

Projeto de Bases de Dados – Parte 4

Número	Nome	Esforço (horas)	Contribuição (%)
93729	João Caldeira	9	45
93730	João Cardoso	10	50
89893	Gonçalo Faria	3	5
Grupo 32			

Docente: Carlota De Oliveira Lopes Dias

Turno: Terça-feira, 08:30 – 10:00

Sala: 0 - 21



RI-análise:

```
create or replace function same_especialidades() returns trigger AS
$$
begin
    if (new.num_cedula IS NULL and
        new.num_doente IS NULL and
        new.data IS NULL)
    then return new;
    end if;
    if (new.especialidade NOT IN (SELECT especialidade
FROM medico
                                WHERE num_cedula = new.num_cedula))
    then raise exception'Especialidade do medico nao corresponde
a especialidade da consulta na analise %',new.num_analise;
    end if;
    return new;
End;
$$ Language plpgsql;
create trigger same_especialidades_trigger before insert OR UPDATE
ON analise for each row execute procedure same_especialidades();
```

RI-100:

```
create or replace function medico_consultas() returns trigger AS
$$
begin
    if new.num_cedula IN (SELECT num_cedula FROM(SELECT
num_cedula, count(num_cedula)
                                FROM consulta as c WHERE EXTRACT(WEEK FROM
c.data) = EXTRACT(WEEK FROM new.data)
                                AND EXTRACT(YEAR FROM c.data) =
EXTRACT(YEAR FROM new.data)
                                AND c.nome_instituicao =
new.nome_instituicao GROUP BY c.num_cedula HAVING
count(num_cedula) >= 100) AS foo
                                GROUP BY num_cedula)
    then
        raise exception'Medico % ja tem 100 consultas nessa
instituicao e nessa semana',new.num_cedula;
    end if;
return new;
End;
$$ Language plpgsql;
create trigger medico_consultas_trigger before insert on
consulta for each row execute procedure medico_consultas();
```

Índices:

1. Para a primeira *Querie* não é necessário utilizar índices ,dado que “num_doente” é *key* , podemos no entanto reorganizar a ordem das *keys para otimizar a procura*.
2. Para as *Queries* 2 e 3 podemos utilizar um Bitmap sobre os as especialidades dos médicos , esta opção é justificada pelo número de especialidades limitado e as exigências de memória em 3. Esta implementação não está disponível no PostgreSQL ,então vamos utilizar Btree com a seguinte implementação “CREATE INDEX medico_especialidade INDEX ON medico (especialidade);”.
3. Para a quarta *Querie* podemos utilizar uma *Btree* com chave de procura composta (num_cedula,data) devido às comparações entre consultas ,médicos e data .Com a seguinte implementação “CREATE INDEX consulta_medico_index ON consulta(num_cedula,data);”.

Para criar a base de dados deve-se alterar o num_venda para UNIQUE na presc_venda em schema.sql e executar os seguintes comandos nesta ordem: “\i schema.sql”(entrega 3), “\i RI.sql”, “\i star_schema.sql” , “\i populate.sql”(entrega 3), “\i etl.sql”.

Modelo Multidimensional:

```
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,
    dia integer NOT NULL,
    dia_da_semana integer NOT NULL,
    semana integer NOT NULL,
    mes integer NOT NULL,
    trimestre integer NOT NULL,
    ano integer NOT NULL,
    PRIMARY KEY(id_tempo));
CREATE TABLE d_instituicao (
    id_inst serial,
    nome varchar(80) NOT NULL,
    tipo varchar(80) NOT NULL,
    num_regiao integer,
```

```

        num_concelho integer,
        PRIMARY KEY(id_inst),
        FOREIGN KEY (nome) REFERENCES instituicao(nome) ON DELETE
CASCADE ,
        FOREIGN KEY (num_concelho,num_regiao) REFERENCES
concelho(num_concelho,num_regiao) ON DELETE CASCADE);
CREATE TABLE f_presc_venda (
    id_presc_venda integer,
    id_medico integer,
    num_doente integer NOT NULL,
    id_data_registo integer NOT NULL,
    id_inst integer,
    substancia varchar(80) NOT NULL,
    quant integer NOT NULL,
    PRIMARY KEY(id_presc_venda),
    FOREIGN KEY(id_presc_venda) REFERENCES
prescricao_venda(num_venda) ON DELETE CASCADE,
    FOREIGN KEY(id_medico) REFERENCES medico(num_cedula) ON
DELETE CASCADE,
    FOREIGN KEY(id_data_registo) REFERENCES d_tempo(id_tempo) ON
DELETE CASCADE ,
    FOREIGN KEY(id_inst) REFERENCES d_instituicao(id_inst) ON
DELETE CASCADE
);
CREATE TABLE f_analise (
    id_analise integer,
    id_medico integer,
    num_doente integer,
    id_data_registo integer,
    id_inst integer,
    nome varchar(80) NOT NULL,
    quant integer NOT NULL,
    PRIMARY KEY(id_analise),
    FOREIGN KEY (id_analise) REFERENCES analise(num_analise) ON
DELETE CASCADE,
    FOREIGN KEY (id_medico) REFERENCES medico(num_cedula) ON
DELETE CASCADE,
    FOREIGN KEY (id_data_registo) REFERENCES d_tempo(id_tempo)
ON DELETE CASCADE,
    FOREIGN KEY (id_inst) REFERENCES d_instituicao(id_inst) ON
DELETE CASCADE
);

```

ETL de carregamento:

```

INSERT INTO instituicao(nome, tipo, num_regiao, num_concelho)
VALUES('farmacia_de_odivelas','farmacia',2,10);
INSERT INTO consulta(num_cedula, num_doente, data,
nome_instituicao)
VALUES(9,9,'02/01/2017','Hospital_Central_de_Arouca');
INSERT INTO consulta(num_cedula, num_doente, data,
nome_instituicao) VALUES(11,11,'02/02/2017','Hospital_do_Porto');
INSERT INTO consulta(num_cedula, num_doente, data,
nome_instituicao)
VALUES(12,12,'03/04/2017','Hospital_Central_de_Arouca');
INSERT INTO consulta(num_cedula, num_doente, data,
nome_instituicao) VALUES(1,1,'02/06/2017','Hospital_do_Porto');
INSERT INTO consulta(num_cedula, num_doente, data,
nome_instituicao)
VALUES(3,3,'02/08/2017','Hospital_de_vila_franca');
INSERT INTO consulta(num_cedula, num_doente, data,
nome_instituicao)
VALUES(2,2,'01/01/2020','Hospital_de_vila_franca');
INSERT INTO consulta(num_cedula, num_doente, data,
nome_instituicao)
VALUES(5,5,'03/02/2020','Hospital_de_vila_franca');
INSERT INTO prescricao(num_cedula, num_doente, data, substancia,
quant) VALUES(9,9,'02/01/2017','acucar',2);
INSERT INTO prescricao(num_cedula, num_doente, data, substancia,
quant) VALUES(11,11,'02/02/2017','acucar',1);
INSERT INTO prescricao(num_cedula, num_doente, data, substancia,
quant) VALUES(12,12,'03/04/2017','acucar',2);
INSERT INTO prescricao(num_cedula, num_doente, data, substancia,
quant) VALUES(12,12,'02/06/2017','acucar',1);
INSERT INTO prescricao(num_cedula, num_doente, data, substancia,
quant) VALUES(2,2,'01/01/2020','glicerina',1);
INSERT INTO prescricao(num_cedula,num_doente,data, substancia,
quant) VALUES(5,5,'03/02/2020','cha_verde',1);
INSERT INTO analise(num_analise, especialidade, num_cedula,
num_doente, data, data_registo, nome, quant, inst)
VALUES(6,'radiologia',9,9,'02/01/2017','02/01/2017','analise_de_gli
cemia',1,'Hospital_Central_de_Arouca');
INSERT INTO analise(num_analise, especialidade, num_cedula,
num_doente, data, data_registo, nome, quant, inst)
VALUES(7,'cardiologia',11,11,'02/02/2017','02/02/2017','analise_de_
glicemia',1,'Hospital_do_Porto');
INSERT INTO analise(num_analise, especialidade, num_cedula,
num_doente, data, data_registo, nome, quant, inst)

```

```

VALUES(8,'ortopedia',12,12,'03/04/2017','03/04/2017','analise_de_glicemia',1,'Hospital_Central_de_Arouca');
INSERT INTO analise(num_analise, especialidade, num_cedula, num_doente, data, data_registro, nome, quant, inst)
VALUES(9,'cardiologia',1,1,'02/06/2017','02/06/2017','analise_de_glicemia',1,'Hospital_do_Porto');
INSERT INTO analise(num_analise, especialidade, num_cedula, num_doente, data, data_registro, nome, quant, inst)
VALUES(10,'anestesiologia',3,3,'02/08/2017','02/08/2017','analise_de_glicemia',1,'Hospital_de_vila_franca');
INSERT INTO venda_farmacia(num_venda, data_registro, substancia, quant, preco, inst)
VALUES(31,'16/01/2017','acucar',10.0,5.75,'farmacia_central');
INSERT INTO venda_farmacia(num_venda, data_registro, substancia, quant, preco, inst)
VALUES(32,'01/01/2020','glicerina',1,2.75,'farmacia_central_de_sint_ra');
INSERT INTO venda_farmacia(num_venda, data_registro, substancia, quant, preco, inst)
VALUES(33,'01/01/2020','glicerina',1,2.75,'farmacia_central_de_sint_ra');
INSERT INTO venda_farmacia(num_venda, data_registro, substancia, quant, preco, inst)
VALUES(34,'03/02/2020','cha_verde',1,1.65,'farmacia_de_odivelas');
INSERT INTO prescricao_venda(num_cedula, num_doente, data, substancia, num_venda) VALUES(2,2,'01/01/2020','glicerina',32);
INSERT INTO prescricao_venda(num_cedula, num_doente, data, substancia, num_venda) VALUES(5,5,'03/02/2020','cha_verde',33);
INSERT INTO d_tempo(dia,dia_da_semana, semana, mes, trimestre, ano)
(SELECT DISTINCT EXTRACT(day FROM data_registro) AS dia,EXTRACT(dow FROM data_registro) AS dia_da_semana,EXTRACT(week FROM data_registro) AS dia_da_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 d_tempo(dia,dia_da_semana, semana, mes, trimestre, ano)
(SELECT DISTINCT EXTRACT(day FROM data_registro) AS dia,EXTRACT(dow FROM data_registro) AS dia_da_semana,EXTRACT(week FROM data_registro) AS dia_da_semana,
EXTRACT(month FROM data_registro) AS mes,EXTRACT(quarter FROM data_registro) AS trimestre,EXTRACT(year FROM data_registro) AS ano
FROM venda_farmacia
WHERE ((SELECT EXTRACT(day FROM data_registro)),(SELECT

```

```

EXTRACT(month FROM data_registro)), (SELECT EXTRACT(year FROM
data_registro))) NOT IN (SELECT dia,mes,ano FROM d_tempo));
INSERT INTO d_tempo(dia,dia_da_semana, semana, mes, trimestre, ano)
(SELECT DISTINCT EXTRACT(day FROM data) AS dia,EXTRACT(dow FROM
data) AS dia_da_semana,EXTRACT(week FROM data) AS dia_da_semana,
EXTRACT(month FROM data) AS mes,EXTRACT(quarter FROM data) AS
trimestre,EXTRACT(year FROM data) AS ano FROM prescricao_venda
WHERE ((SELECT EXTRACT(day FROM data)), (SELECT EXTRACT(month FROM
data)), (SELECT EXTRACT(year FROM data))) NOT IN (SELECT dia,mes,ano
FROM d_tempo));
INSERT INTO d_instituicao(nome,tipo,num_regiao,num_concelho)
(SELECT * FROM instituicao);
INSERT INTO f_analise(select a.num_analise, a.num_cedula,
a.num_doente , t.id_tempo, i.id_inst, a.nome, a.quant FROM
d_instituicao AS i, analise AS a, d_tempo AS t,instituicao AS ins
WHERE a.inst = ins.nome AND a.inst = i.nome AND t.dia = EXTRACT(DAY
FROM a.data_registro) AND t.mes = EXTRACT(MONTH FROM a.data_registro)
AND t.ano = EXTRACT(YEAR FROM a.data_registro));
INSERT INTO f_presc_venda(SELECT DISTINCT
p.num_venda,m.num_cedula,p.num_doente
,t.id_tempo,i.id_inst,p.substancia,q.quant FROM d_instituicao AS i
,consulta AS c
NATURAL JOIN medico AS m NATURAL JOIN prescricao_venda AS p NATURAL
JOIN prescricao AS q ,d_tempo AS t WHERE t.dia = EXTRACT(DAY FROM
p.data)
AND t.semana = EXTRACT(WEEK FROM p.data) AND t.mes = EXTRACT(MONTH
FROM p.data) AND t.ano = EXTRACT(year FROM p.data) AND i.nome =
c.nome_instituicao);

```

Queries OLAP:

```

SELECT DISTINCT count(id_analise) AS
numero_de_analises_de_glicemia ,m.especialidade,t.mes,t.ano
FROM f_analise AS a,medico AS m,d_tempo as t WHERE a.nome =
'analise_de_glicemia'
AND m.num_cedula = a.id_medico AND a.id_data_registro =
t.id_tempo AND t.ano <= 2020 AND t.ano >=2017 GROUP BY
CUBE(m.especialidade,t.mes,t.ano);
SELECT sum(quant) AS quantidade_total, count(id_presc_venda) AS
n_medio_de_prescricoes_diario,substancia,num_concelho
FROM f_presc_venda AS p,d_tempo AS t,d_instituicao AS i WHERE
t.ano =2020 AND p.id_data_registro = t.id_tempo
AND t.trimestre = 1 AND p.id_inst = i.id_inst AND i.num_regiao =
2 GROUP BY ROLLUP(i.num_concelho,(t.dia_da_semana,t.mes),
p.substancia);

```