



Functions in MySQL

Learning Objectives

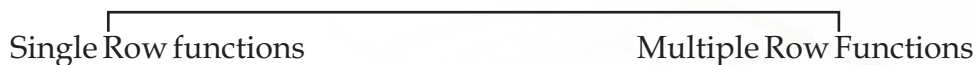
After studying this lesson the students will be able to

Distinguish between two types of functions.

State the syntax and working of most of the Numeric, String and date/Time functions.

Functions are a powerful feature of SQL. Using these functions, we can find sum of values stored in a column or convert all characters of a name to lowercase or round off salaries in a column to two decimal places and so on. MySQL supports many functions to manipulate data. We can broadly categorize functions into two types: Single Row functions and Multiple Row Functions.

Functions



Single-row functions: Single row functions operate on a single value to return a single value. They can accept one or more arguments but return only one result per row. When applied on a table, they return a single result for every row of the queried table. They are further categorized into:

- ❖ Numeric functions
- ❖ String functions
- ❖ Date and Time functions

Multiple Row Functions (also called Aggregate Functions): Multiple row functions operate on a set of rows to return a single value. Examples include SUM(), AVG() and COUNT().

(Note : Multiple Row functions will be discussed in detail in Class XII)



Let us consider the following table named Employee with 5 rows. We will be referring to it in our lesson to learn about Functions.

```
CREATE TABLE Employee (
    id            int,
    first_name    VARCHAR(15),
    last_name     VARCHAR(15),
    date_join    DATE,
    salary        DECIMAL(8,2),
    city          VARCHAR(10)
);
```

The rows in Employee table are as follows:

```
mysql> SELECT * FROM Employee;
```

id	first_name	last_name	date_join	salary	city
1	Amit	Sharma	1996-07-25	25000.00	Delhi
2	Deeksha	Verma	1995-06-27	30000.00	Pune
3	Navkiran	Ahluwalia	1990-02-20	32000.50	Delhi
4	Mamta	Sharma	1989-08-18	37500.50	Mumbai
5	Bhawna	Ahlurkar	2010-03-01	42389.50	Chennai

5 rows in set (0.00 sec)

A) Numeric Functions:

MySQL numeric functions perform operations on numeric values and return numeric values. The following table tells us about the numeric functions of MySQL and what they do.



Sno	Name & Syntax	Description	Example
1	POWER(X,Y) or POW(X,Y)	Returns the value of X raised to the power of Y.	<pre>a)mysql> SELECT POW(2,4); Result: 16 b)mysql> SELECT POW(2,-2); Result: 0.25 c)mysql> SELECT POW(-2,3); Result:-8 d)mysql> 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 +-----+-----+-----+ 5 rows in set (0.00 sec)</pre>
2	ROUND(X,D) ROUND(X)	<p>a) Rounds the argument X to D decimal places.</p> <p>b) If number of decimal places is not specified or is zero, the number rounds to the nearest integer OR (0 decimal places).</p>	<pre>a) mysql> SELECT ROUND(-1.23); Result: -1 b) mysql> SELECT ROUND(-1.58); Result: -2 c) mysql> SELECT ROUND(1.43); Result: 1 d) mysql> SELECT ROUND(6.298, 1); Result: 6.3 e) mysql> SELECT ROUND(6.235, 0); Result: 6 f) mysql> SELECT ROUND(56.235, -1); Result: 60</pre>



		<p>c) If negative value is specified for precision, it counts off that value left from the decimal point.</p> <p>d) If positive value is specified for precision, it counts off that value right from the decimal point.</p>	<p>g) <code>mysql> SELECT id,ROUND(salary,0) FROM employee;</code></p> <p>Result:</p> <pre> +-----+-----+ id round(salary,0) +-----+-----+ 1 25000 2 30000 3 32001 4 37501 5 42390 +-----+-----+ 5 rows in set (0.00 sec) </pre>
3	TRUNCATE (X,D)	Returns the number X, truncated to D decimal places. If D is 0, the result has no decimal point or fractional part. If D is negative, it causes D digits left of the decimal point of the value X to become zero.	<p>a) <code>mysql> SELECT TRUNCATE(7.543,1);</code> Result: 7.5</p> <p>b) <code>mysql> SELECT TRUNCATE(4.567,0);</code> Result: 4</p> <p>c) <code>mysql> SELECT TRUNCATE(-7.45,1);</code> Result: -7.4</p> <p>d) <code>mysql> SELECT TRUNCATE(346,-2);</code> Result: 300</p> <p>e) <code>mysql> SELECT id,TRUNCATE(salary,0) FROM employee;</code></p>



		Note: TRUNCATE does not round a number. It simply chops off digits from a number.	<pre> Result: +-----+-----+ id truncate(salary,0) +-----+-----+ 1 25000 2 30000 3 32000 4 37500 5 42389 +-----+-----+ 5 rows in set (0.00 sec) </pre>
--	--	---	--

B) String(Character) Functions

String functions operate on character type data. String functions are used to extract, change, format or alter character strings. They return either character or numeric values. The following table tells us about the Character functions of MySQL and what they do.

Sno	Name & Syntax	Description	Example
1	LENGTH(str)	Returns the length of a column or a string in bytes.	<pre> a) mysql> SELECT LENGTH ('Informatics'); Result: 11 b) mysql> SELECT LENGTH(First_Name) FROM Employee; Result: +-----+ LENGTH(First_Name) +-----+ 4 7 8 5 6 +-----+ 5 rows in set (0.00 sec) </pre>



2	CONCAT(str1, str2,...)	Returns the string that results from concatenating the arguments. May have one or more arguments.	<pre>a) mysql> SELECT CONCAT ('My', 'S', 'QL'); Result: 'MySQL' b) mysql> SELECT CONCAT('Class', NULL, 'XI'); Result: NULL c) mysql> SELECT CONCAT(First_ Name, ' ', Last_Name) FROM Employee; Result: +-----+ CONCAT(First_Name, ' ', Last_Name) +-----+ Amit Sharma Deeksha Verma Navkiran Ahluwalia Mamta Sharma Bhawna Ahlurkar +-----+ 5 rows in set (0.00 sec)</pre>
3	INSTR(str,substr)	Returns the position of the first occurrence of substring substr in string str.	<pre>a) mysql> SELECT INSTR ('Informatics', 'for'); Result: 3 b) mysql> SELECT INSTR ('Computers', 'pet'); Result: 0 c) mysql> SELECT INSTR (First_Name, 'Kiran') FROM Employee; Result: +-----+ INSTR(First_Name, 'Kiran') +-----+ 0 0 4 0 0 +-----+ 5 rows in set (0.00 sec)</pre>

4	LOWER(str) or LCASE(str)	Returns the argument (str) in lowercase i.e. it changes all the characters of the passed string to lowercase.	<pre>a) mysql> SELECT LOWER ('INFORMATICS'); Result: 'informatics' b) mysql> SELECT LOWER>Last_Name) FROM Employee; Result: +-----+ LOWER>Last_Name) +-----+ sharma verma ahluwalia sharma ahlurkar +-----+ 5 rows in set (0.00 sec)</pre>
5	UPPER(str) or UCASE(str)	Returns the argument in uppercase. i.e. it changes all the characters of the passed string to uppercase.	<pre>a) mysql> SELECT UPPER ('Informatics'); Result: 'INFORMATICS' b) mysql> SELECT UPPER>Last_Name) FROM Employee; Result: +-----+ UPPER>Last_Name) +-----+ SHARMA VERMA AHLUWALIA SHARMA AHLURKAR +-----+ 5 rows in set (0.00 sec)</pre>



6	LEFT(str,n)	Returns the specified number of characters (n) from the left side of string str.	<pre>a) mysql> SELECT LEFT('Informatics', 3); Result: 'Inf'</pre> <pre>b) mysql>select LEFT(first_name,3)FROM Employee; Result:</pre> <pre>+-----+ LEFT(first_name,3) +-----+ Ami Dee Nav Mam Bha +-----+ 5 rows in set (0.00 sec)</pre>
7	RIGHT(str,n)	Returns the specified number of characters (n) from the right side of string str.	<pre>a) mysql> SELECT RIGHT('Informatics', 4); Result: 'tics'</pre> <pre>b) mysql> select RIGHT(first_name,3) FROM Employee; Result:</pre> <pre>+-----+ RIGHT(first_name,3) +-----+ mit sha ran mta wna +-----+ 5 rows in set (0.00 sec)</pre>



8	LTRIM(str)	Removes leading spaces i.e. removes spaces from the left side of the string str.	<pre>a) mysql> SELECT LTRIM (' Informatics'); Result: 'Informatics' b) mysql> SELECT LTRIM(First_Name) FROM Employee; Result: +-----+ LTRIM(First_Name) +-----+ Amit Deeksha Navkiran Mamta Bhawna +-----+ 5 rows in set (0.00 sec)</pre>
9	RTRIM(str)	Removes trailing spaces i.e. removes spaces from the right side of the string str.	<pre>a) mysql> SELECT RTRIM ('Informatics '); Result: 'Informatics' b) mysql> SELECT RTRIM(First_Name) FROM Employee; Result: +-----+ RTRIM(First_Name) +-----+ Amit Deeksha Navkiran Mamta Bhawna +-----+ 5 rows in set (0.02 sec)</pre>



10	TRIM(str)	Removes both leading and trailing spaces from the string str.	<pre>a)mysql> SELECT TRIM(' Informatics '); Result: 'Informatics' b) mysql> SELECT TRIM(First_Name) FROM Employee; Result: +-----+ TRIM(First_Name) +-----+ Amit Deeksha Navkiran Mamta Bhawna +-----+ 5 rows in set (0.00 sec)</pre>
11	SUBSTRING (str,m,n) Or MID(str,m,n)	Returns the specified number of characters from the middle of the string. There are 3 arguments. The first argument is the source string. The second argument is the position of first character to be displayed. The third argument is the number of characters to be displayed.	<pre>a) mysql> SELECT SUBSTRING('Informatics',3); Result: 'formatics' b) mysql> SELECT SUBSTRING('Informatics' FROM 4); Result: 'ormatics' c) mysql> SELECT SUBSTRING('Informatics',3,4); Result: 'form' d) mysql> SELECT SUBSTRING('Computers', -3); Result: 'ers' e) mysql> SELECT SUBSTRING('Computers', -5, 3); Result: 'ute' f) mysql> SELECT SUBSTRING('Computers' FROM -4 FOR 2); Result: 'te'</pre>



		<p>If the third argument is missing, then starting from the position specified, the rest of the string is returned.</p> <p>It is also possible to use a negative value for the second argument ie. position (pos). In such a case, the beginning of the substring is pos characters from the end of the string,</p> <p>Note: SUBSTR is the same as SUBSTRING</p>	<pre>g) mysql> SELECT MID('Informatics', 3,4); Result: 'form'</pre> <pre>h) mysql> select MID(first_name,3,2) FROM Employee;</pre> <p>Result:</p> <pre>+-----+ MID(first_name,3,2) +-----+ it ek vk mt aw +-----+ 5 rows in set (0.00 sec)</pre>
12	ASCII(str)	<p>Returns the ASCII value of the leftmost character of the string str. Returns 0 if str is an empty string. Returns NULL if str is NULL.</p>	<pre>a) mysql> SELECT ASCII('2'); Result: 50 (ASCII value of character '2')</pre> <pre>b) mysql> SELECT ASCII('dx'); Result: 100 (ASCII value of d)</pre> <pre>c) mysql> SELECT ASCII('A'); Result: 65 (ASCII value of 'A')</pre>



C) Date and Time Functions

Date and Time functions allow us to perform many types of tasks on Date type data. The default date format in MySQL is YYYY-MM-DD.

Sno	Name & Syntax	Description	Example
1	CURDATE()	Returns the current date in YYYY-MM-DD format or YYYYMMDD format, depending on whether the function is used in a string or numeric context.	a) <code>mysql> SELECT CURDATE();</code> Result: '2010-02-26'
2	NOW()	Returns the current date and time in 'YYYY-MM-DD HH:MM:SS' or YYYYMMDDHHMSS.uuuuuu format, depending on whether the function is used in a string or numeric context.	a) <code>mysql> SELECT NOW();</code> Result: '2010-02-26 21:30:26'
3	SYSDATE()	Returns the current date and time in 'YYYY-MM-DD HH:MM:SS' or YYYYMMDDHHMSS.uuuuuu format, depending on whether the function	a) <code>mysql> SELECT SYSDATE();</code> Result: '2010-02-26 21:30:26' b) <code>mysql> SELECT SYSDATE() + 0;</code> Result: 20100226213026.000000



		<p>is used in a string or numeric context.</p> <p>Note : SYSDATE() returns the time at which the function executes. SYSDATE() differs from NOW() which returns a constant time that indicates the time at which the statement began to execute.</p> <p>* For difference between SYSDATE() and NOW() refer to NOTE at the end of this table.</p>	
4	DATE(expr)	Extracts the date part of a date or datetime expression	<p>a) <code>mysql> SELECT DATE('2010-02-26 01:02:03');</code> Result: '2010-02-26'</p> <p>b) <code>mysql> SELECT DATE('2009-10-16 01:02:03');</code> Result: '2009-10-16'</p>
5	MONTH(date)	Returns the numeric month from the date passed, in the range 0 to 12. It returns 0 for dates such as '0000-00-00' or '2010-00-00' that have a zero month part.	<p>a) <code>mysql> SELECT MONTH('2010-02-26');</code> Result: 2</p> <p>b) <code>mysql> select id,date_join,month(date_join) from employee;</code></p>



			<pre>Result: +----+-----+-----+ id date_join month(date_join) +----+-----+-----+ 1 1996-07-25 7 2 1995-06-27 6 3 1990-02-20 2 4 1989-08-18 8 5 2010-03-01 3 +----+-----+-----+ 5 rows in set (0.00 sec)</pre>
6	YEAR(date)	Returns the year for date passed in the range 0 to 9999. Returns values like 1998, 2010,1996 and so on.	<pre>a) mysql> SELECT YEAR('2010-02-26'); Result: 2010 b) mysql> SELECT id,date_join,year (date_join) from employee; Result: +----+-----+-----+ id date_join year(date_join) +----+-----+-----+ 1 1996-07-25 1996 2 1995-06-27 1995 3 1990-02-20 1990 4 1989-08-18 1989 5 2010-03-01 2010 +----+-----+-----+ 5 rows in set (0.00 sec)</pre>
7	DAYNAME (date)	If you want to know which day you were born on. Was it a Monday or a Friday? Use DAYNAME function. It	<pre>a) mysql> SELECT YEAR('2009-07-21'); Result: 'Tuesday' b) mysql> Select id,date_join,dayname (date_join) from employee;</pre>



		returns the name of the weekday for the date passed	<pre>Result: +----+-----+-----+ id date_join dayname(date_join) +----+-----+-----+ 1 1996-07-25 Thursday 2 1995-06-27 Tuesday 3 1990-02-20 Tuesday 4 1989-08-18 Friday 5 2010-03-01 Monday +----+-----+-----+ 5 rows in set (0.00 sec)</pre>
8	DAYOFMONT H(date)	Returns the day of the month in the range 0 to 31.	<pre>a) mysql> SELECT DAYOFMONTH('2009-07-21'); Result: 21 b) mysql> select id,date_join,dayofmonth(date_join) from employee; Result: +----+-----+-----+ id date_join dayofmonth(date_join) +----+-----+-----+ 1 1996-07-25 25 2 1995-06-27 27 3 1990-02-20 20 4 1989-08-18 18 5 2010-03-01 1 +----+-----+-----+ 5 rows in set (0.00 sec)</pre>
9	DAYOFWEEK (date)	Returns the day of week in number as 1 for Sunday, 2 for Monday and so on.	<pre>a) mysql> SELECT DAYOFWEEK('2009-07-21'); Result: 3 b) mysql> select id,date_join,dayofweek(date_join) from employee;</pre>



			<pre> Result: +----+-----+-----+ id date_join dayofweek(date_join) +----+-----+-----+ 1 1996-07-25 5 2 1995-06-27 3 3 1990-02-20 3 4 1989-08-18 6 5 2010-03-01 2 +----+-----+-----+ 5 rows in set (0.00 sec) </pre>
10	DAYOFYEAR (date)	Return the day of the year for the given date in numeric format in the range 1 to 366.	<pre> a) mysql> SELECT DAYOFYEAR('2009-07-21'); Result: 202 b) mysql> SELECT DAYOFYEAR('2009-01-01'); Result: 1 c) mysql> select id,date_join,dayofyear (date_join) from employee; Result: +----+-----+-----+ id date_join dayofyear(date_join) +----+-----+-----+ 1 1996-07-25 207 2 1995-06-27 178 3 1990-02-20 51 4 1989-08-18 230 5 2010-03-01 60 +----+-----+-----+ 5 rows in set (0.00 sec) </pre>



Note**Difference between NOW() and SYSDATE()**

(Sleep(argument) pauses for the number of seconds given by the argument.)

```
mysql> select now(),sleep(20),now();
```

now()	sleep(20)	now()
2010-05-08 14:57:42	0	2010-05-08 14:57:42

1 row in set (20.03 sec)

Even after a pause of 20 seconds induced by sleep(20),now() shows the same time i.e. the time at which the statement began to execute.

```
mysql> select sysdate(),sleep(20),sysdate();
```

sysdate()	sleep(20)	sysdate()
2010-05-08 14:58:32	0	2010-05-08 14:58:52

1 row in set (20.00 sec)

After 20 seconds induced by sleep(20), sysdate() shows time incremented by 20 seconds.

Summary

1. Functions perform some operations and return a value.
2. Single row functions operate on a single value to return a single value.
3. Multiple Row functions operate on a set of rows to return a single value.
4. Numeric functions perform operations on numeric values and return numeric values.
5. String functions operate on character type data. They return either character or numeric values.
6. Date and Time functions allow us to manipulate Date type data.



Multiple Choice Questions

- 1) _____ functions operate on a single value to return a single value
 - (a) Multiple Row
 - (b) Aggregate
 - (c) Single Row
 - (d) Summation
- 2) SUM, AVG, COUNT are examples of _____ functions.
 - (a) Date
 - (b) String
 - (c) Multiple Row
 - (d) Single Row
- 3) SELECT POW(-3,2) will display the output:
 - (a) -6
 - (b) -9
 - (c) 9
 - (d) 6
- 4) SELECT TRUNCATE(7.956,2) will result in
 - (a) 7.95
 - (b) 7.96
 - (c) 8
 - (d) 8.0
- 5) INSTR(str,str2) returns the position of the first occurrence of
 - (a) Str in "MySQL"
 - (b) Str in str2
 - (c) str2 in str
 - (d) str2 in "SQL"



- 6) **Any String function returns**
- (a) Only string
 - (b) Only number
 - (c) String or number
 - (d) String, number or date type data.

Answer the following questions.

1. Define a Function.
2. List 3 categories of single row functions. Give two examples in each category.
3. How are numeric functions different from String functions?
4. Which function is used to display the system date?
5. Which Date function displays the result like "Monday" or "Tuesday" etc.
6. Name a
 - i) date function that returns a number.
 - ii) String function that returns a number.
 - iii) date function that returns a date.
7. Write SQL statements to do the following:
 - a) Using the three separate words "We," "study," and "MySQL," produce the following output:
"We study MySQL"
 - b) Use the string "Internet is a boon" and extract the string "net".
 - c) Display the length of the string "Informatics Practices".
 - d) Display the position of "My" in "Enjoying MySQL".
 - e) Display the name of current month.
 - f) Display the date 10 years from now. Label the column "Future."
 - g) Display the day of week on which your birthday will fall or fell in 2010.



8. Write the output that the following statements will produce:
- `SELECT ROUND(7.3456, 2);`
 - `SELECT TRUNCATE(2.3456, 2);`
 - `SELECT DAYOFMONTH('2009-08-25');`
 - `SELECT MONTH('2010-02-26');`
 - `SELECT RIGHT('Informatics', 4);`

Lab Exercises

1. Create the following table named "Charity" and write SQL queries for the tasks that follow:

Table: Charity

P_Id	LastName	FirstName	Address	City	Contribution
1	Bindra	Jaspreet	5B, Gomti Nagar	Lucknow	3500.50
2	Rana	Monica	21 A, Bandra	Mumbai	2768.00
3	Singh	Jatinder	8, Punjabi Bagh	Delhi	2000.50
4	Arora	Satinder	K/1, Shere Punjab Colony	Mumbai	1900.00
5	Krishnan	Vineeta	A-75, Adarsh Nagar		

(Contribution is in Rs.)

- Display all first names in lowercase
- Display all last names of people of Mumbai city in uppercase
- Display Person Id along with First 3 characters of his/her name.
- Display first name concatenated with last name for all the employees.
- Display length of address along with Person Id
- Display last 2 characters of City and Person ID.
- Display Last Names and First names of people who have "at" in the second or third position in their first names.
- Display the position of 'a' in Last name in every row.



- IX. Display Last Name and First name of people who have "a" as the last character in their First names.
- X. Display the first name and last name concatenated after removing the leading and trailing blanks.
- XI. Display Person Id, last names and contribution rounded to the nearest rupee of all the persons.
- XII. Display Person Id, last name and contribution with decimal digits truncated of all the persons.
- XIII. Display Last name, contribution and a third column which has contribution divided by 10. Round it to two decimal points.

2. Consider the table "Grocer" and write SQL queries for the tasks that follow:

Table: Grocer

Item_Id	ItemName	UnitPrice	Quantity (kg)	Date_Purchase
1	Rice	52.50	80	2010-02-01
2	Wheat	25.40	50	2010-03-09
3	Corn	50.80	100	2010-03-11
4	Semolina	28.90	50	2010-01-15

(Unit Price is per kg price)

- I. Display Item name, unit price along with Date of purchase for all the Items.
- II. Display Item name along with Month (in number) when it was purchased for all the items.
- III. Display Item name along with year in which it was purchased for all the items.
- IV. Display Item Id, Date of Purchase and day name of week (e.g. Monday) on which it was purchased for all the items.
- V. Display names of all the items that were purchased on Mondays or Tuesdays.
- VI. Display the day name of the week on which Rice was purchased.
- VII. Display the Item name and unit price truncated to integer value (no decimal digits) of all the items.
- VIII. Display current date

