

## SCRM Technical Interview

Please refer to tables 1 to 3 at the bottom of this sheet to complete the assignments.

### Assignment 1 (SQL)

- 1.1 Write a SQL query that shows which product categories represent the largest share of the total paid amount given by month. The query should return the total amount paid by category as well as the share of the total for all orders.

```
select productCategoryName, sum(paidAmount) as totalPaidAmount, sum(paidAmount) /  
(select sum(paidAmount) from table1) as shareOfTotal  
from table1  
join table2 on table1.productID = table2.productID  
group by productCategoryName  
order by totalPaidAmount desc;
```

- 1.2 Write a SQL query that shows which categories that are measured in weight are less frequently sold with any discount.

```
select productCategoryName, count(*) as totalOrders, sum(discountedAmount) as  
totalDiscountedAmount  
from table1  
join table2 on table1.productID = table2.productID  
join table3 on table1.productID = table3.productID  
where isWeightProduct = true  
group by productCategoryName  
having totalDiscountedAmount = 0  
order by totalOrders desc;
```

- 1.3 Write a SQL query that shows which products and categories are normally part of larger orders (orders with more products or higher amount charged - whatever you think makes more sense).

```
select productCategoryName, productID, count(*) as totalOrders, sum(unitsSold) as  
totalUnitsSold, sum(paidAmount) as totalPaidAmount  
from table1  
join table2 on table1.productID = table2.productID  
group by productCategoryName, productID  
having totalUnitsSold > 10 or totalPaidAmount > 100  
order by totalUnitsSold desc, totalPaidAmount desc;
```

### Assignment 2 (Python)

Assume the same tables are also available as pandas dataframes. Show the same results as for assignment 1 but using Python instead of SQL.

## Import pandas as pd

```
table1 = pd.read_csv('table1.csv')
table2 = pd.read_csv('table2.csv')
table3 = pd.read_csv('table3.csv')

df1 = pd.merge(table1, table2, on='productID')
df1['month'] = pd.DatetimeIndex(df1['date']).month
df1['total'] = df1['paidAmount'] * df1['unitsSold']
df1['share'] = df1['total'] / df1['total'].sum()
df1 = df1.groupby(['month', 'productCategoryName'])['total', 'share'].sum()
df1 = df1.sort_values(by='share', ascending=False)

df2 = pd.merge(table1, table2, on='productID')
df2 = pd.merge(df2, table3, on='productID')
df2 = df2[df2['isWeightProduct'] == True]
df2 = df2[df2['discountedAmount'] == 0]
df2 = df2.groupby(['productCategoryName'])['discountedAmount'].count()
df2 = df2.sort_values(ascending=False)

df3 = pd.merge(table1, table2, on='productID')
df3['total'] = df3['paidAmount'] * df3['unitsSold']
df3 = df3.groupby(['productID', 'productCategoryName'])['total'].sum()
df3 = df3.sort_values(ascending=False)
df3 = df3[df3 > 100]
```

Table 1

OrderID	Date	ProductID	PaidAmount	DiscountedAmount	UnitsSold
39230232	2020-02-20	D231	12.29	null	3
...	...	...	...	...	...
...	...	...	...	...	...

Table 2

ProductID	ProductCategoryName	ProductCategoryID
D231	Tomatoes, 500g Pack	002
...	...	...
...	...	...

Table 3

ProductID	isWeightProduct	isAlcoholicProduct	isActiveProduct
D231	False	False	true
...	...	...	...
...	...	...	...