

Questão: 2

O que são transações no SQL Server e qual a importância do controle de concorrência no processamento de transações?

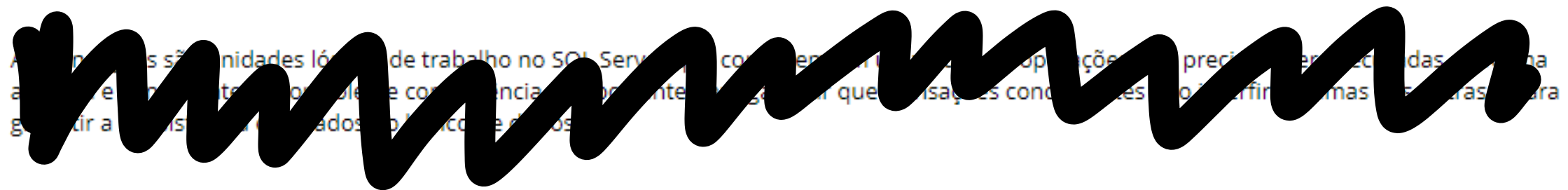
Sua Resposta:



Correções:



Resposta correta (sugestão):



Questão: 3

Marque VERDADEIRO ou FALSO para as afirmações abaixo e depois escolha a alternativa que corresponde a sequência correta:

- ☐ O shared lock, em uma transação, permite que várias outras transações leiam os mesmos dados ao mesmo tempo, mas impede que outras transações atualizem ou excluam esses dados.
- ☐ O exclusive lock, em uma transação, impede que outras transações acessem ou alterarem os dados dessa transação.
- ☐ Quando um objeto dentro de uma transação recebe um exclusive lock, é possível liberar o lock deste objeto sem que a transação tenha sido finalizada.
- ☐ Quando um objeto dentro de uma transação recebe um shared lock, é possível liberar o lock deste objeto sem que a transação tenha sido finalizada.
- ☐ Os shared lock e exclusive lock só são liberado quando uma transação (implícita ou explícita) é finalizada.

- ☒ a) V, V, V, V, V
- ☐ b) V, V, V, F, V
- ☐ c) V, V, F, V, V
- ☐ d) V, V, F, F, V

Questão: 4

Como o uso de índices pode ajudar a minimizar problemas de bloqueio e melhorar a performance no SQL Server? Explique com suas palavras.

Sua Resposta:

[Redacted answer text]

Correções:

[Redacted correction text]

Resposta correta (sugestão):

[Redacted correct answer text]

Questão: 5

Marque VERDADEIRO ou FALSO para as afirmações abaixo e depois escreva a sua resposta na sequencia correta referente à junções de tabelas em bancos de dados:

- ☐ O INNER JOIN retorna todas as linhas de ambas as tabelas que têm valores correspondentes nas colunas de junção especificadas.
- ☐ O LEFT JOIN entre duas tabelas retorna todas as linhas da tabela à esquerda e as linhas correspondentes da tabela à direita. Se não houver correspondência, NULL é retornado para as colunas da tabela à direita.
- ☐ O RIGHT JOIN é similar a um LEFT JOIN, mas ele inverte a ordem das tabelas, retornando todas as linhas da tabela à direita e as linhas correspondentes da tabela à esquerda.
- ☐ O FULL JOIN entre duas tabelas retorna todas as linhas de ambas as tabelas, incluindo as que não têm correspondência. As colunas da tabela sem correspondência são preenchidas com NULL.
- ☐ O CROSS JOIN (também conhecido como produto cartesiano) combina cada linha da primeira tabela com todas as linhas da segunda tabela, criando um conjunto de resultados que contém todas as combinações possíveis entre os registros das duas tabelas, sem levar em consideração qualquer critério de correspondência específico.

Sua Resposta:

☐
☐
☐
☐
☐

Questão: 6

Marque VERDADEIRO ou FALSO para as afirmações abaixo e depois escreva sua resposta na sequência correta referente a índices em bancos de dados:

- ☐ É possível ter mais um índice clustered por tabela, desde que a chave do índice seja diferente.
- ☐ Não é possível criar dois índices nonclustered usando a mesma chave (mesmo coluna para indexação), mesmo que tenham nomes de índices diferentes.
- ☐ Um índice clustered pode ter mais de uma coluna como chave.
- ☐ A sintaxe - *create index idx_data_competicao on resultado_natacao (data_competicao)* - cria um índice chamado data_competicao na tabela resultado_natacao, usando a coluna idx_data_competicao como chave deste índice.
- ☐ No nível folha de um índice clustered, encontramos todos as colunas da tabela, independente da(s) coluna(s) escolhida(s) como chave.
- ☐ No nível folha de um índice nonclustered, encontramos todos as colunas da tabela, independente da(s) coluna(s) escolhida(s) como chave.

Sua Resposta:

- ☒ É possível ter mais um índice clustered por tabela, desde que a chave do índice seja diferente.
- ☒ Não é possível criar dois índices nonclustered usando a mesma chave (mesmo coluna para indexação), mesmo que tenham nomes de índices diferentes.
- ☒ Um índice clustered pode ter mais de uma coluna como chave.
- ☒ A sintaxe - *create index idx_data_competicao on resultado_natacao (data_competicao)* - cria um índice chamado data_competicao na tabela resultado_natacao, usando a coluna idx_data_competicao como chave deste índice.
- ☒ No nível folha de um índice clustered, encontramos todos as colunas da tabela, independente da(s) coluna(s) escolhida(s) como chave.
- ☒ No nível folha de um índice nonclustered, encontramos todos as colunas da tabela, independente da(s) coluna(s) escolhida(s) como chave.

Correções:

Resposta correta:

- ☒ É possível ter mais um índice clustered por tabela, desde que a chave do índice seja diferente.
- ☒ Não é possível criar dois índices nonclustered usando a mesma chave (mesmo coluna para indexação), mesmo que tenham nomes de índices diferentes.
- ☒ Um índice clustered pode ter mais de uma coluna como chave.
- ☒ A sintaxe - *create index idx_data_competicao on resultado_natacao (data_competicao)* - cria um índice chamado data_competicao na tabela resultado_natacao, usando a coluna idx_data_competicao como chave deste índice.
- ☒ No nível folha de um índice clustered, encontramos todos as colunas da tabela, independente da(s) coluna(s) escolhida(s) como chave.
- ☒ No nível folha de um índice nonclustered, encontramos todos as colunas da tabela, independente da(s) coluna(s) escolhida(s) como chave.

Questão: 3

Você foi chamado para avaliar o código abaixo:

```
DECLARE @cor int

SET @cor = 'AZUL'

UPDATE carro
    SET modelo = 'ECO SPORT'
WHERE chassi IS NOT NULL
    AND ano >= 2020 AND ano <= 2022
    AND cor == @cor

PRINT 'O número de linhas atualizadas é : ' + CONVERT(varchar(100), @@ROWCOUNT)
```

Analisando o código acima, marque as afirmações falsas de acordo com a sua análise para correção do código.

- ☒ a) SET @cor, pois o correto seria setar o valor da variável através do comando SELECT.
- ☒ b) DECLARE @cor int, pois o tipo de dado correto deveria ser alfanumérico tipo varchar.
- ☒ c) ano >= 2020 AND ano <= 2022, pois eles deveriam ser escritos da seguinte forma 2020 <= ano <= 2022.
- ☒ d) CONVERT, por ser um comando completamente desnecessário nessa instrução.
- ☒ e) IS NOT NULL, pois ele deveria ser escrito dessa forma NOT IS NULL.
- ☒ f) @@ROWCOUNT, pois deveria ser escrito com somente um caracter ROWCOUNT().

Questão: 6

Uma trigger, uma stored procedure e uma função, RESPECTIVAMENTE, podem suportar:

- ☐ a) Somente parametros de saída; Parametros de entrada e saída; Somente parametros de entrada. [REDACTED]
- ☐ b) Todas as três suportam parâmetros de entrada. [REDACTED]
- ☒ c) Nenhum parâmetro; Parametros de entrada e saída; Somente parametros de entrada. [REDACTED]
- ☐ d) Nenhum parâmetro; Somente parametros de entrada; Somente parametros de entrada. [REDACTED]

Questão: 7

Sobre os temas de procedimentos armazenados (stored procedures) e gatilhos (triggers) e funções (functions), marque somente as afirmações FALSAS.

- ☒ a) O controle transacional (begin tran, commit e rollback) não pode ser aplicado em funções, somente em stored procedures e triggers.
- ☒ b) O uso de stored procedure aumenta o tráfego na rede em uma arquitetura client/server, pois todos os comandos SQL dentro da stored procedure são encaminhados via rede entre a aplicação e o banco de dados.
- ☒ c) A execução de uma função é feita por meio de chamada ao seu nome, podendo ser, inclusive, dentro de uma stored procedure.
- ☒ d) Uma trigger é equivalente a uma stored procedure, inclusive pelo fato de retornar um valor.

Questão: 8

Você precisa criar algo que guarde todas as alterações ocorridas nos dados das tabelas, especialmente os comandos de DML (INSERT, UPDATE e DELETE) para uma eventual auditoria. A estratégia mais apropriada para detectar comandos dessa natureza, considera a criação e uso de:

- ☐ a) Controle de concorrência. [REDACTED]
- ☐ b) Índices [REDACTED]
- ☐ c) Triggers. [REDACTED]
- ☐ d) Stored Procedure. [REDACTED]
- ☐ e) Funções. [REDACTED]