



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

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

Implementation:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

1) Concatenate two strings:

The screenshot shows a SQL IDE interface with a Navigator on the left and a SQL editor on the right. The Navigator displays the 'studentdb' schema with tables like 'classes', 'department', 'exams', 'faculty', 'fees', and 'student'. The SQL editor contains the following query:

```
1 • SELECT * FROM studentdb.student;
2 • use studentdb;
3
4 • SELECT concat(student_name, 'lives in', address) as full_info
5 • from student;
```

The Result Grid shows the output of the query:

| full_info |
|---|
| Viraj Ozalives inMG Road Mira Road |
| Sarvesh Survelives inSV Road Andheri West |
| Rahul Yadavives inLT Road Vasai West |

2) Get the length of student names:

The screenshot shows the same SQL IDE interface. The SQL editor contains the following query:

```
1 • use studentdb;
2
3 • select student_name, LENGTH(student_name) AS name_length
4 • from student;
```

The Result Grid shows the output of the query:

| student_name | name_length |
|---------------|-------------|
| Viraj Oza | 9 |
| Sarvesh Surve | 13 |
| Rahul Yadav | 11 |



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

3) Find the name with the shortest length:

The screenshot shows a SQL IDE interface with a Navigator on the left and a SQL editor on the right. The Navigator displays the 'studentdb' schema with tables: classes, department, exams, faculty, fees, and student. The SQL editor contains the following query:

```
1 • use studentdb;
2
3 • select student_name
4   from student
5  group by student_name
6  order by(student_name) desc
7  limit 1;
```

The Result Grid at the bottom shows the following data:

| student_name |
|--------------|
| Viraj Oza |

4) Calculate the total salary:

The screenshot shows a SQL IDE interface with a Navigator on the left and a SQL editor on the right. The Navigator displays the 'studentdb' schema with tables: classes, department, exams, faculty, fees, and student. The SQL editor contains the following query:

```
1 • use studentdb;
2
3 • SELECT SUM(Salary) AS Total_Salary
4   FROM Faculty;
```

The Result Grid at the bottom shows the following data:

| Total_Salary |
|--------------|
| 144000 |

5) Find the highest and lowest salary:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 12* SQL File 14* x

Limit to 1000 rows

```
1 • use studentdb;
2
3 • SELECT MAX(Salary) As Max_Salary, MIN(Salary) AS Min_Salary
4 FROM Faculty;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

| | Max_Salary | Min_Salary |
|---|------------|------------|
| ▶ | 52000 | 42000 |

Result Grid

6) Calculate the average salary of employees in each department:

Navigator: SCHEMAS

Filter objects

- studentdb
 - Tables
 - classes
 - department
 - exams
 - faculty
 - Columns
 - Faculty_name
 - Faculty_id
 - Phone_no
 - Salary
 - email
 - Indexes
 - Foreign Keys
 - Triggers
 - fees
 - student
 - Views
 - Stored Procedures
 - Functions
 - studentdb_schema
 - sys

SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 12* SQL File 15* x

Limit to 1000 rows

```
1 • use studentdb;
2
3 • SELECT Faculty_id, AVG(Salary) AS Avg_salary
4 FROM Faculty
5 GROUP BY Faculty_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

| | Faculty_id | Avg_salary |
|---|------------|------------|
| ▶ | 22 | 52000 |
| | 1234 | 50000 |
| | 5678 | 42000 |

Result Grid

Form Editor

Field Types

Conclusion:

The syntax and brief explanations for five common aggregate functions in SQL:



1. SUM():

Syntax: 'SUM(column_name)'

Explanation: Calculates the sum of all values in the specified column. It is commonly used to find the total of numeric values in a column.

2. AVG():

Syntax: 'AVG(column_name)'

Explanation: Calculates the average (mean) of all values in the specified column. It is useful for finding the average value of numeric data.

3. MAX():

Syntax: 'MAX(column_name)'

Explanation: Returns the maximum value from the specified column. It is used to find the highest value in a set of data.

4. MIN():

Syntax: 'MIN(column_name)'

Explanation: Returns the minimum value from the specified column. It is used to find the lowest value in a set of data.

5. COUNT():

Syntax: 'COUNT(column_name)' or 'COUNT(*)'

Explanation: Returns the number of rows that match the specified condition. It can count the number of non-null values in a column when a column name is provided, or it can count all rows when 'COUNT(*)' is used.

These aggregate functions are commonly used in SQL queries to perform calculations and summarizations on data within a table.