



Fatec  
Indaiatuba

# MODELAGEM DE BANCO DE DADOS

*Profº Me. Jones Artur Gonçalves*

# CHAVES

Conceito básico para identificar linhas e estabelecer relações entre linhas de tabelas de um banco de dados relacional.



# CHAVES

**Chave-primária:** é uma coluna ou combinação de colunas cujos valores distinguem uma linha das demais dentro de uma tabela.



NrMatric	NmFunc	DtAdm	VrSalario	CdCargo	CdDepto
1001	JOAO SAMPAIO	10/08/93	750	C2	D2
1004	LUCIO TORRES	02/03/94	750	C2	D2
1034	ROBERTO PEREIRA	23/05/92	450	C3	D1
1021	JOSE NOGUEIRA	10/11/94	450	C3	D1

# CHAVES

**Chave-primária composta:** combinação de dois ou mais atributos que tenham a capacidade de se constituir uma chave primária.

CodEmp	NumDep	Nome	Tipo	DataNasc
E1	01	João	Filho	12/01/2001
E1	02	Maria	Filha	20/10/2003
E2	01	Ana	Esposa	12/12/1970

# CHAVES

## Chave estrangeira


- Coluna ou uma combinação de colunas, cujos valores aparecem necessariamente na chave primária de outra tabela.

Dept

CodigoDeppto	NomeDeppto
D1	Compras
D2	Engenharia
D3	Vendas

Emp

CodigoEmp	Nome	CodigoDeppto	CategFuncional	CIC
E1	Souza	D1	-	132.121.331-20
E2	Santos	D2	C5	891.221.111-11
E3	Silva	D2	C5	341.511.775-45
E5	Soares	D1	C2	631.692.754-88



Chave estrangeira

NrMatric	NmFunc	DtAdm	VrSalario	CdCargo	CdDepto
1001	JOAO SAMPAIO	10/08/1993	750	C2	D2
1004	LUCIO TORRES	02/03/1994	750	C2	D2
1034	ROBERTO PEREIRA	23/05/1992	450	C3	D1
1021	JOSE NOGUEIRA	10/11/1994	450	C3	D1
1029	RUTH DE SOUZA	05/01/1992	350	C1	D3
1095	MARIA DA SILVA	03/09/1992	600	C4	D1
1023	LUIZ DE ALMEIDA	12/01/1993	750	C2	D2
1042	PEDRO PINHEIRO	29/01/1994	600	C4	D1
1048	ANA SILVEIRA	01/06/1993	2300	C5	D1
1015	PAULO RODRIGUES	17/08/1992	750	C2	D2

CdCargo	NmCargo
C1	COZINHEIRA
C3	AUX. ESCRITORIO
C7	VIGIA
C2	MECANICO
C5	GERENTE
C4	ESCRITURARIO

CdDepto	NmDepto
D1	ADMINISTRACAO
D2	OFICINA
D3	SERVICOS GERAIS
D4	VENDAS

# CHAVES

## Chave estrangeira

- Restrições que devem ser garantidas em diversas situações de alteração do banco de dados:
  - Quando da inclusão de uma linha na tabela que contém a chave estrangeira
  - Quando da alteração do valor da chave estrangeira
  - Quando da exclusão de uma linha da tabela que contém a chave primária referenciada pela chave estrangeira
  - Quando da alteração do valor da chave primária referenciada pela chave estrangeira



# CHAVES

## Chave estrangeira

- Nem sempre a chave estrangeira referencia uma chave primária de outra tabela.
- Uma chave primária pode referenciar a chave primária da própria tabela.

Emp

CódigoEmp	Nome	CodigoDepto	CodigoEmpGerente
E5	Souza	D1	—
E3	Santos	D2	E5
E2	Silva	D1	E5
E1	Soares	D1	E1

chave estrangeira:  
referencia a chave primária  
da própria tabela



# NULO

**Nulo:** Representa um valor desconhecido para um dado.

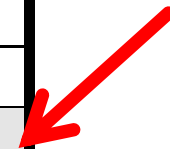
→ Nulo  $\neq$  espaços em branco e número zero que são valores conhecidos.

→ Nulo é a ausência de informação.

Uma coluna de preenchimento obrigatório numa tabela deve possuir todos os seus valores não-nulos

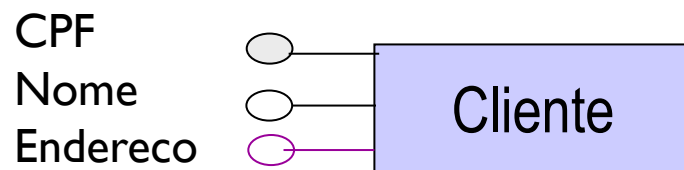
**NULO**

NrMatric	NmAluno	(...)	Sexo	RG	Reservista
22941010	CLAUDIO SILVA	(...)	M	21212121	91929
22941012	ANA RIBEIRO	(...)	F	14141414	?! ?! ?! ?!
(...)	(...)	(...)	(...)	(...)	(...)
22941088	JOSE SANTOS	(...)	M	34343434	828122

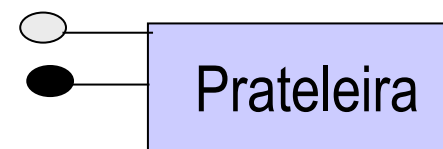


# IDENTIFICANDO ENTIDADES

- Cada entidade deve ter um identificador
- Identificador (também conhecido como chave):
  - É o conjunto de um ou mais atributos ou relacionamentos cujos valores servem para distinguir uma ocorrência da entidade das demais ocorrências da mesma entidade
  - Exemplo: os atributos CPF ou Carteira de Identidade identificam UNICAMENTE um cidadão brasileiro
- Representação no Modelo



NumeroCorredor  
NumeroPrateleira



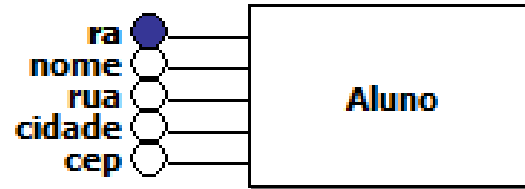
# ENTIDADE

Cada tabela que representa uma entidade do modelo conceitual pode ser classificada em: **Entidade Forte** ou **Entidade Fraca**. Conforme Cougo (1997), esta distinção se dá com a análise de existência de duas condições básicas: dependência de existência ou dependência de identificador. Uma entidade é fraca se um desses dois tipos de dependência existir no relacionamento entre duas entidades.

# TIPOS DE ENTIDADE

## Entidades fortes e fracas

- Uma entidade forte possui chave.

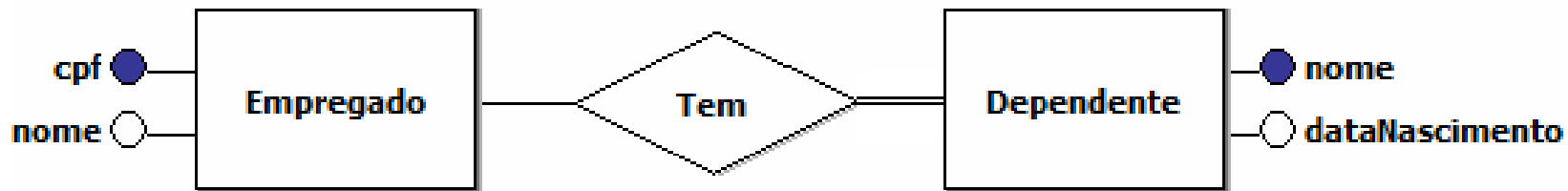


Entidade Forte:  
Aluno(**ra**, nome, rua, cidade, cep)

# TIPOS DE ENTIDADE

## Entidades fortes e fracas

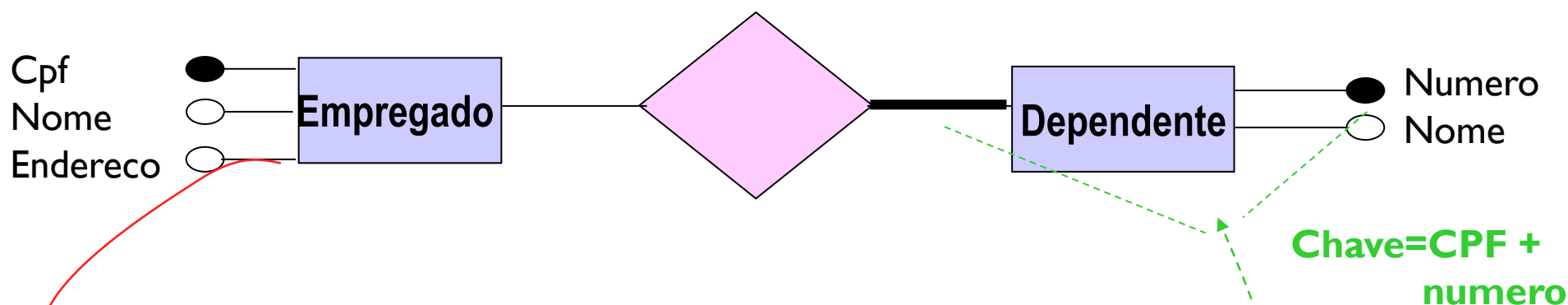
- Uma entidade fraca não possui um atributo chave, tem apenas uma chave parcial e deve participar de um relacionamento com uma entidade forte. A chave da entidade forte é incluída na relação como chave estrangeira.



Entidade Fraca:  
Dependente(**cpf**, nome, dataNascimento)

# IDENTIFICANDO RELACIONAMENTOS

- Quando parte da chave é um relacionamento
  - Exemplo: CPF do Empregado e numero sequencial na entidade Dependente



**Entidade Fraca:** entidade sem chave

A entidade é identificada por algum atributo da entidade + algum relacionamento

**Entidade Forte:** entidade com chave

A entidade é identificada por atributos da própria entidade



# IDENTIFICANDO ENTIDADES

- **Exemplos de Entidades:**
- **Sistema Bancário**
  - Cliente
  - Conta Corrente
  - Conta Poupança
  - Agência
- **Sistema de Controle de Produção de Industria**
  - Produto
  - Empregado
  - Departamento
  - Estoque

## TIPOS DE RELACIONAMENTOS

Um relacionamento descreve uma associação entre entidades como afirma Coronel e Rob (2011).

Os relacionamentos envolvendo tabelas fracas resultam em uma **tabela associativa** e que deve ser representada por meio de um losango com bordas duplas.

## CONCEITOS DO MODELO ER

### ENTIDADE ASSOCIATIVA

A entidade associativa surge de um relacionamento de  $n$  para  $n$ , em que existe uma associação dos atributos identificadores das duas entidades relacionadas, caracterizando uma nova entidade.

A nova entidade gerada possui, normalmente, atributos próprios do relacionamento, isto é, ela só existe por causa do relacionamento.

# CONCEITOS DO MODELO ER

## ENTIDADE ASSOCIATIVA

- **Entidade Associativa:** Permite associar entidades a relacionamentos
- Exemplo: Como associar Medicamentos prescritos em uma Consulta?



?

Como associar?

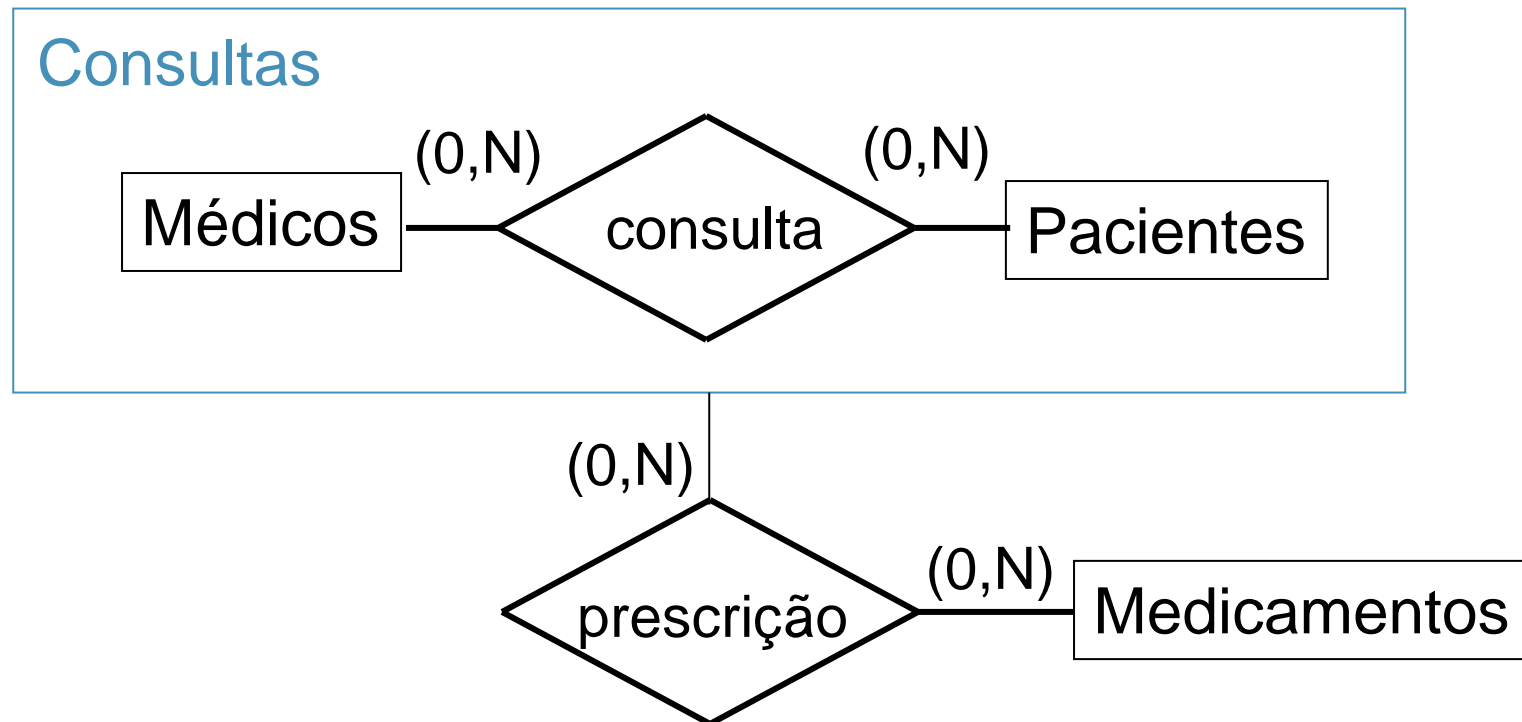
Medicamentos

# CONCEITOS DO MODELO ER

## ENTIDADE ASSOCIATIVA

### ■ Entidade Associativa

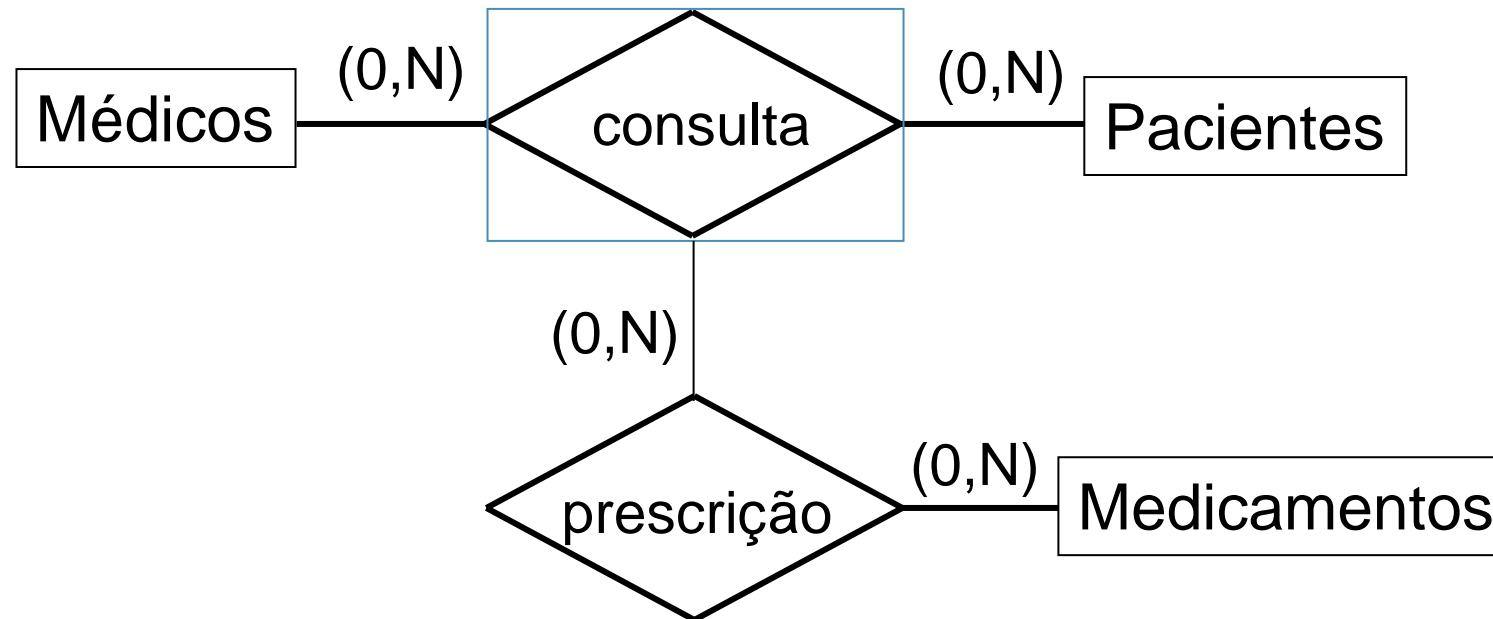
- Solução: tornar consulta uma entidade associativa
- Uma entidade associativa encapsula uma associação entre entidades



# CONCEITOS DO MODELO ER

## ENTIDADE ASSOCIATIVA

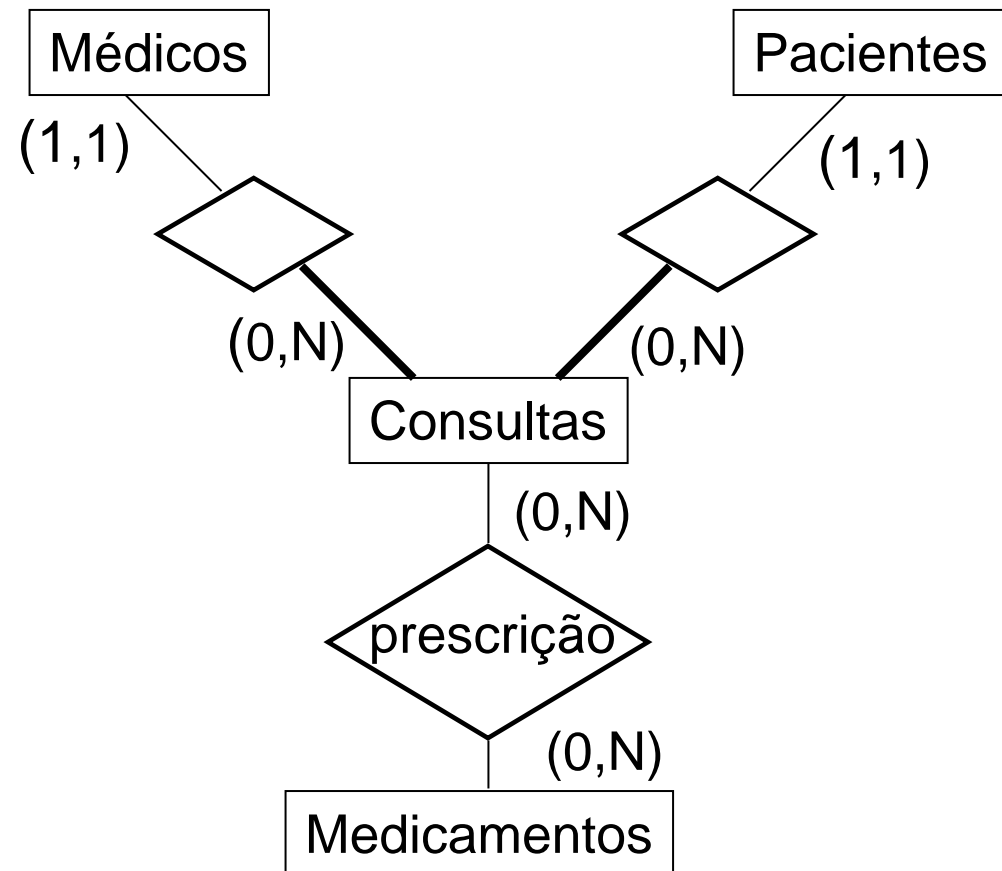
- Entidade Associativa
  - Outra forma de **representar**



# CONCEITOS DO MODELO ER

## ENTIDADE ASSOCIATIVA

- **Entidade Associativa**
  - Outra forma de **modelar**



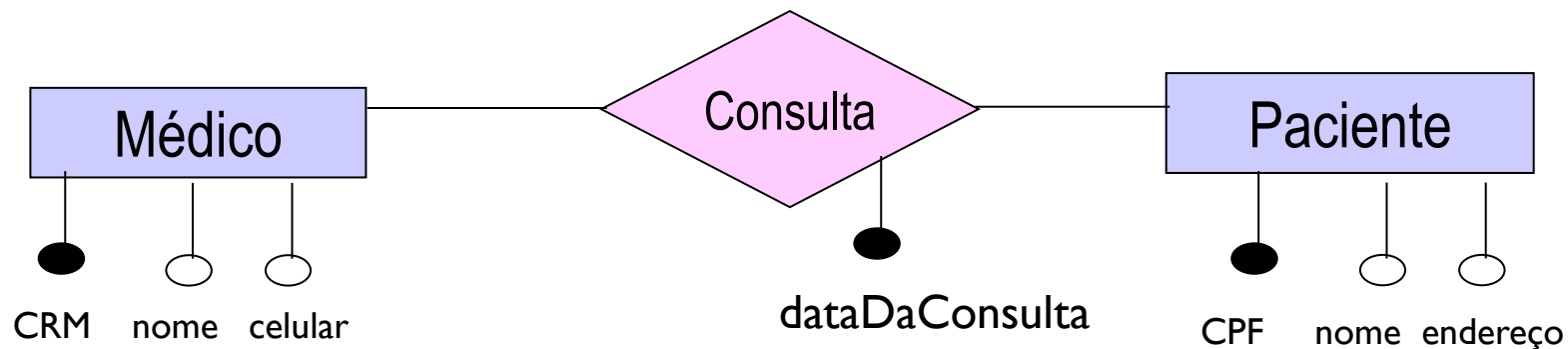


# CONCEITOS DO MODELO ER

## ENTIDADE ASSOCIATIVA

Uma consulta é realizada em uma data e em um horário; possui um preço; pode ser paga por convênio ou pelo paciente; apresenta uma prescrição do médico e a relação de medicamentos.

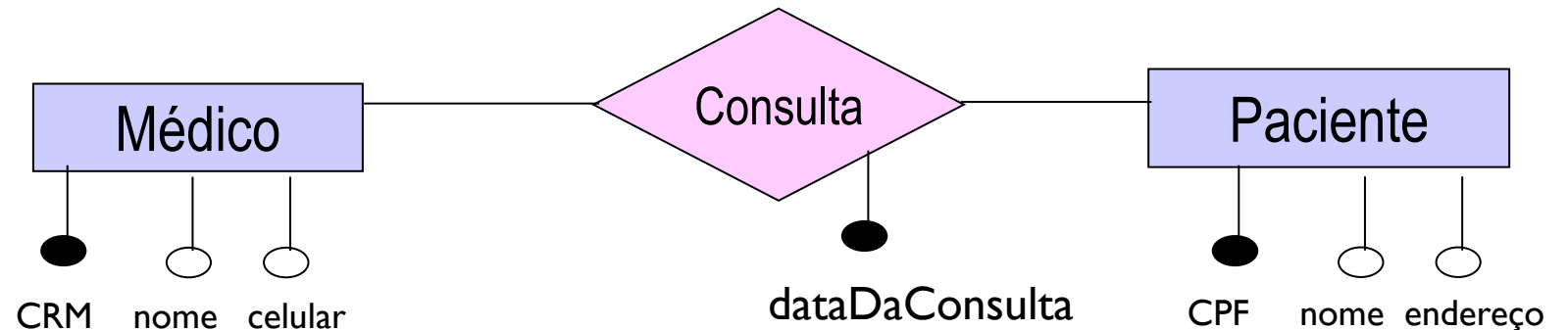
Esses são alguns atributos que pertencem apenas ao relacionamento CONSULTA.



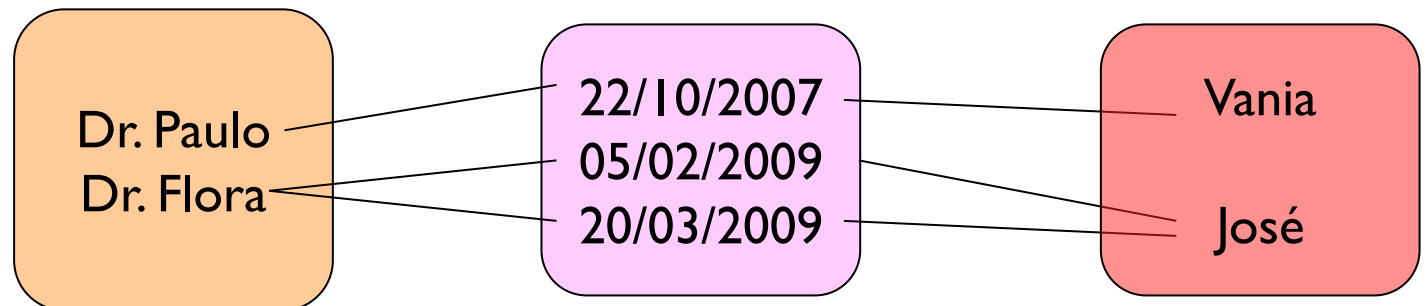
# CONCEITOS DO MODELO ER

## ENTIDADE ASSOCIATIVA

Um médico pode consultar  
n pacientes.



Um paciente pode ser  
consultado por n  
médicos.



## TIPOS DE RELACIONAMENTOS

Quando temos um relacionamento entre duas entidades, o número de ocorrências de uma entidade que está associada, com ocorrências de outra entidade, determina o **grau de relacionamento** ou **cardinalidade** entre as tabelas, como afirma Abreu e Machado (2009).

# TIPOS DE RELACIONAMENTOS

## **Relacionamento unário**

Uma entidade se relaciona com ela mesma.

## **Relacionamento binário**

Liga dois tipos diferentes de entidades. É o evento mais comum

## **Relacionamento ternário**

Três entidades estão conectadas por um mesmo relacionamento.

## **Relacionamento quaternário**

Quatro tabelas estão conectadas.

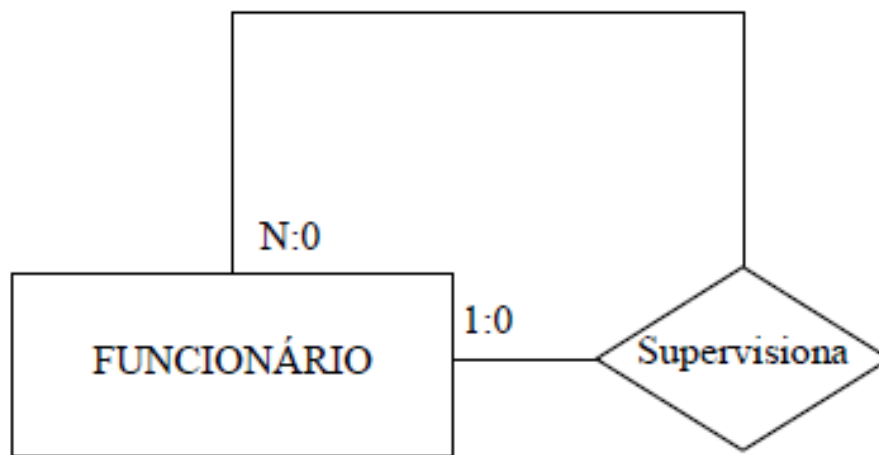
## **Relacionamento n-ário**

Acima de quatro tabelas envolvidas.

# AUTORRELACIONAMENTO - UNÁRIO

## RELACIONAMENTOS REFLEXIVOS (AUTORELACIONAMENTOS)

Elementos de uma mesma entidade relacionam-se entre si devido ao fato de apresentarem uma estrutura de natureza hierárquica.



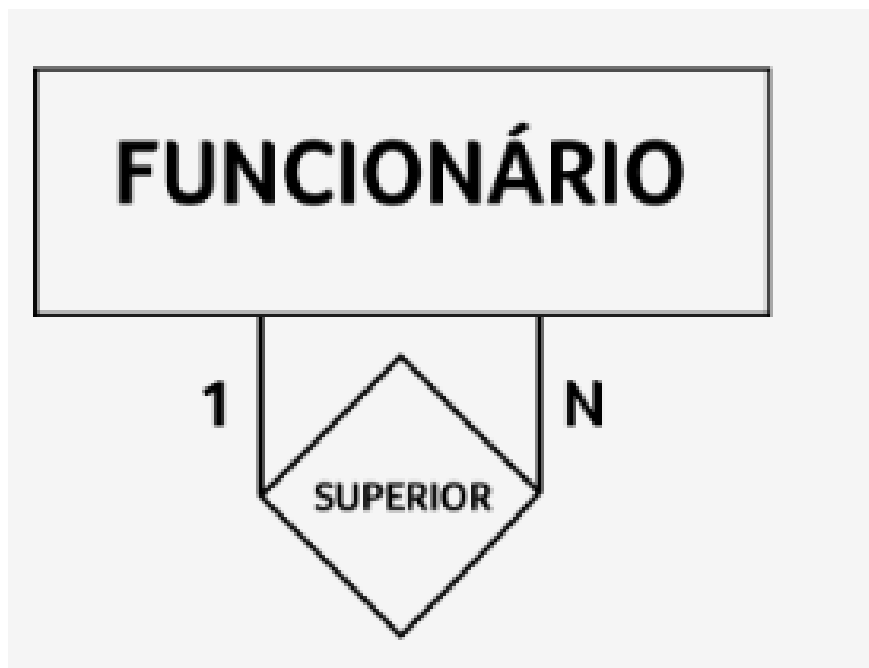
Exemplo de relacionamento reflexivo entre funcionários.

# AUTORRELACIONAMENTO - UNÁRIO

Nome	Cargo	Depto	Supervisor
João Silva	Pedreiro	D01	Roberto Rodrigues
Roberto Rodrigues	Porteiro	D01	Não tem
Ivo Pereira	Pedreiro	D03	Cláudio Silva
Joel Nepomuceno	Desenhista	D02	Júlia Santos
Cláudio Silva	Contador	D03	Não tem
André Paiva	Gerente	D03	Não tem
Júlia Santos	Projetista	D02	Não tem
Ricardo Alves	Porteiro	D02	Júlia Santos
Ana Garcia	Escriturário	D03	Cláudio Silva
Sílvio Coelho	Desenhista	D03	André Paiva

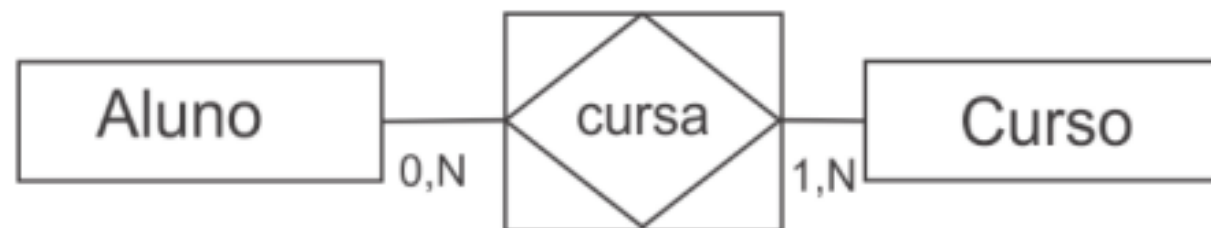
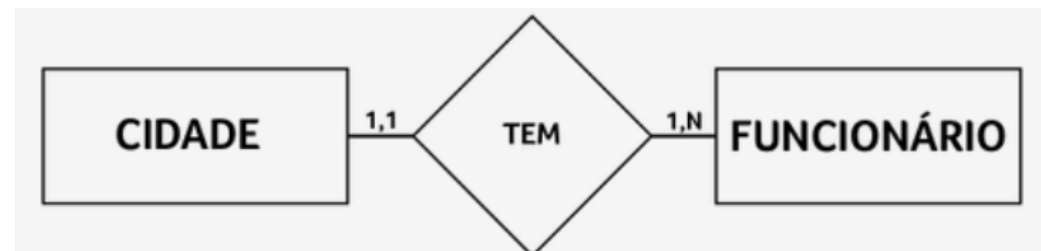
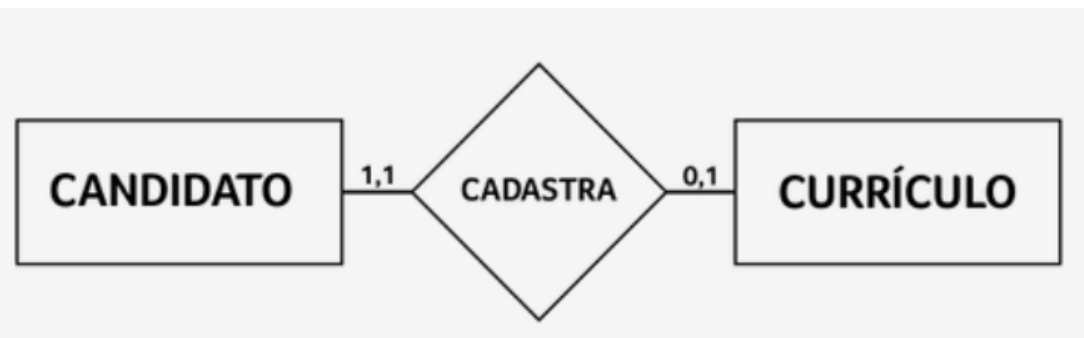
Detalhes sobre os funcionários de uma empresa.

# AUTORRELACIONAMENTO - UNÁRIO

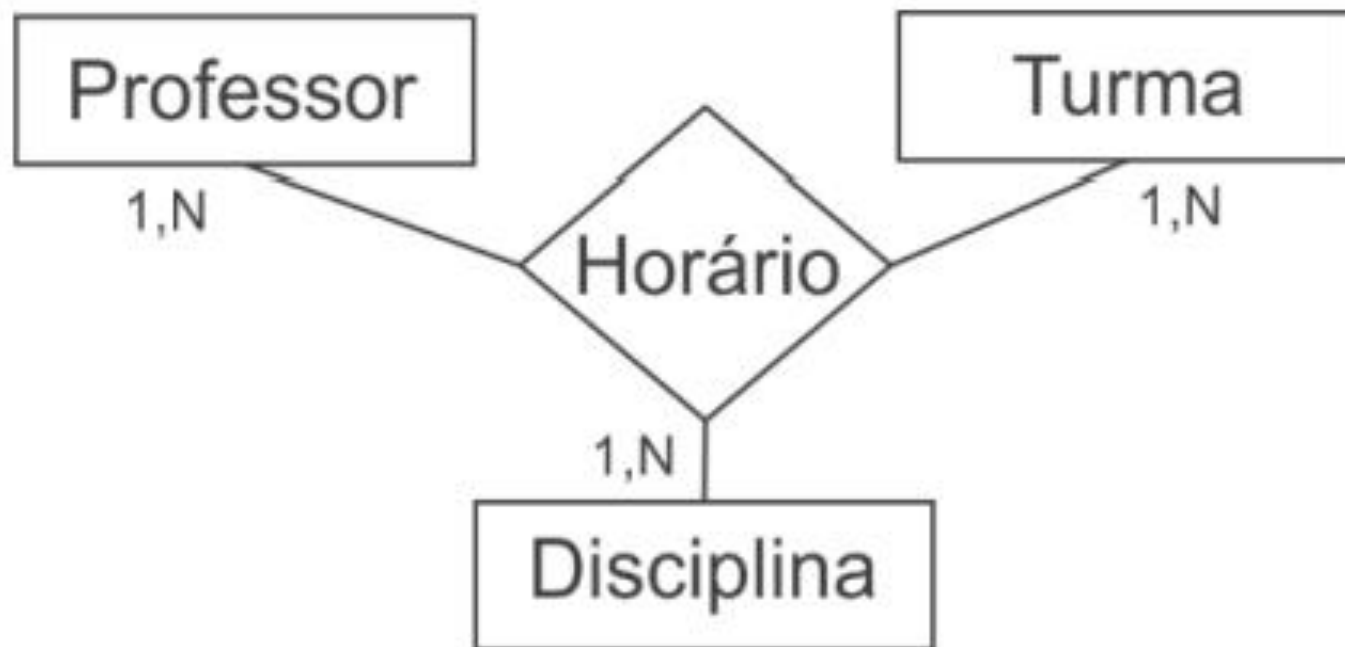




# BINÁRIO



# TERNÁRIO



# QUATERNÁRIO



# INTRODUÇÃO

## **Projeto Lógico:**

- fase posterior à modelagem conceitual do banco de dados.
- dependente da tecnologia que será utilizada.
- importante saber qual banco de dados será utilizado.

# INTRODUÇÃO

## **Projeto Físico:**

- detalhamento dos dispositivos e recursos físicos a serem alocados para o projeto, tais como clusters, discos, etc.
- dependente da tecnologia que será utilizada.
- importante saber qual banco de dados será utilizado.

# REGRAS DE INTEGRIDADE

## Composição de um BD relacional

- Domínios
  - Para cada coluna da tabela deve ser especificado um conjunto de valores (numérico, alfanumérico...)
- Valores Vazios (null)
  - As colunas nas quais não são admitidos valores vazios são chamadas de colunas *obrigatórias*
    - Normalmente, as colunas que compõem uma chave primária devem ser obrigatórias
  - *As colunas nas quais podem aparecer campos vazios são chamadas de colunas opcionais*

## REGRAS DE INTEGRIDADE

São regras fundamentais para a manutenção da consistência do banco de dados.

Temos dois tipos de regras de integridade:

- Integridade de Entidade
- Integridade Referencial



## REGRAS DE INTEGRIDADE

### ➤ **Integridade de Entidade (ou de Identidade ou Existencial) :**

Refere-se às chaves primárias e procura garantir que toda e qualquer linha de uma tabela pode ser acessada com base apenas no conteúdo de sua chave primária.

Regra básica:

*"Nenhum atributo que faça parte de uma chave primária pode ter valor nulo."*

# REGRAS DE INTEGRIDADE

## ➤ **Integridade Referencial**

Diz respeito às chaves estrangeiras e visa manter a integridade dos relacionamentos previstos na base de dados.

## REGRAS DE INTEGRIDADE

→ **Tabela-Pai** (*Parent Table*): é aquela onde o atributo de relacionamento desempenha o papel de chave primária.

→ **Tabela-Filho** (*Dependent Table*): tabela onde o atributo de relacionamento desempenha o papel de chave estrangeira.

Regra básica:

"O conteúdo de uma chave estrangeira deve, necessariamente, ser igual ao de uma ocorrência da Tabela-Pai ou então ser nulo."

## REGRAS DE INTEGRIDADE

<i>Operação</i>	<i>Tabela-Pai</i>	<i>Tabela-Filho</i>
Inclusão	Não afeta.	Não pode incluir se não tiver na Tabela-Pai
Alteração	Se envolver a chave primária, utilizar um dos seguintes critérios: não alterar se tiver dependentes; alterar e colocar nulos nas chaves estrangeiras de todos os dependentes da Tabela-Filho; alterar e colocar o novo valor nas chaves estrangeiras de todos os dependentes na Tabela-Filho.	Se envolver a chave estrangeira, só aceitar valores que existam na Tabela-Pai
Exclusão	Utilizar um destes 3 critérios: não deletar se tiver dependentes; deletar e colocar nulos nas chaves estrangeiras da Tabela-Filho; deletar e também eliminar todos os dependentes.	Não afeta.

## REGRAS DE INTEGRIDADE

**Obrigatoriedade:** indica se deve ou não ser permitida a existência de nulos em uma coluna.

**Unicidade:** indica se deve ser permitido ou não que uma coluna possua valores idênticos em duas ou mais linhas. É uma coluna de valores únicos.

**Verificação de valores específicos:** indica explicitamente qual o conjunto de valores permitidos para uma coluna. Por exemplo, para a coluna Sexo da tabela de alunos, os valores permitidos são { 'M', 'F' }.

# TRANSFORMAÇÃO ER PARA RELACIONAL

Existem algumas regras para transformação de um modelo ER em um modelo Relacional que foram definidas tendo dois objetivos básicos:

→ Obter banco de dados com boa performance de instruções de consulta e alteração do banco.

→ Obter um banco de dados que simplifique o desenvolvimento e manutenção de aplicações.

# TRANSFORMAÇÃO ER PARA RELACIONAL

## Atribuição de Nomes

Tabelas e colunas devem receber nomes próprios.

→ Nem sempre conseguimos usar no esquema das tabelas os mesmos nomes utilizados no modelo E/R

# TRANSFORMAÇÃO ER PARA RELACIONAL

<i>Padrão</i>	<i>SQL Ansi</i>	<i>DBF</i>	<i>Paradox</i>	<i>MS-Access</i>	<i>Oracle</i>	<i>MS-SQL Server</i>	<i>Interbase</i>	<i>MySQL</i>
<i>Aspecto</i>	<i>92</i>	<i>III+</i>	<i>7</i>	<i>2000</i>	<i>8</i>	<i>7</i>	<i>4.2</i>	<i>3.23</i>
Tam. nome tabela	128	***	***	64	30	128	31	64
Tam. nome coluna	128	10	25	64	30	128	31	64
Caracteres permitidos	Letras, números e underscore.	Letras, números e alguns caracteres especiais.  P/ tabelas: ***	P/ colunas, tudo exceto: vírgula; pipe; exclamação.  P/ tabelas: ***	Tudo exceto: ponto dec; exclamação; acento grave; colchetes.	Letras, números, underscore, cifrão e number (#).	Letras, números, underscore, cifrão e number (#).	Letras, números, underscore e cifrão.	Tudo, exceto: ponto dec.  P/ tabelas tb não permite usar contra-barras.
Obs.	Primeiro caracter deve ser uma letra.	Espaços em branco e pontuações não são permitidos	Primeiro caracter não pode ser um espaço	Primeiro caracter não pode ser um espaço	Primeiro caracter deve ser uma letra.	Primeiro caracter deve ser letra ou underscore.	Primeiro caracter deve ser uma letra.	Nomes não podem ser compostos apenas por números.

\*\*\* Varia conforme o sistema de arquivos do sistema operacional



# TRANSFORMAÇÃO ER PARA RELACIONAL

## Tipos de Dados

O detalhamento de uma tabela deve apresentar todas as colunas, juntamente com seu tipo de dados e, quando necessário, o tamanho correspondente.

<i>Padrão</i>	<i>SQL Ansi</i>	<i>DBF</i>	<i>Paradox</i>	<i>MS-Access</i>	<i>Oracle</i>	<i>MS-SQL Server</i>	<i>Interbase</i>	<i>MySQL</i>
<i>Tipo</i>	<i>92</i>	<i>III+</i>	<i>7</i>	<i>2000</i>	<i>8</i>	<i>7</i>	<i>4.2</i>	<i>3.23</i>
Texto	Char Varchar	Character	Alpha	Text Memo	Char Varchar2 Long	Char Varchar	Char Varchar	Char Varchar
Data	Date	Date	Date	Datetime	Date	Datetime	Date	Datetime
Hora	Time	-	Time	Datetime	Date	Datetime	Date	Datetime
Número	Numeric Integer Float	Numeric	- Long Integer Number	- Long Double	Number	Numeric Int Real	Numeric Integer Float	Numeric Integer Double
Moeda	-	-	Money	Currency	-	Money	-	-
Lógico	-	Logic	Logical	Bit	-	Bit	-	-
Contador	-	-	Autoincrement	Counter	-	-	-	-
Binários	-	-	Bynary	LongBinary	BLOB	Varbinary	BLOB	BLOB

Tabela 4: Principais tipos de dados suportados por alguns SGBDs relacionais.

# TRANSFORMAÇÃO ER PARA RELACIONAL

## Tipos de Dados

Tipos de dados	Descrição
<b>char(n)</b>	Sequência de caracteres de tamanho fixo. Máximo 8000 caracteres
<b>varchar(n)</b>	Sequência de caracteres de tamanho variável. Máximo 8000 caracteres
<b>varchar(max)</b>	Sequência de caracteres de tamanho variável. Máximo 1.073.741.824 caracteres
<b>text</b>	Sequência de caracteres de tamanho variável. Máximo 2GB de dados de texto

### Unicode strings:

Tipos de dados	Descrição
<b>nchar(n)</b>	Dados Unicode de comprimento fixo. Máximo 4000 caracteres
<b>nvarchar(n)</b>	Dados Unicode de comprimento variável. Máximo 4000 caracteres
<b>nvarchar(max)</b>	Dados Unicode de comprimento variável. Máximo 536.870.912 caracteres
<b>ntext</b>	Dados Unicode de comprimento variável. Máximo 2GB de dados de texto

## Tipos de Dados

### Qual a diferença entre VARCHAR e CHAR ???

CHAR e VARCHAR são tipos de dados caractere, a diferença é que **CHAR é um tipo de dado de comprimento fixo** e **VARCHAR é de comprimento variável**.

Usamos CHAR quando os tamanhos que desejamos armazenar na coluna de uma tabela são de tamanho consistentes e semelhantes. Exemplo: Número de telefone, CEP, CPF, CNPJ, etc. O tipo CHAR possui um tamanho fixo, assim se você tentar armazenar um valor maior que o definido numa coluna do tipo CHAR ele será truncado.

Se você definir uma coluna da tabela (campo) como **CHAR(10)** e armazenar um **caractere apenas ele vai armazenar mais nove espaços em branco.** ( Por causa desta característica o tipo de dados CHAR é chamado de tipo de dados com tamanho fixo.)

## Tipos de Dados

### Qual a diferença entre VARCHAR e CHAR ???

Use VARCHAR quando os tamanhos a serem armazenados na coluna da tabela variam consideravelmente.

Ex: Endereço, Nomes, URL, etc. Dessa forma um valor menor irá ocupar menos espaço que um valor maior.

O tipo de dados VARCHAR armazena somente a quantidade de caracteres que foram definidos na sua criação.

Assim se você definir uma coluna da tabela (campo) como **VARCHAR(10)** e armazenar um caractere ele vai armazenar somente o caractere sem colocar espaços para completar o tamanho definido na criação.



## Tipos de Dados

### Quando usar **CHAR** ou **VARCHAR** ?

O tipo **CHAR** deve ser usado quando sabemos que todos os dados armazenados em determinada coluna não são variáveis como, por exemplo, uma coluna que armazena a sigla do estado ou o cep que sempre terão o mesmo tamanho.

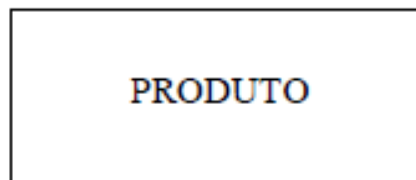
Já o **VARCHAR** deve ser utilizado quando não sabemos o que vamos armazenar. Um exemplo pode ser o nome do cliente, endereço, o e-mail que sempre variam de tamanho.

# TRANSFORMAÇÃO ER PARA RELACIONAL

## Tipos de Dados

Exemplo:

Entidade



@ Cód. Prod.  
Nome  
Preço Unit.  
Qtde Estoque  
Data última compra

Tabela de produto no Oracle

<i>Coluna</i>	<i>Tipo</i>	<i>Tamanho</i>
CdProd	Number	5, 0
NmProd	Varchar2	30
VrUnit	Number	10, 2
QtEstoque	Number	8, 4
DtUltCompra	Date	-

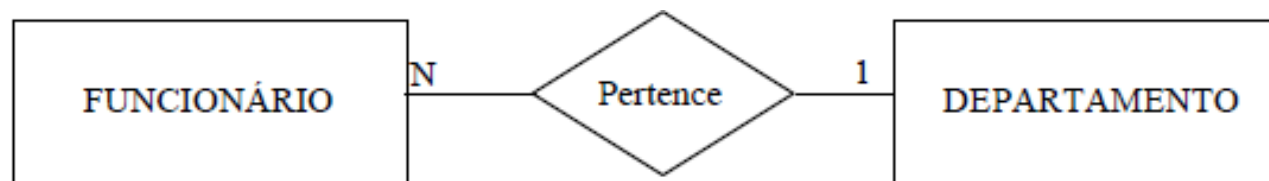
Tabela de produto no MS-SQL Server

<i>Coluna</i>	<i>Tipo</i>	<i>Tamanho</i>
CdProd	Numeric	5, 0
NmProd	Varchar	30
VrUnit	Money	-
QtEstoque	Numeric	8, 4
DtUltCompra	Datetime	-

# TRANSFORMAÇÃO ER PARA RELACIONAL

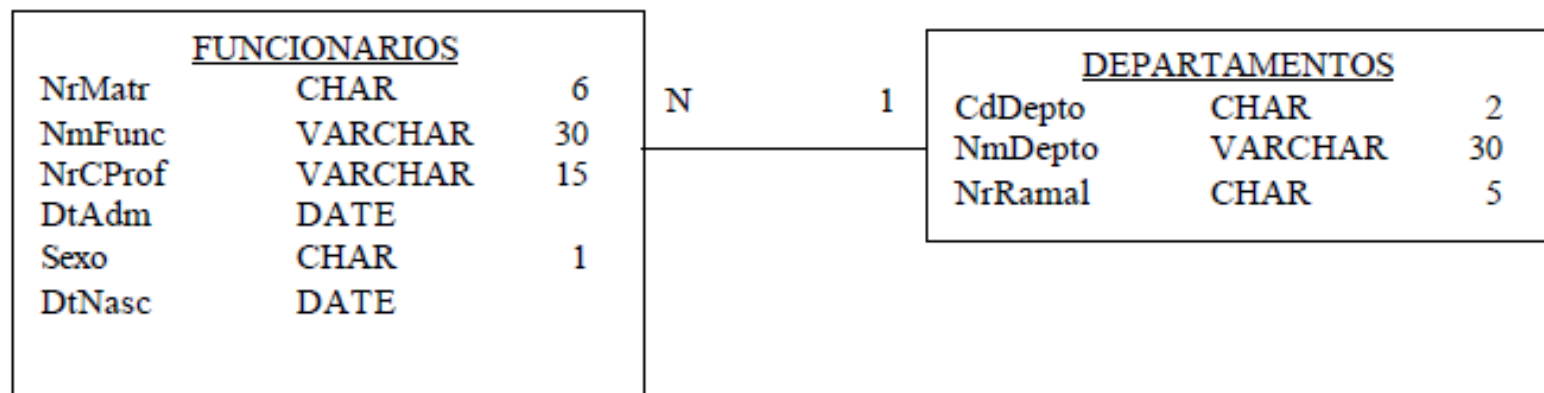
## Regras Básicas de Mapeamento

Analistem a estrutura ao lado e notem como a mesma ficará transformada no modelo lógico.



@ Número de Matrícula  
Nome completo  
Número da Carteira Prof.  
Data de Admissão  
Sexo  
Data de Nascimento

@ Código do Depto  
Nome completo  
Ramal telefônico



## TRANSFORMAÇÃO ER PARA RELACIONAL

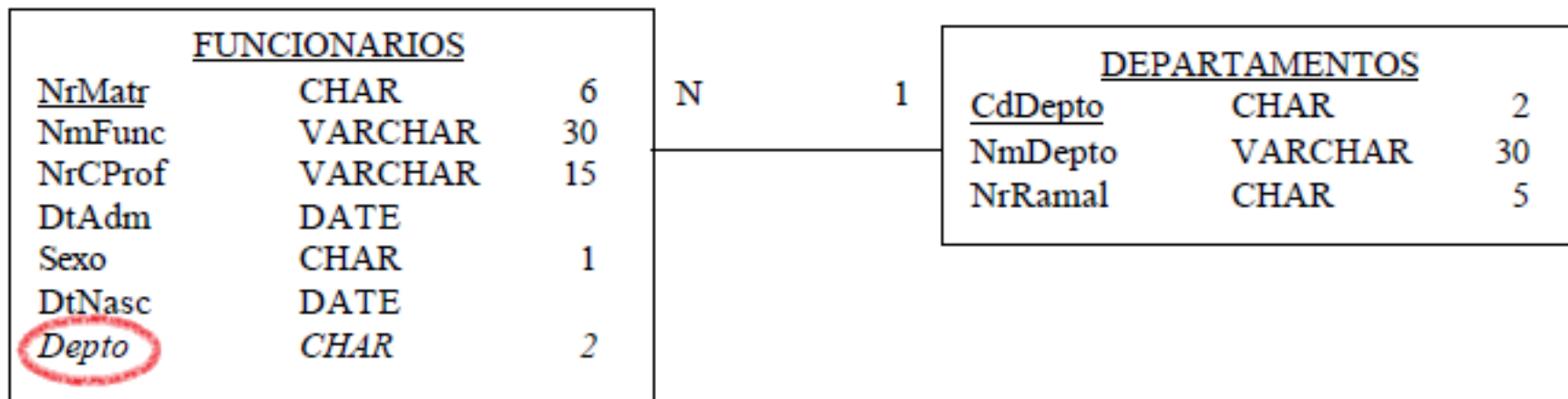
Analizando o exemplo anterior, como vincular a linha de um funcionário com a linha correspondente ao seu respectivo departamento, e vice-versa?

A solução é colocar na tabela Funcionarios uma nova coluna, do tipo CHAR com tamanho 2, destinada a conter o código do departamento ao qual o funcionário pertence.



# TRANSFORMAÇÃO ER PARA RELACIONAL

## Regras Básicas de Mapeamento



# TRANSFORMAÇÃO ER PARA RELACIONAL

## Análise de Relacionamentos

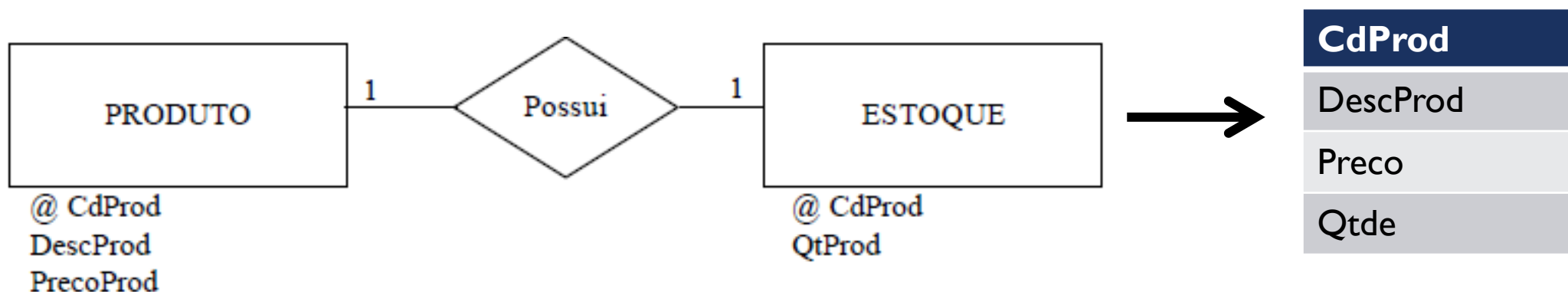
### *Relacionamentos 1 para 1*

Identificar se as duas entidades são realmente distintas ou se elas podem ser unidas em uma única tabela.

Se possuem o mesmo atributo identificador, há uma forte razão para representar essas entidades em uma tabela só, a menos que o relacionamento seja opcional com cardinalidade mínima zero de um lado e um de outro lado.

# TRANSFORMAÇÃO ER PARA RELACIONAL

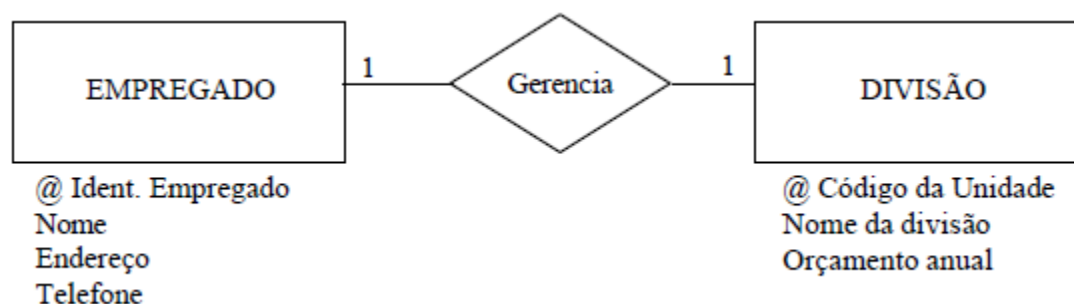
## *Relacionamentos 1 para 1*



Neste diagrama, observamos que as entidades PRODUTO e ESTOQUE possuem seguramente o mesmo identificador, logo podemos representar as informações todas em uma só tabela (PRODUTOS, no caso).

# TRANSFORMAÇÃO ER PARA RELACIONAL

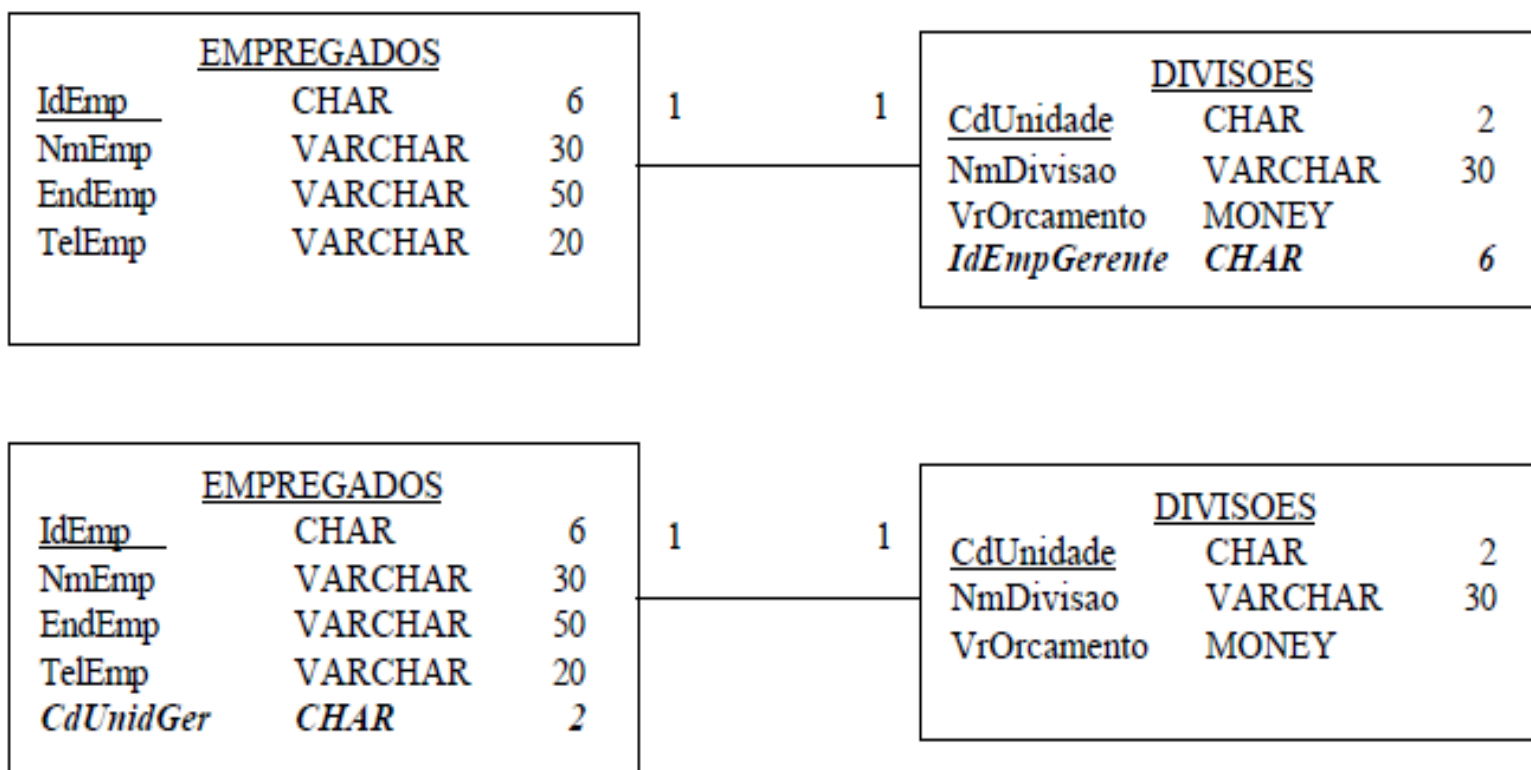
## Relacionamentos 1 para 1



- Se as duas entidades forem realmente distintas, o relacionamento se dará através de um elo explícito, ou seja, uma chave estrangeira.
- Ao projetar, devemos analisar a possibilidade de o relacionamento 1 para 1 futuramente tornar-se um relacionamento 1 para muitos.

# TRANSFORMAÇÃO ER PARA RELACIONAL

## Relacionamentos 1 para 1

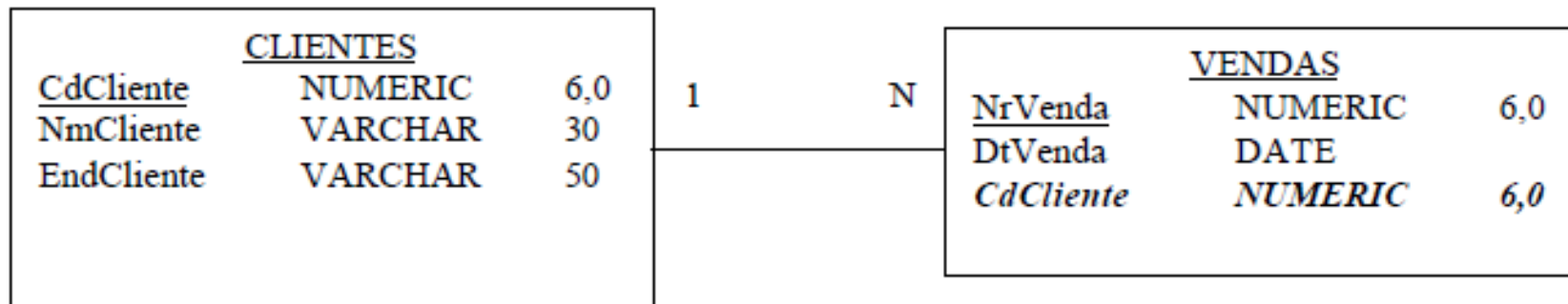
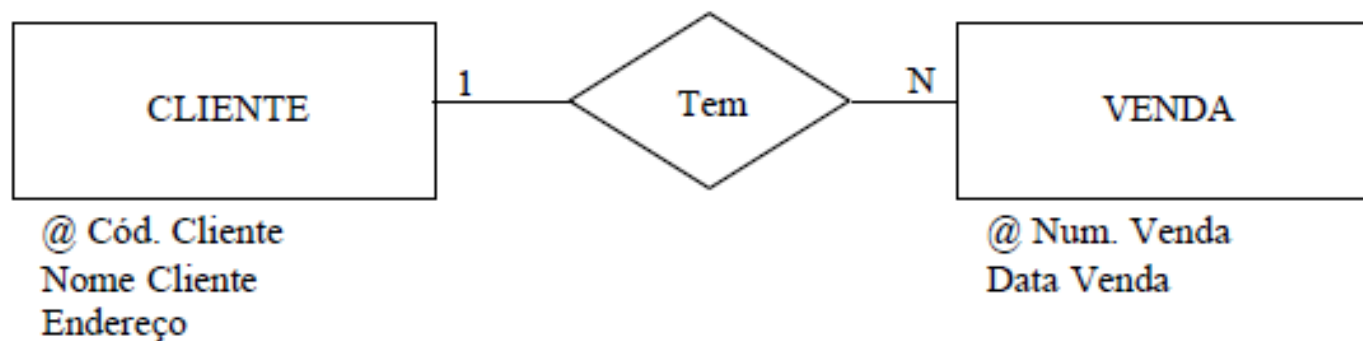


*Duas soluções possíveis para o relacionamento 1:1 entre Empregados e Divisões*

# TRANSFORMAÇÃO ER PARA RELACIONAL

## Relacionamentos 1 para muitos

A chave estrangeira deve ser colocada na tabela do lado MUITOS do relacionamento.



# TRANSFORMAÇÃO ER PARA RELACIONAL

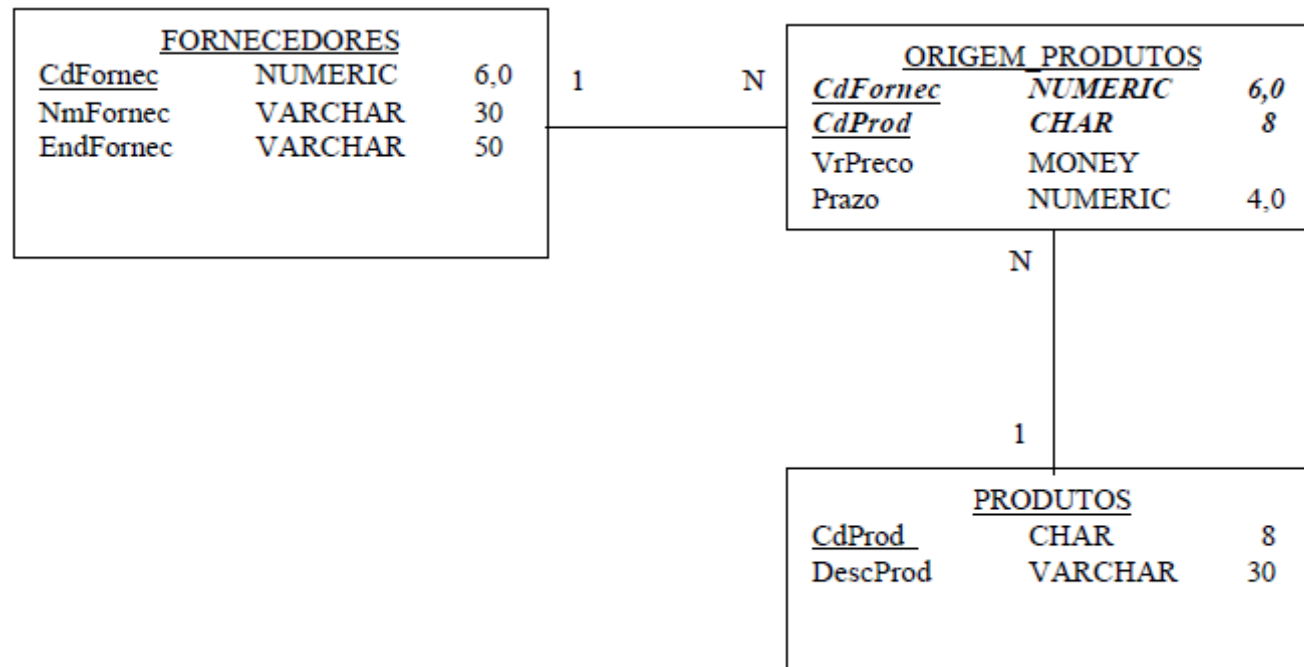
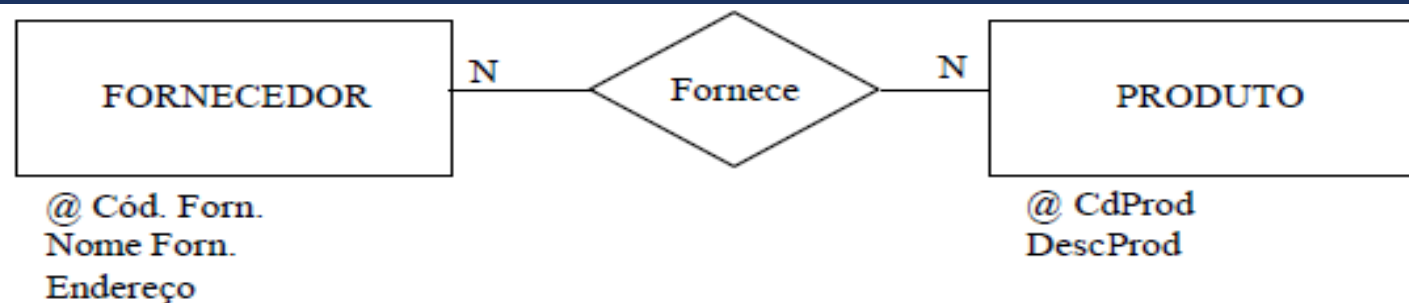
## *Relacionamentos muitos para muitos*

Desdobramento em 2 relacionamentos 1 para muitos e a criação de uma tabela intermediária que faça a associação entre as duas tabelas principais.

A chave primária dessa nova tabela é, em princípio, a concatenação ou combinação das chaves primárias das duas tabelas principais.

# TRANSFORMAÇÃO ER PARA RELACIONAL

*Relacionamentos  
muitos para muitos*





# DICIONÁRIO DE DADOS

Todo modelo de banco de dados produzido precisa ser documentado. Isso é importante por vários motivos: **organização, apresentação e controle.**

Após a elaboração do DER, deve ser realizado o mapeamento das entidades, juntamente com a documentação completa das tabelas, neste caso, um **dicionário de dados**.

É necessário criar o **dicionário de dados** para estabelecer uma padronização e uma documentação sobre cada tabela criada.

# DICIONÁRIO DE DADOS

TABELA: Funcionário				
	CAMPO	DESCRIÇÃO	TIPO	TAMANHO
PK	Cd_Func	Código do Funcionário	VARCHAR	20
	Nm_Func	Nome do Funcionário	VARCHAR	100
	CPF_Func	CPF do Funcionário	VARCHAR	15
	Dt_Nasc_Func	Data de Nascimento Funcionário	Date	-
FK	Id_Cidade	Cidade do Funcionário	Inteiro	-

# DICIONÁRIO DE DADOS

Um dicionário de dados deve ter várias informações, como:

- Descrição dos nomes de: tabelas, relações e atributos.
- Tipos dos dados (seu domínio) e seus respectivos tamanhos.
- Descrição detalhada das chaves utilizadas.
- Nomes dos usuários com suas permissões sobre a tabela.

# DICIONÁRIO DE DADOS

O nível de detalhamento do dicionário de dados é opcional, mas ele acaba se tornando um **documento fundamental**, sendo muito requisitado quando o banco de dados apresenta problemas e precisa sofrer manutenção.

Tabela: funcionário					
Descrição:		Tabela responsável por armazenar as informações dos funcionários da empresa.			
Volume de dados:		Carga inicial de 140 registros e volume mensal estimado em 25% de acréscimo.			
Tempo de retenção:		Permanente.			
Permissões:		Leitura e cravação: funcionário RH nível A – leitura, gravação e alteração – nível A5			
Atributos					
Atributo	Campo	Tipo de dado	Tamanho	Descrição	Restrição
Código	Cd_Func	VARCHAR	20	Código do funcionário	Chave primária
Nome	Nm_Func	VARCHAR	100	Nome do funcionário	Nome completo
CPF	CPF_Func	VARCHAR	15	CPF do funcionário	CPF válido
Data Nasc	Dt_Nasc_Func	Date	-	Data de nascimento funcionário	Data formato dd/mm/aaaa
Cidade	Id_Cidade	Inteiro	-	Cidade do funcionário	Chave estrangeira da tabela cidade obrigatória

# BIBLIOGRAFIA

## BÁSICA:

DATE, C. J. PROJETO DE BANCO DE DADOS E TEORIA RELACIONAL: FORMAS NORMAIS E TUDO O MAIS. SÃO PAULO: NOVATEC, 2015.

ELMASRI, R.; NAVATHE, S. B. SISTEMAS DE BANCO DE DADOS: FUNDAMENTOS E APLICAÇÕES. 7 ED. SÃO PAULO: PEARSON, 2019.

HEUSER, C. A. PROJETO DE BANCO DE DADOS. 6 ED. PORTO ALEGRE: BOOKMAN, 2010.



## COMPLEMENTAR:

HARRINGTON, J. L. Projeto de Bancos de Dados Relacionais: Teoria e Prática. São Paulo: Campus, 2002.

MACHADO, F. N. R., Banco de dados: projeto e implementação. 2 ed. São Paulo: Érica, 2008.

NADEAU, Tom et al. Projeto e Modelagem de Banco de Dados. 5 ed. Rio de Janeiro: Elsevier Brasil, 2013.

SILBERSCHATZ, Abraham; SUNDARSHAN, S.; KORTH, Henry F. Sistema de banco de dados. Rio de Janeiro: Elsevier Brasil, 2016.

# Referências



O.K. TAKAI; I.C.ITALIANO; J.E. FERREIRA, INTRODUÇÃO A BANCO DE DADOS

OSVALDO KOTARO, APOSTILA, DCC-IME-USP – FEVEREIRO - 2005

MATTOSO, MARTA, INTRODUÇÃO À BANCO DE DADOS – AULA

BANCO DE DADOS BÁSICO, UNICAMP, CENTRO DE COMPUTAÇÃO, SLIDES.

BOGORNÝ VANIA, MODELO ENTIDADE-RELACIONAMENTO, SLIDES.

[WWW.JOINVILLE.UDESC.BR/PORTAL/PROFESSORES/MAIA/.../6\\_\\_\\_MODELO\\_ER.PPT](http://WWW.JOINVILLE.UDESC.BR/PORTAL/PROFESSORES/MAIA/.../6___MODELO_ER.PPT) DATA DE ACESSO: 01/07/2015

ABREU, FELIPE MACHADO; ABREU, MAURÍCIO – PROJETO DE BANCO DE DADOS – UMA VISÃO PRÁTICA - ED. ÉRICA – SÃO PAULO

HEUSER, CARLOS ALBERTO. PROJETO DE BANCO DE DADOS – UMA VISÃO PRÁTICA. PORTO ALEGRE: SAGRA LUZATTO, 2004.

KORTH, H. F.; SUDARSHAN, S; SILBERSCHATZ, A. SISTEMA DE BANCO DE DADOS. 5A ED. EDITORA CAMPUS, 2006. - CAPÍTULO 6

REIS, FÁBIO. [HTTP://WWW.BOSONTREINAMENTOS.COM.BR/](http://WWW.BOSONTREINAMENTOS.COM.BR/) DATA DE ACESSO: 03/03/2022

[HTTP://WWW.PROFTONINHO.COM/DOCS/MODELAGEM\\_AULA\\_6\\_ENTID\\_ASSOC.PDF](http://WWW.PROFTONINHO.COM/DOCS/MODELAGEM_AULA_6_ENTID_ASSOC.PDF) DATA DE ACESSO: 01/07/2015

[HTTPS://MATERIALPUBLIC.IMD.UFRN.BR/CURSO/DISCIPLINA/4/56/1/6](https://MATERIALPUBLIC.IMD.UFRN.BR/CURSO/DISCIPLINA/4/56/1/6) DATA DE ACESSO: 01/02/2023

ELMASRI, R.; NAVATHE S. B. SISTEMAS DE BANCO DE DADOS. 4 ED. EDITORA ADDISON-WESLEY. 2005. - CAPÍTULO 3

DAVENPORT, THOMAS H.; PRUSAK, LAURENCE. CONHECIMENTO EMPRESARIAL: COMO AS ORGANIZAÇÕES

GERENCIAM O SEU CAPITAL INTELECTUAL. RIO DE JANEIRO: CAMPUS, 1998.

[HTTP://WWW.IME.UNICAMP.BR/~HILDETE/DADOS.PDF](http://WWW.IME.UNICAMP.BR/~HILDETE/DADOS.PDF) ACESSO EM: 12 MAIO 2016.





OBRIGADO