



Projeto de Bases de Dados - Parte 3

Professor Daniel Faria

Grupo 47 – Turno L07 – 4ª feira às 8h

| Alunos | Esforço |
|---------------------------|----------------|
| Maria Duarte (90415) | 10h (33.3%) |
| Daniela Carvalho (922443) | 10h (33.3%) |
| Laura Baeta (92507) | 10h (33.3%) |

Criação da Base de Dados

→ schema.sql

```
drop table if exists instituicao cascade;
drop table if exists concelho cascade;
drop table if exists regiao cascade;
drop table if exists medico cascade;
drop table if exists consulta cascade;
drop table if exists prescricao cascade;
drop table if exists analise cascade;
drop table if exists venda_farmacia cascade;
drop table if exists prescricao_venda cascade;

create table regiao (
  num_regiao varchar(1) not null, --5 regioes
  nome varchar(8) not null,
  num_habitantes integer not null,
  constraint pk_regiao primary key(num_regiao));

create table concelho (
  num_concelho varchar(3) not null, --308 municipios
  num_regiao varchar(1) not null,
  nome varchar(28) not null,
  num_habitantes integer not null,
  constraint pk_concelho primary key(num_concelho, num_regiao),
  constraint fk_concelho_regiao foreign key(num_regiao) references
regiao(num_regiao));

create table instituicao (
  nome varchar(60) not null,
  tipo varchar(11) not null check(tipo in ('Farmacia', 'Laboratorio', 'Clinica',
'Hospital'))1,
  num_regiao varchar(1) not null,
  num_concelho varchar(3) not null,
  constraint pk_instituicao primary key(nome),
  constraint fk_instituicao_concelho foreign key(num_regiao, num_concelho)
references concelho(num_regiao, num_concelho));

create table medico (
  num_cedula varchar(5) not null,
  nome varchar(150) not null,
  especialidade varchar(50) not null,
  constraint pk_medico primary key(num_cedula));
```

¹ RI-instituição-1: tipo = {farmacia, laboratorio, clinica, hospital}

```
create table consulta2 (
  num_cedula varchar(5) not null,
  num_doente varchar(9) not null,
  data_consulta date not null check( extract (dow from data_consulta) != 0 or
extract (dow from data_consulta) != 6 ),3
  nome_instituicao varchar(60) not null,
  constraint pk_consulta primary key(num_cedula, num_doente, data_consulta),
  constraint fk_consulta_medico foreign key(num_cedula) references
medico(num_cedula),
  constraint fk_consulta_instituicao foreign key(nome_instituicao) references
instituicao(nome));

create table prescricao (
  num_cedula varchar(5) not null,
  num_doente varchar(9) not null,
  data_consulta date not null,
  substancia varchar(50) not null,
  quant varchar(10) not null,
  constraint pk_prescricao primary key(num_cedula, num_doente, data_consulta,
substancia),
  constraint fk_prescricao_consulta foreign key(num_cedula, num_doente,
data_consulta) references consulta(num_cedula, num_doente, data_consulta));

create table analise4 (
  num_analise varchar(7) not null,
  especialidade varchar(50) not null,
  num_cedula varchar(5) not null,
  num_doente varchar(9) not null,
  data_consulta date not null,
  data_registo date not null,
  nome varchar(30) not null,
  quant varchar(10) not null,
  inst varchar(60) not null,
  constraint pk_analise primary key(num_analise),
  constraint fk_analise_consulta foreign key(num_cedula, num_doente, data_consulta)
references consulta(num_cedula, num_doente, data_consulta),
  constraint fk_analise_instituicao foreign key(inst) references instituicao(nome));

create table venda_farmacia (
  num_venda varchar(23) not null,
  data_registo date not null,
  substancia varchar(50) not null,
  quant varchar(10) not null,
  preco numeric(4,2) not null,
  inst varchar(60) not null,
  constraint pk_venda_farmacia primary key(num_venda),
  constraint fk_venda_farmacia_instituicao foreign key(inst) references
instituicao(nome));
```

² RI-consulta-2: um doente não pode ter mais de uma consulta por dia na mesma instituição - consideramos ser uma extensão procedimental

³ RI-consulta-1:um médico não pode ver doentes ao fim de semana

⁴ RI-analise: a consulta associada pode estar omissa; não estando, a especialidade da consulta tem de ser igual à do médico - consideramos ser uma extensão procedimental

```
create table prescricao_venda (  
  num_cedula varchar(5) not null,  
  num_doente varchar(9) not null,  
  data_consulta date not null,  
  substancia varchar(50) not null,  
  num_venda varchar(23) not null,  
  constraint pk_prescricao_venda primary key(num_cedula, num_doente, data_consulta,  
substancia, num_venda),  
  constraint fk_prescricao_venda_venda_farmacia foreign key(num_venda) references  
venda_farmacia(num_venda),  
  constraint fk_prescricao_venda_prescricao foreign key(num_cedula, num_doente,  
data_consulta, substancia) references prescricao(num_cedula, num_doente,  
data_consulta, substancia));
```

Consultas em SQL

→ queries.sql

1) Qual o concelho onde se fez o maior volume de vendas hoje?

```
select c.num_concelho, c.nome as nome_concelho, c.num_regiao, c.num_habitantes  
from (concelho c inner join instituicao i on (c.num_concelho = i.num_concelho and  
c.num_regiao = i.num_regiao)) inner join venda_farmacia v on (v.inst=i.nome)  
where v.data_registo = CURRENT_DATE  
group by c.num_concelho, c.nome, c.num_regiao, c.num_habitantes  
having count(c.num_concelho) >= ALL  
  (select count(c2.num_concelho)  
   from (concelho c2 inner join instituicao i2 on (c2.num_concelho = i2.num_concelho  
and c2.num_regiao = i2.num_regiao)) inner join venda_farmacia v2 on  
(v2.inst=i2.nome)  
   where v2.data_registo = CURRENT_DATE  
   group by c2.num_concelho, c2.nome  
  );
```

2) Qual o médico que mais prescreveu no 1o semestre de 2019 em cada região?

```
with count_pres as(  
  select i.num_regiao, m.num_cedula, m.nome, count(distinct(p.num_cedula))  
    from ((prescricao p inner join medico m on (p.num_cedula = m.num_cedula)) inner  
join consulta c on ( p.num_cedula=c.num_cedula and p.num_doente=c.num_doente and  
p.data_consulta=c.data_consulta)) inner join instituicao i on  
c.nome_instituicao=i.nome  
   where p.data_consulta >= '01-01-2019' and p.data_consulta <= '30-06-2019'  
   group by i.num_regiao, m.num_cedula, m.nome),  
max_count as (  
  select max(count), num_regiao  
    from count_pres  
   group by num_regiao  
)  
select count_pres.num_regiao as regiao, count_pres.nome as nome_medico,  
count_pres.num_cedula as cedula_medico  
from count_pres, max_count  
where max_count.max = count_pres.count and  
max_count.num_regiao=count_pres.num_regiao  
order by count_pres.num_regiao;
```

- 3) Quais são os médicos que já prescreveram aspirina em receitas aviadas em todas as farmácias do concelho de Arouca este ano?

```
with count_pres_in_concelho as (  
  select distinct p.num_cedula, count(distinct (v.inst))  
  from ((venda_farmacia v inner join instituicao i on (v.inst = i.nome)) inner join  
concelho c on (i.num_concelho = c.num_concelho)) inner join prescricao_venda p on  
(p.num_venda=v.num_venda)  
  where c.nome = 'Arouca' and v.substancia = 'Aspirina' and v.data_registo >=  
'01-01-2020'  
  group by p.num_cedula  
) ,  
count_pharmacies_in_concelho as (  
  select count(distinct(v.inst))  
  from (venda_farmacia v inner join instituicao i on (v.inst = i.nome)) inner join  
concelho c on (i.num_concelho = c.num_concelho)  
  where c.nome = 'Arouca'  
)  
select count_pres_in_concelho.num_cedula  
from count_pharmacies_in_concelho, count_pres_in_concelho  
where count_pres_in_concelho.count = count_pharmacies_in_concelho.count;
```

- 4) Quais são os doentes que já fizeram análises mas ainda não aviaram prescrições este mês?

```
with count_pres_current_month as(  
  select pres.num_doente, count(v.data_registo)  
  from venda_farmacia v inner join prescricao_venda pres on (v.num_venda =  
pres.num_venda)  
  where extract(month from v.data_registo) = extract(month from CURRENT_DATE) and  
extract(year from v.data_registo) = extract(year from CURRENT_DATE)  
  group by pres.num_doente  
) ,  
count_analise_current_month as(  
  select a.num_doente, count(a.num_doente)  
  from analise a  
  where extract(month from a.data_registo) = extract(month from CURRENT_DATE) and  
extract(year from a.data_registo) = extract(year from CURRENT_DATE)  
  group by a.num_doente  
)  
select num_doente  
from count_analise_current_month  
where count>0  
except (  
  select num_doente  
  from count_pres_current_month  
  where count>0  
) ;
```

Relação Entre os Diversos Ficheiros

O ficheiro **populate.sql** contém os dados necessários ao preenchimento das tabelas previamente criadas através do ficheiro **schema.sql**. O ficheiro **queries.sql** é o ficheiro com consultas à base de dados formada pelas tabelas do **schema.sql** e preenchida com informação do ficheiro **populate.sql**.

Nota: Na criação do ficheiro populate.sql consideramos que a data atual seria a data de entrega, 20-11-2020, apesar de no queries.sql usarmos o `current_date`.