

## Sistemas de Bases de Dados

1.º teste (com consulta limitada: 2 folhas identificadas) - Duração: 2 horas

N.º:		Nome:	
------	--	-------	--

### Grupo 1 (7 valores)

**1 a)** Para cada uma das seguintes 5 frases, indique a opção correta. No caso de simples afirmações, use **V** (de Verdadeira) ou **F** (de Falsa).

**Nota:** Nesta alínea cada resposta certa vale 0,3 valores, e cada resposta errada vale -0,3 valores (negativo, portanto). Se não sabe a resposta, não responda.

A memória cache é mais rápida e mais cara (por KByte) do que a memória flash.

Para medição de tempos relativos a HDD costuma-se usar  $t_s$  (*seek*) e  $t_T$  (*transfer*), sendo  $t_s > t_T$ .

Quando possível, procura-se que um ficheiro numa relação esteja fragmentado ao longo das várias pistas dum disco, para que a cabeça de leitura possa estar sempre próxima dum seu bloco.

Num *nested loop join* é melhor usar que política de substituição de blocos do *buffer*? *Least (L)* ou *Most (M) Recently Used (RU)* ? (LRU/MRU)

Em organizações RAID usa-se *block-striping* para “limpar” os blocos numa relação (removendo o que não lhe diz respeito), para uma melhor compactação de informação, de modo a reduzir o espaço necessário.

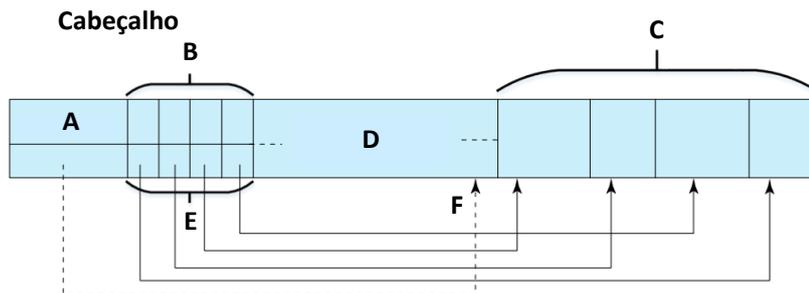
**1 b)** Numa organização de ficheiro numa relação como uma sequência de  $n$  registos de comprimento constante, indique 3 opções de reorganização do ficheiro por remoção do registo número  $i$ .

1	
2	
3	

1 c) Descreva a forma de representação habitual para um registo de tamanho variável (por incluir atributos de tamanho variável, e por poder ter atributos NULL).

1 d) I. Para que serve a estrutura de Slotted Pages?

II. Legende devidamente, para as letras A a F, as respetivas porções da seguinte ilustração duma tal *slotted page*:



A	
B	
C	

D	
E	
F	

N.º:	
------	--

**1 e)** Mostre um exemplo de organização de ficheiro em *clustering* multitabela, ilustrando com um caso concreto de 2 tabelas (com 2 a 6 tuplos cada).

--

**1 f)** Dê 3 exemplos de metadados (relativos a uma base de dados) presentes no dicionário do sistema.

1	
2	
3	

**Grupo 2** (8 valores)

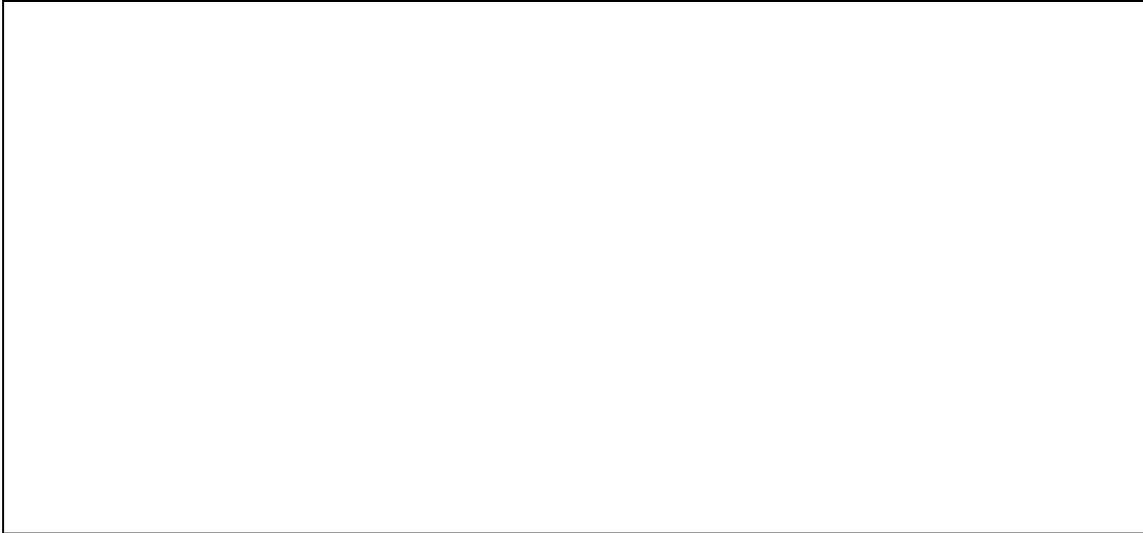
**2 a)** Qual a diferença entre um índice primário e um índice secundário?

**2 b)** Genericamente, indique 4 estratégias de pesquisa para a seguinte consulta, sabendo que não há índices sobre chaves de pesquisa compostas (i.e. com mais do que um atributo):

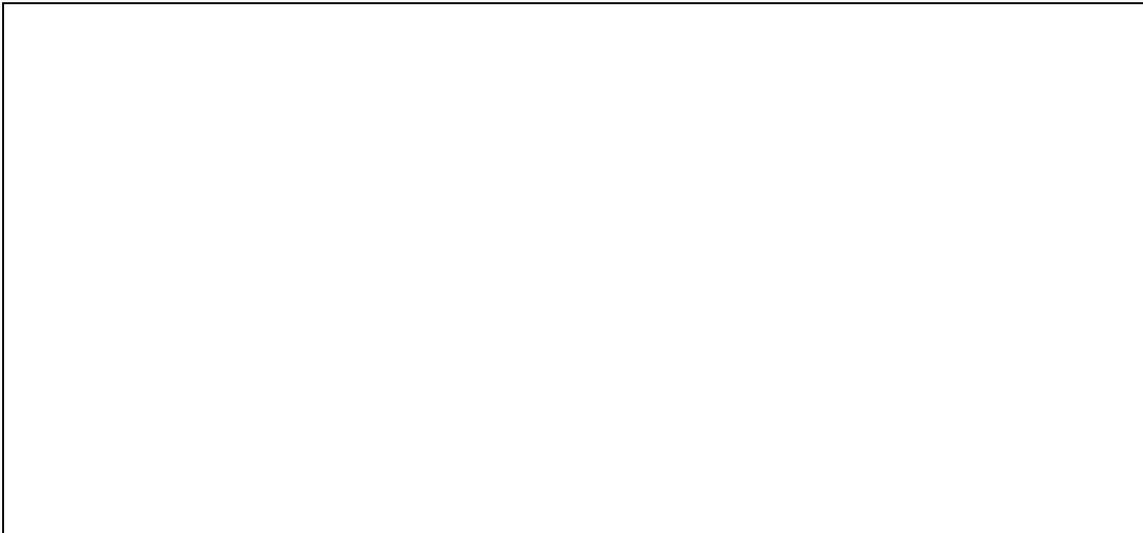
```
Select nome, taças+campeonatos as trofeus from clubes  
where ano_fundacao > 1950 and cidade = 'Lisboa';
```

N.º:	
------	--

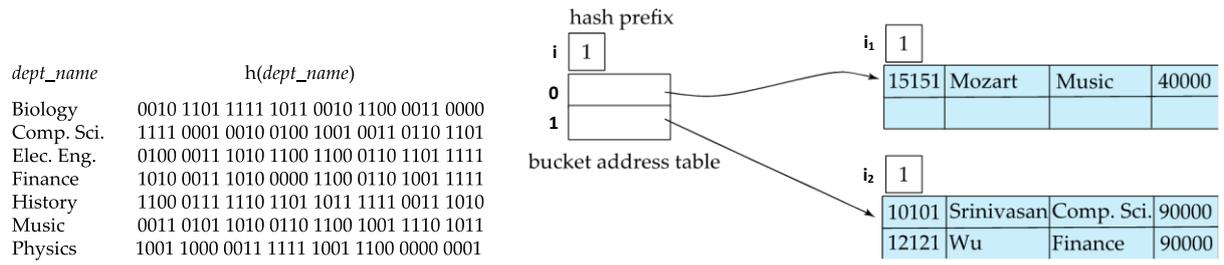
- 2 c) I. Desenhe a árvore B+ (com nós onde cabem 4 apontadores) resultante da inserção dos valores chave de pesquisa (2,3,5,7,11,17) por esta ordem, estando inicialmente vazia.



- II. A árvore resultante da inserção anterior seria a mesma independentemente da ordem de entrada desses valores? Se sim, explique porquê; se não, mostre um exemplo duma outra ordem de inserção e a respetiva árvore B+ diferente resultante.



**2 d)** Num índice de hash dinâmico com a função de hash, e a estrutura de hash extensível mostradas em baixo (onde cada *bucket* suporta 2 tuplos), desenhe a estrutura resultante após a inserção do tuplo (22222,Einstein,Physics,95000).



N.º:	
------	--

**Grupo 3** (5 valores)

- 3 a)** No algoritmo de *sort-merge* aplicado a uma relação com 111 tuplos e *blocking factor*  $f=2$ , quantos *runs* são inicialmente criados e quantos passos de junção (*merge steps*) são necessários para obter o resultado ordenado, se a capacidade da memória for  $M=4$  blocos? (assuma que *buffer blocks*  $b_b=1$ )

- 3 b)** Recorde o algoritmo de *block nested-loop join* para junção das relações  $r$  e  $s$  (verificação para cada par de tuplos, com leituras bloco a bloco):

```

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  $t_r \bullet t_s$  to the result.
  end end end end.

```

Indique duas possíveis melhorias a este algoritmo.

1	
2	

- 3 c) Considere a tabela *temperatura*(*id*, *cidade*, *ano*, *mês*, *dia*, *hora*, *valor*) criada em SQL da seguinte forma:

```
create table temperatura(  
    id INTEGER NOT NULL primary key,  
    Cidade INTEGER NOT NULL references cidades(id),  
    Ano INTEGER NOT NULL,  
    Mes INTEGER NOT NULL check(mes between 1 and 12),  
    Dia INTEGER NOT NULL check(dia between 1 and 31),  
    Hora INTEGER NOT NULL check(hora between 0 and 23),  
    Valor INTEGER NOT NULL,  
    unique(cidade,ano,mes,dia,hora));
```

Foram registados um milhão de valores de temperatura em 100 cidades europeias desde o ano 1999 (há várias medições em falta, no espaço e no tempo, por diferentes motivos). Considere que neste sistema um inteiro ocupa 10 bytes e que são usados blocos de 10000 bytes.

I. Quantos blocos ocupa a tabela *temperatura*?

II. Quantos tuplos se esperam da consulta

'select \* from temperatura where cidade=2 and mes in (12,1) and hora > 8;'

III. Quantos tuplos se esperam da consulta

'select \* from temperatura where cidade=20 or ano=2000;'

IV. Quantos tuplos se esperam da consulta

'with top as (select cidade, ano, max(valor) as maxt from temperatura group by cidade, ano)

(select cidade, ano, mes, maxt-valor

from temperatura natural inner join top);' E quantos blocos em disco ocuparia?

$n$  (tuplos) = ?

$b$  (blocos) = ?