

Exercício de Laboratório

Extração Aleatória de Dados

Objetivo:

Neste exercício, você irá implementar um programa que realiza a extração aleatória de dados a partir de um arquivo de CEPs não ordenado, previamente utilizado em exercícios anteriores. O objetivo é criar um novo arquivo que contenha aproximadamente 80% dos registros do arquivo original. A seleção dos registros será feita de forma aleatória, com base em um número gerado para cada linha.

Instruções:

1. **Entrada de Dados:** Utilizar o arquivo de CEPs não ordenados.
2. **Processamento:**
 - Seu programa deve ler o arquivo de entrada registro por registro.
 - Para cada registro, gere um número aleatório entre 0 e 1.
 - Se o número gerado for menor que 0.8, a linha deverá ser copiada para um novo arquivo.
 - Se o número gerado for maior ou igual a 0.8, a linha deverá ser descartada.
3. **Saída de Dados:** O programa deve criar um novo arquivo de saída contendo aproximadamente 80% dos registros do arquivo original. O resultado final não precisa ter exatamente 80% dos registros originais, mas deve se aproximar dessa proporção.

Implementação da Operação de Interseção com HashJoin

Objetivo:

Neste exercício, você vai implementar a operação de interseção entre dois arquivos que contêm 80% dos registros dos arquivos originais. Para isso, utilizaremos uma variação

simplificada do algoritmo HashJoin, que é uma técnica eficiente para operações de junção e interseção em grandes volumes de dados.

Instruções:

1. Entrada de Dados:

- Você terá dois arquivos, cada um contendo 80% dos registros de seus respectivos arquivos originais. Esses registros são resultados do exercício anterior.
- Cada linha desses arquivos representa uma chave única (por exemplo, um CEP).

2. Processamento:

• Fase 1 - Criação da Tabela de Hash:

- Leia o primeiro arquivo e crie uma tabela de hash.
- Utilize o método de hash em blocos, onde cada posição da tabela de hash pode armazenar múltiplos registros (por exemplo, 1000 registros por bloco).
- Para cada chave do primeiro arquivo, aplique uma função de hash para determinar a posição do bloco onde a chave será armazenada.
- Insira a chave no bloco correspondente.

• Fase 2 - Verificação e Interseção:

- Leia o segundo arquivo e, para cada chave nele contida, aplique a mesma função de hash utilizada para o primeiro arquivo.
- Verifique se a chave do segundo arquivo existe na tabela de hash criada a partir do primeiro arquivo.
- Se a chave estiver presente na tabela de hash, adicione o registro no arquivo de saída.

3. Saída de Dados:

- Crie um novo arquivo de saída que contenha os registros cujas chaves que estão presentes em ambos os arquivos, ou seja, o resultado da operação de interseção.
- O arquivo final deve conter apenas registros cujas chaves sejam comuns aos dois arquivos.