

## Projeto de Bases de Dados – Parte 4

Trabalho realizado por:

Nome	Número	Esforço	Contribuição
Afonso Jorge	93680	12h	33%
Bernardo Quinteiro	93692	12h	33%
Diogo Lopes	93700	12h	33%

**Grupo 38**

**Turno**

Quarta 16h30 – 18h00

**Docente**

Carlota Dias

## RI

create or replace function check\_especialidade\_analise() returns trigger as \$\$

begin

if new.num\_doente is not null and new.num\_cedula is not null and new.data is not null then

if (select especialidade from medico where num\_cedula = new.num\_cedula) =  
new.especialidade then

return new;

else

raise exception 'A especialidade da analise nao e a mesma da do medico';

end if;

end if;

return new;

end;

\$\$ language plpgsql;

create trigger analise\_trigger before insert on analise for each row execute procedure  
check\_especialidade\_analise();

create trigger analise\_trigger2 before update on analise for each row execute procedure  
check\_especialidade\_analise();

create or replace function check\_consulta() returns trigger as \$\$

begin

if (select count(\*) from consulta where num\_cedula = new.num\_cedula and  
nome\_instituicao = new.nome\_instituicao and extract(week from data) = extract(week from  
new.data)) < 100 then

return new;

else

raise exception 'Medico com mais de 100 consultas numa semana na mesma instituicao';

end if;

end;

\$\$ language plpgsql;

create trigger consulta\_trigger before insert on consulta for each row execute procedure  
check\_consulta();

create trigger consulta\_trigger2 before update on consulta for each row execute procedure  
check\_consulta();

## Star Schema

```
drop table d_tempo cascade;  
drop table d_instituicao cascade;  
drop table f_presc_venda cascade;  
drop table f_analise cascade;
```

```
CREATE TABLE d_tempo (  
    id_tempo serial unique,  
    dia int,  
    dia_da_semana int,  
    semana int,  
    mes int,  
    trimestre int,  
    ano int  
);
```

```
CREATE TABLE d_instituicao (  
    id_inst serial unique,  
    nome varchar,  
    tipo varchar,  
    num_regiao int,  
    num_concelho int,  
    FOREIGN KEY(nome) REFERENCES instituicao(nome) on delete cascade on update cascade,  
    FOREIGN KEY(num_regiao) REFERENCES regiao(num_regiao) on delete cascade on update  
cascade,  
    FOREIGN KEY(num_concelho) REFERENCES concelho(num_concelho) on delete cascade on  
update cascade  
);
```

```
CREATE TABLE f_presc_venda (  
    id_presc_venda serial unique,  
    id_medico int,  
    num_doente int,  
    id_data_registo int,  
    id_inst int,
```

```

    substancia varchar,
    quant int,
    PRIMARY KEY(id_presc_venda),
    FOREIGN KEY(id_presc_venda) REFERENCES prescricao_venda(num_venda) on delete
    cascade on update cascade,
    FOREIGN KEY(id_medico) REFERENCES medico(num_cedula) on delete cascade on update
    cascade,
    FOREIGN KEY(id_data_registro) REFERENCES d_tempo(id_tempo) on delete cascade on
    update cascade,
    FOREIGN KEY(id_inst) REFERENCES d_instituicao(id_inst) on delete cascade on update
    cascade
);

```

```

CREATE TABLE f_analise (
    id_analise serial unique,
    id_medico int,
    num_doente int,
    id_data_registro int,
    id_inst int,
    nome varchar,
    quant int,
    PRIMARY KEY(id_analise),
    FOREIGN KEY(id_analise) REFERENCES analise(num_analise) on delete cascade on update
    cascade,
    FOREIGN KEY(id_medico) REFERENCES medico(num_cedula) on delete cascade on update
    cascade,
    FOREIGN KEY(id_data_registro) REFERENCES d_tempo(id_tempo) on delete cascade on
    update cascade,
    FOREIGN KEY(id_inst) REFERENCES d_instituicao(id_inst) on delete cascade on update
    cascade
);

```

## ETL

```

insert into d_instituicao(nome, tipo, num_regiao, num_concelho)
select nome, tipo, num_regiao, num_concelho
from instituicao;

```

```

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(month from data)/4 as
trimestre, extract(year from data) as ano

from analise;

```

```

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(month from data)/4 as
trimestre, extract(year from data) as ano

from prescricao_venda;

```

```

insert into f_presc_venda(id_medico, num_doente, id_data_registro, id_inst, substancia, quant)

select num_cedula as id_medico, num_doente, id_tempo as id_data_registro, id_inst,
substancia, quant

from prescricao_venda inner join d_tempo on (extract(day from data) = dia and extract(DOW
from data) = dia_da_semana and extract(week from data) = semana and extract(month from
data) = mes and extract(month from data)/4 = trimestre and extract(year from data) = ano)

natural join prescricao natural join consulta natural join instituicao natural join d_instituicao;

```

```

insert into f_analise(id_medico, num_doente, id_data_registro, id_inst, nome, quant)

select num_cedula as id_medico, num_doente, id_tempo as id_data_registro, id_inst, nome,
quant

from analise inner join d_tempo on (extract(day from data) = dia and extract(DOW from data)
= dia_da_semana and extract(week from data) = semana and extract(month from data) = mes
and extract(month from data)/4 = trimestre and extract(year from data) = ano)

natural join consulta natural join instituicao natural join d_instituicao;

```

## Indices

```

/* 1st query */

```

```

create index index_num_cedula on consulta using hash(num_cedula);

```

```

/* Pode-se criar um índice hash, uma vez que se trata de uma igualdade simples,
em relação ao num_cedula da consulta, de modo a que a comparação "num_doente" =
<um_valor> se torne mais eficiente */

```

/\* 2nd query \*/

create index index\_especialidade on medico using hash(especialidade);

/\* Uma vez que se trata de uma igualdade podemos tratar com uma hash, de modo a tornar mais eficiente a comparação "especialidade" = "Ei" \*/

/\* 3rd query \*/

create index especialidade\_index on medico using btree(especialidade);

/\* Como o número de médicos por especialidade é o mesmo iriam-se fazer muitos acessos aos blocos do disco, o que pode ser evitado usando uma btree ordenando os médicos por especialidade \*/

/\* 4th query \*/

create index index\_num\_cedula\_consulta on consulta using hash(num\_cedula);

create index index\_num\_cedula\_medico on medico using hash(num\_cedula);

/\* Podem-se criar duas hashes nas tabelas medico e consulta de modo a tornar mais eficiente a comparação "consulta.num\_cedula = medico.num\_cedula".

Relativamente à condição "consulta.data BETWEEN 'data\_1' AND 'data\_2'"

podemos organizar a coluna data na tabela consulta, pelo que usamos uma btree visto que é mais eficiente para ranges (BETWEEN) \*/