



MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
SECRETARIA DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
(Real Academia de Artilharia Fortificação e Desenho - 1792)

BANCO DE DADOS II

TUNING DE ESQUEMA

COMPONENTES DO GRUPO:

Davi Hugo Marques Pontes

Gustavo Alves Casqueiro

Gustavo Atala Testoni

Harllon Oliveira da Paz

Izadora Bandoch

Marcos Paulo Lima Araújo Pereira

Prof^ª. Maria Claudia Reis Cavalcanti

Rio de Janeiro

31 de março de 2021

INTRODUÇÃO

A demanda selecionada foi a de customers que comparam itens de uma determinada cor, no caso a cor antique, pois após consulta às quantidades concluiu-se que é uma das que aparece em maior quantidade:

```
select p_color, count(p_color) from ssb.part group by p_color;
```

	p_color character varying (11)	count bigint
1	almond	2219
2	antique	2857
3	aquamarine	2842
4	azure	2810
5	beige	2899

A escolha dessa consulta foi baseada no fato de que ela passa pelas tabelas customer, part e lineorder, sendo portanto uma das consultas que demandam mais tempo de pesquisa.

DESENVOLVIMENTO

A pesquisa inicial foi realizada através da consulta:

```
select c_name
from ssb.customer
    inner join (select lo_custkey, p_color
from ssb.lineorder
    inner join ssb.part on lo_partkey = p_partkey) b
on lo_custkey = c_custkey
where p_color = 'antique';
```

Cujos tempos de acesso foram:

A frio:

✓ Successfully run. Total query runtime: 237 msec. 28564 rows affected.

E a quente:

✓ Successfully run. Total query runtime: 455 msec. 28564 rows affected.

✓ Successfully run. Total query runtime: 318 msec. 28564 rows affected.

✓ Successfully run. Total query runtime: 364 msec. 28564 rows affected.

✓ Successfully run. Total query runtime: 270 msec. 28564 rows affected.

A tentativa de reduzir o tempo da consulta consistiu em desnormalizar colocando dentro da tabela lineorder os nomes dos customers e as cores correspondentes, de forma que para obter o mesmo resultado não seja mais necessário realizar joins, apenas um select.

Para tanto, foi necessário criar na tabela lineorder as colunas para inserir tais dados:

```
alter table ssb.lineorder add column lo_custurname text;  
  
update ssb.lineorder set lo_custurname = (select c_name from ssb.customer where c_custkey = lo_custkey);  
  
alter table ssb.lineorder add column lo_partcolor text;  
  
update ssb.lineorder set lo_partcolor = (select p_color from ssb.part where p_partkey = lo_partkey);
```

E a consulta fica sendo:

```
select lo_custurname from ssb.lineorder where lo_partcolor = 'azure';
```

Cujos tempos de acesso foram:

A frio:

✓ Successfully run. Total query runtime: 379 msec. 28564 rows affected.

E a quente:

✓ Successfully run. Total query runtime: 280 msec. 28564 rows affected.

✓ Successfully run. Total query runtime: 313 msec. 28564 rows affected.

✓ Successfully run. Total query runtime: 266 msec. 28564 rows affected.

✓ Successfully run. Total query runtime: 250 msec. 28564 rows affected.

CONCLUSÃO

Os tempos obtidos em consultas a frio mostram uma redução aproximada de 10% do tempo da consulta. Essa redução não é mais significativa e também o tempo a quente acabou sendo maior após a alteração, devido ao fato de a tabela lineorder já ser bastante grande - o que torna a consulta a esta tabela bastante custosa - e, com a alteração, ela ter sido distribuída por mais blocos, o que não beneficiou mais expressivamente a consulta em questão, mesmo que não tenha havido mais a necessidade de olhar para as duas outras tabelas (customer e part).

Além disso, deve-se considerar o tempo de atualização da tabela (que nas execuções demorou tempos da ordem de 1 minuto), uma vez que o ganho nas consultas deve compensar o tempo de atualização da lineorder.