

## \* Merge Dataframe

Pandas merge() is defined as the Process of bringing the two datasets together into one and aligning the rows based on the common attributes or columns.

Notes by: Web Developers

```
import pandas as pd
help(pd.merge) (Imp.)
```

### Syntax

dataframe.merge(right, how, on, left\_on, right\_on, left\_index, right\_index, True/False, sort, suffixes, copy, indicator, validate)

	right	on	left_index	right_index	True/False	left	right	String list	String list
String list	→				↓	↓	↓		
					↓	↓	↓		
					True/False	list	T/F	String	String
					↓	↓	↓	↓	↓
					Outer	inner	Cross		

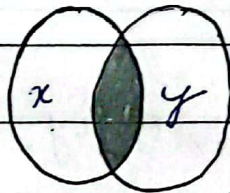
## Types of Merge Operations

- 1) inner Join
- 2) left Join | left Outer Join
- 3) Right Join | Right Outer Join
- 4) Outer Join | full Outer Join
- 5) Cross Join

(Imp.)

## 1) Inner Join

- finds only common records from both tables
- By default, Pandas merge operation works on inner join.



INNER JOIN

## Example

```
import pandas as pd
```

```
data1 = {
```

```
    'EmpID': ['E1', 'E2', 'E3'],
```

```
    'ENAME': ['Palvi', 'JP', 'Web'],
```

```
    'DeptID': ['D1', 'D3', 'D5']
```

```
}
```

```
emp = pd.DataFrame(data1)
```

```
data2 = {
```

```
    'DeptID': ['D1', 'D2', 'D3'],
```

```
    'DeptName': ['Teaching', 'HR', 'Sales']
```

```
}
```

```
departments = pd.DataFrame(data2)
```

```
df_merge = pd.merge(emp, departments, on='DeptID')
```

how = 'inner', sort = 'True')  
print(df.merge)

Output:-

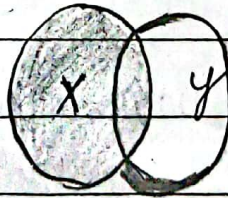
	EmpID	Name	DeptID	DeptName
0	E1	Palvi	D1	Teaching
1	E2	JP	D3	Sales

Notes by: JP web developers

2) Left Join | Left Outer Join

→ It includes all the rows from the left Dataframe and matched rows from the right Data frame.

If there is no match, it fills with Nan.  
how = 'left'



left Outer join

Example

```
import pandas as pd
```

People = pd.DataFrame ({  
'PID': ['P1', 'P2', 'P3'],  
'Name': ['Palvi', 'JP', 'web'],  
'CityID': ['C1', 'C2', 'C4']  
})

Cities = pd.DataFrame ({  
'CityID': ['C1', 'C2', 'C3'],  
'CityName': ['Malaut', 'Bathinda', 'Abohar']  
})

merged\_df = pd.merge (people, cities, on = 'CityID',  
how = 'left')

print (merged\_df)

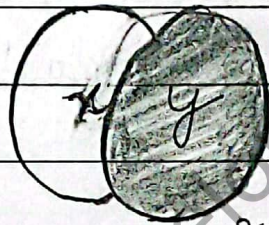
Output:-

	PID	Name	CityID	CityName
0	P1	Palvi	C1	Malaut
1	P2	JP	C2	Bathinda
2	P3	web	C4	NaN

Notes by: jpwwebdevelopers

③ Right Join | Right Outer Join

A Right Join includes all rows from the right DataFrame and matching rows from the left DataFrame.



right Outer join

Program

```
import pandas as pd
people = pd.DataFrame({
    'PID': ['P1', 'P2', 'P3'],
    'Name': ['Palvi', 'jp', 'web'],
    'CityID': ['C1', 'C2', 'C3']
})
```

```
cities = pd.DataFrame({
    'CityID': ['C1', 'C2', 'C3'],
    'City Name': ['Malat', 'Bathinda', 'Abhan']
})
```

```
merged_db = pd.merge(people, cities, on='CityID',
                    how='right')
```

Notes by - jpnwebdevelopers

print (merged\_db)

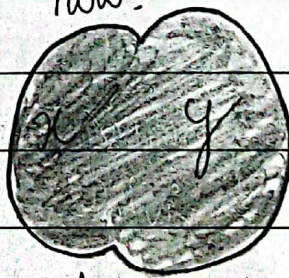
Output	ID	Name	CityID	CityName
0	P1	Palvi	C1	Malout
1	P2	JP	C2	Bathinda
2	NaN	NaN	C3	Abohar

#### ④ Outer Join

→ An Outer Join combines all rows from both DataFrames.

→ If there is no match for a row from one DataFrame, it fills the missing side with NaN values.

how='outer'



full Outer join

#### Program

```
import pandas as pd
data1 = {'ID': [1, 2, 3],
         'Name': ['palvi', 'JP', 'web']}
```

Notes by:- JPwebdevelopers.

```
data2 = { 'ID' : [3,4,5],
          'Age' : [20,25,50] }
```

```
df1 = pd.DataFrame(data1)
df2 = pd.DataFrame(data2)
```

```
merged_df = pd.merge(df1, df2, on = 'ID',
                     how = 'outer')
print(merged_df)
```

Output

	ID	Name	Age
0	1	Palvi	NaN
1	2	JP	NaN
2	3	Web	20.0
3	4	NaN	25.0
4	5	NaN	50.0

5 Cross Join

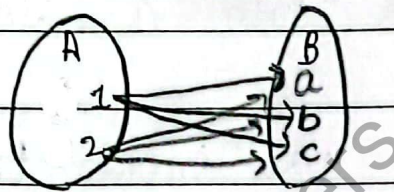
A Cross Join returns the Cartesian Product of two Dataframes.

Every row in the first Dataframe is paired with

every row in the second Dataframe.

→ There is no need for a common column to join or

→ It results in a combination of all possible pairs of rows between the two Dataframes.



### Program

```
import pandas as pd
data1 = {'Id': [1, 2], 'Name': ['Palvi', 'JP']}
data2 = {'City': ['Malout', 'Bathinda']}
df1 = pd.DataFrame(data1)
df2 = pd.DataFrame(data2)
result = pd.merge(df1, df2, how='cross')
print(result)
```

### Output

	Id	Name	City
0	1	Palvi	Malout
1	1	Palvi	Bathinda
2	2	JP	Malout
3	2	JP	Bathinda