

BANK ANALYTICS PROJECT

Presentation By: [Group_1]

K Arjuneswara Rao
Jacob Paul
Lakshmi Priya G
Sandyashree C
Sujata Rohit Pushpalwar



Introduction to Excel:

Microsoft Excel is one of the most widely used tools for organizing, analyzing, and cleaning data. With its user-friendly interface and powerful functions.

Excel allows users to:

- Identify and remove duplicates
- Standardize formats and fix inconsistencies
- Handle missing or incorrect values
- Convert unstructured data into usable formats

How to Clean Data in Excel



Remove Blank Rows & Columns

- Use **Go To Special → Blanks** to select empty cells
- Right-click → **Delete Rows or Columns**
- Use formulas like `=TRIM()`, `=PROPER()`, `=UPPER()`
- Fix capitalization and spacing issues
 - Go to **Data → Remove Duplicates**
- Select target columns and confirm
- Use **Text to Columns** or **Format Cells**
- Fix numbers stored as text
 - Highlight blanks using **Conditional Formatting**
- Fill them with formulas like `=IF(A2="", "N/A", A2)`
- Press **Ctrl + E**
- Excel auto-fills based on the pattern
 - Select data range → Press **Ctrl + T**
- Enables better sorting, filtering, and styling



How to Create a Dashboard in Excel:

- **Collect and Clean Data**

Organize your data in a table format without merged cells or blanks.

- **Name Your Data Table**

Select the data → Press **Ctrl + T** to create an Excel Table (easier for formulas and charts).

- **Create Pivot Tables**

Summarize key metrics using Pivot Tables (e.g., sales by region, profit by category).

- **Insert Charts and Graphs**

Create bar, line, pie, or combo charts based on your summary data.

- **Add Slicers/Filters**

Use slicers with Pivot Tables to make your dashboard interactive.

- **Create a Dashboard Sheet**

Insert a new worksheet, arrange charts and visuals neatly with good formatting.



Bank Analytics Dashboard

BANK LOAN PERFORMANCE DASHBOARD

The dashboard displays the following key metrics:

- Grade:** A, B
- Total Loan Amount Funded:** ₹16285.1K
- Total Loans:** 1464
- Total Collection:** ₹18285.3K
- Total Interest:** ₹3526.0K
- Branch-Wise Performance:**
 - Total Fees:** ₹27.1K
 - Total Interest:** ₹3526.0K
 - Total Revenue:** ₹3553.1K
- State-Wise Loan:** ₹353.5K

Filters available on the left side include:

- Grade: A, B
- State Name: UTTAR PRA...
- Religion: Christian, Hindu
- Purpose: Mobile Ph..., Others
- Is Default: N

Call-to-action button: **CLICK ME**

Key Findings:

Top Branches Displayed:
ASANSOL, AZAMGARH

Total Interest: ₹632.8K

State-Wise Loan Total:
₹7,888.1K

Active Loans dominate with over
10,000+ accounts

Challenges Faced:

Q. Data Inconsistency
Across Sources?

Sol. Implement Data
Cleaning, Use Power Query
to standardize formats

Debit and Credit Dashboard

Debit and Credit banking

TOTAL CREDIT AMOUNT
₹127603.4K

TOTAL DEBIT AMOUNT
₹127603.4K

CREDIT-DEBIT RATIO
1.0025

Net Transaction Amount
₹318.1K

Days (Transaction Date)
Months (Transaction Date)

01-Apr
01-Aug
01-Dec

Month...
Jan
Feb
Mar

Suburban Branch
42176276.35

North Branch
41677131.72

Main Branch
42839450.41

East Branch
42697111.17

Downtown Branch
42587216.19

City Center Branch
42911469.79

Count of Transaction Method
Total

Transaction Type
Credit
Debit

Branch
City Center Branch
Downtown Branch
East Branch

ACCOUNT ACTIVITY
0.0004

Sum of Amount

Count of Transaction Method

Total

Days (Transaction Date)

Months (Transaction Date)

+ -

Key Findings:

Negative Net Transaction Amount: -₹351.9K

Credit Transactions: 11803.3K

Credit-Debit Ratio < 1

Challenges Faced:

Q. Low Visibility into Transaction Categories?

Sol. Introduced additional fields (tags), that classify transactions by type.



SQL

Introduction to SQL:

SQL (Structured Query Language) is a standard programming language used to **store, manage, and manipulate data in relational databases**.

It allows users to:

- **Create and modify** database structures (tables, views)
- **Insert, update, delete, and retrieve** data
- **Filter, sort, and join** data from multiple tables
- Perform **aggregation and analysis** using functions like SUM(), COUNT(), etc.
- **Control user access and permissions**
- **Create stored procedures and triggers**



How to import Excel data to Sql

For MySQL:

- Save Excel as CSV
- Open MySQL Workbench
- Create a Table
- Import CSV File

```
Example: LOAD DATA INFILE 'C:/path/to/file.csv'  
        INTO TABLE your_table  
        FIELDS TERMINATED BY ','  
        ENCLOSED BY ""  
        LINES TERMINATED BY '\n'  
        IGNORE 1 ROWS;
```

```
MySQL Workbench Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator Schemas File 64* SQL File 65* projects* SQL File 69* crowdfundng(project)* SQL File 68* calendar calendar SQL File 71* project 2(insurance) bank (projects)* SQL File 74* debit and credit(sql) SQL 4*
1 * use bank;
2 * DESCRIBE bank;
3 * SHOW COLUMNS FROM bank;
4 * .... Total loan amount ...
5 * SELECT
6   SUM(Funded_Amount) AS total_loan_amount_funded
7   FROM bank;
8 * .... Total loan ...
9 * SELECT
10  COUNT('Client Id') AS total_loans
11  FROM bank;
12 * .... total collection ...
13 * SELECT
14  SUM(COALESCE(`Total Rec Prncp`, 0) + COALESCE(`Total Rec int`, 0)) AS total_collection
15  FROM bank;
16 * .... total interest ...
17 * SELECT
18  SUM(COALESCE(`Total Rec int`, 0)) AS total_interest_revenue
19  FROM bank;
20 * .... branch wise performance ...
21 * SELECT
22   'Branch Name' AS branch,
23   SUM(COALESCE(`Total Rec int`, 0)) AS total_interest,
24   SUM(COALESCE(`Total Fees`, 0))
25   + COALESCE(`Total Rec Late Fee`, 0)
Output Action Output # Time Action Message Duration / Fetch
Object Info Session
Type here to search 24°C Cloudy ENG 12:06 AM IN 07-Aug-25
MySQL Workbench Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator Schemas File 64* SQL File 65* projects* SQL File 69* crowdfundng(project)* SQL File 68* calendar calendar SQL File 71* project 2(insurance) bank (projects)* SQL File 74* debit and credit(sql) SQL 4*
1 * use bank;
2 * DESCRIBE bank;
3 * SHOW COLUMNS FROM bank;
4 * .... Total loan amount ...
5 * SELECT
6   SUM(Funded_Amount) AS total_loan_amount_funded
7   FROM bank;
8 * .... Total loan ...
9 * SELECT
10  COUNT('Client Id') AS total_loans
11  FROM bank;
12 * .... total collection ...
13 * SELECT
14  SUM(COALESCE(`Total Rec Prncp`, 0) + COALESCE(`Total Rec int`, 0)) AS total_collection
15  FROM bank;
16 * .... total interest ...
17 * SELECT
18  SUM(COALESCE(`Total Rec int`, 0)) AS total_interest_revenue
19  FROM bank;
20 * .... branch wise performance ...
21 * SELECT
22   'Branch Name' AS branch,
23   SUM(COALESCE(`Total Rec int`, 0)) AS total_interest,
24   SUM(COALESCE(`Total Fees`, 0))
25   + COALESCE(`Total Rec Late Fee`, 0)
Output Action Output # Time Action Message Duration / Fetch
Object Info Session
Type here to search 24°C Cloudy ENG 12:07 AM IN 07-Aug-25
```

```
MySQL Workbench Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator Schemas File 64* SQL File 65* projects* SQL File 69* crowdfundng(project)* SQL File 68* calendar calendar SQL File 71* project 2(insurance) bank (projects)* SQL File 74* debit and credit(sql) SQL 4*
1 * use bank;
2 * DESCRIBE bank;
3 * SHOW COLUMNS FROM bank;
4 * .... total credit amount ...
5 * SELECT sum(Amount) AS total_credit_amount
6   FROM debit;
7 * WHERE Transaction Type = 'Credit';
8 * .... total debit amount ...
9 * SELECT sum(Amount) AS total_debit_amount
10  FROM debit;
11 * WHERE Transaction Type = 'Debit';
12 * .... credit to debit ratio ...
13 * SELECT
14  sum(CASE WHEN `Transaction Type` = 'Credit' THEN Amount ELSE 0 END) -
15  sum(CASE WHEN `Transaction Type` = 'Debit' THEN Amount ELSE 0 END), 0
16 * AS credit_to_debit_ratio
17  FROM debit;
18 * .... Net transaction amount ...
19 * SELECT
20  sum(CASE WHEN `Transaction Type` = 'Credit' THEN Amount ELSE 0 END) -
21  sum(CASE WHEN `Transaction Type` = 'Debit' THEN Amount ELSE 0 END)
22 * AS net_transaction_amount
23  FROM debits;
24 * .... Account activity ratio ...
25 * SELECT
26   'Customer ID',
27   COUNT(*) / MAX(Balance) AS account_activity_ratio
Output Action Output # Time Action Message Duration / Fetch
Object Info Session
Type here to search 24°C Cloudy ENG 12:08 AM IN 07-Aug-25
```

```
MySQL Workbench Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator Schemas File 64* SQL File 65* projects* SQL File 69* crowdfundng(project)* SQL File 68* calendar calendar SQL File 71* project 2(insurance) bank (projects)* SQL File 74* debit and credit(sql) SQL 4*
21 * SELECT
22   'Branch Name' AS branch,
23   SUM(COALESCE(`total Rec int`, 0)) AS total_interest,
24   SUM(COALESCE(`Total Fees`, 0))
25   + COALESCE(`Total Rec Late Fee`, 0)
26   * COALESCE(`Recoveries`, 0)
27   + COALESCE(`Collection Recovery fee`, 0)
28   ) AS total_fees_and_revenue,
29   SUM(
30   COALESCE(`Total Rec Prncp`, 0)
31   + COALESCE(`Total Rec int`, 0)
32   + COALESCE(`Total Fees`, 0)
33   + COALESCE(`Total Rec Late Fee`, 0)
34   + COALESCE(`Recoveries`, 0)
35   + COALESCE(`Collection Recovery fee`, 0)
36   ) AS total_collection
37  FROM bank
38  GROUP BY Branch Name
39  ORDER BY total_collection DESC;
40 * .... state wide loan ...
41 * SELECT
42   'State Abb.' AS state,
43   COUNT('Client Id') AS total_loans,
44   SUM(COALESCE(`Funded_Amount`, 0)) AS total_funded_amount
45  FROM bank
Output Action Output # Time Action Message Duration / Fetch
Object Info Session
Type here to search 24°C Cloudy ENG 12:07 AM IN 07-Aug-25
MySQL Workbench Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator Schemas File 64* SQL File 65* projects* SQL File 69* crowdfundng(project)* SQL File 68* calendar calendar SQL File 71* project 2(insurance) bank (projects)* SQL File 74* debit and credit(sql) SQL 4*
24 * .... Account activity ratio ...
25 * SELECT
26   'Customer ID',
27   COUNT(*) / MAX(Balance) AS account_activity_ratio
28  FROM debits;
29  GROUP BY Customer ID ;
30 * .... Transaction per day/week/month ...
31 * .... Per Day ...
32 * SELECT
33  STR_TO_DATE(Transaction Date, '%Y-%m-%d') AS transaction_day,
34  COUNT(*) AS transactions_per_day
35  FROM debit
36  GROUP BY transaction_day
37  ORDER BY transaction_day
38 * .... Per Month ...
39 * SELECT
40  DATE_FORMAT(STR_TO_DATE(Transaction Date, '%Y-%m-%d'), '%Y-%m') AS month,
41  COUNT(*) AS transactions_per_month
42  FROM debit
43  GROUP BY month
44  ORDER BY month;
45 * .... total transaction amount by branch ...
46 * SELECT Branch, sum(Amount) AS total_transaction_amount
47  FROM debit
Output Action Output # Time Action Message Duration / Fetch
Object Info Session
Type here to search 24°C Cloudy ENG 12:08 AM IN 07-Aug-25
```

```
MySQL Workbench Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator Schemas File 64* SQL File 65* projects* SQL File 69* crowdfundng(project)* SQL File 68* calendar calendar SQL File 71* project 2(insurance) bank (projects)* SQL File 74* debit and credit(sql) SQL 4*
48 * ....
49 * SELECT Branch, sum(Amount) AS total_transaction_amount
50  FROM debit
Output Action Output # Time Action Message Duration / Fetch
Object Info Session
Type here to search 24°C Cloudy ENG 12:08 AM IN 07-Aug-25
```



Steps to Run/Execute a Query:

- Write Your SQL Query
- Execute the Query
 - Click the "lightning bolt" icon in the toolbar
 - Or press **Ctrl + Enter** (Windows) / **Cmd + Enter** (Mac)
- View Results
 - The results will be displayed in the "**Results Grid**" at the bottom.

The screenshot shows the MySQL Workbench interface. The main window displays a query editor with the following SQL code:

```
-- total transaction amount by branch
SELECT Branch, SUM(Amount) AS total_transaction_amount
FROM debit
GROUP BY Branch
ORDER BY total_transaction_amount DESC;

-- transaction volume branch
SELECT `Bank Name` , SUM(Amount) AS total_transaction_volume
FROM debit
GROUP BY `Bank Name`
ORDER BY total_transaction_volume DESC;

-- transaction method distribution --
SELECT
    `Transaction Method` AS Method,
    COUNT(*) AS MethodCount
FROM debit
GROUP BY `Transaction Method`;

-- branch transaction growth --
SELECT
    Branch,
    ((SUM(CASE WHEN `Transaction Date` >= 20230101 AND `Transaction Date` < 20240101 THEN Amount ELSE 0 END) -
    SUM(CASE WHEN `Transaction Date` >= 20220101 AND `Transaction Date` < 20230101 THEN Amount ELSE 0 END)) /
    SUM(CASE WHEN `Transaction Date` >= 20220101 AND `Transaction Date` < 20230101 THEN Amount ELSE 0 END)) * 100 AS growth_rate
FROM debit
GROUP BY Branch;
-- high risk transaction flag --
```

The interface includes a toolbar with various icons, a Navigator pane showing database schemas like bank, classimode, customers_orders, and debit, and an Output pane at the bottom showing the results of the executed queries.



Introduction to Power BI

Power BI is a powerful business analytics tool developed by Microsoft that allows users to visualize data, share insights, and make data-driven decisions. It offers interactive visualizations, real-time dashboards, and advanced analytics capabilities using simple drag-and-drop features — all without requiring extensive programming knowledge.

With Power BI, users can:

- Connect to a wide range of data sources (Excel, SQL Server, Azure, SharePoint, APIs, etc.)
- Clean, transform, and model data with Power Query and DAX
- Build interactive dashboards and reports
- Share insights across teams or embed them into apps and websites.



Characteristics of Power Bi

- Data Compression
- Efficient Data Modeling
- Data Reduction Techniques
- Incremental Refresh
- DirectQuery & Live Connections
- Custom Visuals and Paginated Reports



How To Import Data In Power Bi From SQL

Open Power BI Desktop

Click on “Get Data” → Choose “SQL Server”.

Enter Server Name and Database Name

- Example: `localhost\SQLEXPRESS` or `192.168.1.100`
- Database name is optional

Choose Data Connectivity Mode

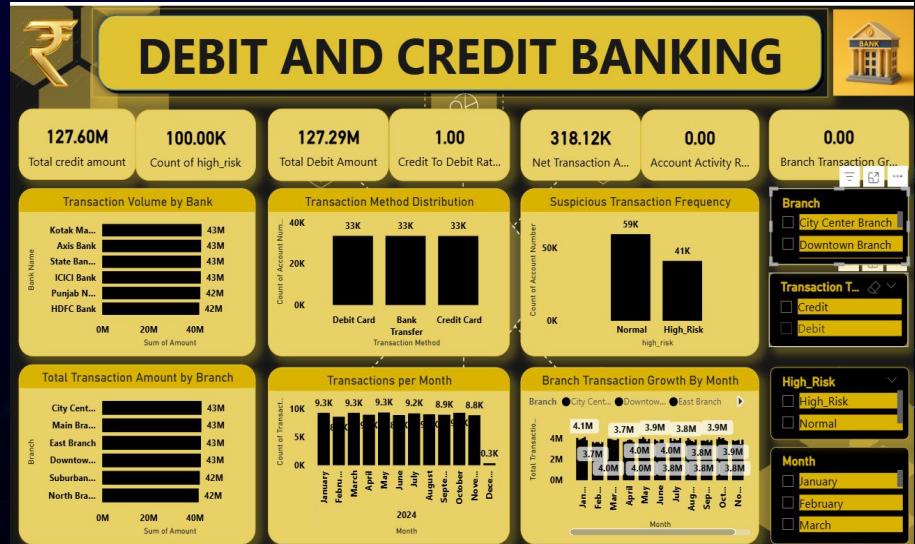
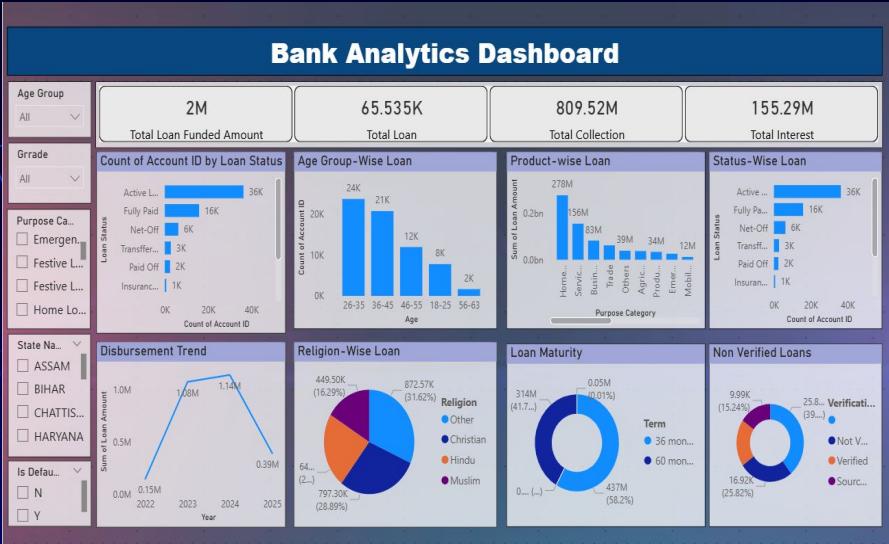
Select Authentication Method

Choose Tables or Write SQL Query

Click “Load” or “Transform Data”

Build Visuals in Power BI

Dashboards





Findings

Bank Analytics Dashboard:

- Total Loan Funded: 2M
- Total Loan Count: 65.535K
- Total Collection: 809.52M
- Total Interest: 155.29M
- Highest loan count from 26–35 age group
- Active loans: 36K
- Major products: Home Loan, Savings, Business
- 31.6% loans taken by Christians

Debit and Credit Dashboard:

- Total Credit Amount: 127.60M
- Total Debit Amount: 127.29M
- Credit to Debit Ratio: 1.00
- High-Risk Transactions: 100K



Introduction to Tableau:

Tableau is a powerful **data visualization and business intelligence tool** used to analyze data and create interactive, shareable dashboards. It helps users turn raw data into meaningful insights through visual storytelling, without needing advanced programming skills.

With Tableau, users can:

- Connect to various data sources
- Clean, transform, and prepare data
- Create interactive visualizations
- Build dynamic dashboards
- Perform real-time data analysis
- Share reports and dashboards securely
- Identify trends and patterns
- Use calculated fields and table calculations
- Collaborate across teams by commenting and sharing visual stories.



Characteristics of Tableau

- User-Friendly Interface
- Data Connectivity
- Interactive Dashboards
- Real-Time Analysis
- Powerful Visualizations
- Data Blending and Joining
- Sharing and Collaboration
- Security and Access Control



How to Import Data in Tableau from SQL:

Open Tableau Desktop:

Click on “**Connect**” → choose “**Microsoft SQL Server**” (or your specific SQL database).

Enter the following details:

- **Server Name**
- **Database Name** (optional)
- **Authentication** (Windows/SQL login)
- **Username and Password**

Click “**Sign In**”

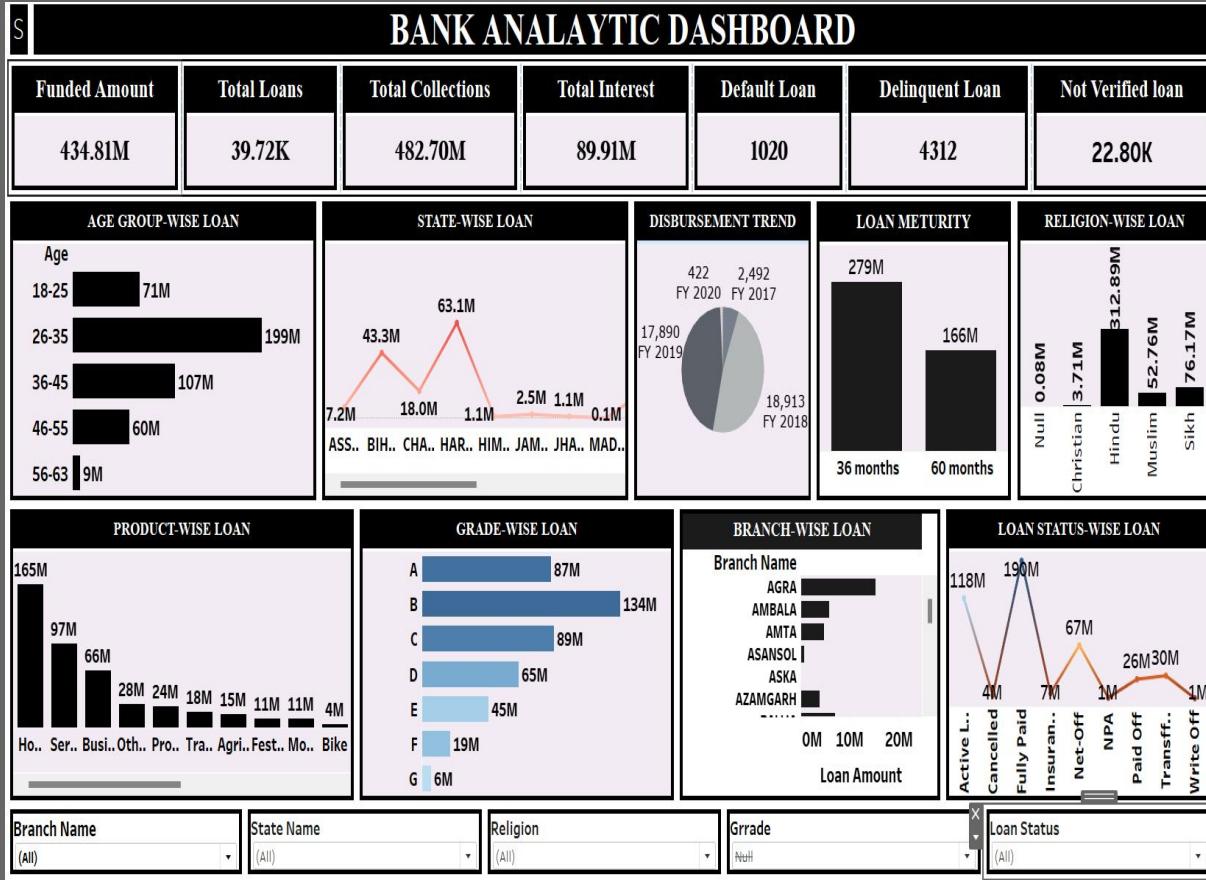
Once connected, select the desired **database** and **tables**

Drag tables to the **canvas** (use joins or relationships if needed)

Click “**Sheet**” to start analyzing your SQL data



Dashboards



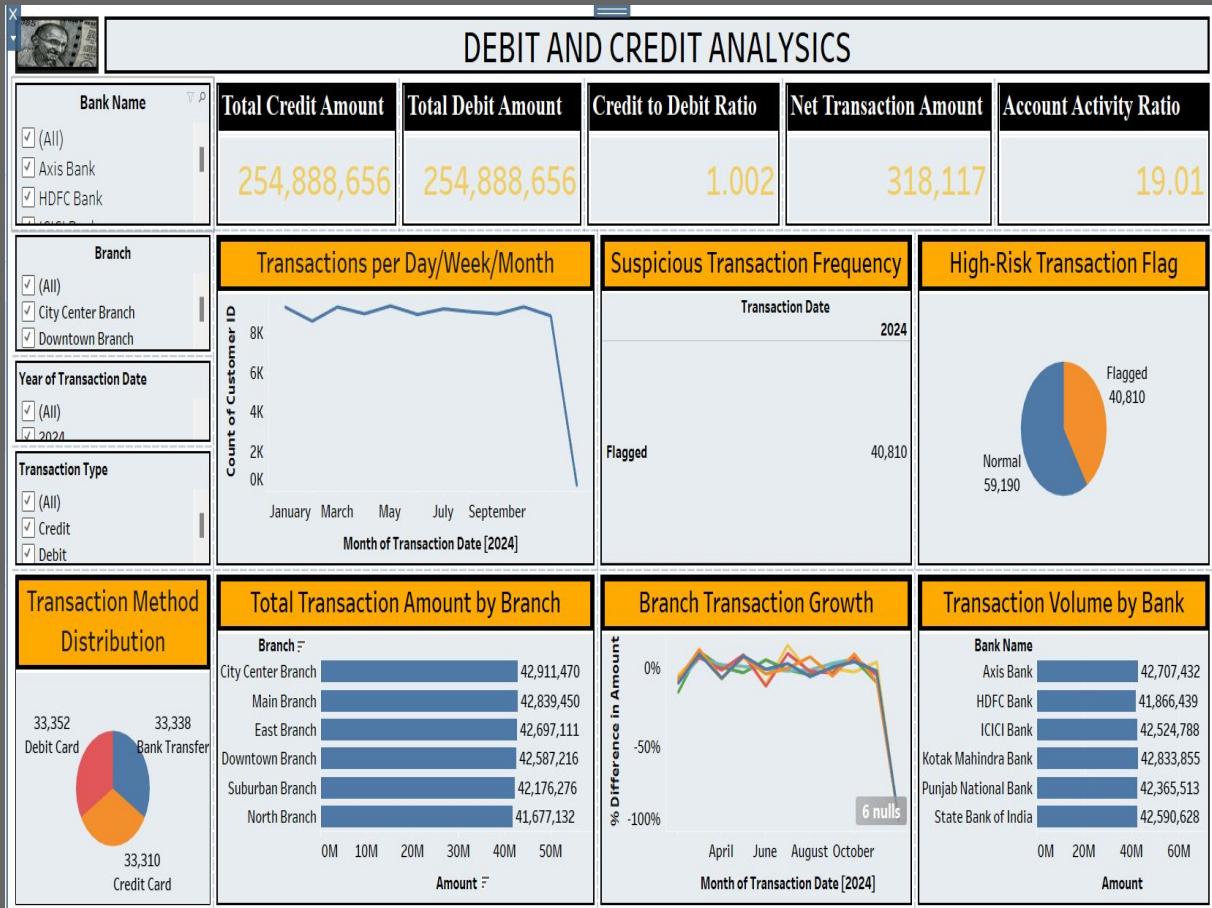
Key Findings:

- **Young Borrowers Drive Loan Volume:** It indicating a strong demand among working professionals.
- **Home Loans Dominate Product-Wise Lending.**
- **Disbursement Peaks in FY2019.**

Challenge Faced:

Q. Lack of Real-Time Risk Alerts or Predictive Insights?

Sol. Create a calculated field in Tableau to assign risk scores based on loan grade, amount, and tenure. Then, classify them into High, Medium, or Low Risk visually for proactive decision-making.



Key Findings:

- Suspicious Transactions.
- Branch Performance
- Transaction Method Parity
- Account Activity Ratio

Challenges Faced:

Q. Handling Null or Missing Values in Monthly Growth Charts(Branch Transaction Growth)?

Sol. Use the **IFNULL()** function in calculated fields to replace nulls with 0 or “No Data”, ensuring a clean and complete visual trend.

Conclusion

This project demonstrates the power of modern data tools and techniques in enhancing decision-making within the banking sector. Through effective visualizations and in-depth data analysis, the dashboard enables banks to minimize risks, optimize lending approaches, and gain deeper insights into customer behavior. It offers a data-driven approach that promotes sustainable growth and boosts operational performance in the financial industry.