

# Bases de Dados 2020/2021

## Enunciado do projecto

### Parte 4

A quarta parte do projeto da disciplina de Bases de Dados consiste na optimização da base de dados do *Sistema de Informação ODISSEIA* no SGBD POSTGRES passando pela codificação de regras de integridade procedimental e implementação de um Data Warehouse..

## Restrições de Integridade

Escreva o código que implemente, usando as extensões procedimentais do SQL (*Stored Procedures* e *Triggers*) no esquema que elaborou na Parte 3, as seguintes restrições de integridade:

**RI-100:** um médico não pode dar mais de 100 consultas por semana na mesma instituição

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

## Índices

Baseando-se no modelo relacional do enunciado da Parte 3, crie o(s) índice(s) em SQL que considerar necessários para cada um dos casos indicados em seguida.

Suponha que não existe qualquer índice sobre as tabelas para além dos implícitos na declaração das chaves primárias e estrangeiras. Não proponha índices novos se puder usar estes índices implícitos. Se for necessário, pode alterar a ordem dos campos das chaves primárias na declaração da tabela para que o índice associado à chave primária sirva também para otimizar a query pretendida. Caso o índice que considere mais apropriado não seja suportado pelo SGBD, apresente-o na mesma e de seguida apresente a alternativa que propõe.

Indique que tipo de índice(s), sobre que atributo(s) e sobre que tabela(s) faria sentido criar (ou usar os que já existem) de modo a acelerar a execução de cada uma das interrogações.

Justifique as suas escolhas e pressupostos, nomeadamente explicando porque é que cada um dos índices que está a propor é mais adequado que os que já existem pela criação das chaves primárias e estrangeiras.

Suponha que a dimensão das tabelas ultrapassa em várias ordens de grandeza a memória disponível.

Queries:

1. Listar as datas de consulta de um doente:

```
select data from consulta where num_doente = <um_valor>
```

2. Considere que há apenas seis especialidades: “E1” a “E6”. Pretende-se saber quantos médicos existem de cada especialidade.

```
select count (*) from medico where especialidade = “Ei”
```

em que Ei é uma das seis especialidades.

3. Nomes dos médicos de uma determinada especialidade. Para a resolução desta alínea considere, para além do referido sobre a dimensão das tabelas, os seguintes aspetos:
  - 1- Os blocos do disco são de 2K bytes e cada registo na tabela ocupa 1K bytes.
  - 2- Os médicos estão uniformemente distribuídos pelas 6 especialidades.

```
select nome from medico where especialidade = ‘Ei’  
em que Ei é uma das seis especialidades.
```

4. Listar os nomes dos médicos que deram consultas entre duas datas.

```
select nome from medico, consulta  
where consulta.num_celula=medico.num_celula AND  
consulta.data BETWEEN ‘data_1’ AND ‘data_2’
```

em que ‘data\_1’ e ‘data\_2’ são duas datas.

## Modelo Multidimensional

Crie na base de dados um esquema em estrela com informação relativa a prescrições e dados da sua venda em farmácia (f\_presc\_venda) e análises laboratoriais (f\_analise) tendo as dimensões abaixo:

d\_tempo(id\_tempo, dia, dia\_da\_semana, semana, mes, trimestre, ano)

d\_instituicao(Id\_inst, nome, tipo, num\_regiao, num\_concelho )

nome: FK(instituicao); num\_regiao: FK(regiao); num\_concelho:: FK(concelho)

f\_presc\_venda(id\_presc\_venda, id\_medico, num\_doente, id\_data\_registo, id\_inst, substancia, quant)

Id\_presc\_venda: FK(prescricao) OU FK(presc\_venda.num\_venda)

id\_medico: FK(medico)

Id\_data\_registo: FK(d\_tempo)

Id\_inst: FK(d\_instituicao)

f\_analise(id\_analise, id\_medico, num\_doente, id\_data\_registo, id\_inst, nome, quant)

Id\_analise: FK(analise); id\_medico: FK(medico)

Id\_data\_registo: FK(d\_tempo)

Id\_inst: FK(d\_instituicao)

Escreva as instruções SQL necessárias para definir e carregar o esquema em estrela a partir das tabelas existentes, carregando primeiro as tabelas de dimensões e depois a tabela de factos. Os atributos **Id\_presc\_venda**, **Id\_data\_registo** e **Id\_inst** são chaves substitutas. Uma forma expedita de implementar a chave substituta **Id\_presc\_venda** consiste em usar o valor do atributo **num\_venda** da tabela **presc\_venda** (já definida na E3); nesse caso deverão também dizer que **num\_venda** é **UNIQUE**.

## Data Analytics

Considerando o esquema em estrela criado na questão anterior, escreva uma interrogação SQL que permita analisar:

- O **número de análises de glicémia** realizadas por especialidade médica, por mês e por ano, **com totais parciais em cada especialidade cada mês e cada ano**, em 2017-2020
- A **quantidade total** e **nº médio de prescrições diário** de cada substância registados [REDACTED], **com rollup no espaço (concelho) e no tempo (dia da semana e mês)** [REDACTED] **mostrando o resultado para** a região de Lisboa durante o 1º trimestre de 2020.

A solução apresentada deverá recorrer às instruções **ROLLUP**, **CUBE**, **GROUPING SETS**, ou à união ("UNION") de cláusulas **GROUP BY**.

## 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
Restrições de integridade	5
Índices	6
Modelo Multidimensional	3
Data analytics	6

O relatório deverá começar com uma folha de rosto com a indicação "**Projeto de Bases de Dados, Parte 4**", 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 <b>GG</b> é o número do grupo, com <b>as RI</b> , os <b>índices e sua justificação</b> , comandos de criação do <b>modelo multidimensional</b> e <b>ETL de carregamento</b> , e <b>queries OLAP</b> .
RI.sql	Ficheiro de criação das restrições de integridade
star_schema.sql	Ficheiro de criação do esquema multidimensional
etl.sql	Ficheiro com o script de carregamento do modelo em estrela
olap.sql	Ficheiro com as consultas OLAP
E3-arquitetura.pdf (opcional)	<b>Ficheiro com explicação da arquitetura da aplicação PHP</b> e das <b>relações entre os diversos ficheiros</b> , caso não tenham entregue com E3
web/ (opcional)	Pasta com os ficheiros HTML e Python da E3 caso não tenham entregue com E3

O trabalho terá que ser entregue em duas versões:

1. Versão digital, em formato ZIP com nome entrega-04-GG.zip (onde GG é o número do grupo), a entregar via Fénix até às 23h59 da data de entrega.
2. Versão em papel, a entregar na aula de laboratório seguinte, caso requerido pelo docente do turno. O código Python não deve ser impresso.