Portfolio Project – MyBank

Data Analysis: Bank Loan Lending Data Analytics.

SQL Script, DDL and DML Commands.

Arnav Chaturvedi

Project Role: Data Analyst – Financial Bank.

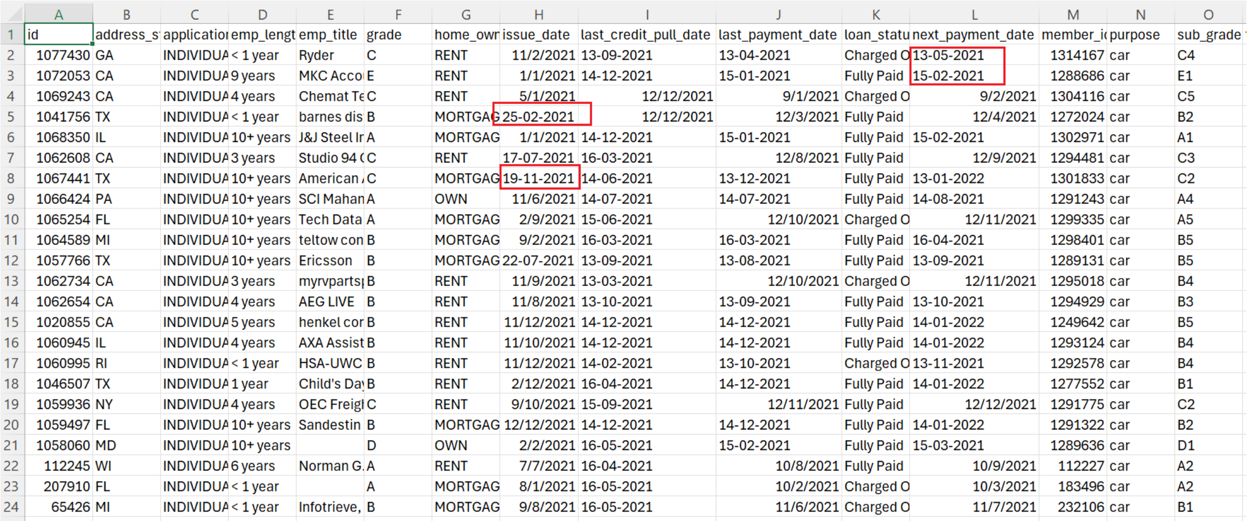
I used following applications and solution steps to import csv data file into MySQL database, performed data validation, data format conversion tasks and User Acceptance Testing planning activities.

Applications and Skills Used:

- MySQL 8.4 Server
- MySQL Workbench 8.0 CE
- Microsoft Excel
- SQL – DDL (Data Definition Language) such as Create, Alter database objects.
- SQL – DML (Data Manipulation Language) for data manipulation operations such as Select, Update, Delete etc.

Step-1: Analyze gathered data (csv) file to identify any data accuracy and formatting issues.

- Date values in gathered data file have inconstant date formats. Few values are in dd/mm/yyyy format, few others are in dd-mm-yyyy format. We need to convert them in mm/dd/yyyy format to be analyzed.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	id	address_s	applicatioi	emp_lengt	emp_title	grade	home_owr	issue_date	last_credit_pull_date	last_payment_date	loan_statu	next_payment_date	member_ic	purpose	sub_grade
2	1077430	GA	INDIVIDUA	< 1 year	Ryder	C	RENT	11/2/2021	13-09-2021	13-04-2021	Charged O	13-05-2021	1314167	car	C4
3	1072053	CA	INDIVIDUA	9 years	MKC Acco	E	RENT	1/1/2021	14-12-2021	15-01-2021	Fully Paid	15-02-2021	1288686	car	E1
4	1069243	CA	INDIVIDUA	4 years	Chemat Te	C	RENT	5/1/2021	12/12/2021	9/1/2021	Charged O	9/2/2021	1304116	car	C5
5	1041756	TX	INDIVIDUA	< 1 year	barnes dis	B	MORTGAG	25-02-2021	12/12/2021	12/3/2021	Fully Paid	12/4/2021	1272024	car	B2
6	1068350	IL	INDIVIDUA	10+ years	J&J Steel Ir	A	MORTGAG	1/1/2021	14-12-2021	15-01-2021	Fully Paid	15-02-2021	1302971	car	A1
7	1062608	CA	INDIVIDUA	3 years	Studio 94 (	C	RENT	17-07-2021	16-03-2021	12/8/2021	Fully Paid	12/9/2021	1294481	car	C3
8	1067441	TX	INDIVIDUA	10+ years	American /	C	MORTGAG	19-11-2021	14-06-2021	13-12-2021	Fully Paid	13-01-2022	1301833	car	C2
9	1066424	PA	INDIVIDUA	10+ years	SCI Mahan	A	OWN	11/6/2021	14-07-2021	14-07-2021	Fully Paid	14-08-2021	1291243	car	A4
10	1065254	FL	INDIVIDUA	10+ years	Tech Data	A	MORTGAG	2/9/2021	15-06-2021	12/10/2021	Charged O	12/11/2021	1299335	car	A5
11	1064589	MI	INDIVIDUA	10+ years	teltow con	B	MORTGAG	9/2/2021	16-03-2021	16-03-2021	Fully Paid	16-04-2021	1298401	car	B5
12	1057766	TX	INDIVIDUA	10+ years	Ericsson	B	MORTGAG	22-07-2021	13-09-2021	13-08-2021	Fully Paid	13-09-2021	1289131	car	B5
13	1062734	CA	INDIVIDUA	3 years	myrvparts	B	RENT	11/9/2021	13-03-2021	12/10/2021	Charged O	12/11/2021	1295018	car	B4
14	1062654	CA	INDIVIDUA	4 years	AEG LIVE	B	RENT	11/8/2021	13-10-2021	13-09-2021	Fully Paid	13-10-2021	1294929	car	B3
15	1020855	CA	INDIVIDUA	5 years	henkel cor	B	RENT	11/12/2021	14-12-2021	14-12-2021	Fully Paid	14-01-2022	1249642	car	B5
16	1060945	IL	INDIVIDUA	4 years	AXA Assist	B	RENT	11/10/2021	14-12-2021	14-12-2021	Fully Paid	14-01-2022	1293124	car	B4
17	1060995	RI	INDIVIDUA	< 1 year	HSA-UWC	B	RENT	11/12/2021	14-02-2021	13-10-2021	Charged O	13-11-2021	1292578	car	B4
18	1046507	TX	INDIVIDUA	1 year	Child's Day	B	RENT	2/12/2021	16-04-2021	14-12-2021	Fully Paid	14-01-2022	1277552	car	B1
19	1059936	NY	INDIVIDUA	4 years	OEC Freigl	C	RENT	9/10/2021	15-09-2021	12/11/2021	Fully Paid	12/12/2021	1291775	car	C2
20	1059497	FL	INDIVIDUA	10+ years	Sandestin	B	MORTGAG	12/12/2021	14-12-2021	14-12-2021	Fully Paid	14-01-2022	1291322	car	B2
21	1058060	MD	INDIVIDUA	10+ years		D	OWN	2/2/2021	16-05-2021	15-02-2021	Fully Paid	15-03-2021	1289636	car	D1
22	112245	WI	INDIVIDUA	6 years	Norman G.	A	RENT	7/7/2021	16-04-2021	10/8/2021	Fully Paid	10/9/2021	112227	car	A2
23	207910	FL	INDIVIDUA	< 1 year		A	MORTGAG	8/1/2021	16-05-2021	10/2/2021	Charged O	10/3/2021	183496	car	A2
24	65426	MI	INDIVIDUA	< 1 year	Infotrieve,	B	MORTGAG	9/8/2021	16-05-2021	11/6/2021	Charged O	11/7/2021	232106	car	B1

Step-2: Create MySQL database/Schema.

- Using MySQL Workbench, created new database/schema (**bankloandb**).

Step-3: Create a new database table.

- Using SQL DDL command, create a new database table (**bank_loan_data**) in this database/schema (**bankloandb**).
- This table used to import bank loan data from csv data file into mySQL database.
- Varchar datatype is being used to import from csv file into this table (temporarily).

```
CREATE TABLE bank_loan_data (
id                          int,
address_state               varchar(10),
application_type            varchar(100),
emp_length                  varchar(100),
emp_title                   varchar(150),
grade                       varchar(10),
home_ownership              varchar(100),
issue_date                  varchar(50),
last_credit_pull_date       varchar(50),
last_payment_date           varchar(50),
loan_status                 varchar(50),
next_payment_date           varchar(50),
member_id                   int,
purpose                     varchar(100),
sub_grade                   varchar(10),
term                        varchar(100),
verification_status         varchar(100),
annual_income               decimal(7,0),
dti                         double(7,4),
installment                 decimal(7,2),
int_rate                    double(7,4),
loan_amount                 decimal(7,2),
total_acc                   int,
total_payment               decimal(7,2));
```

Step-4: Import csv data file into database table.

- Using MySQL Workbench’s ‘Table Data Import Wizard’, import csv file data into table.
- Used proper field separator (,), Line separator (CR LF), Encoding (utf-8), Column mappings.

Step-5: Alter database table for date format conversion.

- Using SQL DDL command, ALTER the table to add new date datatype columns temporarily which will be used to convert date value's format.

```
ALTER TABLE bank_loan_data ADD COLUMN temp_issue_date date;
ALTER TABLE bank_loan_data ADD COLUMN temp_last_credit_pull_date date;
ALTER TABLE bank_loan_data ADD COLUMN temp_last_payment_date date;
ALTER TABLE bank_loan_data ADD COLUMN temp_next_payment_date date;
```

Step-6: Update column values for date format conversion.

- Using SQL DML commands, populate new columns with converted values.
- Convert from Varchar/string datatype to date datatype.

```
UPDATE bank_loan_data SET temp_issue_date = STR_TO_DATE(issue_date, '%d/%m/%Y');
UPDATE bank_loan_data SET temp_last_credit_pull_date = STR_TO_DATE(last_credit_pull_date,
'%d/%m/%Y');
UPDATE bank_loan_data SET temp_last_payment_date = STR_TO_DATE(last_payment_date,
'%d/%m/%Y');
UPDATE bank_loan_data SET temp_next_payment_date = STR_TO_DATE(next_payment_date,
'%d/%m/%Y');
```

Step-7: Alter database table for date format conversion.

- Using SQL DDL command, ALTER the table to drop old Varchar datatype columns.

```
ALTER TABLE bank_loan_data DROP COLUMN issue_date;
ALTER TABLE bank_loan_data DROP COLUMN last_credit_pull_date;
ALTER TABLE bank_loan_data DROP COLUMN last_payment_date;
ALTER TABLE bank_loan_data DROP COLUMN next_payment_date;
```

Step-8: Alter database table for date format conversion.

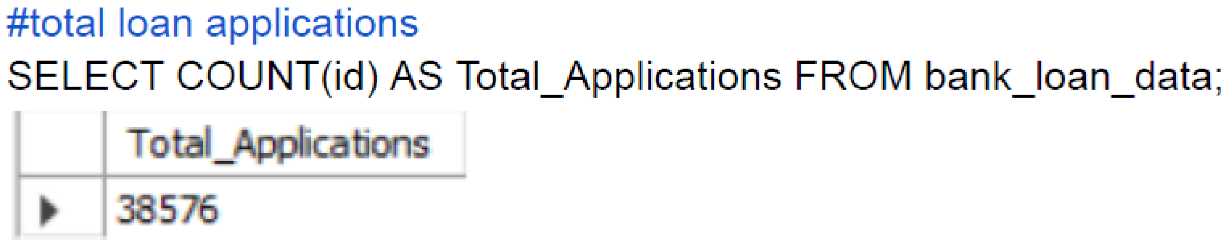
- Using SQL DDL command, ALTER the table to rename the new DATE datatype columns back to original columns names.

```
ALTER TABLE bank_loan_data CHANGE COLUMN temp_issue_date issue_date date;
ALTER TABLE bank_loan_data CHANGE COLUMN temp_last_credit_pull_date last_credit_pull_date
date;
ALTER TABLE bank_loan_data CHANGE COLUMN temp_last_payment_date last_payment_date
date;
ALTER TABLE bank_loan_data CHANGE COLUMN temp_next_payment_date next_payment_date
date;
```

Now, we have all imported data accurately in table which can be used to create Tableau dashboards. Source data can be used either from MySQL database table (bank_loan_data) or directly from csv data file.

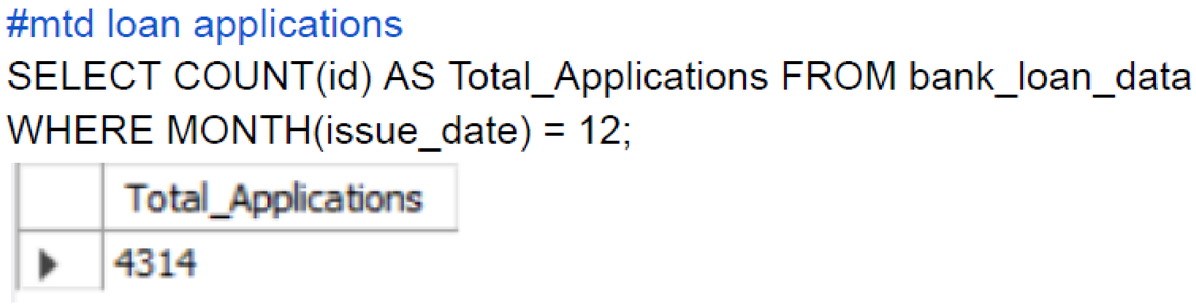
Step-9: After data import, Data validation and UAT preparation.

- Using SQL DML commands, retrieve imported data from table. Write few SELECT queries to get data which will be matched with data shown on Tableau dashboard (during UAT with business users).
- SQL output values should match with related KPI's shown on Tableau dashboard.
- Please refer SQL queries given below.



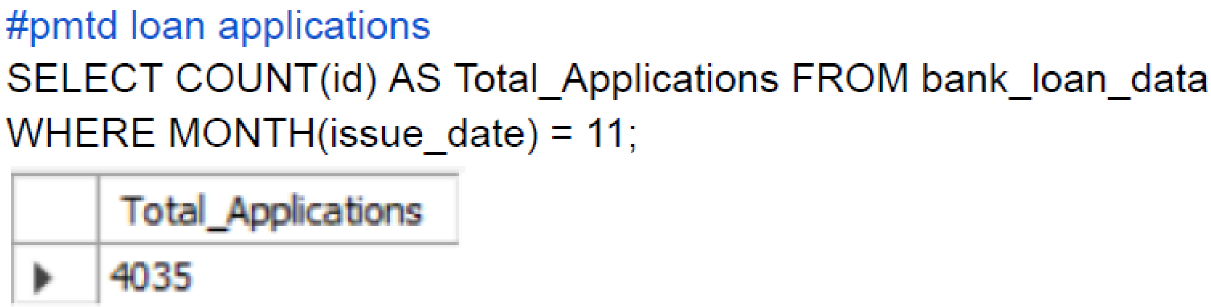
```
#total loan applications
SELECT COUNT(id) AS Total_Applications FROM bank_loan_data;
```

```
   Total_Applications
▶  38576
```



```
#mtd loan applications
SELECT COUNT(id) AS Total_Applications FROM bank_loan_data
WHERE MONTH(issue_date) = 12;
```

```
   Total_Applications
▶  4314
```

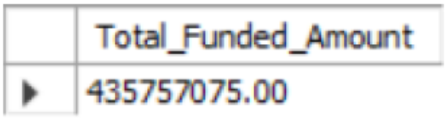


```
#pmtd loan applications
SELECT COUNT(id) AS Total_Applications FROM bank_loan_data
WHERE MONTH(issue_date) = 11;
```

```
   Total_Applications
▶  4035
```

#total funded amount

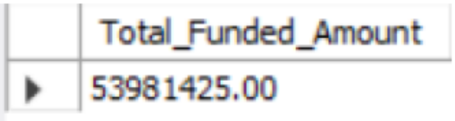
SELECT SUM(loan_amount) AS Total_Funded_Amount FROM bank_loan_data;



	Total_Funded_Amount
▶	435757075.00

#mtd total funded amount

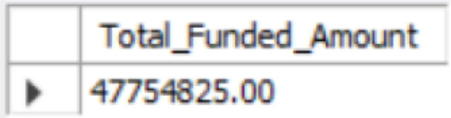
SELECT SUM(loan_amount) AS Total_Funded_Amount FROM bank_loan_data WHERE MONTH(issue_date) = 12;



	Total_Funded_Amount
▶	53981425.00

#pmtd total funded amount

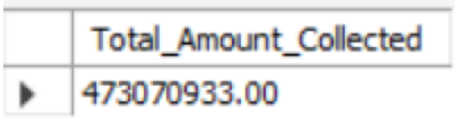
SELECT SUM(loan_amount) AS Total_Funded_Amount FROM bank_loan_data WHERE MONTH(issue_date) = 11;



	Total_Funded_Amount
▶	47754825.00

#total amount received

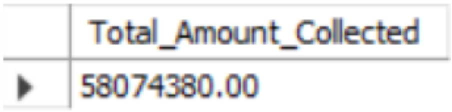
SELECT SUM(total_payment) AS Total_Amount_Collected FROM bank_loan_data;



	Total_Amount_Collected
▶	473070933.00

#mtd total amount received

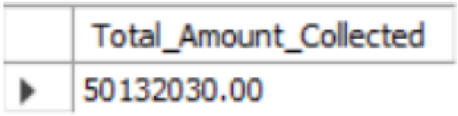
SELECT SUM(total_payment) AS Total_Amount_Collected FROM bank_loan_data WHERE MONTH(issue_date) = 12;



	Total_Amount_Collected
▶	58074380.00

#pmtd total amount received

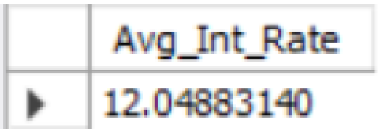
SELECT SUM(total_payment) AS Total_Amount_Collected FROM bank_loan_data WHERE MONTH(issue_date) = 11;



	Total_Amount_Collected
▶	50132030.00

#average interest rate

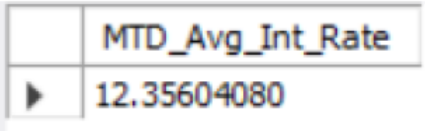
SELECT AVG(int_rate)*100 AS Avg_Int_Rate FROM bank_loan_data;



Avg_Int_Rate
12.04883140

#mtd average interest

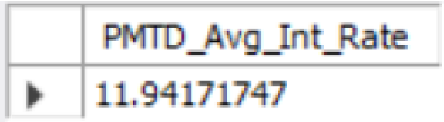
SELECT AVG(int_rate)*100 AS MTD_Avg_Int_Rate FROM bank_loan_data WHERE MONTH(issue_date) = 12;



MTD_Avg_Int_Rate
12.35604080

#pmtd average interest

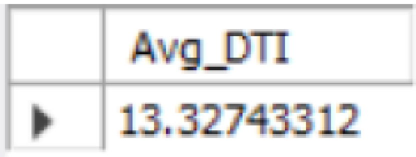
SELECT AVG(int_rate)*100 AS PMTD_Avg_Int_Rate FROM bank_loan_data WHERE MONTH(issue_date) = 11;



PMTD_Avg_Int_Rate
11.94171747

#average dti

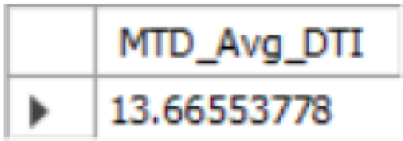
SELECT AVG(dti)*100 AS Avg_DTI FROM bank_loan_data;



Avg_DTI
13.32743312

#mtd average dti

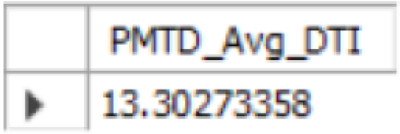
SELECT AVG(dti)*100 AS MTD_Avg_DTI FROM bank_loan_data WHERE MONTH(issue_date) = 12;



MTD_Avg_DTI
13.66553778

#pmtd average dti

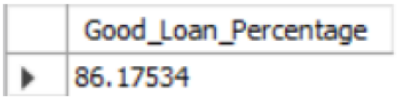
```
SELECT AVG(dti)*100 AS PMTD_Avg_DTI FROM bank_loan_data
WHERE MONTH(issue_date) = 11;
```



PMTD_Avg_DTI
13.30273358

#good loan percentage

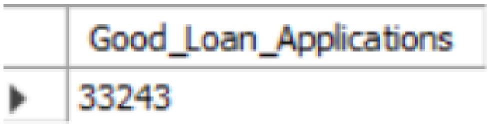
```
SELECT
    (COUNT(CASE WHEN loan_status = 'Fully Paid' OR loan_status = 'Current' THEN id END) *
100.0) / COUNT(id) AS Good_Loan_Percentage
FROM bank_loan_data;
```



Good_Loan_Percentage
86.17534

#good loan applications

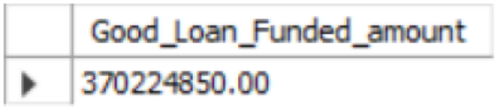
```
SELECT COUNT(id) AS Good_Loan_Applications FROM bank_loan_data
WHERE loan_status = 'Fully Paid' OR loan_status = 'Current';
```



Good_Loan_Applications
33243

#good loan funded amount

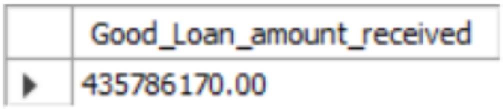
```
SELECT SUM(loan_amount) AS Good_Loan_Funded_amount FROM bank_loan_data
WHERE loan_status = 'Fully Paid' OR loan_status = 'Current';
```



Good_Loan_Funded_amount
370224850.00

#good loan amount received

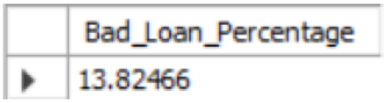
```
SELECT SUM(total_payment) AS Good_Loan_amount_received FROM bank_loan_data
WHERE loan_status = 'Fully Paid' OR loan_status = 'Current';
```



Good_Loan_amount_received
435786170.00

#bad loan percentage

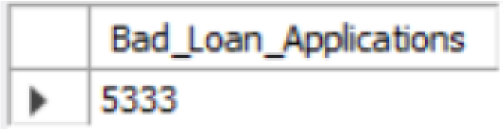
```
SELECT
   (COUNT(CASE WHEN loan_status = 'Charged Off' THEN id END) * 100.0) / COUNT(id) AS
Bad_Loan_Percentage
FROM bank_loan_data;
```



	Bad_Loan_Percentage
▶	13.82466

#bad loan applications

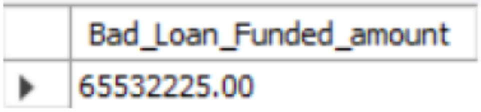
```
SELECT COUNT(id) AS Bad_Loan_Applications FROM bank_loan_data
WHERE loan_status = 'Charged Off';
```



	Bad_Loan_Applications
▶	5333

#bad loan funded amount

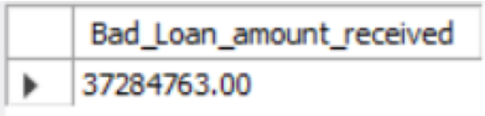
```
SELECT SUM(loan_amount) AS Bad_Loan_Funded_amount FROM bank_loan_data
WHERE loan_status = 'Charged Off';
```



	Bad_Loan_Funded_amount
▶	65532225.00

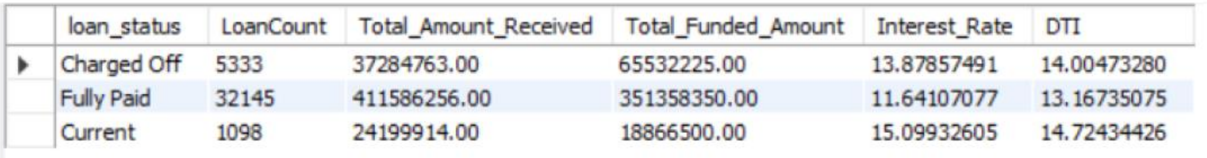
#bad loan amount received

```
SELECT SUM(total_payment) AS Bad_Loan_amount_received FROM bank_loan_data
WHERE loan_status = 'Charged Off';
```



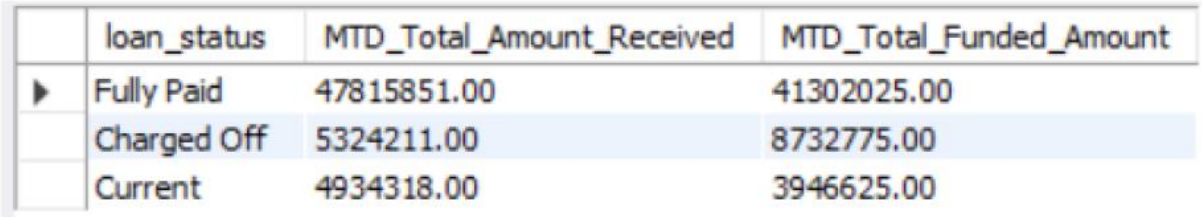
	Bad_Loan_amount_received
▶	37284763.00

```
#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_loan_data
GROUP BY
        loan_status;
```



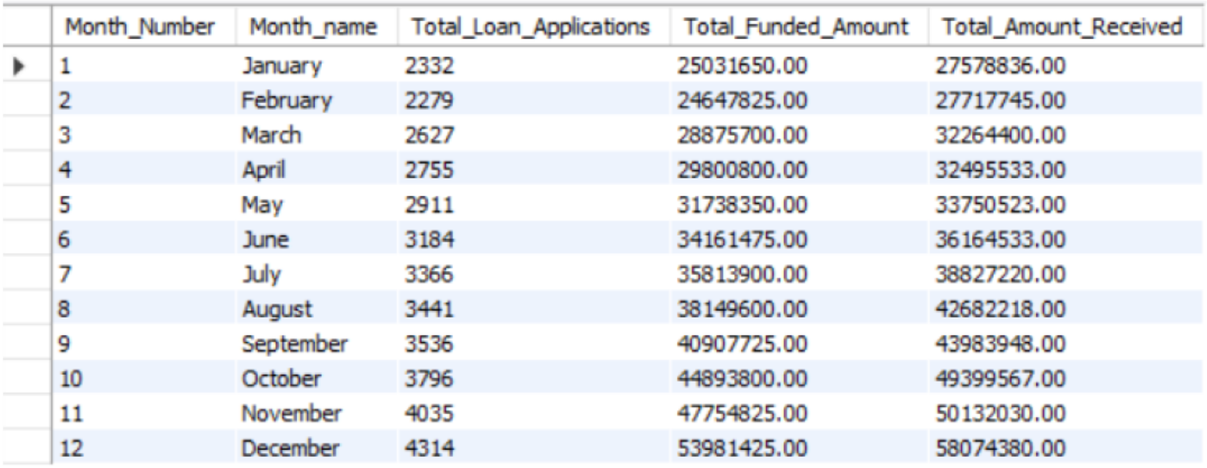
loan_status	LoanCount	Total_Amount_Received	Total_Funded_Amount	Interest_Rate	DTI
Charged Off	5333	37284763.00	65532225.00	13.87857491	14.00473280
Fully Paid	32145	411586256.00	351358350.00	11.64107077	13.16735075
Current	1098	24199914.00	18866500.00	15.09932605	14.72434426

```
SELECT
        loan_status,
        SUM(total_payment) AS MTD_Total_Amount_Received,
        SUM(loan_amount) AS MTD_Total_Funded_Amount
FROM bank_loan_data
WHERE MONTH(issue_date) = 12
GROUP BY loan_status;
```



loan_status	MTD_Total_Amount_Received	MTD_Total_Funded_Amount
Fully Paid	47815851.00	41302025.00
Charged Off	5324211.00	8732775.00
Current	4934318.00	3946625.00

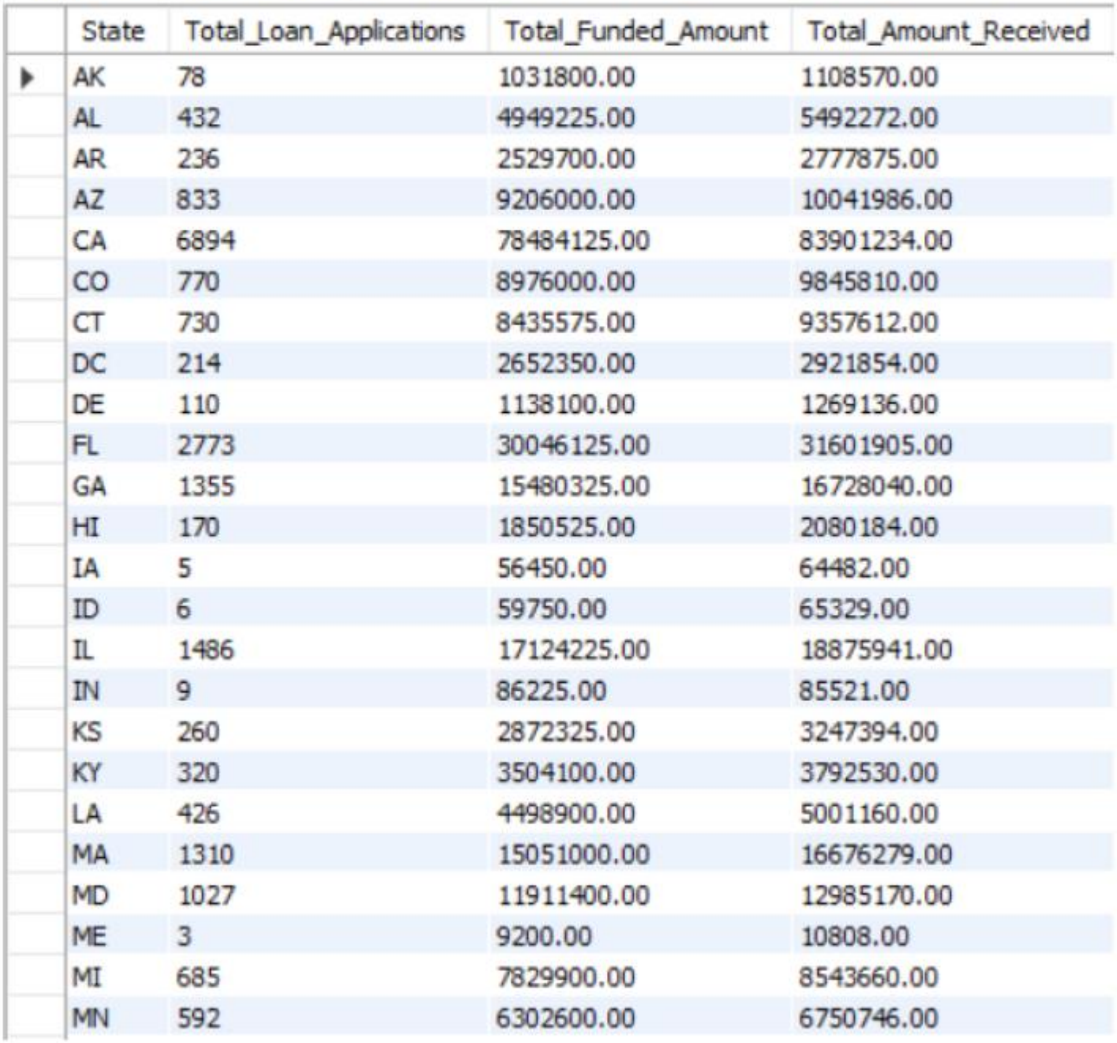
```
#month
SELECT
        MONTH(issue_date) AS Month_Number,
        MONTHNAME(issue_date) AS Month_name,
        COUNT(id) AS Total_Loan_Applications,
        SUM(loan_amount) AS Total_Funded_Amount,
        SUM(total_payment) AS Total_Amount_Received
FROM bank_loan_data
GROUP BY MONTH(issue_date), MONTHNAME(issue_date)
ORDER BY MONTH(issue_date);
```



Month_Number	Month_name	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
1	January	2332	25031650.00	27578836.00
2	February	2279	24647825.00	27717745.00
3	March	2627	28875700.00	32264400.00
4	April	2755	29800800.00	32495533.00
5	May	2911	31738350.00	33750523.00
6	June	3184	34161475.00	36164533.00
7	July	3366	35813900.00	38827220.00
8	August	3441	38149600.00	42682218.00
9	September	3536	40907725.00	43983948.00
10	October	3796	44893800.00	49399567.00
11	November	4035	47754825.00	50132030.00
12	December	4314	53981425.00	58074380.00

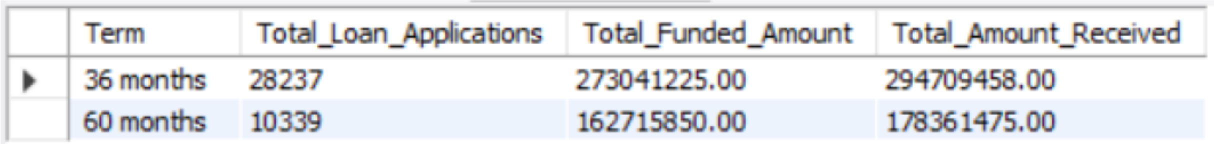
#state

```
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_loan_data
GROUP BY address_state
ORDER BY address_state;
```



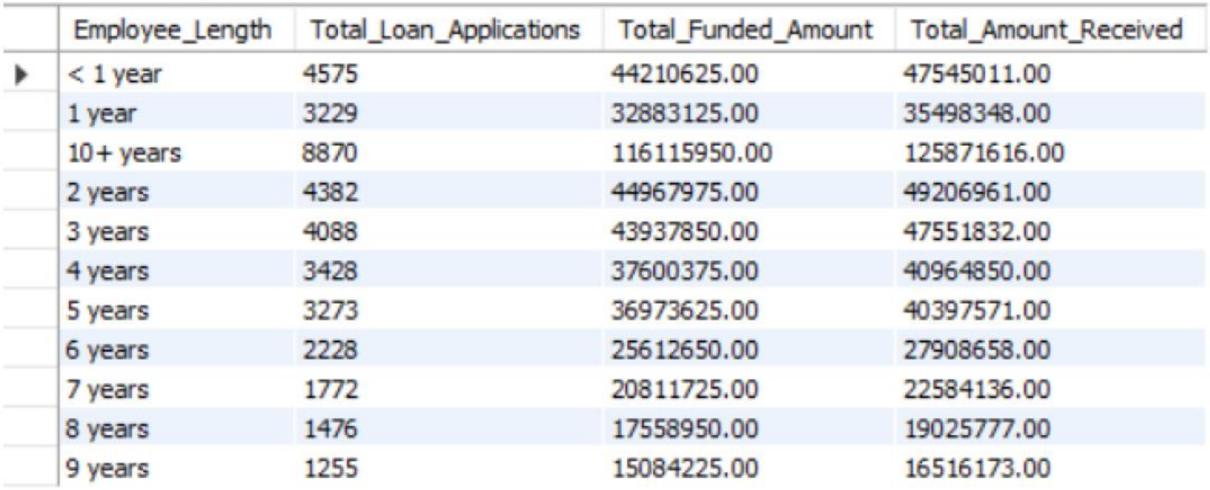
State	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
AK	78	1031800.00	1108570.00
AL	432	4949225.00	5492272.00
AR	236	2529700.00	2777875.00
AZ	833	9206000.00	10041986.00
CA	6894	78484125.00	83901234.00
CO	770	8976000.00	9845810.00
CT	730	8435575.00	9357612.00
DC	214	2652350.00	2921854.00
DE	110	1138100.00	1269136.00
FL	2773	30046125.00	31601905.00
GA	1355	15480325.00	16728040.00
HI	170	1850525.00	2080184.00
IA	5	56450.00	64482.00
ID	6	59750.00	65329.00
IL	1486	17124225.00	18875941.00
IN	9	86225.00	85521.00
KS	260	2872325.00	3247394.00
KY	320	3504100.00	3792530.00
LA	426	4498900.00	5001160.00
MA	1310	15051000.00	16676279.00
MD	1027	11911400.00	12985170.00
ME	3	9200.00	10808.00
MI	685	7829900.00	8543660.00
MN	592	6302600.00	6750746.00

```
#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_loan_data
GROUP BY term
ORDER BY term;
```



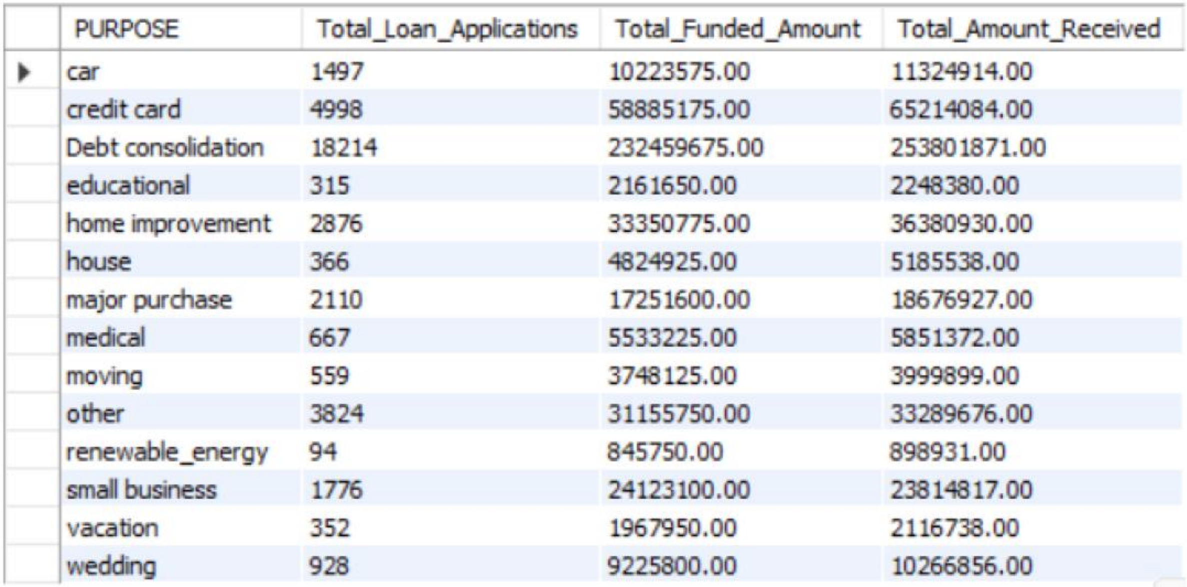
	Term	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
▶	36 months	28237	273041225.00	294709458.00
	60 months	10339	162715850.00	178361475.00

```
#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_loan_data
GROUP BY emp_length
ORDER BY emp_length;
```



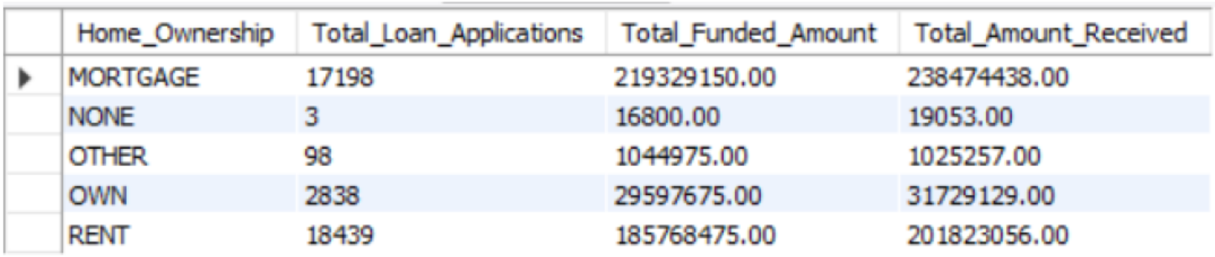
	Employee_Length	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
▶	< 1 year	4575	44210625.00	47545011.00
	1 year	3229	32883125.00	35498348.00
	10+ years	8870	116115950.00	125871616.00
	2 years	4382	44967975.00	49206961.00
	3 years	4088	43937850.00	47551832.00
	4 years	3428	37600375.00	40964850.00
	5 years	3273	36973625.00	40397571.00
	6 years	2228	25612650.00	27908658.00
	7 years	1772	20811725.00	22584136.00
	8 years	1476	17558950.00	19025777.00
	9 years	1255	15084225.00	16516173.00

```
#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_loan_data
GROUP BY purpose
ORDER BY purpose;
```



PURPOSE	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
car	1497	10223575.00	11324914.00
credit card	4998	58885175.00	65214084.00
Debt consolidation	18214	232459675.00	253801871.00
educational	315	2161650.00	2248380.00
home improvement	2876	33350775.00	36380930.00
house	366	4824925.00	5185538.00
major purchase	2110	17251600.00	18676927.00
medical	667	5533225.00	5851372.00
moving	559	3748125.00	3999899.00
other	3824	31155750.00	33289676.00
renewable_energy	94	845750.00	898931.00
small business	1776	24123100.00	23814817.00
vacation	352	1967950.00	2116738.00
wedding	928	9225800.00	10266856.00

```
#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_loan_data
GROUP BY home_ownership
ORDER BY home_ownership;
```



Home_Ownership	Total_Loan_Applications	Total_Funded_Amount	Total_Amount_Received
MORTGAGE	17198	219329150.00	238474438.00
NONE	3	16800.00	19053.00
OTHER	98	1044975.00	1025257.00
OWN	2838	29597675.00	31729129.00
RENT	18439	185768475.00	201823056.00