

PRÁTICA LABORATORIAL 02

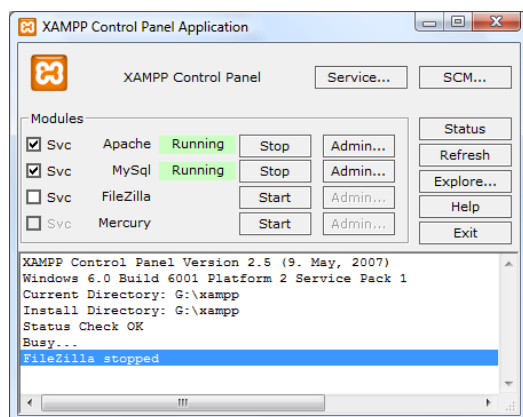
Objetivos:

Criar uma base de dados em MySQL ou SQLServer

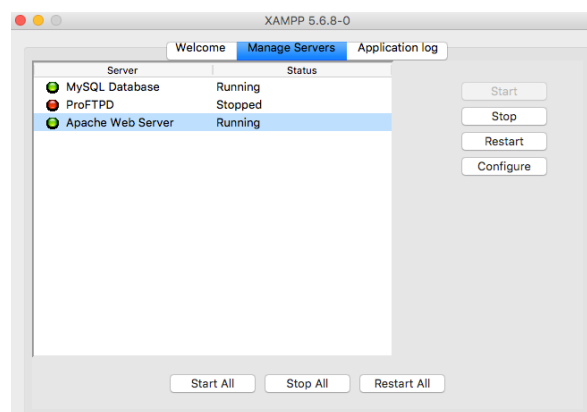
Bom trabalho! 😊

EXERCÍCIOS

1. Instale o XAMPP. (Fazer download de <https://www.apachefriends.org/index.html>)
2. Inicie os serviços Apache e MySQL.



Windows



Mac OS

3. Aceda ao gestor do MySQL através do browser:
 - a. localhost/phpmyadmin



4. Clique no menu SQL e execute o seguinte código no MySQL para criar a Base de Dados:

```
create database ExemploTeorica01;
```

5. Clique na base de dados que acabou de criar (aparece do lado esquerdo – se não aparecer clique em atualizar), clique no menu SQL e execute o seguinte código no MySQL para criar as tabelas:

```
create table Cliente(  
    idCliente int AUTO_INCREMENT PRIMARY KEY,  
    nome varchar(50),  
    cidade varchar(50),  
    codPostal int,  
    cc int unique not null  
);  
  
create table Venda(  
    idVenda int AUTO_INCREMENT PRIMARY KEY,  
    cliente int,  
    foreign key (cliente) references Cliente(idCliente),  
    data datetime,  
    desconto decimal(5,2)  
);  
  
create table Artigo(  
    idArtigo int AUTO_INCREMENT PRIMARY KEY,  
    nome varchar(50),  
    preco decimal(10,2),  
    categoria varchar(20)  
);  
  
create table LinhaVenda(  
    venda int,  
    foreign key (venda) references Venda(idVenda),  
    artigo int,  
    foreign key (artigo) references Artigo(idArtigo),  
    quantidade int,  
    primary key (venda, artigo)  
);
```

6. Execute o seguinte código para inserir dados nas tabelas: **(ATENÇÃO às plicas quando copiadas do PDF)**

```
insert into Cliente (nome, cidade, codPostal, CC) values
('Ana', 'Maia', 4567, 123),
('Bruno', 'Porto', 4200, 456),
('Miguel', 'Porto', 4200, 789),
('Maria', 'Maia', 4567, 234),
('Joana', 'Maia', 4567, 678);

insert into Artigo (nome, preco, categoria) values
('Martelo', 5.5, 'Ferramentas'),
('Parafuso', 0.5, 'Ferramentas'),
('Prego', 0.2, 'Ferramentas'),
('Lampada', 1.5, 'Electricidade'),
('Tripla', 5.5, 'Electricidade');
insert into Venda (cliente, data, desconto) values
(1, '2021-03-29', 0.05),
(1, '2021-03-31', 0.20),
(2, '2021-03-29', 0.5),
(4, '2021-03-29', 0.5);

insert into LinhaVenda (venda, artigo, quantidade) values
(1, 1, 2),
(1, 3, 10),
(2, 2, 20),
(3, 3, 100),
(4, 1, 5);
```

7. Execute o seguinte código para fazer pesquisas e tente perceber a sua funcionalidade:

```
SELECT *
FROM Artigo

SELECT idArtigo, nome, preco
FROM Artigo
WHERE preco<5

SELECT nome, preco
FROM Artigo
WHERE preco>=2 AND preco<=10

SELECT nome, preco
FROM Artigo
WHERE preco BETWEEN 2 AND 10

SELECT nome, preco
FROM Artigo
WHERE preco<5 OR preco>10

SELECT nome, preco
FROM Artigo
WHERE preco NOT BETWEEN 5 AND 10

SELECT nome, preco
FROM Artigo
WHERE preco IN (10,100)
```

```
SELECT *
FROM Artigo
WHERE nome LIKE 'M%'

SELECT * FROM Artigo
ORDER BY preco ASC

SELECT * FROM Artigo
ORDER BY preco DESC

SELECT * FROM Artigo
WHERE preco >500
ORDER BY preco DESC

SELECT COUNT(nome) AS TotalArtigos
FROM Artigo

SELECT SUM(preco) AS SomaPV
FROM Artigo

SELECT AVG(preco) AS MediaPV
FROM Artigo

SELECT MAX(preco) AS MaximoPV
FROM Artigo

SELECT MIN(preco) AS MinimoPV
FROM Artigo

SELECT cidade
FROM Cliente

SELECT DISTINCT cidade
FROM Cliente

SELECT Cliente.nome, Venda.data, Artigo.nome
FROM Artigo, Cliente, Venda, LinhaVenda
WHERE Cliente.idCliente = Venda.cliente AND
Venda.idVenda = LinhaVenda.venda AND
LinhaVenda.artigo = Artigo.idArtigo

SELECT c.nome, v.data, a.nome
FROM Artigo a, Cliente c, Venda v, LinhaVenda l
WHERE c.idCliente = v.cliente AND
v.idVenda = l.venda AND
l.artigo = a.idArtigo

SELECT c.nome, v.data
FROM Cliente c
INNER JOIN Venda v ON c.idCliente = v.cliente

SELECT c.nome, v.data
FROM Cliente c
LEFT JOIN Venda v ON c.idCliente = v.cliente

SELECT c.nome, v.data, SUM(l.quantidade*a.preco) AS Total
FROM Artigo a, Cliente c, Venda v, LinhaVenda l
WHERE c.idCliente = v.cliente AND v.idVenda = l.venda AND l.artigo = a.idArtigo
GROUP BY c.nome, v.data
```

```
SELECT c.nome, v.data
FROM Artigo a, Cliente c, Venda v, LinhaVenda l
WHERE c.idCliente = v.cliente AND v.idVenda = l.venda AND l.artigo = a.idArtigo
ORDER BY v.data DESC
LIMIT 1

SELECT c.codPostal, COUNT(c.codPostal)
FROM Cliente c
GROUP BY c.codPostal
HAVING COUNT(c.codPostal) > 2;

SELECT a.categoria, COUNT(a.nome) AS TotalArtigos
FROM Artigo a
GROUP BY a.categoria

SELECT c.nome
FROM Cliente c
WHERE c.idCliente NOT IN (SELECT cliente FROM Venda)

SELECT nome, (SELECT COUNT(*) FROM Venda WHERE Venda.cliente = Cliente.idCliente)
FROM Cliente

SELECT nome
FROM Artigo
WHERE preco >= ALL (SELECT preco
FROM Artigo)

SELECT nome
FROM Artigo
WHERE preco > ANY (SELECT preco
FROM Artigo)

SELECT c.nome FROM Cliente c
UNION
SELECT a.nome FROM Artigo a
```

Bom trabalho! ☺

PRÁTICA LABORATORIAL 03

Objetivos:

Criar uma base de dados em MySQL ou SQLServer

Bom trabalho! 😊

EXERCÍCIOS

1. Acede ao gestor do MySQL através do browser:
 - a. localhost/phpmyadmin
2. Crie a base de dados *Empresa-SeuNome* (CREATE DATABASE).
3. Crie as seguintes tabelas (CREATE TABLE), defina os tipos de dados de acordo com os valores das tabelas abaixo e as relações com as respetivas chaves estrangeiras.
 - a. **seccao**={numSec, nome, cidade}
 - b. **empregado**={numEmp, nome, *seccao*, posto, chefe, salario, comissao}

HUMSEC	HOME	CIDADE
10	Fabrico	Porto
20	Comercial	Porto
30	Marketing	Braga
40	Planeamento	Guimarães
50	Administração	Porto
60	Informática	Braga
70	Recursos Humanos	Guimarães

HUMEMP	HOME	SECCAO	POSTO	CHEFE	SALARIO	COMISSAO
1	Ana	10	Programador	3	3000	10
2	Nuno	70	Engenheiro	1	1500	40
3	Álvaro	50	Administrador	-	2500	0
4	António	10	Engenheiro	3	1450	20
5	Susana	20	Administrador	-	2750	30
6	Cláudio	60	Vendedor	4	1000	50

(NOTA: quando aparece um ífen “-”, significa que não precisa de ser preenchido. Pode ser substituído por **null**)

4. Execute o seguinte código SQL e indique para cada consulta a sua função. **(ATENÇÃO ÀS PLICAS QUANDO COPIADAS DO PDF!)**

- a. A query retorna o nome e cidade da tabela seccao, que foi abreviado para "s" onde o numero de seccao é igual a 70

```
select s.nome, s.cidade  
from seccao s  
where s.numsec=70
```

- b. A query não retorna nenhum valor porque nessa tabela não existe comissão maior que salário

```
select e.posto, e.salario, e.comissao  
from empregado e  
where e.COMISSAO>e.salario
```

- c. A query retorna o nome onde cidade tem o nome porto

```
select s.nome  
from seccao s  
where s.cidade like 'Porto'
```

- d. A query retorna nome, salario e comissao da tabela empregado que virou "e" onde salario é igual ou maior que 20 multiplicado por comissao ou salario igual 2000

```
select e.nome, e.salario, e.comissao  
from empregado e  
where e.salario>=20*e.comissao or e.salario=2000
```

- e. A query retorna apenas um conteúdo dos atributos

```
select distinct e.posto  
from empregado e
```

- f. A query retorna nome, posto, e salario da tabela empregado onde seccao não é igual a 70 ordenando posto e salario decrescente

```
select e.nome, e.posto, e.salario  
from empregado e  
where e.seccao!=70  
order by e.posto, e.salario desc
```

- g. a query retorna o nome dos empregados onde não são do porto

```
select e.nome  
from empregado e  
where e.seccao not in (select s.numsec  
                      from seccao s  
                      where s.cidade like 'Porto');
```

h. A query

```
select e.nome  
from empregado e  
where e.nome like 'A%a'
```

i. A query retorna nomes que tenham "A" maiúsculo e "a" minúsculo

```
select e.nome, e.salario, f.nome, f.salario  
from empregado e, empregado f  
where e.chefe = f.numemp and e.salario>f.salario
```

j. A query retorna o nome e salario dos funcionario que tem o mesmo chefe que salario seja maior que o salario

```
select e.nome  
from empregado e  
where e.seccao!=10 and e.posto in (select posto from empregado where seccao=10)
```

k. A query retorna nome na tabela empregado onde seccao é diferente de 10 e posto igual ao posto dentro do parentese que é igual a 10

```
select e.nome, f.nome  
from empregado e, empregado f  
where e.chefe = f.numemp and e.salario>=f.salario*2  
order by (e.salario-f.salario) desc
```

l. A query retorna nome de funcionario onde tem o mesmo chefe e salario é maior e igual ou maior multiplicado por 2 ordenando em decrescente o resto da subtração dos mesmo

```
select e.nome, e.posto, e.seccao  
from empregado e  
where e.seccao in (20, 30, 40)
```

ou

```
select e.nome, e.posto, e.seccao  
from empregado e  
where e.seccao=20 or e.seccao=30 or e.seccao=40
```

m. a retorna o nome posto e seccao do empregado onde seccao 20 ou 30 ou 40 entao retorna a primeira nessas condições

```
select s.nome, s.cidade  
from seccao s, empregado e  
where e.seccao=s.numsec
```

n. A query retorna nome e cidade da seccao onde seccao é igual o numero da seccao

```
select sum(e.salario*14+e.salario*12*0.5)  
from empregado e  
where e.posto in (select posto from empregado where posto like 'Engenheiro' or posto like 'Programador')
```


o. _____

```
select e.seccao, count(e.nome) as num_emp
from empregado e
group by e.seccao
```

p. _____

```
select sum(e.salario) as salario_tot
from empregado e
group by e.seccao
```

q. _____

```
select e.posto, count(e.nome) as num_emp
from empregado e
group by e.posto
```

r. _____

```
select e.posto, avg(e.salario) as med_salario
from empregado e
group by e.posto
```

s. _____

```
select e.posto, max(e.salario) as max_salario
from empregado e
group by e.posto
```

t. _____

```
select e.posto, count(e.nome) as num_emp
from empregado e
group by e.posto
having count(e.nome)>=2
```

u. _____

```
select e.posto, avg(e.salario) as med_salario
from empregado e
group by e.posto
having count(e.nome)>=2
```

Bom trabalho! 😊

PRÁTICA LABORATORIAL 04

Objetivos:

Criar uma base de dados em MySQL ou SQLServer

Bom trabalho! 😊

EXERCÍCIOS

1. Crie a base de dados chamada GestComercial_*seuNome*.
2. Crie as seguintes tabelas com as respectivas características:

Cliente

Campo	Restrição
id	Número inteiro, campo automático. (Chave primária)
nome	Permite 50 caracteres e é obrigatório
morada	Permite 250 caracteres
estado	Bit (só permite valores 1 ou 0)

Produto

Campo	Restrição
id	Número inteiro, campo automático. (Chave primária)
descricao	Permite 100 caracteres e é obrigatório.
preco	Número decimal com no máximo 10 dígitos inteiros mais 2 casas decimais.

Venda

Campo	Restrição
prod	Número inteiro (Chave primária e estrangeira)
cli	Número inteiro (Chave primária e estrangeira)
data	Data e Hora (Chave primária)
qtd	Número inteiro que significa a quantidade.

3. Insira na tabela os seguintes valores:

Cliente

id	nome	morada	estado
1	Helena Monteiro	Perafita – Freixieiro	1
2	Raul Simas	Palmela	0

ATENÇÃO: os campos “id” que são incrementados automaticamente não se preenchem no INSERT.

Produto

id	descricao	preco
1	Rato Logitech XPTO 1	10
2	Monitor Sony Vaio XPTO	120

Venda

cli	prod	qtd	data
1	1	1	05-04-2012
2	1	5	05-04-2012

ATENÇÃO: os campos “data” têm de estar dentro de plicas (‘) e no formato ANO-MÊS-DIA.

4. Proceda às seguintes alterações de dados:
 - a. Na venda nr. 2, altere a quantidade para 10.
 - b. Mude a morada do Raul para Porto.
 - c. Altere o preço do monitor para 100euros.
5. Escreva as seguintes consultas SQL:
 - a. Mostre todos os produtos da loja.
 - b. Mostre todos os nomes dos clientes ativos.
 - c. Mostre todas as vendas realizadas ao cliente Raul Simas.
 - d. Mostre todas as vendas realizadas aos clientes com nome começado pela letra H.
 - e. Mostre quantas vendas foram realizadas ao cliente Raul Simas.
 - f. Mostre o valor do produto mais caro na loja.
 - g. Mostre quais os produtos cujo preço é menor ou igual a 100 euros.
 - h. Mostre os nomes dos clientes que nunca compraram nada.
 - i. Mostre o total faturado a cada dia.

Bom trabalho! 😊

PRÁTICA LABORATORIAL 05

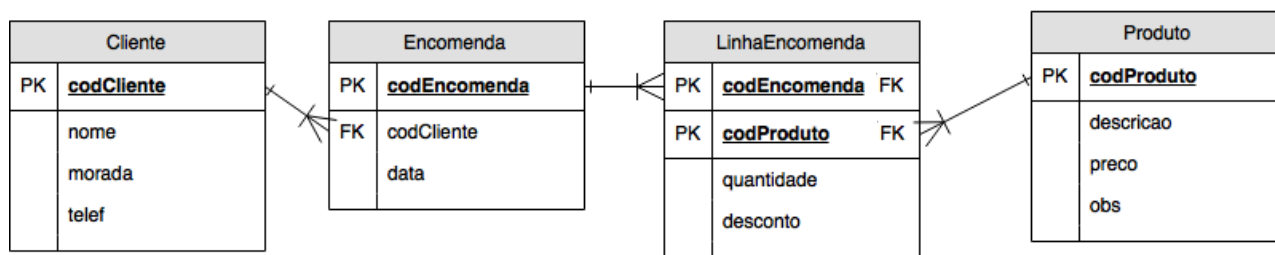
Objetivos:

Criar uma base de dados em MySQL ou SQLServer

Bom trabalho! 😊

EXERCÍCIOS

1. Crie a base de dados Encomendas_*seuNome*.
2. Crie as seguintes tabelas, não esquecendo de definir os tipos de dados de acordo com o que considerar mais adequado para os possíveis valores dos campos das tabelas abaixo, assim como as relações com as respetivas chaves estrangeiras.



3. Insira três registos ao seu gosto em cada tabela.
4. Crie as seguintes consultas SQL:
 - a. Mostre da tabela de Clientes, os clientes cuja morada é Solar do Monte, ou outra a gosto, organizados por nome.
 - b. Mostre da tabela de Produtos, o produto cuja Descrição é martelo, ou outra a gosto, organizado por preço.
 - c. Mostre todas as datas das encomendas realizadas pelo cliente Fonseca, ou outro a gosto.
 - d. Mostre quantos produtos o cliente Silva, ou outro a gosto, efetuou encomenda, agrupado por descrição de Produto.
 - e. Mostre o total pago em cada encomenda realizada.

Bom trabalho! 😊

PRÁTICA LABORATORIAL 06

Objetivos:

Criar uma base de dados em MySQL ou SQLServer

Bom trabalho! 😊

EXERCÍCIOS

1. Crie a base de dados Campeonato_*seuNome*.
2. Crie as seguintes tabelas, não esquecendo de definir os tipos de dados de acordo com o que considerar mais adequado para os possíveis valores dos campos das tabelas abaixo, assim como as relações com as respetivas chaves estrangeiras.
 - a. equipas (id Equipa, nome, cidade, diretor)
 - b. treinadores (id treinador, nome, idade, telefone)
 - c. bolas (referencia, id Equipa, fabricante)
 - d. experiencias (id Equipa, id treinador, escalao, anos)
3. Inserir os registos apresentados em cada uma das tabelas criadas. **(atenção às plicas quando copiadas do PDF!)**

```
INSERT INTO treinadores VALUES(1, 'António', 34, 922424561);
INSERT INTO treinadores VALUES(2, 'Barbosa', 45, 965552936);
INSERT INTO treinadores VALUES(3, 'Tavares', 48, 933332267);
INSERT INTO treinadores VALUES(4, 'Joaquim', 57, 918638465);
INSERT INTO treinadores VALUES(5, 'Alberto', 33, 966785309);
INSERT INTO treinadores VALUES(6, 'Duarte', 54, 912773446);

INSERT INTO equipas VALUES(12, 'Académico', 'Porto', 'Mário');
INSERT INTO equipas VALUES(15, 'Universitário', 'Coimbra', 'João');
INSERT INTO equipas VALUES(20, 'Juventude', 'Braga', 'Silva');
INSERT INTO equipas VALUES(24, 'Tigres', 'Espinho', 'Cardoso');
```

```
INSERT INTO bolas VALUES(1, 12, 'Adidas');  
INSERT INTO bolas VALUES(9, 12, 'Reebok');  
INSERT INTO bolas VALUES(13, 12, 'Adidas');  
INSERT INTO bolas VALUES(1, 15, 'Adidas');  
INSERT INTO bolas VALUES(3, 20, 'Olimpic');  
INSERT INTO bolas VALUES(4, 20, 'Nike');  
INSERT INTO bolas VALUES(18, 24, 'Reebok');  
INSERT INTO bolas VALUES(21, 24, 'Olimpic');  
  
INSERT INTO experiencias VALUES(12, 1, 'juniores', 10);  
INSERT INTO experiencias VALUES(12, 1, 'seniores', 5);  
INSERT INTO experiencias VALUES(12, 2, 'iniciados', 2);  
INSERT INTO experiencias VALUES(12, 2, 'juniores', 3);  
INSERT INTO experiencias VALUES(12, 2, 'juvenis', 4);  
INSERT INTO experiencias VALUES(15, 3, 'juniores', 15);  
INSERT INTO experiencias VALUES(24, 5, 'juvenis', 12);
```

4. Crie as seguintes consultas:

- a. Mostrar todos os dados da tabela Equipas;
- b. Mostrar todos os dados da equipa com o id igual a 12;
- c. Mostrar o id e o nome de todas as equipas;
- d. Mostrar o id, o nome e a idade dos treinadores com menos de 40 anos de idade;
- e. Mostrar todos os dados da tabela Experiencias relativos aos treinadores que treinaram juniores ou que tenham mais do que 10 anos de experiência;
- f. Mostrar todos os dados dos treinadores com idade pertencente ao intervalo [45, 53] e por ordem decrescente da idade;
- g. Mostrar todos os dados das bolas dos fabricantes Reebok e Olimpic;
- h. Mostrar todos os dados dos treinadores cujo nome começa pela letra A.
- i. Mostrar a quantidade de equipas que disputam o campeonato;
- j. Mostrar a quantidade de fabricantes distintos que produzem bolas usadas no campeonato;
- k. Mostrar a quantidade de treinadores com idade superior a 40 anos;
- l. Mostrar a idade do treinador mais velho.
- m. Mostrar o id das equipas que utilizam bolas do fabricante Adidas;

- n. Mostrar o resultado da alínea anterior, mas sem repetições;
- o. Mostrar a média das idades dos treinadores de juvenis;
- p. Mostrar todos os dados dos treinadores e das equipas por eles treinadas;
- q. Mostrar os nomes e os telefones dos treinadores e os nomes das equipas por eles treinadas;
- r. Mostrar todos os dados da equipa do Académico e dos respetivos treinadores;
- s. Mostrar a idade do treinador mais velho do Académico;
- t. Mostrar o total de anos de experiência do treinador António do Académico.

Bom trabalho! 😊

PRÁTICA LABORATORIAL 07

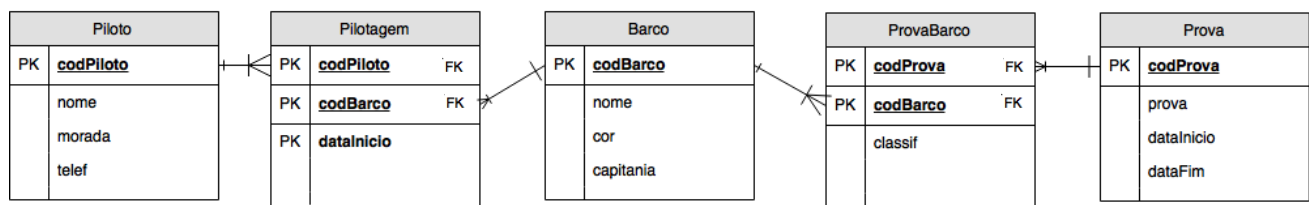
Objetivos:

Criar uma base de dados em MySQL ou SQLServer

Bom trabalho! 😊

EXERCÍCIOS

1. Crie a base de dados *ProvasBarcos_seuNome*.
2. Crie as seguintes tabelas, não esquecendo de definir os tipos de dados de acordo com o que considerar mais adequado para os possíveis valores dos campos das tabelas abaixo, assim como as relações com as respetivas chaves estrangeiras.



3. Insira três registos ao seu gosto em cada tabela.
4. Crie as seguintes consultas SQL:
 - a. Mostrar todos os campos dos barcos da capitania do Porto.
 - b. Mostrar todos os campos dos barcos das capitánias da seguinte lista: Porto, Aveiro, Algarve.
 - c. Mostrar os nomes e telefones de todos os pilotos.
 - d. Mostrar os nomes das provas que decorrem neste mês.
 - e. Mostrar para cada Prova (prova, data de início e data de fim) os barcos (nome) que nela participam e respetiva classificação. Esta consulta deve vir ordenada de forma descendente pelo campo data de início.
 - f. Mostrar em quantas provas participou cada piloto.

Bom trabalho! 😊

PRÁTICA LABORATORIAL 08

Objetivos:

Criar uma base de dados em MySQL ou SQLServer

Bom trabalho! 😊

EXERCÍCIOS

1. Crie uma Base de Dados com o nome *Organizacao_seuNome*
2. Crie as seguintes tabelas, considerando que os atributos sublinhados são chaves primárias e os em itálico são chaves estrangeiras, bem como analise os dados apresentados nas tabelas para decidir que tipo de dados define em cada campo:
 - a. Trabalhador (numero, nome, dataNasc, especialidade)
 - b. Produto (codigoP, designacaoP, area, *numero*)

O campo "numero" indica o Trabalhador responsável por esses Produto.
 - c. Comprador (codigoC, identificacaoC, moradaC, telefoneC)
 - d. Venda (codigoP, codigoC, dataV, quantidadeV, valorV, dataRecebim)
 - e. MateriaPrima (codigoMp, designacaoMp, quantArmaz)
 - f. Utiliza (codigoP, codigoMp, dataU, quantidadeU)
 - g. Fornecedor (codigoF, identificacaoF, moradaF, TelefoneF)
 - h. Compra (codigoMp, codigoF, dataC, quantidadeC, valorC, dataPagam)
3. Inserir os registos apresentados em cada uma das tabelas criadas (as que não tiverem registos, deve inventar).

Tabela Produto

codigoP	designacaoP	area	numero
5	cenoura	4.5	4
6	feijão verde	6.5	1
7	batata	2.6	4
8	rosa	1	2

Tabela Trabalhador

numero	nome	dataNasc	especialidade
1	João Dias	1955-01-05	hortelão
2	Luísa Ferreira	1945-11-25	jardineiro
3	Mário Horta	1960-12-05	motorista
4	Joana Garcia		eng agrónomo

Tabela Comprador

codigoC	identificacaoC	moradaC	telefoneC
1	Manel Maria	Casais de Cima	749658365
2	Luisa Fraga	Casais de Baixo	589016587
3	Duarte Silva	Vila Nova	456123789

Tabela Venda

codigoP	codigoC	dataV	quantidadeV	valorV	dataRecebim
5	1	1999-04-07	50	45000	1999-04-07
5	2	1999-02-20	5.6	1000	
6	2	1999-04-05	100	60000	
6	1	1999-03-29	450	90000	

Tabela Fornecedor

codigoF	identificacaoF	moradaF	telefoneF
1	Ana Sousa	ISA - Tapada da Ajuda	213638161

4. Crie as seguintes consultas:
- Qual a área do produto rosa?
 - Quais os funcionários com especialidade “jardineiro”.
 - De onde é o Comprador Duarte Silva e qual o teu contacto?
 - Qual o valor da venda mais cara?
 - Quais os nomes dos clientes que nunca fizeram qualquer compra?

Bom trabalho! ☺