Reflection Report – Day 16 SQL Practice

🗓️ Date: 26 July 2025

📚 Topic: SQL Practice – Banking Transactions Dataset

🧠 Reflection Summary

Day 16 focused on digital banking data involving Customers, Accounts, and Transactions. The queries covered deposit/withdrawal tracking, customer-account linking, and net flow calculations. This session not only tested SQL proficiency but also schema design and performance considerations.

🛠️ Initial Table Design & Constraints

✅ Customers Table

- CustomerID as primary key – good
- Name as NOT NULL – appropriate
- Age with CHECK (Age>0) – prevents invalid entries
- Country as NOT NULL – ensures completeness

✅ Accounts Table

- AccountID as primary key – good
- CustomerID as NOT NULL + foreign key – proper relationship
- AccountType as NOT NULL – good
- Balance with CHECK (Balance >= 0) – prevents negative values
- ON DELETE CASCADE – ensures dependent cleanup

✅ Transactions Table

- TransactionID as primary key – good
- AccountID with foreign key – proper relationship
- Amount with CHECK (Amount > 0) – correct
- TransactionType with value restriction – works as ENUM
- TransactionDate with CHECK (<= GETDATE()) – prevents future dates
- ON DELETE CASCADE – ensures referential integrity

✅ Indexes

Created indexes on frequently joined columns (CustomerID, AccountID) to ensure query optimization.

🔧 Improvements Made Later

1. **Data Types**

- Switched from CHAR to VARCHAR for names/countries
- Considered increasing DECIMAL precision for large balances

2. **Additional Constraints**

- Added upper limit for Age (< 120)
- Added default for TransactionDate (GETDATE())
- Considered future enhancement: prevent withdrawals over balance

3. **Other Considerations**

- Plan to add timestamp for time-based analytics
- Added `Status` column in Accounts (Active/Inactive/Closed)

☑ Key Learnings & Concepts Practiced

- Used `ISNULL()` in Query 1 for cleaner NULL handling
- Filtered condition directly in JOIN (Query 3) – cleaner logic
- Learned use of `WITH TIES` for ranking tied rows (Query 5)
- Query 7 was tricky – gained more understanding of date filtering
- Query 9 taught new approach using `CASE` for conditional aggregation
- Bonus was logically dense – but satisfying to crack

💥 Final Thoughts

This session wasn't just about writing queries—it emphasized thoughtful database design, indexing, and real-world logic. With each day, I'm getting faster, more accurate, and more confident. Day 16 was another step up in applying layered logic in SQL.