cin.ufpe.br

Centro de Informática

U · F · P · E

Universidade Federal de Pernambuco

Transformação ER/Relacional

Por:

Robson do Nascimento Fidalgo
rdnf@cin.ufpe.br

Transformação ER/Relacional

- Um modelo ER pode ser implementado em diversos esquemas relacionais equivalentes
  - Desempenho, facilidade e dificuldade de construção e manutenção farão a diferença entre eles.
- As regras que serão apresentadas irão fornecer um modelo relacional inicial.
  - O modelo poderá sofrer alterações até obter-se um modelo satisfatório (que ofereça bom desempenho, manutenção,...)
  - Por exigência da aplicação outras regras podem ser usadas

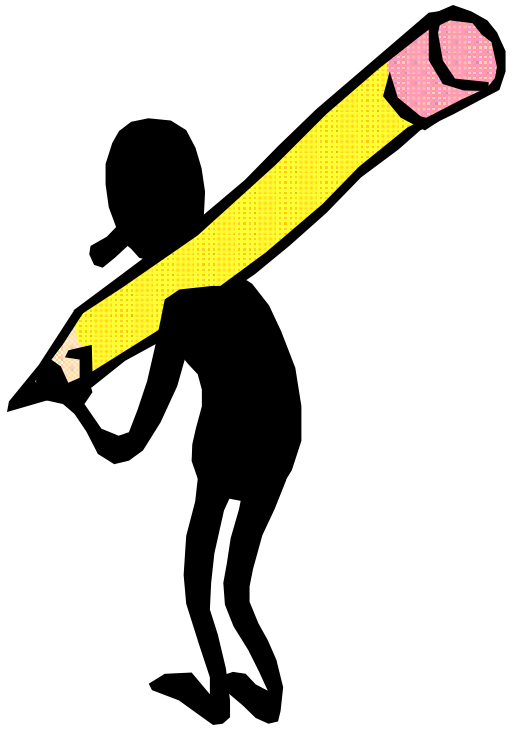
Transformação ER/Relacional

- As regras foram definidas tendo em vista dois objetivos básicos:
  - Bom desempenho
    - Bom desempenho significa basicamente diminuir acesso ao disco
      - Acessos ao disco consomem o maior tempo na execução de uma instrução de banco de dados
  - Simplificação do desenvolvimento e da aplicação de BD

- Além destes dois pontos centrais, as regras também objetivam economizar espaço de armazenamento.
  - Isto não é muito importante (atualmente armazenar é barato)
  - O "X" da questão é equacionar o custo-benefício da economia de armazenamento com os outros dois pontos centrais

Transformação ER/Relacional

- As regras foram definidas tendo por base, entre outros, os seguintes princípios :
  - Evitar junções
  - Diminuir o número de chaves
  - Evitar campos opcionais



Evitar junções

- Junção é uma operação que busca dados de diversas linhas através da igualdade de campos
- SGBDR geralmente armazenam dados continuamente no disco
  - Um único acesso ao disco traz todos os dados do “bloco” para a RAM
- Junções envolvem comparações entre diversas linhas
  - Junções requerem diversos acessos a disco (minimizam o desempenho)

Assim, quando possível, evite usar junções !

Evitar junções

Evitar Junções:

Ter os dados necessários ao resultado da consulta em uma única linha da tabela

Diminuir o número de chaves

- Para implementar eficientemente o controle de chaves primárias o SGBD usa índices para cada chave primária.
  - Índices tendem a ocupar espaço considerável em disco
  - A Inserção ou remoção de entradas em um índice podem exigir diversos acesso a disco

Assim, quando possível, diminua a quantidade de chaves primárias !

Diminuir o número de chaves

Diminuir chaves:

Ter os dados subordinados as ch. primárias em uma única tabela

Diminuir o número de chaves

- Exemplo:

1) Cliente (CPF, Nome, NomeContato, Endereço, E-Mail, Telefone)

Esta é melhor pois cria-se um único índice

2) Cliente (CPF ,Nome, NomeContato)

ClienteContato (CPF ,Endereço, E-Mail, Telefone) CPF referencia Cliente

Nesta há dois índices com exatamente as mesmas entradas

Evitar campos opcionais

- Campo opcional = campo que pode ser VAZIO (NULL).
  - SGBDR não desperdiçam espaço pelo fato de campos de uma linha estarem vazios (usam técnicas de compressão de dados)
  - Campo opcional não tem influência na performance

- Problema: Qdo o SGBDR não controla a obrigatoriedade do campo (qdo o preenchimento de um campo depende de outro)
  - EX: Se a pessoa for casada deve preencher Campo X, Y e Z
  - Neste caso, o controle da obrigatoriedade deve ser feito por programação.

Assim, quando possível, evite campos opcionais!

Evitar campos opcionais

Evitar campos opcionais:

Ter apenas campos obrigatórios

Esta regra é mais "fraca"
do que as precedentes

Regras Transformação ER/Relacional

- Passos da transformação ER para Relacional
  - Tradução inicial de entidades e respectivos atributos
  - Tradução de relacionamentos e respectivos atributos
  - Tradução de generalizações/ especializações

Implementação inicial de entidades

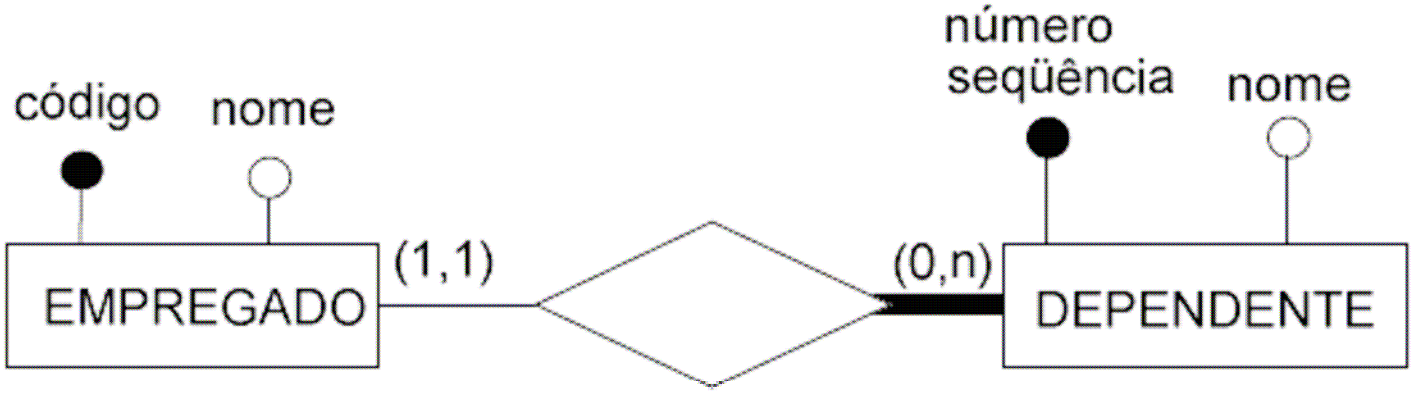
- Cada **entidade** é traduzida para uma **tabela**
- Cada **atributo** da entidade define uma **coluna** desta tabela
- **Atributos identificadores** da entidade correspondem a **chave primária** da tabela.
- Esta é uma tradução inicial:
  - As próximas regras podem fundir tabelas

Implementação de entidade

[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

Pessoa (Codigo,Nome,Endereço,DtNasc,DtAdm)

Tradução de entidade fraca

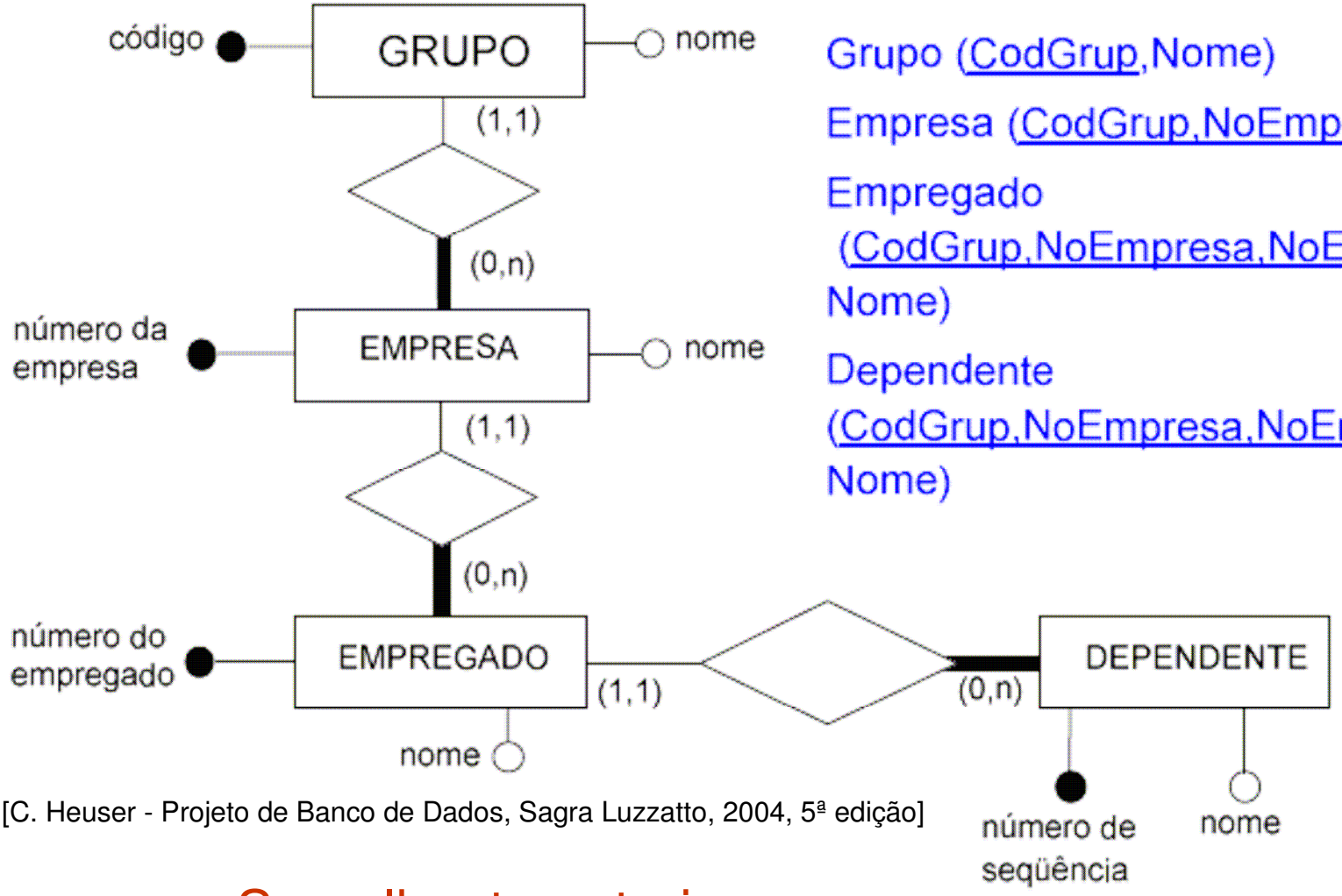


[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

Dependente (Codigo,NoSeq,Nome)

Faz-se a tradução das entidades fracas em tabelas, seus atributos em campos e adiciona-se a chave primária da entidade forte na chave primária da entidade fraca.

Tradução de entidade fraca-recursão



Grupo (CodGrup,Nome)

Empresa (CodGrup,NoEmpresa,Nome)

Empregado (CodGrup,NoEmpresa,NoEmpreg, Nome)

Dependente (CodGrup,NoEmpresa,NoEmpreg,NoSeq, Nome)

[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

Semelhante anterior.

Entretanto faz-se sucessivamente !

Nomes de colunas

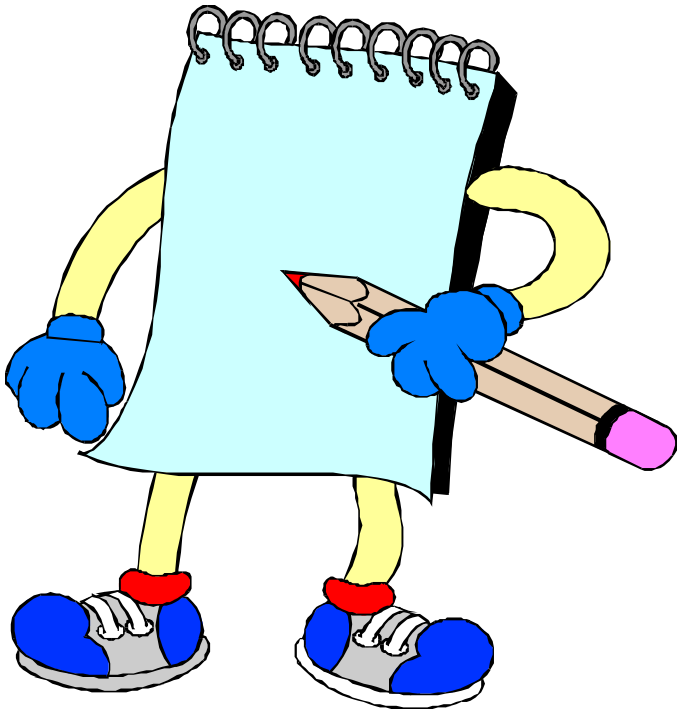
- São freqüentemente referenciados em programas
  - Use nomes das colunas que sejam curtos e significativos. Isto facilita a vida do programador
- No SGBDR os nomes das colunas não podem conter brancos
- Não transcreva os nomes de atributos para nomes de colunas.
  - Abrevie os nomes de atributos compostos de diversas palavras
  - EX:DtNasc, EstCivil, CartMotorista,... (use um padrão Data → Dt)
- Nomes de colunas não necessitam conter o nome da tabela
  - Prefira usar **Nome** do que usar **NomeCli, NomeFor, NomePro**
  - Em SQL pode-se fazer Tabela.Coluna (EX: Cliente.Nome)

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
  - CodCli, CodFor, CodProd

Implementação de relacionamento

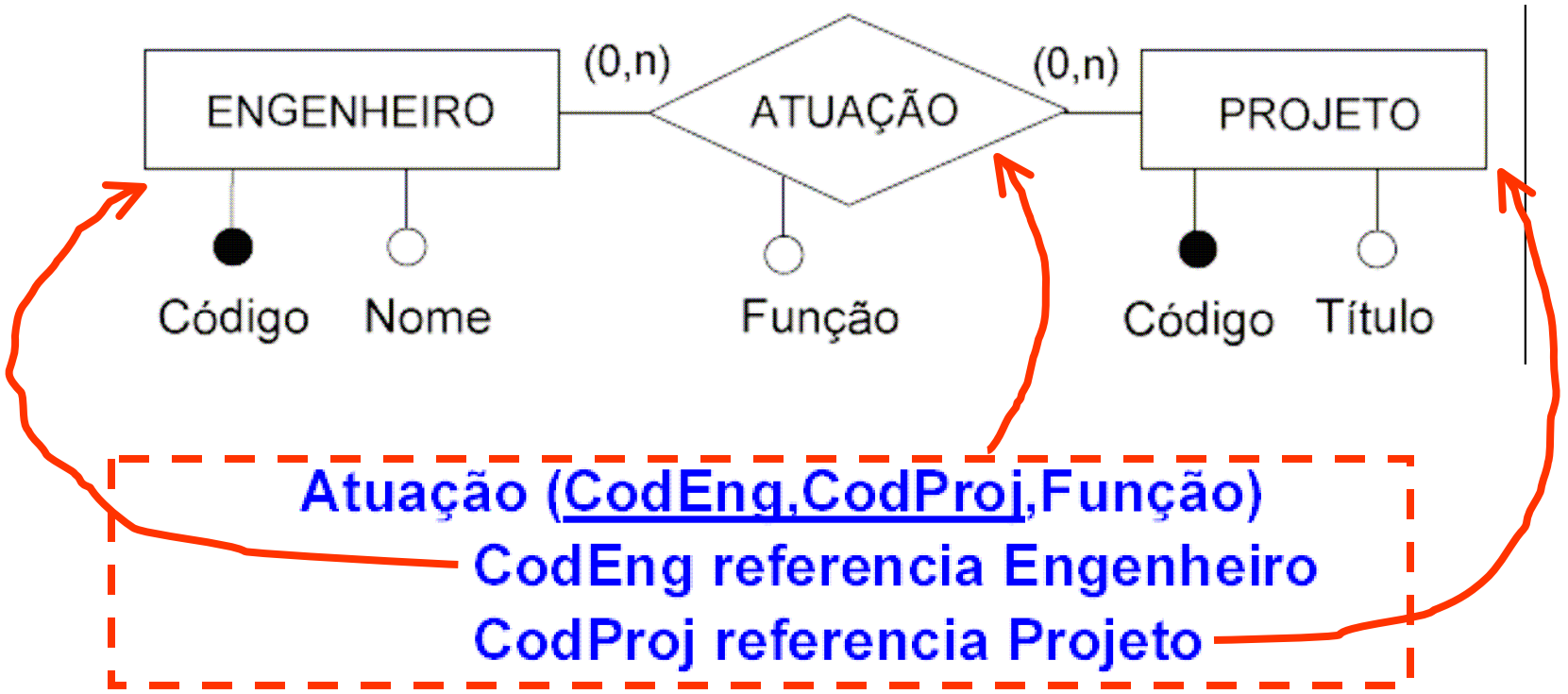
- Alternativas básicas
  - Tabela própria
  - Adição de colunas a uma das tabelas
  - Fusão de tabelas



- A escolha de umas dessas alternativa depende da cardinalidade (máxima e mínima do relacionamento)

Tabela própria

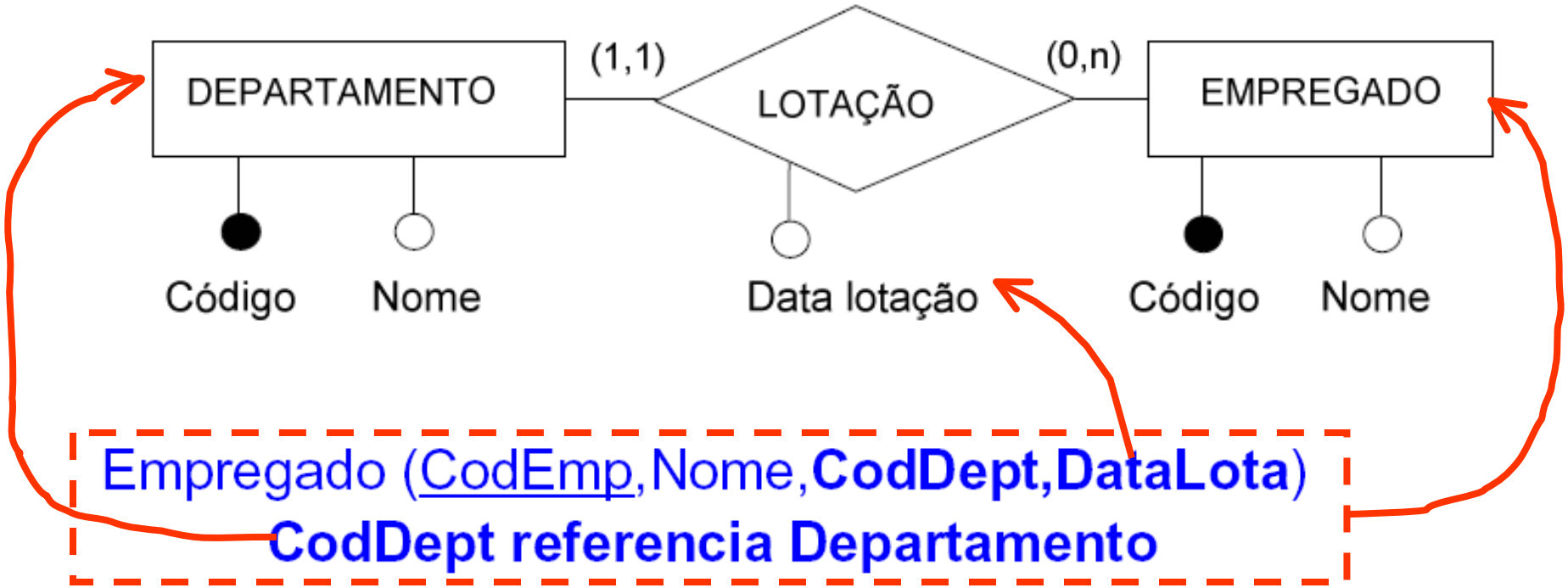
- Nesta, o relacionamento é implementado através de uma tabela própria. Esta contém os atributos identificadores das entidades participantes + os atributos do relacionamento



Adaptado de [C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

Adição de colunas

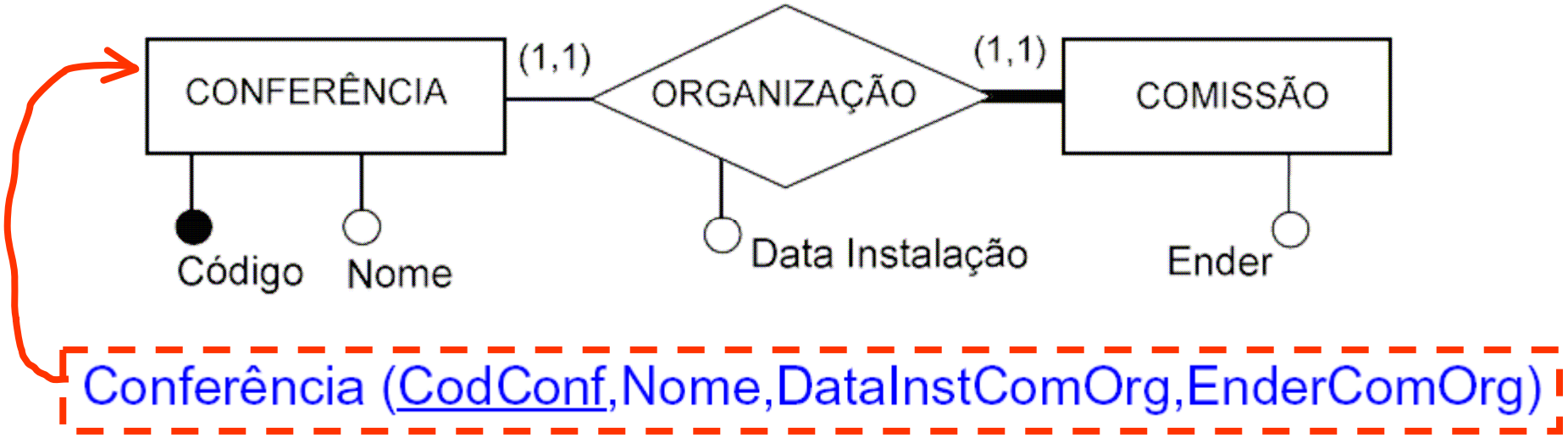
- Nesta, o relacionamento é implementado através da inserção de colunas na tabela oposta a multiplicidade máxima 1.
  - Só é possível quando tem-se relacionamentos com pelo menos uma cardinalidade máxima 1



Adaptado de [C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

Fusão de tabelas

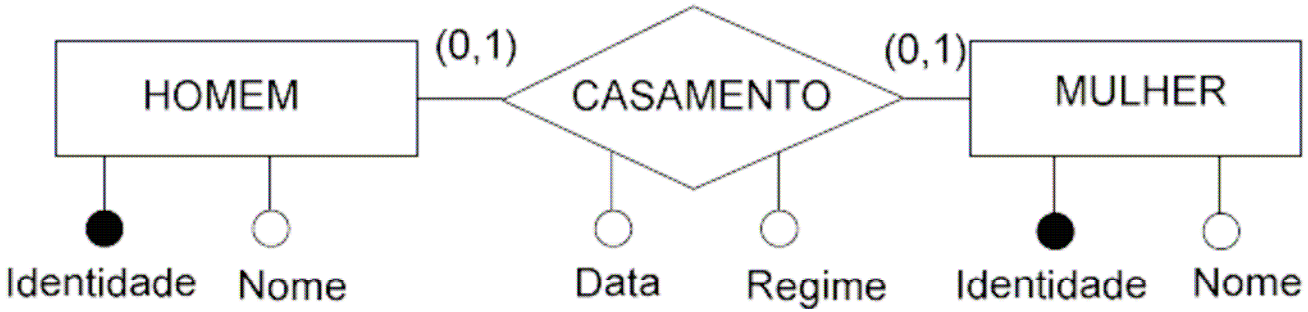
- Nesta, o relacionamento é implementado através da união das tabelas participantes.
  - Só é possível para relacionamentos 1:1



Adaptado de [C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

Implementação de relacionamento

- 1:1 – onde ambas entidades têm relacionamentos opcionais



Mulher (IdentM, Nome, **IdentH, Data, Regime**)

IdentH referencia Homem

Homem (IdentH, Nome)

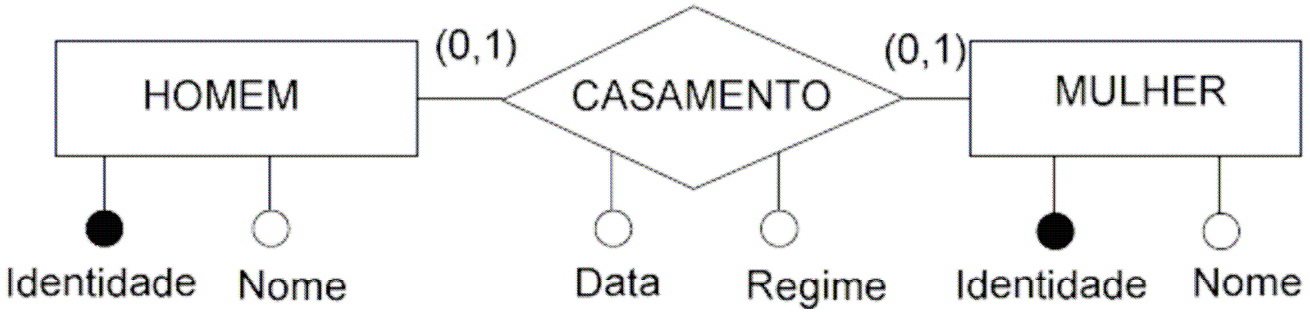
Deve ser único (unique)

Adaptado de [C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

- Melhor tradução:
  - Adição de Colunas (neste caso escolheu-se Mulher arbitrariamente e IdentH tem que ser única)

Implementação de relacionamento

- 1:1 – onde ambas entidades têm relacionamentos opcionais



Mulher (IdentM,Nome)

Homem (IdentH,Nome)

Casamento (IdentM,IdentH,Data,Regime)

IdentM referencia Mulher

IdentH referencia Homem

Deve ser obrigatório e único (not null e unique)

Foi uma escolha arbitrária, poderia ter sido IdentH. IdentH tem que ser única.

Adaptado de [C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

- Tradução alternativa:
  - Tabela Própria (neste caso Casamento torna-se uma tabela)

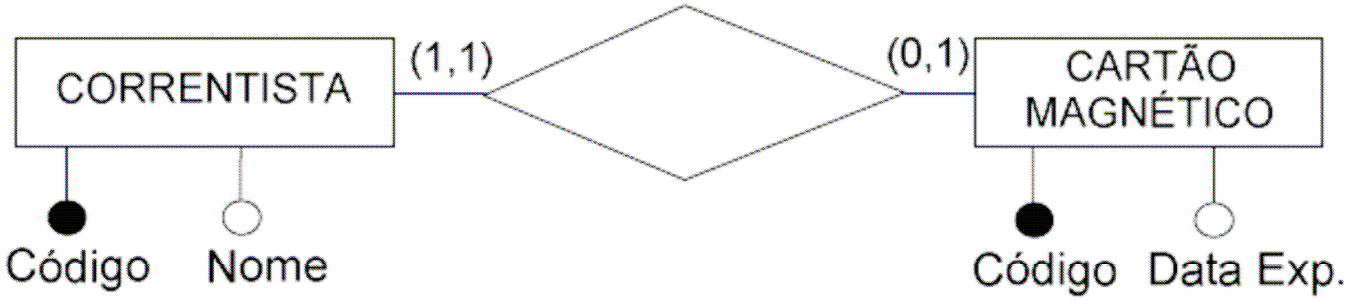
Implementação de relacionamento

- 1:1 – onde ambas entidades têm relacionamentos opcionais
- Discussão:
  - A solução por adição de colunas é a melhor, pois minimiza a quantidade de junções e chaves, entretanto, pode-se ter atributos opcionais, os quais devem ser tratados via programação
  - A solução de tabela própria é aceitável, mas a de fusão de tabela é semanticamente inviável para o contexto

Implementação de relacionamento

- 1:1 – onde uma entidade tem relacionamento opcional e a outra tem relacionamento obrigatório

Adaptado de [C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]



Correntista (CodCorrent,Nome,CodCartão,DataExp)

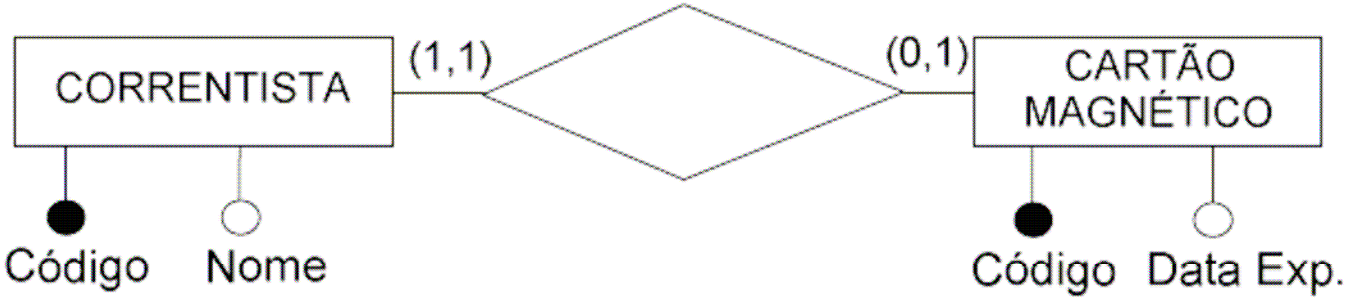
Deve ser único (unique)

- Melhor tradução:
  - Fusão de Tabela

Implementação de relacionamento

- 1:1 – onde uma entidade tem relacionamento opcional e a outra tem relacionamento obrigatório

Adaptado de [C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]



Correntista (CodCorrent,Nome)

Cartão(CodCartão,DataExp,**CodCorrent**)

CodCorrent referencia Correntista

Deve ser obrigatório e único (not null e unique)

- Tradução alternativa:
  - Adição de Colunas

Implementação de relacionamento

- 1:1 – onde uma entidade tem relacionamento opcional e a outra tem relacionamento obrigatório

- Discussão:

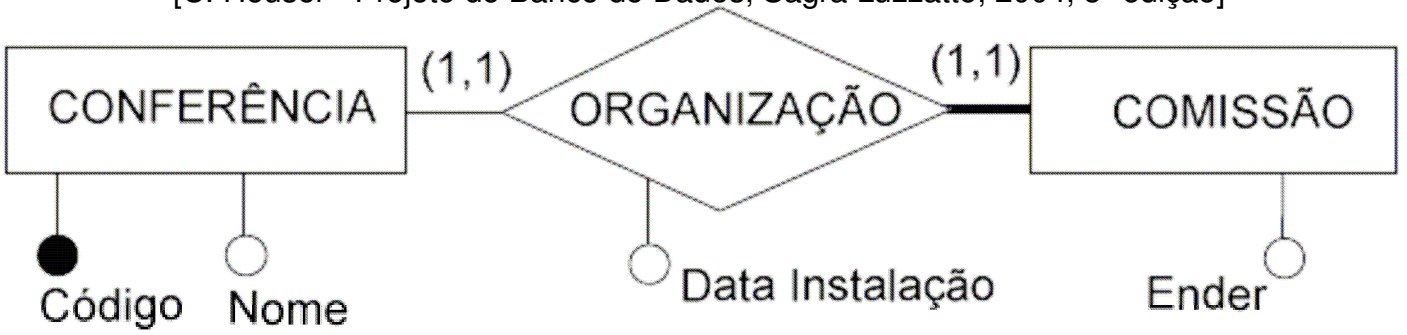
  - A solução por fusão de tabelas é a melhor, pois elimina a necessidade de junção e diminuindo a qtd de chave. Entretanto, pode-se ter atributos opcionais, os quais devem ser tratados por programação

  - A solução por tabela própria é pior que a solução por adição de colunas, pois gera uma quantidade maior de junções e chaves.
    - Ambas não têm campos opcionais.

Implementação de relacionamento

- 1:1 – onde ambas entidades têm relacionamento obrigatório.

[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]



Conferência (CodConf,Nome,DataInstComOrg,EnderComOrg)

Deve ser obrigatório (not null)

- Melhor tradução:
  - Fusão de Tabelas

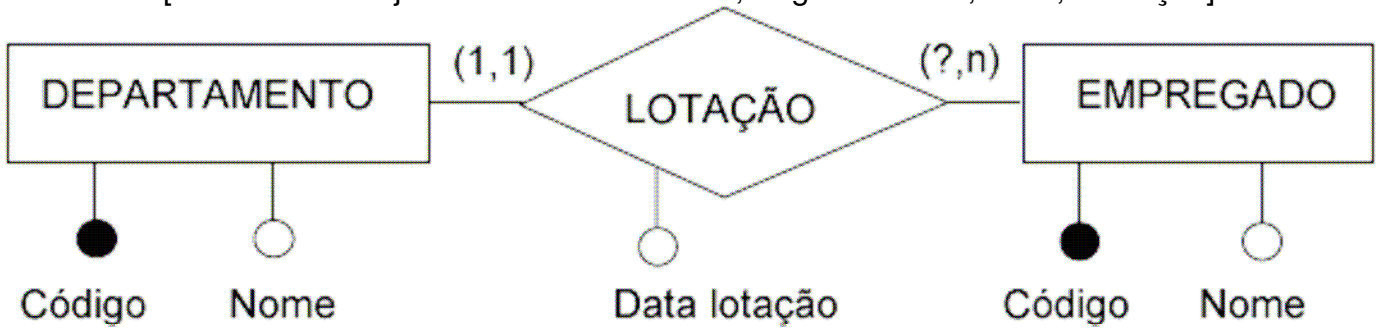
Implementação de relacionamento

- 1:1 – onde ambas entidades têm relacionamento obrigatório.
- Discussão:
  - A solução por fusão de tabelas é a melhor, pois elimina a necessidade de junção, diminui a quantidade de chave e não tem atributos opcionais
  - Nenhumas das duas outras abordagens são soluções adequadas. Pois:
    - As entidades que participam do relacionamento seriam representadas através de duas tabelas distintas, mas com a mesma chave primária e relação um-para-um entre suas linhas
    - Maior número de junções
    - Maior número de chaves

Implementação de relacionamento

- Relacionamentos 1:N onde a entidade que tem multiplicidade máxima 1 é obrigatória.

[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]



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

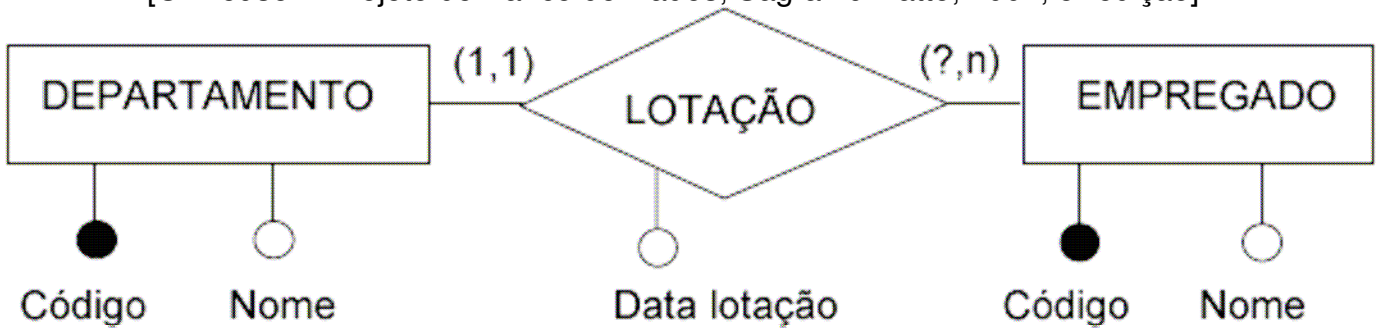
Deve ser obrigatório (not null)

- Melhor tradução:
  - Adição de Colunas

Implementação de relacionamento

- 1:N – onde a entidade que tem multiplicidade máxima 1 é obrigatória.

[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]



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

Deve ser obrigatório (not null)

- Tradução Alternativa:
  - Tabela Própria

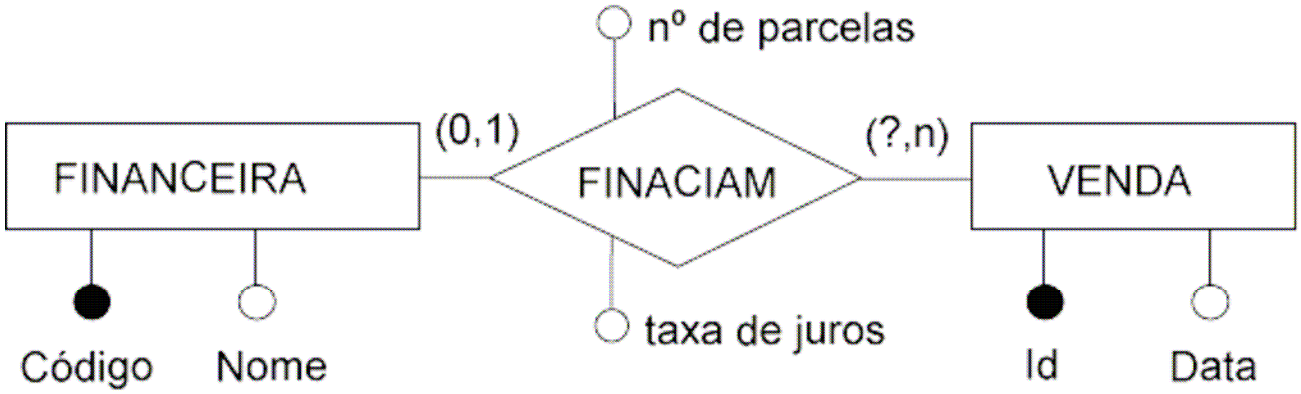
Implementação de relacionamento

- 1:N – onde a entidade que tem multiplicidade máxima 1 é obrigatória.
- Discussão:
  - A solução por adição de colunas é a melhor, pois elimina a necessidade de junção extra, diminui a quantidade de chave e não tem atributos opcionais.
  - A solução de tabela própria é aceitável, mas evite usá-la, pois com esta, tem-se uma junção e uma chave a mais. Quanto a fusão de tabela, esta é inviável para o contexto, pois implicaria em uma redundância desnecessária de dados sobre departamento.

Implementação de relacionamento

- 1:N – onde a entidade que tem multiplicidade máxima 1 é opcional.

[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

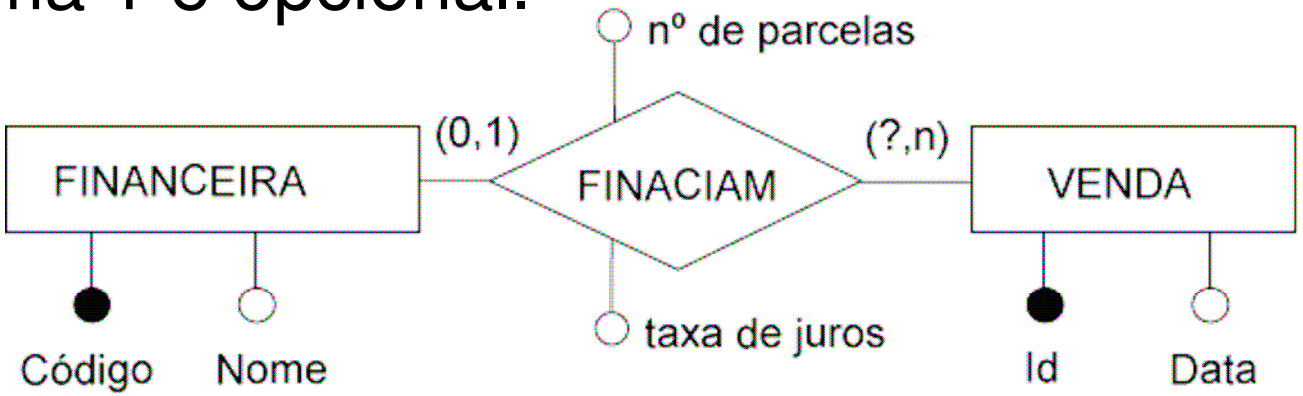


Financeira (CodFin,Nome)
Venda (IdVend,Data,**CodFin,NoParc,TxJuros**)
CodFin referencia Financeira

- Melhor tradução:
  - Adição de Colunas

Implementação de relacionamento

- 1:N – onde a entidade que tem multiplicidade máxima 1 é opcional.



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

Deve ser obrigatório (not null)

[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

- Tradução alternativa:
  - Tabela Própria

Implementação de relacionamento

- 1:N – onde a entidade que tem multiplicidade máxima 1 é opcional.
- Discussão:
  - A solução por adição de colunas é a melhor, pois elimina a necessidade de junção extra e diminui a quantidade de chave. Entretanto, tem-se atributos opcionais, os quais devem ser tratados por programação
  - A solução de tabela própria é aceitável (tem junção a mais, mas não tem atributos opcionais). Contudo, a de fusão de tabela é inviável para o contexto, pois implicaria em uma redundância desnecessária de dados sobre financeira.

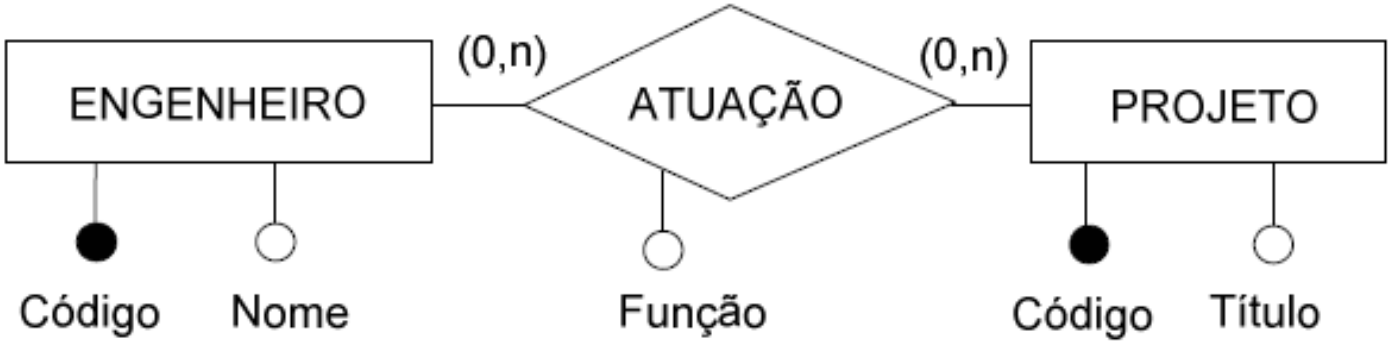
Implementação de relacionamento

Note:

Em relacionamentos 1:N, sempre a melhor tradução é a de Adição de Colunas

Implementação de relacionamento

- N:N – em todas as combinações de multiplicidade, á única tradução possível é de Tabela Própria (onde a Chave Primária é composta).



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

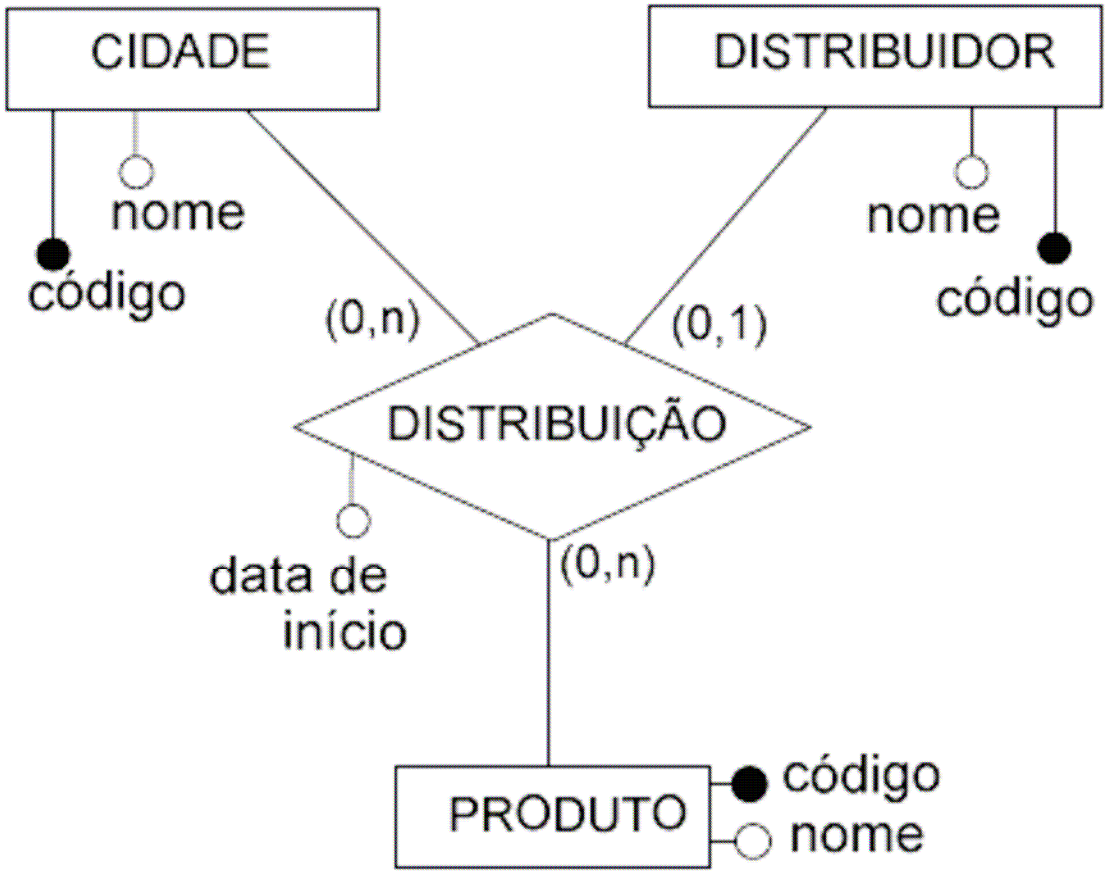
[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

Implementação de relacionamento

- Relacionamentos com grau maior que 2.
  - As regras vistas até agora são aplicáveis apenas a relacionamentos binários.
  - Para relacionamento com cardinalidade/grau maior que 2, não são definidas regras específicas. Mas no geral, o relacionamento é transformado em um entidade, onde sua CP é normalmente composta de forma a atender a regra de negócio.
  - Se não for afetar a regra de negócio transforme relacionamentos com cardinalidade/grau maior que 2 para relacionamentos binários.

Implementação de relacionamento

- Relacionamentos com grau maior que 2.

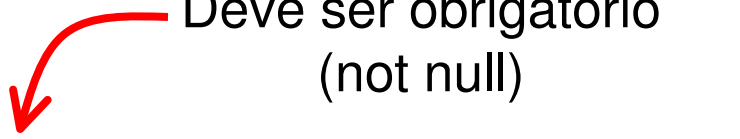


[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

Implementação de relacionamento

- Relacionamentos com grau maior que 2.

Produto (CodProd,Nome)
Cidade (CodCid,Nome)
Distribuidor (CodDistr,Nome)
Distribuição (CodProd,CodDistr,CodCid,DataInicio)
CodProd referencia Produto
CodDistr referencia Distribuidor
CodCid referencia Cidade



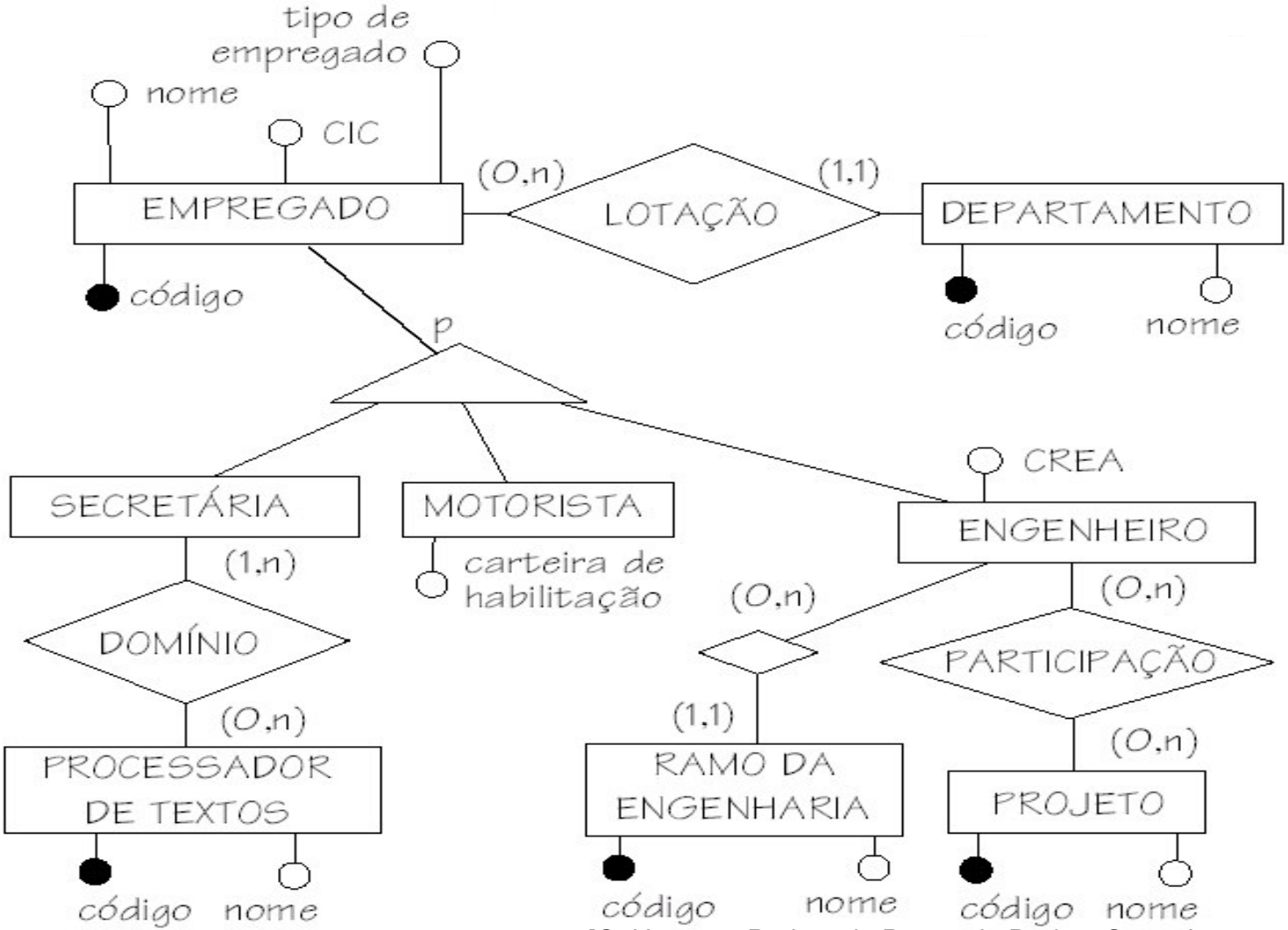
[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

Implementação de relacionamento

- Implementação de generalização/especialização
  - Duas alternativas básicas
    - Uso de uma única tabela para toda hierarquia
    - Uso de uma tabela para cada entidade

Implementação de relacionamento

- Implementação de generalização/especialização



[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

Implementação de relacionamento

- Implementação de generalização/especialização
  - Uso de uma única tabela para toda hierarquia
    - Todas tabelas referentes às especializações são fundidas em uma única tabela, a qual contém:
      - Chave primária correspondente ao identificador da entidade mais genérica
      - Caso não exista uma coluna tipo, a mesma deve ser adicionada
      - Uma coluna para cada atributo da entidade genérica
      - Uma coluna para cada chave estrangeira da entidade genérica
      - Uma coluna para cada atributo de cada entidade especializada
        - » Estas não devem ser colunas obrigatórias
      - Uma coluna para cada chave estrangeira da entidade especializada
        - » Estas também não devem ser obrigatórias

Implementação de relacionamento

- Implementação de generalização/especialização
  - Uso de uma única tabela para toda hierarquia

Emp (CódigoEmp,Tipo,Nome,CIC,CodigoDept,
CartHabil,CREA,CódigoRamo)
CódigoDept referencia Depto
CódigoRamo referencia Ramo
Depto (CódigoDept, Nome)
Ramo (CódigoRamo,Nome)
ProcessTexto (CódigoProc,Nome)
Domínio (CódigoEmp,CódigoProc)
CódigoEmp referencia Emp
CódigoProc referencia ProcessTexto
Projeto (CódigoProj,Nome)
Participação (CódigoEmp,CodigoProj)
CódigoEmp referencia Emp
CódigoProj referencia Projeto

[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

Implementação de relacionamento

- Implementação de generalização/especialização
  - Uso de uma tabela para cada entidade
    - Criar uma tabela para cada entidade que compõe a hierarquia
    - Incluir a chave primária da tabela correspondente à entidade genérica, em cada tabela correspondente a uma entidade especializada

Implementação de relacionamento

- Implementação de generalização/especialização
  - Uso de uma tabela para cada entidade

Emp (CódigoEmp,Tipo,Nome,CIC,CódigoDept)
CódigoDept referencia Depto
Motorista(CódigoEmp,CartHabil)
CódigoEmp referencia Emp
Engenheiro(CódigoEmp,CREA,CódigoRamo)
CódigoEmp referencia Emp
CódigoRamo referencia Ramo
Depto (CódigoDept, Nome)
Ramo (CódigoRamo,Nome)
ProcessTexto (CódigoProc,Nome)
Domínio (CódigoEmp,CódigoProc)
CódigoEmp referencia Emp
Código Proc referencia ProcessTexto
Projeto (CódigoProj,Nome)
Participação (CódigoEmp,CódigoProj)
CódigoEmp referencia Emp
CódigoProj referencia Projeto

[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]

Implementação de relacionamento

- Implementação de generalização/especialização

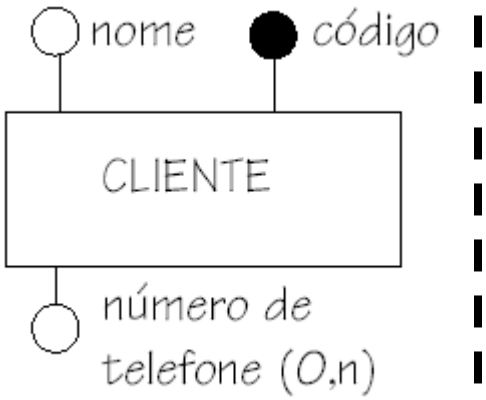
- Discussão:
  - O uso de uma única tabela para toda hierarquia minimiza junções e diminui a quantidade de chaves, entretanto, tem-se atributos opcionais, os quais devem ser tratados por programação.

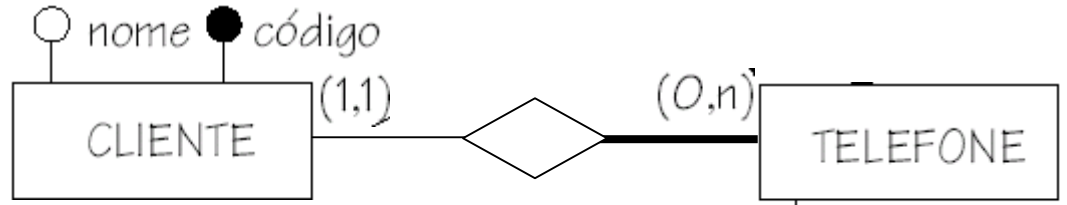
  - A solução com uso de uma tabela para cada entidade não tem atributos opcionais, mas apresenta um número maior de junções e chaves.

Implementação de Atributos Multivalorados

- Atributos Multivalorados
  - Transformar o atributo multivalorado em uma nova tabela fraca, incluindo na sua CP, a CP da tabela origem

[C. Heuser - Projeto de Banco de Dados, Sagra Luzzatto, 2004, 5ª edição]





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

  - ATENÇÃO: Deve-se avaliar a real necessidade de se criar uma nova tabela (evitar junção !)
    - Um cliente pode ter N telefones, mas se dois são suficientes então:

      Cliente (CodCli,Nome,NumTel1,NumTel2)

    - Inconveniente:
      - Consulta por telefone serão mais complicadas, pois devem referenciar todas as colunas referentes ao atributo multivalorado.

cin.ufpe.br

Centro de Informática

U · F · P · E

Universidade Federal de Pernambuco