**MACHETE OFICIAL DE SQL**

**Tipos de datos**

En PL/SQL se pueden utilizar los mismos tipos de datos usados para la declaración de las columnas con la sentencia CREATE TABLE.

Para fines específicos de éste tutorial, sólo se especificarán los tipos de datos que más usaremos.

**Numéricos**

| Tipos de dato en Oracle | | |
| --- | --- | --- |
| **Tipo** | **Almacena** | **Rango** |
| PLS\_INTEGER | Entero con signo | -2,147,483,648  a   2,147,483,647 |
| NATURAL | Entero PLS\_INTEGER no negativo |  |
| NUMBER | Números de coma flotante. Se debe especificar la presición | 1e-130 a 1.0e126 |

**Texto**

| Tipos de dato para caracteres en Oracle | | |
| --- | --- | --- |
| **Tipo** | **Almacena** | **Rango** |
| CHAR | Cadena de texto de tamaño fijo. | 32,767 bytes |
| VARCHAR2 | Cadena de texto de tamaño variable. | 32,767 bytes |
| NCHAR | Cadena de texto de tamaño fijo. Incluye caracteres UNICODE. | 32,767 bytes |
| NVARCHAR2 | Cadena de texto de tamaño variable. Incluye caracteres UNICODE. | 32,767 bytes |

**Lógicos**

| Tipos de dato para valores lógicos en Oracle | | |
| --- | --- | --- |
| **Tipo** | **Almacena** | **Rango** |
| BOOLEAN | Almacena valores lógicos. Incluye el valor NULO | TRUE, FALSE o NULL |

**Fecha y hora**

| Tipos de dato para tiempo en Oracle | | |
| --- | --- | --- |
| **Tipo** | **Almacena** | **Rango** |
| DATE | Fechas fijas, incluye incluye año, mes, día, horas, minutos y segundos | 01/01/1,4712 BC  a  31/12/9999 |
| TIMESTAMP | Extiende el tipo DATE, incluye año, mes, día, horas, minutos y segundos fraccionados. |  |
| TIMESTAMP WITH TIME ZONE | Extiende el tipo de dato TIMESTAMP adicionando una zona horaria |  |

**Funciones string**

Las siguientes son algunas de las funciones que ofrece Oracle para trabajar con cadenas de caracteres:

**- chr(x):** retorna un caracter equivalente al código enviado como argumento "x". Ejemplo:

*select chr(65) from dual;-- retorna 'A'.*

*select chr(100) from dual;-- retorna 'd'.*

**- concat(cadena1,cadena2**): concatena dos cadenas de caracteres; es equivalente al operador ||. Ejemplo:

select concat('Buenas',' tardes') from dual;--retorna 'Buenas tardes'.

**- initcap(cadena):** retorna la cadena enviada como argumento con la primera letra (letra capital) de cada palabra en mayúscula. Ejemplo:

select initcap('buenas tardes alumno') from dual;--retorna 'Buenas Tardes Alumno'.

**- lower(cadena):** retorna la cadena enviada como argumento en minúsculas. "lower" significa reducir en inglés. Ejemplo:

select lower('Buenas tardes ALUMNO') from dual;--retorna "buenas tardes alumno".

**- upper(cadena):** retorna la cadena con todos los caracteres en mayúsculas. Ejemplo:

select upper('www.accenture.com') from dual;-- 'WWW.ACCENTURE.COM'

**- 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 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 izquierda. Ejemplos:

select lpad('alumno',10,'xyz') from dual;-- retorna 'xyzxalumno'

select lpad('alumno',4,'xyz') from dual;-- retorna 'alum'

**- 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'

**- trim(cadena):** retorna la cadena con los espacios de la izquierda y derecha eliminados. "Trim" significa recortar. Ejemplo:

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 (subcae1). Ejemplo:

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. Ejemplo:

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. "lenght" significa longitud en inglés. Ejemplo:

select length('www.oracle.com') from dual;-- devuelve 14.

**- 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. Ejemplos:

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 acracteres. 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'

**Funciones matematicas:**

**- abs(x):** retorna el valor absoluto del argumento "x". Ejemplo:

select abs(-20) from dual;--retorna 20.

La tabla dual es una tabla virtual que existe en todas las Bases de datos Oracle.

**- ceil(x):** redondea a entero, hacia arriba, el argumento "x". Ejemplo:

select ceil(12.34) from dual;--retorna 13.

**- floor(x):** redondea a entero, hacia abajo, el argumento "x". Ejemplo:

select floor(12.34) from dual; --12

**- mod(x,y):** devuelve el resto de la división x/y. Ejemplos:

select mod(10,3) from dual;--retorna 1.

select mod(10,2) from dual;--retorna 0.

**- power(x,y):** retorna el valor de "x" elevado a la "y" potencia. Ejemplo:

select power(2,3) from dual;--retorna 8.

**- 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". Ejemplos:

select round(123.456,2) from dual;-- retorna "123.46", es decir, redondea desde el segundo decimal.

select round(123.456,1) from dual;-- 123.5, es decir, redondea desde el primer decimal.

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. Ejemplo:

select sign(-120) from dual;--retorna -1

select sign(120) from dual;--retorna 1

**- 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. Ejemplo:

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 raiz cuadrada del valor enviado como argumento. Ejemplo:

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

**Funciones de fechas y horas:**

**- 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 tal cantidad de meses; si es negativo, se le resta a la fecha enviada tal cantidad de meses. Ejemplo:

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. Ejemplo:

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. Ejemplo:

select months\_between('19/05/2003','21/06/05') from dual;-- retorna

**- next\_day(fecha,dia**): retorna una fecha correspondiente al primer día especificado en "dia" luego 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;

**- current\_date:** retorna la fecha actual. Ejemplo:

select current\_date from dual;

**- current\_timestamp:** retorna la fecha actual

select current\_timestamp from dual;

Retorna: 10/08/07 09:59:44,109000000 AMERICA/BUENOS\_AIRES

**- sysdate**: retorna la fecha y hora actuales en el servidor de Oracle.

**-systimestamp:** retorna fecha y hora actuales.

select systimestamp from dual;

Retorna 10/08/07 10:33:48,984000000 -03:00

**- to\_date:** convierte una cadena a tipo de dato "date". Ejemplo:

select to\_date ('05-SEP-2007 10:00 AM','DD-MON-YYYY HH:MI AM') from dual;

Retorna 05/09/07

**- to\_char:** convierte una fecha a cadena de caracteres. Ejemplo:

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. Ejemplo:

select extract(month from sysdate) from dual;

retorna el número mes de la fecha actual.

En Oracle: Los operadores aritméticos "+" (más) y "-" (menos) pueden emplearse con fechas. Por ejemplos:

select sysdate-3:

Retorna 3 días antes de la fecha actual.

select to\_date('15/12/2007')-5 from dual;

Retorna 10/12/07

**Funciones de Grupos:**

Las relaciones entre las funciones de agrupamiento y los tipos de datos es la siguiente:

- count: se puede emplear con cualquier tipo de dato.

- min y max: con cualquier tipo de dato.

- sum y avg: sólo en campos de tipo numérico.

La función **"sum()"** retorna la suma de los valores que contiene el campo especificado. Si queremos saber la cantidad total de libros que tenemos disponibles para la venta, debemos sumar todos los valores del campo "cantidad":

select sum(cantidad)

from libros;

Para averiguar el valor máximo o mínimo de un campo usamos las funciones **"max()"** y **"min()"** respectivamente. Queremos saber cuál es el mayor precio de todos los libros:

select max(precio)

from libros;

Entonces, dentro del paréntesis de la función colocamos el nombre del campo del cuál queremos el máximo valor.

La función **"avg()"** retorna el valor promedio de los valores del campo especificado. Queremos saber el promedio del precio de los libros referentes a "PHP":

select avg(precio)

from libros

where titulo like '%PHP%';

Ahora podemos entender porque estas funciones se denominan "funciones de grupo", porque operan sobre conjuntos de registros, no con datos individuales.