In [1]:
```
import pandas as pd

# 1] FULL OUTER JOIN

dic1 = {'id':['1','2','3','4','5'], 'Value1':['A','C','E','G','I'], 'Value2':['B','D','F','H','J']}


dic2 = {'id':['2','3','6','7','8'], 'Value1':['K','M','O','Q','S'], 'Value2':['L','N','P','R','T']}

df1 = pd.DataFrame(dic1)
df2 = pd.DataFrame(dic2)

# The "full outer join" combines the results of
# both the left and the right outer joins.
# The joined data frame will contain all records
# from both the data frames and fill in NaNs for
# missing matches on either side. You can perform
# a full outer join by specifying the how argument
# as outer in merge() function.

# df3 = pd.merge(df1,df2,on='id',how='outer')

df3 = pd.DataFrame(df1,df2,left_on='id',right_on='id',how='outer',suffixes = ('_left','_right'))

print(df3)
```

```
  id Value1_x Value2_x Value1_y Value2_y
0  1        A        B      NaN      NaN
1  2        C        D        K        L
2  3        E        F        M        N
3  4        G        H      NaN      NaN
4  5        I        J      NaN      NaN
5  6      NaN      NaN        O        P
6  7      NaN      NaN        Q        R
7  8      NaN      NaN        S        T
```

In [2]:
```
df3 = pd.DataFrame(df1,df2,left_on='id',right_on='id',how='outer',suffixes = ('_left','_right'))

# We can specify the columns names to use for merging using "on" parameter.
```

```
# We can use 'left_on' & 'right_on' parameter to specift the columns from the
# left and right dataframes to be used for merging.

print(df3)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[2], line 1
----> 1 df3 = pd.DataFrame(df1,df2,left_on='id',right_on='id',how='outer',suffixes = ('_left','_right'))
      3 # We can specify the columns names to use for merging using "on" parameter.
      4 # We can use 'left_on' & 'right_on' parameter to specift the columns from the
      5 # left and right dataframes to be used for merging.
      7 print(df3)

TypeError: DataFrame.__init__() got an unexpected keyword argument 'left_on'
```

In [4]:
```
dic1 = {'id':['1','2','3','4','5'], 'Value1':['A','C','E','G','I'], 'Value2':['B','D','F','H','J']}


dic2 = {'id':['2','3','6','7','8'], 'Value1':['K','M','O','Q','S'], 'Value2':['L','N','P','R','T']}

df1 = pd.DataFrame(dic1)
df2 = pd.DataFrame(dic2)

# The "inner join" produce only those records that
# match in both the data frame. You have to pass
# inner in how argument inside merge() function.


df3 = pd.merge(df1,df2,on='id',how='inner')
print(df3)
```

```
  id Value1_x Value2_x Value1_y Value2_y
0  2        C        D        K        L
1  3        E        F        M        N
```

In [2]:
```
import pandas as pd

dic1 = {'id':['1','2','3','4','5'], 'Value1':['A','C','E','G','I'], 'Value2':['B','D','F','H','J']}
```

```
dic2 = {'id':['2','3','6','7','8'], 'Value1':['K','M','O','Q','S'], 'Value2':['L','N','P','R','T']}

df1 = pd.DataFrame(dic1)
df2 = pd.DataFrame(dic2)

df4 = pd.merge(df1,df2,on='id',how='right')

# The right join produce a complete set of records from data
# frame B(Right side Data Frame) with the matching records
# (where available) in data frame A( Left side data frame).
# If there is no match right side will contain null. You
# have to pass right in how argument inside merge() function.

print(df4)
```

```
  id Value1_x Value2_x Value1_y Value2_y
0  2        C        D        K        L
1  3        E        F        M        N
2  6      NaN      NaN        O        P
3  7      NaN      NaN        Q        R
4  8      NaN      NaN        S        T
```

In [3]:

```
import pandas as pd

dic1 = {'id':['1','2','3','4','5'], 'Value1':['A','C','E','G','I'], 'Value2':['B','D','F','H','J']}


dic2 = {'id':['2','3','6','7','8'], 'Value1':['K','M','O','Q','S'], 'Value2':['L','N','P','R','T']}

df1 = pd.DataFrame(dic1)
df2 = pd.DataFrame(dic2)

df5 = pd.merge(df1,df2,on='id',how='left')

# The left join produce a complete set of records from data frame
# A(Left side Data Frame) with the matching records (where available)
# in data frame B( Right side data frame). If there is no match left
# side will contain null. You have to pass left in how argument inside
# merge() function.

print(df5)
```

```
  id Value1_x Value2_x Value1_y Value2_y
0  1        A        B      NaN      NaN
1  2        C        D        K        L
2  3        E        F        M        N
3  4        G        H      NaN      NaN
4  5        I        J      NaN      NaN
```

In [4]:
```
import pandas as pd

dic1 = {'id':['1','2','3','4','5'], 'Value1':['A','C','E','G','I'], 'Value2':['B','D','F','H','J']}


dic2 = {'id':['2','3','6','7','8'], 'Value1':['K','M','O','Q','S'], 'Value2':['L','N','P','R','T']}

df1 = pd.DataFrame(dic1)
df2 = pd.DataFrame(dic2)

df6 = pd.merge(df1,df2,right_index=True,left_index=True)

#Sometimes you have to perform the join on the indexes or the row labels.
#For that you have to specify right_index( for the indexes of the right data frame )
#and left_index (for the indexes of left data frame) as True.

print(df6)
```

```
  id_x Value1_x Value2_x id_y Value1_y Value2_y
0    1        A        B    2        K        L
1    2        C        D    3        M        N
2    3        E        F    6        O        P
3    4        G        H    7        Q        R
4    5        I        J    8        S        T
```

In [ ]: