Data Science – Pandas – DataFrame – Merging or Joining

17. Pandas – DataFrame - Merging or Joining

Contents

1. Introduction	2
2. merge(p1, p2, p3, p4) function	2
3. Types of joins	2
3.1. Inner join	
3.2. Left join	8
3.3. Right join	11
3.4. Outer Merge / Full outer join	14
4. Other type of joins	17
4.1. One to one	18
4.2. Many to one	20
4.3 Many to many	22

17. Pandas - DataFrame - Merging or Joining

1. Introduction

- ✓ By using pandas we can perform join operations as well.
- ✓ This is same as join operations in database.
- ✓ Here we are performing join in between the DataFrames
- ✓ Joining is the process of bringing two DataFrames into one DataFrame based on common attributes in columns

2. merge(p1, p2, p3, p4) function

- ✓ merge(p1, p2, p3, p4) is a predefined function in pandas.
 - o pd.merge(df1, df2, on = "column", how = "type of join")
- ✓ We should access this function by using pandas library
- ✓ By using this function we can perform join operations over DataFrames

'how' Argument

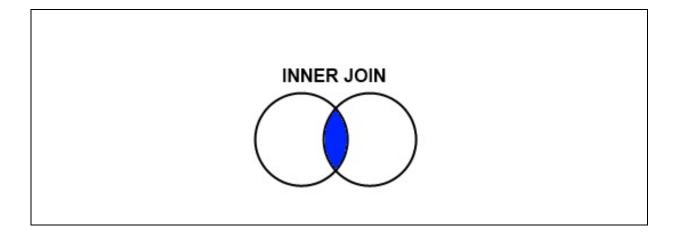
✓ The how argument helps to specify the type of join.

3. Types of joins

- ✓ Inner Merge / Inner join
- ✓ Left Merge / Left outer join
- √ Right Merge / Right outer join
- ✓ Outer Merge / Full outer join

3.1. Inner join

- ✓ We can perform inner join on DataFrames.
- ✓ This inner join is a kind of intersection means, it keeps the common data.



merge(df1, df2, on = "column", how = "type")

- ✓ merge(df1, df2, on = "column", how = "type") is predefined function in pandas.
- ✓ To perform inner join, we need to pass how value as "inner" as per the syntax

Syntax

pd.merge(df1, df2, on = "column", how = "inner")

```
Program
            Creating two DataFrames
Name
            demo1.py
            import pandas as pd
            d1 = {
                  "Id": [1, 2, 3, 4, 5, 6],
                  "Name": ["Pradhan", "Venu", "Madhurima", "Nireekshan",
                  "Shafi", "Veeru"],
                  "Subject": ["English", "Java", "Html", "Python", "C", "dot
                  net"]
            }
            d2 = {
                  "Id": [11, 12, 13, 14, 15, 16],
                  "Name": ["Srinu", "Sumanth", "Neelima", "Daniel", "Arjun",
                  "Veeru"],
                  "Subject": ["Java", "Html", "Cpp", "Python", "C", "dot net"]
            }
            df1 = pd.DataFrame(d1)
            df2 = pd.DataFrame(d2)
            print(df1)
            print()
            print(df2)
```

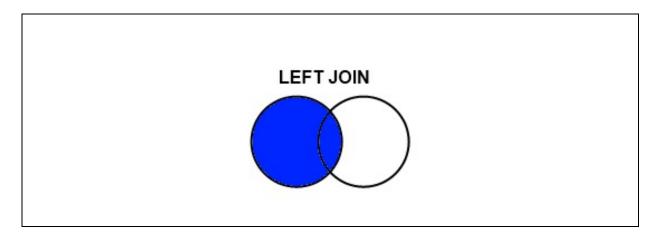
```
Subject
Ιd
         Name
 1
       Pradhan English
 2
         Venu
                  Java
 3
    Madhurima
                  Html
 4
    Nireekshan
                Python
        Shafi
 5
                     C
 6
               dot net
        Veeru
            Subject
Ιd
      Name
      Srinu
11
               Java
12
    Sumanth
               Html
13
    Neelima
                Срр
    Daniel
14
            Python
15
     Arjun
                  C
      Veeru
            dot net
16
```

```
Program
            Inner Join
Name
            demo2.py
            import pandas as pd
            d1 = {
                  "Id":[1, 2, 3, 4, 5, 6],
                  "Name": ["Pradhan", "Venu", "Madhurima", "Nireekshan",
                  "Shafi", "Veeru"],
                  "Subject":["English", "Java", "Html", "Python", "C", "dot
                  net"]
            }
            d2 = {
                  "Id":[11, 12, 13, 14, 15, 16],
                  "Name": ["Srinu", "Sumanth", "Neelima", "Daniel", "Arjun",
                   "Veeru"],
                  "Subject":["Java", "Html", "Cpp", "Python", "C", "dot net"]
            }
            df1 = pd.DataFrame(d1)
            df2 = pd.DataFrame(d2)
            inn join = pd.merge(df1, df2, on = "Subject", how = "inner")
            print(df1)
            print()
            print(df2)
            print()
            print(inn join)
```

```
Ιd
             Name
                   Subject
                   English
   1
          Pradhan
0
1
2
   2
             Venu
                      Java
   3
       Madhurima
                      Html
                    Python
   4
       Nireekshan
   5
            Shafi
                   dot net
   6
            Veeru
   Ιd
          Name
                Subject
   11
         Srinu
                   Java
  12
       Sumanth
                   Html
  13
      Neelima
                    Срр
3
  14
       Daniel
                 Python
        Arjun
  15
                dot net
  16
        Veeru
             Name_x Subject Id_y
   Id_x
                                     Name y
      2
               Venu
                        Java
                                11
                                       Srinu
1
2
3
      3
          Madhurima
                        Html
                                12
                                    Sumanth
      4
        Nireekshan
                      Python
                                     Daniel
                                14
      5
              Shafi
                                15
                                       Arjun
                           C
      6
                     dot net
                                      Veeru
              Veeru
                                16
```

3.2. Left join

- ✓ Keep every row in the left dataframe.
- ✓ Where there are missing values of the "on" variable in the right dataframe filled with NaN values in the result.

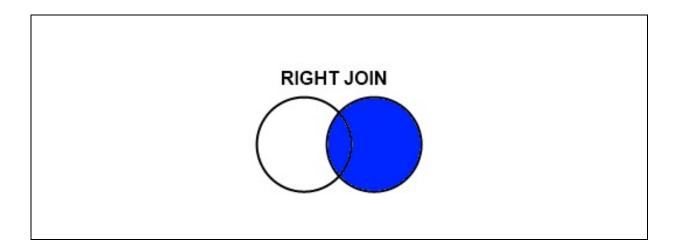


```
Left Join
Program
Name
            demo3.py
            import pandas as pd
            d1 = {
                   "ld":[1, 2, 3, 4, 5, 6],
                   "Name": ["Pradhan", "Venu", "Madhurima", "Nireekshan",
                  "Shafi", "Veeru"],
                   "Subject":["English", "Java", "Html", "Python", "C", "dot
                   net"]
            }
            d2 = {
                   "Id":[11, 12, 13, 14, 15, 16],
                  "Name": ["Srinu", "Sumanth", "Neelima", "Daniel", "Arjun",
                   "Veeru"],
                   "Subject":["Java", "Html", "Cpp", "Python", "C", "dot net"]
            }
            df1 = pd.DataFrame(d1)
            df2 = pd.DataFrame(d2)
            left_join = pd.merge(df1, df2, on = "Subject", how = "left")
            print(df1)
            print()
            print(df2)
            print()
            print(left join)
```

```
Ιd
            Name
                  Subject
         Pradhan
                  English
   1
   2
            Venu
                     Java
       Madhurima
                     Html
   3
      Nireekshan
   4
                  Python
   5
           Shafi
                        C
   6
           Veeru
                  dot net
               Subject
  Ιd
         Name
  11
        Srinu
                  Java
                  Htm1
  12
      Sumanth
  13
      Neelima
                   Срр
               Python
  14
      Daniel
  15
        Arjun
                     C
               dot net
  16
        Veeru
  Id_x
                    Subject Id_y
            Name_x
                                    Name_y
                    English
           Pradhan
                            NaN
                                       NaN
     1
0
1
2
3
4
5
     2
              Venu
                       Java
                             11.0
                                     Srinu
     3
         Madhurima
                       Html 12.0
                                   Sumanth
        Nireekshan
                     Python
                            14.0
                                    Daniel
     4
             Shafi
                             15.0
                                     Arjun
                          С
     6
             Veeru
                    dot net
                             16.0
                                     Veeru
```

3.3. Right join

- ✓ Keep every row in the right dataframe.
- ✓ Where there are missing values of the "on" variable in the left column filled with NaN values in the result.

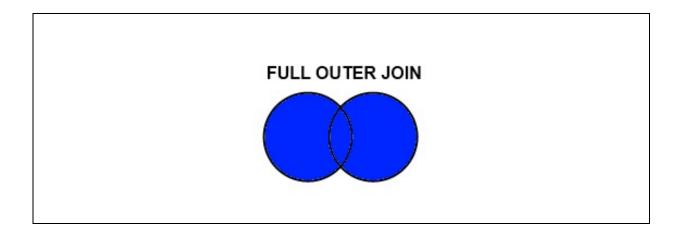


```
Right Join
Program
Name
            demo4.py
            import pandas as pd
            d1 = {
                   "ld":[1, 2, 3, 4, 5, 6],
                   "Name": ["Pradhan", "Venu", "Madhurima", "Nireekshan",
                  "Shafi", "Veeru"],
                   "Subject":["English", "Java", "Html", "Python", "C", "dot
                   net"]
            }
            d2 = {
                   "Id":[11, 12, 13, 14, 15, 16],
                  "Name": ["Srinu", "Sumanth", "Neelima", "Daniel", "Arjun",
                   "Veeru"],
                   "Subject":["Java", "Html", "Cpp", "Python", "C", "dot net"]
            }
            df1 = pd.DataFrame(d1)
            df2 = pd.DataFrame(d2)
            right_join = pd.merge(df1, df2, on = "Subject", how = "right")
            print(df1)
            print()
            print(df2)
            print()
            print(right join)
```

```
Subject
   Ιd
             Name
          Pradhan English
   1
   2
             Venu
                      Java
       Madhurima
                      Html
   3
                    Python
   4
      Nireekshan
   5
            Shafi
   6
                   dot net
            Veeru
         Name
                Subject
   Ιd
  11
         Srinu
                   Java
  12
      Sumanth
                   Html
      Neelima
  13
                    Срр
       Daniel
                 Python
  14
  15
        Arjun
                dot net
  16
        Veeru
                     Subject
  Id x
            Name x
                              Id y
                                     Name y
                                      Srinu
    2.0
               Venu
                        Java
                                11
         Madhurima
                        Html
    3.0
                                12
                                    Sumanth
2
3
4
5
                         Срр
                                    Neelima
                                13
   NaN
                NaN
         Nireekshan
                      Python
                                     Daniel
   4.0
                                14
                                      Arjun
    5.0
              Shafi
                                15
                     dot net
                                      Veeru
   6.0
              Veeru
                                16
```

3.4. Outer Merge / Full outer join

- ✓ A full outer join returns all the rows from the left and right DataFrames.
- ✓ Where there is no common data there it will be fills with NaN values.



```
Program
            Outer Join
Name
            demo5.py
            import pandas as pd
            d1 = {
                  "Id":[1, 2, 3, 4, 5, 6],
                  "Name": ["Pradhan", "Venu", "Madhurima", "Nireekshan",
                  "Shafi", "Veeru"],
                  "Subject":["English", "Java", "Html", "Python", "C", "dot
                  net"]
            }
            d2 = {
                   "Id":[11, 12, 13, 14, 15, 16],
                  "Name": ["Srinu", "Sumanth", "Neelima", "Daniel", "Arjun",
                   "Veeru"],
                  "Subject":["Java", "Html", "Cpp", "Python", "C", "dot net"]
            }
            df1 = pd.DataFrame(d1)
            df2 = pd.DataFrame(d2)
            outer_join = pd.merge(df1, df2, on = "Subject", how = "outer")
            print(df1)
            print()
            print(df2)
            print()
            print(outer join)
```

```
Subject
Ιd
         Name
      Pradhan English
 1
 2
                  Java
         Venu
    Madhurima
 3
                  Html
                Python
 4
   Nireekshan
 5
        Shafi
                    C
 6
        Veeru
               dot net
            Subject
Ιd
      Name
11
     Srinu
               Java
12
   Sumanth
               Html
13 Neelima
                Срр
    Daniel
             Python
14
15
     Arjun
                 C
            dot net
16
     Veeru
                         Id_y
Id x
         Name x Subject
                                Name y
        Pradhan English
 1.0
                         NaN
                                   NaN
 2.0
           Venu
                    Java
                         11.0
                                 Srinu
     Madhurima
                   Html
 3.0
                         12.0
                               Sumanth
     Nireekshan
 4.0
                  Python
                                Daniel
                         14.0
          Shafi
                                 Arjun
 5.0
                     C
                         15.0
                 dot net
 6.0
          Veeru
                         16.0
                                 Veeru
                         13.0
                               Neelima
 NaN
            NaN
                     Cpp
```

4. Other type of joins

- ✓ One to one
- ✓ Many to one
- ✓ Many to many

```
Program
           Creating DataFrames
Name
           demo6.py
           import pandas as pd
           d1 = {
              "Employee": ["Nireekshan", "Veeru", "Lavanya", "Pradhan"],
             "Group": ["Development", "Testing", "Testing", "HR"]
           }
           d2 = {
              "Employee": ["Lavanya", "Nireekshan", "Veeru", "Pradhan"],
              "Hire_date": [2010, 2012, 2014, 2016]
           }
           df1 = pd.DataFrame(d1)
           df2 = pd.DataFrame(d2)
           print(df1)
           print()
           print(df2)
Output
                 Employee
                                Group
              Nireekshan Development
                    Veeru
                               Testing
                               Testing
                  Lavanya
                  Pradhan
                                    HR
                 Employee
                           Hire_date
                  Lavanya
                                2010
              Nireekshan
                                2012
                    Veeru
                                2014
                  Pradhan
                                2016
```

4.1. One to one

✓ This is very simple join and similar to the column-wise concatenation

```
Program
            Creating DataFrames
Name
            demo7.py
            import pandas as pd
            d1 = {
              "Employee": ["Nireekshan", "Veeru", "Lavanya", "Pradhan"],
              "Group": ["Development", "Testing", "Testing", "HR"]
            }
            d2 = {
              "Employee": ["Lavanya", "Nireekshan", "Veeru", "Pradhan"],
              "Hire_date": [2010, 2012, 2014, 2016]
            }
            df1 = pd.DataFrame(d1)
            df2 = pd.DataFrame(d2)
            one_one = pd.merge(df1, df2)
            print(df1)
            print()
            print(df2)
            print()
            print(one one)
```

```
Employee
                     Group
   Nireekshan
               Development
                   Testing
        Veeru
2
                   Testing
      Lavanya
      Pradhan
                        HR
     Employee
               Hire_date
                    2010
      Lavanya
   Nireekshan
                    2012
2
        Veeru
                    2014
      Pradhan
                    2016
     Employee
                     Group
                            Hire_date
   Nireekshan
               Development
                                  2012
                   Testing
        Veeru
                                  2014
2
                   Testing
      Lavanya
                                  2010
      Pradhan
                         HR
                                  2016
```

4.2. Many to one

✓ Many-to-one joins are joins in which one of the two key columns contains duplicate entries

```
Program
            Many to one
            demo8.py
Name
            import pandas as pd
            d1 = {
              "Employee": ["Nireekshan", "Veeru", "Lavanya", "Pradhan"],
              "Group": ["Development", "Testing", "Testing", "HR"]
            }
            d2 = {
              "Employee": ["Lavanya", "Nireekshan", "Veeru", "Pradhan"],
              "Hire date": [2010, 2012, 2014, 2016]
            }
            d3 = {
              "Group": ["Testing", "Development", "HR"],
              "supervisor": ["Shafi", "Daniel", "Neelima"]
            }
            df1 = pd.DataFrame(d1)
            df2 = pd.DataFrame(d2)
            df3 = pd.DataFrame(d3)
            one one = pd.merge(df1, df2)
            many_one = pd.merge(one_one, df3)
            print(df1)
            print()
            print(df2)
            print()
            print(many one)
```

```
Employee
                     Group
  Nireekshan Development
       Veeru
                   Testing
2
      Lavanya
                   Testing
     Pradhan
                        HR
              Hire date
     Employee
     Lavanya
                    2010
  Nireekshan
                    2012
       Veeru
                    2014
     Pradhan
                    2016
    Employee
                            Hire_date supervisor
                     Group
  Nireekshan Development
                                 2012
                                           Daniel
       Veeru
                   Testing
                                 2014
                                            Shafi
                                            Shafi
      Lavanya
                   Testing
                                 2010
                                         Neelima
     Pradhan
                        HR
                                 2016
```

4.3. Many to many

✓ If the key column in both the left and right DataFrame contains duplicates, then the result is a many-to-many merge

```
Program
            Many to many
            demo9.py
Name
            import pandas as pd
            d1 = {
                "Employee": ["Nireekshan", "Veeru", "Lavanya", "Pradhan"],
                "Group": ["Development", "Testing", "Testing", "HR"]
            }
            d2 = {
                "Group": ["Testing", "Testing", "Development",
                "Development", "HR", "HR"],
                "Skills": ["Manual", "Automation", "Coding", "Logical",
                "Spreadsheets", "Organization"]
            }
            df1 = pd.DataFrame(d1)
            df2 = pd.DataFrame(d2)
            many_many = pd.merge(df1, df2)
            print(df1)
            print()
            print(df2)
            print()
            print(many_many)
```

```
Employee
                     Group
  Nireekshan
               Development
        Veeru
                   Testing
2
      Lavanya
                   Testing
      Pradhan
                        HR
                      Skills
         Group
       Testing
                      Manual
                  Automation
       Testing
   Development
                      Coding
   Development
                     Logical
                Spreadsheets
            HR
5
                Organization
            HR
                                   Skills
     Employee
                     Group
  Nireekshan Development
                                   Coding
1
2
3
4
5
                                  Logical
   Nireekshan Development
                   Testing
                                   Manual
        Veeru
        Veeru
                   Testing
                               Automation
      Lavanya
                   Testing
                                   Manual
                   Testing
      Lavanya
                               Automation
      Pradhan
                        HR
                             Spreadsheets
      Pradhan
                        HR
                             Organization
```

Based on column wise

✓ We can also do merge DataFrames based on columns wise as well

```
Program
            Creating DataFrames
            demo10.py
Name
            import pandas as pd
            d1 = {
                   "Id":[1, 2, 3, 4, 5, 6],
                  "Name": ["Pradhan", "Venu", "Madhurima", "Nireekshan",
                  "Shafi", "Veeru"],
                  "Subject":["English", "Java", "Html", "Python", "C", "dot
                  net"]
            }
            d2 = {
                  "ld":[11, 12, 13, 14, 15, 16],
                  "Name": ["Srinu", "Sumanth", "Neelima", "Daniel", "Arjun",
                   "Veeru"],
                  "Subject":["Java", "Html", "Cpp", "Python", "C", "dot net"]
            }
            df1 = pd.DataFrame(d1)
            df2 = pd.DataFrame(d2)
            print(df1)
            print()
            print(df2)
```

```
Name Subject
Ιd
       Pradhan English
 1
 2
                  Java
          Venu
 3
    Madhurima
                  Html
4
    Nireekshan
                 Python
 5
        Shafi
                      С
 6
               dot net
        Veeru
Ιd
      Name
            Subject
      Srinu
11
                Java
12
    Sumanth
                Html
13
   Neelima
                Срр
    Daniel
             Python
14
     Arjun
15
                   C
16
     Veeru
             dot net
```

```
Merging two DataFrames based on column
Program
Name
            demo11.py
            import pandas as pd
            d1 = {
                  "Id":[1, 2, 3, 4, 5, 6],
                  "Name": ["Pradhan", "Venu", "Madhurima", "Nireekshan",
                  "Shafi", "Veeru"],
                  "Subject":["English", "Java", "Html", "Python", "C", "dot
                  net"]
            }
            d2 = {
                  "Id":[11, 12, 13, 14, 15, 16],
                  "Name": ["Srinu", "Sumanth", "Neelima", "Daniel", "Arjun",
                  "Veeru"],
                  "Subject":["Java", "Html", "Cpp", "Python", "C", "dot net"]
            }
            df1 = pd.DataFrame(d1)
            df2 = pd.DataFrame(d2)
            result = pd.merge(df1, df2, on = 'Subject')
            print(df1)
            print()
            print(df2)
            print()
            print(result)
```

```
Ιd
            Name
                  Subject
                  English
   1
         Pradhan
   2
            Venu
                     Java
2
3
       Madhurima
                     Html
   4 Nireekshan
                   Python
           Shafi
   6
           Veeru dot net
         Name Subject
   Ιd
         Srinu
  11
                  Java
                  Htm1
  12
      Sumanth
  13
      Neelima
                   Срр
       Daniel
                Python
  14
  15
         Arjun
  16
        Veeru dot net
   Id_x
            Name_x Subject Id_y
                                    Name_y
              Venu
                       Java
                                     Srinu
      2
                               11
      3
        Madhurima
                       Html
                               12 Sumanth
2
     4 Nireekshan
                     Python
                                    Daniel
                               14
             Shafi
                                     Arjun
                               15
                    dot net
                               16
      6
             Veeru
                                     Veeru
```

```
Merging two DataFrames with multiple columns
Program
Name
            demo12.py
            import pandas as pd
            d1 = {
                  "Id":[1, 2, 3, 4, 5, 6],
                  "Name": ["Pradhan", "Venu", "Madhurima", "Nireekshan",
                  "Shafi", "Veeru"],
                  "Subject":["English", "Java", "Html", "Python", "C", "dot
                  net"]
            }
            d2 = {
                  "Id":[11, 12, 13, 14, 15, 16],
                  "Name": ["Srinu", "Sumanth", "Neelima", "Daniel", "Arjun",
                  "Veeru"],
                  "Subject":["Java", "Html", "Cpp", "Python", "C", "dot net"]
            }
            df1 = pd.DataFrame(d1)
            df2 = pd.DataFrame(d2)
            result = pd.merge(df1, df2, on=["Name", "Subject"])
            print(df1)
            print()
            print(df2)
            print()
            print(result)
```

```
Name
              Subject
Ιd
      Pradhan
               English
 1
 2
         Venu
                 Java
3 Madhurima
                 Html
4 Nireekshan
              Python
5
        Shafi
6
        Veeru dot net
      Name
            Subject
Ιd
11
     Srinu
               Java
              Html
12
   Sumanth
13 Neelima
               Срр
   Daniel
             Python
14
15
     Arjun
            dot net
16
     Veeru
Id_x Name
            Subject
                    Id_y
            dot net
   6
     Veeru
                      16
```