

CONHECIMENTOS ESPECÍFICOS

Julgue os itens seguintes, no que se refere aos benefícios que pode trazer a utilização de sistema gerenciador de banco de dados (SGBD) em relação a sistemas de processamento de arquivos.

- 51 SGBD utiliza o conceito de atomicidade do registro, assegurando que, uma vez detectada uma falha na operação com o registro, os dados sejam salvos em seu último estado consistente, anterior a essa falha.
- 52 SGBD, que é formado por uma coleção de arquivos e programas inter-relacionados, tem por objetivo garantir que o acesso para consultas e alterações dos dados nos arquivos seja limitado a um único administrador, responsável por efetuar as modificações e atualizações requeridas pelos usuários do sistema.
- 53 A utilização de SGBD permite o gerenciamento do acesso concorrente, ou seja, permite atualizações simultâneas nos dados para aumento do desempenho do sistema como um todo e para melhores tempos de resposta.

Julgue os próximos itens, acerca dos sistemas gerenciadores de banco de dados que implementam o modelo relacional de dados.

- 54 O modelo relacional de dados necessita que programas ou sistemas implementem regras para evitar características indesejáveis, tais como repetição de informação, incapacidade de representar parte da informação e perda de informação.
- 55 A estrutura fundamental do modelo relacional de dados é a relação, na forma de tabela, constituída por um ou mais atributos — os campos —, que traduzem o tipo de dados a armazenar. Cada instância do esquema — linha — é chamada de tupla ou registro.
- 56 Assim como nos modelos que o precedem, o modelo relacional de dados precisa de caminhos predefinidos para que os dados sejam acessados.

Julgue os itens a seguir, no que concerne ao conceito de modelo de dados em SGBD.

- 57 O modelo lógico define-se como uma descrição de banco de dados no nível de abstração visto pelo usuário do SGBD, por isso esse modelo é flexível e independente do tipo particular de SGBD que está sendo usado.
- 58 O modelo conceitual define-se como um modelo de dados abstrato, que descreve a estrutura de um banco de dados de forma independente de um SGBD particular.

Com relação ao modelo entidade e relacionamento, julgue os itens subsequentes.

- 59 Relacionamento é o nome dado às associações entre as entidades.
- 60 Os objetos do mundo real são mapeados como entidades nesse modelo.

Julgue os itens a seguir, no que se refere aos conceitos de banco de dados.

- 61 O local onde são armazenados ou agrupados todos os objetos, programas e dados é denominado esquema.
- 62 Instância é o nome técnico atribuído ao conjunto de programas utilizados, em um dado momento, para o gerenciamento de determinado banco de dados.

Julgue os itens 63 e 64, acerca de independência dos dados.

- 63 Independência de dados lógica é a capacidade de modificar o esquema lógico sem que, com isso, qualquer programa de aplicação precise ser reescrito. Modificações no nível lógico são necessárias, ocasionalmente, para aprimorar o desempenho.

- 64 Modificações no nível físico são necessárias sempre que um requisito for alterado, por exemplo, para melhorar o desempenho do processo de inclusão de registros.

Julgue os itens que se seguem, a respeito do modelo relacional.

- 65 O processo de mapeamento do modelo entidade relacionamento para o modelo relacional tem como objetivo evitar as anomalias de acesso concorrente.
- 66 No processo de mapeamento do modelo relacional para o modelo lógico, uma entidade torna-se uma tabela, e o identificador de cada ocorrência de uma entidade, a chave primária da tabela.

Julgue os itens seguintes, a respeito do processo de normalização.

- 67 Uma relação está na terceira forma normal se ela contém dependências funcionais e também dependências transitivas.
- 68 A normalização é um processo passo a passo e irreversível de substituição de uma dada coleção de tabelas por sucessivas coleções de outras tabelas; estas, no entanto, possuem uma estrutura progressivamente mais simples.
- 69 A dependência funcional é uma associação que se estabelece entre duas ou mais relações e define-se do seguinte modo: se X e Y são relações, diz-se que X é funcionalmente dependente de Y se cada um dos valores de X em R , em que R é a relação que contém a chave primária, tem associado a si um e um só valor de Y em R .
- 70 Classifica-se como normalizada a tabela que possui tabela aninhada.

Julgue os itens a seguir, no que diz respeito aos conceitos de dicionário de dados.

- 71 Linguagem de definição de dados proporciona comandos para definição de esquemas de relações, exclusão de relações, criação de índices e modificação nos esquemas de relações.
- 72 O dicionário de dados é um repositório utilizado com a finalidade de armazenar informações dos atributos de cada entidade com o maior número de detalhe possível.
- 73 O dicionário de dados corresponde a um banco de dados em separado que contém informações dos componentes do banco de dados. Dessa forma, para manter segurança das informações, estas devem ser obrigatoriamente armazenadas em local diferente do banco de dados no qual se está trabalhando.

Julgue os itens seguintes, a respeito de arquitetura de banco de dados.

- 74 A principal vantagem da arquitetura de dados em duas camadas é a divisão do armazenamento e do processamento de dados em dois sistemas. Mas esse processo tem um custo que se reflete em considerável aumento do tráfego de dados na rede.
- 75 A redundância controlada é uma das estratégias utilizadas para melhorar a *performance* da arquitetura de um banco de dados distribuídos.
- 76 As funcionalidades de um banco de dados em uma arquitetura dividida em duas camadas podem ser divididas em duas categorias: *front-end* e *back-end*. As funcionalidades de *front-end* são as que permitem gerenciar as estruturas de acesso e realizar o desenvolvimento e a otimização de consultas, além do controle de concorrência e recuperação de dados.

```

CREATE TABLE unidade_da_federacao
(
    codigo_ibge numeric(7,0),
    uf_sigla varchar(2) NOT NULL,
    uf_nome varchar(50),
    tamanho_area numeric(15,5),
    CONSTRAINT pk_uf PRIMARY KEY (codigo_ibge),
    CONSTRAINT uq_uf_ibge UNIQUE (uf_sigla)
);

CREATE TABLE cidade
(
    codigo_ibge numeric (7,0) NOT NULL,
    uf numeric(2,0),
    nome varchar(50),
    tamanho_area numeric(15,5),
    CONSTRAINT pk_cidade PRIMARY KEY (codigo_ibge),
    CONSTRAINT fk_cidade_uf FOREIGN KEY (uf) REFERENCES unidade_da_federacao
);

```

Considerando a linguagem de definição de dados e o código acima, que tem como objetivo criar as relações `unidade_da_federacao` e `cidade`, representando dados referentes a estados e municípios da Federação, julgue os próximos itens.

- 77 Para garantir que não haverá repetição de nome de unidade da Federação na tabela `unidade_da_federacao`, é correto utilizar a expressão SQL abaixo.

```

CREATE UNIQUE INDEX unidade_da_federacao_nome ON
    unidade_da_federacao (uf_nome);

```

- 78 A execução do código abaixo cria corretamente uma *view*, que corresponde a uma lista contendo nome de município, tamanho da área do município, sigla do estado e tamanho da área do estado, não necessariamente nessa ordem.

```

CREATE VIEW tamanho_area_cidades_uf_view as
SELECT a.uf_sigla, b.nome,
    b.tamanho_area area_municipio, a.tamanho_area area_estado
FROM unidade_da_federacao a, cidade b
WHERE a.codigo_ibge=b.codigo_ibge

```

- 79 Considerando-se que, na tabela `unidade_da_federacao`, o campo `tamanho_area` não esteja preenchido e que esse campo seja resultante da soma das áreas de todas as cidades que pertencem a determinada unidade da Federação, a expressão SQL abaixo permite calcular corretamente a área total dessa unidade da Federação, referida no código como `estado`.

```

UPDATE unidade_da_federacao estado SET
    tamanho_area = (SELECT SUM(tamanho_area)
        FROM cidade WHERE uf=estado.codigo_ibge);

```

Julgue os itens a seguir, a respeito de álgebra relacional.

- 80 Considere um banco de dados com as seguintes tabelas e campos, representados na forma de um pseudo código.

```

cliente (nomecliente, codigocliente, cidade, codigoproduto)
    chave estrangeira (codigoproduto) referenciado produto
produto (nomeproduto, codigoproduto, descricaoproduto)

```

Nessa situação, para se encontrarem todos os nomes de clientes que possuem o produto de nome `ACME`, é correto, ao se utilizar a linguagem SQL, o emprego do código abaixo.

```

SELECT nomecliente
FROM cliente as c, produto as p
WHERE c.codigoproduto = p.codigoproduto
AND p.nomeproduto = 'ACME';

```

- 81 A operação de junção interna entre duas tabelas **X** e **Y** e com critério de junção **J** tem a função, na álgebra relacional, de concatenar cada linha da tabela **X** com cada linha da tabela **Y** sempre que o critério de junção **J** for verdadeiro. Caso, para uma linha de **X**, não exista nenhuma linha em **Y** que torne o critério verdadeiro, a linha de **X** aparece no resultado concatenada com campos vazios (NULL).

- 82 Em um banco de dados relacional, com as relações expressas no código a seguir e utilizando-se apenas as operações de álgebra relacional união, interseção, diferença, junção natural, projeção e seleção, a consulta “Qual a profissão dos clientes que trabalham em alguma empresa de nome ALPHA”, para ser processada, exige, ao menos, a operação de interseção das relações `cliente`, `trabalha` e `empresa`.

```

cliente (cpf, profissao)
trabalha (cpf, cnpj, datacontratacao)
empresa (cnpj, nome, endereco)

```

- 83 A operação da álgebra relacional `SELECT` extrai as tuplas específicas de uma relação, e a operação `PROJECT` extrai atributos específicos de uma relação.

Considere a tabela de uma base de dados relacional definida com a estrutura abaixo.

```
colaborador (codcolaborador,
            nomecolaborador,
            codigodepartamento)
```

Considerando que essa tabela tem um índice sobre as colunas (codcolaborador, codigodepartamento), nessa ordem, julgue os itens subsequentes.

- 84 Esse índice é adequado para ser usado pelo SGBD relacional para acelerar uma consulta na qual é fornecido um valor de nomecolaborador.
- 85 Esse índice pode ser usado pelo SGBD relacional para acelerar uma consulta na qual são fornecidos os valores de codcolaborador, codigodepartamento.

```
departamento (codigodepartamento, nomedepartamento)
colaborador (codcolaborador, nomecolaborador, codigodepartamento)
            chave estrangeira (codigodepartamento) referenciando departamento
```

Considerando o pseudo código acima, julgue os itens que se seguem.

- 86 Em SQL, para se criar um relacionamento entre as duas tabelas existentes no SGBD — departamento e colaborador —, é correto o uso da expressão a seguir.

```
ALTER TABLE colaborador ADD CONSTRAINT
colaborador_fk FOREIGN KEY (codigodepartamento)
REFERENCES codigodepartamento (departamento);
```

- 87 Em SQL, para se listarem os nomes dos departamentos da tabela departamento que não possuem colaboradores, é correto utilizar o código abaixo.

```
SELECT nomedepartamento
FROM departamento d
EXCEPT
(SELECT nomedepartamento
FROM departamento d, colaborador c
WHERE d.codigodepartamento = c.codigodepartamento);
```

- 88 Em SQL, para se criar a tabela departamento, é correto que seja utilizada a expressão apresentada no código abaixo.

```
CREATE TABLE departamento (
codigodepartamento numeric(4,0) NOT NULL PRIMARY KEY,
nomedepartamento varchar(200));
```

Considerando o conceito de visões do modelo relacional, julgue os itens seguintes.

- 89 Uma visão relacional é uma consulta virtual que nunca é materializada.
- 90 Aplicativos do banco de dados podem ser executados sobre visões de relações da base de dados.

Julgue os próximos itens, relativos ao processamento de transações em bancos de dados.

- 91 O resumo incorreto, um dos problemas decorrentes da execução descontrolada de transações simultâneas, ocorre quando uma transação, depois de atualizar um item do banco de dados, falha por um algum motivo.
- 92 Transação é uma unidade atômica de processamento no banco de dados e tem a seguinte característica: ou estará terminada totalmente ou não foi realizada de forma alguma.
- 93 Os sistemas de banco de dados que obedecem à propriedade denominada isolamento permitem que transações concorrentes compartilhem resultados intermediários.

Acerca das técnicas de controle de concorrência, julgue os itens subsequentes.

- 94 Consta do protocolo de bloqueio com granularidade múltipla (MGL – *multiple granularity locking*) a seguinte regra: a raiz da árvore precisa ser bloqueada primeiro, em qualquer modo.
- 95 Ocorre *deadlock* quando uma transação de um conjunto com duas ou mais transações concorrentes aguarda por algum item de dado que foi bloqueado por outra transação do mesmo conjunto.

Acerca das técnicas de controle de recuperação de banco de dados, julgue o item abaixo.

- 96 O algoritmo ARIES é exemplo de algoritmo de recuperação utilizado em sistemas de bancos de dados, sendo o *logging write-ahead* um dos conceitos em que esse algoritmo se embasa.

Acerca dos procedimentos armazenados (*stored procedures*), julgue os itens a seguir.

- 97 A forma geral da declaração de procedimentos armazenados é a seguinte.

```
CREATE PROCEDURE <nome do procedimento>
(<parâmetros>)
<declarações locais>
<corpo do procedimento>;
```

Por não serem opcionais, os “parâmetros” e “declarações locais” devem ser especificados.

- 98 Os procedimentos armazenados no servidor de banco de dados podem ser invocados por inúmeras aplicações.

Com relação a visões (*views*) e SQL ANSI, julgue os itens consecutivos.

- 99 Em SQL, caso uma *view* tenha de atualizar dados no banco de dados, a cláusula *WITH CHECK* deverá ser acrescentada ao final da definição da *view*.

- 100 Em SQL, tanto *CREATE VIEW* quanto *CREATE LIST TABLE* são comandos que permitem que seja especificada uma *view*.

- 101 As visões definidas sobre várias tabelas por meio de junções, em geral, não são atualizáveis.

Acerca dos gatilhos (*triggers*) na linguagem SQL ANSI, julgue os itens subsequentes.

- 102** Ao se especificar um gatilho, as palavras-chave obrigatórias `FOR EACH ROW` garantem que a regra será disparada mais de uma vez para cada linha afetada pelo evento de disparo.
- 103** Evento(s), condição e ação são componentes que constituem um gatilho típico.
- 104** Em SQL, o comando `CREATE TRIGGER` permite que se especifique um gatilho.
- 105** Gatilhos não permitem especificar tipo algum de regra ativa em sistemas de bancos de dados.
- 106** Os comandos da linguagem SQL padrão `INSERT`, `DELETE` e `UPDATE` podem ser utilizados para se definir um gatilho.
- 107** Ao se especificar um gatilho, a cláusula opcional `WHEN` é utilizada para indicar as condições que necessitam ser verificadas antes de a ação ser executada.

Acerca das ferramentas de *front-end*, julgue os itens consecutivos.

- 108** A ferramenta DBDesigner versão 4 permite criar um modelo de banco de dados que contenha tabelas, relacionamentos entre tabelas e conexão entre dois relacionamentos.
- 109** A ferramenta Rational Rose Data Modeler possibilita realizar engenharias direta e reversa de um banco de dados.
- 110** A ferramenta de *front-end* DBDesigner versão 4, de domínio público, é utilizada em projeto de banco de dados. Por meio dessa ferramenta, é possível criar relacionamentos binários entre tabelas e relacionamentos recursivos em uma tabela.

Acerca do SQL ANSI, julgue os itens de **111** a **115**.

- 111** Para a criação de uma tabela, a cláusula `UNIQUE` é obrigatória e, por meio dela, especificam-se chaves primárias.
- 112** Considere o seguinte *script* SQL.

```
CREATE TABLE cliente
(nome varchar(60) NOT NULL,
CPF char(11) NOT NULL,
sexo char(1)
);
```

A execução do *script* SQL descrito acima cria a tabela `cliente` com 3 colunas, sendo que nenhuma coluna tem a propriedade de chave primária.

- 113** Considere que a tabela `cliente` tenha sido criada em um banco de dados por meio do seguinte *script* SQL.

```
CREATE TABLE cliente
(nome varchar(50) NOT NULL,
CPF integer NOT NULL,
PRIMARY KEY(CPF)
);
```

Nesse caso, é possível inserir uma nova linha à tabela `cliente` por meio do *script* SQL apresentado abaixo.

```
INSERT INTO cliente VALUES ('Antonio Silva',
123);
```

Também é possível inserir outra nova linha à tabela `cliente` por meio do seguinte *script* SQL.

```
INSERT INTO cliente (nome, CPF) VALUES ();
```

- 114** Em linguagem SQL, os comandos `GRANT` e `REVOKE` permitem especificar, respectivamente, concessão e revogação de privilégios aos usuários de bancos de dados.

- 115** Considere que a tabela `empregado` tenha sido criada em um banco de dados por meio do seguinte *script* SQL.

```
CREATE TABLE empregado
(matricula integer NOT NULL,
salario decimal(10,2) NOT NULL,
PRIMARY KEY(matricula) );
```

Nesse caso, se o *script* SQL mostrado abaixo for executado, todas as linhas da tabela `empregado` serão modificadas, e todos os empregados terão aumento de 10% no salário.

```
UPDATE empregado SET salario = salario * 1,1
WHERE matricula = 2012;
```

Acerca do SGBD ORACLE, julgue os itens seguintes.

- 116** A *tablespace* consiste em um espaço no *hard disk* (disco) onde é possível que se armazenem tabelas. O comando `CREATE TABLESPACE` permite criar uma *tablespace*.

- 117** A execução da consulta SQL `SELECT * FROM (SELECT * FROM empregado ORDER BY matricula DESC) WHERE ROWNUM <= 2 ORDER BY matricula;` terá como resultado apenas as duas primeiras linhas dos empregados ordenados por matrícula de forma decrescente.

- 118** Determinadas características de orientação a objeto incorporadas à versão 8 do ORACLE possibilitam definir atributos multivalorados. O tipo de dados utilizado para essa definição é o `VARRAY`, que possui as propriedades `COUNT` e `LIMIT`.

- 119** *Large object* (LOB) são tipos de dados para armazenar dados não estruturados cujo tamanho mínimo seja de 4.000 *bytes*.

- 120** A execução da consulta SQL

```
SELECT departamento, AVG (salario)
FROM empregado
WHERE AVG (salario) > 2000
GROUP BY departamento;
```

terá como resultado a média dos salários dos empregados agrupada por departamento, se essa média for maior que 2.000.