



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:

product_id	product_name	price	stock_quantity	category
2	Product B	9.88	50	Category 2
3	Product A	10.55	120	Category 3
4	Product A	10.99	100	Category 1
5	Product B	9.88	50	Category 2
6	Product C	15.00	500	Category 3
7	Product D	15.55	200	Category 4
8	Product E	20.10	150	Category 5
9	Ball	30.00	100	Category 1
10	Bat	100.00	50	Category 2
11	Stumps	800.00	300	Category 3
12	Helmet	850.00	200	Category 4
13	Kneepad	500.00	700	Category 5



1)Concatenate two strings:

```
50
51 • SELECT CONCAT(product_name, ' in ', category) AS product_details
52 FROM products;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
product_details				
▶	Product B in Category 2			
	Product A in Category 3			
	Product A in Category 1			
	Product B in Category 2			
	Product C in Category 3			
	Product D in Category 4			
	Product E in Category 5			

2)Get the length of product names:

```
53
54 • SELECT product_name, LENGTH(product_name) AS name_length
55 FROM products;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
product_name name_length				
▶	Product B 9			
	Product A 9			
	Product A 9			
	Product B 9			
	Product C 9			
	Product D 9			
	Product E 9			

3)Find the product with the longest name:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

```
57 • SELECT *
58 FROM products
59 ORDER BY LENGTH(product_name) DESC
60 LIMIT 1;
61
62
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:	Fetch rows:
product_id	product_name	price	stock_quantity	category	
2	Product B	9.88	50	Category 2	
NULL	NULL	NULL	NULL	NULL	

4) Calculate the total price of all products:

```
62 • SELECT SUM(price * stock_quantity) AS total_price
63 FROM products;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
total_price			
1134978.00			

5) Find the highest and lowest price:

```
68 • SELECT MIN(price) AS highest_price
69 FROM products;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
highest_price			
9.88			



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

```
65 • SELECT MAX(price) AS highest_price
66 FROM products;
67
68 • SELECT MIN(price) AS highest_price
69 FROM products;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
highest_price			
850.00			

6) Count the number of stocks in each category:

```
70
71 • SELECT category, COUNT(*) AS product_count
72 FROM products
73 GROUP BY category;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
category	product_count		
Category 2	3		
Category 3	3		
Category 1	2		
Category 4	2		
Category 5	3		

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(*)'



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

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.