Sales analysis with SQLite + Pandas (Jupyter notebook)

This notebook creates `sales_data.db`, seeds sample data, runs queries, shows results, plots revenue-by-product, and exports CSV/chart files.

In [2]:
```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
import os
from IPython.display import FileLink, display

%matplotlib inline
```

In [3]:
```
DB = "sales_data.db"
conn = sqlite3.connect(DB)
cur = conn.cursor()

# Create table
cur.execute("""
CREATE TABLE IF NOT EXISTS sales (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    sale_date TEXT,
    product TEXT NOT NULL,
    quantity INTEGER NOT NULL,
    price REAL NOT NULL
)
""")

# Seed only if empty
cur.execute("SELECT COUNT(*) FROM sales")
count = cur.fetchone()[0]
if count == 0:
    sample_data = [
        ('2025-09-01','Widget A', 10, 9.99),
        ('2025-09-01','Widget B', 5, 19.99),
        ('2025-09-02','Widget A', 3, 9.99),
        ('2025-09-02','Widget C', 8, 4.50),
        ('2025-09-03','Widget B', 7, 19.99),
        ('2025-09-03','Widget C', 2, 4.50),
        ('2025-09-04','Widget D', 1, 99.99),
        ('2025-09-04','Widget A', 6, 9.99),
        ('2025-09-05','Widget B', 2, 19.99),
        ('2025-09-05','Widget E', 4, 14.25),
        ('2025-09-06','Widget C', 5, 4.50),
        ('2025-09-06','Widget D', 2, 99.99),
    ]
    cur.executemany("INSERT INTO sales (sale_date, product, quantity, price) VALUES (?, ?, ?, ?)", sample_data)
    conn.commit()
    print(f"Seeded database with {len(sample_data)} rows.")
else:
    print(f"Database already has {count} rows; not seeding.")

conn.close()
print("Database file created at:", os.path.abspath(DB))
```

```
Seeded database with 12 rows.
Database file created at: C:\Users\Admin\Desktop\Py for data analytics\Excel Workbooks\sales_data.db
```

In [4]:
```
conn = sqlite3.connect(DB)

query = """
SELECT product,
       SUM(quantity) AS total_qty,
       SUM(quantity * price) AS revenue
FROM sales
GROUP BY product
ORDER BY revenue DESC
"""
df = pd.read_sql_query(query, conn)
df['revenue'] = df['revenue'].round(2)   # nicer formatting
conn.close()

# Display results in the notebook
df
```

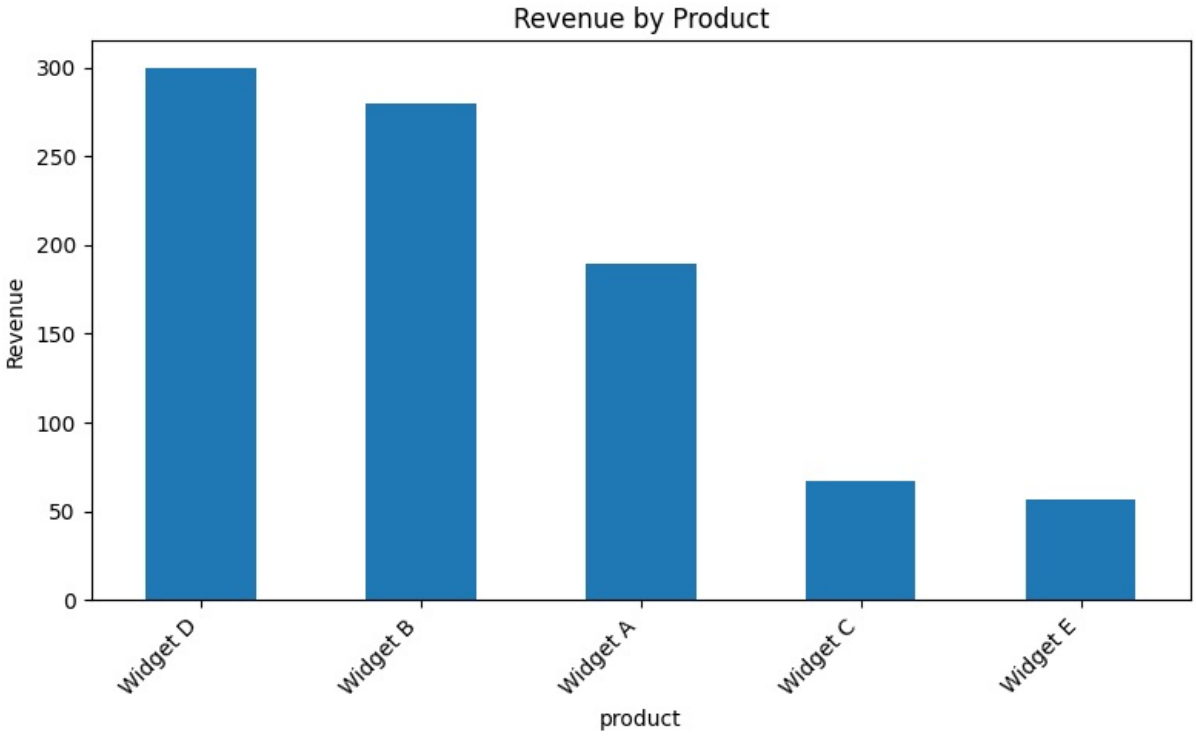
Out[4]:

	product	total_qty	revenue
0	Widget D	3	299.97
1	Widget B	14	279.86
2	Widget A	19	189.81
3	Widget C	15	67.50
4	Widget E	4	57.00

In [5]:
```
ax = df.plot(kind='bar', x='product', y='revenue', legend=False, figsize=(8,5))
ax.set_ylabel("Revenue")
ax.set_title("Revenue by Product")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()

out_file = "sales_chart.png"
plt.savefig(out_file, dpi=150)
plt.show()

# Show a clickable download link for the saved chart
display(FileLink(out_file))
```



sales_chart.png

In [6]:
```
conn = sqlite3.connect(DB)

q_top = """
SELECT product, SUM(quantity) AS total_qty
FROM sales
GROUP BY product
ORDER BY total_qty DESC
LIMIT 1;
"""
q_daily = """
SELECT sale_date, ROUND(SUM(quantity * price),2) AS daily_revenue
FROM sales
GROUP BY sale_date
ORDER BY sale_date;
"""

top_df = pd.read_sql_query(q_top, conn)
daily_df = pd.read_sql_query(q_daily, conn)
conn.close()

print("Top-selling product by quantity:")
display(top_df)

print("\nDaily revenue:")
display(daily_df)
```

Top-selling product by quantity:

	product	total_qty
0	Widget A	19

Daily revenue:

	sale_date	daily_revenue
0	2025-09-01	199.85
1	2025-09-02	65.97
2	2025-09-03	148.93
3	2025-09-04	159.93
4	2025-09-05	96.98
5	2025-09-06	222.48

In [7]:
```
csv_file = "product_revenue.csv"
df.to_csv(csv_file, index=False)
print("Saved CSV to:", os.path.abspath(csv_file))
display(FileLink(csv_file))
```

```
Saved CSV to: C:\Users\Admin\Desktop\Py for data analytics\Excel Workbooks\product_revenue.csv
```

product_revenue.csv

In [8]:
```
for f in ['sales_data.db', 'sales_chart.png', 'product_revenue.csv']:
    print(f, "->", os.path.exists(f))
```

```
sales_data.db -> True
sales_chart.png -> True
product_revenue.csv -> True
```