

Licenciatura em Engenharia Informática e Computação
Tecnologia de Sistemas de Gestão de Bases de Dados
2000/2001

Exame de Avaliação

29 de Junho de 2001

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 exacta 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 150 pontos; as pontuações das perguntas são sensivelmente proporcionais ao tempo aconselhado para responder, por forma a terminar o exame no tempo estabelecido.

Problema	1	2	3	4	5	6	7	8	9	10	Total
Máx. Pontos	10	35	15	10	20	10	20	10	10	10	150
Pontos											

João Correia Lopes

1. Armazenamento de dados: ficheiros e índices [10 pontos]

Considere uma relação guardada em ficheiro de registos por ordem aleatória (*heap file*) para a qual existe um índice não aglomerado num campo *nota*.

- a) Diga se para responder a uma interrogação que pretende os registos com $nota > 10$ o uso deste índice é a melhor escolha e justifique devidamente a sua resposta.

2. Ordenação externa [35 pontos]

Considere um disco com tempo médio de procura (*seek time*) de 10ms, *rotational delay* de 5ms e tempo de transferência de 1ms para uma página de 4k. O custo de ler ou escrever uma página é, então, de 16ms. No entanto, se as páginas forem lidas ou escritas em sequência, o tempo total será igual ao tempo para encontrar a primeira página (10ms+5ms) mais um 1ms por página para transferir os dados.

Considere que tem 320 páginas de buffer e pretende ordenar um ficheiro com 10 000 000 de páginas.

- a) Explique porque é que não é boa ideia usar as 320 páginas como suporte de memória virtual e escolher um algoritmo de ordenação para memória primária, por exemplo Quicksort.

- b) Avalie o custo em I/O de páginas de obter corridas ordenadas de 320 páginas cada, no primeiro passo de um algoritmo de ordenação externa.

- c) Avalie o custo de completar a ordenação através de junções de 319-vias (*319-way merge*).

- d) Avalie o custo de completar a ordenação através de junções de 256-vias, com 256 buffers de entrada de 1 página e um buffer de saída de 64 páginas.

3. Optimização de interrogações [15 pontos]

Considere o seguinte esquema relacional:

```
Fornecedor(id: integer, nome: char(40), cidade: char(20))
Peças(id: integer, nome: char(40), preço: real)
Fornece(f-id: integer, p-id: integer)
```

e a seguinte interrogação em SQL:

```
SELECT F.nome, P.nome
FROM Fornecedores F, Peças P, Fornece N
WHERE F.id=N.f-id AND P.id=N.p-id AND F.cidade='Porto' AND P.preço<1000;
```

- a) Diga que informação acerca destas relações deve possuir o optimizador para poder seleccionar um bom plano de execução para a interrogação dada.

b) Explique que índices poderiam ajudar na execução desta interrogação.

c) Diga de que maneira poderiam ser afectados os planos produzidos se fosse adicionado DISTINCT à interrogação.

4. Limitações do Modelo Relacional e ODMG [10 pontos]

Aplicações de suporte a CAD, CAM, CASE, multimédia, publicação digital e GIS têm colocado a nu limitações do Modelo Relacional e levaram no passado recente à introdução de outros modelos.

a) Enumere e caracterize as principais fraquezas do Modelo Relacional e refira as melhorias propostas noutros modelos de dados, mais adaptados a estas aplicações.

5. SQL3, ADTs e Colecções [20 pontos]

Suponha que tem uma base de dados com informação relativa a uma árvore genealógica, com a seguinte informação de pessoas e respectivos casamentos. A pessoa é identificada pelo seu bi e guarda-se informação acerca do nome, data de nascimento, identificação da mãe, identificação do pai, sexo e morada (rua e cidade). Dos casamentos guarda-se a identificação das duas pessoas, a data em que se realizou, a data do divórcio, a identidade de todos os filhos e, de forma redundante, o número de filhos. Consideram-se válidas as seguintes restrições de integridade:

R1: O bi de pessoa é chave candidata.

R2: O bi do marido em conjunto com o bi da mulher formam uma chave candidata de casamento.

R3: O sexo do marido (mulher) é 'M' ('F')

R4: A data do divórcio é maior que a data do casamento.

R5: A idade de um filho é sempre menor que a dos pais.

R6: Para qualquer casamento, o número de filhos é igual ao número de pessoas com a mãe igual à mulher do casamento e o pai igual ao marido do casamento.

- a) Considere que tem ao seu dispor um SGBD relacional-objecto, como por exemplo o Oracle 8i ou o PostgreSQL, com a possibilidade de definir, por exemplo, tipos abstractos, tipos colecção, tipos referência e tabelas encaixadas. Apresente um esquema relacional-objecto (através dos respectivos comandos SQL) para a árvore genealógica apresentada, sem esquecer as restrições de integridade R1 a R4.

6. Módulos Persistentes em SQL3 [10 pontos]

Para a base de dados do problema 5, por forma a simplificar as respostas, considere o seguinte esquema relacional:

```
Pessoa (bi: CHAR(8), nome, data_nascimento, bi_mae, bi_pai, sexo)
Casamento (bi_marido, bi_mulher, data_inicio, data_fim, #filhos)
```

a) Apresente um módulo persistente de servidor com os seguintes procedimentos:

```
numeroFilhos(bi_pessoa) // devolve o número de filhos da pessoa
casadaCom(bi_mulher) // devolve a identificação do marido da pessoa
numeroIrmaosDe(bi_pessoa) // devolve o número de irmãos da pessoa
```

7. Restrições de Integridade e Gatilhos [20 pontos]

Considere novamente a base de dados e o respectivo esquema relacional apresentado no problema 6.

- a) Escreva uma asserção em SQL3 para impôr a restrição R5.

- b) Escreva um ou mais gatilhos em SQL para impor a restrição R6 de forma incremental: ao inserir, eliminar ou actualizar uma linha da tabela **Pessoa**, deve ser actualizada, se existir, a instância respectiva da tabela **Casamentos**.

8. Estrutura Lógica de Documentos XML [10 pontos]

Considere o seguinte DTD para documentos XML:

```
<!DOCTYPE Univ [
  <!ELEMENT UNIVERSIDADE (DEPARTAMENTO+)>
  <!ELEMENT DEPARTAMENTO (DOCENTE*)>
  <!ATTLIST DEPARTAMENTO Cod ID #REQUIRED
              Nome CDATA #REQUIRED>
  <!ELEMENT DOCENTE (LOCAL, CARGO+)>
  <!ATTLIST DOCENTE Cod ID #REQUIRED
                  Nome CDATA #REQUIRED
                  Tipo (noite | dia ) "dia"
                  Departamentos IDREFS>
  <!ELEMENT LOCAL (#PCDATA)>
  <!ELEMENT CARGO (#PCDATA)>
]>
```

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

```
<?XML VERSION="1.0" STANDALONE="no"?>
<!DOCTYPE Univ SYSTEM "../DTDs/Univ.dtd">
<UNIVERSIDADE>
  <DEPARTAMENTO Cod="D1" Nome="DEEC">
    <DOCENTE Cod="JCL" Nome="jlopes" Departamentos="D1">
      <LOCAL>Manteigas</LOCAL>
      <CARGO>Prof. Auxiliar</CARGO>
      <CARGO>Prof. Agregado</CARGO>
    </DOCENTE>
    <DOCENTE Cod="JCL2" Nome="jlopes" Tipo="noite">
      <LOCAL>Porto</LOCAL>
    </DOCENTE>
  </DEPARTAMENTO>
  <DEPARTAMENTO Cod="D2" Nome="DEQ">
    <DOCENTE Cod="JPF" Nome="jpascoal" Tipo="dia" Departamentos="D1">
      <LOCAL>Porto</LOCAL>
      <CARGO>Prof. Auxiliar</CARGO>
      <LOCAL>Manteigas</LOCAL>
      <CARGO>Prof. Agregado</CARGO>
    </DOCENTE>
  </DEPARTAMENTO>
  <DEPARTAMENTO Cod="D3" Nome="DEC">
    <DOCENTE Sigla="JCA" Departamentos="JCL JPF">
      <CARGO>Prof. Auxiliar</CARGO>
      <CARGO>Prof. Agregado</CARGO>
    </DOCENTE>
  </DEPARTAMENTO>
  <DEPARTAMENTO Cod="D3" Nome="DEC">
  </DEPARTAMENTO>
</UNIVERSIDADE>
```

9. Transformação e apresentação de XML [10 pontos]

Considere novamente o DTD apresentado no problema 8.

- a) Apresente um conjunto de regras de transformação XSLT que permitam passar para HTML para ser mostrado num navegador Web, os nomes dos docentes constantes de documentos XML de acordo com este DTD.

10. Sistemas de Apoio à Decisão [10 pontos]

Considere a seguinte tabela de factos e uma das tabelas de dimensões:

Vendas(prodoto, cor, dia, quantidade, preco)

Produtos(id, categoria, preco)

Suponha que é submetida a seguinte interrogação:

```
SELECT produto, dia, SUM(quantidade*preco)
FROM Vendas, Produtos
GROUP BY categoria, dia;
```

- a) Apresente uma interrogação em SQL que, aplicada de seguida, efectue uma operação de “roller-up”.

- b) Apresente um exemplo em SQL de aplicação de uma das operações de “pivoting”, por exemplo, “slicing”.

FIM.