```
import pandas as pd
import numpy as np

data = {
    'OrderID': [1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010],
    'Date': ['2025-01-05', '2025-01-06', '2025-01-07', '2025-01-07', '2025-01-08',
             '2025-01-09', '2025-01-10', '2025-01-11', '2025-01-12', '2025-01-13'],
    'CustomerID': ['C001', 'C002', 'C003', 'C004', 'C001', 'C005', 'C002', 'C006', 'C003', 'C007'],
    'Product': ['Laptop', 'Smartphone', 'Office Chair', 'Desk', 'Headphones',
                'Monitor', 'Laptop', 'Desk', 'Smartphone', 'Office Chair'],
    'Category': ['Electronics', 'Electronics', 'Furniture', 'Furniture', 'Electronics',
                 'Electronics', 'Electronics', 'Furniture', 'Electronics', 'Furniture'],
    'Quantity': [2, 1, 4, 1, 3, 2, 1, 2, 2, 5],
    'UnitPrice': [700, 500, 150, 300, 50, 200, 700, 300, 500, 150],
    'Total': [1400, 500, 600, 300, 150, 400, 700, 600, 1000, 750],

}

df = pd.DataFrame(data)

df['Date'] = pd.to_datetime(df['Date'])

print(df)
```

```
   OrderID       Date CustomerID       Product     Category  Quantity  \
0     1001 2025-01-05       C001        Laptop  Electronics         2
1     1002 2025-01-06       C002    Smartphone  Electronics         1
2     1003 2025-01-07       C003  Office Chair    Furniture         4
3     1004 2025-01-07       C004          Desk    Furniture         1
4     1005 2025-01-08       C001    Headphones  Electronics         3
5     1006 2025-01-09       C005       Monitor  Electronics         2
6     1007 2025-01-10       C002        Laptop  Electronics         1
7     1008 2025-01-11       C006          Desk    Furniture         2
8     1009 2025-01-12       C003    Smartphone  Electronics         2
9     1010 2025-01-13       C007  Office Chair    Furniture         5

   UnitPrice  Total
0        700   1400
1        500    500
2        150    600
3        300    300
4         50    150
5        200    400
6        700    700
7        300    600
8        500   1000
9        150    750
```

```
# find Total Revenue of  sales dataset
total_revenue = df['Total'].sum()
print("Total Revenue:", total_revenue)
```

```
Total Revenue: 6400
```

```
#find Average Revenue per Order of sales dataset
avg_revenue = df.groupby('OrderID')['Total'].sum().mean()
print("Average Revenue:",avg_revenue)
```

```
Average Revenue: 640.0
```

```
# get Top 5 Products by Sales
top_products = df.groupby('Product')['Total'].sum().sort_values(ascending=False).head(5)
print("Top 5 products by sales:",top_products)
```

```
Top 5 products by sales: Product
Laptop          2100
Smartphone      1500
Office Chair    1350
Desk             900
Monitor          400
Name: Total, dtype: int64
```

```
#find Monthly Sales Trend
df['Date'] = pd.to_datetime(df['Date'])
monthly_sales = df.groupby(df['Date'].dt.to_period('M'))['Total'].sum()
print("monthly sales:",monthly_sales)
```

```
monthly sales: Date
2025-01    6400
Freq: M, Name: Total, dtype: int64
```

```
#Total Units Sold by Category in sales dataset
units_by_category = df.groupby('Category')['Quantity'].sum()
print("Total units sold by category:",units_by_category)
```

```
Total units sold by category: Category
Electronics    11
Furniture      12
Name: Quantity, dtype: int64
```

```
#find Customer with Highest Purchase in sales
top_customer = df.groupby('CustomerID')['Total'].sum().idxmax()
print("Customer with Highest Purchase:",top_customer)
```

```
Customer with Highest Purchase: C003
```

```
#find no of Days with Zero Sales
daily_sales = df.groupby(df['Date'].dt.date)['Total'].sum()
zero_sales_days = (daily_sales == 0).sum()

print("Days with Zero Sales:",zero_sales_days)
```

```
Days with Zero Sales: 0
```

```
#find Most Frequently Sold Product in sales
```

```
most_frequent_product = df['Product'].mode()[0]

print("Most Frequently Sold Product:",most_frequent_product)
```

```
Most Frequently Sold Product: Desk
```

```
#find Percentage of Sales per Category
category_sales_pct = df.groupby('Category')['Total'].sum() / df['Total'].sum() * 100

print(" Percentage of Sales per Category:",category_sales_pct)
```

```
 Percentage of Sales per Category: Category
Electronics    64.84375
Furniture      35.15625
Name: Total, dtype: float64
```

```
#Find the Day with Maximum Sales
max_sales_day = df.groupby(df['Date'].dt.date)['Total'].sum().idxmax()
print("Max sales day:",max_sales_day)
```

```
Max sales day: 2025-01-05
```

```
#Count of Unique Products Sold in sales dataset

unique_products = df['Product'].nunique()
print("unique products:",unique_products)
```

```
unique products: 6
```

```
#find the Total and Average Revenue per Customer
customer_stats = df.groupby('CustomerID')['Total'].agg(['sum', 'mean'])
print(customer_stats)
```

```
              sum   mean
CustomerID
C001         1550  775.0
C002         1200  600.0
C003         1600  800.0
C004          300  300.0
C005          400  400.0
C006          600  600.0
C007          750  750.0
```

```
#Find the Product with the Highest Unit Price
most_expensive_product = df.loc[df['UnitPrice'].idxmax(), 'Product']
print("Product with the Highest Unit Price:",most_expensive_product)
```

```
Product with the Highest Unit Price: Laptop
```

```
#Identify the Most Profitable Product (Based on Total Profit)
df['Profit'] = df['Total'] * 0.20  # Assuming 20% margin
most_profitable_product = df.groupby('Product')['Profit'].sum().sort_values(ascending=False).head(1)
print(" Most Profitable Product:",most_profitable_product)
```

```
 Most Profitable Product: Product
Laptop    420.0
Name: Profit, dtype: float64
```

```
#List all the unique products sold.

print("\nUnique products sold:")
print(df['Product'].unique())
```

```
Unique products sold:
['Laptop' 'Smartphone' 'Office Chair' 'Desk' 'Headphones' 'Monitor']
```

```
#count of unique customers
print("\nNumber of unique customers:")
print(df['CustomerID'].nunique())
```

```
Number of unique customers:
7
```

```
#Total quantity sold
print("\nTotal quantity sold:")
print(df['Quantity'].sum())
```

```
Total quantity sold:
23
```

```
#find the Transactions where quantity > 2
print("\nTransactions with more than 2 units sold:")
print(df[df['Quantity'] > 2])
```

```
Transactions with more than 2 units sold:
   OrderID       Date CustomerID       Product     Category  Quantity  \
2     1003 2025-01-07       C003  Office Chair    Furniture         4
4     1005 2025-01-08       C001    Headphones  Electronics         3
9     1010 2025-01-13       C007  Office Chair    Furniture         5

   UnitPrice  Total  Profit
2        150    600   120.0
4         50    150    30.0
9        150    750   150.0
```

```
 # find Product with lowest unit price
print("\nProduct with the lowest unit price:")
print(df.loc[df['UnitPrice'].idxmin(), 'Product'])
```

```
Product with the lowest unit price:
Headphones
```

```
#find Number of times 'Laptop' was sold
print("\nNumber of transactions where 'Laptop' was sold:")
print(df[df['Product'] == 'Laptop'].shape[0])
```

```
Number of transactions where 'Laptop' was sold:
2
```

```
#Add 'DiscountedPrice' column (10% off UnitPrice) in sales dataset
df['DiscountedPrice'] = df['UnitPrice'] * 0.90
print("\nDataset after adding DiscountedPrice column (10% discount):")
print(df[['Product', 'UnitPrice', 'DiscountedPrice']])
```

```
Dataset after adding DiscountedPrice column (10% discount):
        Product  UnitPrice  DiscountedPrice
0        Laptop        700            630.0
1    Smartphone        500            450.0
2  Office Chair        150            135.0
3          Desk        300            270.0
4    Headphones         50             45.0
5       Monitor        200            180.0
6        Laptop        700            630.0
7          Desk        300            270.0
8    Smartphone        500            450.0
9  Office Chair        150            135.0
```

Start coding or generate with AI.