

Bases de Dados 2018/2019

LEIC-T

Parte 4

Sistema de Gestão de Incêndios Florestais

Nome	Número	NºHoras	Percentagem relativa
Francisco Sousa	87657	5	33.3%
Diogo Sá	87652	5	33.3%
Isabel Dias	84726	5	33.3%

Número do Grupo: 27

Turno: BD817957L08

Docente: André Vasconcelos

Restrições de Integridade

a) **CREATE TRIGGER** insert_solicita_trigger **BEFORE INSERT ON** Sgif.Solicita
FOR EACH row EXECUTE PROCEDURE Sgif.insert_solicita();

CREATE OR REPLACE Sgif.insert_solicita() **RETURNS TRIGGER**
AS \$\$
BEGIN
 IF NOT EXISTS (
 SELECT *
 FROM Sgif.Audita **NATURAL JOIN** Sgif.EventoEmergencia **NATURAL JOIN**
Sgif.Vigia
 WHERE idCoordenador = new.idCoordenador **AND** dataHoraInicio =
 new.dataHoraInicioVideo **AND** numCamara = new.numCamara
)
 THEN
 RAISE EXCEPTION 'Nao e possivel solicitar o video.';
 END IF;
END;
\$\$ LANGUAGE plpgsql;

b) **CREATE TRIGGER** insert_alocado_trigger **BEFORE INSERT ON** Sgif.Alocado
FOR EACH row EXECUTE PROCEDURE Sgif.insert_alocado();

CREATE OR REPLACE FUNCTION Sgif.insert_alocado() **RETURNS TRIGGER**
AS \$\$
BEGIN
 IF NOT EXISTS (
 SELECT *
 FROM Sgif.MeioApoio **NATURAL JOIN** Sgif.Acciona
 WHERE numProcessoSocorro = new.numProcessoSocorro
)
 THEN
 RAISE EXCEPTION 'Nao e possivel alocar o meio de apoio.';
 END IF;
 RETURN new;
END
\$\$ LANGUAGE plpgsql;

Índices

1.
 - a) Para a query pedida, teríamos índices baseados em funções de dispersão sobre as tabelas **Video** e **Vigia**. O uso das funções de dispersão é justificado pelo facto da *query* fazer testes de igualdade e a hash permitir ter uma complexidade de $O(1)$. Na tabela **Video** criamos um índice sobre o atributo **numCamara** e na tabela **Vigia** dois índices sobre os atributos **numCamara** e **moradaLocal**.
 - b)

```
CREATE INDEX idx_vid_numCamara ON Sgif.Video USING HASH (numCamara)
CREATE INDEX idx_vig_numCamara ON Sgif.Vigia USING HASH (numCamara)
CREATE INDEX idx_moradaLocal ON Sgif.Vigia USING HASH (moradaLocal)
```
2.
 - a) Para a query pedida, teríamos índices baseados em funções de dispersão sobre as tabelas **Transporta** e **EventoEmergencia** pela mesma razão explicada na interrogação anterior. Portanto, na tabela **Transporta** criamos um índice sobre o atributo **numProcessoSocorro**, assim como na tabela **EventoEmergencia** criamos um índice sobre o atributo com o mesmo nome, facilitando assim o teste de igualdade. Para os atributos dentro do **GROUP BY** criamos um índice composto porque são atributos únicos que identificam a tabela **EventoEmergencia** e, como referido anteriormente, usar índices compostos é mais eficiente.
 - b)

```
CREATE INDEX idx_tr_numProcessoSocorro ON Sgif.Transporta (numProcessoSocorro) USING
HASH
CREATE INDEX idx_ev_numProcessoSocorro ON Sgif.EventoEmergencia (numProcessoSocorro)
USING HASH
CREATE INDEX idx_numTelefone ON Sgif.EventoEmergencia USING HASH (numTelefone,
instanteChamada)
```

Modelo Multidimensional

```
drop table if exists Sgif.d_evento cascade;
drop table if exists Sgif.d_meio cascade;
drop table if exists Sgif.d_tempo cascade;
drop table if exists Sgif.tf_eventos_meios cascade;

create table Sgif.d_evento (
    idEvento serial,
    numTelefone numeric(9, 0) not null,
    instanteChamada timestamp not null,
    constraint pk_d_evento primary key(idEvento));

create table Sgif.d_meio (
    idMeio serial,
    numMeio int not null,
    nomeMeio varchar(40) not null,
    nomeEntidade varchar(40) not null,
    tipo varchar(40),
    constraint pk_d_meio primary key(idMeio));

create table Sgif.d_tempo (
    idData serial,    dia numeric(2, 0) not null,
    mes numeric(2, 0) not null,
    ano numeric(4, 0) not null,
    constraint pk_d_tempo primary key(idData));

create table Sgif.tf_eventos_meios (
    idEvento int,
    idMeio int,
    idData int,
    constraint pk_tf_eventos_meios primary key(idEvento, idMeio, idData),
    constraint fk_evento foreign key(idEvento) references Sgif.d_evento on delete cascade,
    constraint fk_meio foreign key(idMeio) references Sgif.d_meio on delete cascade,
    constraint fk_tempo foreign key(idData) references Sgif.d_tempo on delete cascade);

INSERT INTO Sgif.d_evento (numTelefone, instanteChamada)
SELECT numTelefone, instanteChamada
FROM Sgif.EventoEmergencia;
```

```

INSERT INTO Sgif.d_meio (numMeio, nomeMeio, nomeEntidade, tipo)
SELECT numMeio, nomeMeio, nomeEntidade, 'combate'
FROM Sgif.Meio NATURAL JOIN Sgif.MeioCombate;

```

```

INSERT INTO Sgif.d_meio (numMeio, nomeMeio, nomeEntidade, tipo)
SELECT numMeio, nomeMeio, nomeEntidade, 'apoio'
FROM Sgif.Meio NATURAL JOIN Sgif.MeioApoio;

```

```

INSERT INTO Sgif.d_meio (numMeio, nomeMeio, nomeEntidade, tipo)
SELECT numMeio, nomeMeio, nomeEntidade, 'socorro'
FROM Sgif.Meio NATURAL JOIN Sgif.MeioSocorro;

```

```

INSERT INTO Sgif.d_meio (numMeio, nomeMeio, nomeEntidade, tipo)
SELECT M.numMeio, M.nomeMeio, M.nomeEntidade, 'sem-tipo'
FROM Sgif.Meio M
WHERE NOT EXISTS (
    SELECT *
    FROM Sgif.MeioCombate C
    WHERE M.numMeio = C.numMeio
) AND NOT EXISTS (
    SELECT *
    FROM Sgif.MeioApoio A
    WHERE M.numMeio = A.numMeio
) AND NOT EXISTS (
    SELECT *
    FROM Sgif.MeioApoio S
    WHERE M.numMeio = S.numMeio
);

```

```

INSERT INTO Sgif.d_tempo (dia, mes, ano)
SELECT DISTINCT
    EXTRACT(DAY FROM instanteChamada),
    EXTRACT(MONTH FROM instanteChamada),
    EXTRACT(YEAR FROM instanteChamada)
FROM Sgif.EventoEmergencia;

```

```

INSERT INTO Sgif.tf_eventos_meios (idEvento, idMeio, idData)
SELECT idEvento, idMeio, idData
FROM Sgif.Acciona NATURAL JOIN Sgif.EventoEmergencia NATURAL JOIN Sgif.d_evento NATURAL JOIN
Sgif.d_meio NATURAL JOIN Sgif.d_tempo;

```

Data Analytics

```
select count(tipo)
from Sgif.tf_eventos_meios natural join Sgif.d_meio natural join Sgif.d_evento natural join
Sgif.d_tempo
where idEvento = 15
group by rollup(tipo, ano, mes);
```

(Testado em Postgres 11.1)

Organização do zip entregue

- *relatorio27.pdf* - relatório.
- *schema.sql* - schema entregue na entrega anterior, mais os *triggers* correspondentes às restrições de integridade.
- *star.sql* - esquema em estrela com a informação relativa aos eventos e meios usados, mais as instruções SQL necessárias para carregar o esquema.
- *olap.sql* - interrogação SQL OLAP pedida.