

CK0117 - Sistemas de Bancos de Dados - 2021-1

Javam Machado; Edvar Filho; Malu Maia

TRABALHO III - Operadores Algébricos

1 Aspectos Gerais

O trabalho consiste em implementar operações básicas existentes em um SGBD convencional: **seleção**, **projeção** e **junção** (descritos na seção 2). Na implementação, desconsidere o uso de SGBDs. Use um índice *Hash* nas operações em que a sua utilização for necessária, procure uma lib que já a implementa na linguagem escolhida.

Um banco de dados será mapeado na memória, onde cada tabela terá um conjunto de páginas sem quantidade definida, cada página terá um conjunto de 16 registros e cada registro terá como base uma estrutura que definirá o tipo da tabela, sendo do tipo Funcionário caso precise apenas de uma tabela para efetuar a operação ou duas tabelas, um do tipo Funcionário e outro do tipo Venda, caso a operação a ser implementada seja **Junção** como mostrado na imagem abaixo.



Figure 1: Abstração do Banco de Dados em Memória

Se necessário, utilize páginas temporárias para implementar o operador. Serão enviados arquivos com os registros para teste. Não serão os mesmos registros na avaliação. O resultado da operação deve conter as **tuplas retornadas da operação**, mas as tabelas originais não devem ser modificadas.

2 Lista de Trabalhos

Cada equipe fará o trabalho designado conforme a lista abaixo.

Edealcia Regina – *Nested Loop Join (NLJ)*: Junção com laço aninhado orientado a página - Algoritmo usa três frames de memória, uma para as páginas da tabela *outer*, um para as páginas da tabela *Inner* e um frame para a saída. O programa deve gravar um arquivo com a tabela de resultado.

Lucas Gouveia – *Index Selection (IS)*: Seleção sobre uma tabela utilizando um índice. Para tal considere um predicado de seleção com igualdade. Você deve usar a sua implementação de índice *hash* para fazer esse trabalho.

Alan/René – *Distinct Hash Projection (DHP)*: Projeção com eliminação de duplicatas construindo as partições (*buckets*). Você deve utilizar uma implementação de índice *hash* no trabalho, como definido na seção 1.

João Esau – *Nested Loop Join (NLJ)*: Junção com laço aninhado orientado a página - Algoritmo usa três frames de memória, uma para as páginas da tabela *outer*, um para as páginas da tabela *Inner* e um frame para a saída. O programa deve gravar um arquivo com a tabela de resultado.

Lucas Assis/George – *Nested Loop Block Join (NLBJ)*: Junção com laço aninhado orientado a bloco - permitir a variação do número de frames de memória para a tabela *outer* a fim de testar o programa.

Gabriel/Mateus – *Nested Loop Index Join (NLIJ)*: Junção com laço aninhado utilizando índice. Você deve utilizar uma implementação de índice *hash* no trabalho, como definido na seção 1.

Evilásio/Luiz Felipe – *Sort Merge Join (SMJ)*: Junção utilizando a ordenação externa das tabelas. Grave as tabelas ordenadas no espaço de memória que simula um disco antes de fazer a etapa de comparação

Daniel/Daniel – *Hash Join (HJ)*: Junção utilizando a criação de partições (*buckets*) das tabelas. Grave as tabelas particionadas no espaço de memória que simula um disco antes de fazer a partição.

Raif – *Nested Loop Join (NLJ)*: Junção com laço aninhado orientado a página - Algoritmo usa três frames de memória, uma para as páginas da tabela *outer*, um para as páginas da tabela *Inner* e um frame para a saída. O programa deve gravar um arquivo com a tabela de resultado.

3 Implementação e Avaliação

- As implementações dos trabalhos deverão ser feitas em Java ou C++;
- A avaliação será composta pela apresentação realizada, verificando se os alunos possuem domínio do conteúdo implementado, além da funcionalidade do sistema realizado que verificaremos a lógica implementada e o resultado obtido.

Data da entrega: Quinta-feira - 22 de julho de 2021 até a 23:59, suscetível à penalidade caso haja atraso. Apresentação do trabalho na aula da sexta-feira 23 de julho de 2021.