```
import pandas as pd
import numpy as np
```

Q1. **Write a Pandas program to display all the records of REGIONS file.**

```
data = pd.read_excel('hrdatabase.xlsx',sheet_name='Region')
data
```

Out[5]:

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

Q2. **Write a Pandas program to display all the location id from locations file.**

```
location = pd.read_excel('hrdatabase.xlsx', sheet_name='locations')
location[['LOCATION_ID']]
```

Out[28]:

	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

Q3. **Write a Pandas program to extract first 7 records from employees file.**

```
emp = pd.read_excel('hrdatabase.xlsx',sheet_name = 'employee', nrows = 7)
```

```
emp
```

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMEN
0	100	Steven	King	SKING	515.123.4567	2003-06-17	AD_PRES	24000	0	0	
1	101	Neena	Kochhar	NKOCHHAR	515.123.4568	2005-09-21	AD_VP	17000	0	100	
2	102	Lex	DeHaan	LDEHAAN	515.123.4569	2001-01-13	AD_VP	17000	0	100	
3	103	Alexander	Hunold	AHUNOLD	590.423.4567	2006-01-03	IT_PROG	9000	0	102	
4	104	Bruce	Ernst	BERNST	590.423.4568	2007-05-21	IT_PROG	6000	0	103	
5	105	David	Austin	DAUSTIN	590.423.4569	2005-06-25	IT_PROG	4800	0	103	
6	106	Valli	Pataballa	VPATABAL	590.423.4560	2006-02-05	IT_PROG	4800	0	103	

Q4. **Write a Pandas program to select distinct department id from employees file.**

```
emp['DEPARTMENT_ID'].unique()
```

```
Out[31]: array([90, 60], dtype=int64)
```

Q5. **Write a Pandas program to display the first and last name, and department number for all employees whose last name is "McEwen".**

```
emp = pd.read_excel('hrdatabase.xlsx', sheet_name = 'employee')
lname = emp[emp.LAST_NAME == 'McEwen']
lname[['FIRST_NAME','LAST_NAME',"DEPARTMENT_ID"]]
```

```
Out[3]:
```

	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
58	Allan	McEwen	80

Q6. **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'.**

```
emp = pd.read_excel('hrdatabase.xlsx', sheet_name = 'employee')
fname = emp[emp['FIRST_NAME'].str[0] == 'S']
fname[['FIRST_NAME','LAST_NAME','SALARY','DEPARTMENT_ID']]
```

Out[4]:

	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
0	Steven	King	24000	90
16	Shelli	Baida	2900	30
17	Sigal	Tobias	2800	30
23	Shanta	Vollman	6500	50
28	Steven	Markle	2200	50
38	Stephen	Stiles	3200	50
61	Sarath	Sewall	7000	80
66	Sundar	Ande	6400	80
73	Sundita	Kumar	6100	80
92	Sarah	Bell	4000	50
94	Samuel	McCain	3200	50
103	Susan	Mavris	6500	40
105	Shelley	Higgins	12008	110

Q7. **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'.**

```
lnames = emp[~(emp['FIRST_NAME'].str.contains(pat ='m',case = False))]
lnames[['FIRST_NAME','LAST_NAME','SALARY','DEPARTMENT_ID']]
```

Out[101]:

	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
0	Steven	King	24000	90
1	Neena	Kochhar	17000	90
2	Lex	DeHaan	17000	90
3	Alexander	Hunold	9000	60
4	Bruce	Ernst	6000	60
5	David	Austin	4800	60
6	Valli	Pataballa	4800	60
7	Diana	Lorentz	4200	60
8	Nancy	Greenberg	12008	100
9	Daniel	Faviet	9000	100
10	John	Chen	8200	100

13	Luis	Popp	6900	100
14	Den	Raphaely	11000	30
15	Alexander	Khoo	3100	30
16	Shelli	Baida	2900	30
17	Sigal	Tobias	2800	30
18	Guy	Himuro	2600	30
19	Karen	Colmenares	2500	30
23	Shanta	Vollman	6500	50
24	Kevin	Mourgos	5800	50
25	Julia	Nayer	3200	50
26	Irene	Mikkilineni	2700	50
28	Steven	Markle	2200	50
29	Laura	Bissot	3300	50
32	TJ	Olson	2100	50
33	Jason	Mallin	3300	50
35	Ki	Gee	2400	50
36	Hazel	Philtanker	2200	50
37	Renske	Ladwig	3600	50
38	Stephen	Stiles	3200	50
39	John	Seo	2700	50
40	Joshua	Patel	2500	50
41	Trenna	Rajs	3500	50
42	Curtis	Davies	3100	50
43	Randall	Matos	2600	50
44	Peter	Vargas	2500	50
45	John	Russell	14000	80
46	Karen	Partners	13500	80
47	Alberto	Errazuriz	12000	80
48	Gerald	Cambrault	11000	80
49	Eleni	Zlotkey	10500	80
50	Peter	Tucker	10000	80
51	David	Bernstein	9500	80
52	Peter	Hall	9000	80
53	Christopher	Olsen	8000	80
54	Nanette	Cambrault	7500	80
55	Oliver	Tuvault	7000	80
56	Janette	King	10000	80
57	Patrick	Sully	9500	80
58	Allan	McEwen	9000	80
59	Lindsey	Smith	8000	80
60	Louise	Doran	7500	80
61	Sarath	Sewall	7000	80
62	Clara	Vishney	10500	80
63	Danielle	Greene	9500	80
65	David	Lee	6800	80
66	Sundar	Ande	6400	80
68	Lisa	Ozer	11500	80
69	Harrison	Bloom	10000	80
73	Sundita	Kumar	6100	80
74	Ellen	Abel	11000	80
75	Alyssa	Hutton	8800	80
76	Jonathon	Taylor	8600	80
77	Jack	Livingston	8400	80
79	Charles	Johnson	6200	80
80	Winston	Taylor	3200	50
81	Jean	Fleaur	3100	50
83	Girard	Geoni	2800	50
84	Nandita	Sarchand	4200	50
85	Alexis	Bull	4100	50
86	Julia	Dellinger	3400	50
87	Anthony	Cabrio	3000	50
88	Kelly	Chung	3800	50
89	Jennifer	Dilly	3600	50
91	Randall	Perkins	2500	50
92	Sarah	Bell	4000	50
93	Britney	Everett	3900	50
95	Vance	Jones	2800	50
96	Alana	Walsh	3100	50
97	Kevin	Feeney	3000	50
98	Donald	OConnell	2600	50
99	Douglas	Grant	2600	50
100	Jennifer	Whalen	4400	10
102	Pat	Fay	6000	20
103	Susan	Mavris	6500	40
105	Shelley	Higgins	12008	110

Q8. **Write a Pandas program to display the first name, last name, salary and department number in ascending order by department number.**

```
Asc = emp.sort_values('FIRST_NAME')
Asc[['FIRST_NAME','LAST_NAME','SALARY','DEPARTMENT_ID']]
```

Out[104]:

	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
21	Adam	Fripp	8200	50
96	Alana	Walsh	3100	50
47	Alberto	Errazuriz	12000	80
3	Alexander	Hunold	9000	60
15	Alexander	Khoo	3100	30
85	Alexis	Bull	4100	50
58	Allan	McEwen	9000	80
75	Alyssa	Hutton	8800	80
67	Amit	Banda	6200	80
87	Anthony	Cabrio	3000	50
93	Britney	Everett	3900	50

Q9. **Write a Pandas program to display the first name, last name, salary and department number in descending order by first name.**

```
Desc = emp.sort_values('FIRST_NAME', ascending = False)
Desc[['FIRST_NAME','LAST_NAME','SALARY','DEPARTMENT_ID']]
```

Out[105]:

	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
80	Winston	Taylor	3200	50
106	William	Gietz	8300	110
71	William	Smith	7400	80
95	Vance	Jones	2800	50
6	Valli	Pataballa	4800	60
41	Trenna	Rajs	3500	50
90	Timothy	Gates	2900	50
70	Tayler	Fox	9600	80
32	TJ	Olson	2100	50
103	Susan	Mavris	6500	40
73	Sundita	Kumar	6100	80

Q10. **Write a Pandas program to display the first name, last name, salary and manger id where manager ids are null.**

```
Mng_id = emp[emp.MANAGER_ID.isnull()]
Mng_id[['FIRST_NAME','LAST_NAME','SALARY','DEPARTMENT_ID']]
```

```
Out[115]:
```

FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID

Q11. **Write a Pandas program to display the first name, last name, salary and manger id where manager ids are not null.**

```
Mnag_id = emp[emp.MANAGER_ID.notnull()]
Mnag_id[['FIRST_NAME','LAST_NAME','SALARY','DEPARTMENT_ID']]
```

Out[116]:

	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
0	Steven	King	24000	90
1	Neena	Kochhar	17000	90
2	Lex	DeHaan	17000	90
3	Alexander	Hunold	9000	60
4	Bruce	Ernst	6000	60
5	David	Austin	4800	60
6	Valli	Pataballa	4800	60
7	Diana	Lorentz	4200	60
8	Nancy	Greenberg	12008	100
9	Daniel	Faviet	9000	100
10	John	Chen	8200	100

Q12. **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.**

```
locat = pd.read_excel('hrdatabase.xlsx',sheet_name = 'locations')
locat['STATE_PROVINCE'].notnull()
```

```
Out[118]: 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
```

Q13. **Write a Pandas program to create a boolean series selecting rows with one or more nulls from locations file.**

```
locat = pd.read_excel('hrdatabase.xlsx',sheet_name = 'locations')
locat[locat.isna().any(axis=1)]
```

Out[37]:

	LOCATION_ID	STREET_ADDRESS	POSTAL_CODE	CITY	STATE_PROVINCE	COUNTRY_ID
0	1000	1297 Via Cola di Rie	989	Roma	NaN	IT
1	1100	93091 Calle della Testa	10934	Venice	NaN	IT
3	1300	9450 Kamiya-cho	6823	Hiroshima	NaN	JP
10	2000	40-5-12 Laogianggen	190518	Beijing	NaN	CN
13	2300	198 Clementi North	540198	Singapore	NaN	SG
14	2400	8204 Arthur St	NaN	London	NaN	UK

Q14. **Write a Pandas program to count the NaN values of all the columns of locations file.**

locat.isnull().sum()

```
LOCATION_ID      0
STREET_ADDRESS   0
POSTAL_CODE      1
CITY             0
STATE_PROVINCE   6
COUNTRY_ID       0
dtype: int64
```

Q15. **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'.**

```
emp = pd.read_excel('hrdatabase.xlsx',sheet_name = 'employee')
res = emp[emp['LAST_NAME'].str[-1]=='m']
res[['FIRST_NAME','LAST_NAME','SALARY','DEPARTMENT_ID']]
```

Out[61]:

	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
69	Harrison	Bloom	10000	80

Q16. **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.**

```
emp = pd.read_excel('hrdatabase.xlsx',sheet_name = 'employee')
res = [emp[emp['FIRST_NAME'].str[-1]=='d'], emp[emp['FIRST_NAME'].str[-1]=='n'],
emp[emp['FIRST_NAME'].str[-1]=='s']]
new = pd.concat(res)
final = new[['FIRST_NAME','LAST_NAME','SALARY','DEPARTMENT_ID']]
final.sort_values(by=['DEPARTMENT_ID'], ascending= False)
```

Out[69]:

	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
13	Luis	Popp	6900	100
10	John	Chen	8200	100
0	Steven	King	24000	90
46	Karen	Partners	13500	80
45	John	Russell	14000	80
51	David	Bernstein	9500	80
65	David	Lee	6800	80
79	Charles	Johnson	6200	80
76	Jonathon	Taylor	8800	80
74	Ellen	Abel	11000	80
69	Harrison	Bloom	10000	80
58	Allan	McEwen	9000	80
48	Gerald	Cambrault	11000	80
104	Hermann	Baer	10000	70
5	David	Austin	4800	60
39	John	Seo	2700	50
38	Stephen	Stiles	3200	50
85	Alexis	Bull	4100	50
83	Girard	Geoni	2800	50
42	Curtis	Davies	3100	50
31	James	Marlow	2500	50
27	James	Landry	2400	50
98	Donald	OConnell	2600	50
97	Kevin	Feeney	3000	50
81	Jean	Fleaur	3100	50
80	Winston	Taylor	3200	50
24	Kevin	Mourgos	5800	50
28	Steven	Markle	2200	50
33	Jason	Mallin	3300	50
103	Susan	Mavris	6500	40
14	Den	Raphaely	11000	30
19	Karen	Colmenares	2500	30

Q17. **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.**

```
dept = emp[emp['DEPARTMENT_ID'].isin([70,90])]
dept[['FIRST_NAME','LAST_NAME','SALARY','DEPARTMENT_ID']]
```

Out[121]:

	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
0	Steven	King	24000	90
1	Neena	Kochhar	17000	90
2	Lex	DeHaan	17000	90
104	Hermann	Baer	10000	70

Q18. **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.**

```
mng = emp[emp['MANAGER_ID'].isin([120,103,145])]
mng[['FIRST_NAME','LAST_NAME','SALARY','MANAGER_ID']]
```

Out[123]:

	FIRST_NAME	LAST_NAME	SALARY	MANAGER_ID
4	Bruce	Ernst	6000	103
5	David	Austin	4800	103
6	Valli	Pataballa	4800	103
7	Diana	Lorentz	4200	103
25	Julia	Nayer	3200	120
26	Irene	Mikkilineni	2700	120
27	James	Landry	2400	120
28	Steven	Markle	2200	120
50	Peter	Tucker	10000	145
51	David	Bernstein	9500	145
52	Peter	Hall	9000	145
53	Christopher	Olsen	8000	145
54	Nanette	Cambrault	7500	145
55	Oliver	Tuvault	7000	145
80	Winston	Taylor	3200	120
81	Jean	Fleaur	3100	120
82	Martha	Sullivan	2500	120
83	Girard	Geoni	2800	120

Q19. **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.**

```
char = emp[emp['FIRST_NAME'].str[2]=='n']
char
```

Out[125]:

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPART
8	108	Nancy	Greenberg	NGREENBE	515.124.4569	2002-08-17	FI_MGR	12008	0.00	101	
9	109	Daniel	Faviet	DFAVIET	515.124.4169	2002-08-16	FI_ACCOUNT	9000	0.00	108	
14	114	Den	Raphaely	DRAPHEAL	515.127.4561	2002-12-07	PU_MAN	11000	0.00	100	
37	137	Renske	Ladwig	RLADWIG	650.121.1234	2003-07-14	ST_CLERK	3600	0.00	123	
43	143	Randall	Matos	RMATOS	650.121.2874	2006-03-15	ST_CLERK	2600	0.00	124	
54	154	Nanette	Cambrault	NCAMBRAU	011.44.1344.987668	2006-12-09	SA_REP	7500	0.20	145	
56	156	Janette	King	JKING	011.44.1345.429268	2004-01-30	SA_REP	10000	0.35	146	
59	159	Lindsey	Smith	LSMITH	011.44.1345.729268	2005-03-10	SA_REP	8000	0.30	146	
63	163	Danielle	Greene	DGREENE	011.44.1346.229268	2007-03-19	SA_REP	9500	0.15	147	
66	166	Sundar	Ande	SANDE	011.44.1346.629268	2008-03-24	SA_REP	6400	0.10	147	
73	173	Sundita	Kumar	SKUMAR	011.44.1343.329268	2008-04-21	SA_REP	6100	0.10	148	
76	176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	2006-03-24	SA_REP	8600	0.20	149	
80	180	Winston	Taylor	WTAYLOR	650.507.9876	2006-01-24	SH_CLERK	3200	0.00	120	
84	184	Nandita	Sarchand	NSARCHAN	650.509.1876	2004-01-27	SH_CLERK	4200	0.00	121	
89	189	Jennifer	Dilly	JDILLY	650.505.2876	2005-08-13	SH_CLERK	3600	0.00	122	
91	191	Randall	Perkins	RPERKINS	650.505.4876	2007-12-19	SH_CLERK	2500	0.00	122	
95	195	Vance	Jones	VJONES	650.501.4876	2007-03-17	SH_CLERK	2800	0.00	123	
98	198	Donald	OConnell	DOCONNEL	650.507.9833	2007-06-21	SH_CLERK	2600	0.00	124	
100	200	Jennifer	Whalen	JWHALEN	515.123.4444	2003-09-17	AD_ASST	4400	0.00	101	

Q20. **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.**

```
dept = emp[~emp['DEPARTMENT_ID'].isin([30,50,80])]
dept[['FIRST_NAME','JOB_ID','SALARY','DEPARTMENT_ID']]
```

Out[128]:

	FIRST_NAME	JOB_ID	SALARY	DEPARTMENT_ID
0	Steven	AD_PRES	24000	90
1	Neena	AD_VP	17000	90
2	Lex	AD_VP	17000	90
3	Alexander	IT_PROG	9000	60
4	Bruce	IT_PROG	6000	60
5	David	IT_PROG	4800	60
6	Valli	IT_PROG	4800	60
7	Diana	IT_PROG	4200	60
8	Nancy	FI_MGR	12008	100
9	Daniel	FI_ACCOUNT	9000	100
10	John	FI_ACCOUNT	8200	100
11	Ismael	FI_ACCOUNT	7700	100
12	JoseManuel	FI_ACCOUNT	7800	100
13	Luis	FI_ACCOUNT	6900	100
78	Kimberely	SA_REP	7000	0
100	Jennifer	AD_ASST	4400	10
101	Michael	MK_MAN	13000	20
102	Pat	MK_REP	6000	20
103	Susan	HR_REP	6500	40
104	Hermann	PR_REP	10000	70
105	Shelley	AC_MGR	12008	110
106	William	AC_ACCOUNT	8300	110

Q21. **Write a Pandas program to display the ID for those employees who did two or more jobs in the past.**

```
job_his = pd.read_excel('hrdatabase.xlsx', sheet_name = 'job history')
a = job_his['EMPLOYEE_ID'].unique()
b = []
for i in a:
    c = 0
    for j in job_his.EMPLOYEE_ID:
        if i==j:
            c+=1
    if c>=2:
        b.append(i)
df = pd.DataFrame({'Employee_ID':b})
df
```

Out[58]:

	Employee_ID
0	101
1	200
2	176

Q22. **Write a Pandas program to calculate minimum, maximum and mean salary from employees file.**

```
salary = pd.read_excel('hrdatabase.xlsx', sheet_name = 'employee')
salary.agg({"SALARY":[np.mean,np.min,np.max]})
```

Out[64]:

	SALARY
mean	6461.831776
amin	2100.000000
amax	24000.000000

Q23. **Write a Pandas program to display the details of jobs in descending sequence on job title.**

```
Jobs = pd.read_excel('hrdatabase.xlsx',sheet_name='jobs')
Jobs.sort_values(by=['JOB_TITLE'], ascending=False)
```

Out[5]:

	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

Q24. **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.**

```
emp = pd.read_excel('hrdatabase.xlsx', sheet_name = 'employee')
lit = [emp.groupby('JOB_ID').get_group('SA_MAN'),emp.groupby('JOB_ID').get_group('SA_REP')]
new_data = pd.concat(lit)
new_data[['FIRST_NAME','LAST_NAME','HIRE_DATE']]
```

Out[24]:

	FIRST_NAME	LAST_NAME	HIRE_DATE
45	John	Russell	2004-10-01
46	Karen	Partners	2005-01-05
47	Alberto	Errazuriz	2005-03-10
48	Gerald	Cambrault	2007-10-15
49	Eleni	Zlotkey	2008-01-29
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
63	Danielle	Greene	2007-03-19
64	Mattea	Marvins	2008-01-24
65	David	Lee	2008-02-23
66	Sundar	Ande	2008-03-24
67	Amit	Banda	2008-04-21
68	Lisa	Ozer	2005-03-11
69	Harrison	Bloom	2006-03-23
70	Tayler	Fox	2006-01-24
71	William	Smith	2007-02-23
72	Elizabeth	Bates	2007-03-24
73	Sundita	Kumar	2008-04-21
74	Ellen	Abel	2004-05-11
75	Alyssa	Hutton	2005-03-19
76	Jonathon	Taylor	2006-03-24
77	Jack	Livingston	2006-04-23
78	Kimberely	Grant	2007-05-24
79	Charles	Johnson	2008-01-04