

Resolvemos colocar aqui pra ficar melhor de entender o que fizemos no relatório, caso não seja possível esse conserto no trabalho, pedimos que ignore o restante do pdf.

Todas as explicações fora daqui constam no relatório.

3) Não otimizado, no relatório mencionamos a window function e o uso de disco do sort.

	QUERY PLAN text	
1	Subquery Scan on ranked_employees (cost=133443.34..165943.34 rows=5000 width=15) (actual time=1251.479..2352.934 rows=33 lo...	
2	Filter: (ranked_employees.rank_salario = 1)	
3	Rows Removed by Filter: 999967	
4	-> WindowAgg (cost=133443.34..153443.34 rows=1000000 width=23) (actual time=1251.472..2264.312 rows=1000000 loops=1)	
5	-> Sort (cost=133443.34..135943.34 rows=1000000 width=15) (actual time=1251.209..1526.307 rows=1000000 loops=1)	
6	Sort Key: empregados.dep_id, empregados.salario DESC	
7	Sort Method: external merge Disk: 25256kB	
8	-> Seq Scan on empregados (cost=0.00..16695.00 rows=1000000 width=15) (actual time=5.173..232.835 rows=1000000 loop...	
9	Planning Time: 0.070 ms	
10	JIT:	
11	Functions: 8	
12	Options: Inlining false, Optimization false, Expressions true, Deforming true	
13	Timing: Generation 0.462 ms, Inlining 0.000 ms, Optimization 0.224 ms, Emission 4.938 ms, Total 5.624 ms	
14	Execution Time: 2358.104 ms	

Otimizado, o sort se mantém pesado com disco, mas o peso da window function some:

	QUERY PLAN text	
1	Unique (cost=133443.34..138443.34 rows=33 width=15) (actual time=1611.342..2116.692 rows=33 loops=1)	
2	-> Sort (cost=133443.34..135943.34 rows=1000000 width=15) (actual time=1611.340..1984.099 rows=1000000 loops=1)	
3	Sort Key: dep_id, salario DESC	
4	Sort Method: external merge Disk: 26688kB	
5	-> Seq Scan on empregados (cost=0.00..16695.00 rows=1000000 width=15) (actual time=3.321..236.528 rows=1000000 loop...	
6	Planning Time: 0.045 ms	
7	JIT:	
8	Functions: 3	
9	Options: Inlining false, Optimization false, Expressions true, Deforming true	
10	Timing: Generation 0.256 ms, Inlining 0.000 ms, Optimization 0.262 ms, Emission 2.999 ms, Total 3.517 ms	
11	Execution Time: 2121.296 ms	

8) Não otimizado:

Cara, demora demais, não sabemos quando terminaria com 1 milhão de tuplas

Otimizado:

	QUERY PLAN	
	text	
1	Hash Join (cost=13955.37..33438.25 rows=333333 width=11) (actual time=263.939..1188.176 rows=494699 loops=1)	
2	Hash Cond: (e.dep_id = empregados.dep_id)	
3	Join Filter: ((e.salario)::numeric > (avg(empregados.salario)))	
4	Rows Removed by Join Filter: 505301	
5	-> Seq Scan on empregados e (cost=0.00..16695.00 rows=1000000 width=15) (actual time=0.039..160.571 rows=1000000 loops=1)	
6	-> Hash (cost=13954.96..13954.96 rows=33 width=36) (actual time=263.893..263.967 rows=33 loops=1)	
7	Buckets: 1024 Batches: 1 Memory Usage: 10kB	
8	-> Finalize GroupAggregate (cost=13946.19..13954.63 rows=33 width=36) (actual time=263.784..263.952 rows=33 loops=1)	
9	Group Key: empregados.dep_id	
10	-> Gather Merge (cost=13946.19..13953.89 rows=66 width=36) (actual time=263.771..263.880 rows=99 loops=1)	
11	Workers Planned: 2	
12	Workers Launched: 2	
13	-> Sort (cost=12946.16..12946.25 rows=33 width=36) (actual time=249.065..249.069 rows=33 loops=3)	
14	Sort Key: empregados.dep_id	
15	Sort Method: quicksort Memory: 27kB	
16	Worker 0: Sort Method: quicksort Memory: 27kB	
17	Worker 1: Sort Method: quicksort Memory: 27kB	
18	-> Partial HashAggregate (cost=12945.00..12945.33 rows=33 width=36) (actual time=249.001..249.015 rows=33 loops=3)	
19	Group Key: empregados.dep_id	
20	Batches: 1 Memory Usage: 24kB	
21	Worker 0: Batches: 1 Memory Usage: 24kB	
22	Worker 1: Batches: 1 Memory Usage: 24kB	
23	-> Parallel Seq Scan on empregados (cost=0.00..10861.67 rows=416667 width=8) (actual time=0.017..58.757 rows=333333 loop...)	
24	Planning Time: 0.181 ms	
25	Execution Time: 1222.133 ms	

9) Não otimizado, a window function custou 847, além do Merge Join:

	QUERY PLAN text	
1	WindowAgg (cost=133445.51..165722.42 rows=1000000 width=47) (actual time=847.223..2424.209 rows=969965 loops=1)	
2	-> Merge Join (cost=133445.51..150722.42 rows=1000000 width=15) (actual time=803.212..1606.704 rows=969965 loops=1)	
3	Merge Cond: (d.dep_id = e.dep_id)	
4	-> Sort (cost=2.16..2.24 rows=33 width=4) (actual time=8.886..8.908 rows=33 loops=1)	
5	Sort Key: d.dep_id	
6	Sort Method: quicksort Memory: 26kB	
7	-> Seq Scan on departamentos d (cost=0.00..1.33 rows=33 width=4) (actual time=8.867..8.875 rows=33 loops=1)	
8	-> Materialize (cost=133443.34..138443.34 rows=1000000 width=15) (actual time=794.286..1293.322 rows=969966 loops=1)	
9	-> Sort (cost=133443.34..135943.34 rows=1000000 width=15) (actual time=794.281..1006.982 rows=969966 loops=1)	
10	Sort Key: e.dep_id	
11	Sort Method: external merge Disk: 26688kB	
12	-> Seq Scan on empregados e (cost=0.00..16695.00 rows=1000000 width=15) (actual time=0.084..224.392 rows=1000000 loop...)	
13	Planning Time: 0.083 ms	
14	JIT:	
15	Functions: 15	
16	Options: Inlining false, Optimization false, Expressions true, Deforming true	
17	Timing: Generation 0.651 ms, Inlining 0.000 ms, Optimization 0.383 ms, Emission 8.529 ms, Total 9.564 ms	
18	Execution Time: 2493.902 ms	

Otimizado, hash join é mais rápido e sem window function:

	QUERY PLAN text	
1	Hash Join (cost=13955.37..33612.49 rows=1000000 width=47) (actual time=252.425..864.254 rows=1000000 loops=1)	
2	Hash Cond: (e.dep_id = d.dep_id)	
3	-> Seq Scan on empregados e (cost=0.00..16695.00 rows=1000000 width=15) (actual time=0.038..134.960 rows=1000000 loops=1)	
4	-> Hash (cost=13954.96..13954.96 rows=33 width=36) (actual time=252.381..252.463 rows=33 loops=1)	
5	Buckets: 1024 Batches: 1 Memory Usage: 10kB	
6	-> Subquery Scan on d (cost=13946.19..13954.96 rows=33 width=36) (actual time=252.234..252.449 rows=33 loops=1)	
7	-> Finalize GroupAggregate (cost=13946.19..13954.63 rows=33 width=36) (actual time=252.233..252.440 rows=33 loops=1)	
8	Group Key: empregados.dep_id	
9	-> Gather Merge (cost=13946.19..13953.89 rows=66 width=36) (actual time=252.224..252.337 rows=99 loops=1)	
10	Workers Planned: 2	
11	Workers Launched: 2	
12	-> Sort (cost=12946.16..12946.25 rows=33 width=36) (actual time=234.397..234.401 rows=33 loops=3)	
13	Sort Key: empregados.dep_id	
14	Sort Method: quicksort Memory: 27kB	
15	Worker 0: Sort Method: quicksort Memory: 27kB	
16	Worker 1: Sort Method: quicksort Memory: 27kB	
17	-> Partial HashAggregate (cost=12945.00..12945.33 rows=33 width=36) (actual time=234.353..234.363 rows=33 loops=3)	
18	Group Key: empregados.dep_id	
19	Batches: 1 Memory Usage: 24kB	
20	Worker 0: Batches: 1 Memory Usage: 24kB	
21	Worker 1: Batches: 1 Memory Usage: 24kB	
22	-> Parallel Seq Scan on empregados (cost=0.00..10861.67 rows=416667 width=8) (actual time=3.370..50.211 rows=333333 loop...)	
23	Planning Time: 0.131 ms	
24	Execution Time: 930.365 ms	