

DI- FCT/UNL

6 de maio de 2017

Sistemas de Bases de Dados

1º teste 2016/17

Duração: 2 horas (sem consulta)

Grupo 1 (14 valores)

Considere parte duma base de dados para uma empresa de TV por cabo a operar num único país (onde os atributos que constituem a chave primária estão sublinhados):

Clientes({ <u>NIF</u> ,nome,morada,idade,sexo,cPacote,...})	Grelhas({cPacote, <u>canal</u> ,posição,preço_canal})
Pacotes({cPacote, <u>descrição</u> ,preço_base})	Canais({canal, <u>nome_canal</u> ,logo,categoria,...})
	Programas({canal, <u>datahora</u> ,nome,tipo,...})

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. Os atributos {cPacote,posição} formam uma chave candidata para a tabela Grelhas, existindo apenas 250 valores distintos para posição.

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 Clientes, Pacotes e Canais ocupa 1KB, um registo de Programas ocupa 512 bytes, e um registo da Grelha ocupa 128 bytes. Num dado momento a tabela de clientes tem 800.000 tuplos, a de pacotes 4 tuplos, a de grelhas 700 tuplos, a de canais 200 tuplos e a de programas 1.825.000 tuplos (correspondendo aos dados de programas de 2016). 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 sabendo que há clientes com idade entre os 18 e 98 anos (inclusive).

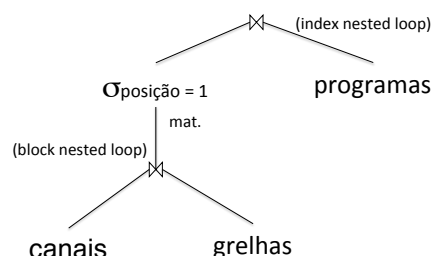
**SELECT nome, preço FROM Clientes NATURAL INNER JOIN Pacotes
WHERE idade <= 28;**

- 1 b) Apresente uma consulta SQL que retire vantagens de uma organização em ficheiro de agrupamento múltiplo de tabelas para canais e programas (*multitable clustering file organization*).

- 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 Programas NATURAL INNER JOIN Canais
WHERE canal = 'RTP1' AND datahora >= '2016-12-30 00:00:00' ;**

- 1 d) Considere o plano de avaliação apresentado na figura ao lado. Estime o 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 tipo, COUNT(*) FROM programas
WHERE tipo IN ('Drama','Comédia')
GROUP BY tipo;**

Grupo 2 (6 valores)

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

- 2 a)** Explique porque não podem existir índices secundários esparsos.
- 2 b)** O algoritmo de ordenação *external sort-merge* 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 *index nested loop-join* tem complexidade no pior caso de $b_r * (t_r + t_s) + n_r * c$ em que c é o custo de percorrer o índice e obter os tuplos correspondentes, assumindo que só se tem um bloco de memória para armazenar tuplos de r . Indique como se deriva a fórmula anterior e como melhoraria a complexidade do algoritmo se existissem mais blocos de memória para armazenar tuplos de r .

```
for each block  $B_r$  of  $r$  do begin
  for each tuple  $t_r$  in  $B_r$  do begin
    use index to retrieve each tuple  $t_s$  that satisfies the condition
    add  $t_r \bullet t_s$  to the result.
  end
end
end
```