Tema8. Consultas con funciones

Contenido

1 Consultas con funciones

Todos los SGBD implementan funciones para facilitar la creación de consultas complejas. Esas funciones dependen del SGBD que utilicemos. Todas las funciones devuelven un resultado que procede de un determinado cálculo.

La mayoría de funciones precisan que se les envíe datos de entrada (parámetros o argumentos) que son necesarios para realizar el cálculo de la función. Este resultado, lógicamente depende de los parámetros enviados. Dichos parámetros se pasan entre paréntesis. De tal manera que la forma de invocar a una función es:

```
nombreFunción[(parámetro1[, parámetro2,...])]
```

Si una función no precisa parámetros (como SYSDATE) no hace falta colocar los paréntesis.

En realidad hay dos tipos de funciones:

- Funciones que operan con datos de la misma fila
- Funciones que operan con datos de varias filas diferentes (funciones de agrupación).

En el siguiente apartado se tratan las funciones del primer tipo y más adelante se comentan las de agrupación.

2 Funciones numéricas

Redondeos

- **Round(n), Round(n, p)**: Redondea el número n al siguiente número más cercano con el número de decimales p indicado. La primera redondea sin decimales.
- **Trunc(n, p)**: Los decimales del número se cortan para que sólo aparezca el número de decimales p indicado. En MySql la función es **Truncate(n, p)**.

Matemáticas

- **MOD(n1, n2)**: Devuelve el resto resultado de dividir n1 entre n2.
- **POWER(valor, exponente)**: Eleva el valor al exponente indicado.
- **SQRT(n)**: Calcula la raíz cuadrada de n.
- **SIGN(n)**: Devuelve 1 si n es positivo, cero si vale cero y -1 si es negativo.
- **ABS(n):** Calcula el valor absoluto de n.
- **EXP(n):** Calcula e^n, es decir el exponente en base e del número n.
- **LN(n):** Logaritmo neperiano de n.
- **LOG(n):** Logaritmo en base 10 de n.
- **SIN(n):** Calcula el seno de n (n tiene que estar en radianes).
- **COS(n):** Calcula el coseno de n (n tiene que estar en radianes).
- **TAN(n):** Calcula la tangente de n (n tiene que estar en radianes).
- **ACOS(n):** Calcula el arco coseno de n (el resultado está en radianes).
- **ASIN(n):** Calcula el arco seno de n (el resultado está en radianes).

- **ATAN(n):** Calcula el arco tangente de n (el resultado está en radianes).
- **SINH(n):** Calcula el seno hiperbólico de n (n tiene que estar en radianes). Esta función no está en MySQL.
- **COSH(n):** Calcula el coseno hiperbólico de n (n tiene que estar en radianes). Esta función no está en MySQL.
- **TANH(n):** Calcula la tangente hiperbólica de n (n tiene que estar en radianes). Esta función no está en MySQL.

3 Funciones de grupos y de valores

Estas funciones, se verá cómo se utilizan en la siguiente unidad.

Función	Significado
COUNT(*)	Cuenta los elementos de un grupo. Se utiliza el asterisco para no tener que indicar un nombre de columna concreto, el resultado es el mismo para cualquier columna
SUM(*expresión*)	Suma los valores de la expresión
AVG(*expresión*)	Calcula la media aritmética sobre la expresión indicada
MIN(*expresión*)	Mínimo valor que toma la expresión indicada
MAX(*expresión*)	Máximo valor que toma la expresión indicada
STDDEV(*expresión*)	Calcula la desviación estándar
VARIANCE(*expresión*)	Calcula la varianza

4 Funciones de caracteres

Conversión de texto entre mayúsculas/minúsculas.

- **LOWER(texto)**: Convierte el texto a minúsculas (funciona con los caracteres españoles)
- **UPPER(texto):** Convierte el texto a minúsculas.
- **INITCAP(texto):** Coloca la primera letra de cada palabra en mayúsculas. No existe en MySQL.

Funciones de transformación

- **RTRIM(texto):** Elimina los espacios a la derecha del texto.
- **LTRIM(texto):** Elimina los espacios a la izquierda que posea el texto.
- **TRIM(texto):** Elimina los espacios en blanco a la izquierda y la derecha del texto y los espacios dobles del interior.
- **TRIM(caracteres FROM texto):** Elimina del texto los caracteres indicados. Por ejemplo **TRIM**('h' **FROM** nombre) elimina las haches de la columna nombre que estén a la izquierda y a la derecha.

- **SUBSTR(texto, n [,m]):** Obtiene los m siguientes caracteres del texto a partir de la posición n (si m no se indica se cogen desde n hasta el final). En MySQL la función es **SUBSTRING(texto, n [,m])**.
- **LENGTH(texto):** Obtiene el tamaño del texto.
- **INSTR(texto, textobuscado [,posinicial, [nAparción]]):** Obtiene la posición en la que se encuentra el texto buscado en el texto inicial. Se puede empezar a buscar a partir de una posición inicial concreta e incluso indicar el número de aparición del texto buscado. Ejemplo, si buscamos la letra a y ponemos 2 en nAparición, devuelve la posición de la segunda letra a del texto). Si no lo encuentra devuelve 0.
- **REPLACE(texto, textoABuscar [,textoReemplazo]):** Busca el texto a buscar en un determinado texto y lo cambia por el indicado como texto de reemplazo. Si no se indica texto de reemplazo, entonces esta función elimina el texto a buscar.
- **TRANSLATE(texto, caracteresACambiar, caracteresSustitutivos):** Potentísima función que permite transformar caracteres. Los caracteresACambiar son los caracteres que se van a cambiar, los caracteresSustitutivos son los caracteres que reemplazan a los anteriores. De tal modo que el primer carácter a cambiar se cambia por el primer carácter sustitutivo, el segundo por el segundo y así sucesivamente. Esta función no existe en MySQL.
- **LPAD(texto, anchuraMáxima, [caracterDeRelleno]) Y RPAD(texto, anchuraMáxima, [caracterDeRelleno]):** Rellena el texto a la izquierda (LPAD) o a la derecha (RPAD) con el carácter indicado para ocupar la anchura indicada. Si el texto es más grande que la anchura indicada, el texto se recorta. Si no se indica carácter de relleno se rellenará el espacio marcado con espacios en blanco.
- **REVERSE(texto):** Invierte el texto (le da la vuelta).

Otras funciones de caracteres

- **ASCII(carácter):** Devuelve el código ASCII del carácter indicado.
- **CHR(número):** Devuelve el carácter correspondiente al código ASCII indicado.

5 Funciones de trabajos con nulos

Permiten definir valores a utilizar en el caso de que las expresiones tomen el valor nulo.

- **NVL(valor, sustituto):** Si el valor es NULL, devuelve el valor sustituto; de otro modo, devuelve valor. En MySQL esta función es **IFNULL(valor, sustituto)**.
- **NVL2(valor, sustituto1, sustituto2):** Variante de la anterior, devuelve el valor sustituto1 si valor no es nulo. Si valor es nulo devuelve el sustituto2. Esta función no está en MySQL.
- **NULLIF(valor1, valor2):** Devuelve nulo si valor1 es igual a valor2. De otro modo devuelve valor1

6 Funciones de fechas y manejo de fechas e intervalos en Oracle

Las fechas se utilizan muchísimo en todas las bases de datos. Oracle proporciona dos tipos de datos para manejar fechas, los tipos **DATE** y **TIMESTAMP**. En el primer caso se almacena una fecha concreta (que incluso puede contener la hora hasta los segundos), en el segundo caso se

almacena un instante de tiempo más concreto que puede incluir incluso fracciones de segundo y la zona horaria.

Hay que tener en cuenta que a los valores de tipo fecha se les pueden sumar números y se entendería que esta suma es de días. Si tiene decimales entonces se suman días, horas, minutos y segundos. La diferencia entre dos fechas también obtiene un número de días.

6.1 Intervalos

Mientras que los tipos DATE y TIMESTAMP indican un momento específico de tiempo, INTERVAL almacena y permite trabajar con duraciones de tiempo, siendo posible definir intervalos de tiempo en términos de años y meses, o de días y segundos:

INTERVAL DAY TO SECOND sirve para representar días, horas, minutos y segundos;

INTERVAL YEAR TO MONTH representa años y meses.

Para los intervalos de año a mes los valores se pueden indicar de estas formas:

```
/* 123 años y seis meses */
INTERVAL '123-6' YEAR(4) TO MONTH
/* 123 años */
INTERVAL '123' YEAR(4) TO MONTH
/* 6 meses */
INTERVAL '6' MONTH(3) TO MONTH
```

La precisión en el caso de indicar tanto años como meses, se indica sólo en el año.

En intervalos de días a segundos los intervalos se pueden indicar como:

```
/* 4 días 10 horas 12 minutos y 7 con 352 segundos */
INTERVAL '4 10:12:7,352' DAY TO SECOND(3)
/* 4 días 10 horas 12 minutos */
INTERVAL '4 10:12' DAY TO MINUTE
/* 4 días 10 horas */
INTERVAL '4 10' DAY TO HOUR
/* 4 días*/
INTERVAL '4' DAY
/*10 horas*/
INTERVAL '10' HOUR
/*25 horas*/
INTERVAL '25' HOUR
/*12 minutos*/
INTERVAL '12' MINUTE
*30 segundos */
INTERVAL '30' SECOND
/*8 horas y 50 minutos */
INTERVAL '8:50' HOUR TO MINUTE
/*7 minutos 6 segundos*/
INTERVAL '7:06' MINUTE TO SECOND
/*8 horas 7 minutos 6 segundos*/
INTERVAL '8:07:06' HOUR TO SECOND
```

Esos **intervalos se pueden sumar** a valores de tipo **DATE** o **TIMESTAMP** para hacer cálculos. Gracias a ello se permiten sumar horas o minutos por ejemplo a los datos de tipo **TIMESTAMP**.

Para ciertas operaciones será necesario cambiar el formato de visualización de las fechas:

```
ALTER SESSION SET nls_date_format='DD/MM/YYYY HH24:MI:SS';
```

6.2 Obtener la fecha y hora actual

- **SYSDATE:** Obtiene la fecha y hora actuales.
- **SYSTIMESTAMP:** Obtiene la fecha y hora actuales en formato TIMESTAMP.

6.3 Calcular fechas

- **ADD_MONTHS(fecha, n):** Añade a la fecha el número de meses indicado por n
- **MONTHS_BETWEEN(fecha1, fecha2**): Obtiene la diferencia en meses entre las dos fechas (puede ser decimal)
- **NEXT_DAY(fecha, día):** Indica cual es el día que corresponde a añadir a la fecha el día indicado. El día puede ser el texto 'Lunes', 'Martes', 'Miércoles',... (si la configuración está en español) o el número de día de la semana (1=lunes, 2=martes,...).
- **LAST_DAY(fecha):** Obtiene el último día del mes al que pertenece la fecha. Devuelve un valor DATE
- **EXTRACT(valor FROM fecha):** Extrae un valor de una fecha concreta. El valor puede ser day (día), month (mes), year (año), etc.
- **GREATEST(fecha1, fecha2,..):** Devuelve la fecha más moderna de la lista
- **LEAST(fecha1, fecha2,..):** Devuelve la fecha más antigua de la lista
- **ROUND(fecha [,'formato**']): Redondea la fecha al valor de aplicar el formato a la fecha. El formato puede ser:
  - ✓ **'YEAR'** Hace que la fecha refleje el año completo
  - ✓ **'MONTH'** Hace que la fecha refleje el mes completo más cercano a la fecha
  - ✓ **'HH24'** Redondea la hora a las 00:00 más cercanas
  - ✓ **'DAY'** Redondea al día más cercano

```
Ejemplo:
SELECT ROUND (TO_DATE ('27-OCT-00'),'YEAR')"New Year"
FROM DUAL;

New Year
---------
01-JAN-01
```

- **TRUNC(fecha [formato**]): Igual que el anterior pero trunca la fecha en lugar de redondearla.

7 Funciones de conversión en Oracle

Oracle es capaz de convertir datos automáticamente a fin de que la expresión final tenga sentido. En ese sentido son fáciles las conversiones de texto a número y viceversa.

SELECT 5+'3' FROM DUAL /*El resultado es 8 */

SELECT 5 || '3' FROM DUAL /* El resultado es 53 */

También ocurre eso con la conversión de textos a fechas. De hecho es forma habitual de asignar fechas.

Pero en diversas ocasiones querremos realizar conversiones explícitas.

7.1 TO_CHAR

Obtiene un texto a partir de un número o una fecha. En especial se utiliza con fechas (ya que de número a texto se suele utilizar de forma implícita).

Fechas

En el caso de las fechas se indica el formato de conversión, que es una cadena que puede incluir estos símbolos (en una cadena de texto):

Símbolo	Significado
YY	Año en formato de dos cifras
YYYY	Año en formato de cuatro cifras
MM	Mes en formato de dos cifras
MON	Las tres primeras letras del mes
MONTH	Nombre completo del mes
DY	Día de la semana en tres letras
DAY	Día completo de la semana
D	Día de la semana (del 1 al 7)
DD	Día en formato de dos cifras
DDD	Día del año
Q	Semestre
WW	Semana del año
AM	Indicador AM
PM	Indicador PM
HH12	Hora de 1 a 12
HH24	Hora de 0 a 23
MI	Minutos (0 a 59)
SS	Segundos (0 a 59)
SSSS	Segundos desde medianoche
/ . ,:;'	Posición de los separadores, donde se pongan estos símbolos aparecerán en el resultado

```
SELECT TO_CHAR(SYSDATE, 'DD/MONTH/YYYY, DAY HH:MI:SS')
FROM DUAL ;
/* Sale : 17/AGOSTO /2015, LUNES 08:35:15, por ejemplo*/
```

Números

Para convertir números a textos se usa está función cuando se desean características especiales. En ese caso en el formato se pueden utilizar estos símbolos:

Símbolo	Significado
9	Posición del número
0	Posición del número (muestra ceros)
$	Formato dólar
L	Símbolo local de la moneda
S	Hace que aparezca el símbolo del signo
D	Posición del símbolo decimal (en español, la coma)
G	Posición del separador de grupo (en español el punto)

TO_CHAR (número, 'formato'). Convierte un número de tipo NUMBER a tipo VARCHAR2 en el formato especificado. Los distintos formatos numéricos son los siguientes:

TABLA DE FORMATOS NUMÉRICOS		
9	999	Devuelve el valor con el número especificado. Si es positivo, deja un espacio, si es negativo escribe el signo -, dejando a blancos los ceros no significativos, excepto si el valor es 0.
0	9990	Muestra un 0 si el valor es 0.
0	0999	Devuelve el valor dejando ceros al principio.
$	$9999	Devuelve el valor con el signo $ a la izquierda.
B	B999	Muestra un espacio en blanco si el valor es 0. Es el formato por omisión.
MI	999MI	Si el número es negativo, el signo menos sigue al número. Por omisión, el signo se pone a la izquierda.
S	S999 999S	Escribe siempre el signo, en la posición indicada.
D	99D99	Devuelve el carácter decimal en la posición especificada.
G	9G999	Devuelve el carácter de los miles en la posición especificada.
L	L999	Devuelve el símbolo de la moneda local en la posición indicada.
,(COMA)	9,999	Devuelve la coma en la posición especificada.
.(PUNTO)	99.99	Devuelve el punto decimal en la posición especificada.

TABLA DE FORMATOS NUMÉRICOS		
EEEE	9.9EEEE	Devuelve el valor usando la notación científica.
RN	RN	Devuelve el valor en números romanos.
FM	FM90.9	Devuelve el valor alineado a la izquierda.

7.2 TO_NUMBER

Convierte textos en números. Se indica el formato de la conversión (utilizando los mismos símbolos que los comentados anteriormente).

7.3 TO_DATE

Convierte textos en fechas. Como segundo parámetro se utilizan los códigos de formato de fechas comentados anteriormente.

7.4 CAST

Función muy versátil que permite convertir el resultado a un tipo concreto.

CAST(expresión **AS** tipoDatos)

Ejemplo:

```
SELECT CAST(2.34567 AS NUMBER(7,6))
FROM DUAL;
```

Lo interesante es que puede convertir de un tipo a otro. Por ejemplo imaginemos que tenemos una columna en una tabla mal planteada en la que el precio de las cosas se ha escrito en Euros. Los datos son (se muestra sólo la columna precio):

precio
25.2 €
2.8 €
123.65 €
.78 €
.123 €
20 €

Imaginemos que queremos doblar el precio, no podremos porque la columna es de tipo texto, por ello debemos tomar sólo la parte numérica y convertirla a número, después podremos mostrar los precios multiplicados por dos:

```
SELECT 2 * CAST(SUBSTR(precio,1,INSTR(precio,'€')-2) AS NUMBER)
FROM precios;
```

La combinación de SUBSTR e INSTR es para obtener sólo los números. Incluso es posible que haya que utilizar REPLACE para cambiar los puntos por comas (para utilizar el separador decimal del idioma español).

8 DECODE

Función que permite realizar condiciones en una consulta. Hace las veces de sentencias CASE o IF-THEN-ELSE, para facilitar consultas condicionales. Se evalúa una expresión y se colocan a continuación pares valor, resultado de forma que si la expresión equivale al valor, se obtiene el resultado indicado. Se puede indicar un último parámetro con el resultado a efectuar en caso de no encontrar ninguno de los valores indicados. Si se omite el valor por defecto, se devolverá un valor nulo en el caso de que la búsqueda no coincida con ninguno de los valores resultantes.

Formato:

```
DECODE (columna|expression, valor_buscado1, resultado1
                            [,valor_buscado2, resultado2,…..,]
                            [, defecto])
```

Ejemplo:

```
SELECT DECODE(cotizacion, 1, salario*0.85,
              2,salario * 0.93,
              3,salario * 0.96,
              salario)
FROM empleados;
```

En el ejemplo dependiendo de la cotización se muestra rebajado el salario: un 15% si la cotización es uno, un 7 si es dos y un 4 si es tres. Si la cotización no es ni uno ni dos ni tres, sencillamente se muestra el salario sin más.