

PROJETO DE BASES DE DADOS

PARTE 3

3º Entrega:
Criação da Base de Dados
Queries SQL
Desenvolvimento da Aplicação

BD81795L03 - Grupo 11

Docente: Tomás Alves

84698 - André Fonseca

84705 - Catarina Custódio

84736 - Leonor Loureiro

Aluno	Horas	Contribuição
84698	10	33,(3)%
84705	10	33,(3)%
84736	10	33,(3)%

CRIAÇÃO DA BASE DE DADOS

/*** CREATE SCHEMA *****/**

create schema if not exists Supermercado;

/*** CREATE DOMAINS *****/**

create domain Supermercado.EAN **as** numeric(13,0);

create domain Supermercado.NIF **as** numeric(9,0);

create domain Supermercado.NRO **as** int;

create domain Supermercado.LADO **as** varchar(3) **check**(value in ('esq','dir'));

create domain Supermercado.ALTURA **as** varchar(8) **check**(value in ('chao','medio', 'superior'));

create domain Supermercado.CAT_NOME **as** varchar(100);

create domain Supermercado.OPERADOR **as** varchar(100);

create domain Supermercado.INSTANTE **as** timestamp;

/*** CREATE TABLES *****/**

create table Supermercado.categoria(
 nome Supermercado.CAT_NOME,
 constraint pk_categoria **primary key** (nome));

create table Supermercado.fornecedor(
 nif Supermercado.NIF,
 nome varchar(100) **not null**,
 constraint pk_fornecedor **primary key** (nif));

create table Supermercado.produto(
 ean Supermercado.EAN,
 design varchar(200),
 categoria Supermercado.CAT_NOME,
 forn_primario Supermercado.NIF **not null**,
 data date **not null default** CURRENT_DATE,
 constraint pk_produto **primary key** (ean),
 constraint fk_categoria **foreign key** (categoria) **references** Supermercado.categoria(nome) **on delete cascade**,
 constraint fk_forn_primario **foreign key** (forn_primario) **references** Supermercado.fornecedor(nif) **on delete cascade**);

create table Supermercado.corredor(
 nro Supermercado.NRO,
 largura numeric(3,2) **not null**,
 constraint pk_corredor **primary key** (nro));

create table Supermercado.prateleira(
 nro Supermercado.NRO,
 lado Supermercado.LADO,
 altura Supermercado.ALTURA,
 constraint pk_prateleira **primary key**(nro,lado,altura),
 constraint fk_nro **foreign key** (nro) **references** Supermercado.corredor **on delete cascade**);

create table Supermercado.planograma(
 ean Supermercado.EAN,
 nro Supermercado.NRO,
 lado Supermercado.LADO,
 altura Supermercado.ALTURA,
 faces numeric(3,0) **not null**,
 unidades numeric(3,0) **not null**,
 loc numeric(3,0) **not null**,
 constraint pk_planograma **primary key**(ean, nro, lado, altura),
 constraint fk_ean **foreign key**(ean) **references** Supermercado.produto **on delete cascade**,

```
constraint fk_nro_lado_altura foreign key(nro, lado, altura) references Supermercado.prateleira on delete cascade);
```

```
create table Supermercado.categoria_simples(  
    nome Supermercado.CAT_NOME,  
    constraint pk_categoria_simples primary key (nome),  
    constraint fk_nome foreign key (nome) references Supermercado.categoria on delete cascade);
```

```
create table Supermercado.super_categoria(  
    nome Supermercado.CAT_NOME,  
    constraint pk_super_categoria primary key (nome),  
    constraint fk_nome foreign key (nome) references Supermercado.categoria on delete cascade);
```

```
create table Supermercado.constituída(  
    super_categoria Supermercado.CAT_NOME,  
    categoria Supermercado.CAT_NOME,  
    /* RI-EA2: super_categoria != categoria_simples */  
    constraint super_diff_sub check (super_categoria != categoria),  
    constraint fk_super_categoria foreign key (super_categoria) references  
    Supermercado.super_categoria(nome) on delete cascade,  
    constraint fk_categoria foreign key (categoria) references Supermercado.categoria(nome) on delete cascade,  
    constraint pk_constituída primary key(super_categoria, categoria));
```

```
create table Supermercado.fornece_sec(  
    nif Supermercado.NIF,  
    ean Supermercado.EAN,  
    constraint pk_fornece_sec primary key (nif, ean),  
    constraint fk_ean foreign key (ean) references Supermercado.produto on delete cascade,  
    constraint fk_nif foreign key (nif) references Supermercado.fornecedor on delete cascade);
```

```
create table Supermercado.evento_reposicao(  
    operador Supermercado.OPERADOR,  
    instante Supermercado.INSTANTE,  
    constraint pk_evento_reposicao primary key (operador, instante));
```

```
create table Supermercado.reposicao(  
    ean Supermercado.EAN,  
    nro Supermercado.NRO,  
    lado Supermercado.LADO,  
    altura Supermercado.ALTURA,  
    operador Supermercado.OPERADOR,  
    instante Supermercado.INSTANTE,  
    unidades int not null,  
    constraint pk_reposicao primary key (ean, nro, lado, altura, operador, instante),  
    constraint fk_ean_nro_lado_altura foreign key (ean, nro, lado, altura) references Supermercado.planograma on delete cascade,  
    constraint fk_operador_instante foreign key (operador, instante) references Supermercado.evento_reposicao on delete cascade);
```

- ★ O ficheiro schema.sql submetido contém algumas instruções extra que implementam algumas das restrições de integridade através de triggers. Contém também algumas instruções que eliminam as tabelas e os domínios com o mesmo nome aos criados no ficheiro, de modo a permitir a reconstrução da base de dados.

CONSULTAS SQL

- a. Qual o nome do fornecedor que forneceu o maior número de categorias?

```
WITH secundarios AS (  
    SELECT nif, categoria  
    FROM Supermercado.fornece_sec NATURAL JOIN Supermercado.produto  
    GROUP BY nif, categoria  
) , primarios AS (  
    SELECT forn_primario AS nif, categoria FROM Supermercado.produto  
    EXCEPT  
    SELECT nif, categoria FROM secundarios  
) , total_fornecedores AS (  
    SELECT nif, categoria FROM primarios  
    UNION  
    SELECT nif, categoria FROM secundarios  
)  
  
SELECT nome  
FROM total_fornecedores NATURAL JOIN Supermercado.fornecedor  
GROUP BY nif, nome  
HAVING COUNT(DISTINCT categoria) >= ALL (  
    SELECT COUNT(DISTINCT categoria)  
    FROM total_fornecedores  
    GROUP BY nif, nome  
);
```

- b. Quais os fornecedores primários (nome e nif) que forneceram produtos de todas as categorias simples?

```
SELECT *  
FROM Supermercado.fornecedor NATURAL JOIN (  
    SELECT forn_primario AS nif  
    FROM Supermercado.produto  
    WHERE categoria IN (  
        SELECT nome  
        FROM Supermercado.categoria_simples  
    )  
    GROUP BY forn_primario  
    HAVING COUNT(DISTINCT categoria) = (  
        SELECT COUNT (*)  
        FROM Supermercado.categoria_simples  
    )  
) AS fornecedor_categoria_simples;
```

- c. Quais os produtos (ean) que nunca foram repostos?

```
SELECT ean FROM Supermercado.produto  
EXCEPT  
SELECT ean FROM Supermercado.reposicao;
```

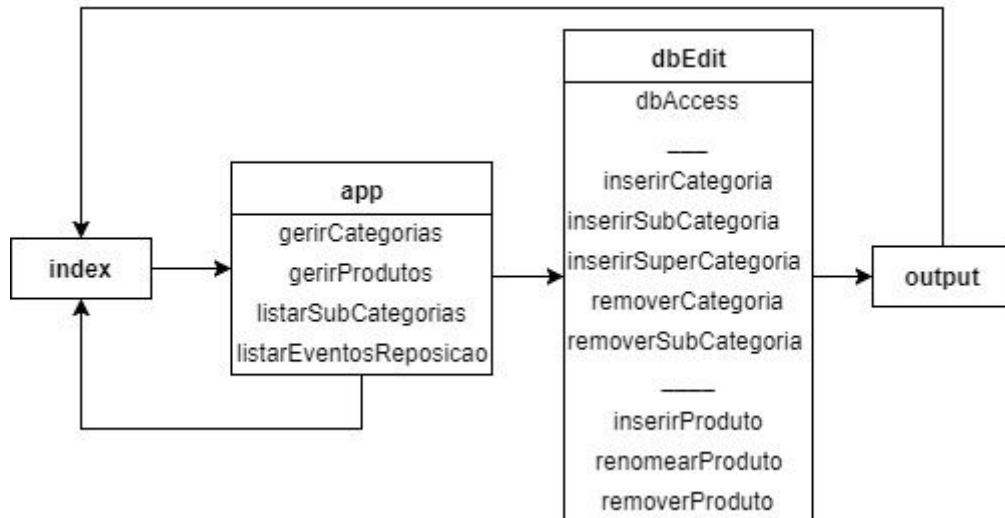
- d. Quais os produtos (ean) com um número de fornecedores secundários superior a 10?

```
SELECT ean  
FROM Supermercado.fornece_sec  
GROUP BY ean  
HAVING COUNT(nif) > 10;
```

- e. Quais os produtos (ean) que foram repostos sempre pelo mesmo operador?

```
SELECT ean  
FROM Supermercado.reposicao  
GROUP BY ean  
HAVING COUNT(DISTINCT operador) = 1;
```

APLICAÇÃO



A aplicação está dividida em 4 partes:

- **Index:**
 - “Main menu” da aplicação.
 - Agrupa e permite acesso os menus/operações básicas que é possível realizar com a aplicação.
- **app:**
 - Contém os menus para a realização das operações disponíveis no index.
 - Caso necessário: recolhe e verifica (parcial/totalmente) o “input” do utilizador, necessário para a execução da operação desejada.
 - Redireciona o input recebido para o **dbEdit**.
- **dbEdit:**
 - (Abreviação de “Database Edit”) Recebe input da **app**.
 - Executa as operações que exigem alterações na base de dados.
 - Faz algumas verificações e gera erros.
 - Redireciona o resultado da execução (sucesso ou insucesso) para o **output**.
- **output:**
 - Mostra o resultado da operação executada, sucesso ou insucesso com o respetivo erro.
 - Permite aceder ao **index** novamente.

app:

`gerirCategorias:`

- mostrar todas as categorias.
- permitir adicionar ou remover categorias e super categorias.
- permitir listar sub-categorias de uma super-categoria.

`gerirProdutos:`

- ver todos os produtos e respectiva informação.
- adicionar ou remover produtos.

`listarSubCategorias:`

- mostrar uma lista das sub-categorias de uma dada super-categoria
- permitir remover ou adicionar sub-categorias à super-categoria listada

`listarEventosReposição:` mostrar uma listagem dos eventos de reposição do supermercado

dbEdit:

`dbAccess:` funções de login, acesso e saída da base de dados.

`inserirCategoria:` insere uma categoria nova na base de dados.

`inserirSuperCategoria:` insere uma nova super-categoria com as respectivas sub-categorias.

`inserirSubCategoria:` associa uma categoria a uma super-categoria como sub-categoria desta.

`removerCategoria:` remove uma categoria pelo seu nome.

`removerSubCategoria:` remove a ligação de 1º nível entre uma super-categoria e uma sub-categoria.

`inserirProduto:` insere um produto novo na base de dados.

`removerProduto:` remove um produto da base de dados através do seu ean.

`renomearProduto:` altera a designação de um produto.

Diagrama da interação entre ficheiros detalhada:

