

Projeto de Bases de Dados - Parte 3

Grupo 7

Turno BD225179577L04 (Segunda-feira, 8:00-9:30, Lab 10)

Docente do Laboratório Tiago Oliveira

Catarina Carreiro, 92438, 33%, 10 horas
Cristiano Clemente, 92440, 33%, 10 horas
Mónica Jin, 92532, 34%, 10 horas

Notas

1 - Assumimos que as tabelas *regiao* e *concelho* já existem e estão bem definidas (regras de integridade) e bem preenchidas pelo administrador da aplicação de forma que na tabela *instituicao* seria apenas preciso introduzir uma restrição de *foreign key* (*schema.sql*, linha 8).

2 - Em relação à restrição de integridade RI-analise:

- Implementámos a possibilidade da consulta estar omissa (deixando no *schema.sql* que os atributos *num_cedula*, *num_doente* e *data* possam ser *NULL*).
- Não implementámos a 2ª parte da RI-analise porque comparar a especialidade da consulta com a especialidade do médico, se existir consulta associada, exigiria uma extensão procedimental (*trigger* ou *stored procedure*).

schema.sql

```
DROP TABLE IF EXISTS instituicao CASCADE;
```

```
CREATE TABLE instituicao (  
    nome varchar(100) PRIMARY KEY,  
    tipo varchar(100) NOT NULL CHECK (tipo IN ('farmacia',  
'laboratorio', 'clinica', 'hospital')),  
    num_regiao integer NOT NULL,  
    num_concelho integer NOT NULL  
    -- FOREIGN KEY (num_regiao, num_concelho) REFERENCES concelho  
(num_regiao, num_concelho) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
DROP TABLE IF EXISTS medico CASCADE;
```

```
CREATE TABLE medico (  
    num_cedula integer PRIMARY KEY,  
    nome varchar(100) NOT NULL,  
    especialidade varchar(100) NOT NULL  
);
```

```
DROP TABLE IF EXISTS consulta CASCADE;
```

```
CREATE TABLE consulta (  
    num_cedula integer REFERENCES medico (num_cedula) ON DELETE  
CASCADE ON UPDATE CASCADE,  
    num_doente integer,  
    data date CHECK (extract(DOW FROM data) NOT IN (6, 0)), --  
dom:0, seg:1, ter:2, ...  
    nome_instituicao varchar(100) NOT NULL REFERENCES instituicao  
(nome) ON DELETE CASCADE ON UPDATE CASCADE,  
    PRIMARY KEY (num_cedula, num_doente, data),  
    UNIQUE (num_doente, data, nome_instituicao)  
);
```

```
DROP TABLE IF EXISTS prescricao CASCADE;
```

```
CREATE TABLE prescricao (  
    num_cedula integer,  
    num_doente integer,  
    data date,  
    substancia varchar(100),  
    quant integer NOT NULL,  
    PRIMARY KEY (num_cedula, num_doente, data, substancia),  
    FOREIGN KEY (num_cedula, num_doente, data) REFERENCES consulta  
(num_cedula, num_doente, data) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
DROP TABLE IF EXISTS analise CASCADE;
```

```
CREATE TABLE analise (  
    num_analise integer PRIMARY KEY,  
    especialidade varchar(100) NOT NULL,  
    num_cedula integer,  
    num_doente integer,  
    data date,  
    data_registro date NOT NULL,  
    nome varchar(100) NOT NULL,  
    quant integer NOT NULL,  
    inst varchar(100) NOT NULL REFERENCES instituicao (nome) ON  
DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (num_cedula, num_doente, data) REFERENCES consulta  
(num_cedula, num_doente, data) ON DELETE SET NULL ON UPDATE CASCADE  
);
```

```
DROP TABLE IF EXISTS venda_farmacia CASCADE;
```

```
CREATE TABLE venda_farmacia (  
    num_venda integer PRIMARY KEY,  
    data_registro date NOT NULL,  
    substancia varchar(100) NOT NULL,  
    quant integer NOT NULL,  
    preco money NOT NULL,  
    inst varchar(100) NOT NULL REFERENCES instituicao (nome) ON  
DELETE CASCADE ON UPDATE CASCADE  
);
```

```
DROP TABLE IF EXISTS prescricao_venda CASCADE;
```

```
CREATE TABLE prescricao_venda (  
    num_cedula integer,  
    num_doente integer,  
    data date,  
    substancia varchar(100),  
    num_venda integer NOT NULL REFERENCES venda_farmacia (num_venda)  
ON DELETE CASCADE ON UPDATE CASCADE,  
    PRIMARY KEY (num_cedula, num_doente, data, substancia,  
num_venda),  
    FOREIGN KEY (num_cedula, num_doente, data, substancia)  
REFERENCES prescricao (num_cedula, num_doente, data, substancia) ON  
DELETE CASCADE ON UPDATE CASCADE  
);
```

queries.sql

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

WITH

```
volume(num_regiao, num_concelho, sum) AS (  
    SELECT num_regiao, num_concelho, SUM(quant * preco)  
    FROM venda_farmacia vf JOIN instituicao i ON vf.inst =
```

i.nome

```
    WHERE data_registo = CURRENT_DATE  
    GROUP BY num_regiao, num_concelho),
```

```
maxVolume(max) AS (  
    SELECT MAX(sum)  
    FROM volume)
```

```
SELECT num_regiao, num_concelho  
FROM maxVolume, volume  
WHERE volume.sum = maxVolume.max;
```

-- 2. Qual o medico que mais prescreveu no 1o semestre de 2019 em cada regiao?

WITH

```
numPrescricao(num_cedula, num_regiao, count) AS(  
    SELECT num_cedula, num_regiao, COUNT(num_cedula)  
    FROM prescricao NATURAL JOIN consulta JOIN instituicao ON  
nome = nome_instituicao  
    WHERE data BETWEEN '01-01-2019' AND '30-06-2019'  
    GROUP BY num_cedula, num_regiao),
```

```
maxPrescricao(num_regiao, max) AS(  
    SELECT num_regiao, MAX(count)  
    FROM numPrescricao  
    GROUP BY num_regiao)
```

```
SELECT num_cedula, maxPrescricao.num_regiao  
FROM numPrescricao NATURAL JOIN maxPrescricao  
WHERE count = max;
```

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

WITH

```
    numFarmacia(num_cedula, num) AS (  
        SELECT num_cedula, COUNT(distinct inst)  
        FROM prescricao_venda NATURAL JOIN venda_farmacia JOIN  
instituicao ON inst = instituicao.nome  
        WHERE substancia = 'aspirina'  
        AND date_part('year', data_registro) = date_part('year',  
CURRENT_DATE)  
        AND num_regiao=0 AND num_concelho=0  
        GROUP BY num_cedula),
```

```
    totalFarmacias(total) AS(  
        SELECT COUNT(*)  
        FROM instituicao  
        WHERE num_regiao=0 AND num_concelho=0 AND tipo =  
'farmacia') --Arouca
```

```
SELECT num_cedula  
FROM numFarmacia, totalFarmacias  
WHERE num = total;
```

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

```
(SELECT num_doente  
FROM analise  
WHERE num_doente IS NOT NULL  
AND date_part('month', data_registro)=date_part('month',  
CURRENT_DATE))  
EXCEPT  
(SELECT num_doente  
FROM prescricao_venda  
WHERE date_part('month', data)=date_part('month', CURRENT_DATE))
```