



# Programação e Sistemas de Informação

CURSO PROFISSIONAL TÉCNICO DE  
GESTÃO E PROGRAMAÇÃO DE SISTEMAS INFORMÁTICOS

## Técnicas de Modelação de Dados

### MÓDULO 13

Professor: João Martiniano

# Conteúdos do Módulo

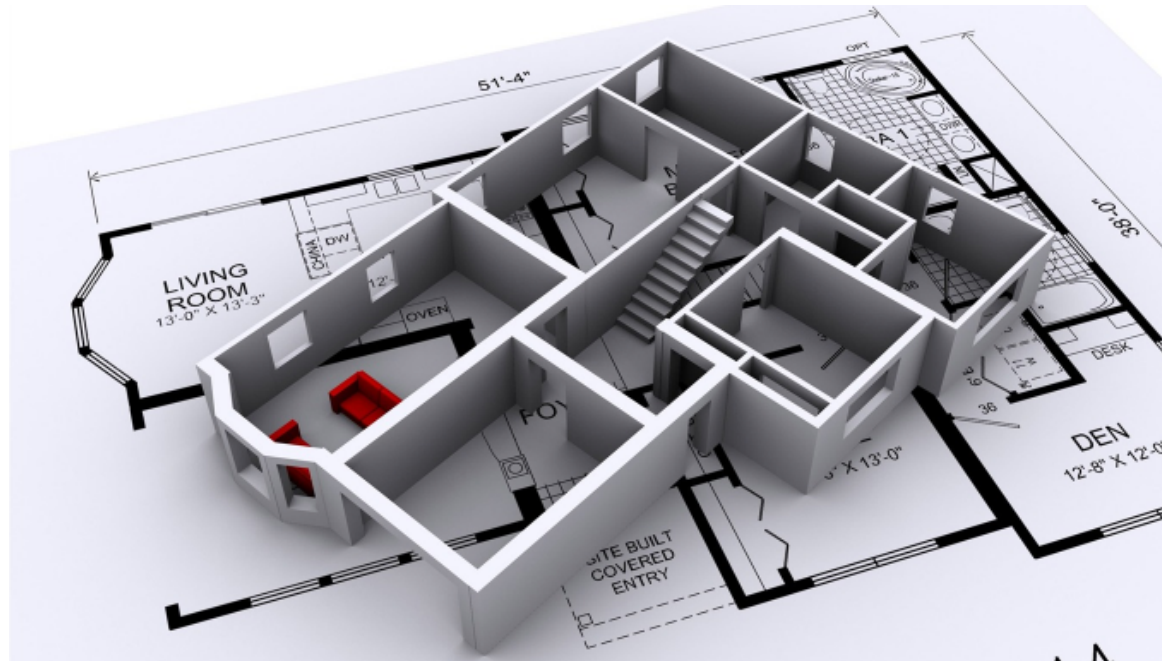
Neste módulo são abordados os seguintes conteúdos:

- Diagramas E-R (Entidade-Relação)
- Modelo Relacional
- Normalização e desnormalização

# | DIAGRAMAS E-R

# Modelação

- A modelação consiste em criar modelos de análise do sistema estudado
- Por exemplo: na arquitetura, antes de construir, criam-se maquetes



- Na informática, antes de criar uma base de dados, criam-se modelos

# Diagramas E-R

- Os diagramas E-R (Entidade-Relação) são uma forma de modelar os dados ou estruturas de um sistema
- Servem como base para uma futura base de dados
- Em 1976 Peter Chen criou uma forma de modelação E-R, conhecida como notação Chen (*Chen notation*)
- Existem outras formas de criar modelos E-R (*crow's foot*, etc.)

# Diagramas E-R

- Os diagramas E-R são compostos por:
  - **Entidades**
    - são "coisas" existentes no mundo real:
      - pessoas
      - organizações (empresas, escolas, departamentos, etc.)
      - coisas (produtos, documentos, livros, etc.)
      - etc.
  - **Atributos**
    - dados armazenados nas entidades:
      - nome, idade, nº de cartão de cidadão, morada, dimensão, etc.
  - **Relacionamentos**
    - associações entre as entidades

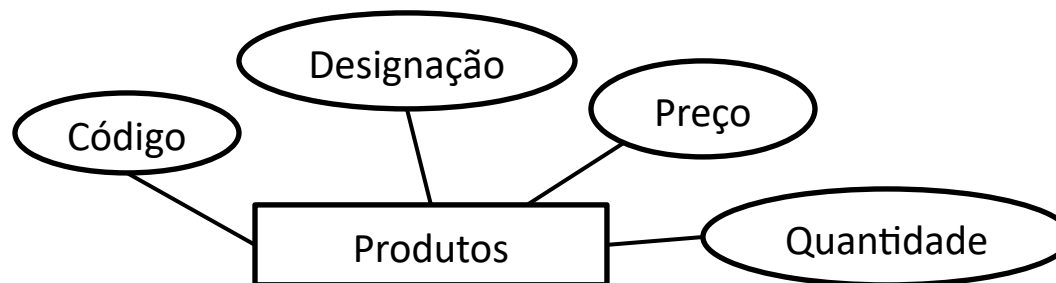
# Diagramas E-R

## Entidades (exemplo):



(as entidades podem ser escritas no singular ou no plural)

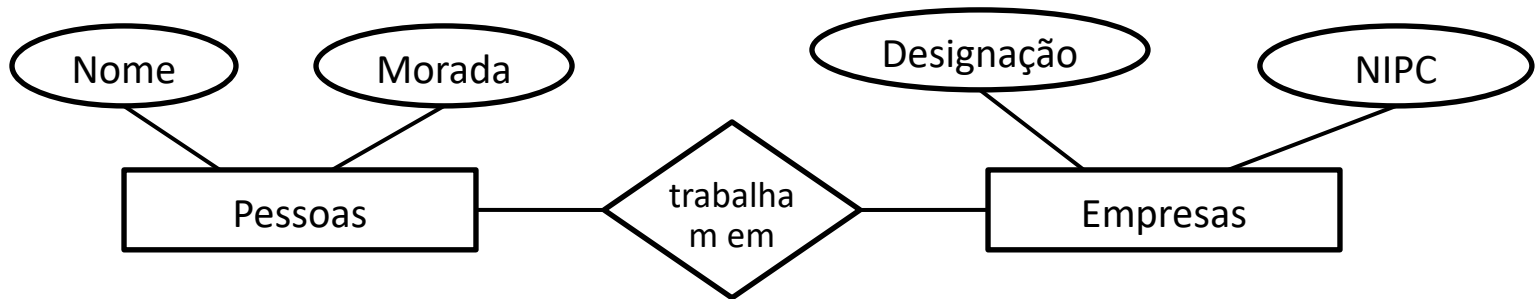
## Atributos (exemplo):



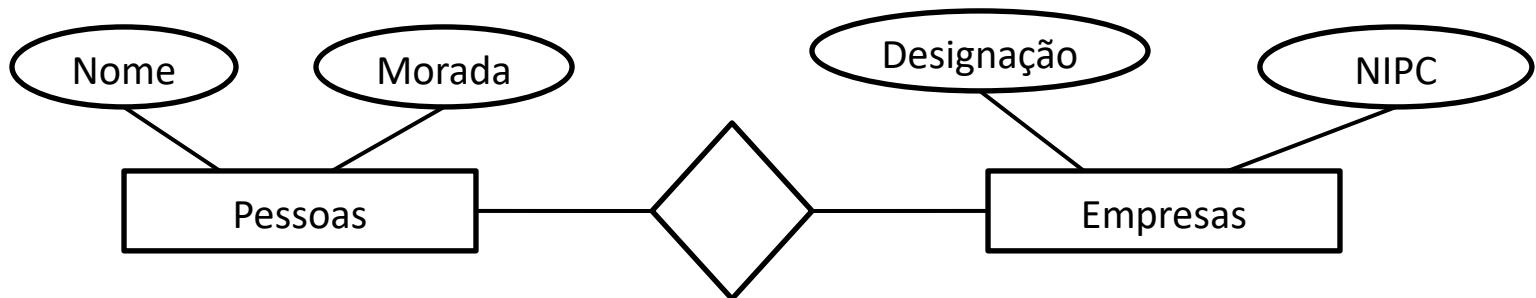
# Diagramas E-R

## Relacionamentos

Exemplo 1:



Exemplo 2 (sem nome no relacionamento):

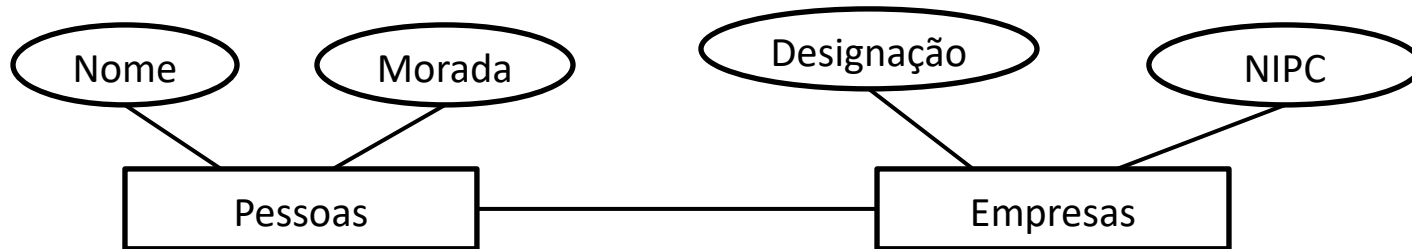




# Diagramas E-R

## Relacionamentos

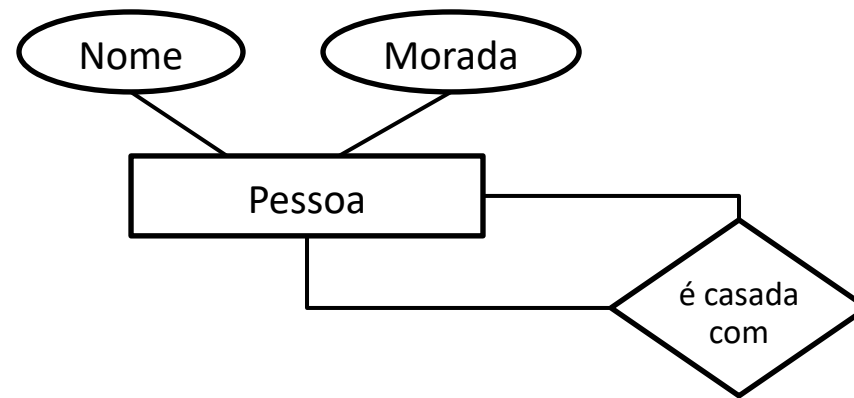
Exemplo 3 (sem losango):



# Diagramas E-R: Tipos de relacionamentos

- Os relacionamentos podem ser:

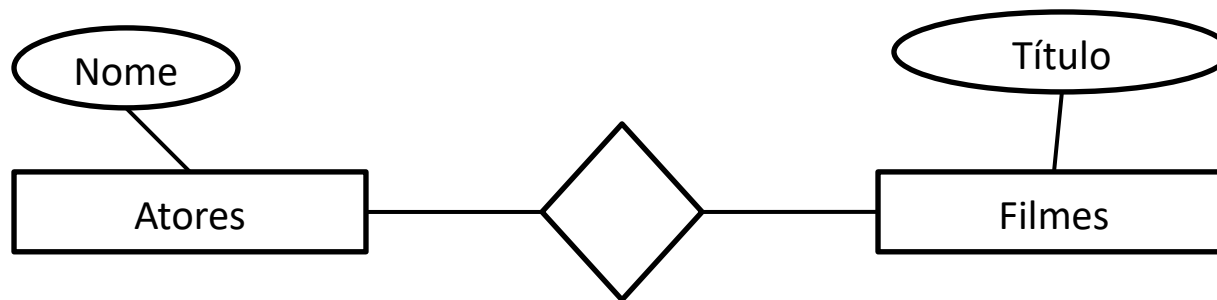
**Unários:** uma entidade mantém um relacionamento consigo própria



# Diagramas E-R: Tipos de relacionamentos

- Os relacionamentos podem ser:

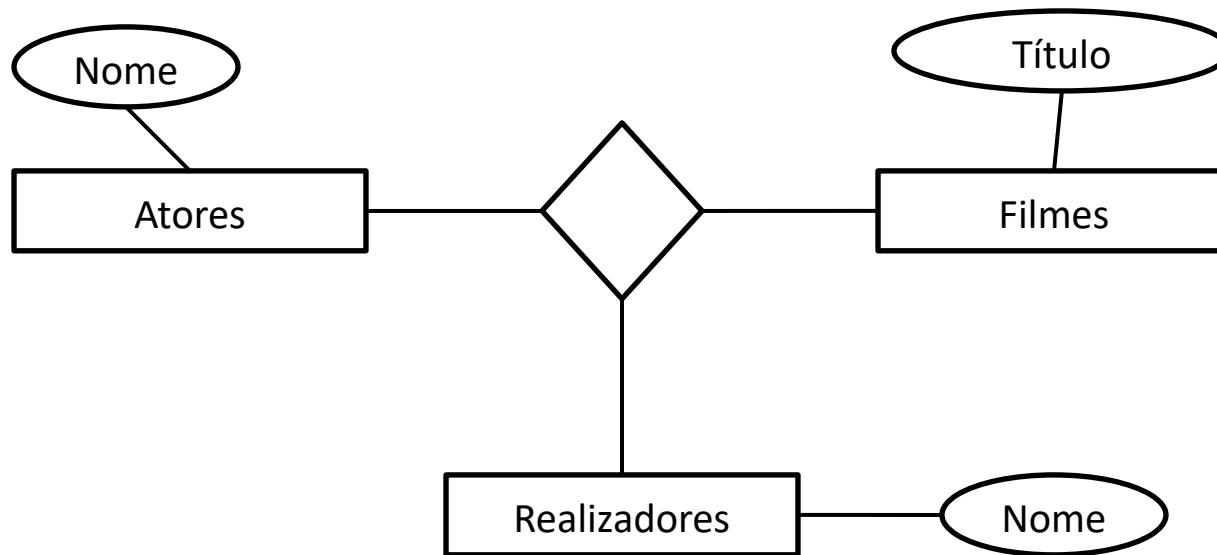
**Binários:** um relacionamento entre duas entidades (é o mais comum)



# Diagramas E-R: Tipos de relacionamentos

- Os relacionamentos podem ser:

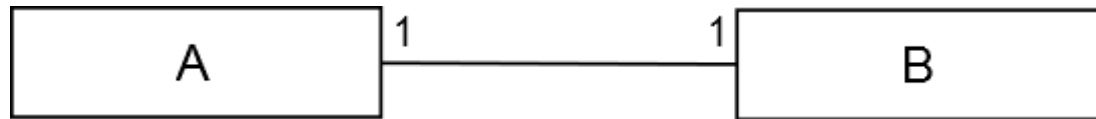
**Ternários:** um relacionamento entre três entidades



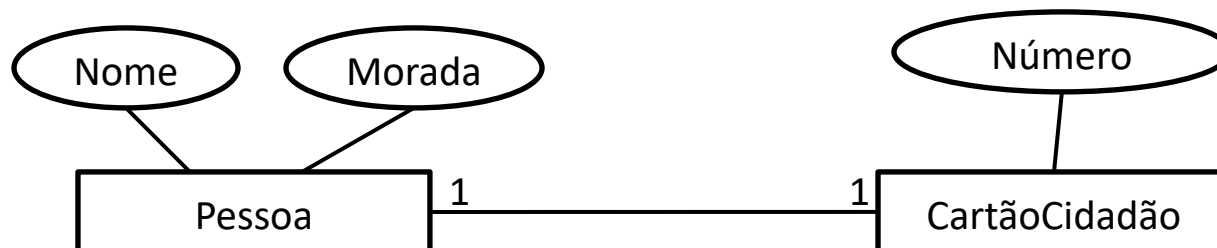
# Diagramas E-R: Cardinalidade

## Relacionamentos 1:1

- Num relacionamento 1:1, cada ocorrência da entidade A está relacionada com apenas uma ocorrência da entidade B, e vice-versa



- Exemplo: uma pessoa tem um cartão de cidadão, um cartão de cidadão apenas pertence a uma pessoa



# Diagramas E-R: Cardinalidade

## Relacionamentos 1:N

- Neste tipo de relacionamento cada ocorrência da entidade A está relacionada com várias ocorrências da entidade B, ou vice-versa



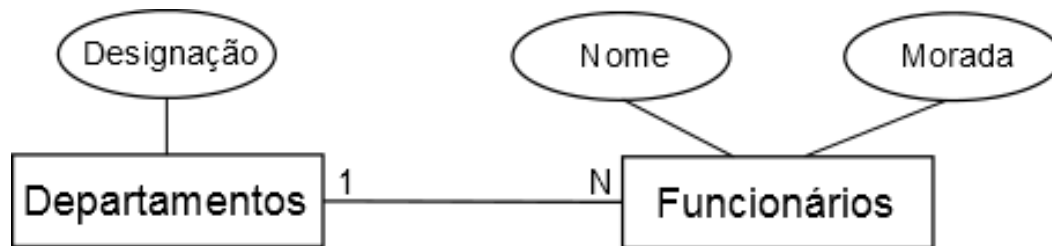
ou



# Diagramas E-R: Cardinalidade

## Relacionamentos 1:N

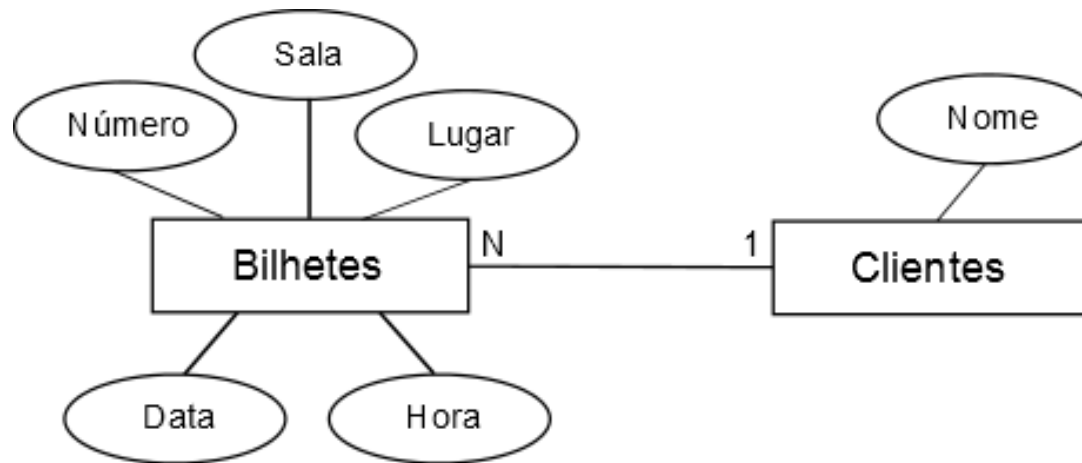
- Exemplo 1:
  - numa empresa, cada departamento tem vários funcionários
  - cada funcionário pertence apenas a um departamento



# Diagramas E-R: Cardinalidade

## Relacionamentos 1:N

- Exemplo 2:
  - num cinema, um bilhete apenas pode ser comprado por um cliente
  - cada cliente pode comprar vários bilhetes





# Diagramas E-R: Cardinalidade

## Relacionamentos M:N

- Ocorre quando cada ocorrência da entidade A está relacionada com várias ocorrências da entidade B, e vice-versa

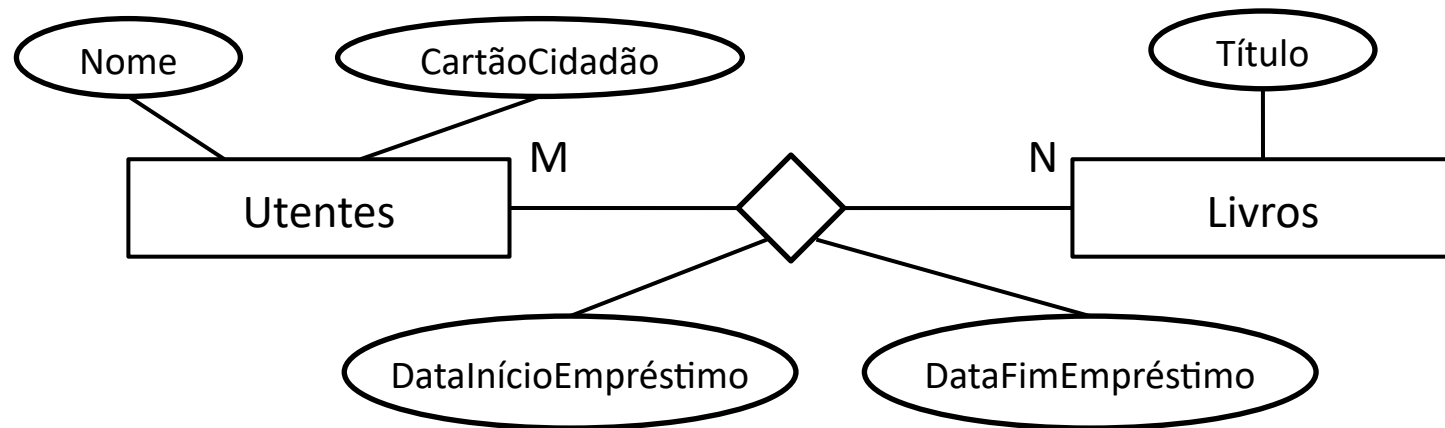


- Exemplo:
  - um filme é protagonizado por vários atores
  - um ator pode ser protagonista em vários filmes



# Diagramas E-R: Atributos nos relacionamentos

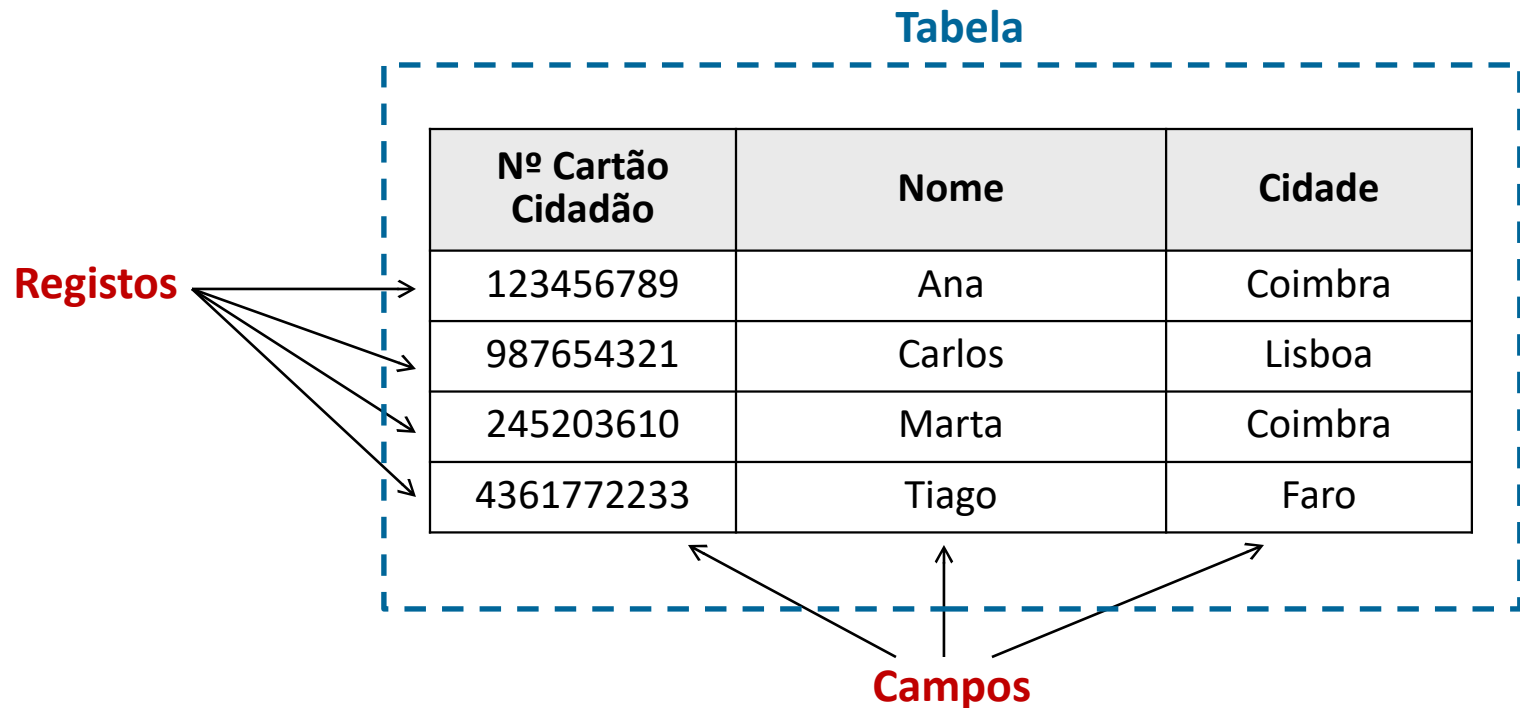
- Por vezes é necessário incluir atributos nos relacionamentos
- Ocorre quando existem dados que dizem respeito não a uma das entidades mas ao relacionamento que ocorre entre estas
- Por exemplo:
  - numa biblioteca, existe um relacionamento entre a entidade Utentes e a entidade Livros: o empréstimo de livros
  - nesse relacionamento são armazenadas duas informações: a data de início e a data de fim do empréstimo



# MODELO RELACIONAL

# Bases de dados relacionais: Conceitos iniciais

- Uma tabela contém dados, organizados em:
  - registos (as linhas)
  - campos (as colunas)



## Bases de dados relacionais: Algumas regras

- Não podem existir campos, na mesma tabela, com o mesmo nome
- Não podem existir registos duplicados, isto é, com os mesmos valores em todos os campos (cada registo representa uma ocorrência única)
- Exemplo de uma tabela com erros:

Número	Nome	Disciplina	Disciplina
1	Ana Carolina	Português	Matemática
2	Carlos Marques	Português	Inglês
3	Marta Santos	Informática	
4	Tiago Alves	Português	Matemática
4	Tiago Alves	Português	Matemática

## Bases de dados relacionais: Algumas regras

- Não podem existir campos, na mesma tabela, com o mesmo nome
- Não podem existir registos duplicados, isto é, com os mesmos valores em todos os campos (cada registo representa uma ocorrência única)
- Exemplo de uma tabela com erros:

Nomes de campos iguais

Número	Nome	Disciplina	Disciplina
1	Ana Carolina	Português	Matemática
2	Carlos Marques	Português	Inglês
3	Marta Santos	Informática	
4	Tiago Alves	Português	Matemática
4	Tiago Alves	Português	Matemática

Registos duplicados

# Chaves

- Chave primária:
  - atributo ou conjunto de atributos que permitem identificar de forma unívoca cada registo numa tabela
- Ou seja, a chave primária é imprescindível em qualquer tabela

**Tabela Produtos**

<u>Produto_ID</u>	Designação	Quantidade
123	Martelo	10
987	Porta	2
245	Martelo	8
436	Dobradiça	33

Chave primária

# Chaves

- Chave simples:
  - contém apenas um campo

**Tabela Produtos**

<u>Produto ID</u>	Designação	Quantidade
123	Martelo	10
987	Porta	2
245	Martelo	8
436	Dobradiça	33

Chave primária simples



# Chaves

- Chave simples:
  - contém apenas um campo
- Chave composta:
  - contém dois ou mais campos

**Tabela Funcionários**

<u>Nome</u>	<u>Departamento</u>	Sala
Ana	Contabilidade	10
Alberto	Contabilidade	10
Paulo	Gestão	15
Mónica	Informática	8

Chave primária composta

# Chaves

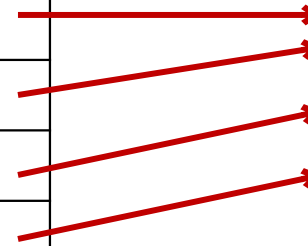
- Chave estrangeira:
  - chave numa tabela que é chave primária noutra tabela

**Tabela Funcionários**

<u>ID</u>	Nome	<u>Departamento_ID</u>
1	Ana Carolina	1
2	Carlos Marques	1
3	Marta Santos	2
4	Tiago Alves	3

**Tabela Departamentos**

<u>ID</u>	Designacao
1	Contabilidade
2	Marketing
3	Informática



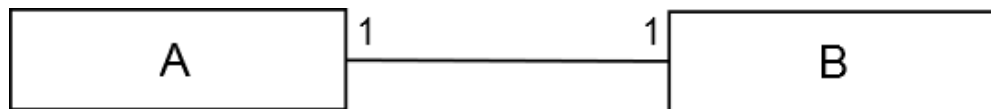
# Relacionamentos entre tabelas

- Os relacionamentos entre tabelas podem ser de 3 tipos:
  - 1:1
  - 1:N
  - M:N

# Relacionamentos entre tabelas

## Relacionamentos 1:1

- Num relacionamento 1:1, cada registo da tabela A está relacionado com apenas um registo da tabela B, e vice-versa



- Neste caso existem duas formas de implementar:
  - criar apenas uma tabela
  - criar duas tabelas
- Normalmente acontece o primeiro caso

# Relacionamentos entre tabelas

## Relacionamentos 1:1

- Exemplo:
  - uma pessoa tem um cartão de cidadão, um cartão de cidadão apenas pertence a uma pessoa



**Tabela Pessoas**

<u>ID</u>	Nome	Morada	NumeroCartaoCidadao
1	Ana Carolina	Rua A	12345678
2	Carlos Marques	Rua B	65429361
3	Marta Santos	Rua C	98765432

# Relacionamentos entre tabelas

## Relacionamentos 1:N

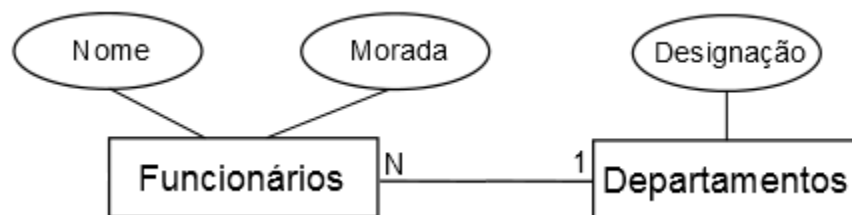
- Num relacionamento de 1:N cada registo da tabela A está relacionado com vários registos da tabela B, ou vice-versa
- Neste caso, cada registo da tabela do lado N tem uma chave estrangeira que o relaciona com o registo correspondente da tabela do lado 1



# Relacionamentos entre tabelas

## Relacionamentos 1:N

- Exemplo 1:
  - numa empresa cada departamento tem vários funcionários
  - cada funcionário pertence apenas a um departamento



**Tabela Funcionários**

<u>ID</u>	Nome	Morada	<u>Departamento_ID</u>
1	Ana Carolina	Rua A	1
2	Carlos Marques	Rua B	4
3	Marta Santos	Rua C	3

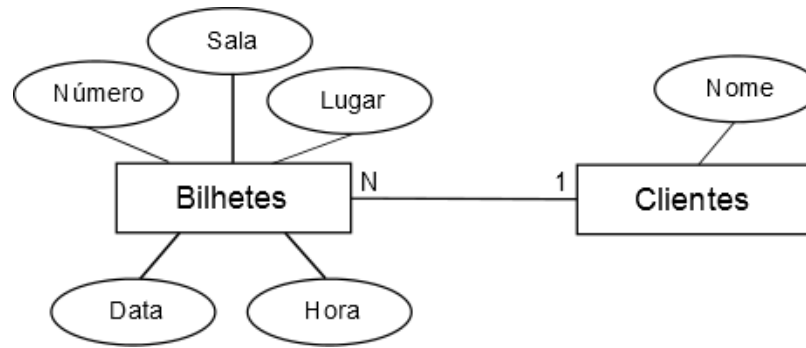
**Tabela Departamentos**

<u>ID</u>	Designacao
1	Informática
2	Contabilidade
3	Marketing
4	Comercial

# Relacionamentos entre tabelas

## Relacionamentos 1:N

- Exemplo 2:
  - num cinema, um bilhete apenas pode ser comprado por um cliente
  - cada cliente pode comprar vários bilhetes



**Tabela Bilhetes**

<u>Número</u>	Sala	Lugar	Data	Hora	<u>Cliente_ID</u>
001	4	18	10/11/2012	21:10	1
002	1	5	24/11/2013	00:05	1
003	7	22	09/12/2013	14:00	2

**Tabela Clientes**

<u>ID</u>	Nome
1	Ana Carolina
2	Carlos Marques
3	Marta Santos
4	Paulo Marques



# Relacionamentos entre tabelas

## Relacionamentos M:N

- Num relacionamento M:N cada registo da tabela A está relacionado com um ou mais registos da tabela B e vice-versa



- Neste tipo de relacionamento terá de ser criada uma tabela adicional, com duas chaves estrangeiras, as quais estão associadas às tabelas A e B
- Esta nova tabela é que permite implementar o relacionamento M:N

# Relacionamentos entre tabelas

## Relacionamentos M:N

- Exemplo:
  - um filme é protagonizado por vários atores
  - um ator pode ser protagonista em vários filmes



# Relacionamentos entre tabelas

**Tabela Filmes**

<u>ID</u>	Titulo	Ano
1	Brazil	1985
2	Taxi Driver	1976
3	Um Peixe Chamado Wanda	1988

**Tabela Atores**

<u>ID</u>	Nome
1	Robert de Niro
2	Michael Palin
3	Jodie Foster
4	John Cleese
5	Jamie Lee Curtis

**Tabela Filmes/Atores**

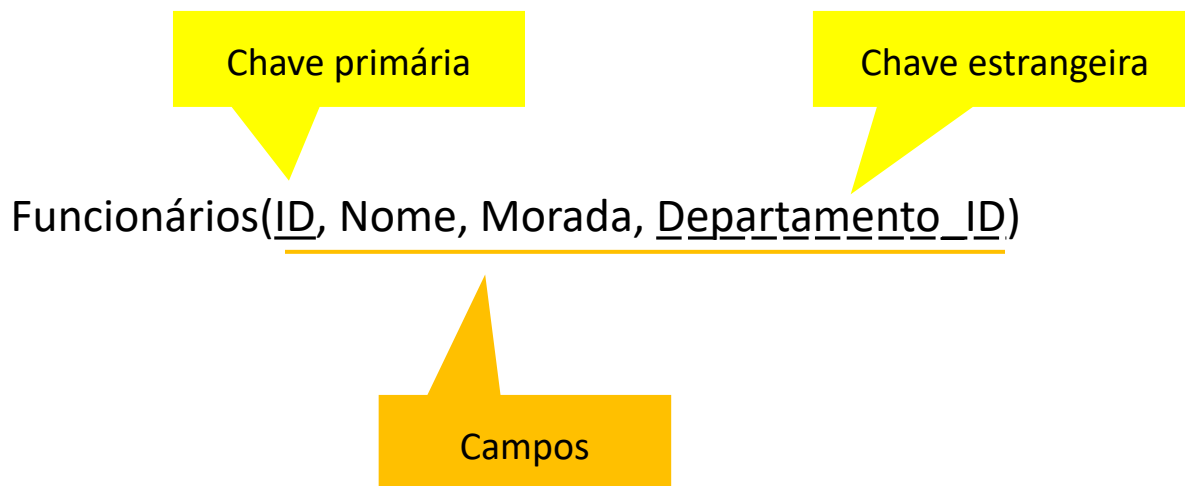
<u>ID</u>	Filme_ID	Ator_ID
1	1	1
2	1	2
3	2	1
4	2	3
5	3	2
6	3	4
7	3	5

# Esquema relacional

- É uma forma compacta de representação das tabelas, campos e relacionamentos de uma base de dados
- Formato:

Tabela(ChavePrimária, Campo1, Campo2, ..., ChaveEstrangeira1, ...)

- Exemplo:



# Integridade e consistência de bases de dados

- Existem duas regras de integridade:
  - integridade de entidade
  - integridade referencial
- As regras de integridade têm como objetivo garantir a integridade e a consistência dos dados de uma base de dados

# Integridade de entidade

- Nenhum componente de uma chave primária pode ser nulo (NULL)
- Exemplo:

**Tabela Alunos**

<u>Número</u>	Nome	Disciplina
1	Ana Carolina	Português
2	Carlos Marques	Português
NULL	Marta Santos	Informática
3	Tiago Alves	Português

# Integridade referencial

- Se existir uma chave estrangeira numa tabela:
  - esta deve corresponder à chave primária de um registo de outra tabela
  - ou, em alternativa, deve ter o valor NULL
- Exemplo:
  - cada funcionário está ligado a um departamento
  - o funcionário nº 4 está ligado a um departamento que não existe

<u>ID</u>	Nome	<u>Departamento_ID</u>
1	Ana Carolina	1
2	Carlos Marques	1
3	Marta Santos	2
4	Tiago Alves	3

<u>ID</u>	Designação
1	Contabilidade
2	Marketing

# Integridade referencial

- Solução:
  - atribuir o valor NULL ao campo Departamento\_ID do funcionário nº 4

**Tabela Funcionários**

<u>ID</u>	Nome	<u>Departamento_ID</u>
1	Ana Carolina	1
2	Carlos Marques	1
3	Marta Santos	2
4	Tiago Alves	NULL

**Tabela Departamentos**

<u>ID</u>	Designação
1	Contabilidade
2	Marketing



# Redundância

- Redundância, em bases de dados, traduz-se em dados repetidos
- Como os dados ocupam espaço, quanto maior a redundância maior será o espaço ocupado por uma base de dados
- Um dos principais objetivos numa base de dados é que esta **ocupe o menor espaço possível**
- Logo, **é necessário reduzir ao máximo a redundância**

# Redundância

- Exemplo:



**Base de dados 1**

**Tabela Alunos**

Nome	Localidade
Ana Carolina	Condeixa
Carlos Marques	Coimbra
Marta Santos	Penacova
Maria Santana	Coimbra
António Cláudio	Coimbra
Tiago Alves	Condeixa



**Base de dados 2**

**Tabela Alunos**

Nome	Localidade_ID
Ana Carolina	1
Carlos Marques	2
Marta Santos	3
Maria Santana	2
António Cláudio	2
Tiago Alves	1

**Tabela Localidades**

ID	Designação
1	Condeixa
2	Coimbra
3	Penacova

# Redundância

- Exemplo:

Aqui as localidades  
repetem-se



Base de dados 1

Tabela Alunos

Nome	Localidade
Ana Carolina	Condeixa
Carlos Marques	Coimbra
Marta Santos	Penacova
Maria Santana	Coimbra
António Cláudio	Coimbra
Tiago Alves	Condeixa



Base de dados 2

Tabela Alunos

Nome	Localidade_ID
Ana Carolina	1
Carlos Marques	2
Marta Santos	3
Maria Santana	2
António Cláudio	2
Tiago Alves	1

Tabela Localidades

ID	Designação
1	Condeixa
2	Coimbra
3	Penacova

Aqui as localidades  
não se repetem

# Redundância

- Exemplo:



**Base de dados 1**

**Tabela Alunos**

Nome	Localidade
Ana Carolina	Condeixa
Carlos Marques	Coimbra
Marta Santos	Penacova
Maria Santana	Coimbra
António Cláudio	Coimbra
Tiago Alves	Condeixa

A base de dados 1 irá ocupar mais espaço



**Base de dados 2**

**Tabela Alunos**

Nome	Localidade_ID
Ana Carolina	1
Carlos Marques	2
Marta Santos	3
Maria Santana	2
António Cláudio	2
Tiago Alves	1

**Tabela Localidades**

ID	Designação
1	Condeixa
2	Coimbra
3	Penacova

# | NORMALIZAÇÃO

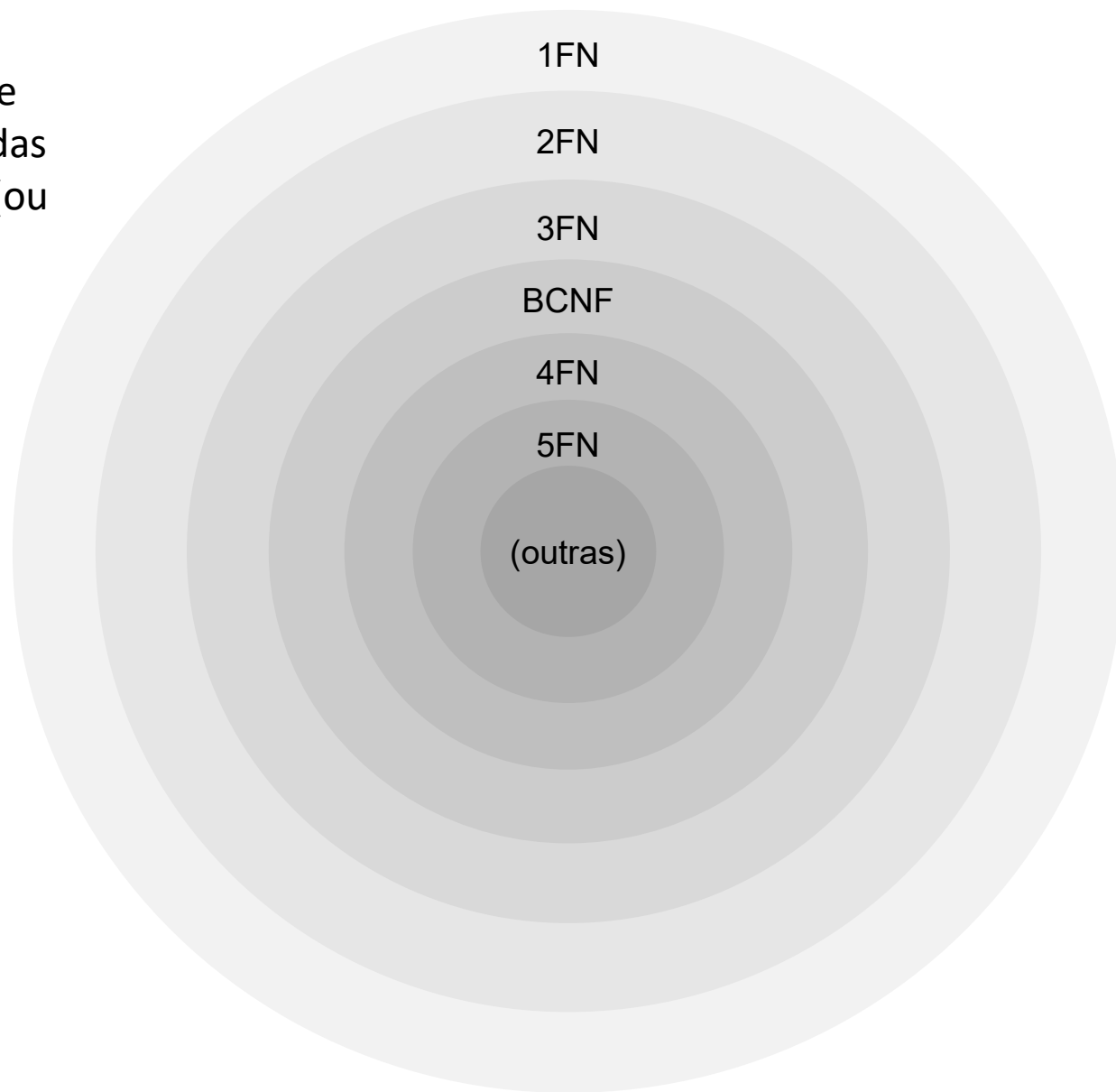
# O que é a normalização?

- Normalizar um produto consiste em uniformizar algumas ou todas as características do mesmo
- Exemplo:
  - a fruta que é comercializada é normalmente objeto de um processo de normalização
  - isto é, apenas são apresentadas ao consumidor frutos que obedecem a um determinado conjunto de regras (ao nível do formato, dimensão, aspeto, etc.)
  - apresenta vantagens ao nível da qualidade do produto bem como de redução de custos
  - por outro lado, tem a desvantagem de implicar a destruição de frutos que estão em perfeito estado de consumo mas não obedecem a algumas ou todas as regras de normalização

# O que é a normalização?

- Nas bases de dados a normalização consiste em:
  - evitar a acumulação de "anomalias" nas tabelas
  - eliminar tanto quanto possível, redundância de informação
- Uma base de dados normalizada é, em princípio, uma base de dados com uma estrutura correta e eficaz
- É, no entanto, importante salientar que o processo de normalização deve ser cuidadosamente aplicado, especialmente se se pretendem efetuar normalizações além da 3FN (BCNF, 4FN e 5FN)
- **Quanto mais normalizada é uma base de dados, mais fragmentada fica**
- A fragmentação de uma base de dados pode conduzir a:
  - degradação da *performance* (especialmente em consultas)
  - complexificar desnecessariamente a estrutura

Existem várias regras de normalização, conhecidas como *formas normais* (ou FN)





# 1ª Forma Normal (1FN)

- Todos os campos devem ser atómicos:
  - ou seja, devem conter apenas um valor
- Não podem existir campos repetitivos:
  - ou seja, campos repetitivos são campos muito semelhantes
- Exemplo:

ID	Nome	Morada	Disciplinas
1	João	Rua A	Port.; Fil.; Mat.
2	José	Rua B	Mat.
3	Ana	Rua C	Ing.; Mat.
4	Carla	Rua D	Port.; Ing.; Mat.

- **A tabela não respeita a 1FN:**
  - o campo Disciplinas contém valores não atómicos

# 1ª Forma Normal (1FN)

## Solução

- Separar a informação dos alunos e das disciplinas:
  - criar a tabela Disciplinas, com dados de disciplinas
- Criar a tabela Alunos\_Disciplinas que liga os alunos às disciplinas:
  - um aluno pode frequentar várias disciplinas
  - uma disciplina pode ser frequentada por vários alunos

# 1ª Forma Normal (1FN)

**Alunos**

ID	Nome	Morada
1	João	Rua A
2	José	Rua B
3	Ana	Rua C
4	Carla	Rua D

**Alunos\_Disciplinas**

ID	Aluno_ID	Disciplina_ID
1	1	1
2	1	2
3	1	3
4	2	3
5	3	4
6	3	3
7	4	1
8	4	4
9	4	3

**Disciplinas**

ID	Designacao
1	Port.
2	Fil
3	Mat.
4	Ing.

# 1ª Forma Normal (1FN)

## Solução

- Alunos(ID, Nome, Morada)
- Alunos\_Disciplinas(ID, Aluno\_ID, Disciplina\_ID)
- Disciplinas(ID, Designacao)

## 2ª Forma Normal (2FN)

- A 2ª Forma Normal é baseada no conceito de dependência funcional:

“

(...) por definição, diz-se que existe uma dependência funcional  $X \rightarrow Y$  entre dois conjuntos de atributos  $X$  e  $Y$ , se uma instância de valores dos atributos de  $X$  determina ou identifica univocamente uma instância de valores dos atributos de  $Y$ . Ou seja, não existem duas instâncias distintas de  $Y$ , para uma mesma instância de  $X$ .

Se  $X \rightarrow Y$  então existe uma dependência funcional entre  $X$  e  $Y$ , em que  $X$  determina  $Y$  ou, o que é o mesmo,  $Y$  depende de  $X$ .

”

**Pereira, J.L. (1998). Tecnologia de Bases de Dados, FCA, Lisboa**

## 2ª Forma Normal (2FN)

- Uma tabela está na 2ª Forma Normal:
  - se estiver na 1ª Forma Normal
  - cada campo não-chave é funcionalmente dependente da chave primária
- Exemplo:
  - a tabela Encomendas guarda informações sobre encomendas de produtos a fornecedores

Encomendas(Produto\_ID, DesignaçãoProduto, Fornecedor\_ID,  
NomeFornecedor, MoradaFornecedor, CódigoFornecedor,  
Quantidade)
- **A tabela não respeita a 2FN porque:**
  - apenas um campo (Quantidade) é dependente da chave primária na sua totalidade
  - os restantes campos são dependentes apenas de uma parte da chave primária

## 2ª Forma Normal (2FN)

- Vamos começar por analisar um exemplo de uma tabela que **está** na 2ª Forma Normal:

Pessoas(ID, CartãoCidadão, Nome, DataNascimento)

- Como é que sabemos que está na 2FN?
- 1FN: a tabela está na 1ª Forma Normal ✓
- 2FN: vamos analisar as dependências funcionais

ID → CartãoCidadão, Nome, DataNascimento

- Os campos CartãoCidadão, Nome e DataNascimento são funcionalmente dependentes do campo ID
- Ou seja, os campos estão dependentes do valor do campo ID
- Dito de outra forma, uma pessoa com um ID terá um cartão de cidadão, nome e data de nascimento diferente de outra pessoa, com outro ID
- Face a isto, podemos dizer que a tabela está na 2ª Forma Normal ✓

## 2ª Forma Normal (2FN)

- Vamos agora analisar um outro exemplo, de uma tabela que **não está** na 2ª Forma Normal
- A tabela Encomendas guarda informações sobre encomendas de produtos a fornecedores

Encomendas(Produto\_ID, DesignaçãoProduto, Fornecedor\_ID,  
NomeFornecedor, MoradaFornecedor, CódigoFornecedor,  
Quantidade)

<u>Produto_ID</u>	DesignaçãoPro duto	<u>Fornecedor_ID</u>	NomeFornecedor	MoradaFornecedor	CódigoFornecedor	Quantidade
1	Armário	9	Matias Lda.	Rua A	801	5
2	Maçã Alcobaça	7	SuperFrutas	Rua B	109	200
3	Cadeira Zenite	2	MegaCroc	Rua C	910AB	15
4	Chapéu Nike	1	MHX	Rua D	C19	40



## 2ª Forma Normal (2FN)

- 1FN: a tabela está na 1ª Forma Normal ✓
- 2FN: vamos analisar as dependências funcionais

Produto\_ID → DesignaçãoProduto

Fornecedor\_ID → NomeFornecedor, MoradaFornecedor, CódigoFornecedor

Produto\_ID, Fornecedor\_ID → Quantidade

- **A tabela não respeita a 2FN porque:**
  - apenas um campo (Quantidade) é dependente da chave primária na sua totalidade
  - os restantes campos são dependentes apenas de uma parte da chave primária

## 2ª Forma Normal (2FN)

### Solução

- Decompor a tabela em duas tabelas adicionais:
  - Produtos
  - Fornecedores
- São colocadas nessas tabelas os campos que dependem de cada parte da chave primária

Produtos(Produto\_ID, DesignaçãoProduto)

Fornecedores(Fornecedor\_ID, NomeFornecedor, MoradaFornecedor,  
CódigoFornecedor)

Encomendas(ID, Produto\_ID, Fornecedor\_ID, Quantidade)

## 3ª Forma Normal (3FN)

- Uma tabela está na 3ª Forma Normal:
  - se estiver na 2ª Forma Normal
  - se não existirem dependências funcionais entre os atributos não-chave
  - ou seja, todos os atributos não-chave de uma tabela devem ser completamente independentes entre si, e completamente dependentes da chave primária
- Analisemos a tabela Fornecedores do exemplo anterior:
- Nesta tabela, cada fornecedor tem um código (armazenado no campo CódigoFornecedor), que é determinado consoante o continente onde está sediado
- Dependências funcionais:

Fornecedor\_ID → NomeFornecedor, MoradaFornecedor, CódigoFornecedor  
MoradaFornecedor → CódigoFornecedor

## 3ª Forma Normal (3FN)

- A tabela Fornecedores não está na 3FN porque:
  - o campo **CódigoFornecedor** depende do campo não-chave **MoradaFornecedor**
- Solução:
  - colocar o campo **CódigoFornecedor** numa nova tabela
- Assim:
  - todos os campos da tabela Fornecedores dependem única e exclusivamente da chave primária
  - também na nova tabela (Códigos) o campo não-chave depende única e exclusivamente da chave primária

Fornecedores(Fornecedor\_ID, NomeFornecedor, MoradaFornecedor)  
Códigos(MoradaFornecedor, CódigoFornecedor)

# | DESNORMALIZAÇÃO

# Desnormalização

- A desnormalização consiste em, numa base de dados normalizada, retirar a normalização numa ou várias tabelas
- Tem como objetivo aumentar a performance da base de dados
- Isto porque, em algumas situações, tabelas normalizadas prejudicam a performance
- As estratégias de desnormalização dependem muito dos casos específicos em que são aplicadas
- A desnormalização pode ser efetuada:
  - de forma muito subtil: por exemplo, acrescentando apenas um campo
  - ou de modo mais pronunciado: por exemplo, criar uma base de dados separada, apenas com dados de histórico, completamente desnormalizados
  - ou de uma forma intermédia: criando por exemplo uma tabela que junta dados de diferentes tabelas

# Desnormalização

- Imaginemos o seguinte exemplo: uma concessionária de auto-estrada armazena os dados das viagens dos veículos

Automoveis(ID, Identificador\_ID, Matricula, Proprietario\_ID)

Proprietarios(ID, Nome, NIF, Telefone, Email)

Identificadores(ID, DataAtivacao)

Proprietarios\_Identificadores(ID, Proprietario\_ID, Identificador\_ID)

Passagem(ID, DataHora, Portico\_ID)

Porticos(ID, Designacao)

- Pode ser criada uma única tabela, para efeitos de resumo que contém as passagens de um utilizador:

Passagens(ID, DataHora, DesignacaoPortico, Identificador, Matricula, NomeProprietario, NIFProprietario)