



# My SQL-Built in Function

# Objective

---

In this presentation, you will learn about-

- Introduction to MySQL Functions.
  - Types of MySQL Functions
    - Numeric Functions
    - String Functions
    - Date & Time Functions
    - Aggregate Functions
-

# Working with Functions

---

## □ What is Function?

A function is a special types of command in MySQL that performs some operation on table and returns a single value as a result.

## □ Types of Functions:

- Numeric Functions
- String Functions
- Date & Time Function
- Aggregate Functions

- ❖ Numeric, String and Date-Time functions are called **Single row functions** because they can accept one row and return only one value. When applied on a table, they return a single result for every row of the queried table.
  - ❖ Aggregate Functions are called **Multiple row functions** because they operate on a set of rows to return a single value.
-

# Numeric Functions

---

These functions may accept some numeric values and performs required operation, returns numeric values as result.

Name	Purpose	Example
<b>MOD (M, N)</b>	Returns remainder of M divide by N	Select MOD(11,4) ; → 3
<b>POWER (M, N)</b> <b>POW (M, N)</b>	Returns $M^N$	Select POWER(3,2); → 9
<b>ROUND (N [,M])</b>	Returns a number rounded off up to <b>M</b> place. If <b>M</b> is -1, it rounds nearest 10. If <b>M</b> is not given, then <b>N</b> is rounded to the nearest Integer.	Select ROUND(15.193,1); →15.2 Select ROUND(15.193); →15 Select ROUND(-1.58); →-2
<b>SQRT (N)</b>	Returns square root of N	Select SQRT(25); → 5
<b>TRUNCATE(N,M)</b>	Returns number after truncating M decimal place.	Select TRUNCATE(15.79,1) → 15.7

---

# String Functions

---

## ❑ **CONCAT()**

Concatenates (Adds) two string.

**CONCAT(Str1 , Str2)**

```
mysql> SELECT CONCAT('ab' , 'cd' ) FROM DUAL;
```

→ **abcd**

```
mysql> SELECT CONCAT('Mr', Name) FROM Student;
```

➤ Concat() can be nested.

```
mysql> SELECT CONCAT(CONCAT(Name,'son of '), Fname)  
FROM Student;
```

## ❑ **LENGTH()**

Returns length of given string.

**LENGTH (Str)**

```
mysql> SELECT LENGTH('abcd' ) FROM DUAL;
```

→ **4**

```
mysql> SELECT Name, LENGTH(Name) FROM Student;
```

---

# String Functions

cont...

## ❑ LOWER() or LCASE()

Converts given string in lower case.

### LOWER (Str)

```
mysql> SELECT LOWER('ABcD' ) FROM DUAL;
```

→ abcd

```
mysql> SELECT LOWER(Name) FROM Student;
```

```
mysql> SELECT LCASE(Fname) FROM Student;
```

## ❑ UPPER() or UCASE()

Converts given string in upper case.

### UPPER (Str)

```
mysql> SELECT UPPER('abcd' ) FROM DUAL;
```

→ ABCD

```
mysql> SELECT UPPER(Name) FROM Student;
```

```
mysql> SELECT UCASE(Fname) FROM Student;
```

# String Functions

cont...

---

## □ LTRIM()

Returns string after removing leading spaces.

```
mysql> SELECT LTRIM('  abcd' ) FROM DUAL;
```

→ abcd

```
mysql> SELECT LTRIM(Name) FROM Student;
```

## □ RTRIM()

Returns string after removing trailing spaces.

```
mysql> SELECT RTRIM('abcd  ') FROM DUAL;
```

→ abcd

```
mysql> SELECT RTRIM(Name) FROM Student;
```

## □ TRIM()

Returns string after removing leading and trailing spaces.

```
mysql> SELECT TRIM('  abcd  ') FROM DUAL;
```

→ abcd

---

# String Functions cont...

---

## □ SUBSTR()

Returns a sub string of given length from specified position.

**SUBSTR (Str, position [,length])**

```
mysql> SELECT SUBSTR('MY COMPUTER', 4,3' ) → COM
```

➤ If position is negative then backward position is counted.

```
mysql> SELECT SUBSTR('ABCDEFGG' , -5, 4) FROM Student;  
→ CDEF
```

➤ If Length is omitted then up to end of the string is considered.

```
mysql> SELECT SUBSTR('ABCDEFGG' , 3) FROM Student;  
→ CDEFG
```

## □ INSTR()

Searches a string in to another string and returns its position.

**INSTR(Str1, Str2)**

```
mysql> SELECT INSTR('CORPORATE FLOOR', 'OR'); → 2
```

```
mysql> SELECT Name, INSTR(Name,'a') FROM Student;
```

---



# String Functions

cont...

## □ LEFT()

Returns leftmost string up to given length.

**LEFT (Str , length)**

```
mysql> SELECT LEFT('MYSQL', 2 ) → MY
```

```
mysql> SELECT LEFT( Name, 4) FROM Student;
```

## □ RIGHT()

Returns rightmost string up to given length.

**RIGHT (Str , length)**

```
mysql> SELECT RIGHT('MYSQL', 3 ) → SQL
```

```
mysql> SELECT RIGHT (Name, 4) FROM Student;
```

## □ MID()

Returns a substring upto given length from given position.

**MID (Str ,Pos, Length)**

```
mysql> SELECT MID('COMPUTER', 4,3 ) → PUT
```

```
mysql> SELECT MID (Name, 4,3) FROM Student;
```

Mid() is similar to Substr()

# Summery of String Functions

---

Name	Purpose	Example
CONCAT(str1, str2)	Returns concatenated string i.e. str1+str2.	Select CONCAT(Name, City) from Student;
LOWER(str) / LCASE(str)	Returns the given string in lower case.	Select LOWER('ABC'); → abc
UPPER(str) / UCASE(str)	Returns the given String in upper case.	Select UPPER('abc'); → ABC
LTRIM(str) RTRIM(str) TRIM(str)	Removes Leading/Trailing/both spaces from given string.	Select TRIM(' ABC '); → 'ABC'
LEFT(str, N) RIGHT(str, N)	Returns the (N) characters from left/right from the given string.	Select LEFT('Computer', 4); → Comp
SUBSTR(str, P, [N]) / MID (str, P, N)	Returns the substring for given position(P) and length (N). If M is (-ve) then backward position counted.	Select SUBSTR('Computer', 3, 2); → mp
INSTR(str1, str2)	Returns the index of first occurrence of str2 in str1.	Select INSTR('Common', 'm'); → 3
LENGTH(str)	Returns the length of given string	Select LENGTH('Common'); → 6

---

# Date & Time Functions

---

## □ **CURDATE() or CURRENT\_DATE()**

Returns current date of the system in **YYYY-MM-DD** format.

```
mysql> SELECT CURDATE() ;
```

→ 2014-01-30

```
mysql> SELECT CURDATE() + 10 ;
```

→ 2014-02-09

## □ **SYSDATE()**

Returns current date and time as YYYY-MM-DD HH:MM:SS

```
mysql> SELECT SYSDATE() ;
```

→ 2014-01-30 10:30:20

## □ **NOW()**

Returns current date and time as YYYY-MM-DD HH:MM:SS

```
mysql> SELECT SYSDATE() FROM DUAL
```

→ 2010-01-30 10:30:20

### **Difference between SYSDATE() & NOW()**

**NOW()** returns the time when command began to execute and does not change time during execution. Where as **SYSDATE()** changes its time continuously.

# Date & Time Functions

cont...

---

## □ DATE()

Returns date part of the given date-time expression.

### DATE (Dt)

```
mysql> SELECT DATE('2008-12-31 01:02:03') ;  
      → 2008-12-32
```

```
mysql> SELECT DATE( SYSDATE());
```

## □ YEAR()

Returns year of the given date expression.

### YEAR (Dt)

```
mysql> SELECT YEAR('2008-12-31') ; → 2008
```

```
mysql> SELECT YAER(DOB) FROM Student;
```

## □ MONTH()

Returns month of the given date expression.

### MONTH (Dt)

```
mysql> SELECT MONTH('2008-12-31') ; → 12
```

```
mysql> SELECT MONTH( CURDATE());
```

---

# Date & Time Functions

cont...

---

## □ DAYOFMONTH()

Returns day of month of the given date expression.

### DAYOFMONTH (Dt)

```
mysql> SELECT DAYOFMONTH('2008-12-31') ;
```

→ 31

```
mysql> SELECT DAYOFMONTH( CURDATE() ) ;
```

```
mysql> SELECT DAYOFMONTH( DOB) FROM Student;
```

## □ DAYNAME()

Returns the name of Week day of the given date expression.

### DAYNAME (Dt)

```
mysql> SELECT DAYNAME('2008-12-31') ;
```

→ SUNDAY

```
mysql> SELECT DAYNAME( CURDATE() ) ;
```

```
mysql> SELECT DAYNAME( DOB) FROM Student;
```

---

# Date & Time Functions

cont...

---

## □ DAYOFWEEK()

Returns day of week i.e. 1- Sunday, 2- Tuesday.. etc. of given date.

### DAYOFWEEK (Dt)

```
mysql> SELECT DAYOFWEEK('2008-12-31') ;
```

→ 1

```
mysql> SELECT DAYOFWEEK(CURDATE()) ;
```

## □ DAYOFYEAR()

Returns the day of year of the given date expression.

### DAYOFYAER (Dt)

```
mysql> SELECT DAYOFYAER('2010-02-05') ;
```

→ 36

```
mysql> SELECT DAYOFYAER( CURDATE()) ;
```

```
mysql> SELECT DAYOFYEAR( DOB) FROM Student;
```

---

# Summery of Date & Time Functions

---

Name	Purpose	Example
CURDATE() / CURRENT_DATE()	Returns the current date in <b>YYYY-MM-DD</b> format.	Select CURDATE(); → 2013-10-02
NOW()	Returns the current date & Time as YYYY-MM-DD HH:MM:SS	Select NOW(); → 2013-10-02 11:30:02
SYSDATE()	Returns the current date & Time as YYYY-MM-DD HH:MM:SS	Select SYSDATE(); → 2013-10-02 11:30:10
DATE()	Returns the date part of a date- time expression.	Select DATE(SYSDATE()); → 2013-10-02
MONTH() YEAR()	Returns the Month/Year from given date argument.	Select MONTH('2012-10-02'); → 10
DAYNAME()	Returns the name of the weekday	Select DAYNAME(CURDATE()); → SUNDAY
DAYOFMONTH()	Returns the day of month (1-31).	Select DAYOFMONTH(CURDATE());
DAYOFWEEK()	Returns the day of week (1-7).	Select DAYOFWEEK(CURDATE());
DAYOFYEAR()	Returns the day of year(1-366).	Select DAYOFYEAR(CURDATE());

---

# Aggregate Functions

---

## □ **SUM()**

Returns sum of given column in the table.

### **SUM (<Field>)**

```
mysql> SELECT SUM (Sal) FROM Emp;
```

```
mysql> SELECT SUM(Sal) FROM Emp WHERE City='Jaipur';
```

## □ **MIN()**

Returns minimum value in the given column of table.

### **MIN (<Field>)**

```
mysql> SELECT MIN (Sal) FROM Emp;
```

```
mysql> SELECT MIN(Sal) FROM Emp WHERE City='Jaipur';
```

## □ **MAX()**

Returns maximum value in the given column of table.

### **MAX (<Field>)**

```
mysql> SELECT MAX (Sal) FROM Emp;
```

```
mysql> SELECT MAX(Sal) FROM Emp WHERE City='Jaipur';
```

---



# Aggregate Functions

---

## □ **AVG()**

Returns average value of given column in the table.

### **AVG (<Field>)**

```
mysql> SELECT AVG (Sal) FROM Emp;
```

```
mysql> SELECT AVG(Sal) FROM Emo WHERE City='Jaipur';
```

## □ **COUNT()**

Returns number of values in the given column of table. It also reflect the number of record in the table.

### **COUNT (<Field| \* >)**

```
mysql> SELECT COUNT (Name) FROM Emp;
```

```
mysql> SELECT COUNT(Name) FROM Emp  
WHERE City='Jaipur';
```

```
mysql> SELECT COUNT (*) FROM Emp;
```

**→ Number of records in the Emp table**

```
mysql> SELECT COUNT(*) FROM Emp  
WHERE City='Jaipur';
```

---

# Aggregate Functions

---

Name	Purpose	Example
SUM()	Returns the sum of given column.	Select SUM(Pay) from Emp; Select Sum(Pay), Sum(Net) from Emp;
MIN()	Returns the minimum value in the given column.	Select MIN(Pay) from Emp;
MAX()	Returns the maximum value in the given column.	Select MAX(Pay) from Emp;
AVG()	Returns the Average value of the given column.	Select AVG(Pay) from Emp;
COUNT()	Returns the total number of values/ records in given column.	Select COUNT(Name) from Emp; Select COUNT(*) from Emp;

Aggregate Functions should not be used with other columns which may have multiple values in the table. The following query is illogical and wrong. Why? Think yourself....

Select sum(pay), name from Employee;