

U.C. 21103

Sistemas de Gestão de Bases de Dados

2020-2021

## Resolução e Critérios de Correção

### INSTRUÇÕES

- O e-fólio é constituído por 6 alíneas com cotação de 0,5 valores cada. A cotação global é de 3 valores.
- O e-fólio deve ser entregue num único ficheiro PDF, não zipado, com fundo branco, com perguntas numeradas e sem necessidade de rodar o texto para o ler. Penalização de 1 a 3 valores.
- Não são aceites e-fólios manuscritos, i.e. tem penalização de 100%.
- O nome do ficheiro deve seguir a normal “eFolioA” + <nº estudante> + <nome estudante com o máximo de 3 palavras>
- Durante a realização do e-fólio, os estudantes devem concentrar-se na resolução do seu trabalho individual, não sendo permitida a colocação de perguntas ao professor ou entre colegas.
- A interpretação das perguntas também faz parte da sua resolução, se encontrar alguma ambiguidade deve indicar claramente como foi resolvida.
- A legibilidade, a objetividade e a clareza nas respostas serão valorizadas, pelo que, a falta destas qualidades será penalizada.

Vetor Cotações:

1a 1b 2a 2b 3a 3b pergunta  
5 5 5 5 5 5 décimas

Critérios de correção gerais: todas as respostas devem ser justificadas, incluir imagens e exemplos com vista a clarificar os argumentos expostos.

1) Relativamente ao tema da Indexação, considere a seguinte tabela de uma base de dados: agências (idAgência -> nomeAgência, cidade, país).

a) Crie um SQL um índice para o nome da agência e um segundo índice com os atributos país e cidade.

**Resposta:**

```
CREATE INDEX nome_index ON agências (nomeAgência);  
CREATE INDEX cidade_index ON agências (país, cidade);
```

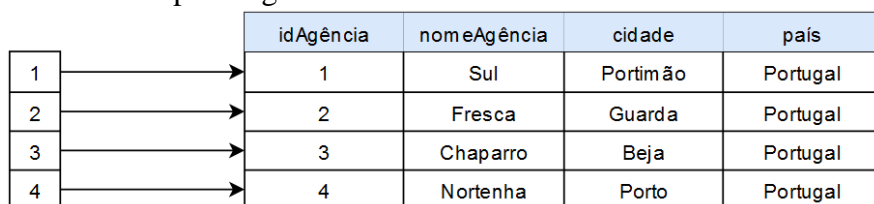
**Critérios de correção:**

- 2 décimas para o primeiro índice
- 3 décimas para o segundo índice
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%

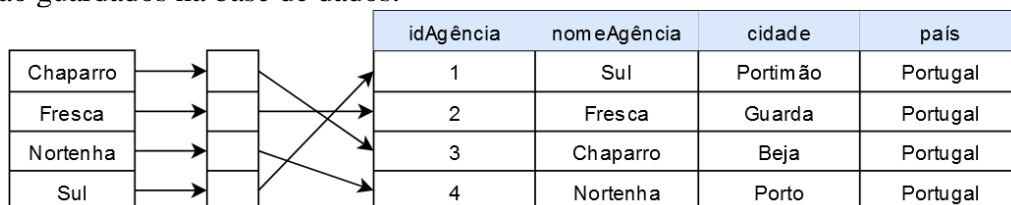
b) Quais diferenças entre índices agregados ('clustered') e não-agregados ('non-clustered'). Justifique e exemplifique com a tabela dada.

**Resposta:**

Índices agregados, também conhecidos como índices primários ou *clustered* são índices cujos atributos de busca estão ordenados sequencialmente nos dados. Exemplo do índice sobre o campo idAgência.



Índices não-agregados, também conhecidos como índices secundários ou *non-clustered* são índices cujos atributos de busca estão numa ordem diferente de como estão guardados na base de dados.



**Critérios de correção:**

- 2 décimas para o índice agregado
- 3 décimas para o índice não-agregado
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%

2) Processamento e Otimização Consultas. Dada a seguinte consulta SQL:  
 Alunos (alunoId -> nome, idade, endereço)  
 Livros (livroId -> título, autor)  
 Requisições (requisiçãoId -> alunoId, livroId, data)

```
SELECT A.nome
FROM Alunos A, Livros L, Requisições R
WHERE A.alunoID = R.alunoId
AND L.livroId = R.livroId
AND L.autor = 'Donald Knuth'
AND A.idade > 12
AND A.idade < 20
```

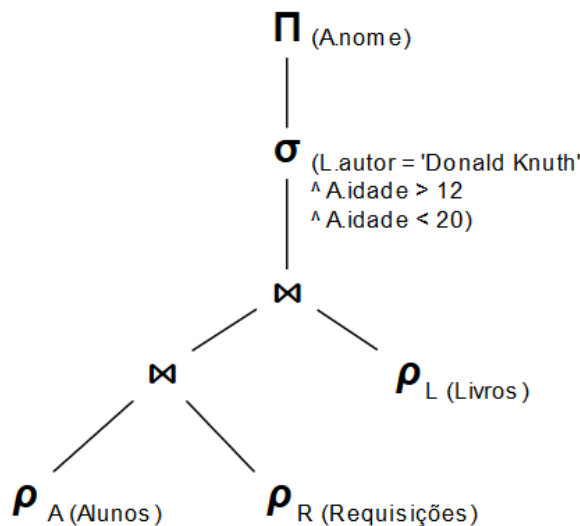
E assumindo:

Existem 10.000 registros de alunos armazenados em 1.000 páginas.  
 Existem 50.000 registros de livros armazenados em 5.000 páginas.  
 Existem 300.000 registros de requisições armazenadas em 15.000 páginas.  
 Existem 500 autores diferentes.  
 As idades dos alunos variam entre os 7 e os 24 anos.

a) Mostre a árvore do plano de consulta, assumindo que não há índices e os dados não estão classificados em nenhum atributo. Justifique e exemplifique a resposta.

**Resposta:**

A árvore do plano de consulta sem otimização será:



Critério de correção:

- 2 décimas, estrutura árvore
- 3 décimas, detalhes de implementação
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%

b) Calcule o custo deste plano de consulta (i) e a cardinalidade do resultado(ii). Justifique e exemplifique a resposta.

**Resposta:**

i) Junção de A com R:

Linhas de A,  $n_A = 10.000$

Linhas de R,  $n_R = 300.000$

Blocos de A,  $b_A = 1.000$  blocos

Blocos de R,  $b_R = 15.000$  blocos

Considerando a junção de blocos em ciclo ('block nested-loop join'), no pior caso, o número de blocos transferidos é  $b_A + b_A * b_R$  e o número de blocos procurados  $2.b_A$ .

Blocos transferidos =  $b_A + b_A * b_R =$

$$1.000 + 1.000 * 15.000 = 15.001.000$$

Blocos procurados =  $2.b_A = 2 * 1.000 = 2.000$

A junção seguinte na árvore procede-se de forma idêntica.

ii) Estimativa da cardinalidade do resultado:

livros: 1 autor 'Donald Knuth' em 500 autores,  $1/500$

idades: ]12..20[ em [7..24], corresponde a  $7/18$

número de requisições: 300.000

cardinalidade =  $300.000 * 1/500 * 7/18 = 233$  linhas ou tuplos

**Critério de correção:**

- 3 décimas, custo plano

- 2 décimas, cardinalidade

- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%

3) Relativo ao tema das Transações, considere o seguinte sequenciamento  $R1(x)$ ,  $W2(x)$ ,  $W1(x)$ ,  $R3(x)$ .

a) O que entende por sequenciamento serializável a conflitos? O que entende por sequenciamento serializável a vistas? Justifique e exemplifique a resposta.

**Resposta:**

i) Definição escalonamento seriável a conflitos

Serialização a conflitos: um escalonamento é seriável por conflitos se o seu grafo de precedência não contiver ciclos.

Na construção do grafo de precedências as arestas são montadas a partir das observações das transações que participa, da escala sendo duas transações  $T_i$  e  $T_j$  haverá uma aresta:  $T_i \rightarrow T_j$  se forem observadas as seguintes condições:

- $T_i$  executa *write*(q) antes de  $T_j$  executar *read*(q)
- $T_i$  executa *read*(q) antes de  $T_j$  executar *write*(q)
- $T_i$  executa *write*(q) antes de  $T_j$  executar *write*(q)

ii) Definição de escalonamento seriável a vistas.

Serialização por vistas: Diz-se que “ $T_i$  lê x de  $T_j$ ”, se  $W_j(x)$  for a última operação de escrita antes de  $R_i(x)$ . Dois escalamentos são equivalentes a vistas, se a relação “lê de” em ambas for a mesma.

Dois escalamentos S1 e S2 são equivalentes se:

- Todos os elementos A têm o mesmo valor inicial  
Se S1:  $r_i(A)$  lê o valor inicial, então S2:  $r_i(A)$  também lê o mesmo valor inicial
- Um elemento A tem a mesma vista nos dois escalamentos  
Se S1:  $w_j(A) \Rightarrow r_i(A)$  então em S2:  $w_j(A) \Rightarrow r_i(A)$ ; têm a mesma vista
- Todos os elementos A têm a mesma vista final  
Se S1:  $T_i$  termina  $w_i(A)$ , então S2:  $T_i$  também termina com  $w_i(A)$ ; os valores finais são os mesmos.

Ver exemplos na alínea seguinte.

**Critério de correção:**

- 2 décimas, escalonamento seriável a conflitos
- 3 décimas, escalonamento seriável a vistas
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%

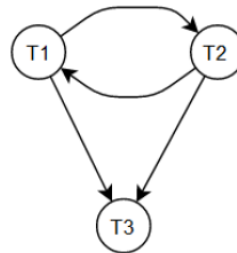
b) O referido sequenciamento é serializável a vistas ou a conflitos? Justifique e exemplifique a resposta.

**Resposta:**

i) Exemplo escalonamento seriável a conflitos:

Divisão por recurso/item: Item x: R1, W2, W1, R3	As precedências são as seguintes: R1 -> W2, W2 -> W1, W2 -> R3, W1 ->R3
---	--

T1	T2	T3
R(x)		
	W(x)	
W(x)		
		R(x)



Existem ciclos, pelo que o escalonamento não é seriável a conflitos.

ii) Exemplos de escalonamento seriável a vistas:

T1	T2	T3
R(x)		
	W(x)	
W(x)		
		R(x)

Sequenciamento original

T1	T2	T3
R(x)		
W(x)		
	W(x)	
		R(x)

Sequenciamento em série

Vamos verificar a primeira condição em que 'todos os elementos têm o mesmo valor inicial': R1(x) tem o mesmo valor inicial, já R3(x) não tem o mesmo valor inicial, no sequenciamento original segue a W1(x) e no sequenciamento em série a W2(x).

Assim, o escalonamento não é seriável a vistas.

Conclusão, o escalonamento/sequenciamento não é seriável nem a vistas nem a conflitos.

**Critério de correção:**

- 2 décimas, escalonamento seriável a conflitos
- 3 décimas, escalonamento seriável a vistas
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%