

DATABASE QUERY USING SQL

FUNCTION in MySQL

❖ **Function:**

A function is a predefined command set that performs some operation and returns the single value. A function can have single, multiple or no arguments at all.

❖ **Types of SQL Functions:**

1) **Single Row Functions:**

- Single row function in SQL can be **character, numeric, date, and conversion functions**.
- These functions are used to modify data items. These functions need one or more input and operate on each row, thereby returning one output value for each row

2) **Multiple row Functions (Aggregate Functions):**

- The Multiple Row Functions in SQL are **used to return either group of values (or) a single value**.
- These functions are basically operated on a set of rows and return one result or one result per group.
- The Multiple row function in Oracle is also called group functions or it is also called aggregate functions.

Single Row Functions:

There are three types of Single Row Functions in SQL

- 1) Character / String Functions
- 2) Numeric Functions
- 3) Date and Time Functions

1) Character / String Functions:

- i. CONCAT()
- ii. LOWER() / LCASE()
- iii. UPPER()/UCASE()
- iv. LTRIM()
- v. TRIM()
- vi. RTRIM()
- vii. SUBSTR()/MID()
- viii. INSTR(),
- ix. LENGTH()
- x. RIGHT()
- xi. LEFT()

2) Numeric / Math Functions:

- i. POWER(),
- ii. ROUND(),
- iii. MOD()

3) Date Functions:

- i. SYSDATE()
- ii. NOW()
- iii. DATE()
- iv. MONTH()
- v. YEAR()
- vi. DAYNAME()
- vii. MONTHNAME()
- viii. DAY()

Math Functions:

1.Pow(x,y)/power(x,y): Returns the value of X raised to the power of Y.

Example:

- (i)Select POW(2,4); Result:16
- (ii)SELECT POW(2,-2; Result:0.25
- (iii)SELECT POW(-2,3); Result: -8
- (iv)SELECT id, salary, POWER(salary,2) FROM employee;

Result:

```
+----+-----+-----+
| id | salary | power(salary,2) |
+----+-----+-----+
| 1 | 25000.00 | 625000000 |
| 2 | 30000.00 | 900000000 |
| 3 | 32000.50 | 1024032000.25 |
| 4 | 37500.50 | 1406287500.25 |
| 5 | 42389.50 | 1796869710.25 |
+----+-----+-----+
```

2.ROUND(X): Rounds the argument to zero decimal place, whereas ROUND(X, d) rounds X to d decimal places.

Example:

- (i) ROUND(-1.23); Result: -1
- (ii) ROUND(-1.68); Result: -2
- (iii) ROUND(1.58); Result: 2
- (iv) ROUND(3.798, 1); Result: 3.8
- (v) ROUND(1.298, 0); Result: 1
- (vi) ROUND(76823.298, -1); Result: 76820
- (vii) ROUND(25.298,-1); Result: 30
- (viii) ROUND(3.798, 1); Result: 3.8
- (ix) ROUND(4536.78965,-3) Result: 5000
- (X) ROUND(4536.564553,-2): Result: 4500

(XI) ROUND(4586.564553,-2): Result: 4600
 (XII)ROUND(76823.298, -2); Result:76800
 XII)ROUND(76823.298, 2); Result: 76823.30
 (XIII) ROUND(3.798, 2); Result: 3.80

3. **MOD(x,y)**: Divides x by y and gives the remainder.

(i)SELECT MOD(12,5); Result: 2

CHARACTER / STRING FUNCTIONS

1. LENGTH(): Returns the length of a string in bytes/no. of characters in string.

Example:

(i) SELECT LENGTH(`#INFOR MATICS#`); Result:14

(ii)SELECT LENGTH(First_Name) FROM Employee;

Result:

```
+-----+
| LENGTH(First_Name) |
+-----+
| 4 |
| 7 |
| 8 |
| 5 |
| 6 |
+-----+
5 rows in set (0.00 sec)
```

2. INSTR(): Returns the index of the first occurrence of substring.

Example:

(i) SELECT INSTR('Informatics',' mat');

Result: 6 (since 'm' of 'mat' is at 6th place)

(ii) SELECT INSTR ('Computers', 'pet');

Result: 0

(iii) mysql> SELECT INSTR (First_Name,'Kiran') FROM Employee;

Result:

```
+-----+
| INSTR(First_Name,'Kiran') |
+-----+
| 0 |
| 0 |
| 4 | Select instr("good morning to all","or")
| 0 |
| 0 |
+-----+
```

(iv) Select instr("good morning to all","or") Result: 7

5. LOWER()/ LCASE(): Convert the string in lowercase.

Example:

SELECT LOWER('INFORMATICS'); Result: informatics

6. UPPER()/ UCASE(): Convert the string in uppercase.

Example:

SELECT UCASE('informatics'); Result: INFORMATICS

7.LEFT(): Returns the given number of characters by extracting them from the left side of the given string

Example:

SELECT LEFT('INFORMATICS PRACTICES', 3); Result: INF

8. RIGHT(): Returns the given number of characters by extracting them from the right side of the given string

Example:

SELECT RIGHT('INFORMATICS PRACTICES',3); Result: CES

9. MID()/ SUBSTR(): Returns a substring starting from the specified position in a given string.

Example:

- (i) SUBSTR('INFORMATICS PRACTICES',3,4); Result: FORM
- (ii) SELECT SUBSTRING('Informatics',3); Result: 'formatics'
- (iii) SELECT SUBSTRING('Computers', -3); Result: 'ers'
- (iv) SELECT SUBSTRING('Computers', -5, 3); Result: 'ute'
- (v) SELECT MID('Informatics',3,4); Result: 'form'
- (vi) SELECT MID(first_name,3,2) FROM Employee;

Result:

```
+-----+
| MID(first_name,3,2) |
+-----+
| it |
| ek |
| vk |
| mt |
| aw |
+-----+
```

10. LTRIM(): Removes leading spaces.

Example:

SELECT LTRIM(' INFORMATICS '); Result: 'INFORMATICS'

11. RTRIM(): Removes trailing spaces.

Example:

SELECT RTRIM(' INFOR MATICS '); Result: ` INFOR MATICS'

12. TRIM(): Removes leading and trailing spaces.

Example:

SELECT TRIM(' \$\$INFOR MATICS\$\$ '); Result: '\$\$INFOR MATICS\$\$'

Date/Time Functions

1. NOW(): Returns the current date and time

Example:

select NOW(); Result: '2020-04-06 13:58:11'

3. DATE(): Extracts the date part of a date or datetime expression

Example:

SELECT DATE('2020-04-06 01:02:03'); Result: '2020-04-06'

3. MONTH(): Returns the month from the date passed as argument.

Example:

SELECT MONTH('2020-03-21'); Result:3

4. YEAR(): Returns the year.

Example:

SELECT YEAR('2020-03-21'); Result: 2020

5. DAYNAME(): Returns the name of the weekday.

Example:

SELECT DAYNAME('2010-07-21'); Result: WEDNESDAY

6. DAY(): Returns the day of the month

Example:

SELECT DAY('2022-07-14'); Result: 14

7. MONTHNAME(): Returns the name of the month

Example:

SELECT MONTHNAME('2022-07-14'); Result: July

Multiple Row Functions (Aggregate Function)

- Aggregate functions summarize the results of a query and return a single value calculated from values in a column instead of providing the listing of all of the rows.

Syntax:

SELECT <FUNCION> (column_name) FROM <table_name>;

The following are aggregate functions:

1) SUM(): returns the total sum of a numeric column. It gives the arithmetic sum of all the values present in a particular column. It can take only one argument. NULL values are not included in the calculations.

Example: SELECT SUM(MARKS) FROM STUDENT;

It displays sum of all the marks in the table student

2) AVG(): returns the average value of any column or expression based on a column. NULL value not included

Example: SELECT AVG(MARKS) FROM STUDENT;

It displays average of all the marks in the table student

3) MAX(): It returns the maximum value among the given set of values of any column or expression based on column.

Example: SELECT MAX(MARKS) FROM STUDENT;

It displays maximum marks from the column marks of student table.

4) MIN(): It returns the minimum value among the given set of values of any column or expression based on column.

Example: SELECT MIN (MARKS) FROM STUDENT;

It displays minimum marks from the column marks of student table.

5) COUNT(): It count the number of non-null values in a column. It can take one argument, which can be a column name or *. When the argument is a column name then COUNT() returns the non-null values in that column. If the argument is an * then COUNT() counts the total number of records / rows along with the NULL values satisfying the condition, if any, in the table. So, it returns the total number of records or rows from the table.

Syntax: SELECT COUNT(COLUMN_NAME) FROM <TABLE_NAME>;

Example: SELECT COUNT(*) FROM STUDENT ;

It will give output as 10 rows.

But while writing SELECT COUNT(MARKS) FROM STUDENT;

Will give output as 7 because there will be 3 null values which is ignored by COUNT()

SORTING IN SQL – ORDER BY

- The SQL **ORDER BY** clause is used to sort data in ascending or descending order based on one or more columns.
- It sorts record in ascending order by default.
- To sort data in descending order DESC keyword is used.

Syntax:

```
SELECT <column_name> FROM <table_name>
```

```
[where <condition>]
```

```
ORDER BY <column_name> [ASC/DESC];
```

Example: To display the roll number, name and marks of students on the basis of their marks in ascending order.

```
SELECT ROLLNO, NAME, MARKS FROM STUDENT
```

```
ORDER BY NAME;
```

Sorting data on Multiple columns:

Syntax:

```
SELECT <column_name> FROM <table_name>  
[where <condition>]  
ORDER BY <column_name> [ASC/DESC] , <column_name> [ASC/DESC];
```

Example: To display the roll number, name and marks of all the students in descending order of their marks and ascending order of their names.

```
SELECT ROLLNO, NAME , MARKS FROM STUDENT  
ORDER BY MARKS DESC, NAME;
```

GROUP BY in SQL

- At times we need to fetch a group of rows on the basis of common values in a column. This can be done using a GROUP BY clause.
- It groups the rows together that contain the same values in a specified column. We can use the aggregate functions (COUNT, MAX, MIN, AVG and SUM) to work on the grouped values.
- HAVING Clause in SQL is used to specify conditions on the rows with GROUP BY clause.

Syntax:

```
SELECT <column1, column2.....> , aggregate function(colname)  
FROM <tablename>  
WHERE <condition>  
GROUP BY <column1>  
HAVING <condition>;
```

Consider the SALE table is given below.

```
+-----+-----+-----+-----+-----+
|InvoiceNo|CarId|CustId| SaleDate | PaymentMode | EmpID| SalePrice|Commission|
+-----+-----+-----+-----+-----+
|I00001   |D001 |C0001 |2019-01-24| Credit Card | E004| 613247.00| 73589.64 |
|I00002   |S001 |C0002 |2018-12-12| Online      | E001| 590321.00| 70838.52 |
|I00003   |S002 |C0004 |2019-01-25| Cheque      | E010| 604000.00| 72480.00 |
|I00004   |D002 |C0001 |2018-10-15| Bank Finance| E007| 659982.00| 79197.84 |
|I00005   |E001 |C0003 |2018-12-20| Credit Card | E002| 369310.00| 44317.20 |
|I00006   |S002 |C0002 |2019-01-30| Bank Finance| E007| 620214.00| 74425.68 |
+-----+-----+-----+-----+-----+
6 rows in set (0.11 sec)
```

(i) Write a query to display number of cars purchased by each customer from the SALE Table.

```
mysql> SELECT CustID, COUNT(*) "Number of Cars" FROM SALE
GROUP BY CustID;
```

```
+-----+-----+
| CustID | Number of Cars |
+-----+-----+
| C0001  |                2 |
| C0002  |                2 |
| C0003  |                1 |
| C0004  |                1 |
+-----+-----+
4 rows in set (0.00 sec)
```

(ii) Write a query to display customer id and number of cars purchased if the customer purchased more than one car from the sale table.

```
mysql> SELECT CustID, COUNT(*) FROM SALE GROUP BY CustID
HAVING Count(*)>1;
```

```
+-----+-----+
| CustID | COUNT(*) |
+-----+-----+
| c0001  |         2 |
| c0002  |         2 |
+-----+-----+
2 rows in set (0.30 sec)
```

(iii) Display number of people in each category of payment mode from the table SALE.

```
mysql> SELECT PaymentMode, COUNT(PaymentMode) FROM SALE  
GROUP BY Paymentmode ORDER BY Paymentmode;
```

```
+-----+-----+  
| PaymentMode | Count(PaymentMode) |  
+-----+-----+  
| Bank Finance |          2 |  
| Cheque       |          1 |  
| Credit Card  |          2 |  
| Online       |          1 |  
+-----+-----+  
4 rows in set (0.00 sec)
```

(iv) Display the payment mode and number of payments made using that mode more than once.

```
mysql> SELECT PaymentMode, Count(PaymentMode) FROM SALE  
GROUP BY Paymentmode HAVING COUNT(*)>1 ORDER BY  
Paymentmode;
```

TOPIC – SQL QUERIES USING HAVING AND GROUP BY CLAUSE

1 Write down name of four functions that can be used with Group by?

Ans Count(), sum(), min(), max()

2. What is Group By clause?

Ans The GROUP BY Clause is utilized in SQL with the SELECT statement to organize similar data into groups. It combines the multiple records in single or more columns using some functions. Generally, these functions are aggregate functions such as min (), max (), avg (), count (), and sum () to combine into single or multiple columns.

3. Why we use Having clause?

Ans The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

4. What is the purpose of Group By clause?

Ans Group by clause is used in a Select statement in conjunction with aggregate functions to group the result based on distinct values in column.

5. You have a table "Company" having column cno, cname, department and salary. Write SQL statement to display average salary of each department.

Ans SELECT department, avg(salary) from company
Group by department;

6. Can a Group by clause be used for more than one column? If yes, given an example.

Ans Yes.

Select name, grade, class
From student
Group by Class, grade

7. Anis has given the following command to arrange the data in ascending order of date.

Select * from travel where order by tdate;

But he is not getting the desired result.

Help him by choosing the correct command.

- a. Select * from travel order by tdate;
- b. Select * from travel in ascending order;
- c. Select tdate from travel order by tdate;

Ans. Select * from travel order by tdate;

8. Find the output of the following SQL queries:

i. SELECT ROUND(7658.345,2); → O/P – 7658.35

ii. SELECT MOD(ROUND(13.9,0),3); → o/p - 2

9. Give any two differences between POWER() and SUM().

Ans: POWER(x,y) – will return value x power to y

SUM() – Aggregate function, and returns sum of values of one column

10. Find the output of the following SQL queries:

i. SELECT SUBSTR('FIT INDIA MOVEMENT',5);

→ O/P – INDIA MOVEMEMNT

ii. SELECT INSTR('ARTIFICIAL INTELLIGENCE', 'IA');

→ O/P - 8

UNSOLVED QUESTIONS

1) Identify the errors in the following queries.

- i. Select * from stock where price = NULL;
- ii. Select sum[price] from stock;

2) Mr. Das has created the following table 'Furniture'

F_id	Type	Price	Qty	Date_of_purchase
F12	Double Bed	40000	3	2020-08-19
F22	Sofa	35000	4	2021-11-23
F41	Dining Table	20000	2	2021-12-30

Write the output of the following queries:

- i. Select * from Furniture where month(Date_of_purchase) = 8;
- ii. Select F_id, Type from Furniture where year(Date_of_purchase) = 2021;

OR

Write the queries of the following:

- i. Display the details of furniture which are purchased in month of December.
- ii. Display the average price of furniture.

3) Write the output of the following:

- i. Select substr("BoardExam@2021", 4, 7);
- ii. Select length("BoardExam@2021");
- iii. Select left(upper("BoardExam@2021"),5);

OR

Consider the table 'School' whose fields are shown below.

Admno, Class, Mobile, Fees, Name

Write the queries to perform the following task.

- i. Display all names in uppercase.
- ii. Display the last two characters from column Name.
- iii. Display the lowest fees.

4) Identify the following functions:

- i. I am a mathematical function and return the remainder.
- ii. I am a string function and help to convert lower case string to upper case.

- iii. I am a date and time function, returns the month name from the specified date.
- 5) Explain the following function with examples.
- i. dayname()
 - ii. instr()
 - iii. now()

5) Explain the difference between group by and order by clause with example.

6) When we use having clause? Explain with example.

7) Write the SQL queries to do the following:

- i. Add a new column "Name" of data type Varchar(30) in table "Hotel".
- ii. Delete a record from table "emp" whose empid is 101.
- iii. Delete all the records of table "stock" along with structure.
- iv. Display the total of column "Salary" from table "emp".
- v.

8) Based on the table: "Emp" given below:

Empid	Salary
1	45000
2	50000
3	55000
4	40000
5	NULL

Write the output of the following:

- i. Select mod(Salary, 100) from emp;
- ii. Select average(Salary) from emp;
- iii. Select sum(Salary) from emp where empid > 3;
- iv. Select max(Salary) from emp;

9) Predict the output:

- i. select substr('Informatics Practices',12,5);
- ii. select year(curdate()) + year(yourbirthdate);

10) Write your birthdate in the query then write the output.

- i. select round(32.567890);
- ii. select lower('pre-board I');

11) Predict the output for following query:

i. `select pow(month(now()),2);`

ii. `select left(dayname(now()),5)`

iii. `select length('Informatics Practice Class 12');`

OR

Explain the functions with suitable example to do this:

i. To find the position of specific word or character in the given text

ii. Display the total number characters from the text

iii. To display remainder of given two numbers

12) Vats is working with functions of MySQL. Explain him the following with example:

i. To remove extra leading spaces from the text

ii. To return only day part from today's date

iii. To return average of particular column from the table

13) Om has written following queries:

i. `select count(*) from student;`

ii. `select count(std_no) from student;`

He was surprised with the output as query (i) returns 5 rows whereas Query(ii) returns only 3 rows. Explain why?

14) Which functions in MySQL extract words from specified character to n number of character from given string. Write the function names and explain them with example.

15) Anuj is student of class XII, trying to execute the following queries, help him to predict the output.

i. `select round (45.9,-2);`

ii. `select round (-101.86,0)`