

RENAN HENRIQUE GOMES DAMAZIO ASSUNÇÃO RA 21038114

ALEX ARANTES GONÇALVES RA 21011214

**RELATÓRIO 2**

Tópicos Emergentes em Bancos de Dados

Santo André – SP

2019

**QUESTÃO 1**

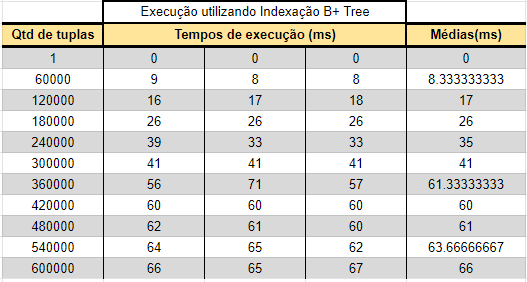
Uma relação contendo 600000 tuplas e 3 atributos foi criada seguindo as seguintes especificações: o primeiro atributo *att1* do tipo *int*, não contendo valores duplicados ou nulos. Os demais atributos também int aleatoriamente escolhidos entre valores 1 e 600000. Um índice do tipo *B+Tree* foi criado apenas para o primeiro atributo

Os scripts utilizados para esta questão encontram-se no arquivo *Q1.txt*. Para todos os testes desta questão foi removido o processamento paralelo os tempos apresentados ignorarem o tempo de planejamento e serem expressos em milissegundos.

Uma query do formato *SELECT \* FROM Rel where att1 between ... and ...;* foi executada aumentando-se progressivamente a quantidade de tuplas em ordem de 10% do total das 600000 tuplas presentes na relação.

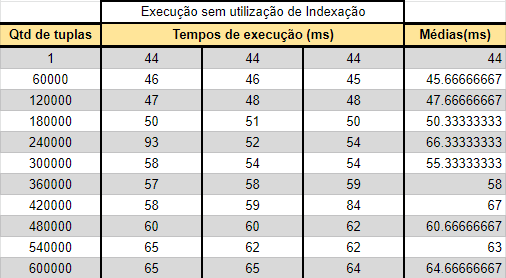
As mesmas queries foram executadas sem que a indexação fosse utilizada.

Cada uma das Queries foi executada três vezes, seus tempos de execução gravados e uma média ponderada foi calculada para elaboração de um gráfico comparativo dos tempos de execuções nas duas condições apresentadas.  
 A tabela 1 mostra os resultados obtidos utilizando-se indexação *B+ Tree*

****

**Tabela 1** - Execução das queries utilizando indexação *B+ Tree*

A Tabela 2 apresenta os resultados obtidos para as execuções sem indexação



**Tabela 2** - Execução das queries sem utilização de indexação

As médias dos valores do experimento foram utilizadas para elaboração do gráfico comparativo - Gráfico 1



**Gráfico 1** - Execução das queries com indexação x sem indexação

Por meio da análise do gráfico é possível fazer algumas conclusões:

1. A quantidade das tuplas afeta o desempenho das execuções indexadas, de modo que o otimizador optou por realizar sequential scan para uma quantidade de tuplas maior ou igual a 360.000.
2. Apesar do desempenho do sequential scan ser inferior para quantidade de tuplas menores que 360000, se comparado ao da indexação *B +Tree*, o tempo de execução sofreu uma menor variação no decorrer das execuções.

**QUESTÃO 2**

Inicialmente decidimo buscar os indices que operavam sobre as relações depto, proj, emp onde foi encontrado um único índice *idx\_depto* do tipo btree que trabalha sobre o campo *coddepto* da tabela *depto.* Todos os scripts utilizados e resultados obtidos seguem no arquivo *Q2.txt*.

As transações apresentadas foram avaliadas levando em conta o tempo de execução, e o plano de execução, gerando os seguines resultados:

* **T1:** Teve tempo médio de execução de 35ms utilizando sequencial scan e não usando o índice *idx\_depto*. Como a consulta retorna em tempo satisfatório e a relação é modificada com muita frequência decidimos não criar nenhum índice para não afetar o tempo tempo das operações de modificação da relação e também porque os benefícios seriam muito baixos nesse momento.
* **T2:** Apresentou tempo médio de execução de 375ms e assim como a anterior fez um sequential scan e não utilizou o índice encontrado. Como das três transações essa foi a que apresente o maior tempo médio, tentamos a criação de índices. O primeiro indice criado foi do tipo btree e o segundo BRIN, porém em ambos os casos os índices não foram utilizados e não houve alteração no tempo médio. Por esse motivo decidimos não criar índices sobre essa relação.
* **T3:** Para realizar teste mais profundos com o uso do betwee primeiro procuramos quantas tuplas haviam na relação depto, obtendo o valor de 500. Em seguida executamos a relação para 5 faixas de coddepto distintas, de 1-100, 1-200, 1-300, 1-400 e 1-500. Apenas a primeira faixa utilizou o índice *idx\_depto*  obtendo tempo de execução de 15ms, as demais utilizaram sequential scan obtendo tempos crescentes de 15ms a 22ms. Devido ao tempo satisfatório de busca e alto indice de atualizações escolhemos nao criar nenhum índice para aperfeiçoar as transações nesta relação.

A análise dos resultados sobre as transações nos leva as seguintes conclusões:

1. Os testes executados mostraram que a criação de índices não ajudariam nas transações avaliadas, por isso decidiu-se não criar nenhum índice
2. Foi encontardo um único índice que poderia contribuir com as trasações analizadas, o *idx\_depto* do tipo btree que trabalha sobre o campo *coddepto* da tabela *depto*, porém não contribui com as transações T1 e T2 e tem baixo impacto em T3. Como ele pode contribuir com outras transações ainda não mencionadas decidimos não altera-lo