<https://www.geeksforgeeks.org/sql-null-functions/>

**SQL Inbuilt Functions**

**1)Null Function**

**2) Numeric Function**

**3)String Function**

**4)Advanced Function**

**1 SQL | NULL functions**

Following are the NULL functions defined in SQL:

1. **ISNULL():**The ISNULL function have different uses in SQL Server and MySQL. In SQL Server, ISNULL() function is used to replace NULL values.  
   **Syntax:**
2. SELECT column(s), ISNULL(column\_name, value\_to\_replace)

FROM table\_name;

Example:  
Consider the following Employee table,

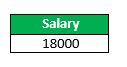


**Query:** Find the sum of salary of all Employee, if Salary of any employee is not available (or NULL value), use salary as 10000.

SELECT SUM(ISNULL(Salary, 10000) AS Salary

FROM Employee;

Output:



In MySQL, ISNULL() function is used to test whether an expression is NULL or not. If the expression is NULL it returns TRUE, else FALSE.  
**Syntax:**

SELECT column(s)

FROM table\_name

WHERE ISNULL(column\_name);

Example:  
Consider the following Employee table,



**Query:** Fetch the name of all employee whose salary is available in the table (not NULL).

SELECT Name

FROM Employee

WHERE ISNULL(Salary);

Output:

https://media.geeksforgeeks.org/wp-content/uploads/4-12.jpg

1. **IFNULL():**This function is available in MySQL, and not in SQL Server or Oracle. This function take two arguments. If the first argument is not NULL, the function returns the first argument. Otherwise, the second argument is returned. This function is commonly used to replace NULL value with another value.  
   **Syntax:**
2. SELECT column(s), IFNULL(column\_name, value\_to\_replace)

FROM table\_name;

Example:  
Consider the following Employee table,

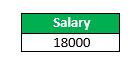


**Query:** Find the sum of salary of all Employee, if Salary of any employee is not available (or NULL value), use salary as 10000.

SELECT SUM(IFNULL(Salary, 10000) AS Salary

FROM Employee;

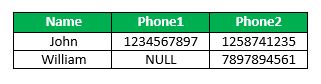
Output:



1. **COALESCE():**COALESCE function in SQL returns the first non-NULL expression among its arguments. If all the expressions evaluate to null, then the COALESCE function will return null.  
   Syntax:
2. SELECT column(s), CAOLESCE(expression\_1,....,expression\_n)

FROM table\_name;

Example:  
Consider the following Contact\_info table,



**Query:** Fetch the name, contact number of each employee.

SELECT Name, COALESCE(Phone1, Phone2) AS Contact

FROM Contact\_info;

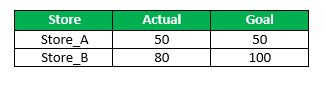
Output:



1. **NULLIF():**The NULLIF function takes two argument. If the two arguments are equal, then NULL is returned. Otherwise the first argument is returned.  
   Syntax:
2. SELECT column(s), NULLIF(expression1, expression2)

FROM table\_name;

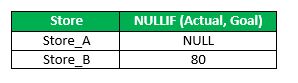
Example:  
Consider the following Sales table,



SELECT Store, NULLIF(Actual, Goal)

FROM Sales;

Output:



**2 SQL | Numeric Functions**

**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

SQL> SELECT ABS(-10);

+--------------------------------------+

| ABS(10)

+--------------------------------------+

| 10

+--------------------------------------+

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

**Syntax:**  SELECT ACOS(0.25);

**Output:**1.318116071652818

1. **ASIN():** It returns the arc sine of a number.

**Syntax:** SELECT ASIN(0.25);

**Output:**0.25268025514207865

1. **ATAN():** It returns the arc tangent of a number.

**Syntax:** SELECT ATAN(2.5);

**Output:**1.1902899496825317

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

**Syntax:** SELECT CEIL(25.75);

**Output:**26

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

**Syntax:** SELECT CEILING(25.75);

**Output:**26

1. **COS():** It returns the cosine of a number.

**Syntax:** SELECT COS(30);

**Output:**0.15425144988758405

1. **COT():** It returns the cotangent of a number.

**Syntax:** SELECT COT(6);

**Output:**-3.436353004180128

1. **DEGREES():** It converts a radian value into degrees.

**Syntax:** SELECT DEGREES(1.5);

**Output:**85.94366926962348

SQL>SELECT DEGREES(PI());

+------------------------------------------+

| DEGREES(PI())

+------------------------------------------+

| 180.000000

+------------------------------------------+

1. **DIV():** It is used for integer division.

**Syntax:** SELECT 10 DIV 5;

**Output:**2

1. **EXP():** It returns e raised to the power of number.

**Syntax:** SELECT EXP(1);

**Output:**2.718281828459045

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

**Syntax:** SELECT FLOOR(25.75);

**Output:**25

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

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

**Output:**125

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

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

**Output:**2

1. **LN():** It returns the natural logarithm of a number.

**Syntax:** SELECT LN(2);

**Output:**0.6931471805599453

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

**Syntax:** SELECT LOG(2);

**Output:**0.6931471805599453

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

**Syntax:** SELECT LOG2(6);

**Output:**2.584962500721156

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

**Syntax:** SELECT MOD(18, 4);

**Output:**2

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

**Syntax:** SELECT PI();

**Output:**3.141593

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

**Syntax:** SELECT POW(4, 2);

**Output:**16

1. **RADIANS():** It converts a value in degrees to radians.

**Syntax:** SELECT RADIANS(180);

**Output:**3.141592653589793

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

**Syntax:** SELECT RAND();

**Output:**0.33623238684258644

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

**Syntax:** SELECT ROUND(5.553);

**Output:**6

1. **SIGN():** It returns a value indicating the sign of a number.

**Syntax:** SELECT SIGN(255.5);

**Output:**1

1. **SIN():** It returns the sine of a number.

**Syntax:** SELECT SIN(2);

**Output:**0.9092974268256817

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

**Syntax:** SELECT SQRT(25);

**Output:**5

1. **TAN():** It returns the tangent of a number.

**Syntax:** SELECT TAN(1.75);

**Output:**-5.52037992250933

1. **ATAN2():** It returns the arctangent of the x and y coordinates, as an angle and expressed in radians.

**Syntax:** SELECT ATAN2(7);

**Output:**1.42889927219073

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

**Syntax:** SELECT TRUNCATE(7.53635, 2);

**Output:**7.53

**3) SQL | String functions**

**String functions**  
are used to perform an operation on input string and return an output string.  
Following are the string functions defined in SQL:

1. **ASCII():** This function is used to find the ASCII value of a character.

**Syntax:** SELECT ascii('t');

**Output:** 116

1. **CHAR\_LENGTH():** This function is used to find the length of a word.

**Syntax:** SELECT char\_length('Hello!');

**Output:** 6

1. **CHARACTER\_LENGTH():** This function is used to find the length of a line.

**Syntax:** SELECT CHARACTER\_LENGTH('geeks for geeks');

**Output:** 15

1. **CONCAT():** This function is used to add two words or strings.

**Syntax:** SELECT 'Geeks' || ' ' || 'forGeeks' FROM dual;

**Output:** ‘GeeksforGeeks’

1. **CONCAT\_WS():** This function is used to add two words or strings with a symbol as concatenating symbol.

**Syntax:** SELECT CONCAT\_WS('\_', 'geeks', 'for', 'geeks');

**Output:** geeks\_for\_geeks

1. **FIND\_IN\_SET():** This function is used to find a symbol from a set of symbols.

**Syntax:** SELECT FIND\_IN\_SET('b', 'a, b, c, d, e, f');

**Output:** 2

1. **FORMAT():** This function is used to display a number in the given format.

**Syntax:** Format("0.981", "Percent");

**Output:** ‘98.10%’

1. **INSERT():** This function is used to insert the data into a database.

**Syntax:** INSERT INTO database (geek\_id, geek\_name) VALUES (5000, 'abc');

**Output:** successfully updated

1. **INSTR():** This function is used to find the occurrence of an alphabet.

**Syntax:** INSTR('geeks for geeks', 'e');

**Output:** 2 (the first occurrence of ‘e’)

**Syntax:** INSTR('geeks for geeks', 'e', 1, 2 );

**Output:** 3 (the second occurrence of ‘e’)

1. **LCASE():** This function is used to convert the given string into lower case.
2. **Syntax:** LCASE ("GeeksFor Geeks To Learn");

**Output:** geeksforgeeks to learn

1. **LEFT():** This function is used to SELECT a sub string from the left of given size or characters.

**Syntax:** SELECT LEFT('geeksforgeeks.org', 5);

**Output:** geeks

1. **LENGTH():** This function is used to find the length of a word.

**Syntax:** LENGTH('GeeksForGeeks');

**Output:** 13

1. **LOCATE():** This function is used to find the nth position of the given word in a string.

**Syntax:** SELECT LOCATE('for', 'geeksforgeeks', 1);

**Output:** 6

1. **LOWER():** This function is used to convert the upper case string into lower case.

**Syntax:** SELECT LOWER('GEEKSFORGEEKS.ORG');

**Output:** geeksforgeeks.org

1. **LPAD():** This function is used to make the given string of the given size by adding the given symbol.

**Syntax:** LPAD('geeks', 8, '0');

1. **Output:**

000geeks

1. **LTRIM():** This function is used to cut the given sub string from the original string.

**Syntax:** LTRIM('123123geeks', '123');

**Output:** geeks

1. **MID():** This function is to find a word from the given position and of the given size.

**Syntax:** Mid ("geeksforgeeks", 6, 2);

**Output:** for

1. **POSITION():** This function is used to find position of the first occurrence of the given alphabet.

**Syntax:** SELECT POSITION('e' IN 'geeksforgeeks');

**Output:** 2

1. **REPEAT():** This function is used to write the given string again and again till the number of times mentioned.

**Syntax:** SELECT REPEAT('geeks', 2);

**Output:** geeksgeeks

1. **REPLACE():** This function is used to cut the given string by removing the given sub string.

**Syntax:** REPLACE('123geeks123', '123');

**Output:** geeks

1. **REVERSE():** This function is used to reverse a string.

**Syntax:** SELECT REVERSE('geeksforgeeks.org');

**Output:** ‘gro.skeegrofskeeg’

1. **RIGHT():** This function is used to SELECT a sub string from the right end of the given size.

**Syntax:** SELECT RIGHT('geeksforgeeks.org', 4);

**Output:** ‘.org’

1. **RPAD():** This function is used to make the given string as long as the given size by adding the given symbol on the right.

**Syntax:** RPAD('geeks', 8, '0');

**Output:** ‘geeks000’

1. **RTRIM():** This function is used to cut the given sub string from the original string.
2. **Syntax:** RTRIM('geeksxyxzyyy', 'xyz');

**Output:** ‘geeks’

1. **SPACE():** This function is used to write the given number of spaces.

**Syntax:** SELECT SPACE(7);

**Output:** ‘ ‘

1. **STRCMP():** This function is used to compare 2 strings.
   * If string1 and string2 are the same, the STRCMP function will return 0.
   * If string1 is smaller than string2, the STRCMP function will return -1.
   * If string1 is larger than string2, the STRCMP function will return 1.

**Syntax:** SELECT STRCMP('google.com', 'geeksforgeeks.com');

**Output:** -1

1. **SUBSTR():** This function is used to find a sub string from the a string from the given position.

**Syntax:**SUBSTR('geeksforgeeks', 1, 5);

**Output:** ‘geeks’

1. **SUBSTRING():** This function is used to find an alphabet from the mentioned size and the given string.

**Syntax:** SELECT SUBSTRING('GeeksForGeeks.org', 9, 1);

**Output:** ‘G’

1. **SUBSTRING\_INDEX():** This function is used to find a sub string before the given symbol.

**Syntax:** SELECT SUBSTRING\_INDEX('www.geeksforgeeks.org', '.', 1);

**Output:** ‘www’

1. **TRIM():** This function is used to cut the given symbol from the string.

**Syntax:** TRIM(LEADING '0' FROM '000123');

**Output:** 123

1. **UCASE():** This function is used to make the string in upper case.

**Syntax:** UCASE ("GeeksForGeeks");

1. **Output:**

GEEKSFORGEEKS

**4) SQL | Advanced Functions**

Following are some of the advanced functions defined in SQL:

1. **BIN():** It converts a decimal number to a binary number.  
   **Syntax:**

SELECT BIN(18);

**Output:**  


1. **BINARY():** It converts a value to a binary string  
   **Syntax:**

SELECT BINARY "GeeksforGeeks";

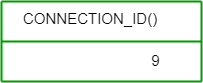
**Output:**  


1. **COALESCE():** It returns the first non-null expression in a list.  
   **Syntax:**

SELECT COALESCE(NULL,NULL,'GeeksforGeeks',NULL,'Geeks');

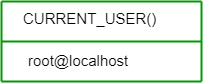
**Output:**  


1. **CONNECTION\_ID():** It returns the unique connection ID for the current connection.  
   **Syntax:**
2. SELECT CONNECTION\_ID();

**Output:**  


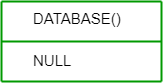
1. **CURRENT\_USER():** It returns the user name and host name for the MySQL account used by the server to authenticate the current client.  
   **Syntax:**

SELECT CURRENT\_USER();

**Output:**  


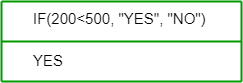
1. **DATABASE():** It returns the name of the default database.  
   **Syntax:**

SELECT DATABASE();

**Output:**  


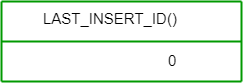
1. **IF():** It returns one value if a condition is TRUE, or another value if a condition is FALSE.  
   **Syntax:**

SELECT IF(200<500, "YES", "NO");

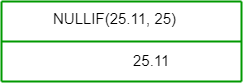
**Output:**  


1. **LAST\_INSERT\_ID():** It returns the first AUTO\_INCREMENT value that was set by the most recent INSERT or UPDATE statement.  
   **Syntax:**

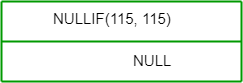
SELECT LAST\_INSERT\_ID();

**Output:**  


1. **NULLIF():** It returns the first expression if the two expressions are not equal. If the expressions are equal, NULLIF returns a null value of the type of the first expression.
   * **Syntax:**
   * SELECT NULLIF(25.11, 25);

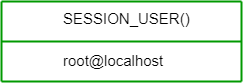
**Output:**  


* + **Syntax:**
  + SELECT NULLIF(115, 115);

**Output:**  


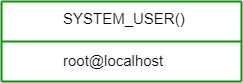
1. **SESSION\_USER():** It returns the user name and host name for the current MySQL user.  
   **Syntax:**

SELECT SESSION\_USER();

**Output:**  


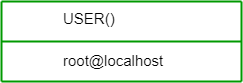
1. **SYSTEM\_USER():** It returns the user name and host name for the current MySQL user.  
   **Syntax:**

SELECT SYSTEM\_USER();

**Output:**  


1. **USER():** It returns the user name and host name for the current MySQL user.  
   **Syntax:**

SELECT USER();

**Output:**  


1. **VERSION():** It returns the version of the MySQL database.  
   **Syntax:**

SELECT VERSION();

**Output:**  
