**EXPERIMENT NO: 5**

**Date of Performance:   
Date of Submission:**

**AIM**:   
Perform string manipulation operations and aggregate functions.

**THEORY**:

SQL string functions are primarily utilized for string manipulation. The built-in SQL String functions make it easier for us to find and alter string values

### Examples of String Functions in SQL

#### ASCII () - It gives you the ASCII value of a character.

Code:

SELECT ASCII ('t');

**Output:**

**116**

#### CHAR\_LENGTH () - It gives you the number of characters in the string.

Code:

SELECT CHAR\_LENGTH ('world!');

**Output:**

**6**

#### CONCAT () - It appends two strings to create the new single string, as shown in the example.

Code:

SELECT CONCAT ('educba', '.com')

**Output:**

**educba.com**

#### CONCAT\_WS () -It appends two strings with a given symbol in between to concatenate them

Code:

SELECT CONCAT\_WS ('\_', 'educba', 'to', 'learn');

**Output:**

**educba\_to\_learn**

#### FIND\_IN\_SET () - It finds out the index position of any symbol or character from the given set

Code:

SELECT FIND\_IN\_SET ('v', 'z, x, v, b, n, m');

**Output:**

**3**

#### FORMAT () - It changes the format of the text from a string to any other format.

Code:

FORMAT ("0.254", "Percent");

**Output:**

**‘25.40%’**

#### INSERT () - It helps you insert text, integer, float, or double into your database.

Code:

INSERT INTO mydb (name, age) VALUES (‘sdf’, 20);

Output:

Inserted successfully

#### INSTR () - It gives you the first occurrence of the index of a character in the string.

Code:

INSTR('educba to learn', 'e');

**Output:**

**1 // (the first occurrence of ‘e’)**

#### LCASE () - It will replace every character in the string in their lowercase letter.

Code:

LCASE("eduCBA.com To Learn");

**Output:**

**educba.com to learn**

#### LEFT () - It is used to get the substring from the left of the string to the given index position.

Code:

SELECT LEFT('educba.com', 3);

**Output:**

**edu**

#### LENGTH () - Gives you the length of the string.

Code:

LENGTH('educba.com');

**Output:**

**10**

#### LOCATE () - It gives you the position of a substring in the given string.

Code:

SELECT LOCATE('cba', 'educba.com', 1);

**Output:**

**4**

#### LOWER () - It converts every character in a string to lowercase from uppercase.

Code:

SELECT LOWER('EDUCBA.COM');

**Output:**

**educba.com**

#### LPAD () - It adds left padding with the given symbol to make the string of a given size.

Code:

SELECT LPAD('yahoo', 7, '@');

**Output:**

**@@yahoo**

#### LTRIM () - It trims the given characters/ blank spaces from the left of the string.

Code:

LTRIM(' yahoo');

**Output:**

**yahoo**

#### MID () - It gives the substring from a given position to the number of characters in the string.

Code:

SELECT MID("educba.com", 3, 2);

**Output:**

**uc**

#### POSITION () - It gives you the index position of the character in the string.

Code:

SELECT POSITION('u' IN 'educba.com');

**Output:**

**3**

#### REPEAT () - It repeats the string to the number of times given.

Code:

SELECT REPEAT('educba ', 2);

**Output:**

**educba educba**

#### REPLACE () - It returns a new string by removing the given string from the original.

Code:

REPLACE ('456yahoo456', '456');

**Output:**

**yahoo**

#### REVERSE () - It reverses the characters in a string.

Code:

SELECT REVERSE('educba.com');

**Output:**

**moc.abcuda**

#### RIGHT () - It is used to get the substring from the right of the string to the given index.

Code:

SELECT RIGHT('educba.com', 4);

**Output:**

**.com**

#### RPAD () - It adds the right padding with the given symbol to make the string of the given size.

Code:

RPAD('educba', 9, '@');

**Output:**

**‘educba@@@’**

#### RTRIM () - It trims the given characters / blank spaces from the right of the string.

Code:

SELECT RTRIM('educba ');

**Output:**

**educba**

#### SPACE () - It adds the number of spaces specified.

Code:

SELECT SPACE(6);

**Output:**

**‘      ‘**

#### STRCMP () -

Matches two strings.

* + If both the strings are the same, it gives 0.
  + If the first is smaller than the second, then it gives -1.
  + If the first is bigger than the second, then it gives 1.

Code:

SELECT STRCMP('zomato.com', 'educba.com');

**Output:**

**1**

#### SUBSTR () - It returns a new substring from a given position to the number of characters.

Code:

SUBSTR('educba.com', 1, 4);

**Output:**

**educ**

#### SUBSTRING () - It gives you the substring character from the given string.

Code:

SELECT SUBSTRING('eduCba.com', 4, 1);

**Output:**

**‘C’**

#### SUBSTRING\_INDEX () - It gives you the substring until it finds the given symbol.

Code:

SELECT SUBSTRING\_INDEX( anillondhe' , 'l', 1);

**Output:**

**ani**

**UCASE ()** – It will replace every character in the string in their uppercase letter.

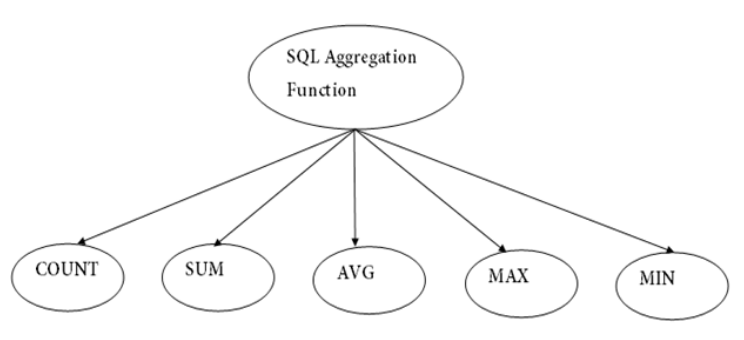
Code:

SELECT UCASE("EduCbA");

**Output:**

**EDUCBA**

# SQL Aggregate Functions



### COUNT FUNCTION

* COUNT function is used to Count the number of rows in a database table. It can work on both numeric and non-numeric data types.
* COUNT function uses the COUNT (\*) that returns the count of all the rows in a specified table. COUNT (\*) considers duplicate and Null.

**Sample table:**

**PRODUCT\_MAST**

| **PRODUCT** | **COMPANY** | **QTY** | **RATE** | **COST** |
| --- | --- | --- | --- | --- |
| Item1 | Com1 | 2 | 10 | 20 |
| Item2 | Com2 | 3 | 25 | 75 |
| Item3 | Com1 | 2 | 30 | 60 |
| Item4 | Com3 | 5 | 10 | 50 |
| Item5 | Com2 | 2 | 20 | 40 |
| Item6 | Cpm1 | 3 | 25 | 75 |
| Item7 | Com1 | 5 | 30 | 150 |
| Item8 | Com1 | 3 | 10 | 30 |
| Item9 | Com2 | 2 | 25 | 50 |
| Item10 | Com3 | 4 | 30 | 120 |

**Syntax**

COUNT (\*)

or

COUNT ([ALL|DISTINCT] expression)

Code:

SELECT COUNT (\*) FROM PRODUCT\_MAST;

**Output:**10

**COUNT with WHERE**

SELECT COUNT(\*)  FROM PRODUCT\_MAST WHERE RATE>=20;

**Output:**7

**COUNT () with DISTINCT**

SELECT COUNT (DISTINCT COMPANY) FROM PRODUCT\_MAST;

**Output:**3

**COUNT() with GROUP BY**

SELECT COMPANY, COUNT(\*) FROM PRODUCT\_MAST GROUP BY COMPANY;

**Output:**Com1 5  
Com2 3  
Com3 2

**COUNT () with HAVING**

SELECT COMPANY, COUNT (\*) FROM PRODUCT\_MAST GROUP BY COMPANY

HAVING COUNT (\*)>2;

**Output:**Com1 5  
Com2 3

### SUM Function

Sum function is used to calculate the sum of all selected columns. It works on numeric fields only.

**Syntax:**

SUM()

or

SUM( [ALL|DISTINCT] expression )

**Code:**

SELECT SUM(COST) FROM PRODUCT\_MAST;

**Output:**

670

**SUM() with WHERE**

**Code:**

SELECT SUM(COST) FROM PRODUCT\_MAST WHERE QTY>3;

**Output:**

320

**SUM() with GROUP BY**

**Code:**

SELECT SUM(COST) FROM PRODUCT\_MAST WHERE QTY>3  GROUP BY COMPANY;

**Output:**Com1 150  
Com2 170

**SUM() with HAVING**

**Code:**

SELECT COMPANY, SUM(COST) FROM PRODUCT\_MAST

GROUP BY COMPANY HAVING SUM(COST)>=170;

**Output:**Com1 335  
Com3 170

### AVG function

The AVG function is used to calculate the average value of the numeric type. AVG function returns the average of all non-Null values.

**Syntax**

AVG()

or

AVG( [ALL|DISTINCT] expression )

**Code:**SELECT AVG(COST)  FROM PRODUCT\_MAST;

**Output:**

67.00

### MAX Function

MAX function is used to find the maximum value of a certain column. This function determines the largest value of all selected values of a column.

MAX()

or

MAX( [ALL|DISTINCT] expression )

**Code:**SELECT MAX(RATE)  FROM PRODUCT\_MAST;

**Output:**

30

### MIN Function

MIN function is used to find the minimum value of a certain column. This function determines the smallest value of all selected values of a column.

**Syntax**

MIN()

or

MIN( [ALL|DISTINCT] expression )

**Code:**SELECT MIN(RATE)  FROM PRODUCT\_MAST;

**Output:**

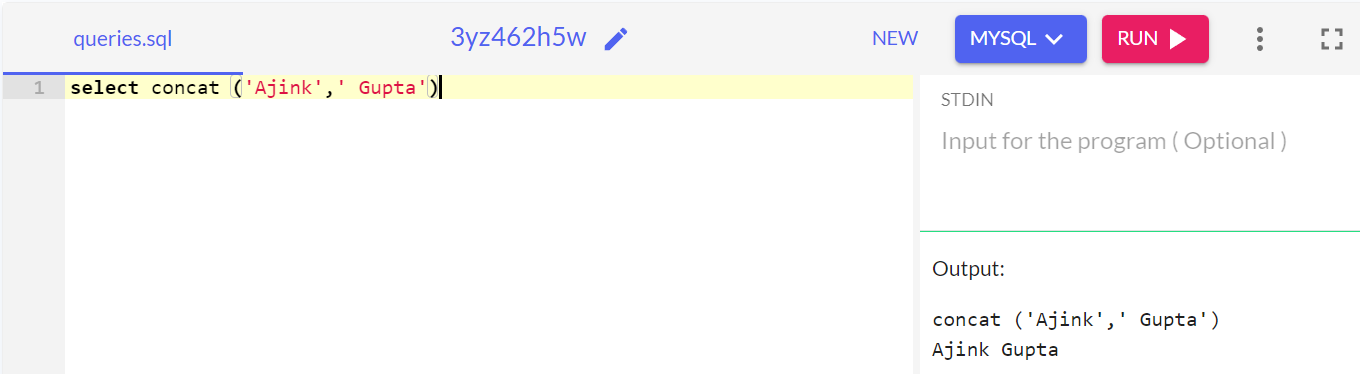
10

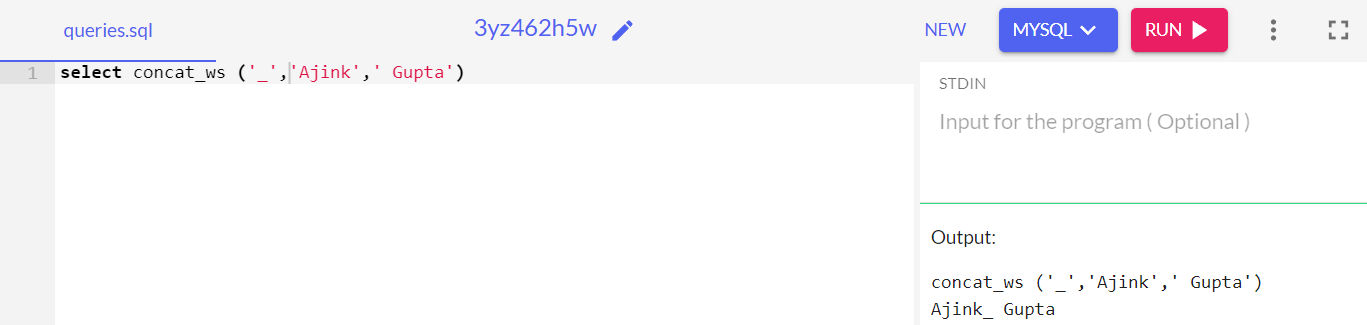
**CONCLUSION / OUTCOME**:

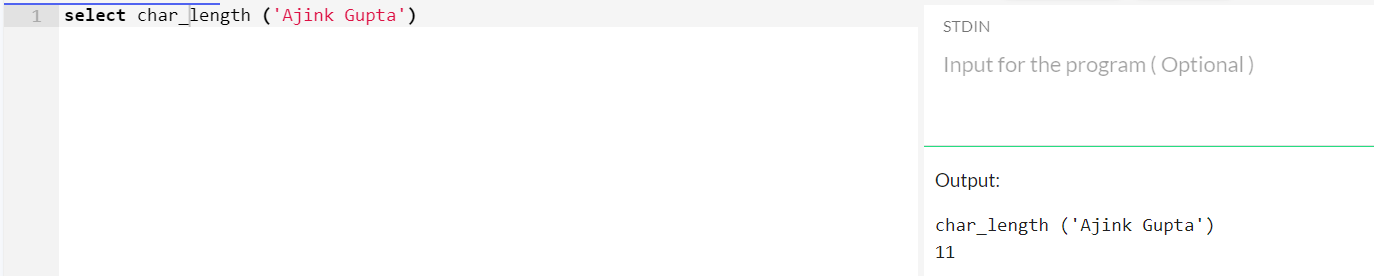
**MARKS & SIGNATURE:**

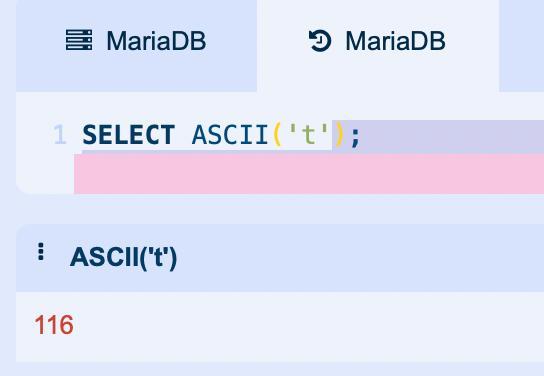
| **R1  (3 Marks)** | **R2  (5 Marks)** | **R3  (4 Marks)** | **R4  (3 Marks)** | **Total  (15 Marks)** | **Signature** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

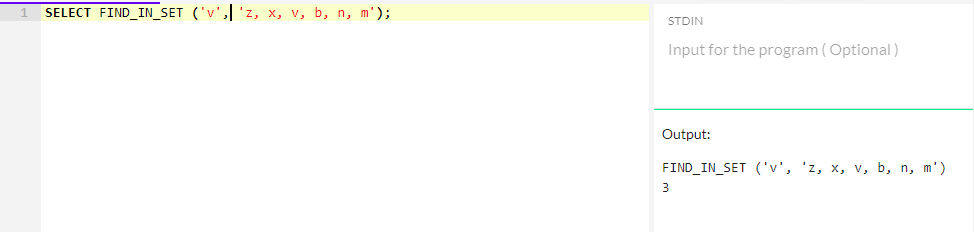
**Outputs :**

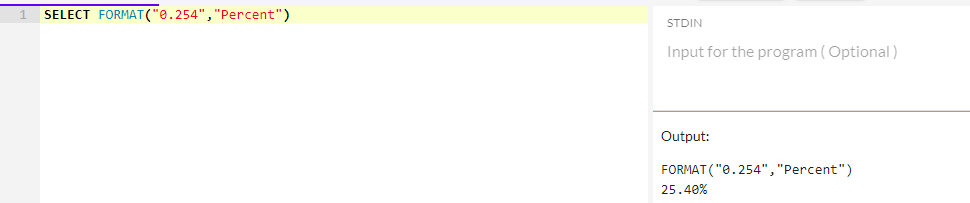
****

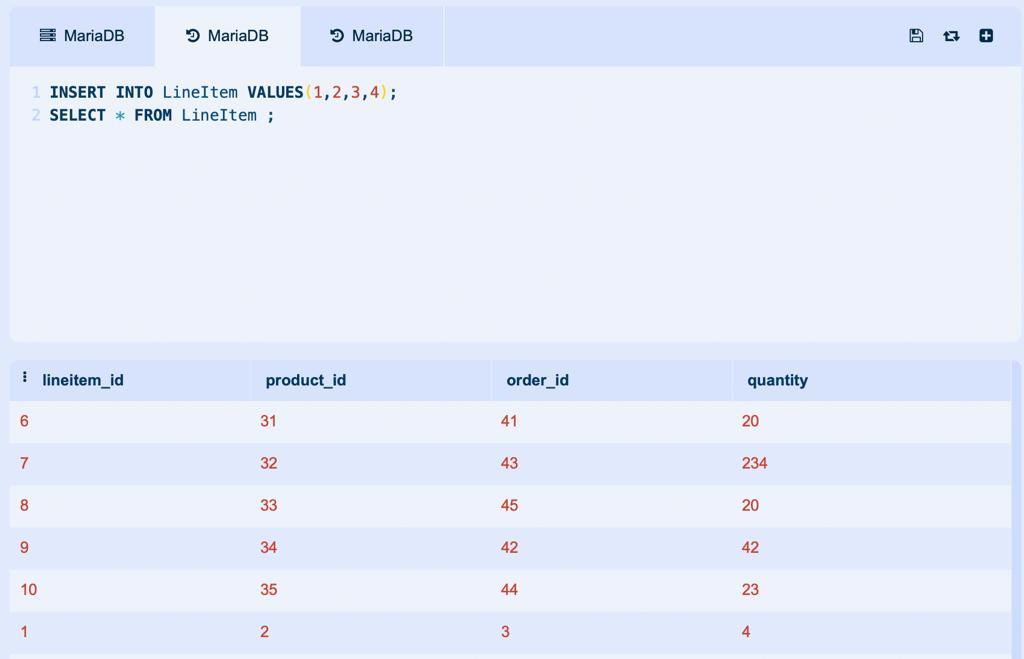
****

****



****

****



****







