📘 SQL 100 Days Challenge – Day 67 Reflection

📅 Dataset Theme: Banking Transactions, Loans & Customer Insights

Today's challenge brought together multiple concepts from real-world **banking systems**, integrating data from **customers, accounts, transactions, and loans** — a perfect mix of business logic and analytics.

⚙️ Concepts Practiced

- **CTEs & Nested CTEs** for multi-step logic building

- **Correlated Subqueries** for comparisons across countries

- **LAG()**, **RANK()**, and **CASE WHEN** for analytical classifications

- **Conditional Aggregations** using SUM(CASE WHEN...)

- **JOIN + Date Logic** to compare loan and transaction timelines

- **Complex multi-CTE queries** for monthly streaks and loyalty scoring

💡 Key Learnings

1. Functions like **LAG()** and **RANK()** now come naturally — logical flow feels intuitive.

2. Understood how to combine **multiple analytical layers (CTE + Window + CASE)** for deep financial insights.

3. Improved at managing **dataset granularity** — ensuring customer-level vs transaction-level logic is accurate.

4. Developed better clarity on **correlated subqueries**, especially for comparing averages and balances.

5. Bonus question taught the value of **breaking complex problems** into smaller, logical CTE steps.

⚔️ Challenges Faced

- Questions **6, 7, 9, 10, and especially the Bonus (11)** were tough and required multiple re-runs.

- The **bonus question** (finding top loyal customers through consecutive month tracking) was the hardest so far, testing both logic and patience.

- Handling **date-based consecutive streaks** was complex — a great learning on practical SQL analytics.

🌱 Reflection

"Day 67 reminded me how SQL can be both logical and artistic — building insights layer by layer. It wasn't easy, but it felt rewarding to finally get the bonus query right!"

❄️ Next Steps

- Keep refining complex analytical thinking.

- Practice more **multi-CTE query structures**.

- Begin focusing on **query optimization and performance tuning** for large datasets.