

Projeto de Bases de Dados, Parte 4

Autores - Grupo 13			
Nome	Numero	Contribuição relativa	Tempo gasto
António Terra	84702	33.(3) %	4h
Miguel Viegas	84747	33.(3) %	4h
Tomás Carrasco	84774	33.(3) %	4h

Docente: Professor André Vasconcelos

Turno: Sexta-feira às 8:30h da manhã

Restrições de Integridade

a)

```
CREATE OR REPLACE FUNCTION cancel_fornece_sec_proc()
    RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS(
        SELECT forn_primario, ean
        FROM produto
        WHERE NEW.ean = produto.ean AND NEW.nif = produto.forn_primario) THEN
        RAISE EXCEPTION 'O fornecedor % ja e fornecedor primario do produto %',
NEW.nif, NEW.ean;
    END IF;
    RETURN NEW;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER cancel_fornece_sec
    AFTER INSERT OR UPDATE ON fornece_sec
    FOR EACH ROW
    EXECUTE PROCEDURE cancel_fornece_sec_proc();

CREATE OR REPLACE FUNCTION cancel_produto_proc()
    RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS(
        SELECT nif, ean
        FROM fornece_sec
        WHERE NEW.ean = fornece_sec.ean AND fornece_sec.nif = NEW.forn_primario)
THEN
        raise exception 'O fornecedor % ja e fornecedor secundario do produto %',
NEW.forn_primario, NEW.ean;
    END IF;
    RETURN NEW;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER cancel_produto
    AFTER UPDATE ON produto
    FOR EACH ROW
    EXECUTE PROCEDURE cancel_produto_proc();
```

b)

```
CREATE TABLE evento_reposicao (
    operador SMALLINT,
    instante TIMESTAMP,

    PRIMARY KEY(operador, instante),

    CHECK(instante <= CURRENT_TIMESTAMP));
```

Índices

- a) No caso da *Query* 1, a otimização pode ser feita com um index na tabela **produto** segundo a coluna **categoria**. O index usa uma Hash, porque é mais eficiente para fazer a comparação entre a categoria e um nome, neste caso, 'Frutos'.

No caso da *Query* 2 não é possível melhorar a eficiência, porque, a realização do *count* requer que se percorra a tabela **fornece_sec** toda, ou seja, é independente da forma como é indexada.

Populamos as tabelas com 400.000 entradas e medimos o tempo de execução da *Query* 1, usando a ferramenta *timing* utilizado nas aulas teóricas, e, ao fim de 15 testes, obtivemos uma média de tempos:

- Sem index: 41,3ms
- Com index: 26,85ms

O index permitiu um *speedup* de aproximadamente de 54% do tempo de execução.

- b) Remoção dos indexes defaults:

- ALTER TABLE fornecedor DROP CONSTRAINT fornecedor_pkey CASCADE;
- ALTER TABLE produto DROP CONSTRAINT produto_pkey CASCADE;

Criação da Hash:

- CREATE INDEX prodCategoria_idx ON produto USING HASH(categoria);

Modelo Multidimensional

```
INSERT INTO d_produto(cean, categoria, nif_fornecedor_principal)
SELECT ean, categoria, forn_primario
FROM produto;
```

```
INSERT INTO d_tempo(dia, mes, ano)
SELECT DISTINCT EXTRACT(DAY FROM instante),
EXTRACT(MONTH FROM instante),
EXTRACT(YEAR FROM instante)
FROM reposicao;
```

```
INSERT INTO reposicoes_facts(cean, dia, mes, ano, unidades_repostas)
SELECT DISTINCT ean, -- DISTINCT porque usamos TIMESTAMP na
EXTRACT(DAY FROM instante), -- reposicao sendo assim possivel repor duas
EXTRACT(MONTH FROM instante), -- vezes o mesmo produto no mesmo dia
EXTRACT(YEAR FROM instante),
unidades
FROM reposicao;
```

```
-- Schema --
```

```
CREATE TABLE d_produto(
cean Numeric(13,0),
```

```
        categoria VARCHAR(40),
        nif_fornecedor_principal Numeric(9,0),
        PRIMARY KEY(cean)
    );

CREATE TABLE d_tempo(
    dia Numeric(2,0),
    mes Numeric(2,0),
    ano Numeric(4,0),

    PRIMARY KEY(dia, mes, ano),
    CHECK (dia >= 1 AND dia <= 31 AND mes IN (1, 3, 5, 7, 9, 11) OR (dia >= 1 AND dia <= 30
    AND mes IN (4, 6, 8, 10, 12)) OR (dia >= 1 AND dia <= 28 AND mes = 2))
);

CREATE TABLE reposicoes_facts(
    cean Numeric(13,0),
    dia Numeric(2,0),
    mes Numeric(2,0),
    ano Numeric(4,0),

    -- measures --

    unidades_repostas SMALLINT,
    PRIMARY KEY(cean, dia, mes, ano),
    FOREIGN KEY(cean) REFERENCES d_produto(cean),
    FOREIGN KEY(dia, mes, ano) REFERENCES d_tempo(dia, mes, ano)
);
```

Data Analytics

```
WITH TEMP AS(
    SELECT ean, categoria, mes, ano
    FROM reposicoes_facts NATURAL JOIN d_produto
    WHERE nif_fornecedor_principal = 123455678)

SELECT categoria, ano, mes, COUNT(ean) AS Num_Reposicoes
FROM TEMP
GROUP BY categoria, ano, mes

UNION

SELECT categoria, ano, NULL AS mes, COUNT(ean) AS Num_Reposicoes
FROM TEMP
GROUP BY categoria, ano

UNION

SELECT categoria, NULL AS ano, NULL AS mes, COUNT(ean) AS Num_Reposicoes
FROM TEMP
GROUP BY categoria
ORDER BY categoria, ano, mes;
```