



Licenciatura em Engenharia Informática e Computação
Tecnologia de Sistemas de Gestão de Bases de Dados
2001/2002

Exame de Avaliação

28 de Junho de 2002

NOME: _____

Observe por favor as seguintes instruções:

- Leia cuidadosamente o exame até ao fim por forma a escolher a sua estratégia.
- O exame tem a duração máxima de duas horas e meia (150 minutos).
- O exame é com consulta de todo o material próprio trazido para o efeito.
- Deve responder nos espaços fornecidos neste exame, podendo usar, em último recurso, o espaço das costas da folha.
- O exame tem 10 perguntas, com as pontuações indicadas, totalizando 100 pontos.

Problema	1	2	3	4	5	6	7	8	9	10	Total	NOTA
Máx. Pontos	5	15	15	5	15	10	10	10	10	5	100	–
Pontos												

João Correia Lopes

1. Armazenamento de Dados: Ficheiros e Índices [5 pontos]

B-trees são estruturas de dados genéricas que podem ser usadas quando se pretende acesso eficiente aos registos correspondentes a uma dada ordenação de uma chave de procura. Estas árvores multicaminho contêm nós com um conjunto de chaves (por exemplo k) e um conjunto de apontadores (por exemplo $k+1$) para outros nós da árvore.

- a) Em bases de dados usam-se árvores ISAM e B+ trees para conseguir índices que melhoram o desempenho na resposta a interrogações que envolvem procura sequencial em intervalos de uma chave (de procura). Diga quais são as diferenças fundamentais entre ISAM, B-trees e B+ trees.

2. Indexação [15 pontos]

Uma organização de ficheiros possível baseia-se na utilização de funções de dispersão (*hash*) para mapear campos de procura em *buckets* por forma a encontrar a página onde esse campo reside.

- a) Apresente um exemplo de uma situação que mostre que a utilização de um índice deste tipo leva a melhor desempenho do que uma organização de ficheiro com registos por ordem aleatória (*heap file*).

- b) Apresente um exemplo de uma situação que mostre que o uso de uma organização de ficheiro com registos por ordem aleatória (*heap file*) leva a melhor desempenho do que a utilização de um índice usando *hashing*.

- c) Apresente um exemplo de uma situação que mostre que a utilização de um índice usando *Extensible Hashing* leva a melhor desempenho do que a utilização de um índice usando *Linear Hashing*.

3. Optimizaçãõ de interrogações [15 pontos]

Considere o seguinte esquema de relação:

```
Artigo(codigo: integer, nome: string(72),  
       qmin: integer, qmax: integer, fornecedor: integer)
```

em que o código de artigo é chave candidata com valores entre 0 e 4 999 999. Considere ainda que os 5 000 000 registos cabem em 500 000 páginas de dados e que a relação está guardada num ficheiro ordenado pelo campo `codigo`, com índices secundários densos.

Considere que, para responder a interrogações SQL, pode usar:

1. acesso directo ao ficheiro ordenado
 2. um índice em *Linear Hashing* no atributo `codigo`
 3. um índice em *B+ tree* no atributo `codigo`
- a) Apresente os cálculos que indiquem claramente qual das três estratégias apresentadas levaria a um custo mínimo para responder à seguinte interrogação em SQL:

```
SELECT * FROM Artigo WHERE codigo=100000;
```

- b) Idem, para responder à seguinte interrogação em SQL:

```
SELECT * FROM Artigo WHERE codigo<100000;
```

c) Idem, para responder à seguinte interrogação em SQL:

```
SELECT * FROM Artigo WHERE codigo>100000 AND codigo<100010;
```

4. Limitações do Modelo Relacional [5 pontos]

O Modelo Relacional apresenta diversas vantagens em relação a outros modelos de dados mas também tem algumas limitações.

a) Diga em que consiste a limitação do Modelo Relacional conhecida como “desadaptação de impedâncias”.

5. **SQL3, ADTs e Coleções [15 pontos]**

Numa dada escola um aluno inscreve-se num dado departamento e pode frequentar por ano uma cadeira de um curso dado por outro departamento da escola. Considere o esquema SQL3 apresentado de seguida:

```
CREATE ROW TYPE InfoAluno AS (nome STRING, departamento STRING)
CREATE ROW TYPE TipoAluno AS (codigo STRING, info InfoAluno)
CREATE ROW TYPE TipoCadeira AS (id INTEGER, departamento STRING, nome STRING)
```

```
CREATE TABLE Aluno OF TYPE TipoAluno PRIMARY KEY (codigo)
CREATE TABLE Cadeira OF TYPE TipoCadeira PRIMARY KEY (id)
CREATE TABLE Frequenta (aluno REF(TipoAluno),
                        cadeira REF(TipoCadeira), nota INTEGER)
```

- a) Escreva uma interrogação SQL3 que encontra os códigos, excluindo os duplicados, de todos os alunos que obtiveram nota superior a 10 numa cadeira de um curso oferecido por um departamento diferente do departamento a que pertencem.

- b) Escreva um ou mais triggers que impeçam um aluno de frequentar mais do que uma cadeira de um departamento diferente do departamento onde o aluno está inscrito.

- c) Escreva uma asserção que impeça os alunos do Departamento de “Línguas Modernas” (valor `linguas` no atributo `departamento`) de frequentar a cadeira de “Bases de Dados” (valor 24 no atributo `id`).

6. Persistência, Coleções e Módulos [10 pontos]

Considere novamente a base de dados do problema 5.

- a) Apresente um módulo persistente de servidor com a função e o procedimento seguintes:

```
positivas_de(cadeira: STRING) : integer; // numero de notas positivas  
altera_nota(aluno: STRING, cadeira: STRING, nota: integer); // altera nota
```

7. Estrutura Lógica de Documentos XML [10 pontos]

Considere o seguinte DTD (Futebol.dtd) para documentos XML:

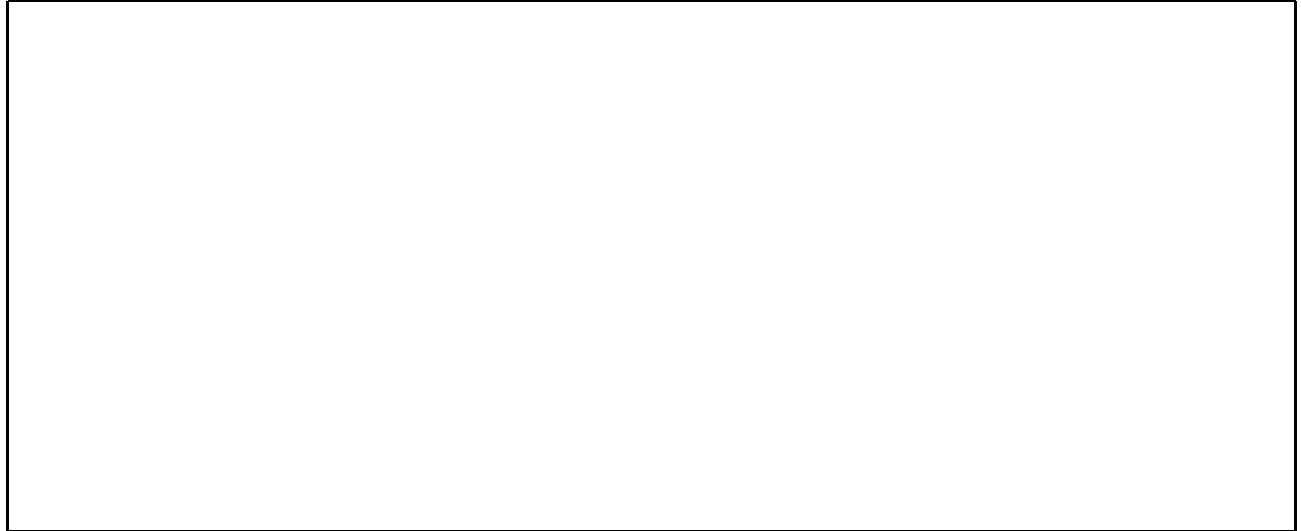
```
<!DOCTYPE Futebol [  
  <!ELEMENT DOC-FUTEBOL (ESTADIO+, EQUIPA+, JOGO*)>  
  <!ELEMENT ESTADIO EMPTY>  
  <!ELEMENT EQUIPA (NOME, CIDADE)>  
  <!ELEMENT JOGO (VISITADO, VISITANTE?)>  
  <!ELEMENT NOME (#PCDATA)>  
  <!ELEMENT CIDADE (#PCDATA)>  
  <!ELEMENT VISITADO (GOLOS | PONTOS)>  
  <!ELEMENT VISITANTE (GOLOS | PONTOS)>  
  <!ELEMENT GOLOS (#PCDATA)>  
  <!ELEMENT PONTOS (#PCDATA)>  
  <!ATTLIST ESTADIO Cod ID #REQUIRED Nome CDATA>  
  <!ATTLIST EQUIPA Cod ID #REQUIRED>  
  <!ATTLIST JOGO Estadio IDREF #REQUIRED>  
  <!ATTLIST VISITADO equipa IDREF>  
  <!ATTLIST VISITANTE equipa IDREF>  
>
```

- a) Verifique se o documento XML seguinte é bem formado e se está conforme com o DTD apresentado (isto é, se é válido); no caso de não estar, assinale os pontos onde isso se verifica.

```
<?XML VERSION="1.0" STANDALONE="no"?>  
<!DOCTYPE Futebol SYSTEM "../DTDs/Futebol.dtd">  
  <DOC-FUTEBOL>  
    <ESTADIO Cod="ES1" Nome="Antas"/>  
    <EQUIPA Cod="EQ1">  
      <NOME>F.C.Porto</NOME>  
      <CIDADE>Porto</CIDADE>  
    </EQUIPA>  
    <ESTADIO Cod="ES2"></ESTADIO>  
    <EQUIPA Cod="EQ2">  
      <NOME>F.C.Porto B</NOME>  
    </EQUIPA>  
    <EQUIPA Cod="ES3">  
      <NOME>F.C.Porto C</NOME>  
    </EQUIPA>  
    <JOGO Estadio="Antas">  
      <VISITADO equipa="EQ1">  
        <GOLOS>5</GOLOS>  
        <PONTOS>3</PONTOS>  
      </VISITADO>  
      <VISITANTE equipa="ES3">  
        <GOLOS>0</GOLOS>  
        <PONTOS>0</PONTOS>  
      </VISITANTE>  
    </JOGO>  
    <JOGO Estadio="ES1">  
      <VISITADO equipa="EQ1">  
        <GOLOS>5</GOLOS></VISITADO>  
      <VISITANTE equipa="EQ2">  
        <GOLOS>0</GOLOS></VISITANTE>  
    </JOGO>
```



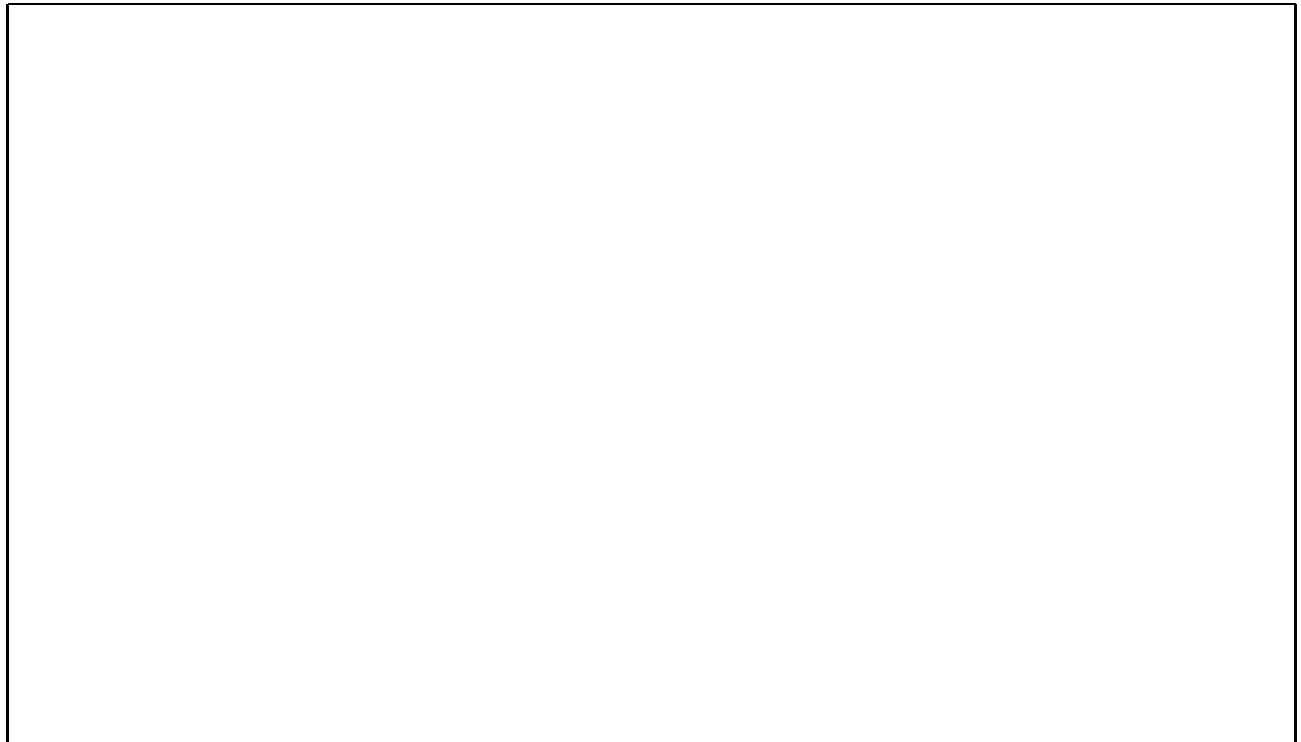
```
<JOGO>
  <VISITADO equipa="EQ1">
    <PONTOS>0</PONTOS></VISITADO>
  <VISITANTE equipa="EQ1">
    <PONTOS>0</PONTOS></VISITANTE>
</JOGO>
</DOC-FUTEBOL>
```



8. Transformação e apresentação de XML [10 pontos]

Considere novamente o DTD apresentado no problema 7.

- a) Apresente um conjunto de regras de transformação XSLT que permitam passar para HTML, para ser mostrado num navegador Web, o nome da equipa visitada em cada jogo e o número de pontos que lhe estão associados em instâncias de documentos XML de acordo com este DTD.



9. XML Schemas [10 pontos]

Considere novamente o DTD apresentado no problema 7.

- a) Considerando apenas os elementos ESTADIO e EQUIPA, apresente a parte de um XML Schema equivalente ao DTD (isto é, que permita as mesmas instâncias de documentos XML)

10. Gestão de Transacções [5 pontos]

Considere o seguinte escalonamento de transacções (ainda incompleto), que contém, por ordem temporal, as seguintes operações de leitura e escrita:

T1:R(A), T1:W(B), T3:W(B), T2:R(B), T1:W(A), T2:R(B)

em que, por exemplo, T1:R(A) é operação de leitura no item A efectuada pela transacção T1 e T3:W(B) é operação de escrita no item B efectuada pela transacção T3.

- a) Averigue se o escalonamento apresentado é possível e seriável; no caso afirmativo apresente o escalonamento série equivalente e no caso contrário apresente acções que o tornam seriável.

FIM.