```
import pandas as pd

df = pd.read_csv('sales_data.csv')
df.head()
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Tot Amou
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	1
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	10
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	5
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	1

Next steps: Generate code with df New interactive sheet

```
import os

os.makedirs('raw', exist_ok=True)
os.makedirs('processed', exist_ok=True)
os.makedirs('output', exist_ok=True)

df.to_csv('raw/raw_data.csv', index=False)
```

```
print("Before:", df.shape)

df = df.drop_duplicates()
df = df.dropna()

print("After:", df.shape)
```

```
Before: (1000, 9)
After: (1000, 9)
```

```
df.columns = (
    df.columns
    .str.lower()
    .str.replace(' ', '_')
)

df.columns
```

```
Index(['transaction_id', 'date', 'customer_id', 'gender', 'age',
       'product_category', 'quantity', 'price_per_unit', 'total_amount'],
      dtype='object')
```

```
df['date'] = pd.to_datetime(df['date'])
df['age'] = df['age'].astype(int)
df['quantity'] = df['quantity'].astype(int)
df['price_per_unit'] = df['price_per_unit'].astype(float)
df['total_amount'] = df['total_amount'].astype(float)

# Derived column
df['calculated_amount'] = df['quantity'] * df['price_per_unit']
df['high_value_txn'] = df['total_amount'].apply(lambda x: 1 if x > 1000 else
```

```
customers = df[['customer_id', 'gender', 'age']].drop_duplicates()

orders = df[['transaction_id', 'date', 'customer_id', 'total_amount', 'high_

products = df[['product_category', 'price_per_unit']].drop_duplicates()
```

```
customers.to_csv('output/customers.csv', index=False)
orders.to_csv('output/orders.csv', index=False)
products.to_csv('output/products.csv', index=False)
```

```
import sqlite3

conn = sqlite3.connect('database.sqlite')

customers.to_sql('customers', conn, if_exists='replace', index=False)
orders.to_sql('orders', conn, if_exists='replace', index=False)
products.to_sql('products', conn, if_exists='replace', index=False)

conn.close()
```