

Sistemas de Bases de Dados
2º teste 2014/15
Duração: 1 hora + 30 minutos (sem consulta)

Grupo 1

Considere parte de uma base de dados registando informação sobre contribuintes identificados por um NIF, podendo ser individuais ou colectivos (empresas), empresas e seus acionistas e respectivo capital social (montante em dinheiro), e atividade económica realizada pela empresa classificada de acordo com códigos de atividade (CAEs) tendo obrigatoriamente um código primário e no máximo 3 secundários (não primários). Essa base de dados inclui as seguintes tabelas, não sendo permitidos valores nulos em nenhum atributo (onde se sublinham os atributos que constituem a chave primária):

contribuintes({ <u>NIF</u> ,Rua,Local,CodPostal,...})	peessoas({ <u>NIF</u> Pess,Nome,Sexo,DataNasc,...})
empresas({ <u>NIF</u> Empr,Nome,Tipo,DataInício,...})	acionistas({ <u>NIF</u> Empr, <u>NIF</u> Contribuinte,Capital})
atividades({ <u>NIF</u> Empr, <u>CAE</u> ,Primário})	codigosAtividade({ <u>CAE</u> ,Designação,...})

Todas as chaves são formadas por atributos inteiros. O atributo Tipo da tabela empresas tem 4 valores possíveis e o atributo Primário da tabela atividades é booleano. Para cada uma das tabelas existe um índice **clustered** de árvore B+ sobre o(s) atributo(s) da chave primária. Além disso são definidas na base de dados as seguintes *chaves estrangeiras*: de NIFPess em pessoas e NIFEmpr em empresas para contribuintes(NIF), de NIFEmpr em acionistas e atividades para empresas, NIFContribuinte para contribuintes(NIF), e CAE em atividades para codigosAtividade. Os CAE encontram-se numerados sequencialmente a partir de 1.

Tendo em conta o sistema de gestão de base de dados usado, tipicamente cabem num bloco 20 tuplos da tabela contribuintes, pessoas e empresas, 40 tuplos da tabela codigosAtividade e 100 tuplos das tabelas atividades e acionistas. Sabemos ainda que num dado momento existem 500.000 contribuintes, dos quais 100.000 empresas e 400.000 pessoas, a tabela acionistas tem 250.000 tuplos e a de atividades 350.000. A tabela de codigosAtividade tem 4000 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 ($t_s = 10 * t_f$). A memória RAM tem espaço para albergar 100 blocos.

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

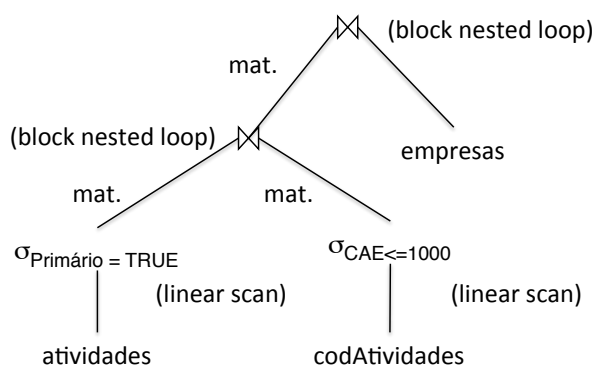
1 a) Estime o número de tuplos resultantes da seguinte consulta SQL e respectiva seletividade da condição nas duas situações: i) sabendo que toda a empresa tem um código de atividade primário obrigatório; ii) não sabendo a informação em i).

```
SELECT * FROM atividades WHERE NOT (Primário = TRUE)
```

1 b) Apresente o plano de execução mais eficiente para a seguinte consulta SQL que determina os números de contribuintes das empresas detidas por pessoas nascidas a 1 de Janeiro. Utilize o algoritmo de junção *nested-loop join* quando for necessário efetuar junções de tabelas.

```
SELECT DISTINCT NIFEmpr
FROM pessoas INNER JOIN contribuintes ON (NIF = NIFPess)
INNER JOIN acionistas ON (NIF = NIFContr)
WHERE MONTH(DataNasc) = 1 AND DAY(DataNasc) = 1
```

1 c) Considere o plano de execução detalhado na figura ao lado. Analise se compensa utilizar o algoritmo *indexed nested-loop join* para implementar a junção presente na raiz do plano, explicitando qual a relação externa e qual a relação interna.



1 d) Discuta se a técnica de minimização de junções pode ser utilizada para otimizar a consulta SQL:

```
SELECT * FROM empresas E
WHERE EXISTS (SELECT * FROM acionistas A WHERE A.NIFContr = E.NIFEmpr)
```

1 e) Apresente uma consulta SQL com a junção de duas tabelas para a qual não poderá ser utilizado o algoritmo de junção *merge-join* estudado.

Grupo 2

Nota: Dê respostas breves mas justifique adequadamente.

2 a) O algoritmo de *block nested-loop join* pode ser otimizado quando se dispõe de mais memória, utilizando-a para leituras de múltiplos blocos da relação externa. Assumindo que a dimensão da memória é de M blocos de memória e que são utilizados $M - 2$ blocos de disco para leituras da relação externa (r), derive ou explique a seguinte fórmula de custo do pior caso para a otimização indicada:

$$\text{Custo} = \lceil b_r / (M-2) \rceil * b_s + b_r \text{ block transfers} + 2 \lceil b_r / (M-2) \rceil \text{ seeks}$$

2 b) Determine quantos processadores serão necessários para executar a junção com o algoritmo *fragment-and-replicate join* de duas relações r e s , com 1 milhão e 2 milhões de tuplos, respectivamente, sabendo que cada processador pode armazenar simultaneamente 100.000 tuplos de r e 100.000 tuplos de s .

2 c) Explique o motivo da utilização de duas funções de dispersão distintas no algoritmo de *hash-join*.