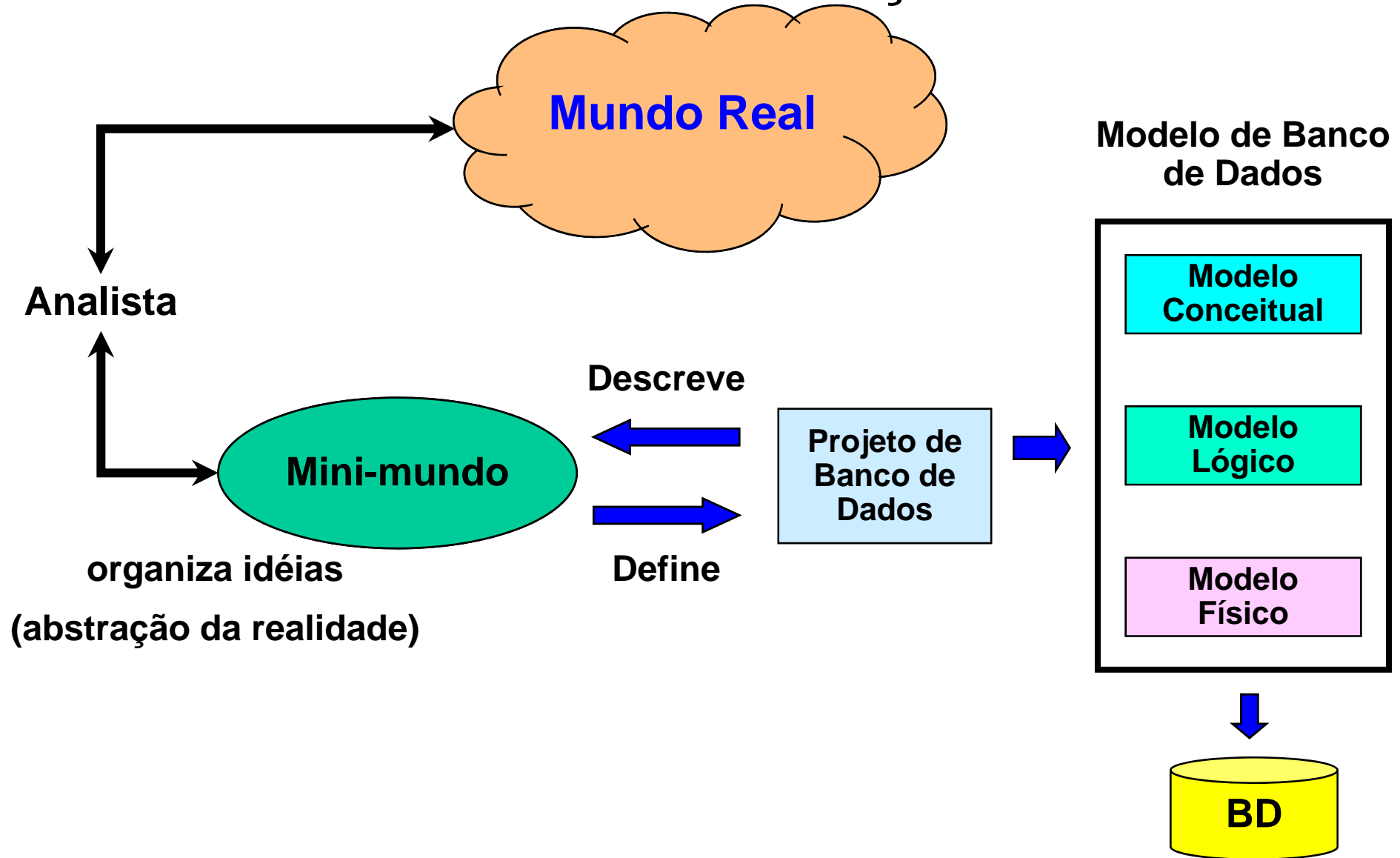


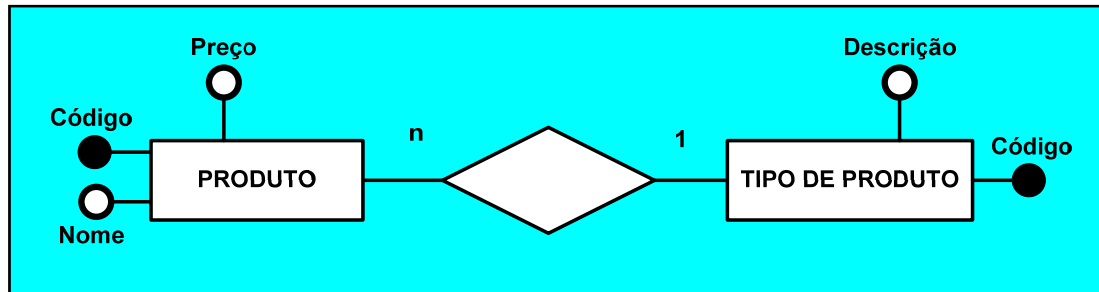
Níveis de Abstração



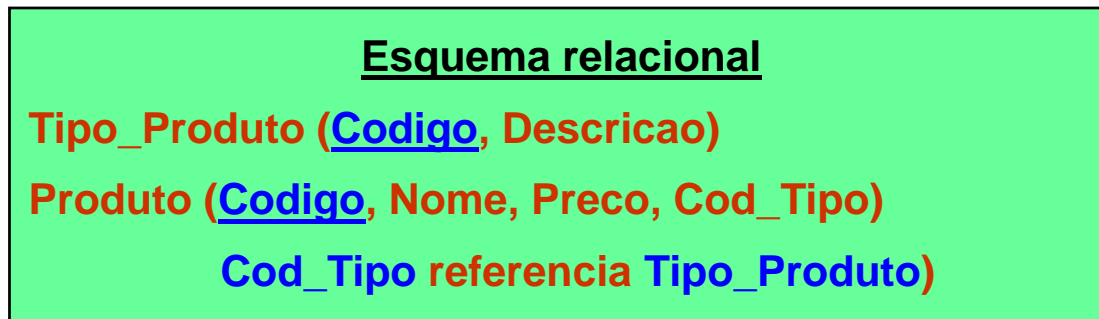
Transformações entre modelos

Projeto de Banco de Dados

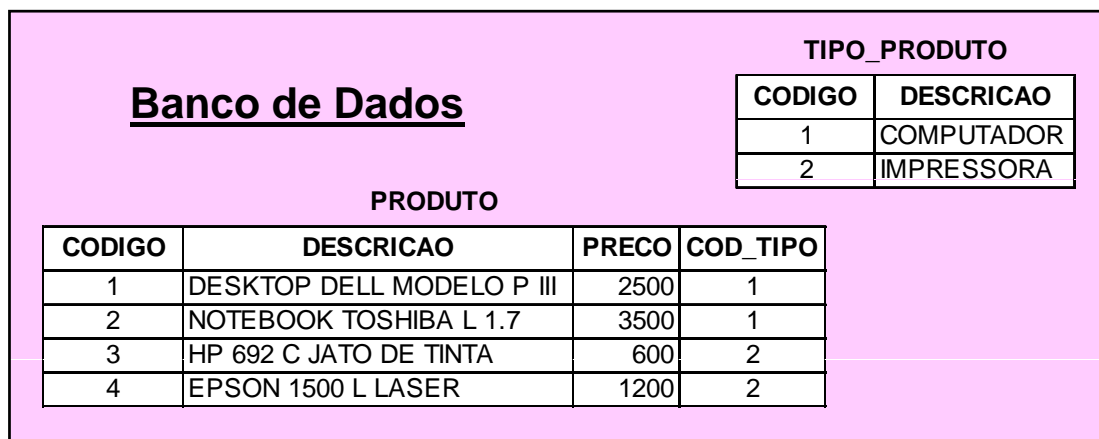
Fases:



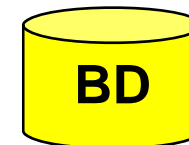
Modelo conceitual



Modelo lógico



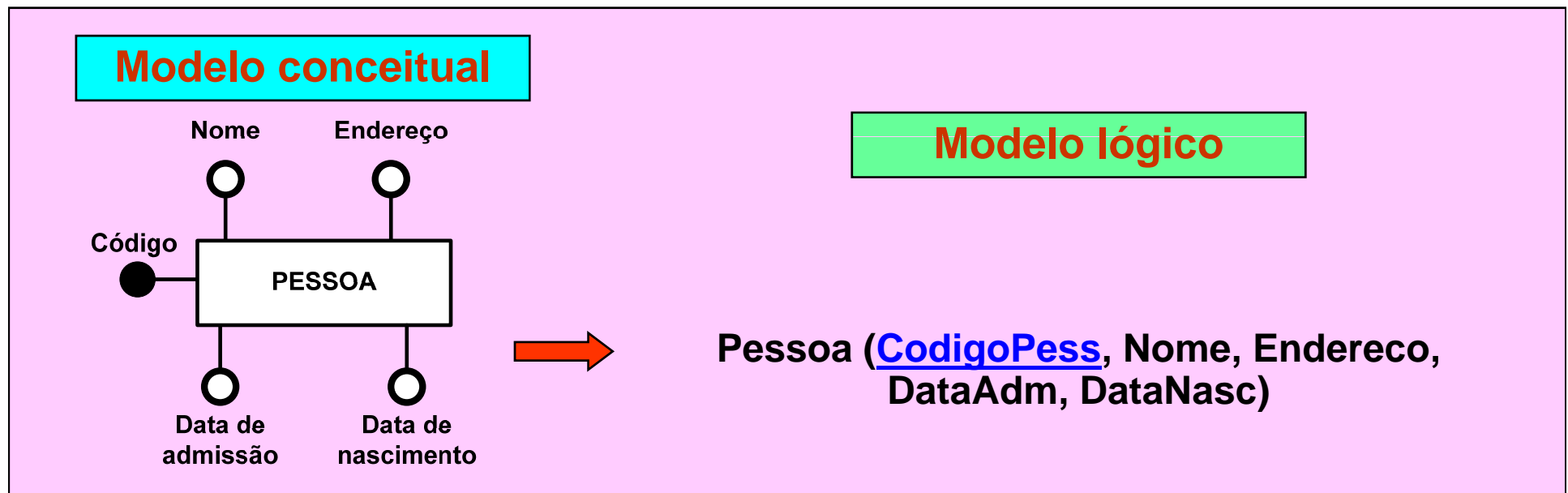
Modelo físico



Transformação de DER para Relacional

■ Implementação de entidades:

- Entidade **normalmente** transforma-se em tabela:
 - **Atributo da entidade (“simples”):** coluna da tabela
 - ❖ **Nomes:** curtos, significativos, sem “brancos”, se preciso usar abreviaturas
 - ❖ **Abreviatura:** usar mesmo princípio em todo BD
 - **Atributo identificador da entidade:** chave primária

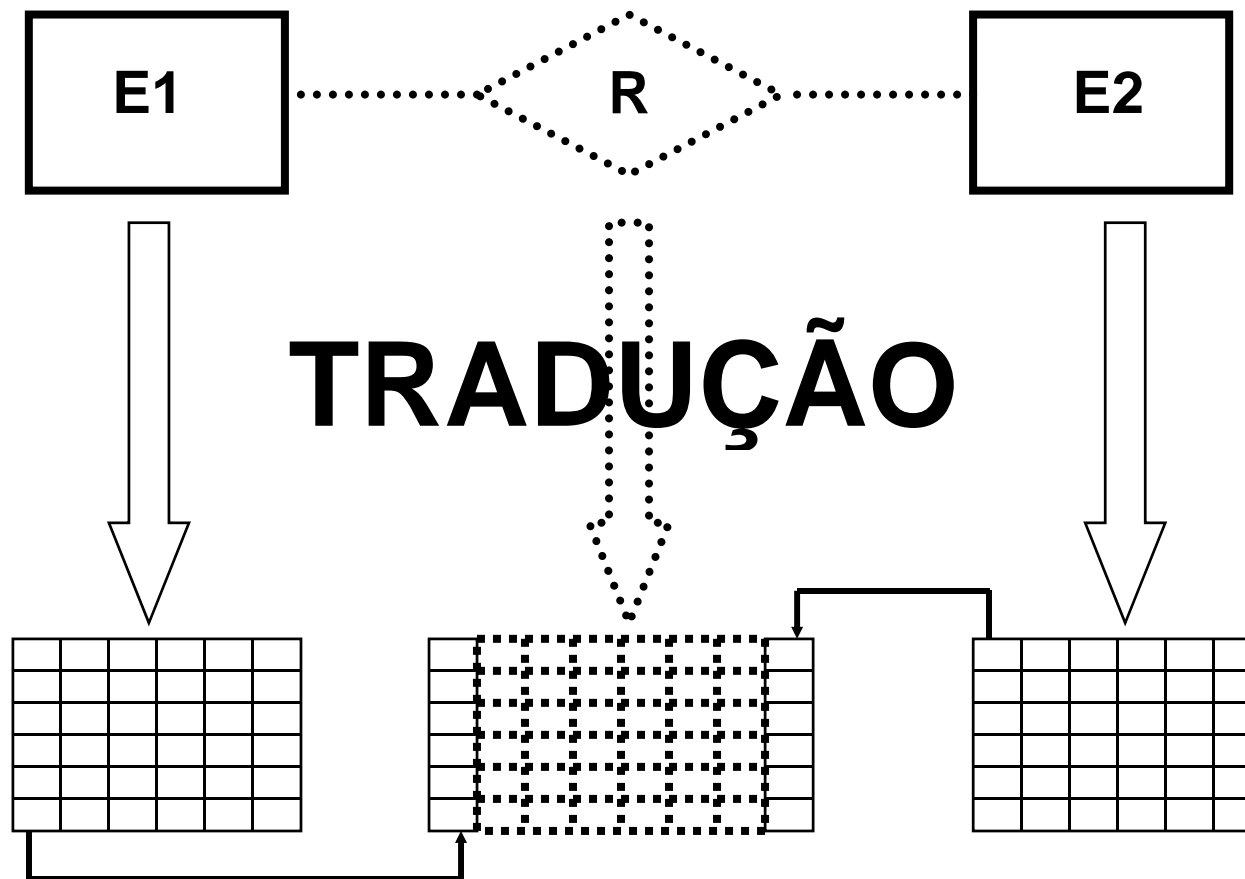


CONVENÇÕES PARA NOMEAR TABELAS E COLUNAS DE TABELAS DURANTE A TRADUÇÃO

- usar os nomes das tabelas no singular;
- nomes de colunas extensivamente utilizados na aplicação = devem ser o mais curtos possíveis;
- nomes de atributos compostos de diversas palavras devem ser abreviados;
- SGBDR não aceita brancos nos nomes de colunas, se preciso usar “underline”;
- não se inclui o nome da tabela no nome da coluna;
- a chave primária de uma tabela é uma exceção = poderá ser uma chave estrangeira de uma outra tabela;
- abreviaturas usadas em nomes de colunas devem ser utilizadas da mesma forma em todo o BD.

TRADUÇÃO DE RELACIONAMENTOS

- As cardinalidades são os fatores determinantes;
- Três formas básicas de implementação:
 - por TABELA PRÓPRIA PARA O RELACIONAMENTO;



TRADUÇÃO DE RELACIONAMENTO POR TABELA PRÓPRIA

Criar uma tabela para o relacionamento com as seguintes colunas:

- os identificadores das entidades que participam do relacionamento;
- as colunas dos atributos do relacionamento.
- A chave primária será o conjunto das colunas correspondentes aos identificadores das entidades que participam do relacionamento.



Engenheiro(CodigoEngenheiro, Nome_Eng)

Projeto(CodigoProjeto, Título_Proj)

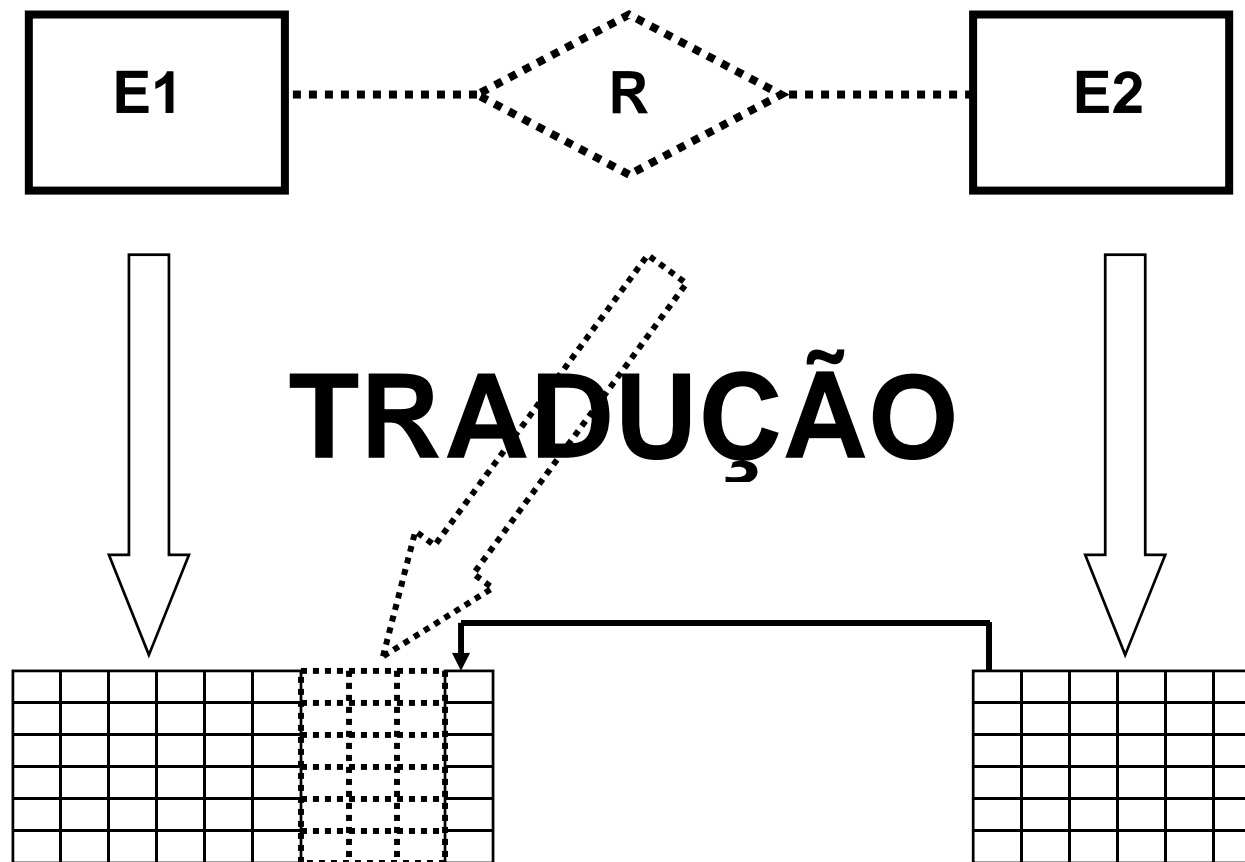
Atuação(CodigoEngenheiro, CodigoProjeto, Função_Atu, Dtalnic_Atu)

CodigoEngenheiro referencia Engenheiro

CodigoProjeto referencia Projeto

TRADUÇÃO DE RELACIONAMENTOS

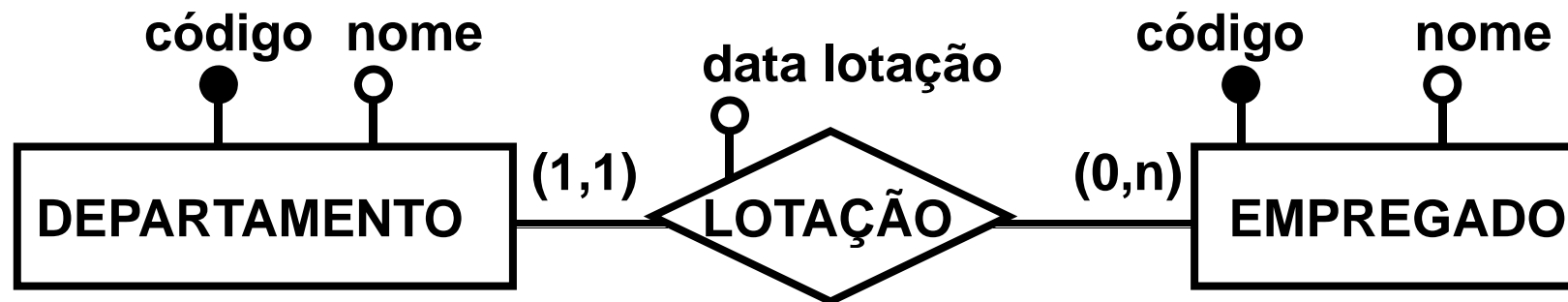
- por ADIÇÃO DE COLUNAS numa das tabelas que participam do relacionamento; e



TRADUÇÃO DE RELACIONAMENTO

POR ADIÇÃO DE COLUNAS NAS TABELAS QUE PARTICIPAM DO RELACIONAMENTO

- Só é possível quando há um relacionamento com cardinalidade máxima 1;
- Insere-se na tabela que tem o relacionamento com cardinalidade n;
- O(s) identificador(es) da outra tabela serão chave(s) estrangeira(s);
- O(s) atributo(s) próprios do relacionamento.



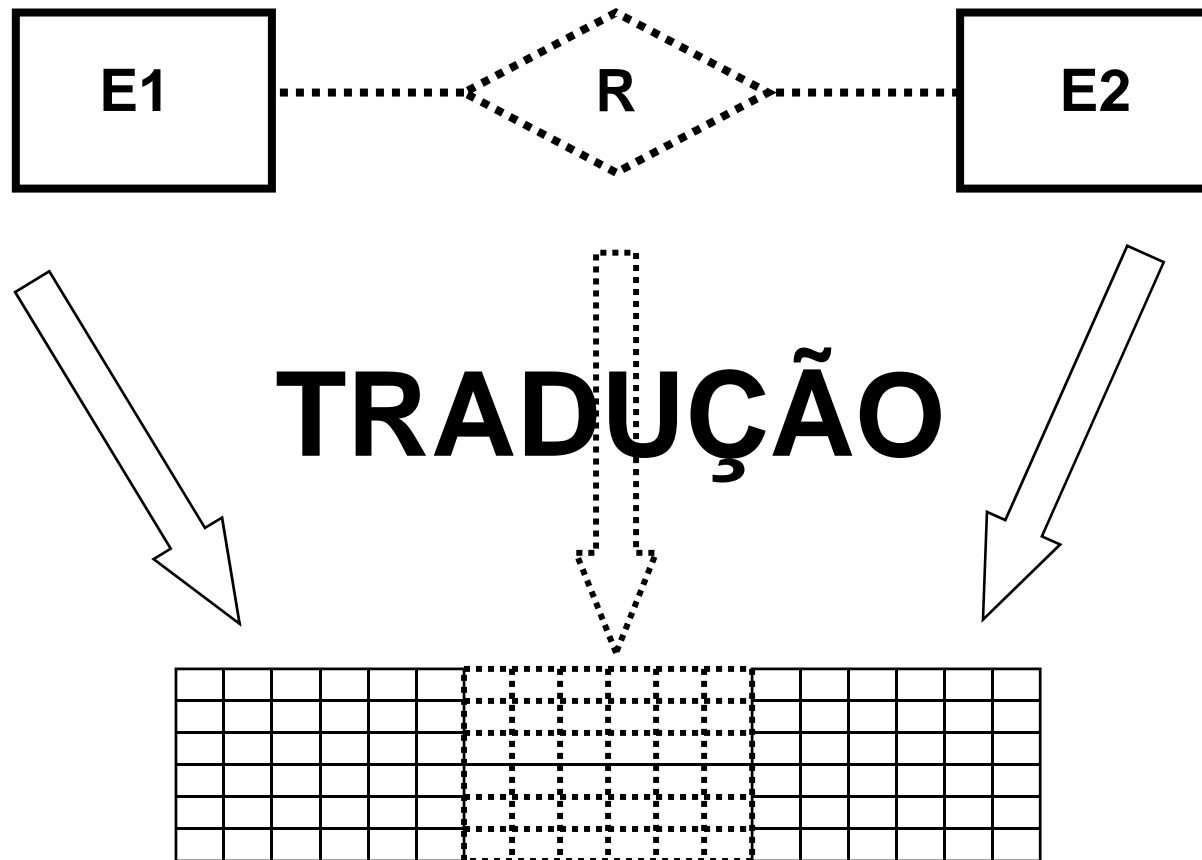
Departamento(CodigoDepartamento,Nome_Dep)

Empregado(CodigoEmpregado,Nome_Emp,CodigoDepartamento,Data_Lot)

CodigoDepartamento referencia Departamento

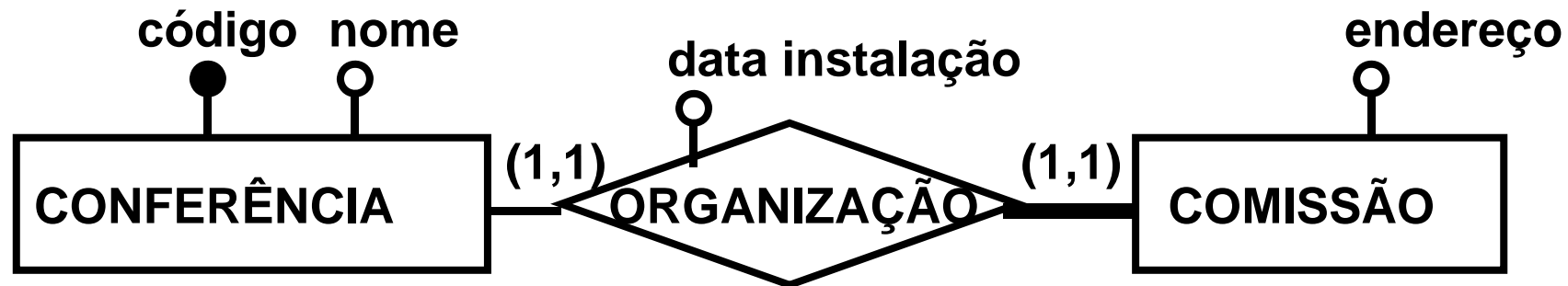
TRADUÇÃO DE RELACIONAMENTOS

- por FUSÃO DAS TABELAS das entidades que participam do relacionamento.



TRADUÇÃO DE RELACIONAMENTO POR FUSÃO DE TABELAS DE ENTIDADES QUE PARTICIPAM DO RELACIONAMENTO

- Só é possível quando o relacionamento é 1:1;
- Insere-se na tabela os atributos das entidades e do próprio relacionamento.

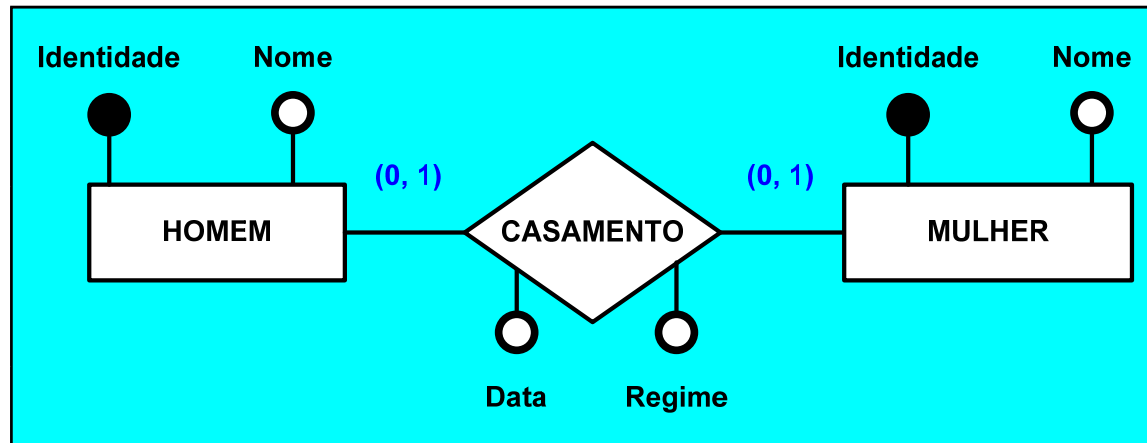


Conferência(CodigoConferência, Nome_Conf, DataInstalação_Org, Endereço_Com)

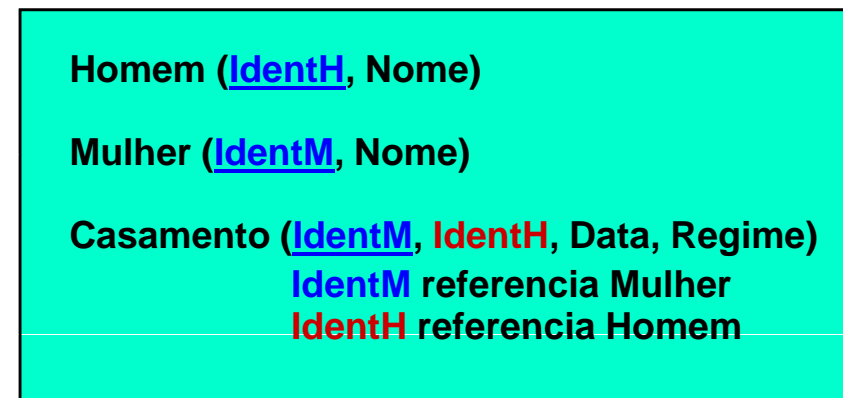
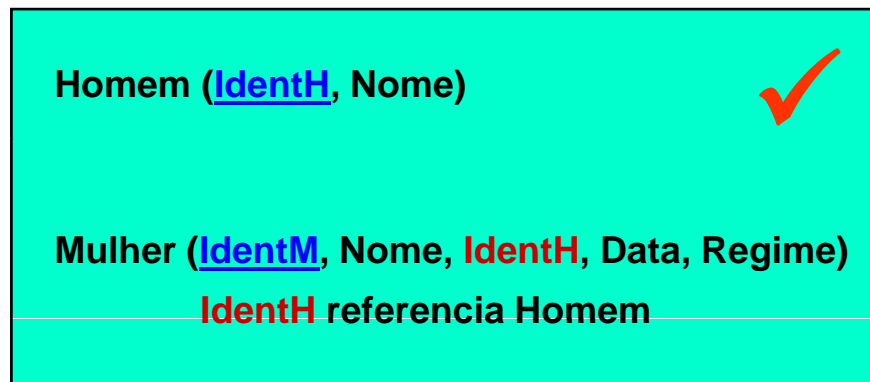
Transformações entre modelos

- Implementação de relacionamentos (1:1) :
 - ambas entidades com participação opcional

Modelo conceitual

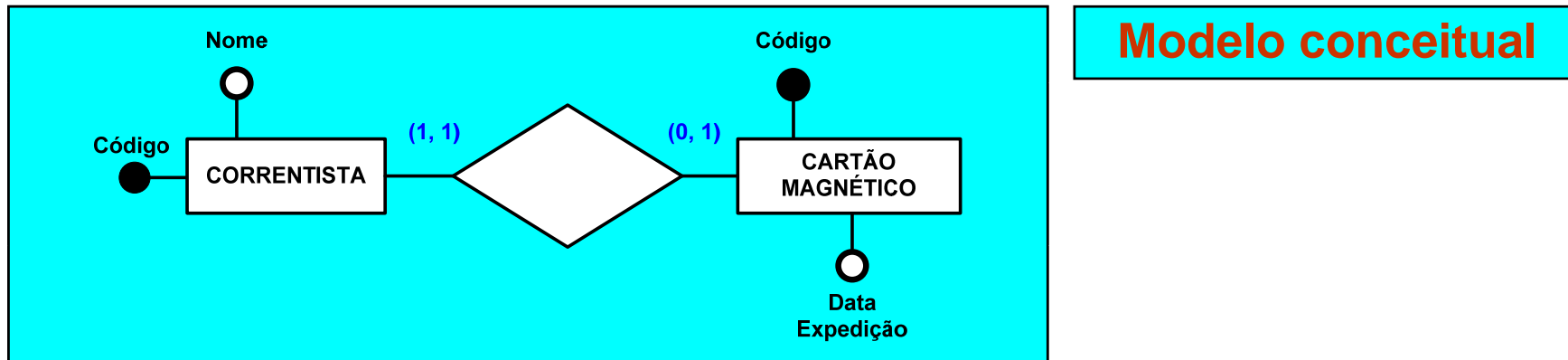


Modelo lógico



Transformações entre modelos

- Implementação de relacionamentos (1:1) :
 - uma entidade com participação opcional e outra obrigatória



Modelo conceitual

Modelo lógico

Correntista (CodCorrent, Nome, CodCartao, DataExp) ✓

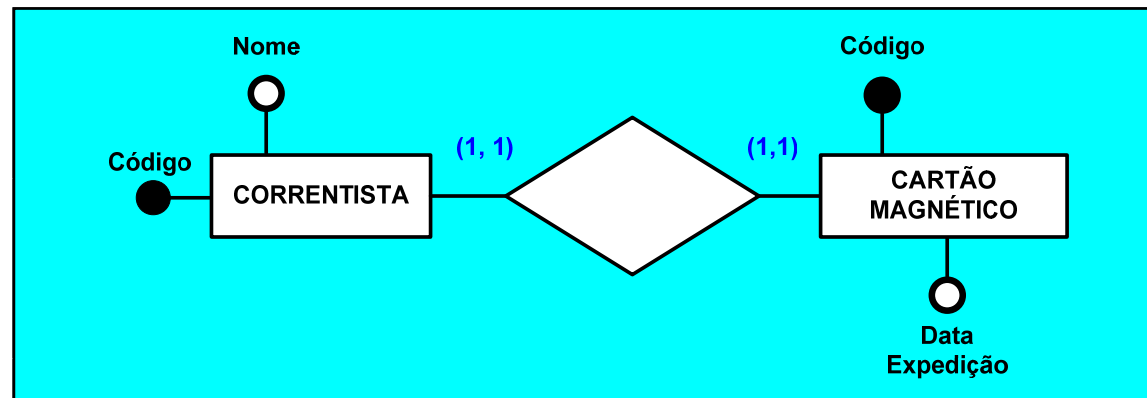
Correntista (CodCorrent, Nome)

Cartao (CodCartao, DataExp, **CodCorrent**)

CodCorrent referencia Correntista

Transformações entre modelos

- Implementação de relacionamentos (1:1) :
 - ambas entidades com participação obrigatória



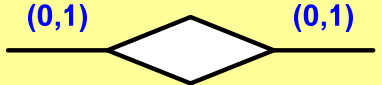
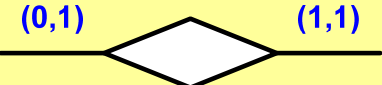
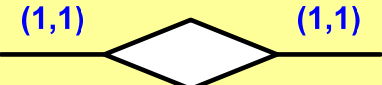
Modelo conceitual



Correntista (CodCorrent, Nome, CodCartao, DataExp)

Modelo lógico

Transformação de DER para Relacional

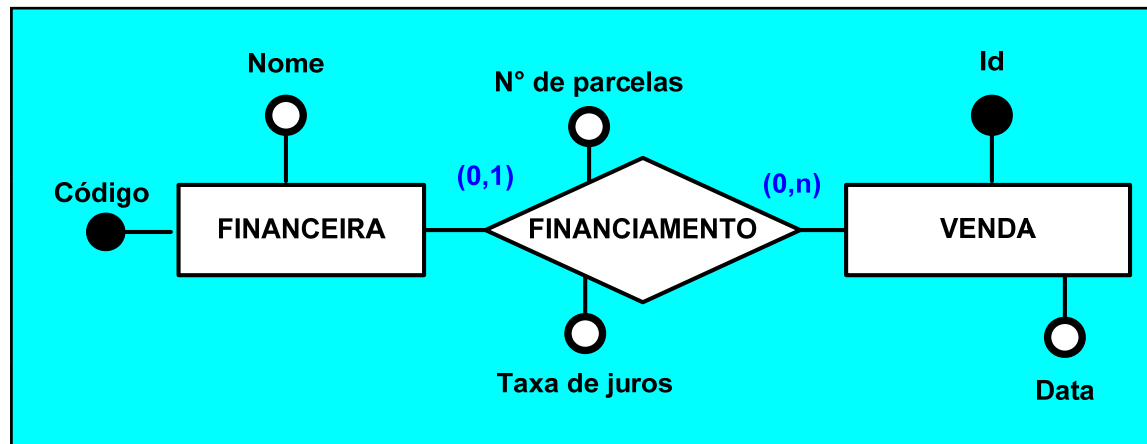
Tipo de relacionamento	Tabela própria	Adição de Coluna	Fusão de Tabelas
Relacionamentos 1:1			
(0,1)  (0,1)	±	✓	X
(0,1)  (1,1)	X	±	✓
(1,1)  (1,1)	X	X	✓

✓	Alternativa preferida
±	Pode ser usada
X	Não usar

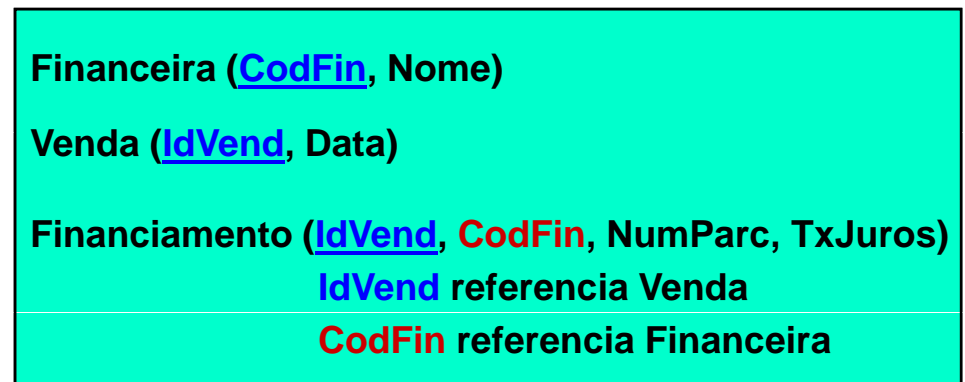
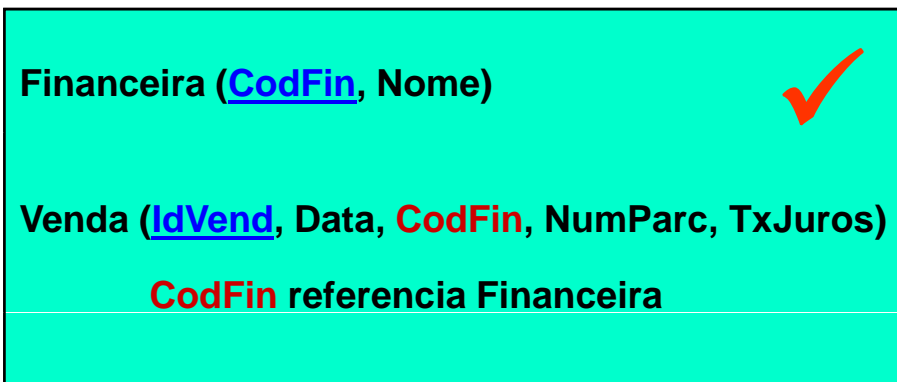
Transformações entre modelos

- Implementação de relacionamentos (1:N) :
→ alternativa preferida: adição de colunas

Modelo conceitual

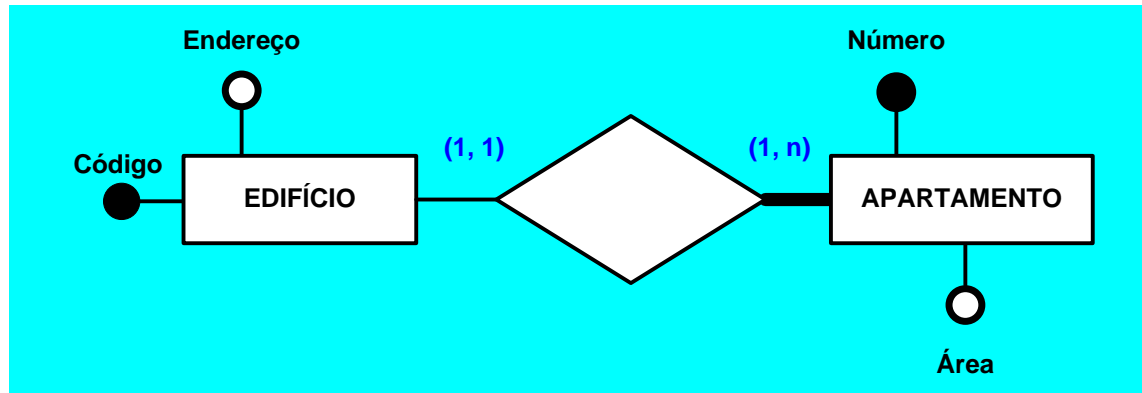


Modelo lógico



Transformações entre modelos

- Implementação de relacionamentos (1:N) :
→ alternativa preferida: adição de colunas



Modelo conceitual





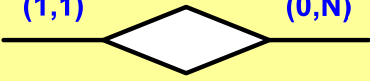

Edificio (CodigoEd, Endereco)

Apartamento (CodigoEd, NumAp, AreaAp)

CodigoEd referencia Edificio

Modelo lógico

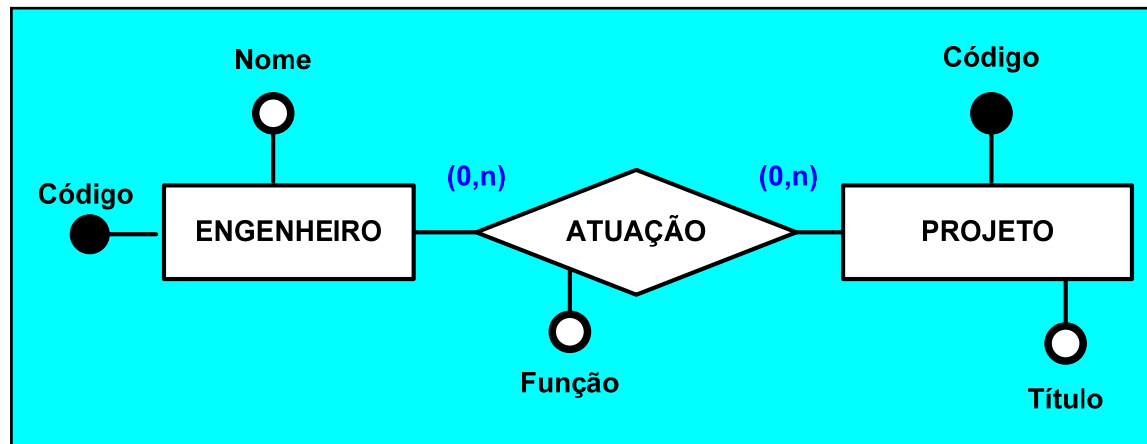
Transformação de DER para Relacional

Tipo de relacionamento	Tabela própria	Adição de Coluna	Fusão de Tabelas
Relacionamentos 1:N			
(0,1)  (0,N)	±	✓	X
(0,1)  (1,N)	±	✓	X
(1,1)  (0,N)	X	✓	X
(1,1)  (1,N)	X	✓	X

✓	Alternativa preferida
±	Pode ser usada
X	Não usar

Transformações entre modelos

- Implementação de relacionamentos (N:N) :
→ criação de uma nova tabela



Modelo conceitual

Engenheiro (CodEng, Nome)

Projeto (CodProj, Titulo)

Atuação (CodEng, CodProj, Funcao)

CodEng referencia Engenheiro

CodProj referencia Projeto

Modelo lógico

Transformação de DER para Relacional

Tipo de relacionamento	Tabela própria	Adição de Coluna	Fusão de Tabelas
Relacionamentos N:N			
	✓	X	X
	✓	X	X
	✓	X	X

✓	Alternativa preferida
±	Pode ser usada
X	Não usar

Transformações entre modelos

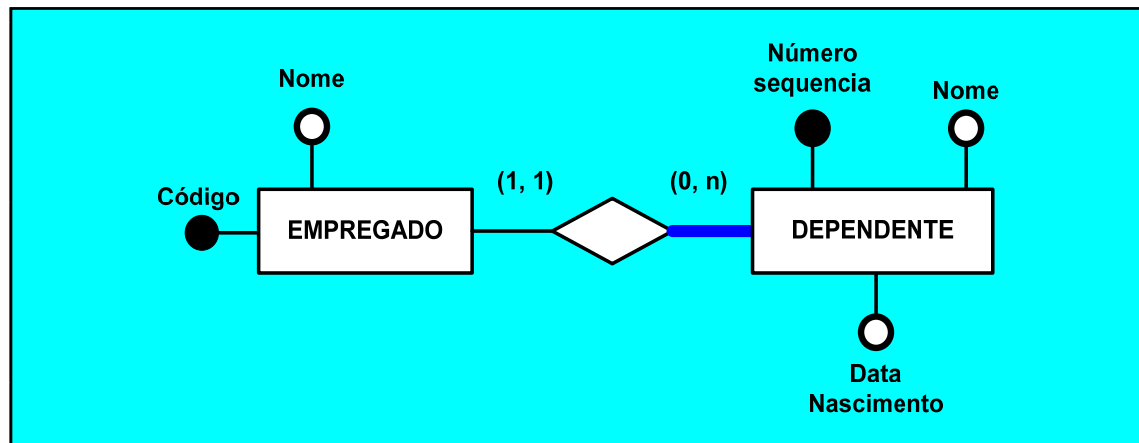
Tipo de relacionamento	Tabela própria	Adição de Coluna	Fusão de Tabelas
Relacionamentos 1:1			
(0,1) (0,1)	±	✓	X
(0,1) (1,1)	X	±	✓
(1,1) (1,1)	X	X	✓
Relacionamentos 1:N			
(0,1) (0,N)	±	✓	X
(0,1) (1,N)	±	✓	X
(1,1) (0,N)	X	✓	X
(1,1) (1,N)	X	✓	X
Relacionamentos N:N			
(0,N) (0,N)	✓	X	X
(0,N) (1,N)	✓	X	X
(1,N) (1,N)	✓	X	X

✓	Alternativa preferida
±	Pode ser usada
X	Não usar

Transformações entre modelos

■ Implementação de entidades com relacionamento identificador

Modelo conceitual



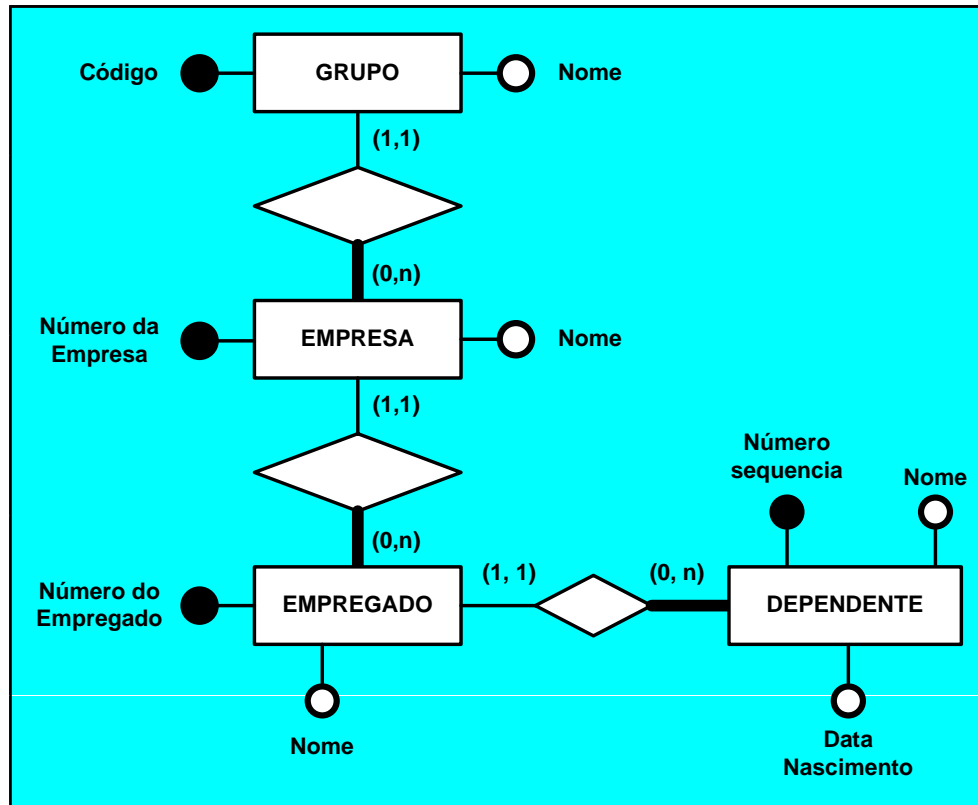
Modelo lógico

Empregado (CodigoEmp, Nome)

Dependente (CodigoEmp, NumSeq, Nome, DataNasc)

Transformações entre modelos

■ Implementação de entidades com relacionamento identificador



Modelo conceitual

Modelo lógico

Grupo (CodGrupo, Nome)

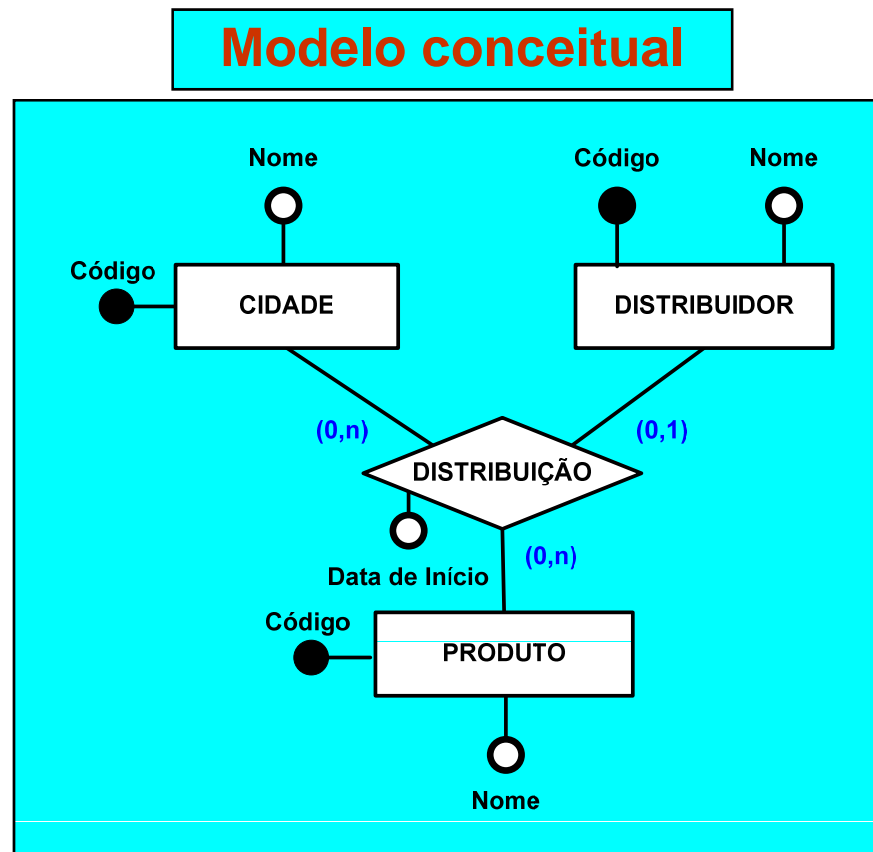
Empresa (CodGrupo, NumEmpresa, Nome)

Empregado (CodGrupo, NumEmpresa, NumEmpregado, Nome)

Dependente (CodGrupo, NumEmpresa, NumEmpregado, NumSeq, Nome, DataNasc)

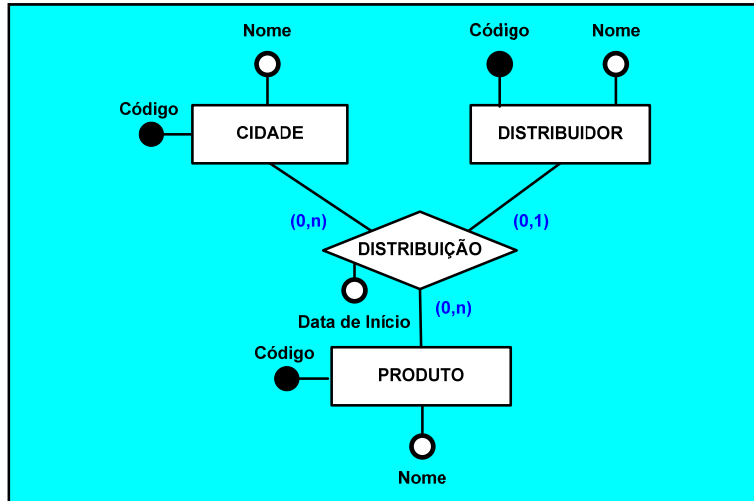
Transformações entre modelos

- **Relacionamentos de Grau maior que 02 (dois) - Ternário**
 - relacionamento é transformado em entidade
 - usa-se as regras de implementação de entidades e relacionamentos binários

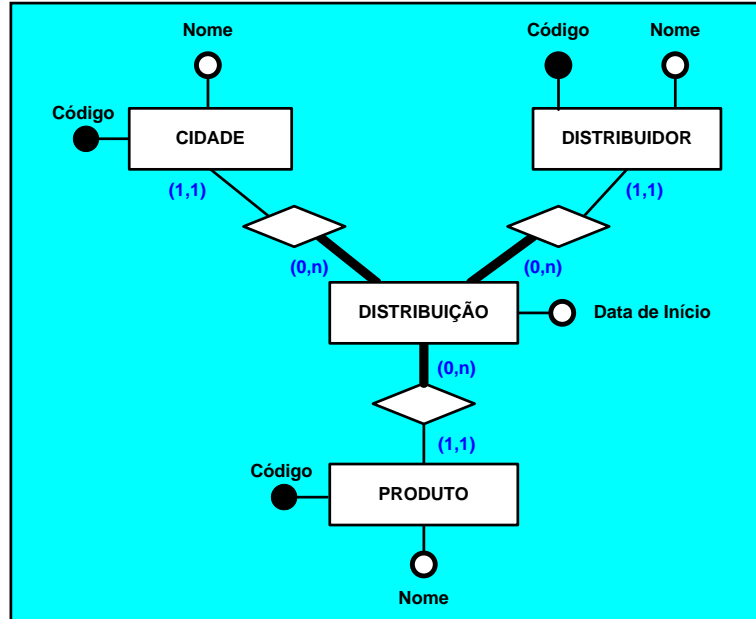


Transformações entre modelos

■ Relacionamentos de Grau maior que 02 (dois):



Modelo conceitual



Modelo lógico

Cidade (CodCid, Nome)

Distribuidor (CodDistr, Nome)

Produto (CodProd, Nome)

Distribuicao (CodCid, CodDistr, CodProd, DataIni)

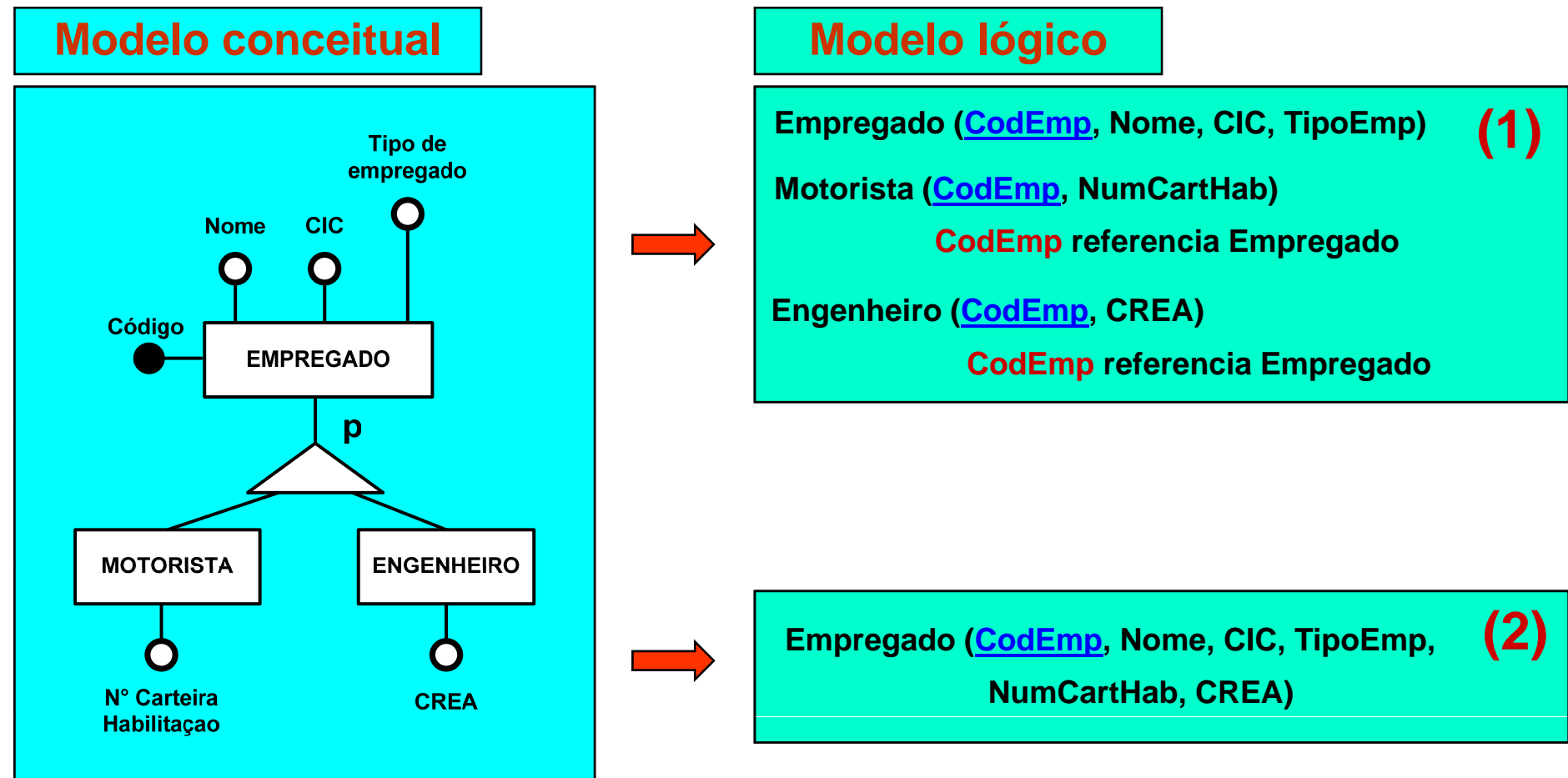
CodCid referencia Cidade

CodDistr referencia Distribuidor

CodProd referencia Produto

Transformações entre modelos

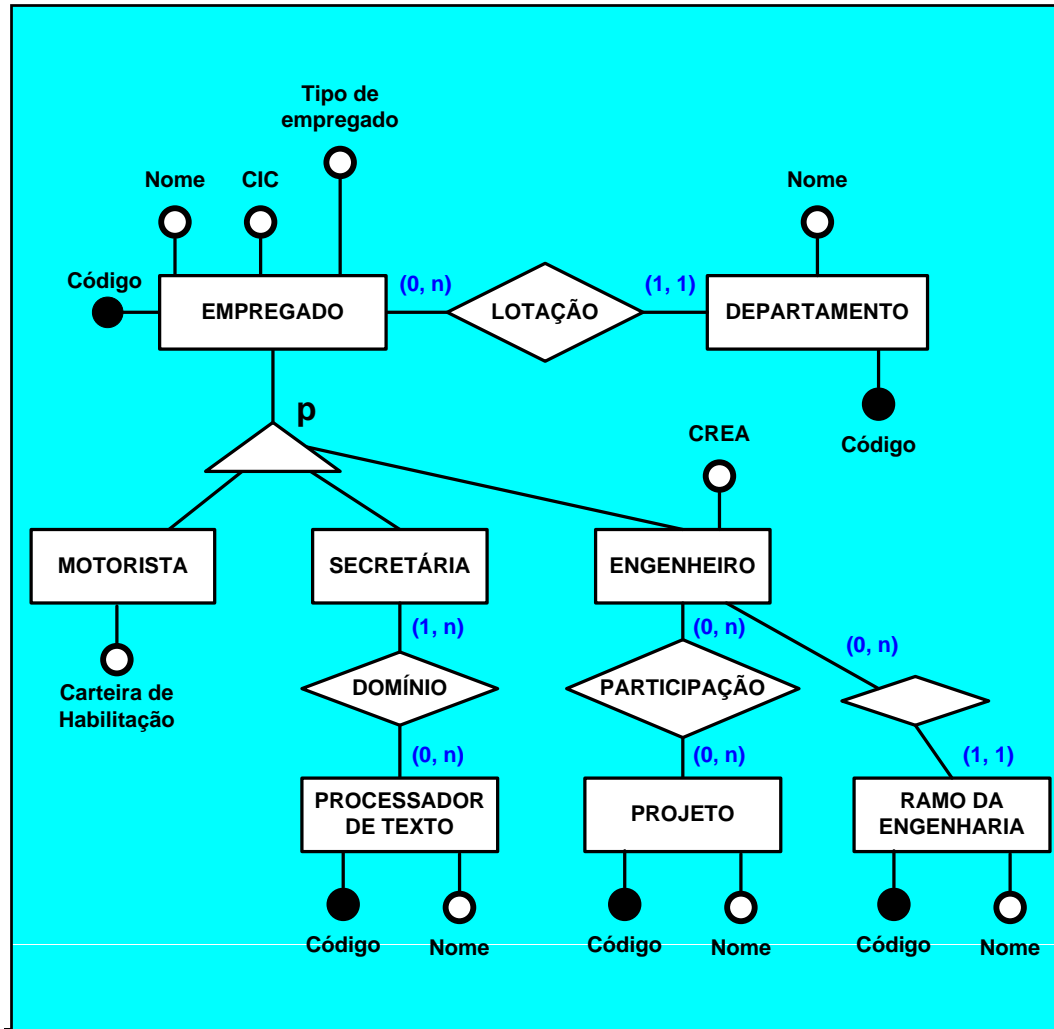
- Implementação generalização / especialização (alternativas):
 - (1) uso de uma tabela para cada entidade
 - (2) uso de uma única tabela para toda hierarquia



Transformações entre modelos

- Implementação generalização / especialização:
 - uma única tabela para toda hierarquia generalização/ especialização

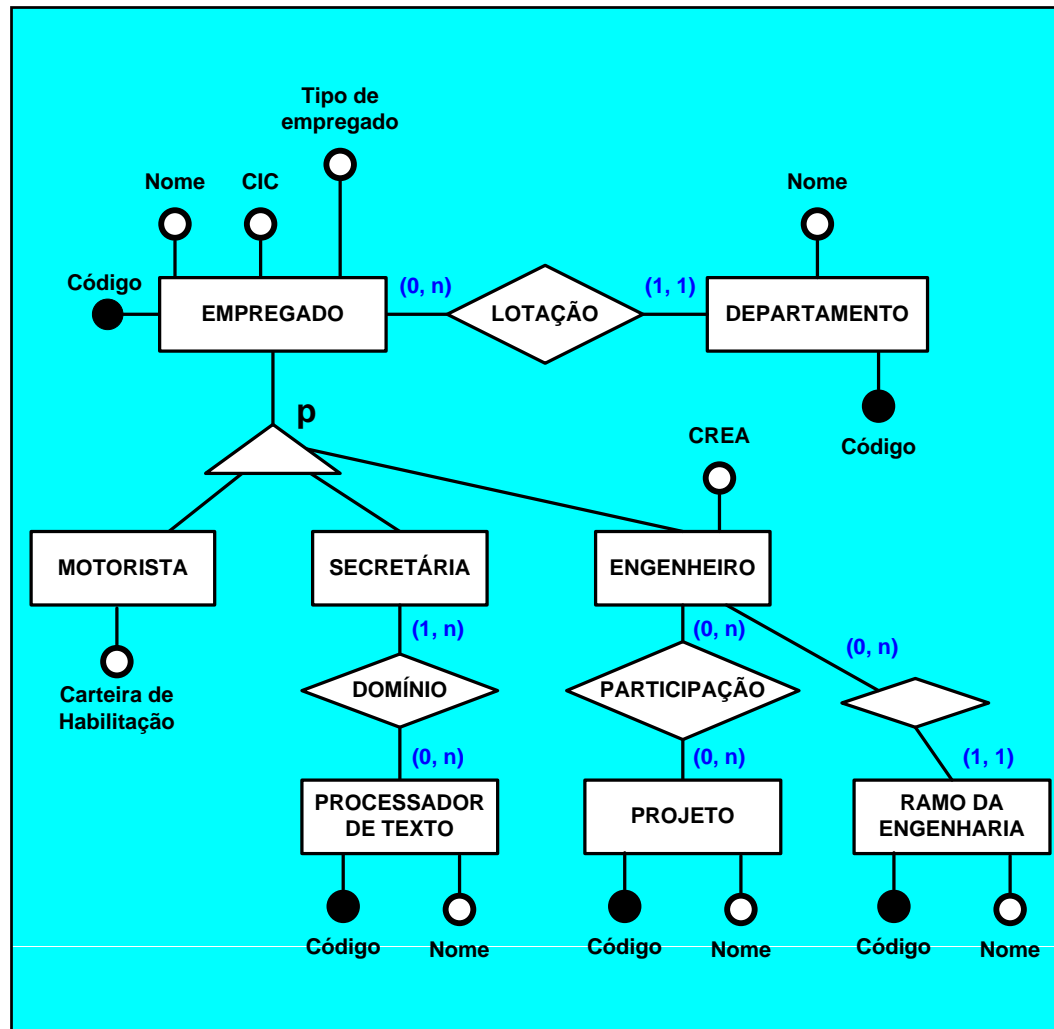
Modelo conceitual



Transformações entre modelos

- Implementação generalização / especialização:
 - uma única tabela para toda hierarquia generalização/ especialização

Modelo conceitual



Modelo lógico

Empregado (CodEmp, Nome, CIC, TipoEmp, NumCartHab, CREA, **CodDep**, **CodRamo**)
CodDep referencia Departamento
CodRamo referencia Ramo

Departamento (CodDep, Nome)

Ramo (CodRamo, Nome)

Projeto (CodProj, Nome)

ProcText (CodProc, Nome)

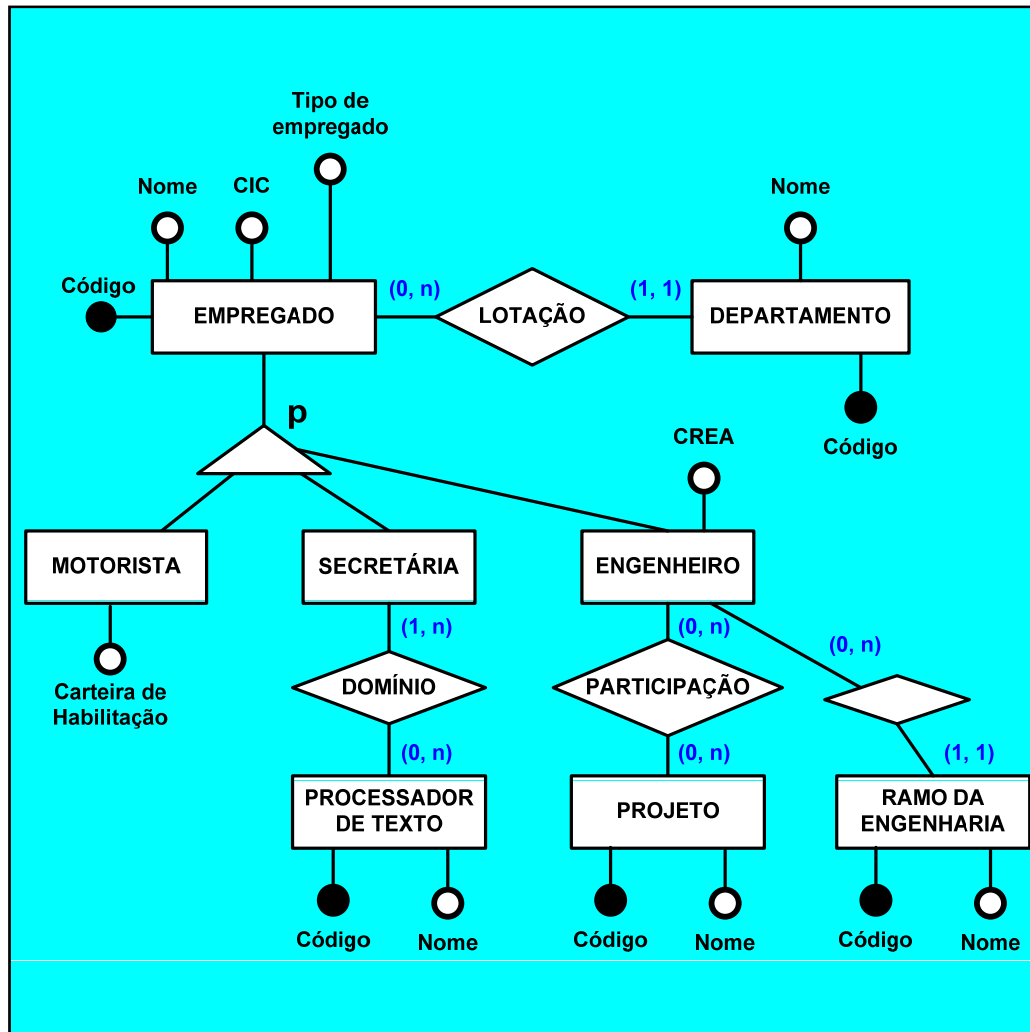
Dominio (CodEmp, CodProc)
CodEmp referencia Empregado
CodProc referencia ProcText

Participacao (CodEmp, CodProj)
CodEmp referencia Empregado
CodProj referencia Projeto

Transformações entre modelos

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

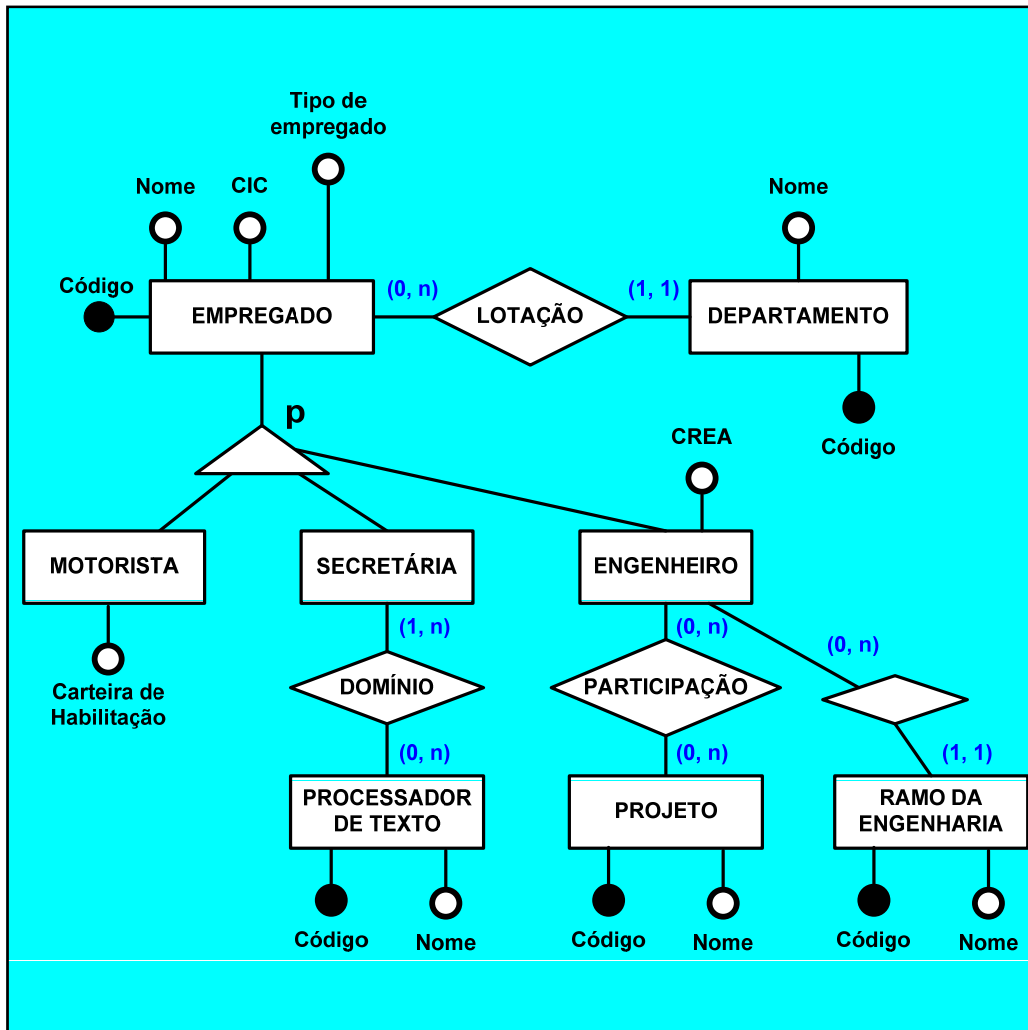
Modelo conceitual



Transformações entre modelos

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

Modelo conceitual



Modelo lógico

Empregado (CodEmp, Nome, CIC, Tipo, **CodDep**)

CodDep referencia Departamento

Departamento (CodDep, Nome)

Motorista (CodEmp, NumCartHab)

Engenheiro (CodEmp, CREA, **CodRamo**)

CodRamo referencia Ramo

Ramo (CodRamo, Nome)

Projeto (CodProj, Nome)

ProcText (CodProc, Nome)

Dominio (CodEmp, CodProc)

CodEmp referencia Empregado

CodProc referencia ProcText

Participacao (CodEmp, CodProj)

CodEmp referencia Empregado

CodProj referencia Projeto

TRADUÇÃO DE GENERALIZAÇÃO/ESPECIALIZAÇÃO COMPARAÇÃO DAS IMPLEMENTAÇÕES UMA TABELA PARA TODA A HIERARQUIA

VANTAGENS

- Dados referentes a uma ocorrência de entidade genérica bem como as ocorrências de suas especializações estão numa mesma linha;
- Chave primária da hierarquia é armazenada uma única vez.

UMA TABELA PARA CADA ENTIDADE DA HIERARQUIA

VANTAGENS

- Colunas opcionais que aparecem são apenas aquelas referentes a atributos que podem ser vazios do ponto de vista da aplicação;
- Controle das colunas opcionais é feito pela aplicação com base no valor da coluna tipo e não pelo SGBDR.

DECISÃO

- O projetista optará pela mais adequada à sua situação.

REFINAMENTOS NO MODELO RELACIONAL

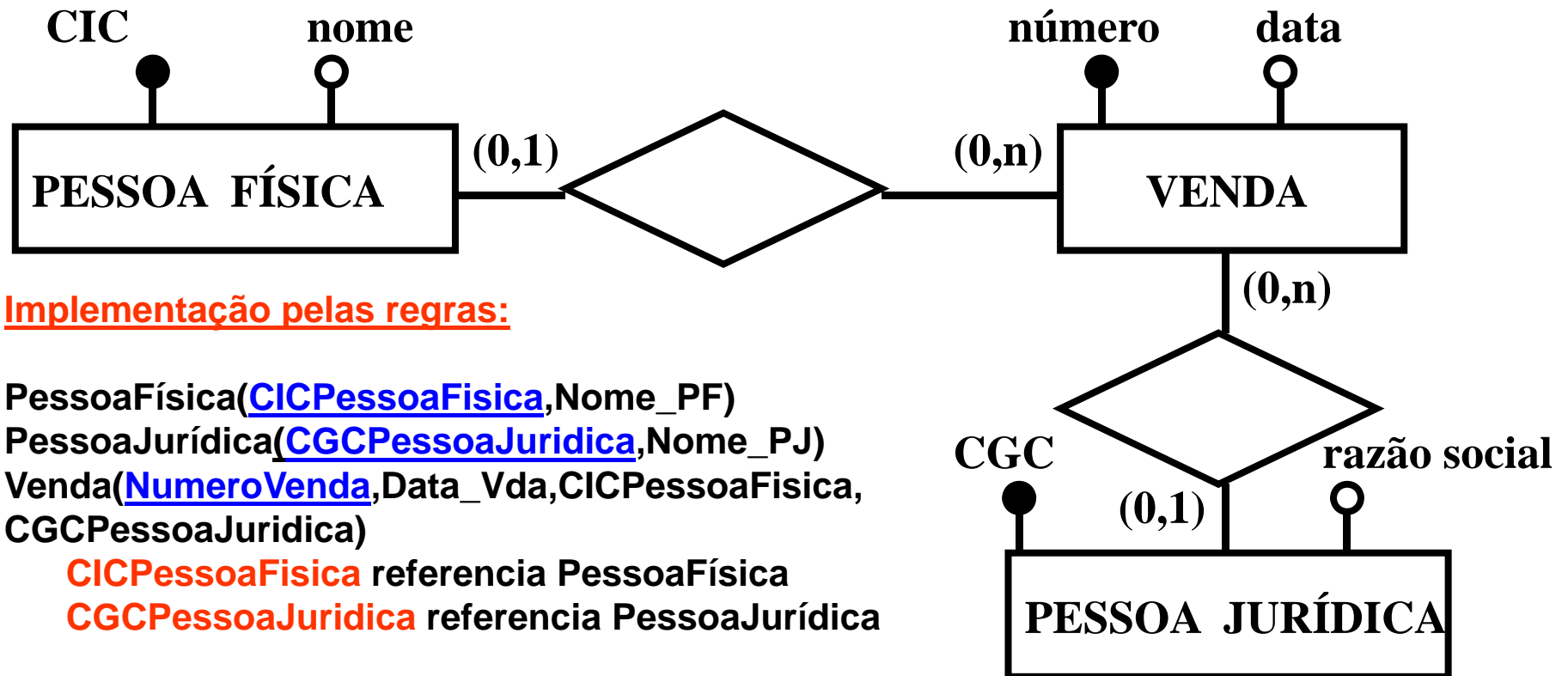
Em certas circunstâncias o projeto de BD desenvolvido pela observação cuidadosa das regras de tradução mostradas, pode não atender aos requisitos de performance impostos ao sistema.

Diante desta situação é preciso buscar alternativas que resultem em melhor performance, mesmo com a desobediência das regras de tradução.

Estas alternativas só devem ser adotadas em último caso, pois o seu uso indiscriminado levam a resultados piores do que o esperado.

Serão mostradas a seguir algumas destas alternativas.

RELACIONAMENTOS MUTUAMENTE EXCLUSIVOS



Implementação pelas regras:

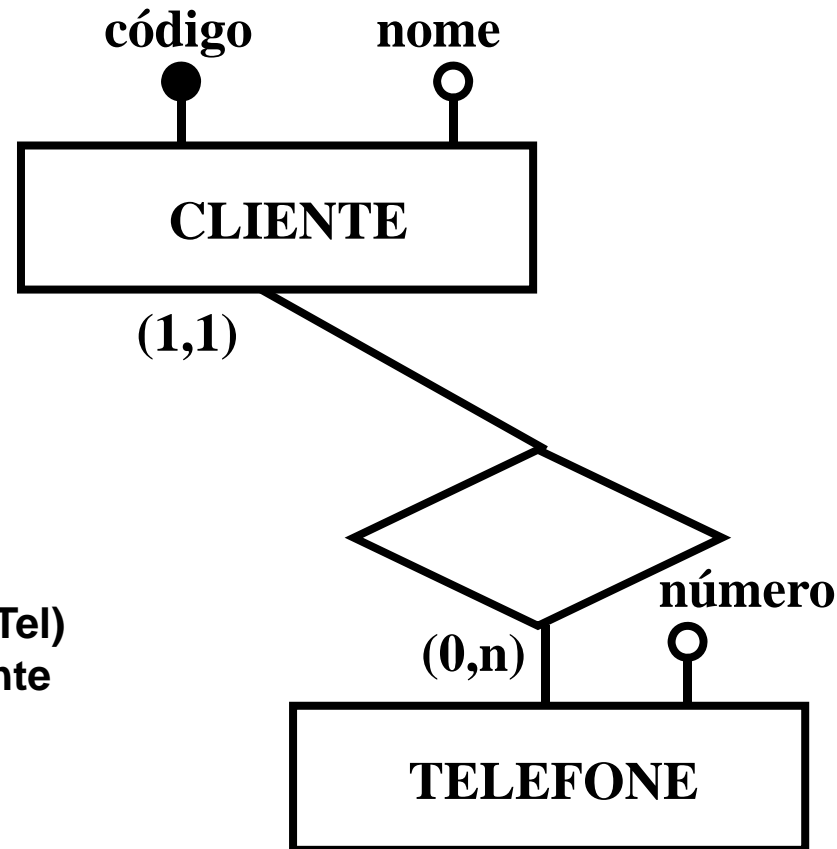
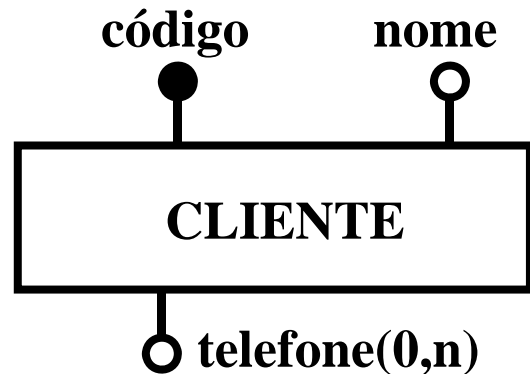
PessoaFísica(**CICPessoaFísica**, Nome_PF)
PessoaJurídica(**CGCPessoaJuridica**, Nome_PJ)
Venda(**NumeroVenda**, Data_Vda, CICPessoaFísica, CGCPessoaJuridica)
CICPessoaFísica referencia PessoaFísica
CGCPessoaJuridica referencia PessoaJurídica

Alternativa:

PessoaFísica(**CICPessoaFísica**, Nome_PF)
PessoaJurídica(**CGCPessoaJuridica**, Nome_PJ)
Venda(**NumeroVenda**, Data_Vda, CIC/CGC, Tipo)

- Evita-se colunas opcionais;
- Não permite especificar ao SGBDR que CIC/CGC é chave estrangeira.

SIMULAÇÃO DE ATRIBUTOS MULTI-VALORADOS



Implementação pelas regras:

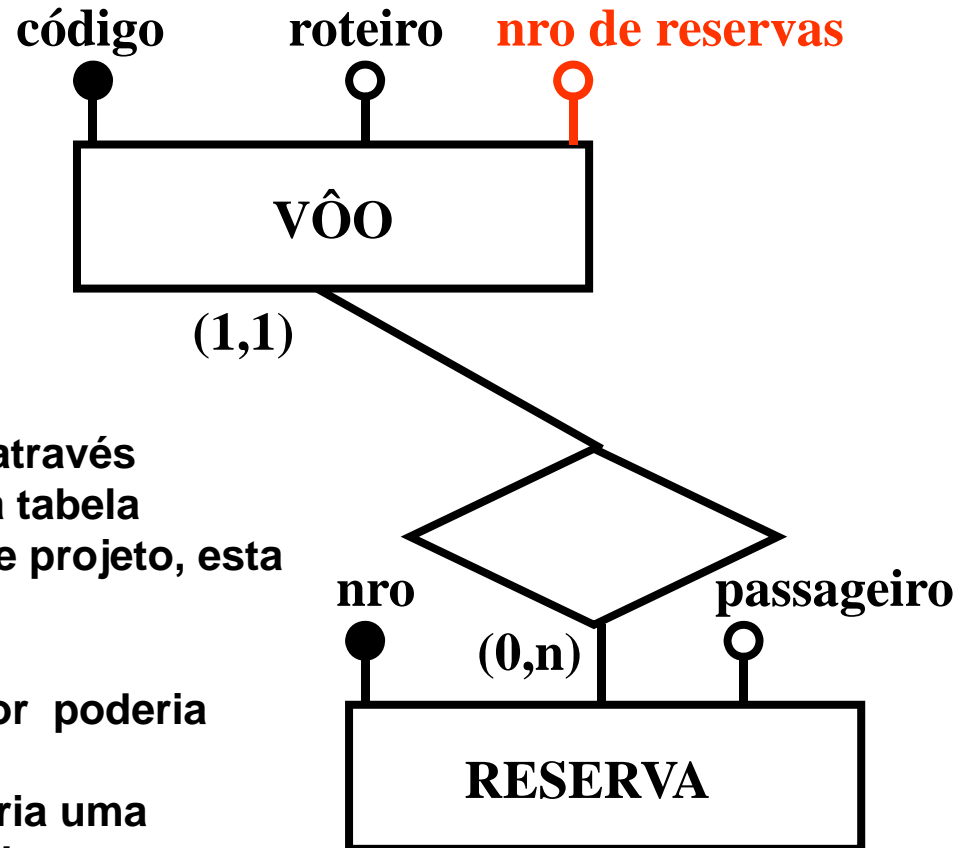
Cliente(CodigoCliente, Nome_Cli)
Telefone(CodigoCliente, Número_Tel)
CodigoCliente referencia Cliente

Alternativa:

Cliente(CodigoCliente, Nome_Cli,
Nro_Tel1, Nro_Tel2)

- Evita-se a junção;
- Pode haver coluna opcional.

USO DE INFORMAÇÕES REDUNDANTES



- O número de reservas é feito através de uma contagem das linhas da tabela reserva. Sob o ponto de vista de projeto, esta é uma informação redundante.

-Um atributo contendo este valor poderia contribuir com a performance, uma vez que não seria necessária uma contagem em toda tabela quando se necessitasse desta informação.