TASK 7: Get Basic Sales Summary from a Tiny SQLite Database using Python - Answers

Objective: Use SQL inside Python to pull simple sales info (total quantity sold, total revenue) and display it using print statements and a simple bar chart.

Tools: Python (sqlite3, pandas, matplotlib), SQLite (built into Python), Jupyter Notebook or .py file

Deliverables / Steps:

1. Create a small SQLite database file (sales_data.db) with one table named 'sales'.
2. Connect to the database from Python using sqlite3.
3. Run SQL queries to compute total quantity and revenue grouped by product.
4. Load the query result into pandas and print the DataFrame.
5. Plot a simple bar chart showing revenue by product using matplotlib.
6. (Optional) Save the chart to a file.

Example Python script (complete):

```
# 1) Create example SQLite DB and insert sample data (run once)
import sqlite3
conn = sqlite3.connect('sales_data.db')
cur = conn.cursor()
cur.execute('''
CREATE TABLE IF NOT EXISTS sales (
order_id INTEGER PRIMARY KEY AUTOINCREMENT,
order_date TEXT,
product TEXT,
quantity INTEGER,
price REAL
)
''')
# sample rows
cur.executemany('INSERT INTO sales (order_date, product, quantity, price) VALUES
(?, ?, ?, ?)',
[('2025-01-05', 'A', 2, 100.0),
('2025-01-07', 'B', 1, 200.0),
('2025-02-10', 'A', 1, 100.0),
('2025-02-15', 'C', 5, 50.0)])
conn.commit()
conn.close()
# 2) Connect and run SQL using pandas
import sqlite3
import pandas as pd
conn = sqlite3.connect('sales_data.db')
query = '''
SELECT product,
SUM(quantity) AS total_qty,
SUM(quantity * price) AS revenue
FROM sales
GROUP BY product
ORDER BY revenue DESC;
'''
df = pd.read_sql_query(query, conn)
print(df)
# 3) Simple bar chart
import matplotlib.pyplot as plt
```

```
df.plot(kind='bar', x='product', y='revenue', legend=False)
plt.title('Revenue by Product')
plt.xlabel('Product')
plt.ylabel('Revenue')
plt.tight_layout()
plt.savefig('sales_chart.png') # optional
plt.show()
conn.close()
```

Interview Questions & Answers

How did you connect Python to a database?

Use the sqlite3 module: conn = sqlite3.connect('sales_data.db') and then use cursor or pandas.read_sql_query.

What SQL query did you run?

SELECT product, SUM(quantity) AS total_qty, SUM(quantity * price) AS revenue FROM sales GROUP BY product;

What does GROUP BY do?

It groups rows that have the same values in specified columns so aggregate functions (SUM, COUNT) are computed per group.

How did you calculate revenue?

Revenue per row = quantity * price. Aggregate revenue = SUM(quantity * price) grouped by product.

How did you visualize the result?

Loaded SQL result into a pandas DataFrame and used df.plot(kind='bar', x='product', y='revenue') with matplotlib.

What does pandas do in your code?

pandas.read_sql_query loads SQL query results into a DataFrame for easy manipulation and plotting.

What's the benefit of using SQL inside Python?

Combines SQL's efficient grouping/filtering with Python's analysis and plotting libraries for end-to-end reproducible workflows.

Could you run the same SQL query directly in DB Browser for SQLite?

Yes. The same query can be run in DB Browser for SQLite or any SQLite client; results will match.