In [6]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from scipy.cluster.hierarchy import linkage,dendrogram,fcluster
```

In [10]:
```
data = pd.read_csv('sales_data_sample.csv', encoding='latin1')
```

In [11]:
```
print(data.head())
print(data.info())
```

```
   ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  \
0        10107               30      95.70                2  2871.00
1        10121               34      81.35                5  2765.90
2        10134               41      94.74                2  3884.34
3        10145               45      83.26                6  3746.70
4        10159               49     100.00               14  5205.27

         ORDERDATE   STATUS  QTR_ID  MONTH_ID  YEAR_ID  ...  \
0   2/24/2003 0:00  Shipped       1         2     2003  ...
1    5/7/2003 0:00  Shipped       2         5     2003  ...
2    7/1/2003 0:00  Shipped       3         7     2003  ...
3   8/25/2003 0:00  Shipped       3         8     2003  ...
4  10/10/2003 0:00  Shipped       4        10     2003  ...

                   ADDRESSLINE1  ADDRESSLINE2           CITY STATE  \
0        897 Long Airport Avenue           NaN            NYC    NY
1             59 rue de l'Abbaye           NaN          Reims   NaN
2  27 rue du Colonel Pierre Avia           NaN          Paris   NaN
3             78934 Hillside Dr.           NaN       Pasadena    CA
4                7734 Strong St.           NaN  San Francisco    CA

  POSTALCODE COUNTRY TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME DEALSIZE
0      10022     USA       NaN              Yu             Kwai    Small
1      51100  France      EMEA         Henriot             Paul    Small
2      75508  France      EMEA        Da Cunha           Daniel   Medium
3      90003     USA       NaN           Young            Julie   Medium
4        NaN     USA       NaN           Brown            Julie   Medium

[5 rows x 25 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 25 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   ORDERNUMBER       2823 non-null   int64
 1   QUANTITYORDERED   2823 non-null   int64
 2   PRICEEACH         2823 non-null   float64
 3   ORDERLINENUMBER   2823 non-null   int64
 4   SALES             2823 non-null   float64
 5   ORDERDATE         2823 non-null   object
 6   STATUS            2823 non-null   object
 7   QTR_ID            2823 non-null   int64
 8   MONTH_ID          2823 non-null   int64
 9   YEAR_ID           2823 non-null   int64
 10  PRODUCTLINE       2823 non-null   object
 11  MSRP              2823 non-null   int64
 12  PRODUCTCODE       2823 non-null   object
 13  CUSTOMERNAME      2823 non-null   object
 14  PHONE             2823 non-null   object
 15  ADDRESSLINE1      2823 non-null   object
 16  ADDRESSLINE2      302 non-null    object
 17  CITY              2823 non-null   object
 18  STATE             1337 non-null   object
 19  POSTALCODE        2747 non-null   object
```

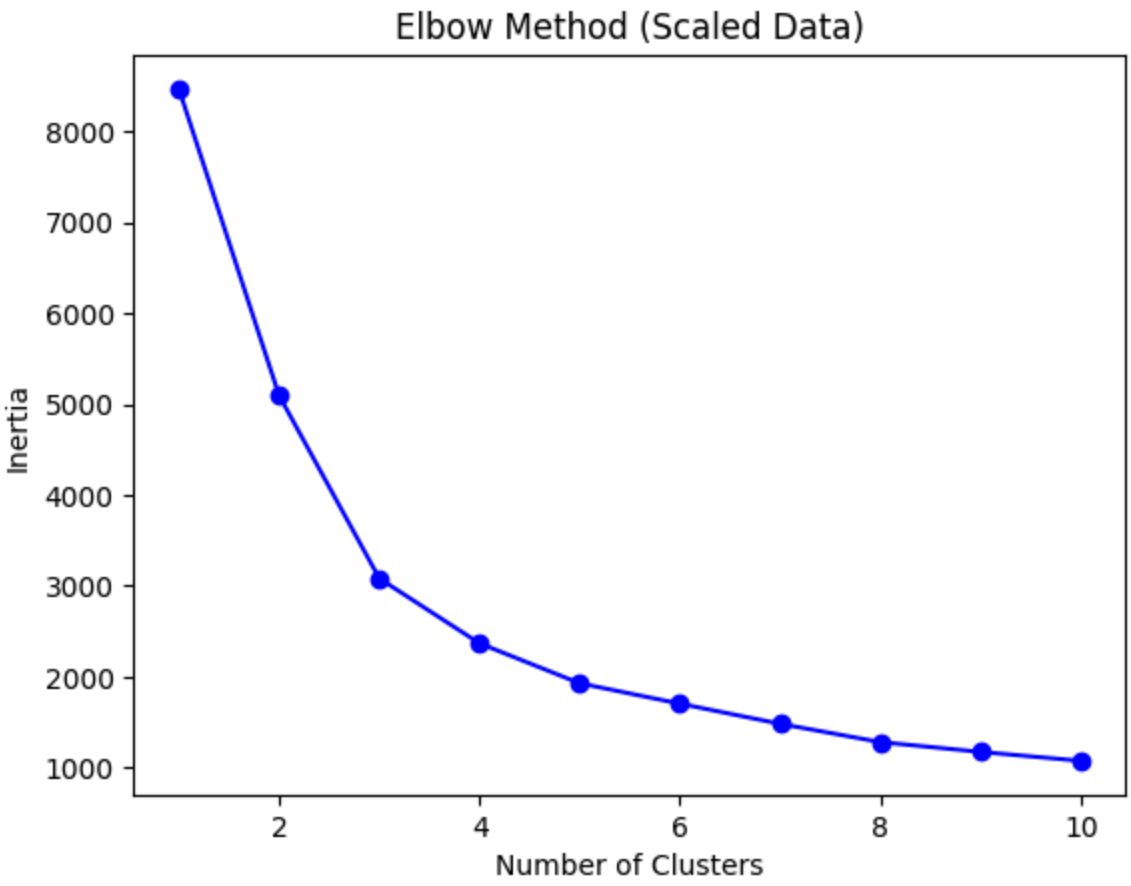
```
 20  COUNTRY           2823 non-null   object
 21  TERRITORY         1749 non-null   object
 22  CONTACTLASTNAME   2823 non-null   object
 23  CONTACTFIRSTNAME  2823 non-null   object
 24  DEALSIZE          2823 non-null   object
dtypes: float64(2), int64(7), object(16)
memory usage: 551.5+ KB
None
```

In [12]:
```
#Select relavent numeric column
df=data[['QUANTITYORDERED','SALES','PRICEEACH']].dropna()
```

In [14]:
```
scalar=StandardScaler()
scalar_data=scalar.fit_transform(df)
```

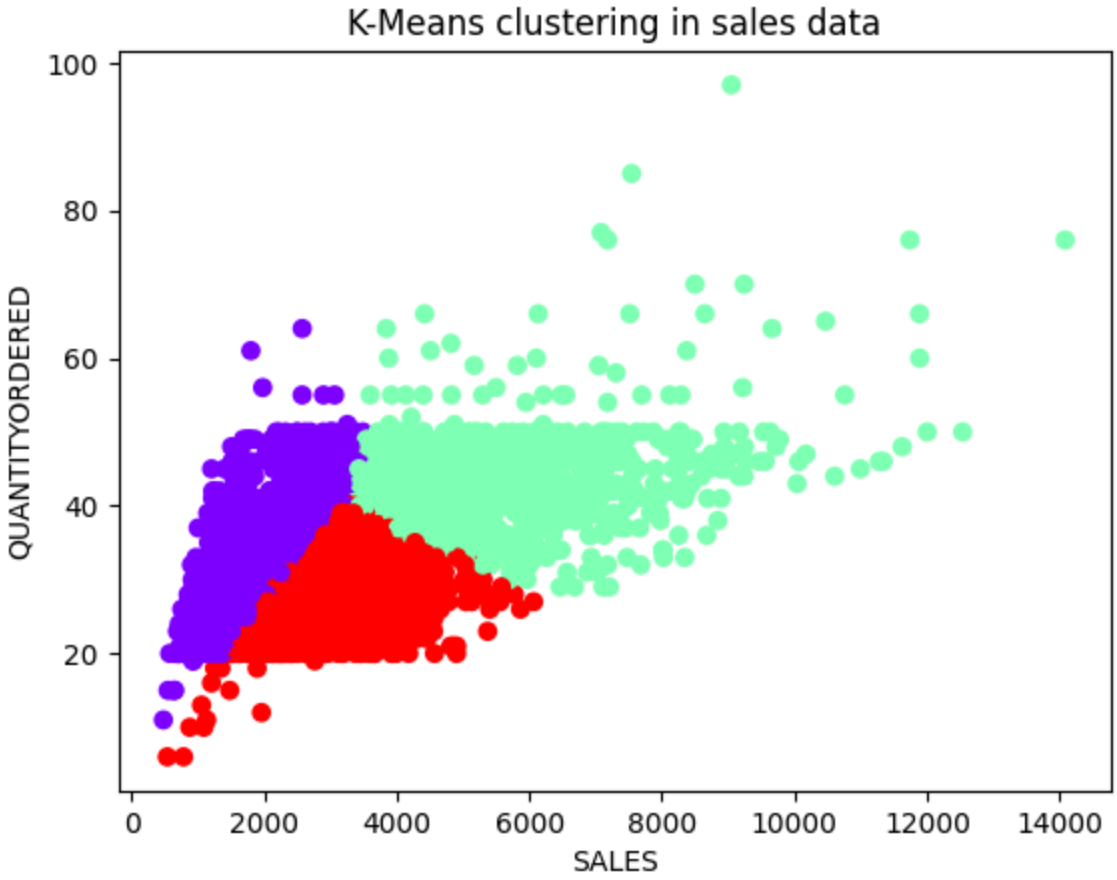
In [18]:
```
#KMeans
inertia=[]
k=range(1,11)
for k in k:
    kmeans=KMeans(n_clusters=k,random_state=42)
    kmeans.fit(scalar_data)
    inertia.append(kmeans.inertia_)
```

In [20]:
```
# 3. Elbow plot
plt.plot(range(1,11), inertia, 'bo-')
plt.title('Elbow Method (Scaled Data)')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.show()
```

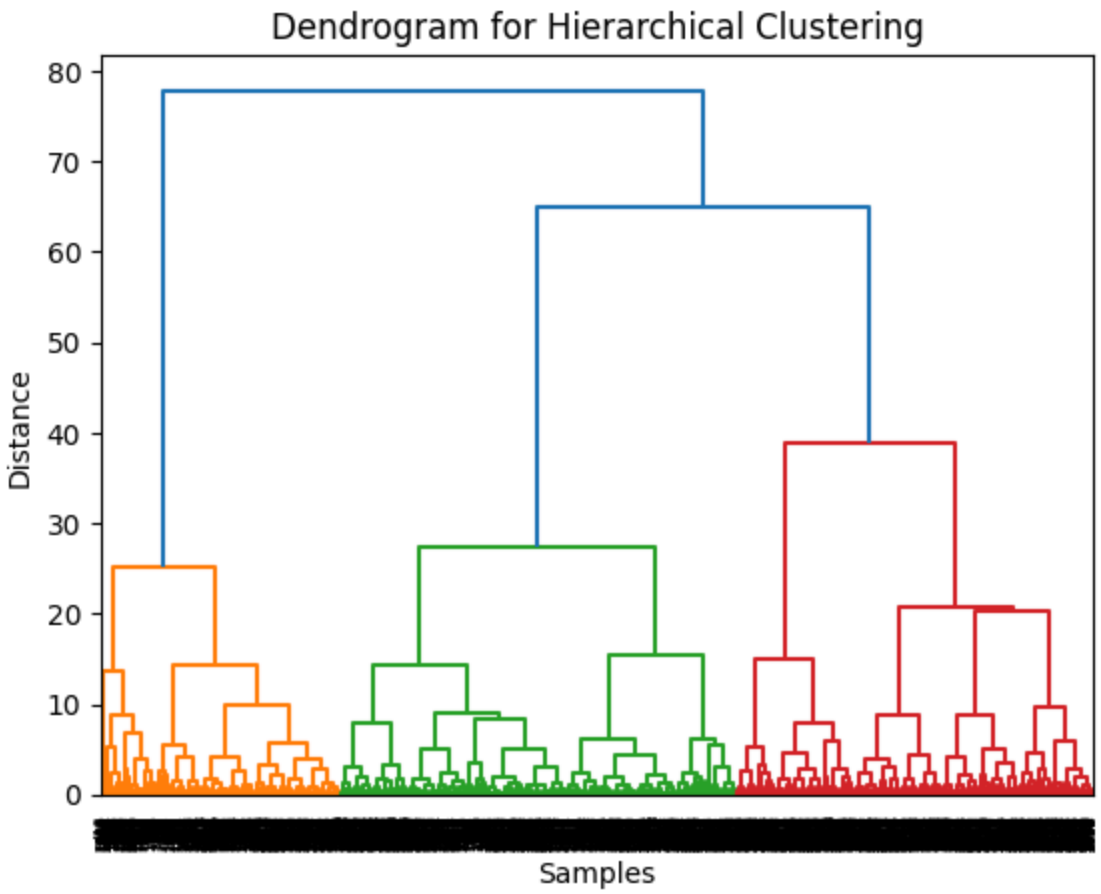


In [21]:
```
#fit k with optimal cluster
kmeans =KMeans(n_clusters=3,random_state=42)
clusters = kmeans.fit_predict(scalar_data)
df['cluster'] =clusters
```

In [25]:
```
#visualize cluster
plt.scatter(df['SALES'],df['QUANTITYORDERED'],c=df['cluster'],cmap='rainbow')
plt.xlabel('SALES')
plt.ylabel('QUANTITYORDERED')
plt.title('K-Means clustering in sales data')
plt.show()
```



In [27]:
```
#hierarchical Clutering
linked=linkage(scalar_data,method='ward')
dendrogram(linked)
plt.title("Dendrogram for Hierarchical Clustering")
plt.xlabel('Samples')
plt.ylabel('Distance')
plt.show()
```



In [29]:
```
#Assign cluster
cluster_h=fcluster(linked,3,criterion='maxclust')
df['Hierarchical_cluster'] = cluster_h
print(df.head())
```

```
   QUANTITYORDERED    SALES  PRICEEACH  cluster  Hierarchical_cluster
0               30  2871.00      95.70        2                     2
1               34  2765.90      81.35        2                     2
2               41  3884.34      94.74        1                     1
3               45  3746.70      83.26        1                     3
4               49  5205.27     100.00        1                     1
```

In [ ]: