**Lab Exercise 10- Optimizing Queries Using EXPLAIN ANALYZE**

**Objective**

Learn how to:

* Use EXPLAIN ANALYZE to inspect query execution plans
* Identify performance bottlenecks
* Apply indexing and query rewrites to improve performance

**Prerequisites**

* PostgreSQL installed on Windows
* Access to SQL Shell (psql) or pgAdmin
* Basic understanding of SQL

**Step 1: Setup Sample Data**

CREATE TABLE employees (

id SERIAL PRIMARY KEY,

name TEXT,

department TEXT,

salary NUMERIC,

hire\_date DATE

);

-- Insert 100000 sample records

INSERT INTO employees (name, department, salary, hire\_date)

SELECT

'Employee\_' || i,

CASE WHEN i % 4 = 0 THEN 'HR'

WHEN i % 4 = 1 THEN 'IT'

WHEN i % 4 = 2 THEN 'Finance'

ELSE 'Marketing' END,

ROUND(random() \* 90000 + 10000, 2),

CURRENT\_DATE - (i % 365)

FROM generate\_series(1, 100000) AS s(i);

**Step 2: Run a Slow Query Without Index**

EXPLAIN ANALYZE

SELECT \* FROM employees WHERE name = 'Employee\_9999';

**Observe:**

* Whether a **sequential scan** is used
* Total execution time

**Step 3: Add Index to Improve Performance**

CREATE INDEX idx\_employees\_name ON employees(name);

Then run the same query again:

EXPLAIN ANALYZE

SELECT \* FROM employees WHERE name = 'Employee\_9999';

**Observe:**

* PostgreSQL should now use an **index scan**
* Execution time should be reduced

**Step 4: Compare Queries with and Without Indexes**

Try this query:

EXPLAIN ANALYZE

SELECT \* FROM employees WHERE department = 'HR';

Then add an index:

CREATE INDEX idx\_department ON employees(department);

And run the query again. Compare execution plans.

**Step 5: Optimize a Range Query**

Without index:

EXPLAIN ANALYZE

SELECT \* FROM employees WHERE salary > 90000;

Add index on salary:

CREATE INDEX idx\_salary ON employees(salary);

Re-run the query and compare performance.

**Step 6: Optimize Date-Based Filtering**

EXPLAIN ANALYZE

SELECT \* FROM employees WHERE hire\_date > CURRENT\_DATE - INTERVAL '30 days';

Add index:

CREATE INDEX idx\_hire\_date ON employees(hire\_date);

Run again and compare the execution plan.

**Step 7: Understanding Output**

Focus on these key metrics:

* **Seq Scan vs Index Scan**: Seq Scan means no index is used.
* **Rows**: Number of rows processed.
* **Execution Time**: Total time to run the query.
* **Actual vs Estimated**: Helps identify if statistics are outdated.

**Optional: Force Sequential Scan**

SET enable\_indexscan = OFF;

EXPLAIN ANALYZE SELECT \* FROM employees WHERE name = 'Employee\_9999';

**Summary**

| **Optimization** | **Benefit** |
| --- | --- |
| Adding Index | Reduces scan time |
| Filtering Efficiently | Helps avoid full table scans |
| EXPLAIN ANALYZE | Diagnoses performance issues |