Exame Preliminar de Doutorado em Banco de Dados

Duração: 4 horas

10 de agosto de 1998

Atenção:
Apesar da soma dos pontos das sete questões desta prova ser 14, tente responder corretamente o número máximo de questões, a fim de maximizar as suas chances de uma boa avaliação. A nota é o máximo das somas das notas de questões, tomado sobre os subconjuntos de questões cujo valor soma 10. Seja claro e preciso em suas respostas.

Boa sorte!

Questão	Valor	Nota
1	3.0	
2	2.0	
3	1.0	
4	2.0	
5	1.0	
6	2.0	
7	3.0	
Total	14.0	

Questão 1. (3.0 pontos) Suponha que o log das requisições de acesso a um servidor web é mantido num banco de dados com o seguinte esquema:

> Requisições(arquivo, requisitante, data)
> Requisitantes(endereço_IP, cidade)

(Na tabela Requisições, o requisitante é o endereço IP da máquina que requisitou o arquivo.)

Expresse cada uma das consultas abaixo em álgebra relacional, em cálculo de tuplas e em SQL:

(a) Obter os nomes dos arquivos que no dia 11/11/97 foram requisitados por alguma máquina situada em Pirapora.

(b) Obter os nomes dos arquivos que nunca tiveram uma requisição proveniente de Porto Alegre.

(c) Obter os endereços IP das máquinas que requisitaram todos os arquivos presentes na tabela Requisições.

Nas consultas em álgebra relacional use apenas as operações básicas $\cup$, $-$, $\times$, σ e Π. Outras operações (junções, por exemplo) devem ser expressas em termos das operações básicas.

Para as consultas em SQL, use o seguinte subconjunto da linguagem SQL-92:

- Somente são admitidas construções (possivelmente aninhadas) da forma

  ```
  select ... from ... where ...
  ```

- Apenas os seguintes ítens podem aparecer na cláusula `where`:
  - negação e conectivos lógicos (`not`, `and`, `or`),
  - comparações (`=`, `<>`, ...),
  - construções `exists`, `not exists`, `in` e `not in`,
  - construções `select ... from ... where ...` aninhadas.

Questão 2. (2.0 pontos) Considere a relação

Horário(disciplina, turma, professor, aluno, sala, dia, hora, duração)

que, além de especificar a atribuição de alunos, professores e salas a turmas de disciplinas oferecidas por um departamento acadêmico, especifica os dias da semana em que cada turma tem aulas, bem como a hora do início e a duração de cada aula.

Cada disciplina pode ter várias turmas. (Uma disciplina "de massa", como Introdução à Computação, poderia ter turmas numeradas de 1 a 20.) Suponha que cada turma de uma disciplina tem um único professor, tem aulas em determinados dias da semana, sempre na mesma sala (que não varia com o dia da semana), em determinadas horas (que podem variar conforme o dia da semana). Uma turma de uma disciplina pode ter mais de uma aula no mesmo dia da semana. Cada aluno pode cursar varias disciplinas, mas não pode estar em mais de uma turma de uma mesma disciplina.

(a) Levando em conta *exclusivamente* a descrição acima, diga quais as dependências funcionais não triviais que são válidas sobre a relação Horário. Não liste exaustivamente o conjunto de dependências funcionais válidas sobre Horário, liste um subconjunto que não tenha dependências redundantes e que implique logicamente todas as dependências válidas.

(b) Faça o mesmo para as dependências multivaloradas válidas sobre Horário.

(c) Exiba uma decomposição de Horário que esteja na BCNF, seja sem perdas de junção e preserve as dependências funcionais.

(d) Exiba uma decomposição de Horário que esteja na 4NF.

Questão 3. (1.0 ponto) Prove ou dê um contra-exemplo:

> Seja R um esquema relacional, F um conjunto de dependências funcionais sobre R, e $a \in R$. Tome $R' = R - \{a\}$, e denote por F' a restrição de F^+ a R'. Se R' está na BCNF relativamente a F', então R está na 3NF relativamente a F.

Questão 4. (2.0 pontos)

(a) Considere as seguintes transações T_1, T_2 e T_3:

T_1 :	T_2 :	T_3 :
read(A)	read(A)	read(B)
A = 1.1*A	A = A + 100	read(A)
write(A)	write(A)	A = A + B
read(C)		write(A)
C = C + 2		read(C)
write(C)		write(C)

Considere o seguinte escalonamento S de T_1, T_2, T_3:

T_1	T_2	T_3
read(A)		
	read(A)	
		read(B)
		read(A)
		A = A + B
		write(A)
	A = A +100	
	write(A)	
A = 1.1*A		
write(A)		
read(C)		
C = C + 2		
write(C)		
		read(C)
		write(C)

O escalonamento S é serializável ? Justifique.

(b) Suponha que você tenha que automatizar a solução para o tipo de problema proposto no item acima, ou seja, escrever um programa que dadas transações quaisquer e um escalonamento S qualquer, decide se o mesmo é serializável. Você pode implementar a mesma forma de argumentação empregada para o exemplo acima ? Justifique. Comente sobre as dificuldades em resolver o problema.

Questão 5. (1.0 ponto) Uma forma de obter transações que executam concorrentemente sem que ocorra um escalonamento não serializável é requerer que o acesso a item de dados seja controlado através do uso de "locks" (do tipo compartilhado ou exclusivo) seguindo o protocolo de duas fases ("two-phase locking protocol").
O protocolo de duas fases pode ser refinado de forma a envolver a conversão de locks (um lock do tipo compartilhado pode ser promovido ao tipo exclusivo — operação UP —, e um lock do tipo exclusivo pode ser rebaixado para um lock compartilhado — operação DOWN).
Qual a motivação da introdução destas duas novas operações ?

Questão 6. (2.0 pontos) Considere um ambiente para execução de transações concorrentes em que há suporte para recuperação de "crash" através da manutenção de um "log". Seja S o seguinte escalonamento das transações T_1, T_2 descritas abaixo onde a operação "lock(x)" obtem um lock de acesso exclusivo ao item de dados x, "unlock(x)" libera o lock associado ao item de dados x e "commit" implica em que todas as mudanças de valores dos itens acessados pelas transação já tenham sido gravados no "log":

T_1	T_2
lock(A)	
A = A -1	
write(A)	
lock(B)	
unlock(A)	
	lock(A)
	read(A)
	A = A * 0.2
read(B)	
	write(A)
	commit
	unlock(A)
B = B / A	

Suponha que a operação "B = B / A" resulte em uma divisão por zero, causando um "crash" da transação T_1.

Descreva os passos que o recuperador de "crash" deve tomar neste exemplo para garantir que o banco de dados fique em um estado consistente.

Questão 7. (3.0 pontos) O proprietário de uma companhia que lida com animais de estimação deseja informatizar as suas lojas. O sistema deve manter informação sobre quais animais estão correntemente disponíveis em que filiais da loja, e quais espera-se que estejam disponíveis num futuro com data conhecida. Características dos animais como nome, tipo, cor e preço deve estar armazenadas no sistema, de forma que mais vendas possar ser viabilizadas através de buscas no repositório de informação. Deseja-se também manter informação sobre os animais vendidos: data da transação de venda e dados sobre o comprador. As lojas oferecem também serviços de cuidados ("boutique") de animais, hotel (clientes podem alugar espaço para deixar os seus animais de estimação durante viagens) e veterinário. Deseja-se disponibilizar no sistema serviço de agendamento de consultas e de cuidados como tosa, guardando também o histórico de cada um dos serviços pelo qual tenha passado um animal.

Faça a modelagem de dados inicial para este sistema usando o Modelo Entidade Relacionamento. Coloque informação sobre os graus dos relacionamentos, relacionamentos totais e atributos determinantes.