

First we need to read both the data given.

```
import pandas as pd
data1 = pd.read_csv(r"C:\Users\shaha\Downloads\Records Matching Task.xlsx - data1.csv")
data2 = pd.read_csv(r"C:\Users\shaha\Downloads\Records Matching Task.xlsx - data2.csv")
```

Since it's mentioned that primary key for both data1 and data2 is Order Id + Product ID. So a new column named pkey_ is created. Hence we combine the values in OrderID and ProductID.

```
data1['pkey1']=data1['Order ID']+data1['Product ID']
data2['pkey2']=data2['Order ID']+data2['Product ID']
```

Now we combine data on both left and right table. Since if no match is found the value will be none on that field.

```
data= pd.merge(data1,data2, how='outer', left_on=['pkey1'], right_on=['pkey2'])
data
```



	Order ID_x	Product ID_x	Qty_x	pkey1	Order ID_y	Product ID_y	Qty_y	pkey2
0	CA-2014-100006	TEC-PH-10002075	3.0	CA-2014-100006TEC-PH-10002075	NaN	NaN	NaN	NaN
1	CA-2014-100090	FUR-TA-10003715	3.0	CA-2014-100090FUR-TA-10003715	CA-2014-100090	FUR-TA-10003715	3.0	CA-2014-100090FUR-TA-10003715
2	CA-2014-100090	OFF-BI-10001597	6.0	CA-2014-100090OFF-BI-10001597	CA-2014-100090	OFF-BI-10001597	6.0	CA-2014-100090OFF-BI-10001597
3	CA-2014-100293	OFF-PA-10000176	6.0	CA-2014-100293OFF-PA-10000176	CA-2014-100293	OFF-PA-10000176	6.0	CA-2014-100293OFF-PA-10000176
4	CA-2014-100328	OFF-BI-10000343	1.0	CA-2014-100328OFF-BI-10000343	CA-2014-100328	OFF-BI-10000343	1.0	CA-2014-100328OFF-BI-10000343
...

Records present in data 1 which missing in data 2 can be achieved by selecting the rows which have null value for pkey2.

Since the matching values are merged.

```
Q1 = data[data['pkey2'].isna()]
Q1
```

	Order ID_x	Product ID_x	Qty_x	pkey1	Order ID_y	Product ID_y	Qty_y	pkey2
0	CA-2014-100006	TEC-PH-10002075	3.0	CA-2014-100006TEC-PH-10002075	NaN	NaN	NaN	NaN
10	CA-2014-100678	OFF-EN-10000056	3.0	CA-2014-100678OFF-EN-10000056	NaN	NaN	NaN	NaN
19	CA-2014-100895	OFF-AR-10004511	2.0	CA-2014-100895OFF-AR-10004511	NaN	NaN	NaN	NaN
35	CA-2014-101560	OFF-BI-10000309	3.0	CA-2014-101560OFF-BI-10000309	NaN	NaN	NaN	NaN
61	CA-2014-102673	OFF-LA-10001771	12.0	CA-2014-102673OFF-LA-10001771	NaN	NaN	NaN	NaN
...
9390	US-2017-160836	OFF-AP-10001626	2.0	US-2017-160836OFF-AP-10001626	NaN	NaN	NaN	NaN
9403	US-2017-162558	FUR-FU-10002364	2.0	US-2017-162558FUR-FU-10002364	NaN	NaN	NaN	NaN

The number of same is achieved here by counting the length of dataframe

```
len(Q1)

507
```

Records present in data 2 which missing in data 1 can be achieved by selecting the rows which have null value for pkey1.

Since the matching values are merged.

```
Q2 = data[data['pkey1'].isna()]
Q2
```

	Order ID_x	Product ID_x	Qty_x	pkey1	Order ID_y	Product ID_y	Qty_y	pkey2
9478	NaN	NaN	NaN	NaN	CA-2014- 100706	TEC-AC- 10001314	2.0	CA-2014- 100706TEC-AC- 10001314
9479	NaN	NaN	NaN	NaN	CA-2014- 100762	OFF-PA- 10001815	3.0	CA-2014- 100762OFF-PA- 10001815
9480	NaN	NaN	NaN	NaN	CA-2014- 101427	OFF-AR- 10002257	3.0	CA-2014- 101427OFF-AR- 10002257
9481	NaN	NaN	NaN	NaN	CA-2014- 102652	FUR-FU- 10001918	7.0	CA-2014- 102652FUR-FU- 10001918

The number of same is achieved here by counting the length of dataframe

```
len(Q2)
```

```
508
```

The task is to find sum of quantity of data point missing in data1 but present in data2. Since we have calculated the same previously records missing in data1 but present in data2. So here we can use sum operator to calculate the sum of quantity in the dataframe which we have .

```
Q2['Qty_y'].sum()
```

```
1956.0
```

Number of common data in both the table can be achieved by using inner join

```
dat= pd.merge(data1,data2, how='inner', left_on=['pkey1'], right_on=['pkey2'])
len(dat)
```

```
8971
```

[Colab paid products](#) - [Cancel contracts here](#)

