

Inbuilt Functions in SQL

Ex. No : 4

Character Functions

It calculates the ASCII equivalent of the first character of the given

input string. ASCII(<Character>)

ascii('A')	would return 65
------------	--------------------

ascii('a')	would return 97
------------	--------------------

ascii('a8')	would re- turn 97
-------------	----------------------

CHR(<Character>)

Returns the character equivalent of the given integer. Example

```
SELECT CHR(65), CHR(97) FROM dual;
```

O/P A a

CONCAT(<string1>,<string2>)

This function returns String2 appended to

String1. Example:

```
SELECT CONCAT('Fname', 'Lname') Emp_name FROM emp;
```

INITCAP(<String>)

This function returns String with the first character of each word in upper case and rest of all in lower case.

Example:

```
SELECT INITCAP('oracle tutorial')
```

```
FROM Dual;
```

O/p Oracle Tutorial

INSTR

instr(string1, string2 [, start_position [, nth_Appearance]]):

where,

1. string1 is the string to search.
2. string2 is the substring to search for in string1.
3. start_position is the position in string1 from where the search will start.

This argument is optional. If not mentioned, it defaults to 1.

The first position in the string is 1. If the start_position is negative, the function counts back-ward direction.

4. nth_appearance is the nth appearance of string2. This is optional. If not defined, it defaults to Example

```
SELECT INSTR('Character','r',1,1) POS1, INSTR('Character','r',1,2) POS2,
INSTR('Character','a',-1,2) POS3,INSTR('character','c',) POS4
```

```
FROM
Dual;
```

pos1	pos2	pos3	pos4
4	9	3	6

LENGTH(<Str>)

Returns length of a string

```
select length('Sql Tutorial') as len
from dual;
O/p len
12
```

LOWER(<Str>)

This function returns a character string with all characters in lower case.

UPPER(<Str>)

This function returns a character string with all characters in upper case.

LPAD(<Str1>,<i>[,<Str2>])

This function returns the character string Str1 expanded in length to i characters, using Str2 to fill in space as needed on the left side of Str1.

Example

```
SELECT LPAD('Oracle',10,'.') lapd_doted from Dual, would return Oracle
SELECT LPAD('RAM', 7) lapd_exa from Dual would return ' RAM'
```

RPAD(<Str1>,<i>[,<Str2>])

RPAD is same as LPAD but Str2 is padded at the right side

LTRIM(<Str1>[,<Str2>])

The LTRIM function removes characters from the left side of the character String, with all the leftmost characters that appear in another text expression removed.

This function returns Str1 without any leading character that appears in Str2. If Str2 characters are leading character in Str1, then Str1 is returned unchanged. Str2 defaults to a single space.

Example

Select

LTRIM('datawarehousing','ing')


trim1 , LTRIM('datawarehousing ')

trim2

, LTRIM(' datawarehousing') trim3

, LTRIM('datawarehousing','data')

trim4 from dual

trim1	trim2	trim3	trim4
			
datawarehousing	datawarehousing	datawarehousing	warehousing

RTRIM(<Str1>[,<Str2>])

Same as LTRIM but the characters are trimmed from the right side

TRIM([<Str1>]<Str2> FROM]<Str3>)

If present Str1 can be one of the following literal: LEADING, TRAILING, BOTH.

This function returns Str3 with all C1(leading trailing or both) occurrences of characters in Str2 removed.

If any of Str1, Str2 or Str3 is Null, this function returns a Null.

Str1 defaults to BOTH, and Str2 defaults to a space character.

Example

```
SELECT TRIM(' Oracle ') trim1, TRIM('Oracle ') trim2 FROM Dual;
```

Ans trim1 trim2

Oracle Oracle

It'll remove the space from both
string.

REPLACE(<Str1>,<Str2>[,<Str3>]

r 3>]

This function returns Str1 with all occurrence of Str2 replaced with Str3

Example

```
SELECT REPLACE (,'Oracle', 'Ora', 'Arti') replace_exa  
FROM Dual;
```

O/p replace_exa

Article

Essential Numeric

Functions

ABS()

Select Absolute value

```
SELECT ABS(-25) "Abs" FROM DUAL;
```

Abs

15 _____

ACOS ()

Select cos value

```
SELECT ACOS(.28)"Arc_Cosine" FROM DUAL;
```

ASIN ()

Select sin value

```
SELECT ASIN(.6)"Arc_Cosine" FROM DUAL;
```

ATAN()

Select tan value

```
SELECT ATAN(.6)"Arc_Cosine" FROM DUAL;
```

CEIL()

Returns the smallest integer greater than or equal to the order total of a specified SELECT CEIL(239.8) FROM Dual would return 240

FLOOR()

Returns the largest integer equal to or less than value.

```
SELECT FLOOR(15.65) "Floor" FROM DUAL;  
Floor
```

15

MOD()

Return modulus value

```
SELECT MOD(11,3) "Mod" FROM DUAL;
```

Modulus

2

POWER()

```
SELECT POWER(3,2) "Power" FROM DUAL;  
power
```

9

ROUND (number)

```
SELECT ROUND(43.698,1) "Round" FROM DUAL;  
Round
```

43.7

TRUNC (number)

The TRUNC (number) function returns n1 truncated to n2 decimal places. If n2 is omitted, then n1 is truncated to 0 places. n2 can be negative to truncate (make zero) n2 digits left of the decimal point.

```
SELECT TRUNC(12.75,1) "Trunc" FROM DUAL;
```

Trunc

12.75

```
SELECT TRUNC(12.75,-1) "Trunc" FROM DUAL;
```

Trunc

10

Date And Time Function

ADD_MONTHS(date,number_of_
month)

```
SELECT SYSDATE, ADD_MONTHS(SYSDATE,2), ADD_MONTHS(SYSDATE,-  
2) FROM
```

DUAL;

Result:

SYSDATE

ADD MONTH

10-Feb-13 10-Apr-13 10-Dec-13

EXTRACT(<type> FROM <date>)

'Type' can be YEAR, MONTH, DAY, HOUR, MIN, SECOND,
TIME_ZONE_HOUR, TIME_ZONE_MINUTE, TIME_ZONE_REGION

```
SELECT SYSDATE, EXTRACT(YEAR FROM SYSDATE)YEAR,  
EXTRACT(DAY FROM SYSDATE)DAY , EXTRACT(TIMEZONE_HOUR  
FROM SYSTIMESTAMP) TZH  
FROM DUAL;
```

LAST_DAY(<date>)

Extract last day of
month Example:

```
SELECT SYSDATE, LAST_DAY(SYSDATE) END_OF_MONTH FROM DUAL;
```

Result: SYS-
DATE END_OF_MO

-----	-----
4-	31-Aug-
Aug-	18
18	

NEXT_DAY(<date>,<day>)

SELECT NEXT_DAY('31-Aug-18','SUN') "FIRST MONDAY OF
SEPTEMBER" FROM DUAL;

O/P FIRST MONDAY OF SEPTEMBER

03-Sep-18

ROUND (date[,<fmt>])

SELECT SYSDATE, ROUND(SYSDATE,'MM'),
ROUND(SYSDATE,'YYYY') FROM DUAL;

Result:

SYSDATE	ROUND(SYSDATE,'MM')	ROUND(SYSDATE,'YYYY')
---------	---------------------	-----------------------

10-FEB-18	01-MAR-18	01-JAN-18
-----------	-----------	-----------

TRUNC(date[,<fmt>])

SELECT SYSDATE, TRUNC(SYSDATE,'MM'), TRUNC(SYSDATE,'YYYY')
FROM DUAL;

Result:	TRUNC(SYSDATE,'MM')	TRUNC(SYSDATE,'YYYY')
SYS-DATE	10-FEB-18	01-JAN-18

10-FEB-18	01-JAN-18
18	

MONTHS_BETWEEN function returns the number of months between

date1 and date2. SYNTAX

The syntax for the Oracle/PLSQL MONTHS_BETWEEN function is:

MONTHS_BETWEEN(date1, date2)

Parameters or Arguments

date1 and date2 are the dates used to calculate the number of months.

If a fractional month is calculated, the MONTHS_BETWEEN function calculates the fraction based on a 31-day month.