

Exame 2017/2018

1. A segunda forma normal exige que os atributos não-primas não sejam funcionalmente dependentes de um subconjunto próprio de qualquer chave candidata, ou seja, todos os seus atributos não primas devem ser totalmente dependentes da sua chave primária.

NrAluno	NrDisciplina	Designação	Nota	Professor
1	1		1	

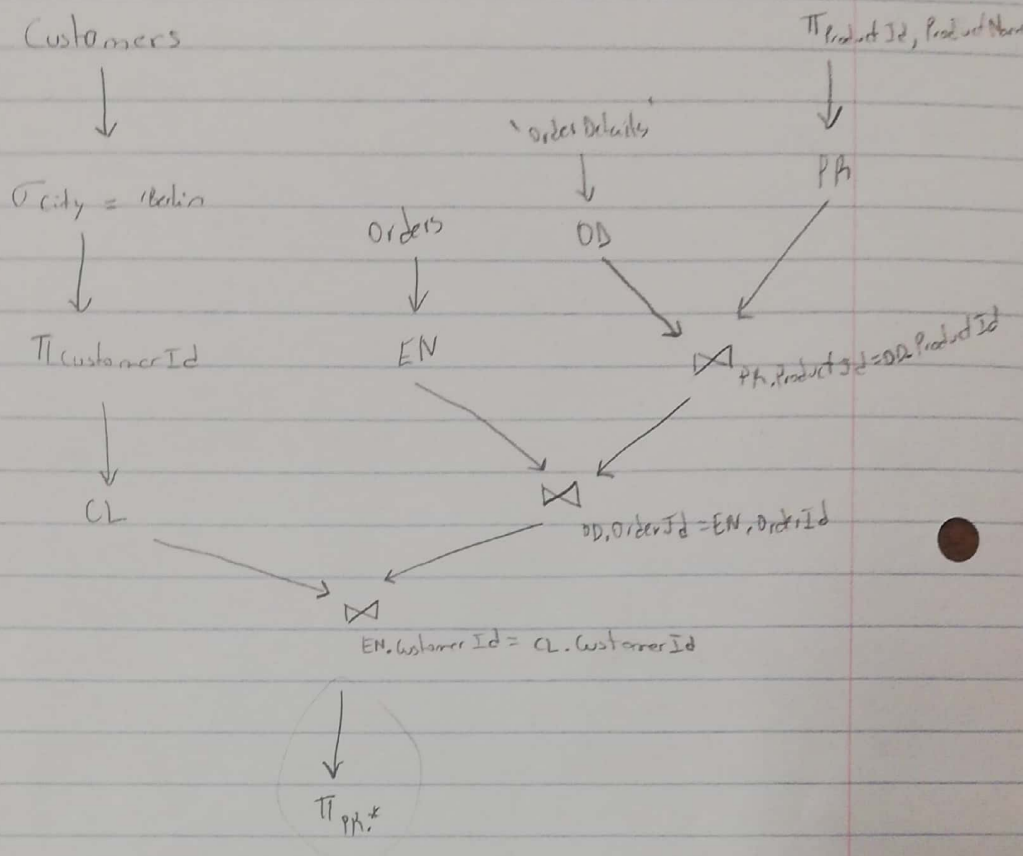
NrAluno	NrDisciplina	Nota

NrDisciplina	Designação	Professor

2. As restrições de integridade garantem a correção e consistência da base de dados, verificando que qualquer operação permitida não a pode deixar num estado incorreto.
- Os principais tipos de integridade são:

- integridade de domínio - qualquer atributo tem um domínio e as restrições devem ser identificadas no momento da escolha dos domínios.
- integridade de entidades - a chave primária de uma relação não pode admitir valores nulos; este tipo de restrição deve ser considerado no momento em que se faz a definição das chaves primárias para cada tipo de entidade.
- integridade referencial - se uma chave estrangeira contém um valor, esse valor deve existir na relação pai para que seja aceite na chave estrangeira.
- restrições de organizações - regras da própria organização que são aplicadas para regular o seu funcionamento.

3.



$\pi_{PR,*} (CL \bowtie_{EN.CustomerId = CL.CustomerId} (EN \bowtie_{OD.OrderId = EN.OrderId} (OD \bowtie_{PR.ProductId = OD.ProductId} PR)))$

$CL \leftarrow \pi_{CustomerId} (\sigma_{City = 'Berlin'} (Customers))$

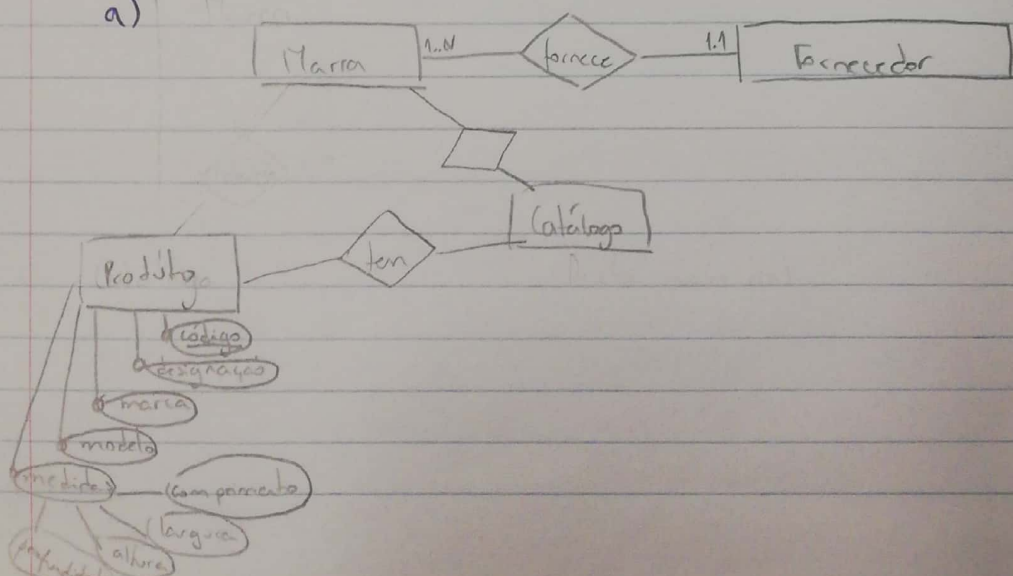
$EN \leftarrow Orders$

$OD \leftarrow 'Order Details'$

$PR \leftarrow \pi_{ProductId, ProductName} (Products)$

4.

a)



5.

```
a) CREATE SCHEMA myDB;  
USE myDB;
```

```
CREATE TABLE Veiculo (  
    idVeiculo INT NOT NULL,  
    'Matrícula' VARCHAR(25) NOT NULL,  
    Marca VARCHAR(45) NOT NULL,  
    Modelo VARCHAR(75),  
    'Características' TEXT,  
    primary key (idVeiculo) );
```

```
CREATE TABLE Equipamento (  
    idEquipamento INT NOT NULL,  
    'Designação' VARCHAR(75) NOT NULL,  
    'Horas Operações' DECIMAL(8,2) NOT NULL,  
    primary key (idEquipamento) );
```

```
CREATE TABLE 'Funcionário' (  
    'idFuncionário' INT NOT NULL,  
    Nome VARCHAR(45) NOT NULL,  
    'Função' VARCHAR(25) NOT NULL,  
    primary key ('idFuncionário') );
```

```
CREATE TABLE 'Intervenção' (  
    'idVeiculo' INT NOT NULL,  
    'idFuncionário' INT NOT NULL,  
    'idEquipamento' INT NOT NULL,  
    HorasTrabalho DECIMAL(5,2),  
    primary key ('idVeiculo', 'idFuncionário', 'idEquipamento'),  
    foreign key ('idVeiculo') references Veiculo(idVeiculo),  
    ...  
);
```


b) Select Name From Funcionário as F
INNER JOIN 'Intervenções' as I
ON F.idFuncionário = I.idFuncionário
INNER JOIN 'Veículo' as V
ON V.idVeículo = I.idVeículo
Where V.Marca = 'Ford' ;

c) Create Trigger higo
on Intervenções
After insert
as

6.

7.

a) Apresenta todos os documentos ^{= objetos} da coleção 'oficiais' cuja
'classificação' é inferior a 2.

b) Apresenta os documentos da coleção 'oficiais' que têm tipo
letras ou Código Postal 4700.