CK0117_CK0446 – Sistemas de Bancos de Dados - 2024-1

Javam Machado, Eduardo Duarte, Malu Maia

TRABALHO I - ÍNDICES HASHING

1 Aspectos Gerais

Um banco de dados armazenado em disco possui uma relação que salva as compras realizadas por um consumidor. A Figura 1 apresenta o esquema da relação **Compras**. Normalmente, a carga de trabalho que envolve essa relação realiza uma consulta considerando o ano no qual a compra foi realizada a fim de se calcular o total de vendas do ano. Assim, o trabalho consiste em implementar um índice para essa relação, sendo que a chave de busca é o atributo **ano**, que é um número inteiro.

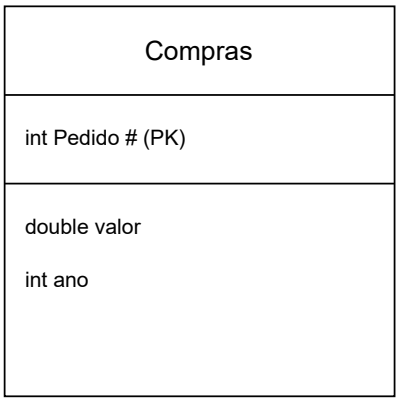

Figure 1: Relação a ser indexada.

Para esse trabalho, deverão ser utilizados os dados presentes na instância da relação **Compras** que está no arquivo **compras.csv** que será publicado junto a esse documento.

Cada equipe de, *no máximo*, dois alunos implementará um índice hash extensível, bastante utilizado e visto em aula. O índice deve considerar que a relação **Compras** está armazenada em um arquivo do disco no qual cada linha (registro) desse arquivo é uma página da tabela no banco de dados.

Os alunos devem implementar as operações de busca (por igualdade), inserção e remoção de entradas de dados.

2 Detalhamento

Nesse trabalho, um banco de dados será mapeado em **disco** sendo representado por: arquivos de dados, *buckets*, e um arquivo ou diretório contendo o índice. A Figura 2 contém a representação do banco. Nela voce pode identificar o "Arquivo de texto" que contem a tabela **Compras**. Para facilitar a implementação, um registro deste arquivo representa uma página da tabela. Por exemplo, se a tabela **Compras** tiver 4 páginas, isto será representado por meio de 4 registros no "Arquivo de texto".

Ainda na Figura 2 voce pode identificar a estrutura de um diretório. Ele contem uma profundidade global – variável PG iniciada com valor 3 – e uma sequência de referências

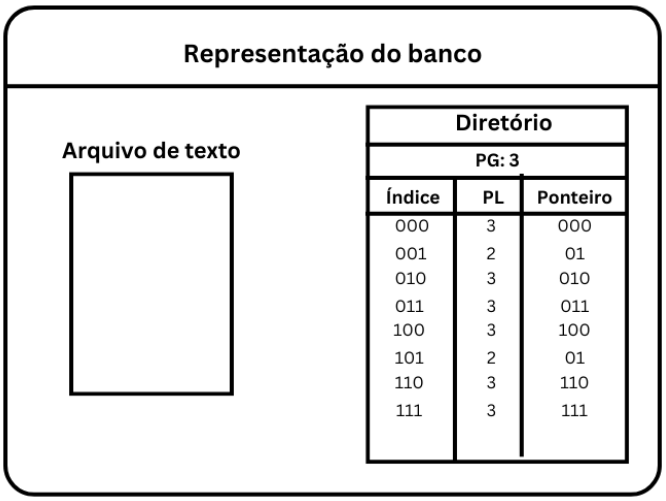

Figure 2: Representação do Banco

para os *buckets*. Cada entrada dessa sequência contem uma lista de três variáveis. O "Indice" simula os bits menos significativos que resultariam de uma função hash sobre o valor da chave de busca; o Ponteiro é a referência para um *bucket*; e o PL é o profundidade local do *bucket* referenciado.

A Figura 3 descreve a área de páginas primárias do índice hash. Cada *bucket* será representado por um arquivo de texto, onde cada linha do arquivo de texto representa uma entrada de dados do *bucket*. Um *bucket* terá no máximo 3 entradas de dados, ou seja, cada arquivo correspondendo a um *bucket* só armazena no máximo 3 registros.

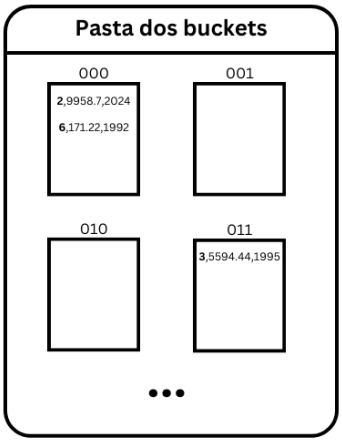

Figure 3: Buckets

Observações importantes:

- A implementação deverá ser feita somente nas linguagens **C**, **C++** ou **Java**. Nada além disso!
- Somente uma única página de dados deve estar em memória por vez. Ou seja, não é permitido manter todos os dados em memória simultaneamente. Única exceção para essa restrição: diretório do hash.
- Deverá existir um arquivo **Main** que será responsável por ler um arquivo de texto com as informações de entrada, bem como gerar um arquivo de texto com as informações de saída. Mais informações a seguir.

O arquivo **Main** será o responsável por realizar chamadas as funcionalidades implementadas. Para tanto, um único arquivo de entrada **in.txt** será fornecido. Nesse arquivo de entrada, estarão as operações de índice que devem ser realizadas, sendo que cada linha desse arquivo é composta por uma das alternativas seguintes:

```
INC:x
REM:x
BUS=:x
```

De maneira intuitiva, INC, REM, BUSC=, representam, respectivamente, as operações de inclusão, remoção e busca por igualdade. Ademais, em todas as operações, **x** representa o inteiro a ser usado nas respectivas operações.

A primeira linha do arquivo de entrada, antes das operações em si, irá indicar a profundidade global do hash extensível. A sintaxe dessa linha é a seguinte:

```
PG/<profundidade global inicial>
```

Por fim, um arquivo de saída nomeado de **out.txt** deve ser gerado. A linha inicial do arquivo deve ser igual à primeira linha do arquivo de entrada e, após cada operação do arquivo de entrada ser realizada, deve ser incluída uma linha no arquivo de saída da seguinte forma:

```
INC:x/<profundidade global>,<profundidade local>
REM:x/<qtd de tuplas removidas>,<profundidade global>,<profundidade local>
BUS:x/<quantidade de tuplas selecionadas>
```

No arquivo do índice em hash extensível, deve-se adicionar a seguinte linha sempre logo após a inclusão de um elemento que duplica o diretório: DUP_DIR:/<profundidade global>,<profundidade local>. Além disso, a última linha do arquivo de saída deve ser P:/<profundidade global final>.

3 Entrega

Data da entrega: Sexta-feira - 26 de abril de 2024 até 09h50m com apresentação e arguição no LEC/DC no mesmo dia, 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: Eduardo Duarte ou Malu Maia via **classroom**.