

```
[7]: import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
import random
from datetime import datetime, timedelta

# Step 1: Connect to SQLite database (creates file if not exists)
conn = sqlite3.connect("sales_data.db")
cursor = conn.cursor()

# Step 2: Create sales table if it doesn't exist
cursor.execute("""
CREATE TABLE IF NOT EXISTS sales (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    product TEXT,
    quantity INTEGER,
    price REAL,
    sale_date TEXT
)
""")
```

```
[4]: # Step 3: Check if data exists, if not insert 100 random rows
cursor.execute("SELECT COUNT(*) FROM sales")
if cursor.fetchone()[0] == 0:
    products = ["Laptop", "Phone", "Tablet", "Headphones", "Monitor", "Keyboard", "Mouse"]
    for _ in range(100):
        product = random.choice(products)
        quantity = random.randint(1, 10)
        price = round(random.uniform(50, 1500), 2)
        sale_date = datetime.now() - timedelta(days=random.randint(0, 365))
        cursor.execute(
            "INSERT INTO sales (product, quantity, price, sale_date) VALUES (?, ?, ?, ?)",
            (product, quantity, price, sale_date.strftime("%Y-%m-%d"))
        )
    conn.commit()
    print("Inserted 100 random sales records.")
```

Inserted 100 random sales records.



```
# Step 4: Run SQL queries

# 4.1 Total quantity and revenue per product
query_summary = """
SELECT product,
       SUM(quantity) AS total_qty,
       SUM(quantity * price) AS revenue
FROM sales
GROUP BY product
"""

df_summary = pd.read_sql_query(query_summary, conn)

# 4.2 Month-wise revenue
query_month = """
SELECT strftime('%Y-%m', sale_date) AS month,
       SUM(quantity * price) AS monthly_revenue
FROM sales
GROUP BY month
ORDER BY month
"""

df_month = pd.read_sql_query(query_month, conn)
```



```
# Step 4: Run SQL queries

# 4.1 Total quantity and revenue per product
query_summary = """
SELECT product,
       SUM(quantity) AS total_qty,
       SUM(quantity * price) AS revenue
FROM sales
GROUP BY product
"""
df_summary = pd.read_sql_query(query_summary, conn)

# 4.2 Month-wise revenue
query_month = """
SELECT strftime('%Y-%m', sale_date) AS month,
       SUM(quantity * price) AS monthly_revenue
FROM sales
GROUP BY month
ORDER BY month
"""
df_month = pd.read_sql_query(query_month, conn)
```