

DI-FCT-NOVA

17 de maio de 2017

# Bases de Dados

## 3º teste, 2016/17

### Uma resolução

#### Grupo 1 [Datalog]

1.

- a)  $\text{paisLisboa}(\text{Nome}) :- \text{países}(\_, \text{Nome}, \text{'Lisboa'})$ .
- b)  $\text{empTrum}(\text{Nome}) :- \text{doadores}(\text{E}, \text{Nome}, \_), \text{empresas}(\text{E}, \_), \text{doações}(\_, \text{E}, \text{'Trump2016'}, \_)$ .  
 $\text{empTrum}(\text{Nome}) :- \text{doadores}(\text{P}, \text{Nome}, \_), \text{empresas}(\text{E}, \_),$   
 $\text{emprega}(\text{E}, \text{P}), \text{doações}(\_, \text{P}, \text{'Trump2016'}, \_)$ .
- c)  $\text{nãoTrump}(\text{Nome}) :- \text{doadores}(\text{E}, \text{Nome}, \_), \text{empresas}(\text{E}, \_), \text{not doouTrump}(\text{Emp})$ .  
 $\text{doouTrump}(\text{Emp}) :- \text{doações}(\_, \text{Emp}, \text{'Trump2016'}, \_)$ .

#### Grupo 2 [Transações]

1.

- a) O escalonamento é serializável. O único escalonamento serializado equivalente corresponde a executar primeiro toda a T2, depois a T3 e, finalmente a T1. Ou seja:

| T1       | T2       | T3      |
|----------|----------|---------|
|          | read(X)  |         |
|          | write(Z) |         |
|          |          | read(Y) |
|          |          | read(Z) |
| read(X)  |          |         |
| write(X) |          |         |
| write(Y) |          |         |

- 2. Considere-se a seguinte identificação para cada operação:

| Transação 1                                                                                                                                                                    | Transação 2                                                                                                                                                                    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>A1. begin transaction</b><br><b>A2. select * from países;</b><br><b>A3. insert into países values ('PT',...);</b><br><b>A4. select * from países;</b><br><b>A5. commit;</b> | <b>B1. begin transaction</b><br><b>B2. select * from países;</b><br><b>B3. insert into países values ('SP',...);</b><br><b>B4. select * from países;</b><br><b>B5. commit;</b> |

- a) Considere o escalonamento com a seguinte ordem das operações: A1; B1; A2; B2; A3; B3; A4; B4; A5; B5. Neste caso, em modo **read uncommitted** a operação A4 devolveria os dois tuplos (introduzidos em A3 e em B3), enquanto que em modo **read committed** devolveria apenas o tuplo introduzido em A3.
- b) Considere agora o escalonamento com a seguinte ordem das operações: A1; B1; A2; B2; A3; B3; A4; A5; B4; B5. Neste caso, em modo **read committed** a operação B4 devolveria os dois tuplos (introduzidos em A3 e em B3), enquanto que em modo **serializable** devolveria apenas o tuplo introduzido em B3.

### Grupo 3 [Objeto-Relacional]

1. **create type** dType **as** (idDoador **number**, nome **varchar**(10), pais **varchar**(4)) **not final**;  
**create type** eType **under** dType **as** (capitalSocial **number**) **final**;  
**create type** pType **under** dType **as** (dataNasc **date**) **final**;  
**create table** doadores **of** dType;  
**create table** empresas **of** eType **under** doadores;  
**create table** pessoas **of** pType **under** doadores;
2. **select** candidatos.nome  
**from** doadores A, **unnest**(A.doações) **as** B, candidatos  
**where** A.nome = 'Gazzpron' and B.candidatura -> pres = candidatos.numCand;

### Grupo 4 [XML]

1. 

```
<?xml version="1.0" encoding="UTF-8"?>
<doacoes informador = "MI6">
  <doador nome = "Gazzpron">
    <pais>Russia</pais>
    <doacao candidatura = "Trump2016" valor = "200000"><data/></doacao>
  </doador>
  <doador nome = "Orange">
    <pais>USA</pais>
    <doacao candidatura = "HillaryKaine" valor = "200000"><data/></doacao>
    <doacao candidatura = "Trump2016" valor = "1">
      <data>7/11/2016</data>
    </doacao>
  </doador>
</doacoes>
```
2.
  - a) //pais
  - b) //doacao[@valor > 100000]/data
  - c) //doacao[@candidatura="Trump2016"]/../@nome
3. A pergunta devolve um elemento por cada doação no ficheiro, com os seguintes subelementos: o doador dessa doação, a candidatura a que foi feita a doação, e o valor doado. Por exemplo, para o ficheiro XML da pergunta 1 devolveria:

```
<doacao>
  <doador>Gazzpron</doador> <cand>Trump2016</cand> <valor>200000</valor>
</doacao>
<doacao>
  <doador>Orange</doador> <cand>HillaryKaine</cand> <valor>200000</valor>
</doacao>
<doacao>
  <doador> Orange </doador> <cand>Trump2016</cand> <valor>1</valor>
</doacao>
```

4. 

```
<!DOCTYPE doacoes[
  <!Element doacoes(doador*, doacao*)>
    <!ATTLIST doacoes informador #REQUIRED>
  <!Element doador(pais*)>
    <!ATTLIST doador nome ID #REQUIRED>
  <!Element doacao(valor,data)>
    <!ATTLIST doacao doador IDREF #REQUIRED>
  <!Element pais(#PCDATA)>
  <!Element valor(#PCDATA)>
  <!Element data(#PCDATA)>
]>
```