

|  |
| --- |
| **Experiment No.5** |
| Perform simple queries, string manipulation operations and aggregate functions. |
| Date of Performance: |
| Date of Submission: |



**Aim :-** Write simple query to manipulate string operations and perform aggregate functions like (MIN, MAX, SUM, AVERAGE, COUNT).

**Objective :-** To apply aggregate functions and string manipulation functions to perform simple queries in the database system

**Theory:**

**Simple Queries in SQL:**

In SQL, a simple query is a request for data from a database table or tables. It allows users to retrieve specific information by specifying the columns they want to retrieve and any conditions for filtering rows based on certain criteria. Simple queries are the backbone of interacting with databases, enabling users to extract the data they need for analysis, reporting, or further processing.

String Manipulation Operations:

String manipulation operations in SQL involve modifying or transforming string values stored in database columns. These operations are crucial for tasks such as formatting data, combining strings, converting case, or extracting substrings. By using string functions and operators, users can manipulate text data to suit their requirements, whether it's for display purposes or for further analysis.

**Aggregate Functions:**

Aggregate functions in SQL are used to perform calculations on sets of values and return a single result. These functions allow users to summarize data across multiple rows, providing insights into the overall characteristics of the dataset. Common aggregate functions include calculating counts, sums, averages, minimums, and maximums of numerical values. They are essential tools for data analysis, enabling users to derive meaningful insights from large datasets.

Benefits of Understanding These Concepts:

● Data Retrieval: Simple queries allow users to fetch specific data from databases,

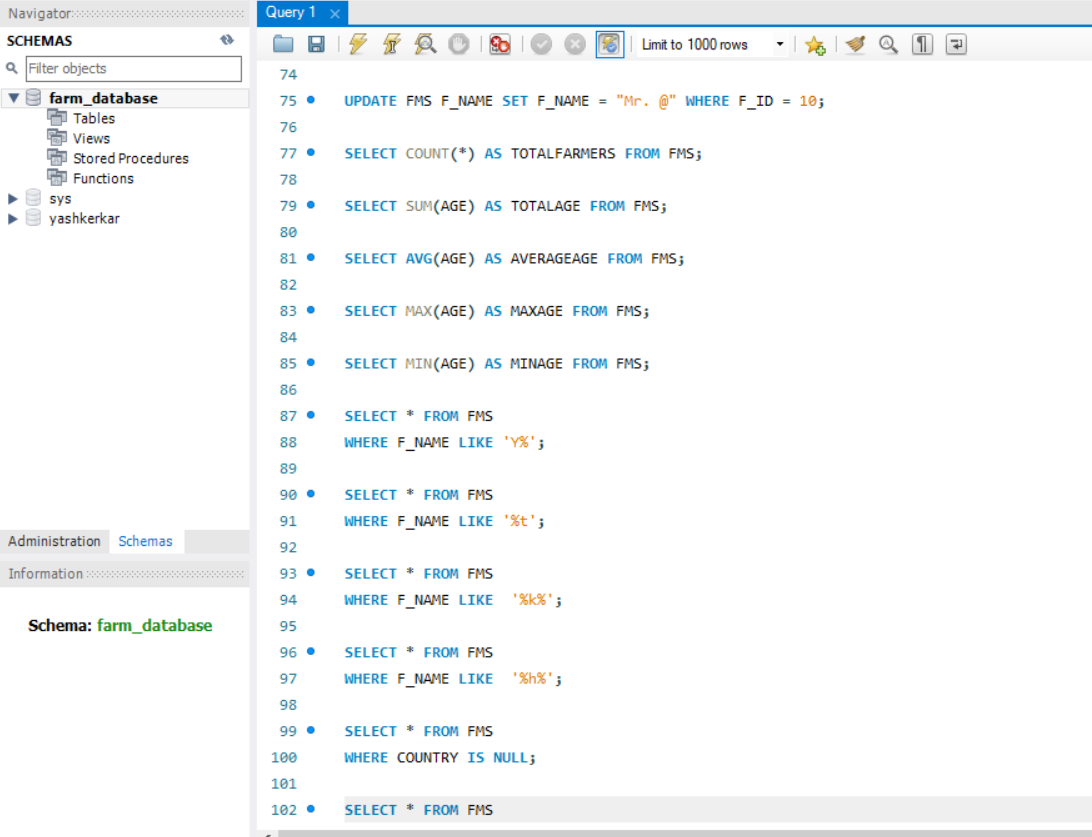
facilitating data retrieval for various purposes.

● Data Transformation: String manipulation operations enable users to format and transform text data according to their needs, improving data consistency and readability.

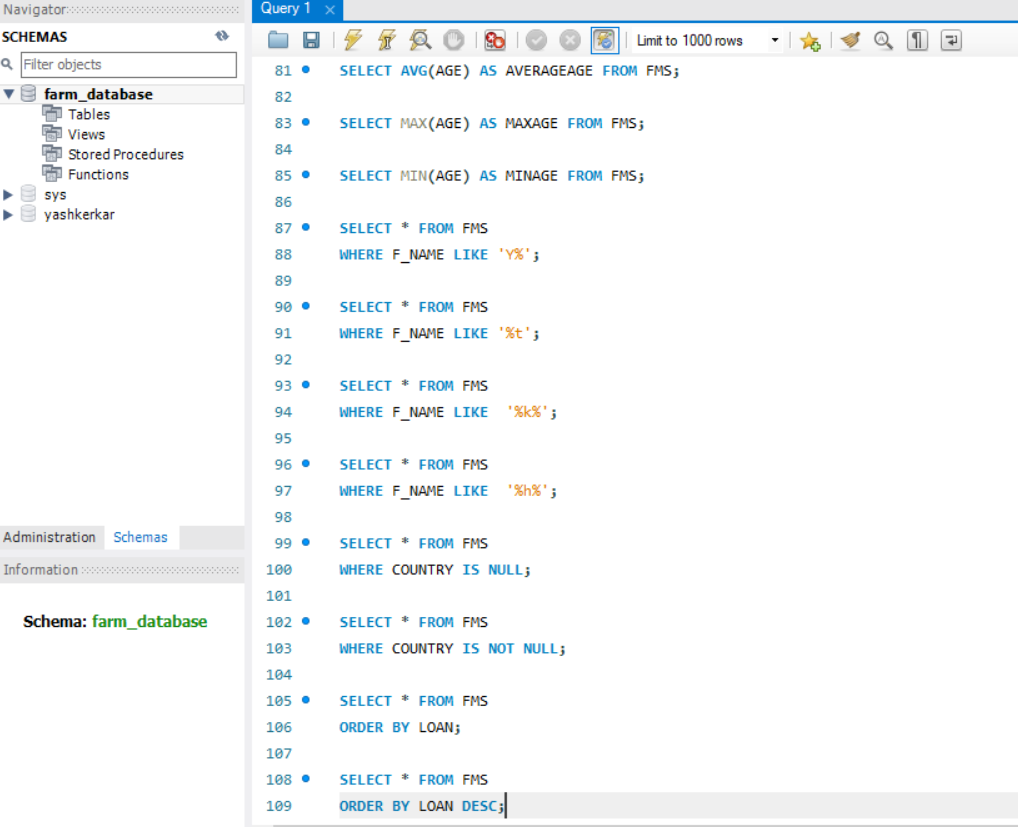
● Data Analysis: Aggregate functions help users summarize and analyze large datasets, providing valuable insights into trends, patterns, and statistical measures.



**Implementation:**

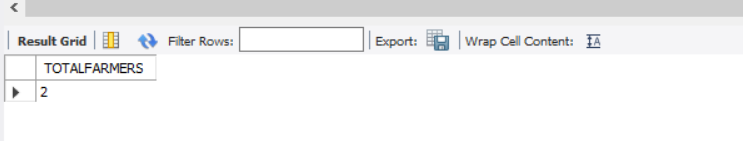
****

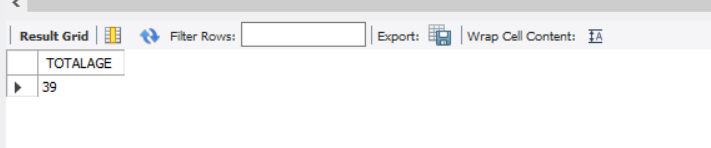


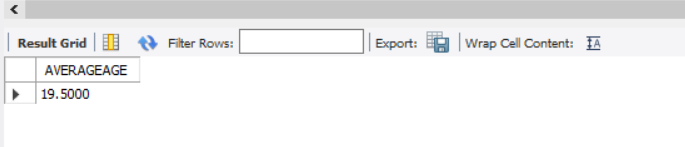
****

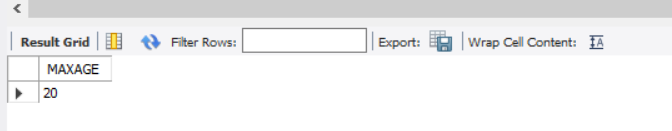


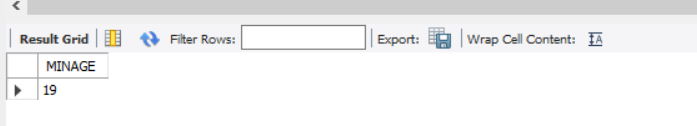
**Output:**

****

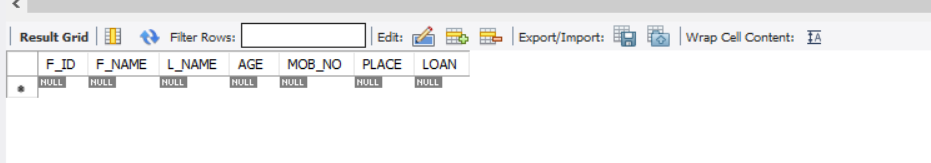
****

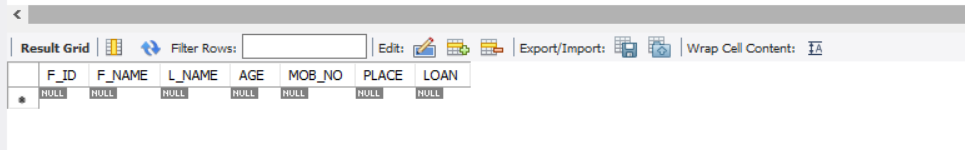
****

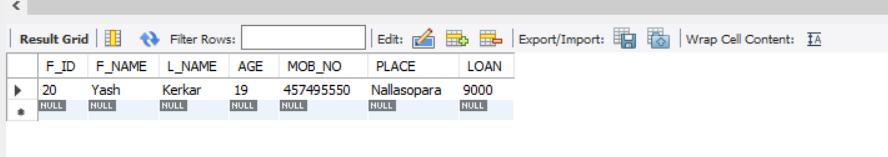
****

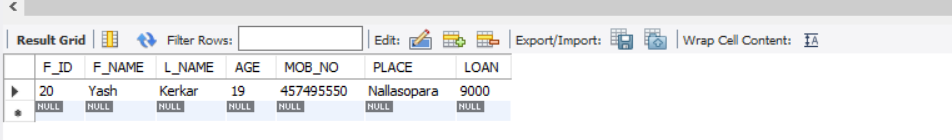
****

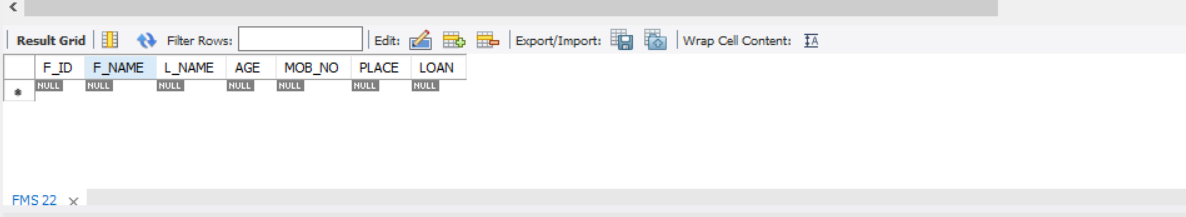


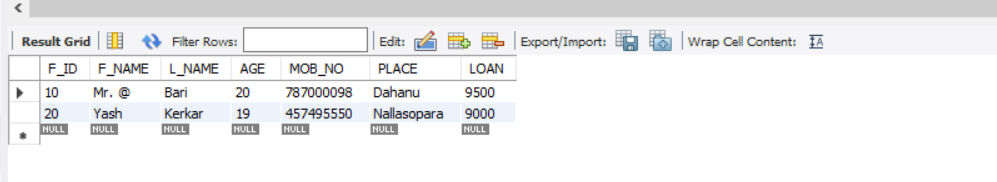
****

****

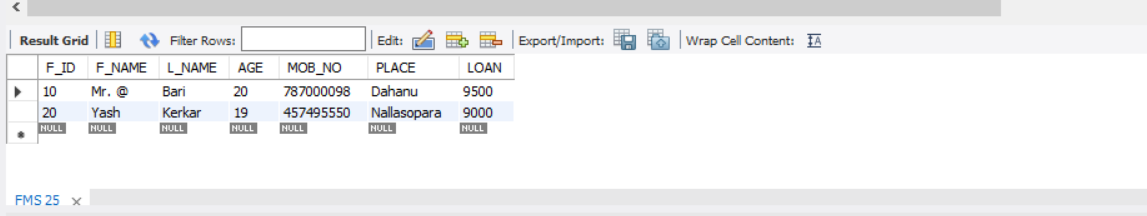
****

****

****

****



****

**Conclusion:**

Through this practical exercise, we have successfully demonstrated the proficiency in performing various queries, string manipulation operations, and aggregate functions in a database management system. These operations are fundamental in managing and extracting meaningful insights from data stored within databases.

1. Write syntax and explanation for each of the five aggregate functions
2. Ans. COUNT():
   * Syntax: **COUNT(expression)**
   * Explanation: This function counts the number of rows in a specified column or all rows in a table if no column is specified. It ignores NULL values unless the expression is **COUNT(\*)**, which counts all rows regardless of NULL values.
3. **SUM()**:
   * Syntax: **SUM(expression)**
   * Explanation: This function calculates the sum of all non-NULL values in a specified column. It is commonly used with numeric data types such as integers, decimals, or floating-point numbers.
4. **AVG()**:
   * Syntax: **AVG(expression)**
   * Explanation: This function calculates the average (mean) of all non-NULL values in a specified column. It is useful for finding the typical value within a dataset an
5. **MAX()**:
   * Syntax: **MAX(expression)**
   * Explanation: This function returns the largest (maximum) value in a specified column. Similar to MIN(), it is applicable to numeric, string, or date/time data types and is used to find the highest value within a dataset.

2. Show results of operations performed.

Ans. **COUNT()**:

SELECT COUNT(\*) AS total\_farmers FROM fms;

total\_farmers

--------------

50

SUM():

SELECT SUM(loan) AS total\_loans FROM FMS;

total\_loans

-----------

12500

**AVG()**:

SELECT AVG(loan) AS average\_loan FROM FMS;

average\_salary

---------------

55000

MIN():

SELECT MIN(age) AS lowest\_age FROM FMS;

lowest\_age

------------

19

**MAX()**:

SELECT MAX(age) AS highest\_age\_date FROM FMS;

lowest\_age

------------