

FGA0137

# Sistemas de Banco de Dados 1

Prof. Maurício Serrano

Material original: Prof. Jose Fernando Rodrigues Junior

**2021/2**

# 4<sup>a</sup>. Forma Normal

Módulo 3

## 4<sup>a</sup>. Forma Normal

# 4ª. Forma Normal

- Forma Normal que se baseia em dependência multivalorada
- Exemplo:

Leciona= {Curso, Professor, Livro},

sem nenhuma dependência funcional

<u>Curso</u>	<u>Professor</u>	<u>Livro</u>
FGA0100	Odemir	Programacao
FGA0100	Odemir	Estruturas
FGA0100	Leonardo	Programacao
FGA0100	Leonardo	Estruturas
FGA0200	Odemir	Algebra
FGA0200	Odemir	Calculo
FGA0200	Odemir	Programacao

# 4ª. Forma Normal

- Leciona = {Curso, Professor, Livro}

<u>Curso</u>	<u>Professor</u>	<u>Livro</u>
FGA0100	Odemir	Programacao
FGA0100	Odemir	Estruturas
FGA0100	Leonardo	Programacao
FGA0100	Leonardo	Estruturas
FGA0200	Odemir	Algebra
FGA0200	Odemir	Calculo
FGA0200	Odemir	Programacao

- Não há dependências funcionais triviais:
  - 1ª. FN: Ok
  - 2ª. FN: Ok
  - 3ª. FN: Ok
  - FNBC: Ok

# 4ª. Forma Normal

- Leciona = {Curso, Professor, Livro}

<u>Curso</u>	<u>Professor</u>	<u>Livro</u>
FGA0100	Odemir	Programacao
FGA0100	Odemir	Estruturas

No entanto, ainda há redundância.  
Qual é o problema?

- Não ha dependencias funcionais triviais:
  - 1ª. FN: Ok
  - 2ª. FN: Ok
  - 3ª. FN: Ok
  - FNBC: Ok

# 4ª. Forma Normal

- Leciona = {Curso, Professor, Livro}

<u>Curso</u>	<u>Professor</u>	<u>Livro</u>
FGA0100	Odemir	Programacao
FGA0100	Odemir	Estruturas
FGA0100	Leonardo	Programacao
FGA0100	Leonardo	Estruturas
FGA0200	Odemir	Algebra
FGA0200	Odemir	Calculo
FGA0200	Odemir	Programacao

- Não há dependências funcionais triviais:
  - 1ª. FN: Ok
  - 2ª. FN: Ok
  - 3ª. FN: Ok
  - FNBC: Ok

# 4ª. Forma Normal

- A relação não está na 4ª. Forma Normal
- Não possui dependências funcionais, mas possui dependências multivaloradas

<u>Curso</u>	<u>Professor</u>	<u>Livro</u>
FGA0100	Odemir	Programacao
FGA0100	Odemir	Estruturas
FGA0100	Leonardo	Programacao
FGA0100	Leonardo	Estruturas
FGA0200	Odemir	Algebra
FGA0200	Odemir	Calculo
FGA0200	Odemir	Programacao

**Se Curso = FGA0100 então Livro = {Programacao, Estruturas}  
diz-se que Curso  $\twoheadrightarrow$  Livro**

Portanto, se Curso = FGA0100 **então** há duas tuplas para cada professor que ministra a disciplina  $\rightarrow$  redundância



# 4ª. Forma Normal

- Curso → Livro

Se **Curso = FGA0200** então **Livro = {Algebra, Calculo, Programacao}**

Portanto, se Curso = FGA0200 **então** há três tuplas para cada professor que ministra a disciplina → redundância

<u>Curso</u>	<u>Professor</u>	<u>Livro</u>
FGA0100	Odemir	Programacao
FGA0100	Odemir	Estruturas
FGA0100	Leonardo	Programacao
FGA0100	Leonardo	Estruturas
FGA0200	Odemir	Algebra
FGA0200	Odemir	Calculo
FGA0200	Odemir	Programacao

# 4ª. Forma Normal

- Simetria no atributo professor
- Se  $\text{Curso} \rightarrow \rightarrow \text{Livro}$  e professor é redundante **então**  
 $\text{Curso} \rightarrow \rightarrow \text{Professor}$ ; e Livro também é redundante
- De fato:

**Se Curso = FGA0100 então Professor = {Odemir, Leonardo}**

Portanto, se Curso = FGA0100 então há duas tuplas para cada livro usado na disciplina

<u>Curso</u>	<u>Professor</u>	<u>Livro</u>
FGA0100	Odemir	Programacao
FGA0100	Odemir	Estruturas
FGA0100	Leonardo	Programacao
FGA0100	Leonardo	Estruturas
FGA0200	Odemir	Algebra
FGA0200	Odemir	Calculo
FGA0200	Odemir	Programacao

# 4ª. Forma Normal

- **Definição [dependência multivalorada]:** seja R um esquema de relação e sejam X, Y, e Z subconjuntos dos atributos de R. Diz-se que a dependência multivalorada  $X \twoheadrightarrow Y$  é válida sobre R se
  - em cada instância válida r de R, cada valor de X está associado a um **conjunto de valores** de Y; e este conjunto é independente dos valores de quaisquer outros atributos.
  - em cada instância válida r de R, cada valor de X está associado a um **conjunto de valores** de Z; e este conjunto é independente dos valores de quaisquer outros atributos.

Diz-se que  $X \twoheadrightarrow Y$  e, por simetria,  $X \twoheadrightarrow Z$ , tal que  $Z = R - (X \cup Y)$ ; exemplo:

<u>Curso (X)</u>	<u>Professor (Z)</u>	<u>Livro (Y)</u>
FGA0100	Odemir	Programacao
FGA0100	Odemir	Estruturas
FGA0100	Leonardo	Programacao
FGA0100	Leonardo	Estruturas
FGA0200	Odemir	Algebra
FGA0200	Odemir	Calculo
FGA0200	Odemir	Programacao

# 4ª. Forma Normal

- **4ª. Forma Normal:** uma relação  $R$  está na quarta forma normal se para todas as dependências multivaloradas não triviais  $A \twoheadrightarrow B$ ,  $A$  é uma chave candidata (além de já satisfazer a FNBC)
- No exemplo da relação Leciona, a 4ª. FN é violada pois Curso é primo, mas não é chave
- **Normalização:** a normalização segue substituindo-se  $R$  por duas relações, constituídas pelos seguintes atributos:
  - $A \cup B$
  - $R - B$
- **Atenção:** após a normalização, não há mais dep. Multivalorada nem na rel. original e nem nas rels. resultantes

# 4ª. Forma Normal

- **4ª. Forma Normal:** uma relação R está na quarta forma normal se para todas as dependências multivaloradas  $A \twoheadrightarrow B$ , A é uma chave candidata

- No exemplo, a dependência multivalorada é violada pois C não é uma chave candidata

- **Normalização:** se R possui dependências multivaloradas, criando-se as seguintes relações:

- A U
- R – B

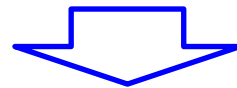
- **Atenção:** após a normalização, não há mais dep. Multivalorada nem na rel. original e nem nas rels. resultantes

A violação da 4ª. Forma Normal traz dois problemas:

- 1) Redundância
- 2) Maior dificuldade em se manter a consistência, pois a manipulação de uma tupla implica na necessidade de manipulação de outras n tuplas

# 4ª. Forma Normal

<u>Curso</u>	<u>Professor</u>	<u>Livro</u>
FGA0100	Odemir	Programacao
FGA0100	Odemir	Estruturas
FGA0100	Leonardo	Programacao
FGA0100	Leonardo	Estruturas
FGA0200	Odemir	Algebra
FGA0200	Odemir	Calculo
FGA0200	Odemir	Programacao



<u>Curso</u>	<u>Professor</u>
FGA0100	Odemir
FGA0100	Leonardo
FGA0200	Odemir

<u>Curso</u>	<u>Livro</u>
FGA0100	Programacao
FGA0100	Estruturas
FGA0200	Algebra
FGA0200	Calculo
FGA0200	Programacao

# 4ª. Forma Normal

- Exercício:

<u>Cargo</u>	<u>Funcao</u>	<u>Projeto</u>
Engenheiro	CalculoEstrutural	Alpha
Engenheiro	GarantiaDeQualidade	Alpha
Engenheiro	SupervisaoDeObras	Alpha
Engenheiro	CalculoEstrutural	Beta
Engenheiro	GarantiaDeQualidade	Beta
Engenheiro	SupervisaoDeObras	Beta
Desenvolvedor	ArquiteturaDeSoftware	Alpha
Desenvolvedor	Codificação	Alpha
Desenvolvedor	Teste	Alpha
Desenvolvedor	Backup	Alpha

\*supõe-se que um cargo sempre tem um conjunto fixo de funções

- Qual é a dependência multivalorada? Justifique com exemplos? Qual é a dependência multivalorada simétrica? Exemplifique. Como fica a normalização?

# 4ª. Forma Normal

- Resposta:
  - A dep multivalorada é  $\text{Cargo} \rightarrow \rightarrow \text{Funcao}$  pois, por exemplo, se  $\text{Cargo} = \text{Engenheiro}$  então  $\text{Funcao} = \{\text{CalculoEstrutural}, \text{GarantiaDeQualidade}, \text{SupervisaoDeObras}\}$  e se  $\text{Cargo} = \text{Desenvolvedor}$  então  $\text{Funcao} = \{\text{ArquiteturaDeSoftware}, \text{Codificação}, \text{Teste}, \text{Backup}\}$
  - A dep. Multivalorada simétrica é dada por  $Z = R - A \cup B$ , isto é,  $\{\text{Cargo}, \text{Funcao}, \text{Projeto}\} - \{\text{Cargo} \cup \text{Funcao}\} = \text{Projeto}$ . Portanto,  $\text{Cargo} \rightarrow \rightarrow \text{Projeto}$  pois, por exemplo, se  $\text{Cargo} = \text{Engenheiro}$  então  $\text{Projeto} = \{\text{Alpha}, \text{Beta}\}$
  - A normalização é possível criando-se duas relações
    - $A \cup B \rightarrow \text{Funcoes} = \{\text{Cargo}, \text{Funcao}\}$
    - $R - B \rightarrow \text{Projetos} = \{\text{Cargo}, \text{Projeto}\}$



# Síntese e Considerações Finais

# Síntese

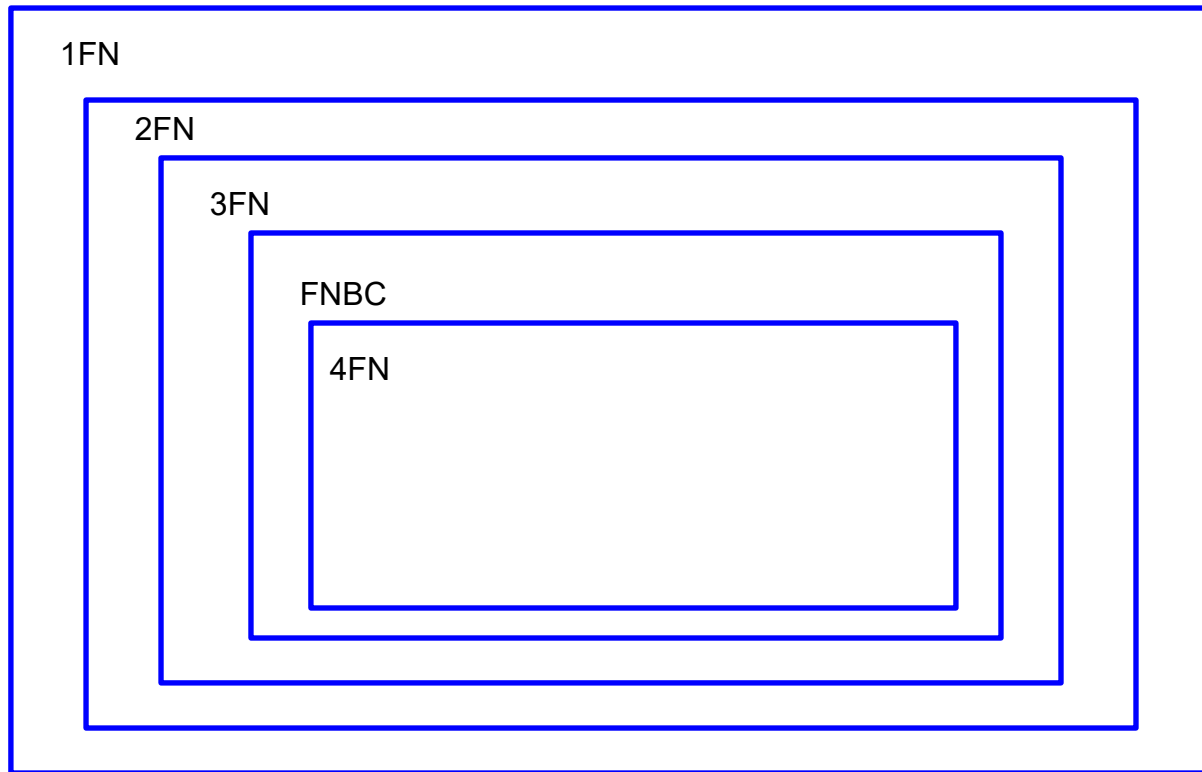
- Dependência funcional
  - Se  $A_1, A_2, \dots, A_n \rightarrow B_1, B_2, \dots, B_n$
- Formais normais

Forma Normal	Característica*
1a. FN	Atomicidade, monovaloração (não há relações aninhadas)
2a. FN	1a. FN e ausência de dependências parciais
3a. FN	2a.FN e ausência de dependências transitivas
Boyce Codd	A esquerda de toda df é chave candidata
4a.FN	A esquerda de toda dm é chave candidata

\* desconsideram-se dfs triviais, isto é,  $X \rightarrow Y$  tal que  $Y \subseteq X$

- Normalização: **decomposição** de relações
- Eliminam redundância → **previnem anomalias** de inserção, atualização, remoção, e junção

# Formais Normais – Visão Geral



# Quando usar normalização

- Quando usar normalização?
  - Resposta: **sempre**
- Quando aplicá-la ao meu projeto?
  - modo geral: **quantidade de operações de leitura** não é elevada
    - Ex.: bases operacionais podem ter alto grau de normalização
  - Diferentes **graus de normalização** em função do sistema
    - Ex.: data warehouses têm baixo grau de normalização (até nenhuma normalização)

# Quando usar normalização

- Mas se eu não vou aplicar a normalização, então porque vou usá-la?
- Identificação de possibilidades de **melhoria de desempenho**
- Verificar se o projeto **está correto** de acordo com a semântica

# Um pouco de história

- Edgar F. Codd (1923 - 2003)
  - Britânico
  - IBM, Nova York
  - Definiu o Modelo Relacional
  - Curiosidade: concorrentes da IBM começaram a usar as idéias de Codd, antes que a própria IBM
  - Com Raymond F. Boyce, propôs a Forma normal de Boyce-Codd

