Project Done By: Varsha Biswal

Technology Used: SQL

E-commerce Customer Churn Analysis using SQL

1. INTRODUCTION:

Customer Churn in E-commerce:

- Customer churn refers to when customers stop doing business with a company, which has negative impacts on customer retention and overall success.

- The fast-paced world of online retail presents challenges for e-commerce companies to retain customers.

- Identifying at-risk customers and implementing targeted retention strategies are crucial in reducing customer churn.

2. PROJECT APPROACH:

- The dataset used for analysis was obtained from Kaggle and includes customer information such as personal details, satisfaction scores, payment mode, and more.

- The analysis involved stages like data cleaning, data exploration, insights generation, and recommendations.

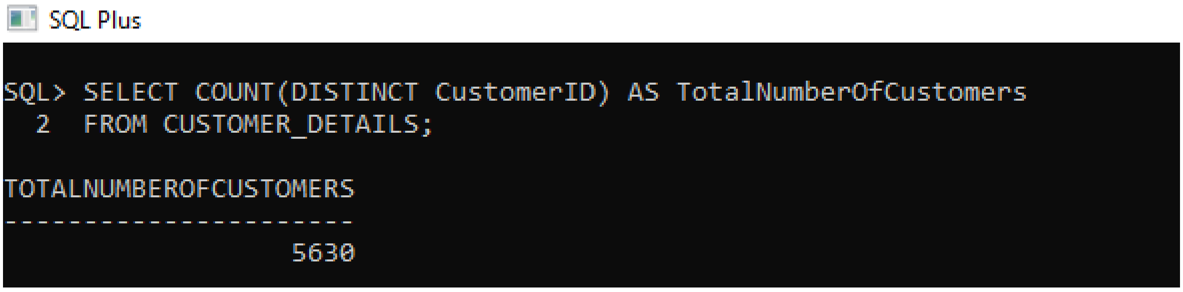
3. DATA CLEANING:

- Ensuring the dataset is clean and reliable is essential for any analysis.

- Handled missing values by filling them with the mean value of their respective columns.

- Created new columns to categorize customer status and complaints for better understanding.

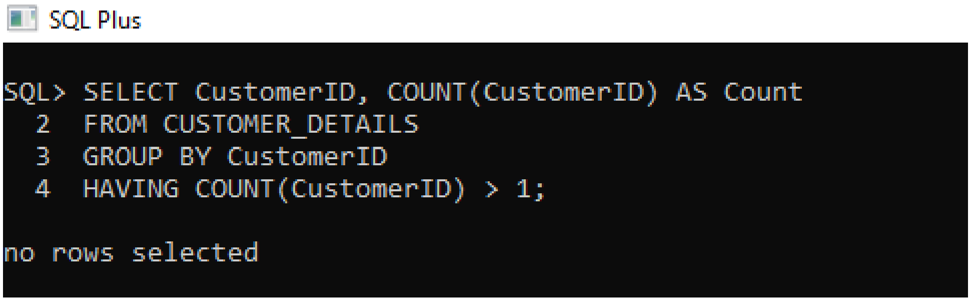
I. Finding the total number of customers



```
SQL> SELECT COUNT(DISTINCT CustomerID) AS TotalNumberOfCustomers
  2  FROM CUSTOMER_DETAILS;

TOTALNUMBEROFCUSTOMERS
----------------------
                  5630
```

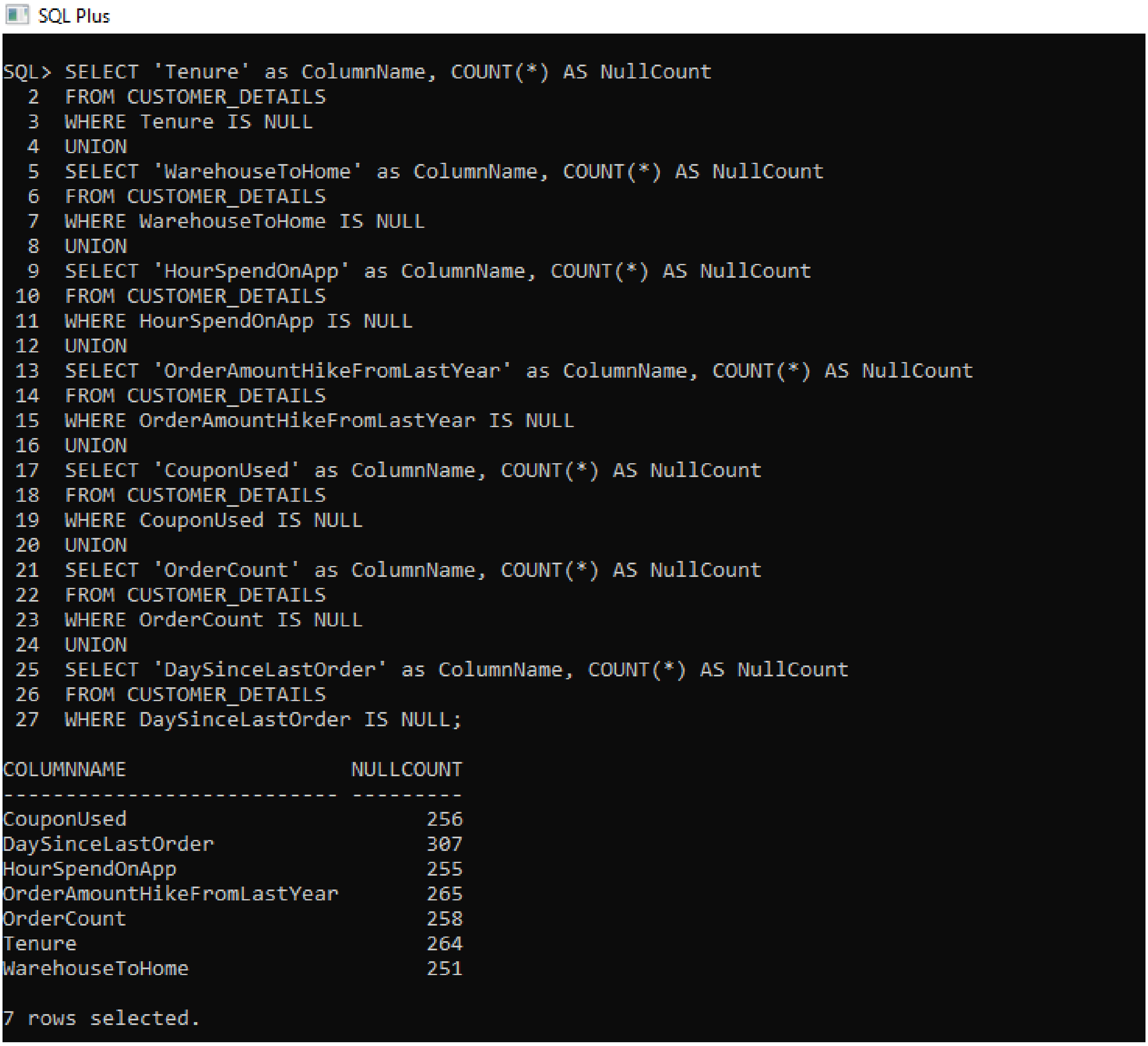
This query will count the number of distinct "CustomerID" values in the "CUSTOMER_DETAILS" table and return the result as "TotalNumberOfCustomers." The DISTINCT keyword ensures that each customer is counted only once, even if they appear multiple times in the table.

II. Checking for duplicate rows



```
SQL> SELECT CustomerID, COUNT(CustomerID) AS Count
  2  FROM CUSTOMER_DETAILS
  3  GROUP BY CustomerID
  4  HAVING COUNT(CustomerID) > 1;

no rows selected
```

III. Checking for null values



```
SQL> SELECT 'Tenure' as ColumnName, COUNT(*) AS NullCount
  2  FROM CUSTOMER_DETAILS
  3  WHERE Tenure IS NULL
  4  UNION
  5  SELECT 'WarehouseToHome' as ColumnName, COUNT(*) AS NullCount
  6  FROM CUSTOMER_DETAILS
  7  WHERE WarehouseToHome IS NULL
  8  UNION
  9  SELECT 'HourSpendOnApp' as ColumnName, COUNT(*) AS NullCount
 10  FROM CUSTOMER_DETAILS
 11  WHERE HourSpendOnApp IS NULL
 12  UNION
 13  SELECT 'OrderAmountHikeFromLastYear' as ColumnName, COUNT(*) AS NullCount
 14  FROM CUSTOMER_DETAILS
 15  WHERE OrderAmountHikeFromLastYear IS NULL
 16  UNION
 17  SELECT 'CouponUsed' as ColumnName, COUNT(*) AS NullCount
 18  FROM CUSTOMER_DETAILS
 19  WHERE CouponUsed IS NULL
 20  UNION
 21  SELECT 'OrderCount' as ColumnName, COUNT(*) AS NullCount
 22  FROM CUSTOMER_DETAILS
 23  WHERE OrderCount IS NULL
 24  UNION
 25  SELECT 'DaySinceLastOrder' as ColumnName, COUNT(*) AS NullCount
 26  FROM CUSTOMER_DETAILS
 27  WHERE DaySinceLastOrder IS NULL;

COLUMNNAME                   NULLCOUNT
--------------------------- ---------
CouponUsed                        256
DaySinceLastOrder                 307
HourSpendOnApp                    255
OrderAmountHikeFromLastYear       265
OrderCount                        258
Tenure                            264
WarehouseToHome                   251

7 rows selected.
```

This query will check for NULL values in each of the specified columns ('Tenure', 'WarehouseToHome', 'HourSpendOnApp', 'OrderAmountHikeFromLastYear', 'CouponUsed', 'OrderCount', 'DaySinceLastOrder') and return the number of NULL values (NullCount) for each column.

The UNION operator combines the results of each individual SELECT query, giving you a single result set that shows the NULL counts for all the specified columns together. Each individual SELECT query retrieves the NULL counts for a specific column and labels it with the corresponding column name using the "ColumnName" alias.

Interpretation:

"CouponUsed": There are 256 NULL values in the "CouponUsed" column.

"DaySinceLastOrder": There are 307 NULL values in the "DaySinceLastOrder" column.

"HourSpendOnApp": There are 255 NULL values in the "HourSpendOnApp" column.

"OrderAmountHikeFromLastYear": There are 265 NULL values in the "OrderAmountHikeFromLastYear" column.

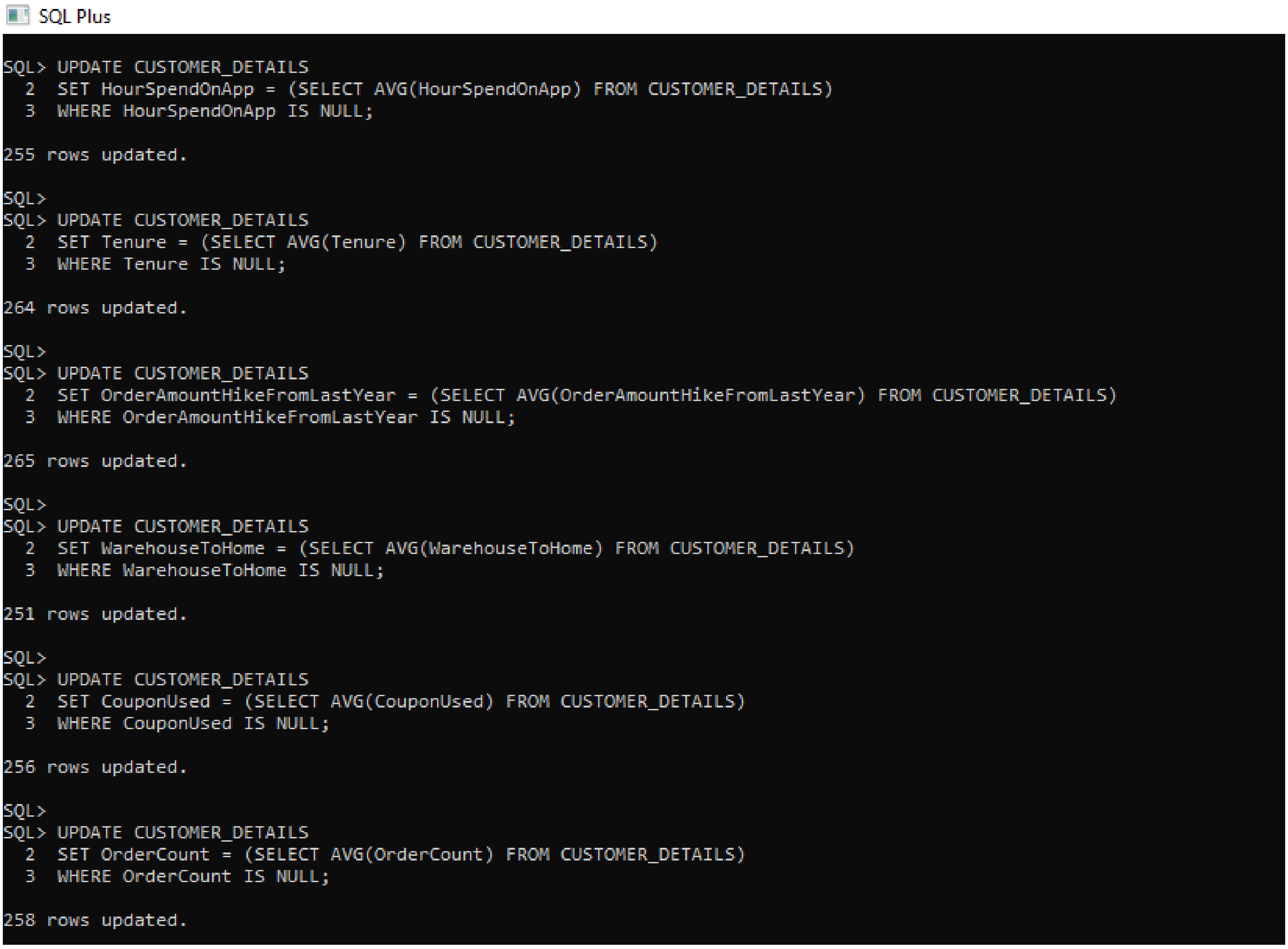
"OrderCount": There are 258 NULL values in the "OrderCount" column.

"Tenure": There are 264 NULL values in the "Tenure" column.

"WarehouseToHome": There are 251 NULL values in the "WarehouseToHome" column.

This information is valuable as it helps identify the number of missing values (NULLs) for each column in the table. We can now consider how to handle or fill these NULL values based on our data analysis or data processing requirements.

IV. Handling null values

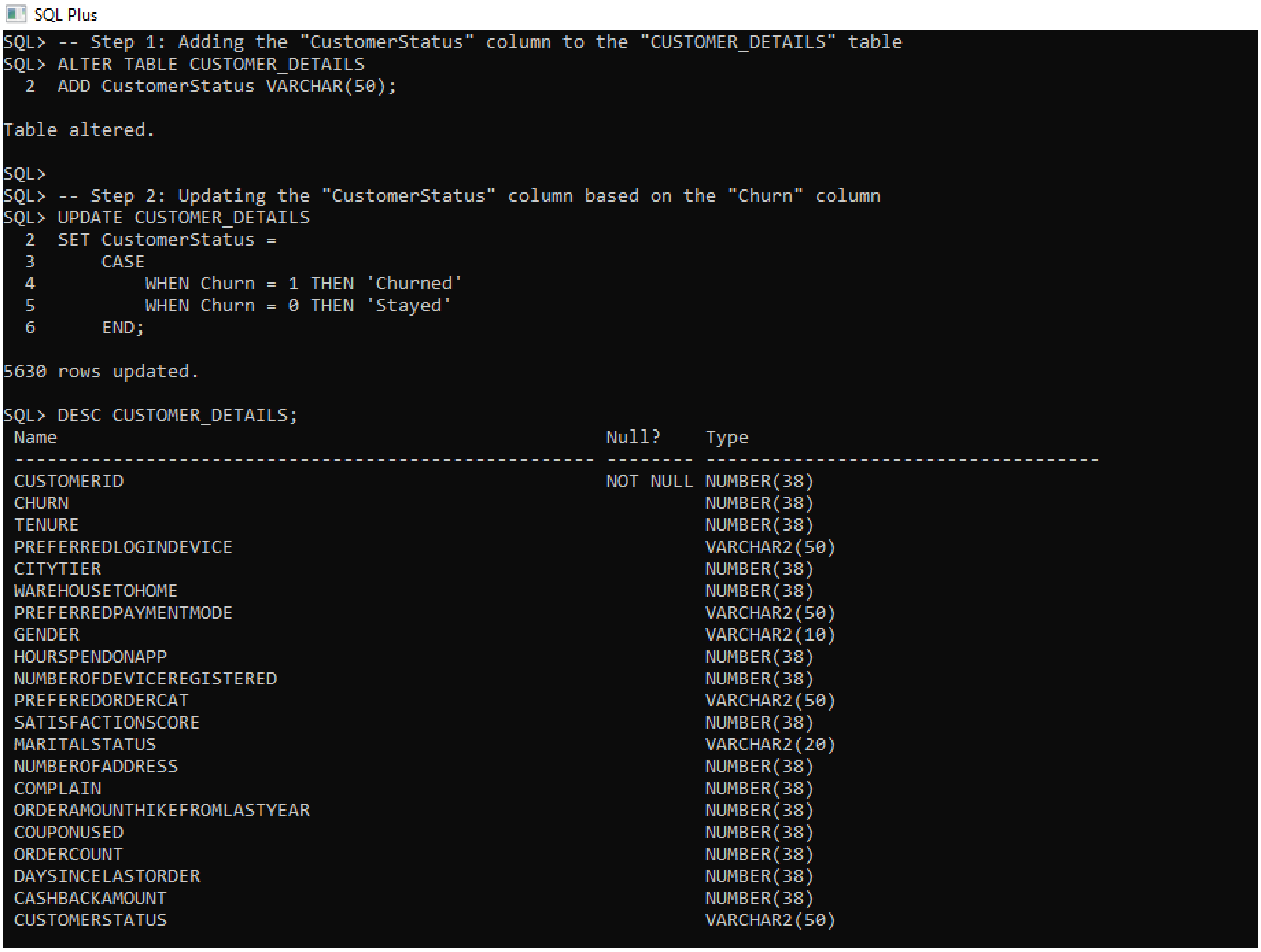


```
SQL> UPDATE CUSTOMER_DETAILS
  2  SET HourSpendOnApp = (SELECT AVG(HourSpendOnApp) FROM CUSTOMER_DETAILS)
  3  WHERE HourSpendOnApp IS NULL;

255 rows updated.

SQL>
SQL> UPDATE CUSTOMER_DETAILS
  2  SET Tenure = (SELECT AVG(Tenure) FROM CUSTOMER_DETAILS)
  3  WHERE Tenure IS NULL;

264 rows updated.

SQL>
SQL> UPDATE CUSTOMER_DETAILS
  2  SET OrderAmountHikeFromLastYear = (SELECT AVG(OrderAmountHikeFromLastYear) FROM CUSTOMER_DETAILS)
  3  WHERE OrderAmountHikeFromLastYear IS NULL;

265 rows updated.

SQL>
SQL> UPDATE CUSTOMER_DETAILS
  2  SET WarehouseToHome = (SELECT AVG(WarehouseToHome) FROM CUSTOMER_DETAILS)
  3  WHERE WarehouseToHome IS NULL;

251 rows updated.

SQL>
SQL> UPDATE CUSTOMER_DETAILS
  2  SET CouponUsed = (SELECT AVG(CouponUsed) FROM CUSTOMER_DETAILS)
  3  WHERE CouponUsed IS NULL;

256 rows updated.

SQL>
SQL> UPDATE CUSTOMER_DETAILS
  2  SET OrderCount = (SELECT AVG(OrderCount) FROM CUSTOMER_DETAILS)
  3  WHERE OrderCount IS NULL;

258 rows updated.
```

This query will update the NULL values in each column (HourSpendOnApp, Tenure, OrderAmountHikeFromLastYear, WarehouseToHome, CouponUsed, OrderCount, DaySinceLastOrder) with the average value of that respective column calculated from the existing data in the "CUSTOMER_DETAILS" table.

V. Creating a new column from an already existing “churn” column

A new column called 'CustomerStatus' is created to indicate whether a customer stayed or churned based on the values of 0 and 1 in the churn column.



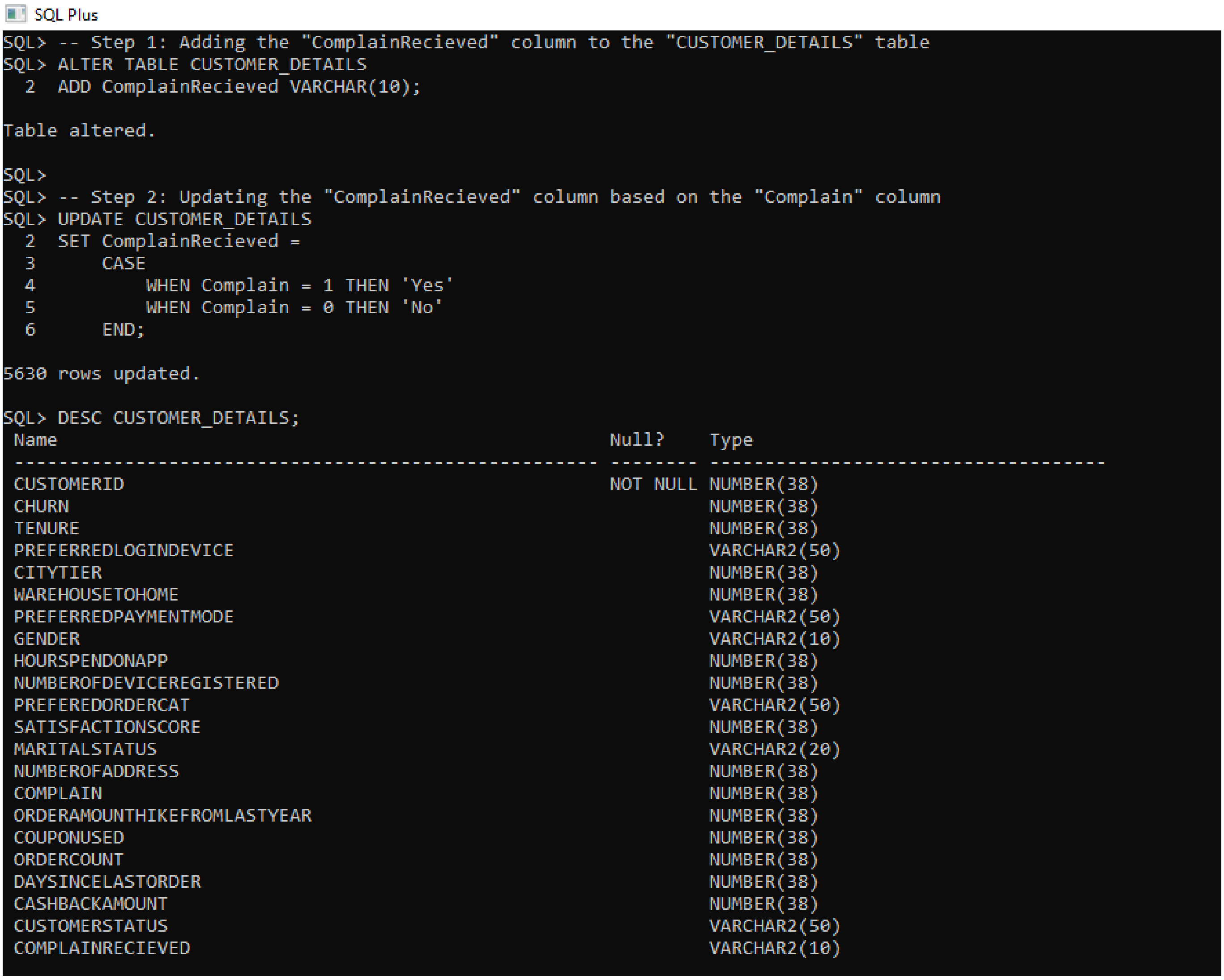
```
SQL> -- Step 1: Adding the "CustomerStatus" column to the "CUSTOMER_DETAILS" table
SQL> ALTER TABLE CUSTOMER_DETAILS
  2  ADD CustomerStatus VARCHAR(50);

Table altered.

SQL>
SQL> -- Step 2: Updating the "CustomerStatus" column based on the "Churn" column
SQL> UPDATE CUSTOMER_DETAILS
  2  SET CustomerStatus =
  3      CASE
  4          WHEN Churn = 1 THEN 'Churned'
  5          WHEN Churn = 0 THEN 'Stayed'
  6      END;

5630 rows updated.

SQL> DESC CUSTOMER_DETAILS;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------------------------------
 CUSTOMERID                                            NOT NULL NUMBER(38)
 CHURN                                                          NUMBER(38)
 TENURE                                                         NUMBER(38)
 PREFERREDLOGINDEVICE                                           VARCHAR2(50)
 CITYTIER                                                       NUMBER(38)
 WAREHOUSETOHOME                                                NUMBER(38)
 PREFERREDPAYMENTMODE                                           VARCHAR2(50)
 GENDER                                                         VARCHAR2(10)
 HOURSPENDONAPP                                                 NUMBER(38)
 NUMBEROFDEVICEREGISTERED                                       NUMBER(38)
 PREFEREDORDERCAT                                               VARCHAR2(50)
 SATISFACTIONSCORE                                              NUMBER(38)
 MARITALSTATUS                                                  VARCHAR2(20)
 NUMBEROFADDRESS                                                NUMBER(38)
 COMPLAIN                                                       NUMBER(38)
 ORDERAMOUNTHIKEFROMLASTYEAR                                    NUMBER(38)
 COUPONUSED                                                     NUMBER(38)
 ORDERCOUNT                                                     NUMBER(38)
 DAYSINCELASTORDER                                              NUMBER(38)
 CASHBACKAMOUNT                                                 NUMBER(38)
 CUSTOMERSTATUS                                                 VARCHAR2(50)
```

In this query, we first add a new column called "CustomerStatus" to the "CUSTOMER_DETAILS" table with a data type of VARCHAR(50). The ALTER TABLE statement allows us to modify the table structure by adding the new column.

Next, the UPDATE statement is used to set the values of the "CustomerStatus" column based on the values of the "Churn" column. If the "Churn" column has a value of 1, the corresponding "CustomerStatus" value will be 'Churned'. If the "Churn" column has a value of 0, the corresponding "CustomerStatus" value will be 'Stayed'.

VI. Creating a new column from an already existing "complain" column

A new column called 'ComplainReceived' is created in the "CUSTOMER_DETAILS" table to indicate whether a customer complained or not, with "No" representing no complaint and "Yes" representing a complaint.



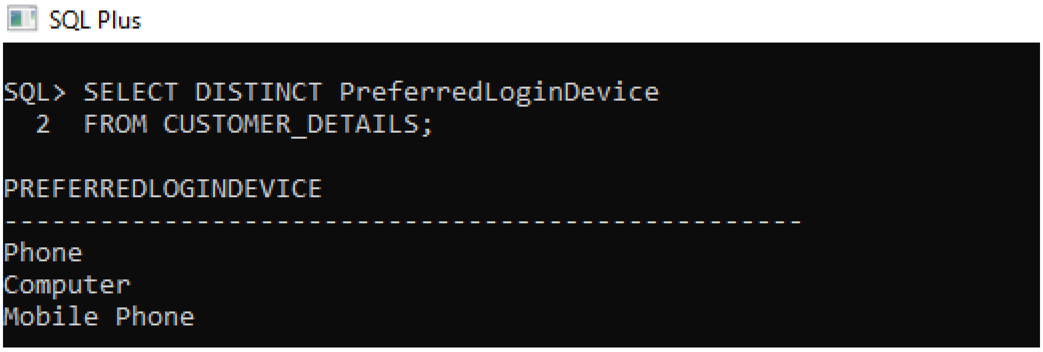
```
SQL> -- Step 1: Adding the "ComplainRecieved" column to the "CUSTOMER_DETAILS" table
SQL> ALTER TABLE CUSTOMER_DETAILS
     ADD ComplainRecieved VARCHAR(10);

Table altered.

SQL>
SQL> -- Step 2: Updating the "ComplainRecieved" column based on the "Complain" column
SQL> UPDATE CUSTOMER_DETAILS
     SET ComplainRecieved =
         CASE
             WHEN Complain = 1 THEN 'Yes'
             WHEN Complain = 0 THEN 'No'
         END;

5630 rows updated.

SQL> DESC CUSTOMER_DETAILS;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------------------------------
 CUSTOMERID                                            NOT NULL NUMBER(38)
 CHURN                                                          NUMBER(38)
 TENURE                                                         NUMBER(38)
 PREFERREDLOGINDEVICE                                           VARCHAR2(50)
 CITYTIER                                                       NUMBER(38)
 WAREHOUSETOHOME                                                NUMBER(38)
 PREFERREDPAYMENTMODE                                           VARCHAR2(50)
 GENDER                                                         VARCHAR2(10)
 HOURSPENDONAPP                                                 NUMBER(38)
 NUMBEROFDEVICEREGISTERED                                       NUMBER(38)
 PREFEREDORDERCAT                                               VARCHAR2(50)
 SATISFACTIONSCORE                                              NUMBER(38)
 MARITALSTATUS                                                  VARCHAR2(20)
 NUMBEROFADDRESS                                                NUMBER(38)
 COMPLAIN                                                       NUMBER(38)
 ORDERAMOUNTHIKEFROMLASTYEAR                                    NUMBER(38)
 COUPONUSED                                                     NUMBER(38)
 ORDERCOUNT                                                     NUMBER(38)
 DAYSINCELASTORDER                                              NUMBER(38)
 CASHBACKAMOUNT                                                 NUMBER(38)
 CUSTOMERSTATUS                                                 VARCHAR2(50)
 COMPLAINRECIEVED                                               VARCHAR2(10)
```

In this query, we first add a new column called "ComplainRecieved" to the "CUSTOMER_DETAILS" table with a data type of VARCHAR(10). The ALTER TABLE statement allows us to modify the table structure by adding the new column.

Next, the UPDATE statement is used to set the values of the "ComplainRecieved" column based on the values of the "Complain" column. If the "Complain" column has a value of 1, the corresponding "ComplainRecieved" value will be 'Yes'. If the "Complain" column has a value of 0, the corresponding "ComplainRecieved" value will be 'No'.

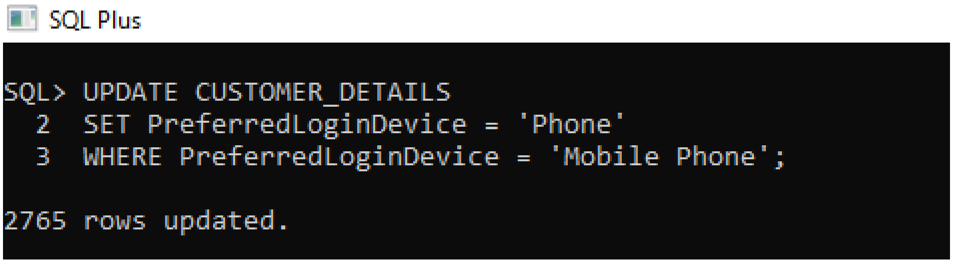
VII. Checking values in each column for correctness and accuracy

a. Fixing redundancy in “PreferedLoginDevice” Column



```
SQL> SELECT DISTINCT PreferredLoginDevice
  2  FROM CUSTOMER_DETAILS;

PREFERREDLOGINDEVICE
--------------------------------------------------
Phone
Computer
Mobile Phone
```

Notice that phone and mobile phone appear in the column, but they mean the same thing. So we will replace the mobile phone with phone.

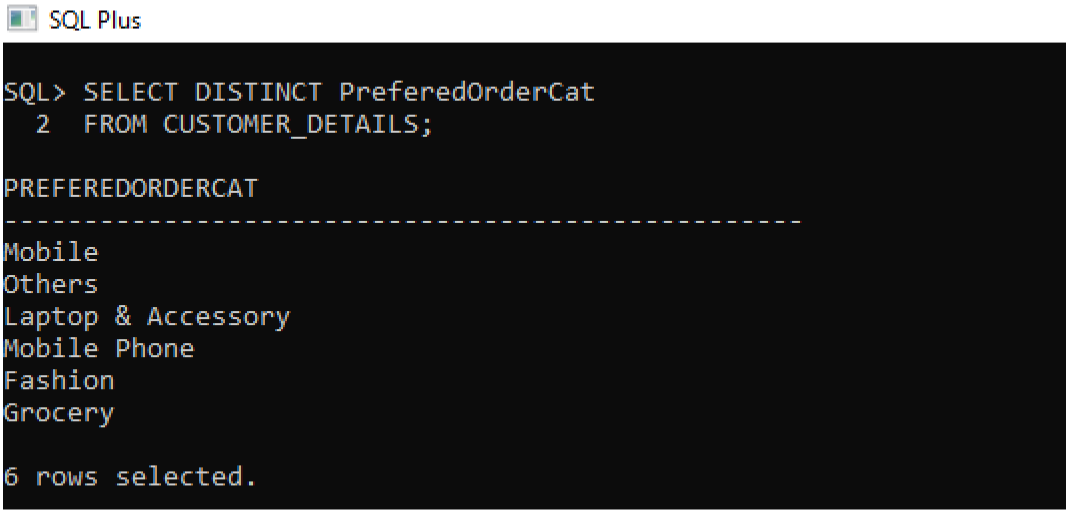


```
SQL> UPDATE CUSTOMER_DETAILS
  2  SET PreferredLoginDevice = 'Phone'
  3  WHERE PreferredLoginDevice = 'Mobile Phone';

2765 rows updated.
```

In this query, we use the UPDATE statement to change the values in the "PreferredLoginDevice" column of the "CUSTOMER_DETAILS" table. The SET clause sets the new value 'phone', and the WHERE clause specifies the condition that the current value must be 'mobile phone' for the update to take place.

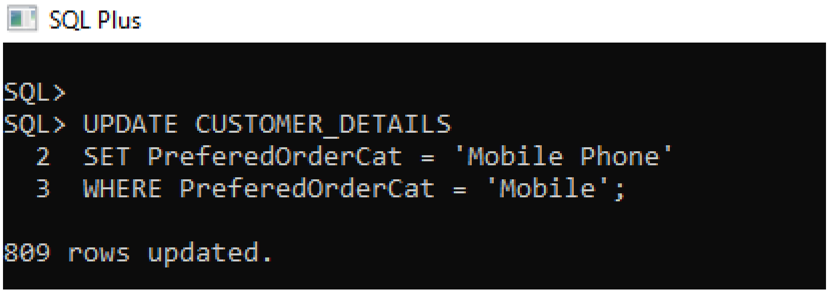
After executing this query, all occurrences of 'mobile phone' in the "PreferredLoginDevice" column will be updated to 'phone'. This allows you to standardize the representation of login devices in the table.

b. Fixing redundancy in “PreferedOrderCat” Column



```
SQL> SELECT DISTINCT PreferedOrderCat
  2  FROM CUSTOMER_DETAILS;

PREFEREDORDERCAT
--------------------------------------------------
Mobile
Others
Laptop & Accessory
Mobile Phone
Fashion
Grocery

6 rows selected.
```

Notice that mobile phone and phone appear in the column, but they mean the same thing. So we will replace the phone with mobile phone as the category name.

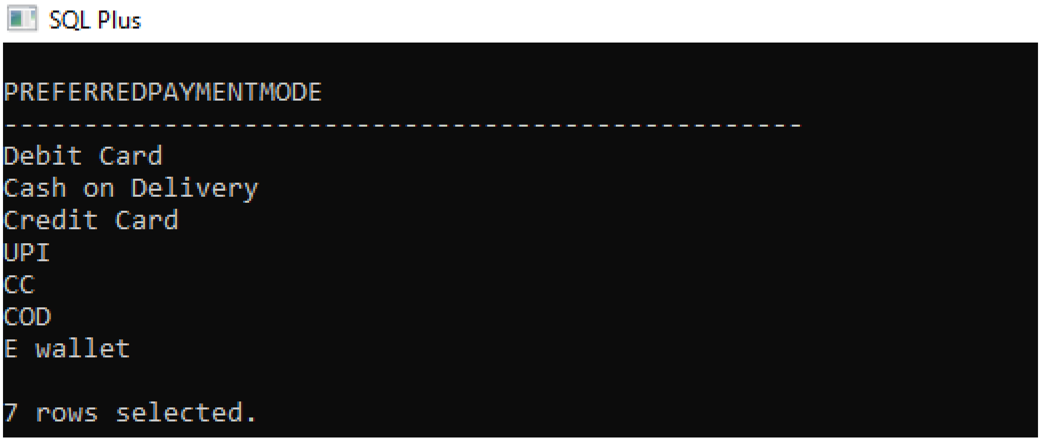


```
SQL>
SQL> UPDATE CUSTOMER_DETAILS
  2  SET PreferedOrderCat = 'Mobile Phone'
  3  WHERE PreferedOrderCat = 'Mobile';

809 rows updated.
```

In this query, we use the UPDATE statement to change the values in the "PreferedOrderCat" column of your specified table (e.g., "CUSTOMER_DETAILS"). The SET clause sets the new value 'Mobile Phone', and the WHERE clause specifies the condition that the current value must be 'Mobile' for the update to take place.

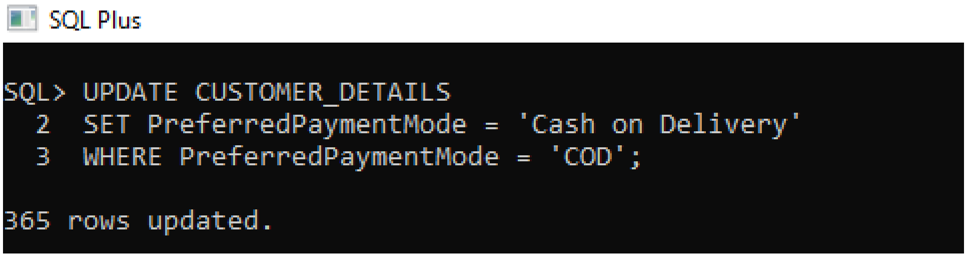
After executing this query, all occurrences of 'Mobile' in the "PreferedOrderCat" column will be updated to 'Mobile Phone'. This allows you to standardize the representation of preferred order categories in the table.

c. Fixing redundancy in “PreferredPaymentMode” Column



```
PREFERREDPAYMENTMODE
--------------------------------------------------
Debit Card
Cash on Delivery
Credit Card
UPI
CC
COD
E wallet

7 rows selected.
```

Notice that Cash on Delivery and COD both appear in the column, but they mean the same thing. So we will replace COD with Cash on Delivery.

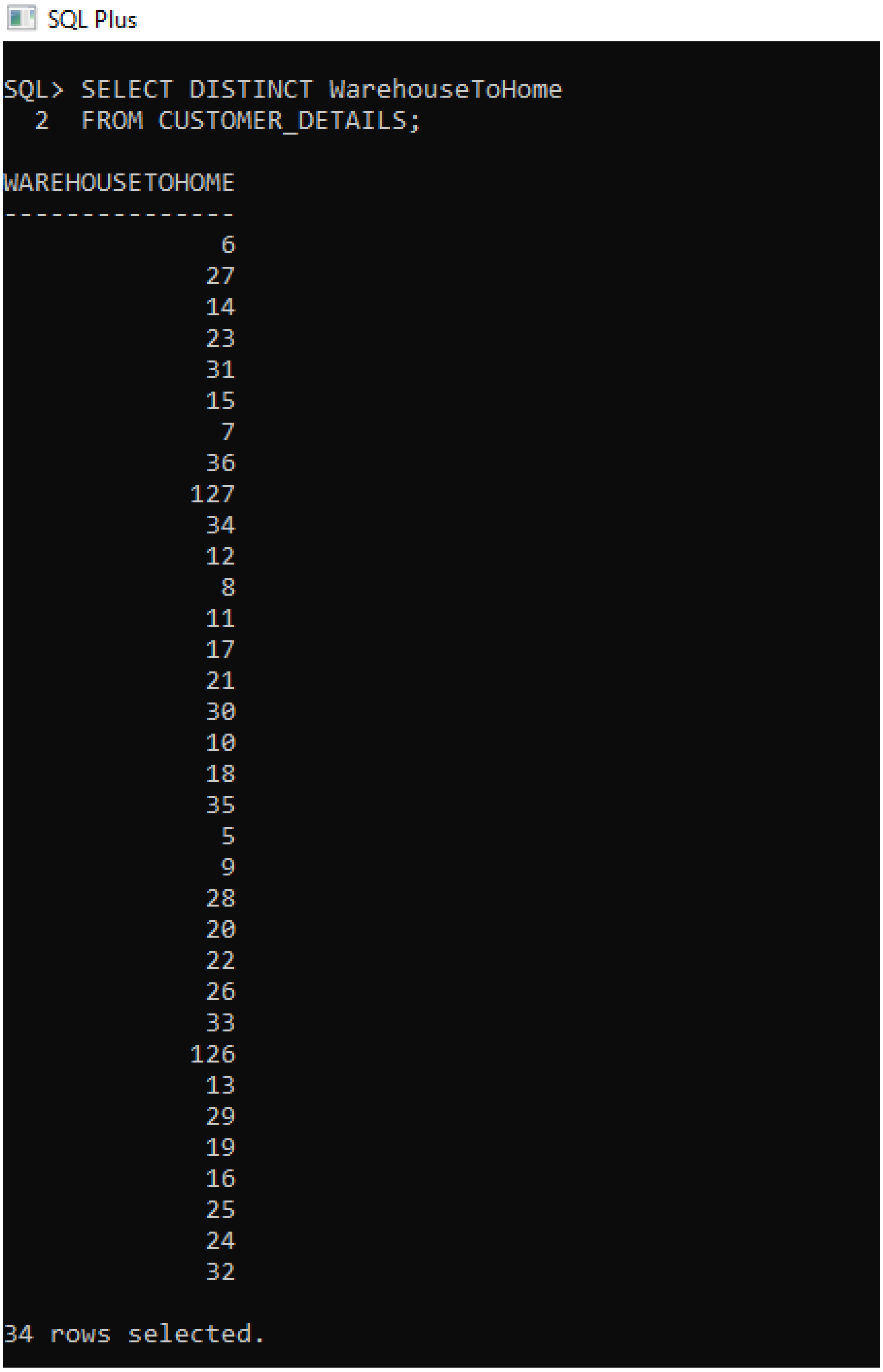


```
SQL> UPDATE CUSTOMER_DETAILS
  2  SET PreferredPaymentMode = 'Cash on Delivery'
  3  WHERE PreferredPaymentMode = 'COD';

365 rows updated.
```

In this query, we use the UPDATE statement to change the values in the "PreferredPaymentMode" column of your specified table (e.g., "CUSTOMER_DETAILS"). The SET clause sets the new value 'Cash on Delivery', and the WHERE clause specifies the condition that the current value must be 'COD' for the update to take place.

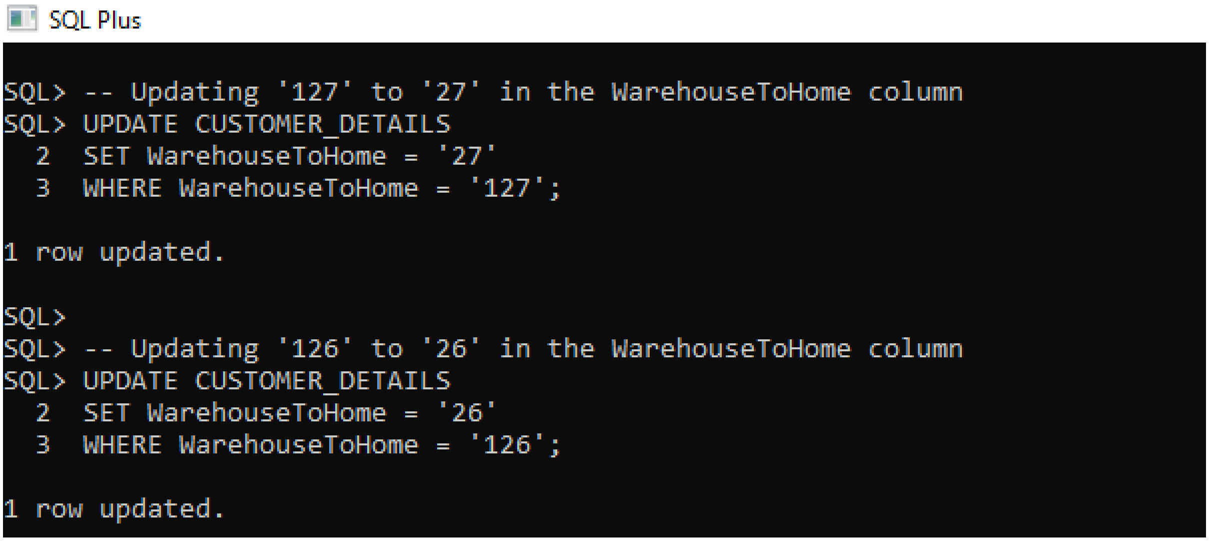
After executing this query, all occurrences of 'COD' in the "PreferredPaymentMode" column will be updated to 'Cash on Delivery'. This allows you to standardize the representation of preferred payment modes in the table.

d. Fixing wrongly entered values in “WarehouseToHome” column



```
SQL> SELECT DISTINCT WarehouseToHome
  2  FROM CUSTOMER_DETAILS;

WAREHOUSETOHOME
---------------
              6
             27
             14
             23
             31
             15
              7
             36
            127
             34
             12
              8
             11
             17
             21
             30
             10
             18
             35
              5
              9
             28
             20
             22
             26
             33
            126
             13
             29
             19
             16
             25
             24
             32

34 rows selected.
```

The table shows a sample of the entire result. However, notice the 126 and 127 values; they are definitely outliers and most likely wrongly entered. To fix this, we will correct the values to 26 and 27 to fall within the range of the values found in this column.



```
SQL> -- Updating '127' to '27' in the WarehouseToHome column
SQL> UPDATE CUSTOMER_DETAILS
     SET WarehouseToHome = '27'
     WHERE WarehouseToHome = '127';

1 row updated.

SQL>
SQL> -- Updating '126' to '26' in the WarehouseToHome column
SQL> UPDATE CUSTOMER_DETAILS
     SET WarehouseToHome = '26'
     WHERE WarehouseToHome = '126';

1 row updated.
```

In these queries, we use the UPDATE statement to change values in the "WarehouseToHome" column of your specified table (e.g., "CUSTOMER_DETAILS"). The SET clause sets the new values '27' and '26', and the WHERE clause specifies the conditions that the current values must be '127' and '126', respectively, for the updates to take place.

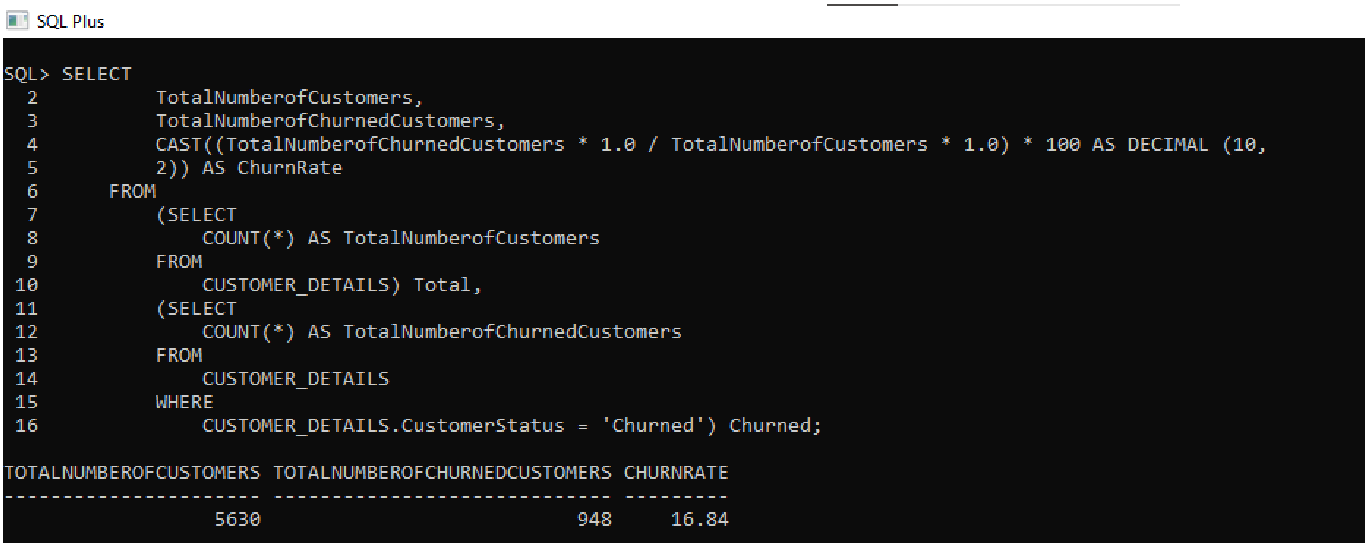
After executing these queries, all occurrences of '127' in the "WarehouseToHome" column will be updated to '27', and all occurrences of '126' will be updated to '26'. This allows you to correct or standardize the recorded distances between customers' warehouse and home addresses in your table.

Our data has been cleaned and is now ready to be explored for insight generation.

4. DATA EXPLORATION, INSIGHTS AND RECOMMENDATIONS:

- ➢ Explored the dataset to answer important business questions.

1. What is the overall customer churn rate?



```
SQL> SELECT
            TotalNumberofCustomers,
            TotalNumberofChurnedCustomers,
            CAST((TotalNumberofChurnedCustomers * 1.0 / TotalNumberofCustomers * 1.0) * 100 AS DECIMAL (10,
            2)) AS ChurnRate
        FROM
            (SELECT
                COUNT(*) AS TotalNumberofCustomers
            FROM
                CUSTOMER_DETAILS) Total,
            (SELECT
                COUNT(*) AS TotalNumberofChurnedCustomers
            FROM
                CUSTOMER_DETAILS
            WHERE
                CUSTOMER_DETAILS.CustomerStatus = 'Churned') Churned;

TOTALNUMBEROFCUSTOMERS TOTALNUMBEROFCHURNEDCUSTOMERS CHURNRATE
---------------------- ----------------------------- ---------
                  5630                           948     16.84
```

In this query, we use two subqueries to calculate the total number of customers and the total number of churned customers. The first subquery calculates the total number of customers in the "CUSTOMER_DETAILS" table, and the second subquery calculates the total number of customers who have churned (based on the "CustomerStatus" column).

Then, we perform a SELECT statement on the two subqueries to retrieve the "TotalNumberofCustomers" and "TotalNumberofChurnedCustomers" values. Additionally, we calculate the churn rate by dividing the "TotalNumberofChurnedCustomers" by the "TotalNumberofCustomers" and multiplying by 100 to get the percentage. The CAST function is used to convert the result to a decimal with two decimal places (DECIMAL(10, 2)).

Interpretation:

Churn Rate: The churn rate of 16.84% indicates that approximately 16.84% of the total customers who were using the e-commerce platform at a given time decided to discontinue their engagement with the platform. In other words, about 16.84 out of every 100 customers stopped using the platform during the specified period.

Importance: A churn rate of 16.84% is relatively high, which suggests that a significant portion of customers are no longer transacting with the platform. High churn rates can have adverse effects on an ecommerce business, leading to lost revenue, decreased customer lifetime value, and increased customer acquisition costs to compensate for the loss.

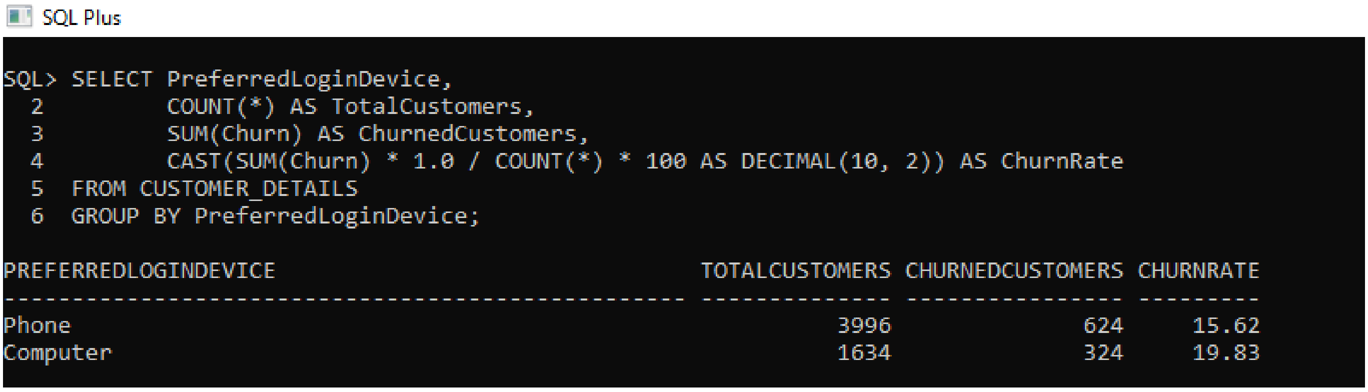
Business Implications: To improve the business's performance and revenue generation, it is crucial to reduce the churn rate. Strategies to retain customers, enhance customer experience, and address any issues or complaints that might be driving the churn are essential. Regularly analyzing the churn rate and taking proactive measures to retain customers can lead to improved customer loyalty and increased revenue.

Benchmarking: It's essential to compare the current churn rate to historical data and industry benchmarks to gain insights into how the business is performing compared to its previous performance and competitors. If the churn rate is higher than industry averages, it may indicate a need for urgent action to improve customer retention efforts.

Retention Efforts: Businesses can implement various customer retention initiatives, such as personalized marketing, loyalty programs, exceptional customer service, and addressing customer feedback promptly. These efforts can help reduce the churn rate and increase customer satisfaction, leading to improved customer loyalty.

Continuous Monitoring: Monitoring the churn rate regularly and analyzing trends over time is vital. It allows the business to identify the effectiveness of its retention strategies and make data-driven decisions to enhance customer experience and reduce churn.

2. How does the churn rate vary based on the preferred login device?



```
SQL> SELECT PreferredLoginDevice,
            COUNT(*) AS TotalCustomers,
            SUM(Churn) AS ChurnedCustomers,
            CAST(SUM(Churn) * 1.0 / COUNT(*) * 100 AS DECIMAL(10, 2)) AS ChurnRate
     FROM CUSTOMER_DETAILS
     GROUP BY PreferredLoginDevice;

PREFERREDLOGINDEVICE                               TOTALCUSTOMERS CHURNEDCUSTOMERS CHURNRATE
-------------------------------------------------- -------------- ---------------- ---------
Phone                                                        3996              624     15.62
Computer                                                     1634              324     19.83
```

In this query, we use the SELECT statement to retrieve the preferred login device, total number of customers for each device, the number of churned customers for each device, and the churn rate for each device. We GROUP BY the "PreferredLoginDevice" column to aggregate the results for each unique login device.

The COUNT(*) function calculates the total number of customers for each login device, and the SUM(Churn) function calculates the total number of churned customers for each device (since "Churn" is assumed to be a 1 or 0, Summing it gives the count of churned customers).

The expression SUM(Churn) * 1.0 / COUNT(*) * 100 calculates the churn rate for each login device. We multiply by 1.0 to ensure a decimal result and then divide the sum of churned customers by the total number of customers to get the churn rate. Finally, we multiply by 100 to express the result as a percentage. The CAST function is used to convert the result to a decimal with two decimal places (DECIMAL(10, 2)).

Interpretation:

Phone: The churn rate for customers who prefer to log in using a phone is 15.62%. This means that approximately 15.62 out of every 100 customers who use a phone to log in have stopped using the platform or churned during the specified period.

Computer: The churn rate for customers who prefer to log in using a computer is 19.83%. This indicates that approximately 19.83 out of every 100 customers who use a computer to log in have churned during the specified period.

Comparison:

The churn rate for the "PHONE" login device (15.62%) is lower than the churn rate for the "COMPUTER" login device (19.83%). This suggests that customers who log in using a phone are more likely to stay or have a lower churn rate compared to those who log in using a computer.

Business Implications:

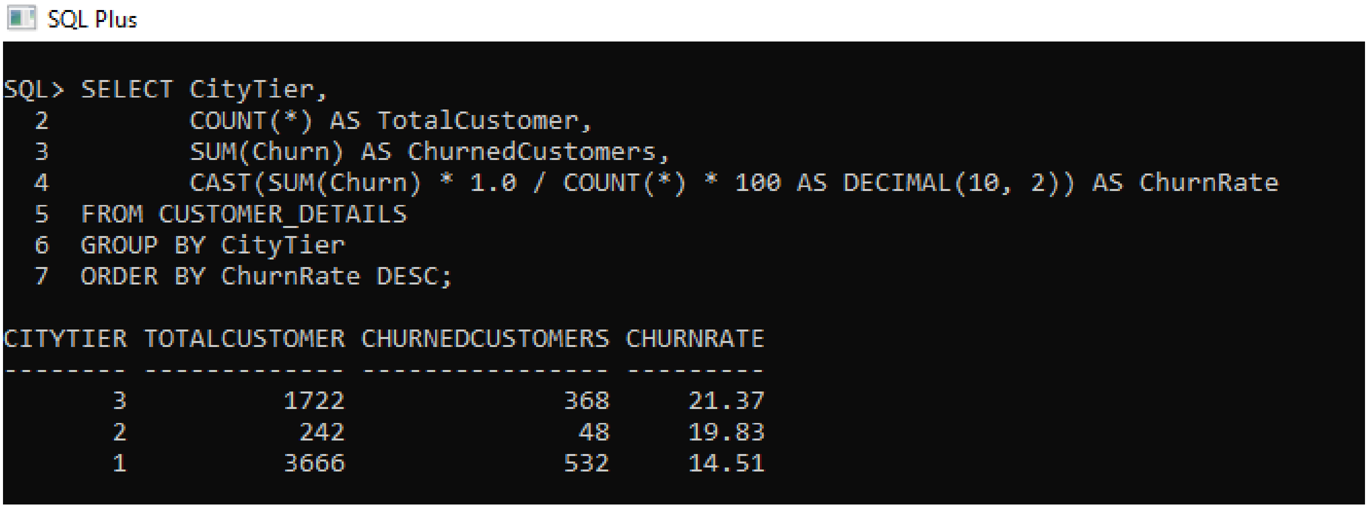
The difference in churn rates for different login devices could indicate that there might be differences in customer behavior, preferences, or experiences between customers who use phones and those who use computers to access the e-commerce platform.

To improve customer retention and reduce churn, the e-commerce platform may consider focusing on understanding the factors influencing churn for each login device. This could involve investigating potential issues or pain points specific to each device and implementing targeted strategies to address them.

Additionally, the platform may consider leveraging the insights gained from this analysis to design personalized experiences for customers based on their preferred login devices. By offering tailored experiences, the platform may increase customer satisfaction and loyalty, which could positively impact the churn rate.

Continuous monitoring of the churn rates for different login devices over time will provide valuable feedback on the effectiveness of retention strategies and whether any improvements are needed to optimize customer retention efforts.

3. What is the distribution of customers across different city tiers?



```
SQL> SELECT CityTier,
            COUNT(*) AS TotalCustomer,
            SUM(Churn) AS ChurnedCustomers,
            CAST(SUM(Churn) * 1.0 / COUNT(*) * 100 AS DECIMAL(10, 2)) AS ChurnRate
     FROM CUSTOMER_DETAILS
     GROUP BY CityTier
     ORDER BY ChurnRate DESC;

CITYTIER TOTALCUSTOMER CHURNEDCUSTOMERS CHURNRATE
-------- ------------- ---------------- ---------
       3          1722              368     21.37
       2           242               48     19.83
       1          3666              532     14.51
```

In this query, we use the SELECT statement to retrieve the city tier, total number of customers for each city tier, the number of churned customers for each city tier, and the churn rate for each city tier. We GROUP BY the "CityTier" column to aggregate the results for each unique city tier.

Interpretation:

City Tier 3: The churn rate for customers in City Tier 3 is 21.37%. This means that approximately 21.37 out of every 100 customers in City Tier 3 have stopped using the platform or churned during the specified period.

City Tier 2: The churn rate for customers in City Tier 2 is 19.83%. This indicates that approximately 19.83 out of every 100 customers in City Tier 2 have churned during the specified period.

City Tier 1: The churn rate for customers in City Tier 1 is 14.51%. This suggests that approximately 14.51 out of every 100 customers in City Tier 1 have churned during the specified period.

Comparison:

The churn rates are presented in descending order, indicating that City Tier 3 has the highest churn rate, followed by City Tier 2 and then City Tier 1. This means that customers in City Tier 3 are more likely to churn compared to those in City Tier 2 and City Tier 1.

Business Implications:

The differences in churn rates between different city tiers could indicate variations in customer behaviors, preferences, or experiences based on geographical locations.

The e-commerce platform can leverage this information to focus on retention efforts in City Tier 3, where the churn rate is the highest. Understanding the reasons behind the higher churn rate in this tier and implementing targeted strategies to address them may help improve customer retention.

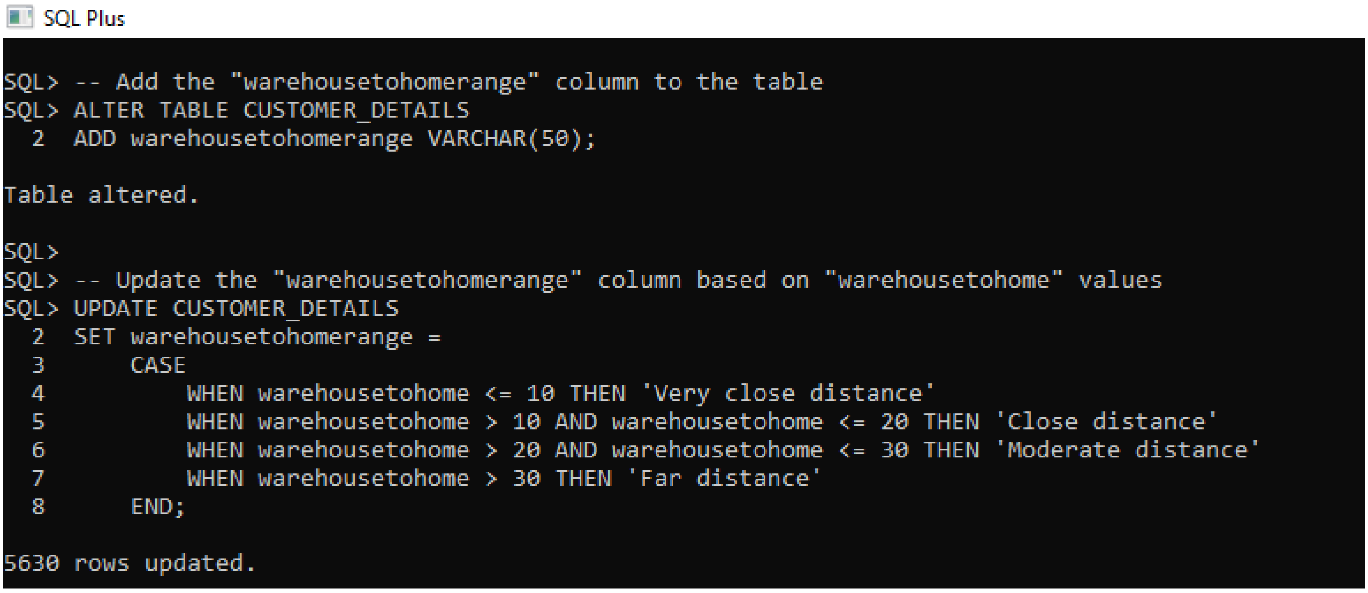
Similarly, for City Tiers 2 and 1, the platform can analyze factors contributing to churn and tailor retention strategies to suit the specific needs and preferences of customers in these tiers.

Continuous monitoring and analysis of churn rates for different city tiers over time will provide valuable insights into the effectiveness of customer retention initiatives. Adjustments to strategies can be made based on the outcomes observed.

Offering personalized experiences, special promotions, or targeted marketing campaigns to customers in different city tiers may help improve customer satisfaction and loyalty, leading to a potential reduction in churn.

4. Is there any correlation between the warehouse-to-home distance and customer churn?

We will create a new column called “WarehouseToHomeRange” that groups the distance into very close, close, moderate, and far using the CASE statement.



```
SQL> -- Add the "warehousetohomerange" column to the table
SQL> ALTER TABLE CUSTOMER_DETAILS
  2  ADD warehousetohomerange VARCHAR(50);

Table altered.

SQL>
SQL> -- Update the "warehousetohomerange" column based on "warehousetohome" values
SQL> UPDATE CUSTOMER_DETAILS
  2  SET warehousetohomerange =
  3      CASE
  4          WHEN warehousetohome <= 10 THEN 'Very close distance'
  5          WHEN warehousetohome > 10 AND warehousetohome <= 20 THEN 'Close distance'
  6          WHEN warehousetohome > 20 AND warehousetohome <= 30 THEN 'Moderate distance'
  7          WHEN warehousetohome > 30 THEN 'Far distance'
  8      END;

5630 rows updated.
```

In this query, we first use the ALTER TABLE statement to add a new column named "warehousetohomerange" to the "CUSTOMER_DETAILS" table. The data type for this new column is set to VARCHAR(50).

Next, we use the UPDATE statement to populate the "warehousetohomerange" column based on the values of the "warehousetohome" column. The CASE statement is used to define the conditions for assigning specific range values based on the "warehousetohome" values. The ranges are as follows:

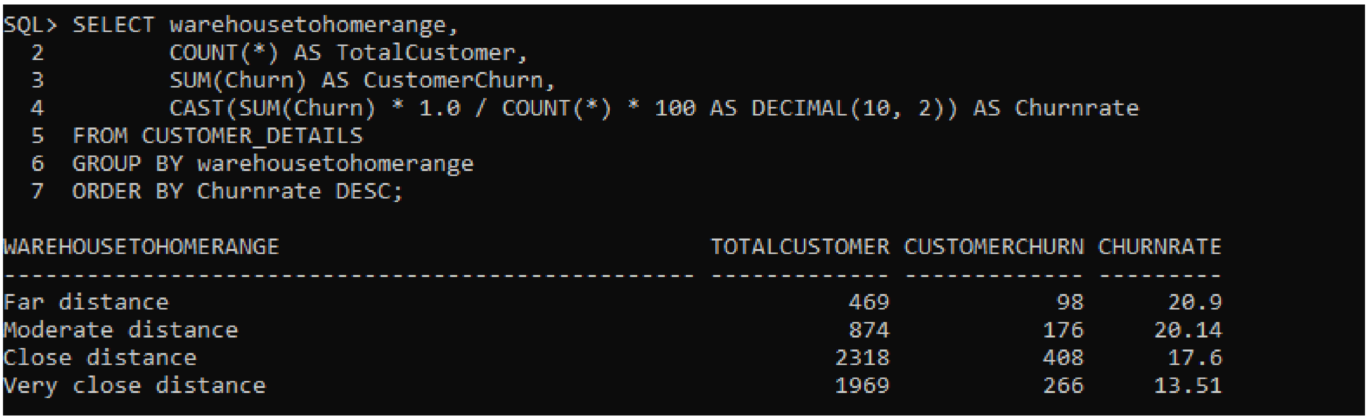
If the "warehousetohome" value is less than or equal to 10, the "warehousetohomerange" is set to 'Very close distance'.

If the "warehousetohome" value is greater than 10 and less than or equal to 20, the "warehousetohomerange" is set to 'Close distance'.

If the "warehousetohome" value is greater than 20 and less than or equal to 30, the "warehousetohomerange" is set to 'Moderate distance'.

If the "warehousetohome" value is greater than 30, the "warehousetohomerange" is set to 'Far distance'.

Finding a correlation between warehouse to home and churn rate.



```
SQL> SELECT warehousetohomerange,
            COUNT(*) AS TotalCustomer,
            SUM(Churn) AS CustomerChurn,
            CAST(SUM(Churn) * 1.0 / COUNT(*) * 100 AS DECIMAL(10, 2)) AS Churnrate
     FROM CUSTOMER_DETAILS
     GROUP BY warehousetohomerange
     ORDER BY Churnrate DESC;

WAREHOUSETOHOMERANGE                           TOTALCUSTOMER CUSTOMERCHURN CHURNRATE
---------------------------------------------- ------------- ------------- ---------
Far distance                                             469            98      20.9
Moderate distance                                        874           176     20.14
Close distance                                          2318           408      17.6
Very close distance                                     1969           266     13.51
```

In this query, we use the SELECT statement to retrieve the "warehousetohomerange," total number of customers for each range, the number of churned customers for each range, and the churn rate for each "warehousetohomerange." We GROUP BY the "warehousetohomerange" column to aggregate the results for each unique range.

"Far distance": In the "Far distance" category, there are 469 customers. Out of these, 98 customers have churned during the specified period. The churn rate for the "Far distance" category is approximately 20.9%. This means that around 20.9 out of every 100 customers who have their warehouse far from their home address have churned.

"Moderate distance": In the "Moderate distance" category, there are 874 customers. Out of these, 176 customers have churned during the specified period. The churn rate for the "Moderate distance" category is approximately 20.14%. This means that around 20.14 out of every 100 customers who have a moderate distance between their warehouse and home address have churned.

"Close distance": In the "Close distance" category, there are 2,318 customers. Out of these, 408 customers have churned during the specified period. The churn rate for the "Close distance" category is approximately 17.6%. This means that around 17.6 out of every 100 customers who have a close distance between their warehouse and home address have churned.

"Very close distance": In the "Very close distance" category, there are 1,969 customers. Out of these, 266 customers have churned during the specified period. The churn rate for the "Very close distance" category is approximately 13.51%. This means that around 13.51 out of every 100 customers who have a very close distance between their warehouse and home address have churned.

Comparison:

The output shows that the churn rate tends to increase with increasing distance between the warehouse and home address. Customers with a "Far distance" and "Moderate distance" have relatively higher churn rates compared to those with "Close distance" and "Very close distance."

Customers with "Very close distance" have the lowest churn rate among the categories, suggesting that customers living very close to the warehouse have a higher likelihood of continuing their relationship with the ecommerce platform.

Business Implications:

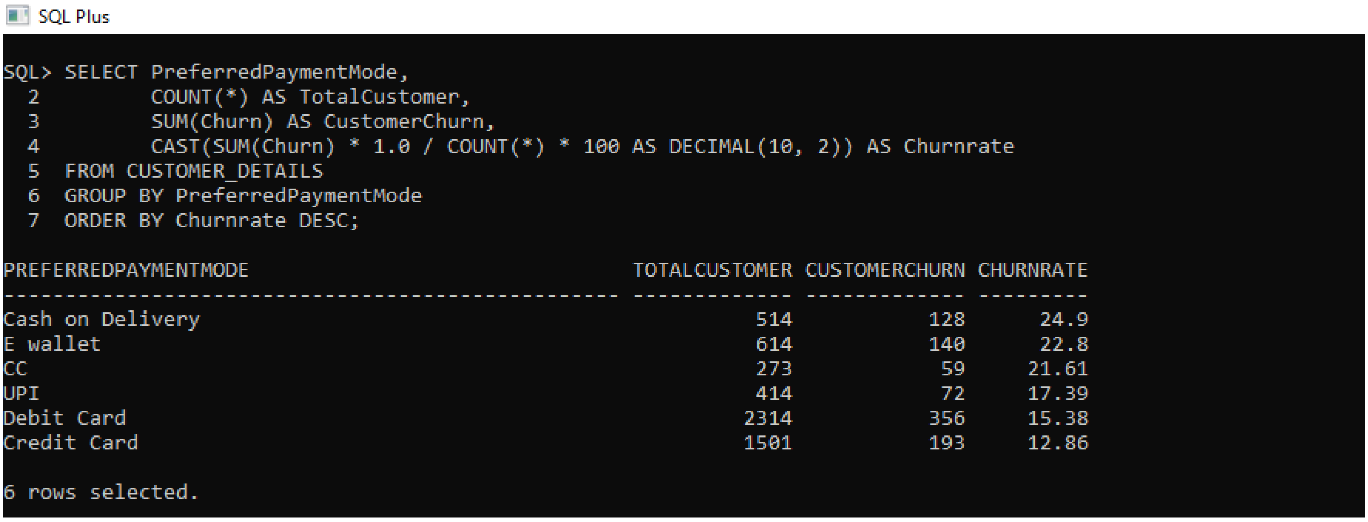
The analysis suggests that customers with warehouses located far from their homes or moderately distant are more likely to churn. This could indicate that customers who face longer delivery times or logistical challenges may be more prone to discontinuing their engagement with the platform.

Businesses can leverage this information to focus on improving the delivery experience for customers with warehouses located at far and moderate distances. Implementing faster and more efficient delivery solutions for these customers may help reduce churn and improve customer satisfaction.

For customers with "Close distance" and "Very close distance," retention efforts may need to be prioritized differently. Offering personalized experiences, loyalty rewards, or special promotions for these customers may help maintain their loyalty and reduce the likelihood of churn.

Continuous monitoring of churn rates based on warehouse-to-home distance can provide valuable insights for refining logistics strategies, optimizing warehouse locations, and improving overall customer experience.

5. Which is the most preferred payment mode among churned customers?



```
SQL> SELECT PreferredPaymentMode,
            COUNT(*) AS TotalCustomer,
            SUM(Churn) AS CustomerChurn,
            CAST(SUM(Churn) * 1.0 / COUNT(*) * 100 AS DECIMAL(10, 2)) AS Churnrate
     FROM CUSTOMER_DETAILS
     GROUP BY PreferredPaymentMode
     ORDER BY Churnrate DESC;

PREFERREDPAYMENTMODE                                  TOTALCUSTOMER CUSTOMERCHURN CHURNRATE
-------------------------------------------------- ------------- ------------- ---------
Cash on Delivery                                             514           128      24.9
E wallet                                                     614           140      22.8
CC                                                           273            59     21.61
UPI                                                          414            72     17.39
Debit Card                                                  2314           356     15.38
Credit Card                                                 1501           193     12.86

6 rows selected.
```

In this query, we use the SELECT statement to retrieve the "PreferredPaymentMode," total number of customers for each payment mode, the number of churned customers for each payment mode, and the churn rate for each "PreferredPaymentMode." We GROUP BY the "PreferredPaymentMode" column to aggregate the results for each unique payment mode.

"Cash on Delivery": In the "Cash on Delivery" category, there are 514 customers. Out of these, 128 customers have churned during the specified period. The churn rate for the "Cash on Delivery" category is approximately 24.9%. This means that around 24.9 out of every 100 customers who prefer to pay through Cash on Delivery have churned.

"E wallet": In the "E wallet" category, there are 614 customers. Out of these, 140 customers have churned during the specified period. The churn rate for the "E wallet" category is approximately 22.8%. This means that around 22.8 out of every 100 customers who prefer to pay through E wallets have churned.

"CC" (Credit Card): In the "CC" category (representing Credit Card payments), there are 273 customers. Out of these, 59 customers have churned during the specified period. The churn rate for the "CC" category is approximately 21.61%. This means that around 21.61 out of every 100 customers who prefer to pay with Credit Cards have churned.

"UPI" (Unified Payment Interface): In the "UPI" category, there are 414 customers. Out of these, 72 customers have churned during the specified period. The churn rate for the "UPI" category is approximately 17.39%. This means that around 17.39 out of every 100 customers who prefer to pay through UPI have churned.

"Debit Card": In the "Debit Card" category, there are 2,314 customers. Out of these, 356 customers have churned during the specified period. The churn rate for the "Debit Card" category is approximately 15.38%. This means that around 15.38 out of every 100 customers who prefer to pay with Debit Cards have churned.

"Credit Card": In the "Credit Card" category, there are 1,501 customers. Out of these, 193 customers have churned during the specified period. The churn rate for the "Credit Card" category is approximately 12.86%. This means that around 12.86 out of every 100 customers who prefer to pay with Credit Cards have churned.

Comparison:

The output shows that the highest churn rate is observed for customers who prefer to pay through "Cash on Delivery" (24.9%), followed by "E wallet" (22.8%) and "CC" (Credit Card) (21.61%).

On the other hand, the lowest churn rate is observed for customers who prefer to pay with "Credit Card" (12.86%).

Business Implications:

The analysis suggests that customers who opt for "Cash on Delivery" and "E wallet" are more likely to churn compared to those who prefer other payment modes.

It may be worth investigating the reasons behind the higher churn rates for "Cash on Delivery" and "E wallet" customers. Addressing any issues related to the payment process for these customers may help improve customer retention.

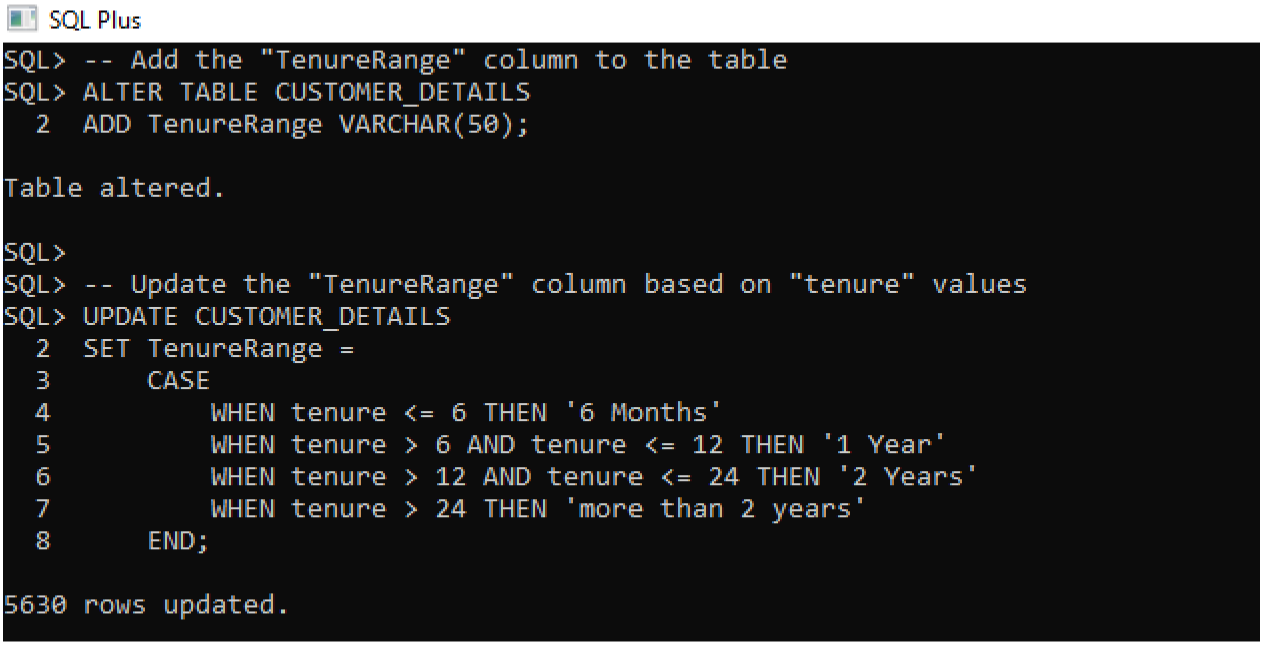
Customers who prefer to pay through "Credit Card" have the lowest churn rate, indicating that this payment mode might be associated with a more loyal customer base. The platform could explore offering additional incentives or promotions for customers using "Credit Card" to further enhance retention.

Offering a seamless and secure payment experience for "UPI" users may also contribute to reducing churn for this payment mode.

Continuous monitoring of churn rates based on payment modes can provide insights into the effectiveness of payment-related initiatives and assist in optimizing the payment options offered on the ecommerce platform.

6. What is the typical tenure for churned customers?

First, we will create a new column called “TenureRange” that groups the customer tenure into 6 months, 1 year, 2 years, and more than 2 years using the CASE statement.



```
SQL> -- Add the "TenureRange" column to the table
SQL> ALTER TABLE CUSTOMER_DETAILS
     ADD TenureRange VARCHAR(50);

Table altered.

SQL>
SQL> -- Update the "TenureRange" column based on "tenure" values
SQL> UPDATE CUSTOMER_DETAILS
     SET TenureRange =
         CASE
             WHEN tenure <= 6 THEN '6 Months'
             WHEN tenure > 6 AND tenure <= 12 THEN '1 Year'
             WHEN tenure > 12 AND tenure <= 24 THEN '2 Years'
             WHEN tenure > 24 THEN 'more than 2 years'
         END;

5630 rows updated.
```

In this query, we first use the ALTER TABLE statement to add a new column named "TenureRange" to the "CUSTOMER_DETAILS" table. The data type for this new column is set to VARCHAR(50).

Next, we use the UPDATE statement to populate the "TenureRange" column based on the values of the "tenure" column. The CASE statement is used to define the conditions for assigning specific range values based on the "tenure" values. The tenure ranges are as follows:

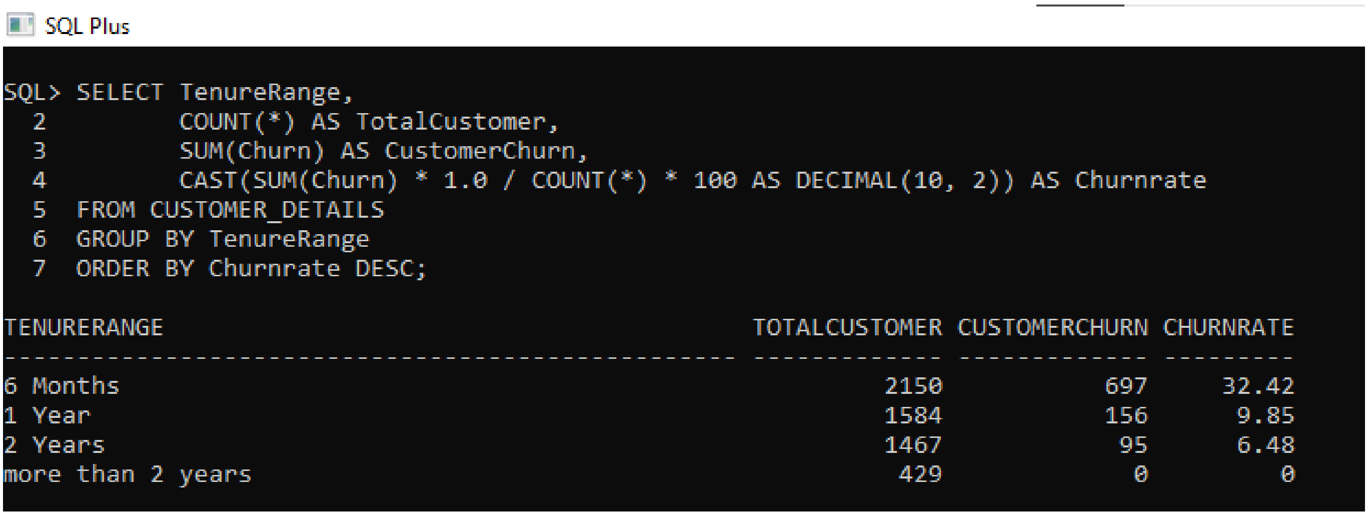
If the "tenure" value is less than or equal to 6, the "TenureRange" is set to '6 Months'.

If the "tenure" value is greater than 6 and less than or equal to 12, the "TenureRange" is set to '1 Year'.

If the "tenure" value is greater than 12 and less than or equal to 24, the "TenureRange" is set to '2 Years'.

If the "tenure" value is greater than 24, the "TenureRange" is set to 'more than 2 years'.

Finding typical tenure for churned customers



```
SQL> SELECT TenureRange,
            COUNT(*) AS TotalCustomer,
            SUM(Churn) AS CustomerChurn,
            CAST(SUM(Churn) * 1.0 / COUNT(*) * 100 AS DECIMAL(10, 2)) AS Churnrate
     FROM CUSTOMER_DETAILS
     GROUP BY TenureRange
     ORDER BY Churnrate DESC;

TENURERANGE                                        TOTALCUSTOMER CUSTOMERCHURN CHURNRATE
-------------------------------------------------- ------------- ------------- ---------
6 Months                                                    2150           697     32.42
1 Year                                                      1584           156      9.85
2 Years                                                     1467            95      6.48
more than 2 years                                            429             0         0
```

In this query, we use the SELECT statement to retrieve the "TenureRange," total number of customers for each tenure range, the number of churned customers for each tenure range, and the churn rate for each

"TenureRange." We GROUP BY the "TenureRange" column to aggregate the results for each unique tenure range.

Interpretation:

"6 Months": In the "6 Months" tenure range, there are 2,150 customers. Out of these, 697 customers have churned during the specified period. The churn rate for the "6 Months" tenure range is approximately 32.42%. This means that around 32.42 out of every 100 customers who have been with the ecommerce platform for 6 months or less have churned.

"1 Year": In the "1 Year" tenure range, there are 1,584 customers. Out of these, 156 customers have churned during the specified period. The churn rate for the "1 Year" tenure range is approximately 9.85%. This means that around 9.85 out of every 100 customers who have been with the ecommerce platform for 1 year have churned.

"2 Years": In the "2 Years" tenure range, there are 1,467 customers. Out of these, 95 customers have churned during the specified period. The churn rate for the "2 Years" tenure range is approximately 6.48%. This means that around 6.48 out of every 100 customers who have been with the ecommerce platform for 2 years have churned.

"More than 2 years": In the "More than 2 years" tenure range, there are 429 customers. Out of these, 0 customers have churned during the specified period. The churn rate for the "More than 2 years" tenure range is 0%. This indicates that there have been no churned customers among those who have been with the ecommerce platform for more than 2 years during the specified period.

Comparison:

The output shows that the churn rate tends to decrease with increasing customer tenure. Customers with shorter tenure (6 Months) have the highest churn rate, while customers with longer tenure (2 Years and more) have lower or no churn.

Business Implications:

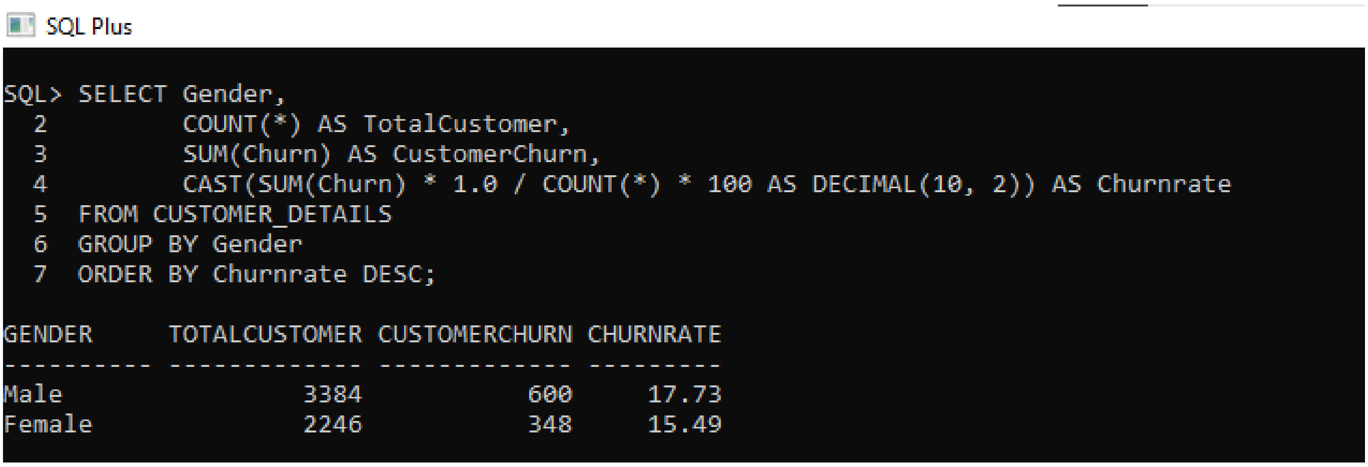
The analysis suggests that new customers who have been with the platform for a shorter period (6 months or less) are more likely to churn. It may be beneficial to focus on retention strategies targeted at this group of customers to reduce churn and increase their loyalty.

Customers who have been with the platform for a longer period (1 year and 2 years) show lower churn rates. However, continuous efforts to maintain customer engagement and satisfaction are still essential to retain these customers in the long term.

It is noteworthy that there have been no churned customers in the "More than 2 years" tenure range during the specified period. This indicates a high level of customer loyalty among those who have been with the platform for an extended period. Acknowledging and rewarding these loyal customers can help strengthen their loyalty further.

Regular monitoring of churn rates by tenure range can provide valuable insights into the effectiveness of customer retention efforts and assist in developing targeted retention strategies for different customer segments based on their tenure.

7. Is there any difference in churn rate between male and female customers?



```
SQL> SELECT Gender,
            COUNT(*) AS TotalCustomer,
            SUM(Churn) AS CustomerChurn,
            CAST(SUM(Churn) * 1.0 / COUNT(*) * 100 AS DECIMAL(10, 2)) AS Churnrate
     FROM CUSTOMER_DETAILS
     GROUP BY Gender
     ORDER BY Churnrate DESC;

GENDER     TOTALCUSTOMER CUSTOMERCHURN CHURNRATE
---------- ------------- ------------- ---------
Male                3384           600     17.73
Female              2246           348     15.49
```

In this query, we use the SELECT statement to retrieve the "Gender," total number of customers for each gender, the number of churned customers for each gender, and the churn rate for each "Gender." We GROUP BY the "Gender" column to aggregate the results for each unique gender.

Interpretation:

"Male": In the "Male" gender group, there are 3,384 customers. Out of these, 600 customers have churned during the specified period. The churn rate for the "Male" gender group is approximately 17.73%. This means that around 17.73 out of every 100 male customers have churned.

"Female": In the "Female" gender group, there are 2,246 customers. Out of these, 348 customers have churned during the specified period. The churn rate for the "Female" gender group is approximately 15.49%. This means that around 15.49 out of every 100 female customers have churned.

Comparison:

The output shows that the churn rate for male customers is slightly higher (17.73%) compared to female customers (15.49%).

Business Implications:

The analysis suggests that male customers have a slightly higher churn rate compared to female customers. Understanding the reasons behind this difference could help in developing targeted retention strategies for each gender.

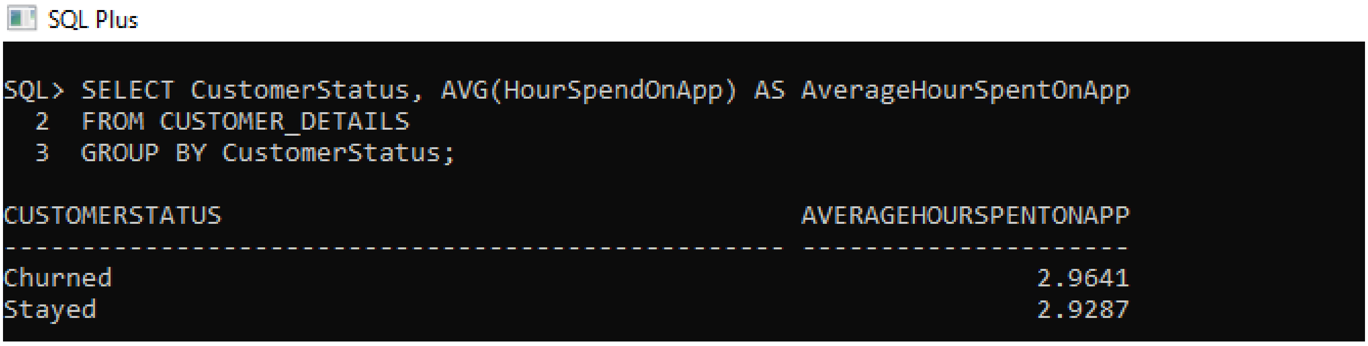
It may be beneficial to conduct further analysis to identify any gender-specific factors that may be influencing churn behavior. Such factors could include preferences, shopping habits, customer service experience, or targeted marketing efforts.

Tailoring marketing and retention strategies to address the needs and preferences of each gender group may help improve customer loyalty and reduce churn.

Customer feedback and satisfaction surveys specific to each gender group can provide valuable insights into the factors influencing churn and help identify areas for improvement.

Continuously monitoring churn rates for different gender groups can help track the effectiveness of retention initiatives and enable data-driven decision-making to enhance customer retention efforts.

8. How does the average time spent on the app differ for churned and non-churned customers?



```
SQL> SELECT CustomerStatus, AVG(HourSpendOnApp) AS AverageHourSpentOnApp
  2  FROM CUSTOMER_DETAILS
  3  GROUP BY CustomerStatus;

CUSTOMERSTATUS                                     AVERAGEHOURSPENTONAPP
-------------------------------------------------- ---------------------
Churned                                                           2.9641
Stayed                                                            2.9287
```

In this query, we use the SELECT statement to retrieve the "CustomerStatus" and the average hours spent on the app ("HourSpendOnApp") for each "CustomerStatus." We use the AVG function to calculate the average value of the "HourSpendOnApp" column for each group of "CustomerStatus."

The GROUP BY clause is used to group the results based on the "CustomerStatus" column. This groups the data into two groups, "Churned" and "Stayed" customers, based on their "CustomerStatus."

Interpretation:

"Churned" Customers: The average hours spent on the app for customers who have churned ("Churned" customer status) is approximately 2.9641 hours.

"Stayed" Customers: The average hours spent on the app for customers who have stayed ("Stayed" customer status) is approximately 2.9287 hours.

Comparison:

The analysis shows that there is a slight difference in the average hours spent on the app between "Churned" and "Stayed" customers. "Churned" customers have a slightly higher average app usage (2.9641 hours) compared to "Stayed" customers (2.9287 hours).

Business Implications:

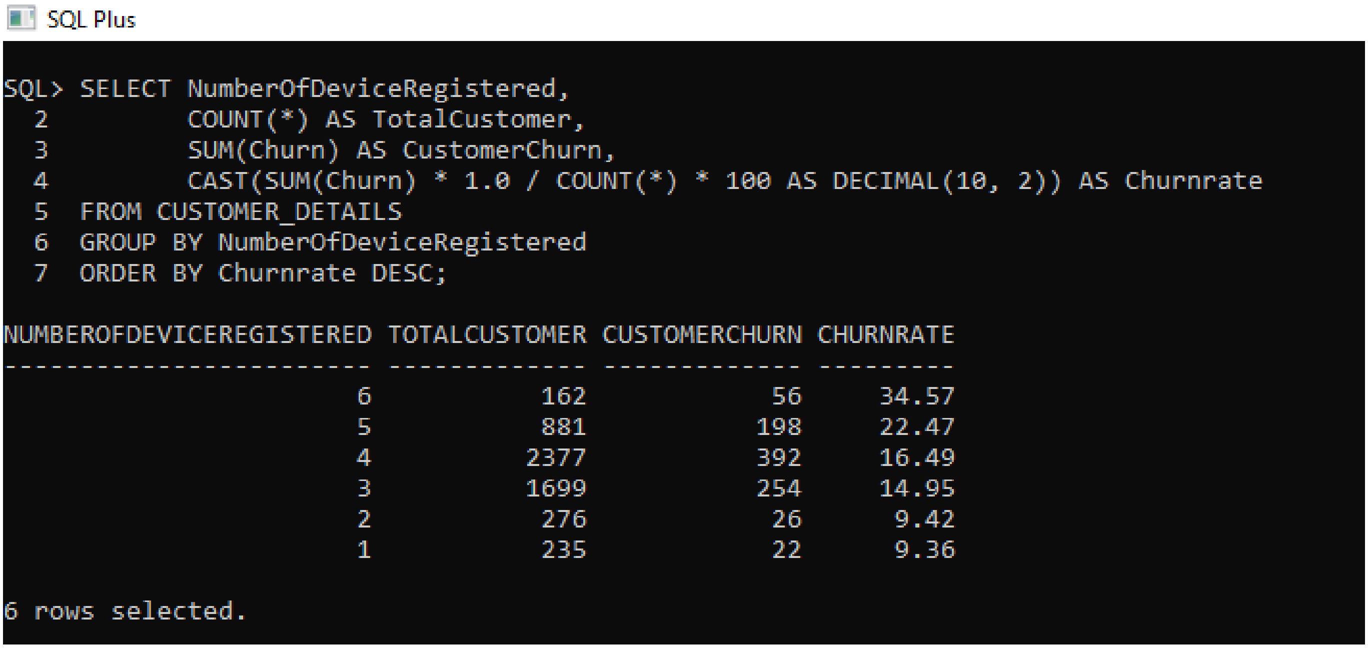
The analysis suggests that, on average, customers who have churned spent slightly more time on the app compared to those who have stayed. This may indicate that higher app engagement is not necessarily linked to customer retention.

It is essential to further investigate the reasons behind the difference in app usage between "Churned" and "Stayed" customers. Analyzing customer behavior patterns, feedback, and satisfaction levels can provide insights into potential factors influencing churn.

Improving app engagement and customer experience can be crucial in reducing churn. Identifying areas for app improvement, addressing pain points, and offering personalized experiences may help increase customer loyalty.

Continuous monitoring of app usage and its correlation with customer churn can help in measuring the effectiveness of app-related initiatives and optimizing app features to improve customer retention.

9. Does the number of registered devices impact the likelihood of churn?



```
SQL> SELECT NumberOfDeviceRegistered,
            COUNT(*) AS TotalCustomer,
            SUM(Churn) AS CustomerChurn,
            CAST(SUM(Churn) * 1.0 / COUNT(*) * 100 AS DECIMAL(10, 2)) AS Churnrate
     FROM CUSTOMER_DETAILS
     GROUP BY NumberOfDeviceRegistered
     ORDER BY Churnrate DESC;

NUMBEROFDEVICEREGISTERED TOTALCUSTOMER CUSTOMERCHURN CHURNRATE
------------------------ ------------- ------------- ---------
                       6           162            56     34.57
                       5           881           198     22.47
                       4          2377           392     16.49
                       3          1699           254     14.95
                       2           276            26      9.42
                       1           235            22      9.36

6 rows selected.
```

In this query, we use the SELECT statement to retrieve the "NumberOfDeviceRegistered," total number of customers for each "NumberOfDeviceRegistered," the number of churned customers for each "NumberOfDeviceRegistered," and the churn rate for each "NumberOfDeviceRegistered." We GROUP BY the "NumberOfDeviceRegistered" column to aggregate the results for each unique number of devices registered.

Interpretation:

"6 Devices Registered": In the group of customers who have registered 6 devices, there are 162 customers. Out of these, 56 customers have churned. The churn rate for this group is approximately 34.57%. This means that around 34.57 out of every 100 customers who have registered 6 devices have churned.

"5 Devices Registered": In the group of customers who have registered 5 devices, there are 881 customers. Out of these, 198 customers have churned. The churn rate for this group is approximately 22.47%. This means that around 22.47 out of every 100 customers who have registered 5 devices have churned.

"4 Devices Registered": In the group of customers who have registered 4 devices, there are 2,377 customers. Out of these, 392 customers have churned. The churn rate for this group is approximately 16.49%. This means that around 16.49 out of every 100 customers who have registered 4 devices have churned.

"3 Devices Registered": In the group of customers who have registered 3 devices, there are 1,699 customers. Out of these, 254 customers have churned. The churn rate for this group is approximately 14.95%. This means that around 14.95 out of every 100 customers who have registered 3 devices have churned.

"2 Devices Registered": In the group of customers who have registered 2 devices, there are 276 customers. Out of these, 26 customers have churned. The churn rate for this group is approximately 9.42%. This means that around 9.42 out of every 100 customers who have registered 2 devices have churned.

"1 Device Registered": In the group of customers who have registered 1 device, there are 235 customers. Out of these, 22 customers have churned. The churn rate for this group is approximately 9.36%. This means that around 9.36 out of every 100 customers who have registered 1 device have churned.

Comparison:

The analysis shows that the churn rate tends to increase with an increasing number of devices registered. Customers who have registered 6 devices have the highest churn rate (34.57%), while those with 1 or 2 devices registered have relatively lower churn rates (9.36% and 9.42%, respectively).

Business Implications:

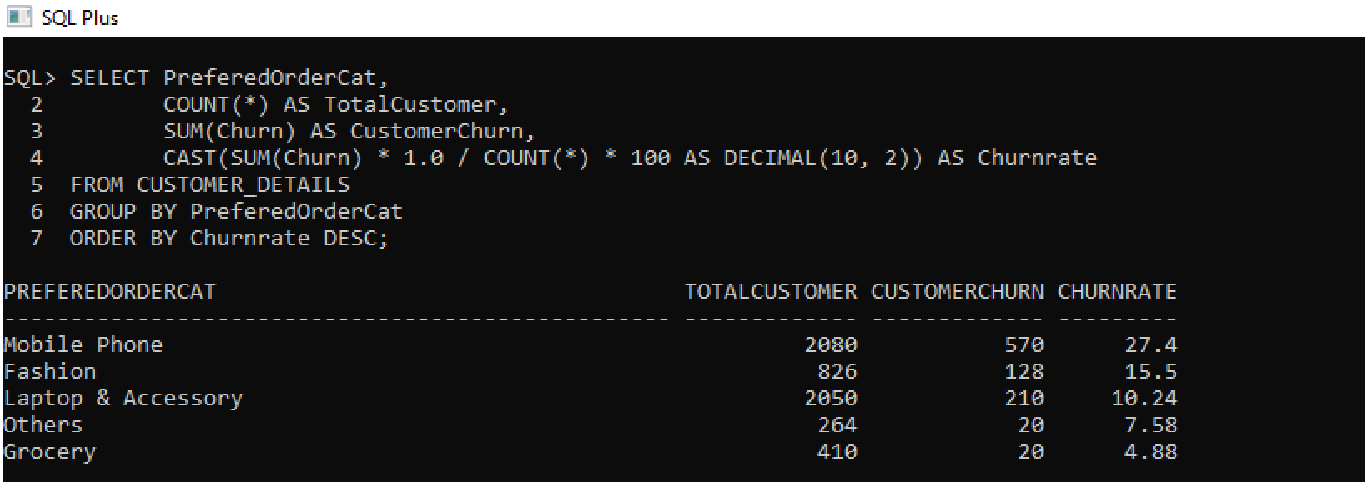
The results indicate that customers with more devices registered have a higher churn rate. This might suggest that customers with multiple devices may have different expectations or usage patterns, leading to varied churn behaviors.

Understanding the reasons behind the higher churn rate for customers with more registered devices can provide insights into areas that need improvement or personalized retention strategies for these customers.

Offering personalized experiences and promotions to customers with multiple devices might help increase customer loyalty and retention in this segment.

Further analysis, customer surveys, and feedback collection can help identify specific pain points or issues that customers with more registered devices might be facing, and address them to improve their overall experience.

10. Which order category is most preferred among churned customers?



```
SQL> SELECT PreferedOrderCat,
            COUNT(*) AS TotalCustomer,
            SUM(Churn) AS CustomerChurn,
            CAST(SUM(Churn) * 1.0 / COUNT(*) * 100 AS DECIMAL(10, 2)) AS Churnrate
     FROM CUSTOMER_DETAILS
     GROUP BY PreferedOrderCat
     ORDER BY Churnrate DESC;

PREFEREDORDERCAT                                   TOTALCUSTOMER CUSTOMERCHURN CHURNRATE
-------------------------------------------------- ------------- ------------- ---------
Mobile Phone                                                2080           570      27.4
Fashion                                                      826           128      15.5
Laptop & Accessory                                          2050           210     10.24
Others                                                       264            20      7.58
Grocery                                                      410            20      4.88
```

In this query, we use the SELECT statement to retrieve the "PreferedOrderCat," total number of customers for each "PreferedOrderCat," the number of churned customers for each "PreferedOrderCat," and the churn rate for each "PreferedOrderCat." We GROUP BY the "PreferedOrderCat" column to aggregate the results for each unique preferred order category.

Interpretation:

"Mobile Phone" Preferred Order Category: In the "Mobile Phone" preferred order category, there are 2,080 customers. Out of these, 570 customers have churned. The churn rate for this category is approximately 27.4%. This means that around 27.4 out of every 100 customers who prefer to order "Mobile Phone" items have churned.

"Fashion" Preferred Order Category: In the "Fashion" preferred order category, there are 826 customers. Out of these, 128 customers have churned. The churn rate for this category is approximately 15.5%. This means that around 15.5 out of every 100 customers who prefer to order "Fashion" items have churned.

"Laptop & Accessory" Preferred Order Category: In the "Laptop & Accessory" preferred order category, there are 2,050 customers. Out of these, 210 customers have churned. The churn rate for this category is approximately 10.24%. This means that around 10.24 out of every 100 customers who prefer to order "Laptop & Accessory" items have churned.

"Others" Preferred Order Category: In the "Others" preferred order category, there are 264 customers. Out of these, 20 customers have churned. The churn rate for this category is approximately 7.58%. This means that around 7.58 out of every 100 customers who prefer to order items from the "Others" category have churned.

"Grocery" Preferred Order Category: In the "Grocery" preferred order category, there are 410 customers. Out of these, 20 customers have churned. The churn rate for this category is approximately 4.88%. This means that around 4.88 out of every 100 customers who prefer to order "Grocery" items have churned.

Comparison:

The analysis shows that the "Mobile Phone" preferred order category has the highest churn rate (27.4%) among all the categories, followed by "Fashion" (15.5%), "Laptop & Accessory" (10.24%), "Others" (7.58%), and "Grocery" (4.88%).

Business Implications:

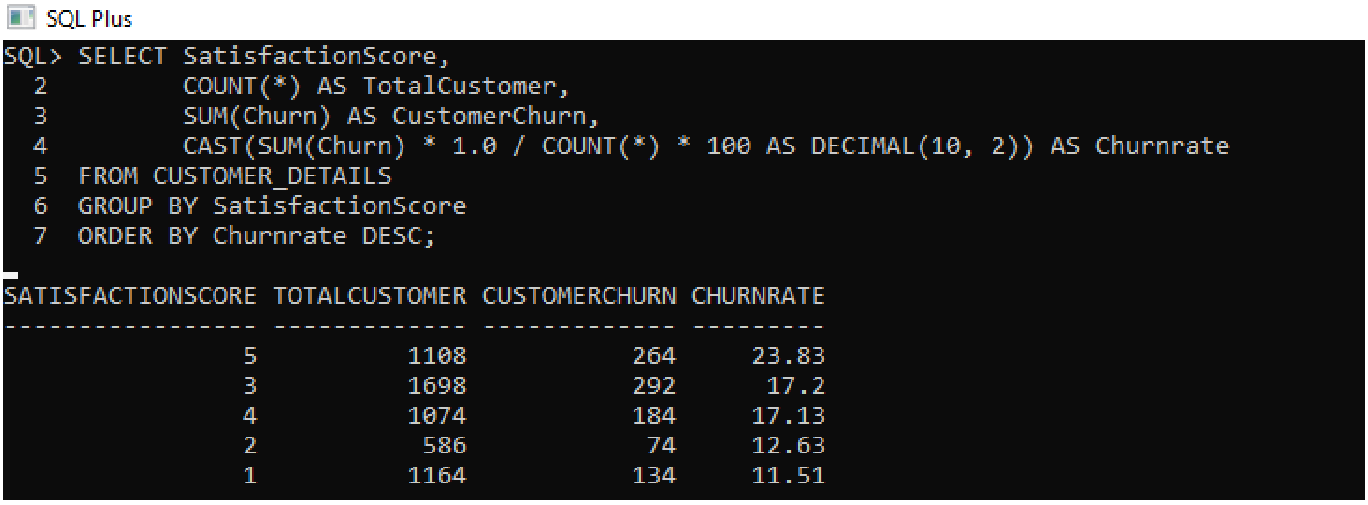
The results suggest that customers who prefer to order "Mobile Phone" items have the highest churn rate. This may indicate that customers who order mobile phones are more likely to churn, and there might be specific reasons or challenges associated with this category that need attention.

The lower churn rates for "Grocery" and "Others" categories may suggest that customers who order these items may have higher retention levels, indicating a stable customer base.

Understanding the factors contributing to higher churn in certain preferred order categories can help in developing targeted retention strategies, improving the customer experience, and addressing potential pain points.

Further analysis, such as customer surveys and feedback collection, can provide insights into specific reasons for churn in each category, helping in optimizing marketing efforts and customer retention initiatives.

11. Is there any relationship between customer satisfaction scores and churn?



```
SQL> SELECT SatisfactionScore,
            COUNT(*) AS TotalCustomer,
            SUM(Churn) AS CustomerChurn,
            CAST(SUM(Churn) * 1.0 / COUNT(*) * 100 AS DECIMAL(10, 2)) AS Churnrate
     FROM CUSTOMER_DETAILS
     GROUP BY SatisfactionScore
     ORDER BY Churnrate DESC;

SATISFACTIONSCORE TOTALCUSTOMER CUSTOMERCHURN CHURNRATE
----------------- ------------- ------------- ---------
                5          1108           264     23.83
                3          1698           292      17.2
                4          1074           184     17.13
                2           586            74     12.63
                1          1164           134     11.51
```

In this query, we use the SELECT statement to retrieve the "SatisfactionScore," total number of customers for each "SatisfactionScore," the number of churned customers for each "SatisfactionScore," and the churn rate for each "SatisfactionScore." We GROUP BY the "SatisfactionScore" column to aggregate the results for each unique satisfaction score.

Interpretation:

Satisfaction Score 5: In the "Satisfaction Score 5" group, there are 1,108 customers. Out of these, 264 customers have churned. The churn rate for this group is approximately 23.83%. This means that around 23.83 out of every 100 customers who have a satisfaction score of 5 have churned.

Satisfaction Score 3: In the "Satisfaction Score 3" group, there are 1,698 customers. Out of these, 292 customers have churned. The churn rate for this group is approximately 17.2%. This means that around 17.2 out of every 100 customers who have a satisfaction score of 3 have churned.

Satisfaction Score 4: In the "Satisfaction Score 4" group, there are 1,074 customers. Out of these, 184 customers have churned. The churn rate for this group is approximately 17.13%. This means that around 17.13 out of every 100 customers who have a satisfaction score of 4 have churned.

Satisfaction Score 2: In the "Satisfaction Score 2" group, there are 586 customers. Out of these, 74 customers have churned. The churn rate for this group is approximately 12.63%. This means that around 12.63 out of every 100 customers who have a satisfaction score of 2 have churned.

Satisfaction Score 1: In the "Satisfaction Score 1" group, there are 1,164 customers. Out of these, 134 customers have churned. The churn rate for this group is approximately 11.51%. This means that around 11.51 out of every 100 customers who have a satisfaction score of 1 have churned.

Comparison:

The analysis shows that customers with a satisfaction score of 5 have the highest churn rate (23.83%), followed by those with a score of 3 (17.2%) and 4 (17.13%).

Customers with a satisfaction score of 1 have the lowest churn rate (11.51%), and those with a score of 2 (12.63%) have a slightly higher churn rate compared to the lowest.

Business Implications:

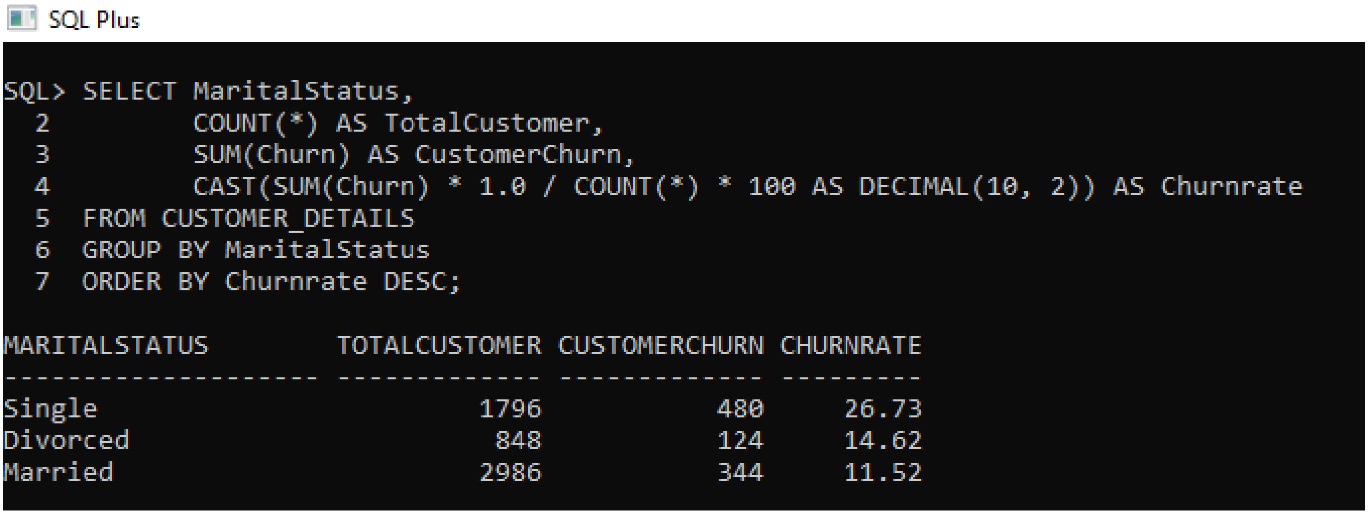
The results suggest that customers with the highest satisfaction score (5) have a higher churn rate. This may seem counterintuitive, but it could be due to various reasons, such as high expectations and a lower tolerance for any issues or changes in services/products.

Understanding the reasons behind the higher churn rate for customers with the highest satisfaction score can help in identifying specific pain points and addressing them to improve customer loyalty and retention.

The analysis also indicates that customers with lower satisfaction scores (1 and 2) have a relatively lower churn rate. Focusing on improving the experience and resolving issues for these customers may help increase customer retention and satisfaction.

Further analysis, customer feedback surveys, and in-depth exploration of customer behaviors and preferences can provide more insights into the factors influencing satisfaction scores and churn rates, leading to better retention strategies and improved overall customer experience.

12. Does the marital status of customers influence churn behavior?



```
SQL> SELECT MaritalStatus,
            COUNT(*) AS TotalCustomer,
            SUM(Churn) AS CustomerChurn,
            CAST(SUM(Churn) * 1.0 / COUNT(*) * 100 AS DECIMAL(10, 2)) AS Churnrate
     FROM CUSTOMER_DETAILS
     GROUP BY MaritalStatus
     ORDER BY Churnrate DESC;

MARITALSTATUS        TOTALCUSTOMER CUSTOMERCHURN CHURNRATE
-------------------- ------------- ------------- ---------
Single                        1796           480     26.73
Divorced                       848           124     14.62
Married                       2986           344     11.52
```

In this query, we use the SELECT statement to retrieve the "MaritalStatus," total number of customers for each "MaritalStatus," the number of churned customers for each "MaritalStatus," and the churn rate for each "MaritalStatus." We GROUP BY the "MaritalStatus" column to aggregate the results for each unique marital status.

Interpretation:

Marital Status "Single": In the "Single" marital status group, there are 1,796 customers. Out of these, 480 customers have churned. The churn rate for this group is approximately 26.73%. This means that around 26.73 out of every 100 customers who are single have churned.

Marital Status "Divorced": In the "Divorced" marital status group, there are 848 customers. Out of these, 124 customers have churned. The churn rate for this group is approximately 14.62%. This means that around 14.62 out of every 100 customers who are divorced have churned.

Marital Status "Married": In the "Married" marital status group, there are 2,986 customers. Out of these, 344 customers have churned. The churn rate for this group is approximately 11.52%. This means that around 11.52 out of every 100 customers who are married have churned.

Comparison:

The analysis shows that customers who are "Single" have the highest churn rate (26.73%) among all the marital status groups, followed by "Divorced" (14.62%) and "Married" (11.52%).

Customers who are "Married" have the lowest churn rate among the different marital status groups.

Business Implications:

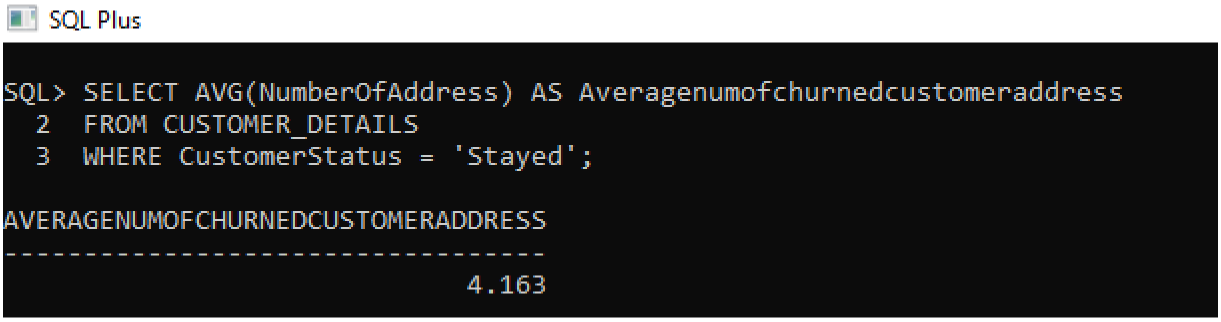
The results suggest that customers who are single have a higher churn rate compared to customers who are married or divorced. This indicates that single customers may have different behaviors or needs that contribute to a higher churn rate.

Understanding the reasons behind the higher churn rate for single customers can help in developing targeted retention strategies and providing personalized experiences to improve customer loyalty.

The lower churn rate for married customers indicates that they may be more loyal or satisfied with the services, products, or experiences offered by the ecommerce platform.

Analyzing customer feedback and conducting surveys among different marital status groups can provide more insights into the factors influencing churn behavior and satisfaction levels, helping in optimizing retention efforts and enhancing overall customer experience.

13. How many addresses do non-churned customers have on average?

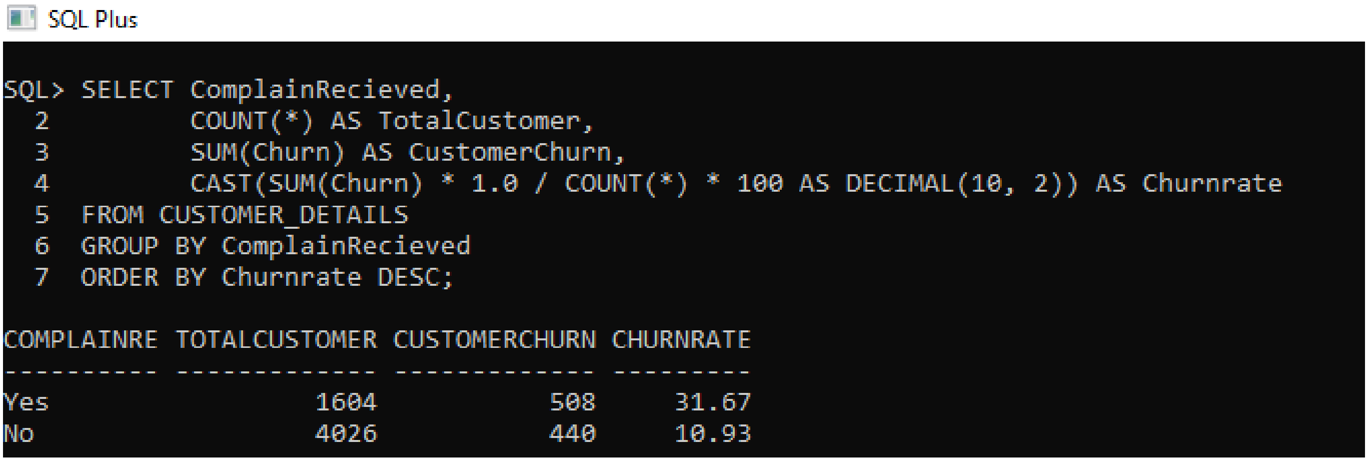


```
SQL> SELECT AVG(NumberOfAddress) AS Averagenumofchurnedcustomeraddress
  2  FROM CUSTOMER_DETAILS
  3  WHERE CustomerStatus = 'Stayed';

AVERAGENUMOFCHURNEDCUSTOMERADDRESS
----------------------------------
                             4.163
```

In this query, we use the SELECT statement to calculate the average number of addresses for customers who have "stayed" (CustomerStatus = 'Stayed'). The AVG function calculates the average of the "NumberOfAddress" column for the specified group of customers.

The average number of addresses for customers who have "stayed" (not churned) on the ecommerce platform is approximately 4.163. This means that, on average, customers who continue to use the platform have about 4.163 addresses associated with their accounts. Understanding such customer behavior can help in providing more personalized services and improving customer experience, potentially leading to increased customer loyalty and retention.

14. Do customer complaints influence churned behavior?



```
SQL> SELECT ComplainRecieved,
            COUNT(*) AS TotalCustomer,
            SUM(Churn) AS CustomerChurn,
            CAST(SUM(Churn) * 1.0 / COUNT(*) * 100 AS DECIMAL(10, 2)) AS Churnrate
     FROM CUSTOMER_DETAILS
     GROUP BY ComplainRecieved
     ORDER BY Churnrate DESC;

COMPLAINRE TOTALCUSTOMER CUSTOMERCHURN CHURNRATE
---------- ------------- ------------- ---------
Yes                 1604           508     31.67
No                  4026           440     10.93
```

In this query, we use the SELECT statement to retrieve the "ComplainRecieved," total number of customers for each "ComplainRecieved," the number of churned customers for each "ComplainRecieved," and the churn rate for each "ComplainRecieved." We GROUP BY the "ComplainRecieved" column to aggregate the results for customers who have received a complaint (ComplainRecieved = 'Yes') and those who have not received a complaint (ComplainRecieved = 'No').

Interpretation:

Complain Received "Yes": In the "Complain Received" group, there are 1,604 customers who have received a complaint. Out of these, 508 customers have churned. The churn rate for this group is approximately 31.67%. This means that around 31.67 out of every 100 customers who have received a complaint have churned.

Complain Received "No": In the "Complain Not Received" group, there are 4,026 customers who have not received any complaints. Out of these, 440 customers have churned. The churn rate for this group is approximately 10.93%. This means that around 10.93 out of every 100 customers who have not received a complaint have churned.

Comparison:

The analysis shows that customers who have received a complaint (Complain Received = 'Yes') have a significantly higher churn rate (31.67%) compared to customers who have not received any complaints (Complain Received = 'No') with a lower churn rate (10.93%).

Business Implications:

The results indicate that customer complaints have a substantial impact on churn behavior. Customers who have received complaints are more likely to churn, suggesting that addressing and resolving customer complaints effectively is crucial for improving customer retention and loyalty.

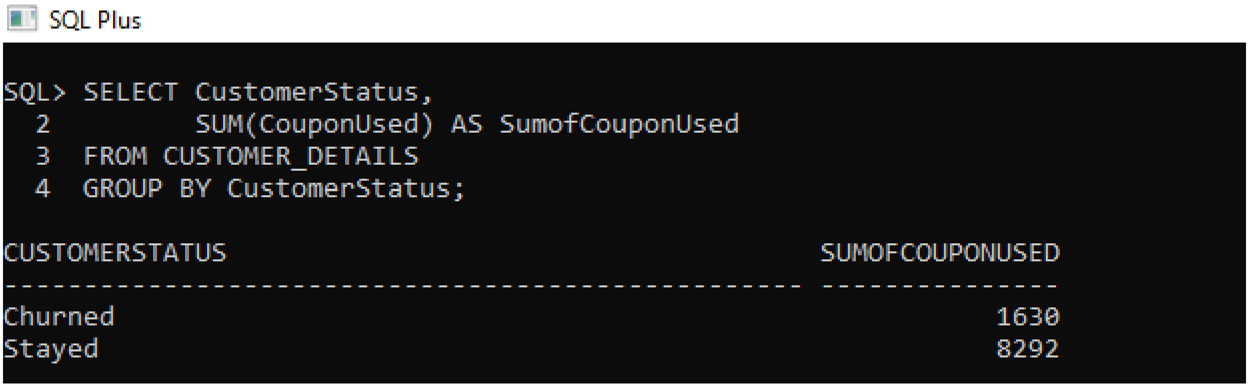
Understanding the reasons behind customer complaints and taking proactive measures to address and resolve them promptly can lead to a decrease in churn rates and an increase in overall customer satisfaction.

Monitoring customer feedback and implementing strategies to reduce the number of complaints can contribute to a more positive customer experience and long-term customer relationships.

Customer support teams and processes should be optimized to handle complaints efficiently, showing customers that their concerns are taken seriously and addressed with care.

Further analysis, including root cause analysis of complaints and follow-up with customers after complaint resolution, can provide more insights into specific pain points and opportunities for improvement in the ecommerce platform's products and services.

15. How does the use of coupons differ between churned and non-churned customers?



```
SQL> SELECT CustomerStatus,
  2         SUM(CouponUsed) AS SumofCouponUsed
  3  FROM CUSTOMER_DETAILS
  4  GROUP BY CustomerStatus;

CUSTOMERSTATUS                                    SUMOFCOUPONUSED
------------------------------------------------- ---------------
Churned                                                      1630
Stayed                                                       8292
```

In this query, we use the SELECT statement to retrieve the "CustomerStatus" and the total sum of coupons used (CouponUsed) for each customer status group. We use the SUM function to calculate the sum of coupon used for each "CustomerStatus" group.

The GROUP BY clause is used to group the results by the "CustomerStatus" column, allowing us to calculate the sum of coupon used for each unique customer status group.

Interpretation:

Customer Status "Churned": Customers who have churned (CustomerStatus = 'Churned') have a total sum of 1630 coupons used. This means that customers who churned collectively used 1630 coupons.

Customer Status "Stayed": Customers who have stayed (CustomerStatus = 'Stayed') have a total sum of 8292 coupons used. This means that customers who remained with the ecommerce platform collectively used 8292 coupons.

Comparison:

The analysis shows that customers who have stayed (not churned) have used a significantly higher number of coupons (8292) compared to customers who churned (1630).

Business Implications:

The results suggest that customers who stay loyal to the ecommerce platform tend to use more coupons, which could be an indicator of customer engagement and interest in promotional offers.

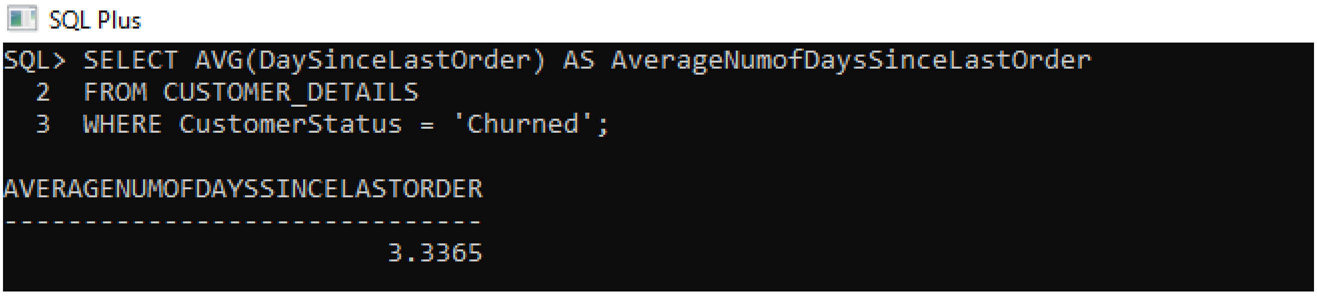
Offering attractive and personalized coupon offers to existing customers can help in increasing customer retention and encouraging repeat purchases.

Analyzing the coupon usage patterns of churned customers can provide insights into the effectiveness of past coupon strategies and may help in refining future coupon campaigns to target potential churned customers.

To improve customer retention and loyalty, consider tailoring coupon offers based on customer segments, preferences, and behaviors. For example, providing exclusive or targeted coupons to specific customer groups can drive higher engagement and encourage repeat purchases.

Continuously monitoring and analyzing coupon usage data, along with customer feedback, can help in optimizing coupon strategies to strike a balance between customer acquisition, retention, and overall business profitability.

16. What is the average number of days since the last order for churned customers?



```
SQL> SELECT AVG(DaySinceLastOrder) AS AverageNumofDaysSinceLastOrder
  2  FROM CUSTOMER_DETAILS
  3  WHERE CustomerStatus = 'Churned';

AVERAGENUMOFDAYSSINCELASTORDER
------------------------------
                        3.3365
```

In this query, we use the SELECT statement to calculate the average number of days since the last order for customers who have churned (CustomerStatus = 'Churned'). The AVG function calculates the average of the "DaysSinceLastOrder" column for the specified group of churned customers.

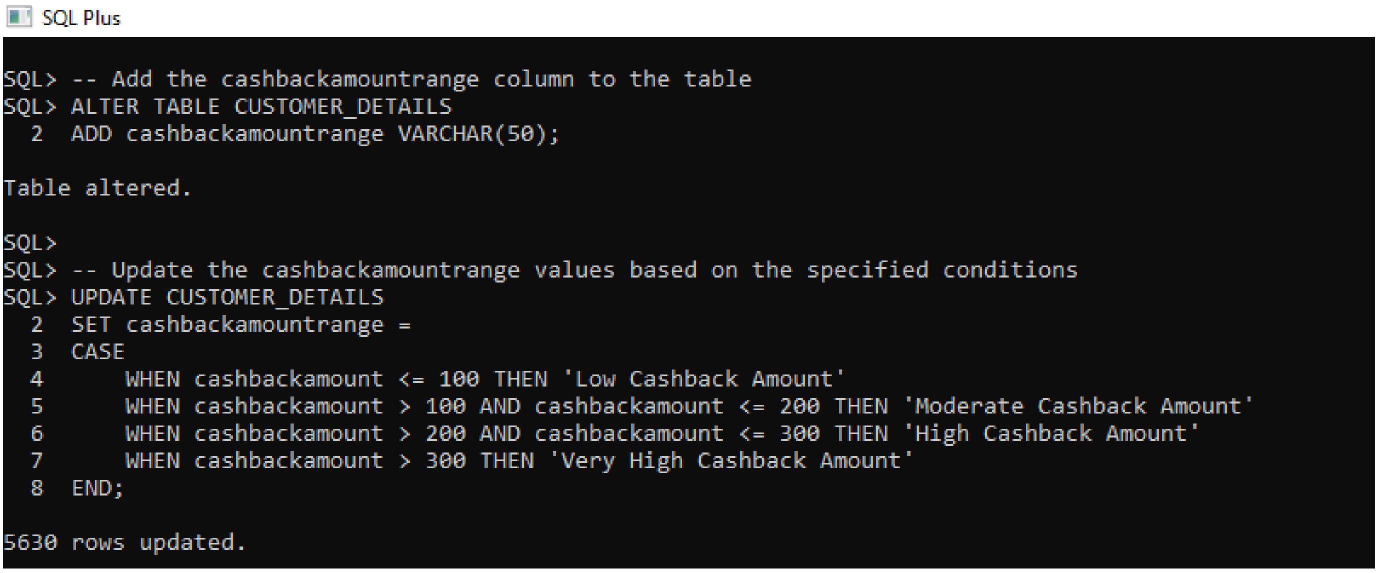
The WHERE clause filters the data to include only churned customers, as indicated by the condition "CustomerStatus = 'Churned'".

The average number of days since the last order for churned customers is approximately 3.3365. This means that, on average, customers who have churned on your ecommerce platform had their last order placed about 3.3365 days before they churned.

Understanding the average time gap between a customer's last order and their decision to churn can provide valuable insights for customer retention strategies. It may help identify factors contributing to churn, such as potential issues with products or services, customer satisfaction, or changes in user behavior. Analyzing this data can guide efforts to improve customer experience, engagement, and potentially reduce churn rates by addressing any identified pain points or concerns proactively.

17. Is there any correlation between cashback amount and churn rate?

First, we will create a new column called “CashbackAmountRange” that groups the cashbackamount into low (less than 100), moderate (between 100 and 200), high(between 200 and 300), and very high (more than 300) using the CASE statement.

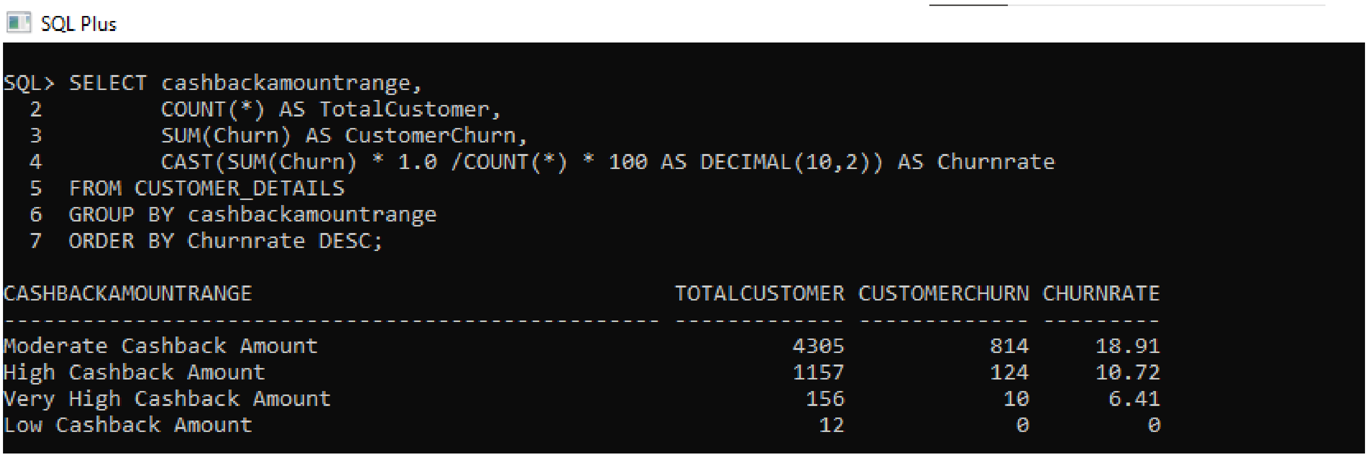


```
SQL> -- Add the cashbackamountrange column to the table
SQL> ALTER TABLE CUSTOMER_DETAILS
  2  ADD cashbackamountrange VARCHAR(50);

Table altered.

SQL>
SQL> -- Update the cashbackamountrange values based on the specified conditions
SQL> UPDATE CUSTOMER_DETAILS
  2  SET cashbackamountrange =
  3  CASE
  4      WHEN cashbackamount <= 100 THEN 'Low Cashback Amount'
  5      WHEN cashbackamount > 100 AND cashbackamount <= 200 THEN 'Moderate Cashback Amount'
  6      WHEN cashbackamount > 200 AND cashbackamount <= 300 THEN 'High Cashback Amount'
  7      WHEN cashbackamount > 300 THEN 'Very High Cashback Amount'
  8  END;

5630 rows updated.
```

In this query, we first use the ALTER TABLE statement to add a new column called "cashbackamountrange" with the data type VARCHAR(50) to the "CUSTOMER_DETAILS" table.

Next, we use the UPDATE statement to set the values of the "cashbackamountrange" column based on the specified conditions. The CASE statement is used to define the conditions, and the corresponding values are assigned to the "cashbackamountrange" column for each row in the table.

Finding the correlation between cashback amount range and churned rate



```
SQL> SELECT cashbackamountrange,
  2         COUNT(*) AS TotalCustomer,
  3         SUM(Churn) AS CustomerChurn,
  4         CAST(SUM(Churn) * 1.0 /COUNT(*) * 100 AS DECIMAL(10,2)) AS Churnrate
  5  FROM CUSTOMER_DETAILS
  6  GROUP BY cashbackamountrange
  7  ORDER BY Churnrate DESC;

CASHBACKAMOUNTRANGE                                TOTALCUSTOMER CUSTOMERCHURN CHURNRATE
-------------------------------------------------- ------------- ------------- ---------
Moderate Cashback Amount                                    4305           814     18.91
High Cashback Amount                                        1157           124     10.72
Very High Cashback Amount                                    156            10      6.41
Low Cashback Amount                                           12             0         0
```

In this query, we use the SELECT statement to retrieve the "cashbackamountrange" and calculate the total number of customers, the total number of churned customers, and the churn rate for each cashback amount range. The SUM function calculates the sum of the "Churn" column for each group, and the COUNT function counts the total number of customers in each group based on the "cashbackamountrange."

The GROUP BY clause is used to group the results by the "cashbackamountrange" column, allowing us to calculate the churn rate for each unique cashback amount range.

The ORDER BY clause sorts the results in descending order based on the churn rate (Churnrate) to display the cashback amount ranges with the highest churn rates at the top.

Interpretation:

Cashback Amount Range "Moderate Cashback Amount": Customers who received moderate cashback amounts fall into this range. There are a total of 4305 customers in this range. Out of these customers, 814 have churned, resulting in a churn rate of approximately 18.91%.

Cashback Amount Range "High Cashback Amount": Customers who received high cashback amounts fall into this range. There are a total of 1157 customers in this range. Out of these customers, 124 have churned, resulting in a churn rate of approximately 10.72%.

Cashback Amount Range "Very High Cashback Amount": Customers who received very high cashback amounts fall into this range. There are a total of 156 customers in this range. Out of these customers, 10 have churned, resulting in a churn rate of approximately 6.41%.

Cashback Amount Range "Low Cashback Amount": Customers who received low cashback amounts fall into this range. There are a total of 12 customers in this range. Interestingly, none of the customers in this range have churned, resulting in a churn rate of 0%.

Comparison:

The analysis reveals that the churn rate decreases as the cashback amount increases. Customers who received higher cashback amounts, such as "Very High Cashback Amount" and "High Cashback Amount," tend to have lower churn rates compared to customers who received moderate or low cashback amounts.

Business Implications:

Offering higher cashback amounts may positively impact customer retention by reducing churn rates. Customers tend to be more engaged and loyal when they receive attractive cashback incentives.

Analyzing the cashback amount range data in combination with other customer attributes and behaviors can help identify optimal cashback strategies for different customer segments.

Careful evaluation of the cost-benefit of cashback programs is crucial to ensure that higher cashback offers are sustainable for the business and align with the overall marketing and sales goals.

Consider conducting A/B testing or experiments to gauge the effectiveness of various cashback amounts in terms of customer retention and overall business revenue.

Customer feedback and surveys can be valuable for understanding the perceived value of cashback programs and preferences of customers concerning cashback incentives.