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.

```sql
-- Jairam_Rai
-- 1: Write a query to list all customers who haven't made any transactions in the last year.
-- How can we make them active again? Provide appropriate region.
SELECT c.customer_id, c.first_name, c.last_name
FROM customers c
inner JOIN accounts a ON c.customer_id = a.customer_id
inner JOIN transactions t ON a.account_number = t.account_number
WHERE t.transaction_date IS NULL OR t.transaction_date < DATE_SUB(CURDATE(), INTERVAL 1 YEAR)
GROUP BY c.customer_id, c.first_name, c.last_name
order by c.customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name
1	Raghav	Warrior
2	Jhanvi	Khalsa
3	Manikya	Sharma
5	Ojas	Sawhney
6	Aarna	Buch
7	Advika	Som
8	Nayantara	Dasgupta

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.

```
26      -- jairam rai
27      -- 2: Summarize the total transaction amount per account per month.
28 •    SELECT t.account_number, YEAR(t.transaction_date) AS year, MONTH(t.transaction_date) AS month, SUM(t.amount) AS total_amount
29      FROM transactions t
30      GROUP BY t.account_number, YEAR(t.transaction_date), MONTH(t.transaction_date)
31      order by YEAR(t.transaction_date), MONTH(t.transaction_date);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

account_number	year	month	total_amount
1016273546	2022	7	3676.69
1162555129	2022	7	2845.17
1174141558	2022	7	1183.43
1126038891	2022	7	212.84
1128643865	2022	7	3062.52
1054361923	2022	7	3779.09
1117992208	2022	7	1437.73

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.

```sql
-- jairam_rai
-- 3 Rank branches based on the total amount of deposits made in the last quarter.
SELECT b.branch_id, ROUND(SUM(CASE WHEN t.transaction_type = 'deposit' THEN t.amount ELSE 0 END),2) AS totaldeposit,
DENSE_RANK() OVER (ORDER BY SUM(CASE WHEN t.transaction_type = 'deposit' THEN t.amount ELSE 0 END)DESC) AS branch_rank
FROM Transactions AS t
JOIN Accounts AS a
ON t.account_number=a.account_number
JOIN branches AS b
ON b.branch_id=a.branch_id
WHERE t.transaction_date>= DATE_SUB(CURDATE(), INTERVAL 3 MONTH)
GROUP BY b.branch_id
ORDER BY branch_rank asc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

branch_id	totaldeposit	branch_rank
27	23688.58	1
14	18448.58	2
29	16348.46	3
32	16109.1	4
17	12548.97	5
2	11969.37	6

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
75    -- 4: Find the name of the customer who has deposited the highest amount.
76    select c.customer_id ,c.First_name , c.Last_name , sum(t.amount) as amount
77    from customers as c join
78    accounts as a on
79    c.customer_id = a.customer_id
80    join transactions as T on
81    t.account_number = a.account_number
82    where Transaction_type = "Deposit"
83    group by c.customer_id,c.First_name , c.Last_name
84    order by sum(t.amount) desc limit 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

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.

```
95      -- Jairam rai
96      -- 5: Identify any accounts that have made more than two transactions in a single day, which could indicate fraudulent activity. How
97      -- can you verify any fraudulent transaction?
98  •   select account_number , day(transaction_date) as day_of_transaction , count(transaction_id) as transaction_count
99      from transactions
100     group by account_number ,  day(transaction_date)
101     having transaction_count > 2
102     order by day(transaction_date);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

account_number	day_of_transaction	transaction_count
1030596655	1	3
1135708497	2	3
1032168449	3	3
1048635902	3	3
1128643865	3	4

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.

```sql
-- Jairam_rai_6
-- 6: Calculate the average number of transactions per customer per account per month over the last year.
SELECT a.customer_id, a.account_number, YEAR(t.transaction_date) AS transaction_year,
MONTH(t.transaction_date) AS transaction_month,
    COUNT(t.transaction_id) AS transaction_count,
    COUNT(t.transaction_id) / 12 AS avg_transactions_per_month_last_year
FROM transactions t
JOIN accounts a ON t.account_number = a.account_number
WHERE t.transaction_date >= DATE_SUB(CURDATE(), INTERVAL 1 YEAR)
GROUP BY a.customer_id, a.account_number, transaction_year, transaction_month
ORDER BY a.customer_id, a.account_number, transaction_year, transaction_month;
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	account_number	transaction_year	transaction_month	transaction_count	avg_transactions_per_month_last_year
1	1032168449	2023	8	1	0.0833
1	1032168449	2023	9	1	0.0833
1	1032168449	2023	10	2	0.1667
1	1032168449	2024	2	1	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.

```
133     -- jairam rai 7
134     -- 7: Write a query to find the daily transaction volume (total amount of all transactions) for the past month.
135 •   SELECT date(t.transaction_date) as transaction_date , round(SUM(t.amount),2) AS daily_volume
136     FROM transactions t
137     WHERE t.transaction_date >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH)
138     GROUP BY date(t.transaction_date)
139     order by date(t.transaction_date);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

transaction_date	daily_volume
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.

```
151     -- jairam rai 8
152     -- 8: Calculate the total transaction amount performed by each age group in the past year. (Age groups: 0-17, 18-30, 31-60, 60+)
153     SELECT
154         CASE
155             WHEN TIMESTAMPDIFF(YEAR, c.date_of_birth, CURDATE()) BETWEEN 0 AND 17 THEN '0-17'
156             WHEN TIMESTAMPDIFF(YEAR, c.date_of_birth, CURDATE()) BETWEEN 18 AND 30 THEN '18-30'
157             WHEN TIMESTAMPDIFF(YEAR, c.date_of_birth, CURDATE()) BETWEEN 31 AND 60 THEN '31-60'
158             ELSE '60+'
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
164     WHERE   t.transaction_date >= DATE_SUB(CURDATE(), INTERVAL 1 YEAR)
165     GROUP BY  age_group ORDER BY   age_group;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

age_group	total_transaction_amount
18-30	520538.68
31-60	1166461.8
60+	1031761.96

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.

```
190     -- Jairam_rai
191     -- 9: Find the branch with the highest average account balance.
192 •   SELECT a.branch_id, AVG(a.balance) AS avg_balance
193     FROM accounts as a
194     JOIN branches as b ON a.branch_id = b.branch_id
195     GROUP BY a.branch_id
196     ORDER BY avg_balance DESC LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

branch_id	avg_balance
2	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.

```
212     -- Jairam_rai
213     -- 10: Calculate the average balance per customer at the end of each month in the last year.
214     select c.customer_id, c.first_name, c.last_name,
215     date_format(a.created_at, '%Y-%m') as month_end,
216     round(avg(a.balance),2) as avg_balance from customers as c
217     inner join accounts as a
218     on c.customer_id = a.customer_id
219     where a.created_at >= date_sub(curdate(), interval 1 year)
220     group by c.customer_id,c.first_name, c.last_name, month_end order by c.customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name	month_end	avg_balance
5	Ojas	Sawhney	2024-04	1141.6
8	Nayantara	Dasgupta	2024-02	1451.93
10	Tanya	Keer	2023-10	6097.51