



// Funciones y cláusulas

IT BOARDING

BOOTCAMP



Índice



01 Consultas SELECT y cláusulas

02 Funciones de Agregación

IT BOARDING

BOOTCAMP



Consultas **SELECT** y cláusulas

IT BOARDING

BOOTCAMP





SELECT

Las consultas de selección son las más habituales y se utilizan para obtener información de la base de datos en forma de registros. Se debe indicar mínimamente, qué tabla se requiere consultar.

```
SELECT Campos FROM Tabla;
```



Lista de campos que deseo obtener, separados por coma



WHERE

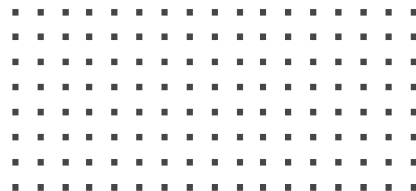
Se utiliza esta cláusula para **filtrar** nuestra consulta dependiendo de la necesidad.

Los campos del **SELECT** no necesariamente tienen que estar en el filtro, es decir, puedo seleccionar campos y en el filtro colocar otros campos que no estén en la selección.

```
SELECT Campos FROM Tabla  
WHERE campo1 = valor;
```



Por ejemplo, se puede filtrar que el campo1 sea igual a un valor. De esta forma, obtengo todos los registros que cumplan con esta condición.





ORDER BY

Esta cláusula permite ordenar los resultados a partir de uno o más campos.

```
SELECT campo1, campo2, campo3  
FROM Tabla  
ORDER BY campo2;
```




Por ejemplo, se puede ordenar por el campo2 (por default el ordenamiento es ascendente).

ORDER BY: Ascendente o Descendente

Esta cláusula permite ordenar de forma ascendente algunos campos y otros campos en forma descendente. Esto dependerá de cómo se requiere mostrar u obtener los datos.

```
SELECT campo1, campo2, campo3  
FROM Tabla  
ORDER BY campo1 DESC;
```



Por ejemplo, se indica explícitamente de qué forma ordenar.

LIMIT

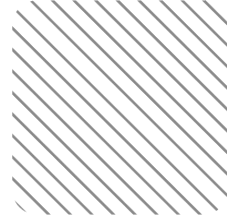
La cláusula **LIMIT** se usa para restringir los registros que se retornan en una consulta **SELECT**.

Se puede establecer un número “N” para limitar a esa cantidad de registros.

```
SELECT campo1, campo2, campo3  
FROM Tabla  
ORDER BY campo1 DESC  
LIMIT 5;
```



Por ejemplo, se indica que en esta consulta se mostrarán solamente 5 registros.

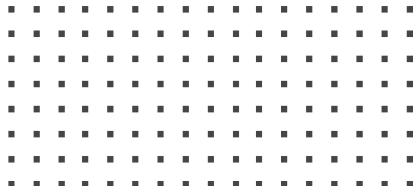


DISTINCT

Se utiliza **DISTINCT** cuando se desea omitir registros que contienen datos duplicados en los campos seleccionados.

Se usa en la instrucción **SELECT** para recuperar valores únicos de una tabla de base de datos.

```
SELECT DISTINCT campo2  
FROM Tabla;
```





OPERADORES



Recuerda que se puede hacer uso de operadores aritméticos, lógicos y de comparación.

```
SELECT campo1, campo2 FROM Tabla  
WHERE campo1 => 0 AND campo3 < 10 ;
```



OPERADORES AND y OR

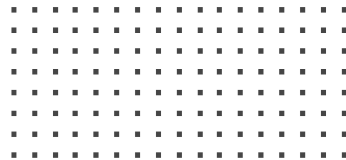
→ El operador **AND** mostrará los resultados cuando se cumplan **las 2 condiciones**.

Ejemplo: condición1 **AND** condición2

→ El operador **OR** mostrará los resultados cuando se cumpla **alguna de las 2 condiciones**.

Ejemplo: condición1 **OR** condición2

```
SELECT campo1, campo2, campo3 FROM Tabla
WHERE campo1 = 1 AND campo2 = 'TEST';
```



LIKE y NOT LIKE

El operador **LIKE** se utiliza en una cláusula **WHERE** para buscar un *patrón específico en una columna*. Para *negar* se puede usar el **NOT LIKE**.

```
SELECT campo1, campo2, campo3
FROM Tabla
WHERE campo1 = 1 AND campo2 LIKE 'T%';
```





BETWEEN

Este operador permite seleccionar un **rango**. Es utilizado para establecer un **criterio de selección** dentro de la cláusula **WHERE** y permite establecer los valores que deben tener los extremos (inclusivos) de ese **rango** deseado.

```
SELECT campo1, campo2, campo3
FROM Tabla
WHERE campo1 = 1 AND campo2 BETWEEN '20200101' AND '20200131';
```



Funciones de Agregación

IT BOARDING

BOOTCAMP





FUNCIONES DE AGREGACIÓN

Se utilizan para visualizar cierta información agrupada y resumida, para esto utilizaremos las funciones de agregación.

- **COUNT:** devuelve el número total de filas seleccionadas por la consulta.
- **MIN:** devuelve el valor mínimo del campo que se especifique.
- **MAX:** devuelve el valor máximo del campo que se especifique.
- **SUM:** suma los valores del campo que se especifique.
- **AVG:** devuelve el valor promedio del campo que se especifique.



FUNCIONES DE AGREGACIÓN

Estas funciones se utilizan en conjunto con la sentencia SELECT.

Ejemplos de sintaxis:

```
SELECT MAX(campo1) FROM Tabla;
```

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




FUNCIONES DE AGREGACIÓN

A modo de ejemplo, teniendo una base de datos de películas se quiere obtener el título de la película que más premios ha obtenido. Podríamos ejecutar:

```
SELECT max(awards), title  
FROM movies
```

Resultado de la consulta:

max(awards)	title
11	Avatar



ALGUNAS BUENAS PRÁCTICAS

- **Evitar** el uso del **SELECT *** (Seleccionar solo los campos que se necesiten).
- **Evitar** realizar consultas **sin filtros** o **limite** de **registros**.
- No se recomienda usar el **order by** para consultas con **grandes volúmenes de datos**.
- Usar las cláusulas adecuadas para acotar el set de datos.
- Si se utilizan varias tablas en una consulta, especificar a qué tabla pertenece cada campo.





Gracias!

IT BOARDING

BOOTCAMP

