

Trabajo con datos

**Funciones DML**



## Funciones DML

Las funciones DML se establecen para realizar operaciones requeridas dentro de las bases de datos; estas operaciones van desde simples procesos de cálculos de operaciones hasta realizar complejos procesos para el mantenimiento de los datos en la base de datos, aunque todos estos procesos se llevan a cabo de acuerdo con necesidades técnicas que el usuario de la aplicación requiera. Dichos procesos pueden estar relacionados con procesos que afectan distintas partes del sistema y, por eso se hace necesario, en algunas ocasiones, establecer un orden específico de ejecución de los mismos para garantizar un correcto funcionamiento en el sistema mismo.

### 1. Funciones de cadena

Las funciones de cadena trabajan con campos CHAR y VARCHAR por lo que los literales que escribamos se deben encerrar entre comillas simples.

Estas funciones que vamos a explicar a continuación pueden manipular cadenas de letras u otros caracteres por lo que las vamos a dividir en dos grupos:

#### Funciones que devuelven caracteres

Función	Utilización
<b>CHR(n)</b>	Nos devuelve el carácter cuyo valor en binario es <b>n</b> .
<b>CONCAT(cad1, cad2)</b>	Nos devuelve <b>cad1</b> concatenada con <b>cad2</b> .
<b>UPPER(cad)</b>	Convierte <b>cad</b> a mayúsculas.
<b>LOWER(cad)</b>	Convierte <b>cad</b> a minúsculas.
<b>LPAD(cad1,n[,cad2])</b>	Con esta función, añadimos caracteres a <b>cad1</b> por la izquierda, hasta una longitud máxima dada por <b>n</b> .
<b>INITCAP(cad)</b>	Convierte la primera letra de <b>cad</b> a mayúscula.
<b>LTRIM(cad [,set])</b>	Elimina un conjunto de caracteres a la izquierda de <b>cad</b> , siendo <b>set</b> el conjunto de caracteres a eliminar.



<b><u>RPAD</u></b> (cad1, n[,cad2])	Con esta función, añadimos caracteres de la misma forma que con la función <b>LPAD</b> , pero esta vez los añadimos a la derecha.
<b><u>RTRIM</u></b> (cad[,set])	Hace lo mismo que <b>LTRIM</b> , pero por la derecha.
<b><u>REPLACE</u></b> (cad,cadena_buscada[,cadena_sustitucion] )	Sustituye un conjunto de caracteres de <b>0</b> o más caracteres, devuelve <b>cad</b> con cada ocurrencia de <b>cadena_buscada</b> , sustituida por <b>cadena_sustitucion</b> .
<b><u>SUBSTR</u></b> (cad, m[,n])	Devuelve la <b>subcadena</b> de <b>cad</b> que abarca desde <b>m</b> hasta la cantidad de caracteres dados por <b>n</b> .
<b><u>TRANSLATE</u></b> (cad1,cad2,cad3)	Convierte los caracteres de una cadena en caracteres diferentes. Devuelve <b>cad1</b> con los caracteres encontrados en <b>cad2</b> y sustituidos por los caracteres de <b>cad3</b> .

Algunos ejemplos aplicados a este proceso, serían los siguientes:

Sentencia SQL que obtiene el nombre de los alumnos sacando por pantalla la siguiente frase: el nombre del alumno es (nombre que está almacenado en la tabla).

```
select CONCAT ('el nombre de alumno es', nombre) from alumno;
```

Sentencia SQL que me devuelve los nombres de los alumnos en mayúsculas.

```
select UPPER(nombre) from alumno;
```

Sentencia SQL que obtiene de un campo nombre, las 3 primeras letras.

```
select SUBSTR(nombre,0,3) from alumno;
```

## 2. Funciones matemáticas

Dentro de las funciones matemáticas encontramos varias instrucciones, las cuales hacen posible que se puedan calcular y realizar operaciones dentro de los procesos de consulta de información; es decir, la base de datos se encarga, en este caso, de usar funciones matemáticas para realizar y presentar el resultado de las operaciones que el sistema requiere. Para esto, se debe tener en cuenta muy bien el tipo de dato que se tiene almacenado en ese campo, porque, por ejemplo, los campos que son caracteres no se pueden utilizar para realizar dichas operaciones; esto solo se deja para las operaciones con campos numéricos. A continuación, veremos algunos ejemplos sobre esta temática.

### Campos calculados

Son operaciones que se pueden realizar sobre las columnas resultantes de una consulta; las operaciones que se pueden realizar son:

- Concatenar dos o más campos.

- Realizar operaciones matemáticas.

```
SELECT (campo1-numerico * campo2-numerico) as "total" FROM nombre-tabla;
```



## Función COUNT

Devuelve el número de valores de una columna especificada. Con el objetivo de tener el total de elementos de una columna, se puede ocupar esta función de la siguiente manera:

```
SELECT COUNT (IdLibro) AS NoLibros FROM Libro WHERE Cod_Ed_=100006
```

También se puede utilizar COUNT(\*) en la sentencia SELECT, pero debe tenerse especial cuidado cuando se utilice, pues a diferencia que la del ejemplo, esta forma cuenta todas las filas de una tabla, independientemente de si existen valores nulos o valores duplicados.

## Función SUM

Devuelve la suma de los valores contenidos en una columna especificada. Debe tenerse presente que esta función solo puede emplearse con campos numéricos.

En esta consulta, únicamente se devolverán dos columnas; en la primera, la cantidad de libros en el que su precio es igual, o sobrepasa los \$65 y, en la segunda, la suma total de los precios de esos libros.

```
SELECT COUNT (IdLibro) AS NoLibros, SUM(Precio) AS TotalPrecio FROM Libros WHERE Precio <= 65
```

## Función MAX, MIN, AVG

La función MAX devuelve el valor máximo contenido en una columna especificada.

La función MIN devuelve el valor mínimo contenido en una columna especificada.

La función AVG devuelve la media de los valores contenidos en una columna especificada.

Las primeras dos funciones pueden emplearse, tanto en campos numéricos como no numéricos, a excepción de la tercera, que solo puede ocuparse en campos numéricos.

El resultado de la consulta mostraría tres columnas en las que se podría observar el valor mínimo de toda la columna Precio y el valor máximo, así como un promedio global del precio de los libros.

```
SELECT MIN(Precio) AS PrecioMin, MAX(Precio) AS PrecioMax, AVG(Precio) As Promedio FROM Libros
```

Como se puede apreciar, las funciones matemáticas permiten realizar operaciones para calcular datos para ser mostrados al usuario, pero esto es posible siempre y cuando el valor que se utilice tenga la característica de ser numérico, es decir, todas estas operaciones se realizan y aplican para campos que contengan valores numéricos, tales como: INT, DOUBLE, FLOAT.

## 3 Funciones fecha/hora

Las funciones de fecha y hora son aplicables para realizar operaciones, teniendo en cuenta que estas se encuentran divididas en varias partes; una de ellas es día, mes y año, teniendo en cuenta que este orden puede variar de acuerdo al idioma utilizado en el sistema operativo y, adicional a ello, de acuerdo a necesidades planteadas por el usuario, que se deben tener en cuenta como factores determinantes en el correcto funcionamiento del sistema como tal. Ahora bien, estas funciones no solo son aplicables en los procesos de recuperación y análisis de información, sino que también permiten establecer, de una manera sencilla, la proyección que puede llegar a tener el sistema en el tiempo; un ejemplo de esto es un software de historia, el cual debe indicar según configuración y política del centro asistencial, cuando debe ser la cita del paciente; si se desea que el sistema realice este proceso de manera automática, se deberá requerir una función de manejo de fechas y horas, para poderlos realizar; a continuación, veremos una tabla que nos permite visualizar dichos comandos y su utilización.



### **add\_months(date,n)**

Devuelve la fecha que corresponde a date más la cantidad de meses, indicada por el entero n. Si la fecha resultante tuviera más días que los disponibles en el mes resultante, el resultado sería el último día de ese mes. De lo contrario, la fecha resultante tiene el mismo componente de día, que date. Por ejemplo:

```
add_months('March 03, 2015',2) devuelve '2015-05-03 00:00:00'
add_months('March 31, 2015',-1) devuelve '2015-02-28 00:00:00'
```

### **ge(ts,ts) age(ts)**

Devuelve el intervalo entre dos indicaciones de fecha y hora. El valor devuelto tiene la misma precisión que el argumento más preciso. Por ejemplo:

```
select age('10-22-2003', '7-6-2002') devuelve '1 year 3 mons 16 days'
select age('10-22-2003 09:46:07.325', '7-6-2002') devuelve '1 year 3 mons 16 days 09:46:07.325'
```

### **date\_part(units,col)**

Extrae el subcampo representado por units del valor de fecha y hora, el intervalo o la duración que se ha especificado para col. Esta función equivale a extract().

### **date\_trunc(units,date)**

Trunca la fecha especificada para date a la precisión especificada por units.

### **extract(units FROM col)**

Extrae el subcampo representado por units del valor de fecha y hora, el intervalo o la duración que se ha especificado para col. Esta función equivale a date\_part().

### **last\_day(date)**

Devuelve el último día del mes de la fecha especificada.

Devuelve el número de meses entre las fechas d1 y d2:

Si d1 es posterior a d2, el resultado es positivo.  
Si d1 es anterior a d2, el resultado es negativo.

Si d1 y d2 son los mismos días del mes o ambos son los últimos días de sus respectivos meses, el resultado es un entero.

En caso contrario, el resultado incluye una parte fraccional basada en un mes de 31 días.

Por ejemplo:

```
select months_between('2015-05-23', '2015-04-23') devuelve 1
select months_between('2015-02-28', '2015-05-31') devuelve -3
select months_between('2015-05-15', '2014-05-05') devuelve 12.322580645 (12 meses completos
más 10 días adicionales; 10/31=0.322580645).
```



### GETDATE()

Esta función devuelve la fecha que se encuentra en el servidor de bases de datos, en ese momento.

### Current\_timestamp

Esta función devuelve la hora que se encuentra configurada en el motor de la base de datos y es utilizada cuando se requiere que el dato sea preciso en cuanto al registro del sistema; ejemplo, ingreso al sistema o salida del sistema.

## 4 Funciones del sistema

Las funciones del sistema están determinadas por los requerimientos que se establezcan en el sistema; esto está estrechamente conectado con el análisis del sistema de almacenamiento, es decir, aquí se establecen todas las funciones que este debe realizar. Si colocamos un ejemplo sobre estos procesos en un sistema de inventario, tendríamos un ejemplo de funciones tal y como se establece en la siguiente tabla:

Código	Especificación de la función
<b>FUN1</b>	El sistema debe verificar que cuando se realice una venta, no exceda el límite establecido de inventario.
<b>FUN2</b>	El sistema debe restar de las existencias, del producto, las cantidades vendidas.
<b>FUN3</b>	El sistema debe sumar la cantidad de unidades adicionales a las existencias del producto, de acuerdo con las compras establecidas.

Como se puede observar en la tabla anterior, estos son ejemplos de funciones del sistema que se deben realizar a través del lenguaje DML, como estudiamos anteriormente; este permite realizar operaciones matemáticas; por lo tanto, a través de estas funciones establecidas, se debe garantizar que el sistema realice las operaciones requeridas y también verificar que las realice de manera correcta; todo este proceso se lleva a cabo, realizando escenarios de prueba para garantizar que los resultados ofrecidos por el sistema, sean acordes con lo esperado.

