Pandas Merge

Basic Merge Using a Dataframe Column

loading into new data frame

In [1]:
```
import pandas as pd
df1 = pd.DataFrame({
    "city": ["new york","chicago","orlando"],
    "temperature": [21,14,35],
})
df1
```

Out[1]:

	city	temperature
0	new york	21
1	chicago	14
2	orlando	35

In [2]:
```
df2 = pd.DataFrame({
    "city": ["chicago","new york","orlando"],
    "humidity": [65,68,75],
})
df2
```

Out[2]:

	city	humidity
0	chicago	65
1	new york	68
2	orlando	75

In [3]:
```
df3 = pd.merge(df1, df2, on="city")
df3
```

Out[3]:

	city	temperature	humidity
0	new york	21	68
1	chicago	14	65
2	orlando	35	75

Type Of DataBase Joins

In [4]:
```
df1 = pd.DataFrame({
    "city": ["new york","chicago","orlando", "baltimore"],
    "temperature": [21,14,35, 38],
})
df1
```

Out[4]:

	city	temperature
0	new york	21
1	chicago	14
2	orlando	35

	city	temperature
3	baltimore	38

In [5]:
```
df2 = pd.DataFrame({
    "city": ["chicago","new york","san diego"],
    "humidity": [65,68,71],
})
df2
```

Out[5]:

	city	humidity
0	chicago	65
1	new york	68
2	san diego	71

In [6]:
```
df3=pd.merge(df1,df2,on="city",how="inner")
df3
```

Out[6]:

	city	temperature	humidity
0	new york	21	68
1	chicago	14	65

In [7]:
```
df3=pd.merge(df1,df2,on="city",how="outer")
df3
```

Out[7]:

	city	temperature	humidity
0	new york	21.0	68.0
1	chicago	14.0	65.0
2	orlando	35.0	NaN
3	baltimore	38.0	NaN
4	san diego	NaN	71.0

In [8]:
```
df3=pd.merge(df1,df2,on="city",how="left")
df3
```

Out[8]:

	city	temperature	humidity
0	new york	21	68.0
1	chicago	14	65.0
2	orlando	35	NaN
3	baltimore	38	NaN

In [9]:
```
df3=pd.merge(df1,df2,on="city",how="right")#on city condition we merge how we r merging u
df3
```

Out[9]:

	city	temperature	humidity
0	chicago	14.0	65
1	new york	21.0	68
2	san diego	NaN	71

In [10]:
```
df3=pd.merge(df1,df2,on="city",how="outer") #give full data outer is also calles full joi
df3
```

Out[10]:

	city	temperature	humidity
0	new york	21.0	68.0
1	chicago	14.0	65.0
2	orlando	35.0	NaN
3	baltimore	38.0	NaN
4	san diego	NaN	71.0

In [11]:
```
df3=pd.merge(df1,df2,on="city",how="left")# it gives data from left side df1 is left tabl
df3
```

Out[11]:

	city	temperature	humidity
0	new york	21	68.0
1	chicago	14	65.0
2	orlando	35	NaN
3	baltimore	38	NaN

In [12]:
```
df3=pd.merge(df1,df2,on="city",how="right") # it gives data from right side df2 is roght
df3
```

Out[12]:

	city	temperature	humidity
0	chicago	14.0	65
1	new york	21.0	68
2	san diego	NaN	71

indicator flag

to find from which table and where

In [15]:
```
df3=pd.merge(df1,df2,on="city",how="outer",indicator=True)
df3
```

Out[15]:

	city	temperature	humidity	_merge
0	new york	21.0	68.0	both
1	chicago	14.0	65.0	both
2	orlando	35.0	NaN	left_only
3	baltimore	38.0	NaN	left_only
4	san diego	NaN	71.0	right_only

suffixes

if we have 2 dataframe and some of the names r same why to present it has 2 we present it as 1 but if not given suffix the name is duplicates

In [16]:
```
df1 = pd.DataFrame({
    "city": ["new york","chicago","orlando", "baltimore"],
    "temperature": [21,14,35,38],
    "humidity": [65,68,71, 75]
})
df1
```

Out[16]:

	city	temperature	humidity
0	new york	21	65
1	chicago	14	68
2	orlando	35	71
3	baltimore	38	75

In [17]:
```
df2 = pd.DataFrame({
    "city": ["chicago","new york","san diego"],
    "temperature": [21,14,35],
    "humidity": [65,68,71]
})
df2
```

Out[17]:

	city	temperature	humidity
0	chicago	21	65
1	new york	14	68
2	san diego	35	71

In [18]:
```
df3= pd.merge(df1,df2,on="city",how="outer", suffixes=('_2020','_2021')) # we need our da
df3
```

Out[18]:

	city	temperature_2020	humidity_2020	temperature_2021	humidity_2021
0	new york	21.0	65.0	14.0	68.0
1	chicago	14.0	68.0	21.0	65.0
2	orlando	35.0	71.0	NaN	NaN
3	baltimore	38.0	75.0	NaN	NaN
4	san diego	NaN	NaN	35.0	71.0

join

join and merge are bothe are same give us all data

In [19]:
```
df1 = pd.DataFrame({
    "city": ["new york","chicago","orlando"],
    "temperature": [21,14,35],
})
df1.set_index('city',inplace=True)
df1
```

Out[19]:

	temperature
city	
new york	21
chicago	14

city	temperature
orlando	35

In [20]:
```
df2 = pd.DataFrame({
    "city": ["chicago","new york","orl"],
    "temperature": [65,68,75],
})
df2.set_index('city',inplace=True)
df2
```

Out[20]:

city	temperature
chicago	65
new york	68
orl	75

In [21]:
```
df1.join(df2,lsuffix='_jan', rsuffix='_feb')# left is default
```

Out[21]:

city	temperature_jan	temperature_feb
new york	21	68.0
chicago	14	65.0
orlando	35	NaN

In [22]:
```
df1.join(df2,lsuffix='_jan', rsuffix='_feb',how='inner')
```

Out[22]:

city	temperature_jan	temperature_feb
new york	21	68
chicago	14	65

In [24]:
```
df1.join(df2,lsuffix='_jan', rsuffix='_feb',how='right')
```

Out[24]:

city	temperature_jan	temperature_feb
chicago	14.0	65
new york	21.0	68
orl	NaN	75

In [25]:
```
tempdf=df1.join(df2,lsuffix='_jan', rsuffix='_feb')
```

In [26]:
```
tempdf.join(df3,lsuffix='_jan_feb', rsuffix='_mar')
```

Out[26]:

city	temperature_jan	temperature_feb	city	temperature_2020	humidity_2020	temperature_2(

city	temperature_jan	temperature_feb	city	temperature_2020	humidity_2020	temperature_2
new york	21	68.0	NaN	NaN	NaN	I
chicago	14	65.0	NaN	NaN	NaN	I
orlando	35	NaN	NaN	NaN	NaN	I

In [27]:
```
tempdf.join(df3)
```

Out[27]:

city	temperature_jan	temperature_feb	city	temperature_2020	humidity_2020	temperature_2
new york	21	68.0	NaN	NaN	NaN	I
chicago	14	65.0	NaN	NaN	NaN	I
orlando	35	NaN	NaN	NaN	NaN	I

In [ ]:
```

```