Day 5 SQL Practice – Reflection on Students, Courses & Enrollments Dataset

Today's session was a significant step forward in my SQL learning journey. Working with the **Students, Courses, and Enrollments** dataset, I tackled complex queries involving **multi-table joins, filtering, window functions, and revenue calculations**. While some challenges required deeper thinking, the progress in understanding key SQL concepts was rewarding.

🔍 Reflections & Query Breakdown

1. List all students with their enrolled courses

A straightforward **JOIN** between Students and Enrollments helped reinforce the basics of table relationships.

2. Average score per course

Used **GROUP BY** and **AVG()**, solidifying my understanding of aggregation functions.

3. Students enrolled in multiple courses

Learned the crucial difference between **COUNT()** (total records) and **COUNT(DISTINCT)** (unique values) when identifying duplicate enrollments.

4. Highest score per course category

Applied **GROUP BY** with **MAX()**, improving my ability to segment data by categories.

5. Students scoring above 80 in any course

Used a simple **WHERE** filter, reinforcing conditional querying.

6. Total revenue per course category

Discovered the key distinction between **potential revenue** (sum of all course fees) and **actual earned revenue** (sum of fees only for enrolled courses). This was an eye-opener for real-world data analysis.

7. Days since student enrollment

Practiced **date functions (DATEDIFF, GETDATE)**, but realized I need more practice with date manipulations.

8. Courses not taken by any student

Mastered the **LEFT JOIN + IS NULL** technique to identify unmatched records, a useful pattern for gap analysis.

9. Rank students by score within each course

Successfully implemented **RANK() and DENSE_RANK()**, understanding their differences in handling ties.

10. Total enrollments and average score by city

Combined **COUNT()** and **AVG()** with **GROUP BY**, improving my ability to summarize data geographically.

Bonus: Top 2 scorers per course category

Used **DENSE_RANK() in a CTE (Common Table Expression)**, a powerful technique for advanced ranking scenarios.

💡 Key Takeaways from Day 5

💡 Core Learnings & Improvements

1. **Aggregation Clarity**
   - Mastered the difference between **COUNT()** (total records) vs. **COUNT(DISTINCT)** (unique values), crucial for analyzing enrollments (Q3).
   - Distinguished **potential revenue** (sum of all fees) from **actual earned revenue** (fees from enrolled courses only) in revenue calculations (Q6).
2. **Advanced Query Techniques**
   - Successfully implemented **RANK()** and **DENSE_RANK()**, understanding their tie-handling differences (Q9, Bonus).
   - Used **CTEs (WITH clauses)** to simplify complex queries like ranking top scorers (Bonus).
3. **JOIN Logic & Optimization**
   - Applied **LEFT JOIN + IS NULL** to find gaps (e.g., untaken courses in Q8) vs. **INNER JOIN + WHERE** for exact matches.
   - Improved speed by reducing trial-and-error through better pattern recognition (all queries).
4. **Areas for Growth**
   - **Date functions (DATEDIFF, GETDATE)** need more practice (Q7).
   - Reinforced the importance of **breaking down problems** before coding (e.g., Bonus question).

🚀 Moving Forward

Today reinforced that **struggle leads to growth**. The more I practice, the more intuitive SQL becomes. Next steps:
✓ **Drill date functions** for better time-based analysis.
✓ **Experiment with more CTEs** to simplify complex queries.
✓ **Revisit revenue calculations** to ensure accurate business insights.

Overall, a day of tangible progress! 🎯