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;

2) 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

- A) SELECT Cliente_nombre, (SELECT COUNT(*) FROM Ordenes WHERE Clientes.cliente id=ordenes.cliente id) AS Ordenes FROM Clientes;
- 3) 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

- B) SELECT Cliente_nombre, COUNT(*) AS Total_Ordenes FROM Clientes JOIN
 Ordenes USING (cliente id) GROUP BY cliente id;
 - 4) ¿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

5) 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;

6) ¿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.

7) 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.

8) 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;

9) 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;

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

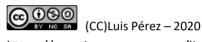
SELECT Ordenes.orden fecha, Ordenes.orden numero, Orden Items.orden item, Productos.producto nombre, Orden Items.cantidad FROM

Ordenes JOIN Orden_Items ON Ordenes.orden_numero = Orden_Items.orden_numero JOIN Productos ON Orden_Items.producto_ID = Productos.producto_ID;

11) 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';



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