CODEFILE

1) CREATE TABLE

```
create table customer_churn(
RowNumber int,
CustomerId int,
Surname varchar (25),
CreditScore int,
Geography varchar (7),
Gender char (6),
Age int,
Tenure int,
Balance decimal(10,2),
NumOfProducts int,
HasCrCard int,
IsActiveMember int,
EstimatedSalary decimal(10,2),
Exited int);
```

2) DATASET IMPORT

```
copy customer_churn (RowNumber, CustomerId, Surname, CreditScore, Geography, Gender, Age,
Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, EstimatedSalary, Exited)
from 'D:\pavan\INTERNSHIP\customer_churn.csv'
DELIMITER ','
CSV HEADER;
select * from customer_churn
```

3) AVERAGE CREDITSCORE AND BALANCE

```
SELECT
  ROUND(AVG(CreditScore),2) AS average_credit_score,
  ROUND(AVG(Balance), 2) AS average_balance
FROM customer_churn;
```

4) SUMMARY STATISTICS FOR NUMERICAL VARIABLES USING SQL

```
SELECT
'Mean' AS Metric,
ROUND(AVG(CreditScore), 2) AS "Credit Score",
ROUND(AVG(Age), 2) AS "Age",
ROUND(AVG(Tenure), 2) AS "Tenure",
ROUND(AVG(Balance), 2) AS "Balance",
ROUND(AVG(EstimatedSalary), 2) AS "EstimatedSalary"
FROM customer_churn
UNION ALL
SELECT
'Median' AS Metric,
ROUND(PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY CreditScore)::numeric, 2) AS "Credit
Score",
ROUND(PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY Age)::numeric, 2) AS "Age",
ROUND(PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY Tenure)::numeric, 2) AS "Tenure",
```

```
ROUND(PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY Balance)::numeric, 2) AS "Balance",
ROUND(PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY EstimatedSalary)::numeric, 2) AS
"EstimatedSalary"
FROM customer_churn
UNION ALL
SELECT
'Min' AS Metric,
MIN(CreditScore) AS "Credit Score",
MIN(Age) AS "Age",
MIN(Tenure) AS "Tenure",
MIN(Balance) AS "Balance",
MIN(EstimatedSalary) AS "EstimatedSalary"
FROM customer_churn
UNION ALL
SELECT
'Max' AS Metric,
MAX(CreditScore) AS "Credit Score",
MAX(Age) AS "Age",
MAX(Tenure) AS "Tenure",
MAX(Balance) AS "Balance",
MAX(EstimatedSalary) AS "EstimatedSalary"
FROM customer_churn
UNION ALL
SELECT
'STDDEV' AS Metric,
ROUND(STDDEV(CreditScore), 2) AS "Credit Score",
ROUND(STDDEV(Age),2) AS "Age",
ROUND(STDDEV(Tenure),2) AS "Tenure",
ROUND(STDDEV(Balance), 2) AS "Balance",
ROUND(STDDEV(EstimatedSalary), 2) AS "EstimatedSalary"
FROM customer_churn;
```

5) QUERY FOR TOTAL CUSTOMER IS CHURNED AND NOT CHURNED

```
select
 SUM(CASE WHEN exited = 1 THEN 1 ELSE 0 END) AS Churned,
 SUM(CASE WHEN exited = 0 THEN 1 ELSE 0 END) AS Non_Churned
FROM customer_churn;
```

6) EXPLORE THE TOP 5 CUSTOMERS

```
WITH RankedCustomers AS (
SELECT CustomerId,CreditScore,Balance,Age,Tenure, NumOfProducts,EstimatedSalary,Exited,
ROW_NUMBER() OVER (PARTITION BY Exited ORDER BY CreditScore DESC) AS RankByCreditScore,
ROW_NUMBER() OVER (PARTITION BY Exited ORDER BY NumOfProducts DESC) AS
RankByNumOfPrducts,
ROW_NUMBER() OVER (PARTITION BY Exited ORDER BY EstimatedSalary DESC) AS
RankByEstimatedSalary
FROM customer_churn
)
```

```
SELECT CustomerId,CreditScore,Balance,Age,Tenure,NumOfProducts,EstimatedSalary,Exited,
CASE WHEN Exited = 1 then 'Churned' else 'NON_Churned' end as Churn_status
FROM RankedCustomers
WHERE RankByCreditScore <= 5 OR RankByNumOfPrducts <=5 OR RankByEstimatedSalary <=5
ORDER BY Exited DESC, RankByCreditScore, RankByNumOfPrducts,RankByEstimatedSalary;
```

7) TOP 5 CUSTOMER BASED ON CREDIT SCORE

```
WITH TopCustomers AS (
SELECT
CustomerId,Surname,CreditScore,Exited,
ROW_NUMBER() OVER (ORDER BY CreditScore DESC) AS RankByCreditScore
FROM customer_churn)
SELECT CustomerId,Surname,CreditScore,
CASE WHEN Exited = 1 then 'Churned' else 'NON_Churned' end as Churn_status
FROM TopCustomers
WHERE RankByCreditScore <= 5;
```