

TASK 7: Get Basic Sales Summary from a Tiny SQLite Database using Python

Objective: Use SQL inside Python to pull simple sales info (like total quantity sold, total revenue), and display it using basic print statements and a simple bar chart.

Tools:

Python (with sqlite3, pandas, matplotlib) SQLite (built into Python — no setup!) Jupyter Notebook or a .py file

1. Create Your SQLite Database

Let's first create a database called `sales_data.db` and a table named `sales`.

Create a script named `create_db.py` or run in a Jupyter cell:

```
In [1]: import sqlite3

# Connect to SQLite database (it will create the file if not exists)
conn = sqlite3.connect('sales_data.db')
cursor = conn.cursor()

# Create a sales table
cursor.execute('''
    CREATE TABLE IF NOT EXISTS sales (
        id INTEGER PRIMARY KEY,
        product TEXT,
        quantity INTEGER,
        price REAL
    )
''')

# Insert sample sales data
sample_data = [
    ('Apple', 10, 5.0),
    ('Banana', 20, 2.5),
    ('Orange', 15, 4.0),
    ('Apple', 5, 5.0),
    ('Banana', 10, 2.5),
    ('Orange', 10, 4.0),
]

cursor.executemany("INSERT INTO sales (product, quantity, price) VALUES (?, ?, ?)"
conn.commit()
conn.close()
```

2. Python Code to Generate the Big Database

```
In [5]: import sqlite3
import random
import datetime

# Connect to SQLite
conn = sqlite3.connect('sales_data.db')
cursor = conn.cursor()

# Drop existing table if needed
cursor.execute("DROP TABLE IF EXISTS sales")

# Create new table
cursor.execute('''
CREATE TABLE sales (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    date TEXT,
    customer TEXT,
    product TEXT,
    quantity INTEGER,
    price REAL,
    region TEXT
)
''')

# Sample data pools
products = ['Laptop', 'Mouse', 'Keyboard', 'Monitor', 'Tablet', 'Phone']
customers = ['Alice', 'Bob', 'Charlie', 'David', 'Eve']
regions = ['North', 'South', 'East', 'West']

# Generate 500 fake sales records
sales_data = []
start_date = datetime.date(2024, 1, 1)

for _ in range(500):
    date = start_date + datetime.timedelta(days=random.randint(0, 180))
    customer = random.choice(customers)
    product = random.choice(products)
    quantity = random.randint(1, 10)
    price = round(random.uniform(50, 1500), 2)
    region = random.choice(regions)

    sales_data.append((date.isoformat(), customer, product, quantity, price, region))

# Insert into table
cursor.executemany("INSERT INTO sales (date, customer, product, quantity, price, region) VALUES (?, ?, ?, ?, ?, ?)", sales_data)
conn.commit()
conn.close()

print("✅ Big database 'sales_data.db' with 500 records created.")
```

✅ Big database 'sales_data.db' with 500 records created.

```
In [7]: import sqlite3
import pandas as pd

# Connect to the database
```

```
conn = sqlite3.connect('sales_data.db')

# Load all data into a DataFrame
df = pd.read_sql_query("SELECT * FROM sales", conn)

# Print top 10 rows
print("Sample Sales Data:")
print(df.head(10))

# Show the number of rows
print(f"\nTotal records: {len(df)}")

conn.close()
```

Sample Sales Data:

	id	date	customer	product	quantity	price	region
0	1	2024-04-13	Charlie	Phone	3	170.76	West
1	2	2024-05-24	Eve	Laptop	6	558.98	West
2	3	2024-04-27	Bob	Laptop	2	172.38	West
3	4	2024-03-03	Eve	Laptop	10	1410.90	West
4	5	2024-01-19	Bob	Keyboard	4	1449.36	East
5	6	2024-05-09	Eve	Mouse	5	1363.21	East
6	7	2024-01-17	Eve	Monitor	4	991.43	West
7	8	2024-02-29	Bob	Keyboard	2	640.60	West
8	9	2024-05-16	Charlie	Phone	8	618.66	South
9	10	2024-03-31	Eve	Tablet	5	533.50	South

Total records: 500



A. Total Revenue by Product (With Comments)

```
In [11]: import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

# Connect to the SQLite database
conn = sqlite3.connect('sales_data.db')

# SQL to get total revenue per product
query = """
SELECT product, SUM(quantity * price) AS revenue
FROM sales
GROUP BY product
ORDER BY revenue DESC
"""

# Load the result into a DataFrame
df_prod = pd.read_sql_query(query, conn)
print("\nRevenue by Product:")
print(df_prod)

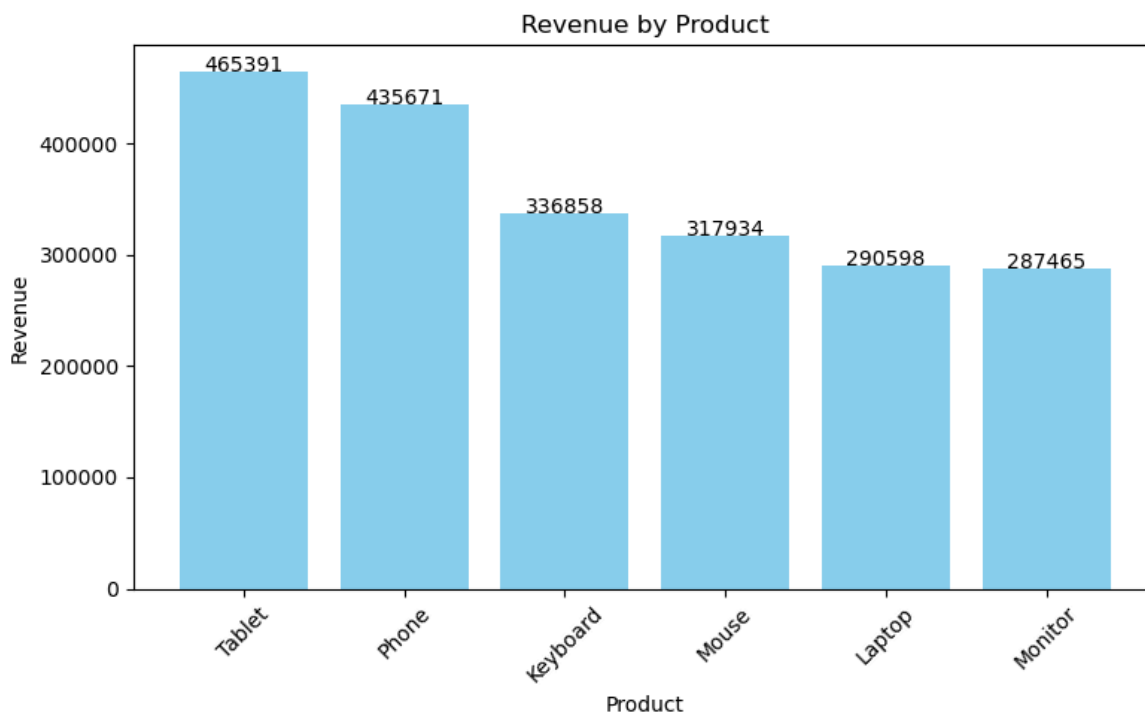
# Plot a bar chart
plt.figure(figsize=(8,5))
plt.bar(df_prod['product'], df_prod['revenue'], color='skyblue')
plt.title('Revenue by Product')
plt.xlabel('Product')
plt.ylabel('Revenue')
plt.xticks(rotation=45)
```

```
# Add value labels on top of bars
for index, value in enumerate(df_prod['revenue']):
    plt.text(index, value + 10, f'{value:.0f}', ha='center')

plt.tight_layout()
plt.show()
```

Revenue by Product:

	product	revenue
0	Tablet	465390.69
1	Phone	435670.62
2	Keyboard	336857.89
3	Mouse	317934.12
4	Laptop	290597.88
5	Monitor	287465.04



B. Total Revenue by Region (With Comments)

```
In [14]: # SQL to get total revenue per region
query = """
SELECT region, SUM(quantity * price) AS revenue
FROM sales
GROUP BY region
ORDER BY revenue DESC
"""

# Load into DataFrame
df_region = pd.read_sql_query(query, conn)
print("\nRevenue by Region:")
print(df_region)

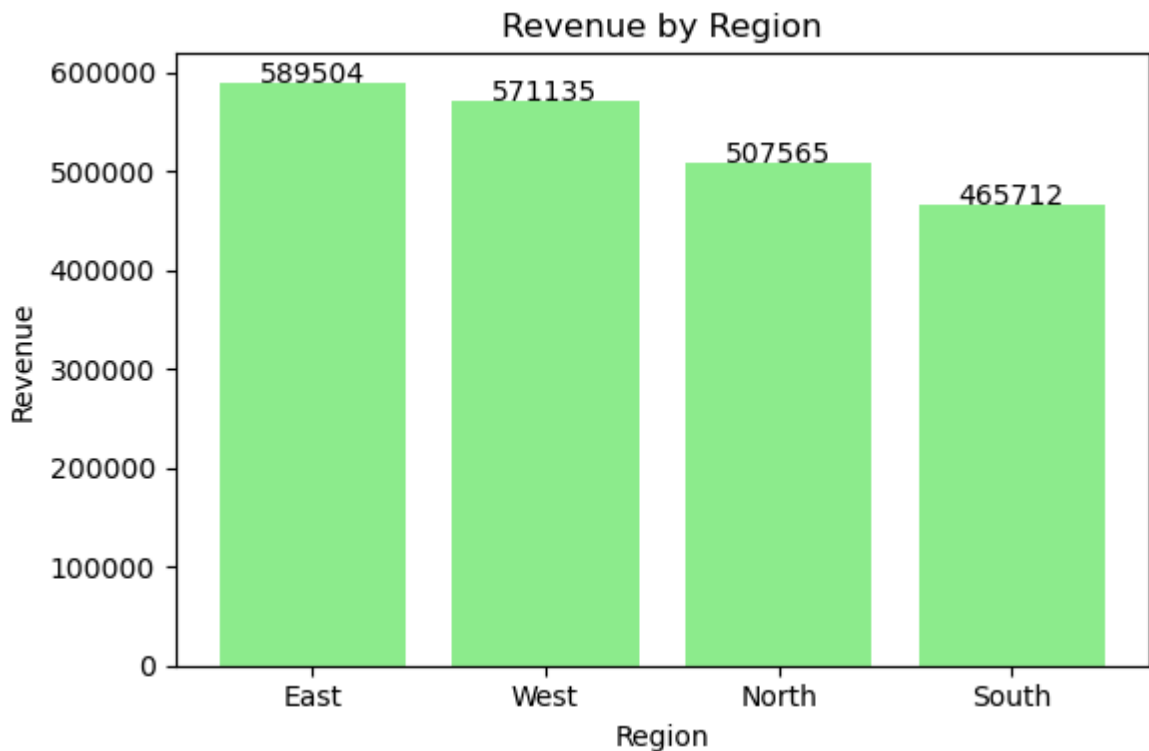
# Bar chart for region revenue
plt.figure(figsize=(6,4))
plt.bar(df_region['region'], df_region['revenue'], color='lightgreen')
plt.title('Revenue by Region')
plt.xlabel('Region')
plt.ylabel('Revenue')
```

```
# Add data labels on bars
for index, value in enumerate(df_region['revenue']):
    plt.text(index, value + 10, f'{value:.0f}', ha='center')

plt.tight_layout()
plt.show()
```

Revenue by Region:

	region	revenue
0	East	589504.38
1	West	571134.64
2	North	507565.37
3	South	465711.85



C. Monthly Revenue Trend (Line Chart with Comments)

```
In [17]: # SQL to group revenue by month
query = """
SELECT
    strftime('%Y-%m', date) AS month,
    SUM(quantity * price) AS revenue
FROM sales
GROUP BY month
ORDER BY month
"""

# Load into DataFrame
df_month = pd.read_sql_query(query, conn)
print("\nMonthly Revenue Trend:")
print(df_month)

# Line chart for monthly revenue
plt.figure(figsize=(10,5))
```

```
plt.plot(df_month['month'], df_month['revenue'], marker='o', linestyle='-', color='orange')
plt.title('Monthly Revenue Trend')
plt.xlabel('Month')
plt.ylabel('Revenue')
plt.xticks(rotation=45)

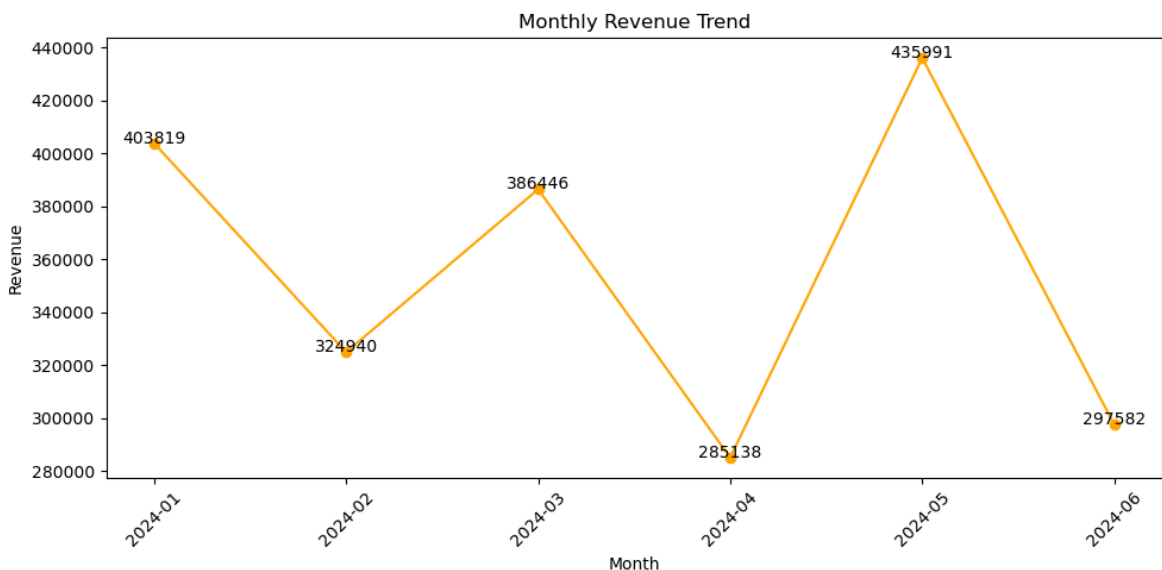
# Add data labels on each point
for i, value in enumerate(df_month['revenue']):
    plt.text(i, value + 10, f'{value:.0f}', ha='center')

plt.tight_layout()
plt.show()

# Close database connection
conn.close()
```

Monthly Revenue Trend:

	month	revenue
0	2024-01	403818.52
1	2024-02	324940.40
2	2024-03	386446.49
3	2024-04	285137.93
4	2024-05	435991.03
5	2024-06	297581.87



4. Top 5 Customers by Total Revenue

```
In [20]: conn = sqlite3.connect('sales_data.db')

query = """
SELECT customer, SUM(quantity * price) AS revenue
FROM sales
GROUP BY customer
ORDER BY revenue DESC
LIMIT 5
"""

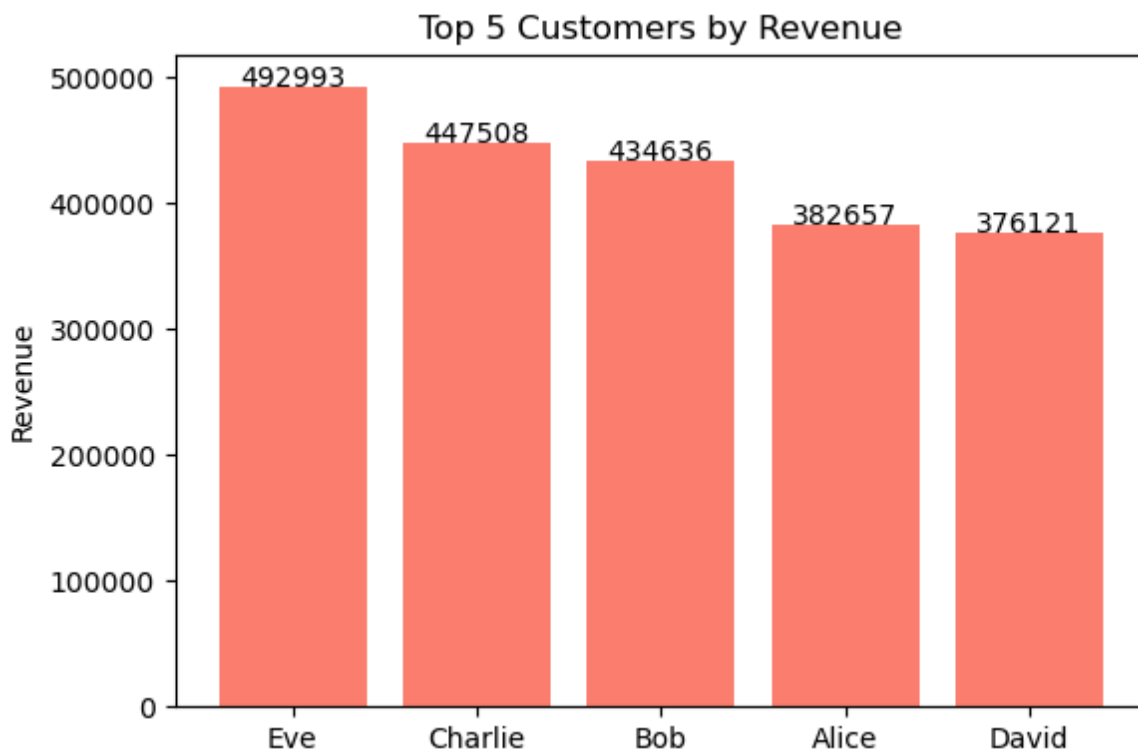
df_top_customers = pd.read_sql_query(query, conn)
print("Top 5 Customers by Revenue:")
print(df_top_customers)

# Plot
plt.figure(figsize=(6,4))
```

```
plt.bar(df_top_customers['customer'], df_top_customers['revenue'], color='salmon')
plt.title('Top 5 Customers by Revenue')
plt.ylabel('Revenue')
for i, val in enumerate(df_top_customers['revenue']):
    plt.text(i, val + 10, f'{val:.0f}', ha='center')
plt.tight_layout()
plt.show()
```

Top 5 Customers by Revenue:

	customer	revenue
0	Eve	492993.42
1	Charlie	447508.25
2	Bob	434636.24
3	Alice	382657.03
4	David	376121.30



✓ 5. Top-Selling Products by Quantity

```
In [23]: query = """
SELECT product, SUM(quantity) AS total_quantity
FROM sales
GROUP BY product
ORDER BY total_quantity DESC
"""

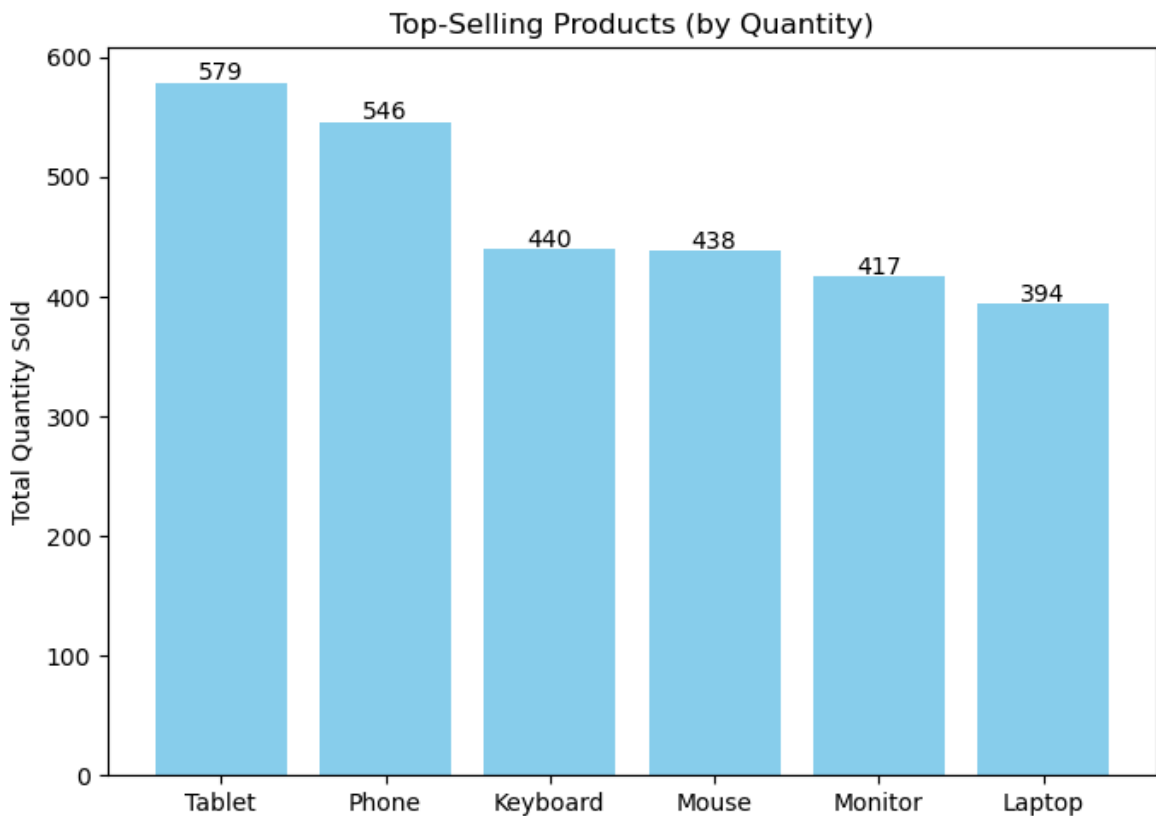
df_top_qty = pd.read_sql_query(query, conn)
print("Top-Selling Products by Quantity:")
print(df_top_qty)

# Plot
plt.figure(figsize=(7,5))
plt.bar(df_top_qty['product'], df_top_qty['total_quantity'], color='skyblue')
plt.title('Top-Selling Products (by Quantity)')
plt.ylabel('Total Quantity Sold')
for i, val in enumerate(df_top_qty['total_quantity']):
    plt.text(i, val + 2, str(val), ha='center')
```

```
plt.tight_layout()
plt.show()
```

Top-Selling Products by Quantity:

	product	total_quantity
0	Tablet	579
1	Phone	546
2	Keyboard	440
3	Mouse	438
4	Monitor	417
5	Laptop	394



6. Average Order Value by Region

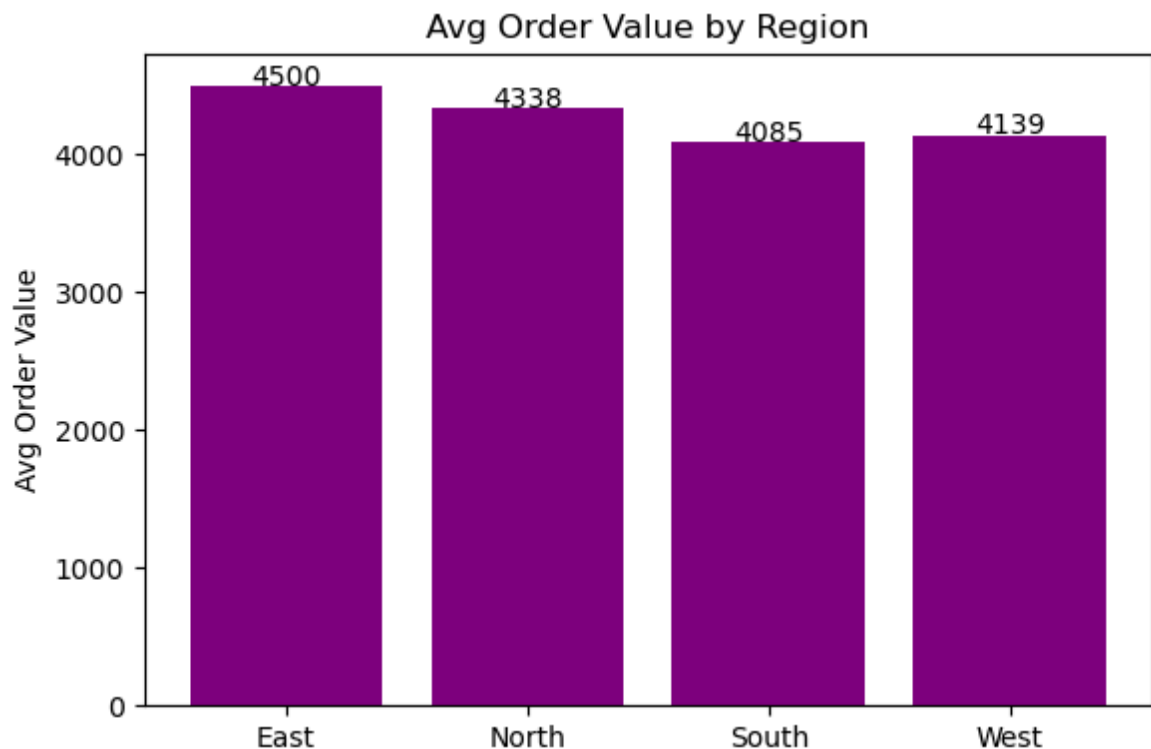
```
In [26]: query = """
SELECT region, AVG(quantity * price) AS avg_order_value
FROM sales
GROUP BY region
"""

df_avg_region = pd.read_sql_query(query, conn)
print("Average Order Value by Region:")
print(df_avg_region)

# Plot
plt.figure(figsize=(6,4))
plt.bar(df_avg_region['region'], df_avg_region['avg_order_value'], color='purple')
plt.title('Avg Order Value by Region')
plt.ylabel('Avg Order Value')
for i, val in enumerate(df_avg_region['avg_order_value']):
    plt.text(i, val + 5, f'{val:.0f}', ha='center')
plt.tight_layout()
plt.show()
```


Average Order Value by Region:

	region	avg_order_value
0	East	4500.033435
1	North	4338.165556
2	South	4085.191667
3	West	4138.656812



In [40]: ##  7. Daily Revenue Trend

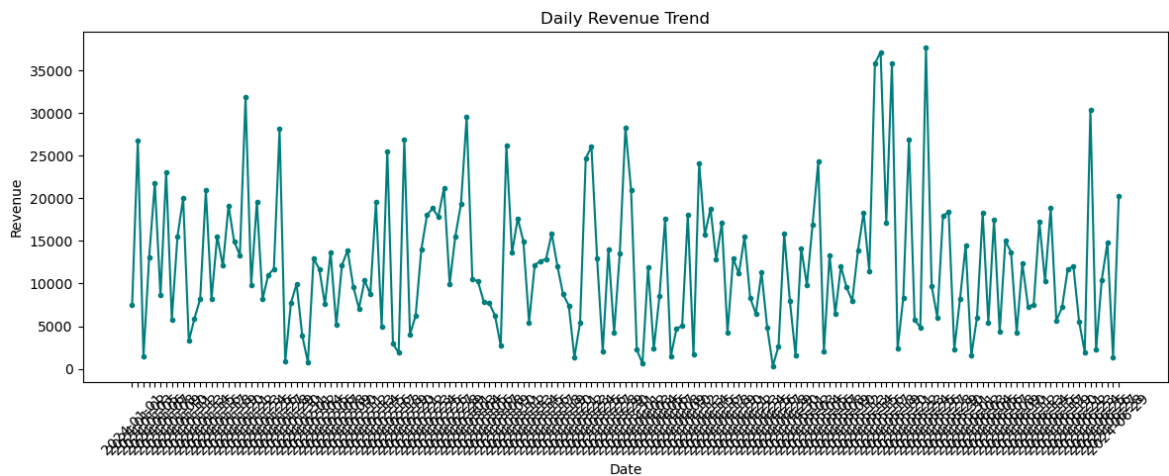
```
In [28]: query = """
SELECT date, SUM(quantity * price) AS daily_revenue
FROM sales
GROUP BY date
ORDER BY date
"""

df_daily = pd.read_sql_query(query, conn)
print("Daily Revenue Trend:")
print(df_daily.head())

# Plot
plt.figure(figsize=(12,5))
plt.plot(df_daily['date'], df_daily['daily_revenue'], marker='.', linestyle='-',
plt.title('Daily Revenue Trend')
plt.xlabel('Date')
plt.ylabel('Revenue')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Daily Revenue Trend:

	date	daily_revenue
0	2024-01-01	7467.56
1	2024-01-02	26770.74
2	2024-01-03	1418.64
3	2024-01-05	13074.60
4	2024-01-06	21767.66



✓ 8. Customer Purchase Frequency

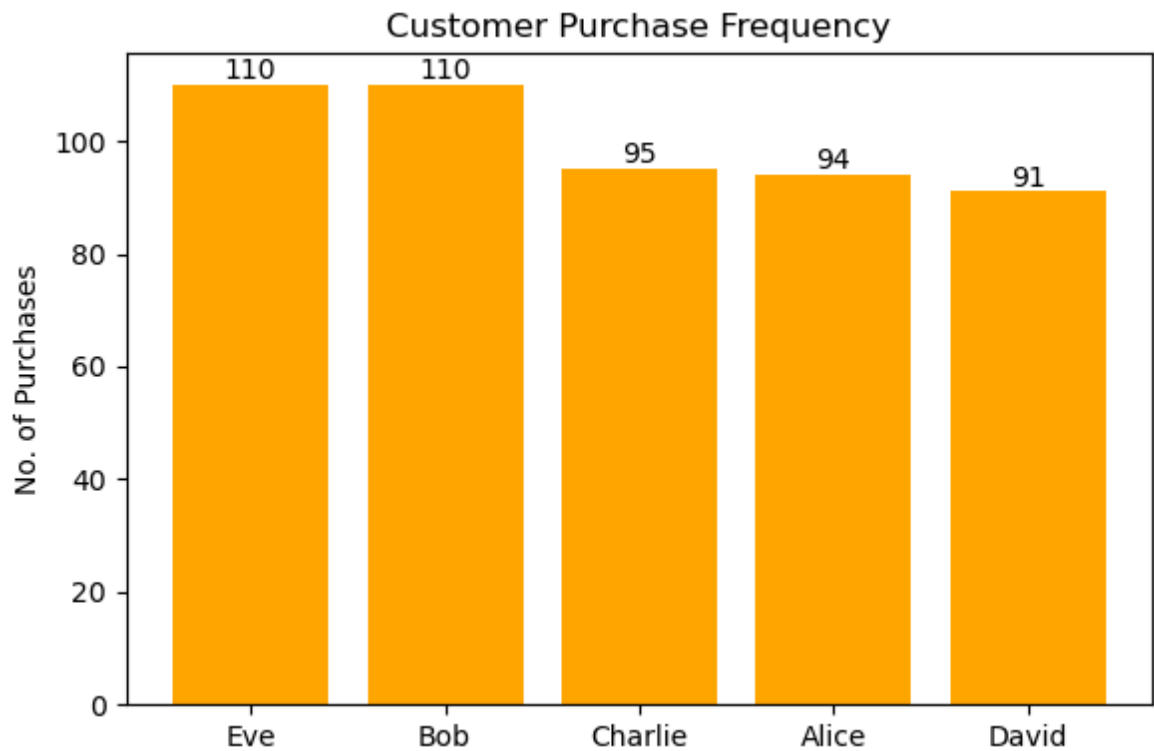
```
In [30]: query = """
SELECT customer, COUNT(*) AS purchase_count
FROM sales
GROUP BY customer
ORDER BY purchase_count DESC
"""

df_freq = pd.read_sql_query(query, conn)
print("Customer Purchase Frequency:")
print(df_freq)

# Plot
plt.figure(figsize=(6,4))
plt.bar(df_freq['customer'], df_freq['purchase_count'], color='orange')
plt.title('Customer Purchase Frequency')
plt.ylabel('No. of Purchases')
for i, val in enumerate(df_freq['purchase_count']):
    plt.text(i, val + 1, str(val), ha='center')
plt.tight_layout()
plt.show()
```

Customer Purchase Frequency:

	customer	purchase_count
0	Eve	110
1	Bob	110
2	Charlie	95
3	Alice	94
4	David	91

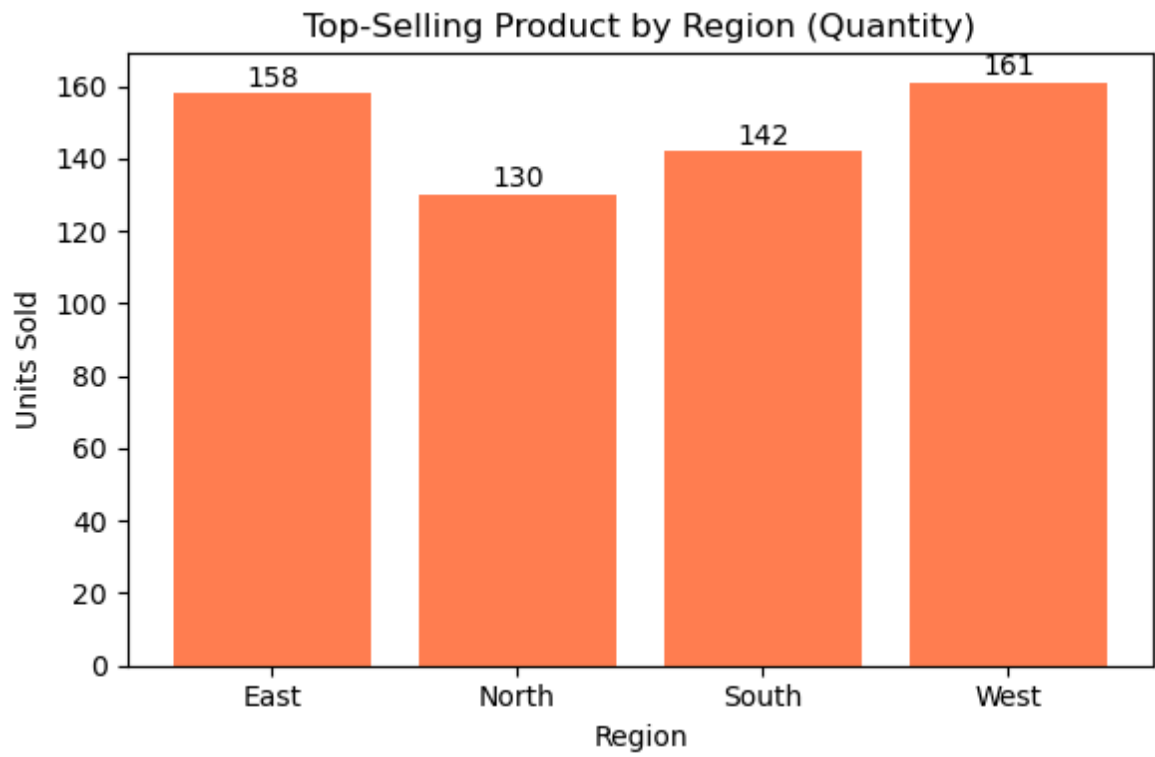


✓ 9. Most Popular Product in Each Region

```
In [36]: plt.figure(figsize=(6,4))
plt.bar(top_per_region['region'], top_per_region['total_qty'], color='coral')
plt.title('Top-Selling Product by Region (Quantity)')
plt.xlabel('Region')
plt.ylabel('Units Sold')

# Add data labels
for i, val in enumerate(top_per_region['total_qty']):
    plt.text(i, val + 2, f'{val}', ha='center')

plt.tight_layout()
plt.show()
```



In []: