

Banco de Dados II - Lista de Fixação de Conteúdo

Questão 1: Dado o banco Northwind, faça uma transação que aumente o preço do produto Tofu (productid = 14) em 20% a partir de 05/06/1996 e altere os pedidos desse dia em diante (tabela order_details).

```
START TRANSACTION;  
UPDATE northwind.order_details  
SET unitprice = unitprice + (unitprice * 0.20)  
WHERE orderid IN  
(SELECT o.orderid  
FROM northwind.orders o JOIN northwind.order_details d  
ON o.orderid = d.orderid  
WHERE d.productid = 14 AND orderdate >= '1996-06-05 00:00:00');  
COMMIT;
```

Questão 2:

a) Crie o atributo qtdProdutos na tabela orders. O atributo deve ser inteiro e ter o valor default zero;

```
ALTER TABLE northwind.orders ADD COLUMN qtdProdutos int DEFAULT 0
```

b) Crie o atributo maisDesconto na tabela orders. O atributo deve ser inteiro e ter o valor default zero;

```
ALTER TABLE northwind.orders ADD COLUMN maisDesconto int DEFAULT 0
```

c) Atualize o atributo qtdProdutos da tabela orders com a quantidade de produtos diferentes comprados naquele pedido. Obs.: não é a soma dos itens comprados, é a quantidade de diferentes produtos adquiridos no pedido;

Questão 3: Considere:

I. Se uma transação é concluída com sucesso (operação commit bem sucedida), então seus efeitos são persistentes.

II. Ou todas as ações da transação acontecem, ou nenhuma delas acontece.

As propriedades (I) e (II) das transações em SGBDs, significam, respectivamente:

a) durabilidade e consistência.

b) persistência e automação.

c) isolamento e atomicidade.

d) durabilidade e atomicidade.

e) consistência e persistência.

Durabilidade: Os efeitos de uma transação em caso de sucesso (commit) devem persistir no banco de dados mesmo em casos de quedas de energia, travamentos ou erros. Garante que os dados estarão disponíveis em definitivo.

Atomicidade: A transação deve ter todas as suas operações executadas em caso de sucesso ou, em caso de falha, nenhum resultado de alguma operação refletido sobre a base de dados. Ou seja, após o término de uma transação (commit ou rollback), a base de dados não deve refletir resultados parciais da transação.

Questão 4: Sejam as transações T1, T2 e T3, as quais desempenham as seguintes operações:

T1 : Adicionar 1 ao A

T2 : Duplicar A

T3 : Exibir A na tela (crie uma função para isso), e então fixar A em um

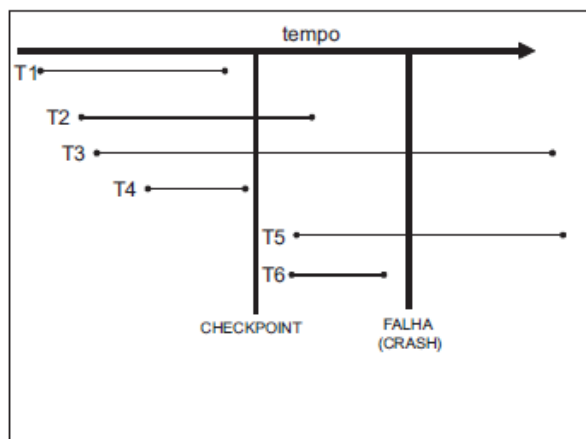
a) Como seriam essas transações em função das operações read e write?

T1 READ(X,A) A := A+1 WRITE(X,A)	T2 READ(X,A) A := A*2 (ou A=A+A) WRITE(X,A)	T3 READ(X,A) EXIBIR(A) A := 1 WRITE(X,A)
---	--	--

b) Suponha que as transações T1, T2 e T3 possam ser executadas concorrentemente. Se A tiver um valor inicial de zero e as transações forem executadas de forma serial, quantos resultados corretos possíveis teremos? Listá-los.

SERIAL Transações	Valor A
T1→T2→T3	0→1→2→1
T1→T3→T2	0→1→1→2
T2→T1→T3	0→0→1→1
T2→T3→T1	0→0→1→2
T3→T1→T2	0→1→2→4
T3→T2→T1	0→1→2→3

Questão 5: Considerando as técnicas e os procedimentos de recuperação em caso de falhas, o que acontece com as transações T1, T2, T3, T4, T5 e T6, que estão sendo executadas em um Sistema Gerenciador de Banco de Dados, conforme apresentado na figura, após a falha, se:



a) a técnica de recuperação usada for a de atualização imediata

T1 - Transação Salva no banco, nada acontece

T2 – Transação Salva no banco, nada acontece

T3 – Transação Não finalizada - UNDO
T4 - Transação Salva no banco, nada acontece
T5 – Transação Não finalizada - UNDO
T6– Transação Salva no banco, nada acontece

b) a técnica de recuperação usada for a de atualização adiada

T1 - Transação Salva no banco, nada acontece
T2 – Transação Salva no banco será refeita, REDO
T3 – Transação Não finalizada - UNDO
T4 - Transação Salva no banco, nada acontece
T5 – Transação Não finalizada - UNDO
T6– Transação Salva no banco será refeita - REDO

Questão 6: Considere um disco com bloco de 512 bytes. Um ponteiro para bloco de 6 bytes e um ponteiro para registro de 7 bytes. Um arquivo contém 75000 registros de tamanho fixo de FILMES. Cada registro tem os seguintes atributos : FilmeID (4 bytes), DiretorID (4 bytes), CategoriaID (4 bytes), Título (35 bytes), AnoLancamento(4 bytes), Duracao (4 bytes), Resenha (50 bytes). Um byte adicional é usado como marcador de deleção do registro. O atributo FilmeID é a chave primária da tabela. O atributo Título é uma chave candidata.

a) Suponha que o arquivo é sequencial. Calcule:

i. O tamanho do índice primário.

ii. O tamanho do índice sobre o atributo AnoLancamento.

b) Suponha que o arquivo é heap. Calcule: i. O tamanho do índice primário.

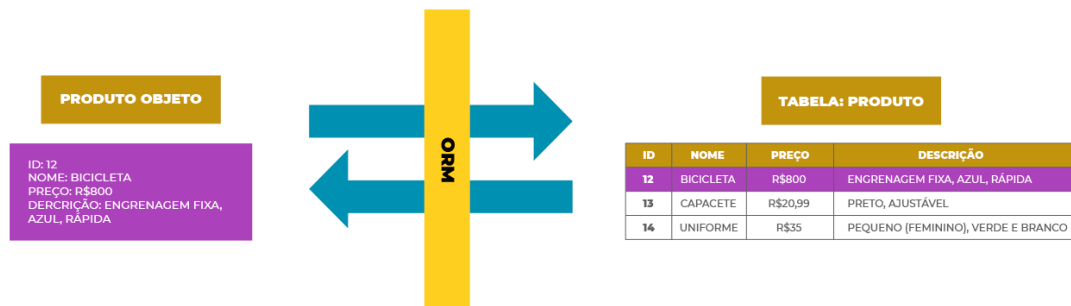
ii. O tamanho do índice sobre o atributo Título.

iii. Qual alternativa pode ser utilizada para reduzir o tamanho do índice sobre o atributo Título?

c) Por que podemos ter no máximo um índice primário ou *clustering*, mas diversos índices secundários para um arquivo de dados sequencial?

d) Ilustre um cenário que use o índice primário criado e outro que não use. Gere as consultas SQL nos dois casos. **OBS.** Em ambos os casos, a consulta deve utilizar a PK.

Questão 7: Considerando os conceitos aprendidos na disciplina de banco de dados, discorra sobre a figura abaixo.



Muitas aplicações são desenvolvidas em linguagens orientadas a objetos. No entanto, a maioria dos bancos de dados são implementados com banco de dados relacionais.

Então, existe uma dificuldade em persistir dados de uma API OO no BDR, que não conhece o conceito de objeto, já que as linguagens OO são semanticamente mais ricas que o MR. Essa incompatibilidade entre os modelos se chama Impedância.

Os dados da API devem ser salvos corretamente no BD, para isso os objetos devem ser destrinchados em tabelas antes de serem salvos no BD. A persistência define como os dados serão armazenados no BD.

Para reduzir a impedância existem várias formas de representar uma estrutura OO no MR. Entre essas técnicas, existe o Modelo Objeto-relacional (ORM) que é a persistência (lógica e física) automatizada e transparente dos objetos em uma API OO para as tabelas de um BDR utilizando metadados, que descrevem o mapeamento entre os objetos e o BD.

Questão 8: Descreva o que cada função abaixo faz no SGBD:

- a) Input: **entrada de dados no banco.**
- b) Output: **saída de dados do banco.**
- c) Read: **Lê dado do arquivo salvo no banco.**
- d) Write: **Escreve no arquivo salvo no banco.**
- e) Commit: **Faz as modificações executadas no banco serem persistentes.**
- f) Rollback: **cancela as modificações executadas no banco antes de um commit.**
- g) Savepoint: **cria um ponto de salvamento no meio de uma transação.**

Questão 9: Considere a relação EMPREGADO (NumeroEmp, RG, nome, sobrenome, salario, endereço, departamento), em que o atributo grifado corresponde à chave primária da relação. Considerando o SGBD PostgreSQL e a relação EMPREGADO descrita:

a) Escreva em SQL os comandos que devem ser utilizados para implementar corretamente a segurança na tabela EMPREGADOS, de forma que o usuário A1 tenha apenas direito de realizar consultas nesta tabela; o usuário A2, além de consultas, possa também realizar inclusões, alterações e exclusões de dados, além de propagar os privilégios para outros usuários; o usuário A3 tem acesso de consulta ao total de empregados por departamento. Considere que estes usuários ainda não possuam nenhum privilégio sobre a tabela EMPREGADOS.

GRANT SELECT * ON TABLE empregados TO A1

GRANT ALL ON TABLE empregados TO A2 WITH GRANT OPTION

GRANT SELECT departamento ON TABLE empregados TO A3

b) Qual o nível de segurança implementado na letra (a)? Tal nível é suficiente para garantir a segurança do banco?

Não pois o correto seria criar um grupo de perfil de usuário dar os privilégios para aquele grupo e só depois alocar cada usuário em um grupo específico de privilégios

c) Qual a diferença entre segurança e integridade?

Segurança: Perda intencional da consistência do banco (invasão, roubo senha)

Integridade: Perda acidental da consistência (CRUD, privilégios)