USING SINGLE-ROW FUNCTIONS

Using Single row functions to customize output

Oracle SQL supplies a rich library of in-built functions which can be employed for various tasks. The essential capabilities of a functions can be the case conversion of strings, in-string or substring operations, mathematical computations on numeric data, and date operations on date type values. SQL Functions optionally take arguments from the user and mandatorily return a value.

On a broader category, there are two types of functions :-

Single Row functions - Single row functions are the one who work on single row and return one output per row. For example, length and case conversion functions are single row functions.

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.

Single row functions

Single row functions can be character functions, numeric functions, date functions, and conversion functions. Note that these functions are used to manipulate data items. These functions require one or more input arguments and operate on each row, thereby returning one output value for each row. Argument can be a column, literal or an expression. Single row functions can be used in SELECT statement, WHERE and ORDER BY clause. Single row functions can be -

- **General functions** - Usually contains NULL handling functions. The functions under the category are NVL, NVL2, NULLIF, COALESCE, CASE, DECODE.
- **Case Conversion functions** - Accepts character input and returns a character value. Functions under the category are UPPER, LOWER and INITCAP.
    - UPPER function converts a string to upper case.
    - LOWER function converts a string to lower case.
    - INITCAP function converts only the initial alphabets of a string to upper case.
-
- **Character functions** - Accepts character input and returns number or character value. Functions under the category are CONCAT, LENGTH, SUBSTR, INSTR, LPAD, RPAD, TRIM and REPLACE.
    - CONCAT function concatenates two string values.
    - LENGTH function returns the length of the input string.
    - SUBSTR function returns a portion of a string from a given start point to an end point.
    - INSTR function returns numeric position of a character or a string in a given string.
    - LPAD and RPAD functions pad the given string upto a specific length with a given character.
    - TRIM function trims the string input from the start or end.
    - REPLACE function replaces characters from the input string with a given character.
- **Date functions** - Date arithmetic operations return date or numeric values. Functions under the category are MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY, LAST_DAY, ROUND and TRUNC.
    - MONTHS_BETWEEN function returns the count of months between the two dates.

  - ADD_MONTHS function add 'n' number of months to an input date.
  - NEXT_DAY function returns the next day of the date specified.
  - LAST_DAY function returns last day of the month of the input date.
  - ROUND and TRUNC functions are used to round and truncates the date value.
- **Number functions** - Accepts numeric input and returns numeric values. Functions under the category are ROUND, TRUNC, and MOD.
  - ROUND and TRUNC functions are used to round and truncate the number value.
  - MOD is used to return the remainder of the division operation between two numbers.

Illustrations

General functions

The SELECT query below demonstrates the use of NVL function.

```
SELECT first_name, last_name, salary, NVL (commission_pct,0)
FROM employees
WHERE rownum < 5;

FIRST_NAME           LAST_NAME                     SALARY NVL(COMMISSION_PCT,0)
-------------------- ------------------------- ---------- ---------------------
Steven               King                           24000                     0
Neena                Kochhar                        17000                     0
Lex                  De Haan                        17000                     0
Alexander            Hunold                          9000                     0
```

Case Conversion functions

The SELECT query below demonstrates the use of case conversion functions.

```
SELECT UPPER (first_name), INITCAP (last_name), LOWER (job_id)
FROM employees
WHERE rownum < 5;

UPPER(FIRST_NAME)    INITCAP(LAST_NAME)        LOWER(JOB_
-------------------- ------------------------- ----------
STEVEN               King                      ad_pres
NEENA                Kochhar                   ad_vp
LEX                  De Haan                   ad_vp
ALEXANDER            Hunold                    it_prog
```

Character functions

The SELECT query below demonstrates the use of CONCAT function to concatenate two string values.

```
SELECT CONCAT (first_name, last_name)
FROM employees
WHERE rownum < 5;

CONCAT(FIRST_NAME,LAST_NAME)
---------------------------------
EllenAbel
SundarAnde
MozheAtkinson
DavidAustin
```

The SELECT query below demonstrates the use of SUBSTR and INSTR functions. SUBSTR function returns the portion of input string from 1st position to 5th position. INSTR function returns the numeric position of character 'a' in the first name.

```
SELECT SUBSTR (first_name,1,5), INSTR (first_name,'a')
FROM employees
WHERE rownum < 5;

SUBST INSTR(FIRST_NAME,'A')
----- ----------------------
Ellen                      0
Sunda                      5
Mozhe                      0
David                      2
```

The SELECT query below demonstrates the usage of LPAD and RPAD to pretty print the employee and job information.

```
SELECT RPAD(first_name,10,'_')||LPAD (job_id,15,'_')
FROM employees
WHERE rownum < 5;

RPAD(FIRST_NAME,10,'_')||
-------------------------
Steven____________AD_PRES
Neena_______________AD_VP
Lex_________________AD_VP
Alexander________IT_PROG
```

Number functions

The SELECT query below demonstrates the use of ROUND and TRUNC functions.

```
SELECT ROUND (1372.472,1)
FROM dual;

ROUND(1372.472,1)
-----------------
           1372.5

SELECT TRUNC (72183,-2)
FROM dual;

TRUNC(72183,-2)
---------------
          72100
```

Date arithmetic operations

The SELECT query below shows a date arithmetic function where difference of employee hire date and sysdate is done.

```
SELECT employee_id, (sysdate - hire_date) Employment_days
FROM employees
WHERE rownum < 5;

EMPLOYEE_ID EMPLOYMENT_DAYS
----------- ---------------
        100      3698.61877
        101      2871.61877
        102      4583.61877
        103      2767.61877
```

Date functions

The SELECT query below demonstrates the use of MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY and LAST_DAY functions.

```
SELECT employee_id, MONTHS_BETWEEN (sysdate, hire_date) Employment_months
FROM employees
WHERE rownum < 5;

EMPLOYEE_ID EMPLOYMENT_MONTHS
----------- -----------------
        100        121.504216
        101        94.3751837
        102        150.633248
        103        90.9558289

SELECT ADD_MONTHS (sysdate, 5), NEXT_DAY (sysdate), LAST_DAY (sysdate)
FROM dual;

ADD_MONTH NEXT_DAY( LAST_DAY(
--------- --------- ---------
01-JAN-14 05-AUG-13 31-AUG-13
```