

FIRST LAB

STRING FUNCTION

ASCII(x);

Returns the ASCII value of the character x.

Select ASCII ('hai') ;

```
ASCII('hai')
```

```
104
```

CHR(x);0

,

Returns the character with the ASCII value of x.

select CHAR(65) ;

```
CHAR(65)  
)
```

```
A
```

CONCAT(x, y);

Concatenates the strings x and y and returns the appended string.

SELECT CONCAT('MY' , 'SQL') ;

```
SELECT CONCAT('MY' ,  
'SQL')
```

```
MYSQL
```

INSERT(STRING ,POSITION ,NUMBER,SUBSTRING);

This function insert a substring into a string at a specified position for a certain number of characters

Syntax: `SELECT INSERT ( 'KITSCOLLEGE' , 1,4,'HAI' ) ;`

```
INSERT ('KITSCOLLEGE' , 1,4,'HAI'
)
```

HAICOLLEGE

- The first position of string is 1.
- If position is not within the length of string , INSERT() will return string.
- If number is not within the length of the rest of string , INSERT() will replace string , starting from position until the end of string.

LENGTH(x);

Returns the number of characters in x.

`SELECT LENGTH('KITS') ;`

```
SELECT
LENGTH('KITS')
```

4

LOWER(x) / LCASE(X)

Converts the letters in x to lowercase and returns that string.

`SELECT LOWER('KITS')`

```
LOWER('KITS'
)
```

kits

LPAD(x, width, [pad_string]) ;

Pads x with spaces to the left, to bring the total length of the string up to width characters.

```
SELECT LPAD('KITSCOLLEGE',20,'ABC')
;
LPAD ('KITSCOLLEGE',20,'ABC')
```

```
ABCABCABCKITSCOLLEGE
```

LTRIM([, trim_string]);

Trims characters or spaces from the left of **x**.

```
SELECT LTRIM('    KITS')
```

```
LTRIM('
KITS')
```

```
KITS
```

RTRIM([, trim_string]);

Trims characters or spaces from the right of **x**.

```
SELECT RTRIM('KITS    ')
```

```
RTRIM('KITS
')
```

```
KITS
```

POSITION() / LOCATE()

This function returns the position of the first occurrence of a substring in a string.

Syntax: POSITION (SUBSTRING IN STRING) ;

```
SELECT POSITION('C'IN'KITSCOLLEGE')
;
```

```
POSITION('C'IN'KITSCOLLEGE')
```

POSITION() performs a case insensitive search. The first position in string is 1 . If substring is not found within the string , the position will return 0(zero) .

REPLACE(x, search_string, replace_string);

This function replaces all occurrence of a specified string. Replace() performs a case_sensitive replacement.

Syntax: REPLACE(string , from_string , to_string) ;

SELECT REPLACE('KITSCOLLEGE', 'L','G') ;

```
REPLACE('KITSCOLLEGE',
'L','G')
```

```
KITSCOGGE
```

RPAD(x, width [, pad_string]);

Pads x to the right.

SELECT RPAD('KITSCOLLEGE',20,'ABC') ;

```
RPAD('KITSCOLLEGE',20,'ABC')
```

```
KITSCOLLEGEABCABCABC
```

REVERSE(x) ;

This function reverse a string and returns the result.

```
SELECT REVERSE('KITSCOLLEGE') +
;
```

SUBSTR(x, start [, length]);

Returns a substring of **x** that begins at the position specified by **start**. An optional length for the substring may be supplied.

```
SELECT SUBSTR('KITSCOLLEGE',5,4)
;
COLL
```

STRCMP()

This function test whether two strings are same .

Syntax: STRCMP(string1,string2);

If string1 equal to string2 returns 0 , if string1 < string2 returns -1 else returns 1.

```
SELECT STRCMP('KITS' , 'KITS')
;
0
```

TRIM()

Trims spaces from the left and right of **x**.

```
SELECT TRIM('  KITS  ')
```

```
KIT
S
```

UPPER(x) / UCASE()

Converts the letters in **x** to uppercase and returns that string.

```
SELECT
UCASE('kits')
KITS
```

NUMERIC FUNCTION

Numeric Functions are used to perform operations on numbers and return numbers. Following are the numeric functions defined in SQL:

1. **ABS()**: It returns the absolute value of a number.

Syntax: `SELECT ABS (-243.5) ;`

Output: 243.5

2. **ACOS()**: It returns the cosine of a number.

Syntax: `SELECT ACOS (0.25) ;`

Output: 1.318116071652818

3. **CEIL()**: It returns the smallest integer value that is greater than or equal to a number.

Syntax: `SELECT CEIL (25.75) ;`

Output: 26

4. **CEILING()**: It returns the smallest integer value that is greater than or equal to a number.

Syntax: `SELECT CEILING (25.75) ;`

Output: 26

5. **FLOOR()**: It returns the largest integer value that is less than or equal to a number.

Syntax: `SELECT FLOOR (25.75) ;`

Output: 25

6. **GREATEST()**: It returns the greatest value in a list of expressions.

Syntax: `SELECT GREATEST (30, 2, 36, 81, 125) ;`

Output: 125

7. **LEAST()**: It returns the smallest value in a list of expressions.

Syntax: SELECT LEAST(30, 2, 36, 81, 125);

Output: 2

8. **LOG10():** It returns the base-10 logarithm of a number.

Syntax: SELECT LOG(2);

Output: 0.6931471805599453

9. **LOG2():** It returns the base-2 logarithm of a number.

Syntax: SELECT LOG2(6);

Output: 2.584962500721156

10. **MOD():** It returns the remainder of n divided by m.

Syntax: SELECT MOD(18, 4);

Output: 2

11. **PI():** It returns the value of PI displayed with 6 decimal places.

Syntax: SELECT PI();

Output: 3.141593

12. **POW():** It returns m raised to the nth power.

Syntax: SELECT POW(4, 2);

Output: 16

13. **RAND():** It returns a random number.

Syntax: SELECT RAND();

Output: 0.33623238684258644

14. **ROUND():** It returns a number rounded to a certain number of decimal places.

Syntax: SELECT ROUND(5.553);

Output: 6

15. **TRUNCATE():** It returns 7.53635 truncated to 2 places right of the decimal point.

Syntax: SELECT TRUNCATE(7.53635, 2);

Output: 7.53

16. **SQRT():** It returns the square root of a number.

Syntax: SELECT SQRT(25);

Output: 5

CURDATE(): This function returns current date as "yyyy-mm-dd" format if used in string context and as yyymmdd format if used in numeric context. The **CURRENTDATE()** function is a synonym for the **CURDATE()**. Also **CURRENT_DATE** is synonym for the **CURDATE()**.

```
SELECT CURDATE() ;
```

2019-12-23

OR

```
SELECT CURRENT_DATE() ;
```

2019-12-23

CURRENT_TIME(): The **Current_Time** returns current time . **CURTIME** and **CURRENT_TIME** are same.

```
SELECT CURRENT_TIME()  
;
```

15:53:18

DATE() : The **DATE()** function extracts the date value from a date or date time expression.

```
SELECT DATE('2017-12-21')  
;
```

2017-12-21

DATEDIFF(): Returns the difference in days between two date values.

```
SELECT DATEDIFF('2017-12-21' , '2017-12-14') ;
```

7

DAYNAME(): Returns the weekday name for a date value.

```
SELECT DAYNAME('2019-01-30') ;
```

Wednesday

DAYOFMONTH() : Returns the day portion of a date value (a number from 1 to 31)

```
SELECT DAYOFMONTH('2019-01-30')  
;
```

30

DAYOFWEEK(): Returns the weekday index for a date(a number from 1 to 7)


```
SELECT DAYOFWEEK('2019-01-30')  
;  
4
```

DAYOFYEAR() : Returns the day of the year a date value(a number from 1 to 365)

```
SELECT DAYOFYEAR('2019-06-15')  
;  
166
```

EXTRACT() : Extracts a part from a date.

```
SELECT EXTRACT(MONTH FROM '2019-06-15')  
;  
6
```