DI-FCT-NOVA **30 de abril de 2016**

Bases de Dados

2º teste, 2015/16

Duração: 1,5 horas (sem consulta)

Considere uma base de dados relacional para armazenar informação sobre entradas, saídas e acolhimento de refugiados em espaço europeu, criada pelos seguintes comandos SQL:

```
create table refugiados(
    passRef varchar(20),
    paisRef varchar(20),
    nome varchar(100) not null,
    dataNasc date,
    primary key (passRef ,paisRef),
    foreign key (paisRef) references paises
);
```

```
create table paises(
    pais varchar(20) primary key,
    capital varchar(40),
    zona char(2) not null,
    check (zona in ('EU','NO'))
);
```

```
create table postosAcolhimento(
    postoAcolh varchar(20) primary key,
    pais varchar(20) not null,
    capacidade number,
    foreign key (pais) references paises
);
```

```
create table postosFronteira(
    poF varchar(20) primary key,
    pais varchar(20) not null,
    foreign key (pais) references paises
);
```

```
create table entradas(
    numE number,
    poF varchar(20),
    passRef varchar(20),
    paisRef varchar(20),
    dataE date not null,
    primary key (numE, poF, passRef, paisRef ),
    foreign key (poF) references postosFronteira,
    foreign key (passRef,paisRef)
                references refugiados
);
```

```
create table saidas(
    numS number,
    poF varchar(20),
    passRef varchar(20),
    paisRef varchar(20),
    dataS date not null,
    primary key (numS, poF, passRef, paisRef ),
    foreign key (poF) references postosFronteira,
    foreign key (passRef,paisRef),
        references refugiados
);
```

```
create table acolhimentos(
    passRef varchar(20),
    paisRef varchar(20),
    postoAcolh varchar(20),
    dataEntr date not null,
    dataSai date,
    primary key (passRef, paisRef, postoAcolh, dataEntr),
    foreign key (postoAcolh) references postosAcolhimento,
    foreign key (passRef,paisRef) references refugiados
);
```

A base de dados tem: uma tabela de refugiados, com informação do seu passaporte, país de origem, nome e data de nascimento; uma tabela de países com nome e capital do país, e indicação (em zona) se é um país Europeu ou NãO-europeu; uma outra com todos os postos de acolhimento de refugiados, com nome (em postoAcolh), país onde está localizado, e quantos refugiados pode acolher em condições normais (em acolhimento); uma com os postos de fronteira. Para registar as entradas e saídas de refugiados no espaço europeu, tem 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 refugiado que entrou ou saiu, respetivamente (com indicação do seu passaporte e país de origem) e a data em que foi feita. Há ainda uma tabela (acolhimentos) que regista que refugiados foram acolhidos em que postos, desde que data de entrada no posto até que data de saída; a data de saída deve estar a **null** caso o refugiado ainda se encontre nesse posto.

Grupo 1

1. Apresente uma expressão em **álgebra relacional e uma consulta SQL** para cada uma das perguntas:

a) **[2 valores (1 álgebra+1 SQL)]** Quais as capitais dos países de origem dos refugiados que entraram alguma vez por um posto de fronteira situado na Grécia?

b) **[3]** Quais os países europeus que nunca acolheram nenhum refugiado Sírio?

c) **[3]** Quais os nome e países de origem dos refugiados que foram imediatamente expulsos do espaço europeu (i.e. que entraram e saíram num mesmo dia, num mesmo posto de fronteira)?

2. Apresente **uma consulta SQL** para cada uma das perguntas:

a) **[1.5]** Quais os pares de postos de acolhimento com exatamente a mesma capacidade?

b) **[1.5]** Por cada país de origem, quantos refugiados há que entraram no espaço europeu e que ainda não saíram?

c) **[1.5]** Quais os postos de acolhimento que estão superlotados (i.e. que têm acolhidos mais refugiados do que a sua capacidade)?

3. **[1.5]** Considere a seguinte *view* e as duas consultas SQL (onde a única diferença entre elas está sublinhada)

```
create view nRefSirios as
   (     select pais, count(*) as n
         from acolhimentos natural inner join postosAcolh
         where paisRef = 'Siria' and dataSai is null
         group by pais);
```

```
select paises.pais, n
from paises inner join nRefSirios
     on (paises.pais = nRefSirios.pais);
```

```
select paises.pais, n
from paises left outer join nRefSirios
     on (paises.pais = nRefSirios.pais);
```

Diga qual é, intuitivamente, o resultado de cada uma das consultas, focando-se na diferença entre os resultados de uma e da outra.

Grupo 2

1. Como certamente terá reparado, o esquema da base de dados tem um problema de modelação. Nomeadamente, não garante que os refugiados têm que ser de um país de origem fora da europa, e que os postos de fronteira e de acolhimento são em países europeus. Nesta pergunta vamos resolver esse problema!

a) **[1]** Comece por criar uma tabela onde só se armazena a informação dos países europeus, e outra onde só se armazena a informação dos países não europeus.

b) **[1]** Insira agora os dados nessas tabelas, com base nos que estão na tabela original de países.

c) **[2]** Finalmente, imponha restrições de integridade sobre a nova base de dados, por forma a garantir aquilo que faltava (i.e. que os refugiados têm que ser de um país de origem fora da europa, que os postos de fronteira e de acolhimento são em países europeus e, já agora, que um país não pode ser simultaneamente europeu e não europeu).

2. **[2]** Decidiu-se que deviam ser emitidos alertas sempre que um posto de acolhimento excedesse a sua capacidade. Para tal foi criada a seguinte tabela:

```
create table alertas(postoAcolh varchar(20), data date);
```

Sempre que um posto de acolhimento exceder a sua capacidade (i.e. passar a ter mais refugiados do que a sua capacidade) deve ser inserido um tuplo nessa tabela com o nome desse posto e a data em que foi excedida. Para não estarem sempre a disparar alarmes, quando um posto de acolhimento já tem a sua capacidade excedida, a chegada de um novo refugiado não deve dar origem a nenhuma ação (de inserção na tabela para posterior alerta).
Crie os mecanismos necessários na base de dados para que a inserção de tuplos nesta tabela seja feita de forma automática. *(Pode, se preferir, usar a linguagem PL/SQL.)*