In [1]:
```
import pandas as pd
#step 1: create sample dataset
data= {
    'Region' : ['North', 'South', 'East', 'West', 'North', 'South', 'East', 'West']
    'Product' : ['A', 'B','A', 'B', 'C', 'C', 'B', 'A' ],
    'Sales' : [150,200,300,400,250,180,220,310],
    'Quantity' : [10,15,20,25,12,14,16,18],
}

```

In [2]:
```
df = pd.DataFrame(data)
print("Sample Dataset: \n", df)
```

```
Sample Dataset:
   Region Product  Sales  Quantity
0  North       A    150        10
1  South       B    200        15
2   East       A    300        20
3   West       B    400        25
4  North       C    250        12
5  South       C    180        14
6   East       B    220        16
7   West       A    310        18
```

In [3]:
```
#step 2: grouping and aggreagration
#aggreagrating sales by region (sum aggregration)
sales_by_region = df.groupby('Region')['Sales'].sum()
print("\n Total sales by region: \n", sales_by_region)
```

```
 Total sales by region:
 Region
East     520
North    400
South    380
West     710
Name: Sales, dtype: int64
```

In [6]:
```
#aggregating sales and quantity by product(mean aggreagration)
mean_by_product = df.groupby('Product')[['Sales', 'Quantity']].mean()
print("\nMean sales by quantity by product: \n ", mean_by_product)
```

```
Mean sales by quantity by product:
                Sales   Quantity
Product
A         253.333333  16.000000
B         273.333333  18.666667
C         215.000000  13.000000
```

In [8]:
```
#aggregating cunt of sales by region(cuint agggreagration)
count_by_region = df.groupby('Region')['Sales'].count()
print("\n Count of sales records by region:\n ",sales_by_region)
```

```
 Count of sales records by region:
  Region
East     520
North    400
South    380
West     710
Name: Sales, dtype: int64
```

In [10]:
```
#custom aggregration: calculate min and max: sales by region
custom_aggregration = df.groupby('Region')['Sales'].agg(['min', 'max'])
print("\n Custom aggregration (min and max sales by region): \n ",custom_aggregrati
```

```
 Custom aggregration (min and max sales by region):
          min  max
Region
East    220  300
North   150  250
South   180  200
West    310  400
```

In [11]:
```
#step 3: multilevel aggregration
#aggregrating sales by region and product
mult_level_agg = df.groupby(['Region', 'Product'])['Sales'].sum()
print("\n sales by rgion and prodcut:\n" , mult_level_agg)
```

```
 sales by rgion and prodcut:
 Region  Product
East    A          300
        B          220
North   A          150
        C          250
South   B          200
        C          180
West    A          310
        B          400
Name: Sales, dtype: int64
```

In [12]:
```
#step 4: reset index for multlevel aggregration
multi_level_agg_reset = mult_level_agg.reset_index()
print("\n Sales by region and product (reset index):\n ", multi_level_agg_reset)
```

```
 Sales by region and product (reset index):
    Region Product  Sales
0   East       A    300
1   East       B    220
2  North       A    150
3  North       C    250
4  South       B    200
5  South       C    180
6   West       A    310
7   West       B    400
```

In [ ]:
```
Objective:

To understand and implement:

Time Aggregation: Aggregating data over different time periods (e.g., monthly, year

Spatial Aggregation: Aggregating data by spatial attributes (e.g., by region, city)
```

In [15]:
```
import pandas as pd
#step 1: extend dataset with time and spatial data
data= {
    'Region' : ['North', 'South', 'East', 'West', 'North', 'South', 'East', 'West']
    'City' :['City1','City2','City3','City4','City1','City2','City3','City4'],
    'Product' : ['A', 'B','A', 'B', 'C', 'C', 'B', 'A' ],
    'Sales' : [150,200,300,400,250,180,220,310],
    'Quantity' : [10,15,20,25,12,14,16,18],
    'Date': pd.to_datetime(['2024-01-01','2024-01-02','2024-02-01','2024-02-03','20
}

```

In [16]:
```
df=pd.DataFrame(data)
print("Extended Dataset:\n" , df)
```

```
Extended Dataset:
   Region   City Product  Sales  Quantity       Date
0  North  City1       A    150        10 2024-01-01
1  South  City2       B    200        15 2024-01-02
2   East  City3       A    300        20 2024-02-01
3   West  City4       B    400        25 2024-02-03
4  North  City1       C    250        12 2024-03-01
5  South  City2       C    180        14 2024-03-02
6   East  City3       B    220        16 2024-04-01
7   West  City4       A    310        18 2024-04-03
```

In [18]:
```
#time aggregraion
#step 2 :set date column as index
df.set_index('Date', inplace=True)
```

In [20]:
```
#aggregrating sales by month
monthly_sales = df.resample('M')['Sales'].sum()
print("\n Total sales by month: \n" , monthly_sales)
```

```
 Total sales by month:
 Date
2024-01-31    350
2024-02-29    700
2024-03-31    430
2024-04-30    530
Freq: M, Name: Sales, dtype: int64
```

In [21]:
```
#aggreagraing sales by quarter
quarterly_sales=df.resample('Q')['Sales'].sum()
print("\n Total sales by quarter:\n ",quarterly_sales)
```

```
 Total sales by quarter:
  Date
2024-03-31    1480
2024-06-30     530
Freq: Q-DEC, Name: Sales, dtype: int64
```

In [22]:
```
#aggreagrating sales by year
yearly_sales = df.resample('Y')['Sales'].sum()
print("\n Total sales by Year" , yearly_sales)
```

```
 Total sales by Year Date
2024-12-31    2010
Freq: A-DEC, Name: Sales, dtype: int64
```

In [23]:
```
#reset index to store original structure
df.reset_index(inplace=True)
```

In [24]:
```
#spatial aggregration
```

In [26]:
```
#step 3: aggragrating sales by region
sales_by_region = df.groupby('Region')['Sales'].sum()
print("\n total sales by region: \n" , sales_by_region)
```

```
 total sales by region:
 Region
East     520
North    400
South    380
West     710
Name: Sales, dtype: int64
```

In [28]:
```
#aggreating sales by city
sales_by_city= df.groupby('City')['Sales'].sum()
print("\n Total sales by city: \n",sales_by_city)
```

```
 Total sales by city:
 City
City1    400
City2    380
City3    520
City4    710
Name: Sales, dtype: int64
```

In [31]:
```
#aggregrating sales by city aND REGION
sales_by_city_and_region = df.groupby(['Region', 'City'])['Sales'].sum()
print("\n sales by city and region: \n", sales_by_city_and_region)
```

```
 sales by city and region:
 Region  City
East    City3    520
North   City1    400
South   City2    380
West    City4    710
Name: Sales, dtype: int64
```

In [32]:
```
#step 4 export spatial aggregration resul;ts
sales_by_city_and_region_reset = sales_by_city_and_region.reset_index()

sales_by_city_and_region_reset.to_csv("Spatial_aggreagraion.csv" , index=False)
print("data saved")0

```

```
data saved
```

In [ ]:
```

```