

## SQL TEMA 7

### REPASEMOS LAS CLAUSULAS

Cláusula	Descripción
FROM	Especifica la tabla de la que se quieren obtener los registros
WHERE	Especifica las condiciones o criterios de los registros seleccionados
GROUP BY	Para agrupar los registros seleccionados en función de un campo
HAVING	Especifica las condiciones o criterios que deben cumplir los grupos
ORDER BY	Ordena los registros seleccionados en función de un campo

### REPASEMOS LOS OPERADORES: DE COMPARACION

Operador	Significado
<	Menor que
>	Mayor que
=	Igual que
>=	Mayor o igual que
<=	Menor o igual que
<>	Distinto que
BETWEEN	Entre. Utilizado para especificar rangos de valores
LIKE	Cómo. Utilizado con caracteres comodín (? *)
In	En. Para especificar registros en un campo en concreto

Los más conocidos, los 5 primeros: muy utilizados a la hora de establecer criterios Between. Operador de comparación para establecer rangos de valores. Ej rangos de fechas, rangos de precios.....

LIKE utilizado con los caracteres comodín

IN para especificar una serie de registros en un campo en concreto

### LOGICOS

Operador	Significado
AND	Y lógico
OR	O lógico
NOT	Negación lógica

## ORDEN DE ESCRITURA DE LAS CLAUSULAS



EJEMPLO PARA VER TRES CAMPOS DE LA TABLA PRODUCTOS DE SECCION CERAMICA

```
1 SELECT NOMBREARTÍCULO, SECCIÓN, PRECIO FROM PRODUCTOS WHERE SECCIÓN="CERÁMICA"|
```

## CLAUSULA ORDER BY

Clausula que ordena los registros seleccionados, en función de un campo

Ejemplo:

Vamos a hacer una selección de dos productos de sección: deportes y cerámica

```
SELECT * FROM PRODUCTOS WHERE SECCIÓN='DEPORTES' OR SECCIÓN='CERÁMICA'
```

Vamos a ordenar ahora todos de forma alfabética, por lo que si antes salían deportes primer, ahora sale cerámica

Y lo hacemos así:

```
PRODUCTOS WHERE SECCIÓN='DEPORTES' OR SECCIÓN='CERÁMICA' ORDER BY SECCIÓN
```

si simplemente añadimos DESC, sería ordenación inversa (de mayor a menor)

Probemos ahora a hacer ordenar por precios (y se desordena el resto)

¿Y si queremos ordenar por dos criterios?

En el ejemplo por, sección y por precio

y si encima queremos un orden uno ascendente otro descendente

```
SELECT * FROM PRODUCTOS WHERE SECCIÓN='DEPORTES' OR SECCIÓN='CERÁMICA' ORDER BY SECCIÓN, PRECIO DESC
```

incluso un tercer criterio de ordenación:

```
= 'CERÁMICA' ORDER BY SECCIÓN, PAÍSDEORIGEN, PRECIO
```

## CONSULTAS DE AGRUPACION

También conocidas como consultas de totales

Son consultas en que se hacen cálculos por grupos:

- Cogemos los registros de una tabla y los agrupamos en base a un registro o campo
- Una vez que los tenemos agrupados, con estos registros realizamos un cálculo: un conteo, una media,

## Funciones de agregado

Función	Descripción
→ AVG	Calcula el promedio de un campo
→ COUNT	Cuenta los registros de un campo
→ SUM	Suma los valores de un campo
→ MAX	Devuelve el máximo de un campo
→ MIN	Devuelve el mínimo de un campo

Vamos a hacer una primera consulta de agrupación que me diga cuanto suman los productos de ls secciones

1. Creamos un campo de agrupación y un campo del cálculo.
  - como campo de agrupación utilizamos el campo sección
  - Campo de cálculo : PRECIO

```
SELECT SECCIÓN, SUM(PRECIO) FROM PRODUCTOS GROUP BY SECCIÓN
```

2. Vamos ahora a ordenar por precio .  
Si intentamos hacerlo por ORDER BY PRECIO, dará error porque no hay ningún campo con ese nombre ( en la consulta, en la tabla si)
- 3.
4. Esto lo sulucionamos con lo que se conoce como ALIAS

```
)) AS SUMA_ARTICULOS FROM PRODUCTOS GROUP BY SECCIÓN ORDER BY SUMA_ARTICULOS
```

```
SELECT SECCIÓN, AVG(PRECIO) AS MEDIA_ARTICULOS FROM PRODUCTOS GROUP BY SECCIÓN HAVING SECCIÓN='DEPORTES' OR SECCIÓN='CONFECCIÓN'
```

Vamoa a ver ahora el caso en que si queremos sacar la media solo de cerámica y deportes. Utilizamos el Where En estos casos de consultas agrupadas es HAVING

```
IA_ARTICULOS FROM PRODUCTOS GROUP BY SECCIÓN WHERE
```

HAVING|

Ahora en la tabla clientes quiero que nos digan cuantos clientes hay de cada población. LO hacemos con la función COUNT()

```
POBLACIÓN, COUNT(CÓDIGOCLIENTE) AS N_CLIENTES FROM CLIENTES GROUP BY POBLACIÓN]
```

- Ojo!! No cuenta los registros en blanco. Por eso para más seguridad se utiliza un campo clave
- 

Ahora queremos saber el precio del artículo más caro de la sección de confección:

```
1 | SELECT SECCIÓN, MAX(PRECIO) AS PRECIO_MAS_ALTO FROM PRODUCTOS WHERE SECCIÓN='CONFECCIÓN' GROUP BY SECCIÓN
```

Y por último queremos saber la media de el precio de los artículos por secciones y que lo ordene luego por precio

```
SELECT SECCIÓN, AVG(PRECIO) AS MEDIA_ARTICULOS FROM PRODUCTOS GROUP BY SECCIÓN HAVING SECCIÓN='DEPORTES' OR SECCIÓN='CONFECCIÓN' ORDER BY MEDIA_ARTICULOS
```