pandas读取csv解决乱码问题

In [46]:
```
import pandas as pd

movies = pd.read_csv("Data/豆瓣电影排名前250位.csv")

movies
```

Out[46]:

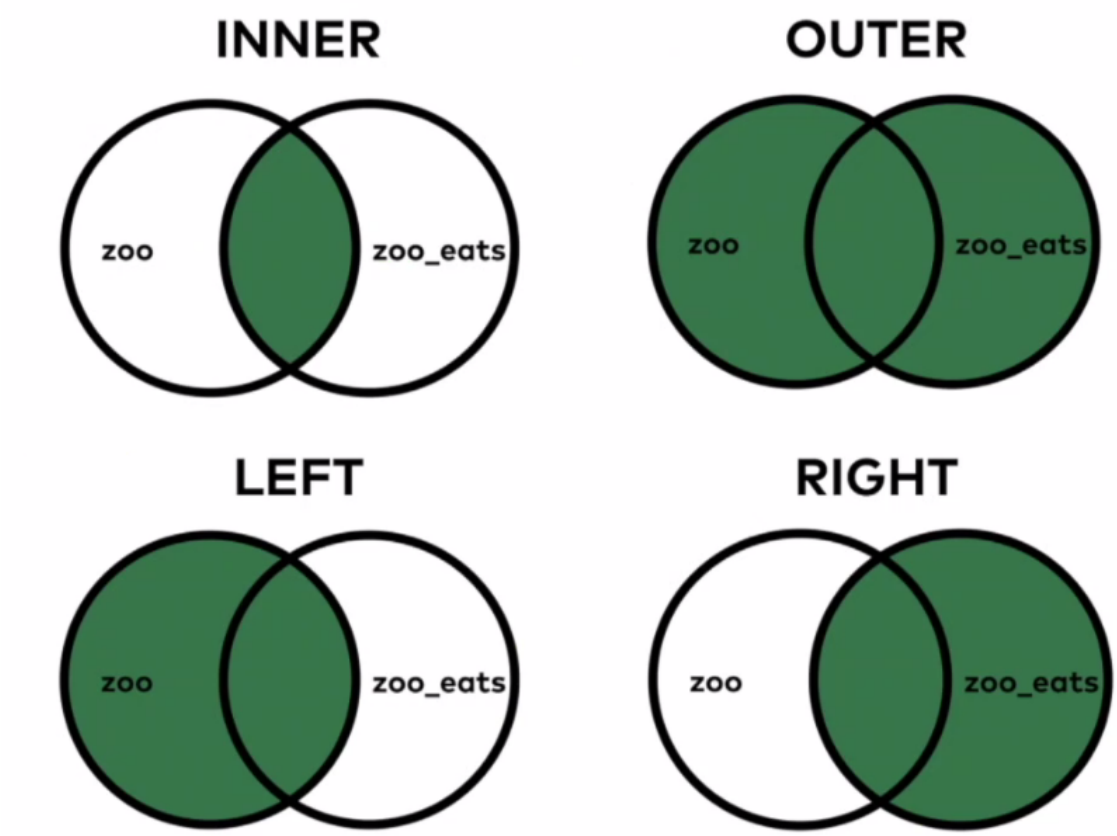
	排名	电影名称	评分
0	第1名	肖申克的救赎	9.7分
1	第2名	霸王别姬	9.6分
2	第3名	阿甘正传	9.5分
3	第4名	这个杀手不太冷	9.4分
4	第5名	美丽人生	9.5分
...	...	...	...
245	第246名	四个春天	8.9分
246	第247名	网络谜踪	8.6分
247	第248名	发条橙	8.5分
248	第249名	E.T. 外星人	8.6分
249	第250名	黑客帝国2：重装上阵	8.6分

250 rows × 3 columns

In [47]:
```
movies.to_csv("Data/top250.csv",index=False)
```

In [48]:
```
#解决乱码问题
movies.to_csv("Data/top250-new.csv",index=False,encoding="utf-8-sig")   #utf-8
#另一种是不包含BOM头的（类unix系统使用），-sig是包含bom头的设置。
```

merge合并



In [16]:
```
#merge合并是按照索引来进行合并，也就是按照列来合并。
```

In [49]:
```
import pandas as pd

zoo = pd.read_csv('Data/zoo.csv')
zoo
```

Out[49]:

	animal	uniq_id	water_need
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220

	animal	uniq_id	water_need
13	zebra	1014	100
14	zebra	1015	80
15	lion	1016	420
16	lion	1017	600
17	lion	1018	500
18	lion	1019	390
19	kangaroo	1020	410
20	kangaroo	1021	430
21	kangaroo	1022	410

In [50]:
```
zoo_eat = pd.read_csv('Data/zoo_eat.csv')
zoo_eat
```

Out[50]:

	animal	food
0	elephant	vegetables
1	tiger	meat
2	kangaroo	vegetables
3	zebra	vegetables
4	giraffe	vegetables

In [51]:
```
#交集
pd.merge(zoo,zoo_eat)
```

Out[51]:

	animal	uniq_id	water_need	food
0	elephant	1001	500	vegetables
1	elephant	1002	600	vegetables
2	elephant	1003	550	vegetables
3	tiger	1004	300	meat
4	tiger	1005	320	meat
5	tiger	1006	330	meat
6	tiger	1007	290	meat
7	tiger	1008	310	meat
8	zebra	1009	200	vegetables
9	zebra	1010	220	vegetables
10	zebra	1011	240	vegetables
11	zebra	1012	230	vegetables
12	zebra	1013	220	vegetables
13	zebra	1014	100	vegetables

	animal	uniq_id	water_need	food
14	zebra	1015	80	vegetables
15	kangaroo	1020	410	vegetables
16	kangaroo	1021	430	vegetables
17	kangaroo	1022	410	vegetables

In [52]:
```
#等价于
pd.merge(zoo,zoo_eat,on='animal',how='inner')
```

Out[52]:

	animal	uniq_id	water_need	food
0	elephant	1001	500	vegetables
1	elephant	1002	600	vegetables
2	elephant	1003	550	vegetables
3	tiger	1004	300	meat
4	tiger	1005	320	meat
5	tiger	1006	330	meat
6	tiger	1007	290	meat
7	tiger	1008	310	meat
8	zebra	1009	200	vegetables
9	zebra	1010	220	vegetables
10	zebra	1011	240	vegetables
11	zebra	1012	230	vegetables
12	zebra	1013	220	vegetables
13	zebra	1014	100	vegetables
14	zebra	1015	80	vegetables
15	kangaroo	1020	410	vegetables
16	kangaroo	1021	430	vegetables
17	kangaroo	1022	410	vegetables

In [53]:
```
#如果两个表格没有一样的列名
zoo_eat = pd.read_csv('Data/zoo_eat.csv')
zoo_eat.columns=['animals','food']
zoo_eat
```

Out[53]:

	animals	food
0	elephant	vegetables
1	tiger	meat
2	kangaroo	vegetables
3	zebra	vegetables

	animals	food
4	giraffe	vegetables

In [54]:
```
pd.merge(zoo,zoo_eat,on='animal',how='inner')  #合并时会出错
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-54-57e0f79f7732> in <module>
----> 1 pd.merge(zoo,zoo_eat,on='animal',how='inner')  #合并时会出错

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\reshape\merge.py in me
rge(left, right, how, on, left_on, right_on, left_index, right_index, sort, s
uffixes, copy, indicator, validate)
     72     validate=None,
     73 ) -> "DataFrame":
---> 74     op = _MergeOperation(
     75         left,
     76         right,

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\reshape\merge.py in __
init__(self, left, right, how, on, left_on, right_on, axis, left_index, right
_index, sort, suffixes, copy, indicator, validate)
    666             self.right_join_keys,
    667             self.join_names,
--> 668         ) = self._get_merge_keys()
    669
    670         # validate the merge keys dtypes. We may need to coerce

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\reshape\merge.py in _g
et_merge_keys(self)
   1031                     if not is_rkey(rk):
   1032                         if rk is not None:
-> 1033                             right_keys.append(right._get_label_or_lev
el_values(rk))
   1034                         else:
   1035                             # work-around for merge_asof(right_index=
True)

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in _get_lab
el_or_level_values(self, key, axis)
   1682             values = self.axes[axis].get_level_values(key)._values
   1683         else:
-> 1684             raise KeyError(key)
   1685
   1686         # Check for duplicates

KeyError: 'animal'
```

In [56]:
```
pd.merge(zoo,zoo_eat,left_on='animal',right_on='animals',how='inner') #需要指定
```

Out[56]:

	animal	uniq_id	water_need	animals	food
0	elephant	1001	500	elephant	vegetables
1	elephant	1002	600	elephant	vegetables
2	elephant	1003	550	elephant	vegetables
3	tiger	1004	300	tiger	meat
4	tiger	1005	320	tiger	meat
5	tiger	1006	330	tiger	meat

	animal	uniq_id	water_need	animals	food
6	tiger	1007	290	tiger	meat
7	tiger	1008	310	tiger	meat
8	zebra	1009	200	zebra	vegetables
9	zebra	1010	220	zebra	vegetables
10	zebra	1011	240	zebra	vegetables
11	zebra	1012	230	zebra	vegetables
12	zebra	1013	220	zebra	vegetables
13	zebra	1014	100	zebra	vegetables
14	zebra	1015	80	zebra	vegetables
15	kangaroo	1020	410	kangaroo	vegetables
16	kangaroo	1021	430	kangaroo	vegetables
17	kangaroo	1022	410	kangaroo	vegetables

In [57]:
```
#并集
pd.merge(zoo,zoo_eat,left_on='animal',right_on='animals',how='outer')   #会把所
```

Out[57]:

	animal	uniq_id	water_need	animals	food
0	elephant	1001.0	500.0	elephant	vegetables
1	elephant	1002.0	600.0	elephant	vegetables
2	elephant	1003.0	550.0	elephant	vegetables
3	tiger	1004.0	300.0	tiger	meat
4	tiger	1005.0	320.0	tiger	meat
5	tiger	1006.0	330.0	tiger	meat
6	tiger	1007.0	290.0	tiger	meat
7	tiger	1008.0	310.0	tiger	meat
8	zebra	1009.0	200.0	zebra	vegetables
9	zebra	1010.0	220.0	zebra	vegetables
10	zebra	1011.0	240.0	zebra	vegetables
11	zebra	1012.0	230.0	zebra	vegetables
12	zebra	1013.0	220.0	zebra	vegetables
13	zebra	1014.0	100.0	zebra	vegetables
14	zebra	1015.0	80.0	zebra	vegetables
15	lion	1016.0	420.0	NaN	NaN
16	lion	1017.0	600.0	NaN	NaN
17	lion	1018.0	500.0	NaN	NaN
18	lion	1019.0	390.0	NaN	NaN
19	kangaroo	1020.0	410.0	kangaroo	vegetables

	animal	uniq_id	water_need	animals	food
20	kangaroo	1021.0	430.0	kangaroo	vegetables
21	kangaroo	1022.0	410.0	kangaroo	vegetables
22	NaN	NaN	NaN	giraffe	vegetables

In [58]:
```
#并集会出现很多缺失值的情况，可以进行填充
pd.merge(zoo,zoo_eat,left_on='animal',right_on='animals',how='inner').fillna(
```

Out[58]:

	animal	uniq_id	water_need	animals	food
0	elephant	1001	500	elephant	vegetables
1	elephant	1002	600	elephant	vegetables
2	elephant	1003	550	elephant	vegetables
3	tiger	1004	300	tiger	meat
4	tiger	1005	320	tiger	meat
5	tiger	1006	330	tiger	meat
6	tiger	1007	290	tiger	meat
7	tiger	1008	310	tiger	meat
8	zebra	1009	200	zebra	vegetables
9	zebra	1010	220	zebra	vegetables
10	zebra	1011	240	zebra	vegetables
11	zebra	1012	230	zebra	vegetables
12	zebra	1013	220	zebra	vegetables
13	zebra	1014	100	zebra	vegetables
14	zebra	1015	80	zebra	vegetables
15	kangaroo	1020	410	kangaroo	vegetables
16	kangaroo	1021	430	kangaroo	vegetables
17	kangaroo	1022	410	kangaroo	vegetables

In [14]:
```
#左联，left_join
pd.merge(zoo,zoo_eat,left_on='animal',right_on='animals',how='left')
```

Out[14]:

	animal	uniq_id	water_need	animals	food
0	elephant	1001	500	elephant	vegetables
1	elephant	1002	600	elephant	vegetables
2	elephant	1003	550	elephant	vegetables
3	tiger	1004	300	tiger	meat
4	tiger	1005	320	tiger	meat
5	tiger	1006	330	tiger	meat
6	tiger	1007	290	tiger	meat

	animal	uniq_id	water_need	animals	food
7	tiger	1008	310	tiger	meat
8	zebra	1009	200	zebra	vegetables
9	zebra	1010	220	zebra	vegetables
10	zebra	1011	240	zebra	vegetables
11	zebra	1012	230	zebra	vegetables
12	zebra	1013	220	zebra	vegetables
13	zebra	1014	100	zebra	vegetables
14	zebra	1015	80	zebra	vegetables
15	lion	1016	420	NaN	NaN
16	lion	1017	600	NaN	NaN
17	lion	1018	500	NaN	NaN
18	lion	1019	390	NaN	NaN
19	kangaroo	1020	410	kangaroo	vegetables
20	kangaroo	1021	430	kangaroo	vegetables
21	kangaroo	1022	410	kangaroo	vegetables

In [59]:
```
#右联, right_join
pd.merge(zoo,zoo_eat,left_on='animal',right_on='animals',how='right')
```

Out[59]:

	animal	uniq_id	water_need	animals	food
0	elephant	1001.0	500.0	elephant	vegetables
1	elephant	1002.0	600.0	elephant	vegetables
2	elephant	1003.0	550.0	elephant	vegetables
3	tiger	1004.0	300.0	tiger	meat
4	tiger	1005.0	320.0	tiger	meat
5	tiger	1006.0	330.0	tiger	meat
6	tiger	1007.0	290.0	tiger	meat
7	tiger	1008.0	310.0	tiger	meat
8	kangaroo	1020.0	410.0	kangaroo	vegetables
9	kangaroo	1021.0	430.0	kangaroo	vegetables
10	kangaroo	1022.0	410.0	kangaroo	vegetables
11	zebra	1009.0	200.0	zebra	vegetables
12	zebra	1010.0	220.0	zebra	vegetables
13	zebra	1011.0	240.0	zebra	vegetables
14	zebra	1012.0	230.0	zebra	vegetables
15	zebra	1013.0	220.0	zebra	vegetables
16	zebra	1014.0	100.0	zebra	vegetables

	animal	uniq_id	water_need	animals	food
17	zebra	1015.0	80.0	zebra	vegetables
18	NaN	NaN	NaN	giraffe	vegetables

concat连接

In [60]:
```
#轴向连接
zoo1 = pd.read_csv('Data/zoo1.csv')
zoo1
```

Out[60]:

	animal	uniq_id	water_need
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80

In [63]:
```
zoo2 = pd.read_csv('Data/zoo2.csv')
zoo2
```

Out[63]:

	animal	uniq_id	water_need
0	lion	1016	420
1	lion	1017	600
2	lion	1018	500
3	lion	1019	390
4	kangaroo	1020	410
5	kangaroo	1021	430
6	kangaroo	1022	410

In [64]:
```
zoo = pd.concat([zoo1,zoo2])
```

```
zoo
```

Out[64]:

	animal	uniq_id	water_need
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80
0	lion	1016	420
1	lion	1017	600
2	lion	1018	500
3	lion	1019	390
4	kangaroo	1020	410
5	kangaroo	1021	430
6	kangaroo	1022	410

In [65]:

```
#如果列索引不相同
zoo2.columns=['animal','id','water_need']
zoo2
```

Out[65]:

	animal	id	water_need
0	lion	1016	420
1	lion	1017	600
2	lion	1018	500
3	lion	1019	390
4	kangaroo	1020	410
5	kangaroo	1021	430
6	kangaroo	1022	410

In [66]:
```
zoo = pd.concat([zoo1,zoo2])
zoo  #合并后会扩充列
```

Out[66]:

	animal	uniq_id	water_need	id
0	elephant	1001.0	500	NaN
1	elephant	1002.0	600	NaN
2	elephant	1003.0	550	NaN
3	tiger	1004.0	300	NaN
4	tiger	1005.0	320	NaN
5	tiger	1006.0	330	NaN
6	tiger	1007.0	290	NaN
7	tiger	1008.0	310	NaN
8	zebra	1009.0	200	NaN
9	zebra	1010.0	220	NaN
10	zebra	1011.0	240	NaN
11	zebra	1012.0	230	NaN
12	zebra	1013.0	220	NaN
13	zebra	1014.0	100	NaN
14	zebra	1015.0	80	NaN
0	lion	NaN	420	1016.0
1	lion	NaN	600	1017.0
2	lion	NaN	500	1018.0
3	lion	NaN	390	1019.0
4	kangaroo	NaN	410	1020.0
5	kangaroo	NaN	430	1021.0
6	kangaroo	NaN	410	1022.0

In [67]:
```
zoo1 = pd.read_csv('Data/zoo1.csv')
zoo1
```

Out[67]:

	animal	uniq_id	water_need
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310

	animal	uniq_id	water_need
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80

In [68]:
```
zoo2 = pd.read_csv('Data/zoo2.csv')
zoo2
```

Out[68]:

	animal	uniq_id	water_need
0	lion	1016	420
1	lion	1017	600
2	lion	1018	500
3	lion	1019	390
4	kangaroo	1020	410
5	kangaroo	1021	430
6	kangaroo	1022	410

In [70]:
```
zoo = pd.concat([zoo1,zoo2],axis = 1)
zoo
```

Out[70]:

	animal	uniq_id	water_need	animal	uniq_id	water_need
0	elephant	1001	500	lion	1016.0	420.0
1	elephant	1002	600	lion	1017.0	600.0
2	elephant	1003	550	lion	1018.0	500.0
3	tiger	1004	300	lion	1019.0	390.0
4	tiger	1005	320	kangaroo	1020.0	410.0
5	tiger	1006	330	kangaroo	1021.0	430.0
6	tiger	1007	290	kangaroo	1022.0	410.0
7	tiger	1008	310	NaN	NaN	NaN
8	zebra	1009	200	NaN	NaN	NaN
9	zebra	1010	220	NaN	NaN	NaN
10	zebra	1011	240	NaN	NaN	NaN
11	zebra	1012	230	NaN	NaN	NaN
12	zebra	1013	220	NaN	NaN	NaN
13	zebra	1014	100	NaN	NaN	NaN

	animal	uniq_id	water_need	animal	uniq_id	water_need
14	zebra	1015	80	NaN	NaN	NaN

In [71]:
```
zoo = pd.concat([zoo1,zoo2],axis = 0) #默认是axis=0
zoo
```

Out[71]:

	animal	uniq_id	water_need
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80
0	lion	1016	420
1	lion	1017	600
2	lion	1018	500
3	lion	1019	390
4	kangaroo	1020	410
5	kangaroo	1021	430
6	kangaroo	1022	410

In [72]:
```
zoo = pd.concat([zoo1,zoo2],axis = 0,ignore_index=True)   #解决行索引不连续的问题
zoo
```

Out[72]:

	animal	uniq_id	water_need
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320

	animal	uniq_id	water_need
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80
15	lion	1016	420
16	lion	1017	600
17	lion	1018	500
18	lion	1019	390
19	kangaroo	1020	410
20	kangaroo	1021	430
21	kangaroo	1022	410

去重

In [73]:
```
data = pd.concat([zoo,zoo1])
data
```

Out[73]:

	animal	uniq_id	water_need
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220

	animal	uniq_id	water_need
13	zebra	1014	100
14	zebra	1015	80
15	lion	1016	420
16	lion	1017	600
17	lion	1018	500
18	lion	1019	390
19	kangaroo	1020	410
20	kangaroo	1021	430
21	kangaroo	1022	410
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80

In [74]:
```
data.duplicated()
```

Out[74]:
```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12    False
13    False
14    False
15    False
16    False
```

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

In [35]:
```
data.drop_duplicates()
```

Out[35]:

	animal	uniq_id	water_need
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80
15	lion	1016	420
16	lion	1017	600
17	lion	1018	500
18	lion	1019	390
19	kangaroo	1020	410
20	kangaroo	1021	430
21	kangaroo	1022	410

In [75]:
```
data
```

Out[75]:

	animal	uniq_id	water_need
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80
15	lion	1016	420
16	lion	1017	600
17	lion	1018	500
18	lion	1019	390
19	kangaroo	1020	410
20	kangaroo	1021	430
21	kangaroo	1022	410
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220

	animal	uniq_id	water_need
13	zebra	1014	100
14	zebra	1015	80

In [76]:
```
data.drop_duplicates(inplace=True)    #默认是不改变原数据，如果要改变，则需要加上inpl
#data.drop_duplicates(inplace=True,keep='first')   等价于这个，保留上面的数据，去掉
data
```

Out[76]:

	animal	uniq_id	water_need
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80
15	lion	1016	420
16	lion	1017	600
17	lion	1018	500
18	lion	1019	390
19	kangaroo	1020	410
20	kangaroo	1021	430
21	kangaroo	1022	410

In [77]:
```
data = pd.concat([zoo,zoo1])
data
```

Out[77]:

	animal	uniq_id	water_need
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550

	animal	uniq_id	water_need
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80
15	lion	1016	420
16	lion	1017	600
17	lion	1018	500
18	lion	1019	390
19	kangaroo	1020	410
20	kangaroo	1021	430
21	kangaroo	1022	410
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80

In [78]:
```
data.drop_duplicates(inplace=True,keep='last')   #保留下面的数据，去除上面的重复数据
```

```
data
```

Out[78]:

	animal	uniq_id	water_need
15	lion	1016	420
16	lion	1017	600
17	lion	1018	500
18	lion	1019	390
19	kangaroo	1020	410
20	kangaroo	1021	430
21	kangaroo	1022	410
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80

In [79]:

```
data = pd.concat([zoo,zoo1])
data
```

Out[79]:

	animal	uniq_id	water_need
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310

	animal	uniq_id	water_need
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80
15	lion	1016	420
16	lion	1017	600
17	lion	1018	500
18	lion	1019	390
19	kangaroo	1020	410
20	kangaroo	1021	430
21	kangaroo	1022	410
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80

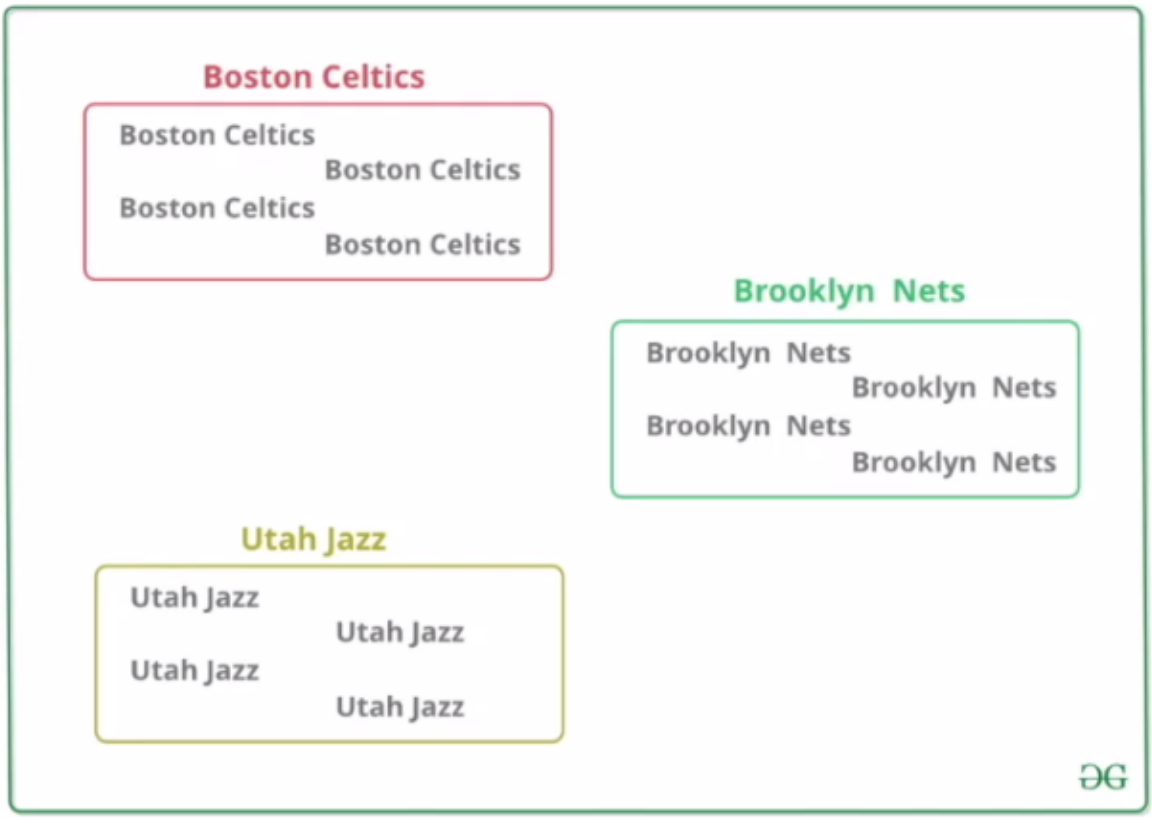
In [45]:
```
data.drop_duplicates(subset=['animal','water_need'],inplace=True,keep='last')
data
```

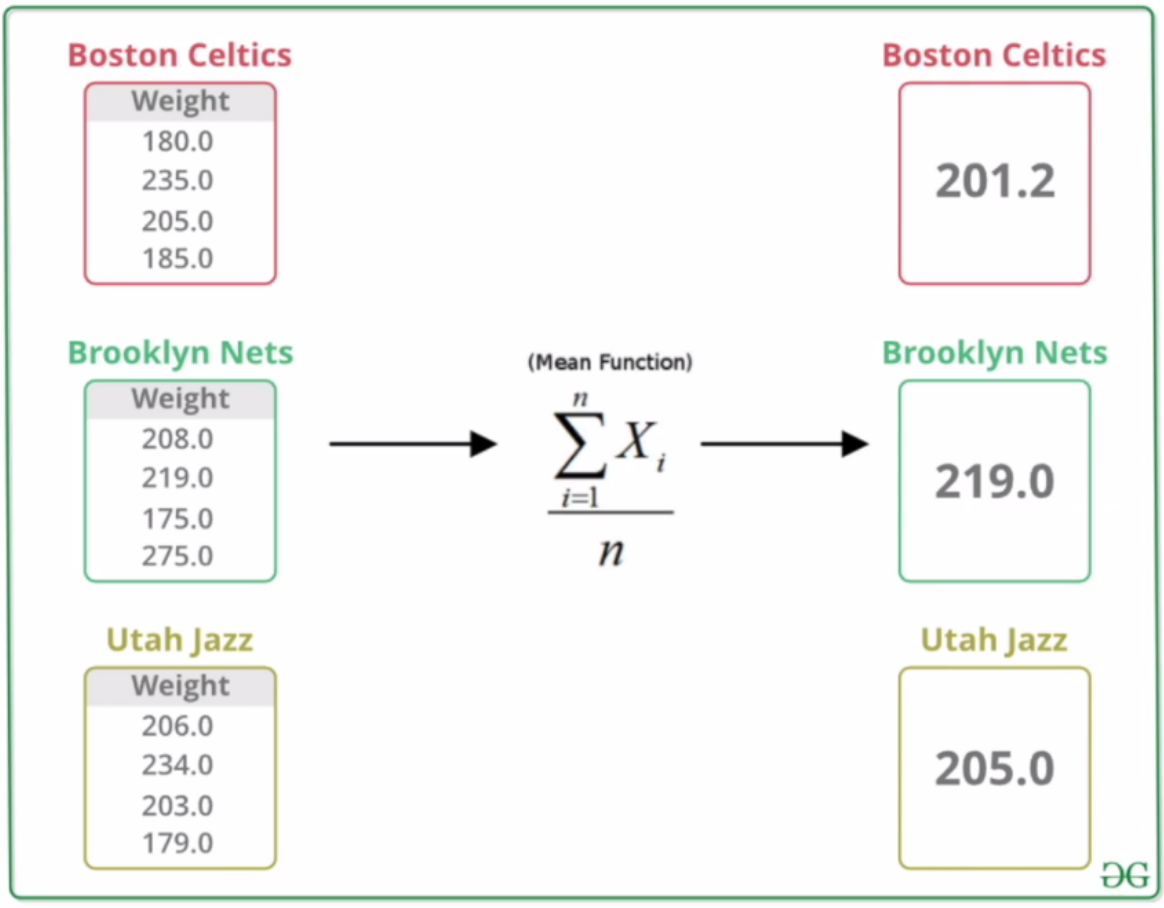
Out[45]:

	animal	uniq_id	water_need
15	lion	1016	420
16	lion	1017	600
17	lion	1018	500

	animal	uniq_id	water_need
18	lion	1019	390
20	kangaroo	1021	430
21	kangaroo	1022	410
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80

groupby分组





In [80]:
```
zoo
```

Out[80]:

	animal	uniq_id	water_need
0	elephant	1001	500

	animal	uniq_id	water_need
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80
15	lion	1016	420
16	lion	1017	600
17	lion	1018	500
18	lion	1019	390
19	kangaroo	1020	410
20	kangaroo	1021	430
21	kangaroo	1022	410

In [81]:
```
grp = zoo.groupby('animal')
type(grp)   #数据类型
```

Out[81]:
```
pandas.core.groupby.generic.DataFrameGroupBy
```

In [83]:
```
zoo.groupby('animal').groups   #分组后的结果
```

Out[83]:
```
{'elephant': [0, 1, 2], 'kangaroo': [19, 20, 21], 'lion': [15, 16, 17, 18],
'tiger': [3, 4, 5, 6, 7], 'zebra': [8, 9, 10, 11, 12, 13, 14]}
```

In [85]:
```
#上述为一个字典对象，不能直接从里面取值
grp = zoo.groupby('animal')
for name,item in grp:
    print(name)
    print(item)
    print('^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^')
```

```
elephant
     animal  uniq_id  water_need
0  elephant     1001         500
1  elephant     1002         600
2  elephant     1003         550
```

```
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
kangaroo
      animal  uniq_id  water_need
19  kangaroo     1020         410
20  kangaroo     1021         430
21  kangaroo     1022         410
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
lion
   animal  uniq_id  water_need
15   lion     1016         420
16   lion     1017         600
17   lion     1018         500
18   lion     1019         390
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
tiger
  animal  uniq_id  water_need
3  tiger     1004         300
4  tiger     1005         320
5  tiger     1006         330
6  tiger     1007         290
7  tiger     1008         310
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
zebra
   animal  uniq_id  water_need
8   zebra     1009         200
9   zebra     1010         220
10  zebra     1011         240
11  zebra     1012         230
12  zebra     1013         220
13  zebra     1014         100
14  zebra     1015          80
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

In [86]:

```
#获取每一组的信息
#get_group() 用于获取具体的数据
grp = zoo.groupby('animal')
grp.get_group('lion')
```

Out[86]:

	animal	uniq_id	water_need
15	lion	1016	420
16	lion	1017	600
17	lion	1018	500
18	lion	1019	390

In [89]:

```
zoo.groupby('animal').sum()
```

Out[89]:

	uniq_id	water_need
animal		
elephant	3006	1650
kangaroo	3063	1250
lion	4070	1910
tiger	5030	1550
zebra	7084	1290

In [90]:

```
zoo.groupby('animal').sum().drop(columns='uniq_id')
```

Out[90]:

animal	water_need
elephant	1650
kangaroo	1250
lion	1910
tiger	1550
zebra	1290

In [91]:
```
zoo.groupby('animal').sum().drop(columns='uniq_id').sort_values('water_need')
```

Out[91]:

animal	water_need
kangaroo	1250
zebra	1290
tiger	1550
elephant	1650
lion	1910

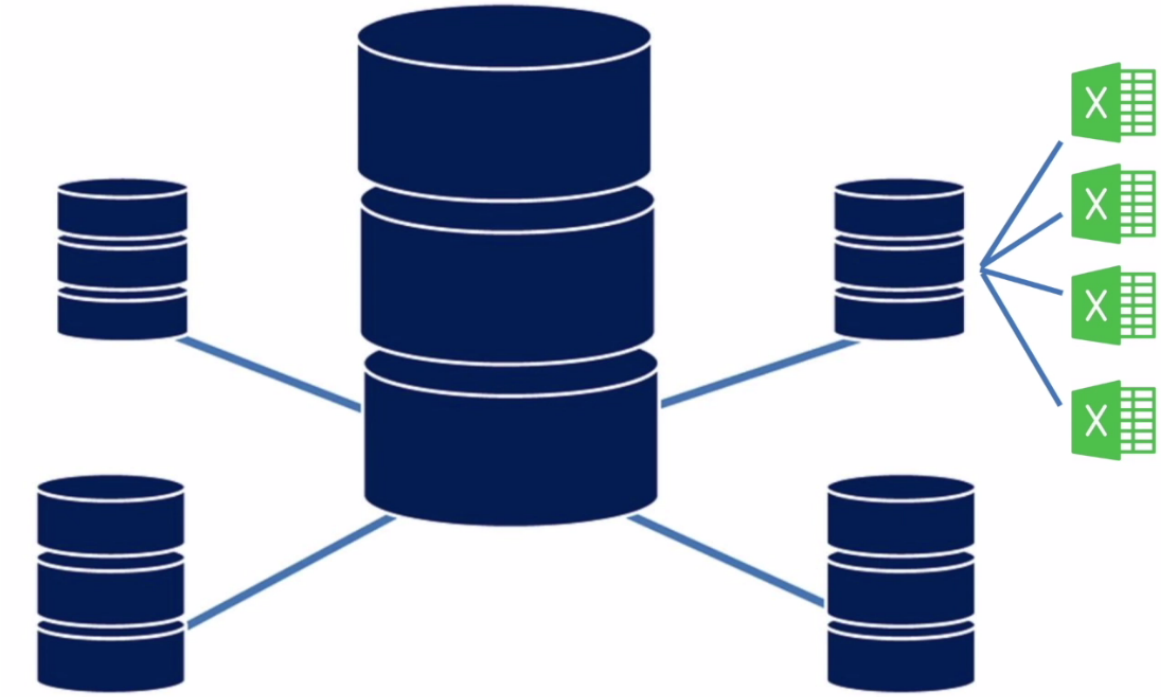
In [92]:
```
zoo.groupby('animal').sum().drop(columns='uniq_id').sort_values('water_need',
```

Out[92]:

animal	water_need
lion	1910
elephant	1650
tiger	1550
zebra	1290
kangaroo	1250

In [ ]:
```
#可以看到，dataframe是支持链式操作的
```

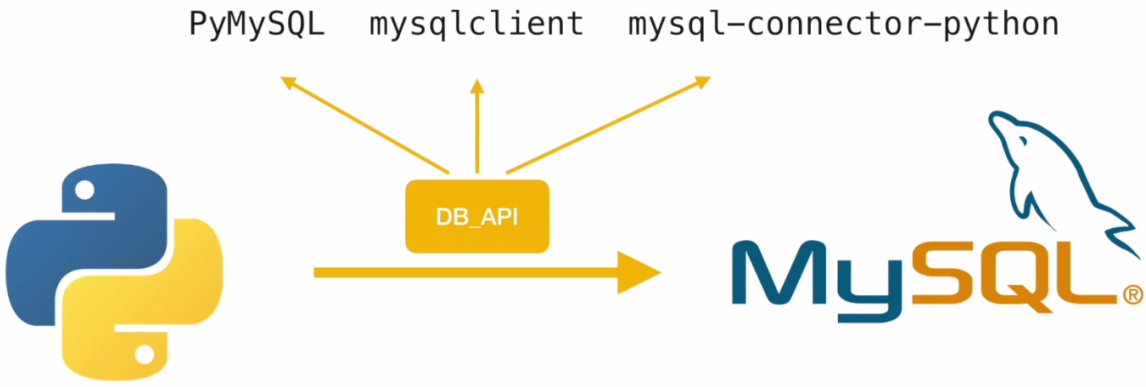
mysql数据库基础及pandas读取mysql数据库



基本概念

- 数据库：一些关联表的集合。
- 数据表：数据的矩阵，在一个数据库中的表看起来像一个简单的电子表格。
- 列：即字段/数据元素，包含了相同的数据，例如邮政编码的数据。
- 行：即元组/记录，是一组相关的数据，例如一条用户订阅的数据。
- 冗余：存储两倍数据，冗余可以使系统速度更快。
- 主键：主键是唯一的。一个数据表中只能包含一个主键。可以使用主键来查询数据。
- 外键：外键用于关联两个表。

python驱动程序

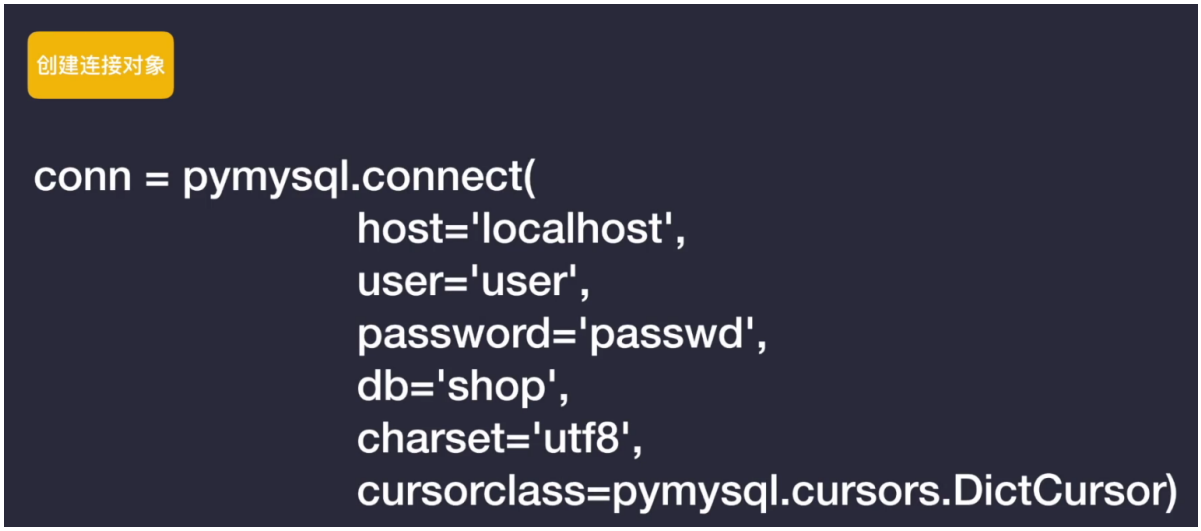


In [5]:
```
#PyMySQL的安装与使用
#cmd下 pip install  pymysql
```

In [94]:
```
#安装mysql数据库以及Navicat可视化操作工具
```

In [95]:
```
#connection连接对象
```

参数	说明
dsn	数据源名称，给出该参数表示数据库依赖
host	主机名
user	用户名
password	用户密码
database	数据库名称



In [99]:
```
import pymysql
conn = pymysql.connect(
                host='localhost',
                user='root',
                password='admin123',
                db='world',
                charset='utf8',
                cursorclass=pymysql.cursors.DictCursor)
conn
```

Out[99]: <pymysql.connections.Connection at 0x2425ddf62e0>

In [100…
```
import pandas as pd
data = pd.read_sql('select * from city',conn)
data
```

Out[100…

	ID	Name	CountryCode	District	Population
0	1	Kabul	AFG	Kabol	1780000

	ID	Name	CountryCode	District	Population
1	2	Qandahar	AFG	Qandahar	237500
2	3	Herat	AFG	Herat	186800
3	4	Mazar-e-Sharif	AFG	Balkh	127800
4	5	Amsterdam	NLD	Noord-Holland	731200
...	...	...	...	...	...
4074	4075	Khan Yunis	PSE	Khan Yunis	123175
4075	4076	Hebron	PSE	Hebron	119401
4076	4077	Jabaliya	PSE	North Gaza	113901
4077	4078	Nablus	PSE	Nablus	100231
4078	4079	Rafah	PSE	Rafah	92020

4079 rows × 5 columns

In [101...

```
#关闭数据库连接
conn.close()
```

In [107...

```
#另一种连接方式
import pymysql
from sqlalchemy import create_engine
import pandas as pd

engine = create_engine('mysql+pymysql://root:admin123@localhost:3306/world')
conn = engine.connect()

data = pd.read_sql('select * from city',conn)
data
```

Out[107...

	ID	Name	CountryCode	District	Population
0	1	Kabul	AFG	Kabol	1780000
1	2	Qandahar	AFG	Qandahar	237500
2	3	Herat	AFG	Herat	186800
3	4	Mazar-e-Sharif	AFG	Balkh	127800
4	5	Amsterdam	NLD	Noord-Holland	731200
...	...	...	...	...	...
4074	4075	Khan Yunis	PSE	Khan Yunis	123175
4075	4076	Hebron	PSE	Hebron	119401
4076	4077	Jabaliya	PSE	North Gaza	113901
4077	4078	Nablus	PSE	Nablus	100231
4078	4079	Rafah	PSE	Rafah	92020

4079 rows × 5 columns

In [108…
```
conn.close()
```

In [109…
```
data
```

Out[109…

	ID	Name	CountryCode	District	Population
0	1	Kabul	AFG	Kabol	1780000
1	2	Qandahar	AFG	Qandahar	237500
2	3	Herat	AFG	Herat	186800
3	4	Mazar-e-Sharif	AFG	Balkh	127800
4	5	Amsterdam	NLD	Noord-Holland	731200
…	…	…	…	…	…
4074	4075	Khan Yunis	PSE	Khan Yunis	123175
4075	4076	Hebron	PSE	Hebron	119401
4076	4077	Jabaliya	PSE	North Gaza	113901
4077	4078	Nablus	PSE	Nablus	100231
4078	4079	Rafah	PSE	Rafah	92020

4079 rows × 5 columns

MySQl查询语句大全 https://www.cnblogs.com/mofujin/p/11355517.html

In [ ]:
```
```