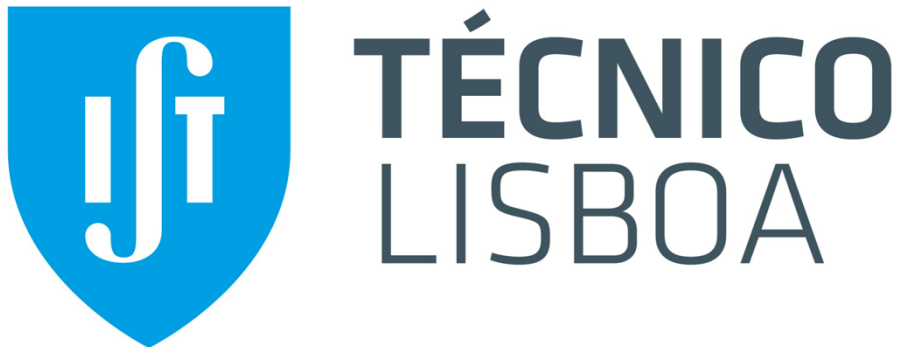


# Projeto de Bases de Dados, 4º Entrega

2017/2018



## Grupo 1

### Membros:

- **84718** – Francisco Aguiar, 33.33% - 9 Horas
- **84719** – Gonalo Marques, 33.33% - 9 Horas
- **84740** – Manuel Sousa, 33.33% - 9 Horas

**Professor:** Toms Almeida e Silva Martins Alves

**Turno:** Quinta-Feira, 8:30 – 10:00 (BD81795L08)

## Restrições de Integridade

a) O fornecedor (primário) de um produto não pode existir na relação fornece\_sec para o mesmo produto.

```
CREATE OR REPLACE FUNCTION PrimaryExistsOnSecondary(nifIN VARCHAR(9),eanIN
VARCHAR(25))
    RETURNS BOOLEAN AS $$
    DECLARE ans BOOLEAN;
BEGIN

    SELECT TRUE INTO ans
    FROM Fornecedor_secundario AS F
    WHERE nif = nifIN AND ean = eanIN;

    IF ans IS NULL THEN ans := FALSE; END IF;
    RETURN ans;
END; $$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION SecondaryExistsOnPrimary(nifIN VARCHAR(9),eanIN
VARCHAR(25))
    RETURNS BOOLEAN AS $$
    DECLARE ans BOOLEAN;
BEGIN

    SELECT TRUE INTO ans
    FROM Produto AS F
    WHERE forn_primario = nifIN AND ean = eanIN;

    IF ans IS NULL THEN ans := FALSE; END IF;
    RETURN ans;
END; $$ LANGUAGE plpgsql;

ALTER TABLE Produto
    ADD CONSTRAINT cantExist CHECK(PrimaryExistsOnSecondary(forn_primario,ean) !=
TRUE);

ALTER TABLE Fornecedor_secundario
    ADD CONSTRAINT cantExist CHECK(SecondaryExistsOnPrimary(nif,ean) != TRUE);
```

b) O instante mais recente de reposição tem de ser sempre anterior ou igual à data atual.

```
ALTER TABLE EventoReposicao
    ADD CONSTRAINT RI_EA3 CHECK(instante <= CURRENT_TIMESTAMP);
```

## Índices

- a) Indique, justificando, que tipo de índice(s), sobre que atributo(s) e sobre que tabela(s) faria sentido criar de modo a acelerar a execução destas interrogações. Analise cada uma das consultas separadamente.

Após análise da query 1, concluímos que deviam ser utilizados os seguintes índices:

- *B-Tree index* primário(clustered) em Fornecedor.nif
- *B-Tree index* primário(clustered) em Produto.fornec\_primario
- *Hashed index* secundário em Produto.categoria

### Justificação:

Este conjunto de índices primários vai fazer com que o join das duas tabelas seja  $O(\#Fornecedores)$  o que é bastante mais eficiente que  $O(\#Produtos * \#Fornecedores)$  que seria a complexidade do join se não tivesse estes índices.

Tendo os produtos indexados por categoria, seria muito mais eficiente encontrar os produtos da categoria “fruta”, supondo que para fazer a procura de um produto da categoria fruta teríamos de percorrer a tabela toda então a complexidade seria  $O(\#Produto)$  mas tendo todos os produtos indexados por categoria teríamos apenas uma procura na ordem de  $O(1)$ .

Após análise da query 2, concluímos que deviam ser utilizados os seguintes índices:

- *B-Tree index* primário(clustered) em Produto.ean
- *B-Tree index* primário(clustered) em Fornecedor\_secundario.ean

### Justificação:

Este conjunto de índices primários vai fazer com que o join das duas tabelas seja  $O(\#Produtos)$  e não  $O(\#Produtos * \#Fornecedores\_secundarios)$  que seria a complexidade do join se não tivesse estes índices.

## Modelo Multidimensional

```
DROP TABLE d_produto CASCADE;
DROP TABLE d_tempo CASCADE;
DROP TABLE info_reposicao CASCADE;

CREATE TABLE d_produto(
    cean VARCHAR(25) NOT NULL,
    categoria VARCHAR(50) NOT NULL,
    nif_fornecedor_principal VARCHAR(9) NOT NULL,
    PRIMARY KEY (cean)
);

CREATE TABLE d_tempo(
    dia INT NOT NULL,
    mes INT NOT NULL,
    ano INT NOT NULL,
    PRIMARY KEY(dia,mes,ano)
);

CREATE TABLE info_reposicao(
    cean VARCHAR(25) NOT NULL,
    dia INT NOT NULL,
    mes INT NOT NULL,
    ano INT NOT NULL,
    numero_reposicoes INT NOT NULL,
    FOREIGN KEY(dia, mes, ano) REFERENCES d_tempo,
    FOREIGN KEY(cean) REFERENCES d_produto,
    PRIMARY KEY(dia,mes,ano,cean)
);

INSERT INTO d_produto
SELECT DISTINCT R.ean AS cean, categoria, forn_primario AS nif_fornecedor_principal
FROM Produto P , Reposicao R
WHERE P.ean = R.ean;

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

INSERT INTO info_reposicao
SELECT DISTINCT cean, dia, mes, ano, nro AS numero_reposicoes
FROM Reposicao, d_produto, d_tempo
WHERE EXTRACT(DAY FROM instante) = dia AND
      EXTRACT(MONTH FROM instante) = mes AND
      EXTRACT(YEAR FROM instante) = ano AND
      cean = ean;
```

## Data Analytics

```
SELECT categoria,ano,mes,SUM(numero_reposicoes)
FROM info_reposicao NATURAL JOIN d_produto
WHERE nif_fornecedor_principal = '123455678'
GROUP BY ROLLUP(categoria, ano, mes);
```