



PROJETO DE BASES DE DADOS – PARTE 4

Grupo3 - Quarta Feira 16:30 (Prof. Carlota Dias)

<i>Número</i>	<i>Nome</i>	<i>Percentagem</i>	<i>Esforço</i>
93694	Carolina Ramos	33%	14 horas
90118	João Diegues	33%	14 horas
93740	Miguel Oliveira	33%	14 horas

Restrições de Integridade

/*1-RI-100: um médico não pode dar mais de 100 consultas por semana na mesma instituição*/

```
create or replace function RI_100() returns trigger as $$
begin
    if new.num_cedula not in (
        SELECT num_cedula
        FROM( SELECT num_cedula, week, count(week)
              FROM ( SELECT num_cedula, extract(week from data) as week, extract(year from data) as year
                    FROM consulta) AS tabela1 group by num_cedula, week ) as tabela2
        WHERE tabela2.count <101 and extract(week from new.data) = tabela2.week)
    then
        raise exception 'O Médico % excedeu o número de consultas.',new.num_cedula;
    end if;
    return new;
End;
$$ Language plpgsql;
create trigger RI_100_trigger before insert on consulta for each row execute procedure RI_100();
```

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

```
create or replace function RI_analise() returns trigger as $$
declare especialidadeMedico varchar(100);
begin
    if new.num_cedula is not null then
        SELECT especialidade into especialidadeMedico
        FROM medico as m
        WHERE new.num_cedula = m.num_cedula;
        if new.especialidade != especialidadeMedico then
            raise exception 'O Médico % não tem a mesma especialidade que a
analise.',new.num_cedula;
        end if;
    end if;
    return new;
End;
$$ Language plpgsql;

create trigger RI_analise_trigger before insert on analise for each row execute procedure RI_analise();
```

Índices

- 1- Para facilitar a comparação direta “num_doente = <valor>” cria-se um índice para organizar a coluna num_doente numa Hash Table, assim a procura relativa ao num_doente torna-se mais eficiente. Uma vez que é feita apenas uma igualdade, não é necessário uma Btree, logo a Hash Table será suficiente. Apesar de o num_doente ser parte da chave primária faz sentido criar o índice, visto que a chave da consulta é composta.

create index indice_num_doente on consulta using hash(num_doente);

- 2- Para aumentar a eficiência da query a melhor opção seria criar um índice para organizar a coluna especialidade numa Hash Table isto permite uma contagem de médicos, que pertencem a essa especialidade, mais eficiente.

create index indice_especialidade on medico using hash(especialidade);

- 3- Tendo em conta que se mantém o índice criado anteriormente ‘indice_especialidade’ e tendo em conta os aspetos referidos, cria-se um segundo índice auxiliar para organizar as colunas especialidade, nome do medico numa Btree. A criação do índice neste caso é mesmo necessário, uma vez que nenhum dos atributos pertence a uma chave primária. A ordem de inserção na Btree depende da seletividade dos atributos. Se especialidade tiver mais seletividade,

create index indice_aux on medico using Btree(especialidade, nome);

Caso contrário,

create index indice_aux on medico using Btree(nome, especialidade);

- 4- Para tornar a listagem mais eficiente cria-se dois índices um para organizar a coluna num_cedula numa Hash Table na tabela consulta e outro para organizar as datas numa Btree na tabela consulta. A utilização da Btree para as datas é a melhor opção uma vez que são usadas funções de comparação que não são suportadas pela Hash Table.

create index indice_cedula on consulta using Hash(num_cedula);

create index indice_data on consulta using Btree(data);

Modelo Multidimensional

```
drop table d_tempo CASCADE;  
drop table d_instituicao CASCADE;  
drop table f_prescr_venda CASCADE;  
drop table f_analise CASCADE;  
drop table fact_table CASCADE;
```

```
create table d_tempo(  
  id_tempo serial not null unique,  
  dia integer check (dia >=1 and dia <=31),  
  dia_da_semana integer check (dia_da_semana >=1 and dia_da_semana <=7),  
  semana integer check (semana >=1),  
  mes integer check (mes >=1 and mes <= 12),  
  trimestre integer check (trimestre >=1 and trimestre <=4),  
  ano integer,  
  constraint pk_d_tempo primary key(id_tempo)  
);
```

```
create table d_instituicao(  
  id_inst serial not null unique,  
  nome char(100),  
  tipo char(100),  
  num_regiao integer,  
  num_concelho integer,  
  constraint pk_d_instituicao primary key(id_inst),  
  constraint fk_d_instituicao_instituicao foreign key(nome) references  
instituicao(nome),  
  constraint fk_d_instituicao_regiao foreign key(num_regiao) references  
região(num_regiao),  
  constraint fk_d_instituicao_concelho foreign key(num_concelho) references  
concelho(num_concelho)  
);
```

```
create table f_presc_venda(  
  id_presc_venda integer unique,  
  id_medico integer,  
  num_doente integer,  
  id_data_registo integer,  
  id_inst integer,  
  substancia char(50),  
  quant integer check(quant>0),  
  constraint pk_f_presc_venda primary key(id_presc_venda),  
  constraint fk_f_presc_venda_prescricao_venda foreign key(id_presc_venda) references  
prescricao_venda(num_venda),  
  constraint fk_f_presc_venda_medico foreign key (id_medico) references  
medico(num_cedula),  
  constraint fk_f_presc_venda_d_tempo foreign key (id_data_registo) references  
d_tempo(id_tempo),
```

```

        constraint fk_f_presc_venda_d_instituicao foreign key(id_inst) references
d_instituicao(id_inst)
);

create table f_analise(
    id_analise integer unique,
    id_medico integer,
    num_doente integer,
    id_data_registro integer,
    id_inst integer,
    nome char(100),
    quant integer,
    constraint pk_f_analise primary key(id_analise),
    constraint fk_f_analise_analise foreign key (id_analise) references analise(num_analise),
    constraint fk_f_analise_medico foreign key (id_medico) references medico(num_cedula),
    constraint fk_f_analise_d_tempo foreign key (id_data_registro) references
d_tempo(id_tempo),
    constraint fk_f_analise_d_instituicao foreign key (id_inst) references d_instituicao(id_inst)
);

create table fact_table(
    id_presc_venda integer unique,
    id_analise integer unique,
    id_tempo serial not null unique,
    id_inst serial not null unique
    constraint pk_fact_table primary key( id_presc_venda, id_analise, id_tempo, id_inst),
    constraint fk_fact_table_f_presc_venda foreign key (id_presc_venda) references
f_presc_venda(id_presc_venda) ON DELETE CASCADE ON UPDATE CASCADE,
    constraint fk_fact_table_f_analise foreign key (id_analise) references
f_analise(id_analise) ON DELETE CASCADE ON UPDATE CASCADE,
    constraint fk_fact_table_d_tempo foreign key (id_tempo) references
d_tempo(id_tempo) ON DELETE CASCADE ON UPDATE CASCADE,
    constraint fk_fact_table_d_instituicao foreign key (id_inst) references
d_instituicao(id_inst) ON DELETE CASCADE ON UPDATE CASCADE
)

```

ETL de carregamento

```
insert into d_instituicao(nome, tipo, num_regiao, num_concelho)
select nome, tipo, num_regiao, num_concelho
from instituicao NATURALJOIN regiao NATURALJOIN concelho;
```

```
insert into d_tempo(dia, dia_da_semana, semana, mes, trimestre, ano)
select extract(DAY from data) as dia, extract(DOW from data) as dia_da_semana,
extract(WEEK from data) as semana, extract(MONTH from data) as mes, extract(QUARTER
from data) as trimestre, extract(YEAR from data) as ano
from prescricao_venda;
```

```
insert into d_tempo(dia, dia_da_semana, semana, mes, trimestre, ano)
select extract(DAY from data_registro) as dia, extract(DOW from data_registro) as
dia_da_semana, extract(WEEK from data_registro) as semana, extract(MONTH from
data_registro) as mes, extract(QUARTER from data_registro) as trimestre, extract(YEAR from
data_registro) as ano
from analise;
```

```
insert into f_presc_venda(num_cedula, num_doente, id_data_registro, id_inst, substancia,
quant)
select num_cedula, num_doente, id_tempo as id_data_registro, id_inst, nome, quant
from prescricao NATURALJOIN d_tempo NATURALJOIN d_instituicao;
```

```
insert into f_analise(num_cedula, num_doente, id_data_registro, id_inst, nome, quant)
select num_cedula, num_doente, id_data_registro, id_inst, nome, quant
from analise NATURALJOIN d_tempo NATURALJOIN d_instituicao;
```

```
insert into fact_table(id_presc_venda, id_analise, id_tempo, id_inst)
select id_presc_venda, id_analise, id_tempo, id_inst
from f_presc_venda NATURALJOIN f_analise NATURALJOIN d_tempo NATURALJOIN
d_instituicao;
```

Queries OLAP

/*1- O número de análises de glicémia realizadas por especialidade médica, por mês e por ano em 2017-2020*/

```
SELECT d.ano, d.mes, a.especialidade, COUNT(id_analise)
```

```
FROM f_analise as a INNER JOIN d_tempo as t on a.id_data_registo = d.id_tempo
```

```
WHERE d.ano > 2016 and d.ano < 2021 group by cube(a.especialidade, d.mes, d.ano); /*2- A quantidade total e nº médio de prescrições diário de cada substância registados em cada dia da semana em instituições da região de Lisboa e Vale do Tejo durante o 1º trimestre de 2020.*/
```

```
SELECT i.num_concelho, tabela.mes, tabela.dia_da_semana, COUNT(substancia)
```

```
FROM (f_presc_venda as p INNER JOIN d_tempo as t on p.id_data_registo=d.id_tempo) as  
tabela NATURAL JOIN d_instituicao as i
```

```
WHERE i.num_regiao=3 and tabela.trimestre=1 group by rollup (i.num_concelho, tabela.mes,  
tabela.dia_da_semana)
```