

Transformação do Modelo ER para o Relacional



Universidade Federal do Maranhão - UFMA

Departamento de Informática

Banco de Dados I

Prof^a.MSc Simara Rocha

simararocha@gmail.com/simara@deinf.ufma.br

www.deinf.ufma.br/~simara

Referências: Elmasri, R. and Navathe, S.B. Sistemas de Bancos de Dados. Editora Addison-Wesley, 6ª edição, 2001.
Date, C.J. Introdução a Sistemas de Bancos de Dados. Editora Campus, 8ª edição, 2004.
Korth, H.F. e Silberschatz, A. Sistemas de Bancos de Dados. Makron Books, 5ª edição, 2006.
Notas de Aula do Prof. Msc. Tiago Eugenio de Melo



Sumário

- Conceitos
- Visão geral do projeto lógico
- Objetivos básicos
- Princípios
- Regras de tradução

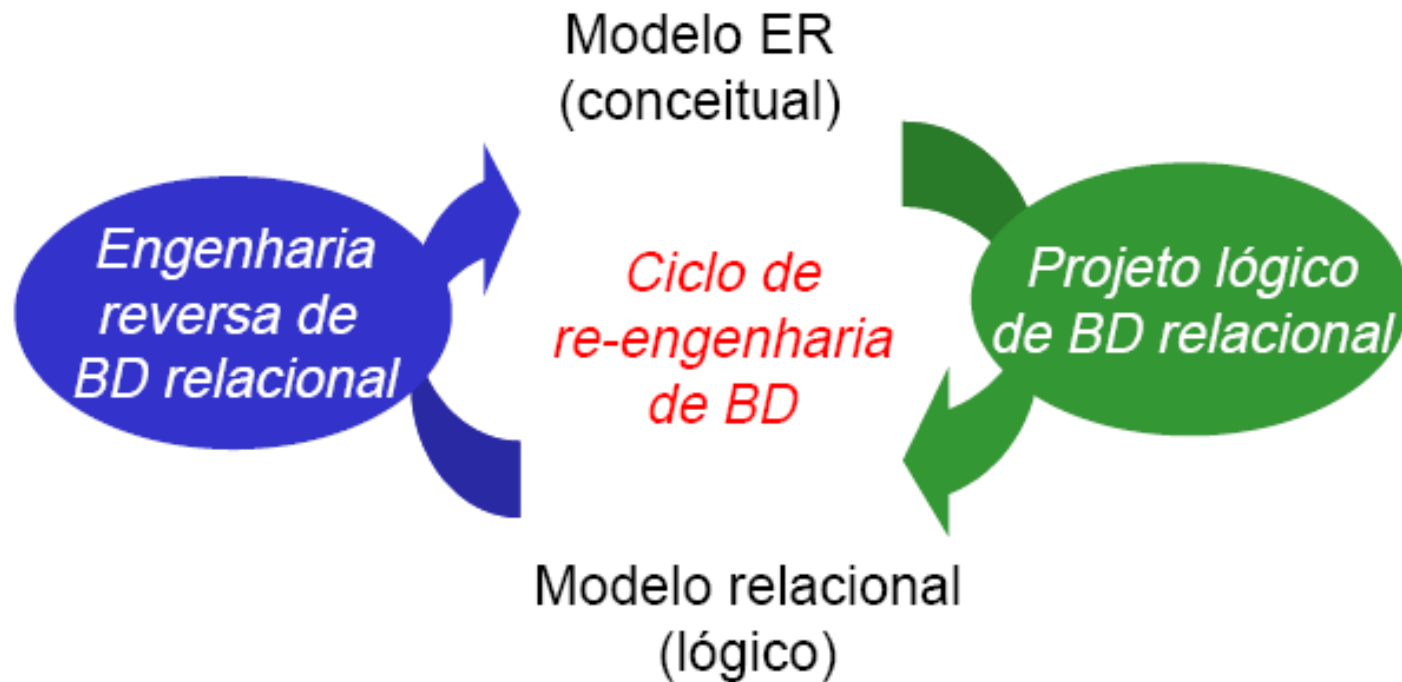


Conceitos

- Regras que definem a transformação de um modelo ER para elementos do modelo relacional
- Regras foram definidas com objetivo de:
 - Obter um BD com bom desempenho quanto à execução das consultas de acesso
 - Obter um BD que simplifique o desenvolvimento e a manutenção das aplicações



Conceitos





Visão Geral do Projeto Lógico

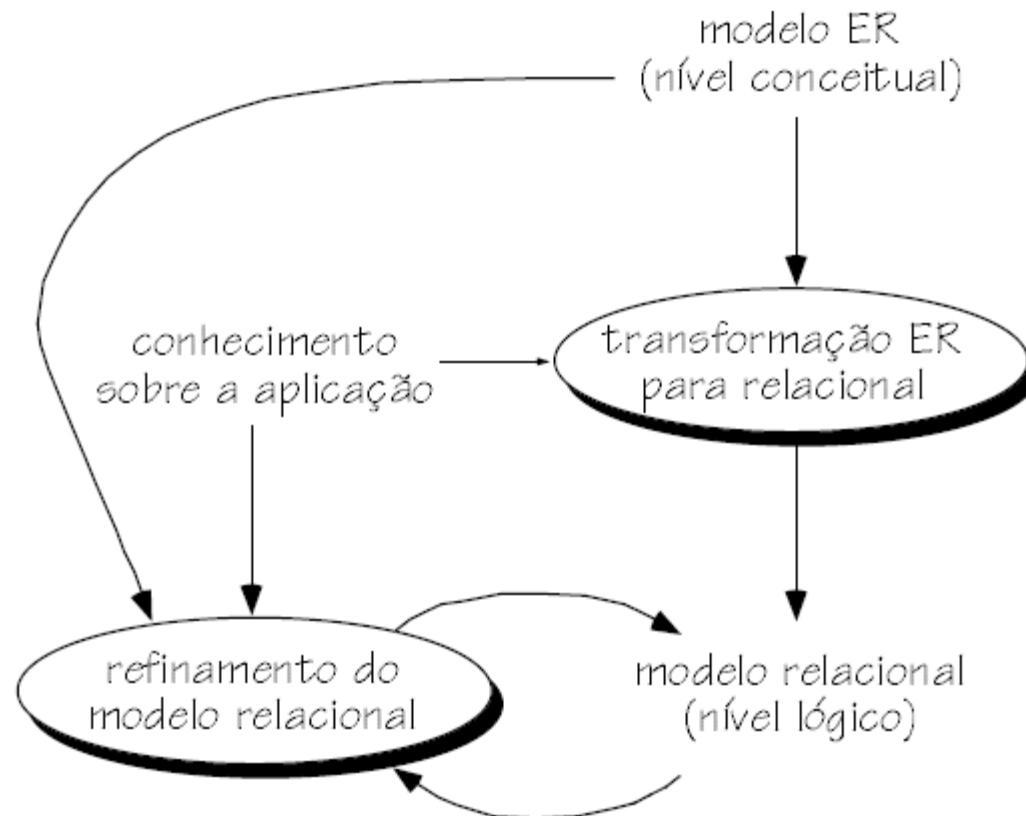
- Um determinado modelo ER pode ser implementado através de diversos modelos relacionais, que contém as informações especificadas pelo diagrama ER
 - Todos podem ser considerados uma implementação correta do modelo ER considerado



Visão Geral do Projeto Lógico

- Entretanto, estes diferentes modelos relacionais podem resultar em diferentes performances do sistema construído sobre o banco de dados
- Além disso, os diferentes modelos relacionais podem implicar maior facilidade, ou dificuldade de desenvolvimento e manutenção do sistema construído sobre o banco de dados

Visão Geral do Projeto Lógico





Regras Gerais

- Aplicáveis à maioria dos casos
- Há situações que, por exigências da aplicação, outros mapeamentos são usados
- Implementadas em ferramentas CASE



Objetivos Básicos

- Obter um banco de dados que permita boa performance de instruções de consulta e alteração do banco de dados
 - Isso significa basicamente diminuir o número de acessos a disco, já que estes consomem o maior tempo na execução de uma instrução de BD
- Obter um banco de dados que simplifique o desenvolvimento e a manutenção de aplicações



Princípios

- Afim de alcançar estes objetivos, as regras de tradução foram definidas tendo por base, entre outros, os seguintes princípios:
 - Evitar junções
 - Diminuir o número de chaves primárias
 - Evitar campos opcionais



Evitar Junções

- Junção
 - Operação para buscar dados de diversas tabelas associadas pela igualdade de campos
 - Exemplo:
 - Buscar os dados de um empregado e os dados de seu departamento (duas tabelas diferentes)



Evitar Junções

- Por que evitar junções?
 - SGBD relacional normalmente armazena os dados de uma linha contiguamente em disco
 - Com isso, todos dados de uma linha são trazidos para a memória em uma operação de acesso a disco
 - Junção envolve diversos acessos a disco
 - Preferível
 - Ter os dados necessários a uma consulta em uma única linha



Diminuir o Número de Chaves Primárias

- Para a implementação eficiente do controle da unicidade da chave primária, o SGBD usa normalmente uma estrutura de acesso auxiliar, um índice, para cada chave primária
 - Índices tendem a ocupar espaço considerável em disco
 - Inserção e remoção de entradas em um índice:
 - podem exigir diversos acessos a disco



Evitar Campos Opcionais

- Campo opcional = campo que podem assumir o valor VAZIO (NULL em SQL).
- SGBD relacional não desperdiça espaço pelo fato de campos de uma linha estarem vazios
 - Pois usam técnicas de compressão de dados e registros de tamanho variável no armazenamento interno de linhas
 - Além disso, há uma cláusula de SQL que especifica ao SGBD se o campo deve estar preenchido ou pode estar vazio



Evitar Campos Opcionais

- Campo opcional não tem influência na performance.
 - Porém, o controle de campo opcional pode complicar a programação
- Uma situação que pode gerar problemas é aquela na qual a obrigatoriedade ou não do preenchimento de um campo depende do valor de outros campos.
 - Neste caso, o controle da obrigatoriedade deve ser feito pelos programas que acessam o banco de dados, o que deve ser evitado



Regras Gerais de Tradução

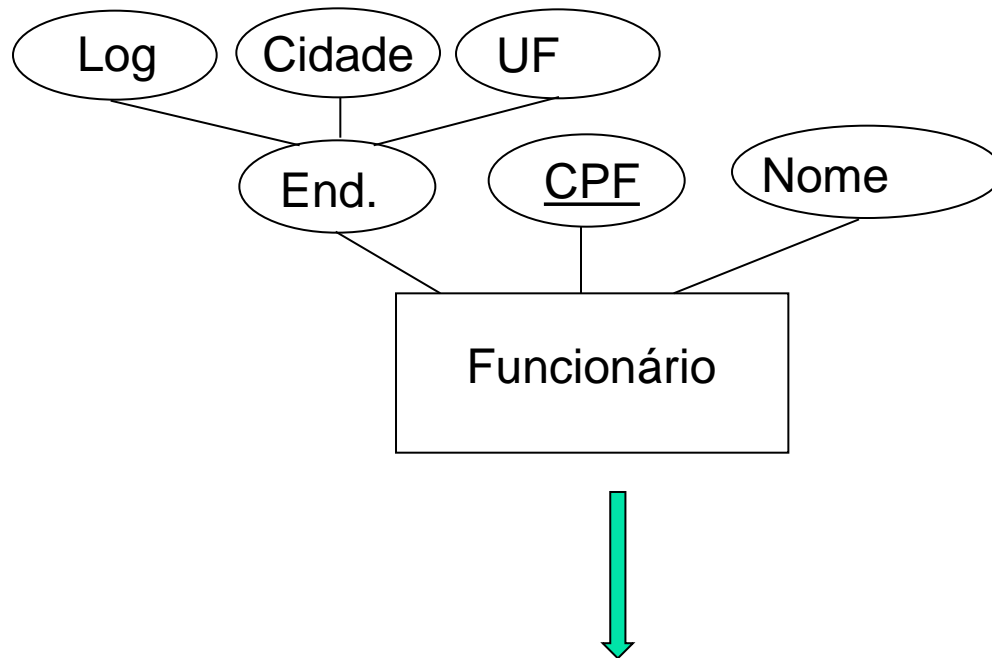
- Cada entidade é traduzida para uma tabela
- Cada atributo da entidade define uma coluna desta tabela
- Atributos identificadores da entidade correspondem a chave primária da tabela
- O algoritmo de mapeamento é composto
 - 7 etapas gerais
 - 2 para MER estendido



Mapeamento de Tipos de Entidade Regular

- Mapear todos os conjuntos de entidades fortes
- Para cada conjunto de entidade E no esquema ER cria-se uma relação R que inclui todos os atributos de E
 - Caso exista atributo composto, inclua todos os atributos elementares que compõem o atributo composto
 - O(s) atributo(s) identificador(es) de E passa(m) a ser chave(s) primária(s) para a relação R

Mapeamento de Tipos de Entidade Regular



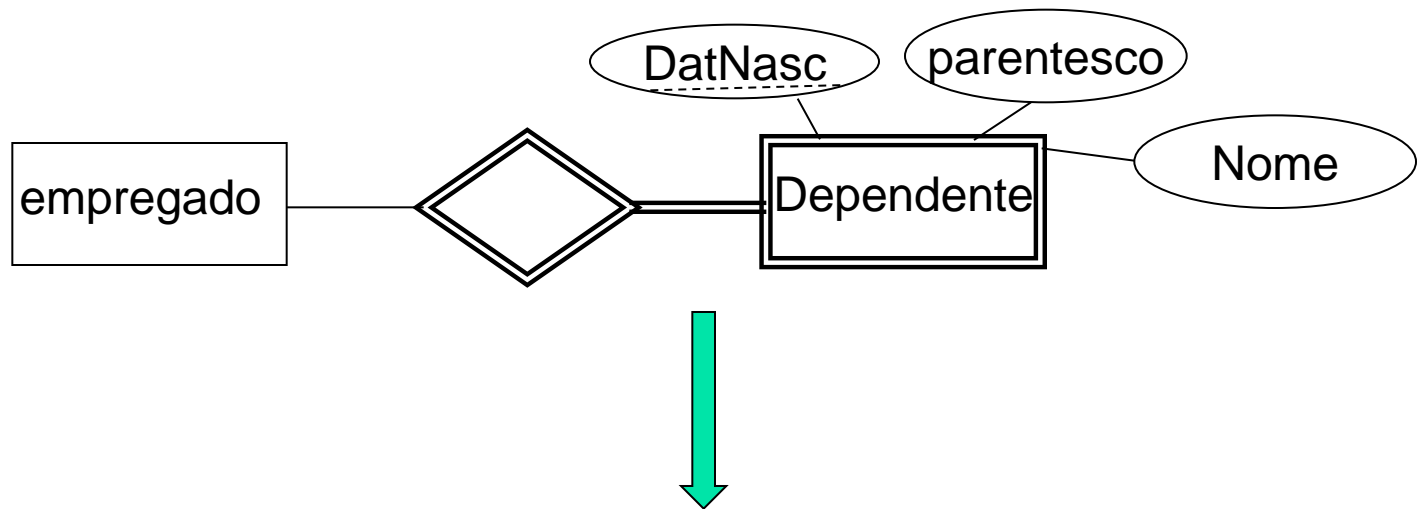
Funcionário(CPF, Nome, Logradouro, Cidade, UF)



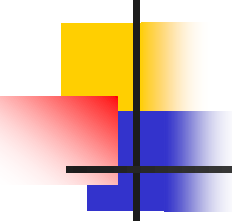
Mapeamento de Tipos de Entidade Fraca

- Para cada entidade fraca F no esquema ER cria-se uma relação R formada por todos os atributos de F mais os atributos que são chave das entidades envolvidas nos relacionamentos com F
- A chave de R é a chave de F concatenada com as chaves das entidades envolvidas com F

Mapeamento de Tipos de Entidade Fraca

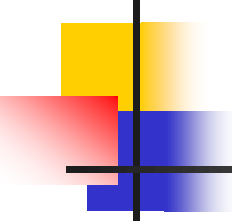


Dependente (CPF, DatNasc, nome, parentesco)
PK (CPF, DatNasc)
FK (CPF) referencia Empregado



Mapeamento dos Tipos de Relacionamento Binários (1:1)

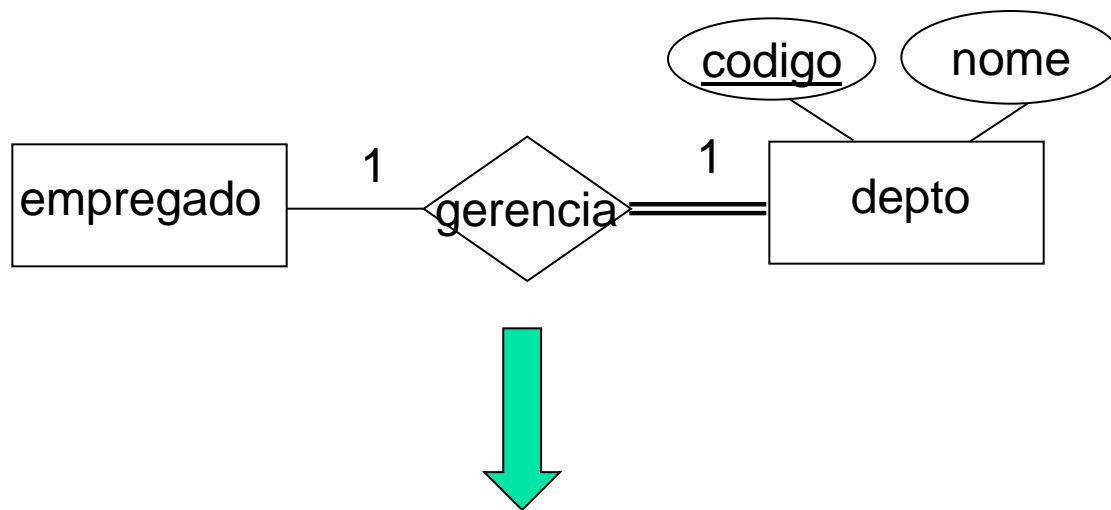
- O mapeamento de conjuntos de relacionamentos (CR) nem sempre resultam em uma nova relação, como é o caso dos conjuntos de relacionamento binário 1:1.
- Deve-se identificar os conjuntos de entidades S e T que participam do relacionamento.
 - Uma das entidades poderá ser escolhida.
 - Se uma entidade participar de forma total no relacionamento, esta deverá ser escolhida



Mapeamento dos Tipos de Relacionamento Binários (1:1)

- Deve-se identificar os conjuntos de entidades S e T que participam do relacionamento (cont.)
 - Acrescenta-se a entidade escolhida os atributos chaves primárias da outra entidade
 - Deve-se ressaltar que os atributos chave que foram acrescentados à entidade escolhida, são incluídos como atributos não chave nesta relação

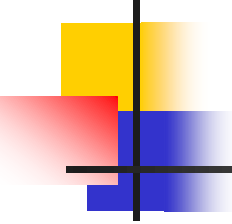
Mapeamento dos Tipos de Relacionamento Binários (1:1)



Depto (codigo, nome, cpf_ger)

PK (codigo)

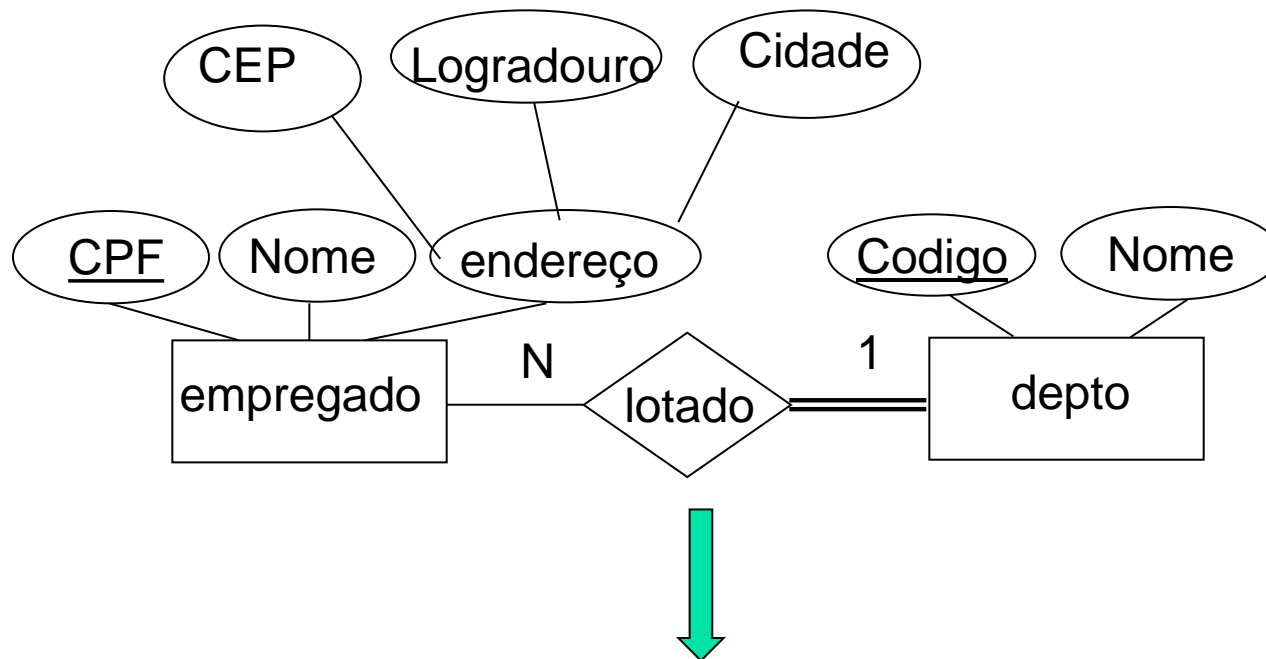
FK (cpf_ger) referencia Empregado



Mapeamento dos Tipos de Relacionamento Binários (1:N)

- Os conjuntos de relacionamento binário regular (não fraco) 1:N também não são representados como novas relações.
- Primeiro identifica-se o conjunto de entidade que participa da relação com cardinalidade N que será chamada de S e o outro conjunto de entidade chamada de T.
 - Os atributos chave da relação que mapeia o conjunto de entidade que participa com cardinalidade 1, representado por T, são também acrescentados no conjunto de entidade S como atributos não chave

Mapeamento dos Tipos de Relacionamento Binários (1:N)



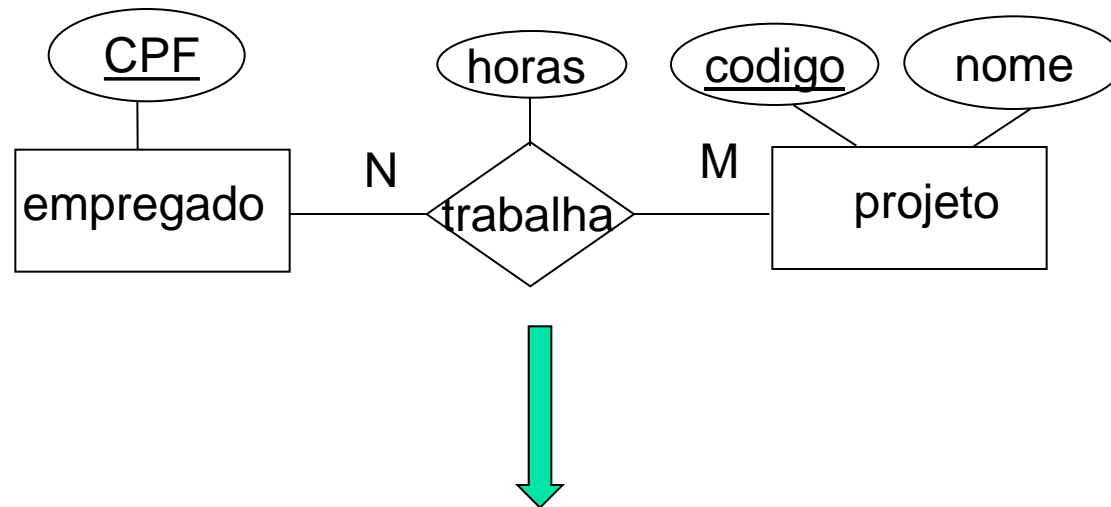
Empregado (CPF, nome, Logradouro, Cidade, CEP, Cod_depto)
PK (CPF)
FK (Cod_depto) referencia Depto



Mapeamento dos Tipos de Relacionamento Binários (M:N)

- Para cada relacionamento binário M:N cria-se uma nova relação S para representar R
 - Adicionar em S (como chave estrangeira) as chave primárias das duas relações participantes de R
 - A combinação acima formará a chave primária de S
 - Adicionar os atributos simples (mesmo de atributos compostos) de R em S (se houver)

Mapeamento dos Tipos de Relacionamento Binários (M:N)

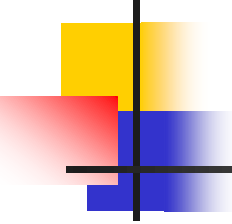


Trabalha (cpf_emp, cod_proj, horas)
PK (cpf_emp, cod_proj)
FK (cpf_emp) referencia Empregado
FK (cod_proj) referencia Projeto



Mapeamento de Atributos Multivalorados

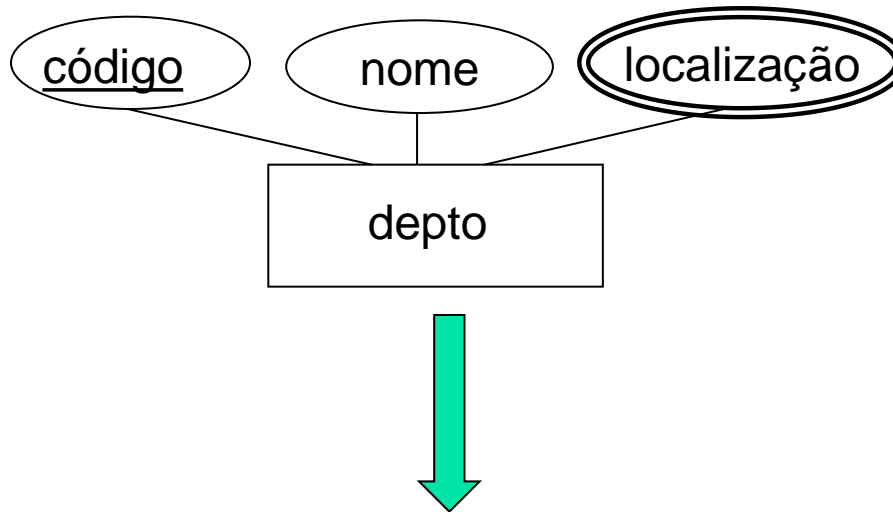
- Existem duas maneiras de mapear atributos multivalorados:
- A primeira maneira não leva-se em conta o conhecimento adicional sobre o atributo que está sendo mapeado.
 - Para cada atributo multivalorado cria-se uma nova relação que tem como chave os atributos chave da relação a qual pertencia juntamente com o atributo multivalorado tomado como um atributo monovalorado.



Mapeamento de Atributos Multivalorados

- A segunda forma leva em conta o conhecimento adicional sobre o atributo que está sendo mapeado.
 - Em alguns casos é possível determinar a quantidade de ocorrências de valores nos atributos.
 - Quando isso acontece e essa quantidade é pequena, pode-se instanciar essa quantidade de atributos como monovalorados na mesma relação que mapeia o conjunto de entidade ou conjunto de relacionamento ao qual o atributo multivalorado está associado.

Mapeamento de Atributos Multivalorados



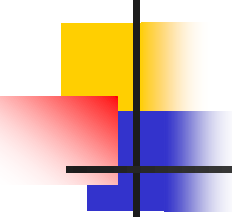
Local_Depto (cod_depto, localização)

PK (cod_depto, localização)

FK (cod_depto) referencia Depto

Ou

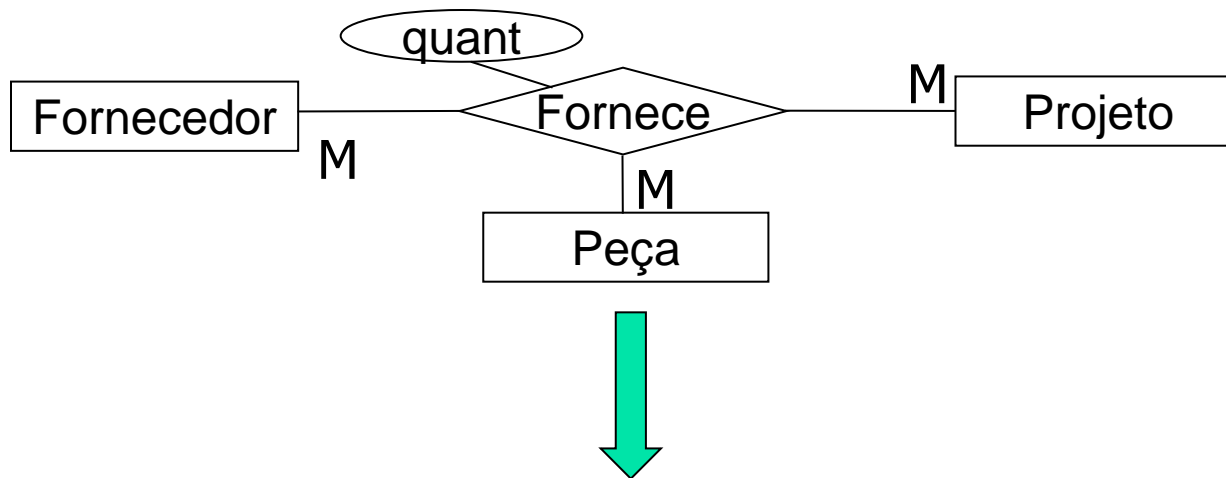
Depto (Codigo, nome, localizacao1, localizacao2)



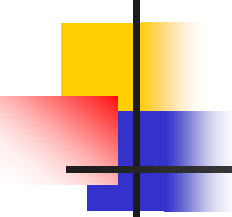
Mapeamento dos Tipos de Relacionamento N-ário ($n > 2$)

- Para conjuntos de relacionamentos n-ário, sempre considera-se que possuam cardinalidade vários:vários:vários ($n:n:n$)
 - Para cada conjunto de relacionamento será criada uma nova relação cujos atributos próprios são os do CR (se existirem)
 - A chave primária é formada pelos atributos chaves primárias das relações que mapeiam os conjuntos de entidades (CE's) envolvidos.
 - exceto se a cardinalidade de alguma entidade for "um"

Mapeamento dos Tipos de Relacionamento N-ário ($n > 2$)



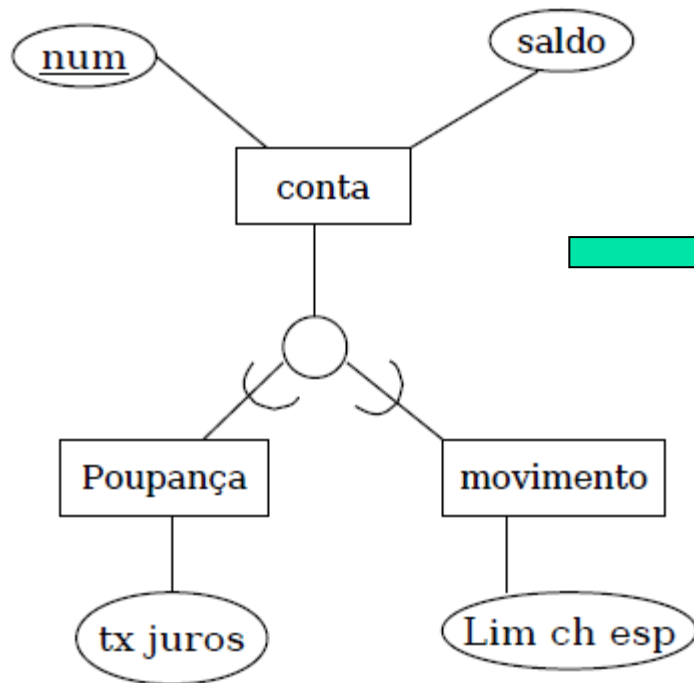
Fornece (cod_forn, cod_proj, cod_peça, quant)
PK (cod_forn, cod_proj, cod_peça)
FK (cod_forn) referencia Fornecedor
FK (cod_proj) referencia Projeto
FK (cod_peça) referencia Peça



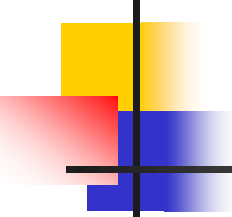
Mapeamento da Especialização ou Generalização

- Existem duas maneiras de transformar uma generalização/especialização em tabelas:
 - Criar uma tabela para o conjunto de entidades de nível superior.
 - Para cada conjunto de entidades de nível inferior, criar uma tabela que inclua uma coluna para cada um dos atributos daquele conjunto de entidades mais uma coluna para cada atributo da chave primária do conjunto de entidades de nível superior

Mapeamento da Especialização ou Generalização

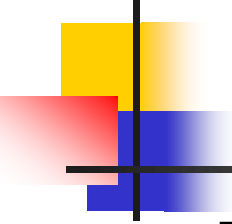


conta = (num, saldo)
poupança = (num, tx juros)
movimento = (num, lim ch esp)



Mapeamento da Especialização ou Generalização

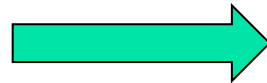
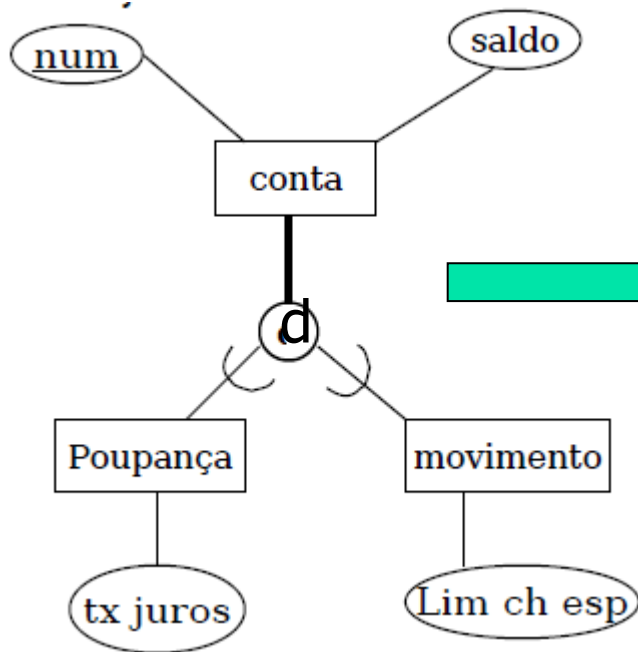
- Se a generalização é mutuamente exclusiva e total, isto é:
 - se nenhuma entidade é membro de mais de um conjunto de entidades de nível imediatamente inferior ao conjunto de entidades de nível superior
 - e se todas as entidades do conjunto de entidades de nível superior são membros também de um dos conjuntos de entidades de nível inferior,
- Então, uma outra representação alternativa é possível.



Mapeamento da Especialização ou Generalização

- Para cada conjunto de entidades de nível inferior, cria-se uma tabela que inclua uma coluna para cada um dos atributos do conjunto de entidades mais uma coluna para cada atributo de conjunto de entidades de nível superior

Mapeamento da Especialização ou Generalização



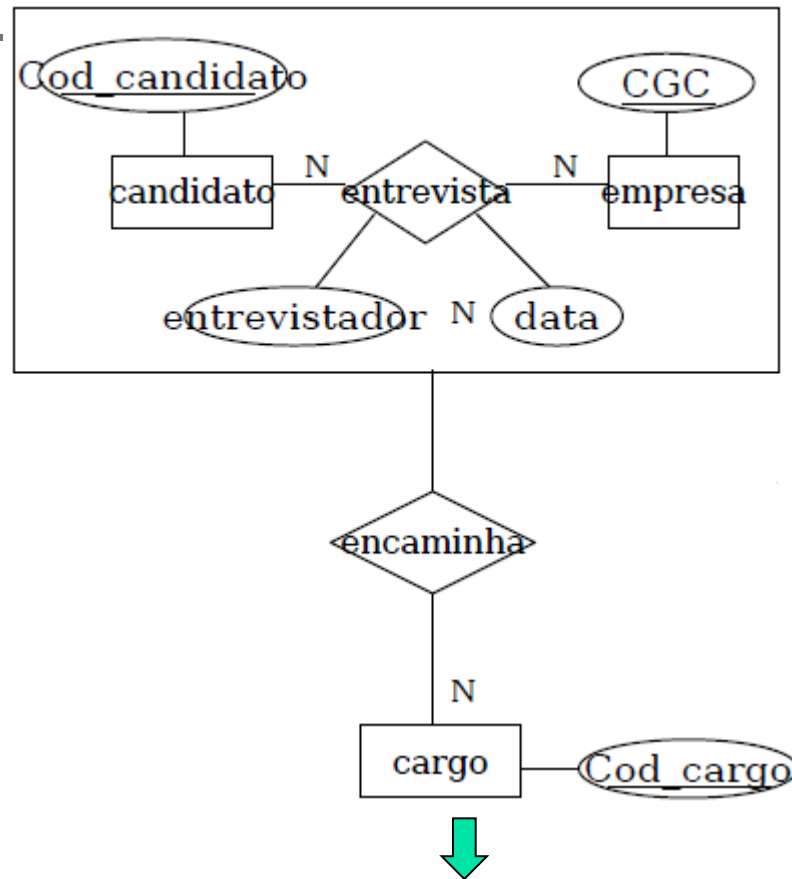
poupança = (num, saldo, tx juros)
movimento = (num, saldo, lim_ch_esp)



Mapeamento de Agregação

- A transformação de agregação em tabela é bastante direta.
- Considere o exemplo a seguir. A tabela para o relacionamento entrevista inclui uma coluna para cada atributo do relacionamento, uma para a chave primária de candidato e uma para empresa

Mapeamento de Agregação



entrevista = {cod candidato, CGC, entrevistador, data}
encamiha = {cod candidato, CGC, cod carga}



Resumo

MODELO ER	MODELO RELACIONAL
Tipo de entidade	Relação de <i>entidade</i>
Tipo de relacionamento 1:1 ou 1:N	Chave estrangeira (ou relação de <i>relacionamento</i>)
Tipo de relacionamento M:N	Relação de <i>relacionamento</i> e <i>duas</i> chaves estrangeiras
Tipo de relacionamento <i>n</i> -ário	Relação de <i>relacionamento</i> e <i>n</i> chaves estrangeiras
Atributo simples	Atributo
Atributo composto	Conjunto de atributos componentes simples
Atributo multivalorado	Relação e chave estrangeira
Conjunto de valores	Domínio
Atributo-chave	Chave primária (ou secundária)

Exercício

