21103 - Sistemas de Gestão de Bases de Dados
2014-2015
e-fólio A

Resolução e Critérios de Correção

PARA A RESOLUÇÃO DO E-FÓLIO, ACONSELHA-SE QUE LEIA ATENTAMENTE O SEGUINTE:

1) O e-fólio é constituído por 4 perguntas. A cotação global é de 2 valores.

2) O e-fólio deve ser entregue num único ficheiro PDF, não zipado, com fundo branco, com perguntas numeradas e sem necessidade de rodar o texto para o ler. Penalização de 1 a 2 valores.

3) Não são aceites e-fólios manuscritos, i.e. tem penalização de 100%.

4) O nome do ficheiro deve seguir a normal "eFolioA" + <nº estudante> + <nome estudante com o máximo de 3 palavras>

5) Durante a realização do e-fólio, os estudantes devem concentrar-se na resolução do seu trabalho individual, não sendo permitida a colocação de perguntas ao professor ou entre colegas.

6) A interpretação das perguntas também faz parte da sua resolução, se encontrar alguma ambiguidade deve indicar claramente como foi resolvida.

7**)** A legibilidade, a objectividade e a clareza nas respostas serão valorizadas, pelo que, a falta destas qualidades serão penalizadas.

Vetor Cotações
1 2 3 4 pergunta
5 5 5 5 décimas

1) Relativo ao Cap.10 - Armazenamento e Estrutura dos Ficheiros

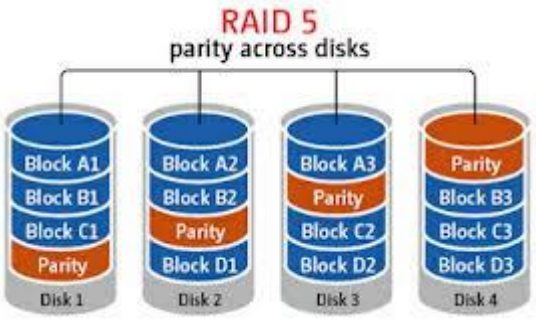
Uma falha de energia ocorreu quando um bloco de disco está a ser escrito, resultando num bloco parcialmente escrito. Suponha que blocos parcialmente escritos podem ser detetados. A propriedade de atomicidade dos blocos é dada quando o bloco do disco é totalmente escrito (ou seja, não são permitidas gravações parciais).

Sugira um procedimento para conseguir o efeito de atomicidade dos blocos com um disco RAID nível 5. O procedimento deve envolver o trabalho no caso de recuperação da falha. Exemplifique o procedimento convenientemente.

Resposta:

Para o RAID 5 teremos a seguinte classificação segundo os critérios de desempenho e redundância:

	desempenho	**redundância**	
RAID	**"striping"**	**"mirroring"**	**paridade**
5	nível bloco	---	distribuída



No RAID 5 para garantir a atomicidade a escrita do bloco tem os seguintes passos:
i. escrita da informação no 1º bloco
ii. escrita da informação do bloco de paridade

No momento da recuperação, consistindo cada conjunto de bloco de ordem n de cada um dos discos é considerado. Se nenhum dos blocos do conjunto ter sido parcialmente escrito, e o conteúdo do bloco de paridade são consistentes com o conteúdo dos blocos de informação, em seguida, mais nenhuma ação devem ser tomadas.

A paridade do RAID 5 utiliza a função XOR, se um disco falha, os dados dos outros dois podem ser combinados e reconstruída a informação em falta. Para 4 discos, com os dados D e a paridade P, teremos:

Drive 1	Drive 2	Drive 3	Drive 4
D0	D1	D2	P0
D3	D4	P1	D5
D6	P2	D7	D8
P3	D9	D10	D11

P0 = D0 XOR D1 XOR D2
P1 = D3 XOR D4 XOR D5
P2 = D6 XOR D7 XOR D8
P3 = D9 XOR D10 XOR D11

No caso de falha da Drive 2, por exemplo, a informação pode ser reconstruída através do operador XOR.

A Drive 4 é utilizada para paridade	No caso de falha da Drive 2, esta pode ser reconstruída
0110 1101 Drive 1 XOR 1101 0100 Drive 2 XOR 0000 0000 Drive 3 1011 1001 Drive 4 Paridade P0	0110 1101 Drive 1 XOR 0000 0000 Drive 3 XOR 1011 1001 Drive 4 1101 0100 Drive 2 Recuperada

Visto que os blocos parcialmente escritos podem ser detetados, no caso de escrita na Drive 2 e falha de atualização na Paridade D0, teremos:

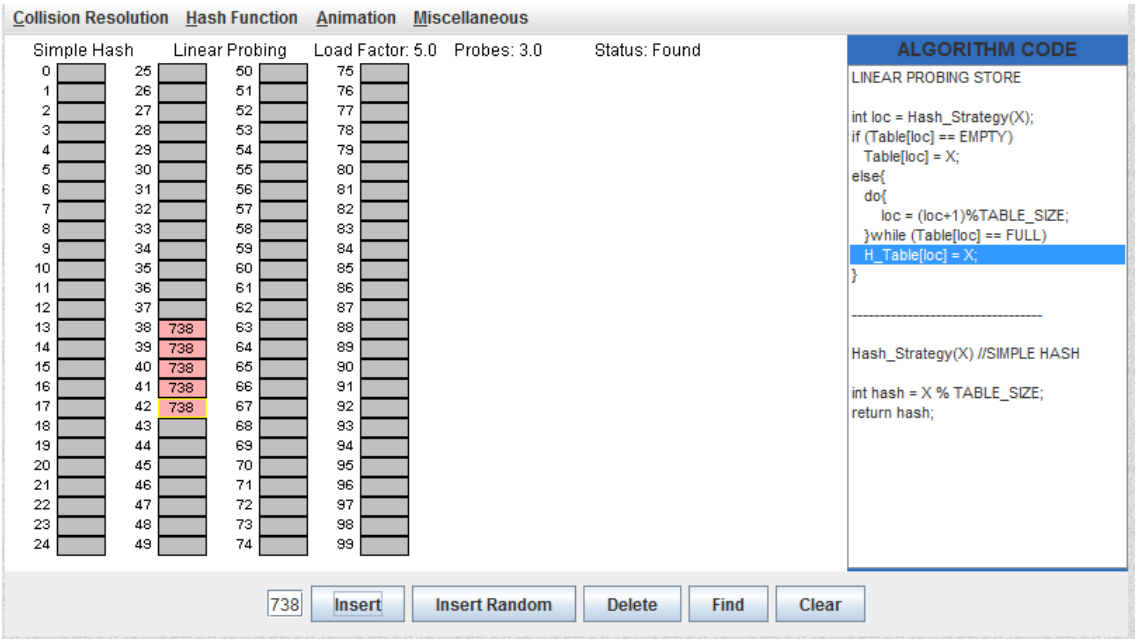
Escrita na Drive 2, dados D1	Na fase de recuperação, a paridade é atualizada
0110 1101 Drive 1 XOR 1111 1110 Drive 2 XOR 0000 0000 Drive 3 1011 1001 Drive 4 Paridade P0	0110 1101 Drive 1 XOR 1111 1110 Drive 2 XOR 0000 0000 Drive 3 1001 0011 Drive 4 Recuperada

Critério de correção:
- (0,2) descrição do procedimento
- (0,3) descrição da recuperação

2) Relativo ao Cap. 11- Indexing and Hashing

No seguinte sítio pode encontrar uma *applet* de *Hash-tables*
http://csilm.usu.edu/lms/nav/activity.jsp?sid=__shared&cid=usu@mills&lid=171

Para as configurações de *Linear Probing* e *Simple Hash* altere o modo de inserção de forma a evitar elementos repetidos. Explique convenientemente o seu código.



Resposta:

O procedimento corrente é o seguinte:

```
void HashTable_Insert (int X)
{  int loc=Hash_Strategy(X)
   if (Table[loc]==Empty) Table[loc]=X;
  else  {   do
              {  loc= (loc +1) % TABLE_SIZE
              } while (Table[loc]==Full )
              Table[loc]=X;
           }
}
```

Para evitar repetidos teremos (alterações a vermelho):

```
void HashTable_InsertUnique (int X)
{  int loc=Hash_Strategy(X)
    if (Table[loc]==Empty) Table[loc]=X;
  else  {   do
              {  loc= (loc +1) % TABLE_SIZE
              }  while (Table[loc]==Full && Table[loc]!=X)
             if (Table[loc]==Empty) Table[loc]=X;
           }
}
```

Critério de correção:
- (0,4) alteração de “Table[loc]!=X”
- (0,1) alteração de “if (Table[loc]==Empty)”
- penalização de 1 ou 5 décimas para incorreções no código

3) Relativo ao Cap. 12- Query Procesing

a) Explique porque é que o SGDB tem de criar planos de execução.

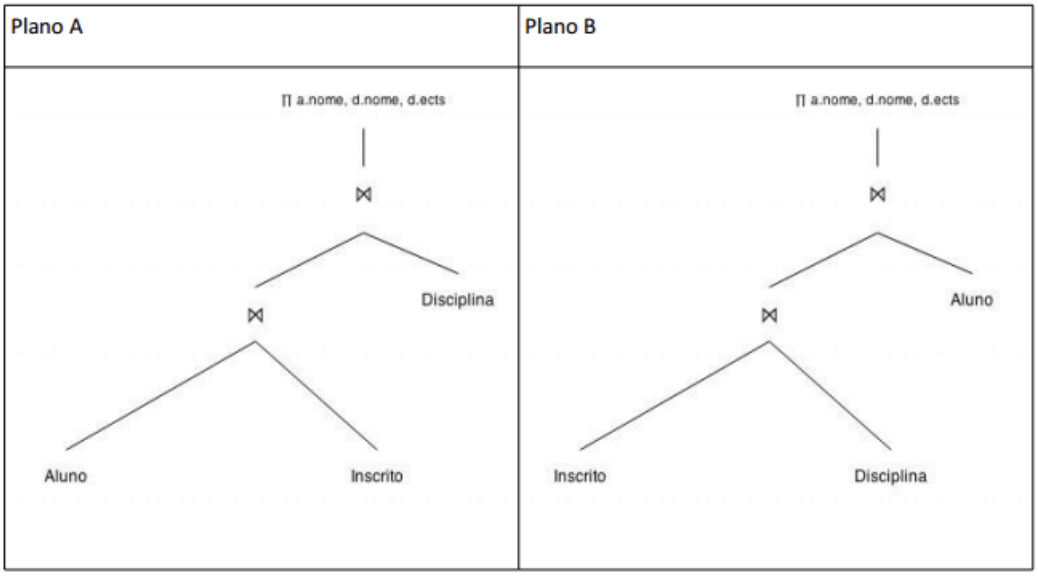
b) Escreva a seguinte consulta em álgebra relacional e desenhe pelo menos dois planos de execução, utilizando um estrutura em árvore.

```
select a.nome, d.nome. d.ects
from aluno a, inscrito i, disciplina d
where a.id= i.aluno_id
and i.disc_id = d.id
```

Resposta:

a) Visto que o SQL é um linguagem declarativa, o SGBD terá de encontrar a sequência de execução ou Plano de Execução.

b) Os dois planos seguintes utilizam a estrutura “left-deep”:



Critério de correção:
- (0,3) alínea a)
- (0,2) alínea b)
- penalização de 1 ou 2 décimas para árvores incorrectas

4) Relativo ao Cap. 13 - Query Optimization

O operador de semi-junção |>< pode ser definido da seguinte forma:
r |><$_\theta$ s = Π_R (r |><|$_\theta$ s) onde R é um conjunto de atributos do esquema r.

a) Considere a consulta da pag. 605, que devolve o nome dos instrutores que ensinaram no ano de 2007.

```
Select name
From instructor
Where exists (Select *
                From teaches
                Where instructor.ID = teaches.ID
                And teaches.year = 2007)
```

Escreva a mesma consulta em álgebra relaciona utilizando o operador de semi-junção.

b) A semi-junção é um operador muito utilizado nas bases de dados distribuídas, explique porquê.

Resposta:

a) Π_{name} [instructor |><$_\theta$ ($\sigma_{year=2007}$ (teaches))]

com θ = (instructor.ID = teaches.ID)

b) Dado a definição de semi-junção (r |><$_\theta$ s) = Π_R (r |><|$_\theta$ s), onde R é um conjunto de atributos do esquema r, podemos constatar que é realizada uma redução nos dados através da projeção Π_R.

A redução dos dados transferidos é manifestamente útil em bases de dados distribuídas.

No exemplo em baixo, o atributo D é omitido na semi-junção:

$$R \ltimes S = \Pi_{Atr(R)}(R \bowtie S)$$

R

A	B	C
c	1	b
d	3	e
f	3	b
d	2	g

S

A	B	D
f	3	e
a	2	j
d	1	n
c	1	e
f	3	b

$R \ltimes S$

A	B	C
c	1	b
f	3	b

Critério de correção:
- (0,3) alínea a) com penalização de 1 ou 2 décimas para incorreções na expressão
- (0,2) alínea b)