In [1]:

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

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

In [3]:

```
query = """
SELECT
    product,
    SUM(quantity) AS total_qty,
    SUM(quantity * price) AS revenue
FROM sales
GROUP BY product
"""
df = pd.read_sql_query(query, conn)
```

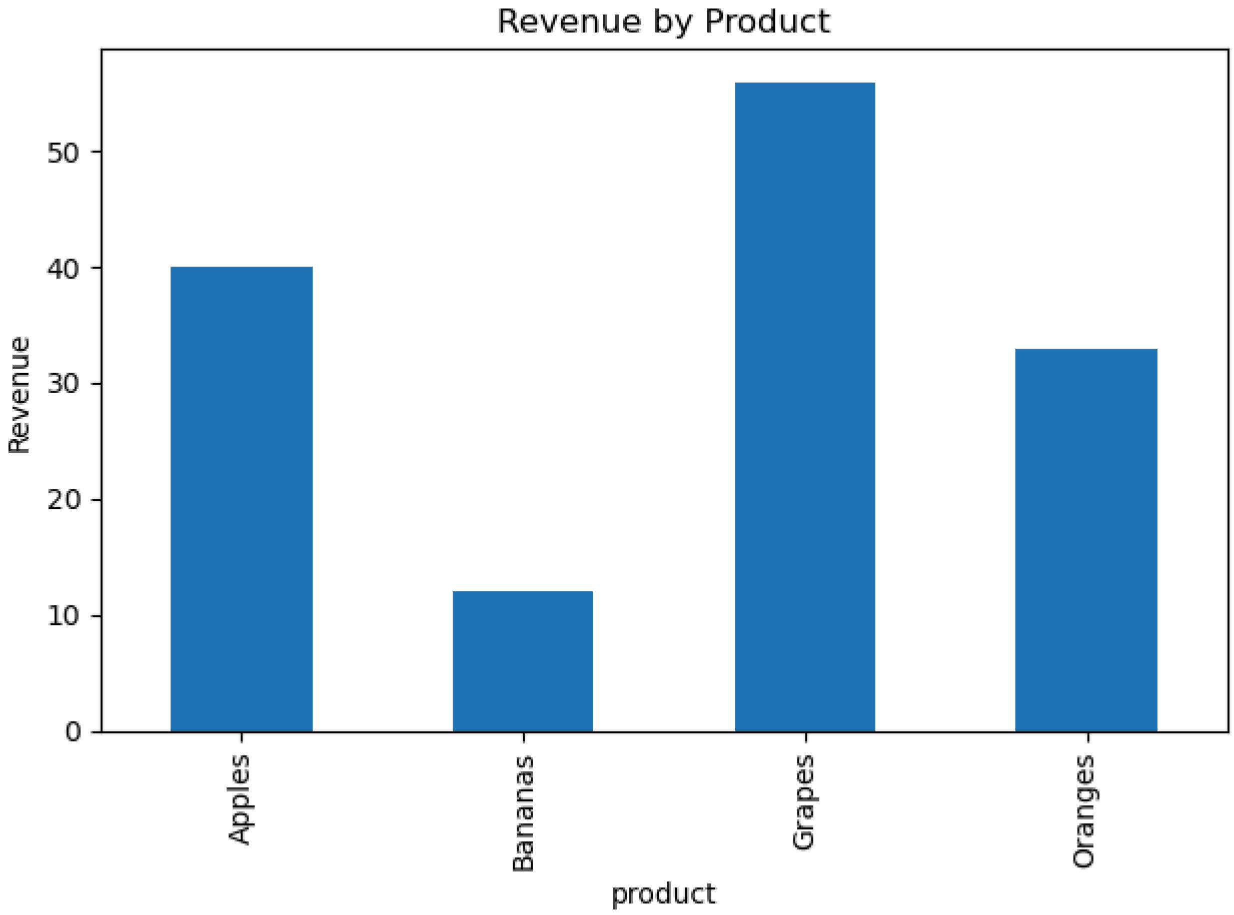
In [4]:

```
print(df)
```

```
   product  total_qty  revenue
0   Apples         80     40.0
1  Bananas         40     12.0
2   Grapes         70     56.0
3  Oranges         55     33.0
```

In [5]:

```
#Revenue by product
df.plot(kind='bar', x='product', y='revenue', title='Revenue by Product', legend=False)
plt.ylabel('Revenue')
plt.tight_layout()
plt.savefig("sales_chart.png")
plt.show()
```



Observation: Revenue by Product (Bar Chart)

The bar chart above shows the **total revenue** generated by each product in our sales data.

- **Grapes** earned the highest revenue, making it the top-performing product.
- **Apples** followed closely, suggesting they are also strong sellers.
- **Oranges** generated moderate revenue, indicating average sales.
- **Bananas** contributed the least revenue, likely due to a lower selling price or quantity.

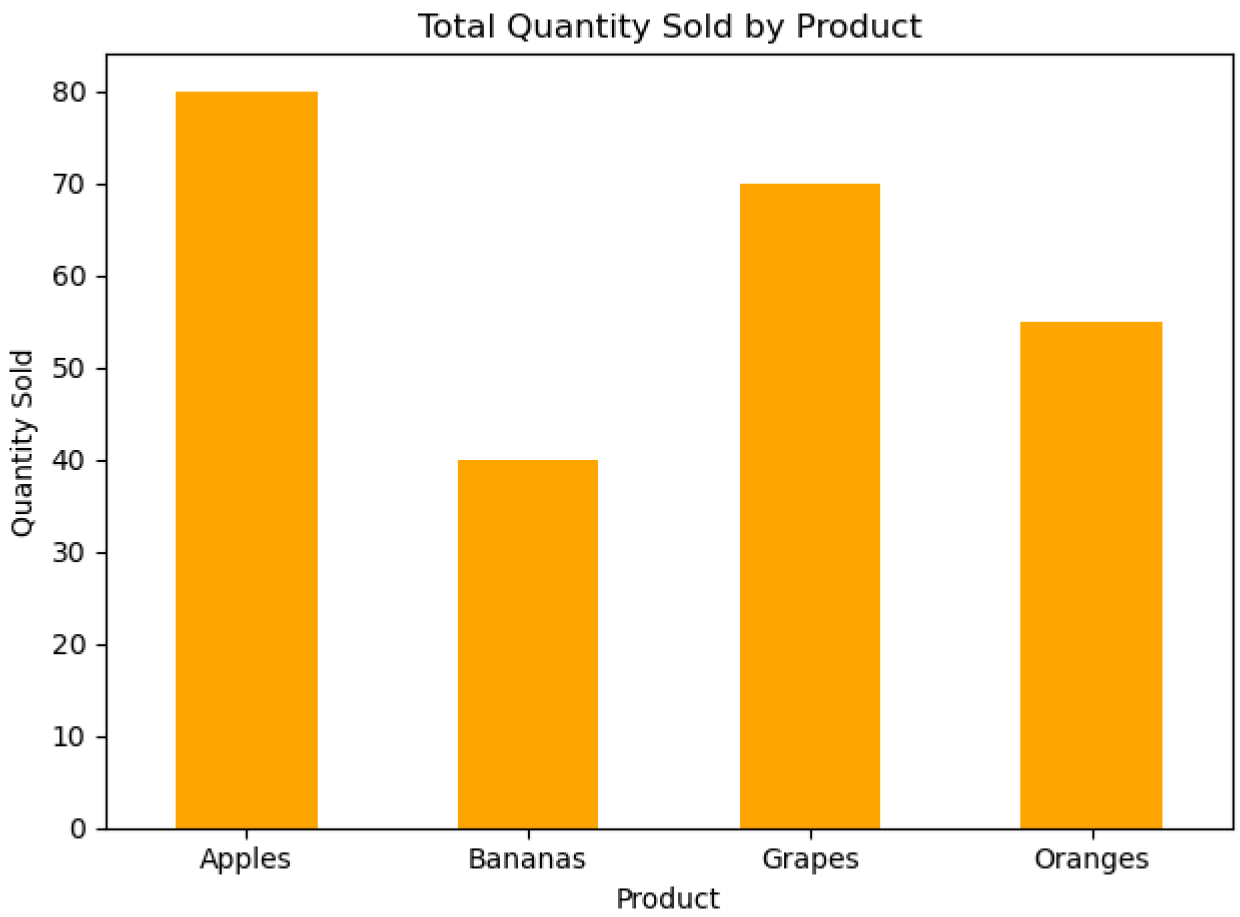
This visualization helps us understand which products are most profitable and can guide inventory or marketing strategies.

In [6]:

```
query_qty = """
SELECT
    product,
    SUM(quantity) AS total_qty
FROM sales
GROUP BY product
"""
df_qty = pd.read_sql_query(query_qty, conn)
```

In [7]:

```
# Plot total quantity sold per product
df_qty.plot(kind='bar', x='product', y='total_qty', legend=False, color='orange')
plt.title('Total Quantity Sold by Product')
plt.ylabel('Quantity Sold')
plt.xlabel('Product')
plt.xticks(rotation=0)
plt.tight_layout()
plt.savefig("quantity_bar_chart.png")
plt.show()
```



Observation: Quantity Sold by Product (Bar Chart)

This chart shows the **total units sold** per product. While **Grapes** and **Apples** lead in quantity sold, **Bananas** and **Oranges** lag behind. This helps us understand not just profitability but also demand volume per item.

In [ ]:

```
## ✅ Summary of Sales Data Analysis
In this notebook, we performed a basic sales analysis using a small SQLite database and visualized the results with Python.
###  Key Steps:
- **Connected** to the SQLite database (`sales_data.db`) using `sqlite3`.
- **Queried** the sales table to calculate:
  - Total **revenue** per product.
  - Total **quantity sold** per product.
- **Loaded** the SQL query results into `pandas` DataFrames.
- **Printed** the summary tables to view the data.
- **Visualized** the data using `matplotlib` with two bar charts:
  1. **Revenue by Product**
  2. **Total Quantity Sold by Product**
### Insights:
- **Grapes** generated the highest revenue and had strong unit sales.
- **Apples** also performed well in both revenue and quantity.
- **Bananas** had the lowest revenue, likely due to a lower price or volume.
- These charts help identify top-performing products and guide sales strategy.
```