Modelo Lógico de Dados

Modelo Relacional

Composição de um Banco de Dados Relacional

- É composto de **tabelas** ou **relações**
- O termo **tabela** é mais comum nos produtos comerciais e na prática
- O termo **relação** foi utilizada na literatura original sobre a abordagem relacional (daí a denominação “relacional”) e é mais comum na área acadêmica

Tabelas

- **Tabela** é um conjunto não ordenado de linhas (tuplas, na terminologia acadêmica)
- Cada **linha** é composta por uma série de campos (valor de atributo na terminologia acadêmica)
- Cada **campo** é identificado por um nome de campo (nome de atributo, na terminologia acadêmica)

Tabelas

- O conjunto de campos homônimos de todas as linhas de uma tabela formam uma **coluna**

Emp

CódigoEmp	Nome	CódigoDepto	Categ Funcional
E5	Souza	D1	C5
E3	Santos	D2	C5
E2	Silva	D1	C2
E1	Soares	D1	-

Tabelas

- Observações:
  - Cada linha representa uma tupla
  - As linhas de uma tabela **não têm ordenação** - a ordem de recuperação é arbitrária, a menos que uma ordenação seja especificada na instrução de consulta
  - Não existem linhas duplicadas
  - Não é possível referenciar linhas de uma tabela por posição
  - Os valores de campo de uma tabela são **atômicos e monovalorados**
  - As consultas podem ser feitas por qualquer critério envolvendo os campos de uma ou mais linhas

Chaves Candidatas

- Coluna ou conjunto de colunas de uma Tabela que identifica de forma única cada linha da tabela e é mínimo (no caso de conjunto) nessa condição
  - O fato de todas as linhas de uma tabela serem distintas entre si garante a existência de ao menos uma chave candidata na tabela

Aluno (RA, Nome, CPF, Data_Nascimento, RG, Emissor_RG)

Chaves candidatas:

RA

CFP

(RG, Emissor_RG) (mínimo, precisa do par de valores)

Chave Candidata

- Observações:
  - Exige-se que seja **mínima** (quando todas as suas colunas forem efetivamente necessárias para garantir o requisito de unicidade de valores da chave)
  - Uma tabela sempre tem ao menos uma chave candidata e pode ter mais do que uma
  - Não existe uma classificação para as chaves candidatas. Todas a mesma relevância na tabela

Chave Primária

- Uma das chaves candidatas da Tabela deve ser escolhida com sua **Chave Primária**
- Não existe regra para decidir qual das chaves candidatas deve ser a escolhida
- Na escolha deve ser considerada a existência de referências a esta chave primária em outras tabelas (chave estrangeira)
  - Ao definir uma chave primária está se definindo uma restrição de integridade e não um caminho de acesso (índice)

Chave Estrangeira

- É uma **coluna** ou uma **combinação de colunas**, cujos valores aparecem necessariamente na chave primária de uma tabela
- Usada para implementar **relacionamento** em um banco de dados relacional

Chave Estrangeira

- Exemplo:

Depto

CódigoDepto	NomeDepto
D1	Compras
D2	Engenharia
D3	Vendas

Emp

CódigoEmp	Nome	CódigoDepto	CIC
E1	Souza	D1	132123123
E2	Santos	D2	123124112
E3	Silva	D2	125323422
E5	Soares	D1	543523345

Chave Estrangeira

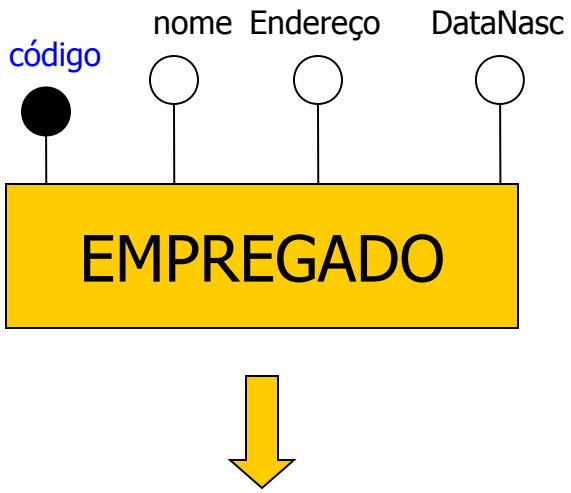
- Observações:
  - No exemplo anterior, a coluna **CodigoDepto** da tabela **Emp** é uma **chave estrangeira em relação à chave primária da tabela Dept**
  - Na tabela Emp, os valores do campo **CodigoDepto** de todas as linhas devem aparecer na coluna **CodigoDepto** da tabela Dept
- A existência de uma chave estrangeira **impõe restrições** que devem ser garantidas ao executar as diversas operações e alteração do banco de dados

Mapeamento do Modelo Conceitual para o Modelo Relacional

- Um modelo conceitual construído utilizando o Modelo Entidade-Relacionamento pode ser mapeado para um modelo lógico Relacional
  - A equivalência mais direta é
    cada Entidade = uma Tabela
    cada Atributo = uma Coluna
    cada Ocorrência = uma Linha

Mapeamento de Entidades

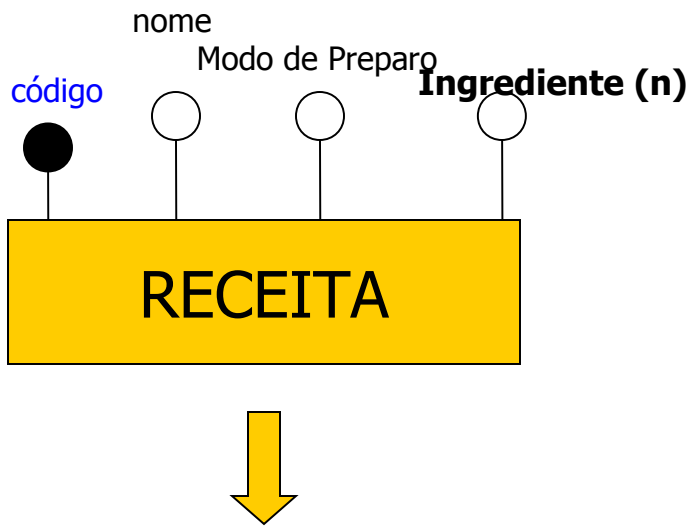
- Para cada Entidade deve ser criada uma relação (tabela)
- Para cada atributo simples incluir uma coluna na tabela
- No caso de atributo composto, incluir somente os atributos simples que o compõe (Endereço)



Empregado(Codigo, Nome, Logradouro, Numero, Bairro, Cidade, Estado, DataNasc)

Mapeamento de Atributos Multivalorados

- Para cada atributo multivalorado deve ser criada uma tabela formada pela chave primária da Tabela/Entidade e pelo atributo multivalorado
- A chave primária da nova tabela será o par de atributos
- Se o atributo multivalorado for composto. Por ex.: Ingrediente formado por Nome do ingrediente e quantidade, todo o grupo vai para a nova tabela



Receita(Codigo, Nome, Modo_Preparo)
Ingrediente_Receita (CodReceita, Ingrediente)

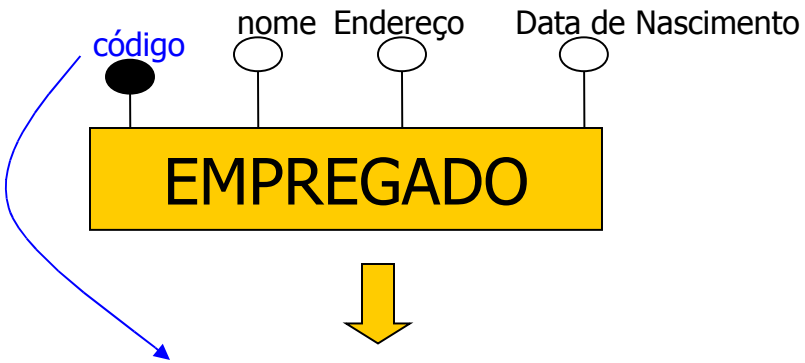
Nome das colunas

- O nome do atributo(modelo conceitual) pode ser diferente do nome do campo da tabela (coluna)
- Objetivo
  - Facilidade de programação como o uso de nomes curtos
  - Evitar nomes iguais. Ex.: Todas entidades com o atributo nome
  - Evitar espaços no nome da coluna, pois são proibidos nos BD relacionais
- Dica
  - Defina um padrão para converter o nome dos atributos, principalmente dos nomes compostos que necessitam abreviação.

Ex.: Nome do Responsável → NomeRes

Nome para a chave primária

- É boa prática compor o nome da chave primária com uma identificação da tabela a qual ela pertence



Empregado(**CodEmp**, Nome, Endereco, DataNasc)

- Como geralmente, elas se tornam chaves estrangeiras de outras tabelas, esta prática facilita a programação e compreensão dos campos

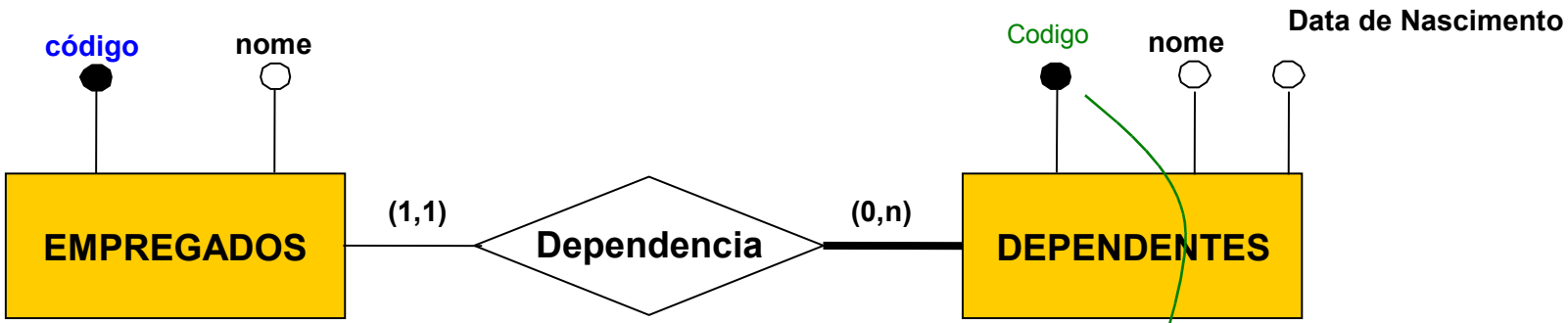
Mapeamento de Relacionamentos

- Um relacionamento pode ser transformado em:
  - Uma tabela
  - Uma coluna de uma das tabelas envolvidas
  - Fusão de duas tabelas

- Esta decisão depende da cardinalidade mínima e máxima dos relacionamentos
- No caso da fusão devemos considerar também outros relacionamentos da entidade

Entidades Fracas

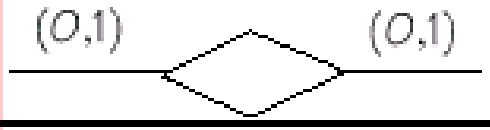
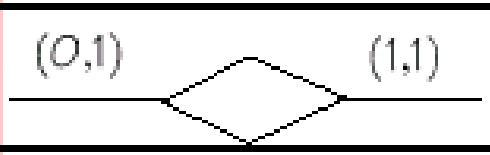
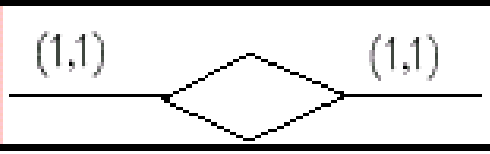
- Criar uma tabela para cada entidade fraca
- Nessa tabela incluir como chave estrangeira a chave primária da tabela que representa a entidade possuidora/identificadora
- As entidades fracas têm chave primária composta de duas partes:
  - A chave primária da tabela (entidade) possuidora
  - A chave parcial da tabela(entidade) fraca



Empregados(CodEmp, Nome)

Dependentes(CodEmp, CodDep, Nome, DataNasc)

Implementação de Relacionamentos 1:1

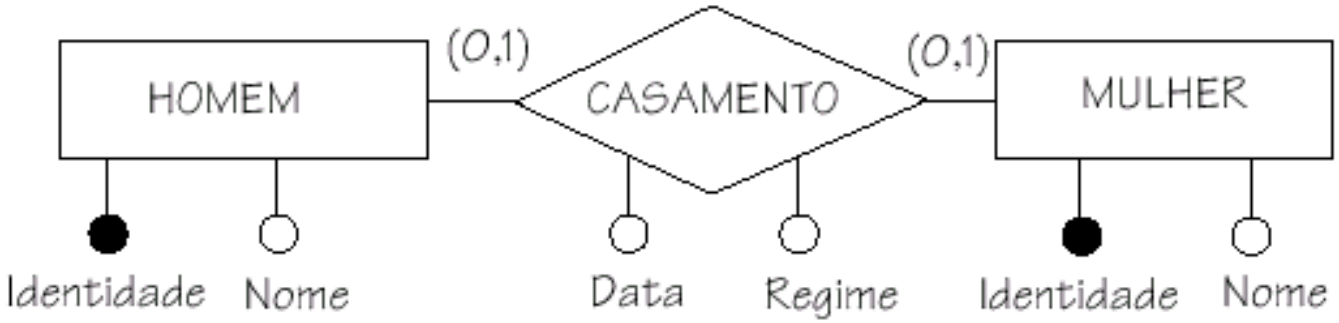
Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição coluna	Fusão tabelas
	±	✓	×
	×	±	✓
	×	×	✓

✓ Alternativa preferida ± Pode ser usada

× Não usar

Relacionamentos Binários - Um para Um

- Ambas entidades têm participação opcional
  - adição de colunas



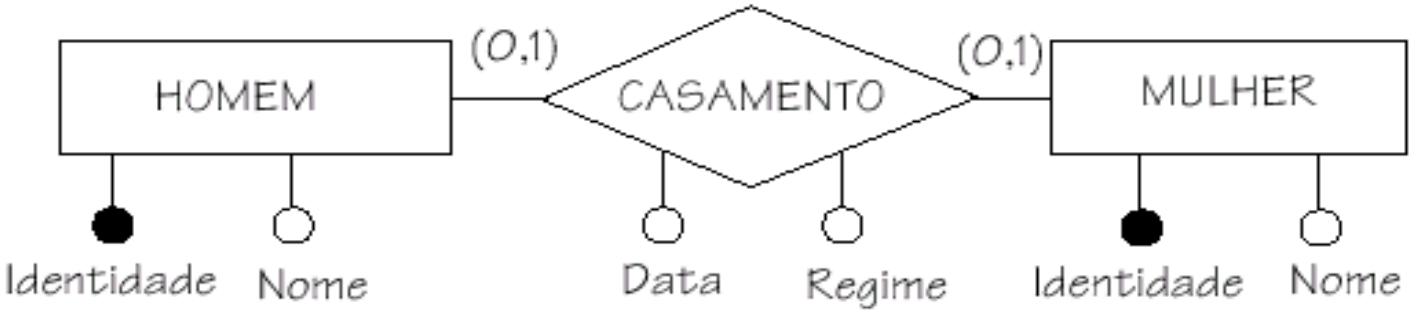
Mulher (IdentM,Nome,**IdentH,Data,Regime)**

IdentH referencia Homem

Homem (IdentH,Nome)

Relacionamentos Binários - Um para Um

- Ambas entidades têm participação opcional
  - tabela própria



Mulher (IdentM,Nome)

Homem (IdentH,Nome)

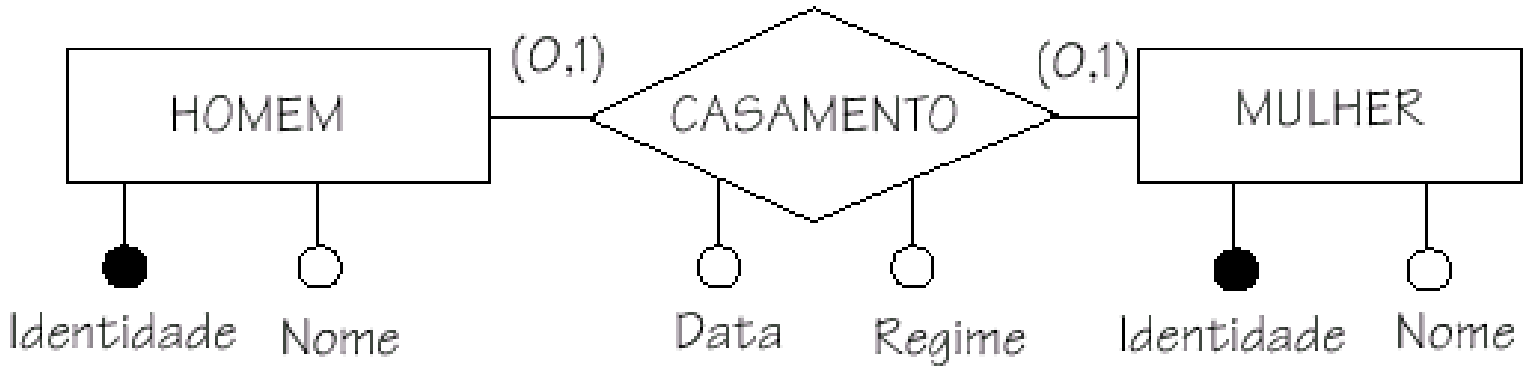
Casamento (IdentM,IdentH,Data,Regime)

IdentM referencia Mulher

IdentH referencia Homem

Relacionamentos Binários - Um para Um

- Ambas entidades têm participação opcional
  - fusão de tabelas

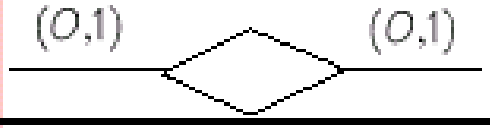
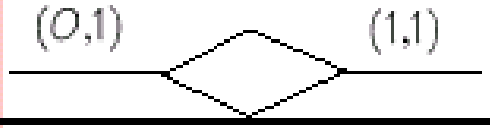


Casamento (IdentM,IdentH,Data,Regime,**NomeH,NomeM**)

Relacionamentos Binários - Um para Um

- Ambas entidades têm participação opcional
- Solução por fusão de tabelas é inviável
  - Chave primária artificial (já que pode não estar completa)
- Solução por adição de colunas: melhor
  - Menor número de junções
  - Menor número de chaves
- Solução por tabela própria: aceitável

Implementação de Relacionamentos 1:1

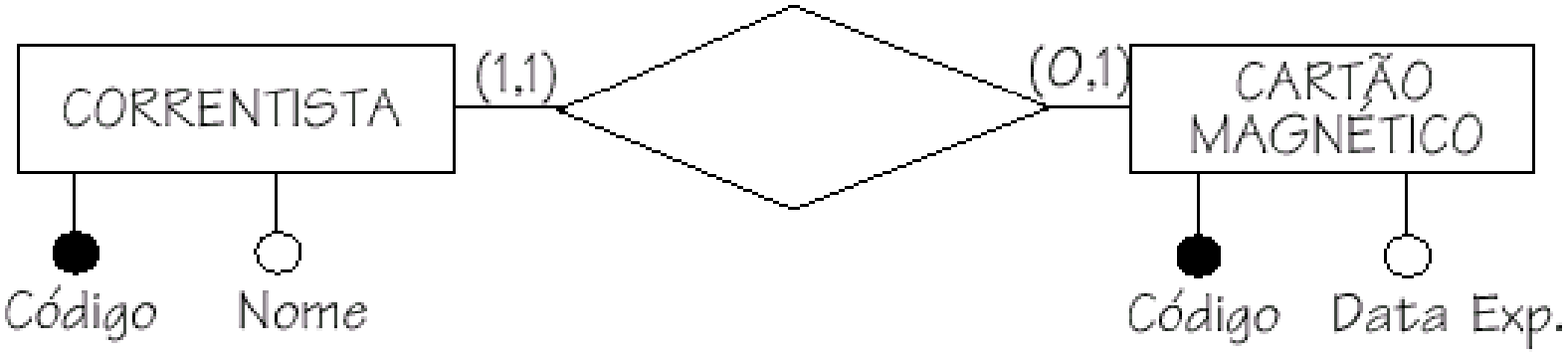
Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição coluna	Fusão tabelas
	±	✓	×
	×	±	✓
	×	×	✓

✓ Alternativa preferida ± Pode ser usada

× Não usar

Relacionamentos Binários - Um para Um

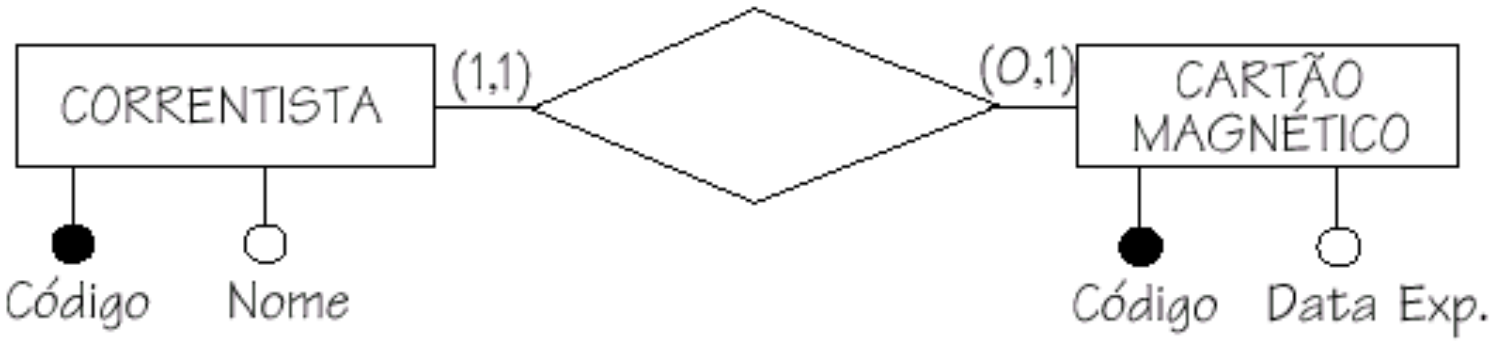
- Participação opcional/obrigatória
  - Fusão de tabelas



Correntista (CodCorrent,Nome,CodCartão,DataExp)

Relacionamentos Binários - Um para Um

- Participação opcional/obrigatória
  - Adição de colunas

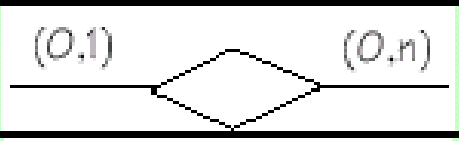
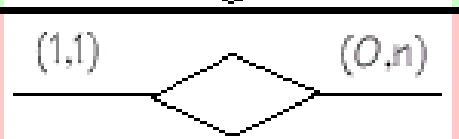
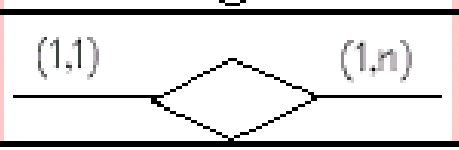


Correntista (CodCorrent,Nome)
Cartão(CodCartão,DataExp,**CodCorrent**)
CodCorrent referencia Correntista

Relacionamentos Binários - Um para Um

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

Relacionamentos Binários - Um para Muitos

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

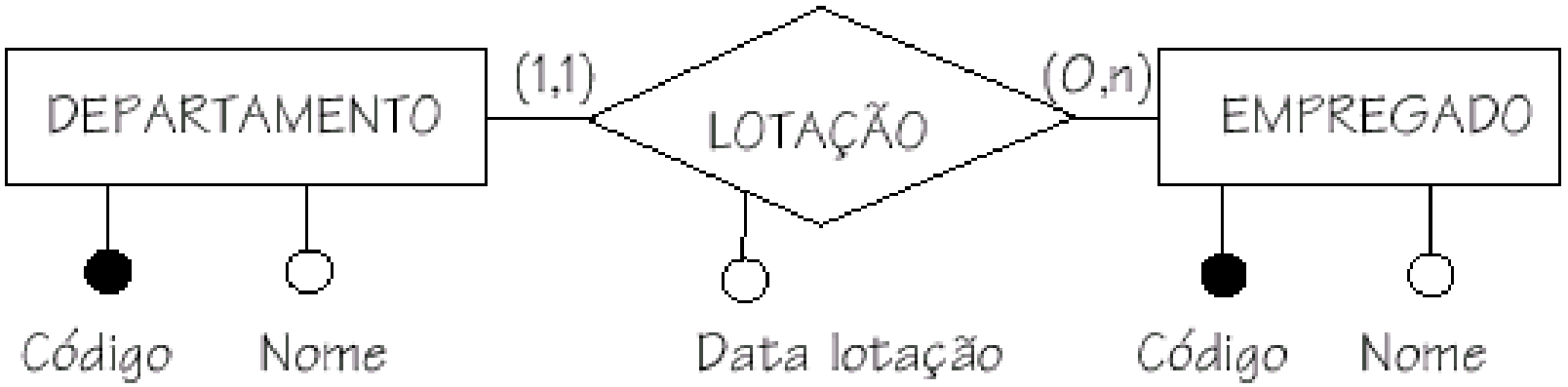
✓ Alternativa preferida ± Pode ser usada

× Não usar

Relacionamentos Binários - Um para Muitos

- Já temos duas tabelas, uma para cada conjunto de entidades que participam do relacionamento
- Incluir como chave estrangeira, na tabela do “lado muitos” (o “lado N”), a chave primária da tabela do “lado um”
- Incluir também colunas com os atributos do relacionamento

Relacionamentos Binários - Um para Muitos

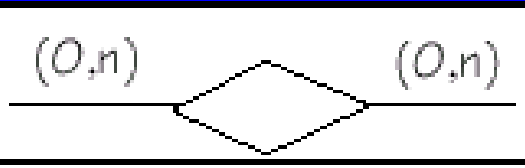
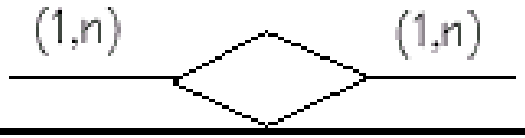


Departamento (CodDept,Nome)

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

CodDept referencia Departamento

Relacionamentos Binários – Muitos para Muitos

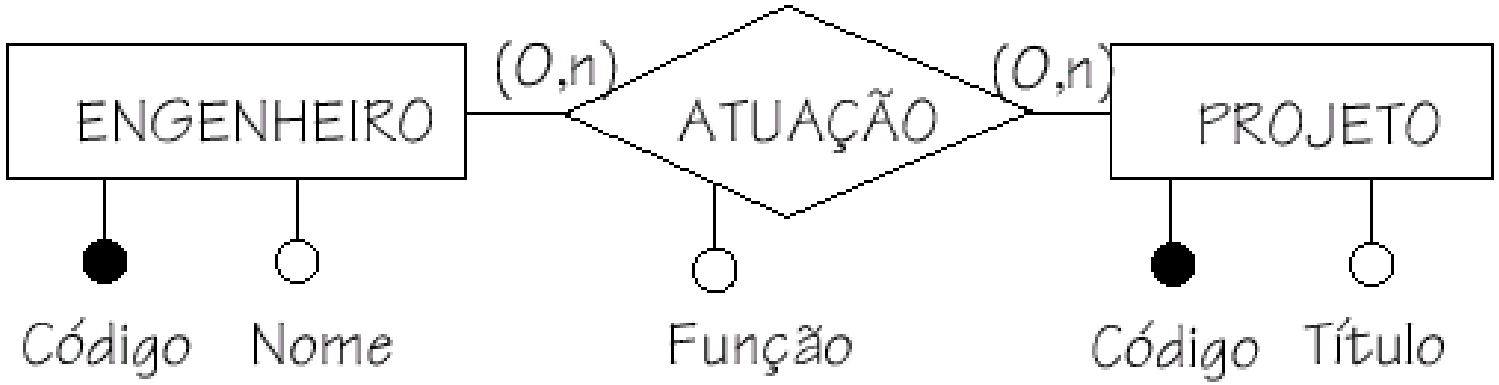
Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição coluna	Fusão tabelas
	✓	×	×
	✓	×	×
	✓	×	×

✓ Alternativa preferida × Não usar

Relacionamentos Binários – Muitos para Muitos

- Já temos duas tabelas, uma para cada conjunto de entidades que participa do relacionamento
- Criar uma nova tabela contendo, como chaves estrangeiras, as chaves primárias dessas duas tabelas
  - A combinação dessas chaves estrangeiras forma a chave primária da nova tabela
- Incluir também colunas com os atributos do relacionamento

Relacionamentos Binários – Muitos para Muitos



Engenheiro (CodEng,Nome)

Projeto (CodProj,Título)

Atuação (CodEng,CodProj,Função)

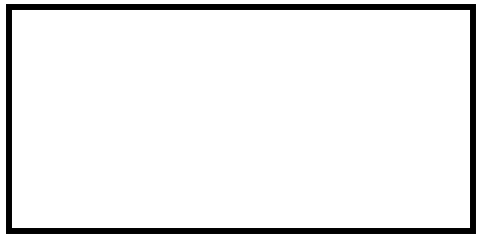
CodEng referencia Engenheiro

CodProj referencia Projeto

Modelo Lógico de Dados
Modelo Relacional
Parte 2
Restrições de Integridade
Generalizações / Especializações

Diagrama para Modelo Lógico Relacional

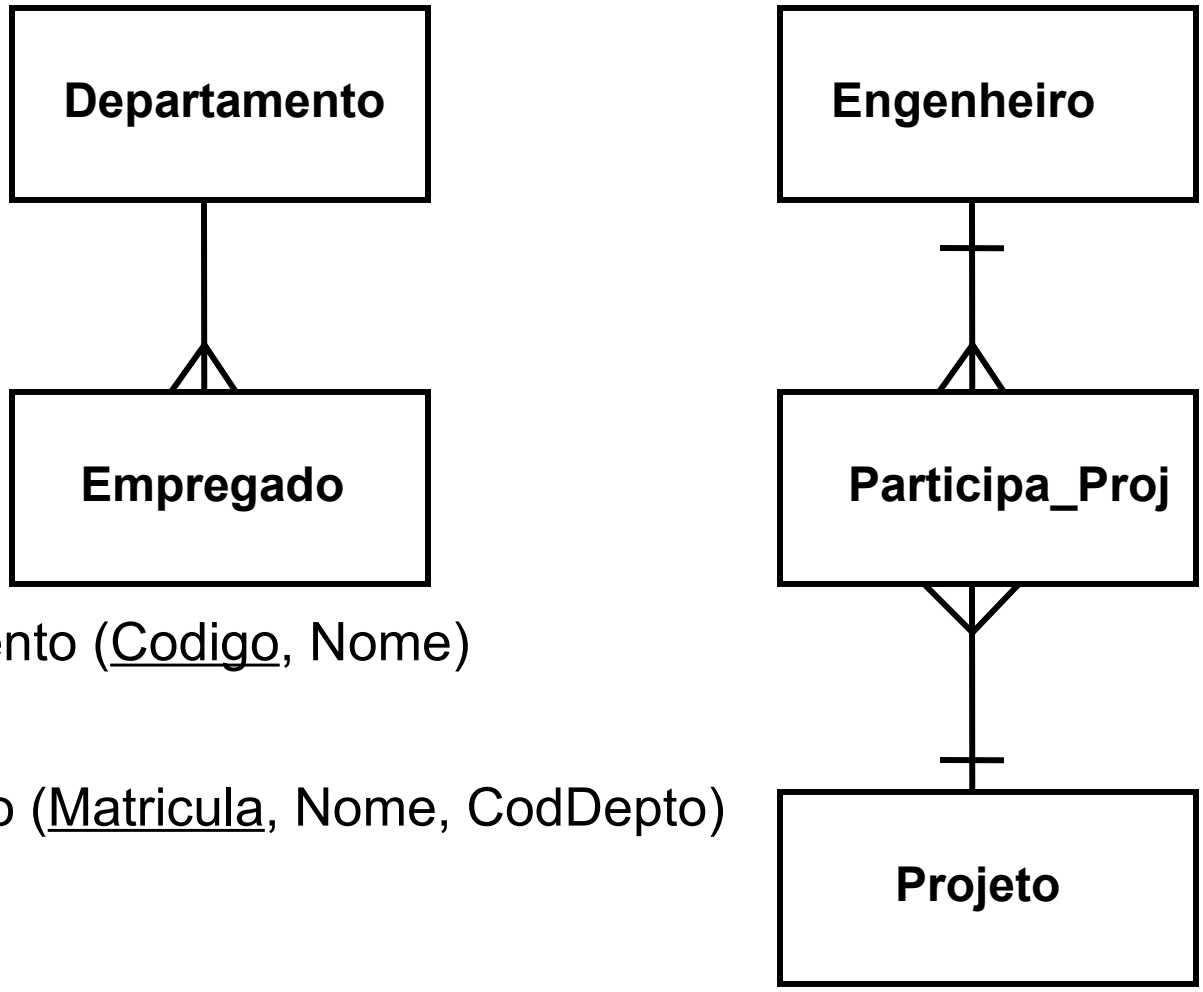
- Tabela: representada por um retângulo



- Relacionamentos somente binários (entre 2 tabelas) – tipo 1:N

- Se for obrigatório, corte transversal:

Exemplos



Departamento (Codigo, Nome)

Empregado (Matricula, Nome, CodDepto)

Chave Estrangeira - restrições impostas

- **Quando da inclusão de uma linha na tabela que contém a chave estrangeira:**

  deve ser garantido que o valor da chave estrangeira, se informado, exista na coluna da chave primária referenciada
  Se o relacionamento não for do tipo obrigatório, a chave estrangeira pode não ser informada

Exemplo: um novo empregado deve atuar em um departamento já existente no banco de dados (na tabela Departamento)

Chave Estrangeira - restrições impostas

- **Quando da alteração do valor da chave estrangeira:**

  deve ser garantido que o novo valor de uma chave estrangeira, se informado, exista na coluna da chave primária referenciada

Exemplo: se um empregado muda de departamento, deve-se garantir que o novo departamento já exista no banco de dados (na tabela Departamento)

Chave Estrangeira - restrições impostas

- **Quando da exclusão de uma linha da tabela que contém a chave primária referenciada pela chave estrangeira:**

  deve ser garantido que na coluna chave estrangeira não apareça o valor da chave primária que está sendo excluída

Exemplo: um departamento não pode ser excluído, caso exista algum empregado que o referencie

Chave Estrangeira - restrições impostas

- **Quando da alteração do valor da chave primária referenciada pela chave estrangeira:**

  deve ser garantido que na coluna chave estrangeira não apareça o antigo valor da chave primária que está sendo alterada

Exemplo: um departamento somente pode ter o seu código alterado se não for referenciado por nenhum empregado

Chave Estrangeira

- Pode acontecer de uma chave estrangeira **referenciar a chave primária da própria tabela** (e não de uma tabela diferente, como no exemplo anterior). **Exemplo:**

Emp

CódigoEmp	Nome	CódigoDepto	Código Emp Gerente
E1	Souza	D1	-
E2	Santos	D2	E5
E3	Silva	D2	E5
E5	Soares	D1	E1

Código Emp Gerente é uma chave estrangeira que Referencia a chave da própria tabela (auto-relacionamento)

Valores Vazios

- Normalmente, os SGBDs relacionais exigem que **todas as colunas que compõem a chave primária sejam obrigatórias**
- A mesma exigência não é feita para as demais chaves
- **Atenção:** Vazio é diferente de zeros ou brancos

Restrições de Integridade

- **Banco de dados íntegro**: reflete corretamente a realidade representada pelo banco de dados e os dados são consistentes entre si
- Um SGBD deve garantir a **integridade dos dados** através do mecanismo de **restrição de integridade**

Restrições de Integridade

- **Restrição de integridade**: regra de consistência de dados que é garantida pelo próprio SGBD
- Na abordagem relacional as restrições de integridade estão classificadas em:
  - Integridade de **domínio**
  - Integridade de **vazio**
  - Integridade de **chave**
  - Integridade **referencial**

Restrições de Integridade

- **Domínio:**
  - É um **conjunto de valores** (alfanumérico, numérico, ...) que um campo de uma tabela pode assumir
  - É chamado de **domínio da coluna** ou **domínio do campo**
- **Integridade de domínio:**
  - especifica que o valor de um campo deve obedecer a definição de valores admitidos para a coluna (o domínio da coluna)
  - Nos SGBDs relacionais é possível usar domínios pré-definidos (número inteiro, número real, alfanumérico de tamanho definido, data, ...)
  - É possível usar um conjunto finito de valores:
    (Masc , Fem), (Sim , Não), (1, 2, , 5, 7, 11, 13)
  - É possível usar uma fórmula: > 0 e < 1000

Restrições de Integridade

- Valores vazios:
  - Deve-se especificar se o conteúdo de uma coluna pode estar vazio (em inglês “null”)
  - Estar vazio significa que o campo **não recebeu nenhum valor** de seu domínio
  - As colunas que podem conter valores vazios são as colunas **opcionais**
  - As colunas que não podem conter valores vazios são as colunas **obrigatórias**
- **Integridade de vazio:**
  - Especifica se os campos de uma coluna podem ou não ser vazios (se a coluna é obrigatória ou opcional)
  - Campos que compõem a chave primária não podem ser vazios

Restrições de Integridade

- **Integridade de chave:**
  - Trata-se da restrição que define que os valores da chave primária devem ser únicos

- **Integridade referencial:**
  - Define que os valores dos campos que aparecem em uma chave estrangeira devem aparecer na chave primária da tabela referenciada

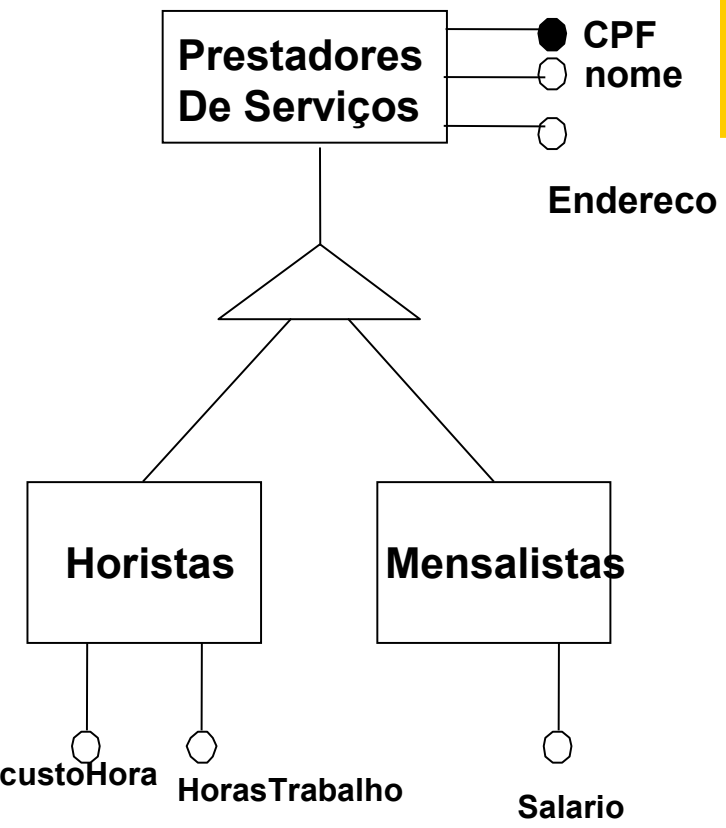
Restrições de Integridade Declarativas

- As restrições citadas devem ser **garantidas automaticamente por um SGBD relacional a partir da sua declaração na definição das tabelas,** não sendo necessário nenhuma programação para garanti-las explicitamente
- São restrições de Integridade Declarativa:
  - Chave Primária
  - Unicidade (chaves candidatas)
  - Integridade Referencial (Chave Estrangeira)
  - Integridade de Domínio
  - Integridade de Vazio

Restrições Semânticas de Integridade

- Existem restrições de integridade que não podem ser definidas de forma declarativa para um SGBD relacional
- São chamadas **restrições semânticas**
- Necessário que seja feita programação "procedural" na aplicação ou no SGBD
- **Exemplos**:
  - um empregado do departamento "Finanças" não pode ter a categoria funcional "Engenheiro"
  - um empregado não pode ter um salário maior que seu superior imediato

Generalização/Especialização (1 Tabela)



Alternativa (1 tabela):

PrestadoresServicos(CPF, Nome, Endereco,
CustoHora, HorasTrabalhadas, Salario, Tipo)

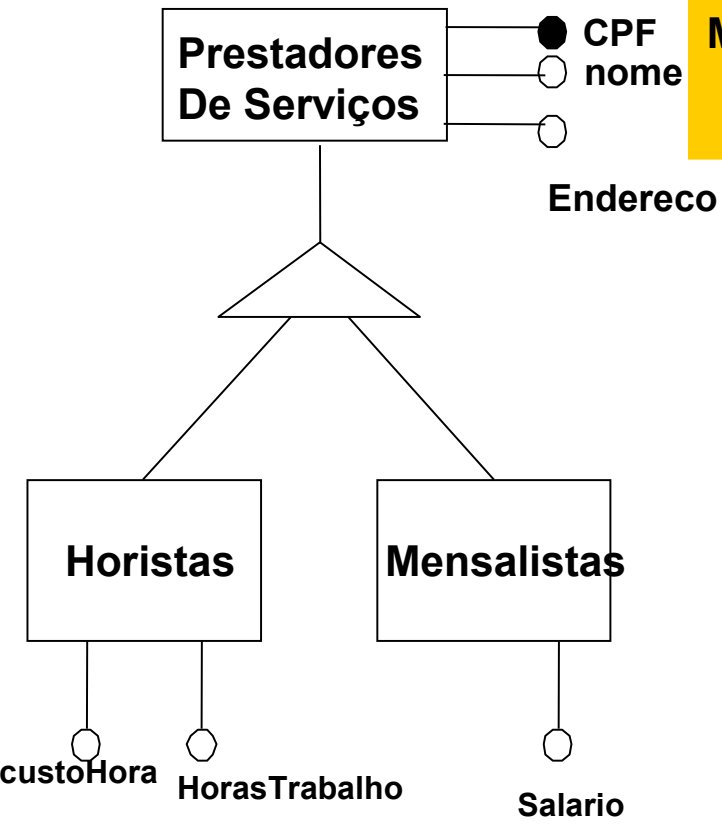
Regras

- Chave primária referente a entidade genérica
- Adicionar uma coluna Tipo
- Uma coluna para cada atributo da entidade genérica e das especializadas
- Possíveis colunas referentes aos relacionamentos envolvendo as entidades

Observações

- Minimizando a junções
- Menor número de chaves
- Muitos atributos opcionais

Generalização/Especialização (2 Tabelas)



Alternativa (2 tabelas): Horistas, Mensalistas

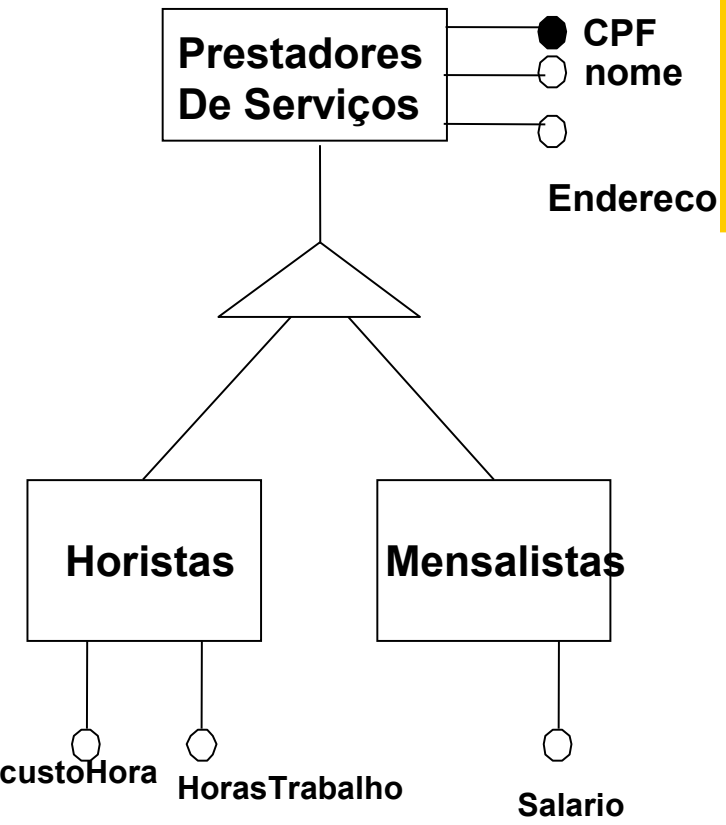
Horistas(CPF,Nome,Endereco,CustoHora,HorasTrabalho)

Mensalistas(CPF, Nome, Endereco Salario)

Observações

- Unicidade da chave primária não pode ser garantida de forma declarativa pelo SGBD, por serem 2 tabelas independentes
- Necessário programação procedural

Generalização/Especialização (3 Tabelas)



Abordagem geral (3 tabelas):
Prestadores_de_Servicos, Horistas, Mensalistas

Prestadores_de_Servicos(CPF, Nome, Endereco, Tipo)
Horistas(CPF,CustoHora,HorasTrabalhadas)
Mensalistas(CPF,Salario)

Regras

- Incluir a chave primária da tabela genérica em cada tabela especializada como chave estrangeira
- Adicionar o atributo Tipo

Observações

- Redução de atributos opcionais
- Ocorrência de uma das 2 tabelas especializadas conforme o valor do atributo Tipo somente pode ser garantida através de programação procedural