**CBO with Example**

**Cost-Based Optimizer (CBO)** in action using CREATE, INSERT, and SELECT statements.

**Objective**

Create a table, insert data, and run a query to see how PostgreSQL chooses the **most cost-efficient** plan using EXPLAIN ANALYZE.

**Step-by-Step Example**

**1. Create the Tables**

-- Departments table

CREATE TABLE departments (

id SERIAL PRIMARY KEY,

name TEXT

);

-- Employees table

CREATE TABLE employees (

id SERIAL PRIMARY KEY,

name TEXT,

department\_id INT,

salary NUMERIC,

FOREIGN KEY (department\_id) REFERENCES departments(id)

);

-- Create an index on department\_id to help the optimizer

CREATE INDEX idx\_employees\_department\_id ON employees(department\_id);

**2. Insert Sample Data**

-- Insert departments

INSERT INTO departments (name)

SELECT 'Department ' || g

FROM generate\_series(1, 10) AS g;

-- Insert 100,000 employees randomly distributed across departments

INSERT INTO employees (name, department\_id, salary)

SELECT

'Employee ' || s,

(random() \* 9 + 1)::INT,

(30000 + random() \* 20000)::NUMERIC

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

**3. Run a SELECT Query and Use EXPLAIN ANALYZE**

Let’s query employees in a specific department and check what execution plan the optimizer picks.

EXPLAIN ANALYZE

SELECT \*

FROM employees

WHERE department\_id = 5;

**Sample Output:**

Index Scan using idx\_employees\_department\_id on employees

(cost=0.29..356.54 rows=9959 width=48)

Index Cond: (department\_id = 5)

...

Planning Time: 0.123 ms

Execution Time: 18.456 ms

**4. Force a Sequential Scan (for comparison)**

Temporarily disable index scans to see how the cost changes:

SET enable\_indexscan = OFF;

EXPLAIN ANALYZE

SELECT \*

FROM employees

WHERE department\_id = 5;

-- Reset it back after testing

RESET enable\_indexscan;

**Output (Simplified):**

Seq Scan on employees

(cost=0.00..2100.00 rows=10000 width=48)

Filter: (department\_id = 5)

...

Execution Time: 89.032 ms

As you can see, the CBO originally picked **Index Scan** because it had a much lower cost and faster execution time.

**Key Takeaways**

| **Step** | **What Happens** |
| --- | --- |
| CREATE INDEX | Helps CBO choose Index Scan when efficient |
| EXPLAIN ANALYZE | Reveals the chosen plan and actual timing |
| SET enable\_indexscan = OFF | Shows what happens without the index |