**SQL FUNCTIONS**

\*\*for implementing every sql functions we will use dummy table named as dual.

**1) abs(number)**

This function Returns modulus of negative number.

**Syntax:**

SELECT abs(-126) “ABSOLUTE” FROM dual;

OUTPUT:

|  |
| --- |
| ABSOLUTE |
| 126 |

**2) power (m,n)**

This function returns the power of a given number where m is number and n is power.

**Syntax:**

SELECT power(3,3) “POWER” FROM dual;

OUTPUT:

|  |
| --- |
| POWER |
| 27 |

**3) Round(number)**

This function Returns round off value for a decimal number. If decimal point is greater than or equal to 0.5 then the next higher value is returned else the same number is returned.

**Syntax:**

SELECT round(178.499) “ROUND” FROM dual;

OUTPUT:

|  |
| --- |
| ROUND |
| 178 |

**4) Truncate(n,m)**

This function Returns the number after mentioning the number of decimal parts to be shown where n is the number and m is the number of decimal parts to be included.

**Syntax:**

SELECT trunc(178.599,2) “TRUNCATE” FROM dual;

OUTPUT:

|  |
| --- |
| TRUNCATE |
| 178.59 |

**5) sqrt(number)**

This function Returns square root of any given number.

**Syntax:**

SELECT sqrt(25) “SQUARE ROOT” FROM dual;

OUTPUT:

|  |
| --- |
| SQUARE ROOT |
| 5 |

**6) Greatest (number1, number2, number3….)**

This function Returns the greatest number amongst the group of numbers.

**Syntax:**

SELECT greatest (14,22,7) “GREATEST” FROM dual;

OUTPUT:

|  |
| --- |
| GREATEST |
| 22 |

**7) Least(number1, number2, number3…..)**

This function Returns the least number amongst all the numbers.

**Syntax:**

SELECT least(14,22,7) “LEAST” FROM dual;

OUTPUT:

|  |
| --- |
| LEAST |
| 7 |

**7) Modulus (n,m)**

This function Returns remainder of the two provided numbers where n is the number divided by m.

**Syntax:**

SELECT mod(15,7) “MODULUS” FROM dual;

OUTPUT:

|  |
| --- |
| MODULUS |
| 1 |

**8) Floor(number)**

This function Returns the same base value of any provided number.

**Syntax:**

SELECT floor(126.99) “FLOOR” FROM dual;

OUTPUT:

|  |
| --- |
| FLOOR |
| 126 |

**9) Ceil(number)**

This function Returns the next higher value for any given number.

**Syntax:**

SELECT ceil(126.12) ”CEIL” FROM dual;

OUTPUT:

|  |
| --- |
| CEIL |
| 127 |

10) This function Returns the exponent value for any given number

select exp(2) from dual

|  |
| --- |
| **EXP(2)** |
| 7.3890560989306502272304274605750078132 |

**STRING FUNCTIONS**

**1) Lower (character/word/sentence)**

This function Returns the provided character in lower case.

**Syntax:**

SELECT lower('Lovely Professional University') "LOWER" FROM dual;

OUTPUT:

|  |
| --- |
| LOWER |
| lovely professional university |

**2) Upper(character/word/sentence)**

This function Returns the character or word with complete upper case.

**Syntax:**

SELECT upper('Lovely Professional University') "UPPER" FROM dual;

OUTPUT:

|  |
| --- |
| UPPER |
| LOVELY PROFESSIONAL UNIVERSITY |

**3) Initcap(character/word/sentence)**

This function Returns the word with its first character being capitalized.

**Syntax:**

SELECT initcap('LOVELY PROFESSIONAL UNIVERSITY') "Capitallize Initial" FROM dual;

OUTPUT:

|  |
| --- |
| **Capitallize Initial** |
| Lovely Professional University |

**4) Length(character/word/sentence)**

This function Returns the length of the word or sentence in integer or number.

**Syntax:**

SELECT length('LOVELY PROFESSIONAL UNIVERSITY') "Length of String" FROM dual;

OUTPUT:

|  |
| --- |
| **Length of String** |
| 30 |

**5) Ltrim(word/sentence, character(to be removed))**

This function Returns the string after omitting the character mentioned in the function from the left.

**Syntax:**

SELECT ltrim('LOVELY PROFESSIONAL UNIVERSITY','LOV') "After LTrim String" FROM dual;

OUTPUT:

|  |
| --- |
| **After LTrim String** |
| ELY PROFESSIONAL UNIVERSITY |

**6) Rtrim(word/sentence, character(to be removed))**

This function Returns the string after omitting the character the mentioned in the function from the right.

**Syntax:**

SELECT rtrim('LOVELY PROFESSIONAL UNIVERSITY','ITY') "After RTrim String" FROM dual;

OUTPUT:

|  |
| --- |
| **After RTrim String** |
| LOVELY PROFESSIONAL UNIVERS |

**7) Trim(TRAILING ‘character’ from word/sentence)**

**Or**

**Trim(LEADING ‘character’ from word/sentence)**

**Or**

**Trim(BOTH ‘character’ from word/sentence)**

This function Returns the word or sentence after removing the specified character from left or front in case of LEADING, from right or last in case of TRAILING and from both the ends in case of BOTH.

**Syntax:**

SELECT trim(TRAILING 'M' from 'MADAM') "TRIM" FROM dual;

SELECT trim(leading 'M' from 'MADAM') "TRIM" FROM dual;

SELECT trim(both 'M' from 'MADAM') "TRIM" FROM dual;

OUTPUT:

|  |
| --- |
| TRIM |
| MADA |

|  |
| --- |
| TRIM |
| ADAM |

|  |
| --- |
| TRIM |
| ADA |
|  |

SELECT ascii('A'),ascii('1'),ascii('a') FROM dual;

|  |  |  |
| --- | --- | --- |
| ASCII('A') | ASCII('1') | ASCII('A') |
| 65 | 49 | 97 |

SELECT REPLACE('JACK and JUE','J','BL') "Changes"

FROM DUAL;

**Changes**

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

BLACK and BLUE

SELECT REPLACE('abcdeabcccabdddeeabcc', 'abc') "Replace string" FROM DUAL;

|  |
| --- |
| **Replace string** |
| **--------------------** |
| deccabdddeec |

SELECT RPAD('LOVELY',12,'#') "RPAD example"

FROM DUAL;

**RPAD example**

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

LOVELY######

SELECT LPAD('LOVELY',12,'#') "RPAD example"

FROM DUAL;

**LPAD example**

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

######LOVELY

SELECT Substr('LOVELY',3,2) "substr example"

FROM DUAL;

**substr example**

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

VE

SELECT INSTR('CORPORATE FLOOR','OR', 3, 2)

"Instring" FROM DUAL;

**Instring**

----------

14

select \* from emp

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

|  |  |  |
| --- | --- | --- |
| EMPID | SALARY | DEPT |
| 1 | 234.34 | HR |
| 2 | 200.34 | Admin |
| 4 | 200.34 | HR |
| 5 | 200.34 | HR |
| 6 | 200.34 | HR |
| 11 | 200 | Admin |

SELECT CONCAT(salary,' ----is very costly')

"Concatenated" FROM emp;

|  |
| --- |
| Concatenated  ----------------------------- |
| 234.34 ----is very costly |
| 200.34 ----is very costly |
| 200.34 ----is very costly |
| 200.34 ----is very costly |
| 200.34 ----is very costly |
| 200 ----is very costly |

SELECT CONCAT(empid,dept)

"employee and its dept" FROM emp;

|  |
| --- |
| employee and its dept |
| ---------------------------- |
| 1HR |
| 2Admin |
| 4HR |
| 5HR |
| 6HR |
| 11Admin |

In the next example, Oracle counts backward from the last character to the third character from the end, which is the first "O" in "FLOOR". Oracle then searches backward for the second occurrence of OR, and finds that this second occurrence begins with the second character in the search string :

SELECT INSTR('CORPORATE FLOOR','OR', -3, 2)

"Reversed Instring"

FROM DUAL;

Reversed Instring

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

2

SELECT reverse(dept)

"reverse" FROM emp;

|  |
| --- |
| reverse |
| ----------- |
| RH |
| nimdA |
| RH |
| RH |
| RH |
| nimdA |

This example assumes a double-byte database character set.

SELECT INSTRB('CORPORATE FLOOR','OR',5,2) "Instring in bytes"

FROM DUAL;

Instring in bytes

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

27

**AVERAGE FUNCTIONS**

**1) Average(column name)**

This function Returns the average of all the numeric values in a particular column.

**Syntax:**

SELECT avg(salary) “AVERAGE SALARY” FROM employ;

OUTPUT:

|  |
| --- |
| AVERAGE SALARY |
| 47500 |

**2) Minimum (column name)**

This function Returns the minimum value amongst all the values in a column.

**Syntax:**

SELECT min(SALARY) “MINIMUM SALARY” FROM employ;

OUTPUT:

|  |
| --- |
| MINIMUM SALARY |
| 23000 |

**3) Maximum(column name)**

This function Returns the maximum value multiple values in a column.

**Syntax:**

SELECT max(SALARY) “MAXIMUM SALARY” FROM employ;

OUTPUT:

|  |
| --- |
| MAXIMUM SALRY |
| 90000 |

**4) Sum(column name)**

This function Returns the sum of the values in a column.

**Syntax:**

SELECT sum(SALARY) “SUM SALARY” FROM employ;

OUTPUT:

|  |
| --- |
| SUM SALARY |
| 380000 |

**5) Count(column name) & Count(\*)**

The difference between count(column name) and count(\*) is that the count(column name) will count the values present in that column whereas the count(\*) will count all the rows in a table.

**Syntax:**

SELECT count(SALARY) “COUNT SALARY” FROM employ;

OUTPUT:

|  |
| --- |
| COUNT SALARY |
| 8 |

**DATE Functions**

1)

select sysdate from dual

|  |
| --- |
| **SYSDATE** |
| 02-OCT-18 |

2)

select sysdate "today's date",add\_months(sysdate,4) "after 4 month" from dual

|  |  |
| --- | --- |
| **today's date** | **after 4 month** |
| 02-OCT-18 | 02-FEB-19 |

3)

select months\_between('02-oct-18','02-oct-19') "months between",months\_between('02-oct-19','02-oct-18') "months between" from dual

4)

select sysdate,next\_day(sysdate,'Tuesday') "next tuesday" from dual

|  |  |
| --- | --- |
| **SYSDATE** | **next tuesday** |
| 02-OCT-18 | 09-OCT-18 |
| 5)  select sysdate,last\_day(sysdate) "Last day of given month" from dual   |  |  | | --- | --- | | **SYSDATE** | **Last day of given month** | | 02-OCT-18 | 31-OCT-18 | |  |