

```
# Reflection - Lab 02: SQL Fundamentals
```

```
**Student:** Maryam  
**Date:** 2026-02-25
```

```
---
```

## ## What Surprised Me About How SQL Works

The most surprising thing I learned was that SQL clauses do not execute in the order we write them. I always assumed the database reads queries from top to bottom, but learning that WHERE runs before SELECT explained so many errors I've made. The fact that you cannot use a column alias in WHERE because it "doesn't exist yet" was a revelation - it's not a bug, it's just how SQL is designed.

Another surprise was how NULL works. I expected `= NULL` to either work or throw an error, but it silently returns zero rows. This is dangerous because you might think your query is correct when it's actually giving wrong results. The distinction that NULL is "absence of a value" rather than a value itself is fundamental but not intuitive.

## ## What Clicked

The CASE WHEN statement finally made sense when I thought of it as SQL's version of if/else. I've used conditional logic in Python, but couldn't figure out how to do the same in SQL. Now I understand that CASE WHEN is how you categorize data, create priority levels, or handle different business rules directly in your queries.

The EXPLAIN ANALYZE output was also illuminating. Seeing "Seq Scan" and understanding that it means reading every row helped me understand why indexes matter. On our small 30-row table it took 0.044ms, but I can now reason about what would happen with millions of rows.

## ## What Is Still Confusing

I'm still not entirely clear on when PostgreSQL will use an Index Scan versus a Seq Scan. The manual mentions that LIKE patterns starting with % cannot use indexes, but I don't fully understand what other factors influence this decision. I also want to understand more about the "top-N heapsort" optimization I saw in the EXPLAIN output.

## ## What I Want to Understand Better Before Week 3

I want to practice more with JOINS since we only did basic single-table queries. The order\_items table references orders and products, but I haven't yet written queries that combine them. I also want to understand aggregate functions (COUNT, SUM, AVG) with GROUP BY better, as these seem essential for data analysis but we didn't cover them deeply in this lab.

```
---
```

```
**Word Count:** ~280 words
```