

Comando EXPLAIN sem alteração no modelo e sem melhoria na QUERY 1

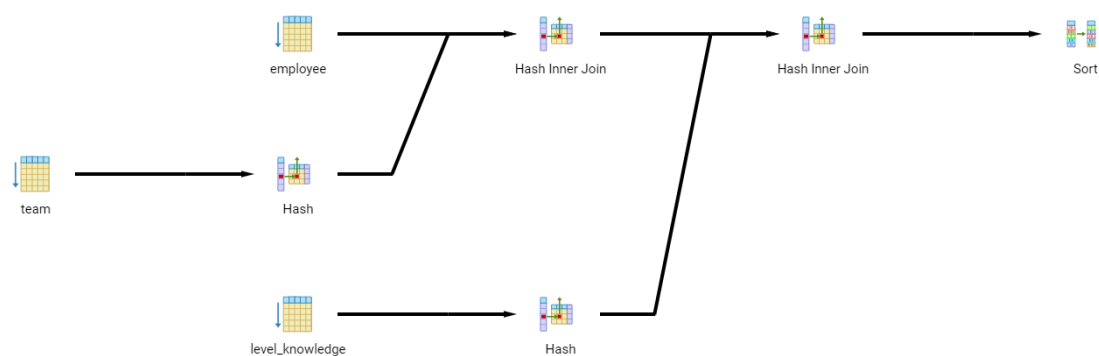
```
explain SELECT A.idt_employee, A.name_employee, B.name_knowledge ,A.idt_team, A.name_team
FROM (SELECT E.idt_level_knowledge, E.name_employee, E.idt_employee, T.name_team, T.idt_team
      FROM Employee as E
      INNER JOIN Team as T
      ON E.idt_team = T.idt_team ) as A INNER JOIN level_knowledge as B
ON A.idt_level_knowledge = B.idt_level_knowledge
ORDER BY A.name_team, B.idt_level_knowledge
```

Consulta executada com o comando EXPLAIN

1. → Sort
2. → Hash Inner Join
Hash Cond: (e.idt_level_knowledge = b.idt_level_knowledge)
3. → Hash Inner Join
Hash Cond: (e.idt_team = t.idt_team)
4. → Seq Scan on employee as e
5. → Hash
6. → Seq Scan on team as t
7. → Hash
8. → Seq Scan on level_knowledge as b

QUERY PLAN	
text	
1	Sort (cost=130.75..132.55 rows=720 width=108)
2	Sort Key: t.name_team, b.idt_level_knowledge
3	-> Hash Join (cost=75.58..96.58 rows=720 width=108)
4	Hash Cond: (e.idt_level_knowledge = b.idt_level_knowledge)
5	-> Hash Join (cost=37.00..56.10 rows=720 width=76)
6	Hash Cond: (e.idt_team = t.idt_team)
7	-> Seq Scan on employee e (cost=0.00..17.20 rows=720 width=44)
8	-> Hash (cost=22.00..22.00 rows=1200 width=36)
9	-> Seq Scan on team t (cost=0.00..22.00 rows=1200 width=36)
10	-> Hash (cost=22.70..22.70 rows=1270 width=36)
11	-> Seq Scan on level_knowledge b (cost=0.00..22.70 rows=1270 width=36)

Plano de consulta executada com o comando EXPLAIN



Árvore de consulta executada com o comando EXPLAIN

Comando EXPLAIN com alteração no modelo e sem melhoria na QUERY 1

```
/* Query 1 correspondente apos modificacoes realizadas no modelo e apos o artefato A
* Pelo fato de que agora apenas um analista ou especialista possuem nivel de conhecimento,
* tais atributos e tabelas foram tirados da consulta a fim de evitar muitos registros 'nulos'.
*/
```

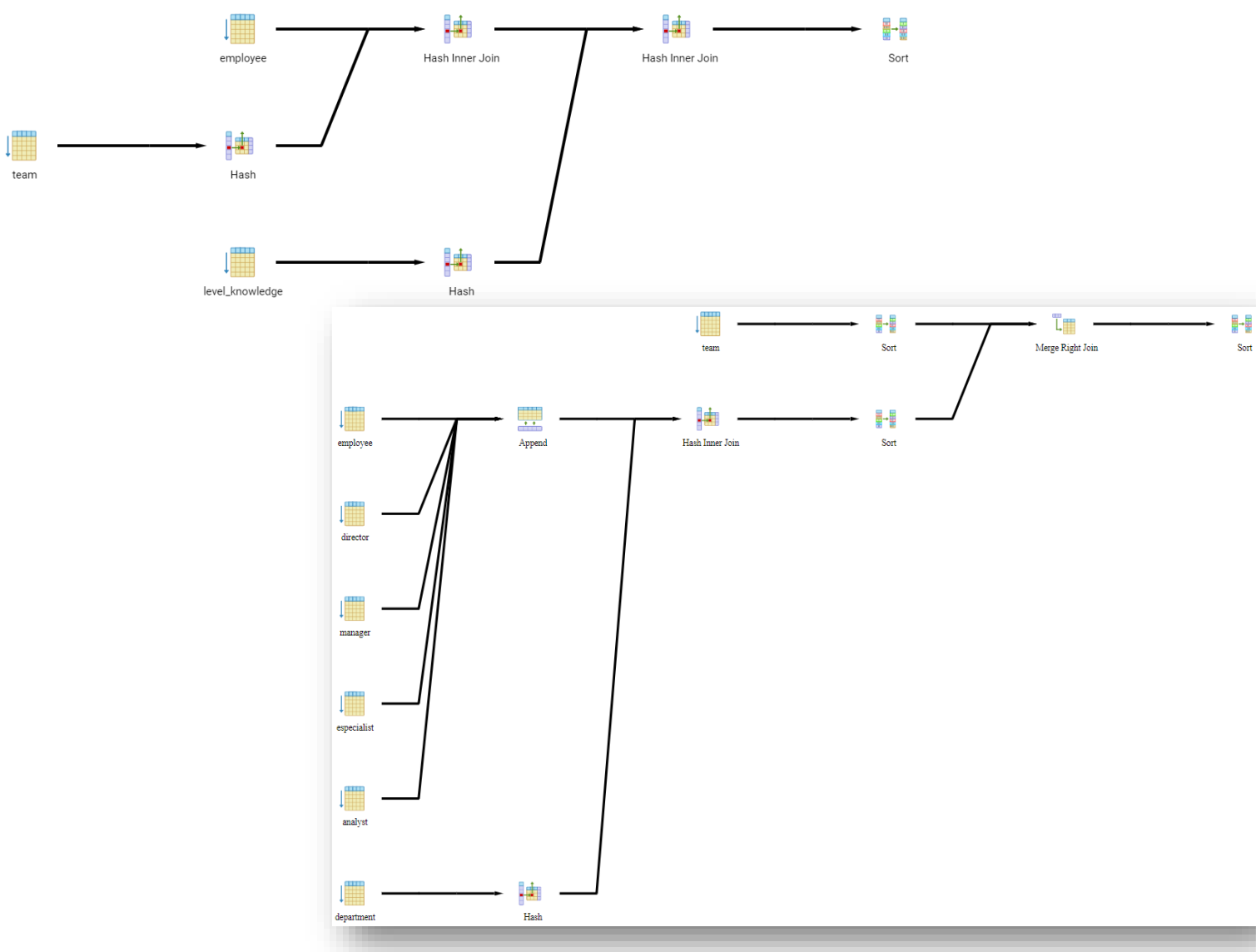
```
explain SELECT a.idt_employee, a.name_employee, t.idt_team, t.name_team
FROM(
    SELECT e.name_employee, e.idt_employee, d.idt_project
    FROM Employee as e
    INNER JOIN Department as d ON d.idt_department = e.idt_department
) as a
LEFT JOIN Team as t ON t.idt_project = a.idt_project
ORDER BY t.name_team;
```

Consulta alterada e executada com o comando EXPLAIN

	QUERY PLAN text
1	Sort (cost=130.75..132.55 rows=720 width=108)
2	Sort Key: t.name_team, b.idt_level_knowledge
3	-> Hash Join (cost=75.58..96.58 rows=720 width=108)
4	Hash Cond: (e.idt_level_knowledge = b.idt_level_knowledge)
5	-> Hash Join (cost=37.00..56.10 rows=720 width=76)
6	Hash Cond: (e.idt_team = t.idt_team)
7	-> Seq Scan on employee e (cost=0.00..17.20 rows=720 width=44)
8	-> Hash (cost=22.00..22.00 rows=1200 width=36)
9	-> Seq Scan on team t (cost=0.00..22.00 rows=1200 width=36)
10	-> Hash (cost=22.70..22.70 rows=1270 width=36)
11	-> Seq Scan on level_knowledge b (cost=0.00..22.70 rows=1270 width=36)

	QUERY PLAN text
1	Sort (cost=914.30..934.71 rows=8166 width=72)
2	Sort Key: t.name_team
3	-> Merge Right Join (cost=255.20..383.69 rows=8166 width=72)
4	Merge Cond: (t.idt_project = d.idt_project)
5	-> Sort (cost=83.37..86.37 rows=1200 width=40)
6	Sort Key: t.idt_project
7	-> Seq Scan on team t (cost=0.00..22.00 rows=1200 width=40)
8	-> Sort (cost=171.83..175.23 rows=1361 width=40)
9	Sort Key: d.idt_project
10	-> Hash Join (cost=37.00..100.99 rows=1361 width=40)
11	Hash Cond: (e.idt_department = d.idt_department)
12	-> Append (cost=0.00..60.41 rows=1361 width=40)
13	-> Seq Scan on employee e (cost=0.00..0.00 rows=1 width=40)
14	-> Seq Scan on director e_1 (cost=0.00..13.40 rows=340 width=40)
15	-> Seq Scan on manager e_2 (cost=0.00..13.40 rows=340 width=40)
16	-> Seq Scan on especialista e_3 (cost=0.00..13.40 rows=340 width=40)
17	-> Seq Scan on analyst e_4 (cost=0.00..13.40 rows=340 width=40)
18	-> Hash (cost=22.00..22.00 rows=1200 width=8)
19	-> Seq Scan on department d (cost=0.00..22.00 rows=1200 width=8)

Comparação entre os planos de consultas antes x depois da alteração do modelo



Comparação entre as árvores de consultas antes x depois da alteração do modelo

A primeira consulta realizada na “Parte 1” do trabalho seleciona o funcionário, seu id, seu nível de conhecimento e seu time, assim como o id do seu time. O que o plano de execução acima nos retorna é uma busca sequencial (“seq scan”) em cada tabela através do campo de chave primaria nas junções (“joins”), fazendo uma varredura na tabela inteira para conseguir o resultado da consulta, logo, esse processo é muito custoso principalmente se fosse uma tabela que tivesse um volume grande de dados.

Após a alteração do modelo foi necessário tirar alguns campos da segunda consulta (por razões de adequação), e por consequência a tabela que utilizava tais campos (campos referentes a tabela “Level_Knowledge”), para que seja possível uma melhor visualização dos dados ao passo que poucas tabelas utilizam esses campos (como descrito no artefato “A”, agora somente “Especialist” e “Analyst” possuem um “Level_Knowledge”), retornando, assim, menos campos ao selecionar o funcionário, o seu time e os respectivos “id’s” (id do funcionário, e id do time).

O que o segundo plano de execução nos retornou logo de cara foi que continuou realizando buscas sequenciais (“seq scan”), porém, agora através do campo “name_team” (nome do time), além dos “sorts” mais custosos do que a busca realizada no primeiro plano de execução. Devemos considerar também que agora a busca “varre” mais tabelas. Um bom exemplo são as quantidades de linhas dos “sorts” realizados, por mais que a janela tenha diminuído, as linhas afetadas aumentaram drasticamente.