


Day 19 – SQL Practice Reflection

 Date: July 30, 2025

 Dataset Theme: Hospital Management – Patients, Doctors, Appointments, and Treatments

Summary

Today's SQL challenge simulated a hospital system environment with appointments, treatments, and doctors' performance. It focused on real-world scenarios such as patients with multiple consultations, specialization-based lookups, and identifying untreated or unconsulted entities.

Key Learnings

1. Initially used `CHECK(Date >= GETDATE())` constraint on AppointmentDate but realized it's too restrictive for hospital systems.
2. Learnt to apply ``ON UPDATE CASCADE`` in foreign keys to ensure integrity on ID changes.
3. Observed that ``COALESCE`` is unnecessary when using ``COUNT()`` since it returns 0 for unmatched groups.
4. Understood that ``INNER JOIN`` is more appropriate than ``LEFT JOIN`` for aggregating actual appointment data.
5. Solved all queries in much less time than previous days — a clear sign of improved fluency and reasoning.

SQL Concepts Practiced

- `COUNT` with `GROUP BY` and `DISTINCT` combinations
- Aggregate filtering using `HAVING`
- `LEFT JOIN` for null-checking scenarios
- Use of ``NOT EXISTS`` and ``INNER JOIN`` for performance-sensitive logic
- Filtering by doctor experience and patient age

Bonus & Observations

- ✓ Bonus question taught practical use of `LEFT JOIN` with `NULL` filters to find unconsulted doctors.
- ✓ Logic behind Q3 (highest average consultation fee) reinforced the importance of using `INNER JOIN`.
- ✓ Q4 required using `DISTINCT` doctor counts per patient — great for understanding grouping depth.