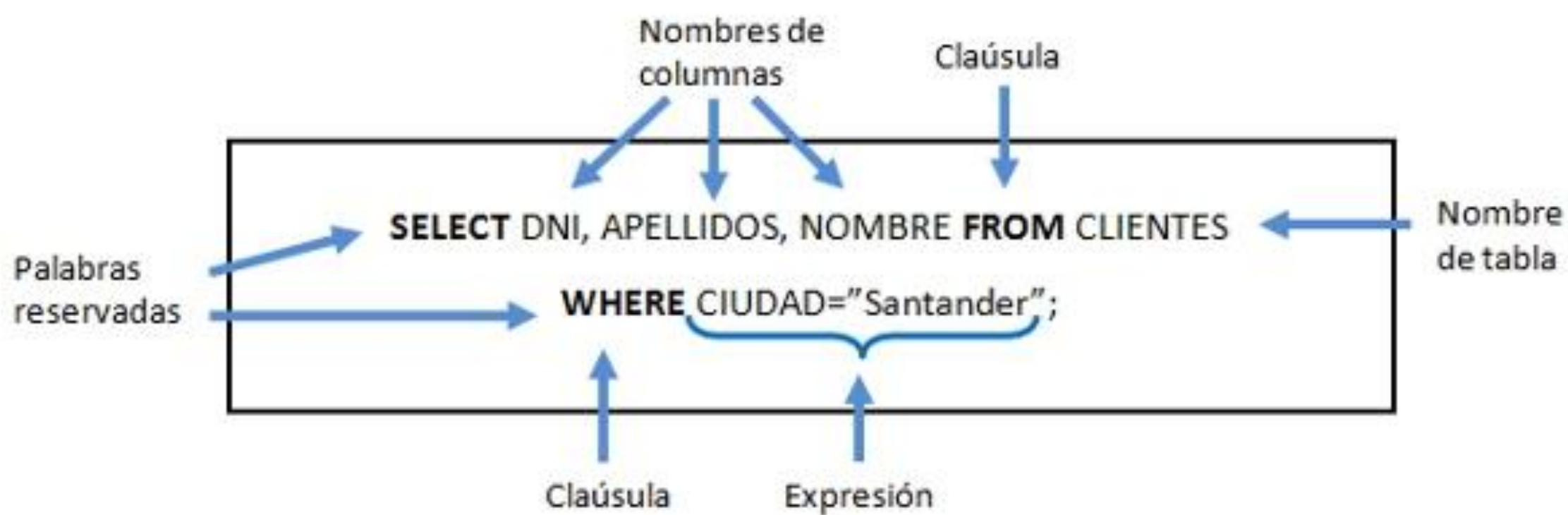




# REPASO EXAMEN

T5 – DML Consultas y subconsultas





# CONSULTAS

**SELECT:** Permite seleccionar las columnas que se van a mostrar y en el orden en que lo van a hacer.

## **ALL / DISTINCT**

**ALL** es el valor predeterminado, especifica que el conjunto de resultados puede incluir filas duplicadas. Por regla general nunca se utiliza.

**DISTINCT** especifica que el conjunto de resultados sólo puede incluir filas únicas.

## **Nombres de campos**

Se debe especificar una lista de nombres de campos de la tabla que nos interesan separados por comas.

Los nombres indicados deben coincidir exactamente con los nombres de los campos de la tabla.

En esta lista se pueden utilizar constantes, expresiones aritméticas, y funciones, para obtener campos calculados de manera dinámica.

Si queremos que nos devuelva todos los campos de la tabla utilizamos el comodín “\*” (**asterisco**).

# CONSULTAS

**AS:** Permite renombrar columnas si lo utilizamos en la cláusula SELECT, o renombrar tablas si lo utilizamos en la cláusula FROM. Es opcional. Con ello podremos crear diversos alias de columnas y tablas.

**FROM:** Esta cláusula permite indicar las tablas o vistas de las cuales vamos a obtener la información. Se pueden renombrar las tablas usando la instrucción "AS".

**WHERE:** Es donde se especifica la **condición de filtro** de las filas devueltas. Se utiliza cuando no se desea que se devuelvan todas las filas de una tabla, sino sólo las que cumplen ciertas condiciones.

**Condiciones:** Son **expresiones lógicas** a comprobar para la condición de filtro, devolverán para cada fila TRUE o FALSE, en función de que se cumplan o no.

# OPERADORES

Por supuesto es posible combinar varias condiciones simples de los operadores anteriores utilizando los operadores lógicos OR, AND y NOT, así como el uso de paréntesis para controlar la prioridad de los operadores (como en matemáticas).

Por ejemplo: ... (Cliente = 100 AND Provincia = 30) OR Ventas > 1000 ... que sería para los clientes de las provincias 100 y 30 o cualquier cliente cuyas ventas superen 1000.

> (Mayor)	< (Menor)
>= (Mayor o igual)	<= (Menor o igual)
= (Igual)	<> o != (Distinto)
IS NULL ((si contiene o no valor nulo)	IS NOT NULL (si contiene o no valor)
LIKE: para la comparación de un modelo.	BETWEEN: para un intervalo de valores.
IN( ): para especificar una relación de valores concretos.	EXISTS/ NOT EXISTS: comprueba si existen o no un valor especificado. Devuelve true/falso.

# CONSULTAS

**ORDER BY:** Define el orden de las filas del conjunto de resultados. Se especifica el campo o campos (separados por comas) por los cuales queremos ordenar los resultados.

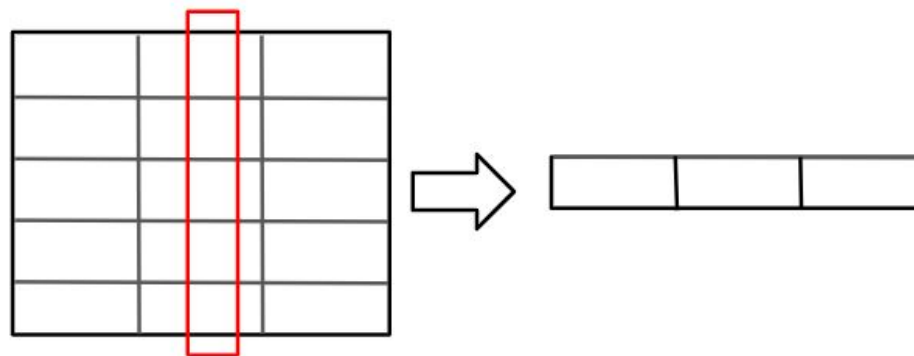
## ASC / DESC

ASC ordenará de forma ascendente, o sea, de menor a mayor (ASC es predeterminado).

DESC se ordenará de forma descendente (de mayor a menor).

OJO: Podemos necesitar ordenar bajo un criterio y en segundo lugar utilizar otro orden, en ese caso escribiremos los criterios separados por comas (ORDER BY columna DESC, columna2 ASC ...).

# CONSULTAS



Funciones de Agregación	
COUNT	Regresa el número total de registros seleccionados
SUM	Suma los valores de una Columna
MIN	Regresa el valor mínimo de una columna
MAX	Regresa el valor máximo de una columna
AVG	Calcula el promedio de una Columna.

```
-- Número de polideportivos hay en cada ciudad, solamente de aquellas  
-- ciudades donde hay más de 10.000  
SELECT ciudad, COUNT(*) AS cantidad  
FROM polideportivos  
GROUP BY ciudad  
HAVING COUNT(*) > 10000;
```

## CONSULTAS

La cláusula **GROUP BY** permite crear agrupaciones de datos y la cláusula **HAVING** sirve para establecer condiciones sobre dichas agrupaciones.

Se pueden utilizar funciones de columna (SUM, MAX, MIN, AVG, COUNT).





SELECT \*

FROM tabla

WHERE condición =

SUBCONSULTA



SELECT id  
FROM tabla  
WHERE Condición

# SUBCONSULTAS

---

Las subconsultas son consultas que se encuentran dentro de otra consulta más grande en una base de datos.

---

Se utilizan para obtener un conjunto de datos específico que se utilizará como parte de la consulta principal.

---

Las subconsultas se pueden aplicar en diferentes partes de una consulta, como en las cláusulas WHERE, JOIN, HAVING, etc.

# SUBCONSULTAS

EN ESTE TEMA SOLO HEMOS VISTO **SUBCONSULTAS EN LA CLÁUSULA WHERE**

Cuando utilizamos las subconsultas:

Si al escribir una consulta necesitamos un dato en la condición WHERE que no tenemos en la tabla, pero si podemos obtenerlo a través de una consulta.

```
SELECT *  
FROM cliente  
WHERE id NOT IN (SELECT id_cliente FROM pedido);
```

Devuelve el listado de los clientes que todavía no han hecho ningún pedido.

# SUBCONSULTAS

SELECT \*

FROM cliente

WHERE id NOT IN (SELECT id\_cliente FROM pedido);

Utiliza un comparador NOT IN que permite que se devuelvan varias filas.

Compara los id (clave primaria) de clientes de la tabla clientes con los id\_cliente (clave ajena) de la tabla pedido que son EL MISMO VALOR.

Devuelve todos los id\_clientes que hay en la tabla pedido

# SUBCONSULTAS

- Los **operadores básicos de comparación** (>, >=, <, <=, !=, <>, =) se pueden usar cuando queremos comparar una expresión con el valor que devuelve una subconsulta.
- Los operadores básicos de comparación los vamos a utilizar para realizar comparaciones con subconsultas que **devuelven un único valor, es decir, una columna y una fila.**
- **ALL y ANY** se utilizan con los operadores de comparación (>, >=, <, <=, !=, <>, =) y nos permiten comparar una expresión con el conjunto de valores que devuelve una subconsulta.
- ALL y ANY los vamos a utilizar para realizar comparaciones con subconsultas que **pueden devolver varios valores, es decir, una columna y varias filas.**
- **IN y NOT IN** nos permiten comprobar si un valor está o no incluido en un conjunto de valores, que puede ser el conjunto de valores que devuelve una subconsulta.
- IN y NOT IN los vamos a utilizar para realizar comparaciones con subconsultas que **pueden devolver varios valores, es decir, una columna y varias filas.**