

Integração de Dados

Duração: 90 minutos

Licenciatura em Engenharia Informática: 2º ano - 2º semestre

2016/2017

TESTE N° 1

Responda às questões deste teste no espaço reservado. Use o ficheiro XML fornecido em anexo.

Nome: _____ N° aluno: _____

1. [20%] Sistemas de Integração de Dados

Analise a situação descrita de seguida e responda às questões colocadas.

A editora de livros **TodosALer** expandiu o seu negócio adquirindo uma livraria de vendas exclusivas on-line **BooksforAll**. No processo de fusão surgiu a necessidade de integração dos dados das bases de dados das duas empresas. A editora **TodosALer** tem três bases de dados principais:

- LivrosPublicados(isbn, titulo, autorId, preco, direitosAutor)
- Escritores(autorId, nome, contacto, data_nasc)
- Clientes(ClienteId, nome, morada, contacto)

A livraria **BooksforAll** tem duas bases de dados principais:

- Livros (isbn, titulo, Nomeautor, preçoVenda, desconto)
- Clientes(ClienteId, nome, apelido, contacto, data_nasc)

No processo de junção tornou-se fundamental fazer pesquisas sobre os clientes e sobre os livros sem obter dados redundantes ou inconsistentes, uma vez que as bases de dados originais possuem informação repetida sobre livros e clientes.

- Pesquisa 1: dado o id de cliente, quais os dados (nome, morada, telefone) desse cliente
- Pesquisa 2: dado o isbn de um livro qual o título, o nome do autor e o preço desse livro

Identifique e detalhe os componentes de um possível sistema de integração de dados capaz de responder às pesquisas anteriores.

Componentes de um sistema de Integração de dados I=<S,G,M>

S: 5 fontes de dados

S1: LivrosPublicados

S2: Escritores

S3: Clientes

S4: Livros

S5: Cliente

Para as pesquisas solicitadas o ideal seria um modelo global G com duas tabelas

Clientes (junção das duas tabelas de clientes S3 e S5) permite responder à pesquisa 1

Livros (junção das tabelas LivrosPublicados - S1 e Escritores – S2 e Livros - S4)

Para responder às pesquisas, teriam de se incluir os seguintes atributos nas tabelas de G:

Clientes(id, nome, morada, telefone)

Livros(isbn, titulo, autor, preco)

Mapeamentos

Atributos de G

id
nome
morada
telefone

isbn
titulo
autor
preco

Atributos originais (Sources)

clienteID de S3 e S5
nome de S3, nome e apelido de S5
morada de S3, não tem mapeamento em S5
contacto de S3 e S5

isbn de S1 e S4
titulo de S1 e S4
nome de S2 (obtido usando o autorID) e nomeAutor de S4
preco de S1 e precoVenda de S4

2. [20%] Expressões Regulares

- a) Assinale as cadeias encontradas pelas seguintes expressões regulares (respostas incorrectas serão penalizadas)

\b(ab)+[ab]+\b

A -- abab
B -- abbb
C -- aabbab
D -- ab
E -- babbb
F -- bbaa
G -- abaaaa

\b[ab][acd]?(^ab)+\b

A -- andre
B -- alic
C -- balde
D -- acida
E -- bonita
F -- bolos
G -- adoro

\b[ana]*[bela]?(^top)\b

A -- ana
B -- bela
C -- top
D -- alo
E -- anel
F -- annaaaa
G -- bis

- b) Escreva uma expressão regular que encontre **palavras** com caracteres alfabéticos. As palavras devem ser começadas e terminadas pelas vogais **a** ou **u** (**maiúscula ou minúscula**) e com tamanho entre 2 e 7 caracteres. Alguns exemplos: **ursa, Alma, Unida, ALuA, UIVA, uva,**

\b[aAuU][a-zA-Z]{0,5}[aAuU]\b

- a) Escreva uma expressão regular que encontre **frases** interrogativas de tamanho entre 10 e 20 caracteres. Nas frases assuma que começam por qualquer carácter maiúsculo, seguidos de caracteres minúsculos, espaçamentos ou dígitos e terminando com o sinal de interrogação.

^[A-Z][a-zA-Z\s0-9]{8,18}\?\$

- b) Escreva uma expressão regular que encontre as **palavras** que comecem por **ga, go, gi** ou **gr** seguidas de qualquer carácter minúsculo.

Na frase seguinte, as palavras a sublinhado mostram exemplos do que a ER deve validar.

o gato preto e grande gosta de gaivotas brancas gigantes

\bg[aoir][a-z]+\b

- c) Analise o seguinte ficheiro de texto que armazena horas de entrada num servidor. O primeiro campo é um identificador, o segundo campo é o dia e o mês de acesso e o terceiro campo a hora.

```
log01 - 12/01 - 09:00
log02 - 05/01 - 06:10
log02 - 05/01 - 06:66
log03 - 07/01 - 10:00
log99 - 19/01 - 08:57
log123 - 21/01 - 09:01
log1099 - 30/01 - 08:32
log012 - 04/02 - 07:45
```

Recorrendo a grupos, construa uma expressão regular que encontre os dias do mês de Janeiro em que os logs foram efectuados antes das 9:00. A ER regular deve encontrar apenas horas válidas, No ficheiro anterior há duas horas inválidas que não devem ser consideradas (06:66 e 01:99).

log[0-9]+ - ([0-9]{2})/01 - 0[0-8]:[0-5][0-9]

a porção assinalada a vermelho é o grupo que permite retirar os dias do mês de Janeiro em que os logs foram feitos antes das 9.00

3. [15%] JSON/XML

Analise o XML da esquerda e proponha uma representação no formato JSON para guardar a mesma informação.

```
<loja>
  <produto id="p001">
    <nome>Bolachas</nome>
    <preco>12</preco>
    <stock status="ok">25</stock>
  </produto>
  <produto id="p002">
    <nome>Cereais</nome>
    <preco>2.5</preco>
    <stock status="nulo">0</stock>
  </produto>
  ....
</loja>
```

```
{
  "loja": {
    "produto": [
      {
        "@id": "p001",
        "nome": "Bolachas",
        "preco": "12",
        "stock": {
          "@status": "ok",
          "#text": "25"
        }
      },
      {
        "@id": "p002",
        "nome": "Cereais",
        "preco": "2.5",
        "stock": {
          "@status": "nulo",
          "#text": "0"
        }
      }
    ]
  }
}
```

4. XPath [25%]

a) Analise o documento XML fornecido em anexo. Escreva as expressões XPath que permitam encontrar a seguinte informação:

- i. Nomes (elemento **nome**) dos produtos de mercearia

```
//mercearia/produto/nome
```

- ii. Lista de produtos de toda a loja com stocks abaixo de 10 unidades e sem desconto.

```
//produto[stock/@quant < 10 and preco/@desconto="não"]
```

- iii. Qual o **tipo** do produto dos frescos mais caro?

```
//frescos/produto[preco=max(//frescos/produto/preco)]/@tipo
```

- iv. Quantos produtos de **papelaria** são da marca Ambar?

```
count(//papelaria/produto[marca="Ambar"])
```

- v. Lista de produtos dos **frescos** que se seguem ao “Leite de Soja”

```
//frescos/produto[nome="Leite de soja"]/following-sibling::produto
```

vi. Quantas marcas de **cadernos** existem no ficheiro?

```
count(//papelaria/produto[contains(nome,"Caderno")]/marca)
```

vii. Listagem dos **nomes** (texto) de todos os tipos de Leite do ficheiro

```
//frescos/produto[contains(nome,"Leite")]/nome/text()
```

viii. Qual a **média** de preços dos produtos de mercearia?

```
avg(//mercearia/produto/preco)
```

ix. **Nome** e **preço** do penúltimo produto dos frescos do ficheiro.

```
//frescos/produto[last()-1]/nome | //frescos/produto[last()-1]/preco
```

x. Elementos **preço** e **validade** (texto) do último produto da lista de frescos do ficheiro. A resposta deve fazer utilização obrigatória do eixo de navegação **following-sibling**.

```
//frescos/produto[last()]/stock/following-sibling::*[1]/text()
```

b) Indique o que devolvem as seguintes pesquisas XPath. Transcreva do ficheiro a informação que é devolvida.

i. `//produto[contains(@tipo,"cereais")]/nome`

```
<nome>Cereais</nome>  
<nome>Cereais Muesli</nome>
```

ii. `/loja/frescos/following-sibling::papelaria/produto[last()-2]/following::marca`

```
<marca>Note</marca>  
<marca>Ferris</marca>
```

iii. `/loja//produto[last()]/preceding-sibling::*[1]/nome`

```
<nome>Cereais Muesli</nome>  
<nome>Alface</nome>  
<nome>Caderno A3 pautado</nome>
```

5. [10%] DTD

a) Escreva o DTD que valide o elemento **mercearia**

```
<!ELEMENT mercearia ( produto+ )>
```

b) Escreva o DTD que valide o elemento **desc**.

```
<!ELEMENT desc ( #PCDATA | nome | contacto )*>
```

c) Escreva o DTD que valide o elemento **loja**.

```
<!ELEMENT loja ( desc , mercearia , frescos , papelaria )>
```

d) Escreva o DTD que valide o elemento **validade**.

```
<!ELEMENT validade ( #PCDATA )>
```

e) Escreva o DTD que valide o elemento **stock**.

```
<!ELEMENT stock EMPTY>
```

f) Escreva o DTD que valide o atributo **desconto** uma enumeração de 2 valores (sim, não) e opcional.

```
<!ATTLIST preco desconto (sim|não) #IMPLIED>
```

g) Escreva o DTD que valide o atributo **quant** (obrigatório).

```
<!ATTLIST stock quant CDATA #REQUIRED>
```

6. [10%] XSD

Escreva apenas o XSD solicitado. Use a instrução **ref** para referenciar os elementos e atributos necessários, assumindo que se encontram definidos.

a) Escreva o XSD que valide o elemento **validade** como data

```
<xsd:element name="validade" type="xsd:date"/>
```

b) Escreva o XSD que valide o atributo **quant** como inteiro

```
<xsd:attribute name="quant" type="xsd:integer"/>
```

c) Escreva o XSD que valide o elemento **frescos**

```
<xsd:element name="frescos">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="produto" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

d) Escreva o XSD que valide o elemento **preco**

```
<xsd:element name="preco">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:double">
        <xsd:attribute ref="desconto" use="optional"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

e) Escreva o XSD que valide o atributo **desconto** uma enumeração de 2 valores (sim, não).

```
<xsd:attribute name="desconto">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="sim"/>
      <xsd:enumeration value="não"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
```

f) Escreva um **tipo de dados** XSD que permita validar os elementos **mercearia**, **frescos** e **papelaria**

```
<xsd:complexType name="tipoProd">
  <xsd:sequence>
    <xsd:element ref="produto" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute ref="actualizado" use="optional"/>
  <xsd:attribute ref="local" use="optional"/>
</xsd:complexType>
```

g) Usando o tipo de dados anterior, escreva o XSD para validar os três elementos **mercearia**, **frescos** e **papelaria**

```
<xsd:element name="mercearia" type="tipoProd"/>
<xsd:element name="frescos" type="tipoProd"/>
<xsd:element name="papelaria" type="tipoProd"/>
```