

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

Exame de Avaliação

26 de Julho 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 120 pontos; as pontuações das perguntas são sensivelmente proporcionais ao tempo aconselhado para responder, por forma a terminar o exame no tempo estabelecido (o exame tem meia-hora de tolerância).

| Problema | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Total | NOTA |
|-------------|----|----|----|----|----|----|----|----|----|----|-------|------|
| Máx. Pontos | 10 | 15 | 15 | 10 | 15 | 15 | 10 | 10 | 10 | 10 | 120 | - |
| Pontos | | | | | | | | | | | | |

João Correia Lopes

1. Armazenamento de Dados: Ficheiros e Índices [10 pontos]

O gestor de buffers de um SGBD pode marcar (*pin*) páginas para impedir que sejam trocadas por outras do disco e mantém uma indicação das páginas de buffer com modificações (*dirty bit*).

- a) Descreva o que acontece se há um pedido para uma página que não está em buffer numa altura em que todas as páginas têm modificações.

2. Indexação [15 pontos]

Considere o seguinte esquema de relação:

`Docentes (bi, nome, telefone, departamento)`

Considere uma instância com 100000 registos e que o atributo `bi` é chave candidata da relação; este atributo tem valores entre 0 a 99999 e cabem 10 registos por página.

A relação está guardada num ficheiro de registos por ordem leatória (*heap file*) com índices secundários densos.

Considere por último a seguinte interrogação em SQL:

```
SELECT D.nome
FROM Docentes D
WHERE D.bi > 100 AND D.bi < 200;
```

- a) Apresente os cálculos que achar necessários para determinar qual é o plano com custo mínimo para responder à interrogação apresentada considerando que, depois de encontrar a entrada de dados do índice, é necessário mais um acesso a disco para aceder ao registo, para todos os seguintes casos:

1. um `scan` do ficheiro;
2. um índice em B+ tree no campo `bi` com M (por exemplo 10) entradas por página de dados do índice e altura h (por exemplo 2);
3. uma tabela de dispersão (*hash*) no campo `bi` com factor de ocupação c (por exemplo 0,8).

3. Optimização de interrogações [15 pontos]

Considere o seguinte esquema relacional:

```
Emp(eid: integer, salario: integer, idade: real, did: integer)
Proj(pid: integer, codigo: integer, relato: varchar)
Dept(did: integer, pid: integer, orcamento: real, estado: char(10))
```

e a seguinte interrogação em SQL:

```
SELECT *
FROM Emp E, Dept D
WHERE E.did=D.did;
```

Considere que existem 20000 tuplos em `Emp` e que cada um ocupa 20 bytes; que existem 5000 tuplos em `Dept` e que cada um ocupa 40 bytes; e que existem 1000 tuplos em `Proj` e que cada um ocupa, em média, 2k bytes. Considere ainda que cada departamento identificado por (*did*) detém, em média, 10 projectos (*pid*).

O sistema de ficheiros suporta página de 4k bytes e existem 12 páginas de buffer.

- a) Supondo a existência apenas de um índice aglomerado do tipo tabela de dispersão no campo *did* de **Emp**, enumere os planos que devem ser considerados pelo otimizador para responder à pergunta e calcule uma estimativa de custos para cada um por forma a escolher o melhor.

4. Limitações do Modelo Relacional [10 pontos]

Um SGBD Orientado por Objectos tem de fornecer suporte para guardar objectos persistentes, isto é, que sobrevivem ao programa que os criou.

A norma ODMG2.0 tem estabelecidos *bindings* para diversas linguagens de programação, nomeadamente para Java, caso em que os objectos não podem ser explicitamente apagados quando deixam de ser úteis.

- a) Identifique e descreva brevemente o método especificado na norma para implementar o suporte aos objectos Java capazes de persistir.

5. SQL3, ADTs e Colecções [15 pontos]

Suponha que pretende guardar alguma informação de uma companhia de aviação.

Os tipos de avião em uso são identificados por um código, têm uma designação (Airbus 430, Boeing 343, Concorde), uma autonomia (distância máxima em km que podem percorrer sem abastecer) e é ainda guardada uma foto.

Os empregados da companhia podem ser identificados por um código, é ainda guardado o nome e o seu salário e pode saber-se, em qualquer altura, a sua idade.

Para os empregados que são pilotos é possível identificar os tipos de avião que podem pilotar.

Considere ainda válidas as seguintes restrições de integridade:

R1: Nenhum piloto ganha menos do que 3000.

R2: Qualquer um dos pilotos ganha mais do que qualquer um dos empregados que não é piloto.

R3: Um piloto perde a capacidade para pilotar aviões com autonomia superior a 10000 km quando o seu salário subir acima de 10000.

- a) Considere que tem ao seu dispor um SGBD relacional-objecto, SQL3, com a possibilidade de definir tipos abstractos, tipos colecção, tipos referência e tabelas encaixadas. Apresente um esquema relacional-objecto, por exemplo usando a notação apresentada nas aulas, para os requisitos enumerados para a aplicação referida, sem esquecer a primeira restrição de integridade (R1).

6. Módulos Persistentes em SQL3 [15 pontos]

Considere novamente a base de dados do problema 5.

a) Apresente um módulo persistente de servidor com as seguintes funções e procedimentos:

```
// devolve o total de pilotos com capacidade para pilotar o tipo
// de avião tpa e que têm salario inferior a sal.
quantosPilotam(tpa, sal): integer;
// escreve no stdout o tipo dos aviões em que todos os empregados
// que os pilotam ganham mais do que o salario sal;
todosPilotam(sal);
```

supondo que dispõe da função `printout(arg)` que escreve no *standard output* o valor de `arg`.

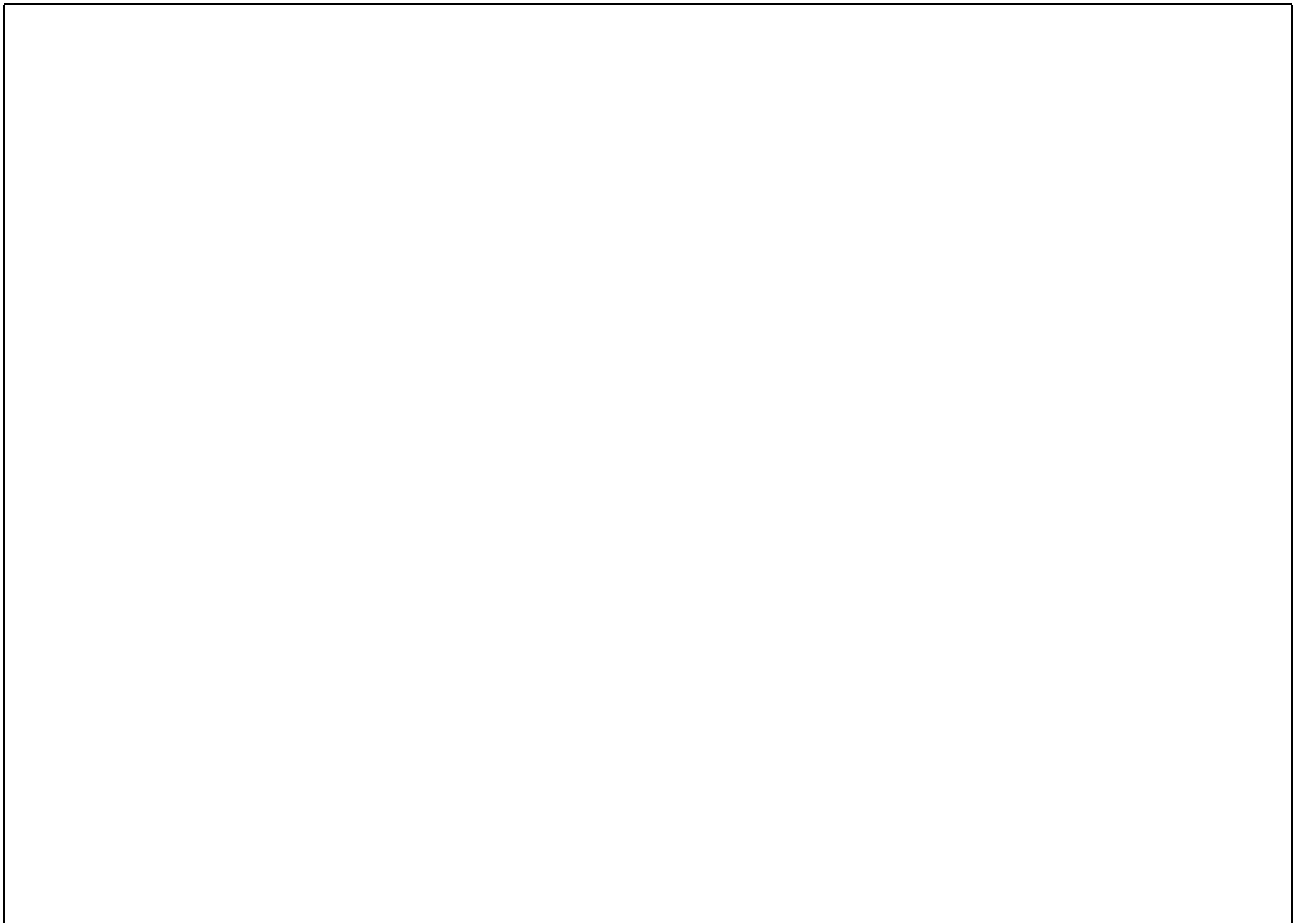
7. Restrições de Integridade e Gatilhos [10 pontos]

Considere novamente a base de dados do problema 5.

- a) Escreva uma asserção para impor a restrição R3.



- b) Escreva um ou mais gatilhos em SQL para impor a restrição R2 de forma incremental ao ser actualizada a informação sobre empregados.



8. Estrutura Lógica de Documentos XML [10 pontos]

Considere o seguinte DTD para documentos XML:

```
<!DOCTYPE Exames [
  <!ELEMENT TRAP-AirP (TIPOAVIAO | VOO)*>
  <!ELEMENT TIPOAVIAO (AVIAO)*>
  <!ATTLIST TIPOAVIAO Cod ID #REQUIRED Nome CDATA>
  <!ELEMENT AVIAO EMPTY>
  <!ATTLIST AVIAO Cod ID #REQUIRED Horas CDATA>
  <!ELEMENT VOO (DATA)*>
  <!ATTLIST VOO Cod ID #REQUIRED De CDATA #REQUIRED Para CDATA #REQUIRED
    TipoAviao IDREFS #REQUIRED>
  <!ELEMENT DATA (AVIAO, (PILOTO)+)*>
  <!ATTLIST DATA Dia CDATA #REQUIRED Mes CDATA #REQUIRED Ano CDATA #REQUIRED>
  <!ELEMENT PILOTO EMPTY>
  <!ATTLIST PILOTO Cod CDATA #REQUIRED Horas CDATA>
]>
```

- a) Verifique se o seguinte documentos XML está conforme com o DTD apresentado (é válido) e, no caso de não estar, assinale os pontos onde isso se verifica e diga qual é o erro.

```
<?XML VERSION="1.0" STANDALONE="no"?>
<!DOCTYPE Exames SYSTEM "../DTDs/Exame.dtd">
  <TRAP-AirP>
    <TIPOAVIAO Cod="T10" Nome="Concorde">
      <AVIAO Cod="A100" Horas="44232"/>
      <AVIAO Cod="A101" Horas="34552"/>
      <AVIAO Cod="A102" Horas="88597"/>
    </TIPOAVIAO>
    <TIPOAVIAO Cod="T11" Nome="A300">
      <AVIAO Cod="A200" Horas="123545"/>
      <AVIAO Cod="A201" Horas="433545"/>
      <AVIAO Cod="A202" Horas="685897"/>
    </TIPOAVIAO>
    <VOO Cod="TP100" De="Porto" Para="New York" TipoAviao="T20">
      <DATA Dia="26" Mes="Julho" Ano="2001">
        <AVIAO Cod="A100"/>
        <PILOTO Cod="P10" Horas="10234"/>
        <PILOTO Cod="P24" Horas="15734"/>
      </DATA>
      <DATA Dia="29" Mes="Julho" Ano="2001">
        <AVIAO Cod="A200"/>
        <PILOTO Cod="P34" Horas="23534"/>
        <AVIAO Cod="A201"/>
        <PILOTO Cod="P24" Horas="34734"/>
      </DATA>
    </VOO>
    <VOO Cod="TP200" De="Porto" Para="Heathrow" TipoAviao="T11">
      <DATA Dia="26" Mes="Julho" Ano="2001">
        <AVIAO Cod="A201"/>
        <PILOTO Cod="P34"/>
        <PILOTO Cod="P24"/>
      </DATA>
    <VOO Cod="TP300" De="Porto" Para="Los Angeles">
    </VOO>
  </TRAP-AirP>
```


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, as datas e os pilotos do voo TP100 existentes em documentos XML de acordo com o DTD apresentado.

10. Gestão de Transacções [10 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:R(B), T1:W(A), T2:R(B), T3:W(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.