



# Regras Gerais de Tradução

▶ Evitar junções

▶ Diminuir o número de chaves

▶ Evitar campos opcionais

## Junção

▶ Junção:

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

Exemplo: Buscar os dados de um funcionário e os dados de seu departamento (dados presentes em 2 tabelas diferentes)

▶ Evitar 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

Cliente (CódCliente, Nome, Endereço, Telefone)

OU

Cliente (CódCliente, Nome)

ClienteEndereço (CódCliente, Endereço, Telefone)



# Campos Opcionais

## ► Campo opcional:

- Campo que pode assumir o valor vazio (*null*)
- 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 a programação
  - Verificar quais campos podem estar vazios, quando isto depende do tipo de linha
- Regra mais “fraca” que as precedentes

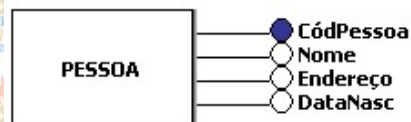


## Passos da Transformação ER para Relacional

- ▶ Tradução inicial de entidades e respectivos atributos
- ▶ Tradução de relacionamentos e respectivos atributos
- ▶ Tradução de generalizações/especializações

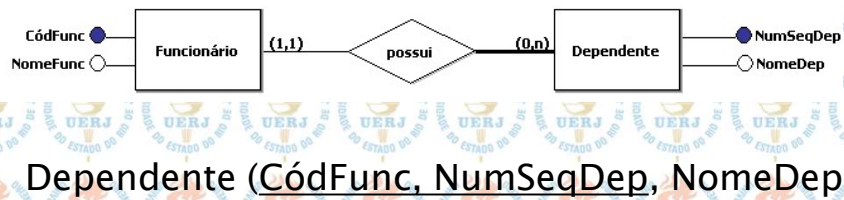
## Implementação Inicial de Entidades

- ▶ 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
- ▶ Tradução inicial:
  - Regras que seguem podem fundir tabelas



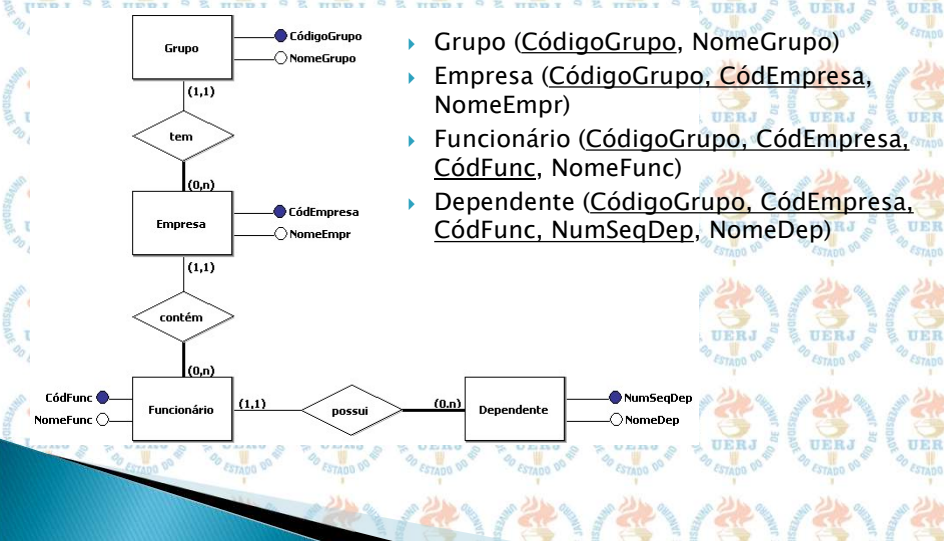
Pessoa (CódPessoa, Nome, Endereço, DataNasc)

## Implementação de Entidade Fraca



Dependente (CódFunc, NumSeqDep, NomeDep)

## Entidade Fraca – Recursão





# Coluna e Relacionamento

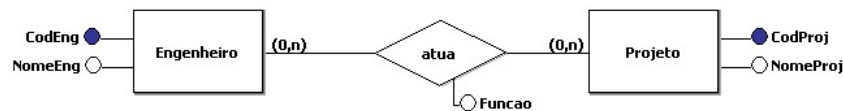
## ► Nomes de Colunas

- Referenciados frequentemente 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

## ► Implementação de Relacionamento

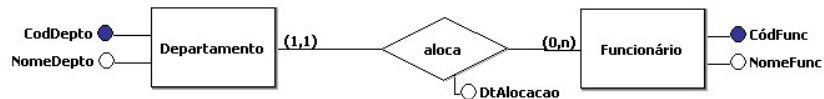
- Tabela própria
- Adição de colunas a uma das tabelas
- Fusão de tabelas
- Alternativa depende da cardinalidade (máxima e mínima do relacionamento)

# Tabela Própria



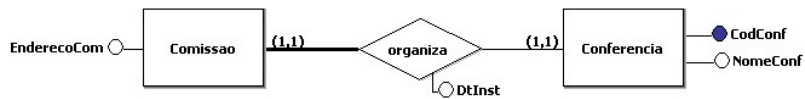
- Engenheiro (CodEng, NomeEng)
- Projeto (CodProj, NomeProj)
- Atua (CodEng, CodProj, Funcao)

## Adição de Colunas



▶ Departamento (CodDepto, NomeDepto)  
 ▶ Funcionario (CódFunc, NomeFunc, CodDepto, DtAlocacao)

## Fusão de Tabelas



▶ Conferencia (CodConf, NomeConf, DtInst, EnderecoCom)



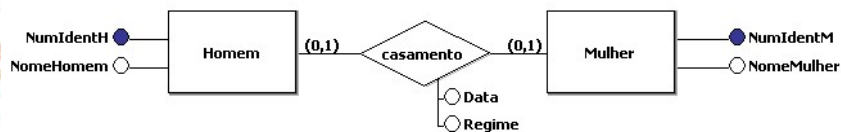
## Implementação de Relacionamentos 1:1

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
$(0,1) \text{---} \diamond \text{---} (0,1)$	☹	☺	☹
$(0,1) \text{---} \diamond \text{---} (1,1)$	☹	☹	☺
$(1,1) \text{---} \diamond \text{---} (1,1)$	☹	☹	☺

### Legenda:

- ☺ - Alternativa Preferida
- ☹ - Pode ser usada
- ☹ - Não usar

## 1:1 Ambas Entidades Opcionais



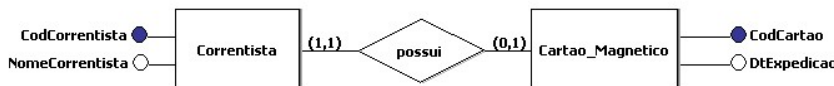
### Tabela Própria

- Mulher (NumIdentM, NomeMulher)
- Homem (NumIdentH, NomeHomem)
- Casamento (NumIdentM, NumIdentH, Data, Regime)

### Adição de Colunas

- Mulher (NumIdentM, NomeMulher, NumIdentH, Data, Regime)
- Homem (NumIdentH, NomeHomem)

## 1.1 – Uma Entidade Opcional e Outra Obrigatória



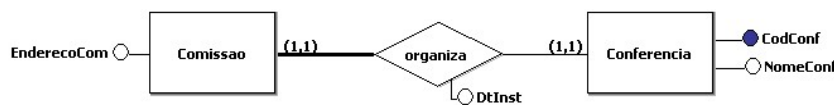
### ► Fusão de Tabelas

- Correntista (CodCorrentista, NomeCorrentista, CodCartao, DtExpedicao)

### ► Adição de Colunas

- Correntista (CodCorrentista, NomeCorrentista)
- Cartao (CodCartao, DtExpedicao, CodCorrentista)

## 1:1 – Ambas Entidades têm Participação Obrigatória



### ► Fusão de Tabelas

- Conferencia (CodConf, NomeConf, DtInst, EnderecoCom)



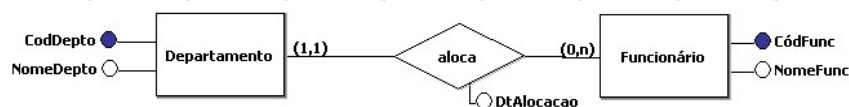
## Implementação de Relacionamentos 1:N

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
	☹️	😊	☹️
	☹️	😊	☹️
	☹️	😊	☹️
	☹️	😊	☹️

### Legenda:

- 😊 - Alternativa Preferida
- ☹️ - Pode ser usada
- ☹️ - Não usar

1:N - Entidade que tem cardinalidade máxima 1 e é obrigatória

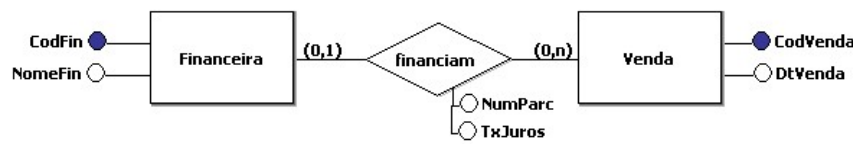


### Adição de Colunas

Departamento (CodDepto, NomeDepto)

Funcionario (CódFunc, NomeFunc, CodDepto, DtAlocacao)

1:N - Entidade que tem cardinalidade máxima 1 e é opcional



### Adição de Colunas

- Financieira (CodFin, NomeFin)
- Venda (CodVenda, DtVenda, CodFin, NumParc, TxJuros)

## Implementação de Relacionamentos N:M

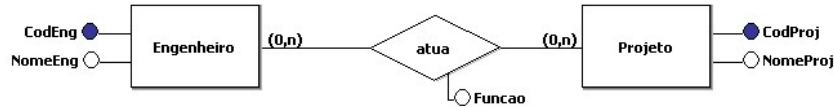
Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
(0,n) — (0,n)	☺	☹	☹
(0,n) — (1,n)	☺	☹	☹
(1,n) — (1,n)	☺	☹	☹

### Legenda:

- ☺ - Alternativa Preferida
- ☹ - Não usar



## Relacionamentos N:M



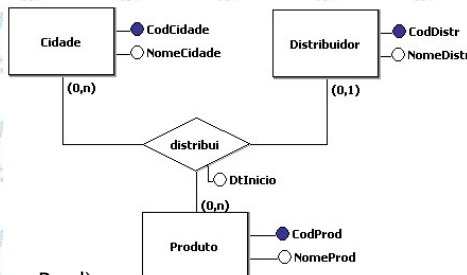
### ► Tabela própria

- Engenheiro (CodEng, NomeEng)
- Projeto (CodProj, NomeProj)
- Atua (CodEng, CodProj, Funcao)

## Implementação de Relacionamento N-ário

### ► Não são definidas regras específicas

- O relacionamento é transformado em uma entidade
- São aplicadas regras de implementação dos relacionamentos binários



► Produto (CodProd, NomeProd)

► Cidade (CodCidade, NomeCidade)

► Distribuidor (CodDistr, NomeDistr)

► Distribui (CodProd, CodDistr, CodCid, DtInicio)

## Implementação de Generalização/Especialização

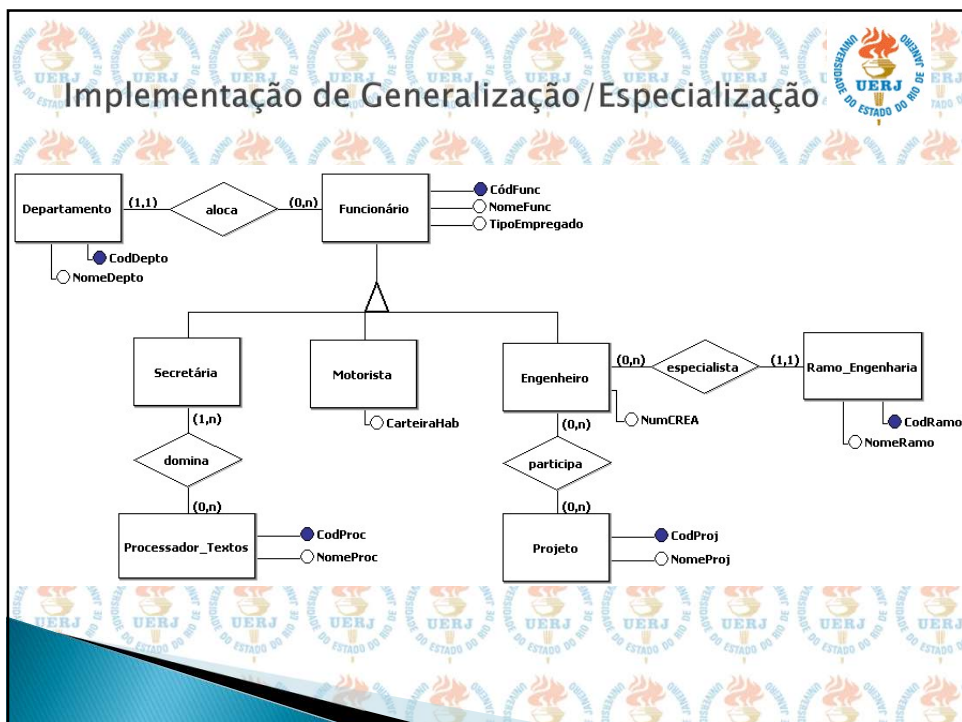
### ► Duas alternativas básicas

- Uso de uma única tabela para toda hierarquia
- Uso de uma tabela para cada entidade

### ► Outra alternativa (exótica)

- Subdivisão de entidade genérica

## Implementação de Generalização/Especialização





## Uma Tabela por Hierarquia

- ▶ Todas as tabelas referentes às especializações são fundidas em uma única tabela
- ▶ Tabela contém:
  - Chave primária correspondente ao identificador da entidade mais genérica
  - Adicionar uma coluna Tipo, caso já não exista
  - Uma coluna para cada atributo da entidade genérica
  - Colunas referentes aos relacionamentos, dos quais a entidade genérica participa e que sejam implementados, através da alternativa de adicionar colunas à tabela da entidade genérica
  - Uma coluna para cada atributo de cada entidade especializada (opcional)
  - Colunas referentes aos relacionamentos, dos quais cada entidade especializada participa e que sejam implementados, através da alternativa de adicionar colunas à tabela da entidade (campo opcional)

## Uma Tabela por Hierarquia

- ▶ Funcionario (CodFunc, NomeFunc, **TipoFuncionario**, CodDepto, CarteiraHab, NumCREA, CodRamo)
- ▶ Departamento (CodDepto, NomeDepto)
- ▶ Ramo\_Engenharia (CodRamo, NomeRamo)
- ▶ Processador\_Textos (CodProc, NomeProc)
- ▶ Domina (CodFunc, CodProc)
- ▶ Projeto (CodProj, NomeProj)
- ▶ Participa (CodFunc, CodProj)

## Uma Tabela por Entidade Especializada

- ▶ 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
- ▶ Funcionario (CodFunc, NomeFunc, **TipoFuncionario**, CodDepto)
- ▶ Motorista (CodFunc, CarteiraHab)
- ▶ Engenheiro (CodFunc, NumCREA, CodRamo)
- ▶ Departamento (CodDepto, NomeDepto)
- ▶ Ramo\_Engenharia (CodRamo, NomeRamo)
- ▶ Processador\_Textos (CodProc, NomeProc)
- ▶ Domina (CodFunc, CodProc)
- ▶ Projeto (CodProj, NomeProj)
- ▶ Participa (CodFunc, CodProj)

Secretária não possui tabela específica porque não possui atributo específico

## 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
- ▶ Desvantagens:
  - Unicidade da chave primária
    - 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 funcionários como um todo.



## Subdivisão da Entidade Genérica

- ▶ FuncOutros (CodFunc, NomeFunc, **TipoFuncionario**, CodDepto)
- ▶ Motorista (CodFunc, NomeFunc, CodDepto, CarteiraHab)
- ▶ Engenheiro (CodFunc, NomeFunc, CodDepto, NumCREA, CodRamo)
- ▶ Departamento (CodDepto, NomeDepto)
- ▶ Ramo\_Engenharia (CodRamo, NomeRamo)
- ▶ Processador\_Textos (CodProc, NomeProc)
- ▶ Domina (CodFunc, CodProc)
- ▶ Projeto (CodProj, NomeProj)
- ▶ Participa (CodFunc, CodProj)

## Comparação entre Implementações

- ▶ Tabela única:
  - Dados referentes à entidade genérica + dados referentes às especializações em uma única linha
  - Minimiza junções
  - Menor número de chaves
- ▶ Uma tabela por entidade especializada:
  - Colunas opcionais:
    - Apenas aquelas referentes a atributos que podem ser vazios do ponto de vista da aplicação
- ▶ Subdivisão da entidade genérica:
  - Dados da entidade especializada + dados da entidade genérica
  - Não há como garantir unicidade de chave primária

## Refinamento do Modelo Relacional

### ► Algumas vezes

- Esquema de BD criado através do uso das regras anteriores 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 sempre piores

## Refinamento do Modelo Relacional

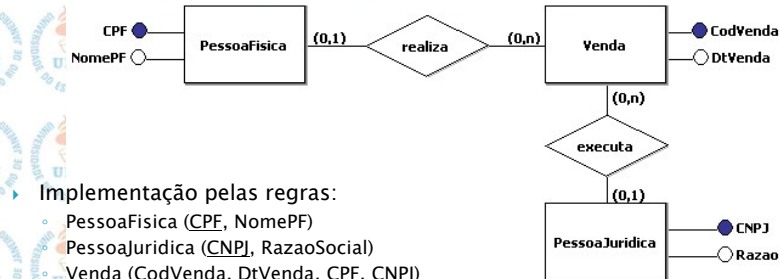
### ► Relacionamentos mutuamente exclusivos

### ► Simulação de atributos multivalorados

### ► Informações redundantes



## Relacionamentos Mutuamente Exclusivos



### Implementação pelas regras:

- PessoaFisica (CPF, NomePF)
- PessoaJuridica (CNPJ, RazaoSocial)
- Venda (CodVenda, DtVenda, CPF, CNPJ)
- Colunas CPF e CNPJ em Venda são especificadas como opcionais.

### Implementação alternativa:

- Criar uma única coluna na qual aparece o CPF ou o CNPJ do comprador
- PessoaFisica (CPF, NomePF)
- PessoaJuridica (CNPJ, RazaoSocial)
- Venda (CodVenda, DtVenda, CPF/CNPJ, TipoComprador)

### Desvantagens:

- Não é possível especificar ao SGBD que o campo CPF/CNPJ é chave estrangeira
- Não referencia uma única tabela

## Tratamento de Atributos Multivalorados

### Implementação padrão:

- Cliente (CodCliente, NomeCliente)
- Telefone (CodCliente, NumTelefone)

### Condições de contorno:

- Raros clientes possuem mais do que 2 telefones. Quando isso ocorrer, é suficiente armazenarmos apenas 2 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 do cliente
- Cliente (CodCliente, NomeCliente, NumTel1, NumTel2)

### Vantagens:

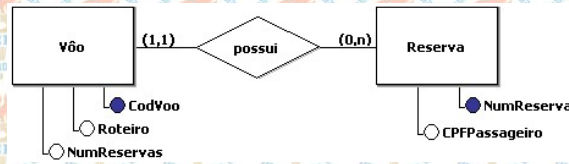
- Permite que os telefones de um cliente sejam obtidos mais rapidamente
- Implica em menos espaço ocupado (não é necessária chave primária da tabela Telefone)

### Desvantagem:

- Consulta usando o número de telefone como critério de busca torna-se mais complicada

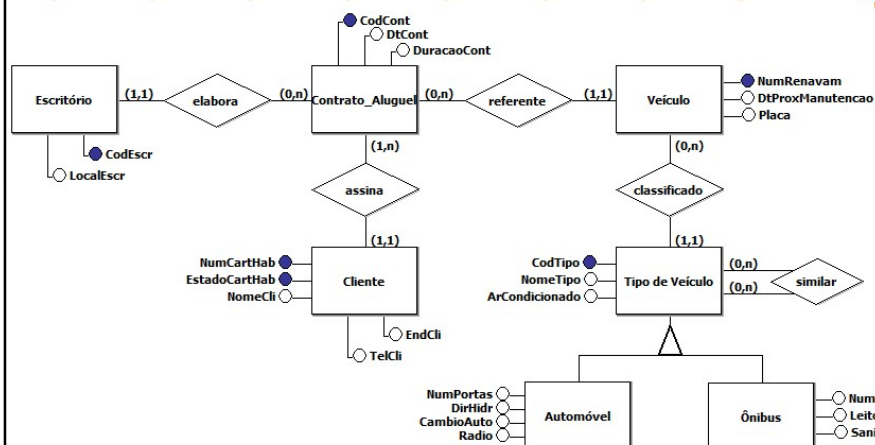


# Informações Redundantes



- ▶ 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

# Exercício



Aplicar os passos de transformação do ER para Relacional.