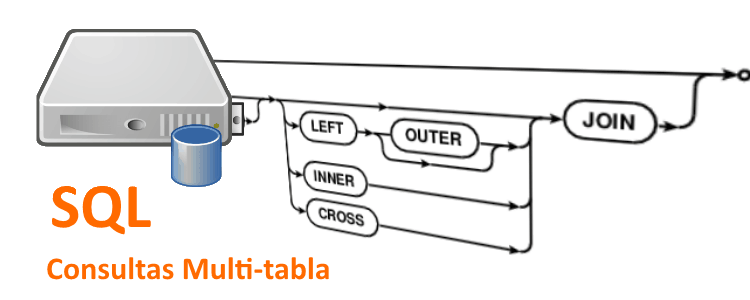
**2 – LENGUAJE DE CONSULTAS A UNA BASE DE DATOS.**

**6. CONSULTAS TABLAS RELACIONADAS**

**Clase 37**

**Objetivo de la jornada:**

* Comprender y ejecutar sentencias para la consulta de datos en base de datos relacionales
* Ejecutar y desarrollar el modelo relacional graficando soluciones para problemas planteados, consultando datos en diversas tablas.



Continuando con nuestro camino. A continuación vamos a complicar un poco la cosa **aprendiendo a realizar consultas en varias tablas** de la base de datos al mismo tiempo. Es habitual que queramos acceder a datos que se encuentran en más de una tabla y mostrar información mezclada de todas ellas como resultado de una consulta. Para ello tendremos que hacer combinaciones de columnas de tablas diferentes.

En [SQL](https://www.campusmvp.es/catalogo/Product-SQL-Server-pr%C3%A1ctico-para-desarrolladores_165.aspx) es posible hacer esto especificando más de una tabla en la cláusula FROM de la instrucción SELECT.

**Tenemos varias formas de obtener esta información.**

**Una de ellas** consiste en **crear combinaciones** que permiten mostrar **columnas de diferentes tablas como si fuese una sola** tabla, haciendo coincidir los valores de las columnas relacionadas.

Este último punto es muy importante, ya que si seleccionamos varias tablas y no hacemos coincidir los valores de las columnas relacionadas, obtendremos una gran duplicidad de filas, **realizándose el**[**producto cartesiano**](https://es.wikipedia.org/wiki/Producto_cartesiano) entre las filas de las diferentes tablas seleccionadas.

Vamos a ver este importante detalle con un ejemplo simple. Consideremos estas tres consultas sobre la [base de datos Northwind](https://www.campusmvp.es/recursos/post/VIDEO-Como-instalar-paso-a-paso-la-base-de-datos-Northwind.aspx):

SELECT COUNT(\*) FROM Customers

SELECT COUNT(\*) FROM Orders

SELECT COUNT(\*) FROM Customers, Orders

La primera instrucción devuelve 91 filas (los 91 clientes), la segunda 830 filas (los pedidos), y la tercera 75.530 (que son 830 x 91, es decir, la combinación de todas las filas de clientes y de pedidos).

**La otra manera** de mostrar información de varias tablas -mucho más habitual y lógica- es **uniendo filas de ambas**, para ello es necesario que las columnas que se van a unir entre las dos tablas sean las mismas y contengan los mismos tipos de datos, es decir, [**mediante una clave externa**](https://www.campusmvp.es/recursos/post/Disenando-una-base-de-datos-en-el-modelo-relacional.aspx).

## Operaciones de unión - JOIN

La **operación JOIN o combinación** permite mostrar columnas de varias tablas como si se tratase de una sola tabla, **combinando entre sí los registros relacionados** usando para ello claves externas.

Las tablas relacionadas se especifican en la cláusula FROM, y además hay que hacer coincidir los valores que relacionan las columnas de las tablas.

Veamos un ejemplo, que selecciona el número de venta, el código y nombre del cliente y la fecha de venta en la base de datos Northwind:

SELECT OrderID, C.CustomerID, CompanyName, OrderDate

FROM Customers C, Orders O

WHERE C.CustomerID = O.CustomerID

Para evitar que se produzca como resultado el producto cartesiano entre las dos tablas, expresamos el vínculo que se establece entre las dos tablas en la cláusula WHERE. En este caso relacionamos ambas tablas mediante el identificador del cliente, clave existente en ambas. Fíjate en como le hemos otorgado un alias a cada tabla (C y O respectivamente) para no tener que escribir su nombre completo cada vez que necesitamos usarlas.

Hay que tener en cuenta que si el nombre de una columna existe en más de una de las tablas indicadas en la cláusula FROM, hay que poner, obligatoriamente, el nombre o alias de la tabla de la que queremos obtener dicho valor. En caso contrario nos dará un error de ejecución, indicando que hay un nombre ambiguo.

Hay otra forma adicional, que es más explícita y clara a la hora de realizar este tipo de combinaciones -y que se incorpora a partir de ANSI SQL-92- que permite utilizar una nueva cláusula llamada **JOIN** en la cláusula FROM, cuya sintaxis es el siguiente:

SELECT [ ALL / DISTINC ] [ \* ] / [ListaColumnas\_Expresiones]

FROM NombreTabla1 JOIN NombreTabla2 ON Condiciones\_Vinculos\_Tablas

De esta manera **relacionamos de manera explícita ambas tablas**sin necesidad de involucrar la clave externa en las condiciones del SELECT (o sea, en el WHERE). Es una manera más clara y limpia de llevar a cabo la relación.

Esto se puede ir aplicando a cuantas tablas necesitemos combinar en nuestras consultas. Veamos un ejemplo en ambos formatos que involucra más tablas, en este caso las tablas de empleados, clientes y ventas:

SELECT OrderID, C.CustomerID, CompanyName, OrderDate

FROM Customers C, Orders O, Employees E

WHERE C.CustomerID = O.CustomerID AND O.EmployeeID = E.EmployeeID

El segundo formato permite distinguir las condiciones que utilizamos para combinar las tablas y evitar el producto cartesiano, de las condiciones de filtro que tengamos que establecer.

Veamos un ejemplo como el anterior, pero ahora además necesitamos que el cliente sea de España o el vendedor sea el número 5.

En el primer formato tendríamos algo como esto:

SELECT OrderID, C.CustomerID, CompanyName, OrderDate

FROM Customers C, Orders O, Employees E

WHERE (C.CustomerID = O.CustomerID AND O.EmployeeID = E.EmployeeID)

AND (C.Country = 'Spain' OR E.EmployeeID = 5)

Es decir, estamos mezclando en el WHERE las uniones de tablas, y las condiciones concretas de filtro de la consulta, quedando todo mucho más liado.

Sin embargo usando el segundo formato con JOIN, la consulta es mucho más clara:

SELECT OrderID, C.CustomerID, CompanyName, OrderDate

FROM Customers C JOIN Orders O ON C.CustomerID = O.CustomerID JOIN Employees E ON O.EmployeeID = E.EmployeeID

WHERE C.Country = 'Spain' OR E.EmployeeID = 5

Aquí se aprecia claramente que la utilización de JOIN simplifica la lectura y comprensión de la instrucción SQL, ya que no necesita el uso de paréntesis y tiene una condición WHERE más sencilla.

También podemos utilizar una misma tabla con dos alias diferentes para distinguirlas. Veamos un ejemplo, supongamos que tenemos una columna sueldo en la tabla de empleados, y queremos saber los empleados que tienen un sueldo superior al del empleado 5:

SELECT E1.EmployeeID

FROM Employees E1 JOIN Employees E2 ON E1.Sueldo > E2.Sueldo

WHERE E2.EmployeeID = 5

Con esto hemos aprendido lo básico de trabajar con varias tablas y generar combinaciones de datos entre éstas.

|  |  |
| --- | --- |
| **Plan Formativo:** | **Nivel de Dificultad:** |
| **Full Stack Java Trainee** | **Básico/medio.** |
| **Módulo:** | **Tema:** |
| **LENGUAJE DE CONSULTAS A UNA BASE DE DATOS** | **Consultas tablas relacionadas.** |
| **Título del Problema** | **Realizando consultas en tablas relacionadas** |
| **Intención del aprendizaje o aprendizaje esperado** | |
| * Integración de consultas multitablas. * Manejo y ejecución de Queries. * Obtención de información de tablas. * Construir sentencias de creación de una tabla utilizando DDL y definiendo campos, tipos de dato, nulidad, llaves primarias y foráneas de acuerdo a un modelo de datos existente para satisfacer un requerimiento | |
| **Planteamiento del Problema** | |
| En base a la QUERY, Entregada obtenga:Consultas sobre una tabla  1. Lista el nombre de todos los productos que hay en la tabla producto. 2. Lista los nombres y los precios de todos los productos de la tabla producto. 3. Lista todas las columnas de la tabla producto. 4. Lista el nombre de los productos, el precio en euros y el precio en dólares estadounidenses (USD). 5. Lista el nombre de los productos, el precio en euros y el precio en dólares estadounidenses (USD). Utiliza los siguientes alias para las columnas: nombre de producto, euros, dólares. 6. Lista los nombres y los precios de todos los productos de la tabla producto, convirtiendo los nombres a mayúscula. 7. Lista los nombres y los precios de todos los productos de la tabla producto, convirtiendo los nombres a minúscula. 8. Lista el nombre de todos los fabricantes en una columna, y en otra columna obtenga en mayúsculas los dos primeros caracteres del nombre del fabricante. 9. Lista los nombres y los precios de todos los productos de la tabla producto, redondeando el valor del precio. 10. Lista los nombres y los precios de todos los productos de la tabla producto, truncando el valor del precio para mostrarlo sin ninguna cifra decimal. 11. Lista el código de los fabricantes que tienen productos en la tabla producto. 12. Lista el código de los fabricantes que tienen productos en la tabla producto, eliminando los códigos que aparecen repetidos. 13. Lista los nombres de los fabricantes ordenados de forma ascendente. 14. Lista los nombres de los fabricantes ordenados de forma descendente. 15. Lista los nombres de los productos ordenados en primer lugar por el nombre de forma ascendente y en segundo lugar por el precio de forma descendente. 16. Devuelve una lista con las 5 primeras filas de la tabla fabricante.  Consultas multitabla (Composición interna) Resuelva todas las consultas utilizando la sintaxis de SQL   1. Devuelve una lista con el nombre del producto, precio y nombre de fabricante de todos los productos de la base de datos. 2. Devuelve una lista con el nombre del producto, precio y nombre de fabricante de todos los productos de la base de datos. Ordene el resultado por el nombre del fabricante, por orden alfabético. 3. Devuelve una lista con el código del producto, nombre del producto, código del fabricante y nombre del fabricante, de todos los productos de la base de datos. 4. Devuelve el nombre del producto, su precio y el nombre de su fabricante, del producto más barato. 5. Devuelve el nombre del producto, su precio y el nombre de su fabricante, del producto más caro. 6. Devuelve una lista de todos los productos del fabricante Lenovo. 7. Devuelve una lista de todos los productos del fabricante Crucial que tengan un precio mayor que 200€. 8. Devuelve un listado con todos los productos de los fabricantes Asus, Hewlett-Packardy Seagate. Sin utilizar el operador IN. 9. Devuelve un listado con todos los productos de los fabricantes Asus, Hewlett-Packardy Seagate. Utilizando el operador IN. 10. Devuelve un listado con el nombre y el precio de todos los productos de los fabricantes cuyo nombre termine por la vocal e. 11. Devuelve un listado con el nombre y el precio de todos los productos cuyo nombre de fabricante contenga el carácter w en su nombre. 12. Devuelve un listado con el nombre de producto, precio y nombre de fabricante, de todos los productos que tengan un precio mayor o igual a 180€. Ordene el resultado en primer lugar por el precio (en orden descendente) y en segundo lugar por el nombre (en orden ascendente) 13. Devuelve un listado con el código y el nombre de fabricante, solamente de aquellos fabricantes que tienen productos asociados en la base de datos.  1.1.5 Consultas multitabla (Composición externa) Resuelva todas las consultas utilizando las cláusulas LEFT JOIN y RIGHT JOIN.   1. Devuelve un listado de **todos los fabricantes** que existen en la base de datos, junto con los productos que tiene cada uno de ellos. El listado deberá mostrar también aquellos fabricantes que no tienen productos asociados. 2. Devuelve un listado donde sólo aparezcan aquellos fabricantes que no tienen ningún producto asociado. 3. ¿Pueden existir productos que no estén relacionados con un fabricante? Justifique su respuesta.  1.1.6 Consultas resumen  1. Calcula el número total de productos que hay en la tabla productos. 2. Calcula el número total de fabricantes que hay en la tabla fabricante. 3. Calcula el número de valores distintos de código de fabricante aparecen en la tabla productos. 4. Calcula la media del precio de todos los productos. 5. Calcula el precio más barato de todos los productos. 6. Calcula el precio más caro de todos los productos. 7. Lista el nombre y el precio del producto más barato. 8. Lista el nombre y el precio del producto más caro. 9. Calcula la suma de los precios de todos los productos.   **(Tiempo aproximado 180 minutos)** | |
| **Datos de apoyo al planteamiento** | |
|  | |
| **Pregunta Guía:** | |
|  | |
| **Recursos Bibliográficos.** | |
| <https://www.campusmvp.es/recursos/post/Fundamentos-de-SQL-Consultas-SELECT-multi-tabla-JOIN.aspx> | |
| **Nombre del Autor y Fecha:** | |
|  | |