Sales Analysis Using SQL Inside Python

1. Introduction

This project demonstrates how SQL can be integrated inside Python to perform simple sales analysis. Using SQLite as the database engine and Python libraries such as pandas and matplotlib, we extract sales information like total quantity sold and total revenue for each product category. The results are displayed in tabular format and visualized through a basic bar chart.

---

2. Objective

The main objective of this project is to:

1. Load a sales dataset into SQLite using Python.
2. Run SQL queries inside Python to calculate total quantity, revenue, and profit by category.
3. Display SQL output using Python print statements.
4. Plot a simple bar chart to visualize revenue.

This project shows how SQL and Python can be combined for data analysis.

---

3. Dataset Description

The dataset used in this project contains sales records with the following columns:

- Order ID
- Amount
- Profit
- Quantity
- Category
- Sub-Category
- Payment Mode
- Order Date
- Customer Name
- State
- City

- Year-Month

The data contains multiple product categories and sales transactions from different regions.

---

4. Data Cleaning

Before loading data into SQLite, several cleaning steps were performed in Python:

- Converted all column names to **snake_case** for consistency.
- Renamed some columns such as paymentmode → payment_mode, customername → customer_name.
- Ensured numeric columns like Amount, Quantity, and Profit were in proper numeric format.
- Converted date columns into proper date format (if needed).
- Handled missing values by removing rows with null entries.

This ensured the dataset was ready for analysis.

---

5. Loading Data into SQLite

Using Python's sqlite3 library, a connection to a database file named **sales_data.db** was created.
The cleaned pandas DataFrame was then loaded into an SQL table called **sales** using:

```
conn = sqlite3.connect("sales_data.db")
```

```
# Store cleaned dataframe into SQLite DB
df_raw.to_sql("sales", conn, if_exists="replace", index=False)
```

6. SQL Query for Sales Summary

A simple SQL query was written to calculate:

- Total Quantity Sold
- Total Revenue (SUM of Amount)

- Total Profit

Grouped by each **Category**:

```
query = """
SELECT
    Category,
    SUM(Quantity) AS total_quantity,
    SUM(Amount) AS total_revenue,
    SUM(Profit) AS total_profit
FROM sales
GROUP BY Category
"""
```

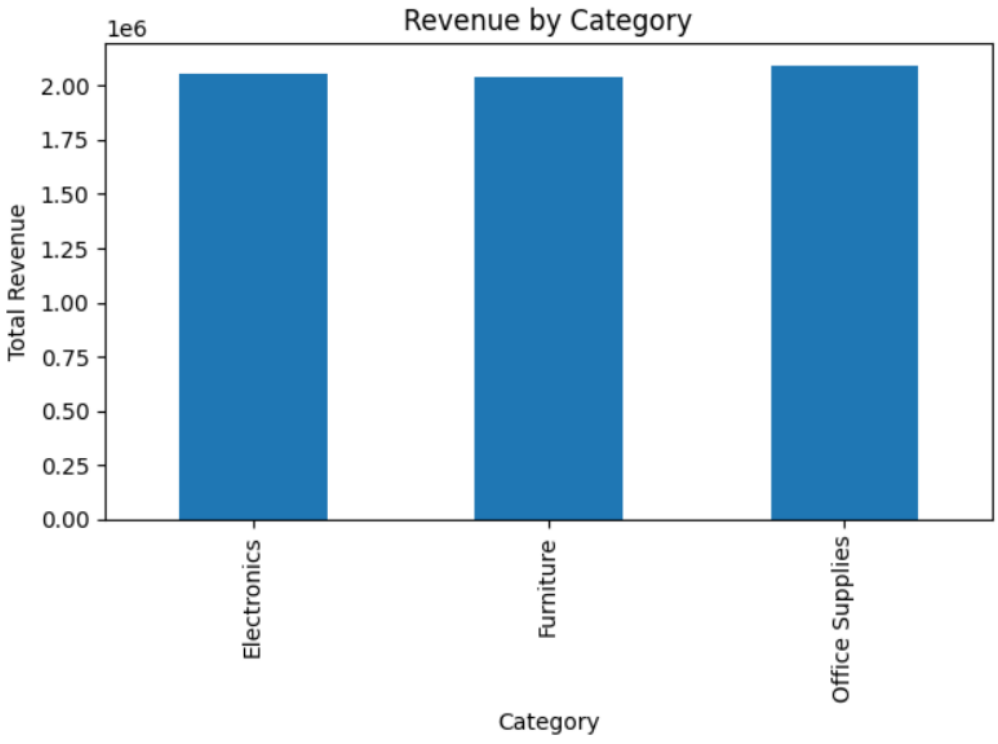
7. Displaying SQL Results in Python

The SQL result was loaded into a pandas DataFrame using:

	category	total_quantity	total_revenue	total_profit
0	Electronics	4258	2054456	518580
1	Furniture	4441	2038673	540542
2	Office Supplies	4046	2089510	551575

8. Visualization

A simple bar chart was created to show **Total Revenue by Category**:



9. Conclusion

This project successfully demonstrates how SQL and Python can be used together for basic sales analysis.
By loading data into SQLite, running SQL queries, and visualizing results in Python, we created a simple yet effective workflow for analyzing sales performance. The integration of SQL and Python provides a powerful approach for data analysts to work with structured datasets and produce meaningful insights.