**🧾 Task 7 – Sales Summary using SQLite and Python**

**📌 Objective**

The goal of this task was to:

* Connect Python to a SQLite database (sales\_data.db)
* Run SQL queries to get product-wise **total quantity** and **total revenue**
* Load the data into a pandas DataFrame
* Display the results in tabular format using print()
* Visualize the revenue per product using a simple bar chart via matplotlib

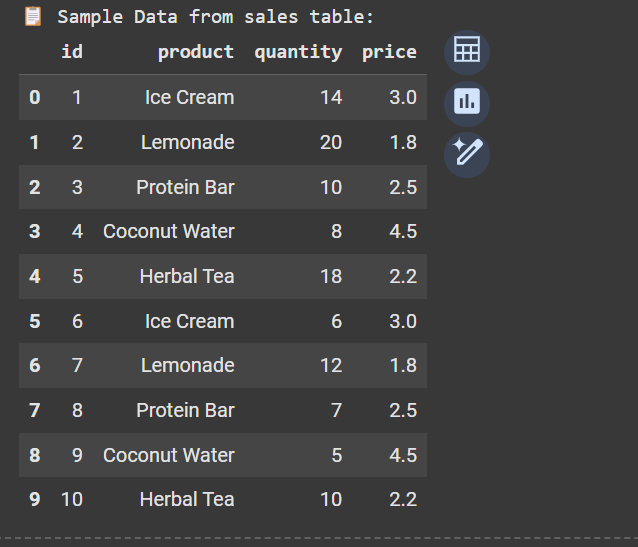
**🛠️ Tools & Libraries Used**

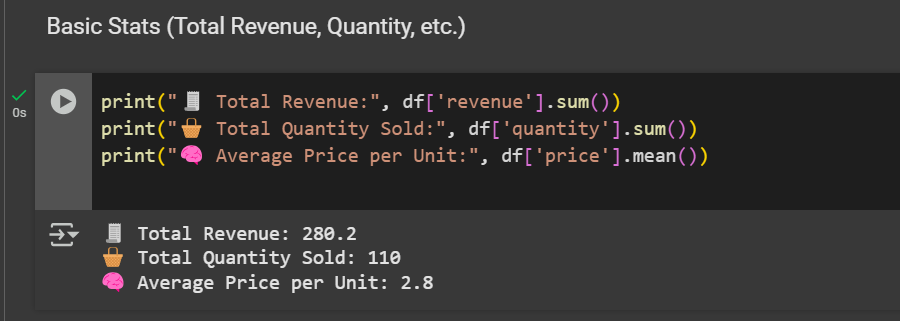
* **Python** – Programming language
* **SQLite** – Lightweight SQL database (no setup needed!)
* **sqlite3** – Python’s built-in library to connect with SQLite
* **pandas** – For data loading and manipulation
* **matplotlib** – For creating a basic bar chart
* **Jupyter Notebook** – For running the code interactively

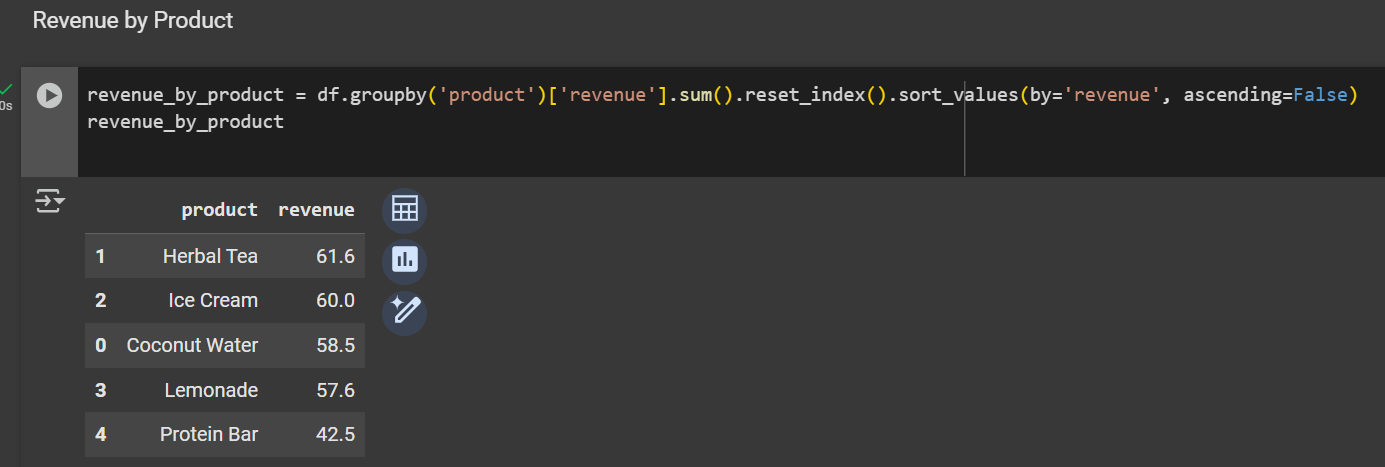
**🧠 What I Did – Step-by-Step**

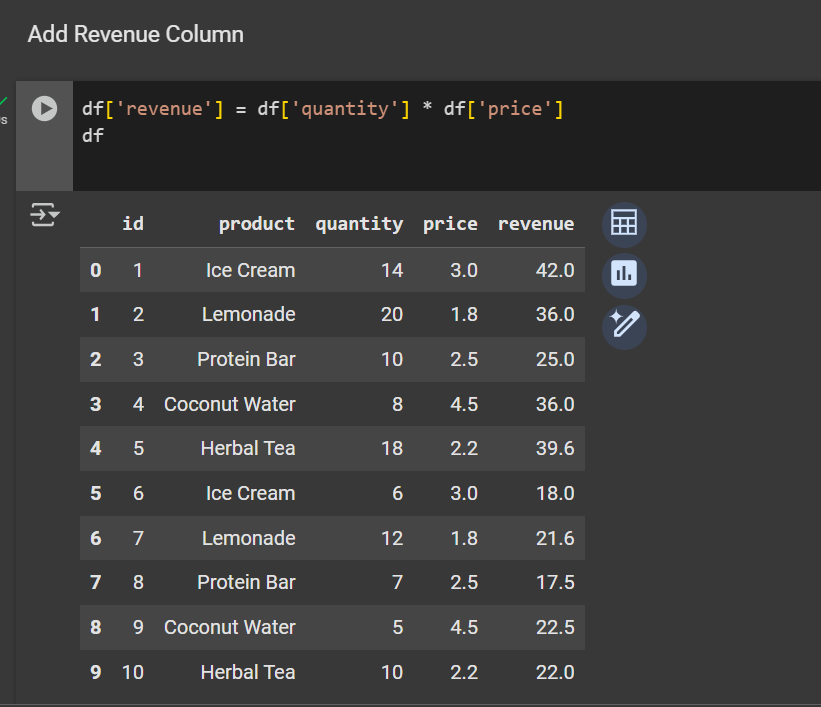
**1️⃣ Created SQLite Database**

A database file named sales\_data.db was used, containing a single table: sales.











**2️⃣ Connected Python to SQLite**

Used sqlite3.connect() to establish a connection:

python

CopyEdit

conn = sqlite3.connect("sales\_data.db")

**3️⃣ Ran SQL Query to Get Summary**

Used SQL to group data by product and calculate total quantity and revenue:

sql

CopyEdit

SELECT

product,

SUM(quantity) AS total\_qty,

SUM(quantity \* price) AS revenue

FROM sales

GROUP BY product

**4️⃣ Loaded Result into Pandas**

Used pd.read\_sql\_query() to fetch query results directly into a DataFrame:

python

CopyEdit

df = pd.read\_sql\_query(query, conn)

print(df)

**5️⃣ Plotted a Bar Chart**

Used matplotlib to create a bar chart of revenue by product:

python

CopyEdit

df.plot(kind='bar', x='product', y='revenue')

plt.title("Revenue by Product")

plt.xlabel("Product")

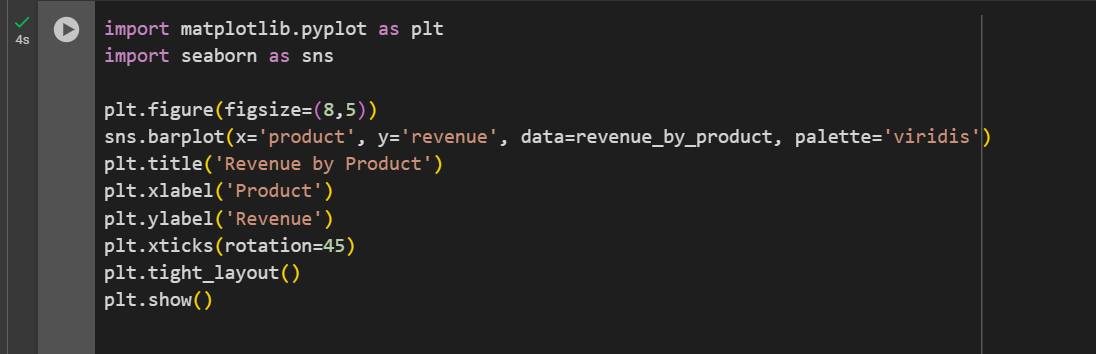
plt.ylabel("Revenue")

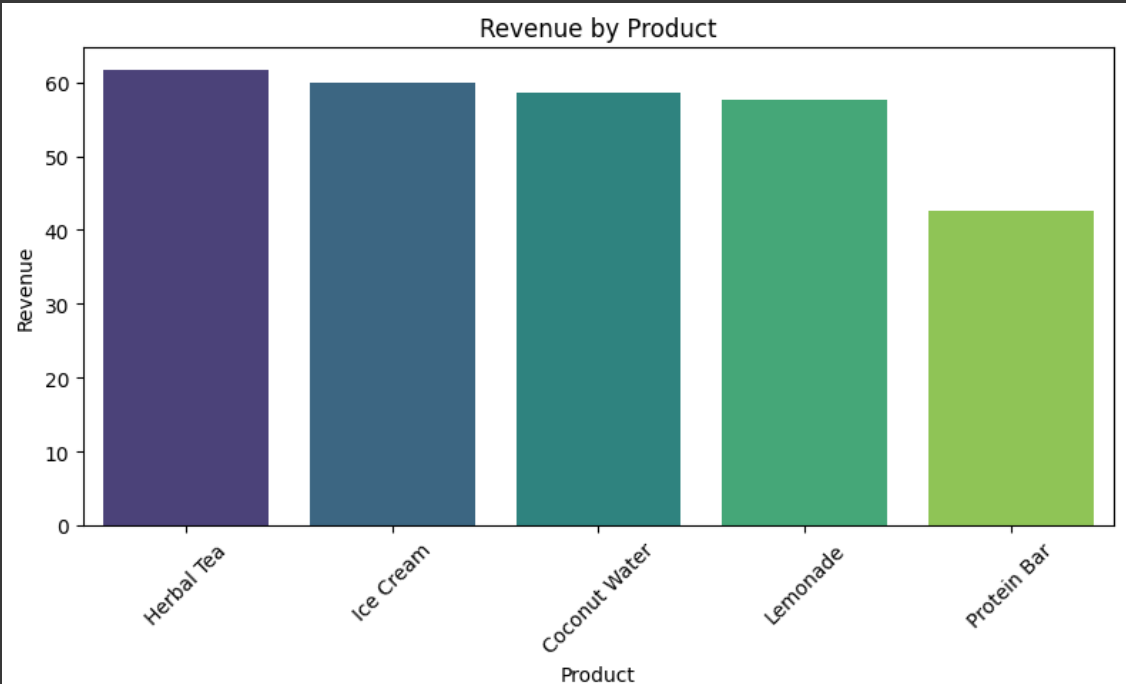
plt.tight\_layout()

plt.show()

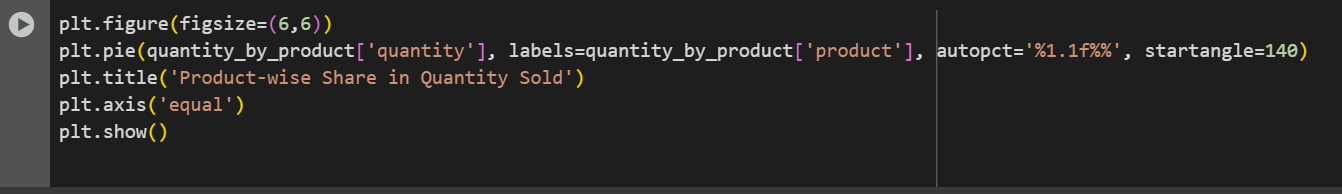
**✅ Visualization**

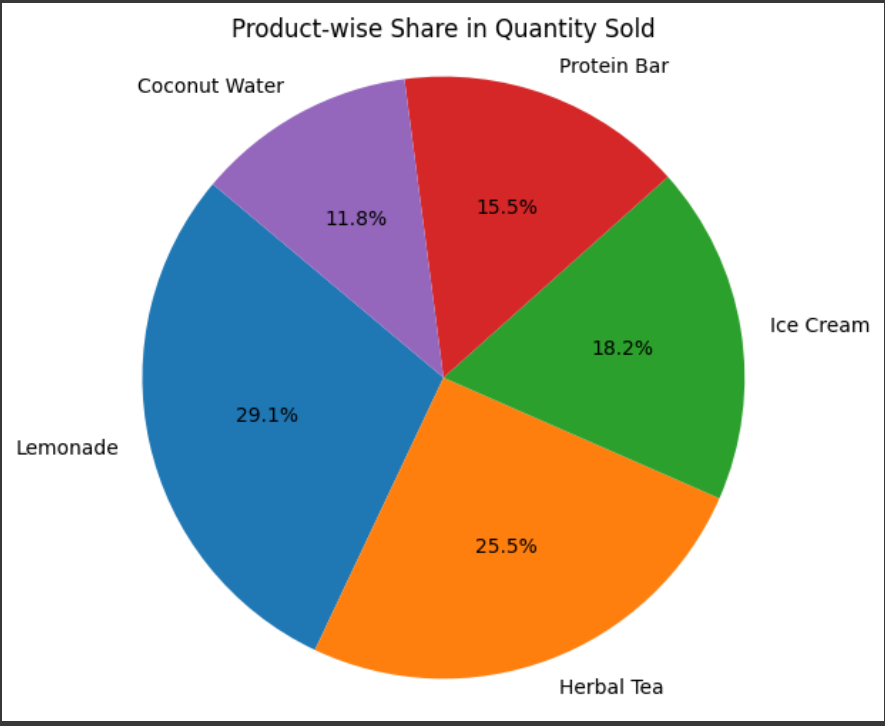
**📊 1. Bar Plot – Revenue by Product**

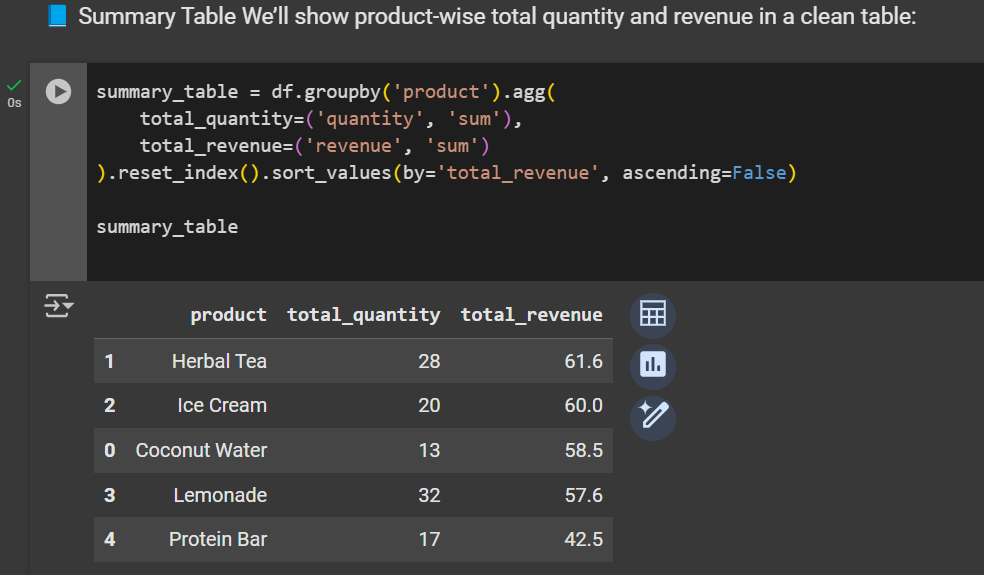
****

****

**🧁 2. Pie Chart – Product-wise Share in Total Quantity Sold**

****

****

****