Bank Loan Analysis

[Presented by: Yerram Sai Kiran]

Query1

Find the percentage change in the total loan amount issued from the previous month to the current month

Purpose

→ This analysis helps to monitor the trend in loan issuance by comparing the total loan amounts issued in the current month with the previous month and to identify any increases or decreases in the loan activity.

Insights

→ Witnessed an increase of 13.04% in issued loans for latest month compared to previous month

[By: Yerram Sai Kiran]

Query

Query1.sql

```sql
With pre_month_loan_amt as
(
Select Sum(loan_amount) as pre_mth_loan_amt
From bank_loan_data
Where
  Year(issue_date) = (Select Year(Date_Sub(max(issue_date), Interval 1 Month)) from bank_loan_data)
    And month(issue_date) = (Select Month(Date_sub(max(issue_date), Interval 1 Month)) from bank_loan_data)
),

cur_month_loan_amt as
(
Select Sum(loan_amount) as cur_mth_loan_amt
From bank_loan_data
Where
  Year(issue_date) = (Select Year(max(issue_date)) from bank_loan_data)
    And month(issue_date) = (Select Month(max(issue_date)) from bank_loan_data)
)

Select
  cur_mth_loan_amt,
  pre_mth_loan_amt,
  Round((cur_mth_loan_amt - pre_mth_loan_amt) * 100/pre_mth_loan_amt, 2) as "chg %"
From (pre_month_loan_amt, cur_month_loan_amt);
```

[By: Yerram Sai Kiran]

Result

	cur_mth_loan_amt	pre_mth_loan_amt	chg %
▶	53981425	47754825	13.04

[By: Yerram Sai Kiran]

Query2

Create a Loan Status grid view report categorized by 'Loan Status'.

→ This analysis provides a comprehensive overview of the lending operations and offers insights about fundings, repayments, no of good loans (if loan status = Fully Paid or Current) and bad loans and thus allowing for monitoring of the performance of lending's

Purpose

Insights

→ 83.33 % of approved loans have completely paid the amount while a significant portion of 13.82% of loans were charged-off and the rest 2.85% of the loans are currently active.

[By: Yerram Sai Kiran]

Query

Query2.sql

```sql
with additional_data as
(
  Select
    loan_status,
    Concat(Round(Sum(loan_amount)/1000000,2), "M") as mtd_funded_amount,
        Concat(Round(Sum(total_payment)/1000000,2), "M") as mtd_received_amount
  From bank_loan_data
    Where
    Year(issue_date) = (Select Year(max(issue_date)) From bank_loan_data)
        And Month(issue_date) = (Select Month(max(issue_date)) From bank_loan_data)
  Group by loan_status
)
Select
  bk.loan_status,
  Count(id) as total_loan_applications,
    Round(Count(id)*100 / (select count(id) from bank_loan_data), 2) as "%_of_total_applications",
    Concat(Round(Sum(loan_amount)/1000000,2), "M") as total_funded_amount,
    Concat(Round(Sum(total_payment)/1000000,2), "M") as total_received_amount,
    mtd_funded_amount,
    mtd_received_amount,
    Round(Avg(int_rate)*100, 2) as avg_interest_rate,
    Round(Avg(dti)*100, 2) as avg_dti
From bank_loan_data bk
Join additional_data ad
  On bk.loan_status = ad.loan_status
Group by loan_status;
```

[By: Yerram Sai Kiran]

Result

loan_status	total_loan_applications	%_of_total_applications	total_funded_amount	total_received_amount	mtd_funded_amount	mtd_received_amount	avg_interest_rate	avg_dti
Fully Paid	32145	83.33	351.36M	411.59M	41.30M	47.82M	11.64	13.17
Charged Off	5333	13.82	65.53M	37.28M	8.73M	5.32M	13.88	14
Current	1098	2.85	18.87M	24.20M	3.95M	4.93M	15.1	14.72

[By: Yerram Sai Kiran]

Query3

Find the % of good loan and bad loan applications

Purpose

→ This calculates the overall loan performance by determining the percentage of successful (fully paid or current) loans vs charged-off loans, providing insights into portfolio health and lending behavior.

Insights

→ 86.18% of loans are found to be good-loans and the remaining 13.82% of the loans are bad-loans

[By: Yerram Sai Kiran]

Query

Query3.sql

```sql
Select
    Round(Count(Case When loan_status = "Fully Paid" Or loan_status = "Current" Then id End) * 100 / Count(id), 2) as good_loans_pct,
    Round(Count(Case When loan_status = "Charged Off" Then id End) * 100 / Count(id), 2) as bad_loans_pct
From bank_loan_data;
```

[By: Yerram Sai Kiran]

Result

good_loans_pct	bad_loans_pct
86.18	13.82

[By: Yerram Sai Kiran]

Query4

Analyze the loan applications by Loan Purpose Breakdown

Purpose

→ Identifying the high-demand loan purposes can drive us to make optimized lending strategies (like making changes in the interest rates) to attract more customers

Insights

→ Among all the loan purposes, Debt consolidation is identified to be the top purpose with max no of loan applications (18214) and funding amount (232.46M)

[By: Yerram Sai Kiran]

Query

Query4.sql

```sql
Select
    purpose as loan_purpose,
    Count(id) as total_loan_applications,
    Concat(Round(Sum(loan_amount)/1000000,2), "M") as total_funded_amount
From bank_loan_data
Group by purpose
Order by total_loan_applications Desc;
```

[By: Yerram Sai Kiran]

Result

loan_purpose	total_loan_applications	total_funded_amount
Debt consolidation	18214	232.46M
credit card	4998	58.89M
other	3824	31.16M
home improvement	2876	33.35M
major purchase	2110	17.25M
small business	1776	24.12M
car	1497	10.22M
wedding	928	9.23M
medical	667	5.53M
moving	559	3.75M
house	366	4.82M
vacation	352	1.97M
educational	315	2.16M
renewable_energy	94	0.85M

[By: Yerram Sai Kiran]

Query5

Create a stored procedure to spot top 5 states with the highest profit from fully paid loans for a given month and year.

Purpose

→ This identifies states contributing most to overall profitability and enables us to make a better risk assessment and resource allocation strategies.

Insights

→ This stored procedure can be used to find the top 5 states with max profits based on the provided input values of year and the month.

[By: Yerram Sai Kiran]

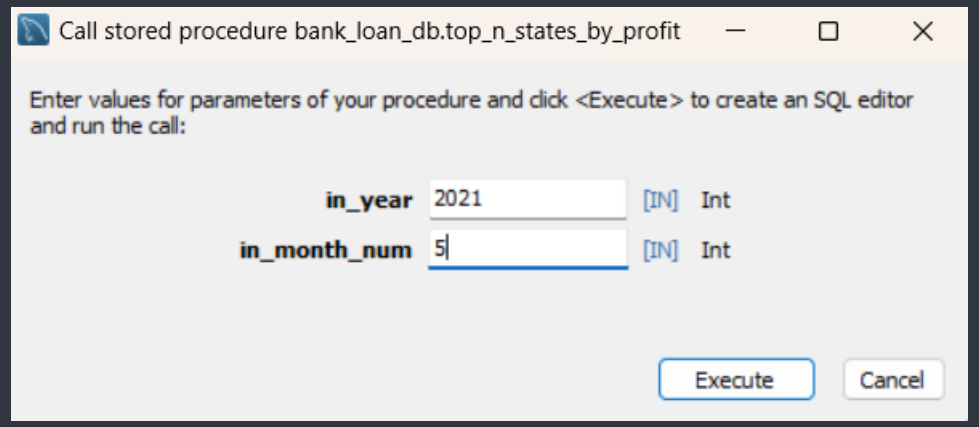
Query

Query5.sql

```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `top_n_states_by_profit`(
In in_year Int,
In in_month_num Int
)
BEGIN
Select
    address_state,
    Count(Case When loan_status = "Fully Paid" Then id End) as loans_count,
    Sum(Case When loan_status = "Fully Paid" Then (total_payment - loan_amount) End) as profit
From bank_loan_data
Where
    year(issue_date) = in_year
    and month(issue_date) = in_month_num
Group by address_state
Order by profit desc
Limit 5;
END
```

[By: Yerram Sai Kiran]

Result



address_state	loans_count	profit
CA	502	879329
NY	197	370083
TX	186	281627
FL	156	235556
VA	103	199437

[By: Yerram Sai Kiran]

Query6

Identify the top sub-grade within each loan grade that have the highest number of charged-off loans

→ This information can be useful for understanding the riskiest segment within each grade and potentially adjusting lending criteria.

Purpose

Insights

→ 'G2' is the most risk associated category with 35.9% charged off loans at subgrade level.

[By: Yerram Sai Kiran]

Query

Query6.sql

```sql
WITH grades_data AS (
    SELECT
        grade,
        sub_grade,
        COUNT(*) AS total_loans,
        Count(Case When loan_status = "Charged Off" Then id End) as charged_off_loans,
        Round(Count(Case When loan_status = "Charged Off" Then id End)*100/Count(id),2) as charged_off_loans_pct
    FROM bank_loan_data
    GROUP BY grade, sub_grade
),
ranked_sub_grades AS (
    SELECT
        *,
        dense_rank() OVER (PARTITION BY grade ORDER BY charged_off_loans DESC) AS sub_grade_ranking
    FROM grades_data
)
SELECT
    grade,
    sub_grade,
    total_loans,
    charged_off_loans,
    charged_off_loans_pct
FROM ranked_sub_grades
WHERE sub_grade_ranking = 1
ORDER BY grade desc, charged_off_loans_pct desc;
```

[By: Yerram Sai Kiran]

Result

grade	sub_grade	total_loans	charged_off_loans	charged_off_loans_pct
G	G2	78	28	35.90
G	G1	101	28	27.72
F	F1	325	89	27.38
E	E1	750	193	25.73
D	D2	1314	258	19.63
C	C1	2089	317	15.17
B	B5	2644	344	13.01
A	A5	2654	204	7.69

[By: Yerram Sai Kiran]

[By: Yerram Sai Kiran]