SQL 100 Days Challenge – Day 48 Reflection

Topic: Banking Analytics – Customers, Accounts & Transactions

Dataset: Customers, Accounts, Transactions

Practice Experience:

- Today's practice was **tough**. Questions **1–6** were manageable, focusing on balances, high transaction customers, channel performance, top spenders, fraud detection, and CLV.
- **Questions 7–11** were much more advanced — running balances with window functions, inactive customer detection with CTEs, recursive CTEs for risk propagation, and multi-metric country-wise insights. These took a lot of time to think through and implement.
- The **Bonus Challenge (Advanced Fraud Detection – Suspicious Patterns)** was especially challenging, combining LAG(), DATEDIFF(), filtering, and sequence aggregation to detect back-to-back high-value debits.

Key Learnings:

1. **Customer & Balance Insights:** Using CASE WHEN with SUM for credits vs debits.
2. **Channel Analytics:** Success vs failure percentages by channel.
3. **Fraud Detection:** Identifying repeated failures and suspicious debit sequences.
4. **CLV:** Normalizing lifetime value with tenure in years.
5. **Window Functions:** Implementing running balances and back-to-back transaction detection.
6. **Recursive CTEs:** Propagating risk across accounts within the same customer.
7. **Country-Wise Insights:** Aggregating multiple KPIs (net flow, average value, total credits/debits).

Insights:

- Online transactions showed more failures compared to branch transactions.
- Some accounts triggered **high-risk flags** due to repeated failed attempts.
- Running balances provided a **clear transaction trail** per account.
- Country-level insights revealed both transaction behavior and net flows.
- Fraud detection logic flagged accounts with **suspicious debit sequences** that could mimic fraud in real banking systems.

Skills Reinforced:

- Complex CASE WHEN calculations for metrics.
- Window functions (LAG, SUM OVER) for balances and fraud patterns.
- Recursive CTEs for hierarchical/risk propagation logic.
- Percentage calculations and multi-KPI country aggregations.
- Advanced fraud detection with pattern recognition in SQL.

Personal Note:
Today's session really tested my **patience and persistence**. Questions 7–11 and the bonus were difficult and time-consuming, but finishing them felt extremely rewarding. It's proof that I'm now able to handle **real-world banking analytics problems in SQL** — something that looked impossible earlier in this challenge.

Next Steps:

- Extend fraud detection to cover cross-channel behaviors.
- Build customer inactivity prediction logic.
- Automate CLV segmentation (high, medium, low).