

Estudo dirigido

- 1) Descreva os protocolos esperar-morrer e ferir-esperar para a prevenção de deadlock.
- 2) Duas transações (T1 e T2) de banco de dados executam as seguintes seqüências de operações:

T1:

Na tabela DEPARTAMENTO, bloqueia a linha N em modo compartilhado;
Na tabela DEPARTAMENTO, lê a coluna DESPESA da linha N;
Na tabela DEPARTAMENTO, desbloqueia a linha N;
Na tabela PROJETO, bloqueia a linha M em modo compartilhado;
Na tabela PROJETO, lê a coluna VERBA da linha M;
Na tabela PROJETO, desbloqueia a linha M;
Na tabela PROJETO, bloqueia a linha M em modo exclusivo;
Na tabela PROJETO, escreve a coluna VERBA da linha M com o valor VERBA + DESPESA;
Na tabela PROJETO, desbloqueia a linha M;

T2:

Na tabela PROJETO, bloqueia linha M em modo compartilhado;
Na tabela PROJETO, lê a coluna VERBA da linha M;
Na tabela PROJETO, desbloqueia a linha M;
Na tabela DEPARTAMENTO, bloqueia a linha N em modo compartilhado;
Na tabela DEPARTAMENTO, lê a coluna DESPESA da linha N;
Na tabela DEPARTAMENTO, desbloqueia a linha N;
Na tabela DEPARTAMENTO, bloqueia a linha N em modo exclusivo;
Na tabela DEPARTAMENTO, escreve a coluna DESPESA da linha N com o valor DESPESA + VERBA;
Na tabela DEPARTAMENTO, desbloqueia a linha N;

É correto afirmar que essas transações

- não são serializáveis e, portanto, não podem ser executadas concorrentemente.
- não podem entrar em bloqueio infinito porque obedecem ao protocolo de bloqueio em duas fases (two-phase locking).
- podem entrar em bloqueio infinito (deadlock) se executadas concorrentemente.
- obedecem ao protocolo de bloqueio em duas fases (two-phase locking).
- são serializáveis e obedecem ao protocolo de bloqueio em duas fases (two-phase locking).

Considere o seguinte enunciado para responder às questões de nos 3 e 4.

Uma empresa de geração de energia deseja armazenar um conjunto de dados importantes sobre os tipos de energia com que trabalha e os seus campos de geração. Cada tipo de energia possui um código (identificador único), um nome e uma descrição. Além disso, cada campo de geração possui um código (identificador único), um nome, uma região e um valor médio por kW. Assume-se que:

- cada campo de geração de energia é de um, e somente um, tipo de energia;
- pode existir mais de um campo de geração para cada tipo de energia;
- podem ser previstos alguns tipos de energia para os quais ainda não existem campos de geração.

Suponha o seguinte esquema de relação que modela o problema descrito anteriormente, para responder às questões de nos 3 e 4.

TipoEnergia (codigo, nome, descricao)

CampoGeracao (codigo, nome, codigoTipoEnergia, regiao, valorMedioPorkW)

observação: o campo codigoTipoEnergia armazena o valor do código do tipo de energia de um campo de geração.

- 3) Qual comando SQL padrão retorna o código e o nome (somente) de todos os tipos de energia que possuem, pelo menos, um campo de geração em uma região chamada 'Santos'?
- a) `SELECT * FROM TipoEnergia WHERE codigo IN (SELECT codigoTipoEnergia FROM CampoGeracao WHERE regiao = 'Santos');`
 - b) `SELECT te.codigo, te.nome FROM TipoEnergia te WHERE te.codigo IN (SELECT cg.codigo FROM CampoGeracao cg WHERE cg.regiao = 'Santos');`
 - c) `SELECT te.codigo, te.nome FROM TipoEnergia te, CampoGeracao cg WHERE te.codigo=cg.codigo AND cg.regiao = 'Santos';`
 - d) `SELECT te.codigo, te.nome FROM TipoEnergia te, CampoGeracao cg WHERE te.codigo=cg.codigoTipoEnergia AND cg.regiao = 'Santos';`
 - e) `SELECT DISTINCT te.* FROM TipoEnergia te, CampoGeracao cg WHERE te.codigo=cg.codigoTipoEnergia AND cg.regiao = 'Santos';`
- 4) Qual comando SQL padrão apresenta, para cada tipo de energia, seu código, seu nome e o valor mínimo do conjunto de valores médios por kW (valorMedioPorkW) dos campos de geração associados a esse tipo de energia?
- a) `SELECT te.codigo, te.nome, MIN (cg.valorMedioPorkW) FROM TipoEnergia te, CampoGeracao cg WHERE te.codigo=cg.codigoTipoEnergia;`
 - b) `SELECT te.codigo, te.nome, MIN (cg.valorMedioPorkW) FROM TipoEnergia te, CampoGeracao cg WHERE te.codigo=cg.codigoTipoEnergia GROUP BY te.codigo, te.nome;`
 - c) `SELECT te.codigo, te.nome, MIN (cg.valorMedioPorkW) FROM TipoEnergia te, CampoGeracao cg HAVING te.codigo=cg.codigoTipoEnergia;`
 - d) `SELECT te.codigo, te.nome, MIN (cg.valorMedioPorkW) FROM TipoEnergia te, CampoGeracao cg GROUP BY te.codigo, te.nome HAVING te.codigo=cg.codigoTipoEnergia;`
 - e) `SELECT te.codigo, te.nome, MIN (cg.valorMedioPorkW) FROM TipoEnergia te, CampoGeracao cg WHERE te.codigo=cg.codigoTipoEnergia GROUP BY te.codigo, te.nome HAVING MIN (cg.valorMedioPorkW)= cg.valorMedioPorkW;`

Considere o script SQL abaixo para responder as questões 5 e 6.

```
CREATE TABLE Album(
  cod_album    INT          NOT NULL,
  nome        VARCHAR(30)   NOT NULL,
  PRIMARY KEY (cod_album)
);

CREATE TABLE AlbumMusica(
  cod_album    INT          NOT NULL,
  cod_musica   INT          NOT NULL,
  PRIMARY KEY (cod_album, cod_musica)
);

CREATE TABLE Artista(
  cod_artista  INT          NOT NULL,
  nome        VARCHAR(50)   NOT NULL,
  cod_país    INT          NOT NULL,
  PRIMARY KEY (cod_artista)
);

CREATE TABLE Musica(
  cod_musica   INT          NOT NULL,
  titulo       VARCHAR(30)   NOT NULL,
  duracao     INT          NOT NULL,
  data_composicao DATETIME   NOT NULL,
  cod_artista  INT          NOT NULL,
  PRIMARY KEY (cod_musica)
);

CREATE TABLE País(
  cod_país    INT          NOT NULL,
  nome        VARCHAR(30)   NOT NULL,
  PRIMARY KEY (cod_país)
);

ALTER TABLE AlbumMusica ADD CONSTRAINT RefAlbum3
FOREIGN KEY (cod_album)
REFERENCES Album(cod_album);

ALTER TABLE AlbumMusica ADD CONSTRAINT
RefMusica6
FOREIGN KEY (cod_musica)
REFERENCES Musica(cod_musica);

ALTER TABLE Artista ADD CONSTRAINT RefPaís1
FOREIGN KEY (cod_país)
REFERENCES País(cod_país);

ALTER TABLE Musica ADD CONSTRAINT RefArtista2
FOREIGN KEY (cod_artista)
REFERENCES Artista(cod_artista);
```

5) Assinale a consulta que retorna o nome dos países dos criadores de todas as músicas.

- a) SELECT p.nome
FROM país p, albummusica am, musica m
WHERE am.cod_musica = m.cod_musica
AND m.cod_país = p.cod_país;
- b) SELECT *
FROM musica m
WHERE m.cod_país=país. cod_país;
- c) SELECT DISTINCT *
FROM artista a, musica m, albummusica am, país
p
WHERE a.cod_artista = m.cod_artista
AND a.cod_país = p.cod_país
AND am.cod_musica=m.cod_artista;
- d) SELECT p.nome
FROM artista a, musica m, albummusica am, país
p
WHERE a.cod_artista = m.cod_artista
AND a.cod_país = p.cod_país
AND am.cod_musica=m.cod_artista;
- e) SELECT DISTINCT p.nome
FROM país p, artista a, musica m
WHERE a.cod_artista = m.cod_artista
AND a.cod_país = p.cod_país;

6) Sobre os relacionamentos previstos no banco definido pelo script SQL, assinale a afirmação **INCORRETA**.

- a) Uma mesma música pode fazer parte de álbuns diferentes.
- b) Um artista, obrigatoriamente, possui um país de origem.
- c) Não há artista sem música.
- d) Toda música tem um artista.
- e) Podem existir várias músicas em um álbum.

Nas questões 7 e 8 considere o esquema relacional abaixo:

Pedidos(cod_ordem, cod_peça, quantidade)
Projeto_Necessita (cod_projeto, cod_peça, quantidade)
Fornecimento (cod_ordem, nome_fornecedor, data_ordem)

7) Para atender à consulta "Que peças foram pedidas por que projetos?", a alteração necessária em uma das relações do esquema relacional apresentado é:

- a) Pedidos(cod_ordem, cod_projeto, cod_peça, quantidade)
- b) Fornecimento (cod_ordem, cod_peça, nome_fornecedor, data_ordem)
- c) Pedidos(cod_ordem, cod_projeto, quantidade)
- d) Pedidos(nome_fornecedor, cod_peça, quantidade)
- e) Projeto_Necessita (cod_projeto, cod_ordem, quantidade, status)

8) Analise as seguintes consultas:

- (I) Que peças foram fornecidas por um dado fornecedor?
- (II) Que fornecedores forneceram peças para um dado pedido?
- (III) Que fornecedores forneceram peças para um dado projeto?
- (IV) Liste os pedidos cuja quantidade de peças pedidas excede as necessidades de um dado projeto.

Considerando que o esquema relacional acima tem problemas de projeto, as consultas que podem ser resolvidas apenas com esse conjunto de relações são somente:

- a) I e II
- b) I e III
- c) II e IV
- d) III
- e) IV

9) Responda (V) verdadeiro ou (F) falso para as afirmativas abaixo:

- a) () Em geral, o objetivo do projeto de banco de dados relacional é gerar um conjunto de esquemas de relação que nos permita armazenar informações sem redundância desnecessária, ainda nos permitindo recuperar informações com facilidade. Isso é realizado projetando esquemas que estejam em uma forma normal apropriada.
- b) () Especialização é o resultado de tomar a união de dois ou mais conjuntos de entidades disjuntos e Generalização é o resultado de tomar um subconjunto de um conjunto de entidades de nível superior para formar um conjunto de entidades de nível inferior.
- c) () Um conjunto de entidades que não possui atributos suficientes para formar uma chave primária é chamado um conjunto de entidades forte.
- d) () Se a decomposição não for preservadora de dependência, dada uma atualização de banco de dados, todas as dependências funcionais podem ser verificáveis em relações individuais, sem computar uma junção de relações da decomposição.
- e) () Os termos superchave, chave candidata e chave primária aplicam-se aos conjuntos de entidades e relacionamentos, bem como a esquemas de relação. Identificar a chave primária de um conjunto de relacionamentos requer alguns cuidados, pois ela é composta de atributos de um ou mais conjuntos de entidades relacionados.

- f) () Quando definimos cuidadosamente um diagrama E-R, identificando todas as entidades corretamente, os esquemas de relação gerados do diagrama E-R , ainda se faz necessário normalizar os esquemas de relação, preferencialmente até a BCNF.
- g) () Uma entidade é um objeto que existe no mundo real e que é distinguível dos outros objetos. Expressamos a distinção associando a cada entidade um conjunto de atributos que descreva o objeto.