



FUNCIONES EN ORACLE

Funciones de fecha

En esta sección se hace un repaso de las funciones de fecha (date function) ejemplos.

Están descritas funciones como ADD_MONTHS, MONTHS_BETWEEN...

Funcion AVG

Determina la media del valor de una columna.

```
AVG([DISTINCT | ALL] n)
```

Funcion UPPER

Devuelve la cadena en mayúsculas.

```
UPPER(cadena)
```

Función GREATEST

Devuelve el mayor de una lista de expresiones.

```
GREATEST(exp1, exp2, ...)
```

Funcion USER

Devuelve el identificador de usuario de la sesión actual.

```
select USER from dual;
```

Funcion TO_NUMBER

Convierte una cadena en un valor de tipo NUMBER. Ver en TO_CHAR ejemplos de formato.

```
TO_NUMBER(char [,fmt])
```

FUNCIONES

- PL/SQL proporciona un amplio catálogo de funciones, tanto de SQL como propias:
 - **De manejo de cadenas de caracteres:** INITCAP(), INSTR(), LENGTH(), LOWER(), LPAD(), LTRIM(), REPLACE(), RPAD(), RTRIM(), SUBSTR(), UPPER().
 - **Numéricas:** ABS(), CEIL(), FLOOR(), MOD(), POWER(), ROUND(), SIGN(), SQRT(), TRUNC().
 - **De manejo de fechas:** ADD_MONTHS(), LAST_DAY(), MONTHS_BETWEEN(), NEXT_DAY(), SYSDATE(), etc.
 - **De conversión:** TO_CHAR(), TO_DATE(), TO_NUMBER().
 - **De control de errores:** SQLCODE(), SQLERRM().
 - **Otras:** NVL(), USER(), DECODE(), etc

Funciones de Cadena

- **chr(x)**: retorna un carácter equivalente al código enviado como argumento "x".
Ejemplo:
=> select chr(65) from dual;-- retorna 'A'.
- **concat(cadena1,cadena2)**: concatena dos cadenas de caracteres; es equivalente al operador ||
- **initcap(cadena)**: retorna la cadena enviada como argumento con la primera letra (letra capital) de cada palabra en mayúscula.
- **lower(cadena)**: retorna la cadena enviada como argumento en minúsculas.
- **upper(cadena)**: retorna la cadena enviada como argumento en mayúsculas.
- **lpad(cadena,longitud,cadenarelleno)**: retorna la cantidad de caracteres especificados por el argumento "longitud", de la cadena enviada como primer argumento (comenzando desde el primer carácter); si "longitud" es mayor que el tamaño de la cadena enviada, rellena los espacios restantes con la cadena enviada como tercer argumento (en caso de omitir el tercer argumento rellena con espacios); el relleno comienza desde la izquierda. Ejemplos:
=> select lpad('alumno',10,'xyz') from dual;-- retorna 'xyzxalumno'
=> select lpad('alumno',4,'xyz') from dual;-- retorna 'alum'

Funciones de Cadena

rpad(cadena,longitud,cadenarelleno): retorna la cantidad de caracteres especificados por el argumento "longitud", de la cadena enviada como primer argumento (comenzando desde el primer caracter); si "longitud" es mayor que el tamaño de la cadena enviada, rellena los espacios restantes con la cadena enviada como tercer argumento (en caso de omitir el tercer argumento rellena con espacios); el relleno comienza desde la derecha (último caracter). Ejemplos:

=> select rpad('alumno',10,'xyz') from dual;-- retorna 'alumnoxyzx'

=> select rpad('alumno',4,'xyz') from dual;-- retorna 'alum'

ltrim(cadena1,cadena2): borra todas las ocurrencias de "cadena2" en "cadena1", si se encuentran al comienzo; si se omite el segundo argumento, se eliminan los espacios. Ejemplo:

=> select ltrim('la casa de la cuadra','la') from dual;-- ' casa de la cuadra'

=> select ltrim(' es la casa de la cuadra','la') from dual;-- no elimina ningún caracter

=> select ltrim(' la casa') from dual;-- 'la casa'

rtrim(cadena1,cadena2): borra todas las ocurrencias de "cadena2" en "cadena1", si se encuentran por la derecha (al final de la cadena); si se omite el segundo argumento, se borran los espacios. Ejemplo:

=> select rtrim('la casa lila','la') from dual;-- 'la casa li'

=> select rtrim('la casa lila ','la') from dual;-- no borra ningún caracter

=> select rtrim('la casa lila ') from dual; --'la casa lila'

Funciones de Cadena

- **trim(cadena):** retorna la cadena con los espacios de la izquierda y derecha eliminados.
=> select trim(' oracle ') from dual;--'oracle'
- **replace(cadena,subcade1,subcade2):** retorna la cadena con todas las ocurrencias de la subcadena de reemplazo (subcade2) por la subcadena a reemplazar (subcade1).
=> select replace('xxx.oracle.com','x','w') from dual; -- retorna 'www.oracle.com'.
- **substr(cadena,inicio,longitud):** devuelve una parte de la cadena especificada como primer argumento, empezando desde la posición especificada por el segundo argumento y de tantos caracteres de longitud como indica el tercer argumento.
=> select substr('www.oracle.com',1,10) from dual;-- 'www.oracle'
=> select substr('www.oracle.com',5,6) from dual;-- 'oracle'
- **length(cadena):** retorna la longitud de la cadena enviada como argumento.

Funciones de Cadena

- **instr (cadena,subcadena):** devuelve la posición de comienzo (de la primera ocurrencia) de la subcadena especificada en la cadena enviada como primer argumento. Si no la encuentra retorna 0.
=> select instr('Jorge Luis Borges','or') from dual;-- 2
=> select instr('Jorge Luis Borges','ar') from dual;-- 0, no se encuentra
- **translate():** reemplaza cada ocurrencia de una serie de caracteres con otra serie de caracteres. La diferencia con "replace" es que aquella trabaja con cadenas de caracteres y reemplaza una cadena completa por otra, en cambio "translate" trabaja con caracteres simples y reemplaza varios. En el siguiente ejemplo se especifica que se reemplacen todos los caracteres "O" por el caracter "0", todos los caracteres "S" por el caracter "5" y todos los caracteres "G" por "6":
=> select translate('JORGE LUIS BORGES','OSG','056') from dual;--'J0R6E LUI5 B0R6E5'
- Se pueden emplear estas funciones enviando como argumento el nombre de un campo de tipo carácter.

Funciones Numéricas

- **abs(x)**: retorna el valor absoluto del argumento "x".
=> select abs(-20) from dual;--retorna 20.
- **ceil(x)**: redondea a entero, hacia arriba, el argumento "x".
=> => select ceil(12.34) from dual;--retorna 13.
- **floor(x)**: redondea a entero, hacia abajo, el argumento "x".
=> select floor(12.34) from dual; --12
- **mod(x,y)**: devuelve el resto de la división x/y.
=> select mod(10,3) from dual;--retorna 1.
- **power(x,y)**: retorna el valor de "x" elevado a la "y" potencia.
=> select power(2,3) from dual;--retorna 8.

Funciones Numéricas

- **round(n,d)**: retorna "n" redondeado a "d" decimales; si se omite el segundo argumento, redondea todos los decimales. Si el segundo argumento es positivo, el número de decimales es redondeado según "d"; si es negativo, el número es redondeado desde la parte entera según el valor de "d".
 - => select round(123.456,2) from dual;-- retorna "123.46"
 - => select round(123.456,1) from dual;-- 123.5
 - => select round(123.456,-1) from dual;-- 120, redondea desde el primer valor entero (hacia la izquierda).
 - => select round(123.456,-2) from dual;-- 100, redondea desde el segundo valor entero (hacia la izquierda).
 - => select round(123.456) from dual;-- 123.
- **sign(x)**: si el argumento es un valor positivo, retorna 1, si es negativo, devuelve -1 y 0 si es 0.
 - => select sign(-120) from dual;--retorna -1
 - => select sign(120) from dual;--retorna 1

Funciones Numéricas

- **trunc(n,d)**: trunca un número a la cantidad de decimales especificada por el segundo argumento. Si se omite el segundo argumento, se truncan todos los decimales. Si "d" es negativo, el número es truncado desde la parte entera.

=> select trunc(1234.5678,2) from dual;--retorna 1234.56 => select trunc(1234.5678,-2) from dual;--retorna 1200 => select trunc(1234.5678,-1) from dual;--retorna 1230

=>select trunc(1234.5678) from dual;--retorna 1234

- **sqrt(x)**: devuelve la raíz cuadrada del valor enviado como argumento.

=> select sqrt(9) from dual;--retorna 3

Oracle dispone de funciones trigonométricas que retornan radianes, calculan seno, coseno, inversas, etc.: acos, asin, atan, atan2, cos, cosh, exp, ln, log, sin, sinh, tan, tanh.

Funciones que devuelven valores numéricos:

ASCII(cad) : Devuelve el valor ASCII de la primera letra de la cadena "cad".

INSTR (cad1, cad2 [, comienzo [,m]]) : Permite una búsqueda de un conjunto de caracteres en una cadena pero no suprime ningún carácter después.

LENGTH (cad) : Devuelve el número de caracteres de cad.

Funciones de Fechas

Oracle dispone de varias funciones que operan con tipos de datos "date". Estas son algunas:

- **add_months(f,n):** agrega a una fecha, un número de meses. Si el segundo argumento es positivo, se le suma a la fecha enviada esa cantidad de meses; si es negativo, se le resta a la fecha enviada la cantidad de meses.

=> select add_months('10/06/2007',5) from dual; --retorna "10/11/07"

=> select add_months('10/06/2007',-5) from dual; --retorna "10/01/07"

=> select add_months('30/01/2007',1) from dual;-- retorna "25/02/07" ya que es el último día de ese mes.

- **last_day(f):** retorna el ultimo día de mes de la fecha enviada como argumento.

=> select last_day('10/02/2007') from dual;-- "28/02/07"

=> select last_day('10/08/2007') from dual;-- "31/08/07"

- **months_between(f1,f2):** retorna el numero de meses entre las fechas enviadas como argumento.

=> select months_between('19/05/2003','21/06/05') from dual;-- retorna -25,06

- **next_day(fecha,dia):** retorna una fecha correspondiente al primer día especificado en "dia" después de la fecha especificada.

En el siguiente ejemplo se busca el lunes siguiente a la fecha especificada:

=> select next_day('10/08/2007','LUNES') from dual; --13/08/07

Funciones de Fechas

- **current_date**: retorna la fecha actual.
=> select current_date from dual;
- **current_timestamp**: retorna la fecha actual
=> select current_timestamp from dual; Retorna: 05/05/15 13:40:18,731000000 EUROPE/PARIS
- **sysdate**: retorna la fecha y hora actuales en el servidor de Oracle.
=> select sysdate from dual; -- 05/05/15
- **systimestamp**: retorna fecha y hora actuales.
=> select systimestamp from dual; Retorna 05/05/15 13:43:07,614000000 +02:00
- **to_date**: convierte una cadena a tipo de dato "date".
=> select to_date ('05-SEP-2007 10:00 AM','DD-MON-YYYY HH:MI AM') from dual; --Retorna 05/09/07

Funciones de Fechas

- **to_char**: convierte una fecha a cadena de caracteres.
=> `select to_char('10/10/2007') from dual;`
- **extract(parte, fecha)**: retorna la parte (especificada por el primer argumento) de una fecha. Puede extraer el año (year), mes (month), día (day), hora (hour), minuto (minute), segundo (second), etc.
=> `select extract(month from sysdate) from dual;` retorna el número mes de la fecha actual. --5
- En Oracle: Los operadores aritméticos "+" (más) y "-" (menos) pueden emplearse con fechas.
=> `select sysdate-3;` Retorna 3 días antes de la fecha actual.
=> `select to_date('15/12/2007')-5 from dual;` --Retorna 10/12/07
- Se pueden emplear estas funciones enviando como argumento el nombre de un campo de tipo date.

Funciones de conversión:

- **TO_CHAR** : Transforma un tipo DATE ó NUMBER en una cadena de caracteres.
TO_CHAR(d, fmt)
Convierte la fecha d a una cadena de caracteres, opcionalmente de acuerdo con el formato fmt.
=> select to_char(sysdate) from dual;
- **TO_DATE** : Transforma un tipo NUMBER ó CHAR en DATE.
TO_DATE(cad,fmt)
Convierte la cadena cad de tipo varchar2 a fecha, opcionalmente de acuerdo con el formato fmt.
=> select to_date('1-JAN-97') from dual;
- **TO_NUMBER** : Transforma una cadena de caracteres en NUMBER.
TO_NUMBER(cad,fmt)
Convierte la cadena cad a un número, opcionalmente de acuerdo con el formato fmt.
=> select to_number('12345') from dual;

Funciones de grupos de valores:

AVG(n) : Calcula el valor medio de "n" ignorando los valores nulos.

COUNT (* | Expresión) : Cuenta el numero de veces que la expresión evalúa algún dato con valor no nulo. La opción "*" cuenta todas las filas seleccionadas.

MAX (expresión) : Calcula el máximo.

MIN (expresión) : Calcula el mínimo.

SUM (expresión) : Obtiene la suma de los valores de la expresión.

GREATEST (valor1, valor2) : Obtiene el mayor valor de la lista.

LEAST (valor1, valor2) : Obtiene el menor valor de la lista.

OTRAS FUNCIONES

- **DECODE(var, val1, cod1, val2, cod2, ..., defecto)**
Convierte el valor de var, de acuerdo con la codificación.
=> Select decode(oficio, 'Presidente', 'P', 'Director', 'D', 'X')
from emp;

NVL(val, exp)

Devuelve la expresión exp si val es NULL, y val en otro caso.

=> select salario+nvl(comision,0) from emp;

Bibliografía

- <http://www.oracleya.com.ar/temarios/descripcion.php?inicio=0&cod=181&punto=23>
- <http://elbauldelprogramador.com/plsql-declaracion-de-variables/>
- <http://www.techonthenet.com/oracle>
- <http://beginner-sql-tutorial.com/es/oracle-functions.htm>