

Universidade de São Paulo
Escola de Artes, Ciências e Humanidades
Disciplina: Laboratório de Banco de Dados
Profª Dra. Fátima Nunes.

Administração de Condomínio

Parte III - Artefato E

Fernando K. G. de Amorim – 10387644
João Guilherme da Costa Seike – 9784634
Lucas Pereira Castelo Branco – 10258772
Victor Gomes de O. M. Nicola – 9844881

Objetivo do artefato E:

- **(Artefato do tipo texto)** Projete 1 consulta que faça uso da estratégia de subconsulta (subquery). Se uma de suas consultas usadas nos itens anteriores atende a esse quesito, você pode usá-la. Reescreva-a sem usar essa estratégia. Execute as 2 consultas criadas e proceda com a análise dos custos e planos (os mesmos itens de relatórios e prints do item (b) devem ser expostos aqui para fins de comparação). **O texto deve ser entregue em formato PDF no e-Disciplinas. O código deverá ser entregue em formato ASCII (.txt) no e-Disciplinas (ver observação em vermelho abaixo).**

Neste artefato, a segunda consulta criada na parte I do trabalho fora reutilizada, pois já contava com a estratégia de subconsulta e foi reescrita sem essa estratégia, conforme consultas abaixo. Todas as representações gráficas abaixo foram retiradas da ferramenta EXPLAIN disponível no software PGAdmin.

Consulta com subconsulta:

WITH top_five AS (

```
SELECT Pessoa.id_pessoa AS id_pessoa, COUNT(es_pessoa.*) AS entradas
FROM adm_condominio.Pessoa AS Pessoa
JOIN adm_condominio.es_pessoa AS es_pessoa
ON es_pessoa.fk_id_pessoa = Pessoa.id_pessoa
GROUP BY 1 ORDER BY 2,1 ASC
LIMIT 5
```

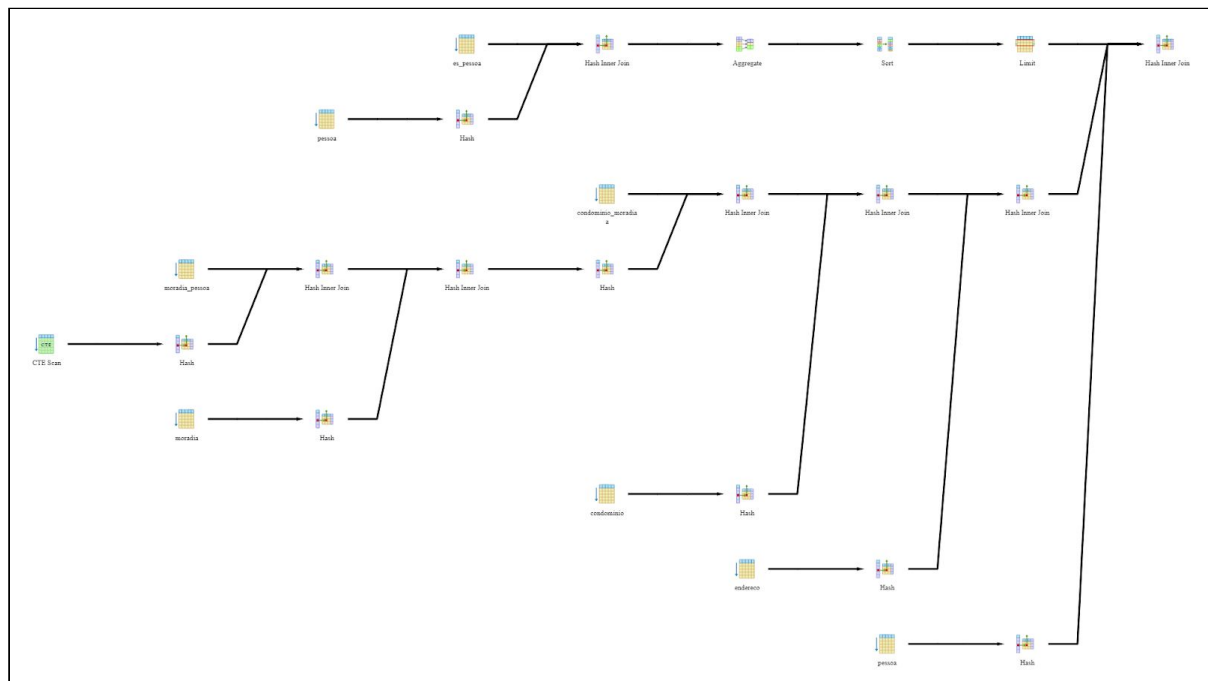
)

```
SELECT Pessoa.nome, top_five.entradas, Endereco.cidade
FROM adm_condominio.Pessoa AS Pessoa
INNER JOIN top_five
ON top_five.id_pessoa = Pessoa.id_pessoa
JOIN adm_condominio.Moradia_Pessoa AS Moradia_Pessoa
ON Moradia_Pessoa.fk_id_pessoa = Pessoa.id_pessoa
JOIN adm_condominio.Moradia AS Moradia
ON Moradia.id_moradia = Moradia_Pessoa.fk_id_moradia
JOIN adm_condominio.Condominio_Moradia AS Condominio_Moradia
ON Condominio_Moradia.fk_id_moradia = Moradia.id_moradia
JOIN adm_condominio.Condominio AS Condominio
ON Condominio.id_condominio = Condominio_Moradia.fk_id_condominio
JOIN adm_condominio.Endereco AS Endereco
ON Endereco.id_endereco = Condominio.fk_id_endereco
```

Plano de execução da consulta com subconsulta:

```
"Hash Join (cost=111.62..170.49 rows=2838 width=31) (actual time=1.211..1.233 rows=5 loops=1)"
" Hash Cond: (top_five.id_pessoa = pessoa.id_pessoa)"
" CTE top_five"
"   -> Limit (cost=55.87..55.88 rows=5 width=12) (actual time=0.236..0.240 rows=5 loops=1)"
"   -> Sort (cost=55.87..56.12 rows=100 width=12) (actual time=0.235..0.235 rows=5 loops=1)"
"       Sort Key: count(es_pessoa.?), pessoa_1.id_pessoa"
"       Sort Method: quicksort Memory: 25kB"
"   -> HashAggregate (cost=53.20..54.20 rows=100 width=12) (actual time=0.191..0.196 rows=10 loops=1)"
"       Group Key: pessoa_1.id_pessoa"
"       -> Hash Join (cost=3.25..41.90 rows=2260 width=36) (actual time=0.162..0.176 rows=10 loops=1)"
"           Hash Cond: (es_pessoa.fk_id_pessoa = pessoa_1.id_pessoa)"
"           -> Seq Scan on es_pessoa (cost=0.00..32.60 rows=2260 width=36) (actual time=0.051..0.058 rows=10 loops=1)"
"           -> Hash (cost=2.00..2.00 rows=100 width=4) (actual time=0.088..0.088 rows=100 loops=1)"
"               Buckets: 1024 Batches: 1 Memory Usage: 12kB"
"               -> Seq Scan on pessoa pessoa_1 (cost=0.00..2.00 rows=100 width=4) (actual time=0.022..0.047 rows=100 loops=1)"
"   -> Hash Join (cost=52.49..107.51 rows=1406 width=26) (actual time=1.021..1.040 rows=5 loops=1)"
"       Hash Cond: (condominio.fk_id_endereco = endereco.id_endereco)"
"       -> Hash Join (cost=48.24..99.42 rows=1406 width=20) (actual time=0.879..0.896 rows=5 loops=1)"
"           Hash Cond: (condominio.moradia.fk_id_condominio = condominio.id_condominio)"
"           -> Hash Join (cost=45.67..93.07 rows=1406 width=20) (actual time=0.680..0.694 rows=5 loops=1)"
"               Hash Cond: (condominio.moradia.fk_id_moradia = moradia.id_moradia)"
"               -> Seq Scan on condominio.moradia (cost=0.00..32.60 rows=2260 width=8) (actual time=0.029..0.035 rows=40 loops=1)"
"               -> Hash (cost=44.97..44.97 rows=56 width=24) (actual time=0.599..0.599 rows=5 loops=1)"
"                   Buckets: 1024 Batches: 1 Memory Usage: 9kB"
"                   -> Hash Join (cost=3.19..44.97 rows=56 width=24) (actual time=0.572..0.590 rows=5 loops=1)"
"                       Hash Cond: (moradia.pessoa.fk_id_moradia = moradia.id_moradia)"
"                       -> Hash Join (cost=0.16..41.80 rows=56 width=20) (actual time=0.306..0.322 rows=5 loops=1)"
"                           Hash Cond: (moradia.pessoa.fk_id_pessoa = top_five.id_pessoa)"
"                           -> Seq Scan on moradia.pessoa (cost=0.00..32.60 rows=2260 width=8) (actual time=0.024..0.031 rows=40 loops=1)"
"                           -> Hash (cost=0.10..0.10 rows=5 width=12) (actual time=0.255..0.255 rows=5 loops=1)"
"                               Buckets: 1024 Batches: 1 Memory Usage: 9kB"
"                               -> CTE Scan on top_five (cost=0.00..0.10 rows=5 width=12) (actual time=0.240..0.247 rows=5 loops=1)"
"                       -> Hash (cost=1.90..1.90 rows=90 width=4) (actual time=0.083..0.083 rows=90 loops=1)"
"                           Buckets: 1024 Batches: 1 Memory Usage: 12kB"
"                           -> Seq Scan on moradia (cost=0.00..1.90 rows=90 width=4) (actual time=0.027..0.049 rows=90 loops=1)"
"               -> Hash (cost=1.70..1.70 rows=70 width=8) (actual time=0.177..0.177 rows=70 loops=1)"
"                   Buckets: 1024 Batches: 1 Memory Usage: 11kB"
"                   -> Seq Scan on condominio (cost=0.00..1.70 rows=70 width=8) (actual time=0.125..0.146 rows=70 loops=1)"
"           -> Hash (cost=3.00..3.00 rows=100 width=14) (actual time=0.115..0.115 rows=100 loops=1)"
"               Buckets: 1024 Batches: 1 Memory Usage: 13kB"
"               -> Seq Scan on endereco (cost=0.00..3.00 rows=100 width=14) (actual time=0.030..0.063 rows=100 loops=1)"
"       -> Hash (cost=2.00..2.00 rows=100 width=17) (actual time=0.164..0.164 rows=100 loops=1)"
"           Buckets: 1024 Batches: 1 Memory Usage: 13kB"
"           -> Seq Scan on pessoa (cost=0.00..2.00 rows=100 width=17) (actual time=0.070..0.101 rows=100 loops=1)"
"Planning time: 5.782 ms"
"Execution time: 1.714 ms"
```

Imagem do plano de execução da consulta com subconsulta:



O plano de execução se inicia com a join das tabelas 'top_five' com 'Pessoa' e, para que isso seja possível, é necessária a criação da tabela 'top_five' por meio da CTE (Common table expression), que é gerada pela cláusula WITH e definida pela subconsulta. A execução da CTE depende do resultado do comando LIMIT, que por sua vez é realizado após a ordenação com quicksort do agrupamento do resultado do JOIN entre tabelas 'Pessoa' e 'es_pessoa'. Para esse aninhamento, o SGBD optou por um INNER JOIN para obter dados relacionados das duas tabelas.

A continuidade do plano se dá após a criação da tabela 'top_five', realizando o JOIN da tabela 'Endereco' com a tabela 'Condominio'. Aqui, o INNER JOIN também foi escolhido para essa operação. Sequencialmente, há outro INNER JOIN para as tabelas 'Condominio' e 'condominio_moradia'. 'condominio_moradia', por sua vez, será unida à tabela 'Moradia', também com o mesmo tipo de JOIN. Por fim, a tabela Moradia será unida à tabela 'moradia_pessoa', mais uma vez pelo INNER JOIN.

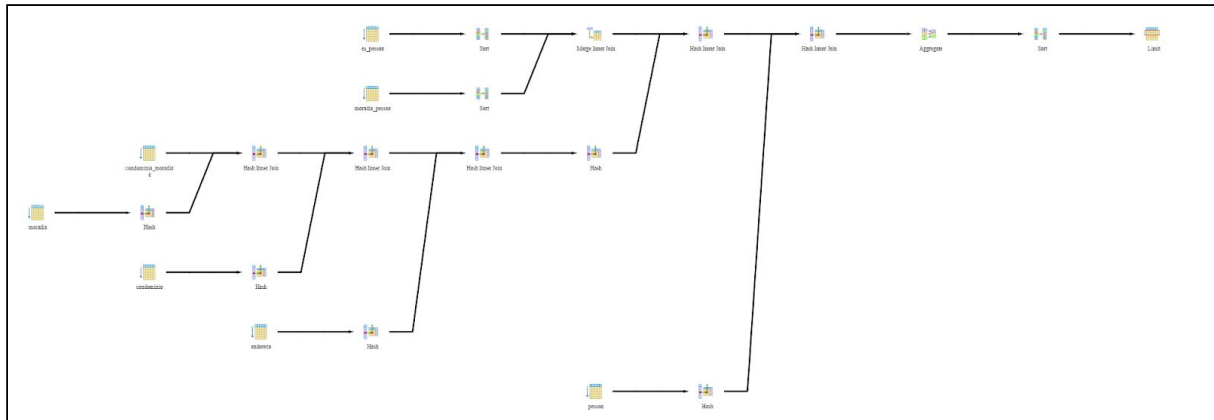
Consulta sem subconsulta:

```
SELECT Pessoa.nome, COUNT(es_pessoa.*), Endereco.cidade
FROM adm_condominio.Pessoa AS Pessoa
JOIN adm_condominio.es_pessoa AS es_pessoa
  ON es_pessoa.fk_id_pessoa = Pessoa.id_pessoa
JOIN adm_condominio.Moradia_Pessoa AS Moradia_Pessoa
  ON Moradia_Pessoa.fk_id_pessoa = Pessoa.id_pessoa
JOIN adm_condominio.Moradia AS Moradia
  ON Moradia.id_moradia = Moradia_Pessoa.fk_id_moradia
JOIN adm_condominio.Condominio_Moradia AS Condominio_Moradia
  ON Condominio_Moradia.fk_id_moradia = Moradia.id_moradia
JOIN adm_condominio.Condominio AS Condominio
  ON Condominio.id_condominio = Condominio_Moradia.fk_id_condominio
JOIN adm_condominio.Endereco AS Endereco
  ON Endereco.id_endereco = Condominio.fk_id_endereco
GROUP BY 1,Pessoa.id_pessoa,3
ORDER BY 2,Pessoa.id_pessoa ASC
LIMIT 5
```

Plano de execução da consulta sem subconsulta:

```
[*Limit (cost=16036.04..16036.05 rows=5 width=35) (actual time=0.487..0.494 rows=5 loops=1)"
" -> Sort (cost=16036.04..16049.79 rows=5500 width=35) (actual time=0.486..0.487 rows=5 loops=1)"
"   Sort Key: (count(es_pessoa.*)), pessoa.id_pessoa"
"   Sort Method: quicksort Memory: 25kB"
"   -> HashAggregate (cost=15889.69..15944.69 rows=5500 width=35) (actual time=0.384..0.411 rows=10 loops=1)"
"     Group Key: pessoa.id_pessoa, endereco.cidade"
"     -> Hash Join (cost=409.26..6270.37 rows=1282575 width=59) (actual time=0.352..0.371 rows=10 loops=1)"
"       Hash Cond: (es_pessoa.fk_id_pessoa = pessoa.id_pessoa)"
"       -> Hash Join (cost=406.01..4548.08 rows=641288 width=50) (actual time=0.285..0.300 rows=10 loops=1)"
"         Hash Cond: (moradia_pessoa.fk_id_moradia = moradia.id_moradia)"
"         -> Merge Join (cost=317.01..711.38 rows=25538 width=44) (actual time=0.086..0.095 rows=10 loops=1)"
"           Merge Cond: (es_pessoa.fk_id_pessoa = moradia_pessoa.fk_id_pessoa)"
"           -> Sort (cost=158.51..164.16 rows=2260 width=36) (actual time=0.046..0.047 rows=10 loops=1)"
"             Sort Key: es_pessoa.fk_id_pessoa"
"             Sort Method: quicksort Memory: 25kB"
"             -> Seq Scan on es_pessoa (cost=0.00..32.60 rows=2260 width=36) (actual time=0.022..0.027 rows=10 loops=1)"
"           -> Sort (cost=158.51..164.16 rows=2260 width=8) (actual time=0.034..0.035 rows=21 loops=1)"
"             Sort Key: moradia_pessoa.fk_id_pessoa"
"             Sort Method: quicksort Memory: 26kB"
"             -> Seq Scan on moradia_pessoa (cost=0.00..32.60 rows=2260 width=8) (actual time=0.011..0.018 rows=40 loops=1)"
"         -> Hash (cost=60.75..60.75 rows=2260 width=18) (actual time=0.181..0.181 rows=40 loops=1)"
"           Buckets: 4096 Batches: 1 Memory Usage: 35kB"
"           -> Hash Join (cost=9.85..60.75 rows=2260 width=18) (actual time=0.142..0.168 rows=40 loops=1)"
"             Hash Cond: (condominio.fk_id_endereco = endereco.id_endereco)"
"             -> Hash Join (cost=5.60..50.31 rows=2260 width=12) (actual time=0.090..0.109 rows=40 loops=1)"
"               Hash Cond: (condominio_moradia.fk_id_condominio = condominio.id_condominio)"
"               -> Hash Join (cost=3.02..41.68 rows=2260 width=12) (actual time=0.047..0.058 rows=40 loops=1)"
"                 Hash Cond: (condominio_moradia.fk_id_moradia = moradia.id_moradia)"
"                 -> Seq Scan on condominio_moradia (cost=0.00..32.60 rows=2260 width=8) (actual time=0.009..0.011 rows=40 loops=1)"
"                 -> Hash (cost=1.90..1.90 rows=90 width=4) (actual time=0.032..0.032 rows=90 loops=1)"
"                   Buckets: 1024 Batches: 1 Memory Usage: 12kB"
"                   -> Seq Scan on moradia (cost=0.00..1.90 rows=90 width=4) (actual time=0.009..0.016 rows=90 loops=1)"
"                 -> Hash (cost=1.70..1.70 rows=70 width=8) (actual time=0.034..0.034 rows=70 loops=1)"
"                   Buckets: 1024 Batches: 1 Memory Usage: 11kB"
"                   -> Seq Scan on condominio (cost=0.00..1.70 rows=70 width=8) (actual time=0.014..0.021 rows=70 loops=1)"
"                 -> Hash (cost=3.00..3.00 rows=100 width=14) (actual time=0.044..0.044 rows=100 loops=1)"
"                   Buckets: 1024 Batches: 1 Memory Usage: 13kB"
"                   -> Seq Scan on endereco (cost=0.00..3.00 rows=100 width=14) (actual time=0.012..0.023 rows=100 loops=1)"
"             -> Hash (cost=2.00..2.00 rows=100 width=17) (actual time=0.057..0.057 rows=100 loops=1)"
"               Buckets: 1024 Batches: 1 Memory Usage: 13kB"
"               -> Seq Scan on pessoa (cost=0.00..2.00 rows=100 width=17) (actual time=0.028..0.038 rows=100 loops=1)"
"Planning time: 2.312 ms"
"Execution time: 0.817 ms"
```

Imagem do plano de execução da consulta sem subquery:



A segunda consulta não possui subconsulta e, portanto, possui dependência de um fluxo menor de instruções. O plano de execução dela inicia-se pela Limitação, ordenação (novamente por quicksort) e agrupamento do resultado da junção da tabela pessoa com o resultado de outra junção. Essa outra junção é um INNER JOIN entre a tabela endereco e o resultado de um Merge Inner Join das tabelas 'es_pessoa' e 'moradia_pessoa', ambas ordenadas, com o mesmo algoritmo das outras ordenações, para otimização do tempo de execução. Para que essa parte do plano seja executada, a tabela 'endereco' precisa ser unida a tabela condomínio. A sequência do plano se dá pelo INNER JOIN da tabela 'condominio' com a tabela 'condominio_moradia', sendo que a tabela 'condominio_moradia' precisa ser um INNER JOIN com a tabela 'moradia';

Apesar de em ambas as consultas, o INNER JOIN ter sido escolhido em todas as junções, trazendo todos os dados das tabelas e diminuindo o desempenho, o tempo de execução da segunda consulta (sem subconsulta) foi inferior devido ao fato de a dependência das instruções ser menor. Afinal, o resultado da consulta principal não depende do resultado da subconsulta. Além disso, é interessante atentar-se o fato de que a escolha do Merge Inner Join em duas tabelas ordenadas pela consulta 2, provavelmente contribuiu para o desempenho superior, pois ao comparar o tempo gasto nessa operação, com todas as outras junções, a diferença foi significativa.