DAY 1 – QUERY PROCESSING LAB PRACTICAL

1. Write a Pandas program to select distinct department id from employees file.

INPUT DATASET:

+---------------+----------------------+------------+-------------+

| DEPARTMENT\_ID | DEPARTMENT\_NAME | MANAGER\_ID | LOCATION\_ID |

+---------------+----------------------+------------+-------------+

| 10 | Administration | 200 | 1700 |

| 20 | Marketing | 201 | 1800 |

| 30 | Purchasing | 114 | 1700 |

| 40 | Human Resources | 203 | 2400 |

| 50 | Shipping | 121 | 1500 |

| 60 | IT | 103 | 1400 |

| 70 | Public Relations | 204 | 2700 |

| 80 | Sales | 145 | 2500 |

| 90 | Executive | 100 | 1700 |

| 100 | Finance | 108 | 1700 |

| 110 | Accounting | 205 | 1700 |

| 120 | Treasury | 0 | 1700 |

| 130 | Corporate Tax | 0 | 1700 |

| 140 | Control And Credit | 0 | 1700 |

| 150 | Shareholder Services | 0 | 1700 |

| 160 | Benefits | 0 | 1700 |

| 170 | Manufacturing | 0 | 1700 |

| 180 | Construction | 0 | 1700 |

| 190 | Contracting | 0 | 1700 |

| 200 | Operations | 0 | 1700 |

| 210 | IT Support | 0 | 1700 |

| 220 | NOC | 0 | 1700 |

| 230 | IT Helpdesk | 0 | 1700 |

| 240 | Government Sales | 0 | 1700 |

| 250 | Retail | 0 | 1700 |

+---------------+----------------------+------------+-------------

CODE:

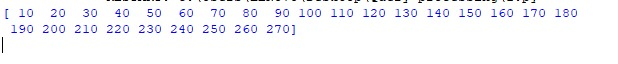
import pandas as pd

employees = pd.read\_csv(r"C:\Users\LENOVO\Desktop\New folder\lab 1 query.csv")

distinct\_department\_ids = employees['DEPARTMENT\_ID'].unique()

print(distinct\_department\_ids)

OUTPUT:



1. Write a Pandas program to display the ID for those employees who did two or more jobs in the past.

+-------------+------------+------------+------------+---------------+

| EMPLOYEE\_ID | START\_DATE | END\_DATE | JOB\_ID | DEPARTMENT\_ID |

+-------------+------------+------------+------------+---------------+

| 102 | 2001-01-13 | 2006-07-24 | IT\_PROG | 60 |

| 101 | 1997-09-21 | 2001-10-27 | AC\_ACCOUNT | 110 |

| 101 | 2001-10-28 | 2005-03-15 | AC\_MGR | 110 |

| 201 | 2004-02-17 | 2007-12-19 | MK\_REP | 20 |

| 114 | 2006-03-24 | 2007-12-31 | ST\_CLERK | 50 |

| 122 | 2007-01-01 | 2007-12-31 | ST\_CLERK | 50 |

| 200 | 1995-09-17 | 2001-06-17 | AD\_ASST | 90 |

| 176 | 2006-03-24 | 2006-12-31 | SA\_REP | 80 |

| 176 | 2007-01-01 | 2007-12-31 | SA\_MAN | 80 |

| 200 | 2002-07-01 | 2006-12-31 | AC\_ACCOUNT | 90 |

+-------------+------------+------------+------------+---------------+

CODE:

import pandas as pd

data={'employee\_ID':[102,101,101,201,114,122,200,176,176,200],

'Start\_date':['13-01-2001','21-09-1997','28-10-2001','17-02-2004',

'24-03-2006','01-01-2007','17-09-1995','24-03-2006','01-01-2007','01-07-2002'],

'End\_date':['24-07-2006','27-10-2001','15-03-2005','19-12-2007','31-12-2007','31-12-2007','17-06-2001','31-12-2006','31-12-2007','31-12-2006'],

'Job\_ID':['IT\_PROG','AC\_ACCOUNT','AC\_MGR','MK\_REP','ST\_CLER','ST\_CLERK','AD\_ASST','SA\_REP','SA\_MAN','AC\_ACCOUNT'],

'Department\_ID':[60,110,110,20,50,50,90,80,80,90]

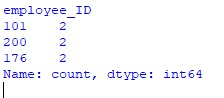
}

df=pd.DataFrame(data)

data2=df['employee\_ID'].value\_counts()

print(data2[data2>1])

OUTPUT:



1. Write a Pandas program to display the details of jobs in descending sequence on job title.

+------------+---------------------------------+------------+------------+

| JOB\_ID | JOB\_TITLE | MIN\_SALARY | MAX\_SALARY |

+------------+---------------------------------+------------+------------+

| AD\_PRES | President | 20080 | 40000 |

| AD\_VP | Administration Vice President | 15000 | 30000 |

| AD\_ASST | Administration Assistant | 3000 | 6000 |

| FI\_MGR | Finance Manager | 8200 | 16000 |

| FI\_ACCOUNT | Accountant | 4200 | 9000 |

| AC\_MGR | Accounting Manager | 8200 | 16000 |

| AC\_ACCOUNT | Public Accountant | 4200 | 9000 |

| SA\_MAN | Sales Manager | 10000 | 20080 |

| SA\_REP | Sales Representative | 6000 | 12008 |

| PU\_MAN | Purchasing Manager | 8000 | 15000 |

| PU\_CLERK | Purchasing Clerk | 2500 | 5500 |

| ST\_MAN | Stock Manager | 5500 | 8500 |

| ST\_CLERK | Stock Clerk | 2008 | 5000 |

| SH\_CLERK | Shipping Clerk | 2500 | 5500 |

| IT\_PROG | Programmer | 4000 | 10000 |

| MK\_MAN | Marketing Manager | 9000 | 15000 |

| MK\_REP | Marketing Representative | 4000 | 9000 |

| HR\_REP | Human Resources Representative | 4000 | 9000 |

| PR\_REP | Public Relations Representative | 4500 | 10500 |

CODE:

import pandas as pd

data = {

'JOB\_ID': ['AD\_PRES', 'AD\_VP', 'AD\_ASST', 'FI\_MGR', 'FI\_ACCOUNT', 'AC\_MGR', 'AC\_ACCOUNT', 'SA\_MAN', 'SA\_REP', 'PU\_MAN', 'PU\_CLERK', 'ST\_MAN', 'ST\_CLERK', 'SH\_CLERK', 'IT\_PROG', 'MK\_MAN', 'MK\_REP', 'HR\_REP', 'PR\_REP'],

'JOB\_TITLE': ['President', 'Administration Vice President', 'Administration Assistant', 'Finance Manager', 'Accountant', 'Accounting Manager', 'Public Accountant', 'Sales Manager', 'Sales Representative', 'Purchasing Manager', 'Purchasing Clerk', 'Stock Manager', 'Stock Clerk', 'Shipping Clerk', 'Programmer', 'Marketing Manager', 'Marketing Representative', 'Human Resources Representative', 'Public Relations Representative'],

'MIN\_SALARY': [20080, 15000, 3000, 8200, 4200, 8200, 4200, 10000, 6000, 8000, 2500, 5500, 2008, 2500, 4000, 9000, 4000, 4000, 4500],

'MAX\_SALARY': [40000, 30000, 6000, 16000, 9000, 16000, 9000, 20080, 12008, 15000, 5500, 8500, 5000, 5500, 10000, 15000, 9000, 9000, 10500]

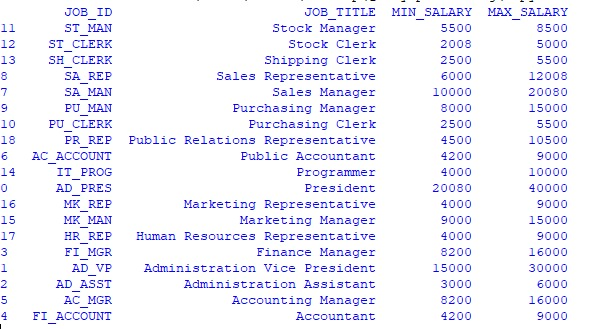
}

df = pd.DataFrame(data)

sorted\_df = df.sort\_values(by='JOB\_TITLE', ascending=False)

print(sorted\_df)

OUTPUT:



1. Write a Pandas program to create a line plot of the historical stock prices of Alphabet Inc. between two specific dates.

CODE:

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv(r'C:\Users\LENOVO\Downloads\WhatsApp Image 2023-11-02 at 09.39.30\_85640aa8.jpg.csv')

# Convert the 'date' column to a datetime object

df['date'] = pd.to\_datetime(df['date'], format='%d-%m-%Y')

# Define the start and end dates

start\_date = '2020-04-06'

end\_date = '2020-04-23'

# Filter the data to include only the rows within the specified date range

filtered\_data = df[(df['date'] >= start\_date) & (df['date'] <= end\_date)]

# Create a line plot of the historical stock prices

plt.figure(figsize=(12, 6))

plt.plot(filtered\_data['date'], filtered\_data['close'], marker='o', linestyle='-', color='b', label='Alphabet Inc. Stock Price')

plt.title('Historical Stock Prices of Alphabet Inc.')

plt.xlabel('Date')

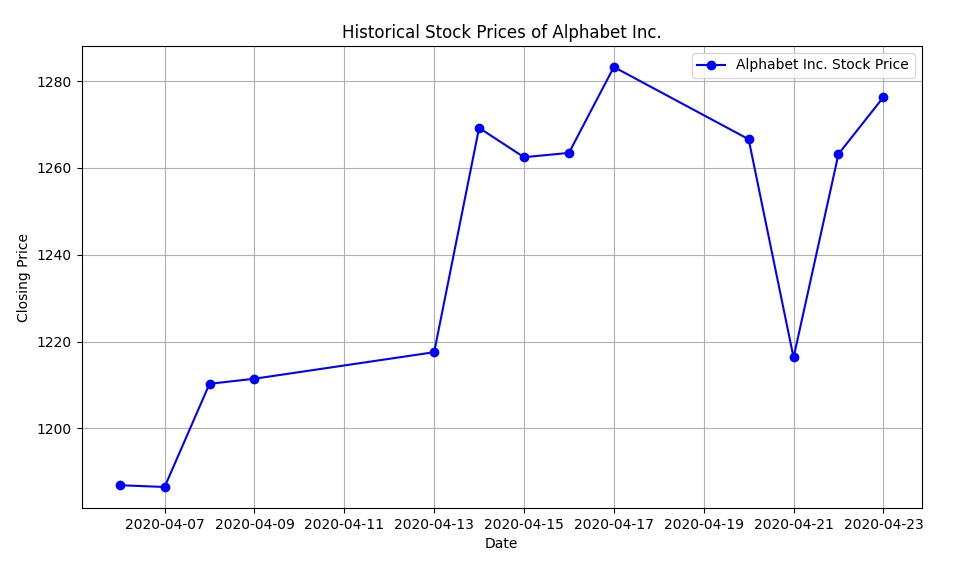
plt.ylabel('Closing Price')

plt.legend()

plt.grid(True)

plt.show()

OUTPUT:



1. Write a Pandas program to create a bar plot of the trading volume of Alphabet Inc. stock between two specific dates.

CODE:

import pandas as pd

import matplotlib.pyplot as plt

# Read the historical stock price data from a CSV file

df = pd.read\_csv(r"C:\Users\LENOVO\Downloads\WhatsApp Image 2023-11-02 at 09.39.30\_85640aa8.jpg.csv")

# Create a DataFrame from the provided data (you can replace this with your data)

# Convert the 'date' column to a datetime object

df['date'] = pd.to\_datetime(df['date'], format='%d-%m-%Y')

# Define the start and end dates

start\_date = '2020-04-06'

end\_date = '2020-04-23'

# Filter the data to include only the rows within the specified date range

filtered\_data = df[(df['date'] >= start\_date) & (df['date'] <= end\_date)]

# Create a bar plot of the trading volume

plt.figure(figsize=(12, 6))

plt.bar(filtered\_data['date'], filtered\_data['volume'], color='b', label='Trading Volume')

plt.title('Trading Volume of Alphabet Inc. Stock')

plt.xlabel('Date')

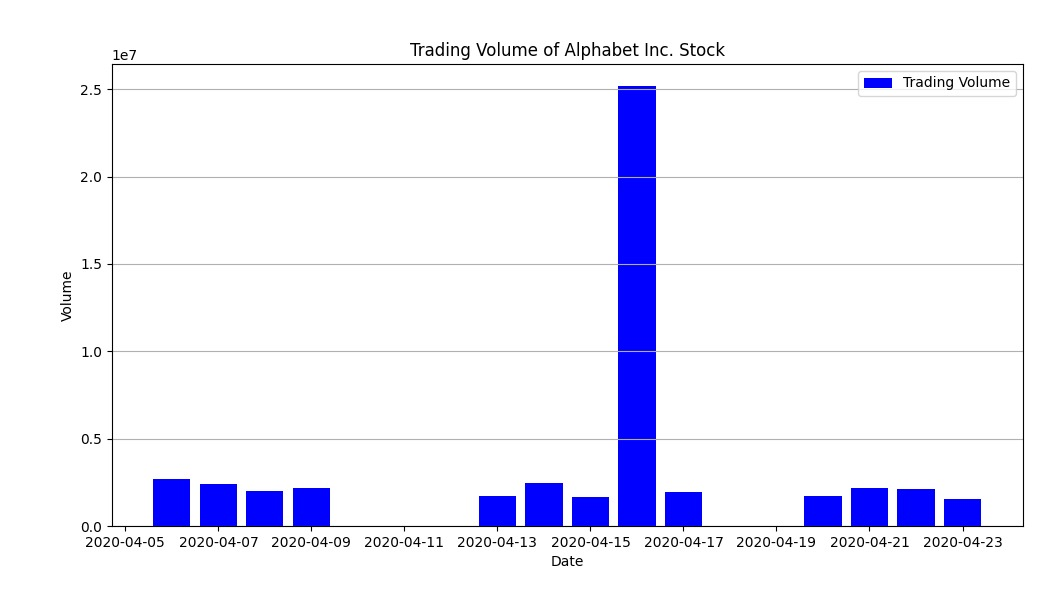
plt.ylabel('Volume')

plt.legend()

plt.grid(axis='y')

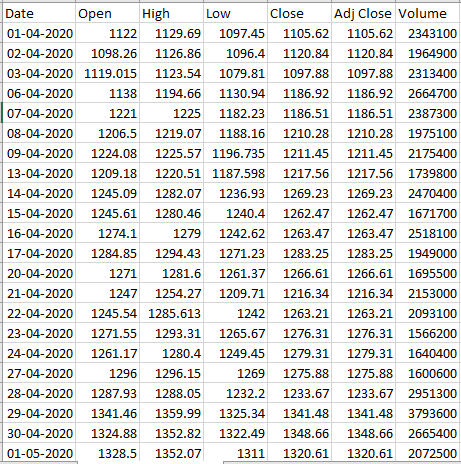
plt.show()

OUTPUT:



1. Write a Pandas program to create a scatter plot of the trading volume/stock prices of Alphabet Inc. stock between two specific dates.

**alphabet\_stock\_data:**



CODE:

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv(r"C:\Users\LENOVO\Downloads\WhatsApp Image 2023-11-02 at 09.39.30\_85640aa8.jpg.csv")

df['date'] = pd.to\_datetime(df['date'], format='%d-%m-%Y')

# Define the start and end dates

start\_date = '2020-04-06'

end\_date = '2020-04-23'

# Filter the data to include only the rows within the specified date range

filtered\_data = df[(df['date'] >= start\_date) & (df['date'] <= end\_date)]

# Create a scatter plot of trading volume vs. stock prices

plt.figure(figsize=(12, 6))

plt.scatter(filtered\_data['volume'], filtered\_data['close'], color='b', label='Volume vs. Stock Prices')

plt.title('Trading Volume vs. Stock Prices of Alphabet Inc. Stock')

plt.xlabel('Volume')

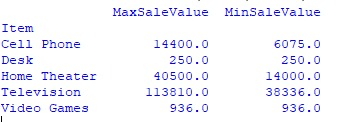
plt.ylabel('Closing Price')

plt.legend()

plt.grid(True)

plt.show()

OUTPUT:



1. Write a Pandas program to create a Pivot table and find the maximum and minimum sale value of the items.(refer sales\_data table)

CODE:

import pandas as pd

# Create a sample sales data DataFrame (replace this with your actual data)

data = {

'OrderDate': ['1-6-18', '1-23-18', '2-9-18', '2-26-18', '3-15-18', '4-1-18', '4-18-18', '5-5-18', '5-22-18', '6-8-18', '6-25-18', '7-12-18', '7-29-18', '8-15-18', '9-1-18', '9-18-18', '10-5-18', '10-22-18'],

'Region': ['East', 'Central', 'Central', 'Central', 'West', 'East', 'Central', 'Central', 'West', 'East', 'Central', 'East', 'East', 'East', 'Central', 'East', 'Central', 'East'],

'Manager': ['Martha', 'Hermann', 'Hermann', 'Timothy', 'Timothy', 'Martha', 'Martha', 'Hermann', 'Douglas', 'Martha', 'Hermann', 'Martha', 'Douglas', 'Martha', 'Douglas', 'Martha', 'Hermann', 'Martha'],

'SalesMan': ['Alexander', 'Shelli', 'Luis', 'David', 'Stephen', 'Alexander', 'Steven', 'Luis', 'Michael', 'Alexander', 'Sigal', 'Diana', 'Karen', 'Alexander', 'John', 'Alexander', 'Sigal', 'Alexander'],

'Item': ['Television', 'Home Theater', 'Television', 'Cell Phone', 'Television', 'Home Theater', 'Television', 'Television', 'Television', 'Home Theater', 'Television', 'Home Theater', 'Home Theater', 'Television', 'Desk', 'Video Games', 'Home Theater', 'Cell Phone'],

'Units': [95, 50, 36, 27, 56, 60, 75, 90, 32, 60, 90, 29, 81, 35, 2, 16, 28, 64],

'Unit\_price': [1198.00, 500.00, 1198.00, 225.00, 1198.00, 500.00, 1198.00, 1198.00, 1198.00, 500.00, 1198.00, 500.00, 500.00, 1198.00, 125.00, 58.50, 500.00, 225.00],

'Sale\_amt': [113810.00, 25000.00, 43128.00, 6075.00, 67088.00, 30000.00, 89850.00, 107820.00, 38336.00, 30000.00, 107820.00, 14500.00, 40500.00, 41930.00, 250.00, 936.00, 14000.00, 14400.00]

}

sales\_data = pd.DataFrame(data)

# Create a pivot table to find the maximum and minimum sale values for each item

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

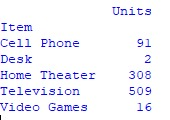
# Rename columns for clarity

pivot\_table.columns = ['MaxSaleValue', 'MinSaleValue']

# Print the pivot table

print(pivot\_table)

OUTPUT:



1. Write a Pandas program to create a Pivot table and find the item wise unit sold. .(refer sales\_data table)

CODE:

import pandas as pd

# Create a sample sales data DataFrame (replace this with your actual data)

data = {

'OrderDate': ['1-6-18', '1-23-18', '2-9-18', '2-26-18', '3-15-18', '4-1-18', '4-18-18', '5-5-18', '5-22-18', '6-8-18', '6-25-18', '7-12-18', '7-29-18', '8-15-18', '9-1-18', '9-18-18', '10-5-18', '10-22-18'],

'Region': ['East', 'Central', 'Central', 'Central', 'West', 'East', 'Central', 'Central', 'West', 'East', 'Central', 'East', 'East', 'East', 'Central', 'East', 'Central', 'East'],

'Manager': ['Martha', 'Hermann', 'Hermann', 'Timothy', 'Timothy', 'Martha', 'Martha', 'Hermann', 'Douglas', 'Martha', 'Hermann', 'Martha', 'Douglas', 'Martha', 'Douglas', 'Martha', 'Hermann', 'Martha'],

'SalesMan': ['Alexander', 'Shelli', 'Luis', 'David', 'Stephen', 'Alexander', 'Steven', 'Luis', 'Michael', 'Alexander', 'Sigal', 'Diana', 'Karen', 'Alexander', 'John', 'Alexander', 'Sigal', 'Alexander'],

'Item': ['Television', 'Home Theater', 'Television', 'Cell Phone', 'Television', 'Home Theater', 'Television', 'Television', 'Television', 'Home Theater', 'Television', 'Home Theater', 'Home Theater', 'Television', 'Desk', 'Video Games', 'Home Theater', 'Cell Phone'],

'Units': [95, 50, 36, 27, 56, 60, 75, 90, 32, 60, 90, 29, 81, 35, 2, 16, 28, 64],

'Unit\_price': [1198.00, 500.00, 1198.00, 225.00, 1198.00, 500.00, 1198.00, 1198.00, 1198.00, 500.00, 1198.00, 500.00, 500.00, 1198.00, 125.00, 58.50, 500.00, 225.00],

'Sale\_amt': [113810.00, 25000.00, 43128.00, 6075.00, 67088.00, 30000.00, 89850.00, 107820.00, 38336.00, 30000.00, 107820.00, 14500.00, 40500.00, 41930.00, 250.00, 936.00, 14000.00, 14400.00]

}

sales\_data = pd.DataFrame(data)

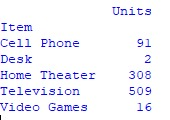
# Create a pivot table to find the unit sold for each item

pivot\_table = sales\_data.pivot\_table(index='Item', values='Units', aggfunc='sum')

# Print the pivot table

print(pivot\_table)

OUTPUT:



1. Write a Pandas program to create a Pivot table and find the total sale amount region wise, manager wise, sales man wise. .(refer sales\_data table)

**Sales\_data:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |

CODE:

import pandas as pd

# Create a sample sales data DataFrame (replace this with your actual data)

data = {

'OrderDate': ['1-6-18', '1-23-18', '2-9-18', '2-26-18', '3-15-18', '4-1-18', '4-18-18', '5-5-18', '5-22-18', '6-8-18', '6-25-18', '7-12-18', '7-29-18', '8-15-18', '9-1-18', '9-18-18', '10-5-18', '10-22-18'],

'Region': ['East', 'Central', 'Central', 'Central', 'West', 'East', 'Central', 'Central', 'West', 'East', 'Central', 'East', 'East', 'East', 'Central', 'East', 'Central', 'East'],

'Manager': ['Martha', 'Hermann', 'Hermann', 'Timothy', 'Timothy', 'Martha', 'Martha', 'Hermann', 'Douglas', 'Martha', 'Hermann', 'Martha', 'Douglas', 'Martha', 'Douglas', 'Martha', 'Hermann', 'Martha'],

'SalesMan': ['Alexander', 'Shelli', 'Luis', 'David', 'Stephen', 'Alexander', 'Steven', 'Luis', 'Michael', 'Alexander', 'Sigal', 'Diana', 'Karen', 'Alexander', 'John', 'Alexander', 'Sigal', 'Alexander'],

'Item': ['Television', 'Home Theater', 'Television', 'Cell Phone', 'Television', 'Home Theater', 'Television', 'Television', 'Television', 'Home Theater', 'Television', 'Home Theater', 'Home Theater', 'Television', 'Desk', 'Video Games', 'Home Theater', 'Cell Phone'],

'Units': [95, 50, 36, 27, 56, 60, 75, 90, 32, 60, 90, 29, 81, 35, 2, 16, 28, 64],

'Unit\_price': [1198.00, 500.00, 1198.00, 225.00, 1198.00, 500.00, 1198.00, 1198.00, 1198.00, 500.00, 1198.00, 500.00, 500.00, 1198.00, 125.00, 58.50, 500.00, 225.00],

'Sale\_amt': [113810.00, 25000.00, 43128.00, 6075.00, 67088.00, 30000.00, 89850.00, 107820.00, 38336.00, 30000.00, 107820.00, 14500.00, 40500.00, 41930.00, 250.00, 936.00, 14000.00, 14400.00]

}

sales\_data = pd.DataFrame(data)

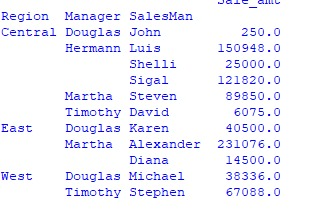
# Create a pivot table to find the unit sold for each item

pivot\_table = sales\_data.pivot\_table(index='Item', values='Units', aggfunc='sum')

# Print the pivot table

print(pivot\_table)

OUTPUT:



10.Create a dataframe of ten rows, four columns with random values. Write a Pandas program to highlight the negative numbers red and positive numbers black.

CODE:

import pandas as pd

import numpy as np

data = np.random.randn(10, 4)

df = pd.DataFrame(data, columns=['B', 'C', 'D','E'])

def highlight\_numbers(val):

color = 'red' if val < 0 else 'black'

return f'color: {color}'

styled\_df = df.style.applymap(highlight\_numbers)

styled\_df

OUTPUT:

