

# Transformação entre Modelos

## Banco de Dados

Prof. Igor Avila Pereira  
[igor.pereira@riogrande.ifrs.edu.br](mailto:igor.pereira@riogrande.ifrs.edu.br)

Divisão de Computação  
Instituto Federal do Rio Grande do Sul - IFRS - Câmpus Rio Grande

## Agenda

### 1 Transformação entre Modelos

- Passo 1: Entidades Fortes
- Passo 2: Entidades Fracas
- Passo 3: Relacionamentos 1:1
- Passo 4: Relacionamentos 1:N
- Passo 5: Relacionamentos M:N
- Passo 6: Atributos multivalorados
- Passo 7: Relacionamentos N-ários

### 2 Domínios e valores vazios

- Tipos - PostgreSQL

## Agenda

- 1 Transformação entre Modelos
  - Passo 1: Entidades Fortes
  - Passo 2: Entidades Fracas
  - Passo 3: Relacionamentos 1:1
  - Passo 4: Relacionamentos 1:N
  - Passo 5: Relacionamentos M:N
  - Passo 6: Atributos multivalorados
  - Passo 7: Relacionamentos N-ários
- 2 Domínios e valores vazios

## Transformação entre Modelos

- A abordagem ER é voltada à modelagem de dados de forma independente do SGBD considerado.
  - É adequada para construção do modelo conceitual.
- Já a abordagem relacional modela os dados a nível de SGBD relacional.
  - Um modelo neste nível de abstração é chamado de modelo lógico.

### Função do Modelo Relacional

O modelo relacional é o modelo utilizado para mapear um esquema de modelo ER para um conjunto de tabelas inter-relacionadas (esquema de B.D relacional).

## Transformação entre Modelos

### Objetivos da aula

Nesta aula, vamos:

- Considerar a relação entre estes dois níveis de modelagem
- Realizar um correto mapeamento entre o diagrama E.R e o modelo relacional
  - Seguindo um conjunto de passos.

## Transformação entre Modelos

### Passo 1: Entidades fortes

- ❶ Para cada entidade forte do MER, crie uma tabela contendo todos os atributos simples da entidade como colunas da tabela.
  - O nome da tabela pode ser o próprio nome da entidade;
- ❷ Inclua apenas os atributos mais simples de cada um dos atributos compostos da entidade como colunas da tabela.
  - Não inclua o atributo composto;

## Transformação entre Modelos

### Passo 1: Entidades fortes

- ③ Escolha uma chave primária.
  - Se a chave da entidade possuir dois ou mais atributos, então a tabela terá duas ou mais colunas como chave primária.
  - **Devemos sublinhar as colunas da tabela que são chaves primárias**, a fim de diferenciá-las de colunas que não são chaves primárias;

#### Chave Primária

É uma coluna ou uma combinação de colunas cujos valores distinguem uma linha das demais dentro de uma tabela.

## Transformação entre Modelos

### Passo 2: Entidades fracas

- ❶ Para cada entidade fraca do MER, crie uma tabela contendo todos os atributos simples da entidade como colunas da tabela.
- ❷ Inclua apenas os atributos mais simples de cada um dos atributos compostos da entidade como colunas da tabela.
  - Não inclua o atributo composto;



## Transformação entre Modelos

### Passo 2: Entidades fracas

- ③ Escolha uma chave parcial para ser parte da chave primária da tabela.
- ④ Esta entidade fraca depende de uma entidade forte.
  - Então pegamos as chaves primárias da entidade forte e as exportamos como colunas da tabela da entidade fraca, surgindo o conceito de chaves estrangeiras.
  - Pode ser necessário que a chave estrangeira também faça parte da chave primária

#### Chave Estrangeira

É um atributo que estabelece a relação da tabela A com a chave primária da tabela B, permitindo uma relação entre A e B.

## Transformação entre Modelos

### Passo 3: Relacionamentos 1:1

- ❶ Para cada relacionamento binário do tipo 1:1 (um-para-um) no MER, identifique as duas entidades participantes.
  - Escolha a entidade que **não** tem **participação total** na relação (**Entidade Forte**).
  - Crie uma coluna na tabela desta entidade que será uma chave estrangeira.
  - Esta chave estrangeira será a chave primária da outra entidade, que tem participação total na relação;

#### Participação Total:

Quando todas as instâncias de uma entidade precisam estar associadas a alguma instância de outra entidade através de um relacionamento (**Entidade Fraca**)

## Transformação entre Modelos

### Passo 3: Relacionamentos 1:1

- ② Inclua todos os atributos simples (e demais atributos simples de cada atributo composto) da relação na tabela escolhida acima.

Sempre que existir relacionamentos 1:1, deve-se perguntar se realmente são duas entidades distintas ou se elas podem ser unidas.

## Tranformação entre Modelos

### Passo 3: Relacionamentos 1:1

Normalmente, ao checarmos a chave de ambas as entidades, chegamos facilmente à conclusão que as entidades devem ser unidas

Da mesma forma, deve-se perguntar se esse relacionamento sempre será um para um ou se existe a possibilidade de, amanhã, vir a ser um para muitos.

- Note que esse relacionamento é efetivamente raro.

## Transformação entre Modelos

### Passo 4: Relacionamentos 1:N

- ❶ Para cada relacionamento binário 1:N (um-para-muitos) no MER, identifique a entidade que representa a entidade participante do lado N da relação.
- ❷ Inclua como chave estrangeira na tabela desta entidade a chave primária da tabela que armazena os valores da entidade do lado com cardinalidade 1.
  - Isso ocorre porque cada registro desta tabela estará relacionado a no máximo um registro da tabela que contém o lado 1 da relação;

## Transformação entre Modelos

### Passo 4: Relacionamentos 1:N

- 3 Inclua todos os atributos simples (e demais atributos simples de cada atributo composto) do relacionamento na tabela do lado N da relação;

## Transformação entre Modelos

### Passo 5: Relacionamentos M:N

- 1 Para cada relacionamento binário N:N (muitos-para-muitos) no MER, **crie uma nova tabela para representar este relacionamento N:N**;
- 2 Inclua como chaves estrangeiras as chaves primárias das duas tabelas que armazenam os registros das duas entidades participantes.
  - **A combinação destas chaves estrangeiras será a chave primária desta nova tabela.**
- 3 Inclua todos os atributos simples (e demais atributos simples de cada atributo composto) da relação na tabela escolhida acima;

## Transformação entre Modelos

### Passo 6: Atributos multivalorados

- ❶ Para cada atributo multivalorado A de uma entidade, crie uma nova tabela.
  - Esta tabela conterá uma coluna correspondente ao atributo A (o atributo propriamente dito) e mais uma coluna que será uma chave estrangeira.
  - Esta chave estrangeira será proveniente da chave primária da tabela onde guardamos os registros da entidade.
- ❷ Caso o atributo multivalorado venha de uma relação, o procedimento é o mesmo.



## Transformação entre Modelos

### Passo 7: Relacionamentos N-ários

- 1 Para cada relacionamento n-ário (ternário, quaternário, etc.), onde  $n > 2$  crie uma nova tabela para representar este relacionamento;
- 2 Como chaves estrangeiras desta tabela, coloque todas as chaves primárias de cada entidade participante da relação;

## Transformação entre Modelos

### Passo 7: Relacionamentos N-ários

- 3 Inclua todos os atributos simples (e demais atributos simples de cada atributo composto) da relação na tabela escolhida acima;
- 4 A chave primária desta nova tabela será a combinação de chaves estrangeiras de cada entidade participante da relação.

## Agenda

- 1 Transformação entre Modelos
- 2 Domínios e valores vazios
  - Tipos - PostgreSQL

## Domínios e valores vazios

### Domínio

- Quando uma tabela do banco de dados é definida, para cada coluna da tabela, deve ser especificado um conjunto de valores (alfanumérico, numérico,...) que os campos da respectiva coluna podem assumir.
- Este conjunto de valores é chamado de **domínio da coluna** ou **domínio do campo**.

## Domínios e valores vazios

- Além disso, deve ser especificado se os campos da coluna podem estar vazios ("null" em inglês) ou não.
  - Estar vazio indica que o campo não recebeu nenhum valor de seu domínio.

As colunas nas quais não são admitidos valores vazios são chamadas de colunas obrigatórias.

## Domínios e valores vazios

As colunas nas quais podem aparecer campos vazios são chamadas de colunas opcionais.

- Normalmente, os SGBD relacional exigem que todas colunas que compõem a chave primária sejam obrigatórias. A mesma exigência não é feita para as demais chaves.

## Domínios e valores vazios

### Tipos - PostgreSQL

Nome	Descrição
boolean	booleano lógico (verdade/falso)
character varying [ (n) ]	cadeia de caracteres de comprimento variável
character [ (n) ]	cadeia de caracteres de comprimento fixo
date	data de calendário (ano, mês,dia)
double precision	número de ponto flutuante de precisão dupla
integer	inteiro de quatro bytes com sinal
money	quantia monetária
numeric [ (p, s) ]	numérico exato com precisão selecionável
real	número de ponto flutuante de precisão simples
serial	inteiro de quatro bytes com auto-incremento
text	cadeia de caracteres de comprimento variável
timestamp	data e hora

# Transformação entre Modelos

## Banco de Dados

Prof. Igor Avila Pereira  
[igor.pereira@riogrande.ifrs.edu.br](mailto:igor.pereira@riogrande.ifrs.edu.br)

Divisão de Computação  
Instituto Federal do Rio Grande do Sul - IFRS - Câmpus Rio Grande