



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

25 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;
```

```
DROP TYPE IF EXISTS SIDE;
DROP TYPE IF EXISTS HEIGHT;
```

-- Create tables

```
CREATE TYPE SIDE AS ENUM ('esquerda', 'direita');
CREATE TYPE HEIGHT AS ENUM ('chao', 'medio', 'superior');
```

```
CREATE TABLE categoria (
    nome VARCHAR(80) NOT NULL,
    PRIMARY KEY (nome)
);
```

```
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)
);
```

```
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,
    data TIMESTAMP 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      SMALLINT NOT NULL,
    largura  SMALLINT NOT NULL,
    PRIMARY KEY (nro)
);

CREATE TABLE prateleira (
    nro      SMALLINT NOT NULL,
    lado     SIDE      NOT NULL,
    altura   HEIGHT    NOT NULL,
    PRIMARY KEY (nro, lado, altura),
    FOREIGN KEY (nro) REFERENCES corredor (nro)
);

CREATE TABLE planograma (
    ean      NUMERIC(13) NOT NULL,
    nro      SMALLINT    NOT NULL,
    lado     SIDE        NOT NULL,
    altura   HEIGHT      NOT NULL,
    face     SMALLINT    NOT NULL,
    unidades SMALLINT    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),
    UNIQUE (operador, instante),
    CHECK (instante > CURRENT_TIMESTAMP)
);

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

```

2 SQL

```

-- a
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
);

-- b
SELECT DISTINCT nif, nome
FROM produto AS outter
    JOIN fornecedor ON forn_primario = nif
WHERE NOT EXISTS(
    SELECT nome
    FROM categoria_simples
    EXCEPT
    SELECT categoria AS nome
    FROM fornece_sec
        NATURAL JOIN produto
    WHERE nif = outter.forn_primario
        OR forn_primario = outter.forn_primario
);

-- c
SELECT ean
FROM produto
EXCEPT
SELECT ean
FROM reposicao;

-- d
SELECT ean
FROM fornece_sec
GROUP BY ean
HAVING COUNT(nif) > 10;

-- e
SELECT ean
FROM reposicao
GROUP BY ean
HAVING COUNT(DISTINCT operador) = 1;

```