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 └ Nes...			
#5 ├ S...			
#6 └ ...			

#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: 0.544ms | 30%
Rows: 740 (Planned: 741)
Cost: 37.2 (Total: 107)

#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: 0.443ms | 24%
Rows: 740 (Planned: 741)
Cost: 24.5 (Total: 70)

#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: 0.258ms | 14%
Rows: 741 (Planned: 741)
Cost: 3.94 (Total: 45.5)

#4 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: 0.507ms | 28%
Rows: 741 (Planned: 741)
Cost: 22.9 (Total: 41.5)

#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: 0.089ms | 5%
Rows: 741 (Planned: 741)
Cost: 18.4 (Total: 18.4)

#6 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: 0.24 (Total: 0.24)
Loops: 741