Step 1: Environment Setup

Step one was creating the development environment. The fundamental programming language used was Python, and we utilized three fundamental libraries:

- sqlite3: A standard Python module that permits us to connect with SQLite databases.
- pandas: To import SQL query results into structured tables (DataFrames) for easier manipulation.
- matplotlib: To plot the sales summary as a bar chart.

Prior to actually coding, installing `pandas` and `matplotlib` via pip was necessary while `sqlite3` was installed along with Python.

Step 2: Create the Database and Sales Table

Here, we employed Python to establish a new SQLite database `sales_data.db`. Next, we defined a table `sales` with columns `product`, `quantity`, and `price`. Finally, we inserted some rows of sample data to represent various product sales. Each row described a product name, the quantity sold, and price per unit.

This step helped simulate real-world scenarios where data is stored in relational databases, which are commonly used in business environments.

Step 3: Querying the Sales Data Using SQL

After creating and populating the database, we wrote an SQL query in Python to summarize the sales data. The query calculated:

* The total quantity of each product sold (`SUM(quantity)`)

* The total revenue per product (`SUM(quantity * price)`)

The `GROUP BY` clause was utilized to aggregate the results based on each product. The SQL query was run utilizing the `pandas.read_sql_query()` method, and the outcome was saved in a DataFrame. This enabled us to see the data summarized directly as well as ready it for visualization.

This step illustrated just how robust and versatile SQL is when utilized inside Python for data analysis.

Step 4: Visualizing the Results

The last thing to do was make a bar chart to show the revenue each product produced. We did that with `matplotlib`. We graphed `product` along the x-axis and `revenue` along the y-axis. We added a label to the chart and optionally saved it to an image file (`sales_chart.png`).

This visualization gave an instant and concise means to determine which product sold best in revenue, a usual requirement in business reporting.

Conclusion

Having done this exercise, we were able to effectively show how to:

- Link Python to a SQLite database
- Write and execute simple SQL queries within Python
- Employ pandas to process SQL output
- Make a simple sales report and visualization

These skills are the foundation for any prospective data analyst, as they bring database querying together with data manipulation and visualization — essential data-driven decision-making tools of the real world.