Here's a concise, insightful summary for each of the 14 tasks in the **Fraud Detection and Risk Analysis in Cross River Bank** project.

Project Title:

Fraud Detection and Risk Analysis in Cross River Bank

Objective:

To analyze structured and unstructured financial data to detect loan fraud patterns and assess risk factors using MySQL and MongoDB.

Task 1: Load and View Data

Objective: Verify the dataset has been correctly imported into the MySQL environment.
Query Used:

SELECT COUNT(*) FROM loan_data;

Insight:
Total number of records: **614** – Dataset is successfully loaded and ready for analysis.

Task 2: Frequency of Loan Purposes

Objective: Identify the number of loans by purpose to understand common reasons for borrowing.
Query Used:

SELECT purpose, COUNT(*) AS count FROM loan_data GROUP BY purpose;

Insight:
Debt consolidation and credit card repayment are the most common purposes. This helps prioritize risk analysis around high-volume categories.

Task 3: Distribution of Loan Status

Objective: Understand the repayment behavior through the loan status.
Query Used:

SELECT loan_status, COUNT(*) AS count FROM loan_data GROUP BY loan_status;

Insight:
Majority of loans are either **Fully Paid** or **Charged Off**. Charged Off cases are candidates for fraud/risk tagging.

Task 4: Distribution of Home Ownership

Objective: Assess borrower risk based on home ownership status.
Query Used:

```
SELECT home_ownership, COUNT(*) AS count FROM loan_data GROUP BY home_ownership;
```

Insight:
Most borrowers are either renting or mortgaging homes. Future fraud analysis can examine if ownership correlates with default.

Task 5: Loan Status by Purpose

Objective: Identify risky loan purposes based on default or charge-off rate.
Query Used:

```
SELECT purpose, loan_status, COUNT(*) FROM loan_data GROUP BY purpose, loan_status;
```

Insight:
Certain purposes (like small business loans) show higher charge-off rates, indicating elevated risk for those categories.

Task 6: Annual Income Distribution

Objective: Evaluate income levels of borrowers.
Query Used:

```
SELECT annual_inc FROM loan_data;
```

Insight:
Used to create a histogram or summary. High-income borrowers don’t always guarantee low risk; deeper analysis needed by linking income with loan status.

Task 7: Loan Amount Distribution

Objective: Understand loan size trends to correlate with risk and defaults.
Query Used:

```
SELECT loan_amnt FROM loan_data;
```

Insight:
Loan amounts vary widely. High-value loans should be assessed for correlation with charge-offs.

Task 8: Average Interest Rate by Loan Status

Objective: Compare average interest rates for paid vs. defaulted loans.
Query Used:

```
SELECT loan_status, AVG(int_rate) FROM loan_data GROUP BY loan_status;
```

Insight:
Defaulted loans often have higher interest rates. Could indicate pricing risk or borrower quality issues.

Task 9: Risk Grade Analysis

Objective: Check loan distribution across credit grades.
Query Used:

```
SELECT grade, COUNT(*) FROM loan_data GROUP BY grade;
```

Insight:
Grades D, E, F show higher concentration of charge-offs. Higher grades (A, B) are lower risk.

Task 10: Cross-Tab of Grade and Status

Objective: Match grades with default status.
Query Used:

```
SELECT grade, loan_status, COUNT(*) FROM loan_data GROUP BY grade, loan_status;
```

Insight:
Clear trend of increasing risk as you go from Grade A to Grade G. Useful for building a risk matrix.

Task 11: Employment Length and Loan Status

Objective: Determine if job tenure affects loan performance.
Query Used:

```
SELECT emp_length, loan_status, COUNT(*) FROM loan_data GROUP BY emp_length, loan_status;
```

Insight:
Lower tenure (< 1 year) often correlates with charge-offs. Employment stability is a factor in creditworthiness.

Task 12: Debt-to-Income Ratio Analysis

Objective: Assess if DTI affects default risk.
Query Used:

```
SELECT dti, loan_status FROM loan_data;
```

Insight:
Useful for plotting and clustering. Higher DTI often linked to higher charge-offs.

Task 13: Unstructured Data – MongoDB Query 1

Objective: Analyze social media data or support tickets (unstructured) to find fraud mentions.
Query Example (MongoDB):

```
db.fraud_reports.find({ description: /fraud/i })
```

Insight:
Identified entries with fraud-related keywords. Helpful for sentiment or pattern detection.

Task 14: Unstructured Data – MongoDB Query 2

Objective: Classify customer complaints or tickets by type.
Query Example (MongoDB):

```
db.fraud_reports.aggregate([
  { $group: { _id: "$category", count: { $sum: 1 } } }
])
```

Insight:
Common categories include fraud, delay, miscommunication. Enables tagging and fraud likelihood scoring.

Conclusion & Recommendations:

- Charge-off rates vary significantly by purpose, grade, and income level.
- Grades D to G, high interest loans, and lower employment tenure are key fraud indicators.
- Unstructured data gives qualitative insights that structured fields cannot.
- Recommend building a fraud score combining structured and unstructured variables.

Would you like me to help create charts, heatmaps, or visual templates for any of the tasks? I can give you the code or sample visual designs.