Plan Raw Query Stats 1.5.0

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

time	rows	estimation	c
#1 Sort			
#2 └ Nested L...			
#3 └ Hash J...			
#4 └ Has...			
#5 ├ S...			
#6 └ H...			

#1 Sort

by `jobs.job_title`

Sort Node sorts a record set based on the specified sort key.

General | IO & Buffers | Output | Workers | Misc

Timing: 0.558ms | 30%
Rows: 740 (Planned: 741)
Cost: 37.2 (Total: 86.7)

#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: 0.552ms | 29%
Rows: 740 (Planned: 741)
Cost: 24.4 (Total: 49.5)

#3 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: 0.243ms | 13%
Rows: 741 (Planned: 741)
Cost: 3.93 (Total: 25.1)

#4 Hash Join

on `employees.job_id = jobs.job_id`

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

General | IO & Buffers | Output | Workers | Misc

Timing: 0.422ms | 22%
Rows: 741 (Planned: 741)
Cost: 2.62 (Total: 21.2)

#5 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: 0.106ms | 6%
Rows: 741 (Planned: 741)
Cost: 17.4 (Total: 17.4)

#6 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: 0.008ms | 0%
Rows: 19 (Planned: 19)
Cost: 1.19 (Total: 1.19)