

Plan

[Plan](#)

[Banking Data Analysis](#)

[Key Achievements:](#)

[Tools and Technologies:](#)

[Impact:](#)

[A-SQL: Utilized for querying and data analysis](#)

[BANK LOAN REPORT | SUMMARY](#)

[KPI's:](#)

[Good Loan v Bad Loan KPI's](#)

[Good Loan](#)

[Bad Loan](#)

[Loan status](#)

[BANK LOAN REPORT OVERVIEW](#)

[B-Power BI: Used for data visualization and dashboard creation.](#)

[Key Performance Indicators \(KPIs\) Requirements:](#)

[Date table](#)

[Bad loan vs good loan](#)

[Overview](#)

[Impact of the Project](#)

Banking Data Analysis

The project involved querying the database to track various loan metrics, including total loan applications, funded amounts, and repayments. Key performance indicators (KPIs) were calculated to monitor month-to-date (MTD) and previous month-to-date (PMTD) trends, providing a comprehensive overview of the bank's loan performance.

Key Achievements:

- **Total Loan Applications:** Queried and calculated the total number of loan applications, along with MTD and PMTD applications to observe month-over-month (MoM) changes.
- **Total Funded Amount:** Assessed the total loan amount disbursed, with detailed MTD and PMTD funding amounts for MoM analysis.
- **Total Amount Received:** Monitored repayments received from borrowers, including MTD and PMTD comparisons to track cash flow.
- **Average Interest Rate and Debt-to-Income Ratio:** Evaluated the average interest rate and debt-to-income ratio (DTI) for all loans, incorporating MTD and PMTD metrics.
- **Loan Status Analysis:** Distinguished between good and bad loans by analyzing loan statuses and calculating respective percentages and amounts funded and received.
- **Dashboard Development:** Created a comprehensive dashboard using Power BI to visualize monthly trends, regional analysis, loan terms, employment length impact, and loan purposes.

Tools and Technologies:

- **A-SQL:** Utilized for querying and data analysis.
- **b-Power BI:** Used for data visualization and dashboard creation.

Impact:

This project provided valuable insights into the bank's lending activities, helping to identify trends, regional disparities, and factors influencing loan performance. The analytical dashboard developed can aid in strategic decision-making and improve loan management efficiency.

A-SQL: Utilized for querying and data analysis

BANK LOAN REPORT | SUMMARY

KPI's:

```
select * from [bank-data]
```

TOTAL LOAN APPLICATION

```
select count(id) as totalloanApplication from dbo.[bank-data]
```

totalloanApplication
38576

(MTD) Loan Applications

```
select count(id) as MTD_Total_Loan_Application from dbo.[bank-data]
```

```
WHERE MONTH(issue_date) = 12
```

```
AND YEAR(issue_date) = 2021
```

MTD_Total_Loan_Application
4314

(PMTD) Loan Applications

```
select count(id) as PMTD_Total_Loan_Application from dbo.[bank-data]
```

```
WHERE MONTH(issue_date) = 11
```

```
AND YEAR(issue_date) = 2021
```

Total Funded Amount:

```
SELECT SUM(loan_amount) AS Total_Funded_Amount from [bank-data]
```

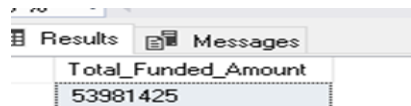
Results	Messages
Total_Funded_Amount	
1	435757075

MTD Total Funded Amount:

```
SELECT SUM(loan_amount) AS Total_Funded_Amount from [bank-data]

WHERE MONTH(issue_date) = 12

AND YEAR(issue_date) = 2021
```



A screenshot of a SQL query results window. The window has two tabs: 'Results' and 'Messages'. The 'Results' tab is active, showing a single row with the column name 'Total_Funded_Amount' and the value '53981425'.

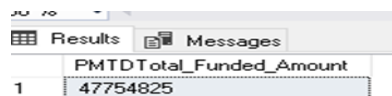
Total_Funded_Amount
53981425

PMTD Total Funded Amount:

```
SELECT SUM(loan_amount) AS PMTDTotal_Funded_Amount from [bank-data]

WHERE MONTH(issue_date) = 11

AND YEAR(issue_date) = 2021
```

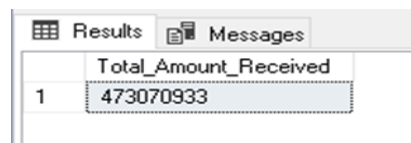


A screenshot of a SQL query results window. The window has two tabs: 'Results' and 'Messages'. The 'Results' tab is active, showing a single row with the column name 'PMTDTotal_Funded_Amount' and the value '47754825'.

PMTDTotal_Funded_Amount
47754825

Total Amount Received:

```
SELECT SUM(total_payment) AS Total_Amount_Received FROM [bank-data]
```



A screenshot of a SQL query results window. The window has two tabs: 'Results' and 'Messages'. The 'Results' tab is active, showing a single row with the column name 'Total_Amount_Received' and the value '473070933'.

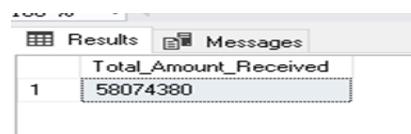
Total_Amount_Received
473070933

MTD Total Amount Received:

```
SELECT SUM(total_payment) AS Total_Amount_Received FROM [bank-data]

WHERE MONTH(issue_date) = 12

AND YEAR(issue_date) = 2021
```



A screenshot of a SQL query results window. The window has two tabs: 'Results' and 'Messages'. The 'Results' tab is active, showing a single row with the column name 'Total_Amount_Received' and the value '58074380'.

Total_Amount_Received
58074380

PMTD Total Amount Received:

```
SELECT SUM(total_payment) AS PMTDTotal_Amount_Received FROM [bank-data]
```

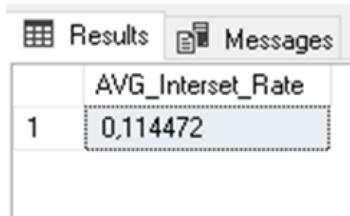
WHERE MONTH(issue_date) = 11

AND YEAR(issue_date) = 2021

Average Interest Rate:

```
select AVG(int_rate)/10000 AS AVG_Interset_Rate from [bank-data]
```

```
select Round(AVG(int_rate), 2)/10000 AS AVG_Interset_Rate from [bank-data]
```



The screenshot shows a SQL Server Results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a single row of data. The column header is 'AVG_Interset_Rate' and the value is '0,114472'.

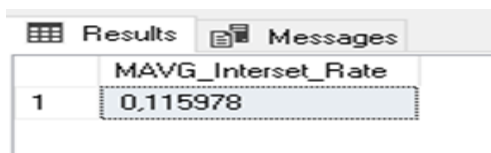
	AVG_Interset_Rate
1	0,114472

MTD Average Interest Rate:

```
select Round(AVG(int_rate), 2)/10000 AS MAVG_Interset_Rate from [bank-data]
```

WHERE MONTH(issue_date) = 12

AND YEAR(issue_date) = 2021



The screenshot shows a SQL Server Results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a single row of data. The column header is 'MAVG_Interset_Rate' and the value is '0,115978'.

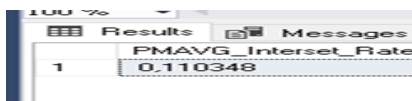
	MAVG_Interset_Rate
1	0,115978

PMTD Average Interest Rate:

```
select Round(AVG(int_rate), 2)/10000 AS PMAVG_Interset_Rate from [bank-data]
```

WHERE MONTH(issue_date) = 11

AND YEAR(issue_date) = 2021



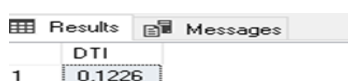
The screenshot shows a SQL Server Results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a single row of data. The column header is 'PMAVG_Interset_Rate' and the value is '0,110348'.

	PMAVG_Interset_Rate
1	0,110348

Average Debt-to-Income Ratio (DTI)

```
select ROUND(dti/ 10000, 4) as DTI from [bank-data]
```

```
select ROUND(avg(dti)/ 10000, 4) as DTI from [bank-data]
```



The screenshot shows a SQL Server Results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a single row of data. The column header is 'DTI' and the value is '0,1226'.

	DTI
1	0,1226

MTD Average Debt-to-Income Ratio (DTI)

```
select ROUND(avg(dti)/ 10000, 4) as mtdDTI from [bank-data]
```

```
WHERE MONTH(issue_date) = 12
```

```
AND YEAR(issue_date) = 2021
```

Results		Messages	
DTI			
1	0,1226		

PMTM Average Debt-to-Income Ratio (DTI)

```
select ROUND(avg(dti)/ 10000, 4) as PMtDDTI from [bank-data]
```

```
WHERE MONTH(issue_date) = 11
```

```
AND YEAR(issue_date) = 2021
```

Results		Messages	
PMtDDTI			
1	0,1189		

Good Loan v Bad Loan KPI's

for good loan is fully charged , current and bad loan is charged off

Good Loan:

1. Good Loan Application Percentage
2. Good Loan Applications
3. Good Loan Funded Amount
4. Good Loan Total Received Amount

Bad Loan:

1. Bad Loan Application Percentage
2. Bad Loan Applications
3. Bad Loan Funded Amount
4. Bad Loan Total Received Amount

Good Loan

1-Good Loan Application Percentage

```
select loan_status from [bank-database].dbo.[bank-data]
```

	loan_status
1	Fully Paid
2	Fully Paid
3	Fully Paid
4	Fully Paid
5	Fully Paid
6	Fully Paid
7	Fully Paid
8	Charged Off
9	Fully Paid
10	Charged Off
11	Fully Paid
12	Fully Paid
13	Fully Paid
14	Fully Paid
15	Fully Paid
16	Fully Paid
17	Fully Paid
18	Fully Paid
19	Fully Paid

SELECT

(COUNT(CASE WHEN loan_status = 'Fully Paid' OR loan_status = 'Current' THEN id END)* 100) /

COUNT(id) AS Good_Loan_Percentage

FROM [bank-database].dbo.[bank-data]

	Good_Loan_Percentage
1	86

2- Good Loan Applications

SELECT COUNT(id) AS Good_Loan_Applications from [bank-database].dbo.[bank-data]
WHERE loan_status = 'Fully Paid' OR loan_status = 'Current'

	Good_Loan_Applications
1	33243

3-good loan funded amount

SELECT sum(loan_amount) AS Good_Loan_funded_amount from [bank-database].dbo.[bank-data]
WHERE loan_status = 'Fully Paid' OR loan_status = 'Current'

	Good_Loan_funded_amount
1	370224850

4-Good Loan Amount Received

SELECT sum(total_payment) AS Good_Loan_received_amount from [bank-database].dbo.[bank-data]
WHERE loan_status = 'Fully Paid' OR loan_status = 'Current'

Results		Messages
	Good_Loan_received_amount	
1	435786170	

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-database].dbo.[bank-data]

Results		Messages
	Bad_Loan_Percentage	
1	13.824657818332	

bad loan Application

select count(id) as bad_loan_app

FROM [bank-database].dbo.[bank-data]

WHERE loan_status = 'charged off'

Results		Messages
	bad_loan_amount	
1	65532225	

Bad loan amount received

select sum(total_payment) as bad_loan_amount_receividd

FROM [bank-database].dbo.[bank-data]

WHERE loan_status = 'charged off'

Results		Messages
	bad_loan_amount_receividd	
1	37284763	

Loan status

In order to gain a comprehensive overview of our lending operations and monitor the performance of loans, we aim to create a grid view report categorized by 'Loan Status.' By providing insights into metrics such as 'Total Loan Applications,' 'Total Funded Amount,' 'Total Amount Received,' 'Month-to-Date (MTD) Funded Amount,' 'MTD Amount Received,' 'Average Interest Rate,' and 'Average Debt-to-Income Ratio (DTI),' this grid view will empower us to make data-driven decisions and assess the health of our loan portfolio.

```
select
loan_status,

Count(id) AS Loanaccount,

Sum(total_payment) as Total_Amount_received,

SUM(loan_amount) AS Total_Amount_Founded,

AVG(int_rate) / 10000 as inetrest_rate,

AVG(dti) / 10000 as DTI

FROM

[bank-database].dbo.[bank-data]

group by

loan_status

order by Loanaccount desc
```

	loan_status	Loanaccount	Total_Amount_received	Total_Amount_Founded	inetrest_rate	DTI
1	Fully Paid	32145	411586256	351358350	0,110478687198631	0,117875047441282
2	Charged Off	5333	37284763	65532225	0,131783030189387	0,123829101818864
3	Current	1098	24199914	18866500	0,147304735883424	0,133435245901639

BANK LOAN REPORT OVERVIEW

1-MONTH

```
select

MONTH(issue_date) AS Month_Number,

Datename(MONTH, issue_date) as Month_Name,

COUNT(id) as Total_Loan_app,

SUM(loan_amount) as Toatal_Amount_Funded,

SUM(total_payment) as Total_Amount_Recived
```

from [bank-database].dbo.[bank-data]

GROUP BY MONTH(issue_date), DATENAME(MONTH, issue_date)

ORDER BY MONTH(issue_date)

Results		Messages			
	Month_Number	Month_Name	Total_Loan_app	Toatal_Amount_Funded	Total_Amount_Recived
1	1	January	2332	25031650	27578836
2	2	February	2279	24647825	27717745
3	3	March	2627	28875700	32264400
4	4	April	2755	29800800	32495533
5	5	May	2911	31738350	33750523
6	6	June	3184	34161475	36164533
7	7	July	3366	35813900	38827220
8	8	August	3441	38149600	42682218
9	9	September	3536	40907725	43983948
10	10	October	3796	44893800	49399567
11	11	November	4035	47754825	50132030
12	12	December	4314	53981425	58074380

2- Regional Analysis

select

address_state,

COUNT(id) as Total_Loan_app,

SUM(loan_amount) as Toatal_Amount_Funded,

SUM(total_payment) as Total_Amount_Recived

from [bank-database].dbo.[bank-data]

GROUP BY address_state

order by address_state

Results		Messages		
	address_state	Total_Loan_app	Toatal_Amount_Funded	Total_Amount_Recived
1	AK	78	1031800	1108570
2	AL	432	4949225	5492272
3	AR	236	2529700	2777875
4	AZ	833	9206000	10041986
5	CA	6894	78484125	83901234
6	CO	770	8976000	9845810
7	CT	730	8435575	9357612
8	DC	214	2652350	2921854
9	DE	110	1138100	1269136
10	FL	2773	30046125	31601905
11	GA	1355	15480325	16728040
12	HI	170	1850525	2080184
13	IA	5	56450	64482
14	ID	6	59750	65329
15	IL	1486	17124225	18875941
16	IN	9	86225	85521
17	KS	260	2872325	3247394
18	KY	320	3504100	3792530
19	LA	426	4498900	5001160
20	MA	1310	15051000	16676279
21	MD	1027	11911400	12985170
22	ME	3	9200	10808
23	MI	685	7829900	8543660
24	MN	592	6302600	6750746
25	MO	660	7151175	7692732
26	MS	19	139125	149342
27	MT	79	829525	892047
28	NC	759	8787575	9534813
29	NE	5	31700	24542

3 --term

SELECT

term,

COUNT(id) as Total_Loan_app,

SUM(loan_amount) as Toatal_Amount_Funded,

SUM(total_payment) as Total_Amount_Recived

from [bank-database].dbo.[bank-data]

GROUP BY term

order by term

Results Messages				
	term	Total_Loan_app	Toatal_Amount_Funded	Total_Amount_Recived
1	36 months	28237	273041225	294709458
2	60 months	10339	162715850	178361475

4 -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-database].dbo.[bank-data]

GROUP BY purpose

ORDER BY purpose

	PURPOSE	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Ri
1	car	1497	10223575	11324914
2	credit card	4998	58885175	65214084
3	Debt consolidation	18214	232459675	253801871
4	educational	315	2161650	2248380
5	home improvement	2876	33350775	36380930
6	house	366	4824925	5185538
7	major purchase	2110	17251600	18676927
8	medical	667	5533225	5851372
9	moving	559	3748125	3999899
10	other	3824	31155750	33289676
11	renewable_energy	94	845750	898931
12	small business	1776	24123100	23814817
13	vacation	352	1967950	2116738
14	wedding	928	9225800	10266856

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-database].dbo.[bank-data]
```

WHERE grade = 'A'

GROUP BY purpose

ORDER BY purpose

Results		Messages		
	PURPOSE	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
1	car	577	3629475	3805538
2	credit card	1353	12636075	13339495
3	Debt consolidation	3753	37216300	38822971
4	educational	79	484000	515639
5	home improvement	933	8359175	8744006
6	house	91	916575	957878
7	major purchase	796	5344575	5604259
8	medical	197	1471850	1526882
9	moving	164	1069450	1099875
10	other	1024	7043175	7397982
11	renewable_energy	29	224150	225827
12	small business	334	3172075	3190467
13	vacation	122	683625	694542
14	wedding	237	2001725	2126202

5- employee length

SELECT

```
emp_length,  
  
COUNT(id) AS Total_Loan_Applications,  
  
SUM(loan_amount) AS Total_Funded_Amount,  
  
SUM(total_payment) AS Total_Amount_Received
```

FROM [bank-database].dbo.[bank-data]

where emp_length = '3 years'

GROUP BY emp_length

ORDER BY emp_length

Results		Messages		
	emp_length	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
	4 years	3428	37600375	40964850
2	5 years	3273	36973625	40397571
3	6 years	2228	25612650	27908658
4	7 years	1772	20811725	22584136
5	8 years	1476	17558950	19025777
6	9 years	1255	15084225	16516173

B-Power BI: Used for data visualization and dashboard creation.

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).
2. **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 analyze the Month-over-Month (MoM) changes in this metric.
3. **Total Amount Received:** Tracking the total amount received from borrowers is essential for assessing the bank's cash flow and loan repayment. We should analyze the Month-to-Date (MTD) Total Amount Received and observe the Month-over-Month (MoM) changes.
4. **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.
5. **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.

Date table

To effectively work with Month-to-Date (MTD), Month-over-Month (MoM), and Prior Month-to-Date (PMTD) metrics, it is essential to utilize time intelligence functions. Therefore, we need to create a date table structured as a calendar.

```
Date_Table = CALENDAR(min('bank-data'[issue_date],MAX('bank-data'[issue_date]))
```

```
Date_Table = CALENDAR(min('bank-data'[issue_date]), MAX('bank-data'[issue_date]))
```

```
Month = FORMAT(Date_Table[Date], "mmm")
```

1-Total Loan Applications:

- YTD Total_Loan_apps = `count('bank-data'[id])`
- MTD Loan_App = `CALCULATE(TOTALMTD([Total_Loan_apps], Date_Table[Date]))`
- PMTD Loan_App = `CALCULATE([Total_Loan_apps], DATESMTD(DATEADD(Date_Table[Date], -1,MONTH)))`
- MOM Loan_App = `([MTD Loan_App]-[PMTD Loan_App]) / [PMTD Loan_App]`

2- Funded Amount:

- Total _funded_Amount = `sum('bank-data'[loan_funded amount])`
- MTD Fundden_AMount = `CALCULATE(TOTALMTD([Total _funded_Amount], Date_Table[Date]))`
- PMTD Total_Funded_Amount = `CALCULATE([Total _funded_Amount], DATESMTD(DATEADD(Date_Table[Date], -1,MONTH)))`
- MOM Total_Funded_amount = `([MTD Fundden_AMount]-[PMTD Total_Funded_Amount]) / [PMTD Total_Funded_Amount]`

3-Total Amount Received

- YTM Total_Amount_Recived = `SUM('bank-data'[total_payment])`
- MTD Received_Amount = `CALCULATE(TOTALMTD([YTM Total_Amount_Recived], Date_Table[Date]))`
- PMTD Total_Received_Amount = `CALCULATE([YTM Total_Amount_Recived], DATESMTD(DATEADD(Date_Table[Date], -1,MONTH)))`
- MOM Total_Received_amount = `([MTD Received_Amount]-[PMTD Total_Received_Amount]) / [PMTD Total_Received_Amount]`

4 -Average Interest RateYTD AVG``_Rate = `AVERAGE('bank-data'[int_rate]) / 10000`

- MTD AVG_Rate = `CALCULATE(TOTALMTD([YTD AVG``_Rate], Date_Table[Date]))`
- PMTD AVG_Rate = `CALCULATE([YTD AVG``_Rate], DATESMTD(DATEADD(Date_Table[Date], -1,MONTH)))`
- MOM AVG_Rate = `([MTD AVG_Rate]-[PMTD AVG_Rate]) / [PMTD AVG_Rate]`

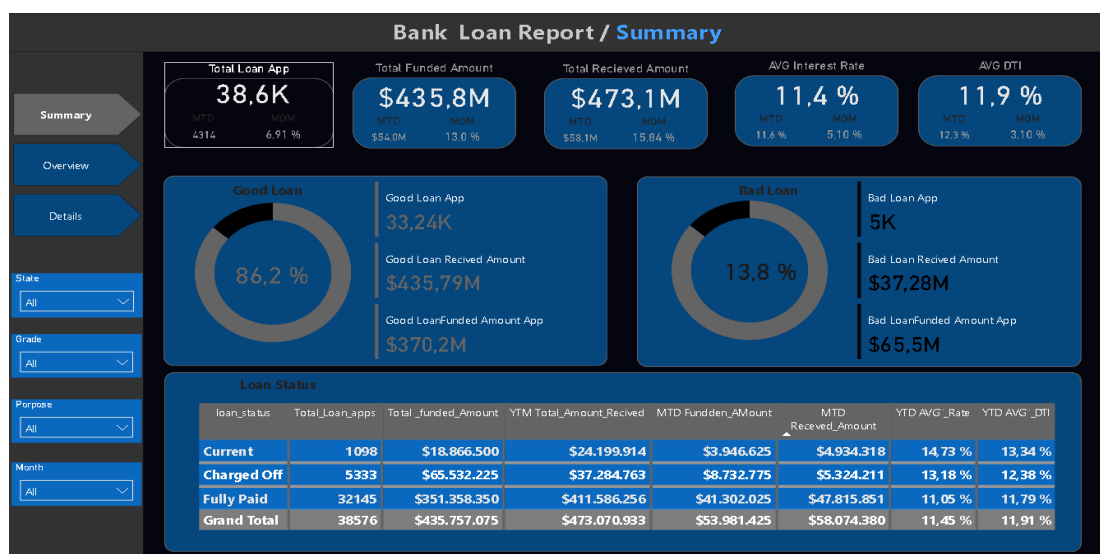
5 Average Debt-to-Income Ratio (DTI

- YTD AVG``_DTI = `AVERAGE('bank-data'[dti]) / 10000`
- MTD AVG_DTI = `CALCULATE(TOTALMTD([YTD AVG``_DTI], Date_Table[Date]))`

- PMTD AVG_DTI = $\text{CALCULATE}([\text{YTD AVG_DTI}], \text{DATESMTD}(\text{DATEADD}(\text{Date_Table}[\text{Date}], -1, \text{MONTH})))$
- MOM AVG_DTI = $([\text{MTD AVG_DTI}] - [\text{PMTD AVG_DTI}]) / [\text{PMTD AVG_DTI}]$

Bad lon vs good loan

- Good Loan % = $(\text{CALCULATE}([\text{Total_Loan_apps}], 'bank-data'[\text{Good vs Bad Loan}] = \text{"Good Loan"})) / [\text{Total_Loan_apps}]$
- Good Loan Funded Amount App = $\text{CALCULATE}([\text{Total_funded_Amount}], 'bank-data'[\text{Good vs Bad Loan}] = \text{"Good Loan"})$
- Good Loan Recived Amount = $\text{CALCULATE}([\text{YTM Total_Amount_Recived}], 'bank-data'[\text{Good vs Bad Loan}] = \text{"Good Loan"})$
- Good Loan Funded Amount App = $\text{CALCULATE}([\text{Total_funded_Amount}], 'bank-data'[\text{Good vs Bad Loan}] = \text{"Good Loan"})$
- Bad Loan % = $(\text{CALCULATE}([\text{Total_Loan_apps}], 'bank-data'[\text{Good vs Bad Loan}] = \text{"Bad Loan"})) / [\text{Total_Loan_apps}]$
- Bad Loan App = $\text{CALCULATE}([\text{Total_Loan_apps}], 'bank-data'[\text{Good vs Bad Loan}] = \text{"Bad Loan"})$
- Bad Loan Recived Amount = $\text{CALCULATE}([\text{YTM Total_Amount_Recived}], 'bank-data'[\text{Good vs Bad Loan}] = \text{"Bad Loan"})$
- Bad Loan Funded Amount App = $\text{CALCULATE}([\text{Total_funded_Amount}], 'bank-data'[\text{Good vs Bad Loan}] = \text{"Bad Loan"})$



Overview

CHARTS

1. Monthly Trends by Issue Date (Line Chart): To identify seasonality and long-term trends in lending activities
2. Regional Analysis by State (Filled Map): To identify regions with significant lending activity and assess regional disparities
3. Loan Term Analysis (Donut Chart): To allow the client to understand the distribution of loans across various term lengths.
4. Employee Length Analysis (Bar Chart): How lending metrics are distributed among borrowers with different employment lengths, helping us assess the impact of employment history on loan applications.
5. Loan Purpose Breakdown (Bar Chart): Will provide a visual breakdown of loan metrics based on the stated purposes of loans, aiding in the understanding of the primary reasons borrowers seek financing.
6. Home Ownership Analysis (Tree Map): For a hierarchical view of how home ownership impacts loan applications and disbursements.



Impact of the Project

The project significantly improved the ability to monitor and analyze the bank's loan performance. The insights gained have equipped me with the skills to:

- Perform detailed data analysis and derive meaningful conclusions.
- Use visualization tools to present complex data in an easily understandable manner.
- Make data-driven recommendations to improve loan management and strategic decision-making.

Overall, this project has been an enriching experience, enhancing my technical skills in SQL and Power BI, and deepening my understanding of banking analytics. It has prepared me to take on more complex data analysis tasks and contribute effectively to financial data-driven projects.