

Módulo 2. Introducción al SQL

Consultas en SQL

Una consulta es la operación que permite separar los datos necesarios de una o varias tablas. Siguiendo con el ejemplo de los estudiantes y los cursos que ya conoces, si la consulta que necesito realizar es sobre los alumnos matriculados en un determinado curso, la base de datos se encargará de separar esos datos de las tablas correspondientes y dárme los como resultado.

Este tipo de consulta que consiste en obtener relaciones de datos de tablas diferentes ya sea para crear vistas, o para filtrar a la hora de modificar o borrar datos de una tabla, etc. es la sentencia más utilizada.

Ya vimos en uno de nuestros vídeos que para realizar una consulta se utiliza la sentencia SELECT. Vamos a aprender un poco más de esta sentencia, ya que es poderosa. Sí, es poderosa, porque es muy rica en opciones, lo que significa que nos permitirá realizar todas las operaciones del álgebra relacional.

Para que puedas entenderlo mejor, la sintaxis original resumida es esta, así que presta mucha atención:

```
SELECT nombre_de_columnas FROM tabla_origen [WHERE condición]
[GROUP BY nombre_de_columna_agrupación] [HAVING condición_group_by]
[ORDER BY nombre_columna];
```

A lo mejor aún no lo ves claro. Espera, no te impacientes y desglosaremos cada una de las opciones de forma separada para que lo entiendas con más facilidad.

La sentencia SELECT básica, la más sencilla, consiste en seleccionar las columnas de una tabla, incluyendo todas las filas de la tabla. Observa:

Indica el **nombre de la columna a seleccionar**. Habitualmente se escribe un asterisco "SELECT *..." lo que indica que se quieren seleccionar todas las columnas de la tabla.

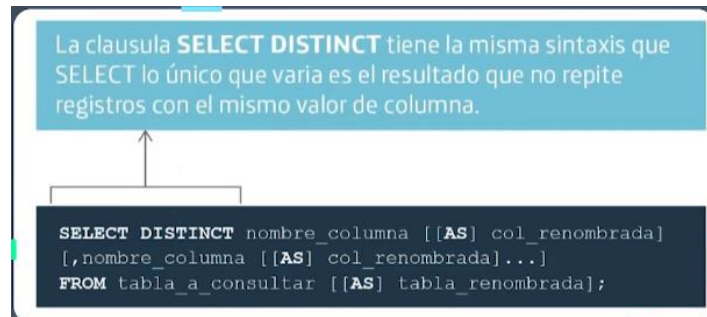
La cláusula **AS** permite **renombrar la columna**. Es decir permite la creación de Alias. El nuevo nombre será renombrado en el resultado de la consulta pero no en la tabla de origen.

```
SELECT nombre_columna [[AS] col_renombrada]
[, nombre_columna [[AS] col_renombrada]...]
FROM tabla_a_consultar [[AS] tabla_renombrada];
```

La cláusula **FROM** es obligatoria en todas las sentencias **SELECT**, e indica la **tabla o tablas a consultar**. También admite la creación de Alias.

Ahora vamos a complicarlo un poquito más. Añadiremos a este formato básico la palabra **DISTINCT** después de **SELECT** para una nueva opción. Este cambio nos permitirá obtener los registros sin repeticiones.

Observa atentamente su sintaxis:



Observa que **SELECT DISTINCT** tiene la misma sintaxis que **SELECT**. Lo único que varía es el resultado, que ya no repite registros con el mismo valor de columna.

Veamos un ejemplo. Seleccionamos todos los clientes de nuestra empresa y mostramos solo las columnas de DNI (a las que le incluimos un Alias), nombre y apellidos.

```
SELECT DNI AS Identificación, Nombre, Apellidos
FROM Clientes;
```

Así es como seleccionamos columnas. Pero ¿y si queremos seleccionar filas?

En el formato anterior no podíamos filtrar los registros de una tabla, porque se seleccionaban todos. Para poder filtrar los registros debemos añadir a la sentencia **SELECT** la cláusula **WHERE**.

En este caso la sintaxis sería:



Hay que tener en cuenta que, al realizar una selección, la tabla generada es temporal.

Veamos otro ejemplo sencillo. Pero esta vez, añadiremos condiciones a nuestra selección. Seleccionaremos las columnas de DNI, Nombre y Apellidos de la tabla clientes que tengan fecha de alta mayor a '01/01/2015'

```
SELECT DNI AS Identificación, Nombre, Apellidos
FROM Clientes
WHERE FECHA_DE_ALTA>'01/01/2015';
```

Como puedes ver, en la condición hemos utilizado un operador de comparación.

Otro aspecto importante que podemos tratar con la sentencia SELECT es la ordenación del resultado de una consulta. Porque podemos conseguir que al hacer la consulta los registros que obtengamos se nos muestren ordenados respecto a una o varias columnas, un orden que nos facilitará mucho su manejo.

La sentencia en este caso seguirá siendo SELECT, y la cláusula que permite la ordenación en las consultas es ORDER BY.

Para entender mejor su funcionamiento veamos detenidamente este gráfico.



Como ves, esta sentencia nos pide que seleccionemos las columnas que queremos ordenar, y nos deja elegir una ordenación ascendente o descendente, entre otras posibilidades.

Veamos un último ejemplo. Imagina ahora que queremos ordenar los clientes según su fecha de alta, del más antiguo al más reciente. Y, a su vez, dentro de esta clasificación, queremos además ordenar los datos por orden alfabético según los apellidos.