



PROJETO DE BASES DE DADOS – PARTE 3

Grupo3 - Quarta Feira 16:30 (Prof. Carlota Dias)

<i>Número</i>	<i>Nome</i>	<i>Porcentagem</i>	<i>Esforço</i>
93694	Carolina Ramos	33%	12 horas
90118	João Diegues	33%	12 horas
93740	Miguel Oliveira	33%	12 horas

Schema

```
drop table prescricao_venda cascade;
```

```
drop table venda_farmacia cascade;
```

```
drop table analise cascade;
```

```
drop table prescricao cascade;
```

```
drop table consulta cascade;
```

```
drop table medico cascade;
```

```
drop table instituicao cascade;
```

```
drop table concelho cascade;
```

```
drop table regioao cascade;
```

```
create table regioao(  
  num_regiao serial not null unique,  
  nome char(100) not null,  
  num_habitantes serial,  
  constraint pk_regiao primary key(num_regiao)  
);
```

```
create table concelho(  
  num_concelho serial not null,  
  num_regiao serial not null,  
  nome char(100) not null,  
  num_habitantes serial,  
  constraint pk_concelho primary key(num_concelho, num_regiao),  
  constraint fk_concelho_regiao foreign key(num_regiao) references regioao(num_regiao)  
);
```

```
create table instituicao(  
  nome char(100) not null unique,  
  tipo char(100) not null,  
  num_regiao serial not null,  
  num_concelho serial not null,  
  constraint pk_instituicao primary key(nome)  
);
```

```
create table medico(
    num_cedula serial not null unique,
    nome char(100) not null,
    especialidade char(50) not null,
    constraint pk_medico primary key(num_cedula)
);
```

```
create table consulta(
    num_cedula serial not null,
    num_doente serial not null,
    data date not null,
    nome_instituicao char(100) not null,
    constraint pk_consulta primary key(num_cedula, num_doente, data),
    constraint fk_consulta_instituicao foreign key(nome_instituicao) references instituicao(nome),
    constraint fk_consulta_medico foreign key(num_cedula) references medico(num_cedula),
    constraint sunday check(extract(dow from data) > 0),
    constraint saturday check(extract(dow from data) < 6)
);
```

```
create table prescricao(
    num_cedula serial not null,
    num_doente serial not null,
    data date not null,
    substancia char(50) not null,
    quant serial not null,
    constraint pk_prescricao primary key(num_cedula, num_doente, data, substancia),
    constraint fk_prescricao_consulta foreign key(num_cedula, num_doente, data) references consulta(num_cedula, num_doente, data)
);
```

```
create table analise(
    num_analise serial not null unique,
    especialidade char(50) not null,
    num_cedula serial not null,
    num_doente serial not null,
```

```

data date not null,
data_registo date not null,
nome char(100),
quant serial not null,
inst char(100),
constraint pk_analise primary key(num_analise),
constraint fk_analise_consulta foreign key(num_cedula, num_doente, data) references
consulta(num_cedula, num_doente, data),
constraint fk_analise_instituicao foreign key(inst) references instituicao(nome)
);

```

```

create table venda_farmacia(
num_venda serial not null unique,
data_registo date not null,
substancia char(50) not null,
quant serial not null,
preco serial not null,
inst char(100) not null,
constraint pk_farmacia primary key(num_venda),
constraint fk_venda_farmacia_instituicao foreign key(inst) references instituicao(nome)
);

```

```

create table prescricao_venda(
num_cedula serial not null,
num_doente serial not null,
data date not null,
substancia char(50) not null,
num_venda serial not null,
constraint pk_prescricao_venda primary key(num_cedula, num_doente, data, substancia, num_venda),
constraint fk_prescricao_venda_prescricao foreign key(num_cedula, num_doente, data, substancia)
references prescricao(num_cedula, num_doente, data, substancia),
constraint fk_prescricao_venda_venda_farmacia foreign key(num_venda) references
venda_farmacia(num_venda)
);

```

SQL

1-

Para conseguir exemplos mais significativos com o nosso ficheiro populate.sql submetido para este query a melhor data a usar é '2020-10-10'.

SELECT concelho

```
FROM (SELECT conc.nome AS concelho, vf.preco * vf.quant AS preco_total
      FROM(( instituicao AS i
            INNER JOIN venda_farmacia AS vf on vf.inst = i.nome)
            INNER JOIN concelho AS conc on i.num_concelho = conc.num_concelho and
            i.num_regiao = conc.num_regiao)
      WHERE vf.data_registo = CURRENT_DATE) AS nome GROUP BY concelho
HAVING SUM(nome.preco_total) >= all(
      SELECT SUM(total.preco_total)
      FROM(
            SELECT conc.nome AS concelho, vf.preco * vf.quant AS preco_total
            FROM(( instituicao AS i
                  INNER JOIN venda_farmacia AS vf on vf.inst = i.nome)
                  INNER JOIN concelho AS conc on i.num_concelho = conc.num_concelho and
                  i.num_regiao = conc.num_regiao)
            WHERE vf.data_registo = CURRENT_DATE) AS total GROUP BY concelho);
```

2-

SELECT medico, regiao

```
FROM( SELECT nome_data.nome AS medico, r.nome AS regiao
      FROM( SELECT nome, nome_instituicao
            FROM consulta
            NATURAL JOIN medico
            NATURAL JOIN prescricao
            WHERE data >= '2019-01-01' and data <= '2019-06-30') AS nome_data,
      instituicao AS i, regiao AS r
      WHERE nome_data.nome_instituicao = i.nome and i.num_regiao =
      r.num_regiao) AS tab GROUP BY medico, regiao
HAVING COUNT(medico) >= all (
      SELECT COUNT(medico)
      FROM( SELECT nome_data.nome AS medico, r.nome AS regiao
            FROM( SELECT nome, nome_instituicao
                  FROM consulta
                  NATURAL JOIN medico
```

```

NATURAL JOIN prescricao

WHERE data>= '2019-01-01' and data<='2019-06-30') AS nome_data, instituicao AS i, regio AS r

WHERE nome_data.nome_instituicao = i.nome and i.num_regiao = r.num_regiao) AS tab
GROUP BY medico, regio);

3-
SELECT m.nome
FROM medico AS m
WHERE NOT EXISTS(
    SELECT i.nome AS instituicao
    FROM (( instituicao AS i
            INNER JOIN concelho AS conc on i.num_concelho = conc.num_concelho and
i.num_regiao = conc.num_regiao))
    WHERE i.tipo ='farmacia' and conc.nome = 'Arouca'
EXCEPT
    (SELECT tab.instituicao
    FROM (SELECT vf.num_venda AS num_venda, vf.inst AS instituicao, conc.nome AS concelho
        FROM (( instituicao AS i
                INNER JOIN venda_farmacia AS vf on vf.inst = i.nome)
                INNER JOIN concelho AS conc on i.num_concelho =
conc.num_concelho and i.num_regiao = conc.num_regiao)
        WHERE conc.nome = 'Arouca' and EXTRACT(YEAR FROM vf.data_registo) =
EXTRACT(YEAR FROM CURRENT_DATE)) AS tab NATURAL JOIN prescricao_venda AS pV
        WHERE m.num_cedula = pV.num_cedula));

```

```

4-
SELECT DISTINCT num_doente
FROM (
    SELECT num_doente
    FROM analise
    WHERE EXTRACT(MONTH FROM data_registo) = EXTRACT(MONTH FROM CURRENT_DATE)) AS
doentes
WHERE num_doente NOT IN (
    SELECT num_doente
    FROM prescricao_venda
    WHERE EXTRACT(MONTH FROM data) = EXTRACT(MONTH FROM CURRENT_DATE));

```

Restrições de Integridade

Várias restrições de integridade além das definidas no `squema.sql` têm de ser feitas procedimentalmente e não estão definidas. A nossa maneira de obter isto é apenas introduzindo no nosso ficheiro `populate.sql` dados que estejam de acordo com essas restrições:

- RI-regiao-1: nome = {Norte, Centro, Lisboa, Alentejo, Algarve}
- RI-concelho-1: nome = {concelhos de portugal continental}
- RI-instituicao-1: tipo = {farmacia, laboratorio, clinica, hospital}
- RI-consulta-2: um doente não pode ter mais de uma consulta por dia na mesma instituição
- RI-analise: a consulta associada pode estar omissa; não estando, a especialidade da consulta tem de ser igual à do médico

Nas primeiras 3 restrições já estão definidas no ficheiro `populate.sql` todas as possibilidades para estas tabelas. Na restrição da análise para esta ser possível consideramos que quando a consulta está omissa os atributos da análise que pertencem à consulta devem ser nulos.