

Transformações entre modelos

Capítulo 5

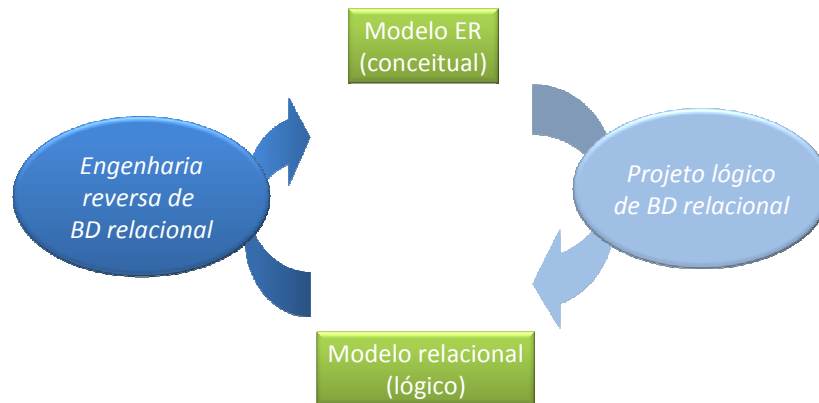
Transformações entre modelos

Modelo ER
(conceitual)

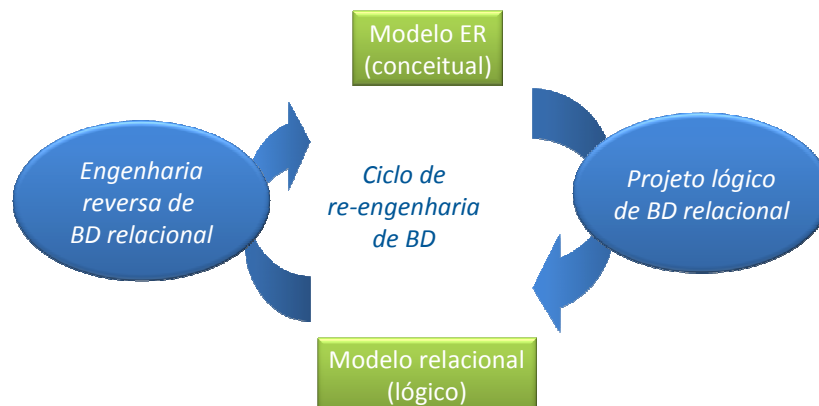
*Projeto lógico
de BD relacional*

Modelo relacional
(lógico)

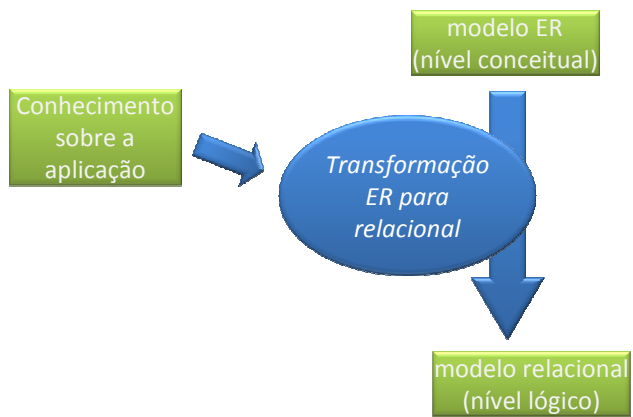
Transformações entre modelos



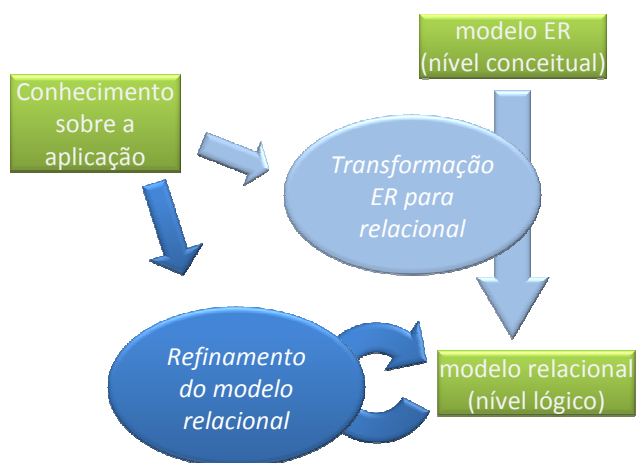
Transformações entre modelos



Projeto lógico



Projeto lógico



Transformação ER para relacional

- Regras gerais:
 - Aplicáveis à maioria dos casos;
 - Implementadas em ferramentas CASE.
- Há situações em que:
 - por exigências da aplicação, outros mapeamentos são usados.
- Objetivos básicos:
 1. Boa performance
 2. Simplificação do desenvolvimento

Princípios por trás das regras de tradução

1. Evitar junções
2. Diminuir o número de chaves
3. Evitar campos opcionais

Junção

Junção

Operação para buscar dados de diversas linhas associadas pela igualdade de campos

Junção

Junção

Operação para buscar dados de diversas linhas associadas pela igualdade de campos

- Exemplo:
 - buscar os dados de um empregado e os dados de seu departamento (duas tabelas diferentes)

Minimizar junções

- SGBD relacional normalmente armazena os dados de uma linha contiguamente em disco.
- Junção envolve diversos acessos a disco.
- Preferível:
 - ter os dados necessários a uma consulta em uma única linha.

Chave e índice

- Implementação eficiente do controle de chaves:
 - SGBD usa um índice.
- Índices tendem a ocupar espaço considerável em disco.
- Inserção e remoção de entradas em um índice:
 - Podem exigir diversos acessos a disco.

Diminuir o número de chaves

- Usar implementações com menos chaves.
- Exemplo:

Cliente (CodCliente, Nome, NomeContato, Endereço, Telefone)

Diminuir o número de chaves

- Usar implementações com menos chaves.
- Exemplo:

Cliente (CodCliente, Nome, NomeContato, Endereço, Telefone)

ou:

Cliente (CodCliente, Nome, NomeContato)

ClienteEnder (CodCliente, Endereço, Telefone)
CodCliente referencia Cliente

Campos opcionais

Campo opcional

campo que pode assumir o valor VAZIO
(NULL em SQL).

Campos opcionais

Campo opcional

campo que pode assumir o valor VAZIO
(NULL em SQL).

- SGBD relacional **não desperdiça espaço** pelo fato de campos de uma linha estarem vazios.
- Campo opcional **não tem influência no desempenho**.

Evitar campos opcionais

- Controle de campo opcional *pode complicar* programação:
 - Verificar quais campos podem estar vazios, quando isto depende do tipo de linha.
- Regra *mais "fraca"* que as precedentes.

Passos da transformação ER para relacional

1. Tradução inicial de *entidades* e respectivos atributos
2. Tradução de *relacionamentos* e respectivos atributos
3. Tradução de *generalizações/especializações*

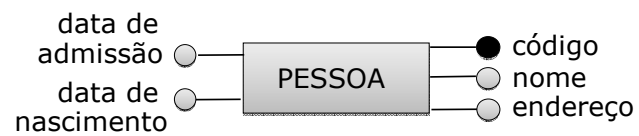
Implementação inicial de entidades

1. Cada **entidade** é traduzida para uma **tabela**.
2. Cada **atributo** da entidade define uma **coluna** desta tabela.
3. Atributos **identificadores** da entidade correspondem a **chave primária** da tabela.

Implementação inicial de entidades

1. Cada **entidade** é traduzida para uma **tabela**.
 2. Cada **atributo** da entidade define uma **coluna** desta tabela.
 3. Atributos **identificadores** da entidade correspondem a **chave primária** da tabela.
- Tradução **inicial**:
 - Passos que seguem podem fundir tabelas.

Implementação de entidade exemplo

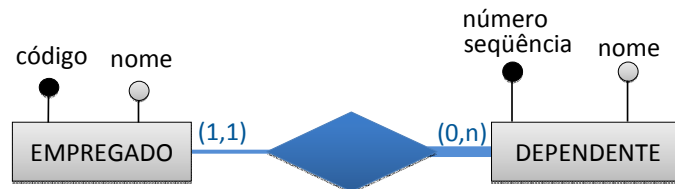


Implementação de entidade exemplo

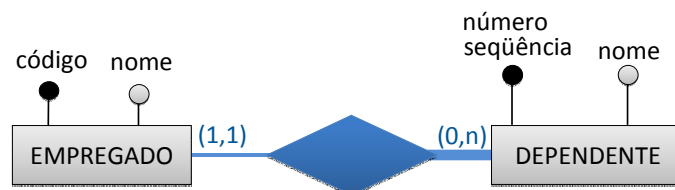


Pessoa (CodigoPess, Nome, Endereço, DataNasc, DataAdm)

Tradução de entidade relacionamento identificador

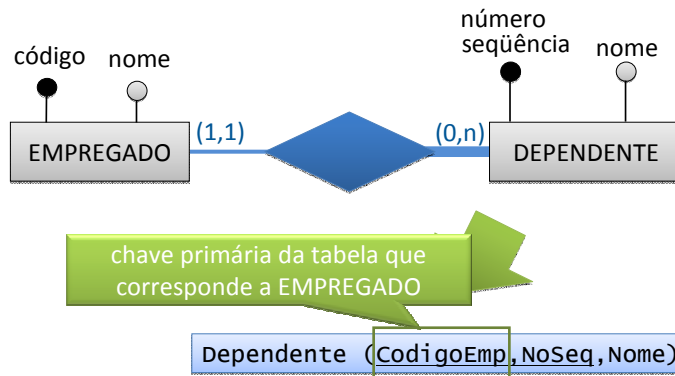


Tradução de entidade relacionamento identificador

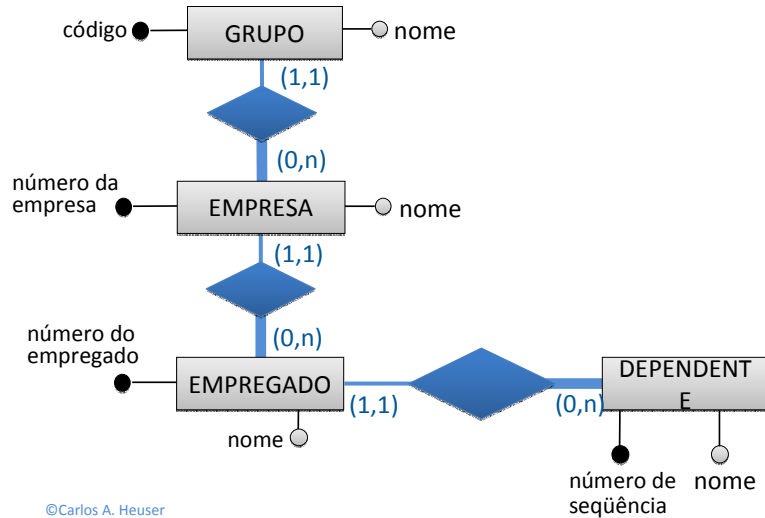


Dependente (CodigoEmp, NoSeq, Nome)

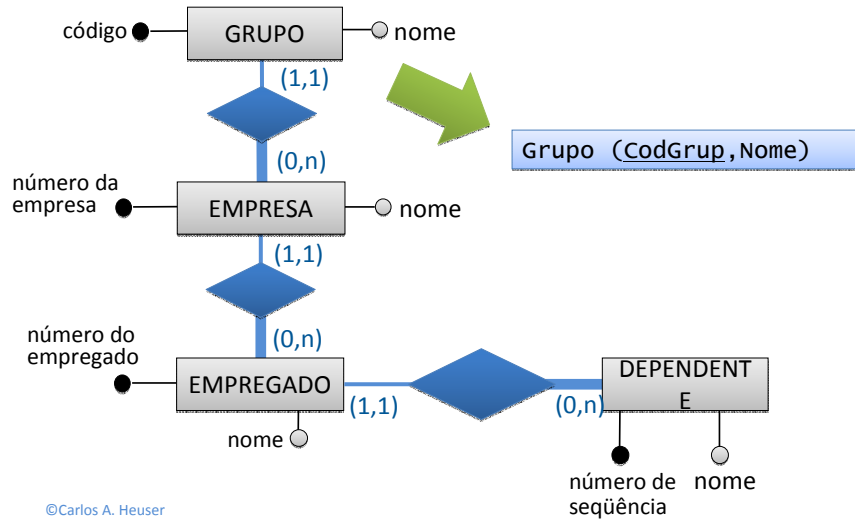
Tradução de entidade relacionamento identificador



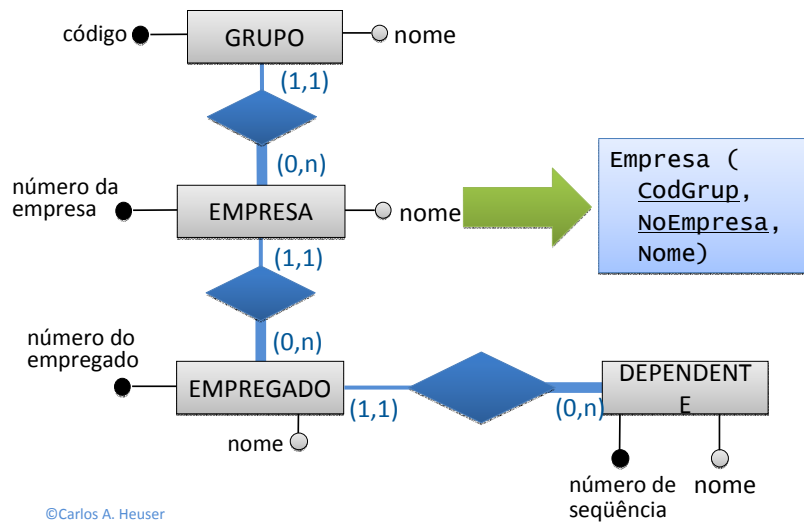
Relacionamento identificador recursão



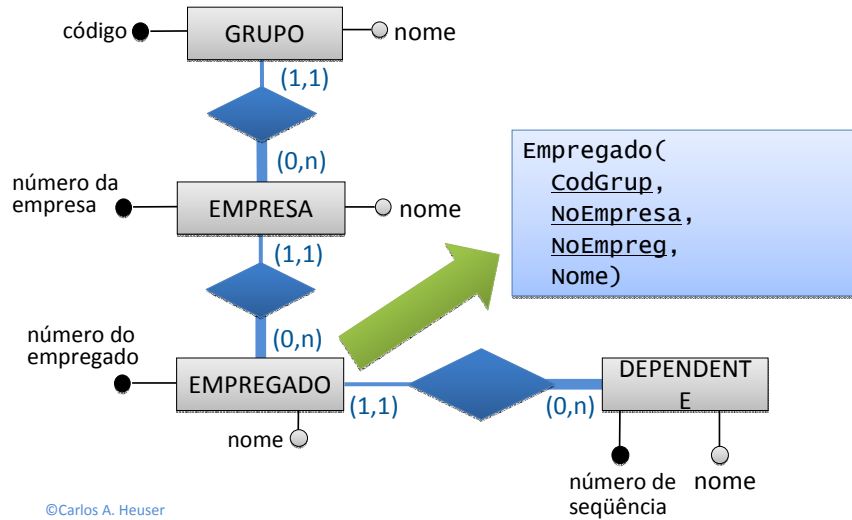
Relacionamento identificador recursão



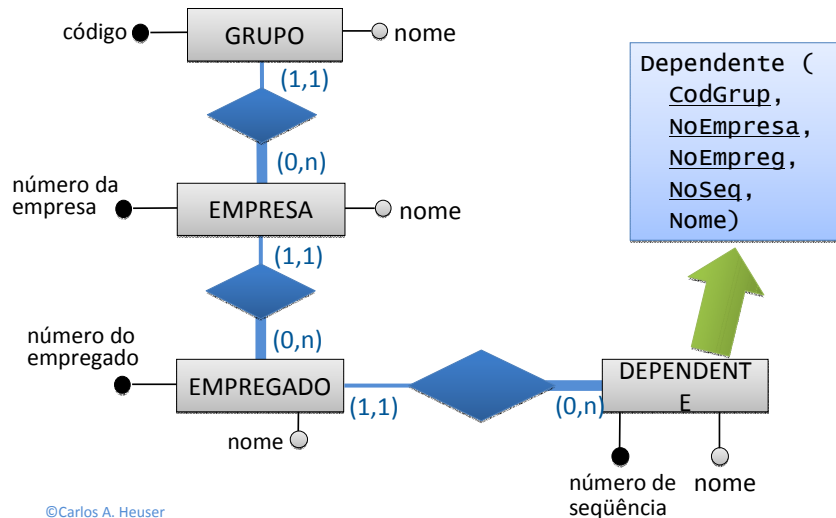
Relacionamento identificador recursão



Relacionamento identificador recursão



Relacionamento identificador recursão



Nomes de colunas

- Referenciados freqüentemente em programas e outras formas de texto em computador.
- Para diminuir o trabalho de programadores:
 - manter os **nomes de colunas** curtos.
- SGBD relacional:
 - nome de uma coluna **não pode conter brancos**.

Nomes de atributos e nomes de colunas

- **Não transcrever** os nomes de **atributos** para nomes de **colunas**.
- Nomes de atributos compostos de **diversas palavras** devem ser **abreviados**.
- Nomes de colunas não necessitam conter o nome da tabela:
 - Preferível usar o nome de coluna **Nome** a usar os nomes de coluna **NomePess** ou **NomePessoa**.
 - SQL já exige muitas vezes a forma:
Pessoa.Nome
 - Exceção: chave primária (ver a seguir)

Nome da coluna chave primária

- Chave primária:
 - pode **aparecer em outras tabelas** na forma de chave estrangeira.
- Recomendável:
 - nomes das colunas que compõem a chave primária:
 - **sufixados** ou **prefixados** com o nome ou sigla da **tabela** na qual aparecem como chave primária.
 - Exemplo
 - **CodigoPess**

Implementação de relacionamento alternativas

1. Tabela própria
2. Adição de colunas a uma das tabelas
3. Fusão de tabelas

Implementação de relacionamento alternativas

1. Tabela própria
 2. Adição de colunas a uma das tabelas
 3. Fusão de tabelas
- Alternativa depende da **cardinalidade** (máxima e mínima do relacionamento)

Tabela própria

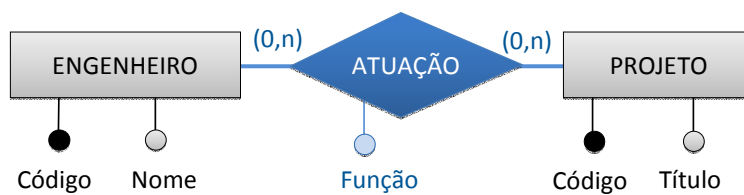
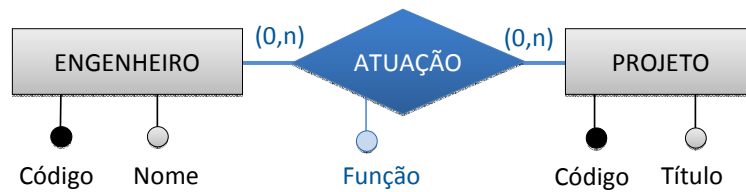


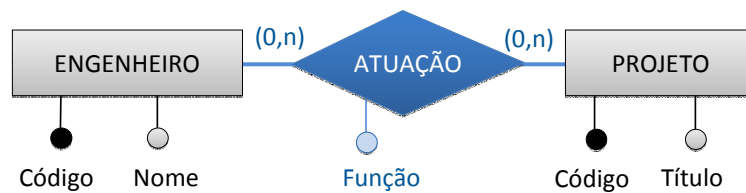
Tabela própria



tradução das entidades

Engenheiro (CodEng, Nome)
Projeto (CodProj, Título)

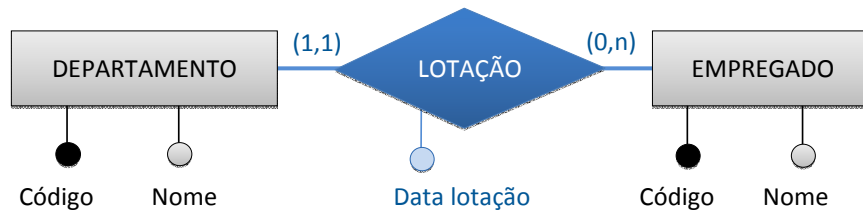
Tabela própria



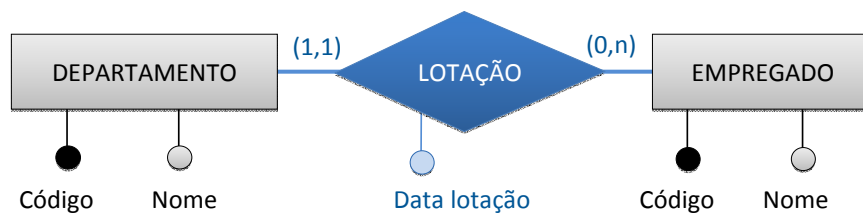
tradução do relacionamento

Engenheiro (CodEng, Nome)
Projeto (CodProj, Título)
Atuação (CodEng, CodProj, Função)
CodEng referencia Engenheiro
CodProj referencia Projeto

Adição de colunas



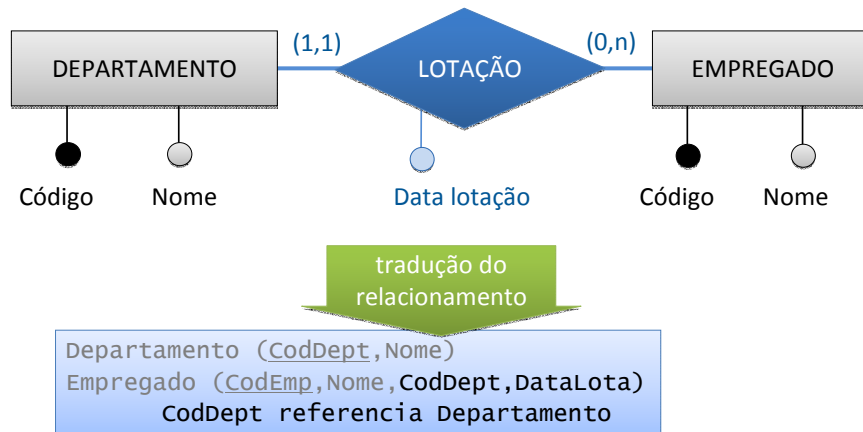
Adição de colunas



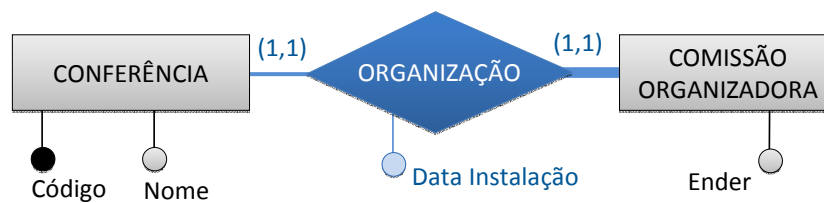
tradução das entidades

Departamento (CodDept, Nome)
Empregado (CodEmp, Nome)

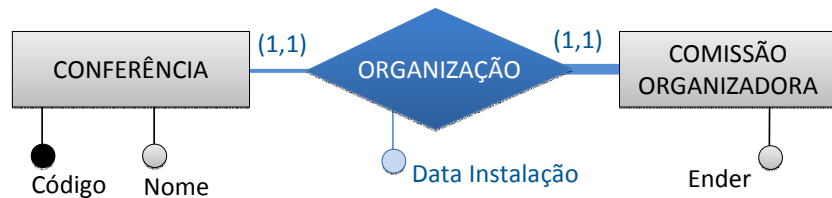
Adição de colunas



Fusão de tabelas



Fusão de tabelas



Conferência (CodConf, Nome, DataInstComOrg, EnderComOrg)

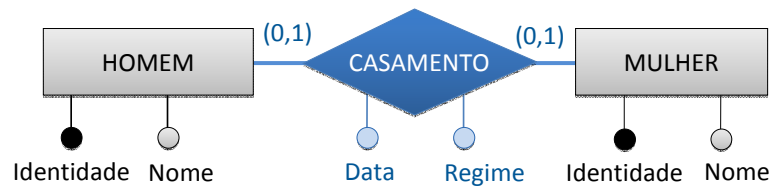
Implementação de relacionamentos 1:1

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
(0,1) (0,1)	±	✓	✗
(0,1) (1,1)	≠	±	✓
(1,1) (1,1)	≠	≠	✓

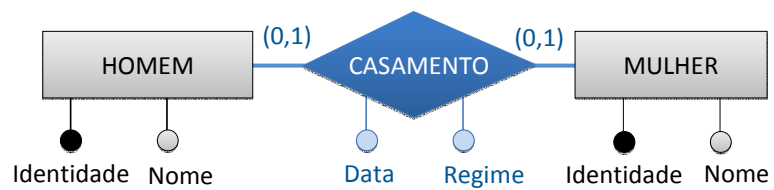
Convenção:

- ✓ - alternativa preferida
- ± - pode ser usada - 1ª opção
- ≠ - pode ser usada - 2ª opção
- ✗ - não cabe

1:1 –ambas entidades opcionais exemplo



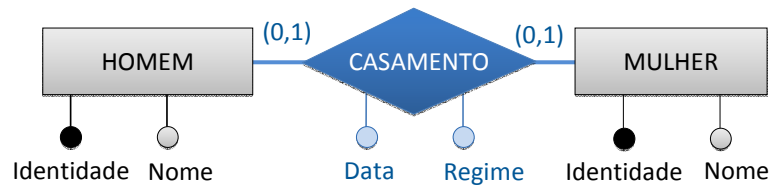
1:1 - ambas opcionais adição de colunas



tradução das
entidades

Mulher (IdentM, Nome)
Homem (IdentH, Nome)

1:1 - ambas opcionais adição de colunas

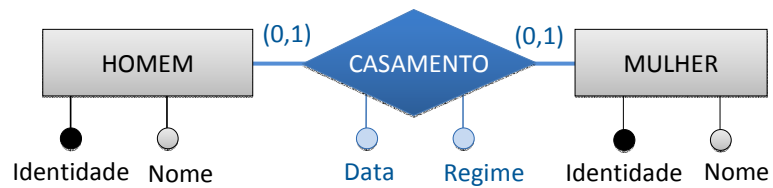


tradução do
relacionamento

```

Mulher (IdentM, Nome, IdentH, Data, Regime)
      IdentH referencia Homem
Homem (IdentH, Nome)
    
```

1:1 - ambas opcionais tabela própria

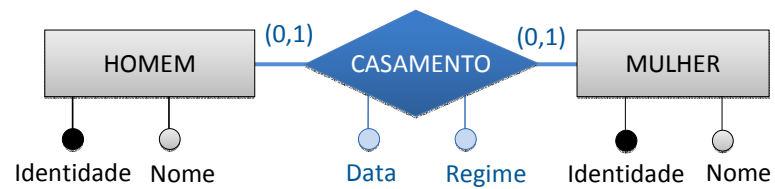


tradução do
relacionamento

```

Mulher (IdentM, Nome)
Homem (IdentH, Nome)
Casamento (IdentM, IdentH, Data, Regime)
      IdentM referencia Mulher
      IdentH referencia Homem
    
```


1:1 - ambas opcionais fusão de tabelas






Casamento (IdentH, NomeH, Data, Regime, IdentM, NomeM)

qual é a chave
primária desta
tabela?

1:1 - ambas opcionais discussão

- Solução por fusão de tabelas é **inviável**:
 - Chave primária **artificial**.
- Solução por adição de colunas **melhor**:
 - Menor número de **junções**;
 - Menor número de **chaves**.
- Solução por **tabela própria aceitável**.

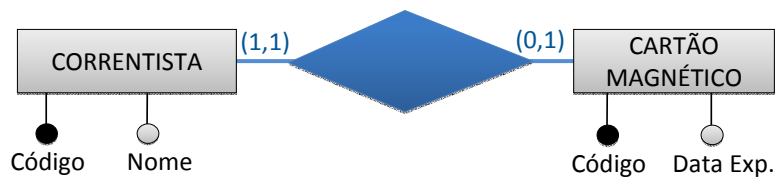
Implementação de relacionamentos 1:1

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
(0,1)  (0,1)	±	✓	✗
(0,1)  (1,1)	∓	±	✓
(1,1)  (1,1)	∓	∓	✓

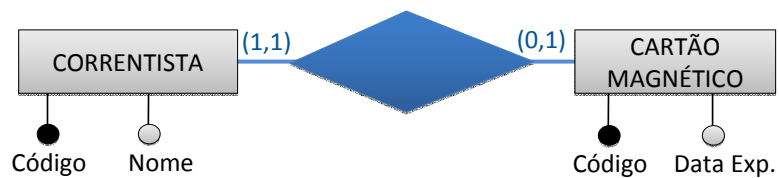
Convenção:

- ✓ - alternativa preferida
- ± - pode ser usada - 1ª opção
- ∓ - pode ser usada - 2ª opção
- ✗ - não cabe

1:1 - uma entidade opcional, outra obrigatória

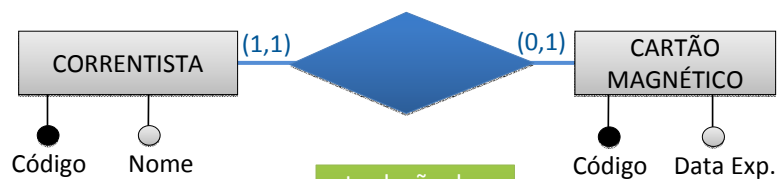


1:1 - opcional/obrigatória fusão de tabelas



Correntista (CodCorrent, Nome, CodCartão, DataExp)

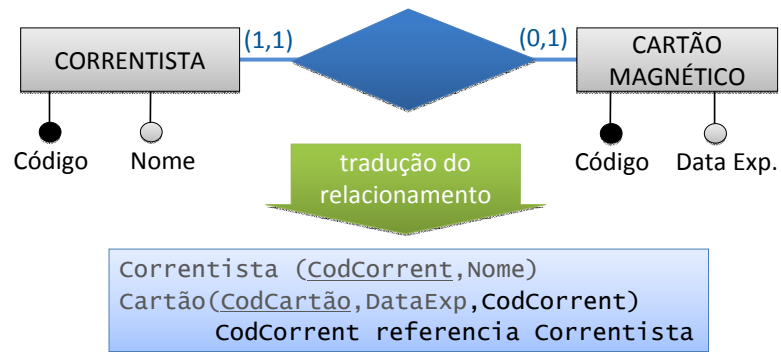
1:1 - opcional/obrigatória adição de colunas



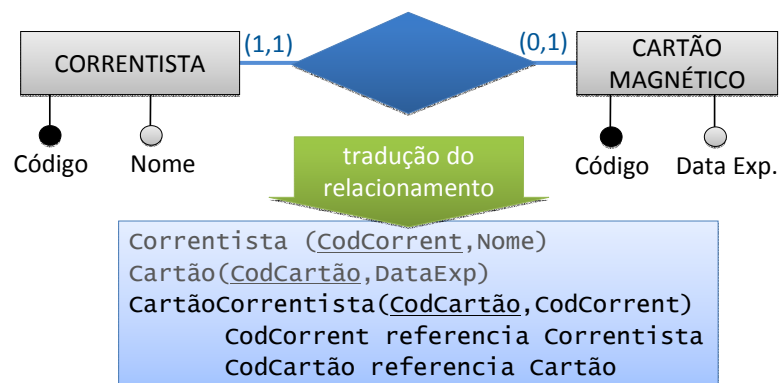
tradução das
entidades

Correntista (CodCorrent, Nome)
Cartão(CodCartão, DataExp)

1:1 - opcional/obrigatória adição de colunas



1:1 - opcional/obrigatória tabela própria






1:1 - opcional/obrigatória discussão

- Solução **por tabela própria** é **pior** que a solução por **adição de colunas**:
 - Maior número de junções;
 - Maior número de índices;
 - Nenhuma tem problema de campos opcionais.

1:1 - opcional/obrigatória discussão

- **Adição de colunas** versus **fusão de tabelas**:
 - Fusão de tabelas é melhor em termos de número de junções e número de chaves;
 - Adição de colunas em melhor em termos de campos opcionais;
 - Fusão de tabelas é considerada a melhor e adição de colunas é aceitável.

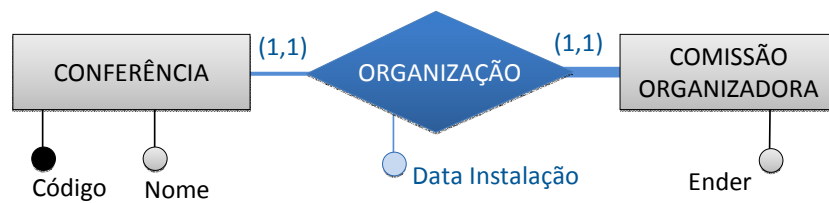
Implementação de relacionamentos 1:1

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
(0,1)  (0,1)	±	✓	✗
(0,1)  (1,1)	∓	±	✓
(1,1)  (1,1)	∓	∓	✓

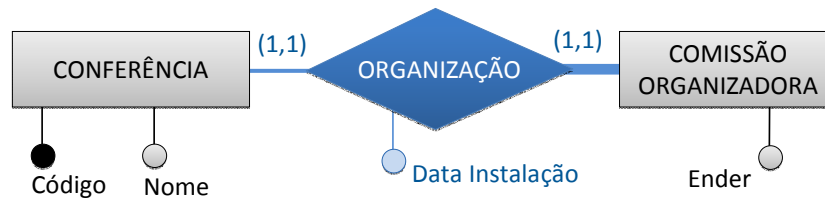
Convenção:

- ✓ - alternativa preferida
- ± - pode ser usada - 1ª opção
- ∓ - pode ser usada - 2ª opção
- ✗ - não cabe

1:1 - ambas entidades obrigatórias



1:1 - ambas obrigatórias fusão de tabelas






Conferência (CodConf, Nome, DataInstComOrg, EnderComOrg)

1:1 - Ambas obrigatórias

- Demais alternativas (adição de colunas, tabela própria).
- Nenhuma das demais alternativas atende plenamente.
- Em ambas:
 - Entidades que participam do relacionamento seriam representadas através de duas tabelas distintas.
 - Estas tabelas teriam a mesma chave primária e relação um-para-um entre suas linhas.
 - Maior número de junções.
 - Maior número de chaves primárias.



Implementação de relacionamentos 1:1

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
	±	✓	✗
	∓	±	✓
	∓	∓	✓

Convenção:

- ✓ - alternativa preferida
- ± - pode ser usada - 1ª opção
- ∓ - pode ser usada - 2ª opção
- ✗ - não cabe

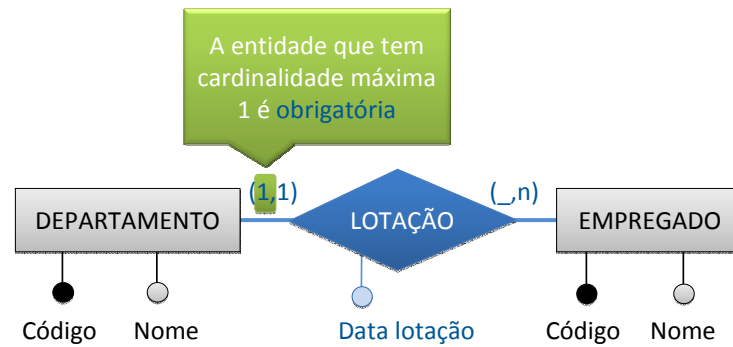
Relacionamentos 1:n

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
	±	✓	✗
	✗	✓	✗

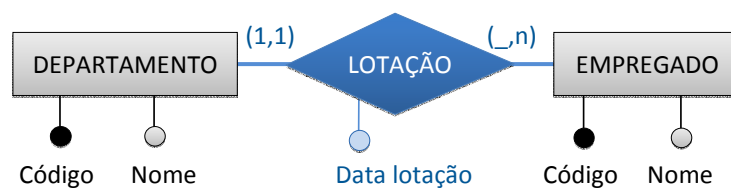
Convenção:

- ✓ - alternativa preferida
- ± - pode ser usada - 1ª opção
- ✗ - não cabe

1:n - caso 1



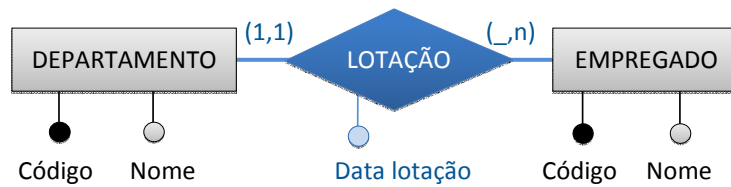
1:n - caso 1 adição de colunas



tradução do
relacionamento

Departamento (CodDept, Nome)
 Empregado (CodEmp, Nome, CodDept, DataLota)
 CodDept referencia Departamento

1:n - caso 1 tabela própria





tradução do
relacionamento

Departamento (CodDept, Nome)
 Empregado (CodEmp, Nome,
 Lotacao(CodEmp, CodDept, DataLota)
 CodDept referencia Departamento
 CodEmp referencia Empregado

1:n - caso 1 discussão

- **Fusão de tabelas:**
 - Não se aplica.
 - Implicaria em:
 - redundância de dados de departamento, ou
 - tabela aninhada.
- **Adição de colunas** é melhor que **tabela própria**:
 - Menor número de chaves;
 - Menor número de junções;
 - Não há o problema de campos opcionais.

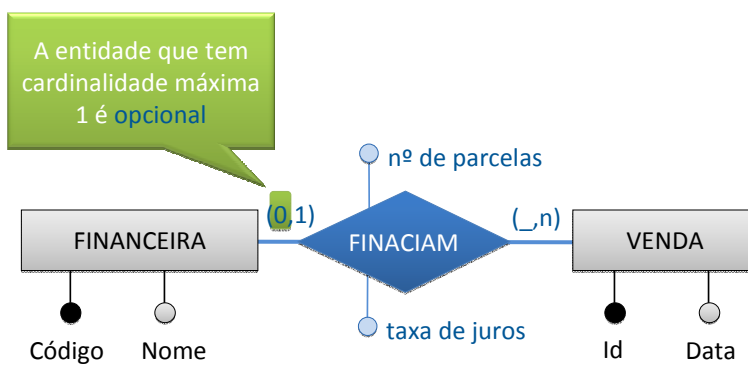
Relacionamentos 1:n

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
(0,1)  (,n)	±	✓	✗
(1,1)  (,n)	✗	✓	✗

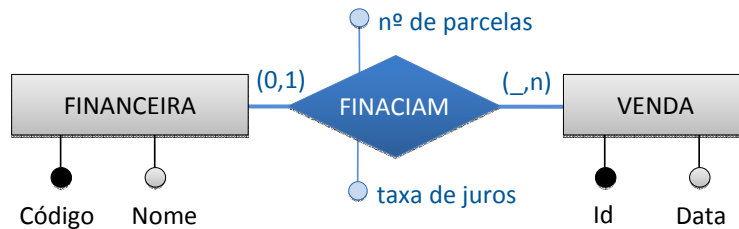
Convenção:

- ✓ - alternativa preferida
- ± - pode ser usada - 1ª opção
- ✗ - não cabe

1:n - caso 2



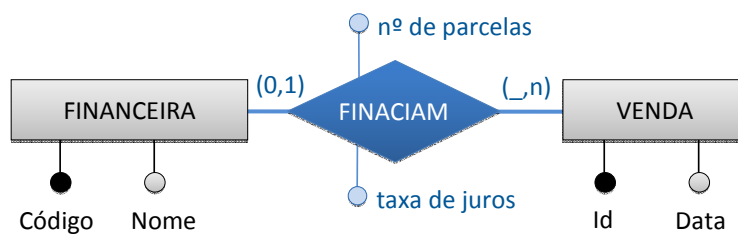
1:n - caso 2 adição de colunas



tradução do
relacionamento

Financeira (CodFin, Nome)
Venda (IdVend, Data, CodFin, NoParc, TxJuros)
CodFin referencia Financeira

1:n - caso 2 tabela própria





tradução do
relacionamento

Financeira (CodFin, Nome)
Venda (IdVend, Data)
Fianciam (IdVend, CodFin, NoParc, TxJuros)
IdVend referencia Venda
CodFin referencia Financeira

1:n - caso 2 discussão

- Implementação por **tabela própria** também é **aceitável**:
 - É melhor em relação a campos opcionais;
 - Perde em relação a junções e número de chaves.


Relacionamentos 1:n

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
(0,1)  (.,n)	±	✓	✗
(1,1)  (.,n)	✗	✓	✗

Convenção:

- ✓ - alternativa preferida
- ± - pode ser usada - 1ª opção
- ✗ - não cabe

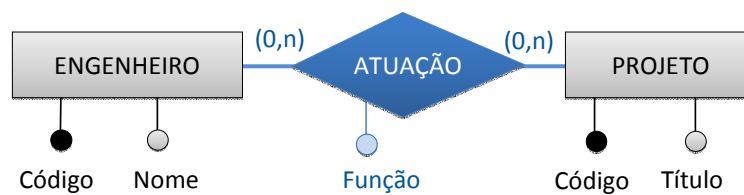
Relacionamentos n:n

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
	✓	✗	✗

Convenção:

- ✓ - alternativa preferida
- ✗ - não cabe

Relacionamentos n:n

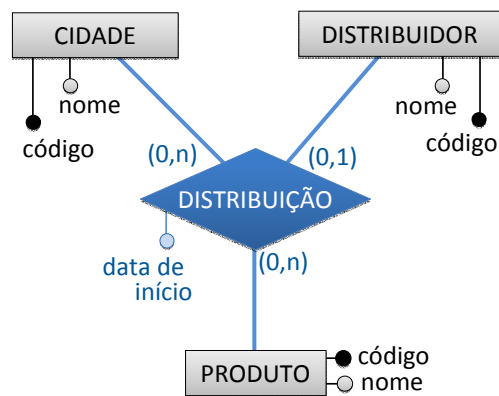


tradução do relacionamento

```

Engenheiro (CodEng, Nome)
Projeto (CodProj, Título)
Atuação (CodEng, CodProj, Função)
CodEng referencia Engenheiro
CodProj referencia Projeto
    
```

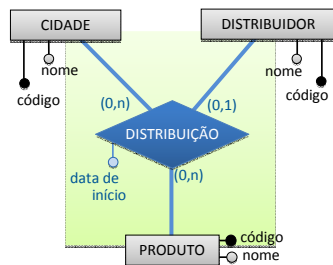
Relacionamento grau > dois



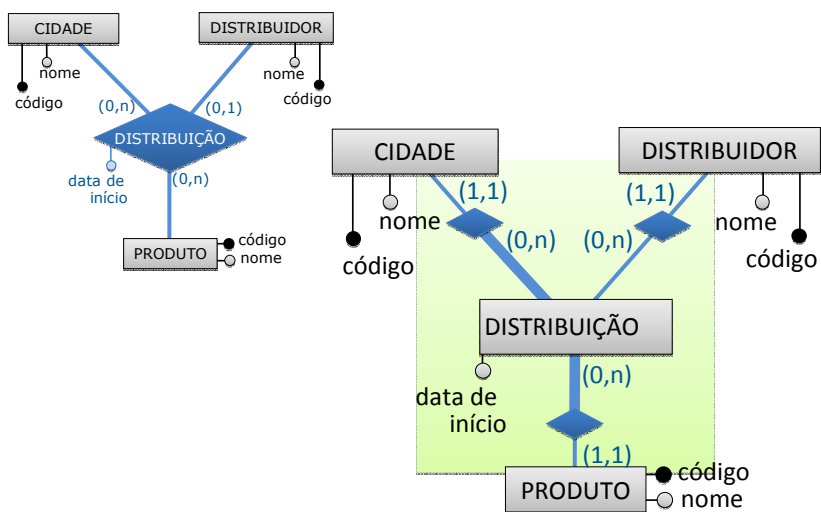
Relacionamento grau > 2

- Não são definidas regras específicas.
- O relacionamento é transformado em uma entidade.
- São aplicadas as regras de implementação de relacionamentos binários.

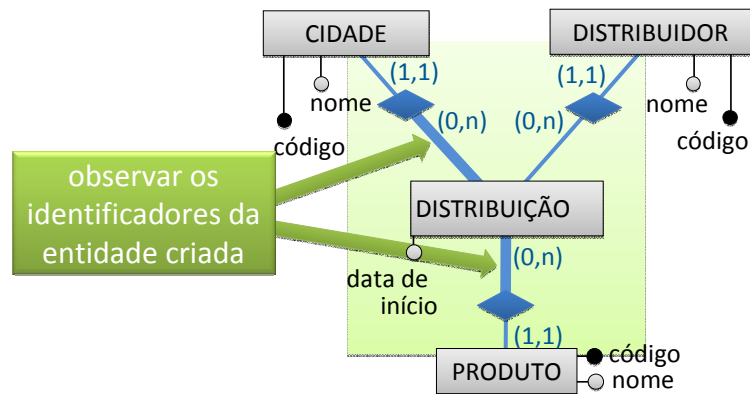
Relacionamento transformado em entidade



Relacionamento transformado em entidade



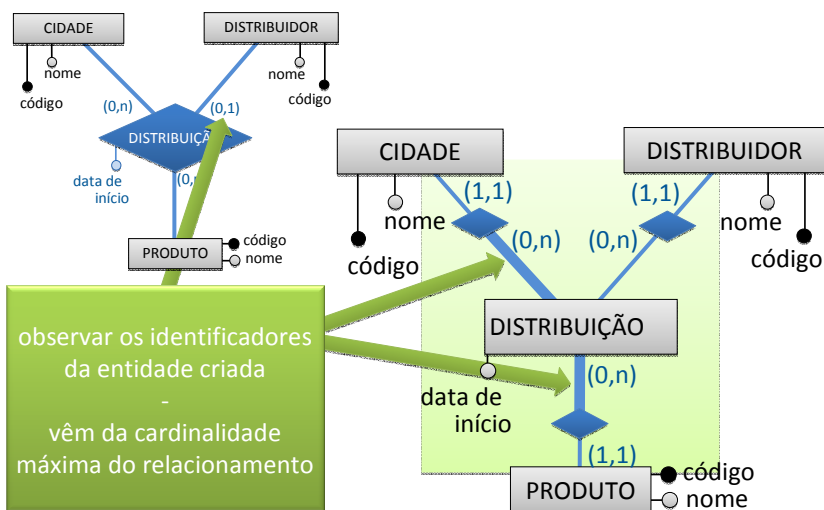
Relacionamento transformado em entidade



©Carlos A. Heuser

81

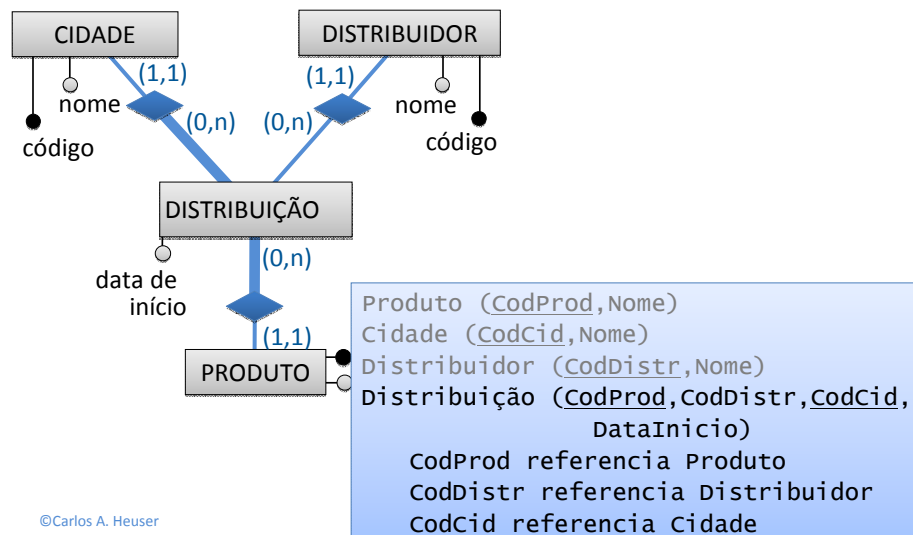
Relacionamento transformado em entidade



©Carlos A. Heuser

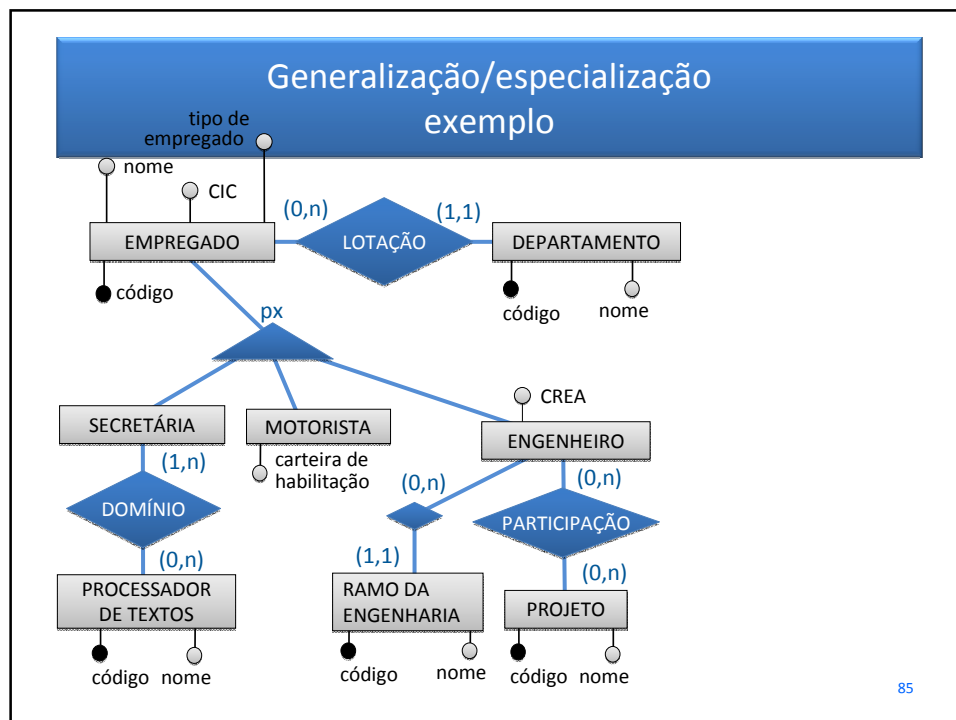
82

Modelo lógico correspondente



Implementação de generalização/especialização

1. Duas alternativas básicas:
 - a) uso de uma tabela para cada entidade
 - b) uso de uma única tabela para toda hierarquia
2. Outra alternativa (exótica):
 - a) Subdivisão de entidade genérica



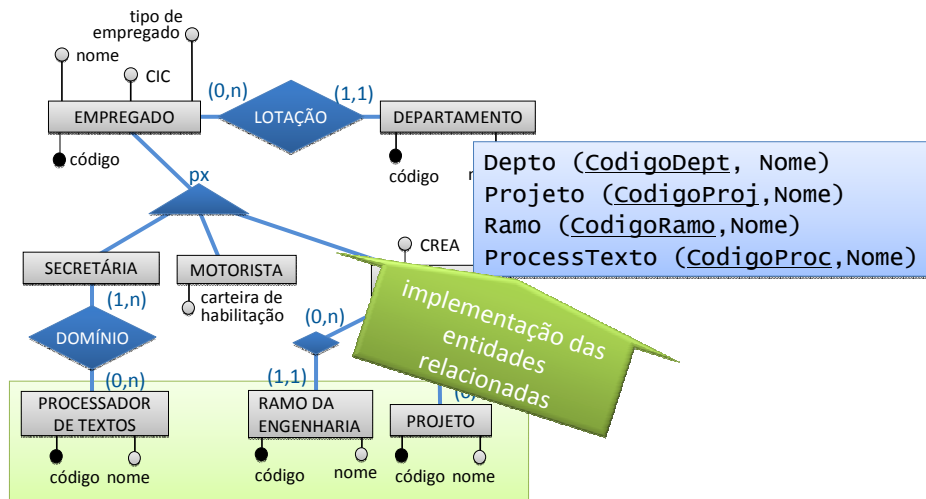
Uma tabela por hierarquia

1. Todas tabelas referentes às especializações são **fundidas em uma única tabela**.
2. Tabela contém:
 - a) Chave primária correspondente ao identificador da entidade mais genérica;
 - b) Caso não exista, uma coluna Tipo;
 - c) Uma coluna para cada atributo da entidade genérica;
 - d) Colunas referentes aos relacionamentos dos quais participa a entidade genérica e que sejam implementados através da alternativa de adicionar colunas à tabela da entidade genérica;
 - e) Uma coluna para cada atributo de cada entidade especializada (opcional)
 - f) Colunas referentes aos relacionamentos dos quais participa cada entidade especializada e que sejam implementados através da alternativa de adicionar colunas à tabela da entidade (campo opcional)

©Carlos A. Heuser

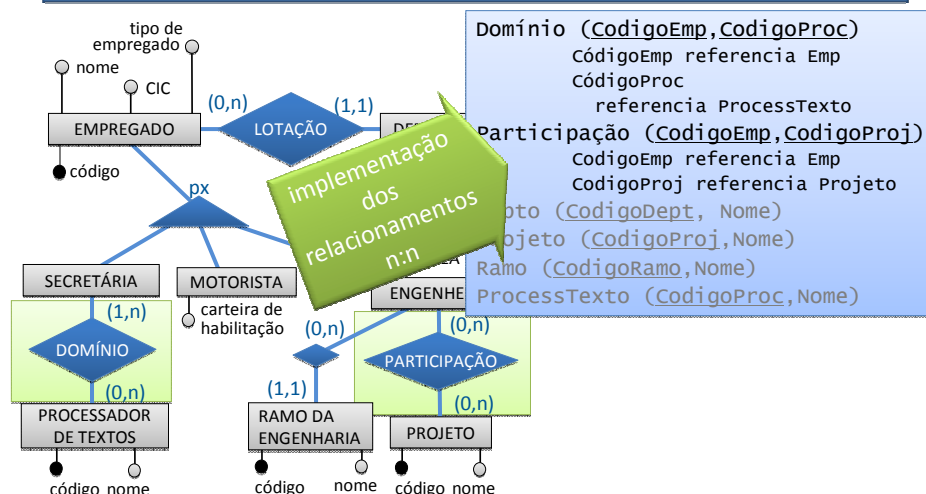
86

Implementação das entidades relacionadas



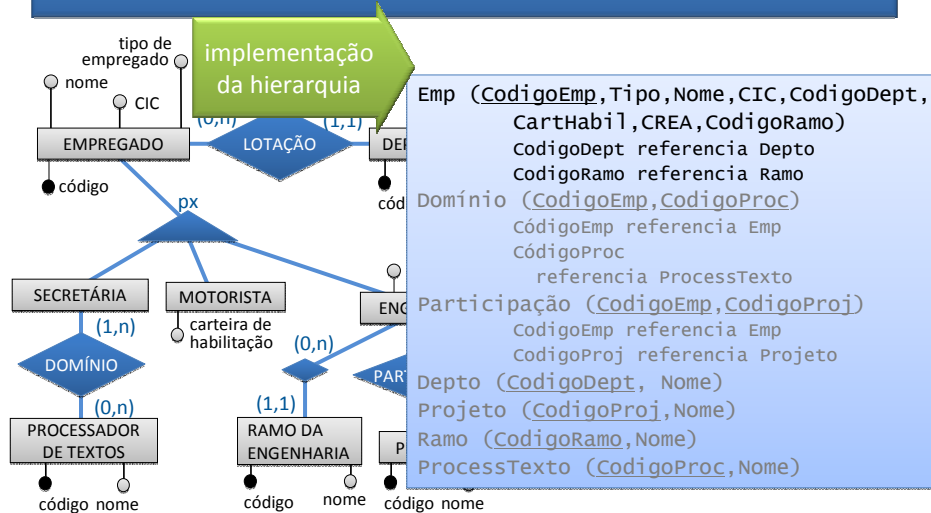
87

Implementação de relacionamentos n:n



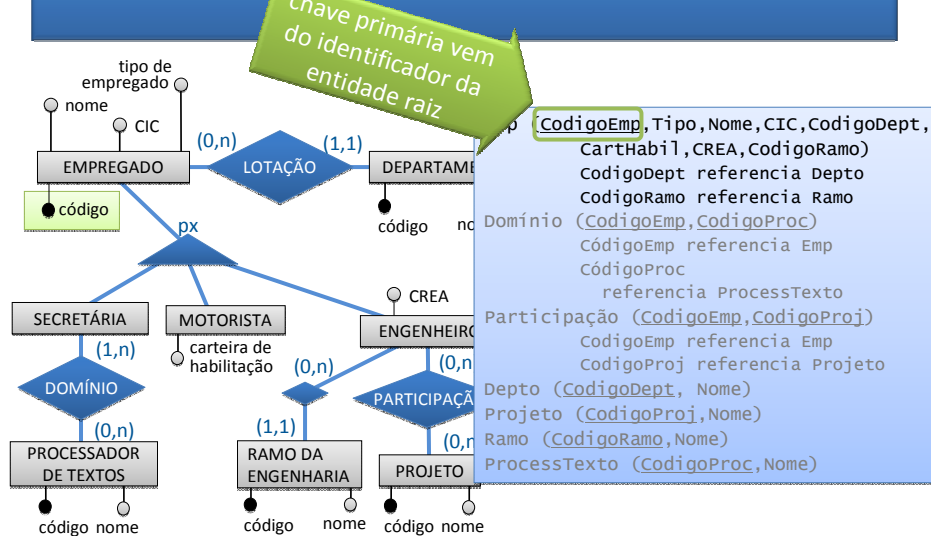
88

Tabela para a hierarquia



89

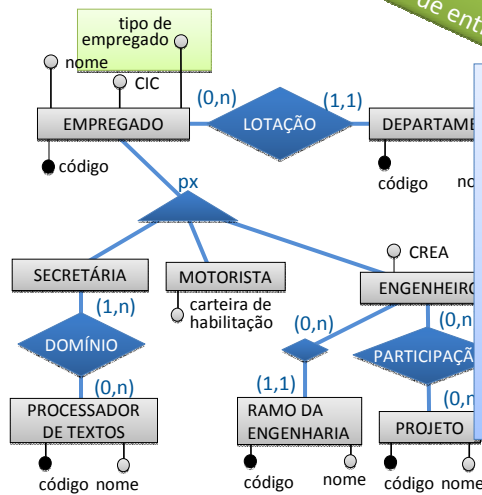
Identificador da entidade raiz



90

Inclusão de uma coluna de tipo

tabela deve ter uma coluna de tipo de entidade



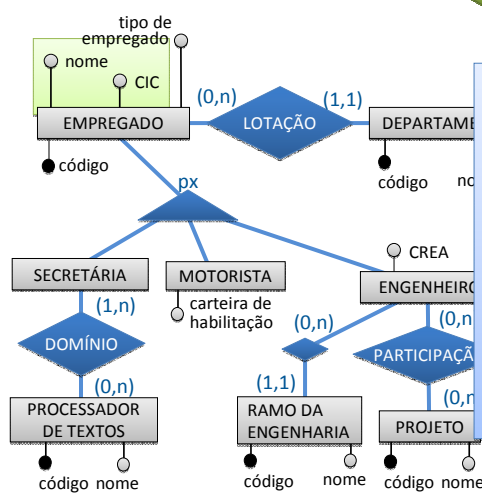
```

Emp (CodigoEmp, Tipo, Nome, CIC, CodigoDept,
    CartHabil, CREA, CodigoRamo)
CodigoDept referencia Depto
CodigoRamo referencia Ramo
Domínio (CodigoEmp, CodigoProc)
CodigoEmp referencia Emp
CodigoProc referencia ProcessTexto
Participação (CodigoEmp, CodigoProj)
CodigoEmp referencia Emp
CodigoProj referencia Projeto
Depto (CodigoDept, Nome)
Projeto (CodigoProj, Nome)
Ramo (CodigoRamo, Nome)
ProcessTexto (CodigoProc, Nome)
    
```

91

Atributos da entidade genérica

colunas para atributos da entidade genérica



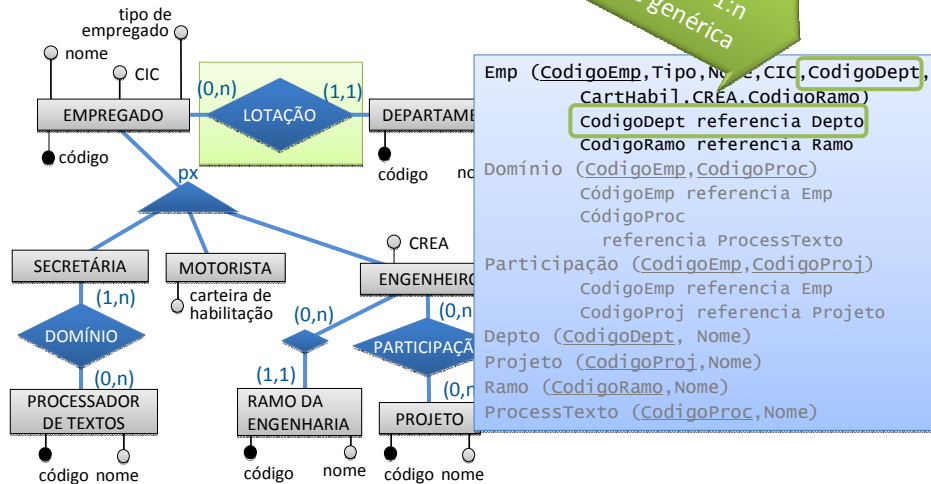
```

Emp (CodigoEmp, Tipo, Nome, CIC, CodigoDept,
    CartHabil, CREA, CodigoRamo)
CodigoDept referencia Depto
CodigoRamo referencia Ramo
Domínio (CodigoEmp, CodigoProc)
CodigoEmp referencia Emp
CodigoProc referencia ProcessTexto
Participação (CodigoEmp, CodigoProj)
CodigoEmp referencia Emp
CodigoProj referencia Projeto
Depto (CodigoDept, Nome)
Projeto (CodigoProj, Nome)
Ramo (CodigoRamo, Nome)
ProcessTexto (CodigoProc, Nome)
    
```

92

Relacionamentos 1:n da implementação da entidade genérica

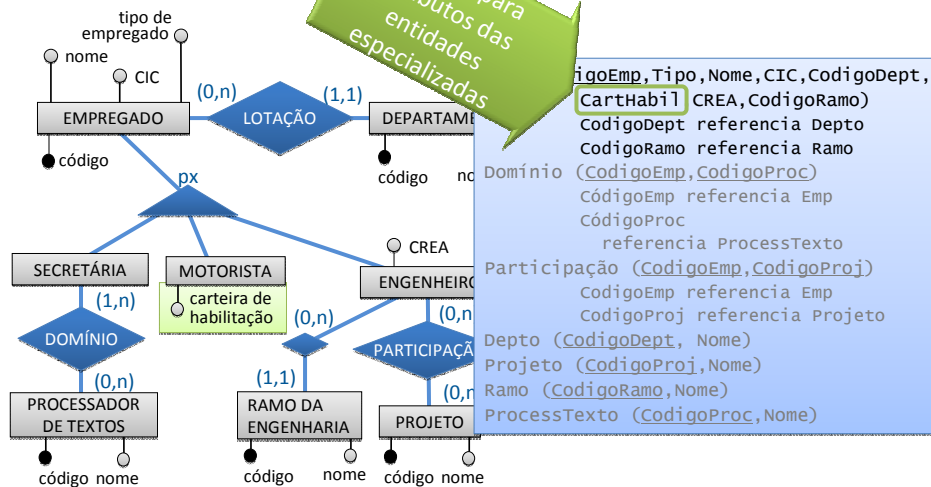
implementação da
relacionamentos 1:n
da entidade genérica



93

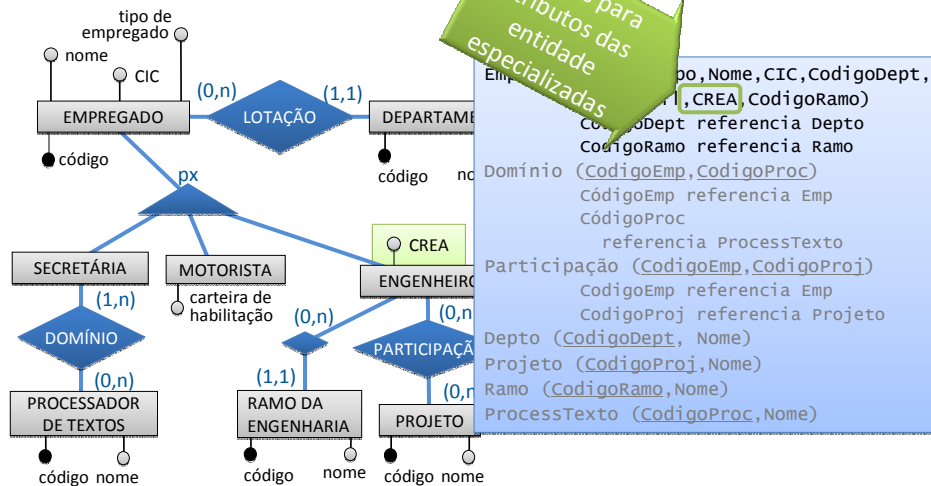
Atributos das especializações

colunas para
atributos das
entidades das
especializações



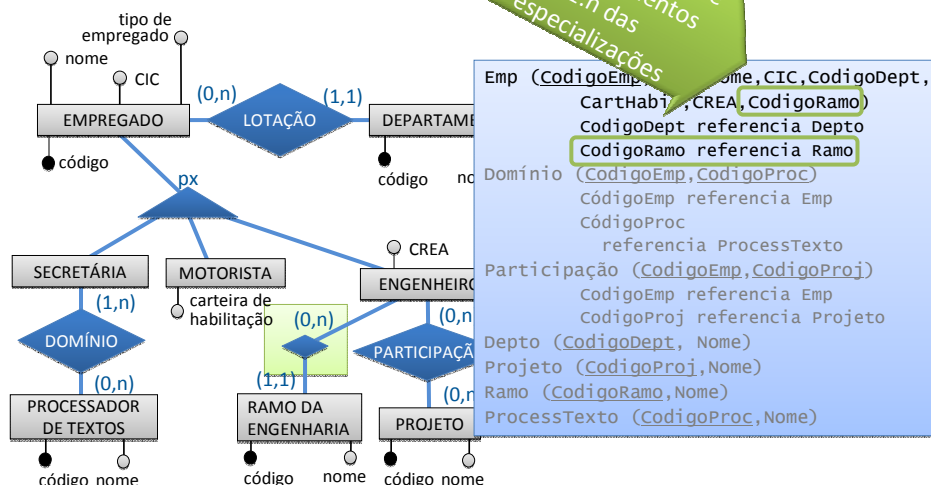
94

Atributos das especializações



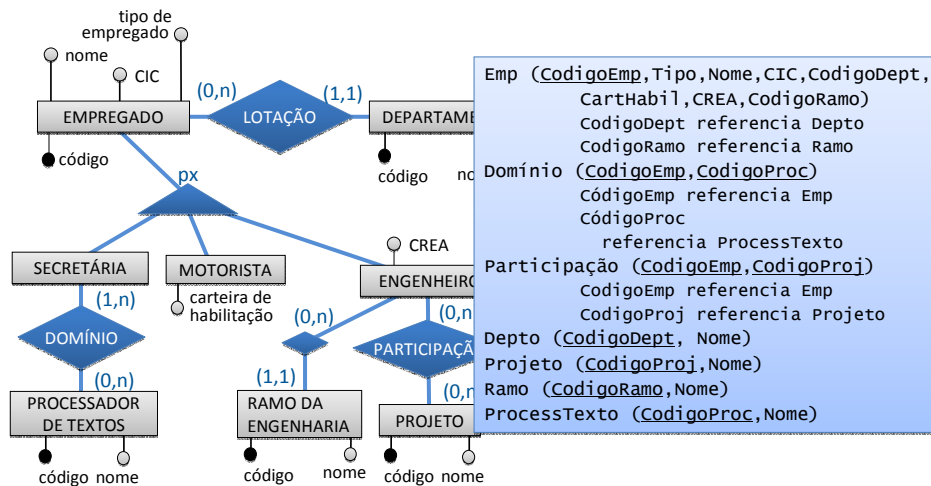
95

Relacionamentos 1:n das especializações



96

Uma tabela por hierarquia - resultado

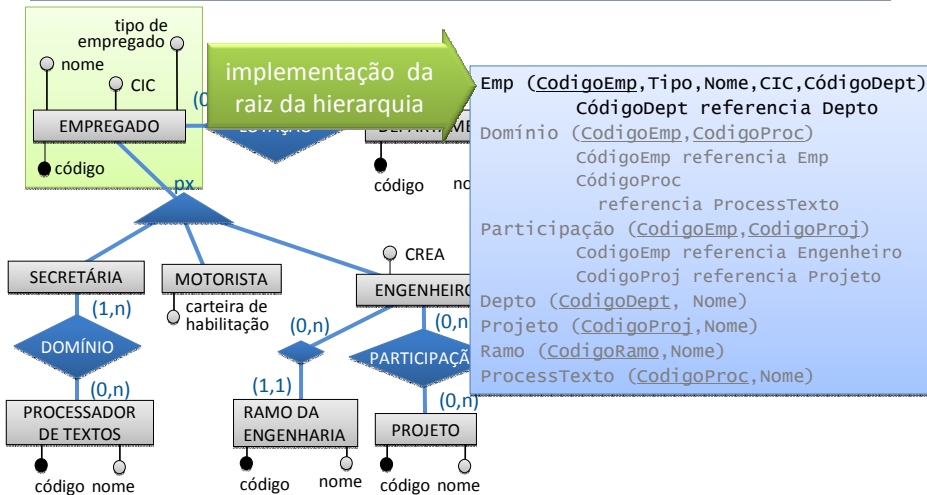


97

Uma tabela por entidade especializada

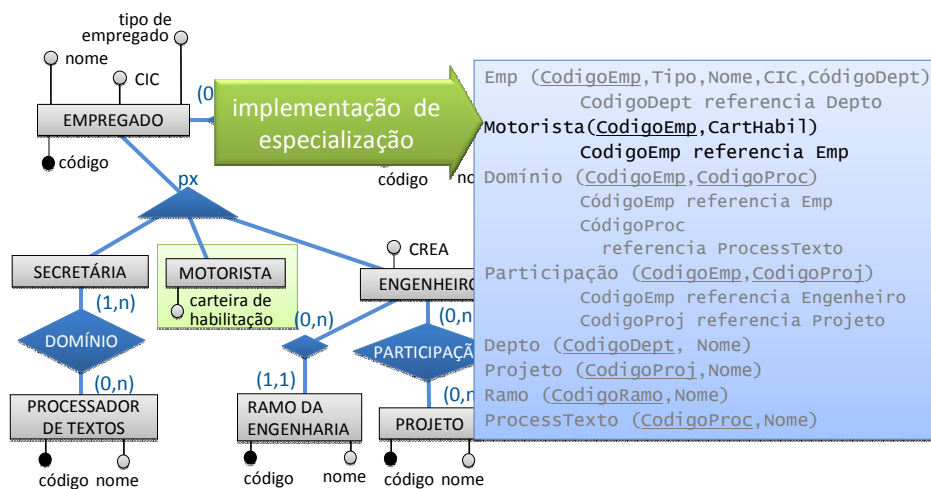
1. Criar uma tabela para cada entidade que compõe a hierarquia.
2. Incluir a chave primária da tabela correspondente à entidade genérica, em cada tabela correspondente a uma entidade especializada.

Raiz da hierarquia



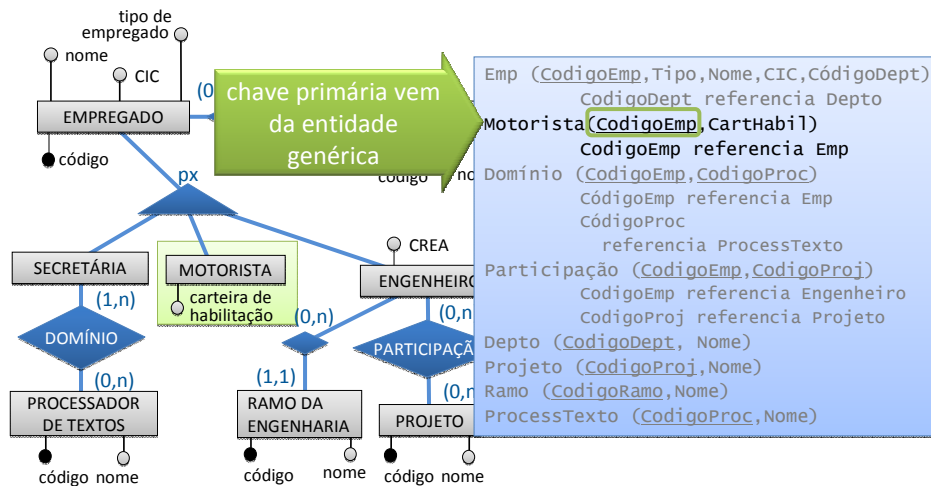
99

Entidade especializada



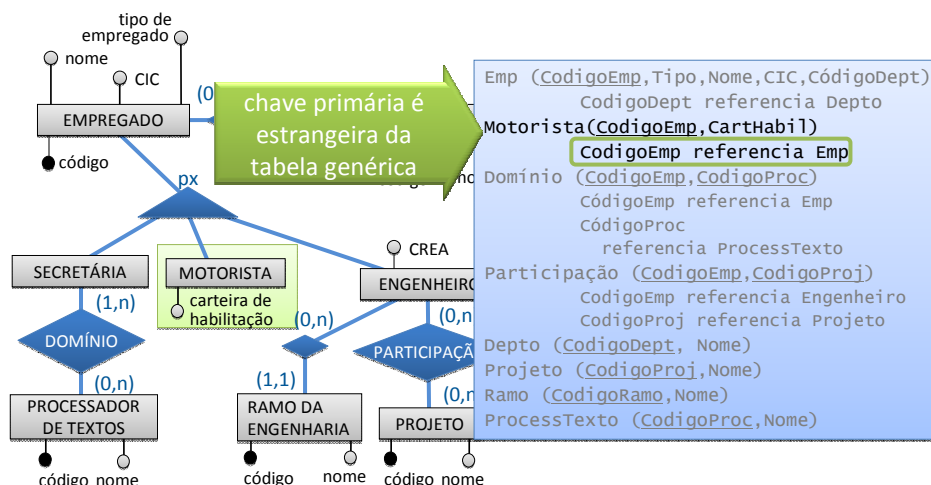
100

Entidade especializada



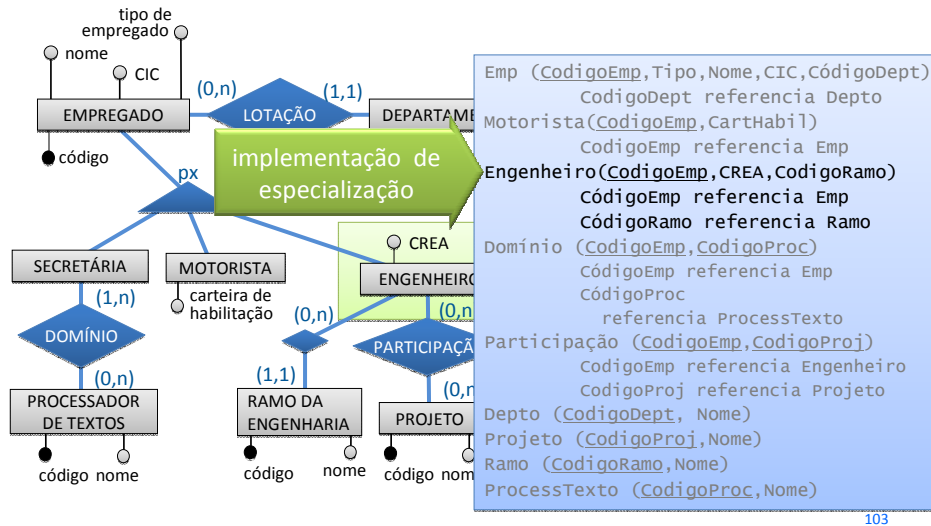
101

Entidade especializada



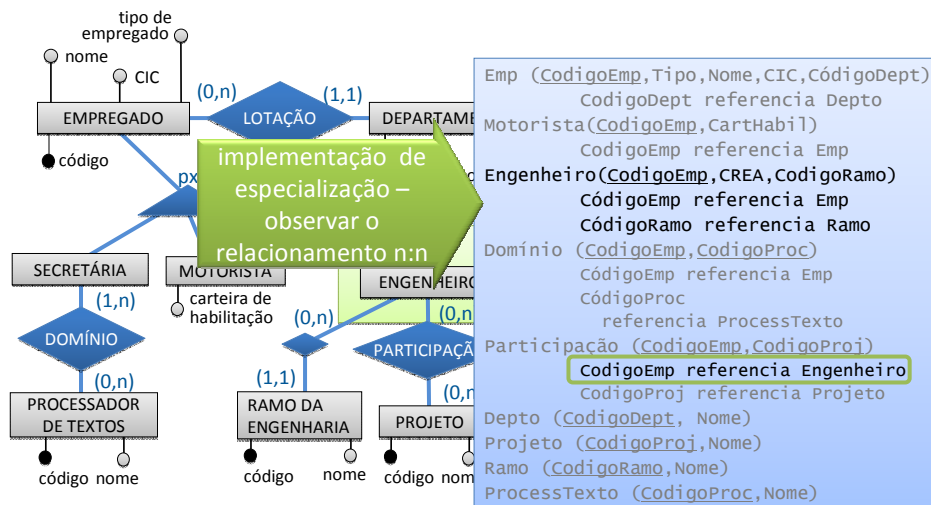
102

Entidade especializada



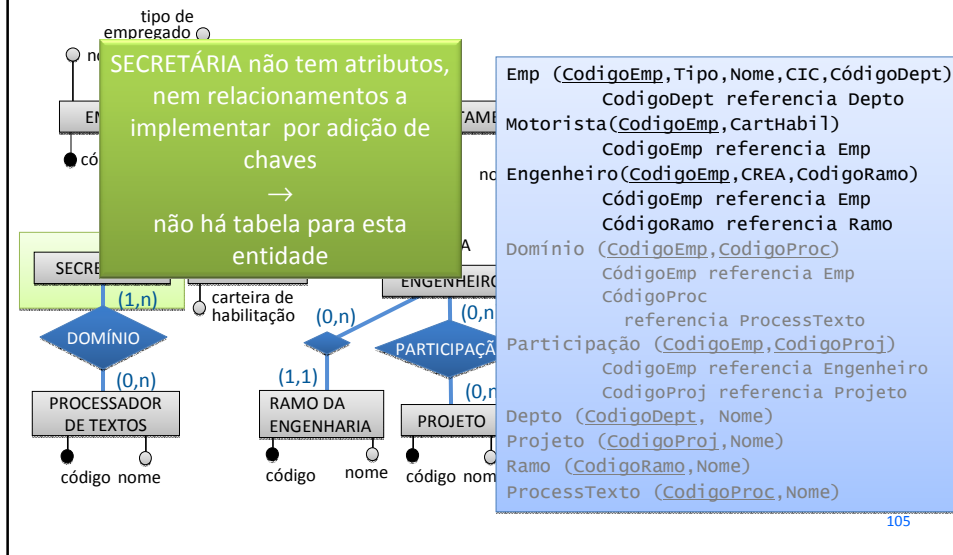
103

Relacionamento n:n com entidade especializada

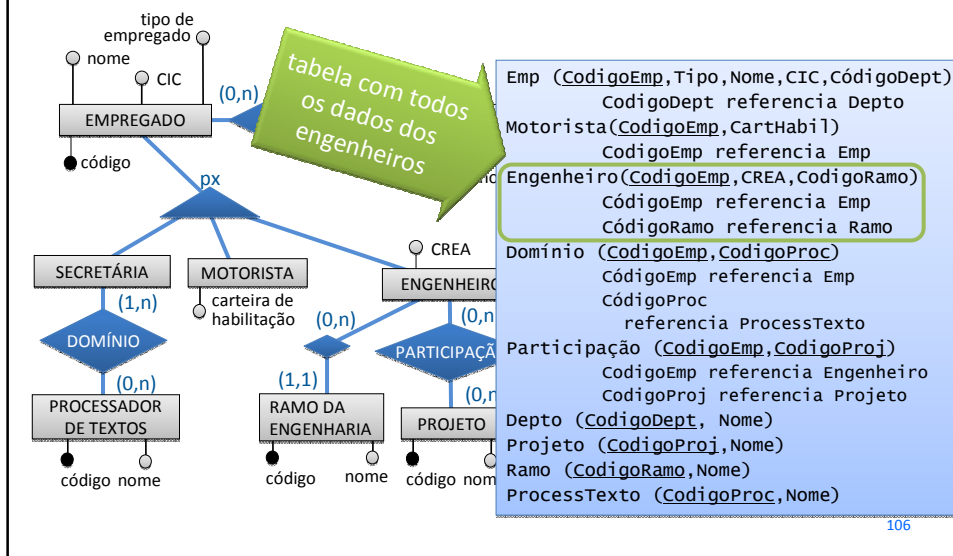


104

Entidade especializada



Uma tabela por entidade especializada resultado



Vantagens da implementação com tabela única

- Dados referentes à entidade genérica + dados referentes às especializações:
 - em uma única linha.
- Minimiza junções.
- Menor número de chaves.

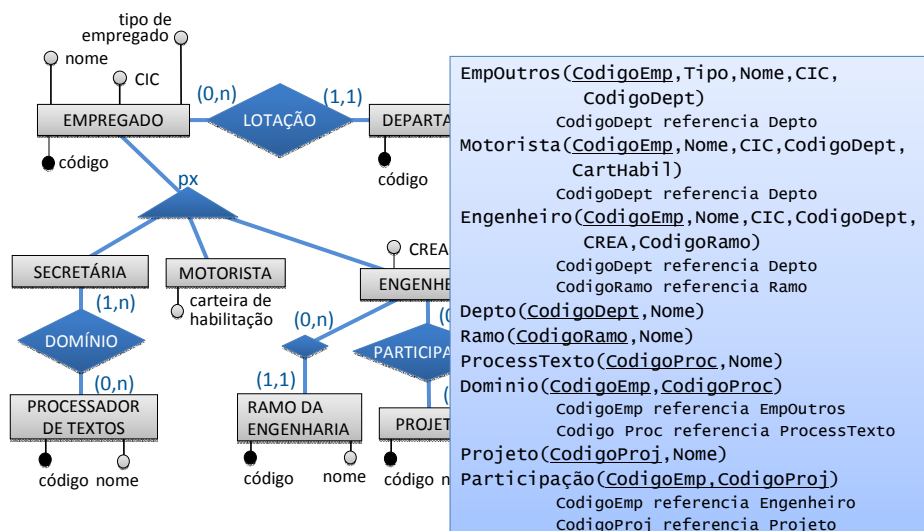
Vantagens da implementação com uma tabela por entidade especializada

- Colunas opcionais:
 - apenas aquelas referentes a atributos opcionais.

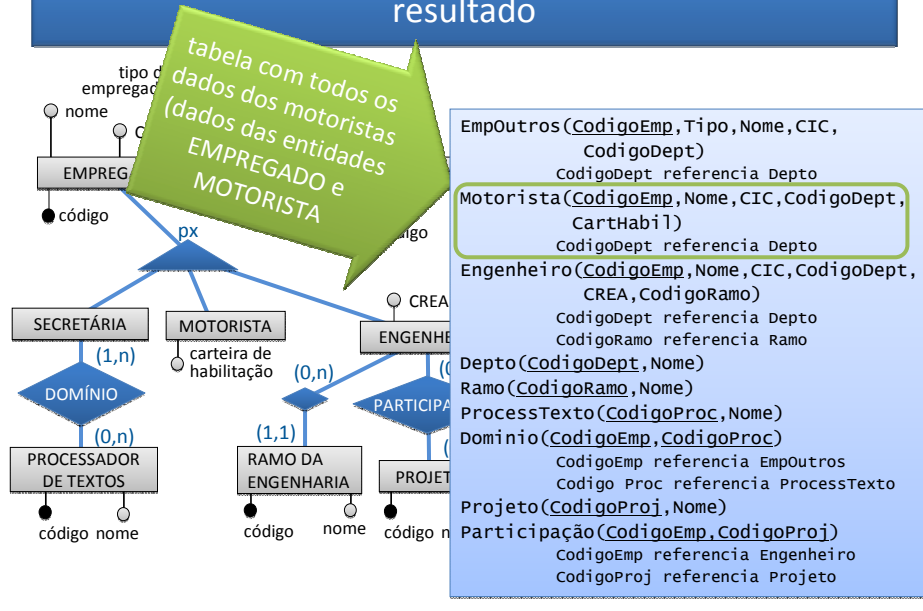
Subdivisão da entidade genérica

- Uma **tabela para cada** entidade especializada que não possua outra especialização (entidade **folha da árvore**).
- Tabela contém:
 - dados da entidade especializada
 - +
 - dados da entidade genérica.

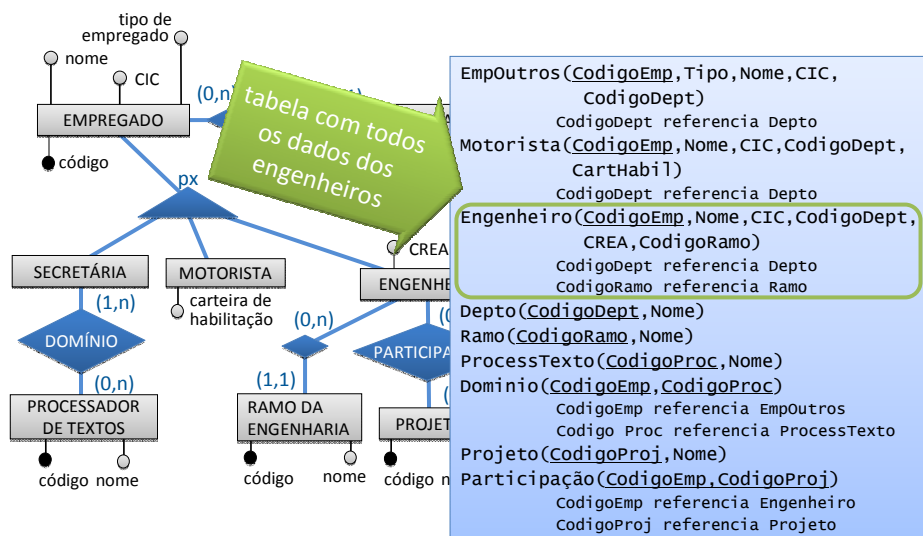
Subdivisão da entidade genérica resultado

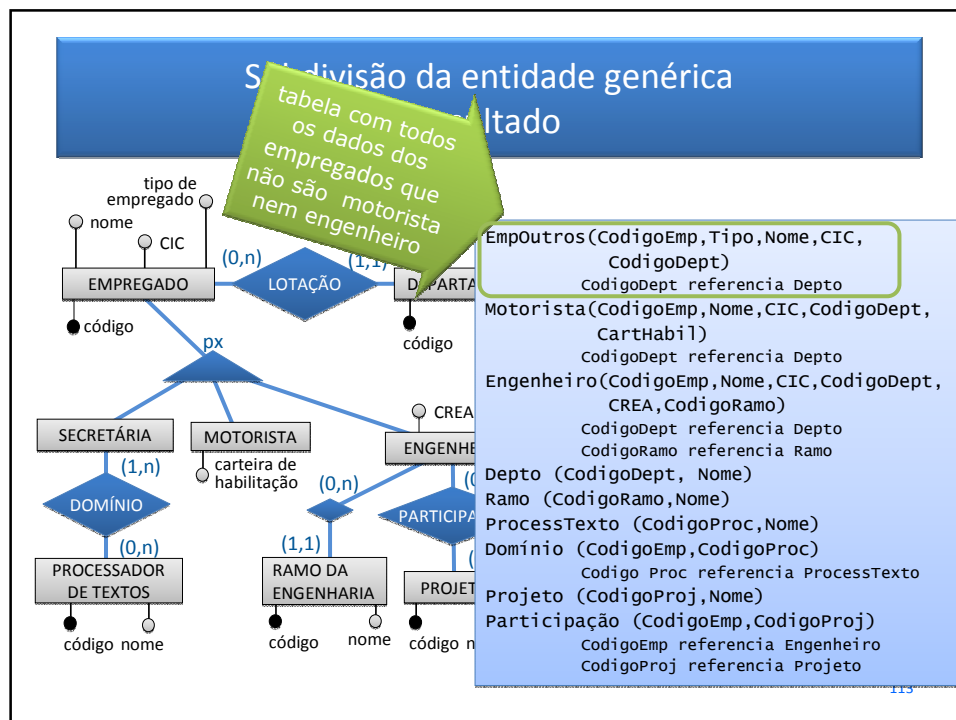


Subdivisão da entidade genérica resultado



Subdivisão da entidade genérica resultado





Subdivisão da entidade genérica

- Desvantagens:
 - Unicidade do identificador de empregado:
 - não é garantida pelo SGBD;
 - deve ser garantida pela aplicação.
 - Não há como especificar ao SGBD restrições de integridade referenciais, que façam referência ao conjunto de empregados como um todo.

©Carlos A. Heuser 114

Refinamento do modelo relacional

- Projeto (engenharia) em geral é:
 - compromisso entre o ideal e o realizável dentro das restrições de recursos impostas pela prática.
- Projeto de banco de dados é:
 - compromisso entre o ideal (regras de implementação) e o alcançável frente a limitações de performance.

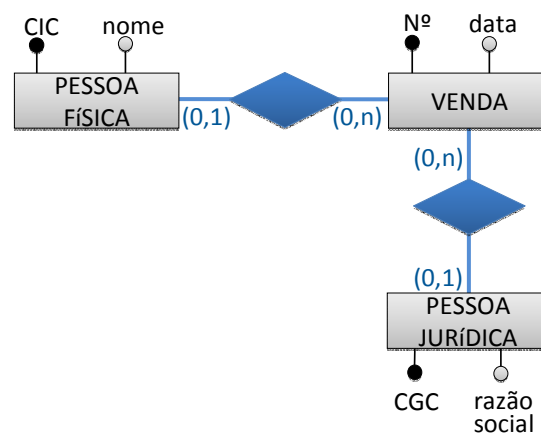
Refinamento do modelo relacional

- Algumas vezes:
 - esquema de BD criado através do uso das regras acima não atende requisitos de desempenho impostos ao sistema.
- Necessário buscar alternativa que resulte em melhor desempenho do sistema.
- Alternativas somente devem ser tentadas em último caso:
 - Do ponto de vista da programação são normalmente piores.

Refinamento do modelo relacional

- Exemplos de alternativas de projeto:
 1. Relacionamentos mutuamente exclusivos
 2. Simulação de atributos multivalorados
 3. Informações redundantes

Relacionamentos mutuamente exclusivos



Relacionamentos mutuamente exclusivos

- Implementação pelas regras:

```
PessFis(CIC, Nome)
PessJur(CGC, RazSoc)
Venda(No, data, CIC, CGC)
      CIC referencia PessFis
      CGC referencia PessJur
```

- As colunas CIC e CGC em Venda são especificadas como **opcionais**.

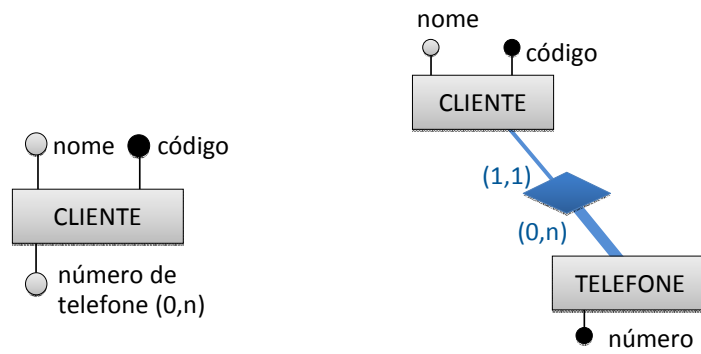
Relacionamentos mutuamente exclusivos

- Implementação alternativa:
 - Criar **uma única coluna** na qual aparece o CIC ou o CGC do comprador:

```
PessFis(CIC, Nome)
PessJur(CGC, RazSoc)
Venda(No, data, CIC/CGC, TipoCompr)
```

- Desvantagem:
 - **Não é possível** especificar ao SGBD que o campo **CIC/CGC** é **chave estrangeira**:
 - seria referência a duas tabelas.

Tratamento de atributos multivalorados



Atributos multivalorados implementação padrão

```

Cliente (CodCli, Nome)
Telefone (CodCli, Número)
CodCli referencia Cliente
  
```

Atributos multivalorados alternativa

- Condições de contorno:
 - Raros clientes possuem mais que dois telefones.
 - Quando isso ocorrer:
 - é suficiente armazenar apenas dois números.
 - Não há consultas ao banco de dados usando o número de telefone como critério de seleção.
 - Números de telefone são apenas exibidos ou impressos juntos às demais informações de cliente.

Simulação de atributos multivalorados

- Implementação “desnormalizada”:

`Cliente (CodCli, Nome, NumTel1, NumTel2)`
- Simular uma coluna multi-valorada, através da criação de diversas colunas NumTel sufixadas por um número.

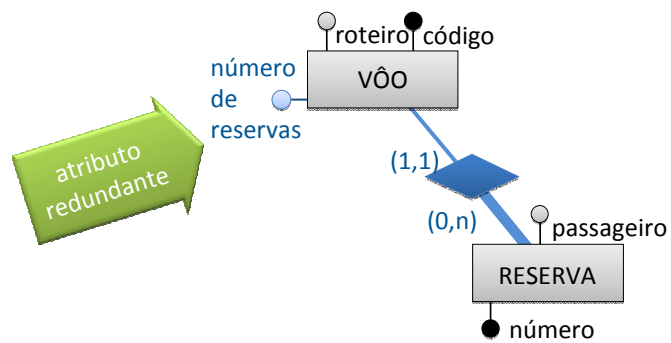
Simulação de atributos multivalorados

- Permite que os telefones de um cliente sejam obtidos mais rapidamente.
- Implica em menos espaço ocupado:
 - não é necessária chave primária da tabela Telefone.
- Inconveniente:
 - Consulta usando o número de telefone como critério de busca torna-se mais complicada.
 - Manter os telefones "alinhados à esquerda" exige rotina complexa.

Informações redundantes

- Exemplo:
 - atributos que resultam de uma operação que envolve diversas entidades do banco de dados;
 - valor destes atributos:
 - deve ser obtido com frequência ou
 - serve frequentemente como critério de busca de informações no banco de dados.
- Pode ser mais eficiente (**desempenho global** do sistema)
 - armazenar **redundantemente** o atributo derivado.

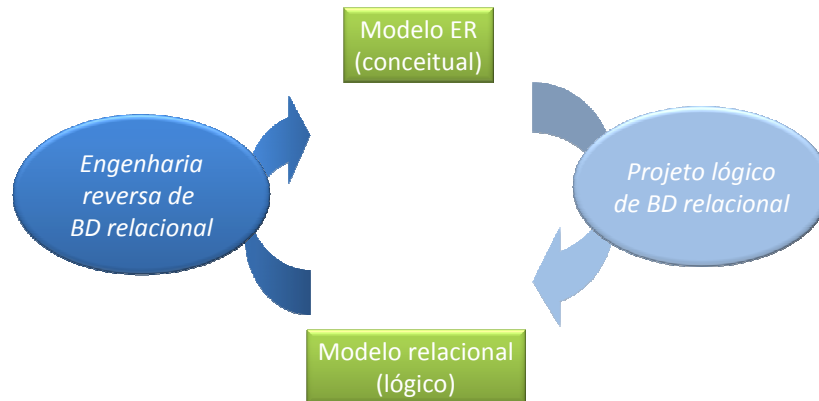
Informações redundantes Exemplo



Engenharia reversa de modelos relacionais

- Engenharia reversa:
 - parte de modelo de implementação;
 - obtem modelo de especificação (modelo conceitual).

Engenharia reversa



Engenharia reversa de modelo relacional

Passos:

1. Identificação da construção ER correspondente a cada tabela;
2. Definição de relacionamentos 1:n e 1:1;
3. Definição de atributos;
4. Definição de identificadores de entidades e relacionamentos.

Esquema relacional exemplo para engenharia reversa

Disciplina (CodDisc, NomeDisc)
Curso (CodCr, NomeCr)
Curric (CodCr, CodDisc, Obr/Opc)
 CodCr referencia Curso
 CodDisc referencia Disciplina
Sala (CodPr, CodSl, Capacidade)
 CodPr referencia Prédio
Prédio (CodPr, Endereço)
Turma (Anosem, CodDisc, SiglaTur, Capacidade, CodPr, CodSl)
 CodDisc referencia Disciplina
 (CodPr, CodSl) referencia Sala
Laboratório (CodPr, CodSl, Equipam)
 (CodPr, CodSl) referencia sala

Identificação da construção ER correspondente a cada tabela

- Uma tabela pode corresponder a:
 1. uma entidade;
 2. um relacionamento n:n;
 3. uma entidade especializada.
- Fator determinante:
 - composição da chave primária da tabela.

Tipos de tabelas para identificação de construção ER

Composição da chave primária	Construção ER correspondente
Múltiplas chaves estrangeiras	Relacionamento n:n
Toda chave primária é uma chave estrangeira	Especialização
Demais casos	Entidade

Passo #1 – identificação de construções ER

Disciplina (CodDisc, NomeDisc)
 Curso (CodCr, NomeCr)
 Curric (CodCr, CodDisc, Obr/Opc)
 CodCr referencia Curso
 CodDisc referencia Disciplina
 Sala (CodPr, CodSl, Capacidade)
 CodPr referencia Prédio
 CodSl referencia Sala
 Prédio (CodPr, Endereço)
 Turma (AnoSem, CodDisc, CodSl)
 CodDisc referencia Disciplina
 (CodPr, CodSl) referencia Sala
 Laboratório (CodPr, CodSl, Equipamento)
 (CodPr, CodSl) referencia Sala

chave primária composta por duas
 (múltiplas) chaves estrangeiras
 =
 tabela representa um relacionamento
 n:n

Passo #1 – identificação de construções ER

Disciplina (CodDisc, NomeDisc)
Curso (CodCr, NomeCr)
Curric (CodCr, CodDisc, Obr/Opc)
 CodCr referencia Curso
 CodDisc referencia Disciplina
Sala (CodPr, CodSl, Capacidade)
 CodPr referencia Prédio
Prédio (CodPr, Endereço)
Turma (Anosem, CodDisc, SiglaTur, Capacidade, CodPr, CodSl)
 CodDisc referencia Disciplina
 (CodPr, CodSl) referencia Sala
Laboratório (CodPr, CodSl, Equipam)
 (CodPr, CodSl) referencia Sala

toda chave **primária** é uma chave **estrangeira**
=
tabela representa uma **especialização**

Passo #1 – identificação de construções ER

Disciplina (CodDisc, NomeDisc)
Curso (CodCr, NomeCr)
Curric (CodCr, CodDisc, Obr/Opc)
 CodCr referencia Curso
 CodDisc referencia Disciplina
Sala (CodPr, CodSl, Capacidade)
 CodPr referencia Prédio
Prédio (CodPr, Endereço)
Turma (Anosem, CodDisc, SiglaTur, Capacidade, CodPr, CodSl)
 CodDisc referencia Disciplina
 (CodPr, CodSl) referencia Sala
Laboratório (CodPr, CodSl, Equipam)
 (CodPr, CodSl) referencia Sala

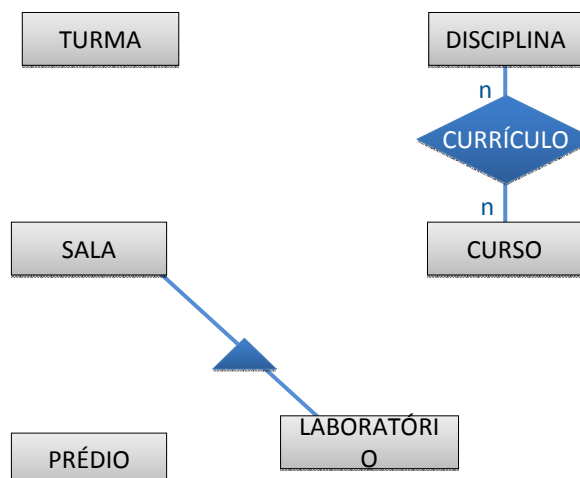
demais casos
=
tabela representa uma **entidade**

Passo #1 – identificação de construções ER

Disciplina (CodDisc, NomeDisc)
 Curso (CodCr, NomeCr)
 Curric (CodCr, CodDisc, Obr/Opc)
 CodCr referencia Curso
 CodDisc referencia Disciplina
 Sala (CodPr, CodSl, Capacidade)
 CodPr referencia Prédio
 Prédio (CodPr, Endereço)
 Turma (Anosem, CodDisc, SiglaTur, Capacidade, CodPr, CodSl)
 CodDisc referencia Disciplina
 (CodPr, CodSl) referencia Sala
 Laboratório (CodPr, CodSl, Equipam)
 (CodPr, CodSl) referencia Sala

chave primária contém uma
 única chave estrangeira, mas
 não é toda ela chave
 estrangeira
 =
 tabela representa uma
 entidade

Construções identificadas



Passo #2 - identificação de relacionamentos 1:n ou 1:1

- As chaves estrangeiras tratadas no passo anterior correspondem a:
 - relacionamentos n:n,
 - especializações.
- Chave estrangeira que não foi tratada no passo anterior representa:
 - relacionamento 1:n
 - ou
 - relacionamento 1:1
- Nem sempre, esquema não informa se é 1:1 ou 1:n

Passo #2 - identificação de relacionamentos 1:n ou 1:1

Disciplina (CodDisc, NomeDisc)
Curso (CodCr, NomeCr)
Curric (CodCr, CodDisc, Obr/Opc)
 CodCr referencia Curso
 CodDisc referencia Disciplina
Sala (CodPr, CodSl, Capacidade)
 CodPr referencia Prédio
Prédio (CodPr, Endereço)
Turma (Anosem, CodDisc, SiglaTur, Capacidade, CodPr, CodSl)
 CodDisc referencia Disciplina
 (CodPr, CodSl) referencia Sala
Laboratório (CodPr, CodSl, Equipam)
 (CodPr, CodSl) referencia Sala

chave estrangeiras ainda
não tratadas

Passo #2 - exemplo

Disciplina (CodDisc, NomeDisc)
Curso (CodCr, NomeCr)
Curric (CodCr, CodDisc, ...)
 CodCr referencia
 CodDisc referencia
Sala (CodPr, CodSl, ...)
 CodPr referencia
Prédio (CodPr, Endereço)
Turma (Anosem, CodDisc, SiglaTur, Capacidade, CodPr, CodSl)
 CodDisc referencia Disciplina
 (CodPr, CodSl) referencia Sala
Laboratório (CodPr, CodSl, Equipam)
 (CodPr, CodSl) referencia sala

campo mono-valorado

relaciona a turma com no máximo uma disciplina

Passo #2 - exemplo

Disciplina (CodDisc, NomeDisc)
Curso (CodCr, NomeCr)
Curric (CodCr, CodDisc, ...)
 CodCr referencia
 CodDisc referencia
Sala (CodPr, CodSl, ...)
 CodPr referencia
Prédio (CodPr, Endereço)
Turma (Anosem, CodDisc, SiglaTur, Capacidade, CodPr, CodSl)
 CodDisc referencia Disciplina
 (CodPr, CodSl) referencia Sala
Laboratório (CodPr, CodSl, Equipam)
 (CodPr, CodSl) referencia sala

CodDisc é obrigatório

cada turma está relacionada com ao menos uma disciplina

Passo #2 - exemplo

Disciplina (CodDisc, NomeDisc)
 Curso (CodCr, NomeCr)
 Curric (CodCr, CodDisc, Obr/Opc)
 CodCr referencia Curso
 CodDisc referencia Disciplina
 Sala (CodPr, CodSl, Equipam)
 CodPr referencia Prédio
 CodSl referencia Sala
 Prédio (CodPr, Endereço)
 Turma (Anosem, CodDisc, SiglaTur, Capacidade, CodPr, CodSl)
 CodDisc referencia Disciplina
 (CodPr, CodSl) referencia Sala
 Laboratório (CodPr, CodSl, Equipam)
 (CodPr, CodSl) referencia Sala

em turma podem aparecer várias linhas com um valor de **CodDisc**

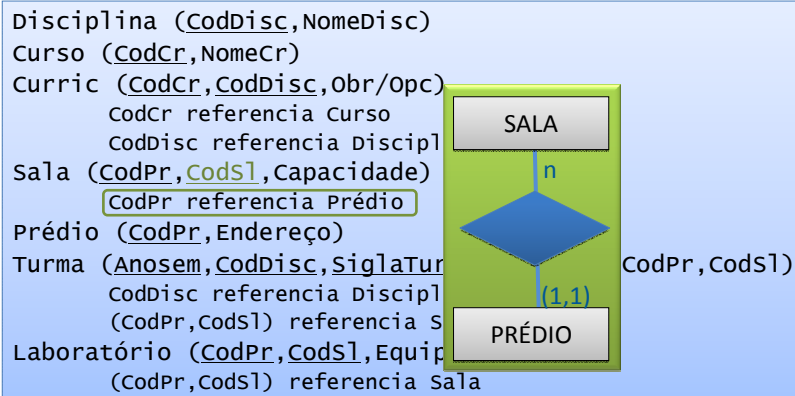
relaciona a **disciplina** com possivelmente várias **turma**

Passo #2 - exemplo

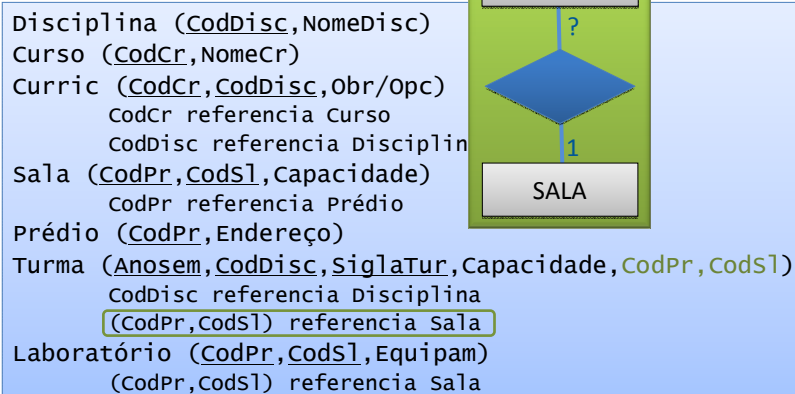
Disciplina (CodDisc, NomeDisc)
 Curso (CodCr, NomeCr)
 Curric (CodCr, CodDisc, Obr/Opc)
 CodCr referencia Curso
 CodDisc referencia Disciplina
 sala (CodPr, CodSl, Equipam)
 CodPr referencia Prédio
 CodSl referencia Sala
 Prédio (CodPr, Endereço)
 Turma (Anosem, CodDisc, SiglaTur, Capacidade, CodPr, CodSl)
 CodDisc referencia Disciplina
 (CodPr, CodSl) referencia Sala
 Laboratório (CodPr, CodSl, Equipam)
 (CodPr, CodSl) referencia Sala



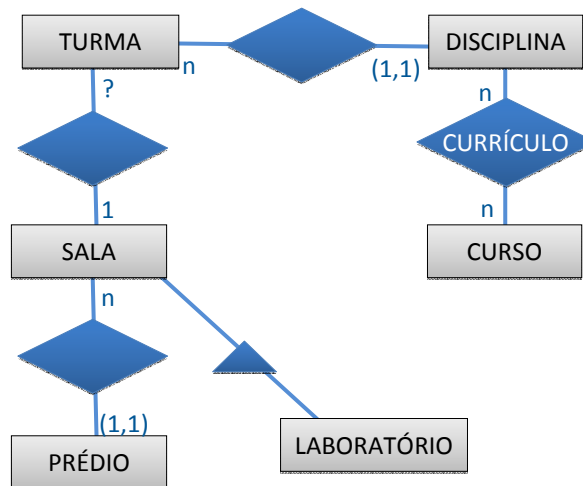
Passo #2 - exemplo



Passo #2 - exemplo



Passo #2 - Construções identificadas



©Carlos A. Heuser

147

Passo #3 - Definição de atributos

- Cada coluna **não chave estrangeira** é:
 - um **atributo** na entidade/relacionamento correspondente à tabela.
- As colunas **chave estrangeira** não correspondem a atributos:
 - correspondem a **relacionamentos**;
 - já foram tratadas nas etapas anteriores.

©Carlos A. Heuser

148

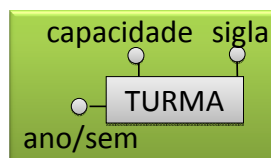
Passo #3 – exemplo

Turma (AnoSem, CodDisc, SiglaTur, Capacidade, CodPr, CodSl)
CodDisc referencia Disciplina
(CodPr, CodSl) referencia Sala

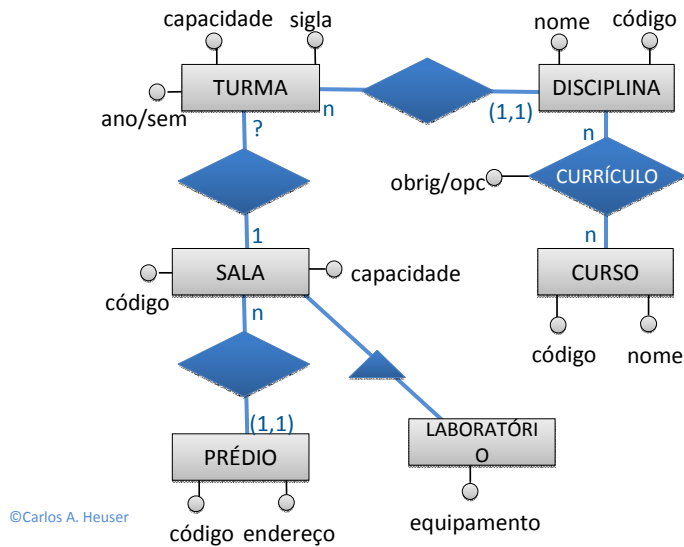
AnoSem, SiglaTur e
Capacidade
não são chaves
estrangeiras
=
correspondem a atributos

Passo #3 – exemplo

Turma (AnoSem, CodDisc, SiglaTur, Capacidade, CodPr, CodSl)
CodDisc referencia Disciplina
(CodPr, CodSl) referencia Sala



Passo #3 - Atributos identificados



Passo #4 – Definição de identificadores de entidades

Chave primária:

1. Coluna da chave primária que **não é chave estrangeira**:
 - corresponde a um **atributo identificador** da entidade ou relacionamento.
2. Coluna da chave primária que **é chave estrangeira**:
 - corresponde a um **relacionamento identificador** da entidade

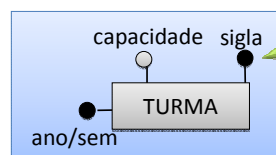
Passo #3 – exemplo

Turma (Anosem, CodDisc, SiglaTur, ...)
CodDisc referencia Disciplina
(CodPr, CodSl) referencia Sala

colunas não chave
estrangeira
=
atributo identificador

Passo #3 – exemplo

Turma (Anosem, CodDisc, SiglaTur, ...)
CodDisc referencia Disciplina
(CodPr, CodSl) referencia Sala



colunas não chave
estrangeira
=
atributo
identificador

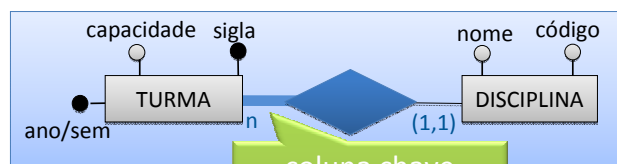
Passo #3 – exemplo

Turma (Anosem, CodDisc, SiglaTur, ...)
CodDisc referencia Disciplina
(CodPr, CodSl) referencia Sala

coluna chave
estrangeira
=
relacionamento
identificador

Passo #3 – exemplo

Turma (Anosem, CodDisc, SiglaTur, ...)
CodDisc referencia Disciplina
(CodPr, CodSl) referencia Sala



coluna chave
estrangeira
=
relacionamento
identificador

Passo #3 – Definição de identificadores de entidades

