Lab 02 Deliverable - SQL Fundamentals with Real Datasets

Student Name: Maryam
Roll Number: [Your Roll Number]
Course: Database Systems Lab
Instructor: Muhammad Usama Afridi
Date Submitted: 2026-02-25
GitHub Repo: https://github.com/maryam/database-labs

Section 1 Performance Report (20 points)

EXPLAIN ANALYZE Output for Query 5

```sql
EXPLAIN ANALYZE
SELECT order_id, customer_id, total_amount, status
FROM orders
ORDER BY total_amount DESC
LIMIT 10;
```

Full Output:
```
Limit  (cost=28.65..28.67 rows=10 width=102) (actual 
time=0.026..0.028 rows=10 loops=1)
  ->  Sort  (cost=28.65..30.12 rows=590 width=102) (actual 
time=0.026..0.027 rows=10 loops=1)
        Sort Key: total_amount DESC
        Sort Method: top-N heapsort  Memory: 26kB
        ->  Seq Scan on orders  (cost=0.00..15.90 rows=590 
width=102) (actual time=0.004..0.008 rows=30 loops=1)
Planning Time: 0.026 ms
Execution Time: 0.044 ms
```

Analysis

Scan Type: Seq Scan (Sequential Scan)

The database performed a **Seq Scan**, meaning it read every row
in the orders table (30 rows). This is expected because:
1. There is no index on the `total_amount` column
2. The table is very small (30 rows), so a full scan is actually

efficient

Execution Time: 0.044 milliseconds

If the table had 5 million rows:
- The Seq Scan would become extremely slow (potentially several seconds or minutes)
- PostgreSQL would likely still use Seq Scan for ORDER BY without LIMIT
- With LIMIT 10, it might use a "top-N heapsort" optimization (as shown) to avoid sorting all rows
- An index on `total amount` would dramatically improve performance, potentially reducing query time from minutes to milliseconds
- The "Memory: 26kB" for heapsort would increase significantly

Key Insight: On small tables, Seq Scan is fine. On large tables, proper indexing is critical for performance.

Section 3 Reflection (10 points)

See attached file: `reflection.md`

Key Points:
- **Surprised by:** SQL execution order (WHERE before SELECT) and NULL behavior
- **What clicked:** CASE WHEN as SQL's if/else, EXPLAIN ANALYZE interpretation
- **Still confusing:** Index usage patterns, top-N heapsort optimization
- **Want to learn:** JOINs, aggregate functions with GROUP BY

Word Count: ~280 words (exceeds 150-word minimum)

Submission Checklist

- [x] ecommerce setup.sql committed to GitHub
- [x] queries.sql committed to GitHub
- [x] All 10 queries with screenshots and explanations
- [x] EXPLAIN ANALYZE output with interpretation
- [x] AI Learning Log with 3 genuine entries

- [x] Reflection (150+ words)
- [x] File named: Lab02_Maryam_[RollNumber].pdf
- [x] GitHub repo URL included

GitHub Repository: https://github.com/maryam/database-labs

Files Submitted:
1. `lab2/ecommerce_setup.sql` - Database schema and seed data
2. `lab2/queries.sql` - All 10 SQL queries
3. `lab2/ai_learning_log.md` - AI interaction documentation
4. `lab2/reflection.md` - Personal reflection
5. `lab2/Lab02_Maryam_[RollNumber].pdf` - This deliverable