Project Title: E-Commerce Customer Churn Analysis

Coding Mentor: Janani Lakshmi Narayanan

Student Name : Parameswari Manthiramoorthi

SEPTEMBER 18, 2024

ENTRI ELEVATE

TN-DSML-MAY24-BATCH4

Table of Contents

1. Introduction

E-commerce businesses rely on customer satisfaction and retention to sustain profitability. One of the major challenges in this domain is identifying patterns that lead to customer churn. Analyzing customer data can provide crucial insights to help e-commerce companies reduce churn rates by implementing targeted retention strategies.

2. Problem Statement

The goal of this project is to understand the factors leading to customer churn in the e-commerce industry. By analyzing attributes such as tenure, preferred devices, payment methods, and purchase behaviors, this project seeks to provide actionable insights to help mitigate customer attrition.

3. Dataset Overview

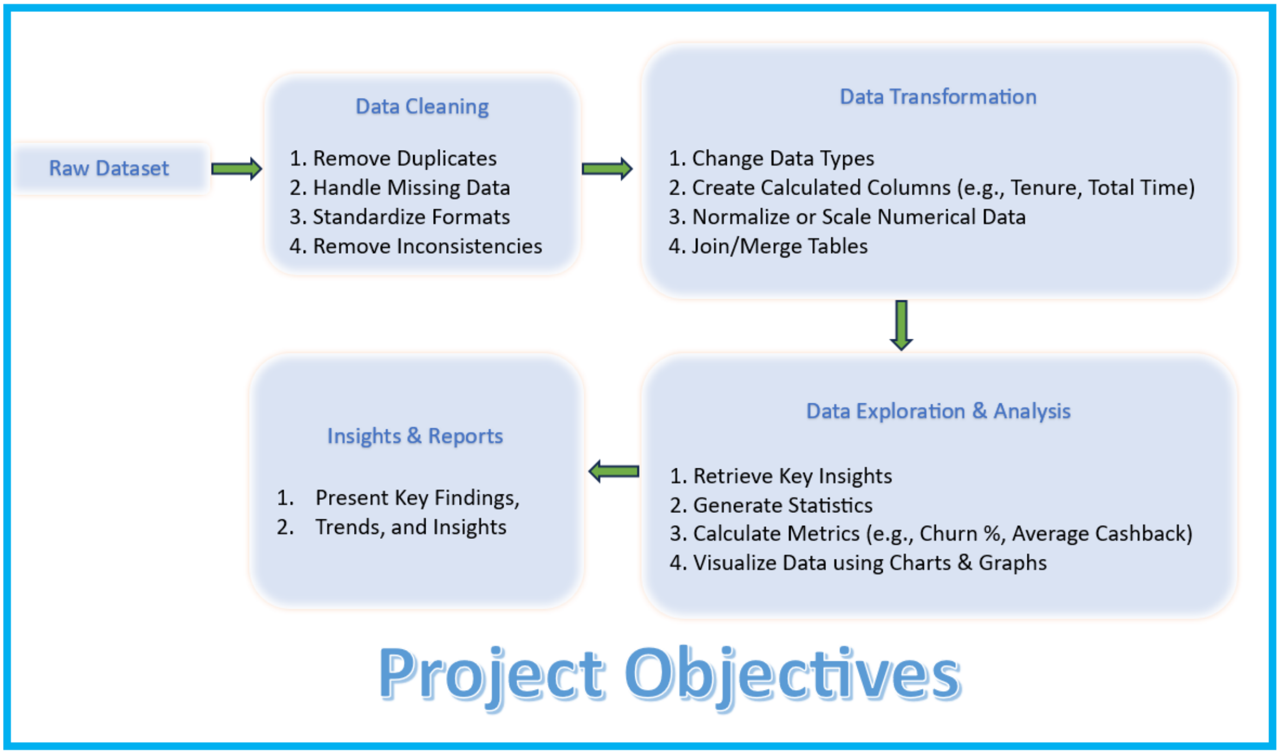
The dataset contains the following fields:

- **CustomerID**: Unique identifier for each customer.
- **Churn**: Whether a customer churned (1 = Churned, 0 = Active).
- **Tenure**: Number of months the customer has been with the company.
- **PreferredLoginDevice**: The device the customer most often uses to log in.
- **CityTier**: Tier of the city the customer lives in.
- **WarehouseToHome**: Distance from warehouse to the customer's home in km.
- **PreferredPaymentMode**: Mode of payment frequently used by the customer.
- **Gender**: Customer's gender.
- **HourSpendOnApp**: Hours spent by the customer on the app.
- **NumberOfDeviceRegistered**: Number of devices registered by the customer.
- **PreferedOrderCat**: Preferred order category of the customer.
- **SatisfactionScore**: Satisfaction score out of 5.
- **MaritalStatus**: Customer's marital status.
- **NumberOfAddress**: Number of addresses provided by the customer.
- **Complain**: Whether the customer complained (1 = Yes, 0 = No).
- **OrderAmountHikeFromlastYear**: Order amount increase from the previous year.
- **CouponUsed**: Number of coupons used by the customer.
- **OrderCount**: Number of orders placed by the customer.
- **DaySinceLastOrder**: Number of days since the last order.
- **CashbackAmount**: Total cashback amount received by the customer.

4. Project Objectives

The main objectives of the project are:

- Clean and transform the data to make it suitable for analysis.
- Identify and handle missing values and outliers.
- Analyze the factors contributing to customer churn.
- Provide actionable insights to reduce churn.



Raw Dataset

Data Cleaning

1. Remove Duplicates
2. Handle Missing Data
3. Standardize Formats
4. Remove Inconsistencies

Data Transformation

1. Change Data Types
2. Create Calculated Columns (e.g., Tenure, Total Time)
3. Normalize or Scale Numerical Data
4. Join/Merge Tables

Data Exploration & Analysis

1. Retrieve Key Insights
2. Generate Statistics
3. Calculate Metrics (e.g., Churn %, Average Cashback)
4. Visualize Data using Charts & Graphs

Insights & Reports

1. Present Key Findings,
2. Trends, and Insights

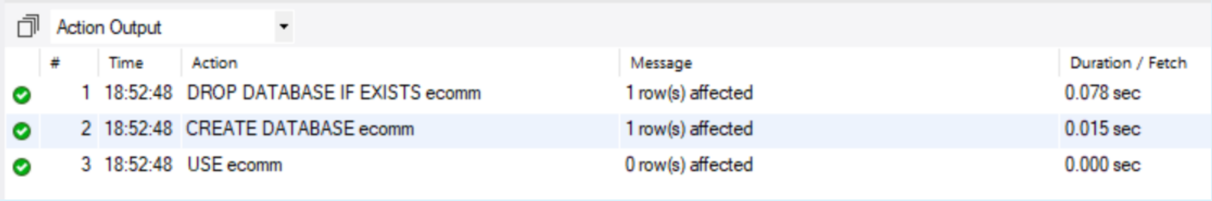
Project Objectives

5. Step-by-Step Implementation

Step 1: Create Database

1.1 Create Database & Table

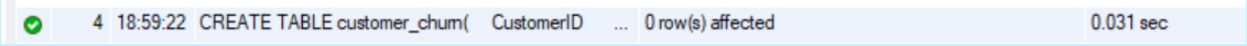
```
/* check database if already exists */
DROP DATABASE IF EXISTS ecomm;
/* create database Named  ecomm */
CREATE DATABASE ecomm;
/*use ecomm database*/
USE ecomm;
```



```
Action Output
     #   Time      Action                          Message               Duration / Fetch
✓    1   18:52:48  DROP DATABASE IF EXISTS ecomm   1 row(s) affected     0.078 sec
✓    2   18:52:48  CREATE DATABASE ecomm           1 row(s) affected     0.015 sec
✓    3   18:52:48  USE ecomm                       0 row(s) affected     0.000 sec
```

1.2 Create Table

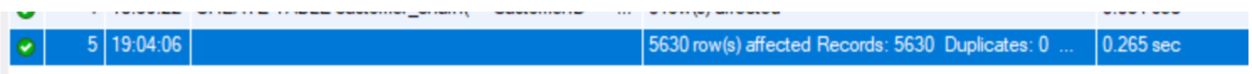
```
/* create table named customer churn*/
CREATE TABLE customer_churn( CustomerID   INT  PRIMARY KEY, Churn    BIT, Tenure
INT,    PreferredLoginDevice       VARCHAR(20), CityTier            INT, WarehouseToHome
INT,    PreferredPaymentMode       VARCHAR(20),Gender
ENUM('Male','Female'),    HourSpendOnApp             INT,  NumberOfDeviceRegistered    INT,
PreferedOrderCat           VARCHAR(20),   SatisfactionScore          INT, MaritalStatus
VARCHAR(10),NumberOfAddress            INT,    Complain        BIT,
OrderAmountHikeFromlastYear       INT, CouponUsed      INT, OrderCount            INT,
DaySinceLastOrder          INT, CashbackAmount            INT );
```



```
✓    4   18:59:22  CREATE TABLE customer_churn(   CustomerID   ...   0 row(s) affected     0.031 sec
```

1.3 Insert values

❖ insert values into customer_churn table



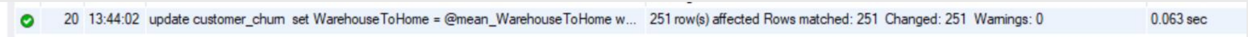
```
✓    5   19:04:06        5630 row(s) affected Records: 5630  Duplicates: 0  ...     0.265 sec
```

Step 2: Data Cleaning

2.1 Handling Missing Values and Outliers:

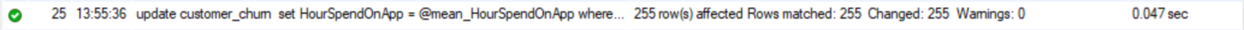
- Impute mean values for WarehouseToHome, HourSpendOnApp, OrderAmountHikeFromlastYear, and DaySinceLastOrder and round them off to the nearest integer.

```
set @mean_WarehouseToHome= (select round(avg(WarehouseToHome)) from
customer_churn);
update customer_churn
set WarehouseToHome = @mean_WarehouseToHome
where WarehouseToHome is Null;
```



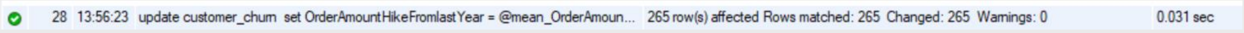
```
20  13:44:02  update customer_chum  set WarehouseToHome = @mean_WarehouseToHome w...  251 row(s) affected Rows matched: 251  Changed: 251  Wamings: 0  0.063 sec
```

```
set @mean_HourSpendOnApp= (select round(avg(HourSpendOnApp)) from
customer_churn);
update customer_churn
set HourSpendOnApp = @mean_HourSpendOnApp
where HourSpendOnApp is Null;
```



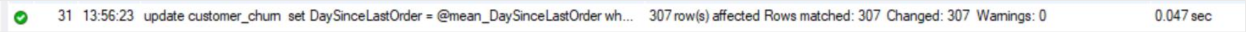
```
25  13:55:36  update customer_chum  set HourSpendOnApp = @mean_HourSpendOnApp where...  255 row(s) affected Rows matched: 255  Changed: 255  Wamings: 0  0.047 sec
```

```
set @mean_OrderAmountHikeFromlastYear= (select
round(avg(OrderAmountHikeFromlastYear)) from  customer_churn);
update customer_churn
set OrderAmountHikeFromlastYear = @mean_OrderAmountHikeFromlastYear
where OrderAmountHikeFromlastYear is Null;
```



```
28  13:56:23  update customer_chum  set OrderAmountHikeFromlastYear = @mean_OrderAmoun...  265 row(s) affected Rows matched: 265  Changed: 265  Wamings: 0  0.031 sec
```

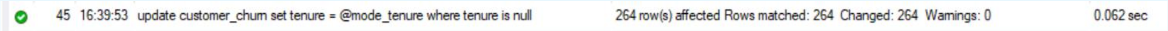
```
set @mean_DaySinceLastOrder= (select round(avg(DaySinceLastOrder)) from
customer_churn);
update customer_churn
set DaySinceLastOrder = @mean_DaySinceLastOrder
where DaySinceLastOrder is Null;
```



```
31  13:56:23  update customer_chum  set DaySinceLastOrder = @mean_DaySinceLastOrder wh...  307 row(s) affected Rows matched: 307  Changed: 307  Wamings: 0  0.047 sec
```

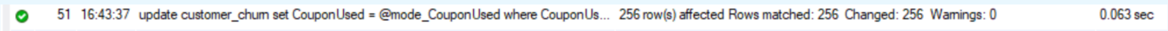
- Impute mode for Tenure, CouponUsed, and OrderCount.

```
select * from customer_churn where tenure is null;
set @mode_tenure=(select (count(*)) count from customer_churn where
tenure is not null group by tenure order by count desc limit 1);

update customer_churn
set tenure = @mode_tenure
where tenure is null;
```



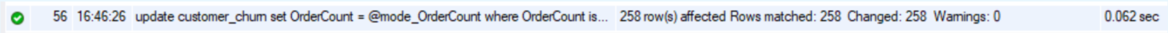
```
45  16:39:53  update customer_churn set tenure = @mode_tenure where tenure is null    264 row(s) affected Rows matched: 264  Changed: 264  Warnings: 0    0.062 sec
```

```
select * from customer_churn where CouponUsed is null;
set @mode_ CouponUsed =(select (count(*)) count from customer_churn
where CouponUsed is not null group by CouponUsed order by count desc
limit 1);

update customer_churn
set CouponUsed = @mode_ CouponUsed
where CouponUsed is null;
```



```
51  16:43:37  update customer_churn set CouponUsed = @mode_CouponUsed where CouponUs...  256 row(s) affected Rows matched: 256  Changed: 256  Warnings: 0    0.063 sec
```

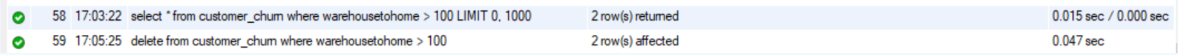
```
select * from customer_churn where OrderCount is null;
set @mode_OrderCount =(select (count(*)) count from customer_churn
where OrderCount is not null group by OrderCount order by count desc limit
1);

update customer_churn
set OrderCount = @mode_OrderCount
where OrderCount is null;
```



```
56  16:46:26  update customer_churn set OrderCount = @mode_OrderCount where OrderCount is...  258 row(s) affected Rows matched: 258  Changed: 258  Warnings: 0    0.062 sec
```

- Remove outliers in the WarehouseToHome column (i.e., remove rows where values > 100).

```
select * from customer_churn where warehousetohome > 100;
delete from customer_churn where warehousetohome > 100;
```

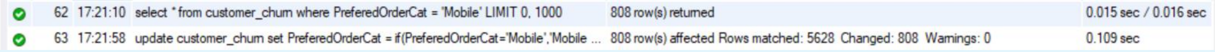


```
58  17:03:22  select * from customer_churn where warehousetohome > 100 LIMIT 0, 1000    2 row(s) returned    0.015 sec / 0.000 sec
59  17:05:25  delete from customer_churn where warehousetohome > 100                   2 row(s) affected    0.047 sec
```

2.2 Dealing with Inconsistencies:

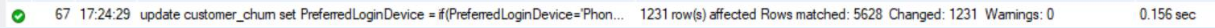
- ❖ Standardize the PreferredLoginDevice and PreferredOrderCat columns by replacing inconsistent entries (e.g., “Phone” should be “Mobile Phone”).

```
select * from customer_churn where PreferedOrderCat = 'Mobile';
update customer_churn
set PreferedOrderCat = if(PreferedOrderCat='Mobile','Mobile
Phone',PreferedOrderCat) ;
```



```
62  17:21:10  select * from customer_churn where PreferedOrderCat = 'Mobile' LIMIT 0, 1000          808 row(s) returned                                                    0.015 sec / 0.016 sec
63  17:21:58  update customer_churn set PreferedOrderCat = if(PreferedOrderCat='Mobile','Mobile ...  808 row(s) affected Rows matched: 5628  Changed: 808  Warnings: 0     0.109 sec
```

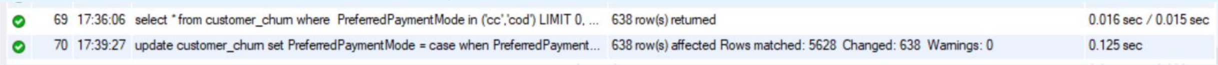
```
select * from customer_churn where PreferredLoginDevice = 'Phone';
update customer_churn
set PreferredLoginDevice = if(PreferredLoginDevice='Phone','Mobile
Phone',PreferredLoginDevice) ;
```



```
67  17:24:29  update customer_churn set PreferredLoginDevice = if(PreferredLoginDevice='Phon...  1231 row(s) affected Rows matched: 5628  Changed: 1231  Warnings: 0     0.156 sec
```

- ❖ Replace "COD" with "Cash on Delivery" and "CC" with "Credit Card".

```
select * from customer_churn where  PreferredPaymentMode in ('cc','cod') ;
update customer_churn
set PreferredPaymentMode = case
                    when PreferredPaymentMode =  'cc' then 'Credit Card'
                    when PreferredPaymentMode = 'cod' then 'Cash On Delivery'
                    else PreferredPaymentMode
                              end;
```



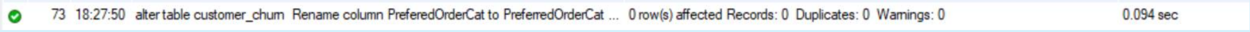
```
69  17:36:06  select * from customer_churn where  PreferredPaymentMode in ('cc','cod') LIMIT 0, ...  638 row(s) returned                                                    0.016 sec / 0.015 sec
70  17:39:27  update customer_churn set PreferredPaymentMode = case when PreferredPayment...      638 row(s) affected Rows matched: 5628  Changed: 638  Warnings: 0     0.125 sec
```

Step 3: Data Transformation

3.1 Column Renaming:

- Rename PreferedOrderCat to PreferredOrderCat and HourSpendOnApp to HoursSpentOnApp.

```
alter table customer_churn
 Rename column PreferedOrderCat to PreferredOrderCat ,
 Rename column HourSpendOnApp to HoursSpentOnApp ;
```



```
73  18:27:50  alter table customer_churn  Rename column PreferedOrderCat to PreferredOrderCat ...  0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0    0.094 sec
```

3.2 Creating New Columns:

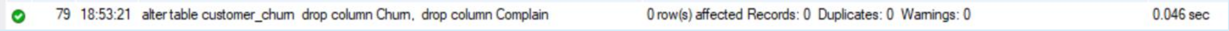
- Create a column ComplaintReceived based on the Complain column, marking "Yes" for complaints and "No" otherwise.
- Create a column ChurnStatus, setting it to "Churned" if the Churn column is 1 and "Active" if 0.

```
alter table  customer_churn
 add column ComplaintReceived enum('Yes','No'),
 add column ChurnStatus enum('Churned','Active');


 update customer_churn
 set ComplaintReceived = if(complain = 1,'Yes','No'),
  ChurnStatus = if(churn = 1,'Churned','Active');
```

3.3 Dropping Redundant Columns:

- Drop the Churn and Complain columns as they have been replaced.

```
alter table customer_churn
 drop column Churn,
 drop column Complain;
```

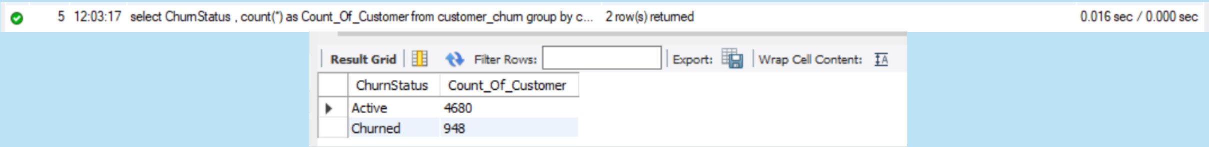


```
79  18:53:21  alter table customer_churn  drop column Churn,  drop column Complain    0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0    0.046 sec
```

Step 4 : Data Exploration and Analysis

1. Retrieve the count of churned and active customers from the dataset.

```
select ChurnStatus , count(*) as Count_Of_Customer from customer_churn group by
churnstatus order by ChurnStatus desc ;
```



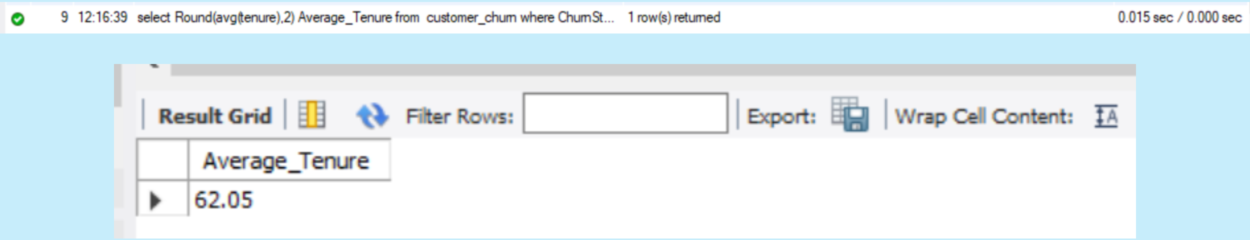
```
5  12:03:17  select ChurnStatus , count(*) as Count_Of_Customer from customer_churn group by c...  2 row(s) returned  0.016 sec / 0.000 sec
```

```
  ChurnStatus  Count_Of_Customer
▶ Active       4680
  Churned      948
```

Explanation: This query groups the customers by their churn status and counts the number of customers in each group, providing a total count of churned and active customers.

2. Display the average tenure of customers who churned.

```
select Round(avg(tenure),2) Average_Tenure from  customer_churn where ChurnStatus =
'Churned' ;
```



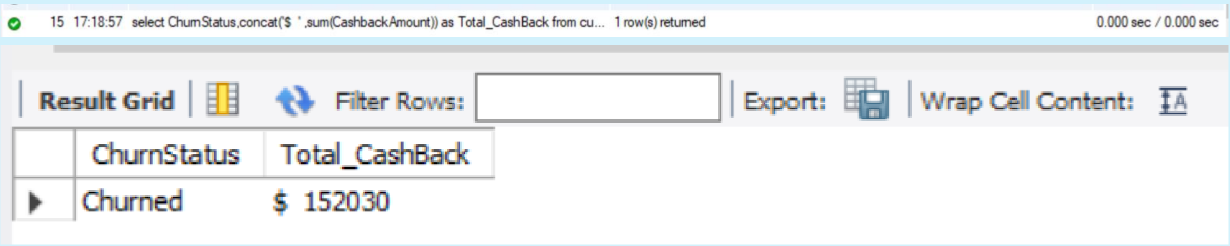
```
9  12:16:39  select Round(avg(tenure),2) Average_Tenure from  customer_churn where ChurnSt...  1 row(s) returned  0.015 sec / 0.000 sec
```

```
  Average_Tenure
▶ 62.05
```

Explanation: This query calculates the average tenure for churned customers by filtering on the churn status and then averaging the Tenure column.

3. Calculate the total cashback amount earned by customers who churned**.**

```
select ChurnStatus,concat('$  ' ,sum(CashbackAmount)) as Total_CashBack from
customer_churn where ChurnStatus = 'Churned';
```



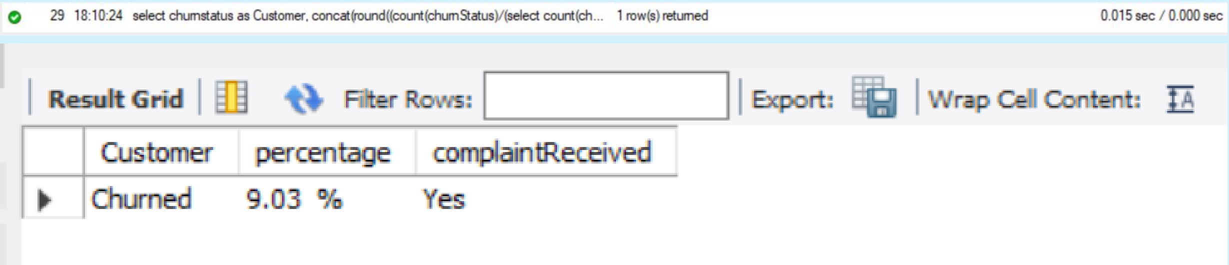
```
15  17:18:57  select ChurnStatus,concat('$  ' ,sum(CashbackAmount)) as Total_CashBack from cu...  1 row(s) returned   0.000 sec / 0.000 sec
```

ChurnStatus	Total_CashBack
Churned	$ 152030

Explanation: This query sums up the CashbackAmount for all customers who have churned.

4. Determine the percentage of churned customers who complained**.**

```
select churnstatus as Customer, concat(round((count(churnStatus)/(select count(churnstatus) from
customer_churn) * 100),2) ,'  % ')as percentage, complaintReceived from customer_churn where
churnstatus = 'Churned' group by ComplaintReceived having ComplaintReceived = 'Yes' ;
```



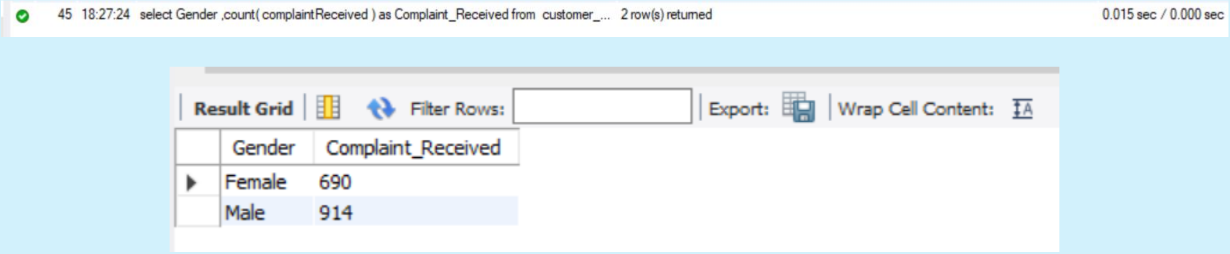
```
29  18:10:24  select churnstatus as Customer, concat(round((count(churnStatus)/(select count(ch...  1 row(s) returned   0.015 sec / 0.000 sec
```

Customer	percentage	complaintReceived
Churned	9.03 %	Yes

Explanation: This query calculates the percentage of churned customers who have complained by dividing the count of complaints by the total number of churned customers.

5. Find the gender distribution of customers who complained**.**

```
select Gender ,count( complaintReceived ) as Complaint_Received from  customer_churn
where complaintReceived ='yes' group by Gender  ;
```



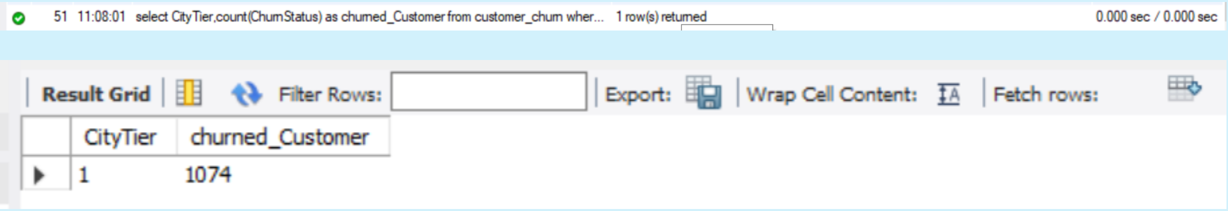
```
45  18:27:24  select Gender ,count( complaintReceived ) as Complaint_Received from  customer_...  2 row(s) returned  0.015 sec / 0.000 sec
```

Gender	Complaint_Received
Female	690
Male	914

Explanation: This query groups the customers who complained by gender and counts the number of complaints per gender.

6. Identify the city tier with the highest number of churned customers whose preferred order category is Laptop & Accessory**.**

```
select CityTier,count(ChurnStatus) as churned_Customer from customer_churn where
PreferredOrderCat = 'Laptop & Accessory' group by CityTier order by churned_Customer desc
limit 1;
```



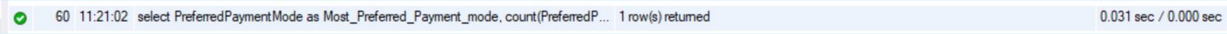
```
51  11:08:01  select CityTier,count(ChurnStatus) as churned_Customer from customer_churn wher...  1 row(s) returned  0.000 sec / 0.000 sec
```

CityTier	churned_Customer
1	1074

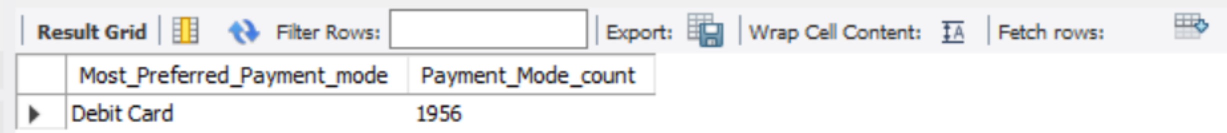
Explanation: This query finds the city tier with the highest number of churned customers who preferred the order category "Laptop & Accessory."

7. Identify the most preferred payment mode among active customers.

```
select PreferredPaymentMode as Most_Preferred_Payment_mode,
count(PreferredPaymentMode) as Payment_Mode_count from customer_churn where
ChurnStatus = 'Active'  group by PreferredPaymentMode order by Payment_Mode_count desc
limit 1;
```



```
60  11:21:02  select PreferredPaymentMode as Most_Preferred_Payment_mode, count(PreferredP...  1 row(s) returned  0.031 sec / 0.000 sec
```

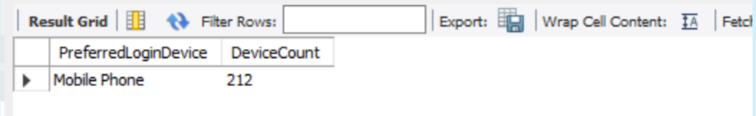


Most_Preferred_Payment_mode	Payment_Mode_count
Debit Card	1956

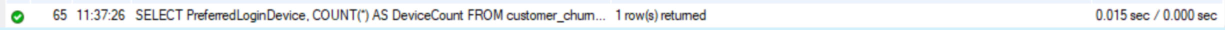
Explanation: This query finds the payment mode that is most frequently used by active customers.

8. List the preferred login device(s) among customers who took more than 10 days since their last order.

```
SELECT PreferredLoginDevice, COUNT(*) AS DeviceCount FROM customer_churn
WHERE DaySinceLastOrder > 10 GROUP BY PreferredLoginDevice ORDER BY
DeviceCount DESC limit 1 ;
```



PreferredLoginDevice	DeviceCount
Mobile Phone	212

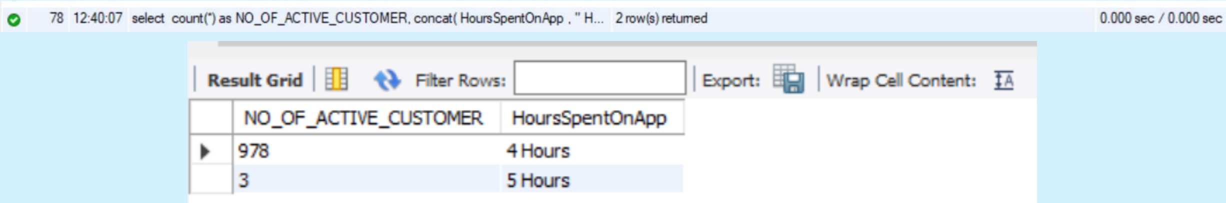


```
65  11:37:26  SELECT PreferredLoginDevice, COUNT(*) AS DeviceCount FROM customer_churn...  1 row(s) returned  0.015 sec / 0.000 sec
```

Explanation: This query lists the preferred login devices of customers who have not placed an order in the last 10 days or more.

9. List the number of active customers who spent more than 3 hours on the app.

```
select  count(*) as NO_OF_ACTIVE_CUSTOMER, concat( HoursSpentOnApp , " Hours")  as
HoursSpentOnApp  from customer_churn where ChurnStatus = 'Active' and
HoursSpentOnApp > 3 group by HoursSpentOnApp;
```



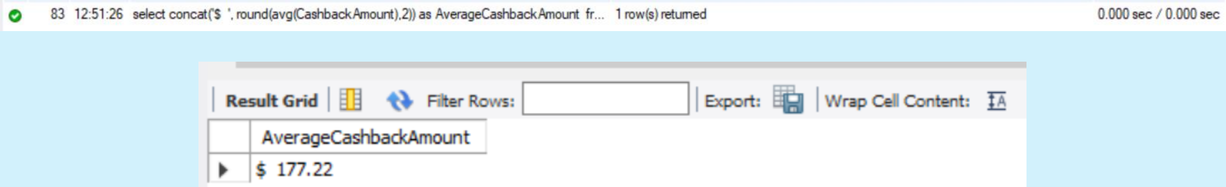
```
78  12:40:07  select  count(*) as NO_OF_ACTIVE_CUSTOMER, concat( HoursSpentOnApp , " H...  2 row(s) returned  0.000 sec / 0.000 sec
```

NO_OF_ACTIVE_CUSTOMER	HoursSpentOnApp
978	4 Hours
3	5 Hours

Explanation: This query counts the number of active customers who spent more than 3 hours on the app.

10. Find the average cashback amount received by customers who spent at least 2 hours on the app.

```
select concat('$  ', round(avg(CashbackAmount),2)) as AverageCashbackAmount  from
customer_churn where HoursSpentOnApp >= 2;
```



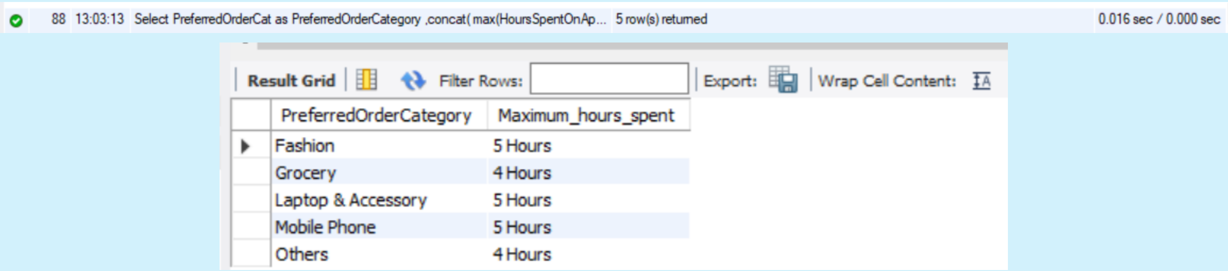
```
83  12:51:26  select concat('$  ', round(avg(CashbackAmount),2)) as AverageCashbackAmount  fr...  1 row(s) returned  0.000 sec / 0.000 sec
```

AverageCashbackAmount
$ 177.22

Explanation: This query calculates the average cashback amount for customers who spent at least 2 hours on the app.

11. Display the maximum hours spent on the app by customers in each preferred order category.

```
Select PreferredOrderCat as PreferredOrderCategory ,concat( max(HoursSpentOnApp), '
Hours') as Maximum_hours_spent from customer_churn group by PreferredOrderCat order by
PreferredOrderCategory ;
```



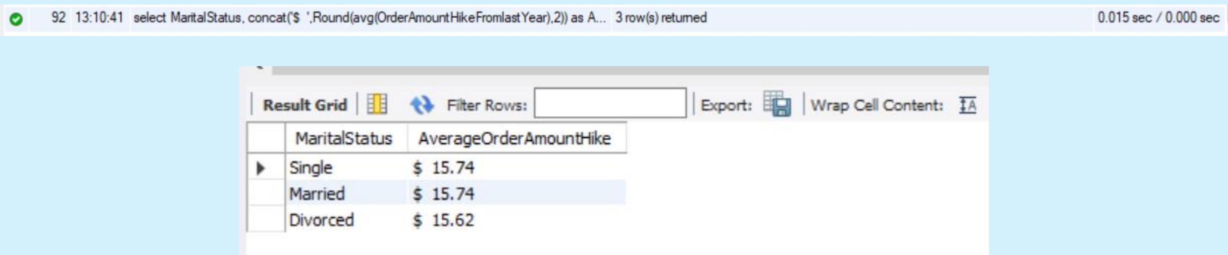
```
88  13:03:13  Select PreferredOrderCat as PreferredOrderCategory ,concat( max(HoursSpentOnAp...  5 row(s) returned  0.016 sec / 0.000 sec
```

PreferredOrderCategory	Maximum_hours_spent
Fashion	5 Hours
Grocery	4 Hours
Laptop & Accessory	5 Hours
Mobile Phone	5 Hours
Others	4 Hours

Explanation: This query finds the maximum hours spent on the app for each preferred order category.

12. Find the average order amount hike from last year for customers in each marital status category.

```
select MaritalStatus, concat('$  ',Round(avg(OrderAmountHikeFromlastYear),2)) as
AverageOrderAmountHike  from customer_churn group by MaritalStatus order by
MaritalStatus desc ;
```



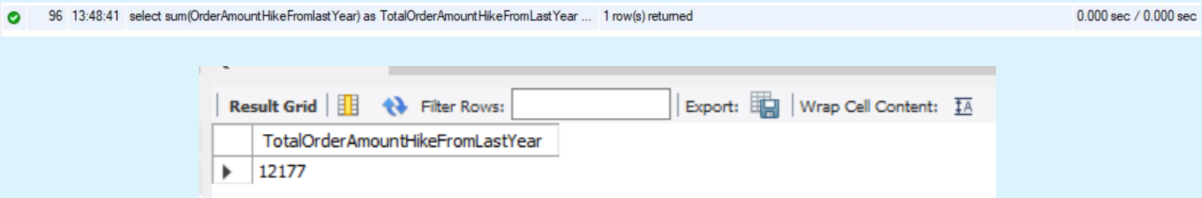
```
92  13:10:41  select MaritalStatus, concat('$  ',Round(avg(OrderAmountHikeFromlastYear),2)) as A...  3 row(s) returned  0.015 sec / 0.000 sec
```

MaritalStatus	AverageOrderAmountHike
Single	$ 15.74
Married	$ 15.74
Divorced	$ 15.62

Explanation: This query calculates the average order amount hike from last year, grouped by marital status.

13. Calculate the total order amount hike from last year for customers who are single and prefer mobile phones for ordering**.**

```
select sum(OrderAmountHikeFromlastYear) as TotalOrderAmountHikeFromLastYear from customer_churn where MaritalStatus = 'Single' and PreferredOrderCat = 'Mobile Phone';
```



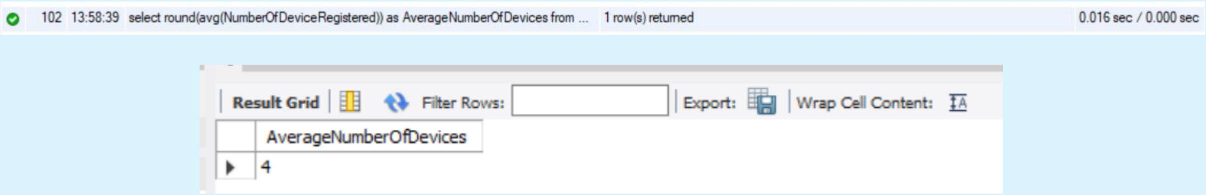
```
96  13:48:41  select sum(OrderAmountHikeFromlastYear) as TotalOrderAmountHikeFromLastYear ...  1 row(s) returned  0.000 sec / 0.000 sec
```

TotalOrderAmountHikeFromLastYear
12177

Explanation: This query sums up the order amount hike from last year for single customers who prefer mobile phones.

14. Find the average number of devices registered among customers who used UPI as their preferred payment mode.

```
select round(avg(NumberOfDeviceRegistered)) as AverageNumberOfDevices from customer_churn where PreferredPaymentMode='UPI';
```



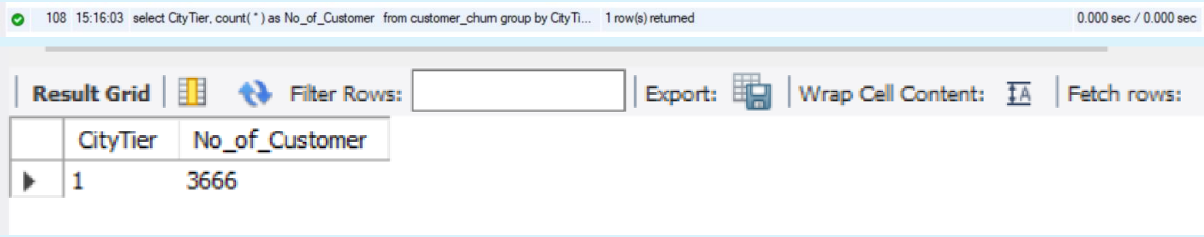
```
102  13:58:39  select round(avg(NumberOfDeviceRegistered)) as AverageNumberOfDevices from ...  1 row(s) returned  0.016 sec / 0.000 sec
```

AverageNumberOfDevices
4

Explanation: This query calculates the average number of devices registered by customers who used UPI as their payment mode.

15. Determine the city tier with the highest number of customers**.**

```
select CityTier, count( * ) as No_of_Customer   from customer_churn group by CityTier
order by No_of_Customer desc limit 1 ;
```



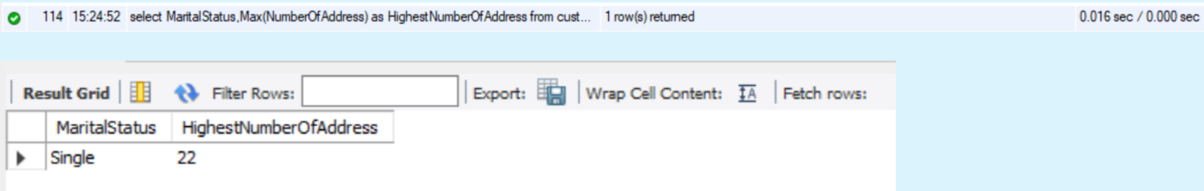
```
108  15:16:03  select CityTier, count( * ) as No_of_Customer  from customer_churn group by CityTi...  1 row(s) returned  0.000 sec / 0.000 sec
```

CityTier	No_of_Customer
1	3666

Explanation: This query finds the city tier with the highest number of customers.

16. Find the marital status of customers with the highest number of addresses.

```
select MaritalStatus,Max(NumberOfAddress) as HighestNumberOfAddress from
customer_churn group by MaritalStatus order by HighestNumberOfAddress desc limit 1;
```



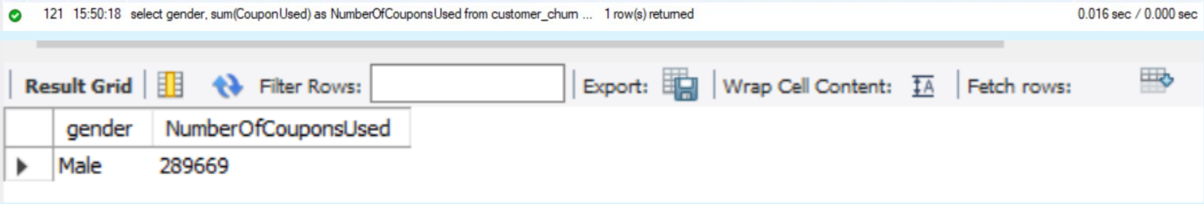
```
114  15:24:52  select MaritalStatus,Max(NumberOfAddress) as HighestNumberOfAddress from cust...  1 row(s) returned  0.016 sec / 0.000 sec
```

MaritalStatus	HighestNumberOfAddress
Single	22

Explanation: This query identifies the marital status of customers with the highest number of addresses.

17. Identify the gender that utilized the highest number of coupons.

```
select gender, sum(CouponUsed) as NumberOfCouponsUsed from customer_churn group by gender order by NumberOfCouponsUsed desc limit 1;
```



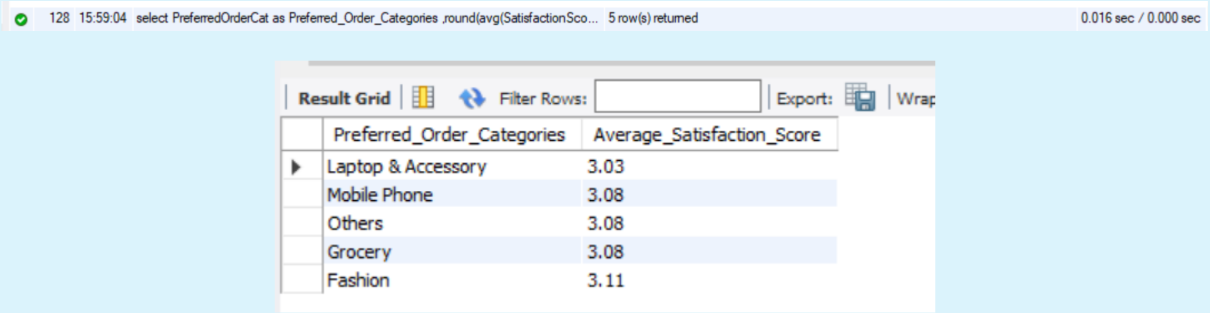
```
121  15:50:18  select gender, sum(CouponUsed) as NumberOfCouponsUsed from customer_churn ...  1 row(s) returned  0.016 sec / 0.000 sec
```

gender	NumberOfCouponsUsed
Male	289669

Explanation: This query finds the gender that has used the most coupons.

18. List the average satisfaction score in each of the preferred order categories.

```
select PreferredOrderCat as Preferred_Order_Categories ,round(avg(SatisfactionScore),2) as Average_Satisfaction_Score from customer_churn group by PreferredOrderCat order by Average_Satisfaction_Score ;
```



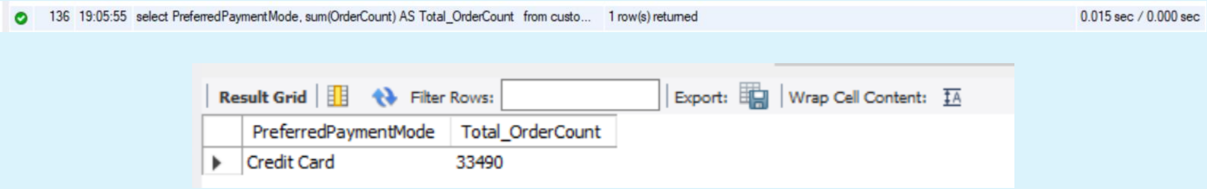
```
128  15:59:04  select PreferredOrderCat as Preferred_Order_Categories ,round(avg(SatisfactionSco...  5 row(s) returned  0.016 sec / 0.000 sec
```

Preferred_Order_Categories	Average_Satisfaction_Score
Laptop & Accessory	3.03
Mobile Phone	3.08
Others	3.08
Grocery	3.08
Fashion	3.11

Explanation: This query calculates the average satisfaction score for each preferred order category.

19. Calculate the total order count for customers who prefer using credit cards and have the maximum satisfaction score.

```
select PreferredPaymentMode, sum(OrderCount) AS Total_OrderCount   from customer_churn where PreferredPaymentMode ='Credit Card' and SatisfactionScore = (select max(SatisfactionScore) from customer_churn) group by PreferredPaymentMode ;
```



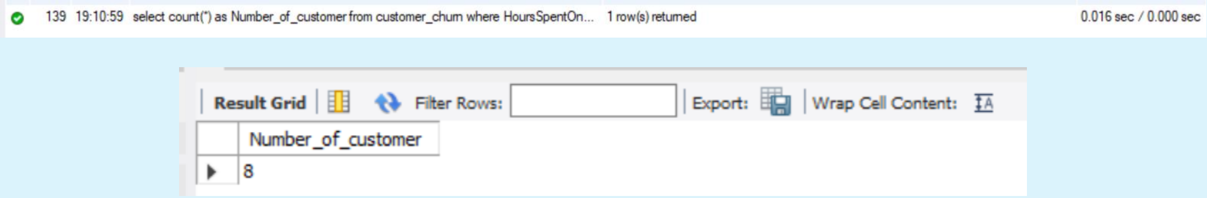
```
136  19:05:55  select PreferredPaymentMode, sum(OrderCount) AS Total_OrderCount  from custo...  1 row(s) returned  0.015 sec / 0.000 sec
```

PreferredPaymentMode	Total_OrderCount
Credit Card	33490

Explanation: This query calculates the total number of orders placed by credit card users who have the maximum satisfaction score.

20. How many customers are there who spent only one hour on the app and days since their last order was more than 5**?**

```
select count(*) as Number_of_customer from customer_churn where HoursSpentOnApp = 1 and DaySinceLastOrder > 5 ;
```



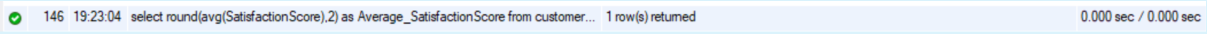
```
139  19:10:59  select count(*) as Number_of_customer from customer_chum where HoursSpentOn...  1 row(s) retumed  0.016 sec / 0.000 sec
```

Number_of_customer
8

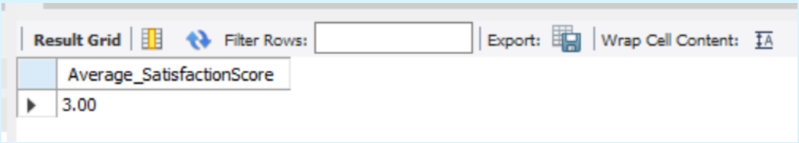
Explanation: This query counts the number of customers who spent exactly one hour on the app and have not placed an order in more than 5 days.

21. What is the average satisfaction score of customers who have complained**?**

```
select round(avg(SatisfactionScore),2) as Average_SatisfactionScore from customer_churn
where ComplaintReceived = 'Yes';
```



```
146 19:23:04 select round(avg(SatisfactionScore),2) as Average_SatisfactionScore from customer... 1 row(s) returned 0.000 sec / 0.000 sec
```

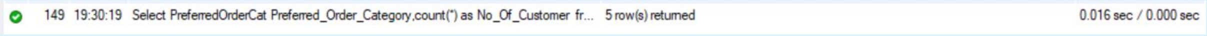


Average_SatisfactionScore
3.00

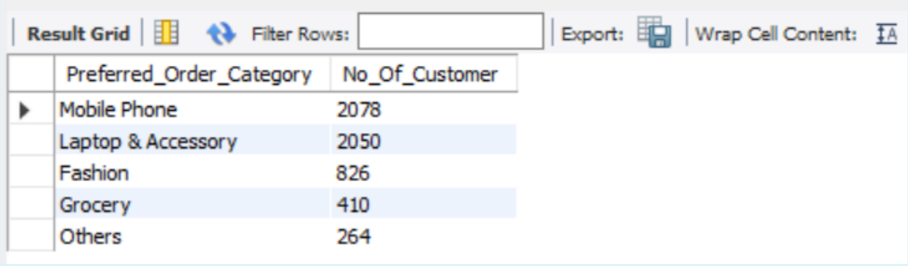
Explanation: This query calculates the average satisfaction score of customers who have complained.

22. How many customers are there in each preferred order category**?**

```
Select PreferredOrderCat Preferred_Order_Category,count(*) as No_Of_Customer  from
customer_churn group by preferredOrderCat order by No_OF_Customer desc;
```



```
149 19:30:19 Select PreferredOrderCat Preferred_Order_Category,count(*) as No_Of_Customer fr... 5 row(s) returned 0.016 sec / 0.000 sec
```

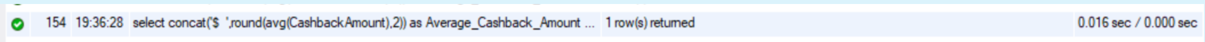


Preferred_Order_Category	No_Of_Customer
Mobile Phone	2078
Laptop & Accessory	2050
Fashion	826
Grocery	410
Others	264

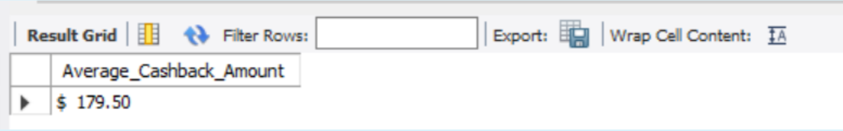
Explanation: This query counts the number of customers in each preferred order category.

23. What is the average cashback amount received by married customers**?**

```
select concat('$  ',round(avg(CashbackAmount),2)) as Average_Cashback_Amount from
customer_churn where MaritalStatus = 'Married' ;
```



```
154  19:36:28  select concat('$  ',round(avg(CashbackAmount),2)) as Average_Cashback_Amount ...  1 row(s) returned  0.016 sec / 0.000 sec
```

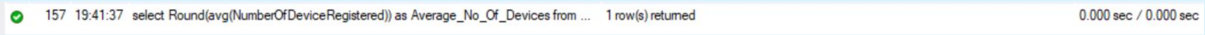


Average_Cashback_Amount
$ 179.50

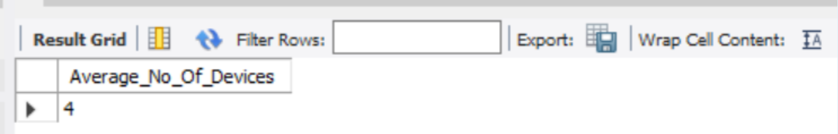
Explanation: This query calculates the average cashback amount received by married customers.

24. What is the average number of devices registered by customers who are not using Mobile Phone as their preferred login device**?**

```
select Round(avg(NumberOfDeviceRegistered)) as Average_No_Of_Devices from
customer_churn where PreferredLoginDevice not in ('Mobile Phone');
```



```
157  19:41:37  select Round(avg(NumberOfDeviceRegistered)) as Average_No_Of_Devices from ...  1 row(s) returned  0.000 sec / 0.000 sec
```

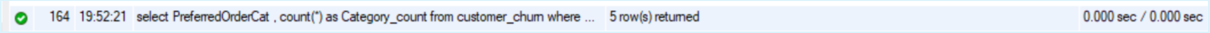


Average_No_Of_Devices
4

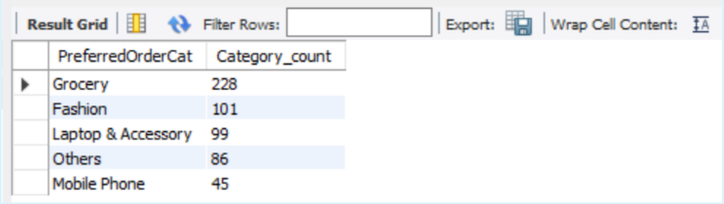
Explanation: This query calculates the average number of devices registered by customers who prefer other devices over mobile phones for logging in.

25. List the preferred order category among customers who used more than 5 coupons:

```
select PreferredOrderCat , count(*) as Category_count from customer_churn where
CouponUsed > 5 group by PreferredOrderCat order by Category_count desc   ;
```



```
164  19:52:21  select PreferredOrderCat , count(*) as Category_count from customer_churn where ...  5 row(s) returned  0.000 sec / 0.000 sec
```

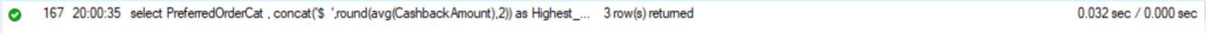


PreferredOrderCat	Category_count
Grocery	228
Fashion	101
Laptop & Accessory	99
Others	86
Mobile Phone	45

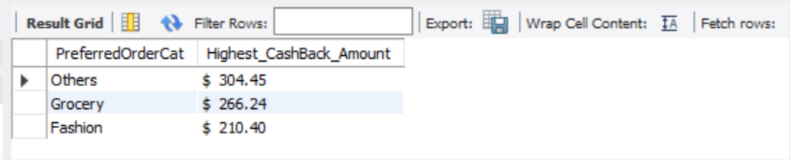
Explanation: This query selects the preferred order category and counts the number of customers who used more than 5 coupons. It groups the results by the preferred order category and orders them in descending order based on the count.

26. List the top 3 preferred order categories with the highest average cashback amount:

```
select PreferredOrderCat , concat('$  ',round(avg(CashbackAmount),2)) as
Highest_CashBack_Amount from customer_churn  group by PreferredOrderCat order by
Highest_CashBack_Amount desc limit 3;
```



```
167  20:00:35  select PreferredOrderCat , concat('$  ',round(avg(CashbackAmount),2)) as Highest_...  3 row(s) returned  0.032 sec / 0.000 sec
```

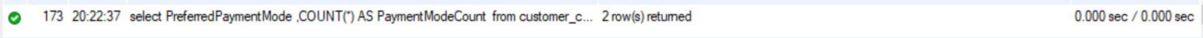


PreferredOrderCat	Highest_CashBack_Amount
Others	$ 304.45
Grocery	$ 266.24
Fashion	$ 210.40

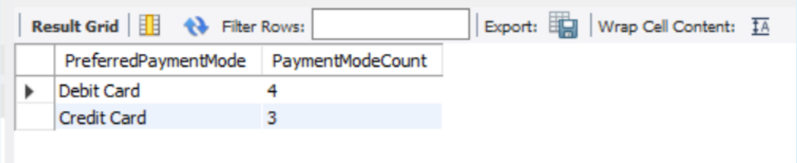
Explanation: This query calculates the average cashback amount for each preferred order category, groups the results by the preferred order category, and orders them in descending order based on the average cashback amount. It limits the results to the top 3 categories.

27. Find the preferred payment modes of customers whose average tenure is 10 months and have placed more than 500 orders:

```
select PreferredPaymentMode ,COUNT(*) AS PaymentModeCount  from customer_churn
WHERE Tenure = 10 AND OrderCount > 500 GROUP BY PreferredPaymentMode ORDER
BY PaymentModeCount DESC;
```



```
173  20:22:37  select PreferredPaymentMode ,COUNT(*) AS PaymentModeCount  from customer_c...  2 row(s) returned    0.000 sec / 0.000 sec
```

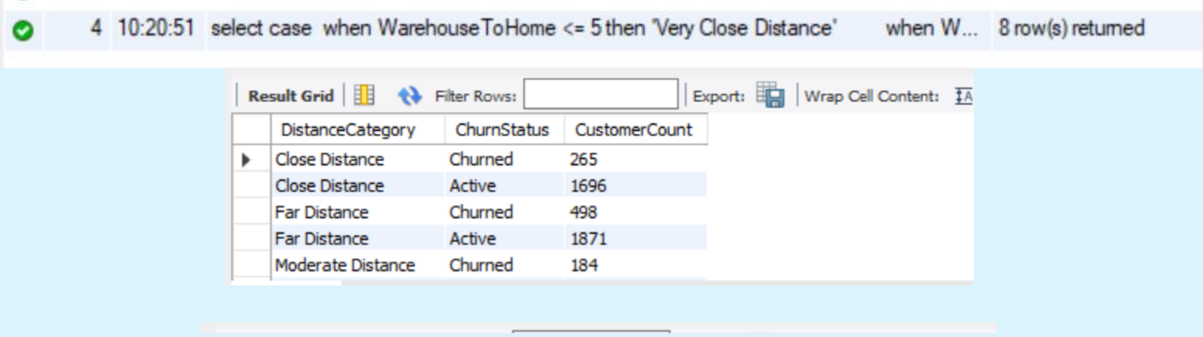


PreferredPaymentMode	PaymentModeCount
Debit Card	4
Credit Card	3

Explanation: This query selects the preferred payment mode and counts the number of customers whose average tenure is 10 months and who have placed more than 500 orders. It groups the results by the preferred payment mode and orders them in descending order based on the count.

28. Categorize customers based on their distance from the warehousetohome such as 'Very Close Distance' for distances <=5km, 'Close Distance' for <=10km, 'Moderate Distance' for <=15km, and 'Far Distance' for >15km. Then display the churn status breakdown for each distance category:

```
select case when WarehouseToHome <= 5 then 'Very Close Distance'
                 when WarehouseToHome <=10 then 'Close Distance'
                  when  WarehouseToHome <=15 then 'Moderate Distance'
                  else 'Far Distance'
                  end  as DistanceCategory,
                   ChurnStatus ,
                   count(*) as CustomerCount  from customer_churn  GROUP BY
DistanceCategory, ChurnStatus ORDER BY DistanceCategory, ChurnStatus;
```



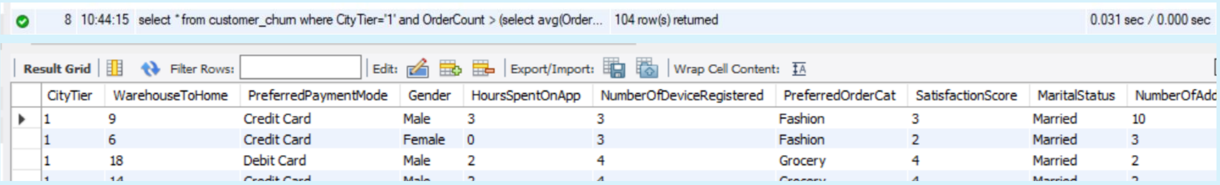
```
4  10:20:51  select case  when WarehouseToHome <= 5 then 'Very Close Distance'     when W...   8 row(s) returned
```

DistanceCategory	ChurnStatus	CustomerCount
Close Distance	Churned	265
Close Distance	Active	1696
Far Distance	Churned	498
Far Distance	Active	1871
Moderate Distance	Churned	184

Explanation: This query categorizes customers based on their distance from the warehouse to home and counts the number of customers in each churn status for each distance category. It groups the results by the distance category and churn status and orders them accordingly.

29. List the customer's order details who are married, live in City Tier-1, and their order counts are more than the average number of orders placed by all customers:

```
select * from customer_churn where CityTier='1' and OrderCount > (select avg(OrderCount)
from customer_churn) and MaritalStatus = 'Married';
```



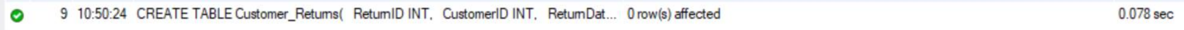
```
8  10:44:15  select * from customer_churn where CityTier='1' and OrderCount > (select avg(Order...  104 row(s) returned    0.031 sec / 0.000 sec
```

CityTier	WarehouseToHome	PreferredPaymentMode	Gender	HoursSpentOnApp	NumberOfDeviceRegistered	PreferredOrderCat	SatisfactionScore	MaritalStatus	NumberOfAdc
1	9	Credit Card	Male	3	3	Fashion	3	Married	10
1	6	Credit Card	Female	0	3	Fashion	2	Married	3
1	18	Debit Card	Male	2	4	Grocery	4	Married	2
1	14	Credit Card	Male	2	4	Grocery	4	Married	2

Explanation: This query calculates the average number of orders placed by all customers and then selects the details of customers who are married, live in City Tier-1, and have placed more orders than the average.

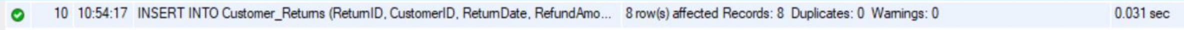
30. a) Create a ‘customer_returns’ table in the ‘ecomm’ database and insert the following data:

```
CREATE TABLE Customer_Returns(
  ReturnID INT,
  CustomerID INT,
  ReturnDate DATE,
  RefundAmount DECIMAL(10, 2)
);
```



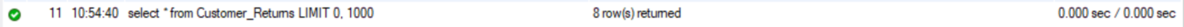
```
9  10:50:24  CREATE TABLE Customer_Returns(  ReturnID INT,  CustomerID INT,  ReturnDat...  0 row(s) affected                0.078 sec
```

```
INSERT INTO Customer_Returns (ReturnID, CustomerID, ReturnDate, RefundAmount)
VALUES
(1001, 50022, '2023-01-01', 2130),
(1002, 50316, '2023-01-23', 2000),
(1003, 51099, '2023-02-14', 2290),
(1004, 52321, '2023-03-08', 2510),
(1005, 52928, '2023-03-20', 3000),
(1006, 53749, '2023-04-17', 1740),
(1007, 54206, '2023-04-21', 3250),
(1008, 54838, '2023-04-30', 1990);
```

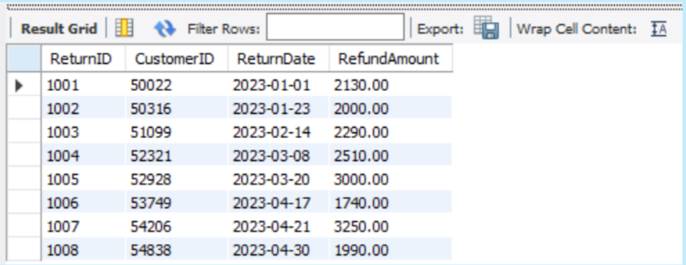


```
10  10:54:17  INSERT INTO Customer_Returns (ReturnID, CustomerID, ReturnDate, RefundAmo...  8 row(s) affected Records: 8  Duplicates: 0  Warnings: 0    0.031 sec
```

```
select * from Customer_Returns;
```



```
11  10:54:40  select * from Customer_Returns LIMIT 0, 1000          8 row(s) returned          0.000 sec / 0.000 sec
```

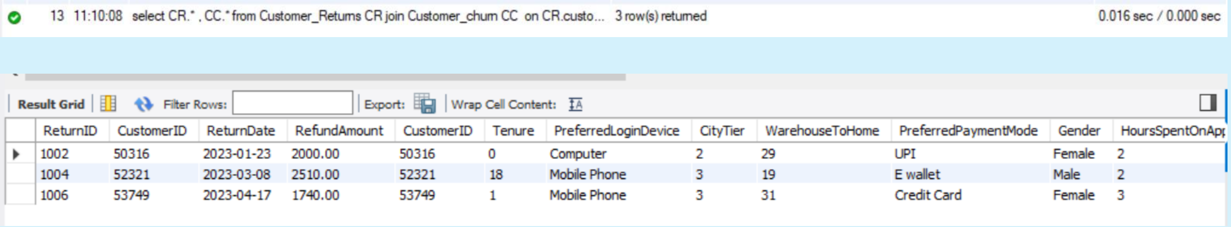


ReturnID	CustomerID	ReturnDate	RefundAmount
1001	50022	2023-01-01	2130.00
1002	50316	2023-01-23	2000.00
1003	51099	2023-02-14	2290.00
1004	52321	2023-03-08	2510.00
1005	52928	2023-03-20	3000.00
1006	53749	2023-04-17	1740.00
1007	54206	2023-04-21	3250.00
1008	54838	2023-04-30	1990.00

Explanation: This query creates a new table Customer_Returns in the ecomm database and inserts the provided data into the table.

30. b) Display the return details along with the customer details of those who have churned and have made complaints:

```
select CR.* , CC.* from Customer_Returns CR
join Customer_churn CC  on CR.customerId =  CC.customerId
where CC.ChurnStatus = 'Churned' and CC.ComplaintReceived='Yes' ;
```



```
13  11:10:08  select CR.* , CC.* from Customer_Returns CR join Customer_churn CC  on CR.custo...  3 row(s) returned  0.016 sec / 0.000 sec
```

ReturnID	CustomerID	ReturnDate	RefundAmount	CustomerID	Tenure	PreferredLoginDevice	CityTier	WarehouseToHome	PreferredPaymentMode	Gender	HoursSpentOnApp
1002	50316	2023-01-23	2000.00	50316	0	Computer	2	29	UPI	Female	2
1004	52321	2023-03-08	2510.00	52321	18	Mobile Phone	3	19	E wallet	Male	2
1006	53749	2023-04-17	1740.00	53749	1	Mobile Phone	3	31	Credit Card	Female	3

Explanation: This query joins the customer_returns table with the dataset table on the CustomerID and selects the return details along with the customer details for those who have churned and have made complaints.

6. Findings and Insights

- **Churn Drivers**: The data shows that customers with longer distances from the warehouse are more likely to churn. Additionally, dissatisfaction in certain preferred order categories such as "Laptop & Accessory" has a significant impact.
- **Preferred Payment Mode**: Credit card users and customers using mobile phones for login have higher retention rates.
- **Retention Strategies**: Targeted strategies such as offering personalized promotions for customers at risk of churn (identified by low satisfaction scores or high complaint rates) could improve retention.

- INSIGHTS TABLE

Insight	**Actionable Insights**
Proximity Impact	Optimize delivery for nearby customers.
High-Value Customers	Target married, Tier-1 city customers with loyalty programs.
Satisfaction Scores	Improve feedback and complaint resolution.
Payment Preferences	Enhance credit card payment experience.
Purchase Behavior	Address issues in “Laptop & Accessory” category.
Complaint Handling	Implement effective complaint resolution processes.
App Usage	Offer in-app promotions to active users.
Cashback Incentives	Extend cashback offers to other customer segments.

- This concise table highlights key insights and actionable steps to improve customer retention and satisfaction.

7. Conclusion

The E-Commerce Customer Churn Analysis reveals key factors influencing customer churn, such as proximity to the warehouse, customer satisfaction, and preferred payment modes. By addressing these factors, businesses can improve retention and satisfaction. Key actions include optimizing delivery for nearby customers, enhancing credit card payment experiences, and resolving complaints effectively. Implementing these strategies will help foster long-term customer relationships and ensure sustained profitability.