



Vidyavardhini's College of Engineering and Technology, Vasai

Department of Artificial Intelligence & Data Science

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



Experiment No.5

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.

- Data Reporting: By combining simple queries, string manipulation operations, and aggregate functions, users can generate reports and visualizations that communicate key findings effectively.

Implementation:

```
1)INSERT INTO Patients (patient_id, name, age, gender, address, phone_number)
VALUES (1, 'John Doe', 30, 'Male', '123 Main St, City', '+1234567890'),
      (2, 'Jane Smith', 25, 'Female', '456 Oak St, Town', '+1987654321');
```

```
2)UPDATE Patients
SET age = 35
WHERE patient_id = 1;
```

```
3) DELETE FROM Patients
WHERE patient_id = 2;
```



Conclusion:

1. Write syntax and explanation for each of the five aggregate functions.

Aggregate functions are SQL functions used to perform calculations on sets of values and return a single result. Here's a brief explanation of each aggregate function along with the syntax and an example of its usage:

- **MIN:** Finds the smallest value in a set. Syntax: `MIN(column_name)` Example: `SELECT MIN(age) AS min_age FROM Player;`
- **MAX:** Finds the largest value in a set. Syntax: `MAX(column_name)` Example: `SELECT MAX(runs_scored) AS max_runs_scored FROM Player;`
- **SUM:** Calculates the sum of values in a set. Syntax: `SUM(column_name)` Example: `SELECT SUM(runs_scored) AS total_runs_scored FROM Player;`
- **AVG:** Calculates the average value in a set. Syntax: `AVG(column_name)` Example: `SELECT AVG(player_salary) AS average_salary FROM Player;`
- **COUNT:** Counts the number of rows in a set. Syntax: `COUNT(*)` or `COUNT(column_name)` Example: `SELECT COUNT(*) AS total_matches FROM Match;`

2. Show results of operations performed.

The screenshot shows a database management interface with a sidebar on the left and a main query editor on the right. The sidebar includes a menu with 'Options File', 'PERFORMANCE', 'Dashboard', 'Performance Reports', and 'Performance Schema Setup'. Below this is a 'ministration' section with 'Schemas' and 'ormation' tabs. A message at the bottom of the sidebar states 'No object selected'. The main query editor displays a series of SQL queries, each preceded by a line number (21-34). The queries are: 21: (empty line), 22: `INSERT INTO Patients (patient_id, name, age, gender, address, phone_number)`, 23: `VALUES (1, 'John Doe', 30, 'Male', '123 Main St, City', '+1234567890'),`, 24: `(2, 'Jane Smith', 25, 'Female', '456 Oak St, Town', '+1987654321');`, 25: (empty line), 26: `UPDATE Patients`, 27: `SET age = 35`, 28: `WHERE patient_id = 1;`, 29: (empty line), 30: `DELETE FROM Patients`, 31: `WHERE patient_id = 2;`, 32: (empty line), 33: (empty line), 34: (empty line).