Experiment 07:

Aim : To write a Pandas program to create a Pivot table and find the maximum and minimum sale value of the items.

Source code:

import pandas as pd

df = pd.read\_csv('sales\_data.csv')

pivot\_table = pd.pivot\_table(df, values='Units', index='Item', aggfunc='sum')

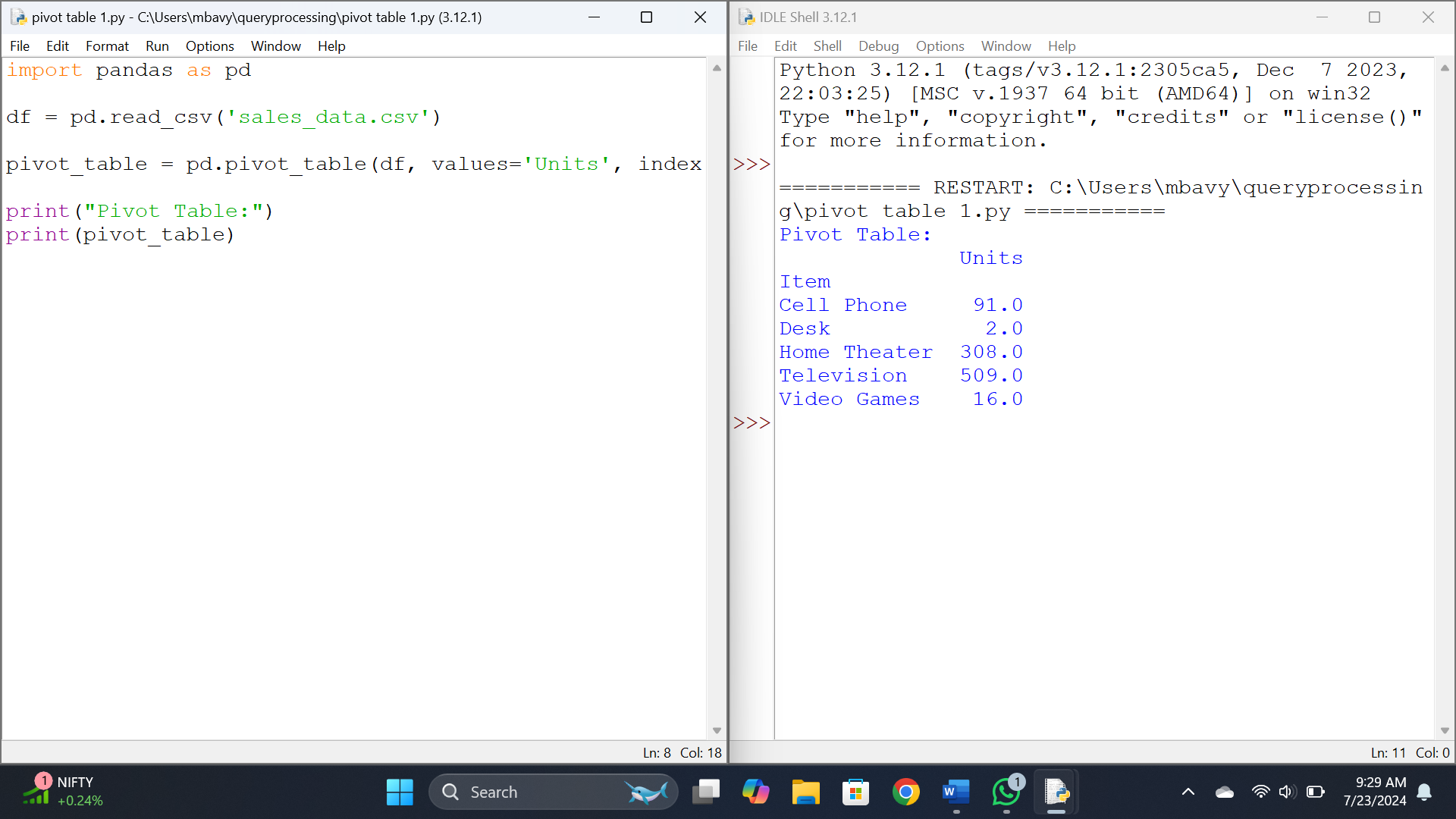
print("Pivot Table:")

print(pivot\_table)

Input:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **OrderDate** | **Region** | **Manager** | **SalesMan** | **Item** | **Units** | **Unit\_price** | **Sale\_amt** |
| 1-6-18 | East | Martha | Alexander | Television | 95 | 1,198.00 | 1,13,810.00 |
| 1-23-18 | Central | Hermann | Shelli | Home Theater | 50 | 500.00 | 25,000.00 |
| 2-9-18 | Central | Hermann | Luis | Television | 36 | 1,198.00 | 43,128.00 |
| 2-26-18 | Central | Timothy | David | Cell Phone | 27 | 225.00 | 6,075.00 |
| 3-15-18 | West | Timothy | Stephen | Television | 56 | 1,198.00 | 67,088.00 |
| 4-1-18 | East | Martha | Alexander | Home Theater | 60 | 500.00 | 30,000.00 |
| 4-18-18 | Central | Martha | Steven | Television | 75 | 1,198.00 | 89,850.00 |
| 5-5-18 | Central | Hermann | Luis | Television | 90 | 1,198.00 | 1,07,820.00 |
| 5-22-18 | West | Douglas | Michael | Television | 32 | 1,198.00 | 38,336.00 |
| 6-8-18 | East | Martha | Alexander | Home Theater | 60 | 500.00 | 30,000.00 |
| 6-25-18 | Central | Hermann | Sigal | Television | 90 | 1,198.00 | 1,07,820.00 |
| 7-12-18 | East | Martha | Diana | Home Theater | 29 | 500.00 | 14,500.00 |
| 7-29-18 | East | Douglas | Karen | Home Theater | 81 | 500.00 | 40,500.00 |
| 8-15-18 | East | Martha | Alexander | Television | 35 | 1,198.00 | 41,930.00 |
| 9-1-18 | Central | Douglas | John | Desk | 2 | 125.00 | 250.00 |
| 9-18-18 | East | Martha | Alexander | Video Games | 16 | 58.50 | 936.00 |
| 10-5-18 | Central | Hermann | Sigal | Home Theater | 28 | 500.00 | 14,000.00 |
| 10-22-18 | East | Martha | Alexander | Cell Phone | 64 | 225.00 | 14,400.00 |

Output:



Results: Thus a Pandas program to create a Pivot table and find the maximum and minimum sale value of the items is done.

Experiment 8 :

Aim :

To write a Pandas program to create a Pivot table and find the item wise unit sold.

Code:

import pandas as pd

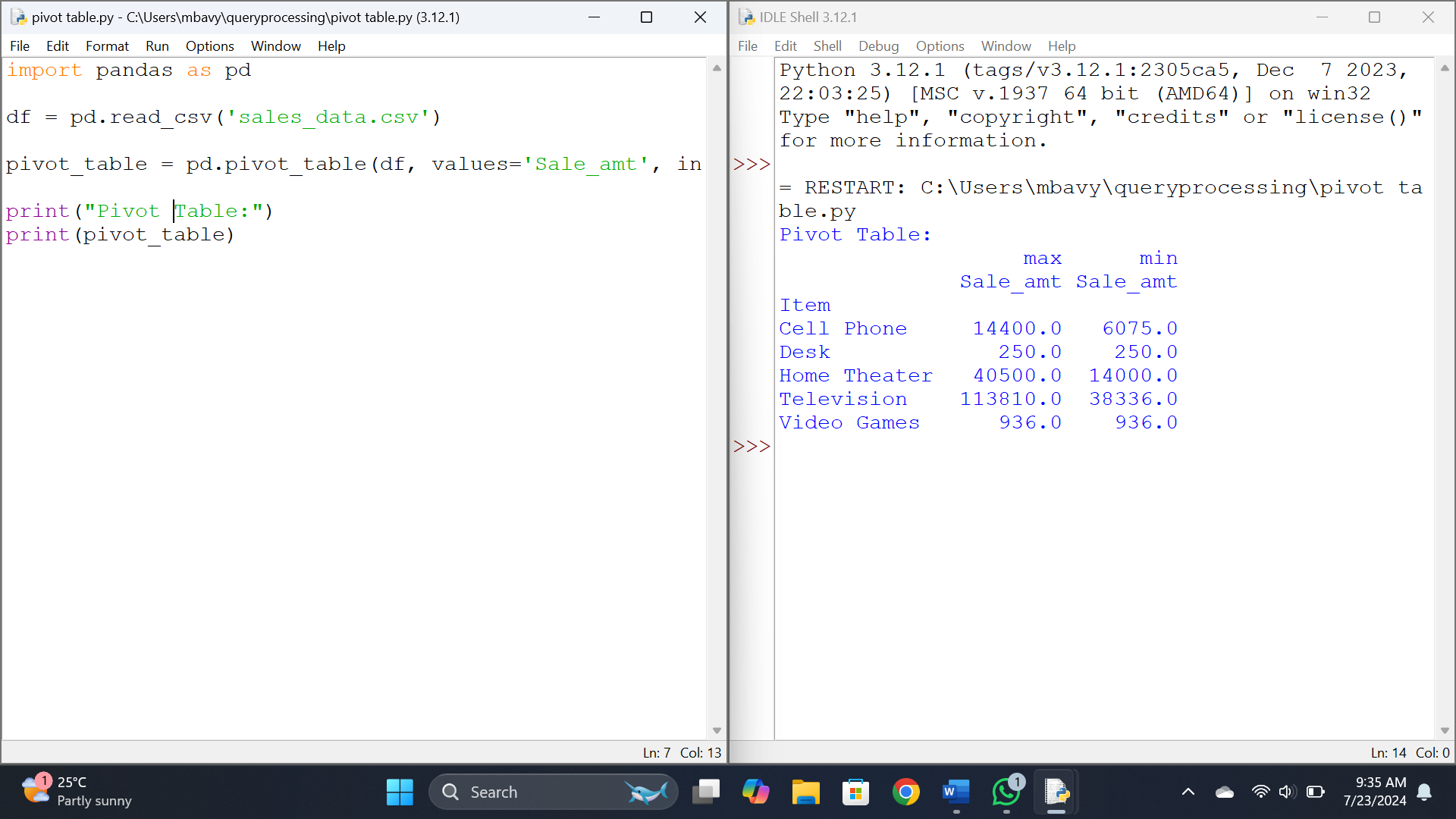
df = pd.read\_csv('sales\_data.csv')

pivot\_table = pd.pivot\_table(df, values='Sale\_amt', index='Item', aggfunc=['max', 'min'])

print("Pivot Table:")

print(pivot\_table)

Output:



Results:

Thus a Pandas program to create a Pivot table and find the item wise unit sold is done.

Experiment 9:

Aim:

To Write a Pandas program to create a Pivot table and find the total sale amount region wise, manager wise, sales man wise.

Code:

import pandas as pd

df = pd.read\_csv('sales\_data.csv')

pivot\_region = pd.pivot\_table(df, values='Sale\_amt', index='Region', aggfunc='sum')

pivot\_manager = pd.pivot\_table(df, values='Sale\_amt', index='Manager', aggfunc='sum')

pivot\_salesman = pd.pivot\_table(df, values='Sale\_amt', index='SalesMan', aggfunc='sum')

print("Total Sale Amount Region-wise:")

print(pivot\_region)

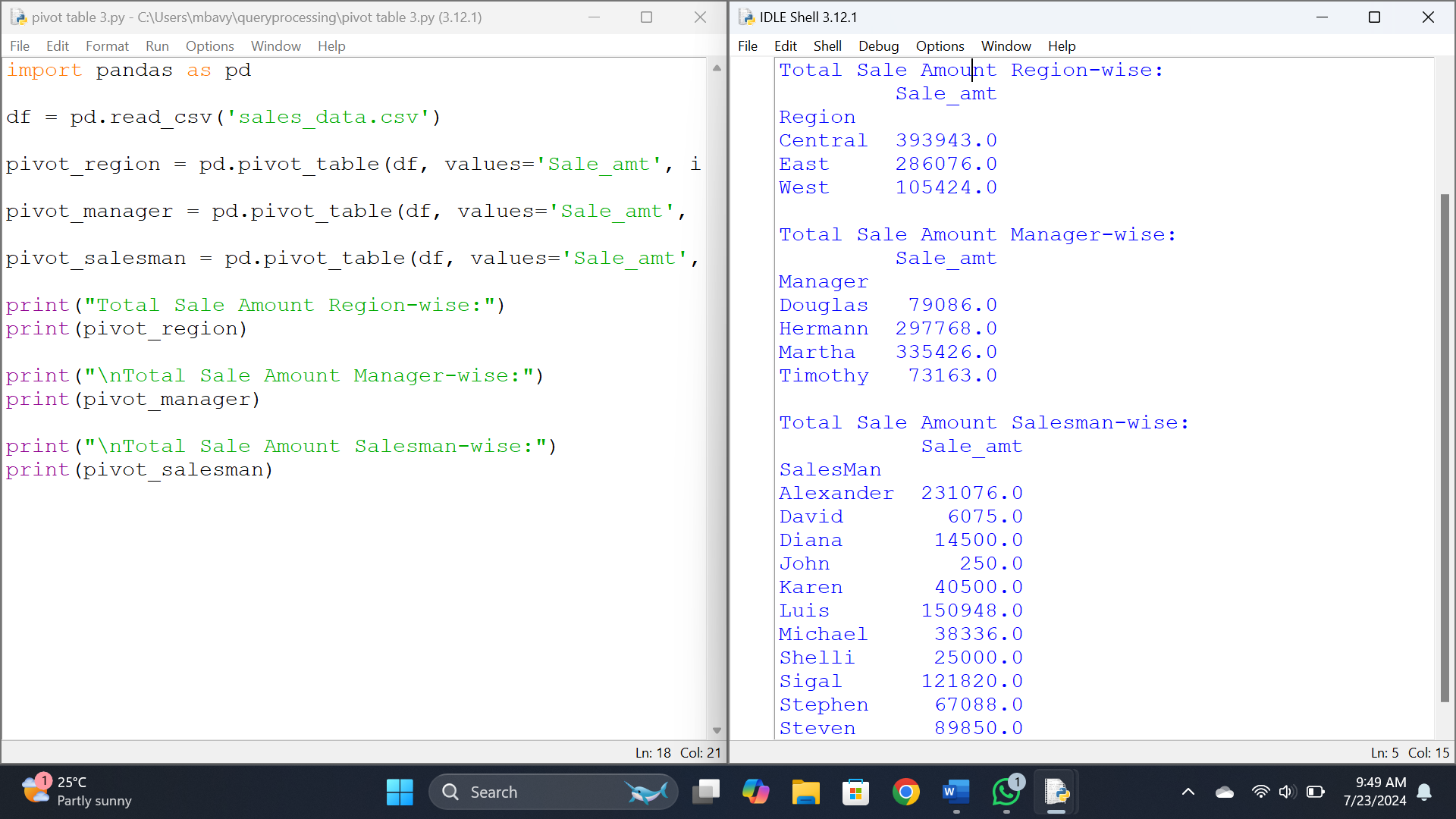
print("\nTotal Sale Amount Manager-wise:")

print(pivot\_manager)

print("\nTotal Sale Amount Salesman-wise:")

print(pivot\_salesman)

Output:



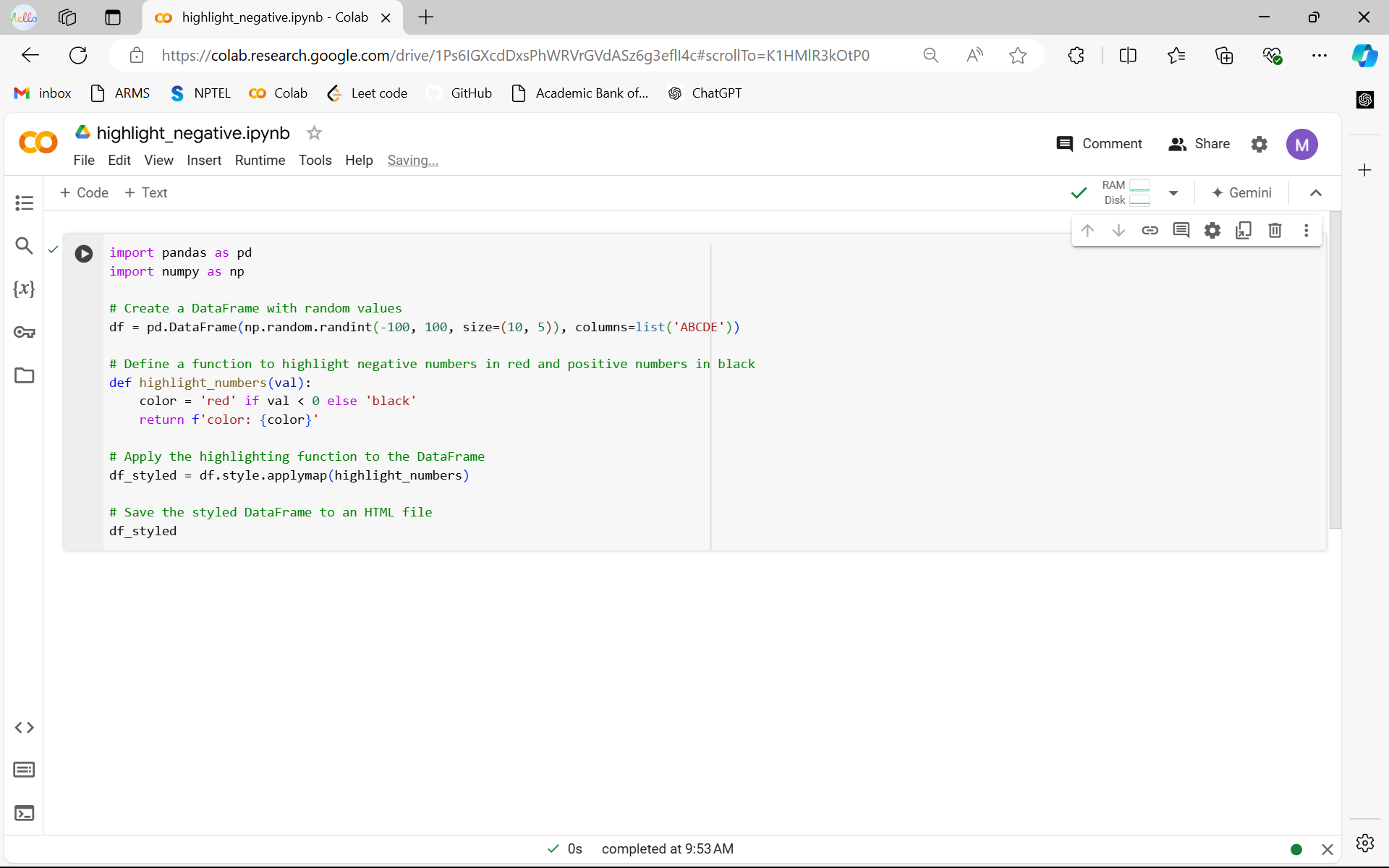
Results:

Thus a Pandas program to create a Pivot table and find the total sale amount region wise, manager wise, sales man wise is done.

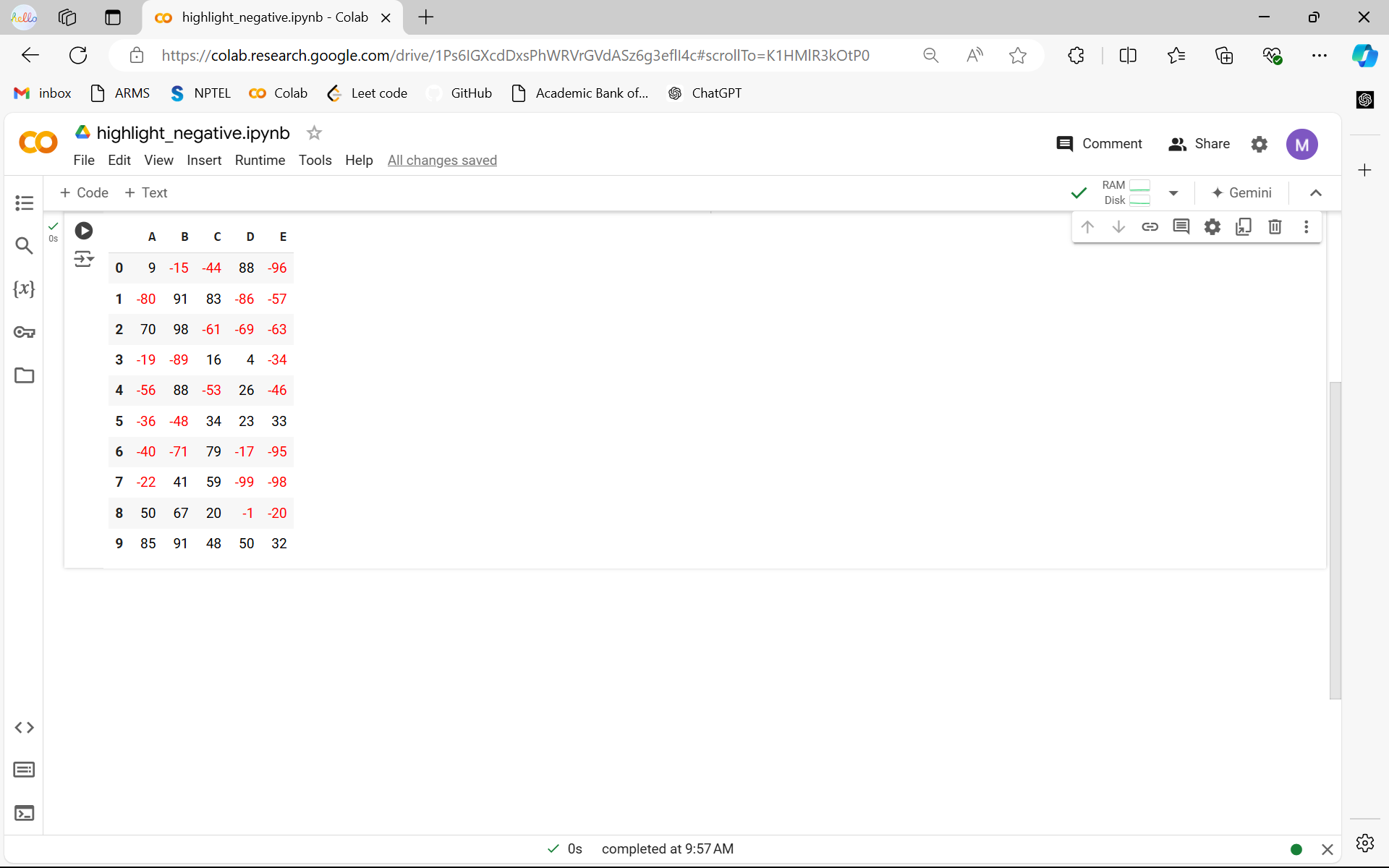
Experiment 10 :

Aim: To create a data frame of ten rows, four columns with random values. Write a Pandas program to highlight the negative numbers red and positive numbers black.

Code:



Output:



Results :

Thus a data frame of ten rows, four columns with random values. Write a Pandas program to highlight the negative numbers red and positive numbers black.

Experiment 11:

Aim: To create a data frame of ten rows, four columns with random values. Convert some values to nan values. Write a Pandas program which will highlight the nan values.

Code:

import pandas as pd

import numpy as np

df = pd.DataFrame(np.random.rand(10, 4), columns=['A', 'B', 'C', 'D'])

nan\_indices = [(1, 2), (3, 0), (5, 1), (7, 3), (9, 0)]

for index in nan\_indices:

df.iat[index] = np.nan

def highlight\_nan(s):

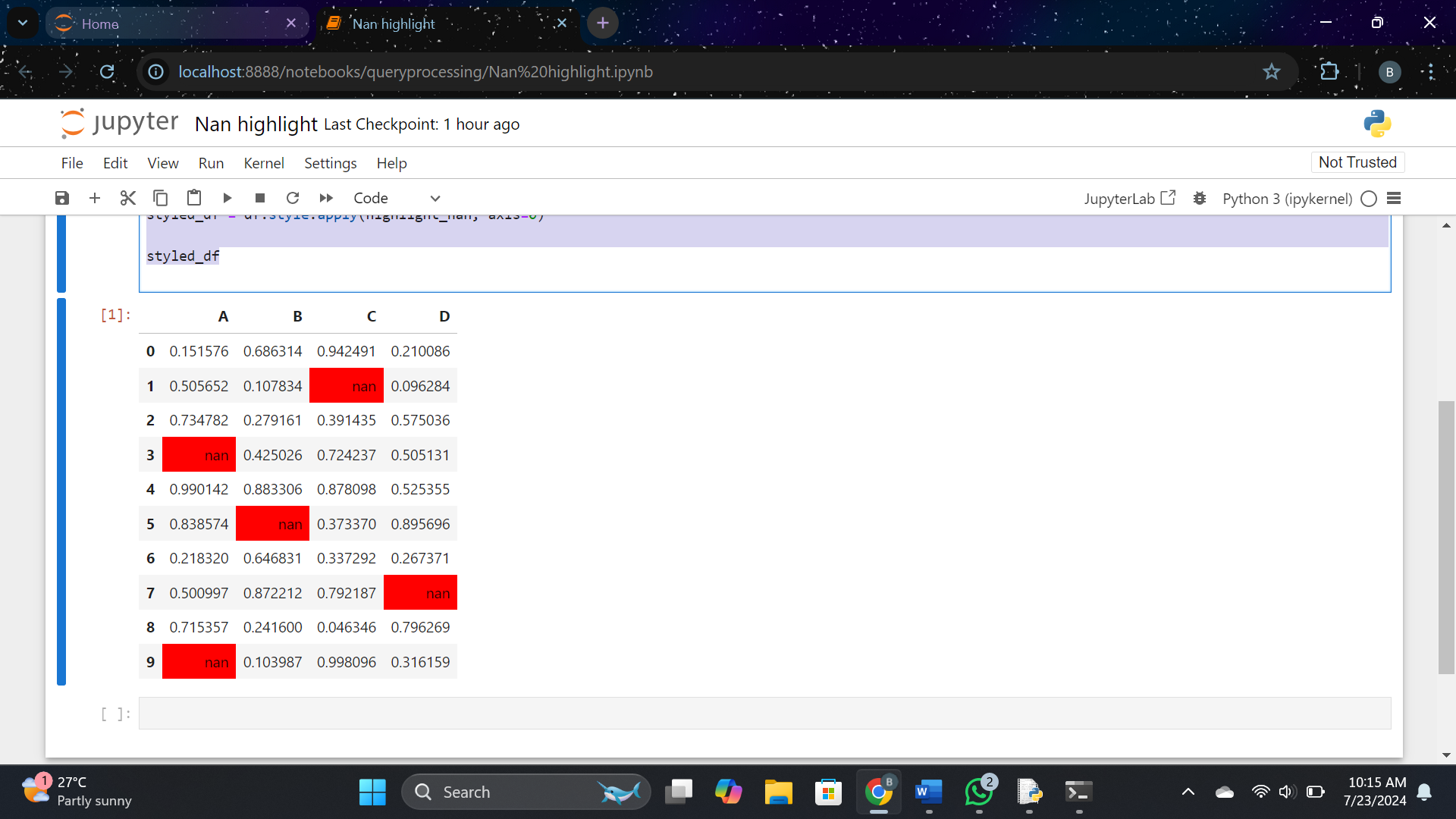
is\_nan = pd.isna(s)

return ['background-color: red' if v else '' for v in is\_nan]

styled\_df = df.style.apply(highlight\_nan, axis=0)

styled\_df

Output:

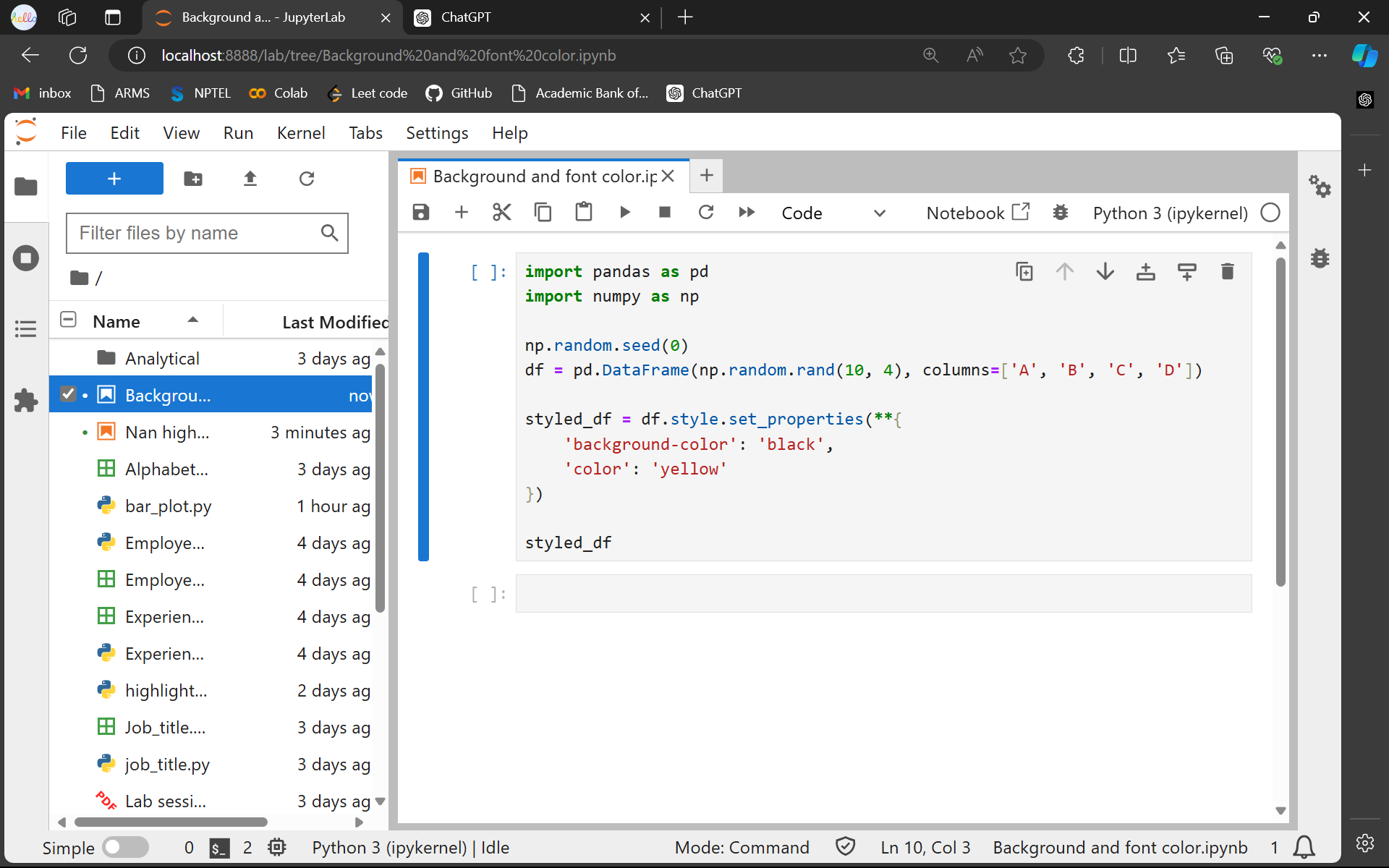


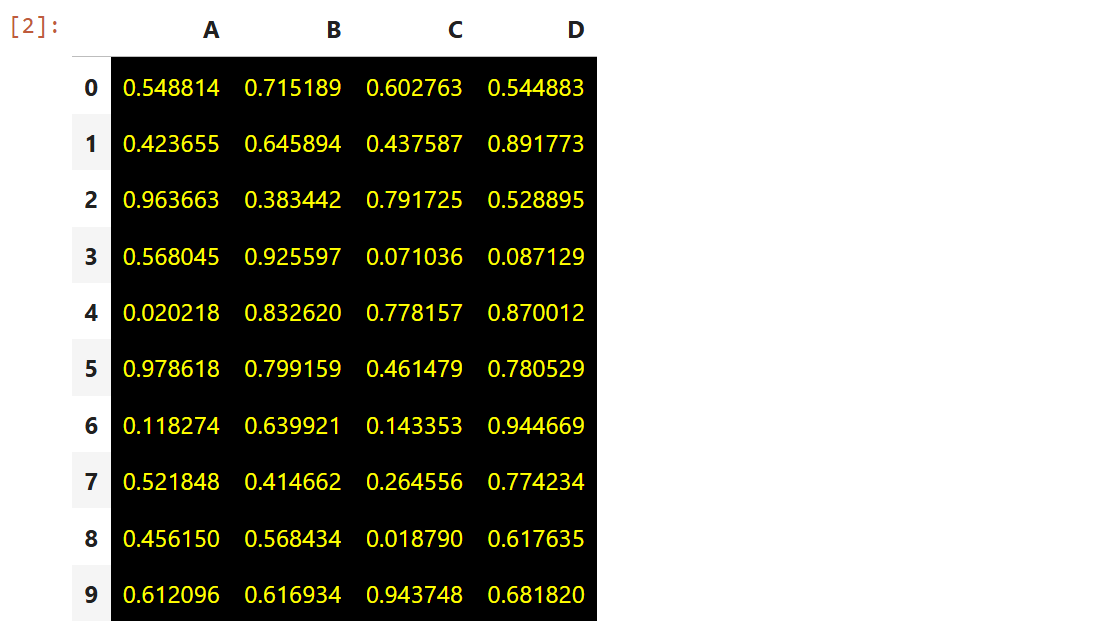
Results :Thus a data frame of ten rows, four columns with random values. Convert some values to nan values. Write a Pandas program which will highlight the nan values.

Experiment 12 :

Aim: To Create a dataframe of ten rows, four columns with random values. Write a Pandas program to set dataframe background Color black and font color yellow

Code:



Output:

Results :

Thus a dataframe of ten rows, four columns with random values. Write a Pandas program to set dataframe background Color black and font color yellow.

Experiment 13 :

Aim: Write a Pandas program to detect missing values of a given DataFrame. Display True or False.

Code:

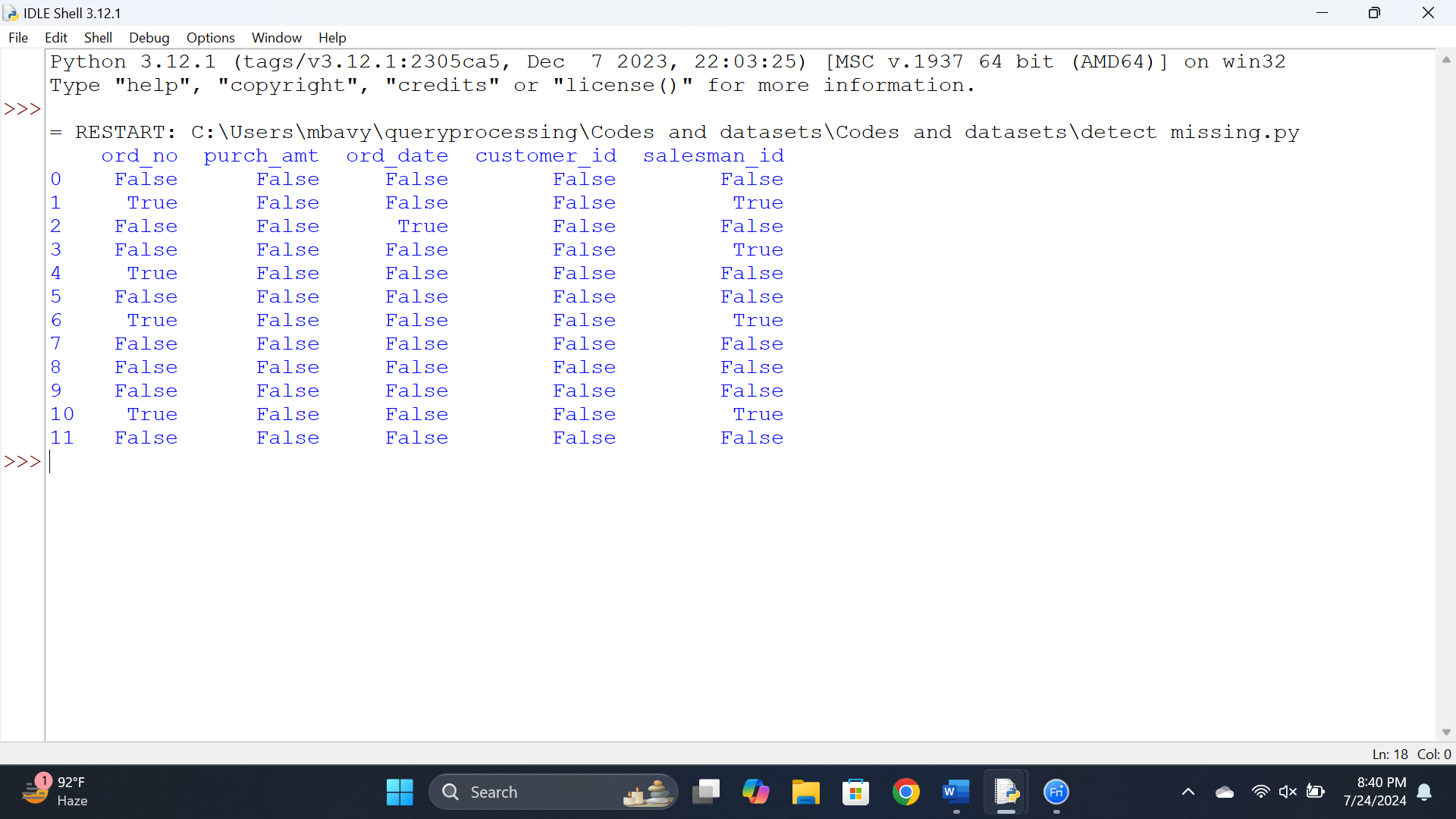
import pandas as pd

df = pd.read\_csv('sales.csv')

missing\_values = df.isnull()

print(missing\_values)

Output:



Results: Write a Pandas program to detect missing values of a given DataFrame. Display True or False.

Experiment 14

Aim: To Write a Pandas program to find and replace the missing values in a given DataFrame which do not have any valuable information.

Code:

import pandas as pd

df = pd.read\_csv('sales.csv')

print("Original DataFrame:")

print(df)

numerical\_columns = ['purch\_amt']

df[numerical\_columns] = df[numerical\_columns].fillna(df[numerical\_columns].mean())

categorical\_columns = ['ord\_no', 'ord\_date', 'customer\_id', 'salesman\_id']

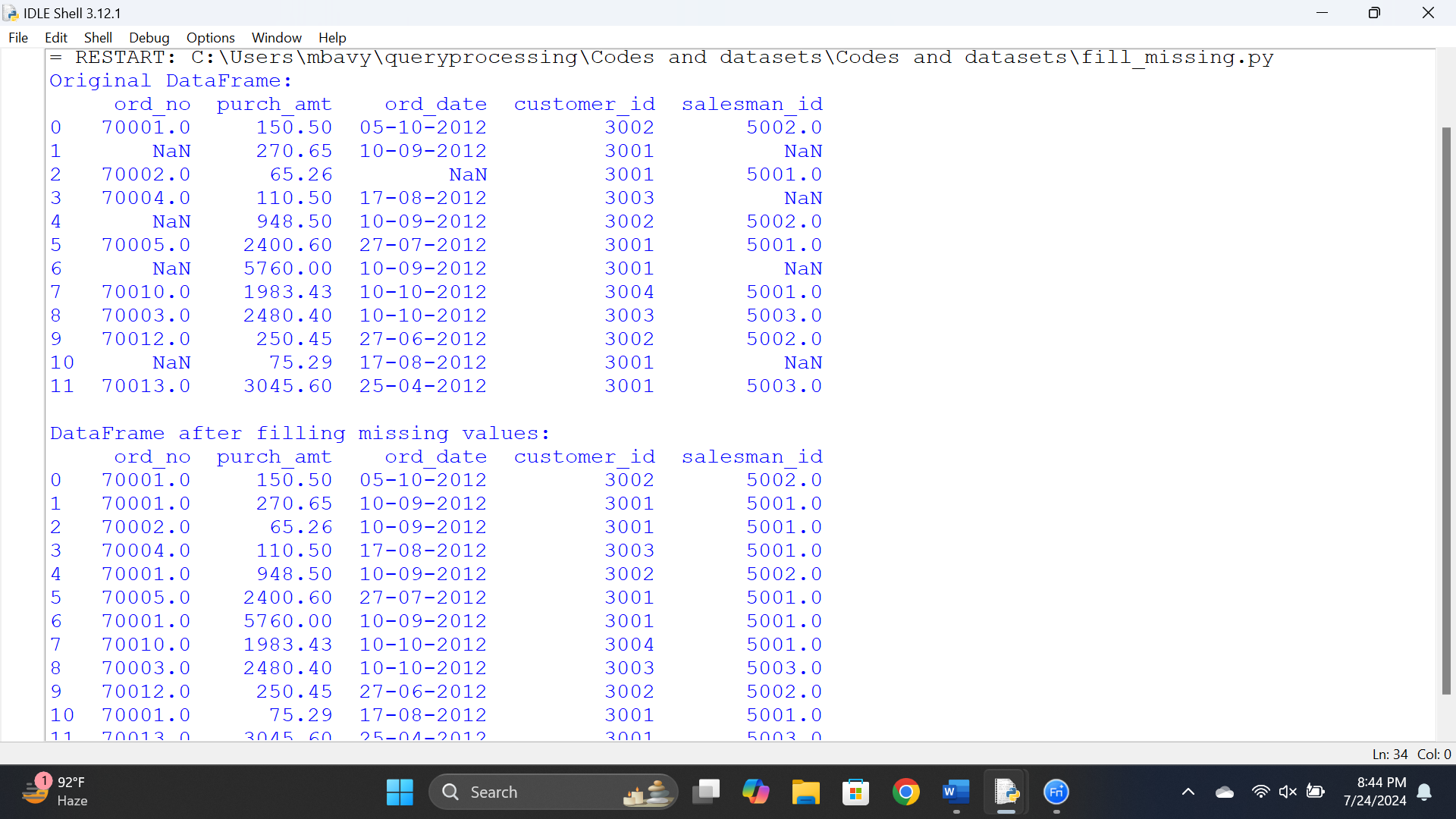
for column in categorical\_columns:

df[column] = df[column].fillna(df[column].mode()[0])

print("\nDataFrame after filling missing values:")

print(df)

Output:



Results :

Thus a Pandas program to find and replace the missing values in a given DataFrame which do not have any valuable information.

Experiment 15

Aim: To Write a Pandas program to keep the rows with at least 2 NaN values in a given DataFrame.

Code :

import pandas as pd

df = pd.read\_csv('sales.csv')

print("Original DataFrame:")

print(df)

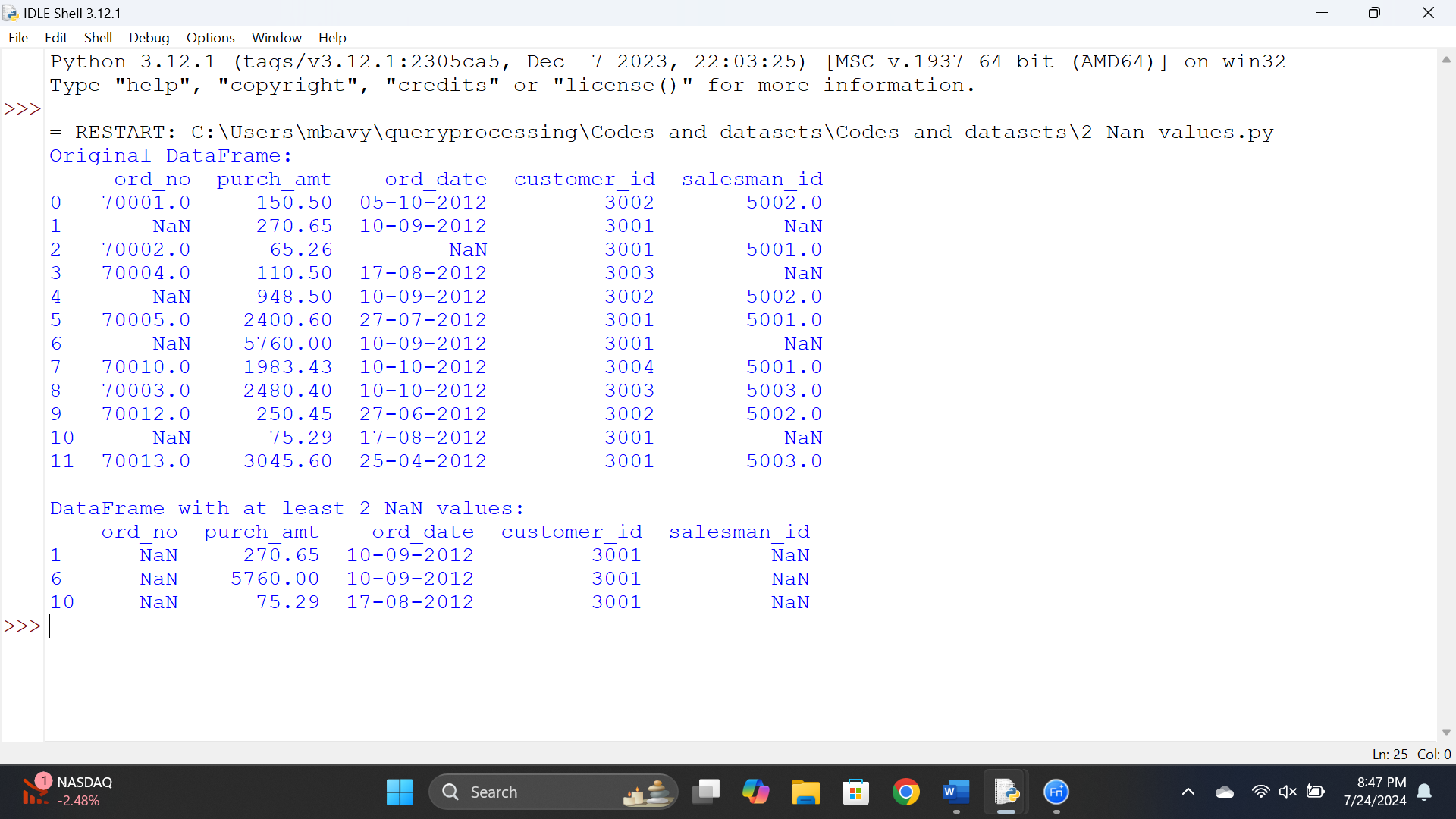
df\_filtered = df[df.isna().sum(axis=1) >= 2]

print("\nDataFrame with at least 2 NaN values:")

print(df\_filtered)

df\_filtered.to\_csv('filtered\_sales.csv', index=False)

Output :



Results :

Thus a Pandas program to keep the rows with at least 2 NaN values in a given DataFrame.

Experiment 16:

Aim: To Write a Pandas program to split the following dataframe into groups based on school code. Also check the type of GroupBy object.

Code:

import pandas as pd

df = pd.read\_csv('school.csv')

grouped = df.groupby('school\_code')

print(type(grouped))

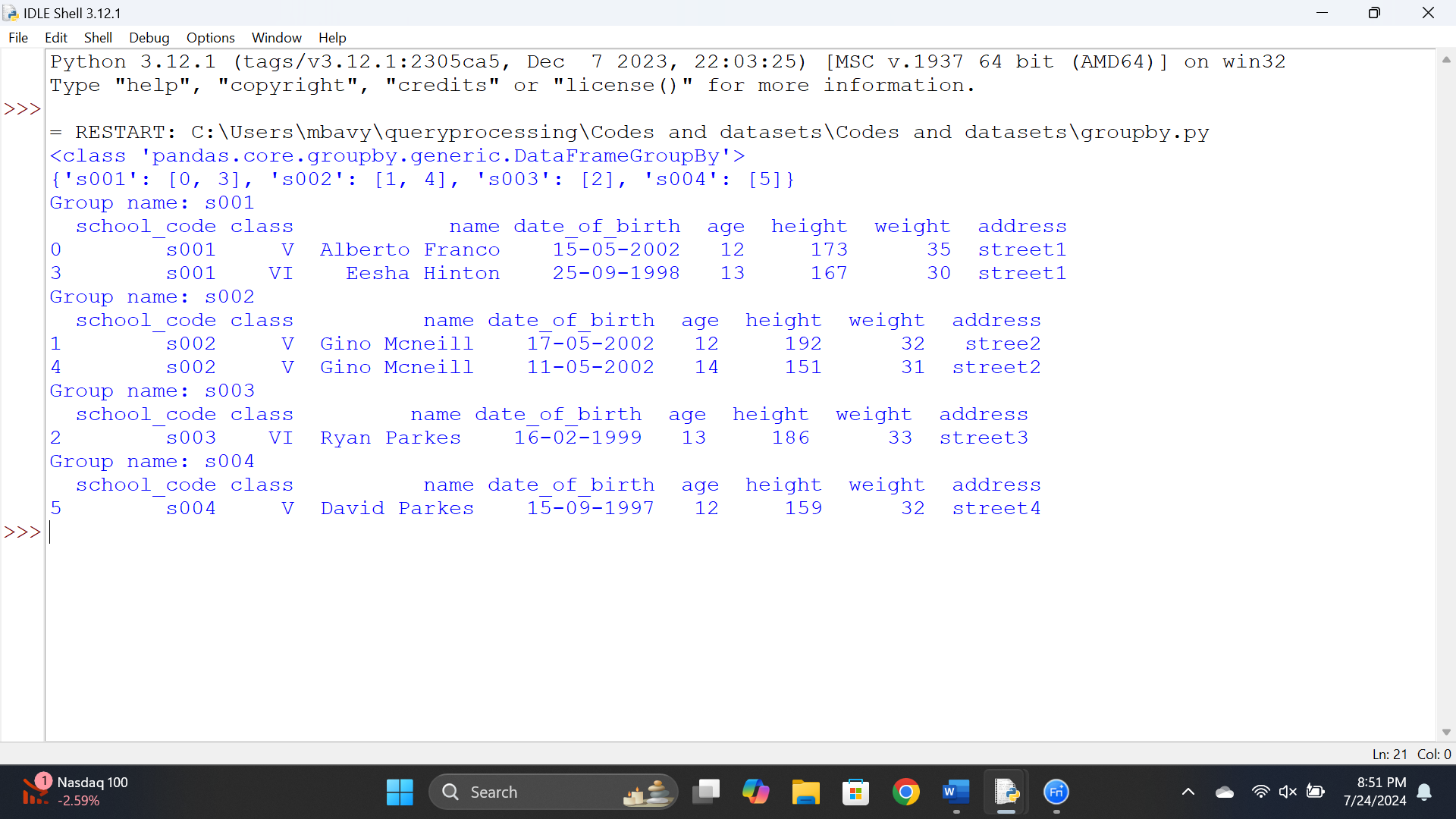
print(grouped.groups)

for name, group in grouped:

print(f"Group name: {name}")

print(group)

Output:



Results :

Thus a Pandas program to split the following dataframe into groups based on school code. Also check the type of GroupBy object.