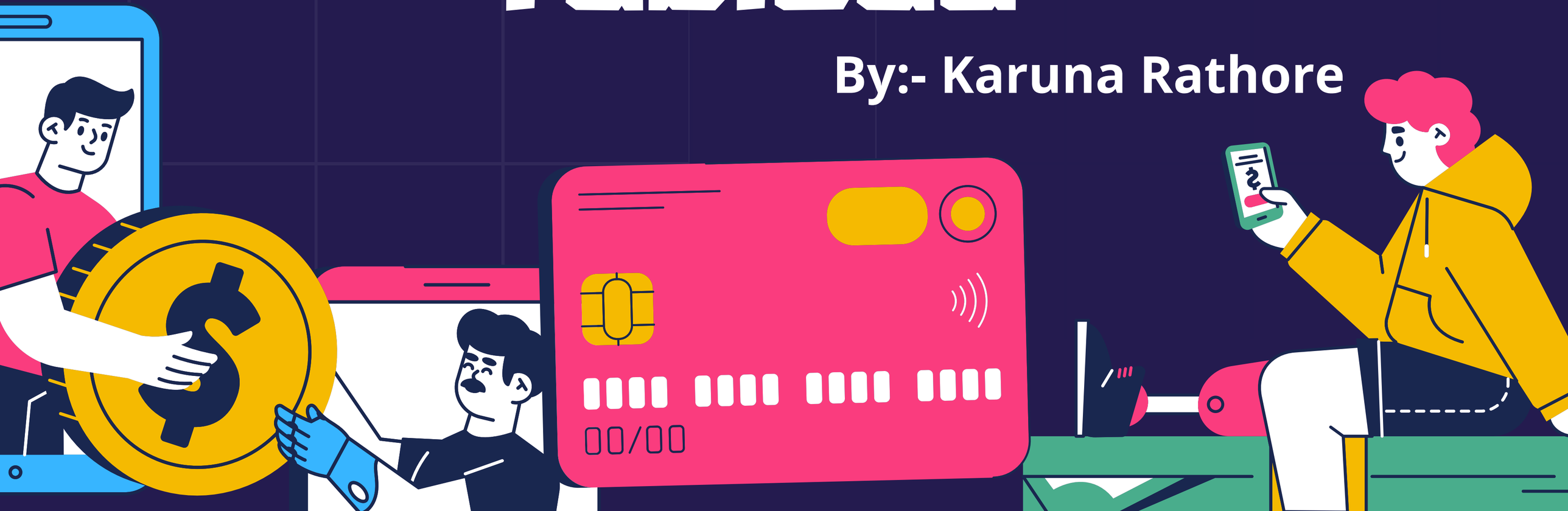
Analysing Bank Loan Data Using SQL & Tableau

By:- Karuna Rathore



Objective of Bank Loan Report

- In order to monitor and assess bank's lending activities and performance. I create a comprehensive Bank Loan Report. This report aims to provide insights into key loan-related metrics and their changes over time. The report will help Bank make data-driven decisions, track Bank loan portfolio's health, and identify trends that can inform lending strategies.

- Key Performance Indicators (KPIs) Includes:

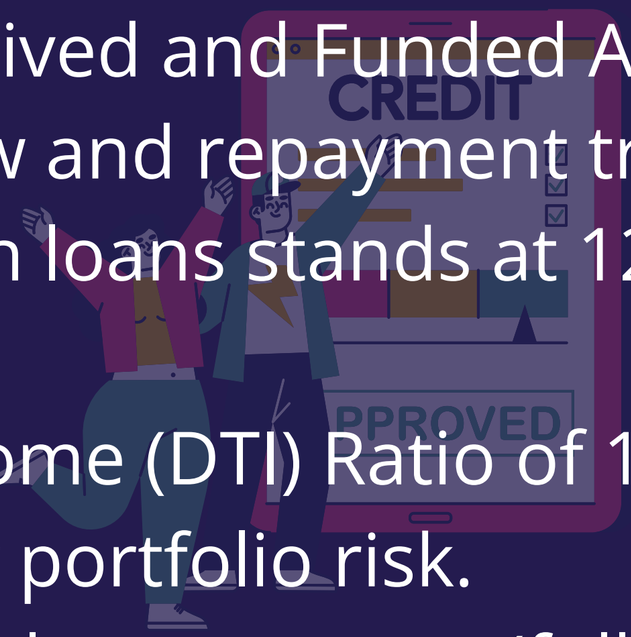
Total Loan Applications, Total Funded Amount, Total Amount Received, Average Interest Rate, Average Debt-to-Income Ratio (DTI), Good Loan and Bad Loans indicators, Month-to-Date (MTD) Funded Amount, and MTD Amount Received.

Reasons for Analyzing Bank Loan Data

- **Risk Assessment:** Evaluate creditworthiness, predict defaults, and set loan terms.
- **Portfolio Management:** Monitor and optimize loan portfolios.
- **Fraud Detection:** Detect unusual patterns and prevent fraud.
- **Regulatory Compliance:** Ensure adherence to regulations through data tracking and reporting.
- **Customer Insights:** Use data to tailor products and marketing strategies.
- **Profitability Analysis:** Assess loan portfolio profitability and costs.
- **Market Research:** Guide product development and competitive strategy.
- **Credit Risk Management:** Support risk strategies, loss provisioning, and stress testing.

Key Findings

- Customer engagement with bank from count of loans received stands at 38,576.
- Difference between Received and Funded Amount is $37,313,858. Highlighting financial flow and repayment trends.
- Average Interest Rates on loans stands at 12% Offers insights into loan pricing trends.
- The Average Debt-to-Income (DTI) Ratio of 13.27%. Indicating healthy financial profiles and low portfolio risk.
- 86.18% loans are in good loan category (fully paid or current) providing a measure of portfolio quality. It indicates strong repayment performance.
- More loan applications are for short-term loans than long-term.



- **Loan status wise Findings:-**

A) **Charged-off Loans:** Higher DTI (14.00%), indicating excessive leverage. Only 57% recovery, highlighting the need for better collection strategies.

B) **Fully Paid Loans:** Lowest DTI (13.17%) and strong revenue, with received amounts exceeding funded amounts due to high repayment rates.

C) **Current Loans:** Performing well, with received amounts ($24,199,914) already exceeding funded amounts ($18,866,500).

- Most loans are for debt consolidation, with a need to diversify into high-potential categories like Credit Card and Home Improvement.

Calculate the Total Number of Loan Applications Received

```
12      -- Retrieve Total Loan Applications --
13  •   SELECT
14        COUNT(ID) AS Total_Loan_Applications
15      FROM
16        LOAN_DATA;
17
```

Result Grid | Filter Rows:

	Total_Loan_Applications
▶	38576

Month-to-Date (MTD) Loan Applications

```sql
-- Retrieve Month-to-Date (MTD) Total Loan Applications --
SELECT
  COUNT(ID) AS MTD_Total_Loan_Applications
FROM
  LOAN_DATA
WHERE
  MONTH(ISSUE_DATE)= '12'
  AND YEAR(ISSUE_DATE)= 2021;
```

Result Grid | Filter Rows:

	MTD_Total_Loan_Applications
▶	4314

Month-over-Month (MoM) Changes In Loan Applications

```sql
-- Track Changes Month-over-Month (MoM) In Loan Applications --
select month(issue_date) Month,
count(id) as Monthwise_Total_Loan_Applications,
LAG(COUNT(ID)) OVER (ORDER BY MONTH(ISSUE_DATE)) AS Previous_Month_Total_Loan_Applications,
COUNT(ID) - LAG(COUNT(ID)) OVER (ORDER BY MONTH(ISSUE_DATE)) AS MoM_Change
from loan_data
group by month(issue_date)
order by 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Month	Monthwise_Total_Loan_Applications	Previous_Month_Total_Loan_Applications	MoM_Change
1	2332	NULL	NULL
2	2279	2332	-53
3	2627	2279	348
4	2755	2627	128
5	2911	2755	156
6	3184	2911	273
7	3366	3184	182
8	3441	3366	75
9	3536	3441	95
10	3796	3536	260
11	4035	3796	239
12	4314	4035	279

Result 10

Total Amount of Funds Disbursed as Loans

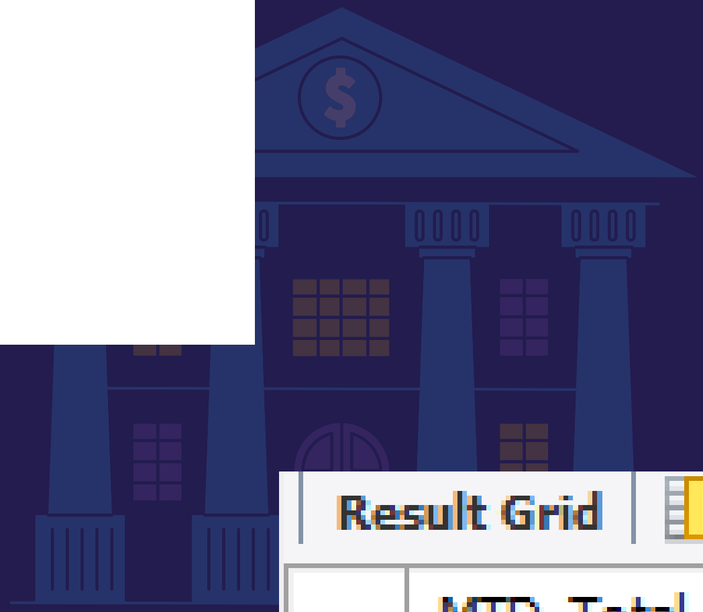
```sql
39      -- Total Amount of Funds Disbursed as Loans --
40  •   SELECT
41          SUM(LOAN_AMOUNT) AS Total_Funded_Amount
42      FROM
43          LOAN_DATA;
```

Result Grid | Filter Rc

	Total_Funded_Amount
▶	435757075

Month-to-Date (MTD) Total Funded Amount

```sql
-- Retrieve Month-to-Date (MTD) Total Funded Amount --
SELECT
  sum(loan_amount) AS MTD_Total_Funded_Amount
FROM
  LOAN_DATA
WHERE
  MONTH(ISSUE_DATE)= '12'
  AND YEAR(ISSUE_DATE)= 2021;
```



Result Grid | Filter Rows:

	MTD_Total_Funded_Amount
▶	53981425

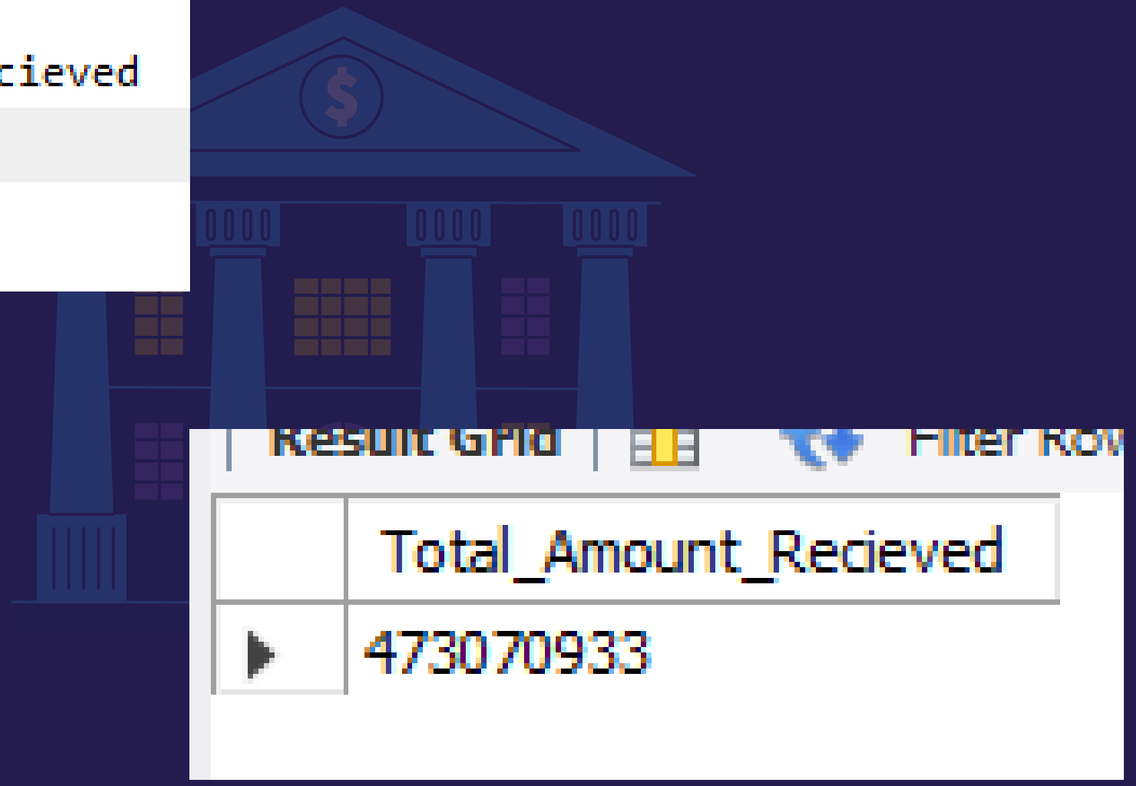
Month-over-Month (MoM) Changes In Total Funded Amount

```sql
-- Track Changes Month-over-Month (MoM) In Total Funded Amount--
select month(issue_date) Month,
sum(loan_amount) as Monthwise_Total_Funded_Amount,
LAG(SUM(loan_amount)) OVER (ORDER BY MONTH(ISSUE_DATE)) AS PreviousMonth_TotalLoanFunded,
SUM(loan_amount) - LAG(sum(loan_amount)) OVER (ORDER BY MONTH(ISSUE_DATE)) AS MoM_Change
from loan_data
group by month(issue_date)
order by 1;
```

Month	Monthwise_Total_Funded_Amount	PreviousMonth_TotalLoanFunded	MoM_Change
1	25031650	NULL	NULL
2	24647825	25031650	-383825
3	28875700	24647825	4227875
4	29800800	28875700	925100
5	31738350	29800800	1937550
6	34161475	31738350	2423125
7	35813900	34161475	1652425
8	38149600	35813900	2335700
9	40907725	38149600	2758125
10	44893800	40907725	3986075
11	47754825	44893800	2861025
12	53981425	47754825	6226600

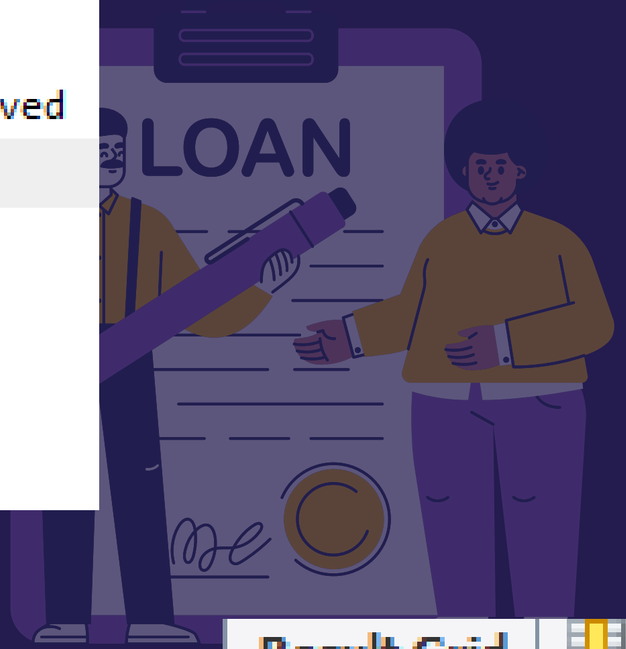
Total Amount Received from Borrowers

```sql
66      -- Total Amount Received From Borrowers --
67  •   SELECT
68        SUM(TOTAL_PAYMENT) AS Total_Amount_Recieved
69      FROM
70        LOAN_DATA;
```



Month-to-Date (MTD) Total Amount Received

```sql
-- Month-to-Date (MTD) Total Amount Received --
SELECT
  sum(TOTAL_PAYMENT) AS MTD_Total_Amount_Recieved
FROM
  LOAN_DATA
WHERE
  MONTH(ISSUE_DATE)= '12'
  AND YEAR(ISSUE_DATE)= 2021;
```



Result Grid | Filter Rows:

MTD_Total_Amount_Recieved
58074380

Month-over-Month Change (MoM) In Total Amount Recieved

```sql
-- Track Changes Month-over-Month (MoM) In Total Amount Recieved --
select month(issue_date) Month,
sum(total_payment) as Monthwise_Total_Amount_Recieved,
LAG(SUM(total_payment)) OVER (ORDER BY MONTH(ISSUE_DATE)) AS PreviousMonth_TotalAmountRecieved,
SUM(total_payment) - LAG(sum(total_payment)) OVER (ORDER BY MONTH(ISSUE_DATE)) AS MoM_Change
from loan_data
group by month(issue_date)
order by 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Month	Monthwise_Total_Amount_Recieved	PreviousMonth_TotalAmountRecieved	MoM_Change
1	27578836	NULL	NULL
2	27717745	27578836	138909
3	32264400	27717745	4546655
4	32495533	32264400	231133
5	33750523	32495533	1254990
6	36164533	33750523	2414010
7	38827220	36164533	2662687
8	42682218	38827220	3854998
9	43983948	42682218	1301730
10	49399567	43983948	5415619
11	50132030	49399567	732463
12	58074380	50132030	7942350

Difference between Received and Funded Amount

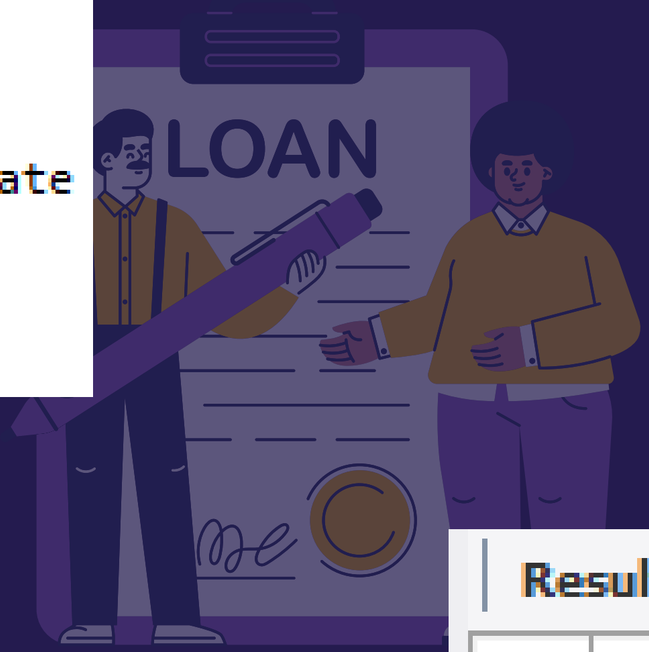
```sql
SELECT
    sum(total_payment)-SUM(LOAN_AMOUNT) AS Total_Funded_Amount
FROM
    LOAN_DATA;
```

Result Grid | Filter R

	Total_Funded_Amount
▶	37313858

Average Interest Rate charged By Bank

```sql
93      -- Average Interest Rate --
94  •   SELECT
95        ROUND(AVG(INT_RATE))*100 AS Avg_Int_Rate
96      FROM
97        LOAN_DATA;
```

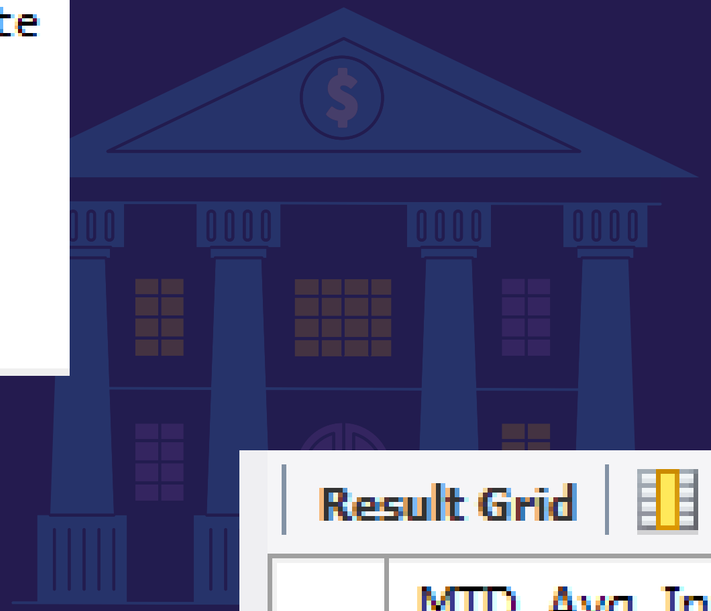


Result Grid

	Avg_Int_Rate
▶	12

MTD Average Interest Rate

```sql
-- MTD Average Interest Rate--
SELECT
    AVG(INT_RATE) * 100 AS MTD_Avg_Int_Rate
FROM
    LOAN_DATA
WHERE
    MONTH(ISSUE_DATE)='12';
```



Result Grid | Filter

	MTD_Avg_Int_Rate
▶	12.356040797403 63

Month over Month Average Interest Rate Change

```sql
-- Month over Month Average Interest Rate Change--
SELECT MONTH(ISSUE_DATE) AS Month,
AVG(INT_RATE)*100 AS Monthly_Avg_Int_Rate,
LAG(AVG(int_rate)*100) OVER (ORDER BY MONTH(ISSUE_DATE)) AS PreviousMonth_Avg_Int_Rate,
AVG(INT_RATE)*100 - LAG(AVG(int_rate)*100) OVER (ORDER BY MONTH(ISSUE_DATE)) AS MoM_Change
from loan_data
group by Month(issue_date)
ORDER BY 1 ASC;
```

Month	Monthly_Avg_Int_Rate	PreviousMonth_Avg_Int_Rate	MoM_Change
1	11.461886792452836	NULL	NULL
2	11.721632294866238	11.461886792452836	0.2597455024134021
3	11.858290826037367	11.721632294866238	0.1366585311712903
4	11.740907441016397	11.858290826037367	-0.1173833850209701
5	12.257794572311965	11.740907441016397	0.516887131295568 7
6	12.27424309045225	12.257794572311965	0.016448518140284918
7	12.23723707664888	12.27424309045225	-0.0370060138033707 6
8	12.300235396687057	12.23723707664888	0.06299832003817762
9	12.003232466063272	12.300235396687057	-0.2970029306237851 3
10	12.024122760800665	12.003232466063272	0.02089029473739323 4
11	11.941717472118764	12.024122760800665	-0.0824052886819011 1
12	12.35604079740363	11.941717472118764	0.4143233252848652 6

Average Debt-to-Income Ratio (DTI)

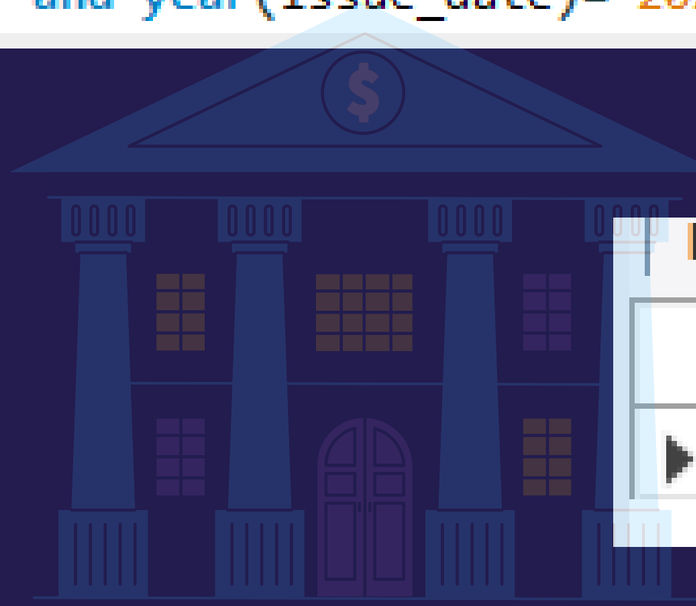
```
119     -- average DTI for borrowers --
120 •   SELECT AVG(DTI)* 100 AS Avg_DTI
121     FROM
122       LOAN_DATA;
```

Result Grid | Filter

	Avg_DTI
▶	13.327433119037712

MTD Average Debt-to-Income Ratio (DTI)

```sql
-- MTD Average Debt To Income Ratio--
SELECT AVG(DTI)* 100 AS MTD_Avg_DTI
FROM LOAN_DATA
where
month(issue_date)='12' and year(issue_date)='2021';
```



Result Grid | Filter Ro

MTD_Avg_DTI
13.665537783959245

Month over Month Average DTI Change

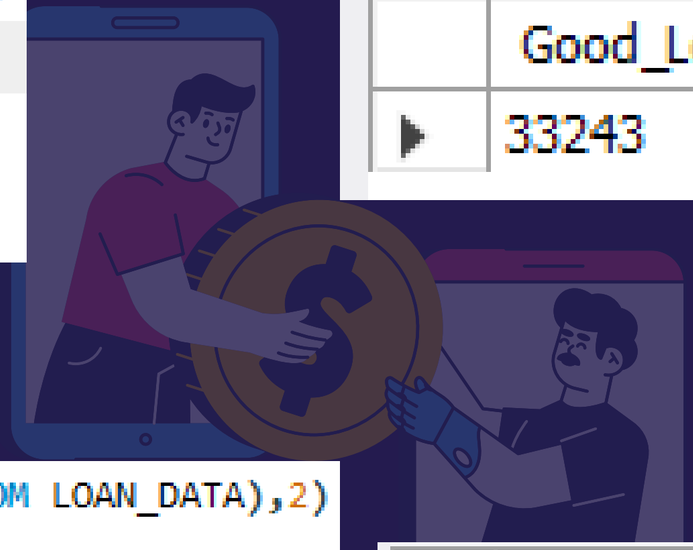
```sql
-- Month over Month Average DTI Change--
SELECT MONTH(ISSUE_DATE) AS Months,
AVG(DTI)*100 AS Monthly_Avg_DTI,
lag(AVG(DTI)*100) OVER (ORDER BY MONTH(ISSUE_DATE)) AS PreviousMonth_Avg_DTI,
AVG(DTI)*100 - lag(AVG(DTI)*100) OVER (ORDER BY MONTH(ISSUE_DATE)) AS PreviousMonth_Avg_DTI
from LOAN_DATA
GROUP BY MONTH(ISSUE_DATE)
ORDER BY 1 ASC;
```

Months	Monthly_Avg_DTI	PreviousMonth_Avg_DTI	PreviousMonth_Avg_DTI
1	12.936985420240138	NULL	NULL
2	13.409328652917935	12.936985420240138	0.47234323267779743
3	13.215614769699272	13.409328652917935	-0.19371388321866334
4	13.219361161524526	13.215614769699272	0.0037463918252544914
5	13.333744417725853	13.219361161524526	0.11438325620132694
6	13.24378140703519	13.333744417725853	-0.08996301069066348
7	13.294815805109925	13.24378140703519	0.05103439807473542
8	13.353231618715508	13.294815805109925	0.05841581360558301
9	13.297847850678746	13.353231618715508	-0.05538376803676215
10	13.414380927291909	13.297847850678746	0.11653307661316248
11	13.30273358116481	13.414380927291909	-0.11164734612709815
12	13.665537783959245	13.30273358116481	0.36280420279443426

Total No of loan applications and their % falling under the 'Good Loan' category

```sql
SELECT COUNT(ID) AS Good_Loan_Applications
FROM loan_data
WHERE loan_status= 'FULLY PAID'
OR loan_status='CURRENT';
```

	Good_Loan_Applications
▶	33243



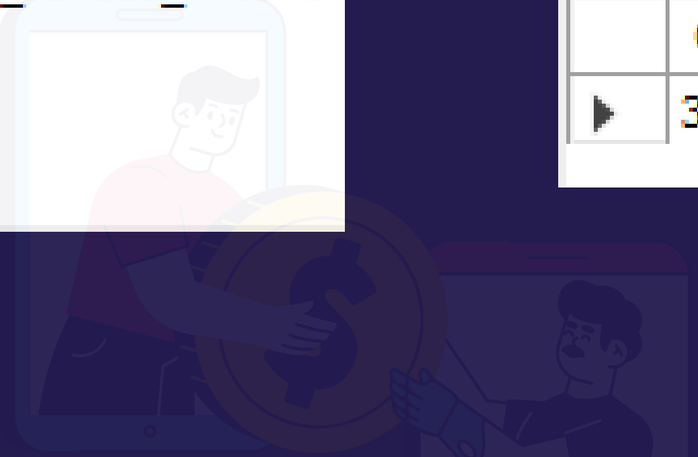
```sql
SELECT ROUND((COUNT(ID)*100)/(SELECT COUNT(ID) FROM LOAN_DATA),2)
AS Good_LoanApplication_Percentage
FROM loan_data
WHERE loan_status= 'FULLY PAID'
OR loan_status='CURRENT';
```

	Good_LoanApplication_Percentage
▶	86.18

Total amount of funds disbursed and Total Amount Received from Borrowers as Good Loans by Bank

```sql
SELECT SUM(LOAN_AMOUNT) as Good_Loan_Funded_Amount
FROM loan_data
WHERE loan_status= 'FULLY PAID'
OR loan_status='CURRENT';
```

Good_Loan_Funded_Amount
370224850



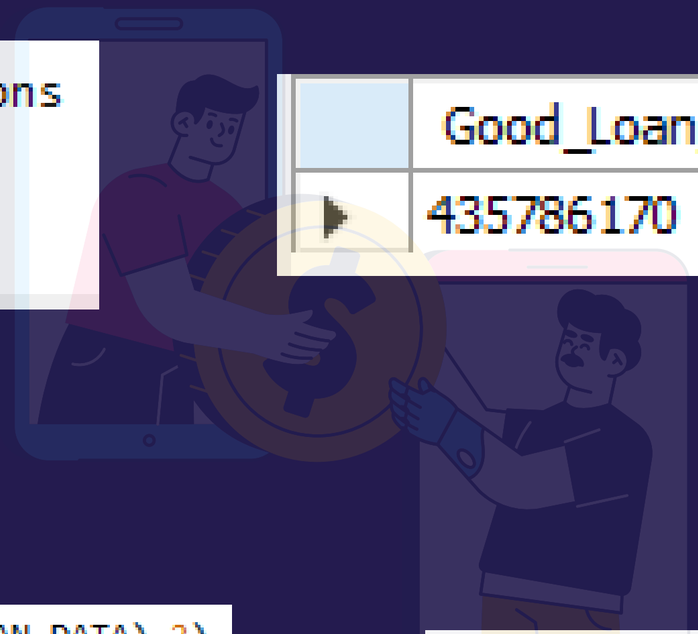
```sql
SELECT SUM(TOTAL_PAYMENT) AS Good_Loan_Recieved_Amount
FROM loan_data
WHERE loan_status= 'FULLY PAID'
OR loan_status='CURRENT';
```

Good_Loan_Recieved_Amount
435786170

Total no of loan applications and their % falling under the 'Bad Loan' category

```
182 •   SELECT COUNT(ID) Bad_Loan_Applications
183     FROM loan_data
184     WHERE loan_status='Charged Off';
```

Good_Loan_Recieved_Amount
435786170



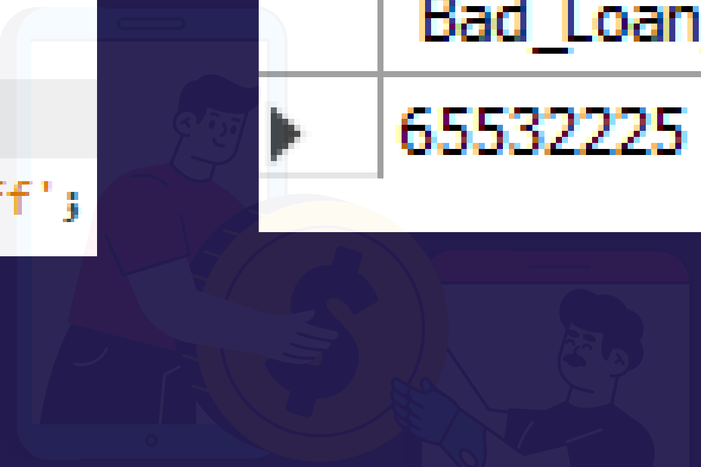
```
88 •    SELECT ROUND(COUNT(ID)*100/ (SELECT COUNT(ID) FROM LOAN_DATA),2)
89      AS Bad_Loan_Percentage
90      FROM loan_data
91      WHERE loan_status='Charged Off';
```

Bad_Loan_Percentage
13.82

Total amount of funds disbursed and Total Amount Received from Borrowers as Bad Loans by Bank

```
195 •   SELECT SUM(LOAN_AMOUNT)
196     AS Bad_Loan_Funded_Amount
197     FROM LOAN_DATA
198     WHERE loan_status='Charged Off';
```

	Bad_Loan_Funded_Amount
▶	65532225

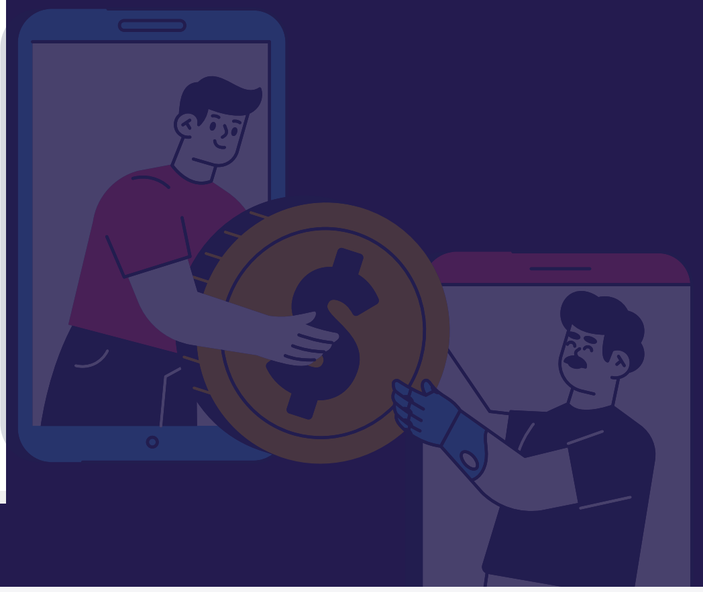


```
02 •   SELECT SUM(TOTAL_PAYMENT) AS Bad_Loan_Recieved_Amount
03     FROM loan_data
04     WHERE loan_status='Charged Off';
05
```

	Bad_Loan_Recieved_Amount
▶	37284763

Loan Status Grid View

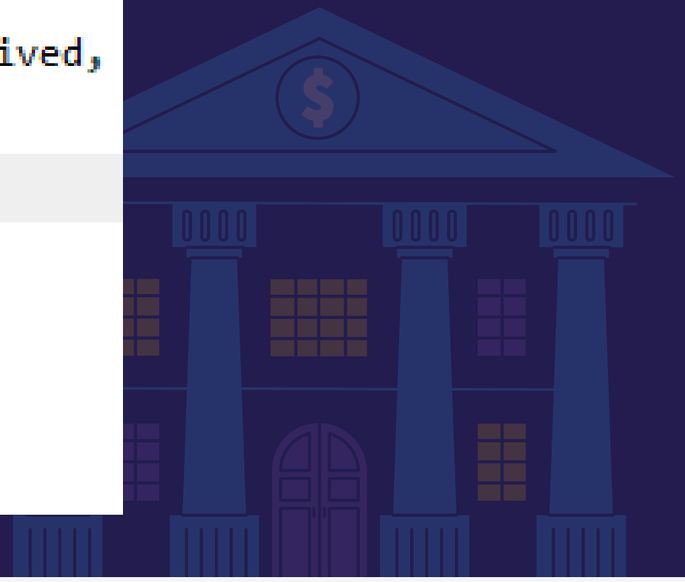
```
209 •   SELECT LOAN_STATUS,
210     COUNT(ID) AS LoanCount,
211     SUM(LOAN_AMOUNT) Total_Funded_Amount,
212     SUM(TOTAL_PAYMENT) AS Total_Amount_Received,
213     AVG(int_rate)*100 AS Interest_Rate,
214     AVG(DTI)*100 AS DTI
215     FROM loan_data
216     GROUP BY loan_status
217     ORDER BY LOANCOUNT DESC;
```



LOAN_STATUS	LoanCount	Total_Funded_Amount	Total_Amount_Received	Interest_Rate	DTI
Fully Paid	32145	351358350	411586256	11.64107077306047	13.167350754394183
Charged Off	5333	65532225	37284763	13.878574910931857	14.004732795799747
Current	1098	18866500	24199914	15.099326047358838	14.724344262295078

Month-to-Date (MTD) Total Amount Received and Total Amount Funded Loan Status wise

```sql
SELECT
    loan_status,
    SUM(total_payment) AS MTD_Total_Amount_Received,
    SUM(loan_amount) AS MTD_Total_Funded_Amount
FROM loan_data
WHERE MONTH(issue_date) = '12'
AND YEAR(issue_date)='2021'
GROUP BY loan_status
ORDER BY 2 DESC;
```



loan_status	MTD_Total_Amount_Received	MTD_Total_Funded_Amount
Fully Paid	47815851	41302025
Charged Off	5324211	8732775
Current	4934318	3946625

CALCULATE MONTH WISE TOTAL APPLICATION RECIEVED, TOTAL FUNDED AMOUNT AND TOTAL RECIEVED AMOUNT BY BANK

```
SELECT MONTH(ISSUE_DATE) AS MONTH,
MONTHNAME(ISSUE_DATE) AS Month_name,
COUNT(ID) AS TOTAL_LOAN_APPLICATIONS,
SUM(LOAN_AMOUNT) AS TOTAL_FUNDED_AMOUNT,
SUM(TOTAL_PAYMENT) AS TOTAL_AMOUNT_RECIEVED
FROM loan_data
GROUP BY MONTH(ISSUE_DATE),MONTHNAME(ISSUE_DATE)
ORDER BY MONTH(ISSUE_DATE) ASC;
```

MONTH	Month_name	TOTAL_LOAN_APPLICATIONS	TOTAL_FUNDED_AMOUNT	TOTAL_AMOUNT_RECIEVED
1	January	2332	25031650	27578836
2	February	2279	24647825	27717745
3	March	2627	28875700	32264400
4	April	2755	29800800	32495533
5	May	2911	31738350	33750523
6	June	3184	34161475	36164533
7	July	3366	35813900	38827220
8	August	3441	38149600	42682218
9	September	3536	40907725	43983948
10	October	3796	44893800	49399567
11	November	4035	47754825	50132030
12	December	4314	53981425	58074380

esult 13

CALCULATE STATEWISE TOTAL APPLICATION RECIEVED, TOTAL FUNDED AMOUNT AND TOTAL RECIEVED AMOUNT

```
252 •   SELECT address_state AS State,
253     COUNT(ID) AS TOTAL_LOAN_APPLICATIONS,
254     SUM(LOAN_AMOUNT) AS TOTAL_FUNDED_AMOUNT,
255     SUM(TOTAL_PAYMENT) AS TOTAL_AMOUNT_RECIEVED
256     FROM loan_data
257     GROUP BY address_state
258     ORDER BY STATE ASC;
259
```

State	TOTAL_LOAN_APPLICATIONS	TOTAL_FUNDED_AMOUNT	TOTAL_AMOUNT_RECIEVED
AK	78	1031800	1108570
AL	432	4949225	5492272
AR	236	2529700	2777875
AZ	833	9206000	10041986
CA	6894	78484125	83901234
CO	770	8976000	9845810
CT	730	8435575	9357612
DC	214	2652350	2921854
DE	110	1138100	1269136
FL	2773	30046125	31601905
GA	1355	15480325	16728040
HI	170	1850525	2080184
IA	5	56450	64482
ID	6	59750	65329

CALCULATE TERM WISE TOTAL APPLICATION RECIEVED, TOTAL FUNDED AMOUNT AND TOTAL RECIEVED AMOUNT

```
263 •   SELECT TERM,
264     COUNT(ID) AS TOTAL_LOAN_APPLICATIONS,
265     SUM(LOAN_AMOUNT) AS TOTAL_FUNDED_AMOUNT,
266     SUM(TOTAL_PAYMENT) AS TOTAL_AMOUNT_RECIEVED
267     FROM loan_data
268     GROUP BY TERM
269     ORDER BY 1 ASC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TERM	TOTAL_LOAN_APPLICATIONS	TOTAL_FUNDED_AMOUNT	TOTAL_AMOUNT_RECIEVED
36 months	28237	273041225	294709458
60 months	10339	162715850	178361475

CALCULATE EMPLOYEE LENGTH WISE TOTAL APPLICATION RECIEVED, TOTAL FUNDED AMOUNT AND TOTAL RECIEVED AMOUNT

```
274 •   SELECT emp_length,
275     COUNT(ID) AS TOTAL_LOAN_APPLICATIONS,
276     SUM(LOAN_AMOUNT) AS TOTAL_FUNDED_AMOUNT,
277     SUM(TOTAL_PAYMENT) AS TOTAL_AMOUNT_RECIEVED
278     FROM loan_data
279     GROUP BY emp_length
280     ORDER BY emp_length ASC;
```

emp_length	TOTAL_LOAN_APPLICATIONS	TOTAL_FUNDED_AMOUNT	TOTAL_AMOUNT_RECIEVED
< 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

CALCULATE PURPOSE WISE TOTAL APPLICATION RECIEVED, TOTAL FUNDED AMOUNT AND TOTAL RECIEVED AMOUNT

```
285 •   SELECT PURPOSE,
286     COUNT(ID) AS TOTAL_LOAN_APPLICATIONS,
287     SUM(LOAN_AMOUNT) AS TOTAL_FUNDED_AMOUNT,
288     SUM(TOTAL_PAYMENT) AS TOTAL_AMOUNT_RECIEVED
289     FROM loan_data
290     GROUP BY purpose
291     ORDER BY 3;
```

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

CALCULATE HOME OWNERSHIP WISE TOTAL APPLICATION RECIEVED, TOTAL FUNDED AMOUNT AND TOTAL RECIEVED AMOUNT BY BANK

```
96 •    SELECT home_ownership,
97      COUNT(ID) AS TOTAL_LOAN_APPLICATIONS,
98      SUM(LOAN_AMOUNT) AS TOTAL_FUNDED_AMOUNT,
99      SUM(TOTAL_PAYMENT) AS TOTAL_AMOUNT_RECIEVED
00      FROM loan_data
01      GROUP BY home_ownership
02      ORDER BY home_ownership ASC;
```

home_ownership	TOTAL_LOAN_APPLICATIONS	TOTAL_FUNDED_AMOUNT	TOTAL_AMOUNT_RECIEVED
MORTGAGE	17198	219329150	238474438
NONE	3	16800	19053
OTHER	98	1044975	1025257
OWN	2838	29597675	31729129
RENT	18439	185768475	201823056

Thank You