

---

# Banco de Dados

## Normalização para Bancos de Dados Relacionais

DCC-UFLA

Prof. Denilson Alves Pereira

[denilsonpereira@dcc.ufla.br](mailto:denilsonpereira@dcc.ufla.br)

<http://lattes.cnpq.br/4120230814124499>

# Conteúdo

---

- Introdução
- Diretrizes de Projeto
- Dependências Funcionais
- Formas Normais e Normalização

# Introdução

---

- Objetivo

avaliar a qualidade do projeto de esquemas de relações por meio de medidas formais

- Dois níveis de boas práticas de esquema

- ◆ Nível lógico (ou conceitual)

como interpretar o esquema de relações e o significado de seus atributos

- ◆ Nível de implementação (ou armazenamento físico)

como as tuplas são armazenadas e atualizadas

# Introdução

---

## ■ Objetivos implícitos

### ◆ Preservação da informação

manter os conceitos capturados originalmente no projeto conceitual

### ◆ Redundância mínima

- diminuir o armazenamento redundante da mesma informação
- reduzir a necessidade de múltiplas atualizações

## ■ Dependência Funcional

principal ferramenta para medir formalmente a adequação dos agrupamentos de atributo em esquemas de relação

## ■ Normalização

série de testes, em relações, para atender a determinados requisitos e decompor as relações quando necessário

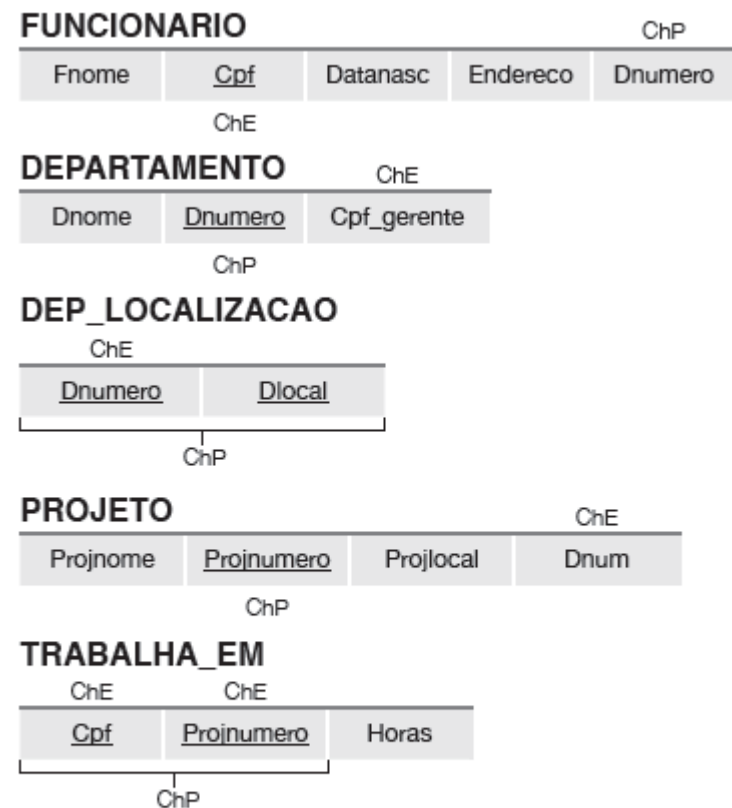
# Diretrizes de Projeto

---

- Diretrizes informais para medir a qualidade de um projeto
  - ◆ Garantir que a semântica dos atributos seja clara no esquema
  - ◆ Reduzir a informação redundante nas tuplas
  - ◆ Reduzir os valores NULL nas tuplas
  - ◆ Reprovar a possibilidade de gerar tuplas falsas

# Semântica dos Atributos nas Relações

- Semântica dos atributos
  - ◆ O projeto do esquema deve ter um significado claro
  - ◆ Deve ser fácil de explicar
- Práticas ruins:
  - ◆ Flags
  - ◆ Nomes ambíguos
  - ◆ União de relações semanticamente distintas

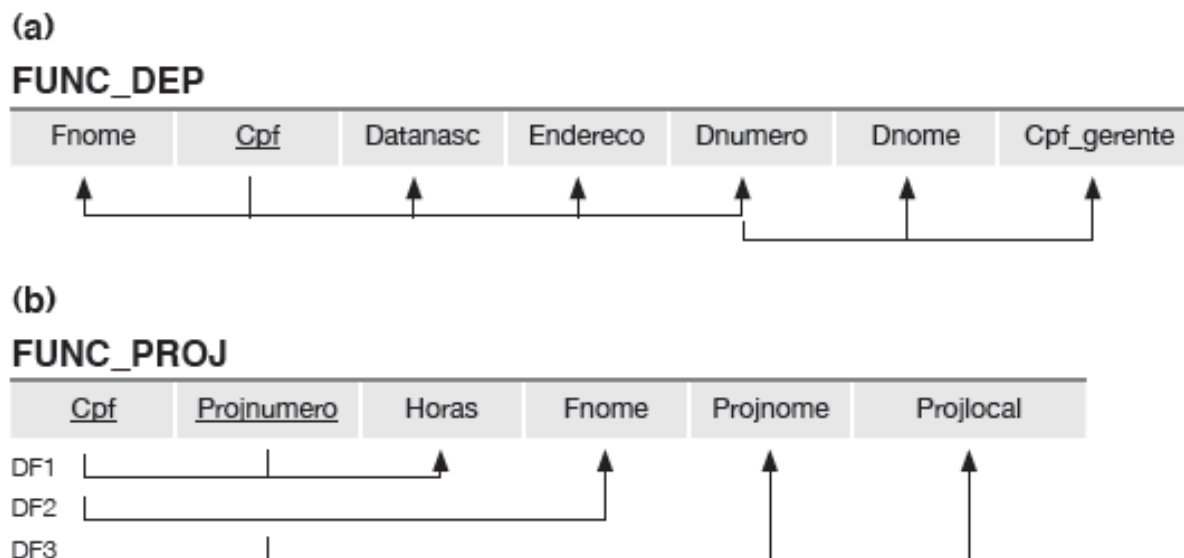


# Semântica dos Atributos nas Relações

## ■ Diretriz 1

- ◆ Projete um esquema de relação de modo que seja fácil explicar seu significado
- ◆ Não combine atributos de vários tipos de entidade e de relacionamento em uma única relação

## ■ Exemplo de violação da Diretriz 1



# Informação Redundante nas Tuplas

- O espaço de armazenamento usado pelas relações deve ser minimizado
- Compare os dados tabela abaixo com os utilizados nas tabelas FUNCIONARIO e DEPARTAMENTO anteriores

FUNC_DEP					Redundância	
Fnome	Cpf	Datanasc	Endereco	Dnumero	Dnome	Cpf_gerente
Silva, João B.	12345678966	09-01-1965	Rua das Flores, 751, São Paulo, SP	5	Pesquisa	33344555587
Wong, Fernando T.	33344555587	08-12-1955	Rua da Lapa, 34, São Paulo, SP	5	Pesquisa	33344555587
Zelaya, Alice J.	99988777767	19-01-1968	Rua Souza Lima, 35, Curitiba, PR	4	Administração	98765432168
Souza, Jennifer S.	98765432168	20-06-1941	Av. Arthur de Lima, 54, Santo André, SP	4	Administração	98765432168
Lima, Ronaldo K.	66688444476	15-09-1962	Rua Rebouças, 65, Piracicaba, SP	5	Pesquisa	33344555587
Leite, Joioe A.	45345345376	31-07-1972	Av. Lucas Obes, 74, São Paulo, SP	5	Pesquisa	33344555587
Pereira, André V.	98798798733	29-03-1969	Rua Timbira, 35, São Paulo, SP	4	Administração	98765432168
Brito, Jorge E.	88866555576	10-11-1937	Rua do Horto, 35, São Paulo, SP	1	Matriz	88866555576

- O armazenamento de informações redundantes leva às **anomalias de atualização**



# Informação Redundante nas Tuplas

---

## ■ Anomalias de Inserção

- ◆ Para inserir uma nova tupla em FUNC\_DEP, os dados do departamento (Dnome e Cpf\_gerente) devem ser coerentes com os valores do departamento nas outras tuplas
- ◆ Como inserir um novo departamento que ainda não tem funcionário trabalhando pra ele?

## ■ Anomalias de exclusão

- ◆ Em FUNC\_DEP, quando o último funcionário de um departamento for excluído, o departamento também é excluído

## ■ Anomalias de modificação

- ◆ Em FUNC\_DEP, se o gerente de um departamento for modificado, todas as tuplas de funcionários do mesmo departamento devem ser também atualizadas

# Informação Redundante nas Tuplas

---

## ■ Diretriz 2

Projete esquemas de relação de modo a evitar as anomalias de inserção, exclusão e modificação

- Às vezes, as diretrizes podem ter de ser violadas a fim de melhorar o desempenho de certas consultas
  - ◆ O tratamento da consistência deve ser feito pelos programas de aplicação ou por meio de *triggers* e *store procedures*
- Em geral, é aconselhável usar relações de base sem anomalias e especificar **visões (views)** que incluem as junções para reunir os atributos frequentemente referenciados nas consultas importantes

# Valores NULL nas Tuplas

---

- NULL significa valor desconhecido
  - ◆ Não pode ser comparado a outros valores
- Pode ter várias interpretações
  - ◆ O atributo não se aplica à tupla
  - ◆ O valor é realmente desconhecido
  - ◆ O valor é conhecido, mas ainda não foi registrado
- Desperdiça espaço no nível de armazenamento
- Ocasiona problemas com a especificação de operações de junção
- Complica as consultas devido à interpretação do seu significado

# Valores NULL nas Tuplas

---

## ■ Diretriz 3

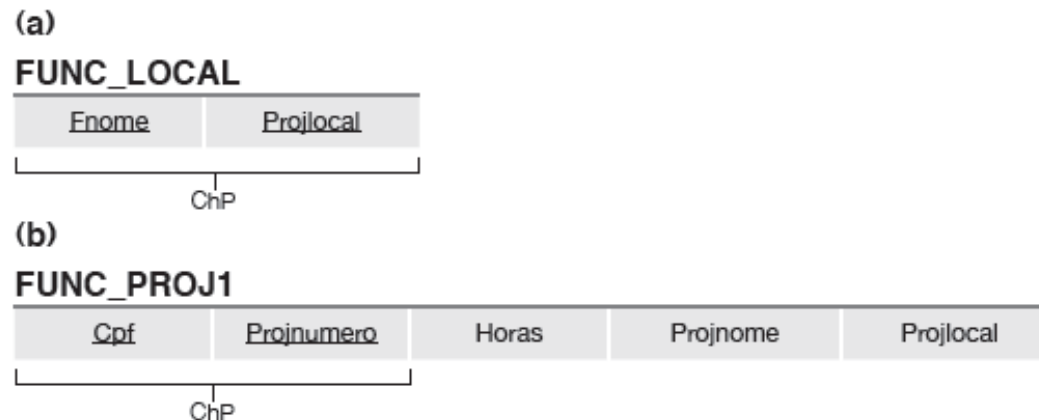
Evite colocar atributos em uma relação cujos valores podem ser NULL com frequência

## ■ Exemplo de como evitar valores nulos:

Se apenas 15% dos funcionários têm um número de escritório, em vez de criar um atributo NumEscritorio na relação FUNCIONARIO, crie uma relação FUNC\_ESCRITORIO (Func\_CPF, NumEscritorio) para incluir tuplas apenas para funcionários com escritórios

# Geração de Tuplas Falsas

- Considere os esquemas de relações abaixo



- Considere essas relações povoadas com os dados do próximo slide
- Se for aplicada uma junção natural entre essas relações, o resultado é mostrado no slide seguinte
  - ◆ O resultado produz tuplas falsas (marcadas com \*)
  - ◆ ProjLocal não é chave primária nem chave estrangeira

# Geração de Tuplas Falsas

**FUNC\_LOCAL**

Fnome	Projlocal
Silva, João B.	Santo André
Silva, João B.	Itu
Lima, Ronaldo K.	São Paulo
Leite, Joice A.	Santo André
Leite, Joice A.	Itu
Wong, Fernando T.	Itu
Wong, Fernando T.	São Paulo
Wong, Fernando T.	Mauá
Zelaya, Alice J.	Mauá
Pereira, André V.	Mauá
Souza, Jennifer S.	Mauá
Souza, Jennifer S.	São Paulo
Brito, Jorge E.	São Paulo

**FUNC\_PROJ1**

Cpf	Projnumero	Horas	Projnome	Projlocal
12345678966	1	32,5	ProdutoX	Santo André
12345678966	2	7,5	ProdutoY	Itu
66688444476	3	40,0	ProdutoZ	São Paulo
45345345376	1	20,0	ProdutoX	Santo André
45345345376	2	20,0	ProdutoY	Itu
33344555587	2	10,0	ProdutoY	Itu
33344555587	3	10,0	ProdutoZ	São Paulo
33344555587	10	10,0	Computadorização	Mauá
33344555587	20	10,0	Reorganização	São Paulo
99988777767	30	30,0	Novosbenefícios	Mauá
99988777767	10	10,0	Computadorização	Mauá
98765432168	10	35,0	Computadorização	Mauá
98765432168	30	5,0	Novosbenefícios	Mauá
98765432168	30	20,0	Novosbenefícios	Mauá
98798798733	20	15,0	Reorganização	São Paulo
88866555576	20	NULL	Reorganização	São Paulo

# Geração de Tuplas Falsas

Cpf	Projnumero	Horas	Projnome	Projlocal	Fnome
12345678966	1	32,5	ProdutoX	Santo André	Silva, João B.
*12345678966	1	32,5	ProdutoX	Santo André	Leite, Joice A.
12345678966	2	7,5	ProdutoY	Itu	Silva, João B.
*12345678966	2	7,5	ProdutoY	Itu	Leite, Joice A.
*12345678966	2	7,5	ProdutoY	Itu	Wong, Fernando T.
66688444476	3	40,0	ProdutoZ	São Paulo	Lima, Ronaldo K.
*66688444476	3	40,0	ProdutoZ	São Paulo	Wong, Fernando T.
*45345345376	1	20,0	ProdutoX	Santo André	Silva, João B.
45345345376	1	20,0	ProdutoX	Santo André	Leite, Joice A.
*45345345376	2	20,0	ProdutoY	Itu	Silva, João B.
45345345376	2	20,0	ProdutoY	Itu	Leite, Joice A.
*45345345376	2	20,0	ProdutoY	Itu	Wong, Fernando T.
*33344555587	2	10,0	ProdutoY	Itu	Silva, João B.
*33344555587	2	10,0	ProdutoY	Itu	Leite, Joice A.
33344555587	2	10,0	ProdutoY	Itu	Wong, Fernando T.
*33344555587	3	10,0	ProdutoZ	São Paulo	Lima, Ronaldo K.
33344555587	3	10,0	ProdutoZ	São Paulo	Wong, Fernando T.
33344555587	10	10,0	Computadorização	Mauá	Wong, Fernando T.
*33344555587	20	10,0	Reorganização	São Paulo	Lima, Ronaldo K.
33344555587	20	10,0	Reorganização	São Paulo	Wong, Fernando T.

⋮

# Geração de Tuplas Falsas

---

## ■ Diretriz 4

Projete esquemas de relação de modo que possam ser unidos com condições de igualdade sobre os atributos que são pares relacionados corretamente (chave primária, chave estrangeira) de modo a garantir que nenhuma tupla falsa será gerada



# Dependência Funcional

---

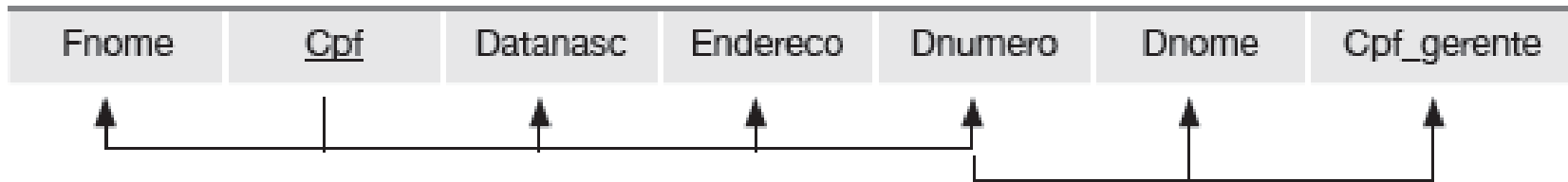
- Uma **Dependência Funcional**, indicada por  $X \rightarrow Y$ , entre dois conjuntos de atributos  $X$  e  $Y$  especifica uma restrição sobre possíveis tuplas de uma relação, onde para cada valor em  $X$  aparece **sempre** o mesmo valor em  $Y$ 
  - ♦ Ao se conhecer o valor em  $X$  pode-se sempre determinar o valor de  $Y$
  - ♦ Lê-se “ $X$  determina  $Y$ ” ou “ $Y$  depende funcionalmente de  $X$ ”
- Exemplos:
  - ♦  $\{\text{cpf}\} \rightarrow \{\text{nomeFunc}\}$
  - ♦  $\{\text{codProduto}\} \rightarrow \{\text{descricaoProduto}, \text{preço}\}$
  - ♦  $\{\text{cpf}, \text{projNumero}\} \rightarrow \{\text{horas}\}$

# Dependência Funcional

- Uma chave candidata (X) em uma relação R sempre determina qualquer subconjunto de atributos (Y) de R
- Notação diagramática para dependência funcional (DF):

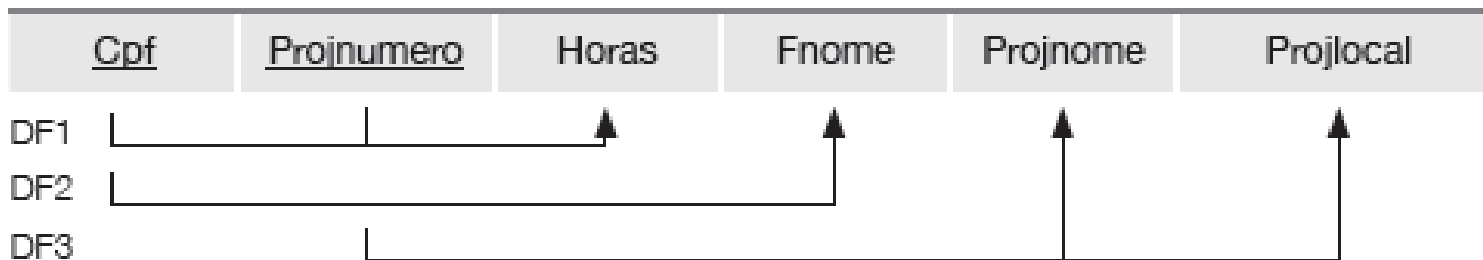
(a)

**FUNC\_DEP**



(b)

**FUNC\_PROJ**



# Normalização de Dados

---

- O objetivo de um projeto de banco de dados é obter um conjunto de esquemas de relação que nos permita armazenar dados sem redundância e que as informações possam ser geradas facilmente
- Normalização de Dados
  - ◆ Processo de análise de esquemas de relação para:
    - minimizar redundâncias e
    - minimizar anomalias de inserção, exclusão e modificação
  - ◆ Proposto por Edgar F. Codd em 1972
  - ◆ Composto por uma série de testes para certificar se o esquema satisfaz certa **forma normal**
  - ◆ Uma relação pode ser decomposta para atender os critérios da forma normal

# Normalização de Dados

---

## ■ Forma Normal

- ◆ Indica o grau ao qual uma relação foi normalizada, ou seja, define o número de regras que foram impostas para manter o esquema de relação consistente
- ◆ Nomes das formas normais:
  - Primeira Forma Normal (1FN)
  - Segunda Forma Normal (2FN)
  - Terceira Forma Normal (3FN)
  - Forma Normal de Boyce-Codd (FNBC)
  - Quarta Forma Normal (4FN)
  - Quinta Forma Normal (5FN)

## ■ Desnormalização

Processo de manipular as relações que estão em Formas Normais mais altas para Formas Normais mais baixas

# Primeira Forma Normal (1FN)

---

- Uma relação está na **Primeira Forma Normal (1FN)** se seus atributos incluam apenas valores atômicos (simples, indivisíveis) e se o valor de qualquer atributo em uma tupla for um *único valor* do domínio desse atributo.
  - ◆ Não são permitidos atributos multivalorados, atributos compostos e suas combinações
  - ◆ Não são permitidas “relações aninhadas”

# Primeira Forma Normal (1FN)

DEPARTAMENTO

Dnome	<u>Dnumero</u>	Cpf_gerente	Dlocal

Diagram showing functional dependencies: Dnome → Dnumero, Dnumero → Cpf\_gerente, and Dnumero → Dlocal (indicated by a dashed line).

1FN

DEPARTAMENTO

Dnome	<u>Dnumero</u>	Cpf_gerente
-------	----------------	-------------

ChE

ChP

DEP\_LOCALIZACAO

ChE

<u>Dnumero</u>	<u>Dlocal</u>
----------------	---------------

ChP

DEPARTAMENTO

Dnome	<u>Dnumero</u>	Cpf_gerente	Dlocal
Pesquisa	5	33344555587	Santo André, Itu, São Paulo
Administração	4	98765432168	Mauá
Matriz	1	88866555576	São Paulo

1FN

DEPARTAMENTO

Dnome	<u>Dnumero</u>	Cpf_gerente
Pesquisa	5	33344555587
Administração	4	98765432168
Matriz	1	88866555576

LOCALIZACAO\_DEP

<u>Dnumero</u>	<u>Dlocal</u>
1	São Paulo
4	Mauá
5	Santo André
5	Itu
5	São Paulo

# Primeira Forma Normal (1FN)

- A solução abaixo também está na 1FN, embora contenha redundância

## DEPARTAMENTO

Dnome	<u>Dnumero</u>	Cpf_gerente	<u>Dlocal</u>
Pesquisa	5	33344555587	Santo André
Pesquisa	5	33344555587	Itu
Pesquisa	5	33344555587	São Paulo
Administração	4	98765432168	Mauá
Matriz	1	88866555576	São Paulo

# Primeira Forma Normal (1FN)

- Relação aninhada (representada por chaves)

FUNC\_PROJ (Cpf, Fnome, {Projnumero, Horas})

FUNC\_PROJ

Cpf	Fnome	Projnumero	Horas
12345678966	Silva, João B.	1	32,5
		2	7,5
66688444476	Lima, Ronaldo K.	3	40,0
45345345376	Leite, Joice A.	1	20,0
		2	20,0
33344555587	Wong, Fernando T.	2	10,0
		3	10,0
		10	10,0
		20	10,0

- Deve ser decomposta em (1FN):

FUNC\_PROJ1 (Cpf, Fnome)

FUNC\_PROJ2 (Cpf, Projnumero, Horas)



# Segunda Forma Normal (2FN)

---

## ■ Dependência Funcional Total

$X \rightarrow Y$  é uma dependência funcional total se a remoção de qualquer atributo  $A \in X$  significar que  $X - \{A\}$  não determina  $Y$  funcionalmente

## ■ Dependência Funcional Parcial

$X \rightarrow Y$  é uma dependência funcional parcial se algum atributo  $A \in X$  puder ser removido de  $X$  e a dependência  $X \rightarrow Y$  ainda se mantiver

## ■ Atributo Principal

- ◆ Atributo membro de alguma chave candidata de uma relação
- ◆ Um atributo é chamado de **não principal** se não for um atributo principal

# Segunda Forma Normal (2FN)

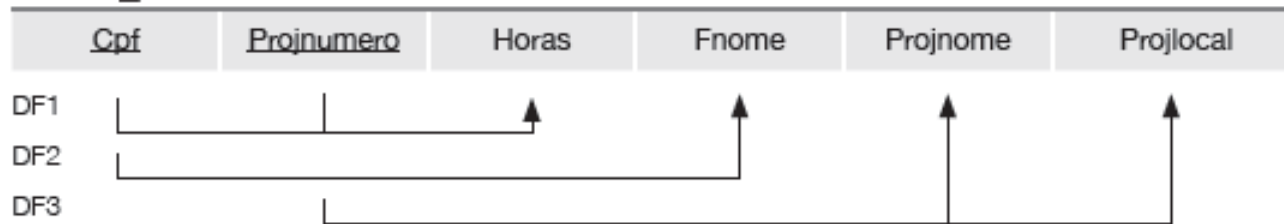
---

- Uma relação R está na **Segunda Forma Normal (2FN)** se estiver na 1FN e nenhum atributo não principal de R possuir Dependência Funcional Parcial em relação a qualquer chave candidata (primária ou secundária)
- Para normalizar para a 2FN:
  - ◆ Certifique-se que a relação está na 1FN
    - Se não existe chave candidata composta, a relação já se encontra na 2FN
  - ◆ Crie uma nova relação para cada chave parcial com seu conjunto de atributos dependentes, certificando-se que nessa nova relação é mantida a dependência funcional total

# Segunda Forma Normal (2FN)

## Exemplo 1:

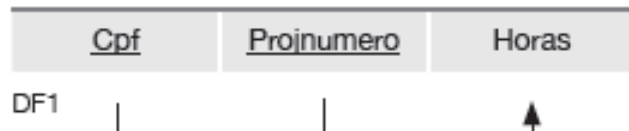
### FUNC\_PROJ



### Normalizacao 2FN

2FN

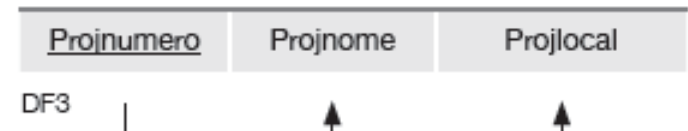
### FP1



### FP2

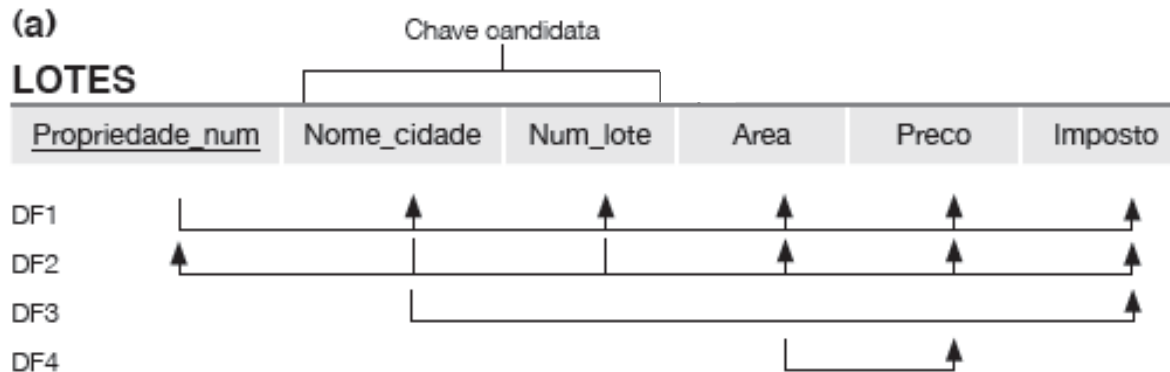


### FP3



# Segunda Forma Normal (2FN)

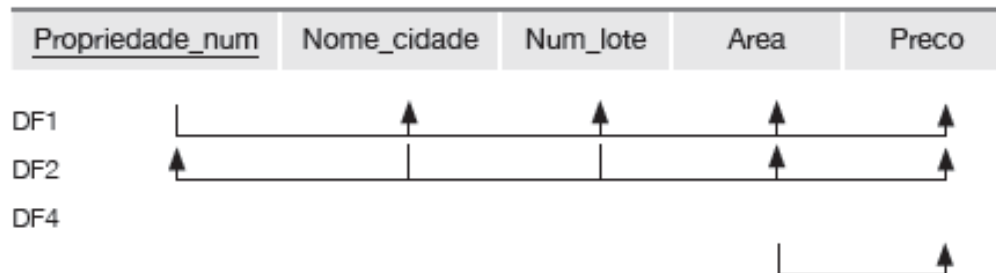
## Exemplo 2:



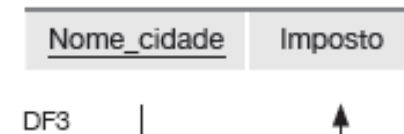
2FN

(b)

**LOTES1**



**LOTES2**



# Terceira Forma Normal (3FN)

## ■ Dependência Transitiva

- ♦ Uma dependência funcional  $X \rightarrow Y$  em um esquema de relação  $R$  é uma dependência transitiva se houver um conjunto de atributos  $Z$  em  $R$  que não seja uma chave candidata e nem um subconjunto de qualquer chave de  $R$ , e tanto  $X \rightarrow Z$  quanto  $Z \rightarrow Y$  se mantiverem
- ♦ Exemplo:

**FUNC\_DEP**



$Cpf \rightarrow Dnumero$  e  $Dnumero \rightarrow Dnome$

Logo:  $Cpf \rightarrow Dnome$  (dependência transitiva)

# Terceira Forma Normal (3FN)

---

- Uma relação R está na **Terceira Forma Normal (3FN)** se estiver na 2FN e nenhum atributo não principal de R for transitivamente dependente de alguma chave candidata (primária ou secundária)
  - ◆ em outras palavras, nenhum atributo não principal de R pode determinar outro atributo não principal
- Para normalizar para a 3FN:
  - ◆ Certifique-se que a relação está na 2FN
  - ◆ Crie uma nova relação que inclua o conjunto de atributos não chave que determina funcionalmente outro conjunto de atributos não chave

# Terceira Forma Normal (3FN)

## Exemplo 1:

**FUNC\_DEP**

Fnome	<u>Cpf</u>	Datanasc	Endereco	Dnumero	Dnome	Cpf_gerente
-------	------------	----------	----------	---------	-------	-------------

**Normalizacao 3FN**

3FN

**DF1**

Fnome	<u>Cpf</u>	Datanasc	Endereco	Dnumero
-------	------------	----------	----------	---------

**DF2**

<u>Dnumero</u>	Dnome	Cpf_gerente
----------------	-------	-------------

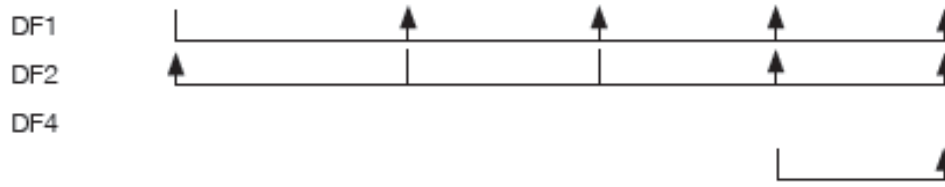
# Terceira Forma Normal (3FN)

## Exemplo 2:

(b)

**LOTES1**

<u>Propriedade_num</u>	Nome_cidade	Num_lote	Area	Preco
------------------------	-------------	----------	------	-------



**LOTES2**

<u>Nome_cidade</u>	Imposto
--------------------	---------



3FN

(c)

**LOTES1A**

<u>Propriedade_num</u>	Nome_cidade	Num_lote	Area
------------------------	-------------	----------	------



**LOTES1B**

<u>Area</u>	Preco
-------------	-------





# Forma Normal de Boyce-Codd (FNBC)

---

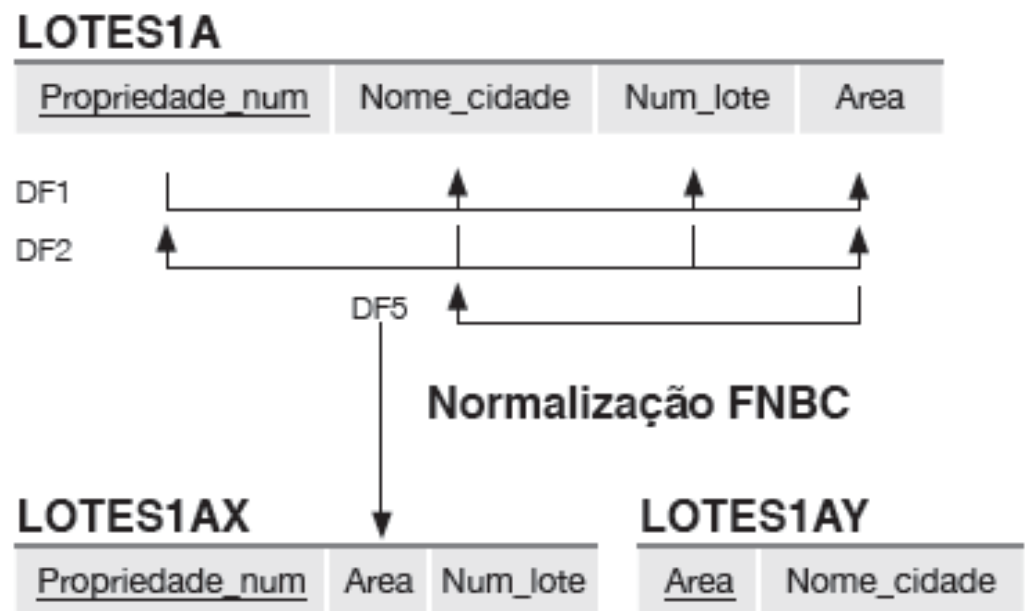
- Uma relação  $R$  está na **Forma Normal de Boyce-Codd (FNBC)** se para toda dependência funcional  $X \rightarrow A$ ,  $X$  é uma chave candidata (primária ou secundária) de  $R$ 
  - ♦ em outras palavras, nenhum atributo não principal de  $R$  pode determinar outro atributo (principal ou não principal)
  - ♦ é mais forte do que a 3FN, pois a 3FN trata apenas das dependências envolvendo atributos não principais
- Para normalizar para a FNBC
  - ♦ Certifique-se que a relação está na 3FN
  - ♦ Decomponha a relação de forma a eliminar as dependências

# Forma Normal de Boyce-Codd (FNBC)

## Exemplo 1:

- Suponha que lotes sejam vendidos em apenas duas cidades: Lavras e Perdões
- E ainda: os lotes em Lavras têm tamanhos 300 ou 450m<sup>2</sup>, e os lotes em Perdões têm tamanhos 200 ou 400 m<sup>2</sup>
- Desta forma, tem-se a Dependência Funcional

Area → Nome\_cidade



# Forma Normal de Boyce-Codd (FNBC)

## ■ Exemplo 2:

**ENSINA**

Aluno	Disciplina	Professor
Lima	Banco de dados	Marcos
Silva	Banco de dados	Navathe
Silva	Sistemas operacionais	Omar
Silva	Teoria	Charles
Souza	Banco de dados	Marcos
Souza	Sistemas operacionais	Antonio
Wong	Banco de dados	Gomes
Zelaya	Banco de dados	Navathe
Lima	Sistemas operacionais	Omar

DF1: {Aluno, Disciplina} → Professor

DF2: Professor → Disciplina

❖ Na FNBC, a relação de ser decomposta em:

R1 (Professor, Disciplina)

R2 (Professor, Aluno)

# Quarta Forma Normal (4FN)

---

## ■ Dependência Multivalorada (MVD)

- ♦ Uma **Dependência Multivalorada** ocorre quando, para cada valor de um atributo A, há um conjunto de valores para outros atributos B e C (independentes entre si) que estão associados a A
  - Ocorre quando se tem dois ou mais atributos multivalorados independentes no mesmo esquema de relação
  - É necessário repetir cada valor de um dos atributos com cada valor do outro atributo
- ♦ Relações com MVD's tendem a ter sua chave composta de todos os seus atributos

# Quarta Forma Normal (4FN)

## ■ Dependência Multivalorada (MVD)

### ◆ Exemplo:

**FUNC**

<u>Enome</u>	<u>Projnome</u>	<u>Dnome</u>
Silva	X	João
Silva	Y	Ana
Silva	X	Ana
Silva	Y	João

- Chave primária composta por todos os atributos
- O esquema FUNC está na FNBC

# Quarta Forma Normal (4FN)

---

- Uma relação R está na **Quarta Forma Normal (4FN)** se estiver na FNBC e não possuir Dependência Multivalorada
- Para normalizar para a 4FN
  - ◆ Certifique-se que a relação está na FNBC
  - ◆ Decomponha a relação de modo que cada MVD seja representada por uma relação separada

# Quarta Forma Normal (4FN)

## Exemplo

**FUNC**

<u>Fnome</u>	<u>Projnome</u>	<u>Dnome</u>
Silva	X	João
Silva	Y	Ana
Silva	X	Ana
Silva	Y	João

4FN

**FUNC\_PROJETOS**

<u>Fnome</u>	<u>Projnome</u>
Silva	X
Silva	Y

**FUNC\_DEPENDENTES**

<u>Fnome</u>	<u>Nome_dependente</u>
Silva	João
Silva	Ana

# Quinta Forma Normal (5FN)

---

## ■ Dependência de Junção

- ♦ Uma **Dependência de Junção**, especificada em um esquema de relação  $R$ , determina uma restrição sobre os estados  $r$  de  $R$ , indicando que a relação pode ser decomposta em  $n$  ( $n > 2$ ) outras relações e juntadas (em uma operação de junção) sem geração de tuplas falsas
- ♦ A Dependência Multivalorada é um caso particular de Dependência de Junção em que  $n = 2$



# Quinta Forma Normal (5FN)

---

- Uma relação R está na **Quinta Forma Normal (5FN)** se estiver na 4FN e não existir Dependência de Junção
- Para normalizar para a 5FN
  - ◆ Certifique-se que a relação está na 4FN
  - ◆ Certifique-se que não existe Dependência de Junção
    - Se existir, as relações resultantes da decomposição estarão na 5FN
- Dependências de junção são muito difíceis de serem detectadas na prática. Assim, a normalização para a 5FN é raramente verificada em projetos de banco de dados

# Quinta Forma Normal (5FN)

- Exemplo 1: relação Fornece **não** está na 5FN

**FORNECE**

<u>Nome_fornece</u>	<u>Nome_peca</u>	<u>Nome_proj</u>
Silva	Peneira	ProjX
Silva	Poroa	ProjY
Adam	Peneira	ProjY
Walter	Poroa	ProjZ
Adam	Prego	ProjX
Adam	Peneira	ProjX
Silva	Peneira	ProjY

} Se estas tuplas existirem

5FN

**R1**

<u>Nome_fornece</u>	<u>Nome_peca</u>
Silva	Peneira
Silva	Poroa
Adam	Peneira
Walter	Poroa
Adam	Prego

**R2**

<u>Nome_fornece</u>	<u>OR_proj</u>
Silva	ProjX
Silva	ProjY
Adam	ProjY
Walter	ProjZ
Adam	ProjX

**R3**

<u>Nome_peca</u>	<u>OR_proj</u>
Peneira	ProjX
Poroa	ProjY
Peneira	ProjY
Poroa	ProjZ
Prego	ProjX

# Quinta Forma Normal (5FN)

## Exemplo 2: relação Fornece **está** na 5FN

### FORNECE

<u>Nome_fornece</u>	<u>Nome_peca</u>	<u>Nome_proj</u>
Silva	Peneira	ProjX
Silva	Poroa	ProjY
Adam	Peneira	ProjY
Walter	Poroa	ProjZ
Adam	Prego	ProjX

<u>Nome_fornece</u>	<u>Nome_peca</u>	<u>Nome_proj</u>
Silva	Peneira	ProjX
Silva	Poroa	ProjY
Adam	Peneira	ProjY
Walter	Poroa	ProjZ
Adam	Prego	ProjX
Adam	Peneira	ProjX
Silva	Peneira	ProjY

tuplas falsas

Decomposição

Junção Natural

### R1

<u>Nome_fornece</u>	<u>Nome_peca</u>
Silva	Peneira
Silva	Poroa
Adam	Peneira
Walter	Poroa
Adam	Prego

### R2

<u>Nome_fornece</u>	<u>OR_proj</u>
Silva	ProjX
Silva	ProjY
Adam	ProjY
Walter	ProjZ
Adam	ProjX

### R3

<u>Nome_peca</u>	<u>OR_proj</u>
Peneira	ProjX
Poroa	ProjY
Peneira	ProjY
Poroa	ProjZ
Prego	ProjX

# Bibliografia Básica

---

ELMASRI, Ramez; NAVATHE, Shamkant B. Sistemas de Bancos de Dados. Pearson Education, 6ª edição, 2011. ISBN-978-85-7936-085-5, Capítulo 15