

# CK0117 - Sistemas de Bancos de Dados - 2022-1

Javam Machado, Daniel Praciano, Edvar Filho, Paulo Amora

## TRABALHO II - Operadores Algébricos

### 1 Aspectos Gerais

O trabalho consiste em implementar as operações básicas existentes em um SGBD relacional: **seleção**, **projeção** e **junção** (descritos na seção 2). Na implementação, desconsidere o uso de SGBDs. Use um índice nas operações em que a sua utilização for necessária, ou seja, a equipe deve utilizar o índice implementado no trabalho anterior.

Para executar os operadores implementados, vamos utilizar o esquema apresentado na Figura 1. Isto é, todos os operadores receberão como entrada uma (ou mais) tabela presente nesse esquema, dependendo se o operador é unário ou binário. Mais detalhes de implementação serão descritos na seção 3.

Resumidamente, observa-se, a partir do esquema apresentado, que existe uma tabela *Vinho*, que possui como chave primária (PK) *vinho\_id* e duas chaves estrangeiras (FK): *uva\_id* que referencia o atributo de mesmo nome da tabela *Uva* e *pais\_producao\_id* que referencia *pais\_id* da tabela *País*. Ademais, essa mesma situação ocorre entre as tabelas *Uva* e *País*.

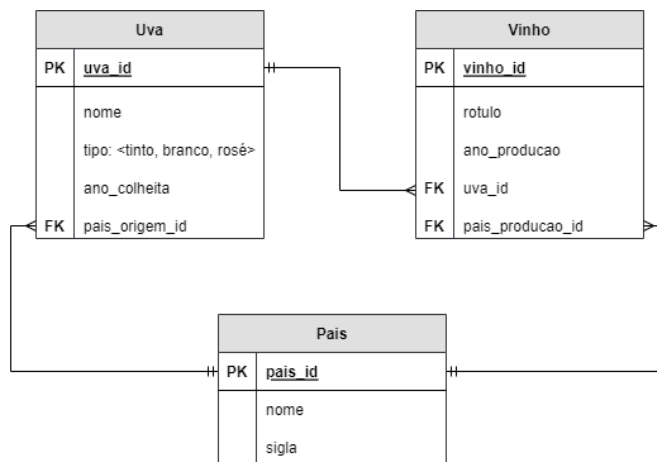


Figure 1: Modelagem E-R do esquema a ser utilizado.

### 2 Lista de Operadores

Cada equipe de, *no máximo*, dois alunos implementará **um e somente um** dos seguintes operadores de maneira **exclusiva**. Além disso, o resultado do operador deve conter as **tuplas retornadas da operação**, mas as tabelas originais **não devem ser modificadas**.

**Tree Index Selection (TIS):** Seleção sobre uma tabela utilizando uma árvore B+. Para tal considere um predicado de seleção com igualdade.

**Distinct Sort Projection (DSP):** Projeção com eliminação de duplicatas realizando a ordenação. Para tal você precisa realizar a ordenação externa da relação.

**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.

**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.

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

**Sort Merge Join (SMJ):** Junção utilizando a ordenação externa das tabelas. Grave as tabelas ordenadas em arquivos do disco antes de fazer a etapa de comparação.

**Hash Join (HJ):** Junção utilizando a criação de partições (*buckets*) das tabelas. Grave as tabelas particionadas em arquivos do disco antes de fazer a partição.

### 3 Implementação

Nesse trabalho, um banco de dados será mapeado em **disco** de modo que cada tabela terá um conjunto de páginas sem quantidade definida, sendo que cada página terá um conjunto de 12 tuplas e cada tupla terá como base uma estrutura adjacente baseada no esquema da tabela. A implementação dessas abstrações deverá utilizar as classes propostas na figura abaixo. **Todas essas estruturas precisam estar armazenadas em arquivos do disco em formato de texto.** Portanto, uma maneira de como realizar o mapeamento dessas estruturas em arquivos deve ser definida.

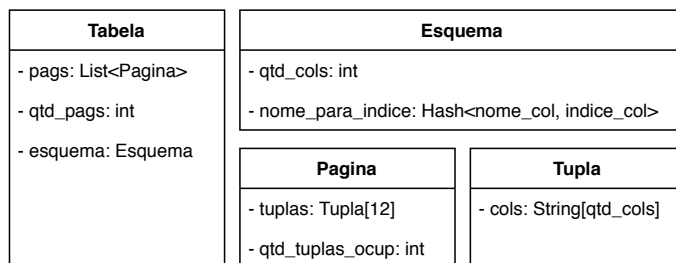


Figure 2: Sugestão de implementação por meio do diagrama de classes.

Além disso, será necessário manter **ATÉ 3 páginas, no total, em memória** (se NLBJ, considere ATÉ quantidade de *frames*) e utilizar o arquivo **Main** da respectiva linguagem na sua implementação. Esse arquivo contém uma interface que deverá ser implementada, juntamente com alguns casos de testes para validar a implementação.

### 4 Entrega

**Data da entrega:** Quinta-feira - 19 de maio de 2022 até 23h59m com apresentação e arguição no LEC/DC no dia seguinte, 20 de maio, no horário da aula. O código do trabalho deve ser enviado no **classroom** até o final do horário da entrega. Envios posteriores serão penalizados. Quaisquer dúvidas podem ser enviadas aos monitores: Daniel Praciano (daniel.praciano@lsbd.ufc.br) ou Edvar Filho (edvar.filho@lsbd.ufc.br).