**CREDIT CARD SQL QUERRIES**

**KPI’s**

**--KPI 1: Total Customers**

select distinct COUNT (id) as total\_customer

from cleaned\_data;



**--KPI 2: Total Loan Amount**

SELECT

SUM(LIMIT\_BAL) AS Total\_Loan\_Amount

FROM

Cleaned\_Data;



-**-KPI 2: Total Customer by Marriage Status**

SELECT

cleaned\_marraige as Marraige,

COUNT(ID) AS Customer\_Count,

CAST(COUNT(ID) AS DECIMAL(10, 4)) \* 100.0 / SUM(COUNT(ID)) OVER () AS Percentage\_of\_Total

FROM

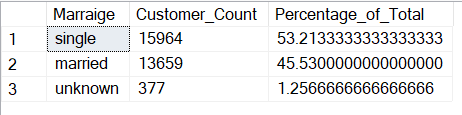
Cleaned\_Data

GROUP BY

cleaned\_marraige

ORDER BY

Customer\_Count DESC;



**--KPI 3:Total Customer by Age Bucket**

SELECT

age\_bucket,

COUNT(ID) AS Customer\_Count,

CAST(COUNT(ID) AS DECIMAL(10, 4)) \* 100.0 / SUM(COUNT(ID)) OVER () AS Percentage\_of\_Total

FROM

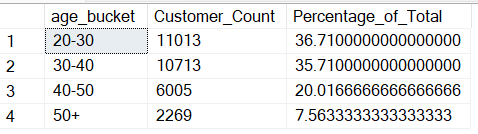
Cleaned\_Data

GROUP BY

age\_bucket

ORDER BY

Customer\_Count DESC;

****

**--KPI 3: Overall portfolio Default Rate**

select

cast(sum(CASE WHEN default\_payment\_next\_month = 1 THEN 1 ELSE 0 END) as decimal (10,4))\*100/count(id) as overall\_default\_rate\_percentage

from cleaned\_data;

****

**--KPI 4: Average Portfolio Credit Utilization**

SELECT

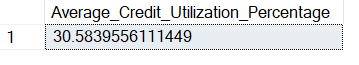
CAST(SUM(BILL\_AMT1) AS DECIMAL(18, 4)) \* 100.0 / SUM(LIMIT\_BAL) AS Average\_Credit\_Utilization\_Percentage

FROM

Cleaned\_Data

WHERE

LIMIT\_BAL > 0;

****

**-- KPI 5: monthly default rate (Trend Analysis)**

WITH MonthlyPaymentStatus AS (

-- Unpivoting PAY\_1 to PAY\_6 into a monthly structure

-- PAY\_1 (April)

SELECT ID, 'April' AS Payment\_Month, CASE WHEN PAY\_0 >= 2 THEN 1 ELSE 0 END AS Monthly\_Default\_Flag FROM Cleaned\_Data WHERE PAY\_0 IS NOT NULL

UNION ALL

-- PAY\_2 (May)

SELECT ID, 'May' AS Payment\_Month, CASE WHEN PAY\_2 >= 2 THEN 1 ELSE 0 END AS Monthly\_Default\_Flag FROM Cleaned\_Data WHERE PAY\_2 IS NOT NULL

UNION ALL

-- PAY\_3 (June)

SELECT ID, 'June' AS Payment\_Month, CASE WHEN PAY\_3 >= 2 THEN 1 ELSE 0 END AS Monthly\_Default\_Flag FROM Cleaned\_Data WHERE PAY\_3 IS NOT NULL

UNION ALL

-- PAY\_4 (July)

SELECT ID, 'July' AS Payment\_Month, CASE WHEN PAY\_4 >= 2 THEN 1 ELSE 0 END AS Monthly\_Default\_Flag FROM Cleaned\_Data WHERE PAY\_4 IS NOT NULL

UNION ALL

-- PAY\_5 (August)

SELECT ID, 'August' AS Payment\_Month, CASE WHEN PAY\_5 >= 2 THEN 1 ELSE 0 END AS Monthly\_Default\_Flag FROM Cleaned\_Data WHERE PAY\_5 IS NOT NULL

UNION ALL

-- PAY\_6 (September)

SELECT ID, 'September' AS Payment\_Month, CASE WHEN PAY\_6 >= 2 THEN 1 ELSE 0 END AS Monthly\_Default\_Flag FROM Cleaned\_Data WHERE PAY\_6 IS NOT NULL

)

SELECT

mps.Payment\_Month,

CAST(SUM(mps.Monthly\_Default\_Flag) AS DECIMAL(10, 4)) \* 100.0 / COUNT(mps.ID) AS Monthly\_Default\_Rate\_Percentage

FROM

MonthlyPaymentStatus mps

GROUP BY

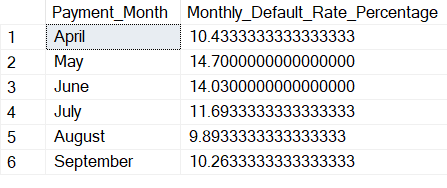
mps.Payment\_Month

ORDER BY

CASE mps.Payment\_Month

WHEN 'April' THEN 1 WHEN 'May' THEN 2 WHEN 'June' THEN 3 WHEN 'July' THEN 4 WHEN 'August' THEN 5 WHEN 'September' THEN 6 ELSE 7

END;



**--KPI 6: Monthly Bill amount & pay amount**

WITH MonthlyAmounts AS (

-- Unpivot BILL\_AMT columns

SELECT ID, 'April' AS Month\_Name, 'Bill Amount' AS Metric\_Type, BILL\_AMT1 AS Amount FROM Cleaned\_Data UNION ALL

SELECT ID, 'May' AS Month\_Name, 'Bill Amount' AS Metric\_Type, BILL\_AMT2 AS Amount FROM Cleaned\_Data UNION ALL

SELECT ID, 'June' AS Month\_Name, 'Bill Amount' AS Metric\_Type, BILL\_AMT3 AS Amount FROM Cleaned\_Data UNION ALL

SELECT ID, 'July' AS Month\_Name, 'Bill Amount' AS Metric\_Type, BILL\_AMT4 AS Amount FROM Cleaned\_Data UNION ALL

SELECT ID, 'August' AS Month\_Name, 'Bill Amount' AS Metric\_Type, BILL\_AMT5 AS Amount FROM Cleaned\_Data UNION ALL

SELECT ID, 'September' AS Month\_Name, 'Bill Amount' AS Metric\_Type, BILL\_AMT6 AS Amount FROM Cleaned\_Data UNION ALL

-- Unpivot PAY\_AMT columns

SELECT ID, 'April' AS Month\_Name, 'Payment Amount' AS Metric\_Type, PAY\_AMT1 AS Amount FROM Cleaned\_Data UNION ALL

SELECT ID, 'May' AS Month\_Name, 'Payment Amount' AS Metric\_Type, PAY\_AMT2 AS Amount FROM Cleaned\_Data UNION ALL

SELECT ID, 'June' AS Month\_Name, 'Payment Amount' AS Metric\_Type, PAY\_AMT3 AS Amount FROM Cleaned\_Data UNION ALL

SELECT ID, 'July' AS Month\_Name, 'Payment Amount' AS Metric\_Type, PAY\_AMT4 AS Amount FROM Cleaned\_Data UNION ALL

SELECT ID, 'August' AS Month\_Name, 'Payment Amount' AS Metric\_Type, PAY\_AMT5 AS Amount FROM Cleaned\_Data UNION ALL

SELECT ID, 'September' AS Month\_Name, 'Payment Amount' AS Metric\_Type, PAY\_AMT6 AS Amount FROM Cleaned\_Data

)

SELECT

ma.Month\_Name,

ma.Metric\_Type,

SUM(ma.Amount) AS Total\_Monthly\_Amount

FROM

MonthlyAmounts ma

GROUP BY

ma.Month\_Name,

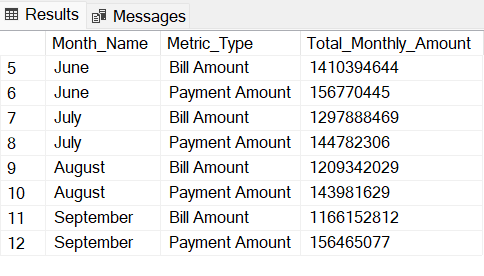
ma.Metric\_Type

ORDER BY

CASE ma.Month\_Name

WHEN 'April' THEN 1 WHEN 'May' THEN 2 WHEN 'June' THEN 3 WHEN 'July' THEN 4 WHEN 'August' THEN 5 WHEN 'September' THEN 6 ELSE 7

END,

ma.Metric\_Type;  
  
****

**B. DEMOGRAPHIC DEFAULT RATES**

**--1. Default rate by Gender (sex)**

select cleaned\_sex,

cast (sum(case when default\_payment\_next\_month = 1 then 1 else 0 end)as decimal (10,4))\*100/COUNT(id)AS Default\_Rate\_Percentage

FROM

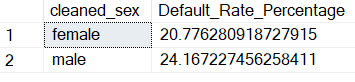
Cleaned\_Data

GROUP BY

cleaned\_sex

ORDER BY

cleaned\_sex;



**-- 2 Default Rate by Age Group**

SELECT

age\_bucket,

CAST(SUM(CASE WHEN default\_payment\_next\_month = 1 THEN 1 ELSE 0 END) AS DECIMAL(10, 4)) \* 100.0 / COUNT(ID) AS Default\_Rate\_Percentage

FROM

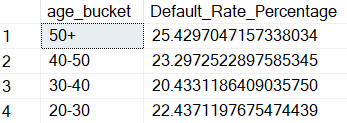
Cleaned\_Data

GROUP BY

age\_bucket

ORDER BY

age\_bucket desc;



**-- 3 Default Rate by Education Segment**

SELECT

Education\_segment,

CAST(SUM(CASE WHEN default\_payment\_next\_month = 1 THEN 1 ELSE 0 END) AS DECIMAL(10, 4)) \* 100.0 / COUNT(ID) AS Default\_Rate\_Percentage

FROM

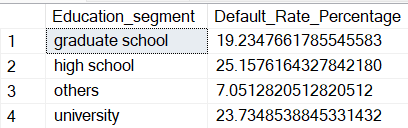
Cleaned\_Data

GROUP BY

EDUCATION\_Segment

ORDER BY

EDUCATION\_Segment;



**-- 4 Default Rate by Marriage Status**

SELECT

Cleaned\_marraige as Marraige,

CAST(SUM(CASE WHEN default\_payment\_next\_month = 1 THEN 1 ELSE 0 END) AS DECIMAL(10, 4)) \* 100.0 / COUNT(ID) AS Default\_Rate\_Percentage

FROM

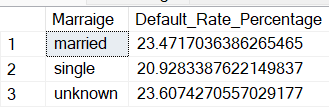
Cleaned\_Data

GROUP BY

cleaned\_marraige

ORDER BY

cleaned\_marraige;

****

**C. BEHAVIORAL DEFAULT RATES**

**--1 Default Rate by Max Delay Segment**

SELECT

MAX\_Delay\_Segment,

CAST(SUM(CASE WHEN default\_payment\_next\_month = 1 THEN 1 ELSE 0 END) AS DECIMAL(10, 4)) \* 100.0 / COUNT(ID) AS Default\_Rate\_Percentage

FROM

Cleaned\_Data

GROUP BY

MAX\_Delay\_Segment

ORDER BY

-- Order by a logical sequence for delay segments if not alphabetical

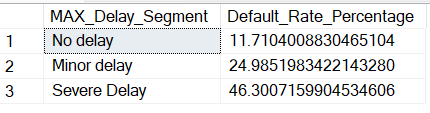
CASE MAX\_Delay\_Segment

WHEN 'No Delay' THEN 1

WHEN 'Minor Delay' THEN 2

else 3

END;



**--2 Default Rate by Delay Trend (delay\_trend\_0\_2)**

SELECT

delay\_trend\_0\_2,

CAST(SUM(CASE WHEN default\_payment\_next\_month = 1 THEN 1 ELSE 0 END) AS DECIMAL(10, 4)) \* 100.0 / COUNT(ID) AS Default\_Rate\_Percentage

FROM

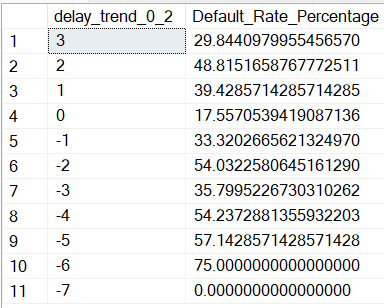
Cleaned\_Data

GROUP BY

delay\_trend\_0\_2

ORDER BY

delay\_trend\_0\_2 desc;



**--3 Default Rate by Utilization Category (Utilization\_Category)**

SELECT

Utilization\_Category,

CAST(SUM(CASE WHEN default\_payment\_next\_month = 1 THEN 1 ELSE 0 END) AS DECIMAL(10, 4)) \* 100.0 / COUNT(ID) AS Default\_Rate\_Percentage

FROM

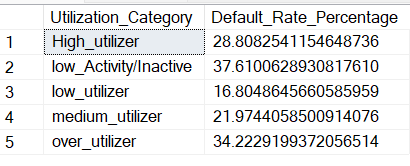
Cleaned\_Data

GROUP BY

Utilization\_Category

ORDER BY

Utilization\_Category ;



**-- 4 Default Rate by Repayment Category**

SELECT

Repayment\_Category,

CAST(SUM(CASE WHEN default\_payment\_next\_month = 1 THEN 1 ELSE 0 END) AS DECIMAL(10, 4)) \* 100.0 / COUNT(ID) AS Default\_Rate\_Percentage

FROM

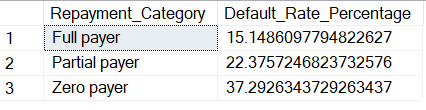
Cleaned\_Data

GROUP BY

Repayment\_Category

ORDER BY

Repayment\_Category;

****