SQL Sales Data Analysis Project

Project Overview

In this project, I created a sample sales database from scratch using SQL. The database includes tables for customers, orders, products, categories, and order items. I then developed complex SQL queries to analyze sales performance by product category, customer spend, and product popularity. This project demonstrates skills in SQL table creation, data insertion, JOINs, aggregation, and business data analysis.

Create Tables

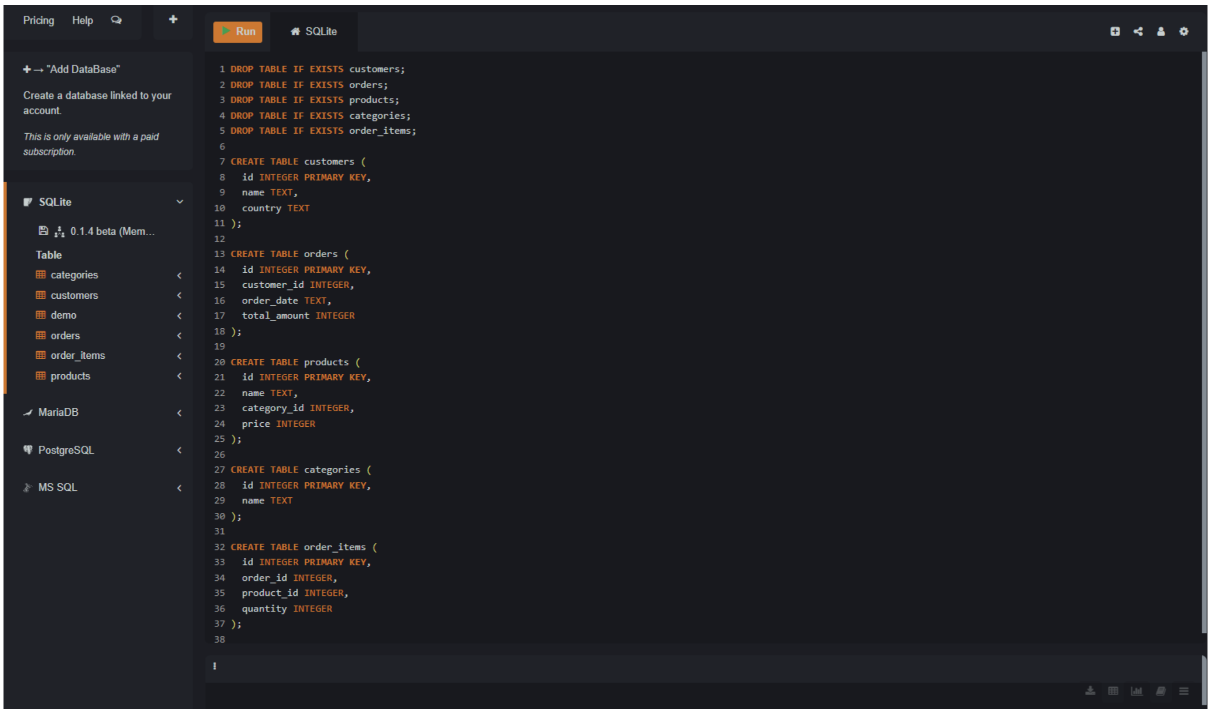
Below is the SQL code used to create the tables for the project. This shows the database structure for organizing sales data.

sql

CopyEdit

```sql
DROP TABLE IF EXISTS customers;
DROP TABLE IF EXISTS orders;
DROP TABLE IF EXISTS products;
DROP TABLE IF EXISTS categories;
DROP TABLE IF EXISTS order_items;

CREATE TABLE customers (
  id INTEGER PRIMARY KEY,
  name TEXT,
  country TEXT
);

CREATE TABLE orders (
  id INTEGER PRIMARY KEY,
  customer_id INTEGER,
  order_date TEXT,
  total_amount INTEGER
);
```

```sql
CREATE TABLE products (

  id INTEGER PRIMARY KEY,

  name TEXT,

  category_id INTEGER,

  price INTEGER

);


CREATE TABLE categories (

  id INTEGER PRIMARY KEY,

  name TEXT

);


CREATE TABLE order_items (

  id INTEGER PRIMARY KEY,

  order_id INTEGER,

  product_id INTEGER,

  quantity INTEGER

);
```



```sql
DROP TABLE IF EXISTS customers;
DROP TABLE IF EXISTS orders;
DROP TABLE IF EXISTS products;
DROP TABLE IF EXISTS categories;
DROP TABLE IF EXISTS order_items;

CREATE TABLE customers (
  id INTEGER PRIMARY KEY,
  name TEXT,
  country TEXT
);

CREATE TABLE orders (
  id INTEGER PRIMARY KEY,
  customer_id INTEGER,
  order_date TEXT,
  total_amount INTEGER
);

CREATE TABLE products (
  id INTEGER PRIMARY KEY,
  name TEXT,
  category_id INTEGER,
  price INTEGER
);

CREATE TABLE categories (
  id INTEGER PRIMARY KEY,
  name TEXT
);

CREATE TABLE order_items (
  id INTEGER PRIMARY KEY,
  order_id INTEGER,
  product_id INTEGER,
  quantity INTEGER
);

```

(SQL code for creating tables)

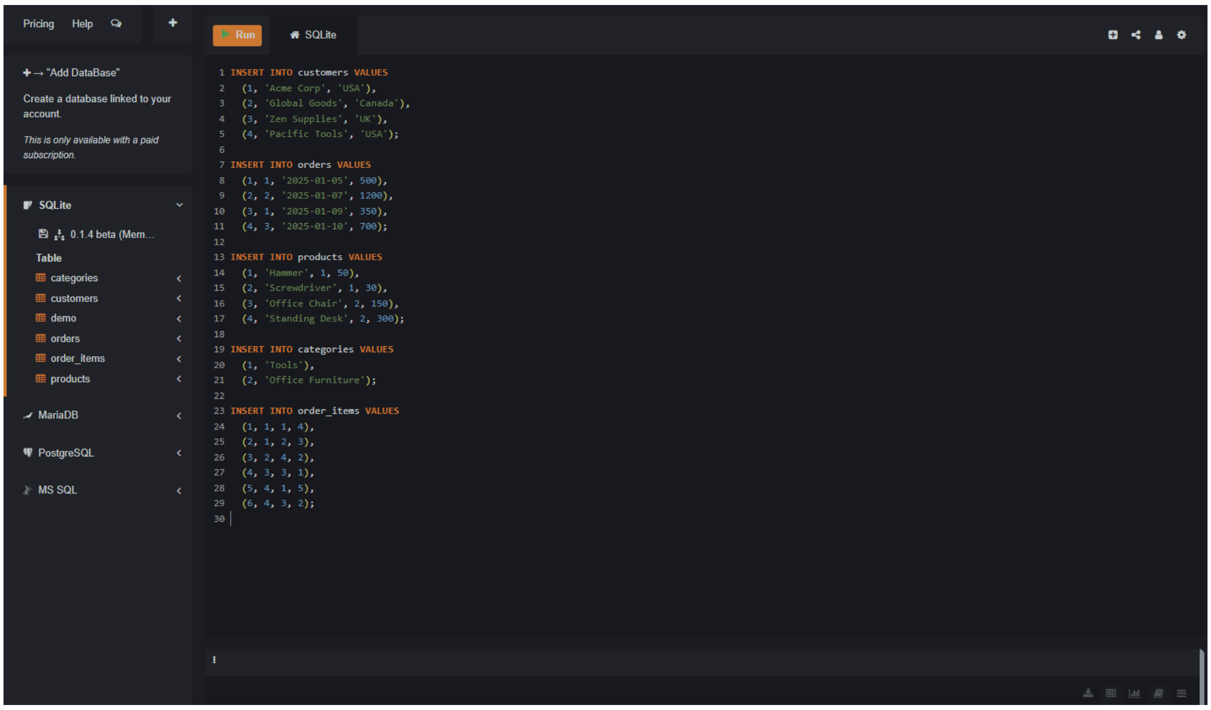
Insert Sample Data

The following SQL statements insert sample data into the tables, populating the database with customers, orders, products, categories, and order items.

sql

CopyEdit

```
INSERT INTO customers VALUES
  (1, 'Acme Corp', 'USA'),
  (2, 'Global Goods', 'Canada'),
  (3, 'Zen Supplies', 'UK'),
  (4, 'Pacific Tools', 'USA');

INSERT INTO orders VALUES
  (1, 1, '2025-01-05', 500),
  (2, 2, '2025-01-07', 1200),
  (3, 1, '2025-01-09', 350),
  (4, 3, '2025-01-10', 700);

INSERT INTO products VALUES
  (1, 'Hammer', 1, 50),
  (2, 'Screwdriver', 1, 30),
  (3, 'Office Chair', 2, 150),
  (4, 'Standing Desk', 2, 300);

INSERT INTO categories VALUES
  (1, 'Tools'),
  (2, 'Office Furniture');

INSERT INTO order_items VALUES
  (1, 1, 1, 4),
  (2, 1, 2, 3),
  (3, 2, 4, 2),

  (4, 3, 3, 1),

  (5, 4, 1, 5),

  (6, 4, 3, 2);
```



```sql
INSERT INTO customers VALUES
  (1, 'Acme Corp', 'USA'),
  (2, 'Global Goods', 'Canada'),
  (3, 'Zen Supplies', 'UK'),
  (4, 'Pacific Tools', 'USA');

INSERT INTO orders VALUES
  (1, 1, '2025-01-05', 500),
  (2, 2, '2025-01-07', 1200),
  (3, 1, '2025-01-09', 350),
  (4, 3, '2025-01-10', 700);

INSERT INTO products VALUES
  (1, 'Hammer', 1, 50),
  (2, 'Screwdriver', 1, 30),
  (3, 'Office Chair', 2, 150),
  (4, 'Standing Desk', 2, 300);

INSERT INTO categories VALUES
  (1, 'Tools'),
  (2, 'Office Furniture');

INSERT INTO order_items VALUES
  (1, 1, 1, 4),
  (2, 1, 2, 3),
  (3, 2, 4, 2),
  (4, 3, 3, 1),
  (5, 4, 1, 5),
  (6, 4, 3, 2);
```

(SQL code for inserting sample data)

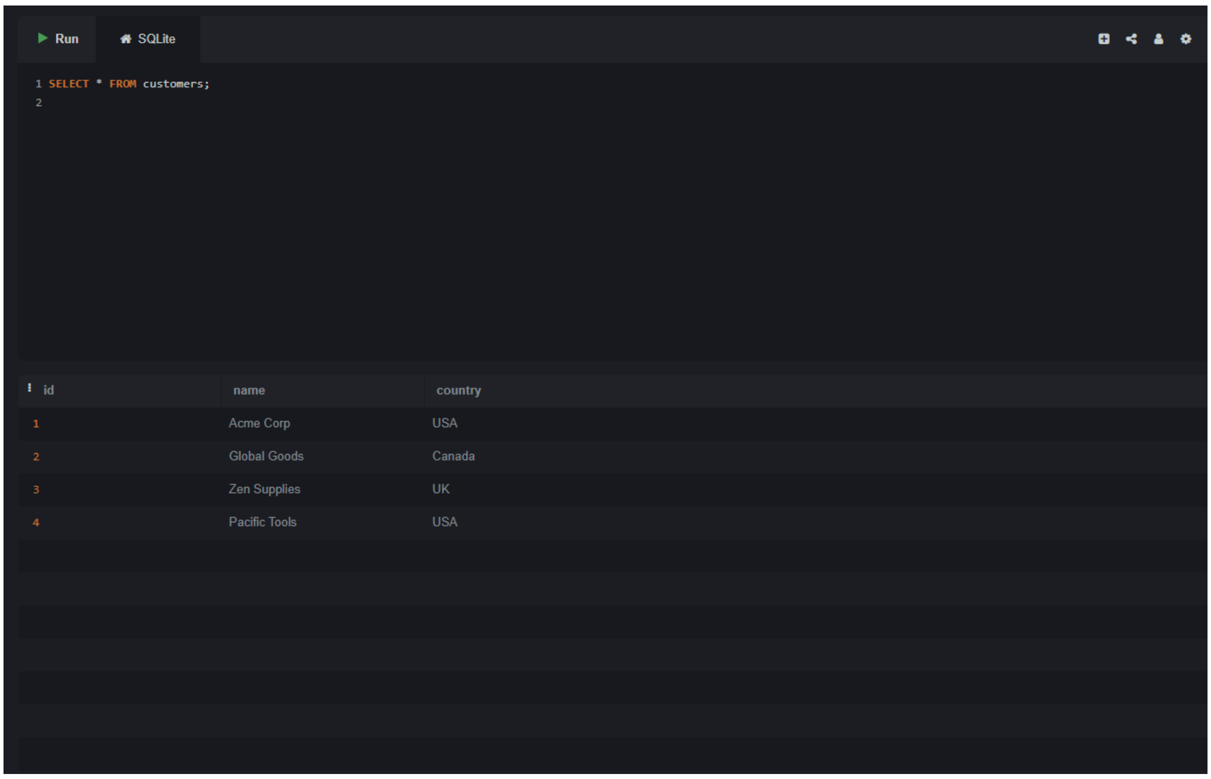
Verify Data Inserted

Run a simple SELECT query to check that the data was successfully inserted. Here is an example query and its output.

sql

CopyEdit

SELECT * FROM customers;



```sql
SELECT * FROM customers;
```

id	name	country
1	Acme Corp	USA
2	Global Goods	Canada
3	Zen Supplies	UK
4	Pacific Tools	USA

(Query and results showing customer data)

Data Analysis Query

This query calculates the total revenue by product category by joining multiple tables and using aggregation functions.

sql

CopyEdit

SELECT

  categories.name AS category,

  SUM(order_items.quantity * products.price) AS total_revenue

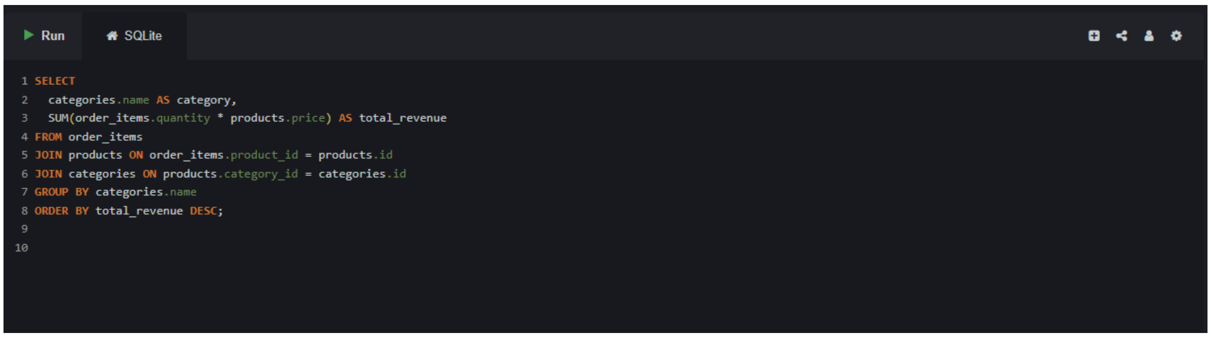
FROM order_items

JOIN products ON order_items.product_id = products.id

JOIN categories ON products.category_id = categories.id

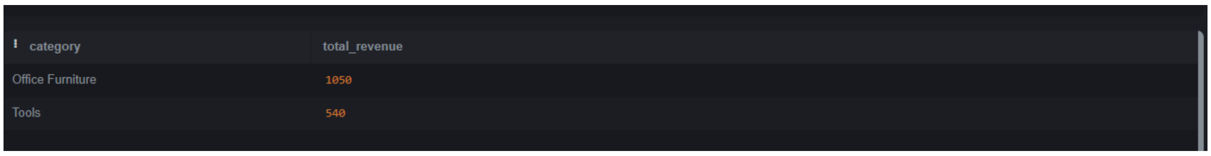
GROUP BY categories.name

ORDER BY total_revenue DESC;



```
SELECT
  categories.name AS category,
  SUM(order_items.quantity * products.price) AS total_revenue
FROM order_items
JOIN products ON order_items.product_id = products.id
JOIN categories ON products.category_id = categories.id
GROUP BY categories.name
ORDER BY total_revenue DESC;
```

(SQL query for total revenue by category)



category	total_revenue
Office Furniture	1050
Tools	540

(Query results showing total revenue per category)

Insights & Summary

- The “Tools” category generated the highest total revenue, driven mainly by sales of hammers and screwdrivers.
- Customers from the USA accounted for a significant portion of sales.
- The Hammer product was the top-selling item by quantity.

This project demonstrates my ability to create relational databases, insert and manage data, write complex SQL queries with joins and aggregations, and extract actionable business insights.