FEUP Universidade do Porto
Faculdade de Engenharia

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

Exame de Avaliação

13 de Julho de 2001

NOME: **Resolução do Exame (30 de Julho de 2001)** _____

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

   Os Sistemas de Gestão de Base de Dados (SGBD) fazem a sua própria gestão de disco e de páginas em memória primária (*buffer management*) pouco usando das facilidades correspondentes do Sistema Operativo (SO) onde executam.

   a) Identifique dois mecanismos mantidos pela gestão de memória em SGBD que não se encontram em gestão de memória dos SO. Diga em que consistem e para que servem esses dois mecanismos.

   Resposta:
   Distintamente diferente dos Sistemas Operativos nas políticas de troca de páginas, os SGBDs usam: marcar uma página (**pin**) para impedir que seja trocada pela gestão de páginas; e **forçar** uma página para disco (necessário para permitir recuperação garantindo que fica em memória não volátil). [Para além destes mecanismos: os SGBDs (por saberem de índices, etc.) usam *prefetch* de forma distinta dos SOs; e os ficheiros nas BDs, ao contrário dos SOs, podem atravessar discos.]

2. **Indexação em Árvore [15 pontos]**

   Considere uma relação guardada num ficheiro de registos por ordem aleatória (*heap file*) com 20 000 registos e páginas de 1kB. Considere ainda que acaba de construir um índice em B+tree, usando o algoritmo de *bulk-loading* apresentado nas aulas, enchendo ao máximo os nós dos vários níveis da árvore. O índice é denso, usa a Alternativa (2) e a sua chave de procura é uma string com 40 bytes de comprimento, que é chave candidata da relação. Todos os apontadores (*ids* de página de índice e *ids* de registo de dados) têm no máximo 10 bytes de comprimento. As páginas em disco têm 1kB.

   a) Apresente os cálculos que achar necessários para determinar quantos níveis tem a árvore construída.

   Resposta:
   Já que o índice é denso e a chave de procura é chave da relação, teremos tantas entradas de dados quantos os registos do ficheiro (20 000).
   Cada página de índice tem no máximo $2d$ chaves e $2d+1$ apontadores; assim teremos $2d*40+(2d+1)*10 \leq 1000$. Logo $d=9$ e temos 18 chaves e 19 apontadores em cada página de índice.
   Usando a Alternativa (2), um registo de página folha tem uma chave ($k*$) e um apontador (rid) o que dá $40+10=50$ bytes. Cada página folha tem espaço para $1000/50=20$ entradas.

O número de níveis é $\lceil log_{19}(20000/20) + 1 \rceil = 4$.

b) Calcule quantos nós existem em cada nível da árvore.

Resposta:

Considerando que cada nó está o mais cheio possível, há $\lceil 20000/20 \rceil = 1000$ nós folha no nível 4.

Um nó de índice cheio tem 19 filhos e por isso há $\lceil 1000/19 \rceil = 53$ páginas de índice no nível 3, $\lceil 53/19 \rceil = 3$ páginas de índice no nível 2 e uma página de índice no nível 1 (raiz).

3. Optimização de interrogações [15 pontos]

Considere o seguinte esquema de relação:

```
Docentes(nome: char(20), cargo: char(20), cidade: char(20), dept: char(20))
```

e a seguinte interrrogação em SQL:

```
SELECT D.cargo, D.nome
FROM Docentes D
WHERE D.cargo="Prof Auxiliar";
```

Considere ainda que apenas 10% dos registos satisfaz a condição de selecção, que a relação contém 10 000 páginas e que existem 10 páginas de buffer.

a) Supondo a existência apenas de um índice aglomerado do tipo B+ tree no campo *nome*, calcule o custo em I/O e apresente o melhor plano para responder à pergunta.

Resposta:

Olhando para a condição WHERE podemos concluir que um índice B+tree aglomerado em *nome* não ajuda nada. Por isso o melhor plano é um *scan* com custo 10000 I/O.

b) Supondo a existência apenas de um índice aglomerado do tipo B+ tree no campo *cargo*, calcule o custo em I/O e apresente o melhor plano para responder à pergunta.

Resposta:

Para o melhor plano, usar a B+tree no campo *cargo* ajuda, uma vez que há uma selecção nesse campo.

O plano envolve a procura do primeiro cargo que verifica a condição (da raiz ao 1°, custo típico 2 ou 3) e, porque o índice é aglomerado, as páginas da relação podem ser percorridas a partir dessa referência de índice. Para isso é necessário percorrer os apontadores das páginas de índice se for usada a Alternativa (2).

O custo é (número página de índice + número de páginas) * factor de selecção.

Assim, verificando que o campo índice tem 1/4 do tamanho do registo, temos para o custo total 2 + 10% * (2500+10000) = 1252 I/O.

4. Limitações do Modelo Relacional [10 pontos]

Suporte para tipos de dados abstractos (ADT) foi primeiramente introduzido nas bases de dados nos sistemas INGRES e POSTGRES da Universidade de Berkeley, em meados da década de 80.

a) Enumere as limitações do modelo relacional que podem ser soluccionadas pelo uso de ADTs e mostre como isso é possível.

Resposta:

O Modelo Relacional está desadaptado a dados com estrutura complexa ou que tenham associadas funções específicas para operar com eles (e.g. dados multimédia). Este problema do fosso semântico e do número limitado e não extensível de operações, é resolvido pelo uso de ADTs.

Para além disso, passa a haver polimorfismo de inclusão (e herança), encapsulamento e mecanismos de protecção.

5. SQL3, ADTs e Colecções [15 pontos]

Suponha que pretende guardar o enunciado de exames (do tipo deste exame a que está a responder).

Um exame tem um título, uma data, uma cotação total em pontos e é constituído por perguntas. Cada pergunta tem uma cotação e pode incluir alíneas ou figuras. Para cada pergunta deve ser possível identificar todas as suas alíneas e figuras; para cada alínea e para cada figura deve ser possível indentificar a pergunta de que faz parte. As alíneas têm um texto (que pode ser relativamente longo). As figuras podem ser pedidas (para visualização) em formato *gif* ou *eps*.

Considere válidas as seguintes restrições de integridade:

R1: Todas as cotações têm de ser maiores do que zero.

R2: O exame tem uma cotação máxima de 150 pontos.

R3: Nenhuma pergunta pode ter cotação superior a 30 pontos.

R4: A cotação do exame é igual á soma das cotações das suas perguntas.

R5: Uma pergunta não pode conter mais do que 4 alíneas.

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 de exames apresentada, sem esquecer as restrições de integridade R1 a R3.

Resposta:

```
// usando a notação usada nas Teóricas
CREATE TYPE Figura (
    texto       BLOB(1M),
    pergunta    REF(Pergunta),
    public function getGif() returns bitplane,
    public function getJpg() returns bitplane
);
CREATE TYPE Alinea (
    texto       CLOB(1k),
    pergunta    REF(Pergunta)
);
CREATE TYPE Pergunta (
    pontos      INT CHECK (pontos > 0 AND pontos < 30),
    alineas     LIST(REF(Alinea)),
    figuras     LIST(REF(Figura))
);
CREATE TABLE Exames (
    codigo      INT PRIMARY KEY,
    titulo      CHAR(30),
    data        DATE,
    pontos      INT CHECK (pontos > 0 AND pontos < 150),
    perguntas   TABLE (pergunta Pergunta)
);
```

NOTA: No entanto, para responder às perguntas seguintes prefiro usar este esquema alternativo.

```
CREATE TYPE Figura (
    texto       BLOB(1M),
    pergunta    REF(Pergunta),
```

```
    public function getGif() returns bitplane,
    public function getJpg() returns bitplane
);
CREATE TYPE Alinea (
    texto       CLOB(1k),
    pergunta    REF(Pergunta)
);
CREATE TYPE Pergunta (
    alineas     LIST(REF(Alinea)),
    figuras     LIST(REF(Figura))
);
CREATE TABLE Perguntas (
    codigo      INT PRIMARY KEY,
    pergunta    Pergunta,
    pontos      INT CHECK (pontos > 0 AND pontos < 30),
    exame       INT REFERENCES Exames
);
CREATE TABLE Exames (
    codigo      INT PRIMARY KEY,
    titulo      CHAR(30),
    data        DATE,
    pontos      INT CHECK (pontos > 0 AND pontos < 150),
);
```

6. Módulos Persistentes em SQL3 [10 pontos]

Considere novamente a base de dados do problema 5.

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

```
pontosDoExame(exame): integer; // soma o total de pontos das perguntas
alineasDoExame(exame): integer; // conta n. de alineas do exame
```

Resposta:

```
CREATE MODULE Exames
        LANGUAGE SQL;
    CREATE FUNCTION pontosDoExame(ex: INT): INT
    BEGIN
        DECLARE soma INT = 0;
        SELECT SUM(pontos) INTO soma
        FROM Perguntas p
        WHERE p.exame = ex;
        RETURN soma
    END;
    CREATE FUNCTION alineasDoExame(ex: INT): INT
    BEGIN
        DECLARE nalieas INT = 0;
        SELECT count(*) INTO nalieas
        FROM Perguntas p, TABLE (p.pergunta..alineas) AS a
        WHERE p.exame = ex;
        RETURN nalieas
    END;
END;
```

7. Restrições de Integridade e Gatilhos [15 pontos]

Considere novamente a base de dados do problema 5.

a) Escreva um ou mais gatilhos em SQL para impor a restrição R4 de forma incremental: ao inserir, eliminar ou actualizar uma pergunta, deve ser actualizada a cotação do exame.

Resposta:

```
CREATE TRIGGER cotacao_R4_1
    AFTER
        INSERT ON Perguntas
        UPDATE OF pontos ON Perguntas
    REFERENCING NEW AS new_t
    FOR EACH ROW
    WHEN ( TRUE )
    BEGIN
        UPDATE Exames
          SET pontos = pontosDoExame(new_t.exame)
          WHERE codigo = new_t.exame;
END;
CREATE TRIGGER cotacao_R4_2
    AFTER
        DELETE ON Perguntas
    REFERENCING OLD AS old_t
    FOR EACH ROW
    WHEN ( TRUE )
    BEGIN
        UPDATE Exames
          SET pontos = pontosDoExame(old_t.exame)
          WHERE codigo = old_t.exame;
END;
CREATE TRIGGER cotacao_R4_3
    AFTER
        UPDATE OF exame ON Perguntas
    REFERENCING NEW AS new_t
    FOR EACH ROW
    WHEN ( new_t.exame <> old_t.exame )
    BEGIN
        UPDATE Exames SET
          pontos = pontosDoExame(new_t.exame)
          WHERE codigo = new_t.exame;
        UPDATE Exames
          SET pontos = pontosDoExame(old_t.exame)
          WHERE codigo = old_t.exame;
END;
```

b) Escreva uma asserção para impor a restrição R5.

Resposta:

```
CREATE ASSERTION alineas_R5
    AFTER
        UPDATE OF alineas ON Perguntas
    CHECK ( NOT EXISTS
        ( SELECT * FROM Pergunta AS p
          WHERE 4 < ( SELECT count(*) FROM TABLE (p.alineas))
        ) );
```

8. Estrutura Lógica de Documentos XML [10 pontos]

Considere o seguinte DTD para documentos XML:

```
<!DOCTYPE Exames [
  <!ELEMENT DOC-EXAME (DISCIPLINA+,EXAME*)>
  <!ELEMENT DISCIPLINA EMPTY>
  <!ATTLIST DISCIPLINA Cod ID #REQUIRED Nome CDATA #REQUIRED>
  <!ELEMENT EXAME (PERGUNTA*)>
  <!ATTLIST EXAME Cod ID #REQUIRED Data CDATA Pontos CDATA Disciplinas IDREFS>
  <!ELEMENT PERGUNTA (ALINEA | FIGURA)*>
  <!ATTLIST PERGUNTA Pontos CDATA>
  <!ELEMENT ALINEA (TEXTO)>
  <!ATTLIST ALINEA Pontos CDATA>
  <!ELEMENT FIGURA (TEXTO)>
  <!ATTLIST FIGURA Formato (gif | eps) "eps">
  <!ELEMENT TEXTO (#PCDATA)>
]>
```

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.

```
<?XML VERSION="1.0" STANDALONE="no"?>
<!DOCTYPE Exames SYSTEM "../DTDs/Exame.dtd">
  <DOC-EXAME>
    <DISCIPLINA Cod="D1" Nome="SBD"/>
    <DISCIPLINA Cod="D2" Nome="AW"/>
    <DISCIPLINA Cod="D3" Nome="AW"/>
    <DISCIPLINA Cod="D5" Nome="TSGBD"/>
    <EXAME Cod="E1" Data="20010622" Disciplinas="D1 D2">
      <PERGUNTA Pontos="40">
        <ALINEA Pontos="10">Descreva a cor do céu.</ALINEA> // 1.
        <ALINEA Pontos="10">Descreva a cor do mar.</ALINEA> // 2.
        <FIGURA Pontos="20"/> // 2. 3.
      </PERGUNTA>
      <PERGUNTA Pontos="20">
        <ALINEA>Descreva a cor do terra.</ALINEA> // 1.
      </PERGUNTA>
      <PERGUNTA Pontos="20">
        <FIGURA>00030406FFDE34DDDED5AED865F2</FIGURA> // 1.
      </PERGUNTA>
    </EXAME>
    <EXAME Cod="E2" Data="20010709" Disciplinas="E1">
      <PERGUNTA Pontos="20">
        <ALINEA Pontos="10"/> // 3.
        <ALINEA Pontos="10">Descreva fama.</ALINEA> // 1.
        <FIGURA formato="gif">30406FFDE34DDDED5AED8</FIGURA> // 1. 4.
      </PERGUNTA>
      <PERGUNTA Pontos="30">
        <ALINEA Pontos="10">Descreva terror.</ALINEA> // 1.
        <FIGURA Formato="eps">30406FFDE34DDDED5AED8</FIGURA> // 1.
      </PERGUNTA>
      <PERGUNTA Pontos="20"/>
    </EXAME>
    <EXAME Cod="E2" Data="20010719" Disciplinas="AW"> // 5. 6.
      <PERGUNTA Pontos="20">
```

```
        <ALINEA Pontos="10">Descreva música.</ALINEA> // 1.
      </PERGUNTA>
      <PERGUNTA Pontos="30">
        <ALINEA Pontos="20"/> // 2.
        <FIGURA Pontos="10"/> // 2. 3.
      </PERGUNTA>
    </EXAME>
  </DOC-EXAME>
```

Resposta:

1/ Alinea e Figura têm o texto dentro do elemento <TEXTO>
2/ Alinea e Figura não podem ser EMPTY
3/ Figura não tem o atributo Pontos
4/ O atributo de Figura é Formato
5/ Atributo Cod de Exame não pode conter duplicados porque é ID
6/ AW não é ID (a disciplina é D3)

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 o texto das alíneas dos exames constantes de documentos XML de acordo com este DTD.

Resposta:

```
<?xml version="1.0" encoding="utl-8"?>
<stylesheet xmlns:xsl="htpp://www.w3.org/XSL/TRANSFORM/1.0">
<xsl:template match="/">
    <HTML>
    <BODY>
    <xsl:for-each select="Exame">
        <xsl:value-of select="@Date"/>
        <P>
        <xsl:for-each select="//Alinea">
          <xsl:apply-templates select="Texto"/>
          <BR>
        </xsl:for-each>
    </xsl:for-each>
    </BODY>
    </HTML>
</xsl:template>
</stylesheet>
```

10. Gestão de Transacções [10 pontos]

Considere o seguinte esquema de relações em BCNF para guardar os docentes e o departamento a que pertencem:

```
Docentes(codigo, nome, salario, idade, departamento)
Departamentos(codigo, local, orcamento)
```

e a seguinte modificação em SQL:

```
UPDATE Docentes
SET salario= 1,1*salario
WHERE nome="jlopes";
```

a) Dê um exemplo de uma interrogação SQL que entre em conflito (do ponto de vista da concorrência) com a modificação apresentada, se for corrida ao mesmo tempo. Explique

o que pode correr mal e como o problema poderia ser resolvido através do uso de bloqueios (*locks*).

Resposta:

Por exemplo a interrogação:

```
SELECT d.codigo
FROM Docentes d
WHERE d.salario=10000;
```

Considerando dois tuplos com salário 10000.

Se a modificação correr primeiro, nenhum deles é seleccionado; se correrem ao mesmo tempo, um tuplo pode ser modificado e outro seleccionado. A utilização de bloqueios assegura que ou são seleccionados primeiro ou são modificados primeiro.

[No caso geral, bloqueios não são suficientes para garantir serialidade.]

FIM.