Customer Churn Analysis: Insights from Data Analysis Project Using SQL Server

In this project, I've conducted a customer churn analysis for a bank using SQL Server. Using various analytical techniques, I've examined multiple factors that may contribute to customer churn. The queries address topics ranging from the overall churn rate to churn based on demographics, customer behavior, account balance, number of products, and customer satisfaction. The results from this analysis can serve as a valuable resource for understanding the patterns and factors that contribute to customer attrition. Consequently, this information can guide strategic decision-making to improve customer retention and maintain a loyal customer base. The project also includes the creation of views to facilitate further exploration and visualization of the data in Tableau, thereby enabling a more intuitive and interactive presentation of the findings.

```sql
--What is the overall churn rate?
--COUNT (*) TO FIND THE TOTAL NUMBER OF CUSTOMERS.
-- THE CAST FUNCTION TO ENSURE THE RESULT IS DECIMAL.

SELECT
    CAST(SUM(CASE WHEN Exited = 1 THEN 1 ELSE 0 END) AS FLOAT) / COUNT(*) AS
ChurnRate
FROM [dbo].['Customer-Churn-Records$']
```

Results Messages

	ChurnRate
1	0.2038

```sql
--How does churn vary by gender and geography?

SELECT Geography, Gender,
ROUND(CAST(SUM(CASE WHEN Exited = 1 THEN 1 ELSE 0 END) AS FLOAT) / COUNT(*),2) AS
ChurnRate
FROM [dbo].['Customer-Churn-Records$']
GROUP BY Geography, Gender;
```

Results Messages

	Geography	Gender	ChurnRate
1	Germany	Male	0.28
2	Germany	Female	0.38
3	Spain	Male	0.13
4	Spain	Female	0.21
5	France	Male	0.13
6	France	Female	0.2

```sql
--What is the average credit score for churned customers versus retained
customers?
SELECT Exited, ROUND(AVG(CREDITSCORE),2) AS AVGCREDITSCORE
FROM [dbo].['Customer-Churn-Records$']
GROUP BY Exited
```

Results Messages

	Exited	AVGCREDITSCORE
1	0	651.84
2	1	645.41

```sql
--Do customers with higher balances have a higher churn rate?
--FIRST: I WILL CREATE A BALANCERANGE COLUMN USING THE CASE STATMENT
--SECOND:A TOTAL CUSTOMERS COLUMN BY ADDING THE NUMBER OF ALL ROWS.
--THIRD: CHURN CUSTOMERS TOTAL COLUMN BY ADDING THE NUMBERS IN THE COLUMN SINCE
THEY ARE 0 FOR NON-CHURN AND 1 FOR CHURN CUSTOMERS
SELECT
    CASE
        WHEN Balance >= 0 AND Balance < 10000 THEN '0-10,000'
        WHEN Balance >= 10000 AND Balance < 50000 THEN '10,000-50,000'
        WHEN Balance >= 50000 AND Balance < 100000 THEN '50,000-100,000'
        WHEN Balance >= 100000 THEN '100,000+'
        ELSE 'Unknown'
    END AS BalanceRange,
    COUNT(*) AS TotalCustomers,
    SUM(Exited) AS ChurnedCustomers,
    100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
FROM [dbo].['Customer-Churn-Records$']
GROUP BY
    CASE
        WHEN Balance >= 0 AND Balance < 10000 THEN '0-10,000'
        WHEN Balance >= 10000 AND Balance < 50000 THEN '10,000-50,000'
        WHEN Balance >= 50000 AND Balance < 100000 THEN '50,000-100,000'
        WHEN Balance >= 100000 THEN '100,000+'
        ELSE 'Unknown'
    END
ORDER BY TotalCustomers
```

	BalanceRange	TotalCustomers	ChurnedCustomers	ChurnRate
1	10,000-50,000	74	25	33.7837837837838
2	50,000-100,000	1509	301	19.946984758118
3	0-10,000	3618	501	13.8474295190713
4	100,000+	4799	1211	25.2344238382996

```sql
--Does the number of products a customer has affect their likelihood to churn?

SELECT
    NumOfProducts,
    COUNT(*) AS TotalCustomers,
    SUM(Exited) AS ChurnedCustomers,
    100.0 * SUM(Exited) / COUNT(*) AS ChurnRate

    FROM [dbo].['Customer-Churn-Records$']
GROUP BY
    NumOfProducts
ORDER BY
    NumOfProducts;
```

Results | Messages

	NumOfProducts	TotalCustomers	ChurnedCustomers	ChurnRate
1	1	5084	1409	27.7143981117231
2	2	4590	349	7.60348583877996
3	3	266	220	82.7067669172932
4	4	60	60	100

```sql
-- TO MAKE SURE THAT ALL CUSTOMERS WITH 4 PRODUCTS CHURNED
    SELECT NumOfProducts, EXITED
    FROM [dbo].['Customer-Churn-Records$']
    WHERE NumOfProducts = 4
```

Results | Messages

	NumOfProducts	EXITED
1	4	1
2	4	1
3	4	1
4	4	1
5	4	1
6	4	1
7	4	1
8	4	1
9	4	1
10	4	1
11	4	1
12	4	1
13	4	1
14	4	1
15	4	1
16	4	1

```sql
--How does the churn rate differ for customers with a credit card versus those
without?

SELECT
    HasCrCard
    COUNT(*) AS TotalCustomers,
    SUM(Exited) AS ChurnedCustomers,
    100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
    FROM [dbo].['Customer-Churn-Records$']
GROUP BY
    HasCrCard;
```

Results Messages

	HasCrCard_New	TotalCustomers	ChurnedCustomers	ChurnRate
1	Yes	7055	1425	20.198440822112
2	No	2945	613	20.8149405772496

```sql
--Is there a difference in the churn rate for active and inactive members?
SELECT [IsActiveMember],
    100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
    FROM [dbo].['Customer-Churn-Records$']
    GROUP BY [IsActiveMember]
```

Results Messages

	IsActiveMember	ChurnRate
1	0	26.8715199010105
2	1	14.2690739662202

```sql
--Does the estimated salary of customers affect the churn rate?

SELECT
CASE
    WHEN EstimatedSalary < 50000 THEN 'Less than 50,000'
    WHEN EstimatedSalary >= 50000 AND EstimatedSalary < 100000 THEN '50,000 -
99,999'
    WHEN EstimatedSalary >= 100000 AND EstimatedSalary < 150000 THEN '100,000 -
149,999'
    ELSE '150,000 or more'
END AS SalaryRange,
100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
FROM [dbo].['Customer-Churn-Records$']
GROUP BY
CASE
    WHEN EstimatedSalary < 50000 THEN 'Less than 50,000'
    WHEN EstimatedSalary >= 50000 AND EstimatedSalary < 100000 THEN '50,000 -
99,999'
    WHEN EstimatedSalary >= 100000 AND EstimatedSalary < 150000 THEN '100,000 -
149,999'
    ELSE '150,000 or more'
END
ORDER BY
SalaryRange;
```

Results Messages

	SalaryRange	ChurnRate
1	100,000 - 149,999	20.2348336594912
2	150,000 or more	21.5071283095723
3	50,000 - 99,999	19.8659834450138
4	Less than 50,000	19.9347737464329

```sql
--Is there a correlation between a customer's satisfaction score and their
likelihood to churn?
SELECT [Satisfaction Score], 100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
FROM [dbo].['Customer-Churn-Records$']
GROUP BY  [Satisfaction Score]
ORDER BY 1 ASC
```

Results | Messages

	Satisfaction Score	ChurnRate
1	1	20.0310559006211
2	2	21.7974180734856
3	3	19.6376101860921
4	4	20.6175298804781
5	5	19.810379241517

```sql
--Do customers who have previously complained have a higher churn rate?
SELECT [Complain], 100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
FROM [dbo].['Customer-Churn-Records$']
GROUP BY [Complain]
```

Results | Messages

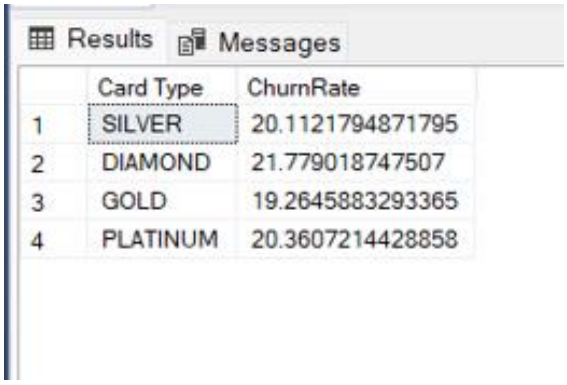
	Complain	ChurnRate
1	0	0.0502765208647562
2	1	99.5107632093933

```sql
--What is the average tenure of churned customers versus retained customers?
SELECT Exited, AVG([Tenure])
FROM [dbo].['Customer-Churn-Records$']
GROUP BY Exited
```

Results | Messages

	Exited	avgtenure
1	0	5.03278070836473
2	1	4.93473994111874

```sql
--Does the type of card a customer has affect their churn rate?
SELECT [Card Type], 100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
FROM [dbo].['Customer-Churn-Records$']
GROUP BY  [Card Type]
```



Results | Messages

	Card Type	ChurnRate
1	SILVER	20.1121794871795
2	DIAMOND	21.779018747507
3	GOLD	19.2645883293365
4	PLATINUM	20.3607214428858

```sql
-- CREATING VIEWS FOR THE TABLEAU DASHBOARD
--1
CREATE VIEW CHUNRATEBYCOUNTRY AS
SELECT  Geography, Gender,
ROUND(CAST(SUM(CASE WHEN Exited = 1 THEN 1 ELSE 0 END) AS FLOAT) / COUNT(*),2) AS
ChurnRate
FROM [dbo].['Customer-Churn-Records$']
GROUP BY Geography, Gender;


--2
  CREATE VIEW CHUNRATEBYCREDITSCORE AS
  SELECT Exited, ROUND(AVG(CREDITSCORE),2) AS AVGCREDITSCORE
FROM [dbo].['Customer-Churn-Records$']
GROUP BY Exited
--3
CREATE VIEW CHUNRATEBYBALANCERANGE AS
SELECT
    CASE
        WHEN Balance >= 0 AND Balance < 10000 THEN '0-10,000'
        WHEN Balance >= 10000 AND Balance < 50000 THEN '10,000-50,000'
        WHEN Balance >= 50000 AND Balance < 100000 THEN '50,000-100,000'
        WHEN Balance >= 100000 THEN '100,000+'
        ELSE 'Unknown'
    END AS BalanceRange,
    COUNT(*) AS TotalCustomers,
```

```
    SUM(Exited) AS ChurnedCustomers,
    100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
FROM [dbo].['Customer-Churn-Records$']
GROUP BY
    CASE
        WHEN Balance >= 0 AND Balance < 10000 THEN '0-10,000'
        WHEN Balance >= 10000 AND Balance < 50000 THEN '10,000-50,000'
        WHEN Balance >= 50000 AND Balance < 100000 THEN '50,000-100,000'
        WHEN Balance >= 100000 THEN '100,000+'
        ELSE 'Unknown'
    END

    --4
    CREATE VIEW CHUNBYNUMOFPRODUCTS AS
    SELECT
    NumOfProducts,
    COUNT(*) AS TotalCustomers,
    SUM(Exited) AS ChurnedCustomers,
    100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
    FROM [dbo].['Customer-Churn-Records$']
GROUP BY
    NumOfProducts

    --5
    CREATE VIEW CHUNIFHAVECRCARD AS
    SELECT
    HasCrCard_New,
    COUNT(*) AS TotalCustomers,
    SUM(Exited) AS ChurnedCustomers,
    100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
    FROM [dbo].['Customer-Churn-Records$']
GROUP BY
    HasCrCard_New;

    --6
    CREATE VIEW CHUNIFACTIVEORNOT AS
    SELECT [IsActiveMember],
    100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
    FROM [dbo].['Customer-Churn-Records$']
    GROUP BY [IsActiveMember]

    --7
    CREATE VIEW BYESTIMATEDSALARY AS

SELECT
```

```sql
CASE
    WHEN EstimatedSalary < 50000 THEN 'Less than 50,000'
    WHEN EstimatedSalary >= 50000 AND EstimatedSalary < 100000 THEN '50,000 -
99,999'
    WHEN EstimatedSalary >= 100000 AND EstimatedSalary < 150000 THEN '100,000 -
149,999'
    ELSE '150,000 or more'
END AS SalaryRange,
100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
FROM [dbo].['Customer-Churn-Records$']
GROUP BY
CASE
    WHEN EstimatedSalary < 50000 THEN 'Less than 50,000'
    WHEN EstimatedSalary >= 50000 AND EstimatedSalary < 100000 THEN '50,000 -
99,999'
    WHEN EstimatedSalary >= 100000 AND EstimatedSalary < 150000 THEN '100,000 -
149,999'
    ELSE '150,000 or more'
END

--8
CREATE VIEW CORRBYSATISFACTION AS
SELECT [Satisfaction Score], 100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
FROM [dbo].['Customer-Churn-Records$']
GROUP BY  [Satisfaction Score]


--9
CREATE VIEW COMPLAINEDBEFORE AS
SELECT [Complain], 100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
FROM [dbo].['Customer-Churn-Records$']
GROUP BY [Complain]

--10
CREATE VIEW AVGTENURE AS
SELECT Exited, AVG([Tenure]) AS AVGTENURE
FROM [dbo].['Customer-Churn-Records$']
GROUP BY Exited

--11
CREATE VIEW CARDTYPE AS
SELECT [Card Type], 100.0 * SUM(Exited) / COUNT(*) AS ChurnRate
FROM [dbo].['Customer-Churn-Records$']
GROUP BY  [Card Type]
```