

# Projeto de Bases de Dados, Parte 3

Docente de Laboratório:

Professor Miguel Amaral

Nr. de Aluno	Nome	Percentagem Relativa de Contribuição	Esforço	Grupo	Turno
83449	Duarte Galvão	33,33%	13h00	28	Qua 11:00 - 12:30 LAB 8 (L07)
83496	Leonardo Epifânio	33,33%	13h00		
83540	Pedro Lopes	33,33%	13h00		

## Criação da Base de Dados

```
drop table if exists categoria cascade;
drop table if exists categoria_simples cascade;
drop table if exists super_categoria cascade;
drop table if exists constituida cascade;
drop table if exists produto cascade;
drop table if exists fornecedor cascade;
drop table if exists fornece_sec cascade;
drop table if exists corredor cascade;
drop table if exists prateleira cascade;
drop table if exists planograma cascade;
drop table if exists evento_reposicao cascade;
drop table if exists reposicao cascade;

create table categoria
  (nome varchar(50) not null unique,
   constraint pk_categoria primary key(nome));

create table categoria_simples
  (nome varchar(50) not null unique,
   constraint pk_categoria_simples primary key(nome),
   constraint fk_categoria_simples_categoria
     foreign key(nome) references categoria(nome)
     on delete cascade on update cascade);

create table super_categoria
  (nome varchar(50) not null unique,
   constraint pk_super_categoria primary key(nome),
   constraint fk_super_categoria_categoria
     foreign key(nome) references categoria(nome)
     on delete cascade on update cascade);

create table constituida
  (super_categoria varchar(50) not null,
   categoria varchar(50) not null unique,
   constraint pk_constituida primary key(super_categoria, categoria),
   constraint fk_constituida_super_categoria
     foreign key(super_categoria) references super_categoria(nome)
     on delete cascade on update cascade,
   constraint fk_constituida_categoria
     foreign key(categoria) references categoria(nome)
     on delete cascade on update cascade);
```

```

create table fornecedor
  (nif int not null unique,
   nome varchar(80) not null,
   constraint pk_fornecedor primary key(nif));

create table produto
  (ean bigint not null unique,
   design varchar(120) not null,
   categoria varchar(50) not null,
   forn_primario int not null,
   data date not null,
   constraint pk_produto primary key(ean),
   constraint fk_produto_categoria
     foreign key(categoria) references categoria(nome)
     on delete cascade on update cascade,
   constraint fk_produto_fornecedor
     foreign key(forn_primario) references fornecedor(nif)
     on delete cascade on update cascade);

create table fornece_sec
  (nif int not null,
   ean bigint not null,
   constraint pk_fornece_sec primary key(nif, ean),
   constraint fk_fornece_sec_fornecedor
     foreign key(nif) references fornecedor(nif)
     on delete cascade on update cascade,
   constraint fk_fornece_sec_produto
     foreign key(ean) references produto(ean)
     on delete cascade on update cascade);

create table corredor
  (nro int not null unique,
   largura int not null,
   constraint pk_corredor primary key(nro));

create table prateleira
  (nro int not null,
   lado smallint not null,
   altura smallint not null,
   constraint pk_prateleira primary key(nro, lado, altura),
   constraint fk_prateleira_corredor
     foreign key(nro) references corredor(nro)
     on delete cascade on update cascade);

```

```

create table planograma
  (ean bigint not null,
   nro int not null,
   lado smallint not null,
   altura smallint not null,
   face int not null,
   unidades int not null,
   loc int not null,
   constraint pk_planograma primary key(ean, nro, lado, altura),
   constraint fk_planograma_producto
     foreign key(ean) references produto(ean)
     on delete cascade on update cascade,
   constraint fk_planograma_prateleira
     foreign key(nro, lado, altura)
     references prateleira(nro, lado, altura)
     on delete cascade on update cascade);

create table evento_reposicao
  (operador varchar(80) not null,
   instante timestamp not null,
   constraint pk_evento_reposicao primary key(operador, instante));

create table reposicao
  (ean bigint not null,
   nro int not null,
   lado smallint not null,
   altura smallint not null,
   operador varchar(80) not null,
   instante timestamp not null,
   unidades int not null,
   constraint pk_reposicao
     primary key(ean, nro, lado, altura, operador, instante),
   constraint fk_reposicao_planograma
     foreign key(ean, nro, lado, altura)
     references planograma(ean, nro, lado, altura)
     on delete cascade on update cascade,
   constraint fk_reposicao_evento_reposicao
     foreign key(operador, instante)
     references evento_reposicao(operador, instante)
     on delete cascade on update cascade);

```

# SQL

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

```
select nome
from (
  select nif
  from (
    (select forn_primario as nif, categoria from produto)
    union (
      select nif, categoria
      from fornece_sec natural join
      (select ean, categoria from produto) as cat_produto
    )
  ) as categorias_fornecedores
  group by nif
  having count(categoria) >= all(
    select distinct count(categoria)
    from (
      (select forn_primario as nif, categoria from produto)
      union (
        select nif, categoria
        from fornece_sec natural join
        (select ean, categoria from produto) as cat_produto
      )
    ) as categorias_fornecedores
    group by nif
  )
) as nifs_forneceram_mais_categorias natural join fornecedor;
```

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

```
select nif, nome
from (
  select forn_primario as nif
  from produto
  where categoria in (
    select nome
    from categoria_simples
  )
  group by nif
  having count(categoria) = (
    select count(nome)
    from categoria_simples
  )
) as nifs_division natural join fornecedor;
```

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

```
select ean
from produto
where ean not in (
    select ean
    from reposicao);
```

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

```
select ean
from fornece_sec
group by ean
having count(nif) > 10;
```

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

```
select ean
from (
    select distinct ean, operador
    from reposicao
) as ean_operador
group by ean
having count(operador) = 1;
```

# Arquitetura da Aplicação PHP

A aplicação é composta pelo ficheiro index.html, que contém o menu inicial, pelos ficheiros PHP correspondentes a cada uma das operações possíveis (alíneas a), b), c), d) e e)), e pelo ficheiro config.php que contém o script para aceder à base de dados e criar o PDO. Este ficheiro é included em cada um dos ficheiros PHP.

Cada página das operações tem vários modos, que definem o seu estado. Em qualquer situação existe a possibilidade de voltar ao menu anterior.

**a.php:** A estrutura é definida por 4 modos. O modo default é um menu inicial com todas as operações necessárias para criar uma categoria, definindo o nome, e escolhendo uma ou nenhuma das categorias existentes para categoria-mãe. Ao fazer isto e clicar em submit, o modo é alterado para add\_cat, que irá realizar o insert na DB. No menu default existe também uma lista de categorias e um form para procurar uma categoria pelo nome, para a remoção de categorias. Quando se efetua uma procura, o modo é alterado para search, sendo efetuada a procura na DB e o display do resultado. Ao clicar em algum dos botões 'Remover' a categoria é removida da DB se for possível, e é feito o display novamente.

**b.php:** O modo inicial (home) tem duas opções, 'Inserir produto' e 'Remover produto'. Ao clicar na primeira opção, o modo é mudado para insert\_form e é feito display de um form com campos a preencher para o EAN, Designação, Data, dois drop down para Categoria e Fornecedor Primário, e uma tabela com checkboxes para escolher os Fornecedores Secundários. Ao fazer submit (modo insert\_product), é inserido (se possível) o novo produto na base de dados, fazendo display de uma mensagem de feedback. No menu de 'Remover Produto' (modo remove\_form), é apresentado um form para procurar o produto por EAN e uma tabela com todos os produtos. Ao clicar num dos botões 'Remover', o produto correspondente é removido, quando se efetua a procura, o comportamento é análogo ao de a.php.

**c.php:** No modo default (home) é apresentado um form de procura por EAN, com um comportamento análogo aos referidos anteriormente, e uma tabela com todos os produtos na DB e a opção 'Listar reposições' para cada um. Ao clicar em 'Listar reposições', o modo é alterado para list, e é feito o display dos eventos de reposição do produto correspondente.

**d.php:** O menu inicial (modo home) contém um form de procura por EAN e uma tabela com os produtos e com as opções para alterar a designação de cada produto. Ao clicar em 'Alterar', é feito o display de outro form para a nova designação do produto correspondente (modo show). Ao fazer submit, alterna-se para o modo change e uma mensagem de feedback aparece no ecrã.

**e.php:** No modo home é apresentado, tal como em c.php e d.php um form de procura (desta vez por nome) e uma tabela com as categorias e as opções para listar sub-categorias. Ao clicar num destes botões é feito o display dos nomes das sub-categorias correspondentes (modo list).