Basic Sales Summary

Task 7: Using Python, SQL, and Pandas for Data Analysis

Data Analyst Internship Project

Project Objective

Query Data

Use SQL within a Python script to pull simple sales information (total quantity and revenue) from an SQLite database.

Display Results

Show the summarized data clearly using basic print statements in the console.

Visualize Findings

Create a simple bar chart to visually represent the sales revenue by product.

The Technology Stack

Python & SQLite3

The core language and the sqlite3 module for database connection and query execution.

Pandas

Used to load the SQL query results directly into a powerful and easy-to-use DataFrame.

Matplotlib

The visualization library used (via Pandas) to plot the final df.plot(kind='bar') chart.

Dataset: The sales Table

Column	Type	Description
product	TEXT	The name of the product sold.
quantity	INTEGER	The number of units sold.
price	REAL	The price per unit.

A simple, single-table database was created to simulate a real-world sales ledger.

The "Heart" of the Code: The SQL Query

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

Key Functions

`SUM(quantity * price)`: This calculates the total revenue for each item before summing it for the group.

`GROUP BY product`: This is the key. It collapses all rows for the same product into one, allowing the SUM functions to operate on each unique product.

Python + Pandas Workflow

Step 1: Connect

```
sqlite3.connect()
```

Step 2: Define Query

```
query = "SELECT..."
```

Step 3: Execute & Load

```
pd.read_sql_query()
```

Step 4: Display & Plot

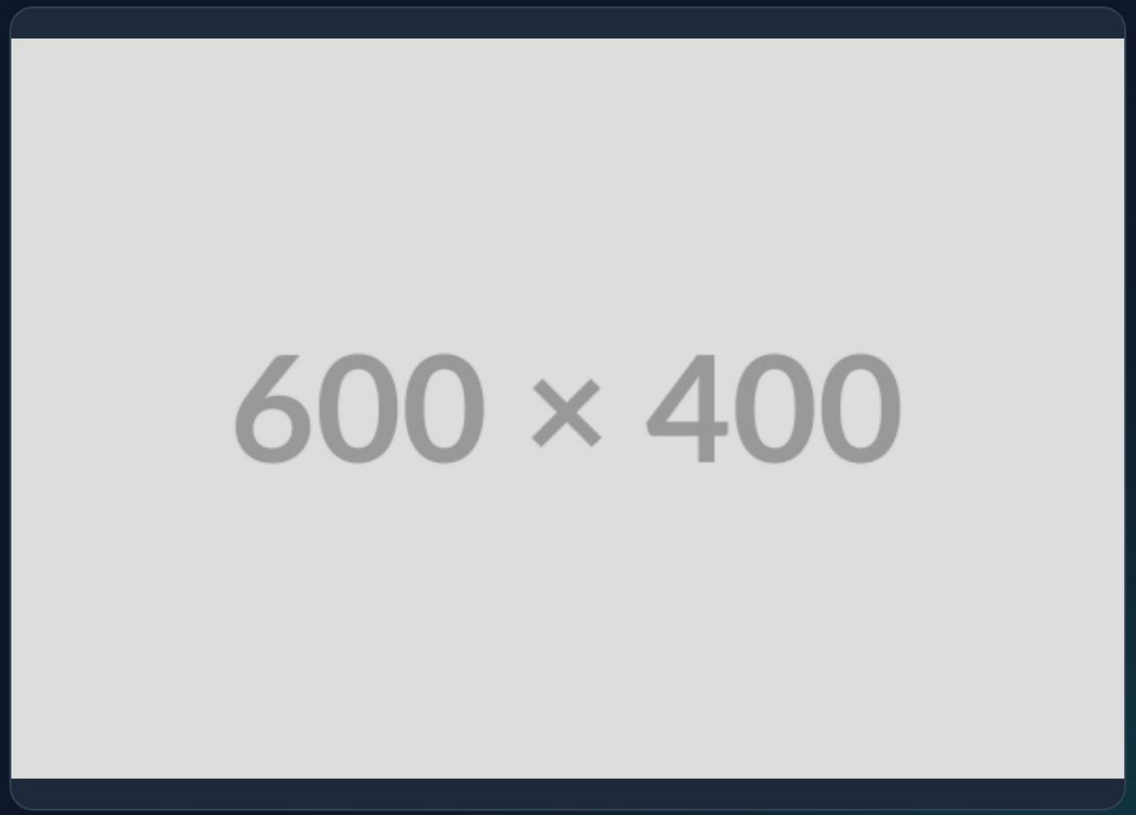
```
print(df) & df.plot()
```

Deliverable 1: Console Output

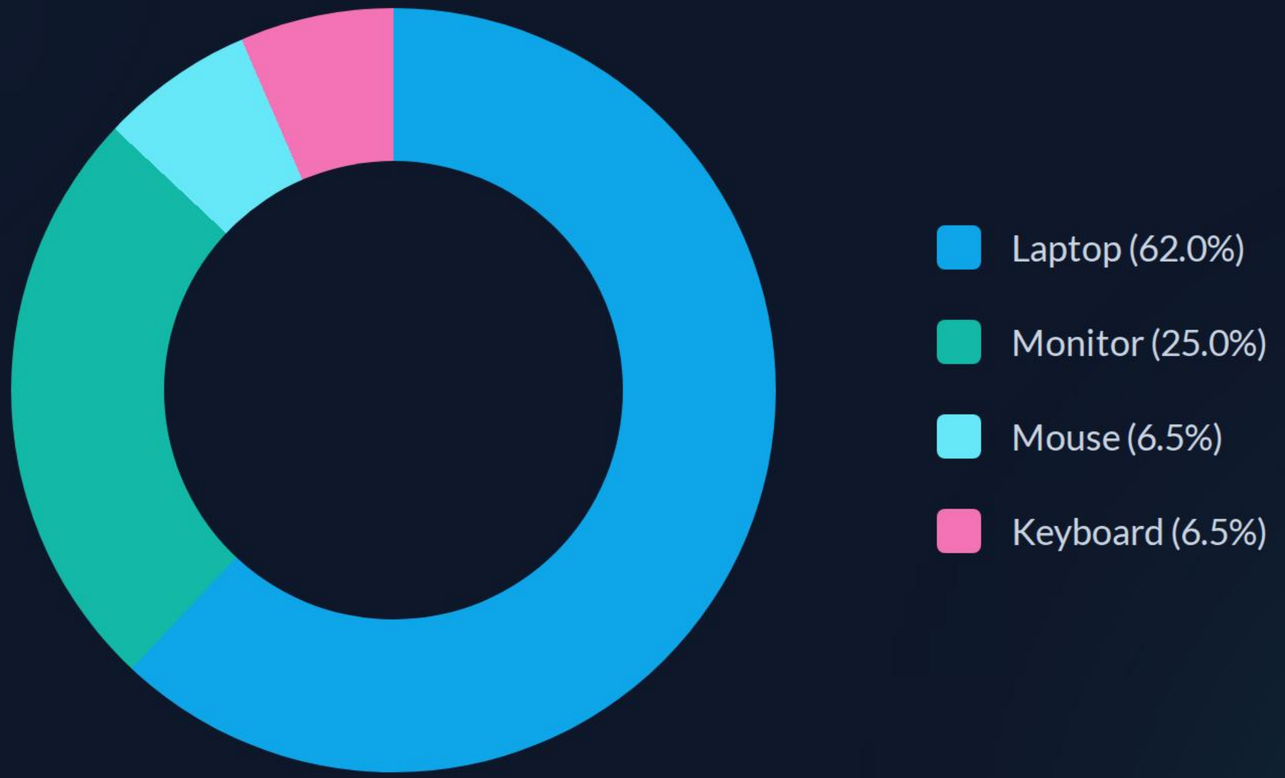
Summarized Data

The first deliverable was a clean, summarized table printed directly to the console.

The `pd.read_sql_query` function made it simple to load the SQL results directly into a formatted Pandas DataFrame, which is easy to read and manipulate further.



Deliverable 2: Sales Revenue Chart



The chart clearly visualizes the summarized data, making it easy to see that "Laptop" is the highest-grossing product.

Project Outcomes & Key Learnings

- **SQL Querying:** Practiced writing basic SQL aggregate queries (SUM, GROUP BY).
- **Python-SQL Integration:** Learned how to connect, query, and import data from a database within a single Python script.
- **Data Summarization:** Performed a simple but complete ETL (Extract, Transform, Load) and summarization task.
- **Data to Visualization:** Mastered the complete pipeline from a raw database table to a final sales chart.

Key Interview Questions

What's the benefit of using SQL inside Python?

It combines SQL's power for efficient data querying with Python's rich ecosystem for analysis, visualization (Matplotlib), and app development.

What does GROUP BY do?

It groups rows that have the same values in specified columns into summary rows. It's essential for aggregate functions like SUM() or COUNT() to work on each group.

Project Submission

- **GitHub Repository:** A new repository was created for the task.
- **Code & Assets:** Submitted all Python scripts (.py), notebook files (.ipynb), and saved chart images (.png).
- **README.md:** A short README.md file was included to explain the project and how to run the code.

Thank You

Questions?