemmie	Betchley	1973-11-07		5 Spohn Circle	Arlington	TX	3675
6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
7	Ilene	Dowison	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
10	Levy	Mynett	1969-10-13	404-246-3370	68 Lavin Avenue	Atlanta	GA	796
9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
8	Thatcher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205

The left sidebar shows the database schema with the 'customers' table selected. The bottom status bar shows the system clock as 17:07 on 18/11/2022.

3. Query 2 shows a points + 10 customers Data

The screenshot shows the MySQL Workbench interface. The 'Query 1' window displays the following SQL query:

```
1 • SELECT last_name, first_name, points, points +10
2 FROM CUSTOMERS
```

The 'Result Grid' shows the following data:

last_name	first_name	points	points +10
MacCaffrey	Babara	2273	2283
Brushfield	Ines	947	957
Boagey	Freddi	2967	2977
Roseburgh	Ambur	457	467
Betchley	Clemmie	3675	3685
Twiddell	Elka	3073	3083
Dowson	Ilene	1672	1682
Naseby	Thacher	205	215
Rumgay	Romola	1486	1496
Mynett	Levy	796	806

The 'Table: customers' structure is shown in the bottom left:

Columns:

- customer_id int AI PK
- first_name varchar(50)
- last_name varchar(50)
- birth_date date
- phone varchar(50)
- address varchar(50)
- city varchar(50)
- state char(2)
- points int

The 'Output' window shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	16:58:48	USE sql_store	0 row(s) affected	0.000 sec
2	16:58:48	SELECT * FROM CUSTOMERS LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
3	17:06:50	USE sql_store	0 row(s) affected	0.000 sec
4	17:06:50	SELECT * FROM CUSTOMERS - WHERE CUSTOMER_ID=1 order by first_name LIMIT 0, 1000	10 row(s) returned	0.016 sec / 0.000 sec
5	17:37:08	SELECT last_name, first_name, points, points +10 FROM CUSTOMERS LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

Task 1

Using the Query 2 shows a discount header and changing the (point + 10) *100

The screenshot shows the MySQL Workbench interface. The 'Query 2' window displays the following SQL query:

```
1 • USE sql_store;
2
3 • SELECT last_name, first_name, points, points*10, points+10
4
5
```

The 'Result Grid' shows the following data:

last_name	first_name	points	points*10	points+100
MacCaffrey	Babara	2273	22730	2373
Brushfield	Ines	947	9470	1047
Boagey	Freddi	2967	29670	3067
Roseburgh	Ambur	457	4570	557
Betchley	Clemmie	3675	36750	3775
Twiddell	Elka	3073	30730	3173
Dowson	Ilene	1672	16720	1772
Naseby	Thacher	205	2050	305
Rumgay	Romola	1486	14860	1586

Task 2

4. Below data shows the 10% increased price and original price for customers.

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT NAME, unit_price, unit_price* 1.1 AS new_price
2 FROM products
```

The 'Result Grid' shows the following data:

NAME	unit_price	new_price
Foam Dinner Plate	1.21	1.331
Pork - Bacon,back Peameal	4.65	5.115
Lettuce - Romaine, Heart	3.35	3.685
Broccoli - Gaylan, Chinese	4.53	4.983
Sauce - Ranch Dressing	1.63	1.793
Petit Baguette	2.39	2.629
Sweet Pea Sprouts	3.29	3.619
Island Oasis - Raspberry	0.74	0.814
Longan	2.26	2.486
Broom - Push	1.09	1.199

The 'Table: customers' information is also visible on the left:

Table: customers

Columns:

- customer_id: int, A1 PK
- first_name: varchar(50)
- last_name: varchar(50)
- birth_date: date
- phone: varchar(50)
- address: varchar(50)
- city: varchar(50)
- state: char(2)
- points: int

The 'Output' tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
2	16:58:48	SELECT * FROM CUSTOMERS LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
3	17:06:50	USE sql_store	0 row(s) affected	0.000 sec
4	17:06:50	SELECT * FROM CUSTOMERS -- WHERE CUSTOMER_ID=1 order by first_name LIMIT 0, 1000	10 row(s) returned	0.016 sec / 0.000 sec
5	17:37:08	SELECT last_name, first_name, points, points +10 FROM CUSTOMERS LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
6	17:50:28	SELECT NAME, unit_price, unit_price* 1.1 AS new_price FROM products LIMIT 0, 1000	10 row(s) returned	0.016 sec / 0.000 sec

Task 3

5. In this task shows the customer data who are born after "1990-01-01"

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```
1 USE sql_store;
2 SELECT * FROM customers where birth_date > '1990-01-01';
3
```

The 'Result Grid' shows the following data:

customer_id	first_name	last_name	birth_date	phone	address	city	state	points
6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	30
8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	20
9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	10
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

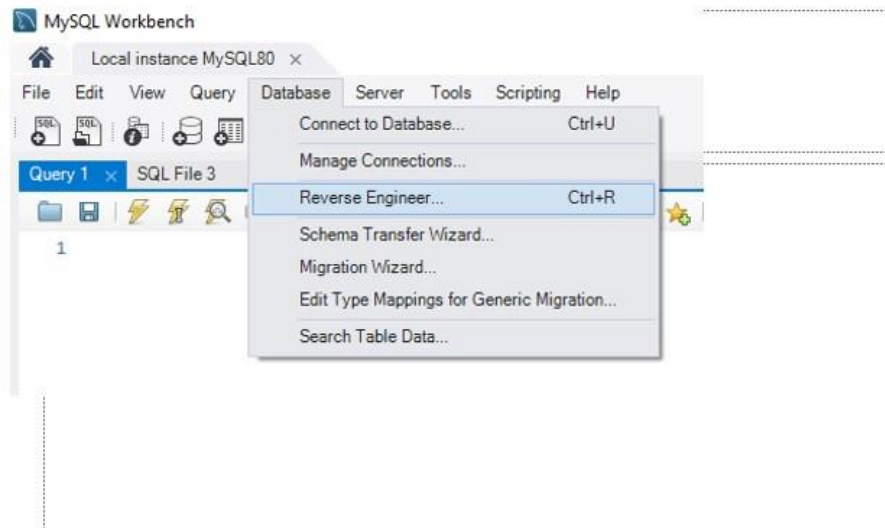
The 'Output' tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	17:50:28	USE sql_store	0 row(s) affected	0.000 sec
2	17:50:28	SELECT * FROM customers where birth_date > '1990-01-01'	10 row(s) returned	0.016 sec / 0.000 sec

EER Diagram

Enhanced Entity-Relationship (EER) diagrams are an essential part of the modeling interface in MySQL Workbench. EER diagrams provide a visual representation of the relationships among the tables in your model. Revisions made with the Model Editor are shown in the associated diagram.

1. To create EER Diagram go to Database tab in that view Reverse Engineer.



2. It Shows the Home page

Reverse Engineer Database

Connection Options

- Connect to DBMS
- Select Schemas
- Retrieve Objects
- Select Objects
- Reverse Engineer
- Results

Set Parameters for Connecting to a DBMS

Stored Connection: Select from saved connection settings

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: 127.0.0.1 Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: root Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.

Back Next Cancel

3. The following wizard setup shows the summary of Tables of schema “sql_store”

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

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:

☐ sql_hr

☐ sql_inventory

☐ sql_invoicing

☐ sql_store

Back

Next

Cancel

Reverse Engineer Database

Connection Options

Connect to DBMS

Select Schemas

Retrieve Objects

Select Objects

Reverse Engineer

Results

Retrieve and Reverse Engineer Schema Objects

The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.

☒ Retrieve Objects from Selected Schemas

☒ Check Results

Retrieval Completed Successfully
Finished.

Show Logs

Back

Next

Cancel

Reverse Engineer Database

Connection Options

Connect to DBMS

Select Schemas

Retrieve Objects

Select Objects

Reverse Engineer

Results

Select Objects to Reverse Engineer

☒ Import MySQL Table Objects

7 Total Objects, 7 Selected

Show Filter

☒ Place imported objects on a diagram

Back

Execute >

Cancel

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:
- 7 tables from schema 'sql_store'

Back

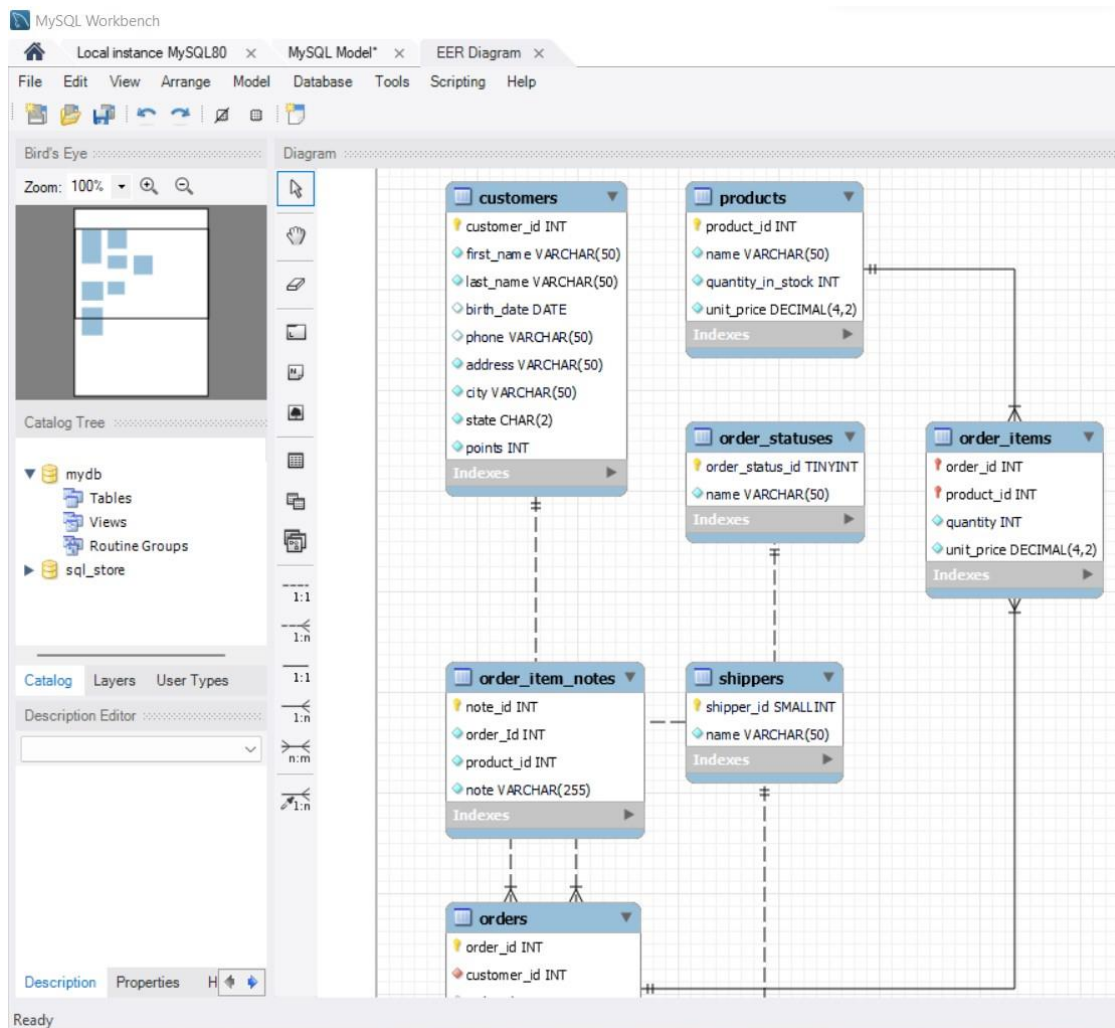
Finish

Cancel

EER Diagram for this Database

EER Diagrams basically help in creating and maintaining excellent databases with the help of smart and efficient techniques. In addition to this, it is a visual representation of the plan or the overall outlook of the database you intend to create.

In this database, the customers are an entity with attributes like Customer_id, first_name, last_name, birth_date, phone, city, address etc. The order_item can be another entity with attributes like note_id, order, product etc. and there is a relationship between them.



Conclusion

The **MySQL** server stores data and response to requests from MySQL clients. The client is always packaged with the server, but it can be used as a stand-alone application to communicate with remote databases. It is also used in web databases and data warehousing.

Conceptually **ER Model** is very easy to build. If we know the relationship between the attributes and the entities we can easily build the ER Diagram for the model. Effective Communication Tool: This model is used widely by the database designers for communicating their ideas