

# **Projeto de Bases de Dados, Parte 4**

Grupo 09

16:30 Quarta-Feira – Prof. Carlota Dias

Miguel Silva, ist193739 – 6 Horas

António Venâncio, ist193689 – 6 Horas

Nuno Carvalho, ist193744 – 7 Horas

Índices

```

/**
 * 1.
 * Listar as datas de consulta de um doente
 */

/* RESPOSTA:
Criar um índice para organizar a coluna num_doente da tabela consulta, utilizando o já existente num_doente,
de forma a tornar mais eficiente a comparação direta "num_doente = <valor>"
*/

create index indice1 on consulta(num_doente);

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

/* RESPOSTA:
Criar um índice para organizar os médicos por diferentes especialidades usando uma hashtable,
desta forma, acedemos diretamente a uma lista com todos os médicos da especialidade pretendida
*/

create index indice2 on medico using hash(especialidade);

/**
 * 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.
 */

/* RESPOSTA:
É possível conter 2 registos por cada bloco
Desta forma, e sabendo que os médicos estão igualmente distribuídos,
podemos criar uma hashtable na coluna das especialidades para acelerar o acesso
*/

create index indice3 on medico using hash(especialidade);

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

/* RESPOSTA:
Utilizar um índice BTREE na coluna data_consulta da tabela consulta, de forma a otimizar
a seleção entre duas datas.
Para facilitar a operação "consulta.num_cedula = medico.num_cedula" cria-se um
índice em ambas colunas.
*/

create index index4 on consulta using BTREE(data_consulta);
create index index5 on consulta(num_cedula);
create index index6 on medico(num_cedula);

```

# Modelo Multidimensional

```
1  /* Eliminação das tabelas */
2
3  drop table if exists d_tempo cascade;
4  drop table if exists d_instituicao cascade;
5  drop table if exists f_presc_venda cascade;
6  drop table if exists f_analise cascade;
7
8
9  /* Definição das tabelas */
10
11  create table d_tempo(
12      id_tempo serial not null,
13      dia smallint check (dia < 32 and dia > 0),
14      dia_da_semana smallint check (dia_da_semana < 8 and dia_da_semana > 0),
15      semana smallint check (semana < 54 and semana > 0),
16      mes smallint check (mes < 13 and mes > 0),
17      trimestre smallint check (trimestre < 5 and trimestre > 0),
18      ano integer not null,
19      primary key(id_tempo)
20  );
21
22  create table d_instituicao(
23      id_inst serial not null,
24      nome char(30) not null,
25      tipo char(11) not null,
26      num_regiao integer not null,
27      num_concelho integer not null,
28      primary key(id_inst),
29      foreign key(nome) references instituicao(nome),
30      foreign key(num_regiao) references regiao(num_regiao),
31      foreign key(num_concelho) references concelho(num_concelho)
32  );
33
34  create table f_presc_venda(
35      id_presc_venda integer not null,
36      id_medico integer not null,
37      num_doente integer not null,
38      id_data_registo integer not null,
39      id_inst integer not null,
40      substancia char(30) not null,
41      quant integer not null,
42      primary key(id_presc_venda),
43      foreign key(id_presc_venda) references prescricao_venda(num_venda),
44      foreign key(id_medico) references medico(num_cedula),
45      foreign key(id_data_registo) references d_tempo(id_tempo),
46      foreign key(id_inst) references d_instituicao(id_inst)
47  );
48
49  create table f_analise(
50      id_analise integer,
51      id_medico integer not null,
52      num_doente integer not null,
53      id_data_registo integer not null,
54      id_inst integer not null,
55      nome char(20) not null,
56      quant integer not null,
57      primary key(id_analise),
58      foreign key(id_analise) references analise(num_analise),
59      foreign key(id_medico) references medico(num_cedula),
60      foreign key(id_data_registo) references d_tempo(id_tempo),
61      foreign key(id_inst) references d_instituicao(id_inst)
62  );
```

# Restrições de Integridade

```
/**
 * 1.
 * RI-100: um médico não pode dar mais de 100 consultas por semana na mesma instituição
 */
drop trigger if exists max_consultas_trigger on consulta;

create or replace function max_consultas() returns trigger as $$
begin
    if (select count(*) from consulta where
        consulta.num_cedula == new.num_cedula and
        consulta.nome_instituicao == new.nome_instituicao and
        consulta.data_consulta
        group by weekly
        having count(*) > 100) != null

        raise exception 'O Médico % excedeu o número máximo de consultas possíveis semanais.', new.num_cedula;
    end if;

    return new;

End;
$$ language plpgsql;

create trigger max_consultas_trigger after update on consulta
for each row execute procedure max_consultas();

/**
 * 2.
 * RI-análise: numa análise, a consulta associada pode estar omissa; não estando, a especialidade
 * da consulta tem de ser igual à do médico.
 */
drop trigger if exists check_analise_consulta_trigger on analise;

create or replace function check_analise_consulta() returns trigger as $$
begin
    if (new.num_cedula != null and new.num_doente != null and new.data_consulta != null)
        if new.especialidade not in (select especialidade from medico where medico.num_cedula == new.num_cedula)
            raise exception 'A especialidade % da analise não é a mesma que a do médico que a realizou.', new.especialidade;
        end if;
    end if;

    return new;

End;
$$ language plpgsql;

create trigger check_analise_consulta_trigger after update on analise
for each row execute procedure check_analise_consulta();
```

# Data Analytics

```
1 SELECT especialidade, mes, ano, SUM(id_analise) as numAnalises
2 FROM (f_analise NATURAL JOIN medico NATURAL JOIN d_tempo)
3 WHERE nome = 'glicemia' AND
4        ano <= 2020 AND
5        a >= 2017
6 GROUP BY
7        CUBE (especialidade, mes, ano)
```

## Esquema Web

