BANK LOAN REPORT QUERY DOCUMENT

A. BANK LOAN REPORT | SUMMARY

KPI's:

Total Loan Applications

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

	total_loan_applications bigint
1	38576

MTD Loan Applications

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

	mtd_total_loan_applications bigint
1	4314

PMTD Loan Applications

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

	pmtd_total_loan_applications bigint
1	4035

Month-over-Month (MoM) loan applications (i.e., the percentage change from one month to the next)

```
WITH

mtd AS (

    SELECT COUNT(id) AS total

    FROM bank_loan_data

    WHERE EXTRACT(MONTH FROM issue_date) = 12

      AND EXTRACT(YEAR FROM issue_date) = 2021

),

pmtd AS (

    SELECT COUNT(id) AS total

    FROM bank_loan_data

    WHERE EXTRACT(MONTH FROM issue_date) = 11

      AND EXTRACT(YEAR FROM issue_date) = 2021

)
```

```sql
SELECT
    mtd.total AS MTD_total_loan_applications,
    pmtd.total AS PMTD_total_loan_applications,
    ROUND(
      CASE
        WHEN pmtd.total = 0 THEN NULL -- or 0, or your logic for no previous month
        ELSE ((mtd.total::numeric - pmtd.total::numeric) / pmtd.total::numeric) * 100
      END, 2
    ) AS MoM_Percentage_Change
FROM mtd, pmtd;
```

	mtd_total_loan_applications bigint	pmtd_total_loan_applications bigint	mom_percentage_change numeric
1	4314	4035	6.91

Total Funded Amount

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

	total_funded_amount bigint
1	435757075

Month-over-Month (MoM) funded amount (i.e., the percentage change from one month to the next)

```sql
WITH
mtd AS (
    SELECT SUM(loan_amount) AS total
    FROM bank_loan_data
    WHERE EXTRACT(MONTH FROM issue_date) = 12
      AND EXTRACT(YEAR FROM issue_date) = 2021
),
pmtd AS (
    SELECT SUM(loan_amount) AS total
    FROM bank_loan_data
    WHERE EXTRACT(MONTH FROM issue_date) = 11
      AND EXTRACT(YEAR FROM issue_date) = 2021
)
```

```
SELECT
    mtd.total AS MTD_total_funded_amount,
    pmtd.total AS PMTD_total_funded_amount,
    ROUND(
      CASE
        WHEN pmtd.total = 0 THEN NULL -- or 0, or your logic for no previous month
        ELSE ((mtd.total::numeric - pmtd.total::numeric) / pmtd.total::numeric) * 100
      END, 2
    ) AS MoM_Percentage_Change
FROM mtd, pmtd;
```

	mtd_total_funded_amount bigint	pmtd_total_funded_amount bigint	mom_percentage_change numeric
1	53981425	47754825	13.04

Total Amount Received

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

	total_amount_collected bigint
1	473070933

Month-over-Month (MoM) amount received (i.e., the percentage change from one month to the next)

```
WITH
mtd AS (
    SELECT SUM(total_payment) AS total
    FROM bank_loan_data
    WHERE EXTRACT(MONTH FROM issue_date) = 12
      AND EXTRACT(YEAR FROM issue_date) = 2021
),
pmtd AS (
    SELECT SUM(total_payment) AS total
    FROM bank_loan_data
    WHERE EXTRACT(MONTH FROM issue_date) = 11
      AND EXTRACT(YEAR FROM issue_date) = 2021
)
```

```sql
SELECT
    mtd.total AS MTD_Total_Amount_Collected,
    pmtd.total AS PMTD_Total_Amount_Collected,
    ROUND(
      CASE
        WHEN pmtd.total = 0 THEN NULL -- or 0, or your logic for no previous month
        ELSE ((mtd.total::numeric - pmtd.total::numeric) / pmtd.total::numeric) * 100
      END, 2
    ) AS MoM_Percentage_Change
FROM mtd, pmtd;
```

	mtd_total_amount_collected bigint	pmtd_total_amount_collected bigint	mom_percentage_change numeric
1	58074380	50132030	15.84

Average Interest Rate

```sql
SELECT AVG(int_rate)*100 AS Avg_Interest_Rate FROM bank_loan_data;
```

	avg_interest_rate double precision
1	12.048831417204827

MoM (%) Change in Average Interest Rate

```sql
WITH
mtd AS (
    SELECT AVG(int_rate) AS Average
    FROM bank_loan_data
    WHERE EXTRACT(MONTH FROM issue_date) = 12
      AND EXTRACT(YEAR FROM issue_date) = 2021
),
pmtd AS (
    SELECT AVG(int_rate) AS Average
    FROM bank_loan_data
    WHERE EXTRACT(MONTH FROM issue_date) = 11
      AND EXTRACT(YEAR FROM issue_date) = 2021
)
```

```
SELECT
    mtd.Average AS MTD_Avg_Int_Rate,
    pmtd.Average AS PMTD_Avg_Int_Rate,
    ROUND(
      CASE
        WHEN pmtd.Average = 0 THEN NULL -- or 0, or your logic for no previous month
        ELSE ((mtd.Average::numeric - pmtd.Average::numeric) / pmtd.Average::numeric) *
100
      END, 2
    ) AS MoM_Percentage_Change
FROM mtd, pmtd;
```

	mtd_avg_int_rate double precision	pmtd_avg_int_rate double precision	mom_percentage_change numeric
1	12.356040867604229	11.941717549826132	3.47

Avg DTI

```
SELECT AVG(dti)*100 AS Avg_DTI FROM bank_loan_data
```

	avg_dti double precision
1	13.327433119037751

MoM (%) Change in Average DTI

```
WITH
mtd AS (
    SELECT AVG(dti) AS AvgDTI
    FROM bank_loan_data
    WHERE EXTRACT(MONTH FROM issue_date) = 12
      AND EXTRACT(YEAR FROM issue_date) = 2021
),
pmtd AS (
    SELECT AVG(dti) AS AvgDTI
    FROM bank_loan_data
    WHERE EXTRACT(MONTH FROM issue_date) = 11
      AND EXTRACT(YEAR FROM issue_date) = 2021
)
```

```
SELECT
    mtd.AvgDTI AS MTD_Avg_DTI,
    pmtd.AvgDTI AS PMTD_Avg_DTI,
    ROUND(
      CASE
        WHEN pmtd.AvgDTI = 0 THEN NULL -- or 0, or your logic for no previous month
        ELSE ((mtd.AvgDTI::numeric - pmtd.AvgDTI::numeric) / pmtd.AvgDTI::numeric) * 100
      END, 2
    ) AS MoM_Percentage_Change
FROM mtd, pmtd;
```

	mtd_avg_dti double precision	pmtd_avg_dti double precision	mom_percentage_change numeric
1	13.66553778395922	13.302733581164853	2.73

GOOD LOAN ISSUED

Good Loan Percentage

```
SELECT (COUNT(CASE WHEN loan_status IN ('Fully Paid', 'Current') THEN id
END)*100.0)/COUNT(id) AS Good_Loan_Percentage FROM bank_loan_data;
```

	good_loan_percentage numeric
1	86.1753421816673579

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 bigint
1	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 bigint
1	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 bigint
1	435786170

BAD LOAN ISSUED

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_percentage numeric
1	13.8246578183326421

Bad Loan Applications

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

	bad_loan_applications bigint
1	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 bigint
1	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 bigint
1	37284763

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 character varying (50)	total_loan_applications bigint	total_amount_received bigint	total_funded_amount bigint	interest_rate double precision	dti double precision
1	Current	1098	24199914	18866500	15.099326080094704	14.724344262295068
2	Fully Paid	32145	411586256	351358350	11.64107079180918	13.167350754394162
3	Charged Off	5333	37284763	65532225	13.878574931828878	14.004732795799695

```
SELECT
	loan_status,
	SUM(total_payment) AS MTD_Total_Amount_Received,
	SUM(loan_amount) AS MTD_Total_Funded_Amount
FROM bank_loan_data
WHERE EXTRACT(MONTH FROM issue_date) = 12 AND EXTRACT(YEAR FROM issue_date) = 2021
GROUP BY loan_status
```

	loan_status character varying (50)	mtd_total_amount_received bigint	mtd_total_funded_amount bigint
1	Charged Off	5324211	8732775
2	Current	4934318	3946625
3	Fully Paid	47815851	41302025

A. BANK LOAN REPORT | OVERVIEW

1. Monthly Trends by Issue Date

```
SELECT
	EXTRACT(MONTH FROM issue_date) AS Month_Number,
	TO_CHAR(issue_date, 'Month') AS Month_name,
	COUNT(id) AS Total_Loan_Applications,
	SUM(loan_amount) AS Total_Funded_Amount,
	SUM(total_payment) AS Total_Received_Amount
FROM bank_loan_data
GROUP BY EXTRACT(MONTH FROM issue_date), TO_CHAR(issue_date, 'Month')
ORDER BY EXTRACT(MONTH FROM issue_date)
```

	month_number numeric	month_name text	total_loan_applications bigint	total_funded_amount bigint	total_received_amount bigint
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 by 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 character varying (50)	total_loan_applications bigint	total_funded_amount bigint	total_amount_received bigint
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

	state character varying (50)	total_loan_applications bigint	total_funded_amount bigint	total_amount_received bigint
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
40	SC	464	5080475	5462458
41	SD	63	606150	656514
42	TN	17	162175	141522
43	TX	2664	31236650	34392715
44	UT	252	2849225	2952412
45	VA	1375	15982650	17711443
46	VT	54	504100	534973
47	WA	805	8855525	9531739
48	WI	446	5070450	5485161
49	WV	167	1830525	1991936
50	WY	79	890750	1046050

3. Loan Term Analysis

```
SELECT

	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 character varying (50)	total_loan_applications bigint	total_funded_amount bigint	total_amount_received bigint
1	36 months	28237	273041225	294709458
2	60 months	10339	162715850	178361475

4. Employee Length Analysis

```
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
```

	employee_length character varying (50)	total_loan_applications bigint	total_funded_amount bigint	total_amount_received bigint
1	< 1 year	4575	44210625	47545011
2	1 year	3229	32883125	35498348
3	10+ years	8870	116115950	125871616
4	2 years	4382	44967975	49206961
5	3 years	4088	43937850	47551832
6	4 years	3428	37600375	40964850
7	5 years	3273	36973625	40397571
8	6 years	2228	25612650	27908658
9	7 years	1772	20811725	22584136
10	8 years	1476	17558950	19025777
11	9 years	1255	15084225	16516173

5. Loan Purpose Breakdown

```
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
```

	purpose character varying (50)	total_loan_applications bigint	total_funded_amount bigint	total_amount_received bigint
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

6. Home Ownership Analysis

```
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
```

	home_ownership character varying (50)	total_loan_applications bigint	total_funded_amount bigint	total_amount_received bigint
1	MORTGAGE	17198	219329150	238474438
2	NONE	3	16800	19053
3	OTHER	98	1044975	1025257
4	OWN	2838	29597675	31729129
5	RENT	18439	185768475	201823056