Query Document for Financial Loan Report

Report Requirements:

* Total Loan Applications:

SELECT COUNT(\*) AS Tot\_loan\_Apps FROM financial\_loan;

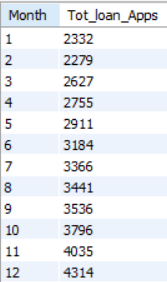


* Total Loan Applications per Month:

SELECT month(issue\_date) as Month, COUNT(id) AS Tot\_loan\_Apps FROM financial\_loan

GROUP BY month(issue\_date)

ORDER BY month(issue\_date);



* Month over Month Growth Percentage of Loan Applications:

WITH cte AS(

SELECT MONTH(issue\_date) as Month, COUNT(id) AS Tot\_loan\_Apps

FROM financial\_loan

GROUP BY MONTH (issue\_date)

ORDER BY MONTH (issue\_date)),

cte2 as (

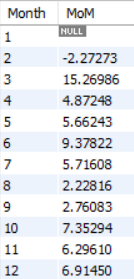
SELECT Month, LAG(Tot\_loan\_Apps,1,0) OVER(order by month) AS Prev\_month

FROM cte)

SELECT cte.Month, (Tot\_loan\_Apps - Prev\_month)\*100.0/Prev\_month as MoM

FROM cte

JOIN cte2 ON cte2.Month = cte.Month;



* Total Funded Amount by the Bank:

SELECT SUM(loan\_amount) AS Tot\_amt\_funded

FROM financial\_loan;



* Month over Month Growth Percentage of Funded Amount:

WITH cte AS(

SELECT MONTH(issue\_date) as Month, SUM(loan\_amount) AS Funded\_amt\_per\_mon

FROM financial\_loan

GROUP BY MONTH (issue\_date)

ORDER BY MONTH (issue\_date)),

cte2 as (

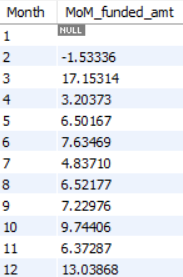
SELECT Month, LAG(Funded\_amt\_per\_mon,1,0) OVER(order by month) AS Prev\_month\_funded\_amt

FROM cte)

SELECT cte.Month, (Funded\_amt\_per\_mon - Prev\_month\_funded\_amt)\*100.0/Prev\_month\_funded\_amt as MoM\_funded\_amt

FROM cte

JOIN cte2 ON cte2.Month = cte.Month;



* Total Amount Received by the Bank:

SELECT SUM(total\_payment) as tot\_amt\_rec

FROM financial\_loan;



* Payment Received Month over Month Percentage:

WITH cte AS(

SELECT MONTH(issue\_date) as Month, SUM(total\_payment) AS rec\_amt\_per\_mon

FROM financial\_loan

GROUP BY MONTH (issue\_date)

ORDER BY MONTH (issue\_date)),

cte2 as (

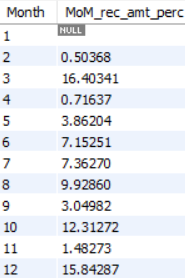
SELECT Month, LAG(rec\_amt\_per\_mon,1,0) OVER(order by month) AS Prev\_month\_rec\_amt

FROM cte)

SELECT cte.Month, (rec\_amt\_per\_mon - Prev\_month\_rec\_amt)\*100.0/Prev\_month\_rec\_amt as MoM\_rec\_amt\_perc

FROM cte

JOIN cte2 ON cte2.Month = cte.Month;



* Average Interest Rate:

SELECT ROUND(AVG(int\_rate),4) as avg\_int\_rate

FROM financial\_loan;



The average interest rate is 12%

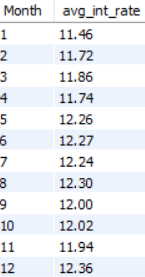
* Average Interest Rate per Month:

SELECT MONTH(issue\_date) as Month, CAST(AVG(int\_rate)\*100 AS DECIMAL(4,2)) AS avg\_int\_rate

FROM financial\_loan

GROUP BY MONTH (issue\_date)

ORDER BY MONTH (issue\_date);



* Average Debt to Income Ratio:

SELECT ROUND(AVG(dti),4) AS avg\_dti

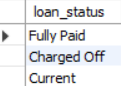
FROM financial\_loan;



* Categories In Loan Status:

SELECT DISTINCT loan\_status

FROM financial\_loan;



Among these, fully paid and current are considered as good loans while charged off is considered as bad loan.

* Good Loan Percentage:

SELECT ROUND(COUNT(CASE

WHEN loan\_status = 'Fully Paid' OR loan\_status = 'Current' THEN id END)\*100.0/COUNT(id), 2)

AS good\_loan\_perc

FROM financial\_loan;



* Good Loan Applications:

SELECT COUNT(id)

FROM financial\_loan

WHERE loan\_status IN ('Fully Paid' ,'Current');



* Good Loan Funded Amount:

SELECT SUM(loan\_amount) AS good\_loan\_funded\_amt

FROM financial\_loan

WHERE loan\_status IN ('Fully Paid' ,'Current');



* Good Loan Amount Received:

SELECT SUM(total\_payment) AS good\_loan\_rec\_amt

FROM financial\_loan

WHERE loan\_status IN ('Fully Paid' ,'Current');



* Bad Loan Percentage:

SELECT ROUND(COUNT(CASE

WHEN loan\_status = 'Charged Off' THEN id END)\*100.0/COUNT(id), 2)

AS bad\_loan\_perc

FROM financial\_loan;



* Bad Loan Applications:

SELECT COUNT(id) AS bad\_loan\_tot\_apps

FROM financial\_loan

WHERE loan\_status = 'Charged Off';



* Bad Loan Funded Amount:

SELECT SUM(loan\_amount) AS bad\_loan\_funded\_amt

FROM financial\_loan

WHERE loan\_status = 'Charged Off';



* Bad Loan Amount Received:

SELECT SUM(total\_payment) AS bad\_loan\_rec\_amt

FROM financial\_loan

WHERE loan\_status = 'Charged Off';



* Metrics Grouped by Loan Status:

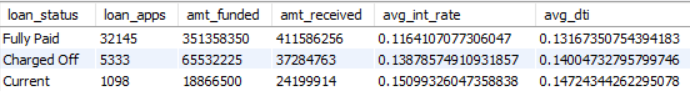
SELECT loan\_status, COUNT(id) as loan\_apps, SUM(loan\_amount) as amt\_funded,

SUM(total\_payment) as amt\_received, AVG(int\_rate) as avg\_int\_rate,

AVG(dti) as avg\_dti

FROM financial\_loan

GROUP BY loan\_status;



* Report Overview w.r.t Month:

SELECT DATE\_FORMAT(issue\_date, '%M') as month, MONTH(issue\_date) as month\_no,

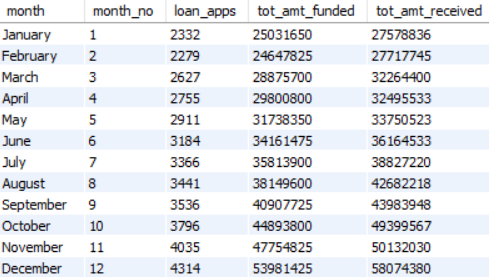
COUNT(id) as loan\_apps, SUM(loan\_amount) as tot\_amt\_funded,

SUM(total\_payment) as tot\_amt\_received

FROM financial\_loan

GROUP BY DATE\_FORMAT(issue\_date, '%M'),MONTH(issue\_date)

ORDER BY MONTH(issue\_date);



* Report Overview w.r.t Term:

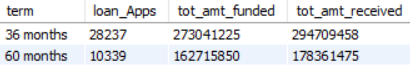
SELECT term, COUNT(id) AS loan\_Apps, SUM(loan\_amount) AS tot\_amt\_funded,

SUM(total\_payment) AS tot\_amt\_received

FROM financial\_loan

GROUP BY term

ORDER BY COUNT(id) DESC;



* Report Overview w.r.t States(top 10 states with highest total amount received):

SELECT address\_state AS State, COUNT(id) AS Total\_Loan\_Applications,

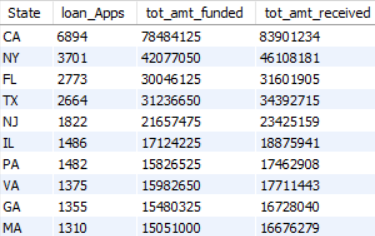
SUM(loan\_amount) AS tot\_amt\_funded, SUM(total\_payment) AS tot\_amt\_received

FROM financial\_loan

GROUP BY address\_state

ORDER BY COUNT(id) DESC

LIMIT 10;



* Report Overview w.r.t Employee Length:

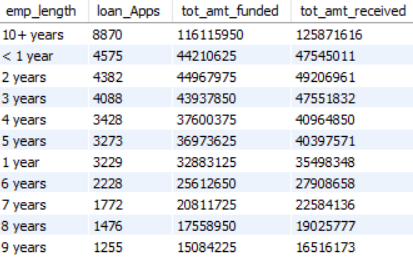
SELECT emp\_length , COUNT(id) AS loan\_Apps, SUM(loan\_amount) AS tot\_amt\_funded,

SUM(total\_payment) AS tot\_amt\_received

FROM financial\_loan

GROUP BY emp\_length

ORDER BY COUNT(id) DESC;



* Report Overview w.r.t Purpose of taking Loan:

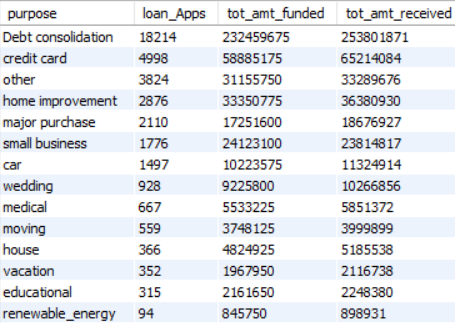
SELECT purpose , COUNT(id) AS loan\_Apps, SUM(loan\_amount) AS tot\_amt\_funded,

SUM(total\_payment) AS tot\_amt\_received

FROM financial\_loan

GROUP BY purpose

ORDER BY COUNT(id) DESC;



* Report Overview w.r.t Home Ownership:

SELECT home\_ownership , COUNT(id) AS loan\_Apps, SUM(loan\_amount) AS tot\_amt\_funded,

SUM(total\_payment) AS tot\_amt\_received

FROM financial\_loan

GROUP BY home\_ownership

ORDER BY COUNT(id) DESC;

