**# KPI 1**

select year (issue\_D) as year\_of\_issue\_d , sum(loan\_amnt) as Total\_Loan\_amt

from finance\_1

group by year\_of\_issue\_d

order by year\_of\_issue\_d;

Explanation :

1. select year (issue\_D) as year\_of\_issue\_d , sum(loan\_amnt) as Total\_Loan\_amt

This part of the query selects two columns:

* year(issue\_D) AS year\_of\_issue\_d: It extracts the year from the issue\_D column (presumably the date of loan issuance) and assigns it an alias year\_of\_issue\_d.
* sum(loan\_amnt) AS Total\_Loan\_amt: It calculates the sum of the loan\_amnt column (presumably representing the loan amount) for each group of years and assigns it an alias Total\_Loan\_amt.

1. from finance\_1

This specifies the table from which the data is being selected, in this case, finance\_1.

1. Group by year\_of\_issue\_d

This clause groups the rows of the table by the year\_of\_issue\_d column, so that the subsequent aggregate function (SUM) operates on each group separately.

1. Order by year\_of\_issue\_d

This clause orders the result set by the year\_of\_issue\_d column in ascending order.

**# KPI 2**

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, sub\_grade;

Explanation :

1. select grade, sub\_grade, sum(revol\_bal) as Total\_revol\_bal:

• This part of the query selects three columns:

• grade: It selects the grade column from the finance\_1 table.

• sub\_grade: It selects the sub\_grade column from the finance\_1 table.

• sum(revol\_bal) as Total\_revol\_bal: It calculates the sum of the revol\_bal column (presumably representing revolving balance) for each group of grade and sub\_grade and assigns it an alias Total\_revol\_bal.

2. from finance\_1:

• This specifies the primary table from which the data is being selected, in this case, finance\_1.

3. inner join finance\_2 on (finance\_1.id = finance\_2.id):

• This clause performs an inner join between finance\_1 and finance\_2 tables based on the id column. It links rows from finance\_1 with rows from finance\_2 where the id values match.

4. group by grade, sub\_grade:

• This clause groups the rows of the combined result set by the grade and sub\_grade columns, so that the subsequent aggregate function (SUM) operates on each group separately.

5. order by grade, sub\_grade:

• This clause orders the result set by the grade and sub\_grade columns in ascending order.

**# KPI 3**

select verification\_status ,

concat("$", format(round(sum(total\_pymnt)/1000000,2),2),"M") as total\_payment

from finance\_1 inner join finance\_2

on (finance\_1.id = finance\_2.id)

group by verification\_status ;

Explanation :

1. select verification\_status:

This part of the query selects the verification\_status column from the combined dataset.

1. concat("$", format(round(sum(total\_pymnt)/1000000,2),2),"M") as total\_payment:
   * This part of the query calculates the total payment in millions, formats it as currency, and assigns it an alias total\_payment.
   * sum(total\_pymnt): It calculates the total payment by summing up the total\_pymnt column. round(sum(total\_pymnt)/1000000,2): It rounds the total payment to 2 decimal places and converts it to millions.
   * format(...,"$"): It formats the rounded total payment as currency with the dollar sign.
   * "M": It appends the "M" symbol to indicate millions.
2. from finance\_1 inner join finance\_2 on (finance\_1.id = finance\_2.id):

This clause performs an inner join between finance\_1 and finance\_2 tables based on their id columns.

1. group by verification\_status:

This clause groups the combined data by verification\_status.

**# KPI 4**

select addr\_state , last\_Credit\_pull\_D , loan\_status

from finance\_1 inner join finance\_2

on (finance\_1.id = finance\_2.id)

group by addr\_state , last\_Credit\_pull\_D , loan\_status

order by last\_Credit\_pull\_D ;

Explanation :

1. select addr\_state, last\_credit\_pull\_d, loan\_status:

This part of the query selects three columns: addr\_state, last\_credit\_pull\_d, and loan\_status from the combined dataset.

1. from finance\_1 inner join finance\_2 on (finance\_1.id = finance\_2.id):

This clause performs an inner join between the finance\_1 and finance\_2 tables based on their id columns.

1. group by addr\_state, last\_credit\_pull\_d, loan\_status:

This clause groups the combined data by the columns addr\_state, last\_credit\_pull\_d, and loan\_status.

1. order by last\_credit\_pull\_d:

This clause orders the grouped result set by the last\_credit\_pull\_d column.

**# KPI 5**

home\_ownership,

last\_pymnt\_d,

concat("$", format(round(sum(last\_pymnt\_amnt) / 10000, 2), 2), "K") as total\_amount

from finance\_1

inner join finance\_2 on finance\_1.id = finance\_2.id

group by home\_ownership, last\_pymnt\_d

order by last\_pymnt\_d desc, home\_ownership desc;

Explanation :

1. select home\_ownership, last\_pymnt\_d, concat("$", format(round(sum(last\_pymnt\_amnt) / 10000, 2), 2), "K") as total\_amount:

This part of the query selects three columns: home\_ownership, last\_pymnt\_d, and a formatted total amount concatenated with "$" and "K".

1. from finance\_1 inner join finance\_2 on finance\_1.id = finance\_2.id:

This clause performs an inner join between the finance\_1 and finance\_2 tables based on their id columns.

1. group by home\_ownership, last\_pymnt\_d:

This clause groups the combined data by the columns home\_ownership and last\_pymnt\_d.

1. order by last\_pymnt\_d desc, home\_ownership desc:

This clause orders the grouped result set by last\_pymnt\_d in descending order, then by home\_ownership in descending order.