**SQL QUERIES**

**Key Performance Indicators (KPIs) Requirements:**

1. **Total Loan Applications:** We need to calculate the total number of loan applications received during a specified period. Additionally, it is essential to monitor the Month-to-Date (MTD) Loan Applications and track changes Month-over-Month (MoM).

SELECT COUNT(id) AS Total\_Loan\_Applications

FROM bank\_loan\_data

A screenshot of a message

Description automatically generated

MTD

SELECT COUNT(id) AS MTD\_Total\_Loan\_Applications

FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 12 AND YEAR(issue\_date) = 2021

A screenshot of a message box

Description automatically generated

1. **Total Funded Amount:** Understanding the total amount of funds disbursed as loans is crucial. We also want to keep an eye on the MTD Total Funded Amount and analyse the Month-over-Month (MoM) changes in this metric.

SELECT SUM(loan\_amount) AS Total\_Funded\_Amount

FROM bank\_loan\_data

A screenshot of a computer

Description automatically generated

MTD

SELECT SUM(loan\_amount) AS MTD\_Total\_Funded\_Amount

FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 12 AND YEAR(issue\_date) = 2021

A screenshot of a computer

Description automatically generated

1. **Total Amount Received:** Tracking the total amount received from borrowers is essential for assessing the bank's cash flow and loan repayment. We should analyse the Month-to-Date (MTD) Total Amount Received and observe the Month-over-Month (MoM) changes.

SELECT SUM(total\_payment) AS Total\_Amount\_Received

FROM bank\_loan\_data

A screenshot of a computer

Description automatically generated

MTD

SELECT SUM(total\_payment) AS MTD\_Total\_Amount\_Received

FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 12 AND YEAR(issue\_date) = 2021

A screenshot of a message

Description automatically generated

PMTD

SELECT SUM(total\_payment) AS MTD\_Total\_Amount\_Received

FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 11 AND YEAR(issue\_date) = 2021

A screenshot of a computer

Description automatically generated

1. **Average Interest Rate:** Calculating the average interest rate across all loans, MTD, and monitoring the Month-over-Month (MoM) variations in interest rates will provide insights into our lending portfolio's overall cost.

SELECT ROUND(AVG(int\_rate)\*100, 2) AS Avg\_Interest\_Rate

FROM bank\_loan\_data

A screenshot of a computer

Description automatically generated

1. **Average Debt-to-Income Ratio (DTI):** Evaluating the average DTI for our borrowers helps us gauge their financial health. We need to compute the average DTI for all loans, MTD, and track Month-over-Month (MoM) fluctuations.

**Secure Loan KPIs:**

1. **Secure Loan Application Percentage:** We need to calculate the percentage of loan applications classified as 'Secure Loans.' This category includes loans with a loan status of 'Fully Paid' and 'Current.'

SELECT

(COUNT(CASE WHEN loan\_status = 'Fully Paid' OR loan\_status = 'Current' THEN id END)\*100) / COUNT(id) AS Secure\_Loan\_Percentage

FROM bank\_loan\_data

1. **Secure Loan Applications:** Identifying the total number of loan applications falling under the 'Secure Loan' category, which consists of loans with a loan status of 'Fully Paid' and 'Current.'

SELECT COUNT(id) AS Secure\_Loan\_Applications

FROM bank\_loan\_data

WHERE loan\_status = 'Fully Paid' OR loan\_status = 'Current'

1. **Secure Loan Funded Amount:** Determining the total amount of funds disbursed as 'Secure Loans.' This includes the principal amounts of loans with a loan status of 'Fully Paid' and 'Current.'

SELECT SUM(loan\_amount) AS Secure\_Loan\_Funded\_Amount

FROM bank\_loan\_data

WHERE loan\_status = 'Fully Paid' OR loan\_status = 'Current'

1. **Secure Loan Total Received Amount:** Tracking the total amount received from borrowers for 'Secure Loans,' which encompasses all payments made on loans with a loan status of 'Fully Paid' and 'Current.'

SELECT SUM(total\_payment) AS Secure\_Loan\_Received\_Amount

FROM bank\_loan\_data

WHERE loan\_status = 'Fully Paid' OR loan\_status = 'Current'

**Risky Loan KPIs:**

1. **Risky Loan Application Percentage:** Calculating the percentage of loan applications categorized as 'Risky Loans.' This category specifically includes loans with a loan status of 'Charged Off.'

SELECT

(COUNT(CASE WHEN loan\_status = 'Charged off' THEN id END)\*100) / COUNT(id) AS Risky\_Loan\_Percentage

FROM bank\_loan\_data

1. **Risky Loan Applications:** Identifying the total number of loan applications categorized as 'Risky Loans,' which consists of loans with a loan status of 'Charged Off.'

SELECT COUNT(id) AS Risky\_Loan\_Applications

FROM bank\_loan\_data

WHERE loan\_status = 'Charged off'

1. **Risky Loan Funded Amount:** Determining the total amount of funds disbursed as 'Risky Loans.' This comprises the principal amounts of loans with a loan status of 'Charged Off.'

SELECT SUM(loan\_amount) AS Risky\_Loan\_Received\_Amount

FROM bank\_loan\_data

WHERE loan\_status = 'Charged off'

1. **Risky Loan Total Received Amount:** Tracking the total amount received from borrowers for 'Risky Loans,' which includes all payments made on loans with a loan status of 'Charged Off.'

SELECT SUM(total\_payment) AS Risky\_Loan\_Received\_Amount

FROM bank\_loan\_data

WHERE loan\_status = 'Charged off'

SELECT

loan\_status,

COUNT(id) AS Total\_Loan\_Application,

SUM(loan\_amount) AS Total\_Funded\_Amount,

SUM(total\_payment) AS Total\_Amount\_Received,

AVG(int\_rate \* 100) AS Interest\_Rate,

AVG(dti \* 100) AS DTI

FROM bank\_loan\_data

GROUP BY loan\_status

A screenshot of a computer

Description automatically generated

1. **Monthly Trends by Issue Date (Line Chart):**

SELECT

MONTH(issue\_date) AS Month\_Number,

DATENAME(MONTH, issue\_date) AS Month\_Name,

COUNT(id) AS Total\_Loan\_Application,

SUM(loan\_amount) AS Total\_Funded\_Amount,

SUM(total\_payment) AS Total\_Amount\_Received

FROM bank\_loan\_data

GROUP BY MONTH(issue\_date), DATENAME(MONTH, issue\_date)

ORDER BY MONTH(issue\_date)

**A screenshot of a computer

Description automatically generated**

1. **Regional Analysis by State (Filled Map):**

SELECT

address\_state,

COUNT(id) AS Total\_Loan\_Application,

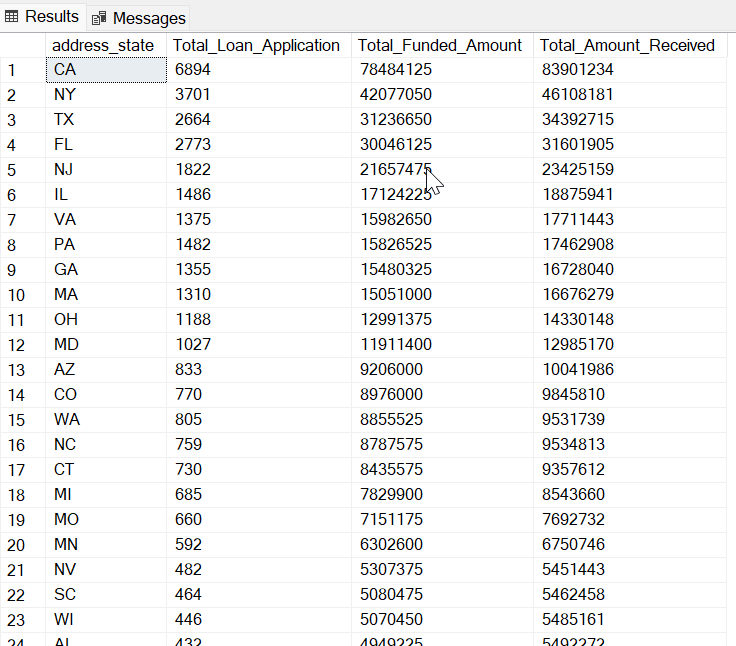
SUM(loan\_amount) AS Total\_Funded\_Amount,

SUM(total\_payment) AS Total\_Amount\_Received

FROM bank\_loan\_data

GROUP BY address\_state

ORDER BY SUM(loan\_amount) DESC



1. **Loan Term Analysis (Donut Chart):**

SELECT

term,

COUNT(id) AS Total\_Loan\_Application,

SUM(loan\_amount) AS Total\_Funded\_Amount,

SUM(total\_payment) AS Total\_Amount\_Received

FROM bank\_loan\_data

GROUP BY term

ORDER BY term DESC

A screenshot of a computer

Description automatically generated

1. **Employee Length Analysis (Bar Chart):**

SELECT

emp\_length,

COUNT(id) AS Total\_Loan\_Application,

SUM(loan\_amount) AS Total\_Funded\_Amount,

SUM(total\_payment) AS Total\_Amount\_Received

FROM bank\_loan\_data

GROUP BY emp\_length

ORDER BY emp\_length DESC

A screenshot of a computer

Description automatically generated

1. **Loan Purpose Breakdown (Bar Chart):**

SELECT

purpose,

COUNT(id) AS Total\_Loan\_Application,

SUM(loan\_amount) AS Total\_Funded\_Amount,

SUM(total\_payment) AS Total\_Amount\_Received

FROM bank\_loan\_data

GROUP BY purpose

ORDER BY Total\_Funded\_Amount DESC

A screenshot of a computer

Description automatically generated

1. **Home Ownership Analysis (Tree Map):**

SELECT

home\_ownership,

COUNT(id) AS Total\_Loan\_Application,

SUM(loan\_amount) AS Total\_Funded\_Amount,

SUM(total\_payment) AS Total\_Amount\_Received

FROM bank\_loan\_data

GROUP BY home\_ownership

ORDER BY Total\_Funded\_Amount DESC

A screenshot of a computer

Description automatically generated