Bank Loan Report Terminologies

- **Loan ID**: Unique identifier for each loan application, used by banks to track and manage loans, repayments, and customer inquiries.

- **Address State**: Indicates borrower’s location, helping banks assess regional risk, comply with regulations, and analyze loan demand by geography.

- **Employment Length**: Reflects job stability; banks use this to assess repayment ability, as stable employment reduces default risk.

- **Employee Title**: Borrower’s job title; helps banks understand income sources, financial capacity, and tailor loan offers.

- **Grade**: Risk classification based on creditworthiness; higher grades indicate lower risk, guiding loan pricing.

- **Sub Grade**: Further differentiates risk within each grade, allowing banks to adjust rates to borrower risk profiles.

- **Home Ownership**: Borrower's housing status, used to assess financial stability and potential collateral.

- **Issue Date**: Loan origination date, essential for tracking, interest accrual, and portfolio management.

- **Last Credit Pull Date**: Last date the borrower’s credit was reviewed, aiding banks in credit risk assessment.

- **Last Payment Date**: Date of the most recent payment, used to track payment behavior and delinquency.

- **Loan Status**: Current loan state (e.g., fully paid, current, default), indicating loan performance and provisioning needs.

- **Next Payment Date**: Scheduled next payment date, assisting banks in cash flow and revenue projections.

- **Purpose**: Loan's reason (e.g., debt consolidation), used to customize loan terms and segment borrowers.

- **Term**: Duration in months, defining the repayment period, aiding in structuring agreements and payment scheduling.

- **Verification Status**: Indicates verification of financial information, essential for data accuracy and application credibility.

- **Annual Income**: Borrower’s yearly income, used to assess loan eligibility, DTI ratio, and creditworthiness.

- **DTI (Debt-to-Income Ratio)**: Measures debt against income, helping banks assess loan payment capacity.

- **Instalment**: Monthly repayment amount, calculated for affordability and structuring loan terms.

- **Interest Rate**: Annual borrowing cost as a percentage; helps banks price loans and manage margins.

- **Loan Amount**: Total borrowed principal; defines loan size and helps determine lending terms.

Each term provides essential insights for banks to evaluate risk, customize loan offerings, and effectively manage portfolios.

BANK LOAN REPORT QUERY DOCUMENT

A. BANK LOAN REPORT | SUMMARY

KPI's:

Total Loan Applications

```
SELECT COUNT(id) AS Total_Applications FROM bank_loan_data;
```

Total_Applications
38576

MTD Loan Applications

```
SELECT COUNT(id) AS Total_Applications FROM bank_loan_data
WHERE MONTH(issue_date) = 12 AND YEAR(issue_date) = 2021;
```

MTD_Total_Loan_Applications
4314

PMTD Loan Applications

```
SELECT COUNT(id) AS PMTD_Total_Applications FROM bank_loan_data
WHERE MONTH(issue_date) = 11 AND YEAR(issue_date) = 2021;
```

PMTD_Total_Applications
4035

Total Funded Amount

```
SELECT SUM(loan_amount) AS Total_Funded_Amount FROM bank_loan_data;
```

Total_Funded_Amount
435757075

MTD Total Funded Amount

```
SELECT SUM(loan_amount) AS MTD_Total_Funded_Amount FROM bank_loan_data
WHERE MONTH(issue_date) = 12 AND YEAR(issue_date) = 2021;
```

MTD_Total_Funded_Amount
53981425

PMTD Total Funded Amount

```
SELECT SUM(loan_amount) AS PMTD_Total_Funded_Amount FROM bank_loan_data
WHERE MONTH(issue_date) = 11 AND YEAR(issue_date) = 2021;
```

Total Amount Received

```
SELECT SUM(total_payment) AS Total_Amount_Collected FROM bank_loan_data;
```

Total_Amount_Collected
473070933

MTD Total Amount Received

```
SELECT SUM(total_payment) AS MTD_Total_Amount_Collected FROM bank_loan_data
WHERE MONTH(issue_date) = 12 AND YEAR(issue_date) = 2021;
```

MTD_Total_Amount_Collected
58074380

PMTD Total Amount Received

```
SELECT SUM(total_payment) AS PMTD_Total_Amount_Collected FROM bank_loan_data
WHERE MONTH(issue_date) = 11 AND YEAR(issue_date) = 2021;
```

PMTD_Total_Amount_Collected
50132030

Average Interest Rate

```
SELECT ROUND(AVG(int_rate)*100,2) AS Avg_Int_Rate FROM bank_loan_data;
```

Avg_Int_Rate
12.05

MTD Average Interest

```
SELECT ROUND(AVG(int_rate)*100,2) AS MTD_Avg_Int_Rate FROM bank_loan_data
WHERE MONTH(issue_date) = 12 AND YEAR(issue_date) = 2021;
```

MTD_Avg_Int_Rate
12.36

PMTD Average Interest

```
SELECT ROUND(AVG(int_rate)*100,2) AS PMTD_Avg_Int_Rate FROM bank_loan_data
WHERE MONTH(issue_date) = 11 AND YEAR(issue_date) = 2021;
```

PMTD_Avg_Int_Rate
11.94

PMTD_Total_Funded_Amount
47754825

Avg DTI

```
SELECT ROUND(AVG(dti)*100,2) AS Avg_DTI FROM bank_loan_data;
```

Avg_DTI
13.33

MTD Avg DTI

```
SELECT ROUND(AVG(dti)*100,2) AS MTD_Avg_DTI FROM bank_loan_data
WHERE MONTH(issue_date) = 12 AND YEAR(issue_date) = 2021;
```

MTD_Avg_DTI
13.67

PMTD Avg DTI

```
SELECT ROUND(AVG(dti)*100,2) AS PMTD_Avg_DTI FROM bank_loan_data
WHERE MONTH(issue_date) = 11 AND YEAR(issue_date) = 2021;
```

PMTD_Avg_DTI
13.3

GOOD LOAN ISSUED

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 bank_loan_data
```

Good_Loan_Percentage
86

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_Applications
33243

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'
```

Good_Loan_Funded_amount
370224850

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'
```

Good_Loan_amount_received
435786170

- - CALCULATE THE BANK PROFIT PERCENTAGE

BANK PROFIT PERCENTAGE

```
WITH Good_Loan_CTE AS (
    SELECT SUM(loan_amount) AS Good_Loan_Funded_amount
    FROM bank_loan_data
    WHERE loan_status = 'Fully Paid' OR loan_status = 'Current'
),
Good_Loan_Amount_Received_CTE AS (
    SELECT SUM(total_payment) AS Good_Loan_amount_received
    FROM bank_loan_data
    WHERE loan_status = 'Fully Paid' OR loan_status = 'Current'
)

SELECT
    ROUND(
        ((Good_Loan_Amount_Received_CTE.Good_Loan_amount_received -
Good_Loan_CTE.Good_Loan_Funded_amount) * 100.0)
        / Good_Loan_CTE.Good_Loan_Funded_amount, 2
    ) AS Good_Loan_Percentage
FROM
    Good_Loan_CTE,
    Good_Loan_Amount_Received_CTE;
```

Bank_Profit_Percentage
17.710000000000

BAD LOAN ISSUED

Bad Loan Percentage

```
SELECT
    (COUNT(CASE WHEN loan_status = 'Charged Off' THEN id END) * 100) /
       COUNT(id) AS Bad_Loan_Percentage
FROM bank_loan_data
```

Bad_Loan_Percentage
13

Bad Loan Applications

```
SELECT COUNT(id) AS Bad_Loan_Applications FROM bank_loan_data
WHERE loan_status = 'Charged Off'
```

Bad_Loan_Applications
5333

Bad Loan Funded Amount

```
SELECT SUM(loan_amount) AS Bad_Loan_Funded_amount FROM bank_loan_data
WHERE loan_status = 'Charged Off'
```

Bad_Loan_Funded_amount
65532225

Bad Loan Amount Received

```
SELECT SUM(total_payment) AS Bad_Loan_amount_received FROM bank_loan_data
WHERE loan_status = 'Charged Off'
```

Bad_Loan_amount_received
37284763

- - CALCULATE THE BANK LOSS PERCENTAGE

BANK LOSS PERCENTAGE

```
WITH Bad_Loan_Funded AS (
    SELECT SUM(loan_amount) AS Bad_Loan_Funded_amount
       FROM bank_loan_data
       WHERE loan_status = 'Charged Off'),
Bad_Loan_Received_CTE AS (
    SELECT SUM(total_payment) AS Bad_Loan_amount_received
       FROM bank_loan_data
       WHERE loan_status = 'Charged Off')
SELECT
    ROUND(
        ((Bad_Loan_Funded.Bad_Loan_Funded_amount -
Bad_Loan_Received_CTE.Bad_Loan_amount_received) * 100.0)
        / Bad_Loan_Funded.Bad_Loan_Funded_amount, 2
    ) AS Bank_Loss_Percentage
FROM
    Bad_Loan_Funded,
    Bad_Loan_Received_CTE;
```

Bank_Loss_Percentage
43.100000000000

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
 bank_loan_data
GROUP BY
 loan_status
```

	loan_status	Total_Loan_Applications	Total_Amount_Received	Total_Funded_Amount	Interest_Rate	DTI
1	Fully Paid	32145	411586256	351358350	11.6410707918092	13.1673507557434
2	Charged Off	5333	37284763	65532225	13.8785749318289	14.0047328005517
3	Current	1098	24199914	18866500	15.0993260800947	14.7243442736843

```
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 AND YEAR(issue_date) = 2021

GROUP BY loan_status;
```

loan_status	MTD_Total_Amount_Received	MTD_Total_Funded_Amount
Fully Paid	47815851	41302025
Charged Off	5324211	8732775
Current	4934318	3946625

B. BANK LOAN REPORT | OVERVIEW

MONTH

```
SELECT
 MONTH(issue_date) AS Month_Number,
 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);
```

	Month_Number	Month_name	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
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

```sql
SELECT
        MONTH(issue_date) AS Month_Number,
        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 COUNT(id) DESC;
```

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

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;
```

	State	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
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
30	NH	161	1917900	2101386
31	NJ	1822	21657475	23425159
32	NM	183	1916775	2084485
33	NV	482	5307375	5451443
34	NY	3701	42077050	46108181
35	OH	1188	12991375	14330148
36	OK	293	3365725	3712649
37	OR	436	4720150	4966903
38	PA	1482	15826525	17462908
39	RI	196	1883025	2001774

```
SELECT TOP 10
        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 COUNT(id) DESC;
```

	State	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
1	CA	6894	78484125	83901234
2	NY	3701	42077050	46108181
3	FL	2773	30046125	31601905
4	TX	2664	31236650	34392715
5	NJ	1822	21657475	23425159
6	IL	1486	17124225	18875941
7	PA	1482	15826525	17462908
8	VA	1375	15982650	17711443
9	GA	1355	15480325	16728040
10	MA	1310	15051000	16676279

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;
```

	Term	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
1	36 months	28237	273041225	294709458
2	60 months	10339	162715850	178361475

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 COUNT(id) DESC;
```

	Employee_Length	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
1	10+ years	8870	116115950	125871616
2	< 1 year	4575	44210625	47545011
3	2 years	4382	44967975	49206961
4	3 years	4088	43937850	47551832
5	4 years	3428	37600375	40964850
6	5 years	3273	36973625	40397571
7	1 year	3229	32883125	35498348
8	6 years	2228	25612650	27908658
9	7 years	1772	20811725	22584136
10	8 years	1476	17558950	19025777
11	9 years	1255	15084225	16516173

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 Total_Loan_Applications DESC;
```

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

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 Total_Loan_Applications DESC; --which categories applies most
```

	Home_Ownership	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
1	RENT	18439	185768475	201823056
2	MORTGAGE	17198	219329150	238474438
3	OWN	2838	29597675	31729129
4	OTHER	98	1044975	1025257
5	NONE	3	16800	19053

```
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 Total_Funded_Amount DESC;-- who recieves the maximum loan
```

	Home_Ownership	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
1	MORTGAGE	17198	219329150	238474438
2	RENT	18439	185768475	201823056
3	OWN	2838	29597675	31729129
4	OTHER	98	1044975	1025257
5	NONE	3	16800	19053

```
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 Total_Amount_Received DESC; -- who repays the maximum amount
```

	Home_Ownership	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
1	MORTGAGE	17198	219329150	238474438
2	RENT	18439	185768475	201823056
3	OWN	2838	29597675	31729129
4	OTHER	98	1044975	1025257
5	NONE	3	16800	19053

```sql
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
```

DAX QUERY USED

1. Total_Loan_Applications = COUNT(bank_loan_data[id])
2. Total Funded Amount = SUM(bank_loan_data[loan_amount])
3. Total Recieved Amount = SUM(bank_loan_data[total_payment])
4. MTD Loan Applications = CALCULATE(TOTALMTD([Total_Loan_Applications], 'Date Table'[Date]))
5. PMTD Loan Applications = CALCULATE([Total_Loan_Applications], DATESMTD(DATEADD('Date Table'[Date],-1,MONTH)))
6. MOM Total Loan Application = ([MTD Total Loan Applications]-[PMTD Total Loan Applications])/ [PMTD Total Loan Applications]
7. MTD Total Funded Amount = CALCULATE(TOTALMTD([Total Funded Amount], 'Date Table'[Date]))
8. PMTD Total Funded Amount = CALCULATE([MTD Total Funded Amount], DATESMTD(DATEADD('Date Table'[Date],-1,MONTH)))
9. MTD Total Received Amount = CALCULATE(TOTALMTD([Total Recieved Amount], 'Date Table'[Date]))
10. PMTD Total Received Amount = CALCULATE([MTD Total Received Amount], DATESMTD(DATEADD('Date Table'[Date],-1,MONTH)))
11. MTD Total Received Amount = CALCULATE(TOTALMTD([Total Recieved Amount], 'Date Table'[Date]))
12. MOM Total Received Amount = ([MTD Total Received Amount]-[PMTD Total Received Amount])/ [PMTD Total Received Amount]

```
13. Average Interest Rate = AVERAGE(bank_loan_data[int_rate])


14. MTD Average Interest Rate = CALCULATE(TOTALMTD([Average Interest Rate],
 'Date Table'[Date]))


15. PMTD Average Interest Rate = CALCULATE([MTD Average Interest Rate],
 DATESMTD(DATEADD('Date Table'[Date],-1,MONTH)))


16. MOM Average Interest Rate = ([MTD Average Interest Rate]-[PMTD Average
 Interest Rate])/ [PMTD Average Interest Rate]

17. Average DTI = AVERAGE(bank_loan_data[dti])


18. MTD Average DTI = CALCULATE(TOTALMTD([Average DTI], 'Date Table'[Date]))

19. PMTD Average DTI = CALCULATE([MTD Average DTI], DATESMTD(DATEADD('Date
 Table'[Date],-1,MONTH)))


20. MOM Average DTI = ([MTD Average DTI]-[PMTD Average DTI])/ [PMTD Average DTI]

21. Good Loan Applications = CALCULATE([Total_Loan_Applications],
 bank_loan_data[Good vs Bad Loan] = "Good Loan")

22. Good Loan Funded Amount = CALCULATE([Total Funded Amount],
 bank_loan_data[Good vs Bad Loan] = "Good Loan")

23. Good Loan Received Amount = CALCULATE([Total Recieved Amount],
 bank_loan_data[Good vs Bad Loan] = "Good Loan")

24. Good Loan % = (CALCULATE([Total_Loan_Applications], bank_loan_data[Good vs
 Bad Loan] = "Good Loan"))/ [Total_Loan_Applications]

25. Bad Loan Applications = CALCULATE([Total_Loan_Applications],
 bank_loan_data[Good vs Bad Loan] = "Bad Loan")

26. Bad Loan Funded Amount = CALCULATE([Total Funded Amount],
 bank_loan_data[Good vs Bad Loan] = "Bad Loan")

27. Bad Loan Received Amount = CALCULATE([Total Recieved Amount],
 bank_loan_data[Good vs Bad Loan] = "Bad Loan")

28. Bad Loan % = (CALCULATE([Total_Loan_Applications], bank_loan_data[Good vs
 Bad Loan] = "Bad Loan"))/ [Total_Loan_Applications]
```