

Banco de Dados, Otimização de Consultas

1) Criar uma consulta com join sem índice (caso exista, apagá-los) utilizando duas tabelas (supplies e supplier)

```
EXPLAIN ANALYZE SELECT a.sid, b.sid
FROM supplies as a
FULL OUTER JOIN supplier as b
    on a.sid = b.sid;
```

2) Sobre a mesma consulta, criar índices sobre os atributos utilizados na consulta

```
CREATE INDEX index_um ON supplies(sid);
CREATE INDEX index_dois ON supplier(sid);
```

3) Comparar e reportar o custo de execução com e sem índice usando o comando `EXPLAIN`. A consulta está acionando o índice? Explique. Gerar o plano.

Sempre quando fazemos a primeira consulta, ela leva mais tempo que o normal, isso com os índices, comparado com a consulta que não possui índices.

Mas depois da primeira consulta, os valores se normalizam, e não há diferença grande nas duas consultas. As duas ficam em torno de 0.055 milissegundos.

A consulta não parece estar acionando o índice. Acreditamos que é por causa do tamanho pequeno de linhas da consulta entre as tabelas. Nossa consulta nessas tabelas retorna um máximo de 24 linhas.

Sem index:

```
transactions=# EXPLAIN ANALYZE SELECT a.sid, b.sid FROM supplies as a FULL OUTER JOIN
supplier as b on a.sid = b.sid;
                                QUERY PLAN
-----
Hash Full Join  (cost=1.20..2.52 rows=24 width=8) (actual time=0.029..0.038 rows=24 l
oops=1)
  Hash Cond: (a.sid = b.sid)
    -> Seq Scan on supplies a  (cost=0.00..1.24 rows=24 width=4) (actual time=0.005..0
.006 rows=24 loops=1)
    -> Hash  (cost=1.09..1.09 rows=9 width=4) (actual time=0.012..0.012 rows=9 loops=1
)
      Buckets: 1024  Batches: 1  Memory Usage: 9kB
    -> Seq Scan on supplier b  (cost=0.00..1.09 rows=9 width=4) (actual time=0.008..0.009 rows=9 loops=1)
Planning Time: 0.997 ms
Execution Time: 0.054 ms
(8 rows)
```

Com index:

```
transactions=# EXPLAIN ANALYZE SELECT a.sid, b.sid FROM supplies as a FULL OUTER JOIN
supplier as b on a.sid = b.sid;

                                QUERY PLAN

-----
Hash Full Join  (cost=1.20..2.52 rows=24 width=8) (actual time=0.022..0.030 rows=24 loops=1)
  Hash Cond: (a.sid = b.sid)
    -> Seq Scan on supplies a  (cost=0.00..1.24 rows=24 width=4) (actual time=0.005..0.006 rows=24 loops=1)
    -> Hash  (cost=1.09..1.09 rows=9 width=4) (actual time=0.013..0.013 rows=9 loops=1)
          Buckets: 1024  Batches: 1  Memory Usage: 9kB
          -> Seq Scan on supplier b  (cost=0.00..1.09 rows=9 width=4) (actual time=0.009..0.010 rows=9 loops=1)
  Planning Time: 1.616 ms
  Execution Time: 0.046 ms
(8 rows)
```

4) Qual algoritmo foi usado para gerar o join? Explique seu funcionamento.

O comando utilizado foi o `FULL OUTER JOIN`, que une todos os dados de uma tabela com a outra, sem exceção. Também é conhecido como `FULL JOIN`.

