

CAPSTONE PROJECT

IN

MySQL

E-Commerce Customer Churn Analysis 2024

Presented by
Keerthana .N

Project Title: E-Commerce Customer Churn Analysis

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.

Problem solving

Data cleaning

Data transformation

Data exploration and analysis

Problem solved sql data



sql.E-commerce Customer churn db.sql



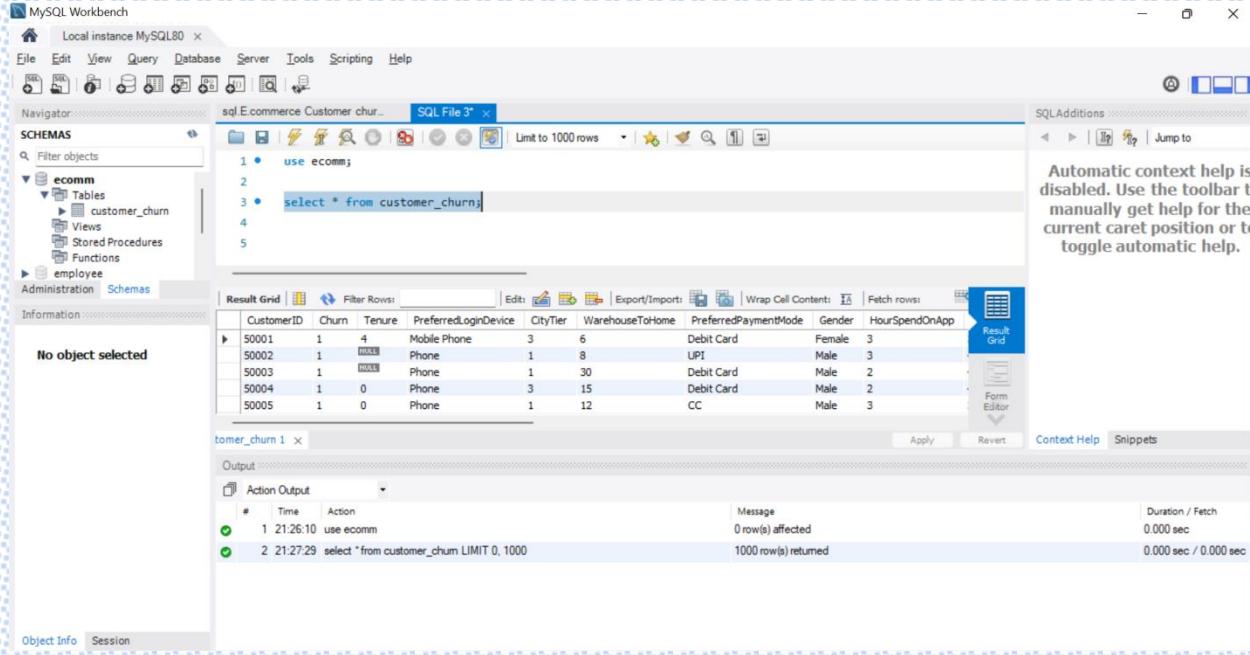
sql project.sql

Project Steps and Objectives:

- Handling Missing Values and Outliers:
- Dealing with Inconsistencies:
- Column Renaming:
- Creating New Columns:
- Column Dropping:

Handling Missing Values and Outliers:

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



The screenshot shows the MySQL Workbench interface with a query editor window titled "SQL File 3". The code is:

```

1 • use ecomm;
2
3 • select * from customer_churn;
4
5

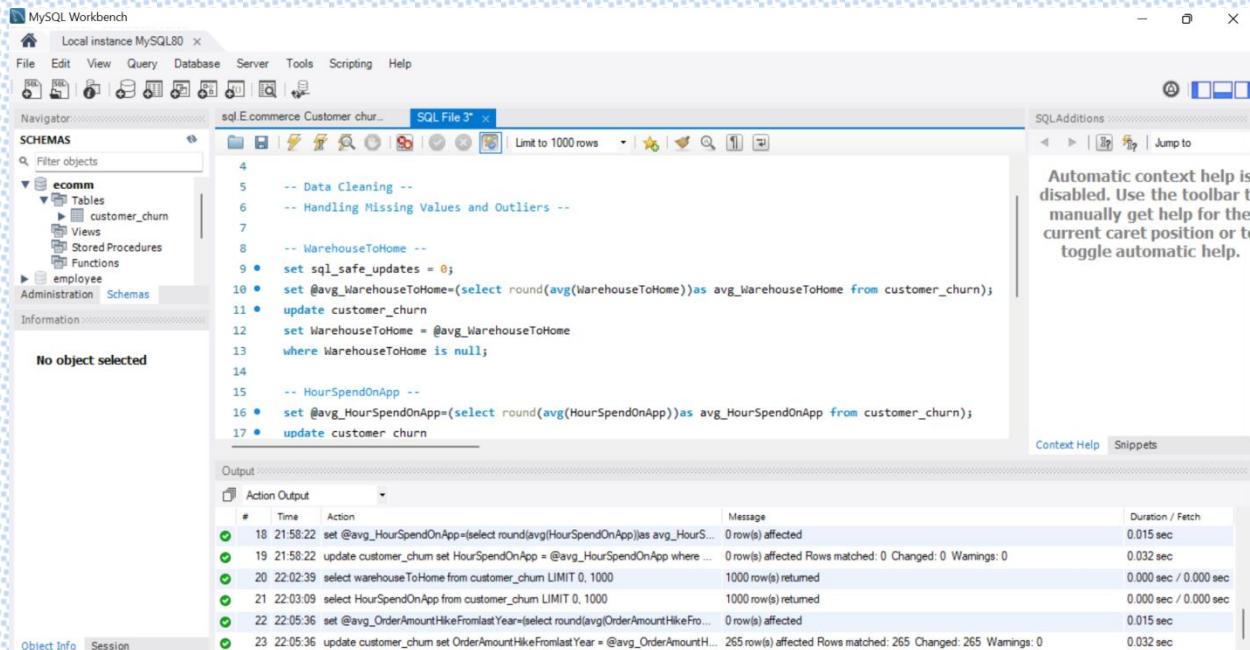
```

The results grid displays the following data:

| CustomerID | Churn | Tenure | PreferredLoginDevice | CityTier | WarehouseToHome | PreferredPaymentMode | Gender | HourSpendOnApp |
|------------|-------|-------------|----------------------|----------|-----------------|----------------------|--------|----------------|
| 50001 | 1 | 4 | Mobile Phone | 3 | 6 | Debit Card | Female | 3 |
| 50002 | 1 | NULL | Phone | 1 | 8 | UPI | Male | 3 |
| 50003 | 1 | NULL | Phone | 1 | 30 | Debit Card | Male | 2 |
| 50004 | 1 | 0 | Phone | 3 | 15 | Debit Card | Male | 2 |
| 50005 | 1 | 0 | Phone | 1 | 12 | CC | Male | 3 |

The output pane shows the execution log:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|--|----------------------|-----------------------|
| 1 | 21:26:10 | use ecomm | 0 row(s) affected | 0.000 sec |
| 2 | 21:27:29 | select * from customer_churn LIMIT 0, 1000 | 1000 row(s) returned | 0.000 sec / 0.000 sec |



The screenshot shows the MySQL Workbench interface with a query editor window titled "SQL File 3". The code is:

```

4
5 -- Data Cleaning --
6 -- Handling Missing Values and Outliers --
7
8 -- WarehouseToHome --
9 • set sql_safe_updates = 0;
10 • set @avg_WarehouseToHome=(select round(avg(WarehouseToHome))as avg_WarehouseToHome from customer_churn);
11 • update customer_churn
12   set WarehouseToHome = @avg_WarehouseToHome
13   where WarehouseToHome is null;
14
15 -- HourSpendOnApp --
16 • set @avg_HourSpendOnApp=(select round(avg(HourSpendOnApp))as avg_HourSpendOnApp from customer_churn);
17 • update customer_churn

```

The output pane shows the execution log:

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|--|-----------------------|
| 18 | 21:58:22 | set @avg_HourSpendOnApp=(select round(avg(HourSpendOnApp))as avg_HourS... | 0 row(s) affected | 0.015 sec |
| 19 | 21:58:22 | update customer_churn set HourSpendOnApp = @avg_HourSpendOnApp where ... | 0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0 | 0.032 sec |
| 20 | 22:02:39 | select warehouseToHome from customer_churn LIMIT 0, 1000 | 1000 row(s) returned | 0.000 sec / 0.000 sec |
| 21 | 22:03:09 | select HourSpendOnApp from customer_churn LIMIT 0, 1000 | 1000 row(s) returned | 0.000 sec / 0.000 sec |
| 22 | 22:05:36 | set @avg_OrderAmountHikeFromlastYear=(select round(avg(OrderAmountHikeFro... | 0 row(s) affected | 0.015 sec |
| 23 | 22:05:36 | update customer_churn set OrderAmountHikeFromlastYear = @avg_OrderAmountH... | 265 row(s) affected Rows matched: 265 Changed: 265 Warnings: 0 | 0.032 sec |

MySQL Workbench

Local instance MySQL800 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

ecomm

- Tables
 - customer_churn
- Views
- Stored Procedures
- Functions

employee Administration Schemas

No object selected

Information

Object Info Session

sql E-commerce Customer churn SQL File 3... x

```

14
15  -- HourSpendOnApp --
16 •  set @avg_HourSpendOnApp=(select round(avg(HourSpendOnApp))as avg_HourSpendOnApp from customer_churn);
17 •  update customer_churn
18  set HourSpendOnApp = @avg_HourSpendOnApp
19  where HourSpendOnApp is null;
20
21  -- OrderAmountHikeFromlastYear --
22 •  set @avg_OrderAmountHikeFromlastYear=(select round(avg(OrderAmountHikeFromlastYear))as avg_OrderAmountHike
23 •  update customer_churn
24  set OrderAmountHikeFromlastYear = @avg_OrderAmountHikeFromlastYear
25  where OrderAmountHikeFromlastYear is null;
26
27

```

Output

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|--|-----------------------|
| 18 | 21:58:22 | set @avg_HourSpendOnApp=(select round(avg(HourSpendOnApp))as avg_HourSpendOnApp from customer_churn); | 0 row(s) affected | 0.015 sec |
| 19 | 21:58:22 | update customer_churn set HourSpendOnApp = @avg_HourSpendOnApp where HourSpendOnApp is null; | 0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0 | 0.032 sec |
| 20 | 22:02:39 | select warehouseToHome from customer_churn LIMIT 0, 1000 | 1000 row(s) returned | 0.000 sec / 0.000 sec |
| 21 | 22:03:09 | select HourSpendOnApp from customer_churn LIMIT 0, 1000 | 1000 row(s) returned | 0.000 sec / 0.000 sec |
| 22 | 22:05:36 | set @avg_OrderAmountHikeFromlastYear=(select round(avg(OrderAmountHikeFromlastYear))as avg_OrderAmountHikeFromlastYear); | 0 row(s) affected | 0.015 sec |
| 23 | 22:05:36 | update customer_churn set OrderAmountHikeFromlastYear = @avg_OrderAmountHikeFromlastYear; | 265 row(s) affected Rows matched: 265 Changed: 265 Warnings: 0 | 0.032 sec |

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

➤ Impute mode for the following columns: Tenure, CouponUsed, OrderCount.

MySQL Workbench

Local instance MySQL800 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

ecomm

- Tables
 - customer_churn
- Views
- Stored Procedures
- Functions

employee Administration Schemas

No object selected

Information

Object Info Session

sql E-commerce Customer churn SQL File 3... x

```

33  -- Impute mode for the following columns: Tenure, CouponUsed, OrderCount --
34 •  update customer_churn
35  set Tenure =@Tenure_mode
36  where Tenure is null;
37
38  -- couponUsed --
39 •  update customer_churn
40  set couponUsed =@couponUsed_mode
41  where couponUsed is null;
42
43  -- OrderCount --
44 •  update customer_churn
45  set OrderCount =@OrderCount_mode
46  where OrderCount is null;

```

Output

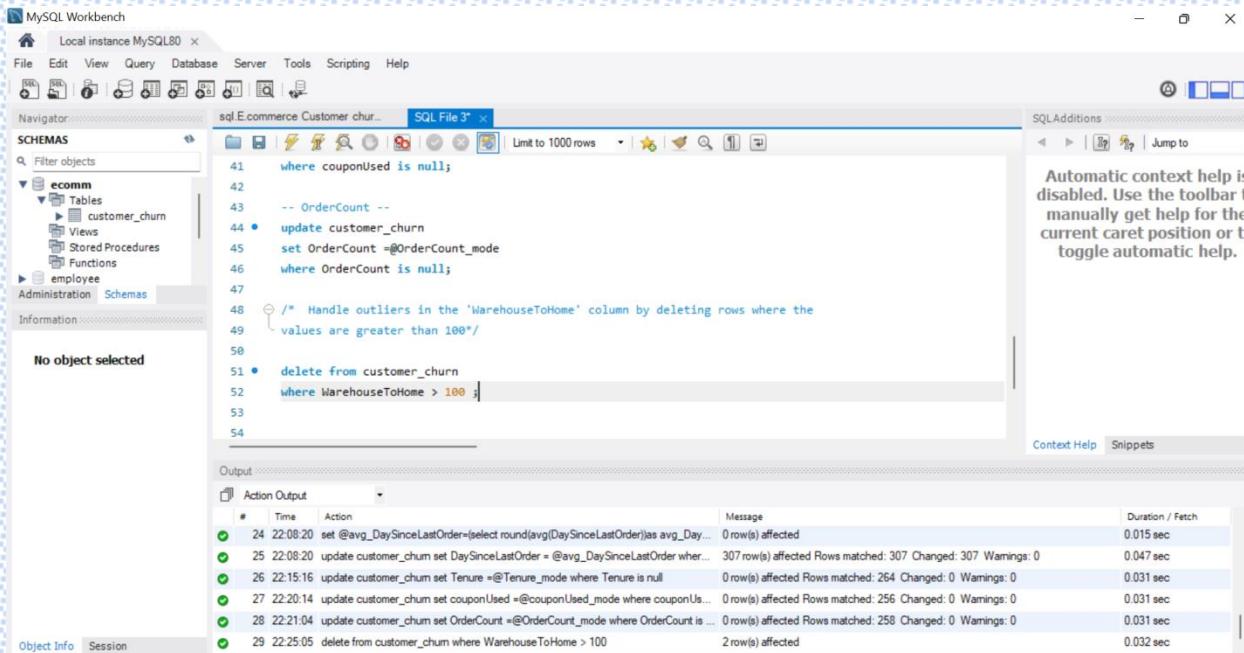
| # | Time | Action | Message | Duration / Fetch |
|----|----------|---|--|------------------|
| 23 | 22:05:36 | update customer_churn set OrderAmountHikeFromlastYear =@avg_OrderAmountHikeFromlastYear; | 265 row(s) affected Rows matched: 265 Changed: 265 Warnings: 0 | 0.032 sec |
| 24 | 22:08:20 | set @avg_DaySinceLastOrder=(select round(avg(DaySinceLastOrder))as avg_DaySinceLastOrder); | 0 row(s) affected | 0.015 sec |
| 25 | 22:08:20 | update customer_churn set DaySinceLastOrder = @avg_DaySinceLastOrder where DaySinceLastOrder is null; | 307 row(s) affected Rows matched: 307 Changed: 307 Warnings: 0 | 0.047 sec |
| 26 | 22:15:16 | update customer_churn set Tenure =@Tenure_mode where Tenure is null; | 0 row(s) affected Rows matched: 264 Changed: 0 Warnings: 0 | 0.031 sec |
| 27 | 22:20:14 | update customer_churn set couponUsed =@couponUsed_mode where couponUsed is null; | 0 row(s) affected Rows matched: 256 Changed: 0 Warnings: 0 | 0.031 sec |
| 28 | 22:21:04 | update customer_churn set OrderCount =@OrderCount_mode where OrderCount is null; | 258 row(s) affected Rows matched: 258 Changed: 0 Warnings: 0 | 0.031 sec |

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

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



The screenshot shows the MySQL Workbench interface with the following details:

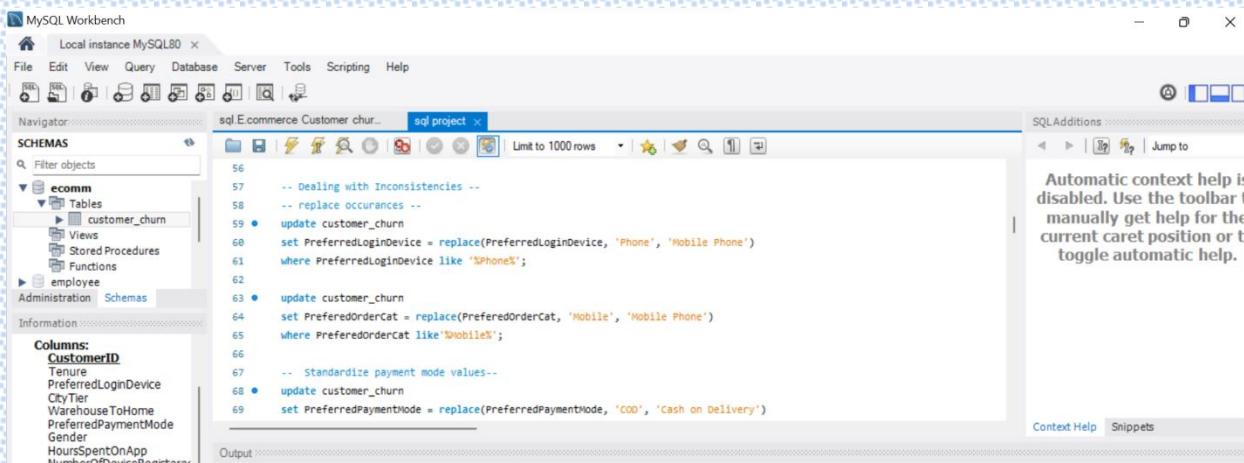
- Schemas:** ecomm (Tables: customer_churn, Views, Stored Procedures, Functions)
- SQL Editor:** SQL File 3*


```

41     where couponUsed is null;
42
43     -- OrderCount --
44 • update customer_churn
45     set OrderCount =@OrderCount_mode
46     where OrderCount is null;
47
48 /* Handle outliers in the 'WarehouseToHome' column by deleting rows where the
49 values are greater than 100*/
50
51 • delete from customer_churn
52     where WarehouseToHome > 100;
53
54
      
```
- Output:** Action Output table showing the results of the executed statements.

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|--|------------------|
| 24 | 22:08:20 | set @avg_DaySinceLastOrder=(select round(avg(DaySinceLastOrder))as avg_Day... | 0 row(s) affected | 0.015 sec |
| 25 | 22:08:20 | update customer_churn set DaySinceLastOrder = @avg_DaySinceLastOrder wher... | 307 row(s) affected Rows matched: 307 Changed: 307 Warnings: 0 | 0.047 sec |
| 26 | 22:15:16 | update customer_churn set Tenure =@Tenure_mode where Tenure is null | 0 row(s) affected Rows matched: 264 Changed: 0 Warnings: 0 | 0.031 sec |
| 27 | 22:20:14 | update customer_churn set couponUsed =@couponUsed_mode where couponUs... | 0 row(s) affected Rows matched: 256 Changed: 0 Warnings: 0 | 0.031 sec |
| 28 | 22:21:04 | update customer_churn set OrderCount =@OrderCount_mode where OrderCount is ... | 0 row(s) affected Rows matched: 258 Changed: 0 Warnings: 0 | 0.031 sec |
| 29 | 22:25:05 | delete from customer_churn where WarehouseToHome > 100 | 2 row(s) affected | 0.032 sec |

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



The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** ecomm (Tables: customer_churn, Views, Stored Procedures, Functions)
- SQL Editor:** SQL project*


```

56
57     -- Dealing with Inconsistencies --
58     -- replace occurrences --
59 • update customer_churn
60     set PreferredLoginDevice = replace(PreferredLoginDevice, 'Phone', 'Mobile Phone')
61     where PreferredLoginDevice like '%Phone%';
62
63 • update customer_churn
64     set PreferredOrderCat = replace(PreferredOrderCat, 'Mobile', 'Mobile Phone')
65     where PreferredOrderCat like '%Mobile%';
66
67     -- Standardize payment mode values--
68 • update customer_churn
69     set PreferredPaymentMode = replace(PreferredPaymentMode, 'COO', 'Cash on Delivery')
      
```
- Output:** Action Output table showing the results of the executed statements.

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

The screenshot shows the MySQL Workbench interface. The title bar says "MySQL Workbench". The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, Help. The toolbar has icons for New, Open, Save, Run, Stop, Refresh, and Help. The left sidebar shows "Local instance MySQL80" and "Navigator". Under "SCHEMAS", there is a tree view with "ecommerce" selected, showing Tables (customer_churn), Views, Stored Procedures, Functions, and a "Views" node under "employee". Below the schema tree are "Administration" and "Schemas" buttons. The main area is a "sql project" tab with the following SQL code:

```
-- Standardize payment mode values--  
update customer_churn  
set PreferredPaymentMode = replace(PreferredPaymentMode, 'C0D', 'Cash on Delivery')  
where PreferredPaymentMode like '%C0D%';  
  
update customer_churn  
set PreferredPaymentMode = replace(PreferredPaymentMode, 'CC', 'Credit Card')  
where PreferredPaymentMode like '%CC%';  
  
-- Data Transformation --  
-- Column Renaming --  
-- Rename the column "PreferredOrderCat" to "PreferredOrderCat"--  
alter table customer_churn  
rename column PreferredOrderCat to PreferredOrderCat;
```

The status bar at the bottom right shows "SQLAdditions" and "Jump to". A help icon in the top right corner has a tooltip: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." The bottom right also has "Context Help" and "Snippets" buttons.

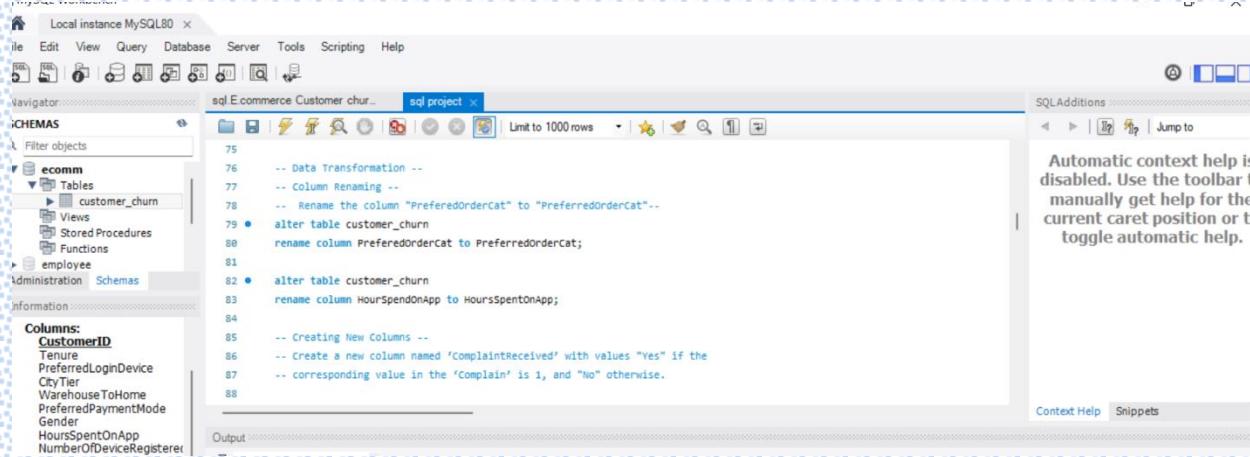
- Rename the column "PreferredOrderCat" to "PreferredOrderCat".

The screenshot shows the MySQL Workbench interface with a SQL project titled "sql project". The code in the editor is as follows:

```
75 -- Data Transformation --
76 -- Column Renaming --
77 -- Rename the column "PreferredOrderCat" to "PreferredOrderCat"--.
78 alter table customer_churn
79 rename column PreferredOrderCat to PreferredOrderCat;
80
81
82 alter table customer_churn
83 rename column HourSpendOnApp to HoursSpentOnApp;
84
85 -- Creating New Columns --
86 -- Create a new column named 'ComplaintReceived' with values "Yes" if the
87 -- corresponding value in the 'Complain' is 1, and "No" otherwise.
88
```

The left sidebar shows the schema browser with the "ecom" database selected, displaying tables like "customer_churn", views, stored procedures, and functions. The "employee" schema is also listed. The bottom left shows the "Columns" for the "CustomerID" table.

➤ Rename the column "HourSpendOnApp" to "HoursSpentOnApp". Creating New Columns:



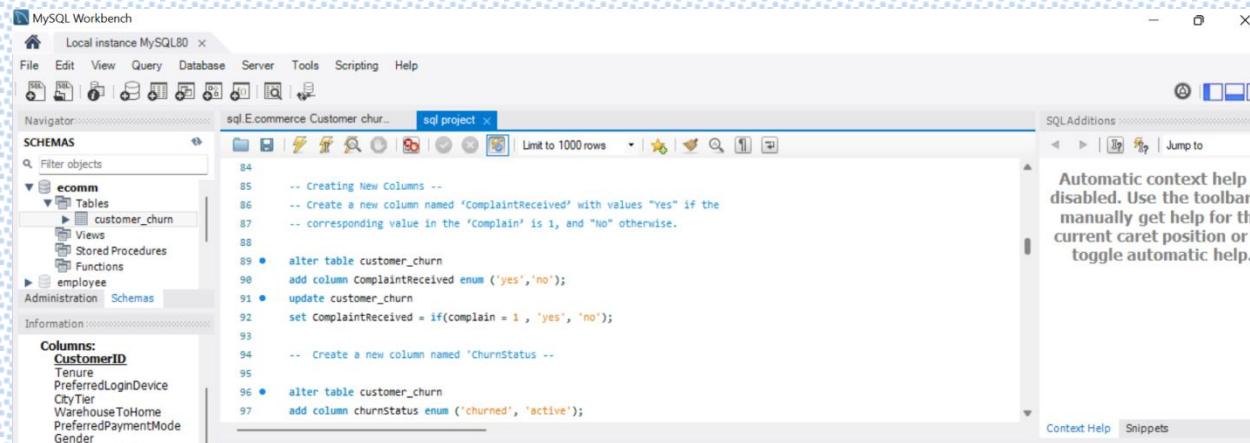
The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard MySQL icons for file operations, database navigation, and search.
- Navigator:** Shows the schema 'ecomm' with tables like 'customer_churn'. The 'customer_churn' table is selected.
- SQL Editor:** Contains the following SQL script:


```

75    -- Data Transformation --
76    -- Column Renaming --
77    -- Rename the column "PreferredOrderCat" to "PreferredOrderCat"-->
78    alter table customer_churn
79    rename column PreferredOrderCat to PreferredOrderCat;
80
81    alter table customer_churn
82    rename column HourSpendOnApp to HoursSpentOnApp;
83
84    -- Creating New Columns --
85    -- Create a new column named 'ComplaintReceived' with values "Yes" if the
86    -- corresponding value in the 'Complain' is 1, and "No" otherwise.
87
88
89    alter table customer_churn
90    add column ComplaintReceived enum ('yes','no');
91    update customer_churn
92    set ComplaintReceived = if(complain = 1 , 'yes', 'no');
93
94    -- Create a new column named 'ChurnStatus' --
95
96    alter table customer_churn
97    add column ChurnStatus enum ('churned', 'active');
      
```
- SQLAdditions Panel:** Displays a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."
- Output Panel:** Shows the output of the executed SQL statements.
- Bottom Bar:** Context Help, Snippets.

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



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard MySQL icons for file operations, database navigation, and search.
- Navigator:** Shows the schema 'ecomm' with tables like 'customer_churn'. The 'customer_churn' table is selected.
- SQL Editor:** Contains the following SQL script:


```

84
85    -- Creating New Columns --
86    -- Create a new column named 'ComplaintReceived' with values "Yes" if the
87    -- corresponding value in the 'Complain' is 1, and "No" otherwise.
88
89    alter table customer_churn
90    add column ComplaintReceived enum ('yes','no');
91    update customer_churn
92    set ComplaintReceived = if(complain = 1 , 'yes', 'no');
93
94    -- Create a new column named 'ChurnStatus' --
95
96    alter table customer_churn
97    add column ChurnStatus enum ('churned', 'active');
      
```
- SQLAdditions Panel:** Displays a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."
- Output Panel:** Shows the output of the executed SQL statements.
- Bottom Bar:** Context Help, Snippets.

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

```

MySQL Workbench - Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator: sql_E-commerce Customer churn... sql project
schemas
  ecomm
    Tables: customer_churn
    Views
    Stored Procedures
    Functions
  employee
Administration Schemas
Information
Columns:
  CustomerID
  Tenure
  PreferredLoginDevice
  CityTier
  WarehouseToHome
  PreferredPaymentMode
  Gender
  HoursSpentOnApp
  NumberOfDeviceRegistered
  PreferredOrderCat

Schemas: ecomm
Tables: customer_churn
Views
Stored Procedures
Functions
employee

SQLAdditions
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

```

-- Create a new column named 'ChurnStatus' --

```

94  alter table customer_churn
95  add column churnStatus enum ('churned', 'active');
96  update customer_churn
97  set churnstatus = if (churn = 1, 'churned', 'active');
98
99  -- Column Dropping --
100 -- Drop the columns "Churn" and "Complain" from the table
101
102 alter table customer_churn
103 drop column churn,
104 drop column complain;
105
106
107

```

- Drop the columns "Churn" and "Complain" from the table. Data Exploration and Analysis:

```

MySQL Workbench - Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator: sql_E-commerce Customer churn... sql project
schemas
  ecomm
    Tables: customer_churn
    Views
    Stored Procedures
    Functions
  employee
Administration Schemas
Information
Columns:
  CustomerID
  Tenure
  PreferredLoginDevice
  CityTier
  WarehouseToHome
  PreferredPaymentMode
  Gender
  HoursSpentOnApp
  NumberOfDeviceRegistered

Schemas: ecomm
Tables: customer_churn
Views
Stored Procedures
Functions
employee

SQLAdditions
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

```

-- Column Dropping --

```

101 -- Drop the columns "Churn" and "Complain" from the table
102
103
104 alter table customer_churn
105 drop column churn,
106 drop column complain;
107
108 -- Data Exploration and Analysis:
109 -- Retrieve the count of churned and active customers from the dataset --
110 select count(*) churned from customer_churn;
111 select count(*) active_customer from customer_churn;
112
113 -- Display the average tenure and total cashback amount of customers who churned. --
114 select ChurnStatus, avg(Tenure) as avg_tenure,

```

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

```

MySQL Workbench - Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator: sql_E-commerce Customer churn... sql project
schemas
  ecomm
    Tables: customer_churn
    Views
    Stored Procedures
    Functions
  employee
Administration Schemas
Information
Columns:
  CustomerID
  Tenure
  PreferredLoginDevice
  CityTier
  WarehouseToHome
  PreferredPaymentMode
  Gender
  HoursSpentOnApp
  NumberOfDeviceRegistered

Schemas: ecomm
Tables: customer_churn
Views
Stored Procedures
Functions
employee

SQLAdditions
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

```

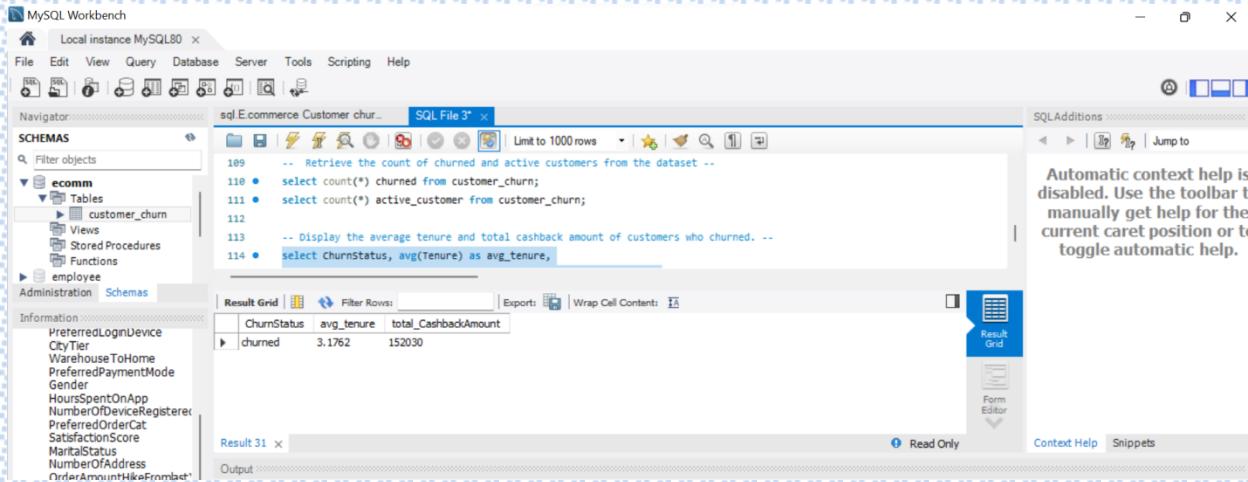
-- Column Dropping --

```

101 -- Drop the columns "Churn" and "Complain" from the table
102
103
104 alter table customer_churn
105 drop column churn,
106 drop column complain;
107
108 -- Data Exploration and Analysis:
109 -- Retrieve the count of churned and active customers from the dataset --
110 select count(*) churned from customer_churn;
111 select count(*) active_customer from customer_churn;
112
113 -- Display the average tenure and total cashback amount of customers who churned. --
114 select ChurnStatus, avg(Tenure) as avg_tenure,

```

➤ Display the average tenure and total cashback amount of customers who churned.



The screenshot shows the MySQL Workbench interface with a query editor window titled "sql E-commerce Customer churn...". The code retrieves the count of churned and active customers, and then displays the average tenure and total cashback amount for churned customers. The result grid shows one row with ChurnStatus as 'churned', avg_tenure as 3.1762, and total_CashbackAmount as 152030.

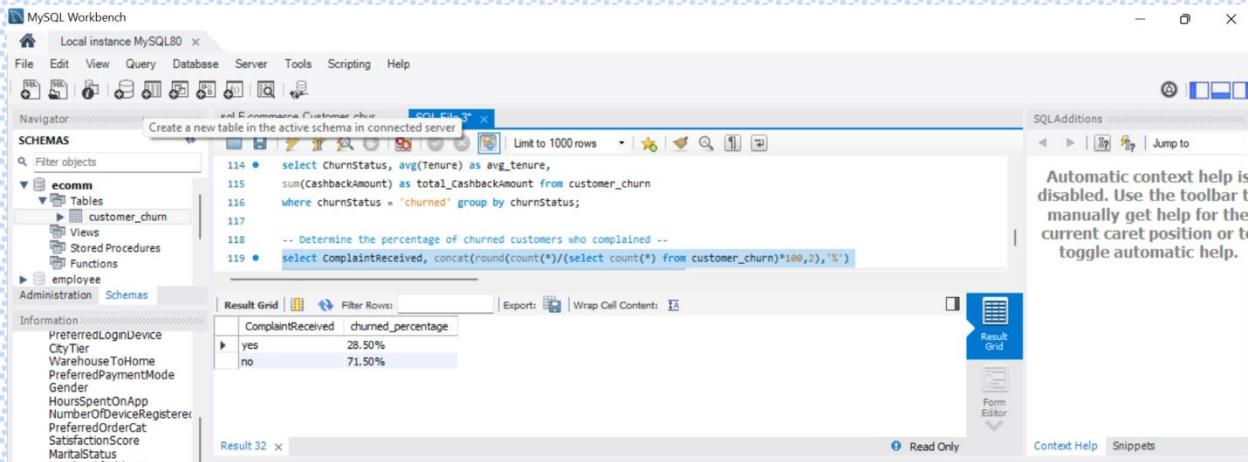
```

109  -- Retrieve the count of churned and active customers from the dataset --
110  select count(*) churned from customer_churn;
111  select count(*) active_customer from customer_churn;
112
113  -- Display the average tenure and total cashback amount of customers who churned. --
114  select ChurnStatus, avg(Tenure) as avg_tenure,

```

| ChurnStatus | avg_tenure | total_CashbackAmount |
|-------------|------------|----------------------|
| churned | 3.1762 | 152030 |

➤ Determine the percentage of churned customers who complained.



The screenshot shows the MySQL Workbench interface with a query editor window titled "sql E-commerce Customer churn...". The code calculates the average tenure and total cashback for churned customers, and then determines the percentage of churned customers who complained. The result grid shows two rows: one for 'yes' with 28.50% and one for 'no' with 71.50%.

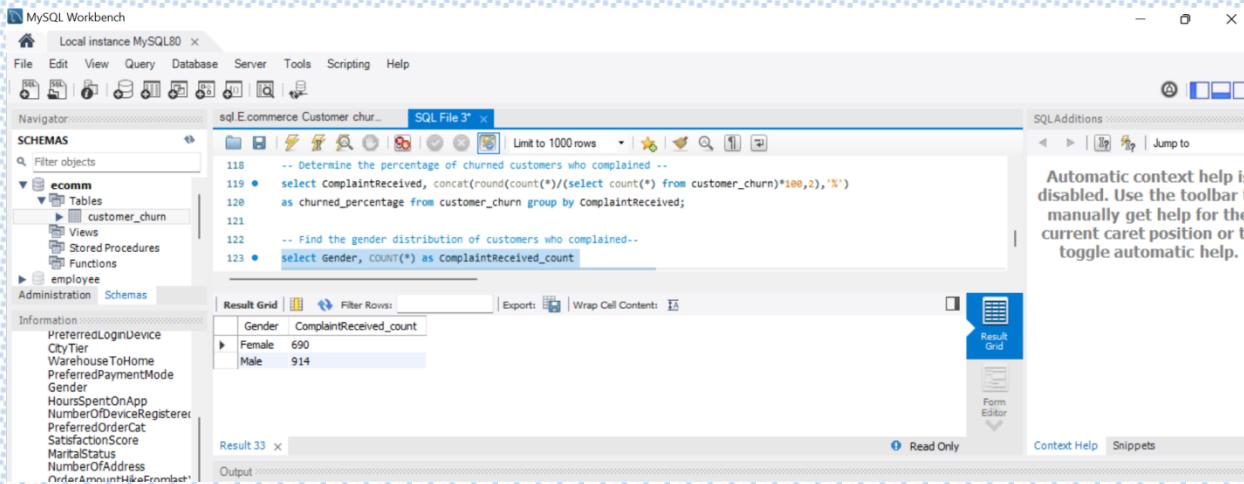
```

114  select ChurnStatus, avg(Tenure) as avg_tenure,
115  sum(CashbackAmount) as total_cashbackAmount from customer_churn
116  where ChurnStatus = 'churned' group by ChurnStatus;
117
118  -- Determine the percentage of churned customers who complained --
119  select ComplaintReceived, concat(round(count(*)/(select count(*) from customer_churn)*100,2),'%')

```

| ComplaintReceived | churned_percentage |
|-------------------|--------------------|
| yes | 28.50% |
| no | 71.50% |

➤ Find the gender distribution of customers who complained.



The screenshot shows the MySQL Workbench interface with the following details:

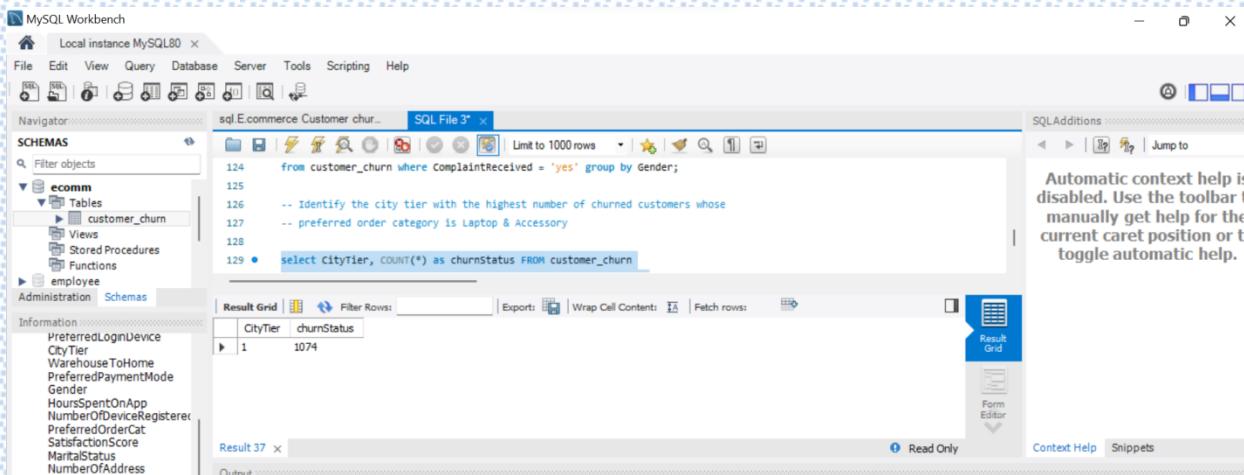
- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80, Schemas (selected), ecomm, employee.
- SQL Editor:** SQL File 3*, containing the following SQL code:

```

118 -- Determine the percentage of churned customers who complained --
119 select ComplaintReceived, concat(round(count(*)/(select count(*) from customer_churn)*100,2),'%')
120 as churned_percentage from customer_churn group by ComplaintReceived;
121
122 -- Find the gender distribution of customers who complained--
123 select Gender, COUNT(*) as ComplaintReceived_count
    
```
- Result Grid:** Shows the results of the last query:

| Gender | ComplaintReceived_count |
|--------|-------------------------|
| Female | 690 |
| Male | 914 |
- Information:** PreferredLoginDevice, CityTier, WarehouseToHome, PreferredPaymentMode, Gender, HoursSpentOnApp, NumberOfDeviceRegistered, PreferredOrderCat, SatisfactionScore, MaritalStatus, NumberOfAddress, OrderAmountWkAgoFromLastBuy.
- Output:** Result 33, Read Only, Context Help, Snippets.

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



The screenshot shows the MySQL Workbench interface with the following details:

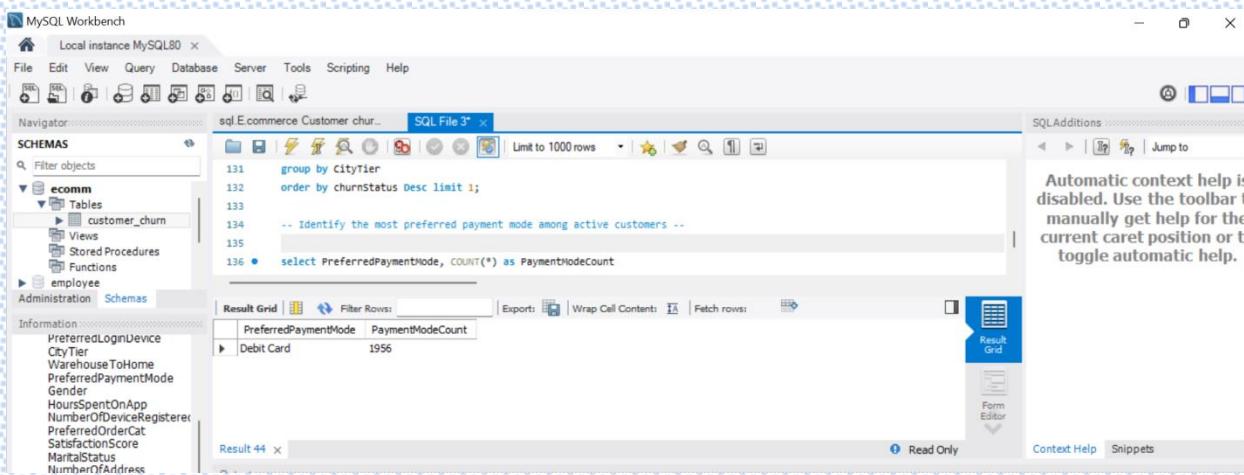
- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80, Schemas (selected), ecomm, employee.
- SQL Editor:** SQL File 3*, containing the following SQL code:

```

124 from customer_churn where ComplaintReceived = 'yes' group by Gender;
125
126 -- Identify the city tier with the highest number of churned customers whose
127 -- preferred order category is Laptop & Accessory
128
129 select CityTier, COUNT(*) as churnStatus FROM customer_churn
    
```
- Result Grid:** Shows the results of the last query:

| CityTier | chunStatus |
|----------|------------|
| 1 | 1074 |
- Information:** PreferredLoginDevice, CityTier, WarehouseToHome, PreferredPaymentMode, Gender, HoursSpentOnApp, NumberOfDeviceRegistered, PreferredOrderCat, SatisfactionScore, MaritalStatus, NumberOfAddress.
- Output:** Result 37, Read Only, Context Help, Snippets.

➤ Identify the most preferred payment mode among active customers.



The screenshot shows the MySQL Workbench interface with the following details:

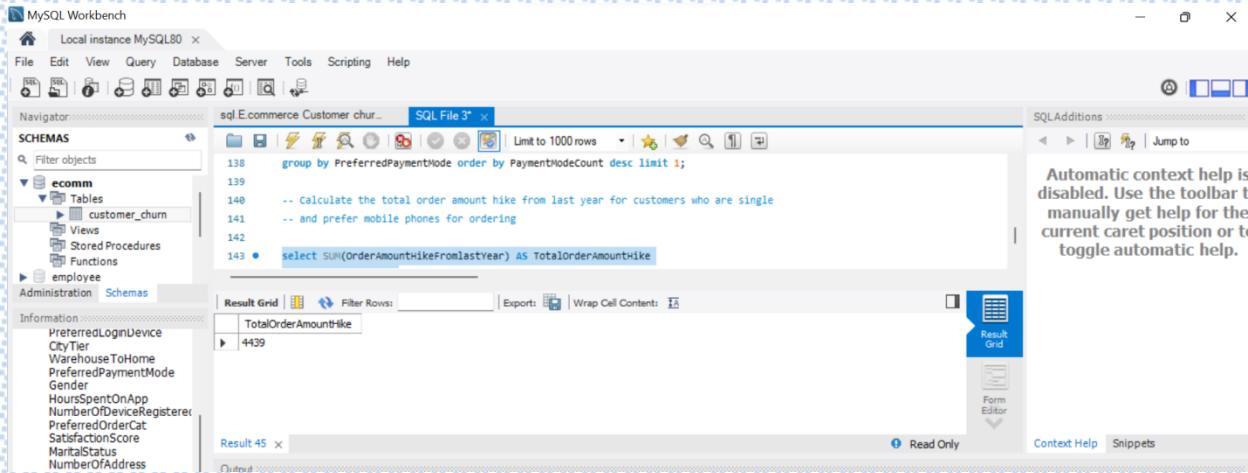
- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80, Schemas (selected), ecomm, employee.
- SQL Editor:** SQL File 3*, containing the following SQL code:

```

131 group by CityTier
132 order by chunstatus Desc limit 1;
133
134 -- Identify the most preferred payment mode among active customers --
135
136 select PreferredPaymentMode, COUNT(*) as PaymentModeCount
    
```
- Result Grid:** Shows the results of the last query:

| PreferredPaymentMode | PaymentModeCount |
|----------------------|------------------|
| Debit Card | 1956 |
- Information:** PreferredLoginDevice, CityTier, WarehouseToHome, PreferredPaymentMode, Gender, HoursSpentOnApp, NumberOfDeviceRegistered, PreferredOrderCat, SatisfactionScore, MaritalStatus, NumberOfAddress.
- Output:** Result 44, Read Only, Context Help, Snippets.

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



The screenshot shows the MySQL Workbench interface with a SQL editor window titled "sql E-commerce Customer churn...". The code in the editor is:

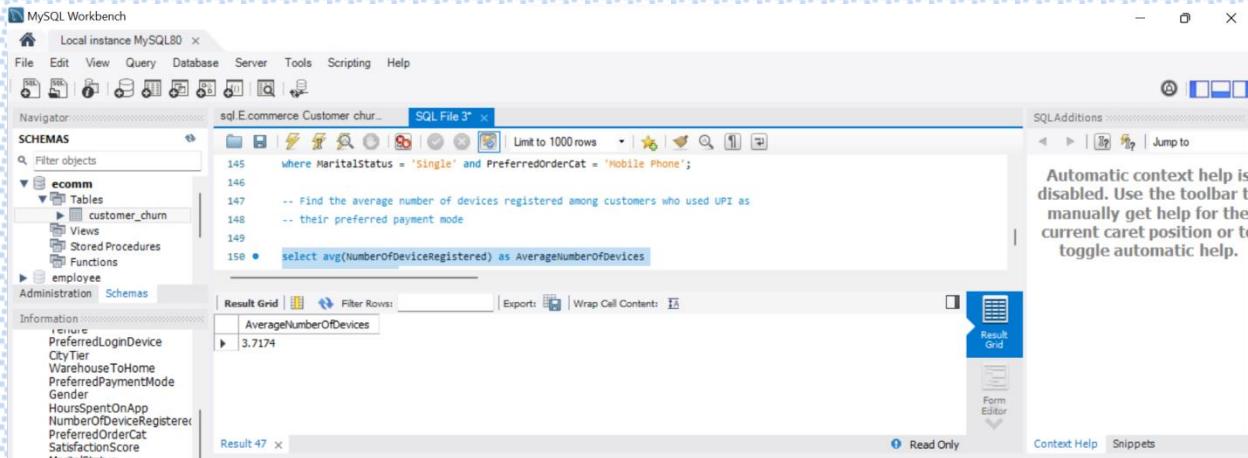
```

138 group by PreferredPaymentMode order by PaymentModeCount desc limit 1;
139
140 -- Calculate the total order amount hike from last year for customers who are single
141 -- and prefer mobile phones for ordering
142
143 • select sum(OrderAmountHikeFromLastYear) AS TotalOrderAmountHike

```

The result grid shows one row with the value 4439, labeled "TotalOrderAmountHike". A tooltip on the right side of the interface says: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

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



The screenshot shows the MySQL Workbench interface with a SQL editor window titled "sql E-commerce Customer churn...". The code in the editor is:

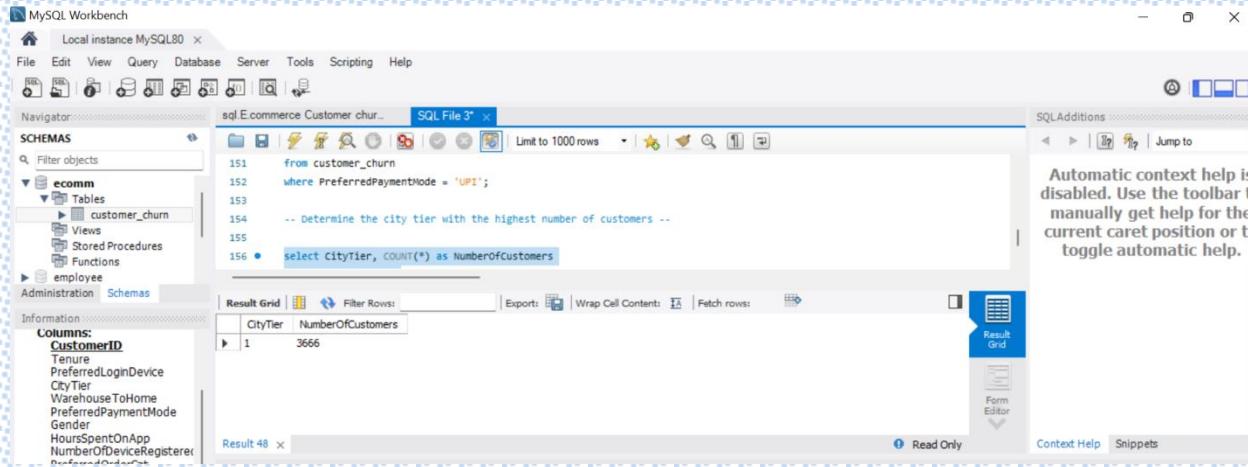
```

145 where MaritalStatus = 'Single' and PreferredOrderCat = 'Mobile Phone';
146
147 -- Find the average number of devices registered among customers who used UPI as
148 -- their preferred payment mode
149
150 • select avg(NumberOfDeviceRegistered) as AverageNumberOfDevices

```

The result grid shows one row with the value 3.7174, labeled "AverageNumberOfDevices". A tooltip on the right side of the interface says: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

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



The screenshot shows the MySQL Workbench interface with a query editor window titled "sql_E-commerce Customer churn...". The code is as follows:

```

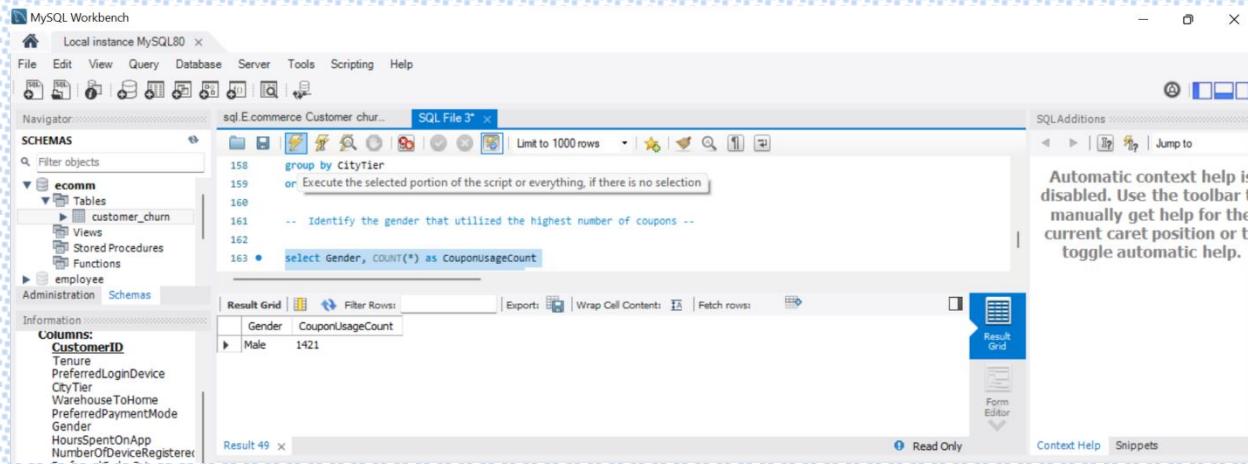
151  from customer_churn
152  where PreferredPaymentMode = 'UPI';
153
154  -- Determine the city tier with the highest number of customers --
155
156  • select CityTier, COUNT(*) as NumberOfCustomers

```

The result grid shows one row of data:

| CityTier | NumberOfCustomers |
|----------|-------------------|
| 1 | 3666 |

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



The screenshot shows the MySQL Workbench interface with a query editor window titled "sql_E-commerce Customer churn...". The code is as follows:

```

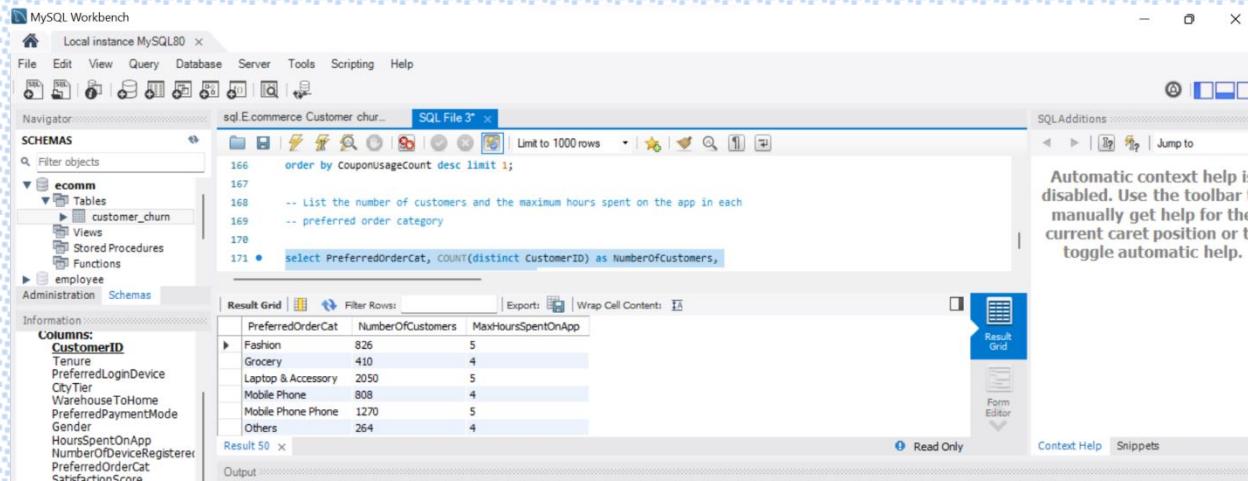
158  group by CityTier
159  or Execute the selected portion of the script or everything, if there is no selection
160
161  -- Identify the gender that utilized the highest number of coupons --
162
163  • select Gender, COUNT(*) as CouponUsageCount

```

The result grid shows one row of data:

| Gender | CouponUsageCount |
|--------|------------------|
| Male | 1421 |

- List the number of customers and the maximum hours spent on the app in each preferred order category.



The screenshot shows the MySQL Workbench interface with a query editor window titled "sql_E-commerce Customer churn...". The code is as follows:

```

166  order by CouponUsageCount desc limit 1;
167
168  -- List the number of customers and the maximum hours spent on the app in each
169  -- preferred order category
170
171  • select PreferredOrderCat, COUNT(distinct CustomerID) as NumberOfCustomers,

```

The result grid shows data for 10 categories:

| PreferredOrderCat | NumberOfCustomers | MaxHoursSpentOnApp |
|--------------------|-------------------|--------------------|
| Fashion | 826 | 5 |
| Grocery | 410 | 4 |
| Laptop & Accessory | 2050 | 5 |
| Mobile Phone | 808 | 4 |
| Mobile Phone Phone | 1270 | 5 |
| Others | 264 | 4 |
| | Result 50 | |

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas: ecomm Tables: customer_churn Views Stored Procedures Functions

SQL File 3* | Limit to 1000 rows | Result Grid | Filter Rows: | Export: | Wrap Cell Content: | SQLAdditions | Context Help Snippets

```

174 group by PreferredOrderCat;
175   -- Execute the selected portion of the script or everything, if there is no selection
176   -- calculate the total order count for customers who prefer using credit cards and
177   -- have the maximum satisfaction score.
178
179 • select sum(ordercount) as total_ordercount,

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | SQLAdditions | Context Help Snippets

| total_ordercount | maximum_Satisfaction_Score |
|------------------|----------------------------|
| 5409 | 5 |

Result 51 x | Output: | Read Only | SQLAdditions | Context Help Snippets

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

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas: ecomm Tables: customer_churn Views Stored Procedures Functions

SQL File 3* | Limit to 1000 rows | Result Grid | Filter Rows: | Export: | Wrap Cell Content: | SQLAdditions | Context Help Snippets

```

181 where preferredpaymentmode = 'credit card';
182   -- Execute the selected portion of the script or everything, if there is no selection
183   -- How many customers are there who spent only one hour on the app and days
184   -- since their last order was more than 5?
185
186 • select count(distinct CustomerID) as numberOfCustomers

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | SQLAdditions | Context Help Snippets

| NumberOfCustomers |
|-------------------|
| 8 |

Result 52 x | Output: | Read Only | SQLAdditions | Context Help Snippets

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

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas: ecomm Tables: customer_churn Views Stored Procedures Functions

SQL File 3* | Limit to 1000 rows | Result Grid | Filter Rows: | Export: | Wrap Cell Content: | SQLAdditions | Context Help Snippets

```

187 from customer_churn where HoursSpentOnApp = 1
188 and DaysSinceLastOrder > 5;
189
190   -- what is the average satisfaction score of customers who have complained --
191
192 • select avg(SatisfactionScore) as AverageSatisfactionScore

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | SQLAdditions | Context Help Snippets

| AverageSatisfactionScore |
|--------------------------|
| 2.9988 |

Result 53 x | Output: | Read Only | SQLAdditions | Context Help Snippets

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

The screenshot shows the MySQL Workbench interface with a query editor window titled "SQL File 3". The code in the editor is:

```
193 from customer_churn  
194 where ComplaintReceived = 1;  
195  
196 -- List the preferred order category among customers who used more than 5 coupons.  
197  
198 select PreferredOrderCat, COUNT(*) as NumberOfCustomers
```

The results grid shows the following data:

| PreferredOrderCat | NumberOfCustomers |
|--------------------|-------------------|
| Others | 28 |
| Laptop & Accessory | 99 |
| Fashion | 89 |
| Mobile Phone | 15 |
| Grocery | 42 |
| Mobile Phone Phone | 30 |

Output: Result 54

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

The screenshot shows the MySQL Workbench interface with a query editor window titled "SQL File 3". The code in the editor is:

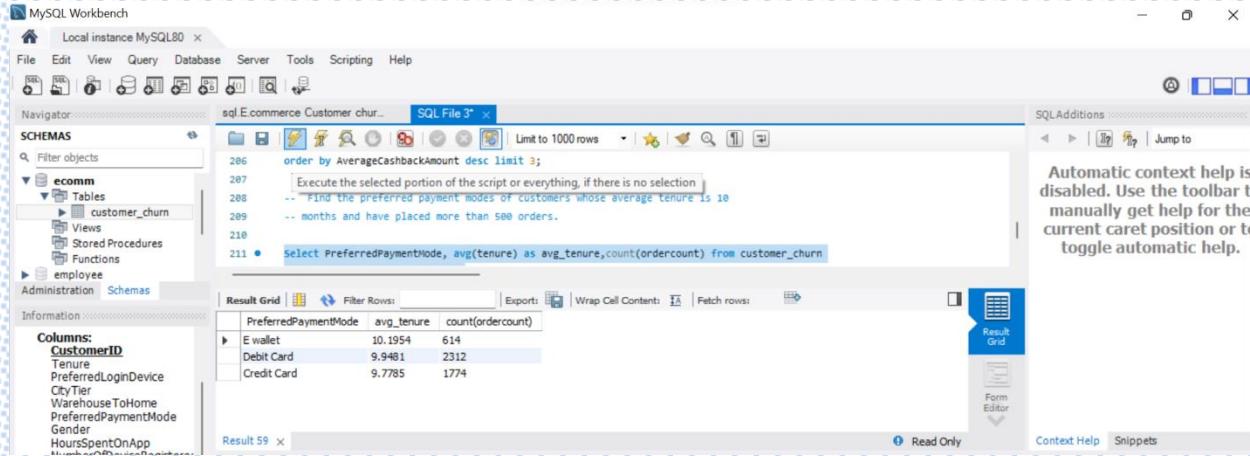
```
199 from customer_churn where CouponUsed > 5  
200 -- Execute the selected portion of the script or everything, if there is no selection  
201  
202 -- List the top 3 preferred order categories with the highest average cashback amount.  
203  
204 select PreferredOrderCat, avg(CashbackAmount) as AverageCashbackAmount
```

The results grid shows the following data:

| PreferredOrderCat | AverageCashbackAmount |
|-------------------|-----------------------|
| Others | 304.4545 |
| Grocery | 266.2366 |
| Fashion | 210.4031 |

Output: Result 55

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



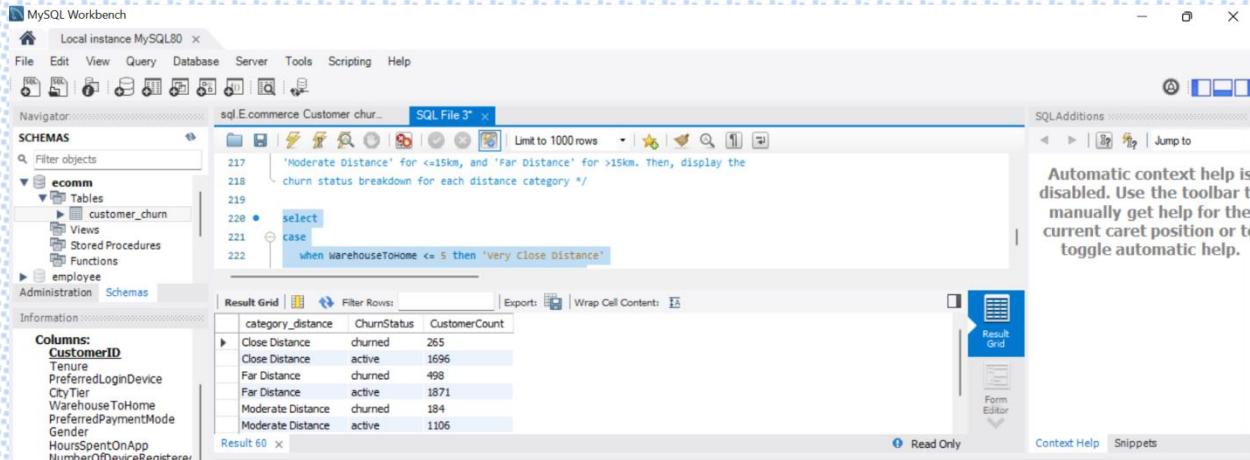
```

MySQL Workbench - Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator: Schemas SQL File 3*
Schemas
Filter objects
ecommerce Tables customer_churn
Views Stored Procedures Functions
employee Administration Schemas
Information Columns: CustomerID, Tenure, PreferredLoginDevice, CityTier, WarehouseToHome, PreferredPaymentMode, Gender, HoursSpentOnApp, NumberofDeviceRegistered
Result Grid Filter Rows: Export: Wrap Cell Content: Fetch Rows: Result Grid Form Editor
PreferredPaymentMode avg_tenure count(ordercount)
E wallet 10.1954 614
Debit Card 9.9481 2312
Credit Card 9.7785 1774

```

The screenshot shows the MySQL Workbench interface. The SQL editor contains a query to find the preferred payment modes of customers whose average tenure is 10 months and have placed more than 500 orders. The results are displayed in a grid, showing three rows: E wallet, Debit Card, and Credit Card, with their respective average tenure and count of order counts.

- Categorize customers based on their distance from the warehouse to home such as 'Very Close Distance' for distances ≤ 5 km, 'Close Distance' for ≤ 10 km, 'Moderate Distance' for ≤ 15 km, and 'Far Distance' for > 15 km. Then, display the churn status breakdown for each distance category.



```

MySQL Workbench - Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator: Schemas SQL File 3*
Schemas
Filter objects
ecommerce Tables customer_churn
Views Stored Procedures Functions
employee Administration Schemas
Information Columns: CustomerID, Tenure, PreferredLoginDevice, CityTier, WarehouseToHome, PreferredPaymentMode, Gender, HoursSpentOnApp, NumberofDeviceRegistered
Result Grid Filter Rows: Export: Wrap Cell Content: Fetch Rows: Result Grid Form Editor
category_distance ChurnStatus CustomerCount
Close Distance churned 265
Close Distance active 1696
Far Distance churned 498
Far Distance active 1871
Moderate Distance churned 184
Moderate Distance active 1106

```

The screenshot shows the MySQL Workbench interface. The SQL editor contains a query to categorize customers based on their distance from the warehouse to home and display the churn status breakdown for each distance category. The results are displayed in a grid, showing six categories: Close Distance (265 churned, 1696 active), Far Distance (498 churned, 1871 active), and Moderate Distance (184 churned, 1106 active).

- 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.

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas: ecomm Tables: customer_churn Views Stored Procedures Functions employee Administration Schemas

Information: Columns: CustomerID, Tenure, PreferredLoginDevice, CityTier, WarehouseToHome, PreferredPaymentMode, Gender, HoursSpentOnApp, NumberOfDeviceRegistered, PreferredOrderCat

SQL File 3* Limit to 1000 rows

```

231 /* List the customer's order details who are married, live in City Tier-1, and their
232   or Execute the selected portion of the script or everything, if there is no selection */
233
234
235 -- without using CTE
236 • select customerID , round(avg(ordercount)) as avg_ordercount from customer_churn
  
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 61 x

| customerID | avg_ordercount |
|------------|----------------|
| 50133 | 3 |
| 50180 | 3 |
| 50600 | 3 |
| 50864 | 3 |
| 50965 | 3 |
| 51036 | 3 |

Result 61 x Read Only Context Help Snippets

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas: ecomm Tables: customer_churn Views Stored Procedures Functions employee Administration Schemas

Information: Columns: CustomerID, Tenure, PreferredLoginDevice, CityTier, WarehouseToHome, PreferredPaymentMode, Gender, HoursSpentOnApp, NumberOfDeviceRegistered, PreferredOrderCat

SQL File 3* Limit to 1000 rows

```

238 group by customerID
239   l Execute the selected portion of the script or everything, if there is no selection
240   order by avg_ordercount ;
241
242 -- using CTE
243 • with average_order_count as (select round(avg(ordercount))
  
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 62 x

| customerID | order_count |
|------------|-------------|
| 50119 | 4 |
| 50367 | 4 |
| 50385 | 4 |
| 50453 | 4 |
| 50727 | 4 |
| 50734 | 4 |

Result 62 x Read Only Context Help Snippets

- a) Create a 'customer_returns' table in the 'ecomm' database

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas: ecomm Tables: customer_churn Views Stored Procedures Functions employee Administration Schemas

Information: Columns: CustomerID, Tenure, PreferredLoginDevice, CityTier, WarehouseToHome, PreferredPaymentMode, Gender, HoursSpentOnApp, NumberOfDeviceRegistered

SQL File 3* Limit to 1000 rows

```

246 where maritalstatus = 'married' and citytier = 1
247 group by customerID
248 having (select avg_order from average_order_count) < order_count
249 order by order_count ;
250
251 -- Create a 'customer_returns' table in the 'ecomm' database and insert the following data:
252
253 • create table ecomm.customer_returns (
254   ReturnID INT primary key,
255   CustomerID INT,
256   ReturnDate DATE,
257   RefundAmount DECIMAL(10, 2)
258 );
  
```

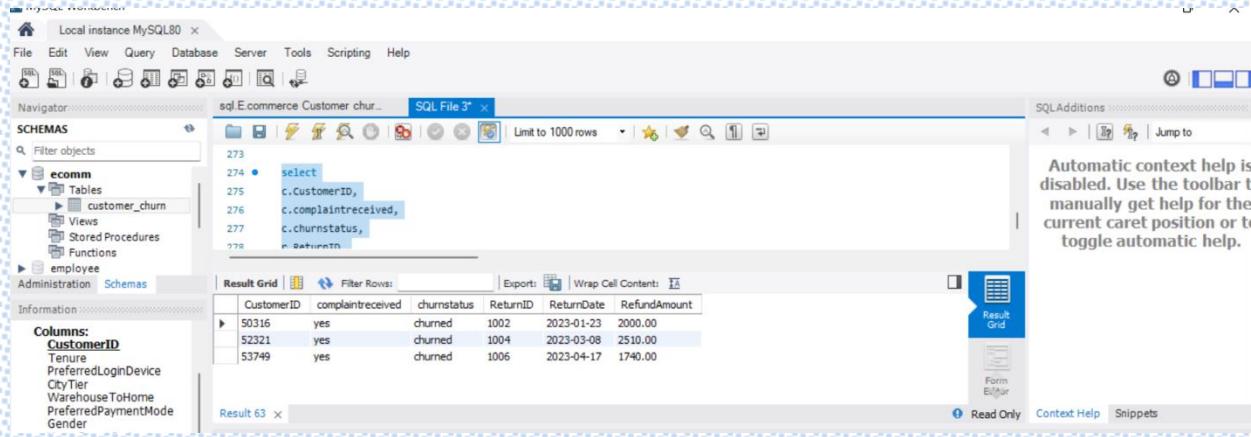
Output:

SQL Additions | Jump to

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

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



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree, with 'ecomm' selected, showing tables like 'customer_churn'. The main area contains a SQL editor window titled 'sql.E-commerce Customer chur...' with the following query:

```
273
274 • select
275   c.CustomerID,
276   c.complaintreceived,
277   c.churnstatus,
278   r.ReturnID,
```

The results grid shows the following data:

| CustomerID | complaintreceived | churnstatus | ReturnID | ReturnDate | RefundAmount |
|------------|-------------------|-------------|----------|------------|--------------|
| 50316 | yes | churned | 1002 | 2023-01-23 | 2000.00 |
| 52321 | yes | churned | 1004 | 2023-03-08 | 2510.00 |
| 53749 | yes | churned | 1006 | 2023-04-17 | 1740.00 |

The status bar at the bottom indicates 'Result 63 x'.