**Tasks with MySQL**

* Customer Risk Analysis: Identify customers with low credit scores and high-risk loans to predict potential defaults and prioritize risk mitigation strategies.

SELECT

customer\_id,

name,

credit\_score,

risk\_category,

active\_loans

FROM

customer\_table

WHERE

credit\_score < 600

AND risk\_category = 'High';

* Loan Purpose Insights: Determine the most popular loan purposes and their associated revenues to align financial products with customer demands

SELECT

loan\_purpose,

COUNT(loan\_id) AS number\_of\_loans,

SUM(loan\_amount) AS total\_revenue

FROM

loan\_table

GROUP BY

loan\_purpose

ORDER BY

total\_revenue DESC;

* High-Value Transactions: Detect transactions that exceed 30% of their respective loan amounts to flag potential fraudulent activities

SELECT

loan\_purpose,

COUNT(loan\_id) AS number\_of\_loans,

SUM(loan\_amount) AS total\_revenue

FROM

loan\_table

GROUP BY

loan\_purpose

ORDER BY

total\_revenue DESC;

* Missed EMI Count: Analyze the number of missed EMIs per loan to identify loans at risk of default and suggest intervention strategies

SELECT

t.transaction\_id,

t.loan\_id,

t.customer\_id,

t.transaction\_date,

t.transaction\_amount,

l.loan\_amount,

CASE

WHEN t.transaction\_amount > 0.3 \* l.loan\_amount THEN 'High-Value Transaction'

ELSE 'Normal Transaction'

END AS transaction\_flag

FROM

transaction\_table t

JOIN

loan\_table l ON t.loan\_id = l.loan\_id

WHERE

t.transaction\_amount > 0.3 \* l.loan\_amount;

* Regional Loan Distribution: Examine the geographical distribution of loan disbursements to assess regional trends and business opportunities.

SELECT

t.transaction\_id,

t.loan\_id,

t.customer\_id,

c.name AS customer\_name,

t.transaction\_date,

t.transaction\_amount,

l.loan\_amount,

CASE

WHEN t.transaction\_amount > 0.3 \* l.loan\_amount THEN 'High-Value Transaction'

ELSE 'Normal Transaction'

END AS transaction\_flag

FROM

transaction\_table t

JOIN

loan\_table l ON t.loan\_id = l.loan\_id

JOIN

customer\_table c ON l.customer\_id = c.customer\_id

WHERE

t.transaction\_amount > 0.3 \* l.loan\_amount;

* Loyal Customers: List customers who have been associated with Cross River Bank for over five years and evaluate their loan activity to design loyalty programs.

SELECT

t.transaction\_id,

t.loan\_id,

c.customer\_id,

c.name,

t.transaction\_date,

t.transaction\_amount,

l.loan\_amount,

CASE

WHEN t.transaction\_amount > 0.3 \* l.loan\_amount THEN 'High-Value Transaction'

ELSE 'Normal Transaction'

END AS transaction\_flag

FROM

transaction\_table t

JOIN

loan\_table l ON t.loan\_id = l.loan\_id

JOIN

customer\_table c ON t.customer\_id = c.customer\_id

WHERE

t.transaction\_amount > 0.3 \* l.loan\_amount;

* High-Performing Loans: Identify loans with excellent repayment histories to refine lending policies and highlight successful products.

SELECT

t.transaction\_id,

t.loan\_id,

t.customer\_id,

t.transaction\_date,

t.transaction\_amount,

l.loan\_amount,

CASE

WHEN t.transaction\_amount > 0.3 \* l.loan\_amount THEN 'High-Value Transaction'

ELSE 'Normal Transaction'

END AS transaction\_flag

FROM

transaction\_table t

JOIN

loan\_table l ON t.loan\_id = l.loan\_id

WHERE

t.transaction\_amount > 0.3 \* l.loan\_amount;

* Age-Based Loan Analysis: Analyze loan amounts disbursed to customers of different age groups to design targeted financial products.

SELECT

CASE

WHEN c.age < 25 THEN 'Under 25'

WHEN c.age BETWEEN 25 AND 34 THEN '25-34'

WHEN c.age BETWEEN 35 AND 44 THEN '35-44'

WHEN c.age BETWEEN 45 AND 54 THEN '45-54'

WHEN c.age BETWEEN 55 AND 64 THEN '55-64'

ELSE '65 and above'

END AS age\_group,

SUM(l.loan\_amount) AS total\_loan\_amount,

COUNT(l.loan\_id) AS number\_of\_loans

FROM

customer\_table c

JOIN

loan\_table l ON c.customer\_id = l.customer\_id

GROUP BY

age\_group

ORDER BY

age\_group;

* Seasonal Transaction Trends: Examine transaction patterns over years and months to identify seasonal trends in loan repayments.

SELECT

YEAR(t.transaction\_date) AS transaction\_year,

MONTH(t.transaction\_date) AS transaction\_month,

COUNT(t.transaction\_id) AS total\_transactions,

SUM(t.transaction\_amount) AS total\_repayments

FROM

transaction\_table t

JOIN

loan\_table l ON t.loan\_id = l.loan\_id

GROUP BY

transaction\_year,

transaction\_month

ORDER BY

transaction\_year,

transaction\_month;

* Fraud Detection: Highlight potential fraud by identifying mismatches between customer address locations and transaction IP locations.

SELECT

t.transaction\_id,

t.loan\_id,

t.customer\_id,

t.transaction\_date,

t.transaction\_amount,

l.loan\_amount,

CASE

WHEN t.transaction\_amount > 0.3 \* l.loan\_amount THEN 'High-Value Transaction'

ELSE 'Normal Transaction'

END AS transaction\_flag

FROM

transaction\_table t

JOIN

loan\_table l ON t.loan\_id = l.loan\_id

WHERE

t.transaction\_amount > 0.3 \* l.loan\_amount;

* Repayment History Analysis: Rank loans by repayment performance using window functions.

SELECT

loan\_id,

customer\_id,

loan\_amount,

loan\_date,

loan\_status,

repayment\_history,

interest\_rate,

loan\_purpose,

collateral,

default\_risk,

RANK() OVER (ORDER BY repayment\_history DESC) AS repayment\_rank

FROM

loan\_table;

* Credit Score vs. Loan Amount: Compare average loan amounts for different credit score ranges.

SELECT

CASE

WHEN c.credit\_score < 600 THEN 'Below 600'

WHEN c.credit\_score BETWEEN 600 AND 699 THEN '600-699'

WHEN c.credit\_score BETWEEN 700 AND 799 THEN '700-799'

WHEN c.credit\_score >= 800 THEN '800 and above'

END AS credit\_score\_range,

AVG(l.loan\_amount) AS average\_loan\_amount

FROM

loan\_table l

JOIN

customer\_table c ON l.customer\_id = c.customer\_id

GROUP BY

credit\_score\_range

ORDER BY

credit\_score\_range;

* Top Borrowing Regions: Identify regions with the highest total loan disbursements.

SELECT

t.transaction\_id,

t.loan\_id,

t.customer\_id,

t.transaction\_date,

t.transaction\_amount,

l.loan\_amount,

CASE

WHEN t.transaction\_amount > 0.3 \* l.loan\_amount THEN 'High-Value Transaction'

ELSE 'Normal Transaction'

END AS transaction\_flag

FROM

transaction\_table t

JOIN

loan\_table l ON t.loan\_id = l.loan\_id

WHERE

t.transaction\_amount > 0.3 \* l.loan\_amount;

* Early Repayment Patterns: Detect loans with frequent early repayments and their impact on revenue.

SELECT

t.transaction\_id,

t.loan\_id,

t.customer\_id,

t.transaction\_date,

t.transaction\_amount,

l.loan\_amount, -- Assuming loan\_amount is a column in the loan\_table

CASE

WHEN t.transaction\_amount > 0.3 \* l.loan\_amount THEN 'High-Value Transaction'

ELSE 'Normal Transaction'

END AS transaction\_flag

FROM

transaction\_table t

JOIN

loan\_table l ON t.loan\_id = l.loan\_id

WHERE

t.transaction\_amount > 0.3 \* l.loan\_amount;

* Feedback Correlation: Correlate customer feedback sentiment scores with loan statuses.

SELECT

t.transaction\_id,

t.loan\_id,

t.customer\_id,

t.transaction\_date,

t.transaction\_amount,

l.loan\_amount,

CASE

WHEN t.transaction\_amount > 0.3 \* l.loan\_amount THEN 'High-Value Transaction'

ELSE 'Normal Transaction'

END AS transaction\_flag

FROM

transaction\_table t

JOIN

loan\_table l ON t.loan\_id = l.loan\_id

WHERE

t.transaction\_amount > 0.3 \* l.loan\_amount;