



Universidade Eduardo Mondlane
Faculdade de Engenharia
Departamento de Engenharia Electrotécnica
Curso de Engenharia Informática

Base de Dados II

Ficha de Exercícios

Parte I

- ☐ Introdução ao processamento de transações (Definição e Propriedades)
- ☐ Conceito ACID de Transações
- ☐ Problemas de Concorrência em Transações

Parte II

- ☐ Serialização
- ☐ Bloqueios

Parte III

- ☐ Mapa de Acesso Lógico
- ☐ Formulário de Volume de Transações
- ☐ Matriz CRUD

Parte IV

- ☐ Desnormalização

Parte V

- ☐ Particionamentos

Msc Sérgio Mavie
Eng. Cristiliano Maculuve
Clementina Elihud

Parte I

1. O que é uma transação em um SGBD?
2. Explique os estados de uma transação
3. Discuta os tipos de falhas nas transação
4. O que significa a execução concorrente de transacoes em um sistema multiusuário?
5. Discuta as ações tomadas pelas operações read_item e write_item em um banco de dados.
6. Desenhe um diagrama de estados para um caso concreto de transação e discuta os estados típicos pelos quais a transação passa durante a execução.
7. Para que é usado o log do sistema? Quais são os tipos característicos de registros em um log do sistema? O que são pontos de confirmação da transação e por que eles são importantes?
8. Discuta os quatro níveis de isolamento em SQL
9. Discuta as propriedades do ACID.
10. Discuta os problemas das transações concorrentes.
11. Defina as violações causadas por cada um dos seguintes itens: Leitura suja, leitura não repetitiva
12. Que problema ocorre durante a execução das transações abaixo? O que pode ser feito para corrigir?

a)

| T1 | T2 |
|----------------|---------------|
| Read_item(X); | |
| X:= X+1; | |
| Write_item(X); | |
| | read_item(X); |
| | X:= X+2; |
| Read_item(Y) | ... |
| roolback | |

b)

| T1 | T2 |
|--------------|---------------|
| | Start Trans |
| Start Trans | Read(X) |
| Read(X) | $X = X + 100$ |
| $X = X - 10$ | Write(X) |
| Write(X) | Commint |
| Commint | |
| | |

c)

| T1 | T2 |
|--------------|-------------|
| Start trans | Start trans |
| Read(X) | Sum=0 |
| $X = X - 20$ | Read(X) |
| Write(X) | Sum=Sum+X |
| | Read(Y) |
| Read(Z) | |
| $Z = Z + 10$ | Sum=Sum+X |
| Write(Z) | |
| Commint | Read(Z) |
| | Sum = Sum+X |
| | Commint |

Parte II

1. O que é um Schedule serial?
2. O que é um Schedule serializavel?
3. Por que um Schedule serial é considerado correcto?
4. Explica como a serialização é usada para impor o controle de concorrência em um sistema de banco de dado.
5. Quantos schedules seriais existem para as três transações abaixo?

| Transação T_1 | Transação T_2 | Transação T_3 |
|-----------------|-----------------|-----------------|
| read_item(X); | read_item(Z); | read_item(Y); |
| write_item(X); | read_item(Y); | read_item(Z); |
| read_item(Y); | write_item(Y); | write_item(Y); |
| write_item(Y); | read_item(X); | write_item(Z); |
| | write_item(X); | |

6. Dadas as planificações:

- a. $r_1(X); r_3(X); w_1(X); r_2(X); w_3(X);$
- b. $r_1(X); r_3(X); w_3(X); w_1(X); r_2(X);$
- c. $r_3(X); r_2(X); w_3(X); r_1(X); w_1(X);$
- d. $r_3(X); r_2(X); r_1(X); w_3(X); w_1(X);$

- 6.1. Qual dos schedules é serializavel (por conflito)?
- 6.2. Qual dos schedules é serializavel (por Visão)?
- 6.3. Para cada Schedule serializável determine os schedules seriais equivalentes.

7. A planificação abaixo é Serializável? Justifique

| T1 | T2 |
|-------------|-------------|
| | Start Trans |
| Start Trans | Read(X) |
| Read(X) | X=X+100 |
| X=X-10 | Write(X) |
| Write(X) | Commint |
| Commint | |
| | |

8. Considere as transacções T1, T2 e T3 e os Schedules S1 e S2. Desenhe os Grafos de serialização dos Schedules S1 e S2. Se o Schedule for serializável determine o Schedule serial equivalente.

T1: r1(X), r1(Z), w1(X);

T2: r2(Z), r2(Y), w2(Z), w2(Y);

T3: r3(X), r3(Y), w3(Y);

S1: r1(X); r2(Z); r1(Z); r3(X); r3(Y); w1(X); w3(Y); r2(Y); w2(Z); w2(Y);

S2: r1(X), r2(Z); r3(X); r1(Z); r2(Y); r3(Y); w1(X); w2(Z); w3(Y); w2(Y);

9. Suponha duas transacções T1 e T2:

BEGIN TRANSACTION T1

UPDATE Banco

SET Saldo = ((SELECT Saldo FROM Banco WHERE Conta = 10) – 100)

WHERE Conta = 10

UPDATE Banco

SET Saldo = ((SELECT Saldo FROM Banco WHERE Conta = 20) + 100)

WHERE Conta = 20

COMMIT TRANSACTION T1

BEGIN TRANSACTION T2

UPDATE Banco

SET Saldo = ((SELECT Saldo FROM Banco WHERE Conta = 20) – 200)

WHERE Conta = 20

UPDATE Banco

SET Saldo = ((SELECT Saldo FROM Banco WHERE Conta = 30) + 200)

WHERE Conta = 30

COMMIT TRANSACTION T2

- 9.1. As instruções de leitura e escrita dos dados da transacção T1, obtidas a partir da SQL, são as seguintes:

READ Conta 10

WRITE Conta 10

READ Conta 20

WRITE Conta 20

- 9.2. Seguindo o exemplo acima, obtenha as instruções de leitura e escrita dos dados da transacção T2.

- 9.3. Determine as possíveis planificações de execução concorrente serializáveis em conflito e/ou em visão.

- 9.4. Proponha planificações de execução concorrente serializáveis usando os protocolos:

a) bloqueio em duas fases, com granularidade de bloqueio por tupla.

b) *timestamps*

c) validação

- 9.5. Verifique se existe a possibilidade de *deadlock* ao executar T1 e T2 de modo concorrente.
- 9.6. Suponha uma nova transação T3 concorrente com T1 e T2 e mostre que existe a possibilidade de *deadlock* ao usar o bloqueio em duas fases:

```
BEGIN TRANSACTION T3
UPDATE Banco
SET Saldo = ((SELECT Saldo FROM Banco WHERE Conta = 30) - 200)
WHERE Conta = 30
UPDATE Banco
SET Saldo = ((SELECT Saldo FROM Banco WHERE Conta = 10) + 200)
WHERE Conta = 10
COMMIT TRANSACTION T3
```

- 9.7. Recupere o *deadlock* verificados na questão anterior abortando a transação com o menor tempo de execução. Mostre uma possível planificação de execução a partir da recuperação do *deadlock*.
- 9.8. Proponha uma planificação de execução concorrente serializável para T1, T2 e T3 usando multi-versão.
- 9.9. Com que granularidade de bloqueio é possível executar concorrentemente T1, T2 e a transação T4?

```
BEGIN TRANSACTION T4
SELECT AVG(Saldo) FROM Banco
COMMIT TRANSACTION T4
```

10. Verifique a planificação ao lado:

- 10.1. É serializável em conflito
- 10.2. É recuperável
- 10.3. Evita *rollback* em cascata

| T5 | T6 | T7 |
|----------|----------|----------|
| Read(A) | | |
| | Read(A) | |
| Write(B) | | |
| | Write(A) | |
| | | Read(B) |
| | | Write(A) |

11. O que é o protocolo de bloqueio em duas fases?
12. Discuta os problemas de deadlock e inanição e as diferentes técnicas para lidar com esses problemas.
13. Compare os bloqueios binários com os bloqueios exclusivos/compartilhado. Por que esse último tipo de bloqueio é preferível?
14. Como as técnicas de controle de concorrência otimizadas diferem de outras técnicas de controle de concorrência?
15. Que tipo de bloqueio é necessário para operações de inserção e exclusão?
16. O que são bloqueios de intenção?
17. Prove que o protocolo básico de bloqueio em duas fases garante a serialização de conflito dos schedules
18. Discuta os seguintes tipos de bloqueios:
 - 18.1. Bloqueios Binários;
 - 18.2. Bloqueios compartilhados/Exclusivos(leitura/gravação);
 - 18.3. Bloqueio de Certificação;
19. Preenche (use **sim/não**) e explique as seguintes tabelas de compatibilidade entre bloqueios
 - 19.1.

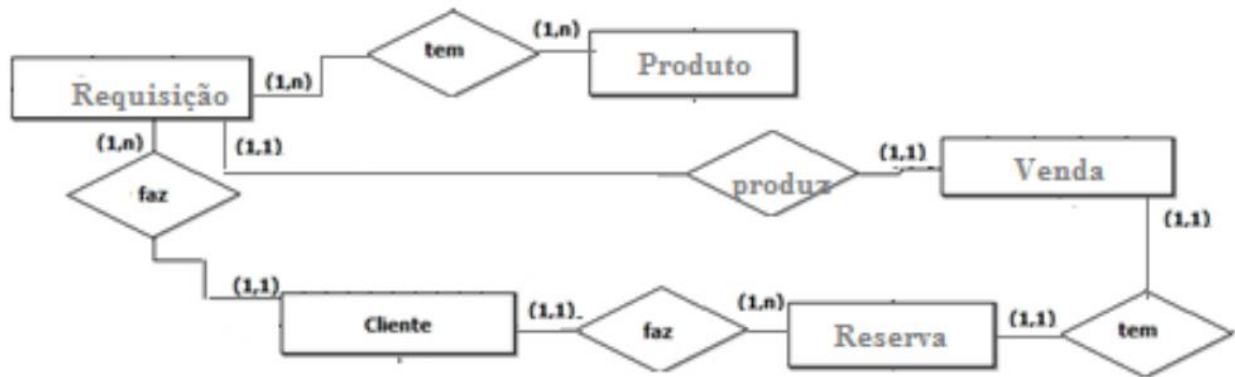
| Bloqueio | Compartilhado/leitura | Exclusivo/gravação |
|-----------------------|-----------------------|--------------------|
| Compartilhado/leitura | | |
| Exclusivo/gravação | | |

19.2.

| Bloqueio | Compartilhado/leitura | Exclusivo/gravação | Certificação |
|-----------------------|-----------------------|--------------------|--------------|
| Compartilhado/leitura | | | |
| Exclusivo/gravação | | | |
| Certificação | | | |

Parte III

1. Seja dado o seguinte Modelo Entidade Relacional sobre Um Sistema de facturação duma Boutique.



- a) Sabendo que:

- A boutique tem cerca de 1000 produtos diferentes.
- A boutique atende diariamente 5000 Clientes dos quais em Média 300 preferem fazer reservas sendo que um Cliente faz em Média Uma única reserva por dia.
- Diariamente a boutique recebe 6000 requisições de Clientes e em Média o cliente requisita 5 produtos diferentes em cada requisição.
- Diariamente consegue-se fazer 6200 vendas.

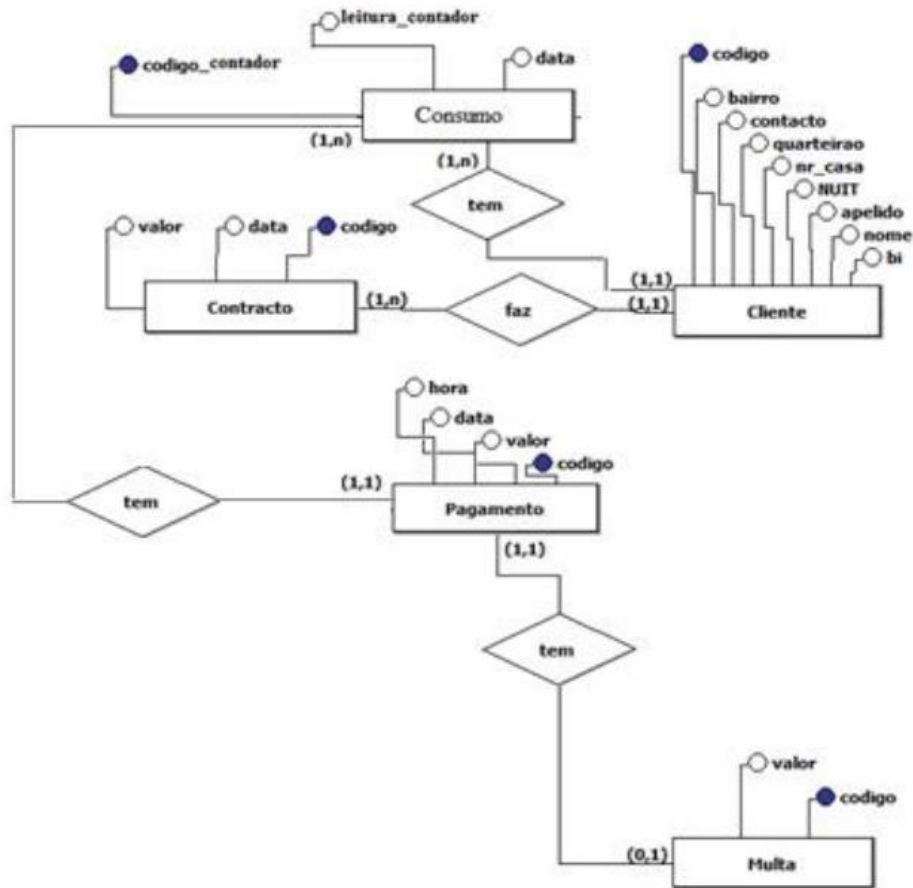
Faça o mapa de Acesso Lógico (MAL).

- b) Considerando o mesmo modelo. Suponha que tenha as seguintes operações e as suas respectivas frequências de ocorrência:

- Registrar uma reserva: 300 vezes/dia;
- Registrar um novo produto: 1 vez/mês
- Efectuar venda: 6200 vezes/dia;

Construa a tabela de Volume de Acessos Lógicos de Operações

2. Considere o seguinte Modelo Entidade Relacional para uma base de dados de pagamento de Agua duma determinada Distribuidora.



a) Do modelo acima sabe-se que:

- A distribuidora tem registados:
 - 1500 Clientes;
 - 3000 Contractos;
 - 50 000 Consumos;
 - 5.000 Multas;
 - 49.000 Pagamentos.
- Em média, um Cliente Abre dois Contractos;
- A probabilidade de haver uma Multa num pagamento é de 0.25%

Faça o mapa de Acesso Lógico (MAL). Interprete-o.

b) Ainda sobre o mesmo Modelo considere as Seguintes Operações e as suas respectivas Frequências de Ocorrência:

- Abrir um contracto para um Cliente: 20 vezes/mês
- Actualizar dados do Cliente. Frequência: 3 vezes/mês

- Efectuar Pagamento. Frequência: 1800 vezes/mês

Construa a tabela de Volume de Acessos Lógicos de Operações

Parte IV

1. Explique as razões para um analista realizar a desnormalização de um projeto de banco de dados.

Parte V

1. Explique e dê exemplos de uso dos seguintes tipos de índices:
 - a) HASH
 - b) HEAP
 - c) BTREE
 - d) ISAM