

BASES DE DADOS

PROJETO — PARTE 4

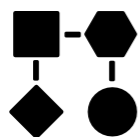
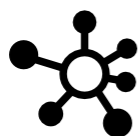
“SUPERMARKET MANAGEMENT”

ANO LETIVO 2017/2018

GRUPO 12 — TURNO L06 — QUARTA-FEIRA, 08H00

DOCENTE: ANDRÉ VASCONCELOS

Aluno	Número	Esforço (em horas)	Esforço (em %)
Hélio Domingos	83473	10	33,3%
Manuel Coimbra	83505	10	33,3%
Miguel Regouga	83530	10	33,3%



RESTRIÇÕES DE INTEGRIDADE

ALÍNEA A)

```
CREATE OR REPLACE FUNCTION check_fornecedor()
  RETURNS trigger as $$
  DECLARE f_aux integer;
BEGIN
  SELECT forn_primario INTO f_aux
  FROM produto
  WHERE forn_primario = new.nif AND ean = new.ean;
  IF new.nif = f_aux
    THEN RAISE EXCEPTION 'O fornecedor com o nif % e fornecedor primario do produto %', new.nif,
    new.ean;
  END IF;
  RETURN NULL;
END
$$ LANGUAGE plpgsql;
CREATE TRIGGER check_forne AFTER INSERT ON fornece_sec FOR EACH ROW
  EXECUTE PROCEDURE check_fornecedor();
```

ALÍNEA B)

CHECK (localtimestamp >= instante)

ÍNDICES

ALÍNEA A)

INTERROGAÇÃO 1

Na primeira interrogação do enunciado do projeto, a computação essencial encontra-se na comparação efetuada entre o NIF do fornecedor e o NIF do fornecedor primário de um produto, e entre a categoria do produto e a categoria 'Frutos'.

Foram assim implementados índices para aumentar a eficiência da interrogação, sobre a tabela *produto* e sobre os atributos *forn_primario* e *categoria*. Tratando-se de duas comparações envolvendo atributos da mesma tabela, o tipo de índice adequado a implementar tem de ser composto e baseado em árvores (*BTREE*).

Este índice é secundário, uma vez que a chave de pesquisa não contém a chave primária. É ainda desagrupado e denso.

No caso apresentado, optámos por criar apenas um índice sobre a tabela *produto*, uma vez que não faria sentido usar um índice na tabela *fornecedor* para o atributo *nif*, dado que para cada entrada da tabela há um *nif* diferente, assim, iria haver um índice diferente para cada linha.

INTERROGAÇÃO 2

Assim como na primeira interrogação, a computação essencial encontra-se numa comparação, desta vez efetuada entre o EAN de um dado produto e o EAN de um dado fornecedor secundário de um produto.

Tratando-se de uma comparação, o tipo de índice adequado a implementar tem de ser baseado em funções de dispersão (*hash*). Foi assim implementado um índice sobre a tabela *fornece_sec* e sobre o atributo *ean*.

O índice é primário, denso e desagrupado. Aqui, não faria sentido criar um índice para a tabela *produto*, uma vez que para cada linha há um *ean* diferente, o que criaria um índice diferente para cada linha e não melhoraria a performance da interrogação em questão.

ALÍNEA B)

INTERROGAÇÃO 1

```
DROP INDEX nif_index;
```

```
CREATE INDEX nif_index ON produto USING BTREE  
(forn_primario, categoria);
```

INTERROGAÇÃO 2

```
DROP INDEX ean_index;
```

```
CREATE INDEX ean_index ON fornece_sec USING HASH  
(ean);
```

MODELO MULTIDIMENSIONAL

CRIAÇÃO DO SISTEMA EM ESTRELA (populate.sql)

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

```
CREATE TABLE d_produto(  
    produto_id SERIAL,  
    cean varchar (80) NOT NULL,  
    categoria varchar(80) NOT NULL,  
    nif_fornecedor_principal integer NOT NULL,  
    PRIMARY KEY (produto_id)  
);
```

```
CREATE TABLE d_tempo(  
    tempo_id timestamp,  
    dia integer NOT NULL,  
    mes integer NOT NULL,  
    ano integer NOT NULL,  
    PRIMARY KEY (tempo_id)  
);
```

```
CREATE TABLE factos(  
    tempo_id timestamp NOT NULL,  
    produto_id integer NOT NULL,  
    FOREIGN key (produto_id) REFERENCES d_produto(produto_id)  
        ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (tempo_id) REFERENCES d_tempo(tempo_id)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

POPULAÇÃO DO SISTEMA EM ESTRELA (populate.sql)

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

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

INSERT INTO factos(tempo_id,produto_id)
    SELECT DISTINCT
        tempo_id,
        produto_id
    FROM
        reposicao INNER JOIN d_tempo ON reposicao.instante=d_tempo.temp_id INNER JOIN d_produto ON
        reposicao.ean=d_produto.ean;
```

DATA ANALYTICS

```
SELECT categoria, ano, mes, count(categoria) FROM factos NATURAL JOIN d_tempo NATURAL JOIN d_produto
WHERE nif_fornecedor_principal='123455678'
GROUP BY categoria, ano, mes
```

UNION

```
SELECT categoria, ano, null, count(categoria) FROM factos NATURAL JOIN d_tempo NATURAL JOIN d_produto
WHERE nif_fornecedor_principal='123455678'
GROUP BY categoria, ano
```

UNION

```
SELECT categoria, null, null, count(categoria) FROM factos NATURAL JOIN d_produto
WHERE nif_fornecedor_principal='123455678'
GROUP BY categoria
```

UNION

```
SELECT null, null, null, count(categoria) FROM factos NATURAL JOIN d_produto
WHERE nif_fornecedor_principal='123455678';
```