Customer Churn Project Report Writing

Introduction

Customer churn, or the rate at which customers stop doing business with a company, is a critical metric for businesses, especially in the telecom industry. High churn rates can significantly impact a company's revenue and growth potential. By understanding and analyzing customer churn, businesses can develop strategies to retain customers, improve services, and enhance overall customer satisfaction.

Objective

The primary objective of this project is to predict and analyze customer churn in the telecom industry using data visualization and analysis tools like Power BI and Tableau, and SQL queries. This will help in identifying at-risk customers, understanding the factors contributing to churn, and developing strategies to mitigate churn.

Problem Statement

The telecom industry faces a significant challenge with customer churn, leading to revenue loss and increased customer acquisition costs. To address this, it is essential to identify the factors leading to churn, analyze customer behavior, and implement targeted retention strategies. This project aims to provide a comprehensive solution for understanding and reducing customer churn through data-driven insights.

Agenda

Phase 1: Data Collection and Preparation

- **Data Collection:** Gather data from various sources including customer demographics, service usage, account information, and churn details.
- **Data Cleaning:** Remove duplicates, handle missing values, and correct inconsistencies.
- **Data Integration:** Combine data from different sources into a unified dataset.

Phase 2: Data Exploration and Feature Engineering

- **Exploratory Data Analysis (EDA):** Analyze the data to understand distributions, correlations, and patterns.
- **Feature Engineering:** Create new features that can help in better understanding and predicting churn, such as tenure, monthly charges, total charges, and interaction measures.

Phase 3: Visualization and Dashboard Creation

- **Power BI Visualizations:**
  - Identify the total number of customers and the churn rate.
  - Find the average age of churned customers.
  - Discover the most common contract types among churned customers.
  - Analyze the distribution of monthly charges among churned customers.

  - Identify customers with high total charges who have churned.
  - Calculate the average monthly charges for different contract types among churned customers.
  - Identify customers with the highest monthly charges in each contract type.
- **Tableau Visualizations:**
  - Calculate the total charges distribution for churned and non-churned customers.
  - Identify the gender distribution among customers who have churned and are on yearly contracts.
  - Find the customers who have churned and are not using online services, and their average total charges.

Phase 4: Business Use Case Recommendations

- **Customer Retention:**
  - Identify at-risk customers and proactively implement retention strategies.
- **Marketing Campaigns:**
  - Tailor marketing efforts towards customers who are more likely to churn.
- **Service Improvement:**
  - Analyze churn patterns to improve service offerings and customer support.
- **Revenue Optimization:**
  - Reduce churn rates to maintain a steady revenue stream.
- **Customer Segmentation:**
  - Segment customers based on churn probability to offer personalized experiences.

SQL-Based Analysis

- **Identify Contract Types Prone to Churn:** Create a query to identify which contract types are most associated with churn.
- **Calculate Churn Rate:** Develop a stored procedure to calculate the churn rate.
- **Identify High-Value Customers at Risk of Churning:** Develop a stored procedure to identify high-value customers who are at risk of churning.

Detailed Dashboard Explanations

Power BI Dashboard

- **Sum of Total Revenue:** Total revenue generated.
- **Sum of Total Charges:** Total charges incurred by customers.
- **Total Customers:** Total number of customers.

- **ARPU (Average Revenue Per User):** Revenue per user segmented by gender.
- **Avg Monthly Charges and Churned Customers by Contract Type:** Average monthly charges and the number of churned customers segmented by contract type.
- **Total Customers by Customer Status:** Number of customers segmented by their status (Stayed, Churned, Joined).
- **Sum of Monthly Charges by Monthly Charge Bins and Customer Status:** Monthly charges distribution for churned customers.
- **Churned Customers by Contract Type:** Number of churned customers segmented by contract type.
- **Total Churned Customers by Gender:** Distribution of churned customers segmented by gender.

Tableau Dashboard

- **Churned Customers - No Internet Service and Their Avg Total Charges:** Average total charges for churned customers who do not use internet services.
- **Gender Distribution - Churned Customers on Yearly Contracts:** Gender distribution of churned customers on yearly contracts.
- **Total Charges Distribution for Churned and Non-Churned Customers:** Comparison of total charges between churned and non-churned customers.

By following this agenda, the project aims to provide comprehensive insights into customer churn, enabling the telecom company to develop effective retention strategies and improve overall business performance.

Phase 1: Data Preparation and Initial Analysis

Step 1: Data Acquisition and Preprocessing

1.1 Data Download and Understanding

- **Data Source**: I started by downloading the initial dataset named customer_churn (1).xlsx. This dataset includes various attributes about customers, such as their demographic details, the services they use, and information about churn (whether they have left the service).

1.2 Conversion to Multiple Sheets

- **CSV to Excel:** To better organize and analyze the data, I converted the single CSV file into an Excel file with four separate sheets. This helps in managing the data more effectively.
- **Sheets Created:**
  - CustomerDemographics: This sheet contains demographic information about the customers.
  - AccountInformation: This sheet holds details about customer accounts.
  - ServiceInformation: This sheet includes information about the services used by the customers.
  - ChurnInformation: This sheet contains details about customer churn, such as reasons for leaving.

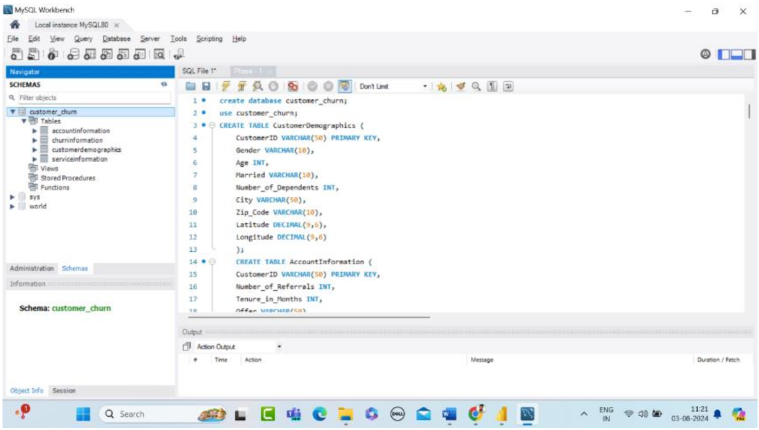
1.3 Folder and File Organization

- **CSV Folder:** For better organization, I created a folder named CSV and uploaded the original CSV file into it. This ensures I have a backup and can easily reference the original data.

Step 2: Data Loading into MySQL

2.1 Schema and Table Creation

- **Creating the Schema**: I set up a schema named customer_churn in MySQL to logically group the tables.



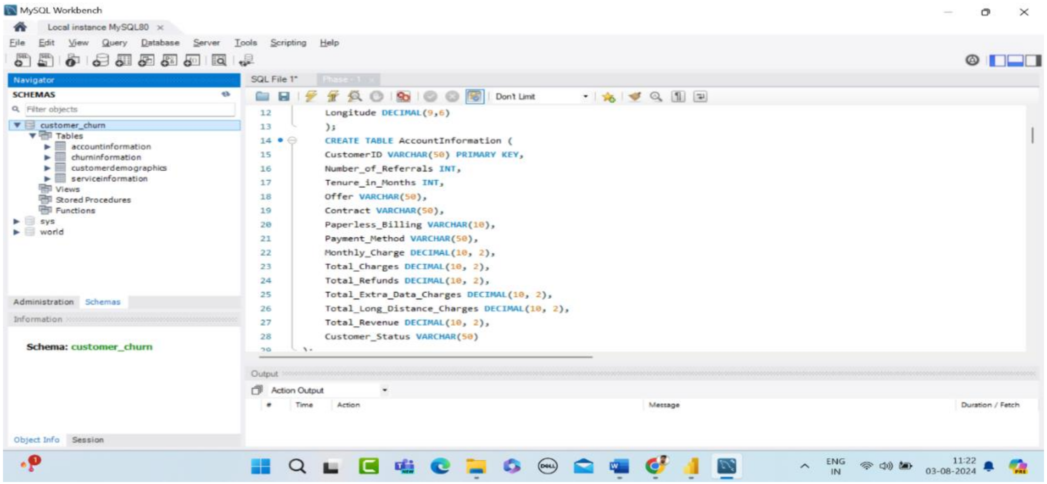
```
create database customer_churn;
use customer_churn;
CREATE TABLE CustomerDemographics (
    CustomerID VARCHAR(50) PRIMARY KEY,
    Gender VARCHAR(10),
    Age INT,
    Married VARCHAR(10),
    Number_of_Dependents INT,
    City VARCHAR(50),
    Zip_Code VARCHAR(10),
    Latitude DECIMAL(9,6),
    Longitude DECIMAL(9,6)
    );
    CREATE TABLE AccountInformation (
    CustomerID VARCHAR(50) PRIMARY KEY,
    Number_of_Referrals INT,
    Tenure_in_Months INT,
    Offer VARCHAR(50)
```

- **Creating the Tables:** I created four tables within this schema to match the Excel sheets:
  - CustomerDemographics
  - AccountInformation
  - ServiceInformation
  - ChurnInformation

These tables were created using SQL queries to define their structure accurately.

2.2 Data Import

- **Importing Data**: I imported the data from the Excel sheets into the corresponding MySQL tables using the data import table wizard.
- **Handling Errors:** During the import, I addressed issues like handling blank values to ensure all data was correctly loaded. After importing, I cross-checked the data with queries to verify its accuracy.



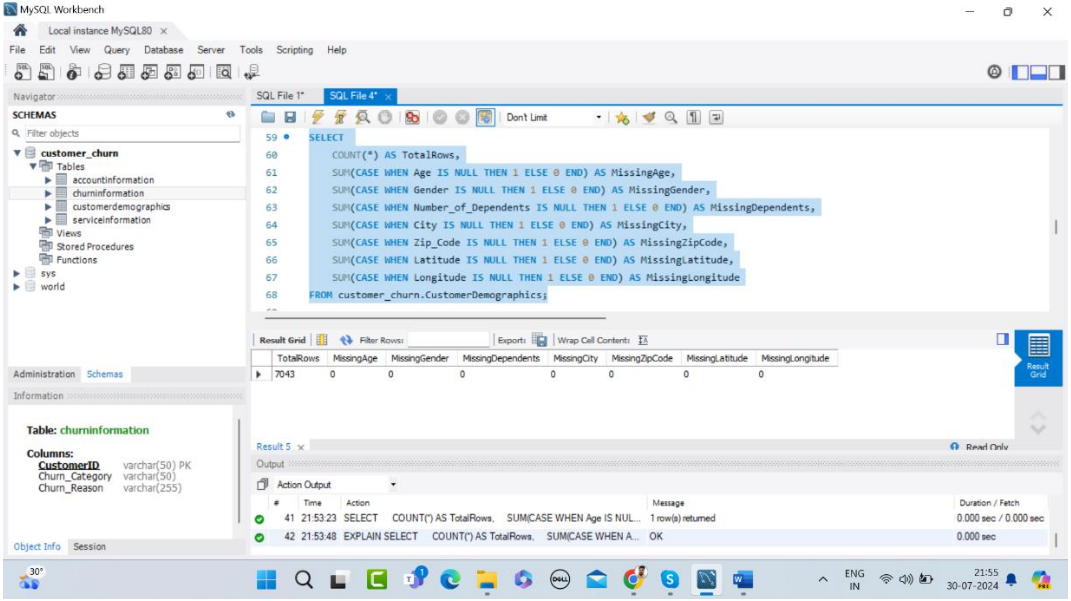
```sql
    Longitude DECIMAL(9,6)
    );
    CREATE TABLE AccountInformation (
    CustomerID VARCHAR(50) PRIMARY KEY,
    Number_of_Referrals INT,
    Tenure_in_Months INT,
    Offer VARCHAR(50),
    Contract VARCHAR(50),
    Paperless_Billing VARCHAR(10),
    Payment_Method VARCHAR(50),
    Monthly_Charge DECIMAL(10, 2),
    Total_Charges DECIMAL(10, 2),
    Total_Refunds DECIMAL(10, 2),
    Total_Extra_Data_Charges DECIMAL(10, 2),
    Total_Long_Distance_Charges DECIMAL(10, 2),
    Total_Revenue DECIMAL(10, 2),
    Customer_Status VARCHAR(50)
);
```

Step 3: Data Cleaning and Normalization

3.1 Handling Missing Values

- **Identifying Missing Values**: I identified any missing values in the dataset using SQL queries.
- **Filling Missing Values**: To handle these missing values, I filled them using appropriate methods,

Here’s an example of how I did it:



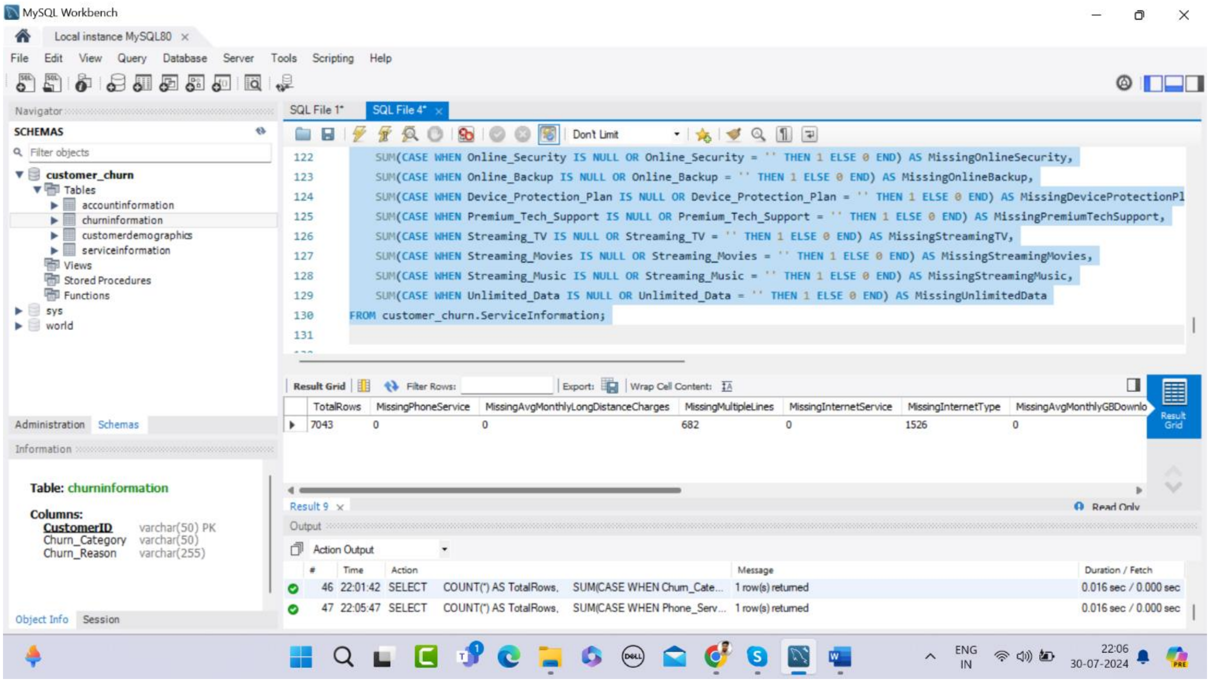
```sql
SELECT
    COUNT(*) AS TotalRows,
    SUM(CASE WHEN Age IS NULL THEN 1 ELSE 0 END) AS MissingAge,
    SUM(CASE WHEN Gender IS NULL THEN 1 ELSE 0 END) AS MissingGender,
    SUM(CASE WHEN Number_of_Dependents IS NULL THEN 1 ELSE 0 END) AS MissingDependents,
    SUM(CASE WHEN City IS NULL THEN 1 ELSE 0 END) AS MissingCity,
    SUM(CASE WHEN Zip_Code IS NULL THEN 1 ELSE 0 END) AS MissingZipCode,
    SUM(CASE WHEN Latitude IS NULL THEN 1 ELSE 0 END) AS MissingLatitude,
    SUM(CASE WHEN Longitude IS NULL THEN 1 ELSE 0 END) AS MissingLongitude
FROM customer_churn.CustomerDemographics;
```

TotalRows	MissingAge	MissingGender	MissingDependents	MissingCity	MissingZipCode	MissingLatitude	MissingLongitude
7043	0	0	0	0	0	0	0

Table: churninformation

Columns:
CustomerID varchar(50) PK
Churn_Category varchar(50)
Churn_Reason varchar(255)

```
#   Time      Action                                                                  Message             Duration / Fetch
41  21:53:23  SELECT    COUNT(*) AS TotalRows,   SUM(CASE WHEN Age IS NUL...          1 row(s) returned   0.000 sec / 0.000 sec
42  21:53:48  EXPLAIN SELECT    COUNT(*) AS TotalRows,   SUM(CASE WHEN A...           OK                  0.000 sec
```



```sql
    SUM(CASE WHEN Online_Security IS NULL OR Online_Security = '' THEN 1 ELSE 0 END) AS MissingOnlineSecurity,
    SUM(CASE WHEN Online_Backup IS NULL OR Online_Backup = '' THEN 1 ELSE 0 END) AS MissingOnlineBackup,
    SUM(CASE WHEN Device_Protection_Plan IS NULL OR Device_Protection_Plan = '' THEN 1 ELSE 0 END) AS MissingDeviceProtectionPl
    SUM(CASE WHEN Premium_Tech_Support IS NULL OR Premium_Tech_Support = '' THEN 1 ELSE 0 END) AS MissingPremiumTechSupport,
    SUM(CASE WHEN Streaming_TV IS NULL OR Streaming_TV = '' THEN 1 ELSE 0 END) AS MissingStreamingTV,
    SUM(CASE WHEN Streaming_Movies IS NULL OR Streaming_Movies = '' THEN 1 ELSE 0 END) AS MissingStreamingMovies,
    SUM(CASE WHEN Streaming_Music IS NULL OR Streaming_Music = '' THEN 1 ELSE 0 END) AS MissingStreamingMusic,
    SUM(CASE WHEN Unlimited_Data IS NULL OR Unlimited_Data = '' THEN 1 ELSE 0 END) AS MissingUnlimitedData
FROM customer_churn.ServiceInformation;
```

TotalRows	MissingPhoneService	MissingAvgMonthlyLongDistanceCharges	MissingMultipleLines	MissingInternetService	MissingInternetType	MissingAvgMonthlyGBDownlo
7043	0	0	682	0	1526	0

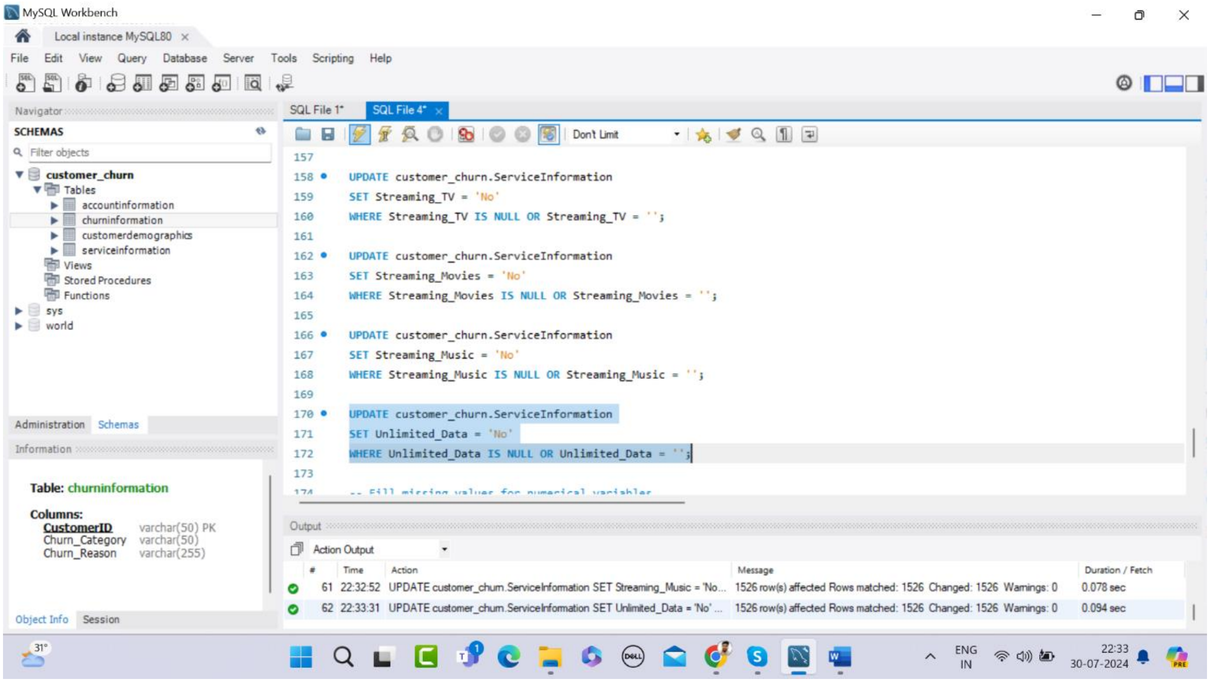
Table: churninformation

Columns:
CustomerID varchar(50) PK
Churn_Category varchar(50)
Churn_Reason varchar(255)

```
#   Time      Action                                                                  Message             Duration / Fetch
46  22:01:42  SELECT    COUNT(*) AS TotalRows,   SUM(CASE WHEN Churn_Cate...          1 row(s) returned   0.016 sec / 0.000 sec
47  22:05:47  SELECT    COUNT(*) AS TotalRows,   SUM(CASE WHEN Phone_Serv...          1 row(s) returned   0.016 sec / 0.000 sec
```

Found Missing Values in Service Information Table

Filling Missing Values and Verified



```
UPDATE customer_churn.ServiceInformation
SET Streaming_TV = 'No'
WHERE Streaming_TV IS NULL OR Streaming_TV = '';

UPDATE customer_churn.ServiceInformation
SET Streaming_Movies = 'No'
WHERE Streaming_Movies IS NULL OR Streaming_Movies = '';

UPDATE customer_churn.ServiceInformation
SET Streaming_Music = 'No'
WHERE Streaming_Music IS NULL OR Streaming_Music = '';

UPDATE customer_churn.ServiceInformation
SET Unlimited_Data = 'No'
WHERE Unlimited_Data IS NULL OR Unlimited_Data = '';

-- Fill missing values for numerical variables
```

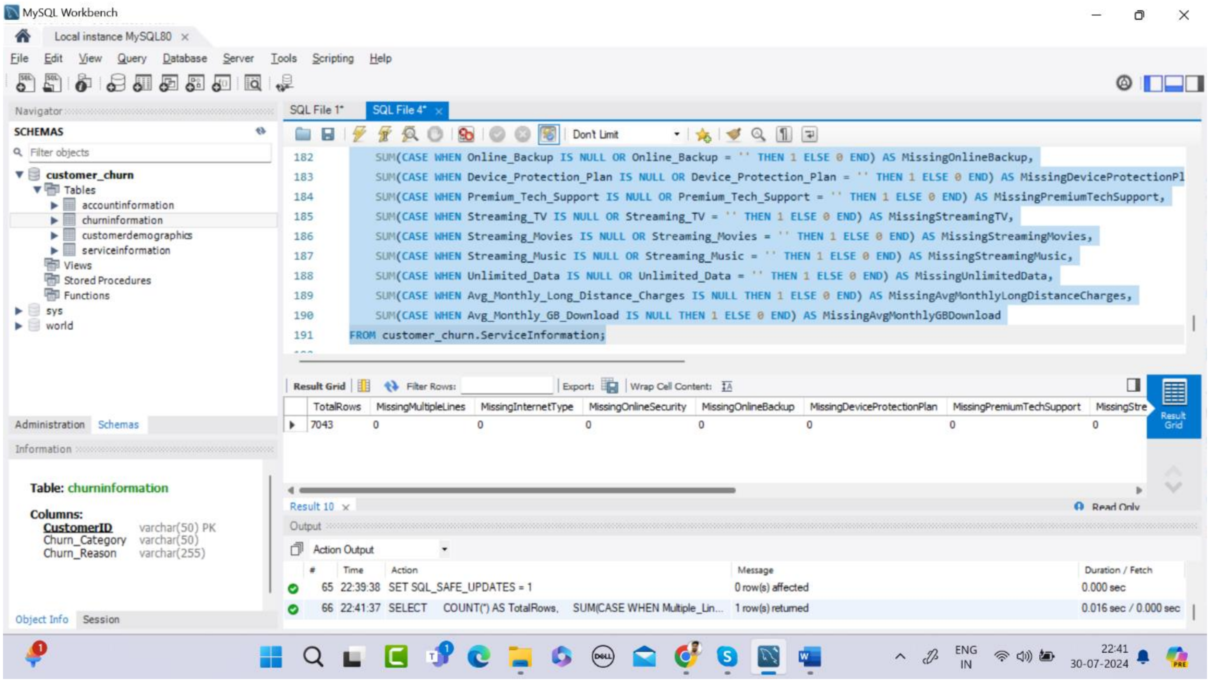
Table: churninformation

Columns:

CustomerID	varchar(50) PK
Churn_Category	varchar(50)
Churn_Reason	varchar(255)

Action Output

#	Time	Action	Message	Duration / Fetch
61	22:32:52	UPDATE customer_churn.ServiceInformation SET Streaming_Music = 'No...	1526 row(s) affected Rows matched: 1526 Changed: 1526 Warnings: 0	0.078 sec
62	22:33:31	UPDATE customer_churn.ServiceInformation SET Unlimited_Data = 'No' ...	1526 row(s) affected Rows matched: 1526 Changed: 1526 Warnings: 0	0.094 sec



```
    SUM(CASE WHEN Online_Backup IS NULL OR Online_Backup = '' THEN 1 ELSE 0 END) AS MissingOnlineBackup,
    SUM(CASE WHEN Device_Protection_Plan IS NULL OR Device_Protection_Plan = '' THEN 1 ELSE 0 END) AS MissingDeviceProtectionPl
    SUM(CASE WHEN Premium_Tech_Support IS NULL OR Premium_Tech_Support = '' THEN 1 ELSE 0 END) AS MissingPremiumTechSupport,
    SUM(CASE WHEN Streaming_TV IS NULL OR Streaming_TV = '' THEN 1 ELSE 0 END) AS MissingStreamingTV,
    SUM(CASE WHEN Streaming_Movies IS NULL OR Streaming_Movies = '' THEN 1 ELSE 0 END) AS MissingStreamingMovies,
    SUM(CASE WHEN Streaming_Music IS NULL OR Streaming_Music = '' THEN 1 ELSE 0 END) AS MissingStreamingMusic,
    SUM(CASE WHEN Unlimited_Data IS NULL OR Unlimited_Data = '' THEN 1 ELSE 0 END) AS MissingUnlimitedData,
    SUM(CASE WHEN Avg_Monthly_Long_Distance_Charges IS NULL THEN 1 ELSE 0 END) AS MissingAvgMonthlyLongDistanceCharges,
    SUM(CASE WHEN Avg_Monthly_GB_Download IS NULL THEN 1 ELSE 0 END) AS MissingAvgMonthlyGBDownload
FROM customer_churn.ServiceInformation;
```

TotalRows	MissingMultipleLines	MissingInternetType	MissingOnlineSecurity	MissingOnlineBackup	MissingDeviceProtectionPlan	MissingPremiumTechSupport	MissingStre
7043	0	0	0	0	0	0	0

Table: churninformation

Columns:

CustomerID	varchar(50) PK
Churn_Category	varchar(50)
Churn_Reason	varchar(255)

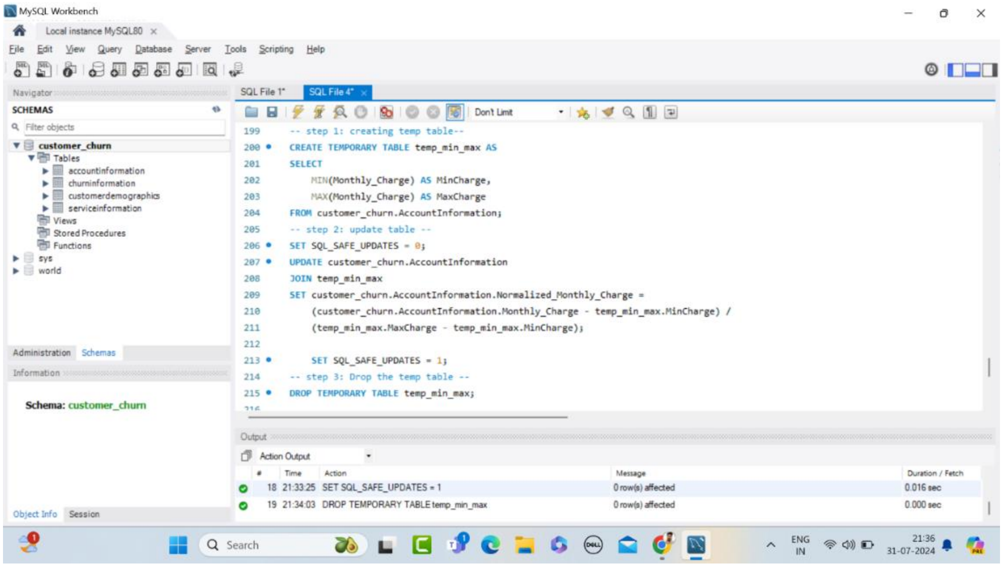
Action Output

#	Time	Action	Message	Duration / Fetch
65	22:39:38	SET SQL_SAFE_UPDATES = 1	0 row(s) affected	0.000 sec
66	22:41:37	SELECT COUNT(*) AS TotalRows, SUM(CASE WHEN Multiple_Lin...	1 row(s) returned	0.016 sec / 0.000 sec

3.2 Normalization

- **Normalization Process:** Normalized numerical features to bring them to a comparable scale.
  - Added new columns for Monthly Charge.
  - Calculated min and max monthly charge values using SQL.

- Drop temp table



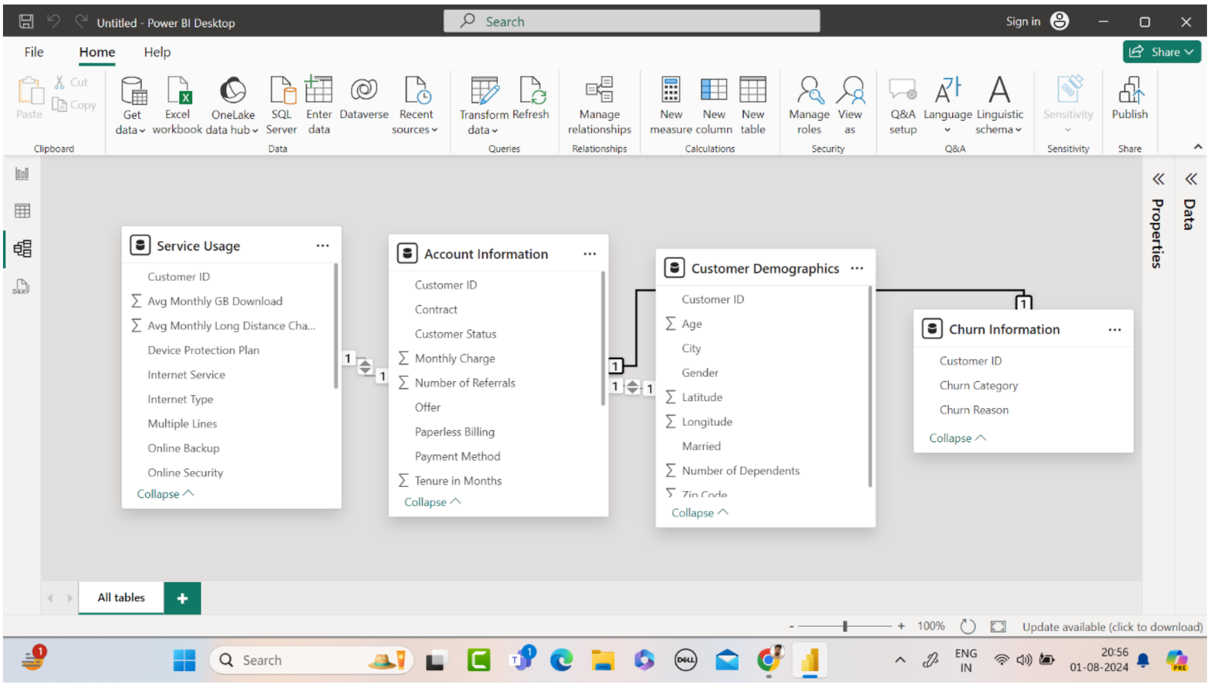
```sql
-- step 1: creating temp table--
CREATE TEMPORARY TABLE temp_min_max AS
SELECT
    MIN(Monthly_Charge) AS MinCharge,
    MAX(Monthly_Charge) AS MaxCharge
FROM customer_churn.AccountInformation;
-- step 2: update table --
SET SQL_SAFE_UPDATES = 0;
UPDATE customer_churn.AccountInformation
JOIN temp_min_max
SET customer_churn.AccountInformation.Normalized_Monthly_Charge =
    (customer_churn.AccountInformation.Monthly_Charge - temp_min_max.MinCharge) /
    (temp_min_max.MaxCharge - temp_min_max.MinCharge);

    SET SQL_SAFE_UPDATES = 1;
-- step 3: Drop the temp table --
DROP TEMPORARY TABLE temp_min_max;
```

```
#   Time      Action                              Message              Duration / Fetch
18  21:33:25  SET SQL_SAFE_UPDATES = 1            0 row(s) affected    0.016 sec
19  21:34:03  DROP TEMPORARY TABLE temp_min_max   0 row(s) affected    0.000 sec
```

Step 4: Exploratory Data Analysis (EDA) in Power BI

4.1 Data Upload and Relationship Definition

- **Uploading Data:** I uploaded the Excel file to Power BI for visualization and analysis.
- **Defining Relationships:** I defined relationships between the tables based on the Customer ID field to ensure the data could be analyzed as a connected dataset.



4.2 Summary Statistics

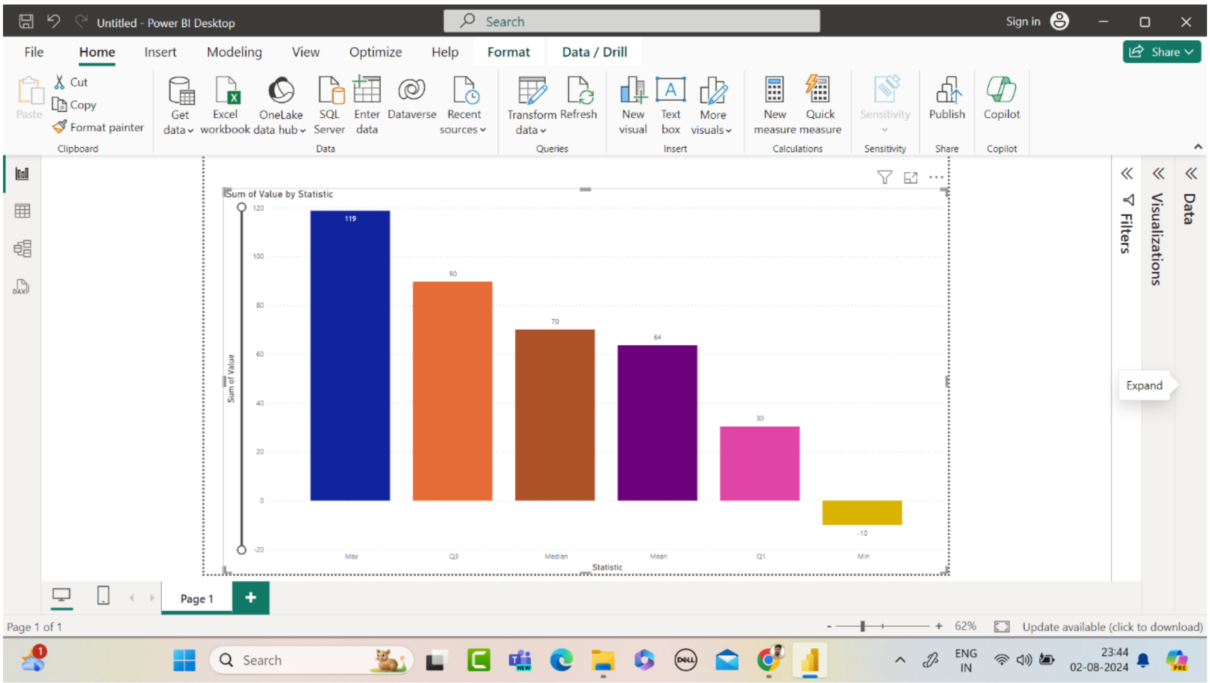
- **Calculating Measures:** I measured the following statistics using DAX in Power BI:
  - Mean
  - Median
  - Minimum (Min)
  - Maximum (Max)
  - First Quartile (Q1)
  - Third Quartile (Q3)

4.3 Correlation Analysis

- **Correlation Matrix:** I calculated the correlation values between numerical variables to identify relationships and dependencies**.**

4.4 Visualization

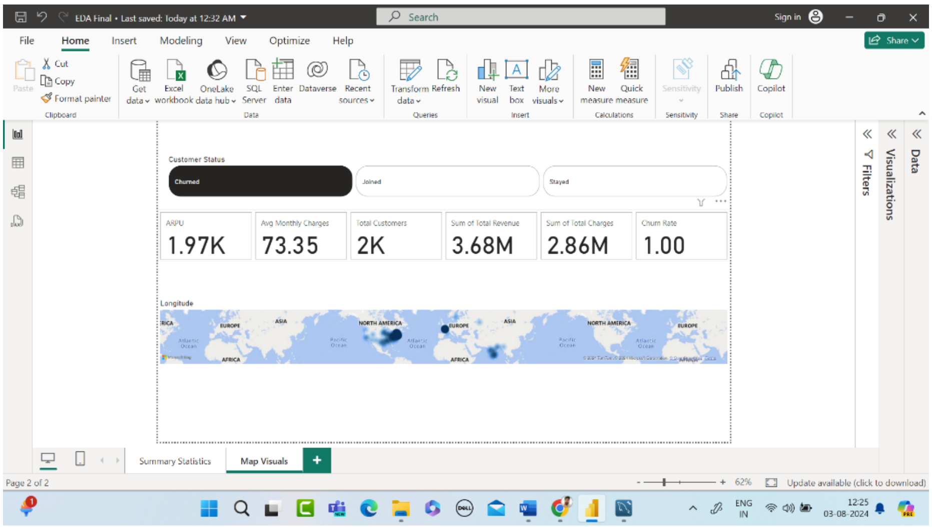
- **Creating Visuals:**
  - Bar Chart: I created a bar chart to visualize the summary statistics.
  - Correlation Heatmap: I used a heatmap to visualize the correlation matrix, making it easier to spot strong and weak correlations.



In Power BI:

- Bar Chart Customization: I customized the bar chart by applying different colors for each statistic to make it more readable.

- Correlation Visualization: The heatmap provided a clear visual representation of how different variables relate to each other.



Customer Status

Churned | Joined | Stayed

ARPU	Avg Monthly Charges	Total Customers	Sum of Total Revenue	Sum of Total Charges	Churn Rate
1.97K	73.35	2K	3.68M	2.86M	1.00

Conclusion

In Phase 1, I successfully prepared the data by converting it into a structured format, loaded it into MySQL for further processing, handled missing values and normalization, and performed an initial exploratory data analysis using Power BI. This set a solid foundation for more detailed analysis and model building in subsequent phases.

Phase 2: Feature Engineering and Data Analysis

In this phase, we will develop specific methods and analyses to address critical business inquiries regarding customer attrition. We will explore the following:

- Identify the total number of customers and the churn rate
- Discover the most common contract types among churned customers
- Create a query to identify the contract types that are most prone to churn
- Calculate the total charges distribution for churned and non-churned customers
- Calculate the average monthly charges for different contract types among churned customers
- Stored Procedure to Calculate Churn Rate
- Stored Procedure to Identify High-Value Customers at Risk of Churning
- Identify the gender distribution among customers who have churned and are on yearly contracts
- Create a view to find the customers with the highest monthly charges in each contract type
- Find the customers who have churned and are not using online services and their average total charges.
- Find the average age of churned customers
- Analyze the distribution of monthly charges among churned customers
- Identify customers with high total charges who have churned

1. Identify the total number of customers and the churn rate

Measures Created in Power BI:

- **Total Customer Calculation:**

Total Customers = COUNT('Customer Demographics'[Customer ID])

- **Total Churned Customers Calculation:**

TotalChurnedCustomers = CALCULATE(COUNT('Account Information'[Customer ID]),

'Account Information'[Customer Status] = "Churned"

- **Churn Rate Calculation:**

ChurnRate = DIVIDE([TotalChurnedCustomers], [Total Customers]) * 100

Explanation:

- The Total Customers measure counts the total number of rows in the Customer Demographics table.
- The TotalChurnedCustomers measure calculates the number of customers with a status of "Churned" in the Account Information table.
- The Churn Rate measure calculates the percentage of churned customers by dividing the number of churned customers by the total number of customers and multiplying by 100 to get the percentage.

2. Discover the most common contract types among churned customers

- **Churned Customers by Contract Type:**

ChurnedCustomersByContractType = CALCULATE(COUNTROWS('Account Information'),

'Customer Demographics'[Customer Status] = "Churned")

- **Churn Rate Calculation:**

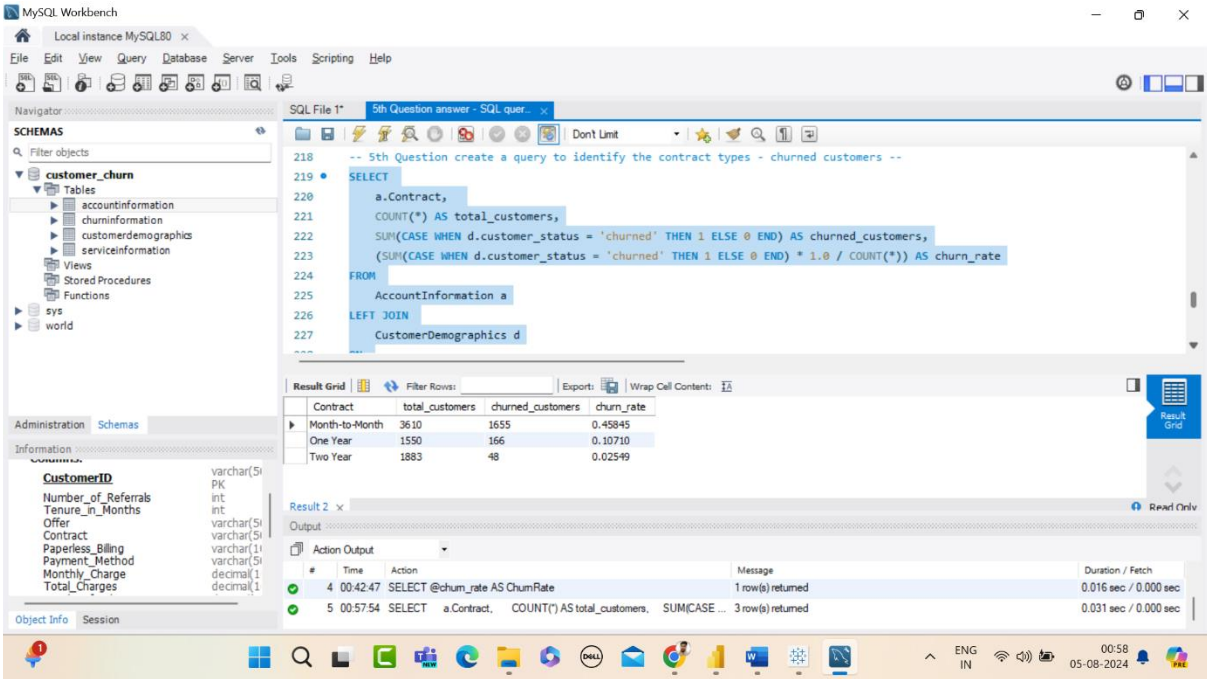
ChurnRate = DIVIDE([TotalChurnedCustomers], [Total Customers]) * 100

Explanation:

- The Churned Customers by Contract Type measure calculates the number of churned customers for each contract type.
- The Churn Rate measure calculates the percentage of churned customers by dividing the number of churned customers by the total number of customers and multiplying by 100 to get the percentage.

3. Create a query to identify the contract types that are most prone to churn

SQL Query



Columns:	
CustomerID	varchar(5 PK
Number_of_Referrals	int
Tenure_in_Months	int
Offer	varchar(5
Contract	varchar(5
Paperless_Billing	varchar(1
Payment_Method	varchar(5
Monthly_Charge	decimal(1
Total_Charges	decimal(1

```sql
-- 5th Question create a query to identify the contract types - churned customers --
SELECT
    a.Contract,
    COUNT(*) AS total_customers,
    SUM(CASE WHEN d.customer_status = 'churned' THEN 1 ELSE 0 END) AS churned_customers,
    (SUM(CASE WHEN d.customer_status = 'churned' THEN 1 ELSE 0 END) * 1.0 / COUNT(*)) AS churn_rate
FROM
    AccountInformation a
LEFT JOIN
    CustomerDemographics d
```

Contract	total_customers	churned_customers	churn_rate
Month-to-Month	3610	1655	0.45845
One Year	1550	166	0.10710
Two Year	1883	48	0.02549

```
#  Time      Action                                                          Message            Duration / Fetch
4  00:42:47  SELECT @churn_rate AS ChurnRate                                 1 row(s) returned  0.016 sec / 0.000 sec
5  00:57:54  SELECT    a.Contract,    COUNT(*) AS total_customers,    SUM(CASE ...  3 row(s) returned  0.031 sec / 0.000 sec
```

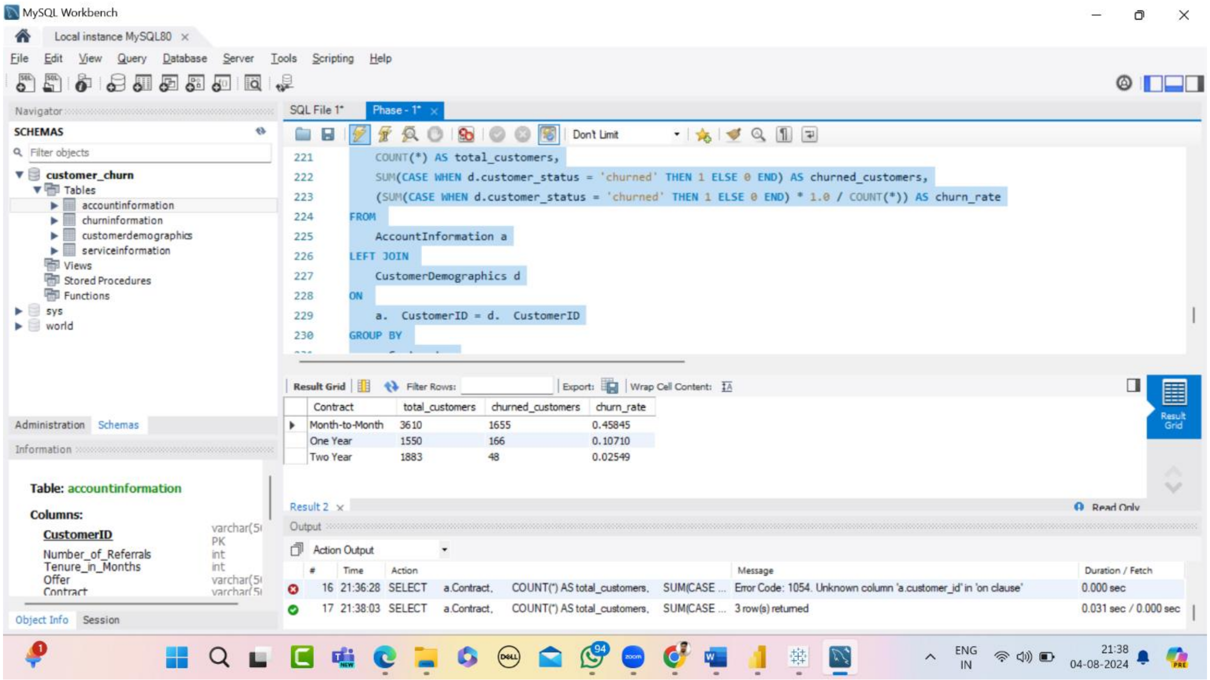


Table: accountinformation

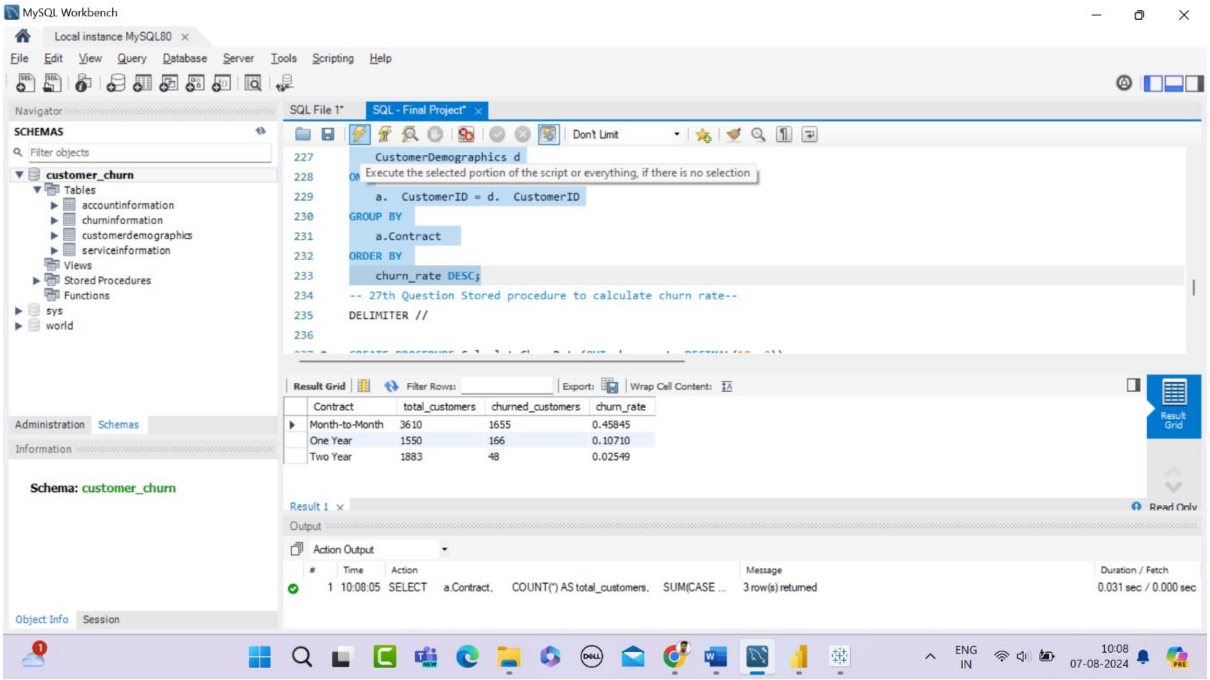
Columns:

CustomerID	varchar(5 PK
Number_of_Referrals	int
Tenure_in_Months	int
Offer	varchar(5
Contract	varchar(5

```
    COUNT(*) AS total_customers,
    SUM(CASE WHEN d.customer_status = 'churned' THEN 1 ELSE 0 END) AS churned_customers,
    (SUM(CASE WHEN d.customer_status = 'churned' THEN 1 ELSE 0 END) * 1.0 / COUNT(*)) AS churn_rate
FROM
    AccountInformation a
LEFT JOIN
    CustomerDemographics d
ON
    a.  CustomerID = d.  CustomerID
GROUP BY
```

Contract	total_customers	churned_customers	churn_rate
Month-to-Month	3610	1655	0.45845
One Year	1550	166	0.10710
Two Year	1883	48	0.02549

```
#   Time      Action                                                      Message                                                          Duration / Fetch
16  21:36:28  SELECT  a.Contract,  COUNT(*) AS total_customers,  SUM(CASE ...  Error Code: 1054. Unknown column 'a.customer_id' in 'on clause'  0.000 sec
17  21:38:03  SELECT  a.Contract,  COUNT(*) AS total_customers,  SUM(CASE ...  3 row(s) returned                                                0.031 sec / 0.000 sec
```



Schema: customer_churn

```
    CustomerDemographics d
ON
    a.  CustomerID = d.  CustomerID
GROUP BY
    a.Contract
ORDER BY
    churn_rate DESC;
-- 27th Question Stored procedure to calculate churn rate--
DELIMITER //
```

Contract	total_customers	churned_customers	churn_rate
Month-to-Month	3610	1655	0.45845
One Year	1550	166	0.10710
Two Year	1883	48	0.02549

```
#  Time      Action                                                      Message            Duration / Fetch
1  10:08:05  SELECT  a.Contract,  COUNT(*) AS total_customers,  SUM(CASE ...  3 row(s) returned  0.031 sec / 0.000 sec
```

Explanation:

- **Selecting the Required Fields:**
  - a.Contract: The type of contract.
  - COUNT(*) AS total_customers: Total number of customers for each contract type.
  - SUM(CASE WHEN d.customer_status = 'churned' THEN 1 ELSE 0 END) AS churned_customers: Total number of churned customers for each contract type.

    - (SUM(CASE WHEN d.customer_status = 'churned' THEN 1 ELSE 0 END) * 1.0 / COUNT(*)) AS churn_rate: Churn rate for each contract type, calculated as the number of churned customers divided by the total number of customers.
- **FROM Clause:**
    - AccountInformation a: Using the AccountInformation table aliased as a.
- **LEFT JOIN Clause:**
    - CustomerDemographics d ON a.CustomerID = d.CustomerID: Joining the AccountInformation table with the CustomerDemographics table on the CustomerID.
- **GROUP BY Clause:**
    - a.Contract: Grouping the results by the contract type.
- **ORDER BY Clause:**
    - churn_rate DESC: Ordering the results by churn rate in descending order.

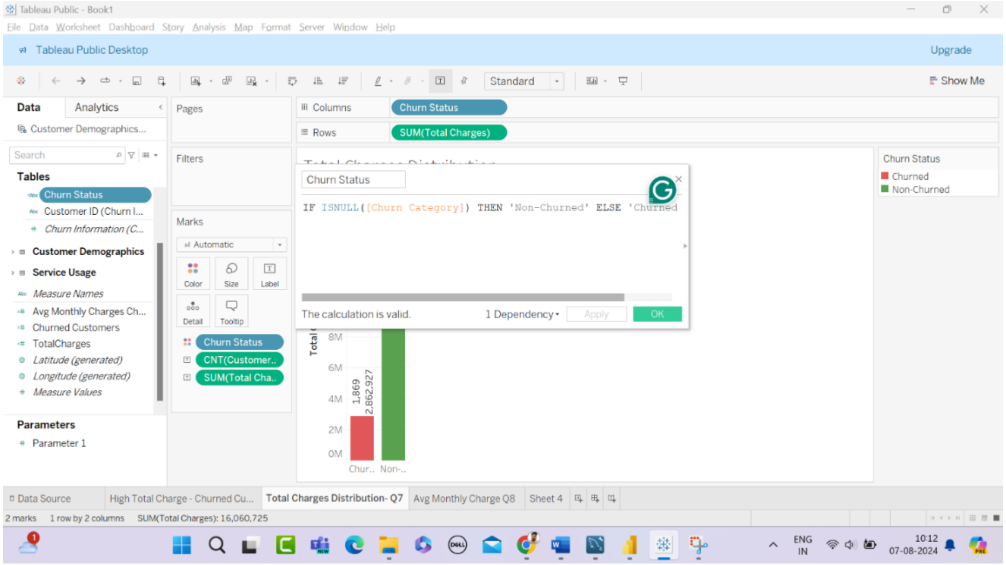
Key Points:

- The LEFT JOIN ensures that all records from AccountInformation are included, even if there are no matching records in CustomerDemographics.
- The CASE statement is used to count churned customers conditionally.
- The 1.0 in the churn rate calculation ensures the division is treated as a floating-point division to get the correct percentage.

By running this query, you will get a list of contract types along with the total number of customers, the number of churned customers, and the churn rate for each contract type, sorted by the churn rate in descending order.

4. Calculate the total charges distribution for churned and non-churned customers

Create a calculated field for Churn Status:



Churn Status

```
IF ISNULL([Churn Category]) THEN 'Non-Churned' ELSE 'Churned
```

The calculation is valid.

2 marks 1 row by 2 columns SUM(Total Charges): 16,060,725

Explanation:

- The calculated field Churn Status distinguishes between churned and non-churned customers.

5. Calculate the average monthly charges for different contract types among churned customers

Measures Created in Power BI:

- **Average Monthly Charges Churned:**

AvgMonthlyChargesChurned = CALCULATE(AVERAGE('Account Information'[Monthly Charge]), 'Account Information'[Customer Status] = "Churned")

- **Churned Customers by Contract Type:**

ChurnedCustomersByContractType = CALCULATE(COUNTROWS('Account Information'),

'Customer Demographics'[Customer Status] = "Churned")

Explanation:

- The Avg Monthly Charges Churned measure calculates the average monthly charges for churned customers.
- The Avg Monthly Charges Churned by Contract Type measure calculates the average monthly charges for churned customers grouped by contract type.

6. Stored Procedure to Calculate Churn Rate

Stored Procedure in MySQL:

Creating a stored procedure to calculate the churn rate in MySQL involves the following steps:

1. **Define the stored procedure**: This includes setting up the parameters and the structure of the procedure.
2. **Calculate the total number of customers**.
3. **Calculate the number of churned customers**.
4. **Calculate the churn rate**.
5. **Return the result**.

Explanation:

1. **DELIMITER** //: Changes the delimiter from ; to // to allow for multi-line procedure definition.
2. **CREATE PROCEDURE CalculateChurnRate**: Defines a new stored procedure named CalculateChurnRate.
3. **OUT churn_rate DECIMAL(10, 2)**: Defines an output parameter churn_rate with a decimal data type.
4. **DECLARE total_customers INT; DECLARE churned_customers INT;**: Declares two integer variables to store the total number of customers and the number of churned customers.

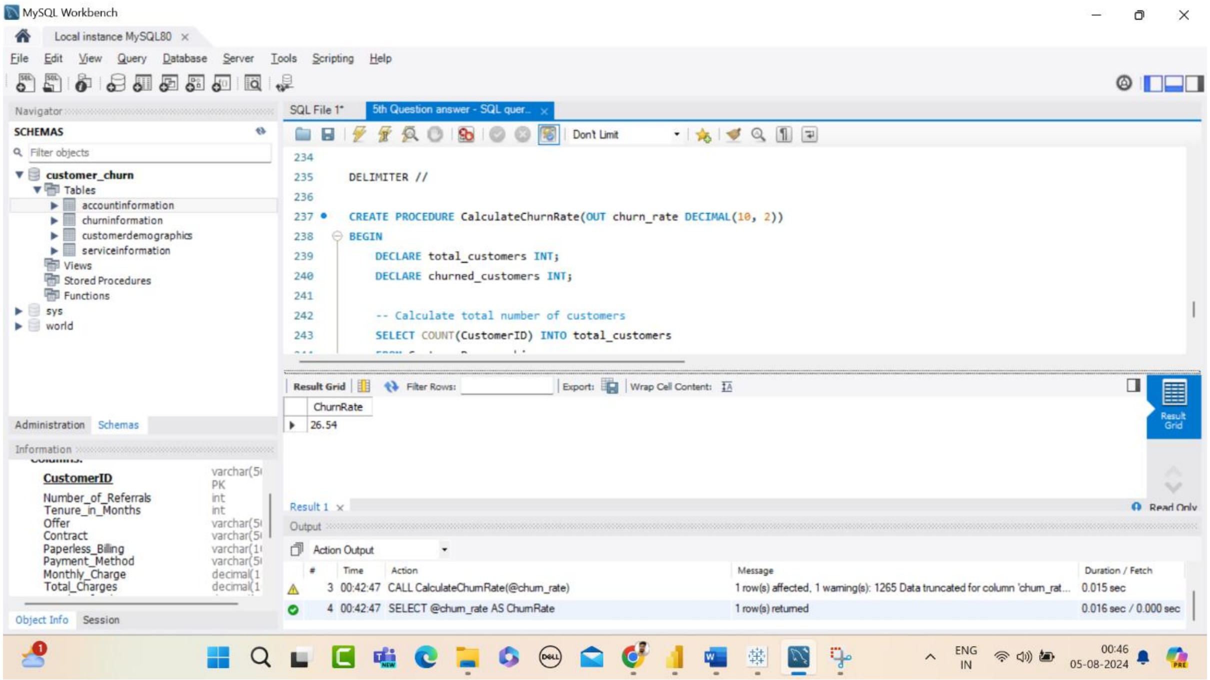
5. **SELECT COUNT(CustomerID) INTO total_customers FROM CustomerDemographics;**: Calculates the total number of customers and stores the result in total_customers.

6. **SELECT COUNT(CustomerID) INTO churned_customers FROM CustomerDemographics WHERE Customer_Status = 'Churned';**: Calculates the number of churned customers and stores the result in churned_customers.

7. **SET churn_rate = (churned_customers / total_customers) * 100;**: Calculates the churn rate as a percentage.

8. **END**: Ends the procedure definition.

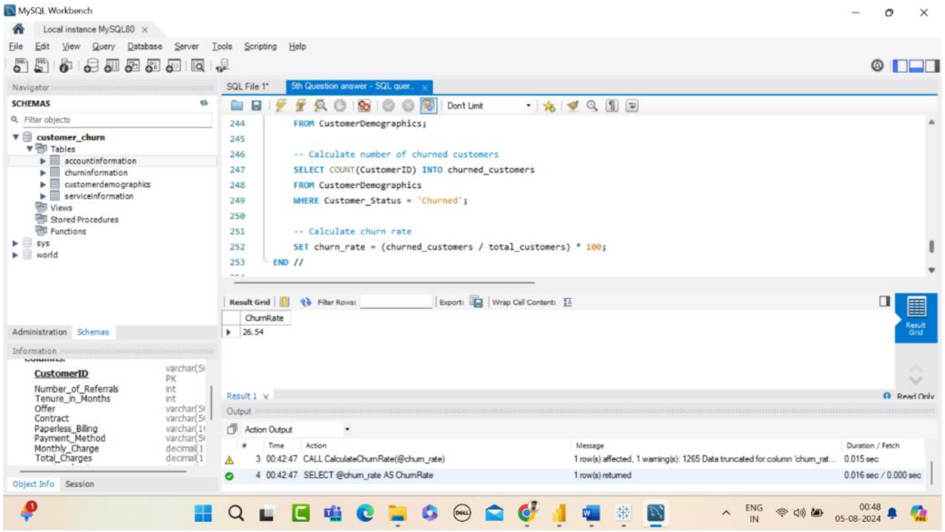
9. **DELIMITER ;**: Resets the delimiter back to ;.



```
DELIMITER //

CREATE PROCEDURE CalculateChurnRate(OUT churn_rate DECIMAL(10, 2))
BEGIN
    DECLARE total_customers INT;
    DECLARE churned_customers INT;

    -- Calculate total number of customers
    SELECT COUNT(CustomerID) INTO total_customers
```

ChurnRate
26.54

```
#  Time      Action                                 Message                                                                   Duration / Fetch
3  00:42:47  CALL CalculateChurnRate(@churn_rate)   1 row(s) affected, 1 warning(s): 1265 Data truncated for column 'churn_rat...   0.015 sec
4  00:42:47  SELECT @churn_rate AS ChurnRate        1 row(s) returned                                                         0.016 sec / 0.000 sec
```



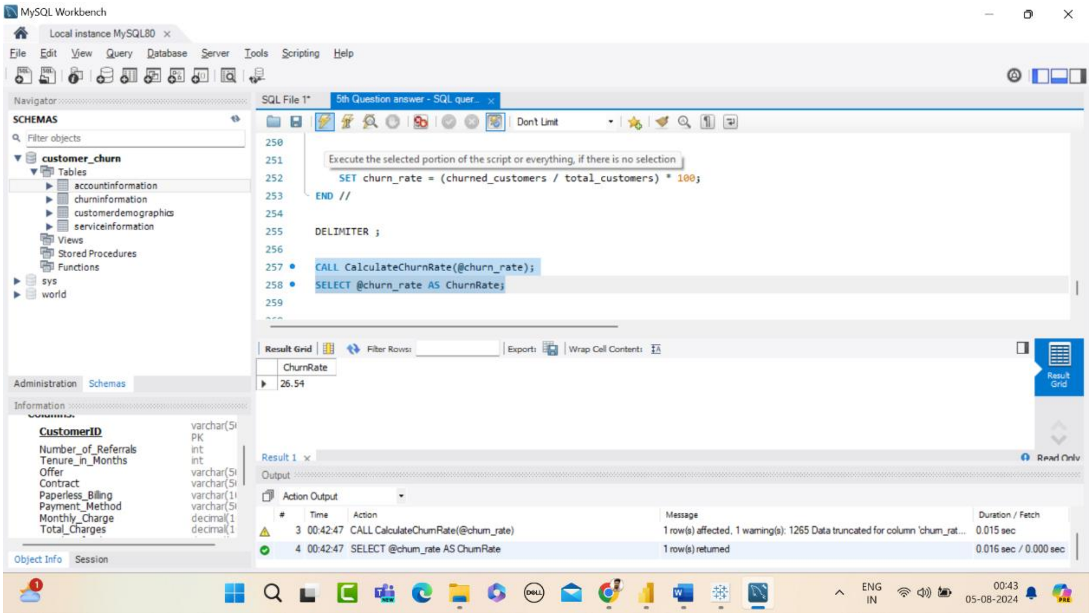
```
    FROM CustomerDemographics;

    -- Calculate number of churned customers
    SELECT COUNT(CustomerID) INTO churned_customers
    FROM CustomerDemographics
    WHERE Customer_Status = 'Churned';

    -- Calculate churn rate
    SET churn_rate = (churned_customers / total_customers) * 100;
END //
```

ChurnRate
26.54

```
#  Time      Action                                 Message                                                                   Duration / Fetch
3  00:42:47  CALL CalculateChurnRate(@churn_rate)   1 row(s) affected, 1 warning(s): 1265 Data truncated for column 'churn_rat...   0.015 sec
4  00:42:47  SELECT @churn_rate AS ChurnRate        1 row(s) returned                                                         0.016 sec / 0.000 sec
```

How to Call the Stored Procedure:

To call this stored procedure and get the churn rate, you can use the following SQL:



```
    SET churn_rate = (churned_customers / total_customers) * 100;
END //

DELIMITER ;

CALL CalculateChurnRate(@churn_rate);
SELECT @churn_rate AS ChurnRate;
```

```
ChurnRate
26.54
```

```
#  Time      Action                                Message                                                                  Duration / Fetch
3  00:42:47  CALL CalculateChurnRate(@churn_rate)  1 row(s) affected, 1 warning(s): 1265 Data truncated for column 'churn_rat...  0.015 sec
4  00:42:47  SELECT @churn_rate AS ChurnRate       1 row(s) returned                                                        0.016 sec / 0.000 sec
```

7. Stored Procedure to Identify High-Value Customers at Risk of Churning

Stored Procedure in MySQL:

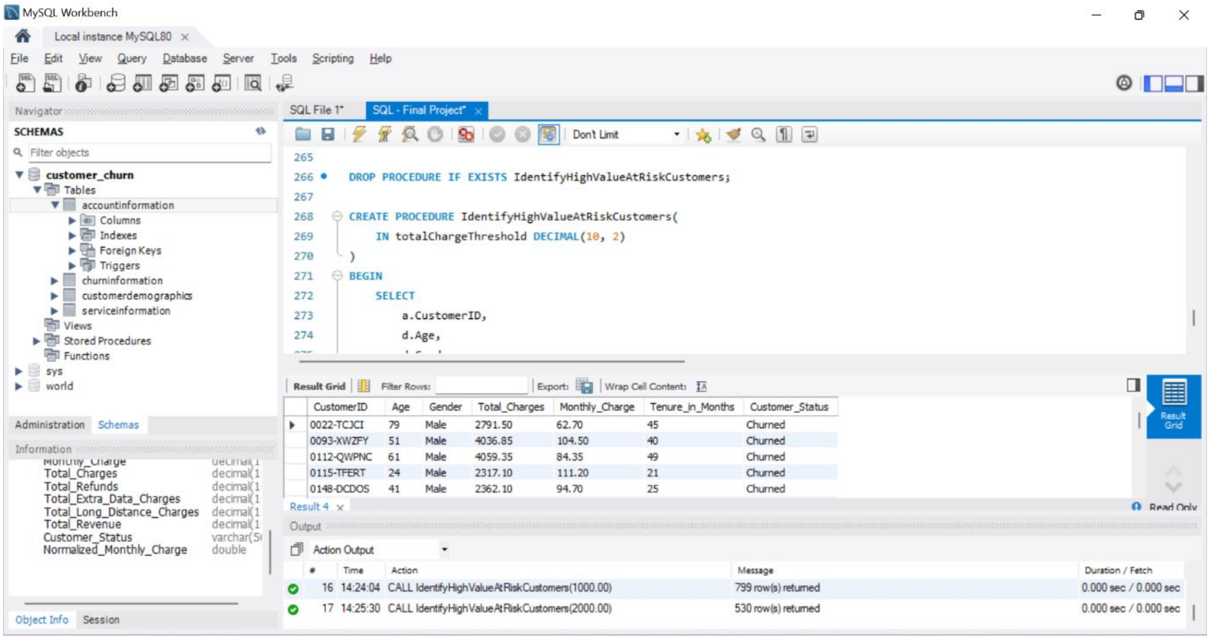
The following steps outline the creation and execution of a stored procedure in MySQL to identify high-value customers at risk of churning. The stored procedure filters customers based on a specified threshold for total charges and recognises those who have churned.

Explanation of the Stored Procedure

1. **DELIMITER** //:
   - Changes the default statement delimiter (semicolon) to // to allow for the creation of the stored procedure without interference from the semicolon in the procedure body.
2. **DROP PROCEDURE IF EXISTS IdentifyHighValueAtRiskCustomers**:
   - This command ensures that if a procedure with the same name already exists, it is dropped before creating a new one.
3. **CREATE PROCEDURE IdentifyHighValueAtRiskCustomers(IN totalChargeThreshold DECIMAL(10, 2))**:
   - **CREATE PROCEDURE**: Defines a new stored procedure named IdentifyHighValueAtRiskCustomers.
   - **IN totalChargeThreshold DECIMAL(10, 2)**: Declares an input parameter named totalChargeThreshold of type DECIMAL(10, 2) to specify the threshold for total charges.
4. **BEGIN ... END**:
   - Encloses the body of the stored procedure.
5. **SELECT Statement**:

   - **SELECT a.CustomerID, d.Age, d.Gender, a.Total_Charges, a.Monthly_Charge, a.Tenure_in_Months, a.Customer_Status**:
      - Specifies the columns to retrieve from the AccountInformation and CustomerDemographics tables.
   - **FROM AccountInformation a JOIN CustomerDemographics d ON a.CustomerID = d.CustomerID**:
      - Performs an inner join between the AccountInformation (aliased as a) and CustomerDemographics (aliased as d) tables based on the CustomerID column.
   - **WHERE a.Total_Charges > totalChargeThreshold AND a.Customer_Status = 'churned'**:
      - Filters the results to include only customers with total charges greater than the specified threshold and a customer status of 'churned'.

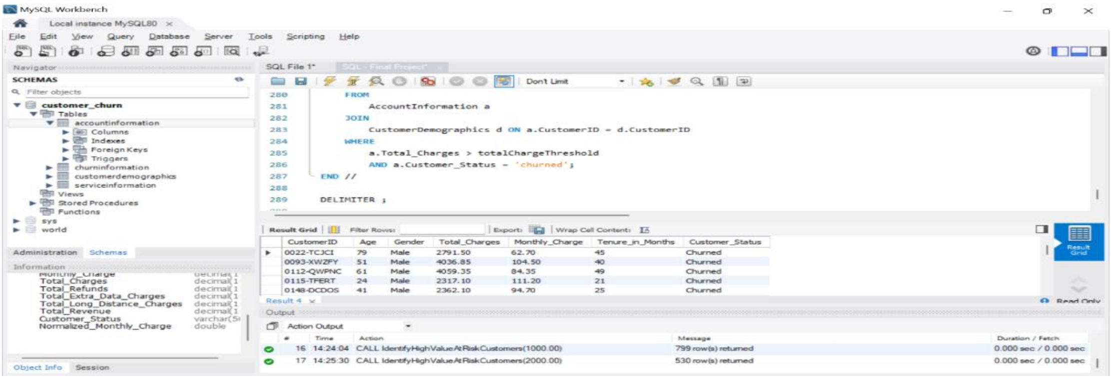
6. **END //**:
   - Ends the body of the stored procedure.
   - The // delimiter marks the end of the procedure definition.
7. **DELIMITER ;**:
   - Resets the statement delimiter back to the default semicolon.
8. **CALL IdentifyHighValueAtRiskCustomers(2000.00);**:
   - Executes the stored procedure with a totalChargeThreshold value of 2000.00, retrieving customers who have total charges above this threshold and have churned.



```
DROP PROCEDURE IF EXISTS IdentifyHighValueAtRiskCustomers;

CREATE PROCEDURE IdentifyHighValueAtRiskCustomers(
    IN totalChargeThreshold DECIMAL(10, 2)
)
BEGIN
    SELECT
        a.CustomerID,
        d.Age,
```

CustomerID	Age	Gender	Total_Charges	Monthly_Charge	Tenure_in_Months	Customer_Status
0022-TCJCI	79	Male	2791.50	62.70	45	Churned
0093-XWZFY	51	Male	4036.85	104.50	40	Churned
0112-QWPNC	61	Male	4059.35	84.35	49	Churned
0115-TFERT	24	Male	2317.10	111.20	21	Churned
0148-DCDOS	41	Male	2362.10	94.70	25	Churned

#	Time	Action	Message	Duration / Fetch
16	14:24:04	CALL IdentifyHighValueAtRiskCustomers(1000.00)	799 row(s) returned	0.000 sec / 0.000 sec
17	14:25:30	CALL IdentifyHighValueAtRiskCustomers(2000.00)	530 row(s) returned	0.000 sec / 0.000 sec



```
        FROM
            AccountInformation a
        JOIN
            CustomerDemographics d ON a.CustomerID = d.CustomerID
        WHERE
            a.Total_Charges > totalChargeThreshold
            AND a.Customer_Status = 'churned';
    END //

    DELIMITER ;
```

CustomerID	Age	Gender	Total_Charges	Monthly_Charge	Tenure_in_Months	Customer_Status
0022-TCJCI	79	Male	2791.50	62.70	45	Churned
0093-XWZFY	51	Male	4036.85	104.50	40	Churned
0112-QWPNC	61	Male	4059.35	84.35	49	Churned
0115-TFERT	24	Male	2317.10	111.20	21	Churned
0148-DCDOS	41	Male	2362.10	94.70	25	Churned

#	Time	Action	Message	Duration / Fetch
16	14:24:04	CALL IdentifyHighValueAtRiskCustomers(1000.00)	799 row(s) returned	0.000 sec / 0.000 sec
17	14:25:30	CALL IdentifyHighValueAtRiskCustomers(2000.00)	530 row(s) returned	0.000 sec / 0.000 sec

How to Modify the Threshold Limit

If you want to change the threshold limit after the stored procedure has been created, you can simply call the stored procedure with a different value for the totalChargeThreshold parameter:

CALL IdentifyHighValueAtRiskCustomers(3000.00);

This call will now filter customers with total charges above 3000.00 and who have churned. You do not need to modify the stored procedure itself; just adjust the parameter value when calling it.

8. Identify the gender distribution among customers who have churned and are on yearly contracts

To identify the gender distribution among customers who have churned and are on yearly contracts in Tableau, follow these steps:

Steps and Explanation:

Filter Churned Customers on Yearly Contracts:

- **Description:**
  - Drag CustomerStatus to the Filters shelf and select the status that represents churned customers (e.g., "Churned").
  - Drag Contract to the Filters shelf and select the contract type that represents yearly contracts (e.g., "one year" and “two years”).
  - **Description:** Drag COUNT(CustomerID) to Label on the Marks card to display the count of customers directly on the bars.
  - Drag Gender to Color on the Marks card to differentiate genders by color.

Summary:

The use of filters, customizations, and labels ensures that the visualization is focused, clear, and informative, aiding in effective data analysis and decision-making.

9. Create a view to find the customers with the highest monthly charges in each contract type

For this question, - I used contract type, monthly charge, customer ID for the count, and For more visual appeal used card and slice

10. Find the customers who have churned and are not using online services, and their average total charges.

For this question, I used the filter – customer status “churned” and Internet service “No”

Avg total charges of churned customers measure found. For more clarity added gender wise classification in the label.

11.Find the average age of churned customers

AvgAgeChurnedCustomers = CALCULATE(AVERAGE('Customer Demographics'[Age]), 'Account Information'[Customer Status] = "Churned")

Explanation: This measure calculates the average age for customers whose status is "Churned".

12.Analyze the distribution of monthly charges among churned customers

Monthly Charge field to create bins. – created bin group

Filter for Churned Customers:

- Apply a filter to include only churned customers.
- Explanation: Filtering ensures the analysis is specific to churned customers.

13.Identify customers with high total charges who have churned

Found Churned Total Charges measure, filter churned customer,

ChurnedTotalCharges = CALCULATE(SUM('Account Information'[Total Charges]), 'Account Information'[Customer Status] = "Churned")

Conclusion:

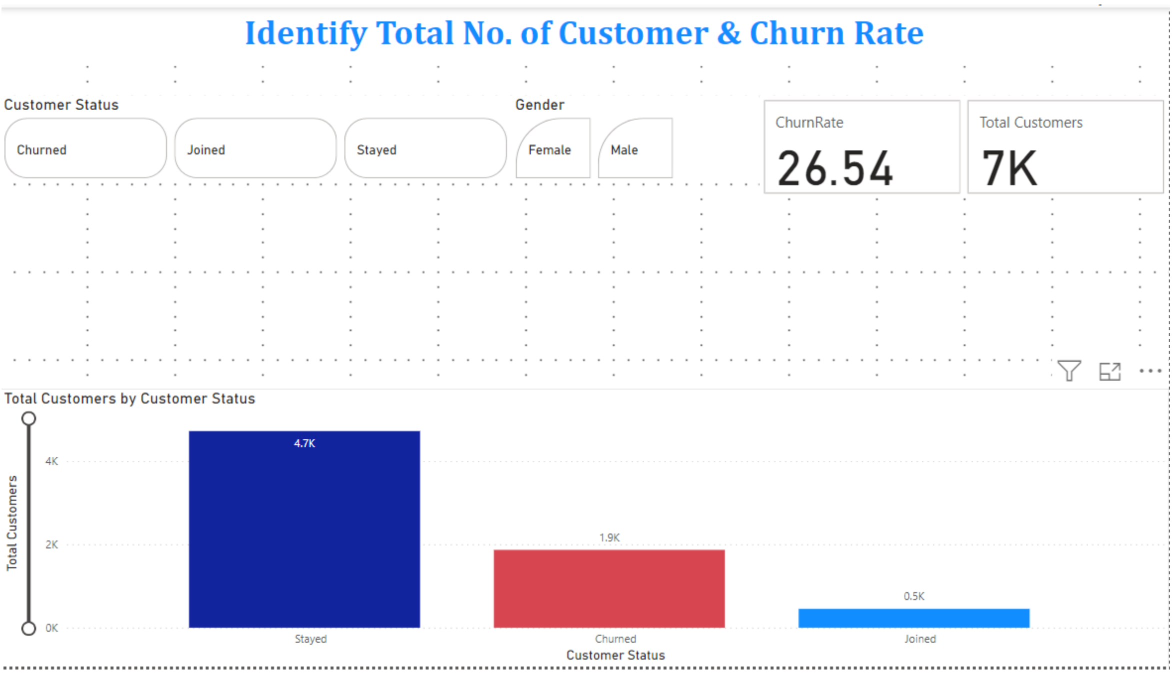
By following these methods, businesses can effectively analyze customer churn data, identify key insights, and develop strategies to improve customer retention.

Phase 3: Visualization and Dashboard Creation

In this phase, I created visualizations and dashboards in Power BI and Tableau to provide insights into customer churn.

Power BI Visualizations

1. **Identify the Total Number of Customers and the Churn Rate**



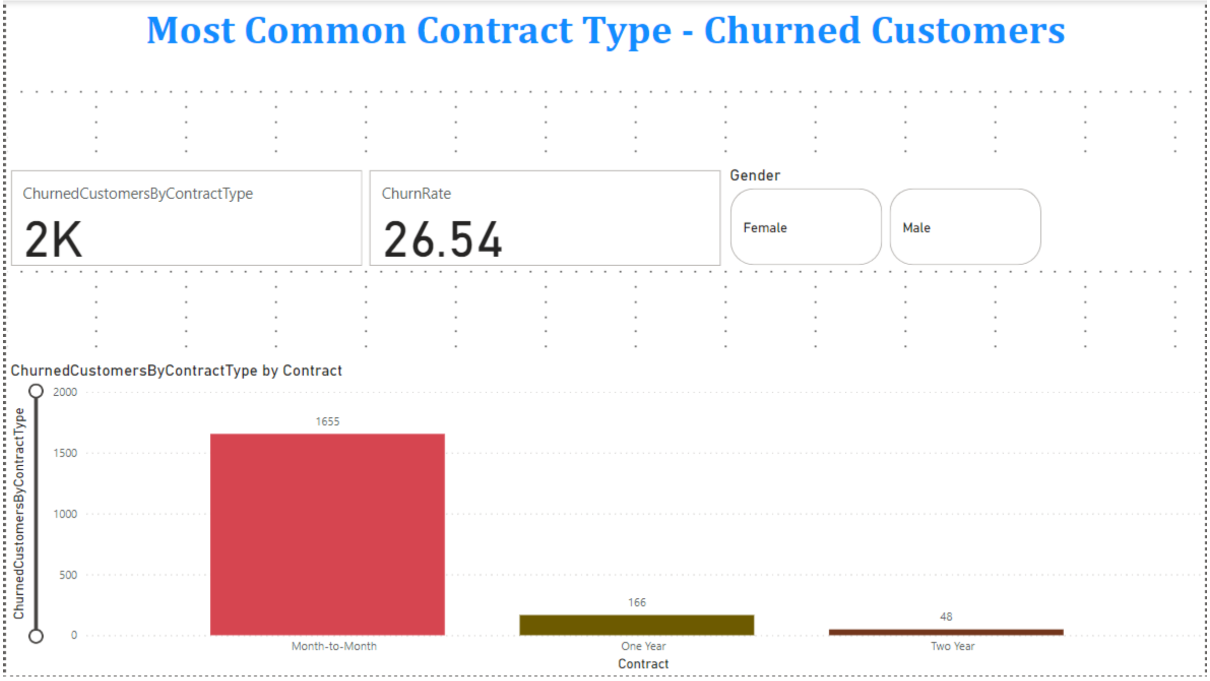
Identify Total No. of Customer & Churn Rate

- From this visual, we can understand the total no. of customers, churn rate.

- X-axis Customer status
- Y-axis Total Customers

- Added cards, and slicers for a more appealing visual

2. Discover the Most Common Contract Types Among Churned Customers


Most Common Contract Type - Churned Customers
ChurnedCustomersByContractType
2K
ChurnRate
26.54
Gender
Female
Male
ChurnedCustomersByContractType by Contract
ChurnedCustomersByContractType
2000
1500
1000
500
0
1655
166
48
Month-to-Month
One Year
Two Year
Contract

- X-axis Contract
- Y-axis Churned Customers by Contract Type

- For better clarity, added a card visual and a slice visual.

SQL Visualization:

SQL Visualization and Analysis

In this section, we have addressed three key business questions using SQL. The focus is on identifying contract types prone to churn, calculating the churn rate, and identifying high-value customers at risk of churning.

3. Create a Query to Identify the Contract Types That Are Most Prone to Churn

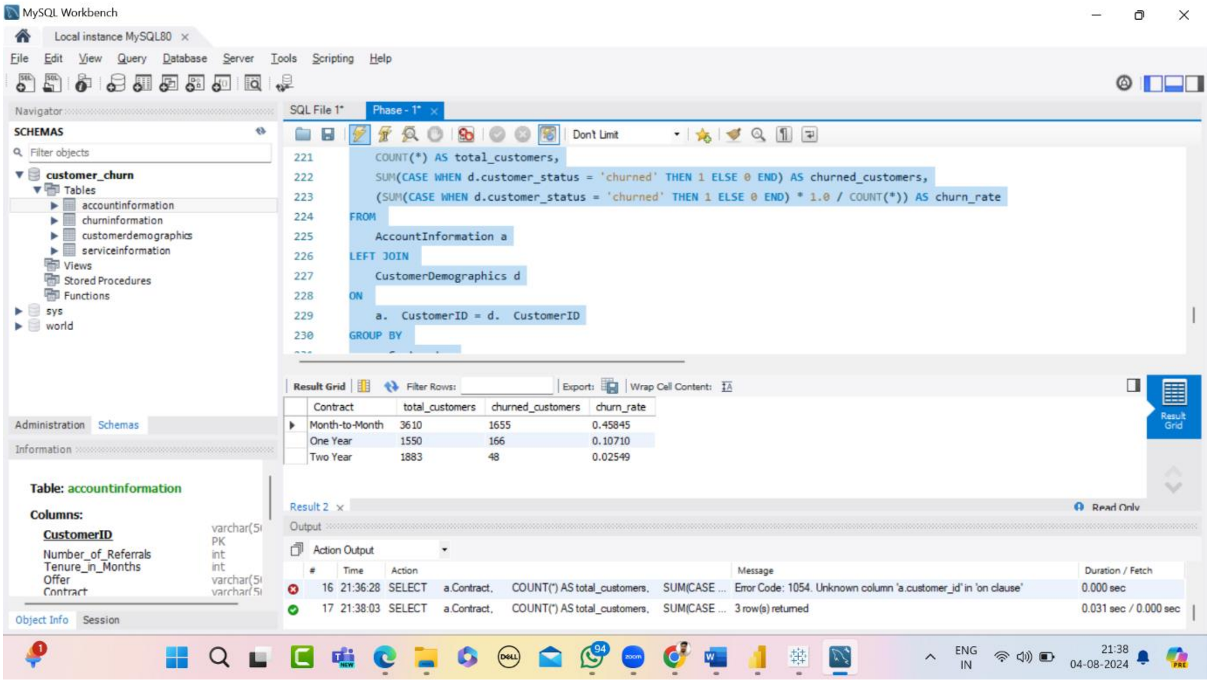


Table: accountinformation

Columns:

CustomerID	varchar(5 PK
Number_of_Referrals	int
Tenure_in_Months	int
Offer	varchar(5
Contract	varchar(5

```sql
    COUNT(*) AS total_customers,
    SUM(CASE WHEN d.customer_status = 'churned' THEN 1 ELSE 0 END) AS churned_customers,
    (SUM(CASE WHEN d.customer_status = 'churned' THEN 1 ELSE 0 END) * 1.0 / COUNT(*)) AS churn_rate
FROM
    AccountInformation a
LEFT JOIN
    CustomerDemographics d
ON
    a.  CustomerID = d.  CustomerID
GROUP BY
```

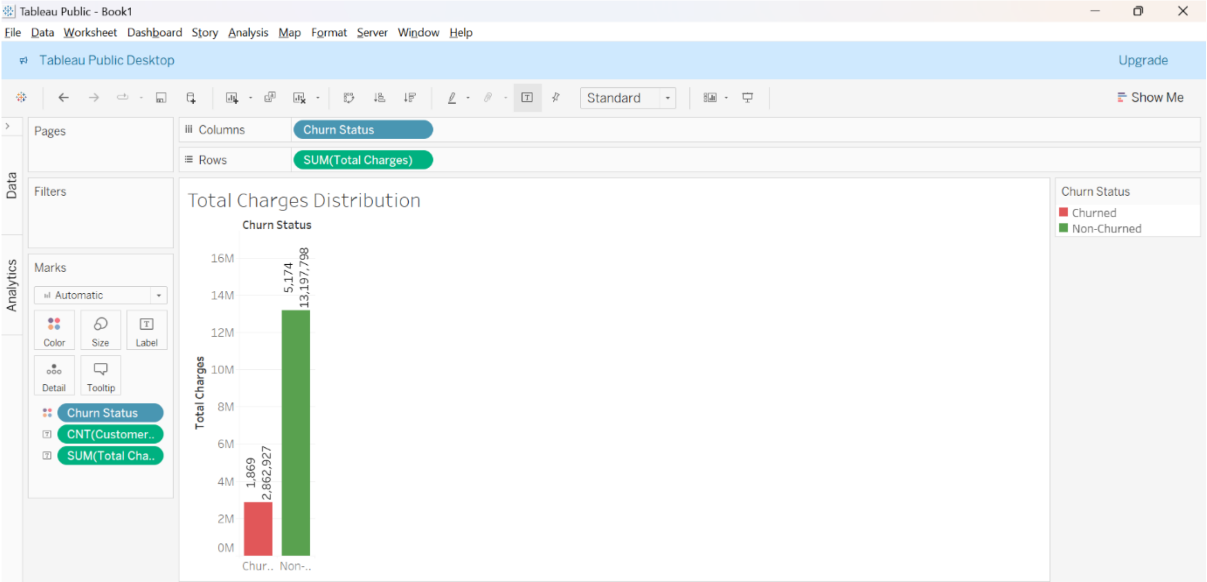
Contract	total_customers	churned_customers	churn_rate
Month-to-Month	3610	1655	0.45845
One Year	1550	166	0.10710
Two Year	1883	48	0.02549

```
#   Time      Action                                                   Message                                                      Duration / Fetch
16  21:36:28  SELECT  a.Contract,  COUNT(*) AS total_customers,  SUM(CASE ...  Error Code: 1054. Unknown column 'a.customer_id' in 'on clause'  0.000 sec
17  21:38:03  SELECT  a.Contract,  COUNT(*) AS total_customers,  SUM(CASE ...  3 row(s) returned                                            0.031 sec / 0.000 sec
```

This query provides a list of contract types along with the total number of customers, the number of churned customers, and the churn rate for each contract type, sorted by the churn rate in descending order.

Tableau Visualizations

4.Total Charges Distribution for Churned and Non-Churned Customers



Columns: Churn Status

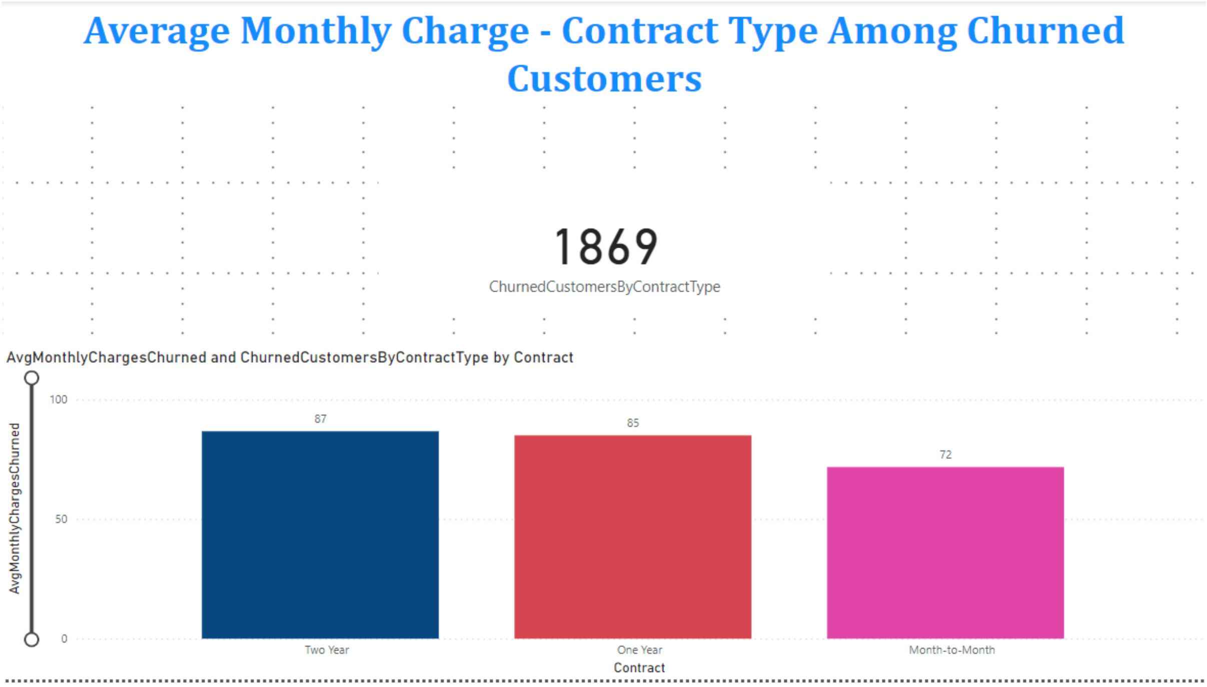
Rows: SUM(Total Charges)

Marks: Churn Status, CNT(Customer.., SUM(Total Cha..

- Created a field to distinguish between churned and non-churned customers.
- From the bar visual, total charges distribution for both churned and non-churned customers.
- X-axis Churn Status
- Y-axis Total Charges

Power BI Visualization:

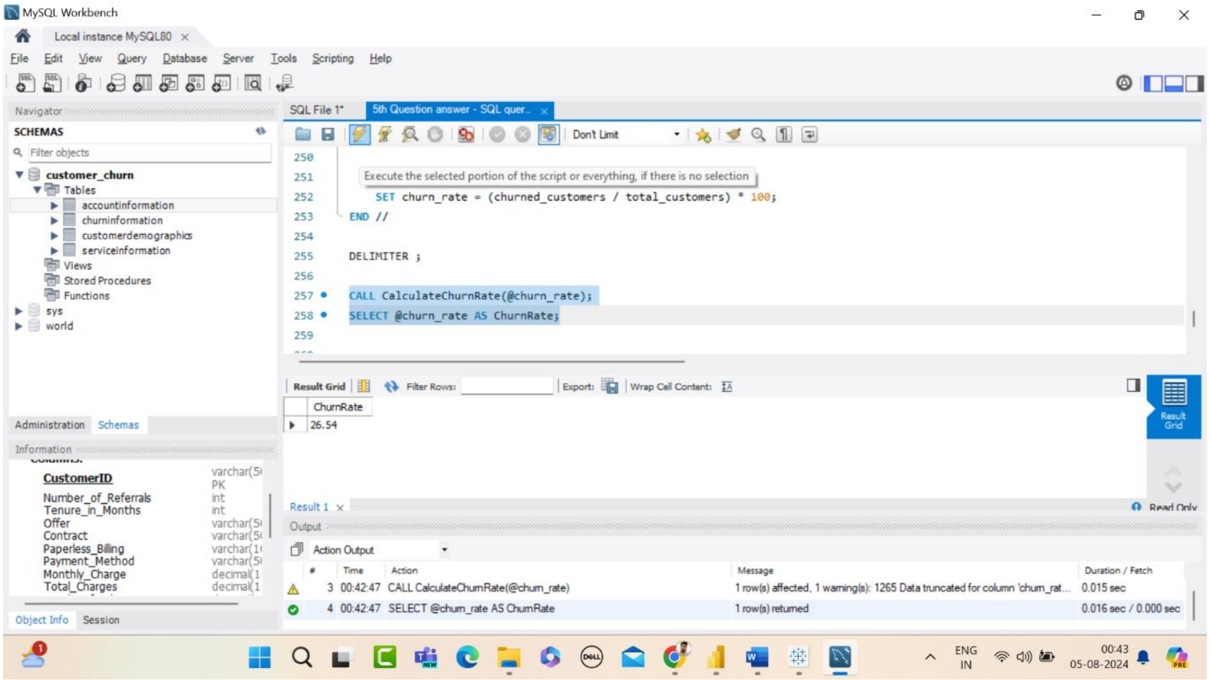
5. Calculate the Average Monthly Charges for Different Contract Types Among Churned Customers


Average Monthly Charge - Contract Type Among Churned Customers
1869
ChurnedCustomersByContractType
AvgMonthlyChargesChurned and ChurnedCustomersByContractType by Contract
AvgMonthlyChargesChurned
100
50
0
87
85
72
Two Year
One Year
Month-to-Month
Contract

- For this visual avg. monthly charges churned measure created and churned customers by contract type measure used.
- X-axis Contract
- Y-axis Avg. Monthly Charges Churned
- Churned Customer by Contract type used for clear visual.

SQL Visualization:

6. Stored Procedure to Calculate Churn Rate



CustomerID	varchar(5 PK
Number_of_Referrals	int
Tenure_in_Months	int
Offer	varchar(5
Contract	varchar(5
Paperless_Billing	varchar(1
Payment_Method	varchar(5
Monthly_Charge	decimal(1
Total_Charges	decimal(1

```

    SET churn_rate = (churned_customers / total_customers) * 100;
END //

DELIMITER ;

CALL CalculateChurnRate(@churn_rate);
SELECT @churn_rate AS ChurnRate;

```

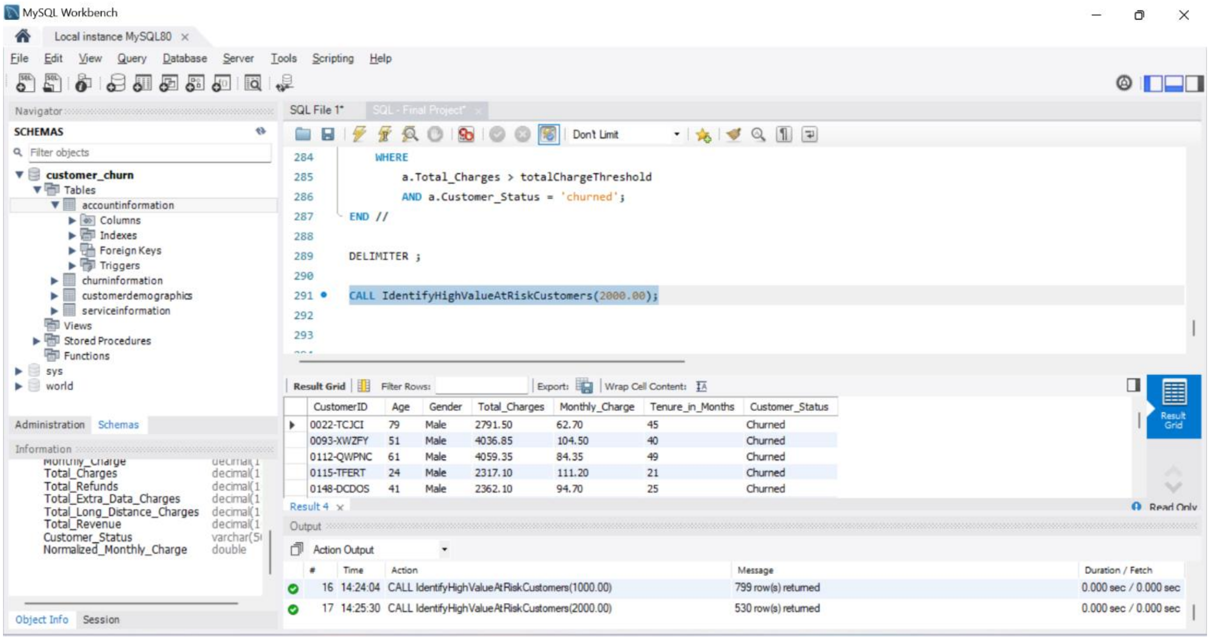
ChurnRate
26.54

```
   #  Time      Action                                 Message                                                                 Duration / Fetch
⚠  3  00:42:47  CALL CalculateChurnRate(@churn_rate)   1 row(s) affected, 1 warning(s): 1265 Data truncated for column 'churn_rat...   0.015 sec
✓  4  00:42:47  SELECT @churn_rate AS ChurnRate        1 row(s) returned                                                       0.016 sec / 0.000 sec
```

- This stored procedure calculates the churn rate and outputs it as a percentage.

SQL Visualization:

7. Stored Procedure to Identify High-Value Customers at Risk of Churning



```sql
        WHERE
            a.Total_Charges > totalChargeThreshold
            AND a.Customer_Status = 'churned';
END //

DELIMITER ;

CALL IdentifyHighValueAtRiskCustomers(2000.00);
```

CustomerID	Age	Gender	Total_Charges	Monthly_Charge	Tenure_in_Months	Customer_Status
0022-TCJCI	79	Male	2791.50	62.70	45	Churned
0093-XWZFY	51	Male	4036.85	104.50	40	Churned
0112-QWPNC	61	Male	4059.35	84.35	49	Churned
0115-TFERT	24	Male	2317.10	111.20	21	Churned
0148-DCDOS	41	Male	2362.10	94.70	25	Churned

```
#   Time      Action                                          Message              Duration / Fetch
16  14:24:04  CALL IdentifyHighValueAtRiskCustomers(1000.00)  799 row(s) returned  0.000 sec / 0.000 sec
17  14:25:30  CALL IdentifyHighValueAtRiskCustomers(2000.00)  530 row(s) returned  0.000 sec / 0.000 sec
```

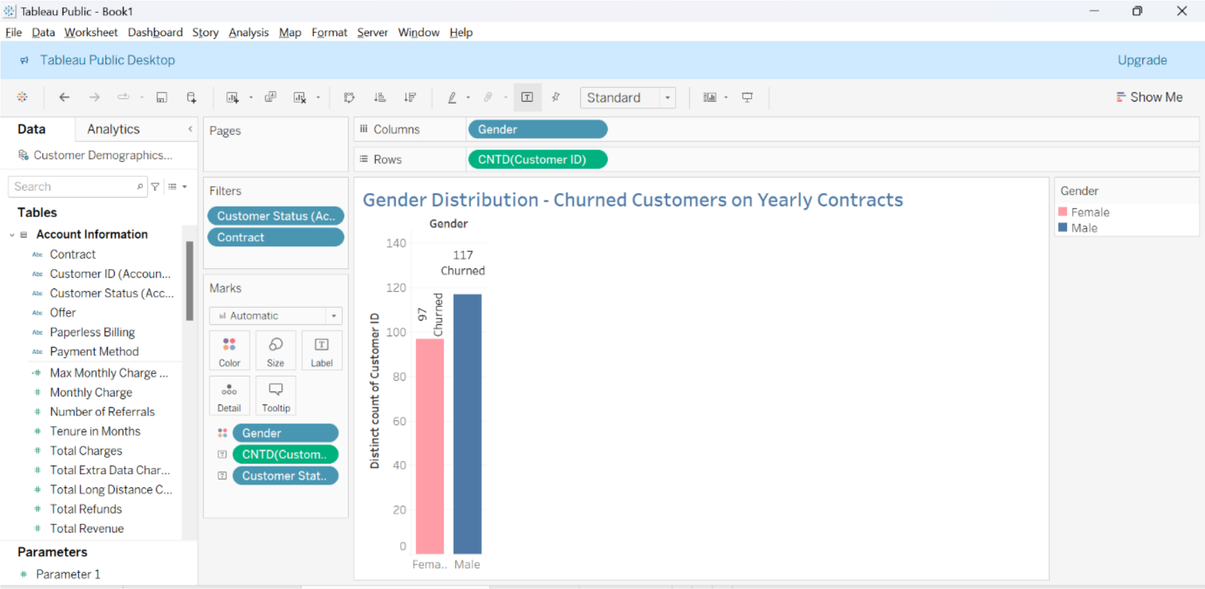
This stored procedure identifies high-value customers who have churned and had total charges above a specified threshold.

Summary:

By using these queries and stored procedures, we can gain valuable insights into customer behavior, identify at-risk customers, and develop strategies to reduce churn.

Tableau Visualization:

8. Gender Distribution Among Customers Who Have Churned and Are on Yearly Contracts



Columns: Gender

Rows: CNTD(Customer ID)

Filters: Customer Status (Ac.., Contract

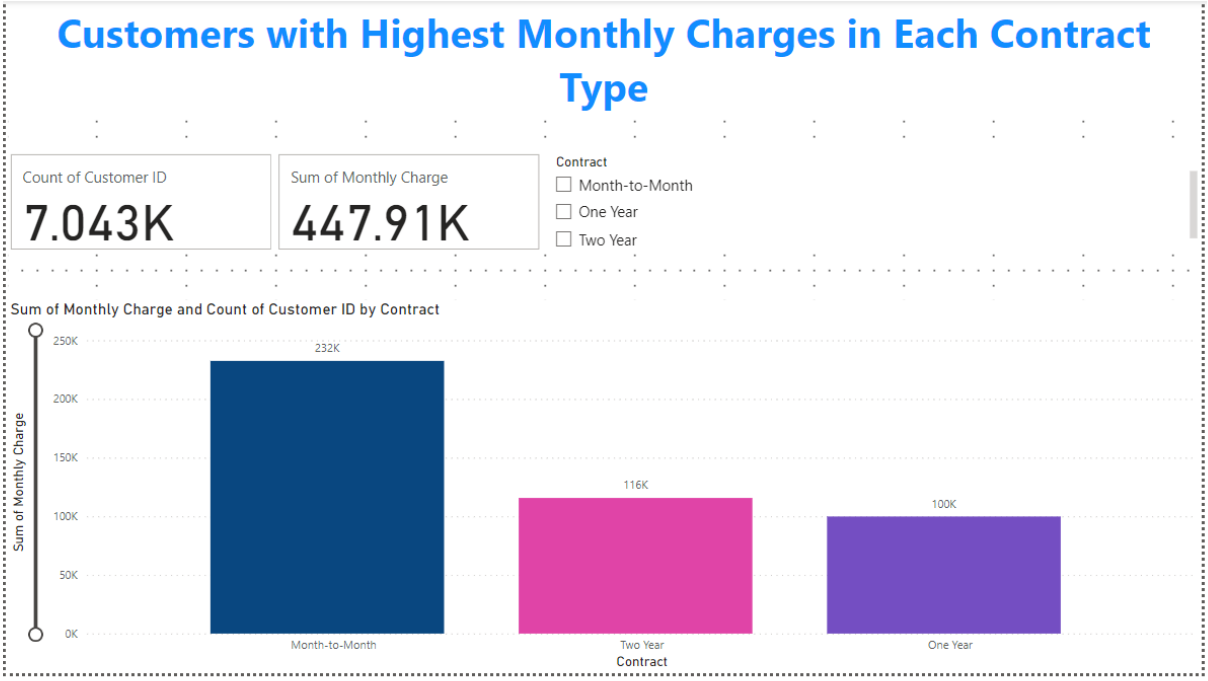
Marks: Gender, CNTD(Custom.., Customer Stat..

Gender Distribution - Churned Customers on Yearly Contracts

- From this visual, we can understand the gender distribution among churned customers on yearly contracts.
- Selected both 1 year and 2 year contract customers

Power BI Visualization:

9. Customers with the Highest Monthly Charges in Each Contract Type



Customers with Highest Monthly Charges in Each Contract Type

Count of Customer ID
7.043K

Sum of Monthly Charge
447.91K

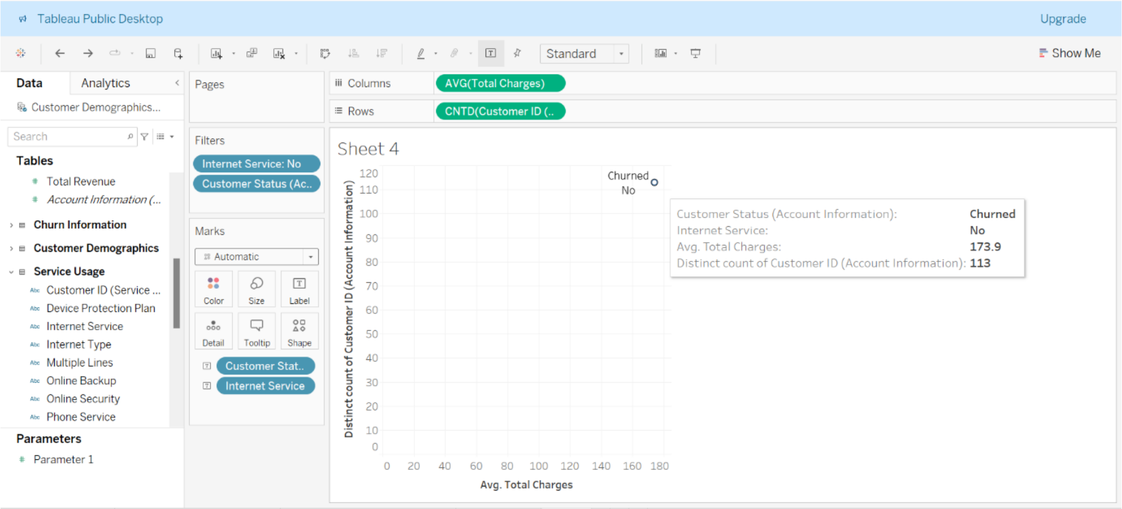
Contract
□ Month-to-Month
□ One Year
□ Two Year

- X-axis Contract
- Y-Monthly Charge

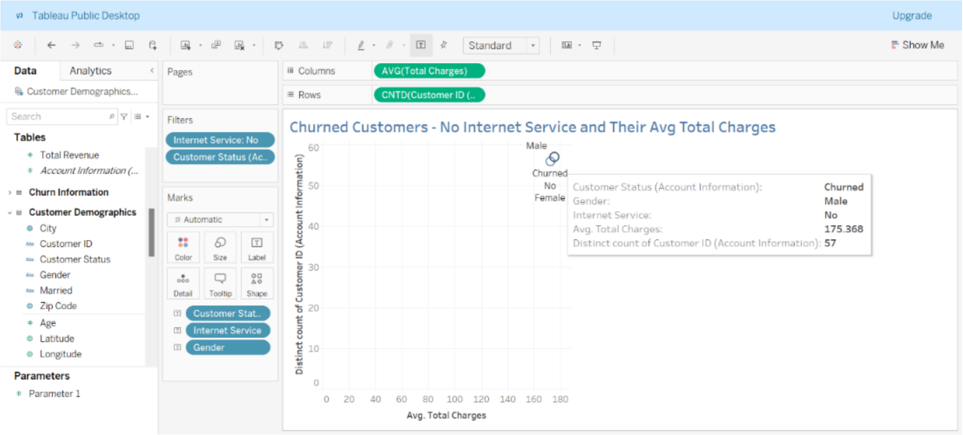
- For more clarity, added a card visual – total customers, monthly charge. Slicer used for contract type.

Tableau Visualization:

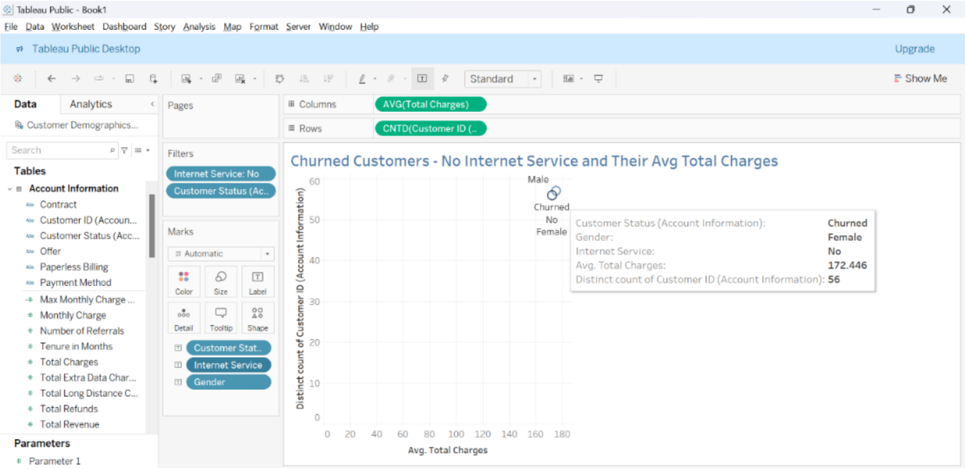
10. Customers Who Have Churned and Are Not Using Online Services, and Their Average Total Charges



Customer Status (Account Information): Churned
Internet Service: No
Avg. Total Charges: 173.9
Distinct count of Customer ID (Account Information): 113



Customer Status (Account Information): Churned
Gender: Male
Internet Service: No
Avg. Total Charges: 175.368
Distinct count of Customer ID (Account Information): 57

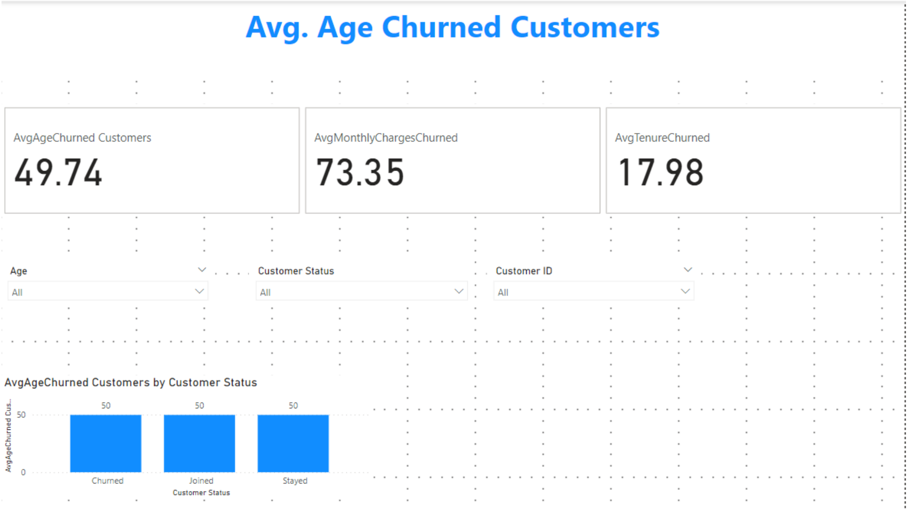


Customer Status (Account Information): Churned
Gender: Female
Internet Service: No
Avg. Total Charges: 172.446
Distinct count of Customer ID (Account Information): 56

- From the visuals, among churned customers who do not have internet service and average. total charges.

Power BI Visualizations:

11. Find the Average Age of Churned Customers



Avg. Age Churned Customers

AvgAgeChurned Customers 49.74

AvgMonthlyChargesChurned 73.35

AvgTenureChurned 17.98

Age: All

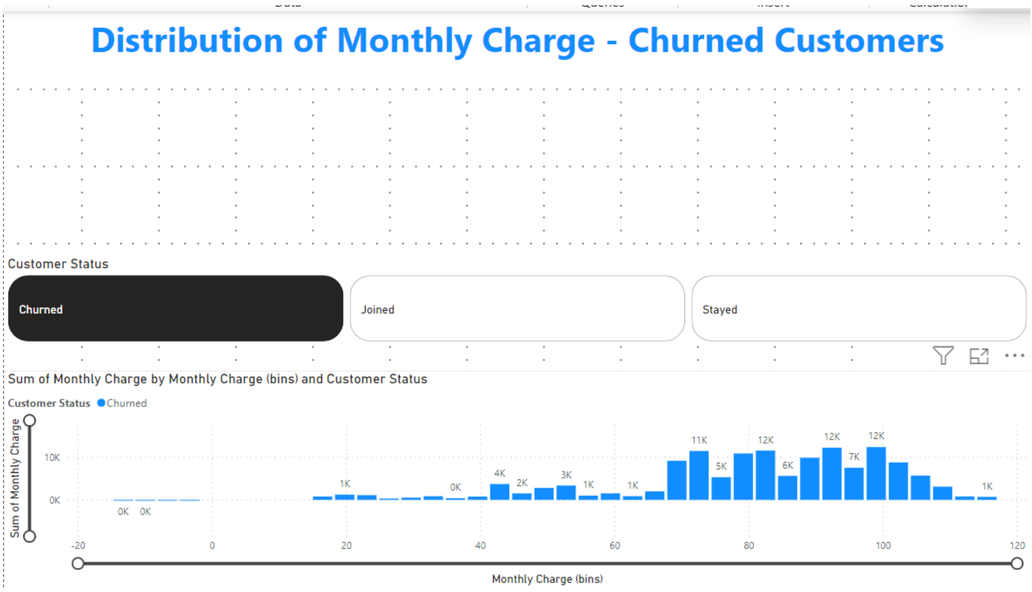
Customer Status: All

Customer ID: All

AvgAgeChurned Customers by Customer Status

- Bar visual created, avg. age of churned customer measure clearly shows the average. age of churned customers.

12. Analyze the Distribution of Monthly Charges Among Churned Customers



Distribution of Monthly Charge - Churned Customers

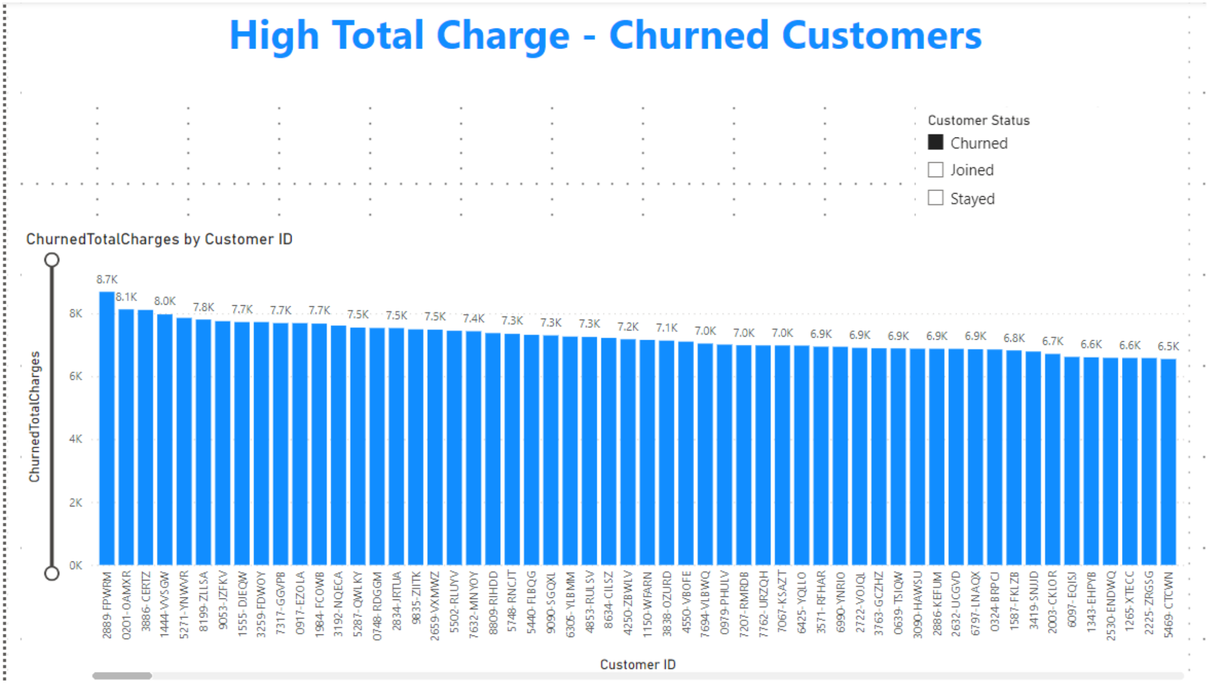
Customer Status

Churned | Joined | Stayed

Sum of Monthly Charge by Monthly Charge (bins) and Customer Status

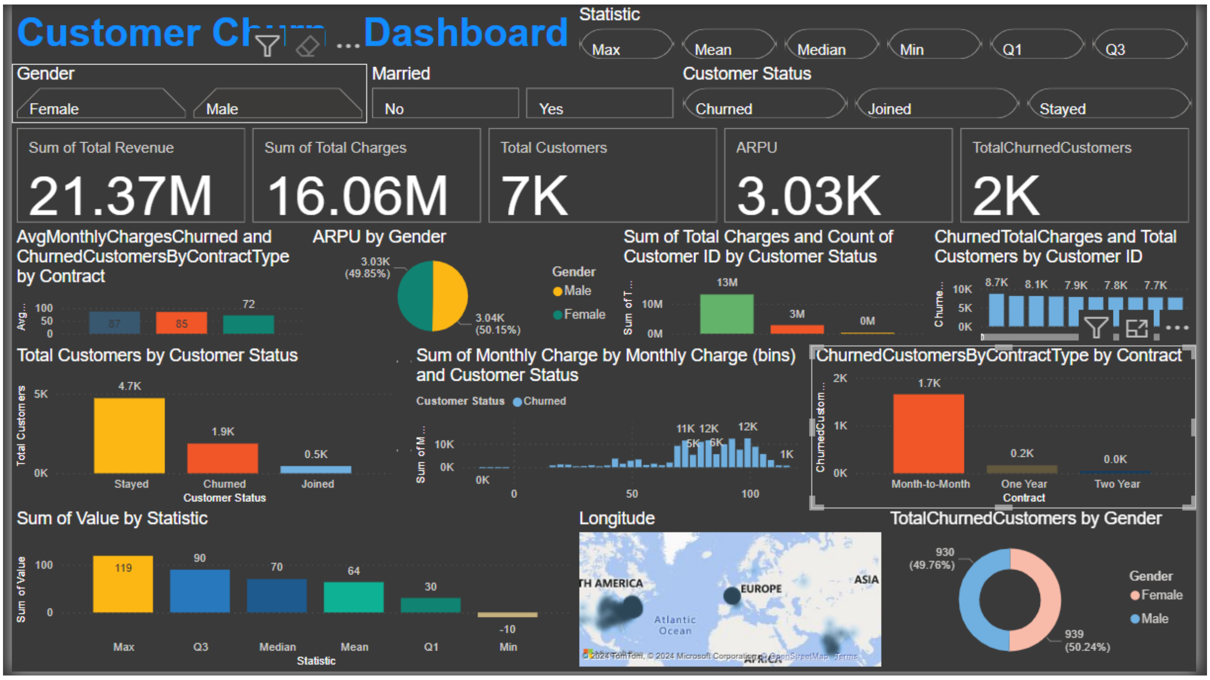
- For this visual, a monthly bin group was created, and customer status – was set as “churned’.

13. Identify Customers with High Total Charges Who Have Churned



- From this, we can understand the High Total Charge among churned customers.
- Churned Total Charge measure used.
- Customer status "churned"

Dashboards:


Customer Churn Dashboard
Statistic
Max
Mean
Median
Min
Q1
Q3
Gender
Female
Male
Married
No
Yes
Customer Status
Churned
Joined
Stayed
Sum of Total Revenue
21.37M
Sum of Total Charges
16.06M
Total Customers
7K
ARPU
3.03K
TotalChurnedCustomers
2K
AvgMonthlyChargesChurned and ChurnedCustomersByContractType by Contract
100
50
0
87
85
72
ARPU by Gender
3.03K (49.85%)
3.04K (50.15%)
Gender
Male
Female
Sum of Total Charges and Count of Customer ID by Customer Status
13M
3M
0M
10M
0M
ChurnedTotalCharges and Total Customers by Customer ID
10K
5K
0K
8.7K
8.1K
7.9K
7.8K
7.7K
Total Customers by Customer Status
Total Customers
5K
0K
4.7K
1.9K
0.5K
Stayed
Churned
Joined
Customer Status
Sum of Monthly Charge by Monthly Charge (bins) and Customer Status
Customer Status
Churned
10K
0K
11K
12K
5K
6K
12K
1K
0K
0
50
100
ChurnedCustomersByContractType by Contract
2K
1K
0K
1.7K
0.2K
0.0K
Month-to-Month
One Year
Two Year
Contract
Sum of Value by Statistic
Sum of Value
100
0
119
90
70
64
30
-10
Max
Q3
Median
Mean
Q1
Min
Statistic
Longitude
TH AMERICA
EUROPE
ASIA
Atlantic Ocean
AFRICA
© 2024 TomTom, © 2024 Microsoft Corporation, © OpenStreetMap Terms
TotalChurnedCustomers by Gender
930 (49.76%)
939 (50.24%)
Gender
Female
Male

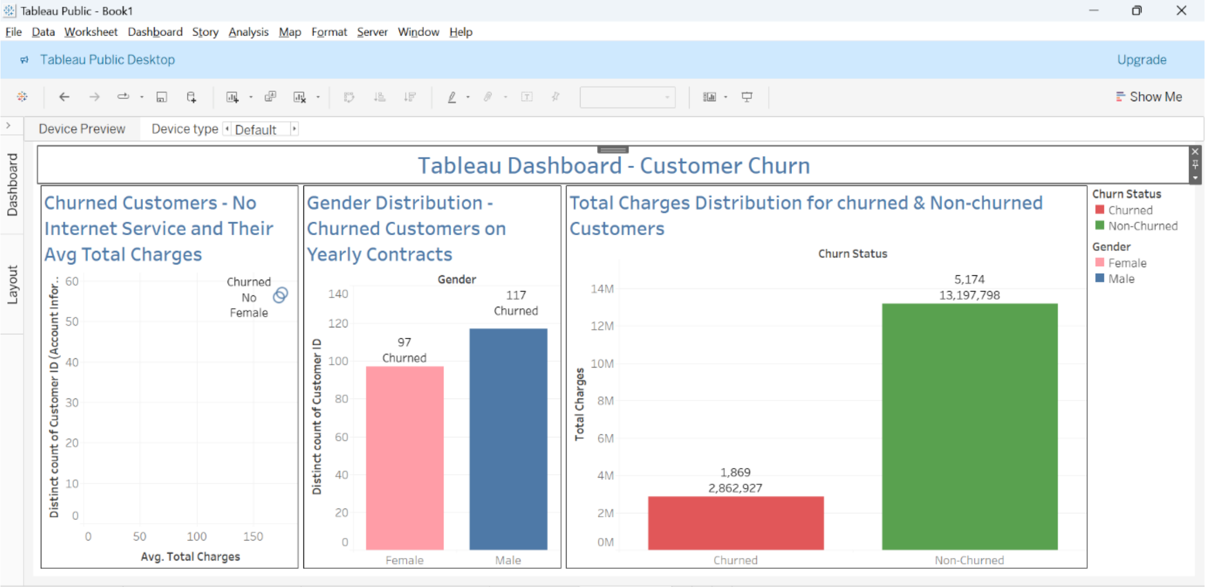


Tableau Public - Book1
File
Data
Worksheet
Dashboard
Story
Analysis
Map
Format
Server
Window
Help
Tableau Public Desktop
Upgrade
Show Me
Device Preview
Device type
Default
Dashboard
Layout
Tableau Dashboard - Customer Churn
Churned Customers - No Internet Service and Their Avg Total Charges
Distinct count of Customer ID (Account Infor...
60
50
40
30
20
10
0
Churned
No
Female
0
50
100
150
Avg. Total Charges
Gender Distribution - Churned Customers on Yearly Contracts
Gender
Distinct count of Customer ID
140
120
100
80
60
40
20
0
97
Churned
117
Churned
Female
Male
Total Charges Distribution for churned & Non-churned Customers
Churn Status
Total Charges
14M
12M
10M
8M
6M
4M
2M
0M
1,869
2,862,927
5,174
13,197,798
Churned
Non-Churned
Churn Status
Churned
Non-Churned
Gender
Female
Male

Power BI Dashboard Explanation

Filters at the Top:

- **Gender**: Allows filtering data by gender (Female, Male).
- **Married**: Allows filtering by marital status (Yes, No).
- **Customer Status**: Allows filtering by customer status (Churned, Joined, Stayed).

Key Metrics:

- **Sum of Total Revenue**: Displays the total revenue generated.
- **Sum of Total Charges**: Displays the total charges.
- **Total Customers**: Displays the total number of customers.
- **ARPU (Average Revenue Per User)**: Displays the ARPU value.
- **Total Churned Customers**: Displays the total number of customers who have churned.

Charts and Graphs:

- **Avg Monthly Charges Churned and Churned Customers by Contract Type**: A bar chart showing average monthly charges for churned customers categorized by contract type.
- **Total Customers by Customer Status**: A bar chart displaying the number of customers by their status (Stayed, Churned, Joined).
- **ARPU by Gender**: A pie chart showing ARPU values for Male and Female customers.
- **Sum of Total Charges and Count of Customer ID by Customer Status**: A bar chart representing the sum of total charges and count of customer IDs, categorized by customer status.
- **Sum of Monthly Charge by Monthly Charge (bins) and Customer Status**: A histogram showing the distribution of monthly charges for churned customers.
- **Churned Total Charges and Total Customers by Customer ID**: A bar chart showing total charges and the number of churned customers.
- **Churned Customers by Contract Type by Contract**: A bar chart showing the number of churned customers by contract type.
- **Total Churned Customers by Gender**: A pie chart representing the gender distribution of churned customers.

Map:

- **Total Churned Customers by Geography**: A map visual showing the geographical distribution of churned customers.

Other Visuals:

- **Sum of Value by Statistic**: A bar chart displaying the sum of values by various statistics like Max, Q3, Median, Mean, Q1, Min.

Tableau Dashboard Explanation

1. **Churned Customers - No Internet Service and Their Average Total Charges**:
    - This visualization shows the distribution of churned customers who do not use internet service and their average total charges.
2. **Gender Distribution - Churned Customers on Yearly Contracts**:
    - This bar chart shows the distribution of churned customers on yearly contracts by gender.
3. **Total Charges Distribution for Churned and Non-Churned Customers**:
    - This bar chart compares the total charges for churned and non-churned customers.

General Insights from Both Dashboards

- **Churn Analysis by Contract Type**:
    - In Power BI, the dashboard reveals that month-to-month contracts have the highest churn rate, which is visually represented in the "Churned Customers by Contract Type" bar chart.
- **Gender Distribution**:
    - Both dashboards provide insights into the gender distribution of churned customers. In Tableau, a bar chart specifically highlights churned customers on yearly contracts by gender.
- **Revenue and Charges**:
    - Both dashboards display metrics like total revenue, total charges, and ARPU. Power BI provides detailed breakdowns and distributions of these metrics.
- **Customer Status and Churn Rate**:
    - Both dashboards give an overview of customer status, highlighting the number of churned, stayed, and joined customers. Power BI goes further by breaking down these metrics by various dimensions like contract type and monthly charges.
- **Geographical Distribution**:
    - The Power BI dashboard includes a map visual showing the geographical distribution of churned customers, which helps in understanding regional churn patterns.

These dashboards collectively provide a comprehensive view of customer churn, enabling the identification of key factors contributing to churn, demographic insights, and financial impacts. They can be used to inform strategic decisions to improve customer retention.

Phase-4: Recommendations Based on Business Use Cases

1. Customer Retention: Identify At-Risk Customers and Proactively Implement Retention Strategies

Objective: Minimize customer churn by identifying and addressing the needs of at-risk customers.

Recommendations:

- **Utilize Predictive Analytics:** Use machine learning models to predict which customers are most likely to churn. Implement targeted retention strategies for these customers.
- **Personalized Communication:** Reach out to at-risk customers with personalized offers and communication. Use data from the dashboards to tailor messages based on customer behavior and preferences.
- **Loyalty Programs:** Implement or enhance loyalty programs to reward long-term customers and provide incentives for staying.
- **Customer Feedback:** Regularly collect and act on customer feedback to address issues before they lead to churn.

2. Marketing Campaigns: Tailor Marketing Efforts Towards Customers Who Are More Likely to Churn

Objective: Increase the effectiveness of marketing campaigns by focusing on customers who are at risk of leaving.

Recommendations:

- **Segmented Campaigns:** Use customer segmentation data to create targeted marketing campaigns. For instance, customers with high monthly charges but who are not using certain services can be targeted with campaigns promoting those services.
- **Special Offers:** Develop special offers and discounts specifically aimed at at-risk customers to encourage them to stay.
- **Engagement Strategies:** Implement engagement strategies that address the specific needs and pain points of at-risk customers.

3. Service Improvement: Analyze Churn Patterns to Improve Service Offerings and Customer Support

Objective: Enhance service quality and customer support to reduce churn rates.

Recommendations:

- **Service Usage Analysis:** Analyze the service usage patterns of churned customers to identify which services are underutilized or problematic. Improve these services based on customer feedback and usage data.
- **Support Enhancements:** Invest in improving customer support channels, including quicker response times, better training for support staff, and more comprehensive self-service options.

- **Quality Assurance:** Regularly monitor and improve service quality, ensuring that customers receive consistent and reliable service.

4. Revenue Optimization: Reduce Churn Rates to Maintain a Steady Revenue Stream

Objective: Ensure stable and growing revenue by minimizing customer churn.

Recommendations:

- **Churn Impact Analysis:** Regularly assess the financial impact of churn and prioritize efforts to retain high-value customers.
- **Revenue Forecasting:** Use historical churn data to forecast revenue and identify potential risks. Adjust business strategies accordingly.
- **Bundling Services:** Offer bundled services at a discounted rate to increase customer stickiness and reduce churn.

5. Customer Segmentation: Segment Customers Based on Churn Probability to Offer Personalized Experiences

Objective: Enhance customer experience by providing personalized services and offers.

Recommendations:

- **Behavioral Segmentation:** Use data from the dashboards to segment customers based on their behavior, such as service usage, support interactions, and purchase history.
- **Personalized Offers:** Develop personalized offers and recommendations for different customer segments. For instance, customers on monthly contracts might be offered discounts for switching to annual contracts.
- **Experience Enhancement:** Use segmentation data to enhance the overall customer experience, addressing specific needs and preferences of different customer groups.

Conclusion

By leveraging the insights from the dashboards and implementing these recommendations, the business can proactively address customer churn, tailor marketing efforts, improve service quality, optimize revenue, and provide personalized experiences to different customer segments. This holistic approach will help in achieving long-term customer retention and business growth.