Merge.

tcs	wipro
101	101
102	102
103	103
106	105

id

Right join

left join

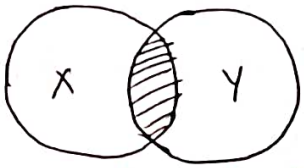
inner join

The Pandas module contains various features to perform various operations on Dataframes like join, concatenate, delete, add, etc.

There are five types of Joins in Pandas.

Inner Join → Inner join is the most common type of join you'll be working with. It returns a Dataframe with only those rows that have common characteristics.
This is similar to the intersection of two sets.

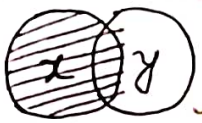
how='inner'



natural join

Left Join → With a left outer join, all the records from the first Dataframe will be displayed, irrespective of whether the keys in the first Dataframe can be found in the second Dataframes. Whereas, for the second Dataframe, only the records with the keys in the second Dataframe that can be found in the first Dataframe will be displayed.

how=left



left outer join

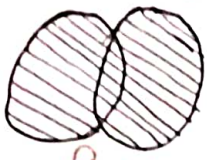
Right Outer Join – For a right join, all the records from the second Dataframe will be displayed. However, only the records with the keys in the first Dataframe that can be found in the second Dataframe will be displayed.

how = (right)

right outer join

Full Outer Join - All outer join returns all the rows from the left Dataframe, and all the rows from the right Dataframe, and matches up rows where possible, with NaNs elsewhere. But if the Dataframe is complete, then we get the same output.

how = 'outer'



Full outer join

Index Join — To merge the Dataframe on indices pass the left_index and right_index arguments as True i.e. both the Dataframes are merged on an index using default Inner Join.

Merge

Merge is a process to merge two tables horizontally.

```
tcs = {'id':[101, 102, 103, 106],
       'name': ['Mithun', 'Dipin', 'jose', 'maneesh'],
       'age' :[25, 26, 27, 29]}
wipro = {'id': [101, 102, 103, 105],
         'name': ['dev', 'sreenath', 'sreeraj', 'varun'],
         'salary':[25000, 34000, 37000, 20000]}

tcs_emp = pd.DataFrame(tcs)
wipro_emp = pd.DataFrame(wipro)
```

pd.concat([tcs_emp, wipro_emp])

Output

	id	Name	Age
0	101	Mithun	25
1	102	Dipin	26
2	103	jose	27
3	104	Rahul	28
0	101	dev	24
1	102	sreenath	24
2	103	sreerag	27
3	104	shafeel	26

pd.concat([tcs_emp, wipro_emp], ignore_index=True)

Output

	id	Name	Age
0	101	Mithun	25
1	102	Dipin	26
2	103	jose	27
3	104	Rahul	28
4	101	dev	24
5	102	sreenath	24
6	103	sreerag	27
7	104	shafeel	26

data = pd.concat([tcs_emp, wipro_emp], keys=["TCS", "WIPRO"])

call data

Output

		id	Name	Age
TCS	0	101	Mithun	25
	1	102	Dipin	26
	2	103	jose	27
	3	104	Rahul	28
WIPRO	0	101	dev	24
	1	102	sreenath	24
	2	103	sreerag	27
	3	104	shafeel	26

data = pd.concat([tcs_emp, wipro_emp], axis=1)

call data

Output

	id	Name	Age	id	Name	Age
0	101	Mithun	25	101	dev	24
1	102	Dipin	26	102	sreenath	24
2	103	jose	27	103	sreerag	27
3	104	Rahul	28	104	shafeel	26

```
Salary = pd.Series([65000, 75000, 15000, 25000],
                   name='salary')
pd.concat([tcs_emp, salary], axis=1)
```

Output

	id	Name	Age	Salary
0	101	Mithun	25	65000
1	102	Dipin	26	75000
2	103	jose	27	15000
3	104	Rahul	28	25000

```
tcs = {'id':[101,102,103,106],
       'Name':['Mithun','Dipin','jose','maneesh'],
       'Age':[25,26,27,29]}
wipro = {'id':[101,102,103,105],
         'name':['dev','sreenath','sreerag','varun'],
         'Salary':[25000,34000,37000,20000]}
tcs_emp = pd.DataFrame(tcs)
wipro_emp = pd.DataFrame(wipro)
pd.merge(tcs_emp, wipro_emp, on='id')
pd.merge(tcs_emp, wipro_emp, on='id',
         suffixes=['left','right'])
```

Both have same output

Output

	id	Name	Age	name	salary
0	101	Mithun	25	dev	25000
1	102	Dipin	26	Sreenath	34000
2	103	jose	27	Sreerag	37000

pd.merge(tcs_emp, wipro_emp, on='id', how='outer')

Output

	id	Name	Age	name	salary
0	101	Hithun	25.0	dev	25000.0
1	102	Dipin	26.0	sreenath	34000.0
2	103	jose	27.0	sreeraj	37000.0
3	106	maneesh	29.0	NaN	NaN
4	105	NaN	NaN	varun	20000

pd.merge(tcs_emp, wipro_emp, on='id', how='outer', indicator=True)

Output

	id	Name	Age	name	salary	_merge
0	101	Hithun	25.0	dev	25000.0	both
1	102	Dipin	26.0	sreenath	34000.0	both
2	103	jose	27.0	sreeraj	37000.0	both
3	106	maneesh	29.0	NaN	NaN	left_only
4	105	NaN	NaN	varun	20000.0	right_only

pd.merge(tcs_emp, wipro_emp, on='id', how='left')

Output

	id	Name	Age	name	salary
0	101	Hithun	25	dev	25000.0
1	102	Dipin	26	sreenath	34000.0
2	103	jose	27	sreeraj	37000.0
3	106	maneesh	29	NaN	NaN

pd.merge(tcs_emp, wipro_emp, on='id', how='right')

Output

	id	Name	Age	name	salary
0	101	Hithun	25.0	dev	25000
1	102	Dipin	26.0	sreenath	34000
2	103	jose	27.0	sreeraj	37000
3	105	NaN	NaN	varun	20000