KPI's

1. Total Loan Applications

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

	total_loan_applications bigint
1	38576

Month-to-Date Loan Applications

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

	mtd_total_loan_applications bigint
1	4314

Previous Month-to-Date Loan Applications

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

	pmtd_total_loan_applications bigint
1	4035

Month-to-Month Changes

```
SELECT
    EXTRACT(YEAR FROM issue_date) AS year,
    EXTRACT(MONTH FROM issue_date) AS month,
    COUNT(id) AS total_loan_applications,
    LAG(COUNT(id)) OVER (ORDER BY EXTRACT(YEAR FROM issue_date), EXTRACT(MONTH FROM issue_date)) AS previous_month_applications,
    COUNT(id) - LAG(COUNT(id)) OVER (ORDER BY EXTRACT(YEAR FROM issue_date), EXTRACT(MONTH FROM issue_date)) AS MoM_change
FROM loans
GROUP BY year, month
ORDER BY year, month;
```

	year numeric	month numeric	total_loan_applications bigint	previous_month_applications bigint	mom_change bigint
1	2021	1	2332	[null]	[null]
2	2021	2	2279	2332	-53
3	2021	3	2627	2279	348
4	2021	4	2755	2627	128
5	2021	5	2911	2755	156
6	2021	6	3184	2911	273
7	2021	7	3366	3184	182
8	2021	8	3441	3366	75
9	2021	9	3536	3441	95
10	2021	10	3796	3536	260
11	2021	11	4035	3796	239
12	2021	12	4314	4035	279

2. Total Funded Amount

```
SELECT SUM(loan_amount) AS total_funded_amount
FROM loans;
```

	total_funded_amount double precision
1	435757075

Month-to-Date

```
SELECT SUM(loan_amount) AS MTD_total_funded_amount
FROM loans
WHERE EXTRACT(MONTH FROM issue_date) = 12
    AND EXTRACT(YEAR FROM issue_date) = 2021;
```

	mtd_total_funded_amount double precision
1	53981425

Month-to-Month Changes

```
SELECT
    EXTRACT(YEAR FROM issue_date) AS year,
    EXTRACT(MONTH FROM issue_date) AS month,
    SUM(loan_amount) AS total_loan_amount,
    LAG(SUM(loan_amount)) OVER (ORDER BY EXTRACT(YEAR FROM issue_date), EXTRACT(MONTH FROM issue_date)) AS previous_month_amount,
    SUM(loan_amount) - LAG(SUM(loan_amount)) OVER (ORDER BY EXTRACT(YEAR FROM issue_date), EXTRACT(MONTH FROM issue_date)) AS MoM_change
FROM loans
GROUP BY year, month
ORDER BY year, month;
```

	year numeric	month numeric	total_loan_amount double precision	previous_month_amount double precision	mom_change double precision
1	2021	1	25031650	[null]	[null]
2	2021	2	24647825	25031650	-383825
3	2021	3	28875700	24647825	4227875
4	2021	4	29800800	28875700	925100
5	2021	5	31738350	29800800	1937550
6	2021	6	34161475	31738350	2423125
7	2021	7	35813900	34161475	1652425
8	2021	8	38149600	35813900	2335700
9	2021	9	40907725	38149600	2758125
10	2021	10	44893800	40907725	3986075
11	2021	11	47754825	44893800	2861025
12	2021	12	53981425	47754825	6226600

3. Total Amount Received

```
SELECT SUM(total_payment) as total_amount_received
FROM loans;
```

	total_amount_received double precision
1	473070933

Month-to-Date

```
SELECT SUM(total_payment) AS MTD_total_amount_received
FROM loans
WHERE EXTRACT(MONTH FROM issue_date) = 12
    AND EXTRACT(YEAR FROM issue_date) = 2021;
```

	mtd_total_amount_received double precision
1	58074380

Month-to-Month Changes

```
SELECT
    EXTRACT(YEAR FROM issue_date) AS year,
    EXTRACT(MONTH FROM issue_date) AS month,
    SUM(loan_amount) as total_amount_received,
    LAG(SUM(loan_amount)) OVER (ORDER BY EXTRACT(YEAR FROM issue_date), EXTRACT(MONTH FROM issue_date)) AS previous_month_amount_received,
    SUM(loan_amount) - LAG(SUM(loan_amount)) OVER (ORDER BY EXTRACT(YEAR FROM issue_date), EXTRACT(MONTH FROM issue_date)) AS MoM_change
FROM loans
GROUP BY year, month
ORDER BY year, month;
```

	year numeric	month numeric	total_amount_received double precision	previous_month_amount_received double precision	mom_change double precision
1	2021	1	25031650	[null]	[null]
2	2021	2	24647825	25031650	-383825
3	2021	3	28875700	24647825	4227875
4	2021	4	29800800	28875700	925100
5	2021	5	31738350	29800800	1937550
6	2021	6	34161475	31738350	2423125
7	2021	7	35813900	34161475	1652425
8	2021	8	38149600	35813900	2335700
9	2021	9	40907725	38149600	2758125
10	2021	10	44893800	40907725	3986075
11	2021	11	47754825	44893800	2861025
12	2021	12	53981425	47754825	6226600

4. Average Interest Rate

```
SELECT ROUND(AVG(int_rate)::numeric, 4) * 100 AS avg_int_rate
FROM loans;
```

	avg_int_rate numeric
1	12.0500

Month-to-Date

```
SELECT ROUND(AVG(int_rate)::numeric, 4) * 100 AS MTD_avg_int_rate
FROM loans
WHERE EXTRACT(MONTH FROM issue_date) = 12
    AND EXTRACT(YEAR FROM issue_date) = 2021;
```

MoM Changes

```
SELECT
    EXTRACT(YEAR FROM issue_date) AS year,
    EXTRACT(MONTH FROM issue_date) AS month,
    ROUND(AVG(int_rate)::numeric, 4) * 100 AS avg_int_rate,
    LAG(ROUND(AVG(int_rate)::numeric, 4) * 100) OVER (ORDER BY EXTRACT(YEAR FROM issue_date), EXTRACT(MONTH FROM issue_date)) AS previous_month_int_rate,
    ROUND(AVG(int_rate)::numeric, 4) * 100 - LAG(ROUND(AVG(int_rate)::numeric, 4) * 100) OVER (ORDER BY EXTRACT(YEAR FROM issue_date), EXTRACT(MONTH FROM issue_date)) AS MoM_change
FROM loans
GROUP BY year, month
ORDER BY year, month;
```

	year numeric	month numeric	avg_int_rate numeric	previous_month_int_rate numeric	mom_change numeric
1	2021	1	11.4600	[null]	[null]
2	2021	2	11.7200	11.4600	0.2600
3	2021	3	11.8600	11.7200	0.1400
4	2021	4	11.7400	11.8600	-0.1200
5	2021	5	12.2600	11.7400	0.5200
6	2021	6	12.2700	12.2600	0.0100
7	2021	7	12.2400	12.2700	-0.0300
8	2021	8	12.3000	12.2400	0.0600
9	2021	9	12.0000	12.3000	-0.3000
10	2021	10	12.0200	12.0000	0.0200
11	2021	11	11.9400	12.0200	-0.0800
12	2021	12	12.3600	11.9400	0.4200

5. Average Debt-to-Income (DTI) Ratio

```
SELECT ROUND(AVG(dti)::numeric, 4) * 100 AS avg_dti
FROM loans;
```

	avg_dti numeric
1	13.3300

Month-to-Date

```
SELECT ROUND(AVG(dti)::numeric, 4) * 100 AS MTD_avg_dti
FROM loans
WHERE EXTRACT(MONTH FROM issue_date) = 12
    AND EXTRACT(YEAR FROM issue_date) = 2021;
```

	mtd_avg_dti numeric
1	13.6700

MoM Changes

```
SELECT
    EXTRACT(YEAR FROM issue_date) AS year,
    EXTRACT(MONTH FROM issue_date) AS month,
    ROUND(AVG(dti)::numeric, 4) * 100 AS avg_dti,
    LAG(ROUND(AVG(dti)::numeric, 4) * 100) OVER (ORDER BY EXTRACT(YEAR FROM issue_date), EXTRACT(MONTH FROM issue_date)) AS previous_month_dti,
    ROUND(AVG(dti)::numeric, 4) * 100 - LAG(ROUND(AVG(dti)::numeric, 4) * 100) OVER (ORDER BY EXTRACT(YEAR FROM issue_date), EXTRACT(MONTH FROM issue_date)) AS MoM_change
FROM loans
GROUP BY year, month
ORDER BY year, month;
```

	year numeric	month numeric	avg_dti numeric	previous_month_dti numeric	mom_change numeric
1	2021	1	12.9400	[null]	[null]
2	2021	2	13.4100	12.9400	0.4700
3	2021	3	13.2200	13.4100	-0.1900
4	2021	4	13.2200	13.2200	0.0000
5	2021	5	13.3300	13.2200	0.1100
6	2021	6	13.2400	13.3300	-0.0900
7	2021	7	13.2900	13.2400	0.0500
8	2021	8	13.3500	13.2900	0.0600
9	2021	9	13.3000	13.3500	-0.0500
10	2021	10	13.4100	13.3000	0.1100
11	2021	11	13.3000	13.4100	-0.1100
12	2021	12	13.6700	13.3000	0.3700

Good Loans

1. Good Loan Application Percentage

```
SELECT
    (COUNT(CASE WHEN loan_status = 'Fully Paid'
            OR loan_status = 'Current'
            THEN id END)) * 100 /
    COUNT(id) AS good_loan_perc
FROM loans;
```

	good_loan_perc bigint
1	86

2. Good Loan Applications

```
SELECT
    COUNT(id)
FROM loans
WHERE loan_status = 'Fully Paid' OR loan_status = 'Current';
```

	count bigint
1	33243

3. Good Loan Funded Amount

```
SELECT
    SUM(loan_amount) AS good_loan_funded_amount
FROM loans
WHERE loan_status = 'Fully Paid' OR loan_status = 'Current';
```

	good_loan_funded_amount double precision
1	370224850

4. Good Loan Total Received Amount

```
SELECT
    SUM(total_payment) AS good_loan_received_amount
FROM loans
WHERE loan_status = 'Fully Paid' OR loan_status = 'Current';
```

	good_loan_received_amount double precision
1	435786170

The received amount is more than the funded (or, invested) amount in good loans. This means the bank is making profit from good loans.

Bad Loans

1. Good Loan Application Percentage

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

	good_loan_perc bigint
1	13

2. Bad Loan Applications

```
SELECT
    COUNT(id)
FROM loans
WHERE loan_status = 'Charged Off';
```

	count bigint
1	5333

3. Bad Loan Funded Amount

```
SELECT
    SUM(loan_amount) AS bad_loan_funded_amount
FROM loans
WHERE loan_status = 'Charged Off';
```

	bad_loan_funded_amount double precision
1	65532225

4. Bad Loan Total Received Amount

```
SELECT
    SUM(total_payment) AS bad_loan_received_amount
FROM loans
WHERE loan_status = 'Charged Off';
```

	bad_loan_funded_amount double precision
1	65532225

The banks are losing money as the customers are not paying back their loans/installments. We suggest further investigations into the customers, such as credit reports and scores.

Loan Status

```
SELECT
    loan_status,
    COUNT(id) as total_loan_applications,
    SUM(total_payment) as total_amt_received,
    SUM(loan_amount) AS total_funded_amt,
    AVG(int_rate)*100 AS avg_int_rate,
    AVG(dti)*100 AS DTI
FROM loans
GROUP BY loan_status;
```

	loan_status character varying (50)	total_loan_applications bigint	total_amt_received double precision	total_funded_amt double precision	avg_int_rate double precision	dti double precision
1	Current	1098	24199914	18866500	15.099326047358838	14.724344262295071
2	Fully Paid	32145	411586256	351358350	11.641070773059624	13.16735075439415
3	Charged Off	5333	37284763	65532225	13.878574910931937	14.004732795799802

```
SELECT
    loan_status,
    SUM(total_payment) as MTD_total_amt_received,
    SUM(loan_amount) AS MTD_total_funded_amt
FROM loans
WHERE EXTRACT(MONTH FROM issue_date) = 12
GROUP BY loan_status;
```

	loan_status character varying (50)	mtd_total_amt_received double precision	mtd_total_funded_amt double precision
1	Charged Off	5324211	8732775
2	Current	4934318	3946625
3	Fully Paid	47815851	41302025

Monthly Trends by Issue Date

To identify seasonality and long-term trends in lending activities.

```
SELECT
    EXTRACT(MONTH FROM issue_date),
    to_char(issue_date, 'Month') AS month,
    COUNT(id) AS total_loan_applications,
    SUM(loan_amount) AS total_funded_amt,
    SUM(total_payment) AS total_received_amt
FROM loans
GROUP BY EXTRACT(MONTH FROM issue_date), month
ORDER BY EXTRACT(MONTH FROM issue_date), month;
```

	extract numeric	month text	total_loan_applications bigint	total_funded_amt double precision	total_received_amt double precision
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

Regional Analysis by State

To identify regions with significant lending activity and assess regional disparitites.

```
SELECT
    address_state AS state,
    COUNT(id) AS total_loan_applications,
    SUM(loan_amount) AS total_funded_amt,
    SUM(total_payment) AS total_received_amt
FROM loans
GROUP BY address_state
ORDER BY SUM(loan_amount) DESC;
```

	state character varying (50)	total_loan_applications bigint	total_funded_amt double precision	total_received_amt double precision
1	CA	6894	78484125	83901234
2	NY	3701	42077050	46108181
3	TX	2664	31236650	34392715
4	FL	2773	30046125	31601905
5	NJ	1822	21657475	23425159
6	IL	1486	17124225	18875941
7	VA	1375	15982650	17711443
8	PA	1482	15826525	17462908
9	GA	1355	15480325	16728040
10	MA	1310	15051000	16676279
11	OH	1188	12991375	14330148

Loan Term Analysis

Allow client to understand the distribution of loans across various term lengths.

```
SELECT
    term,
    COUNT(id) AS total_loan_applications,
    SUM(loan_amount) AS total_funded_amt,
    SUM(total_payment) AS total_received_amt
FROM loans
GROUP BY term
ORDER BY term;
```

	term text	total_loan_applications bigint	total_funded_amt double precision	total_received_amt double precision
1	36 months	28237	273041225	294709458
2	60 months	10339	162715850	178361475

Employee Length Analysis

How lending metrics are distributed among borrowers with different employment lengths, helping us to assess the impact of employment history on loan applications.

```
SELECT
    emp_length,
    COUNT(id) AS total_loan_applications,
    SUM(loan_amount) AS total_funded_amt,
    SUM(total_payment) AS total_received_amt
FROM loans
GROUP BY emp_length
ORDER BY total_loan_applications;
```

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

Loan Purpose Breakdown

A visual breakdown will be provided in Tableau on the loan metrics based on stated purposes of loans, aiding in the understanding of the primary reasons borrowers seeking financing.

```
SELECT
    purpose,
    COUNT(id) AS total_loan_applications,
    SUM(loan_amount) AS total_funded_amt,
    SUM(total_payment) AS total_received_amt
FROM loans
GROUP BY purpose
ORDER BY total_loan_applications;
```

	purpose character varying (100)	total_loan_applications bigint	total_funded_amt double precision	total_received_amt double precision
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 Analysis

For a hierarchical view of how home ownership impacts loan applications and disbursements.

```
SELECT
    home_ownership,
    COUNT(id) AS total_loan_applications,
    SUM(loan_amount) AS total_funded_amt,
    SUM(total_payment) AS total_received_amt
FROM loans
GROUP BY home_ownership
ORDER BY total_loan_applications DESC;
```

	home_ownership character varying (50)	total_loan_applications bigint	total_funded_amt double precision	total_received_amt double precision
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