BANK LOAN REPORT QUERY DOCUMENT

BANK LOAN REPORT | SUMMARY

* Dashboard 1

# KPI’s:

## LOAN -> TOTAL, CMTD, MOM

### Total Loan Applications

SELECT COUNT(id) AS Total\_Applications FROM bank\_loan\_data

A close up of a box

Description automatically generated

12 = December and this is the current month  
11= November current-1

### CMTD Loan Applications

SELECT COUNT(id) AS Total\_Applications FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 12

--This to get automatic latest month!!!  
WITH Max\_Issue\_Date AS ( -- Find the most recent issue date SELECT MAX(issue\_date) AS max\_issue\_date FROM bank\_loan\_data )



### PMTD Loan Applications

SELECT COUNT(id) AS Total\_Applications FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 11



Using alias u can now call it by name   
max\_issue\_date

### MOM Loan Applications

MOM = (CMTD-PMTD)/PMTD

WITH Loan\_Applications AS (

SELECT

COUNT(CASE WHEN MONTH(issue\_date) = 12 AND YEAR(issue\_date) = 2021 THEN id END) AS CMTD\_Total\_Loan\_Application,

COUNT(CASE WHEN MONTH(issue\_date) = 11 AND YEAR(issue\_date) = 2021 THEN id END) AS PMTD\_Total\_Loan\_Application

FROM Bank\_loan\_data

)

SELECT

CMTD\_Total\_Loan\_Application,

PMTD\_Total\_Loan\_Application,

(CAST(CMTD\_Total\_Loan\_Application AS DECIMAL(10,2)) - CAST(PMTD\_Total\_Loan\_Application AS DECIMAL(10,2)))

/ NULLIF(CAST(PMTD\_Total\_Loan\_Application AS DECIMAL(10,2)), 0) \* 100 AS MoM\_Percentage\_Change

FROM Loan\_Applications;  


## FUNDED -> TOTAL, CMTD, MOM

### Total Funded Amount

SELECT SUM(loan\_amount) AS Total\_Funded\_Amount FROM bank\_loan\_data



### CMTD Total Funded Amount

SELECT SUM(loan\_amount) AS Total\_Funded\_Amount FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 12



### PMTD Total Funded Amount

SELECT SUM(loan\_amount) AS Total\_Funded\_Amount FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 11



### MOM Funded Applications

WITH Funded\_Amounts AS (

SELECT

SUM(CASE WHEN MONTH(issue\_date) = 12 AND YEAR(issue\_date) = 2021 THEN loan\_amount END) AS CMTD\_Total\_Funded\_Amount,

SUM(CASE WHEN MONTH(issue\_date) = 11 AND YEAR(issue\_date) = 2021 THEN loan\_amount END) AS PMTD\_Total\_Funded\_Amount

FROM bank\_loan\_data

)

SELECT

CMTD\_Total\_Funded\_Amount,

PMTD\_Total\_Funded\_Amount,

(CAST(CMTD\_Total\_Funded\_Amount AS DECIMAL(10,2)) - CAST(PMTD\_Total\_Funded\_Amount AS DECIMAL(10,2)))

/ NULLIF(CAST(PMTD\_Total\_Funded\_Amount AS DECIMAL(10,2)), 0) \* 100 AS MoM\_Percentage\_Change

FROM Funded\_Amounts;  


## Received Amount -> TOTAL, CMTD, MOM

### Total Amount Received

SELECT SUM(total\_payment) AS Total\_Amount\_Collected FROM bank\_loan\_data



### CMTD Total Amount Received

SELECT SUM(total\_payment) AS Total\_Amount\_Collected FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 12



### PMTD Total Amount Received

SELECT SUM(total\_payment) AS Total\_Amount\_Collected FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 11

A close-up of a receipt

Description automatically generated

### MOM Funded Applications

WITH Recived\_Amounts AS (

SELECT

SUM(CASE WHEN MONTH(issue\_date) = 12 AND YEAR(issue\_date) = 2021 THEN total\_payment END) AS CMTD\_Total\_Recived\_Amount,

SUM(CASE WHEN MONTH(issue\_date) = 11 AND YEAR(issue\_date) = 2021 THEN total\_payment END) AS PMTD\_Total\_Recived\_Amount

FROM bank\_loan\_data

)

SELECT

CMTD\_Total\_Recived\_Amount,

PMTD\_Total\_Recived\_Amount,

(CAST(CMTD\_Total\_Recived\_Amount AS DECIMAL(10,2)) - CAST(PMTD\_Total\_Recived\_Amount AS DECIMAL(10,2)))

/ NULLIF(CAST(PMTD\_Total\_Recived\_Amount AS DECIMAL(10,2)), 0) \* 100 AS MoM\_Recived\_Change

FROM Recived\_Amounts;



# Average interest Rate -> Average, CMTD, MOM

### Average Interest Rate

SELECT AVG(int\_rate)\*100 AS Avg\_Int\_Rate FROM bank\_loan\_data  
SELECT AVG(int\_rate)\*100 AS Avg\_Int\_Rate FROM bank\_loan\_data

 

### MTD Average Interest

SELECT AVG(int\_rate)\*100 AS MTD\_Avg\_Int\_Rate FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 12

 

### PMTD Average Interest

SELECT AVG(int\_rate)\*100 AS PMTD\_Avg\_Int\_Rate FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 11

 

### MOM Average Interest

WITH Interest\_Rate AS (

SELECT

AVG(CASE WHEN MONTH(issue\_date) = 12 AND YEAR(issue\_date) = 2021 THEN int\_rate END) AS CMTD\_Interest\_Rate,

AVG(CASE WHEN MONTH(issue\_date) = 11 AND YEAR(issue\_date) = 2021 THEN int\_rate END) AS PMTD\_Interest\_Rate

FROM bank\_loan\_data

)

SELECT

Had to change the size of the Decimal because it display 0.0000

CMTD\_Interest\_Rate,

PMTD\_Interest\_Rate,

(CAST(CMTD\_Interest\_Rate AS DECIMAL(18,10)) - CAST(PMTD\_Interest\_Rate AS DECIMAL(18,10)))

/ NULLIF(CAST(PMTD\_Interest\_Rate AS DECIMAL(18,10)), 0) \* 100 AS MoM\_Interest\_Change

FROM Interest\_Rate;   


WITH Interest\_Rate AS (

SELECT

AVG(CASE WHEN MONTH(issue\_date) = 12 AND YEAR(issue\_date) = 2021 THEN int\_rate END) AS CMTD\_Interest\_Rate,

AVG(CASE WHEN MONTH(issue\_date) = 11 AND YEAR(issue\_date) = 2021 THEN int\_rate END) AS PMTD\_Interest\_Rate

FROM bank\_loan\_data

)

SELECT

ROUND(CMTD\_Interest\_Rate, 2) AS Rounded\_CMTD\_Interest\_Rate,

ROUND(PMTD\_Interest\_Rate, 2) AS Rounded\_PMTD\_Interest\_Rate,

ROUND(

(CAST(CMTD\_Interest\_Rate AS DECIMAL(18,10)) - CAST(PMTD\_Interest\_Rate AS DECIMAL(18,10)))

/ NULLIF(CAST(PMTD\_Interest\_Rate AS DECIMAL(18,10)), 0) \* 100,

2 -- Round to 2 decimal places

) AS Rounded\_MoM\_Interest\_Change

FROM Interest\_Rate;



## Average Debt-to-Income Ratio (DTI)**:**

### Average DTI

SELECT AVG(dti)\*100 AS Avg\_DTI FROM bank\_loan\_data



### MTD Average DTI

SELECT AVG(dti)\*100 AS MTD\_Avg\_DTI FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 12



### PMTD Average DTI

SELECT AVG(dti)\*100 AS PMTD\_Avg\_DTI FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 11

Because DTI is a ratio so we \*100



### MOM Average DTI

WITH Debt\_to\_IncomeRatio AS (

SELECT

AVG(CASE WHEN MONTH(issue\_date) = 12 AND YEAR(issue\_date) = 2021 THEN dti END)\*100 AS CMTD\_dti,

AVG(CASE WHEN MONTH(issue\_date) = 11 AND YEAR(issue\_date) = 2021 THEN dti END)\*100 AS PMTD\_dti

FROM bank\_loan\_data

)

SELECT

CMTD\_dti,

PMTD\_dti,

(CAST(CMTD\_dti AS DECIMAL(18,10)) - CAST(PMTD\_dti AS DECIMAL(18,10)))

/ NULLIF(CAST(PMTD\_dti AS DECIMAL(18,10)), 0) \* 100 AS MoM\_Interest\_Change

FROM Debt\_to\_IncomeRatio;



# Good Loan vs Bad Loan KPI:

## Good\_Loan

### **Good Loan Percentage**

SELECT

    (COUNT(CASE WHEN loan\_status = 'Fully Paid' OR loan\_status = 'Current' THEN id END) \* 100.0) /

COUNT(id) AS Good\_Loan\_Percentage

FROM bank\_loan\_data

**A screenshot of a computer

Description automatically generated**

### **Good Loan Applications**

SELECT COUNT(id) AS Good\_Loan\_Applications FROM bank\_loan\_data

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

****

### **Good Loan Funded Amount**

SELECT SUM(loan\_amount) AS Good\_Loan\_Funded\_amount FROM bank\_loan\_data

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

**A close up of a box

Description automatically generated**

### **Good Loan Amount Received**

SELECT SUM(total\_payment) AS Good\_Loan\_amount\_received FROM bank\_loan\_data

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

**A screenshot of a computer

Description automatically generated**

## Bad\_Loan

### **Bad Loan Percentage**

SELECT

    (COUNT(CASE WHEN loan\_status = 'Charged Off' THEN id END) \* 100.0) /

COUNT(id) AS Bad\_Loan\_Percentage

FROM bank\_loan\_data

****

### **Bad Loan Applications**

SELECT COUNT(id) AS Bad\_Loan\_Applications FROM bank\_loan\_data

WHERE loan\_status = 'Charged Off'

****

### **Bad Loan Funded Amount**

SELECT SUM(loan\_amount) AS Bad\_Loan\_Funded\_amount FROM bank\_loan\_data

WHERE loan\_status = 'Charged Off'

****

### **Bad Loan Amount Received**

SELECT SUM(total\_payment) AS Bad\_Loan\_amount\_received FROM bank\_loan\_data

WHERE loan\_status = 'Charged Off'

****

# Loan Status KPI:

SELECT

        loan\_status,

        COUNT(id) AS LoanCount,

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

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

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

        AVG(dti \* 100) AS DTI

    FROM

        bank\_loan\_data

    GROUP BY

        loan\_status

****

SELECT

loan\_status,

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

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

FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 12

GROUP BY loan\_status

****

# Loan Status Grid View

SELECT

        loan\_status,

        COUNT(id) AS LoanCount,

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

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

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

        AVG(dti \* 100) AS DTI

    FROM

        bank\_loan\_data

    GROUP BY

        loan\_status

****

SELECT

loan\_status,

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

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

FROM bank\_loan\_data

WHERE MONTH(issue\_date) = 12

GROUP BY loan\_status

****

* Dashboard 2

# MONTH

SELECT

MONTH(issue\_date) AS Month\_Munber,

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

COUNT(id) AS Total\_Loan\_Applications,

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 table

Description automatically generated

# STATE

SELECT

address\_state AS State,

COUNT(id) AS Total\_Loan\_Applications,

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

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

FROM bank\_loan\_data

GROUP BY address\_state

ORDER BY address\_state

A screenshot of a computer

Description automatically generated

# TERM

SELECT

term AS Term,

COUNT(id) AS Total\_Loan\_Applications,

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

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

FROM bank\_loan\_data

GROUP BY term

ORDER BY term

A screenshot of a calculator

Description automatically generated

# EMPLOYEE LENGTH

SELECT

emp\_length AS Employee\_Length,

COUNT(id) AS Total\_Loan\_Applications,

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

A screenshot of a computer

Description automatically generated

# PURPOSE

SELECT

purpose AS PURPOSE,

COUNT(id) AS Total\_Loan\_Applications,

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

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

FROM bank\_loan\_data

GROUP BY purpose

ORDER BY purpose

A screenshot of a table

Description automatically generated

# HOME OWNERSHIP

SELECT

home\_ownership AS Home\_Ownership,

COUNT(id) AS Total\_Loan\_Applications,

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

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

FROM bank\_loan\_data

GROUP BY home\_ownership

ORDER BY home\_ownership

A screenshot of a computer

Description automatically generated

*Note: We have applied multiple Filters on all the dashboards. You can check the results for the filters as well by modifying the query and comparing the results.*

*For e.g*

*See the results when we hit the Grade A in the filters for dashboards.*

*SELECT*

*purpose AS PURPOSE,*

*COUNT(id) AS Total\_Loan\_Applications,*

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

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

*FROM bank\_loan\_data*

*WHERE grade = 'A'*

*GROUP BY purpose*

*ORDER BY purpose*