

```
import pandas as pd
```

```
# 1. Write a Pandas program to display all the records of REGIONS file.
```

```
regions = pd.read_csv('./csv_file/REGIONS.csv')
```

```
regions
```

	REGION_ID	REGION_NAME
0	1	Europe
1	2	Americas
2	3	Asia
3	4	Middle East and Africa

```
# 2. Write a Pandas program to display all the location id from locations file.
```

```
location = pd.read_csv('./csv_file/LOCATIONS.csv')
```

```
location['location_id']
```

```
0    1000
1    1100
2    1200
3    1300
4    1400
5    1500
6    1600
7    1700
8    1800
9    1900
10   2000
11   2100
12   2200
13   2300
14   2400
15   2500
16   2600
17   2700
18   2800
19   2900
20   3000
21   3100
22   3200
Name: location_id, dtype: int64
```

```
# 3. Write a Pandas program to extract first 7 records from employees file.
```

```
employee = pd.read_csv('./csv_file/EMPLOYEES.csv')
```

```
employee.head(7)
```

	employ_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
0	100	Steven	King	SKING	515.123.4567	2003-06-17	AD_PRES	24000	NaN	NaN	90.0
1	101	Neena	Kochhar	NKOCHHAR	515.123.4568	2005-09-21	AD_VP	17000	NaN	100.0	90.0
2	102	Lex	De Haan	LDEHAAN	515.123.4569	2001-01-13	AD_VP	17000	NaN	100.0	90.0
3	103	Alexander	Hunold	AHUNOLD	590.423.4567	2006-01-03	IT_PROG	9000	NaN	102.0	60.0
4	104	Bruce	Ernst	BERNST	590.423.4568	2007-05-21	IT_PROG	6000	NaN	103.0	60.0
5	105	David	Austin	DAUSTIN	590.423.4569	2005-06-25	IT_PROG	4800	NaN	103.0	60.0
6	106	Valli	Pataballa	VPATABAL	590.423.4560	2006-02-05	IT_PROG	4800	NaN	103.0	60.0

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

```
employee['department_id'].unique()
```

```
array([ 90.,  60., 100.,  30.,  50.,  80., nan,  10.,  20.,  40.,  70.,
        110.])
```

# 5. Write a Pandas program to display the first and last name, and department number for all employees whose last name is "McEwen".

```
filtered_record = employee[employee['last_name'].str.contains('McEwen')]
```

```
filtered_record[['first_name', 'last_name', 'department_id']]
```

	first_name	last_name	department_id
58	Allan	McEwen	80.0

# 6. Write a Pandas program to display the first, last name, salary and department number for those employees whose first name starts with the letter 'S'.

```
filtere_record = employee[employee['first_name'].str.lower().str.startswith('s')]
```

```
record = filtere_record[['first_name', 'last_name', 'salary', 'department_id']]
```

```
record
```

	first_name	last_name	salary	department_id
0	Steven	King	24000	90.0
16	Shelli	Baida	2900	30.0
17	Sigal	Tobias	2800	30.0
23	Shanta	Vollman	6500	50.0
28	Steven	Markle	2200	50.0
38	Stephen	Stiles	3200	50.0
61	Sarath	Sewall	7000	80.0
66	Sundar	Ande	6400	80.0
73	Sundita	Kumar	6100	80.0
92	Sarah	Bell	4000	50.0
94	Samuel	McCain	3200	50.0
103	Susan	Mavris	6500	40.0
105	Shelley	Higgins	12000	110.0

# 7. Write a Pandas program to display the first, last name, salary and department number for those employees whose first name does not contain the letter 'M'.

```
import pandas as pd
```

```
employee = pd.read_csv('./csv_file/EMPLOYEES.csv')
```

```
filtered_employee = employee[~employee['first_name'].str.lower().str.startswith('m')]
```

```
result_df = filtered_employee[['first_name', 'last_name', 'salary', 'department_id']]
```

```
result_df
```

	first_name	last_name	salary	department_id
0	Steven	King	24000	90.0
1	Neena	Kochhar	17000	90.0
2	Lex	De Haan	17000	90.0
3	Alexander	Hunold	9000	60.0
4	Bruce	Ernst	6000	60.0
...	...	...	...	...
102	Pat	Fay	6000	20.0
103	Susan	Mavris	6500	40.0
104	Hermann	Baer	10000	70.0
105	Shelley	Higgins	12000	110.0
106	William	Gietz	8300	110.0

101 rows × 4 columns

# 8. Write a Pandas program to display the first name, last name, salary and department number in ascending order by department number.

```
sorted_employee = employee.sort_values(by='department_id', ascending=True, ignore_index=True)
```

```
sorted_employee[['first_name', 'last_name', 'salary', 'department_id']]
```

	first_name	last_name	salary	department_id
0	Kimberely	Grant	7000	0
1	Jennifer	Whalen	4400	10
2	Michael	Hartstein	13000	20
3	Pat	Fay	6000	20
4	Karen	Colmenares	2500	30
...	...	...	...	...
102	John	Chen	8200	100
103	Daniel	Faviet	9000	100
104	Nancy	Greenberg	12000	100
105	Shelley	Higgins	12000	110
106	William	Gietz	8300	110

07 rows × 4 columns

# 9. Write a Pandas program to display the first name, last name, salary and department number in descending order by first name

```
sorted_by_name = employee.sort_values(by='first_name', ascending=True, ignore_index=True)
```

```
sorted_by_name[['first_name', 'last_name', 'salary', 'department_id']]
```

	first_name	last_name	salary	department_id
0	Adam	Fripp	8200	50.0
1	Alana	Walsh	3100	50.0
2	Alberto	Errazuriz	12000	80.0
3	Alexander	Hunold	9000	60.0
4	Alexander	Khoo	3100	30.0
...	...	...	...	...
102	Valli	Pataballa	4800	60.0
103	Vance	Jones	2800	50.0
104	William	Smith	7400	80.0
105	William	Gietz	8300	110.0
106	Winston	Taylor	3200	50.0

107 rows × 4 columns

# 10. Write a Pandas program to display the first name, last name, salary and manger id where manager ids are null.

```
filtered_record = employee[employee['manager_id'].isnull()]
filtered_record[['first_name','last_name','salary','manager_id']]
```

	first_name	last_name	salary	manager_id
0	Steven	King	24000	NaN

# 11. Write a Pandas program to display the first name, last name, salary and manger id where manager ids are not null.

```
filtered_record = employee[~employee['manager_id'].isnull()]
filtered_record[['first_name','last_name','salary','manager_id']]
```

	first_name	last_name	salary	manager_id
1	Neena	Kochhar	17000	100.0
2	Lex	De Haan	17000	100.0
3	Alexander	Hunold	9000	102.0
4	Bruce	Ernst	6000	103.0
5	David	Austin	4800	103.0
...	...	...	...	...
102	Pat	Fay	6000	201.0
103	Susan	Mavris	6500	101.0
104	Hermann	Baer	10000	101.0
105	Shelley	Higgins	12000	101.0
106	William	Gietz	8300	205.0

106 rows × 4 columns

# 12. Write a Pandas program to create and display a boolean series, where True for not null and False for null values or missing values in state\_province column of locations file.

```
boolean = location['state_province'].notnull()
```

Boolean

```
0      False
1      False
2       True
3      False
4       True
5       True
6       True
7       True
8       True
9       True
10     False
11      True
12      True
13     False
14     False
15      True
16      True
17      True
18      True
19      True
20      True
21      True
22      True
Name: state_province, dtype: bool
```

# 13. Write a Pandas program to create a boolean series selecting rows with one or more nulls from locations file.

```
boolean = location.isnull().any(axis=1)
```

Boolean

```
Out[1]: 0      True
1      True
2     False
3      True
4     False
5     False
6     False
7     False
8     False
9     False
10     True
11     False
12     False
13     True
14     True
15     False
16     False
17     False
18     False
19     False
20     False
21     False
22     False
dtype: bool
```

# 14. Write a Pandas program to count the NaN values of all the columns of locations file.

```
count_nan = location.isnull().sum()
```

count\_nan

```
location_id      0
street_address   0
postal_code      1
city             0
state_province   6
country_id       0
dtype: int64
```

# 15. Write a Pandas program to display the first name, last name, salary and department number for those employees whose first name ends with the letter 'm'.

```
filtered_record = employee[employee['first_name'].str.lower().str.endswith('m')]
```

```
filtered_record[['first_name','last_name','salary','department_id']]
```

	first_name	last_name	salary	department_id
21	Adam	Fripp	8200	50.0
22	Payam	Kaufling	7900	50.0
71	William	Smith	7400	80.0
106	William	Gietz	8300	110.0

# 16. Write a Pandas program to display the first name, last name, salary and department number for those employees whose first name ends with the letter 'd' or 'n' or 's' and also arrange the result in descending order by department id.

```
filtered = employee[employee['first_name'].str.lower().str.endswith(('s','d','n'))]
```

```
filtered.sort_values(by='department_id',ascending=False)[['first_name','last_name','salary','department_id']]
```

	first_name	last_name	salary	department_id
10	John	Chen	8200	100.0
13	Luis	Popp	6900	100.0
0	Steven	King	24000	90.0
45	John	Russell	14000	80.0
79	Charles	Johnson	6200	80.0
76	Jonathon	Taylor	8600	80.0
74	Ellen	Abel	11000	80.0
69	Harrison	Bloom	10000	80.0
65	David	Lee	6800	80.0
58	Allan	McEwen	9000	80.0
51	David	Bernstein	9500	80.0
46	Karen	Partners	13500	80.0
48	Gerald	Cambrault	11000	80.0
104	Hermann	Baer	10000	70.0
5	David	Austin	4800	60.0
39	John	Seo	2700	50.0
99	Douglas	Grant	2600	50.0
98	Donald	OConnell	2600	50.0

# 17. Write a Pandas program to display the first name, last name, salary and department number for employees who works either in department 70 or 90.

```
filtered_record = employee[employee['department_id'].isin([70,90])]
```

```
filtered_record[['first_name','last_name','salary','department_id']]
```

	first_name	last_name	salary	department_id
0	Steven	King	24000	90.0
1	Neena	Kochhar	17000	90.0
2	Lex	De Haan	17000	90.0
104	Hermann	Baer	10000	70.0

# 18. Write a Pandas program to display the first name, last name, salary and department number for those employees whose managers are hold the ID 120, 103 or 145.

```
filtered_record= employee[employee['manager_id'].isin([120,103,145])]
```

```
filtered_record[['first_name','last_name','salary','department_id']]
```

	first_name	last_name	salary	department_id
4	Bruce	Ernst	6000	60.0
5	David	Austin	4800	60.0
6	Valli	Pataballa	4800	60.0
7	Diana	Lorentz	4200	60.0
25	Julia	Nayer	3200	50.0
26	Irene	Mikkilineni	2700	50.0
27	James	Landry	2400	50.0
28	Steven	Markle	2200	50.0
50	Peter	Tucker	10000	80.0
51	David	Bernstein	9500	80.0
52	Peter	Hall	9000	80.0
53	Christopher	Olsen	8000	80.0
54	Nanette	Cambrault	7500	80.0
55	Oliver	Tuvault	7000	80.0
80	Winston	Taylor	3200	50.0
81	Jean	Fleaur	3100	50.0
82	Martha	Sullivan	2500	50.0
83	Girard	Geoni	2800	50.0

# 19. Write a Pandas program to display the first, last name, salary and department number for those employees who holds a letter n as a 3rd character in their first name.

```
filtered_record = employee[employee['first_name'].str.get(2) == 'n']
```

```
filtered_record[['first_name','last_name','salary','department_id']]
```

	first_name	last_name	salary	department_id
8	Nancy	Greenberg	12000	100.0
9	Daniel	Faviet	9000	100.0
14	Den	Raphaely	11000	30.0
37	Renske	Ladwig	3600	50.0
43	Randall	Matos	2600	50.0
54	Nanette	Cambrault	7500	80.0
56	Janette	King	10000	80.0
59	Lindsey	Smith	8000	80.0
63	Danielle	Greene	9500	80.0
66	Sundar	Ande	6400	80.0
73	Sundita	Kumar	6100	80.0
76	Jonathon	Taylor	8600	80.0
80	Winston	Taylor	3200	50.0
84	Nandita	Sarchand	4200	50.0
89	Jennifer	Dilly	3600	50.0
91	Randall	Perkins	2500	50.0
95	Vance	Jones	2800	50.0
98	Donald	OConnell	2600	50.0
00	Jennifer	Whalen	4400	10.0



# 20. Write a Pandas program to display the first name, job id, salary and department for those employees not working in the departments 50,30 and 80.

```
filtered_record = employee[~employee['department_id'].isin([50,30,80])]
```

```
filtered_record[['first_name','job_id','salary','department_id']]
```

```
:
```

	first_name	job_id	salary	department_id
0	Steven	AD_PRES	24000	90.0
1	Neena	AD_VP	17000	90.0
2	Lex	AD_VP	17000	90.0
3	Alexander	IT_PROG	9000	60.0
4	Bruce	IT_PROG	6000	60.0
5	David	IT_PROG	4800	60.0
6	Valli	IT_PROG	4800	60.0
7	Diana	IT_PROG	4200	60.0
8	Nancy	FI_MGR	12000	100.0
9	Daniel	FI_ACCOUNT	9000	100.0
10	John	FI_ACCOUNT	8200	100.0
11	Ismael	FI_ACCOUNT	7700	100.0

# 22. Write a Pandas program to calculate minimum, maximum and mean salary from employees file.

```
employee['salary'].agg(['max','min','mean'])
```

```
max      24000.000000
min      2100.000000
mean     6461.682243
Name: salary, dtype: float64
```

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

```
jobs = job.sort_values(by='job_title',ascending=False)
```

	job_id	job_title	min_salary	max_salary
11	ST_MAN	Stock Manager	5500	8500
12	ST_CLERK	Stock Clerk	2000	5000
13	SH_CLERK	Shipping Clerk	2500	5500
8	SA_REP	Sales Representative	6000	12000
7	SA_MAN	Sales Manager	10000	20000
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	20000	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

# 24. Write a Pandas program to display the first and last name and date of joining of the employees who is either Sales Representative or Sales Man

```
merge_df = pd.merge(employee,job,on='job_id')
```

```
filtered_df = merge_df[merge_df['job_title'].isin(['Sales Representative','Sales man'])]
```

```
filtered_df[['first_name','last_name','hire_date']]
```

	first_name	last_name	hire_date
50	Peter	Tucker	2005-01-30
51	David	Bernstein	2005-03-24
52	Peter	Hall	2005-08-20
53	Christopher	Olsen	2006-03-30
54	Nanette	Cambrault	2006-12-09
55	Oliver	Tuvault	2007-11-23
56	Janette	King	2004-01-30
57	Patrick	Sully	2004-03-04
58	Allan	McEwen	2004-08-01
59	Lindsey	Smith	2005-03-10
60	Louise	Doran	2005-12-15
61	Sarath	Sewall	2006-11-03
62	Clara	Vishney	2005-11-11