SQL functions are built into Oracle and are available for use in various appropriate SQL statements. You can also create your own function using PL/SQL.

**Single-Row Functions**

Single-row functions return a single result row for every row of a queried table or view. These functions can appear in select lists, WHERE clauses, START WITH and CONNECT BY clauses, and HAVING clauses.

Oracle SQL Functions can be divided into following categories

* **Number Functions**
* **Character Functions**
* **Miscellaneous Single Row Functions**
* **Aggregate Functions**
* **Date and Time Functions**

Here are the explanation and example of these functions

**Number Functions (also known as Math Functions)**

Number functions accept numeric input and return numeric values. Most of these functions return values that are accurate to 38 decimal digits.

The number functions available in Oracle are:

ABS  ACOS ASIN ATAN ATAN2 BITAND CEIL COS COSH EXP FLOOR LN LOG

MOD POWER ROUND (number) SIGN SIN SINH SQRT TAN TANH TRUNC (number)

**ABS**

ABS returns the absolute value of n.

The following example returns the absolute value of -87:

SELECT ABS(-87) "Absolute" FROM DUAL;

  Absolute

  ----------

        87

**ACOS**

ACOS returns the arc cosine of n. Inputs are in the range of -1 to 1, and outputs are in the range of 0 to pi and are expressed in radians.

The following example returns the arc cosine of .3:

SELECT ACOS(.3)"Arc\_Cosine" FROM DUAL;

Arc\_Cosine

----------

1.26610367

Similar to ACOS, you have ASIN (Arc Sine), ATAN (Arc Tangent) functions.

**CIEL**

Returns the lowest integer above the given number.

Example:

The following function return the lowest integer above 3.456;

select ciel(3.456) “Ciel” from dual;

Ciel

---------

        4

**FLOOR**

Returns the highest integer below the given number.

Example:

The following function return the highest integer below 3.456;

select floor(3.456) “Floor” from dual;

Floor

------------

        3

**COS**

Returns the cosine of an angle (in radians).

Example:

The following example returns the COSINE angle of 60 radians.

select  cos(60) “Cosine” from dual;



**SIN**

Returns the Sine of an angle (in radians).

Example:

The following example returns the SINE angle of 60 radians.

select  SIN(60) “Sine” from dual;

**TAN**

Returns the Tangent of an angle (in radians).

Example:

The following example returns the tangent angle of 60 radians.

select  Tan(60) “Tangent” from dual;

Similar to SIN, COS, TAN  functions hyperbolic functions  SINH, COSH, TANH are also available in oracle.

**MOD**

Returns the remainder after dividing m with n.

Example

The following example returns the remainder after dividing 30 by 4.

Select mod(30,4) “MOD” from dual;

MOD

---------

        2

**POWER**

Returns the power of m, raised to n.

Example

The following example returns the 2 raised to the power of 3.

select  power(2,3) “Power” from dual;

POWER

---------

        8

**EXP**

Returns the e raised to the power of n.

Example

The following example returns the e raised to power of 2.

select exp(2) “e raised to 2” from dual;

E RAISED TO 2

-------------

**LN**

Returns natural logarithm of n.

Example

The following example returns the natural logarithm of 2.

select ln(2) from dual;

LN

------------

**LOG**

Returns the logarithm, base m, of n.

Example

The following example returns the log of 100.

select log(10,100) from dual;

LOG

---------

        2

**ROUND**

Returns a decimal number rounded of to a given decimal positions.

Example

The following example returns the no. 3.4573 rounded to 2 decimals.

select round(3.4573,2) “Round” from dual;

Round

------------

        3.46

**TRUNC**

Returns a decimal number Truncated to a given decimal positions.

Example

The following example returns the no. 3.4573 truncated to 2 decimals.

select round(3.4573,2) “Round” from dual;

Round

------------

        3.45

**SQRT**

Returns  the square root of a given number.

Example

The following example returns the square root of  16.

select  sqrt(16) from dual;

SQRT

---------

        4