

BANK LOAN ANALYSIS USING POSTGRESQL

Problem Statement:



In the financial sector, managing loan portfolios effectively is crucial for minimizing risk and maximizing profitability. Lenders often struggle with identifying patterns that differentiate good loans from potential defaulters, leading to significant financial losses. A comprehensive analysis of loan data can provide actionable insights, enabling businesses to better assess risk, improve decision-making, and optimize their lending strategies.













Project Objectives:

This project aims to analyze a bank's loan data using PostgreSQL to provide key performance indicators (KPIs) and other vital metrics that distinguish between good loans and defaulters.

Data Import and Setup:

- 1st step is to create a database or use an existing database
 - Syntax to create a new database:
CREATE DATABASE your_database_name;
 - 2nd step is to create a table and import data into it
 - 3rd Create appropriate tables and schemas to organize the data efficiently.
 - 4th Data Analysis
-

 postgres/postgres@PostgreSQL 15 ▼ 

  ▼  ▼  ▼ No limit ▼    ▼   ▼   

Query Query History

```
1  ▼ create table financial_loan
2  (
3  id numeric,
4  address_state varchar(4),
5  application_type varchar(20),
6  emp_length varchar(20),
7  emp_title varchar(25),
8  grade char,
9  home_ownership varchar(15),
10 issue_date date,
11 last_credit_pull_date date,
12 last_payment_date date,
13 loan_status varchar(20),
14 next_payment_date date,
15 member_id numeric,
16 purpose varchar(15),
17 sub_grade varchar(5),
18 term varchar(25),
19 verification_status varchar(20),
20 annual_income numeric,
21 dti int,
22 installment int,
23 int_rate int,
24 loan_amount numeric,
25 total_acc numeric,
26 total pavment numeric):
```

```

SET datestyle = 'DMY';

ALTER TABLE financial_loan
ALTER COLUMN installment TYPE NUMERIC,
ALTER COLUMN int_rate TYPE NUMERIC;

ALTER TABLE financial_loan
ALTER COLUMN purpose Type varchar(40),
ALTER COLUMN emp_title Type varchar(100);

COPY financial_loan
FROM 'D:/Study/Notes/Projects & Data/Finance Domain bank loan pr
DELIMITER ','
CSV HEADER;




```

```

1 ✓ SELECT * FROM public.financial_loan
2
3 ALTER TABLE financial_loan
4 ALTER COLUMN dti TYPE float;
5
6 ✓ ALTER TABLE financial_loan
7 ALTER COLUMN installment TYPE float;
8
9 ✓ ALTER TABLE financial_loan
10 ALTER COLUMN int_rate TYPE float;
11
12

```

Calculate KPI's

QUERY	Results				
Total Loan Applications SELECT COUNT(id) AS Total_Applications FROM financial_loan	<table> <tr> <td></td><td>total_application </td></tr> <tr> <td>1</td><td>38576</td></tr> </table>		total_application 	1	38576
	total_application 				
1	38576				

Month-To-Date Total Loan Applications

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

mtd bigint	🔒
4314	

Previous MTD Total Loan Applications

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

pmtd bigint	🔒
4035	

MoM Growth Calculation of Loan Applications

```
WITH CurrentMonth AS (
  SELECT COUNT(id) AS current_month_count
  FROM financial_loan
  WHERE EXTRACT(MONTH FROM issue_date) = 12
  AND EXTRACT(YEAR FROM issue_date) = 2021
),
PreviousMonth AS (
  SELECT COUNT(id) AS previous_month_count
  FROM financial_loan
  WHERE EXTRACT(MONTH FROM issue_date) = 11
  AND EXTRACT(YEAR FROM issue_date) = 2021
)
SELECT
  CurrentMonth.current_month_count,
  PreviousMonth.previous_month_count,
  CASE
    WHEN PreviousMonth.previous_month_count = 0 THEN NULL
    ELSE ((CurrentMonth.current_month_count -
  PreviousMonth.previous_month_count) * 100.0 /
  PreviousMonth.previous_month_count)
  END AS mom_growth_percentage
FROM
  CurrentMonth,
  PreviousMonth;
```

current_month_count bigint	🔒
4314	

previous_month_count bigint	🔒
4035	

mom_growth_percentage numeric	🔒
6.9144981412639405	

Total Funded Amount

```
SELECT sum(loan_amount) AS Total_Funded_Amount FROM
financial_loan
```

total_funded_amount numeric	🔒
435757075	

MTD Total Funded Amount

```
SELECT sum(loan_amount) AS MTD_Total_Funded_Amount FROM
financial_loan
WHERE EXTRACT(MONTH FROM issue_date) = 12
AND EXTRACT(YEAR FROM issue_date) = 2021;
```

mtd_total_funded_amount numeric	🔒
53981425	

Total Amount Received SELECT sum(total_payment) AS Total_Amount_Received FROM financial_loan	<div>total_amount_received</div> <div>numeric</div> <div>473070933</div>
Receivable Amount SELECT (SELECT SUM(total_payment) AS Total_Amount_Received FROM financial_loan WHERE EXTRACT(MONTH FROM issue_date) = 12 AND EXTRACT(YEAR FROM issue_date) = 2021) - (SELECT SUM(loan_amount) AS MTD_Total_Funded_Amount FROM financial_loan WHERE EXTRACT(MONTH FROM issue_date) = 12 AND EXTRACT(YEAR FROM issue_date) = 2021) AS Remaning_amount;	<div>remaning_amount</div> <div>numeric</div> <div>4092955</div>
Average Interest Rate SELECT avg(int_rate) * 100 AS Avg_Interest_Rate FROM financial_loan	<div>avg_interest_rate</div> <div>double precision</div> <div>12.048831397760178</div>
Debt to Income SELECT AVG(dti) * 100 AS Avg_Dti FROM financial_loan;	<div>avg_dti</div> <div>double precision</div> <div>13.32743311903776</div>

Good Loan Info

SELECT (COUNT (CASE WHEN loan_status = 'Fully Paid' OR loan_status = 'Current' THEN id END) * 100) / COUNT(id) AS Good_Loan_Percentage FROM financial_loan;	<div>good_loan_percentage</div> <div>bigint</div> <div>86</div>
SELECT COUNT(id) AS Good_Loan_Applicants FROM financial_loan	

where loan_status = 'Fully Paid' OR loan_status = 'Current'

good_loan_applicants bigint
33243

SELECT SUM(loan_amount) AS Good_Loan_Amount
FROM financial_loan
where loan_status = 'Fully Paid' OR loan_status = 'Current'

good_loan_amount numeric
370224850

SELECT SUM(total_payment) AS Good_Loan_amount_received
FROM financial_loan
where loan_status = 'Fully Paid' OR loan_status = 'Current'

good_loan_amount_received numeric
435786170

Bad Loan Info

SELECT
 (COUNT
 (CASE WHEN loan_status = 'Charged Off' THEN id END) * 100.0)
 / COUNT(id) AS bad_Loan_Percentage
FROM financial_loan;

bad_loan_percentage numeric
13.8246578183326421

SELECT COUNT(id) AS bad_Loan_Applicants
FROM financial_loan
where loan_status = 'Charged Off'

bad_loan_applicants bigint
5333

```
SELECT sum(loan_amount) AS bad_Loan_Amount
FROM financial_loan
where loan_status = 'Charged Off'
```

bad_loan_amount	numeric
65532225	

```
SELECT sum(total_payment) AS bad_Loan_amount_received
FROM financial_loan
where loan_status = 'Charged Off'
```

bad_loan_amount_received	numeric
37284763	

Other Metrics

```
SELECT loan_status,
count(id) as LoanCount,
sum(loan_amount) as Total_Funded_Amount,
sum(total_payment) as Total_Amount_Received,
AVG(int_rate * 100) AS Interest_Rate,
AVG(dti * 100) AS DTI
FROM financial_loan
group by loan_status
```

	loan_status character varying (20)	loancount bigint	total_funded_amount numeric	total_amount_received numeric	interest_rate double precision	dti double precision
1	Current	1098	18866500	24199914	15.0993260473588	14.724344262295068
2	Fully Paid	32145	351358350	411586256	11.641070773058658	13.167350754394162
3	Charged Off	5333	65532225	37284763	13.878574910931917	14.004732795799695

```
SELECT loan_status,
count(id) as LoanCount,
sum(loan_amount) as MTD_Funded_Amount,
sum(total_payment) as MTD_Amount_Received
FROM financial_loan
where extract (month from issue_date) = 12
group by loan_status
```

	loan_status character varying (20) 🔒	loancount bigint 🔒	mtd_funded_amount numeric 🔒	mtd_amount_received numeric 🔒
1	Charged Off	649	8732775	5324211
2	Current	213	3946625	4934318
3	Fully Paid	3452	41302025	47815851

```
SELECT address_state,
count(id) as LoanCount,
sum(loan_amount) as MTD_Funded_Amount,
sum(total_payment) as MTD_Amount_Received
FROM financial_loan
group by address_state
order by address_state
```

	address_state character varying (4) 🔒	loancount bigint 🔒	mtd_funded_amount numeric 🔒	mtd_amount_received numeric 🔒
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

Total rows: 50 of 50 Query complete 00:00:00.310 Ln 4, Col 1


```

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_Amount_Received
FROM financial_loan
GROUP BY Month_Number, Month_Name
ORDER BY Month_Number;

```

	month_number numeric	month_name text	total_loan_applications bigint	total_funded_amount numeric	total_amount_received numeric
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

```

SELECT
    term,
    COUNT(id) AS Total_Loan_Applications,
    SUM(loan_amount) AS Total_Funded_Amount,
    SUM(total_payment) AS Total_Amount_Received
FROM financial_loan
GROUP BY term

```

	term character varying (25)	total_loan_applications bigint	total_funded_amount numeric	total_amount_received numeric
1	36 months	28237	273041225	294709458
2	60 months	10339	162715850	178361475

```

SELECT
    emp_length,
    COUNT(id) AS Total_Loan_Applications,
    SUM(loan_amount) AS Total_Funded_Amount,
    SUM(total_payment) AS Total_Amount_Received
FROM financial_loan
GROUP BY emp_length
ORDER BY emp_length

```

	emp_length character varying (20) 🔒	total_loan_applications bigint 🔒	total_funded_amount numeric 🔒	total_amount_received numeric 🔒
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

```

SELECT
    purpose,
    COUNT(id) AS Total_Loan_Applications,
    SUM(loan_amount) AS Total_Funded_Amount,
    SUM(total_payment) AS Total_Amount_Received
FROM financial_loan
GROUP BY purpose
ORDER BY Total_Funded_Amount desc

```

	purpose character varying (40) 🔒	total_loan_applications bigint 🔒	total_funded_amount numeric 🔒	total_amount_received numeric 🔒
1	Debt consolidation	18214	232459675	253801871
2	credit card	4998	58885175	65214084
3	home improvement	2876	33350775	36380930
4	other	3824	31155750	33289676
5	small business	1776	24123100	23814817
6	major purchase	2110	17251600	18676927
7	car	1497	10223575	11324914
8	wedding	928	9225800	10266856
9	medical	667	5533225	5851372
10	house	366	4824925	5185538
11	moving	559	3748125	3999899
12	educational	315	2161650	2248380
13	vacation	352	1967950	2116738
14	renewable_energy	94	845750	898931

```

SELECT
  home_ownership,
  COUNT(id) AS Total_Loan_Applications,
  SUM(loan_amount) AS Total_Funded_Amount,
  SUM(total_payment) AS Total_Amount_Received
FROM financial_loan
GROUP BY home_ownership;

```

	home_ownership character varying (15) 🔒	total_loan_applications bigint 🔒	total_funded_amount numeric 🔒	total_amount_received numeric 🔒
1	MORTGAGE	17198	219329150	238474438
2	RENT	18439	185768475	201823056
3	NONE	3	16800	19053
4	OTHER	98	1044975	1025257
5	OWN	2838	29597675	31729129