SQL FUNCTIONS

Prepared By: Dr. Vipul Vekariya

.

SQL FUNCTIONS

- Definition of Function
- Types of SQL Function
- Numeric Function
- String Function
- Conversion Function
- Date Function

SQL Function

- Sub program of SQL Lang.
- Used to do Operation on the SQL Expression.
- SQL Functions works on Columns of Table.
- Returns a Value as a Result of the Operation.
- After the Execution of the Function One or More Values can be returned by the Function.

Types of SQL Function

- **According to the Processing on the Value of Column Sql Function can be Classified.**
- **Two Types of SQL Function**
  - **Group Functions**
    - **"Functions act on Set of Values are known as Group Function"**
    - **SUM,AVG, MIN, MAX (Aggregate Function)**
  - **Scalar Functions**
    - **"Functions act on only one value at a time known as Scalar Fn"**
    - **Length, ASCII**

Types of SQL Functions

- According to the SQL Data Type
  - Numeric Functions:
    - To processing Number Data type
  - String Functions:
    - To Processing on String Data type
  - Conversion Functions:
    - To Convert Data type from one data type to another
  - Date Functions:
    - To processing on Date Data Type

AVG Function

- Returns Average of all Row Values for particular Column.
- Null Values are ignored in Calculation.

- **Syntax:**
  - AVG ([DISTINCT | ALL] column_name) ;
- **Example:**
  - SELECT AVG(BALANCE) “AVERAGE _BALANCE” FROM ACCOUNT_MASTER;

MIN Function

- **Return a minimum Value of Expression**

- **Syntax:**
  - **MIN ([DISTINCT | ALL]** column_name**) ;**

- **Example:**
  - **SELECT MIN(BALANCE) "Minimum_Balance"**
    FROM ACCOUNT;

MAX Function

- **Returns Maximum value present in particular column of Data table.**

- **Syntax:**
  - **MAX ([DISTINCT | ALL]** column_name**);**

- **Example:**
  - **SELECT MAX(BALANCE) "maximum balance"**
    FROM ACCOUNT_MASTER ;

COUNT Function

- **Returns Number of Rows Present in Particular Column**
- **Count(*) returns the number of rows in the table including duplicates and those with nulls.**

- **Syntax:**
  - **COUNT ([DISTINCT | ALL]** column_name**) ;**

- **Example:**
  - **SELECT COUNT(ACCOUNT_NO) "no. of records" FROM ACCOUNT_MASTER;**

SUM Function

- **Returns Sum of Values present in particular Column**

- **Syntax:**
  - **SUM([DISTINCT | ALL]** column_name**) ;**

- **Example:**
  - **SELECT SUM(BALANCE) "TOTAL_BALANCE"**
    FROM ACCOUNT_MASTER;

Dual Table In SQL

- Table DUAL is very simple. It has only one column and contains only one row with value 'X'.
- SQL> desc dual;
- Name Null? Type

DUMMY VARCHAR2(1)

- SQL> select * from dual;

D

X

Dual Table In SQL(Cont).

- Table DUAL - is part of system dictionary and you never ever should make any changes to it in real database with working users.
- DUAL is a part data dictionary and owned by SYS. You should not make modifications to this table.
- Used to refer an object which does not have any physical reference in database table.
- For example we need a calculation 2*2
- Then we always used select 2*2 from ? (which table name with from)
- So at that time we used dummy table as dual.

Numeric Functions

ABS Function

POWER Function

ROUND Function

SQRT Function

EXP Function

MOD Function

GREATEST Function

FLOOR Function

TRUNC Function

ABS Function

- **Absolute Value Expression returned by this function**
- **Syntax:**
  - **ABS(n);**

- **Example :**
  - **Select ABS(-10) From dual;**
  - **Output:**
    - **10 Display on Oracle O/P Screen**

POWER Function

- **Returns Power of the Expression.**

- **Syntax:**
  - **POWER(m,n);**

- **Example:**
  - **Select Power(6,2) "Power" From dual;**
  - **Output:**
    - **36 Displayed on The Oracle Screen → 6^2**

ROUND Function

- Used to Get Rounded Value of Expression
- Syntax:
  - ROUND (n,m);
  - n = Numeric Value with Decimal Point
  - m = Rounded Position
- Example:
  - Select Round(10.29,1) “Rounded_Value” From dual;
  - Output:
    - 10.3

SQRT Function

- **Used to Find out the Square Root of Expression.**

- **Syntax:**
  - **SQRT (n);**
    - **Returns Square root of n**
    - **n must be Positive if n < 0 then Null will be returned**

- **Example:**
  - **Select Sqrt(25) "Square_Value" From dual;**
  - **Output:**
    - **5**

Exponent Function

- **Returns e raised to nth Power.**

- **Syntax:**
  - **EXP (n);**
  - **e^n**
- **Example:**
  - **Select exp(5) "Exponent" From dual;**
  - **Output:**
    - 148.413159 (e = 2.71828183)

GREATEST Function

- **Used to Find out the Greatest Valued from the Expression.**

- **Syntax:**
  - **GREATEST (exp1,exp2,…,expn);**
- **Example:**
  - **Select Greatest(21,10,30) “Great_Value” From dual;**
  - **Output:**
    - **30**

LEAST Function

- **Used to Find out Lowest Value from the Expression**

- **Syntax:**
  - **LEAST (exp1,exp2,…,expn);**

- **Example:**
  - **Select Least(35,75,25) “Least_Value” From dual;**
  - **Output:**
    - **25**

MOD Function

- **Used to Find out the Reminder of Division Function**

- **Syntax:**
  - **MOD (m,n);**
  - **m/n → Reminder is a result**

- **Example:**
  - **Select Mod(18,7) "Reminder" From dual;**

Truncation Function

- **Returns Truncated Values after the decimal position**

- **Syntax:**
  - **TRUNC (number, decimal_Places);**

- **Example:**
  - **Select Trunc(17.235,1) "Truncated_Value" From dual;**
  - **Output:**
    - **17.2**

Floor Function

- **Returns Largest Integer Value of Expression.**
- **Syntax:**
  - **FLOOR (n)**
- **Example:**
  - **Select Floor(24.18) “Large_Int” From dual;**
  - **Output:**
    - 24

CEIL Function

- **Returns Smallest Integer value which is Greater or Equal to the number.**

- **Syntax:**
  - **CEIL (n);**
- **Example:**
  - **Select ceil(24.83) "Value" From dual;**
  - **Output:**
    - **25**

String Function

LOWER Function

UPPER Function

INITCAP Function

SUBSTR Function

ASCII Function

COMPOSE Function

DECOMPOSE Function

INSTR Function

TRANSLATE Function

LENGTH Function

LTRIM Function

RTRIM Function

TRIM Function

LPAD Function

RPAD Function

VSIZE Function

LOWER Function

- **Return all character with lowercase letters.**
- **SYNTAX:**
  - **LOWER(Char);**
- **EXAMPLE:**
  - **Select Lower('XYZ') "Lower" FROM dual;**
  - **OUTPUT:**

Lower
xyz

UPPER Function

- **Return character with Uppercase Letter.**
- **Syntax:**
  - **UPPER (Char);**
- **Example:**
  - **Select Upper('xyz') "Upper_Case" From dual;**
  - **Output:**

Upper_Case
XYZ

INITCAP Function

- **Return First Character of String with Upper Case Letter.**
- **Syntax:**
  - **INITCAP (Char) ;**
- **Example:**
  - **Select Initcap('abc') "First_Case" From dual;**
  - **Output:**

First_Case
Abc

SUBSTR Function

- **Returns Substring of Main String according to the Specified position of Characters upto specified length of Characters.**
- **Syntax:**
  - **SUBSTR (string, start_position, length) ;**
- **Example:**
  - **Select Substr('SECURE', 3,4) "Sub_Str" From dual;**
  - **Output:**

Sub_Str
CURE

ASCII Function

- **Returns Ascii value of Specified Character.**
- **Syntax:**
  - **ASCII (Char) ;**
- **Example:**
  - **Select ASCII('a') "Ascii_Value" From dual;**
  - **Output:**

Ascii_Value
97

INSTR Function

- **Return location of substring in the main string.**
- **Syntax:**
- **INSTR (string1, string2, start_position, nth_appearance)**
  - **String1- Main String**
  - **String2 – sub string which is find out from main string**
  - **Start_postion – position in string1 where the search will start**
  - **Nth appearance – is the nth appearance of string**
- **Example:**
  - **Select Instr('SCT on the net','t') "Instr"from dual;**
  - **Output:**

Instr
8

LENGTH Function

- Returns the length of words in the string
- Syntax:
  - LENGTH (word) ;
- Example:
  - Select Length('xyz') "Length" From dual;
  - Output:

Length
3

LTRIM Function

- **Remove Character from left of String.**
- **Syntax:**
  - **LTRIM (char, set);**
- **Example:**
  - **Select Ltrim('xyz','x') "Ltrim" From dual;**
  - **Output:**

Ltrim
yz

RTRIM Function

- **Remove character from Right of String.**
- **Syntax:**
  - **RTRIM (char, set);**
- **Example:**
  - **Select Rtrim('xyz','z') "Rtrim" From dual;**
  - **Output:**

Rtrim
xy

LPAD Function

- **Return String of with specified Character at left side of string.**
- **Syntax:**
  - **LPAD (char1, n,[char2]);**
- **Example:**
  - **Select Lpad('xyz' , 5 , '*') "Lpad" From dual;**
  - **Output:**

Lpad
**xyz

RPAD Function

- **Add specified character to the right side of main string**
- **Syntax:**
  - **RPAD (char1, n, [char2]);**
- **Example:**
  - **Select Rpad('xyz',6,'p') "Rpad" From dual;**
  - **Output:**

Rpad
xyzppp

VSIZE Function

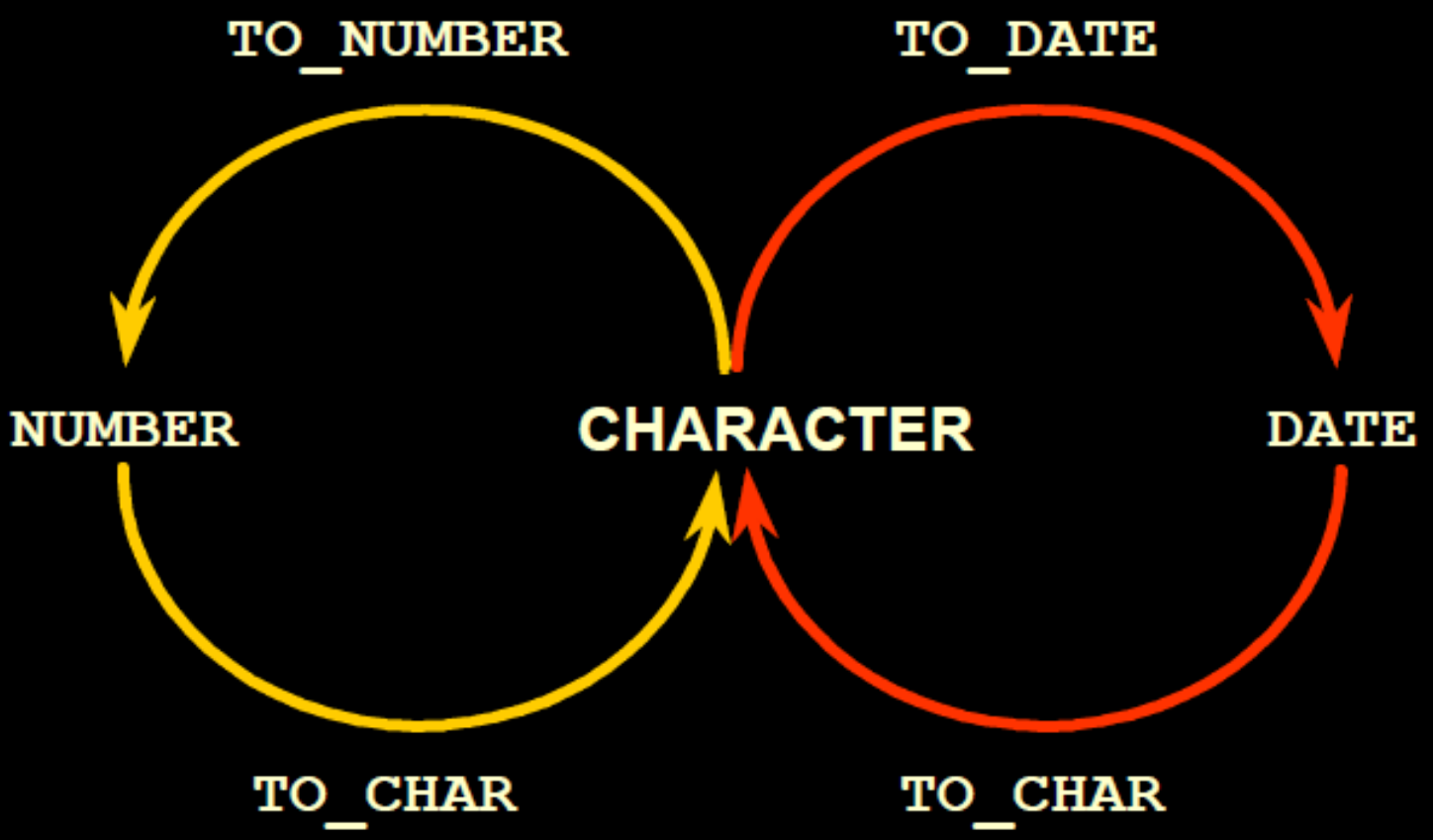
- **Return the number of bytes occupied by the expression**
- **Syntax:**
  - **VSIZE (expression);**
- **Example:**
  - **Select Vsize('abc XYZ') "Byte" From dual;**
  - **Output:**

Byte
7

Conversion Function

- **Conversion functions are used to convert the data type of table field**
- **TO_Number**
- **TO_CHAR**

Explicit Data Type Conversion



TO_NUMBER Function

- **Convert a char data type of table field to Number data type**

- **Syntax:**
  - **TO_NUMBER(char);**

- **Example:**
  - **select to_number('15‘) from dual;**

```
TO_NUMBER('15')
--------------
            15
```

TO_CHAR Function

- Convert the value of Number/Date data type to the Character Data type.
- Syntax:

```
TO_CHAR(date, 'format_model')
```

Elements of the Date Format Model

YYYY	Full year in numbers
YEAR	Year spelled out
MM	Two-digit value for month
MONTH	Full name of the month
MON	Three-letter abbreviation of the month
DY	Three-letter abbreviation of the day of the week
DAY	Full name of the day of the week
DD	Numeric day of the month

Example

```
SELECT last name,
       TO_CHAR(hire_date, 'fmDD Month YYYY')
       AS HIREDATE
FROM   employees;
```

LAST_NAME	HIREDATE
King	17 June 1987
Kochhar	21 September 1989
De Haan	13 January 1993
Hunold	3 January 1990
Ernst	21 May 1991
Lorentz	7 February 1999
Mourgos	16 November 1999

...

20 rows selected.

Date Function

- ADD_MONTHS()
- LAST_DAY()
- MONTHS_BETWEEN()
- NEXT_DAY()
- TO_DATE()

ADD_MONTHS

- **Return date after adding the number of the months specified in the function.**
- **Syntax:**
  - **ADD_MONTHS (d,n);**
- **Example:**
  - **Select Add_Months(Sysdate,4)"Add_Month" from dual;**
  - **Output:**

Add_Month

27-JAN-09

LAST_DAY

- **Return last date of month which is specified in the Function.**
- **Syntax:**
  - **LAST_DAY (d) ;**
- **Example:**
  - **Select SYSDATE, LAST_DAY(Sysdate)"Last_Date" from dual;**
  - **Output:**

Sysdate	Last_Date
29-sep-08	30-sep-08

MONTHS_BETWEEN

- **Returns number of months between two date which is specified in the function**
- **Syntax:**
  - **MONTHS_BETWEEN(d1 , d2)**
- **Example:**
  - **Select Months_Between('02-feb-08,'02-jan-08') "Months" from dual;**
  - **Output:**

Months
1

NEXT_DAY

- Returns the date of the first weekday named by char that is after the date named by date.
- Syntax:
  - NEXT_DAY (date, char)
- Example:
  - Select NEXT_DAY ('06-JULY-02','Saturday') "Next_Day" from dual;
  - Output:

Next_Day

13-july-02

TO_DATE()

- TO_DATE convert a char value in to date value.

TO_DATE(<char value>[<fmt>])

select TO_DATE('09/06/83','DD/MM/YY) from dual;

o/p: 09-june-83