In [3]:

```
import pandas as pd
```

In [4]:

```
tata1 = pd.read_csv("D:\KSR\Python\Assignments\Assignment_2\TATA_TB1.csv", encoding = 'IS
```

In [5]:

```
tata1.head(5)
```

Out[5]:

	OrderID
0	BN-2011-7407039
1	AZ-2011-9050313
2	AZ-2011-6674300
3	BN-2011-2819714
4	AZ-2011-617423

In [6]:

```
tata2 = pd.read_csv("D:\KSR\Python\Assignments\Assignment_2\TATA_TB2.csv", encoding = 'IS
```

In [7]:

```
tata2.head(5)
```

Out[7]:

	OrderID
0	BN-2011-7407039
1	AZ-2011-9050313
2	AZ-2011-6674300
3	BN-2011-2819714
4	BN-2011-2819714

In [109]:

```
df = pd.merge(tata1,tata2, on ='OrderID',how='outer')
```

In [110]:

```
df.head(2)
```

Out[110]:

	OrderID
0	AZ-2011-1029887
1	AZ-2011-1029887

In [6]:

#1) Write a query to calculate the total records in two tables?

In [20]:

```
pd.merge(tata1,tata2, on ='OrderID',how='outer')[['OrderID']].count()
```

Out[20]:

```
OrderID    8047
dtype: int64
```

In [21]:

#2) Write a query to calculate the total unique count of customers?

In [9]:

```
tata1['CustomerName'].nunique()
```

Out[9]:

792

In [10]:

#3) Write a query to fetch the oldest order date and latest order date?

In [10]:

```
tata1['OrderDate'].agg({'min','max'})
```

Out[10]:

```
max    2014-12-31
min    2011-01-01
Name: OrderDate, dtype: object
```

In [11]:

#4) Write query to get unique years?

In [11]:

```
import datetime
import warnings
warnings.filterwarnings("ignore")
```

In [13]:

```
tata1['OrderDate'] = pd.to_datetime(tata1['OrderDate'])
```

In [15]:

```
tata1.dtypes
```

Out[15]:

```
OrderID                 object
OrderDate       datetime64[ns]
CustomerName            object
City                    object
Country                 object
Region                  object
Segment                 object
ShipDate                object
ShipMode                object
State                   object
lon                    float64
lat                    float64
dtype: object
```

In [16]:

```
tata1['Year'] = tata1['OrderDate'].dt.year
```

In [17]:

```
tata1.head(3)
```

Out[17]:

	OrderID
0	BN-2011-7407039
1	AZ-2011-9050313
2	AZ-2011-6674300

In [18]:

```
tata1[['Year']].nunique()
```

Out[18]:

```
Year    4
dtype: int64
```

In [12]:

```
#5) Write a query to get the no. of regions and display the region names
```

In [22]:

```
tata1.groupby('Region')[['Region']].count().rename(columns={'Country':'TotalCountry'})
```

Out[22]:

	Region
Region	
Central	2234
North	950
South	933

In [14]:

#6) Write a query to get the no. of countries and display the country names?

In [19]:

```
tata1.groupby('Country')[['Country']].count().rename(columns={'Country':'TotalCountry'})
```

Out[19]:

	TotalCountry
Country	
Austria	135
Belgium	68
Denmark	29
Finland	34
France	991
Germany	806
Ireland	50
Italy	493
Netherlands	194
Norway	37
Portugal	37
Spain	403
Sweden	100
Switzerland	40
United Kingdom	700

In [16]:

#7) Write a query to get the no. of states and display the state names?

In [18]:

```
tata1.groupby('State')[['State']].count().rename(columns={'State':'TotalStates'})
```

Out[18]:

	TotalStates
State	
Abruzzi	7
Alsace-Champagne-Ardenne-Lorraine	56
Andalusía	49

Antwerp	20
Apulia	42
...	...
Wales	15
West Flanders	3
Zealand	1
Zeeland	1
Zürich	17

127 rows × 1 columns

In [23]:

#8) Write a query to get the no. of cities and display the city names?

In [24]:

```
tata1.groupby('City')[['City']].count().rename(columns={'City':'TotalCities'})
```

Out[24]:

	TotalCities
City	
Aachen	6
Aalen	1
Aalst	1
Abbeville	2
Aberdeen	2
...	...
Zamora	2
Zeist	4
Zurich	15
Zwickau	2
Zwolle	2

999 rows × 1 columns

In [25]:

#9) Write a query to calculate the total count of products?

In [34]:

```
tata2['ProductName'].nunique()
```

Out[34]:

In [35]:

#10)Write a query to calculate total sales, total profit and total order quantity?

In [41]:

```
tata2.agg({"Sales":"sum","Profit":"sum","OrderQuantity":"sum"})
```

Out[41]:

```
Sales            2348482
Profit            283240
OrderQuantity      30354
dtype: int64
```

In [42]:

```
#11)Write a query to calculate the total sales amount, total order quantity for each cate
#Display the category, total sales, and total order qty and order by total sales from
#highest to lowest?
```

In [44]:

```
tata2.groupby('Category').agg({'Sales':'sum','OrderQuantity':'sum'})  \
.rename(columns={'Sales':'TotalSales','OrderQuantity':'TotalOrderQuantity'})  \
.sort_values('TotalSales', ascending = False)
```

Out[44]:

	TotalSales
Category	
Technology	886015
Office Supplies	823658
Furniture	638809

In [45]:

```
#12)Write a query to calculate the total profit amount for each category. Display the
#category, total profit, and total order qty and order by total profit from highest to lo
```

In [46]:

```
tata2.groupby('Category').agg({'Profit':'sum','OrderQuantity':'sum'})  \
.rename(columns={'Profit':'TotalProfit','OrderQuantity':'TotalOrderQuantity'})  \
.sort_values('TotalProfit', ascending = False)
```

Out[46]:

	TotalProfit
Category	
Office Supplies	124952
Technology	108554
Furniture	49734

In [47]:

#13)Write a query to fetch the subcategories where total sales are greater than 100000?

In [65]:

```
tata2.groupby('SubCategory')[['Sales']].sum() \
.query('Sales>100000')  \
.rename(columns = {"Sales":"TotalSales"})
```

Out[65]:

	TotalSales
SubCategory	
Accessories	131309
Appliances	209900
Art	127184
Bookcases	294396
Chairs	186698
Copiers	290081
Machines	182066
Phones	282559
Storage	272489

In [66]:

#14)Write a query to fetch the products where total profit is greater than 2500 and sort
#based on profit from highest to lowest?

In [67]:

```
tata2.groupby('ProductName')[['Profit']].sum() \
.query('Profit>2500')  \
.rename(columns = {"Profit":"TotalProfit"})  \
.sort_values('TotalProfit', ascending = False)
```

Out[67]:

	TotalProfit
ProductName	
Nokia Smart Phone, Full Size	7583
Hoover Stove, Red	6139
Hamilton Beach Stove, Silver	5778
SAFCO Executive Leather Armchair, Black	4324
Safco Classic Bookcase, Metal	4183
Cisco Smart Phone, with Caller ID	4055
Brother Fax Machine, Laser	3918
Eldon Lockers, Industrial	3611
Cisco Smart Phone, Cordless	3388
Hamilton Beach Stove, Red	2738
Belkin Router, USB	2677

Nokia Smart Phone, Cordless	2633
Cuisinart Refrigerator, Black	2627
Eldon File Cart, Single Width	2539

In [68]:

#15)Write a query to get the total sales and total profit for Office Supplies category?

In [74]:

```
tata2.query('Category == "Office Supplies"') \
.agg({'Sales':'sum','Profit':'sum'})
```

Out[74]:

```
Sales     823658
Profit    124952
dtype: int64
```

In [75]:

#16)Write a query to get the total sales and total profit for Furniture category and Tabl

In [82]:

```
tata2.query(('Category == "Furniture"') and ('SubCategory == ["Tables","Bookcases"]')) \
.agg({'Sales':'sum','Profit':'sum'})
```

Out[82]:

```
Sales     383874
Profit     22924
dtype: int64
```

In [83]:

#17)Write a query to get the total sales and total profit for Technology category and the
#Accessories, Copiers, Phones sub-categories ?

In [84]:

```
tata2.query(('Category == "Technology"') and ('SubCategory == ["Accessories","Copiers","P
.agg({'Sales':'sum','Profit':'sum'})
```

Out[84]:

```
Sales     703949
Profit     97236
dtype: int64
```

In [85]:

```
#18)Write a query to get total sales and total profit by Region, Segment and sort the sal
#from highest to lowest??
```

In [102]:

```
pd.merge(tata1,tata2, on = 'OrderID', how = 'inner' ) [['Segment','Region','Sales','Profi
.groupby(["Region","Segment"]).agg({'Sales':'sum','Profit':'sum'}) \
.rename(columns = {"Sales":"TotalSales", "Profit":"TotalProfit"}) \
.sort_values('TotalSales', ascending = False)
```

Out[102]:

Region	Segment
Central	Consumer
	Corporate
North	Consumer
South	Consumer
Central	Home Office
South	Corporate
North	Corporate
	Home Office
South	Home Office

In [103]:

```
#19)Write a query to get total sales and total profit by Country, State and city and sort
```

In [106]:

```
pd.merge(tata1,tata2, on = 'OrderID', how = 'inner' ) [['Country','State','City','Sales',
.groupby(['Country','State','City']).agg({'Sales':'sum','Profit':'sum'}) \
.rename(columns = {"Sales":"TotalSales", "Profit":"TotalProfit"}) \
.sort_values('TotalSales', ascending = False)
```

Out[106]:

Country	State
United Kingdom	England
Germany	Berlin
Austria	Vienna
Spain	Madrid
France	Ile-de-France
...	...
Germany	Hesse
France	Provence-Alpes-Côte d'Azur
	Languedoc-Roussillon-Midi-Pyrénées
	Auvergne-Rhône-Alpes
	Ile-de-France

1001 rows × 2 columns

In [107]:

#20)Write a query to get total sales and total orderqty by CustomerName sort it by total
#sales from highest to lowest?

In [109]:

```
pd.merge(tata1,tata2, on = 'OrderID', how = 'inner' ) [['CustomerName','Sales','OrderQuan
.groupby(['CustomerName']).agg({'Sales':'sum','OrderQuantity':'sum'}) \
.rename(columns = {"Sales":"TotalSales", "OrderQuantity":"TotalOrderQuantity"}) \
.sort_values('TotalSales', ascending = False)
```

Out[109]:

	TotalSales
CustomerName	
Angie Massengill	16146
Lola Hughes	13191
Ashton Charles	13056
Isaac David	11271
Philip Newsom	10893
...	...
Alexandra Mahmood	47
Max Baker	44
Grace Brown	23
Jodie Garner	16
James Brown	14

792 rows × 2 columns

In [110]:

#21)Identify the top 5 products with the highest sales (by sales amount). Show the produc
#name, total sales, and total qty?

In [116]:

```
pd.merge(tata1,tata2, on = 'OrderID', how = 'inner' ) [['CustomerName','Sales','OrderQuan
.groupby(['CustomerName']).agg({'Sales':'sum','OrderQuantity':'sum'}) \
.rename(columns = {"Sales":"TotalSales", "OrderQuantity":"TotalOrderQuantity"}) \
.sort_values('TotalSales', ascending = False).head(5)
```

Out[116]:

	TotalSales
CustomerName	
Angie Massengill	16146
Lola Hughes	13191

Ashton Charles	13056
Isaac David	11271
Philip Newsom	10893

In [112]:

#22)Write a query to get total sales by City having sales greater than 35000?

In [129]:

```
pd.merge(tata1,tata2, on = 'OrderID', how = 'inner' ) [['City','Sales']] \
.groupby(['City'])[['Sales']].sum() \
.query('Sales>35000')  \
.rename(columns = {"Sales":"TotalSales"}) \
.sort_values('TotalSales', ascending = False)
```

Out[129]:

	TotalSales
City	
London	69230
Berlin	52555
Vienna	51844
Madrid	44981
Paris	42245

In [122]:

#23)Write a query to get total sales by CustomerName having sales greater than 10000?

In [128]:

```
pd.merge(tata1,tata2, on = 'OrderID', how = 'inner' ) [['CustomerName','Sales']] \
.groupby(['CustomerName'])[['Sales']].sum() \
.query('Sales>10000')  \
.rename(columns = {"Sales":"TotalSales"}) \
.sort_values('TotalSales', ascending = False)
```

Out[128]:

	TotalSales
CustomerName	
Angie Massengill	16146
Lola Hughes	13191
Ashton Charles	13056
Isaac David	11271
Philip Newsom	10893
Joel Peters	10477
Bettie Lang	10466
Audrey Knowles	10363

In [130]:

#24)Write a query to get total sales and total profit by shipmode and sort the sales in

#ascending order and profit in descending order?

In [133]:

```
pd.merge(tata1,tata2, on = 'OrderID', how = 'inner' ) [['ShipMode','Sales','Profit']] \
.groupby(['ShipMode']).agg({'Sales':'sum','Profit':'sum'}) \
.rename(columns = {"Sales":"TotalSales","Profit":"TotalProfit"}) \
.sort_values(['TotalSales','TotalProfit'] , ascending = [True,False])
```

Out[133]:

	TotalSales
ShipMode	
Immediate	131314
Priority	320426
Economy Plus	483965
Economy	1412777

In [134]:

#25)Write a query to get total sales for North and central region?

In [136]:

```
pd.merge(tata1,tata2, on = 'OrderID', how = 'inner' ) [['Region','Sales']] \
.query(('Region=="North"') and('Region=="North"'))[['Sales']].sum()
```

Out[136]:

```
Sales    515979
dtype: int64
```

In [137]:

#26)Write a query to get total sales and total profit for Italy and Spain countries?

In [141]:

```
pd.merge(tata1,tata2, on = 'OrderID', how = 'inner' ) [['Country','Sales','Profit']] \
.query(('Country=="Italy"') and ('Country=="Spain"')) \
.agg({'Sales':'sum','Profit':'sum'})
```

Out[141]:

```
Sales     249402
Profit     47067
dtype: int64
```

In [142]:

#28) Find the top 10 customers who spent the most across all transactions. Display the

#customer name, total amount spent, and number of orders placed?

In [157]:

```
pd.merge(tata1,tata2, on = 'OrderID', how = 'inner' ) [['OrderID','CustomerName','Sales']
.groupby('CustomerName').agg({'Sales':'sum','OrderID':'count'}) \
.rename(columns = {'Sales':'TotalSpent','OrderID':'No.ofOrdersPlaced'}) \
.sort_values('TotalSpent', ascending = False).head(10)
```

Out[157]:

	TotalSpent
CustomerName	
Angie Massengill	16146
Lola Hughes	13191
Ashton Charles	13056
Isaac David	11271
Philip Newsom	10893
Joel Peters	10477
Bettie Lang	10466
Audrey Knowles	10363
Lilly Le Grand	9962
Elijah Sodeman	9689

In [158]:

#29)Write a query to find which products are most preferred by customers based on the
#total sales. Display customer name, favorite products (top 3 products per each
#customer), and total sales on that product?

In [162]:

```
pd.merge(tata1,tata2, on = 'OrderID', how = 'inner' ) [['CustomerName','ProductName','Sal
.groupby(['CustomerName','ProductName'])[['Sales']].sum() \
.rename(columns = {'Sales':'TotalSales'}) \
.sort_values(['CustomerName','TotalSales'], ascending = [True,False]) \
.groupby('CustomerName').head(3)
```

Out[162]:

CustomerName	ProductName
Aaron Bootman	Brother Fax and Copier, Color
	HP Personal Copier, Digital
	Belkin Keyboard, Erganomic
Aaron Cunningham	Eldon File Cart, Single Width
	Hon Rocking Chair, Red
...	...
Zoe Lowin	Panasonic Calculator, Wireless
	Deflect-O Frame, Durable
Zona Meyer	Hewlett Copy Machine, Color
	Stanley Sketch Pad, Water Color

	Novimex Executive Leather Armchair, Adjustable

2322 rows × 1 columns

In [165]:

#30)Write a query to get 7th rank customer name based on total sales? Display customer na

In [24]:

```
df =pd.merge(tata1,tata2, on = 'OrderID', how = 'inner').groupby('CustomerName')[['Sales'
.rename(columns = {'Sales':'TotalSales'})
```

In [25]:

```
df.head(10)
```

Out[25]:

	TotalSales
CustomerName	
Aaron Bootman	4281
Aaron Cunningham	2463
Aaron Davey	2238
Aaron Macrossan	81
Abbie Perry	1313
Abby Colebe	2805
Abby Mei	5940
Abby Muramats	1112
Abigail Humffray	813
Ada Dalton	3520

In [28]:

```
df['First_Rank'] = df['TotalSales'].rank(method = 'first', ascending = False).astype('int
```

In [30]:

```
df.sort_values('First_Rank', ascending = True)
```

Out[30]:

	TotalSales
CustomerName	
Angie Massengill	16146
Lola Hughes	13191
Ashton Charles	13056
Isaac David	11271
Philip Newsom	10893
...	...
Alexandra Mahmood	47

Max Baker	44
Grace Brown	23
Jodie Garner	16
James Brown	14

792 rows × 2 columns

In [31]:

```
df.query('First_Rank == 7')
```

Out[31]:

	TotalSales
CustomerName	
Bettie Lang	10466

In [166]:

```
#31)Write query to get total sales and total profit in years 2011 and 2013 Display year w
```

In [107]:

```
df = pd.merge(tata1,tata2, on = 'OrderID', how = 'inner' )[['Year','Sales','Profit']] \
.groupby("Year").agg({'Sales':'sum','Profit':'sum'}).query(('Year == 2011 or Year == 2013
.rename(columns = {"Sales":"TotalSales","Profit":"TotalProfit"})
```

In [108]:

```
df.head(5)
```

Out[108]:

	TotalSales
Year	
2011	414348
2013	630224

In [167]:

```
#32)Write a query to get total sales, total profit and total order qty by country, state,
#and sub-category and sort it from highest to lowest based on sales column?
```

In [168]:

```
pd.merge(tata1,tata2, on = 'OrderID', how = 'inner' ) [['Country','State','Category','Sub
.groupby(['Country','State','Category','SubCategory'])  \
.agg({'Sales':'sum','Profit':'sum','OrderQuantity':'sum'})\
.rename(columns = {'Sales':'TotalSales','Profit':'TotalProfit','OrderQuantity':'TotalOdrQ
.sort_values('TotalSales', ascending = False)
```

Out[168]:

Country	State
United Kingdom	England
...	...
Belgium	Brussels
	Limburg
Netherlands	Limburg
Finland	Uusimaa
Italy	Umbria

1364 rows × 3 columns

In [169]:

#33)write a function to get the region sales dynamically, if we pass any region, that reg

In [119]:

```
rSales = df.groupby('Region')[['Sales']].sum()
```

In [120]:

```
def region_sales(region = ''):
    return rSales.query('Region == @region')
```

In [122]:

```
region_sales('South')
```

Out[122]:

	Sales
Region	
South	517250

In [ ]:

#34)write a function to get the country sales, profit dynamically, if we pass any country
#country sales, profit should be displayed?

In [125]:

```
CSales = df.groupby('Country')[['Sales']].sum()
```

In [126]:

```
def country_sales(country = ''):
    return CSales.query('Country == @country')
```

In [127]:

```
country_sales('Germany')
```

Out[127]:

	Sales
Country	
Germany	488681

In [171]:

```
#35)write a function to get the Category orderquantity dynamically, if we pass any Catego
#that Category orderqunatity should be displayed.?
```

In [130]:

```
cOrderQuantity = df.groupby('Category')[['OrderQuantity']].sum()
```

In [131]:

```
def category_quantity(category=''):
    return cOrderQuantity.query('Category==@category')
```

In [132]:

```
category_quantity('Furniture')
```

Out[132]:

	OrderQuantity
Category	
Furniture	4641

In [ ]:

```
#36)write a function to get the City sales and profit dynamically, if we pass any city, t
#sales and profit should be displayed?
```

In [133]:

```
citysales = df.groupby('City')[['Sales','Profit']].sum()
```

In [134]:

```
def city_data(city=''):
    return citysales.query('City == @city')
```

In [135]:

```
city_data('Leipzig')
```

Out[135]:

	Sales
City	
Leipzig	12380

In [ ]:

0-8859-1')

OrderDate	CustomerName	City	Country
01-01-2011	Ruby Patel	Stockholm	Sweden
03-01-2011	Summer Hayward	Southport	United Kingdom
04-01-2011	Devin Huddleston	Valence	France
04-01-2011	Mary Parker	Birmingham	United Kingdom
05-01-2011	Daniel Burke	Echirolles	France

0-8859-1')

ProductName	Discount	Sales	Profit
Enermax Note Cards, Premium	0.5	45	-26
Dania Corner Shelving, Traditional	0	854	290
Binney & Smith Sketch Pad, Easy-Erase	0	140	21
Boston Markers, Easy-Erase	0.5	27	-22
Eldon Folders, Single Width	0.5	17	-1

OrderDate	CustomerName	City	Country
31-07-2011	Brooke Hodgson	Leipzig	Germany
31-07-2011	Brooke Hodgson	Leipzig	Germany

OrderDate	CustomerName	City	Country
01-01-2011	Ruby Patel	Stockholm	Sweden
03-01-2011	Summer Hayward	Southport	United Kingdom
04-01-2011	Devin Huddleston	Valence	France

gory.

TotalOrderQuantity
5811
19902
4641

west?

TotalOrderQuantity
19902
5811
4641

it

es, Bookcases sub-categories ?

hones″]')) \

es

t']] \

TotalSales	TotalProfit
701892	82146
396437	52115
267955	35686
266435	29615
216924	23375
177709	13647
163991	26872
84033	8881
73106	10903

the sales from highest to lowest??

'Profit']] \

	TotalSales	TotalProfit
City		
London	69230	13931
Berlin	52555	5942
Vienna	51844	13207
Madrid	44981	11129
Paris	42245	6680
...	...	...
Friedberg	18	6
Sorgues	14	3
Sète	14	7
Aix-les-Bains	10	4
Osny	5	2

tity']] \

TotalOrderQuantity
102
127
63
110
80
...
2
1
1
1
2

t

tity']] \

TotalOrderQuantity
102
127

63
110
80

TotalProfit
17569
32639
54336
178696

] \

No.ofOrdersPlaced
28
26
16
29
21
32
22
20
21
23

es']] \

TotalSales
1543
632
245
809
392
...
302
214
1060
571

410

me, sales amount and rank.

]].sum() \

')

First_Rank
1
2
3
4
5
...
788

789
790
791
792

First_Rank
7

ise total sales and total profit?

')) \

TotalProfit
54487
77200

category

Category','Sales','Profit','OrderQuantity']] \

ty'}) \

		TotalSales	TotalProfit

Category	SubCategory		
Furniture	Bookcases	52576	12790
Technology	Phones	52262	11124
	Copiers	49025	12408
Office Supplies	Storage	42752	5987
Technology	Machines	33920	7653
...	...	...	...
Office Supplies	Fasteners	11	3
Office Supplies	Labels	10	2
Office Supplies	Envelopes	9	-9
Office Supplies	Labels	9	3
Office Supplies	Labels	9	4

ion sales should be displayed?

, that

ry,

hat City

Profit
2034

Region	Segment	ShipDate	ShipMode	State	lon
North	Home Office	05-01-2011	Economy Plus	Stockholm	18.068581
North	Consumer	07-01-2011	Economy	England	-3.010113
Central	Consumer	08-01-2011	Economy	Auvergne-Rhône-Alpes	4.89236
North	Corporate	09-01-2011	Economy	England	-1.890401
Central	Home Office	07-01-2011	Priority	Auvergne-Rhône-Alpes	5.718034

OrderQuantity	Category	SubCategory
3	Office Supplies	Paper
7	Furniture	Bookcases
3	Office Supplies	Art
2	Office Supplies	Art
2	Office Supplies	Storage

Region	Segment	ShipDate	ShipMode	State	lon
Central	Consumer	03-08-2011	Economy Plus	Saxony	12.373075
Central	Consumer	03-08-2011	Economy Plus	Saxony	12.373075

Region	Segment	ShipDate	ShipMode	State	lon
North	Home Office	05-01-2011	Economy Plus	Stockholm	18.068581
North	Consumer	07-01-2011	Economy	England	-3.010113
Central	Consumer	08-01-2011	Economy	Auvergne-Rhône-Alpes	4.89236

TotalOdrQty

237
198
214
569
187
...
1
2
2
2
1

lat
59.329324
53.645708
44.933393
52.486243
45.142151

lat	Year	ProductName	Discount	Sales	Profit
51.339695	2011	Deflect-O Door Stop, Erganomic	0	85	15
51.339695	2011	Novimex Color Coded Labels, 5000 Label Set	0	26	7

lat	Year
59.329324	2011
53.645708	2011
44.933393	2011

OrderQuantity	Category	SubCategory
2	Furniture	Furnishings
2	Office Supplies	Labels