📘 SQL 100 Days Challenge – Day 64 Reflection

📅 **Date: Day 64**

📁 **Dataset Theme: Banking Transactions & Fraud Detection Analysis**

✅ Overview

Today's dataset simulated a **banking environment** involving customers, accounts, and transaction histories. The goal was to analyze deposits, debits, balances, and potential fraud patterns.

The session offered a good balance of **fundamental SQL logic** and **fraud-style analytics** using **window functions, CTEs, and conditional aggregations**. Most queries were straightforward — except the 10th one, which was **conceptually challenging** but rewarding once solved.

🔑 Key Learnings & Query Reflections

1. **Basic Join** – Listed customers with their current balances.
   → Easy and smooth start to the session.
2. **Aggregation with CASE** – Calculated total credits, debits, and net amount per account.
   → Implemented cleanly using conditional aggregation.
3. **Date Filtering** – Extracted March 2022 transactions using date range logic.
   → Reinforced comfort with date functions and filtering.
4. **CASE Statement Logic** – Categorized accounts as *High Balance* or *Low Balance*.
   → Simple but practical business case usage.
5. **Subquery Practice** – Identified customers with the highest balance using both direct and subquery methods.
   → Quick and efficient.
6. **Window Function (ROW_NUMBER)** – Retrieved latest transaction per account.
   → Straightforward; growing confidence with partition logic.
7. **Window Function (RANK)** – Ranked customers by total credited amount.
   → Easy to interpret and implement.
8. **Running Balance (CTE)** – Generated a dynamic account balance using window aggregation.
   → Excellent reinforcement of analytical CTEs.
9. **Advanced Aggregation** – Calculated average debit per transaction per customer.
   → Easy to build and verify with manual cross-check logic.
10. **Fraud-Style Query (Challenging)** –

- Detected accounts with more than two transactions above ₹50,000 within 60 days.
- Used **self-joins + CTEs + conditional filtering**.
- Conceptually tough but provided deep insights into **pattern detection using SQL**.

11. **Bonus Challenge** – Found the customer with the highest share of total bank credits.
    → Reinforced skills in percentage contribution and aggregate logic.

🌟 Reflection & Takeaways

- Except for the **10th question**, the entire session felt **comfortable and well-paced**.
- Fraud-style queries continue to build **advanced problem-solving skills** — learning how to simulate real-world analytics inside SQL.
- Comfort with **CTEs, ranking, and date logic** is becoming second nature now.
- Day 64 was a great reminder that even simple datasets can offer deep analytical insight when explored creatively.

💭 Final Thought

“Consistency builds clarity — and clarity builds confidence.”
Every day’s query, whether easy or tough, is a brick in the foundation of SQL mastery.