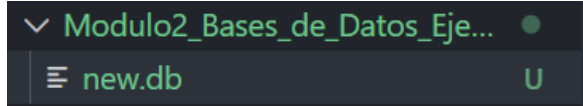


SQL

1. Cree una nueva base de datos en SQLite.



2. Replique las tablas creadas anteriormente en Ejercicios de Bases de Datos, con sus respectivos PKs, FKs, constraints, y demás requerimientos.

- Investigue cómo hacer que los PKs se generen **automáticamente**.
- Utilice los tipos de datos adecuados.
- Si existe alguna limitante por SQLite, documéntela y resuelva la limitante como considere adecuado.

3. Modifique la tabla de Facturas creada en el ejercicio anterior y agregue una columna para almacenar también el número de teléfono del comprador, y otra para el código de empleado del cajero que realizó la venta.

Ambas respuestas del punto 2 y 3 están abajo

↓↓

↓↓

↓↓

```
new.db U -- SQLite Untitled-1
1  -- SQLite
2  CREATE TABLE producto ( "producto": Misspelled word.
3      id INTEGER PRIMARY KEY AUTOINCREMENT, "AUTOINCREMENT": Unknown word
4      codigo SMALLINT NOT NULL, "codigo": Unknown word.
5      nombre VARCHAR, "nombre": Unknown word.
6      precio SMALLINT NOT NULL, "precio": Unknown word.
7      fecha_de_ingreso TIMESTAMP "fecha": Unknown word.
8  );
9
10 CREATE TABLE facturas ( "facturas": Unknown word.
11     id INTEGER PRIMARY KEY AUTOINCREMENT, "AUTOINCREMENT": Unknown word
12     numero_de_factura SMALLINT NOT NULL, "numero": Unknown word.
13     fecha_de_compra TIMESTAMP, "fecha": Unknown word.
14     correo_del_comprador VARCHAR, "correo": Unknown word.
15     telefono_del_comprador INT(12), "telefono": Unknown word.
16     codigo_de_empleado_cajero INT(5), "codigo": Unknown word.
17     monto_total SMALLINT "monto": Unknown word.
18 );
19
20 CREATE TABLE carrito_de_compras ( "carrito": Unknown word.
21     id_carrito INTEGER PRIMARY KEY AUTOINCREMENT, "carrito": Unknown word
22     correo_usuario VARCHAR "correo": Unknown word.
23 );
```

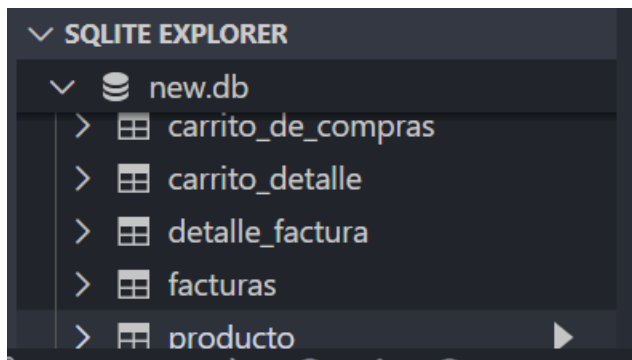
YA ESTA CORREGIDO FACTURAS

```
CREATE TABLE facturas ( "facturas": Unknown word.
    id INTEGER PRIMARY KEY AUTOINCREMENT, "AUTOINCREMENT": Unknown word.
    numero_de_factura SMALLINT NOT NULL, "numero": Unknown word.
    fecha_de_compra TIMESTAMP, "fecha": Unknown word.
    correo_del_comprador VARCHAR, "correo": Unknown word.
    telefono_del_comprador VARCHAR(20), "telefono": Unknown word.
    codigo_de_empleado_cajero INT(5), "codigo": Unknown word.
    monto_total SMALLINT "monto": Unknown word.
);
```

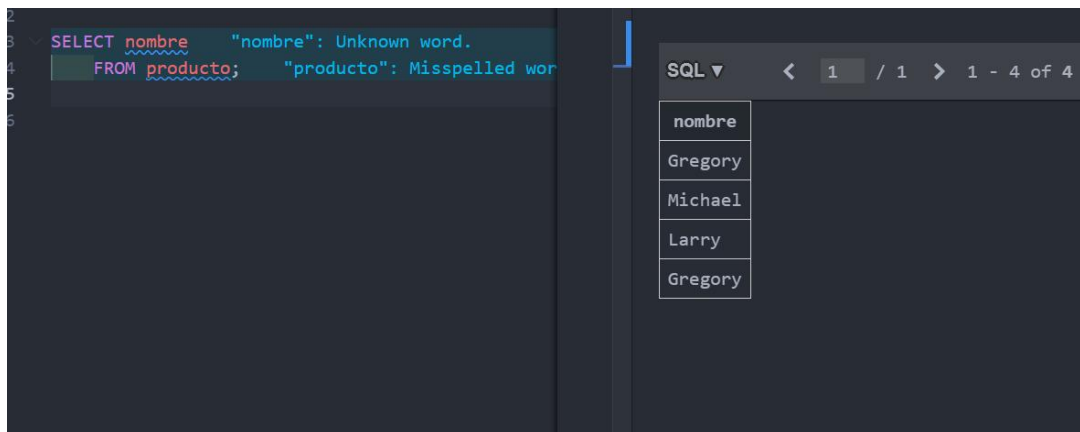
```

32
33 CREATE TABLE carrito_detalle ( "carrito": Unknown word.
34     id_carrito_detalle INTEGER PRIMARY KEY AUTOINCREMENT, "carrito": Unkn
35     id_carrito INT(4) REFERENCES carrito_de_compra(id_carrito), "carrito"
36     codigo_producto INT(10) REFERENCES producto(codigo), "codigo": Unknow
37     cantidad SMALLINT "cantidad": Unknown word.
38 );
39

```



4. Realice los siguientes SELECT:
- Obtenga todos los productos almacenados



(Hagamos de cuenta que esos nombres son productos, es que me confundí y puse nombre de personas)

Obtenga todos los productos que tengan un precio mayor a 50000

```
SELECT nombre, precio
FROM producto
WHERE precio > 50000;
```

nombre	precio
Michael	60000

CORREGIDO LA FORMA DE OBTENER LOS PRODUCTOS CON PRECIO MAYOR A 50000

Obtenga todas las compras de un mismo producto por id.

```
SELECT id_carrito_detalle, carrito, codigo_producto
FROM carrito_detalle
WHERE codigo_producto = 'A1234567';
```

id_carrito_detalle	carrito	codigo_producto	cantidad
1	1	A1234567	1
4	2	A1234567	4

Obtenga todas las compras agrupadas por producto, donde se muestre el total comprado entre todas las compras.

```
SELECT codigo_de_producto, SUM(subtotal) AS total_comprado
FROM detalle_factura
GROUP BY codigo_de_producto;
```

codigo_de_producto	total_comprado
A1234564	5000
A1234567	27500
A1234568	60000

ACTUALIZADO

Obtenga todas las facturas realizadas por el mismo comprador

```
SELECT id, numero_de_factura, monto_total, correo_del_comprador
FROM facturas
WHERE correo_del_comprador = 'example@outlook.com'
```

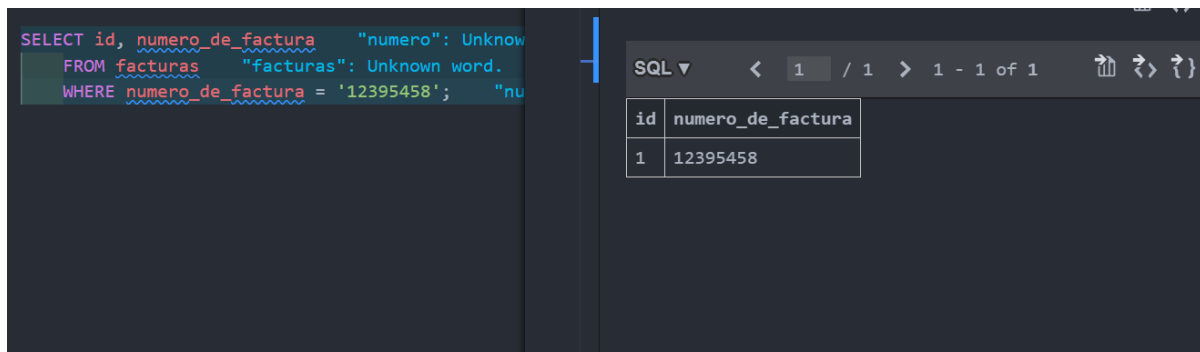
id	numero_de_factura	monto_total	correo_del_comprador
1	12395458	2500	example@outlook.com
4	12395499	25000	example@outlook.com

Obtenga todas las facturas ordenadas por monto total de forma descendente

```
SELECT id, numero_de_factura, monto_total
FROM facturas
ORDER BY monto_total DESC;
```

id	numero_de_factura	monto_total
2	12395457	60000
4	12395499	25000
3	12395450	5000
1	12395458	2500

Obtenga una sola factura por número de factura.



The screenshot shows a SQL IDE interface. On the left, a query is written in a dark-themed editor:

```
SELECT id, numero_de_factura
FROM facturas
WHERE numero_de_factura = '12395458';
```

On the right, the results pane shows a table with two columns: `id` and `numero_de_factura`. The table contains one row with the values `1` and `12395458`.

id	numero_de_factura
1	12395458