SQL Day 10: SQL Functions

SQL functions
There are two types of functions
1) Single row function
2) Multi-row function

Q. What is Single row functions?
Ans: The function which operate on single row is called single row function

Q. What is Multi-row function?

Ans: The function which operate on more than one row is called multi-row functions

Types of Single row Function
1) Character functions
2) Number Functions
3) Date Functions
4) Conversion Functions
5) General Functions

A) Character Functions:
=>Character function accept character input and return both character and number values.
=>Character functions are two types
1) Case Manipulation
2) Character Manipulation

Character Function- Case Manipulation

1) lower(source_String)
2) Upper(upper_string)
3) InitCap(source_String

1) lower(source_String)

i) This function convert upper case characters into lower case.
ii) This function take one input argument of String type
iii) Return type of this function is character.

Example: select lower('SQL Course') from dual;

Out put is
sql course

Here dual is dummy table that you can use to view result from function and calculation.

2) Upper(source_string)

i) This function convert lowe case characters into upper case.
ii) This function take one input argument of String type.
iii) Return type of this function is String.

Example:

```
SQL> select upper('sql course') from dual;

UPPER('SQL
----------
SQL COURSE
```

3) InitCap(source_String)

i) This function convert only first character into Upper case

ii) This function take one input argument of String type

iii) Return type of this function is String

Example:

```
SQL> select initcap('SQL COURSE') from dual;

INITCAP('S
----------
Sql Course
```

Examples on Character function-Case manipulation

Consider the Employee table having column name like emp_no, emp_name, salary and emp_city

1) Display all employee names in upper case

```
select upper(emp_name) from Employee;
```

2) Display all employee names in lower case

```
Select lower(emp_name) from Employee;
```

3) Display employee name first letter in capital and emp_city in Upper case

```
Select initcap(emp_name), upper(emp_city) from Employee;
```

2) Character function - Character Manipulation

1) contact:
   i) This function joins values together
   ii) This function take two input arguments of String type.
   iii) Return type of this function is character

Example

```
SQL> select concat('SQL', 'Course') from dual;

CONCAT('S
---------
SQLCourse

SQL> select concat(111,222) from dual;

CONCAT
------
111222
```

Note: If we are passing input arguments as number or float then internally it get converted into String type.

2) Length
 i) This function is used to calculate the length of String.
 ii) Length of String means number of character present in the given string
 iii) While calculating the length space is also calculated single space is considered as one character.
iv) This function take one input argument of String String type.
v) If we are passing input argument as number or float then internally it is considered as String

vi) Return type of this function is number.

Example:

```
SQL> select length('SQL  Course') from dual;

LENGTH('SQLCOURSE')
-------------------
                 11

SQL> select length('SQL   Course') from dual;

LENGTH('SQLCOURSE')
-------------------
                 12

SQL> select length('SQL    Course') from dual;

LENGTH('SQLCOURSE')
-------------------
                 13
```

3) Instr()

1) This function is used to find the numeric position of given character.
2) This function accept two input arguments, so syntax of this function is as below

   instr(source_string, target_char)

   Example: instr('ABCDEFGH', 'B')

   out put is : 2
3) If given character is present then it will return numeric position of that character

4) If given character is not present then it will return zero.

Example: instr('ABCS', 'W')

Out put is : 0

5) If character occured multiple times then it will return the numeric position of first occurance of that character

Example: instr('Hello', 'l')

Out put is : 3

6) This function is case sensitive. i.e If we are finding lower case character numeric position but in source string that character is in upper case then it will return zero as out put

Example : instr('Hello','h')

Out put is : 0

Practical Examples

```
SQL> select instr('Hello','H') from dual;

INSTR('HELLO','H')
------------------
                 1

SQL> select instr('Hello','l') from dual;

INSTR('HELLO','L')
------------------
                 3

SQL> select instr('Hello','b') from dual;

INSTR('HELLO','B')
------------------
                 0
```

4) Lpad()

1) This function add the character value to left hand side of string.

2) Here meaning of pad is to add.

3) This function is used to format the out put.

4) This function takes 3 input arguments, so syntax of this function is as below

lpad(source_string, n, target_character)

5) Here source string and target character value can be any type- char, varchar, varchar2, CLOB, Number

6) The value of n shoud not be 0 or negative. by mistake if we are passing negative value then query get executed but no out is returned.

7) In lpad function we have two cases.

Case 1: If value of n is less than source string.

Example: lpad('ABCD', 2, '*')

Here length of source string (ABCD) is 4 and value of n is 2. so here in out put it will show only first two characters.

```
SQL> select lpad('ABCD',2,'*') from dual;

LP
--
AB
```

Case 2: If value of n is greater than length of source String

Example: lpad('ABCD',10,'*')

Here, length of source string is 4 and value of n is 10. Value of n is greater than length of string so in that case to match the expected out put * character get added 6 times to left hand side of string so final out put string become 10 character.

```
SQL> select lpad('ABCD',10,'*') from dual;

LPAD('ABCD
----------
******ABCD
```

5) rpad() function is exactly same as lpad but rpad function add the character value to right hand side of source string

Example:

```
SQL> select rpad('ABCD',10,'*') from dual;

RPAD('ABCD
----------
ABCD******
```

trim()

1) trim function removes given character from beging or from ending of source string

2) by default trim function removes the character from both side.

3) if we want remove character from beging then use leading keyword or use LTRIM() function

4) If we want to remove character from ending then use trailing keyword or use RTRIM() function

5) trim function is case sensitive.

Examples

```
SQL> select trim('H' from 'Hello HI') from dual;

TRIM('H
-------
ello HI

SQL> select trim('H' from 'Hello HiH') from dual;

TRIM('H
-------
ello Hi
```

```
SQL> select trim(leading 'H' from 'Hello HiH') from dual;

TRIM(LEA
--------
ello HiH

SQL> select trim(trailing 'H' from 'Hello HiH') from dual;

TRIM(TRA
--------
Hello Hi
```

```
SQL> select ltrim('Hello IHH','H') from dual;

LTRIM('H
--------
ello IHH

SQL> select rtrim('Hello IHH','H') from dual;

RTRIM('
-------
Hello I
```

Replace()

1) Replace function is used to replace a character or sequence of characters with another character or sequence of character.

2) This function takes 3 input parameters, so syntax of Replace function is as below

replace(source_string, source_char, target_char)

Example: replace('AABBCCDDABA' ,'AA', 'W')

OutPut : WBBCCDDABA

3) Replace function is case sensitive.

substr()

1) This function is used to find sub string from given string

2) This function takes 3 input arguments, so Syntax of this function is as below

substr(source_string, start_index, end_index)

Example : substr('HelloWorld', 1,5)

Out put is : Hello

Example : substr('HelloWorld', 3,7)

Out put is : lloWo

3) if value of starting index is positive then counting start from left hand side.

4) if value of starting index is negative then counting start from right hand side.

Examples : substr('HelloWorld', -4, 5)

out put is : orld

```
SQL> select substr('HelloWorld',1,5) from dual;

SUBST
-----
Hello

SQL> select substr('HelloWorld',2,5) from dual;

SUBST
-----
elloW
```

```
SQL> select substr('HelloWorld',-2,5) from dual;

SU
--
ld
```

If value of end index is negative, we don’t get out

```
SQL> select substr('HelloWorld',2,-5) from dual;

S
-
```