

QUERY 2 – INTERVENÇÃO NO BANCO DE DADOS

```

/* Query 2 seleciona a média salarial geral dos diretores, gestores, especialistas e analistas,
* além de retornar a quantidade de funcionarios em cada tabela;
*/
explain SELECT COUNT (d.idt_employee) as qty_Director, AVG(d.salary) as mean_salary_Director,
COUNT (m.idt_employee) as qty_Manager, AVG(m.salary) as mean_salary_Manager,
COUNT (es.idt_employee) as qty_Especialist, AVG(es.salary) as mean_salary_Especialist,
COUNT (a.idt_employee) as qty_Analyst, AVG(a.salary) as mean_salary_Analyst
FROM employee as e
LEFT JOIN director as d ON d.idt_employee = e.idt_employee
LEFT JOIN manager as m ON m.idt_employee = e.idt_employee
LEFT JOIN especialista as es ON es.idt_employee = e.idt_employee
LEFT JOIN analyst as a ON a.idt_employee = e.idt_employee

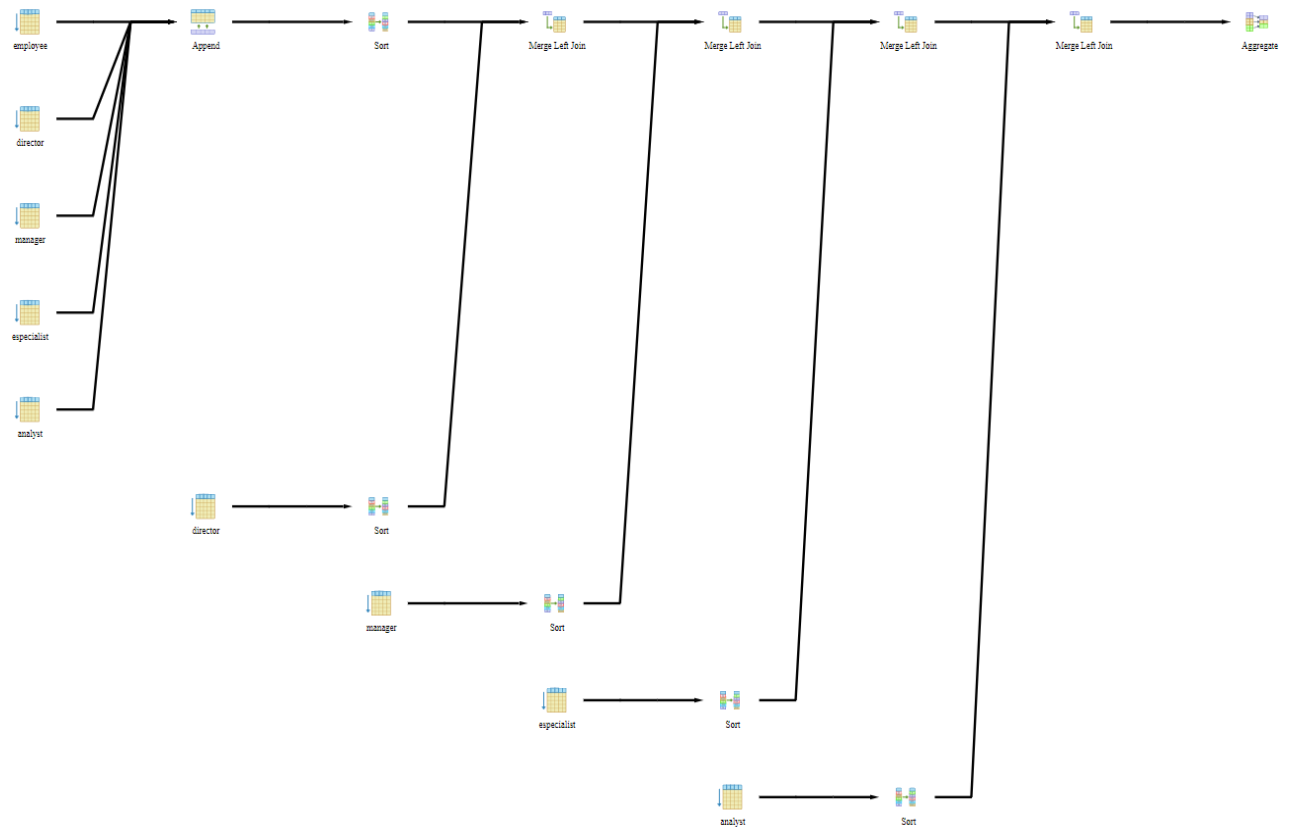
```

Consulta

Plano de Consulta:

	QUERY PLAN
	text
1	Aggregate (cost=873.18..873.19 rows=1 width=160)
2	-> Merge Left Join (cost=242.03..645.77 rows=11370 width=144)
3	Merge Cond: (e.idt_employee = a.idt_employee)
4	-> Merge Left Join (cost=214.34..430.80 rows=6688 width=112)
5	Merge Cond: (e.idt_employee = es.idt_employee)
6	-> Merge Left Join (cost=186.64..292.95 rows=3934 width=76)
7	Merge Cond: (e.idt_employee = m.idt_employee)
8	-> Merge Left Join (cost=158.94..200.46 rows=2314 width=40)
9	Merge Cond: (e.idt_employee = d.idt_employee)
10	-> Sort (cost=131.25..134.65 rows=1361 width=4)
11	Sort Key: e.idt_employee
12	-> Append (cost=0.00..60.41 rows=1361 width=4)
13	-> Seq Scan on employee e (cost=0.00..0.00 rows=1 width=4)
14	-> Seq Scan on director e_1 (cost=0.00..13.40 rows=340 width=4)
15	-> Seq Scan on manager e_2 (cost=0.00..13.40 rows=340 width=4)
16	-> Seq Scan on especialista e_3 (cost=0.00..13.40 rows=340 width=4)
17	-> Seq Scan on analyst e_4 (cost=0.00..13.40 rows=340 width=4)
18	-> Sort (cost=27.70..28.55 rows=340 width=36)
19	Sort Key: d.idt_employee
20	-> Seq Scan on director d (cost=0.00..13.40 rows=340 width=36)
21	-> Sort (cost=27.70..28.55 rows=340 width=36)
22	Sort Key: m.idt_employee
23	-> Seq Scan on manager m (cost=0.00..13.40 rows=340 width=36)
24	-> Sort (cost=27.70..28.55 rows=340 width=36)
25	Sort Key: es.idt_employee
26	-> Seq Scan on especialista es (cost=0.00..13.40 rows=340 width=36)
27	-> Sort (cost=27.70..28.55 rows=340 width=36)
28	Sort Key: a.idt_employee
29	-> Seq Scan on analyst a (cost=0.00..13.40 rows=340 width=36)

Árvore de Consulta:



QUERY 2: CRIAÇÃO DE ÍNDICES

```
CREATE INDEX indicesegundaQuery  
ON director (idt_employee);
```

```
CREATE INDEX indicesegundaQueryManager  
ON manager (idt_employee);
```

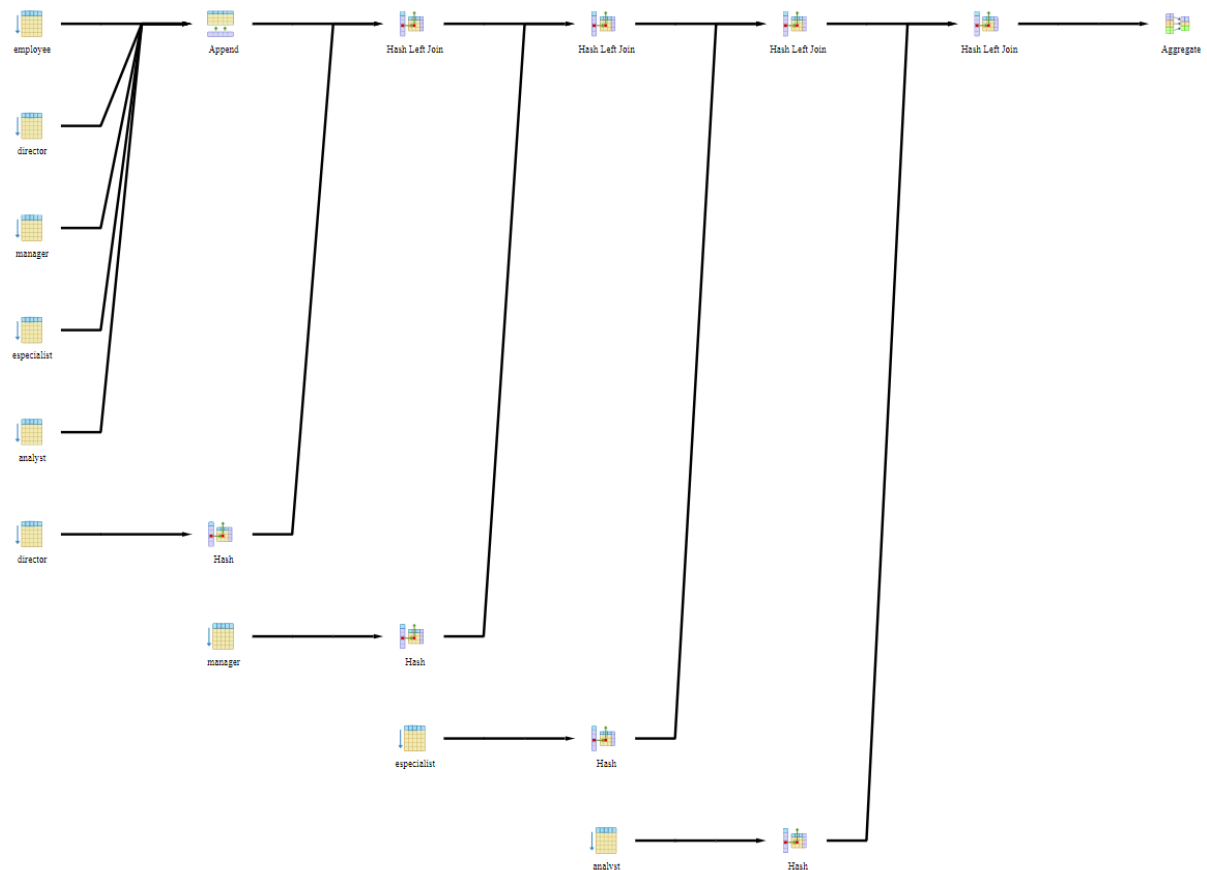
```
CREATE INDEX indicesegundaQuerySpecialist  
ON specialist (idt_employee);
```

```
CREATE INDEX indicesegundaQueryAnalyst  
ON analyst (idt_employee);
```

Relatório do plano de consulta a partir do comando explain do Postgress

1	Aggregate (cost=10.19..10.20 rows=1 width=160)	
2	-> Hash Left Join (cost=4.58..9.64 rows=27 width=144)	
3	Hash Cond: (e.idt_employee = a.idt_employee)	
4	-> Hash Left Join (cost=3.25..8.06 rows=27 width=112)	
5	Hash Cond: (e.idt_employee = es.idt_employee)	
6	-> Hash Left Join (cost=2.13..6.79 rows=27 width=76)	
7	Hash Cond: (e.idt_employee = m.idt_employee)	
8	-> Hash Left Join (cost=1.02..5.53 rows=27 width=40)	
9	Hash Cond: (e.idt_employee = d.idt_employee)	
10	-> Append (cost=0.00..4.39 rows=27 width=4)	
11	-> Seq Scan on employee e (cost=0.00..0.00 rows=1 width=4)	
12	-> Seq Scan on director e_1 (cost=0.00..1.01 rows=1 width=4)	
13	-> Seq Scan on manager e_2 (cost=0.00..1.05 rows=5 width=4)	
14	-> Seq Scan on especialista e_3 (cost=0.00..1.05 rows=5 width=4)	
15	-> Seq Scan on analyst e_4 (cost=0.00..1.15 rows=15 width=4)	
16	-> Hash (cost=1.01..1.01 rows=1 width=36)	
17	-> Seq Scan on director d (cost=0.00..1.01 rows=1 width=36)	
18	-> Hash (cost=1.05..1.05 rows=5 width=36)	
19	-> Seq Scan on manager m (cost=0.00..1.05 rows=5 width=36)	
20	-> Hash (cost=1.05..1.05 rows=5 width=36)	
21	-> Seq Scan on especialista es (cost=0.00..1.05 rows=5 width=36)	
22	-> Hash (cost=1.15..1.15 rows=15 width=36)	
23	-> Seq Scan on analyst a (cost=0.00..1.15 rows=15 width=36)	:

Árvore do plano de consulta a partir do comando explain do Postgress:



Comparação antes e depois da alteração

A consulta realizada demonstra a média salarial de cada tipo de funcionário da empresa, seus managers, directors, analysts e specialists, assim como sua quantidade.

Analisando o relatório antes da inclusão dos índices vemos através do comando explain a progressão da consulta, seguindo a mesma progressão da primeira query analisada no artefato C vemos que há um grande custo, tanto do aggregate quanto da soma dos vários sorts presentes, com vários merges extremamente custosos, necessitando então de uma melhor saída, tendo em vista a longevidade da consulta, que provavelmente não seria sustentável em buscas de larga escala em grandes quantidades de dados.

Portanto, houve a criação de quatro índices, “indicessegundaQuery”, “indicessegundaQueryManager”, “indicessegundaQuerySpecialist” e “indicessegundaQueryAnalyst” que “indexam” o atributo idt_employee nas tables de director, manager, specialist e analyst, respectivamente, o que nos leva ao segundo resultado exposto com o comando explain. Percebe-se com grande facilidade que houve uma grande diminuição no custo da iteração, logo na primeira linha há uma diminuição de quase 100% no custo e logo depois segue-se o mesmo padrão, não há mais merges, sendo todos substituídos por hashes, todos com valor em custo ínfimo, mostrando-se muito eficaz, evidenciando que a mudança foi essencial e necessária

Por fim, conclui-se que após a inclusão dos índices que consultas com maior carga de dados não seria um problema, já que o custo foi reduzido em níveis muito altos, permitindo assim maior inclusão de funcionários e seus salários, assim como o contínuo cálculo de suas respectivas médias salariais.