

Projeto de Bases de Dados – Parte 3

Professor Daniel Faria

Grupo 8 – Turno L09 – Quarta-feira 09:30		
Aluno		Esforço (horas)
David Baptista	92446	4,5 horas – 33,33%
João Antunes	92498	4,5 horas – 33,33%
Leonor Santos	92508	4,5 horas – 33,33%

A primeira etapa da nossa solução passou por criar o esquema da BD (schema.sql) tendo por base o esquema relacional fornecido. Neste esquema acrescentamos ainda as regras de integridade possíveis de representar sem recorrer a *Stored Procedures* nem a *Triggers*.

Tendo em conta estas últimas restrições, não foi possível representar a seguinte RI: “Quando a consulta não está omissa, a especialidade da consulta tem de ser igual à do médico”.

Esta só seria possível representar recorrendo a um Trigger que fosse ativado quando adicionamos ou alteramos uma linha da tabela análise.

De seguida, criamos o ficheiro populate.sql que utilizamos para preencher a base de dados criada anteriormente.

Na terceira etapa, desenvolvemos as queries (queries.sql) exigidas, executadas sobre a base de dados já criada e que tiram partido dos dados inseridos.

Por último, implementamos a aplicação (app.cgi) que faz a ligação entre o site que interage com o utilizador e a base de dados.

A arquitetura da aplicação é a seguinte

Dentro da pasta /web temos o ficheiro app.cgi e uma segunda pasta /templates com todas as páginas html com que o utilizador vai interagir.

Temos um conjunto de páginas que permitem inserir, editar e remover registos de médicos, instituições, prescrições e análises. Estas páginas, apresentam um form para inserir os registos e uma tabela com os registos atuais e com botões para editar e apagar em cada linha.

O botão editar leva a outra página só para atualizar os campos pretendidos.

Para além disso, de forma a realizar vendas em farmácias, temos duas páginas distintas (com ou sem prescrição) que pedem os dados consoante o tipo de venda.

Temos ainda uma página que pede o identificador do médico e um período de tempo para listar as substâncias prescritas por esse médico nesse período.

Finalmente, há uma última página que lista os valores de glicémia mais alto e mais baixo em cada conelho e respectivo doente.

schema.sql

```
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;

CREATE TABLE regiao
(
    num_regiao NUMERIC(1, 0) NOT NULL UNIQUE,
    nome VARCHAR(80) NOT NULL CONSTRAINT nomes_regiao CHECK (nome IN ('Norte', 'Centro', 'Lisboa',
    'Alentejo', 'Algarve')),
    num_habitantes NUMERIC(10,0),
    CONSTRAINT pk_num_regiao PRIMARY KEY(num_regiao));

CREATE TABLE concelho
(
    num_concelho NUMERIC(10, 0) NOT NULL UNIQUE,
    num_regiao NUMERIC(1, 0) NOT NULL,
    nome VARCHAR(80) CONSTRAINT nomes_concelho CHECK (nome IN('Agueda', 'Albergaria-a-Velha', 'Anadia',
    'Arouca', 'Aveiro', 'Castelo de Paiva', 'Espinho', 'Estarreja', 'Ilhavo', 'Mealhada', 'Murtosa', 'Oliveira
    de Azemeis', 'Oliveira do Bairro', 'Ovar', 'Santa Maria da Feira', 'Sao Joao da Madeira', 'Sever do Vouga',
    'Vagos', 'Vale de Cambra', 'Aljustrel', 'Almodovar', 'Alvito', 'Barrancos', 'Beja', 'Castro Verde', 'Cuba',
    'Ferreira do Alentejo', 'Mertola', 'Moura', 'Odemira', 'Ourique', 'Sarpa', 'Vidigueira', 'Amares',
    'Barcelos', 'Braga', 'Cabeceiras de Basto', 'Celorico de Basto', 'Esposende', 'Fafe', 'Guimaraes', 'Povoa de
    Lanhoso', 'Terras de Bouro', 'Vieira do Minho', 'Vila Nova de Famalicao', 'Vila Verde', 'Vizela', 'Alfandega
    da Fe', 'Braganca', 'Carraceda de Ansiaes', 'Freixo de Espada a Cinta', 'Macedo de Cavaleiros', 'Miranda do
    Douro', 'Mirandela', 'Mogadouro', 'Torre de Moncorvo', 'Vila Flor', 'Vimioso', 'Vinhais', 'Belmonte',
    'Castelo Branco', 'Covilha', 'Fundao', 'Idanha-a-Nova', 'Oleiros', 'Penamacor', 'Proenca-a-Nova', 'Serta',
    'Vila de Rei', 'Vila Velha de Rodao', 'Arganil', 'Cantanhede', 'Coimbra', 'Condeixa-a-Nova', 'Figueira da
    Foz', 'Gois', 'Lousa', 'Mira', 'Miranda do Corvo', 'Montemor-o-Velho', 'Oliveira do Hospital', 'Pampilhosa
    da Serra', 'Penacova', 'Penela', 'Soure', 'TABua', 'Vila Nova de Poiares', 'Alandroal', 'Arraiolos',
    'Borba', 'Estremoz', 'Evora', 'Montemor-o-Novo', 'Mora', 'Mourao', 'Portel', 'Redondo', 'Reguengos de
    Monsaraz', 'Vendas Novas', 'Viana do Alentejo', 'Vila Viciosa', 'Albufeira', 'Alcoutim', 'Aljezur', 'Castro
    Marim', 'Faro', 'Lagoa', 'Lagos', 'Loule', 'Monchique', 'Olhao', 'Portimao', 'Sao Bras de Alportel',
    'Silves', 'Tavira', 'Vila do Bispo', 'Vila Real de Santo Antonio', 'Aguiar da Beira', 'Almeida', 'Celorico
    da Beira', 'Figueira de Castelo Rodrigo', 'Fornos de Algodres', 'Gouveia', 'Guarda', 'Manteigas', 'Mêda',
    'Pinhel', 'Sabugal', 'Seia', 'Trancoso', 'Vila Nova de Foz Coa', 'Alcobaca', 'AlvaiAzere', 'Ansiao',
    'Batalha', 'Bombarral', 'Caldas da Rainha', 'Castanheira de Pêra', 'Figueiro dos Vinhos', 'Leiria', 'Marinha
    Grande', 'Nazare', 'Obidos', 'Pedrogao Grande', 'Peniche', 'Pombal', 'Porto de Mos', 'Alenquer', 'Amadora',
    'Arruda dos Vinhos', 'Azambuja', 'Cadaval', 'Cascais', 'Lisboa', 'Loures', 'Lourinha', 'Mafra', 'Odivelas',
    'Oeiras', 'Sintra', 'Sobral de Monte Agraco', 'Torres Vedras', 'Vila Franca de Xira', 'Alter do Chao',
    'Arronches', 'Avis', 'Campo Maior', 'Castelo de Vide', 'Crato', 'Elvas', 'Fronteira', 'Gaviao', 'Marvao',
    'Monforte', 'Nisa', 'Ponte de Sor', 'Portalegre', 'Sousel', 'Amarante', 'Baiao', 'Felgueiras', 'Gondomar',
    'Lousada', 'Maia', 'Marco de Canaveses', 'Matosinhos', 'Pacos de Ferreira', 'Paredes', 'Penafiel', 'Porto',
    'Povoa de Varzim', 'Santo Tirso', 'Trofa', 'Valongo', 'Vila do Conde', 'Vila Nova de Gaia', 'Abrantes',
    'Alcanena', 'Almeirim', 'Alpiarca', 'Benavente', 'Cartaxo', 'Chamusca', 'Constancia', 'Coruche',
    'Entroncamento', 'Ferreira do Zêzere', 'Golega', 'Macao', 'Ourem', 'Rio Maior', 'Salvaterra de Magos',
    'Santarem', 'Sardoal', 'Tomar', 'Torres Novas', 'Vila Nova da Barquinha', 'Alcacer do Sal', 'Alcochete',
```

```
'Almada', 'Barreiro', 'Grandola', 'Moita', 'Montijo', 'Palmela', 'Santiago do Cacem', 'Seixal', 'Sesimbra',
'Setubal', 'Sines', 'Arcos de Valdevez', 'Caminha', 'Melgaco', 'Moncao', 'Paredes de Coura', 'Ponte da
Barca', 'Ponte de Lima', 'Valenca', 'Viana do Castelo', 'Vila Nova de Cerveira', 'Alijo', 'Boticas',
'Chaves', 'Mesao Frio', 'Mondim de Basto', 'Montalegre', 'Murca', 'Peso da Regua', 'Ribeira de Pena',
'Sabrosa', 'Santa Marta de Penaguião', 'Valpacos', 'Vila Pouca de Aguiar', 'Vila Pouca de Aguiar', 'Vila
Real', 'Armamar', 'Carregal do Sal', 'Castro Daire', 'Cinfaes', 'Lamego', 'Mangualde', 'Moimenta da Beira',
'Mortagua', 'Nelas', 'Oliveira de Frades', 'Penalva do Castelo', 'Penedono', 'Resende', 'Santa Comba Dao',
'Sao Joao da Pesqueira', 'Sao Pedro do Sul', 'Satao', 'Sernancelhe', 'Tabuaco', 'Tarouca', 'Tondela', 'Vila
Nova de Paiva', 'Viseu', 'Vouzela'))),
    num_habitantes NUMERIC(10,0),
    CONSTRAINT pk_concelho PRIMARY KEY(num_concelho, num_regiao),
    CONSTRAINT fk_num_regiao FOREIGN KEY(num_regiao) REFERENCES regiao(num_regiao) ON DELETE CASCADE);

CREATE TABLE instituicao
    (nome VARCHAR(80) NOT NULL UNIQUE,
     tipo VARCHAR(80) NOT NULL CONSTRAINT tipo_instituicao CHECK (tipo IN('farmacia', 'laboratorio',
'clinica', 'hospital'))),
    num_regiao NUMERIC(1, 0),
    num_concelho NUMERIC(10, 0),
    CONSTRAINT pk_nome PRIMARY KEY(nome),
    CONSTRAINT fk_num_regiao FOREIGN KEY(num_regiao) REFERENCES regiao(num_regiao) ON DELETE CASCADE,
    CONSTRAINT fk_num_concelho FOREIGN KEY(num_concelho) REFERENCES concelho(num_concelho) ON DELETE
CASCADE);

CREATE TABLE medico
    (num_cedula NUMERIC(10, 0) NOT NULL UNIQUE,
     nome VARCHAR(255),
     especialidade VARCHAR(80),
     CONSTRAINT pk_num_cedula PRIMARY KEY(num_cedula));

CREATE TABLE consulta
    (num_cedula NUMERIC(10, 0) NOT NULL,
     num_doente NUMERIC(10, 0) NOT NULL,
     data_consulta DATE NOT NULL CONSTRAINT data_fds CHECK(EXTRACT(DOW FROM data_consulta) NOT IN(0, 6)),
     nome_instituicao VARCHAR(80),
     CONSTRAINT pk_consulta PRIMARY KEY(num_cedula, num_doente, data_consulta),
     CONSTRAINT fk_num_cedula FOREIGN KEY(num_cedula) REFERENCES medico(num_cedula) ON DELETE CASCADE,
     CONSTRAINT fk_nome_instituicao FOREIGN KEY(nome_instituicao) REFERENCES instituicao(nome) ON DELETE
CASCADE,
     CONSTRAINT ri_consulta UNIQUE(num_doente, data_consulta, nome_instituicao));

CREATE TABLE prescricao
    (num_cedula NUMERIC(10, 0) NOT NULL,
     num_doente NUMERIC(10, 0) NOT NULL,
     data_consulta DATE NOT NULL,
     substancia VARCHAR(80) NOT NULL,
     quant NUMERIC(10, 0),
     CONSTRAINT pk_prescricao PRIMARY KEY(num_cedula, num_doente, data_consulta, substancia),
     CONSTRAINT fk_consulta FOREIGN KEY(num_cedula, num_doente, data_consulta) REFERENCES
consulta(num_cedula, num_doente, data_consulta) ON DELETE CASCADE);
```

```
CREATE TABLE analise
(
    num_analise NUMERIC(10, 0) NOT NULL,
    especialidade VARCHAR(80) NOT NULL,
    num_cedula NUMERIC(10, 0),
    num_doente NUMERIC(10, 0),
    data_consulta DATE,
    data_registo DATE NOT NULL,
    nome VARCHAR(80) NOT NULL,
    quant NUMERIC(3, 0) NOT NULL,
    inst VARCHAR(80) NOT NULL,
    CONSTRAINT pk_analise PRIMARY KEY(num_analise),
    CONSTRAINT fk_consulta FOREIGN KEY(num_cedula, num_doente, data_consulta) REFERENCES
consulta(num_cedula, num_doente, data_consulta) ON DELETE CASCADE,
    CONSTRAINT fk_instituicao FOREIGN KEY(inst) REFERENCES instituicao(nome) ON DELETE CASCADE
    /* RI-analise: a consulta associada pode estar omissa; não estando, a especialidade da consulta tem de
ser igual à do médico. */
);

CREATE TABLE venda_farmacia
(
    num_venda NUMERIC(10, 0) NOT NULL,
    data_registo DATE NOT NULL,
    substancia VARCHAR(80),
    quant NUMERIC(10, 0) NOT NULL,
    preco NUMERIC(5, 2) NOT NULL,
    inst VARCHAR(80) NOT NULL,
    CONSTRAINT pk_venda_farmacia PRIMARY KEY(num_venda),
    CONSTRAINT fk_instituicao FOREIGN KEY(inst) REFERENCES instituicao(nome) ON DELETE CASCADE);

CREATE TABLE prescricao_venda
(
    num_cedula NUMERIC(10, 0) NOT NULL,
    num_doente NUMERIC(10, 0) NOT NULL,
    data_consulta DATE NOT NULL,
    substancia VARCHAR(80) NOT NULL,
    num_venda NUMERIC(10, 0) NOT NULL,
    CONSTRAINT pk_prescricao_venda PRIMARY KEY(num_cedula, num_doente, data_consulta, substancia,
num_venda),
    CONSTRAINT fk_num_venda FOREIGN KEY(num_venda) REFERENCES venda_farmacia(num_venda) ON DELETE CASCADE,
    CONSTRAINT fk_prescricao FOREIGN KEY(num_cedula, num_doente, data_consulta, substancia) REFERENCES
prescricao(num_cedula, num_doente, data_consulta, substancia) ON DELETE CASCADE);
```

queries.sql

```
SELECT c.nome, sum(quant * preco) volume_de_venda
FROM concelho c
JOIN instituicao i ON c.num_concelho = i.num_concelho AND c.num_regiao = i.num_regiao
JOIN venda_farmacia v ON i.nome = v.inst
WHERE v.data_registo = current_date
GROUP BY c.nome
HAVING sum(quant * preco) >= ALL
    (SELECT sum(quant * preco)
    FROM concelho co
    JOIN instituicao it ON co.num_concelho = it.num_concelho
```

```
JOIN venda_farmacia ve ON it.nome = ve.inst
WHERE ve.data_registo = current_date
GROUP BY co.nome);
WITH query_name AS (SELECT m.nome, r.nome AS regiao, COUNT(c.num_doente) AS prescricoes
FROM medico m
JOIN prescricao p ON m.num_cedula = p.num_cedula
JOIN consulta c ON p.num_cedula = c.num_cedula AND p.num_doente = c.num_doente AND p.data_consulta =
c.data_consulta
JOIN instituicao i ON c.nome_instituicao = i.nome
JOIN regiao r ON i.num_regiao = r.num_regiao
WHERE EXTRACT(MONTH FROM c.data_consulta) IN ('1','2','3','4','5','6')
AND EXTRACT(YEAR FROM c.data_consulta) = '2019'
GROUP BY m.nome, r.nome)
SELECT regiao, nome, prescricoes FROM (
SELECT regiao, MAX(prescricoes) AS prescricoes
FROM query_name
AS result
GROUP BY regiao)
AS A1
NATURAL JOIN query_name;

SELECT nome FROM medico
WHERE NOT EXISTS (
SELECT DISTINCT i.nome
FROM instituicao i JOIN concelho c ON c.num_concelho = i.num_concelho AND c.num_regiao = i.num_regiao
WHERE LOWER(c.nome) = 'arouca' AND LOWER(i.tipo) = 'farmacia'
EXCEPT
SELECT DISTINCT inst FROM (
SELECT m.nome, v.num_venda, v.inst
FROM prescricao_venda p
JOIN venda_farmacia v ON v.num_venda = p.num_venda AND v.substancia = p.substancia
JOIN instituicao i ON i.nome = v.inst
JOIN concelho c ON c.num_concelho = i.num_concelho AND c.num_regiao = i.num_regiao
JOIN medico m ON p.num_cedula = m.num_cedula
WHERE EXTRACT(YEAR FROM v.data_registo) = EXTRACT(YEAR FROM current_date)
AND LOWER(c.nome) = 'arouca' AND LOWER(p.substancia) = 'aspirina'
) AS result
WHERE nome = medico.nome
);
SELECT a.num_doente FROM analise a
WHERE EXTRACT(MONTH FROM a.data_registo) = EXTRACT(MONTH FROM current_date)
AND NOT EXISTS (
SELECT p.num_doente
FROM prescricao_venda p NATURAL JOIN venda_farmacia v
WHERE EXTRACT(MONTH FROM data_registo) = EXTRACT(MONTH FROM current_date)
AND p.num_doente = a.num_doente
);
```