Plan Raw Query Stats 1.5.0

Execution time: N/A | Planning time: 0.228ms | Triggers: N/A | Settings

time	rows	estimation	c

#1 HashAggr...
#2 └ Hash Join
#3 ├ Seq Sc...
#4 └ Hash
#5 └ Seq...

#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: 0.311ms | 33%
Rows: 11 (Planned: 11)
Cost: 3.81 (Total: 26.2)

#2 Hash Join

on `employees.department_id = departments.department_id`

Hash Join Node joins two record sets by hashing one of them (using a **Hash Scan**).

General | IO & Buffers | Output | Workers | Misc

Timing: 0.452ms | 48%
Rows: 741 (Planned: 741)
Cost: 2.83 (Total: 22.4)

#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: 0.169ms | 18%
Rows: 741 (Planned: 741)
Cost: 18.4 (Total: 18.4)

#4 Hash

Hash Node generates a hash table from the records in the input recordset. Hash is used by **Hash Join**.

General | IO & Buffers | Output | Workers | Misc

Timing: 0.007ms | 1%
Rows: 11 (Planned: 11)

#5 Seq Scan

on `departments`

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: 0.009ms | 1%
Rows: 11 (Planned: 11)
Cost: 1.11 (Total: 1.11)