E-Commerce Customer Churn Analysis

A

KANNAN_RAJENDRAN

DATA_BASE

Contents

Introduction

A Data-Driven Approach to Customer Retention in E-commerce

In today's competitive e-commerce landscape, customer retention is paramount to sustained business success. This project aims to address the challenge of customer churn by leveraging historical transactional data. By analyzing key customer attributes such as tenure, preferred payment methods, satisfaction scores, and purchase behavior, we seek to uncover the underlying factors driving customer attrition.

Through rigorous data cleaning and preprocessing, we will ensure data quality and consistency. Subsequently, advanced data mining techniques will be employed to identify patterns and trends within the data. By segmenting customers based on their characteristics and behaviors, we can develop tailored retention strategies.

The ultimate goal of this project is to provide e-commerce businesses with actionable insights that can be used to implement targeted retention initiatives. By proactively addressing customer needs and concerns, we can foster long-lasting relationships and mitigate the risk of customer churn.

Problem Statement:

In the realm of e-commerce, businesses face the challenge of understanding customer churn patterns to ensure customer satisfaction and sustained profitability. This project aims to delve into the dynamics of customer churn within an e-commerce domain, utilizing historical transactional data to uncover underlying patterns and drivers of churn. By analyzing customer attributes such as tenure, preferred payment modes, satisfaction scores, and purchase behavior, the project seeks to investigate and understand the dynamics of customer attrition and their propensity to churn. The ultimate objective is to equip e-commerce enterprises with actionable insights to implement targeted retention strategies and mitigate churn, thereby fostering long-term customer relationships and ensuring business viability in a competitive landscape.

Dataset Download:

https://drive.google.com/uc?export=download&id=1iKKCze_Fpk2n_g3BIZBiSjcDFdFcEn3D

Project Steps and Objectives:

Data Cleaning:

Handling Missing Values and Outliers:

- Impute mean for the following columns, and round off to the nearest integer if required:
  1. WarehouseToHome,

```
-- mean values update by warehouseToHome column
set @mean_warehousetohome= round((select avg(WarehouseToHome is not null) from customer_churn),0);
select @mean_warehousetohome;
set sql_safe_updates=0;
update customer_churn
set WarehouseToHome= @mean_warehousetohome
where WarehouseToHome is null;
```

  2. HourSpendOnApp,

```
-- mean values update missing field by HourSpendOnApp column
set @HourSpendOnApp= round((select avg(HourSpendOnApp is not null) from customer_churn),0);
select @HourSpendOnApp;
update customer_churn
set HourSpendOnApp = @HourSpendOnApp
where HourSpendOnApp is null;
```

  3. OrderAmountHikeFromlastYear,

```
-- mean values update missing field by OrderAmountHikeFromlastYear column
set @OrderAmountHikeFromlastYear = round((select avg(OrderAmountHikeFromlastYear is not null) from customer_churn),0);
select @OrderAmountHikeFromlastYear;
update customer_churn
set OrderAmountHikeFromlastYear = @OrderAmountHikeFromlastYear
where OrderAmountHikeFromlastYear is null;
```

4. DaySinceLastOrder.

```
-- mean values update missing field by DaySinceLastOrder column --
set @DaySinceLastOrder = round((select avg(DaySinceLastOrder is not null) from customer_churn),0);
select @DaySinceLastOrder;
update customer_churn
set DaySinceLastOrder = @DaySinceLastOrder
where DaySinceLastOrder is null;
```

- Impute mode for the following columns:

1. Tenure,

```
-- Impute mode for the missing field of the Tenure column --
set @Tenure=(select Tenure from customer_churn group by Tenure order by count(Tenure) desc limit 1 );
select @Tenure;
update customer_churn
set tenure =@Tenure
where tenure is null;
```

2. CouponUsed,

```
-- Impute mode for the missing field of the CouponUsed column --
select CouponUsed, count(couponused) from customer_churn group by CouponUsed order by count(CouponUsed) desc ;
select CouponUsed from customer_churn group by CouponUsed order by count(CouponUsed) desc limit 1;
set @CouponUsed =(select CouponUsed from customer_churn group by CouponUsed order by count(CouponUsed) desc limit 1);
select @CouponUsed;
update customer_churn
set couponused = @couponused
where couponused is null;
```

3. OrderCount.

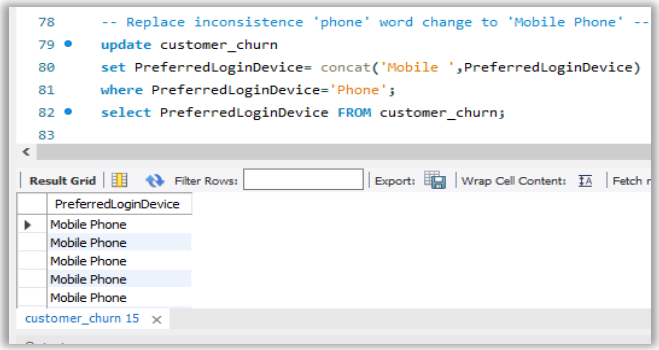
```
-- Impute mode for the missing field of the OrderCount column --
select OrderCount, count(OrderCount) from customer_churn group by ordercount order by count(ordercount) desc;
select ordercount from customer_churn group by ordercount order by count(ordercount) desc limit 1;
set @ordercount = (select ordercount from customer_churn group by ordercount order by count(ordercount) desc limit 1);
select @ordercount;
update customer_churn
set ordercount = @ordercount
where ordercount is null;
```

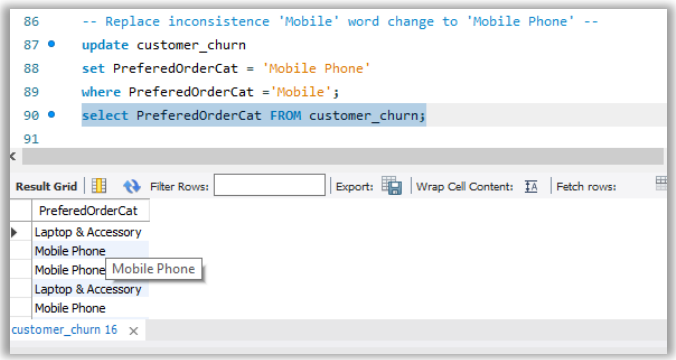
- Handle outliers in the 'WarehouseToHome' column by deleting rows where the values are greater than 100.

```
--  Handle outliers in the 'WarehouseToHome' column by deleting rows where the values are greater than 100. --
-- Delete the outlier WarehouseToHome column --
DELETE FROM customer_churn WHERE WarehouseToHome > 100;
```

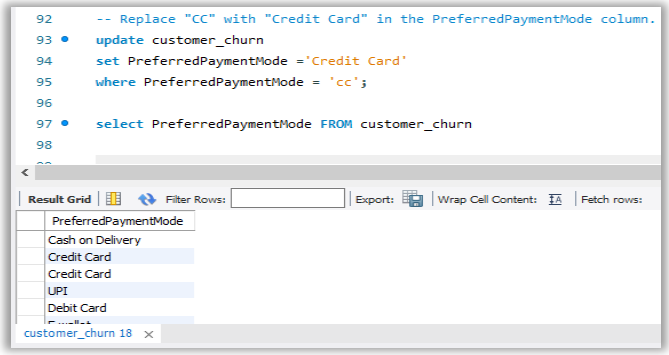
Dealing with Inconsistencies:

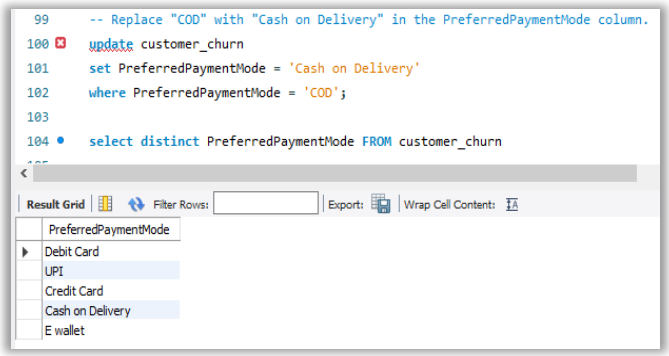
- Replace occurrences of “Phone” in the 'PreferredLoginDevice' column and “Mobile” in the 'PreferedOrderCat' column with “Mobile Phone” to ensure uniformity.





- Standardize payment mode values: Replace "COD" with "Cash on Delivery" and "CC" with "Credit Card" in the PreferredPaymentMode column.





Data Transformation:

Column Renaming:

- Rename the column "PreferedOrderCat" to "PreferredOrderCat".
- Rename the column "HourSpendOnApp" to "HoursSpentOnApp".

```
-- Rename the column "PreferedOrderCat" to "PreferredOrderCat". --
ALTER TABLE customer_churn RENAME COLUMN PreferedOrderCat TO PreferredOrderCat;


-- Rename the column "HourSpendOnApp" to "HoursSpentOnApp".--
ALTER TABLE customer_churn
RENAME COLUMN HourSpendOnApp TO HoursSpentOnApp;
```

Creating New Columns:

- Create a new column named 'ComplaintReceived' with values "Yes" if the corresponding value in the 'Complain' is 1, and "No" otherwise.

```
114    -- CREATE TABLE 'ComplaintReceived' --
115    ALTER TABLE customer_churn
116    ADD  ComplaintReceived VARCHAR(5);
117    update customer_churn
118    set ComplaintReceived = case when complain=1 then 'YES' else 'No' end;
119    select complain,ComplaintReceived FROM customer_churn
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

complain	ComplaintReceived
1	YES
1	YES
1	YES
0	No
0	No

customer_churn 22

- Create a new column named 'ChurnStatus'. Set its value to "Churned" if the corresponding value in the 'Churn' column is 1, else assign "Active".

```
121    -- CREATE TABLE 'ChurnStatus'
122    ALTER TABLE customer_churn
123    ADD COLUMN ChurnStatus VARCHAR(10);
124    UPDATE customer_churn
125    SET ChurnStatus = IF (Churn=1, 'Churn','Active');
126    select distinct Churn ,ChurnStatus FROM customer_churn
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Churn	ChurnStatus
1	Churn
0	Active

Column Dropping:

- Drop the columns "Churn" and "Complain" from the table.

```sql
-- Drop the "Churn" column --
ALTER TABLE customer_churn
DROP COLUMN Churn;

-- Drop the "Complain" column --
ALTER TABLE customer_churn
DROP COLUMN Complain;
```

Data Exploration and Analysis:

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

```sql
-- Retrieve the count of churned and active customers from the dataset.
select ChurnStatus, count(ChurnStatus) as count_of_ChurnStatus from customer_churn
group by ChurnStatus order by ChurnStatus;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ChurnStatus	count_of_ChurnStatus
Active	4680
Churn	948

2. Display the average tenure of customers who churned.

```sql
-- Display the average tenure of customers who churned.
SELECT ChurnStatus, ROUND(AVG(tenure),2) AS Average_tenure FROM customer_churn
WHERE ChurnStatus='churn';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ChurnStatus	Average_tenure
Churn	3.18

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

```
148     -- Calculate the total cashback amount earned by customers who churned.--
149 •   SELECT ChurnStatus, SUM(CashbackAmount) AS total_cashback_amount FROM customer_churn
150     WHERE ChurnStatus='churn';
151
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ChurnStatus	total_cashback_amount
Churn	152030

4. Determine the percentage of churned customers who complained.

```
154     -- Determine the percentage of churned customers who complained.--
155 •   SELECT ChurnStatus, round((sum(IF(ComplaintReceived='Yes',1,0))*100/count(*)),2)
156     AS percentage_of_churned_who_complained
157     FROM customer_churn WHERE ChurnStatus='churn';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ChurnStatus	percentage_of_churned_who_complained
Churn	53.59

5. Find the gender distribution of customers who complained.

```
160     -- Find the gender distribution of customers who complained.--
161 •   SELECT Gender, COUNT(*) AS Complained FROM customer_churn WHERE ComplaintReceived='Yes'
162     GROUP BY Gender;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Gender	Complained
Female	690
Male	914

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

```
166     /*-- Identify the city tier with the highest number of churned customers
167     whose preferred order category is Laptop & Accessory.*/
168 •    SELECT CityTier, Count(*) AS number_of_churned FROM customer_churn
169     where ChurnStatus='churn' AND PreferredOrderCat='Laptop & Accessory'
170     GROUP BY CityTier ORDER BY CityTier DESC LIMIT 1;
171
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

CityTier	number_of_churned
3	150

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

```
172    -- Identify the most preferred payment mode among active customers.
173 •  SELECT PreferredPaymentMode,COUNT(*) AS preferred_payment_mode_count FROM customer_churn
174    WHERE ChurnStatus='active'
175    GROUP BY PreferredPaymentMode
176    ORDER BY preferred_payment_mode_count DESC LIMIT 1;
177
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

PreferredPaymentMode	preferred_payment_mode_count
Debit Card	1956

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

```
179    -- List the preferred login device(s) among customers who took more than 10 days since their last order.
180 •  SELECT PreferredLoginDevice, COUNT(PreferredLoginDevice) AS Count_of_PreferredLoginDevice FROM customer_churn
181    WHERE DaySinceLastOrder>10
182    GROUP BY PreferredLoginDevice;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PreferredLoginDevice	Count_of_PreferredLoginDevice
Mobile Phone	212
Computer	96

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

```
184    -- List the number of active customers who spent more than 3 hours on the app.
185 •  SELECT ChurnStatus, HoursSpentOnApp, COUNT(*) AS Number_of_customer_spentOnApp FROM  customer_churn
186    WHERE ChurnStatus='active' AND HoursSpentOnApp>3
187    GROUP BY HoursSpentOnApp;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ChurnStatus	HoursSpentOnApp	Number_of_customer_spentOnApp
Active	4	978
Active	5	3

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

```
189    -- Find the average cashback amount received by customers who spent at least 2 hours on the app.
190 •  SELECT ROUND(AVG(CashbackAmount),2) AS average_cashback_amount FROM customer_churn
191    WHERE HoursSpentOnApp>=2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

average_cashback_amount
179.53

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

```
193    -- Display the maximum hours spent on the app by customers in each preferred order category.
194    SELECT PreferredOrderCat, MAX(HoursSpentOnApp) AS Maximum_hours_spent FROM customer_churn
195    GROUP BY PreferredOrderCat;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PreferredOrderCat	Maximum_hours_spent
Laptop & Accessory	5
Mobile Phone	5
Others	4
Fashion	5
Grocery	4

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

```
197    -- Find the average order amount hike from last year for customers in each marital status category.
198    SELECT MaritalStatus, ROUND(AVG(OrderAmountHikeFromlastYear),2) AS AVERAGE_Order_Amount_Hike_From_lastYear FROM customer_churn
199    GROUP BY MaritalStatus;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MaritalStatus	AVERAGE_Order_Amount_Hike_From_lastYear
Single	15.26
Divorced	14.65
Married	14.97

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

```
201    -- Calculate the total order amount hike from last year for customers who are single and prefer mobile phones for ordering.
202    SELECT MaritalStatus, PreferredOrderCat, SUM(OrderAmountHikeFromlastYear) AS Total_OrderAmountHikeFromlastYear FROM customer_churn
203    WHERE MaritalStatus= 'single' AND PreferredOrderCat= 'Mobile Phone';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MaritalStatus	PreferredOrderCat	Total_OrderAmountHikeFromlastYear
Single	Mobile Phone	12177

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

```
206    -- Find the average number of devices registered among customers who used UPI as their preferred payment mode.
207    SELECT PreferredPaymentMode, ROUND(AVG(NumberOfDeviceRegistered),2) AS AVERAGE_NumberOfDeviceRegistered FROM customer_churn
208    WHERE PreferredPaymentMode='UPI';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PreferredPaymentMode	AVERAGE_NumberOfDeviceRegistered
UPI	3.72

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

```
210    -- Determine the city tier with the highest number of customers
211    SELECT CityTier, COUNT(*) AS COUNT_OF_CUSTOMERS FROM customer_churn
212    GROUP BY CityTier
213    ORDER BY COUNT_OF_CUSTOMERS DESC LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

CityTier	COUNT_OF_CUSTOMERS
1	3666

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

```
-- Find the marital status of customers with the highest number of addresses.
SELECT MaritalStatus, count(NumberOfAddress) AS Highest_number_of_addresses FROM customer_churn
GROUP BY MaritalStatus
ORDER BY Highest_number_of_addresses DESC LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

MaritalStatus	Highest_number_of_addresses
Married	2984

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

```
-- Identify the gender that utilized the highest number of coupons.
SELECT Gender, COUNT(CouponUsed) AS Highest_number_of_coupon FROM customer_churn
GROUP BY Gender
ORDER BY Highest_number_of_coupon DESC LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Gender	Highest_number_of_coupon
Male	3382

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

```
-- List the average satisfaction score in each of the preferred order categories.
SELECT PreferredOrderCat, ROUND(AVG(SatisfactionScore),2) AS Average_satisfaction_each_category FROM customer_churn
GROUP BY PreferredOrderCat
ORDER BY Average_satisfaction_each_category DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PreferredOrderCat	Average_satisfaction_each_category
Fashion	3.11
Mobile Phone	3.08
Others	3.08
Grocery	3.08
Laptop & Accessory	3.03

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

```
-- Calculate the total order count for customers who prefer using credit cards and have the maximum satisfaction score.
SELECT PreferredPaymentMode, SUM(OrderCount) AS Total_order_count FROM customer_churn
WHERE (SELECT MAX(SatisfactionScore) FROM CUSTOMER_CHURN)
GROUP BY PreferredPaymentMode  HAVING  PreferredPaymentMode= 'Credit Card';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PreferredPaymentMode	Total_order_count
Credit Card	5409

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

```
/*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(*) COUNT_OF_CUSTOMERS FROM customer_churn
WHERE HoursSpentOnApp =1 AND DaySinceLastOrder>5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

COUNT_OF_CUSTOMERS
55

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

```
248     -- What is the average satisfaction score of customers who have complained
249 •   SELECT ComplaintReceived, round(AVG(SatisfactionScore),2) AS Average_satisfactionscore FROM customer_churn
250     WHERE ComplaintReceived='Yes';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

COUNT_OF_CUSTOMERS
55

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

```
253 •   SELECT PreferredOrderCat, COUNT(*) AS count_of_customer_PreferredOrderCat FROM  customer_churn
254     GROUP BY PreferredOrderCat
255     ORDER BY count_of_customer_PreferredOrderCat DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PreferredOrderCat	count_of_customer_PreferredOrderCat
Mobile Phone	2078
Laptop & Accessory	2050
Fashion	826
Grocery	410
Others	264

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

```
257      -- What is the average cashback amount received by married customers
258 •   SELECT MaritalStatus, round(AVG(CashbackAmount),2) AS Average_CashbackAmount FROM customer_churn
259     WHERE MaritalStatus = 'Married';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MaritalStatus	Average_CashbackAmount
Married	179.50

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

```
262     -- What is the average number of devices registered by customers who are not using Mobile Phone as their preferred login device?
263 •   SELECT ROUND(AVG(NumberOfDeviceRegistered),2) AS Average_number_of_devices_registered FROM customer_churn
264     WHERE PreferredLoginDevice<>'Mobile Phone';
265 •   SELECT ROUND(AVG(NumberOfDeviceRegistered),2) AS Average_number_of_devices_registered FROM customer_churn
266     GROUP BY PreferredLoginDevice
267     HAVING 'Mobile Phone' NOT IN (PreferredLoginDevice);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Average_number_of_devices_registered
3.72

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

```
271    -- List the preferred order category among customers who used more than 5 coupons
272    SELECT CustomerID,PreferredOrderCat, CouponUsed  FROM customer_churn
273    WHERE CouponUsed> 5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

CustomerID	PreferredOrderCat	CouponUsed
50011	Others	9
50021	Laptop & Accessory	6
50022	Fashion	11
50086	Fashion	7
50094	Fashion	12
50127	Fashion	10

customer_churn 13

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

```
276    -- List the top 3 preferred order categories with the highest average cashback amount.
277    SELECT PreferredOrderCat, ROUND(AVG(CashbackAmount),2) AS Average_CashbackAmount FROM customer_churn
278    GROUP BY PreferredOrderCat
279    ORDER BY Average_CashbackAmount DESC LIMIT 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

PreferredOrderCat	Average_CashbackAmount
Others	304.45
Grocery	266.24
Fashion	210.40

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

```
283    -- Find the preferred payment modes of customers whose average tenure is 10 months and have placed more than 500 orders.
284    SELECT PreferredPaymentMode, ROUND(AVG(Tenure)) AS Average_Tenure, SUM(OrderCount) AS Sum_of_orders_placed FROM customer_churn
285    GROUP BY PreferredPaymentMode
286    HAVING  Sum_of_orders_placed>500 AND Average_Tenure= 10;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PreferredPaymentMode	Average_Tenure	Sum_of_orders_placed
Debit Card	10	6802
Credit Card	10	5409
E wallet	10	1849

28. Categorize customers based on their distance from the warehouse to home 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.

```
/* 28)  Categorize customers based on their distance from the warehouse to home 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 CustomerID, WarehouseToHome, IF(WarehouseToHome<=5, 'Very Close Distance',IF(WarehouseToHome<=10,'Close Distance',
IF(WarehouseToHome<=15,'Moderate Distance', 'Far Distance'))) AS Distance_category FROM customer_churn;

SELECT CustomerID, WarehouseToHome,
    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 Distance_category
FROM customer_churn;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

CustomerID	WarehouseToHome	Distance_category
50001	6	Close Distance
50002	8	Close Distance
50003	30	Far Distance
50004	15	Moderate Distance
50005	12	Moderate Distance

Result 22 | Result 23

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.

```
/* 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 MaritalStatus= 'married' AND CityTier= '1' AND OrderCount>(SELECT AVG(OrderCount) FROM customer_churn);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

CustomerID	Tenure	PreferredLoginDevice	CityTier	WarehouseToHome	PreferredPaymentMode	Gender	HoursSpentOnApp	NumberOfDeviceRegistered	PreferredOrderCat	SatisfactionScor
50049	3	Computer	1	15	Credit Card	Male	1	3	Laptop & Accessory	5
50119	13	Mobile Phone	1	30	Cash on Delivery	Male	3	3	Laptop & Accessory	2
50132	15	Mobile Phone	1	1	Credit Card	Female	2	4	Mobile Phone	5
50133	13	Mobile Phone	1	8	Credit Card	Male	2	3	Others	3
50152	2	Mobile Phone	1	28	Debit Card	Female	3	2	Fashion	4

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

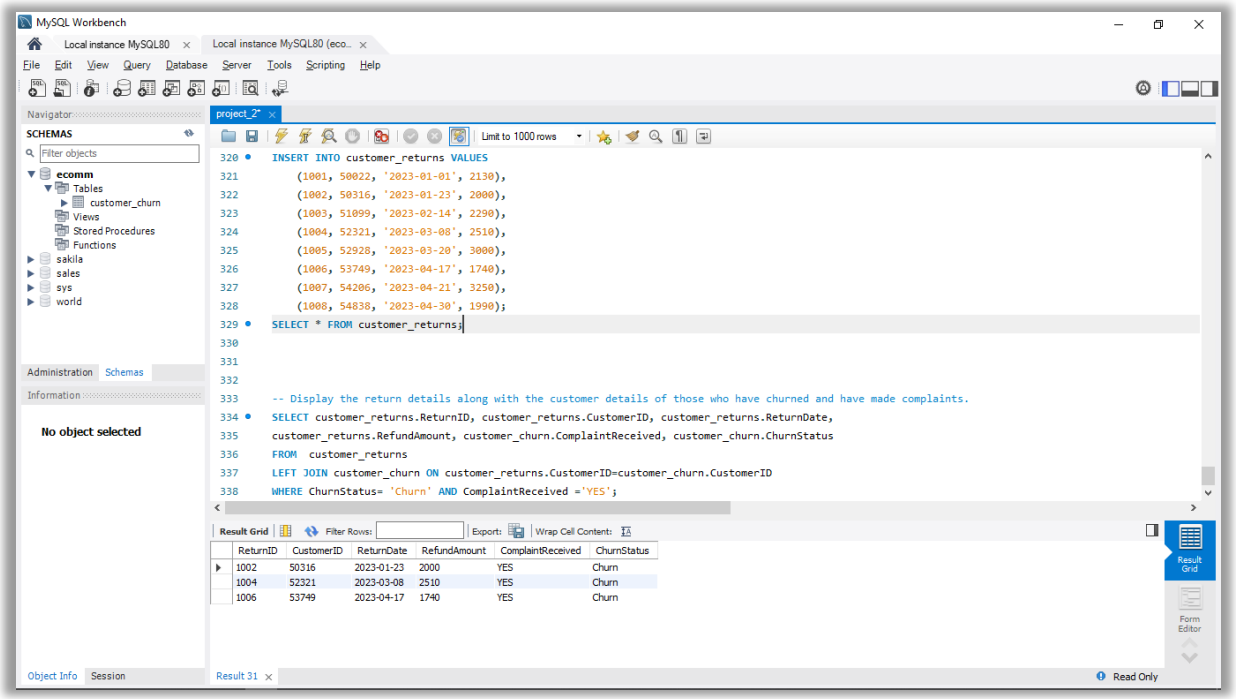
ReturnID	CustomerID ReturnDate RefundAmount
1001	50022 2023-01-01 2130
1002	50316 2023-01-23 2000
1003	51099 2023-02-14 2290
1004	52321 2023-03-08 2510
1005	52928 2023-03-20 3000
1006	53749 2023-04-17 1740
1007	54206 2023-04-21 3250
1008	54838 2023-04-30 1990

```
311     -- Create a 'customer_returns' table in the 'ecomm' database and insert the data: --
312     CREATE TABLE customer_returns(
313         ReturnID INT PRIMARY KEY,
314         CustomerID INT,
315         FOREIGN KEY (CustomerID) REFERENCES customer_churn (CustomerID),
316         ReturnDate DATE,
317         RefundAmount INT
318     );
319     --          INSERT VALUES INTO customer_returns TABLE
320     INSERT INTO customer_returns VALUES
321         (1001, 50022, '2023-01-01', 2130),
322         (1002, 50316, '2023-01-23', 2000),
323         (1003, 51099, '2023-02-14', 2290),
324         (1004, 52321, '2023-03-08', 2510),
325         (1005, 52928, '2023-03-20', 3000),
326         (1006, 53749, '2023-04-17', 1740),
327         (1007, 54206, '2023-04-21', 3250),
328         (1008, 54838, '2023-04-30', 1990);
329     SELECT * FROM customer_returns;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ReturnID	CustomerID	ReturnDate	RefundAmount
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

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



Conclusion:

By meticulously analyzing the historical transactional data, we have gained valuable insights into the factors driving customer churn within the e-commerce domain. Through data cleaning, transformation, and exploratory data analysis, we have identified key metrics and patterns that significantly impact customer retention.

The findings from this study provide a solid foundation for e-commerce businesses to implement targeted retention strategies. By understanding the preferences, behaviors, and pain points of different customer segments, businesses can proactively address their needs and improve overall customer satisfaction.

Furthermore, the insights gained from this analysis can be leveraged to optimize marketing campaigns, personalize customer experiences, and enhance customer support services. By focusing on customer retention, e-commerce businesses can not only reduce costs associated with acquiring new customers but also build long-lasting relationships that drive sustainable growth.