



Transformação de modelos (Conceitual – Relacional)

TIAGO G MORAES



❑ Modelo relacional (lógico)

- conceitos básicos
- chave
- Restrições de integridade
- Representação modelo

❑ Transformação entre modelos (conceitual – lógico)

- Implementação entidades
- Implementação relacionamentos
- Implementação generalização/especialização e entidade associativa

Modelo Relacional



❑ Revisando:

- Os dados são armazenados em tabelas (relações)
- Cada tabela é formada por:
 - Conjunto não ordenado de linhas (tuplas)
 - Colunas (atributos, campos)
 - Valores atômicos (campo sem subdivisões) e
 - Monovalorados (apenas um valor)

Nome	Idade	Sexo
João Carlos	25	Masculino
Júlio	20	Femino

Modelo Relacional - chave



- ❑ Chave: conceito para identificar linhas
- ❑ Chave primária (PK)
 - Coluna (ou combinação de colunas) que possui **valores** que não se repetem (**únicos**)
 - A combinação de linhas deve ser mínima (mínimo necessário para identificar unicamente uma linha na tabela)
 - **Uma** por tabela
- ❑ Chave candidata (ou alternativa):
 - Caso exista outra possibilidade de escolha de uma chave primária, essa é uma chave alternativa ou candidata
 - Pode ter **mais de uma** por tabela

Modelo Relacional - chave



- ❑ Chave: conceito para identificar linhas
- ❑ Chave estrangeira (FK)
 - Identifica unicamente uma linha de **outra** tabela
 - Faz referência a PK de outra tabela



Modelo Relacional - chave

- ❑ Chave: conceito para identificar linhas
- ❑ Chave estrangeira (FK)
 - Identifica unicamente uma linha de **outra** tabela
 - Faz referência a PK de outra tabela

Funcionário

CPF	Nome	Idade	Sexo	RG	CodDepo
009.287.323-09	João Carlos	25	Masculino	548759641	01
286.254.390-87	Júlio	20	Femino	207732202	05

Chave Primária (PK)

Chave Candidata

Chave Estrangeira (FK)

Modelo Relacional - chave



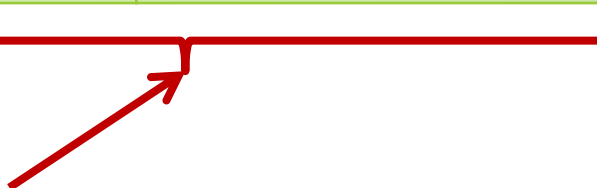
❑ Chave primária composta

- PK → (CodMedico, Data/Hora) ou (CodPaciente, Data/Hora)
 - Apenas os 2 campos juntos para identificar uma linha unicamente

ConsultaMedica

CodMedico	Data/Hora	CodPaciente	Duração
0011	12/05/2015 – 13:00:00	002	30 min
0011	18/05/2015 – 13:00:00	002	25 min
0011	18/05/2015 – 13:50:00	003	30 min
0007	12/05/2015 – 13:00:00	003	45 min

Chave Primária (PK)



→ Qualquer tabela que
referenceie ConsultaMedica,
terá uma FK de 2 colunas!!!

Modelo Relacional - Restrições



❑ Restrições de integridade

- Existem para manter a integridade dos dados de um BD
 - Dados que condizem com a realidade
 - Dados consistentes
- Mecanismo oferecido pelo próprio SGBD

❑ Tipos:

- Integridade de domínio
- Integridade de vazio
- Integridade de chave
- Integridade referencial

Modelo Relacional - Restrições



❑ Integridade de domínio

- Diz respeito aos possíveis valores associados a um campo
- Exemplo:
 - Coluna idade → inteiros positivos
 - Coluna sexo → caractere 'F' ou 'M'

❑ Integridade de vazio

- Especifica se os campos podem ou não ser vazios (ou nulos)
- Trata se o campo é opcional ou obrigatório



❑ Integridade de chave

- Diz que os valores chave para a tabela (chave primária e candidata) devem ser únicos
- Não poderá existir duas linhas com o mesmo valor para determinado campo

❑ Integridade referencial

- Define que o valor em FK deve existir correspondência em PK da tabela que se refere
- Quando um valor de PK referenciada por FK é alterado ou excluído o SGBD deve saber como lidar com a situação

Modelo Relacional - representação



□ Textual

- Representa tabelas, colunas e chaves primária e estrangeira
- Não representa domínio das colunas
- Notação

`NomeTab` (`colPK1`, ..., `colPKN`, `col1`, ..., `colN`, `colFK1`, ..., `colFKN`)
(`colFK1`, ..., `colFKN`) referencia `nomeTabReferenciada`

○ Exemplo

`Departamento` (`codDepto`, `nome`)

`Empregado` (`codEmp`, `codDepto`, `nome`, `RG`, `salario`)
(`codDepto`) referencia `Departamento`

Modelo Relacional - representação

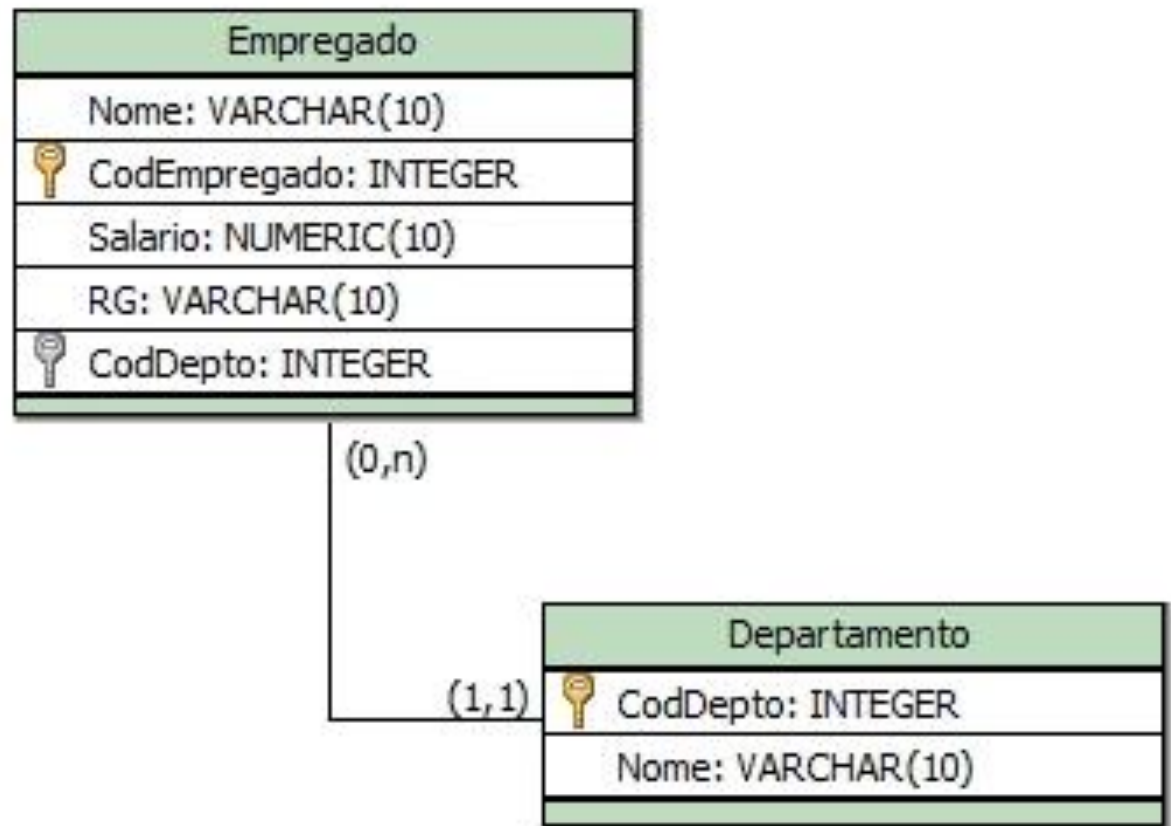


Diagramática

Possui mais informações:

- Cardinalidade
- Domínio campo

Exemplo:



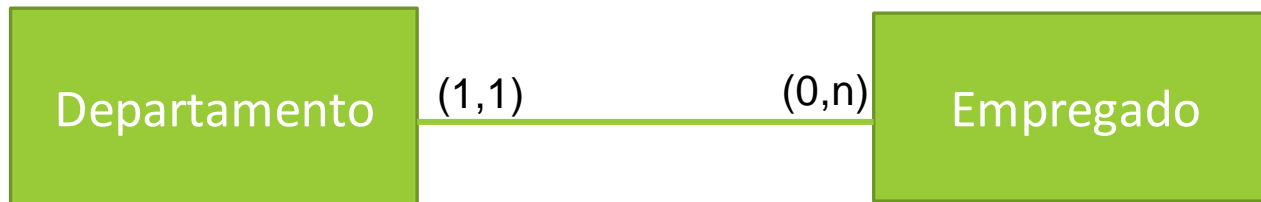
Modelo Relacional - representação



□ Textual (complementada)

`Departamento` (`codDepto`, `nome`)

`Empregado` (`codEmp`, `codDepto`, `nome`, `RG`, `salario`)
(`codDepto`) referencia `Departamento`



Transformação entre modelos



- ❑ Um modelo **conceitual** pode ser transformado para um modelo **lógico**, e **vice-versa**
 - **Lógico** → **Conceitual**: engenharia reversa
 - **Conceitual** → **Lógico**: projeto lógico
- ❑ Projeto lógico:
 - Primeiro se mapeia o ER em modelo relacional
 - Depois se refina o modelo a partir dos dados do domínio do problema

Transformação entre modelos



❑ Projeto lógico:

○ Objetivos:

1. Performance (desempenho → poucas operações de I/O)
2. Simplicidade (para uso, desenvolvimento e manutenção)

○ Princípios:

1. **Evitar junções:** quando uma consulta lê dados de mais de uma tabela
2. **Evitar chaves:** chaves geram complexidade no que se refere a estrutura de dados.
 - São criados índices para fácil acesso.
 - Deve-se utilizá-las na medida certa.
 - Exemplo: endereço como coluna (string) ou outra tabela com colunas?
3. **Evitar campos opcionais:** só se necessário o campo será opcional.
 - Não deve-se terceirizar tal responsabilidade (a um programa cliente)
 - Consultas com campos que possuem valor NULL podem ser complicadas

Transformação entre modelos



❑ Projeto lógico:

○ Passos:

1. Implementação **entidades** e respectivos atributos
2. Implementação de **relacionamentos** binários e seus atributos
3. Implementar demais relacionamentos, generalizações/especializações e entidades associativas

Transformação entre modelos



❑ Implementação **entidades**

- Cada entidade → uma tabela
 - Cada atributo monovalorado → uma coluna
 - Atributo identificador → chave primária
-
- Nomes devem ser: curtos, sem hífen e espaços
 - Caso não exista atributo identificador se cria uma PK:
 - Padrão:
 - “Id” ou “Cod”+”nome tabela” → CodPessoa, IdAluno, IdFoto ...

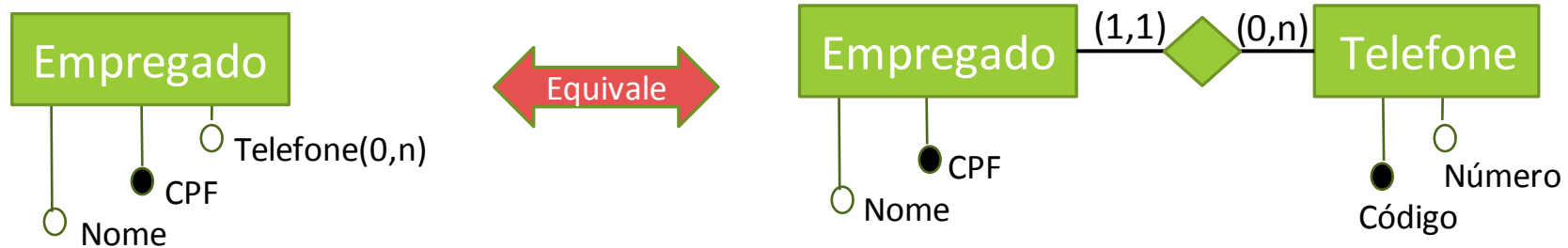
Transformação entre modelos



❑ Implementação **entidades**

○ Atributo multivalorado:

- Cria-se uma tabela para tal atributo (relacionamento 1:n) com a entidade que possuía o atributo
- Cria-se uma coluna a mais (FK) referente ao relacionamento identificador
- Exemplo:



Empregado (CPF, Nome)

Telefone (CodTelefone, numero, CodEmpregado)

CodEmpregado referencia Empregado

Transformação entre modelos



❑ Implementação **relacionamentos** binários

- Fator principal: **cardinalidades** min e max
- Opções:
 - Relacionamento mapeado para **nova tabela**
 - Relacionamento mapeado por **coluna (FK)**
 - Relacionamento mapeado para **fusão** de tabelas

Transformação entre modelos



❑ Implementação **relacionamentos** binários

○ Fator principal: **cardinalidades** min e max

○ Opções:

- Relacionamento mapeado para tabela
 - N:N
- Relacionamento mapeado por coluna (FK)
 - 1:N
- Relacionamento mapeado para fusão de tabelas
 - 1:1

Transformação entre modelos



❑ Implementação **relacionamentos** binários

○ Fator principal: **cardinalidades** min e max

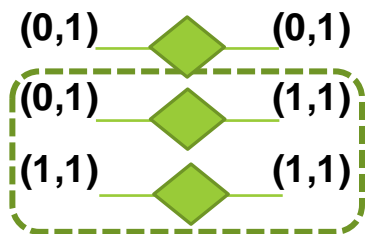
○ Opções:

- Relacionamento mapeado para tabela
 - N:N
- Relacionamento mapeado por coluna (FK)
 - 1:N
- Relacionamento mapeado para fusão de tabelas
 - 1:1

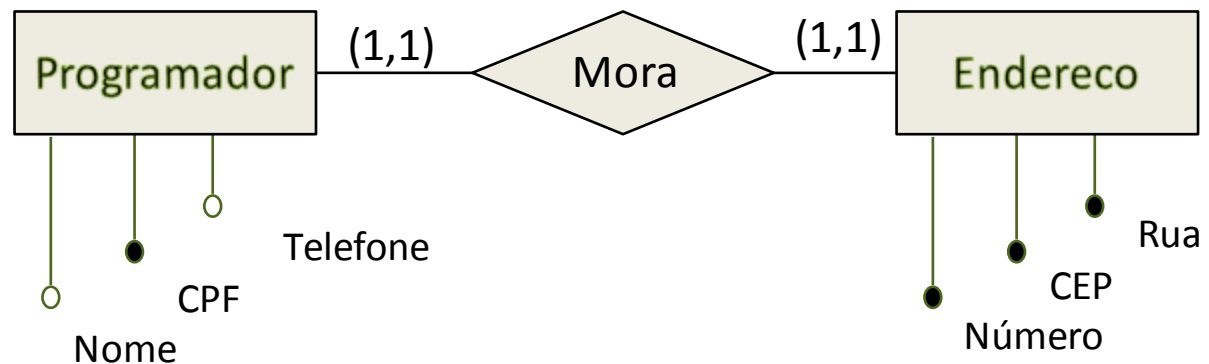
Transformação entre modelos



❑ Implementação **relacionamentos** binários 1:1



Fusão tabelas

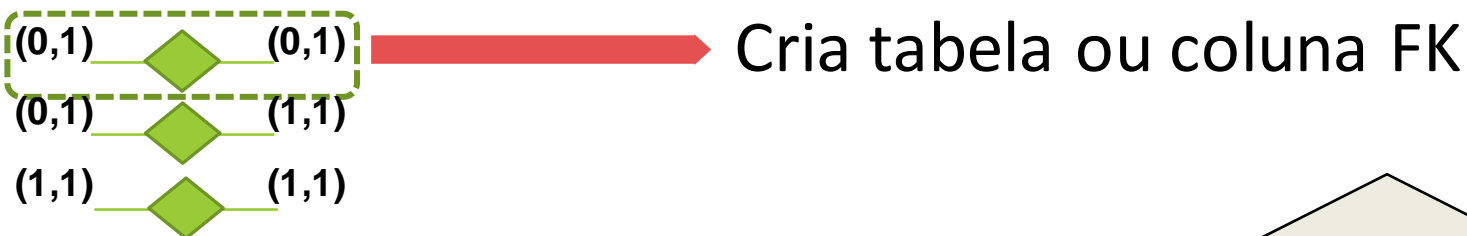


Programador(CPF, nome, telefone, EndNum, EndCEP, EndRua)

Transformação entre modelos



❑ Implementação **relacionamentos** binários 1:1

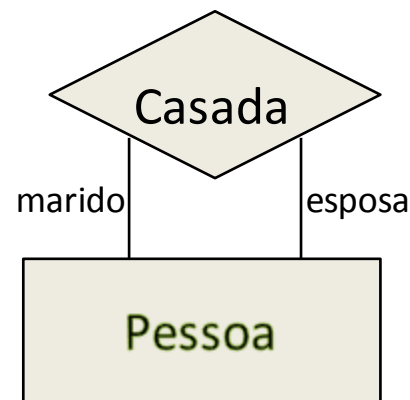


Pessoa (CPF, nome, Sexo)

Casamento (CPFMarido, CPFEsposa)

CPFMarido referencia Pessoa

CPFEsposa referencia Pessoa



OU

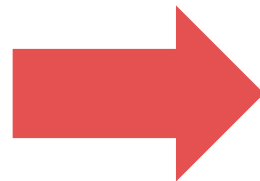
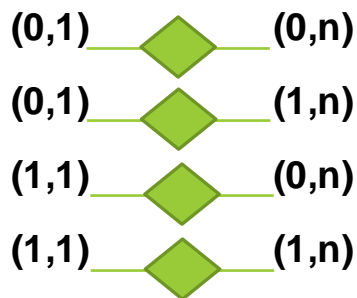
Pessoa (CPF, nome, Sexo, CPFConjuge)

CPFConjuge referencia Pessoa

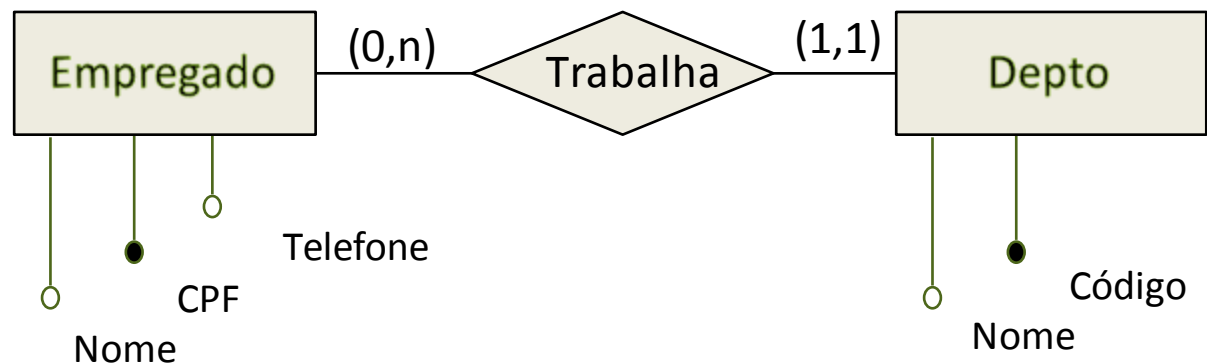
Transformação entre modelos



□ Implementação **relacionamentos** binários 1:N



FK no lado N

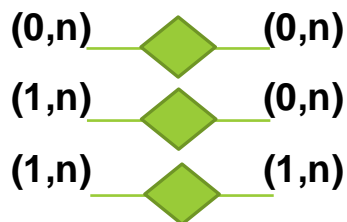


Empregado(CPF, nome, telefone, **codDepto**)
CodDepto referencia Depto
Depto(codDepto, nome)

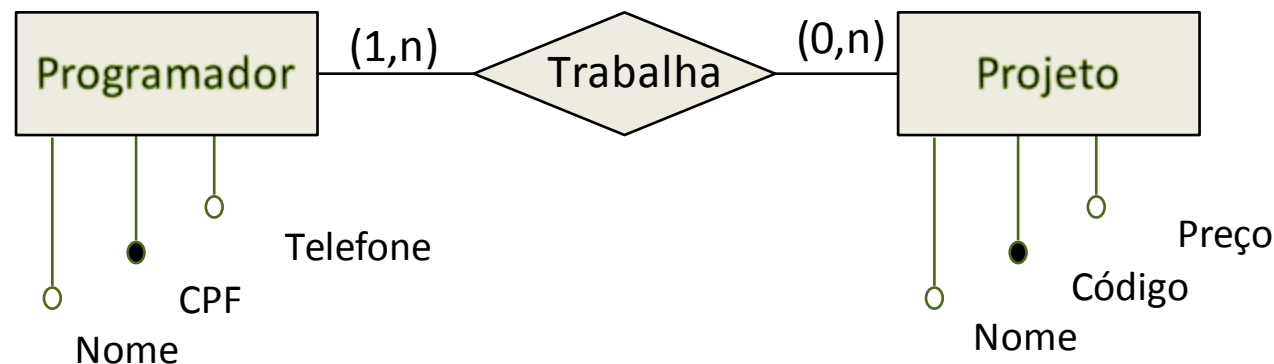
Transformação entre modelos



□ Implementação **relacionamentos** binários N:N



➔ Nova tabela com FK's



Programador(CPF, nome, telefone)

Projeto(codProjeto, nome, preco)

ProgramadorProjeto (CPF, CodProjeto)











CodProjeto referencia Projeto

CPF referencia Programador



Transformação entre modelos

Resumo Relacionamento binários

Relacionamento	Nova tabela	Nova Coluna (FK)	Fusão tabelas
<u>(0,1)</u>  <u>(0,1)</u>	+ -	OK	×
<u>(0,1)</u>  <u>(1,1)</u>	+ - -	+ -	OK
<u>(1,1)</u>  <u>(1,1)</u>	+ - -	+ - -	OK
<u>(0,1)</u>  <u>(0,n)</u>	+ -	OK	×
<u>(0,1)</u>  <u>(1,n)</u>	+ -	OK	×
<u>(1,1)</u>  <u>(0,n)</u>	+ - -	OK	×
<u>(1,1)</u>  <u>(1,n)</u>	+ - -	OK	×
<u>(0,n)</u>  <u>(0,n)</u>	OK	×	×
<u>(0,n)</u>  <u>(1,n)</u>	OK	×	×
<u>(0,n)</u>  <u>(1,n)</u>	OK	×	×

Legenda:
OK: melhor opção
+ - : outra opção menos usada
+ - - : opção raramente usada
× : não utilizar



❑ Implementação **relacionamentos não binários**

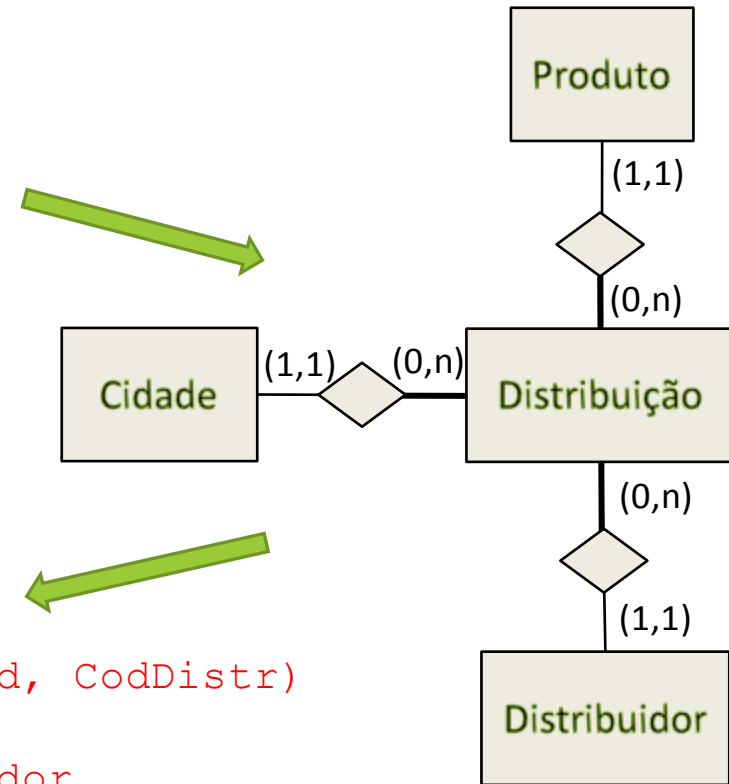
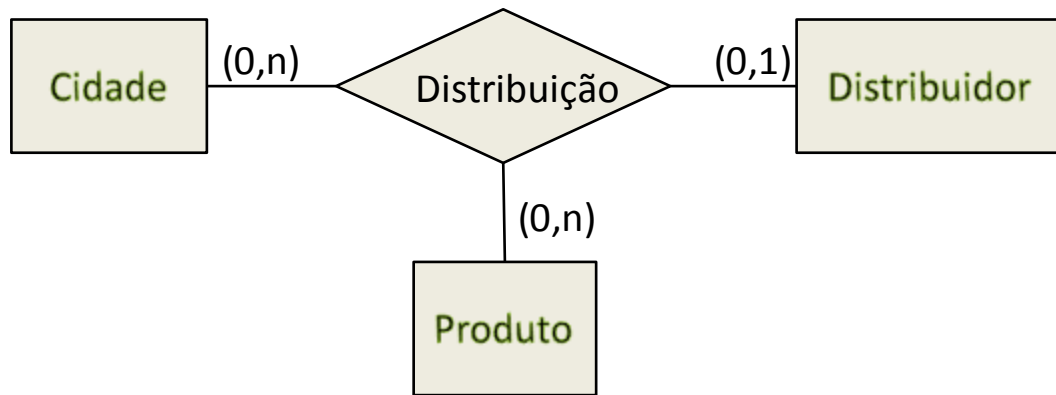
- Um relacionamento não binário pode ser entendido como um conjunto de relacionamentos binários
- Um relacionamento binário tem implementação já conhecida



Transformação entre modelos

❑ Implementação **relacionamentos não binários**

○ Exemplo: relacionamento ternário



Produto(CodProduto, Nome)
Cidade(Codcidade, Nome)
Distribuidor(CodDistribuidor, Nome)
Distribuicao(CodDistribuicao, CodProd, CodCid, CodDistr)
CodProd referencia Produto
CodDistr referencia Distribuidor
CodCid referencia Cidade



❑ Implementação **Generalização/Especialização**

- Opção 1: Implementar uma tabela apenas para a entidade genérica
- Opção 2: Implementar uma tabela para cada entidade (genérica e especialista)
- Opção 3: Implementar uma tabela pra cada entidade especialista

Transformação entre modelos



❑ Implementação **Generalização/Especialização**

- Opção 1: Implementar uma tabela para a entidade genérica
 - Colunas das tabelas especializadas (referentes aos atributos e relacionamentos das entidades especialistas)
 - Virão colunas opcionais da tabela genérica
- Tabela referente a entidade genérica terá coluna “tipo” para diferenciar

Transformação entre modelos



❑ Implementação **Generalização/Especialização**

- Opção 2: Implementar uma tabela para cada entidade (genérica e especialista)
 - Tabelas das entidades especialistas terão chave para tabela referente a entidade genérica
 - Tabela referente a entidade genérica poderá ter coluna “tipo” para diferenciar

Transformação entre modelos



❑ Implementação **Generalização/Especialização**

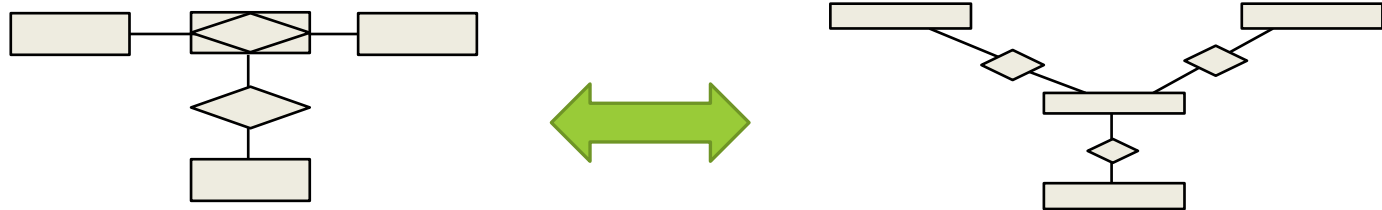
- Opção 3: Implementar uma tabela para a cada entidade especialista
 - Todos atributos herdados aparecem na tabela especialista
 - Chaves primárias espalhadas em mais de uma tabela
 - Como garantir unicidade de chaves primárias?

Transformação entre modelos



❑ Implementação **Entidade Associativa**

- Uma entidade associativa é um relacionamento que também se comporta como entidade
- Esse tipo de relacionamento possui uma implementação alternativa com um entidade e dois relacionamentos binários



- Um relacionamento binário tem implementação já conhecida