ACH 2025 - Laboratório de Bases de Dados

Lista de Exercícios

Profa. Dra. Sarajane Marques Peres
EACH-USP

1. (Baseado em Navathe) Considere o esquema e as consultas abaixo:

   Empregado (ssn, pnome, mnome, unome, datanasc, endereço, sexo, salário, superssn, dno)
   Departamento (dnome, dnumero, gerssn, derdatainicio)
   Dept_localizaçoes (dnumero, dlocalização)
   Projeto (pjnome, pnumero, plocalizacao, dnum)
   Trabalha_em (essn, pno, horas)
   Dependente (essn, nome_dependente, sexo, datanasc, parentesco)

   a. SELECT pnome, unome, endereço FROM empregado, departamento WHERE dnome = 'pesquisa' AND dnumero = dno;

   b. SELECT e.pnome, e.unome, s.pnome, s.unome FROM empregado e, empregado s WHERE e.superssn = s.ssn;

   c. SELECT e.pnome, e.nome, e.endereço FROM empregado e, departamento d WHERE d.nome = 'pesquisa' AND d.numero = e.numero;

   d. (SELECT DISTINCT pnumero FROM projeto, departamento, empregado WHERE dnum = dnumero AND gerssn = ssn AND unome = 'Smith') UNION (SELECT DISTINCT pnumero, projeto, trabalha_em, empregado WHERE pnumero=pno AND essn = ssn AND unome='Smith');

   e. SELECT pnumero, pnome, COUNT(*) FROM projeto, trabalha_em, empregado WHERE pnumero=pno AND SSN = ESSN AND DNO = 5 GROUP BY pnumero, pnome.

   I. Transforme cada uma das consultas expressas em SQL em consultas expressas em Álgebra Relacional.
   II. Projete duas árvores de consulta que podem representar cada uma dessas consultas. Aponte as regras de transformação (regras de equivalência) aplicadas para provar que uma árvore é equivalente à outra. Sob quais circunstâncias você utilizaria cada uma de suas árvores de consulta?
   III. Desenhe uma árvore de consulta inicial (não otimizada) para cada uma dessas consultas e mostre como a árvore é otimizada usando o algoritmo de otimização baseado em heurística.

2. Marque cada uma das regras de transformação abaixo como (V) válidas ou (I) inválidas, se necessário defina sob que circunstâncias a regra é válida. Justifique a sua resposta por meio da criação de instâncias de relações.

Obs.: Exercite sua capacidade de raciocínio, evite olhar as regras expostas nos livros texto. A intenção é que você saiba identificar, logicamente, se elas valem ou não.

a. ( ) $\sigma_{c1}(\sigma_{c2}(R)) \equiv \sigma_{c2}(\sigma_{c1}(R))$
b. ( ) $\pi_L(R \cap S) \equiv (\pi_L(R)) \cap (\pi_L(S))$
c. ( ) $\sigma_c(R \times S) \equiv (R \,|\times|\, S)$
d. ( ) $\pi_{Lista1}(\pi_{Lista2}(R)) \equiv \pi_{Lista2}(\pi_{Lista1}(R))$
e. ( ) $\pi_L(R \cup S) \equiv (\pi_L(R)) \cup (\pi_L(S))$
f. ( ) $\sigma_c(R \times S) \equiv (R \,|\times|_c\, S)$
g. ( ) (Korth) $(\pi_A(R - S)) \equiv \pi_A(R) - \pi_A(S)$
h. ( ) (Korth) $\sigma_\theta(R\_|\times|S) \equiv \sigma_\theta(R)\ \_|\times|\ S)$
onde $=$ $\_|\times|$ $junção\ externa$ à $esquerda$

3. Considere as relações r_1(A,B,C), r_2(C, D, E) e r_3(E,F), com chaves primárias A, C e E, respectivamente. Suponha que r_1 tenha 1.000 tuplas, r_2 tenha 1.500 tuplas e r_3 tenha 750 tuplas. Estime o tamanho de r_1 $|\times|$ r_2 $|\times|$ r_3 e dê uma estratégia eficiente para calcular a junção.

4. (baseado em Navathe) Considere as transações

T1:	**T2:**
Ler_item(X); X := X – N; Escrever_item(X); Ler_item(Y); Y:= Y+N; Escrever_item(Y)	Ler_item(X); X := X + M; If X > 90 then exit Else escrever_item(X);

Discuta o resultado final dos diferentes escalonamentos abaixo, onde M = 2 e N = 2, com respeito aos resultados obtidos e a regra de consistência associada à X em T2.

Escalonamento 1

T1	**T2**
Ler_item(X)	
X := X - N	
	Ler_item(X)
	X := X + M;
Escrever_item(X)	
Ler_item(Y)	
	If X > 90 then exit
	Escrever_item(X)
Y := Y + N;	
Escrever_item(Y)	

Escalonamento 2

T1	**T2**
Ler_item(X)	
X := X - N	
Escrever_item(X)	
	Ler_item(X)
	X := X + M;
	If X > 90 then exit
	Escrever_item(X)
Ler_item(Y)	
Y := Y + N;	
Escrever_item(Y)	

5. (Navathe) Repita o exercício anterior adicionando uma verificação em T1 de forma que Y não exceda 90.
6. (baseado em Navathe) Adicione uma operação *commit* no final de cada uma das transações T1 e T2 da Figura 17.2 (acesse o livro) e liste um plano concorrente restaurável, um plano concorrente livre de cascata e um plano concorrente restrito. Explique suas respostas.

7. (baseado em Navathe) Considere os escalonamentos abaixo. Determina quais são serializáveis, mostre o escalonamentos seriais equivalentes e como você os obteve, verifique quais são livres de cascatas e para os três últimos verifique se são recuperáveis ou não. Justifique suas respostas.

    a. $r_1(X); r_3(X); w_1(X); r_2(X); w_3(X);$
    b. $r_1(X); r_3(X); w_3(X); w_1(X); r_2(X);$
    c. $r_3(X); r_2(X); w_3(X); r_1(X); w_1(X);$
    d. $r_3(X); r_2(X); r_1(X); w_3(X); w_1(X);$
    e. $r_1(X); r_1(Z); w_1(X);$
    f. $r_2(Z); r_2(Y); w_2(Z); w_2(Y);$
    g. $r_3(X); r_3(Y); w_3(Y);$
    h. $r_1(X); r_2(Z); r_1(Z); r_3(X); r_3(Y); w_1(X); w_3(Y); r_2(Y); w_2(Z); w_2(Y);$
    i. $r_1(X); r_2(Z); r_3(X); r_1(Z); r_2(Y); r_3(Y); w_1(X); w_2(Z); w_3(Y); w_2(Y);$
    j. $r_1(X); r_2(Z); r_1(Z); r_3(X); r_3(Y); w_1(X); c_1; w_3(Y); r_2(Y); w_2(Z); w_2(Y); c_2;$
    k. $r_1(X); r_2(Z); r_1(Z); r_3(X); r_3(Y); w_1(X); w_3(Y); r_2(Y); w_2(Z); w_2(Y); c_1; c_2; c_3;$
    l. $r_1(X); r_2(Z); r_3(X); r_1(Z); r_2(Y); r_3(Y); w_1(X); c_1; w_2(Z); w_3(Y); w_2(Y); c_3; c_2;$

8. (Korth) Considere as duas transações abaixo

    T1: read(A); read(B); IF A = 0 THEN B : = B + 1; write(B);
    T2: read(B); read(A); IF B = 0 THEM A := A + 1; write(A);

    Considere que o requisito de consistência é A = 0 **OU** B = 0, com A = B = 0 sendo os valores iniciais.

    a. Mostre que cada execução serial envolvendo essas duas transações preserva a consistência do banco de dados.
    b. Mostre uma execução concorrente de T1 e T2 que produz um escalonamento não serializável;
    c. Existe uma execução concorrente de T1 e T2 que produza um escalonamento serializável?

9. (baseado em Navathe) Por meio da criação e análises de grafos de serialização por conflito, verifique que se ele possui um ciclo, então pelo menos uma das transações participantes do escalonamento não obedece ao protocolo de bloqueio em duas fases.

10. Crie alguns escalonamentos que provocam deadlocks. Mostre que a execução de tais escalonamentos sob as regras dos protocolos de prevenção de deadlocks “esperar-morrer” e “ferir-esperar” não aconteceriam.

11. Para os escalonamentos abaixo, aplique os protocolos de controle de concorrência (a) de bloqueio em duas fases, (b) ordenamento por timestamp e (c) ordenamento por timestamp multiversão e analise os efeitos produzidos.

Escalonamento 1 (Navathe)

T1	T2	T3
	Ler_item(Z)	
	Ler_item(Y)	
	Escrever_item(Y)	
		Ler_item(Y)
		Ler_item(Z)
Ler_item(X)		
Escrever_item(X)		
		Escrever_item(Y)
		Escrever_item(Z)
	Ler_item(X)	
Ler_item(Y)		
Escrever_item(Y)		
	Escrever_item(X)	

Escalonamento 2 (Navathe)

T1	T2	T3
		Ler_item(Y)
		Ler_item(Z)
Ler_item(X)		
Escrever_item(X)		
		Escrever_item(Y)
		Escrever_item(Z)
	Ler_item(Z)	
Ler_item(Y)		
Escrever_item(Y)		
	Ler_item(Y)	
	Escrever_item(Y)	
	Ler_item(X)	
	Escrever_item(X)	

Escalonamento 3

T1	T2	T3
	R(A)	
	W(A)	
R(B)		
W(B)		
		R(C)
		W(C)
	R(C)	
	W(C)	
R(C)		
W(C)		
		R(B)
		W(B)
	R(B)	
R(D)		
W(D)		
	W(B)	

Escalonamento 4

T1	T2	T3
	R(A)	
	W(A)	
R(B)		
W(B)		
		R(A)
		R(B)
		R(C)
	R(C)	
	W(C)	
		W(A)
		W(B)
R(A)		
R(C)		
R(D)		
W(D)		

Obs.: Para exercitar seu entendimento dos protocolos, na ocorrência de uma reversão de uma transação ou entrada em estado de espera, simule as providências tomadas pelo SGBD e continue a execução das transações em uma ordem de intercalamento possível dentro do protocolo.

12. (Korth) Considere o protocolo de ordenação por timestamp e duas transações, uma que escreve dois itens de dados (*p* e *q*), e outra que lê os mesmos dois itens de dados. Dê um escalonamento no qual o teste de timestamp para uma operação write falha e faz com que a primeira transação seja interrompida, por sua vez, causando um aborto em cascata da outra transação, e então reinicie. Mostre como isso poderia resultar na estagnação das duas transações.
    Obs.: Essa situação, em que dois ou mais processos executam ações, mas são incapazes de completar sua tarefa devido à interação com outros processos, é chamada de **livelock**.

13. Dadas as transações abaixo, responda as questões que estão na seqüência.

T1	T2	T3	T4	T5	T6
Read(A)	Read(E)	Read(C)	Read(G)	Read(H)	Read(A)
Read(B)	Write(E)	Write(C)	Write(G)	Write(H)	Read(E)
Write(A)	Read(F)	Read(G)	Read(B)	Read(I)	Read(G)
Read(C)	Write(F)	Read(F)	Write(B)	Write(I)	Write(G)
Read(D)	Read(A)	Write(F)		Read(F)	Read(H)
Write(C)	Read(C)			Write(F)	Write(H)
Write(B)	Write(A)				

a) Assuma o protocolo de bloqueio em duas fases e crie alguns escalonamentos concorrentes para as transações acima. Faça com que, em alguns casos aconteça deadlock. Crie diferentes situações de deadlock.

b) Dos escalonamentos que você conseguiu com deadlock, escolha um para simular um sistema real. Na ocorrência de um deadlock, aplique uma medida corretiva, e continue operando com as transações, inclusive inserindo novamente transações que foram abortadas por conta das medidas corretivas. As medidas corretivas que você pode adotar são:

    i. Time out
    ii. Detecção de deadlock e escolha de uma transação para morrer

Obs.: Cuidado. Se você usar o protocolo em duas fases que não evita rollback em cascatas, você precisa lembrar que uma transação depende da outra e que se uma é abortada, pode ser que outras o sejam também.

c) A partir dos escalonamentos que você criou e que estão produzindo deadlocks, crie uma execução dos mesmos, mas usando um protocolo de prevenção de deadlock. Ou seja, o escalonamento que você criou não acontecerá efetivamente porque o protocolo de prevenção vai atuar sobre ele. Neste momento, continue simulando a execução das transações no sistema e produza um escalonamento completo que execute todas as transações. As medidas de prevenção que você pode adotar são:

    i. Protocolo Esperar-morrer
    ii. Protocolo Ferir-esperar

    Dica: Na execução dos escalonamentos do exercício c, escolha de maneira aleatória as transações que deverão entrar no sistema e os momentos de troca de direito de execução entre transações. Como as transações operam dados em comum, é provável que os protocolos de prevenção precisem atuar com freqüência.

14. A figura abaixo mostra o LOG correspondente a determinado plano, para quatro transações T1, T2, T3 e T4, no ponto da queda do sistema. Suponha que o protocolo de atualização imediata com checkpoint esteja sendo usado. Descreva o processo para recuperação da queda do sistema. Especifique quais transações serão revertidas, quais operações do LOG serão refeitas e quais serão desfeitas, e se poderá ocorrer alguma reversão em cascata.

<start, T1>
<T1, D, 20, 25>
<commit, T1>
<checkpoint,L>
<start, T2>
<T2, B, 12, 18>
<start, T4>
<T4, D, 25, 15>
<start, T3>

<T3, C, 30, 40>
<T4, A, 30, 20>
<commit, T4>
<T2, D, 15, 25>
Queda do sistema

15. Assuma as execuções que você criou para a resolução dos exercícios sobre escalonamentos e concorrência e, assumindo o protocolo de recuperação imediata, estude as conseqüências das falhas.