About Dataset

The tables **finance_1** and **finance_2** are part of a financial database. **finance_1** includes details about loans, such as loan amount, term, interest rate, and borrower information. The **id** column serves as the primary key, linking to corresponding records in **finance_2**, which contains credit-related data like delinquency history, credit lines, and payment information. **finance_2** is linked to **finance_1** through the common **id** column. Both tables collectively offer a comprehensive view of members' financial profiles, encompassing loan status, credit history, and transaction details. Common queries involve aggregations, comparisons, and calculations to derive insights into members' financial behavior, loan performance, and overall credit health. The dataset enables analysis and decision-making processes within the context of financial operations and risk management.

About Dataset,

```
select * from finance_1;

select * from Finance_2;

Question 1 : Year Wise Loan Amount

select sum(loan_amnt) as 'LoanAmount', YEAR(issue_d)  as 'Year' from Finance_1
group by YEAR(issue_d) order by YEAR(issue_d);



Question 2 : Grade-Subgrade wise revolution balance

select grade, sub_grade, sum(revol_bal) as total_revol_bal from Finance_1 inner join
Finance_2 on
(Finance_1.id = Finance_2.id) group by grade,sub_grade order by grade;



Question 3 : Total payment for verified and non verified status

select verification_status as Veri_Status, round(SUM(total_pymnt)/1000000,2) as
Total_payment from Finance_1 inner join Finance_2 on
(Finance_1.id = Finance_2.id) group by verification_status;
```

Question 4 : State wise and last_credit_pull_d wise loan status

```
select loan_status,addr_state, last_credit_pull_d from Finance_1 inner join  Finance_2
on(Finance_1.id = Finance_2.id) group by loan_status,addr_state, last_credit_pull_d
order by loan_status
```

Question 5 : Home Ownership Vs last payment data status

```
select home_ownership, year(last_pymnt_d) as last_payment_year
,round(SUM(total_pymnt)/1000000,2) as total_payment_amount from Finance_1 inner join
Finance_2
on(Finance_1.id = Finance_2.id) group by home_ownership, last_pymnt_d order by
total_payment_amount desc
```

Question 6 : Retrieve the total loan amount funded (funded_amnt) for each grade

```
select grade,round(sum(funded_amnt),2) as Funded_amount  from Finance_1 group by grade
order by Funded_amount desc
```

Question 7 : Find the top 5 states (addr_state) with the highest average annual income

```
select top 5 addr_state,round(avg(annual_inc),2) as Annual_income
from Finance_1
group by addr_state
order by Annual_income desc;
```

Question 8 : List the loan status (loan_status) along with the count of loans for each status from

```
select top 10 loan_status,count(loan_status) as loan_status_count,addr_state
from Finance_1
where loan_status = 'Fully Paid'
group by loan_status,addr_state
order by loan_status_count desc

select top 10 loan_status,count(loan_status) as loan_status_count,addr_state
from Finance_1
where loan_status = 'Charged Off'
group by loan_status,addr_state
order by loan_status_count desc
```

Question 9 : Calculate the average interest rate (int_rate) for each loan term (term) in the

```
SELECT term,
       AVG(CAST(REPLACE(REPLACE(int_rate, '%', ''), ',', '') AS DECIMAL(4,2))) AS
average_interest_rate
FROM finance_1
GROUP BY term;
```

Question 10 : Identify the members top 10 (member_id) with the highest total payments (total_pymnt)

```
select top 10 member_id As Member_ID, SUM(total_pymnt) as Total_Payment_amount
from Finance_1
inner join Finance_2
on (Finance_1.id = Finance_2.id)
group by Member_ID
order by Total_Payment_amount desc;
```

Question 11 : Determine the average revolve balance (revol_bal) for each home ownership (home_ownership) category

```
select home_ownership as Home_Ownership, AVG(revol_bal) as average_revolve_balance
from Finance_1
inner join Finance_2
on (Finance_1.id = Finance_2.id)
group by home_ownership
order by average_revolve_balance desc;
```

Question 12 : Find the loan amount (loan_amnt) with the highest late fee (total_rec_late_fee)

```
select loan_amnt as Loan_Amount, total_rec_late_fee as Highest_late_fee from Finance_1
inner join Finance_2
on (Finance_1.id = Finance_2.id)
group by loan_amnt,total_rec_late_fee
order by total_rec_late_fee desc
```

Question 13 : Calculate the percentage of total payments (total_pymnt) recovered (recoveries) for each loan status (loan_status)

```
select loan_status,total_pymnt,recoveries,concat(((recoveries/total_pymnt)*100),'%')
as percentage_of_total_amount from Finance_1
inner join Finance_2
on (Finance_1.id = Finance_2.id) where recoveries !=0
group by loan_status,total_pymnt,recoveries;
```

Question 14 : Identify the members (member_id) who have never been delinquent (delinq_2yrs = 0) and have the highest annual income (annual_inc)

```
select top 5 member_id, annual_inc, delinq_2yrs from Finance_1
inner join Finance_2
on (Finance_1.id = Finance_2.id)
where delinq_2yrs = 0
group by member_id, annual_inc, delinq_2yrs
order by annual_inc desc;
```

Question 15 : Retrieve the top 5 loan purposes (purpose) with the highest average annual income (annual_inc) from the finance_1

```
select top 5 purpose,concat(round(AVG(annual_inc/1000),2),'K') as avg_annual_income
from Finance_1
group by purpose
order by avg_annual_income desc;
```

Question 16 : Identify the top 3 states (addr_state) with the highest average total payments (total_pymnt) for loans

```
select top 3 addr_state, (round(AVG(total_pymnt),3)) as avg_total_payment from
Finance_1
inner join Finance_2
on (Finance_1.id = Finance_2.id)
group by addr_state
order by avg_total_payment desc
```

Question 17 : Calculate the average debt-to-income ratio (dti) for each employment length (emp_length) category

```
select emp_length, (round(AVG(dti),3)) as avg_dti from Finance_1
group by emp_length,dti
order by avg_dti desc
```

Question 18 : List the members (member_id) who have the highest total recovery amount (recoveries) for each loan status (loan_status)

```
select top 10 member_id,loan_status,sum(recoveries) as total_recovery_amt from
Finance_1
inner join Finance_2
on (Finance_1.id = Finance_2.id)
group by member_id,loan_status,recoveries
order by total_recovery_amt desc
```

Question 19 : Calculate the average revolving utilization (revol_util) for each grade (grade)

```
select grade,
avg(cast(replace(replace(revol_util,'%',''),',','') as decimal(4,2))) as
avg_revol_util from Finance_1
inner join Finance_2
on (Finance_1.id = Finance_2.id)
group by grade
order by avg_revol_util desc
```

Question 20 : Identify the top 5 members (member_id) with the highest late fees (total_rec_late_fee) as a percentage of their total payments (total_pymnt)

```
select top 5
member_id,total_rec_late_fee,total_pymnt,((total_rec_late_fee/total_pymnt)*100) as
late_fee_percen from Finance_1
inner join Finance_2
on (Finance_1.id = Finance_2.id) where total_rec_late_fee != 0
order by late_fee_percen desc
```

Question 21 : Calculate the average interest rate (int_rate) for loans where the annual
income (annual_inc) is above the overall average annual income

```
select AVG(CAST(REPLACE(REPLACE(int_rate, '%', ''), ',', '') AS DECIMAL(4,2))) AS
average_interest_rate, annual_inc
from Finance_1 where annual_inc > (select avg(annual_inc) from Finance_1)
group by int_rate,annual_inc
order by int_rate desc
```

Thank-You