read the data from given url to DataFrame as sales_data https://raw.githubusercontent.com/svkarthik86/Assignment/main/Sales_Data/Sales_January_2019.csv

Q1.Rename The column name with valid format: oldname:Order ID,Product,Quantity Ordered,Price Each,Order Date,Purchase Address
newname:Order_ID,Product,Quantity_Ordered,Price_Each,Order_Date,Purchase_Address

Q2.insert new column as order_price compute order_price =Price_Each*Quantity_Ordered

Q3.Delete the column Purchase_Address

Q4.insert new row data as at last 100001,OnePhone,1,600, 01/22/19 21:25,600

Q5.Gruopby Product and apply the aggregate function min,max,mean on order_price

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

read the data from given url to DataFrame as sales_data https://raw.githubusercontent.com/svkarthik86/Assignment/main/Sales_Data/Sales_January_2019.csv

In [3]:
```
url="https://raw.githubusercontent.com/svkarthik86/Assignment/main/Sales_Data/Sales_January_2019.csv"
sales_data=pd.read_csv(url)
sales_data
```

Out[3]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	141234	iPhone	1	700	01/22/19 21:25	944 Walnut St, Boston, MA 02215
1	141235	Lightning Charging Cable	1	14.95	01/28/19 14:15	185 Maple St, Portland, OR 97035
2	141236	Wired Headphones	2	11.99	01/17/19 13:33	538 Adams St, San Francisco, CA 94016
3	141237	27in FHD Monitor	1	149.99	01/05/19 20:33	738 10th St, Los Angeles, CA 90001
4	141238	Wired Headphones	1	11.99	01/25/19 11:59	387 10th St, Austin, TX 73301
...	...	...	...	...	...	...
9718	150497	20in Monitor	1	109.99	01/26/19 19:09	95 8th St, Dallas, TX 75001
9719	150498	27in FHD Monitor	1	149.99	01/10/19 22:58	403 7th St, San Francisco, CA 94016
9720	150499	ThinkPad Laptop	1	999.99	01/21/19 14:31	214 Main St, Portland, OR 97035
9721	150500	AAA Batteries (4-pack)	2	2.99	01/15/19 14:21	810 2nd St, Los Angeles, CA 90001
9722	150501	Google Phone	1	600	01/13/19 16:43	428 Cedar St, Boston, MA 02215

9723 rows × 6 columns

Q1.Rename The column name with valid format: oldname:Order ID,Product,Quantity Ordered,Price Each,Order Date,Purchase Address newname:Order_ID,Product,Quantity_Ordered,Price_Each,Order_Date,Purchase_Address

In [6]:
```
sales_data.rename(columns={"Order ID":"Order_ID","Product":"Product","Quantity Ordered":"Quantity_Ordered","Price Each":"Price_Each","Order Date":"Order_Date","Purchase Address":"Pu
```

In [7]:
```
sales_data.dropna(inplace=True)
```

In [8]:
```
sales_data
```

Out[8]:

	Order_ID	Product	Quantity_Ordered	Price_Each	Order_Date	Purchase_Address
0	141234	iPhone	1	700	01/22/19 21:25	944 Walnut St, Boston, MA 02215
1	141235	Lightning Charging Cable	1	14.95	01/28/19 14:15	185 Maple St, Portland, OR 97035
2	141236	Wired Headphones	2	11.99	01/17/19 13:33	538 Adams St, San Francisco, CA 94016
3	141237	27in FHD Monitor	1	149.99	01/05/19 20:33	738 10th St, Los Angeles, CA 90001
4	141238	Wired Headphones	1	11.99	01/25/19 11:59	387 10th St, Austin, TX 73301
...	...	...	...	...	...	...
9718	150497	20in Monitor	1	109.99	01/26/19 19:09	95 8th St, Dallas, TX 75001
9719	150498	27in FHD Monitor	1	149.99	01/10/19 22:58	403 7th St, San Francisco, CA 94016
9720	150499	ThinkPad Laptop	1	999.99	01/21/19 14:31	214 Main St, Portland, OR 97035
9721	150500	AAA Batteries (4-pack)	2	2.99	01/15/19 14:21	810 2nd St, Los Angeles, CA 90001
9722	150501	Google Phone	1	600	01/13/19 16:43	428 Cedar St, Boston, MA 02215

9697 rows × 6 columns

In [9]:
```
sales_data.drop_duplicates(inplace=True)
```

In [10]:
```
sales_data[sales_data.duplicated()]
```

Out[10]:

Order_ID	Product	Quantity_Ordered	Price_Each	Order_Date	Purchase_Address

In [ ]:
```
sales_data=sales_data[~(sales_data.Price_Each=="Price Each")]
```

Q2.insert new column as order_price and compute order_price =Price_Each*Quantity_Ordered

In [ ]:
```
sales_data["order_price"]=sales_data.Price_Each.astype(float) * sales_data.Quantity_Ordered.astype(float)
```

C:\Users\RGS\AppData\Local\Temp\ipykernel_10376\210156011.py:1: 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 sales_data["order_price"]=sales_data.Price_Each.astype(float) * sales_data.Quantity_Ordered.astype(float)

In [14]:
```
sales_data
```

Out[14]:

	Order_ID	Product	Quantity_Ordered	Price_Each	Order_Date	Purchase_Address
0	141234	iPhone	1	700	01/22/19 21:25	944 Walnut St, Boston, MA 02215
1	141235	Lightning Charging Cable	1	14.95	01/28/19 14:15	185 Maple St, Portland, OR 97035
2	141236	Wired Headphones	2	11.99	01/17/19 13:33	538 Adams St, San Francisco, CA 94016
3	141237	27in FHD Monitor	1	149.99	01/05/19 20:33	738 10th St, Los Angeles, CA 90001
4	141238	Wired Headphones	1	11.99	01/25/19 11:59	387 10th St, Austin, TX 73301
...	...	...	...	...	...	...
9718	150497	20in Monitor	1	109.99	01/26/19 19:09	95 8th St, Dallas, TX 75001
9719	150498	27in FHD Monitor	1	149.99	01/10/19 22:58	403 7th St, San Francisco, CA 94016
9720	150499	ThinkPad Laptop	1	999.99	01/21/19 14:31	214 Main St, Portland, OR 97035
9721	150500	AAA Batteries (4-pack)	2	2.99	01/15/19 14:21	810 2nd St, Los Angeles, CA 90001
9722	150501	Google Phone	1	600	01/13/19 16:43	428 Cedar St, Boston, MA 02215

9672 rows × 6 columns

Q3.Delete the column Purchase_Address

In [16]:
```
sales_data.drop("Purchase_Address",axis=1,inplace=True)
```

In [ ]:
```
C:\Users\RGS\AppData\Local\Temp\ipykernel_10376\236795181.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  sales_data.drop("Purchase_Address",axis=1,inplace=True)
```

In [21]:
```
sales_data
```

Out[21]:

	Order_ID	Product	Quantity_Ordered	Price_Each	Order_Date
0	141234	iPhone	1	700	01/22/19 21:25
1	141235	Lightning Charging Cable	1	14.95	01/28/19 14:15
2	141236	Wired Headphones	2	11.99	01/17/19 13:33
3	141237	27in FHD Monitor	1	149.99	01/05/19 20:33
4	141238	Wired Headphones	1	11.99	01/25/19 11:59
...	...	...	...	...	...
9718	150497	20in Monitor	1	109.99	01/26/19 19:09
9719	150498	27in FHD Monitor	1	149.99	01/10/19 22:58
9720	150499	ThinkPad Laptop	1	999.99	01/21/19 14:31
9721	150500	AAA Batteries (4-pack)	2	2.99	01/15/19 14:21
9722	150501	Google Phone	1	600	01/13/19 16:43

9672 rows × 5 columns

Q4.insert new row data as at last 100001,OnePhone,1,600, 01/22/19 21:25,600

In [22]:
```
sales_data=pd.concat([sales_data,pd.DataFrame([[100001,"OnePhone",1,600,"01/22/19 21:25",600]],columns=['Order_ID', 'Product', 'Quantity_Ordered', 'Price_Each', 'Order_Date',
       'order_price'])],ignore_index=True)
```

In [23]:
```
sales_data
```

Out[23]:

	Order_ID	Product	Quantity_Ordered	Price_Each	Order_Date	order_price
0	141234	iPhone	1	700	01/22/19 21:25	NaN
1	141235	Lightning Charging Cable	1	14.95	01/28/19 14:15	NaN
2	141236	Wired Headphones	2	11.99	01/17/19 13:33	NaN
3	141237	27in FHD Monitor	1	149.99	01/05/19 20:33	NaN
4	141238	Wired Headphones	1	11.99	01/25/19 11:59	NaN
...	...	...	...	...	...	...
9668	150498	27in FHD Monitor	1	149.99	01/10/19 22:58	NaN
9669	150499	ThinkPad Laptop	1	999.99	01/21/19 14:31	NaN
9670	150500	AAA Batteries (4-pack)	2	2.99	01/15/19 14:21	NaN
9671	150501	Google Phone	1	600	01/13/19 16:43	NaN
9672	100001	OnePhone	1	600	01/22/19 21:25	600.0

9673 rows × 6 columns

Q5.Gruopby Product and apply the aggregate function min,max,mean on order_price

In [24]:
```
sales_data.groupby("Product")["order_price"].agg([min,max,np.mean])
```

Out[24]:

Product	min	max	mean
20in Monitor	NaN	NaN	NaN
27in 4K Gaming Monitor	NaN	NaN	NaN
27in FHD Monitor	NaN	NaN	NaN
34in Ultrawide Monitor	NaN	NaN	NaN
AA Batteries (4-pack)	NaN	NaN	NaN
AAA Batteries (4-pack)	NaN	NaN	NaN
Apple Airpods Headphones	NaN	NaN	NaN
Bose SoundSport Headphones	NaN	NaN	NaN
Flatscreen TV	NaN	NaN	NaN
Google Phone	NaN	NaN	NaN
LG Dryer	NaN	NaN	NaN
LG Washing Machine	NaN	NaN	NaN
Lightning Charging Cable	NaN	NaN	NaN
Macbook Pro Laptop	NaN	NaN	NaN
OnePhone	600.0	600.0	600.0
Product	NaN	NaN	NaN
ThinkPad Laptop	NaN	NaN	NaN
USB-C Charging Cable	NaN	NaN	NaN
Vareebadd Phone	NaN	NaN	NaN
Wired Headphones	NaN	NaN	NaN
iPhone	NaN	NaN	NaN

In [ ]:
```

```