



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:

Database:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

	MOVIE_ID	MOVIE_NAME	MOVIE_DATE	MOVIE_TIME	MOVIE_DETAILS
▶	1	ROWDY_RATHOD	12/3/2025	2:00PM	action/comedy
	2	HANUMan	14/3/2025	3:00PM	action
	3	TARE_ZAMEEN_PAR	16/3/2025	2:00PM	comedy
	4	Baby	20/3/2025	3:00PM	action/thriller
	5	Prem_Ratan_Dhan_Payo	25/3/2025	5:00PM	drama
	6	khiladi	2/4/2025	12:00PM	action/thriller
	7	Baahubali:_The_Beginning	12/4/2025	2:00PM	action/thriller

1)Concatenate two strings:

```
SELECT CONCAT (MOVIE_NAME, 'IS', MOVIE_DETAILS) AS MOVIE_IFO FROM MOVIE;
```

	MOVIE_IFO
▶	ROWDY_RATHODISaction/comedy
	HANUManISaction
	TARE_ZAMEEN_PARIScomedy
	BabyISaction/thriller
	Prem_Ratan_Dhan_PayoISdrama
	khiladiISaction/thriller
	Baahubali:_The_BeginningISaction/thriller

2)Get the length of employee names:

```
SELECT MOVIE_NAME, LENGTH(MOVIE_NAME) AS NAME_LENGTH FROM MOVIE;
```

	MOVIE_NAME	NAME_LENGTH
▶	ROWDY_RATHOD	12
	HANUMan	7
	TARE_ZAMEEN_PAR	15
	Baby	4
	Prem_Ratan_Dhan_Payo	20
	khiladi	7
	Baahubali:_The_Beginning	24

3)Find the department with the longest name:

```
SELECT MOVIE_NAME
FROM MOVIE
GROUP BY MOVIE_NAME
ORDER BY LENGTH(MOVIE_NAME) DESC
LIMIT 1;
```

	MOVIE_NAME
▶	Baahubali:_The_Beginning

4)Calculate the total salary expenditure for all employees:

```
SELECT SUM(MOVIE_PRICE) AS TOTAL_EARN FROM MOVIE;
```



Result Grid	Filter Rows:
TOTAL_EARN	
1220	

5) Find the highest and lowest salary in the company:

- ```
SELECT MAX(MOVIE_PRICE) AS HIGHEST_PRICE, MIN(MOVIE_PRICE) AS LOWEST_PRICE
FROM MOVIE;
```

|   | HIGHEST_PRICE | LOWEST_PRICE |
|---|---------------|--------------|
| ▶ | 300           | 50           |

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

- ```
SELECT MOVIE_NAME, AVG(MOVIE_PRICE) AS AVG_PRICE  
FROM MOVIE  
GROUP BY MOVIE_NAME
```

	MOVIE_NAME	AVG_PRICE
▶	ROWDY_RATHOD	50.0000
	HANUMAN	100.0000
	TARE_ZAMEEN_PAR	200.0000
	Baby	200.0000
	Prem_Ratan_Dhan_Payo	250.0000
	khiladi	300.0000

7) Count the number of employees in each department:

- ```
SELECT MOVIE_PRICE, COUNT(*) AS NUM_MOVIE
FROM MOVIE
GROUP BY MOVIE_PRICE;
```

|   | MOVIE_PRICE | NUM_MOVIE |
|---|-------------|-----------|
| ▶ | 50          | 1         |
|   | 100         | 1         |
|   | 200         | 2         |
|   | 250         | 1         |
|   | 300         | 1         |
|   | 120         | 1         |

### 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.