



Projeto de Bases de Dados, Parte 4

83533	Mihail Brinza	.%	h
83557	Ricardo Brancas	.%	h
83883	David Nunes	.%	h

Grupo 37, turno BD2251795L10
Professor Miguel Amaral

1 Restrições de Integridade

```
-- a)

CREATE FUNCTION forn_prim_sec()
    RETURNS TRIGGER AS $$
BEGIN

    IF EXISTS(SELECT *
              FROM fornece_sec
              WHERE ean = NEW.ean
                AND nif = NEW.forn_primario)

    THEN
        RAISE EXCEPTION
        'Tried_to_set_product_%_with_forn_prim_%,_but_product_already_has_forn_sec_',
        NEW.ean, NEW.forn_primario, NEW.forn_primario;
    END IF;

END
$$ LANGUAGE plpgsql;

CREATE TRIGGER forn_prim_sec
BEFORE INSERT OR UPDATE ON produto
FOR EACH ROW EXECUTE PROCEDURE forn_prim_sec();

CREATE FUNCTION forn_sec_prim()
    RETURNS TRIGGER AS $$
BEGIN

    IF EXISTS(SELECT *
              FROM produto
              WHERE ean = NEW.ean
                AND forn_primario = NEW.nif)

    THEN
        RAISE EXCEPTION
        'Tried_to_set_forn_sec_%_for_product_%,_but_product_already_has_forn_prim_',
        NEW.nif,NEW.ean, NEW.nif;
    END IF;

END
$$ LANGUAGE plpgsql;

CREATE TRIGGER forn_sec_prim
BEFORE INSERT OR UPDATE ON fornece_sec
FOR EACH ROW EXECUTE PROCEDURE forn_sec_prim();

-- b)

ALTER TABLE reposicao
    ADD CHECK (instante <= CURRENT_TIMESTAMP);
```

2 Índices

2.1 Índice 1

Esta *query* contém um **JOIN** cuja condição é de igualdade; a forma mais eficiente de o realizar é percorrendo uma das tabelas (ou um dos índices a ela associados) na totalidade e depois verificar (através de um índice de **HASH**) se existe um registo correspondente na outra tabela. No entanto uma das tabelas (*fornecedor*) já contém um índice sobre a coluna correta (*produto*). Apesar de este índice ser do tipo **BTREE**, muito provavelmente não compensa criar um novo índice do tipo **HASH** (dependendo, no entanto, da situação particular).

Para acelerar a condição **WHERE** devemos ainda criar um índice, também de **HASH**, sobre a coluna *produto.categoria*. Apresentamos os índices em questão:

```
-- CREATE INDEX fornecedor_nif_idx          ON fornecedor USING HASH (nif);
CREATE INDEX produto_categoria_idx         ON produto      USING HASH  (categoria);
```

2.2 Índice 2

Em relação à otimização do **JOIN**, estamos numa situação idêntica à da secção 2.1, não sendo necessário criar nenhum índice adicional (devido à existência da *primary key*).

Para otimizar o **GROUP BY** seria útil criar um índice do tipo **BTREE** sobre a coluna `produto.ean` (ou `fornece_sec.ean`) mas, mais uma vez, já existe um índice com essas características associado à *primary key*. Como tal, não é necessário criar nenhum índice adicional para otimizar esta *query*.

3 Modelo Multidimensional

```
DROP TABLE IF EXISTS reposition_facts;
DROP TABLE IF EXISTS d_produto;
DROP TABLE IF EXISTS d_tempo;

CREATE TABLE d_produto (
    cean          NUMERIC(13),
    categoria     VARCHAR(80),
    nif_fornecedor_principal NUMERIC(9),
    PRIMARY KEY (cean)
);

CREATE TABLE d_tempo (
    tid NUMERIC(8),
    dia NUMERIC(2),
    mes NUMERIC(2),
    ano NUMERIC(4),
    PRIMARY KEY (tid),
    CHECK (CAST(tid AS INT) / 10000 = ano),
    CHECK (CAST(tid AS INT) % 10000 / 100 = mes),
    CHECK (CAST(tid AS INT) % 100 = dia)
);

CREATE TABLE reposition_facts (
    cean NUMERIC(13),
    tid NUMERIC(8),
    PRIMARY KEY (cean, tid),
    FOREIGN KEY (cean) REFERENCES d_produto (cean),
    FOREIGN KEY (tid) REFERENCES d_tempo (tid)
);

INSERT INTO d_tempo
SELECT DISTINCT
    to_number(to_char(instante, 'YYYYMMDD'), '00000000'),
    date_part('day', instante),
    date_part('month', instante),
    date_part('year', instante)
FROM reposicao;

INSERT INTO d_produto
SELECT DISTINCT
    ean,
    categoria,
    forn_primario
FROM reposicao NATURAL JOIN produto;

INSERT INTO reposition_facts
SELECT
    ean,
    to_number(to_char(instante, 'YYYYMMDD'), '00000000')
FROM reposicao;
```

4 Data Analytics

```
SELECT COUNT(cean) as n, categoria, NULL AS ano, NULL AS mes
FROM reposition_facts
    NATURAL JOIN d_tempo
    NATURAL JOIN d_produto
WHERE nif_fornecedor_principal = 123455678
GROUP BY categoria
```

```
UNION
SELECT COUNT(cean) as n, categoria, ano, mes
FROM reposition_facts
      NATURAL JOIN d_tempo
      NATURAL JOIN d_produto
WHERE nif_fornecedor_principal = 123455678
GROUP BY categoria, ano, mes;
```