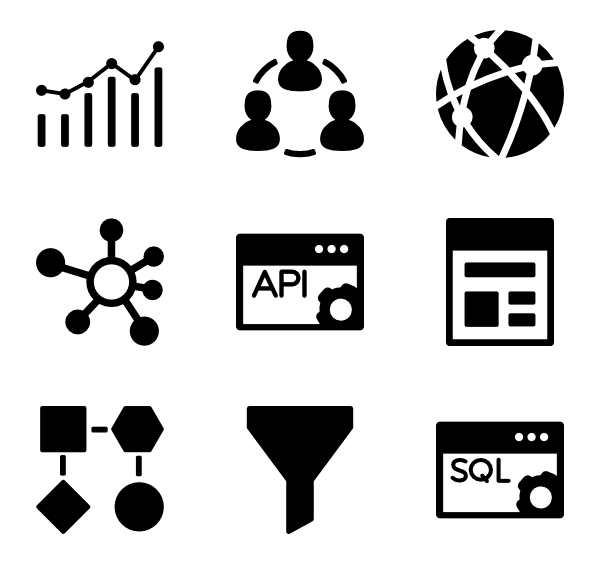
**BASES DE DADOS**

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% |



Licenciatura em Engenharia Informática e de Computadores  
Instituto superior técnico

# 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 (cean, categoria, nif\_fornecedor\_principal)

SELECT ean AS cean, 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.tempo\_id INNER JOIN d\_produto ON reposicao.ean=d\_produto.cean;

# 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';