**Projeto de Bases de Dados**

**Parte 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Grupo:** 50 | | | |
| **Número** | **Nome** | **Contribuição** | **Esforço (em Horas)** |
| 93695 | Catarina Sofia dos Santos Sousa | 33% | 3h |
| 93743 | Nelson Alexandre Geada Trindade | 34% | 3h |
| 93754 | Rodrigo Rodrigues Major | 33% | 3h |

**Sala:** 1-29

**Turno:** 4ºFeira – 15h00

**Docente:** Carlota De Oliveira Lopes Dias

**Criação da Base de Dados**

-- #### CREATE TABLES ####

-- AUXILIAR Tables

CREATE TABLE nome\_concelho (nome varchar(24) NOT NULL UNIQUE PRIMARY KEY);

CREATE TABLE nome\_regiao (nome varchar(8) NOT NULL UNIQUE PRIMARY KEY);

CREATE TABLE tipo\_instituicao (tipo varchar(11) PRIMARY KEY);

REVOKE ALL PRIVILEGES ON Table nome\_concelho FROM public;

REVOKE ALL PRIVILEGES ON Table nome\_regiao FROM public;

REVOKE ALL PRIVILEGES ON Table tipo\_instituicao FROM public;

-- Principal Tables

CREATE TABLE regiao (

  num\_regiao int PRIMARY KEY,

  nome varchar(8) NOT NULL UNIQUE,

  num\_habitantes double precision,

  UNIQUE(num\_regiao, nome),

  FOREIGN KEY (nome) REFERENCES nome\_regiao(nome)

);

CREATE TABLE concelho (

  num\_concelho int,

  num\_regiao int,

  nome varchar(24),

  num\_habitantes double precision,

  PRIMARY KEY (num\_concelho, num\_regiao),

  FOREIGN KEY (num\_regiao) REFERENCES regiao(num\_regiao),

  FOREIGN KEY (nome) REFERENCES nome\_concelho(nome)

);

REVOKE ALL PRIVILEGES ON Table regiao FROM public;

REVOKE ALL PRIVILEGES ON Table concelho FROM public;

CREATE TABLE instituicao (

  nome varchar(255) PRIMARY KEY,

  tipo varchar(11) NOT NULL,

  num\_regiao int NOT NULL,

  num\_concelho int NOT NULL,

  FOREIGN KEY (num\_concelho,num\_regiao) REFERENCES concelho(num\_concelho,num\_regiao),

  FOREIGN KEY (tipo) REFERENCES tipo\_instituicao(tipo)

);

CREATE TABLE medico (

  num\_cedula int PRIMARY KEY,

  nome varchar(255),

  especialidade varchar(25)

);

CREATE TABLE consulta (

  num\_cedula int,

  num\_doente int,

  data date,

  nome\_instituicao varchar(255),

  CHECK (EXTRACT(DOW from data) < 6), --RI-consulta-1

  UNIQUE (num\_doente, data, nome\_instituicao), --RI-consulta-2

PRIMARY KEY (num\_cedula, num\_doente, data),

  FOREIGN KEY (num\_cedula) REFERENCES medico(num\_cedula),

  FOREIGN KEY (nome\_instituicao) REFERENCES instituicao(nome)

);

CREATE TABLE prescricao (

  num\_cedula int,

  num\_doente int,

  data date,

  substancia varchar(255),

  quant int,

  PRIMARY KEY (num\_cedula, num\_doente, data, substancia),

  FOREIGN KEY (num\_cedula, num\_doente, data)

    REFERENCES consulta(num\_cedula, num\_doente, data)

);

CREATE TABLE analise (

  num\_analise int PRIMARY KEY,

  especialidade varchar(25),

  num\_cedula int,

  num\_doente int,

  data date,

  data\_registo date,

  nome varchar(255),

  quant int,

  inst varchar(255),

  FOREIGN KEY (num\_cedula, num\_doente, data)

    REFERENCES consulta(num\_cedula, num\_doente, data),

  FOREIGN KEY (inst) REFERENCES instituicao(nome)

  --RI: 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 int PRIMARY KEY,

  data\_registo date,

  substancia varchar(255),

  quant int,

  preco int,

  inst varchar(255),

  FOREIGN KEY (inst) REFERENCES instituicao(nome)

);

CREATE TABLE prescricao\_venda (

  num\_cedula int,

  num\_doente int,

  data date,

  substancia varchar(255),

  num\_venda int,

  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)

);

**Consultas em SQL**

**Query 1**

-- QUERY 1

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

SELECT i.num\_concelho

FROM instituicao i

    INNER JOIN venda\_farmacia v

    ON v.inst = i.nome

WHERE v.data\_registo = CURRENT\_DATE

GROUP BY i.num\_concelho

HAVING SUM(preco) >= all(

    SELECT SUM(preco)

    FROM instituicao i

        INNER JOIN venda\_farmacia v

        ON i.nome = v.inst

    WHERE v.data\_registo = CURRENT\_DATE

    GROUP BY num\_concelho

);

**Query 2**

-- QUERY 2

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

SELECT num\_regiao, num\_cedula

FROM prescricao

    NATURAL JOIN consulta

    INNER JOIN instituicao

    ON consulta.nome\_instituicao = instituicao.nome

WHERE prescricao.data >= '2019/01/01' and prescricao.data <= '2019/06/30'

GROUP BY num\_regiao, num\_cedula

HAVING COUNT(data) >= all(

    SELECT COUNT(data)

    FROM prescricao

    NATURAL JOIN consulta

    INNER JOIN instituicao

    ON consulta.nome\_instituicao = instituicao.nome

    WHERE prescricao.data >= '2019/01/01' and prescricao.data <= '2019/06/30'

    GROUP BY num\_regiao, num\_cedula

);

**Query 3**

-- QUERY 3

-- Quais são os médicos que já prescreveram aspirina em receitas aviadas em todas as farmácias

-- do concelho de Arouca este ano?

SELECT prescricao\_venda.num\_cedula

FROM prescricao\_venda

    NATURAL JOIN venda\_farmacia

    INNER JOIN instituicao

    ON venda\_farmacia.inst = instituicao.nome

WHERE prescricao\_venda.substancia = 'aspirina'

    AND instituicao.tipo = 'farmacia'

    AND instituicao.num\_concelho = 34

    AND EXTRACT(YEAR from prescricao\_venda.data) = EXTRACT(YEAR from CURRENT\_DATE)

GROUP BY prescricao\_venda.num\_cedula

HAVING COUNT(prescricao\_venda.num\_cedula) >= all(

    SELECT COUNT(instituicao.nome)

    FROM instituicao

    WHERE instituicao.tipo = 'farmacia'

        AND instituicao.num\_concelho = 34

    GROUP BY instituicao.nome

);

**Query 4**

-- QUERY 4

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

SELECT DISTINCT num\_doente

FROM analise

WHERE EXTRACT(MONTH from data) = EXTRACT(MONTH from CURRENT\_DATE)

    AND EXTRACT(YEAR from data) = EXTRACT(YEAR from CURRENT\_DATE)

    AND num\_doente NOT IN

        (

        SELECT num\_doente

        FROM prescricao\_venda

        WHERE EXTRACT(MONTH from data) = EXTRACT(MONTH from CURRENT\_DATE)

            AND EXTRACT(YEAR from data) = EXTRACT(YEAR from CURRENT\_DATE)

        );

**Arquitetura da Aplicação HTML (Flask)**

Temos de colocar alguma cena aqui