1️⃣ Data Selection & Indexing

Used to **select specific rows or columns** from a DataFrame using column names, index values, `loc`, and `iloc`.

2️⃣ Data Cleaning

Process of **removing or fixing dirty data** like missing values, duplicates, and wrong data types to make data usable.

3️⃣ Filtering Data

Used to **extract required data** based on conditions (for example, marks > 80).

4️⃣ Adding & Modifying Columns

Used to **create new columns or update existing columns** using calculations or values.

5️⃣ Sorting Data

Used to **arrange data** in ascending or descending order based on one or more columns.

6️⃣ GroupBy

Used to **group data and perform aggregate operations** like sum, mean, count, etc. (similar to SQL GROUP BY).

7️⃣ Merging & Joining

Used to **combine multiple DataFrames** based on a common column (similar to SQL JOIN).

8️⃣ Apply & Lambda Functions

Used to **apply custom logic or functions** to rows or columns in a DataFrame.

STEP 1: Data Selection & Indexing

- **Select columns**
- **Select rows**
- **Use `loc` and `iloc`**

```
df["Marks"]
df[["Name", "Age"]]
df.loc[0]
df.iloc[1]
```

📌 **This is asked in interviews.**

◆ STEP 2: Data Cleaning

- **Missing values**
- **Duplicates**
- **Changing data types**

```
df.isnull()
df.dropna()
df.fillna(0)
df.duplicated()
df.drop_duplicates()
```

◆ STEP 3: Filtering Data

how to apply conditions:

```
df[df["Marks"] > 80]
df[(df["Age"] > 20) & (df["Marks"] > 85)]
```

◆ STEP 4: Adding & Modifying Columns

```
df["Result"] = "Pass"
```

```
df["Bonus"] = df["Marks"] + 5
```

◆ STEP 5: Sorting Data

```
df.sort_values("Marks")
df.sort_values("Marks", ascending=False)
```

◆ STEP 6: GroupBy

Used in real projects.

```
df.groupby("Age")["Marks"].mean()
```

◆ STEP 7: Merging & Joining DataFrames

Like SQL joins:

```
pd.merge(df1, df2, on="ID")
```

◆ STEP 8: Apply & Lambda Functions

```
df["Grade"] = df["Marks"].apply(lambda x: "A" if x > 85 else "B")
```

1 Data Selection & Indexing

Definition: **Used to select specific rows or columns from a DataFrame**.

```
import pandas as pd

df = pd.DataFrame({
```

```
    "Name": ["Aman", "Riya", "Neha"],
    "Age": [20, 21, 22],
    "Marks": [85, 90, 78]
})

print(df["Marks"])     # Select column
print(df.loc[0])       # Select row by label
print(df.iloc[1])      # Select row by index
```

---

2 Data Cleaning

Definition: **Used to handle missing values and duplicate data.**

```
df = pd.DataFrame({
    "Name": ["Aman", "Riya", "Riya"],
    "Marks": [85, None, 90]
})

print(df.isnull())        # Check missing values
df.fillna(0, inplace=True) # Fill missing values
df.drop_duplicates(inplace=True)
```

3 Filtering Data

Definition: **Used to extract data based on conditions.**

```
print(df[df["Marks"] > 80])
```

4 Adding & Modifying Columns

Definition: **Used to create or update columns.**

```
df["Result"] = "Pass"
df["Marks"] = df["Marks"] + 5
print(df)
```

5 Sorting Data

Definition: **Used to arrange data in ascending or descending order.**

```
df.sort_values("Marks", ascending=False, inplace=True)
print(df)
```