

Principales funciones en SQL

 diego.com.es/principales-funciones-en-sql

Contenido modificable

Si ves errores o quieres modificar/añadir contenidos, puedes [crear un pull request](#).
Gracias

El **lenguaje SQL** tiene **funciones incorporadas para hacer cálculos sobre los datos**. Las funciones se pueden dividir en dos grupos (existen muchas mas, que dependen del **sistema de bases de datos** que se utilice):

Funciones agregadas SQL, devuelven un sólo valor, calculado con los valores de una columna.

1. AVG() - La **media de los valores**
2. COUNT() - El **número de filas**
3. MAX() - El **valor más grande**
4. MIN() - El **valor más pequeño**
5. SUM() - La **suma de los valores**
6. GROUP BY - Es una **sentencia** que va muy ligada a las funciones agregadas

Funciones escalares SQL, devuelven un sólo valor basándose en el valor de entrada.

1. UCASE() - Convierte un campo a **mayúsculas**
2. LCASE() - Convierte un campo a **minúsculas**
3. MID() - **Extrae** caracteres de un campo de texto
4. LEN() - Devuelve la **longitud** de un campo de texto
5. NOW() - Devuelve la **hora y fecha** actuales del sistema
6. FORMAT() - Da formato a un **formato** para mostrarlo

Para la explicación de las funciones, una de las tablas con las que vamos a trabajar es **Productos**:

ProductID	NombreProducto	Descripcion	Precio	Stock
1	Camiseta	Camiseta negra simple de talla única	10	16
2	Pantalón	Pantalón largo azul tipo chino	20	24
3	Gorra	Gorra azul con el logo de los Yankees	15	32
4	Zapatillas	Zapatillas de running de color blanco y verde	35	13

Funciones agregadas SQL

1. AVG()

La función **AVG()** devuelve la **media de valores de una columna numérica**.

```
SELECT AVG (nombreColumna) FROM nombreTabla;
```

La siguiente sentencia SQL devuelve la media del stock que hay en el almacén:

```
SELECT AVG (Stock) FROM Productos;
```

La siguiente sentencia muestra el NombreProducto y el Precio de los registros que tienen un Precio por encima de la media:

```
SELECT NombreProducto, Precio FROM Productos  
WHERE Precio > (SELECT AVG (Precio) FROM Productos);
```

2. COUNT()

La función **COUNT()** devuelve el **número de filas** que cumplen con un determinado criterio:

Número de valores en una columna (los valores NULL no se cuentan):

```
SELECT COUNT (nombreColumna) FROM nombreTabla;
```

Número de registros en una tabla

```
SELECT COUNT(*) FROM nombreTabla;
```

Número de valores distintos de una columna

```
SELECT COUNT (DISTINCT nombreColumna) FROM nombreTabla;
```

Vamos a emplear ahora la tabla **Pedidos**:

PedidoID	ClienteID	Factura
234	4	160
235	5	48
236	12	64
237	4	92

Si queremos conocer el **número de pedidos del cliente número 4**:

```
SELECT COUNT (ClienteID) AS PedidosCliente4 FROM Pedidos  
WHERE ClienteID=4;
```

Si queremos simplemente contar el **número de pedidos totales**:

```
SELECT COUNT(*) AS PedidosTotales FROM Pedidos;
```

Si queremos contar el **número de clientes únicos** desde la tabla Pedidos:

```
SELECT COUNT (DISTINCT ClienteID) AS NumeroClientes FROM Pedidos;
```

3. MAX()

La función **MAX()** devuelve el **mayor valor de la columna seleccionada**:

```
SELECT MAX (nombreColumna) FROM nombreTabla;
```

De la tabla productos vamos a coger **el producto más caro**:

```
SELECT MAX (Precio) AS ProductoMasCaro FROM Productos;
```

4. MIN()

La función **MIN()** devuelve el **menor valor de la columna seleccionada**:

```
SELECT MIN (nombreColumna) FROM nombreTabla;
```

De la tabla productos vamos a coger **el producto más barato**:

```
SELECT MIN (Precio) AS ProductoMasBarato FROM Productos;
```

5. SUM()

La función **SUM()** devuelve la **suma de una columna numérica**:

```
SELECT SUM (nombreColumna) FROM nombreTabla;
```

De la tabla productos vamos a coger el **número total de productos en stock**:

```
SELECT SUM (Stock) AS ProductosTotales FROM Productos;
```

6. GROUP BY

La sentencia **GROUP BY** se utiliza junto con las funciones agregadas para **agrupar en un result-set una o más columnas**.

```
SELECT nombreColumna, funcion_agregada(nombreColumna)
FROM nombreTabla
WHERE nombreColumna operador valor
GROUP BY nombreColumna;
```

Si tenemos las tablas **Productos, Pedidos y Clientes**:

Productos:

ProductoID	NombreProducto	Descripcion	Precio	Stock
1	Camiseta	Camiseta negra simple de talla única	10	16
2	Pantalón	Pantalón largo azul tipo chino	20	24
3	Gorra	Gorra azul con el logo de los Yankees	15	32
4	Zapatillas	Zapatillas de running de color blanco y verde	35	13

Pedidos:

PedidoID	ClienteID	Factura
234	4	160
235	5	48
236	12	64
237	4	92

Clientes:

ClienteID	NombreCliente	Contacto
1	Lorena Higgins	456443552
2	Raúl González	445332221
3	Carmen Smith	488982635
4	Fernando Stewart	412436773

Vamos a obtener el **número de pedidos** realizados **por cada cliente**:

```
SELECT Clientes.NombreCliente, Count(Pedidos.PedidoID) AS NumeroPedidos FROM Pedidos
LEFT JOIN Clientes
ON Pedidos.ClienteID=Clientes.ClienteID
GROUP BY NombreCliente;
```

Podemos **utilizar GROUP BY en más de una columna**:

```
SELECT Clientes.NombreCliente, Productos.NombreProducto, COUNT (Pedidos.PedidoID) AS
NumeroPedidos
FROM ((Pedidos
INNER JOIN Clientes ON Pedidos.ClienteID=Clientes.ClienteID)
INNER JOIN Productos ON Pedidos.ProductoID=Productos.ProductoID)
GROUP BY NombreCliente, NombreProducto;
```

Funciones escalares SQL

7. UCASE()

La función **UCASE()** convierte el **valor de un campo a mayúsculas**.

```
SELECT UCASE (nombreColumna) FROM nombreTabla;
```

Si queremos obtener todos los **nombres de los clientes en mayúsculas**:

```
SELECT UCASE (NombreCliente) AS Cliente FROM Clientes;
```

8. LCASE()

La función **LCASE()** convierte el **valor de un campo en minúsculas**:

```
SELECT LCASE (nombreColumna) FROM nombreTabla;
```

Si queremos obtener todos los **nombres de los clientes en minúsculas**:

```
SELECT LCASE (NombreCliente) FROM Clientes;
```

9. MID()

La función **MID()** se usa para **extraer caracteres desde un campo de texto**.

```
SELECT MID (nombreColumna, inicio [, longitud]) AS unAlias FROM nombreTabla;
```

Los campos obligatorios son **nombreColumna** e **inicio** (especifica la posición donde comenzar la extracción). El campo opcional es **longitud**, que especifica el número de caracteres a devolver. Si se omite, devuelve todos los caracteres hasta el final del texto.

Vamos a extraer por ejemplo los **primeros cuatro caracteres** de **NombreProducto** de la tabla **Productos**:

```
SELECT MID (nombreProducto, 1, 4) AS ProductoCorto
From Productos;
```

10. LEN()

La función **LEN()** devuelve la longitud de un campo de texto.

```
SELECT LEN (nombreColumna) FROM nombreTabla;
```

Si queremos obtener **nombreProducto** y la **longitud de las descripciones** de los productos de la tabla **Productos**:

```
SELECT NombreProducto, LEN(Descripcion) as LongitudDescripcion  
FROM Productos;
```

11. NOW()

La función **NOW()** devuelve la hora y fecha actuales.

```
SELECT NOW() FROM nombreTabla;
```

Si queremos devolver el **nombreProducto** y **Precio** de hoy de la tabla **Productos**:

```
SELECT NombreProducto, Precio, Now() AS PrecioProductosHoy FROM Productos;
```

12. FORMAT()

La función **FORMAT()** se usa para formatear cómo se mostrará un campo.

```
SELECT FORMAT(nombreColumna, formato) FROM nombreTabla;
```

Vamos a mostrar un **formato de fecha del ejemplo anterior**:

```
SELECT NombreProducto, Precio, FORMAT(Now(), 'YYYY-MM-DD') AS PrecioProductosHoy FROM  
Productos;
```