

## ATIVIDADE

### 1 - Descreva o que está acontecendo com as consultas para gerar tal comportamento.

```
mysql> explain select * from nfsi where nfsi.nr_documento = 160721 and cd_filial = 1;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | nfsi | NULL | ref | xxx | xxx | 10 | const,const | 8 | 100.00 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0,00 sec)

mysql> explain select * from nfsi where nfsi.nr_documento = 160721;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | nfsi | NULL | ALL | NULL | NULL | NULL | NULL | 646 | 10.00 | Using where |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0,00 sec)
```

#### 1ª consulta: explain select \* from nfsi where nfsi.nr\_documento = 160721 and cd\_filial = 1

Segundo a documentação do explain, podemos observar que na primeira consulta, que o type é ref, ou seja, todas as linhas com valor e índice correspondentes serão lidas nessa tabela para cada combinação das tabelas anteriores. Na coluna possible\_keys mostra a quantidade de combinações possíveis, nesse exemplo, são três, podendo usar mais um and para a comparação, como mostrado exemplos em sala, pode se usar a primeira constante, a primeira constante e segunda, e as três juntas, mas a primeira sempre é preciso estar inclusa. No campo ref mostra as constantes usadas para a comparação de índice, assim, selecionando as linhas da tabela. Com a comparação de duas constantes, como mostra no explain, a consulta se torna mais rápida, pois o número de linhas que será percorrido, é significativamente menor.

#### 2ª consulta: explain select \* from nfsi where nfsi.nr\_documento = 160721

Já na segunda consulta, não temos a segunda constante para usar as comparações das linhas, apenas o número de documento, por isso a possibly\_keys é NULL, não é utilizado o and para as comparações, assim percorrendo um número muito maior de linhas, pois a comparação se torna genérica, ou menos eficaz, ao ponto de ser necessário fazer mais verificações para mostrar os dados de determinada coluna ou linha.

**2- O banco de dados MySQL resulta em tempos diferentes para fazer as duas consultas. Já o Postgres, apresenta o mesmo tempo de execução para as duas consultas. Explique o motivo.**

```
teste=# explain select * from nfsi where nfsi.nr_documento = 160721 and cd_filial = 1;
               QUERY PLAN
-----
Index Only Scan using cd_key on nfsi  (cost=0.15..8.17 rows=1 width=12)
    Index Cond: ((cd_filial = 1) AND (nr_documento = 160721))
(2 filas)

teste=# explain select * from nfsi where nfsi.nr_documento = 160721;
               QUERY PLAN
-----
Bitmap Heap Scan on nfsi  (cost=19.45..30.01 rows=10 width=12)
    Recheck Cond: (nr_documento = 160721)
    -> Bitmap Index Scan on cd_key  (cost=0.00..19.45 rows=10 width=0)
        Index Cond: (nr_documento = 160721)
(4 filas)

teste=# █
```

Como visto nas execuções do mysql, o postgres é semelhante, pois quando tem duas constantes para efetuar a pesquisa, se torna mais eficaz, percorrendo apenas as linhas que corresponde a pesquisa, já quando é apenas com uma constante, ele precisa percorrer mais linhas para dar o retorno do que se é desejado.

## REFERÊNCIAS

<https://dev.mysql.com/doc/refman/8.0/en/explain-output.html>

<https://www.postgresql.org/docs/9.3/static/sql-explain.html>