

```
In [2]: import pandas as pd
import numpy as np

# Create the Project DataFrame
project_data = {
    'ID': ['A001', 'A002', 'A003', 'A004', 'A005', 'A001', 'A002',
    'Project': ['Project 1', 'Project 2', 'Project 3', 'Project 4',
    'Cost': [1000000, 2000000, None, 500000, 680000, 750000, 120000
    'Status': ['Finished', 'Ongoing', 'Finished', 'Ongoing', 'Finis
}
project_df = pd.DataFrame(project_data)
project_df
```

```
Out[2]:
```

	ID	Project	Cost	Status
0	A001	Project 1	1000000.0	Finished
1	A002	Project 2	2000000.0	Ongoing
2	A003	Project 3	NaN	Finished
3	A004	Project 4	500000.0	Ongoing
4	A005	Project 5	680000.0	Finished
5	A001	Project 6	750000.0	Finished
6	A002	Project 7	1200000.0	Ongoing
7	A003	Project 8	3000000.0	Finished
8	A004	Project 9	500000.0	Failed
9	A005	Project 10	1000000.0	Finished
10	A001	Project 11	900000.0	Ongoing
11	A002	Project 12	4000000.0	Finished
12	A003	Project 13	3000000.0	Finished
13	A004	Project 14	2000000.0	Finished

```
In [ ]:
```

```
In [3]: # Create the Employee DataFrame
employee_data = {
    'ID': ['A001', 'A002', 'A003', 'A004', 'A005'],
    'Name': ['John Alter', 'Tina Jon', 'Lomer Harry', 'Nina Arfa',
    'Gender': ['M', 'F', 'M', 'F', 'F'],
    'City': ['Paris', 'London', 'Berlin', 'Newyork', 'Madrid'],
    'Age': [25, 27, 29, 31, 30]
}
employee_df = pd.DataFrame(employee_data)
employee_df
```

Out[3]:

	ID	Name	Gender	City	Age
0	A001	John Alter	M	Paris	25
1	A002	Tina Jon	F	London	27
2	A003	Lomer Harry	M	Berlin	29
3	A004	Nina Arfa	F	Newyork	31
4	A005	Ajony Madrid	F	Madrid	30

In [4]:

```
# Create the Seniority Level DataFrame
seniority_data = {
    'ID': ['A001', 'A002', 'A003', 'A004', 'A005'],
    'Designation Level': [2, 3, 2, 3, 3]
}
seniority_df = pd.DataFrame(seniority_data)
seniority_df
```

Out[4]:

	ID	Designation Level
0	A001	2
1	A002	3
2	A003	2
3	A004	3
4	A005	3

In [5]:

```
# Save the dataframes to CSV
project_path = 'project_df.csv'
employee_path = 'employee_df.csv'
seniority_path = 'seniority_df.csv'

project_df.to_csv(project_path, index=False)
employee_df.to_csv(employee_path, index=False)
seniority_df.to_csv(seniority_path, index=False)

project_path, employee_path, seniority_path
```

Out[5]: ('project_df.csv', 'employee_df.csv', 'seniority_df.csv')

In [6]:

```
# -----
# Task 2: Fill missing 'Cost' using running average (with for loop)
# -----

# Load saved project.csv
project_df = pd.read_csv("project_df.csv")

# Convert Cost column to list for processing
costs = project_df['Cost'].tolist()

# Use a for loop to compute and replace missing values using running
```

```

for i in range(len(costs)):
    if pd.isna(costs[i]):
        # Take average of all previous non-missing values
        valid_previous_costs = [costs[j] for j in range(i) if not pd.isna(costs[j])]
        if valid_previous_costs:
            running_avg = sum(valid_previous_costs) / len(valid_previous_costs)
            costs[i] = running_avg

# Update the DataFrame
project_df['Cost'] = costs

# Save the updated dataframe
project_updated_path = "project_df.csv"
project_df.to_csv(project_updated_path, index=False)

# Show updated rows with previously missing value (to confirm fix)
project_df[project_df['Project'] == 'Project 3']

```

Out[6]:

	ID	Project	Cost	Status
2	A003	Project 3	1500000.0	Finished

In [7]:

```

# -----
# Task 3: Split Name into First Name and Last Name in employee data
# -----

# Load employee.csv
employee_df = pd.read_csv("employee_df.csv")

# Split the 'Name' column into two new columns: 'First Name' and 'Last Name'
employee_df[['First Name', 'Last Name']] = employee_df['Name'].str.split(' ', n=1, expand=True)

# Drop the original 'Name' column
employee_df.drop('Name', axis=1, inplace=True)

# Save updated employee dataframe
employee_updated_path = "employee_df.csv"
employee_df.to_csv(employee_updated_path, index=False)

# Show updated dataframe
employee_df

```

Out[7]:

	ID	Gender	City	Age	First Name	Last Name
0	A001	M	Paris	25	John	Alter
1	A002	F	London	27	Tina	Jon
2	A003	M	Berlin	29	Lomer	Harry
3	A004	F	Newyork	31	Nina	Arfa
4	A005	F	Madrid	30	Ajony	Madrid

In [8]:

```

# -----
# Task 4: Join all three dataframes in one single dataframe. Name i

```

```
# -----

# First merge: Project + Employee
merged_df = pd.merge(project_df, employee_df, on="ID")

# Second merge: + Seniority
final_df = pd.merge(merged_df, seniority_df, on="ID")

# Save the final dataframe
final_df.to_csv("Final.csv", index=False)

final_df
```

Out[8]:

	ID	Project	Cost	Status	Gender	City	Age	First Name	Last Name
0	A001	Project 1	1000000.0	Finished	M	Paris	25	John	Alt
1	A002	Project 2	2000000.0	Ongoing	F	London	27	Tina	Jo
2	A003	Project 3	1500000.0	Finished	M	Berlin	29	Lomer	Har
3	A004	Project 4	500000.0	Ongoing	F	Newyork	31	Nina	Ar
4	A005	Project 5	680000.0	Finished	F	Madrid	30	Ajony	Madr
5	A001	Project 6	750000.0	Finished	M	Paris	25	John	Alt
6	A002	Project 7	1200000.0	Ongoing	F	London	27	Tina	Jo
7	A003	Project 8	3000000.0	Finished	M	Berlin	29	Lomer	Har
8	A004	Project 9	500000.0	Failed	F	Newyork	31	Nina	Ar
9	A005	Project 10	1000000.0	Finished	F	Madrid	30	Ajony	Madr
10	A001	Project 11	900000.0	Ongoing	M	Paris	25	John	Alt
11	A002	Project 12	4000000.0	Finished	F	London	27	Tina	Jo
12	A003	Project 13	3000000.0	Finished	M	Berlin	29	Lomer	Har
13	A004	Project 14	2000000.0	Finished	F	Newyork	31	Nina	Ar

In [10]:

```
# -----
# Task 5: Add a 'Bonus' column: employees who have finished the pr
```

```
# -----  
  
# Load the Final.csv file  
final_df = pd.read_csv("Final.csv")  
  
# Ensure 'Cost' is numeric  
final_df['Cost'] = pd.to_numeric(final_df['Cost'], errors='coerce')  
  
# Create a boolean mask for finished projects  
is_finished = final_df['Status'].str.strip().str.lower() == 'finish'  
  
# Calculate bonus using NumPy's where  
final_df['Bonus'] = np.where(is_finished, 0.05 * final_df['Cost'],  
  
# Save the updated DataFrame  
final_df.to_csv("Final.csv", index=False)  
final_df
```

Out[10]:

	ID	Project	Cost	Status	Gender	City	Age	First Name	Last Name
0	A001	Project 1	1000000.0	Finished	M	Paris	25	John	Alt
1	A002	Project 2	2000000.0	Ongoing	F	London	27	Tina	Jo
2	A003	Project 3	1500000.0	Finished	M	Berlin	29	Lomer	Har
3	A004	Project 4	500000.0	Ongoing	F	Newyork	31	Nina	Ar
4	A005	Project 5	680000.0	Finished	F	Madrid	30	Ajony	Madr
5	A001	Project 6	750000.0	Finished	M	Paris	25	John	Alt
6	A002	Project 7	1200000.0	Ongoing	F	London	27	Tina	Jo
7	A003	Project 8	3000000.0	Finished	M	Berlin	29	Lomer	Har
8	A004	Project 9	500000.0	Failed	F	Newyork	31	Nina	Ar
9	A005	Project 10	1000000.0	Finished	F	Madrid	30	Ajony	Madr
10	A001	Project 11	900000.0	Ongoing	M	Paris	25	John	Alt
11	A002	Project 12	4000000.0	Finished	F	London	27	Tina	Jo
12	A003	Project 13	3000000.0	Finished	M	Berlin	29	Lomer	Har
13	A004	Project 14	2000000.0	Finished	F	Newyork	31	Nina	Ar

```

In [16]: # -----
# Task 5: Demote the designation level by 1, whose projects have s
# -----

'''
Step 1: Reduce 'Designation Level' by 1 for employees whose project
Step 2: Remove employees whose 'Designation Level' is above 4 after
'''

# Load Final.csv
final_df = pd.read_csv("Final.csv")

# Step 1: Demote designation level by 1 where status is "Failed"
for i in range(len(final_df)):
    if final_df.loc[i, 'Status'].lower() == 'failed':

```

```

        final_df.loc[i, 'Designation Level'] -= 1

# Step 2: Remove rows with Designation Level > 4
final_df = final_df[final_df['Designation Level'] <= 4]

# Save updated Final.csv
final_df.to_csv("Final.csv", index=False)
final_df

```

Out[16]:

	ID	Project	Cost	Status	Gender	City	Age	First Name	Last Name
0	A001	Project 1	1000000.0	Finished	M	Paris	25	John	Alt
1	A002	Project 2	2000000.0	Ongoing	F	London	27	Tina	Jo
2	A003	Project 3	1500000.0	Finished	M	Berlin	29	Lomer	Har
3	A004	Project 4	500000.0	Ongoing	F	Newyork	31	Nina	Ar
4	A005	Project 5	680000.0	Finished	F	Madrid	30	Ajony	Madr
5	A001	Project 6	750000.0	Finished	M	Paris	25	John	Alt
6	A002	Project 7	1200000.0	Ongoing	F	London	27	Tina	Jo
7	A003	Project 8	3000000.0	Finished	M	Berlin	29	Lomer	Har
8	A004	Project 9	500000.0	Failed	F	Newyork	31	Nina	Ar
9	A005	Project 10	1000000.0	Finished	F	Madrid	30	Ajony	Madr
10	A001	Project 11	900000.0	Ongoing	M	Paris	25	John	Alt
11	A002	Project 12	4000000.0	Finished	F	London	27	Tina	Jo
12	A003	Project 13	3000000.0	Finished	M	Berlin	29	Lomer	Har
13	A004	Project 14	2000000.0	Finished	F	Newyork	31	Nina	Ar

In [17]:

```

# -----
# Task 7: Add "Mr." and "Mrs." to the first name column and drop t
# -----

# Load Final.csv
final_df = pd.read_csv("Final.csv")

```

```

# Add "Mr." or "Mrs." to the First Name based on Gender
for i in range(len(final_df)):
    prefix = 'Mr. ' if final_df.loc[i, 'Gender'] == 'M' else 'Mrs.'
    final_df.loc[i, 'First Name'] = prefix + final_df.loc[i, 'First Name']

# Drop the Gender column
final_df.drop('Gender', axis=1, inplace=True)

# Save the updated file
final_df.to_csv("Final.csv", index=False)
final_df

```

Out[17]:

	ID	Project	Cost	Status	City	Age	First Name	Last Name	Designation
0	A001	Project 1	1000000.0	Finished	Paris	25	Mr. John	Alter	Manager
1	A002	Project 2	2000000.0	Ongoing	London	27	Mrs. Tina	Jon	Developer
2	A003	Project 3	1500000.0	Finished	Berlin	29	Mr. Lomer	Harry	Developer
3	A004	Project 4	500000.0	Ongoing	Newyork	31	Mrs. Nina	Arfa	Manager
4	A005	Project 5	680000.0	Finished	Madrid	30	Mrs. Ajony	Madrid	Developer
5	A001	Project 6	750000.0	Finished	Paris	25	Mr. John	Alter	Manager
6	A002	Project 7	1200000.0	Ongoing	London	27	Mrs. Tina	Jon	Developer
7	A003	Project 8	3000000.0	Finished	Berlin	29	Mr. Lomer	Harry	Developer
8	A004	Project 9	500000.0	Failed	Newyork	31	Mrs. Nina	Arfa	Manager
9	A005	Project 10	1000000.0	Finished	Madrid	30	Mrs. Ajony	Madrid	Developer
10	A001	Project 11	900000.0	Ongoing	Paris	25	Mr. John	Alter	Manager
11	A002	Project 12	4000000.0	Finished	London	27	Mrs. Tina	Jon	Developer
12	A003	Project 13	3000000.0	Finished	Berlin	29	Mr. Lomer	Harry	Developer
13	A004	Project 14	2000000.0	Finished	Newyork	31	Mrs. Nina	Arfa	Manager

In [18]:

```

# -----
# Task 8: Promote designation level by 1 for the employees whose ag

```



```
# -----

# Load Final.csv
final_df = pd.read_csv("Final.csv")

# Promote designation level for age > 29 using a for loop
for i in range(len(final_df)):
    if final_df.loc[i, 'Age'] > 29:
        final_df.loc[i, 'Designation Level'] += 1

# Save updated DataFrame
final_df.to_csv("Final.csv", index=False)
final_df
```

Out[18]:

	ID	Project	Cost	Status	City	Age	First Name	Last Name	Designation Level
0	A001	Project 1	1000000.0	Finished	Paris	25	Mr. John	Alter	1
1	A002	Project 2	2000000.0	Ongoing	London	27	Mrs. Tina	Jon	1
2	A003	Project 3	1500000.0	Finished	Berlin	29	Mr. Lomer	Harry	2
3	A004	Project 4	500000.0	Ongoing	Newyork	31	Mrs. Nina	Arfa	2
4	A005	Project 5	680000.0	Finished	Madrid	30	Mrs. Ajony	Madrid	2
5	A001	Project 6	750000.0	Finished	Paris	25	Mr. John	Alter	1
6	A002	Project 7	1200000.0	Ongoing	London	27	Mrs. Tina	Jon	1
7	A003	Project 8	3000000.0	Finished	Berlin	29	Mr. Lomer	Harry	2
8	A004	Project 9	500000.0	Failed	Newyork	31	Mrs. Nina	Arfa	2
9	A005	Project 10	1000000.0	Finished	Madrid	30	Mrs. Ajony	Madrid	2
10	A001	Project 11	900000.0	Ongoing	Paris	25	Mr. John	Alter	1
11	A002	Project 12	4000000.0	Finished	London	27	Mrs. Tina	Jon	1
12	A003	Project 13	3000000.0	Finished	Berlin	29	Mr. Lomer	Harry	2
13	A004	Project 14	2000000.0	Finished	Newyork	31	Mrs. Nina	Arfa	2

In [20]:

```
# -----
```

```

# Task 9: Add the cost of all projects for each Employee and save
# -----

# Load Final.csv
final_df = pd.read_csv("Final.csv")

# Group by ID and First Name, and sum the Cost
total_proj_cost = final_df.groupby(['ID', 'First Name'])['Cost'].su

# Rename the column to "Total Cost"
total_proj_cost.rename(columns={'Cost': 'Total Cost'}, inplace=True)

# Print the result to verify
print(total_proj_cost)

# Save the new DataFrame
total_proj_cost.to_csv("TotalProjCost.csv", index=False)

```

	ID	First Name	Total Cost
0	A001	Mr. John	2650000.0
1	A002	Mrs. Tina	7200000.0
2	A003	Mr. Lomer	7500000.0
3	A004	Mrs. Nina	3000000.0
4	A005	Mrs. Ajony	1680000.0

```

In [21]: # -----
# Task 8: Print all the employee details whose city name contains
# -----

# Load Final.csv
final_df = pd.read_csv("Final.csv")

# Filter rows where 'City' contains 'o' or '0'
filtered_df = final_df[final_df['City'].str.contains('o', case=False)]

# Print the filtered result
print(filtered_df)

```

	ID	Project	Cost	Status	City	Age	First Name	L
ast Name \								
1	A002	Project 2	2000000.0	Ongoing	London	27	Mrs. Tina	
Jon								
3	A004	Project 4	500000.0	Ongoing	Newyork	31	Mrs. Nina	
Arfa								
6	A002	Project 7	1200000.0	Ongoing	London	27	Mrs. Tina	
Jon								
8	A004	Project 9	500000.0	Failed	Newyork	31	Mrs. Nina	
Arfa								
11	A002	Project 12	4000000.0	Finished	London	27	Mrs. Tina	
Jon								
13	A004	Project 14	2000000.0	Finished	Newyork	31	Mrs. Nina	
Arfa								

	Designation	Level	Bonus
1		3	0.0
3		4	0.0
6		3	0.0
8		-1	0.0
11		3	200000.0
13		4	100000.0

In []:

In []: