

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

The image shows a Python IDE with a file named "1.Write a Pandas program to select distinct department id from employees file.py". The code in the editor is as follows:

```
import pandas as pd
data = {
    'DEPARTMENT_ID': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260, 270],
    'DEPARTMENT_NAME': ['Administration', 'Marketing', 'Purchasing', 'Human Resources', 'Shipping', 'IT', 'Public Relations', 'Sales', 'Executive', 'Finance', 'Accounting', 'Treasury', 'Cost Management'],
    'MANAGER_ID': [200, 201, 114, 203, 121, 103, 204, 145, 100, 108, 205, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    'LOCATION_ID': [1700, 1800, 1700, 2400, 1500, 1400, 2700, 2500, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700, 1700]
}

df = pd.DataFrame(data)
distinct_department_ids = df['DEPARTMENT_ID'].unique()
print("Distinct Department IDs:")
print(distinct_department_ids)
```

The IDE Shell window shows the output of the program:

```
IDLE Shell 3.12.2
Python 3.12.2 (tags/v3.12.2:6abdd9d, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/Query Processing/1.Write a Pandas program to select distinct department id from employees file.py
Distinct Department IDs:
[ 10  20  30  40  50  60  70  80  90 100 110 120 130 140 150 160 170 180
 190 200 210 220 230 240 250 260 270]
```

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

The image shows a Python IDE window titled "2. Write a Pandas program to display the ID for those employees who did two or more jobs in the past.py - D:\Query Processing\2. Write a Pandas program to display the ID for those employees who did two or more jobs in the past.py (3.12.2)". The code in the editor is as follows:

```
import pandas as pd

data = {
    'EMPLOYEE_ID': [102, 101, 101, 201, 114, 122, 200, 176, 176, 200],
    'START_DATE': ['2001-01-13', '1997-09-21', '2001-10-28', '2004-02-17', '2006-03-24', '2007-01-01', '1995-09-17', '2006-03-24', '2007-01-01', '2002-07-01'],
    'END_DATE': ['2006-07-24', '2001-10-27', '2005-03-15', '2007-12-19', '2007-12-31', '2007-12-31', '2001-06-17', '2006-12-31', '2007-12-31', '2006-12-31'],
    'JOB_ID': ['IT_PROG', 'AC_ACCOUNT', 'AC_MGR', 'WM_REP', 'ST_CLERK', '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)
employee_job_counts = df[['EMPLOYEE_ID']].value_counts()
employees_with_multiple_jobs = employee_job_counts[employee_job_counts >= 2].index
print("Employees with two or more jobs:")
print(employees_with_multiple_jobs.tolist())
```

Below the code editor, a terminal window titled "IDLE Shell 3.12.2" is open, showing the execution of the program. The output is:

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> = RESTART: D:\Query Processing\2. Write a Pandas program to display the ID for those employees who did two or more jobs in the past.py
Employees with two or more jobs:
[101, 200, 176]
>>>
```

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

```

3. Write a Pandas program to display the details of jobs in descending sequence on job title.py - D:/Query Processing/3. Write a Pandas program to display the details of jobs in descending sequence on job title.py (3.12.2)
File Edit Format Run Options Window Help
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, 10500]
}

df = pd.DataFrame(data)
sorted_df = df.sort_values(by='JOB_TITLE', ascending=False)
print("Details of jobs in descending sequence on job title:")
print(sorted_df)

```

```

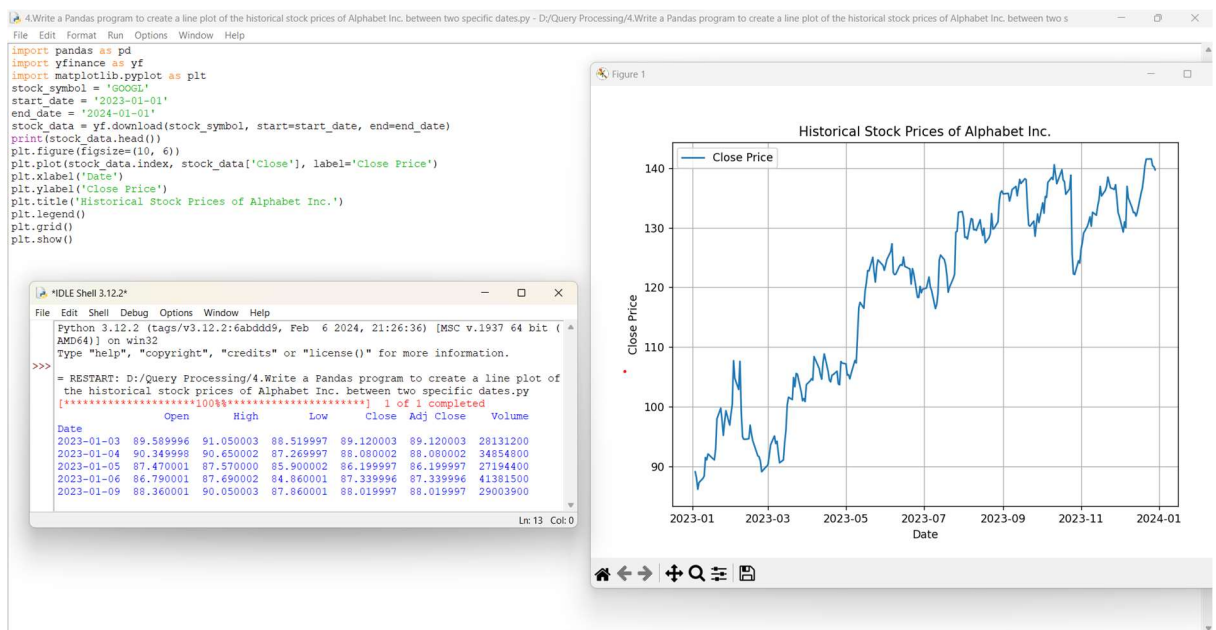
IDLE Shell 3.12.2
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> = RESTART: D:/Query Processing/3. Write a Pandas program to display the details of jobs in descending sequence on job title.py
Details of jobs in descending sequence on job title:

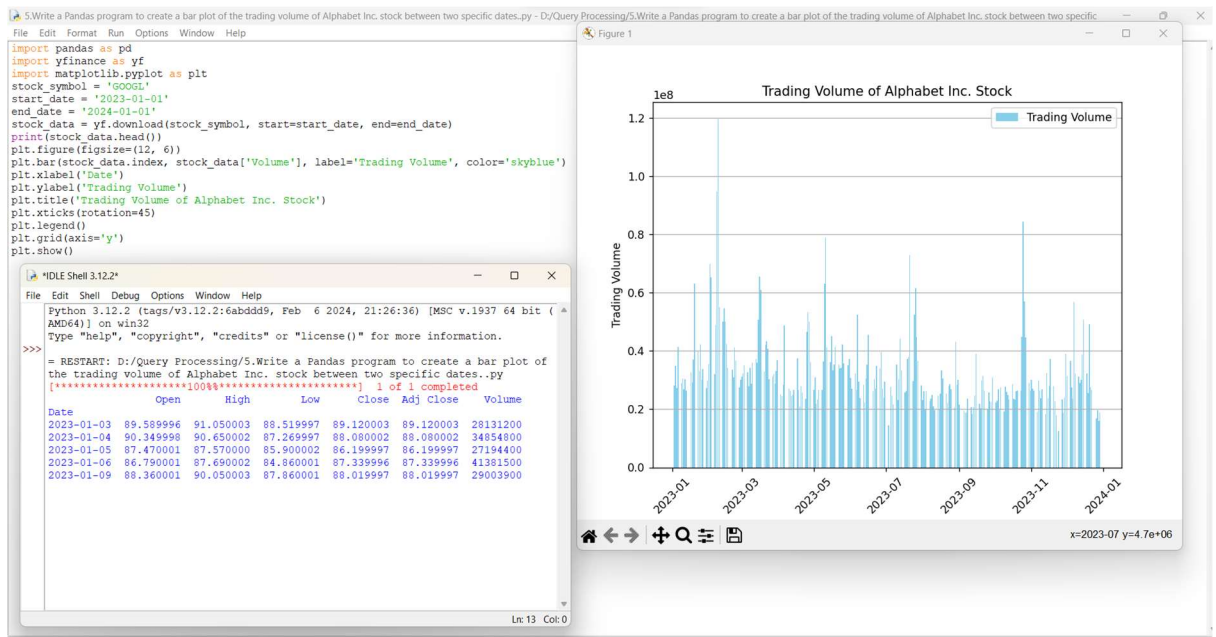
```

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY	
11	ST_MAN	Stock Manager	5500	8500
12	ST_CLERK	Stock Clerk	2008	5000
13	SH_CLERK	Shipping Clerk	2500	5500
8	SA_REP	Sales Representative	6000	12008
7	SA_MAN	Sales Manager	10000	20080
9	PU_MAN	Purchasing Manager	8000	15000
10	PU_CLERK	Purchasing Clerk	2500	5500
18	PR_REP	Public Relations Representative	4500	10500
6	AC_ACCOUNT	Public Accountant	4200	9000
14	IT_PROG	Programmer	4000	10000
0	AD_PRES	President	20080	40000
16	MK_REP	Marketing Representative	4000	9000
15	MK_MAN	Marketing Manager	9000	15000
17	HR_REP	Human Resources Representative	4000	9000
3	FI_MGR	Finance Manager	8200	16000
1	AD_VP	Administration Vice President	15000	30000
2	AD_ASST	Administration Assistant	3000	6000
5	AC_MGR	Accounting Manager	8200	16000
4	FI_ACCOUNT	Accountant	4200	9000

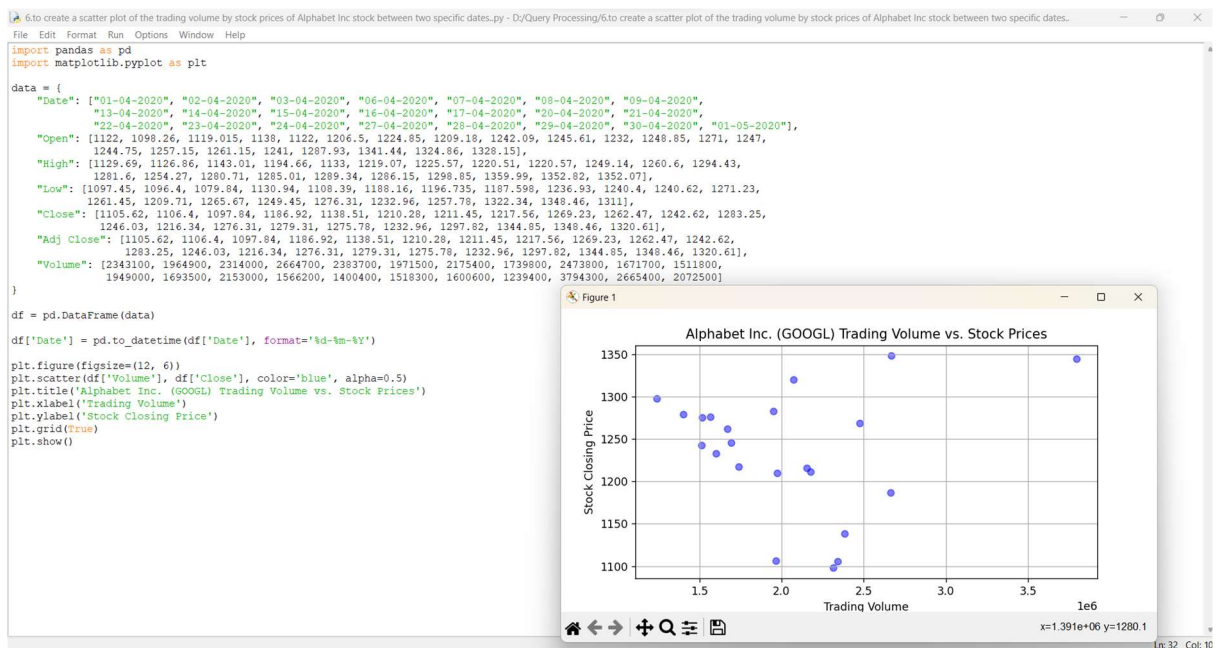
4. Write a Pandas program to create a line plot of the historical stock prices of Alphabet Inc. between two specific dates.



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



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



7. Write a Pandas program to create a Pivot table and find the maximum and minimum sale value of the items.

```
7. Write a Pandas program to create a Pivot table and find the maximum and minimum sale value of the items.(refer sales_data table).py - D:/Query Processing/7. Write a Pandas program to create a Pivot table and find the maximum and minimum sale value o...

import pandas as pd
data = {
    'Item': ['Item_A', 'Item_B', 'Item_A', 'Item_C', 'Item_B', 'Item_A', 'Item_C'],
    'Sale_Value': [200, 150, 300, 400, 350, 250, 450],
    'Date': ['2020-01-01', '2020-01-02', '2020-01-03', '2020-01-04',
            '2020-01-05', '2020-01-06', '2020-01-07']
}
df = pd.DataFrame(data)
pivot_table = df.pivot_table(values='Sale_Value', index='Item', aggfunc=['max', 'min'])
pivot_table.columns = ['Max_Sale_Value', 'Min_Sale_Value']
print(pivot_table)
```

```
IDLE Shell 3.12.2
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/Query Processing/6.to create a scatter plot of the trading volume by stock prices of Alphabet Inc stock between two specific dates..py
= RESTART: D:/Query Processing/6.to create a scatter plot of the trading volume by stock prices of Alphabet Inc stock between two specific dates..py
>>>
= RESTART: D:/Query Processing/7. Write a Pandas program to create a Pivot table and find the maximum and minimum sale value of the items.(refer sales_data table).py
Max_Sale_Value  Min_Sale_Value
Item
Item_A          300          200
Item_B          350          150
Item_C          450          400
>>>
```

8. Write a Pandas program to create a Pivot table and find the item wise unit sold.

```
8. Write a Pandas program to create a Pivot table and find the item wise unit sold.(refer sales_data table).py - D:/Query Processing/8. Write a Pandas program to create a Pivot table and find the item wise unit sold. (refer sales_data table).py (3.12.2)

import pandas as pd
sales_data = {
    'Date': ['2024-01-01', '2024-01-01', '2024-01-02', '2024-01-02', '2024-01-03'],
    'Item': ['Item_A', 'Item_B', 'Item_A', 'Item_B', 'Item_A'],
    'Units Sold': [10, 20, 15, 25, 12]
}
df = pd.DataFrame(sales_data)
pivot_table = df.pivot_table(index='Item', values='Units Sold', aggfunc='sum')
print(pivot_table)
```

```
IDLE Shell 3.12.2
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/Query Processing/8. Write a Pandas program to create a Pivot table and find the item wise unit sold. (refer sales_data table).py
Units Sold
Item
Item_A          37
Item_B          45
>>>
```

9. 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)

```

9. 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).py - D:/Query Processing/9. Write a Pandas program to create a Pivot table and find the total sale amount r
File Edit Format Run Options Window Help
import pandas as pd
sales_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', 'East'],
    'Manager': ['Martha', 'Hermann', 'Hermann', 'Timothy', 'Timothy', 'Martha', 'Martha', 'Hermann', 'Douglas',
               'Martha', 'Hermann', 'Martha', 'Douglas', 'Martha', 'Douglas', 'Hermann', 'Martha'],
    'SalesMan': ['Alexander', 'Shell', 'Luis', 'David', 'Stephen', 'Alexander', 'Steven', 'Luis', 'Michael',
                'Alexander', 'Sigal', 'Diana', 'Karen', 'Alexander', 'John', 'Sigal', 'Alexander', 'Alexander'],
    'Item': ['Television', 'Home Theater', 'Television', 'Cell Phone', 'Television', 'Home Theater', 'Television',
            'Desk', 'Home Theater', 'Cell Phone', 'Video Games'],
    'Units': [95, 50, 36, 27, 56, 60, 75, 90, 32, 60, 90, 29, 81, 35, 2, 28, 64, 16],
    'Unit_price': [1198.00, 500.00, 1198.00, 125.00, 1198.00, 500.00, 1198.00, 1198.00, 1198.00, 500.00, 1198.00,
                  500.00, 500.00, 1198.00, 125.00, 500.00, 225.00, 58.50],
    '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, 14000.00, 14400.00, 936.00]
}

df = pd.DataFrame(sales_data)
pivot_table = df.pivot_table(index=['Region', 'Manager', 'SalesMan'], values='Sale_amt', aggfunc='sum')
print(pivot_table)

```

```

Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
= RESTART: D:/Query Processing/9. 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).py

Region Manager SalesMan Sale_amt
Central Douglas John 250.0
Hermann Luis 150948.0
Shell 25000.0
Sigal 107820.0
Martha Alexander 14400.0
Steven 89850.0
East Timothy David 6075.0
Douglas Karen 40500.0
Hermann Sigal 14000.0
Martha Alexander 216676.0
Diana 14500.0
West Douglas Michael 38336.0
Timothy Stephen 67088.0
>>>

```

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.

```

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.
File Edit Format Run Options Window Help
import pandas as pd
import numpy as np
df = pd.DataFrame(np.random.randn(10, 4), columns=list('ABCD'))
def highlight_values(data):
    color = np.where(data < 0, 'color: red;', 'color: black;')
    return pd.DataFrame(color, index=data.index, columns=data.columns)
styled_df = df.style.apply(highlight_values, axis=0)
styled_df.to_html('styled_dataframe.html')
print("Styled DataFrame has been saved to 'styled_dataframe.html'.")

```

```

A B C D
0 2.114205 1.428821 -0.120132 1.515776
1 0.607021 -0.480288 1.062179 -1.039054
2 -1.259297 -0.092422 1.359802 -0.814773
3 -0.016681 -1.040425 -0.993316 1.204727
4 -0.449085 -1.186698 0.349828 1.244467
5 -0.182951 0.627811 -0.410642 -0.701708
6 -1.367676 1.371007 -0.840148 -0.864881
7 -0.400104 -0.873684 0.874501 -0.600658
8 -0.121208 0.661987 -0.273793 -0.588211
9 -0.267740 -0.406057 0.964380 1.617409

```