

MBA EM **ENGENHARIA**
DE SOFTWARE

Bancos de Dados Relacionais I

Profa. Dra. Kelly Rosa Braghetto

MBA USP
ESALQ

A responsabilidade pela idoneidade, originalidade
e licitude dos conteúdos didáticos apresentados é do professor.

Proibida a reprodução, total ou parcial, sem autorização.

Lei nº 9610/98

MBA USP
ESALQ

Introdução aos Sistemas de Bancos de Dados

Slides baseados no livro *Sistemas de Bancos de Dados* (7^a edição),
de Elmasri e Navathe. Pearson, 2010 – Capítulos 1 e 2

O que é um banco de dados?

- ◆ **Banco de dados** (BD) – coleção de dados relacionados
- ◆ **Dados** – fatos conhecidos que podem ser registrados e que possuem significado implícito
- ◆ Propriedades implícitas de um (BD):
 - 1) Representar um aspecto do mundo real (= minimundo)
 - 2) Ser uma coleção lógica e coerente de dados com algum significado inerente.
Uma coleção “aleatória” de dados não é um BD!
 - 3) Ser projetado, construído e povoado com dados que possuem um objetivo específico.

Um BD deve possuir um grupo de usuários em potencial e algumas aplicações pré-concebidas, nas quais esses usuários estão interessados.

Softwares para a Manutenção de Bancos de Dados

Um dado BD informatizado pode ser criado e mantido por:

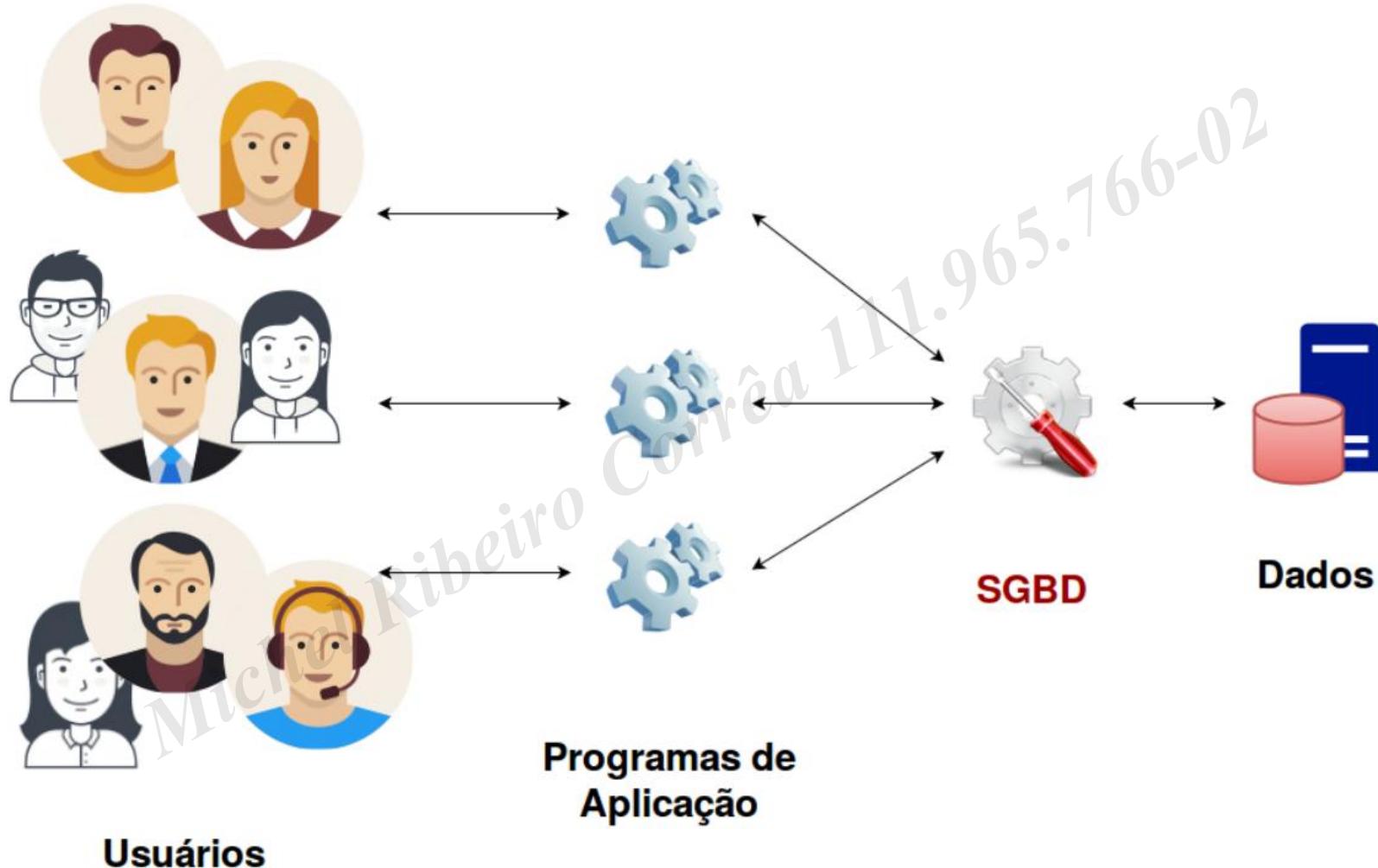
- ◆ Um programa de aplicação desenvolvido especificamente para essa tarefa ou
- ◆ Um **Sistema de Gerenciamento de Banco de Dados (SGBD)**:
 - Sistema de software de propósito geral que facilita o processo de definição, construção, manipulação e compartilhamento de BDs entre vários usuários e aplicações

Sistema de Gerenciamento de Bancos de Dados (SGBD)

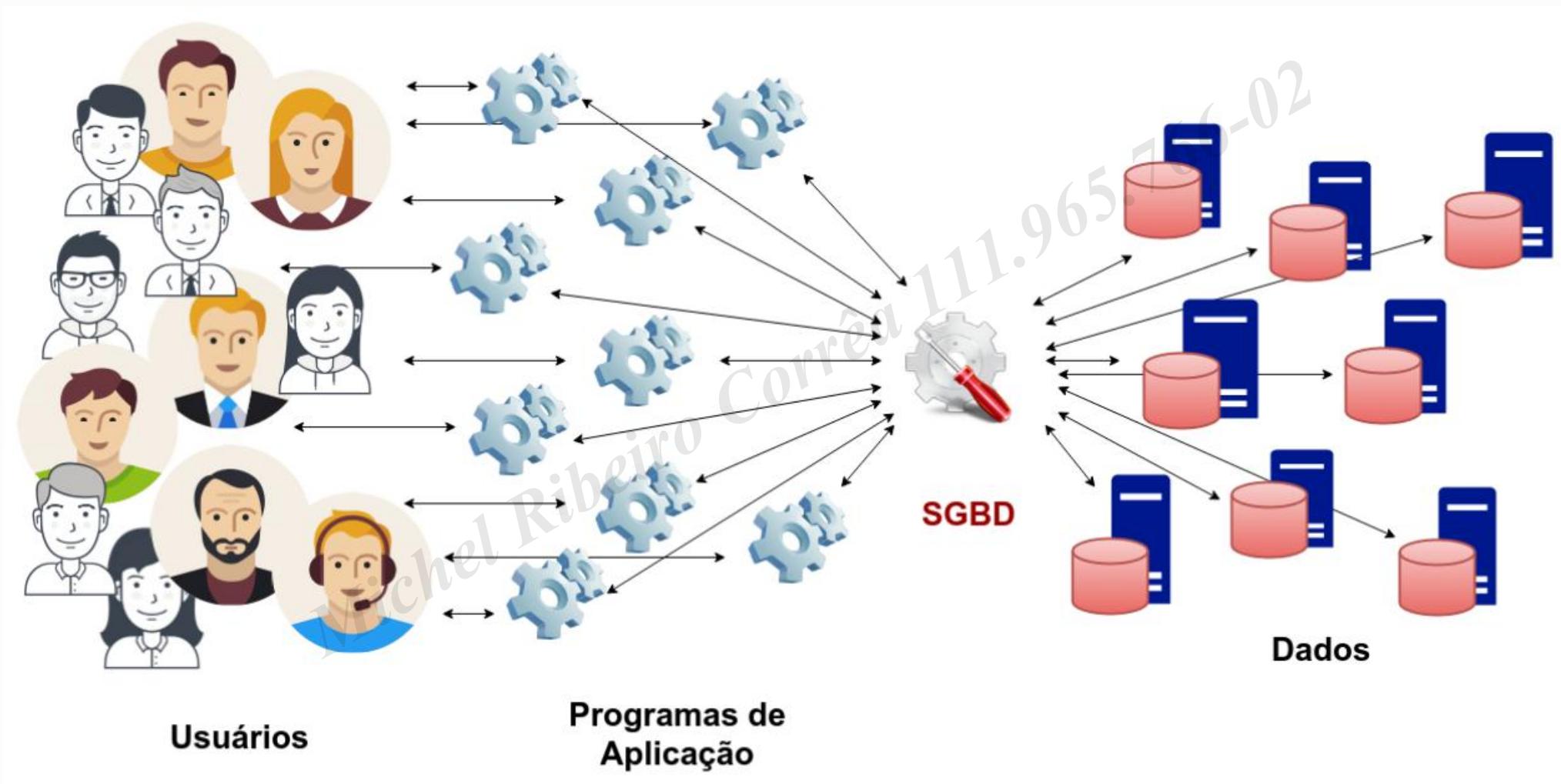
Apoia o ciclo de vida de um BD:

- ◆ **Definir um BD** ⇒ especificar os tipos, as estruturas e as restrições para os dados que serão armazenados no BD
- ◆ **Construir um BD** ⇒ gravar os dados em algum meio de armazenamento (controlado pelo SGBD)
- ◆ **Manipular um BD** ⇒ realizar funções como consultas ao BD para recuperar dados específicos, atualizar o BD para refletir mudanças no minimundo, etc.
- ◆ **Compartilhar um BD** ⇒ permitir que múltiplos usuários e programas accessem-no simultaneamente

Desafios do Gerenciamento de Bancos de Dados



Desafios do Gerenciamento de Bancos de Dados



Sistema de Gerenciamento de Banco de Dados (SGBD)

Funções Importantes de um SGBD:

- ◆ Manter os dados por um longo período de tempo
- ◆ Garantir a consistência dos dados
- ◆ Prover eficiência no acesso aos dados
- ◆ Proteger os dados contra acessos não autorizados
- ◆ Proteger os dados contra falhas de hardware ou software
- ◆ Prover isolamento entre programas e dados
- ◆ Prover suporte a visões múltiplas dos dados
- ◆ Possibilitar o compartilhamento de dados e processar transações multiusuários

Sistema de Gerenciamento de Banco de Dados (SGBD)

Oferecer abstração de dados é uma característica fundamental dos SGBDs, ocultando de usuários e aplicações detalhes sobre a organização e armazenamento dos dados.

◆ A abstração é feita por meio de modelos de dados:

- **Modelo de dados** – conjunto de conceitos usados para descrever a estrutura de um banco de dados + operações básicas para a recuperação e atualização de dados do banco
- **Estrutura de um banco de dados** – define os tipos de dados, relacionamentos e restrições que se aplicam aos dados

Categorias de Modelos de Dados

- ◆ **Modelos de dados conceituais (ou de alto nível)** – possuem conceitos que descrevem os dados como os usuários os percebem
- ◆ **Modelos de dados físicos (ou de baixo nível)** – possuem conceitos que descrevem os detalhes de como os dados estão armazenados no computador, com informações como o formato de um registro, as ordenações dos registros e os caminhos de acesso
- ◆ **Modelos de dados representativos (ou de implementação)** – estão a um meio termo dos dois anteriores, pois podem ser entendidos pelos usuários finais, ao mesmo tempo em que não estão tão distantes da forma como os dados são organizados e armazenados no computador
 - Os representativos são os mais usados nos SGBDs comerciais tradicionais

Sistema de Gerenciamento de Bancos de Dados (SGBD)

Natureza autodescritiva

- ◆ BDs estruturados costumam ser mantidos com uma descrição completa de sua estrutura e restrições (metadados)
- ◆ Os metadados são armazenados no catálogo do SGBD (uma área de armazenamento dedicada a isso), e são usados tanto pelo SGBD quanto por usuários dos BDs
- ◆ O SGBD usa os metadados para trabalhar de forma eficiente com bancos de dados com estruturas diferentes
 - Os metadados são usados no processamento e otimização de operações de modificação e recuperação de dados

Exemplo: BD com dados de alunos e disciplinas

ALUNO

Nome	Numero_aluno	Tipo_aluno	Curso
Silva	17	1	CC
Braga	8	2	CC

DISCIPLINA

Nome_disciplina	Numero_disciplina	Creditos	Departamento
Introd. à ciênciada computação	CC1310	4	CC
Estruturas de dados	CC3320	4	CC
Matemática discreta	MAT2410	3	MAT
Banco de dados	CC3380	3	CC

TURMA

Identificacao_turma	Numero_disciplina	Semestre	Ano	Professor
85	MAT2410	Segundo	07	Kleber
92	CC1310	Segundo	07	Anderson
102	CC3320	Primeiro	08	Carlos
112	MAT2410	Segundo	08	Chang
119	CC1310	Segundo	08	Anderson
135	CC3380	Segundo	08	Santos

Fonte da figura: Elmasri e Navathe, Sistemas de Bancos, 7ª Edição, Pearson.

HISTORICO_ESCOLAR

Numero_aluno	Identificacao_turma	Nota
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PRE_REQUISITO

Numero_disciplina	Numero_pre_requisito
CC3380	CC3320
CC3380	MAT2410
CC3320	CC1310

Exemplo: Metadados do BD de alunos e disciplinas

RELACOES

Nome_relacao	Numero_de_colunas
ALUNO	4
DISCIPLINA	4
TURMA	5
HISTORICO_ESCOLAR	3
PRE_REQUISITO	2

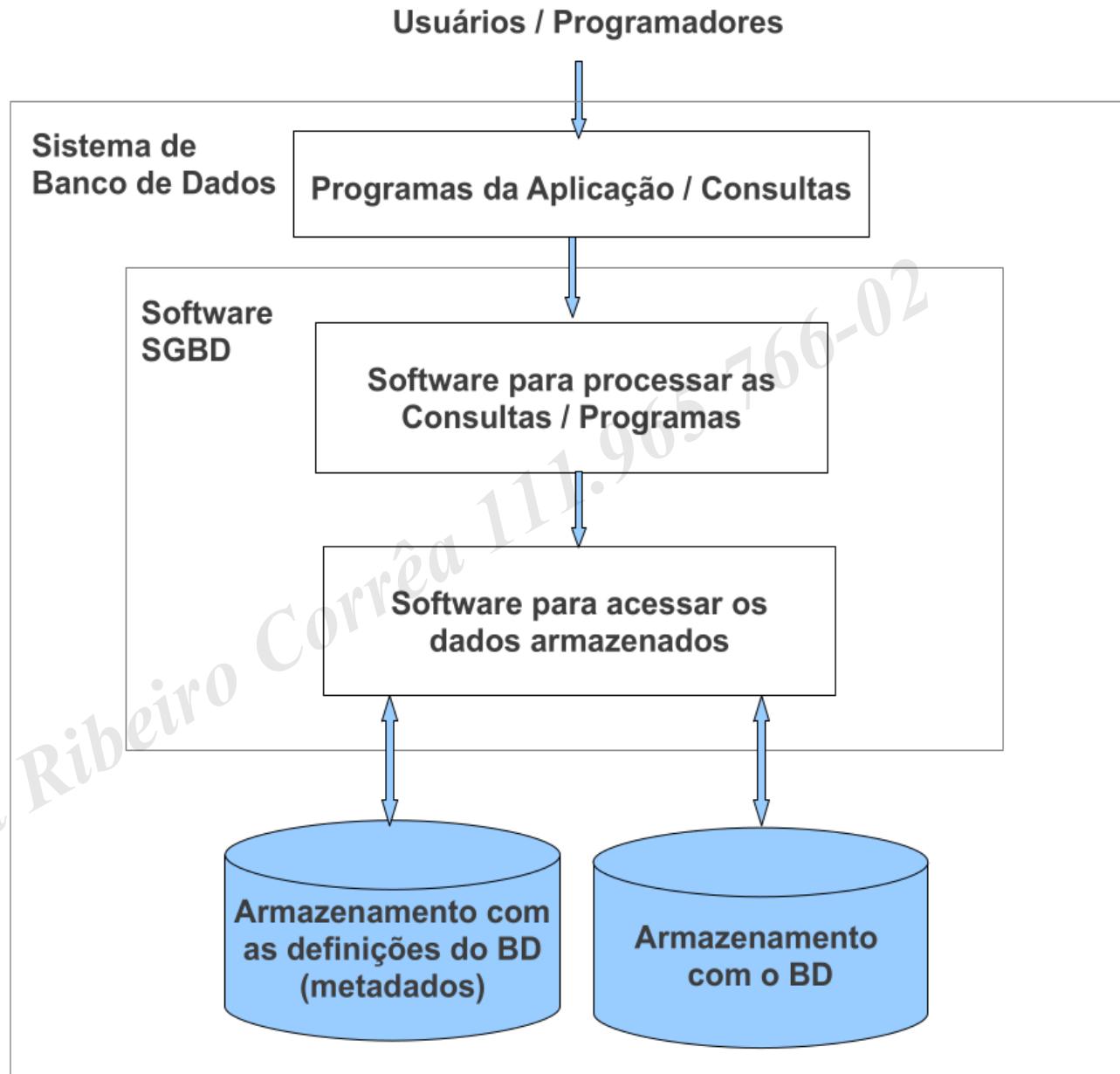
Fonte da figura: Elmasri e Navathe, Sistemas de Bancos, 7ª Edição, Pearson.

COLUMNAS

Nome_coluna	Tipo_de_dado	Pertence_a_relacao
Nome	Caractere (30)	ALUNO
Numero_aluno	Caractere (4)	ALUNO
Tipo_aluno	Inteiro (1)	ALUNO
Curso	Tipo_curso	ALUNO
Nome_disciplina	Caractere (10)	DISCIPLINA
Numero_disciplina	XXXXNNNN	DISCIPLINA
...
...
...
Numero_pre_requisito	XXXXNNNN	PRE-REQUISITO

Nome do item de dados	Posicionamento inicial no registro	Tamanho em caracteres (bytes)
Nome	1	30
Numero_aluno	31	4
Tipo_aluno	35	1
Curso	36	4

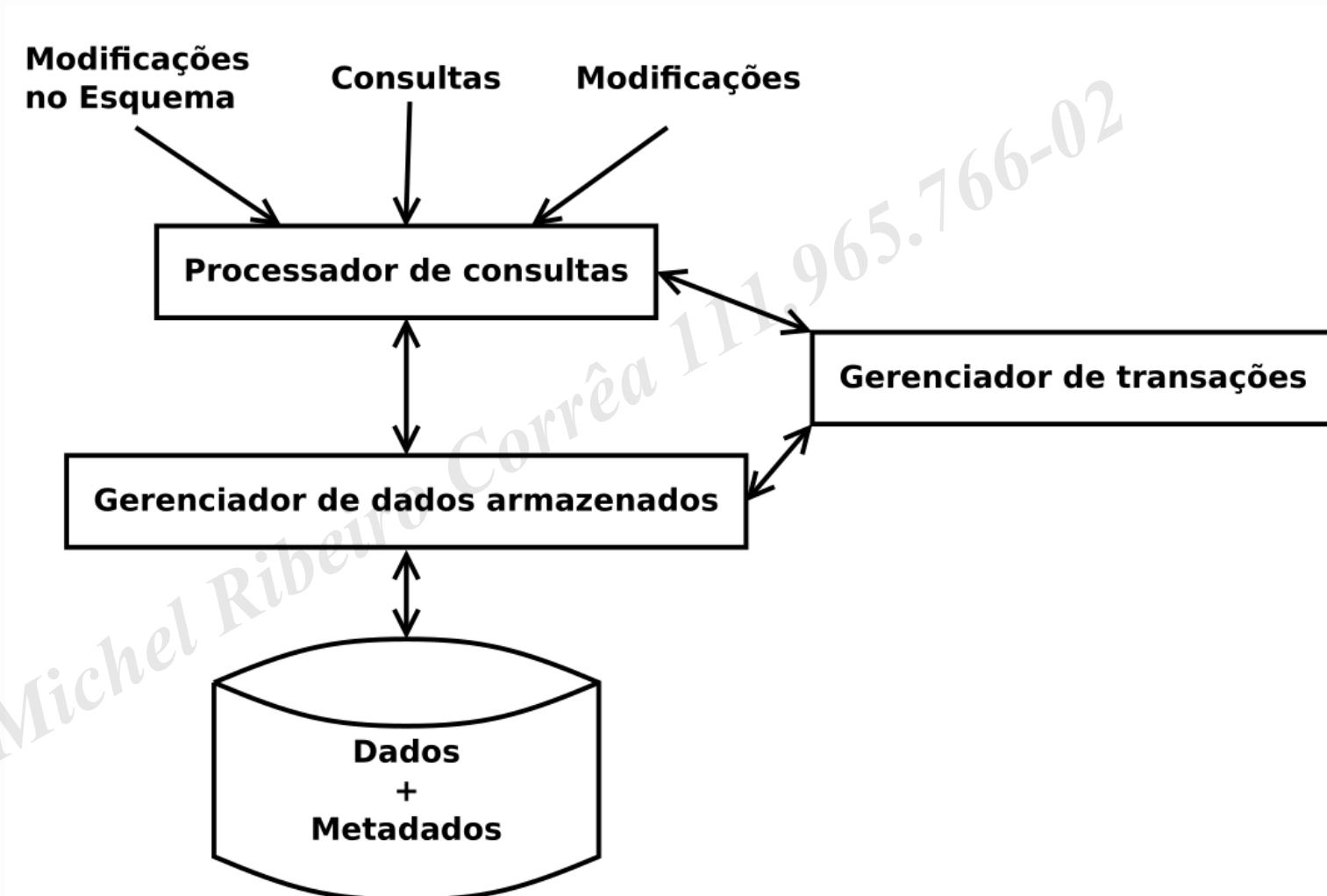
Sistema de Banco de Dados



O Acesso a um Banco de Dados

- ◆ Um usuário ou programa acessa um banco de dados enviando **consultas** ou **transações** para o SGBD que o mantém
 - **Consulta** – comando que recupera dados do BD
 - **Transação** – conjunto de comandos que lê ou escreve dados do/no BD
- ◆ Consultas e transações são enviadas a um SGBD por meio de:
 - Um software cliente de conexão
 - APIs (bibliotecas de funções acessíveis por programação), distribuídas pelos fornecedores do SGBD

Arquitetura de um SGBD (Visão Simplificada)



Linguagens de um SGBD

◆ Linguagem de Definição dos Dados (DDL – *Data Definition Language*)

- É usada por DBAs (administradores) e projetistas de BDs para definir esquemas, ou seja, a estrutura dos dados do BD

◆ Linguagem de Manipulação de Dados (DML – *Data Manipulation Language*)

- É a linguagem oferecida pelo SGBD para a manipulação de dados
- Permite a definição de operações de manipulação típicas, como recuperação, inserção, alteração e remoção de dados

◆ Linguagem de Definição de Visões (VDL – *View Definition Language*)

- permite especificar visões de usuários e seus mapeamentos para o esquema conceitual

O Projeto de Bancos de Dados

Slides baseados no livro *Sistemas de Bancos de Dados* (7^a edição),
de Elmasri e Navathe. Pearson, 2010 – Capítulo 2

Projeto de Bancos de Dados

◆ Envolve as seguintes etapas:

1. Levantamento e análise dos requisitos
2. Projeto conceitual
3. Projeto lógico
4. Projeto físico

Fase 1: Levantamento e análise dos requisitos

Nessa fase, o projetista:

- ◆ Registra de forma detalhada e completa os requisitos dos usuários com relação aos dados
- ◆ Identifica também requisitos funcionais conhecidos das aplicações, as operações de consulta e modificação que serão realizadas no BD
- ◆ Sobre os requisitos:
 - São levantados por meio de entrevistas com os produtores e os usuários dos dados
 - Incluem os dados exigidos para processamento, os seus relacionamentos naturais e as informações relevantes para a escolha da plataforma de software para o BD

Fase 2: Projeto Conceitual

- ◆ Fase de criação de um esquema conceitual para o BD, utilizando um **modelo de dados conceitual** (= de alto nível).
- ◆ **Esquema conceitual (definição):**
 - Descrição de requisitos de dados dos usuários, contendo descrições detalhadas sobre os tipos de entidades, relacionamentos e restrições, expressos de acordo com os conceitos existentes no modelo de dados conceitual empregado
- ◆ Características de um esquema conceitual:
 - Fácil compreensão (não envolve detalhes de implementação)
 - Pode ser usado na comunicação com usuários não técnicos
- ◆ Exemplos bastante usados de modelos conceituais:
 - **Modelo Entidade-Relacionamento (ER)**
 - Diagramas de Classe da UML

Fase 3: Projeto Lógico

◆ Essa fase corresponde ao **mapeamento do esquema conceitual para um esquema num modelo de dados de implementação**.

- Passo necessário para a implementação do BD utilizando um SGBD comercial

◆ Etapa opcional: refinamento do esquema

- Ele identifica problemas em potencial no esquema lógico criado e aplica técnicas para melhorar o esquema

◆ Exemplos de modelos de implementação bastante usados:

- Modelo Relacional
- Modelo Objeto-Relacional

Fase 4: Projeto Físico

- ◆ Fase na qual são definidas as estruturas de armazenamento interno, índices, configurações de arquivos (para os arquivos do BD) e outros ajustes finos.
 - Finalidade: otimizar o desempenho das operações de consulta e manipulação dos dados
- ◆ Pode até mesmo modificar o projeto de BD resultante das fases anteriores, a fim de satisfazer critérios de desempenho desejados
 - Ex.: introduzir redundância de dados para acelerar consultas

Modelo Entidade-Relacionamento (ER)

Slides baseados no livro *Sistemas de Bancos de Dados* (7^a edição),
de Elmasri e Navathe. Pearson, 2010 – Capítulo 3

Modelo Entidade-Relacionamento (ER)

- ◆ Criado por Peter Chen em 1976
 - Artigo: Chen, Peter Pin-Shan. "The entity-relationship model—toward a unified view of data." ACM transactions on database systems (TODS) 1.1 (1976): 9-36. <https://dl.acm.org/doi/epdf/10.1145/320434.320440>
- ◆ É simples e de interpretação intuitiva (mesmo para usuários não-especialistas)
- ◆ É o modelo mais usado para a modelagem conceitual de BDs
- ◆ Descreve os dados com base em três conceitos principais:
 - Entidades
 - Relacionamentos
 - Atributos

Entidades e Atributos

- ◆ **Entidade** – “algo” do mundo real, com uma existência independente. Pode ter existência física (ex.: uma pessoa, uma casa, um carro, etc.) ou conceitual (ex.: um projeto, uma disciplina).
- ◆ **Atributos** – propriedades particulares que descrevem uma entidade (ex.: nome da pessoa, endereço, função, etc.). Um dada entidade terá um valor para cada um de seus atributos. Esses valores constituem a maior parte dos dados armazenados em um BD.

Exemplo de tipo de entidade e seus atributos



Tipos de atributos no modelo ER

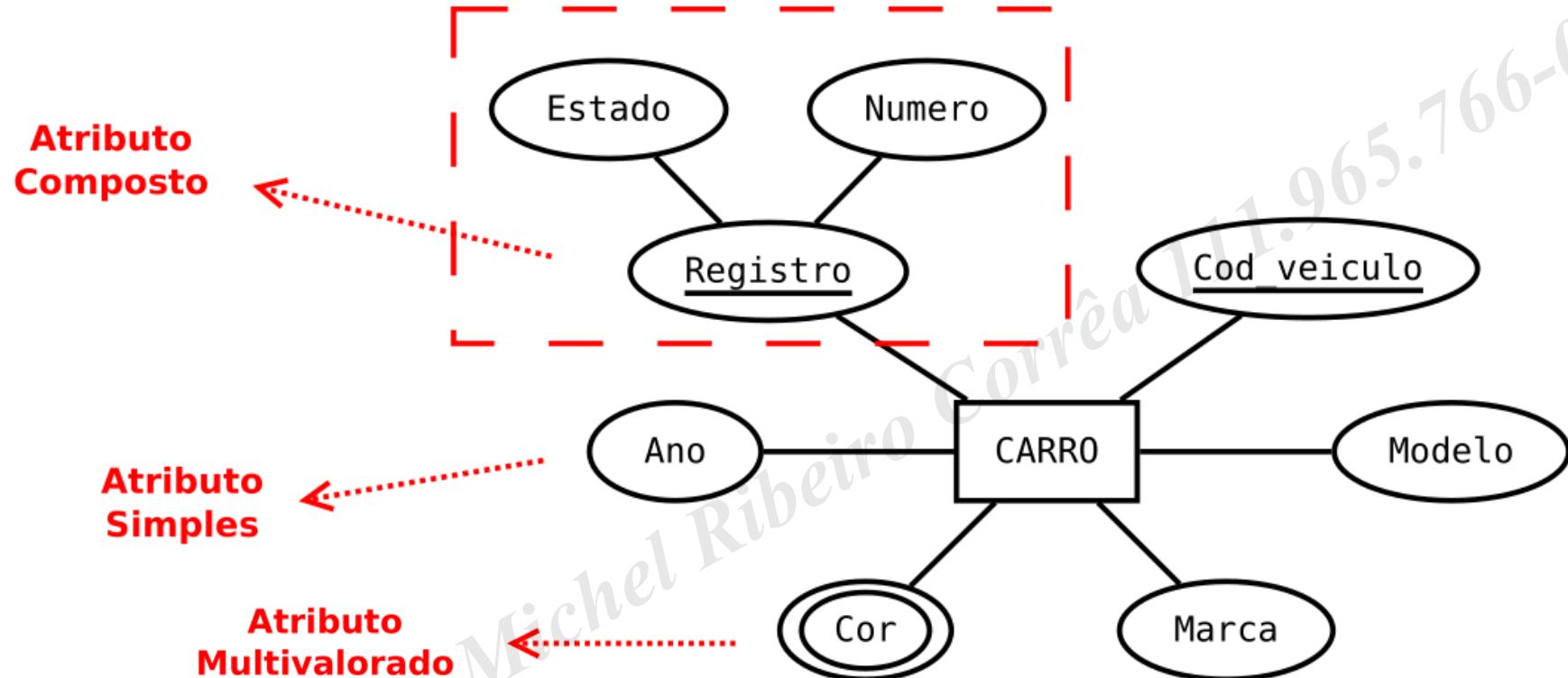
◆ Simples (atômicos) × Compostos

- **Simples (ou atômicos)** – os atributos que não são divisíveis.
- **Compostos** – podem ser divididos em partes menores. O seu valor é dado pela concatenação dos valores dos atributos simples que os compõem.
Ex.: o atributo endereço pode ser dividido em Rua, Cidade, Estado e CEP.

◆ Monovalorados × Multivalorados

- **Monovalorado** – atributo que tem um único valor para uma dada entidade.
Ex.: o atributo idade para uma pessoa.
- **Multivalorado** – atributo para o qual diferentes entidades podem ter diferentes quantidades de valores.
Ex.: atributo titulação para uma pessoa; algumas pessoas não possuem título algum, outras possuem 1, ou 2, ...

Tipos de atributos no modelo ER



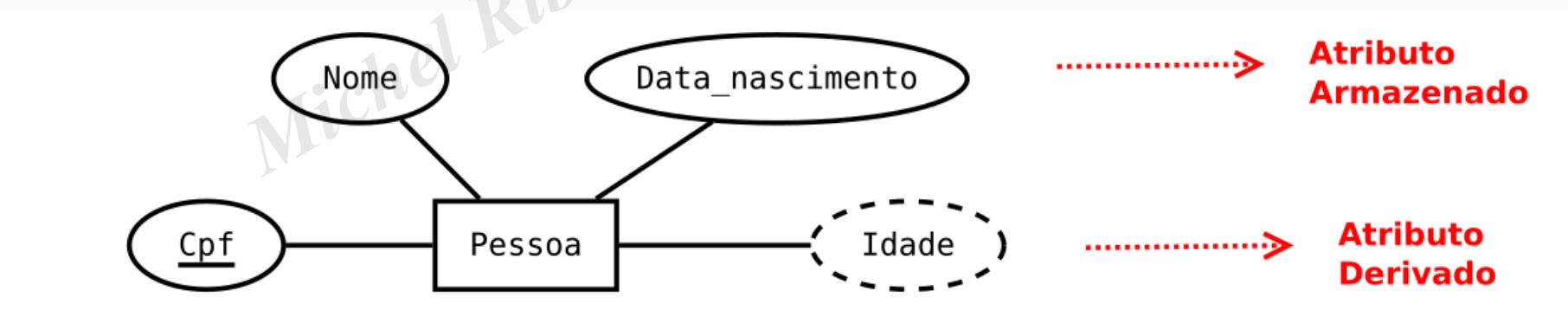
Cada atributo simples possui um domínio de valores, mas eles não são mostrados no diagrama ER.

Tipos de atributos no modelo ER

◆ Armazenados × Derivados

- Atributo derivado – é derivado a partir de outro(s) atributo(s) ou entidade(s) relacionado(s). Exemplos:
- o atributo **idade** é derivado a partir da data de nascimento e da data corrente
- o atributo **número de empregados** de um departamento pode ser derivado da contagem do número de empregados associados ao departamento

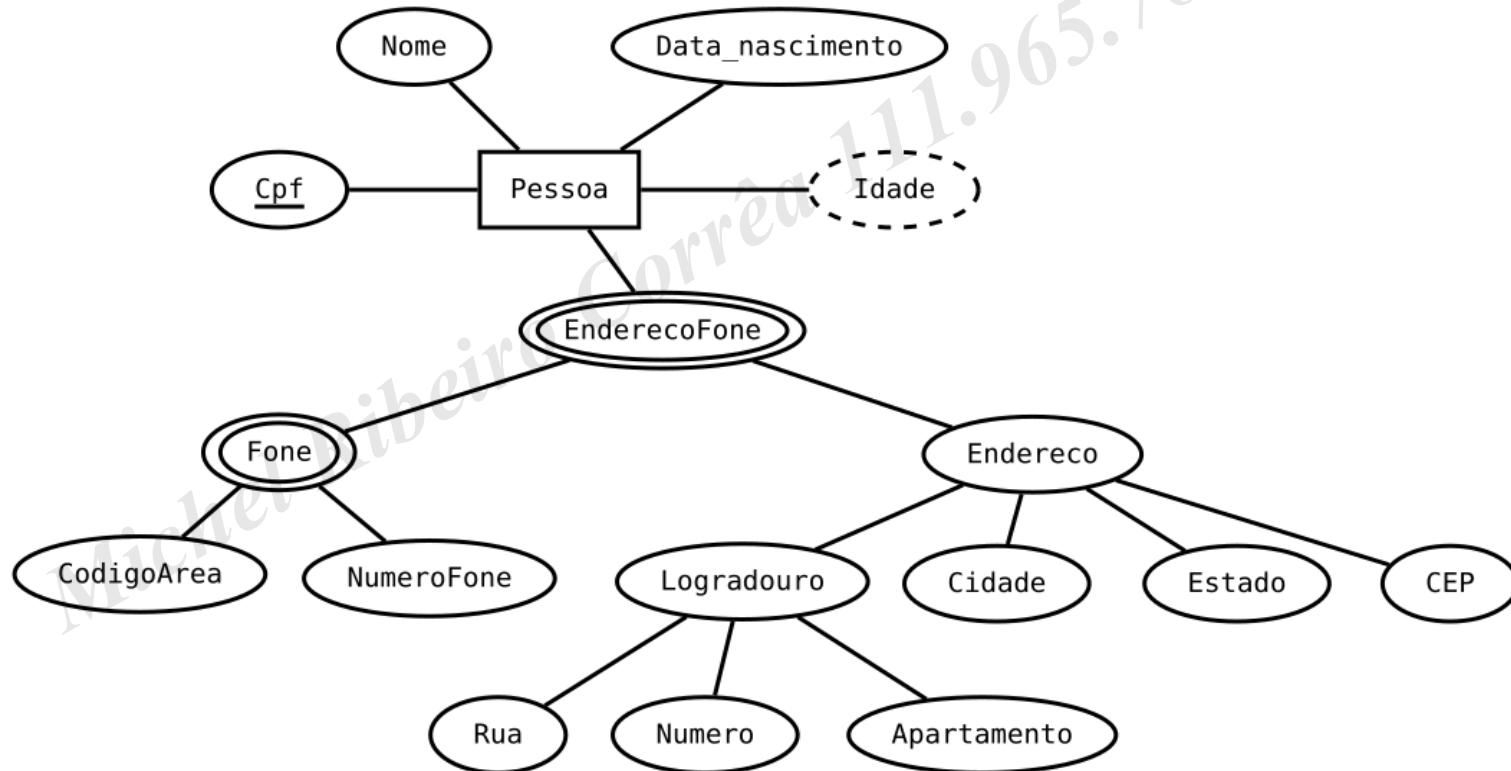
◆ Atributo armazenado – é um atributo que não é derivado



Tipos de atributos no modelo ER

◆ Atributos Complexos

- São atributos compostos e multivalorados, que podem ser aninhados.
- Ex.: uma pessoa pode ter mais de uma residência e cada uma delas pode ter múltiplos telefones.



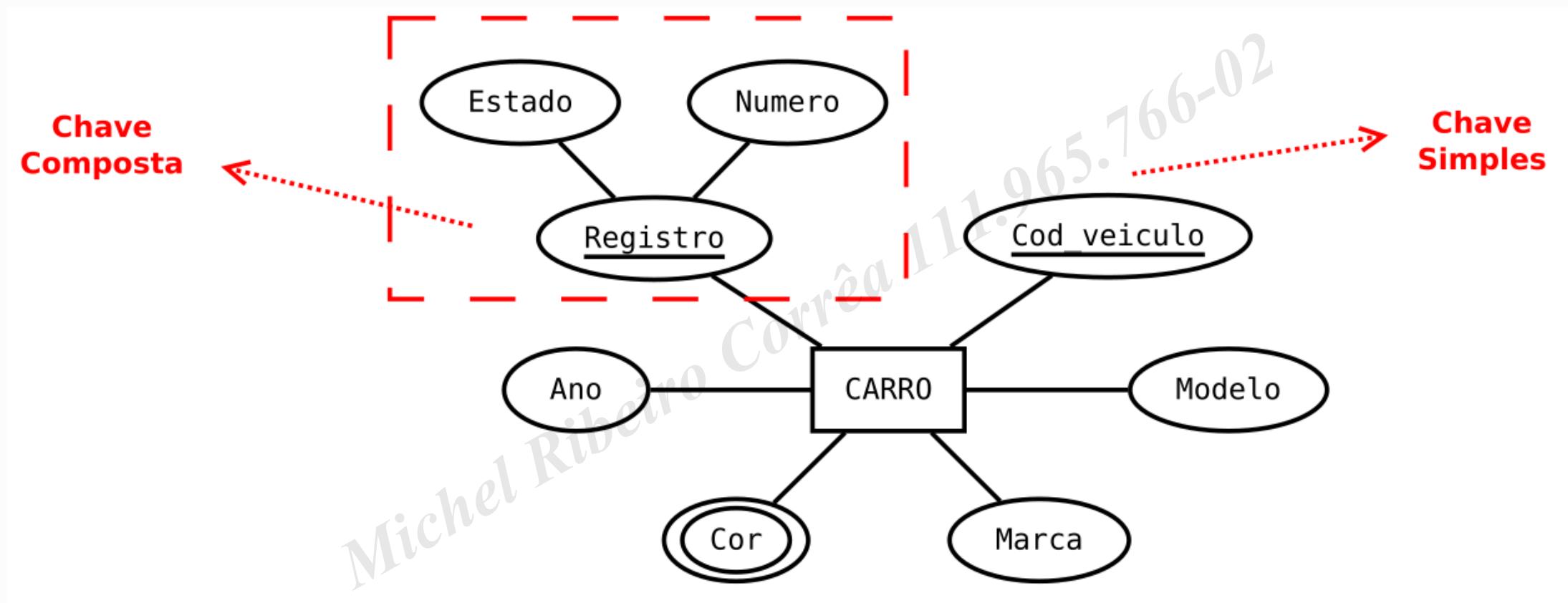
Valor NULL (nulo)

- ◆ É um valor especial, usado quando uma entidade **não possuiu valor para um atributo**.
- ◆ O NULL serve tanto para indicar que um atributo não se aplica a uma dada entidade, quanto para indicar que o valor para um atributo de uma dada entidade é desconhecido.
- ◆ “Desconhecido” se aplica a dois casos distintos:
 - quando é sabido que existe um valor para o atributo, mas ele está faltando (ex.: Altura – todo mundo tem!)
 - quando não é sabido se o valor existe ou não (ex.: FoneResidencial – uma pessoa pode ou não ter)

Tipo de Entidade

- ◆ Um **tipo de entidade** define um conjunto de entidades que possuem os mesmos atributos.
- ◆ Conceitos importantes:
 - **Atributo-chave (restrição de exclusividade)** – é um atributo para o qual o valor é distinto para toda entidade do conjunto de entidades do tipo a que pertence. Os valores do atributo-chave **identificam cada entidade univocamente**.
 - **Chave composta** – é formada por diversos atributos, cuja **combinação dos valores é distinta para cada entidade**. Uma chave composta deve ser **minimal** – todos os atributos componentes devem ser necessários para a garantia da unicidade.
 - Alguns tipos de entidade têm mais de um atributo chave (ex.: os atributos NUSP e CPF para ALUNO). Outros, podem nem ter uma chave – são os chamados **tipos de entidade fraca**.

Notação para Atributos Chave no ER



Exemplo: BD EMPRESA - Requisitos

◆ O objetivo do banco de dados de EMPRESA é registrar dados de funcionários, departamentos e projetos de uma empresa.

◆ Requisitos de dados - Parte 1

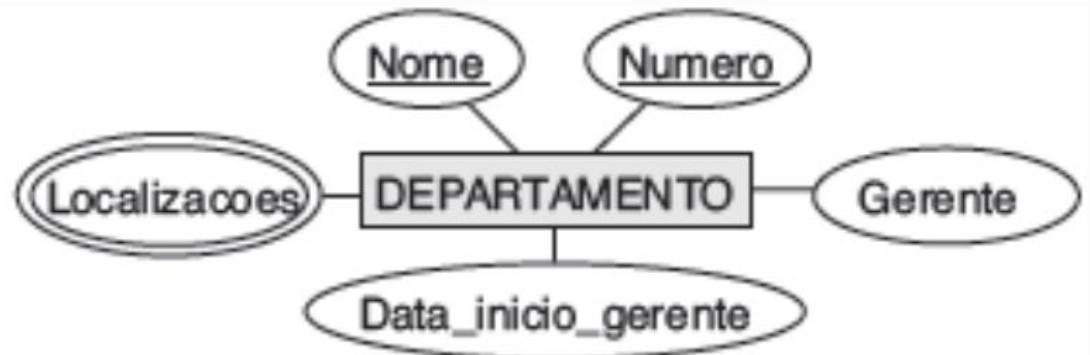
- A empresa é organizada em departamentos. Cada departamento tem um nome exclusivo, um número exclusivo e um funcionário em particular que o gerencia. Registraremos a data inicial em que esse funcionário começou a gerenciar o departamento. Um departamento pode ter vários locais.
- Um departamento controla uma série de projetos, cada um deles com um nome exclusivo, um número exclusivo e um único local.

Exemplo: BD EMPRESA - Requisitos

◆ Requisitos de dados - Parte 2

- Armazenamos o nome (primeiro nome, inicial do meio e último nome), número do Cadastro de Pessoa Física (CPF), endereço, salário, sexo (gênero) e data de nascimento de cada funcionário.
- Um funcionário é designado para um departamento, mas pode trabalhar em vários projetos, que não necessariamente são controlados pelo mesmo departamento. Registraremos o número atual de horas por semana que um funcionário trabalha em cada projeto. Também registramos o supervisor direto de cada funcionário (que é outro funcionário).
- Queremos registrar os dependentes de cada funcionário para fins de seguro. Para cada dependente, mantemos o nome, sexo, data de nascimento e parentesco com o funcionário.

Diagrama ER inicial para o esquema EMPRESA

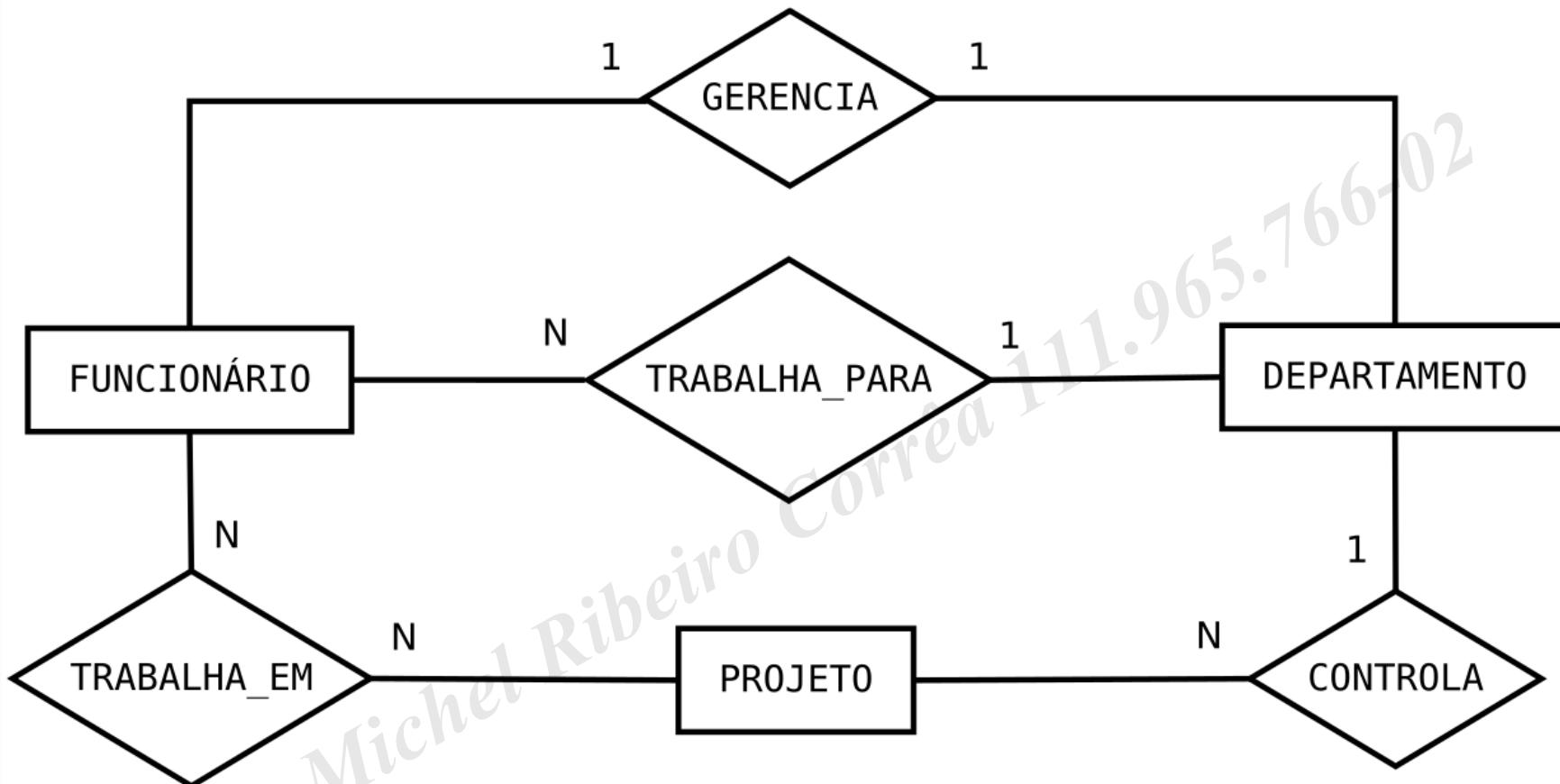


Fonte das figuras Elmasri e Navathe, Sistemas de Bancos, 7^a Edição, Pearson.

Tipo de Relacionamento

- ◆ Um tipo de relacionamento R entre n tipos de entidades E_1, E_2, \dots, E_n define um conjunto de associações (= relacionamentos) entre as entidades desses tipos.
- ◆ Matematicamente, um tipo de relacionamentos R é um conjunto de instâncias de relacionamento r_i , onde cada r_i associa n entidades individuais (e_1, e_2, \dots, e_n), e cada entidade e_j em r_i é um membro do tipo de entidade E_j , $1 \leq j \leq n$.
- ◆ Cada tipo de entidade E_j é dito **participante** do tipo de relacionamento R , da mesma forma que cada entidade e_j é dita participante da instância de relacionamento r_i .
- ◆ Informalmente, cada r_i em R representa uma **associação de entidades** existente no minimundo, que inclui exatamente uma entidade de cada tipo de entidade participante.
- ◆ Denotado graficamente por um losango no diagrama ER.

Alguns Tipos de Relacionamentos do esquema EMPRESA



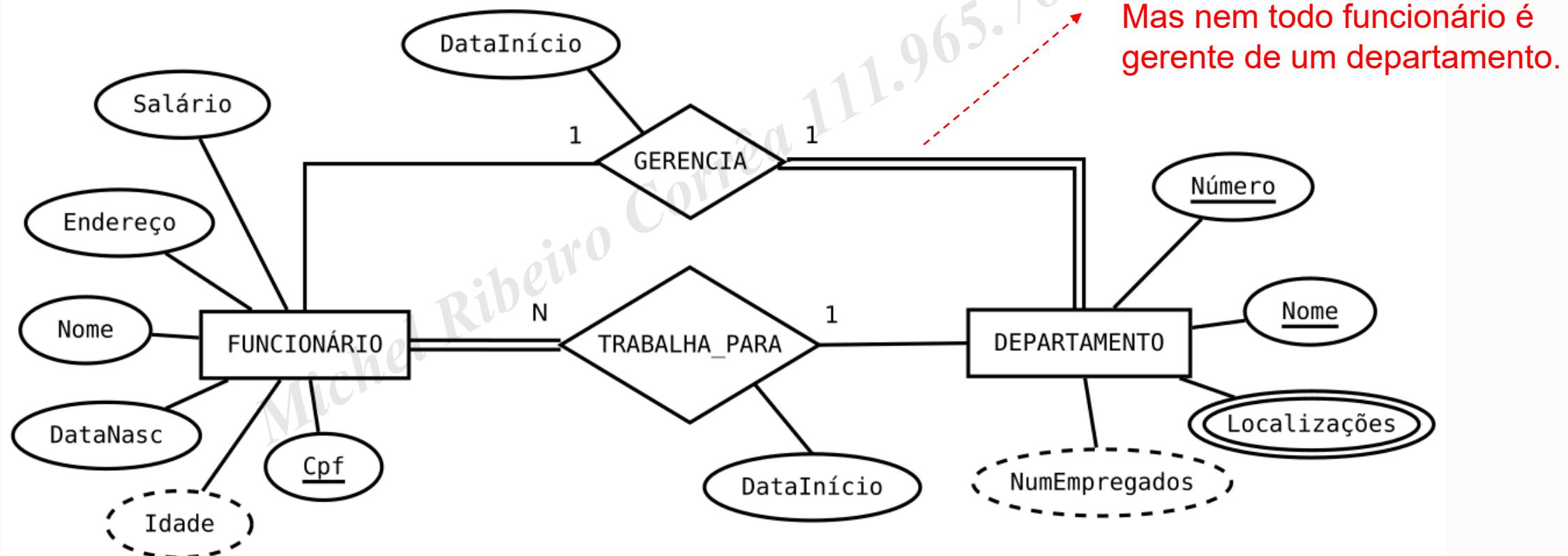
Os atributos foram ocultados aqui só para enfatizar os tipos de relacionamento e suas restrições de cardinalidade.

Propriedades de um Tipo de Relacionamento

- ◆ **Grau:** é o número de tipos de entidades que participam do tipo de relacionamento. Os mais frequente são os relacionamentos binários (de grau 2).
- ◆ **Nome do papel:** Indica o papel que cada tipo de entidade desempenha em um tipo de relacionamento.
- ◆ **Restrições:** Existem restrições (determinadas por situações do minimundo) que limitam as combinações de entidades que podem participar de um relacionamento binário. Tipos:
 - ◆ **Razão de cardinalidade:** especifica o número máximo de instâncias do relacionamento em que uma entidade pode participar. Podem ser: **1:1 (um para um)**, **1:N (um para muitos)**, **N:1 (muitos para um)**, **M:N (muitos para muitos)**
 - ◆ **Restrição de participação (ou de cardinalidade mínima):** Determina o número mínimo de instâncias de relacionamento em que uma entidade deve participar. Pode ser **total (linha dupla)** ou **parcial (linha simples)**.

Refinando os Tipos de Relacionamentos de EMPRESA

Tipos de relacionamento podem ter atributos, de forma similar aos tipos de entidade.



DEPARTAMENTO tem participação total: todo departamento tem um funcionário que o gerencia. Mas nem todo funcionário é gerente de um departamento.

Tipo de Entidade Fraca

- ◆ **Tipo de entidade forte** – tipo de entidade que possui um atributo-chave.
- ◆ **Tipo de entidade fraca** – tipo de entidade que não possui um atributo-chave. São denotadas por retângulos de linha dupla.
 - Entidades de tipos de entidade fraca são identificadas por estarem relacionadas (associadas) a entidades de um outro tipo de entidade (chamado de tipo de entidade identificador ou tipo de entidade proprietária).
 - Esse tipo de relacionamento é chamado de **relacionamento identificador** do tipo de entidade fraca e é denotado pelo losango de linha dupla.

Tipo de Entidade Fraca

- ◆ Um tipo de entidade fraca sempre tem uma restrição de participação total em relação ao seu relacionamento identificador.
- ◆ Um tipo de entidade fraca normalmente tem uma **chave parcial**, que é um conjunto de atributos que identifica univocamente as entidades fracas que estão relacionadas a uma mesma entidade proprietária.

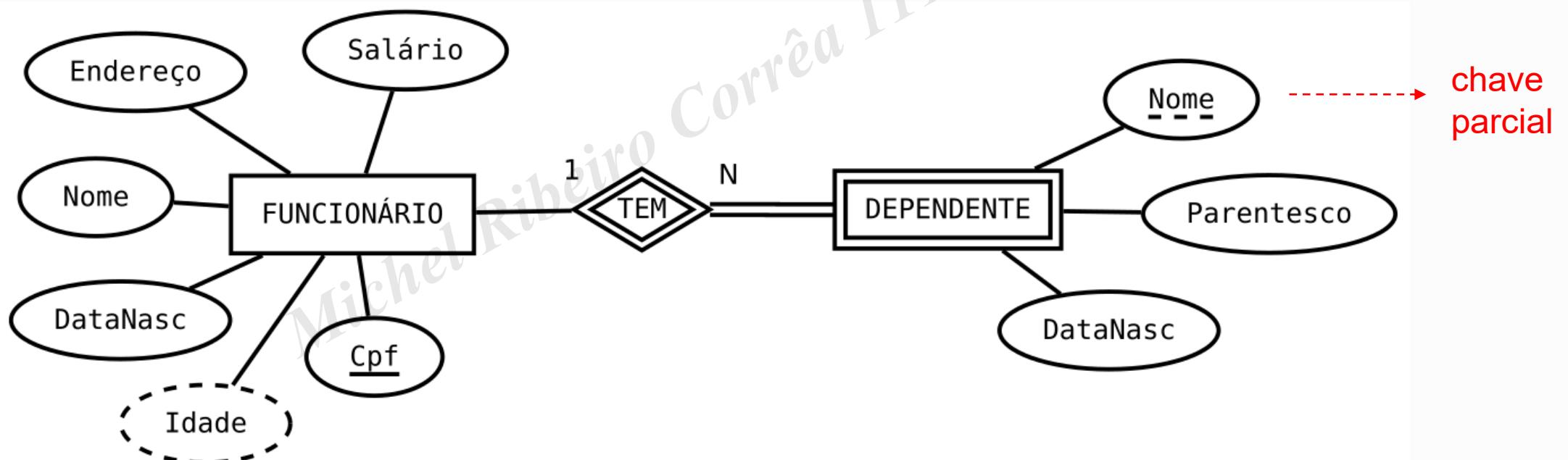
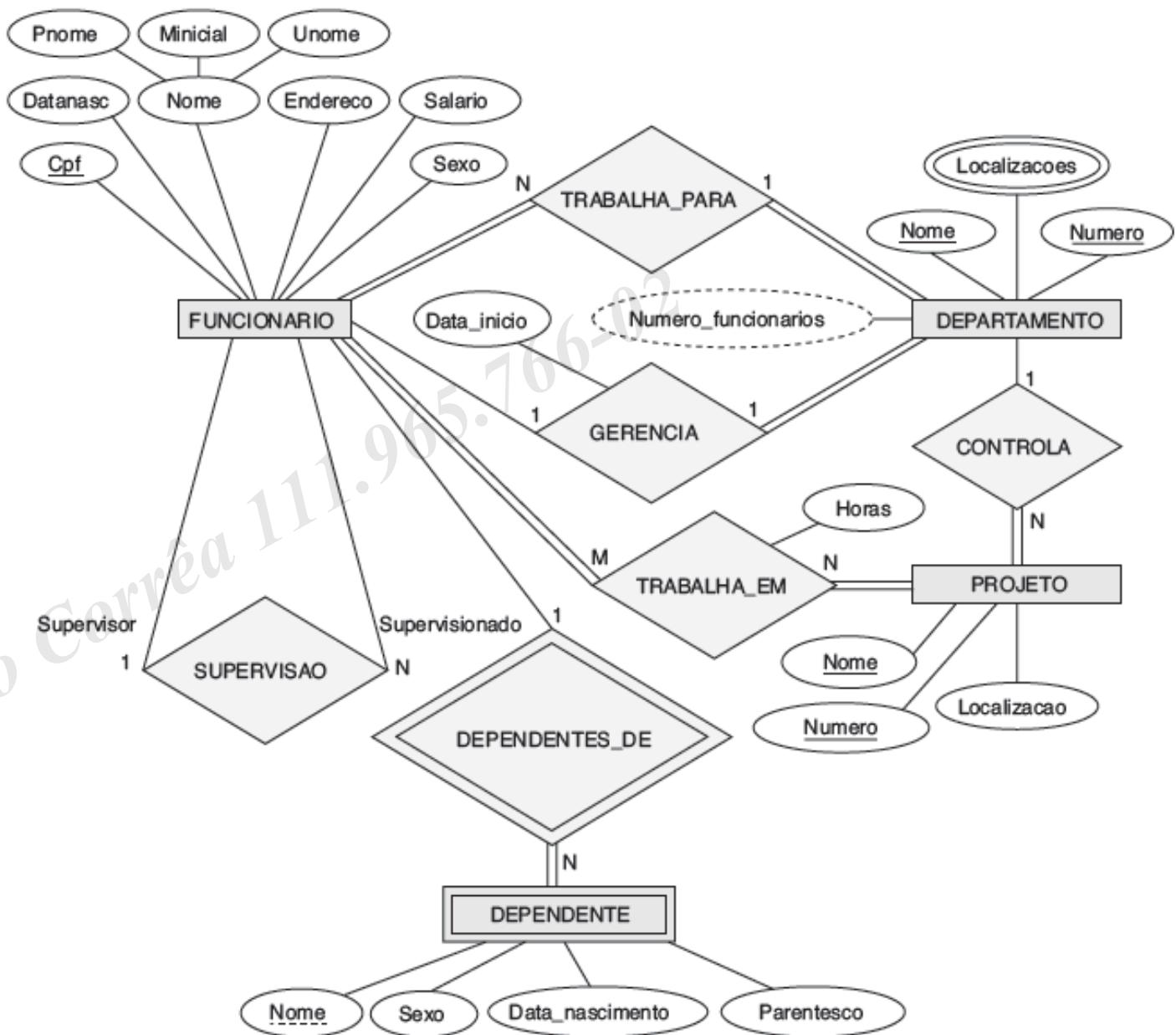
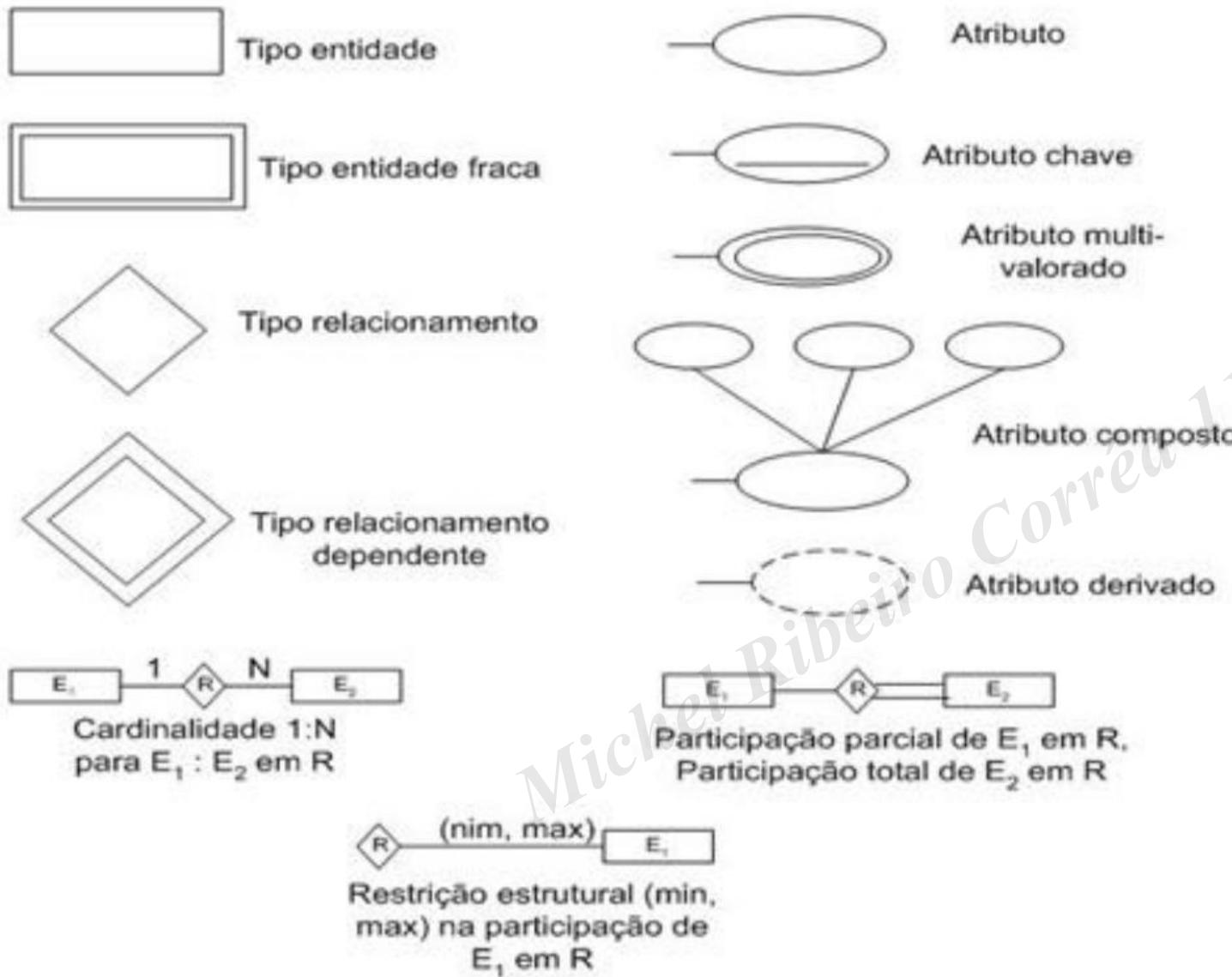


Diagrama completo do esquema ER para o BD EMPRESA



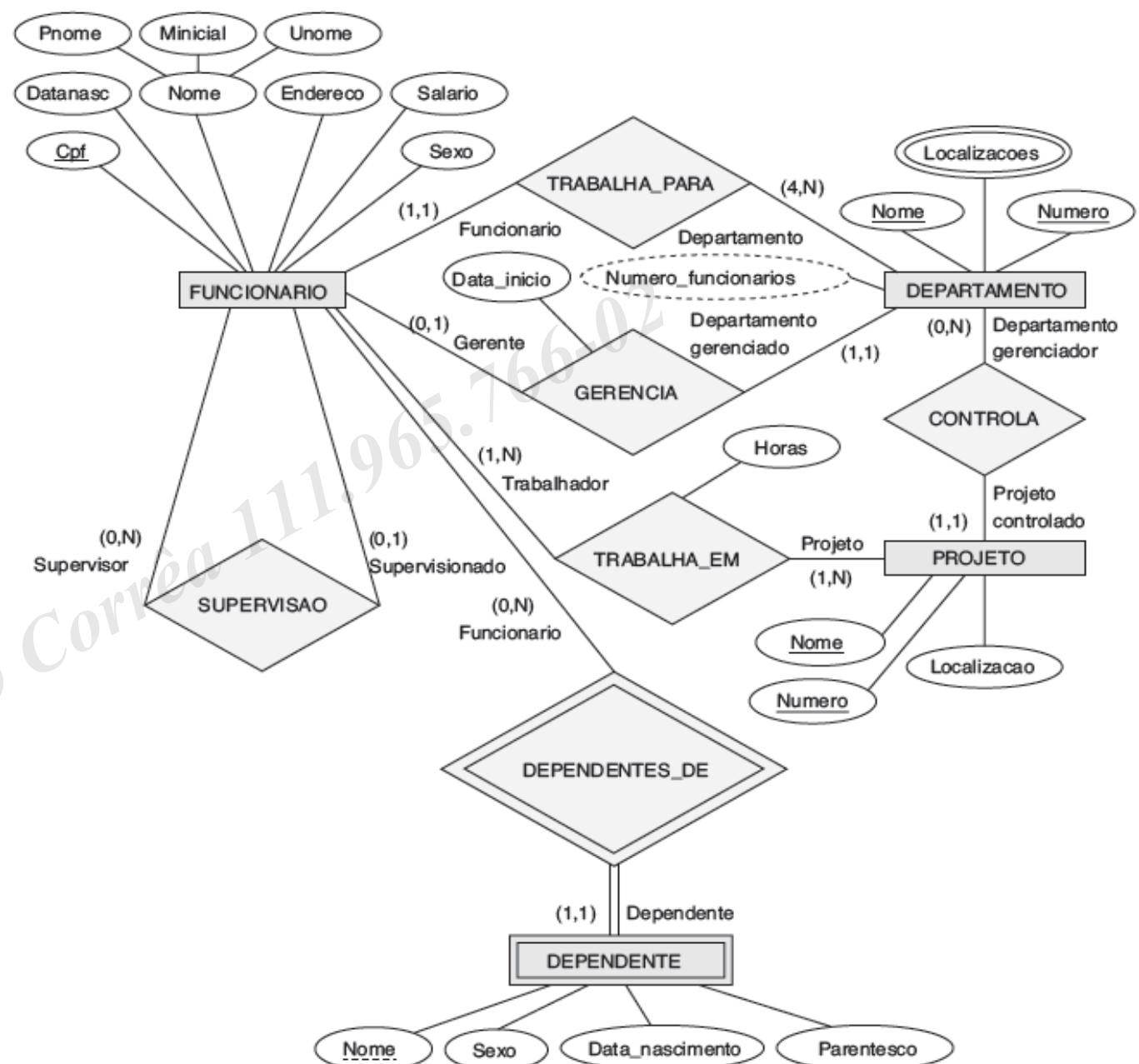
Fonte da figura: Elmasri e Navathe, Sistemas de Bancos, 7^a Edição, Pearson.

Resumo da Notação ER



Fonte da figura: Elmasri e Navathe, Sistemas de Bancos, 7^a Edição, Pearson.

Diagrama do esquema ER para o BD EMPRESA, agora com notação (min,max) e o nomes de papéis



Fonte da figura: Elmasri e Navathe, Sistemas de Bancos, 7^a Edição, Pearson.

Notações Alternativas para Diagramas ER

Símbolos para tipo de entidade, atributo e relacionamento

Símbolos de tipo/
classe de entidade



Símbolos de atributo



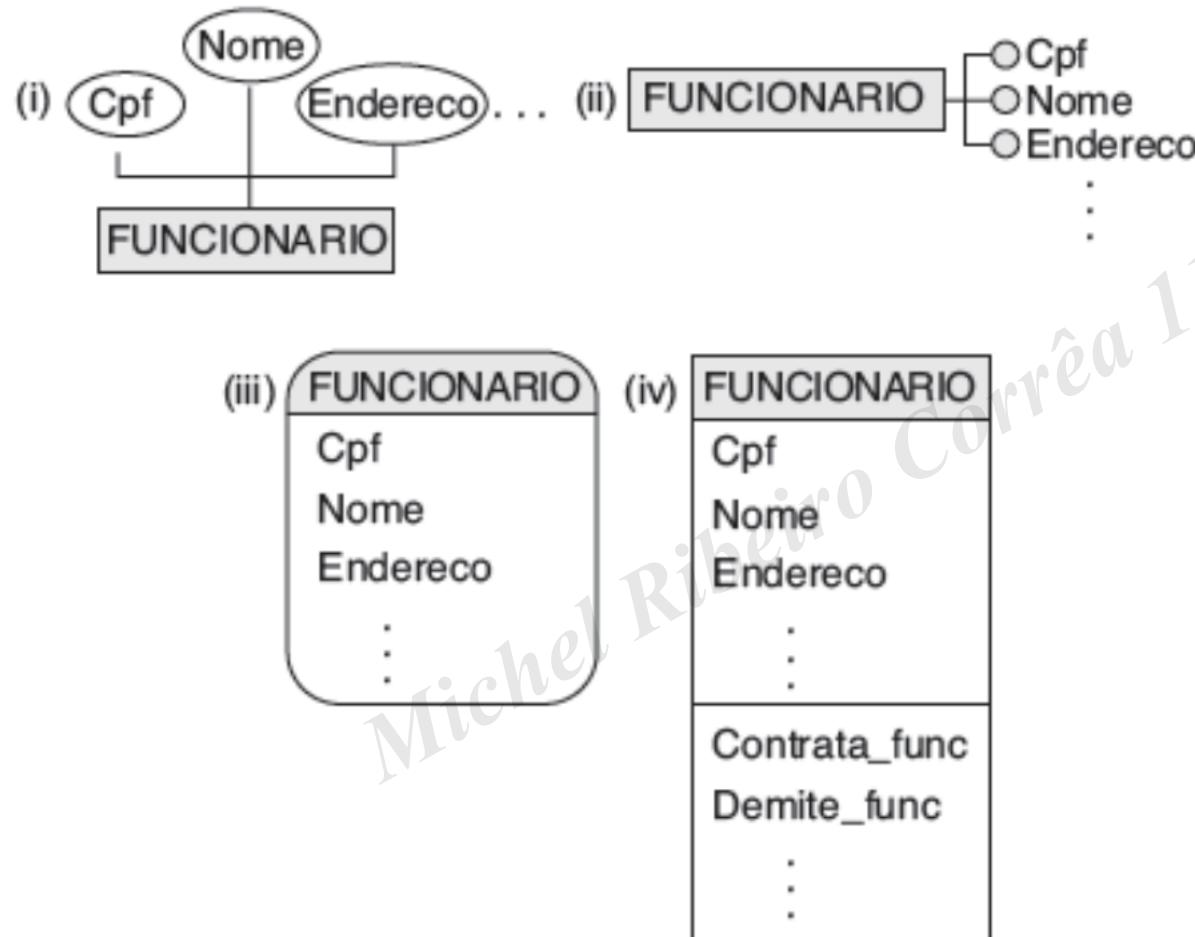
Símbolos de relacionamento



Fonte da figura: Elmasri e Navathe, Sistemas
de Bancos, 7^a Edição, Pearson.

Notações Alternativas para Diagramas ER

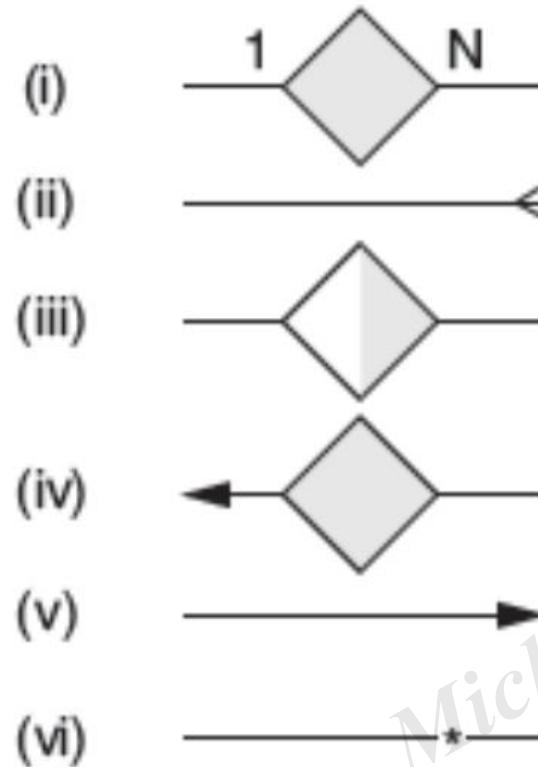
Exibindo atributos



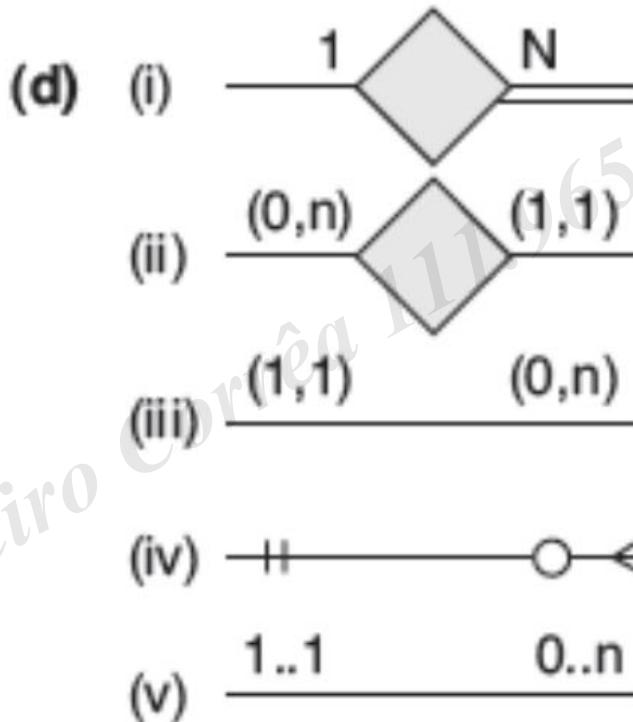
Fonte da figura: Elmasri e Navathe,
Sistemas de Bancos, 7^a Edição,
Pearson.

Notações Alternativas para Diagramas ER

Exibindo cardinalidades

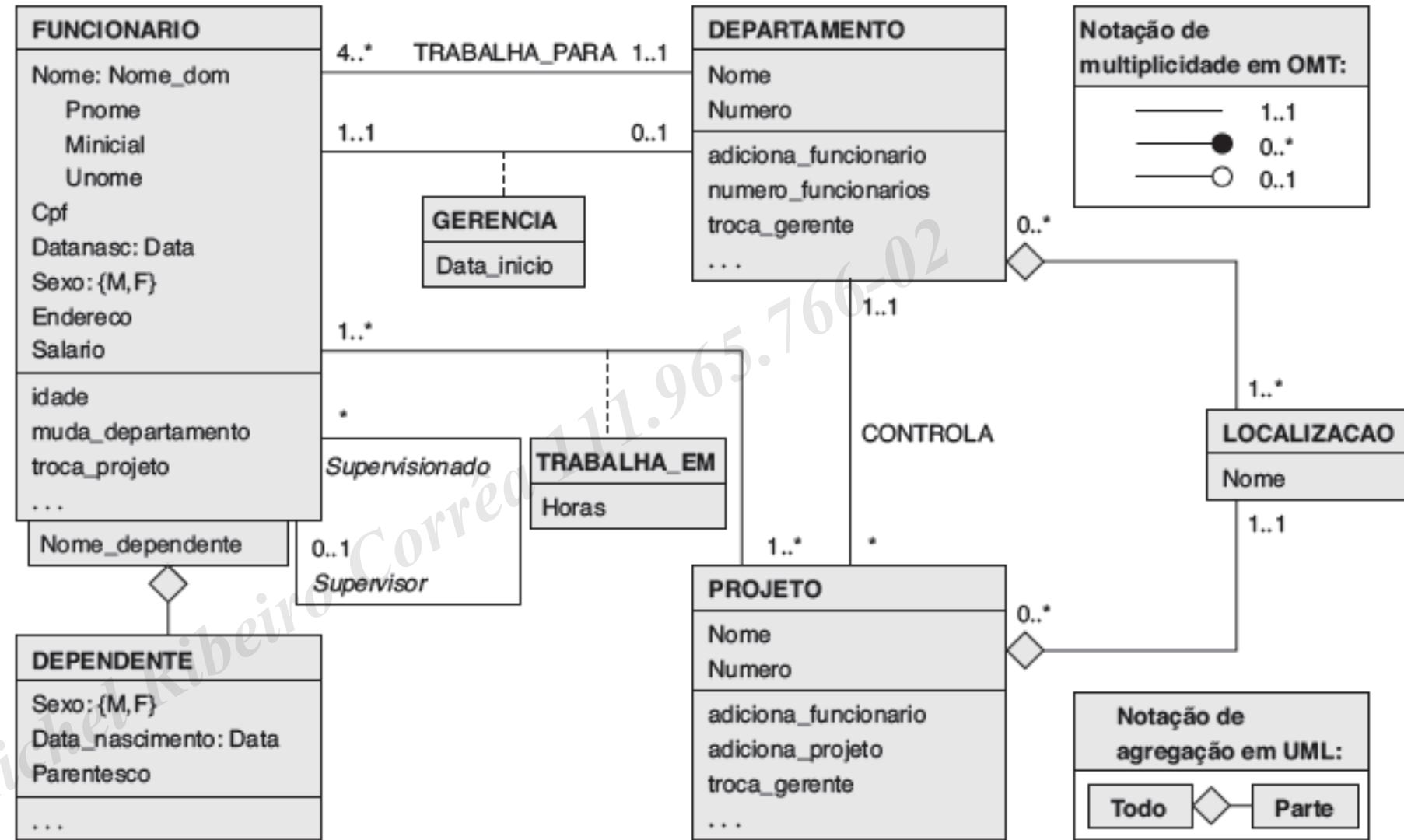


Notações (min,max)



Fonte da figura: Elmasri e Navathe, Sistemas de Bancos, 7^a Edição, Pearson.

Diagrama UML para o BD EMPRESA



Fonte da figura: Elmasri e Navathe, Sistemas de Bancos, 7^a Edição, Pearson.

O Modelo Relacional

Slides baseados no livro *Sistemas de Bancos de Dados* (7^a edição),
de Elmasri e Navathe. Pearson, 2010 – Capítulos 3

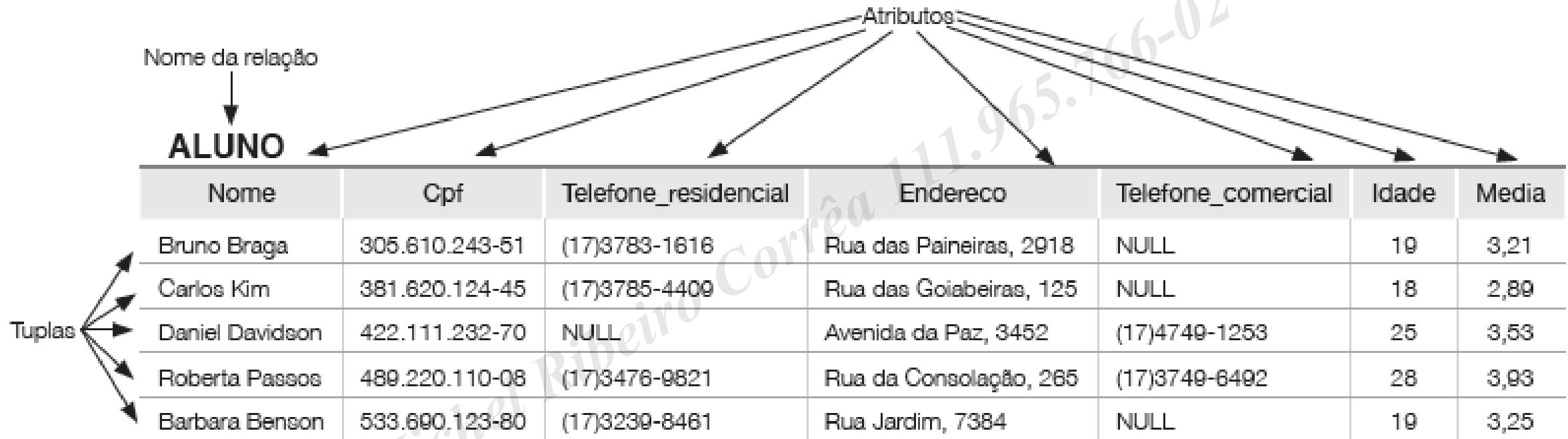
Modelo Relacional

- ◆ É um modelo de dados de implementação
- ◆ Foi introduzido por Edgar Frank "Ted" Codd, da IBM Research, em 1970
 - Codd, Edgar F. "A relational model of data for large shared data banks." *Communications of the ACM* 13.6 (1970): 377-387.
- ◆ Primeiros Sistemas Gerenciadores de Banco de Dados Relacionais (SGBDRs) comerciais surgiram na década de 1980
- ◆ É um modelo simples, mas muito bem fundamentado matematicamente
 - Bloco de construção básico: conceito de relação matemática
 - Bases teóricas: teoria dos conjuntos e a lógica de predicados de primeira ordem

Modelo Relacional

- ◆ No modelo relacional, um banco de dados é uma **coleção de relações**
- ◆ Cada **relação** pode ser vista como uma **tabela de valores**
- ◆ Cada **linha da tabela** representa uma **entidade ou relacionamento** do mundo real
- ◆ Os nomes de tabela e de coluna auxiliam a interpretação do significado dos valores de cada linha
- ◆ Todos os valores em uma mesma coluna possuem um mesmo **domínio** de valores possíveis
- ◆ No modelo relacional, os nomes correspondentes à linha, coluna e tabela são, respectivamente, **tupla, atributo e relação**

Exemplo: Relação ALUNO



Fonte da figura: Elmasri e Navathe, Sistemas de Bancos, 7^a Edição, Pearson.

Domínio

- ◆ Um domínio D é conjunto de valores atômicos
- ◆ Um domínio geralmente é especificado por:
 - um **nome** (para ajudar na interpretação de seus valores)
 - um **tipo de dado** (do qual são retirados os valores que formam o domínio)
 - um **formato**
- ◆ Ex.: o domínio `Numeros_telefone_nacionais`
 - Sequência de caracteres na forma `(dd)ddddddd-dddd` onde cada d é um dígito numérico e os dois primeiros dígitos formam um código de área de telefone válido

Esquema de Relação

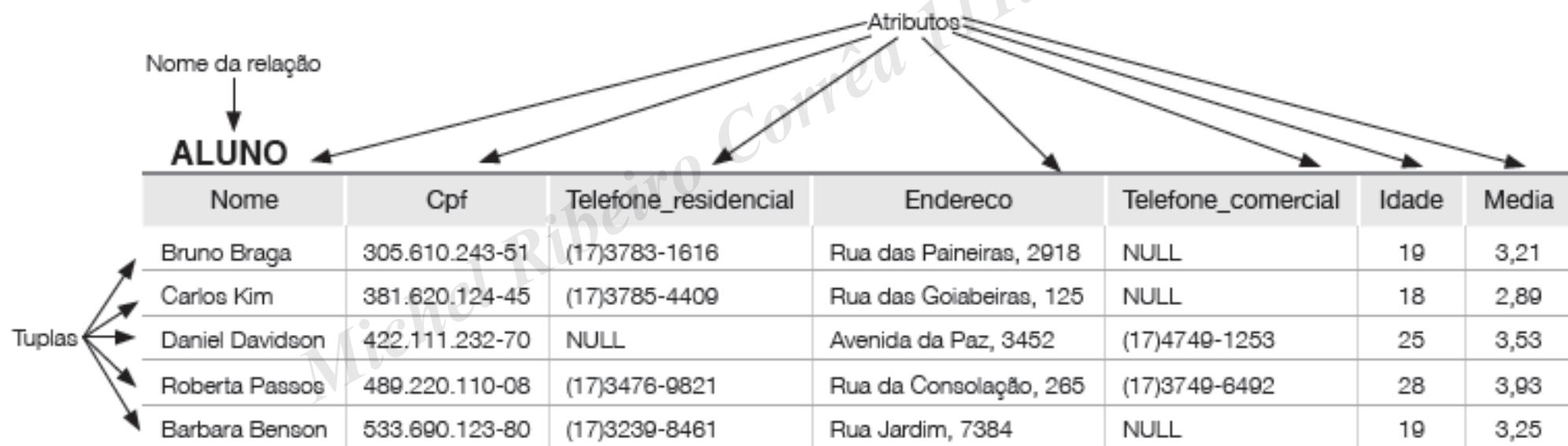
- ◆ **Esquema de relação** é usado para descrever uma relação
- ◆ Um esquema de relação R , indicado por $R = (A_1; A_2; \dots; A_n)$, é composto de um **nome de relação R** e uma **lista de atributos $A_1; A_2; \dots; A_n$**
- ◆ Cada atributo A_i é o **nome de um papel** desempenhado por algum domínio D no esquema de relação R
 - D é chamado domínio de A_i e é denotado por $\text{dom}(A_i)$
- ◆ O **grau de uma relação** é o **número de atributos n** no seu esquema de relação

Relação (ou estado de relação)

- ◆ Uma **relação r (ou estado de relação)** do esquema de relação $R = (A_1; A_2; \dots; A_n)$, também denotada por $r(R)$, é um conjunto de tuplas $r = \{t_1, t_2, \dots, t_m\}$
 - ◆ Cada tupla t_j , $1 \leq j \leq m$, é uma lista ordenada de n valores $t_j = \langle v_1, v_2, \dots, v_n \rangle$, onde cada valor v_i , $1 \leq i \leq n$, é um elemento de $\text{dom}(A_i)$ ou o valor especial NULL
- ◆ Uma relação $r(R)$ é uma relação matemática de grau n sobre os domínios $\text{dom}(A_1), \text{dom}(A_2), \dots, \text{dom}(A_n)$
 - ◆ $r(R)$ é um subconjunto do produto cartesiano dos domínios que definem R :
 - ◆ $r(R) \subseteq (\text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n))$

Exemplo: Relação ALUNO

- ◆ Relação de grau sete, que armazena informações sobre alunos universitários:
- ◆ ALUNO(Nome, Cpf, Telefone_residencial, Endereco, Telefone_comercial, Idade, Media)
- ◆ Usando o tipo de dado de cada atributo, a definição algumas vezes é descrita por:
 - ALUNO(Nome: string, Cpf: string, Telefone_residencial: string, Endereco: string, Telefone_comercial: string, Idade: integer, Media: real)



Fonte da figura: Elmasri e Navathe, Sistemas de Bancos, 7^a Edição, Pearson.

Características das Relações

- ◆ Uma relação é um conjunto de tuplas. Logo, as tuplas em uma relação não possuem nenhuma ordem em particular.
 - Quando os registros são armazenados fisicamente no disco, ou quando exibimos a relação como uma tabela, as linhas possuem uma ordem. Essa ordenação das tuplas não faz parte da definição da relação.
 - Uma relação tenta expressar fatos em um nível lógico ou abstrato.
- ◆ Cada valor em uma tupla é atômico. Atributos compostos ou multivalorados não são permitidos.
 - Atributos compostos são representados nas relações apenas em termos dos seus atributos componentes simples
 - Atributos multivalorados são representados em relações separadas

Características das Relações

- ◆ O valor especial NULL é usado para indicar:
 - que o valor para um atributo é desconhecido para uma dada tupla
 - que um atributo não se aplica a uma dada tupla
- ◆ A comparação de valores NULL leva à ambiguidade; por essa razão, devemos sempre que possível evitar o NULL no projeto de BDs

Interpretação de uma Relação

- ◆ O modelo relacional representa fatos sobre entidades e relacionamentos uniformemente: tudo são relações
- ◆ Dificuldade de compreensão: descobrir se uma relação representa um tipo de entidade ou um tipo de relacionamento
- ◆ O projeto conceitual de bancos de dados (modelo ER) lida com esse problema de forma apropriada

Restrições do Modelo Relacional

- ◆ O estado de um BD como um todo corresponde aos estados de todas as suas relações em um determinado instante
- ◆ Geralmente, há muitas restrições para os valores reais em um estado do BD
 - A maioria das restrições são derivadas de regras do minimundo que o BD representa
 - Algumas restrições são inerentes ao modelo relacional

◆ Categorias de Restrições:

- ◆ **Restrições implícitas** – as que são inerentes ao modelo relacional
- ◆ **Restrições explícitas** – as que podem ser expressas diretamente nos esquemas do modelo de dados
- ◆ **Restrições semânticas** (ou regras de negócio) – as que devem ser expressas e impostas pelos programas de aplicação ou por meio de funcionalidades que alguns SGBDs implementam (como *assertions* e *triggers*)

Restrição Implícita

- ◆ Uma relação não possui tuplas repetidas (já que ela é um conjunto de tuplas)

Restrições Explícitas

1. Restrições de domínio

- 2.– Especificam que, dentro de cada tupla, o valor de cada atributo A deve ser um valor atômico do domínio $dom(A)$

3. Restrições sobre valores NULL

- 4.– Restrição sobre os atributos, especifica se valores NULL são ou não permitidos

5. Restrições de chave

6. Restrições de integridade (de entidade e referencial)

Restrições Explícitas: Restrições de Chave

- ◆ Não existe em uma relação duas tuplas que possuam a mesma combinação de valores para todos os seus atributos
- ◆ Normalmente, existem subconjuntos de atributos de um esquema de relação R com a propriedade de que duas tuplas em qualquer estado de relação r de R não têm a mesma combinação de valores para esses atributos
 - Qualquer um desses subconjuntos é chamado de superchave de R
- ◆ Uma **superchave especifica uma restrição de unicidade**, na qual duas tuplas distintas em qualquer estado r de R não podem ter o mesmo valor para os atributos da superchave
- ◆ Uma **chave K** de um esquema R é uma **superchave mínima** de R , ou seja, não podemos remover dela nenhum atributo e ainda manter a restrição de unicidade.

Restrições Explícitas: Restrições de Chave

- ◆ Um esquema de relação pode possuir mais de uma chave; nesses casos, cada uma das chaves é chamada de **chave candidata**.
- ◆ Geralmente, indica-se uma das chaves candidatas como **chave primária** da relação; **essa é a chave usada para identificar as tuplas da relação**.
- ◆ As demais chaves candidatas são designadas como **chaves únicas (unique)**.
- ◆ Notação: num esquema de relação, os atributos da chave primária devem aparecer sublinhados.

Restrições Explícitas: Restrições de Chave

CARRO

<u>Placa</u>	Numero_chassi	Marca	Modelo	Ano
Itatiaia ABC-7039	A6935207586	Volkswagen	Gol	02
Ilu TVP-3470	B4369668697	Chevrolet	Corolla	05
Santos MPO-2902	X8355447376	Fiat	Uno	01
Itanhaem TFY-6858	C4374268458	Chevrolet	Celta	99
Itatiba RSK-6279	Y8293586758	Renault	Clio	04
Atibaia RSK-6298	U0283657858	Volkswagen	Parati	04

- ◆ A relação CARRO possui duas chaves candidatas: Placa e Numero_chassi.
- ◆ O atributo Placa foi definido como chave primária (está denotado com sublinhado), enquanto Numero_chassi foi definido como chave única.

Fonte da figura: Elmasri e Navathe, Sistemas de Bancos, 7ª Edição, Pearson.

Restrições Explícitas

- ◆ Até aqui, vimos apenas restrições explícitas que se aplicam apenas a relações isoladas e seus atributos.
- ◆ As demais restrições explícitas se referem à forma como as relações de um BD se relacionam entre si. Por isso, antes de prosseguir na definição das restrições, definiremos o que é um BD relacional.

Banco de Dados Relacional – Definição Formal

- ◆ Um esquema de banco de dados relacional S é um conjunto de esquemas de relação $S = \{R_1, R_2, \dots, R_m\}$ e um conjunto de restrições de integridade RI .
- ◆ Um estado de um banco de dados relacional BD com esquema S é um conjunto de relações $BD = \{r_1, r_2, \dots, r_m\}$ tal que r_i é uma relação de R_i e satisfaz as restrições de integridade especificadas em RI .
- ◆ O termo **banco de dados relacional** refere-se, implicitamente, ao seu **esquema** e ao seu **estado atual**.
- ◆ Um **estado válido** é um estado de um BD relacional que **satisfaz a todas as restrições de integridade**.

FUNCIONARIO

Pnome	Minitial	Unome	<u>Cpf</u>	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
-------	----------	-------	------------	----------	----------	------	---------	----------------	-----

DEPARTAMENTO

Dnome	<u>Dnumero</u>	Cpf_gerente	Data_inicio_gerente
-------	----------------	-------------	---------------------

LOCALIZACAO_DEP

<u>Dnumero</u>	<u>Dlocal</u>
----------------	---------------

PROJETO

Projnome	<u>Projnumero</u>	Projlocal	Dnum
----------	-------------------	-----------	------

TRABALHA_EM

Fopf	<u>Pnr</u>	Horas
------	------------	-------

DEPENDENTE

Fopf	<u>Nome_dependente</u>	Sexo	Datanaso	Parentesco
------	------------------------	------	----------	------------

Exemplo de um diagrama de esquema para um BD relacional de uma empresa.

Fonte da figura: Elmasri e Navathe, Sistemas de Bancos, 7^a Edição, Pearson.

FUNCIONARIO

Pnome	Minitial	Unome	Cpf	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
-------	----------	-------	-----	----------	----------	------	---------	----------------	-----

DEPARTAMENTO

Dnome	<u>Dnumero</u>	Cpf_gerente	Data_inicio_gerente
-------	----------------	-------------	---------------------

LOCALIZACAO_DEP

<u>Dnumero</u>	<u>Dlocal</u>
----------------	---------------

PROJETO

Projnome	Projnumero	Projlocal	<u>Dnum</u>
----------	------------	-----------	-------------

TRABALHA_EM

Fopf	Pnr	Horas
------	-----	-------

DEPENDENTE

Fopf	Nome_dependente	Sexo	Datanaso	Parentesco
------	-----------------	------	----------	------------

Os atributos que representam um mesmo conceito no mundo real não precisam ter o mesmo nome em relações diferentes.

Exemplo: Dnumero em DEPARTAMENTO, Dnum em PROJETO e Dnr em FUNCIONARIO

Fonte da figura: Elmasri e Navathe, Sistemas de Bancos, 7ª Edição, Pearson.

FUNCIONARIO

Pnome	Mnicial	Unome	Cpf	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
-------	---------	-------	-----	----------	----------	------	---------	----------------	-----

DEPARTAMENTO

Dnome	Dnumero	Cpf_gerente	Data_inicio_gerente
-------	---------	-------------	---------------------

LOCALIZACAO_DEP

Dnumero	Dlocal
---------	--------

PROJETO

Projnome	Projnumero	Projlocal	Dnum
----------	------------	-----------	------

TRABALHA_EM

Fopf	Pnr	Horas
------	-----	-------

DEPENDENTE

Fopf	Nome_dependente	Sexo	Datanaso	Parentesco
------	-----------------	------	----------	------------

Atributos que representam conceitos diferentes não precisam ter nomes diferentes em diferentes relações.

Exemplo: Poderíamos ter um atributo Nome em PROJETO e outro atributo nome em DEPARTAMENTO (no lugar dos atributos Projnome e Dnome)

Fonte da figura: Elmasri e Navathe, Sistemas de Bancos, 7^a Edição, Pearson.

De Volta às Restrições Explícitas: Restrições de Integridade

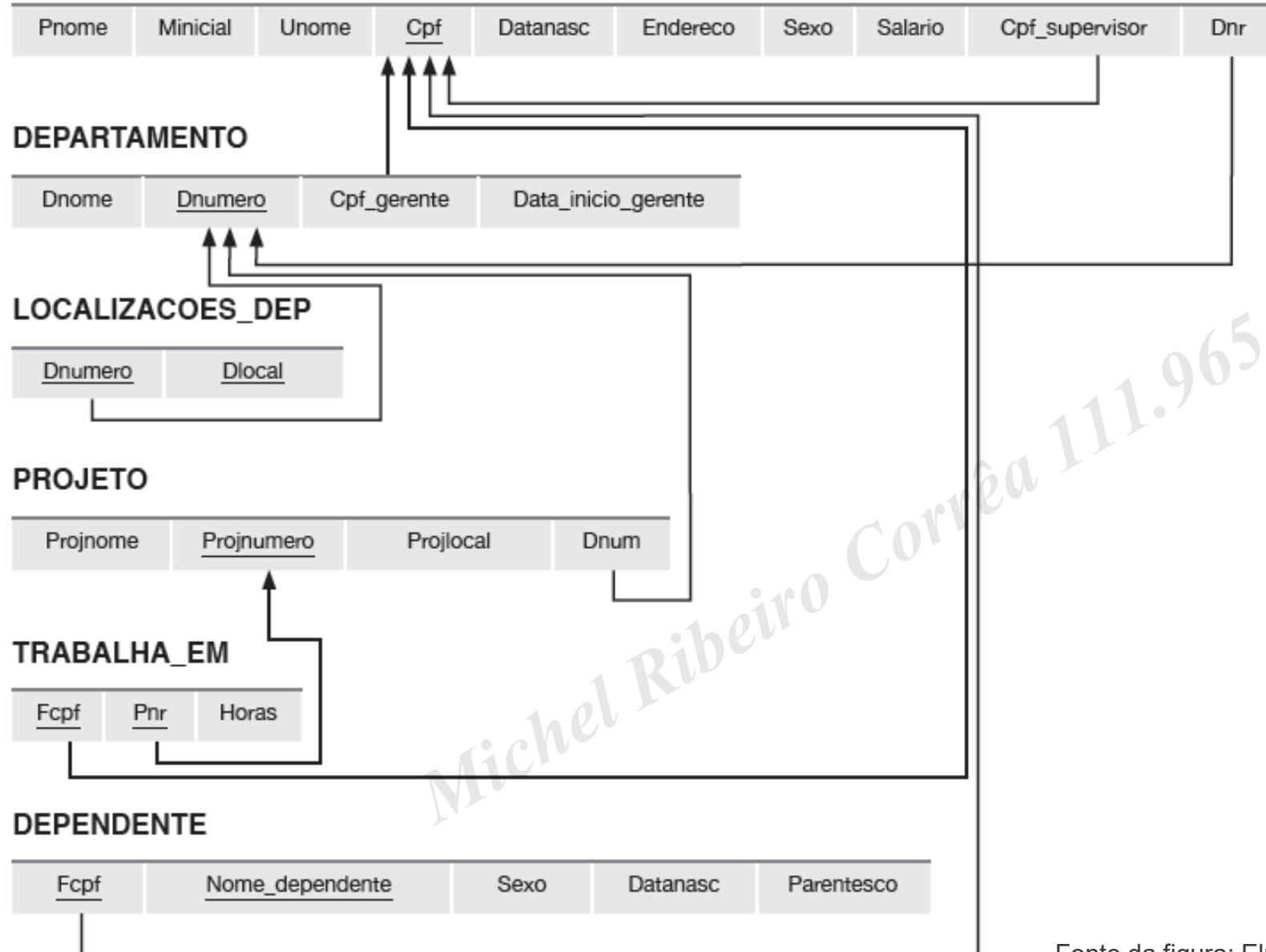
- ◆ **Restrição de integridade de entidade** – estabelece que nenhum valor de chave primária pode ser NULL. Se a chave primária pudesse ser nula, algumas tuplas poderiam não ser identificadas.
- ◆ **Restrição de integridade referencial** – declara que uma tupla em uma relação que faz referência a uma outra relação (ou à mesma relação) deve se referir a uma tupla existente nessa relação. Essa referência é feita por meio de **chaves estrangeiras**.

Chave Estrangeira

Uma **chave estrangeira ChE** de um esquema de relação $R1$ para um esquema de relação $R2$ é um subconjunto de atributos de $R1$ que possuem os mesmo domínios dos atributos da chave primária ChP de $R2$.

Além disso, toda tupla $t1$ no estado atual $r1$ de $R1$ **ou possui um valor NULL para ChE ou $t1[ChE] = t2[ChP]$** , onde $t2$ é uma tupla do estado atual $r2$ de $R2$.

FUNCIONARIO



Exemplo de um diagrama de esquema para um BD relacional de uma empresa, agora com as restrições de integridade referencial (chaves estrangeiras).

Fonte da figura: Elmasri e Navathe, Sistemas de Bancos, 7ª Edição, Pearson.

FUNCIONARIO

Pnome	Minitial	Uname	Cpf	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
João	B	Silva	12345678966	09-01-1965	Rua das Flores, 751, São Paulo, SP	M	30.000	33344555587	5
Fernando	T	Wong	33344555587	08-12-1955	Rua da Lapa, 34, São Paulo, SP	M	40.000	88866555576	5
Aloie	J	Zelaya	99988777767	19-01-1968	Rua Souza Lima, 35, Curitiba, PR	F	25.000	98765432168	4
Jennifer	S	Souza	98765432168	20-06-1941	Av. Arthur de Lima, 54, Santo André, SP	F	43.000	88866555576	4
Ronaldo	K	Lima	66688444476	15-09-1962	Rua Rebouças, 65, Piracicaba, SP	M	38.000	33344555587	5
Joice	A	Leite	45345345376	31-07-1972	Av. Lucas Obes, 74, São Paulo, SP	F	25.000	33344555587	5
André	V	Pereira	98798798733	29-03-1969	Rua Timbira, 35, São Paulo, SP	M	25.000	98765432168	4
Jorge	E	Brito	88866555576	10-11-1937	Rua do Horto, 35, São Paulo, SP	M	55.000	NULL	1

Um exemplo de estado válido do BD Empresa

DEPARTAMENTO

Dnome	Dnumero	Cpf_gerente	Data_inicio_gerente
Pesquisa	5	33344555587	22-05-1988
Administração	4	98765432168	01-01-1995
Matriz	1	88866555576	19-06-1981

LOCALIZACAO_DEP

Dnumero	Dlocal
1	São Paulo
4	Mauá
5	Santo André
5	Itu
5	São Paulo

TRABALHA_EM

Fcpf	Pnr	Horas
12345678966	1	32,5
12345678966	2	7,5
66688444476	3	40,0
45345345376	1	20,0
45345345376	2	20,0
33344555587	2	10,0
33344555587	3	10,0
33344555587	10	10,0
33344555587	20	10,0
99988777767	30	30,0
99988777767	10	10,0
98798798733	10	35,0
98798798733	30	5,0
98765432168	30	20,0
98765432168	20	15,0

PROJETO

Projnome	Projnumero	Projlocal	Dnum
ProdutoX	1	Santo André	5
ProdutoY	2	Itu	5
ProdutoZ	3	São Paulo	5
Informatização	10	Mauá	4
Reorganização	20	São Paulo	1
Novosbenefícios	30	Mauá	4

DEPENDENTE

Fcpf	Nome_dependente	Sexo	Datanasc	Parentesco
33344555587	Alicia	F	06-04-1986	Filha
33344555587	Tiago	M	26-10-1983	Filho
33344555587	Janaína	F	03-05-1958	Esposa
98765432168	Antonio	M	28-02-1942	Marido
12345678966	Michael	M	04-01-1988	Filho
12345678966	Alicia	F	30-12-1988	Filha

Operações do Modelo Relacional

As operações do modelo relacional podem ser divididas em duas categorias: **recuperações e modificações**.

Recuperações

- ◆ As recuperações podem ser especificadas por meio de operações da **Álgebra Relacional** – que provê fundamentos formais para as operações de consulta do modelo relacional
- ◆ A Álgebra Relacional é usada como uma base para implementar e otimizar consultas nos SGBDRs
- ◆ Seus conceitos estão implementados na linguagem de consulta padrão para os SGBDRs – a SQL (*Structured Query Language*)

Operações do Modelo Relacional

As operações do modelo relacional podem ser divididas em duas categorias: **recuperações e modificações**.

Modificações

- ◆ Compreendem 3 tipos de operações sobre relações: **inserção de tuplas, remoção de tuplas e alteração de valores de atributos de tuplas**
- ◆ Essas operações só são aplicadas quando as restrições de integridade especificadas no esquema do banco de dados relacional não são violadas
- ◆ A linguagem SQL também provê comandos para a realização de modificações

Violações Causadas por Operações

- ◆ Uma operação de inserção ou alteração pode violar qualquer um dos quatro tipos de restrições explícitas (de domínio, de NULL, de integridade de entidade, de chave, de integridade referencial)
- ◆ Uma operação de remoção pode causar apenas violação de integridade referencial
- ◆ Se uma operação violar uma ou mais restrições, a ação padrão do SGBDR é rejeitar a operação
 - Mas é possível definir tratamento automático para a violação de integridade referencial, evitando a rejeição da operação

FUNCIONARIO

Pnome	Minitial	Uname	Cpf	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
João	B	Silva	12345678966	09-01-1965	Rua das Flores, 751, São Paulo, SP	M	30.000	33344555587	5
Fernando	T	Wong	33344555587	08-12-1955	Rua da Lapa, 34, São Paulo, SP	M	40.000	88866555576	5
Aloie	J	Zelaya	99988777767	19-01-1968	Rua Souza Lima, 35, Curitiba, PR	F	25.000	98765432168	4
Jennifer	S	Souza	98765432168	20-06-1941	Av. Arthur de Lima, 54, Santo André, SP	F	43.000	88866555576	4
Ronaldo	K	Lima	66688444476	15-09-1962	Rua Rebouças, 65, Piracicaba, SP	M	38.000	33344555587	5
Joice	A	Leite	45345345376	31-07-1972	Av. Lucas Obes, 74, São Paulo, SP	F	25.000	33344555587	5
André	V	Pereira	98798798733	29-03-1969	Rua Timbira, 35, São Paulo, SP	M	25.000	98765432168	4
Jorge	E	Brito	88866555576	10-11-1937	Rua do Horto, 35, São Paulo, SP	M	55.000	NULL	1

Tuplas impactadas pela remoção do departamento de Pesquisa ou pela alteração do valor de sua chave primária

DEPARTAMENTO

Dnome	Dnumero	Cpf_gerente	Data_inicio_gerente
Pesquisa	6	33344555587	22-05-1988
Administração	4	98765432168	01-01-1995
Matriz	1	88866555576	19-06-1981

LOCALIZACAO_DEP

Dnumero	Dlocal
1	São Paulo
4	Mauá
5	Santo André
5	Itu
5	São Paulo

TRABALHA_EM

Fcpf	Pnr	Horas
12345678966	1	32,5
12345678966	2	7,5
66688444476	3	40,0
45345345376	1	20,0
45345345376	2	20,0
33344555587	2	10,0
33344555587	3	10,0
33344555587	10	10,0
33344555587	20	10,0
99988777767	30	30,0
99988777767	10	10,0
98798798733	10	35,0
98798798733	30	5,0
98765432168	30	20,0
98765432168	20	15,0

PROJETO

Projnome	Projnumero	Projlocal	Dnum
ProdutoX	1	Santo André	5
ProdutoY	2	Itu	5
ProdutoZ	3	São Paulo	5
Informatização	10	Mauá	4
Reorganização	20	São Paulo	1
Novosbenefícios	30	Mauá	4

DEPENDENTE

Fcpf	Nome_dependente	Sexo	Datanasc	Parentesco
33344555587	Alicia	F	06-04-1986	Filha
33344555587	Tiago	M	25-10-1983	Filho
33344555587	Janaína	F	03-05-1958	Esposa
98765432168	Antonio	M	28-02-1942	Marido
12345678966	Michael	M	04-01-1988	Filho
12345678966	Alicia	F	30-12-1988	Filha

Violações Causadas por Operações

Opções disponíveis para o tratamento de uma violação de integridade referencial causada por uma operação de remoção de tupla ou de alteração do valor da chave primária de uma tupla:

- ◆ **Restrict** – rejeita a remoção/alteração (opção padrão nos SGBDRs)
- ◆ **Cascade** – propaga a remoção/alteração, excluindo/alterando também as tuplas que referenciam aquela que está sendo excluída
- ◆ **Set NULL** – atribui NULL como valor para a chave estrangeira nas tuplas que referenciam a tupla cuja remoção/alteração causa a violação, desfazendo os vínculos com ela
- ◆ **Set Default** – atribui um valor padrão para a chave estrangeira nas tuplas que referenciam a tupla cuja remoção/alteração causa a violação, desfazendo o vínculo com ela

Obrigada!

Prof^a Kelly Rosa Braghetto | linkedin.com/in/kelly-rosa-braghetto/

