In [147]:
```python
import pandas as pd
import numpy as np
```

In [148]:
```python
filePath = 'G:/PRCTICE/Python/'
inputFileName = 'SalesOrderHeader.xlsx'
outputFileName = 'SalesOrderHeader_Analysis_Result.xlsx'
inputFile = filePath + inputFileName
outputFile = filePath + outputFileName
```

In [149]:
```python
data = pd.read_excel(inputFile)
```

Chage index name and serial:

In [150]:
```python
data.index.name = "Index"
data.index = data.index + 1
```

Substring:

In [151]:
```python
SalesOrderCode  = data['SalesOrderNumber'].astype(str).str[:3]
```

Insert new column in a specific position:

In [152]:
```python
data.insert(loc=7, column='SalesOrderCode', value=SalesOrderCode)
```

In [153]:
```python
data.head(3)
```

Out[153]:

	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderCode	SalesOrderNumber	PurchaseOrderNumbe
Index										
1	43659	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43659	PO52214578
2	43660	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43660	PO1885012750
3	43661	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43661	PO1847318962

3 rows × 27 columns

Replace:

In [154]:
```python
data['DueDate']=data['DueDate'].replace('1970-01-01', np.nan)
```

Apply IN, Between and & with where clause:

In [155]:
```python
condition1 = data['SalesOrderCode'].isin(['SO4','SO5'])
condition2 = data['OrderDate'].between('2011-05-31','2013-05-31')
data[condition1 & condition2].head(2)
```

Out[155]:

	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderCode	SalesOrderNumber	PurchaseOrderNumbe
Index										
1	43659	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43659	PO52214578
2	43660	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43660	PO1885012750

2 rows × 27 columns

In [156]:
```python
data[condition1 & condition2].tail(2)
```

Out[156]:

	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderCode	SalesOrderNumber	PurchaseOrderNumbe
Index										
7543	51201	8	2013-05-31	2013-06-12	2013-06-07	5	1	SO5	SO51201	NaN
7544	51202	8	2013-05-31	2013-06-12	2013-06-07	5	1	SO5	SO51202	NaN

2 rows × 27 columns

In [157]:
```python
data.where(condition1 & condition2).head(2)
```

Out[157]:

	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderCode	SalesOrderNumber	PurchaseOrderNumbe
Index										
1	43659.0	8.0	2011-05-31	2011-06-12	2011-06-07	5.0	0.0	SO4	SO43659	PO52214578
2	43660.0	8.0	2011-05-31	2011-06-12	2011-06-07	5.0	0.0	SO4	SO43660	PO1885012750

2 rows × 27 columns

In [158]:
```python
data.query("RevisionNumber in [8,7]").head(3)
```

Out[158]:

	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderCode	SalesOrderNumber	PurchaseOrderNumbe
Index										
1	43659	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43659	PO52214578
2	43660	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43660	PO1885012750
3	43661	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43661	PO1847318962

3 rows × 27 columns

group by with and without meging index column:

In [159]:
```python
data_groupedby = data.groupby(['SalesOrderCode'], as_index=False).agg('count')
```

In [160]:
```python
data_groupedby.head(3)
```

Out[160]:

	SalesOrderCode	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderNumber	PurchaseOrderNumber	..
0	SO4	6341	6341	6341	6341	6341	6341	6341	6341	1758	..
1	SO5	10000	10000	10000	10000	10000	10000	10000	10000	1147	..
2	SO6	10000	10000	10000	10000	10000	10000	10000	10000	720	..

3 rows × 27 columns

In [161]:
```python
data_groupedby = data.groupby(['SalesOrderCode']).agg('count')
```

In [162]:
```python
data_groupedby.head(3)
```

Out[162]:

	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderNumber	PurchaseOrderNumber	Acco
SalesOrderCode										
SO4	6341	6341	6341	6341	6341	6341	6341	6341	1758	
SO5	10000	10000	10000	10000	10000	10000	10000	10000	1147	
SO6	10000	10000	10000	10000	10000	10000	10000	10000	720	

3 rows × 26 columns

Export every column with distinct value to excel:

In [163]:
```python
writer = pd.ExcelWriter(outputFile, engine='xlsxwriter')
```

In [164]:
```python
data_groupedby.to_excel(writer, 'Tally')
```

In [165]:
```python
total = len(data.index)
```

In [166]:
```python
data.insert(loc=0, column='SL#', value=data.index)
```

In [167]:
```python
for column in data:
    if(data[column].name == 'SalesOrderCode' or  data[column].name == 'SL#'):
        continue
    else:
        col_names =  [data[column].name]
        my_df  = pd.DataFrame(columns = col_names)
        my_df[data[column].name] = data[column].unique().tolist()
        if((my_df[column].nunique()/total) <= 0.02):
            grouped_data = data[['SalesOrderCode',data[column].name,'SL#']]
            grouped_data = grouped_data.groupby(['SalesOrderCode',data[column].name], as_index=False).agg('count')
            #Rename index column here.
            grouped_data = grouped_data.rename(columns={'SL#': 'Count'})
            if(len(grouped_data) > 0):
                grouped_data.to_excel(writer, sheet_name = data[column].name)
```

In [168]:
```python
writer.save()
writer.close()
```

Set, Reset index:

In [169]:
```python
data.set_index("SL#",inplace= True)
```

In [170]:
```python
data.head(3)
```

Out[170]:

	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderCode	SalesOrderNumber	PurchaseOrderNumber
SL#										
1	43659	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43659	PO522145787
2	43660	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43660	PO18850127500
3	43661	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43661	PO18473189620

3 rows × 27 columns

In [171]:
```python
data.reset_index(drop=False, inplace=True)
```

In [172]:
```python
data.head(2)
```

Out[172]:

	SL#	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderCode	SalesOrderNumber	...	CreditCardID
0	1	43659	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43659	...	16281.0
1	2	43660	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43660	...	5618.0

2 rows × 28 columns

Show data by loc & iloc:

In [173]:
```python
data.iloc[0:1]
```

Out[173]:

	SL#	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderCode	SalesOrderNumber	...	CreditCardID
0	1	43659	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43659	...	16281.0

1 rows × 28 columns

In [174]:
```python
data.iloc[0,6:8]
```

Out[174]:
```
Status             5
OnlineOrderFlag    0
Name: 0, dtype: object
```

Rename column:

In [175]:
```python
data.rename(columns= {'ShipDate':'ShipingDate'},inplace= False).head(3)
```

Out[175]:

	SL#	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipingDate	Status	OnlineOrderFlag	SalesOrderCode	SalesOrderNumber	...	CreditCardID
0	1	43659	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43659	...	16281.0
1	2	43660	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43660	...	5618.0
2	3	43661	8	2011-05-31	2011-06-12	2011-06-07	5	0	SO4	SO43661	...	1346.0

3 rows × 28 columns

In [176]:
```python
import pandas as pd
import numpy as np
```

Change Style/Design of data table:

In [177]:
```
%%HTML
<style type="text/css">
    table.dataframe td, table.dataframe th {
        border-style: inset;
    }
</style>
```

In [178]:
```python
filePath = 'G:/PRCTICE/Python/'
inputFileName = 'BusinessEntity.xlsx'
outputFileName = 'SalesOrderHeader_Analysis_Result.xlsx'
inputFile = filePath + inputFileName
outputFile = filePath + outputFileName
```

In [179]:
```python
data = pd.read_excel(inputFile)
```

In [181]:
```python
data.head(3)
```

Out[181]:

	BusinessEntityID	Title	FirstName	MiddleName	LastName	Suffix	PhoneNumber	PhoneNumberType	EmailAddress	EmailPromotion	AddressTy
0	1699	Mr.	David	R.	Robinett	NaN	238-555-0100	Home	david22@adventure-works.com	1	Ho
1	1700	Ms.	Rebecca	A.	Robinson	NaN	648-555-0100	Cell	rebecca3@adventure-works.com	0	Ho
2	1701	Ms.	Dorothy	B.	Robinson	NaN	423-555-0100	Cell	dorothy3@adventure-works.com	2	Ho

In [183]:
```python
data.set_index(keys=['CountryRegionName','StateProvinceName','City']).head()
```

Out[183]:

			BusinessEntityID	Title	FirstName	MiddleName	LastName	Suffix	PhoneNumber	PhoneNumberType
CountryRegionName	StateProvinceName	City								
Germany	Nordrhein-Westfalen	Solingen	1699	Mr.	David	R.	Robinett	NaN	238-555-0100	Home
Australia	Victoria	Seaford	1700	Ms.	Rebecca	A.	Robinson	NaN	648-555-0100	Cell
		Geelong	1701	Ms.	Dorothy	B.	Robinson	NaN	423-555-0100	Cell
United Kingdom	England	Lancaster	1702	Ms.	Carol Ann	F.	Rockne	NaN	439-555-0100	Cell
Australia	Queensland	East Brisbane	1703	Mr.	Scott	M.	Rodgers	NaN	989-555-0100	Cell

In [184]:
```python
data_count = data.count()
```

In [185]:
```python
data_count.head()
```

Out[185]:
```
BusinessEntityID    18508
Title                 101
FirstName           18508
MiddleName          10666
LastName            18508
dtype: int64
```

In [186]:
```python
col_names =  ["Column","Value"]
my_df  = pd.DataFrame([data_count],columns = col_names)
```

In [187]:
```python
my_df
```

Out[187]:

	Column	Value
0	NaN	NaN

In [188]:
```python
# for index,row in data.iterrows():
#     print(len(str(row['DFIAccountNumber'])))
```

Insert data by index & column in dataframe:

In [189]:
```python
import pandas as pd
```

In [190]:
```python
myEmptyFrame1 = pd.DataFrame(columns=["X"])
```

In [191]:
```python
myEmptyFrame1.at[0, "X"] = 10
```

In [192]:
```python
myEmptyFrame1["Y"] = 0
```

In [193]:
```python
myEmptyFrame1
```

Out[193]:

	X	Y
0	10	0

In [194]:
```python
myEmptyFrame1.at[0, "Y"] = 0
```

In [195]:
```python
myEmptyFrame1
```

Out[195]:

	X	Y
0	10	0

In [196]:
```python
myEmptyFrame1.at[1, "X"] = 1
myEmptyFrame1.at[1, "Y"] = 1
```

In [197]:
```python
myEmptyFrame1.head()
```

Out[197]:

	X	Y
0	10.0	0.0
1	1.0	1.0

In [198]:
```python
myDataFrame1 = myEmptyFrame1
```

Joining:

In [199]:
```python
myDataFrame2 = pd.DataFrame(columns=["X","Y"])
```

In [200]:
```python
myDataFrame2.at[0,"X"] = 10
myDataFrame2.at[0,"Y"] = 0

myDataFrame2.at[1,"X"] = 10
myDataFrame2.at[1,"Y"] = 10
```

In [201]:
```python
myDataFrame2.head()
```

Out[201]:

	X	Y
0	10	0
1	10	10

In [202]:
```python
len(myDataFrame1)
len(myDataFrame2)
```

Out[202]: 2

In [203]:
```python
# concat as SQL union
myDataFrame3= pd.concat([myDataFrame1,myDataFrame2],ignore_index=True) # works.
```

In [204]:
```python
myDataFrame3.head()
```

Out[204]:

	X	Y
0	10	0
1	1	1
2	10	0
3	10	10

In [205]:
```python
myDataFrame3= pd.concat([myDataFrame1,myDataFrame2],keys=["A","B"]) # works.
```

In [206]:
```python
myDataFrame3.head()
```

Out[206]:

		X	Y
A	0	10	0
	1	1	1
B	0	10	0
	1	10	10

In [207]:
```python
myDataFrame3.at[("A",0),"X"]
```

Out[207]: 10.0

In [208]:
```python
#Append
appendDataFrame = myDataFrame1.append(myDataFrame2,ignore_index=True)
```

In [209]:
```python
appendDataFrame.head()
```

Out[209]:

	X	Y
0	10	0
1	1	1
2	10	0
3	10	10

In [210]:
```python
#Inner join single column in ON clause:
innerJoin = myDataFrame1.merge(myDataFrame2,how="inner",on="X") #suffixes=('_x', '_y'),
```

In [211]:
```python
innerJoin.head()
```

Out[211]:

	X	Y_x	Y_y
0	10	0.0	0
1	10	0.0	10

In [212]:
```python
#Inner join multy column in ON clause:
innerJoin = myDataFrame1.merge(myDataFrame2,how="inner",on=["X","Y"]) #suffixes=('_x', '_y'),
```

In [213]:
```python
innerJoin.head()
```

Out[213]:

	X	Y
0	10	0

In [214]:
```python
#Outer join  in ON clause:
outerJoin = myDataFrame1.merge(myDataFrame2,how="outer",on=["X"],indicator=True) #suffixes=('_x', '_y')
```

In [215]:
```python
outerJoin.head()
```

Out[215]:

	X	Y_x	Y_y	_merge
0	10	0.0	0	both
1	10	0.0	10	both
2	1	1.0	NaN	left_only

In [216]:
```python
#Left join  in ON clause:
leftJoin = myDataFrame1.merge(myDataFrame2,how="left",on=["X"],indicator=True) #suffixes=('_x', '_y')
```

In [217]:
```python
leftJoin.head()
```

Out[217]:

	X	Y_x	Y_y	_merge
0	10	0.0	0	both
1	10	0.0	10	both
2	1	1.0	NaN	left_only

In [218]:
```python
#Left Right ON: in ON clause:
leftRightOnJoin = myDataFrame1.merge(myDataFrame2,how="left",left_on="X",right_on="Y",indicator=True) #suffixes=('_x', '_y')
```

In [219]:
```python
leftRightOnJoin.head()
```

Out[219]:

	X_x	Y_x	X_y	Y_y	_merge
0	10	0.0	10	10	both
1	1	1.0	NaN	NaN	left_only

In [220]:
```python
#Left Right ON: in ON clause:
myDataFrame1.merge(myDataFrame2,how="left",left_on="X",right_index=True,indicator=True).head()
```

Out[220]:

	X_x	Y_x	X_y	Y_y	_merge
0	10.0	0.0	NaN	NaN	left_only
1	1.0	1.0	10	10	both

In [221]:
```python
#Left Right ON: in ON clause:
myDataFrame1.merge(myDataFrame2,how="left",left_on="X",left_index=True,right_index=True,indicator=True).head()
```

Out[221]:

	X_x	Y_x	X_y	Y_y	_merge
0	10.0	0.0	10	0	both
1	1.0	1.0	10	10	both

In [222]:
```python
#Join
myDataFrame1.join(leftRightOnJoin).head()
```

Out[222]:

	X	Y	X_x	Y_x	X_y	Y_y	_merge
0	10.0	0.0	10	0.0	10	10	both
1	1.0	1.0	1	1.0	NaN	NaN	left_only

In [225]:
```python
#Trim both Ltrim & Rtrim .str.strip()
' Khairul Alam     '.strip()
```

Out[225]: 'Khairul Alam'

In [ ]: