

Conceitos Básicos

1. Conceito de Dados e Informação

Necessidade de Informação:

- Ao ser humano
- Especialização
- Conhecimento
- Novas ideias

À organização (Empresa)

- Vital para o seu funcionamento

Características da informação:

- Precisa (correcta e verdadeira)
- Concisa (de fácil manipulação)
- Simples (de fácil compreensão)
- Oportuna (existe no momento e local correcto)

Dados versus Informação

Dados: Factos do mundo real (podem não ser informação).

Informação: Dados depois de processados, de forma a representarem algo para o ser humano.

2. A Evolução dos Ficheiros ao Longo dos Tempos

Ficheiros manuais

Exemplo:

- Arquivo de uma biblioteca
- Fichas de clientes

Têm como principais inconvenientes:

- Espaço físico ocupado
- Duplicação de informação desnecessária (Redundância de Informação)
- Difícil manipulação da informação armazenada
- Deterioração do suporte utilizado (Papel ou cartão)

Ficheiros Clássicos

Exemplo:

- Magnético
- Óptico

Têm como principais vantagens:

- Pouco espaço físico ocupado
- Eliminação da Redundância de Informação.
- Fácil manipulação da informação armazenada
- Menor deterioração do suporte utilizado

Neste tipo de ficheiros, a informação é organizada por registos, constituídos por campos.

Registo: Conjunto de dados

Ficheiro: Registos correlacionados entre si

Exemplo:

Registo: Ficha de um sócio de um Vídeo Clube.

Ficheiro: Conjunto de todos os sócios.

3. Sistemas de Gestão de Base de Dados

Para manipular a informação foram criadas ferramentas do tipo SGBD – Sistema de Gestão de Base de Dados.

Um sistema SGBD permitem criar e manipular Bases de Dados em que os dados são estruturados com independência relativamente aos programas de aplicação que os manipulam.

A Arquitectura de um SGBD pode ser dividido em 3 Níveis:

1. Nível Físico

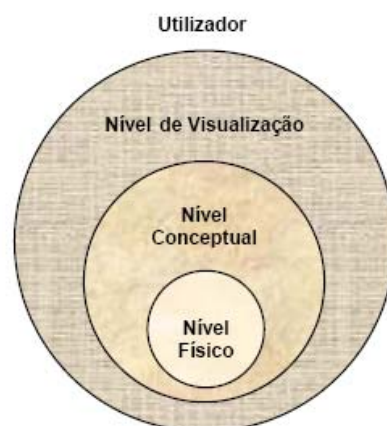
- Suportes informáticos para o armazenamento dos ficheiros de dados (discos, disquetes, etc.)
- Forma como eles se encontram organizados nesses suportes.

2. Nível Conceptual

- Estruturação ou organização da informação em tabelas (entidades) e formas de relacionamento entre entidades. (Número e tipo de campos em que a informação é estruturada)

3. Nível de Visualização

- Forma como os dados são apresentados aos utilizadores finais, através de interfaces gráficos proporcionados pelo SGBD.



Modelo de Base de Dados

1. O Modelo Lógico

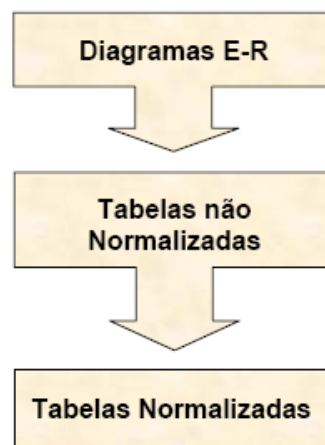
Para descrevermos a nossa aplicação em termos lógicos (Desenho Lógico), vamos utilizar o Modelo Relacional, também chamado Modelo E-R (Entidade-Relação).

Etapas do Método

Etapa 1 – Diagramas E-R: Análise das necessidades de informação da Empresa e elaboração do Modelo E-R.

Etapa 2 – Tabelas não Normalizadas: Transformação do Diagrama E-R num conjunto de Tabelas

Etapa 3 – Tabelas Normalizadas: Normalização das Tabelas



2. Análise das Necessidades de Informação da Empresa e Elaboração do Modelo E-R.

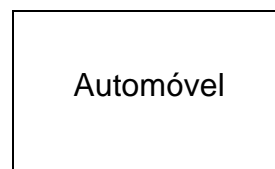
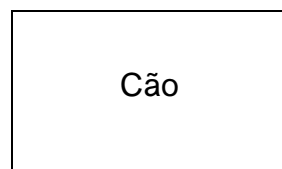
Alguns Conceitos e Definições

Entidade: Objectos ou conceitos que possuem um conjunto de características comuns. Compostas ou caracterizadas por um conjunto de atributos. ***Corresponde a uma tabela de uma Base de Dados.***

Exemplos:

Pastor Alemão	Fiat
Boxer	Renault
Setter	Ford
Doberman	BMW
<hr/>	<hr/>
Entidade Cão	Entidade Automóvel

Representa-se por:



Atributo: Características comuns aos objectos ou conceitos definidos pela Entidade. É qualquer propriedade de uma Entidade. ***Corresponde aos campos de uma tabela.***

Representa-se por:

Cão (Nome, Raça, Sexo, DataNascimento)

Ocorrência: Para determinada situação, vamos ter valores concretos para os atributos. **Corresponde aos registos de uma tabela.**

Exemplo:

Cão

Nome	Raça	Sexo	DataNascimento
Yankee	Boxer	M	Jan85
Rocky	Pastor	F	Mai88
Fritz	Setter	M	Fev83

- Uma entidade ou classe de entidades corresponde a uma tabela
- Os atributos da entidade correspondem aos campos da tabela
- Os vários elementos da entidade correspondem aos registos.

Chave Candidata: É um atributo ou conjunto de atributos que poderá ser usado como Chave Primária para uma dada entidade.

Exemplo: Nome, Nome e Raça

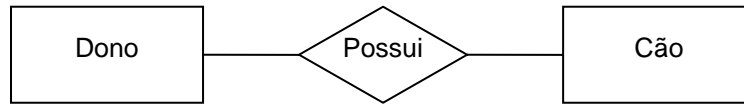
Chave Primária: É um atributo ou conjunto de atributos de uma entidade, escolhido de entre as Chaves Candidatas, que identifica uma ocorrência específica dessa mesma entidade, distinguindo-a das restantes, de forma inequívoca. Não pode ser nula, nem repetida.

Exemplo: Nome

Representa-se da seguinte forma:

Cão (**Nome**, Raça, Sexo, DataNascimento)

Relação: Representa a interligação entre 2 ou mais entidades.



A Chave primária identifica sem ambiguidades cada entidade concreta (Registo de uma tabela). É Através delas que são estabelecidos os relacionamentos entre diferentes entidades ou tabelas

Chave Simples: constituída por um único atributo (Campo).

Chave Composta: constituída por mais do que um atributo (Conjunto de campos).

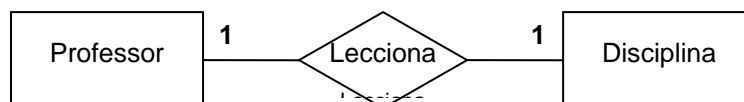
A Chave é Unívoca: Identifica de forma única um registo de uma tabela.

Tipos de Relacionamentos

Existem 3 tipos de relacionamentos binários:

- Relação 1:1
- Relação 1:N
- Relação N:N

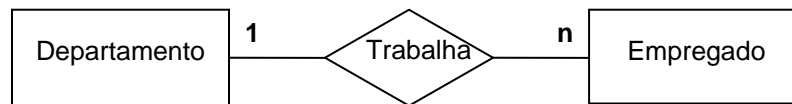
Relação 1:1



Lê-se:

- 1 Professor lecciona uma e uma só Disciplina
- 1 Disciplina é leccionada por um e um só Professor

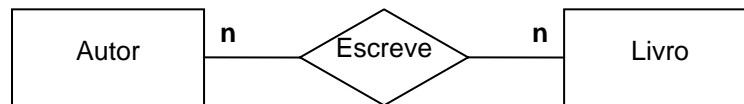
Relação 1:N



Lê-se:

- Num Departamento trabalham vários Empregados
- 1 Empregado trabalha num e num só Departamento

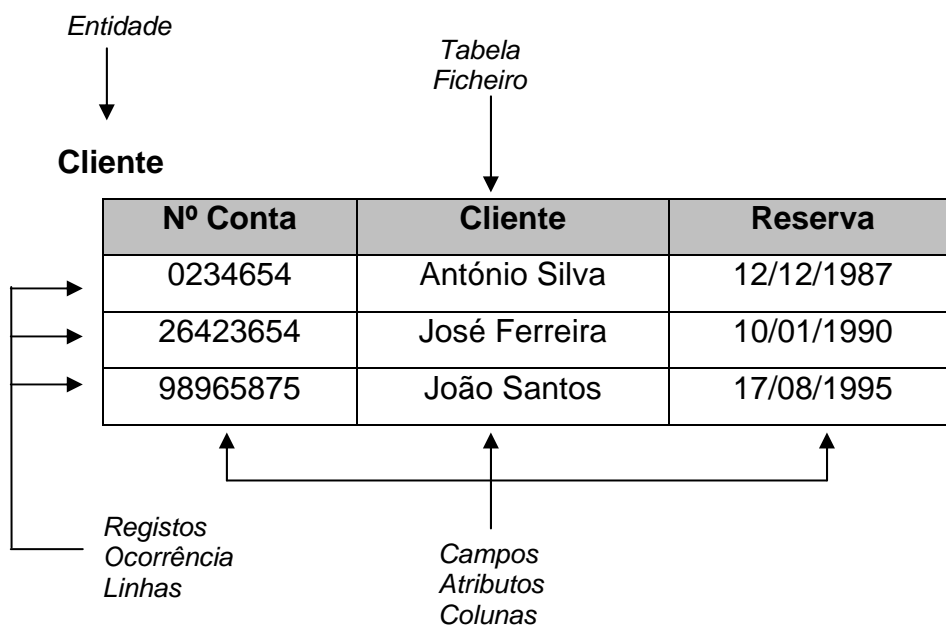
Relação N:N



Lê-se:

- 1 Autor escreve vários Livros
- 1 Livro é escrito por vários Autores

Descrição Informática de uma Entidade



- Numa tabela as colunas correspondem aos campos ou atributos de uma entidade
- As linhas correspondem aos registos
- As várias linhas (registos) podem conter dados repetidos em alguns campos, mas não podem existir duas linhas iguais.
- A ordem pela qual se dispõem as colunas (campos) não é importante e pode ser alterada sem que isso modifique o significado da informação contida na tabela
- A ordem pela qual se dispõem as linhas (registos) também não é importante e pode ser alterada sem que isso signifique alteração da informação
- Não podem haver duas colunas (campos) com o mesmo nome.
- Não deve haver campos vazios
- Não é permitido incluir mais do que um valor em cada campo de cada registo
- Não podem haver registos duplicados.

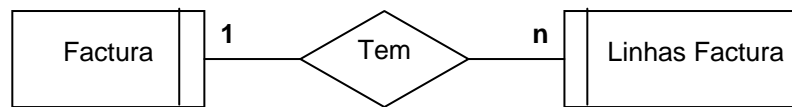
Conectividade de uma Relação

Uma entidade pode participar numa relação de 2 formas:

- **Participação Obrigatória:** Não pode existir nenhuma ocorrência dessa entidade que não esteja associada a alguma ocorrência da outra entidade que participa na relação.
- **Participação não Obrigatória:** Podem existir ocorrências dessa entidade que não estejam relacionadas a alguma ocorrência da outra entidade que participa na relação.

Representação:

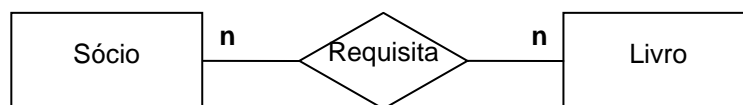
Exemplo1



Lê-se:

- 1 Factura tem várias Linhas de Factura
- 1 Linha de Factura pertence a uma e uma só Factura
- 1 Factura tem Obrigatoriamente Linhas de Factura
- 1 Linha de Factura pertence Obrigatoriamente a uma e uma só Factura

Exemplo2



Lê-se:

- 1 Sócio requisita vários Livros
- 1 Livro é requisitado por vários Sócios
- Nem todos os Sócios requisitam Livros
- Nem todos os Livros são requisitados por Sócios

3. Transformação do Modelo E-R num Conjunto de Tabelas Não Normalizadas

Cada entidade irá dar origem a uma tabela, com:

- Chave Primária
- Atributos
- Chaves Primárias de outras Entidades com que está relacionada (Chave Externa)

Em função do grau da relação e da participação das diversas entidades, iremos obter um conjunto de tabelas não normalizadas.

Relação 1:1

1. Ambas as entidades têm Participação Obrigatória



Empregado (**CodEmpregado**, Nome, Morada)

Carro (**Matricula**, Marca, Modelo)

Basta uma única tabela, com o somatório dos campos das 2 entidades. A chave primária é de uma das entidades.

Empregado (**CodEmpregado**, Nome, Morada, Matricula, Marca, Modelo)

2. Apenas uma entidade tem Participação Obrigatória



São necessárias 2 tabelas, uma para cada entidade, e coloca-se a chave primária da entidade com participação não obrigatória na outra tabela, passando a designar-se por chave externa.

Empregado (**CodEmpregado**, Nome, Morada)

Carro (**Matricula**, Marca, Modelo, CodEmpregado)

Nota: A chave externa é CodEmpregado

3. Nenhuma das entidades tem Participação Obrigatória



São necessárias 3 tabelas, uma para cada entidade e outra para a relação, e coloca-se a chave primária de cada uma das entidades na terceira tabela. A chave primária poderá ser qualquer uma das chaves externas.

Empregado (**CodEmpregado**, Nome, Morada)

Carro (**Matricula**, Marca, Modelo)

Emp_Carro (**CodEmpregado**, Matricula)

Nota: As chaves externas são CodEmpregado e Matrícula

Relação 1:N

1. A entidade do lado N tem Participação Obrigatória



São necessárias 2 tabelas, uma para cada entidade, e coloca-se a chave primária da entidade com grau 1 na tabela com grau N, passando a designar-se por chave externa.

Empregado (**CodEmpregado**, Nome, Morada)

Carro (**Matricula**, Marca, Modelo, CodEmpregado)

Nota: A chave externa é CodEmpregado

2. A entidade do lado N não tem Participação Obrigatória



São necessárias 3 tabelas, uma para cada entidade e outra para a relação, e coloca-se a chave primária de cada uma das entidades na terceira tabela. A chave primária será a chave primária da tabela do lado N.

Empregado (**CodEmpregado**, Nome, Morada)

Carro (**Matricula**, Marca, Modelo)

Emp_Carro (CodEmpregado, **Matricula**)

Nota: As chaves externas são CodEmpregado e Matricula

Relação N:N



São sempre necessárias 3 tabelas, uma para cada entidade e outra para a relação, e coloca-se a chave primária de cada uma das entidades na terceira tabela. A chave primária será a concatenação das chaves primárias de cada uma das tabelas, obtendo-se, desta forma, uma Chave Composta.

Empregado (**CodEmpregado**, Nome, Morada)

Carro (**Matricula**, Marca, Modelo)

Emp_Carro (**CodEmpregado**, **Matricula**)

Notas: Chaves Externas: CodEmpregado e Matricula, separadamente

Chave Primária: CodEmpregado e Matrícula, em conjunto (Chave Composta)

Integridade da Informação

Um sistema SGBD deve preservar a Integridade da Informação:

- **Integridade de Entidade**

Os valores dos atributos da chave no seu conjunto não se podem repetir nem serem nulos em qualquer dos registos.

- **Integridade Referencial**

Impõe que um valor de uma chave externa tem obrigatoriamente de existir como elemento da chave primária da tabela relacionada com aquela chave externa.

Os objectivos a atingir num projecto de uma base de dados são:

- Tornar possível a inclusão de toda a informação relevante.
- Evitar a redundância ou repetição desnecessária.
- Procurar assegurar a consistência e integridade da informação.
- Proporcionar interfaces fáceis de compreender e manipular.
- Proporcionar formas de controlo do acesso à informação.

Normalização de Tabelas

Ao normalizarmos as tabelas, usando as Formas Normais, iremos eliminar a informação que se repete desnecessariamente (**Redundância da Informação**).

Existem 3 Formas Normais:

- Primeira Forma Normal – 1FN
- Segunda Forma Normal – 2FN
- Terceira Forma Normal – 3FN

1. Primeira Forma Normal – 1FN

Uma tabela está na 1ª Forma Normal (FNF - First Normal Form) quando todos os seus atributos são elementares.

Não pode existir uma lista de valores para um determinado atributo.

Vamos considerar a tabela como exemplo:

Existência_de_Peças (NºPeça, Armazém)

NºPeça	Armazém
T232	Lisboa, Porto, Leiria
H995	Aveiro, Lisboa

Esta tabela não se encontra na 1ª Forma Normal, uma vez que o atributo *Armazém* consiste numa lista de valores.

Para podermos obter a 1ª Forma Normal deveremos subdividir as diversas peças pelos diversos armazéns.

Assim, a tabela

Existência_de_Peças (NºPeça, Armazém)

NºPeça	Armazém
T232	Lisboa
T232	Porto
T232	Leiria
H995	Aveiro
H995	Lisboa

já se encontra na 1ª Forma Normal.

De notar que a chave primária passou a ser Chave Primária Composta.

2. Segunda Forma Normal – 2FN

Uma tabela está na 2ª Forma Normal (SNF - Second Normal Form) quando estiver na 1FN e os seus atributos que não são chave dependerem inteiramente da Chave Primária.

Esta situação só faz sentido quando tivermos uma **Chave Primária Composta**. Para o nosso exemplo, vamos aumentar o número de atributos.

Existência_de_Peças (NºPeça, Armazém, Qtd_Stock, Telefone, NºCaixotes)

NºPeça	Armazém	Qtd_Stock	Telefone	NºCaixotes
T232	Lisboa	467	112345644	47
T232	Porto	319	298765489	32
T232	Leiria	121	445678922	13
H995	Aveiro	578	341111154	58
H995	Lisboa	223	112345644	23

Inconvenientes de uma Tabela na 1FN

1. A Base de Dados ocupa mais espaço em disco (**Redundância da Informação**).

2. A actualização de um dos registos pode fazer perder a integridade da Base de Dados, criando uma **anomalia de actualização**.

***Por exemplo:** Se um armazém mudar o nº de telefone, ou se actualizam todos os registos que dizem respeito a esse armazém, ou a Base de Dados perde a sua integridade, uma vez que teremos o mesmo armazém com nºs de telefone diferentes.*

3. A eliminação de um registo pode provocar idêntica perda de integridade, criando uma **anomalia de apagamento**.

***Por exemplo:** Se uma peça deixar de existir, e portanto se apagar o respectivo registo, pode desaparecer a indicação de um armazém com o respectivo nº de telefone.*

Para o nosso exemplo:

- A Tabela está na 2ª Forma Normal? Qual a Chave da Tabela?
 - Se a Chave Primária não for concatenada, está na 2ª Forma Normal.
- E se a Chave Primária for concatenada?
 - Há atributos que não são chave e que dependem apenas de parte da chave? Se não, está na 2FN.
- Para o nosso exemplo, verificamos que o atributo **Telefone** só depende do atributo **Armazém** e, como tal, a tabela não se encontra na 2FN.

Vamos então transformar a tabela em duas tabelas na 2FN.

Existência_de_Peças (**NºPeça**, **Armazém**, Qtd_Stock, NºCaixotes)

NºPeça	Armazém	Qtd_Stock	NºCaixotes
T232	Lisboa	467	47
T232	Porto	319	32
T232	Leiria	121	13
H995	Aveiro	578	58
H995	Lisboa	223	23

Armazéns (**Armazém**, Telefone)

Armazém	Telefone
Lisboa	112345644
Porto	298765489
Leiria	445678922
Aveiro	341111154

De notar que a Chave Primária da Tabela *Existência_de_Peças* é: ***NºPeça***, ***Armazém***. ***Armazém***, isoladamente, funciona como Chave Externa para a tabela *Armazéns*.

De notar, também, que se resolveram todos os inconvenientes da 1FN.

3. Terceira Forma Normal – 3FN

Uma tabela está na 3ª Forma Normal (Third Normal Form) quando estiver na 2FN e nenhum atributo que não seja Chave Primária dependa de outro que também não seja chave.

Para o nosso exemplo e considerando que as peças são empacotadas em conjuntos de 10 unidades, verificamos que sabendo a **Qtd_Stock**, conseguimos determinar o **NºCaixotes**, sendo as restantes colocadas num outro caixote. Portanto, o atributo **NºCaixotes**, que não é chave, depende do atributo **Qtd_Stock**, que também não é chave. Assim a tabela não se encontra na 3FN. Para que a mesma esteja normalizada, devemos eliminar o atributo **NºCaixotes**, obtendo desta forma:

Existência_de_Peças (**NºPeça**, **Armazém**, Qtd_Stock)

NºPeça	Armazém	Qtd_Stock
T232	Lisboa	467
T232	Porto	319
T232	Leiria	121
H995	Aveiro	578
H995	Lisboa	223

Armazéns (**Armazém**, Telefone)

Armazém	Telefone
Lisboa	112345644
Porto	298765489
Leiria	445678922
Aveiro	341111154

4. Utilização de Códigos

Depois de as tabelas se encontrarem na 3FN, ainda podemos tentar simplificar o conteúdo das respectivas tabelas, utilizando códigos.

Utilizam-se códigos em 3 situações:

1. Quando não temos a garantia que o conteúdo de uma Chave Primária é único.

Exemplo: Se tivermos a tabela:

Empregado (**Nome**, Morada, Cargo)

não podemos ter a certeza que não existam 2 empregados com o nome *António Silva*. A solução será colocar um código que passará a ser a nova Chave Primária.

Empregado (**CodEmpregado**, Nome, Morada, Cargo)

2. Quando o conteúdo de um campo é uma palavra comprida que se repete várias vezes.

Para o nosso exemplo: **Armazém** é um atributo que se vai repetir várias vezes na tabela **Existência de Peças**, pelo que deveremos criar um novo código para os armazéns, obtendo desta forma:

Existência de Peças (**NºPeça**, **CodArmazém**, Qtd_Stock)

Armazéns (**CodArmazém**, Armazém, Telefone)

3. Quando queremos garantir que o conteúdo de um determinado atributo tem o mesmo valor para a mesma situação.

Exemplo: Se tivermos a seguinte tabela:

Filme (**CodFilme**, Título, Género, AnoEdição)

Se quisermos garantir que para o atributo **Género**, tenhamos sempre, por exemplo, o valor **Acção** (e não ACCÇÃO, Action, ACT, Acc., etc...), para todos os filmes cujo género seja **Acção**, deveremos criar uma nova tabela:

Géneros (**CodGénero**, Género)

Filme (**CodFilme**, Título, CodGénero, AnoEdição)