```sql
/* 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 especialist as es ON es.idt_employee = e.idt_employee
LEFT JOIN analyst as a ON a.idt_employee = e.idt_employee
```

QUERY 2 - ESCOLHIDA

Estratégia equivalente utilizada: CROSS JOIN

	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 especialist 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 especialist 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)

VS

	QUERY PLAN text
1	Aggregate (cost=74329.77..74329.78 rows=1 width=160)
2	-> Nested Loop (cost=214.34..28860.92 rows=2273442 width=144)
3	-> Merge Right Join (cost=214.34..422.67 rows=6688 width=108)
4	Merge Cond: (a.idt_employee = e.idt_employee)
5	-> Sort (cost=27.70..28.55 rows=340 width=36)
6	Sort Key: a.idt_employee
7	-> Seq Scan on analyst a (cost=0.00..13.40 rows=340 width=36)
8	-> Materialize (cost=186.64..302.79 rows=3934 width=76)
9	-> Merge Left Join (cost=186.64..292.95 rows=3934 width=76)
10	Merge Cond: (e.idt_employee = es.idt_employee)
11	-> Merge Left Join (cost=158.94..200.46 rows=2314 width=40)
12	Merge Cond: (e.idt_employee = m.idt_employee)
13	-> Sort (cost=131.25..134.65 rows=1361 width=4)
14	Sort Key: e.idt_employee
15	-> Append (cost=0.00..60.41 rows=1361 width=4)
16	-> Seq Scan on employee e (cost=0.00..0.00 rows=1 width=4)
17	-> Seq Scan on director e_1 (cost=0.00..13.40 rows=340 width=4)
18	-> Seq Scan on manager e_2 (cost=0.00..13.40 rows=340 width=4)
19	-> Seq Scan on especialist e_3 (cost=0.00..13.40 rows=340 width=4)
20	-> Seq Scan on analyst e_4 (cost=0.00..13.40 rows=340 width=4)
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 especialist es (cost=0.00..13.40 rows=340 width=36)
27	-> Materialize (cost=0.00..15.10 rows=340 width=36)
28	-> Seq Scan on director d (cost=0.00..13.40 rows=340 width=36)

QUERY 2: LEFT JOIN

QUERY 2: CROSS JOIN

Tentamos pensar em mais estratégias e melhores formas de tentarmos implementar a cláusula “CROSS JOIN” como estratégia utilizada. A forma que foi mais “minimizada” foi utilizando após a tabela “Director”. É possível perceber que muitos os custos se mantiveram bem parecidos em todos os pontos, inclusive em rotinas que não ocorreram na query 2 (sem modificação). Apesar da estratégia equivalente implementada retornar um custo “melhor, isso ocorre pela

falha da implementação, ocorrendo um retorno inválido do campo de quantidade de funcionários com o cargo de “Director”.