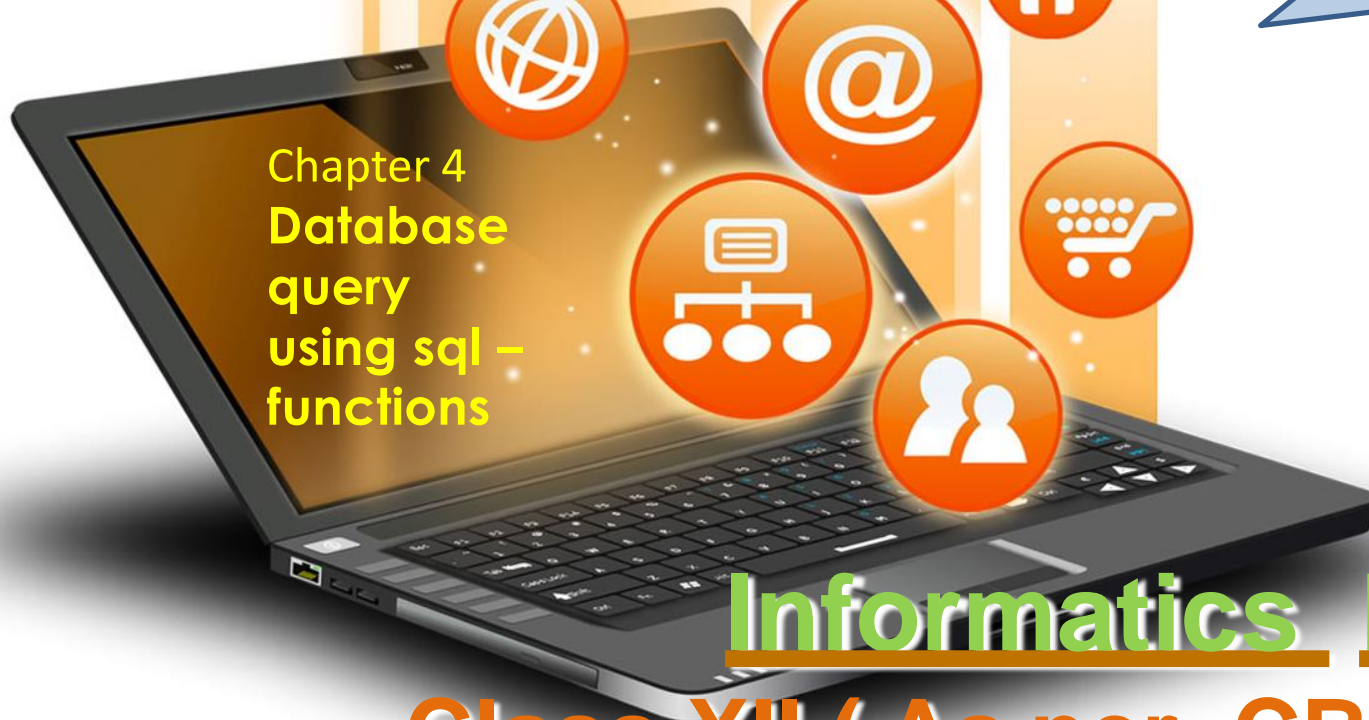


New  
syllabus  
2021-22



# Informatics Practices Class XII ( As per CBSE Board)

Visit : [python.mykvs.in](http://python.mykvs.in) for regular updates



# SQL functions

Basically, it is a set of SQL statements that accept only input parameters, perform actions and return the result. A function can return an only a single value or a table. Functions are not alternate to sql commands but are used as a part of sql command(generally select command).

## Types of Function(System defined)

A **scalar function** is a function that operates on scalar values -- that is, it takes one (or more) input values as arguments directly and returns a value. Maths, text, date functions etc. *These functions can be applied over column(s) of a table to perform relevant operation on value of each record.*

*For e.g. select left(name,4) from student;*

*Will display 4 left side letters of each row of name field from student table.*

An **aggregate function** is a function that operates on aggregate data -- that is, it takes a complete set of data as input and returns a value that is computed from all the values in the set. E.g. max(), min(), count(), sum(), avg(). Generally these are used for report preparation & mostly used with group by and having clause.



## SQL functions

**Mathematical functions** – Perform operation over numeric value

**POWER()** – power() returns the value of a number raised to the power of another number. The synonym of power() is pow().

Syntax - pow(m,n)

m      A number which is the base of the exponentiation.

n      A number which is the exponent of the exponentiation.

E.g.

```
Mysql> select pow(2,3);
```

```
Mysql>8
```

```
Mysql> select pow(2.37,3.45);
```

```
Mysql>19.6282.....
```



# SQL functions

## Mathematical functions

**ROUND()** – the round() function returns a number rounded to a certain number of decimal places.

Syntax - ROUND(column\_name,decimals)  
column\_name -Required. The field to round.

decimals -Required, Specifies the number of decimals to be returned.

454.352 ← Value to be rounded  
| | | | |  
-2-1 0 1 2 3 ← Decimal places

Decimal places position value is rounded to next integer ,if its next right side number is  $\geq 5$

Default decimal place is 0 position if we not specify

```
mysql> select round(454.352,2);
+-----+
| round(454.352,2) |
+-----+
|           454.35 |
+-----+
1 row in set (0.00 sec)

mysql> select round(454.352,0);
+-----+
| round(454.352,0) |
+-----+
|           454 |
+-----+
1 row in set (0.00 sec)

mysql> select round(454.352,-1);
+-----+
| round(454.352,-1) |
+-----+
|           450 |
+-----+
1 row in set (0.00 sec)

mysql> select round(454.352,-2);
+-----+
| round(454.352,-2) |
+-----+
|           500 |
+-----+
1 row in set (0.00 sec)
```



# SQL functions

## Mathematical functions

**MOD()** – The MOD() function returns the remainder of one number divided by another. The following shows the syntax of the MOD() function:

Syntax - MOD(dividend,divisor)

Dividend - is a literal number or a numeric expression to divide.

Divisor- is a literal number or a numeric expression by which to divide the dividend.

E.g.

```
Mysql> SELECT MOD(11, 3);
```

```
Mysql>2
```

```
Mysql> SELECT MOD(10.5, 3);
```

```
Mysql>1.5
```



## SQL functions

**Text functions-** Perform operation over string values.

**UPPER()** – UPPER(str)

Returns the string str with all characters changed to uppercase.

```
mysql> SELECT UPPER('Tej');
```

```
-> 'TEJ'
```

**UCASE(str)**-UCASE() is a synonym for UPPER().

**LOWER(str)**-Returns the string str with all characters changed to lowercase

```
mysql> SELECT LOWER('QUADRATICALLY');
```

```
-> 'quadratically'
```

**LCASE(str)**

**LCASE()** is a synonym for LOWER().



# SQL functions

**Text functions-** Perform operation over string values.

**SUBSTRING(str,pos) - SUBSTRING(str FROM pos),**

**SUBSTRING(str,pos,len)- SUBSTRING(str FROM pos FOR len)**

The forms without a len argument return a substring from string str starting at position pos. The forms with a len argument return a substring len characters long from string str, starting at position pos. The forms that use FROM are standard SQL syntax. It is also possible to use a negative value for pos. In this case, the beginning of the substring is pos characters from the end of the string, rather than the beginning.

```
mysql> SELECT SUBSTRING('practically',5);
```

```
-> 'tically'
```

```
mysql> SELECT SUBSTRING('foofarbar' FROM 4);
```

```
-> 'farbar'
```

```
mysql> SELECT SUBSTRING('Quadratically',5,6);
```

```
-> 'ratica'
```

```
mysql> SELECT SUBSTRING('Aakila', -3);
```

```
-> 'ila'
```

```
mysql> SELECT SUBSTRING('Aakila', -5, 3);
```

```
-> 'aki'
```

```
mysql> SELECT SUBSTRING('Aakila' FROM -4 FOR 2);
```

```
-> 'ki'
```

**MID(str,pos,len)**

**MID(str,pos,len) is a synonym for  
SUBSTRING(str,pos,len),substr()**



## SQL functions

**Text functions**- Perform operation over string values.

**LENGTH(str)** - Returns the length of the string str

```
mysql> SELECT LENGTH('text');
```

-> 4

**LEFT(str,len)** - Returns the leftmost len characters from the string str, or NULL if any argument is NULL.

```
mysql> SELECT LEFT('Toolbar', 4);
```

-> 'Tool'

**RIGHT(str,len)**-Returns the rightmost len characters from the string str, or NULL if any argument is NULL.

```
mysql> SELECT RIGHT('Toolbar', 3);
```

-> 'bar'





## SQL functions

**Text functions-** Perform operation over string values.

**INSTR(str,substr)**-Returns the position of the first occurrence of substring substr in string str.

```
mysql> SELECT INSTR('Toobarbar', 'bar');
```

-> 4

```
mysql> SELECT INSTR('xbar', 'ybar');
```

-> 0



# SQL functions

**Text functions-** Perform operation over string values.

**LTRIM(str)**-Returns the string str with leading space characters removed.

```
mysql> SELECT LTRIM(' Toolbar');  
-> 'Toolbar'
```

**RTRIM(str)**-Returns the string str with trailing space characters removed.

```
mysql> SELECT RTRIM('Toolbar ');  
-> 'Toolbar'
```

**TRIM([{BOTH | LEADING | TRAILING} [remstr] FROM] str)**- Returns the string str with all remstr prefixes or suffixes removed. If none of the specifiers BOTH, LEADING, or TRAILING is given, BOTH is assumed.

```
mysql> SELECT TRIM(' tool ');  
-> 'bar'
```

```
mysql> SELECT TRIM(LEADING 'x' FROM 'xxxtoolxxx');  
-> 'toolxxx'
```

```
mysql> SELECT TRIM(BOTH 'x' FROM 'xxxtoolxxx');  
-> 'tool'
```

```
mysql> SELECT TRIM(TRAILING 'xyz' FROM 'toolxxx');  
-> 'tool'
```



## SQL functions

**Date functions-** Perform operation over date values.

**NOW()-**Returns the current date and time as a value in 'YYYY-MM-DD hh:mm:ss' or YYYYMMDDhhmmss format, depending on whether the function is used in string or numeric context.

```
mysql> SELECT NOW();  
-> '2020-04-05 23:50:26'  
mysql> SELECT NOW() + 0;  
-> 20200415235026.000000
```

Here +0 means +0 second

**DATE(expr)-**Extracts the date part of the date or datetime expression expr.

```
mysql> SELECT DATE('2003-12-31 01:02:03');  
-> '2003-12-31'
```



## SQL functions

**Date functions-** Perform operation over date values.

**MONTH(date)**-Returns the month for date, in the range 1 to 12 for January to December, or 0 for dates such as '0000-00-00' or '2008-00-00' that have a zero month part.

```
mysql> SELECT MONTH('2008-02-03');
```

-> 2

**MONTHNAME(date)**-Returns the full name of the month for date.

```
mysql> SELECT MONTHNAME('2008-02-03');
```

-> 'February'



## SQL functions

**Date functions-** Perform operation over date values.

**YEAR(date)**-Returns the year for date, in the range 1000 to 9999, or 0 for the “zero” date.

```
mysql> SELECT YEAR('1987-01-01');  
-> 1987
```

**DAY(date)**-Returns the day of the month for date, in the range 1 to 31, or 0 for dates such as '0000-00-00' or '2008-00-00' that have a zero day part.

```
mysql> SELECT DAYOFMONTH('2007-02-03');  
-> 3
```

**DAYNAME(date)**-Returns the name of the weekday for date.

```
mysql> SELECT DAYNAME('2007-02-03');  
-> 'Saturday'
```



## SQL functions

**Aggregate Functions & NULL-** Perform operation over set of values

Consider a table Emp having following records as-

Null values are excluded while (avg) aggregate function is used

Emp		
Code	Name	Sal
E1	Mohak	NULL
E2	Anuj	4500
E3	Vijay	NULL
E4	Vishal	3500
E5	Anil	4000

### SQL Queries

mysql> Select Sum(Sal) from EMP;

Result of query

12000

mysql> Select Min(Sal) from EMP;

3500

mysql> Select Max(Sal) from EMP;

4500

mysql> Select Count(Sal) from EMP;

3

mysql> Select Avg(Sal) from EMP;

4000

mysql> Select Count(\*) from EMP;

5

Visit : [python.mykvs.in](http://python.mykvs.in) for regular updates