

## **Projeto de Bases de Dados – Parte 3**

Trabalho realizado por:

<b>Nome</b>	<b>Número</b>	<b>Contribuição (%)</b>	<b>Esforço (horas)</b>
Gonçalo Mateus	93713	33%	10
Guilherme Saraiva	93717	33%	10
Tomás Paiva	96961	33%	10

### **Grupo 37**

#### **Turno**

Quarta 15:00 – 16:30

#### **Docente**

André Pereira

## Comandos de criação da base de dados

```
drop table regiao cascade;
drop table concelho cascade;
drop table instituicao cascade;
drop table medico cascade;
drop table consulta cascade;
drop table prescricao cascade;
drop table analise cascade;
drop table venda_farmacia cascade;
drop table prescricao_venda cascade;
drop function obtem_consultas(doente int, data_consulta date, instituicao varchar);
drop function verifica_especialidade(n_medico int, doente int, data_consulta date,
especialidade_medico varchar);

create function obtem_consultas(doente int, data_consulta date, instituicao varchar(50))
returns int
    language plpgsql
as $$
begin

    if instituicao is null then
        return (
            select count(*)
            from consulta
            where consulta.num_doente = doente
            and consulta.data = data_consulta
        );
    end if;

    return (
        select count(*)
        from consulta
        where consulta.num_doente = doente
        and consulta.data = data_consulta
        and consulta.nome_instituicao = instituicao
    );

end;
$$;

create function verifica_especialidade(n_medico int, doente int, data_consulta date,
especialidade_medico varchar(50))
returns bool
    language plpgsql
as $$
begin
    if obtem_consultas(doente, data_consulta, null ) > 0 then
```

```

        return (select medico.especialidade
                from medico
                where medico.num_cedula = n_medico) = especialidade_medico;
else

        return true;

end if;
end;
$$;

create table regioao(
    num_regiao serial not null unique,
    nome varchar(50) not null,
    num_habitante int not null,
    check ( nome in ('Norte', 'Centro', 'Lisboa', 'Alentejo', 'Algarve') ),
    primary key(num_regiao)
);

create table concelho(
    num_concelho serial not null,
    num_regiao serial not null ,
    nome varchar(50) not null ,
    num_habitantes int not null ,
    check ( nome in ('Amarante', 'Arcos de Valdevez', 'Azurara', 'Barcelos', 'Braga', 'Cabeceiras de
Basto', 'Caldas das Taipas and Caldelas',
                    'Caminha', 'Carracedo', 'Chaves', 'Espinho', 'Esposende', 'Fao', 'Freixo de Espada a
Cinta', 'Gondomar', 'Guimaraes', 'Lamego',
                    'Marco de Canaveses', 'Melgaco', 'Mesao Frio', 'Minho', 'Miranda do Douro',
'Mirandela', 'Moncao', 'Montalegre', 'Murca',
                    'Pacos de Ferreira', 'Penafiel', 'Peso da Regua', 'Ponte da Barca', 'Ponte de Lima',
'Porto', 'Povoa de Varzim', 'Ribeira de Pena',
                    'Santa Maria de Bouro', 'Santo Tirso', 'S. Joao de Tarouca', 'Sernancelhe', 'Tras-os-
Montes', 'Valenca', 'Viana do Castelo', 'Vieira do Minho',
                    'Vila do Conde', 'Vila Flor', 'Vila Nova de Cerveira', 'Vila Nova de Gaia', 'Vila Nova de
Foz Coa', 'Vila Real', 'Vilar de Frades', 'Vizela',
                    'Agueda', 'Aguiar da Beira', 'Anadia', 'Almeida', 'Arganil', 'Arouca', 'Aveiro',
'Belmonte', 'Bucaco', 'Caramulo', 'Castelo Branco',
                    'Celorico da Beira', 'Coimbra', 'Conimbriga', 'Covilha', 'Curia', 'Figueira da Foz',
'Figueira de Castelo Rodrigo', 'Fundao', 'Gouveia',
                    'Guarda', 'Idanha-a-Nova', 'Idanha-a-Velha', 'Ilhavo', 'Linhares da Beira', 'Lorvao',
'Lourinha', 'Mangualde', 'Marialva', 'Mealhada', 'Minde',
                    'Mira', 'Monsanto', 'Monte Real', 'Montemor-o-Velho', 'Oliveira do Hospital',
'Penamacor', 'Piodao', 'Pombal', 'Porto de Mos', 'Proenca-a-Velha',
                    'S. Pedro do Sul', 'Seia', 'Serra da Estrela', 'Sortelha', 'Trancoso', 'Viseu', 'Vouzela',
'Abrantes', 'Almeirim', 'Azenhas do Mar', 'Batalha', 'Berlengas',
                    'Caldas da Rainha', 'Cartaxo', 'Cascais', 'Chamusca', 'Colares', 'Constancia', 'Ericeira',

```

```

'Estoril', 'Fatima', 'Leiria', 'Lisboa',
    'Macao', 'Mafra', 'Minde', 'Obidos', 'Palmela', 'Peniche', 'Porto de Mos', 'Queluz',
'Salvaterra de Magos', 'Santarem', 'Sardoal', 'Seixal', 'Serra da Arrabida',
    'Sesimbra', 'Setubal', 'Sintra', 'Tomar', 'Torres Novas', 'Torres Vedras', 'Vila Nova da
Barquinha', 'Vila de Rei', 'Vimeiro', 'Alandroal', 'Arraiolos', 'Borba',
    'Castelo de Vide', 'Elvas', 'Estremoz', 'Evora', 'Marvao', 'Monsaraz', 'Montemor-o-
Novo', 'Mourao', 'Portalegre', 'Portel', 'Redondo', 'Reguengos de Monsaraz',
    'Sousel', 'Vendas Novas', 'Viana do Alentejo', 'Vila Vicosa', 'Albufeira', 'Alcoutim',
'Aljezur', 'Almansil', 'Alte', 'Cacela', 'Carvoeiro', 'Castro Marim',
    'Estombar', 'Faro', 'Lagoa', 'Lagos', 'Loule', 'Moncarapacho', 'Monchique', 'Monte
Gordo', 'Olhao', 'Ponta da Piedade', 'Porches', 'Portimao', 'Sao Bras de Alportel',
    'Sao Bartolomeu de Messines', 'Sagres', 'Silves', 'Vila do Bispo', 'Vilamoura', 'Vila
Real de Santo Antonio', 'Tavira') ),
    foreign key(num_regiao) references regiao(num_regiao) on delete cascade,
    primary key (num_concelho, num_regiao)
);

```

```

create table instituicao(
    nome varchar(50) not null unique ,
    tipo varchar(50) not null,
    num_regiao serial not null,
    num_concelho serial not null ,
    check ( tipo in ('farmacia', 'laboratorio', 'clinica', 'hospital') ),
    foreign key (num_regiao, num_concelho) references concelho(num_regiao, num_concelho)
on delete cascade,
    primary key (nome)
);

```

```

create table medico(
    num_cedula int not null unique,
    nome varchar(50) not null,
    especialidade varchar(50) not null,
    primary key (num_cedula)
);

```

```

create table consulta(
    num_cedula int not null ,
    num_doente int not null ,
    data date not null ,
    nome_instituicao varchar(50) not null ,
    unique(num_doente, data, nome_instituicao),
    check ( extract(isodow from data) not in (6,7) ),
    foreign key (num_cedula) references medico(num_cedula) on delete cascade ,
    foreign key (nome_instituicao) references instituicao(nome) on delete cascade ,
    primary key (num_cedula, num_doente, data)
);

```

```

create table prescricao(
    num_cedula int not null ,
    num_doente int not null ,
    data date not null ,
    substancia varchar(50) not null ,
    quant int not null ,
    foreign key (num_cedula, num_doente, data) references consulta(num_cedula,
num_doente, data) on delete cascade ,
    primary key (num_cedula, num_doente, data, substancia)
);

```

```

create table analise(
    num_analise int not null unique ,
    especialidade varchar(50) not null ,
    num_cedula int ,
    num_doente int ,
    data date ,
    data_registro date not null ,
    nome varchar(50) not null ,
    quant integer not null ,
    inst varchar(50) not null ,
    check ( verifica_especialidade(num_cedula, num_doente, data, especialidade) ),
    foreign key (inst) references instituicao(nome) on delete cascade ,
    foreign key (num_cedula, num_doente, data) references consulta(num_cedula,
num_doente, data) on delete cascade ,
    primary key (num_analise)
);

```

```

create table venda_farmacia(
    num_venda serial not null unique ,
    data_registro date not null ,
    substancia varchar(50) not null ,
    quant int not null ,
    preco int not null ,
    inst varchar(50) not null ,
    foreign key (inst) references instituicao(nome) on delete cascade ,
    primary key (num_venda)
);

```

```

create table prescricao_venda(
    num_cedula serial not null ,
    num_doente int not null ,
    data date not null ,
    substancia varchar(50) not null ,
    num_venda serial not null ,
    foreign key (num_venda) references venda_farmacia(num_venda) on delete cascade ,
    foreign key (num_cedula, num_doente, data, substancia) references
prescricao(num_cedula, num_doente, data, substancia) on delete cascade ,
    primary key (num_cedula, num_doente, data, substancia, num_venda));

```

## Consultas em SQL

/\* Query 1 \*/ -- testar com '2020-11-19'

```
select i.num_concelho
from instituicao i inner join venda_farmacia vf on i.nome = vf.inst
where vf.data_registro = current_date
group by i.num_concelho
having sum(vf.preco) >= all( select sum(vf1.preco)
                           from instituicao i1 inner join venda_farmacia vf1 on i1.nome = vf1.inst
                           where vf1.data_registro = current_date
                           group by i1.num_concelho );
```

/\* Query 2 \*/

```
select t1.num_regiao, t1.num_cedula, t1.total_perscricoes
from ( select i.num_regiao, m.num_cedula, count(m.num_cedula) as total_perscricoes
      from medico m inner join prescricao p on m.num_cedula = p.num_cedula
      inner join consulta c on p.num_cedula = c.num_cedula and
      p.num_doente = c.num_doente and
      p.data = c.data
      inner join instituicao i on c.nome_instituicao = i.nome
      where extract(year from p.data) = 2019 and extract(month from p.data) <= 6
      group by i.num_regiao, m.num_cedula )
as t1
inner join ( select t3.num_regiao, max(t3.total_perscricoes) as max_perscricoes
            from ( select i.num_regiao, m.num_cedula, count(m.num_cedula) as total_perscricoes
                  from medico m inner join prescricao p on m.num_cedula = p.num_cedula
                  inner join consulta c on p.num_cedula = c.num_cedula and
                  p.num_doente = c.num_doente and
                  p.data = c.data
                  inner join instituicao i on c.nome_instituicao = i.nome
                  where extract(year from p.data) = 2019 and extract(month from p.data) <= 6
                  group by i.num_regiao, m.num_cedula ) as t3
            group by t3.num_regiao)
as t2 on t1.num_regiao = t2.num_regiao and t1.total_perscricoes = t2.max_perscricoes
group by t1.num_regiao, t1.num_cedula, t1.total_perscricoes ;
```

/\* Query 3 \*/

```
select m.num_cedula
from medico m inner join prescricao_venda pv on m.num_cedula = pv.num_cedula
      inner join venda_farmacia vf on pv.num_venda = vf.num_venda
      inner join instituicao i on vf.inst = i.nome
where vf.substancia = 'aspirina'
      and extract(year from pv.data) = extract(year from current_date)
      and i.num_concelho = 56 and i.tipo = 'farmacia'
group by m.num_cedula
having count(distinct(vf.inst)) = ( select count(*)
                                   from instituição where num_concelho = 56 and tipo = 'farmacia' );
```

```

/* Query 4 */
select distinct a.num_doente
from analise a
where extract(month from a.data) = extract(month from current_date)
and a.num_doente not in( select num_doente
                        from prescricao_venda pv
                        where extract(month from pv.data) = extract(month from current_date)
                        group by num_doente );

```

## Notas extra

- Usar a data '2020-11-19' para testar a query 1
- As restrições que foram definidas **sem** recurso a extensões procedimentais são:
  - RI-regiao-1: nome = {Norte, Centro, Lisboa, Alentejo, Algarve}
  - RI-concelho-1: nome = {concelhos de portugal connental}
  - RI-num\_regiao-1: tipo = {farmacia, laboratorio, clinica, hospital}
  - RI-consulta-1: um médico não pode ver doentes ao fim de semana
  - RI-consulta-2: um doente não pode ter mais de uma consulta por dia na mesma instituição.
- A restrição que foi definida **com** recurso a extensões procedimentais (funções, neste caso), pois de outra maneira não dava para definir é:
  - RI: a consulta associada pode estar omissa; não estando, a especialidade da consulta tem de ser igual à do médico