

# Bases de Dados

## Engenharia Informática (2ºano)

## Tecnologias Digitais e Gestão (2ºano)

### Capítulo 3. Modelação e Normalização de Dados

João Paulo Pereira | [jprp@ipb.pt](mailto:jprp@ipb.pt)  
Marisa Ortega | [marisa.ortega@ipb.pt](mailto:marisa.ortega@ipb.pt)  
David Dias | [davide.dias@ipb.pt](mailto:davide.dias@ipb.pt)  
Tiago Santos | [tiago.santos@ipb.pt](mailto:tiago.santos@ipb.pt)

2025



# Índice

## ● Conteúdo

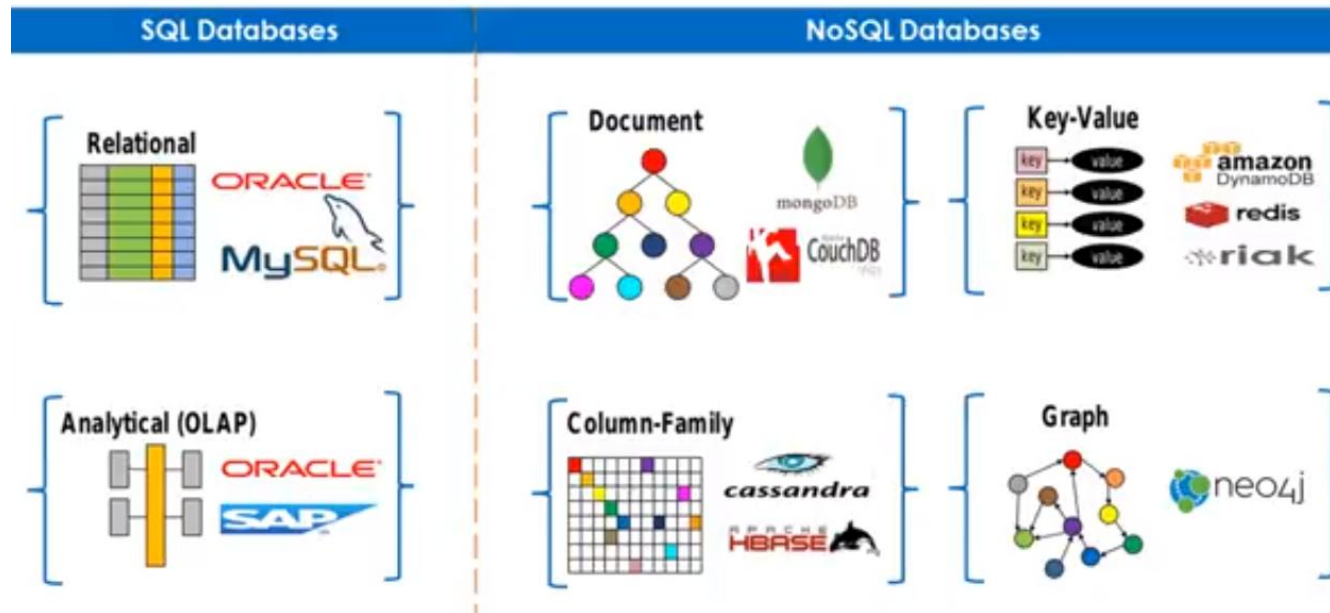
- ➔ 1. Introdução aos ambientes de Base de Dados
  - ⇒ - Noção de Sistema de Informação
  - ⇒ - A Informação nas Organizações
  - ⇒ - Tecnologias de Informação
  - ⇒ - Gestão de Informação
- ➔ 2. Sistemas de Gestão de Bases de Dados
  - ⇒ - Abordagem e Vantagens
  - ⇒ - Arquitetura de um SGBD
  - ⇒ - Tipos de Utilizadores num SGDB
- ➔ 3. Modelação e Normalização de Dados
  - ⇒ - Manutenção da Integridade
  - ⇒ - Redundância e Chaves
  - ⇒ - Diagramas E-R
  - ⇒ - Modelo Relacional
- ➔ 4. Álgebra Relacional e SQL - (MySQL)
  - ⇒ - Conceitos e aplicação de Álgebra Relacional
  - ⇒ - Ferramentas de Administração MySQL
  - ⇒ - Comandos DDL
  - ⇒ - Comandos DML
- ➔ 5. Introdução às Bases de Dados NoSQL
  - ⇒ Bases de Dados NoSQL,
  - ⇒ Tipos de Bases de Dados NoSQL,
  - ⇒ Bases de dados orientadas a documentos
- ➔ 6. MongoDB
  - ⇒ Estruturas JSON e BSON,
  - ⇒ Modelação de dados,
  - ⇒ Criação de coleções e documentos,
  - ⇒ Operações CRUD e agregação,
  - ⇒ Indexação e transações



## Tipos de SGBD mais usados

- SQL e NoSql

### SQL and NoSQL



- SGBD relacionais mais populares



- SGBD orientados a documento mais populares



## Abordagem relacional

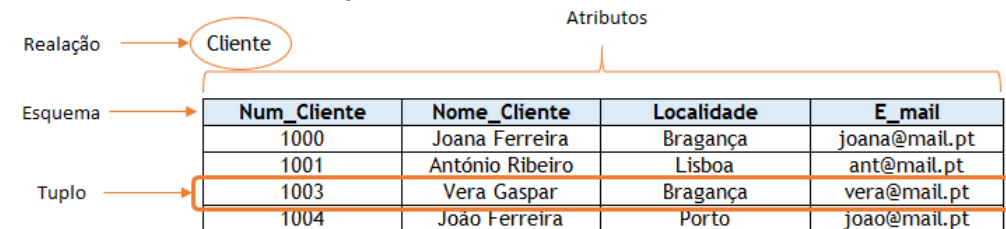
- Álgebra Relacional
  - A Álgebra Relacional é uma linguagem processual de questões.
  - Consiste num conjunto de operações que têm uma ou duas relações como entradas e produzem uma nova relação como resultado (ou saída).
  - Pode-se dizer que a Álgebra Relacional é uma linguagem formal de interrogação a bases de dados relacionais.
- As operações básicas da Álgebra Relacional são as operações equivalentes da teoria dos conjuntos (reunião, intersecção, diferença, produto cartesiano), e algumas operações específicas que são úteis ao modelo relacional:
  - Seleção de tuplos (linhas) numa relação;
  - Seleção de atributos de uma relação (projeção);
  - Junção natural de duas relações;
  - Semi-junção natural de duas relações;
  - Divisão de duas relações;
  - Renomeação de relações ou de atributos de uma relação.



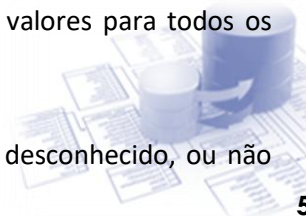
# Abordagem relacional

## ● Modelo relacional

- ➔ O modelo relacional tornou-se o modelo conceptual de bases de dados mais difundido e utilizado pelos SGBDs atualmente existentes no mercado.
- ➔ Devido à sua relativa simplicidade, o modelo relacional contribuiu decisivamente para a massificação da utilização da tecnologia de bases de dados nas organizações.
- ➔ A grande maioria dos SGBDs disponíveis no mercado atualmente são baseados no modelo relacional de dados: Oracle, SQL Server, MySQL, PostgreSQL, etc.
- ➔ O modelo relacional é baseado na teoria matemática dos conjuntos, no qual a informação e as suas associações internas são descritas:
  - ⇒ por várias **relações** (também designadas como **tabelas**), por serem constituídas por linhas e colunas.
  - ⇒ A relação é a estrutura fundamental do modelo relacional. Uma relação é uma estrutura bidimensional com um determinado **esquema**, e zero ou mais **instâncias**.
  - ⇒ O esquema de uma relação é constituído por um ou mais **atributos** (também chamados de **campos**, que correspondem a cada uma das colunas da tabela), os quais traduzem o **tipo de dados** que serão armazenados na relação. Os atributos são as características da entidade real que a relação pretende representar.



- ➔ No modelo relacional, as relações possuem algumas características que devem ser sempre tidas em conta:
  - ⇒ O valor de cada atributo num tuplo é atómico, isto é, no cruzamento de uma linha (tuplo) com uma coluna (atributo), só é possível encontrar um valor.
  - ⇒ Os atributos de uma relação devem ter identificadores distintos. Apesar de num mesmo esquema de base de dados relacional poderem surgir atributos com identificadores idênticos, estes atributos terão que pertencer a relações distintas.
  - ⇒ Os tuplos de uma relação devem ser distintos, isto é, numa relação não podem existir dois tuplos com os mesmos valores para todos os atributos. Por outras palavras, não pode haver linhas duplicadas numa relação.
  - ⇒ A ordem dos tuplos numa relação não tem qualquer significado.
  - ⇒ A ordem dos atributos no esquema de uma relação não tem qualquer significado.
  - ⇒ Todos os valores de um atributo provêm do seu domínio. Porém, o valor de alguns atributos num tuplo poderá ser desconhecido, ou não existir. Para estes casos, diz-se que o valor do atributo é nulo.



# Abordagem relacional

## ● Modelo relacional

- A arquitetura de uma base de dados relacional é vista pelo utilizador apenas como um conjunto de tabelas relacionadas, independentemente das técnicas de organização interna da informação, ao nível do suporte físico.
- O SGBD é responsável pela implementação do sistema, ao nível do software, “escondendo” do utilizador as complexidades da gestão interna dos dados e possibilitando uma visão lógica da estrutura da informação, compatível com o modelo conceptual.
- Deve-se ter em conta que o modelo de bases de dados relacionais é uma teoria:
  - ⇒ Os SGBDs permitem transformar essa teoria em prática, ou seja, permitem implementar o modelo nos computadores.
- Exemplo:
  - ⇒ As tabelas representam uma implementação das relações Clientes e Encomendas, respetivamente, usando o modelo relacional.
  - ⇒ O objetivo destas tabelas é guardar os dados referentes aos clientes e às encomendas, respetivamente, na base de dados.
  - ⇒ Estas tabelas relacionam os clientes com as encomendas através do atributo **Num\_Cliente**, que representa o número de um cliente (por exemplo: o cliente “1000 / Joana” tem 2 encomendas).

Cliente

Num_Cliente	Nome_Cliente	Localidade	E_mail
1000	Joana Ferreira	Bragança	joana@mail.pt
1001	António Ribeiro	Lisboa	ant@mail.pt
1003	Vera Gaspar	Bragança	vera@mail.pt
1004	João Ferreira	Porto	joao@mail.pt

Encomenda

Num_Cliente	Num_Encomenda	Data_Encomenda
1000	123	01/03/2022
1000	150	01/03/2022
1004	245	12/03/2022

## Abordagem relacional

### ● Modelo Entidade – Relacionamento (DER)

- ➔ O desenvolvimento de um sistema de informação baseado em bases de dados implica o desenvolvimento de um **modelo conceptual de dados** onde se descreve a estrutura lógica dos dados, que depois será traduzido para um modelo particular de base de dados.
  - ⇒ Um SGBD permite que os dados sejam definidos em termos de um modelo conceptual de dados, o qual é, por sua vez, uma descrição de alto nível dos dados.
  - ⇒ Essa descrição permite ignorar muitos detalhes de armazenamento de baixo nível, em especial as interações com o sistema operativo para o acesso aos ficheiros físicos nos quais os dados serão efetivamente guardados.
- ➔ O modelo de dados Entidade-Relacionamento (mais conhecida pela sigla E-R) foi desenvolvido para auxiliar o projeto de bases de dados, através da especificação de um esquema lógico que define a organização ou a estrutura da base de dados.
- ➔ A modelação ER é a técnica de modelação de dados mais popular, dado ser simples, de fácil compreensão e leitura e um bom meio de discussão e análise em processos de definição para esquemas de bases de dados.
- ➔ O modelo E-R é muito útil para modelar, com um diagrama simples, um sistema de informação completo.
  - ⇒ O diagrama criado é depois melhor detalhado com a ajuda de outra técnica de modelação denominada por normalização.
- ➔ Informação do SI é organizada recorrendo ao conceito de:
  - ⇒ entidade,
  - ⇒ relação e
  - ⇒ atributo.



## Abordagem relacional

### ● Modelo Entidade – Relacionamento (DER)

#### ➔ Entidade

- ⇒ Qualquer coisa, real ou abstrata, sobre a qual guardamos dados para produzir informação requerida pelo o SI em estudo.
- ⇒ Os dados que guardamos sobre as entidades são os atributos da entidade.
- ⇒ Uma entidade tem de ter pelo menos dois atributos.
- ⇒ Não há entidades “padrão” - a importância de uma “coisa” varia com a natureza e objetivo do SI em estudo.
- ⇒ Os nomes das entidades devem ser um substantivo no singular:
  - Exemplo: Aluno, Requisição, Cliente, Fornecedor, etc.
- ⇒ A entidade representa o conjunto de todas as ocorrências possíveis:
  - Exemplo: Entidade - Aluno; duas das suas ocorrências podem ser Ana Cardoso, Joaquim Santos





# Abordagem relacional

## ● Modelo Entidade – Relacionamento (DER)

### → Atributos

⇒ Para cada entidade é necessário conhecer as suas propriedades relevantes para o sistema.

⇒ Atributo é qualquer propriedade de uma entidade.

→ Exemplo: N° Cliente e Nome de Cliente

⇒ São sempre elementos atómicos (indivisíveis) de informação.

⇒ Um atributo poderá assumir diferentes valores de um domínio.

⇒ Tipos de Atributos

→ Identificadores (Chaves candidatas)

→ Descritores

Identificador

Num_Cliente	Nome_Cliente	Localidade
1000	Joana Ferreira	Bragança
1001	António Ribeiro	Lisboa
1003	Vera Gaspar	Bragança
1004	João Ferreira	Porto

⇒ De todos os atributos de uma entidade algum (ou alguns) identificam uma ocorrência específica dessa entidade distinguindo-a das restantes.

→ A estes atributos chamamos **identificadores** ou **chaves candidatas**

- Para que um atributo seja identificador é preciso que não existam duas ocorrências distintas dessa entidade em que esse atributo assuma o mesmo valor.

- Exemplo : O N° Cliente é o único identificador para cliente no caso presente

→ Aos restantes atributos, que só descrevem ou caracterizam as ocorrências da entidade, chamamos descritores.



# Abordagem relacional

## ● Modelo Entidade – Relacionamento (DER)

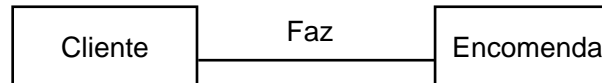
### → Relação (Associação)

- ⇒ Um **relacionamento** é uma associação entre duas ou mais entidades.
  - Os relacionamentos são introduzidos no modelo E-R depois de as entidades terem sido identificadas, porque torna-se necessário perceber e definir o modo como as entidades do sistema se associam ou se relacionam entre si.
  - A todos os relacionamentos entre conjuntos de entidades dá-se o nome de **conjunto de relacionamentos**.

- ⇒ As entidades não existem isoladas, estão associadas entre si no SI.
  - O nome da relação deve ser expresso na voz ativa, ser significativo e não muito longo.

### ⇒ Exemplo:

→ O Cliente **faz** Encomenda



- ⇒ Grau de Associação (Forma como duas entidades podem estar relacionadas):
  - Para cada ocorrência da entidade tipo A há uma e uma só ocorrência da entidade tipo B.
  - Para cada ocorrência da entidade tipo A há uma ou nenhuma ocorrência da entidade tipo B.
  - Para cada ocorrência da entidade tipo A há uma ou mais ocorrências da entidade tipo B.
  - Para cada ocorrência da entidade tipo A há uma, várias ou nenhuma ocorrência da entidade tipo B.

### ⇒ Assim, uma associação pode relacionar :

- Duas entidades entre si ( binária )
- Várias Entidades entre si ( Complexa )
- Uma entidade consigo própria ( Unária )

- ⇒ Um relacionamento entre entidades é caracterizado pela sua cardinalidade, ou seja, a quantificação de entidades que podem se relacionar através de um determinado relacionamento.

- ⇒ Considerando dois conjuntos de entidades A e B, a cardinalidade dos relacionamentos entre entidades de A e de B pode ser de 3 tipos distintos:

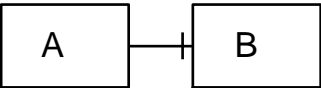
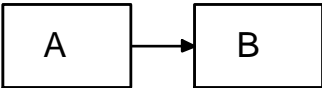
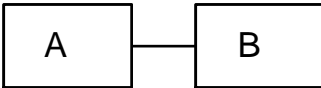
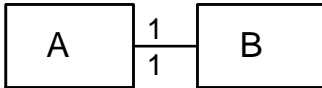
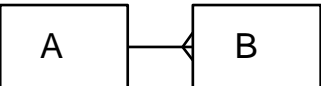
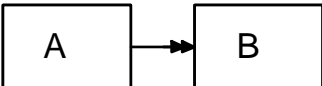
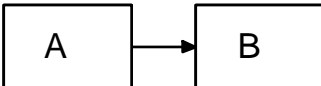
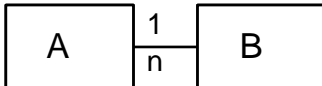
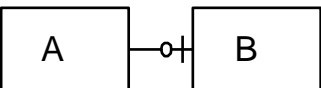
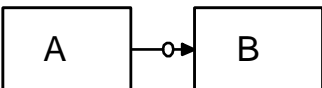
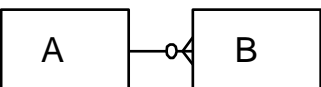
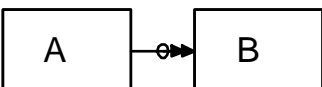
- **Um para um (1:1):** uma entidade de A está associada apenas a uma entidade de B, e uma entidade de B está associada apenas a uma entidade de A.
- **Um para vários (1:N):** uma entidade de A está associada a qualquer número de entidades de B, mas uma entidade de B só está associada a uma única entidade de A.
- **Vários para vários (N:N):** uma entidade de A pode estar associada a qualquer número de entidades de B, e uma entidade de B pode estar associada a qualquer número de entidades de A

Ao analisar um sistema, a primeira análise para distinguir entidades e associações pode socorrer-se da notação utilizada pelas organizações:

- Substantivos: Para fazer referência a entidades
- Verbos: Para fazer referência a associações

# Abordagem relacional

## ● Grau de Associação (Cardinalidade)

Grau Associação \ Notação	Pé de Galinha	Ross	Bachman	Chen
Um A está associado com um B				
Um A está associado com mais B's				
Um A está associado com zero ou um B				
Um A está associado com zero, um ou mais B's				



## Abordagem relacional

### ● Modelo Entidade – Relacionamento (DER)

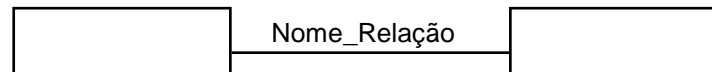
#### → Notação do DER

⇒ Entidade é sempre representada por um retângulo.

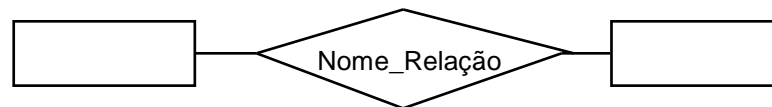
Nome\_Entidade

⇒ Tipo de Relação

→ sobre a linha



→ dentro de um losango

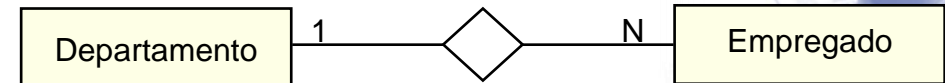


#### → Exemplo:

Uma associação que relacione funcionário com departamento pode limitar-se a indicar-nos:

⇒ Os funcionários afetos a um departamento;

⇒ A que departamento está afeto um funcionário.



Tem Afecto

## Abordagem relacional

- Como construir um Diagrama ER?

- ➔ Identificar as entidades

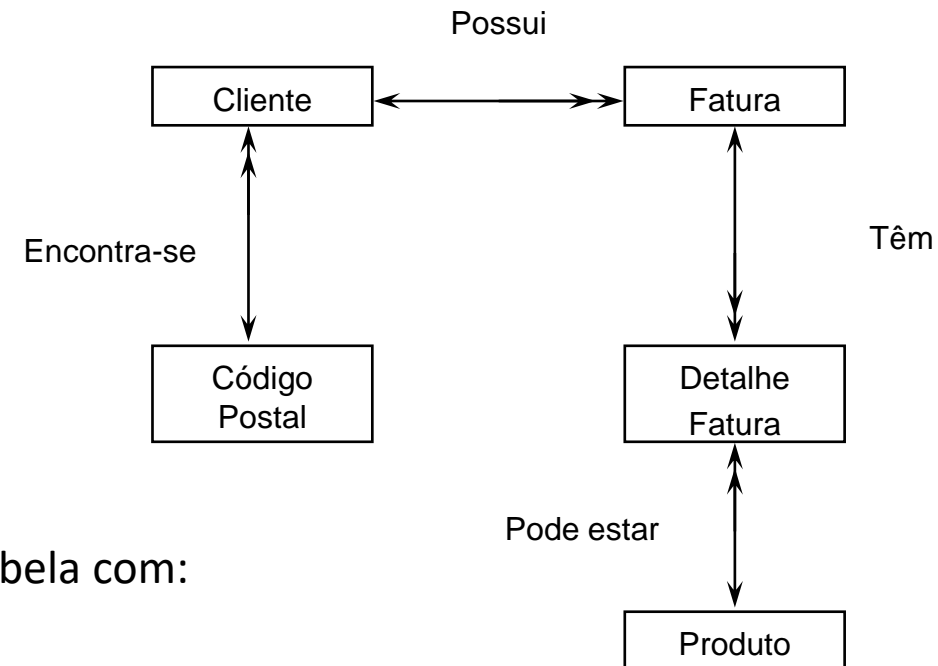
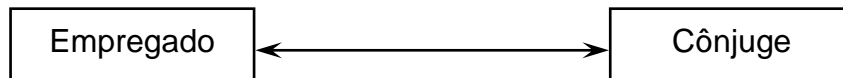
- ⇒ Sobre o que temos que guardar dados

- ➔ Olhar para cada par de entidades e verificar se existe alguma relação entre elas...

- ⇒ Se existe:

- Identificar o tipo de relação

- Identificar o grau de associação



- Cada entidade dá normalmente origem a uma tabela com:

- ➔ Identificador da entidade (Chave da Relação)

- ➔ Restantes atributos da entidade ( Descritores da Relação)

- ➔ Identificadores de outras entidades que eventualmente lhes estejam associadas Chaves estrangeiras



# Abordagem relacional

## ● Normalização

→ Técnica para analisar dados.

→ Problemas:

- ⇒ **redundância** da informação
- ⇒ mais **espaço em disco**,
- ⇒ mais **tempo de processamento**,
- ⇒ dificuldade de **manutenção da integridade** dos dados

→ O objetivo da normalização é eliminar a redundância de factos numa base de dados relacional.

- ⇒ Por outras palavras, através da análise das relações de uma base de dados relacional, são identificadas as **chaves primárias e estrangeiras dessas relações**, e as dependências funcionais existentes entre os atributos dessas relações, por forma a eliminar a redundância dos dados.

→ Características:

- ⇒ Permite derivar estruturas de dados flexíveis e sem anomalias.
- ⇒ Geralmente usado em conjunção com o diagrama ER.
- ⇒ A teoria da normalização foi construída à volta do conceito das Formas Normais
- ⇒ Diz-se que uma estrutura está numa determinada forma normal se satisfizer um conjunto de restrições.

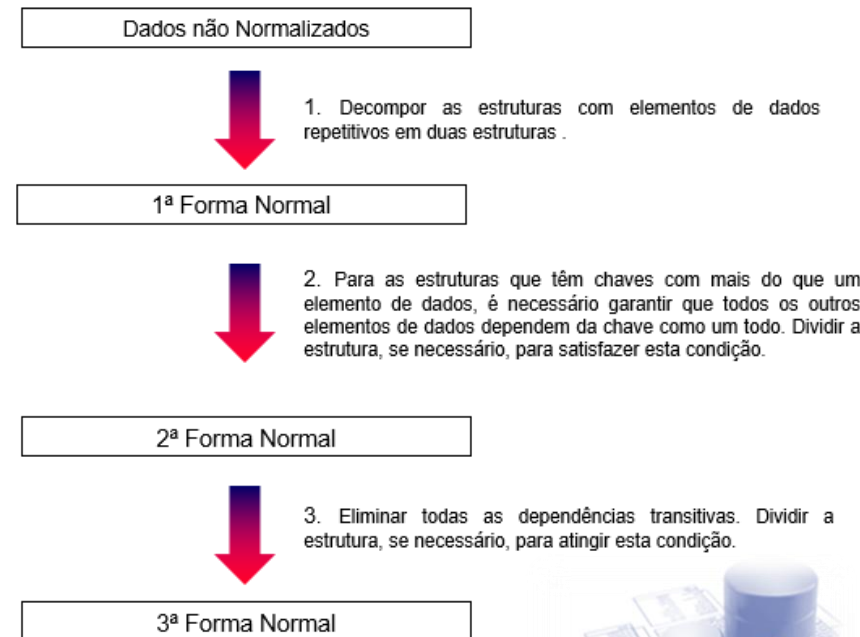
→ Uma estrutura diz-se não normalizada quando nem sequer está na 1ª FN.

→ Restrições para uma estrutura se encontrar numa determinada forma normal:

- ⇒ Uma estrutura está na **1ª Forma Normal** se não houver grupos repetitivos de atributo(s), isto é cada atributo assume um só valor.
- ⇒ Uma estrutura está na **2ª Forma Normal** se já estiver na 1ª FN e cada atributo não chave depende inteiramente da chave.
- ⇒ Uma estrutura está na **3ª Forma Normal** se já estiver na 2ª FN e não há nenhuma dependência entre atributos não-chave.

Duas metodologias (podem ser usadas isoladamente ou combinadas):

- Normalização
- Entidades e relacionamentos



# Abordagem relacional

## ● Normalização

- A normalização é um processo sistemático, conduzido por regras que traduzem as dependências funcionais entre os atributos, constituído por decomposições sucessivas de relações maiores em relações cada vez menores que, ao longo de vários passos, tenta identificar e remover a redundância própria das associações dos dados tal e qual existem no mundo real.
- Apesar de a normalização alterar a estrutura das relações, não pode alterar o conteúdo global da informação.
  - ⇒ Ou seja, durante o processo de normalização não pode haver perda de informação: deverá ser sempre possível regressar à estrutura original da base de dados.
- Para efetuar a normalização de uma base de dados, deve-se começar por analisar a informação da base de dados e estruturá-la com vista à construção das relações.
  - ⇒ Deve-se procurar incluir nessas relações todos os atributos considerados importantes.
- A normalização pode ser aplicada sobre um conjunto de dados “em bruto”, isto é, sobre a relação universal, ou então sobre as estruturas resultantes da modelação E-R, com o objetivo de as refinar.
- Redundância:
  - ⇒ A redundância ocorre sempre que o mesmo facto é armazenado mais do que uma vez na mesma base de dados, ou quando factos derivados são armazenados numa base de dados.
  - ⇒ Os problemas que a redundância pode trazer a um sistema de bases de dados podem ser resumidos nos pontos seguintes:
    - **Problemas de manutenção:** Redundância significa o armazenamento repetido dos mesmos dados. Como resultado, uma simples alteração ou remoção pode implicar o acesso a várias partes da base de dados, tornando-se difícil manter a coerência dos dados armazenados.
    - **Custos de espaço de armazenamento:** Apesar de, atualmente, não ser um problema significativo (o preço das memórias secundárias desce cada vez mais), não deixa de ser importante evitar ocupar espaço de armazenamento com algo que não acrescenta nada ao que já existe armazenado.
    - **Problemas de desempenho:** Apesar de não ser tão evidente, a redundância pode também trazer problemas de desempenho ao sistema. A razão para isso é que a razão mais importante para um mau desempenho de um SGBD são os acessos aos discos onde as bases de dados se encontram armazenadas. Se a redundância for significativa, isso implicará mais acessos aos discos para trazer os mesmos dados.

# Abordagem relacional

## ● Normalização – Exemplo

- ➔ Informalmente, a normalização consiste em dividir uma tabela inicial em várias tabelas minimizando a redundância da informação.
- ➔ Iniciar pela identificação de uma chave
  - ⇒ A chave é um conjunto de campos, que identifica de forma única cada registo da tabela (Pode ser simples ou composta)

### a) Estrutura não-normalizada

Numero\_aluno  
 Nome  
 Morada  
 Nome\_mae  
 Nome\_pai  
 Departamento  
 Codigo\_curso  
 Nome\_curso  
**Disciplinas\***  
 Codigo\_disciplina  
 Nome\_disciplina  
 Nota

### b) Estrutura na 1ª Forma Normal

Numero\_aluno  
 Nome  
 Morada  
 Nome\_mae  
 Nome\_pai  
 Departamento  
 Codigo\_curso  
 Nome\_curso

Numero\_aluno  
Codigo\_disciplina  
 Nome\_disciplina  
 Nota

(resultaram 2 estruturas na 1ªFN)

### c) Estrutura na 2ª Forma Normal

Numero\_aluno  
 Nome  
 Morada  
 Nome\_mae  
 Nome\_pai  
 Departamento  
 Codigo\_curso  
 Nome\_curso

Numero\_aluno  
Codigo\_disciplina  
 Nota

Codigo\_disciplina  
 Nome\_disciplina

(resultaram 3 estruturas na 2ª FN)

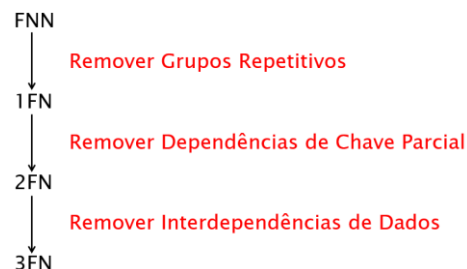
### d) Estrutura na 3ª Forma Normal

Numero\_aluno  
 Nome  
 Morada  
 Nome\_mae  
 Nome\_pai  
 Codigo\_curso  
Codigo Curso  
 Nome Curso

Numero\_aluno  
Codigo\_disciplina  
 Nota

Codigo\_disciplina  
 Nome\_disciplina

(resultaram 4 estruturas na 3ª FN)





# Abordagem relacional

## ● Normalização

### ➔ O porquê da 1ªFN, 2ªFN e 3ªFN

⇒ Porquê a 1ªFN?

→ Base de dados relacionais não permitem registos de comprimento variável.

⇒ Porquê a 2ª FN?

→ Evitar redundância e consequentemente evitar inconsistência.

→ Diminuir o espaço necessário para guardar a informação.

⇒ Porquê a 3ª FN?

→ Evitar redundância e consequentemente evitar inconsistência.

→ Diminuir o espaço necessário para guardar a informação.

Exemplo de uma estrutura que não satisfaz a 2ªFN:

Numero\_aluno

Codigo\_disciplina

Nome\_disciplina

Nota

- O Nome\_disciplina é repetida para todas as ocorrências referentes a um aluno diferente (outro Numero\_aluno) mas que frequenta a mesma cadeira (Codigo\_disciplina igual).
- Se temos 1000 alunos diferentes a frequentar a mesma disciplina repetimos 1000 vezes o mesmo nome da disciplina ---> redundância
- Se por qualquer razão muda o nome da disciplina temos que alterá-lo 1000 vezes ---> possibilidade de inconsistência

Exemplo de uma estrutura que não satisfaz a 3ªFN:

Numero\_aluno

Nome

Morada

Nome da Mãe

Nome do Pai

Departamento

Código Curso

Nome Curso

- O Nome\_curso é repetida para todas as ocorrências referentes a um aluno diferente (outro Numero\_aluno) que frequenta o mesmo curso (Código\_curso igual).
- Se temos 1000 alunos diferentes a frequentar o mesmo curso repetimos 1000 vezes o mesmo nome da curso ---> redundância
- Se por qualquer razão muda o nome do curso temos que altera-lo 1000 vezes ---> possibilidade de inconsistência

# Abordagem relacional

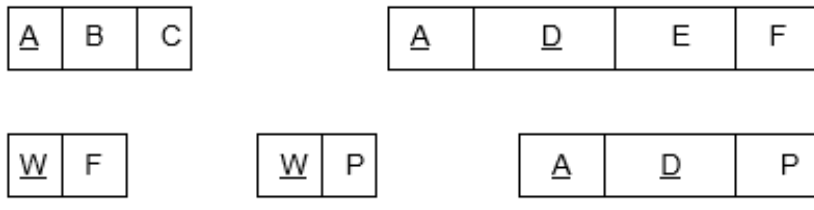
## ● Normalização

→ Estruturas Finais - Combinar as diferentes estruturas resultantes da normalização numa estrutura final também normalizada.

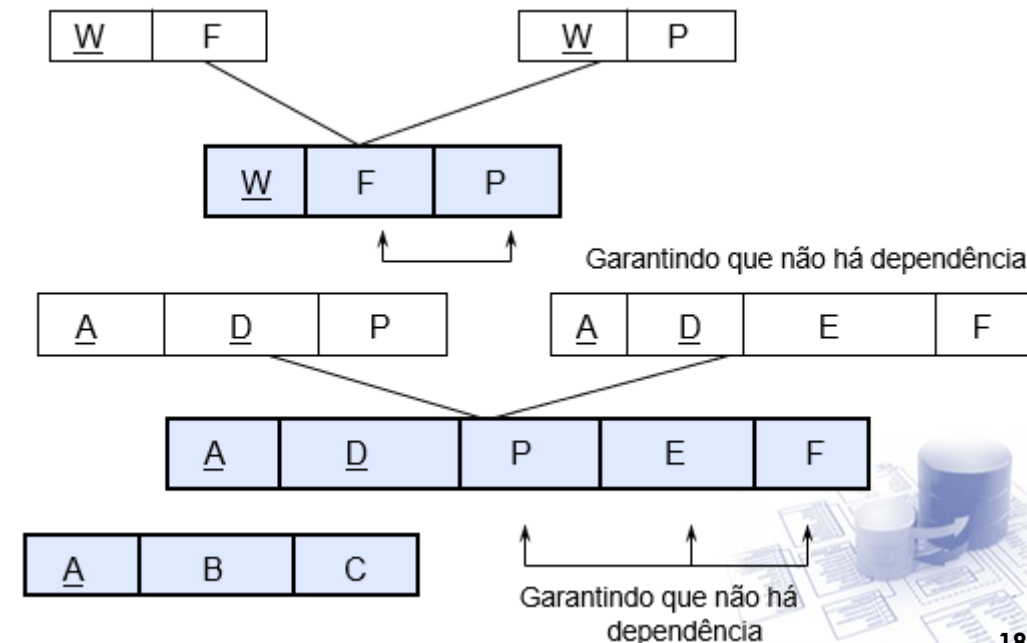
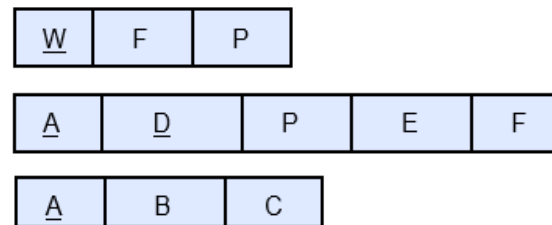
→ **Todas as estruturas resultantes da normalização, (3ªFN) e que tenham o mesmo elemento de dados como chave, podem ser combinadas numa estrutura única garantindo que essa estrutura final também se encontra na 3ª FN.**

→ Exemplo:

⇒ Após normalização das diferentes estruturas definidas num determinado SI obtiveram as seguintes estruturas na 3ªFN.



⇒ Estruturas finais:



# Abordagem relacional

## ● Normalização (Resumo)

Em geral, o processo de normalização é aplicado sobre uma estrutura não normalizada designada de relação universal, a qual deve conter todos os atributos dos quais é necessário guardar informação.

➔ Passo 1: Nesta fase inicial do processo, o analista deve:

- Identificar quais os atributos que poderão ser repetidos.
- Identificar os atributos que poderão ser chave: um atributo que identifique os atributos não repetidos, e um outro atributo para identificar os atributos repetidos.

➔ Passo 2: Aplicar as regras da 1.ª forma normal, isto é, separar os atributos repetidos dos atributos não repetidos. Para isso, o analista deve:

- Converter os atributos não atómicos em atributos atómicos, para que não seja possível incluir mais do que um valor em cada atributo de uma relação.
- Separar os atributos repetidos dos não repetidos, passando a considerá-los como elementos de uma nova relação. A chave primária da relação dos atributos não repetidos deve ser acrescentada, como chave estrangeira, à chave primária dos atributos repetidos.

➔ Passo 3: Aplicar as regras da 2.ª forma normal; para isso, terá de verificar se os atributos não-chave de uma relação dependem da totalidade da chave primária dessa relação.

- Se a relação só tem um atributo como chave primária, e se essa relação já estiver na 1.ª forma normal, então a relação também se encontra na 2.ª forma normal.
- Se a chave primária é composta, e se algum atributo não-chave depende apenas de uma parte da chave primária, então a relação deverá ser decomposta, para que cada atributo dependa da totalidade da chave primária.

➔ Passo 4: aplicação das regras da 3.ª forma normal, a qual obriga a analisar a dependência dos atributos não-chave de cada relação.

- O analista deverá procurar dependências funcionais entre os atributos não-chave de uma relação.
- Se uma relação que já esteja na 2.ª forma normal tiver, no máximo, um atributo não-chave, então essa relação também se encontra na 3.ª forma normal.
- Se existir algum conjunto de atributos não-chave de uma relação que tenha dependência funcional em relação a um outro conjunto de atributos não-chave daquela mesma relação, então a relação deverá ser decomposta, de modo que qualquer atributo não-chave dessa relação só dependa da chave primária da relação.

