**🧠 Top-Level Node: Gather**

**What it is:**

* Gather is a node that **combines results** from multiple **parallel workers**.
* It's the root of the plan tree when PostgreSQL uses **parallel query execution**.
* **Key Fields:**

| **Field** | **Meaning** |
| --- | --- |
| "Node Type": "Gather" | Collects rows from parallel workers. |
| "Parallel Aware": false | The gather node itself isn’t managing parallelism. Lower nodes do that. |
| "Startup Cost": 1000 | Estimated cost before returning the first row. |
| "Total Cost": 69277.33 | Estimated cost to return **all** rows. |
| "Plan Rows": 20000 | Estimated number of result rows. |
| "Plan Width": 1556 | Estimated row size in **bytes** (very large in this case). |
| "Workers Planned": 2 | PostgreSQL **planned** to use 2 parallel workers. |
| "Single Copy": false | Each worker has its own scan (not a shared scan). |

**🔁 Nested Node: Seq Scan (Sequential Scan)**

This is the **actual workhorse** scanning the employees table.

**Fields and Their Meaning:**

| **Field** | **Value** | **Explanation** |
| --- | --- | --- |
| "Node Type" | "Seq Scan" | A **sequential scan** of the whole employees table. |
| "Relation Name" | "employees" | Table being scanned. |
| "Alias" | "e1\_0" | Alias used (likely from ORM like Hibernate). |
| "Parallel Aware" | true | This scan is parallelized (each worker scans a slice). |
| "Startup Cost" | 0 | No cost to start scanning (common in Seq Scan). |
| "Total Cost" | 66277.33 | Estimated full cost for this part. |
| "Plan Rows" | 8333 | This is per-worker row estimate (2 workers planned → ~16666 total). |
| "Plan Width" | 1556 | Each row is estimated to be 1556 bytes — unusually large (maybe large TEXT/BLOB fields). |
| "Filter" | "first\_name = ?" | Rows are filtered where first\_name matches a parameter. |

**📊 What This Tells You**

**🔎 The Query is Costly**

* **High total cost**: 69,277.33 is large → suggests large table or poor filtering.
* **Sequential scan** on a likely large table → performance will degrade as table grows.

**⚠️ Filtering is Not Using Index**

* The filter first\_name = ? is **not using an index**.
* PostgreSQL is scanning the **entire table** and applying the filter after reading rows — very inefficient for large datasets.

**✅ Recommendations**

1. **Add an Index on first\_name**:

sql

CopyEdit

CREATE INDEX idx\_employees\_first\_name ON employees(first\_name);

This could change the plan from Seq Scan → Index Scan or Bitmap Index Scan.

1. **Check Row Width**
   * 1556 bytes per row is huge.
   * Investigate if employees has any large TEXT, JSON, BYTEA, or unused columns that can be trimmed.
2. **Partitioning (if data is huge)**
   * If employees is massive and first\_name has low cardinality, consider **table partitioning**.

**🧩 Visualizing the Plan**

yaml

CopyEdit

┌──────────────────────────────────────┐

│ Gather │

│ Total Cost: 69277.33, Rows: 20000 │

│ Workers Planned: 2 │

└──────────────────────────────────────┘

▲

│

▼

┌────────────────────────────────────────────┐

│ Seq Scan on employees │

│ Cost: 0 → 66277.33, Rows: 8333 (per core) │

│ Parallel Aware: true │

│ Filter: first\_name = ? │

└────────────────────────────────────────────

**🔍 What Is a PostgreSQL Execution Plan?**

When you run a SQL query in PostgreSQL, the database doesn’t just *"run it blindly."* Instead, it:

1. **Analyzes** the query.
2. **Chooses the fastest way** to execute it.
3. **Generates a plan** showing the steps it will take — this is called the **execution plan**

**1. Node Type 🧱**

This tells you *what operation* PostgreSQL is doing at that step.

| **Node Type** | **Meaning** |
| --- | --- |
| Seq Scan | Scans every row in the table (slow for large tables). |
| Index Scan | Uses an index to find specific rows (fast). |
| Bitmap Heap Scan | A hybrid method using an index to find blocks, then rows. |
| Nested Loop | For joining small tables. |
| Hash Join | For joining medium-to-large tables. |
| Aggregate | For GROUP BY, COUNT, etc. |
| Gather | Combines results from **parallel** workers. |

**2. Cost Estimates 💰**

Every node has two cost values:

**Startup Cost: 1000.0**

**Total Cost: 69277.33**

| **Cost Type** | **What it means** |
| --- | --- |
| Startup Cost | Cost to get the first row |
| Total Cost | Cost to get all rows |

These are **PostgreSQL's internal units**. They're not time, but *relative cost* used to compare plans.

🔧 Lower cost = better performance (usually).

**5. Parallelism ⚡**

If your query uses parallel workers:

* The plan shows a Gather node.
* It means PostgreSQL is splitting the work (e.g., scanning the table in parallel).
* Useful for big scans or aggregates.