PROJETO DE BASES DE DADOS – PARTE 3

Ano letivo 2020/2021

Grupo 2 – L05 – Segunda-feira 12:30h
Professor Tiago Oliveira

Aluno	Esforço em horas
Leonor Veloso, nº92509	8h
Pedro Ramos, nº92539	8h
Sancha Barroso, nº92557	8h

Instituto Superior Técnico
LEIC-A

Criação da Base de Dados

```
drop table if exists Regiao cascade;
drop table if exists Concelho cascade;
drop table if exists Instituicao cascade;
drop table if exists Medico cascade;
drop table if exists Consulta cascade;
drop table if exists Prescricao cascade;
drop table if exists VendaFarmacia cascade;
drop table if exists PrescricaoVenda cascade;
drop table if exists Analise cascade;

create table Regiao(
    num_regiao integer not null unique,
    nome char(50) not null unique check(nome in ('Norte', 'Centro',
'Lisboa', 'Alentejo', 'Algarve')),
    num_habitantes integer not null check(num_habitantes > 0),
    primary key(num_regiao)
);

create table Concelho(
    num_concelho integer not null unique,
    num_regiao integer not null,
    nome char(50) not null,
    num_habitantes integer not null check(num_habitantes > 0),
    foreign key(num_regiao) references Regiao(num_regiao) on delete
cascade on update cascade,
    primary key(num_concelho, num_regiao)
);

create table Instituicao(
    nome char(50) not null unique,
    tipo char(20) not null check(tipo in ('farmacia',
'laboratorio', 'clinica', 'hospital')),
    num_concelho integer not null,
    num_regiao integer not null,
    foreign key(num_concelho, num_regiao) references
Concelho(num_concelho, num_regiao) on delete cascade on update
cascade,
    primary key(nome)
);

create table Medico(
```

```
    num_cedula integer not null unique,
    nome char(50) not null,
    especialidade char(50) not null,
    primary key(num_cedula)
);

create table Consulta(
    num_cedula integer not null,
    num_doente integer not null,
    dia_hora timestamp not null check(extract(dow from dia_hora)
not in (0, 6)),
    nome_instituicao char(50) not null,
    foreign key(num_cedula) references Medico(num_cedula) on delete
cascade on update cascade,
    foreign key(nome_instituicao) references Instituicao(nome) on
delete cascade on update cascade,
    primary key(num_cedula, num_doente, dia_hora),
    constraint RI_consulta_2 unique(num_doente, dia_hora,
nome_instituicao)
);

create table Prescricao(
    num_cedula integer not null,
    num_doente integer not null,
    dia_hora timestamp not null,
    substancia char(50) not null,
    quant integer not null check(quant > 0),
    foreign key(num_cedula, num_doente, dia_hora) references
Consulta(num_cedula, num_doente, dia_hora) on delete cascade on
update cascade,
    primary key(num_cedula, num_doente, dia_hora, substancia)
);



create table VendaFarmacia(
    num_venda integer not null unique,
    inst char(50) not null,
    data_registo timestamp not null,
    substancia char(50) not null,
    quant integer not null check(quant > 0),
    preco decimal(6,2) not null check(preco > 0),
    foreign key(inst) references Instituicao(nome) on delete
cascade on update cascade,
```

```
    primary key(num_venda)
);

create table PrescricaoVenda(
    num_cedula integer not null,
    num_doente integer not null,
    dia_hora timestamp not null,
    substancia char(50) not null,
    num_venda integer not null,
    foreign key(num_venda) references VendaFarmacia(num_venda) on
delete cascade on update cascade,
    foreign key(num_cedula, num_doente, dia_hora, substancia)
references Prescricao(num_cedula, num_doente, dia_hora, substancia)
on delete cascade on update cascade,
    primary key(num_cedula, num_doente, dia_hora, substancia,
num_venda)
);


/*
extensão procedimental para a restrição de integridade da analise

create or replace function getEspecialidade ()
returns char(50) as $especialidade$
declare
    especialidade char(50);
begin
   select m.especialidade into especialidade from Medico as m where
m.num_cedula = num_cedula;
   return especialidade;
end;
$especialidade$ language plpgsql;
*/

create table Analise(
    num_analise integer not null,
    especialidade char(50) not null,
    num_cedula integer,
    num_doente integer,
    dia_hora timestamp,
    data_registo timestamp not null,
    nome char(50) not null,
    quant integer not null check(quant > 0),
    inst char(50) not null,
```

```
    foreign key(num_cedula, num_doente, dia_hora) references
Consulta(num_cedula, num_doente, dia_hora),
    foreign key(inst) references Instituicao(nome) on delete
cascade on update cascade,
    primary key(num_analise)
    /*
    constraint RI_analise check((num_cedula = null and num_doente =
null and dia_hora = null) or
    getEspecialidade() = especialidade)

    comentado porque é uma restrição de integridade definida com
recurso a extensões procedimentais
    */
);
```

Consultas em SQL

```
/*QUERY 1*/
select c.nome
from concelho as c, instituicao as i, vendafarmacia as v
where date(v.data_registo) = current_date and c.num_concelho =
i.num_concelho and i.nome = v.inst
group by c.nome
having sum(preco) >= ALL
    (select sum(v.preco)
    from concelho as c, instituicao as i, vendafarmacia as v
    where date(v.data_registo) = current_date and i.nome = v.inst
and i.num_concelho = c.num_concelho
    group by c.nome
    );


/*QUERY 2*/
select r.nome, m.nome
from consulta as c, prescricao as p, instituicao as i, medico as m,
regiao as r
where c.num_cedula = p.num_cedula and c.num_doente = p.num_doente
and c.dia_hora = p.dia_hora and
p.dia_hora between '2019-1-1 00:00:00' and '2019-6-30 23:59:59' and
p.num_cedula = m.num_cedula and c.nome_instituicao = i.nome and
i.num_regiao = r.num_regiao
group by r.num_regiao, m.nome
having count(p.dia_hora) >= ALL
    (select count(p.dia_hora)
```

```
    from consulta as c natural join prescricao as p, instituicao as
i, medico as m, regiao as r
    where p.dia_hora between '2019-1-1 00:00:00' and '2019-6-30
23:59:59' and p.num_cedula = m.num_cedula and c.nome_instituicao =
i.nome and i.num_regiao = r.num_regiao
    group by r.num_regiao, m.nome
    );

/*QUERY 3*/
select distinct m.nome
from prescricaovenda as p_v, medico as m, vendafarmacia as v_f,
instituicao as i,concelho as c
where p_v.num_cedula = m.num_cedula and p_v.num_venda =
v_f.num_venda and
v_f.inst = i.nome and i.num_concelho = c.num_concelho and c.nome =
'Arouca' and i.tipo = 'farmacia'
group by m.nome
having count(p_v.num_venda) = (
    select count(i.nome)
    from instituicao as i, concelho as c
    where c.nome = 'Arouca' and i.num_concelho = c.num_concelho and
i.tipo = 'farmacia'
);

/*QUERY 4*/
select distinct a.num_doente
from analise as a
where extract(year from a.data_registo) = extract(year from
current_date) and
extract(month from a.data_registo) = extract(month from
current_date)
and not exists (
    select * from venda_farmacia as v_f, prescricaovenda as p_v
    where a.num_doente = num_doente and extract(year from
a.data_registo) = extract(year from current_date) and
    extract(month from a.data_registo) = extract(month from
current_date) and v_f.num_venda = p_v.num_venda
    );
```

Nota: A aplicação em python será entregue na parte 4 do relatório.