|  |  |
| --- | --- |
| **Name** | **Vivek Dash [ABADS BATCH 16]** |
| **Email** | [**vivekdash08@gmail.com**](mailto:vivekdash08@gmail.com) |
| **Phone** | **9776217627** |
|  | |
| **Task 1** | **Customer Risk Analysis:** Identify customers with low credit scores and high-risk loans to predict potential defaults and prioritize risk mitigation strategies. |
| **Solution 1** | SELECT  c.customer\_id, c.name, c.credit\_score, l.loan\_id, l.default\_risk  FROM cross\_river\_bank.customer\_table as c  JOIN cross\_river\_bank.loan\_table as l  ON c.customer\_id = l.customer\_id  WHERE c.credit\_score < 600 & l.default\_risk = 'High'  GROUP BY c.customer\_id, c.name, c.credit\_score, l.loan\_id, l.loan\_amount, l.default\_risk  ORDER BY c.credit\_score ASC; |
| **Task 2** | **Loan Purpose Insights:** Determine the most popular loan purposes and their associated revenues to align financial products with customer demands |
| **Solution 2** | SELECT  l.loan\_purpose,  COUNT(l.loan\_id) AS total\_loans,  SUM(l.loan\_amount) AS total\_revenue  FROM  cross\_river\_bank.loan\_table AS l  WHERE  l.loan\_status IN ('Approved', 'Closed')  GROUP BY  l.loan\_purpose  ORDER BY  total\_loans DESC,  total\_revenue DESC; |
| **Task 3** | **High-Value Transactions:** Detect transactions that exceed 30% of their respective loan amounts to flag potential fraudulent activities |
| **Solution 3** | SELECT  t.transaction\_id,  t.customer\_id,  t.loan\_id,  t.transaction\_date,  t.transaction\_amount,  l.loan\_amount,  (t.transaction\_amount / l.loan\_amount) \* 100 AS trans\_percentage  FROM  cross\_river\_bank.transaction\_table AS t  JOIN  cross\_river\_bank.loan\_table AS l ON t.loan\_id = l.loan\_id  WHERE  (t.transaction\_amount / l.loan\_amount) > 0.30  ORDER BY  trans\_percentage DESC; |
| **Task 4** | **Missed EMI Count:** Analyze the number of missed EMIs per loan to identify loans at risk of default and suggest intervention strategies |
| **Solution 4** | SELECT  l.loan\_id,  l.customer\_id,  COUNT(t.transaction\_id) AS total\_missed\_emis,  l.loan\_amount,  l.loan\_status,  c.name AS customer\_name,  c.credit\_score  FROM  cross\_river\_bank.transaction\_table AS t  JOIN  cross\_river\_bank.loan\_table AS l ON t.loan\_id = l.loan\_id  JOIN  cross\_river\_bank.customer\_table AS c ON l.customer\_id = c.customer\_id  WHERE  t.transaction\_type = 'EMI Payment'  AND t.status = 'Failed'  GROUP BY  l.loan\_id, l.customer\_id, l.loan\_amount, l.loan\_status, c.name, c.credit\_score  HAVING  total\_missed\_emis > 0  ORDER BY  total\_missed\_emis DESC; |
| **Task 5** | **Regional Loan Distribution:** Examine the geographical distribution of loan disbursements to assess regional trends and business opportunities. |
| **Solution 5** | SELECT  c.address,  COUNT(l.loan\_id) AS total\_loans\_disbursed,  SUM(l.loan\_amount) AS total\_loan\_amount,  AVG(l.loan\_amount) AS average\_loan\_amount  FROM  cross\_river\_bank.loan\_table AS l  JOIN  cross\_river\_bank.customer\_table AS c ON l.customer\_id = c.customer\_id  WHERE  l.loan\_status IN ('Approved', 'Closed')  GROUP BY  c.address  ORDER BY  total\_loan\_amount DESC; |
| **Task 6** | **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. |
| **Solution 6** | SELECT  c.customer\_id,  c.name,  c.customer\_since,  TIMESTAMPDIFF(YEAR, STR\_TO\_DATE(c.customer\_since, '%m/%d/%Y'), CURDATE()) AS years\_with\_bank,  COUNT(l.loan\_id) AS total\_loans,  SUM(l.loan\_amount) AS total\_loan\_amount,  AVG(l.loan\_amount) AS average\_loan\_amount  FROM  cross\_river\_bank.customer\_table AS c  LEFT JOIN  cross\_river\_bank.loan\_table AS l ON c.customer\_id = l.customer\_id  WHERE  TIMESTAMPDIFF(YEAR, STR\_TO\_DATE(c.customer\_since, '%m/%d/%Y'), CURDATE()) > 5  GROUP BY  c.customer\_id, c.name, c.customer\_since  ORDER BY  years\_with\_bank DESC, total\_loan\_amount DESC; |
| **Task 7** | **High-Performing Loans:** Identify loans with excellent repayment histories to refine lending policies and highlight successful products. |
| **Solution 7** | SELECT  l.loan\_id,  l.loan\_purpose,  sum(l.loan\_amount),  avg(l.repayment\_history)  FROM  cross\_river\_bank.loan\_table AS l  JOIN  cross\_river\_bank.customer\_table AS c ON l.customer\_id = c.customer\_id  WHERE  l.repayment\_history >= 9  AND l.loan\_status = 'Closed'  group by l.loan\_purpose, l.loan\_id |
| **Task 8** | **Age-Based Loan Analysis:** Analyze loan amounts disbursed to customers of different age groups to design targeted financial products. |
| **Solution 8** | 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,  COUNT(l.loan\_id) AS total\_loans,  SUM(l.loan\_amount) AS total\_loan\_amount,  AVG(l.loan\_amount) AS average\_loan\_amount  FROM  cross\_river\_bank.loan\_table AS l  JOIN  cross\_river\_bank.customer\_table AS c ON l.customer\_id = c.customer\_id  GROUP BY  age\_group  ORDER BY  total\_loan\_amount DESC; |
| **Task 9** | **Seasonal Transaction Trends:** Examine transaction patterns over years and months to identify seasonal trends in loan repayments. |
| **Solution 9** | SELECT  YEAR(STR\_TO\_DATE(t.transaction\_date, '%m/%d/%Y')) AS transaction\_year,  MONTH(STR\_TO\_DATE(t.transaction\_date, '%m/%d/%Y')) AS transaction\_month,  COUNT(t.transaction\_id) AS total\_transactions,  SUM(t.transaction\_amount) AS total\_transaction\_amount,  AVG(t.transaction\_amount) AS average\_transaction\_amount  FROM  cross\_river\_bank.transaction\_table AS t  GROUP BY  transaction\_year, transaction\_month  ORDER BY  transaction\_year ASC, transaction\_month ASC; |
| **Task 10** | **Fraud Detection:** Highlight potential fraud by identifying mismatches between customer address locations and transaction IP locations. |
| **Solution 10** | SELECT  t.transaction\_id,  t.customer\_id,  c.address AS customer\_address,  t.ip\_location AS transaction\_ip\_location,  t.transaction\_date,  t.transaction\_amount  FROM  cross\_river\_bank.transaction\_table AS t  JOIN  cross\_river\_bank.customer\_table AS c ON t.customer\_id = c.customer\_id  WHERE  c.address NOT LIKE CONCAT('%', t.ip\_location, '%')  ORDER BY  t.transaction\_date DESC; |
| **Task 11** | **Repayment History Analysis:** Rank loans by repayment performance using window functions. |
| **Solution 11** | SELECT  l.loan\_id,  l.customer\_id,  l.loan\_amount,  l.repayment\_history,  l.loan\_status,  RANK() OVER (ORDER BY l.repayment\_history DESC, l.loan\_amount DESC) AS repayment\_rank  FROM  cross\_river\_bank.loan\_table AS l  WHERE  l.loan\_status IN ('Closed', 'Approved')  ORDER BY  repayment\_rank ASC; |
| **Task 12** | **Credit Score vs. Loan Amount:** Compare average loan amounts for different credit score ranges. |
| **Solution 12** | SELECT  CASE  WHEN c.credit\_score < 500 THEN 'Very Poor'  WHEN c.credit\_score BETWEEN 500 AND 599 THEN 'Poor'  WHEN c.credit\_score BETWEEN 600 AND 699 THEN 'Fair'  WHEN c.credit\_score BETWEEN 700 AND 799 THEN 'Good'  ELSE 'Excellent'  END AS credit\_score\_range,  COUNT(l.loan\_id) AS total\_loans,  AVG(l.loan\_amount) AS average\_loan\_amount  FROM  cross\_river\_bank.loan\_table AS l  JOIN  cross\_river\_bank.customer\_table AS c ON l.customer\_id = c.customer\_id  GROUP BY  credit\_score\_range  ORDER BY  average\_loan\_amount DESC; |
| **Task 13** | **Top Borrowing Regions:** Identify regions with the highest total loan disbursements. |
| **Solution 13** | SELECT  c.address AS region,  COUNT(l.loan\_id) AS total\_loans,  SUM(l.loan\_amount) AS total\_loan\_disbursements,  AVG(l.loan\_amount) AS average\_loan\_amount  FROM  cross\_river\_bank.loan\_table AS l  JOIN  cross\_river\_bank.customer\_table AS c ON l.customer\_id = c.customer\_id  GROUP BY  c.address  ORDER BY  total\_loan\_disbursements DESC; |
| **Task 14** | **Early Repayment Patterns:** Detect loans with frequent early repayments and their impact on revenue. |
| **Solution 14** | SELECT  l.loan\_id,  l.customer\_id,  l.loan\_amount,  COUNT(t.transaction\_id) AS early\_repayments\_count,  SUM(t.transaction\_amount) AS total\_early\_repayments,  (SUM(t.transaction\_amount) / l.loan\_amount) \* 100 AS early\_repayment\_percentage  FROM  cross\_river\_bank.loan\_table AS l  JOIN  cross\_river\_bank.transaction\_table AS t ON l.loan\_id = t.loan\_id  WHERE  t.transaction\_type = 'Prepayment'  GROUP BY  l.loan\_id, l.customer\_id, l.loan\_amount  HAVING  early\_repayment\_percentage > 10  ORDER BY  early\_repayments\_count DESC, total\_early\_repayments DESC; |
| **Task 15** | **Feedback Correlation:** Correlate customer feedback sentiment scores with loan statuses. |
| **Solution 15** | SELECT  CASE  WHEN t.remarks= 'Late penalty.' THEN 'Late penalty.'  WHEN t.remarks= 'On-time payment.' THEN 'On-time payment.'  WHEN t.remarks= 'Partial payment.' THEN 'Partial payment.'  WHEN t.remarks= 'Payment missed.' THEN 'Payment missed.'  END AS Feedback\_sentiment,  l.loan\_status,  SUM(l.loan\_amount) AS total\_loan\_amount,  AVG(l.loan\_amount) AS average\_loan\_amount  FROM  cross\_river\_bank.loan\_table AS l  JOIN  cross\_river\_bank.transaction\_table AS t ON l.customer\_id = t.customer\_id  GROUP BY  Feedback\_sentiment , l.loan\_status  ORDER BY  total\_loan\_amount DESC; |

Github Link : [Vivekdash123/Fraud-Detection-and-Risk-Analysis-in-SQL](https://github.com/Vivekdash123/Fraud-Detection-and-Risk-Analysis-in-SQL)