In [1]:

```
# import the pandas library

import pandas as pd
```

In [2]:

```
# Load different sheets from the same excel file
df_GOOGL = pd.read_excel('https://s3.eu-west-1.amazonaws.com/neueda.conygre.com/pydat
df_IBM = pd.read_excel('https://s3.eu-west-1.amazonaws.com/neueda.conygre.com/pydata/
```

In [3]:

```
# Create small dataframes for demonstration
cols = ['High', 'Low']

df1 = df_IBM[cols].loc['2017-Jan-01' : '2017-Jan-17'].sort_index()
df2 = df_GOOGL[cols].loc['2017-Jan-10' : '2017-Jan-24'].sort_index()

# show both dataframes
print("== IBM ==")
display(df1)
print("== GOOGLE ==")
display(df2)
```

```
== IBM ==
```

	High	Low
Date		
2017-01-03	167.869995	166.009995
2017-01-04	169.869995	167.360001
2017-01-05	169.389999	167.259995
2017-01-06	169.919998	167.520004
2017-01-09	169.800003	167.619995
2017-01-10	168.089996	165.339996
2017-01-11	167.759995	165.600006
2017-01-12	168.009995	165.559998
2017-01-13	168.479996	166.880005
2017-01-17	168.179993	166.119995

```
== GOOGLE ==
```

	High	Low
Date		
2017-01-10	829.409973	823.140015
2017-01-11	829.900024	821.469971
2017-01-12	830.380005	821.010010
2017-01-13	834.650024	829.520020
2017-01-17	830.179993	823.200012
2017-01-18	829.809998	824.080017
2017-01-19	833.000000	823.960022
2017-01-20	829.239990	824.599976
2017-01-23	845.539978	828.700012
2017-01-24	851.520020	842.280029

Outer Merge

We will do an outer merge with `indicator=True`

In [4]:

```
# merge how=outer on=Date indicator=True
df_merge = df1.merge(df2, on='Date', how='outer', indicator=True)

df_merge
```

Out[4]:

	High_x	Low_x	High_y	Low_y	_merge
Date					
2017-01-03	167.869995	166.009995	NaN	NaN	left_only
2017-01-04	169.869995	167.360001	NaN	NaN	left_only
2017-01-05	169.389999	167.259995	NaN	NaN	left_only
2017-01-06	169.919998	167.520004	NaN	NaN	left_only
2017-01-09	169.800003	167.619995	NaN	NaN	left_only
2017-01-10	168.089996	165.339996	829.409973	823.140015	both
2017-01-11	167.759995	165.600006	829.900024	821.469971	both
2017-01-12	168.009995	165.559998	830.380005	821.010010	both
2017-01-13	168.479996	166.880005	834.650024	829.520020	both
2017-01-17	168.179993	166.119995	830.179993	823.200012	both
2017-01-18	NaN	NaN	829.809998	824.080017	right_only
2017-01-19	NaN	NaN	833.000000	823.960022	right_only
2017-01-20	NaN	NaN	829.239990	824.599976	right_only
2017-01-23	NaN	NaN	845.539978	828.700012	right_only
2017-01-24	NaN	NaN	851.520020	842.280029	right_only

Now the new `_merge` column shows us which rows were in each of the original dataframes.

We can use this to select those rows that were **not** in both

In [7]:

```
# now filter where _merge contains the string 'both'
```

```
df_anti_join = df_merge[ df_merge['_merge'] != 'both' ].copy()

df_anti_join
```

Out[7]:

	High_x	Low_x	High_y	Low_y	_merge
Date					
2017-01-03	167.869995	166.009995	NaN	NaN	left_only
2017-01-04	169.869995	167.360001	NaN	NaN	left_only
2017-01-05	169.389999	167.259995	NaN	NaN	left_only
2017-01-06	169.919998	167.520004	NaN	NaN	left_only
2017-01-09	169.800003	167.619995	NaN	NaN	left_only
2017-01-18	NaN	NaN	829.809998	824.080017	right_only
2017-01-19	NaN	NaN	833.000000	823.960022	right_only
2017-01-20	NaN	NaN	829.239990	824.599976	right_only
2017-01-23	NaN	NaN	845.539978	828.700012	right_only
2017-01-24	NaN	NaN	851.520020	842.280029	right_only

now drop the _merge column

In [8]:

```
df_anti_join.drop('_merge', inplace=True, axis=1)
df_anti_join
```

Out[8]:

	High_x	Low_x	High_y	Low_y
Date				
2017-01-03	167.869995	166.009995	NaN	NaN
2017-01-04	169.869995	167.360001	NaN	NaN
2017-01-05	169.389999	167.259995	NaN	NaN
2017-01-06	169.919998	167.520004	NaN	NaN
2017-01-09	169.800003	167.619995	NaN	NaN
2017-01-18	NaN	NaN	829.809998	824.080017
2017-01-19	NaN	NaN	833.000000	823.960022
2017-01-20	NaN	NaN	829.239990	824.599976
2017-01-23	NaN	NaN	845.539978	828.700012
2017-01-24	NaN	NaN	851.520020	842.280029