Sharodindu Biswas

Bank loan report

Problem statement:

This report aims to provide insights into key loan-related metrics and their changes over time. The report will help us make data-driven decisions, track our loan portfolio's health, and identify trends that can inform our lending strategies.

In order to gain a comprehensive overview of our lending operations and monitor the performance of loans, we aim to create a grid view report categorized by 'Loan Status.' This report will serve as a valuable tool for analysing and understanding the key indicators associated with different loan statuses. By providing insights into metrics such as 'Total Loan Applications,' 'Total Funded Amount,' 'Total Amount Received,' 'Month-to-Date (MTD) Funded Amount,' 'MTD Amount Received,' 'Average Interest Rate,' and 'Average Debt-to-Income Ratio (DTI),' this grid view will empower us to make data-driven decisions and assess the health of our loan portfolio.

This diverse analysis will enhance our ability to visualize and communicate loan-related insights effectively, supporting data-driven decisions and strategic planning within our lending operations.

In our Bank Loan Report project, we recognize the need for a comprehensive 'Details Dashboard' that provides a consolidated view of all the essential information within our loan data. This Details Dashboard aims to offer a holistic snapshot of key loan-related metrics and data points, enabling users to access critical information efficiently.

Technologies used in performing the analysis:

1. Microsoft Excel
2. Microsoft word
3. SQL server management studio (SSMS)
4. Kaggle

Sharodindu Biswas

Introduction to Data

1. **Loan ID**: Loan ID is a unique identifier assigned to each loan application.
2. **Address State**: Address State indicates the borrower's location.
3. **Employee Length**: Employee Length provides insights into the borrower's employment stability.
4. **Employee Title**: Employee Title specifies the borrower's occupation or job title.
5. **Grade:** Grade represents a risk classification assigned to the loan based on creditworthiness.
6. **Loan Status**: Loan Status indicates the current state of the loan (e.g., fully paid, current, default). It tracks loan performance.
7. **Next Payment Date:** Next Payment Date estimates the date of the next loan payment.
8. **Purpose:** Purpose specifies the reason for the loan.
9. **Term:** Term defines the duration of the loan in months.
10. **Verification Status:** Verification Status indicates whether the borrower's financial information has been verified.
11. **Annual Income:** Annual Income reflects the borrower's total yearly earnings.
12. **Sub Grade:** Sub Grade refines the risk assessment within a grade, providing additional risk differentiation.
13. **Home Ownership:** Home Ownership indicates the borrower's housing status.
14. **Issue Date:** Issue Date marks the loan's origination date.
15. **Last Credit Pull Date:** Last Credit Pull Date records when the borrower's credit report was last accessed. It helps monitor creditworthiness.
16. **Last Payment Date:** Last Payment Date marks the most recent loan payment received. It tracks the borrower's payment history.
17. **DTI (Debt-to-Income Ratio):** DTI measures the borrower's debt burden relative to income.
18. **Instalment:** Instalment is the fixed monthly payment amount for loan repayment, including principal and interest.
19. **Interest Rate:** Interest Rate represents the annual cost of borrowing expressed as a percentage.
20. **Loan Amount:** Loan Amount is the total borrowed sum.

Key Performance indicators (KPI)

- A performance indicator or key performance indicator (KPI) is a type of performance measurement. KPI's evaluate the success of an organization or of a particular activity (such as projects, programs, products and other initiatives) in which it engages. KPIs provide a focus for strategic and operational improvement, create an analytical basis for decision making and help focus attention on what matters most.
- The key performance indicators play an important part in the analysis these are the information retrieved after exploratory analysis of the loan data of the bank.

There are two categories of indicators:

1. **Qualitative**: represents non-numeric conformance to a standard, or interpretation of personal feelings, tastes, opinions or experiences.
2. **Quantitative**: facts presented with a specific objective numeric value measured against a standard.

Objective:

- The primary objective of the exploratory financial data analysis is to provide a comprehensive and user-friendly interface for accessing vital loan data.
- It will serve as a one-stop solution for users seeking detailed insights into our loan portfolio, borrower profiles, and loan performance.

Sharodindu Biswas

Total loan application:

```
/* Total application received and month over month growth of loan application FY-2021 */

SELECT COUNT(id) as Total_application
FROM financial_loan;
```

Result: The above query returns all the applications received by the bank.

Month-to-Date (MTD):

```
/*Month to date loan application or most recent loan application*/

SELECT COUNT(id) as Total_application_dec
FROM financial_loan
WHERE month(issue_date)=12 and year(issue_date)=2021;
```

Result: The above query returns all the applications received by the bank in the month of Dec.

Month-over-month (MoM):

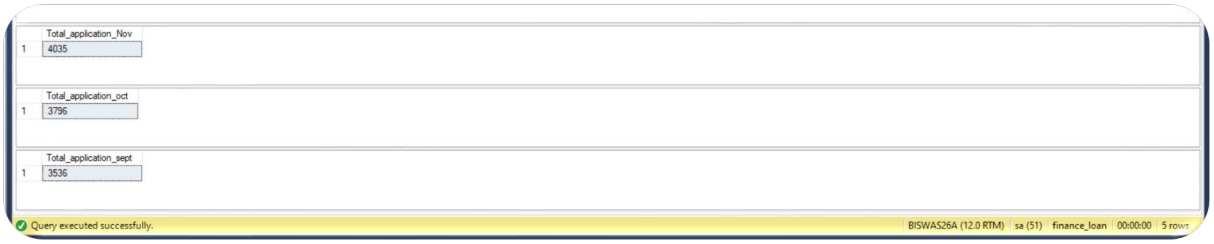
```
/*Month over month growth*/

SELECT COUNT(id) as Total_application_Nov
FROM financial_loan
WHERE month(issue_date) = 11 and year(issue_date) =2021;

 SELECT COUNT(id) as Total_application_oct
FROM financial_loan
WHERE month(issue_date) = 10 and year(issue_date) =2021;

SELECT COUNT(id) as Total_application_sept
FROM financial_loan
WHERE month(issue_date) =9 and year(issue_date)=2021;
```

Result: The above query returns the application counts growth month over month.



Sharodindu Biswas

Total Funded Amount:

```sql
/* Total funded amount or total amount of funds disbursed and the month to month flow of funds FY-2021 */

select sum(loan_amount) as total_amount_funded
from financial_loan;
```

Result: The above query calculates the total amount funded by the bank to its customer.

	total_amount_funded
1	435757075

Month-to-date (MTD):

```sql
/* Total funded amount recent months, total amount of funds disbursed */

select sum(loan_amount) as disbursed_on_dec
from financial_loan
where month(issue_date)=12 and year(issue_date) = 2021;
```

Result: The above query calculates the total amount funded by the bank in the month of Dec.

	disbursed_on_dec
1	53981425

Month-over-month (MoM):

```sql
/* Total funded amount recent months, total amount of funds disbursed */

select sum(loan_amount) as disbursed_on_dec
from financial_loan
where month(issue_date)=12 and year(issue_date) = 2021;

/* Total disbursed amount on month of November */

select sum(loan_amount) as disbursed_on_nov
from financial_loan
where month(issue_date)=11 and year(issue_date) = 2021;

/* Total disbursed amount on month of oct */

select sum(loan_amount) as disbursed_on_oct
from financial_loan
where month(issue_date)=10 and year(issue_date) = 2021;
```

Result: The above query calculates the month over month amount funded by the bank.

	disbursed_on_dec
1	53981425

	disbursed_on_nov
1	47754825

	disbursed_on_oct
1	44893800

Sharodindu Biswas

Total Amount Received:

```
/* Total amount received and the month over month(MoM) inflow of funds disbursed */

select sum(total_payment) as total_amount_received
from financial_loan;
```

Result: The above query returns the total of all the amounts disbursed from the bank.

	total_amount_received
1	473070933

Month-over-month inflow of disbursed funds:

```
/* The month over month inflow of funds disbursed FY-2021 */

select sum(total_payment) as amount_rececived_dec
from financial_loan
where month(issue_date) = 12 and year(issue_date) = 2021;

select sum(total_payment) as amount_rececived_nov
from financial_loan
where month(issue_date) = 11 and year(issue_date) = 2021;

select sum(total_payment) as amount_rececived_Oct
from financial_loan
where month(issue_date) = 10 and year(issue_date) = 2021;
```

Result: The above query analyses and returns the difference between the inflow of funds.

	amount_rececived_dec
1	58074380

	amount_rececived_nov
1	50132030

	amount_rececived_Oct
1	49399567

Query executed successfully. BISWAS26A (12.0 RTM) | sa (54) | finance_loan | 00:00:00 | 4 rows

Average Interest Rate and the month-over-month change in the average interest rate:

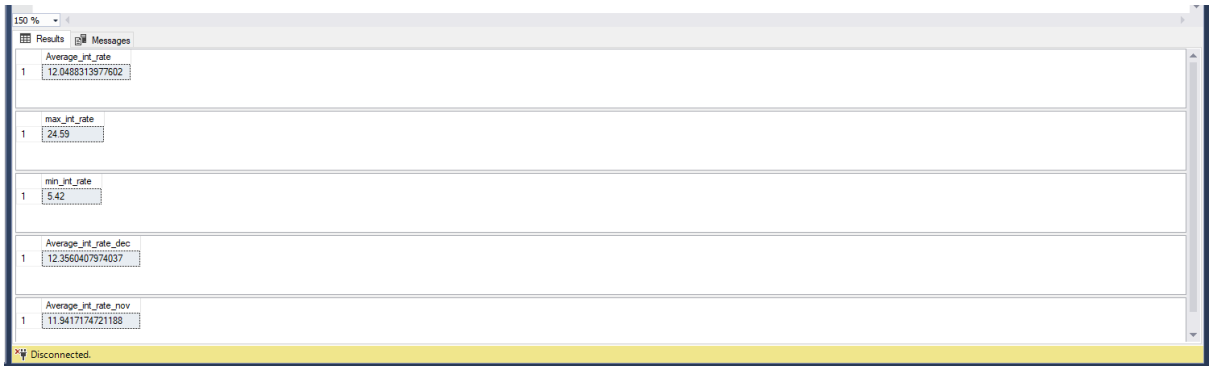
```
/* Interest rate and the month over month change in the interest rate FY-2021 */

select avg(int_rate)*100 as Average_int_rate
from financial_loan;
select max(int_rate)*100 as max_int_rate
from financial_loan;
select min(int_rate)*100 as min_int_rate
from financial_loan;

/* Month to date interest rate followed by the previous month average interest rate */

select avg(int_rate)*100 as Average_int_rate_dec
from financial_loan
where month(issue_date) = 12 and year(issue_date) =2021;
select avg(int_rate)*100 as Average_int_rate_nov
from financial_loan
where month(issue_date) = 11 and year(issue_date) =2021;
```

Result: By executing the above query, we can retrieve the average interest rate and the month over month change in the average interest rate.



Average Debt-to-Income Ratio (DTI):

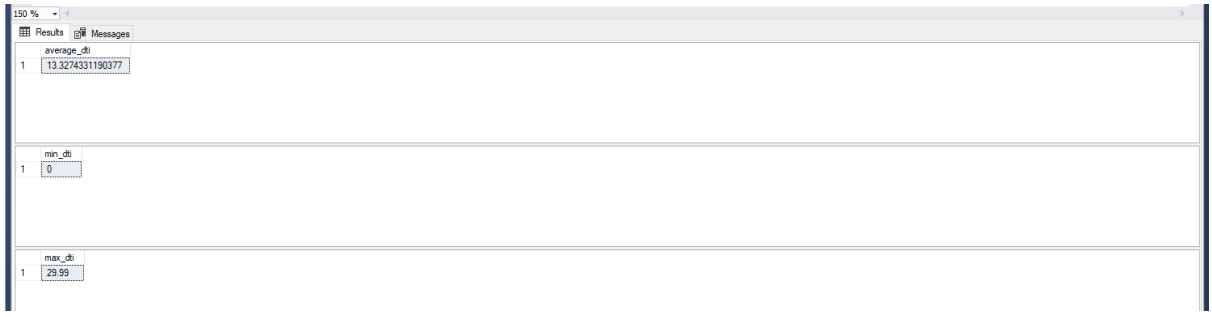
```
/* Debt to income ratio (DTI) and the month over month change in the DTI (Avg, Max, Min) */

select avg(dti)*100 as average_dti
from financial_loan;

select min(dti)*100 as min_dti
from financial_loan;

select max(dti)*100 as max_dti
from financial_loan;
```

Result: The above query returns the average, minimum and the maximum debt-to-income ratio.



Sharodindu Biswas

Good Loan Application Percentage:

```sql
/* Good loan percentage:By this query we can retrive the percentage of good loans disbursed from the bank FY-2021 */

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;
```

Result: The above query returns the percentage of good loan application disbursed from the bank.

	Good_loan_percentage
1	86

Good Loan Funded Amount & Good Loan Total Received Amount:

```sql
/* Good loan application count */

select count(id) as good_loan_application
from financial_loan
where loan_status = 'Fully Paid' or loan_status = 'Current';

/* Total good loan amount funded */

select sum(loan_amount) as good_loan_amount
from financial_loan
where loan_status = 'Fully Paid' or loan_status = 'Current';

/* Total good loan amount received */

select sum(total_payment) as Good_loan_amount_rec
 from financial_loan
 where loan_status = 'Fully Paid' or loan_status = 'Current';
```

Result: The above query returns the total good loan application count followed by amount funded and the funds received from the disbursement.

	good_loan_application
1	33243

	good_loan_amount
1	370224850

	Good_loan_amount_rec
1	435786170

	total_profit
1	65561320

Query executed successfully. BISWAS26A (12.0 RTM) | sa (53) | finance_loan | 00:00:00 | 5 rows

Total Profits: By running this query, we can retrieve the profit earned by the bank.

```sql
/* Profit gained from the funded amount */
select
435786170 - 370224850 /* good loan amount - good loan amount rec */
as total_profit;
```

	total_profit
1	65561320

Sharodindu Biswas

Bad Loan Application Percentage:

```sql
/* Bad loan percentage: By executing this query we can retrive the percentage of bad loans */
SELECT
    (COUNT(CASE WHEN loan_status = 'Charged Off' THEN id END) * 100.0) /
    COUNT(id) AS Bad_Loan_Percentage
FROM financial_loan;
```

Result: The above query will return the percentage of bad loan application received.

	Bad_Loan_Percentage
1	13.824657818332

Bad loan application count:

```sql
/* Count of bad loan application */

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

Result: The query will return the number of bad loan application charged off.

	Bad_Loan_Applications
1	5333

Bad loan amount funded & Received:

```sql
/* Total bad loan amount funded */

select sum(loan_amount) as bad_loan_funded
from financial_loan
where loan_status = 'Charged Off';

/* Total bad loan amount received */

select sum(total_payment) as bad_loan_amnt_rec
from financial_loan
where loan_status = 'Charged Off';
```

Result: The query will return the total amount of funds disbursed against bad loan application.

	bad_loan_funded
1	65532225

	bad_loan_amnt_rec
1	37284763

Loss Incurred by the bank:

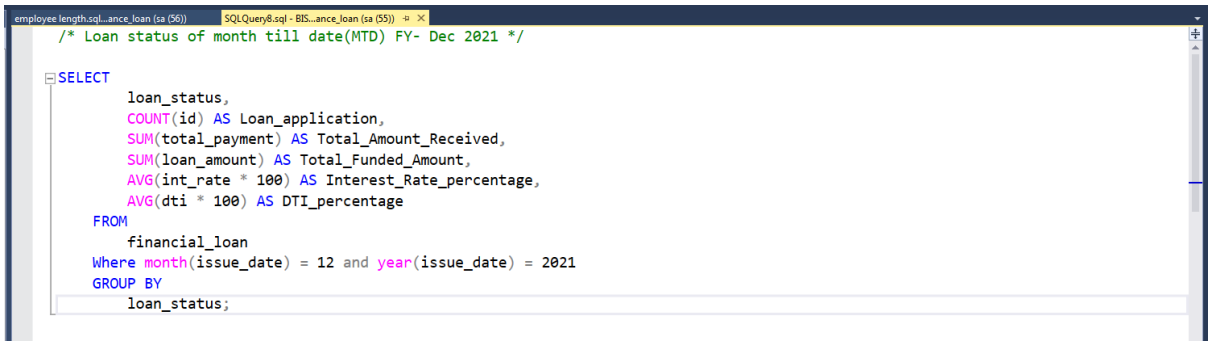
```sql
select (select sum(loan_amount) as bad_loan_funded
from financial_loan
where loan_status = 'Charged Off') - (select sum(total_payment) as bad_loan_amnt_rec
from financial_loan
where loan_status = 'Charged Off') as loss_incured;
```

Result: The above query, will return the loss incurred by the bank.

	loss_incured
1	28247462

Query executed successfully. BISWAS26A (12.0 RTM) | sa (54) | finance_loan | 00:00:00 | 5 rows

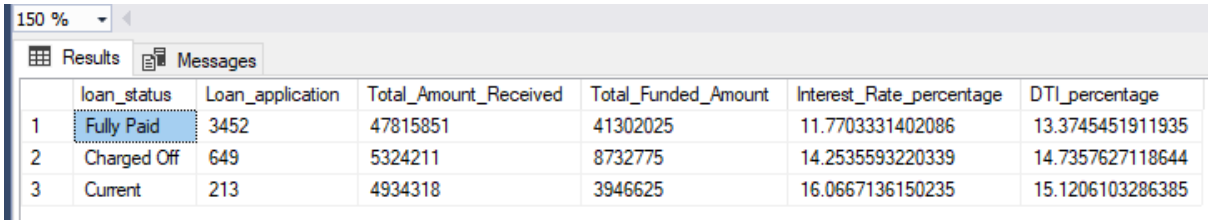
Loan status of Month-till-date (MTD) FY-2021:



```sql
/* Loan status of month till date(MTD) FY- Dec 2021 */

SELECT
        loan_status,
        COUNT(id) AS Loan_application,
        SUM(total_payment) AS Total_Amount_Received,
        SUM(loan_amount) AS Total_Funded_Amount,
        AVG(int_rate * 100) AS Interest_Rate_percentage,
        AVG(dti * 100) AS DTI_percentage
    FROM
        financial_loan
    Where month(issue_date) = 12 and year(issue_date) = 2021
    GROUP BY
        loan_status;
```

Result: The above query returns the count of loan application, total amount funded, total amount received, Interest rate and the DTI percentage according to the loan status.



150 %

Results | Messages

	loan_status	Loan_application	Total_Amount_Received	Total_Funded_Amount	Interest_Rate_percentage	DTI_percentage
1	Fully Paid	3452	47815851	41302025	11.7703331402086	13.3745451911935
2	Charged Off	649	5324211	8732775	14.2535593220339	14.7357627118644
3	Current	213	4934318	3946625	16.0667136150235	15.1206103286385

Employee Length Analysis:

```sql
/* 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 financial_loan
GROUP BY emp_length
ORDER BY emp_length;
```

Result: The query will help assess the impact of employment history on loan applications.

150 %

Results | Messages

	Employee_Length	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
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

Monthly trends:

```sql
/* Monthly trends by issue date */
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 financial_loan
GROUP BY MONTH(issue_date), DATENAME(MONTH, issue_date)
ORDER BY MONTH(issue_date);
```

Result: This query will showcase how 'Total Loan Applications,' 'Total Funded Amount,' and 'Total Amount Received' vary over time, allowing us to identify seasonality and long-term trends in lending activities

150 %

Results | Messages

	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

Loan term analysis:

```sql
/* 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 financial_loan
GROUP BY term
ORDER BY term;
```

Result: This query will allow us to understand the distribution of loans across various term lengths.

150 %

Results | Messages

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

Regional Analysis by State:

```sql
/* 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 financial_loan
GROUP BY address_state
ORDER BY address_state;
```

Result: This query will represent lending metrics categorized by state, enabling us to identify regions with significant lending activity and assess regional disparities.

Results | Messages

	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

Query executed successfully.

Loan purpose breakdown:

```sql
/* Loan purpose */

select
purpose,
COUNT(id) as total_loan_appliction,
SUM(loan_amount) as funded_amount,
SUM(total_payment) as received_amount
from financial_loan
group by purpose
order by purpose;
```

Result: The query will provide a visual breakdown of loan metrics based on the stated purposes of loans, aiding in the understanding of the primary reasons borrowers seek financing.

Results | Messages

	Home_Ownership	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
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

Home Ownership Analysis:

```sql
/* 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 financial_loan
GROUP BY home_ownership
ORDER BY home_ownership;
```

Result: The above query will display loan metrics categorized by different home ownership statuses, allowing for a hierarchical view of how home ownership impacts loan applications and disbursements.

Results | Messages

	Home_Ownership	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
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

Conclusion

- In the Bank Loan Report project, we recognize the need for a comprehensive detail that provides a consolidated view of all the essential information within our loan data.
- This Details analysis aims to offer a holistic snapshot of key loan-related metrics and data points, enabling users to access critical information efficiently.