School of Engineering & Technology
Department of Computer Science & Technology

Data Base Management System Lab
(CSE249)

Lab File
(2024-2025)

for

B. Tech. (CSE)
4th Semester

Submitted To:

Dr. Abhishek Singh Verma

Associate Professor,

Department of Computer Science & Engineering

School of Engineering & Technology

Sharda University

Submitted By:

Masood Aslam

B. Tech. CSE [4th Semester]

2301010501

Group 1

EXPERIMENT-1

OBJECTIVE : Program to Create Table.

SOFTWARE USED : MySQL Workbench 8.0 CE

THEORY

CREATE : It is a DDL command. It is used to create table in SQL.

SYNTAX

CREATE TABLE Table_name

(

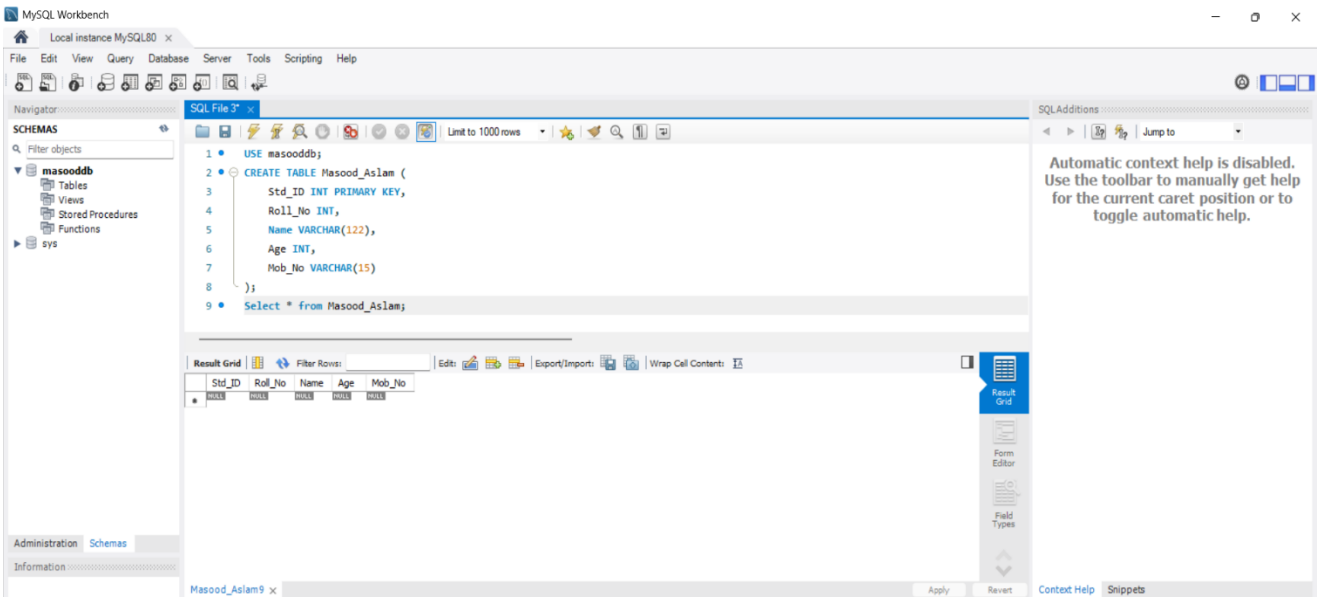
Attribute_name1 Data_type(Size),

Attribute_name2 Data_type(Size),

……

Attribute_name3 Data_type(Size)

);



EXPERIMENT 2

OBJECTIVE : Program to Alter Table.

SOFTWARE USED : MySQL Workbench 8.0 CE

THEORY

ALTER : It is a DDL command. It is used to modify table structure in SQL.

SYNTAX

1. **To add a column.**

   Alter table <Table Name>
   ADD (new col.name data type(size));

2. **To drop a column**

   ALTER Table <Table_Name>
   DROP column column_name ;

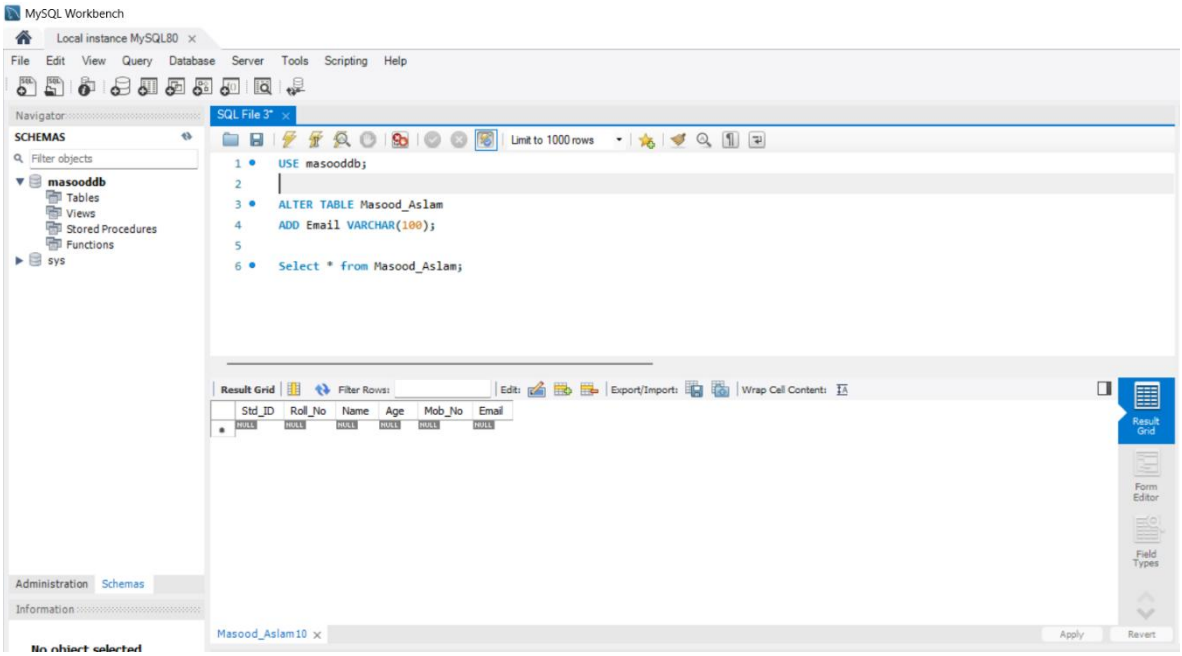
3. **To modify a column**

   ALTER table table_name
   MODIFY column_name new_Datatype (New Size);

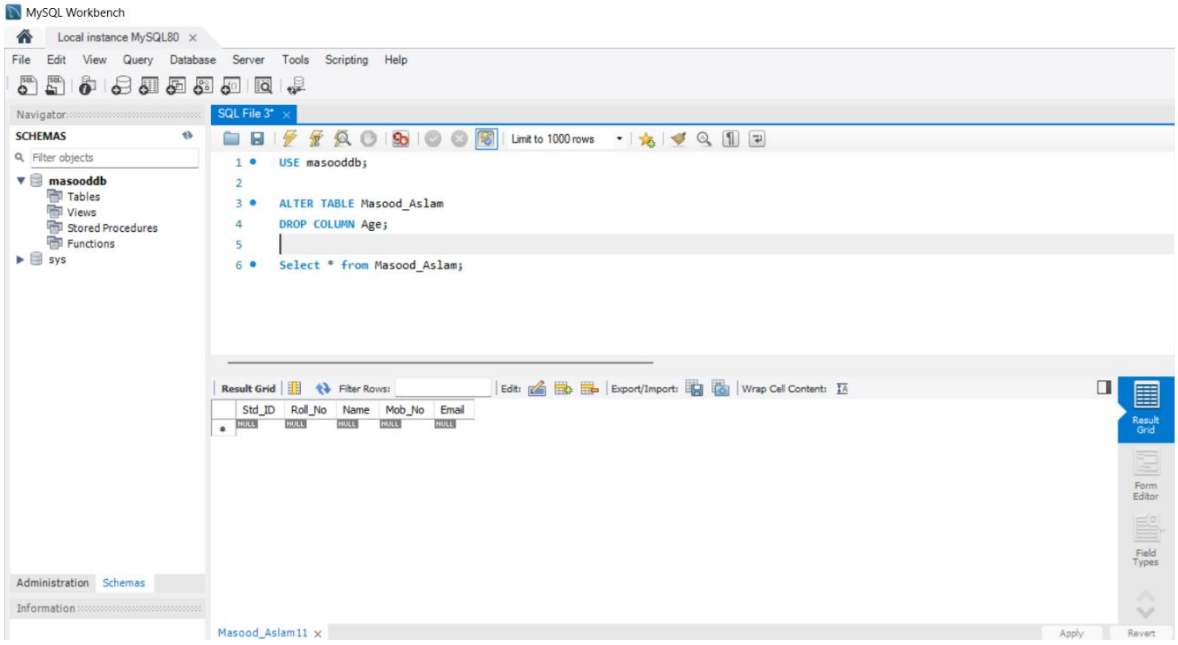
4. **To Rename a Column**

   ALTER table table_name
   RENAME column column_name to new_column_name

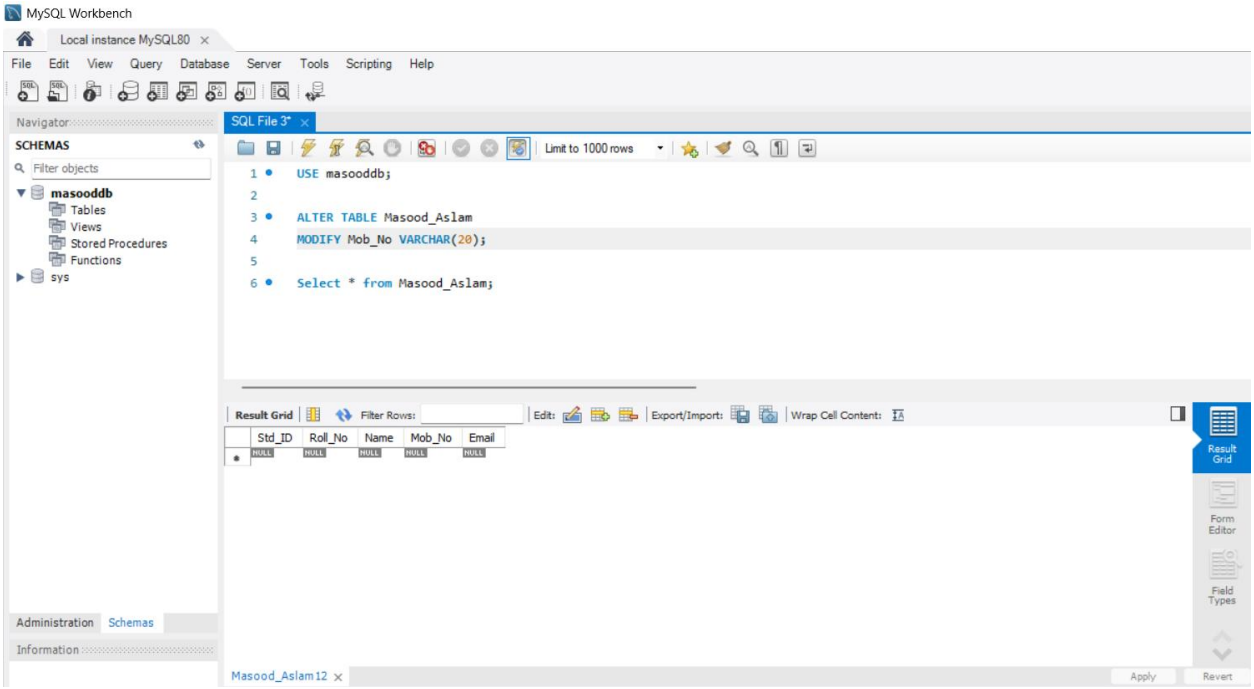
Adding a Column



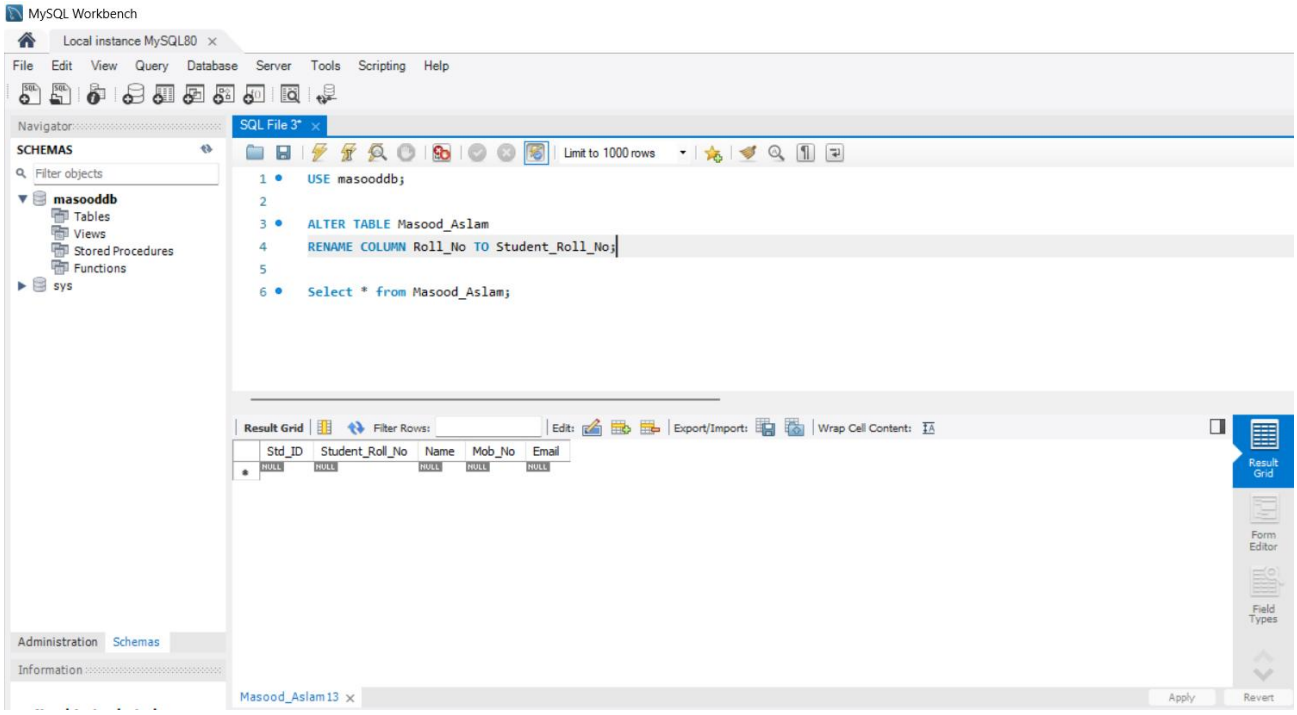
Dropping a Column



Modifying a Column



Renaming a Column



EXPERIMENT 3

OBJECTIVE : Program to Insert the Values in a table.

SOFTWARE USED : MySQL Workbench 8.0 CE

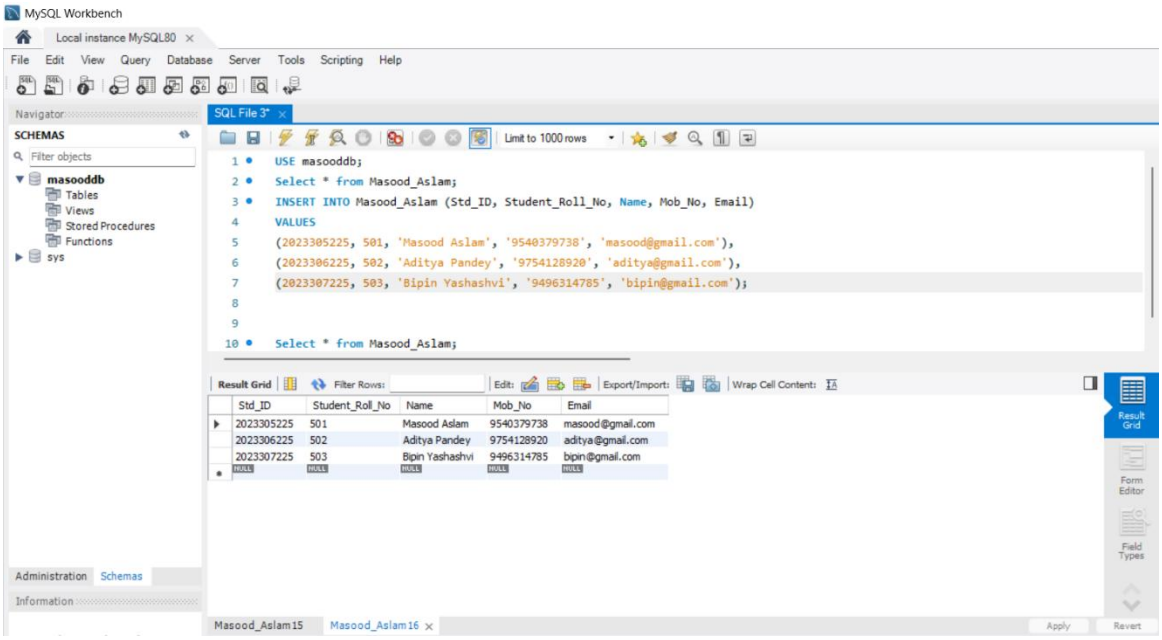
THEORY

INSERT: It is a DDL command. The insert statement uses the set clause, which specifies the values of a column. You can use more than one set clause per insert statement, and each set clause can set the values in more than one column. Multiple set clauses are not separated by commas. If you specify an optional list of columns, then you can set a value only for a column that is specified in the list of columns to be inserted and when character value insert then apply (‘ ‘) example (‘value’) otherwise not apply.

SYNTAX

1. **To add values in to table**
   Insert into table_name
   Values (value,……..,value);

Inserting values in a table



EXPERIMENT 4

OBJECTIVE : Program to View & Filter data in table.

SOFTWARE USED : MySQL Workbench 8.0 CE

THEORY

Once data has been inserted into a table, the next most logical operation would be to view what has been inserted. While viewing data from a table, it is rare that all the data from table will be required each time. Hence, SQL must give us a method of filtering out data that is not data required.

SYNTAX

1. **All Rows and All Columns**
   Select * from table_name;

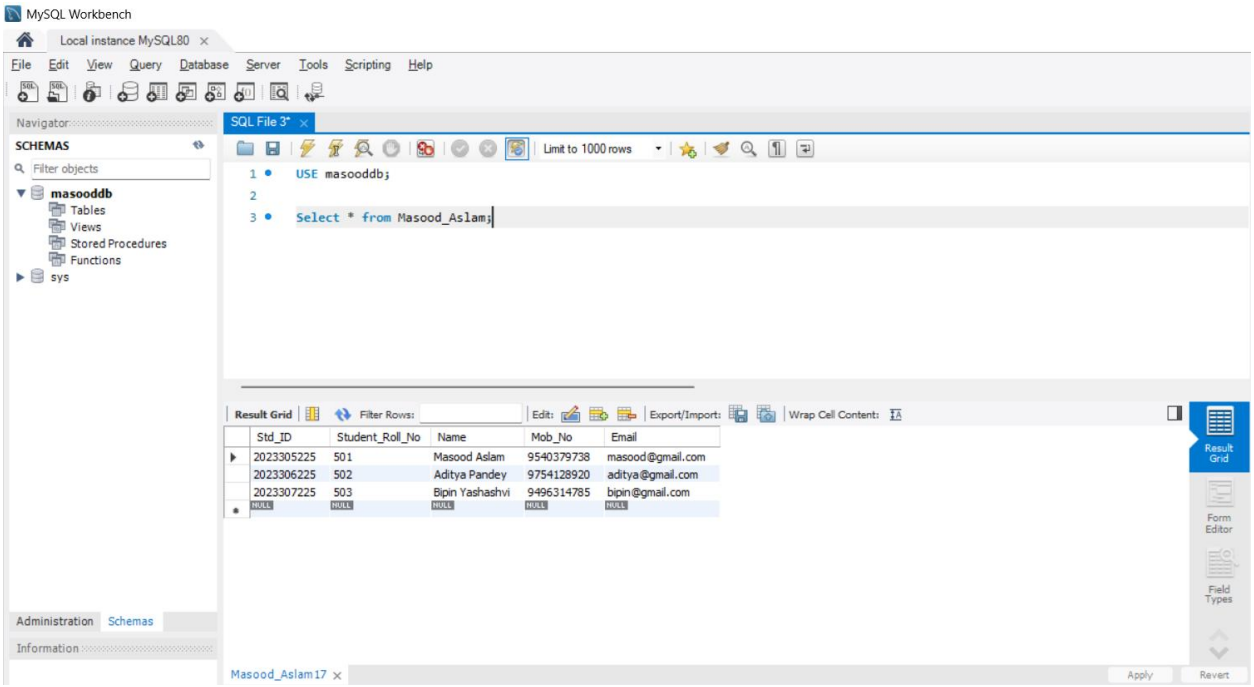
2. **Filtering Table Data**

   a. **Select Columns & all Rows**
      select column_name1, column_name2
      from table_name;

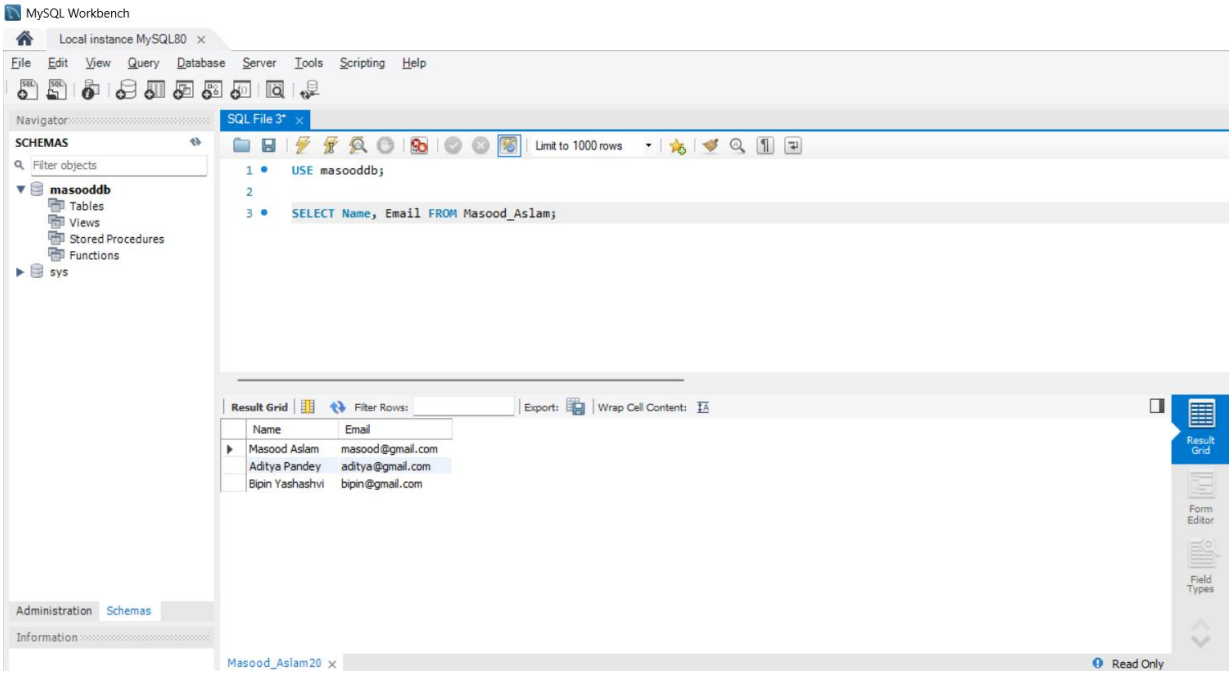
   b. **Select Rows and all Columns**
      select * from table_name
      where condition;

   c. **Select Rows and select Columns**
      select column_name1, column_name2
      from table_name
      where condition;

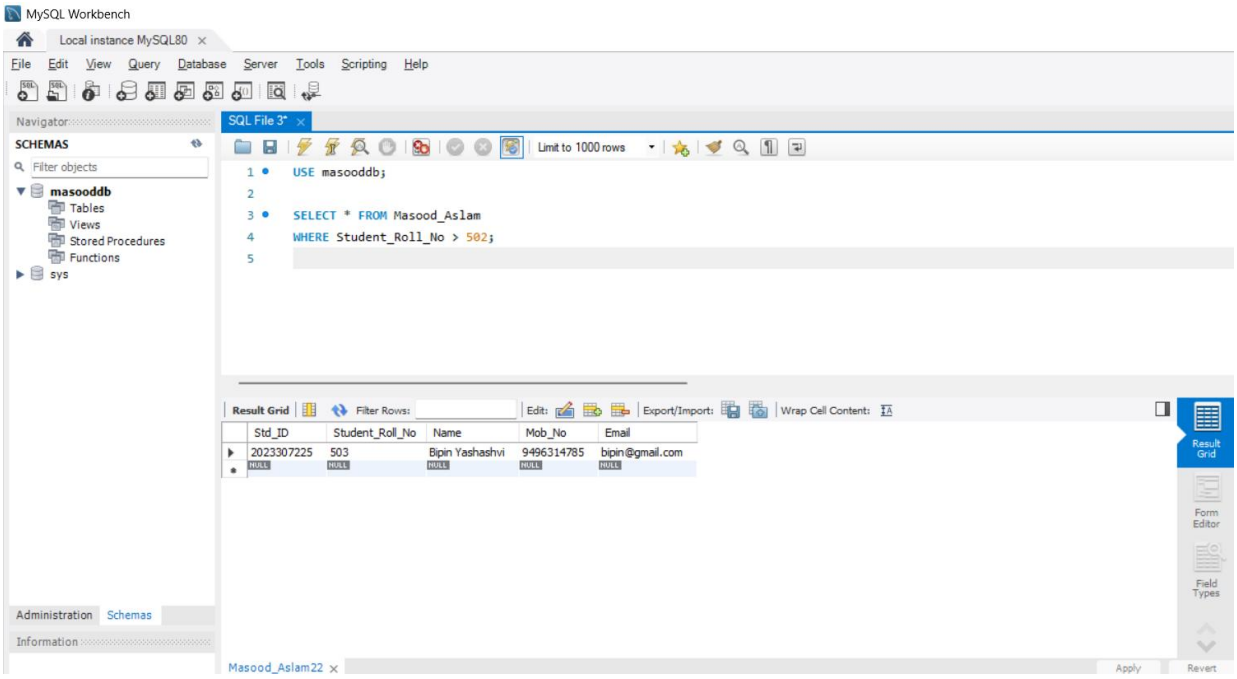
All Rows and All Columns



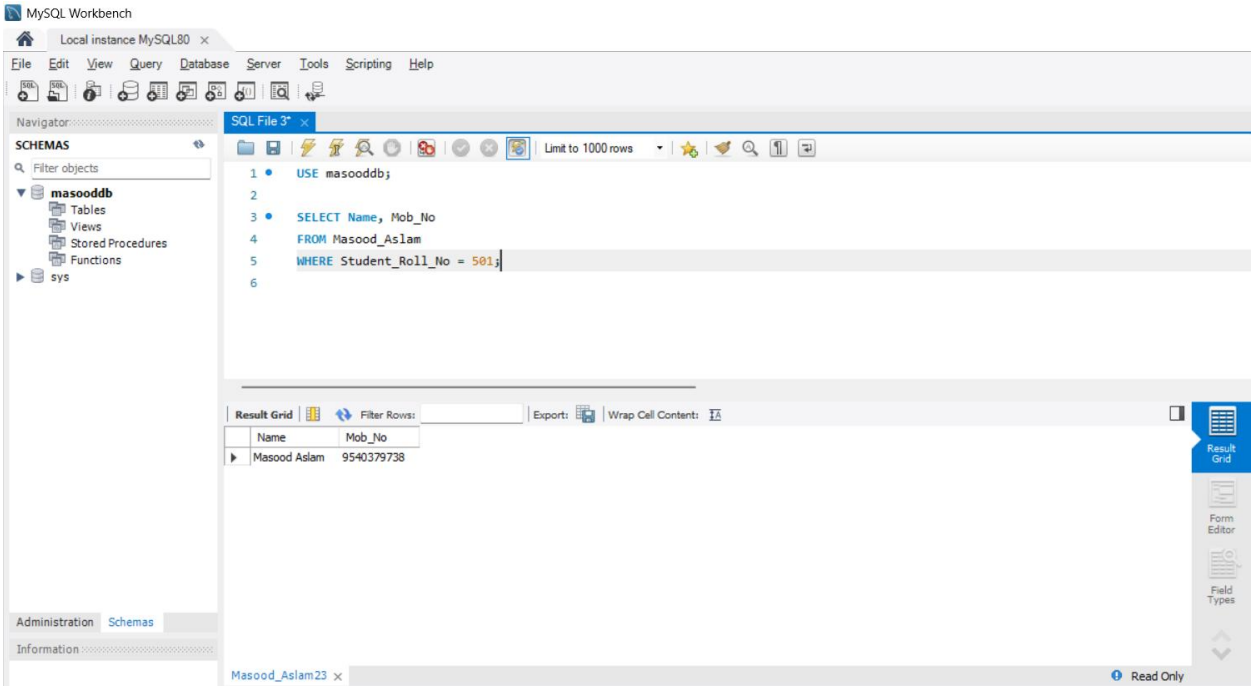
Select Columns & all Rows



Select Rows and all Columns



Select Rows and select Columns



EXPERIMENT 5

OBJECTIVE : Program to delete record of a table.

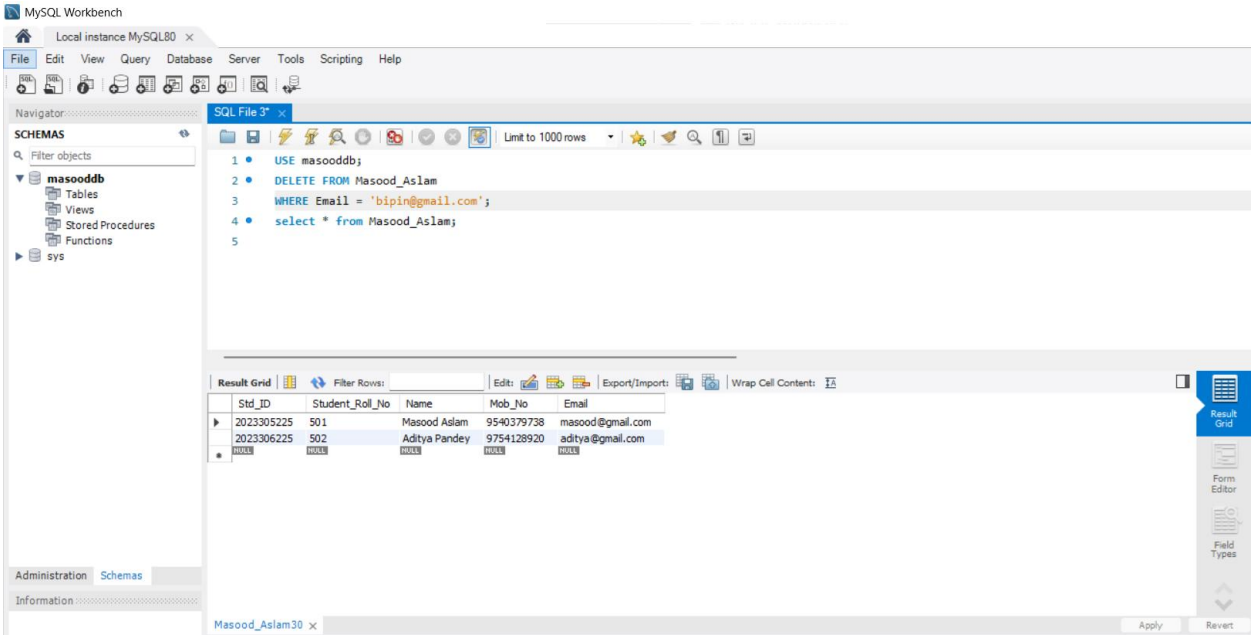
SOFTWARE USED : Microsoft SQL Server 2008 R2

THEORY

The DELETE statement is used to delete rows in a table.

SYNTAX

DELETE FROM *table_name*
WHERE *some_column=some_value*;



EXPERIMENT 6

OBJECTIVE : Program to update the content of a table.

SOFTWARE USED : Microsoft SQL Server 2008 R2

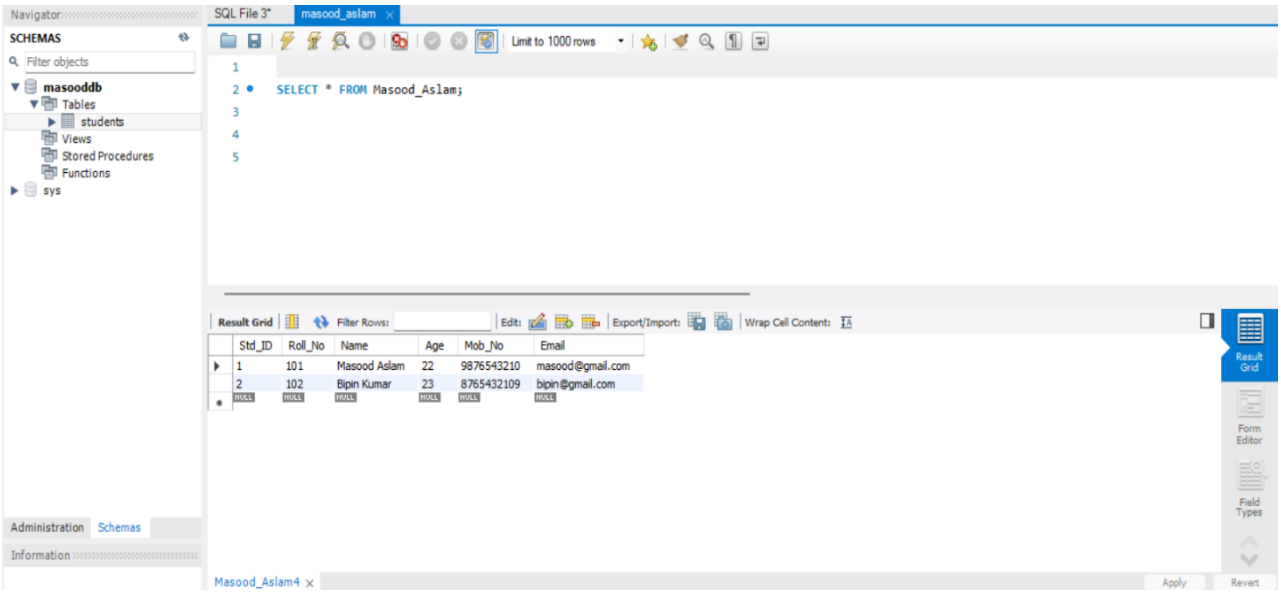
THEORY

The UPDATE statement is used to update existing records in a table.

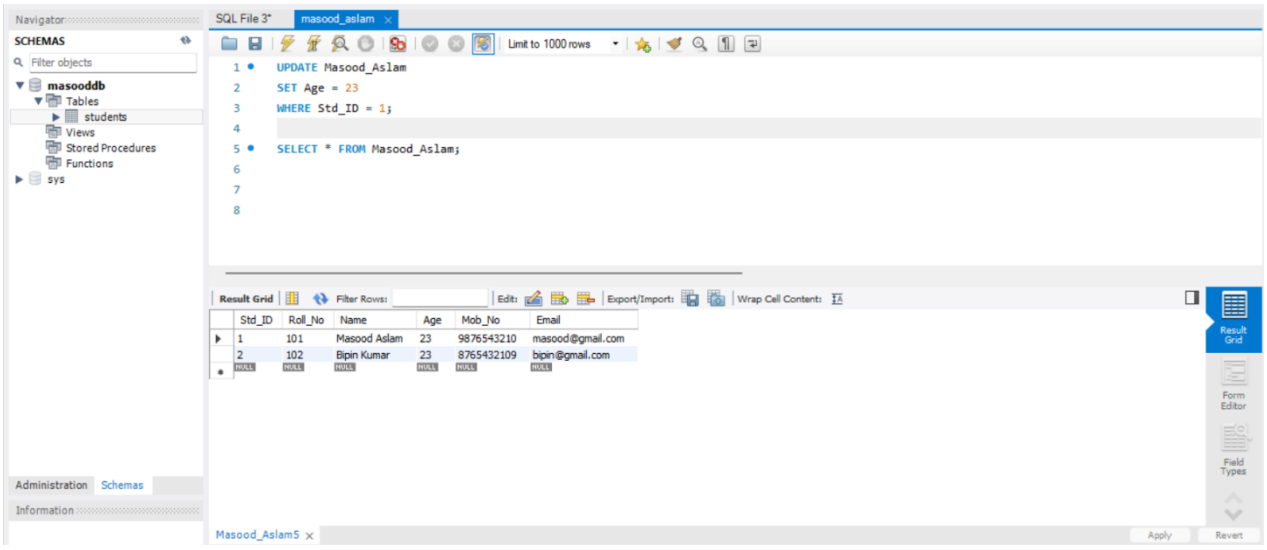
SYNTAX

UPDATE *table_name*
SET *column1=value1,column2=value2,...*
WHERE *some_column=some_value*;

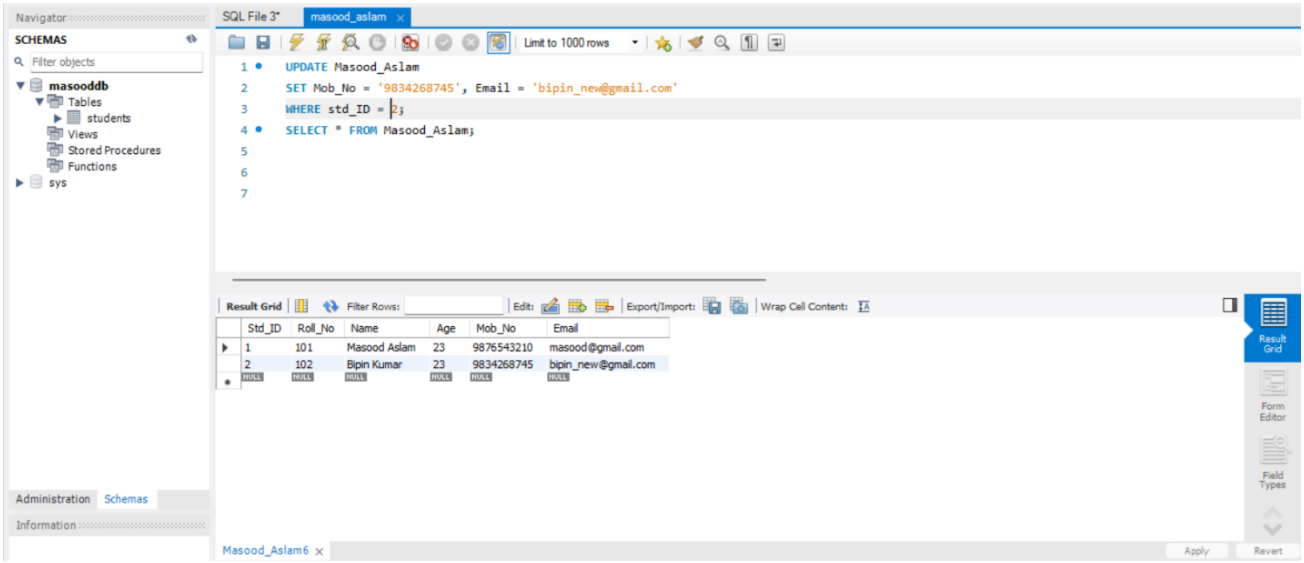
Before Updates



Updating a Single Column



Updating Multiple Columns



EXPERIMENT 7

OBJECTIVE : Write a program to sort the data of a table.

SOFTWARE USED : Microsoft SQL Server 2008 R2

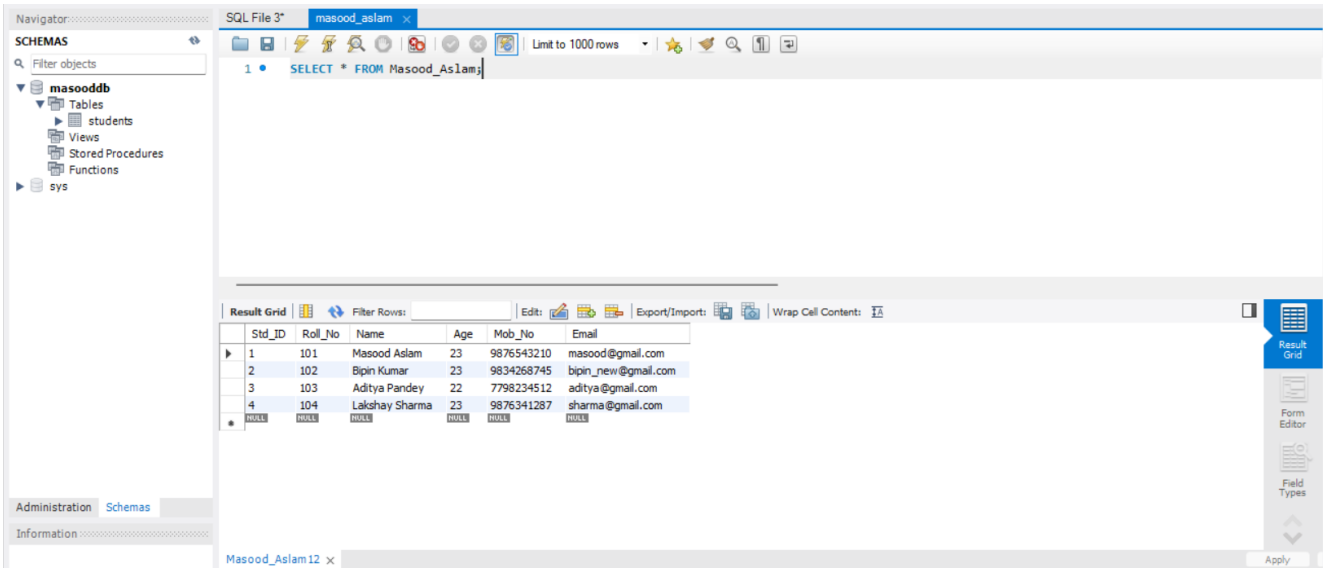
THEORY

The ORDER BY keyword is used to sort the result-set by one or more columns. The ORDER BY keyword sorts the records in ascending order by default. To sort the records in a descending order, you can use the DESC keyword.

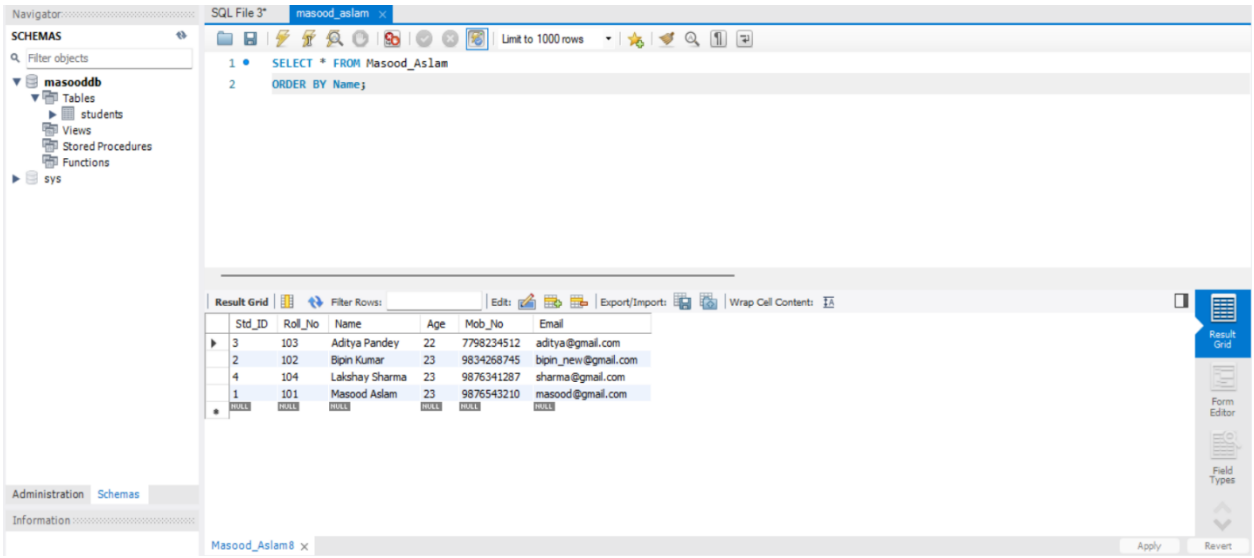
SYNTAX

```
SELECT column_name, column_name
FROM table_name
ORDER BY column_name ASC|DESC, column_name ASC|DESC;
```

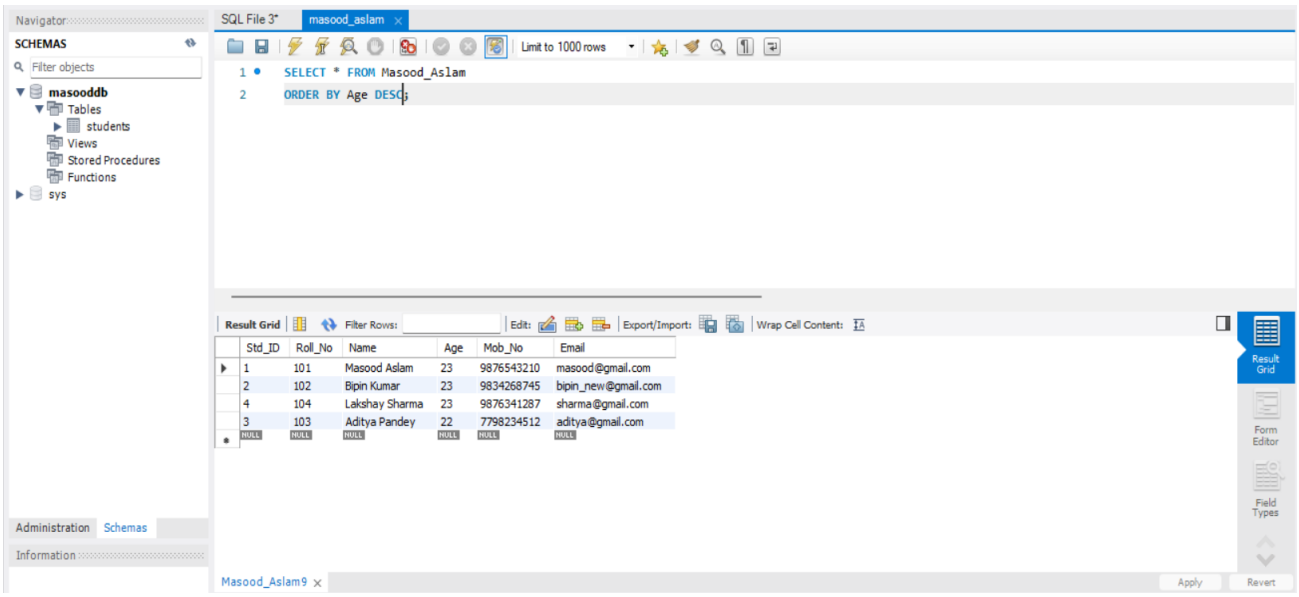
Before Sorting:



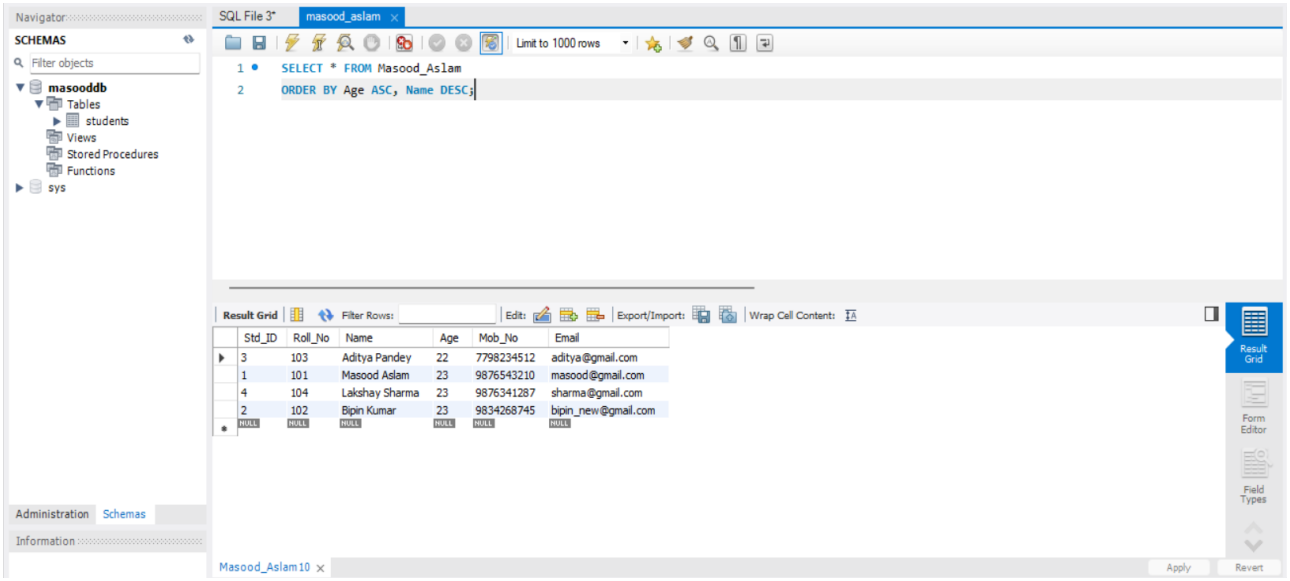
Sorting Data in Ascending Order (Default):



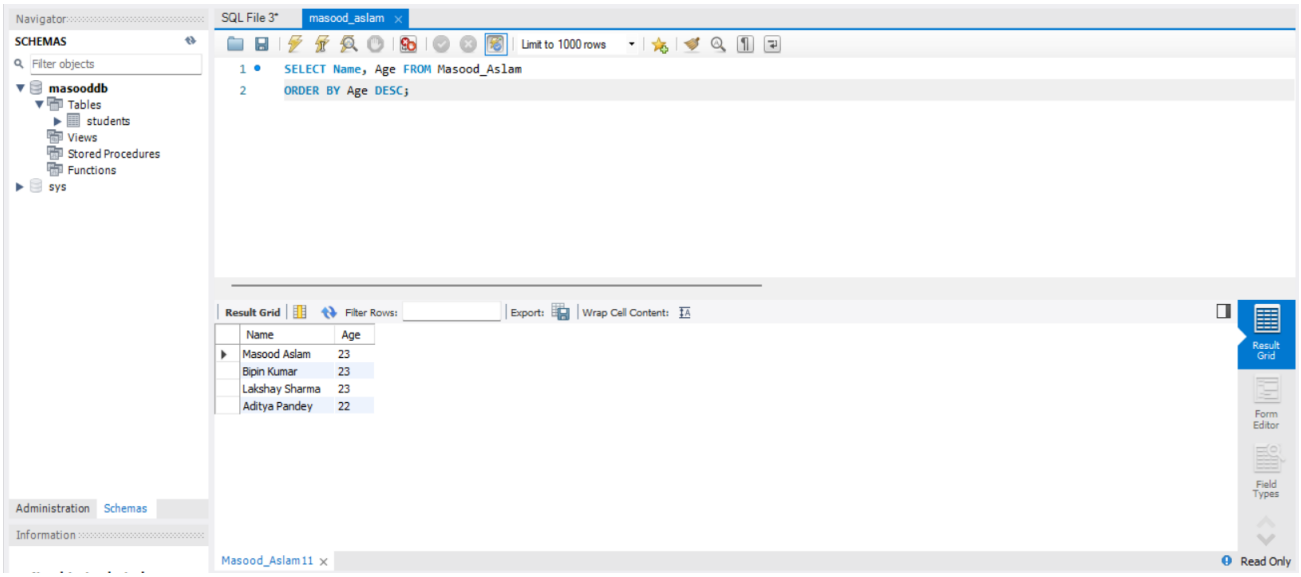
Sorting Data in Descending Order:



Sorting by Multiple Columns:



Sorting Specific Columns:



EXPERIMENT 8

OBJECTIVE : Write a program to implement set functioning SQL.

SOFTWARE USED : Microsoft SQL Server 2008 R2

THEORY

SQL supports few operations to be performed on table data.

UNION

Union is used to combine the results of two or more select statements.

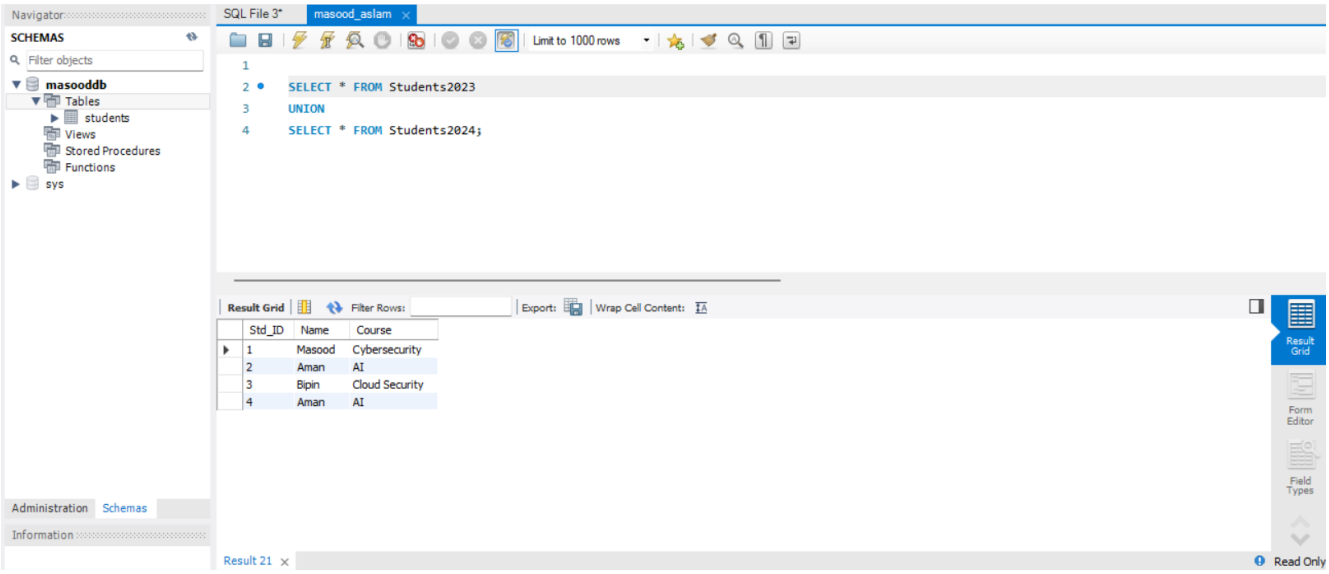
SYNTAX

SELECT * FROM first_table

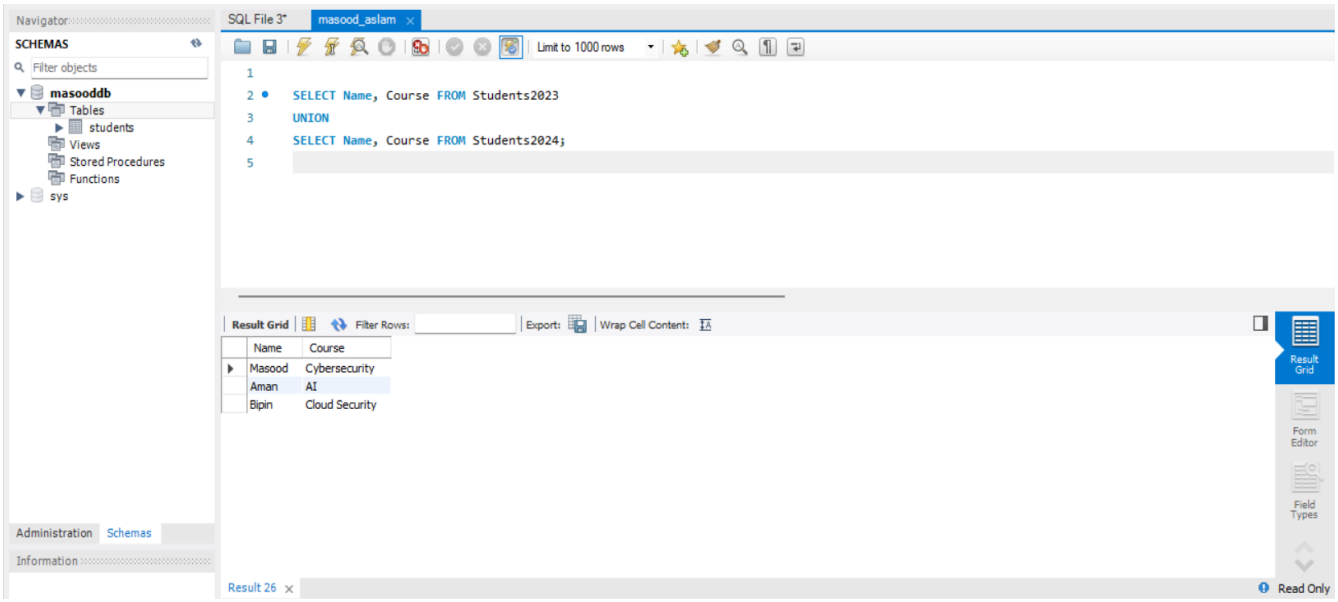
UNION

SELECT * FROM Second_table;

Using UNION to Merge Both Tables:



Using UNION with Specific Columns:



EXPERIMENT 9

OBJECTIVE : Write a program to implement JOIN operation on two tables.

SOFTWARE USED : Microsoft SQL Server 2008 R2

THEORY

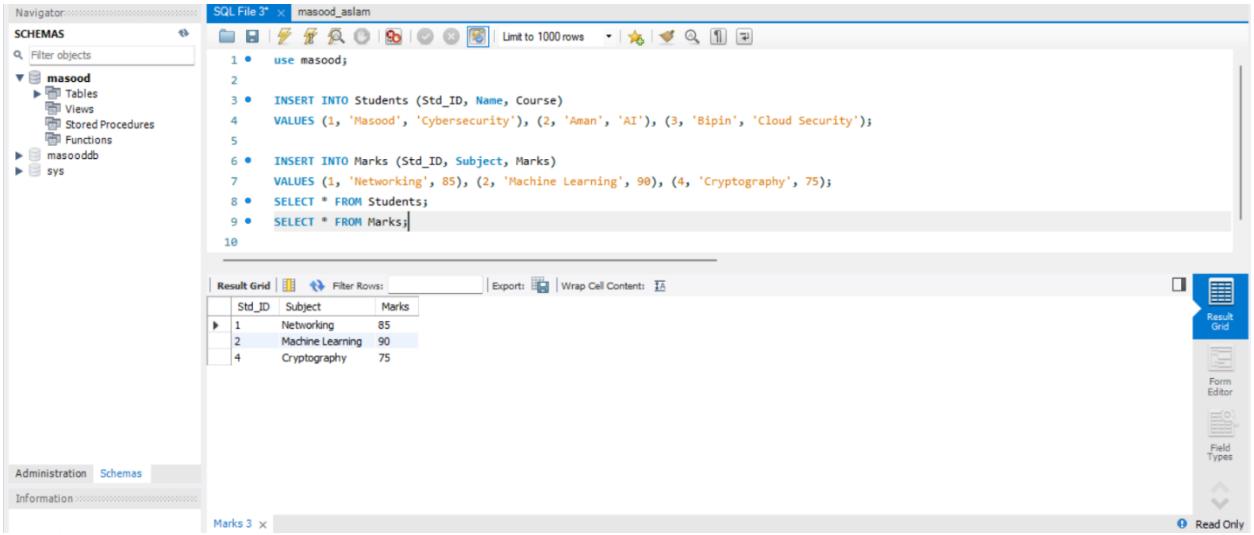
An SQL JOIN clause is used to combine rows from two or more tables, based on a common field between them. The most common type of join is: SQL INNER JOIN (simple join).An SQL INNER JOIN returns all rows from multiple tables where the join condition is met.

SYNTAX

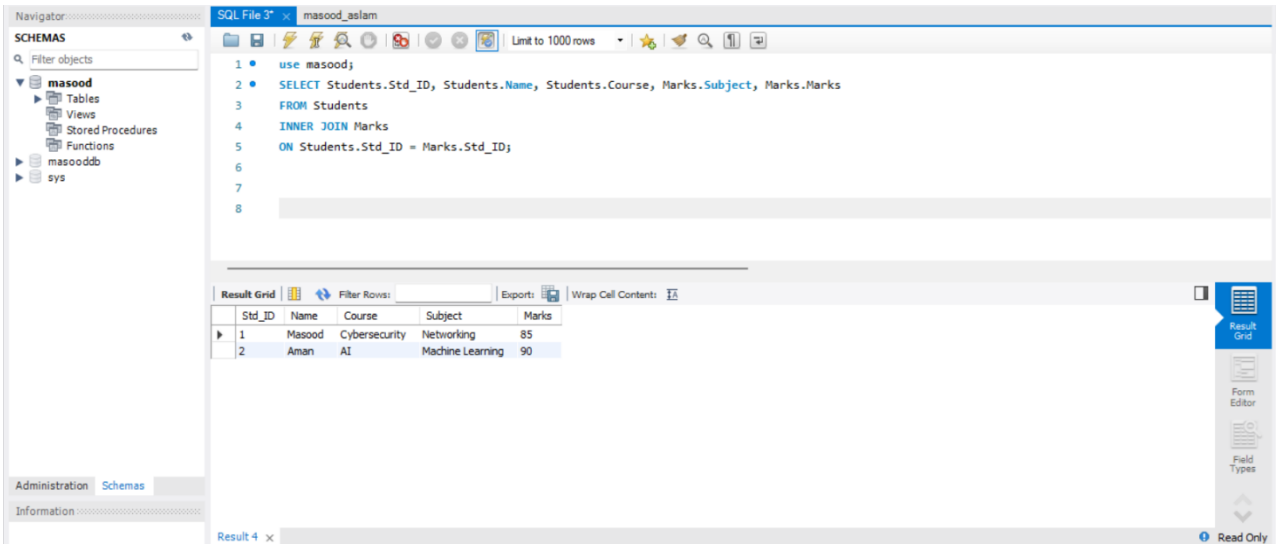
SELECT * FROM first_table,Second_table

WHERE first_table.common_attribute=second_table.common_attribute;

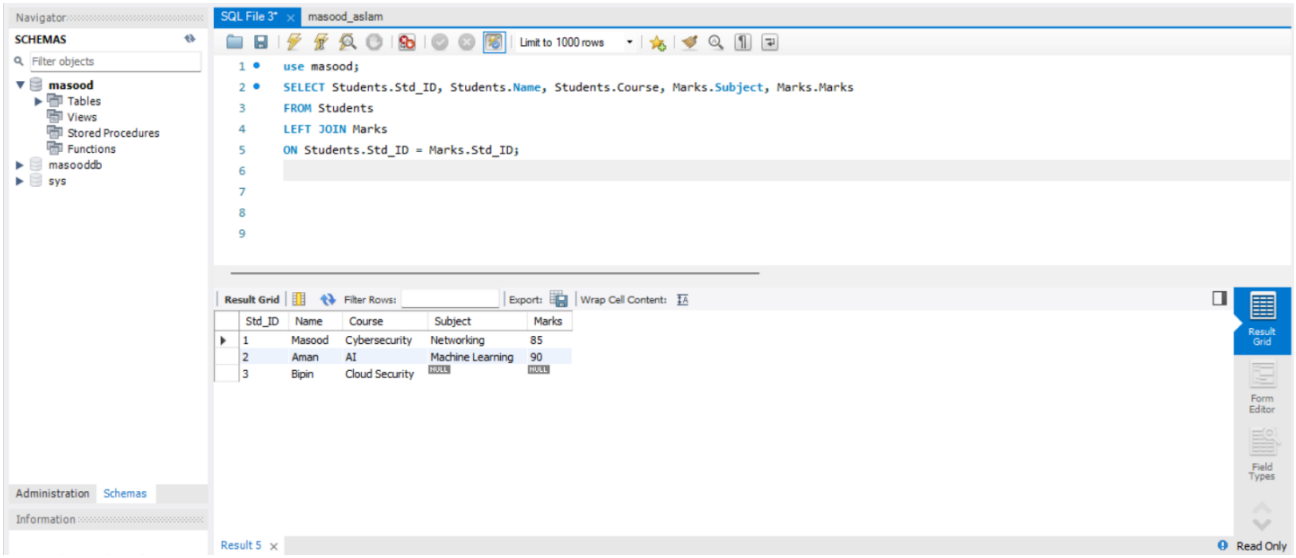
Inserting Sample Data



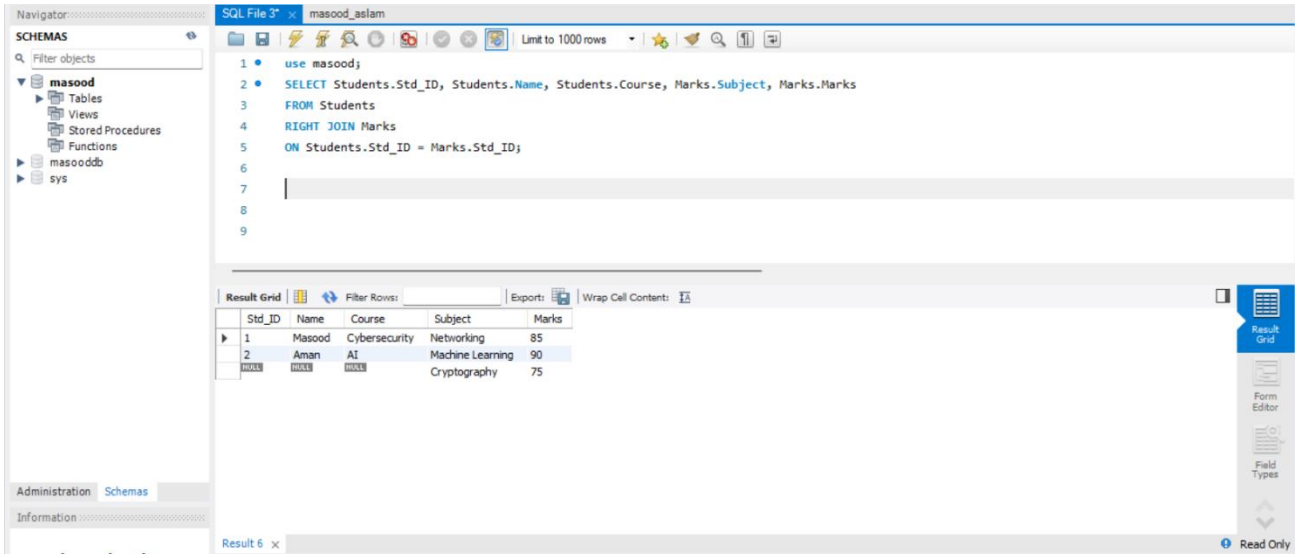
INNER JOIN (Common Matching Records Only)



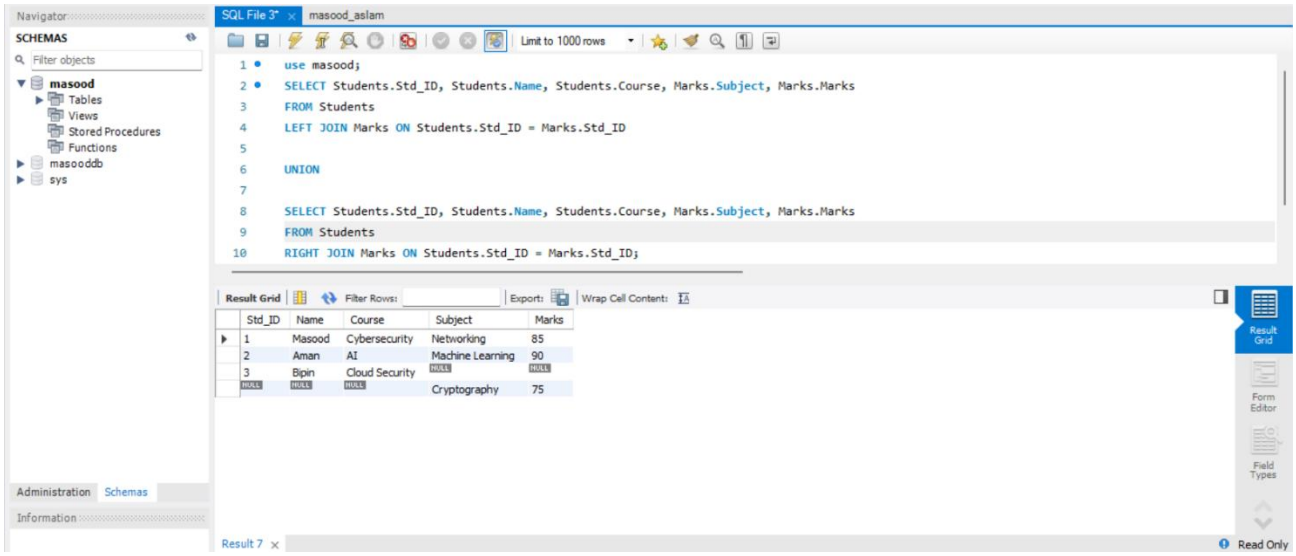
LEFT JOIN (All Students + Marks if Available)



RIGHT JOIN (All Marks + Students if Available)



FULL OUTER JOIN (All Data, Even If No Match)



EXPERIMENT 10

OBJECTIVE **:** Write a program to implement Aggregate functions using SQL.

SOFTWARE USED **:** Microsoft SQL Server 2008 R2

THEORY

The SQL aggregate functions, as their title suggests are used to retrieve minimum and maximum values from a column, to sum values in a column, to get the average of a column values, or to simply count a number of records according to a search condition.

SYNTAX

The most commonly used SQL aggregate function is the COUNT function;;

SELECT COUNT(*) FROM table_name WHERE condition;

We can select minimum and maximum from a Table as follows:

SELECT MIN(Column_name) FROM table_name;

SELECT MAX(Column_name)FROM table_name;

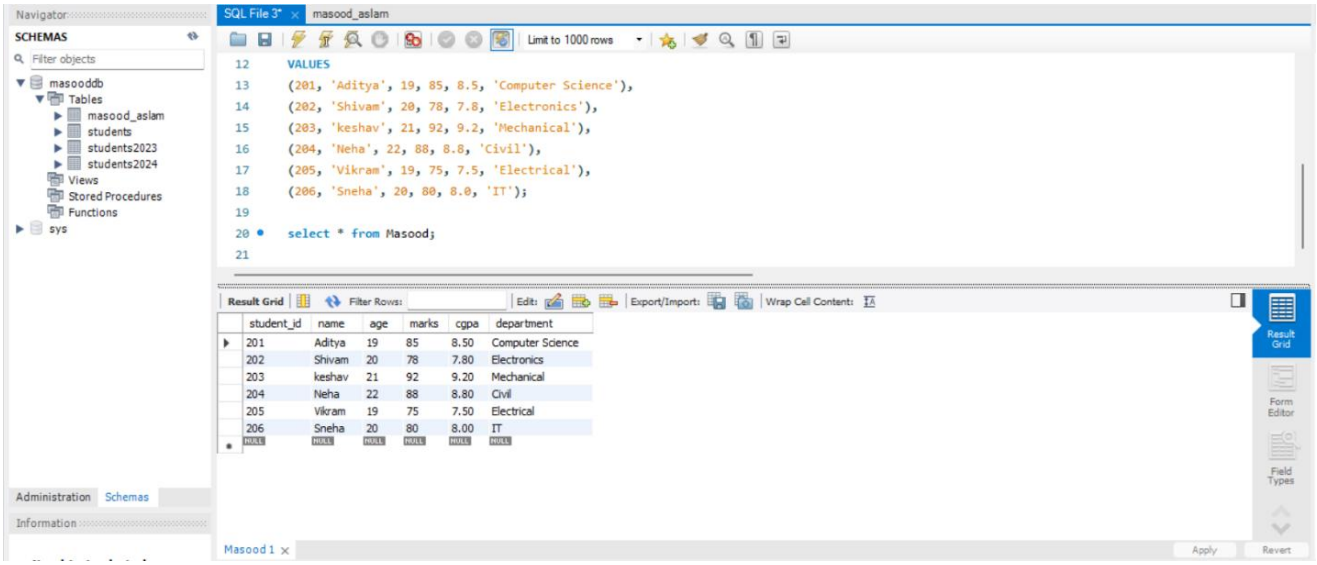
We can also select the average from a Table as follows:

SELECT AVG(Column_name)FROM Table_name;

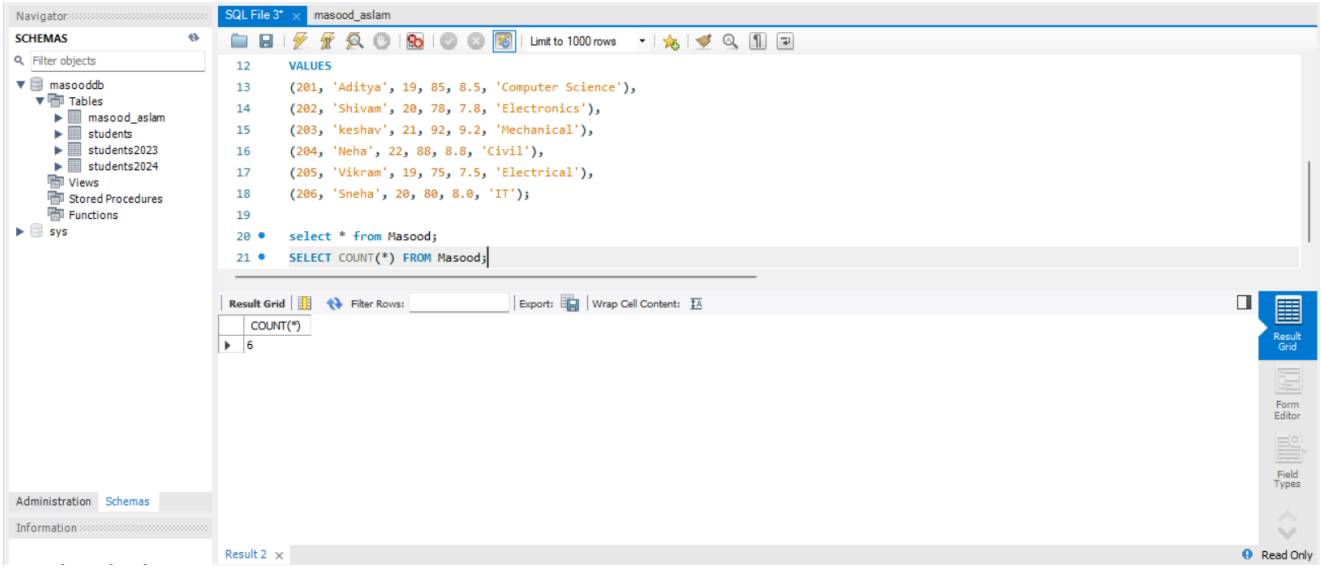
We can use the SUM aggregate function to get the sum of values in a certain column:

SELECT SUM(Column_name) FROM table_name;

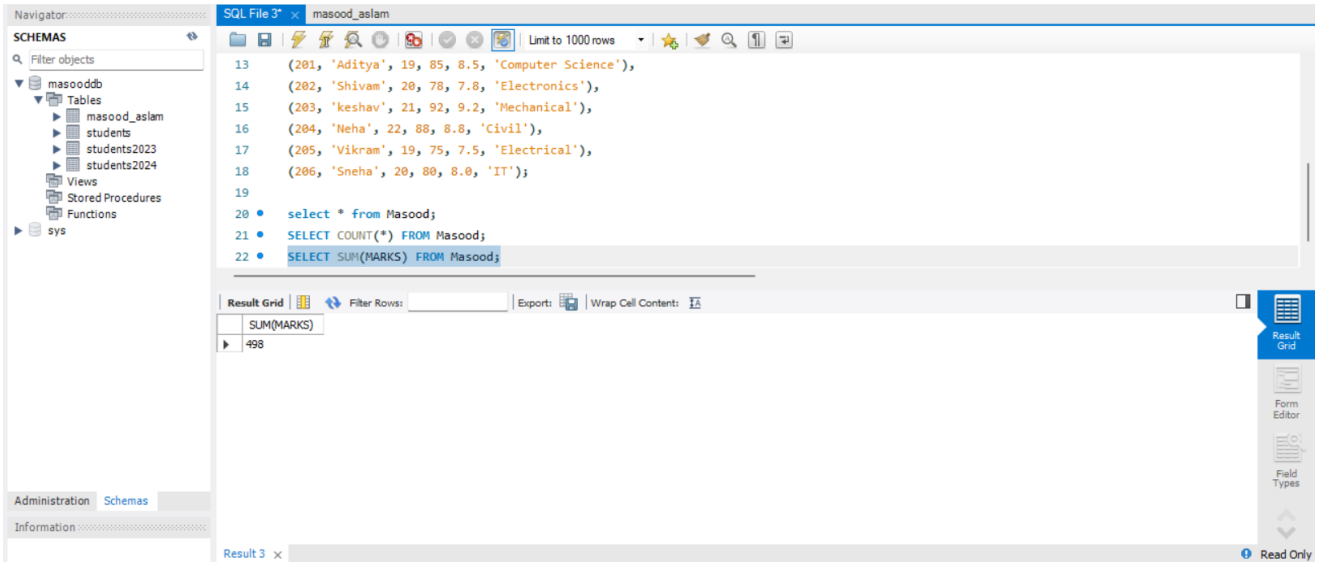
Inserting Sample Data



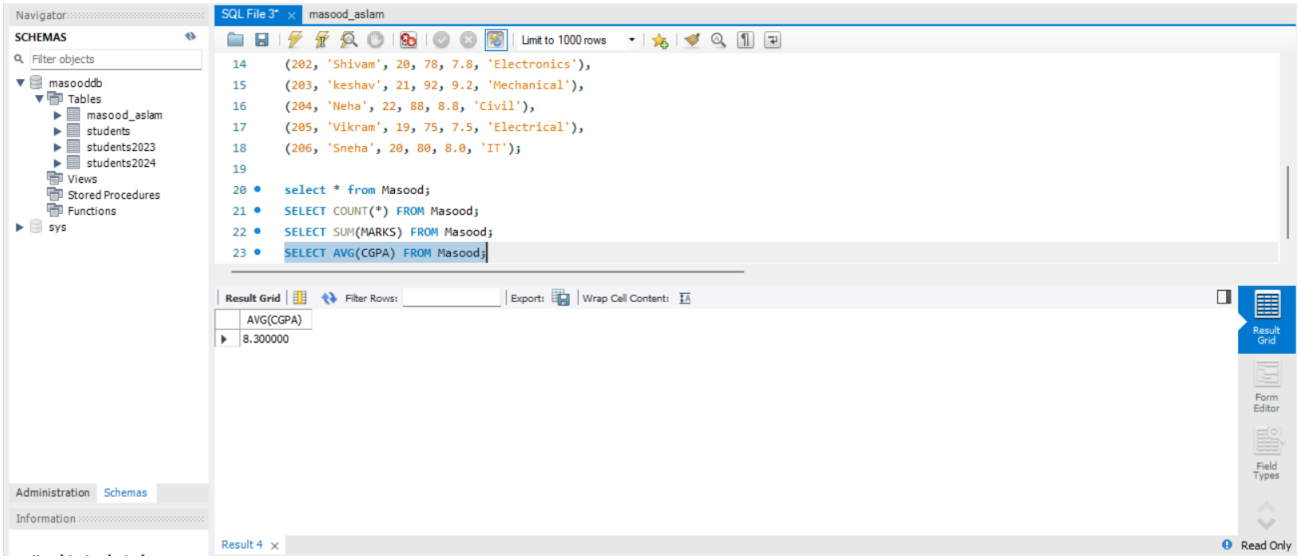
Count Total Rows



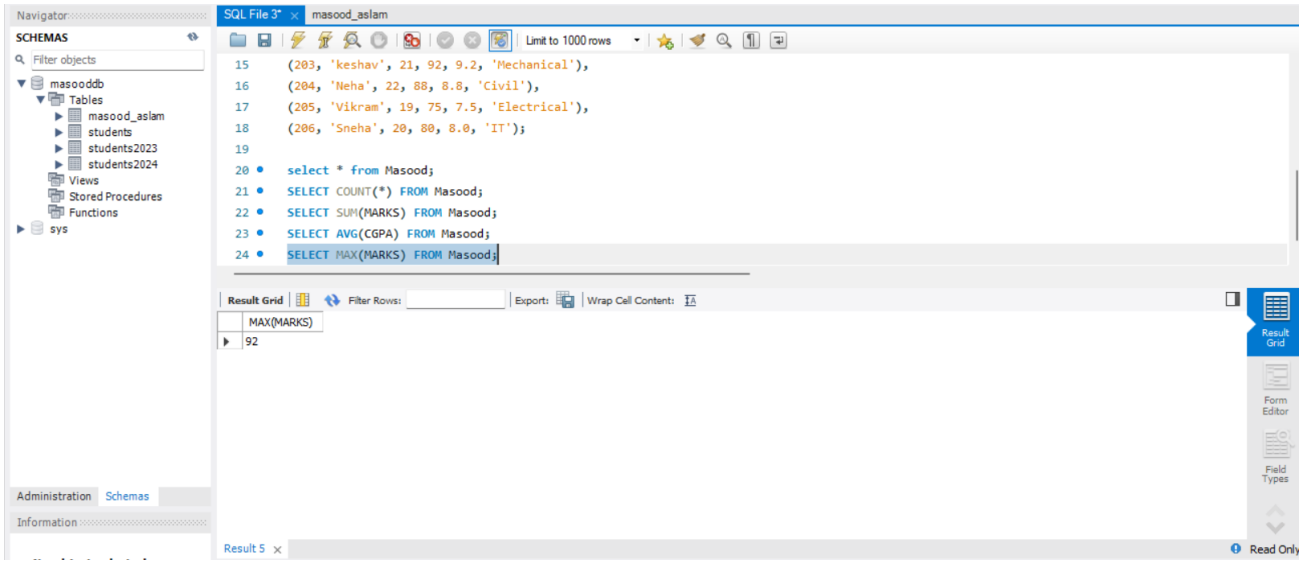
Find Sum of Values

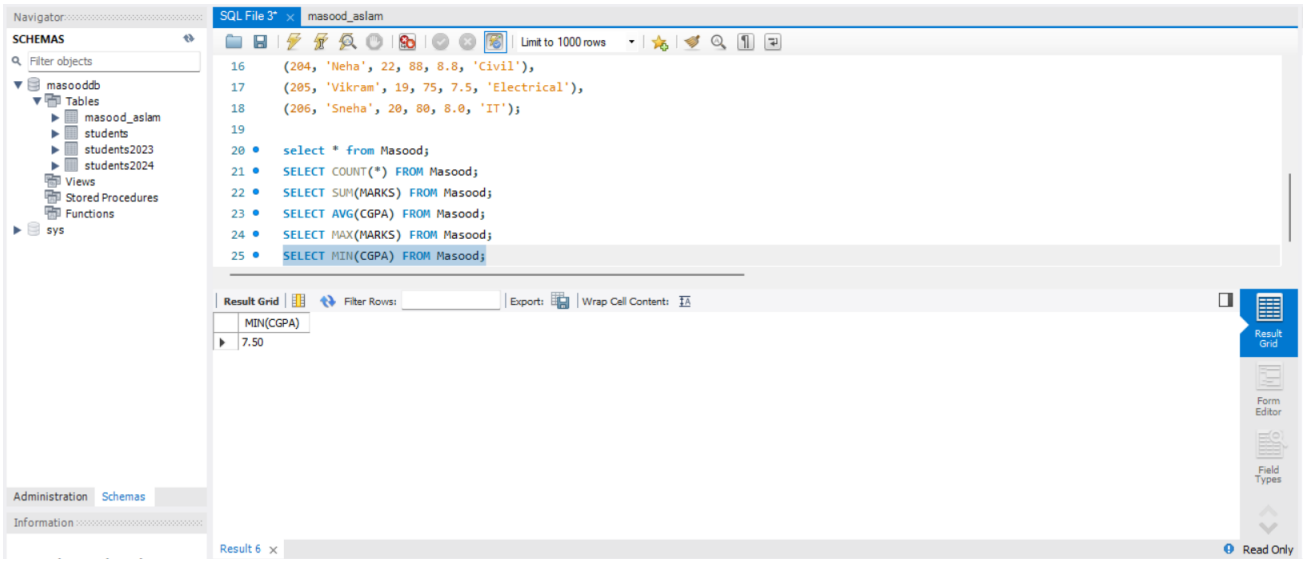


Calculate Average Value



Find Minimum & Maximum Values





EXPERIMENT 13

OBJECTIVE : Write a program to implement ORACLE functions in SQL.

SOFTWARE USED : MySQL Workbench 8.0 CE

THEORY

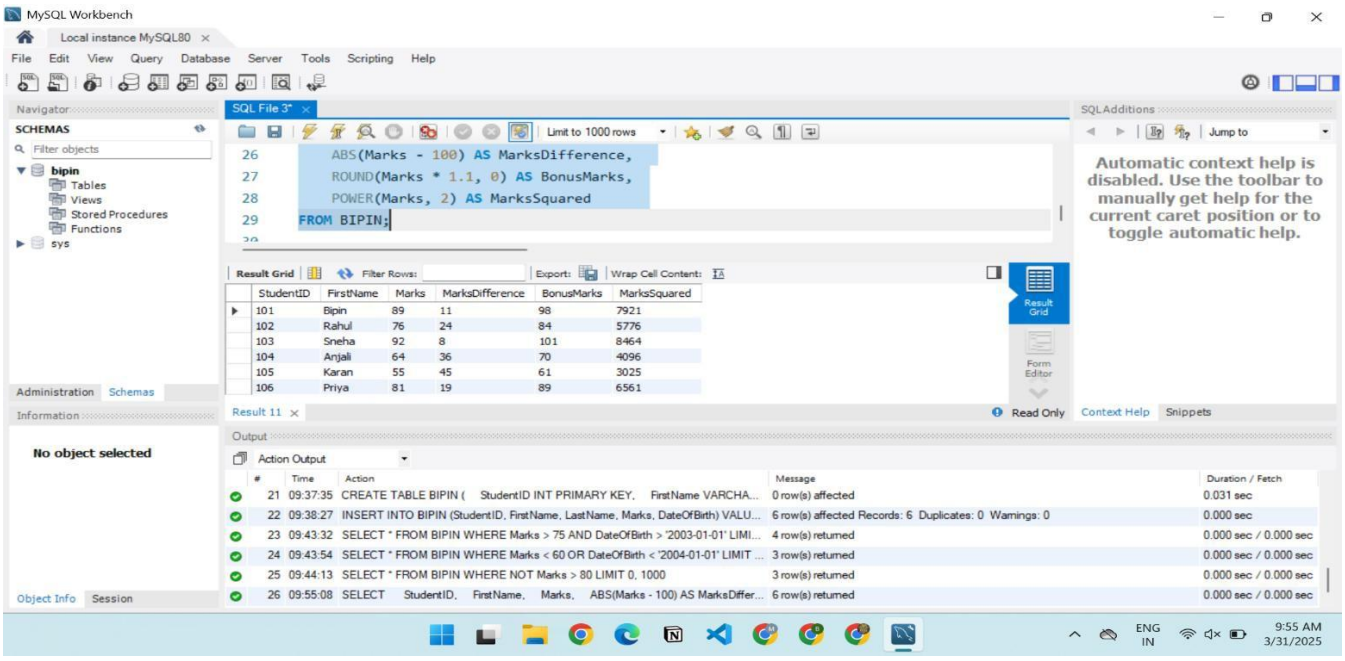
There are four types of single row functions:

1. NUMERIC FUNCTIONS: These are functions that accept numeric input and return numeric values.
2. CHARACTER FUNCTIONS: These are functions that accept character input and can return both character and number values.
3. DATE FUNCTIONS: These are that take values that are of data type DATE as input and return values of data type DATE.
4. CONVERSION FUNCTIONS: These are functions that help us to convert a value in one form to another datatype.

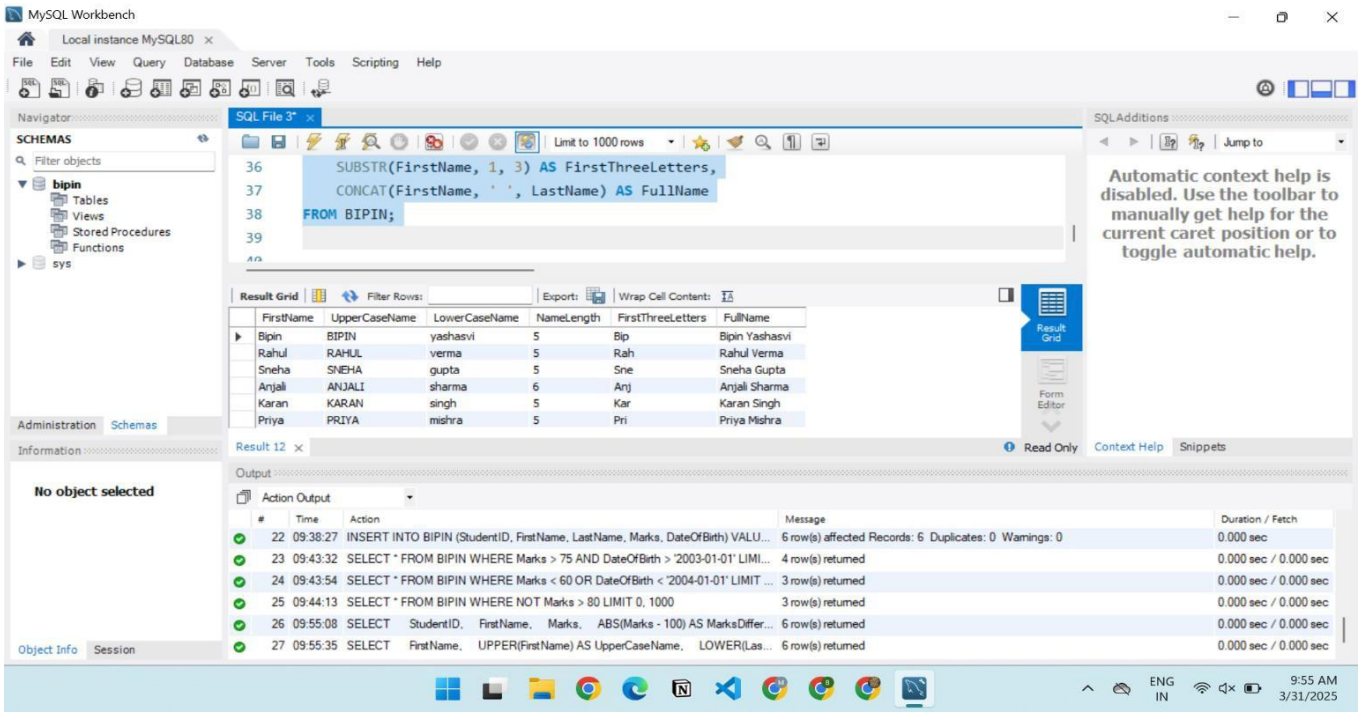
DUAL TABLE

This is a single row and single column dummy table provided by oracle. This is used to perform mathematical calculations without using a table.

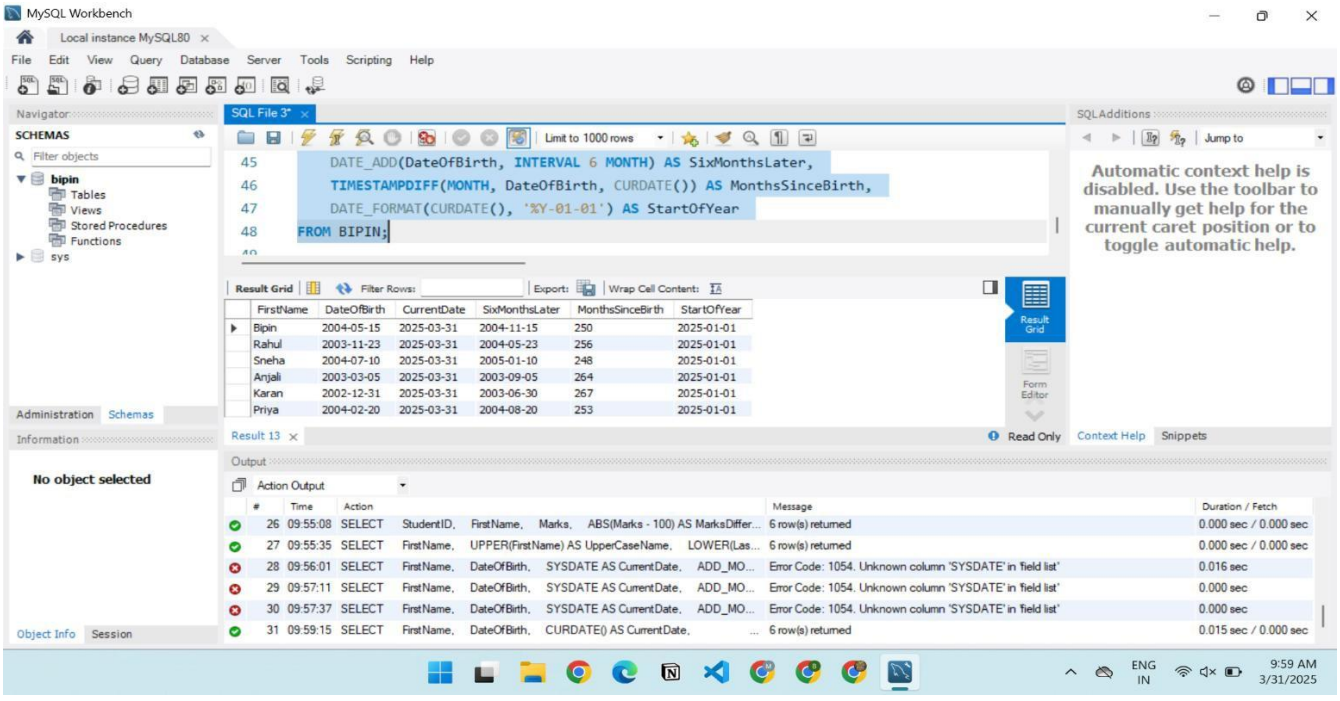
NUMERIC FUNCTION



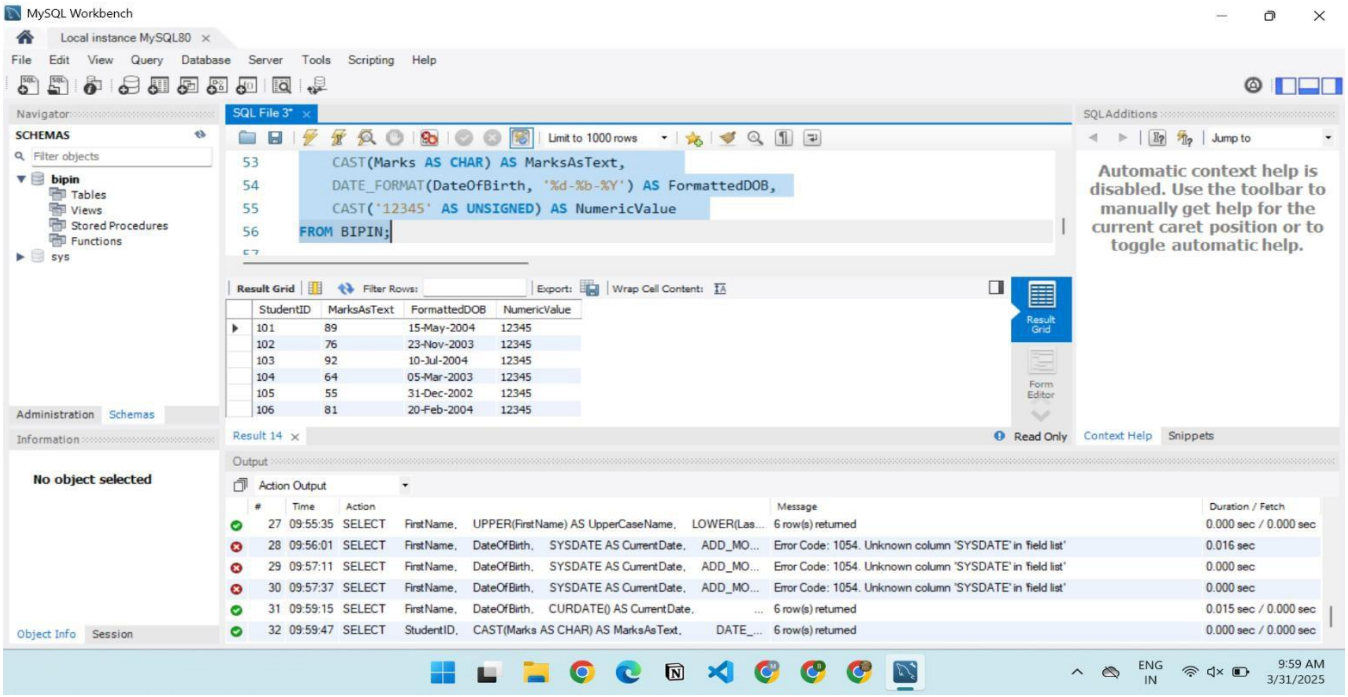
CHRACTER FUNCTION



DATE FUNCTION



CONVERSION FUNCUTION



EXPERIMENT 11

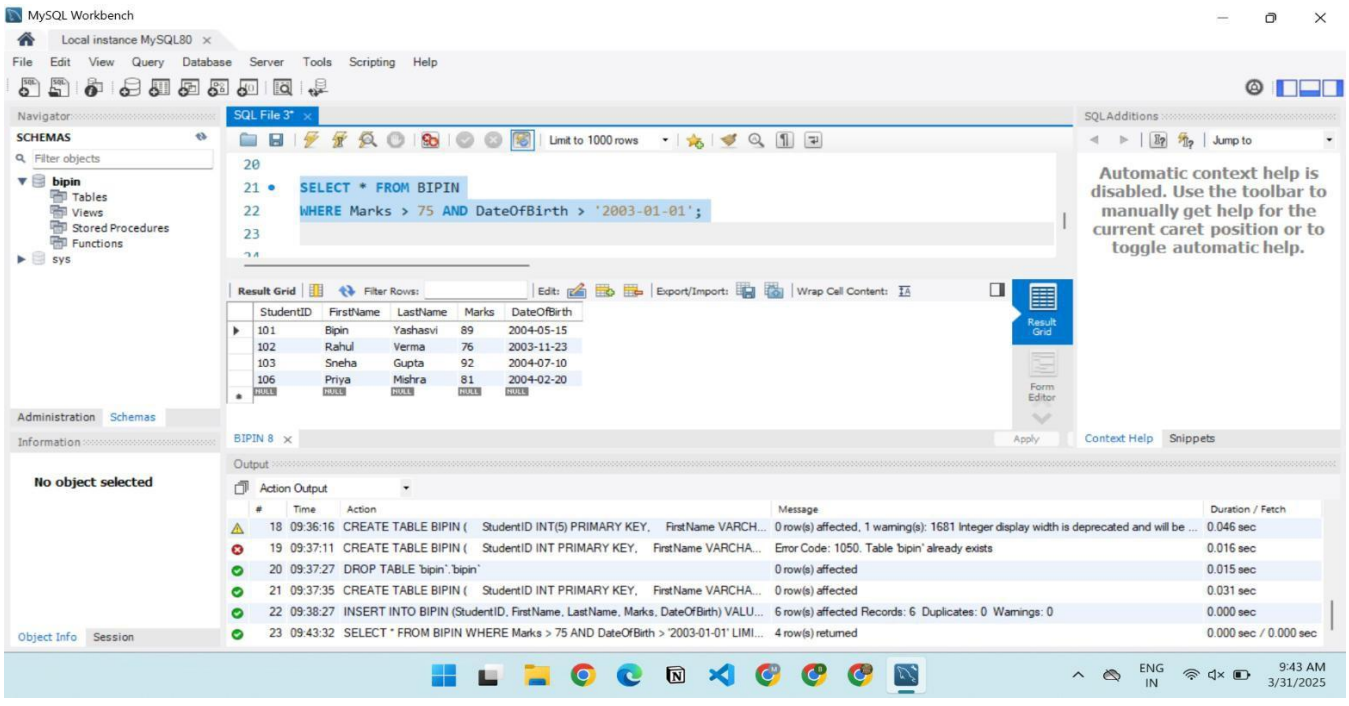
OBJECTIVE : Write a program to retrieve data using logical and arithmetic operators in SQL.

SOFTWARE USED : MySQL Workbench 8.0 CE

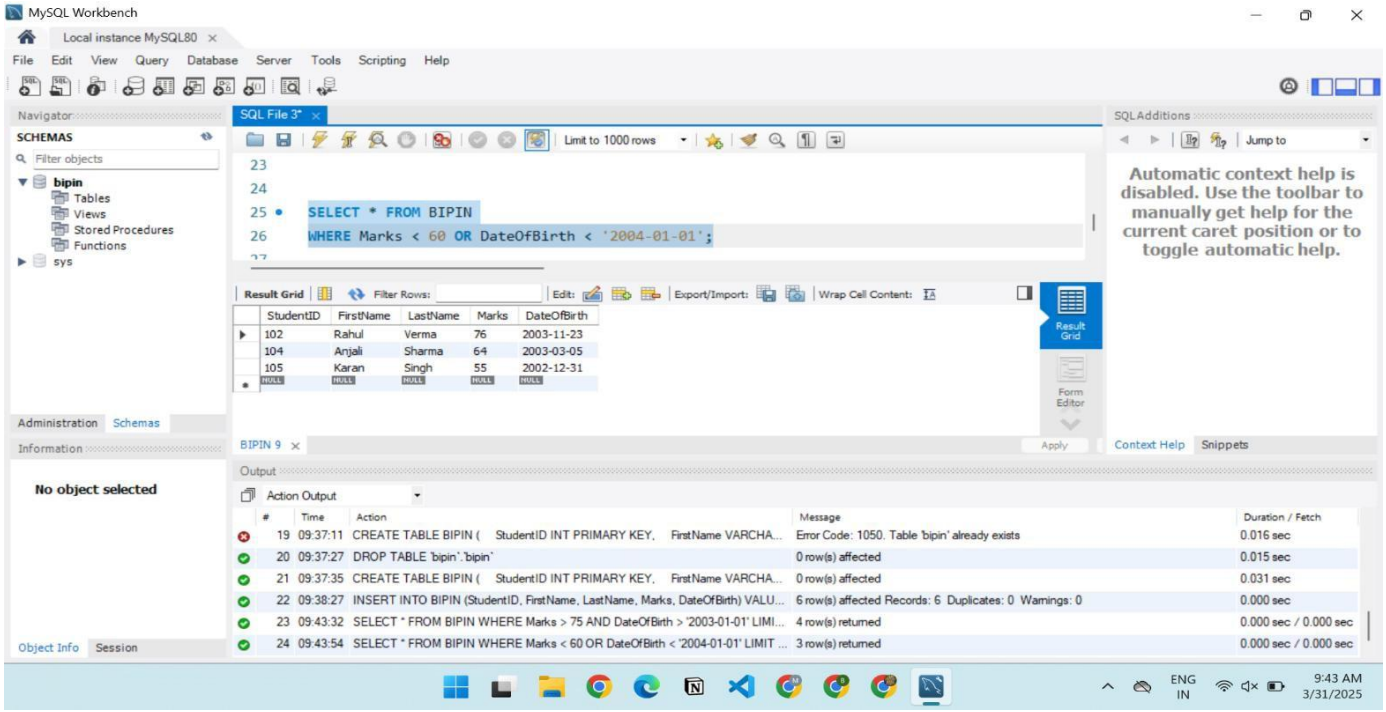
THEORY

SQL has many logical and arithmetic operators like AND, OR, NOT which can be used to retrieve specific data sets from a database.

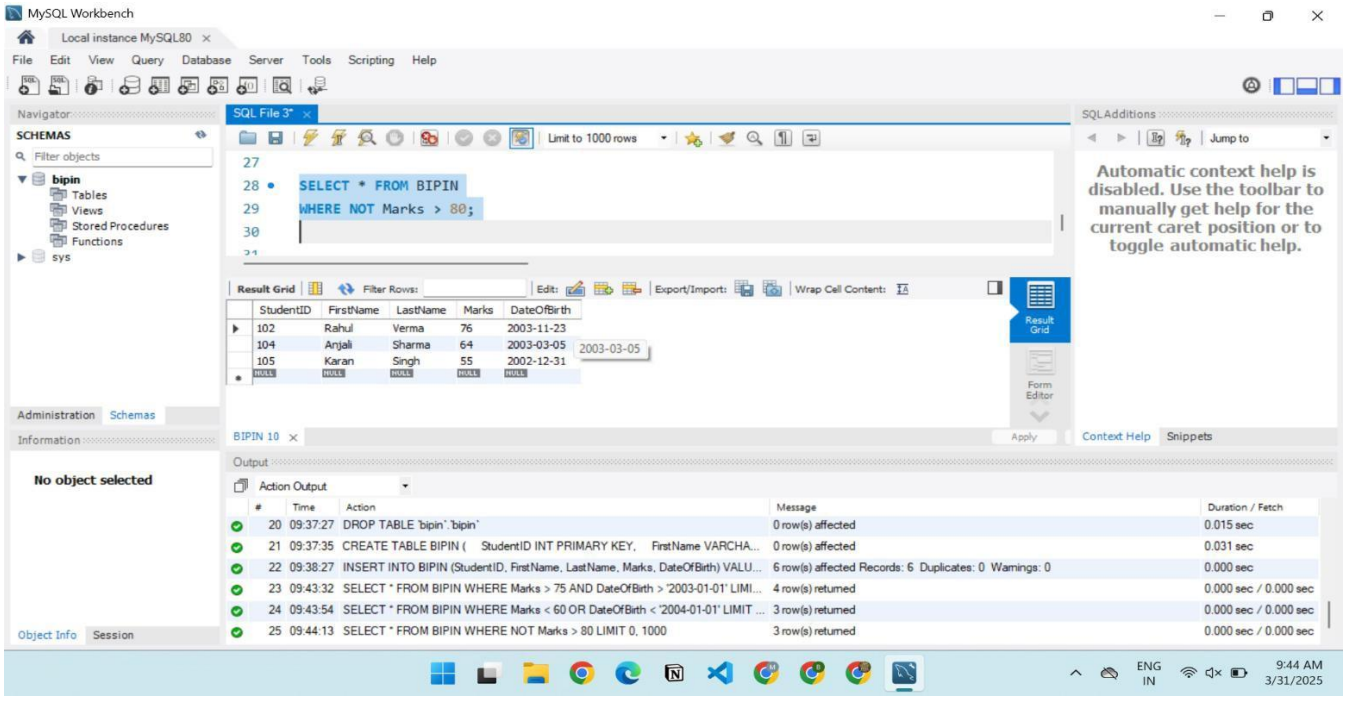
AND OPERATOR



OR OPERATOR



NOT OPERATOR

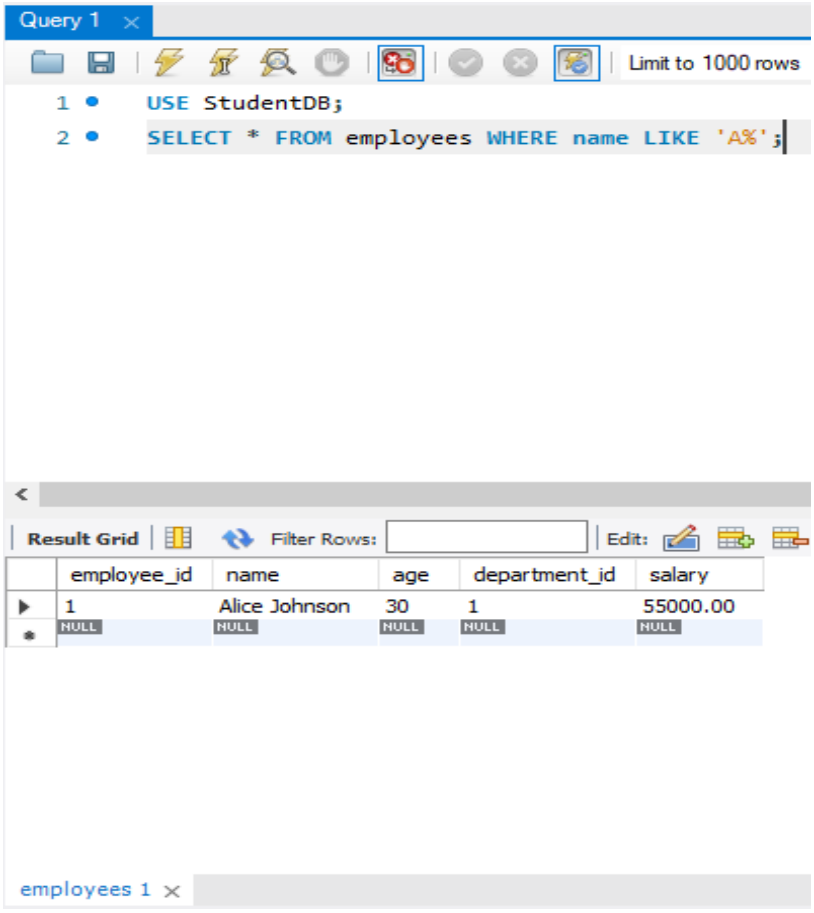


EXPERIMENT-14

Q: To implement Different operators(LIKE, BETWEEN, IN, WILDCARDS)

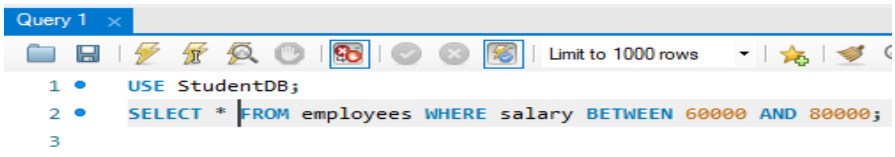
LIKE Operator:

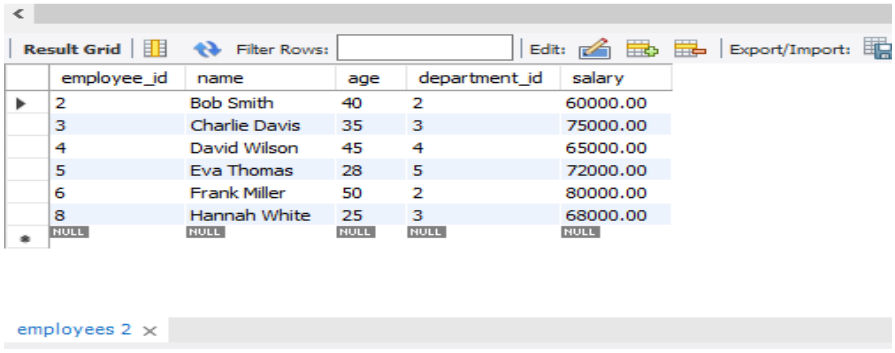
Find employees whose names start with the letter "A":



BETWEEN Operator:

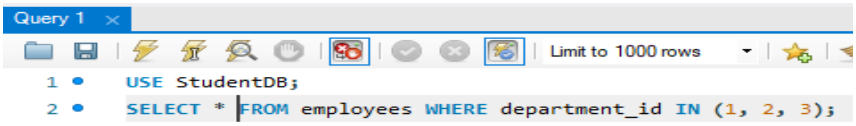
Find employees whose salary is between $60,000 and $80,000:

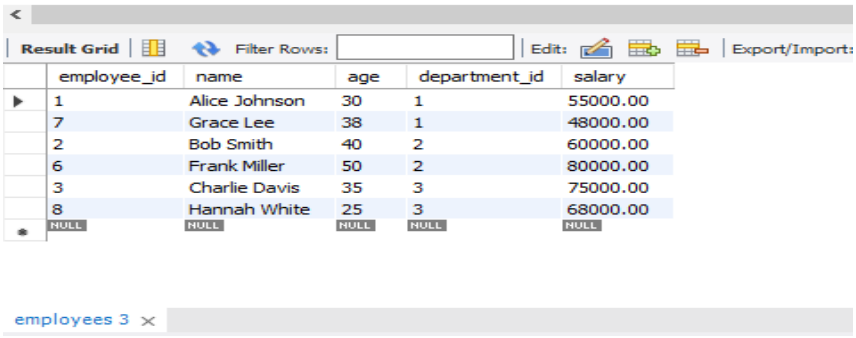




IN Operator

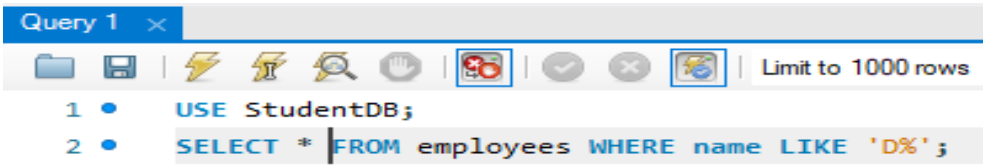
Find employees who work in the "HR", "Sales", or "IT" departments:

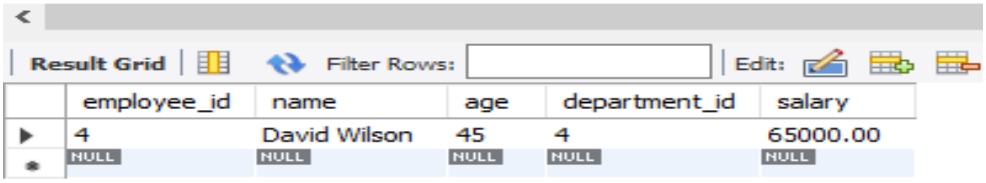




Wildcards:

Find employees whose names start with "D" and have any characters after it (wildcard %):





16. Introduction of VIEW. How to create view. Explain with example.

In the context of databases, a view is a virtual table that consists of a subset of data from one or more tables. Views are useful for various purposes, such as simplifying complex queries, providing a layer of security by restricting access to certain columns or rows, and presenting data in a particular format for reporting or analysis.

Creating a view in a typical MySQL database -

```CREATE VIEW view_name AS

SELECT column1, column2,

FROM table_name

WHERE condition;```

```
mysql> select * from new;
+----+-------------+-----+
| id | name | age |
+----+-------------+-----+
1	akash_rattan	56
2	oggy	30
3	jack	10
4	panther	5
+----+-------------+-----+
4 rows in set (0.00 sec)

mysql> create view second as select name,age from new where age>30;
Query OK, 0 rows affected (0.02 sec)

mysql> select * from second;
+-------------+-----+
| name | age |
+-------------+-----+
| akash_rattan | 56 |
+-------------+-----+
1 row in set (0.00 sec)
```
```

Exercise-1

Sales Table:

```
CREATE TABLE Sales (
sale_id INT PRIMARY KEY,
product_id INT,
quantity_sold INT,
sale_date DATE,
total_ price DECIMAL(10, 2)
);
```

Products Table:

```
CREATE TABLE Products (
product_id INT PRIMARY KEY,
product_name VARCHAR(100),
category VARCHAR(50),
unit_price DECIMAL(10, 2)
);
```

Questions:

1. Retrieve all columns from the Sales table.

2. Retrieve the product name and unit_ price from the Products table.

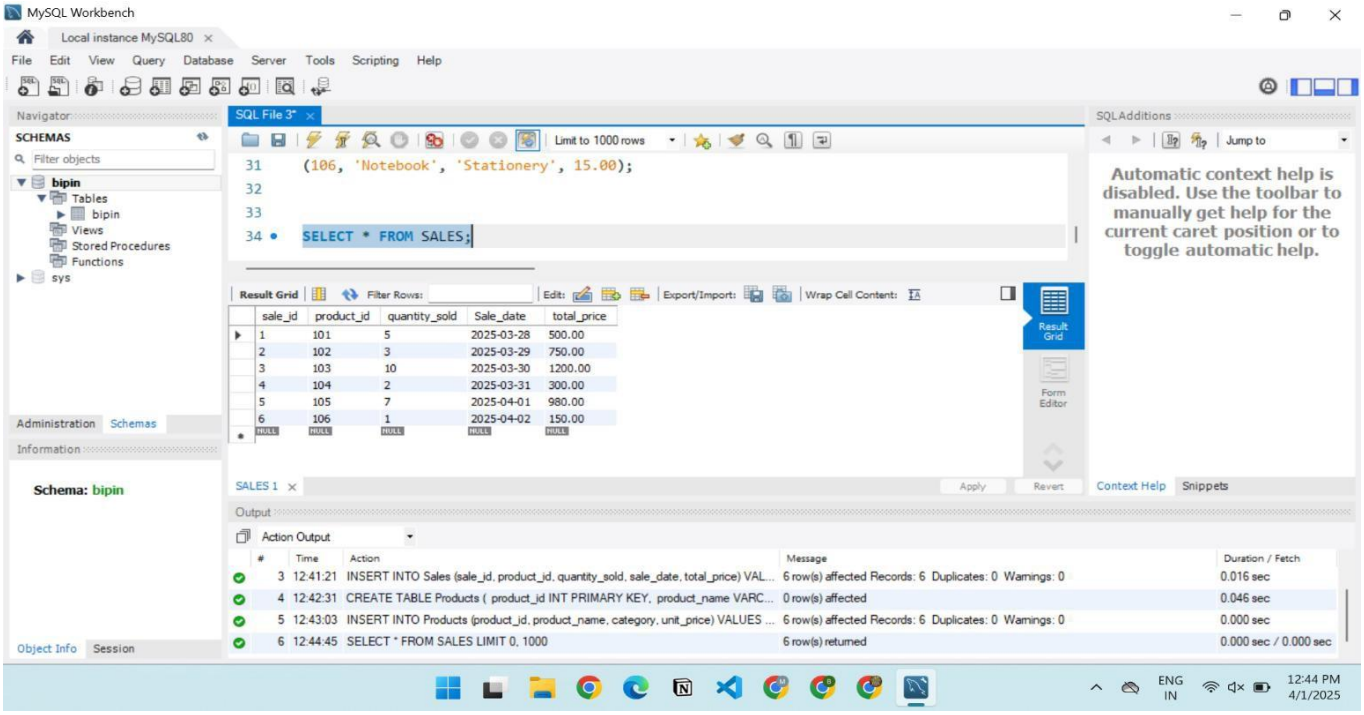
3. Filter the Sales table to show only sales with a total price greater than $100.

4. Filter the Products table to show only products in the 'Electronics' category.

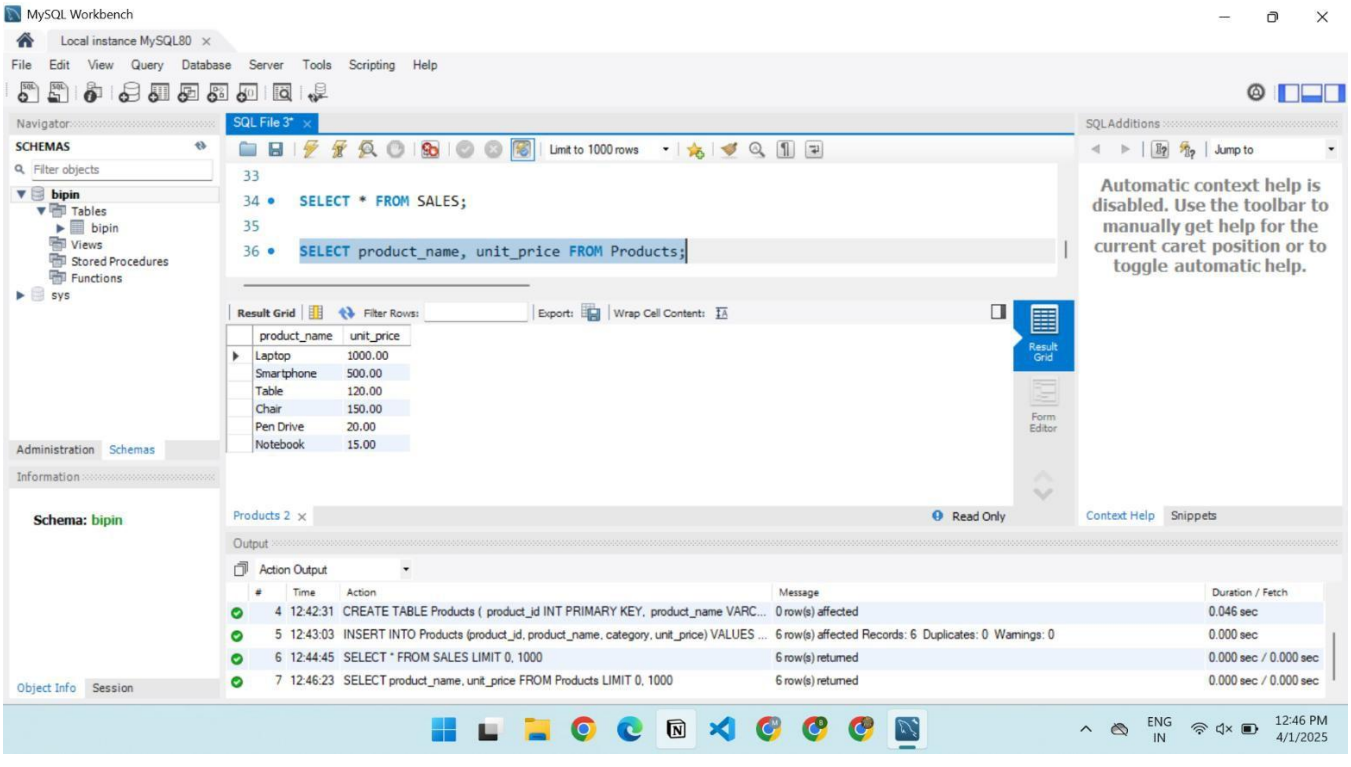
5. Retrieve the sale_ id and total _price from the Sales table for sales made on January 3, 2024.

6. Retrieve the product_id and product name from the Products table for products with a unit price greater than $100.

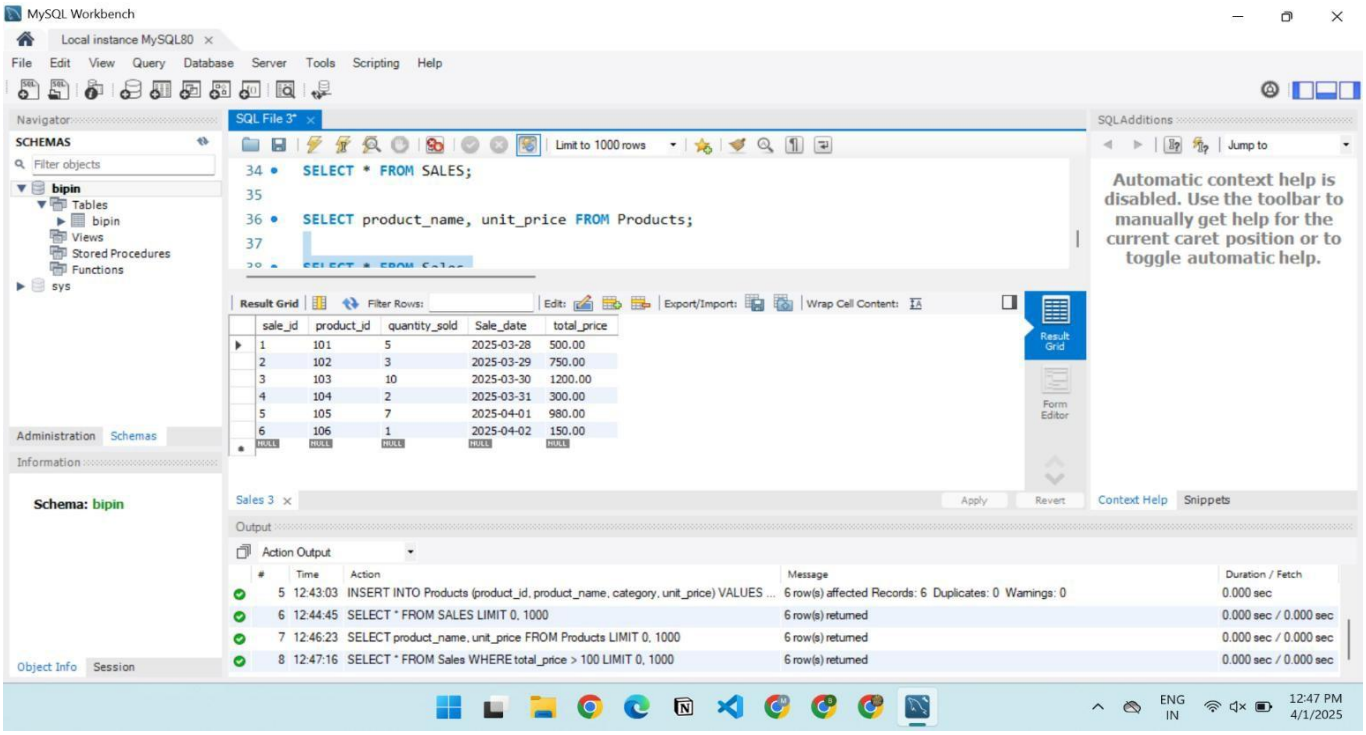
1. **Retrieve all columns from the Sales table.**



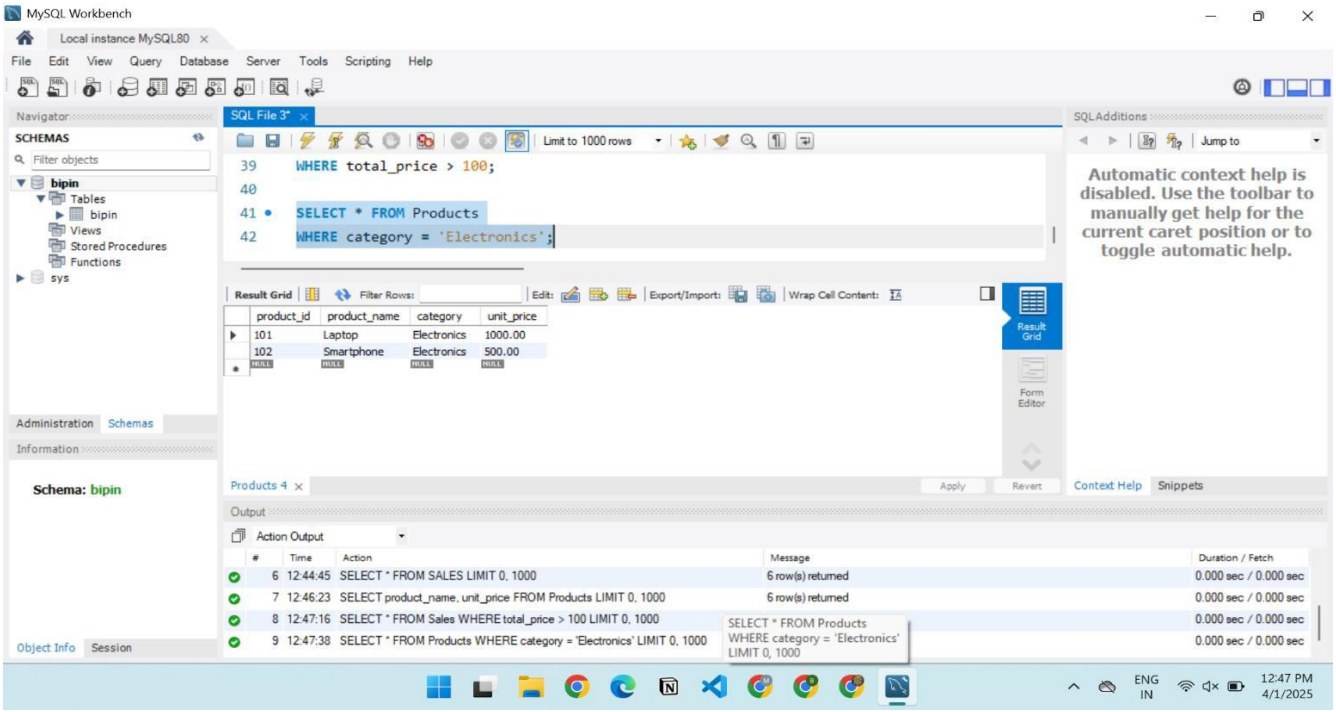
2. **Retrieve the product_name and unit_ price from the Products table.**



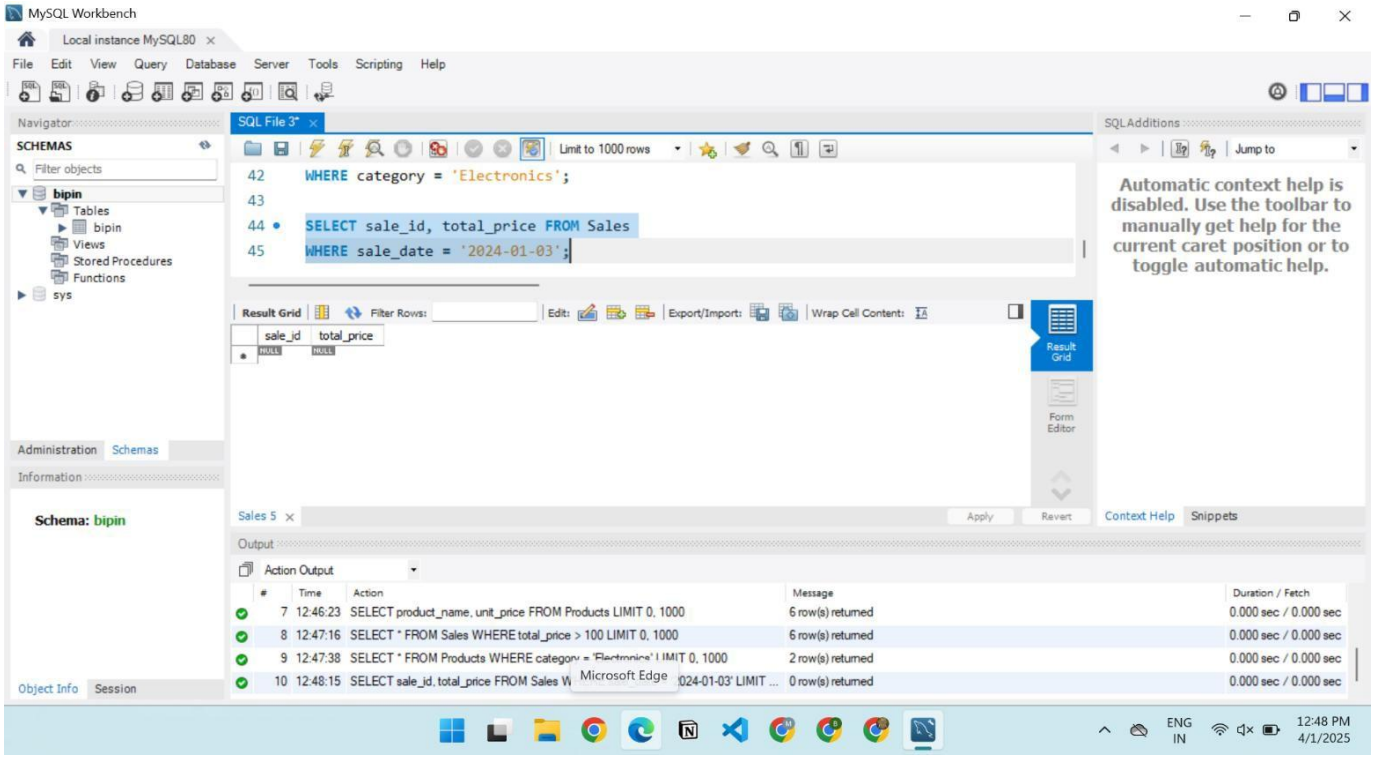
3. Filter the Sales table to show only sales with a total price greater than $100.



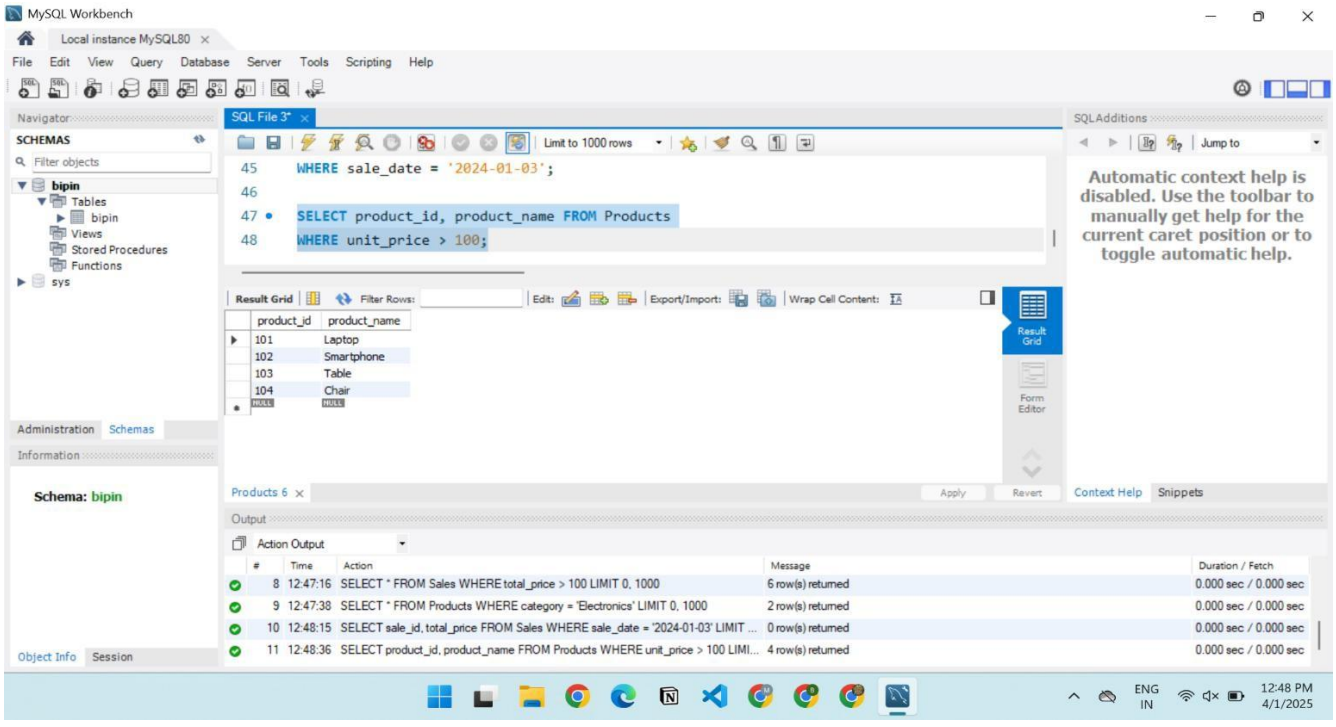
4. Filter the Products table to show only products in the 'Electronics' category.



5. Retrieve the sale_ id and total_price from the Sales table for sales made on January 3, 2024.



6. Retrieve the product_id and product name from the Products table for products with a unit price greater than $100.



SQL Exercise-2

EmployeeInfo Table:

EmpID	EmpFname	EmpLname	Department	Project	Address	DOB	Gender
1	Sanjay	Mehra	HR	P1	Hyderabad(HYD)	01/12/1976	M
2	Ananya	Mishra	Admin	P2	Delhi(DEL)	02/05/1968	F
3	Rohan	Diwan	Account	P3	Mumbai(BOM)	01/01/1980	M
4	Sonia	Kulkarni	HR	P1	Hyderabad(HYD)	02/05/1992	F
5	Ankit	Kapoor	Admin	P2	Delhi(DEL)	03/07/1994	M

EmployeePosition Table:

EmpID	EmpPosition	DateOfJoining	Salary
1	Manager	01/05/2024	500000
2	Executive	02/05/2024	75000
3	Manager	01/05/2024	90000
2	Lead	02/05/2024	85000
1	Executive	01/05/2024	300000

Consider the above-mentioned tables, write the answer of following SQL queries:

1. Write a query to fetch the EmpFname from the EmployeeInfo table in the upper case and use the ALIAS name as EmpName.
2. Write a query to fetch the number of employees working in the department ‘HR’.
3. Write a query to get the current date.
4. Write a query to retrieve the first four characters of EmpLname from the EmployeeInfo table.
5. Write a query to fetch only the place name (string before brackets) from the Address column of EmployeeInfo table.
6. Write a query to create a new table that consists of data and structure copied from the other table.
7. Write q query to find all the employees whose salary is between 50000 to 100000.
8. Write a query to find the names of employees that begin with ‘S’
9. Write a query to fetch top N records.
10. Write a query to retrieve the EmpFname and EmpLname in a single column as “FullName”. The first name and the last name must be separated with space.

Exercise-2

EmployeeInfo Table:

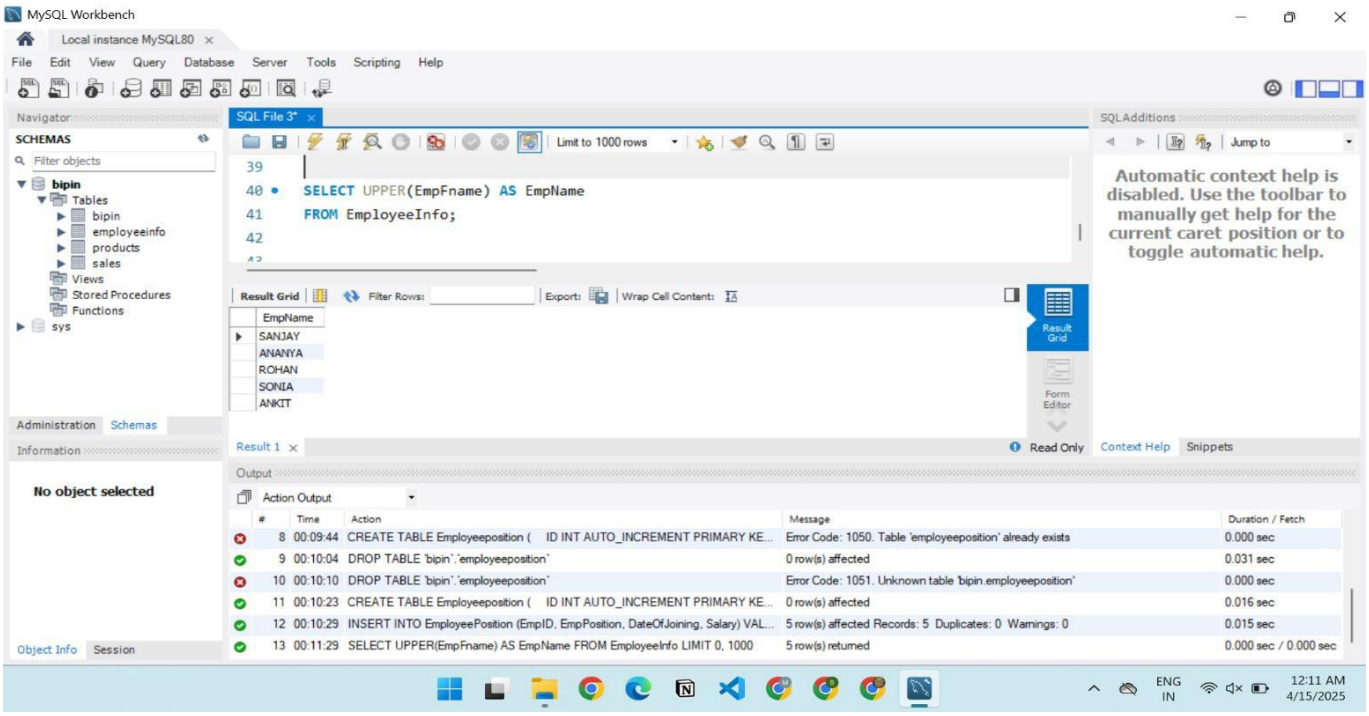
```
CREATE TABLE EmployeeInfo (
    EmpID INT PRIMARY KEY,
    EmpFname VARCHAR(50),
    EmpLname VARCHAR(50),
    Department VARCHAR(50),
    Project VARCHAR(50),
    Address VARCHAR(100),
    DOB DATE,
    Gender VARCHAR(10)
);
```

EmployeePosition Table:

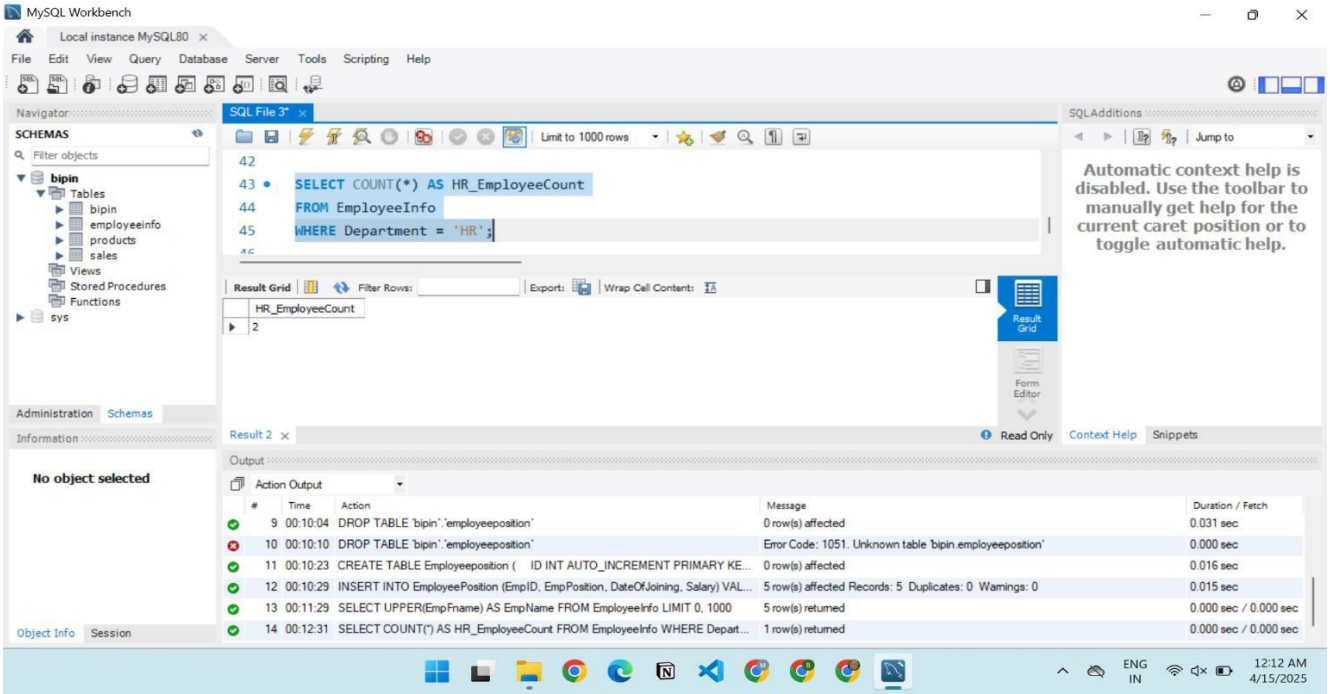
```
CREATE TABLE Employeeposition (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    EmpID INT,
    EmpPosition VARCHAR(50),
    DateOfJoining DATE,
    Salary DECIMAL(10, 2)
);
```

Questions:

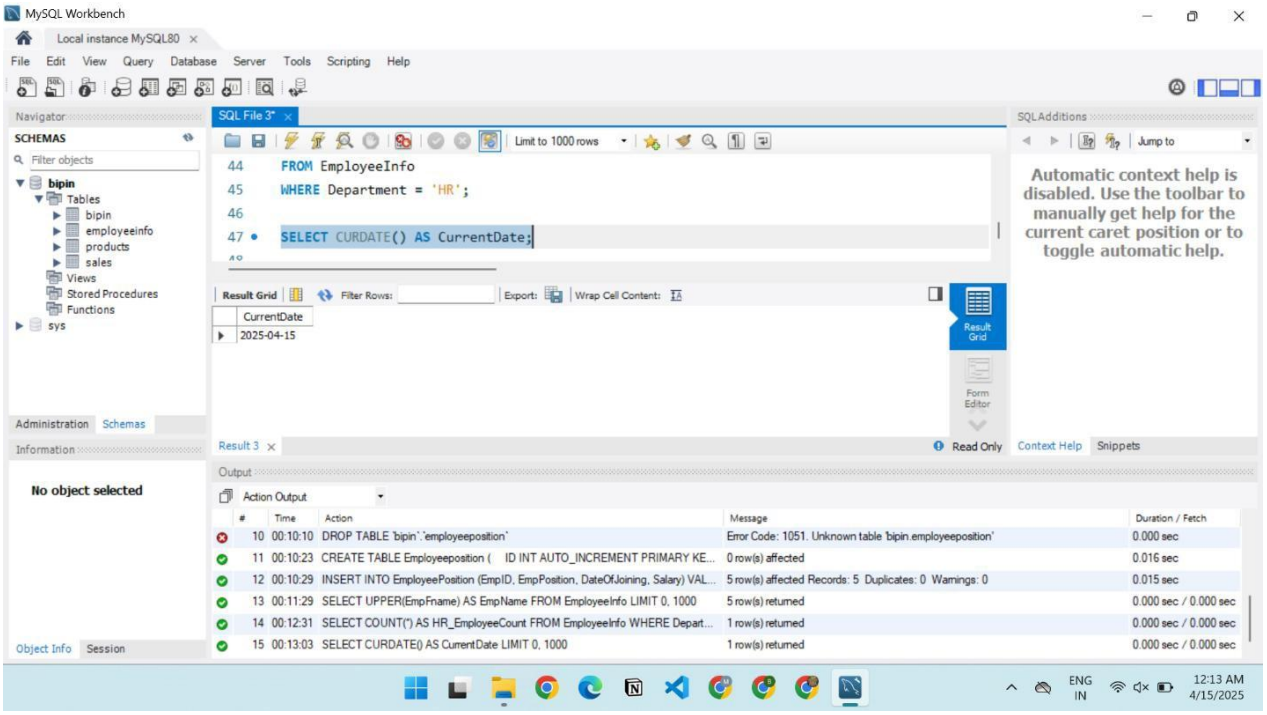
1. Write a query to fetch the EmpFname from the EmployeeInfo table in the upper case and use the ALIAS name as EmpName.



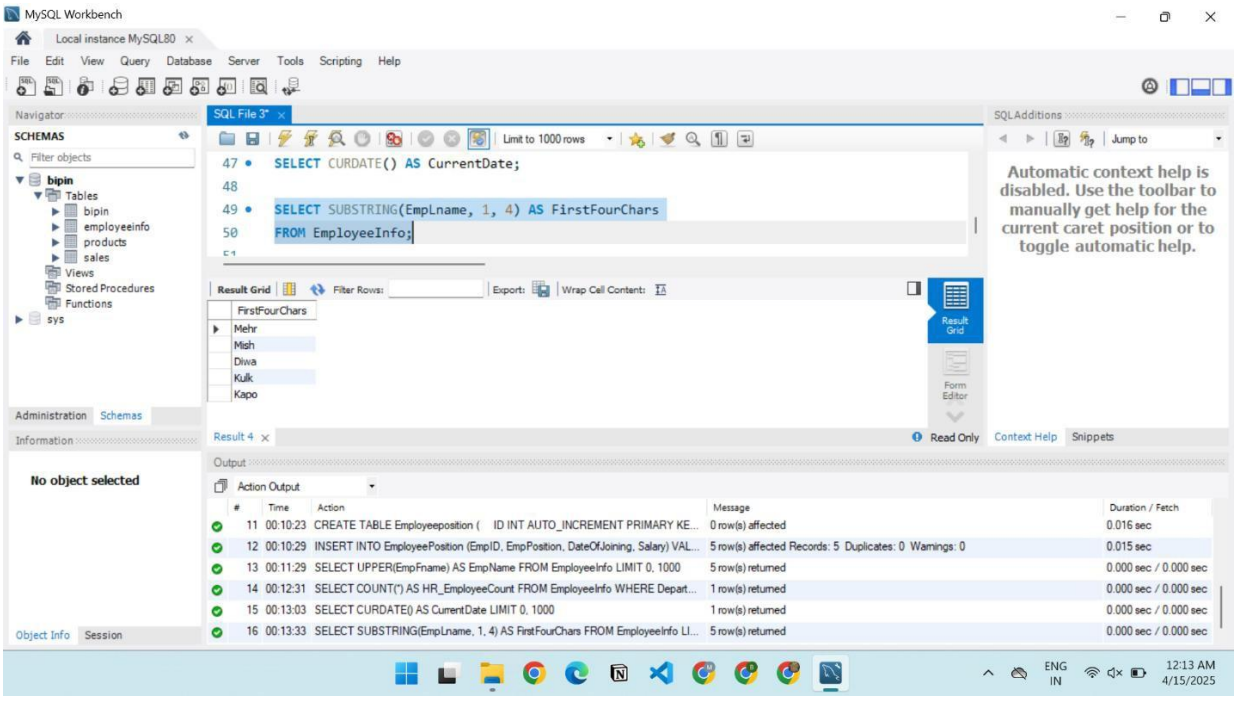
2. Write a query to fetch the number of employees working in the department 'HR'.



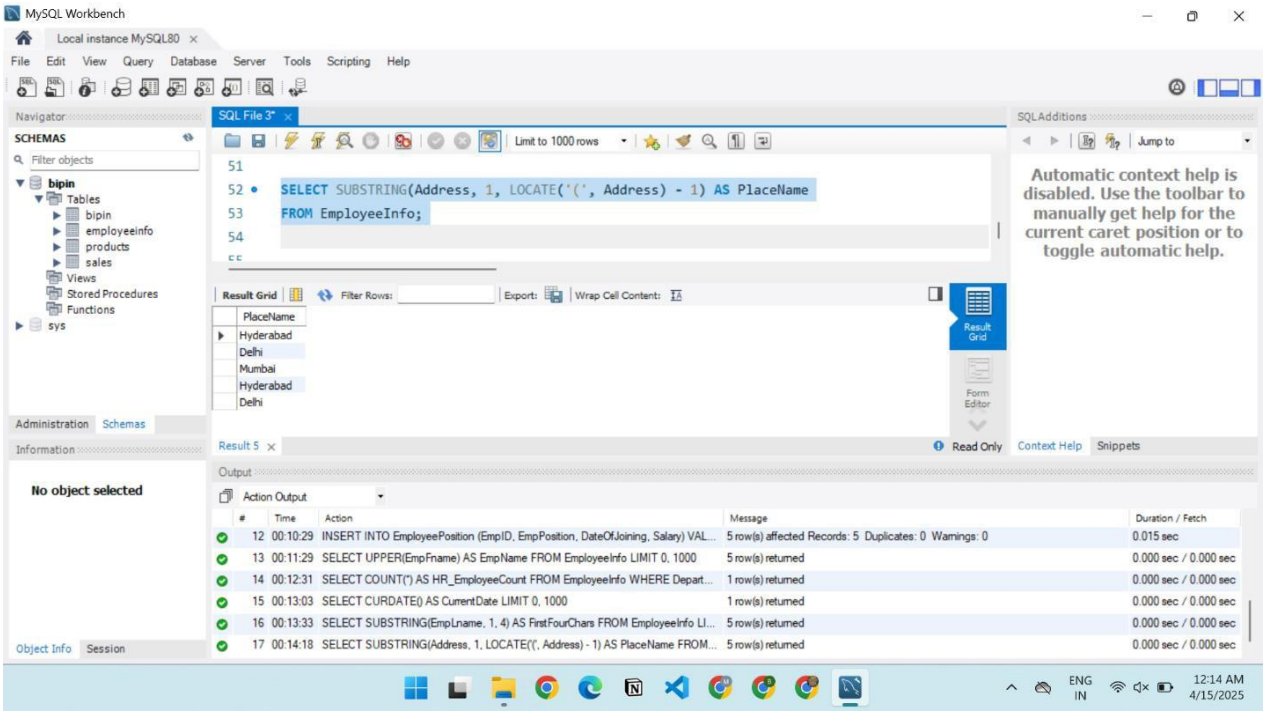
3. Write a query to get the current date.



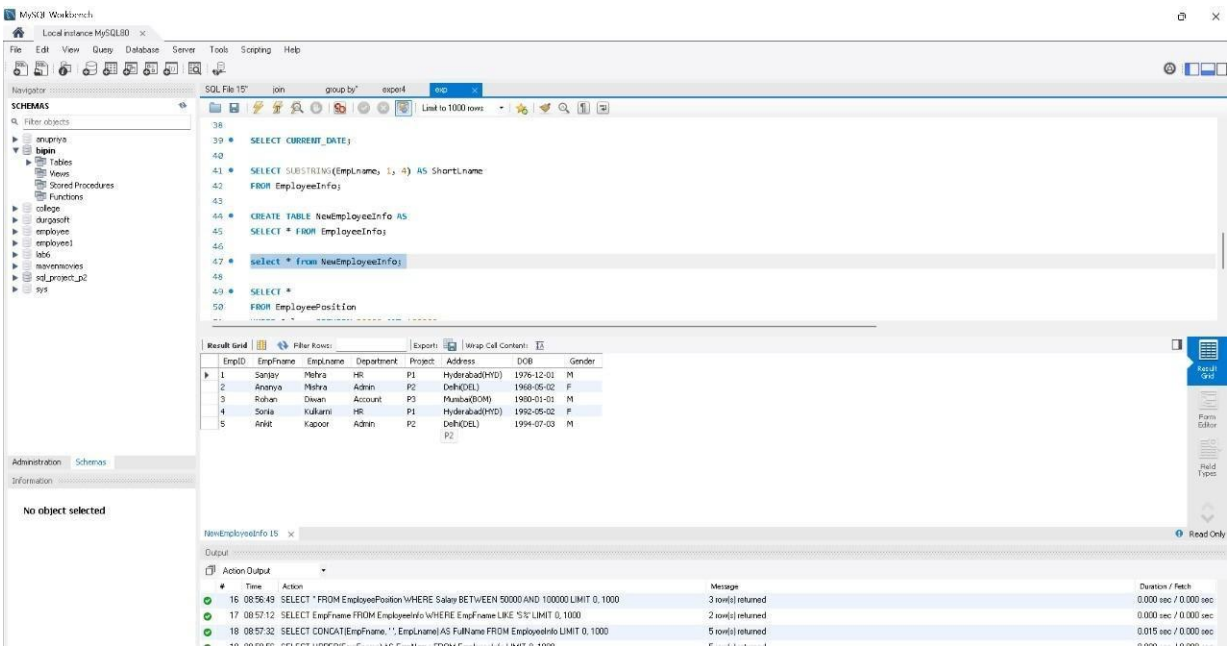
4. Write a query to retrieve the first four characters of EmpLname from the EmployeeInfo table.



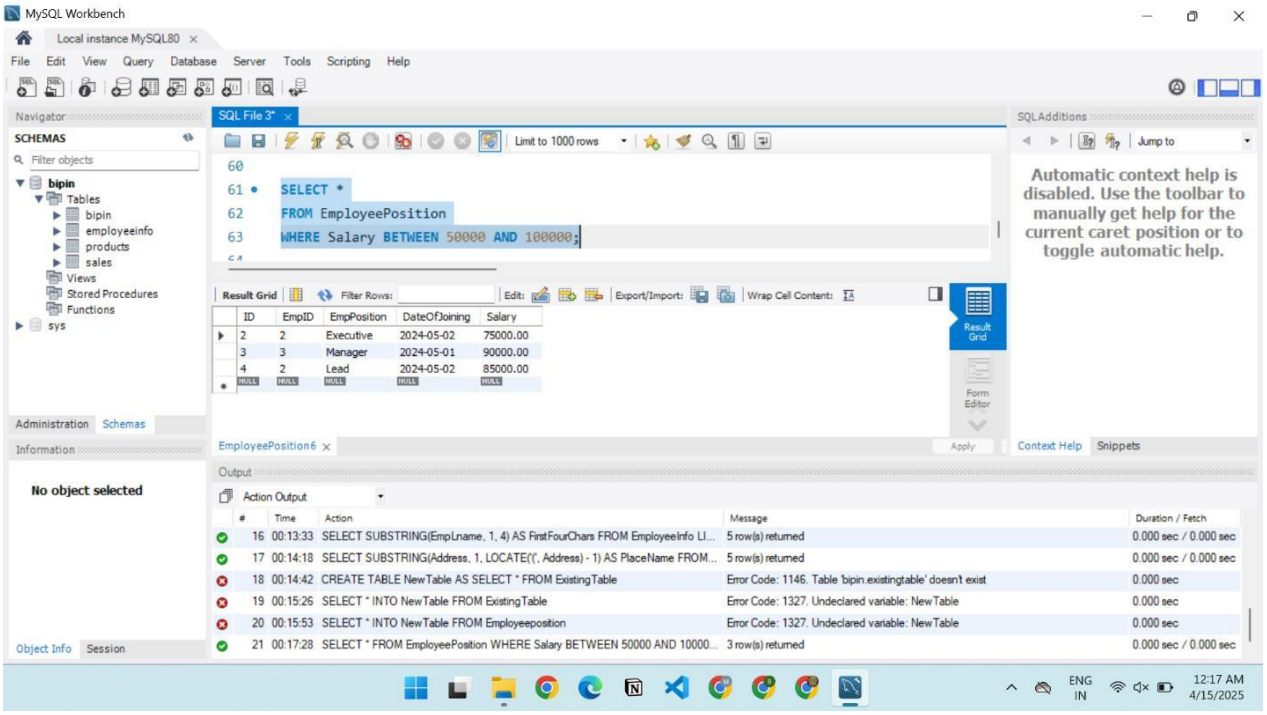
5. Write a query to fetch only the place name (string before brackets) from the Address column of EmployeeInfo table.



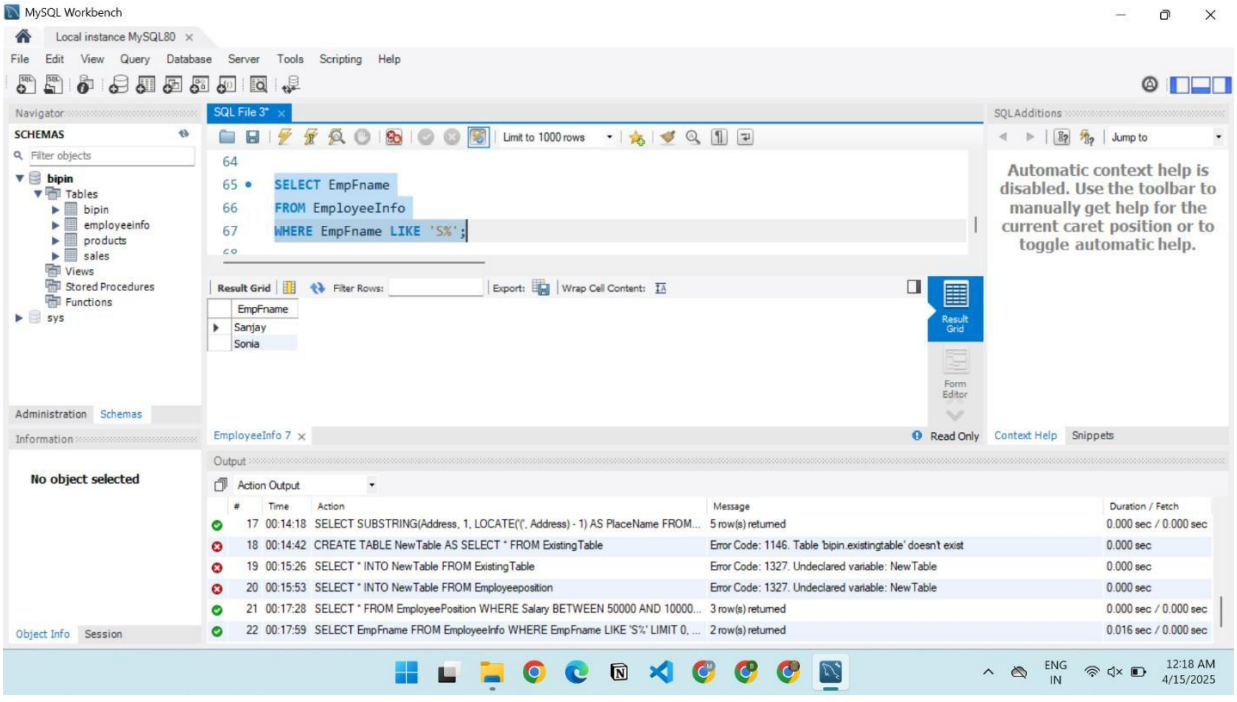
6. Write a query to create a new table that consists of data and structure copied from the other table.



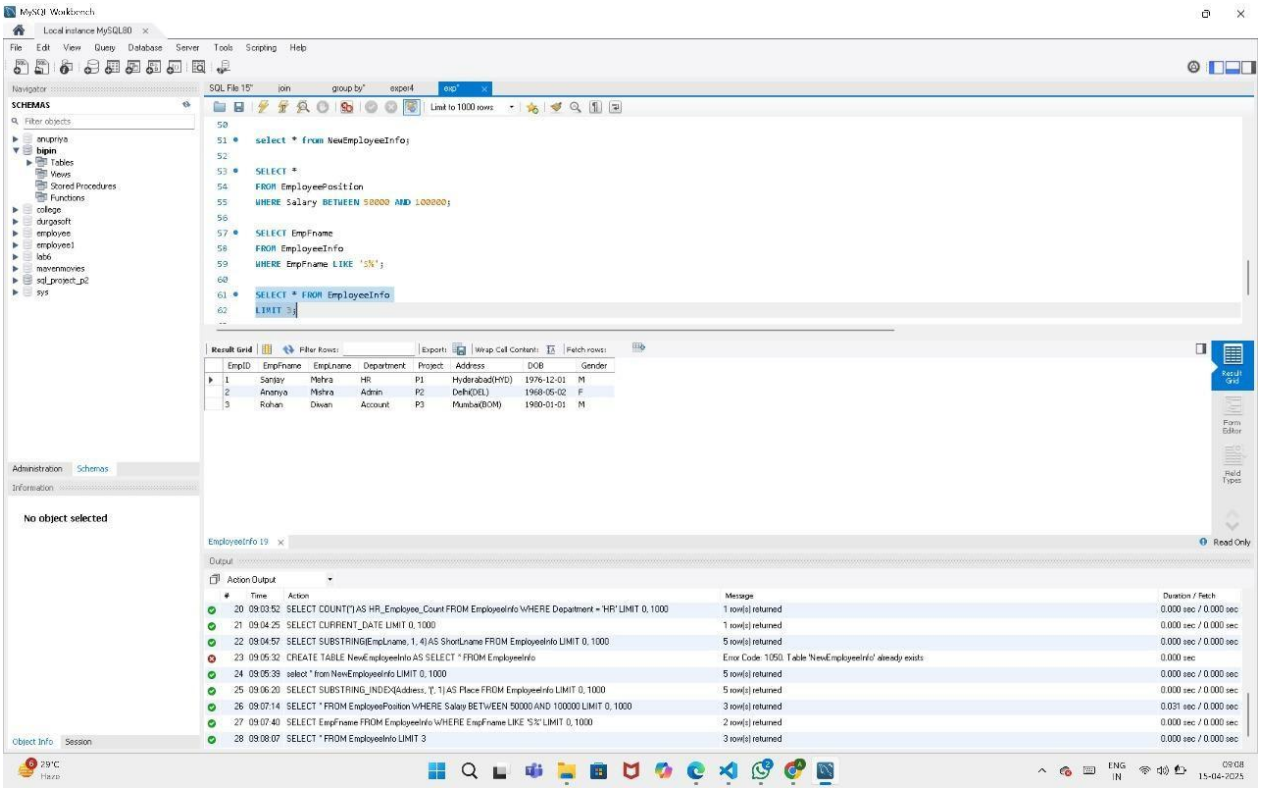
7. Write q query to find all the employees whose salary is between 50000 to 100000.



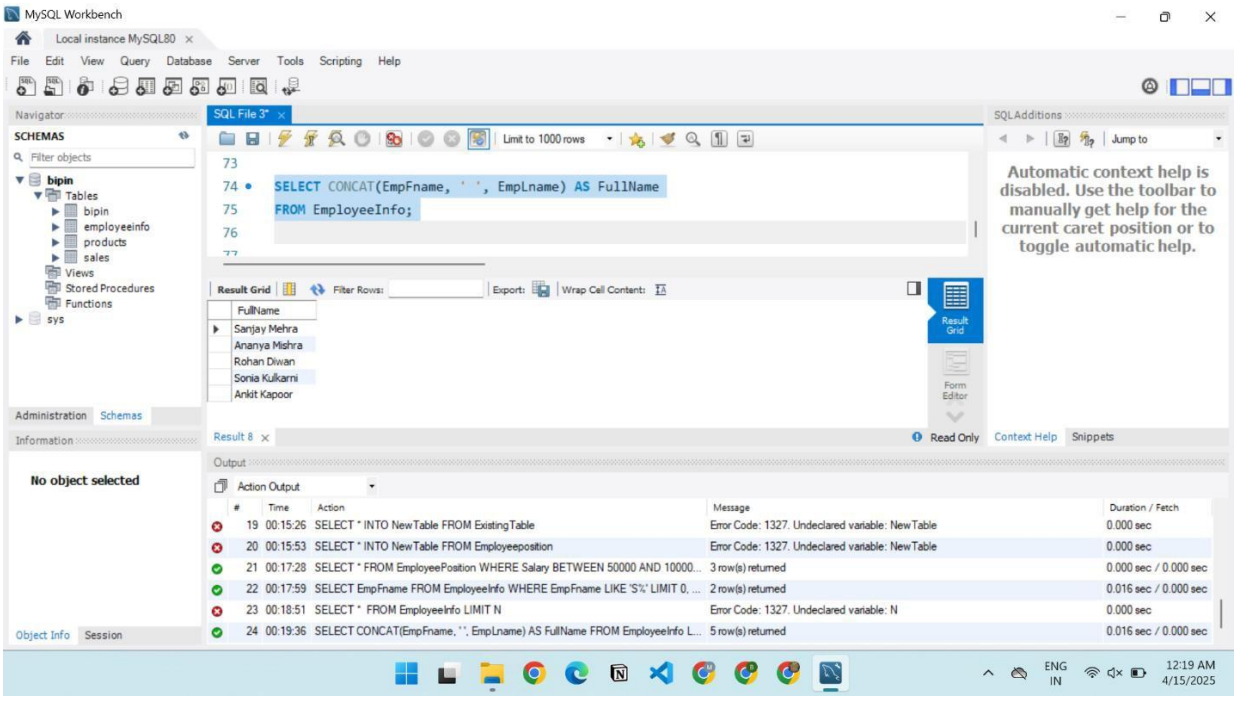
8. Write a query to find the names of employees that begin with 'S'.



9. Write a query to fetch top N records.



10. Write a query to retrieve the EmpFname and EmpLname in a single column as "FullName". The first name and the last name must be separated with space.



8.To implement Joins (inner, outer, natural, equi join) and related example.

Joins in SQL are used to combine rows from two or more tables based on a related column between them. Here are the common types of joins:

```
mysql> select * from a1;
+------+--------------+
| id   | name         |
+------+--------------+
|    1 | akash_rattan |
|    2 | a            |
|    3 | b            |
|    4 | c            |
+------+--------------+
4 rows in set (0.00 sec)

mysql> select * from a2;
+------+---------+
| id   | city    |
+------+---------+
|    1 | noida   |
|    2 | delhi   |
|    3 | lucknow |
+------+---------+
```

Inner Join

An inner join returns rows when there is at least one match in both tables.

```
mysql> select a1.name,a2.city from a1 inner join a2 on a1.id=a2.id;
+--------------+---------+
| name         | city    |
+--------------+---------+
| akash_rattan | noida   |
| a            | delhi   |
| b            | lucknow |
+--------------+---------+
```

Left Outer Join

A left outer join returns all rows from the left table (table1), and the matched rows from the right table (table2). The result is NULL from the right side, if there is no match.

```
mysql> select a1.name,a2.city from a1 left join a2 on a1.id=a2.id;
+--------------+---------+
| name         | city    |
+--------------+---------+
| akash_rattan | noida   |
| a            | delhi   |
| b            | lucknow |
| c            | NULL    |
+--------------+---------+
```

Right Outer Join

A right outer join returns all rows from the right table (table2), and the matched rows from the left table (table1). The result is NULL from the left side, when there is no match.

```
mysql> select a1.name,a2.city from a1 right join a2 on a1.id=a2.id;
+--------------+---------+
| name         | city    |
+--------------+---------+
| akash_rattan | noida   |
| a            | delhi   |
| b            | lucknow |
+--------------+---------+
```

Full Outer Join

A full outer join returns all records when there is a match in either left (table1) or right (table2) table records.

```
mysql> select a1.name,a2.city from a1 left join a2 on a1.id=a2.id
    -> union
    -> select a1.name,a2.city from a1 right join a2 on a1.id=a2.id;
+-------+---------+
| name  | city    |
+-------+---------+
| akash | noida   |
| a     | delhi   |
| b     | lucknow |
| c     | NULL    |
+-------+---------+
```

Natural Join

A natural join is based on all columns in the two tables that have the same name and selects rows with equal values in those columns.

```
mysql> select * from a1 natural join a2;
+------+--------------+---------+
| id   | name         | city    |
+------+--------------+---------+
|    1 | akash_rattan | noida   |
|    2 | a            | delhi   |
|    3 | b            | lucknow |
+------+--------------+---------+
3 rows in set (0.00 sec)
```

Equi Join

An equi join is a specific type of join that combines tables based on matching values in specified columns.

```
mysql> select a1.name,a2.city from a1 ,a2 where a1.id=a2.id;
+--------------+---------+
| name         | city    |
+--------------+---------+
| akash_rattan | noida   |
| a            | delhi   |
| b            | lucknow |
+--------------+---------+
```

9. Introduction of VIEW. How to create view. Explain with example.

In the context of databases, a view is a virtual table that consists of a subset of data from one or more tables. Views are useful for various purposes, such as simplifying complex queries, providing a layer of security by restricting access to certain columns or rows, and presenting data in a particular format for reporting or analysis.

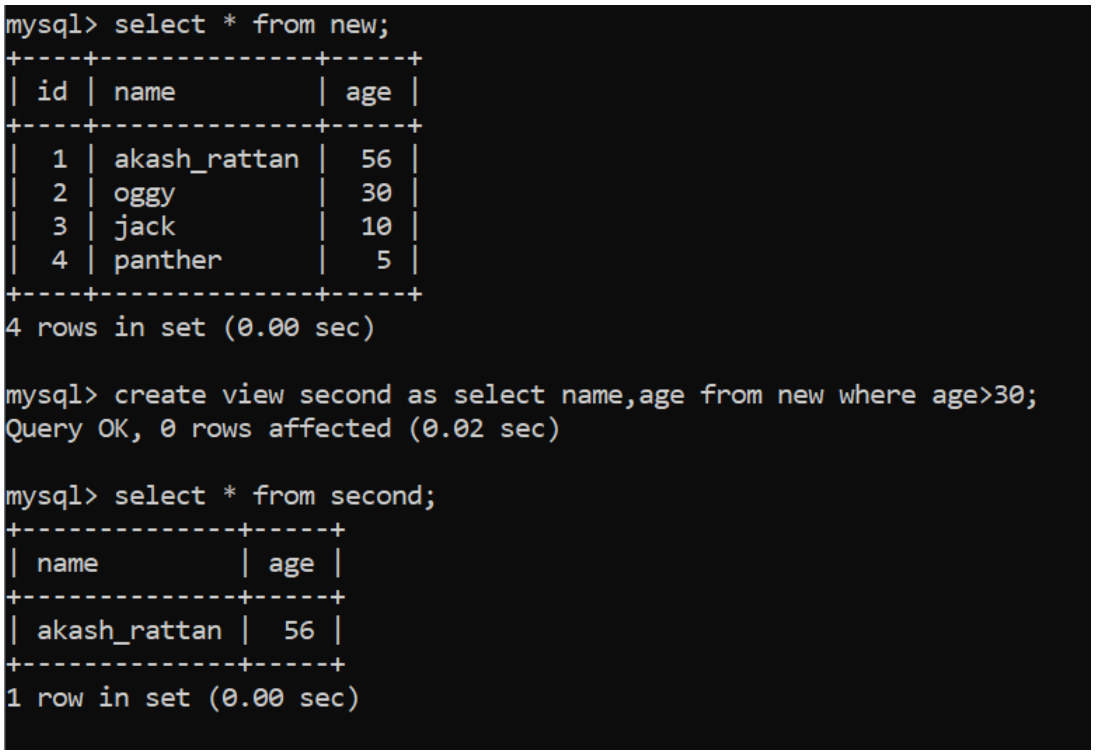
Creating a view in a typical MySQL database -

```CREATE VIEW view_name AS

SELECT column1, column2,

FROM table_name

WHERE condition;```


```