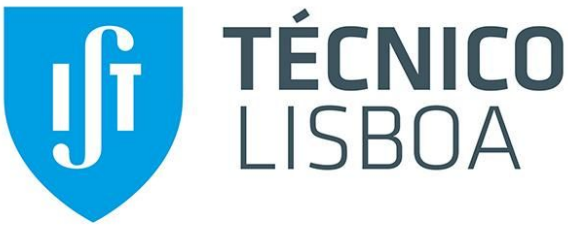

PROJETO DE BASES DE DADOS

Parte 4

Grupo 42 - Turno L09 - Segunda-Feira 12:30	
Professor: Taras Lykhenko	
Aluno	Esforço
João Palet - 86447	**12h** (33,3%)
Miguel Grilo - 86489	**12h** (33,3%)
Simão Nunes - 86512	**12h** (33,3%)

Restrições de Integridade

a)

```
CREATE OR REPLACE FUNCTION verifica_audita() RETURNS TRIGGER AS
$BODY$
BEGIN
   IF NEW.numCamara NOT IN(SELECT numcamara
                  FROM audita NATURAL JOIN eventoemergencia NATURAL JOIN vigia
                  WHERE idcoordenador = NEW.idCoordenador) THEN
                  RAISE EXCEPTION
                  'O Coordenador % nao audita accionamento de meios num local vigiado pela
camara %.', NEW.idCoordenador, NEW.numCamara;
   END IF;
   RETURN NEW;
END;
$BODY$ LANGUAGE plpgsql;

CREATE TRIGGER verifica_audita_trigger BEFORE INSERT OR UPDATE ON solicita
FOR EACH ROW EXECUTE PROCEDURE verifica_audita();
```

b)

```
CREATE OR REPLACE FUNCTION verifica_accionamento() RETURNS TRIGGER AS
$BODY$
BEGIN
   IF NOT EXISTS(SELECT 1
            FROM acciona
            WHERE numMeio = NEW.numMeio AND nomeEntidade = NEW.nomeEntidade
AND numProcessoSocorro = NEW.numProcessoSocorro) THEN
            RAISE EXCEPTION
            'O Meio de Apoio % de % nao e accionado pelo Processo %.', NEW.numMeio,
NEW.nomeEntidade, NEW.numProcessoSocorro;
   END IF;
   RETURN NEW;
END;
$BODY$ LANGUAGE plpgsql;

CREATE TRIGGER verifica_accionamento_trigger BEFORE INSERT OR UPDATE ON
alocado
FOR EACH ROW EXECUTE PROCEDURE verifica_accionamento();
```

Índices

1.

a)

Dado que, no *WHERE,* a escolha da tabela onde é feito o lookup depende do tamanho das tabelas em questão (*video* e *vigia*), e estas têm tamanho variável, temos duas possíveis alternativas para os índices a utilizar.

Independentemente da tabela onde seja feito o lookup, a utilização de um índice para a *moradaLocal* na tabela *vigia* irá otimizar a query. Também será útil um índice para o *numCamara* na tabela *video*. Todos estes índices serão baseados numa função de dispersão (hash table) pois estes são melhores para testes de igualdade.

Assim sendo, caso o lookup seja feito na tabela *video*:

b) **CREATE INDEX** video_numCamara_index **ON** video **USING HASH** (numCamara);
CREATE INDEX vigia_moradaLocal_index **ON** vigia **USING HASH** (moradaLocal);

Caso seja feito na tabela *vigia*:

b) **CREATE INDEX** vigia_numCamara_index **ON** vigia **USING HASH** (numCamara);
CREATE INDEX video_numCamara_index **ON** video **USING HASH** (numCamara);
CREATE INDEX vigia_moradaLocal_index **ON** vigia **USING HASH** (moradaLocal);

2.

a)

Dado que, no *WHERE,* a escolha da tabela onde é feito o lookup depende do tamanho das tabelas em questão (*transporta* e *eventoEmergencia*), e estas têm tamanho variável, temos duas possíveis alternativas para os índices a utilizar. Estes serão baseados numa função de dispersão (hash table) pois estes são melhores para testes de igualdade.

Independentemente da tabela onde seja feito o lookup, a utilização de um índice composto em *numTelefone*, *instanteChamada* aumentará a eficiência da query para o *GROUP BY*. Este será um índice com estrutura *BTREE*, pois índices compostos não são suportados pela estrutura hash table.

Assim sendo, caso o lookup seja feito na tabela *transporta*:

b) **CREATE INDEX** transporta_numProcessoSocorro_index **ON** video **USING HASH** (numProcessoSocorro);
CREATE INDEX eventoEmergencia_numProcessoSocorro **ON** vigia **USING BTREE** (numTelefone, instanteChamada);

Caso seja feito na tabela *eventoEmergencia*:

b) **CREATE INDEX** eventoEmergencia_numProcessoSocorro **ON** vigia **USING HASH** (numProcessoSocorro);
CREATE INDEX eventoEmergencia_numProcessoSocorro **ON** vigia **USING BTREE** (numTelefone, instanteChamada);

Modelo Multidimensional

```
DROP TABLE IF EXISTS d_evento;
DROP TABLE IF EXISTS d_meio;
DROP TABLE IF EXISTS d_tempo;
DROP TABLE IF EXISTS factos;

CREATE TABLE d_evento(
   idEvento SERIAL,
   numTelefone VARCHAR(13) NOT NULL,
   instanteChamada TIMESTAMP NOT NULL,
   PRIMARY KEY(idEvento)
);

CREATE TABLE d_meio(
   idMeio SERIAL,
   numMeio INT NOT NULL,
   nomeMeio VARCHAR(255) NOT NULL,
   nomeEntidade VARCHAR(255) NOT NULL,
   tipo VARCHAR(255) NOT NULL,
   PRIMARY KEY(idMeio)
);

CREATE TABLE d_tempo(
   idData SERIAL,
   dia INT,
   mes INT,
   ano INT,
   PRIMARY KEY(idData)
);

CREATE TABLE factos(
   idEvento INT NOT NULL,
   idMeio INT NOT NULL,
   idData INT NOT NULL,
   PRIMARY KEY(idEvento, idMeio, idData),
   FOREIGN KEY(idEvento) REFERENCES d_evento(idEvento) ON DELETE CASCADE ON
UPDATE CASCADE,
   FOREIGN KEY(idMeio) REFERENCES d_meio(idMeio) ON DELETE CASCADE ON
UPDATE CASCADE,
   FOREIGN KEY(idData) REFERENCES d_tempo(idData) ON DELETE CASCADE ON
UPDATE CASCADE);
```

```
-- EVENTOS

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

-- MEIOS

INSERT INTO d_meio (numMeio, nomeMeio, nomeEntidade, tipo)
SELECT numMeio, nomeMeio, nomeEntidade, 'N/A' AS tipo
FROM meio
EXCEPT
SELECT numMeio, nomeMeio, nomeEntidade, 'N/A' AS tipo
FROM meio NATURAL JOIN meioCombate
EXCEPT
SELECT numMeio, nomeMeio, nomeEntidade, 'N/A' AS tipo
FROM meio NATURAL JOIN meioApoio
EXCEPT
SELECT numMeio, nomeMeio, nomeEntidade, 'N/A' AS tipo
FROM meio NATURAL JOIN meioSocorro;

INSERT INTO d_meio (numMeio, nomeMeio, nomeEntidade, tipo)
SELECT numMeio, nomeMeio, nomeEntidade, 'Combate' AS tipo
FROM meio NATURAL JOIN meioCombate;

INSERT INTO d_meio (numMeio, nomeMeio, nomeEntidade, tipo)
SELECT numMeio, nomeMeio, nomeEntidade, 'Apoio' AS tipo
FROM meio NATURAL JOIN meioApoio;

INSERT INTO d_meio (numMeio, nomeMeio, nomeEntidade, tipo)
SELECT numMeio, nomeMeio, nomeEntidade, 'Socorro' AS tipo
FROM meio NATURAL JOIN meioSocorro;
```

```
-- TEMPO

INSERT INTO d_tempo (dia, mes, ano)
SELECT date_part('day',day::date) as dia,
   date_part('month',day::date) as mes,
   date_part('year',day::date) as ano
FROM (
SELECT day::date
FROM generate_series(date '2016-01-01', 2018-12-31, '1 day') day
) AS date_table;


-- FACTOS

INSERT INTO factos
SELECT idEvento, idMeio, idData
FROM (d_evento NATURAL JOIN eventoEmergencia NATURAL JOIN acciona NATURAL
JOIN d_meio)
   INNER JOIN d_tempo ON extract(year from eventoEmergencia.instanteChamada) =
d_tempo.ano AND
   extract(month from eventoEmergencia.instanteChamada) = d_tempo.mes AND
   extract(day from eventoEmergencia.instanteChamada) = d_tempo.dia;
```

Data Analytics

```
SELECT tipo, ano, mes, count(*) AS numMeios
FROM factos NATURAL JOIN d_meio NATURAL JOIN d_evento NATURAL JOIN d_tempo
WHERE idEvento =  15
GROUP BY ROLLUP (tipo, ano, mes)
```