

Transformações entre modelos

Parte 1

Profa. Carla Diacui Medeiros Berkenbrock

Departamento de Ciência da Computação
Centro de Ciências Tecnológicas - CCT
carla.berkenbrock@udesc.br

<https://www.udesc.br/colabora>

Transformações entre modelos

Aula de hoje...

- 1 Modelos de dados
 - Modelo Conceitual
 - Modelo Lógico
- 2 Transformação entre Modelos
 - Passos para a Transformação ER para relacional
- 3 Relacionamentos 1:1
- 4 Relacionamentos 1:n
- 5 Relacionamentos n:n

Modelos de Banco de Dados

Modelo de dados

Modelo para organização dos dados de um banco de dados

Descrição sobre os tipos de informações que estão armazenadas em um banco de dados

- *Exemplo: Informações sobre produtos (código, preço e descrição)*

O modelo pode variar de acordo com a intenção de quem está modelando os dados. Por exemplo:

- *Modelo utilizado para explicar a um usuário leigo sobre a organização de um BD*
- *Modelo usado por um técnico para otimizar o desempenho de um BD*

Modelos de Banco de Dados

Existem modelos para diferentes níveis de abstração de representação de dados

- **Modelo conceitual** – independente de implementação em SGBD
- **Modelo lógico** – dependente de SGBD
- Modelos físico
 - organização dos arquivos de dados em disco (organização seqüencial, uso de índices hashing ou B-trees, ...)
 - não são manipulados por usuários ou aplicações que acessam o BD

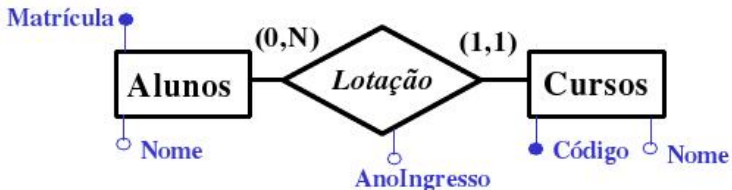
Modelo Conceitual

Modelo Conceitual =

Modelo de dados abstrato, que descreve a estrutura de um banco de dados de forma independente de um SGBD particular

Técnica mais difundida: **Modelo Entidade-Relacionamento (ER)**

- Diagrama ER

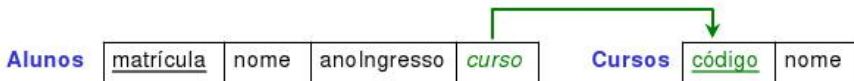


Modelo Lógico

Modelo Lógico =

Modelo de que representa a estrutura de dados de um banco de dados conforme vista pelo usuário do SGBD

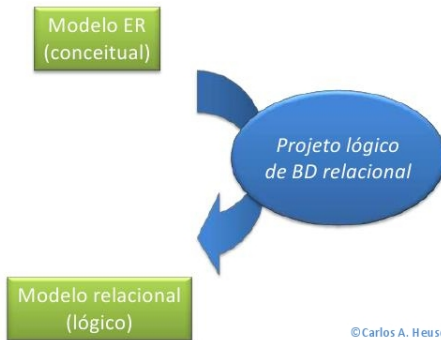
O modelo lógico mais amplamente utilizado: **modelo relacional**



Projeto Lógico de BD

Projeto Lógico =

Transformação de um modelo lógico que implementa, a nível de SGBD relacional, os dados representados abstratamente no modelo ER



Visão Geral do Projeto Lógico de BD

- Um determinado **modelo ER** pode ser implementado através de **diversos modelos relacionais**
- **Diferentes modelos** podem resultar em **diferentes desempenhos**
- Algumas **regras**, baseadas na experiência de alguns autores **podem ser aplicadas**
- Contudo, o **modelo relacional inicial** pode ser **refinado** até que ele atinja um modelo satisfatório

Transformação ER para relacional

Objetivos básicos:

- **Bom desempenho** de instruções de consulta e alteração do BD - **diminuir número de acesso a disco**
- **Simplificar o desenvolvimento** e a manutenção de aplicações

Até pouco tempo atrás era grande a preocupação em reduzir o espaço de armazenameto do BD...

Transformação ER para relacional

Princípios por traz das regras de tradução:

- **Evitar junções:** ter os dados necessários a uma consulta em uma única linha
 - dados de uma linha armazenados contiguamente em disco
 - quando possível, preferível ter dados necessários a consulta em uma única linha
- **Diminuir o número de chaves**
 - SGBD normalmente cria índice para chave primária, o que ocupa espaço em disco
- **Evitar campos opcionais**
 - SGBD usualmente não desperdiçam espaço quando o campo esta vazio
 - Problema ocorre quando a obrigatoriedade deve ser feita pelos programas

Transformação ER para relacional

Cliente (CodCliente, Nome, NomeContato, Endereço, Telefone)

ou:

Cliente (CodCliente, Nome, NomeContato)

ClienteEnder (CodCliente, Endereço, Telefone)

CodCliente referencia Cliente

©Carlos A. Heuser

Implementação inicial de entidades

- 1 Cada **entidade** é convertida em uma **tabela**
- 2 Cada **atributo** da entidade define uma **coluna** dessa tabela
- 3 Atributos **identificadores** da entidade correspondem a **chave primária** da tabela

Tradução inicial

Passos que seguem podem fundir tabelas

Entidades - Implementação inicial

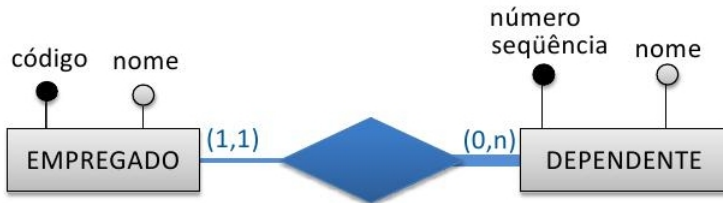


Entidades - Implementação inicial

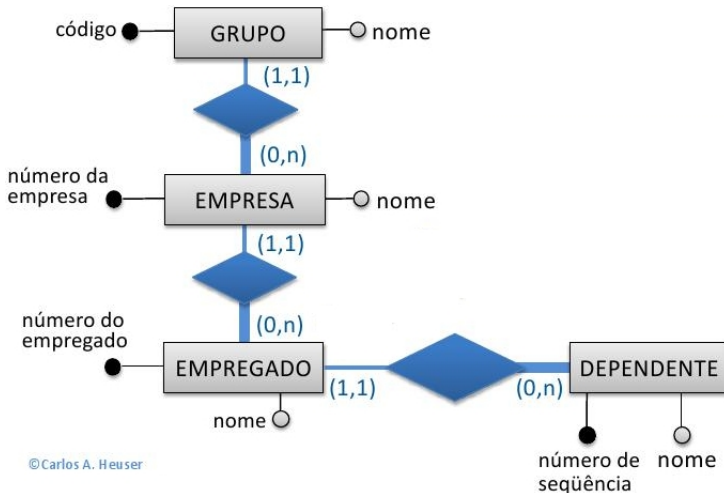
Nomes e atributos de tabelas:

- Não aconselhável transcrever nomes de atributos para nomes de colunas
 - Conveniente manter **nomes de colunas curtos**
 - Nomes de colunas em SGBD relacional **não deve conter brancos**
 - Nomes compostos de diversas palavras devem ser evitados
- Nomes de colunas não necessitam manter o nome da tabela
 - Exemplo: Preferível usar **nome** a usar **nomePess** ou **nomePessoa**
 - SQL permite **Pessoa.nome**
 - Exceção: **chave primária**
- Recomendação em padronizar abreviaturas (Exemplo: cod, id, no ou num)

Relacionamento - relacionamento identificador



Relacionamento - relacionamento identificador



Relacionamento - Tabela Própria

Relacionamento do tipo n:n



Relacionamento - Tabela Própria



Relacionamento - Adição de Colunas

Uma das entidades que participa do relacionamento tem cardinalidade máxima 1



Relacionamento - Fusão de Tabelas

Relacionamento do tipo 1:1



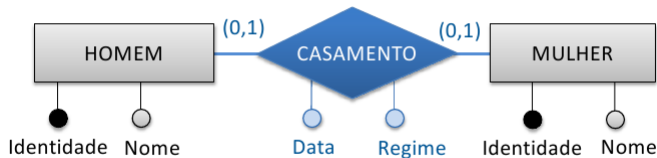
Relacionamentos 1:1

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
 (0,1) (0,1)	±	✓	✗
 (0,1) (1,1)	≠	±	✓
 (1,1) (1,1)	≠	≠	✓

Convenção:

- ✓ - alternativa preferida
- ± - pode ser usada - 1ª opção
- ≠ - pode ser usada - 2ª opção
- ✗ - não cabe

Relacionamentos 1:1 - Ambas entidades tem participação opcional



©Carlos A. Heuser

Relacionamentos 1:1 - Ambas entidades tem participação opcional

adição de colunas:

Mulher(identM, nome, **identH, data, regime**)

identH referencia Homem

Homem(identH, nome)

tabela propria:

Mulher(identM, nome)

Homem(identH, nome)

Casamento(identM, identH, data, regime)

identM referencia Mulher

identH referencia Homem

fusão de tabelas(qual a chave primária?)

Casamento(identM, nomeM, identH, nomeH, data, regime)

Relacionamentos 1:1 - Ambas entidades tem participação opcional

Solução por fusão de tabelas é **inviável**




- chave primária **artificial**

Solução por adição de colunas é **melhor**

- Menor número de **junções**
- Menos número de **chaves**

Solução por **tabela própria** aceitável

Relacionamentos 1:1

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

Convenção:

- ✓ - alternativa preferida
- ± - pode ser usada - 1ª opção
- ≠ - pode ser usada - 2ª opção
- ✗ - não cabe

Relacionamentos 1:1 - Uma entidade opcional, outra obrigatória



© Carlos A. Heuser

Relacionamentos 1:1 - Uma entidade opcional, outra obrigatória

fusão de tabelas

Correntista(codCorrent, nome, **codCartao**, **dtExp**)

adição de colunas:

Correntista(codCorrent, nome)

Cartao(codCartao, dtExp, **codCorrent**)
codCorrent referencia Correntista

tabela propria:

Correntista(codCorrent, nome)

Cartao(codCartao, dtExp)

CartaoCorrentista(codCartao,codCorrent)
codCorrent referencia Correntista
codCartao referencia Cartao

Relacionamentos 1:1 - Uma entidade opcional, outra obrigatória




Solução por fusão de tabelas é **melhor** em termos de número de junções e número de chaves

Solução por adição de colunas é **aceitável**

Solução por **tabela própria é a pior solução**

- maior número de junções
- maior número de índices
- nenhuma tem problema de campos opcionais

Relacionamentos 1:1 - Uma entidade opcional, outra obrigatória

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

Convenção:

- ✓ - alternativa preferida
- ± - pode ser usada - 1ª opção
- ≠ - pode ser usada - 2ª opção
- ✗ - não cabe

Relacionamentos 1:1 - Ambas as entidades são obrigatórias



©Carlos A. Heuser

Relacionamentos 1:1 - Ambas as entidades são obrigatórias




fusão de tabelas

Conferencia(codConf, nome, dtInstComOrg, EndComOrg)

Nenhuma das demais alternativas atende plenamente!

- Tabelas teriam mesma chave e relação 1 para 1 entre as suas linhas



Relacionamentos 1:1 - Ambas as entidades são obrigatórias

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

Convenção:

- ✓ - alternativa preferida
- ± - pode ser usada - 1ª opção
- ≠ - pode ser usada - 2ª opção
- ✗ - não cabe

Relacionamentos 1:n

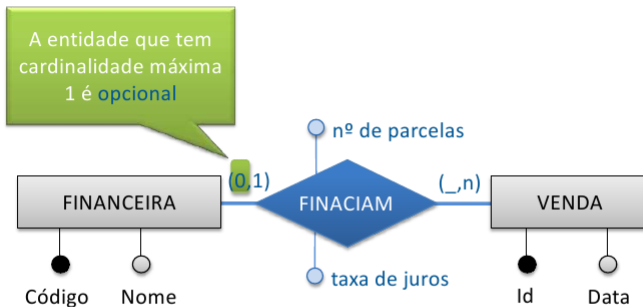
Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
 (0,1) (__,n)	±	✓	✗
 (1,1) (__,n)	✗	✓	✗

Convenção:

- ✓ - alternativa preferida
- ± - pode ser usada - 1ª opção
- ✗ - não cabe

©Carlos A. Heuser

Relacionamentos 1:n - caso 1



Relacionamentos 1:n - Discussão do caso 1

Solução por adição de colunas é **melhor** em termos de número de junções e número de chaves

Solução por tabela própria é **aceitável**

- é melhor em relação a campos opcionais
- perde em relação a junções e número de chaves

Relacionamentos 1:n - caso 2

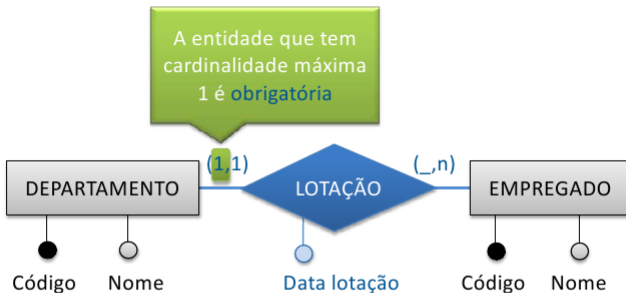
Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
$(0,1) \text{ } \diamond \text{ } (-,n)$	\pm	✓	✗
$(1,1) \text{ } \diamond \text{ } (-,n)$	✗	✓	✗

Convenção:

- ✓ - alternativa preferida
- \pm - pode ser usada - 1ª opção
- ✗ - não cabe

©Carlos A. Heuser

Relacionamentos 1:n - caso 2





Relacionamentos 1:n - Discussão do caso 2

Solução por **adição de colunas é melhor** do que tabela própria em termos de número de junções e número de chaves

Solução por fusão de tabelas **não se aplica**

- implicaria em redundância de dados


Relacionamentos 1:n

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição de coluna	Fusão de tabelas
<u>(0,1)</u>  <u>(_,n)</u>	±	✓	✗
<u>(1,1)</u>  <u>(_,n)</u>	✗	✓	✗

Convenção:

- ✓ - alternativa preferida
- ± - pode ser usada - 1ª opção
- ✗ - não cabe

Relacionamentos n:n

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

Convenção:

- ✓ - alternativa preferida
- ✗ - não cabe

© Carlos A. Heuser

Relacionamento - Tabela Própria



Atividade 1

Considere as seguintes alternativas de implementação de um BD relacional:

Alternativa 1

Aluno(codAl, nome, codCurso, endereco)

Alternativa 2

Aluno(codAl, nome, codCurso)

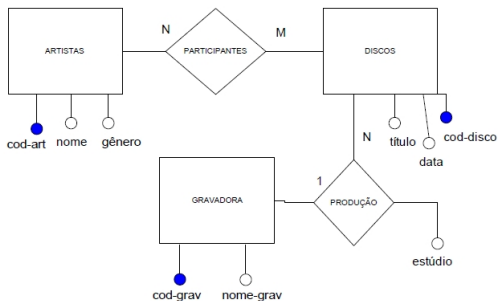
EnderecoAluno(codAl, endereco)

codAl referencia Aluno

Discuta qual das alternativas é preferível, considerando os princípios que baseiam as regras de tradução de diagramas ER

Atividade 2

(POSCOMP 2008) Considere o projeto lógico do banco de dados representado pelo modelo E-R abaixo.



Diga de que tabelas e campos ele resulta (obs: campos chave estão sublinhados):

Atividade 2

- A) ARTISTAS (cod-art, nome, gênero) PARTICIPANTES (cod-art, cod-disco)
DISCOS (cod-disco, título, data) PRODUÇÃO (cod-grav, cod-disco, estúdio)
GRAVADORA (cod-grav, nome-grav)
- B) ARTISTAS (cod-art, nome, gênero) PARTICIPANTES (cod-art, cod-disco)
DISCOS (cod-disco, título, data, cod-grav, estúdio) GRAVADORA (cod-grav,
nome-grav)
- C) ARTISTAS (cod-art, nome, gênero) PARTICIPANTES (cod-art, cod-disco)
DISCOS (cod-disco, título, data, cod-grav, nome-grav, estúdio)
- D) ARTISTAS (cod-art, nome, gênero) DISCOS (cod-disco, título, data,
cod-art) PRODUÇÃO (cod-grav, cod-disco, estúdio) GRAVADORA (cod-grav,
nome-grav)
- E) ARTISTAS (cod-art, nome, gênero) DISCOS (cod-disco, título, data,
cod-grav, estúdio, cod-art) GRAVADORA (cod-grav, nome-grav)

*“Educamos muito mais pelo que somos do que pelo falamos.
Devemos educar sempre e, se necessário, usar as palavras”
(A. Cury, em O código da inteligência)*

FIM

Profa. Carla Diacui Medeiros Berkenbrock -
carla.berkenbrock@udesc.br