

Projeto de Bases de Dados: Parte 4

Nome	Número	Contribuição	Horas de trabalho
Daniel Castro	ist187644	33,33%	8h
Daniela Mendes	ist187646	33,33%	8h
Miguel Oliveira	ist187689	33,33%	8h

Grupo Nº1

Turno: BD817957L08

Professor: André Vasconcelos

Restrições de Integridade

- a) Um Coordenador só pode solicitar vídeos de câmaras colocadas num local cujo accionamento de meios esteja a ser (ou tenha sido) auditado por ele próprio:

```
create or replace function update_solicita()
returns trigger as $$
begin
    if not exists (
        select moradaLocal
        from audita natural join eventoEmergencia natural join vigia
        where audita.idCoordenador = new.idCoordenador
        and audita.numCamara = new.numCamara)

        raise exception "Erro";
    end if;
    return new;
end;
$$ language plpgsql;
```

```
create trigger update_solicita_trigger before insert on solicita
for each row execute procedure update_solicita();
```

- b) Um Meio de Apoio só pode ser alocado a Processos de Socorro para os quais tenha sido accionado:

```
create or replace function update_alocado()
returns trigger as $$
begin
    if not exists (
        select numMeio
        from acciona
        where acciona.numMeio = new.numMeio
        and acciona.nomeEntidade = new.nomeEntidade
        and acciona.numProcessoSocorro = new.numProcessoSocorro)

        raise exception "Erro";
    end if;
    return new;
end;
$$ language plpgsql;
```

```
create trigger update_alocado_trigger before insert on allocated
for each row execute procedure update_alocado();
```

Índices

1.
 - a. Índice do tipo hash, dado que o número da câmara é um valor específico, sobre a tabela vídeo, para o atributo numCamara.
Índice do tipo hash visto que, para além do número da câmara, a morada também tem um valor específico; sobre a tabela vigia, nos atributos numCamara e moradaLocal, porque
 - b. `create index video_idx on video(numCamara) using hash;`
`create index video_idx on vigia(numCamara, moradaLocal) using hash;`
2.
 - a. Índice do tipo b+tree pois temos de ordenar os números de telefone e posteriormente, para cada número de telefone, temos que ordenar os instantes de chamada; sobre a tabela EventoEmergencia, nos atributos numTelefone e instanteChamada.
 - b. `create index evento_idx on vigia(numTelefone, instanteChamada) using b+tree;`

Modelo Multidimensional

```
create table d_meio(  
  idMeio serial,  
  numMeio int not null,  
  nomeMeio varchar(80) not null,  
  nomeEntidade varchar(80) not null,  
  tipo varchar(80),  
  constraint pk_d_meio primary key(idMeio),  
  UNIQUE(numMeio, nomeEntidade, tipo));
```

```
create table d_evento(  
  idEvento serial,  
  numTelefone int not null,  
  instanteChamada timestamp not null,  
  constraint pk_d_evento primary key(idEvento),  
  UNIQUE(numTelefone, instanteChamada));
```

```
create table d_tempo(  
  idTempo serial,  
  dia int not null,  
  mes int not null,  
  ano int not null,  
  constraint pk_d_tempo primary key(idTempo),  
  UNIQUE(dia, mes, ano));
```

```
create table d_junta(  
  idTempo int not null,  
  idEvento int not null,  
  idMeio int not null,  
  constraint pk_d_junta primary key(idTempo, idEvento, idMeio));
```

Populate

```
CREATE OR REPLACE FUNCTION populateTempo()
```

```
RETURNS VOID AS $$
```

```
DECLARE
```

```
date DATE;
```

```
duration INTERVAL;
```

```
BEGIN
```

```
duration := '24 HOURS';
```

```
date = '2018-01-01';
```

```
LOOP
```

```
exit when date = '2019-01-01';
```

```
INSERT INTO d_tempo(dia,mes,ano)
```

```
VALUES (EXTRACT(day from date),EXTRACT(month from date),EXTRACT(year from date));
```

```
date:= date + duration;
```

```
end loop;
```

```
END
```

```
$$ LANGUAGE plpgsql;
```

```
SELECT populateTempo();
```

```
insert into d_evento (numTelefone, instanteChamada) select numTelefone, instanteChamada from eventoEmergencia;
```

```
insert into d_meio (numMeio, nomeMeio, nomeEntidade, tipo) select numMeio, nomeMeio, nomeEntidade, 'Combate' from meio natural join meioCombate;
```

```
insert into d_meio (numMeio, nomeMeio, nomeEntidade, tipo) select numMeio, nomeMeio, nomeEntidade, 'Apoio' from meio natural join meioApoio;
```

```
insert into d_meio (numMeio, nomeMeio, nomeEntidade, tipo) select numMeio, nomeMeio, nomeEntidade, 'Socorro' from meio natural join meioSocorro;
```

```
insert into d_meio (numMeio, nomeMeio, nomeEntidade, tipo)
```

```
select numMeio, nomeMeio, nomeEntidade, null
```

```
from meio where (numMeio, nomeEntidade) not in (
```

```
select * from meioCombate union
```

```
select * from meioSocorro union
```

```
select * from meioApoio);
```

```
drop table if exists test cascade;
```

```
create table test(
```

```
numTelefone int not null,
```

```
instanteChamada timestamp not null,
```

```
numMeio int not null,
```

```
nomeEntidade varchar(80) not null,
```

```
tipo varchar(80),
```

```
dia int not null,
```

```
mes int not null,
```

```
ano int not null );
```

```
insert into test(numTelefone, instanteChamada, numMeio, nomeEntidade, tipo, dia, mes, ano)
  select numTelefone, instanteChamada, numMeio, nomeEntidade, 'Combate',
    EXTRACT(day from instanteChamada), EXTRACT(month from instanteChamada), EXTRACT(year from
instanteChamada)
  from eventoemergencia natural join acciona natural join meioCombate;
```

```
insert into test(numTelefone, instanteChamada, numMeio, nomeEntidade, tipo, dia, mes, ano)
  select numTelefone, instanteChamada, numMeio, nomeEntidade, 'Apoio',
    EXTRACT(day from instanteChamada), EXTRACT(month from instanteChamada), EXTRACT(year from
instanteChamada)
  from eventoemergencia natural join alocado;
```

```
insert into test(numTelefone, instanteChamada, numMeio, nomeEntidade, tipo, dia, mes, ano)
  select numTelefone, instanteChamada, numMeio, nomeEntidade, 'Socorro',
    EXTRACT(day from instanteChamada), EXTRACT(month from instanteChamada), EXTRACT(year from
instanteChamada)
  from eventoemergencia natural join transporta;
```

```
insert into test(numTelefone, instanteChamada, numMeio, nomeEntidade, tipo, dia, mes, ano)
  select numTelefone, instanteChamada, numMeio, nomeEntidade, null,
    EXTRACT(day from instanteChamada), EXTRACT(month from instanteChamada), EXTRACT(year from
instanteChamada)
  from eventoemergencia natural join transporta
  where (numMeio, nomeEntidade) not in (
    select * from meioCombate union
    select * from meioSocorro union
    select * from meioApoio);
```

```
insert into d_junta(idTempo, idEvento, idMeio)
  select idTempo, idEvento, idMeio
  from test natural join d_evento natural join d_tempo natural join d_meio;
```

```
drop table if exists test cascade; --apenas para limpar
```

Data Analytics

Com Union:

```
select tipo, ano, mes, count(idMeio)
  from d_junta natural join d_meio natural join d_tempo
  where idEvento = 15
  group by tipo, ano, mes
  union
  select tipo, ano, null, count(idMeio)
  from d_junta natural join d_meio natural join d_tempo
  where idEvento = 15
```

```
group by tipo, ano
union
select tipo, null, null, count(idMeio)
from d_junta natural join d_meio natural join d_tempo
where idEvento = 15
group by tipo
order by tipo, ano, mes;
```

Com rollup:

```
select tipo, ano, mes, count(idMeio)
from d_junta natural join d_meio natural join d_tempo
where idEvento = 15
group by tipo, rollup(ano, mes)
order by tipo, ano, mes;
```