DI- FCT/UNL **21 de abril de 2016**

Sistemas de Bases de Dados

1º teste 2015/16

Duração: 2 horas (sem consulta)

Grupo 1

Considere parte duma base de dados para uma empresa de autocarros para uma rede europeia englobando 32 países (onde os atributos que constituem a chave primária estão sublinhados):

Locais({codLocal,nome,país,população,...}) Carreiras({número,localOrigem,localDestino})
Viagens({número,data,cMotorista,matrícula}) Motoristas(cMotorista,nome,sexo,dataNasc,...)

Para cada uma destas tabelas existe um índice secundário de árvore B+ sobre o(s) atributo(s) da chave primária, criado com a ordem das colunas indicada nas tabelas.

O SGBD adoptado usa blocos de 8KB (8192 bytes). Os registos de todas as tabelas têm dimensão variável, sendo que, em média, um registo da tabela de locais ocupa 0,5KB, motoristas ocupa 1KB, um registo de carreiras e de viagens ocupa 128 bytes. Num dado momento a tabela de locais tem 3.200 tuplos, a de carreiras 12.000 tuplos, a de viagens 3.000.000 tuplos e a de motoristas 1.000 tuplos. Um nó da árvore B^+ pode conter cerca de 100 chaves de pesquisa e sabe-se que o tempo de um seek é dez vezes superior ao da transferência de um bloco ($t_S = 10 * t_T$), e a memória comporta apenas 100 blocos.

Nota: Neste grupo, sempre que se solicitarem exemplos, estes devem ser **exclusivamente** sobre esta base de dados. Além disso, **todas** as respostas deverão conter uma **breve justificação**.

1 a) Apresente dois planos para execução da seguinte pergunta SQL, justificando brevemente qual deles deverá ter um menor custo na base de dados em causa.

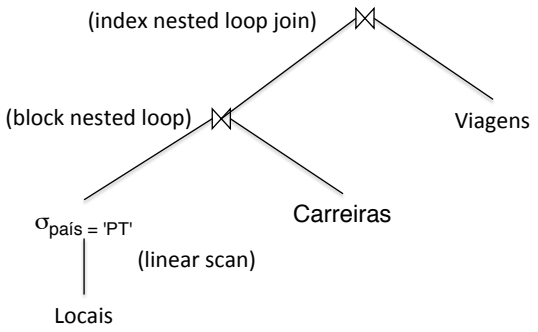
```
SELECT número, nome FROM  Motoristas NATURAL INNER JOIN Viagens
WHERE sexo = 'F'
ORDER BY nome
```

1 b) Suponha que o itinerário de cada carreira passa também a ser armazenado numa coluna dessa tabela, passando um tuplo a ocupar em média 10Kb. Poder-se-ia usar o mecanismo de *"slotted pages"* para armazenar os tuplos dessa tabela? Porquê?

1 c) Indique se na sua opinião a pergunta SQL abaixo pode ser eficientemente respondida pela base de dados, recorrendo aos índices existentes.

```
SELECT * FROM  Carreiras NATURAL INNER JOIN Viagens
WHERE data BETWEEN '2016-04-18' AND '2016-04-21'
```

1 d) Considere o plano de avaliação apresentado na figura ao lado. Estime o menor custo de execução desse plano, indicando explicitamente qual a relação "inner" e "outer" para cada operação de junção.



1 e) Indique que índice(s) criaria para tornar a execução da seguinte pergunta mais eficiente:

```
SELECT * FROM Carreiras c, Locais l1, Locais l2
WHERE c.localOrigem = l1.codLocal AND c.localDestinto = l2.codLocal;
```

Grupo 2

Nota: A resposta a cada uma das alíneas deste grupo **não pode em caso algum exceder uma página**.

2 a) *"A forma como se faz a alocação de registos da base de dados a blocos de disco afecta de forma significativa a performance dum sistema de bases de dados".*

Justifique porque razões tal se passa.

2 b) O algoritmo de *hash-join* tem duas fases distintas. Explique em que consiste cada uma delas, a sua função, limitações e se é sempre necessário realizá-las.

2 c) O algoritmo de *block nested loop-join*, com o pseudo-código apresentado abaixo, tem uma complexidade de pior caso $b_r * b_s + b_r$ transferências de blocos + 2 * b_r seeks, e no melhor caso $b_r + b_s$ transferências de blocos + 2 seeks. Indique como se derivam estas fórmulas não esquecendo de destrinçar as situações de melhor e pior casos.

```
for each block B_r of r do begin
    for each block B_s of s do begin
        for each tuple t_r in B_r do begin
            for each tuple t_s in B_s do begin
                Check if (t_r,t_s) satisfy the join condition
                if they do, add t_r • t_s to the result.
            end
        end
    end
end
```