

DI-FCT-NOVA

11 de maio de 2016

Bases de Dados

3º teste, 2015/16

Duração: 2 horas (consulta limitada)

Mantendo o tema dos testes anteriores, com o qual já deve estar acostumado, considere uma base de dados relacional para armazenar informação sobre entradas, saídas e acolhimento de refugiados em espaço europeu, com o seguinte esquema (onde os atributos sublinhados são os que constituem a chave primária de cada uma das relações):

refugiados(passRef,paisRef,nome,dataNasc)
postosAcolhimento(postoAcolh,pais,capacidade) postosFronteira(poF,pais)
entradas(numE,poF,passRef,dataE) saidas(numS,poF,passRef,dataS)
acolhimentos(passRef,postoAcolh,dataEntr,dataSai)

Relembrando do teste anterior, a base de dados tem: uma tabela de refugiados, com informação do seu passaporte, país de origem, nome e data de nascimento – desta vez, para simplificar o problema **consideraremos que o nº do passaporte é chave na tabela de refugiados**; uma tabela com todos os postos de acolhimento de refugiados, com nome (em postoAcolh), país onde está localizado, e quantos refugiados pode acolher (em capacidade); uma com os postos de fronteira. Para registar as entradas e saídas de refugiados no espaço europeu, há uma tabela de entradas e uma de saídas. Cada uma delas tem um nº, o nome do posto de fronteira onde foi feita (em poF), a informação do (passaporte do) refugiado que entrou ou saiu, respetivamente, e a data em que foi feita. Há ainda uma tabela (acolhimentos) que regista que refugiados foram acolhidos em que postos e entre que datas.

Para além disso, esta base de dados tem ainda uma tabela que guarda informação sobre os progenitores de cada um dos refugiados. Essa tabela tem o seguinte esquema:

progenitor(passFilho,passProg)

Um tuplo nesta tabela significa que um dos progenitores (pai ou mãe) do refugiado com passaporte passFilho é o refugiado com passaporte passProg.

Grupo 1 [Datalog]

1. Apresente um conjunto de regras em Datalog que permita obter resposta às seguintes perguntas:
 - a) **[1,5 valores]** Quais os nomes dos refugiados Sírios (i.e. com a Síria como país de origem)?
 - b) **[1]** Quais os nomes dos refugiados que entraram em espaço europeu, mas que nunca foram acolhidos em nenhum posto de acolhimento?
 - c) **[1]** Quais os antepassados (pai, mãe, avós, bisavós, etc) de cada um dos refugiados?
A resposta deverá ter um tuplo (Pass,PassAnt) sempre que um refugiado com passaporte Pass tenha como um dos seus antepassados o refugiado com passaporte PassAnt.

Grupo 2 [Transações]

1. Considere o seguinte escalonamento de operações em duas transações concorrentes:

update postosAcolhimento
set capacidade = capacidade - 50
where postoAcolh = 'A';

update postosAcolhimento
set capacidade = capacidade + 50
where postoAcolh = 'B';

update postosAcolhimento
set capacidade = capacidade * 1,1
where postoAcolh = 'A';

update postosAcolhimento
set capacidade = capacidade * 1,1
where postoAcolh = 'B'

Ou, se preferir com a notação com *reads* e *writes* usada nos slides das aulas, considere o escalonamento:

```
read(CapA)
CapA := CapA - 50
write(CapA)
```

```
read(CapA)
CapA := CapA * 1.1
write(CapA)
```

```
read(CapB)
CapB := CapB + 50
write(CapB)
```

```
read(CapB)
CapB := CapB * 1.1
write(CapB)
```

- a) [1,5] Este escalonamento é serializável de conflito?
Se sim, apresente um escalonamento serializado equivalente. Se não, diga quais as operações em conflito que causam a não seriabilidade.
2. Considere as seguintes transações concorrentes (não assuma, para já, nenhuma ordem entre as operações de uma e da outra transação – i.e. não se sabe se o **select** da transação 2 é executado antes do 1º **update** da transação 1, ou entre o 1º e o 2º **update**, ou depois do 2º **update**):

Transação 1	Transação 2
update postosAcolhimento set capacidade = capacidade - 50 where postoAcolh = 'A'; update postosAcolhimento set capacidade = capacidade + 50 where postoAcolh = 'B';	select sum(capacidade) from postosAcolhimento where postoAcolh in ('A','B');

- a) [1,5] Sabendo que antes da execução destas transações a capacidade de cada um dos postos de acolhimento A e B é de 100, que não há mais transações a ser executadas na base de dados para além destas duas, e que ambas são executadas com isolation mode serializable, diga qual (ou quais) os possíveis resultados devolvidos pela operação de **select** da transação 2.
- b) [1,5] Suponha agora que as transações são ambas executadas em isolation mode read uncommitted (i.e. sem nenhum tipo de isolamento). Apresente um escalonamento das 3 operações envolvidas nas duas transações, em que o resultado do **select** seja diferente daquele que se obtinha no modo serializable, e diga qual esse resultado.

Grupo 3 [Objeto-Relacional]

Chegou-se à conclusão que a maior parte dos refugiados só têm uma entrada. E mesmo aqueles que têm mais do que uma, não têm muitas.

Como tal, usando as *features* objeto-relacional do SGBD optou-se (talvez irrefletidamente, mas enfim...) por substituir a tabela de refugiados por uma que para além do passaporte, país, nome e data de nascimento, tem também um atributo entradas com o conjunto das entradas desse refugiado, em que cada entrada nesse conjunto tem a data em que foi feita, e uma referência para o posto de fronteira onde ocorreu.

- [2] Apresente o código SQL completo de criação desta nova tabela de refugiados.
- [2] Considerando a base de dados com esta tabela em vez da original de refugiados e de entradas, apresente o código SQL para a pergunta:

“Quais os países dos postos de fronteira por onde já deu entrada o refugiado com passaporte 123456-A da Síria?”

Grupo 4 [XML]

Para transferir dados das várias entradas de refugiados em cada posto de fronteira, decidiu-se usar ficheiros XML de acordo com o seguinte DTD:

```
<!DOCTYPE refugiados[
  <!Element refugiados(refugiado*,entrada*)>
    <!ATTLIST refugiados nomePostoFronteira #REQUIRED>
  <!Element refugiado(nome,pais,idade)>
    <!ATTLIST refugiado pass ID #REQUIRED>
  <!Element entrada(data)>
    <!ATTLIST entrada refugiado IDREF #REQUIRED>
  <!Element nome(#PCDATA)>
  <!Element pais(#PCDATA)>
  <!Element idade(#PCDATA)>
  <!Element data(#PCDATA)>
]>
```

1. [2] Apresente o texto dum documento XML que use este DTD, e que contenha informação sobre as seguintes entradas de refugiados no posto de fronteira de *Lesbos*:
 - O refugiado Sírio, *Adnan*, com passaporte 12345-A, de 25 anos, entrou no dia 7 de maio de 2016;
 - A refugiada Síria, *Bana*, com passaporte 23456-B, de 32 anos, entrou no dia 8 de Maio de 2016.
2. Apresente expressões XPATH sobre um ficheiro de acordo com o DTD acima que devolvam os resultados das seguintes perguntas:
 - a) [1] Em que datas entraram refugiados?
 - b) [1] Quais os nomes dos refugiados Sírios com menos de 6 anos?
 - c) [1] Quais os nomes dos refugiados que entraram no dia 11 de maio de 2016?
3. [1,5] Após usar ficheiros XML de acordo com o DTD acima para transferir os dados, constatou-se que talvez não fosse o melhor formato. Optou-se então por usar ficheiros XML organizados da seguinte forma:
 - O ficheiro continua a ter a identificação do posto de acolhimento;
 - Depois disso vem um conjunto de elementos referentes a dias onde entraram refugiados, cada um com a data e os vários refugiados que entraram nessa data;
 - Cada elemento de um refugiado entrado numa data tem informação do seu passaporte, nome, país e idade.

Proponha um DTD para ficheiros XML com esta estrutura.

4. [1,5] O que faz a seguinte pergunta XQuery, quando aplicada a um ficheiro XML de acordo com o DTD apresentado no início deste grupo? Exemplifique, e.g. mostrando o seu resultado no ficheiro XML da pergunta 1 deste grupo.

```
for    $r in /refugiados/refugiado
      $e in /refugiados/entrada[@refugiado=$r/@pass]
return <entr> {$r/nome $e/data} </entr>
```