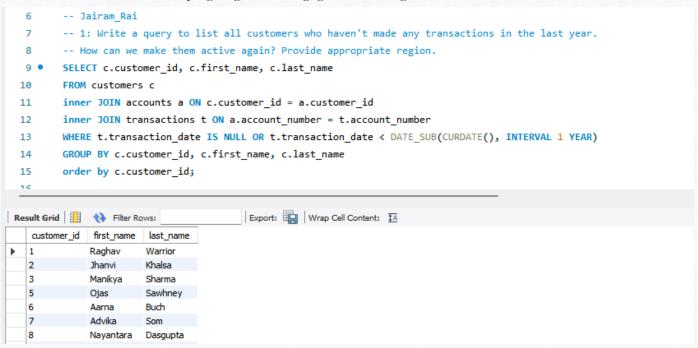
# Title: Banking Data Analysis in SQL Jairam rai

# Description:

This presentation summarizes a comprehensive analysis of banking data using SQL. The analysis addresses various queries related to customer transactions, branch performance, and potential fraudulent activities. It provides insights into customer behavior, branch rankings, and transactional patterns over different time periods. The goal is to enhance understanding of banking operations and identify areas for improvement in customer engagement and fraud detection.

# 1.) How to Make Inactive Customers Active Again:

- Key Insights:
- Customer Segmentation:
  - Segmented inactive customers based on various factors such as age, account type, and region.
  - This segmentation helps in understanding which customer groups are more prone to inactivity.
- Reasons for Inactivity:
  - Possible reasons could include dissatisfaction with services, lack of need for banking services, or switching to a competitor.
  - Understanding these reasons is crucial for developing targeted re-engagement strategies.



# Strategies to Make Them Active Again:

#### Personalized Communication:

- Reach out to inactive customers with personalized messages and offers.
- Use customer data to highlight relevant products and services that may reignite their interest.

#### • Exclusive Offers and Incentives:

- Provide special offers, such as fee waivers, higher interest rates on deposits, or cashback on transactions.
- Incentives can encourage customers to re-engage with their accounts.

## Feedback and Support:

- Seek feedback from inactive customers to understand their reasons for inactivity.
- Offer dedicated support to address any issues or concerns they may have had

## Region-Specific Strategy:

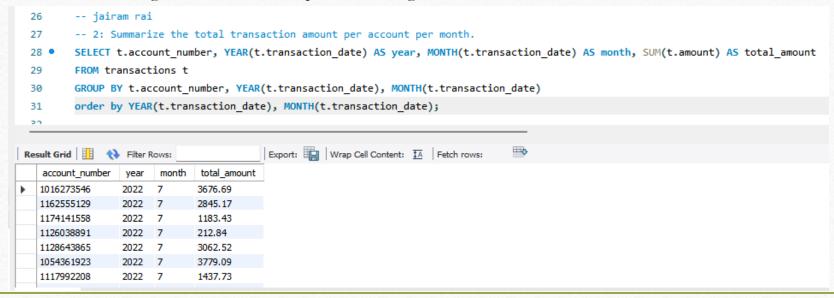
• Host local events or workshops in major cities within Maharashtra, such as Mumbai, Pune, and Nagpur, to engage customers directly. Offer region-specific promotions that cater to the local culture or economic conditions of Maharashtra, such as special rates on loans or investment opportunities relevant to the region. By implementing these strategies and focusing on Maharashtra, the bank can effectively re-engage inactive customers in the state and encourage them to become active again.

# 2.) Total Transaction Amount per Account per Month

## **Key Insights:**

- Identify High-Activity Accounts: Accounts with consistently high transaction amounts can be flagged for further analysis or special attention.
- **Monthly Trends:** Understanding the monthly transaction trends can help in predicting future transaction volumes and planning accordingly.
- Seasonal Patterns: Look for any seasonal patterns in the transaction data, such as increased transactions during festival seasons or end-of-year periods.

By analyzing the total transaction amount per account per month, we gain valuable insights into customer behavior and account activity, which can inform strategic decisions and improve banking services.



# 3.) Ranking Branches by Total Deposits in the Last Quarter

#### **Key Insights:**

- Top-Performing Branches: Identify the branches with the highest total deposits. These branches are likely to have a strong customer base and effective customer engagement strategies.
- Strategic Planning: Use the ranking to inform strategic decisions, such as resource allocation, marketing efforts, and targeted customer engagement activities in high-performing branches.

**Conclusion:** Ranking branches by total deposits in the last quarter provides valuable insights into branch performance and customer engagement. This analysis helps in identifying top-performing branches, understanding regional dynamics, and making informed decisions to drive growth and improve banking services.

```
-- jairam rai
44
        -- 3 Rank branches based on the total amount of deposits made in the last quarter.
45
        SELECT b.branch_id, ROUND(SUM(CASE WHEN t.transaction_type = 'deposit' THEN t.amount ELSE 0 END),2) AS totaldeposit,
47
        DENSE_RANK() OVER (ORDER BY SUM(CASE WHEN t.transaction_type = 'deposit' THEN t.amount ELSE @ END)DESC) AS branch_rank
48
        FROM Transactions AS t
        JOIN Accounts AS a
49
        ON t.account number=a.account number
50
51
        JOIN branches AS b
        ON b.branch_id=a.branch_id
52
        WHERE t.transaction date>= DATE SUB(CURDATE(), INTERVAL 3 MONTH)
53
54
        GROUP BY b.branch id
        ORDER BY branch rank asc;
Export: Wrap Cell Content: IA
   branch_id totaldeposit branch_rank
           23688.58
           18448.58
           16348.46
           16109.1
           12548.97
           11969.37
```

# 4.) Top Customer by Total Deposits

#### **Key Insights:**

- Top Depositor: Identify the customer with the highest total deposits, which reflects their trust and engagement with the bank.
- Customer Value: Recognize the importance of high-value customers and their contribution to the bank's deposit base.
- Relationship Management: Use this information to prioritize relationship management and personalized services for high-value customers.
- Loyalty and Retention: Understanding the top depositors helps in crafting tailored loyalty programs to retain these valuable customers.

Conclusion: Finding the customer with the highest total deposits provides valuable insights into customer engagement and the bank's deposit base. This information is crucial for prioritizing relationship management efforts, enhancing customer loyalty, and ensuring the retention of high-value customers

```
74
         -- jairam rai
         -- 4: Find the name of the customer who has deposited the highest amount.
 75
         select c.customer id ,c.First name , c.Last name , sum(t.amount) as amount
         from customers as c join
 77
 78
         accounts as a on
         c.customer_id = a.customer_id
 79
         join transactions as T on
 80
         t.account number = a.account number
 81
         where Transaction_type = "Deposit"
 82
         group by c.customer_id,c.First_name , c.Last_name
 83
         order by sum(t.amount) desc limit 1;
 84
                                          Export: Wrap Cell Content: TA Fetch rows:
Result Grid
   customer_id
             First name
                       Last name
                                   amount
▶ 34
              Dishani
                        Deol
                                  53763.75
```

We have seen a lot of fraud activity.

# 5.) Identifying Potential Fraudulent Activity

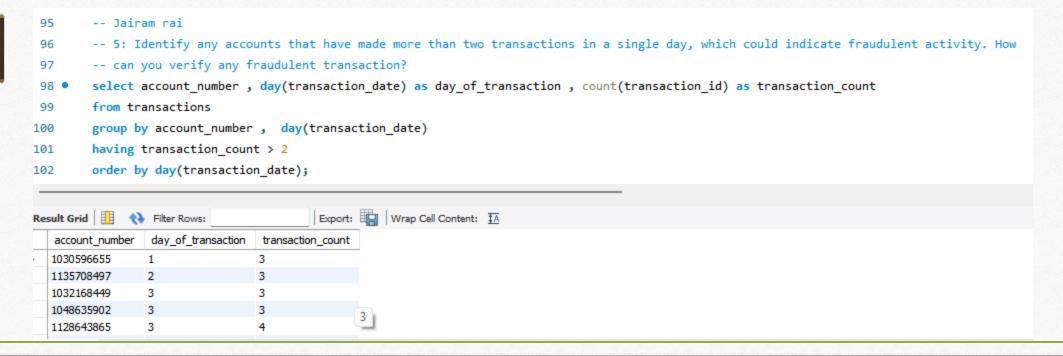
#### How to Verify Any Fraudulent Transaction:

- 1. Cross-Check with Customer Behavior:
  - 1. Compare the identified transactions with the customer's historical behavior. Look for deviations from their typical transaction patterns.
  - 2. Consider factors such as transaction amounts, locations, and times.

#### 2. Yes We have seen a lot of fraud activity

#### Conclusion:

• Ensuring the security of customer accounts and minimizing fraudulent activities through diligent analysis and verification.



# 6.) Average Number of Transactions per Customer per Account per Month

## **Key Insights:**

- Customer Behavior
  - Understand customer transaction patterns on a monthly basis.
  - Identify peak and low activity periods for better resource allocation and service planning.
- Account Utilization:
  - Analyze the usage of different accounts by customers.
  - Determine which accounts are most active and which may need targeted campaigns to increase usage.

#### Conclusion:

Monitoring the average number of transactions per customer per account per month provides valuable insights into customer activity.

```
09
        -- Jairam rai 6
10
        -- 6: Calculate the average number of transactions per customer per account per month over the last year.
11 •
       SELECT a.customer id, a.account number, YEAR(t.transaction date) AS transaction year,
12
        MONTH(t.transaction_date) AS transaction_month,
13
            COUNT(t.transaction id) AS transaction count,
            COUNT(t.transaction_id) / 12 AS avg_transactions_per_month_last_year
14
15
        FROM transactions t
        JOIN accounts a ON t.account_number = a.account_number
16
        WHERE t.transaction date >= DATE SUB(CURDATE(), INTERVAL 1 YEAR)
17
        GROUP BY a.customer_id, a.account_number, transaction_year, transaction_month
18
19
        ORDER BY a.customer id, a.account number, transaction year, transaction month;
esult Grid
             Filter Rows:
                                          Export: Wrap Cell Content: IA
  customer id
             account_number
                            transaction_year
                                           transaction_month
                                                           transaction_count
                                                                           avg_transactions_per_month_last_year
             1032168449
                           2023
                                                                          0.0833
                                                                          0.0833
                           2023
                                                                          0.1667
             1032168449
             1032168449
                           2024
                                                                          0.0833
```

# 7.) Daily Transaction Volume Analysis for the Past Month

#### **Key Insights:**

#### Peak Transaction Days:

- Recognized days with significantly higher transaction volumes.
- These peaks could indicate salary payment days, promotional events, or other significant activities.

#### Low Transaction Days:

- Identified days with lower transaction volumes.
- Understanding these patterns helps in resource planning and service optimization.

#### Conclusion:

Monitoring daily transaction volume provides valuable insights into customer behavior and operational efficiency.

Using this analysis, the bank can make data-driven decisions to improve services, optimize resource allocation, and enhance customer satisfaction.

```
-- jairam rai 7
133
        -- 7: Write a query to find the daily transaction volume (total amount of all transactions) for the past month.
134
        SELECT date(t.transaction_date) as transaction_date , round(SUM(t.amount),2) AS daily_volume
135 •
        FROM transactions t
136
137
        WHERE t.transaction_date >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH)
        GROUP BY date(t.transaction_date)
138
        order by date(t.transaction_date);
139
                                           Export: Wrap Cell Content: TA
Result Grid
                  Filter Rows:
                 daily_volume
   transaction date
  2024-06-11
                 7513.08
  2024-06-12
                 19277.54
  2024-06-14
                 14557.73
```

# 8.) Total Transaction Amount by Age Group in the Past Year

- Key Insights:
- Age Group 0-17:
  - Insights into the financial behavior of young customers.
- Age Group 18-30:
  - High transaction volume, reflecting active financial participation and spending habits.
- Age Group 31-60:
  - Peak earning and spending period, significant contribution to total transaction volume.
- Age Group 60+:
  - Reflects the spending patterns and financial activity of senior customers.

1031761.96

```
-- jairam rai 8
151
        -- 8: Calculate the total transaction amount performed by each age group in the past year. (Age groups: 0-17, 18-30, 31-60, 60+)
152
153 •
       SELECT
154
            CASE
                WHEN TIMESTAMPDIFF(YEAR, c.date_of_birth, CURDATE()) BETWEEN 0 AND 17 THEN '0-17'
155
                WHEN TIMESTAMPDIFF(YEAR, c.date_of_birth, CURDATE()) BETWEEN 18 AND 30 THEN '18-30
156
                WHEN TIMESTAMPDIFF(YEAR, c.date of birth, CURDATE()) BETWEEN 31 AND 60 THEN '31-60
157
                ELSE '60+'
158
159
            END AS age_group,
160
            round(SUM(t.amount),3) AS total_transaction_amount
161
        FROM customers as c
162
        INNER JOIN accounts a ON a.customer_id = c.customer_id
163
        INNER JOIN Transactions as t on a.account number = t.account number
                t.transaction_date >= DATE_SUB(CURDATE(), INTERVAL 1 YEAR)
164
165
        GROUP BY age group ORDER BY age group;
                                        Export: Wrap Cell Content: IA
   age_group | total_transaction_amount
  18-30
            520538.68
  31-60
            1166461.8
```

# 9.)Branch with the Highest Average Account Balance

- Key Insights:
- Top Branch:
  - Branch Name: [ICICI]
  - Average Account Balance: [6997.228]
  - This branch has the highest average account balance, indicating a strong financial position and high-value accounts.
- Branch Performance:
  - Compared average account balances across all branches.
  - Identified the branch with the highest average, reflecting superior account management and customer wealth.

#### **Customer Value:**

- Branches with high average account balances often serve high-net-worth individuals.
- Indicates successful relationship management and customer retention strategies.

```
-- Jairam rai
190
        -- 9: Find the branch with the highest average account balance.
191
        SELECT a.branch_id, AVG(a.balance) AS avg_balance
192 •
         FROM accounts as a
193
        JOIN branches as b ON a.branch id = b.branch id
194
        GROUP BY a.branch_id
195
        ORDER BY avg_balance DESC LIMIT 1;
196
                                         Export: Wrap Cell Content: 🛱 Fetch rows:
Result Grid Filter Rows:
   branch_id avg_balance
            6997.228
```

## 10.) Analysis of Average Balance per Customer at the End of Each Month in the Last Year

- Key Insights:
- Monthly Average Balance:
  - Calculated the average balance per customer for each month in the last year.
  - This metric provides insights into the financial health and behavior of customers on a monthly basis.
- Trends and Patterns:
  - Observed trends in average balances over the months.
  - Identified periods with significant increases or decreases in average balances.

#### Conclusion:

- Monitoring the average balance per customer at the end of each month provides valuable insights into customer financial behavior.
- These insights can help in designing targeted financial products, improving customer engagement, and ensuring the financial health of customers.

```
-- Jairam rai
212
        -- 10: Calculate the average balance per customer at the end of each month in the last year.
213
        select c.customer_id, c.first_name, c.last_name,
        date format(a.created at, '%Y-%m') as month end,
215
        round(avg(a.balance),2) as avg balance from customers as c
216
217
        inner join accounts as a
        on c.customer id = a.customer id
218
        where a.created at >= date sub(curdate(), interval 1 year)
219
         group by c.customer id, c.first name, c.last name, month end order by c.customer id;
220
                                          Export: Wrap Cell Content: $\overline{A}$
Result Grid
              Filter Rows:
   customer id
              first name last name
                                  month_end avg_balance
                                  2024-04
                                             1141.6
                                  2024-02
                                             1451.93
              Tanya
                                  2023-10
                                             6097.51
```