

Orientações:

- Deve ser realizado em grupos de **até** duas pessoas;
- Entrega exclusivamente via Tidia4 no formato PDF (Relatório) e TXT (SQL utilizados na confecção do Relatório);
- Plágio e soluções copiadas de outros colegas, implicarão na nota ZERO a todos os envolvidos.

1 Questão (50%) - Análise Geral

Até que momento o otimizador entende que um dado índice \mathcal{I} do tipo B^+Tree é a melhor opção para acessar uma grande relação? O acesso sequencial (*Full Scan Table*) é sempre ruim? Curioso em responder essa pergunta, aplique um experimento para colher dados e demonstrar a partir de qual instante (se é que existe) um índice é pior do que o acesso sequencial.

Para esse experimento, utilize a seguinte configuração:

- Crie uma relação Rel com 600.000 tuplas e três atributos. O primeiro atributo ($att1$) deve ser do tipo *int* e não pode conter duplicados ou nulos. Os demais atributos podem ser de qualquer tipo de sua preferência;
- Crie um índice \mathcal{I} do tipo B^+Tree para a referida relação **somente** sobre o primeiro atributo ($att1$);
- Execute repetidas vezes uma *query* Q com o seguinte formato:

```
SELECT * FROM Rel where att1 between ... and ...;
```

A cada execução, modifique o critério de seleção de modo a aumentar paulatinamente a quantidades de tuplas selecionadas pela condição *between*. Inicie selecionando uma única tupla e, então, siga aumentando em 10% a cada nova execução até alcançar 100% das tuplas da relação Rel . Lembre-se de executar cada critério de execução com e sem o uso do índice \mathcal{I} ;

- Além de observar as informações do plano de execução, não esqueça de observar as estatísticas.

2 Questão (50%) - Análise focada no Projeto

Recentemente você notou um novo padrão de consultas envolvendo a relação *depto*, sendo as massivamente utilizadas elencadas abaixo.

```
T1. select * from depto inner join proj on (depto.coddepto = proj.coddepto);
T2. select * from depto inner join emp on (depto.coddepto = emp.coddepto);
T3. select * from depto inner join proj on (depto.coddepto = proj.coddepto) where
    depto.coddepto between .... and ....;
```

Não estando certo se as configurações atuais do *schema* atendem a esse novo padrão, você decidiu realizar uma análise de desempenho para responder questões como:

- Existem índices que podem ser utilizados e que não estão presentes no *schema* atual? Caso afirmativo, qual é o ganho de desempenho e o impacto (espaço, etc.) desses índices?
- Existem índices que não estão contribuindo para as transações acima?

Seu monitoramento determinou que a relação *emp* é moderadamente atualiza, mas raramente um empregado muda de departamento. Por outro lado, a relação *proj* possui um alto índice de atualizações e, dentre elas, estão a transferência de projetos entre departamentos.

Baseado em sua análise e nas respostas das questões acima, aplique os ajustes necessários ao seu modelo de referência.

Boa análise!