SQL QUERIES

Statistics Summary for Credit Score & Balance

```
SELECT
    'CreditScore' AS Metric,
    ROUND(CAST(AVG(CreditScore) AS NUMERIC), 2) AS Mean,
    ROUND(CAST(PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY CreditScore) AS NUMERIC), 2) AS Median,
    ROUND(CAST(STDDEV(CreditScore) AS NUMERIC), 2) AS Std_Dev,
    MIN(CreditScore) AS Min,
    MAX(CreditScore) AS Max
FROM Customers
UNION ALL
SELECT
    'Balance',
    ROUND(CAST(AVG(Balance) AS NUMERIC), 2),
    ROUND(CAST(PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY Balance) AS NUMERIC), 2),
    ROUND(CAST(STDDEV(Balance) AS NUMERIC), 2),
    ROUND(CAST(MIN(Balance) AS NUMERIC), 2),
    ROUND(CAST(MAX(Balance) AS NUMERIC), 2)
FROM Customers;
```

Customer Churn Analysis by Key Metrics

```
SELECT
    Exited,
    COUNT(*) AS Total_Customers,
    ROUND(AVG(Age)::numeric, 2) AS Avg_Age,

    ROUND(AVG(CreditScore)::numeric, 2) AS Avg_CreditScore,

    ROUND(AVG(Balance)::numeric, 2) AS Avg_Balance,

    ROUND(AVG(EstimatedSalary)::numeric, 2) AS Avg_EstimatedSalary,

    ROUND(AVG(NumOfProducts)::numeric, 2) AS Avg_NumOfProducts,

    ROUND(AVG(IsActiveMember)::numeric, 2) AS Avg_ActiveMember

FROM Customers

GROUP BY Exited;
```

Churn by Geography, Gender, and Age Group

```
SELECT

    Geography,

    Gender,

    CASE

        WHEN Age < 30 THEN 'Under 30'

        WHEN Age BETWEEN 30 AND 50 THEN '30-50'

        ELSE 'Above 50'

    END AS Age_Group,

    Exited,

    COUNT(*) AS Total_Customers

FROM Customers

GROUP BY Geography, Gender, Age_Group, Exited

ORDER BY Geography, Gender, Age_Group, Exited;
```

Churn by Credit Score, Balance, and Number of Products

```
SELECT

    CASE

        WHEN CreditScore < 500 THEN 'Low (Below 500)'
        WHEN CreditScore BETWEEN 500 AND 700 THEN 'Medium (500-700)'
        ELSE 'High (Above 700)'
    END AS Credit_Score_Category,
    CASE
        WHEN Balance = 0 THEN 'No Balance'
        WHEN Balance BETWEEN 1 AND 100000 THEN 'Low Balance (<100K)'
        ELSE 'High Balance (>100K)'
    END AS Balance_Category,
    NumOfProducts,
    Exited,
    COUNT(*) AS Total_Customers
FROM Customers
GROUP BY Credit_Score_Category, Balance_Category, NumOfProducts, Exited
ORDER BY Credit_Score_Category, Balance_Category, NumOfProducts, Exited;
```

Churn Analysis by Credit Card Ownership

```
SELECT
    HasCrCard,
    Exited,
    COUNT(*) AS customer_count,
    ROUND(100.0 * COUNT(*) / SUM(COUNT(*)) OVER (PARTITION BY HasCrCard), 2) AS churn_percentage
FROM customers
GROUP BY HasCrCard, Exited
ORDER BY HasCrCard, Exited DESC;
```

Credit Card Holders & Churn Analysis

```
SELECT
    HasCrCard,
    Exited,
    COUNT(*) AS churn_count,
    ROUND(AVG(CreditScore)::numeric, 2) AS avg_credit_score,
    ROUND(AVG(Balance)::numeric, 2) AS avg_balance,
    ROUND(AVG(Age)::numeric, 2) AS avg_age
FROM customers
WHERE Exited = 1
GROUP BY HasCrCard, Exited
ORDER BY HasCrCard;
```

Credit Card Ownership & Overall Churn Rate

```
SELECT
    HasCrCard,
    COUNT(*) AS total_customers,
    SUM(CASE WHEN Exited = 1 THEN 1 ELSE 0 END) AS churned_customers,
    ROUND(100.0 * SUM(CASE WHEN Exited = 1 THEN 1 ELSE 0 END) / COUNT(*), 2) AS churn_rate
FROM customers
GROUP BY HasCrCard
ORDER BY churn_rate DESC;
```

Top 10 Customers Based on Credit Score, Balance & Products

```
SELECT
    CustomerId,
    Surname,
    CreditScore,
    Balance,
    NumOfProducts,
    Age,
    ROW_NUMBER() OVER (ORDER BY CreditScore DESC, Balance DESC, NumOfProducts DESC) AS rank
FROM customers
LIMIT 10;
```

Top 5 Customers for Special Interest Rate

```
SELECT
    CustomerId,
    Surname,
    CreditScore,
    Balance,
    NumOfProducts,
    Age,
    IsActiveMember
FROM customers
WHERE CreditScore > 750 AND Balance > 100000 AND NumOfProducts > 1
ORDER BY CreditScore DESC, Balance DESC
LIMIT 5;
```

Churn Analysis by Geography and Active Membership

```
SELECT
    Geography,
    IsActiveMember,
    Exited,
    COUNT(*) AS churn_count,
    ROUND(AVG(CreditScore)::numeric, 2) AS avg_credit_score,
    ROUND(AVG(Balance)::numeric, 2) AS avg_balance,
    ROUND(AVG(Age)::numeric, 2) AS avg_age
FROM customers
```

```
GROUP BY Geography, IsActiveMember, Exited

ORDER BY Exited DESC, churn_count DESC;
```

Churn Analysis by Geography & Active Status (Excluding Credit Score)

```
SELECT

    Geography,

    IsActiveMember,

    Exited,

    COUNT(*) AS churn_count,

    ROUND(AVG(Balance)::numeric, 2) AS avg_balance,

    ROUND(AVG(Age)::numeric, 2) AS avg_age

FROM customers

GROUP BY Geography, IsActiveMember, Exited

ORDER BY Exited DESC, churn_count DESC;
```