

```
In [1]: ▶ import pandas as pd
sales = pd.read_csv('500000 Sales Records.csv')
```

```
In [68]: ▶ sales.head()
```

Out[68]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold
0	Sub-Saharan Africa	South Africa	Fruits	Offline	M	7/27/2012	443368995	7/28/2012	1593
1	Middle East and North Africa	Morocco	Clothes	Online	M	9/14/2013	667593514	10/19/2013	4611
2	Australia and Oceania	Papua New Guinea	Meat	Offline	M	5/15/2015	940995585	6/4/2015	360
3	Sub-Saharan Africa	Djibouti	Clothes	Offline	H	5/17/2017	880811536	7/2/2017	562
4	Europe	Slovakia	Beverages	Offline	L	10/26/2016	174590194	12/4/2016	3973

```
In [23]: ▶ sales.shape
```

Out[23]: (500000, 14)

```
In [24]: ▶ Order_table = sales[["Order ID", 'Order Priority', 'Sales Channel']]
```

In [25]: `Order_table`

Out[25]:

	Order ID	Order Priority	Sales Channel
0	443368995	M	Offline
1	667593514	M	Online
2	940995585	M	Offline
3	880811536	H	Offline
4	174590194	L	Offline
...
499995	134026181	L	Offline
499996	987753570	H	Online
499997	766976869	C	Online
499998	537369906	H	Offline
499999	984919011	H	Online

In [2]: `sales['year'] = pd.DatetimeIndex(sales['Order Date']).year
sales['month'] = pd.DatetimeIndex(sales['Order Date']).month
sales['Day'] = pd.DatetimeIndex(sales['Order Date']).day
sales['DayofWeek'] = pd.DatetimeIndex(sales['Order Date']).dayofweek
sales['Dayofyear'] = pd.DatetimeIndex(sales['Order Date']).dayofyear
sales['Week'] = pd.DatetimeIndex(sales['Order Date']).week`

In [27]: `sales.head()`

Out[27]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold
0	Sub-Saharan Africa	South Africa	Fruits	Offline	M	7/27/2012	443368995	7/28/2012	1593
1	Middle East and North Africa	Morocco	Clothes	Online	M	9/14/2013	667593514	10/19/2013	4611
2	Australia and Oceania	Papua New Guinea	Meat	Offline	M	5/15/2015	940995585	6/4/2015	360
3	Sub-Saharan Africa	Djibouti	Clothes	Offline	H	5/17/2017	880811536	7/2/2017	562
4	Europe	Slovakia	Beverages	Offline	L	10/26/2016	174590194	12/4/2016	3973

```
In [35]: sales.columns
```

```
Out[35]: Index(['Region', 'Country', 'Item_Type', 'Sales_Channel', 'Order_Priority',
               'Order_Date', 'Order_ID', 'Ship_Date', 'Unit_Sold', 'Unit_Price',
               'Unit_Cost', 'Total_Revenue', 'Total_Cost', 'Total_profit', 'year',
               'month', 'Day', 'DayofWeek', 'Dayofyear', 'Week'],
              dtype='object')
```

```
In [3]: sales = sales.rename(columns = {"Item Type":"Item_Type","Sales Channel":"Sales_Channel"})
```

```
In [260]: import pyodbc as db # SQL Server
conn = db.connect('Driver={SQL Server};'
                  'Server=DESKTOP-UVT33PM;'
                  'Database=SalesProject;'
                  'Trusted_Connection=yes;')
cursor = conn.cursor()
```

INSERTION OF VALUES INTO THE TABLE (ETL)

```
In [6]: for i,row in sales.iterrows():
        sales.at[i,"Item_Type"] = row.Item_Type.replace(" ", "_")
```

```
In [7]: sales.head()
```

Out[7]:

	Region	Country	Item_Type	Sales_Channel	Order_Priority	Order_Date	Order_ID	Ship_Date
0	Sub-Saharan Africa	South Africa	Fruits	Offline	M	7/27/2012	443368995	7/28/2012
1	Middle East and North Africa	Morocco	Clothes	Online	M	9/14/2013	667593514	10/19/2013
2	Australia and Oceania	Papua New Guinea	Meat	Offline	M	5/15/2015	940995585	6/4/2015
3	Sub-Saharan Africa	Djibouti	Clothes	Offline	H	5/17/2017	880811536	7/2/2017
4	Europe	Slovakia	Beverages	Offline	L	10/26/2016	174590194	12/4/2016

```
In [168]: itemdf = sales[['Item_Type','Unit_Price','Unit_Cost']]
```

```
In [169]: itemdf['Item_Type'].unique()
```

```
Out[169]: array(['Fruits', 'Clothes', 'Meat', 'Beverages', 'Office_Supplies',  
                'Cosmetics', 'Snacks', 'Personal_Care', 'Household', 'Vegetables',  
                'Baby_Food', 'Cereal'], dtype=object)
```

```
In [170]: values = {'Id':[1,2,3,4,5,6,7,8,9,10,11,12],  
                  'Item_Type':['Fruits', 'Clothes', 'Meat', 'Beverages', 'Office_Suppli
```

```
In [171]: item = pd.DataFrame(data=values)
```

```
In [172]: item
```

```
Out[172]:
```

	Id	Item_Type
0	1	Fruits
1	2	Clothes
2	3	Meat
3	4	Beverages
4	5	Office_Supplies
5	6	Cosmetics
6	7	Snacks
7	8	Personal_Care
8	9	Household
9	10	Vegetables
10	11	Baby_Food
11	12	Cereal

```
In [173]: ▶ result = pd.merge(itemdf,item, how='inner', on=None, left_on=None, right_on=None,
left_index=False, right_index=False, sort=False,
suffixes=('_x', '_y'), copy=True, indicator=False,
validate=None)
result
```

Out[173]:

	Item_Type	Unit_Price	Unit_Cost	Id
0	Fruits	9.33	6.92	1
1	Fruits	9.33	6.92	1
2	Fruits	9.33	6.92	1
3	Fruits	9.33	6.92	1
4	Fruits	9.33	6.92	1
...
499995	Cereal	205.70	117.11	12
499996	Cereal	205.70	117.11	12
499997	Cereal	205.70	117.11	12
499998	Cereal	205.70	117.11	12
499999	Cereal	205.70	117.11	12

500000 rows × 4 columns

```
In [174]: ▶ itemdf = result.drop_duplicates()
itemdf
```

Out[174]:

	Item_Type	Unit_Price	Unit_Cost	Id
0	Fruits	9.33	6.92	1
41684	Clothes	109.28	35.84	2
83373	Meat	421.89	364.69	3
125046	Beverages	47.45	31.79	4
166662	Office_Supplies	651.21	524.96	5
208240	Cosmetics	437.20	263.33	6
249957	Snacks	152.58	97.44	7
291663	Personal_Care	81.73	56.67	8
333452	Household	668.27	502.54	9
375069	Vegetables	154.06	90.93	10
416724	Baby_Food	255.28	159.42	11
458381	Cereal	205.70	117.11	12

```
In [52]: ➤ for i,row in itemdf.iterrows():  
        sql = "INSERT INTO SalesProject.dbo.ITEMTYPE_DIMENTION(ID,ITEMNAME,UNITPR  
        values("+str(row.Id)+", '"+str(row.Item_Type)+"', '"+str(row.Unit_Price)+"'  
        cursor.execute(sql)
```

```
In [53]: ➤ sql
```

```
Out[53]: "INSERT INTO SalesProject.dbo.ITEMTYPE_DIMENTION(ID,ITEMNAME,UNITPRICE,UNIT  
        COST)    values(12,'Cereal','205.7',117.11);"
```

```
In [54]: ➤ conn.commit()
```

```
In [78]: ➤ sql
```

```
Out[78]: 'INSERT INTO SalesForecast.dbo.SALES_FACT_TABLE(LOCATIONID,ITEMTYPEID,ORDER  
        ID,DATEID,TOTALREVENUE,TOTALCOST,TOTALPROFIT,UNITSOLD)values(500000,500000,  
        500000,500000,5531270.79,4159523.58,1371747.21,8277);'
```

Insertion of rows in location dimention

```
In [55]: ➤ for i,row in sales.iterrows():  
        sales.at[i,"Region"] = row.Region.replace(" ", "_")  
        sales.at[i,"Country"] = row.Country.replace(" ", "_")
```

```
In [175]: ➤ location = sales[["Region","Country"]]
```

```
In [59]: ➤ location['Country'].nunique()
```

```
Out[59]: 185
```

```
In [ ]: ➤
```

```
In [176]: values = {'Id' :[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,
    'Country' :['South_Africa', 'Morocco', 'Papua_New_Guinea', 'Djibou
    'Slovakia', 'Sri_Lanka', 'Seychelles_', 'Tanzania', 'Ghana',
    'Taiwan', 'Algeria', 'Singapore', 'Vietnam', 'Uganda', 'Zimbabwe',
    'Ethiopia', 'France', 'The_Bahamas', 'Haiti', 'Nicaragua',
    'Turkmenistan', 'United_Kingdom', 'Dominican_Republic', 'China',
    'Kuwait', 'United_Arab_Emirates', 'Estonia', 'Malaysia', 'Vanuatu',
    'India', 'Samoa_', 'Kazakhstan', 'Czech_Republic', 'Belgium',
    'Finland', 'Oman', 'Dominica', 'Serbia', 'Sao_Tome_and_Principe',
    'Brunei', 'Israel', 'Solomon_Islands', 'Togo', 'Mauritius_',
    'Canada', 'Lebanon', 'South_Korea', 'Indonesia',
    'Antigua_and_Barbuda_', 'Tunisia_', 'Thailand', 'Nepal',
    'Montenegro', 'Greece', 'Monaco', 'Albania', 'Saint_Lucia',
    'Italy', 'Switzerland', 'Netherlands', 'Sweden', 'Burundi',
    'Iceland', 'Rwanda', 'Japan', 'Romania', 'Belize', 'Egypt',
    'Tonga', 'East_Timor', 'The_Gambia', 'Mali', 'Moldova_',
    'Pakistan', 'Madagascar', 'United_States_of_America',
    'Democratic_Republic_of_the_Congo', 'New_Zealand', 'Liberia',
    'Malawi', 'Equatorial_Guinea', 'El_Salvador', 'Greenland',
    'Myanmar', 'Costa_Rica', 'Armenia', 'Somalia', 'Kenya', 'Zambia',
    'Marshall_Islands', 'Syria', 'Niger', 'Mongolia', 'Sierra_Leone',
    'Cape_Verde', 'Denmark', 'Saint_Kitts_and_Nevis_', 'Saudi_Arabia',
    'Bulgaria', 'Sudan', 'Yemen', 'Cambodia', 'Trinidad_and_Tobago',
    'Central_African_Republic', 'Latvia', 'Grenada', 'Portugal',
    'Bahrain', 'Cameroon', 'Bhutan', 'Georgia', 'Kiribati', 'Croatia',
    'Turkey', 'Iraq', 'Comoros', 'Uzbekistan', 'Tuvalu', 'Gabon',
    'Hungary', 'Jordan', 'Andorra', 'Luxembourg', 'Guinea-Bissau',
    'Bangladesh', 'Laos', 'Nigeria', 'Germany', 'Jamaica', 'Botswana',
    'Vatican_City', 'Panama', 'Namibia', 'Ireland',
    'Federated_States_of_Micronesia', 'Poland', 'Kosovo', 'Palau',
    'Angola', 'Benin', 'Mexico', 'Malta', 'Iran', 'Mauritania',
    'Saint_Vincent_and_the_Grenadines', 'Guatemala', 'Lesotho',
    'Mozambique', 'North_Korea', 'Kyrgyzstan', 'Belarus',
    'Liechtenstein', 'Burkina_Faso', 'Australia', 'Macedonia',
    'Bosnia_and_Herzegovina', 'Barbados', 'Senegal', 'Tajikistan',
    'South_Sudan', 'Cuba', 'Russia', 'Swaziland', 'Azerbaijan',
    "Cote_d'Ivoire", 'Austria', 'Chad', 'Libya', 'Norway',
    'Afghanistan', 'Lithuania', 'Eritrea', 'Cyprus', 'Maldives',
    'Slovenia', 'Guinea', 'Philippines', 'Qatar', 'Ukraine', 'Nauru',
    'Honduras', 'Republic_of_the_Congo', 'Spain', 'San_Marino', 'Fiji']}]}
```

```
In [177]: country = pd.DataFrame(data=values)
```

```
In [178]: result = pd.merge(location,country, how='inner', on=None, left_on=None, right_on=None,
left_index=False, right_index=False, sort=False,
suffixes=('_x', '_y'), copy=True, indicator=False,
validate=None)
result
```

Out[178]:

	Region	Country	Id
0	Sub-Saharan_Africa	South_Africa	1
1	Sub-Saharan_Africa	South_Africa	1
2	Sub-Saharan_Africa	South_Africa	1
3	Sub-Saharan_Africa	South_Africa	1
4	Sub-Saharan_Africa	South_Africa	1
...
499995	Australia_and_Oceania	Fiji	185
499996	Australia_and_Oceania	Fiji	185
499997	Australia_and_Oceania	Fiji	185
499998	Australia_and_Oceania	Fiji	185
499999	Australia_and_Oceania	Fiji	185

500000 rows × 3 columns

```
In [179]: countrydf = result.drop_duplicates()
countrydf
```

Out[179]:

	Region	Country	Id
0	Sub-Saharan_Africa	South_Africa	1
2717	Middle_East_and_North_Africa	Morocco	2
5493	Australia_and_Oceania	Papua_New_Guinea	3
8181	Sub-Saharan_Africa	Djibouti	4
10918	Europe	Slovakia	5
...
486438	Central_America_and_the_Caribbean	Honduras	181
489152	Sub-Saharan_Africa	Republic_of_the_Congo	182
491879	Europe	Spain	183
494646	Europe	San_Marino	184
497348	Australia_and_Oceania	Fiji	185

185 rows × 3 columns


```
In [162]: ➤ for i,row in countrydf.iterrows():
            countrydf.at[i,"Country"] = row.Country.replace("'", "_")
```

```
In [163]: ➤ countrydf
```

Out[163]:

	Region	Country	Id
0	Sub-Saharan_Africa	South_Africa	1
2717	Middle_East_and_North_Africa	Morocco	2
5493	Australia_and_Oceania	Papua_New_Guinea	3
8181	Sub-Saharan_Africa	Djibouti	4
10918	Europe	Slovakia	5
...
486438	Central_America_and_the_Caribbean	Honduras	181
489152	Sub-Saharan_Africa	Republic_of_the_Congo	182
491879	Europe	Spain	183
494646	Europe	San_Marino	184
497348	Australia_and_Oceania	Fiji	185

185 rows × 3 columns

```
In [164]: ➤ for i,row in countrydf.iterrows():
            sql = "INSERT INTO SalesProject.dbo.LOCATION_DIMENTION(LOCATIONID,CONTINENT,COUNTRY)VALUES(1, 'Sub-Saharan_Africa', 'South_Africa');"
            cursor.execute(sql)
```

```
In [165]: ➤ conn.commit()
```

```
In [89]: ➤ sql
```

Out[89]: "INSERT INTO SalesProject.dbo.LOCATION_DIMENTION(LOCATIONID,CONTINENT,COUNTRY)VALUES(1, 'Sub-Saharan_Africa', 'South_Africa');"

Insertion of rows in DATE_TIME_DIM table

```
In [186]: ▶ date_time = sales[['Order_Date', 'year', 'month', 'Day', 'DayofWeek', 'Dayofyear'],  
date_time
```

Out[186]:

	Order_Date	year	month	Day	DayofWeek	Dayofyear	Week
0	7/27/2012	2012	7	27	4	209	30
1	9/14/2013	2013	9	14	5	257	37
2	5/15/2015	2015	5	15	4	135	20
3	5/17/2017	2017	5	17	2	137	20
4	10/26/2016	2016	10	26	2	300	43
...
499995	1/16/2015	2015	1	16	4	16	3
499996	1/22/2011	2011	1	22	5	22	3
499997	1/25/2017	2017	1	25	2	25	4
499998	3/13/2016	2016	3	13	6	73	10
499999	5/13/2015	2015	5	13	2	133	20

500000 rows × 7 columns

```
In [187]: ▶ date = date_time[['Order_Date']]
```

```
In [188]: ▶ date = date.drop_duplicates()  
date
```

Out[188]:

	Order_Date
0	7/27/2012
1	9/14/2013
2	5/15/2015
3	5/17/2017
4	10/26/2016
...	...
17327	3/6/2015
17780	1/19/2017
18225	4/6/2010
20777	6/7/2017
21851	4/27/2017

2766 rows × 1 columns

```
In [120]: ▶ import numpy as np
```

```
In [189]: ▶ date['id'] = np.arange(len(date))  
date
```

Out[189]:

	Order_Date	id
0	7/27/2012	0
1	9/14/2013	1
2	5/15/2015	2
3	5/17/2017	3
4	10/26/2016	4
...
17327	3/6/2015	2761
17780	1/19/2017	2762
18225	4/6/2010	2763
20777	6/7/2017	2764
21851	4/27/2017	2765

2766 rows × 2 columns

```
In [190]: result = pd.merge(date,date_time, how='inner', on=None, left_on=None, right_on=None,
                             left_index=False, right_index=False, sort=False,
                             suffixes=('_x', '_y'), copy=True, indicator=False,
                             validate=None)
result
```

Out[190]:

	Order_Date	id	year	month	Day	DayofWeek	Dayofyear	Week
0	7/27/2012	0	2012	7	27	4	209	30
1	7/27/2012	0	2012	7	27	4	209	30
2	7/27/2012	0	2012	7	27	4	209	30
3	7/27/2012	0	2012	7	27	4	209	30
4	7/27/2012	0	2012	7	27	4	209	30
...
499995	4/27/2017	2765	2017	4	27	3	117	17
499996	4/27/2017	2765	2017	4	27	3	117	17
499997	4/27/2017	2765	2017	4	27	3	117	17
499998	4/27/2017	2765	2017	4	27	3	117	17
499999	4/27/2017	2765	2017	4	27	3	117	17

500000 rows × 8 columns

```
In [191]: datedf = result.drop_duplicates()
datedf
```

Out[191]:

	Order_Date	id	year	month	Day	DayofWeek	Dayofyear	Week
0	7/27/2012	0	2012	7	27	4	209	30
179	9/14/2013	1	2013	9	14	5	257	37
368	5/15/2015	2	2015	5	15	4	135	20
545	5/17/2017	3	2017	5	17	2	137	20
720	10/26/2016	4	2016	10	26	2	300	43
...
499113	3/6/2015	2761	2015	3	6	4	65	10
499293	1/19/2017	2762	2017	1	19	3	19	3
499468	4/6/2010	2763	2010	4	6	1	96	14
499656	6/7/2017	2764	2017	6	7	2	158	23
499823	4/27/2017	2765	2017	4	27	3	117	17

2766 rows × 8 columns

```
In [133]: ➤ for i,row in result.iterrows():
            sql = "INSERT INTO SalesProject.dbo.DATE_TIME_DIM(DATEID, DAYOFWEEK_, DAYOF
            cursor.execute(sql)
```

```
In [134]: ➤ conn.commit()
```

```
In [37]: ➤ sql
```

```
Out[37]: 'INSERT INTO SalesForecast.dbo.DATE_TIME_DIM(DATEID, DAYOFWEEK_, DAYOFYEAR_, M
ONTH_, YEAR_, DATE_)values(1,4,209,7,2012,7/27/2012);'
```

Insertion of rows in order dimation

```
In [277]: ➤ order1 = sales[['Order_Priority', 'Sales_Channel']]
```

```
In [278]: ➤ order = sales[['Order_Priority', 'Sales_Channel']]
```

```
In [263]: ➤ sales.head()
```

Out[263]:

	Region	Country	Item_Type	Sales_Channel	Order_Priority
0	Sub-Saharan_Africa	South_Africa	Fruits	Offline	M
1	Middle_East_and_North_Africa	Morocco	Clothes	Online	M
2	Australia_and_Oceania	Papua_New_Guinea	Meat	Offline	M
3	Sub-Saharan_Africa	Djibouti	Clothes	Offline	H
4	Europe	Slovakia	Beverages	Offline	L

```
In [279]: order = order[['Order_Priority']]
order =order.drop_duplicates()
order
```

Out[279]:

Order_Priority	
0	M
3	H
4	L
12	C

```
In [280]: order['id'] = np.arange(len(order))
order
```

Out[280]:

Order_Priority	id	
0	M	0
3	H	1
4	L	2
12	C	3

```
In [253]: result = pd.merge(order,order1, how='inner', on=None, left_on=None, right_on=
left_index=False, right_index=False, sort=False,
suffixes=('_x', '_y'), copy=True, indicator=False,
validate=None)
result
```

Out[253]:

	Order_Priority	id	Sales_Channel
0	M	0	Offline
1	M	0	Online
2	M	0	Offline
3	M	0	Online
4	M	0	Offline
...
499995	C	3	Online
499996	C	3	Offline
499997	C	3	Offline
499998	C	3	Online
499999	C	3	Online

500000 rows × 3 columns

```
In [281]: orderdf = result.drop_duplicates()
orderdf['id'] = np.arange(len(orderdf))
orderdf
```

<ipython-input-281-235f2c0a0c02>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
orderdf['id'] = np.arange(len(orderdf))
```

Out[281]:

	Order_Priority	id	Sales_Channel
0	M	0	Offline
1	M	1	Online
124833	H	2	Offline
124836	H	3	Online
249820	L	4	Offline

```
In [267]: orderdf['id'].unique()
```

Out[267]: array([0, 1, 2, 3, 4, 5, 6, 7])

```
In [282]: orderdf.count()
```

Out[282]: Order_Priority 8
id 8
Sales_Channel 8
dtype: int64

```
In [283]: orderdf
```

Out[283]:

	Order_Priority	id	Sales_Channel
0	M	0	Offline
1	M	1	Online
124833	H	2	Offline
124836	H	3	Online
249820	L	4	Offline
249821	L	5	Online
374958	C	6	Online
374959	C	7	Offline

```
In [284]: ▶ for i,row in orderdf.iterrows():
            sql = "INSERT INTO SalesProject.dbo.ORDER_DIM(ORDERID,ORDERPRIORITY,SALESCHANNEL)values("+str(row.id)+", '"+str(row.Order_Priority)+"', '"+str(row.Sales_Channel)+"');"
```

IntegrityError Traceback (most recent call last)

```
<ipython-input-284-c22089e2a448> in <module>
      1 for i,row in orderdf.iterrows():
      2     sql = "INSERT INTO SalesProject.dbo.ORDER_DIM(ORDERID,ORDERPRIORITY,SALESCHANNEL)values("+str(row.id)+", '"+str(row.Order_Priority)+"', '"+str(row.Sales_Channel)+"');"
```

```
-----> 3     cursor.execute(sql)

IntegrityError: ('23000', "[23000] [Microsoft][ODBC SQL Server Driver][SQL Server]Violation of PRIMARY KEY constraint 'PK__ORDER_DI__491E41929D1D980F'. Cannot insert duplicate key in object 'dbo.ORDER_DIM'. The duplicate key value is (0). (2627) (SQLExecDirectW); [23000] [Microsoft][ODBC SQL Server Driver][SQL Server]The statement has been terminated. (3621)")
```

```
In [269]: ▶ conn.commit()
```

```
In [194]: ▶ sales.head()
```

Out[194]:

	Region	Country	Item_Type	Sales_Channel	Order_Priority
0	Sub-Saharan_Africa	South_Africa	Fruits	Offline	M
1	Middle_East_and_North_Africa	Morocco	Clothes	Online	M
2	Australia_and_Oceania	Papua_New_Guinea	Meat	Offline	M
3	Sub-Saharan_Africa	Djibouti	Clothes	Offline	H
4	Europe	Slovakia	Beverages	Offline	L

Insertion of rows in Sales Fact table

```
In [316]: ▶ fact = sales[['Sales_Channel','Order_Priority','Country','Item_Type','Order_D
```

```
In [317]: ▶ x = fact
```


In [318]:

```
x
```

Out[318]:

	Sales_Channel	Order_Priority	Country	Item_Type	Order_Date	Total_R
0	Offline	M	South_Africa	Fruits	7/27/2012	1.
1	Online	M	Morocco	Clothes	9/14/2013	50
2	Offline	M	Papua_New_Guinea	Meat	5/15/2015	15
3	Offline	H	Djibouti	Clothes	5/17/2017	6
4	Offline	L	Slovakia	Beverages	10/26/2016	18
...
499995	Offline	L	Myanmar	Fruits	1/16/2015	
499996	Online	H	New_Zealand	Office_Supplies	1/22/2011	461.
499997	Online	C	Lithuania	Snacks	1/25/2017	66.
499998	Offline	H	Malaysia	Beverages	3/13/2016	40
499999	Online	H	Slovakia	Household	5/13/2015	553

500000 rows × 9 columns

```
In [319]: result = pd.merge(x,countrydf, how='inner', on=None, left_on=None, right_on=None,
                             left_index=False, right_index=False, sort=False,
                             suffixes=('_x', '_y'), copy=True, indicator=False,
                             validate=None)
result
```

Out[319]:

	Sales_Channel	Order_Priority	Country	Item_Type	Order_Date	Total_Revenue
0	Offline	M	South_Africa	Fruits	7/27/2012	14862.66
1	Offline	M	South_Africa	Meat	7/30/2013	1307015.22
2	Offline	M	South_Africa	Clothes	6/25/2010	571315.84
3	Offline	H	South_Africa	Clothes	7/17/2017	1051929.28
4	Offline	L	South_Africa	Beverages	9/14/2015	311699.05
...
499995	Offline	H	Fiji	Baby_Food	11/4/2012	1197773.76
499996	Online	C	Fiji	Household	6/19/2017	4946534.54
499997	Online	H	Fiji	Meat	7/10/2017	2079917.76
499998	Online	L	Fiji	Cosmetics	6/28/2012	562676.46

```
In [320]: result = result.rename(columns = {"Id":"CountryId"})
```

```
In [321]: x = result
```

```
In [333]: result = pd.merge(x,datedf, how='inner', on=None, left_on=None, right_on=None,
                             left_index=False, right_index=False, sort=False,
                             suffixes=('_x', '_y'), copy=True, indicator=False,
                             validate=None)
result
```

Out[333]:

	Sales_Channel	Order_Priority	Country	Item_Type	Order_Date
0	Offline	M	South_Africa	Fruits	7/27/201
1	Offline	M	Sweden	Fruits	7/27/201
2	Offline	M	Uzbekistan	Fruits	7/27/201
3	Offline	M	Guinea	Fruits	7/27/201
4	Offline	M	Republic_of_the_Congo	Fruits	7/27/201
...
499995	Online	M	Sao_Tome_and_Principe	Baby_Food	5/3/201
499996	Online	M	Democratic_Republic_of_the_Congo	Baby_Food	5/3/201
499997	Offline	C	Estonia	Baby_Food	5/3/201
499998	Offline	C	Sierra_Leone	Baby_Food	5/3/201
499999	Offline	C	Grenada	Baby_Food	5/3/201

500000 rows × 22 columns

```
In [334]: result = result.rename(columns = {"id":"DateId"})
```

In [335]: `result.head()`

Out[335]:

	Sales_Channel	Order_Priority	Country	Item_Type	Order_Date	Total_Revenue
0	Offline	M	South_Africa	Fruits	7/27/2012	14862.69
1	Offline	M	Sweden	Fruits	7/27/2012	75507.69
2	Offline	M	Uzbekistan	Fruits	7/27/2012	42852.69
3	Offline	M	Guinea	Fruits	7/27/2012	10197.69
4	Offline	M	Republic_of_the_Congo	Fruits	7/27/2012	27560.82

5 rows × 7 columns

In [339]: `result.column`

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-339-bd3d5f2a7262> in <module>
----> 1 result.column
```

```
~\Anaconda3\lib\site-packages\pandas\core\generic.py in __getattr__(self, name)
    5272         if self._info_axis._can_hold_identifiers_and_holds_name
    (name):
    5273             return self[name]
-> 5274         return object.__getattr__(self, name)
    5275
    5276     def __setattr__(self, name: str, value) -> None:
```

AttributeError: 'DataFrame' object has no attribute 'column'

In [340]: `orderdf`

Out[340]:

	Order_Priority	id	Sales_Channel
0	M	0	Offline
1	M	1	Online
124833	H	2	Offline
124836	H	3	Online
249820	L	4	Offline
249821	L	5	Online
374958	C	6	Online
374959	C	7	Offline

```
In [326]: ▶ result = pd.merge(x,orderdf, how='inner', on=None, left_on=None, right_on=None,
                             left_index=False, right_index=False, sort=False,
                             suffixes=('_x', '_y'), copy=True, indicator=False,
                             validate=None)
result
```

Out[326]:

	Sales_Channel	Order_Priority	Country	Item_Type	Order_Date	Total_Revenue	Total_Profit
0	Offline	M	South_Africa	Fruits	7/27/2012	14862.69	
1	Offline	M	South_Africa	Meat	7/30/2013	1307015.22	11
2	Offline	M	South_Africa	Clothes	6/25/2010	571315.84	1
3	Offline	M	South_Africa	Household	3/27/2015	1549049.86	11
4	Offline	M	South_Africa	Cereal	7/19/2011	8022.30	
...
499995	Offline	C	Fiji	Cosmetics	1/15/2013	1060647.20	6
499996	Offline	C	Fiji	Vegetables	7/19/2012	1302269.18	7
499997	Offline	C	Fiji	Fruits	5/1/2011	88998.87	
499998	Offline	C	Fiji	Vegetables	11/27/2013	1464032.18	8
499999	Offline	C	Fiji	Fruits	4/27/2012	69956.34	

500000 rows × 12 columns

```
In [327]: ▶ result = result.rename(columns = {"id":"OrderId"})
```

```
In [328]: ▶ x = result
```

```
In [329]: ▶ result = pd.merge(x,itemdf, how='inner', on=None, left_on=None, right_on=None,
                             left_index=False, right_index=False, sort=False,
                             suffixes=('_x', '_y'), copy=True, indicator=False,
                             validate=None)
result
```

Out[329]:

	Sales_Channel	Order_Priority	Country	Item_Type	Order_Date	Total_Revenue	T
0	Offline	M	South_Africa	Fruits	7/27/2012	14862.69	
1	Offline	M	South_Africa	Fruits	7/18/2015	40902.72	
2	Offline	M	South_Africa	Fruits	9/15/2015	25386.93	
3	Offline	M	South_Africa	Fruits	5/28/2015	60355.77	
4	Offline	M	South_Africa	Fruits	6/24/2012	4394.43	
...	
499995	Offline	C	Fiji	Baby_Food	2/4/2014	363263.44	1
499996	Offline	C	Fiji	Baby_Food	10/27/2016	1891114.24	1
499997	Offline	C	Fiji	Baby_Food	8/17/2012	378324.96	1
499998	Offline	C	Fiji	Baby_Food	6/13/2014	1817593.60	1
499999	Offline	C	Fiji	Baby_Food	10/14/2015	2159668.80	1

500000 rows × 15 columns

```
In [330]: ▶ result = result.rename(columns = {"Id":"ItemId"})
```

```
In [341]: ▶ x = result
```

```
In [342]: ▶ x.head()
```

Out[342]:

	Sales_Channel	Order_Priority	Country	Item_Type	Order_Date	Total_Revenue
0	Offline	M	South_Africa	Fruits	7/27/2012	14862.69
1	Offline	M	Sweden	Fruits	7/27/2012	75507.69
2	Offline	M	Uzbekistan	Fruits	7/27/2012	42852.69
3	Offline	M	Guinea	Fruits	7/27/2012	10197.69
4	Offline	M	Republic_of_the_Congo	Fruits	7/27/2012	27560.82

5 rows × 22 columns

In []: ▶

In [310]: ▶ x

Out[310]:

	Sales_Channel	Order_Priority	Country	Item_Type	Order_Date	Total_Revenue	Total_Cost
0	Offline	M	South_Africa	Fruits	7/27/2012	14862.69	
1	Offline	M	South_Africa	Meat	7/30/2013	1307015.22	11
2	Offline	M	South_Africa	Clothes	6/25/2010	571315.84	1
3	Offline	M	South_Africa	Household	3/27/2015	1549049.86	11
4	Offline	M	South_Africa	Cereal	7/19/2011	8022.30	
...
499995	Offline	C	Fiji	Cosmetics	1/15/2013	1060647.20	6
499996	Offline	C	Fiji	Vegetables	7/19/2012	1302269.18	7
499997	Offline	C	Fiji	Fruits	5/1/2011	88998.87	
499998	Offline	C	Fiji	Vegetables	11/27/2013	1464032.18	8
499999	Offline	C	Fiji	Fruits	4/27/2012	69956.34	

500000 rows × 12 columns

```
In [343]: ▶ for i,row in x.iterrows():
            sql = "INSERT INTO SalesProject.dbo.SALES_FACT_TABLE(LOCATIONID,ITEMTYPEID,ORDERID,DATEID,TOTALREVENUE,TOTALCOST,TOTALPROFIT,UNITSOLD)values(37,1,97,97,74201.49,55034.76,19166.73,7953);"
            cursor.execute(sql)
```

```
In [344]: ▶ conn.commit()
```

```
In [272]: ▶ sql
```

```
Out[272]: 'INSERT INTO SalesProject.dbo.SALES_FACT_TABLE(LOCATIONID,ITEMTYPEID,ORDERID,DATEID,TOTALREVENUE,TOTALCOST,TOTALPROFIT,UNITSOLD)values(37,1,97,97,74201.49,55034.76,19166.73,7953);'
```