



## Projeto de Bases de Dados, Parte 3

83533	Mihail Brinza	.%	h
38557	Ricardo Brancas	.%	h
83883	David Nunes	.%	h

Grupo 37, turno BD2251795L10  
Professor Miguel Amaral

23 de Novembro de 2017

# 1 Criação da Base de Dados

```
-- Drop tables in case they already exist. Will delete all data

DROP TABLE IF EXISTS reposicao;
DROP TABLE IF EXISTS evento_reposicao;
DROP TABLE IF EXISTS planograma;
DROP TABLE IF EXISTS prateleira;
DROP TABLE IF EXISTS corredor;
DROP TABLE IF EXISTS fornece_sec;
DROP TABLE IF EXISTS produto;
DROP TABLE IF EXISTS fornecedor;
DROP TABLE IF EXISTS constituida;
DROP TABLE IF EXISTS super_categoria;
DROP TABLE IF EXISTS categoria_simples;
DROP TABLE IF EXISTS categoria;

-- Create tables

CREATE TABLE categoria (
    nome VARCHAR(80) NOT NULL,
    PRIMARY KEY (nome)--,
    -- CHECK (nome IN (SELECT nome --RE1, RE2
    -- FROM (SELECT nome
    -- FROM categoria_simples
    -- UNION
    -- SELECT nome
    -- FROM super_categoria) AS x
    -- HAVING count(nome) = 1))
);

CREATE TABLE categoria_simples (
    nome VARCHAR(80) NOT NULL,
    PRIMARY KEY (nome),
    FOREIGN KEY (nome) REFERENCES categoria (nome)
);

CREATE TABLE super_categoria (
    nome VARCHAR(80) NOT NULL,
    PRIMARY KEY (nome),
    FOREIGN KEY (nome) REFERENCES categoria (nome)--,
    --CHECK (nome IN (SELECT super_categoria FROM constituida)) --sor ricardo, has
    -- de ver isto
);

CREATE TABLE constituida (
    super_categoria VARCHAR(80) NOT NULL,
    categoria VARCHAR(80) NOT NULL,
    PRIMARY KEY (super_categoria, categoria),
    FOREIGN KEY (super_categoria) REFERENCES super_categoria (nome),
    FOREIGN KEY (categoria) REFERENCES categoria (nome),
    CHECK (super_categoria <> categoria)
);

CREATE TABLE fornecedor (
    nif NUMERIC(9) NOT NULL,
    nome VARCHAR(80) NOT NULL,
    PRIMARY KEY (nif)
);

CREATE TABLE produto (
    ean NUMERIC(13) NOT NULL,
    design VARCHAR(80) NOT NULL,
    categoria VARCHAR(80) NOT NULL,
    forn_primario NUMERIC(9) NOT NULL,
    PRIMARY KEY (ean),
    FOREIGN KEY (categoria) REFERENCES categoria (nome),
    FOREIGN KEY (forn_primario) REFERENCES fornecedor (nif)
);
```

```

CREATE TABLE fornece_sec (
    nif NUMERIC(9) NOT NULL,
    ean NUMERIC(13) NOT NULL,
    FOREIGN KEY (nif) REFERENCES fornecedor (nif),
    FOREIGN KEY (ean) REFERENCES produto (ean)
);

CREATE TABLE corredor (
    nro VARCHAR(4) NOT NULL,
    largura VARCHAR(3) NOT NULL,
    PRIMARY KEY (nro)
);

CREATE TABLE prateleira (
    nro VARCHAR(4) NOT NULL,
    lado VARCHAR(10) NOT NULL,
    altura VARCHAR(10) NOT NULL,
    PRIMARY KEY (nro, lado, altura),
    FOREIGN KEY (nro) REFERENCES corredor (nro),
    CHECK (lado IN ('esquerdo', 'direito')),
    CHECK (altura IN ('chao', 'medio', 'superior'))
);

CREATE TABLE planograma (
    ean NUMERIC(13) NOT NULL,
    nro VARCHAR(4) NOT NULL,
    lado VARCHAR(10) NOT NULL,
    altura VARCHAR(10) NOT NULL,
    face VARCHAR(5) NOT NULL,
    unidades VARCHAR(4) NOT NULL,
    PRIMARY KEY (ean, nro, lado, altura),
    FOREIGN KEY (ean) REFERENCES produto (ean),
    FOREIGN KEY (nro, lado, altura) REFERENCES prateleira (nro, lado, altura)
);

CREATE TABLE evento_reposicao (
    operador VARCHAR(80) NOT NULL,
    instante TIMESTAMP NOT NULL,
    PRIMARY KEY (operador),
    UNIQUE (instante),
    CHECK (instante > CURRENT_TIMESTAMP)
);

CREATE TABLE reposicao (
    ean NUMERIC(13) NOT NULL,
    nro VARCHAR(4) NOT NULL,
    lado VARCHAR(10) NOT NULL,
    altura VARCHAR(10) NOT NULL,
    unidades VARCHAR(80) NOT NULL,
    operador VARCHAR(80) NOT NULL,
    instante TIMESTAMP NOT NULL,
    PRIMARY KEY (ean, nro, lado, altura),
    UNIQUE (operador),
    UNIQUE (instante),
    FOREIGN KEY (ean, nro, lado, altura) REFERENCES planograma (ean, nro, lado,
        altura)
);

```

## 2 SQL

```

SELECT nome
FROM (
    SELECT forn_primario AS nif, categoria
    FROM produto
    UNION ALL
    SELECT nif, categoria
    FROM fornece_sec NATURAL JOIN produto
) AS c
NATURAL JOIN fornecedor

```

```

GROUP BY nif, nome
HAVING COUNT(categoria) >= ALL (
    SELECT MAX(count)
    FROM (SELECT
        nif,
        COUNT(categoria) AS count
    FROM (SELECT forn_primario AS nif, categoria
        FROM produto
        UNION ALL
        SELECT nif, categoria
        FROM fornece_sec NATURAL JOIN produto) AS a
    GROUP BY nif) AS b
);

```