

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 Recently Used* (RU) ? (LRU/MRU)

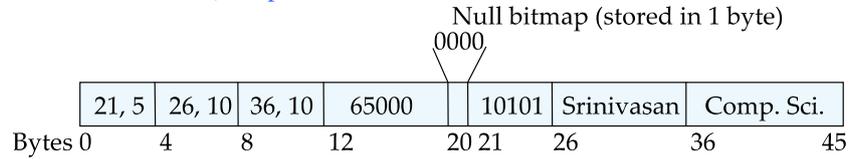
Em organizações RAID usa-se *block-striping* para “limpar” os blocos numa relação (removendo o que não lhe diz respeito), para uma melhor compactação de informação, de modo a reduzir o espaço necessário.

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	Deslocar todos os registos $i+1 \dots n$ para a posição imediatamente anterior.
2	Passar o registo n para a posição i .
3	Não mover registos, mas manter uma lista de posições livres.

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).

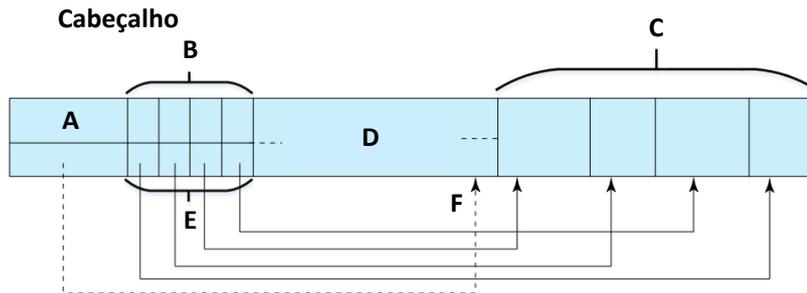
Estrutura com uma primeira parte de tamanho fixo com todos os atributos armazenados por ordem, sendo que os de tamanho variável são representados por um par (de tamanho fixo) (distância, comprimento) indicando (em bytes) onde (à frente) pode ser lido o valor atual. Os campos a null são representados por um bitmap (normalmente colocado entre a informação de todos os atributos (primeira parte de tamanho fixo) e a parte dos valores dos atributos de tamanho variável. E.g.



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

Uma slotted page (normalmente do tamanho de um bloco) é uma estrutura usada para armazenar registos de tamanho variável

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



A	N.º de registos
B	Tamanhos dos registos
C	Registos

D	Espaço livre
E	Apontadores para os registos
F	Fim do espaço livre

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).

E.g. Tabelas Países(País, Língua) com tuplos {(Portugal,Português),(Brasil,Português)} e Cidades(Cidade, País) com tuplos {(Lisboa,Portugal), (Porto,Portugal), (Brasília,Brasil), (São Paulo,Brasil), (Rio de Janeiro,Brasil)}, com *clustering* pelo atributo País:

Portugal	Português
Lisboa	
Porto	
Brasil	Português
Brasília	
São Paulo	
Rio de Janeiro	

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

1	Nomes das tabelas e suas características (e.g. número de atributos, localização, ...)
2	Nomes dos atributos nas tabelas, e suas características (e.g. tipo, comprimento...)
3	Nomes dos índices sobre tabelas e suas características (e.g. tipo, atributos, ...)

(Adicionalmente também se podia indicar dados de vistas, de utilizadores, ...)

Grupo 2 (8 valores)

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

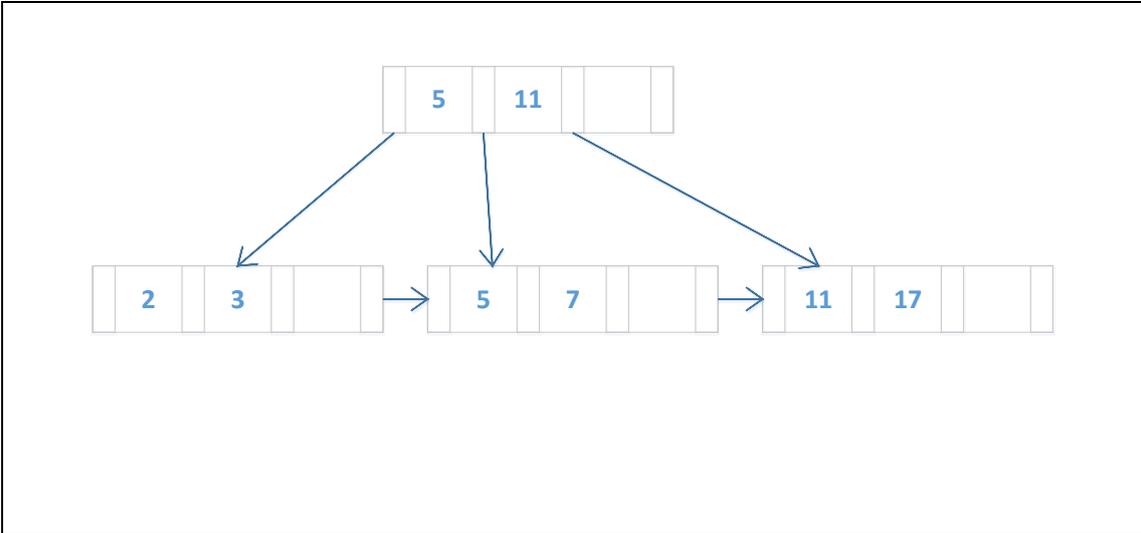
Num índice primário a chave de pesquisa corresponde à ordenação sequencial do ficheiro de dados, ao contrário do índice secundário (onde o ficheiro não está sequencialmente ordenado pela chave de pesquisa do índice).

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';
```

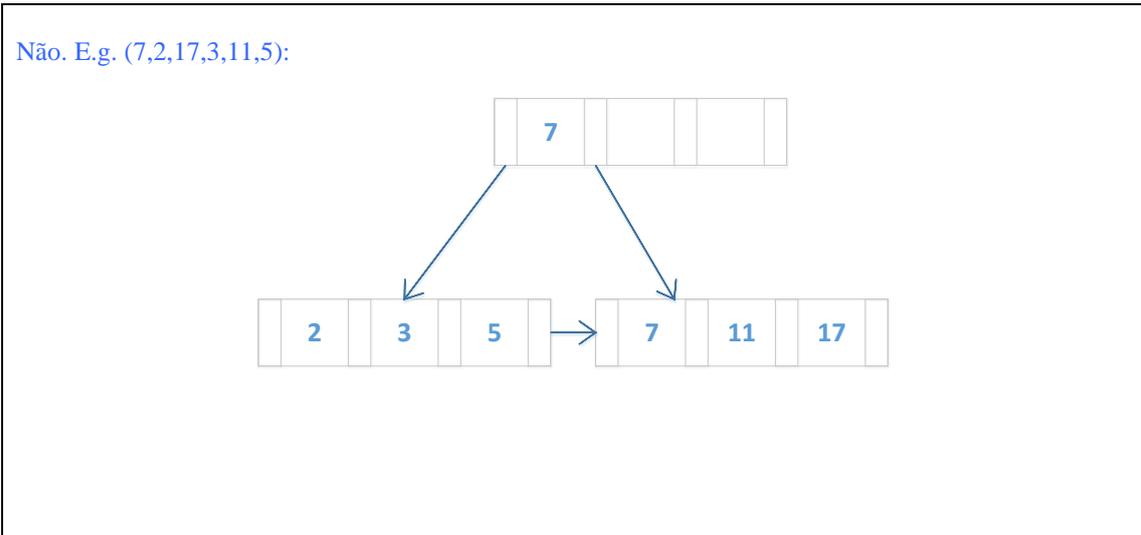
- 1- usar uma pesquisa linear (*linear scan*) no ficheiro, testando a condição conjunta (a conjunção na cláusula '*where*') para todos os tuplos;
- 2- usar índice sobre *ano_fundacao* para apanhar todos os tuplos onde *ano_fundacao*>1950 e testar se *cidade*='Lisboa'
- 3- usar índice sobre *cidade* para apanhar todos os tuplos relativos a Lisboa e testar se *ano_fundacao*>1950;
- 4- usar ambos os índices (sobre *ano_fundacao* e sobre *cidade*) para obter os apontadores dos tuplos que satisfazem a respetiva condição. Fazer então a interseção dos 2 conjuntos obtidos e seguir os apontadores para obter os tuplos desejados.

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.

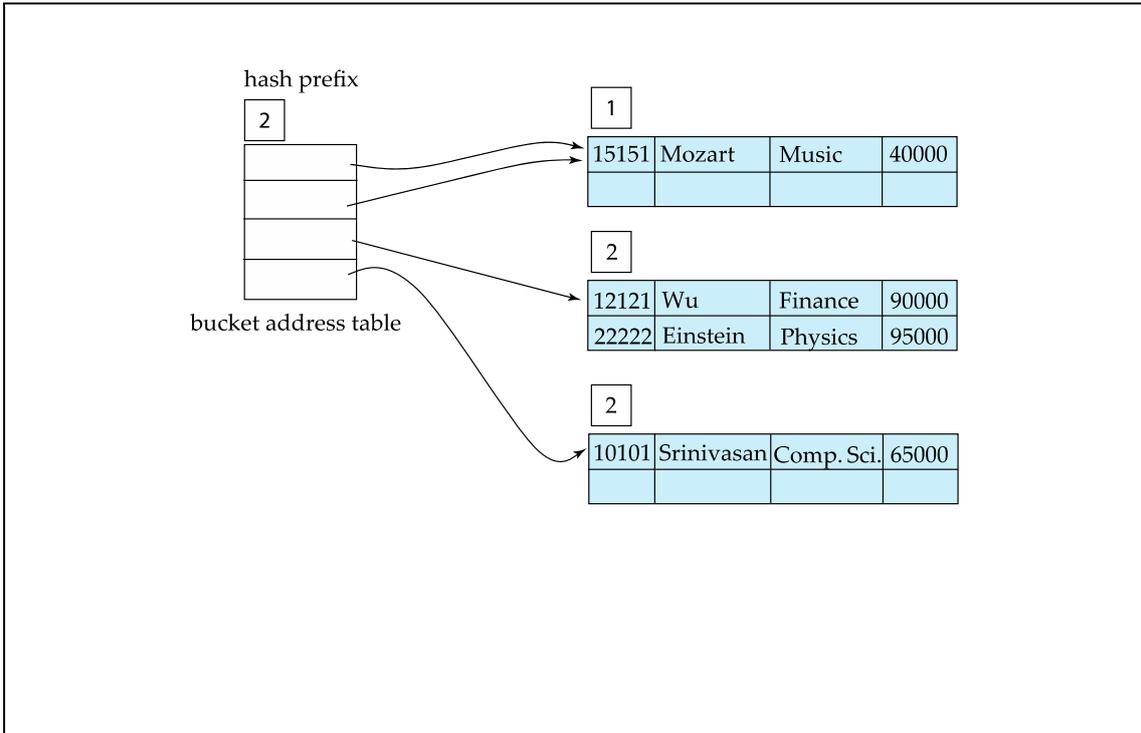
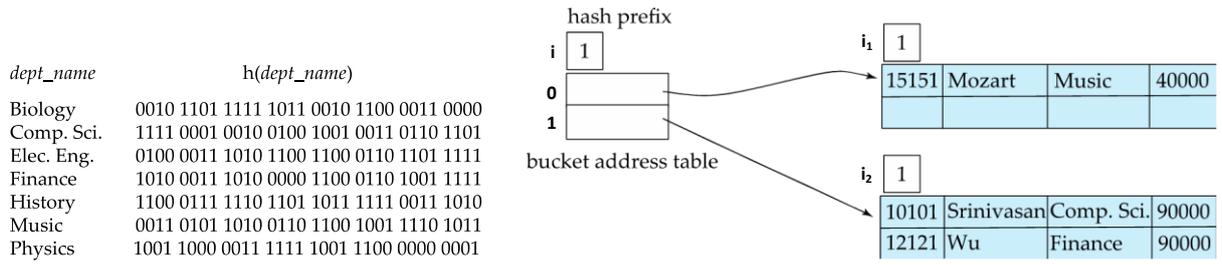


II. A árvore resultante da inserção anterior seria a mesma independentemente da ordem de entrada desses valores? Se sim, explique porquê; se não, mostre um exemplo duma outra ordem de inserção e a respetiva árvore B+ diferente resultante.

Não. E.g. (7,2,17,3,11,5):



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* $b_b=1$)

Como apenas cabem 2 tuplos num bloco ($f=2$), a relação ocupa $b=56$ blocos ($\lceil 111/2 \rceil$).
 Com $M=4$ teremos $56/4=14$ runs.
 O número de *merge steps* é dado por $\lceil \log_{M-1}(b/M) \rceil = \lceil \log_3(14) \rceil = 3$

- 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 ler $M-2$ blocos (em vez de um bloco) de cada vez na relação exterior (M é o tamanho da memória em blocos; os restantes 2 blocos são os buffers para a relação interior e para a saída/output).
- 2 Percorrer o ciclo (de blocos) interior alternadamente para a frente e para trás para aproveitar blocos ainda no buffer (assumindo política LRU).

(Poderiam ainda ser indicados: a) o uso dum eventual índice na relação interior para obter mais rapidamente os tuplos que fazem junção com o tuplo corrente; e b) se o(s) atributo(s) de junção corresponderem a uma chave da relação interior, pode-se parar o ciclo interior na primeira correspondência)

- 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*?

$$b_{temperatura} = 7 \cdot 10^6 \cdot 1000000 / 10000 = 7000$$

II. Quantos tuplos se esperam da consulta

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

$$n = 1000000 / (100 \cdot 6 \cdot 24 / 15) = 10000 \cdot 15 / (6 \cdot 24) = 10000 \cdot 5 / (2 \cdot 24) = 25000 / 24 = 1000 + 1000 / 24 (\approx 1042)$$

III. Quantos tuplos se esperam da consulta

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

$$n = 1000000 \cdot (1 - (1 - 1/100) \cdot (1 - 1/20)) = 1000000 \cdot (1 - (99/100) \cdot (19/20)) = 1000000 \cdot (1 - 1881/2000) = 1000000 \cdot (119/2000) = 1190000 / 2 = 595000$$

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?

$$n \text{ (tuplos)} = ?$$

$$n = 1000000$$

$$b \text{ (blocos)} = ?$$

$$b = 4 \cdot 10^6 \cdot 1000000 / 10000 = 4000$$