



**UC1:** Desenvolver algoritmos e estrutura de dados  
para aplicativos móveis

## **Bancos de Dados**

**Modelo Lógico**

**Docente:** Felipe B. Benutti

# Visão Geral de um Projeto (etapas)

## 1. Modelagem com ER (**Conceitual**)

- Modelagem de entidades (e seus atributos), relacionamentos (e suas cardinalidades)
- Proposta de solução.

## 2. **Transformação entre Modelos (Lógico)**

- **Transformação de ER para Relacional**

## 3. Definição do BD em SQL (**Físico**)

- Criação de Scripts DDL;

# Chave Primária

- Deve ser possível distinguir um registro (linha) de todas as outras.
  - Isso é expresso através de seus atributos.
  - Os valores dos atributos de uma linha devem ser tais que permitam identificá-la de uma forma única.

ALUNO			
id_aluno	nome	rua	cidade
1	José	Gertrudes	São João
2	João	Oscar P.	São João
3	Maria	Av. Paulist.	São Paulo
4	Joaquim	Oscar P.	A. Prata
5	José	Gertrudes	São João
6	Maria	B. Faria L.	São Paulo

# Chave Primária

- Na tabela ao lado, a única coluna que tem **dados únicos** é a **id\_aluno**.
- Por tanto, ela será escolhida como nossa **Chave Primária**.

ALUNO			
 id_aluno	nome	rua	cidade
1	José	Gertrudes	São João
2	João	Oscar P.	São João
3	Maria	Av. Paulist.	São Paulo
4	Joaquim	Oscar P.	A. Prata
5	José	Gertrudes	São João
6	Maria	B. Faria L.	São Paulo



# Chave Estrangeira



- Uma tabela *T1* pode possuir entre os seus atributos a chave primária de uma outra tabela *T2*.
- Este atributo é denominado de **chave estrangeira**.
- Neste exemplo:
  - “id\_produto” na tabela VENDA é chave estrangeira para a tabela PRODUTO

PRODUTO			
id_produto	descricao	valor	qtd_estoq
1	Livro XX	29,90	10
2	Notebook YY	3.990,00	2
3	Blu-ray ZZ	25,90	20
4	Jogo PS4 WW	149,90	7

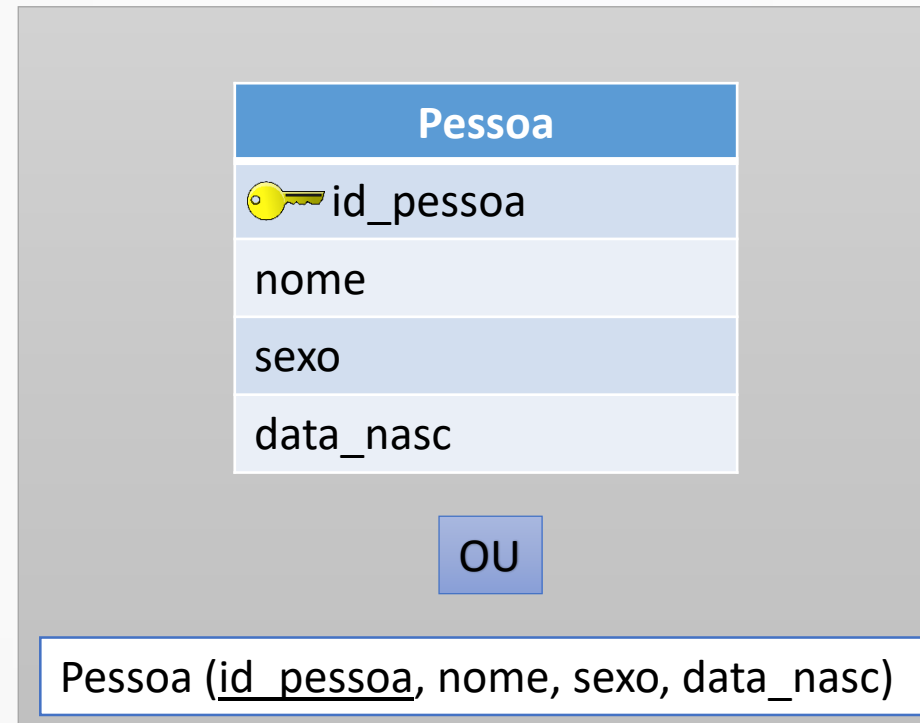
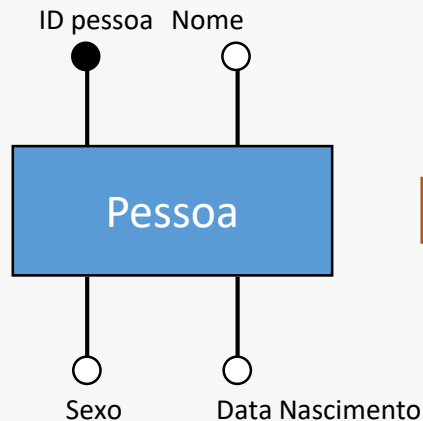
VENDA			
id_venda	id_produto	qtd_prod	data_venda
1	1	5	10/01/2018
2	1	1	04/02/2018
3	2	1	30/03/2018
4	3	3	16/05/2018

# Mapeamento de ER (Conceitual) para Relacional (Lógico)

- Possui regras, para obter um modelo de dados:
  - Com **bom desempenho** para consultas;
  - Que **simplifique o desenvolvimento e manutenção** de aplicações de usuário;
- Passos:
  - Tradução inicial de **entidades**
  - Tradução de **relacionamentos** e respectivos **atributos**

# Implementação inicial de entidades

- Cada entidade é traduzida para uma tabela
- Cada atributo define uma coluna da tabela
- Utilizar sufixos/prefixos nos nomes das chaves
- Ex:



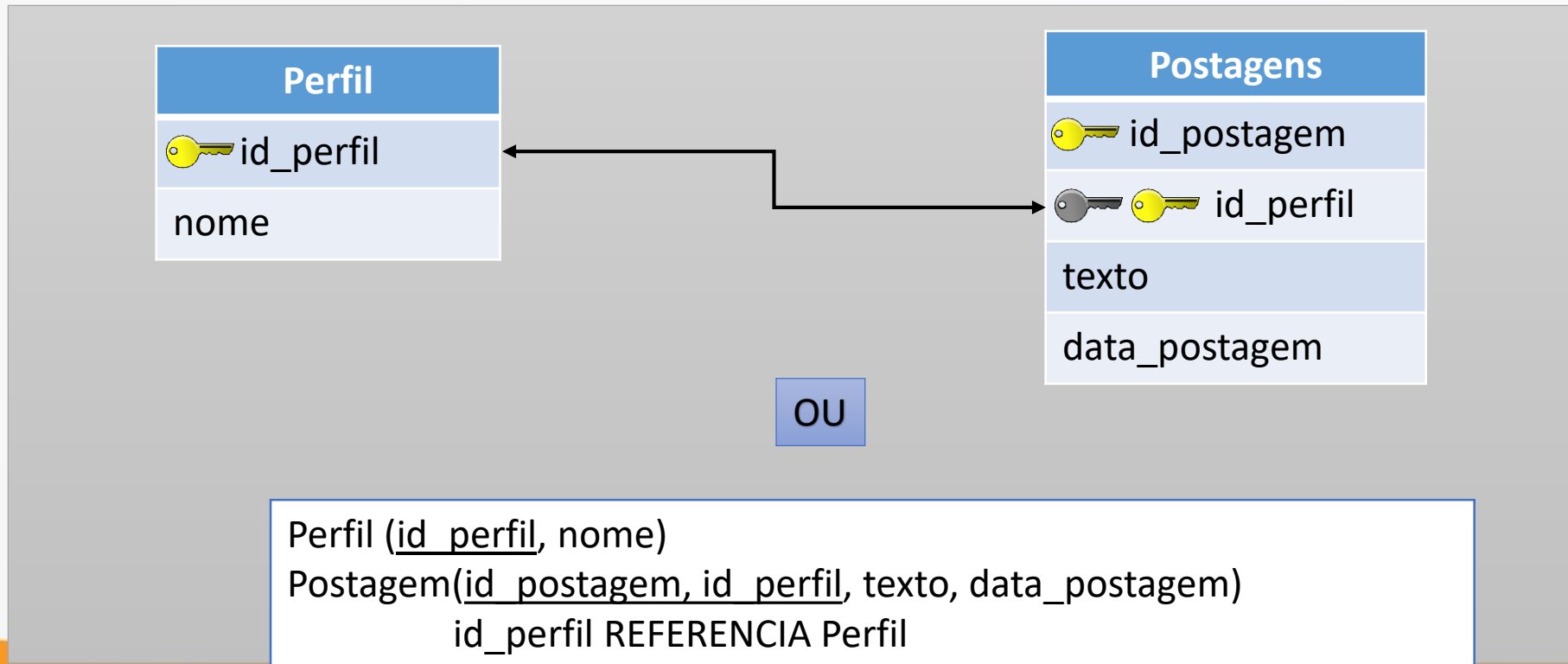
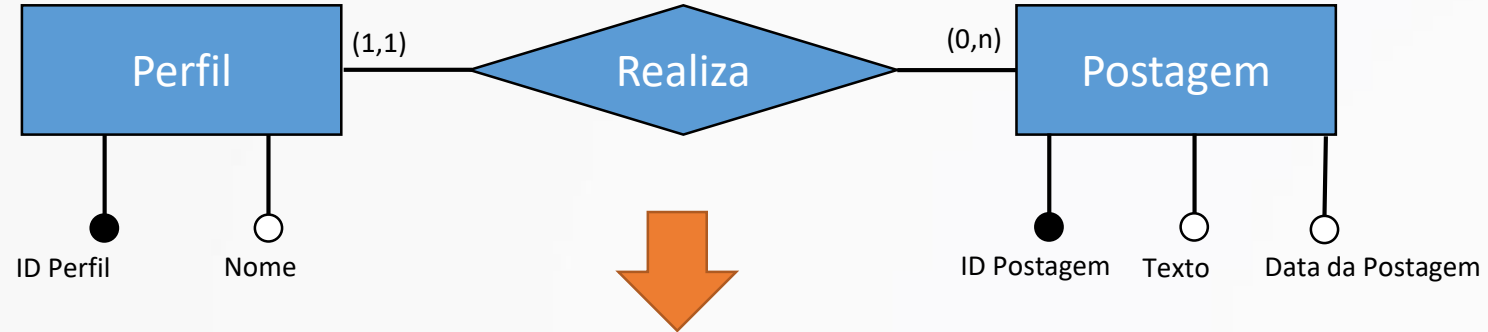
# Implementação de relacionamentos – 1 para N

- **Colunas adicionais** dentro de tabela de entidade
  - Coluna(s) é(são) inseridas em uma das tabelas que participam do relacionamento
  - Acontece quando a entidade tem cardinalidade máxima 1
  - São inseridas colunas correspondentes ao identificador da entidade relacionada e colunas correspondentes aos atributos do relacionamento



# Implementação de relacionamentos – 1 para N

- Ex:



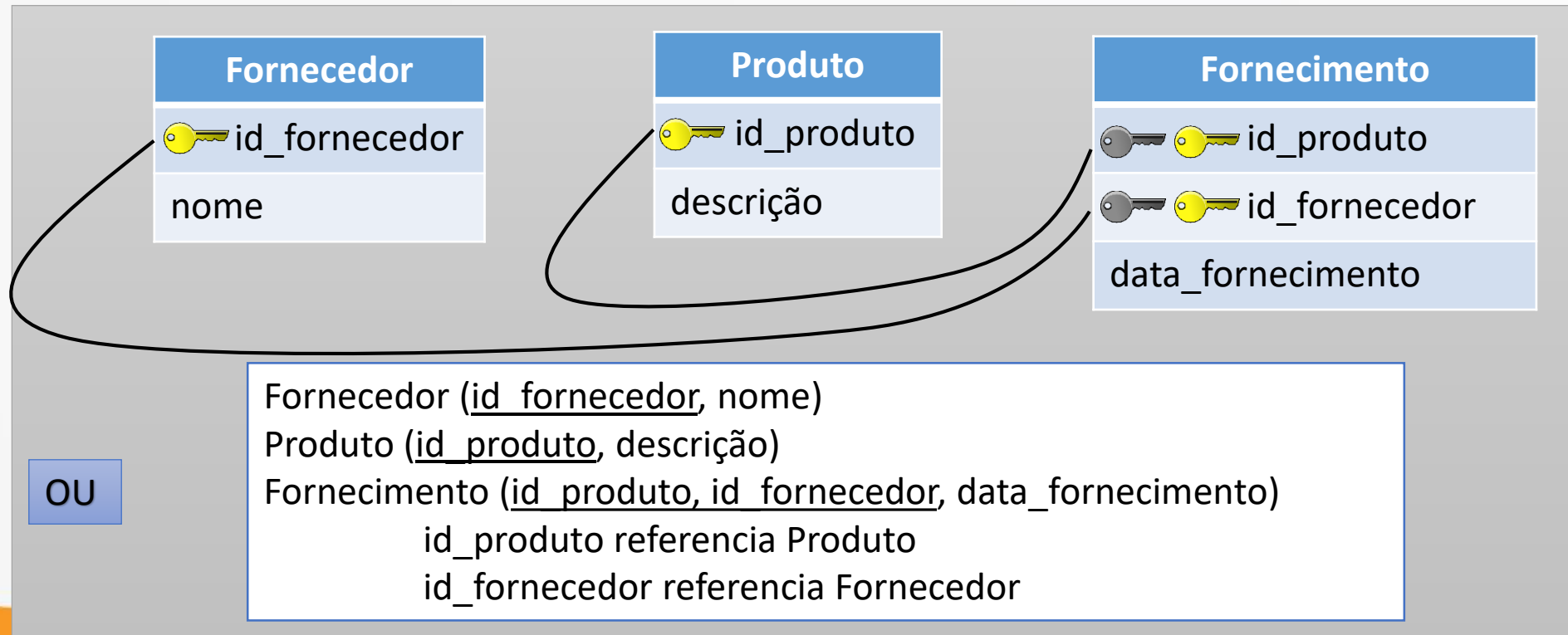
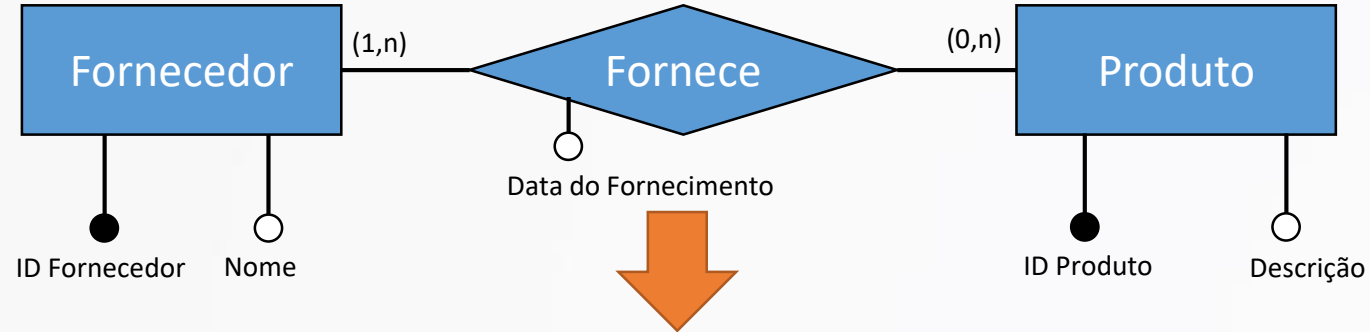
# Implementação de relacionamentos – N para N

- **Tabela própria**

- Tabela contém colunas correspondentes aos **identificadores das entidades relacionadas** e atributos do relacionamento
- Chave primária é formada pelas colunas correspondentes aos identificadores das entidades relacionadas e pelos atributos identificadores do relacionamento

# Implementação de relacionamentos – N para N

• Ex:

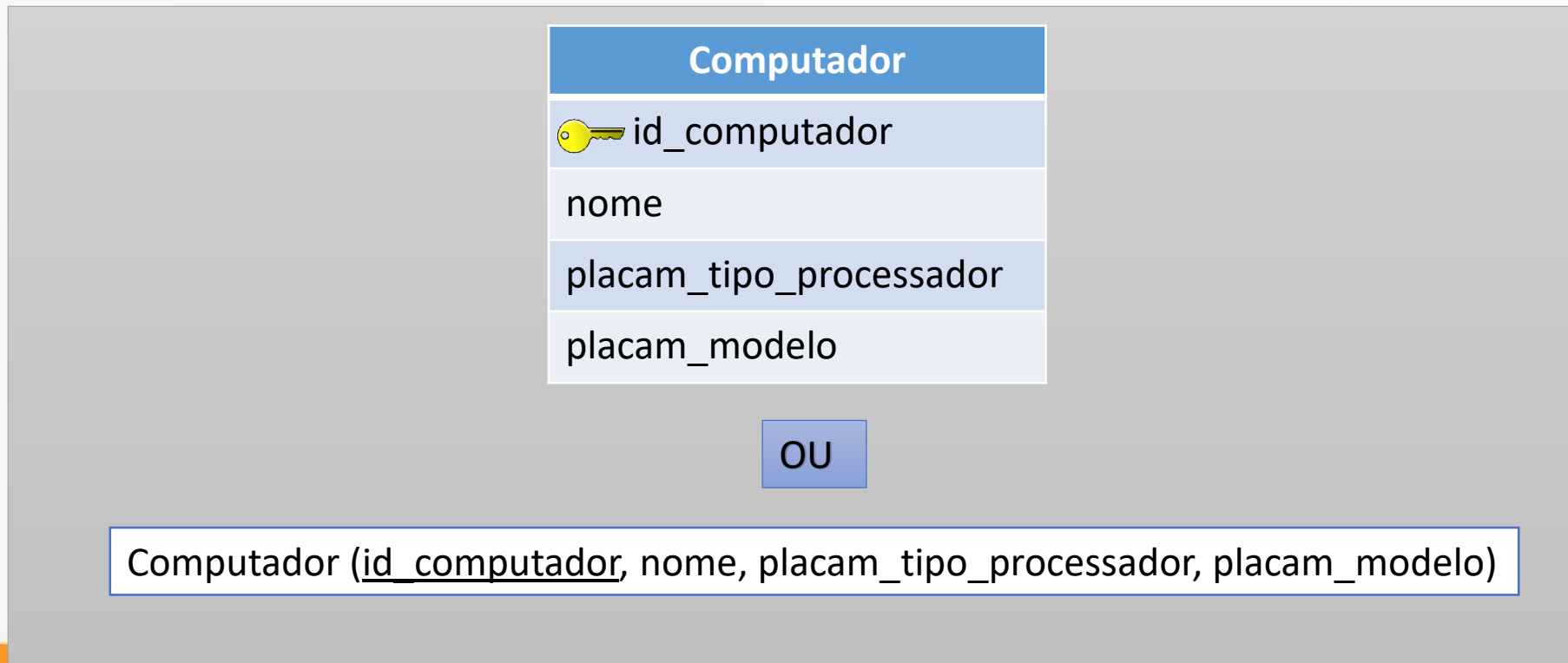
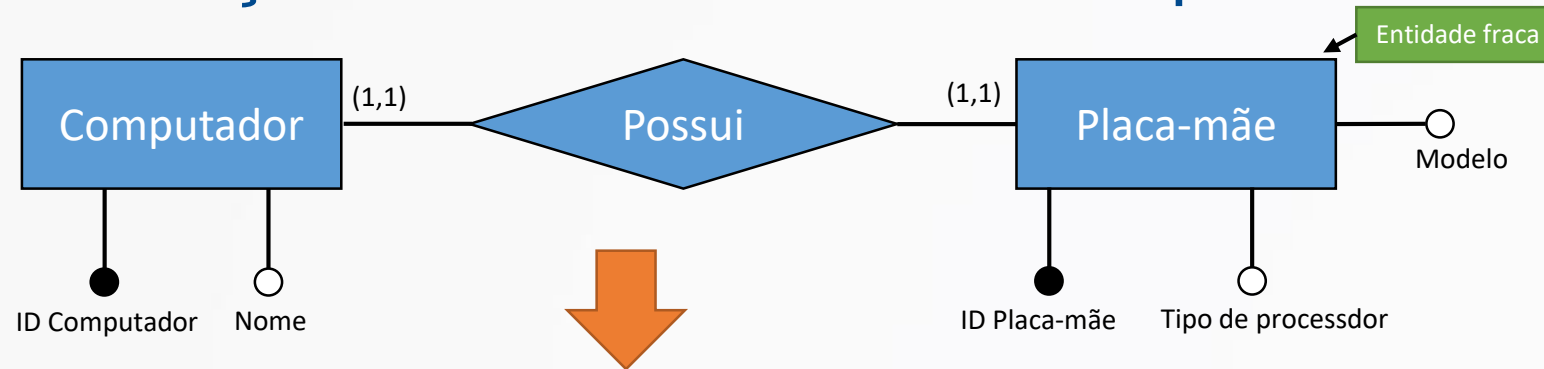


# Implementação de relacionamentos – 1 para 1

- **Fusão de tabelas de entidades**
  - Pode ser aplicada quando o relacionamento é do tipo 1:1
  - Todos os atributos de ambas as entidades são implementados em uma única tabela

# Implementação de relacionamentos – 1 para 1

• Ex:



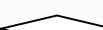

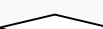
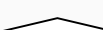
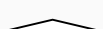
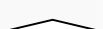




# Tabela de Conversões: ER para Relacional

✓ Alternativa Preferida

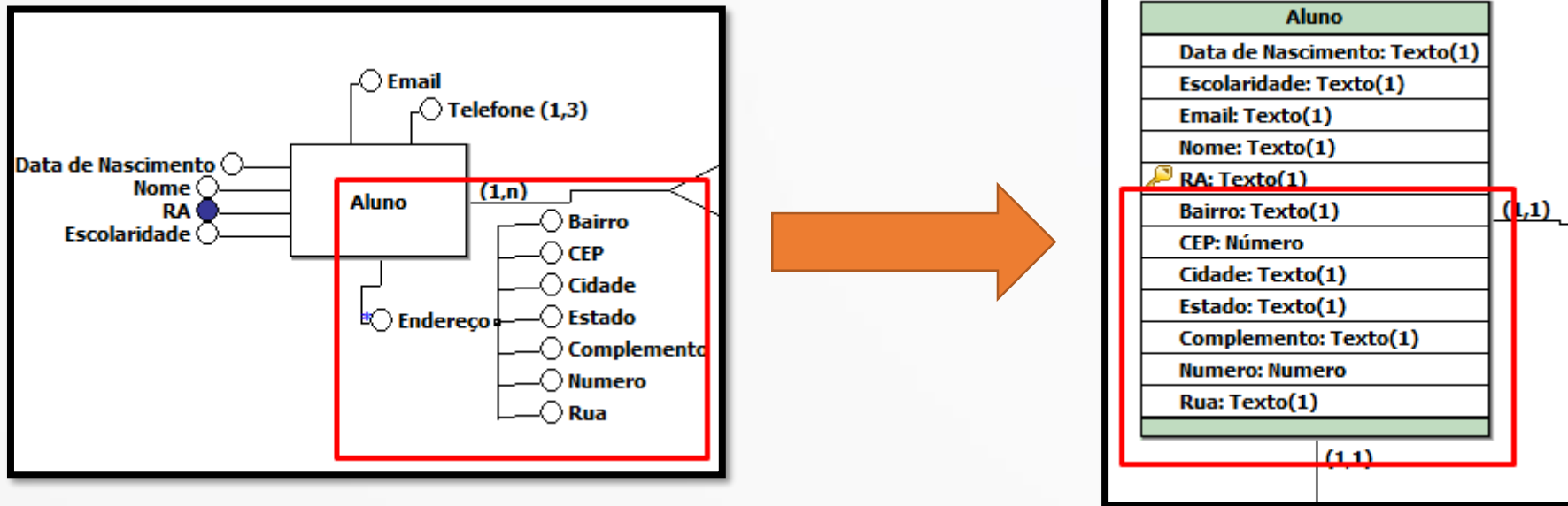
± Pode ser usada

⊘ Não usar

Tipo de Relacionamento	Regra de implementação		
	Tabela Própria	Adição Coluna	Fusão Tabelas
Relacionamento 1:1			
<u>(0,1)</u> —  — <u>(0,1)</u>	±	✓	⊘
<u>(0,1)</u> —  — <u>(1,1)</u>	⊘	±	✓
<u>(1,1)</u> —  — <u>(1,1)</u>	⊘	⊘	✓
Relacionamento 1:n			
<u>(0,1)</u> —  — <u>(0,n)</u>	±	✓	⊘
<u>(0,1)</u> —  — <u>(1,n)</u>	±	✓	⊘
<u>(1,1)</u> —  — <u>(0,n)</u>	⊘	✓	⊘
<u>(1,1)</u> —  — <u>(1,n)</u>	⊘	✓	⊘
Relacionamento n:n			
<u>(0,n)</u> —  — <u>(0,n)</u>	✓	⊘	⊘
<u>(0,n)</u> —  — <u>(1,n)</u>	✓	⊘	⊘
<u>(1,n)</u> —  — <u>(1,n)</u>	✓	⊘	⊘

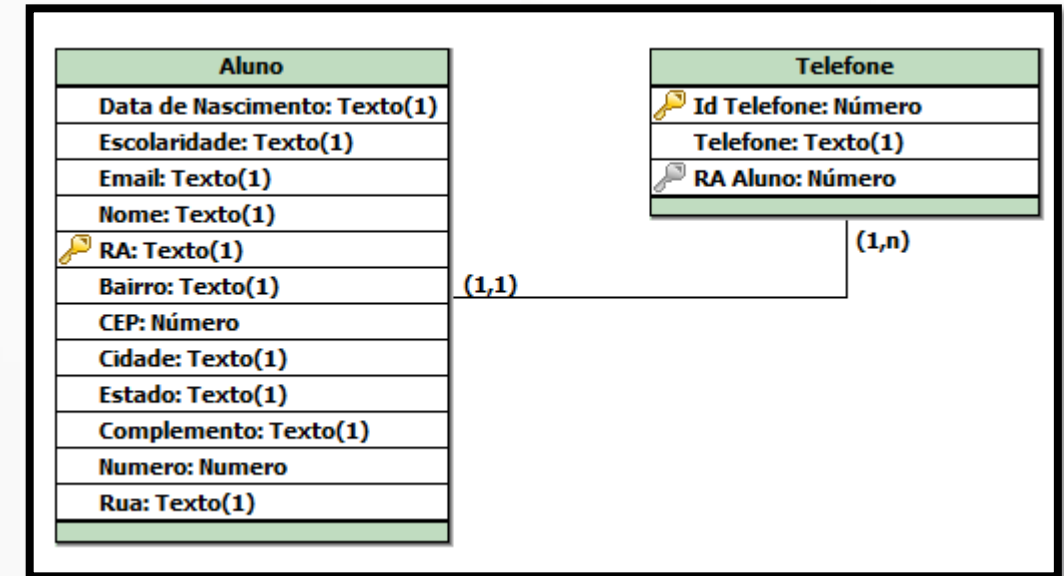
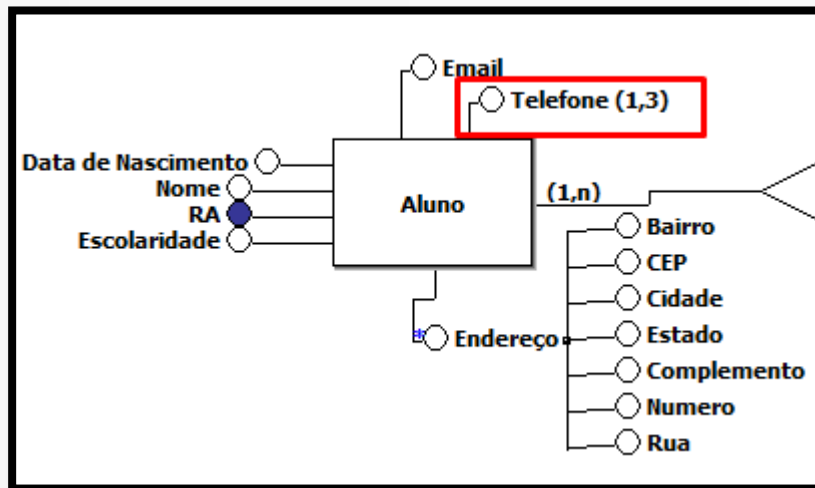
# Atributos Compostos

- Atributos compostos, por via de regra, sempre irão ter todos os seus atributos incorporados à tabela pertencente. Exemplo:



# Atributos Multivalorados

- Atributos multivalorados, por via de regra, sempre irão criar uma tabela própria para acomodar os múltiplos valores de um determinado atributo. Exemplo:





# Modelo lógico

- Ferramenta para realizar o desenvolvimento dos diagramas:
  - brModelo;
- Explicação na prática:
  - Realizar a conversão de um modelo conceitual (escola) utilizando o programa brModelo;

# Modelo lógico

- Exercício de fixação:
  - Converter os DER criados no exercício anterior para o modelo lógico dos mesmos:
    1. Biblioteca;
    2. Loja de materiais de construção (estoque);
    3. Rede Social (Instagram);

# Modelo Lógico

- Exercícios extra:
  - Converter os DER criados no exercício anterior para o modelo lógico dos mesmos:
    1. Escritório de advocacia (clientes);
    2. Corretor de imóveis;