PL/SQL Functions

SQL provides various inbuilt functions to simplify significantly large tasks. SQL functions can be broadly classified as:

- ✓ Group Functions – This helps in grouping of rows returned by the query. This is used to compute aggregate values such as Sum, Average etc. and hence is also called as Aggregate Function.
- ✓ Single Row Functions – These are further classified as Numeric functions, Date Functions, Conversion functions and Character/Text functions.

1. GROUP FUNCTIONS:

General syntax for using aggregate functions is:

SELECT X, AVG(Y) FROM TABLE GROUP BY X; where X and Y are two columns of table

- ✓ **AVG(x) –** computes the average value of a given set of values
- ✓ **COUNT(x) -** counts rows in specified table or given column
- ✓ **MIN(x) –** provides minimum value from a set of values
- ✓ **MAX(x) –** provides maximum value from a set of values
- ✓ **SUM(x) –** computes the sum of values

2. SINGLE ROW FUNCTIONS:

Below are mentioned all the types of single row functions:

❖ Numeric Functions:

- ✓ **ABS(y)** – Provides absolute value of y
  SELECT ABS(-89.78) FROM DUAL;
  Output of query –> 89.78

- ✓ **CEIL(y)** – Provides value greater than or equal to y

  SELECT CEIL(2.56) FROM DUAL;

  Output of query –> 3

  SELECT CEIL(-1.4) FROM DUAL;
  Output of query –> -1

- ✓ **FLOOR(y) -** Provides value less than or equal to y

SELECT CEIL(2.56) FROM DUAL;

Output of query –> 2

SELECT CEIL(-1.4) FROM DUAL;
Output of query –> -2

- ✓ **TRUNC(x,y) –** Truncates the value of x up to y decimal places

  SELECT TRUNC(25.67,1) FROM DUAL;

  Output of query –> 25.6

- ✓ **ROUND(x,y) -** Provides rounded off value of x up to y decimal places

  SELECT ROUND(25.67,1) FROM DUAL;

  Output of query –> 25.7

❖ Conversion Functions:

- ✓ **NVL(x,y) –** Replace x with y in case x is null where x and y are of same data type

  SELECT NVL(null,3) FROM DUAL;

  Output of query –> 3

- ✓ **DECODE (x, y, z, default value) –** Checks if x=y, then returns z else returns default value

  SELECT DECODE (1,1,2,3) FROM DUAL;

  Output of query –> 2

- ✓ **TO_CHAR(x, 'y') –** converts x into the format specified in y

  SELECT TO_CHAR(5000,'$9999') FROM DUAL;

  Output of query –> $5000

- ✓ **TO_DATE(x, date_format) –** converts value of x to a valid date format specified by date_ format

  SELECT TO_DATE('2012-07-18', 'YYYY-MM-DD') FROM DUAL;

  Output of query –> '2012-07-18

❖ Character/Text Function:

- ✓ **LOWER(StringValue) –** converts complete string to lower case

  SELECT LOWER('Hello World') FROM DUAL;

  Output of query –> hello world

- ✓ **UPPER(StringValue) –** converts complete string to upper case

  SELECT UPPER('Hello World') FROM DUAL;

  Output of query –> HELLO WORLD

- ✓ **INITCAP(StringValue) –** makes the first letter appear in capital case

  SELECT INITCAP('Hello World') FROM DUAL;

  Output of query –> Hello World

- ✓ **LTRIM(StringValue, TrimValue) –** TrimValue text is removed from the left of String
  SELECT LTRIM('Hello World', 'Hello') FROM DUAL;
  Output of query –> World

- ✓ **RTRIM(StringValue, TrimValue) –** TrimValue text is removed from the right of String

  SELECT RTRIM('Hello World', World) FROM DUAL;
  Output of query –> Hello

- ✓ **SUBSTR(StringValue, m, n) –** Returns n number of characters starting from position m

  SELECT SUBSTR('Hello World',7,5) FROM DUAL;

  Output of query –> World

- ✓ **LENGTH(StringValue) –** Provides the total length of StringValue

  SELECT LENGTH('Hello World') FROM DUAL;

  Output of query –> 11

- ✓ **LPAD(StringValue,n, PadValue) –** Provides StringValue left padded with pad value and length n

  SELECT LPAD('Hello',6,'$') FROM DUAL;

  Output of query –> $Hello

- ✓ **RPAD(StringValue,n,PadValue) –** Provides StringValue right padded with pad value and length n

  SELECT RPAD('Hello',6,'$') FROM DUAL;

  Output of query –> Hello$

- ✓ **REPLACE(StringValue, x, y) –** replaces every occurrence of x with y

  SELECT REPLACE('Hello','H','C') FROM DUAL;

  Output of query –> Cello

- ✓ **TRIM(TrimText from StringValue) –** Removes all occurrence of TrimText from left and right of StringValue. Where TrimText can also be a single character

  SELECT TRIM('o' from 'Hello World') FROM DUAL;

  Output of query –> Hell Wrld

❖ Date Functions:

- ✓ **ADD_MONTHS(date , n) –** Adds n months to the date value

  SELECT ADD_MONTHS('26-Sep-86',3) FROM DUAL;

  Output of query –> 26-Dec-86

- ✓ **MONTHS_BETWEEN(y1 , y2)** – returns the number of months between two dates

  SELECT MONTHS_BETWEEN('26-Sep-86','26-Dec-86') FROM DUAL;

  Output of query –> 3

- ✓ **NEXT_DAY(x, WeekDay)** – returns the next date of the WeekDay on or after the occurrence of date x

  SELECT NEXT_DAY('02-Oct-17','Wednesday') FROM DUAL;

  Output of query –> 04-Oct-17

- ✓ **LAST_DAY(x)** – this provides the date of the last day of month containing date x

  SELECT LAST_DAY('02-Oct-17') FROM DUAL;

  Output of query –> 31-Oct-17

- ✓ **NEW_TIME (x, zone1, zone2) –** returns the time in zone2 if x represents time in zone1

  SELECT NEW_TIME('01-Jun-08','IST','EST') FROM DUAL;

  Output of query –> 31-May-08

- ✓ **SYSDATE –** provides the current date and time

CONCLUSION:

Above mentioned are important in-built functions in SQL. These functions can be easily incorporated to compute several different logical operations based on query requirement.
You cand read more about PL/SQL language on the site https://www.plsql.co .