

# Normalização

*(Formas Normais – 3FN e 4FN)*

Carina F. Dorneles  
dorneles@inf.ufsc.br

INE5423 - Banco de Dados I

# Passos da Normalização

---

Primeira  
Forma  
Normal  
1FN

Segunda  
Forma  
Normal  
2FN

Terceira  
Forma  
Normal  
3FN

Boyce-  
Codd,  
4FN,  
5FN

DIMINUI A REDUNDÂNCIA DE DADOS

AUMENTA O DESEMPENHO DAS CONSULTAS



# Terceira Forma Normal – 3FN

---

- ▶ Estabelece que:
  - ▶ Deve estar na 2FN
  - ▶ Não deve have **dependência funcional transitiva**



# Dependência funcional ***transitiva***

---

Dependência funcional transitiva:  
**atributo(s) não-chave** depende(m)  
de outro(s) **atributo(s) não-chave**.



# Dependência funcional *transitiva*- exemplo

## Funcionario

<b>CodFunc</b>	<b>Nome</b>	<b>Categoria</b>	<b>Salario</b>
FC001	Luca	C1	800
FC023	Ana	C2	950
FC870	Luca	C2	950
FC445	Jana	C1	800

- ▶ Colunas **Salario** possui dependência *transitiva* com {**Categoria**}
- ▶ **Categoria** → {**Salario**}  
Determinante Dependente

Significa que: para todo valor de {**Salario**} existe o mesmo valor de **Categoria**

# Passagem para 3FN

---

## ▶ Solução:

- ▶ Passo 1: criar uma **nova tabela** com as dependências
  - ▶ Determinante é PK
- ▶ Passo 2: eliminar da **tabela original** o(s) atributo(s) não-chave dependente(s)
  - ▶ Determinante é FK



# Passagem para 3FN - *exemplo*

---

## ► Tabela em 2FN:

**Funcionario**

<b>CodFunc</b>	<b>Nome</b>	<b>Categoria</b>	<b>Salario</b>
FC001	Luca	C1	800
FC023	Ana	C2	950
FC870	Luca	C2	950
FC445	Jana	C1	800

## Dependências:

**Categoria → {Salario}**

- **Passo 1:** nova tabela com as dependências:
- **Passo 2:** eliminar, na tabela original, atributo(s) dependente(s):



# Passagem para 3FN - *exemplo*

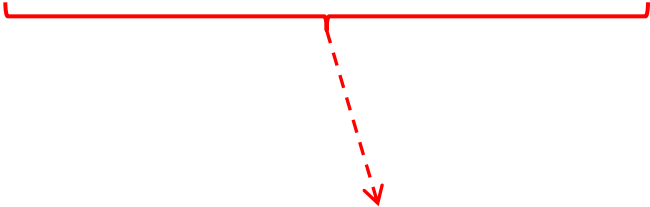
## ► Tabela em 2FN:

**Funcionario**

<b>CodFunc</b>	<b>Nome</b>	<b>Categoria</b>	<b>Salario</b>
FC001	Luca	C1	800
FC023	Ana	C2	950
FC870	Luca	C2	950
FC445	Jana	C1	800

## Dependências:

**Categoria → {Salario}**



<b>Categoria</b>	<b>Salario</b>
C1	800
C2	950

- **Passo 1:** nova tabela com as dependências:
- **Passo 2:** eliminar, na tabela original, atributo(s) dependente(s):



# Passagem para 3FN - *exemplo*

## ► Tabela em 2FN:

**Funcionario**

<b>CodFunc</b>	<b>Nome</b>	<b>Categoria</b>	<b>Salario</b>
FC001	Luca	C1	800
FC023	Ana	C2	950
FC870	Luca	C2	950
FC445	Jana	C1	800

## Dependências:

**Categoria → {Salario}**

► **Passo 1:** nova tabela com as dependências:

<b>Categoria</b>	<b>Salario</b>
C1	800
C2	950

► **Passo 2:** eliminar, na tabela original, atributo(s) dependente(s):

<b>CodFunc</b>	<b>Nome</b>	<b>Categoria</b>
FC001	Luca	C1
FC023	Ana	C2
FC870	Luca	C2
FC445	Jana	C1

# Resumo:

## Passagem para 3FN

---

### ► Tabela em 2FN

Funcionario (CodFunc, nome, categoria, salario)

**Funcionario**

<b>CodFunc</b>	<b>Nome</b>	<b>Categoria</b>	<b>Salario</b>
FC001	Luca	C1	800
FC023	Ana	C2	950
FC870	Luca	C2	950
FC445	Jana	C1	800

### ► Tabela em 3FN

Funcionario (CodFunc, nome, categoria#)

Categoria (Categoria, salario)

**Funcionario**

<b>CodFunc</b>	<b>Nome</b>	<b>Categoria</b>
FC001	Luca	C1
FC023	Ana	C2
FC870	Luca	C2
FC445	Jana	C1

**Categoria**

<b>Categoria</b>	<b>Salario</b>
C1	800
C2	950

# Passos da Normalização

---

Primeira  
Forma  
Normal  
1FN

Segunda  
Forma  
Normal  
2FN

Terceira  
Forma  
Normal  
3FN

Boyce-  
Codd,  
4FN,  
5FN

DIMINUI A REDUNDÂNCIA DE DADOS

AUMENTA O DESEMPENHO DAS CONSULTAS

# Outras Formas Normais

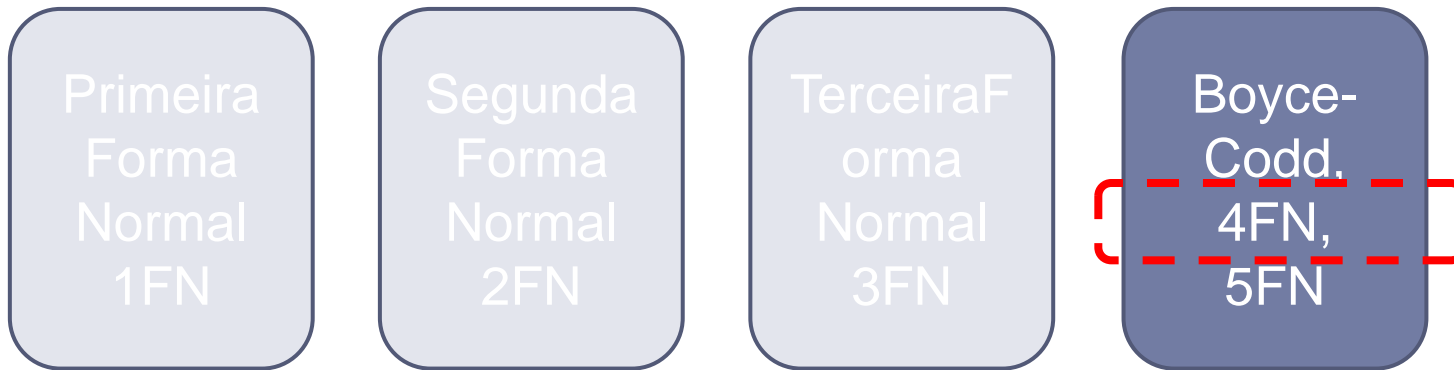
---

- ▶ Boyce/Codd, 4FN e 5FN
  - ▶ Na grande maioria dos casos, normalizar um esquema relacional até a 3FN é suficiente
  - ▶ Mais do que isso, a performance do banco pode despencar na hora da execução de uma consulta



# Passos da Normalização

---



DIMINUI A REDUNDÂNCIA DE DADOS

AUMENTA O DESEMPENHO DAS CONSULTAS



# Quarta Forma Normal – 4FN

---

- ▶ Estabelece que:
  - ▶ Deve estar na 3FN
  - ▶ Não deve have **dependência funcional multivalorada**



# Dependência funcional *multivalorada*?

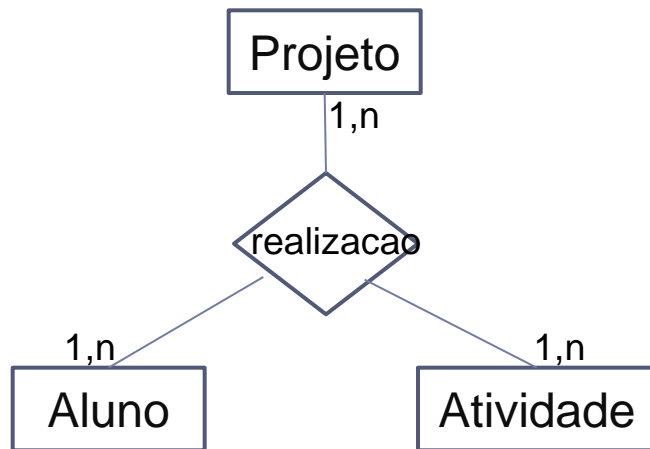
---

Um valor do atributo determinante **identifica repetidas vezes um conjunto** de valores na coluna dependente



# Dependência funcional ***multivalorada*** - *exemplo*

- Implementação de um relacionamento *n-ário*



**Realizacao**

<u>CodProjeto</u>	<u>CodAluno</u>	<u>CodAtividade</u>
1	1	10
1	2	15
1	3	20
2	4	1
1	1	25
1	2	30
1	3	35
2	4	5

**CodProjeto  $\rightarrow \rightarrow$  {CodAluno}**



# Passagem para 4FN

---

## ▶ Solução:

- ▶ Passo 1: criar uma nova tabela com as dependências
  - ▶ Determinante é PK
- ▶ Passo 2: eliminar da tabela original o(s) atributo(s) não-chave dependente(s)



# Passagem para 4FN - *exemplo*

---

## ► Tabela em 3FN:

Realizacao

<u>CodProjeto</u>	<u>CodAluno</u>	<u>CodAtividade</u>
1	1	10
1	2	15
1	3	20
2	4	1
1	1	25
1	2	30
1	3	35
2	4	5

## Dependências:

**CodProjeto** → → {Localizacao}

► **Passo 1:** nova tabela com as dependências:

► **Passo 2:** eliminar, na tabela original, atributo(s) dependente(s):



# Passagem para 4FN - *exemplo*

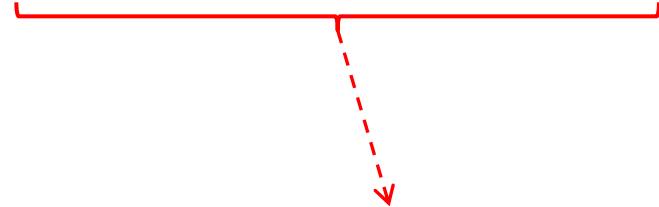
## ► Tabela em 3FN:

Realizacao

<u>CodProjeto</u>	<u>CodAluno</u>	<u>CodAtividade</u>
1	1	10
1	2	15
1	3	20
2	4	1
1	1	25
1	2	30
1	3	35
2	4	5

## Dependências:

**CodProjeto** → → {Localizacao}



<u>CodProjeto</u>	<u>CodAluno</u>
1	1
1	2
1	3
2	4

► **Passo 1:** nova tabela com as dependências:

► **Passo 2:** eliminar, na tabela original, atributo(s) dependente(s):

# Passagem para 4FN - *exemplo*


## ► Tabela em 3FN:

Realizacao

<u>CodProjeto</u>	<u>CodAluno</u>	<u>CodAtividade</u>
1	1	10
1	2	15
1	3	20
2	4	1
1	1	25
1	2	30
1	3	35
2	4	5

## Dependências:

CodProjeto → → {Localizacao}



<u>CodProjeto</u>	<u>CodAluno</u>
1	1
1	2
1	3
2	4

► **Passo 1:** nova tabela com as dependências:

► **Passo 2:** eliminar, na tabela original, atributo(s) dependente(s).

<u>CodProjeto</u>	<u>CodAtividade</u>
1	10
1	15
1	20
2	1
1	25
1	30
1	35
2	5

# Resumo:

## Passagem para 4FN

### ► Tabela em 3FN

Realizacao (CodProjeto#, CodAluno#, CodAtividade#)

Realizacao

<u>CodProjeto</u>	<u>CodAluno</u>	<u>CodAtividade</u>
1	1	10
1	2	15
1	3	20
2	4	1
1	1	25
1	2	30
1	3	35
2	4	5

### ► Tabela em 4FN

Proj\_Aluno (CodProjeto#, CodAluno#)

Localizacao (CodProjeto#, CodAtividade#)

Proj\_Aluno

<u>CodProjeto</u>	<u>CodAluno</u>
1	1
1	2
1	3
2	4

Proj\_Ativi

<u>CodProjeto</u>	<u>CodAtividade</u>
1	10
1	15
1	20
2	1
1	25
1	30
1	35
2	5

# Observação!

## ► Tabela em 3FN

Realizacao (CodProjeto#, CodAluno#, CodAtividade#)

Realizacao

<u>CodProjeto</u>	<u>CodAluno</u>	<u>CodAtividade</u>
1	1	10
1	2	15
1	3	20
2	4	1
1	1	25
1	2	30
1	3	35
2	4	5

## ► Tabela em 4FN

Proj\_Aluno (CodProjeto#, CodAluno#)

Localizacao (CodProjeto#, CodAtividade#)

Proj\_Aluno

<u>CodProjeto</u>	<u>CodAluno</u>
1	1
1	2
1	3
2	4

Proj\_Ativi

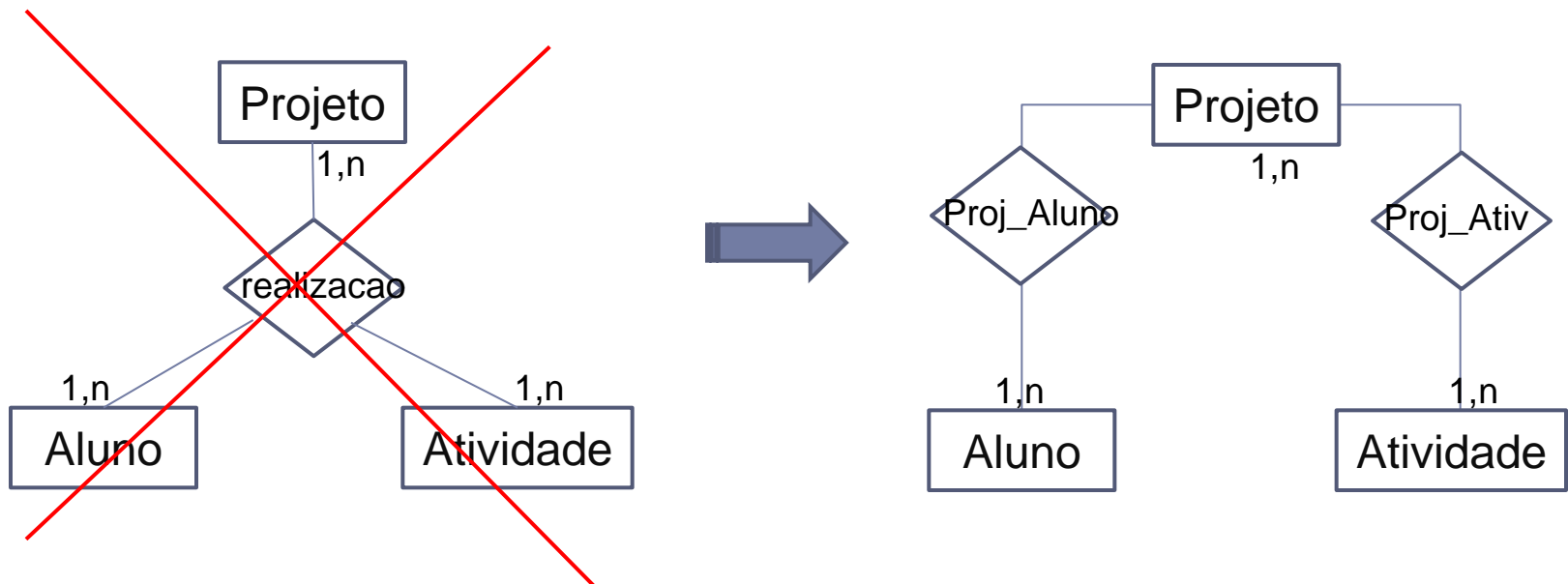
<u>CodProjeto</u>	<u>CodAtividade</u>
1	10
1	15
1	20
2	1
1	25
1	30
1	35
2	5

*Algumas vezes, há perda de informação semântica...*

*Como saber qual aluno executa qual atividade dentro do projeto?*

# 4FN e o Modelo Conceitual

- ▶ Se o modelo conceitual não possuir relacionamentos n-ários, com  $n > 2$ :
  - ▶ *A implementação no BD já estará em 4FN*



# Boyce-Codd e 5FN

---

- ▶ As passagens para Boyce-Codd e 5FN raramente acontecem
- ▶ Consideram
  - ▶ Dependências funcionais
  - ▶ Conceito de chaves primárias candidatas

