Task-7

I used VS code to do this task and below is the code

Code:-

import sqlite3

import pandas as pd

import matplotlib.pyplot as plt

def create\_database():

    """Create and populate the SQLite database"""

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

    cursor = conn.cursor()

    # Create table

    cursor.execute('''

    CREATE TABLE IF NOT EXISTS sales (

        id INTEGER PRIMARY KEY AUTOINCREMENT,

        product TEXT NOT NULL,

        quantity INTEGER NOT NULL,

        price REAL NOT NULL,

        sale\_date TEXT

    )

    ''')

    # Insert sample data

    sample\_data = [

        ('Laptop', 5, 999.99, '2023-01-15'),

        ('Phone', 10, 699.99, '2023-01-15'),

        ('Tablet', 8, 349.99, '2023-01-16'),

        ('Laptop', 3, 999.99, '2023-01-17'),

        ('Phone', 7, 699.99, '2023-01-18'),

        ('Headphones', 15, 99.99, '2023-01-18'),

        ('Tablet', 5, 349.99, '2023-01-19'),

        ('Headphones', 10, 99.99, '2023-01-20')

    ]

    cursor.executemany('''

    INSERT INTO sales (product, quantity, price, sale\_date)

    VALUES (?, ?, ?, ?)

    ''', sample\_data)

    conn.commit()

    conn.close()

    print("Database created with sample data!\n")

def analyze\_sales():

    """Query and visualize sales data"""

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

    # Main analysis query

    query = """

    SELECT

        product,

        SUM(quantity) AS total\_quantity,

        SUM(quantity \* price) AS revenue,

        ROUND(AVG(price), 2) AS avg\_price,

        COUNT(\*) AS transaction\_count

    FROM sales

    GROUP BY product

    ORDER BY revenue DESC

    """

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

    total\_revenue = df['revenue'].sum()

    df['revenue\_pct'] = (df['revenue'] / total\_revenue \* 100).round(2)

    # Daily sales query

    daily\_query = """

    SELECT

        sale\_date,

        SUM(quantity) AS daily\_quantity,

        SUM(quantity \* price) AS daily\_revenue

    FROM sales

    GROUP BY sale\_date

    ORDER BY sale\_date

    """

    daily\_df = pd.read\_sql\_query(daily\_query, conn)

    conn.close()

    # Print results

    print("=== PRODUCT SALES SUMMARY ===")

    print(df.to\_string(index=False))

    print(f"\nTotal Revenue: ${total\_revenue:,.2f}")

    print("\n=== DAILY SALES ===")

    print(daily\_df.to\_string(index=False))

    # Visualizations

    plt.style.use('ggplot')

    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 6))

    # Product analysis

    df.plot(kind='bar', x='product', y='revenue', ax=ax1,

            title='Revenue by Product', color='teal')

    ax1.set\_ylabel('Revenue ($)')

    df.plot(kind='bar', x='product', y='total\_quantity', ax=ax2,

            title='Units Sold', color='orange')

    ax2.set\_ylabel('Quantity')

    plt.tight\_layout()

    plt.savefig('product\_analysis.png')

    # Daily trend

    plt.figure(figsize=(10, 5))

    daily\_df.plot(kind='line', x='sale\_date', y='daily\_revenue',

                 marker='o', title='Daily Revenue Trend', color='green')

    plt.ylabel('Revenue ($)')

    plt.tight\_layout()

    plt.savefig('daily\_trend.png')

    print("\nCharts saved as 'product\_analysis.png' and 'daily\_trend.png'")

    plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

    create\_database()

    analyze\_sales()

outputs:-





