

Bases de Dados 2020/2021

Enunciado do projecto

Parte 3

A terceira parte do projeto da disciplina de Bases de Dados consiste na realização da base de dados do *Sistema de Informação ODISSEIA* no SGBD POSTGRES passando pelo desenvolvimento do script de criação de tabelas, consultas SQL, restrições de integridade e criação de um protótipo de aplicação web.

Criação e Preenchimento da Base de Dados

Usando a linguagem SQL, apresente um ficheiro com as instruções (*statements*) para criar o esquema de base de dados no SGBD Postgres correspondente ao esquema relacional **apresentado no Anexo A¹**.

Os tipos de dados escolhidos para cada atributo devem ser os mais apropriados. Em particular, no que diz respeito aos tipos de dados e tamanhos dos campos. A utilização de caracteres acentuados e cedilhas deve ser evitada.

Devem ser também especificadas as **restrições de integridade correspondentes às chaves primárias e estrangeiras presentes** em cada tabela, assim como todas as restrições de integridade que possam ser **definidas sem recurso a extensões procedimentais** (*Stored Procedures* ou *Triggers*). Devem identificar as restrições que necessitam de extensões procedimentais, mas não é necessário defini-las.

Uma vez criada, a base de dados deve ser preenchida, de forma consistente, com os registos necessários em cada tabela de forma a assegurar que todas as interrogações SQL, solicitadas adiante, tem **resultado não vazio**. A criação de registos e o carregamento da base de dados podem ser realizados através do método que entenda ser mais adequado (manualmente, folha Excel, script SQL, Python, ou outro).

SQL

Apresente, a consulta SQL² mais sucinta correspondente a cada uma das seguintes consultas:

1. Qual o concelho onde se fez o maior volume de vendas hoje?
2. Qual o médico que mais prescreveu no 1º semestre de 2019 em cada região?
3. Quais são os médicos que já prescreveram aspirina em receitas aviadas em todas as farmácias do concelho de Arouca este ano?

¹ O esquema é apenas inspirado no modelo da Parte 2, não é uma solução do enunciado anterior.

² Não podem ser utilizadas instruções SQL não façam parte do standard (tais como a instrução `LIMIT`).

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

Desenvolvimento da Aplicação

Crie um conjunto de páginas em Python e HTML simples que permita ao utilizador:

- a) Inserir, editar e remover instituições e médicos
- b) Inserir, editar e remover prescrições e análises
- c) Realizar vendas em farmácia com e sem prescrição (se o cliente tem prescrição, aceder à base de dados para obter os dados da prescrição e registar (inserir) na prescricao_venda além de na venda_farmacia, caso contrário, registar a venda apenas em venda_farmacia).
- d) Listar as substância prescritas por um médico num dado mês do ano
- e) Listar os valores de glicémia mais alto e mais baixo em cada concelho e respectivo doente.

A solução deve primar pela segurança, prevenindo ataques via SQL INJECTION. Adicionalmente, deve garantir-se a **atomicidade das operações actualização** da base de dados.

Relatório

O projeto será avaliado a partir do relatório entregue pelos alunos e pela discussão. O relatório deverá conter todas as respostas aos itens pedidos acima. Na tabela seguinte indica-se a valorização de cada parte do trabalho a desenvolver.

Item	Valores
Criação da Base de Dados	4
SQL	10
Aplicação	6

O relatório deverá começar com uma folha de rosto com a indicação “**Projeto de Bases de Dados, Parte 3**”, o **nome e número dos alunos**, **a percentagem relativa de contribuição de cada um, juntamente com o esforço (em horas)** que cada elemento do grupo dedicou ao projeto, o **número do grupo**, o **turno** a que o grupo pertence, o **nome do docente de laboratório** e, além da folha de rosto, o relatório deverá ter no máximo **6 páginas**.

Entrega

A entrega no sistema fénix deve ser um ficheiro **zip** estruturado da seguinte forma:

relatorioGG.pdf (onde GG é o número do grupo)	O relatório em pdf onde GG é o número do grupo, contendo os comandos de criação da base de dados , as consultas em SQL e uma explicação da arquitetura da aplicação PHP e das relações entre os diversos ficheiros . Não deve incluir as instruções de população da base de dados.
schema.sql	Ficheiro de criação do esquema da base de dados.
queries.sql	Ficheiro com as consultas SQL.
populate.sql	Ficheiro com instruções para preencher as tabelas com dados de teste.
web/	Pasta com os ficheiros HTML e Python.

O trabalho terá que ser entregue em duas versões:

1. **Versão digital**, em formato ZIP com nome entrega-03-GG.zip³ (onde **GG** é o número do grupo), a entregar via Fénix até às 23h59 da data de entrega.
2. **Os grupos podem optar por entregar a pasta Web mais tarde, juntamente com a Entrega 4.**
3. **Versão em papel**, a entregar na aula de laboratório seguinte, caso requerido pelo docente do turno. O código Python não deve ser impresso.

³ △ O formato do ficheiro deve ser exclusivamente ZIP ou GZ. Outros formatos de arquivo (tais como RAR) não serão aceites.

Anexo A

Modelo Relacional⁴

regiao(num_regiao, nome, num_habitantes)

RI-regiao-1: nome = {Norte, Centro, Lisboa, Alentejo, Algarve}

Nota: é aceite que se pressuponha que as tabelas regiao e concelho apenas possam ser lidas pelos utilizadores, sendo preenchidas uma única vez pelo administrador da aplicação.

concelho(num_concelho, num_regiao, nome, num_habitantes)

num_regiao: FK regiao (num_regiao)

RI-concelho-1: nome = {concelhos de portugal continental}

instituicao(nome, tipo, num_regiao, num_concelho)

num_regiao, num_concelho: FK concelho (regiao, concelho)

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

medico(num_cedula, nome, especialidade)

consulta(num_cedula, num_doente, data, nome_instituicao)

num_cedula: FK medico (num_cedula)

nome_instituicao: FK instituicao (nome)

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

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

prescricao(num_cedula, num_doente, data, substancia, quant)

num_cedula, num_doente, data: FK consulta (num_cedula, num_doente, data)

analise(num_analise, especialidade, num_cedula, num_doente, data, data_registo, nome, quant, inst)

num_cedula, num_doente, data: FK consulta (num_cedula, num_doente, data)

inst: FK instituicao (nome)

RI-analise: a consulta associada pode estar omissa; não estando, a especialidade da consulta tem de ser igual à do médico.

venda_farmacia(num_venda, data_registo, substancia, quant, preco, inst)

inst: FK instituicao (nome)

prescricao_venda(num_cedula, num_doente, data, substancia, num_venda)

num_venda: FK venda_farmacia (num_venda)

num_cedula, num_doente, data, substancia: FK prescricao (num_cedula, num_doente, data, substancia)

⁴ Há uma ligeira mudança de notação das chaves estrangeiras relativamente à Parte 2: a,b: FK A(x,y) significa que a,b é chave estrangeira, com a referenciar x, e b y.