

---

# Banco de Dados

## **FORMAS NORMAIS**

Prof. Dr. Edimar Manica

[edimar.manica@ibiruba.ifrs.edu.br](mailto:edimar.manica@ibiruba.ifrs.edu.br)

# NORMALIZAÇÃO

---

- O processo de normalização visa minimizar a redundância de dados e evitar as anomalias de atualização, permitindo:
  - Reduzir o espaço físico necessário para armazenar o banco de dados
  - Melhorar a organização dos dados
  - Reduzir o impacto de atualizações, inserções e remoções nos dados do banco de dados

# NORMALIZAÇÃO

---

- Objetivo Geral
  - Medir a qualidade do projeto
- Objetivo específico
  - Medir formalmente porque um agrupamento de atributos em um esquema é melhor que outro
- Aspectos considerados
  - Lógico: interpretação do esquema e dos atributos
  - Armazenamento: como as tuplas são armazenadas e atualizadas

# NORMALIZAÇÃO

---

- Processo através do qual esquemas são sucessivamente decompostos até que satisfaçam determinadas propriedades ou formas normais
- Resultados esperados:
  - Compatibilidade semântica entre os atributos de uma mesma relação
  - Ausência de valores redundantes nas relações (inexistência de anomalias de atualização)
  - Número reduzido de valores nulos nas relações
  - Impossibilidade de se gerar tuplas espúrias ao se recompor as relações decompostas

# NORMALIZAÇÃO

---

- O **processo de normalização** é constituído por um conjunto de formas normais
- As **formas normais** especificam critérios que definem quando uma tabela está bem estruturada ou não
- Para saber se uma tabela está bem estruturada, você deve verificar se a estrutura da tabela satisfaz todas as formas normais

# PASSOS DA NORMALIZAÇÃO

---

1ª FN

2ª FN

3ª FN

...

Diminui a redundância de dados

Aumenta o desempenho das consultas

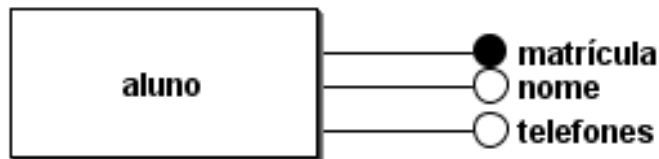
# 1ª FORMA NORMAL

---

- Uma tabela está na **Primeira Forma Normal (1FN)** se e somente se todos os atributos contiverem apenas dados atômicos, ou seja, cada atributo pode ter apenas um valor por registro (tupla)

# 1ª FORMA NORMAL

- Exemplo de estrutura de tabela que **não** está na 1FN



aluno

matrícula	nome	telefones
75968	João	54991552725, 54999252623, 5193567845
75967	Maria	54998752623, 54991784512
75968	José	54991456532

Esta tabela não está na 1FN porque o atributo **telefones** possui mais de um telefone por linha (tupla). Por exemplo, o aluno João possui três telefones.



# 1ª FORMA NORMAL

---

Para adequar uma tabela que não está na 1FN é necessário realizar os seguintes passos:

1. Criar uma tabela para conter os dados do atributo não atômico
2. Criar na nova tabela um atributo para conter o atributo não atômico da tabela original
3. Criar na nova tabela um atributo para conter a chave primária da tabela original
4. Definir uma chave estrangeira para garantir a relação entre a nova tabela e a tabela original
5. Definir a chave primária da nova tabela
6. Remover o atributo não atômico da tabela original.

# 1ª FORMA NORMAL

aluno

matrícula	nome	telefones
75968	João	54991552725, 54999252623, 5193567845
75967	Maria	54998752623, 54991784512
75968	José	54991456532

----- antes de normalizar  
depois de normalizar

aluno

matrícula	nome
75968	João
75967	Maria
75968	José

telefone

mat_aluno	telefone
75968	54991552725
75968	54999252623
75968	5193567845
75967	54998752623
75967	54991784512
75968	54991456532

# 1ª FORMA NORMAL

aluno

matrícula	nome	telefones
75968	João	54991552725, 54999252623, 5193567845
75967	Maria	54998752623, 54991784512
75968	José	54991456532

antes de normalizar  
depois de normalizar

aluno

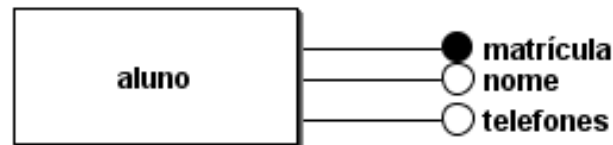
matrícula	nome	mat_aluno	telefone
75968	João	75968	54991552725
75967	Maria	75968	54999252623
75968	José	75968	5193567845
		75967	54998752623
		75967	54991784512
		75968	54991456532

O atributo **telefones**  
foi removido da  
tabela **aluno**.

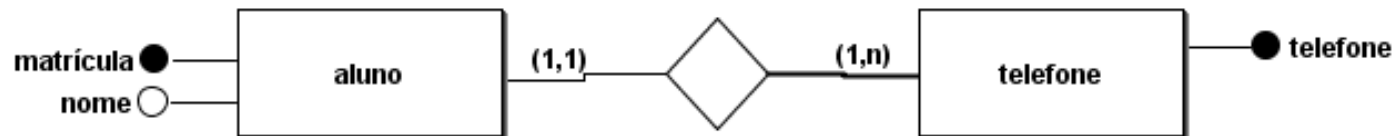
A tabela criada, **telefone**, recebe  
o atributo não atômico **telefone**,  
bem como a chave primária da  
tabela **aluno** como chave  
estrangeira

# 1ª FORMA NORMAL

## ➤ Estrutura antes da normalização

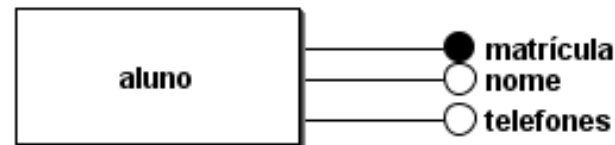


## ➤ Estrutura após a normalização

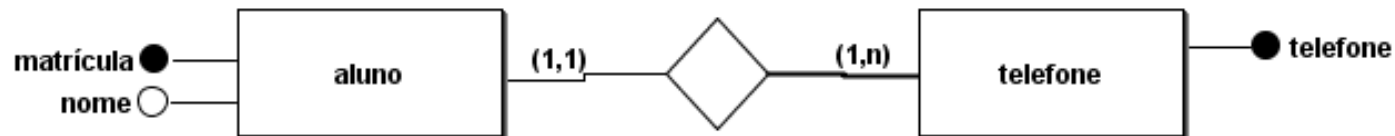


# 1ª FORMA NORMAL

## ➤ Estrutura antes da normalização



## ➤ Estrutura após a normalização



Note que o processo de normalização de uma tabela para a 1FN é equivalente ao mapeamento de um atributo multivalorado do modelo ER para o modelo relacional, ou seja, se você definir bem seu modelo ER e fizer o mapeamento correto, suas tabelas já estarão na 1FN.

# 2ª FORMA NORMAL

---

- Uma tabela está na **Segunda Forma Normal (2FN)** se e somente se ela estiver na 1FN e todos os atributos não chave primária puderem ser obtidos da combinação de todos os atributos que formam a chave primária.
- Não pode haver dependência funcional parcial
  - Atributos não-chave não podem depender de apenas **parte da chave**

# 2ª FORMA NORMAL

- Exemplo de estrutura de tabela que **não** está na 2FN

Chave primária composta

matrícula	cod_disciplina	nome_aluno	nome_disciplina	ch_disciplina	nota
75965	CC01	João	Algoritmos	66	7,85
75965	CC02	João	Lógica	33	8,90
75966	CC01	Maria	Algoritmos	66	9,00
75966	CC02	Maria	Lógica	33	10,00
75967	CC01	Pedro	Algoritmos	66	5,00

# 2ª FORMA NORMAL

- Exemplo de estrutura de tabela que **não** está na 2FN

matrícula	cod_disciplina	nome_aluno	nome_disciplina	ch_disciplina	nota
75965	CC01	João	Algoritmos	66	7,85
75965	CC02	João	Lógica	33	8,90
75966	CC01	Maria	Algoritmos	66	9,00
75966	CC02	Maria	Lógica	33	10,00
75967	CC01	Pedro	Algoritmos	66	5,00

Os valores de **nome\_aluno** podem ser obtidos apenas a partir do atributo **matrícula**.





# 2ª FORMA NORMAL

- Exemplo de estrutura de tabela que **não** está na 2FN

matrícula	cod_disciplina	nome_aluno	nome_disciplina	ch_disciplina	nota
75965	CC01	João	Algoritmos	66	7,85
75965	CC02	João	Lógica	33	8,90
75966	CC01	Maria	Algoritmos	66	9,00
75966	CC02	Maria	Lógica	33	10,00
75967	CC01	Pedro	Algoritmos	66	5,00

Os valores de **nome\_disciplina** e **ch\_disciplina** podem ser obtidos apenas a partir do atributo **cod\_disciplina**.



# 2ª FORMA NORMAL

- Exemplo de estrutura de tabela que **não** está na 2FN

matrícula	cod_disciplina	nome_aluno	nome_disciplina	ch_disciplina	nota
75965	CC01	João	Algoritmos	66	7,85
75965	CC02	João	Lógica	33	8,90
75966	CC01	Maria	Algoritmos	66	9,00
75966	CC02	Maria	Lógica	33	10,00
75967	CC01	Pedro	Algoritmos	66	5,00

Os valores de **nota** só podem ser obtidos a partir do par de atributos **matrícula, cod\_disciplina**.



## 2ª FORMA NORMAL

---

Para adequar uma tabela que não está na 2FN é necessário realizar os seguintes passos:

1. Criar duas novas tabelas para armazenar os dados dos campos redundantes, em que seus valores apresentam repetição de valores
2. Remover os campos com valores redundantes da tabela original
3. Criar chaves primárias nas novas tabelas criadas com base na chave primária da tabela original
4. Criar relacionamentos 1XN entre as novas tabelas criadas e a tabela original

# 2ª FORMA NORMAL

matrícula	cod_disciplina	nome_aluno	nome_disciplina	ch_disciplina	nota
75965	CC01	João	Algoritmos	66	7,85
75965	CC02	João	Lógica	33	8,90
75966	CC01	Maria	Algoritmos	66	9,00
75966	CC02	Maria	Lógica	33	10,00

----- antes de normalizar  
depois de normalizar

aluno		disciplina			notas		
matrícula	nome	codigo	nome	ch	mat_aluno	cod_disciplina	nota
75965	João	CC01	Algoritmos	66	75965	CC01	7,85
75966	Maria	CC02	Lógica	33	75965	CC02	8,90
					75966	CC01	9,00
					75966	CC02	10,00

# 2ª FORMA NORMAL

matrícula	cod_disciplina	nome_aluno	nome_disciplina	ch_disciplina	nota
75965	CC01	João	Algoritmos	66	7,85
75965	CC02	João	Lógica	33	8,90
75966	CC01	Maria	Algoritmos	66	9,00
75966	CC02	Maria	Lógica	33	10,00

----- antes de normalizar  
depois de normalizar

aluno		disciplina			notas		
matrícula	nome	codigo	nome	ch	mat_aluno	cod_disciplina	nota
75965	João	CC01	Algoritmos	66	75965	CC01	7,85
75966	Maria	CC02	Lógica	33	75965	CC02	8,90
					75966	CC01	9,00
					75966	CC02	10,00

# 2ª FORMA NORMAL

matrícula	cod_disciplina	nome_aluno	nome_disciplina	ch_disciplina	nota
75965	CC01	João	Algoritmos	66	7,85
75965	CC02	João	Lógica	33	8,90
75966	CC01	Maria	Algoritmos	66	9,00
75966	CC02	Maria	Lógica	33	10,00

----- antes de normalizar  
depois de normalizar

aluno		disciplina			notas		
matrícula	nome	codigo	nome	ch	mat_aluno	cod_disciplina	nota
75965	João	CC01	Algoritmos	66	75965	CC01	7,85
75966	Maria	CC02	Lógica	33	75965	CC02	8,90
					75966	CC01	9,00
					75966	CC02	10,00

# 2ª FORMA NORMAL

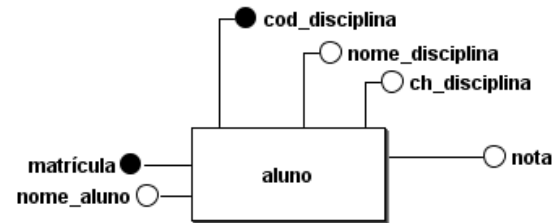
matrícula	cod_disciplina	nome_aluno	nome_disciplina	ch_disciplina	nota
75965	CC01	João	Algoritmos	66	7,85
75965	CC02	João	Lógica	33	8,90
75966	CC01	Maria	Algoritmos	66	9,00
75966	CC02	Maria	Lógica	33	10,00

----- antes de normalizar  
depois de normalizar

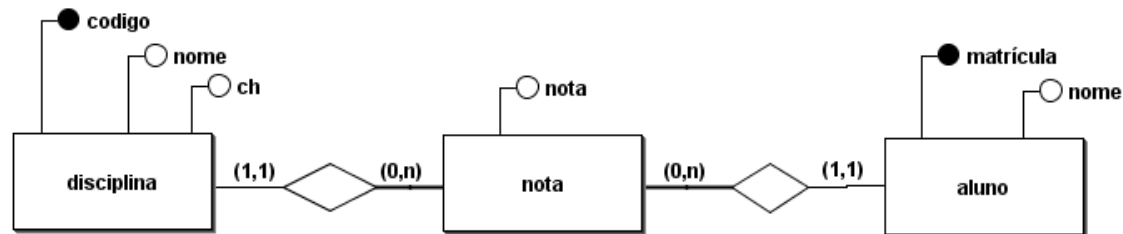
aluno		disciplina			notas		
matrícula	nome	codigo	nome	ch	mat_aluno	cod_disciplina	nota
75965	João	CC01	Algoritmos	66	75965	CC01	7,85
75966	Maria	CC02	Lógica	33	75965	CC02	8,90
					75966	CC01	9,00
					75966	CC02	10,00

# 2ª FORMA NORMAL

## ➤ Estrutura antes da normalização



## ➤ Estrutura após a normalização





# 3ª FORMA NORMAL

---

- Uma tabela está na **Terceira Forma Normal (3FN)** se e somente se ela estiver na 1FN e na 2FN e todos os atributos não chave primária puderem ser obtidos somente através da chave primária
- Não pode haver dependência transitiva ou indireta
  - Atributos não-chave não podem depender de atributos **não-chave**

# 3ª FORMA NORMAL

- Exemplo de estrutura de tabela que **não** está na 3FN

funcionário

matrícula	nome	sexo	categoria	salário
75975	Cris	F	Programador Jr	R\$ 2.500,00
75976	Sofia	F	Programador Sr	R\$ 6.500,00
75977	Renato	M	Programador Jr	R\$ 2.500,00
75978	Olívio	M	Analista	R\$ 9.500,00
75979	Samantha	F	Programador Sr	R\$ 6.500,00




chave primária  
simples

# 3ª FORMA NORMAL

- Exemplo de estrutura de tabela que **não** está na 3FN

funcionário

matrícula	nome	sexo	categoria	salário
75975	Cris	F	Programador Jr	R\$ 2.500,00
75976	Sofia	F	Programador Sr	R\$ 6.500,00
75977	Renato	M	Programador Jr	R\$ 2.500,00
75978	Olívio	M	Analista	R\$ 9.500,00
75979	Samantha	F	Programador Sr	R\$ 6.500,00



Através da chave primária é possível obter os valores de todos os outros atributos.




# 3ª FORMA NORMAL

- Exemplo de estrutura de tabela que **não** está na 3FN

funcionário

matrícula	nome	sexo	categoria	salário
75975	Cris	F	Programador Jr	R\$ 2.500,00
75976	Sofia	F	Programador Sr	R\$ 6.500,00
75977	Renato	M	Programador Jr	R\$ 2.500,00
75978	Olívio	M	Analista	R\$ 9.500,00
75979	Samantha	F	Programador Sr	R\$ 6.500,00



No entanto, eu também posso obter o valor do **salário** pelo atributo **categoria**, o que viola a 3FN.



# 3ª FORMA NORMAL

---

Para adequar uma tabela que não está na 3FN é necessário realizar os seguintes passos:

1. Criar uma tabela para conter os atributos que não podem ser obtidos exclusivamente da chave primária da tabela original
2. Definir como chave primária da tabela criada o atributo que é capaz de obter os dados não chaves da tabela original
3. Mover os atributos não chave que não são obtidos exclusivamente pela chave primária da tabela original para a nova tabela
4. Definir como chave estrangeira o atributo que é capaz de obter os dados não chaves da tabela original

# 3ª FORMA NORMAL

funcionário

matrícula	nome	sexo	categoria	salário
75975	Cris	F	Programador Jr	R\$ 2.500,00
75976	Sofia	F	Programador Sr	R\$ 6.500,00
75977	Renato	M	Programador Jr	R\$ 2.500,00
75978	Olívio	M	Analista	R\$ 9.500,00
75979	Samantha	F	Programador Sr	R\$ 6.500,00

antes de normalizar

depois de normalizar

funcionário

matrícula	nome	sexo	cod_categoria
75975	Cris	F	1
75976	Sofia	F	2
75977	Renato	M	1
75978	Olívio	M	3
75979	Samantha	F	2

categoria

codigo	descrição	salário
1	Programador Jr	R\$ 2.500,00
2	Programador Sr	R\$ 6.500,00
3	Analista	R\$ 9.500,00

# 3ª FORMA NORMAL

funcionário

matrícula	nome	sexo	categoria	salário
75975	Cris	F	Programador Jr	R\$ 2.500,00
75976	Sofia	F	Programador Sr	R\$ 6.500,00
75977	Renato	M	Programador Jr	R\$ 2.500,00
75978	Olívio	M	Analista	R\$ 9.500,00
75979	Samantha	F	Programador Sr	R\$ 6.500,00

antes de normalizar  
depois de normalizar

funcionário

matrícula	nome	sexo	cod_categoria
75975	Cris	F	1
75976	Sofia	F	2
75977	Renato	M	1
75978	Olívio	M	3
75979	Samantha	F	2

categoria

codigo	descrição	salário
1	Programador Jr	R\$ 2.500,00
2	Programador Sr	R\$ 6.500,00
3	Analista	R\$ 9.500,00

# 3ª FORMA NORMAL

funcionário

matrícula	nome	sexo	categoria	salário
75975	Cris	F	Programador Jr	R\$ 2.500,00
75976	Sofia	F	Programador Sr	R\$ 6.500,00
75977	Renato	M	Programador Jr	R\$ 2.500,00
75978	Olívio	M	Analista	R\$ 9.500,00
75979	Samantha	F	Programador Sr	R\$ 6.500,00

antes de normalizar  
depois de normalizar

funcionário

matrícula	nome	sexo	cod_categoria
75975	Cris	F	1
75976	Sofia	F	2
75977	Renato	M	1
75978	Olívio	M	3
75979	Samantha	F	2

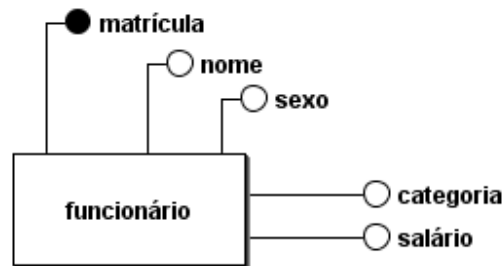
categoria

codigo	descrição	salário
1	Programador Jr	R\$ 2.500,00
2	Programador Sr	R\$ 6.500,00
3	Analista	R\$ 9.500,00

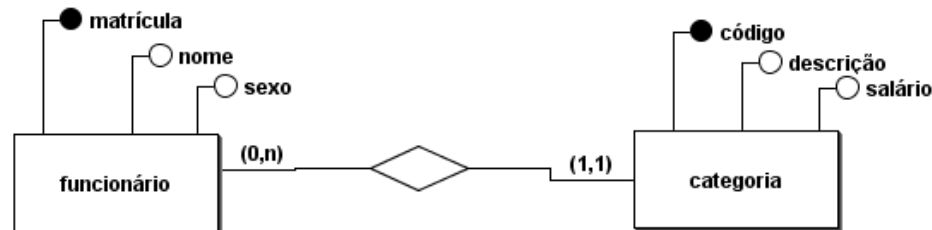


# 3ª FORMA NORMAL

## ➤ Estrutura antes da normalização



## ➤ Estrutura após a normalização



# RESUMO

Forma Normal	Teste	Remédio (normalização)
1ª FN	Relação não tem atributos multivalorados ou relações aninhadas	Definir novas relações para cada atributo multivalorado ou relação aninhada
2ª FN	Para relações onde a chave primária contém múltiplos atributos, nenhum atributo não-chave deve ser dependente funcionalmente de apenas uma parte da chave primária	Decompor e definir uma nova relação para cada chave parcial com seus atributos dependentes. Mantenha a relação original com a sua chave e seus atributos funcionalmente dependentes
3ª FN	Relação não pode ter um atributo não-chave determinado funcionalmente por outro atributo não-chave. Não tem dependência transitiva de um atributo não chave	Decompor e definir uma relação que inclua os atributos não-chave que determinam funcionalmente os outros atributos não-chave

# FORMAS NORMAIS - FECHAMENTO

---

- Informalmente, podemos dizer que um esquema de relação R está normalizado (ou seja, na 3FN) se todos os seus atributos são dependentes exclusivamente de suas chaves
- A forma normal de um relação refere à maior forma normal à qual a relação satisfaz, e então indica a qual grau a relação foi normalizada
- Não necessariamente a 3FN

# FORMAS NORMAIS - FECHAMENTO

---

- Formas normais sozinhas não garantem um bom projeto de BD
- Não é suficiente conferir cada relação individualmente verificando se satisfaz a 3FN
- Outras duas propriedades necessárias
  - **Lossless join**: garante que tuplas espúrias não são geradas quando se faz uma junção
  - **Preservação de dependência**: garante que cada dependência funcional é representada em alguma relação

# FORMAS NORMAIS - FECHAMENTO

---

- Além da terceira forma normal, existem várias outras formas normais, tais como: 4FN, 5FN, Boyce-Codd Normal Form (BCNF), Domain Key Normal Form (DKNF), entre outras
- Nesta disciplina, iremos estudar até a 3FN, porque concordamos com a opinião de alguns autores — Powell (2006) e Connolly, Begg (2000), da literatura de banco de dados — que afirmam que, acima da 3FN, o resultado da normalização pode ficar tão complexo que compromete o desempenho do banco de dados

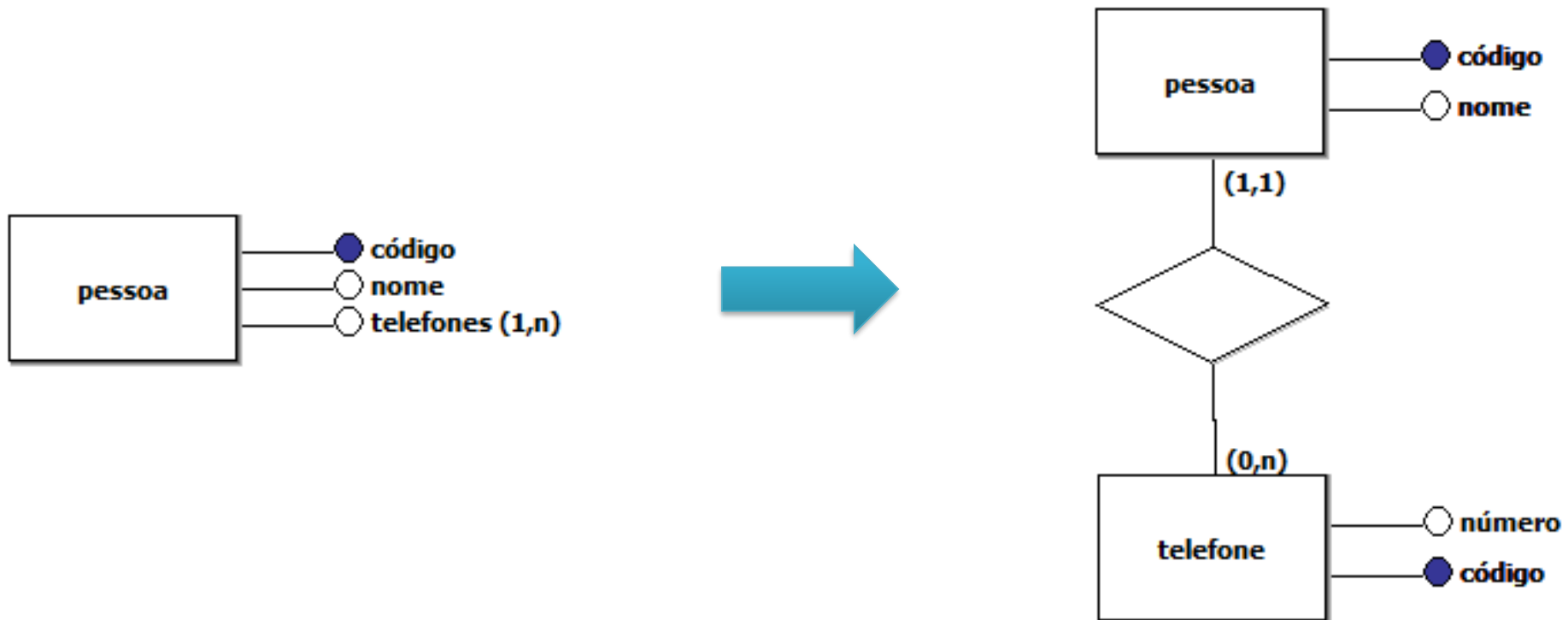
# DICAS PRÁTICAS

---

Já pensar o modelo normalizado!

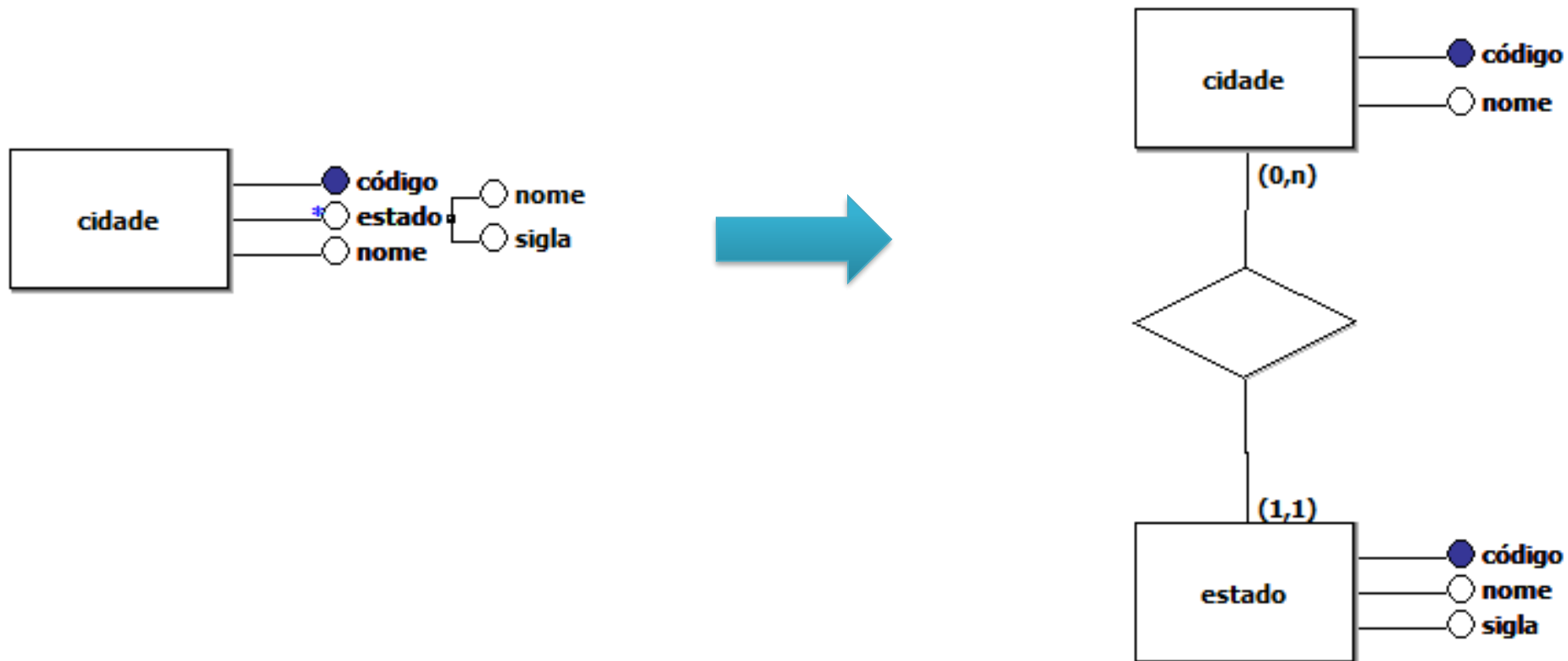
# #DICA 01

- Converta atributos multivalorados em entidades



# #DICA 02

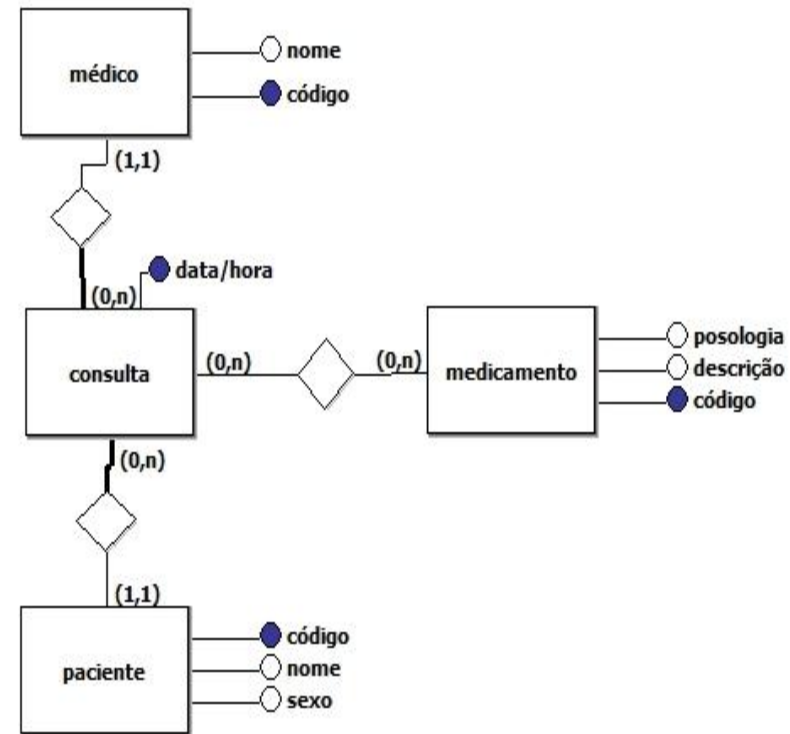
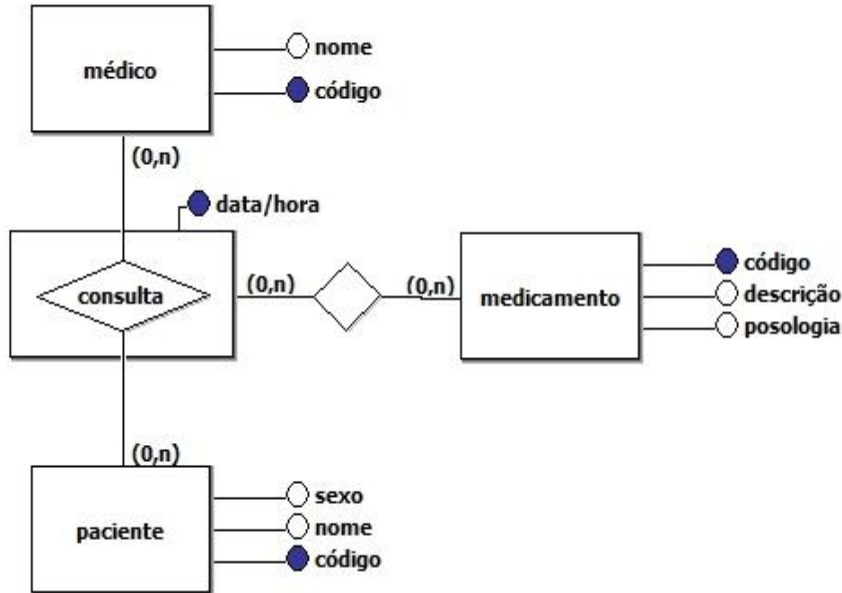
- Converta atributos compostos em entidades





# #DICA 03

- Converta entidades associativas para entidades



# #DICA 04

- Converta atributos em entidades quando há um conjunto de opções de valores possíveis e esse conjunto pode mudar ao longo da vida útil do SW

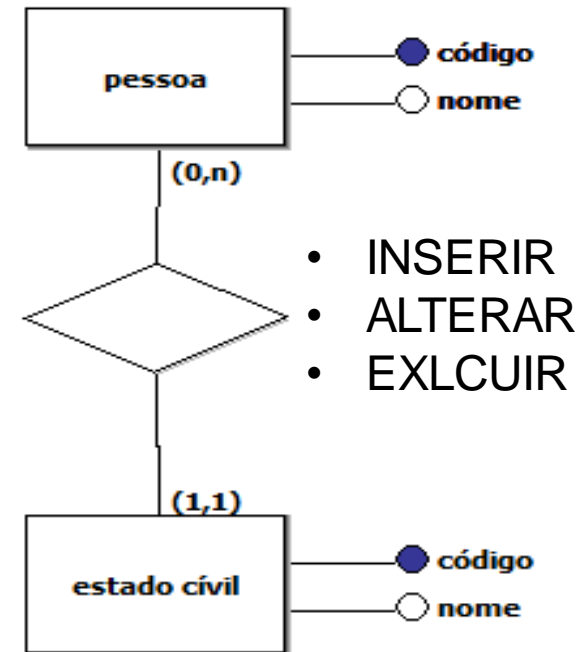
## Opções Fixas



- SOLTEIRO
- CASADO
- SEPARADO
- DIVORCIADO
- VIÚVO

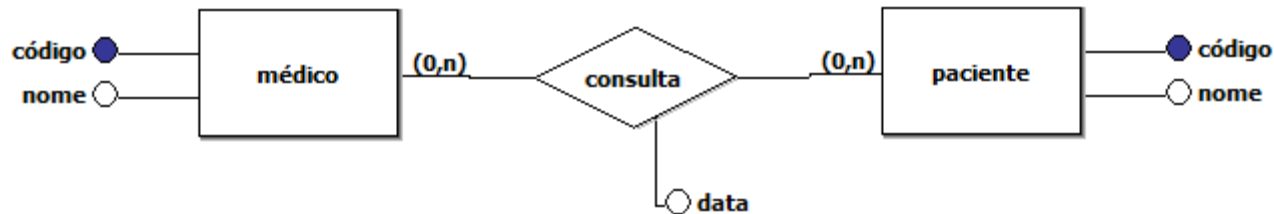
X

## Opções Variáveis

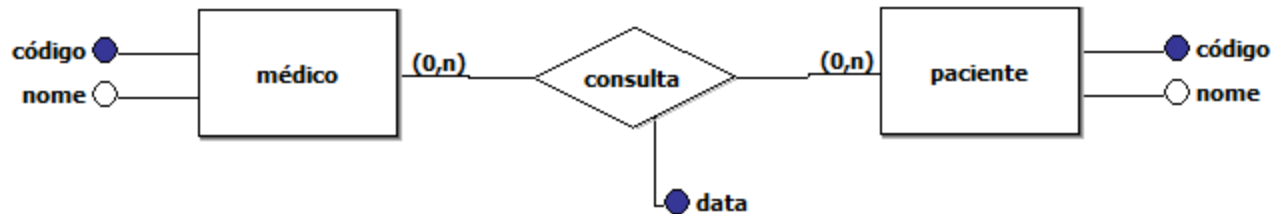


# #DICA 5

- Lembre-se: chaves primárias são únicas
  - Um paciente consulta apenas uma vez com o mesmo médico

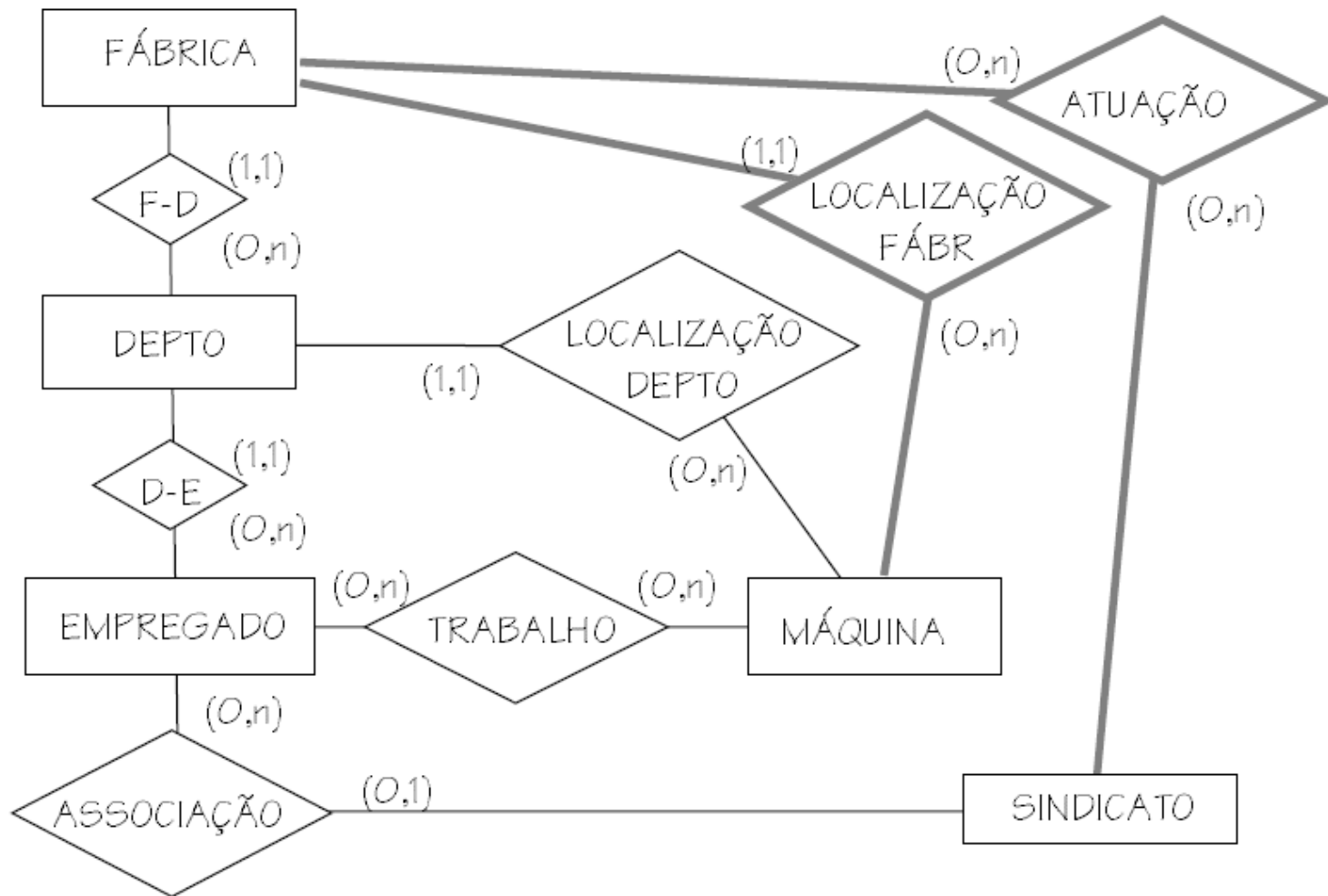


- Um paciente consulta apenas uma vez com o mesmo médico na mesma data



# #DICA 6

- Evite redundâncias nos relacionamentos



# #DICA 7

- Evite redundâncias nos atributos

