Bank Loan Analysis – SQL Query Report

Executive Summary

This report compiles the SQL queries used to generate key metrics for a bank loan dataset. The analysis calculates various performance indicators (such as total loan applications, funded amounts, interest rates, etc.) using SQL on the financial loan data. The purpose of this report is to organize and explain each SQL query clearly, providing a plain-English description and sample output for an interviewer or stakeholder. The findings from these queries support data-driven decisions by highlighting portfolio trends and borrower health.

Table of Contents

Loan Applications Summary

This section summarizes overall loan application and funding metrics. Each query below targets a different KPI.

```sql
SELECT COUNT(*) AS Total_Loan_Applications
FROM financial_loan;
```

This query counts all records in the `financial_loan` table to find the total number of loan applications.

Total_Loan_Applications
15,276

```sql
SELECT COUNT(*) AS MTD_Loan_Applications
FROM financial_loan
WHERE MONTH(issue_date) = 12 AND YEAR(issue_date) = 2021;
```

This query counts applications issued in a specific month (e.g., December 2021) for the month-to-date (MTD) metric.

MTD_Loan_Applications
1,204

```sql
SELECT COUNT(*) AS PMTD_Loan_Applications
FROM financial_loan
WHERE MONTH(issue_date) = 11 AND YEAR(issue_date) = 2021;
```

This query counts applications from the previous month (e.g., November 2021) for a prior-MTD comparison.

PMTD_Loan_Applications
1,150

```sql
SELECT SUM(loan_amount) AS Total_Funded_Amount
FROM financial_loan;
```

This query calculates the total funded loan amount across all applications.

Total_Funded_Amount
250,000,000

```sql
SELECT SUM(loan_amount) AS MTD_Funded_Amount
FROM financial_loan
WHERE MONTH(issue_date) = 12 AND YEAR(issue_date) = 2021;
```

This query sums the loan amounts funded in the specified month (MTD).

MTD_Funded_Amount
20,500,000

```sql
SELECT SUM(loan_amount) AS PMTD_Funded_Amount
FROM financial_loan
WHERE MONTH(issue_date) = 11 AND YEAR(issue_date) = 2021;
```

This query sums the loan amounts from the previous month (PMTD) for comparison.

PMTD_Funded_Amount
18,750,000

```sql
SELECT SUM(total_payment) AS Total_Amount_Received
FROM financial_loan;
```

This query calculates the total amount received from borrowers (sum of all payments to date).

Total_Amount_Received
180,000,000

```sql
SELECT SUM(total_payment) AS MTD_Amount_Received
FROM financial_loan
WHERE MONTH(issue_date) = 12 AND YEAR(issue_date) = 2021;
```

This query sums payments received in the specified month (MTD).

MTD_Amount_Received
15,200,000

```sql
SELECT SUM(total_payment) AS PMTD_Amount_Received
FROM financial_loan
WHERE MONTH(issue_date) = 11 AND YEAR(issue_date) = 2021;
```

This query sums payments from the previous month (PMTD).

PMTD_Amount_Received
14,600,000

```sql
SELECT AVG(int_rate) * 100 AS Avg_Interest_Rate
FROM financial_loan;
```

This query computes the average interest rate (as a percentage) across all loans.

Avg_Interest_Rate
10.25

```
SELECT AVG(int_rate) * 100 AS MTD_Avg_Interest
FROM financial_loan
WHERE MONTH(issue_date) = 12 AND YEAR(issue_date) = 2021;
```

This query finds the average interest rate for loans issued in the specified month (MTD).

MTD_Avg_Interest
10.40

```
SELECT AVG(int_rate) * 100 AS PMTD_Avg_Interest
FROM financial_loan
WHERE MONTH(issue_date) = 11 AND YEAR(issue_date) = 2021;
```

This query finds the average interest rate for loans issued in the previous month (PMTD).

PMTD_Avg_Interest
10.10

```
SELECT AVG(dti) * 100 AS Avg_DTI
FROM financial_loan;
```

This query calculates the average debt-to-income ratio (in percentage) across all borrowers.

Avg_DTI
18.75

```
SELECT AVG(dti) * 100 AS MTD_Avg_DTI
FROM financial_loan
WHERE MONTH(issue_date) = 12 AND YEAR(issue_date) = 2021;
```

This query computes the average DTI for loans issued in the specified month (MTD).

MTD_Avg_DTI
19.10

```sql
SELECT AVG(dti) * 100 AS PMTD_Avg_DTI
FROM financial_loan
WHERE MONTH(issue_date) = 11 AND YEAR(issue_date) = 2021;
```

This query computes the average DTI for loans issued in the previous month (PMTD).

PMTD_Avg_DTI
18.50

```sql
SELECT DISTINCT loan_status
FROM financial_loan;
```

This query lists all distinct loan status values present in the data (e.g., “Fully Paid”, “Current”, “Charged Off”, etc.).

loan_status
Fully Paid
Current
Charged Off
Late (31-120 days)
Late (16-30 days)

Good Loans

This section focuses on “good” loans (typically loans that are current or fully paid).

```sql
SELECT
  (COUNT(CASE WHEN loan_status IN ('Fully Paid','Current') THEN 1 END) * 100.
0) / COUNT(*) AS Good_Loan_Percentage
FROM financial_loan;
```

This query calculates the percentage of loans that are in “Fully Paid” or “Current” status (deemed good loans).

Good_Loan_Percentage
89.4

```
SELECT COUNT(*) AS Good_Loan_Applications
FROM financial_loan
WHERE loan_status IN ('Fully Paid','Current');
```

This query counts the number of loan applications considered good (fully paid or current).

Good_Loan_Applications
13,640

```
SELECT SUM(loan_amount) AS Good_Loan_Funded_Amount
FROM financial_loan
WHERE loan_status IN ('Fully Paid','Current');
```

This query sums the funded amount for all good loans.

Good_Loan_Funded_Amount
230,000,000

```
SELECT SUM(total_payment) AS Good_Loan_Amount_Received
FROM financial_loan
WHERE loan_status IN ('Fully Paid','Current');
```

This query sums the total payments received for all good loans.

Good_Loan_Amount_Received
170,000,000

Bad Loans

This section covers loans that are “bad” (e.g., charged off).

```
SELECT
  (COUNT(CASE WHEN loan_status = 'Charged Off' THEN 1 END) * 100.0) / COUNT(*)
AS Bad_Loan_Percentage
FROM financial_loan;
```

This query calculates the percentage of loans that are “Charged Off” (considered bad loans).

Bad_Loan_Percentage
6.5

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

This query counts the number of bad loan applications (charged off).

Bad_Loan_Applications
990

```
SELECT SUM(loan_amount) AS Bad_Loan_Funded_Amount
FROM financial_loan
WHERE loan_status = 'Charged Off';
```

This query sums the funded amount for all bad loans.

Bad_Loan_Funded_Amount
12,000,000

```
SELECT SUM(total_payment) AS Bad_Loan_Amount_Received
FROM financial_loan
WHERE loan_status = 'Charged Off';
```

This query sums the total payments received for all bad loans (likely much lower than funded amount).

Bad_Loan_Amount_Received
1,800,000

Loan Status Reports

This section provides breakdowns of loan data by status.

```
SELECT loan_status,
       COUNT(*) AS Total_Applications,
       SUM(loan_amount) AS Total_Funded_Amount,
       SUM(total_payment) AS Total_Amount_Received,
       ROUND(AVG(int_rate * 100),2) AS Average_Interest_Rate,
       ROUND(AVG(dti * 100),2) AS Average_DTI
FROM financial_loan
GROUP BY loan_status;
```

This query groups the loans by status and calculates total applications, funded amount, total received, average interest rate, and average DTI for each status.

loan_status	Total_Applications	Total_Funded_Amount	Total_Amount_Received	Average_Interest_Rate	Average_
Fully Paid	8,500	120,000,000	95,000,000	9.80	17.50
Current	5,140	110,000,000	75,000,000	10.10	20.25
Charged Off	990	12,000,000	1,800,000	15.25	35.00
Late (16-30 days)	300	5,000,000	500,000	12.00	25.00
Late (31-120 days)	200	3,000,000	200,000	13.50	30.00

```
SELECT loan_status,
       SUM(total_payment) AS MTD_Total_Received_Amount,
       SUM(loan_amount) AS MTD_Total_Funded_Amount
FROM financial_loan
WHERE MONTH(issue_date) = 12 AND YEAR(issue_date) = 2021
GROUP BY loan_status;
```

This query provides the total received and funded amounts by loan status for the specified month (MTD).

loan_status	MTD_Total_Received_Amount	MTD_Total_Funded_Amount
Fully Paid	6,000,000	7,500,000
Current	3,800,000	6,000,000
Charged Off	100,000	1,000,000
Late (16-30)	50,000	300,000
Late (31-120)	20,000	200,000

Conclusion

This document presents a structured set of SQL queries for analyzing the bank’s loan portfolio. Each query is clearly formatted and accompanied by an explanation and sample output, facilitating review by interviewers or stakeholders. The queries cover overall summary metrics, detailed breakdowns of good vs. bad loans, and status-based summaries. Together, they illustrate how SQL is used to compute KPIs and derive insights that support data-driven decisions. The report is formatted in a clean, professional style appropriate for corporate review, ensuring clarity and consistency of information.