

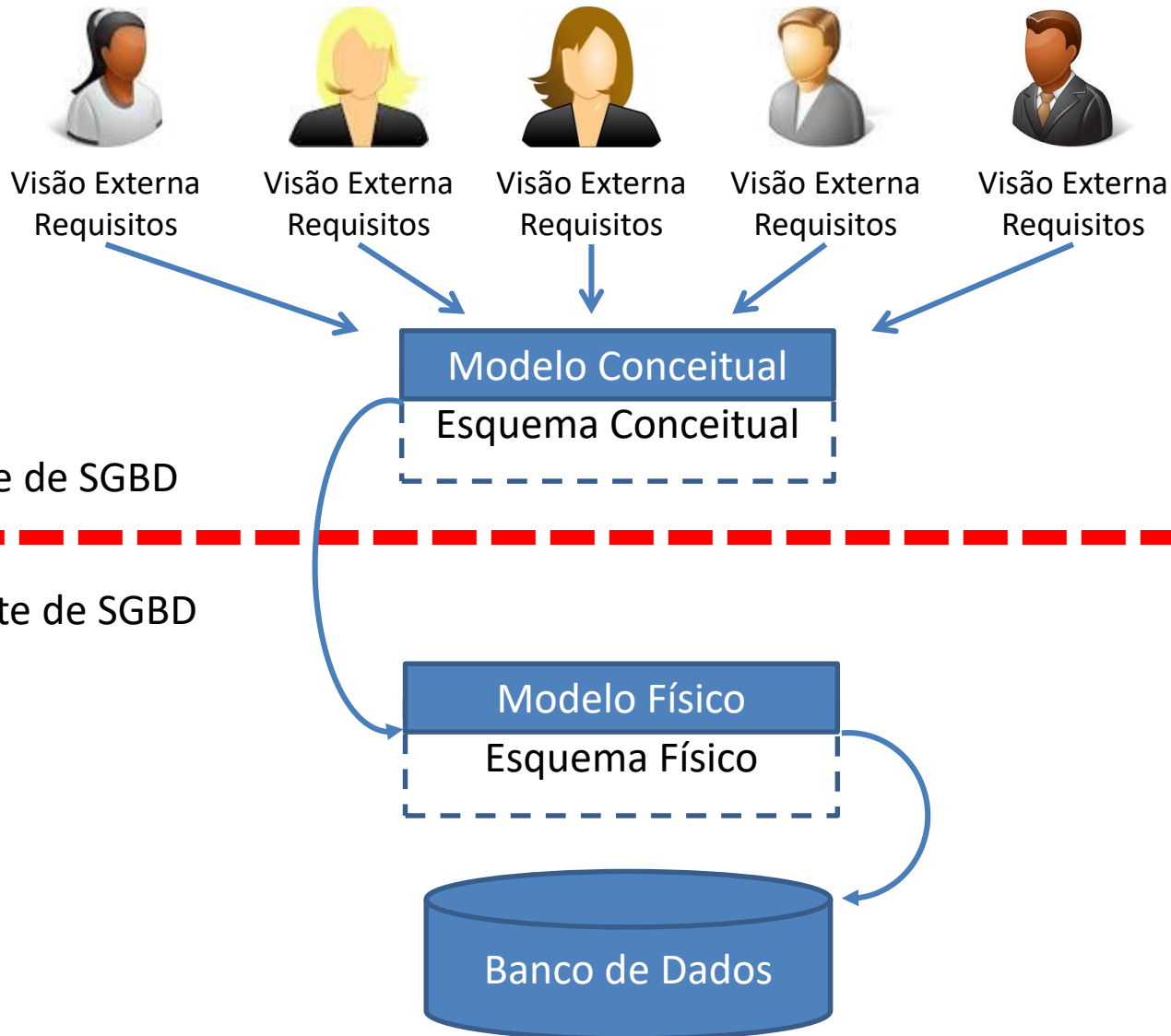


Banco de Dados

03 – Relacionamentos

Prof. André Ulisses
andre.ulisses@edu.sc.senai.br

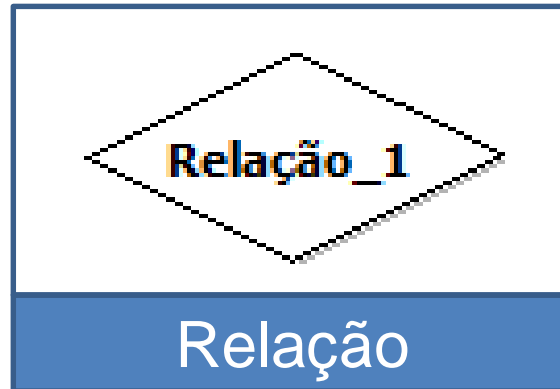
Projeto de Banco de Dados



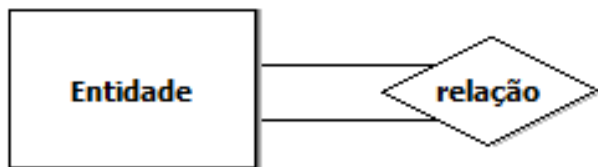
Um modelo conceitual é uma descrição do banco de dados de forma independente de implementação em um SGBD. O modelo conceitual registra que dados podem aparecer no banco de dados, mas não registra como estes dados estão armazenados a nível de SGBD.

Registra **QUAIS** dados podem aparecer no banco, mas não registra **COMO** estes dados estão armazenados.

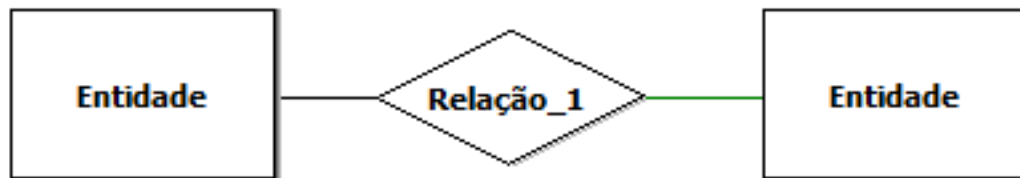
Conceitos e Símbolos



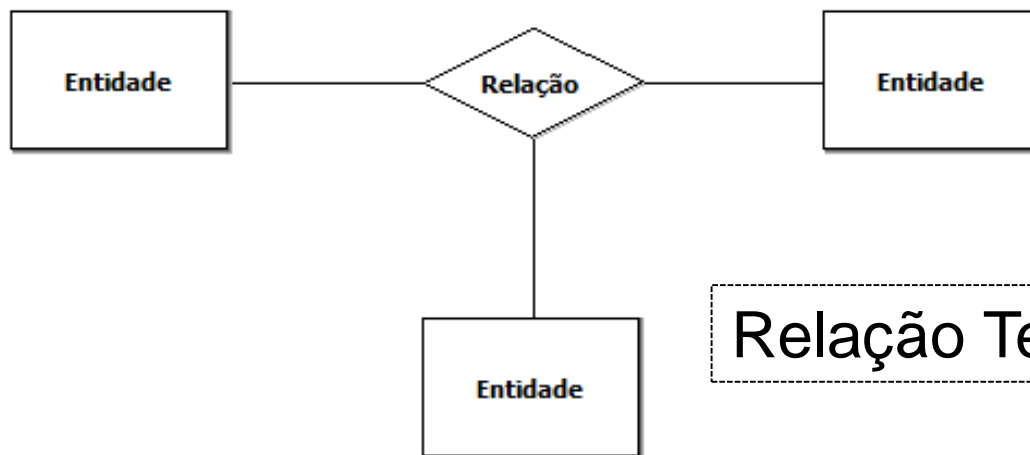
Relação Tipos



Auto-Relação



Relação Binária



Relação Ternária

Relação



Além de especificar os objetos sobre os quais deseja-se manter informações, uma das propriedades sobre as quais pode ser desejável manter informações é a associação entre objetos.

Cardinalidade

Para fins de projeto de banco de dados, uma propriedade importante de um relacionamento é a de quantas ocorrências de uma entidade podem estar associadas a uma determinada ocorrência através do relacionamento.

Esta propriedade é chamada de cardinalidade de uma entidade em um relacionamento.

Há duas cardinalidades a considerar: a cardinalidade máxima e a cardinalidade mínima.

Cardinalidade Mínima

Uma outra informação que pode ser representada por um modelo ER é o número mínimo de ocorrências de entidade que são associadas a uma ocorrência de uma entidade através de um relacionamento.

Podemos classificar os relacionamentos em:



Mínima 0

Associação Opcional



Mínima 1

Associação Obrigatória

Cardinalidade Máxima

A cardinalidade máxima pode ser usada para classificar relacionamentos binários. Um relacionamento binário é aquele cujas ocorrências envolvem duas entidades, como todos vistos até aqui.

Podemos classificar os relacionamentos em:



1:n (um-para-muitos)

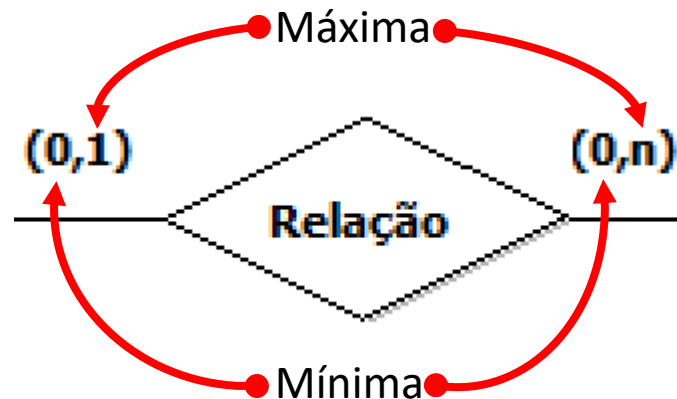
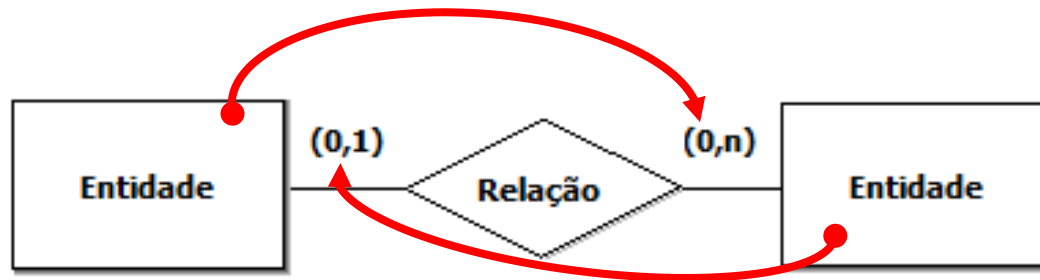


1:1 (um-para-um).



n:n (muitos-para-muitos)

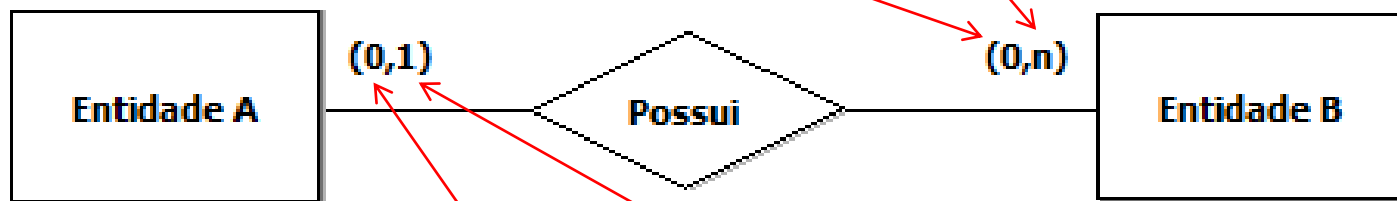
Cardinalidade Representação



Cardinalidade Representação



A entidade **A** Possui no mínimo **0** e no máximo **N** registros na Entidade **B**



A entidade **B** Possui no mínimo **0** e no máximo **1** registros na Entidade **A**



O modelo lógico é o resultado ou produto da conversão de um modelo conceitual para um determinado tipo de banco de dados, ou seja:

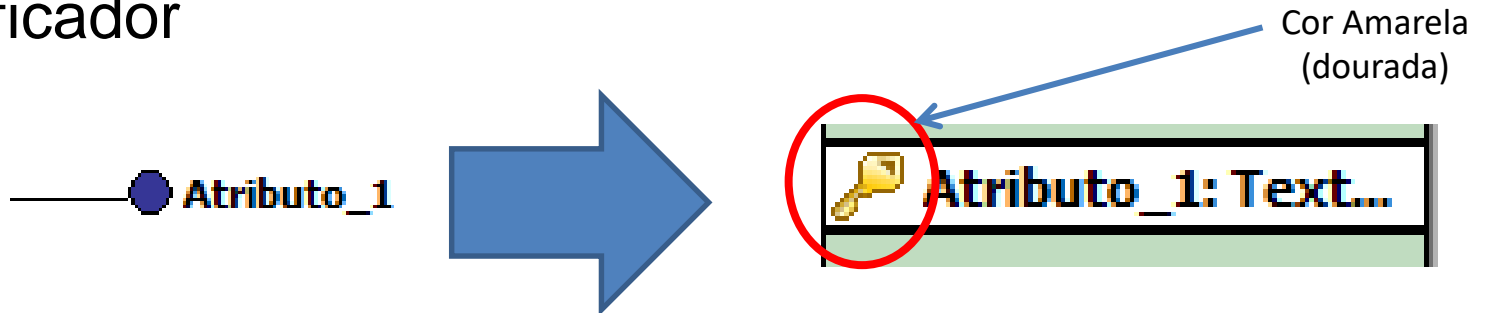
“Um modelo lógico é uma descrição de um banco de dados no nível de abstração visto pelo usuário do sistema gerenciador de banco de dados”

O conceito básico para estabelecer relações entre linhas de tabelas de um banco de dados relacional é o da chave.

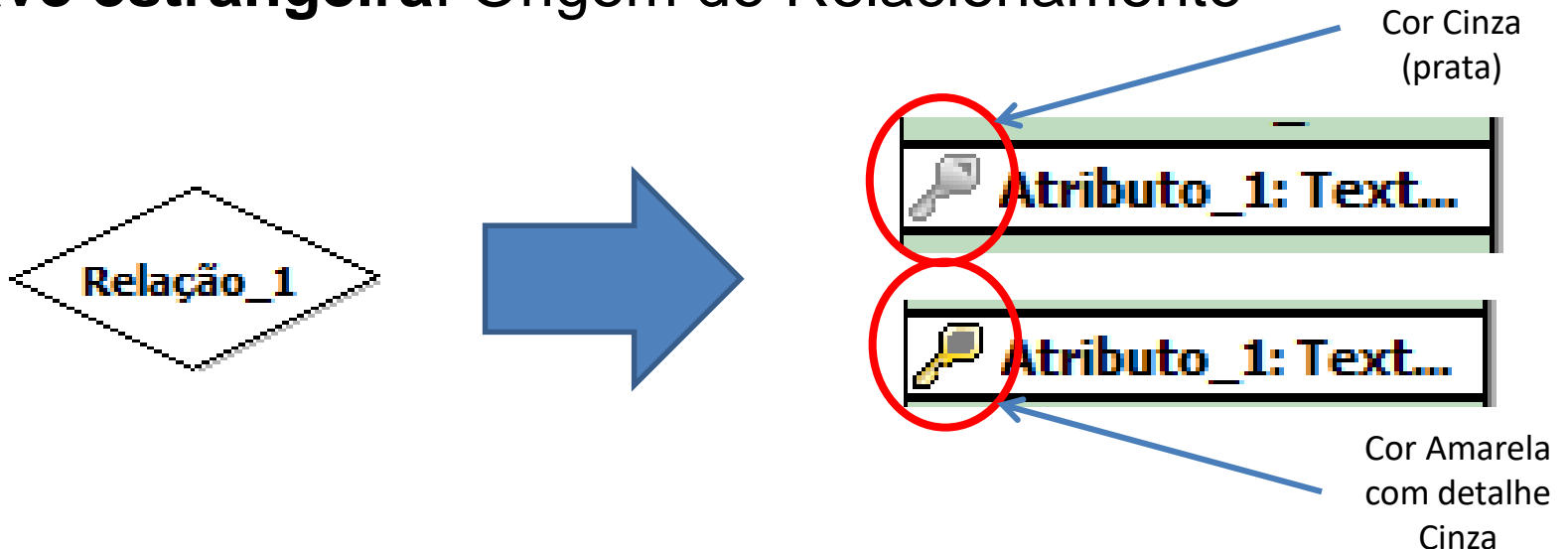
- **Chave primária Simples / Compostas** : apenas uma coluna ou a combinação de duas ou mais colunas
- **Chave estrangeira**: A existência de uma chave estrangeira impõe restrições que devem ser garantidas em diversas situações
- **Chave alternativa**: Em alguns casos, os valores de uma coluna não pode ser repetidos, mesmo essa coluna não fazendo parte de uma chave primária, assim temos as chaves alternativas.

CONCEITO DE CHAVE

- **Chave primária Simples / Compostas:** Origem do Identificador

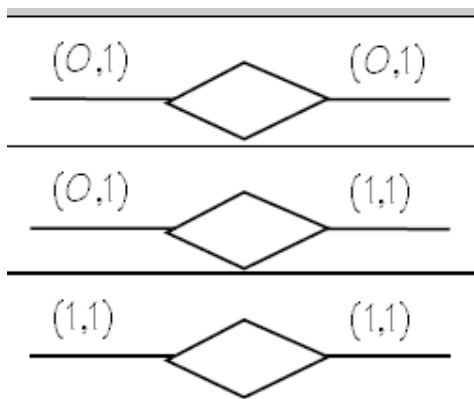


- **Chave estrangeira:** Origem do Relacionamento

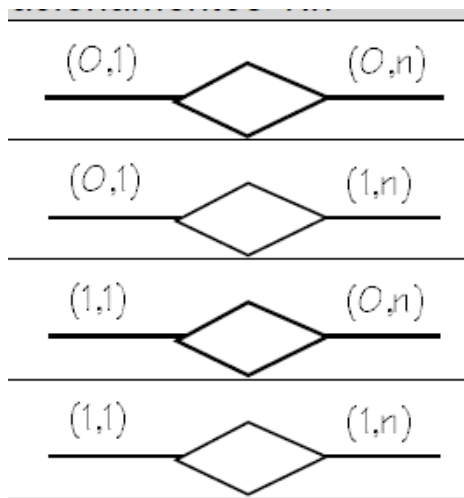


A transformação é em função da cardinalidade máxima usada para classificar relacionamentos binários.

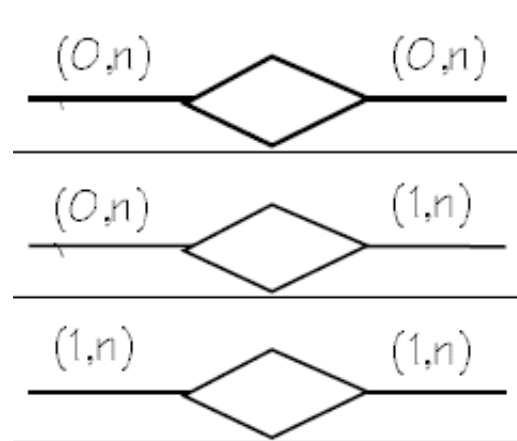
1:1 (um-para-um)



1:n (um-para-muitos)



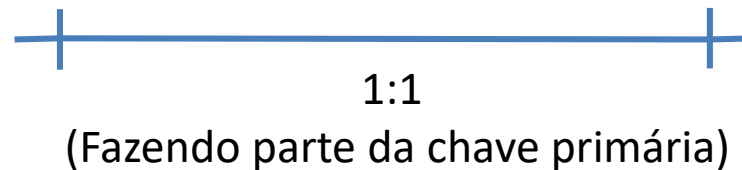
n:n (muitos-para-muitos)



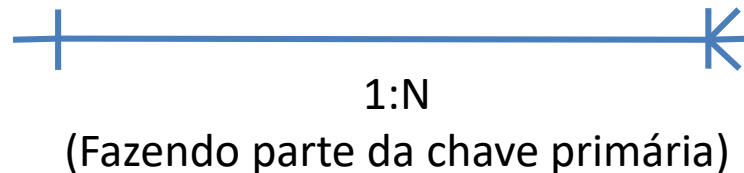
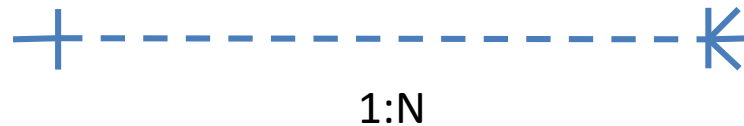
O produto da transformação do relacionamento na modelagem lógica está dividido em :

- **Tabela própria:** O relacionamento vira uma nova entidade (tabela) recebendo o identificador das duas tabelas que fazem parte do relacionamento, tornando-se chave primária;
- **Adição de Coluna:** A entidade de maior cardinalidade máxima recebe o identificador da entidade de menor cardinalidade máxima;
- **Fusão de Tabela:** As duas entidades se tornam apenas uma e some o relacionamento;

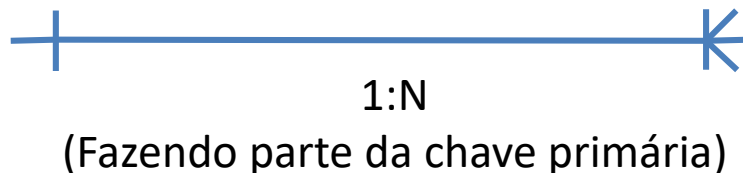
Transformação - Relacionamentos 1:1 (conceitual) (Adição de Coluna)



Transformação - Relacionamentos 1:n (conceitual) (Adição de Coluna)



Transformação - Relacionamentos n:n (conceitual) (adição de coluna com a tabela criada)



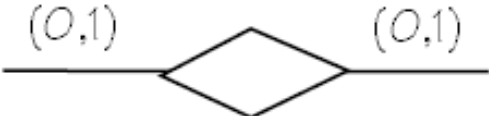
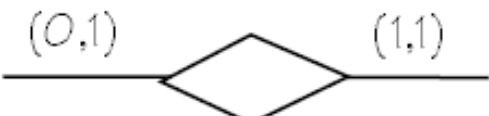
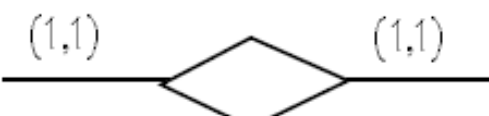
Chaves Estrangeira

```
CREATE TABLE nome_da_tabela (  
    nome_campo tipo_campo ,  
  
    FOREIGN KEY (nome_campo)  
    REFERENCES nome_tabela_relacionada (nome_campo_relacionado)  
);
```

Exemplo:

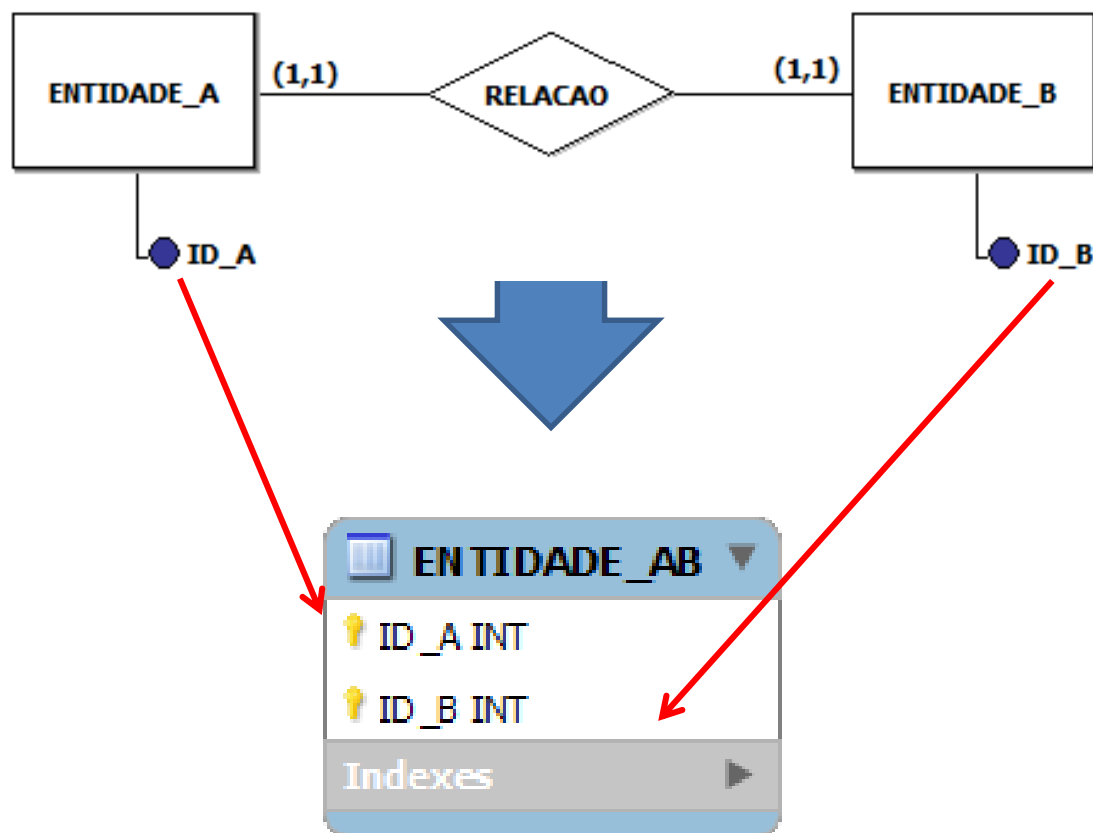
```
CREATE TABLE funcionario (  
    departamento int,  
  
    FOREIGN KEY (departamento) REFERENCES departamento  
    (codigo)  
);
```

Transformação - Relacionamentos 1:1

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição coluna	Fusão tabelas
Relacionamentos 1:1			
	±	✓	×
	×	±	✓
	×	±	✓

✓ Alternativa preferida ± Pode ser usada × Não usar

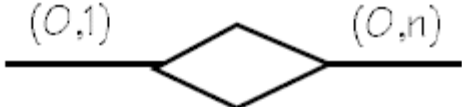
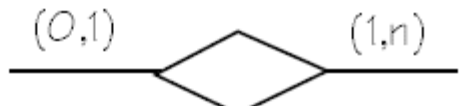
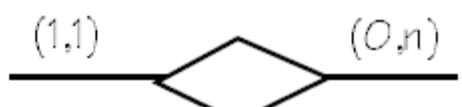
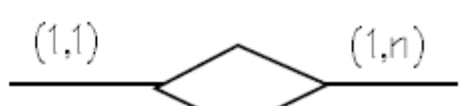
Transformação - Relacionamentos 1:1 (Fusão de Tabela)



SQL – DEFINIÇÃO DE DADOS

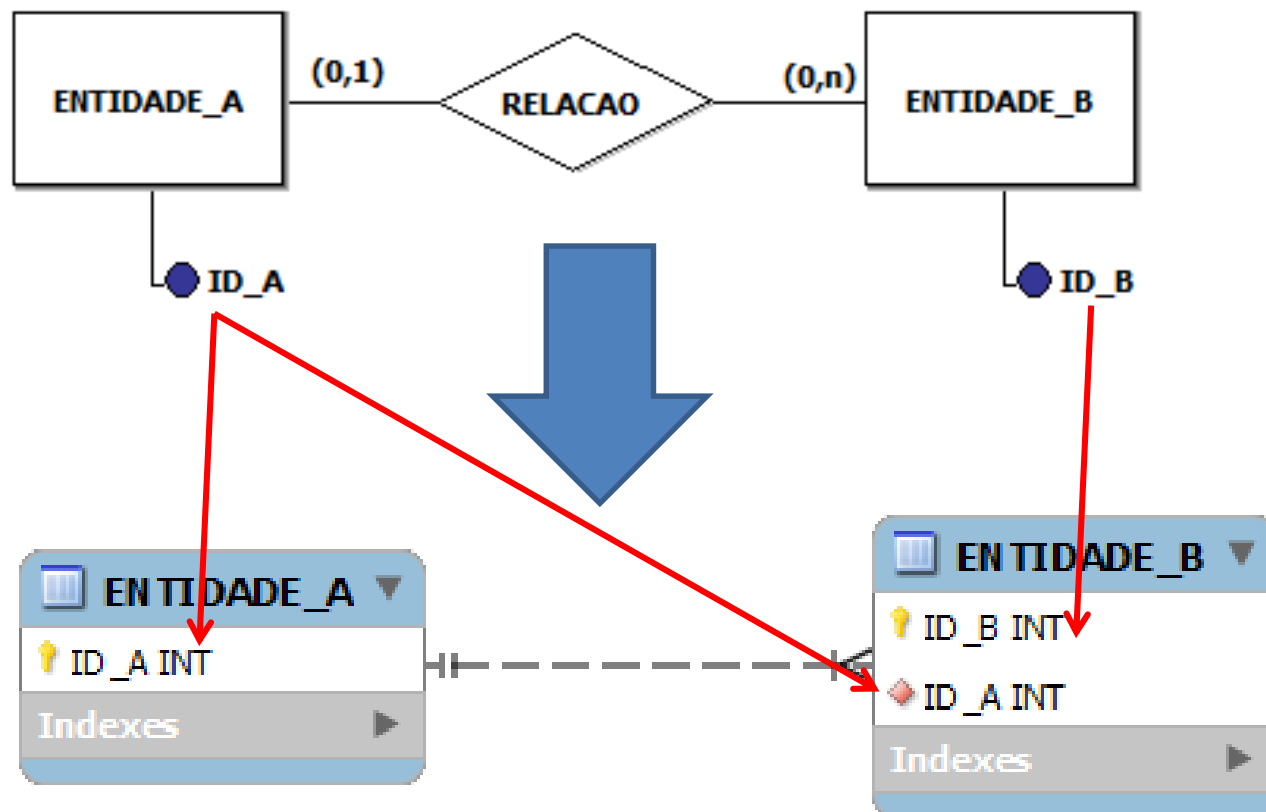
```
CREATE TABLE ENTIDADE_AB (  
    ID_A INT NOT NULL,  
    ID_B INT NOT NULL,  
    PRIMARY KEY (ID_A, ID_B),  
);
```

Transformação - Relacionamentos 1:n

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição coluna	Fusão tabelas
Relacionamentos 1:n			
	±	✓	×
	±	✓	×
	×	✓	×
	×	✓	×

✓ Alternativa preferida ± Pode ser usada × Não usar

Transformação - Relacionamentos 1:n (Adição de Coluna)

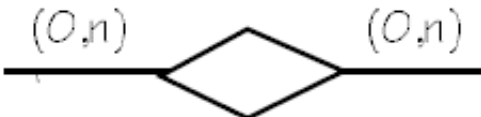
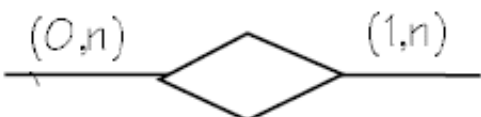
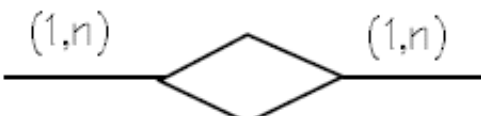


SQL – DEFINIÇÃO DE DADOS

```
CREATE TABLE ENTIDADE_A (  
    ID_A INT NOT NULL,  
    PRIMARY KEY (ID_A),  
);
```

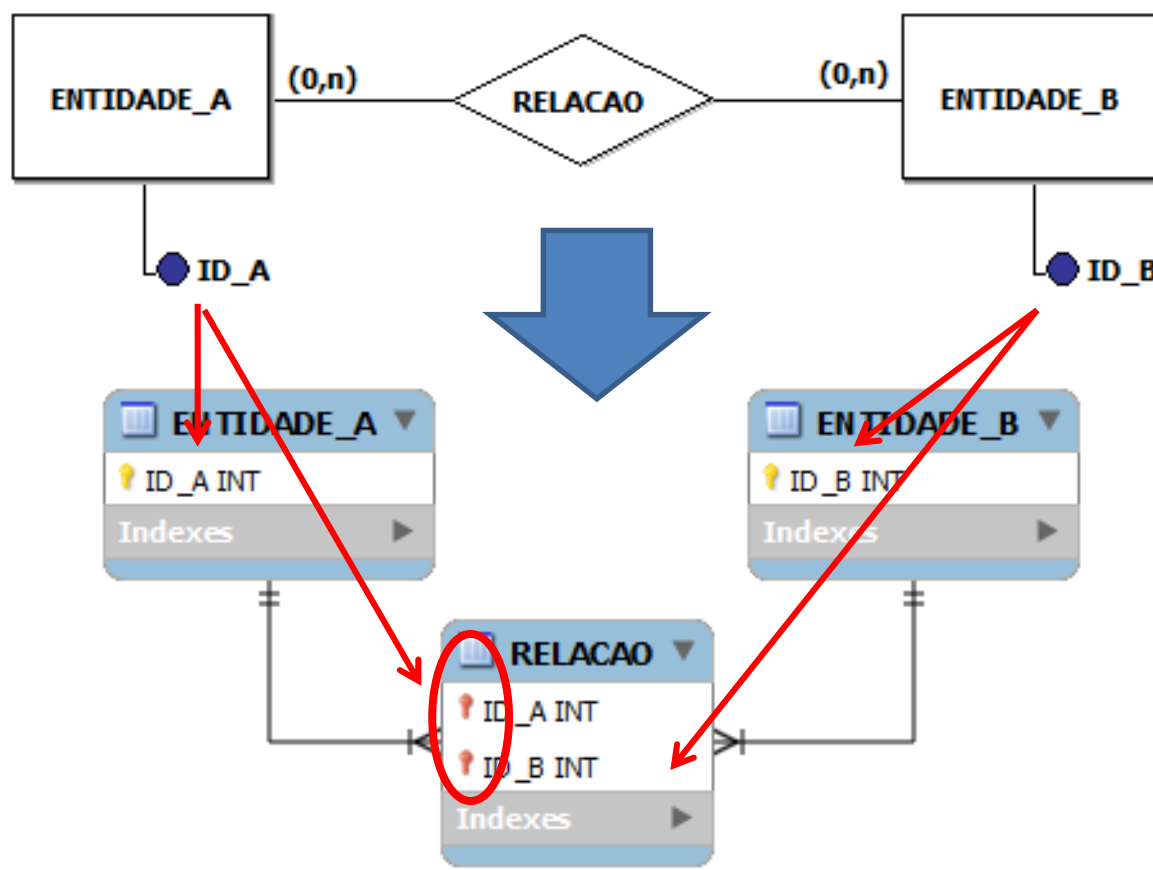
```
CREATE TABLE ENTIDADE_B (  
    ID_B INT NOT NULL,  
    ID_A INT,  
    PRIMARY KEY (ID_B),  
    FOREIGN KEY (ID_A) REFERENCES ENTIDADE_A (ID_A)  
);
```

Transformação - Relacionamentos n:n

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição coluna	Fusão tabelas
Relacionamentos n:n			
	✓	×	×
	✓	×	×
	✓	×	×

✓ Alternativa preferida ± Pode ser usada × Não usar

Transformação - Relacionamentos n:n (Tabela Própria)



SQL – DEFINIÇÃO DE DADOS

```
CREATE TABLE ENTIDADE_A (  
    ID_A INT NOT NULL,  
    PRIMARY KEY (ID_A),  
);
```

```
CREATE TABLE ENTIDADE_B (  
    ID_B INT NOT NULL,  
    PRIMARY KEY (ID_B),  
);
```

```
CREATE TABLE RELACAO (  
    ID_A INT NOT NULL,  
    ID_B INT NOT NULL,  
    PRIMARY KEY (ID_A, ID_B),  
    FOREIGN KEY (ID_A) REFERENCES ENTIDADE_A (ID_A)  
    FOREIGN KEY (ID_B) REFERENCES ENTIDADE_B (ID_B)  
);
```