**Day-10 Morning Assessment**

**create a table with columns and rows column name : students,technology,marks add the column teacher and delete the technology later and try to to practice on all the example methods**

import pandas as pd  
data = {  
 "students": ["alice", "bob", "charlie","amitha","vidya","sravya","yash","nithish","vishnu","meghana","jithin","prem","dinesh","vijay","pavan"],  
 "technology": ["Python", "Java", "HTML", "CSS", "javascript", "C", "sql", "AI/ML", "DSA", "OS", "automation","cybersecurity","hackathon","devops",".net"],  
 "marks": [85, 78, 92, 95, 94, 93, 96, 80, 85, 89, 90, 87, 88, 84, 90]  
}  
df = pd.DataFrame(data)  
print("Initial Table:\n", df, "\n")  
  
df["teacher"] = ["Mr.Smith", "Mr.Smith", "Mr.Smith", "Mr.Smith", "Mr.Smith", "Mr.Smith", "Mr.john","Mr.john","Mr.john","Mr.john","Mr.john","Mr.john","Mr.john","Mr.john","Mr.john"]  
print("After adding 'teacher' column:\n", df, "\n")  
  
df = df.drop(columns=["technology"])  
print("After deleting 'technology' column:\n", df, "\n")  
  
new\_row = pd.DataFrame([{"students": "kiran", "marks": 88, "teacher": "Mrs. Johnson"}])  
df = pd.concat([df, new\_row], ignore\_index=True)  
print("After adding a new row:\n", df, "\n")  
  
df.loc[df["students"] == "Alice", "marks"] = 90  
print("After updating Alice's marks:\n", df, "\n")  
  
df = df[df["students"] != "Bob"]  
print("After deleting Bob:\n", df, "\n")  
  
filtered\_df = df[df["marks"] > 85]  
print("Students with marks > 85:\n", filtered\_df, "\n")  
  
sorted\_df = df.sort\_values(by="marks", ascending=False)  
print("Sorted by marks (descending):\n", sorted\_df, "\n")

o/p:

Initial Table:

students technology marks

0 alice Python 85

1 bob Java 78

2 charlie HTML 92

3 amitha CSS 95

4 vidya javascript 94

5 sravya C 93

6 yash sql 96

7 nithish AI/ML 80

8 vishnu DSA 85

9 meghana OS 89

10 jithin automation 90

11 prem cybersecurity 87

12 dinesh hackathon 88

13 vijay devops 84

14 pavan .net 90

After adding 'teacher' column:

students technology marks teacher

0 alice Python 85 Mr.Smith

1 bob Java 78 Mr.Smith

2 charlie HTML 92 Mr.Smith

3 amitha CSS 95 Mr.Smith

4 vidya javascript 94 Mr.Smith

5 sravya C 93 Mr.Smith

6 yash sql 96 Mr.john

7 nithish AI/ML 80 Mr.john

8 vishnu DSA 85 Mr.john

9 meghana OS 89 Mr.john

10 jithin automation 90 Mr.john

11 prem cybersecurity 87 Mr.john

12 dinesh hackathon 88 Mr.john

13 vijay devops 84 Mr.john

14 pavan .net 90 Mr.john

After deleting 'technology' column:

students marks teacher

0 alice 85 Mr.Smith

1 bob 78 Mr.Smith

2 charlie 92 Mr.Smith

3 amitha 95 Mr.Smith

4 vidya 94 Mr.Smith

5 sravya 93 Mr.Smith

6 yash 96 Mr.john

7 nithish 80 Mr.john

8 vishnu 85 Mr.john

9 meghana 89 Mr.john

10 jithin 90 Mr.john

11 prem 87 Mr.john

12 dinesh 88 Mr.john

13 vijay 84 Mr.john

14 pavan 90 Mr.john

After adding a new row:

students marks teacher

0 alice 85 Mr.Smith

1 bob 78 Mr.Smith

2 charlie 92 Mr.Smith

3 amitha 95 Mr.Smith

4 vidya 94 Mr.Smith

5 sravya 93 Mr.Smith

6 yash 96 Mr.john

7 nithish 80 Mr.john

8 vishnu 85 Mr.john

9 meghana 89 Mr.john

10 jithin 90 Mr.john

11 prem 87 Mr.john

12 dinesh 88 Mr.john

13 vijay 84 Mr.john

14 pavan 90 Mr.john

15 kiran 88 Mrs. Johnson

After updating Alice's marks:

students marks teacher

0 alice 85 Mr.Smith

1 bob 78 Mr.Smith

2 charlie 92 Mr.Smith

3 amitha 95 Mr.Smith

4 vidya 94 Mr.Smith

5 sravya 93 Mr.Smith

6 yash 96 Mr.john

7 nithish 80 Mr.john

8 vishnu 85 Mr.john

9 meghana 89 Mr.john

10 jithin 90 Mr.john

11 prem 87 Mr.john

12 dinesh 88 Mr.john

13 vijay 84 Mr.john

14 pavan 90 Mr.john

15 kiran 88 Mrs. Johnson

After deleting Bob:

students marks teacher

0 alice 85 Mr.Smith

1 bob 78 Mr.Smith

2 charlie 92 Mr.Smith

3 amitha 95 Mr.Smith

4 vidya 94 Mr.Smith

5 sravya 93 Mr.Smith

6 yash 96 Mr.john

7 nithish 80 Mr.john

8 vishnu 85 Mr.john

9 meghana 89 Mr.john

10 jithin 90 Mr.john

11 prem 87 Mr.john

12 dinesh 88 Mr.john

13 vijay 84 Mr.john

14 pavan 90 Mr.john

15 kiran 88 Mrs. Johnson

Students with marks > 85:

students marks teacher

2 charlie 92 Mr.Smith

3 amitha 95 Mr.Smith

4 vidya 94 Mr.Smith

5 sravya 93 Mr.Smith

6 yash 96 Mr.john

9 meghana 89 Mr.john

10 jithin 90 Mr.john

11 prem 87 Mr.john

12 dinesh 88 Mr.john

14 pavan 90 Mr.john

15 kiran 88 Mrs. Johnson

Sorted by marks (descending):

students marks teacher

6 yash 96 Mr.john

3 amitha 95 Mr.Smith

4 vidya 94 Mr.Smith

5 sravya 93 Mr.Smith

2 charlie 92 Mr.Smith

10 jithin 90 Mr.john

14 pavan 90 Mr.john

9 meghana 89 Mr.john

12 dinesh 88 Mr.john

15 kiran 88 Mrs. Johnson

11 prem 87 Mr.john

0 alice 85 Mr.Smith

8 vishnu 85 Mr.john

13 vijay 84 Mr.john

7 nithish 80 Mr.john

1 bob 78 Mr.Smith

**Pandas**

import pandas as pd  
from io import StringIO  
  
# 1. Read a CSV file into a Pandas DataFrame  
data\_csv = """Name,Age,Department,Salary  
Alice,23,Data Engineer,20000  
Bob,24,System Engineer,30000  
Charlie,33,HR,40000  
David,23,ASE,50000  
Eva,25,ASE,60000"""  
df = pd.read\_csv(StringIO(data\_csv))  
print("1. First 5 Rows:")  
  
print(df.head()) # display first 5 rows  
  
# 2. Check the shape of the DataFrame  
  
print("\n2. Shape of DataFrame (rows, columns):")  
  
print(df.shape)  
  
# 3. Get summary statistics for numeric columns  
  
print("\n3. Summary Statistics:")  
  
print(df.describe())  
  
# 4. Select a single column  
  
print("\n4. 'Age' Column:")  
  
age\_series = df["Age"]  
  
print(age\_series)  
  
# 5. Filter rows where Salary > 50000  
  
print("\n5. Rows where Salary > 50000:")  
  
high\_salary = df[df["Salary"] > 50000]  
  
print(high\_salary)  
  
# 6. Filter with multiple conditions: Department is 'HR' and Age > 30  
  
print("\n6. Department = 'HR' and Age > 30:")  
  
filtered\_df = df[(df["Department"] == "HR") & (df["Age"] > 30)]  
  
print(filtered\_df)  
  
# 7. Check for missing values  
  
print("\n7. Missing Values in Columns:")  
  
missing = df.isna().sum()  
  
print(missing)  
  
# 8. Replace missing values in Salary with 0  
  
print("\n8. Fill missing 'Salary' with 0:")  
  
df["Salary"] = df["Salary"].fillna(0)  
  
print(df["Salary"])  
  
# 9. Remove duplicate rows and reset index  
  
print("\n9. DataFrame after removing duplicates:")  
  
df\_no\_duplicates = df.drop\_duplicates().reset\_index(drop=True)  
  
print(df\_no\_duplicates)  
  
# 10. Sort the DataFrame by "Age" in descending order  
  
print("\n10. Sorted by Age (descending):")  
  
sorted\_df = df.sort\_values(by="Age", ascending=False)  
  
print(sorted\_df)  
  
# 11. Group by Department and find average Salary  
  
avg\_salary = df.groupby("Department")["Salary"].mean()  
print("\n11. Average Salary by Department:")  
print(avg\_salary)  
  
# 12. Count unique values in "Department"  
  
print("\n12. Number of unique departments:")  
  
unique\_dept\_count = df["Department"].nunique()  
  
print(unique\_dept\_count)

o/p:

1. First 5 Rows:

Name Age Department Salary

0 Alice 23 Data Engineer 20000

1 Bob 24 System Engineer 30000

2 Charlie 33 HR 40000

3 David 23 ASE 50000

4 Eva 25 ASE 60000

2. Shape of DataFrame (rows, columns):

(5, 4)

3. Summary Statistics:

Age Salary

count 5.000000 5.000000

mean 25.600000 40000.000000

std 4.219005 15811.388301

min 23.000000 20000.000000

25% 23.000000 30000.000000

50% 24.000000 40000.000000

75% 25.000000 50000.000000

max 33.000000 60000.000000

4. 'Age' Column:

0 23

1 24

2 33

3 23

4 25

Name: Age, dtype: int64

5. Rows where Salary > 50000:

Name Age Department Salary

4 Eva 25 ASE 60000

6. Department = 'HR' and Age > 30:

Name Age Department Salary

2 Charlie 33 HR 40000

7. Missing Values in Columns:

Name 0

Age 0

Department 0

Salary 0

dtype: int64

8. Fill missing 'Salary' with 0:

0 20000

1 30000

2 40000

3 50000

4 60000

Name: Salary, dtype: int64

9. DataFrame after removing duplicates:

Name Age Department Salary

0 Alice 23 Data Engineer 20000

1 Bob 24 System Engineer 30000

2 Charlie 33 HR 40000

3 David 23 ASE 50000

4 Eva 25 ASE 60000

10. Sorted by Age (descending):

Name Age Department Salary

2 Charlie 33 HR 40000

4 Eva 25 ASE 60000

1 Bob 24 System Engineer 30000

0 Alice 23 Data Engineer 20000

3 David 23 ASE 50000

11. Average Salary by Department:

Department

ASE 55000

Data Engineer 20000

HR 40000

System Engineer 30000

Name: Salary, dtype: int64

12. Number of unique departments:

4