

Aula 021

Professores:

Geraldo Xexéo
Geraldo Zimbrão

Conteúdo:

Conversão ER para Relacional

Roteiro

➡ Introdução

➡ Conversão do Modelo ER para Relacional

- ➡ Entidades e atributos

- ➡ Relacionamentos

 - ➡ 1 - 1

 - ➡ 1 - N

 - ➡ M - N

- ➡ Considerações

Introdução

- ➡ O Modelo Relacional é baseado em tabelas, atributos e relacionamentos
- ➡ Tanto o modelo ER quanto o modelo OO (classes e objetos) são semanticamente mais ricos
 - ➡ É possível convertê-los para um modelo relacional
 - ➡ Há vários padrões de conversão bem simples e diretos

Conversão de um Modelo ER para Relacional

- ➡ As construções do Modelo ER serão mapeadas para tabelas, atributos, relacionamentos e restrições
- ⇒ Tabelas e atributos serão utilizados para representar as entidades e atributos
 - ⇒ Chaves primárias e candidatas serão utilizadas para os atributos identificadores
 - ⇒ Chaves estrangeiras serão utilizadas para os relacionamentos
 - ⇒ Restrições serão utilizadas para representar as cardinalidades

Entidades, Tabelas e Atributos

➡ Cada Entidade será mapeada para uma tabela

- ➡ Escolhe-se atributos identificadores para a chave primária da tabela
- ➡ Possivelmente a chave primária pode conter mais de um atributo
- ➡ Os outros atributos serão atributos comuns
- ➡ É necessário especificar os tipos dos atributos

Aluno

CPF

NomeAluno

EnderecoAluno

NomePai

NomeMae

EscolaOrigem

EnderecoEscolaOrigem

Criação da Tabela

Create Table Aluno (
 CPF Char(11) not null primary key,
 NomeAluno varchar(80) not null,
 EnderecoAluno varchar(120),
 NomePai varchar(80),
 NomeMae varchar(80),
 EscolaOrigem varchar(80),
 EnderecoEscolaOrigem varchar(120);



Estabelecemos os tipos
refinando o modelo

— Detalhar os requisitos

Aluno

CPF

NomeAluno

EnderecoAluno

NomePai

NomeMae

EscolaOrigem

EnderecoEscolaOrigem

Relacionamentos 1 para 1



Os relacionamentos 1 para 1 podem ser mapeados de três formas principais

- ⇒ Cada tabela possui sua própria chave primária e uma das tabelas possui uma chave estrangeira para a outra
- ⇒ Uma tabela possui uma chave primária e a outra possui uma chave primária que também é chave estrangeira para a outra tabela
- ⇒ Uma tabela possui uma chave primária e a outra possui uma chave primária composta com uma chave estrangeira para a outra tabela

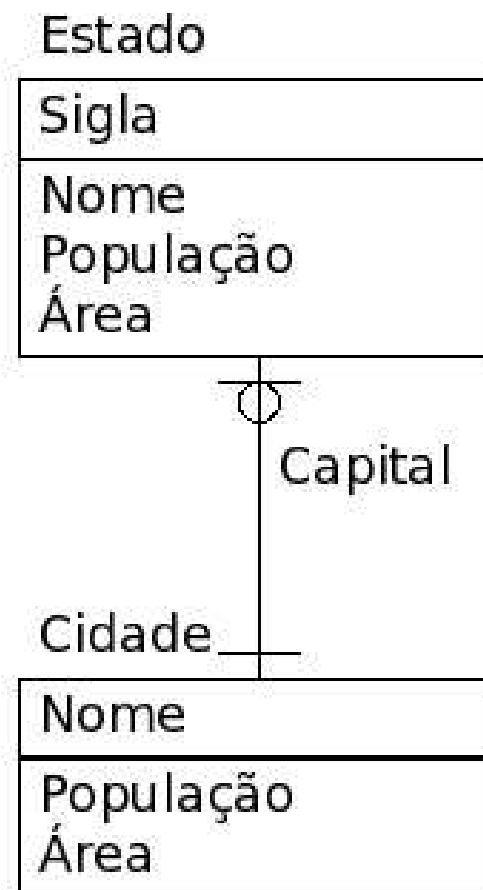
Relacionamentos 1 para 1 - Caso 1

- ➡ Cada tabela possui sua própria chave primária e uma das tabelas possui uma chave estrangeira para a outra
- ➡ Usada para representar relacionamentos $(0,1)(0,1)$ e $(0,1)(1,1)$ não identificadores
 - ➡ No caso $(0,1)(0,1)$ qualquer uma das tabelas pode ser escolhida para ter a chave estrangeira
 - ➡ No caso $(0,1)(1,1)$ a tabela onde o relacionamento for obrigatório é que deve conter a chave estrangeira
 - ➡ A cardinalidade é garantida pelo uso das cláusulas ***not null*** e ***unique***

Exemplo

```
Create Table Cidade (
  Nome varchar(80) not null primary key,
  Populacao integer,
  Area integer
);
```

```
Create Table Estado (
  Sigla char(2) not null primary key,
  Nome varchar(80) not null,
  Populacao integer,
  Area integer,
  Capital varchar(80) not null unique
  references Cidade
);
```



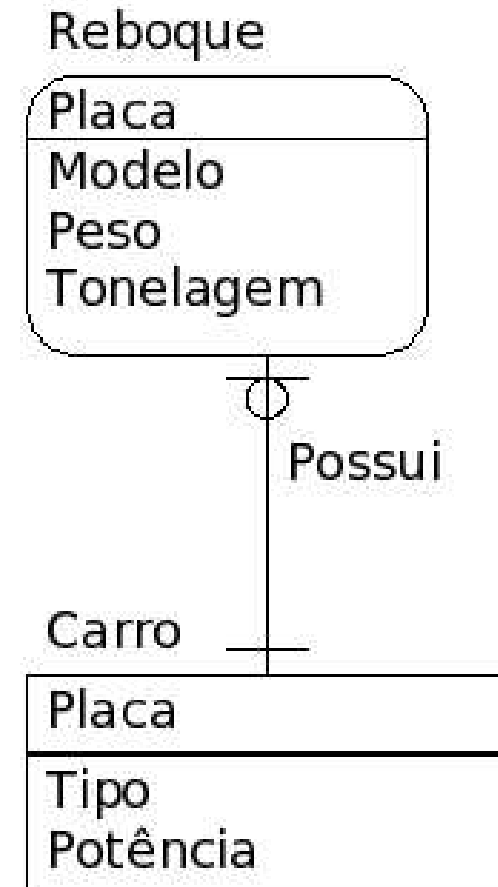
Relacionamentos 1 para 1 - Caso 2

- ➡ Uma tabela possui uma chave primária e a outra possui uma chave primária que também é chave estrangeira para a outra tabela
- ➡ Usada para representar relacionamentos (0,1)(1,1) identificadores
 - ➡ A chave estrangeira deve ficar na tabela que é identificada pelo relacionamento

Exemplo

```
Create Table Cidade (  
  Nome varchar(80) not null primary key,  
  Populacao integer,  
  Area integer  
);
```

```
Create Table Reboque (  
  Placa varchar(7) not null primary key  
  references Carro,  
  Modelo varchar(80) not null,  
  Peso integer,  
  Tonelagem integer  
);
```



Relacionamentos 1 para 1 - Caso 3

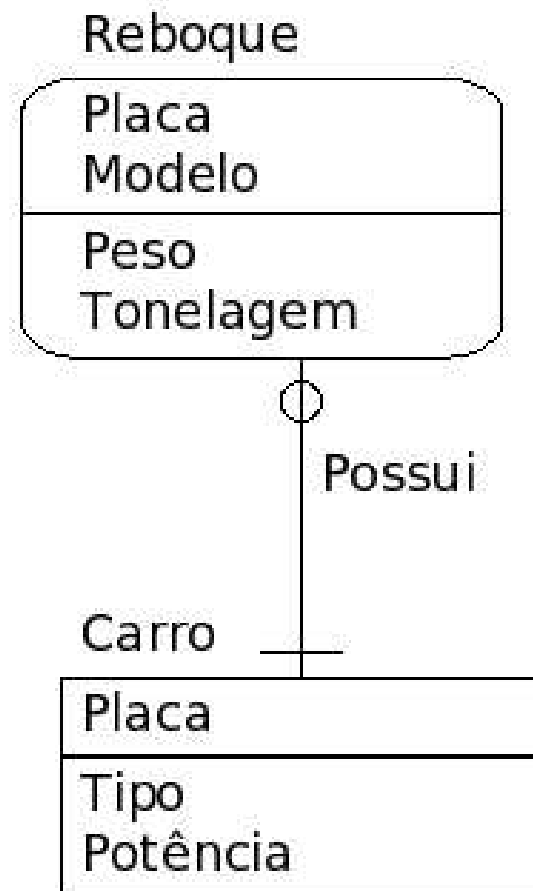
➡ Uma tabela possui uma chave primária e a outra possui uma chave primária composta com uma chave estrangeira para a outra tabela

- ➡ Usada para representar relacionamentos $(0,1)(1,1)$ identificadores
- ➡ A chave composta deve ficar na tabela que é identificada pelo relacionamento
- ➡ Pouco usado pois pode violar a 2FN

Exemplo

```
Create Table Carro (
  Placa varchar(7) not null primary key,
  Tipo varchar(40),
  Potencia integer
);
```

```
Create Table Reboque (
  Placa varchar(7) not null unique
  references Carro,
  Modelo varchar(80) not null,
  Peso integer,
  Tonelagem integer,
  Primary key( Placa, Modelo )
);
```



Relacionamentos 1 para N



Os relacionamentos 1 para N podem ser mapeados de duas formas principais

- ⇒ Cada tabela possui sua própria chave primária e a tabela da parte N do relacionamento possui uma chave estrangeira para a outra
- ⇒ A tabela da parte 1 do relacionamento possui uma chave primária e a tabela da parte N possui uma chave primária composta com uma chave estrangeira para a outra tabela

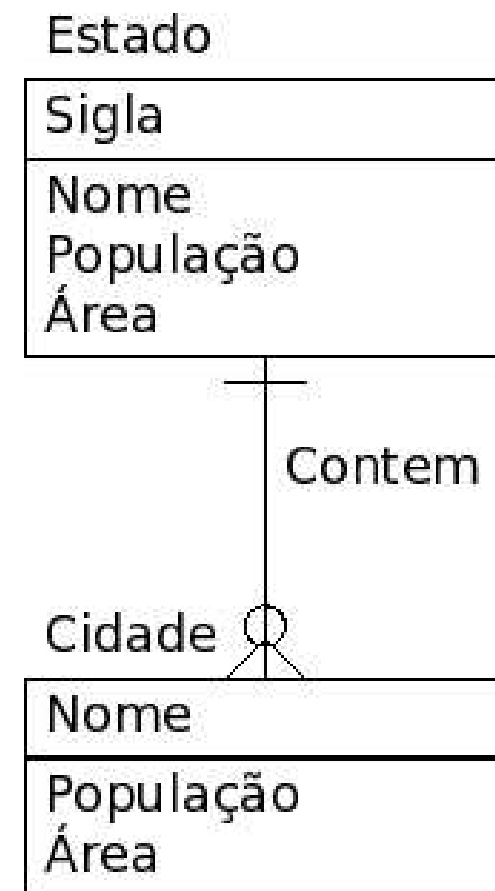
Relacionamentos 1 para N - Caso 1

- ➡ Cada tabela possui sua própria chave primária e a tabela da parte N do relacionamento possui uma chave estrangeira para a outra
- Usada para representar relacionamentos $(0,1)(0,N)$ e $(0,1)(1,N)$ não identificadores
 - Acrescentando uma restrição de integridade entre a chave primária da tabela da parte 1 do relacionamento e a tabela N pode-se modelar relacionamentos $(1,1)(0,N)$
 - Nem todos os SGBDs permitem isso

Exemplo

```
Create Table Estado (
  Sigla char(2) not null primary key,
  Nome varchar(80) not null,
  Populacao integer,
  Area integer
);
```

```
Create Table Cidade (
  Nome varchar(80) not null primary key,
  Populacao integer,
  Area integer,
  SiglaEstado char(2) not null
  references Estado
);
```



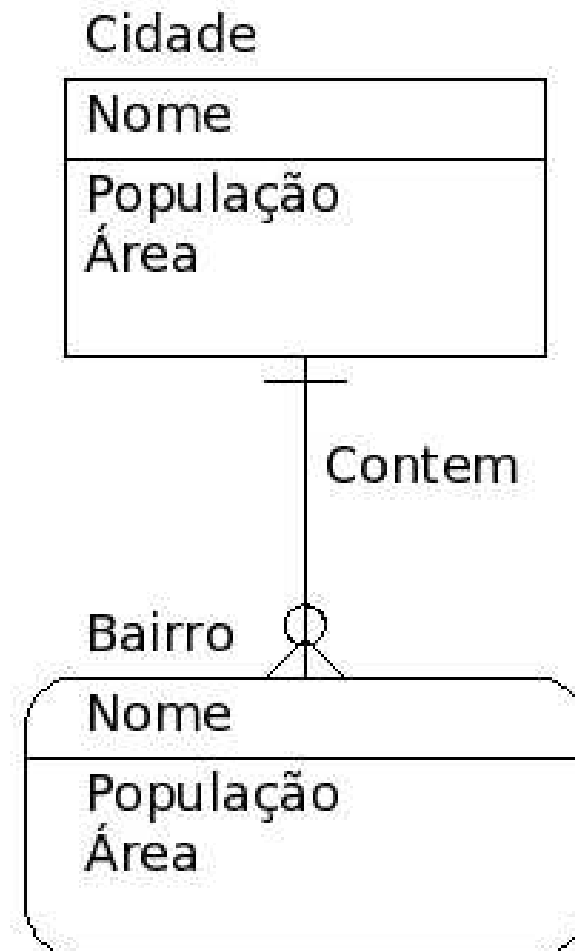
Relacionamentos 1 para N - Caso 2

- ➡ A tabela da parte 1 do relacionamento possui uma chave primária e a tabela da parte N possui uma chave primária composta com uma chave estrangeira para a outra tabela
- ▢ Usada para representar relacionamentos $(0,1)(1,N)$ identificadores

Exemplo

```
Create Table Cidade (  
  Nome varchar(80) not null primary key,  
  Populacao integer,  
  Area integer  
);
```

```
Create Table Bairro (  
  Nome varchar(80) not null,  
  Populacao integer,  
  Area integer,  
  Cidade varchar(80) not null  
    references Cidade,  
  Primary key( Nome, Cidade )  
);
```



Relacionamentos N para M

➡ Os relacionamentos N para M serão mapeados da seguinte forma

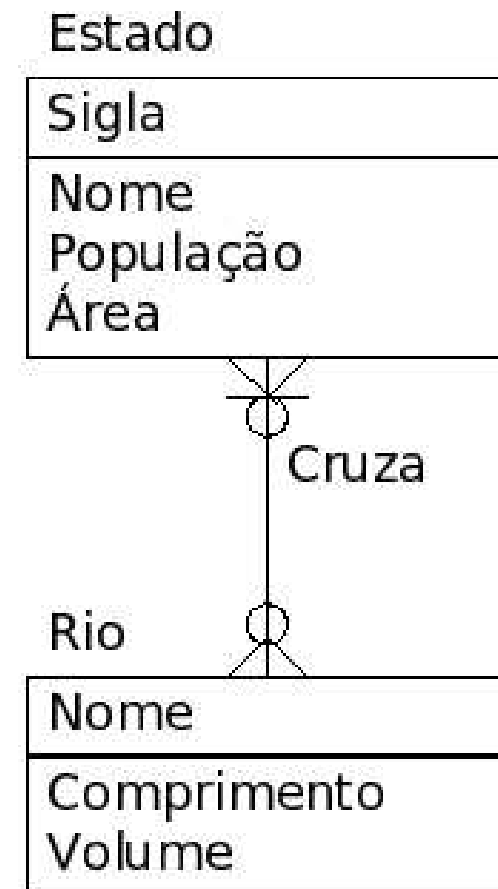
- ➡ Cada tabela possui sua própria chave primária
- ➡ Uma tabela é criada para representar o relacionamento, contendo uma chave composta de chaves estrangeiras para as duas tabelas do relacionamento
- ➡ Usado para representar relacionamentos (0,N)(0,M)

Exemplo

```
Create Table Estado (
  Sigla char(2) not null primary key,
  Nome varchar(80) not null,
  Populacao integer,
  Area integer
);
```

```
Create Table Rio (
  Nome varchar(80) not null primary key,
  Comprimento integer,
  Volume integer
);
```

```
Create Table RioEstado (
  Estado char(2) not null references Estado,
  Rio varchar(80) not null references Rio,
  Primary key( Estado, Rio )
);
```



Considerações

➡ Normalmente não se modela relacionamentos onde ambas as partes sejam obrigatórias:

⇒ $(1,1)(1,1)$, $(1,1)(1,N)$ e $(1,N)(1,M)$

⇒ Dificultam a inserção de dados

⇒ Em geral $(1,1)(1,1)$ é um erro de modelagem

➡ Algumas restrições devem ser implementadas por procedimentos (*triggers*)

⇒ Cardinalidades específicas, por exemplo:

⇒ $(0,1)(0,3)$

⇒ $(0,N)(1,M)$