

# Las consultas de SQL

## Objetivo

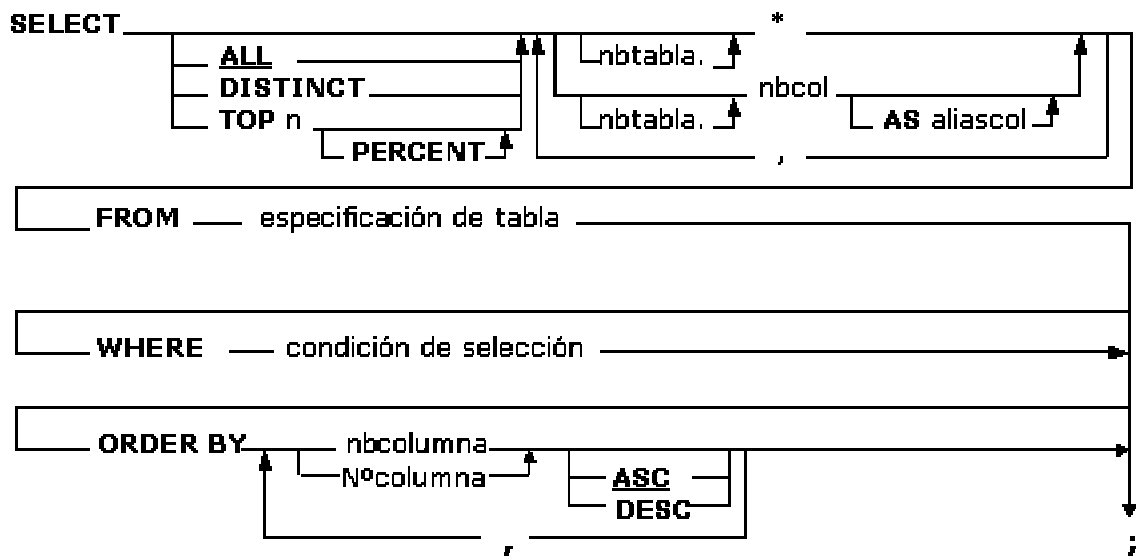
Empezaremos por estudiar la sentencia **SELECT**, que permite **recuperar datos** de una o varias **tablas**. La sentencia SELECT es con mucho la más compleja y potente de las sentencias SQL. Empezaremos por ver las **consultas** más **simples**, **basadas** en una sola **tabla**.

Esta sentencia forma parte del DML (lenguaje de manipulación de datos), en este tema veremos cómo **seleccionar columnas** de una tabla, cómo **seleccionar filas** y cómo obtener las **filas ordenadas** por el criterio que queramos.

El resultado de la consulta es una **tabla lógica**, porque no se guarda en el disco sino que está en memoria y cada vez que ejecutamos la consulta se vuelve a calcular.

Cuando ejecutamos la consulta se visualiza el resultado en forma de tabla con columnas y filas, pues en la SELECT tenemos que indicar qué columnas queremos que tenga el resultado y qué filas queremos seleccionar de la tabla origen.

## Sintaxis de la sentencia SELECT

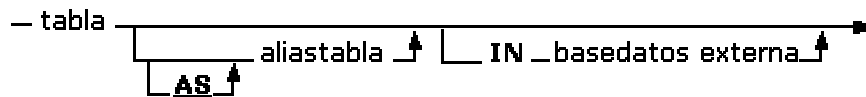


## La tabla origen - FROM -

Con la cláusula **FROM** indicamos **en qué tabla** tiene que **buscar la información**. En este capítulo de consultas simples el resultado se obtiene de una única tabla. La sintaxis de la cláusula es:

### FROM especificación de tabla

Una especificación de tabla puede ser el nombre de una consulta guardada (las que aparecen en la ventana de base de datos), o el nombre de una tabla que a su vez puede tener el siguiente formato:



**Aliastabla** es un nombre de **alias**, es como un **segundo nombre** que asignamos a la **tabla**, si en una consulta definimos un alias para la tabla, esta se deberá nombrar utilizando ese nombre y no su nombre real, además **ese nombre sólo es válido en la consulta** donde se define. El alias se suele emplear en consultas basadas en más de una tabla que veremos en el tema siguiente. La palabra **AS** que se puede poner delante del nombre de alias es opcional y es el valor por defecto por lo que no tienen ningún efecto.

Ejemplo: **SELECT .....FROM oficinas ofi** ; equivalente a **SELECT .....FROM oficinas AS ofi** esta sentencia me indica que se van a buscar los datos en la tabla *oficinas* que queda renombrada en esta consulta con *ofi*.

En una **SELECT** podemos utilizar tablas que no están definidas en la base de datos (siempre que tengamos los permisos adecuados), si la tabla no está en la base de datos activa, debemos indicar en qué base de datos se encuentra con la cláusula **IN**.

En la cláusula **IN** el nombre de la base de datos debe incluir el **camino completo**, la **extensión** (.mdb), y estar entre **comillas simples**.

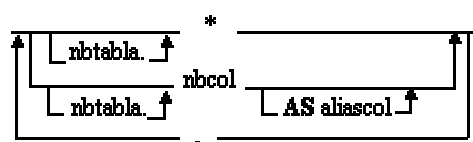
Supongamos que la tabla empleados estuviese en otra base de datos llamada *otra* en la carpeta *c:\mis documentos\*, habría que indicarlo así:

```
SELECT *
FROM empleados IN 'c:\mis documentos\otra.mdb'
```

Generalmente tenemos las tablas en la misma base de datos y no hay que utilizar la cláusula **IN**.

## Selección de columnas

La lista de **columnas** que queremos que **aparezcan** en el **resultado** es lo que llamamos **lista de selección** y se especifica delante de la cláusula **FROM**.



### Utilización del \*

Se utiliza el asterisco \* en la lista de selección para indicar **'todas las columnas de la tabla'**.

Tiene dos **ventajas**:

Evitar nombrar las columnas una a una (es más corto).

Si añadimos una columna nueva en la tabla, esta nueva columna saldrá sin tener que modificar la consulta.

Se puede combinar el \* con el nombre de una tabla (ej. *oficinas.\**), pero esto se utiliza más cuando el origen de la consulta son dos tablas.

**SELECT \* FROM oficinas**

o bien

**SELECT oficinas.\* FROM oficinas**      Lista todos los datos de las oficinas

#### **columnas de la tabla origen**

Las columnas se pueden especificar mediante su **nombre simple** (nbc) o su **nombre cualificado** (nbt.nbc), el nombre de la columna precedido del nombre de la tabla que contiene la columna y separados por un punto).

El nombre cualificado se puede emplear siempre que queramos y es obligatorio en algunos casos que veremos más adelante.

Cuando el **nombre** de la **columna** o de la tabla **contiene espacios en blanco**, hay que poner el nombre **entre corchetes [ ]** y además el número de espacios en blanco debe coincidir. Por ejemplo [codigo de cliente] no es lo mismo que [ código de cliente] (el segundo lleva un espacio en blanco delante de código)

Ejemplos :

**SELECT nombre, oficina, contrato FROM ofiventas**

Lista el nombre, oficina, y fecha de contrato de todos los empleados.

**SELECT idfab, idproducto, descripcion, precio FROM productos**

Lista una tarifa de productos

#### **Alias de columna.**

Cuando se visualiza el resultado de la consulta, normalmente las columnas toman el nombre que tiene la columna en la tabla, si queremos cambiar ese nombre lo podemos hacer definiendo un alias de columna mediante la cláusula **AS** será el nombre que aparecerá como **título de la columna**.

Ejemplo:

**SELECT idfab AS fabricante, idproducto, descripcion FROM productos**

Como título de la primera columna aparecerá fabricante en vez de idfab

#### **Columnas calculadas.**

Además de las columnas que provienen directamente de la tabla origen, una consulta SQL puede incluir **columnas calculadas** cuyos valores se calculan a partir de los valores de los datos almacenados.

Para solicitar una columna calculada, se especifica en la lista de selección una **expresión** en vez de un nombre de columna. La expresión puede contener sumas, restas, multiplicaciones y divisiones, concatenación & , paréntesis y también funciones predefinidas).

Ejemplos:

**SELECT ciudad, región, (ventas-objetivo) AS superavit FROM oficinas**

Lista la ciudad, región y el superavit de cada oficina.

**SELECT idfab, idproducto, descripcion, (existencias \* precio) AS valoracion FROM productos**

De cada producto obtiene su fabricante, idproducto, su descripción y el valor del inventario

**SELECT nombre, MONTH(contrato), YEAR(contrato) FROM repventas**

Lista el nombre, mes y año del contrato de cada vendedor.

La función MONTH() devuelve el mes de una fecha

La función YEAR() devuelve el año de una fecha

**SELECT oficina, 'tiene ventas de ', ventas FROM oficinas**

Listar las ventas en cada oficina con el formato: 22 tiene ventas de 186,042.00 €