

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

↓↓

↓↓

↓↓

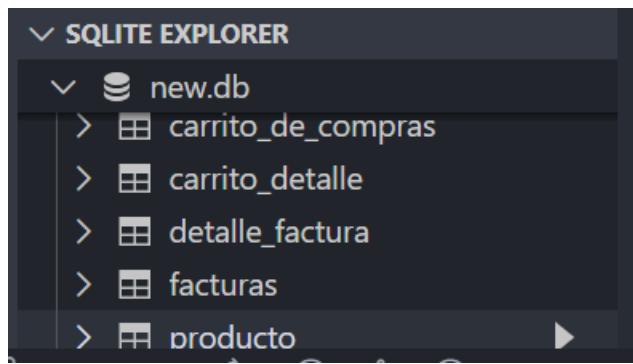
```
-- SQLite
CREATE TABLE producto (    "producto": Misspelled word.
                            id INTEGER PRIMARY KEY AUTOINCREMENT,      "AUTOINCREMENT": Unknown word
                            codigo SMALLINT NOT NULL,      "codigo": Unknown word.
                            nombre VARCHAR,      "nombre": Unknown word.
                            precio SMALLINT NOT NULL,      "precio": Unknown word.
                            fecha_de_ingreso TIMESTAMP      "fecha": Unknown word.
);
;

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 INT(12),      "telefono": Unknown word.
                            codigo_de_empleado_cajero INT(5),      "codigo": Unknown word.
                            monto_total SMALLINT      "monto": Unknown word.
);
;

CREATE TABLE carrito_de_compras (    "carrito": Unknown word.
                            id_carrito INTEGER PRIMARY KEY AUTOINCREMENT,      "carrito": Unknown word
                            correo_usuario VARCHAR      "correo": Unknown word.
);
;

CREATE TABLE detalle_factura (    "detalle": Unknown word.
                            id_detalle INTEGER PRIMARY KEY AUTOINCREMENT,      "detalle": Unknown word
                            numero_de_factura INT REFERENCES facturas(numero_de_factura),      "numero": Unknown word
                            codigo_de_producto INT(5) REFERENCES producto(codigo),      "codigo": Unknown word
                            cantidad SMALLINT,      "cantidad": Unknown word.
                            subtotal SMALLINT      "SMALLINT": Unknown word.
);
;

CREATE TABLE carrito_detalle (    "carrito": Unknown word.
                            id_carrito_detalle INTEGER PRIMARY KEY AUTOINCREMENT,      "carrito": Unknown word
                            id_carrito INT(4) REFERENCES carrito_de_compra(id_carrito),      "carrito": Unknown word
                            codigo_producto INT(10) REFERENCES producto(codigo),      "codigo": Unknown word
                            cantidad SMALLINT,      "cantidad": Unknown word.
);
;
```



4. Realice los siguientes SELECT:

- Obtenga todos los productos almacenados

nombre
Gregory
Michael
Larry
Gregory

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

Obtenga todos los productos que tengan un precio mayor a 50000

nombre	precio > 50000
Gregory	0
Michael	1
Larry	0
Gregory	0

Obtenga todas las compras de un mismo producto por id.

The screenshot shows a MySQL query editor and a results viewer. The query is:

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

The results table has four columns: id_carrito_detalle, carrito, codigo_producto, and cantidad. The data is:

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.

The screenshot shows two MySQL queries and their results. The first query is:

```
SELECT id_carrito_detalle, carrito, codigo_producto
FROM carrito_detalle;
```

The second query is:

```
SELECT id, numero_de_factura, monto_total
FROM facturas;
```

The results table has four columns: id_carrito_detalle, carrito, codigo_producto, and cantidad. The data is:

id_carrito_detalle	carrito	codigo_producto	cantidad
1	1	A1234567	1
2	2	A1234568	24
3	1	A1234564	2
4	2	A1234567	4

The second results table has three columns: id, numero_de_factura, and monto_total. The data is:

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

Obtenga todas las facturas realizadas por el mismo comprador

The screenshot shows a MySQL query and its results. The query is:

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

The results table has four columns: id, numero_de_factura, monto_total, and correo_del_comprador. The data is:

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 "nu  
FROM facturas "facturas": Unknown word.  
ORDER BY monto_total DESC; "monto": Unkn
```

SQL ▼ 1 / 1 1 - 4 of 4

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.

```
SELECT id, numero_de_factura "numero": Unknow  
FROM facturas "facturas": Unknown word.  
WHERE numero_de_factura = '12395458'; "nu
```

SQL ▼ 1 / 1 1 - 1 of 1 ⌂ ⌃ ⌄

id	numero_de_factura
1	12395458