BANK LOAN

1. Total Loan Applications

select count(id) as total\_applications from bank\_loan



1. MTD loan Applications

SELECT

count(id) as Total\_Applications from financial\_loan where

MONTH(STR\_TO\_DATE(issue\_date, '%Y-%m-%d')) = 12 ;



PMTD Loan Applications

SELECT

count(id) as Total\_Applications from financial\_loan where

MONTH(STR\_TO\_DATE(issue\_date, '%Y-%m-%d')) = 11;



1. MTD\_Total Funded Amount

SELECT SUM(loan\_amount) as MTD\_Total\_Funded\_Amount from financial\_loan

WHERE month(str\_to\_date(issue\_date, '%y-%m-%d')) = 12



1. PMTD\_Total Funded Amount

SELECT SUM(loan\_amount) as PMTD\_Total\_Funded\_Amount from financial\_loan

WHERE month(str\_to\_date(issue\_date, '%y-%m-%d')) = 11;



1. Total\_Amount\_Received

SELECT SUM(total\_payment) as Total\_Amount\_Received from financial\_loan



1. MTD\_Total\_Amount\_Received

SELECT SUM(total\_payment) as MTD\_Total\_Amount\_Received from financial\_loan

where month(str\_to\_date(issue\_date, '%y-%m-%d')) = 12



1. PMTD\_Total\_Amount\_Received

SELECT SUM(total\_payment) as PMTD\_Total\_Amount\_Received from financial\_loan

where month(str\_to\_date(issue\_date, '%y-%m-%d')) = 11



1. Average Interest Rate

SELECT ROUND(AVG(int\_rate) , 4) \* 100 as Average\_Interest\_Rate from financial\_loan



1. MTD\_Average\_Interest\_Rate

SELECT ROUND(AVG(int\_rate) , 4) \* 100 as MTD\_Average\_Interest\_Rate from financial\_loan

where month(str\_to\_date(issue\_date, '%y-%m-%d')) = 12



1. PMTD\_Average\_Interest\_Rate

SELECT ROUND(AVG(int\_rate) , 4) \* 100 as PMTD\_Average\_Interest\_Rate from financial\_loan

where month(str\_to\_date(issue\_date, '%y-%m-%d')) = 11



1. Average dti ratio

SELECT round(avg(dti), 4) \*100 as Average\_dti from financial\_loan



1. MTD\_Average dti

SELECT round(avg(dti), 4) \*100 as MTD\_Avg\_dti from financial\_loan

WHERE MONTH(str\_to\_date(issue\_date, '%y-%m-%d')) =12;



1. PMTD Average dti ratio

SELECT round(avg(dti), 4) \*100 as PMTD\_Avg\_dti from financial\_loan

WHERE MONTH(str\_to\_date(issue\_date, '%y-%m-%d')) =11;



1. Good Loan Percentage

select

(count(CASE WHEN loan\_status='Fully Paid' or loan\_status = 'Current' then id end)\*100) /

count(id) as Good\_Loan\_Percentage

from financial\_loan;



1. Good Loan Applications Count

select count(id) as Good\_Loan\_Applications from financial\_loan

where loan\_status ='Fully Paid' OR loan\_status ='Current';



1. Good loan Funded Amount

select sum(loan\_amount) as Good\_Loan\_Funded\_Amount from financial\_loan

where loan\_status in ('Fully Paid','Current');



1. Good loan received amount

select sum(total\_payment) as Good\_Loan\_Received\_Amount from financial\_loan

where loan\_status in ('Fully Paid','Current');



1. Bad Loan Amount Percentage

select

(count(case when loan\_status = 'Charged Off' then id end) \* 100.0) /

count(id) as Bad\_Loan\_Percentage

from financial\_loan;



1. Bad Loan Applications

select count(id) as Bad\_Loan\_Applications from financial\_loan

where loan\_status = 'charged off'



1. Bad loan Funded Amount

select sum(loan\_amount) as Bad\_loan\_funded\_amount from financial\_loan

where loan\_status = 'charged off'



1. Bad loan Amount Received

select sum(total\_payment) as Bad\_loan\_amount\_received from financial\_loan

where loan\_status = 'charged off'



**LOAN STATUS**

select

loan\_status,

count(id) as Total\_Loan\_Applications,

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

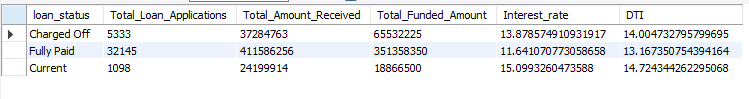
sum(loan\_amount) as Total\_Funded\_Amount,

AVG(int\_rate \* 100) as Interest\_rate,

AVG(dti \* 100) as DTI

FROM financial\_loan

group by loan\_status;



**MTD Level Loan Status :**

select

loan\_status,

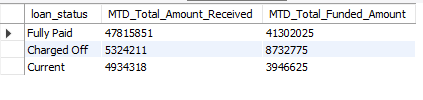
SUM(total\_payment) as MTD\_Total\_Amount\_Received,

sum(loan\_amount) as MTD\_Total\_Funded\_Amount

FROM financial\_loan

WHERE MONTH(str\_to\_date(issue\_date , '%y-%m-%d')) = 12

group by loan\_status;



DASHBOARD -2

MONTHLY Trend:

SELECT

month(str\_to\_date(issue\_date, '%y-%m-%d')) as Month\_Number,

monthname(str\_to\_date(issue\_date, '%y-%m-%d')) as Month\_Name,

count(id) as Total\_Loan\_Applications,

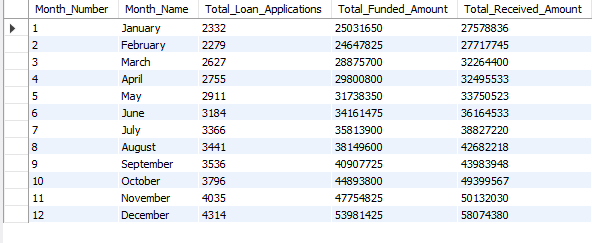
sum(loan\_amount) as Total\_Funded\_Amount,

sum(total\_payment) as Total\_Received\_Amount

from financial\_loan

group by month(str\_to\_date(issue\_date, '%y-%m-%d')) , monthname(str\_to\_date(issue\_date, '%y-%m-%d'))

order by month(str\_to\_date(issue\_date, '%y-%m-%d')) ;



2. **Regional Level Analysis by State :**

SELECT

address\_state,

count(id) as Total\_Loan\_Applications,

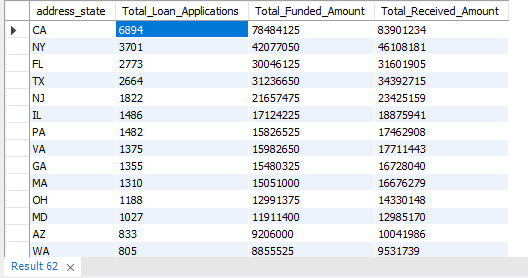
sum(loan\_amount) as Total\_Funded\_Amount,

sum(total\_payment) as Total\_Received\_Amount

from financial\_loan

group by address\_state

order by count(id) desc ;



3**. Loan Term Analysis** :

SELECT

term,

count(id) as Total\_Loan\_Applications,

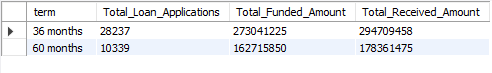
sum(loan\_amount) as Total\_Funded\_Amount,

sum(total\_payment) as Total\_Received\_Amount

from financial\_loan

group by term

order by term ;



4. **Employee Length Analysis:**

SELECT

emp\_length,

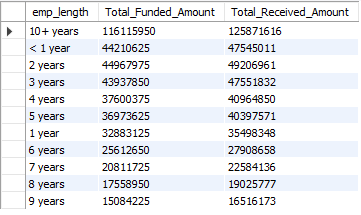
sum(loan\_amount) as Total\_Funded\_Amount,

sum(total\_payment) as Total\_Received\_Amount

from financial\_loan

group by emp\_length

order by count(id) desc;



5. **Purpose wise Analysis**

SELECT

purpose,

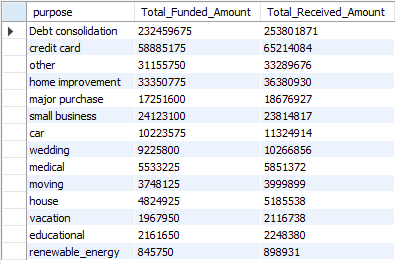
sum(loan\_amount) as Total\_Funded\_Amount,

sum(total\_payment) as Total\_Received\_Amount

from financial\_loan

group by purpose

order by count(id) desc;



Ownership Wise :

SELECT

home\_ownership,

count(id) as Total\_Applications,

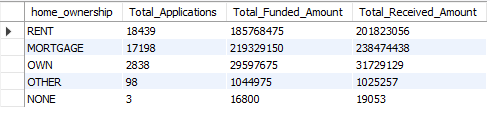
sum(loan\_amount) as Total\_Funded\_Amount,

sum(total\_payment) as Total\_Received\_Amount

from financial\_loan

group by home\_ownership

order by count(id) desc;



Power BI :

**DAX functions 1:**

“PMTD Loan Applications = CALCULATE([Total Loan Applications],DATESMTD(DATEADD('Date Table'[Date],-1,MONTH))”

This DAX (Data Analysis Expressions) function is used to calculate the total loan applications for the current month-to-date (MTD), compared to the previous month. Let's break down the function step by step:

1. \*\*CALCULATE:\*\* This function is used to apply filters to a calculation. It takes an expression and one or more filter arguments. In this case, the expression is `[Total Loan Applications]`, which likely represents a measure that calculates the total number of loan applications.

2. \*\*DATESMTD:\*\* This function returns a table of dates for the current month-to-date (MTD) period. It requires a date column as its argument. In this case, it's applied to the 'Date Table'[Date] column.

3. \*\*DATEADD:\*\* This function is used to add or subtract a specified number of units from a date. Here, it's subtracting 1 month from each date in the 'Date Table'[Date] column. This effectively gives you the previous month.

4. \*\*MONTH:\*\* The MONTH function extracts the month number from a date. It's used here to get the month number of the date one month ago.

5. \*\*Total Loan Applications:\*\* This is likely a measure defined elsewhere in the Power BI model. It calculates the total number of loan applications.

Putting it all together, the DAX expression calculates the total loan applications for the current month-to-date (MTD) period, but compares it with the total loan applications for the previous month's MTD period. It achieves this by subtracting 1 month from each date in the 'Date Table'[Date] column and then calculating the total loan applications within that time frame. This allows for month-over-month comparison of loan applications.

**DAX Function 2:**

“MTD DTI = CALCULATE(TOTALMTD([Average DTI],'Date Table'[Date]))”

1. \*\*CALCULATE:\*\* This function modifies the context in which the calculation is made. It takes an expression and one or more filter arguments.

2. \*\*TOTALMTD:\*\* This function calculates a total for a column across a specified date range, which is Month-to-Date (MTD) in this case. It requires two arguments: the expression to aggregate (in this case, `[Average DTI]`) and the date column to use for the calculation (`'Date Table'[Date]`).

3. \*\*[Average DTI]:\*\* This likely represents a measure that calculates the average Debt-to-Income (DTI) ratio for loan applications. It's the expression being aggregated within the MTD calculation.

4. \*\*'Date Table'[Date]:\*\* This is the date column used to define the date range for the MTD calculation. It's likely a column from a Date Table in your Power BI model.

Putting it all together, the DAX expression calculates the Month-to-Date (MTD) average Debt-to-Income (DTI) ratio for loan applications. It does this by aggregating the `[Average DTI]` measure across the dates in the 'Date Table'[Date] column within the current month-to-date period. This allows you to see the average DTI ratio for loan applications dynamically as the month progresses.