*** SQL Functions –**

A function is a predefined formula which takes one or more arguments as input then processes the arguments and returns an output.

Oracle SQL supplies a rich library of in-built functions which can be used for various tasks. There are two types of SQL functions,

1) Single Row functions (Scalar Functions)
2) Multiple Row functions (Aggregate or Group functions).

1) Single Row Functions –

Single row or Scalar functions return a value for every row that is processed in a query. Single row functions can be,

a) Character Functions
b) Numeric Functions
c) Date Functions
d) Conversion Functions
e) General Functions

a) Character or String or Text Functions –

Character or string or text functions are used to manipulate text strings. They accept strings or characters or number as input and can return string or character or number value as output.

1) ASCII Function –

The ASCII function returns the NUMBER code that represents the specified character. It accepts only single character at a time.

Syntax –

ASCII (single_character)

Example –

SELECT ASCII ('A') AS "ASCII" FROM DUAL;

ASCII
65

2) CHR Function –

The CHR function is the opposite of the ASCII function. It returns the character based on the NUMBER code.

Syntax –

CHR (number_code)

Example –

SELECT CHR (65) AS "CHAR" FROM DUAL;

C
A

3) LENGTH Function –

The LENGTH function returns the length of the specified string.

Syntax –

LENGTH (string)

Example –

SELECT LENGTH ('INDIA') AS "LENGTH" FROM DUAL;

LENGTH
5

4) LOWER Function –

The LOWER function converts all letters in the specified string to lowercase.

Syntax –

LOWER (string)

Example –

SELECT LOWER ('INDIA') AS "SMALL" FROM DUAL;

SMALL
india

5) UPPER Function –

UPPER function converts all letters in the specified string to uppercase.

Syntax –

UPPER (string)

Example –

SELECT UPPER ('india') AS "UPPER" FROM DUAL;

UPPER
INDIA

6) INITCAP Function –

The INITCAP function sets the first character in each word to uppercase and the rest to lowercase.

Syntax –

INITCAP (string)

Example –

SELECT INITCAP ('india is my country') AS "INITCAP" FROM DUAL;

INITCAP
India Is My Country

7) CONCAT Function –

The CONCAT function allows you to concatenate two strings together.

Syntax –

CONCAT (string1, string2)

Example –

SELECT CONCAT ('India is ','my country') AS "CONCAT" FROM DUAL;

CONCAT
India is my country

b) Numeric Functions –

A numeric function executes a mathematical operation usually based on input values that are provided as arguments. and return a numeric value as the result of the operation.

1) ABS Function –

The ABS function returns the absolute value of a number.

Syntax –

ABS (number)

Example –

SELECT ABS (-23) AS "ABS" FROM DUAL;

ABS
23

2) SQRT Function –

The SQRT function returns the square root of n.

Syntax –

SQRT (n)

Example –

SELECT SQRT (100) AS "SQRT" FROM DUAL;

SQRT
10

3) SIGN Function –

The SIGN function returns a value indicating the sign of a number.

If number < 0, then sign returns -1,
If number = 0, then sign returns 0, and
If number > 0, then sign returns 1.

Syntax –

SIGN (number)

Example –

a) SELECT SIGN (-23) AS "SIGN" FROM DUAL;

SIGN
-1

b) SELECT SIGN (0) AS "SIGN" FROM DUAL;

SIGN
0

c) SELECT SIGN (23) AS "SIGN" FROM DUAL;

SIGN
1

4) GREATEST Function –

The GREATEST function returns the greatest value in a list of expressions.

Syntax –

GREATEST (expr1, expr2, ... expr_n)

Example –

SELECT GREATEST (2, 5, 12) AS "LARGER" FROM DUAL;

LARGER
12

5) LEAST Function –

The LEAST function returns the smallest value in a list of expressions.

Syntax –

LEAST (expr1, expr2, ... expr_n)

Example –

SELECT LEAST (2, 5, 12) AS "SMALLER" FROM DUAL;

SMALLER
2

6) MOD Function –

The MOD function returns the remainder of m divided by n.

Syntax –

MOD (m, n)

Example –

SELECT MOD (13, 5) AS "REMAINDER" FROM DUAL;

REMAINDER
3

7) POWER Function –

The POWER function returns m raised to the nth power (m^n).

Syntax –

POWER (m, n)

Example –

SELECT POWER (3, 2) AS "POWER" FROM DUAL;

POWER
9

c) Date Functions –

These are functions that take values that are of datatype DATE as input and return values of datatype DATE, except for the MONTHS_BETWEEN function, which returns a number.

1) ADD_MONTHS Function –

The ADD_MONTHS function returns a date with a specified number of months added.

Syntax –

ADD_MONTHS (date_value, number_months)

Example –

a) SELECT ADD_MONTHS ('04-AUG-17', 3) AS "NEW DATE" FROM DUAL;

NEW DATE
04-NOV-17

b) SELECT ADD_MONTHS ('04-AUG-17', -3) AS "NEW DATE" FROM DUAL;

NEW DATE
04-MAY-17

2) EXTRACT Function –

The EXTRACT function extracts a value from a date.

Syntax –

EXTRACT ({YEAR | MONTH | DAY} FROM DATE date_value)

Example –

SELECT EXTRACT (YEAR FROM DATE '1985-09-11') AS "YEAR" FROM DUAL;

YEAR
1985

3) LAST_DAY Function –

The LAST_DAY function returns the last day of the month based on a date value.

Syntax –

LAST_DAY (date_value)

Example –

SELECT LAST_DAY ('03-AUG-85') AS "LAST DAY" FROM DUAL;

LAST DAY
31-AUG-85

4) MONTHS_BETWEEN Function –

The MONTHS_BETWEEN function returns the number of months between date1 and date2.

Syntax –

MONTHS_BETWEEN (date1, date2)

Example –

SELECT MONTHS_BETWEEN ('03-AUG-17', '03-MAY-17')
AS "DIFFERENCE" FROM DUAL;

DIFFERENCE
3

5) SYSDATE Function –

The SYSDATE function returns the current system date of your local database.

Syntax –

SYSDATE

Example –

SELECT SYSDATE AS "NOW" FROM DUAL;

NOW
07-AUG-17

d) Conversion Functions –

These are functions that help us to convert a value in one form to another form. For Example: a null value into an actual value, or a value from one datatype to another datatype.

1) BIN_TO_NUM Function –

The BIN_TO_NUM function converts a bit vector to a number.

Syntax –

BIN_TO_NUM (expr1, expr2, ...)

Example –

SELECT BIN_TO_NUM (1, 1, 0, 1) AS "DECIMAL" FROM DUAL;

DECIMAL
13

2) CAST Function –

The CAST function converts one datatype to another.

Syntax –

CAST (expr AS type_name)

Example –

SELECT CAST (100 AS VARCHAR2 (10)) AS "CONVERSION" FROM DUAL;

CONVERSION
100

This would convert the number (i.e.100) into a varchar2 (10) value.

3) TO_CHAR Function –

The TO_CHAR function converts a number or date to a string.

Syntax –

TO_CHAR (value)

Example –

a) SELECT TO_CHAR (123, '000') AS "CONVERT" FROM DUAL;

CONVERT
123

b) SELECT TO_CHAR (SYSDATE, 'dd/mm/yyyy') AS "CONVERT" FROM DUAL;

CONVERT
06/08/2017

10

4) TO_DATE Function –

The TO_DATE function converts a string to a date.

Syntax –

TO_DATE (string, format_mask)

Example –

SELECT TO_DATE ('2017/08/07', 'yyyy/mm/dd') AS "CONVERT" FROM DUAL;

CONVERT
07-AUG-17

5) TO_NUMBER Function –

The TO_NUMBER function converts a string to a number.

Syntax –

TO_NUMBER (string)

Example –

SELECT TO_NUMBER ('123') AS "CONVERT" FROM DUAL;

CONVERT
123

e) General Functions –

These functions are mainly used to handle null values.

1) NVL Function –

The NVL function takes two arguments as its input. If the first argument is NULL, then it returns the second argument otherwise it returns the first argument.

Syntax –

NVL (expr, value)

Example –

a) SELECT NVL (10, 2) AS "RESULT" FROM DUAL;

RESULT
10

b) SELECT NVL (NULL, 'Oracle') AS "RESULT" FROM DUAL;

RESULT
Oracle

2) NVL2 Function –

The NVL2 function takes three arguments as its input. If the expr1 is NOT NULL, NVL2 function returns expr2. If expr1 is NULL, then NVL2 returns expr3.

Syntax –

NVL2 (expr1, expr2, expr3)

Example –

a) SELECT NVL2 (10, 2, 5) AS "RESULT" FROM DUAL;

RESULT
2

b) SELECT NVL2 (NULL, 'SQL', 'Oracle') AS "RESULT" FROM DUAL;

RESULT
Oracle

3) NULLIF Function –

The NULLIF function compares the two expressions and returns NULL if they are equal otherwise it returns the first expression.

Syntax –

NULLIF (expr1, expr2)

Example –

a) SELECT NULLIF ('Oracle', 'MySQL') AS "RESULT" FROM DUAL;

RESULT
Oracle

b) SELECT NULLIF (10, 10) AS "RESULT" FROM DUAL;

RESULT

This would return the NULL value.

4) COALESCE Function –

The COALESCE function takes N number of arguments as its input and returns the first NON-NULL argument as output.

Syntax –

COALESCE (expr1, expr2, expr3, ...)

Example –

SELECT COALESCE (NULL, 10, 2) AS "RESULT" FROM DUAL;

RESULT
10

*** Multiple Row functions –**

Multiple row functions work upon group of rows and return one result for the complete set of rows. They are also known as group functions or aggregate functions.

Consider the following STUDENT table.

RN	NAME	CITY	FEES
1	AMOL	LATUR	10000
2	ATUL		15000
3	RAHUL	LATUR	20000
4	AMARJEET		25000

1) COUNT Function –

The COUNT function returns the count of an expression.

Syntax –

SELECT COUNT (expression) FROM table [WHERE condition(s)];

The COUNT function will only include the records in the count where the value of expression in COUNT (expression) is NOT NULL.

Example –

a) SELECT COUNT (NAME) AS "COUNT" FROM STUDENT;

COUNT
4

b) SELECT COUNT (CITY) AS "COUNT" FROM STUDENT;

COUNT
2

This COUNT example will only return 2, since only two CITY value in the query's result set is NOT NULL.

2) SUM Function –

The SUM function returns the summed value of a given numeric expression.

Syntax –

SELECT SUM (expression) FROM table [WHERE condition(s)];

Example – a) Calculate total fees of all students.

```
SELECT SUM (FEES) AS "TOTAL" FROM STUDENT;
```

TOTAL
70000

b) Calculate total fees of all students, whose FEES is above 15000.

```
SELECT SUM (FEES) AS "TOTAL" FROM STUDENT
WHERE FEES > 15000;
```

TOTAL
45000

3) AVG Function –

The AVG function returns the average of a given numeric expression.

Syntax –

SELECT AVG (expression) FROM table [WHERE condition(s)];

Example – a) Calculate the average fee.

```
SELECT AVG (FEES) AS "AVERAGE" FROM STUDENT;
```

AVERAGE
17500

b) Calculate the average fee, who's FEES, is above 15000.

SELECT AVG (FEES) AS "AVERAGE" FROM STUDENT
WHERE FEES > 15000;

AVERAGE
22500

4) MIN Function –

The MIN function returns the minimum value of an expression.

Syntax –

SELECT MIN (expression) FROM table [WHERE condition(s)];

Example – a) Find out the minimum value from FEES column.

SELECT MIN (FEES) AS "MINIMUM" FROM STUDEN0T:

MINIMUM
10000

b) Find out the minimum value from NAME column.

SELECT MIN (NAME) AS "MINIMUM" FROM STUDENT;

MINIMUM
AMARJEET

5) MAX Function –

The MAX function returns the maximum value of an expression.

Syntax –

SELECT MAX (expression) FROM table [WHERE condition(s)];

Example – a) Find out the maximum value from FEES column.

SELECT MAX (FEES) AS "MAXIMUM" FROM STUDENT;

MAXIMUM
25000

16

b) Find out the maximum value from NAME column.

```
SELECT MAX (NAME) AS "MAXIMUM" FROM STUDENT;
```

MAXIMUM
RAHUL