20250630_Skillovilla_Capstone_py101

July 1, 2025

[1]:

```
#Impots libraries for the analysis

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from functools import reduce
```

0.0.1 *Task 1: Creation of DataFrames and saving them into .CSV file*

There are three different tables as given above. Please make three dataframe in python and save them as three .csv files. From Task 2 to Task 10, use the saved .csv files only.

[2]:

```
# Creating the data - 'Project' using the data provided

Project = {
    'ID' : ['A001', 'A002' ,'A003' ,'A004' ,'A005' ,'A002' ,'A005' ,'A003'␣
 ↪,'A001' ,'A003' ,'A001' ,'A004' ,'A004' ,'A005'],
    'Project' : ['Project 1', 'Project 2' ,'Project 3' ,'Project 4' ,'Project␣
 ↪5' ,'Project 6' ,'Project 7' ,'Project 8' ,'Project 9' ,'Project 10'␣
 ↪,'Project 11' ,'Project 12' ,'Project 13' ,'Project 14'],
    'Cost' : [1002000, 2000000, 4500000, 5500000, None, 680000, 400000, 350000,␣
 ↪None, 300000, 2000000, 1000000, 3000000, 200000],
    'Status' : ['Finished', 'Ongoing' ,'Finished' ,'Ongoing' ,'Finished'␣
 ↪,'Failed' ,'Finished' ,'Failed' ,'Ongoing' ,'Finished' ,'Failed' ,'Ongoing'␣
 ↪,'Finished' ,'Finished' ]
}
```

[3]:

```
# Creating the data - 'Employee' using the data provided

Employee = {
    'ID' : ['A001', 'A002' ,'A003' ,'A004' ,'A005'],
    'Name' : ['John Alter', 'Alice Luxumberg', 'Tom Sabestine', 'Nina Adgra',␣
 ↪'Amy Johny'],
    'Gender' : ['M', 'F', 'M', 'F', 'F'],
    'City' : ['Paris', 'London', 'Berlin', 'New York', 'Madrid'],
    'Age' : [25,27,29,31,30]
}
```

[4]:

```
# Creating the data - 'Saniority_Level' using the data provided

Seniority_Level = {
    'ID' : ['A001', 'A002' ,'A003' ,'A004' ,'A005'],
    'Designation Level' : [2,2,3,2,3]
}
```

[5]:

```
#Converting the datas into DataFrames respetively

df_project = pd.DataFrame(Project)
df_employee = pd.DataFrame(Employee)
df_sl = pd.DataFrame(Seniority_Level)
```

[6]:

```
df_project.head()
```

[6]:

```
     ID    Project       Cost    Status
0  A001  Project 1  1002000.0  Finished
1  A002  Project 2  2000000.0   Ongoing
2  A003  Project 3  4500000.0  Finished
3  A004  Project 4  5500000.0   Ongoing
4  A005  Project 5        NaN  Finished
```

[7]:

```
#Exporting the DataFrames into .CSV files respectively

df_project.to_csv(r'E:\Data Storage\Skillovilla_Capstone_python\Project.csv',␣
 ↪index = False)
df_employee.to_csv(r'E:\Data Storage\Skillovilla_Capstone_python\Employee.csv',␣
 ↪index= False)
df_sl.to_csv(r'E:\Data Storage\Skillovilla_Capstone_python\Seniority_Level.
 ↪csv', index= False)
```

[8]:

```
#Reading the 'Project.csv' data into python using pandas

df_project_up = pd.read_csv(r'E:\Data␣
 ↪Storage\Skillovilla_Capstone_python\Project.csv')
```

[9]:

```
print(df_project_up)
```

```
     ID    Project       Cost    Status
0  A001  Project 1  1002000.0  Finished
1  A002  Project 2  2000000.0   Ongoing
2  A003  Project 3  4500000.0  Finished
3  A004  Project 4  5500000.0   Ongoing
4  A005  Project 5        NaN  Finished
5  A002  Project 6   680000.0    Failed
6  A005  Project 7   400000.0  Finished
7  A003  Project 8   350000.0    Failed
8  A001  Project 9        NaN   Ongoing
```

```
9   A003  Project 10   300000.0  Finished
10  A001  Project 11  2000000.0    Failed
11  A004  Project 12  1000000.0   Ongoing
12  A004  Project 13  3000000.0  Finished
13  A005  Project 14   200000.0  Finished
```

[10]:
```
#Reading the 'Employee.csv' data into python using pandas

df_emp_up = pd.read_csv(r'E:\Data Storage\Skillovilla_Capstone_python\Employee.
 ↪csv')
```

[11]:
```
print(df_emp_up)
```

```
     ID             Name Gender      City  Age
0  A001       John Alter      M     Paris   25
1  A002  Alice Luxumberg      F    London   27
2  A003    Tom Sabestine      M    Berlin   29
3  A004       Nina Adgra      F  New York   31
4  A005        Amy Johny      F    Madrid   30
```

[12]:
```
#Reading the 'Seniority_Level.csv' data into python using pandas

df_sl_up = pd.read_csv(r'E:\Data␣
 ↪Storage\Skillovilla_Capstone_python\Seniority_Level.csv')
```

[13]:
```
print(df_sl_up)
```

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

0.1

0.1.1 *Task 2: Modify the Missing values in the 'Cost' column in Project DataFrame by using the running average*

The cost column in the dataframe "Project" has some missing values. Your task is to compute these missing values. Replace the missing values by running average. You should use the "For" loop for this task.

[14]:
```
# Converting the 'Null' value of 'Cost' column into the running average of the␣
 ↪preceeding values

c = 0
n = 0
for i in range(len(df_project_up)):
    if pd.isnull(df_project_up.loc[i, 'Cost']):
        if n == 0:
            df_project_up.loc[i, 'Cost'] = 0
        else:
            df_project_up.loc[i, 'Cost'] = c/n
    else:
        c+= df_project_up.loc[i, 'Cost']
        n+= 1
df_project_up['Cost'] = df_project_up['Cost'].astype(int)
```

[15]:
```
print(df_project_up)
```

```
      ID     Project     Cost    Status
0   A001   Project 1  1002000  Finished
1   A002   Project 2  2000000   Ongoing
2   A003   Project 3  4500000  Finished
3   A004   Project 4  5500000   Ongoing
4   A005   Project 5  3250500  Finished
5   A002   Project 6   680000    Failed
6   A005   Project 7   400000  Finished
7   A003   Project 8   350000    Failed
8   A001   Project 9  2061714   Ongoing
9   A003  Project 10   300000  Finished
10  A001  Project 11  2000000    Failed
11  A004  Project 12  1000000   Ongoing
12  A004  Project 13  3000000  Finished
13  A005  Project 14   200000  Finished
```

0.2

0.2.1 *Task 3: Spliting 'Name' into 'First Name' and 'Last Name' and removal of 'Name' column*

Split the name column in the Employee dataframe into two new columns "First Name", and "LastName" and remove the older "name" column.

[16]:
```
# Spliting the 'Name' column into 'First_name' and 'Last_name' columns␣
 ↪respectively

df_emp_up[['First_name', 'Last_name']] = df_emp_up['Name'].str.split(' ', n=1,␣
 ↪expand = True)
```

[17]:
```
df_emp_up.head()
```

[17]:
```
     ID             Name Gender    City  Age First_name  Last_name
0  A001       John Alter      M   Paris   25       John      Alter
1  A002  Alice Luxumberg      F  London   27      Alice  Luxumberg
2  A003    Tom Sabestine      M  Berlin   29        Tom  Sabestine
```

```
3  A004       Nina Adgra      F  New York   31        Nina       Adgra
4  A005        Amy Johny      F    Madrid   30         Amy       Johny
```

[18]:

```
df_emp_up.columns
```

[18]:

```
Index(['ID', 'Name', 'Gender', 'City', 'Age', 'First_name', 'Last_name'],
dtype='object')
```

[19]:

```
# Removal of old 'Name' column and sorting the table in order of ID,␣
 ↪Name(First, Last), Gender, City and Age

df_emp_up = df_emp_up[['ID', 'First_name', 'Last_name',  'Gender', 'City',␣
 ↪'Age']]
```

[20]:

```
df_emp_up.head()
```

[20]:

```
     ID First_name  Last_name Gender      City  Age
0  A001       John      Alter      M     Paris   25
1  A002      Alice  Luxumberg      F    London   27
2  A003        Tom  Sabestine      M    Berlin   29
3  A004       Nina      Adgra      F  New York   31
4  A005        Amy      Johny      F    Madrid   30
```

0.3

0.3.1 *Task 4: Join all three DataFrames in one called 'Final'*

Join all three dataframes in one single dataframe. Name it "Final"

[21]:

```
# Joining 'Project' and 'Employee' into 'Final1'

df_final1 = df_project_up.merge(df_emp_up, on = 'ID', how = 'left')

#Joining 'Final1' and 'Senioriry_Level' into 'Final'

Final = df_final1.merge(df_sl_up, on = 'ID', how = 'left')
```

[22]:

```
Final.head(15)
```

[22]:

```
      ID     Project     Cost    Status First_name  Last_name Gender  \
0   A001   Project 1  1002000  Finished       John      Alter      M
1   A002   Project 2  2000000   Ongoing      Alice  Luxumberg      F
2   A003   Project 3  4500000  Finished        Tom  Sabestine      M
3   A004   Project 4  5500000   Ongoing       Nina      Adgra      F
4   A005   Project 5  3250500  Finished        Amy      Johny      F
5   A002   Project 6   680000    Failed      Alice  Luxumberg      F
6   A005   Project 7   400000  Finished        Amy      Johny      F
7   A003   Project 8   350000    Failed        Tom  Sabestine      M
```

```
8   A001   Project 9  2061714   Ongoing       John      Alter      M
9   A003  Project 10   300000  Finished        Tom  Sabestine      M
10  A001  Project 11  2000000    Failed       John      Alter      M
11  A004  Project 12  1000000   Ongoing       Nina      Adgra      F
12  A004  Project 13  3000000  Finished       Nina      Adgra      F
13  A005  Project 14   200000  Finished        Amy      Johny      F

        City  Age  Designation Level
0      Paris   25                  2
1     London   27                  2
2     Berlin   29                  3
3   New York   31                  2
4     Madrid   30                  3
5     London   27                  2
6     Madrid   30                  3
7     Berlin   29                  3
8      Paris   25                  2
9     Berlin   29                  3
10     Paris   25                  2
11  New York   31                  2
12  New York   31                  2
13    Madrid   30                  3
```

```
[23]: Final.columns
```

```
[23]: Index(['ID', 'Project', 'Cost', 'Status', 'First_name', 'Last_name', 'Gender',
             'City', 'Age', 'Designation Level'],
            dtype='object')
```

```
[24]: # Rearranging the Columns for better understanding of the data

      Final = Final[['ID','First_name', 'Last_name', 'Gender',
             'City', 'Age', 'Designation Level', 'Project', 'Cost', 'Status']]
```

```
[25]: Final.head()
```

```
[25]:      ID First_name  Last_name Gender      City  Age  Designation Level  \
      0  A001       John      Alter      M     Paris   25                  2
      1  A002      Alice  Luxumberg      F    London   27                  2
      2  A003        Tom  Sabestine      M    Berlin   29                  3
      3  A004       Nina      Adgra      F  New York   31                  2
      4  A005        Amy      Johny      F    Madrid   30                  3

           Project     Cost    Status
      0  Project 1  1002000  Finished
      1  Project 2  2000000   Ongoing
      2  Project 3  4500000  Finished
```

```
3   Project 4  5500000   Ongoing
4   Project 5  3250500  Finished
```

0.4

0.4.1 *Task 5: Create a new column 'Bonus' and assign a 5% of the project cost to the employee who has completed their projects*

Add a new bonus column in the Final dataframe. Give a 5% bonus concerning project cost only to employees who have finished the projects.

[26]:

```
# Creating a new column 'Bonus' and using Lambda function of assign a 5% bonus␣
 ↪for employees with Finished 'Status'

Final['Bonus'] = Final.apply(lambda row: row['Cost']*0.05 if row['Status'] ==␣
 ↪'Finished' else None, axis = 1)
```

[27]:

```
Final.head(14)
```

[27]:

```
      ID First_name  Last_name Gender      City  Age  Designation Level  \
0   A001       John      Alter      M     Paris   25                  2
1   A002      Alice  Luxumberg      F    London   27                  2
2   A003        Tom  Sabestine      M    Berlin   29                  3
3   A004       Nina      Adgra      F  New York   31                  2
4   A005        Amy      Johny      F    Madrid   30                  3
5   A002      Alice  Luxumberg      F    London   27                  2
6   A005        Amy      Johny      F    Madrid   30                  3
7   A003        Tom  Sabestine      M    Berlin   29                  3
8   A001       John      Alter      M     Paris   25                  2
9   A003        Tom  Sabestine      M    Berlin   29                  3
10  A001       John      Alter      M     Paris   25                  2
11  A004       Nina      Adgra      F  New York   31                  2
12  A004       Nina      Adgra      F  New York   31                  2
13  A005        Amy      Johny      F    Madrid   30                  3

       Project     Cost    Status     Bonus
0    Project 1  1002000  Finished   50100.0
1    Project 2  2000000   Ongoing       NaN
2    Project 3  4500000  Finished  225000.0
3    Project 4  5500000   Ongoing       NaN
4    Project 5  3250500  Finished  162525.0
5    Project 6   680000    Failed       NaN
6    Project 7   400000  Finished   20000.0
7    Project 8   350000    Failed       NaN
8    Project 9  2061714   Ongoing       NaN
9   Project 10   300000  Finished   15000.0
10  Project 11  2000000    Failed       NaN
11  Project 12  1000000   Ongoing       NaN
```

```
12  Project 13  3000000  Finished  150000.0
13  Project 14   200000  Finished   10000.0
```

0.5

0.5.1 *Task 6.1: Demoting employees level by 1 who failed to complete the project 1*

Demote the designation level by 1, whose projects have status "fail". Delete the employees record whose designation level is above 4.

[28]:

```
# Using Lambda function to check the status and decrease the 'Designation␣
 ↪Level' by 1 when 'Status' is Failed

Final['Designation Level'] = Final.apply(lambda row: row['Designation Level']-1␣
 ↪if row['Status'] == 'Failed' else row['Designation Level'], axis = 1)
```

[29]:

```
Final.head(14)
```

[29]:

```
      ID First_name  Last_name Gender      City  Age  Designation Level  \
0   A001       John      Alter      M     Paris   25                  2
1   A002      Alice  Luxumberg      F    London   27                  2
2   A003        Tom  Sabestine      M    Berlin   29                  3
3   A004       Nina      Adgra      F  New York   31                  2
4   A005        Amy      Johny      F    Madrid   30                  3
5   A002      Alice  Luxumberg      F    London   27                  1
6   A005        Amy      Johny      F    Madrid   30                  3
7   A003        Tom  Sabestine      M    Berlin   29                  2
8   A001       John      Alter      M     Paris   25                  2
9   A003        Tom  Sabestine      M    Berlin   29                  3
10  A001       John      Alter      M     Paris   25                  1
11  A004       Nina      Adgra      F  New York   31                  2
12  A004       Nina      Adgra      F  New York   31                  2
13  A005        Amy      Johny      F    Madrid   30                  3

       Project     Cost    Status     Bonus
0    Project 1  1002000  Finished   50100.0
1    Project 2  2000000   Ongoing       NaN
2    Project 3  4500000  Finished  225000.0
3    Project 4  5500000   Ongoing       NaN
4    Project 5  3250500  Finished  162525.0
5    Project 6   680000    Failed       NaN
6    Project 7   400000  Finished   20000.0
7    Project 8   350000    Failed       NaN
8    Project 9  2061714   Ongoing       NaN
9   Project 10   300000  Finished   15000.0
10  Project 11  2000000    Failed       NaN
11  Project 12  1000000   Ongoing       NaN
12  Project 13  3000000  Finished  150000.0
```

```
13  Project 14   200000  Finished   10000.0
```

0.6

0.6.1 *Task 7: Add 'Mr.' or 'Mrs.' based on gender to First Name and remove 'Gender'*

Add "Mr." and "Mrs." to the first name column and drop the gender column.

[30]:

```
# Using Lambda function and based on 'Gender' adding 'Mr.' and 'Mrs.' to First␣
↪Name of the Employees

Final['First_name'] = Final.apply(lambda row: 'Mr. '+row['First_name'] if␣
↪row['Gender'] == 'M' else 'Mrs. '+row['First_name'], axis = 1)
```

[31]:

```
Final.head(14)
```

[31]:

```
      ID  First_name  Last_name Gender      City  Age  Designation Level  \
0   A001    Mr. John      Alter      M     Paris   25                  2
1   A002  Mrs. Alice  Luxumberg      F    London   27                  2
2   A003     Mr. Tom  Sabestine      M    Berlin   29                  3
3   A004   Mrs. Nina      Adgra      F  New York   31                  2
4   A005    Mrs. Amy      Johny      F    Madrid   30                  3
5   A002  Mrs. Alice  Luxumberg      F    London   27                  1
6   A005    Mrs. Amy      Johny      F    Madrid   30                  3
7   A003     Mr. Tom  Sabestine      M    Berlin   29                  2
8   A001    Mr. John      Alter      M     Paris   25                  2
9   A003     Mr. Tom  Sabestine      M    Berlin   29                  3
10  A001    Mr. John      Alter      M     Paris   25                  1
11  A004   Mrs. Nina      Adgra      F  New York   31                  2
12  A004   Mrs. Nina      Adgra      F  New York   31                  2
13  A005    Mrs. Amy      Johny      F    Madrid   30                  3

       Project     Cost    Status     Bonus
0    Project 1  1002000  Finished   50100.0
1    Project 2  2000000   Ongoing       NaN
2    Project 3  4500000  Finished  225000.0
3    Project 4  5500000   Ongoing       NaN
4    Project 5  3250500  Finished  162525.0
5    Project 6   680000    Failed       NaN
6    Project 7   400000  Finished   20000.0
7    Project 8   350000    Failed       NaN
8    Project 9  2061714   Ongoing       NaN
9   Project 10   300000  Finished   15000.0
10  Project 11  2000000    Failed       NaN
11  Project 12  1000000   Ongoing       NaN
12  Project 13  3000000  Finished  150000.0
13  Project 14   200000  Finished   10000.0
```

[32]:
```
# Dropping the "Gender" column from the 'Final' DataFrame

Final.drop(columns = ['Gender'], inplace = True)
```

[33]:
```
Final.head(14)
```

[33]:
```
      ID First_name  Last_name      City  Age  Designation Level     Project  \
0   A001   Mr. John      Alter     Paris   25                  2   Project 1
1   A002 Mrs. Alice  Luxumberg    London   27                  2   Project 2
2   A003    Mr. Tom  Sabestine    Berlin   29                  3   Project 3
3   A004  Mrs. Nina      Adgra  New York   31                  2   Project 4
4   A005   Mrs. Amy      Johny    Madrid   30                  3   Project 5
5   A002 Mrs. Alice  Luxumberg    London   27                  1   Project 6
6   A005   Mrs. Amy      Johny    Madrid   30                  3   Project 7
7   A003    Mr. Tom  Sabestine    Berlin   29                  2   Project 8
8   A001   Mr. John      Alter     Paris   25                  2   Project 9
9   A003    Mr. Tom  Sabestine    Berlin   29                  3  Project 10
10  A001   Mr. John      Alter     Paris   25                  1  Project 11
11  A004  Mrs. Nina      Adgra  New York   31                  2  Project 12
12  A004  Mrs. Nina      Adgra  New York   31                  2  Project 13
13  A005   Mrs. Amy      Johny    Madrid   30                  3  Project 14

       Cost    Status     Bonus
0   1002000  Finished   50100.0
1   2000000   Ongoing       NaN
2   4500000  Finished  225000.0
3   5500000   Ongoing       NaN
4   3250500  Finished  162525.0
5    680000    Failed       NaN
6    400000  Finished   20000.0
7    350000    Failed       NaN
8   2061714   Ongoing       NaN
9    300000  Finished   15000.0
10  2000000    Failed       NaN
11  1000000   Ongoing       NaN
12  3000000  Finished  150000.0
13   200000  Finished   10000.0
```

0.7

0.7.1 *Task 8: Promoting employee's level by 1 whose Age>29*

Promote designation level by 1 for the employees whose age is more than 29 years using IF condition.

[34]:
```
# Adding +1 to the Designation_level whose age is greater than 29 using Lambda␣
 ↪and If condition function
```

```
Final['Designation Level'] = Final.apply(lambda row: row['Designation Level']+1␣
 ↪if row['Age'] > 29 else row['Designation Level'], axis = 1)
```

[35]:

```
Final.head(14)
```

[35]:

```
      ID  First_name  Last_name      City  Age  Designation Level     Project  \
0   A001    Mr. John      Alter     Paris   25                  2   Project 1
1   A002  Mrs. Alice  Luxumberg    London   27                  2   Project 2
2   A003     Mr. Tom  Sabestine    Berlin   29                  3   Project 3
3   A004   Mrs. Nina      Adgra  New York   31                  3   Project 4
4   A005    Mrs. Amy      Johny    Madrid   30                  4   Project 5
5   A002  Mrs. Alice  Luxumberg    London   27                  1   Project 6
6   A005    Mrs. Amy      Johny    Madrid   30                  4   Project 7
7   A003     Mr. Tom  Sabestine    Berlin   29                  2   Project 8
8   A001    Mr. John      Alter     Paris   25                  2   Project 9
9   A003     Mr. Tom  Sabestine    Berlin   29                  3  Project 10
10  A001    Mr. John      Alter     Paris   25                  1  Project 11
11  A004   Mrs. Nina      Adgra  New York   31                  3  Project 12
12  A004   Mrs. Nina      Adgra  New York   31                  3  Project 13
13  A005    Mrs. Amy      Johny    Madrid   30                  4  Project 14

       Cost    Status     Bonus
0   1002000  Finished   50100.0
1   2000000   Ongoing       NaN
2   4500000  Finished  225000.0
3   5500000   Ongoing       NaN
4   3250500  Finished  162525.0
5    680000    Failed       NaN
6    400000  Finished   20000.0
7    350000    Failed       NaN
8   2061714   Ongoing       NaN
9    300000  Finished   15000.0
10  2000000    Failed       NaN
11  1000000   Ongoing       NaN
12  3000000  Finished  150000.0
13   200000  Finished   10000.0
```

0.8

0.8.1 *Task 6.2: Deleting the records of employee with Designation Level >4 2*

Demote the designation level by 1, whose projects have status "fail". Delete the employees record whose designation level is above 4.

[36]:

```
# Removing recoreds having Designation Level greater than 4

Final = Final[~(Final['Designation Level']>4)]
```

[37]:
```
Final.head(14)
```

[37]:
```
      ID  First_name  Last_name      City  Age  Designation Level     Project  \
0   A001    Mr. John      Alter     Paris   25                  2   Project 1
1   A002  Mrs. Alice  Luxumberg    London   27                  2   Project 2
2   A003     Mr. Tom  Sabestine    Berlin   29                  3   Project 3
3   A004   Mrs. Nina      Adgra  New York   31                  3   Project 4
4   A005    Mrs. Amy      Johny    Madrid   30                  4   Project 5
5   A002  Mrs. Alice  Luxumberg    London   27                  1   Project 6
6   A005    Mrs. Amy      Johny    Madrid   30                  4   Project 7
7   A003     Mr. Tom  Sabestine    Berlin   29                  2   Project 8
8   A001    Mr. John      Alter     Paris   25                  2   Project 9
9   A003     Mr. Tom  Sabestine    Berlin   29                  3  Project 10
10  A001    Mr. John      Alter     Paris   25                  1  Project 11
11  A004   Mrs. Nina      Adgra  New York   31                  3  Project 12
12  A004   Mrs. Nina      Adgra  New York   31                  3  Project 13
13  A005    Mrs. Amy      Johny    Madrid   30                  4  Project 14

       Cost    Status     Bonus
0   1002000  Finished   50100.0
1   2000000   Ongoing       NaN
2   4500000  Finished  225000.0
3   5500000   Ongoing       NaN
4   3250500  Finished  162525.0
5    680000    Failed       NaN
6    400000  Finished   20000.0
7    350000    Failed       NaN
8   2061714   Ongoing       NaN
9    300000  Finished   15000.0
10  2000000    Failed       NaN
11  1000000   Ongoing       NaN
12  3000000  Finished  150000.0
13   200000  Finished   10000.0
```

0.9

0.9.1 *Task 9: Add 'Cost' of All Project by Employee*

Add the cost of all projects for each Employee and save it in new dataframe "TotalProjCost" with three columns ID, First Name, and Total cost

[38]:
```
# Used Groupby function ot create the 'TotalProjCost' Dataset from Final

TotalProjCost = Final.groupby(['ID','First_name'] , as_index = False)['Cost'].
 ↪sum()
```

[39]:
```
TotalProjCost.head()
```

[39]:
```
     ID  First_name     Cost
0  A001    Mr. John  5063714
1  A002  Mrs. Alice  2680000
2  A003     Mr. Tom  5150000
3  A004   Mrs. Nina  9500000
4  A005    Mrs. Amy  3850500
```

[40]:
```python
# Renamed the 'Cost' column to 'Total Cost' as requested in the Question

TotalProjCost.rename(columns = {'Cost': 'Total Cost'}, inplace = True)
```

[41]:
```python
TotalProjCost.head()
```

[41]:
```
     ID  First_name  Total Cost
0  A001    Mr. John     5063714
1  A002  Mrs. Alice     2680000
2  A003     Mr. Tom     5150000
3  A004   Mrs. Nina     9500000
4  A005    Mrs. Amy     3850500
```

0.10

0.10.1 *Task 10: Print 'Employee' details whose 'City' contains 'o' in it*

Print all the employee details whose city name contains the letter "o" in it.

[42]:
```python
# Printed all the details present in the 'Final' dataset for the Employee whose 'City' contains 'o'

print(Final[Final['City'].str.contains('o' , case = False, na = False)])
```

```
      ID  First_name  Last_name      City  Age  Designation Level     Project  \
1    A002  Mrs. Alice  Luxumberg    London   27                  2   Project 2
3    A004   Mrs. Nina      Adgra  New York   31                  3   Project 4
5    A002  Mrs. Alice  Luxumberg    London   27                  1   Project 6
11   A004   Mrs. Nina      Adgra  New York   31                  3  Project 12
12   A004   Mrs. Nina      Adgra  New York   31                  3  Project 13

       Cost    Status     Bonus
1   2000000   Ongoing       NaN
3   5500000   Ongoing       NaN
5    680000    Failed       NaN
11  1000000   Ongoing       NaN
12  3000000  Finished  150000.0
```

[43]:
```python
#Printed only the Personal Details for the Employee with 'City' containing 'o' from 'Final' table

```

```
print(Final[Final['City'].str.contains('o' , case = False, na = False)][['ID',␣
↪'First_name', 'Last_name', 'City', 'Age', 'Designation Level']].
↪drop_duplicates())
```

```
     ID  First_name  Last_name      City  Age  Designation Level
1  A002  Mrs. Alice  Luxumberg    London   27                  2
3  A004   Mrs. Nina      Adgra  New York   31                  3
5  A002  Mrs. Alice  Luxumberg    London   27                  1
```

[44]:

```
#Printed only the Personal Details for the Employee with 'City' containing 'o'␣
↪from 'Employee' table

print(df_emp_up[df_emp_up['City'].str.contains('o' , case = False, na = False)])
```

```
     ID First_name  Last_name Gender      City  Age
1  A002      Alice  Luxumberg      F    London   27
3  A004       Nina      Adgra      F  New York   31
```

0.11