

SQL



**A
PROJECT REPORT
On**

**“RETAILERS DETAILS” AND THEIR “SHOP NAMES INFO ”
ANALYSIS USING A SQL**

A report submitted to

Seven Mentors Institute, Pune 2023



Date :

20 April, 2023

Submitted by :

Pratik Kishor Pol

DECLARATION OF CERTIFICATE

Sub : SQL

DATE: 20 April, 2023

I hereby declare that, The project entitled is an outcome of my own efforts Under the guidance of **Mr. Sagar Gade Sir**. The project is submitted to the **Seven Mentor Institute Pune**. For the partial fulfilment of the “**Structure Query Language Course**” 2023.

I also declare that this project report has not been previously submitted to any other institute.

PLACE: Pune

Pratik Kishor Pol

INDEX

| <u>SR. NO.</u> | <u>Particulars</u> | <u>Page No.</u> |
|---------------------------|-----------------------------|------------------------|
| 1. | Acknowledgement | |
| 2. | Introduction | |
| 3. | Technology/Tools | |
| 4. | Diagram | |
| 5. | Explanation | |
| 6. | Input/Output Screens | |
| 7. | Problem Statement | |
| 8. | Conclusion | |
| 9. | Future Scope | |

| | | |
|------------|-------------------|--|
| 10. | References | |
|------------|-------------------|--|

ACKNOWLEDGEMENT

I'm sincerely thankful to our Training Institute 'Seven Mentor' for their valuable timely help.

I would like to take this opportunity to express our deep since of humbleness as well as our sincere, earful and proud gratitude to our project guide Mr. Sagar Gade Sir.

A special thanks to Mrs. Arti Rahulkar for being so helpful and kind during the course completion.

INTRODUCTION

Quotation provided by different suppliers might include different contents, items, formats, etc. To give a brief introduction of the quotation content, we list the items from commonly-used quotation formats and explain the items you need in different stages.

Items in Quotation

- Retailer info: First name, Last name, age , location , product**

- **Selling price, Profit and Loss.**
- **Project name: Retailer and Their Shop Info.**

TECHNOLOGY USED IN PROJECT

The latest support for working with MySQL is version number v5.8. It contains many essential changes, including new features added and removed, fixed bugs and security issues, etc. This version contains the release history from MySQL 8.0 to MySQL 8.0.21. It is available from April 2018 and ends the support in April 2026.

•

SQL is designed for a specific purpose: to query data contained in a relational database. SQL is a set-based, declarative programming language, not an imperative programming language like C or BASIC. However, extensions to Standard SQL add procedural programming language functionality, such as control-of-flow constructs.

- **SQL stands for Structured Query Language**
- **SQL lets you access and manipulate databases**
- **SQL became a standard of the American National Standards Institute**
- **(ANSI) in 1986, and of the International Organization for**
- **Standardization (ISO) in 1987**
- **SQL can execute queries against a database**
- **SQL can retrieve data from a database**
- **SQL can insert records in a database**
- **SQL can update records in a database**
- **SQL can delete records from a database**

- ❑ **SQL can create new databases**
- ❑ **SQL can create new tables in a database**
- ❑ **SQL can create stored procedures in a database**
- ❑ **SQL can create views in a database**
- ❑ **SQL can set permissions on tables, procedures, and views**
- ❑ **An RDBMS database program (i.e. MS Access, SQL Server, MySQL)**
- ❑ **To use a server-side scripting language, like PHP or ASP**
- ❑ **To use SQL to get the data you want**

✚ Some of The Most Important 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)

EXPLANATION

Making of the Project

SAMPLE CODES FOR RETAILER DETAILS AND THIER SHOP INFO:

```
CREATE DATABASE shop_info;
```

```
USE Shop_info;
```

```
CREATE TABLE Retailers
```

```
(
```

```
create table retailers (
```

```
ID int primary key ,
```

```
f_name varchar(50) not null,
```

```
l_name varchar(50) not null,
```

```
age int ,
```

```
location varchar(30) not null,
```

```
products varchar(30) not null,
```

```
selling_price varchar(50) not null,
```

```
profit int,
```

```
loss int
```

```
);
```

```
SHOW TABLES;
```

DESC Retailers;

■ **SECOND TABLE**

Create Table shop_name(

ID int primary key,

Seller_ID int,

Shop_Names Varchar(50),

dep_location varchar(50));Quotation_Amount decimal(10,2),

Customer_No int,

Quotation_Date date,

PRIMARY KEY (Quotation_No)

);

SHOW

■ **INSERTED VALUES INTO FIRST TABLE**

INSERT INTO Retailers;

insert into retailers values(1, 'Mahesh','Lande',22,'pune','RealMe',70000,50000,1000);

insert into retailers values(2,'Aditi','Jadhav',22,'Beed','RealMe',45000,3000,0);

insert into retailers values(3,'Shraddha','Kale',21,'nagar','oppo',30000,2500,2000);

insert into retailers values(4,'Swin','Willa',43,'Pune','Asus TUF',75000,20000,0);

insert into retailers values(5,'Iris','Devine',33,'Mumbai','Dell Vostro',25000,1500,0);

insert into retailers values(6,'Peter','Kewin',40,'Nashik','HP 14s',40000,0,0);

insert into retailers values(7,'Joseph','Alzery',26,'Pune','Asus Vivobook',44000,9000,0);

insert into retailers values(8,'John','Bottom',26,'Mumbai','HP Pavilion', 60000,4000,2000);

insert into retailers values(9,'Stephen','Hwkward',35,'Nashik','Asus Vivobook',50000,15000,500);

insert into retailers values(10, 'Suzi','Yann',45,'pune','Asus TUF',60000, 5000,0);SELECT*FROM Customer;

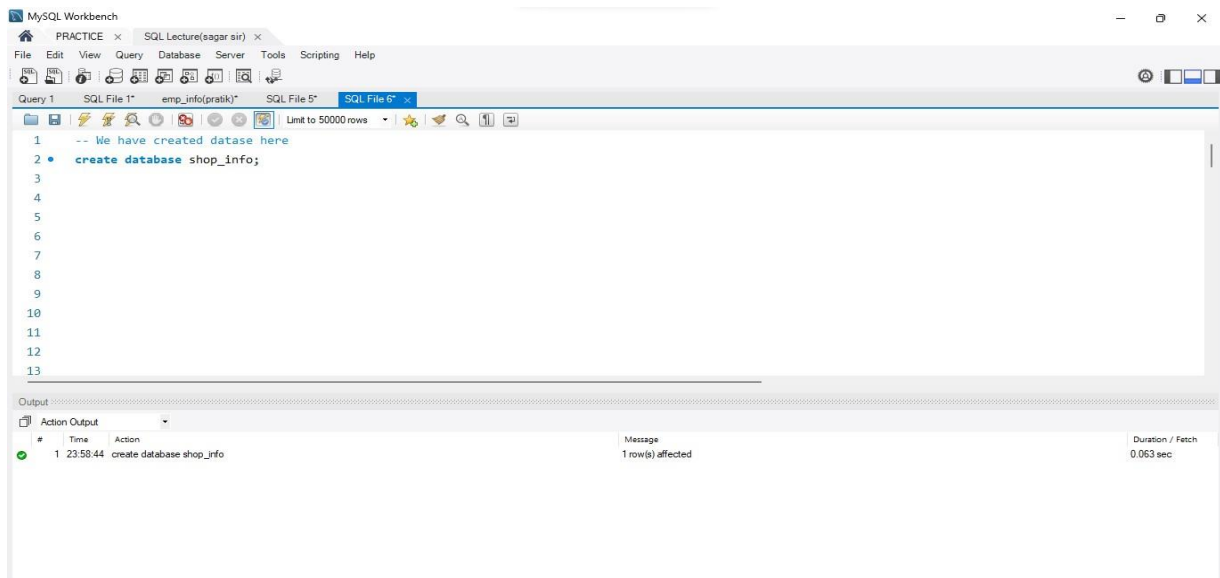
■ **INSERTED VALUES IN SECOND TABLE :**

```
insert into shop_name values (1,101,'Mahii Computers','Pune');  
insert into shop_name values (2,102,'Sai Computers','Beed');  
insert into shop_name values (3,103,'Shraddha Computers','Beed');  
insert into shop_name values (4,104,'Sairaj Computers','Mumbai');  
insert into shop_name values (5,105,'Abhi Computers', 'Mumbai');  
insert into shop_name values (6,106,'Ekta Computers', 'Delhi');  
insert into shop_name values (7,107, 'Sarang Computers','Nashik');  
insert into shop_name values (8,108,'Rupali Computers','Delhi');  
insert into shop_name values (9,109,'Mahesh Computers','Pune');  
insert into shop_name values (10,110,'Shiv Computers','Delhi');SELECT *FROM Quotation;
```

Input/Output Screens:

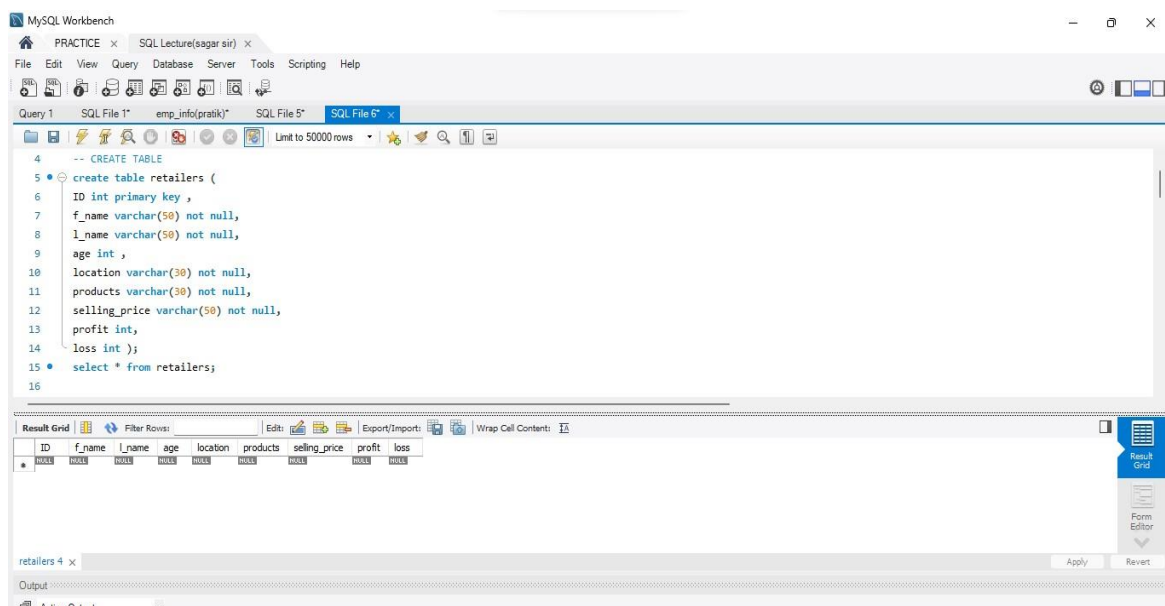
- We have created a DATABASE here,

```
CREATE DATABASE shop_info;
```



CREATED TABLE < Table_name >

```
CREATE TABLE retailers;
```



■ INSERT VALUES IN TABLE :

INSERT INTO Retailer values (...);

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```
-- INSERT VALUES IN 'RETAILER' TABLE.  
30 • insert into retailers values(1, 'Mahesh', 'Lande', 22, 'pune', 'RealMe', 70000, 50000, 1000);  
31 • insert into retailers values(2, 'Aditi', 'Jadhav', 22, 'Beed', 'RealMe', 45000, 3000, 0);  
32 • insert into retailers values(3, 'Shraddha', 'Kale', 21, 'nagar', 'oppo', 30000, 2500, 2000);  
33 • insert into retailers values(4, 'Swin', 'Willa', 43, 'Pune', 'Asus TUF', 75000, 20000, 0);  
34 • insert into retailers values(5, 'Iris', 'Devine', 33, 'Mumbai', 'Dell Vostro', 25000, 1500, 0);  
35 • insert into retailers values(6, 'Peter', 'Kewin', 40, 'Nashik', 'HP 15s', 40000, 0, 0);  
36 • insert into retailers values(7, 'Joseph', 'Alzery', 26, 'Pune', 'Asus Vivobook', 44000, 9000, 0);  
37 • insert into retailers values(8, 'John', 'Bottom', 26, 'Mumbai', 'HP Pavilion', 60000, 4000, 2000);  
38 • insert into retailers values(9, 'Stephen', 'Hkward', 35, 'Nashik', 'Asus Vivobook', 50000, 15000, 500);  
39 • insert into retailers values(10, 'Suzi', 'Yann', 45, 'pune', 'Asus TUF', 60000, 5000, 0);  
40 • select * from retailers;
```

The Result Grid shows the data inserted into the 'retailers' table:

| ID | f_name | l_name | age | location | products | selling_price | profit | loss |
|----|----------|--------|-----|----------|---------------|---------------|--------|------|
| 1 | Mahesh | Lande | 22 | pune | RealMe | 70000 | 50000 | 1000 |
| 2 | Aditi | Jadhav | 22 | Beed | RealMe | 45000 | 3000 | 0 |
| 3 | Shraddha | Kale | 21 | nagar | oppo | 30000 | 2500 | 2000 |
| 4 | Swin | Willa | 43 | Pune | Asus TUF | 75000 | 20000 | 0 |
| 5 | Iris | Devine | 33 | Mumbai | Dell Vostro | 25000 | 1500 | 0 |
| 6 | Peter | Kewin | 40 | Nashik | HP 15s | 40000 | 0 | 0 |
| 7 | Joseph | Alzery | 26 | Pune | Asus Vivobook | 44000 | 9000 | 0 |
| 8 | John | Bottom | 26 | Mumbai | HP Pavilion | 60000 | 4000 | 2000 |
| 9 | Stephen | Hkward | 35 | Nashik | Asus Vivobook | 50000 | 15000 | 500 |
| 10 | Suzi | Yann | 45 | pune | Asus TUF | 60000 | 5000 | 0 |

TO CREATE SECOND TABLE :

CREATE TABLE shop_info

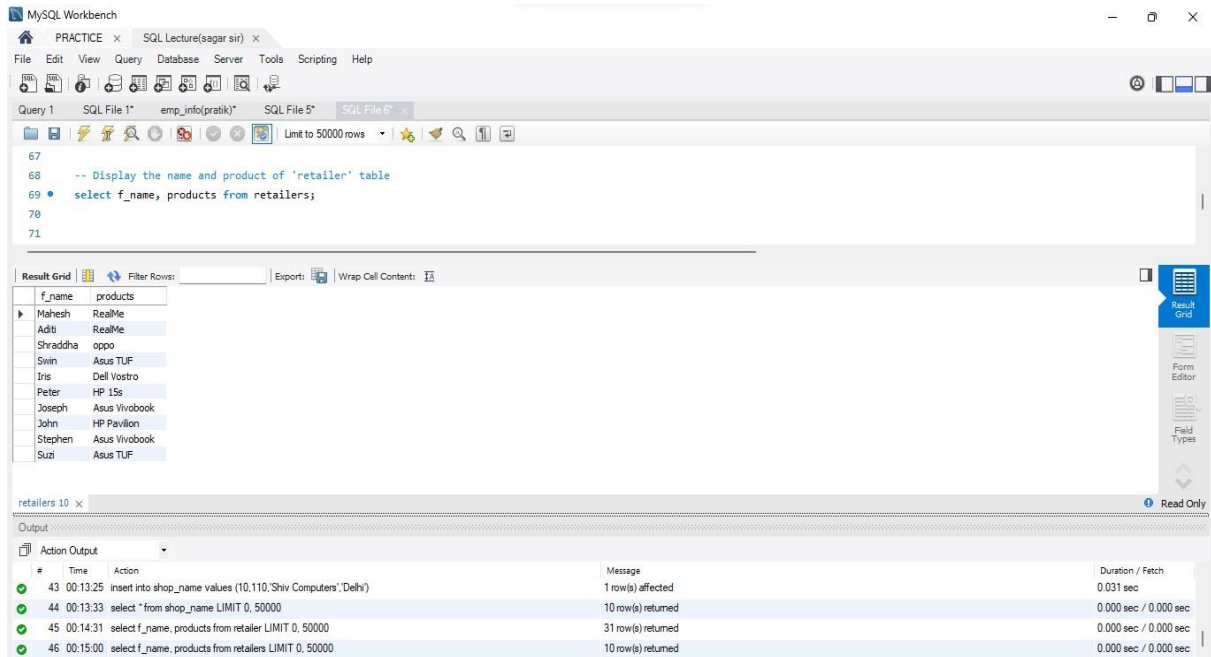
The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
-- Created another shop_name table  
44 • create table shop_name(  
45   ID int primary key,  
46   Seller_ID int,  
47   shop_names varchar(50),  
48   dep_location varchar(50));  
49 • select * from shop_name;  
50
```

The Result Grid shows the data inserted into the 'shop_name' table:

| ID | Seller_ID | shop_names | dep_location |
|----|-----------|------------|--------------|
|----|-----------|------------|--------------|

Query 1 : Display the Name and Products of 'retails' in table



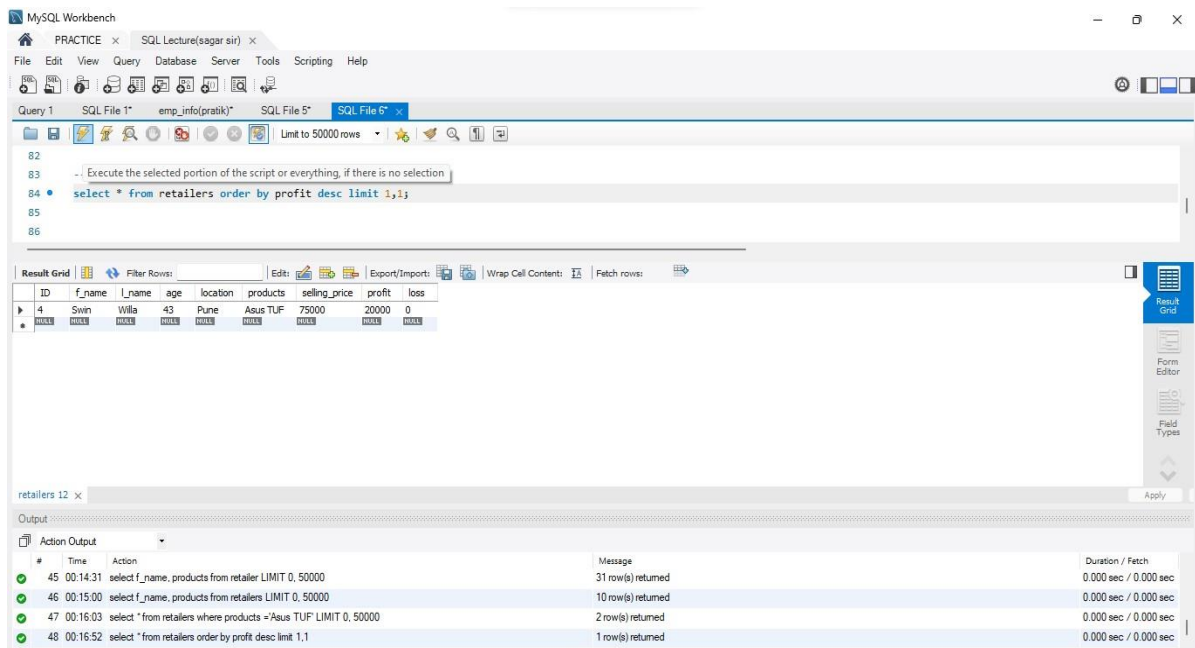
MySQL Workbench interface showing Query 1: Display the Name and Products of 'retails' in table. The query is executed, and the result grid displays 10 rows of data.

| f_name | products |
|---------|---------------|
| Maresh | RealMe |
| Aditi | RealMe |
| Shradha | oppo |
| Swin | Asus TUF |
| Iris | Dell Vostro |
| Peter | HP 15s |
| Joseph | Asus Vivobook |
| John | HP Pavilion |
| Stephen | Asus Vivobook |
| Suzi | Asus TUF |

Output window showing the execution of the query:

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|--------------------|-----------------------|
| 43 | 00:13:25 | insert into shop_name values (10,110,'Shiv Computers','Delhi') | 1 row(s) affected | 0.031 sec |
| 44 | 00:13:33 | select * from shop_name LIMIT 0, 50000 | 10 row(s) returned | 0.000 sec / 0.000 sec |
| 45 | 00:14:31 | select f_name, products from retailer LIMIT 0, 50000 | 31 row(s) returned | 0.000 sec / 0.000 sec |
| 46 | 00:15:00 | select f_name, products from retailers LIMIT 0, 50000 | 10 row(s) returned | 0.000 sec / 0.000 sec |

Query 2 : Show the second highest 'Profit' of product



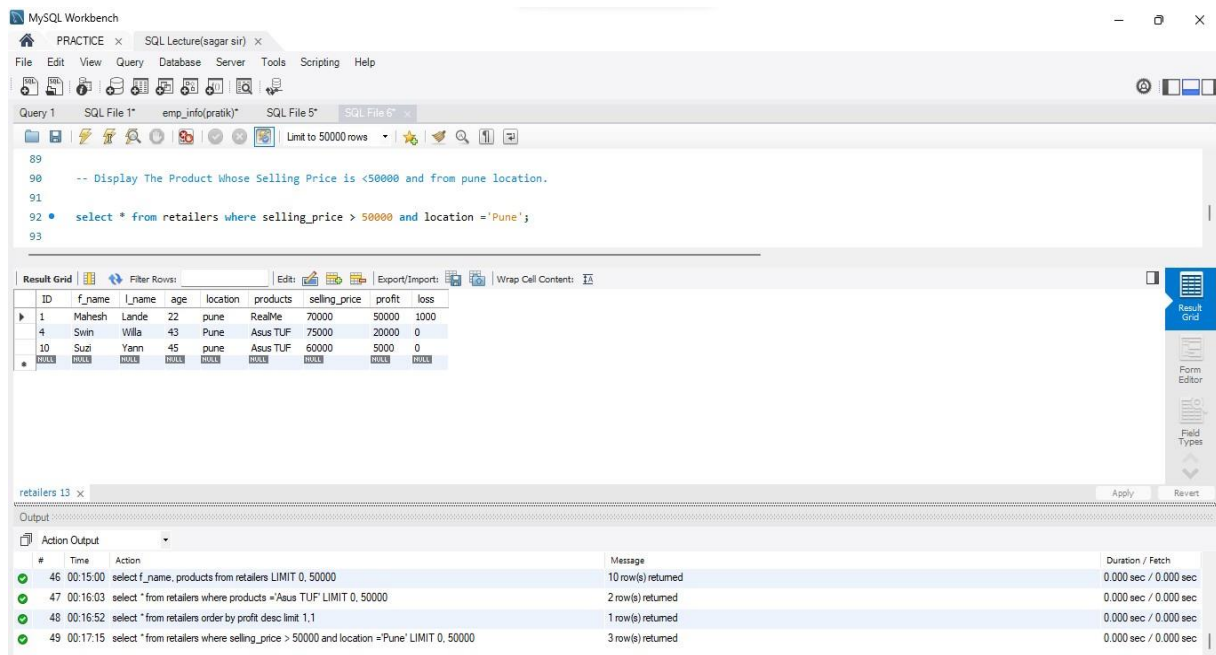
MySQL Workbench interface showing Query 2: Show the second highest 'Profit' of product. The query is executed, and the result grid displays 1 row of data.

| ID | f_name | l_name | age | location | products | selling_price | profit | loss |
|----|--------|--------|-----|----------|----------|---------------|--------|------|
| 4 | Swin | Willa | 43 | Pune | Asus TUF | 75000 | 20000 | 0 |

Output window showing the execution of the query:

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|--------------------|-----------------------|
| 45 | 00:14:31 | select f_name, products from retailer LIMIT 0, 50000 | 31 row(s) returned | 0.000 sec / 0.000 sec |
| 46 | 00:15:00 | select f_name, products from retailers LIMIT 0, 50000 | 10 row(s) returned | 0.000 sec / 0.000 sec |
| 47 | 00:16:03 | select * from retailers where products = 'Asus TUF' LIMIT 0, 50000 | 2 row(s) returned | 0.000 sec / 0.000 sec |
| 48 | 00:16:52 | select * from retailers order by profit desc limit 1,1 | 1 row(s) returned | 0.000 sec / 0.000 sec |

Query 3 : Display the Product whose selling price is greater than 50000 and from pune location.



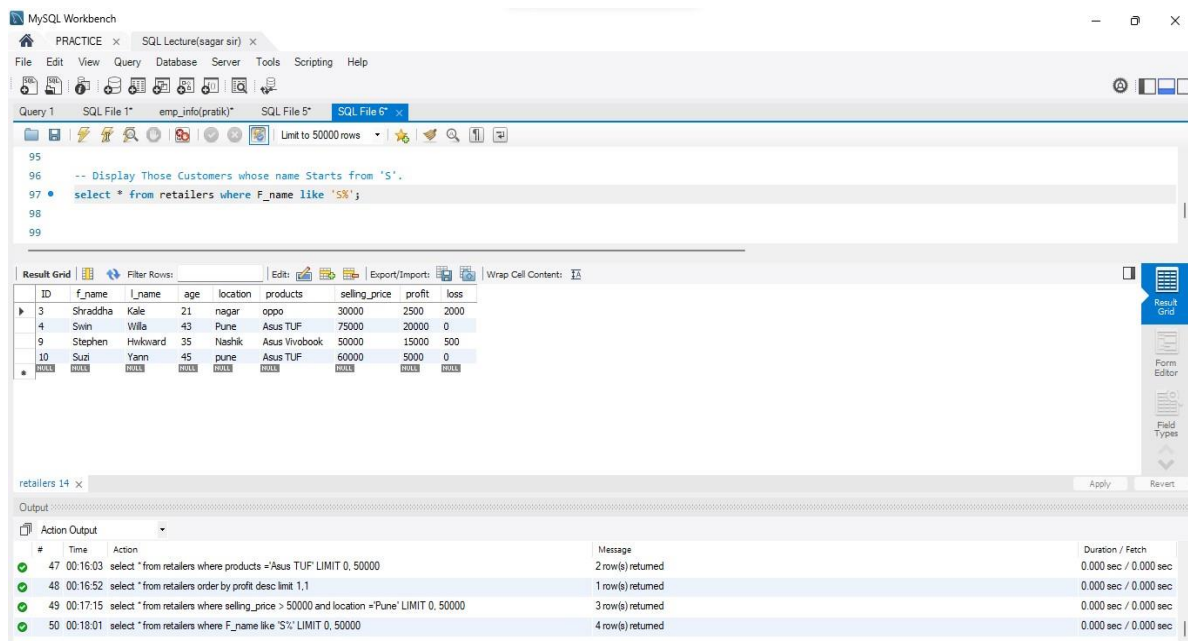
MySQL Workbench interface showing Query 3 execution. The query is: `select * from retailers where selling_price > 50000 and location = 'Pune';`. The result grid displays 3 rows of data:

| ID | f_name | l_name | age | location | products | selling_price | profit | loss |
|----|--------|--------|-----|----------|----------|---------------|--------|------|
| 1 | Mahesh | Lande | 22 | pune | RealMe | 70000 | 50000 | 1000 |
| 4 | Swin | Willa | 43 | Pune | Asus TUF | 75000 | 20000 | 0 |
| 10 | Suzi | Yann | 45 | pune | Asus TUF | 60000 | 5000 | 0 |

The Action Output pane shows the following messages:

- 46 00:15:00 select f_name, products from retailers LIMIT 0, 50000: 10 row(s) returned
- 47 00:16:03 select * from retailers where products = 'Asus TUF' LIMIT 0, 50000: 2 row(s) returned
- 48 00:16:52 select * from retailers order by profit desc limit 1,1: 1 row(s) returned
- 49 00:17:15 select * from retailers where selling_price > 50000 and location = 'Pune' LIMIT 0, 50000: 3 row(s) returned

Query 4 : Display those customers whose name starts from 'S' in table.



MySQL Workbench interface showing Query 4 execution. The query is: `select * from retailers where F_name like 'S%';`. The result grid displays 4 rows of data:

| ID | f_name | l_name | age | location | products | selling_price | profit | loss |
|----|----------|----------|-----|----------|---------------|---------------|--------|------|
| 3 | Shraddha | Kale | 21 | nagar | oppo | 30000 | 2500 | 2000 |
| 4 | Swin | Willa | 43 | Pune | Asus TUF | 75000 | 20000 | 0 |
| 9 | Stephen | Hawkeard | 35 | Nashik | Asus Vivebook | 50000 | 15000 | 500 |
| 10 | Suzi | Yann | 45 | pune | Asus TUF | 60000 | 5000 | 0 |

The Action Output pane shows the following messages:

- 47 00:16:03 select * from retailers where products = 'Asus TUF' LIMIT 0, 50000: 2 row(s) returned
- 48 00:16:52 select * from retailers order by profit desc limit 1,1: 1 row(s) returned
- 49 00:17:15 select * from retailers where selling_price > 50000 and location = 'Pune' LIMIT 0, 50000: 3 row(s) returned
- 50 00:18:01 select * from retailers where F_name like 'S%' LIMIT 0, 50000: 4 row(s) returned

Query 6 : Display the name, location, product, selling price from retailers.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
101 -- Display the name, location, product of those retailers whose Selling Price between 40000 to 60000 in decendeing order
102
103 • select F_name, l_name, location, products, selling_price from retailers
104 where selling_price between 40000 and 60000 order by selling_price desc;
105
```

The Result Grid shows the following data:

| F_name | l_name | location | products | selling_price |
|---------|---------|----------|---------------|---------------|
| John | Bottom | Mumbai | HP Pavilion | 60000 |
| Suzi | Yann | pune | Asus TUF | 60000 |
| Stephen | Hikward | Nashik | Asus Vivobook | 50000 |
| Aditi | Jadhav | Beed | RealMe | 45000 |
| Joseph | Alzery | Pune | Asus Vivobook | 44000 |
| Peter | Kewin | Nashik | HP 15s | 40000 |

The Output panel shows the execution log:

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|-------------------|-----------------------|
| 48 | 00:16:52 | select * from retailers order by profit desc limit 1.1 | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 49 | 00:17:15 | select * from retailers where selling_price > 50000 and location = 'Pune' LIMIT 0.50000 | 3 row(s) returned | 0.000 sec / 0.000 sec |
| 50 | 00:18:01 | select * from retailers where F_name like 'S%': LIMIT 0.50000 | 4 row(s) returned | 0.000 sec / 0.000 sec |
| 51 | 00:18:23 | select F_name, l_name, location, products, selling_price from retailers where selling_price between 40000 and 600... | 6 row(s) returned | 0.000 sec / 0.000 sec |

Query 7 : Write a query to find the name of all retailer department location in Mumbai

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
111
112
113 -- Write a query to find the name of all retsailer dep_location in mumbai (from both tables).
114
115 • select retailers.f_name, l_name, shop_info.dep_location from retailers
116 inner join shop_info on shop_info.seller_id=retailers.id where dep_location='delhi';
117
```

The Result Grid shows the following data:

| f_name | l_name | dep_location |
|----------|---------|--------------|
| Aditi | Jadhav | Delhi |
| Shraddha | Kale | Delhi |
| Swin | Willa | Delhi |
| Iris | Devine | Delhi |
| Peter | Kewin | Delhi |
| Joseph | Alzery | Delhi |
| John | Bottom | Delhi |
| Stephen | Hikward | Delhi |
| Suzi | Yann | Delhi |

The Output panel shows the execution log:

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|-------------------|-----------------------|
| 57 | 00:25:16 | select f_name, l_name, selling_price from retailers where selling_price > select Selling_price from retailer where f_n... | 4 row(s) returned | 0.000 sec / 0.000 sec |
| 58 | 00:25:43 | select retailers.f_name, l_name, shop_info.dep_location from retailers inner join shop_info on shop_info.seller_id=reta... | 9 row(s) returned | 0.016 sec / 0.000 sec |

Query 8 : Write a query to find the name and selling price of the employees who have higher selling price than 'Aditi' retailer.

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

```

106
107
108 -- Write a query to find the name and the selling price of the employees who have
109 -- higher selling Price than the 'Aditi'.
110 • select f_name ,l_name, selling_price from retailers where selling_price >(select Selling_price from retailer where f_name='Aditi');
111
112

```

The Result Grid shows the following data:

| f_name | l_name | selling_price |
|--------|--------|---------------|
| Maresh | Lande | 70000 |
| Swin | Willa | 75000 |
| John | Bottom | 60000 |
| Suzi | Yann | 60000 |

The Action Output shows the execution of the query:

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|-------------------|-----------------------|
| 56 | 00:23:43 | select retailers.f_name,l_name,shop_info.dep_location from retailers inner join shop_info on shop_info.seller_id=ret... | 9 row(s) returned | 0.000 sec / 0.000 sec |
| 57 | 00:25:16 | select f_name,l_name, selling_price from retailers where selling_price >(select Selling_price from retailer where f_n... | 4 row(s) returned | 0.000 sec / 0.000 sec |

Query 9 : Write a query to display all the products names where no. of products are less than 2

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

```

157
158 -- write a query to display all the products names where no. of products are less than 2
159
160 • select products from retailer group by products having count(*) < 2 ;
161
162
163
164
165
166
167
168
169

```

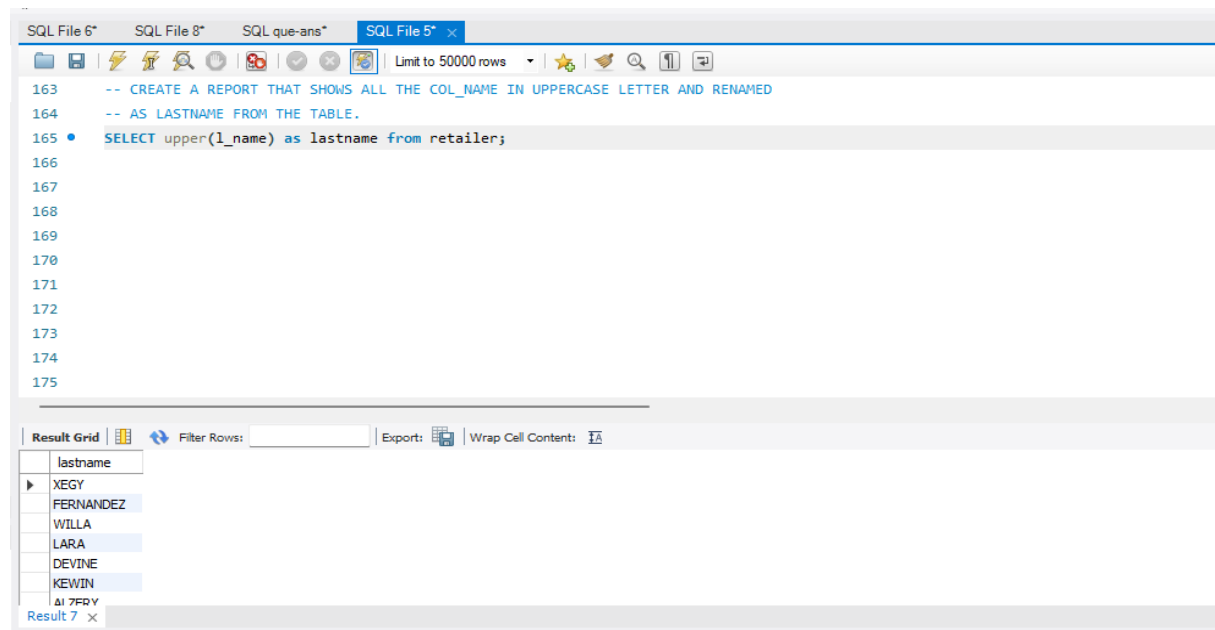
The Result Grid shows the following data:

| products |
|----------|
| HP 14s |

The Action Output shows the execution of the query:

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|-------------------|-----------------------|
| 56 | 00:23:43 | select retailers.f_name,l_name,shop_info.dep_location from retailers inner join shop_info on shop_info.seller_id=ret... | 9 row(s) returned | 0.000 sec / 0.000 sec |
| 57 | 00:25:16 | select f_name,l_name, selling_price from retailers where selling_price >(select Selling_price from retailer where f_n... | 4 row(s) returned | 0.000 sec / 0.000 sec |

Query 10 : Create a Report that shows all the col_name inUPPER case and renamed as lastname from the table.



The screenshot shows a SQL IDE interface with a query editor and a results grid. The query editor contains the following SQL code:

```
163 -- CREATE A REPORT THAT SHOWS ALL THE COL_NAME IN UPPERCASE LETTER AND RENAMED
164 -- AS LASTNAME FROM THE TABLE.
165 • SELECT upper(l_name) as lastname from retailer;
166
167
168
169
170
171
172
173
174
175
```

The results grid displays the output of the query, showing a single column named 'lastname' with the following values:

| lastname |
|-----------|
| XEGY |
| FERNANDEZ |
| WILLA |
| LARA |
| DEVINE |
| KEWIN |
| Δ 1 7PRY |

The interface also includes a toolbar with various icons for file operations, a 'Limit to 50000 rows' dropdown, and a 'Filter Rows' input field. The 'Export' button is also visible.

PROBLEM STATEMENT

In this project I had some queries to run, while doing that I faced problem such as the queries were correct but still it used to not be executed.

The MYSQL workbench gets updated if the updates are available sometimes the software give problem (giving syntax error).

So I tried to run the queries on other device. surprisingly, It could run on that device.

CONCLUSION

This module has presented all the basic hurdles to using and understanding SQL. You should understand, play with, and feel comfortable with these statements because you will use them constantly when you work with an SQL database system. You will be using the SELECT statement in extracting the information needed for the class project. It's the core of most of the work you will be doing with SQL.

It would be a good idea to review this module before continuing. From here, we will build on the SELECT statement and show how you can do more advanced database queries. The balance of the language is built on the INSERT, SELECT, UPDATE, and DELETE statements.

SQL statements let you perform simple tasks with a database such as creating a new table or inserting a record. By combining many SQL statements into a script, you can perform elaborate procedures such as creating and initializing a database from scratch. It describes the benefits of using scripts to create databases and discusses some of the issues that you should understand before writing those scripts.

FUTURE SCOPE

•

SQL Server Integration Services (SSIS): This service helps you with the integration of multiple databases, database objects and entities as per your requirement.

- **Microsoft SQL Server:** This is the most difficult yet intriguing job for SQL developer where he has to work on server performance, integrity as well as server maintenance. This is one of the highly paid SQL Developer skills where developers get attractive remuneration and other opportunities too.
- **Analytical skills:** Developers must analyse and understand the needs of the user and then design the software accordingly to meet those needs.
- **Creativity:** Developers must be creative enough to find new solutions to conventional problems.
- **Detail-oriented:** Developers usually work on several parts of an application or system simultaneously and therefore must be able to focus and pay attention to even the minute details.
- **Interpersonal skills:** Developers must be able to work in harmony with others who help in developing and creating successful software.
- **Problem-solving skills:** Since developers are in charge of software from beginning to end, they should be capable of solving problems that may occur along the design process.