Plan Raw Query Stats 1.5.0

Execution time: N/A | Planning time: 0.787ms | Triggers: N/A | Settings

time	rows	estimation	c

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

#1 HashAggregate

by locations.city

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: 0.011ms | 16%
Rows: 7 (Planned: 11)
Cost: 0.16 (Total: 14.8)

#2 Hash Join

on locations.location_id = departments.location_id

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

General | IO & Buffers | Output | Workers | Misc

Timing: 0.015ms | 22%
Rows: 11 (Planned: 11)
Cost: 1.15 (Total: 14.7)

#3 Seq Scan

on locations

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: 0.007ms | 10%
Rows: 7 (Planned: 240) | ↑ over estimated by **34** ×
Cost: **12.4** (Total: 12.4)

#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: 0.009ms | 13%
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: 0.026ms | 38%
Rows: 11 (Planned: 11)
Cost: 1.11 (Total: 1.11)