

Sistemas de Bases de Dados
Notas de Leitura

02 >> O Ciclo de Vida do Desenvolvimento de uma Base de Dados

Orlando Belo

Departamento de Informática, Escola de Engenharia, Universidade do Minho
PORTUGAL

> www.di.uminho.pt/~omb
> www.researchgate.net/profile/Orlando_Belo

2020

02



Estrutura da Apresentação

- Introdução
- O Ciclo de Vida de Desenvolvimento de uma Base de Dados
- Definição do Sistema - Contextualização, Fundamentação e Viabilidade
- Levantamento e Análise de Requisitos
- Modelação Conceptual de uma Base de Dados
- Modelação Lógica de uma Base de Dados



1

Introdução

- Representar e armazenar informação.
- Modelos de dados, diagramas e esquemas.
- Construção de modelos de dados.



Introdução

- As bases de dados são **elementos computacionais importantes** que marcam presença na grande generalidade das aplicações de software.
- Independentemente da sua natureza, as bases de dados são peças de enorme importância no intrincado mundo do software, desempenhando um papel vital na sua operacionalidade: **o armazenamento de informação**.



Introdução

- A construção de uma base de dados assenta, em termos gerais, num processo de desenvolvimento, bastante sustentado, de um modelo de dados, que representam, de alguma maneira, a forma como a informação é representada e acolhida numa base de dados.
- Basicamente, os modelos de dados são constituídos por diagramas, frequentemente designados por esquemas, que representam a tradução de um conjunto de requisitos a diferentes níveis de abstração.



Introdução

- Todavia, para que um modelo de dados possa cumprir de forma adequada a sua função, é necessário que seja projetado e desenvolvido corretamente, refletindo, tanto quanto possível, a realidade da necessidade de armazenamento de dados de uma dada aplicação.
- A construção de um modelo de dados, em particular, e de uma base de dados, em geral, não deve ser realizada de forma ad hoc, devendo-se fazê-lo de uma forma metódica, seguindo padrões de modelação e implementação já estabelecidos.



2

O Ciclo de Vida do Desenvolvimento de uma Base de Dados

- Caracterização geral.
- As etapas de desenvolvimento.
- Uma metodologia de desenvolvimento.

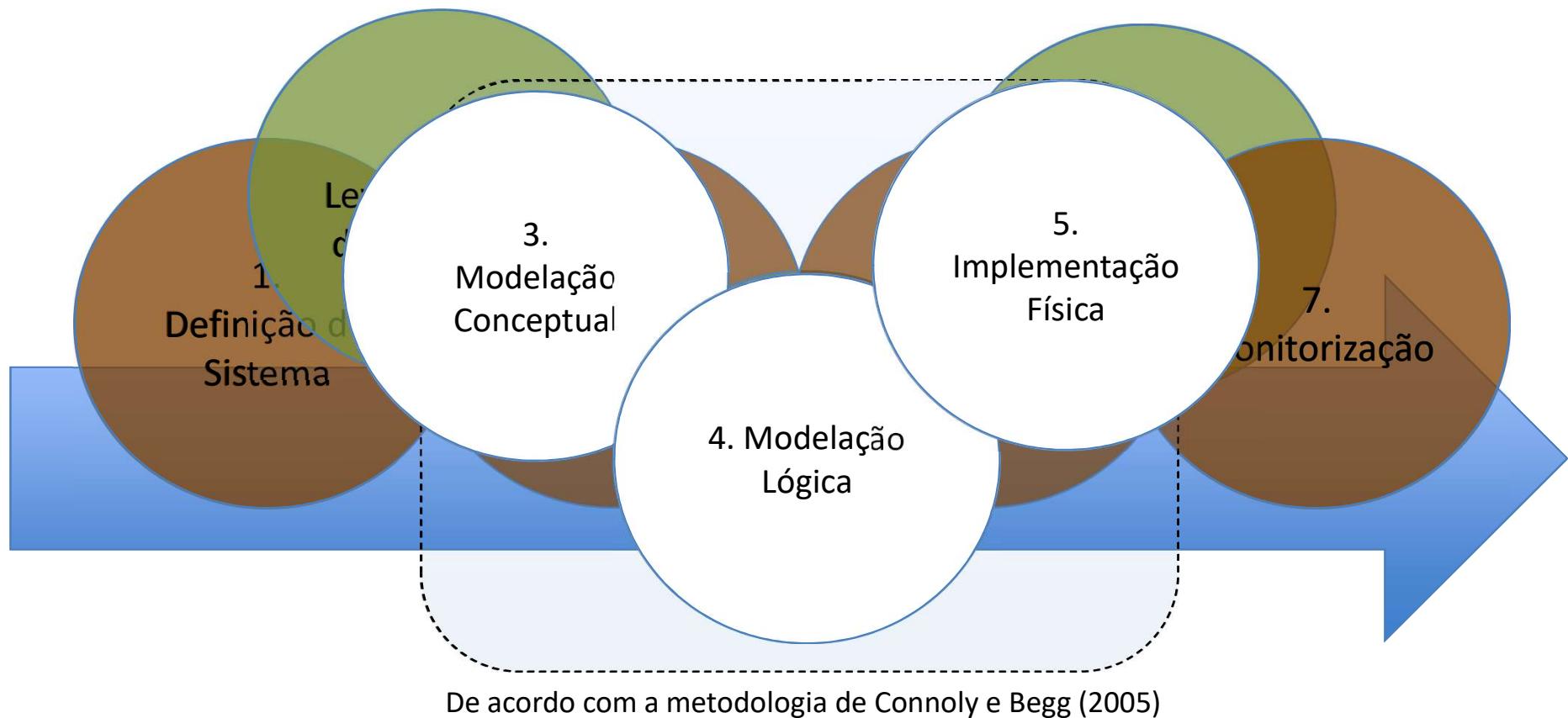


O Ciclo de Vida

- O ciclo de vida de uma base de dados integra um conjunto de tarefas específicas que usualmente são realizadas ao longo do processo de desenvolvimento de uma base de dados.
- A forma como iniciamos e desenvolvemos o processo de construção de uma base de dados não deve ser feita de uma forma qualquer, sem método, sem qualquer planeamento – “vamos fazendo e depois veremos”, mas sim adotando e seguindo uma determinada metodologia de desenvolvimento.



O Ciclo de Vida



Alguns Aspectos Importantes

- No processo de desenvolvimento de uma base de dados é importante ter em consideração alguns **aspectos**, que se revelaram no passado muito importantes para **garantir o sucesso** do desenvolvimento de uma base de dados. De referir:
 - Seguir uma **metodologia estruturada** durante o processo de modelação.
 - Trabalhar com os **utilizadores** interactivamente tanto quanto possível.
 - Utilizar uma **aproximação orientada/conduzida pelos dados**.
 - Incorporar nos modelos de dados considerações estruturais e de **integridade**.
 - Combinar técnicas de conceptualização, normalização e de validação de transações na metodologia de modelação de dados.



Alguns Aspectos Importantes

- e ainda:
 - Utilizar tanto quanto possível aproximações baseadas em **diagramas** para representar os modelos de dados.
 - Utilizar **uma linguagem de desenho** de bases de dados no sentido de representar informação semântica adicional sobre os dados.
 - Construir um **dicionário de dados** no sentido de complementar os diagramas do modelo de dados.
 - Ser capaz e ter **vontade e disponibilidade** para repetir **algumas das ações** realizadas durante o processo de modelação de dados.



Uma Metodologia

- Connolly e Begg (2005) propuseram uma **metodologia** para suporte à análise e à modelação de um sistema de bases de dados pode integrando três das etapas que anteriormente apresentámos para o ciclo de vida de uma base de dados, nomeadamente:
 - (Parte I) **Desenho Conceptual da Base de Dados**. Processo de construção de um modelo da informação utilizada numa organização independente de todas as considerações físicas.
 - (Parte II) **Desenho Lógico da Base de Dados**. Processo de construção de um modelo de informação utilizado numa empresa baseado num modelo de dados específico, mas independente de qualquer sistemas de gestão de bases de dados ou de outras considerações físicas.



Uma Metodologia

- (Parte III) Desenho Físico da Base de Dados. Processo de descrição de uma descrição da implementação da base de dados em sistemas de armazenamento secundária, o que inclui a descrição das estruturas de armazenamento e métodos de acesso utilizados para se obter um acesso eficiente aos dados.



3

Definição do Sistema (Fase I) Contextualização, Fundamentação e Viabilidade

- Contextualização de uma base de dados.
- Fundamentação do processo de desenvolvimento.
- Análise de Viabilidade.



O Problema

- É importante que antes de se começar a realizar um processo de levantamento de requisitos ou a construir um qualquer modelo para uma base de dados, **se conheça de forma detalhada e suportada o porquê** de levar a cabo um processo de construção de uma base de dados.
- De facto, a **contextualização, a fundamentação e a realização de uma análise de viabilidade** contribuem significativamente para a compreensão do problema por toda a equipa de desenvolvimento, bem como contribui para o estabelecimento de um melhor processo de levantamento de requisitos e, consequentemente, de construção da base de dados.



Contextualização

- A contextualização de um processo de desenvolvimento de uma base de dados deve anteceder o próprio processo de construção, sendo essencial para se conhecer a organização de acolhimento, o meio operacional, os processos envolvidos e os seus intervenientes.
- Na definição da contextualização deve-se ter o cuidado de definir de forma concreta as “fronteiras” do problema que temos em mãos, de forma a não abordar áreas ou processos irrelevantes.



Fundamentação

- A fundamentação do processo de desenvolvimento de uma base de dados deve **descrever em traços gerais o porquê da necessidade da base de dados**, apresentando de forma breve os sistemas e processos que suportará, como estes interagirão com ela e que informação a base de dados irá acolher.
- Além disso deverá também elucidar como é que a base de dados se integrará na estrutura computacional da organização.



Análise de Viabilidade

- Analisar a viabilidade a construção de uma base de dados deve ser sustentada por **um pequeno estudo** que deve ser realizado numa fase preliminar do levantamento de requisitos (ou antes), cujo objetivo principal é fornecer a resposta para questões como:
 - Quais os principais **objetivos** a atingir com a construção da base de dados?
 - Será que a base de dados poderá ser construída de acordo com o **calendário** estabelecido e com os **recursos** disponíveis?
 - Os sistemas cuja operacionalidade será suportada pela base de dados estão **preparados** adequadamente?
 - (...)



4

Levantamento de Requisitos (Fase II)

- Identificação e caracterização de elementos relevantes.
- Requisitos e sua coleta.
- Organização de requisitos.



Levantamento de Requisitos

- O levantamento de requisitos num processo de construção de uma base de dados é uma tarefa fundamental e crucial para seu o sucesso.
- Neste processo procura-se **identificar e caracterizar os elementos de dados mais relevantes** para o problema em causa, que precisam por um qualquer motivo de serem guardados na futura base de dados.



Levantamento de Requisitos

- Existem diversas técnicas que podem ser utilizadas numa tarefa de levantamento de requisitos para a construção de uma base de dados. Algumas das técnicas mais utilizadas são:
 - Análise de documentação existente sobre o domínio em causa.
 - Realização de entrevistas a todos que realizam tarefas no domínio em causa.
 - Observação dos processos que se realizam dentro do domínio de aplicação.
 - Preparação e realização de questionários.
 - Investigação de processos operacionais.
 - Análise de sistemas existentes.



Levantamento de Requisitos

- Num processo de levantamento de requisitos devemos ter em consideração algumas tarefas importantes, nomeadamente:
 - Fazer a identificação da informação essencial do problema real.
 - Remover informação redundante ou irrelevante para o problema em causa.
 - Clarificar informação recolhida que esteja pouco clara.
 - Preencher eventuais falhas ou lacunas encontradas.
 - Saber identificar os dados e os processos que trabalham com eles.



Levantamento de Requisitos

- Ao fazermos o levantamento de requisitos num processo de desenvolvimento de uma base de dados devemos ter o cuidado de os **organizar** de acordo com as diversas vertentes de implementação.
- Sugiro que se faça essa organização **dividindo os requisitos em classes**, de acordo com o seu âmbito de aplicação, nomeadamente:
 - Administração – acesso, monitorização, segurança, etc.
 - Exploração – queries, relatórios, procedimentos, etc.
 - Descrição – objetos de dados, domínios, restrições, etc.



Exemplos de Requisitos de Descrição

- **RD1** – “A caracterização de um cliente na base de dados deve incluir um identificador único, o seu nome, em formato ‘<apelido>,<nome próprio>’, um endereço postal, ...”.
- **RD2** – “O número de um cliente deve incorporar o número da zona comercial à qual pertence, complementado com um número interno, gerado sequencialmente pelo departamento comercial”.
- **RD3** – “Todos os regtos relativos à entrada de produtos no armazém geral devem ser numerados sequencialmente, não sendo permitido reutilizar números de regtos anteriores que tenham sido removidos”.
- (...).



Exemplos de Requisitos de Exploração

- RE1 – “Os dados relativos a um dado cliente devem poder ser acedidos através do número do cliente, número de contribuinte ou número de contacto”.
- RE2 – “Os dados das vendas realizadas devem ser apresentados sempre ordenados temporalmente, mesmo nos casos em que se apliquem filtros para a seleção de dados ou de limitação de resultados”.
- (...).



Exemplos de Requisitos de Administração

- RA1 – “Os dados dos fornecedores da empresa só podem ser acedidos pelos funcionários com credenciais atribuídas pelos departamentos de aprovisionamento, de finanças ou de administração”.
- RA2 – “Os funcionários do departamento de vendas apenas podem realizar operações de consulta sobre a tabela de artigos”.
- RA3 – “A remoção de registos apenas pode ser realizada pelo gestor operacional do sistema”.
- (...).



5

- Características gerais.
- Etapas de desenvolvimento.
- Vistas de utilizadores.
- Elementos básicos de um modelo.
- Revisão do modelo.

Desenho Conceptual de uma Base de Dados (Fase III)



Desenho Conceptual

- O **desenho conceptual** de uma base de dados consiste na construção de um modelo de informação utilizado numa organização que é independente de todas as considerações **físicas**, tais como:
 - Sistema de gestão de bases de dados alvo.
 - Programas aplicacionais.
 - Linguagens de programação.
 - (...)



Uma Vista, um Modelo

- Uma **vista de um utilizador**, ou vista de utilização, é o conjunto de dados requerido por um determinado utilizador para tomar uma decisão ou realizar uma tarefa - tipicamente é uma **área funcional de uma organização**.
- Podemos identificar as **vistas dos utilizadores** durante a fase de levantamento de requisitos, através de métodos como:
 - Análise de diagramas de fluxos de dados.
 - Realização de entrevistas aos utilizadores.
 - Análise e estudo de procedimentos, relatórios, formulários.
 - Observar o modo de funcionamento da organização.
- No modelo de dados conceptual as **vistas dos utilizadores** que pretendemos modelar são designadas, cada uma delas, como **modelo de dados conceptual local**.

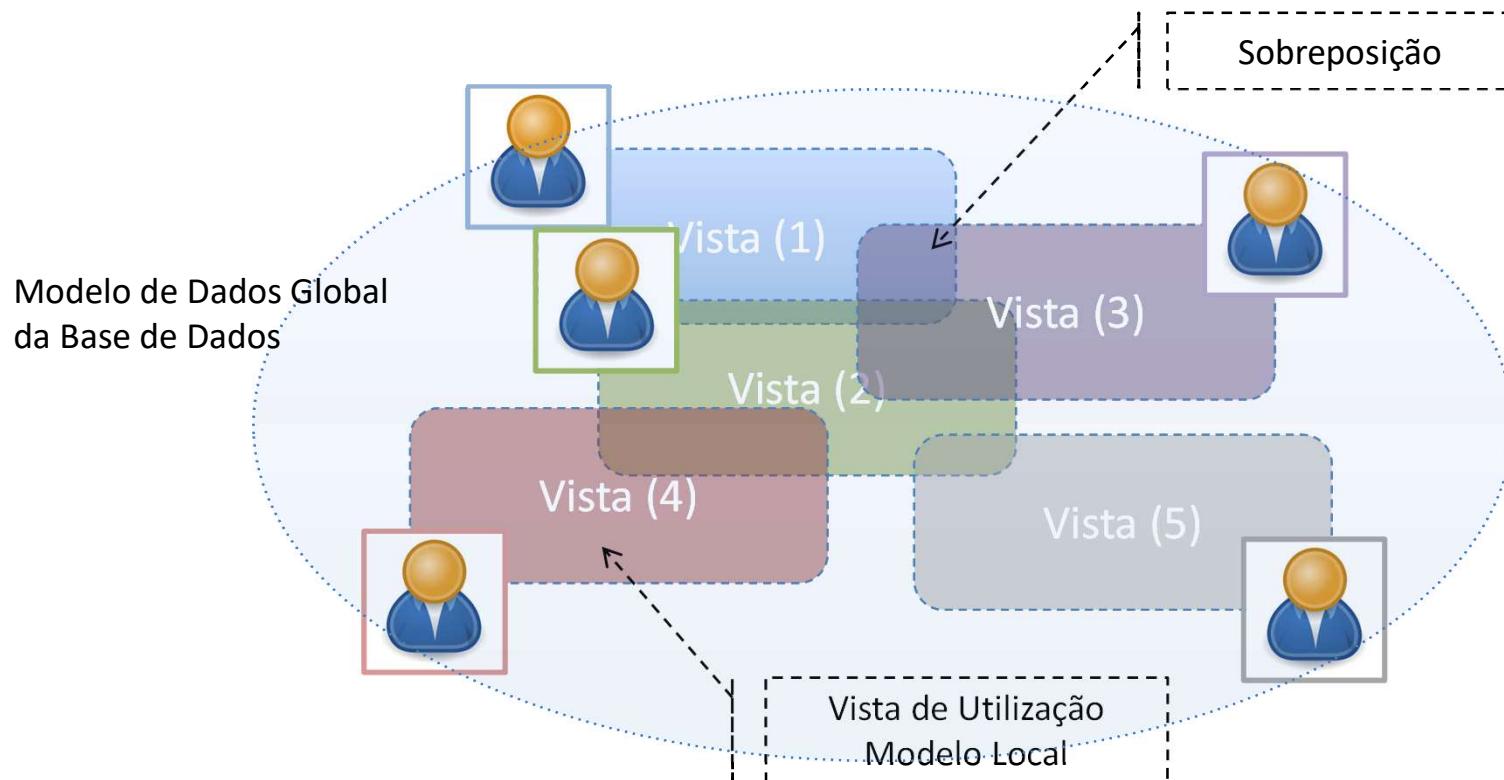


Modelos de Dados Locais

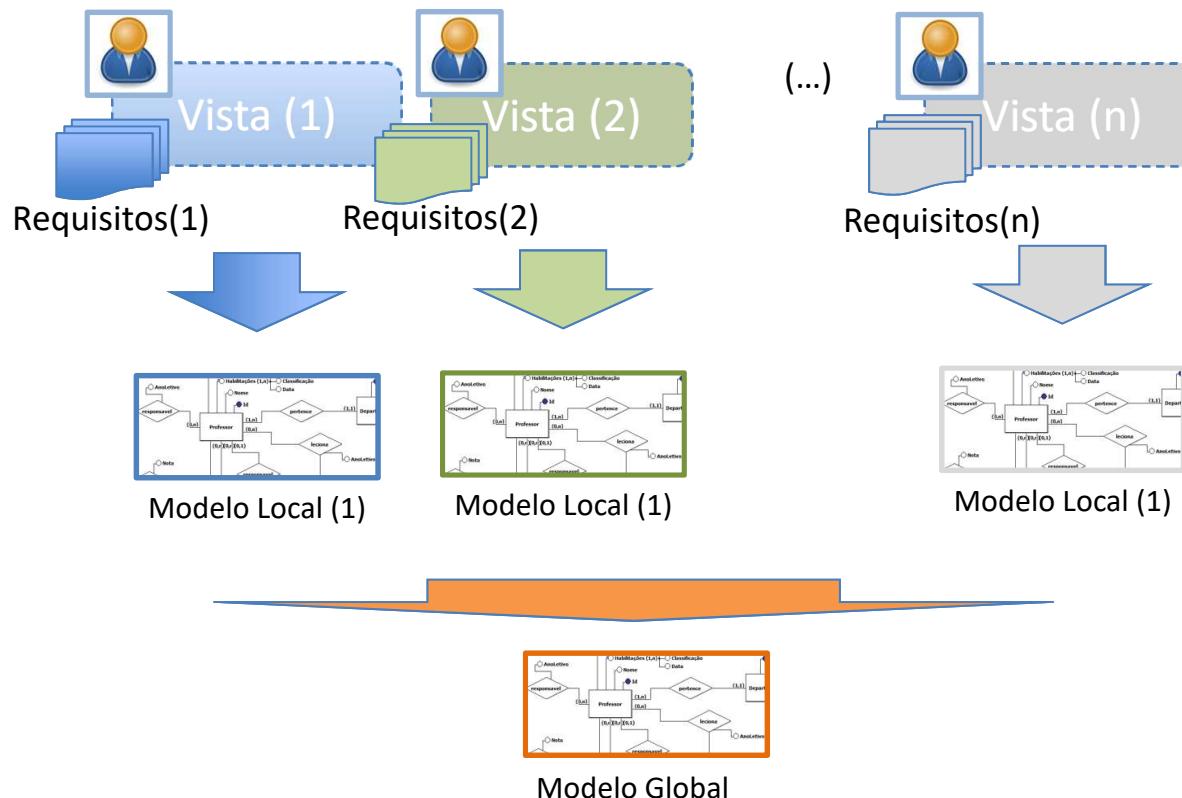
- Um **modelo de dados local** representa apenas os dados relativos a uma determinada **vista de utilização**, correspondendo apenas a uma parte bem definida do problema que temos em mãos para implementar, acolhendo apenas as especificidades de armazenamento de um parte do modelo de negócio.
- O seu objetivo principal é **facilitar o processo de modelação** de um processo ou das atividades de um dado utilizador, sendo uma visão parcial ou uma descrição redefinida do esquema global da base de dados.



Modelos de Dados Locais



Integração de Vistas



Modelo Conceptual Local

- Cada um dos modelos de dados conceptuais locais que seja definido deverá integrar os seguintes elementos:
 - Tipos de entidades.
 - Tipos de relacionamentos.
 - Atributos.
 - Domínios dos atributos.
 - Chaves candidatas.
 - Chaves primárias.



Etapas da Modelação Conceptual

- O **modelo de dados conceptual** é suportado por um conjunto muito vasto e diversificado de documentos que vão sendo gerados ao longo do seu processo de desenvolvimento.
- O processo de desenvolvimento do modelo de dados conceptual pode incluir as seguintes **fases**:
 1. Construção do modelo de dados conceptual local para cada vista de utilizador.
 2. Identificação dos tipos de entidades.
 3. Identificação dos tipos de relacionamentos.



Etapas da Modelação Conceptual

- E ainda:
 4. Identificação e associação dos atributos com tipos de entidades ou relacionamentos.
 5. Identificação dos domínios dos atributos.
 6. Identificação dos atributos chave.
 7. Detalhe ou generalização dos tipos de entidades.
 8. Desenho de diagramas ER.
 9. Revisão do modelo de dados com o utilizador.



Entidades

- O principal objectivo desta primeira fase do processo é a **identificação dos principais tipos de entidades** presentes na vista do utilizador da organização.
 - Uma entidade é um objecto ou um conceito que está claramente identificado numa organização como tendo uma existência independente.
 - Um dos métodos para identificar as entidades é através da **análise da especificação de requisitos** de uma dada função de um utilizador dentro da organização – ex.: nomes, identificações pessoais, endereços, etc.



Entidades

- Uma forma alternativa de identificação pode ser realizada através da **observação de objetos** que têm uma existência por direito próprio – ex.: mobiliário, pessoal, equipamentos, etc.
- Por vezes é difícil identificar as entidades, dada a possibilidade **dos utilizadores confundirem conceitos e matérias ou utilizarem frequentemente sinónimos e homónimos**, ou simplesmente devido a não ser possível fazer uma identificação óbvia.
- Os **processos de análise são subjetivos** - diferentes projetistas e desenhistas têm diferentes critérios de análise de abordagem de problemas, gerando consequentemente diferentes conjuntos de entidades para um mesmo problema.



Entidades

- A definição das diversas entidades de um modelo conceptual deve ser devidamente documentada, de forma a que seja possível entender o papel de cada uma delas dentro da bases de dados.
- Conforme o processo de identificação avança devemos atribuir a cada entidade designações óbvias e com significado
 - os nomes e as descrições de cada uma das entidades identificadas devem ser registadas no dicionário de dados, assim como, se possível, o seu número de ocorrências e os seus diversos sinónimos se existirem.



Entidades

- A caracterização de uma entidade deverá incluir os seguintes elementos:
 - Nome da entidade.
 - Descrição.
 - Sinónimos e outras designações conhecidas.
 - Ocorrência.



Relacionamentos

- O objectivo desta fase é **a identificação dos relacionamentos** que possam existir entre as entidades que foram identificados na fase anterior.
- A identificação dos relacionamento pode ser feita com base nos requisitos dos utilizadores ou nas formas como são referidas eventuais ligações que existem entre as entidades.
- Os relacionamentos na maioria dos casos são **binários** - envolvem duas entidades – mas poderão existir relacionamentos mais **complexos** envolvendo três ou mais entidades ou relacionamentos recursivos que envolvem uma única entidade.



Relacionamentos

- Na identificação de um relacionamento devemos ter em conta os seguintes aspectos:
 - Determinar a cardinalidade e restrições de participação dos tipos de relacionamentos.
 - Documentar os tipos de relacionamentos.
 - Utilizar diagramas Entidade-Relacionamento (ER) no processo de modelação.



Diagramas ER

- Um **modelo entidade-relacionamento** “é um instrumento gráfico de enorme valor para todos aqueles que estão envolvidos no processo de desenvolvimento de uma base de dados.
- Segundo Peter Chen, “*The entity-relationship model adopts the more natural view that the real world consists of entities and relationships. It incorporates some of the important semantic information about the real world.*”



(Representação de um esquema com a nomenclatura de Chen.)

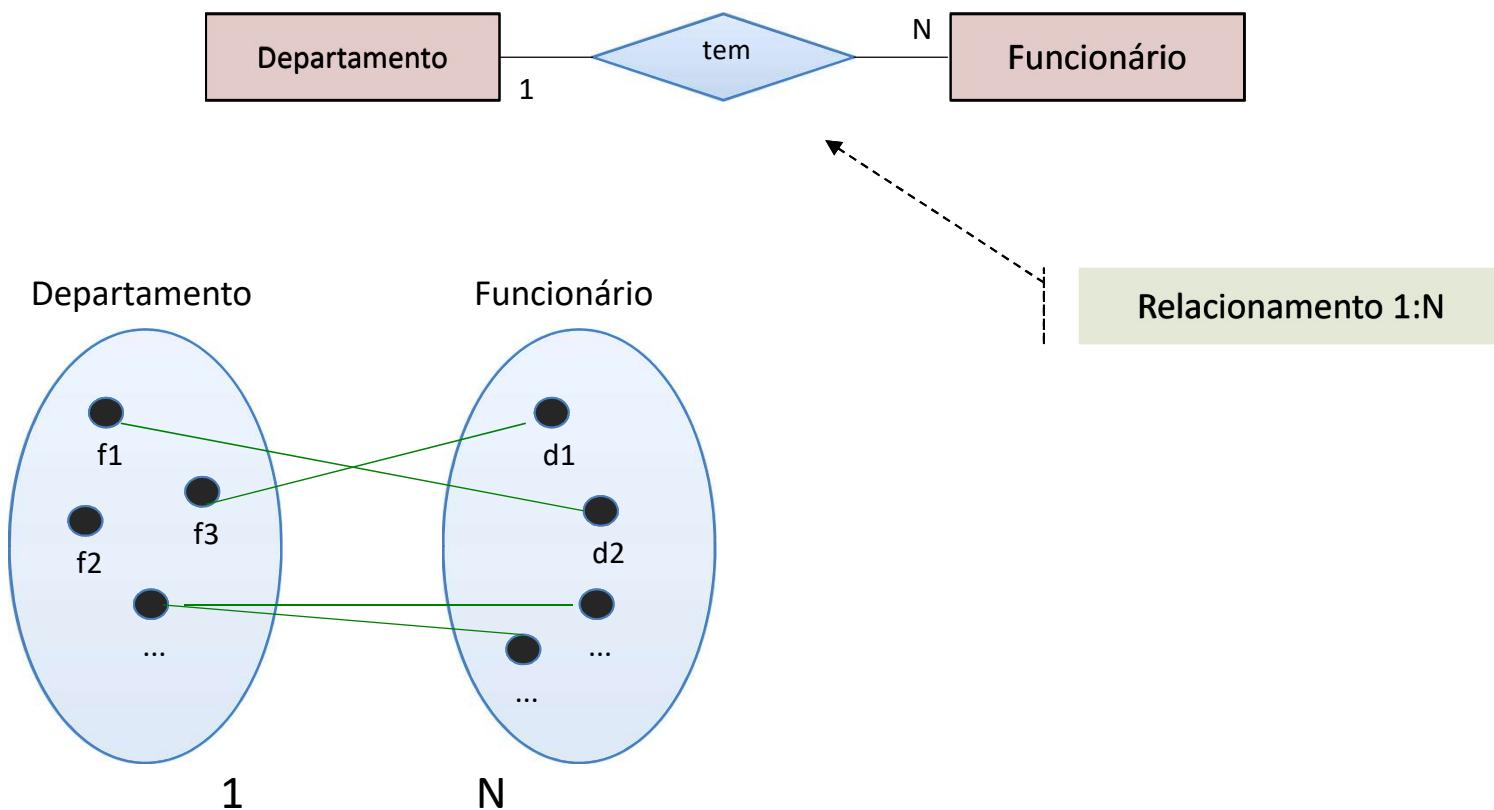


Relacionamentos

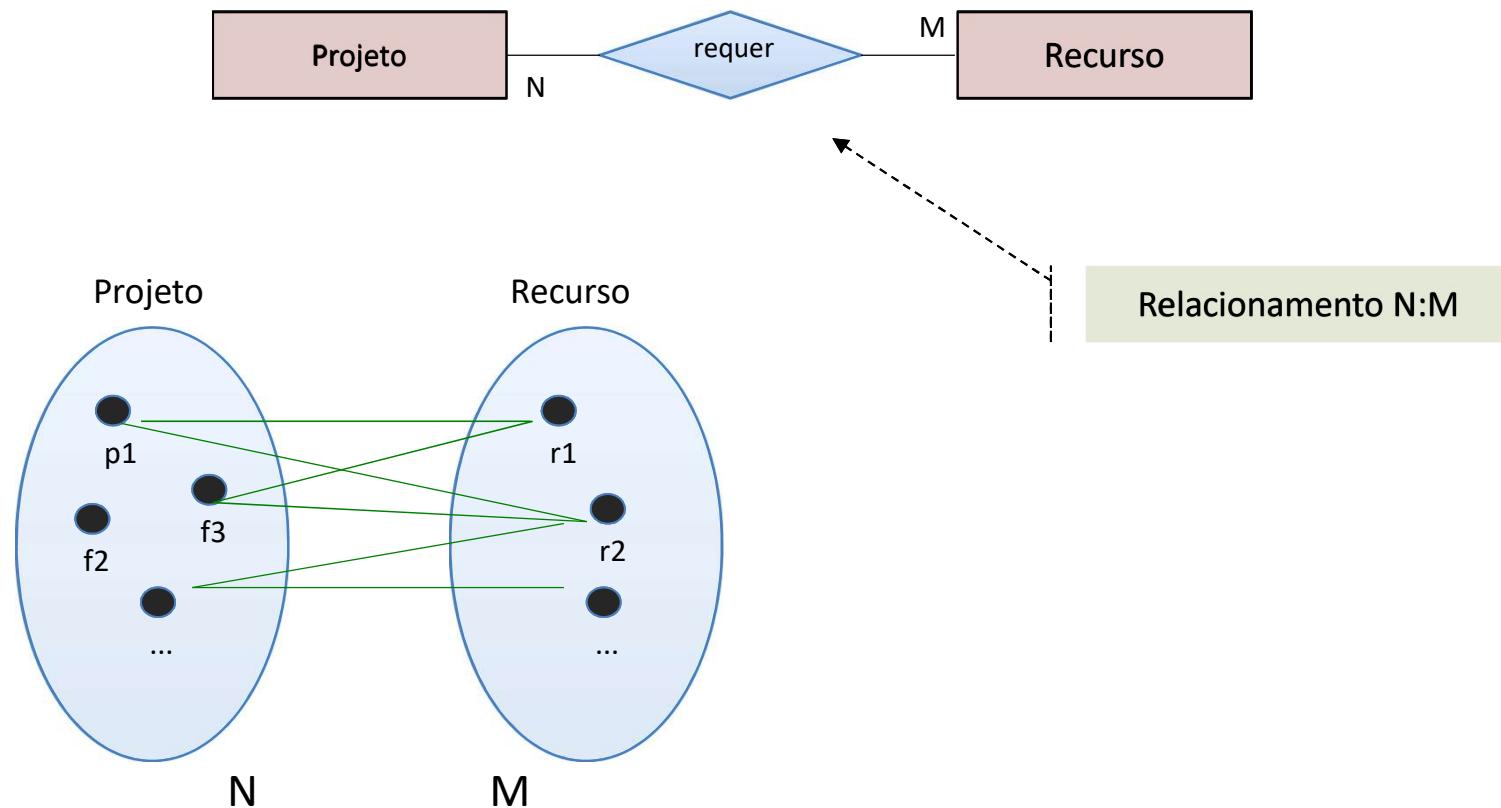
- A documentação sobre um relacionamento deverá incluir os seguintes elementos:
 - Nome das entidades envolvidas.
 - Designação do relacionamento.
 - Cardinalidade –1:1,1:N,N:M.
 - Participação – P:P, P:T, T:T.



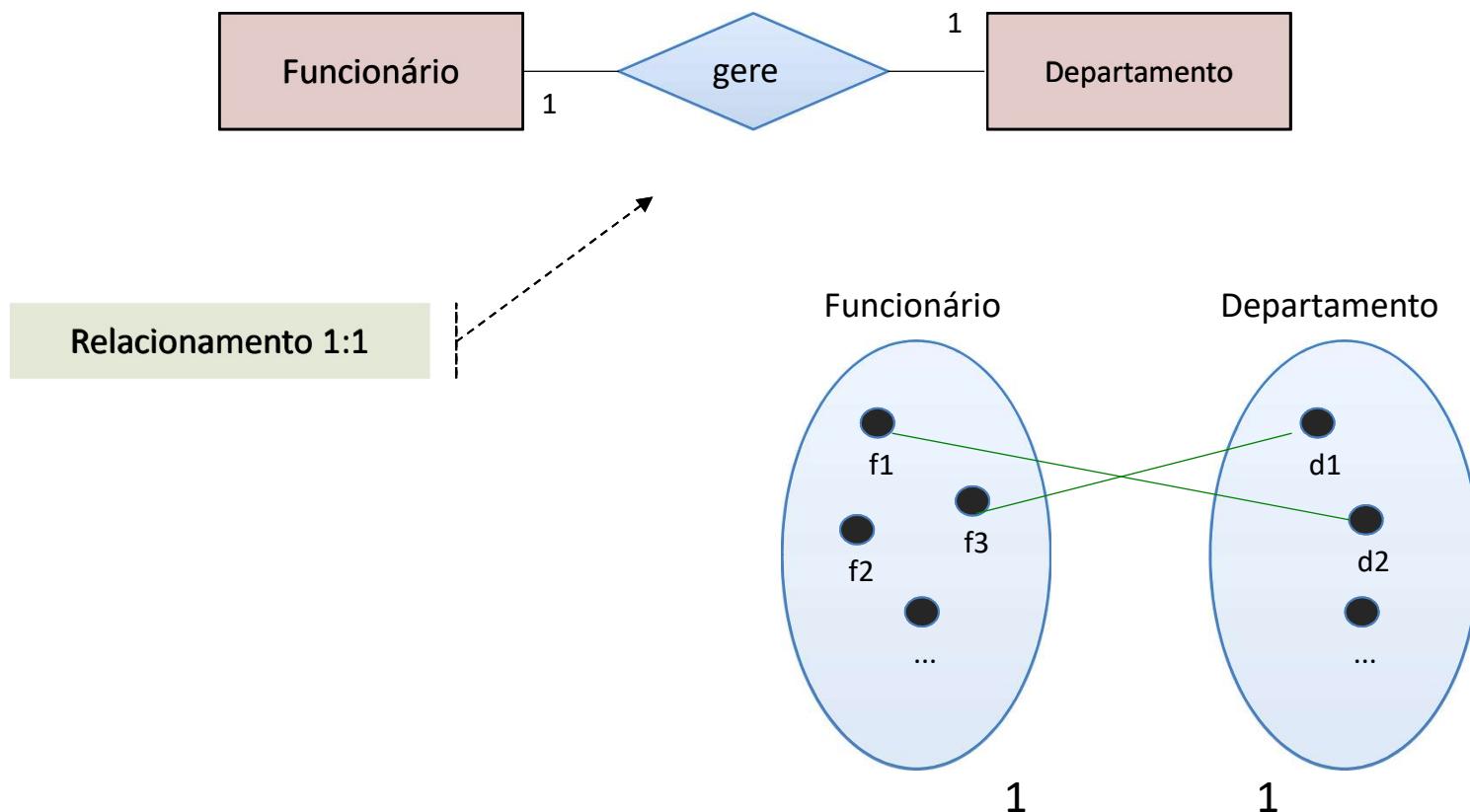
Exemplos de Relacionamentos



Exemplos de Relacionamentos



Exemplos de Relacionamentos



Associação de Atributos

- Esta etapa considera a **identificação dos tipos de factos** acerca das **entidades e relacionamentos** que foram escolhidos.
- Mais uma vez, a forma como se pode proceder para identificar os atributos pode ser através da análise da especificação dos requisitos dos utilizadores, tentando **identificar nomes, propriedades, ou características das entidades ou dos relacionamentos**.
- O processo de identificação de um atributo considera a **atribuição de uma designação** que deverá ser esclarecedora para o utilizador acerca do seu significado e conteúdo.



Atributos

- No processo de identificações de atributos, para caracterização de uma entidade ou de um relacionamento, podemos encontrar **diferentes tipos de atributos**, em particular:
 - **Simples** que são peças elementar de caracterização de uma entidade ou relacionamento.
 - **Compostos**, os atributos deste tipo são constituídos por vários atributos simples; por vezes, quando não existe a necessidade de aceder separadamente a cada um dos atributos simples que compõem um atributo composto, o atributo composto deverá ser considerado como um atributo simples.



Atributos

- E ainda:
 - **Derivados ou calculados**: são atributos cujo valor é encontrado com base nos valores de outros atributos; estes atributos nem sempre são apresentados nos modelos de dados, mas quando um ou mais dos atributos que lhe servem para o cálculo do seu valor podem ser modificados ou removidos é bom que eles se mantenham no modelo para evitar algum tipo de perda de informação.



Atributos

- Para fazer a **caracterização** de um atributo precisamos de registar a seguinte informação:
 - O nome e caracterização do atributo.
 - Designações alternativas ou sinónimos que o atributo possa ter.
 - Tipo de dados e comprimento.
 - Valores de omissão.
 - Em que situações o atributo poderá ter ou não valores nulos.
 - No caso de um atributo composto, quais são os atributos simples que o integram.
 - Quando o atributo é derivado, indicar a forma como é calculado.
 - Indicar quando um atributo é multivalor.



Domínios dos Atributos

- O domínio de um atributo é um conjunto de valores dos quais um ou mais atributos assumem os seus valores.
- Um modelo de dados completo deve apresentar a especificação dos domínios para todos os atributos e inclui normalmente:
 - O conjunto de valores permitidos para um atributo.
 - Tamanhos e formatos dos campos dos atributos.
- Tal como para os outros casos, todos os domínios definidos – designações e características - devem ser registados no dicionário de dados.



Atributos Chave

- Esta fase desenvolve-se em torno da **identificação das chave(s) candidata(s)** para uma entidade e posteriormente na seleção de uma chave primária.
- Uma **chave candidata** é um atributo, ou um mínimo conjunto, de uma entidade que identifica de forma única uma instância dessa entidade.
- Podemos identificar mais do que uma chave candidata, mas devemos escolher apenas uma **chave primária**. As restantes chaves candidatas são normalmente designadas por **chaves alternativas ou secundárias**.



Atributos Chave

- No processo de **escolha de uma chave primária** a partir de um conjunto de chaves candidatas podemos seguir algumas linhas de orientação, ou seja escolher a chave candidata:
 - com o menor conjunto de atributos.
 - com menos possibilidades de ter os seus valores modificados.
 - com menos probabilidade de perder a sua unicidade no futuro.
 - de menor tamanho.
 - mais “conveniente”, que seja mais fácil de utilizar.



Detalhar ou Generalizar Entidades

- O objectivo principal deste passo (opcional) é a identificação de tipos de **entidades super-classe ou sub-classe**.
- O conceito de especialização(detalhe) / generalização está associado normalmente com a modelação ER.
- No caso de seleccionarmos uma abordagem do tipo especialização tentaremos evidenciar as diferenças através da definição de uma ou mais subclasses para uma entidade. Este processo é designado por **especialização de super-classe**.



Detalhar ou Generalizar Entidades

- Porém, se selecionarmos uma abordagem do tipo generalização, tentaremos evidenciar as características comuns entre entidades por forma a definir uma entidade mais genérica. Este processo é designado por generalização de super-classe.
- Não existe um critério rígido e muito concreto acerca da forma como estes processos podem ser realizados. A escolha é muitas vezes subjetiva e dependente das características do problema que queremos modelar e do seu contexto.



Uma Regra Útil

- Uma espécie de regra útil, entenda-se “rule of thumb”, que podemos seguir no processo de especialização ou generalização diz-nos que devemos sempre tentar representar da forma mais clara possível num diagrama ER as entidades mais importantes, assim como os seus relacionamentos.
- Desta forma, o grau de especialização ou generalização representado num diagrama ER deverá ser orientado pela legibilidade do diagrama e pela clareza de modelação dessas entidades e relacionamentos.



Outra Regra Útil

- Na realidade o processo de modelação de dados não é complicado, mas por vezes surgem algumas dificuldades na abstração do problema o que torna o processo ambíguo, inconsistente ou com erros.
- A maioria dessa situações anómalas ocorrem pelo simples facto de se tentar representar situações complexas ou menos bem esclarecidas. Para que isso possa ser evitado (ou pelo menos atenuado) **devemos manter o processo tão simples quanto possível.**

Keep it simple. 

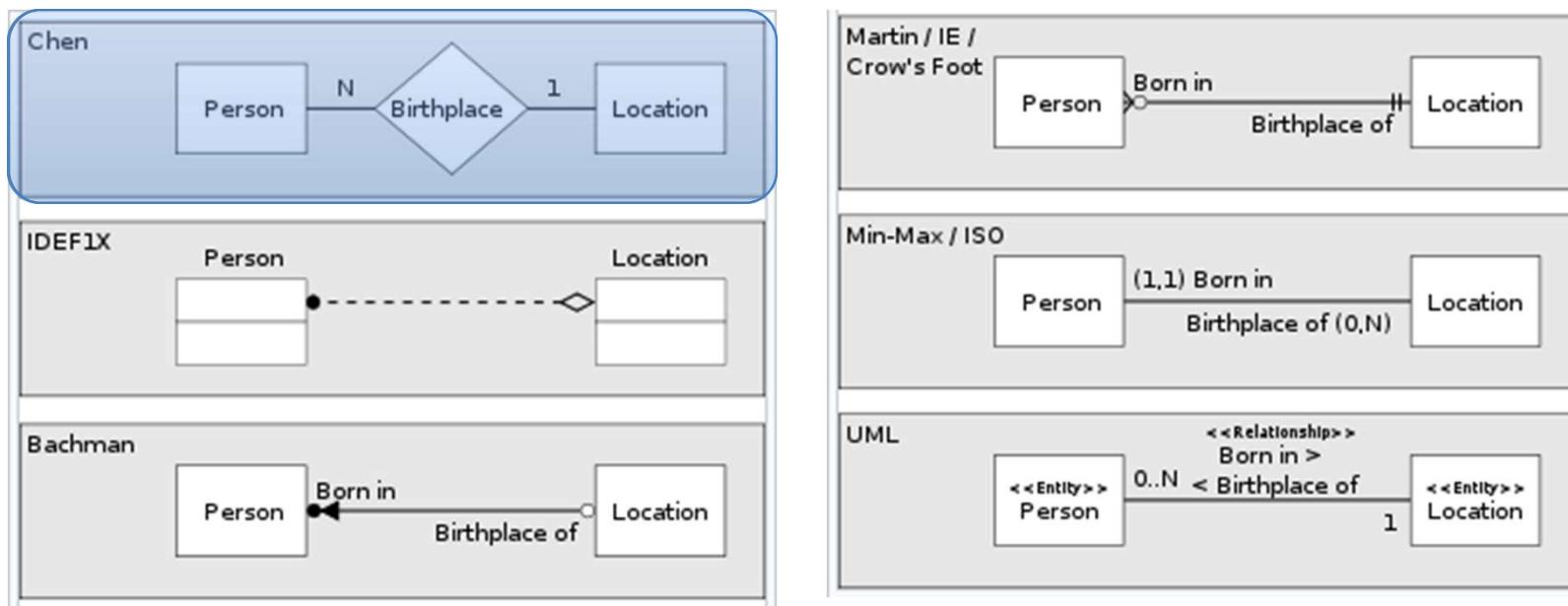


Desenhar o Diagrama ER

- O objectivo do desenho de um diagrama ER é essencialmente representar do ponto de vista conceptual a vista do utilizador sobre a organização.
- A construção de um diagrama ER pode ser suportado por várias notações. A generalidade das notações é muito semelhante. Apesar de puderem variar em termos de elementos gráficos, a sua semântica de representação é muito similar.



Notações Existentes

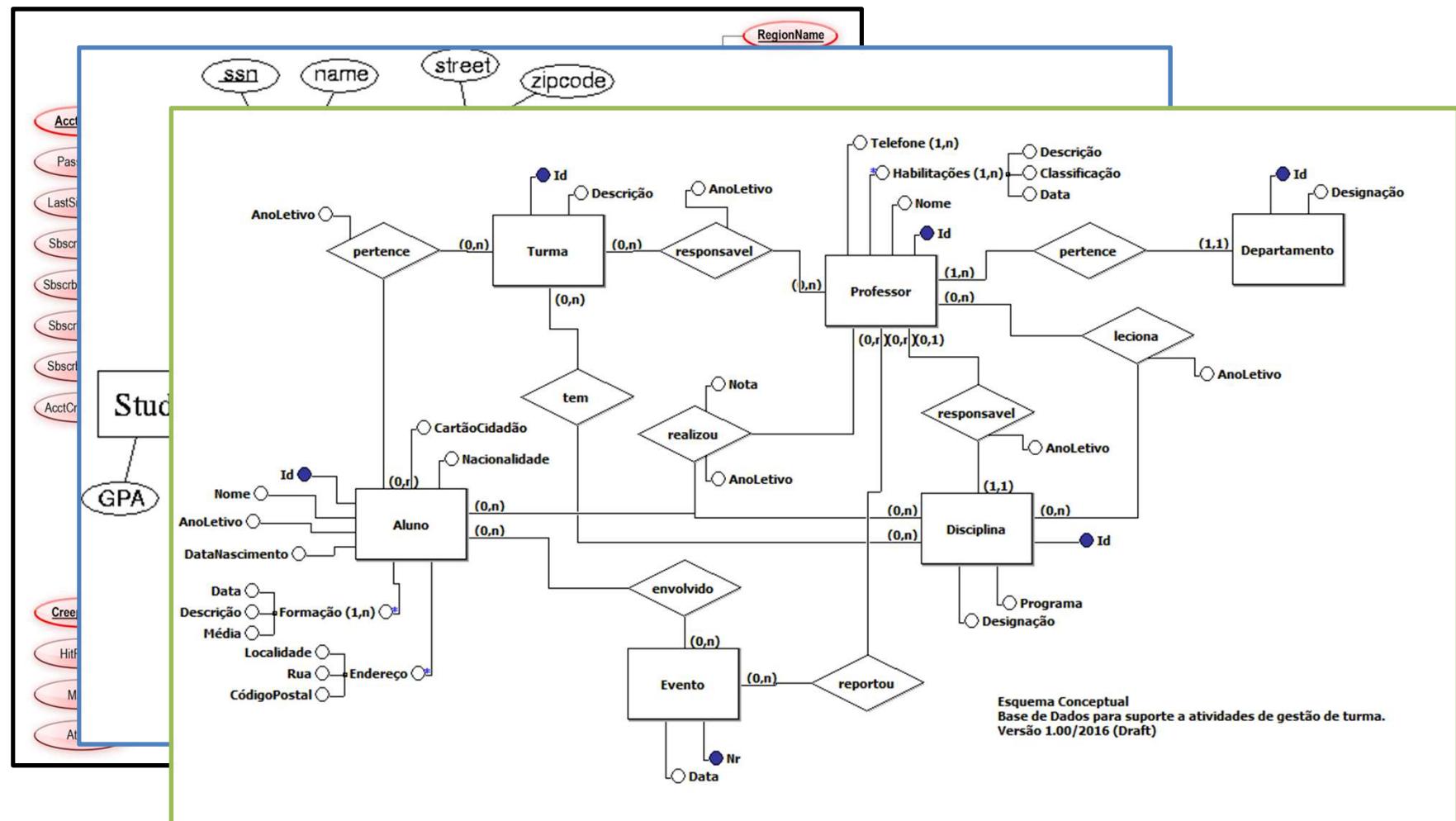


Obs.: Na generalidade dos modelos de dados que iremos desenvolver utilizaremos a notação CHEN

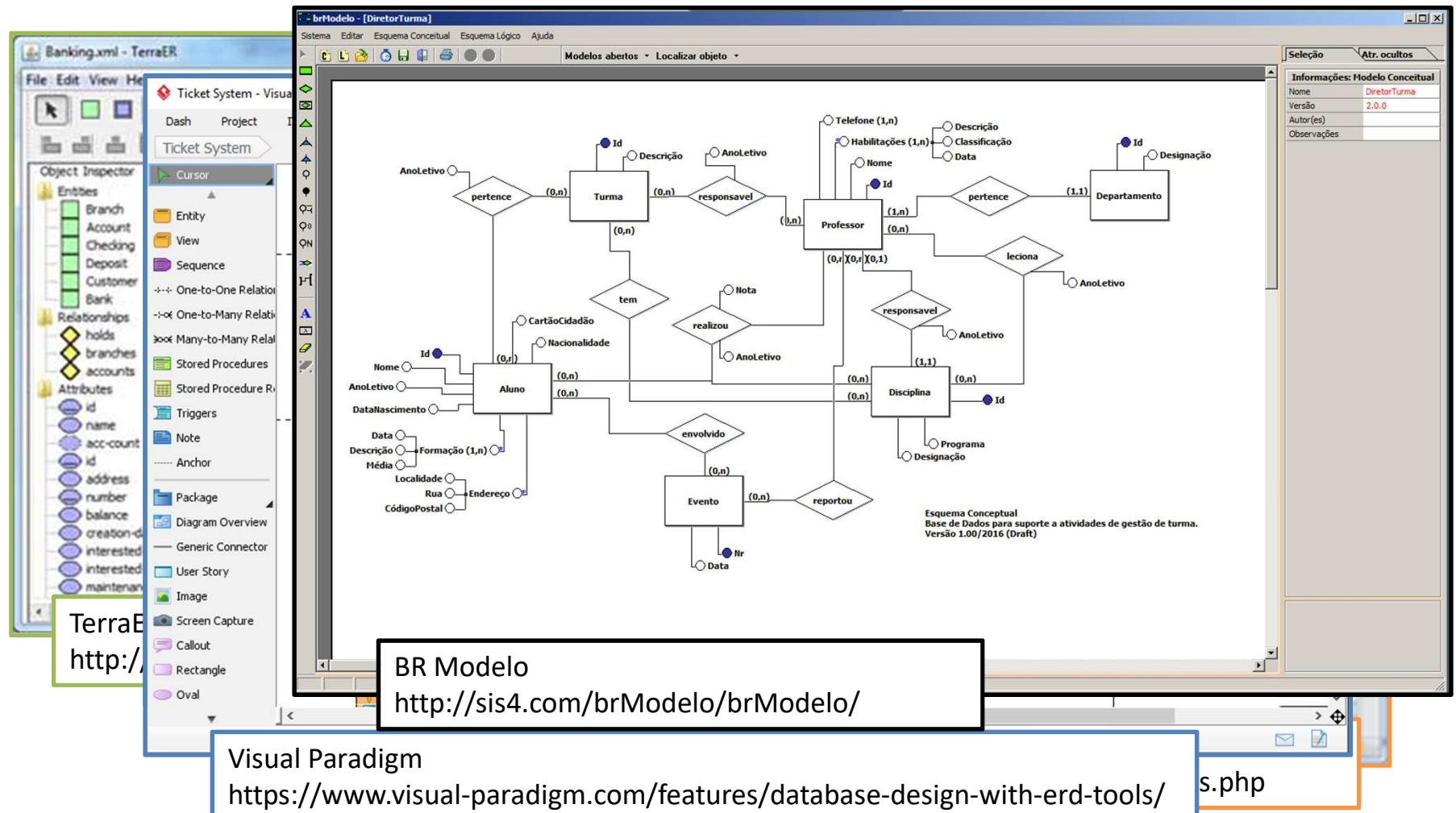
Fonte: https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model



Exemplos de Diagramas ER



Ferramentas de Modelação



Visual Paradigm
<https://www.visual-paradigm.com/features/database-design-with-erd-tools/>

s.php



61

Revisão do Modelo

- Esta é a última etapa da primeira fase de modelação e consiste na **revisão do modelo de dados** desenvolvido com o utilizador no sentido de assegurar que o modelo é uma representação verdadeira de vista do utilizador sobre a organização.
- O modelo de dados conceptual inclui diagramas ER e a respectiva documentação de suporte incluindo a sua descrição – o **dicionário de dados**.
- Se qualquer anomalia for detectada no modelo é necessário tomar as medidas necessárias para a sua correta adaptação à perspectiva de utilização em causa. É possível que seja necessário repetir alguns dos passos realizados anteriormente.



6

Desenho Lógico de uma Base de Dados (Fase IV)

- Caracterização da etapa.
- Fases do projeto lógico.
- Construção de modelos de dados lógicos locais.
- Construção de modelos de dados lógicos globais.



O Desenho Lógico

- Na segunda parte do projeto de uma base de dados procede-se ao desenvolvimento do seu **modelo de dados lógico**.
- O projeto lógico de uma base de dados é o processo de construção de um dado **modelo de informação** baseado num **modelo de dados específico**, mas que é independente de qualquer SGBD ou de outras considerações físicas.

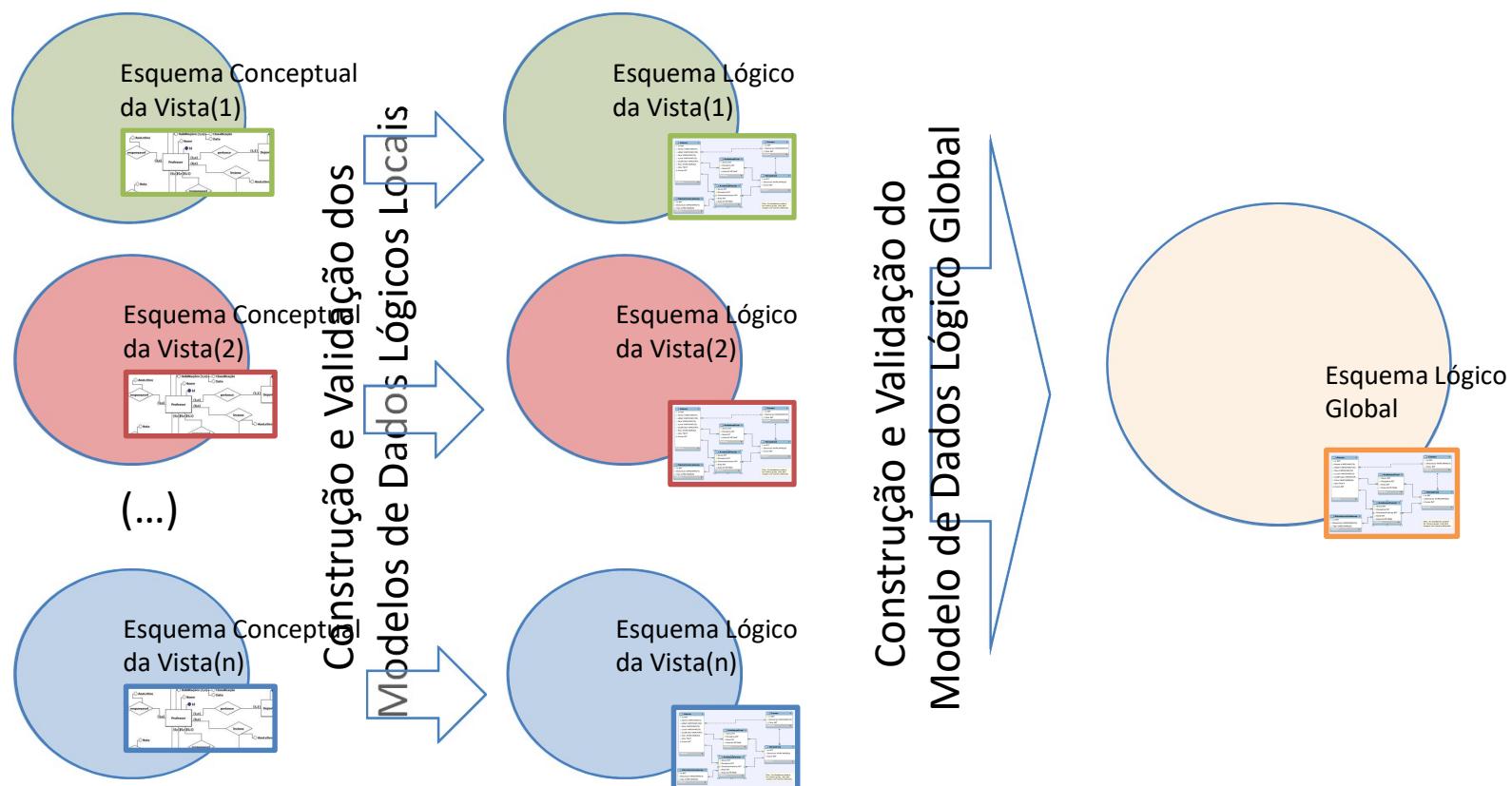


Etapas do Projeto Lógico

- As principais etapas do projeto lógico de uma base de dados são:
 - 1) Construção e validação do **modelo de dados lógico local** (ou parcial) correspondente a cada vista de utilizador.
 - 2) Construção e validação do **modelo de dados lógico global**.



Etapas do Projeto Lógico



Construção dos Modelos de Dados Lógicos Locais

- Primeiro, fazemos a conversão dos modelos conceptuais locais para os correspondentes modelos lógicos, realizando as seguintes tarefas:
 - 1) Derivação de relações do modelo de dados local.
 - 2) Validação do modelo através da normalização.
 - 3) Validação do modelo com as transações do utilizador.
 - 4) Desenho do modelo lógico.
 - 5) Definição de restrições de integridade.
 - 6) Revisão do modelo de dados local com o utilizador.



Construção do Modelo de Dados Lógico Global

- De seguida, após obtenção dos modelos lógicos locais, procedemos à sua junção para obter um único modelo lógico global. Para isso, devemos realizar as seguintes tarefas:
 - 1) Validar o modelo de dados global.
 - 2) Analisar o crescimento da base de dados no futuro.
 - 3) Desenhar a versão final do diagrama do esquema lógico.
 - 4) Revisão do modelo global final com os utilizadores.



Conversão do Modelo Conceptual num Modelo Lógico

- O objectivo principal desta etapa é o refinamento do modelo de dados conceptual local no sentido de remover as suas características indesejáveis e fazer o seu mapeamento para um modelo de dados lógico.
- Nesta altura do projeto tem-se um modelo de dados conceptual para cada uma dada vista de utilização de um utilizador, no caso de termos optado por uma abordagem multivista, obviamente.



Conversão do Modelo Conceptual num Modelo Lógico

- É normal que os modelos produzidos contenham algumas estruturas de dados que não sejam facilmente modeladas pelos SGBD convencionais - é nesta etapa que se transformam estas estruturas para uma forma de mais fácil manuseio para esses sistemas: **estruturas que possam ser implementadas computacionalmente.**
- Atualmente, existem muitas formas de representar e implementar tais estruturas. A mais usual (e conhecida) é através da utilização do **modelo de dados relacional**. Este será o modelo com que iremos trabalhar.

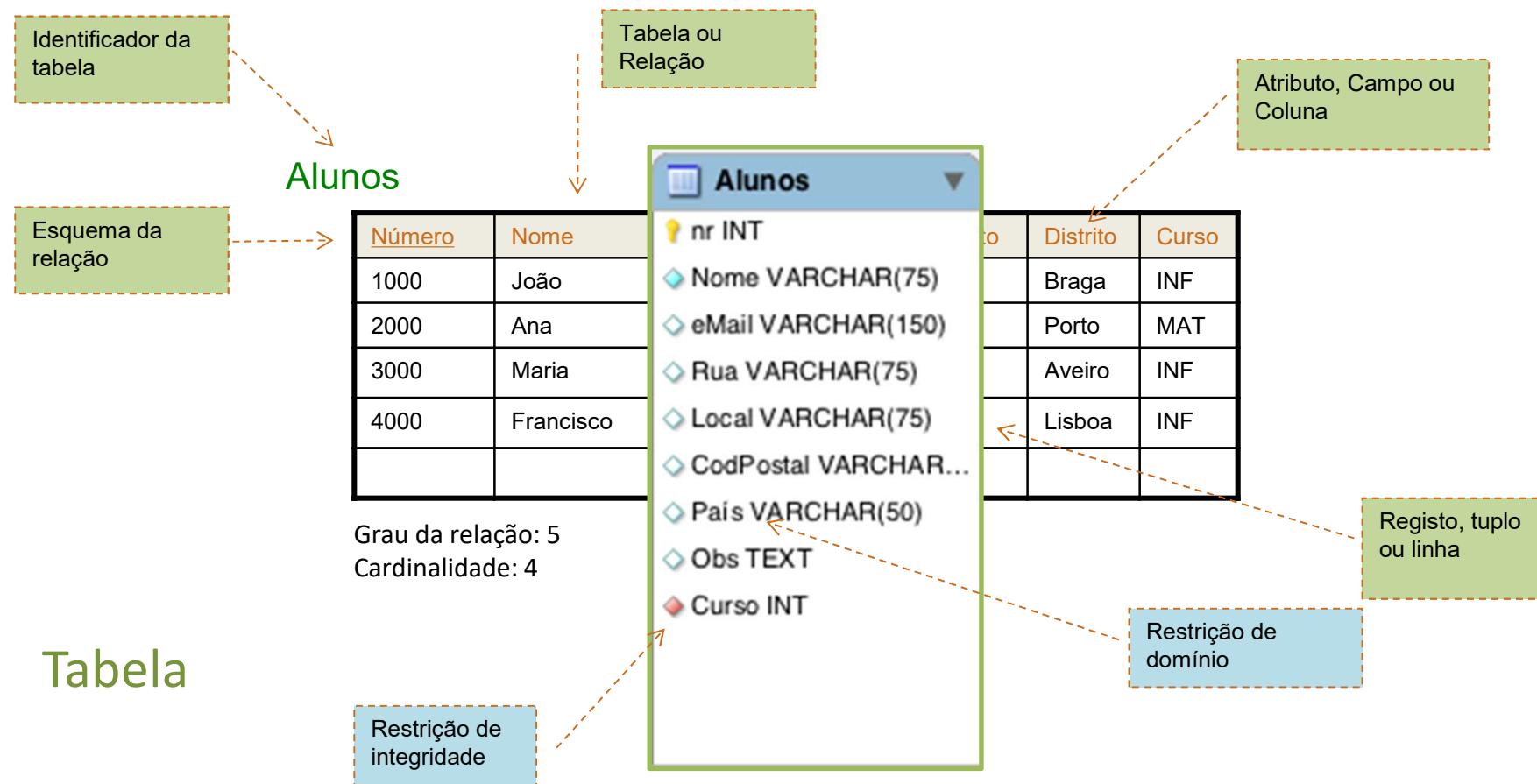


O Modelo de Dados Relacional

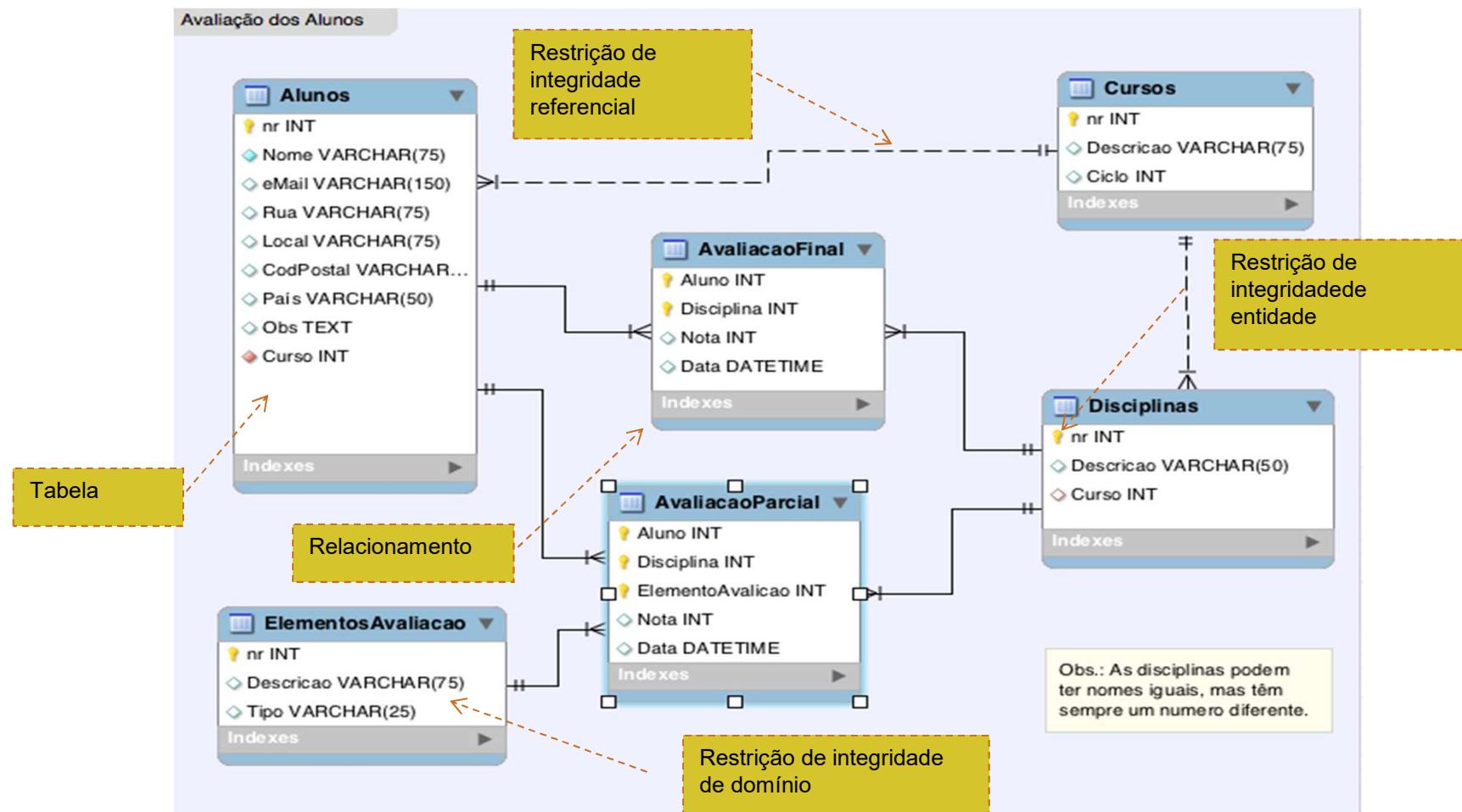
- O **modelo de dados relacional** foi proposto por Codd (IBM Research Fellow) em 1970 (Codd, 1970), no qual os dados são representados em tuplos, que por sua vez estão agrupados em relações.
- Uma base de dados assente no modelo relacional de dados designa-se por **base de dados relacional**.



O Elemento de Dados



Um Esquema Relacional



Conceitos Relevantes

- Base de dados relacional
- Tabela ou relação.
- Atributos, campos ou colunas.
- Esquema de uma tabela.
- Registros, tuplos ou linhas.
- Grau e cardinalidade de uma tabela.
- Chave candidata, primária, secundária, alternada, estrangeira e super-chave.
- Restrições de integridade, de identidade, de domínio, de valor e referencial.



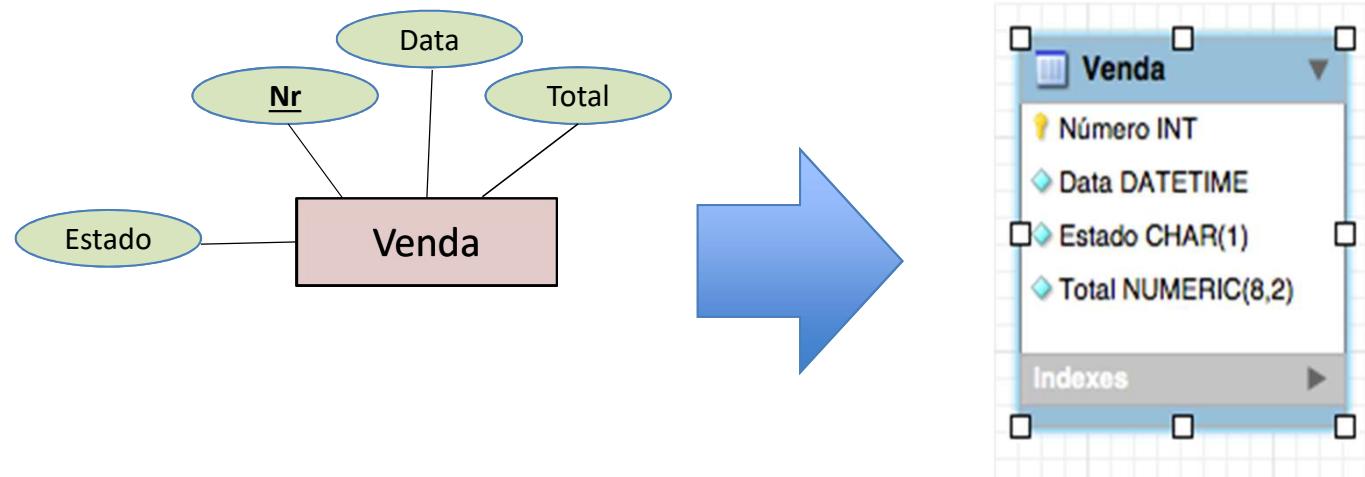
Mapeamento Conceptual-Lógico

- No mapeamento do esquema conceptual para o esquema lógico começamos por realizar a operação mais elementar que, basicamente, é fazer a **conversão das entidades** em tabelas base.
- Todas as **entidades base** num modelo conceptual dão origem a **uma tabela** no correspondente modelo lógico. Porém, nem sempre este mapeamento é tão simples e direto.
- Veja-se com atenção ao caso dos **atributos multivalor de uma entidade** e a forma como se faz o seu mapeamento para um modelo lógico.



Mapeamento de Entidades

Exemplo 1 – Entidade com atributos simples e compostos.

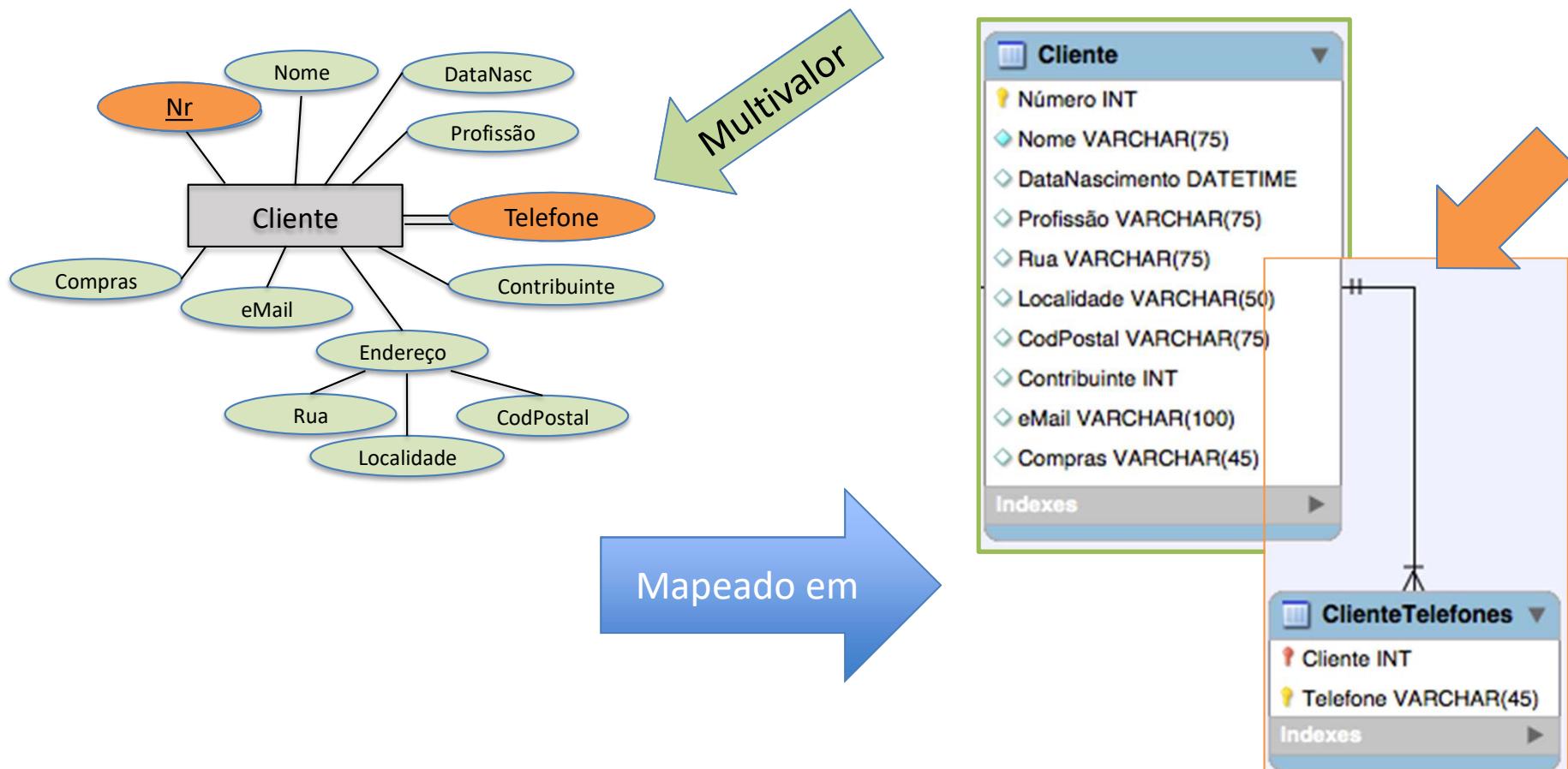


Atributos Multivalor

- Um **atributo multivalor** acolhe vários (múltiplos) valores para uma única entidade.
- Se um atributo multivalor estiver representado no modelo de dados, devemos fazer a decomposição no sentido de identificar uma entidade correspondente.



Mapeamento de Entidades



Mapeamento Conceptual-Lógico

- Após o mapeamento das entidades passamos ao **mapeamento dos relacionamentos** definidos no esquema conceptual.
- O mapeamento dos relacionamentos faz-se de acordo com um conjunto de regras bem definido, que nos indicam a forma como transformamos cada um dos tipos de relacionamentos nos correspondentes objetos de dados relacionais.



Mapeamento Conceptual-Lógico

- Os relacionamentos definidos no modelo conceptual são mapeados através da realização de operações de transformação estabelecidas de acordo com a sua cardinalidade e atributos associados. De referir:
 - “Remover” relacionamentos M:N, com ou sem atributos.
 - “Remover” relacionamentos complexos.
 - “Remover” relacionamentos recursivos.



Mapeamento Conceptual-Lógico

- Depois do mapeamento dos relacionamentos entramos na fase final de transformação, na qual devemos ver como:
 - Reexaminar os relacionamentos 1:1.
 - Remover relacionamentos redundantes.

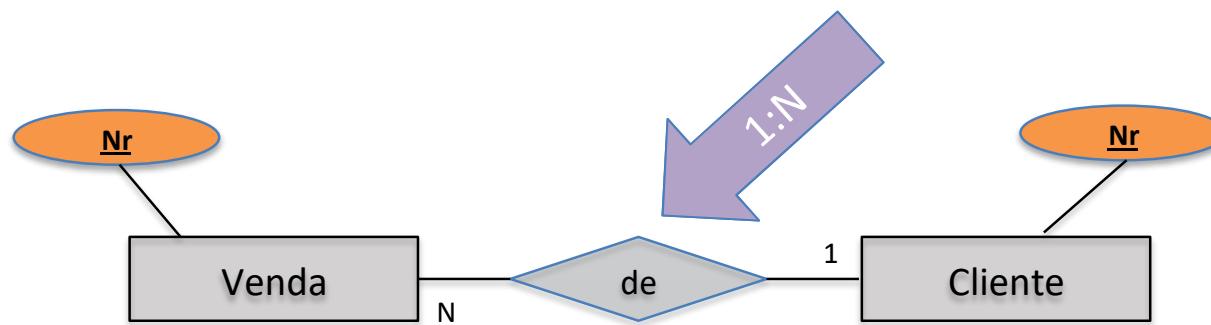


Relacionamentos 1:N

- O **relacionamento 1:N** é aquele que mais encontramos em modelos conceptuais de bases de dados.
- O mapeamento deste tipo de relacionamento para um esquema lógico também dá origem a um novo objeto de dados. Porém, este objeto não é uma nova tabela. É apenas um atributo, que emerge na tabela correspondente à entidade cuja cardinalidade associada do relacionamento é de muitos (N).
- O atributo que sustenta o relacionamento é designado por chave estrangeira ou externa.



Relacionamentos 1:N



Mapeado em

Database schema diagram showing the **Venda** table:

Venda	
?	Número INT
?	Data DATETIME
?	Estado CHAR
?	Total NUMERIC
?	Cliente INT

Indexes

Chave estrangeira



Relacionamentos M:N

- Se no modelo conceptual estiver representado um relacionamento M:N (muitos-para-muitos) dever-se-á decompô-lo no sentido de identificar uma “entidade” intermédia. O relacionamento M:N será substituído por dois relacionamentos 1:N para a nova “entidade”.

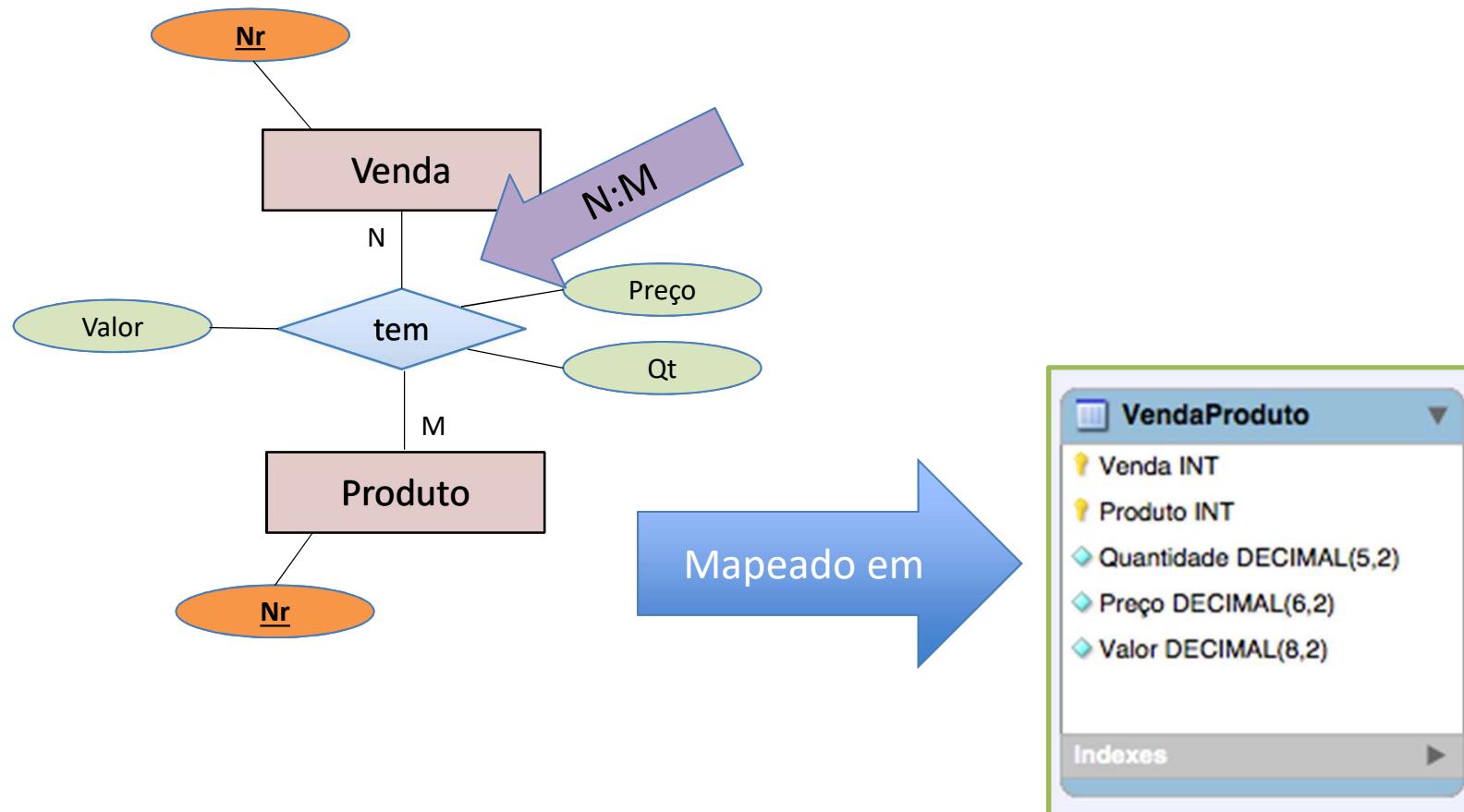


Relacionamentos com Atributos

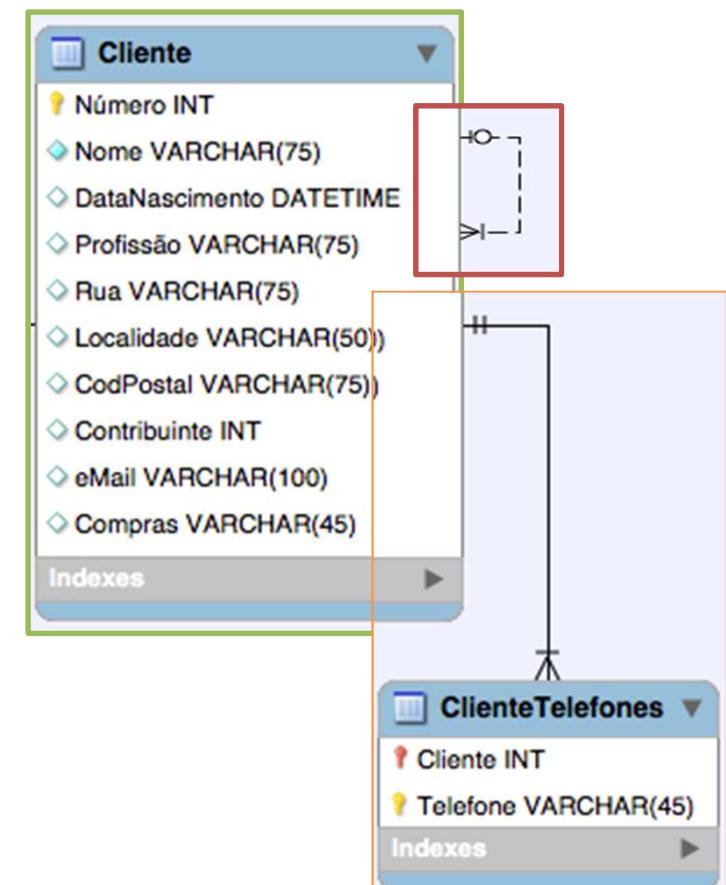
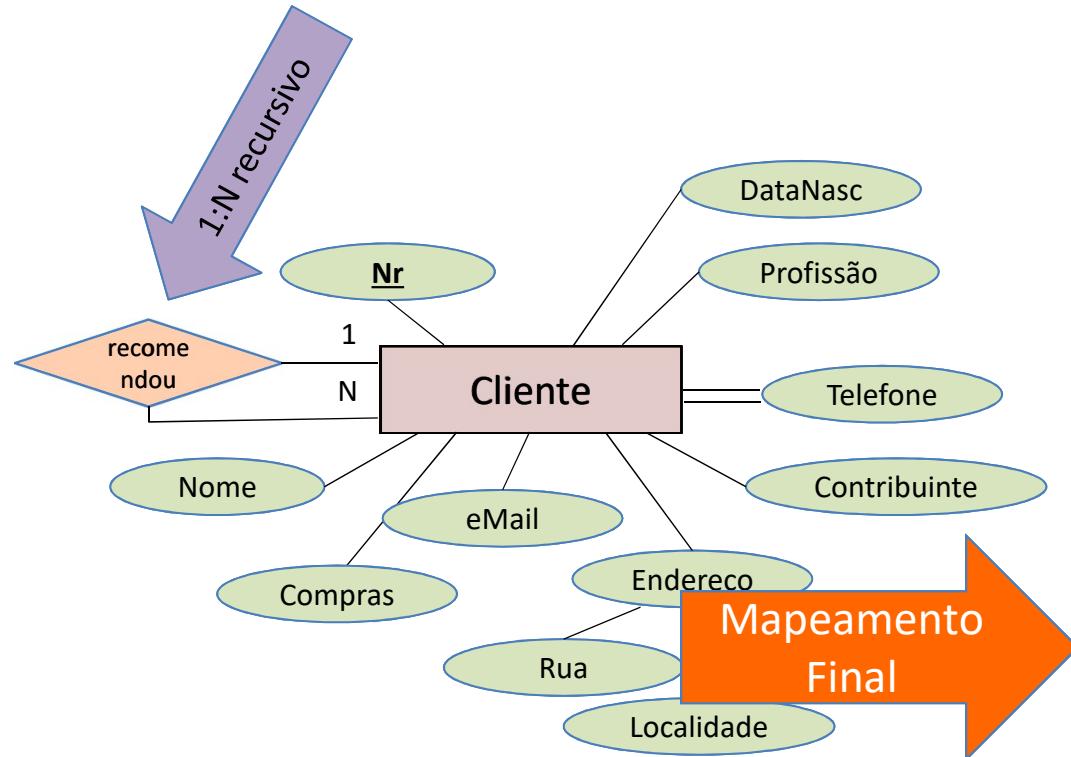
- Se num modelo conceptual existirem **relacionamentos com atributos**, então dever-se-á fazer a decomposição do relacionamento para identificar uma “identidade” intermédia.
- A decomposição num relacionamento N:M originará uma “entidade” fraca e dois relacionamentos 1:N.



Relacionamentos M:N



Relacionamentos Recursivos

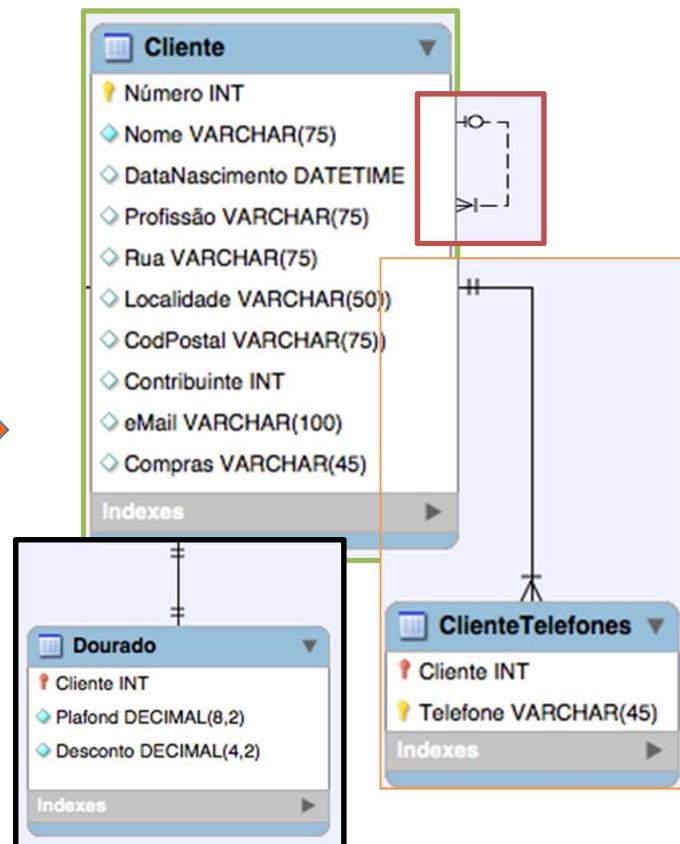
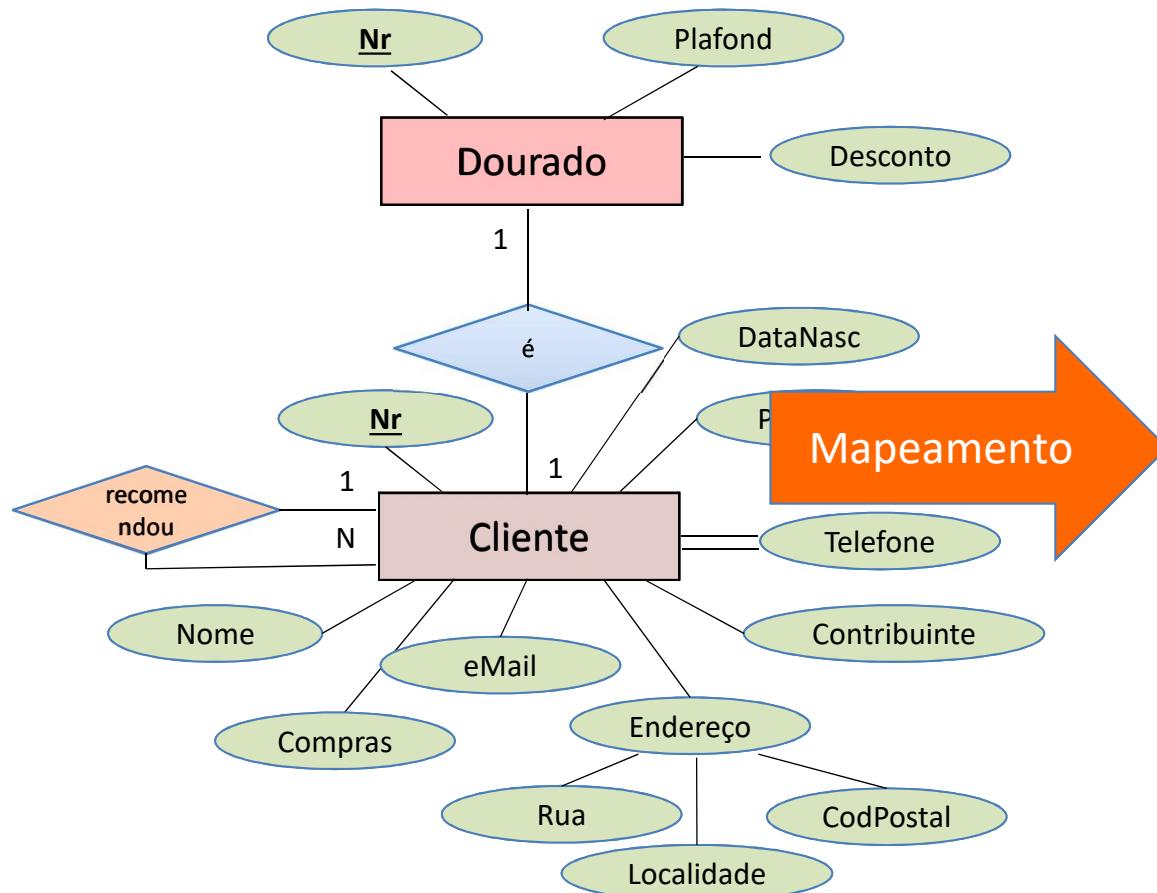


Reexaminar os Relacionamentos 1:1

- Na identificação de entidades podemos identificar por vezes duas entidades que representam o mesmo objecto no modelo de dados de uma organização.
- Quando suceder uma destas situações, as duas entidades devem ser transformadas numa única entidade.
- Se as chaves primárias forem diferentes, dever-se-á escolher uma delas para ser a chave primária ficando a outra como chave alternativa.
- Porém, apesar de serem relacionamento muito particulares e de difícil descoberta, os relacionamento binários 1:1 existem de facto.



Relacionamentos 1:1

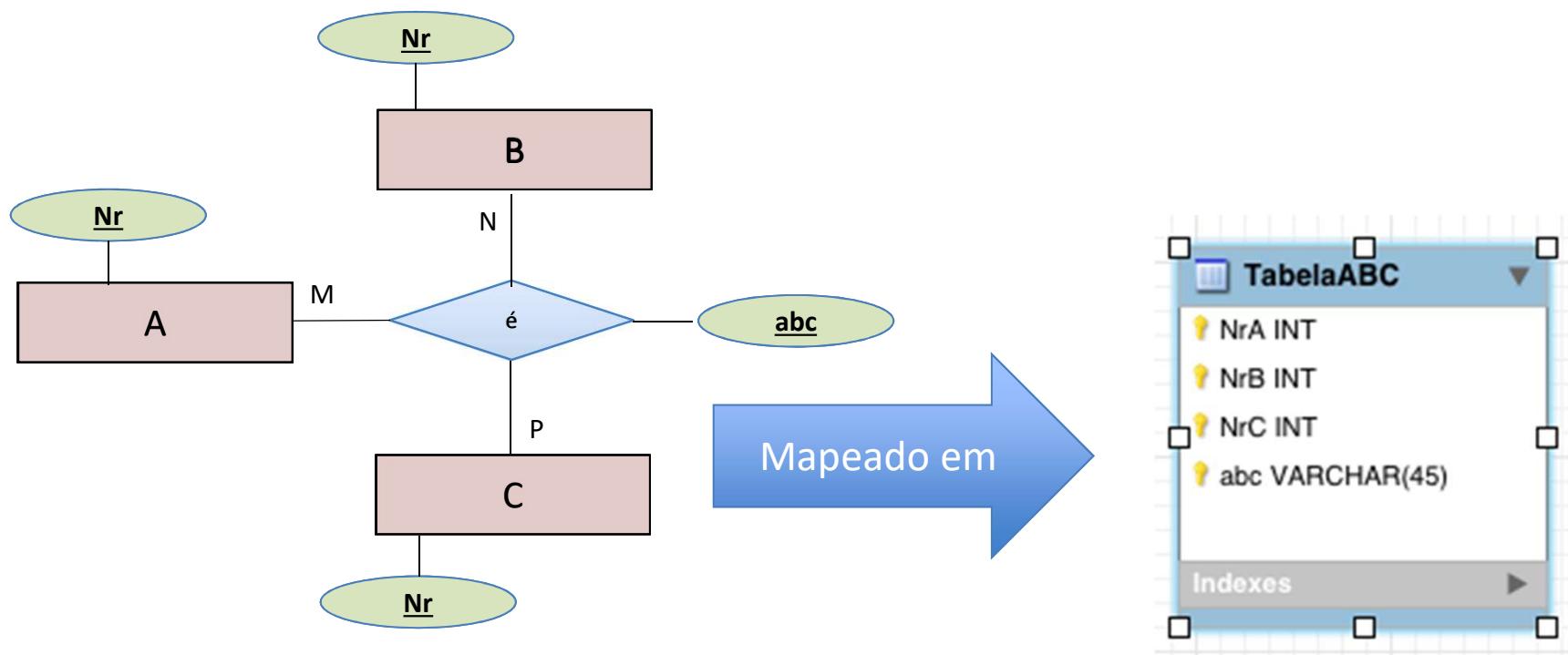


Relacionamentos Complexos

- Um **relacionamento complexo** é aquele que está definido entre três ou mais entidades, não sendo objetos com grande frequência num esquema, mas ocorrem em muitos casos de aplicação de bases de dados.
- Se um relacionamento complexo estiver representado num modelo conceptual, dever-se-á decompô-lo para se identificar uma “entidade” intermédia e estabelecendo tantos relacionamentos binários (1:N) quantos os requeridos por essa nova “entidade”.



Relacionamentos Complexos



Relacionamentos Redundantes

- Um **relacionamento é redundante** se a informação que acolhe pode ser também obtida através de outro(s) relacionamento(s).
- Deve-se conceber e implementar um modelo mínimo. Como tal, os relacionamentos redundantes são desnecessários e devem ser removidos.



Validação do Modelo

- A **normalização** é utilizada para melhorar o modelo de dados já que a aplicação das suas regras permite evitar a duplicação de dados - o desenho lógico pode não ser o desenho final.
- A normalização assegura que o modelo resultante é um modelo mais próximo da organização que representa, mais consistente, com um baixo grau de redundância e grande estabilidade.
- Por vezes, discute-se que um esquema normalizado não contribui para a máxima eficiência de processamento.



Validação do Modelo

- A elaboração de um **esquema normalizado** pode ser **justificado** através das seguintes razões:
 - Um desenho normalizado organiza os dados de acordo com as suas dependências funcionais.
 - Um desenho normalizado é robusto e está livre de anomalias provocadas por modificações de dados.
 - Dada a atual situação dos computadores, por vezes é razoável implementar um desenho que ganha em termos de facilidade de utilização com um custo de processamento adicional.



Validação do Modelo

- E ainda:
 - A normalização força-nos a entender cada um dos atributos das relações representadas na base de dados.
 - A normalização produz um desenho da base de dados flexível, que pode facilmente ser estendido.



Análise de Interrogações

- A análise de interrogações é um passo bastante importante, uma vez que se destina a verificar se as principais (todas) ações de exploração da base de dados , realizadas por queries do sistema ou das aplicações cliente - podem ou não ser satisfeitas.
- Usualmente procede-se à elaboração de um mapa de interrogações, cujo objetivo é mapear as queries sobre o esquema da base de dados. Este exercício ajuda a verificar quais serão os objetos de dados (as tabelas) que serão mais “fustigadas” pelas queries e com que frequência.



Análise de Interrogações

- Esta análise permite-nos identificar, caso existam, algumas fragilidades da base de dados, como, por exemplo:
 - potenciais pontos de estrangulamento.
 - queries menos bem projetadas.
 - necessidade de criar ou eliminar atributos derivados.
 - criação de índices.
 - (...)
- Posteriormente, caso ainda necessitem de ser tratados, todos estes aspetos poderão ser mais bem analisados através da análise dos **planos de execução** que alguns SGBD disponibilizam.



Exemplos de Interrogações

- Para o caso de estudo da “Mercearia da D. Acácia” podemos assumir a necessidade de interrogações como as seguintes:
 - Quais são os meus clientes?
 - Quais foram as vendas realizadas com os clientes ‘João Maria Castro’ e ‘Joana Manuel Pires’?
 - Quanto é foi vendido (em valor) durante o último mês?
 - Quais são os meus melhores clientes?
 - (...)
- Como é que poderemos ver como é que o esquema da base de dados é “afetado” por estas queries?



Análise de Transações

- O objectivo deste passo é assegurar que o modelo desenvolvido suporta as transações requeridas pela vista de utilização, definidas com base na especificação dos requisitos dos utilizadores.
- Usualmente procede-se à elaboração de um mapa de transações.



Desenho Final do Esquema Lógico

- Esta etapa considera o desenho final do esquema lógico que é uma representação lógica local de uma dada vista de utilizador acerca da organização.
- O diagrama final terá que ser validado usando as técnicas de normalização e verificando as transações que ele poderá vir a suportar.



Assegurar a Integridade Referencial

- A **aplicação de restrições de integridade** são um dos aspectos mais importantes na garantia da consistência de uma base de dados. Porém, como é que se pode assegurar a integridade referencial?
- Para contribuir para essa garantia, pode-se definir um conjunto de condições – **restrições de existência** - sob as quais uma chave candidata ou uma chave estrangeira poderá ser inserida, modificada ou apagada.



Assegurar a Integridade Referencial

- Consideremos, por exemplo, um relacionamento entre as relações “Alunos” e “Cursos” e analisemos os seguintes casos de manipulação de dados com base nesse relacionamento:
 - Inserção de um registo na relação filho.
 - Remoção de um registo da relação filho.
 - Modificação do valor da chave estrangeira num registo da relação filho.
 - Inserção de um registo na relação pai.
 - Remoção de um registo da relação pai.
 - Modificação do valor de uma chave primária de um registo da relação pai.



Tipos de Restrições de Integridade

- Podemos considerar os seguintes tipos de restrições de integridade:
 - Exigência de dados: alguns atributos devem conter sempre um valor válido, isto é não devem aceitar valores nulos.
 - 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.



Tipos de Restrições de Integridade

- Restrições de domínios de atributos: qualquer atributo tem um domínio, ou seja um conjunto de valores dos quais o atributo poderá assumir um; estas restrições devem ser identificados no momento da escolha dos domínios.



Tipos de Restrições de Integridade

- **Integridade referencial.** Uma chave estrangeira liga cada ocorrência na relação filho com a respectiva ocorrência na relação pai que contém o valor da chave candidata correspondente. A integridade referencial significa que se uma chave estrangeira contém um valor, esse valor deve existir na relação pai para que seja aceite na chave estrangeira. Os valores nulos devem ser analisados com cuidado, sendo sua utilização dependente do tipo de participação – **total ou parcial** – da chave estrangeira no relacionamento.



Tipos de Restrições de Integridade

- Restrições da organização. Estas restrições, também conhecidas por regras de negócio, são normalmente regras da própria organização que são aplicadas para regulamentar as suas próprias atividades.



Revisão do Modelo

- No momento da realização desta etapa, o modelo de dados já deverá estar completo e documentado.
- Por fim, é necessário rever o modelo desenvolvido e a documentação gerada com os futuros utilizadores da base de dados em construção.



Fusão dos Modelos Locais

- Num pequeno sistema, com apenas **duas ou três vidas de utilização** e um pequeno número de entidades e relacionamentos, **não deverá ser difícil comparar os modelos locais e fazer a sua fusão num único modelo global**, resolvendo eventuais incompatibilidades entre eles.
- Porém num sistema de grandes dimensões e complexo **é necessário abordar o problema da fusão de esquemas de uma forma mais sistemática**.



Fusão dos Modelos Locais

- Para fazermos a fusão de esquemas locais podemos seguir a seguinte abordagem:
 - 1) Rever o nome das entidades e das suas chaves primárias.
 - 2) Rever os nomes dos relacionamentos.
 - 3) Fundir as entidades das vistas locais.
 - 4) Incluir, sem fundir, as entidades não partilhadas por cada vista local no global.
 - 5) Fundir os relacionamentos de cada esquema local.
 - 6) Incluir, sem fundir, os relacionamentos não partilhados por cada vista local no global.



Fusão dos Modelos Locais

- 7) Detectar eventuais falhas de integração de entidades e relacionamentos.
- 8) Verificar as chaves estrangeiras.
- 9) Verificar as restrições de integridade.
- 10) Desenvolver o modelo de dados lógico global.
- 11) Atualizar a documentação.



Validar o Modelo de Dados Global

- O objectivo desta etapa é a validação do modelo global definido através da sua verificação com base em técnicas de normalização e analisar sobre ele o efeito das transações requeridas.



Analisar a Evolução da Base de Dados

- É importante que o modelo de dados lógico global alcançado possa ser **facilmente expandido**.
 - Se o modelo apenas for capaz de suportar os atuais requisitos operacionais, então é provável que a sua vida seja relativamente curta e que no caso de ser necessário acrescentar novos requisitos ao modelo seja necessário algum trabalho.
- Assim, nesta fase, espera-se **determinar quando é que no futuro poderão ocorrer modificações e saber estabelecer o momento quando é que o modelo de dados poderá acolher tais modificações** com um mínimo de impacto possível.



Alguns Aspetos a ter em Conta

- No cálculo do tamanho de uma base de dados, bem como na análise da sua evolução (crescimento) ao longo do tempo, devemos ter em alguns aspetos, nomeadamente:
 - cada base de dados pode utilizar um **formato diferente no armazenamento dos seus dados**, o que faz com que tanto o seu tamanho inicial como o seu conhecimento seja diferente de formato para formato.
 - Para cada tabela, de acordo com o formato da base de dados escolhida, calcular **o tamanho de cada um dos seus registos**.



Alguns Aspetos a ter em Conta

- E ainda:
 - Para que seja possível fazer o cálculo do tamanho da base de dados é necessário **fazer algumas assunções**, por exemplo o número de registos a carregar em cada uma das tabelas da base de dados para que esta possa começar a operar ou, para um dado período (dia, semana, mês, ou ano), qual o número de registos novos a inserir.
 - O tamanho de uma tabela não depende apenas da sua estrutura base, mas também dos **índices** (e dos seus tipos) que tiver associados.



Outros Aspetos

- Estimar o tamanho de uma base de dados pode ajudar a determinar se o seu desenho precisa de ser **refinado**. Vejamos porquê;
 - se o tamanho estimado para a base de dados for (muito) grande para ser implementado então provavelmente precisamos de **normalizar** mais a base de dados, mas
 - se o tamanho estimado for pequeno, então talvez seja bom aplicar algumas técnicas de **desnormalização** para melhorar o desempenho da base de dados.



A Versão Final do Modelo Lógico

- Nesta fase temos o modelo de dados lógico global completo, por isso podemos **desenhar o diagrama lógico final**, que representará o modelo de dados lógico global da parte da organização que se pretendeu modelar.
- Todos os documentos que descrevem o modelo, incluindo o **esquema relacional e o dicionário de dados devem ser** também atualizados nesta fase.



Revisão do Modelo Global

- Nesta altura o modelo de dados lógico global deverá estar **completo e preciso**.
- O modelo de dados lógico e toda a documentação que o descreve **deverão ser revistos com os utilizadores** por forma a assegurar que o modelo é uma representação verdadeira da parte da organização que se pretendeu modelar.



Referências

- Connolly, T., Begg, C., **Database Systems: A Practical Approach to Design, Implementation, and Management**, 6th edition, Pearson, January, 2014.
- Codd, E.F., **A relational model of data for large shared data banks**. In Communications of the ACM, Vol. 13 Issue 6, pp 377-387, June 1970.
- Teorey, T., Lightstone, S., Nadeau, T., Jagadish, H.V., **Database Modeling and Design**, 4th Edition, Morgan Kaufmann, September, 2005.
- Teorey, T., **Database Modeling and Design: The Fundamental Principles**, II Edição, Morgan Kaufmann, 1994.
- Date, C., **An Introduction to Database Systems**, 8th Edition, Pearson, July, 2003.



Referências

- Garcia-Molina, H., Ullman, J., Widom, J., **Database Systems: The Complete Book**, 2nd Edition, Pearson, June, 2008.
- Chen, P., **The entity-relationship model - Toward a unified view of data**. ACM Trans. Database Syst. 1, 1, 9-36, March 1976.



Referências Web

- <https://www.c-sharpcorner.com/article/requirement-gathering-and-creating-initial-er-model/>
- https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model
- http://www.inf.fu-berlin.de/lehre/SS06/19513-V/Folien-etc/dbs06_02_conceptDesign-1pp.pdf
- <https://www.relationaldbdesign.com/database-design/module4/intro-requirements-analysis.php>

