1.*IMPORT LIBRARIES*

In [2]:
```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
```

2.*CREATE TABLE AND INSERT DATA*

Connect to SQLite database

In [5]:
```
# Connect to SQLite database
conn = sqlite3.connect("sales_data.db")
cursor = conn.cursor()
```

Create table

In [24]:
```
cursor.execute('''
CREATE TABLE IF NOT EXISTS sales (
    id INTEGER PRIMARY KEY,
    product TEXT,
    quantity INTEGER,
    price REAL
)
''')
```

Out[24]:
```
<sqlite3.Cursor at 0x28c17ca29c0>
```

Insert professional sample data

In [26]:
```
sample_data = [
    ('papaya', 4, 55000.0),
    ('Mango', 10, 1500.0),
    ('Grape', 6, 25000.0),
    ('Laptop', 2, 55000.0),
    ('Headphones', 5, 1500.0)
]

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

print("Sample data inserted successfully.")
```

```
Sample data inserted successfully.
```

3.*EXECUTE SQL AND LOAD WITH PANDAS*

SQL query to summarize total quantity and revenue

In [40]:
```
query = '''
SELECT
    product,
    SUM(quantity) AS total_quantity,
    SUM(quantity * price) AS total_revenue
FROM sales
GROUP BY product
'''
```

Load result into a pandas DataFrame

In [42]:
```
df = pd.read_sql_query(query, conn)
```

Display the result

In [44]:
```
print("Sales_data.db")
print(df)
```

```
Sales_data.db
      product  total_quantity  total_revenue
0  Headphones              30        22500.0
1    Keyboard              46        57000.0
2      Laptop              32      1650000.0
3     Monitor              18       216000.0
4       Mouse              69        32000.0
5  Smartphone               6       150000.0
6   USB Cable             120        24000.0
```

4.*PLOT THE BAR CHART*

Plot revenue per product

In [ ]:
```
df.plot(kind='bar', x='product', y='total_revenue', legend=False, color='
plt.title('Total Revenue by Product')
plt.xlabel('Product')
plt.ylabel('Revenue (INR)')
plt.tight_layout()
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

Save the chart

In [ ]:
```
plt.savefig("sales_chart.png")
```

5.*CLOSE CONNECTION*

In [ ]:
```
conn.close()
```