In [1]:
```
#Given a data frame containing information about customer's purchase history,
#find the following information:

#The top 5 most frequent items purchased by customers.
#The average number of items purchased per transaction.
#The total revenue generated by each item.
#The total revenue generated by the top 10% of customers (based on total revenue generated).
#The percentage of revenue generated by each item category.
```

In [2]:
```
import pandas as pd

# Create a sample dataframe
data = {'customer_id': [1,1,1,2,2,2,3,3,3,3,4,4,4,4],
        'item_name': ['item1', 'item2', 'item3', 'item4', 'item5',
                      'item6', 'item1', 'item7', 'item8', 'item9',
                      'item10', 'item11', 'item12', 'item13'],
        'item_price': [10, 20, 15, 25, 30, 20,25,30,15,25,30,20,15,25],
        'quantity': [2, 3, 1, 1, 2, 3,2,1,1,1,1,1,1,1],
        'category': ['cat1', 'cat2', 'cat1', 'cat3', 'cat2', 'cat3',
                     'cat1', 'cat2', 'cat3', 'cat3','cat1', 'cat2', 'cat3', 'cat3']}
df = pd.DataFrame(data)
```

In [3]:
```
df
```

Out[3]:

	customer_id	item_name	item_price	quantity	category
0	1	item1	10	2	cat1
1	1	item2	20	3	cat2
2	1	item3	15	1	cat1
3	2	item4	25	1	cat3
4	2	item5	30	2	cat2
5	2	item6	20	3	cat3
6	3	item1	25	2	cat1
7	3	item7	30	1	cat2
8	3	item8	15	1	cat3
9	3	item9	25	1	cat3
10	4	item10	30	1	cat1
11	4	item11	20	1	cat2
12	4	item12	15	1	cat3
13	4	item13	25	1	cat3

In [11]:
```
#The top 5 most frequent items purchased by customers.
df1 = df.groupby('item_name')['quantity'].sum()

# find the top 5 most frequent items purchased by customers
top_5_items = df1.nlargest(5).reset_index()
top_5_items
```

Out[11]:

	item_name	quantity
0	item1	4
1	item2	3
2	item6	3
3	item5	2
4	item10	1

In [12]:
```
#The average number of items purchased per transaction.
# group the data by customer_id
# and find the average number of items per transaction
avg_items_per_transaction = df.groupby(
    'customer_id')['quantity'].mean()
print(avg_items_per_transaction)
```

```
customer_id
1    2.00
2    2.00
3    1.25
4    1.00
Name: quantity, dtype: float64
```

In [13]:
```
#The total revenue generated by each item.
df
```

Out[13]:

	customer_id	item_name	item_price	quantity	category
0	1	item1	10	2	cat1
1	1	item2	20	3	cat2
2	1	item3	15	1	cat1
3	2	item4	25	1	cat3
4	2	item5	30	2	cat2
5	2	item6	20	3	cat3
6	3	item1	25	2	cat1
7	3	item7	30	1	cat2
8	3	item8	15	1	cat3
9	3	item9	25	1	cat3
10	4	item10	30	1	cat1
11	4	item11	20	1	cat2
12	4	item12	15	1	cat3
13	4	item13	25	1	cat3

In [14]:
```
df['revenue'] = df['quantity']*df['item_price']
df
```

Out[14]:

	customer_id	item_name	item_price	quantity	category	revenue
0	1	item1	10	2	cat1	20
1	1	item2	20	3	cat2	60
2	1	item3	15	1	cat1	15
3	2	item4	25	1	cat3	25
4	2	item5	30	2	cat2	60
5	2	item6	20	3	cat3	60
6	3	item1	25	2	cat1	50
7	3	item7	30	1	cat2	30
8	3	item8	15	1	cat3	15
9	3	item9	25	1	cat3	25
10	4	item10	30	1	cat1	30
11	4	item11	20	1	cat2	20
12	4	item12	15	1	cat3	15
13	4	item13	25	1	cat3	25

In [15]:
```
#The total revenue generated by each item.
df4 = df.pivot_table(
    index='item_name', values='revenue', aggfunc='sum')
print(df4)
#or
df5 = df.groupby('item_name')['revenue'].sum().reset_index()
df5
```

```
           revenue
item_name
item1           70
item10          30
item11          20
item12          15
item13          25
item2           60
item3           15
item4           25
item5           60
item6           60
item7           30
item8           15
item9           25
```

Out[15]:

	item_name	revenue
0	item1	70
1	item10	30
2	item11	20
3	item12	15
4	item13	25
5	item2	60
6	item3	15
7	item4	25
8	item5	60
9	item6	60
10	item7	30
11	item8	15
12	item9	25

In [16]:
```
#total revenue generated by the top 10% of customers (based on total revenue generated).
# group the data by customer_id
# and find the total revenue generated by each customer
dff = df.groupby(
    'customer_id')['revenue'].sum().reset_index()
dff
```

Out[16]:

	customer_id	revenue
0	1	95
1	2	145
2	3	120
3	4	90

In [18]:
```
# sort the data by revenue in descending order
#df7 = df7.sort_values('revenue', ascending=False)
#df7
```

In [19]:
```
# Get top 10% customers based on the revenue generated
top_10_percent_customers = dff.nlargest(
    int(len(dff)*0.5), 'revenue')
top_10_percent_customers

# Find the total revenue generated by these customers
top_10_percent_revenue = top_10_percent_customers['revenue'].sum()
print(top_10_percent_revenue)
```

```
265
```