Plan Raw Query Stats 1.5.0

Execution time: N/A | Planning time: N/A | Triggers: N/A | Settings

time	rows	estimation	c
#1 HashAggr...			
#2 └ Nested L...			
#3 ├ Seq Sc...			
#4 └ Memo...			
#5 └ Ind...			

#1 HashAggregate

by `departments.department_name`

HashAggregate Node groups records together based on a GROUP BY or aggregate function (like sum()). Hash Aggregate uses a hash to first organize the records by a key.

General | IO & Buffers | Output | Workers | Misc

- **Timing:** 0.284ms | 28%
- **Rows:** 11 (Planned: 200) | ↑ over estimated by 18 ×
- **Cost:** 5.7 (Total: 45.4)

#2 Nested Loop

Nested Loop Node merges two record sets by looping through every record in the first set and trying to find a match in the second set. All matching records are returned.

General | IO & Buffers | Output | Workers | Misc

- **Timing:** 0.454ms | 45%
- **Rows:** 741 (Planned: 741)
- **Cost:** 21.1 (Total: 39.7)

#3 Seq Scan

on `employees`

Seq Scan Node finds relevant records by sequentially scanning the input record set. When reading from a table, Seq Scans (unlike Index Scans) perform a single read operation (only the table is read).

General | IO & Buffers | Output | Workers | Misc

- **Timing:** 0.233ms | 23%
- **Rows:** 741 (Planned: 741)
- **Cost:** 18.4 (Total: 18.4)

#4 Memoize

Memoize Node is used to cache the results of the inner side of a nested loop. It avoids executing underlying nodes when the results for the current parameters are already in the cache.

General | IO & Buffers | Output | Workers | Misc

- **Timing:** 0ms | 0%
- **Rows:** ~741 (Planned: ~741)
- **Cost:** 0.01 (Total: 0.24)
- **Loops:** 741

#5 Index Scan

on `departments`
using `departments_pkey`

Index Scan Node finds relevant records based on an **Index**. Index Scans perform 2 read operations: one to read the index and another to read the actual value from the table.

General | IO & Buffers | Output | Workers | Misc

- **Timing:** 0.033ms | 3%
- **Rows:** ~11 (Planned: ~11)
- **Cost:** 0.23 (Total: 0.23)
- **Loops:** 11