

time	rows	estimation	cost
#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: 0.524ms | 32%

☰ Rows: 740 (Planned: 741)

\$ Cost: 37.2 (Total: 87.8)

#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: 0.416ms | 26%

☰ Rows: 740 (Planned: 741)

\$ Cost: 24.5 (Total: 50.6)

#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: 0.275ms | 17%

☰ Rows: 741 (Planned: 741)

\$ Cost: 3.93 (Total: 26.1)

#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: 0.298ms | 18%

☰ Rows: 741 (Planned: 741)

\$ Cost: 2.62 (Total: 22.2)

#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: 0.102ms | 6%

☰ Rows: 741 (Planned: 741)

\$ Cost: 18.4 (Total: 18.4)

#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: 0.01ms | 1%

☰ Rows: 19 (Planned: 19)

\$ Cost: 1.19 (Total: 1.19)