**### \*\*Exercise 5: Handling Missing Values\*\*  
#1. Create a DataFrame with missing values:**import pandas as pd  
data = {  
 "Name": ["Amit", "Neha", "Raj", "Priya"],  
 "Age": [28, None, 35, 29],  
 "City": ["Delhi", "Mumbai", None, "Chennai"]  
}  
df = pd.DataFrame(data)  
print(df)

**# 2. Fill missing values in the `"Age"` column with the average age.**

df['Age'].fillna(df['Age'].mean(), inplace=True)  
print(df)  
  
**# 3. Drop rows where any column has missing data.**df\_dropped = df.dropna()  
print(df\_dropped)  
  
**# ### \*\*Exercise 6: Adding and Removing Columns\*\*  
# 1. Add a new column `"Salary"` with the following values: `[50000, 60000, 70000, 65000]`.**

df['Salary'] = [50000, 60000, 70000, 65000]  
  
**# 2. Remove the `"City"` column from the DataFrame.**

dfdrop= df.drop(columns=['City'])  
  
**# ### \*\*Exercise 7: Sorting Data\*\*  
# 1. Sort the DataFrame by `"Age"` in ascending order.**

df\_sorted = df.sort\_values(by='Age')  
print(df\_sorted)

**# 2. Sort the DataFrame first by `"City"` and then by `"Age"` in descending order.**

df = df.sort\_values(by=['City', 'Age'], ascending=[True, False])  
print(df)

**# ### \*\*Exercise 8: Grouping and Aggregation\*\*  
# 1. Group the DataFrame by `"City"` and calculate the average `"Age"` for each city.**

df= df.groupby('City')['Age'].mean()  
print(df)

**# 2. Group the DataFrame by `"City"` and `"Age"`, and count the number of occurrences for each group.**

df= df.groupby(['City', 'Age']).size()  
print(df)

**# ### \*\*Exercise 9: Merging DataFrames\*\*  
# 1. Create two DataFrames:**

df1 = pd.DataFrame({  
 "Name": ["Amit", "Neha", "Raj"],  
 "Department": ["HR", "IT", "Finance"]  
})  
df2 = pd.DataFrame({  
 "Name": ["Neha", "Raj", "Priya"],  
 "Salary": [60000, 70000, 65000]  
})

**# 2. Merge `df1` and `df2` on the `"Name"` column using an inner join.**

df\_inner = pd.merge(df1, df2, on="Name", how="inner")  
print(df\_inner)

**# 3. Merge the same DataFrames using a left join.**

df\_left = pd.merge(df1, df2, on="Name", how="left")  
print(df\_left)