



Projeto de Bases de Dados, Parte 4

Trabalho realizado por:

Nome	Número	Esforço
Bernardo Valente	87521	8 horas
Francisco Machado	87530	8 horas
João Felício	87542	8 horas

Grupo: 34

Turno: Quarta-feira, 15h - BD817957L05

Docente: Raquel Casteleiro



Projeto de Bases de Dados, Parte 4

Restrições de Integridade

a)

```
DROP TRIGGER IF EXISTS verifica_solicita_trigger ON SOLICITA;  
CREATE OR REPLACE FUNCTION verifica_solicita () RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
    IF (SELECT moradaLocal  
        FROM vigia  
        WHERE numCamara = new.numCamara)  
    NOT IN (select moradaLocal  
            FROM solicita NATURAL JOIN audita NATURAL JOIN eventoEmergencia  
            WHERE idCoordenador = new.idCoordenador )  
    THEN  
        RAISE EXCEPTION 'Camara invalida';  
    END IF;  
    RETURN new;
```

```
END;
```

```
$$ Language plpgsql;
```

```
CREATE TRIGGER verifica_solicita_trigger BEFORE INSERT ON solicita  
FOR EACH ROW EXECUTE PROCEDURE verifica_solicita();
```

b)

```
DROP TRIGGER IF EXISTS insert_alocado_trigger ON solicita;  
CREATE OR REPLACE FUNCTION insert_alocado () RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
    IF (SELECT numMeio, nomeEntidade  
        FROM Alocado  
        WHERE numMeio = new.numMeio AND nomeEntidade = new.nomeEntidade)  
    NOT IN (SELECT numMeio, nomeEntidade  
            FROM Alocado NATURAL JOIN Acciona)  
    THEN  
        RAISE EXCEPTION 'Meio nao alocado'  
        USING HINT = 'Tem que ser um meio accionado pelo processo de socorro';
```

```
END IF;
```



Projeto de Bases de Dados, Parte 4

```
RETURN new;  
END;  
$$ Language plpgsql;
```

```
CREATE TRIGGER insert_alocado_trigger BEFORE INSERT ON Solicita  
FOR EACH ROW EXECUTE PROCEDURE insert_alocado();
```

Índices

1.

a)

As tabelas Vigia e Video têm índice para a coluna numCamara. Na tabela Vigia há também um índice do tipo CLUSTERED para a coluna moradaLocal, para ser mais rápido encontrarmos todos os valores “Loures”.

Os índices criados devem ser densos porque a sua utilização proporciona uma velocidade de procura inferior em relação aos índices esparsos. Para operações de inserção em tabelas, os índices densos têm a desvantagem de serem mais lentos relativamente aos índices esparsos. Esta particularidade não afeta o objectivo em questão porque apenas são feitas *queries* do tipo SELECT.

b)

```
CREATE INDEX videoCamara ON Video USING HASH(numCamara);  
CREATE INDEX vigiaCamara ON Vigia USING HASH(numCamara);  
CREATE INDEX vigiaMorada ON Vigia(moradaLocal);  
CLUSTER vigiaMorada ON Vigia;
```

2.

a)

Foram criados índices nos atributos numProcesso nas tabelas trasporta e eventoEmergencia. Também criámos um índice composto para os valores (numTelefone, instanteChamada) na tabela eventoEmergencia, este índice é do tipo Clustered porque precisamos de fazer um group by.

Os índices criados devem ser densos porque a sua utilização proporciona uma velocidade de procura inferior em relação aos índices esparsos. Para operações de inserção em tabelas, os



Projeto de Bases de Dados, Parte 4

Índices densos têm a desvantagem de serem mais lentos relativamente aos índices esparsos. Esta particularidade não afeta o objectivo em questão porque nós apenas fazemos queries do tipo SELECT.

b)

```
CREATE INDEX transporta_idx ON Transporta USING HASH(numProcessoSocorro);
CREATE INDEX evento_idx ON EventoEmergencia USING HASH(numProcessoSocorro);
CREATE INDEX group_idx ON EventoEmergencia(numTelefone, instanteChamada);
CLUSTER group_idx ON EventoEmergencia;
```

Modelo Multidimensional

```
DROP TABLE IF EXISTS d_evento cascade;
DROP TABLE IF EXISTS d_meio cascade;
DROP TABLE IF EXISTS d_tempo cascade;
DROP TABLE IF EXISTS d_facts;
```

```
CREATE TABLE d_evento(
    idEvento          serial not null,
    numTelefone        integer not null,
    instanteChamada    timestamp not null,
    PRIMARY KEY(idEvento),
    UNIQUE(numtelefone, instantechamada),
    FOREIGN KEY(numTelefone, instanteChamada) REFERENCES EventoEmergencia
);
```

```
CREATE TABLE d_meio(
    idMeio            serial not null,
    numMeio           integer not null,
    nomeMeio          varchar(80) not null,
    nomeEntidade      varchar(80) not null,
    tipo              varchar(80),

    PRIMARY KEY (idMeio),
    UNIQUE (nummeio, nomeMeio, nomeentidade, tipo),
```



Projeto de Bases de Dados, Parte 4

```
FOREIGN KEY(nummeio, nomeentidade) REFERENCES Meio(nummeio,  
nomeentidade)  
);
```

```
CREATE TABLE d_tempo(  
    idData        serial not null,  
    dia           integer not null,  
    mes           integer not null,  
    Ano           integer not null,  
    PRIMARY KEY (idData),  
    UNIQUE (dia, mes, ano)  
);
```

```
CREATE TABLE d_facts(  
    idEvento      integer not null,  
    idMeio        integer not null,  
    idData        integer not null,  
  
    PRIMARY KEY(idEvento, idMeio, idData),  
    FOREIGN KEY(idEvento) REFERENCES d_evento,  
    FOREIGN KEY(idMeio) REFERENCES d_meio,  
    FOREIGN KEY(idData) REFERENCES d_tempo);
```

Populate:

```
INSERT INTO d_evento(numTelefone, instanteChamada)  
SELECT numTelefone, instanteChamada  
FROM eventoEmergencia;
```

```
INSERT INTO d_meio(numMeio, nomeMeio, nomeEntidade)  
SELECT numMeio, nomeMeio, nomeEntidade  
FROM Meio  
WHERE (numMeio, nomeEntidade) NOT IN (  
    SELECT numMeio, nomeEntidade  
    FROM MeioApoio  
UNION  
    SELECT numMeio, nomeEntidade  
    FROM MeioCombate  
UNION
```



Projeto de Bases de Dados, Parte 4

```
SELECT numMeio, nomeEntidade
FROM MeioSocorro
);
```

```
INSERT INTO d_meio(numMeio,nomeMeio,nomeEntidade, tipo)
SELECT numMeio,nomeMeio, nomeEntidade, 'Apoio'
FROM MeioApoio NATURAL JOIN Meio;
```

```
INSERT INTO d_meio(numMeio,nomeMeio,nomeEntidade, tipo)
SELECT numMeio,nomeMeio, nomeEntidade, 'Combate'
FROM MeioCombate NATURAL JOIN Meio;
```

```
INSERT INTO d_meio(numMeio,nomeMeio,nomeEntidade,tipo)
SELECT numMeio,nomeMeio,nomeEntidade, 'Socorro'
FROM MeioSocorro NATURAL JOIN Meio;
```

```
INSERT INTO d_tempo(dia, mes, ano)
SELECT DISTINCT
    EXTRACT(day from instanteChamada),
    EXTRACT(month from instanteChamada),
    EXTRACT(year from instanteChamada)
FROM eventoemergencia;
```

```
INSERT INTO d_facts(idEvento, idMeio, idData)
SELECT DISTINCT idEvento, idMeio, idData
FROM (d_evento NATURAL JOIN eventoEmergencia
      NATURAL JOIN d_meio
      NATURAL JOIN acciona) A, d_tempo T
WHERE EXTRACT(day from A.instanteChamada) = T.dia
      AND EXTRACT(month from A.instanteChamada) = T.mes
      AND EXTRACT(year from A.instanteChamada) = T.ano;
```



Projeto de Bases de Dados, Parte 4

Data Analytics

```
SELECT tipo, count(numMeio), ano, mes
FROM d_tempo NATURAL JOIN d_meio NATURAL JOIN d_evento NATURAL JOIN d_facts
WHERE idEvento=15
GROUP BY tipo, ROLLUP (ano, mes);
```

OU

```
SELECT tipo, count(numMeio), ano, mes
FROM d_tempo NATURAL JOIN d_meio NATURAL JOIN d_evento NATURAL JOIN d_facts
WHERE idEvento=15
GROUP BY tipo, ano, mes
UNION
SELECT tipo, count(numMeio), null, mes
FROM d_tempo NATURAL JOIN d_meio NATURAL JOIN d_evento NATURAL JOIN d_facts
WHERE idEvento=15
GROUP BY tipo, mes
UNION
SELECT tipo, count(numMeio), ano, null
FROM d_tempo NATURAL JOIN d_meio NATURAL JOIN d_evento NATURAL JOIN d_facts
WHERE idEvento=15
GROUP BY tipo, ano
UNION
SELECT tipo, count(numMeio), null, null
FROM d_tempo NATURAL JOIN d_meio NATURAL JOIN d_evento NATURAL JOIN d_facts
WHERE idEvento=15
GROUP BY tipo;
```