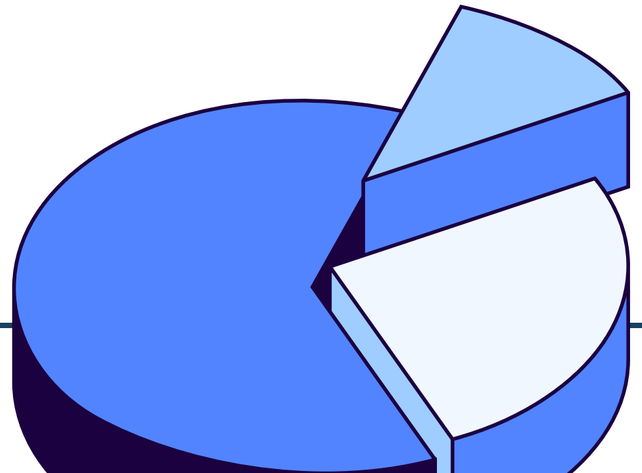
Financial Data 2025

BANKING INSTITUTION



Problems statements

1. Running Total of Credit Card Transactions
2. Calculate the 4-week moving average of the creditLimit for each client.
3. Calculate the mom% growth and wow% groth on transaction amount.
4. Calculate Customer Acquisition Cost (CAC) as a Ratio of Transaction Amount.
5. Calculate the yearly average of avg_utilization_ratio for all clients.
6. Calculate the percentage of Interest_Earned compared to Total_Revolving_Bal for each client.
7. Calculate Top 5 Clients by Total Transaction Amount.
8. Identify clients whose Avg_Utilization_Ratio exceeds 80%.
9. Customer Churn Indicator: Create a KPI that flags clients who have not made any transactions (Total_Trans_Amt = 0) in the last 6 months.
10. Delinquency Rate: Calculate the percentage of clients with Delinquent_Acc > 0.
11. Credit Risk Score: Create a score for each client based on their Avg_Utilization_Ratio, Delinquent_Acc, and Total_Revolving_Bal.
12. Income vs Credit Limit Correlation: Show the correlation between Income and Credit_Limit for all clients.
13. Average Customer Satisfaction Score by Credit Card Category: Calculate the average Cust_Satisfaction_Score by Card_Category.
14. Loan Approval vs Credit Limit: Analyze how Credit_Limit affects Personal_loan approval by calculating the average credit limit for clients with and without loans.
15. High Risk Clients Flag: Create a flag for clients whose Total_Revolving_Bal exceeds 90% of their Credit_Limit and who have a high Avg_Utilization_Ratio.

1) Running Total of Credit Card Transactions

```
Running_total = CALCULATE(
  SUM(credit_card[Total_Trans_Amt])
  ,FILTER(ALL(credit_card)
   ,credit_card[Week_Start_Date] <= MAX(credit_card[Week_Start_Date])
    )
  )
```

Week_Start_Date	Total transaction amount	Running_total
01 January 2023	$8,35,767	$8,35,767
08 January 2023	$8,44,739	$16,80,506
15 January 2023	$9,23,367	$26,03,873
22 January 2023	$8,69,235	$34,73,108
29 January 2023	$8,49,078	$43,22,186
05 February 2023	$8,98,867	$52,21,053
12 February 2023	$8,90,756	$61,11,809
19 February 2023	$8,68,091	$69,79,900

2) Calculate the 4-week moving average of the creditLimit for each client.

```
Moving_average_of_4_weeks =
VAR Week_4 =
    DATESINPERIOD(Calendar[Date], MAX(Calendar[Date]), -28, DAY)

VAR Total_amount =
    CALCULATE([Total transaction amount], Week_4)

VAR week_num =
    CALCULATE(DISTINCTCOUNT('Credit card'[Week_Num]), Week_4)

RETURN
    DIVIDE(Total_amount, week_num, 0)
```

Week_number	Total transaction amount	Moving_average_of_4_weeks
1	$8,35,767	$8,35,767.00
2	$8,44,739	$8,40,253.00
3	$9,23,367	$8,67,957.67
4	$8,69,235	$8,68,277.00
5	$8,49,078	$8,71,604.75
6	$8,98,867	$8,85,136.75
7	$8,90,756	$8,76,984.00
8	$8,68,091	$8,76,698.00
9	$8,81,861	$8,84,893.75
10	$7,93,080	$8,58,447.00
11	$9,15,725	$8,64,689.25
12	$8,90,081	$8,70,186.75

3) Calculate the mom% growth and wow% groth on transaction amount.

```
MoM_Growth_% =
VAR PrevMonth =
  CALCULATE (
    SUM ( 'credit_card'[Total_Trans_Amt] ),
    DATEADD ( 'Calendar'[Date], -1, MONTH )
  )
VAR CurrMonth =
  SUM ( 'credit_card'[Total_Trans_Amt] )
RETURN
  DIVIDE ( CurrMonth - PrevMonth, PrevMonth, 0 )
```

```
WoW_Growth_% =
VAR PrevWeek =
 CALCULATE (
 SUM ( 'credit_card'[Total_Trans_Amt] ),
 DATEADD ( 'Calendar'[Date], -7, DAY )
 )
VAR CurrWeek =
 SUM ( 'credit_card'[Total_Trans_Amt] )
RETURN
 DIVIDE ( CurrWeek - PrevWeek, PrevWeek, 0 )
```

4) Calculate Customer Acquisition Cost (CAC) as a Ratio of Transaction Amount.

```
cac_ta =
  DIVIDE (
    SUM ( 'credit_card'[Customer_Acq_Cost] ),
    SUM ( 'credit_card'[Total_Trans_Amt] )
  )
```

5) Calculate the yearly average of avg_utilization_ratio for all clients.

```
Yearly_Utilization_Ratio =

DIVIDE(
   SUM('credit_card'[Total_Trans_Amt]),
   sum(credit_card[Credit_Limit])
)
```

6) Calculate the percentage of Interest_Earned compared to Total_Revolving_Bal for each client.

```
interest_by_rev_bal =
  DIVIDE (
    SUM ( 'credit_card'[Interest_Earned] ),
    SUM ( 'credit_card'[Total_Revolving_Bal] ),
    0
  )
```

7) Calculate Top 5 Clients by Total Transaction Amount.

```
Top_5_Clients_Table =
TOPN (
  5,
  SUMMARIZE (
    'credit_card',
    'credit_card'[Client_Num],
    "Total_Amount", SUM ( 'credit_card'[Total_Trans_Amt] )
  ),
  [Total_Amount],
  DESC
)
```

8) Identify clients whose Avg_Utilization_Ratio exceeds 80%.

```
avg_uti_exceeds_80 =
IF (
    AVERAGE ( 'credit_card'[Avg_Utilization_Ratio] ) > 0.8,
    TRUE,
    FALSE
)
```

9) Customer Churn Indicator: Create a KPI that flags clients who have not made any transactions (Total_Trans_Amt = 0) in the last 6 months.

```
no_trans_in_last_6_months =
VAR months_6 =
    CALCULATE (
        SUM ( 'credit_card'[Total_Trans_Amt] ),
        DATESINPERIOD (
            'calendar'[Date],
            MAX ( 'calendar'[Date] ),
            -6,
            MONTH
        )
    )
RETURN
    IF ( months_6 = 0 || ISBLANK ( months_6 ), TRUE, FALSE )
```

10) Delinquency Rate: Calculate the percentage of clients with Delinquent_Acc > 0.

```
delinquency_rate =

var delinquent_acc =
    CALCULATE(
        COUNTROWS('credit_card'),
        'credit_card'[Delinquent_Acc] > 0
    )

var total_accounts =
    COUNTROWS('credit_card')

return
DIVIDE(delinquent_acc, total_accounts, 0)
```

11) Credit Risk Score: Create a score for each client based on their Avg_Utilization_Ratio, Delinquent_Acc, and Total_Revolving_Bal.

```
normalized_revolving_balance =
VAR min_value = MIN('credit_card'[Total_Revolving_Bal])
VAR max_value = MAX('credit_card'[Total_Revolving_Bal])
VAR total_rev_bal = SUM('credit_card'[Total_Revolving_Bal])
RETURN DIVIDE(
  total_rev_bal - min_value,
  max_value - min_value,
  0
)
credit_risk_score =

0.5 * [avg_uti_exceeds_80]+0.3*[delinquency_rate]+0.2*
[normalized_revolving_balance]
```

12) Income vs Credit Limit Correlation: Show the correlation between Income and Credit_Limit for all clients.

```
Income_Credit_Correlation =
VAR AvgX = AVERAGE(Customers[Income])
VAR AvgY = AVERAGE(Credit_Card[Credit_Limit])
VAR Co_variance =
  AVERAGEX(
    Customers,
    (Customers[Income] - AvgX) *
    (RELATED(Credit_Card[Credit_Limit]) - AvgY)
  )
VAR StdDevX = STDEVX.P(Customers, Customers[Income])
VAR StdDevY = STDEVX.P(Customers,
RELATED(Credit_Card[Credit_Limit]))

RETURN DIVIDE(Co_variance, StdDevX * StdDevY)
```

13) Average Customer Satisfaction Score by Credit Card Category: Calculate the average Cust_Satisfaction_Score by Card_Category.

```
avg_score_by_card_category =
SUMMARIZE(
  'credit_card',
  'credit_card'[Card_Category],
  "Avg Score",
  ROUND(
    AVERAGEX(
      RELATEDTABLE('customers'),
      customers[Cust_Satisfaction_Score]
    ),
    2
  )
)
```

14) Loan Approval vs Credit Limit: Analyze how Credit_Limit affects Personal_loan approval by calculating the average credit limit for clients with and without loans.

```
loan_yes =
CALCULATE(
  AVERAGE('credit_card'[Credit_Limit]),
  FILTER(
    'customers',
    'customers'[Personal_loan] = "yes"
  )
)
loan_no =
CALCULATE(
  AVERAGE('credit_card'[Credit_Limit]),
  FILTER(
    'customers',
    'customers'[Personal_loan] = "no"
  )
)
```

15) High Risk Clients Flag: Create a flag for clients whose Total_Revolving_Bal exceeds 90% of their Credit_Limit and who have a high Avg_Utilization_Ratio.

```
HighRiskFlag =
VAR RevolveBal =
CALCULATE(SUM('credit_card'[Total_Revolving_Bal]))
VAR CreditLimit = CALCULATE(SUM('credit_card'[Credit_Limit]))
var avg_uti_ratio =
CALCULATE(AVERAGE(credit_card[Avg_Utilization_Ratio]))
RETURN
IF(
  RevolveBal > 0.9 * CreditLimit
  && avg_uti_ratio > 0.8,
  1, 0
)
```

Thank You

Contact : Farook Mohammad

Phone No. : +91 9467671237