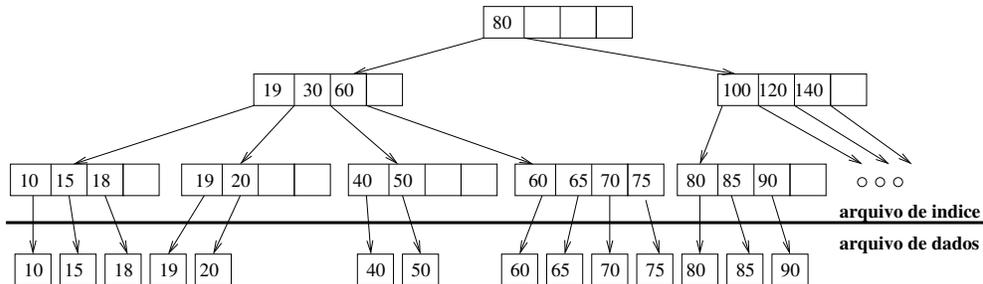


Sistemas de Bancos de Dados

Lista de Exercícios

Processamento de Consultas, Otimização, Controle de Concorrência e Recuperação

Ex 1: Considere a árvore B+ abaixo. Quantas páginas serão lidas do disco para responder a cada uma das seguintes consultas. Assuma que o buffer pool está vazio antes da execução de cada consulta, e que ele é grande o suficiente para conter todo o banco de dados. Assuma também que 4 tuplas da relação base cabem em uma página no arquivo de dados e que todas as páginas estão cheias:



- 1.1 Consultar todas as tuplas com valor de chave de pesquisa de 40 se o índice for agrupado (clustered).
- 1.2 Consultar todas as tuplas com valor de chave de pesquisa de 40 se o índice for não agrupado (unclustered).
- 1.3 Consultar todas as tuplas com valor de chave de pesquisa no intervalo [60, 90] se o índice for agrupado (clustered).
- 1.4 Consultar todas as tuplas com valor de chave de pesquisa no intervalo [60, 90] se o índice for não agrupado (unclustered). Assuma o pior cenário em termos de como as chaves são armazenadas nas páginas.

Ex 2: Considere as relações R e S e um buffer de 11 páginas. O arquivo que contém R tem 40.000 páginas, com 10 registros em cada página e a relação S é armazenada em um arquivo de 10.000 páginas (com 5 registros em cada página).

- 2.1 Qual o custo de $R \bowtie S$ utilizando *sort-merge join*?
- 2.2 Qual o custo de $R \bowtie S$ utilizando *blocked nested loop*?

Ex. 3: Considere as relações $R(a, b, c)$, $S(d, e, f, g)$ e $T(h, i, j)$, com as seguintes características:

Relação R:

- $NTuplas(R) = 10.000$ $NPaginas(R) = 1.000$
- Índice (I_a) B+ clusterizado sobre o atributo a :
 - $IAltura(I_a) = 3$; $IMenor(I_a) = 0$; $IMaior(I_a) = 9000$
- Quantidade de valores distintos do atributo b : 100

Relação S:

- $NTuplas(S) = 10.000$ $NPaginas(S) = 1.000$
- Índice B+ ($I_{e,f}$) não clusterizado sobre os atributos (e, f):
 - $IAltura(I_{e,f}) = 2$; $NChaves(I_{e,f}) = 1000$

Relação T:

- $NTuplas(T) = 10.000$ $NPaginas(T) = 1.000$
- Índice hash sobre o atributo h

3.1 Qual a seletividade das expressões abaixo?

- $a > 100$ and $a \leq 1000$
- $e = 10$ and $f = 1000$
- $b = 100$ or $b = 101$

3.2 Qual a plano de execução gerado se as seleções e projeções forem empurrados “para baixo” na árvore?

3.3 Considerando distribuição uniforme de valores e que todos os atributos das relações tem o mesmo tamanho, qual o custo de execução do plano apresentado na questão anterior?

Ex. 4: Considere uma base de dados com os objetos X, Y e Z e suponha que há duas transações, $T1$ e $T2$. A transação $T1$

1. lê os objetos X e Y
2. escreve em X

A transação $T2$:

1. lê os objetos X e Y
2. escreve no objeto Y
3. lê novamente os objetos X e Y
4. escreve em X
5. lê Z
6. escreve em Z

Forneça três exemplos de escalonamentos para as transações $T1$ e $T2$ para ilustrar cada uma das situações abaixo:

- Seu escalonamento deve conter um conflito de escrita-leitura que cause uma leitura suja em uma das transações.
- Seu escalonamento deve conter um conflito de leitura-escrita que cause uma leitura não repetível em uma das transações.
- Seu escalonamento deve conter um conflito de escrita-escrita que cause uma atualização perdida.

Em cada caso, seu escalonamento pode conter conflitos adicionais, mas deve conter pelo menos um conflito do tipo indicado. Em particular, você pode fornecer um único escalonamento que ilustre todos os três conflitos!. Indique no escalonamento o tipo de conflito ilustrado.

Ex. 5 O escalonamento abaixo é serializável por conflito? Justifique.

Tempo	T_1	T_2	T_3
1	read(a)		
2	write(a)		
3			read(a)
4			write(a)
5		read(a)	
6	read(b)		
7			read(b)
8	write(b)		
9			write(b)
10		read(b)	
11		commit	
12	commit		
13			commit

Ex 6: Considere o escalonamento abaixo, no qual $TS(T_1) < TS(T_2) < TS(T_3)$.

Obs: $TS(T_i)$ representa o timestamp da transação T_i .

Tempo	T_1	T_2	T_3
1	begin(T_1)		
2		begin(T_2)	
3		read(a)	
4			begin(T_3)
5			read(a)
6			write(a)
7		read(b)	
8		write(b)	
9	read(b)		commit(T_3)
10	commit(T_1)		
11		commit(T_2)	

- a) Este escalonamento é possível utilizando o protocolo de controle de concorrência de ordenação por *timestamp*? Por que?
- b) Este escalonamento é possível utilizando o protocolo de bloqueio de duas fases (2PL)? Por que?
- c) O escalonamento é recuperável? Por que?

Ex 7: Explique como funciona e compare as estratégias de prevenção de deadlock esperar-morrer e ferir-esperar.

Ex 8: Considere o escalonamento abaixo.

Linha	T1	T2
1	begin;	
2		begin;
3		select * from venda;
4	update venda set preco= (3*preco) where cerveja='Brahma';	
5	update venda set preco=(3*preco) where cerveja='Stella';	
6	commit;	
7		select * from venda;
8		commit;
9		select * from venda;

Considere que a tabela **venda** inicialmente contém as seguintes tuplas:

cerveja	preco
Brahma	2
Stella	3

8.a) Se o nível de isolamento de T1 e T2 for **read committed**, o que será impresso nos comandos **select** das **linhas 3, 7 e 9**?

8.b) Se o nível de isolamento de T1 e T2 for **repeatable read**, o que será impresso nos comandos **select** das **linhas 3, 7 e 9**?

8.c) Considere que o banco de dados utiliza a estratégia de **modificação imediata**. Suponha que o comando **commit** da Linha 6 seja trocado por **rollback**. Considerando as Linhas 1 a 6, qual é o conteúdo

do **log** após a execução da operação de recuperação `undo(T1)`? (10 pontos)

Ex 9: Considere que o arquivo de log com o conteúdo abaixo.

```
<T1, start>
<T1, A, 10, 15>
<T2, start>
<T3, start>
<checkpoint, [T1, T2, T3]>
<T1, B, 20, 30>
<T4, start>
<T3, C, 10, 15>
<T1, commit>
<T2, B, 30, 35>
<T3, C, 10>
<T4, D, 12, 20>
<T3, abort>
<T3, C, 10, 20>
-- Falha --
```

Apresente o conteúdo do log após a recuperação da falha no momento indicado.