

VERİ KÜMELERİNİ BİRLEŞTİRME

PANDAS kütüphanesi kullanılarak veriler birkaç şekilde birleştirilebilir: pandas.merge, veri şablonundaki satırları bir veya daha fazla anahtara göre bağlar.

pandas.concat, bir eksen boyunca verileri birleştirir.

comb_first, bir verideki eksik değerleri diğerinden gelen değerlerle doldurmak için üst üste binen verileri birbirine ekler.

```
In [3]: import pandas as pd  
import numpy as np
```

```
In [4]: df1=pd.DataFrame({'key':['b','b','a','c','a','a','b'],  
                          'data1': range(7)})  
df1
```

```
Out[4]:
```

	key	data1
0	b	0
1	b	1
2	a	2
3	c	3
4	a	4
5	a	5
6	b	6

```
In [5]: df2=pd.DataFrame({'key':['a','b','d'],  
                          'data2': range(3)})  
df2
```

```
Out[5]:
```

	key	data2
0	a	0
1	b	1
2	d	2

```
In [6]: pd.merge(df1,df2, on='key')
```

Out[6]:

	key	data1	data2
0	b	0	1
1	b	1	1
2	a	2	0
3	a	4	0
4	a	5	0
5	b	6	1

	key	data1	data2
0	b	0	1
1	b	1	1
2	a	2	0
3	a	4	0
4	a	5	0
5	b	6	1

```
In [7]: df3=pd.DataFrame({'lkey': ['b', 'b', 'a', 'c', 'a', 'a', 'b'],  
                        'data1': range(7)})  
df3
```

Out[7]:

	lkey	data1
0	b	0
1	b	1
2	a	2
3	c	3
4	a	4
5	a	5
6	b	6

	lkey	data1
0	b	0
1	b	1
2	a	2
3	c	3
4	a	4
5	a	5
6	b	6

```
In [8]: df4=pd.DataFrame({'rkey': ['a', 'b', 'd'],  
                        'data2': range(3)})  
df4
```

Out[8]:

	rkey	data2
0	a	0
1	b	1
2	d	2

	rkey	data2
0	a	0
1	b	1
2	d	2

```
In [9]: pd.merge(df3, df4, left_on='lkey', right_on='rkey')
```

Out[9]:

	lkey	data1	rkey	data2
0	b	0	b	1
1	b	1	b	1
2	a	2	a	0
3	a	4	a	0
4	a	5	a	0
5	b	6	b	1

Birleştirme işlemi sonucunda 'c' ve 'd' değerleri ve ilişkili verilerin eksik olduğu görülüyor. Birleştirme işlemi, anahtar noktalarda kesişimin ve bu kesişime göre her iki tabloda bulunan verilerin ortak kümedir.

Diğer olası birleştirme seçeneği ise hem sol hem sağ birleştirmelerini kullanarak tamamı için verilerin birleşimini vermektir.

```
In [11]: pd.merge(df1, df2, how='outer')
```

Out[11]:

	key	data1	data2
0	a	2.0	0.0
1	a	4.0	0.0
2	a	5.0	0.0
3	b	0.0	1.0
4	b	1.0	1.0
5	b	6.0	1.0
6	c	3.0	NaN
7	d	NaN	2.0

Eğer birleştirme işlemi birden çok anahtarla yapılmak isteniyorsa sütun adları listesinin belirtilmesi gerekmektedir.

```
In [44]: left=pd.DataFrame({'key1': ['elma', 'elma', 'armut'],
                             'key2': ['bir', 'iki', 'bir'],
                             'count1': [1,2,3]})

right=pd.DataFrame({'key1': ['elma', 'elma', 'armut', 'armut'],
                    'key2': ['bir', 'bir', 'bir', 'iki'],
                    'count2': [4,5,6,7]})
```

```
In [46]: left
```

```
Out[46]:
```

	key1	key2	count1
0	elma	bir	1
1	elma	iki	2
2	armut	bir	3

```
In [48]: right
```

```
Out[48]:
```

	key1	key2	count2
0	elma	bir	4
1	elma	bir	5
2	armut	bir	6
3	armut	iki	7

```
In [50]: pd.merge(left, right, on=['key1','key2'],how='outer')
```

```
Out[50]:
```

	key1	key2	count1	count2
0	armut	bir	3.0	6.0
1	armut	iki	NaN	7.0
2	elma	bir	1.0	4.0
3	elma	bir	1.0	5.0
4	elma	iki	2.0	NaN

Veri şablonu nesnelerinde çakışan adlar olması durumunda otomatik olarak _x, _y son ekleri gelir. İstenirse suffixes komutu ile değişken adının sonuna eklenecek olan ek kullanıcı tarafından belirlenebilir.

```
In [53]: pd.merge(left, right, on='key1')
```

```
Out[53]:
```

	key1	key2_x	count1	key2_y	count2
0	elma	bir	1	bir	4
1	elma	bir	1	bir	5
2	elma	iki	2	bir	4
3	elma	iki	2	bir	5
4	armut	bir	3	bir	6
5	armut	bir	3	iki	7

```
In [55]: pd.merge(left, right, on='key1', suffixes=('_left', '_right'))
```

Out[55]:

	key1	key2_left	count1	key2_right	count2
0	elma	bir	1	bir	4
1	elma	bir	1	bir	5
2	elma	iki	2	bir	4
3	elma	iki	2	bir	5
4	armut	bir	3	bir	6
5	armut	bir	3	iki	7