

-- CONHECIMENTOS ESPECÍFICOS --**CONCEITOS BÁSICOS****Questão 31**

Assinale a opção correta em relação a *deadlock*.

- Ⓐ *Deadlocks* só ocorrem em bancos de dados distribuídos com múltiplos servidores que concorrem pelos mesmos recursos.
- Ⓑ *Deadlock* é uma falha temporária causada por perda de conexão com o banco de dados, sendo resolvida automaticamente com o *commit* da transação.
- Ⓒ *Deadlocks* são causados por falhas de *hardware* e não estão relacionados à lógica de bloqueios entre transações.
- Ⓓ Ocorre *Deadlock* quando duas ou mais transações acessam os mesmos dados simultaneamente em modo leitura, impedindo o progresso de ambas.
- Ⓔ Um *deadlock* ocorre quando transações entram em espera circular por recursos que nunca serão liberados, sendo possível evitar a situação por meio de escalonamento otimista.

Questão 32

A respeito de redes locais (LAN), assinale a opção correta.

- Ⓐ As LAN utilizam exclusivamente tecnologia sem fio para garantir mobilidade total dos dispositivos.
- Ⓑ O protocolo principal utilizado em LAN é o HTTP, que garante comunicação direta entre *switches* e roteadores.
- Ⓒ As LAN são projetadas para conectar dispositivos em diferentes cidades, por meio de enlaces de longa distância.
- Ⓓ Uma característica comum das LAN é a alta taxa de transferência de dados em distâncias curtas, como em área interna a um edifício.
- Ⓔ As LAN, cuja instalação requer autorização federal, são geralmente administradas por provedores de Internet.

TÉCNICAS E LINGUAGENS DE PROGRAMAÇÃO**Questão 33**

A programação estruturada surgiu como uma alternativa à programação não estruturada, com o objetivo de facilitar o desenvolvimento, a leitura e a manutenção de programas. Um dos princípios fundamentais da programação estruturada é

- Ⓐ a eliminação da necessidade de estruturas condicionais como *if e switch*, substituindo-as por tabelas de decisão.
- Ⓑ o uso intensivo de instruções *goto* para garantir flexibilidade no fluxo de execução.
- Ⓒ a estruturação do programa em funções ou procedimentos, com controle de fluxo baseado em sequência, seleção e repetição.
- Ⓓ a divisão de código em múltiplas *threads* que são executadas simultaneamente.
- Ⓔ o uso de variáveis globais para facilitar o compartilhamento de dados entre partes do programa.

Questão 34

Considerando que, no desenvolvimento de *software*, o conceito de componentização é utilizado para melhorar a organização e a reutilização do código, assinale a opção correta.

- Ⓐ A componentização substitui a necessidade de modelagem arquitetural no desenvolvimento de sistemas complexos.
- Ⓑ Um componente deve possuir alta coesão e baixo acoplamento, o que favorece sua reutilização e manutenção.
- Ⓒ Um componente deve ser implementado como uma função recursiva, para garantir flexibilidade à componentização.
- Ⓓ O principal objetivo da componentização é eliminar completamente a necessidade de testes unitários.
- Ⓔ Componentes devem compartilhar diretamente o estado de variáveis globais para a simplificação da comunicação entre eles.

MODELAGEM DE DADOS**Questão 35**

Assinale a opção em que é apresentada uma ação adequada de segurança dos dados no contexto da administração de banco de dados.

- Ⓐ habilitar o acesso remoto ao banco de dados diretamente na porta padrão (por exemplo, 3306 ou 5432), liberada para todos os IP confiáveis
- Ⓑ manter procedimentos de *backup* manuais para que apenas o administrador de banco de dados principal tenha acesso aos dados em caso de falha
- Ⓒ criar políticas de *backup* automatizado e rotinas de verificação de integridade dos dados periodicamente
- Ⓓ utilizar a conta *dba* como acesso principal e compartilhado entre todos os membros da equipe técnica
- Ⓔ utilizar um mecanismo de autenticação baseado em IP confiável, em substituição à autenticação por *login/senha*

Questão 36

Uma equipe de desenvolvimento está criando um sistema para gerenciar as operações de uma agência bancária. Após as primeiras entrevistas com os gerentes do banco, foram identificados os seguintes elementos do domínio:

- clientes realizam diversas transações bancárias;
- cada conta bancária pertence a um cliente específico;
- transações podem ser depósitos, saques ou transferências;
- as contas possuem saldo e data de abertura;
- a agência possui funcionários que atendem aos clientes;
- cada funcionário pode estar lotado em uma agência diferente ao longo do tempo.

Considerando a situação hipotética precedente, assinale a opção em que são apresentados somente elementos que devem ser modelados como entidades, de acordo com o modelo entidade-relacionamento.

- Ⓐ agência, conta, transação, funcionário, saldo atual
- Ⓑ cliente, conta, transação, tipo de transação, saldo
- Ⓒ conta, cliente, agência, funcionário, data da abertura da conta
- Ⓓ cliente, funcionário, saque, depósito, transferência
- Ⓔ cliente, conta, transação, funcionário, agência

Questão 37

Assinale a opção correta em relação aos tipos de atributos no contexto da modelagem de dados de acordo com o modelo entidade-relacionamento.

- Ⓐ Atributos chave nunca podem ser atributos simples; eles devem ser compostos por pelo menos dois subatributos.
- Ⓑ Atributos simples podem ser decompostos em subpartes com significado próprio.
- Ⓒ Atributos multivalorados podem conter mais de um valor, mas não podem ser combinados com atributos compostos.
- Ⓓ Atributos derivados são aqueles cujo valor pode ser calculado a partir de outros atributos armazenados no banco de dados.
- Ⓔ Atributos compostos possuem múltiplas ocorrências de valores em um mesmo registro.

Questão 38

Uma equipe de analistas está desenvolvendo o banco de dados de uma agência bancária e, durante o levantamento de requisitos, colheu os seguintes dados:

- cada cliente possui um código único, nome, CPF, endereço e pode possuir mais de uma conta bancária;
- cada conta bancária pertence a um único cliente, possui número exclusivo, tipo (corrente ou poupança) e saldo;
- a agência deseja armazenar os dados de transações, que envolvem valores movimentados em contas bancárias, com data, hora e tipo (depósito, saque ou transferência).

Com base nessa situação hipotética, assinale a opção em que é corretamente descrita a modelagem conceitual dos dados conforme o modelo entidade-relacionamento.

- A** Endereço deve ser modelado como entidade, já que contém múltiplos campos e se relaciona com o cliente.
- B** Cliente, conta e transação devem ser modelados como entidades, sendo do tipo 1:N o relacionamento entre cliente e conta.
- C** Cliente deve ser modelado como atributo da entidade conta, pois cada conta pertence a um único cliente.
- D** Transação deve ser modelada como um relacionamento entre cliente e conta para permitir rastreamento de operações.
- E** Transação deve ser modelada como atributo multivalorado da entidade conta, pois várias transações podem ocorrer por conta.

Questão 39

No modelo relacional de banco de dados, diferentes tipos de chaves são utilizados para garantir a integridade e a unicidade das tuplas nas tabelas. Nesse modelo, superchave é

- A** o único atributo obrigatório em toda tabela relacional, o qual não pode conter valores nulos.
- B** uma chave externa que referencia a chave primária de outra tabela, estabelecendo integridade referencial.
- C** qualquer conjunto de atributos que identifica unicamente uma tupla em uma relação.
- D** uma chave alternativa que, embora única, não foi escolhida como a chave primária.
- E** uma chave composta por dois ou mais atributos obrigatoriamente extraídos de tabelas diferentes.

Questão 40

Em um banco de dados que armazena informações sobre pedidos de clientes, uma tabela foi criada inicialmente com as seguintes colunas: `PedidoID`, `ClienteNome`, `ClienteTelefone`, `ProdutoID`, `ProdutoNome`, `Quantidade` e `PreçoUnitario`. Essa estrutura apresenta redundâncias e possíveis anomalias.

Nessa situação, a normalização tem como objetivo

- A** garantir que todas as tabelas possuam apenas uma chave primária e nenhuma chave estrangeira.
- B** duplicar os dados mais acessados em diferentes tabelas para melhorar o desempenho.
- C** eliminar todos os relacionamentos entre tabelas para reduzir a complexidade da estrutura do banco de dados.
- D** reduzir a redundância e eliminar anomalias de inserção, atualização e exclusão.
- E** aumentar a velocidade de consulta por meio da replicação de dados redundantes.

Questão 41

Em um processo de normalização de banco de dados relacional, uma relação estará na terceira forma normal (3FN) se

- A** forem eliminadas todas as redundâncias, inclusive as dependências funcionais entre atributos não chave.
- B** permitir dependências transitivas, desde que envolvam apenas atributos não primos.
- C** não tiver nenhuma dependência funcional parcial em relação à chave candidata.
- D** estiver na segunda forma normal (2FN) e todos os atributos não primos forem dependentes transitivamente da chave primária.
- E** estiver na segunda forma normal (2FN) e todos os atributos não primos forem dependentes diretamente da chave primária.

Questão 42

Considerando que as formas normais organizam os dados em estruturas lógicas que favorecem a eficiência e a consistência, é correto afirmar que uma relação está na forma normal de *Boyce-Codd* (FNBC) se

- A** todos os seus determinantes forem chaves primárias candidatas.
- B** estiver na 3FN e não possuir atributos multivalorados.
- C** todos os atributos não primos forem dependentes diretamente da chave primária.
- D** toda dependência funcional nela contida tiver como determinante uma superchave.
- E** todas as dependências transitivas e parciais forem eliminadas, independentemente das chaves.

Questão 43

Uma empresa financeira armazena os dados de seus empréstimos em uma única tabela, nomeada como *Empréstimos*, com a seguinte estrutura simplificada: *Empréstimos* (`CPF_Cliente`, `Nome_Cliente`, `Codigo_Empréstimo`, `Data_Empréstimo`, `Valor`, `Nome_Agente`, `Email_Agente`). Durante as operações do sistema, observou-se que, para atualizar o nome de um agente de crédito, seria preciso modificar várias linhas, o que geraria inconsistência caso alguma linha não fosse atualizada corretamente.

Nessa situação hipotética, considerados os conceitos de normalização, o problema descrito está relacionado

- A** a atributos não atômicos, sendo resolvido ao se aplicar a 1FN.
- B** à decomposição sem preservação de dependência, sendo resolvido ao se aplicar a 5FN.
- C** a uma dependência funcional parcial, sendo resolvido ao se aplicar a 2FN.
- D** a uma dependência transitiva, sendo resolvido ao se aplicar a 3FN.
- E** a uma dependência multivalorada, sendo resolvido ao se aplicar a 4FN.

FUNDAMENTOS DE BANCO DE DADOS**Questão 44**

Assinale a opção em que é mencionada técnica ou estrutura de otimização de desempenho que permite ao SGBD acessar rapidamente registros específicos sem necessidade de examinar todos os dados de uma tabela.

- A** *backup* incremental
- B** índice
- C** normalização
- D** relacionamento
- E** controle de concorrência

Questão 45

A respeito das funcionalidades oferecidas por um SGBD, assinale a opção correta.

- Ⓐ A principal função de um SGBD é armazenar fisicamente os dados no disco.
- Ⓑ Um SGBD moderno implementa funcionalidades como controle de concorrência, integridade de dados, segurança e recuperação de falhas.
- Ⓒ O controle de integridade referencial em um banco de dados deve ser sempre tratado exclusivamente na camada da aplicação.
- Ⓓ A segurança e a integridade dos dados são responsabilidades exclusivas do administrador do sistema operacional.
- Ⓔ Um SGBD não possui mecanismos próprios de controle de concorrência, cabendo essa responsabilidade ao sistema operacional.

Questão 46

Em relação à organização física de dados e ao uso de estruturas auxiliares para acesso eficiente a bancos de dados relacionais, assinale a opção correta.

- Ⓐ Em um SGBD relacional, é utilizada exclusivamente varredura completa de tabelas (*full scan*) para se localizarem registros.
- Ⓑ A utilização de índices *bitmap* é restrita a bancos de dados baseados em grafos.
- Ⓒ O uso de índices prejudica o desempenho das consultas, pois força a leitura de toda a tabela.
- Ⓓ A criação de índices não afeta a forma como os dados são armazenados fisicamente em disco.
- Ⓔ Um índice clusterizado armazena fisicamente os dados em disco na mesma ordem dos valores do índice.

Questão 47

A arquitetura em três esquemas, proposta pelo modelo ANSI/SPARC, define níveis de abstração que separam a visão do usuário final, a estrutura lógica dos dados e a organização física do armazenamento. A respeito dessa arquitetura, assinale a opção correta.

- Ⓐ O nível interno da arquitetura lida com a representação física dos dados, como estruturas de armazenamento, ordenação e compactação.
- Ⓑ A arquitetura ANSI/SPARC é composta por quatro níveis: externo, conceitual, interno e transacional.
- Ⓒ O nível externo da arquitetura é responsável por definir a estrutura global e lógica dos dados no banco de dados.
- Ⓓ O nível interno representa a estrutura lógica central da arquitetura, compartilhada por todos os usuários do banco de dados.
- Ⓔ O nível conceitual da arquitetura descreve como os dados são fisicamente organizados em dispositivos de armazenamento.

Questão 48

Considerando que as transações devem obedecer ao conjunto de propriedades conhecido como ACID, assinale a opção correta.

- Ⓐ As propriedades ACID são relevantes em bancos de dados distribuídos e não se aplicam a bancos de dados relacionais centralizados.
- Ⓑ A propriedade de atomicidade permite que uma transação seja parcialmente concluída, desde que a maioria das operações tenha sido realizada.
- Ⓒ A consistência assegura que uma transação leve o banco de dados de um estado válido a outro também válido, mantendo as regras definidas no esquema.
- Ⓓ A propriedade de isolamento garante que as transações sejam sempre executadas em sequência, sem qualquer sobreposição temporal.
- Ⓔ A durabilidade assegura que os dados sejam mantidos em memória principal até que a transação seja encerrada.

Questão 49

Assinale a opção correta, a respeito de serializabilidade, um dos critérios fundamentais utilizados no controle de concorrência em transações.

- Ⓐ Esse critério é ignorado em sistemas que utilizam controle de concorrência baseado em bloqueios.
- Ⓑ Um escalonamento com conflitos de leitura sempre será não serializável.
- Ⓒ Um escalonamento é considerado serializável apenas se as transações forem executadas de forma completamente sequencial.
- Ⓓ O referido critério baseia-se na equivalência entre a execução concorrente e a execução sequencial de transações.
- Ⓔ Escalonamentos serializáveis não garantem a integridade dos dados em bancos de dados transacionais.

Questão 50

Os métodos de acesso utilizados em bancos de dados têm impacto direto no desempenho das consultas, destacando-se, entre os principais, as estruturas baseadas em árvores e o método de *hashing*. A esse respeito, assinale a opção correta.

- Ⓐ O uso de índices sempre impede a realização de varredura completa da tabela (*full scan*).
- Ⓑ *Hashing* e árvores B+ têm desempenho equivalente em qualquer tipo de consulta.
- Ⓒ O método de *hashing* é ideal para consultas que exigem ordenação dos resultados.
- Ⓓ Árvores B+ são inadequadas para realizar buscas sequenciais eficientes em grandes volumes de dados.
- Ⓔ Índices baseados em árvores B+ permitem acesso eficiente tanto para buscas pontuais quanto para intervalos ordenados.

Questão 51

Considerando que a modelagem conceitual com o modelo entidade-relacionamento é uma etapa essencial no projeto de bancos de dados relacionais, assinale a opção correta.

- Ⓐ A modelagem conceitual descreve como os dados serão fisicamente armazenados em disco.
- Ⓑ Uma entidade fraca é aquela que possui chave primária composta por mais de um atributo.
- Ⓒ A generalização permite que se abstraíam características comuns de entidades em uma entidade-pai mais genérica.
- Ⓓ Um relacionamento binário ocorre entre três ou mais entidades ao mesmo tempo.
- Ⓔ A agregação é usada para representar atributos multivalorados.

Questão 52

Em sistemas de gerenciamento de banco de dados, o controle de concorrência tem como objetivo garantir que múltiplas transações executadas simultaneamente não comprometam a integridade dos dados. Um dos principais protocolos utilizados nesses sistemas é o 2PL (*two-phase locking*). Acerca desse protocolo, assinale a opção correta.

- Ⓐ O protocolo 2PL estrito exige que todos os bloqueios sejam liberados após a fase de crescimento, antes do *commit*.
- Ⓑ O protocolo 2PL estrito garante serializabilidade ao manter os bloqueios até o final da transação.
- Ⓒ O 2PL básico impede completamente a ocorrência de interbloqueios (*deadlocks*).
- Ⓓ O protocolo 2PL não se aplica a bancos de dados que utilizam transações distribuídas.
- Ⓔ O uso de bloqueios pessimistas elimina a necessidade de controle de concorrência.

Questão 53

Assinale a opção correta acerca do protocolo 2PC (*two-phase commit*).

- Ⓐ O protocolo 2PC garante a atomicidade em transações distribuídas ao coordenar *commit* ou *abort* entre participantes.
- Ⓑ O referido protocolo elimina totalmente o risco de bloqueio em caso de falha do coordenador.
- Ⓒ Esse protocolo é utilizado apenas em sistemas baseados em bancos de dados relacionais centralizados.
- Ⓓ O uso desse protocolo impede a necessidade de *logs* nos participantes das transações.
- Ⓔ O protocolo 2PC exige que todos os participantes liberem seus bloqueios antes da fase de *commit*.

Questão 54

Para a garantia da consistência e da confiabilidade dos dados em um sistema de gerenciamento de banco de dados, as transações devem atender às propriedades ACID. Assim, na situação em que uma única transação envolva, simultaneamente, o débito de R\$ 100 na conta corrente e o crédito do mesmo valor na conta poupança de um mesmo correntista, caso ocorra falha de sistema antes da confirmação (*commit*) da transação, a garantia de que nenhum débito/crédito será efetivado é assegurada

- Ⓐ pelos bloqueios.
- Ⓑ pela atomicidade.
- Ⓒ pela consistência.
- Ⓓ pela durabilidade.
- Ⓔ pelo isolamento.

Questão 55

Em um ambiente de banco de dados de um sistema bancário, duas transações são executadas simultaneamente: uma delas adquire bloqueio exclusivo em A e, em seguida, em B; a outra adquire bloqueio exclusivo em B e, em seguida, em A. Ambas só liberam todos os bloqueios ao término da execução.

Nesse cenário, é mais provável que ocorra

- Ⓐ *starvation* da transação mais lenta, sendo a melhor prevenção a implementação de um escalonador baseado em prioridades.
- Ⓑ violação de isolamento, sendo a melhor prevenção a elevação do nível de isolamento para *serializable*.
- Ⓒ detecção de *deadlock* por *timeout*, sendo a melhor prevenção a imposição de limites de tempo para a aquisição de bloqueios.
- Ⓓ *deadlock*, sendo a melhor prevenção a imposição de uma ordenação total aos recursos, exigindo-se que todas as transações adquiram bloqueios em uma mesma sequência predefinida.
- Ⓔ condição de corrida (*race condition*), sendo a melhor prevenção a utilização de bloqueios compartilhados em vez de exclusivos.

Questão 56

Assinale a opção em que são corretamente apresentados os comandos necessários para a criação, em um banco de dados relacional, da tabela de nome funcionários, contendo as colunas: *id* de inteiros e chave primária; *nome* com VARCHAR(100), obrigatório; *salario* com DECIMAL(10,2); e *data_admissao* de DATE.

- Ⓐ CREATE funcionarios (id INTEGER PRIMARY KEY, nome VARCHAR(100) NOT NULL, salario DECIMAL(10,2), data_admissao DATE);
- Ⓑ INSERT TABLE funcionarios (id INTEGER PRIMARY KEY, nome VARCHAR(100) NOT NULL, salario DECIMAL(10,2), data_admissao DATE);
- Ⓒ CREATE TABLE funcionarios (id INTEGER PRIMARY KEY, nome VARCHAR(100) NOT NULL, salario DECIMAL(10,2), data_admissao DATE);
- Ⓓ DEFINE TABLE funcionarios (id INTEGER PRIMARY KEY, nome VARCHAR(100) NOT NULL, salario DECIMAL(10,2), data_admissao DATE);
- Ⓔ CREATE TABLE funcionarios (id INTEGER KEY, nome VARCHAR(100) REQUIRED, salario DECIMAL(10,2), data_admissao DATE);

Espaço livre

Questão 57

Assinale a opção em que é corretamente apresentado o comando DDL que permite criar a tabela de nome `itens_pedido` e relacionar cada pedido a seus produtos, de modo que: `pedido_id` e `produto_id` sejam chaves estrangeiras que referenciem a tabela; quantidade seja um inteiro maior que zero; `preco_unitario` seja um decimal maior que zero; e a combinação (`pedido_id`, `produto_id`) seja a chave primária, garantindo unicidade e não null.

- A** `CREATE TABLE itens_pedido (`
`pedido_id INTEGER REFERENCES pedidos(id),`
`produto_id INTEGER REFERENCES produtos(id),`
`quantidade INTEGER CHECK (quantidade > 0),`
`preco_unitario DECIMAL(10,2) CHECK`
`(preco_unitario > 0),`
`UNIQUE (pedido_id, produto_id)`
`);`
- B** `CREATE itens_pedido (`
`pedido_id INTEGER REFERENCES pedidos(id),`
`produto_id INTEGER REFERENCES produtos(id),`
`quantidade INTEGER CHECK quantidade > 0,`
`preco_unitario DECIMAL(10,2) CHECK`
`preco_unitario > 0,`
`PRIMARY KEY (pedido_id, produto_id)`
`);`
- C** `CREATE TABLE itens_pedido (`
`pedido_id INTEGER,`
`produto_id INTEGER,`
`quantidade INTEGER CHECK (quantidade > 0),`
`preco_unitario DECIMAL(10,2) CHECK`
`(preco_unitario > 0),`
`FOREIGN KEY (pedido_id) pedidos(id),`
`FOREIGN KEY (produto_id) produtos(id),`
`PRIMARY KEY (pedido_id, produto_id)`
`);`
- D** `CREATE TABLE itens_pedido (`
`pedido_id INTEGER REFERENCES pedidos(id),`
`produto_id INTEGER REFERENCES produtos(id),`
`quantidade INTEGER CHECK (quantidade > 0),`
`preco_unitario DECIMAL(10,2) CHECK`
`(preco_unitario > 0),`
`PRIMARY KEY (pedido_id, produto_id)`
`);`
- E** `CREATE TABLE itens_pedido (`
`item_id INTEGER PRIMARY KEY,`
`pedido_id INTEGER REFERENCES pedidos(id),`
`produto_id INTEGER REFERENCES produtos(id),`
`quantidade INTEGER CHECK (quantidade > 0),`
`preco_unitario DECIMAL(10,2) CHECK`
`(preco_unitario > 0),`
`UNIQUE (pedido_id, produto_id)`
`);`

Questão 58

Em um sistema de biblioteca, existem três tabelas com as seguintes colunas.

- livros:** com um identificador único (`id`) para cada livro, o título do livro, um campo `autor_id`, que indica o autor que cadastrou o livro, um campo `categoria_id`, que indica a que categoria o livro pertence, e um campo booleano disponível para indicar se o livro está disponível para empréstimo (`TRUE`) ou não (`FALSE`).
- autores:** com um identificador único (`id`) para cada autor, e o nome do autor.
- categorias:** com um identificador único (`id`) para cada categoria, e o nome da categoria.

Há uma relação implícita entre `livros.autor_id` e `autores.id` e entre `livros.categoria_id` e `categorias.id`, de modo que, para um livro válido, deve existir um autor correspondente e uma categoria correspondente.

A partir das informações precedentes, assinale a opção em que é corretamente apresentada a consulta SQL que permite a obtenção apenas dos livros cujo campo `disponivel` seja `TRUE`, desde que já existam registros correspondentes em `autores` e `categorias`, e, além disso, filtre somente os livros cuja categoria tenha o nome `Ficção Científica`, e apresente como resultado exatamente três colunas: título do livro, nome do autor e nome da categoria.

- A** `SELECT titulo, nome, nome`
`FROM livros l`
`INNER JOIN autores a ON l.autor_id = a.id`
`INNER JOIN categorias c ON l.categoria_id = c.id`
`WHERE disponivel = TRUE`
`AND c.nome = 'Ficção Científica';`
- B** `SELECT l.titulo, a.nome, c.nome`
`FROM livros l`
`LEFT JOIN autores a`
`ON l.autor_id = a.id`
`LEFT JOIN categorias c`
`ON l.categoria_id = c.id`
`WHERE l.disponivel = TRUE`
`AND c.nome = 'Ficção Científica';`
- C** `SELECT l.titulo, a.nome, c.nome`
`FROM livros l, autores a, categorias c`
`WHERE l.autor_id = a.id`
`AND l.categoria_id = c.id`
`AND l.disponivel = TRUE`
`AND c.nome LIKE '%Ficção Científica%';`
- D** `SELECT l.titulo, a.nome, c.nome`
`FROM livros AS l`
`INNER JOIN autores AS a`
`ON l.autor_id = a.id`
`INNER JOIN categorias AS c`
`ON l.categoria_id = c.id`
`WHERE l.disponivel = 'S'`
`AND c.nome = 'Ficção Científica';`
- E** `SELECT l.titulo, a.nome AS autor, c.nome AS categoria`
`FROM livros AS l`
`INNER JOIN autores AS a`
`ON l.autor_id = a.id`
`INNER JOIN categorias AS c`
`ON l.categoria_id = c.id`
`WHERE l.disponivel = TRUE`
`AND c.nome = 'Ficção Científica';`

Questão 59

Considerando que, em um sistema de vendas, haja a tabela de nome `vendas_completo(vendedor, regioao, valor)`, em que cada linha registre uma venda feita por determinado vendedor em sua região, assinale a opção em que é corretamente apresentada a consulta SQL que, sem usar JOIN, apresenta como resultado exatamente as seguintes quatro colunas: `vendedor`; `regiao`; `total_vendas`, correspondente à soma de valor por vendedor; e `ranking_regiao`, de modo que empates estejam na mesma posição e “pulem” postos.

- A**

```
SELECT DISTINCT
  vendedor,
  regioao,
  SUM(valor) OVER (PARTITION BY vendedor,
    regioao) AS total_vendas,
  RANK() OVER (
    PARTITION BY regioao
    ORDER BY SUM(valor) OVER (PARTITION BY
      vendedor, regioao) DESC
  ) AS ranking_regiao
FROM vendas_completo;
```
- B**

```
SELECT
  vendedor,
  regioao,
  SUM(valor) AS total_vendas,
  RANK() OVER (
    PARTITION BY regioao
    ORDER BY SUM(valor) DESC
  ) AS ranking_regiao
FROM vendas_completo
GROUP BY vendedor, regioao;
```
- C**

```
SELECT
  vendedor,
  regioao,
  SUM(valor) OVER (PARTITION BY vendedor) AS
total_vendas,
  DENSE_RANK() OVER (
    PARTITION BY regioao
    ORDER BY SUM(valor) OVER (PARTITION BY
      vendedor) DESC
  ) AS ranking_regiao
FROM vendas_completo
GROUP BY vendedor, regioao;
```
- D**

```
SELECT DISTINCT
  vendedor,
  regioao,
  SUM(valor) OVER (PARTITION BY vendedor) AS
total_vendas,
  ROW_NUMBER() OVER (
    PARTITION BY regioao
    ORDER BY SUM(valor) OVER (PARTITION BY
      vendedor) DESC
  ) AS ranking_regiao
FROM vendas_completo;
```
- E**

```
SELECT
  vendedor,
  regioao,
  SUM(valor) OVER (PARTITION BY vendedor,
    regioao) AS total_vendas,
  ROW_NUMBER() OVER (
    PARTITION BY regioao
    ORDER BY SUM(valor) OVER (PARTITION BY
      vendedor, regioao) DESC
  ) AS ranking_regiao
FROM vendas_completo;
```

Questão 60

Em bancos de dados relacionais, *stored procedures*

- I podem ter parâmetros de entrada (IN), saída (OUT) e de entrada/saída (INOUT).
- II permitem agrupar múltiplas instruções DML (INSERT, UPDATE, DELETE) e controle de fluxo (IF, LOOP, EXCEPTION).
- III apresentam diretamente conjuntos de resultados (tabelas) sempre que são executados.
- IV podem ser chamados dentro de uma cláusula SELECT como se fossem funções escalares.

Estão certos apenas os itens

- A** I e II.
- B** I e III.
- C** II e IV.
- D** I, III e IV.
- E** II, III e IV.

Questão 61

Em sistemas de gerenciamento de banco de dados relacionais, uma *view*

- A** continuará executando e retornando os resultados calculados previamente, ainda que a tabela-base seja dela removida.
- B** pode receber privilégios de SELECT independentes da(s) tabela(s)-base, permitindo que usuários consultem apenas colunas específicas sem ter acesso pleno às tabelas originais.
- C** não armazena dados fisicamente e só reflete alterações nos dados das tabelas-base quando for redefinida.
- D** impede a exposição de colunas de uma tabela, mas não impede a exposição de linhas específicas que atendam a algum critério.
- E** é definida por meio de consulta simples em uma única tabela, sem poder incluir junções ou subconsultas.

Questão 62

Em sistemas de gerenciamento de banco de dados relacionais, *triggers*

- A** BEFORE TRUNCATE são executados antes de apagar as linhas e permitem cancelar a operação lançando erro.
- B** AFTER DELETE são acionados antes de remover fisicamente a linha, de modo que, durante seu processamento, a linha ainda permanece na tabela.
- C** BEFORE INSERT disparam antes de alocar espaço para a nova linha e podem ajustar valores usando NEW, ao passo que triggers AFTER INSERT disparam somente depois de todas as restrições terem sido verificadas e a linha ter sido gravada permanentemente.
- D** BEFORE UPDATE são executados depois de atualizar os valores, porém antes de reindexar a tabela, permitindo reverter a operação caso alguma condição falhe.
- E** AFTER TRUNCATE permitem o acesso a dados antigos das linhas por meio de uma pseudotabela OLD, possibilitando registrar o que foi apagado.

ADMINISTRAÇÃO DE BANCO DE DADOS RELACIONAIS

Questão 63

Em bancos de dados relacionais, o comando `GRANT` permite a definição de colunas específicas para as instruções de

- A `CREATE`, `DELETE` e `INSERT`.
- B `DELETE`, `SELECT` e `UPDATE`.
- C `DELETE`, `INSERT` e `SELECT`.
- D `INSERT`, `SELECT` e `UPDATE`.
- E `CREATE`, `INSERT` e `UPDATE`.

Questão 64

No projeto de um SGBD relacional, devem ser implementadas diversas funcionalidades, como a independência de dados, que consiste em

- A não permitir acesso não autorizado.
- B permitir que usuários simultâneos operem os dados.
- C não precisar de alguma aplicação para entender os dados.
- D fornecer código para acessar e estruturar os dados.
- E impedir um identificador duplicado ou um valor fora da lista.

Questão 65

Na política de *backup*, o procedimento que permite recuperar o banco de dados em um ponto específico do tempo é conhecido como

- A *log recovery*.
- B *full restore*.
- C *incremental backup*.
- D *automatic backup*.
- E *point-in-time recovery*.

Questão 66

Os três principais elementos de um banco de dados relacional são

- A tabelas, colunas e linhas.
- B tabelas, índices e colunas.
- C colunas, índices e *views*.
- D colunas, linhas e índices.
- E índices, tabelas e linhas.

Questão 67

De acordo com a técnica de duas fases para controle de concorrência, o bloqueio que possui os estados de `read_lock`, `write_lock` e `unlock` é o bloqueio

- A de protocolo.
- B binário.
- C de conversão.
- D compartilhado.
- E unitário.

Questão 68

No processo de *backup*, a etapa que consiste em copiar e transferir grandes volumes de informações de um sistema para outro é denominada

- A *export*.
- B *create*.
- C *dump*.
- D *log*.
- E *copy*.

Questão 69

Nos bancos de dados relacionais, a estrutura que armazena objetos, tais como tabelas e índices, é conhecida como

- A *file*.
- B *site*.
- C *metadata*.
- D *tablespace*.
- E *directory*.

BANCO DE DADOS

Questão 70

Por padrão, a replicação do MySQL v8 é

- A por evento.
- B síncrona.
- C semissíncrona.
- D assíncrona.
- E por *trigger*.

Questão 71

Para um banco de dados MySQL com oito bases, sendo uma máster e as demais réplicas, o sucesso de uma replicação semissíncrona é garantido quando o número de réplicas atualizadas é, pelo menos, igual a

- A sete.
- B um.
- C seis.
- D dois.
- E quatro.

Questão 72

No Oracle 19c, uma das vantagens da utilização de mais *tablespaces* consiste

- A na separação dos dados de aplicativos diferentes.
- B na garantia da atomicidade dos dados na base.
- C na redução do uso de espaço em disco.
- D no aumento da disponibilidade dos dados em cada *tablespace*.
- E na diminuição da necessidade de controle.

Questão 73

No banco de dados Oracle, a ferramenta que permite a modelagem compatível objeto/relacional denomina-se

- A XML Parser.
- B Multimodel Design.
- C Object/Relational Model Tool.
- D Extended Modeler.
- E JSON Relational Duality.

Questão 74

O MS SQL Server possui diversos componentes embutidos, entre eles, o MSL, que é responsável por

- A distribuir dados de um banco de dados para outro e sincronizá-los entre os bancos de dados para manter a consistência.
- B fornecer acesso controlado e processamento de transações para atender aos requisitos dos aplicativos de consumo de dados mais exigentes.
- C oferecer a funcionalidade de relatórios corporativos habilitados para a Web.
- D oferecer suporte à integração de aprendizado de máquina em fluxos de trabalho corporativos.
- E criar soluções de integração de dados de alto desempenho.

PERFORMANCE E TUNING**Questão 75**

Com base nos conceitos relacionados a índices do tipo B-tree, assinale a opção correta.

- A** O índice B-tree não se adapta bem a operações de inserção ou atualização, sendo mais indicado para bases de dados estáticas, nas quais a estrutura da tabela não sofre alterações frequentes.
- B** O uso do índice B-tree é vantajoso principalmente porque evita a utilização de memória adicional, pois esse índice não mantém ponteiros para os registros das tabelas.
- C** O índice B-tree possui estrutura de árvore desbalanceada, o que facilita inserções em tabelas de grande volume de dados e permite acelerar buscas sequenciais.
- D** Recomenda-se o uso de índices B-tree principalmente quando se pretende recuperar um grande volume de registros, equivalente a mais de 50% do total da tabela, pois eles melhoram a *performance* nesses casos.
- E** Um índice B-tree permite que, em determinadas consultas, os dados sejam recuperados diretamente do índice, sem necessidade de se acessar a tabela, quando todas as colunas requeridas estão presentes no índice.

Questão 76

Em um sistema bancário, existem duas tabelas: *clientes* (*id_cliente*, *nome*); e *contas* (*id_conta*, *id_cliente*, *saldo*). Pretende-se obter uma lista que contenha o *id_conta* e o *nome* dos *clientes* que possuem conta associada, incluídos somente os registros em que haja saldo informado na conta (ou seja, *saldo* não nulo).

Com base nessa situação hipotética, assinale a opção que corresponde à consulta correta para exibir os dados solicitados.

- A**

```
SELECT c.nome, ct.id_conta
FROM clientes c
LEFT JOIN contas ct ON ct.id_cliente =
c.id_cliente
WHERE ct.saldo IS NOT NULL
```
- B**

```
SELECT ct.id_conta
FROM contas ct
LEFT JOIN clientes c ON c.id_cliente =
ct.id_cliente
```
- C**

```
SELECT ct.id_conta, c.nome
FROM contas ct
LEFT JOIN clientes c ON ct.id_cliente =
c.id_cliente
WHERE ct.saldo IS NOT NULL
```
- D**

```
SELECT c.nome, ct.id_conta
FROM contas ct
LEFT JOIN clientes c ON ct.id_cliente =
c.id_cliente
WHERE ct.saldo IS NOT NULL
```
- E**

```
SELECT c.nome, ct.id_conta
FROM contas ct
LEFT JOIN clientes c ON ct.id_cliente =
c.id_cliente
WHERE ct.saldo IS NULL
```

Questão 77

Em certo banco, há as tabelas *clientes* e *conta*. Os dados a seguir foram salvos nessas tabelas no banco de dados.

Tabela *cliente*

<i>id_cliente</i>	<i>nome</i>
1	Maria Souza
2	João Lima
3	Ana Torres

Tabela *conta*

<i>id_conta</i>	<i>id_cliente</i>	<i>saldo</i>
1	1	1500
2	1	500
3	2	900
4	3	2000

A partir dos dados informados na situação hipotética precedente, assinale a opção em que é apresentada a consulta SQL que exibirá os nomes dos clientes que possuem pelo menos uma conta com saldo superior a R\$ 1.000.

- A**

```
SELECT nome
FROM clientes c
WHERE EXISTS (
    SELECT 1 FROM contas ct
    WHERE ct.id_cliente = c.id_cliente AND
    ct.saldo > 1000)
```
- B**

```
SELECT nome
FROM clientes c
WHERE NOT EXISTS (
    SELECT * FROM contas ct
    WHERE ct.id_cliente = c.id_cliente AND
    ct.saldo < 1000)
```
- C**

```
SELECT nome
FROM clientes c
JOIN contas ct ON c.id_cliente = ct.id_cliente
WHERE ct.saldo > 1000
```
- D**

```
SELECT DISTINCT nome
FROM clientes c
JOIN contas ct ON c.id_cliente = ct.id_cliente
WHERE ct.saldo >= 1000
```
- E**

```
SELECT nome
FROM clientes c
WHERE EXISTS (
    SELECT 1 FROM contas ct
    WHERE ct.id_cliente = c.id_cliente AND
    ct.saldo >= 1000)
```

Questão 78

Em um sistema de gestão bancária, pretende-se identificar os clientes que possuem saldo total (somadas todas as suas contas) superior à média dos saldos das contas da agência em que cada conta está vinculada. Nesse sistema, as tabelas `clientes` e `contas` foram criadas pelos comandos a seguir.

```
CREATE TABLE clientes (
  id_cliente INT PRIMARY KEY,
  nome VARCHAR(50));

CREATE TABLE contas (
  id_conta INT PRIMARY KEY,
  id_cliente INT,
  id_agencia INT,
  saldo NUMERIC(10,2),
  FOREIGN KEY (id_cliente) REFERENCES
  clientes(id_cliente))
```

Considerando o uso de subconsulta correlacionada para o atendimento da demanda, assinale a opção que corresponde à consulta SQL que retornará o que se pretende na situação hipotética precedente.

- A**

```
SELECT DISTINCT c.nome
FROM clientes c
JOIN contas ct ON c.id_cliente = ct.id_cliente
WHERE (
  SELECT SUM(ct2.saldo)
  FROM contas ct2
  WHERE ct2.id_cliente = ct.id_cliente
) > (
  SELECT AVG(ct3.saldo)
  FROM contas ct3
  WHERE ct3.id_agencia = ct.id_agencia)
```
- B**

```
SELECT c.nome
FROM clientes c
JOIN contas ct ON c.id_cliente = ct.id_cliente
WHERE ct.saldo > (
  SELECT AVG(saldo) FROM contas WHERE
  id_agencia = ct.id_agencia)
```
- C**

```
SELECT nome
FROM clientes
WHERE id_cliente IN (
  SELECT id_cliente
  FROM contas
  GROUP BY id_cliente
  HAVING SUM(saldo) > (
    SELECT AVG(saldo) FROM contas ))
```
- D**

```
SELECT c.nome
FROM clientes c
JOIN contas ct ON c.id_cliente = ct.id_cliente
WHERE ct.saldo > ALL (
  SELECT saldo FROM contas WHERE id_agencia =
  ct.id_agencia)
```
- E**

```
SELECT c.nome
FROM clientes c
JOIN contas ct ON c.id_cliente = ct.id_cliente
WHERE (
  SELECT SUM(saldo) FROM contas WHERE
  id_cliente = c.id_cliente
) > (
  SELECT MAX(saldo) FROM contas WHERE
  id_agencia = ct.id_agencia)
```

Questão 79

Assinale a opção correta acerca das variáveis *bind* em sistemas gerenciadores de bancos de dados relacionais.

- A** Variáveis *bind* são utilizadas apenas em instruções DDL, como `CREATE` e `ALTER`, para parametrizar a criação de objetos no banco de dados.
- B** Variáveis *bind* são utilizadas exclusivamente dentro de *triggers*, para armazenar o valor antigo e o novo de uma coluna afetada por uma atualização.
- C** Variáveis *bind* permitem que comandos SQL sejam escritos com parâmetros genéricos, o que evita a necessidade de se reescrever a instrução sempre que o valor do parâmetro mudar.
- D** O uso de variáveis *bind* reduz o desempenho das consultas SQL, pois impede que o SGBD reutilize planos de execução previamente gerados.
- E** As variáveis *bind* substituem valores literais diretamente na *string* SQL antes de sua execução, realizando a interpolação de valores de forma automática.

Questão 80

No que se refere a planos de execução (*explain plan*) em bancos de dados relacionais, assinale a opção correta.

- A** A geração de um plano de execução é um processo manual e demorado, que requer que o analista defina explicitamente os índices que serão utilizados.
- B** O plano de execução inclui informações sobre métodos de junção entre tabelas, mas não sobre filtros, ordenações ou agrupamentos.
- C** O plano de execução descreve a sequência de passos definida pelo otimizador, incluindo métodos de acesso, junção e operações de dados, o que permite ao analista identificar oportunidades de otimização.
- D** O plano de execução é utilizado apenas em consultas que envolvem múltiplas tabelas, não se aplicando a instruções simples com apenas uma tabela.
- E** O otimizador de consultas executa sempre todas as alternativas de plano, selecionando a que apresentar o menor tempo de execução real.

Espaço livre