

Bootcamp Java Developer

Fase 1 - Java Analyst
Módulo 4



Funciones de agregado y de agrupamiento

Funciones de agregado / agrupamiento

Existen funciones en SQL que nos permiten contar registros, calcular sumas, promedios, obtener valores máximos y mínimos.

Estas funciones se denominan **funciones de agrupamiento** porque operan sobre conjuntos de registros, en lugar de hacerlo sobre datos individuales. Tienen la característica de **agrupar los resultados en un solo registro de salida**.



Función **COUNT()**

Esta función retorna la **cantidad de valores** que contiene un campo especificado.

Por ejemplo, si se quiere saber la cantidad de productos que hay en la tabla **Productos**, la sentencia SQL a utilizar será la que se muestra a la derecha:

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

¡Recuerda!

SELECT COUNT(*) FROM Productos;	Correcto
SELECT COUNT (*) FROM Productos;	Incorrecto

Y la sentencia de la diapositiva anterior, puede combinarse con la cláusula **WHERE**. Por ej., si se quiere saber cuántos productos contienen la palabra *iPhone* en el campo **Nombre** de la tabla **Productos**, la sentencia sería:

```
SELECT COUNT(*) FROM Productos WHERE Nombre LIKE "%iPhone%";
```



Función *SUM()*

Esta función retorna **la suma de los valores** que contiene el campo especificado.

Por ejemplo, si se quiere saber el stock de productos que hay en la tabla ***Productos***, la sentencia a ejecutar será:

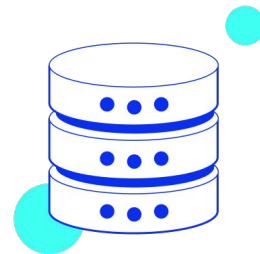
```
SELECT SUM(Stock) FROM Productos;
```

Función *MIN()*

Esta función permite calcular el **valor mínimo** de un campo.

Por ejemplo, para conocer cuál es el **menor precio** que figura en la tabla ***Productos***, la instrucción SQL a ejecutar será la siguiente:

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

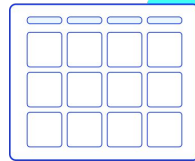


Función MAX()

Permite averiguar el **valor máximo** de un campo.

Por ejemplo, para conocer cuál es el **mayor precio** de todos los **Productos**, la instrucción SQL a ejecutar será:

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

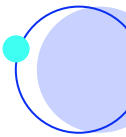


Función AVG()

Esta función retorna el **valor promedio** de los valores del campo especificado.

Por ejemplo, para calcular el **precio promedio** de todos los productos contenidos en la tabla ***Productos***, la instrucción SQL a ejecutar será:

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



Cláusula **GROUP BY**

La **agrupación** es un concepto básico de Bases de Datos. La cláusula **GROUP BY**, como su traducción lo indica (**agrupar por**) tiene como propósito **agrupar información** de acuerdo a un criterio en común.

- Por lo general se utiliza **con funciones de agrupación o de agregación** (**COUNT, MIN, MAX, AVG, SUM**).
- El comportamiento de la cláusula **GROUP BY** dependerá de la función de agrupación que se esté utilizando.



Mecanismo de uso

Ejemplo:

ID	Nombre	Precio	Marca	Categoría	Presentación	Stock	Disponible
1	iPhone 6	499.99	Apple	Smartphone	16GB	500	SI
2	iPad Pro	599.99	Apple	Smartphone	128GB	300	SI
3	Nexus 7	299.99	LG	Smartphone	32GB	250	NO
4	Galaxy S7	459.99	Samsung	Smartphone	64GB	200	SI
5	Impresora T23	489.99	Epson	Impresoras	Color	180	NO
6	Impresora T33	399	Epson	Impresoras	Color	200	NO
7	Lavarropa 7000	1679	LG	Lavarropas	Automático	100	SI
8	Camara Digital 760	649	Kodak	Fotografía	Sin detalle	150	NO
9	Notebook CQ40-300	2999	HP	Notebooks	Intel Core i3	100	SI

Suponiendo que se quiere calcular la cantidad de productos existentes en la categoría **Smartphone**, entonces habrá que agrupar los registros por el campo **Categoría**. La cláusula **GROUP BY** permite hacer esto de manera automática a partir de un valor o dato común:

```
SELECT Categoria, SUM(Stock) FROM Productos GROUP BY Categoria;
```



Cláusula *HAVING*

La cláusula ***HAVING*** permite **hacer selecciones (filtrar) en situaciones en las que no es posible usar la cláusula *WHERE***, dado que se establece un criterio sobre un valor dado por una **función de agrupamiento** y no por valores de registros.

Ejemplo:

```
SELECT Categoria, SUM(Stock) FROM Productos GROUP BY Categoria  
HAVING SUM(Stock) > 250;
```

**¡Sigamos
trabajando!**