Introducción a SQLite

ALUMNO: Bisutti Moscatelli German DNI: 45928501

**PARTE 3: Data Manipulation Language, JOIN.**

Para esta práctica utilizaremos la base de datos incluida en el mail. Para simplificar el acceso, descargue todos los archivos en una carpeta nueva, coloque una copia de SQLITE3.EXE, y ejecute Ordenes.bat.

Nos centraremos en la recuperación de datos relacionados de varias tablas de la base de datos, veremos el comando de SQL: JOIN. Este comando nos permite juntar o unir 2 tablas, relacionando sus datos, y tomarla como una sola. Veremos algunos ejemplos y ustedes tendrán contestar las consignas, para ello pueden probar sus comandos en SQLITE.

Veamos 3 formas distintas de juntar 2 tablas. Vamos a listar el nombre del proveedor de la tabla Proveedores y el nombre del producto de la tabla Productos que están relacionados por el campo proveedor\_id (que sirve de clave principal de la tabla Proveedores y clave foránea de la tabla Productos)

Sin usar JOIN, usando WHERE:

SELECT proveedor\_nombre, producto\_nombre FROM Proveedores, Productos WHERE Proveedores.proveedor\_id=Productos.proveedor\_id;

Usando JOIN el comando equivalente es:

SELECT proveedor\_nombre, producto\_nombre FROM Proveedores JOIN Productos ON Proveedores.proveedor\_id=Productos.proveedor\_id;

Como ven en las dos tablas el campo de que las referencian tiene el mismo nombre (Proveedor\_id) por lo que también se puede utilizar:

SELECT proveedor\_nombre, producto\_nombre FROM Proveedores JOIN Productos USING (proveedor\_id);

1. Escribir el comando para listar los nombres de productos y sus cantidades de la orden de compra 20008. Como ayuda, pueden combinar un JOIN con WHERE.

SELECT producto\_nombre, cantidad FROM Productos JOIN Orden\_Items ON Productos.producto\_ID = Orden\_Items.producto\_ID WHERE Orden\_Items.orden\_numero = 20008;

1. Escribir el comando para listar los nombres de los clientes, el número de orden y la fecha.

SELECT cliente\_nombre, orden\_numero, orden\_fecha FROM Clientes JOIN Ordenes ON Clientes.cliente\_ID = Ordenes.cliente\_ID;

También pueden utilizar un comando SELECT dentro de otro SELECT realizando una sub-búsqueda. Por ejemplo veamos como listar los nombres de cliente con el total de órdenes pedidas

1. SELECT Cliente\_nombre, (SELECT COUNT(\*) FROM Ordenes WHERE Clientes.cliente\_id=ordenes.cliente\_id) AS Ordenes FROM Clientes;
2. Escribir el comando para listar los nombres de los proveedores con el total de productos que tiene.

SELECT proveedor\_nombre, (SELECT COUNT(\*) FROM Productos WHERE Proveedores.proveedor\_ID = Productos.proveedor\_ID) AS Productos FROM Proveedores;

Un resultado similar pueden obtener con

1. SELECT Cliente\_nombre, COUNT(\*) AS Total\_Ordenes FROM Clientes JOIN Ordenes USING (cliente\_id) GROUP BY cliente\_id;
2. ¿Cuál es la diferencia en el resultado de utilizar el comando A) y el B)? ¿Cuál es más conveniente utilizar?

el comando A devuelve todos los clientes incluso si no tiene ordenes devolverá 0, en cambio e B no mostraría aquellos que no tengan órdenes. El comando B es más eficaz porque no realiza una consulta por cada cliente, pero tiene la desventaja de no mostrar los clientes con 0 consultas

1. Reescribir el comando del punto 3) utilizando la JOIN

SELECT Proveedores.proveedor\_nombre, COUNT(Productos.producto\_ID) AS Productos FROM Proveedores LEFT JOIN Productos USING (proveedor\_ID) GROUP BY proveedor\_ID;

1. ¿Cuál es la diferencia en el resultado entre el comando B) y el siguiente:?

SELECT Cliente\_nombre, COUNT(\*) AS Total\_Ordenes FROM Clientes JOIN Ordenes USING (cliente\_id) GROUP BY cliente\_nombre;

¿Por qué?

El Comando A cuenta las órdenes de los clientes utilizando cliente\_id, lo que permite distinguir entre clientes únicos, incluso si tienen el mismo nombre. Esto garantiza que cada cliente con un cliente\_id diferente se cuente por separado. El Comando B agrupa los resultados por cliente\_nombre, lo que significa que todos los clientes con el mismo nombre se combinan en una sola fila. Esto puede llevar a un conteo que no refleja con precisión el número de clientes únicos, especialmente si hay duplicados en la tabla Clientes.

1. Utilizando la bibliografía, describa las diferencias y similitudes entre JOIN y:

NATURAL JOIN: une tablas basándose en todas las columnas que tienen el mismo nombre en ambas tablas. No es necesario especificar la condición de unión.

LEFT JOIN Devuelve todas las filas de la tabla de la izquierda (tabla1) y las filas coincidentes de la tabla de la derecha (tabla2). Si no hay coincidencias, el resultado de las columnas de la tabla de la derecha será NULL.

1. Ahora un poco más difícil. Escribir el comando para listar Nombre del cliente, Número de Orden y el Precio total de la orden (la suma de Item\_precio por cantidad)

Primer comando - SELECT Clientes.cliente\_nombre, Ordenes.orden\_numero, SUM(Orden\_Items.item\_precio \* Orden\_Items.cantidad) AS Precio\_Total

Segundo comando - FROM Clientes JOIN Ordenes ON Clientes.cliente\_ID = Ordenes.cliente\_ID JOIN Orden\_Items ON Ordenes.orden\_numero = Orden\_Items.orden\_numero

Tercer comando - GROUP BY Ordenes.orden\_numero;

1. Escribir el comando para listar Nombre del proveedor, nombre del producto y la cantidad total a pedir entre todas las órdenes de compra. Si se animan, ordenar el listado por proveedor.

SELECT Proveedores.proveedor\_nombre, Productos.producto\_nombre, SUM(Orden\_Items.cantidad) AS Cantidad\_Total

FROM Proveedores JOIN Productos ON Proveedores.proveedor\_ID = Productos.proveedor\_ID JOIN Orden\_Items ON Productos.producto\_ID = Orden\_Items.producto\_ID

GROUP BY Proveedores.proveedor\_nombre, Productos.producto\_nombre;

1. Escribir el comando para listar Fecha de la orden, Numero de orden, Numero de ítem, Nombre del producto y cantidad.

SELECT Ordenes.orden\_fecha, Ordenes.orden\_numero, Orden\_Items.orden\_item, Productos.producto\_nombre, Orden\_Items.cantidad FROM

Ordenes JOIN Orden\_Items ON Ordenes.orden\_numero = Orden\_Items.orden\_numero JOIN Productos ON Orden\_Items.producto\_ID = Productos.producto\_ID;

1. El producto con ID BR03 no está disponible. Necesitamos un listado del nombre, contacto y mail de todos los clientes que hayan pedido el producto BR03.

SELECT Clientes.cliente\_nombre, Clientes.cliente\_contacto, Clientes.cliente\_email FROM Clientes JOIN Ordenes ON Clientes.cliente\_ID = Ordenes.cliente\_ID

JOIN Orden\_Items ON Ordenes.orden\_numero = Orden\_Items.orden\_numero WHERE Orden\_Items.producto\_ID = 'BR03';

 (CC)Luis Pérez – 2020

<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode.es>