

Instruções

- Coloque apenas a sua MATRÍCULA na folha resposta.
- Aparelhos eletrônicos desligados (ou no modo silencioso).
- As questões podem ser resolvidas a lápis porém o professor se reserva a não aceitar reclamações oriundas da correção das questões.
- Consulta permitida apenas a cola oficial.

Avaliação A1

Matrícula: 20230001178

1. Em relação ao conceito de índices em banco de dados, analise as seguintes afirmativas:

- I. Índice denso é caracterizado por conter todos os valores da chave do índice. ✓
- II. Os índices baseados em árvores são exemplos de uso de índice denso. ✗
- III. Índice esparsos são caracterizados por não conter todos os valores da chave do índice. ✓
- IV. Índices bitmap são muito utilizados quando a chave do índice não possui repetição. ✗

Escolha abaixo a alternativa mais coerente em relação às afirmativas:

- Apenas a alternativa I é correta.
- I, II e III estão corretas.
- III e IV estão corretas.
- Nenhuma das alternativas está correta.

MRU	LRU
✓ 1-15	1-15
1=5 2=5	1=5 16+20

2. Imagine um buffer com 15 páginas disponíveis para executar uma consulta. Uma consulta recupera 2 vezes a tabela inteira ocupando 20 páginas cada consulta. Abaixo são dadas algumas afirmativas sobre as políticas de troca de página de buffer: MRU u LRU.

- I. MRU: tanto a primeira carga quanto a segunda farão 5 trocas. Totalizando 10 trocas.
- II. LRU: a primeira carga fará 5 trocas e a segunda fará 20 trocas. Totalizando 25 trocas.
- III. Ambas, MRU e LRU, farão 10 trocas.
- IV. LRU: a primeira carga fará 5 trocas e a segunda 10. Totalizando 15

Escolha abaixo a alternativa mais coerente em relação às afirmativas:

- Apenas a alternativa I é correta.
- Apenas III está correta.
- I e II estão corretas.
- Nenhuma das alternativas está correta.

MRU: 1-15	LRU 1-15
16-20-15	1
5	2
10+10T = 20T	3
	4
	5
	6

Revisão

3. Uma empresa quer rastrear todas as atualizações de salário que ocorrem na tabela *employee*. A cada atualização nessa tabela, uma tupla deve ser inserida na tabela *aud_employee*. Os esquemas são: *employee(id integer, name varchar(40), address varchar(60), email varchar(40), salary numeric(10,2))* e *aud_employee(idemp integer, user varchar(30), date_time timestamp, old_sal numeric(10,2), new_sal numeric(10,2))*. Onde (i) *idemp* armazena o *id* do funcionário que teve o salário alterado, (ii) *user* armazena o nome do usuário que realizou a alteração (variável *current_user* armazena o usuário que se conectou ao banco), (iii) *date_time* armazena a data e hora da operação (função *now()* retorna esses

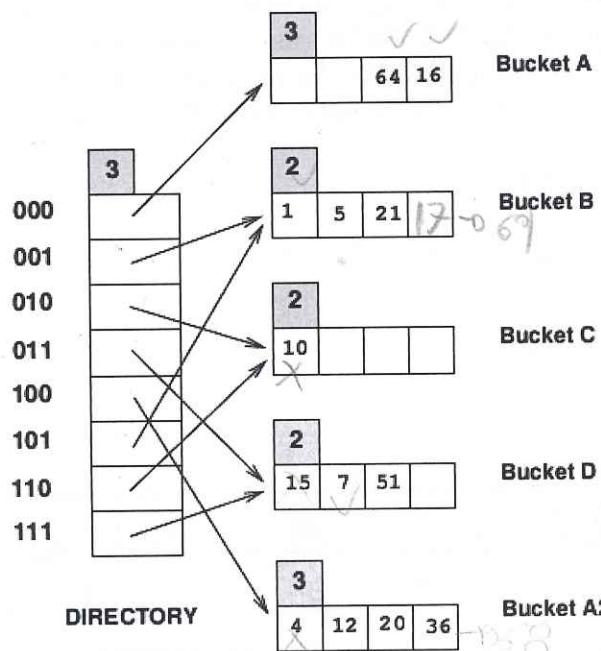
valores), e (iv) *old_sal* e *new_sal* armazenam, respectivamente, o salário antigo e o novo do funcionário. Com essas informações, crie uma *trigger* para realizar o restreamento desejado.

```
CREATE FUNCTION sal-las ()  
RETURNS TRIGGER AS $$sallos$$  
DECLARE  
    O_idemp INTEGER,  
    O_user VARCHAR(30),  
    O_date-time TIMESTAMP,  
    O_old_sal NUMERIC(10,2),  
    O_new_sal NUMERIC(10,2);  
    cr-emp CURSOR;  
BEGIN  
    START CURSOR cr-emp  
        ON Employee(id=NEW.id)  
        O_idemp=NEW.id;  
        O-user=NEW.nome;  
        O-date-time=now();  
        O-old-sal=OLD.Salary;  
        O-new-sal=NEW.Salary;  
    END;  
    INSERT INTO aud_employee VALUES (  
        O_idemp,  
        O-user,  
        O-date-time,  
        O-old-sal,  
        O-new-sal  
    );  
END; $sallos$ language plpgsql;  
CREATE TRIGGER att-sal AFTER UPDATE  
ON employee FOR EACH STATEMENT  
EXECUTE PROCEDURE sal-las; before
```

4. Dado o arquivo de índice extensível abaixo (usando os bits de menor ordem):

pt

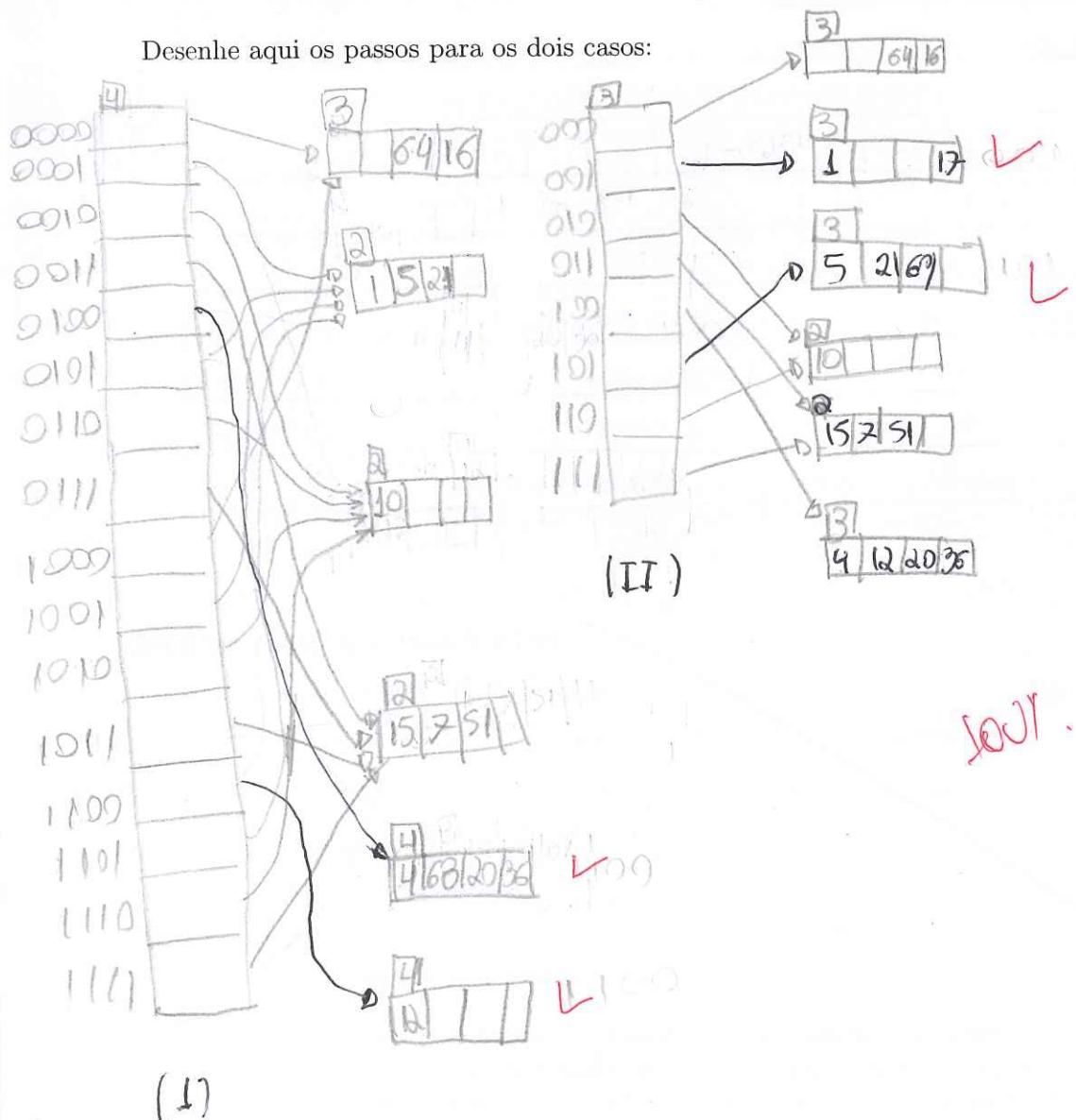
2



Cola: (64,0000), (16,0000), (1,0001), (17,0001),
 (10,1010), (15,1111), (7,0111), (51,0011),
 (4,0100), (12,1100), (20,0100), (36,0100),
 (5,0101), (21,0101), e (69,0101).

- (i) Apresente como índice ficará com a inserção da chave *hashing* 68(01000100), e
 (ii) Sem considerar as inserções anteriores apresente o índice com as inserções de 17
 (00010001) e 69 (01000101) (no original).

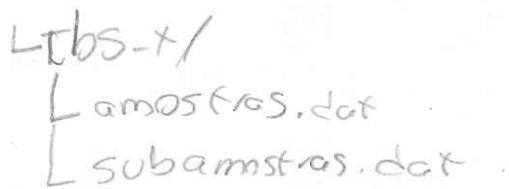
Desenhe aqui os passos para os dois casos:



- 2** 5. Um usuário quer criar um banco de dados para gerenciar as seguintes informações de suas amostras coletadas em seu laboratório: código da amostra, classe e descrição da amostra coletada, número de subamostras, peso de cada subamostra. Projete o banco de dados (um ou mais arquivos) para o armazenamento solicitado. O usuário não quer utilizar um SGBD. Ao projetar o BD, apresente em forma de pseudo-código como serão feitas as consultas, exclusões e inserções no BD. Exemplos de duas amostras:

pt

Código: Amostra Y	Código: Caldíneo
Descrição: Y12	Descrição: MH12
Quantidade: 4	Quantidade: 3
Peso 1: 0.02	Peso 1: 1.5
Peso 2: 0.01	Peso 2: 0.5
Peso 3: 0.09	Peso 3: 0.7
Peso 4: 0.1	



amosta(cod varchar(10), classe varchar(10), descr TEXTUAL)
subamostra(FK_cod varchar(10), peso Numeric(5,3))

Consultas: consultas de maneira sequencial a partir do início do arquivo Amostras.dat, a quantidade é sempre com zero e só incrementada fazendo uma nova consulta no arquivo Subamostras.dat sempre que cod = FK_cod, imprimir subamostra.peso e Quantidade++.

fol. Exclusões: consulta de maneira sequencial, quando query atributo = amostra.atributo ou query.atributo = subamostra.atributo, excluir o registro e ajustar o endereço que aponta para o último registro da tupla anterior para que seja o endereço da próxima tupla ao registro excluído.

Inserções: percorrer o arquivo até o final e inserir a tupla.

BOA PROVA & BOA SORTE.