

Construção de Sistemas de Banco de Dados

Relatório - Trabalho 1

Equipe:

Macilio Ferreira

Tharsis Salathiel

Consulta 1

```
select vr_receita, dt_receita, ds_titulo, cd_titulo, nm_adm, cd_cpf_adm
from receita_candidato
where cd_num_cpf = 4233514;
```

Item a,

Seq Scan on receita_candidato (cost=0.00..109979.98 rows=43 width=86) (actual time=6002.178..11150.639 rows=4 loops=1)

Filter: (cd_num_cpf = 4233514)

Rows Removed by Filter: 2996394

Planning time: 0.230 ms

Execution time: 11150.721 ms

(5 rows)

Tempo da consulta: 11613,799 ms

Item b,

create index index_cpf on receita_candidato using HASH (cd_num_cpf);

CREATE INDEX

Time: 17461,576 ms

Item c,

Bitmap Heap Scan on receita_candidato (cost=4.33..173.73 rows=43 width=86) (actual time=0.034..0.042 rows=4 loops=1)

Recheck Cond: (cd_num_cpf = 4233514)

Heap Blocks: exact=4

-> Bitmap Index Scan on index_cpf (cost=0.00..4.32 rows=43 width=0) (actual time=0.021..0.021 rows=4 loops=1)

Index Cond: (cd_num_cpf = 4233514)

Planning time: 0.095 ms

Execution time: 0.074 ms

(7 rows)

Tempo da consulta: 0,800 ms

Item d,

create index index_cpf on receita_candidato using BTREE (cd_num_cpf);

CREATE INDEX

Time: 16124,436 ms

Item e,

Bitmap Heap Scan on receita_candidato (cost=4.76..174.16 rows=43 width=86) (actual time=0.043..0.053 rows=4 loops=1)

Recheck Cond: (cd_num_cpf = 4233514)

Heap Blocks: exact=4

-> Bitmap Index Scan on index_cpf (cost=0.00..4.75 rows=43 width=0) (actual time=0.029..0.029 rows=4 loops=1)

Index Cond: (cd_num_cpf = 4233514)

Planning time: 0.135 ms

Execution time: 0.094 ms

(7 rows)

Tempo da consulta: 64,169 ms

Item f,

O tempo gasto para a realização da consulta exact-match utilizando o índice HASH é menor do que o tempo gasto pela execução da mesma consulta pelo índice BTREE. Ou seja, podemos afirmar que para este tipo de consulta o índice HASH foi mais rápido do que o índice BTREE.

Consulta 02

```
select vr_receita, dt_receita, ds_titulo, cd_titulo, nm_adm, cd_cpf_adm  
from receita_candidato  
where cd_num_cpf > 4233514;
```

Item a,

Seq Scan on receita_candidato (cost=0.00..107365.98 rows=2995975 width=85) (actual time=39.917..7871.831 rows=2996047 loops=1)

Filter: (cd_num_cpf > 4233514)

Rows Removed by Filter: 351

Total runtime: 9007.790 ms

Tempo da Consulta: 9874,186 ms

Item b,

create index index_cpf on receita_candidato using BTREE (cd_num_cpf);

CREATE INDEX

Time: 23456,647 ms

Item c,

Bitmap Heap Scan on receita_candidato (cost=56087.24..163447.92 rows=2995975 width=85) (actual time=1157.809..8535.078 rows=2996047 loops=1)

Recheck Cond: (cd_num_cpf > 4233514)

Rows Removed by Index Recheck: 251

-> Bitmap Index Scan on index_cpf (cost=0.00..55338.24 rows=2995975 width=0) (actual time=1114.484..1114.484 rows=2996047 loops=1)

Index Cond: (cd_num_cpf > 4233514)

Total runtime: 9683.411 ms

Tempo da Consulta: 10432,239 ms

Item d,

Como pode ser visto no item anterior, o índice foi utilizado para realizar a consulta.

Item e,

create index index_cpf on receita_candidato using HASH (cd_num_cpf);

CREATE INDEX

Time: 4815,000 ms

Item f,

Seq Scan on receita_candidato (cost=0.00..107365.98 rows=2996070 width=85) (actual time=31.506..9098.340 rows=2996047 loops=1)

Filter: (cd_num_cpf > 4233514)

Rows Removed by Filter: 351

Total runtime: 10169.474 ms

Tempo da Consulta: 10716,357 ms

Item g,

O tempo gasto para a realização da consulta utilizando o índice BTREE é menor do que o tempo gasto fazendo a leitura sequencial. O índice HASH em cd_num_cpf criado no interior não foi utilizado, realizamos testes com diferentes valores de dados e “forçamos” o CSGBD a utilizar o índice. No entanto, para todas as entradas de dados somente a leitura sequencial foi realizada. O motivo para isso acontecer é que o índice HASH não permite leituras por intervalos (range), assim para este tipo de consulta o mesmo nunca vai ser utilizado. Executamos uma consulta de exact-match e o CSGBD utilizou o índice HASH criado no item anterior.

Consulta 03

```
SELECT C.cd_num_cpf, C.nm_candidato, RC.nm_doador
FROM candidato C, receita_candidato RC
WHERE C.cd_num_cpf=RC.cd_num_cpf
```

Item a,

Hash Join (cost=14895.51..222426.95 rows=2996398 width=59) (actual time=862.245..13141.358 rows=2996398 loops=1)

Hash Cond: (rc.cd_num_cpf = c.cd_num_cpf)

-> Seq Scan on receita_candidato rc (cost=0.00..102488.98 rows=2996398 width=35) (actual time=34.989..10148.956 rows=2996398 loops=1)

-> Hash (cost=9125.78..9125.78 rows=298378 width=32) (actual time=827.218..827.218 rows=298378 loops=1)

Buckets: 8192 Batches: 8 Memory Usage: 2374kB

-> Seq Scan on candidato c (cost=0.00..9125.78 rows=298378 width=32) (actual time=10.346..701.459 rows=298378 loops=1)

Planning time: 0.129 ms

Execution time: 13253.674 ms

(8 rows)

Tempo da consulta: 14832,251 ms

Item b,

create index index_cpf on receita_candidato using hash (cd_num_cpf);

CREATE INDEX

Time: 7973,784 ms

Item c,

Nested Loop (cost=10000000000.00..10000769788.08 rows=2996398 width=59) (actual time=40.056..163700.527 rows=2996398 loops=1)

-> Seq Scan on candidato c (cost=10000000000.00..10000009125.78 rows=298378 width=32) (actual time=22.346..486.839 rows=298378 loops=1)

-> Index Scan using index_cpf on receita_candidato rc (cost=0.00..2.10 rows=45 width=35) (actual time=0.482..0.543 rows=10 loops=298378)

Index Cond: (cd_num_cpf = c.cd_num_cpf)

Rows Removed by Index Recheck: 0

Planning time: 0.140 ms

Execution time: 164078.351 ms

(7 rows)

Tempo da consulta: 111095,767 ms

Item d,

create index index_cpf2 on candidato using hash (cd_num_cpf);

CREATE INDEX

Time: 587,193 ms

Item e,

Nested Loop (cost=10000000000.00..10000221052.92 rows=2996398 width=59) (actual time=22.886..18597.633 rows=2996398 loops=1)

-> Seq Scan on receita_candidato rc (cost=10000000000.00..10000102488.98 rows=2996398 width=35) (actual time=14.608..8100.862 rows=2996398 loops=1)

-> Index Scan using index_cpf2 on candidato c (cost=0.00..0.03 rows=1 width=32) (actual time=0.002..0.003 rows=1 loops=2996398)

Index Cond: (cd_num_cpf = rc.cd_num_cpf)

Rows Removed by Index Recheck: 0

Planning time: 28.682 ms

Execution time: 18737.018 ms

(7 rows)

Tempo da consulta: 13738,351 ms

Item f,

A consulta realizada no item “e” foi consideravelmente mais rápida do que a consulta do item “c”. Na consulta do item “c” o SGBD utilizou um escaneamento sequencial para a tabela candidato e o index hash para a tabela receita_candidato, e o oposto no item “e”. É importante ressaltar que no item “e”, mesmo com os dois índices criados e o escaneamento desativado o SGBD não utiliza os dois índices. Isso acontece porque a tabela candidato é menor que a tabela receita_candidato, portanto utilizar o índice hash para procurar determinado valor de cpf na tabela candidato é mais rápido do que procurar um valor de cpf na tabela receita_candidato .

Consulta 04

```
select nm_adm, count(distinct vr_receita)
from receita_candidato
group by nm_adm;
```

Item a,

GroupAggregate (cost=637291.47..659890.42 rows=12597 width=32) (actual time=137314.214..171017.997 rows=89894 loops=1)

-> Sort (cost=637291.47..644782.46 rows=2996398 width=32) (actual time=137314.125..160587.308 rows=2996398 loops=1)

Sort Key: nm_adm

Sort Method: external merge Disk: 90336kB

-> Seq Scan on receita_candidato (cost=0.00..99874.98 rows=2996398 width=32) (actual time=37.922..6935.061 rows=2996398 loops=1)

Total runtime: 171113.365 ms

Tempo da Consulta: 172175,170 ms

Item b,

create index nm_adm on receita_candidato using HASH (nm_adm);

CREATE INDEX

Time: 9213,136 ms

Item c,

GroupAggregate (cost=637291.47..659890.42 rows=12597 width=32) (actual time=44832.949..55576.964 rows=89894 loops=1)

-> Sort (cost=637291.47..644782.46 rows=2996398 width=32) (actual time=44832.916..52361.711 rows=2996398 loops=1)

Sort Key: nm_adm

Sort Method: external merge Disk: 90336kB

-> Seq Scan on receita_candidato (cost=0.00..99874.98 rows=2996398 width=32) (actual time=45.193..4804.433 rows=2996398 loops=1)

Total runtime: 55614.347 ms

Tempo da Consulta: 109776,694 ms