BANK LOAN REPORT QUERY DOCUMENT

A. BANK LOAN REPORT | SUMMARY

KPI's:

Total Loan Applications

```
SELECT COUNT(id) AS Total_Applications FROM bank
```

Total_Applications
38576

MTD Loan Applications

```
SELECT COUNT(id) AS mtd_total_applications_latest_month

FROM public.bank

WHERE issue_date >= DATE_TRUNC('month', (SELECT MAX(issue_date) FROM public.bank))

  AND issue_date <= (SELECT MAX(issue_date) FROM public.bank)
```

	mtd_total_applications_latest_month bigint
1	4314

Month-Over-Month (MoM)

```
WITH MonthlyApplications AS (
    SELECT
        EXTRACT(YEAR FROM issue_date) AS app_year,
        EXTRACT(MONTH FROM issue_date) AS app_month,
        COUNT(id) AS current_month_applications,
        LAG(COUNT(id), 1, 0) OVER (
            PARTITION BY EXTRACT(YEAR FROM issue_date)
            ORDER BY EXTRACT(MONTH FROM issue_date)
        ) AS previous_month_applications
    FROM
        public.bank
    GROUP BY
        EXTRACT(YEAR FROM issue_date),
        EXTRACT(MONTH FROM issue_date)
    ORDER BY
        EXTRACT(YEAR FROM issue_date),
        EXTRACT(MONTH FROM issue_date)
)
SELECT
    app_year,
    app_month,
```

```
    current_month_applications,
    previous_month_applications,
    (current_month_applications - previous_month_applications) AS absolute_change,
    CASE
        WHEN previous_month_applications = 0 THEN NULL
        ELSE ((current_month_applications - previous_month_applications)::NUMERIC /
previous_month_applications) * 100
    END AS "percentage change"
FROM
    MonthlyApplications
ORDER BY
    app_year,
    app_month;
```

	app_year numeric	app_month numeric	current_month_applications bigint	previous_month_applications bigint	absolute_change bigint	percentage change numeric
1	2021	1	2332	0	2332	[null]
2	2021	2	2279	2332	-53	-2.27272727272727272700
3	2021	3	2627	2279	348	15.26985519964896884600
4	2021	4	2755	2627	128	4.87247811191473163300
5	2021	5	2911	2755	156	5.66243194192377495500
6	2021	6	3184	2911	273	9.37822054276880797000
7	2021	7	3366	3184	182	5.71608040201005025100
8	2021	8	3441	3366	75	2.22816399286987522300
9	2021	9	3536	3441	95	2.76082534147050276100
10	2021	10	3796	3536	260	7.35294117647058823500
11	2021	11	4035	3796	239	6.29610115911485774500
12	2021	12	4314	4035	279	6.91449814126394052000

Total Funded Amount

```
SELECT SUM(loan_amount) AS Total_Funded_Amount FROM bank
```

Total_Funded_Amount
435757075

MTD Total Funded Amount

```
SELECT SUM(loan_amount) AS mtd_funded_amount

FROM public.bank

WHERE issue_date >= DATE_TRUNC('month', (SELECT MAX(issue_date) FROM public.bank))

  AND issue_date <= (SELECT MAX(issue_date) FROM public.bank);
```

	mtd_funded_amount numeric
1	53981425.00

Month-Over-Month (MoM)

```
WITH MonthlyFunds AS (
    SELECT
        EXTRACT(YEAR FROM issue_date) AS fund_year,
        EXTRACT(MONTH FROM issue_date) AS fund_month,
        SUM(loan_amount) AS current_month_funds
    FROM public.bank
    WHERE EXTRACT(YEAR FROM issue_date) = 2021
    GROUP BY 1, 2
    ORDER BY 1, 2
)
SELECT
    fund_year,
    fund_month,
    current_month_funds,
    LAG(current_month_funds, 1, 0) OVER (ORDER BY fund_year, fund_month) AS
previous_month_funds,
    (current_month_funds - LAG(current_month_funds, 1, 0) OVER (ORDER BY fund_year,
fund_month)) AS absolute_change_funds,
    CASE
        WHEN LAG(current_month_funds, 1, 0) OVER (ORDER BY fund_year, fund_month) > 0
        THEN (current_month_funds - LAG(current_month_funds, 1, 0) OVER (ORDER BY
fund_year, fund_month)) * 100.0 / LAG(current_month_funds, 1, 0) OVER (ORDER BY
fund_year, fund_month)
        ELSE NULL
 END AS percentage_change_funds
FROM MonthlyFunds;
```

	fund_year numeric	fund_month numeric	current_month_funds numeric	previous_month_funds numeric	absolute_change_funds numeric	percentage_change_funds numeric
1	2021	1	25031650.00	0	25031650.00	[null]
2	2021	2	24647825.00	25031650.00	-383825.00	-1.5333587677999652
3	2021	3	28875700.00	24647825.00	4227875.00	17.1531362300730389
4	2021	4	29800800.00	28875700.00	925100.00	3.2037318575826733
5	2021	5	31738350.00	29800800.00	1937550.00	6.5016710960779576
6	2021	6	34161475.00	31738350.00	2423125.00	7.6346911543920840
7	2021	7	35813900.00	34161475.00	1652425.00	4.8371008570326662
8	2021	8	38149600.00	35813900.00	2335700.00	6.5217694805648086
9	2021	9	40907725.00	38149600.00	2758125.00	7.2297612556881330
10	2021	10	44893800.00	40907725.00	3986075.00	9.7440642323668696
11	2021	11	47754825.00	44893800.00	2861025.00	6.3728733143552116
12	2021	12	53981425.00	47754825.00	6226600.00	13.0386824786814735

Total Amount Received

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

Total_Amount_Collected
473070933

MTD Total Amount Received

```
SELECT SUM(total_payment) AS mtd_amount_received
FROM public.bank
WHERE issue_date >= DATE_TRUNC('month', (SELECT MAX(issue_date) FROM public.bank))
  AND issue_date <= (SELECT MAX(issue_date) FROM public.bank);
```

	mtd_amount_received numeric
1	58074380.00

Month-Over-Month (MoM)

```
WITH MonthlyPayments AS (
    SELECT
        EXTRACT(YEAR FROM issue_date) AS payment_year,
        EXTRACT(MONTH FROM issue_date) AS payment_month,
        SUM(total_payment) AS current_month_total_payment
    FROM public.bank
    WHERE EXTRACT(YEAR FROM issue_date) = 2021
    GROUP BY 1, 2
    ORDER BY 1, 2
)
SELECT
    payment_year,
    payment_month,
    current_month_total_payment,
    LAG(current_month_total_payment, 1, 0) OVER (ORDER BY payment_year, payment_month)
AS previous_month_total_payment,
```

```
    (current_month_total_payment - LAG(current_month_total_payment, 1, 0) OVER (ORDER
BY payment_year, payment_month)) AS absolute_change_total_payment,

    CASE

        WHEN LAG(current_month_total_payment, 1, 0) OVER (ORDER BY payment_year,
payment_month) > 0

        THEN (current_month_total_payment - LAG(current_month_total_payment, 1, 0)
OVER (ORDER BY payment_year, payment_month)) * 100.0 /
LAG(current_month_total_payment, 1, 0) OVER (ORDER BY payment_year, payment_month)

        ELSE NULL

    END AS percentage_change_total_payment

FROM MonthlyPayments;
```

	payment_year numeric	payment_month numeric	current_month_total_payment numeric	previous_month_total_payment numeric	absolute_change_total_payment numeric	percentage_change_total_payment numeric
1	2021	1	27578836.00	0	27578836.00	[null]
2	2021	2	27717745.00	27578836.00	138909.00	0.5036797056989642 3475
3	2021	3	32264400.00	27717745.00	4546655.00	16.4034087188550151
4	2021	4	32495533.00	32264400.00	231133.00	0.7163716046168532 5002
5	2021	5	33750523.00	32495533.00	1254990.00	3.8620385146475363
6	2021	6	36164533.00	33750523.00	2414010.00	7.1525113847865409
7	2021	7	38827220.00	36164533.00	2662687.00	7.3627025682870010
8	2021	8	42682218.00	38827220.00	3854998.00	9.9285964846311428
9	2021	9	43983948.00	42682218.00	1301730.00	3.0498180764645361
10	2021	10	49399567.00	43983948.00	5415619.00	12.3127169029937922
11	2021	11	50132030.00	49399567.00	732463.00	1.4827316198945630
12	2021	12	58074380.00	50132030.00	7942350.00	15.8428653298101034

Average Interest Rate

```
SELECT AVG(int_rate)*100 AS Avg_Int_Rate FROM bank
```

Avg_Int_Rate
12.0488314172048

MTD Average Interest

```
SELECT ROUND(AVG(int_rate) * 100, 4) AS MTD_Avg_Int_Rate

FROM public.bank
```

```
WHERE issue_date >= DATE_TRUNC('month', (SELECT MAX(issue_date) FROM public.bank))
  AND issue_date <= (SELECT MAX(issue_date) FROM public.bank);
```

MTD_Avg_Int_Rate
12.3560408676042

Month-Over-Month (MoM)

```
WITH MonthlyAvgRate AS (
    SELECT
        EXTRACT(YEAR FROM issue_date) AS rate_year,
        EXTRACT(MONTH FROM issue_date) AS rate_month,
        ROUND(AVG(int_rate) * 100, 4) AS current_month_avg_rate
    FROM public.bank
    WHERE EXTRACT(YEAR FROM issue_date) = 2021
    GROUP BY 1, 2
    ORDER BY 1, 2
)
SELECT
    rate_year,
    rate_month,
    current_month_avg_rate,
    LAG(current_month_avg_rate, 1) OVER (ORDER BY rate_year, rate_month) AS
previous_month_avg_rate,
    (current_month_avg_rate - LAG(current_month_avg_rate, 1) OVER (ORDER BY rate_year,
rate_month)) AS absolute_change_avg_rate,
    CASE
        WHEN LAG(current_month_avg_rate, 1) OVER (ORDER BY rate_year, rate_month) IS
NOT NULL AND LAG(current_month_avg_rate, 1) OVER (ORDER BY rate_year, rate_month) <> 0
        THEN (current_month_avg_rate - LAG(current_month_avg_rate, 1) OVER (ORDER BY
rate_year, rate_month)) * 100.0 / LAG(current_month_avg_rate, 1) OVER (ORDER BY
rate_year, rate_month)
        ELSE NULL
    END AS percentage_change_avg_rate
FROM MonthlyAvgRate;
```

	rate_year numeric	rate_month numeric	current_month_avg_rate numeric	previous_month_avg_rate numeric	absolute_change_avg_rate numeric	percentage_change_avg_rate numeric
1	2021	1	11.4619	[null]	[null]	[null]
2	2021	2	11.7216	11.4619	0.2597	2.2657674556574390
3	2021	3	11.8583	11.7216	0.1367	1.1662230412230412
4	2021	4	11.7409	11.8583	-0.1174	-0.99002386514087179444
5	2021	5	12.2578	11.7409	0.5169	4.4025585772811284
6	2021	6	12.2742	12.2578	0.0164	0.13379236078252214916
7	2021	7	12.2372	12.2742	-0.0370	-0.30144530804451613954
8	2021	8	12.3002	12.2372	0.0630	0.51482365246951917105
9	2021	9	12.0032	12.3002	-0.2970	-2.4145948846360222
10	2021	10	12.0241	12.0032	0.0209	0.17412023460410557185
11	2021	11	11.9417	12.0241	-0.0824	-0.68529037516321387879
12	2021	12	12.3560	11.9417	0.4143	3.4693552844234908

Avg DTI

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

Avg_DTI
13.3274331211432

MTD Avg DTI

```
SELECT (AVG(dti) * 100),  AS MTD_Avg_DTI
FROM public.bank
WHERE issue_date >= DATE_TRUNC('month', (SELECT MAX(issue_date) FROM public.bank))
  AND issue_date <= (SELECT MAX(issue_date) FROM public.bank);
```

MTD_Avg_DTI
13.6655377880425

Month-Over-Month (MoM)

```
WITH MonthlyAvgDTI AS (
    SELECT
        EXTRACT(YEAR FROM issue_date) AS dti_year,
        EXTRACT(MONTH FROM issue_date) AS dti_month,
        ROUND(AVG(dti) * 100, 4) AS current_month_avg_dti
    FROM public.bank
    WHERE EXTRACT(YEAR FROM issue_date) = 2021
    GROUP BY 1, 2
    ORDER BY 1, 2
)
SELECT
    dti_year,
    dti_month,
    current_month_avg_dti,
    LAG(current_month_avg_dti, 1) OVER (ORDER BY dti_year, dti_month) AS
previous_month_avg_dti,
    (current_month_avg_dti - LAG(current_month_avg_dti, 1) OVER (ORDER BY dti_year,
dti_month)) AS absolute_change_avg_dti,
    CASE
        WHEN LAG(current_month_avg_dti, 1) OVER (ORDER BY dti_year, dti_month) IS NOT
NULL AND LAG(current_month_avg_dti, 1) OVER (ORDER BY dti_year, dti_month) <> 0
        THEN (current_month_avg_dti - LAG(current_month_avg_dti, 1) OVER (ORDER BY
dti_year, dti_month)) * 100.0 / LAG(current_month_avg_dti, 1) OVER (ORDER BY dti_year,
dti_month)
        ELSE NULL
    END AS percentage_change_avg_dti
FROM MonthlyAvgDTI;
```

	dti_year numeric	dti_month numeric	current_month_avg_dti numeric	previous_month_avg_dti numeric	absolute_change_avg_dti numeric	percentage_change_avg_dti numeric
1	2021	1	12.9370	[null]	[null]	[null]
2	2021	2	13.4093	12.9370	0.4723	3.6507691118497333
3	2021	3	13.2156	13.4093	-0.1937	-1.4445198481650795
4	2021	4	13.2194	13.2156	0.0038	0.02875389690971276370
5	2021	5	13.3337	13.2194	0.1143	0.86463833456889117509
6	2021	6	13.2438	13.3337	-0.0899	-0.67423145863488754059
7	2021	7	13.2948	13.2438	0.0510	0.38508585149277397726
8	2021	8	13.3532	13.2948	0.0584	0.43926948882269759605
9	2021	9	13.2978	13.3532	-0.0554	-0.41488182607914207830
10	2021	10	13.4144	13.2978	0.1166	0.87683676999202875664
11	2021	11	13.3027	13.4144	-0.1117	-0.83268726145038167939
12	2021	12	13.6655	13.3027	0.3628	2.7272658933900637

GOOD LOAN ISSUED

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
```

Good_Loan_Percentage
86.175342181667

Good Loan Applications

```
SELECT COUNT(id) AS Good_Loan_Applications FROM bank
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
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
WHERE loan_status = 'Fully Paid' OR loan_status = 'Current'
```

Good_Loan_amount_received
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
```

Bad_Loan_Percentage
13.824657818332

Bad Loan Applications

```
SELECT COUNT(id) AS Bad_Loan_Applications FROM bank
```

```
WHERE loan_status = 'Charged Off'
```

Bad_Loan_Applications
5333

Bad Loan Funded Amount

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

Bad_Loan_amount_received
37284763

LOAN STATUS

```
    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
GROUP BY
    loan_status
```

	loan_status	LoanCount	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

WHERE issue_date >= DATE_TRUNC('month', (SELECT MAX(issue_date) FROM public.bank))
```

```
  AND issue_date <= (SELECT MAX(issue_date) FROM public.bank)

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_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
GROUP BY MONTH(issue_date), DATENAME(MONTH, issue_date)
ORDER BY MONTH(issue_date)
```

	Month_Munber	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

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
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

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
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
GROUP BY emp_length
ORDER BY emp_length
```

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

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

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
GROUP BY home_ownership
ORDER BY home_ownership
```

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