



# Dependência Funcional e Normalização

Profa. Maria Camila Nardini Barioni  
[camila.barioni@ufu.br](mailto:camila.barioni@ufu.br)

Bloco B - sala 1B137

2º semestre de 2023

# Avisos sobre as próximas aulas

- ◆ Estarei afastada na próxima semana para participar de uma banca de concurso
- ◆ Como fica a programação das aulas
  - 12/09 → TDE. Será disponibilizada uma lista de exercícios para ser entregue pelo MS Teams (a entrega vale presença)
  - 13/09
    - ◆ Nos dois primeiros horários → Prof. Humberto irá aplicar a prova
    - ◆ Nos dois últimos horários → TDE. Será disponibilizada uma atividade para ser entregue pelo MS Teams (a entrega vale presença)

The slide features a decorative design of thin blue lines. A vertical line on the left and a horizontal line at the top intersect at the top-left corner, marked with a small blue quarter-circle. Another horizontal line is positioned below the title. A vertical line on the right and a horizontal line at the bottom intersect at the bottom-right corner, also marked with a small blue quarter-circle.

# Normalização

Relembrando e continuando...

# Relembrando: Primeira Forma Normal (1FN)

- ◆ Uma relação R está na 1FN se:
  - todo valor em R for atômico e monovalorado
    - ◆ ou seja, R não contém grupos de repetição

# Relembrando: Segunda Forma Normal (2FN)

- ◆ Uma relação R está na 2FN se:
  - está na 1FN
  - não existe atributo não chave que é dependente de somente uma parte da chave
    - ◆ ou seja, não pode existir dependência funcional parcial
  - houver **dependência funcional total**

# Terceira Forma Normal (3FN)

◆ Uma relação R está na 3FN se:

- está na 2FN
- não existem atributos não chave que sejam dependentes de outros atributos não chave (determinante não chave)
  - **dependência transitiva**

◆ Dependência transitiva  $X \rightarrow Y$  em R

- se  $X \rightarrow Z$  e  $Z \rightarrow Y$  e Z não for nem a chave candidata nem um subconjunto de qualquer chave de R

# Terceira Forma Normal (3FN)

Uma relação está na **3ª FN** se já estiver na 2ª e ...

**TODOS** os atributos que **NÃO** fazem parte da chave primária **NÃO** possuírem nenhuma dependência entre si.

OU SEJA,

Na 2ª FN verifica-se a dependência em relação aos atributos que fazem parte da CHAVE PRIMÁRIA, enquanto que ....

Na 3ª FN verifica-se a dependência em relação aos atributos que **NÃO** fazem parte da CHAVE PRIMÁRIA

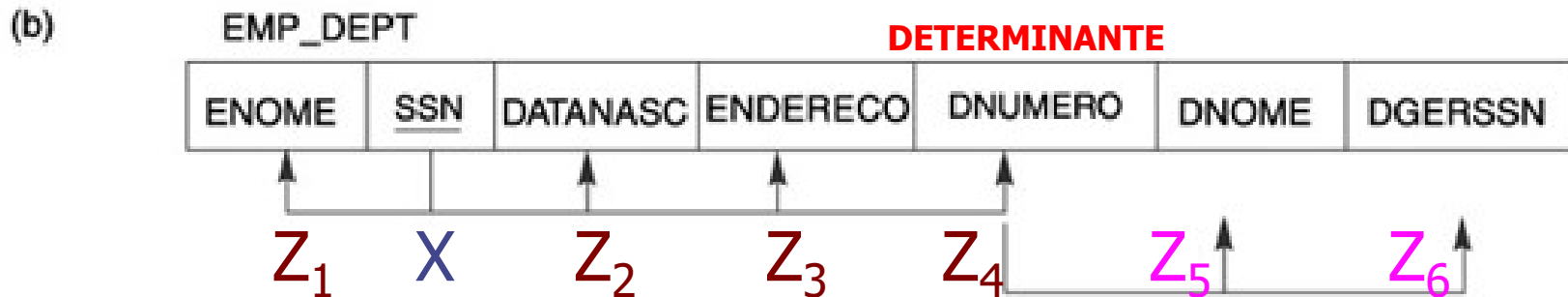
# Terceira Forma Normal (3FN)

- ◆ Dependência transitiva  $X \rightarrow Y$  em R
  - se  $X \rightarrow Z$  e  $Z \rightarrow Y$  e  $Z$  não for nem a chave candidata nem um subconjunto de qualquer chave de R



# Terceira Forma Normal (3FN)

## ◆ Exemplo de dependência transitiva



- DNOME e DGERSSN dependem funcionalmente de DNUMERO ( $Z_4 \rightarrow \{Z_5, Z_6\} \Leftrightarrow Z \rightarrow Y$ )
- DNUMERO depende funcionalmente de SSN ( $X \rightarrow \{Z_1, \dots, Z_4\}$ )
  - ◆ DNUMERO não é chave, nem parte de chave
- DNOME e DGERSSN dependem transitivamente de SSN
  - ◆  $X \rightarrow Y$

# Terceira Forma Normal (3FN)

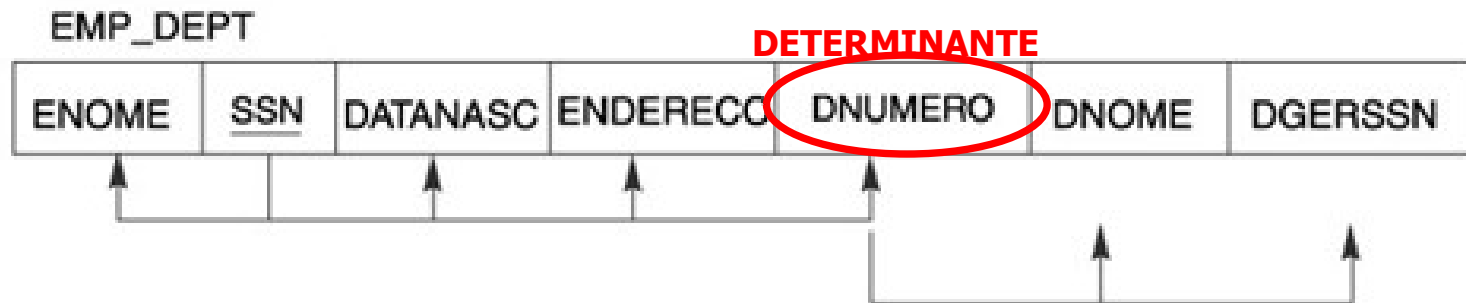
## ◆ Método para corrigir o problema:

- para cada **determinante** que não é uma chave candidata, remover da relação os atributos que dependem desse determinante
- criar uma nova relação contendo todos os atributos da relação original que dependem desse determinante
- tornar o determinante a chave primária da nova relação

# Terceira Forma Normal (3FN)

## ◆ Exemplo 1:

(b)



NORMALIZAÇÃO 3FN



# Terceira Forma Normal (3FN)

## ◆ Exemplo 2:

- cliente (nro-cliente, nome-cliente, end-cliente, nro-vendedor, nome-vendedor)

X

Z

nro-cliente → nome-cliente, end-cliente,  
nro\_vendedor

Z<sub>3</sub>

Y

nro-vendedor → nome\_vendedor

# Terceira Forma Normal (3FN)

- ◆ Problema: cliente (nro-cliente, nome-cliente, end-cliente, nro-vendedor, nome-vendedor)

*Corrigindo o problema ...*

- ◆ Solução:  
cliente (nro-cliente, nome-cliente, end-cliente, nro-vendedor)  
vendedor (nro-vendedor, nome-vendedor)

# Definições Genéricas

## ◆ Segunda forma normal

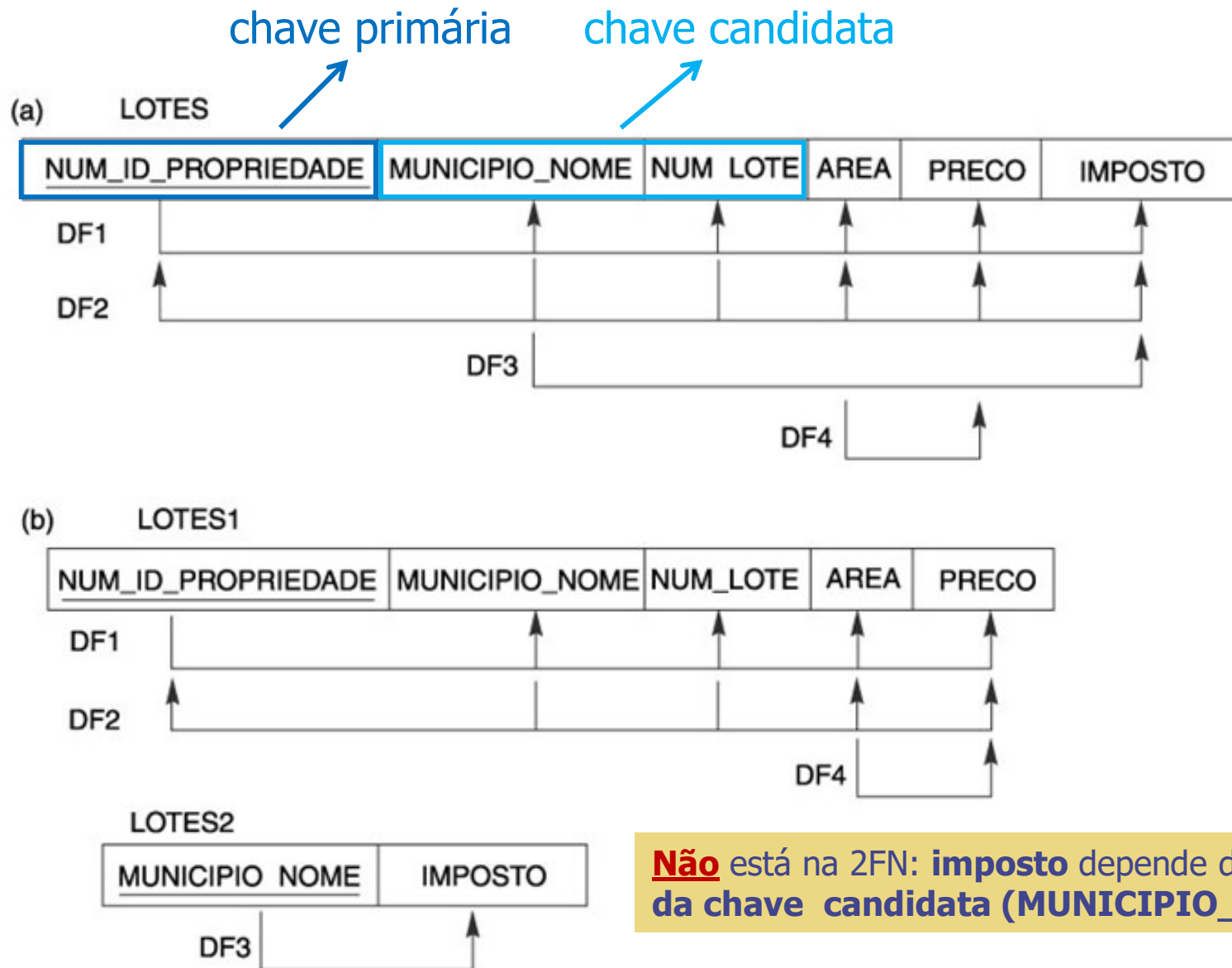
- um esquema de relação  $R$  está na 2FN se cada atributo não primário de  $R$  não for parcialmente dependente de nenhuma chave de  $R$  ... além da chave primária, candidatas

## ◆ Terceira forma normal

- um esquema de relação  $R$  está na 3FN se para cada dependência funcional  $X \rightarrow A$ ,  $X$  é uma superchave de  $R$  ou  $A$  é um atributo primário de  $R$

# Definições Genéricas

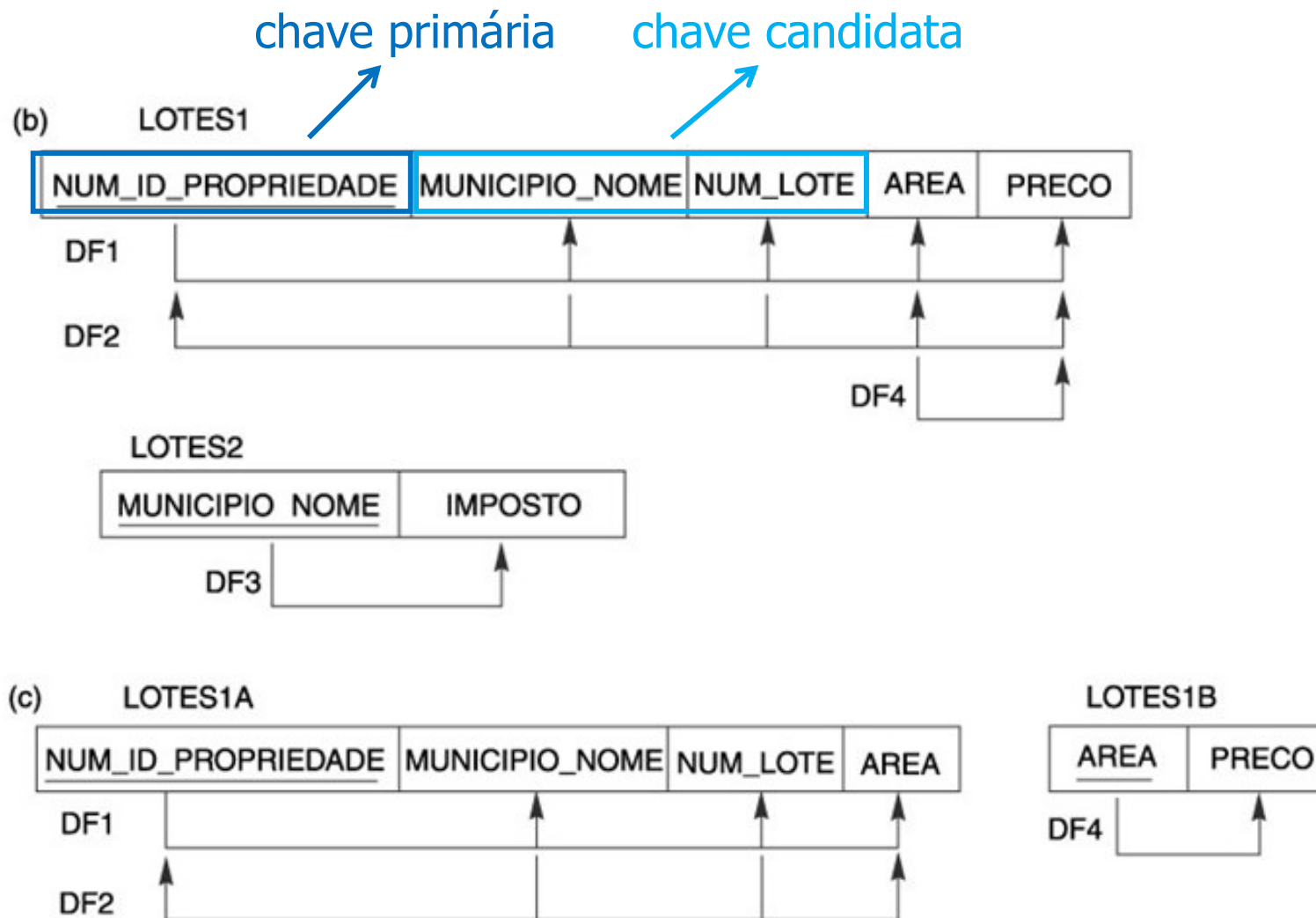
## Segunda forma normal



**Não** está na 2FN: **imposto** depende de **parte** da **chave candidata (MUNICIPIO\_NOME)**

# Definições Genéricas

## Terceira forma normal



**Não** está na 3FN: **PRECO** depende de **AREA** (ambos não chave)



# Forma Normal de Boyce-Codd

## ◆ BCNF

- um esquema de relação R está na BCNF se para cada dependência funcional  $X \rightarrow A$ , X é uma superchave de R

## ◆ BCNF e 3FN

- relação está na BCNF  $\rightarrow$  relação está na 3FN
- relação está na 3FN  ~~$\rightarrow$~~  relação está na BCNF

## ◆ Prática

- maioria dos esquemas de relação que está na 3FN também está na BCNF

# Forma Normal de Boyce-Codd

## ◆ BCNF

- se  $X \rightarrow A$  então
  - ◆  $X$  é superchave

## ◆ 3FN

- se  $X \rightarrow A$  então
  - ◆  $X$  é superchave ou  $A$  é atributo primário

# Exemplo

- ◆ Considere o esquema LOTES, que descreve lotes à venda em vários municípios
- ◆ Considere as chaves NUM\_ID\_PROP e {MUNICIPIO\_NOME, NUM\_LOTE}
  - NUM\_LOTE diferentes apenas dentro do município
  - NUM\_ID\_PROP diferentes entre municípios

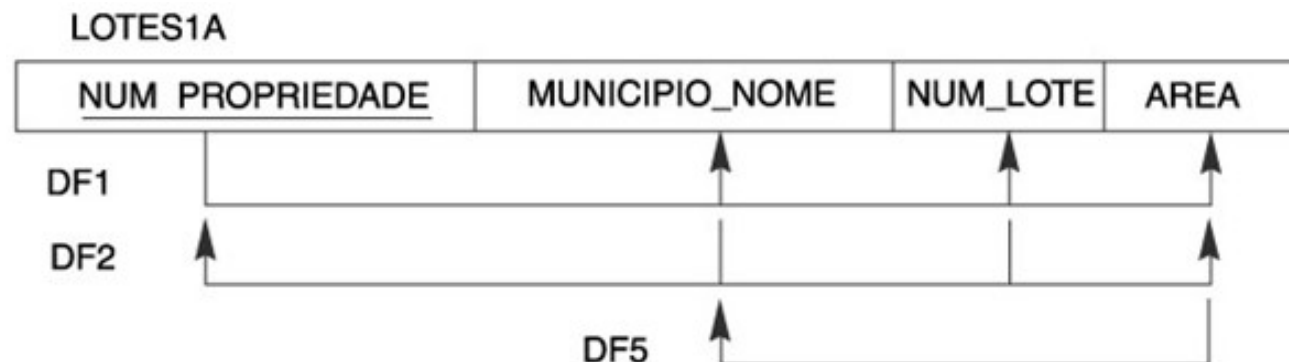


# Exemplo

- ◆ Consideremos que há milhares de lotes, mas apenas nos municípios de Uberlândia e Araguari
  - em Uberlândia só existem lotes com 100, 200 e 300 m<sup>2</sup>
  - em Araguari só existem lotes com 150, 250 e 350 m<sup>2</sup>

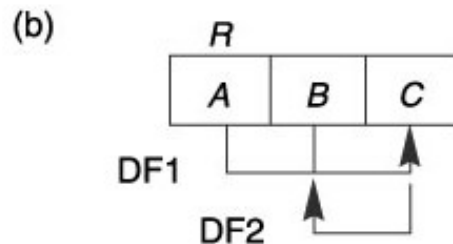
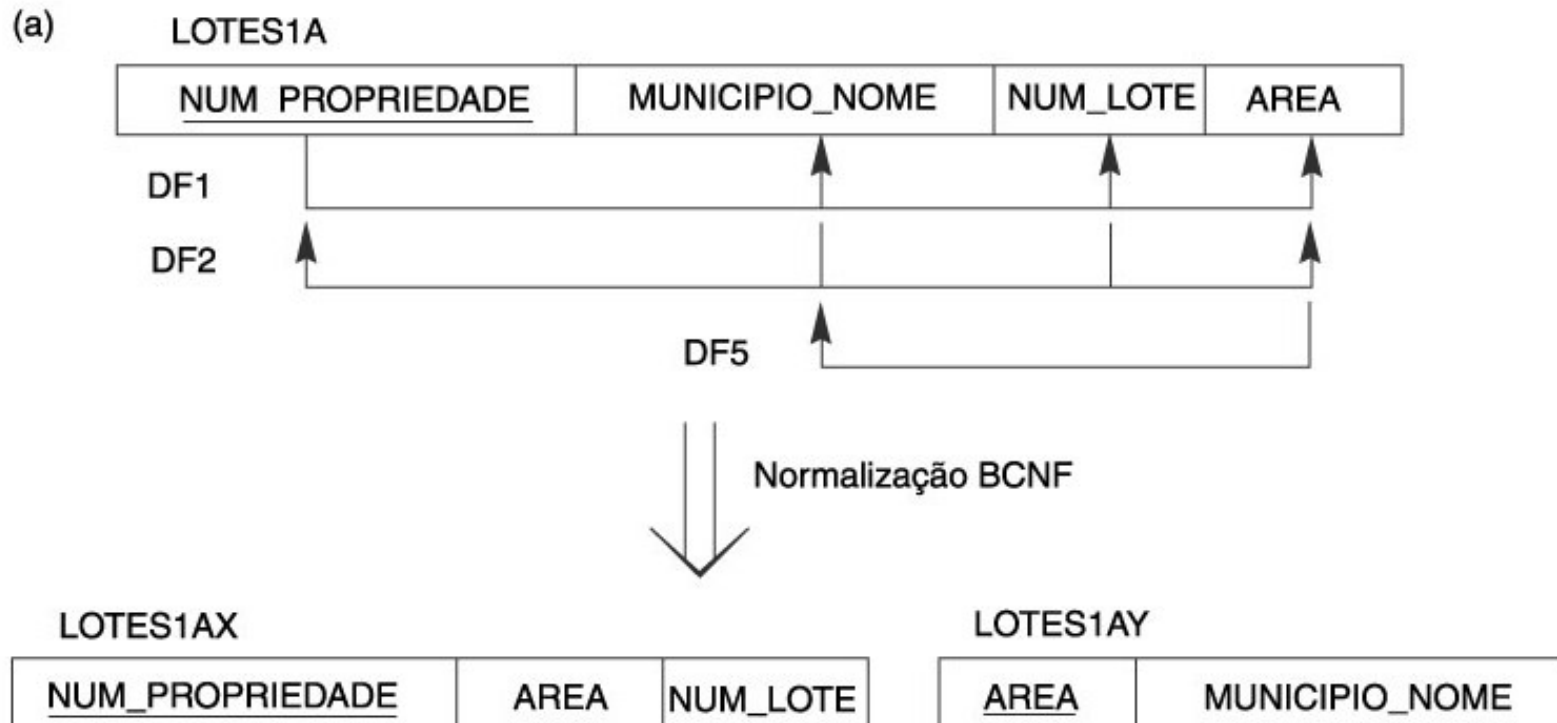
## ◆ Nesse caso

- DF5: AREA → MUNICIPIO\_NOME



# Forma Normal de Boyce-Codd

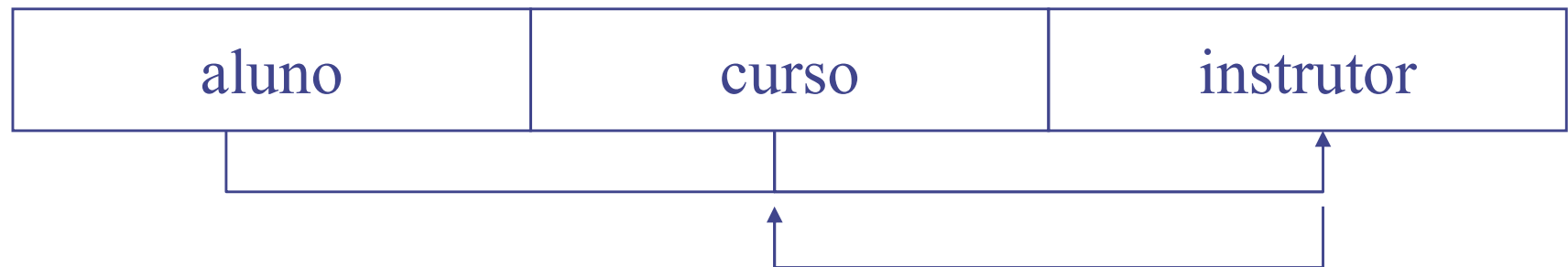
## Exemplo



Uma Relação R esquemática com DF's. Está na 3FN mas não na BCNF

# Forma Normal de Boyce-Codd

## Outro Exemplo

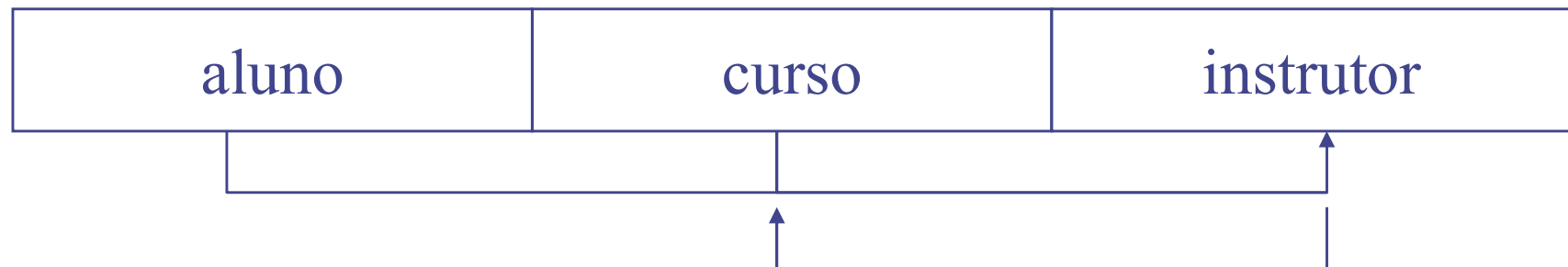


### ◆ Dependências Funcionais

- $\{\text{aluno}, \text{curso}\} \rightarrow \text{instrutor}$
- $\text{instrutor} \rightarrow \text{curso}$ 
  - ◆ essa dependência, que representa que cada instrutor ministra um curso, é uma restrição particular da aplicação
- R (aluno, curso, instrutor)

# Forma Normal de Boyce-Codd

## Outro Exemplo



ENSINA

ALUNO	CURSO	INSTRUTOR
Narayan	Banco de dados	Mark
Smith	Banco de dados	Navathe
Smith	Sistemas operacionais	Ammar
Smith	Teoria	Schulman
Wallace	Banco de dados	Mark
Wallace	Sistemas operacionais	Ahamad
Wong	Banco de dados	Omiecinski
Zelaya	Banco de dados	Navathe

# Forma Normal de Boyce-Codd

## Outro Exemplo

R (aluno, curso, instrutor)

### ◆ Solução 1

- aluno\_instrutor (aluno, instrutor)
- aluno\_curso (aluno, curso)

### ◆ Solução 2

- instrutor\_curso (instrutor, curso)
- aluno\_curso (aluno, curso)

### ◆ Solução 3

- instrutor\_curso (instrutor, curso)
- aluno\_instrutor (aluno, instrutor)

ALUNO	INSTRUTOR	ALUNO	CURSO
Narayan	Mark	Narayan	Banco de dados
Smith	Navathe	Smith	Banco de dados
Smith	Ammar	Smith	Sistemas operacionais
Smith	Schulman	Smith	Teoria
Wallace	Mark	Wallace	Banco de dados
Wallace	Ahamad	Wallace	Sistemas operacionais
Wong	Omiecinski	Wong	Banco de dados
Zelaya	Navathe	Zelaya	Banco de dados

instrutor	curso
Mark	Banco de Dados
Navathe	Banco de Dados
Ammar	Sistemas Operacionai
Schulman	Teoria
Ahamad	Sistemas Operacionai
Omiecinski	Banco de Dados

**Solução 3:** não  
gera tuplas  
espúrias



# Forma Normal de Boyce-Codd

## Outro Exemplo

### ◆ Solução 1

- aluno\_instrutor (aluno, instrutor)
- aluno\_curso (aluno, curso)

ALUNO	INSTRUTOR
Narayan	Mark
Smith	Navathe
Smith	Ammar
Smith	Schulman
Wallace	Mark
Wallace	Ahamad
Wong	Omiecinski
Zelaya	Navathe

ALUNO	CURSO
Narayan	Banco de dados
Smith	Banco de dados
Smith	Sistemas operacionais
Smith	Teoria
Wallace	Banco de dados
Wallace	Sistemas operacionais
Wong	Banco de dados
Zelaya	Banco de dados

- **SOLUÇÃO 1:** há geração de “**Tuplas Espúrias**” pois há uma combinação de ALUNOXCURSO com cada INSTRUTOR distinto (em que **aluno** coincide): **o problema é que há alunos que cursaram cursos com diferentes instrutores.**

ALUNO	CURSO	INSTRUTOR
Narayan	Banco de dados	Mark
Smith	Banco de dados	Navathe
Smith	Sistemas operacionais	Ammar
Smith	Teoria	Schulman
Wallace	Banco de dados	Mark
Wallace	Sistemas operacionais	Ahamad
Wong	Banco de dados	Omiecinski
Zelaya	Banco de dados	Navathe

8 tuplas

aluno	curso	instrutor
Narayan	Banco de Dados	Mark
Smith	Banco de Dados	Navathe
Smith	Banco de Dados	Ammar
Smith	Banco de Dados	Schulman
Smith	Sistemas Operacionai	Navathe
Smith	Sistemas Operacionai	Ammar
Smith	Sistemas Operacionai	Schulman
Smith	Teoria	Navathe
Smith	Teoria	Ammar
Smith	Teoria	Schulman
Wallace	Banco de Dados	Mark
Wallace	Banco de Dados	Ahamad
Wallace	Sistemas Operacionai	Mark
Wallace	Sistemas Operacionai	Ahamad
Wong	Banco de Dados	Omiecinski
Zelaya	Banco de Dados	Navathe

16

# Forma Normal de Boyce-Codd

## Outro Exemplo

### ◆ Solução 2

- instrutor\_curso (instrutor, curso)
- aluno\_curso (aluno, curso)

instrutor	curso
Mark	Banco de Dados
Navathe	Banco de Dados
Ammar	Sistemas Operacionais
Schulman	Teoria
Ahamad	Sistemas Operacionais
Omiecinski	Banco de Dados

ALUNO	CURSO
Narayan	Banco de dados
Smith	Banco de dados
Smith	Sistemas operacionais
Smith	Teoria
Wallace	Banco de dados
Wallace	Sistemas operacionais
Wong	Banco de dados
Zelaya	Banco de dados

- **SOLUÇÃO 2:** há geração de “Tuplas Espúrias” pois há uma combinação de ALUNOXCURSO com cada INSTRUTOR distinto (em que curso coincide): **o problema é que há cursos ministrado por instrutores distintos**

8 tuplas

ALUNO	CURSO	INSTRUTOR
Narayan	Banco de dados	Mark
Smith	Banco de dados	Navathe
Smith	Sistemas operacionais	Ammar
Smith	Teoria	Schulman
Wallace	Banco de dados	Mark
Wallace	Sistemas operacionais	Ahamad
Wong	Banco de dados	Omiecinski
Zelaya	Banco de dados	Navathe

aluno	curso	instrutor
Narayan	Banco de Dados	Mark
Narayan	Banco de Dados	Navathe
Narayan	Banco de Dados	Omiecinski
Smith	Banco de Dados	Mark
Smith	Banco de Dados	Navathe
Smith	Banco de Dados	Omiecinski
Smith	Sistemas Operacionais	Ammar
Smith	Sistemas Operacionais	Ahamad
Smith	Teoria	Schulman
Wallace	Banco de Dados	Mark
Wallace	Banco de Dados	Navathe
Wallace	Banco de Dados	Omiecinski
Wallace	Sistemas Operacionais	Ammar
Wallace	Sistemas Operacionais	Ahamad
Wong	Banco de Dados	Mark
Wong	Banco de Dados	Navathe
Wong	Banco de Dados	Omiecinski
Zelaya	Banco de Dados	Mark
Zelaya	Banco de Dados	Navathe
Zelaya	Banco de Dados	Omiecinski

20

# Forma Normal de Boyce-Codd

## Outro Exemplo

### ◆ Solução 3

- instrutor\_curso (instrutor, curso)
- aluno\_instrutor (aluno, instrutor)

- Na **SOLUÇÃO 3** há uma combinação de “INSTRUTOR” com todos os CURSOS (1:1) e INSTRUTOR com todos os alunos (1:8)

instrutor	curso
Mark	Banco de Dados
Navathe	Banco de Dados
Ammar	Sistemas Operacionai
Schulman	Teoria
Ahamad	Sistemas Operacionai
Omieciski	Banco de Dados

ALUNO	INSTRUTOR
Narayan	Mark
Smith	Navathe
Smith	Ammar
Smith	Schulman
Wallace	Mark
Wallace	Ahamad
Wong	Omiecinski
Zelaya	Navathe

aluno	curso	instrutor
Narayan	Banco de Dados	Mark
Smith	Sistemas Operacionai	Ammar
Smith	Banco de Dados	Navathe
Smith	Teoria	Schulman
Wallace	Sistemas Operacionai	Ahamad
Wallace	Banco de Dados	Mark
Wong	Banco de Dados	Omieciski
Zelaya	Banco de Dados	Navathe

8 tuplas

ALUNO	CURSO	INSTRUTOR
Narayan	Banco de dados	Mark
Smith	Banco de dados	Navathe
Smith	Sistemas operacionais	Ammar
Smith	Teoria	Schulman
Wallace	Banco de dados	Mark
Wallace	Sistemas operacionais	Ahamad
Wong	Banco de dados	Omiecinski
Zelaya	Banco de dados	Navathe

8

Solução 3: Solução desejada pois **não** gera tuplas espúrias

# Dependências Multivaloradas

- ◆ A normalização de relações por meio de dependências funcionais é uma das maneiras de evitar inconsistências em relações
- ◆ E quando não é possível especificar uma restrição semântica como uma DF?
  - dependência multivalorada



# Dependências Multivaloradas

## ◆ Dependência Funcional

- o valor de um conjunto de atributos pode ser determinado a partir do valor de outro conjunto de atributos

## ◆ Dependência Multivalorada:

- um conjunto de atributos não determina o valor de outro(s) atributo(s), mas sim restringe os valores possíveis para este(s) atributo(s)

# Dependências Multivaloradas

- ◆ Tal como ocorre com a dependência funcional, a multi-dependência não pode ser inferida pelo SGBD, e portanto também deve ser identificada e considerada pelo projetista do sistema de banco de dados

# Dependências Multivaloradas

- ◆ São conseqüências da Primeira Forma Normal (1FN)
  - quando se opta por eliminar a multivaloração com “repetições” de tuplas e não quebrando em uma nova relação
- ◆ Ocorrem quando dois ou mais atributos **independentes** multivalorados existem na mesma relação do BD
- ◆ Problema apresentado:
  - necessidade de se repetir cada um dos valores de um atributo com cada valor do outro atributo, para manter as instâncias da relação de maneira consistente

# Exemplo

vendedor ( nro\_vend, { cliente (nro\_cli) },  
{ filho (nome\_filho\_vend) })

## ◆ Informações sobre:

- vendedores
- clientes atendidos pelos vendedores
- filhos dos vendedores

## ◆ Observações:

- um vendedor pode atender vários clientes
- um cliente pode ser atendido por vários vendedores
- um vendedor pode possuir vários filhos



# Exemplo

nro_vend	nro_cli	nome_filho_vend
123	12805	Marcos Pedro Paulo
456	37573 24139 36273	Maria
444	57384	Ricardo
vendedor ( <u>nro_vend</u> , {nro_cli}, {nome_filho_vend} )		

– *vendedor nem mesmo pode ser qualificado como uma relação ...*

# Dependência Multivalorada

- ◆ Um atributo B de um esquema de relação R é multidependente de um outro atributo A de R se um valor para A é associado a uma coleção específica de valores para B, independentemente de qualquer valor que um terceiro atributo C de R possa assumir
- ◆ Se B é multidependente de A, então A multidetermina B
- ◆ Notação:  $A \twoheadrightarrow B$

# Dependência Multivalorada

relação  $R$  (atributo  $A$ , {atributo  $B$ },  
{atributo  $C$ })

atributo  $A \twoheadrightarrow$  atributo  $B$

↑  
1 valor de  $A$   
(não existe outro)

↑  
vários valores para  
o atributo  $B$

**independentemente** dos valores do atributo  $C$

# Quarta Forma Normal (4FN)

- ◆ Uma relação R está na 4FN se:
  - não existe dependência multivalorada
- ◆ Forma prática de se tratar a 4FN:
  - prevenir dependências multivaloradas no processo inicial de transformação da relação não normalizada (contendo o(s) grupo(s) de repetição para a 1FN)

# Outro Exemplo

(a)

**EMP**

<u>ENOME</u>	<u>PNOME</u>	<u>DNOME</u>
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

(b)

**EMP\_PROJETOS**

<u>ENOME</u>	<u>PNOME</u>
Smith	X
Smith	Y

**EMP\_DEPENDENTES**

<u>ENOME</u>	<u>DNOME</u>
Smith	John
Smith	Anna

DFMV:

ENOME  $\twoheadrightarrow$  PNOME

ENOME  $\twoheadrightarrow$  DNOME

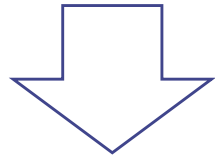
# Quarta Forma Normal (4FN)

## ◆ Método para corrigir o problema

- para cada grupo de repetição separado, gera-se uma nova relação correspondente contendo este grupo de repetição e a chave primária da relação original
- determinar a chave primária da nova relação, a qual será a concatenação da chave primária da relação original com a chave para o grupo de repetição

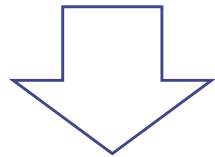
# Exemplo

vendedor ( nro\_vend, { cliente (nro\_cli) },  
{ filho (nome\_filho\_vend) })



nro\_vend  $\longrightarrow$  nro\_cli

nro\_vend  $\longrightarrow$  nome\_filho\_vend



vend\_cli ( nro\_vend, nro\_cli )

vend\_filho ( nro\_vend, nome\_filho\_vend )

# Exemplo

<u>nro_vend</u>	<u>nro_cli</u>
123	12805
123	24139
456	37573
456	24139
456	36273
444	57384

vend\_cli ( nro\_vend, nro\_cli )

<u>nro_vend</u>	<u>nome_filho_vend</u>
123	Marcos
123	Pedro
123	Paulo
456	Maria
444	Ricardo

vend\_filho ( nro\_vend, nome\_filho\_vend )



# Ilustrando a importância da 4FN

(a) **EMP**

<u>ENOME</u>	<u>PNAME</u>	<u>DNOME</u>
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John
Brown	W	Jim
Brown	X	Jim
Brown	Y	Jim
Brown	Z	Jim
Brown	W	Joan
Brown	X	Joan
Brown	Y	Joan
Brown	Z	Joan
Brown	W	Bob
Brown	X	Bob
Brown	Y	Bob
Brown	Z	Bob

16

(b) **EMP\_PROJETOS**

<u>ENOME</u>	<u>PNAME</u>
Smith	X
Smith	Y
Brown	W
Brown	X
Brown	Y
Brown	Z

6

+

**EMP\_DEPENDENTES**

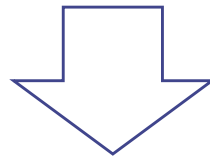
<u>ENOME</u>	<u>DNOME</u>
Smith	Anna
Smith	John
Brown	Jim
Brown	Joan
Brown	Bob

5

Além de evitar a ocorrência de anomalias de atualização

# Exemplo Mais Geral

vendedor ( nro\_vend, nome\_vend, {cliente  
(nro\_cli, nome\_cli)}, {filho (nome\_filho\_vend,  
idade\_filho\_vend) } )



vendedor ( nro\_vend, nome\_vend )  
cliente ( nro\_vend, nro\_cli, nome\_cli )  
filho ( nro\_vend, nome\_filho\_vend,  
idade\_filho\_vend)

# Considerações Finais

- ◆ A normalização para as FN apoiadas em DF sempre se atinge com a separação dos atributos em duas ou mais relações
  - Isso aumenta o número de relações
  - Requer operações de junção na recuperação de informações
- ◆ Normalizar evita inconsistências nas relações, porém obriga a execução de custosas operações de junção para a consulta de informações

# Considerações Finais

- ◆ Normalizar ou não uma relação?
  - O que é mais importante
    - ◆ garantir a eliminação de inconsistências no banco de dados ou a eficiência de acesso?
  - Se a consistência não for um fator fundamental pode-se abrir mão da normalização

# **ESTUDO DE CASO: A LOJA NA INTERNET**

# Estudo de caso: A loja na internet

## Análise de requisitos

- ◆ A livraria B&N decidiu se tornar on-line. Para isso precisa contratar uma consultoria especializada para projetar e implementar seu banco de dados. O proprietário da B&N forneceu o seguinte sumário sobre o que desejava:
- ◆ “Gostaria que meus clientes fossem capazes de navegar no meu catálogo de livros e solicitar pedidos pela Internet. Atualmente, aceito pedidos pelo telefone. A maioria de meus clientes corporativos me liga e me fornece o número ISBN de um livro e a quantidade; eles normalmente pagam com cartão de crédito.

# Estudo de caso: A loja na internet

## Análise de requisitos

- ◆ Preparo, então, uma remessa que contém os livros solicitados. Se não tenho cópias suficientes em estoque, solicito cópias adicionais e atraso a remessa até que as novas cópias cheguem; quero enviar o pedido inteiro de um cliente junto. Meu catálogo inclui todos os livros que vendo.
- ◆ Para cada livro, o catálogo contém seu número ISBN, título, autor, preço de aquisição, preço de venda, e o ano em que o livro foi publicado.

# Estudo de caso: A loja na internet

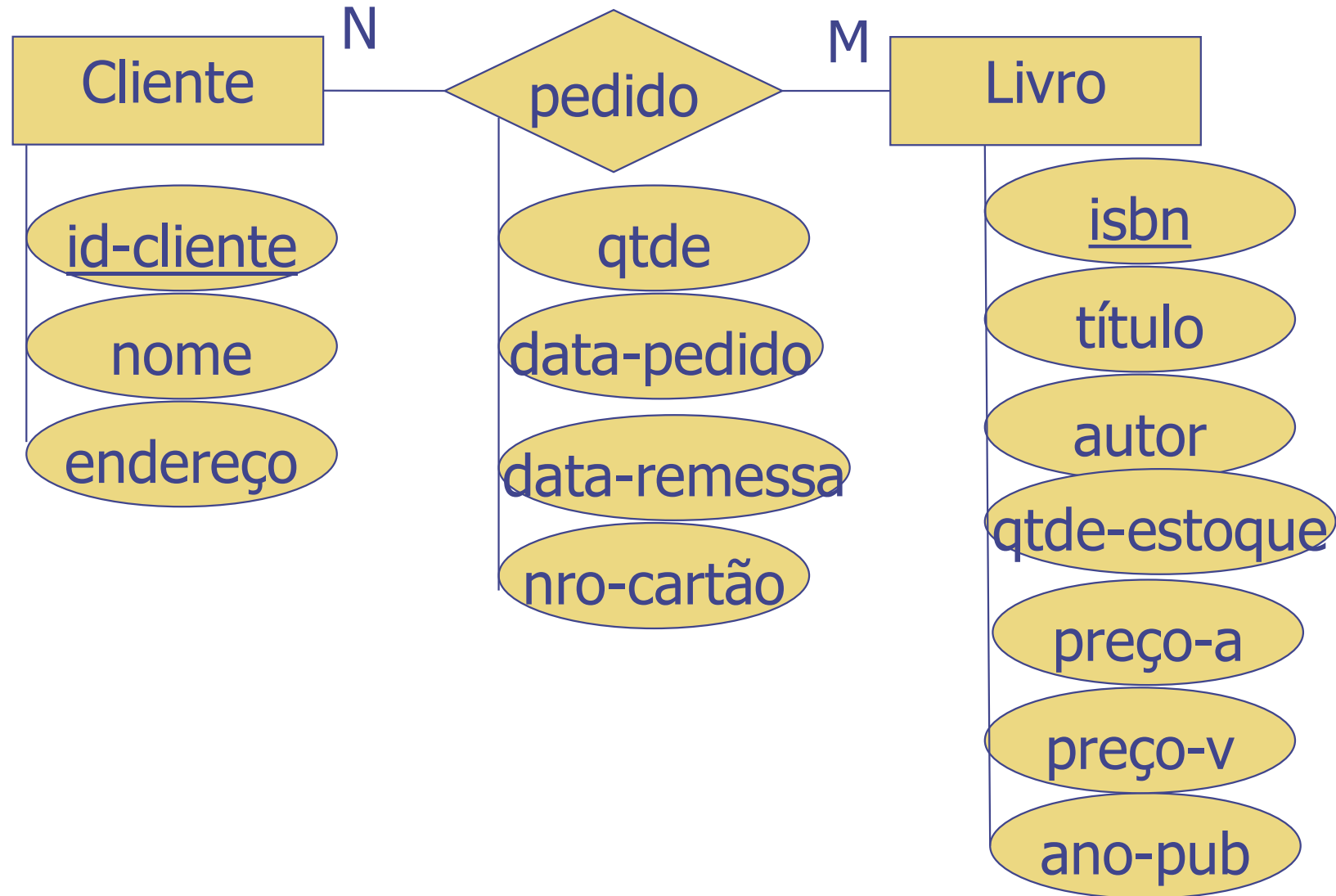
## Análise de requisitos

- ◆ A maioria dos meus clientes é cadastrada, e tenho registros com seus nomes e endereços.
- ◆ Novos clientes devem me ligar primeiro e estabelecer uma conta antes que possam usar meu site.
- ◆ Em meu novo site, os clientes devem se identificar primeiro por meio de seu número único de identificação de cliente. Depois, eles devem ser capazes de navegar no meu catálogo e solicitar pedidos on-line.”



# Estudo de caso: A loja na internet

## Modelo Entidade Relacionamento



# Estudo de caso: A loja na internet

## Modelo Relacional

- ◆ Livro (isbn, titulo, autor, qtde-estoque, preco-a, preco-v, ano-pub)
- ◆ Cliente(id-cliente, nome, endereco)
- ◆ Pedido(isbn (Livro.isbn), id-cliente(Cliente.id-cliente), nro-cartao, qtde, data-pedido, data-remessa)

# Estudo de caso: A loja na internet

## Modelo Relacional

- ◆ Livro (isbn, titulo, autor, qtde-estoque, preco-a, preco-v, ano-pub)
- ◆ Cliente(id-cliente, nome, endereco)
- ◆ Pedido(isbn (Livro.isbn), id-cliente(Cliente.id-cliente), nro-cartao, qtde, data-pedido, data-remessa)
- ◆ Obs: nesse modelo um cliente não pode pedir o mesmo livro em dias diferentes, uma restrição que não era pretendida. O que fazer?

# Estudo de caso: A loja na internet

## Requisitos adicionais

- ◆ “ Os clientes podem adquirir vários livros diferentes em um único pedido.
- ◆ Sobre a política de remessa, assim que existirem exemplares suficientes de um livro pedido, ele é enviado, mesmo que um pedido contenha vários livros. Além disso, os clientes podem fazer mais de um pedido por dia, e eles querem identificar os pedidos que fazem.”
- ◆ Requisitos:
  - Deve ser possível solicitar vários livros diferentes em um único pedido
  - Um cliente deve ser capaz de distinguir entre vários pedidos feitos no mesmo dia

# Estudo de caso: A loja na internet

## Modelo Relacional alterado

- ◆ Livro (isbn, titulo, autor, qtde-estoque, preco-a, preco-v, ano-pub)
- ◆ Cliente(id-cliente, nome, endereco)
- ◆ Pedido(nro-pedido, isbn (Livro.isbn), id-cliente(Cliente.id-cliente), nro-cartao, qtde, data-pedido, data-remessa)
- ◆ Observações
  - nro-pedido identifica univocamente cada pedido e, portanto, o cliente que está fazendo o pedido
  - Como vários livros podem ser comprados em um único pedido, nro-pedido e isbn são ambos necessários para se determinar qtde e data da remessa

# Estudo de caso: A loja na internet

## Analizando o Modelo Relacional alterado

◆ Livro (isbn, titulo, autor, qtde-estoque, preco-a, preco-v, ano-pub)

- tem apenas um atributo chave e nenhuma outra DF. Livro está na FNBC

◆ Cliente(id-cliente, nome, endereço)

- tem apenas um atributo chave e nenhuma outra DF. Cliente está na FNBC

# Estudo de caso: A loja na internet

## Analizando o Modelo Relacional alterado

- ◆ Pedido(nro-pedido, isbn (Livro.isbn), id-cliente(Cliente.id-cliente), nro-cartao, qtde, data-pedido, data-remessa)
  - nro-pedido e isbn formam a PK de Pedido
  - {nro-pedido, isbn} → {qtde, data-remessa}
  - além disso, como cada pedido é feito por um cliente em uma data específica, com um número de cartão de crédito específico, as três DF são válidas
    - ◆ nro-pedido → id-cliente
    - ◆ nro-pedido → data-pedido
    - ◆ nro-pedido → nro-cartao
- Pedido está em qual forma normal?

# Estudo de caso: A loja na internet

## Analizando o Modelo Relacional alterado

- ◆ Pedido(nro-pedido, isbn (Livro.isbn), id-cliente(Cliente.id-cliente), nro-cartao, qtde, data-pedido, data-remessa)
  - nro-pedido e isbn formam a PK de Pedido
  - {nro-pedido, isbn} → {qtde, data-remessa}
  - além disso, como cada pedido é feito por um cliente em uma data específica, com um número de cartão de crédito específico, as três DF são válidas
    - ◆ nro-pedido → id-cliente
    - ◆ nro-pedido → data-pedido
    - ◆ nro-pedido → nro-cartao
- Pedido está em qual forma normal?
  - ◆ Não está nem na 2FN



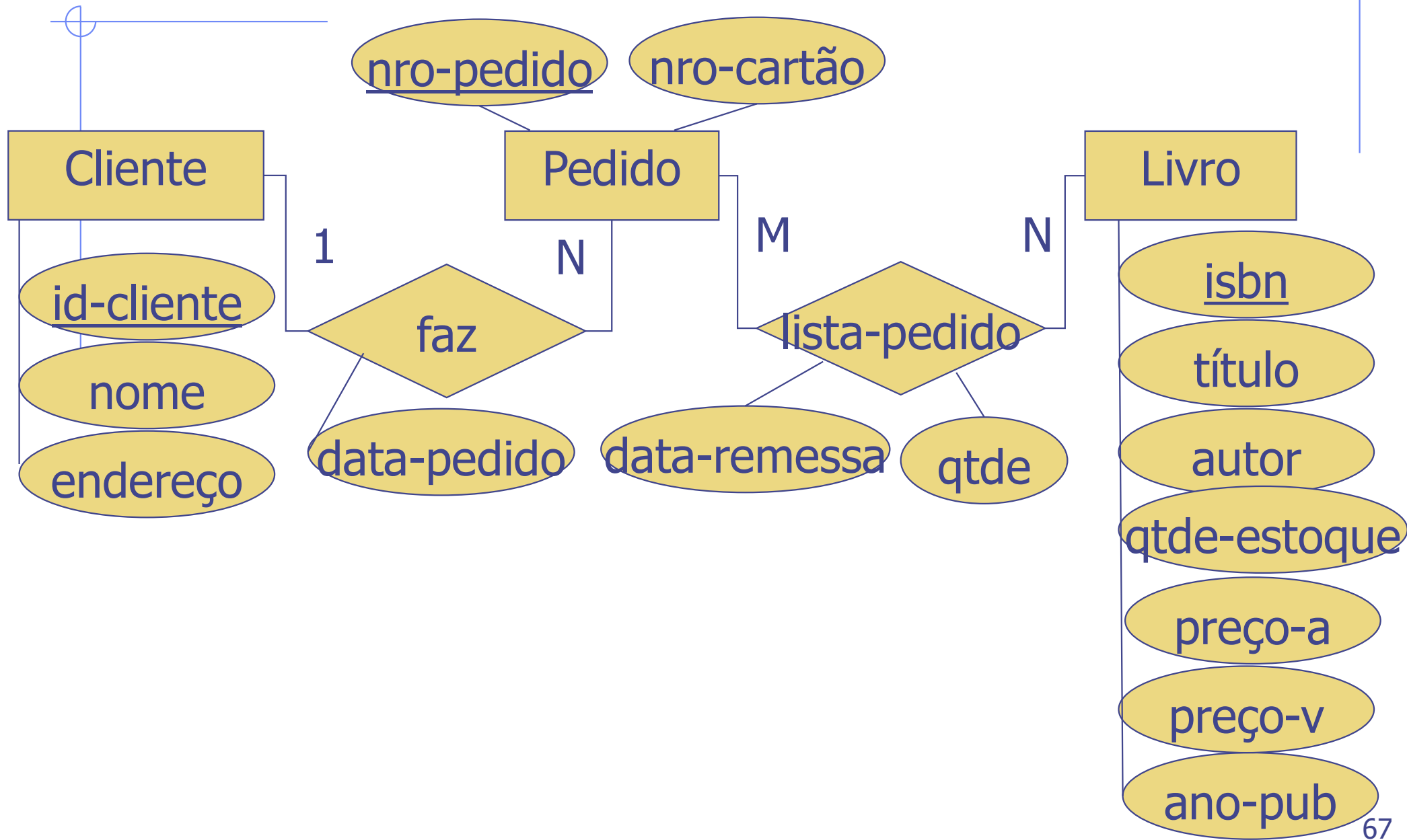
# Estudo de caso: A loja na internet

## Analizando o Modelo Relacional alterado

- ◆ Pedido(nro-pedido, isbn (Livro.isbn), id-cliente(Cliente.id-cliente), nro-cartao, qtde, data-pedido, data-remessa)
- ◆ Normalizando Pedido...
- ◆ Pedido(nro-pedido, id-cliente(Cliente.id-cliente), nro-cartao, data-pedido)
- ◆ Lista-Pedido(nro-pedido, isbn (Livro.isbn), qtde, data-remessa)

# Estudo de caso: A loja na internet

## Modelo Entidade Relacionamento Atualizado



# Exercícios complementares

## Normalização

- ◆ Diga em que forma normal (Nenhuma, 1 FN, 2 FN, 3 FN ou FNBC) está cada relação abaixo, justificando sua resposta. Depois, se necessário, indique os passos que devem ser realizados para normalizar para a forma normal mais restrita possível.
- ◆ LIVROS = {Titulo, Autor, Tipo, Preço, {FiliacaoDoAutor}, Editora}
  - DF: Titulo  $\rightarrow$  {Editora, Tipo}, Tipo  $\rightarrow$  Preço, Autor  $\rightarrow$  FiliacaoDoAutor
- ◆ FORNECEDOR = {CNPJ, RazaoSocial, NomeFantasia, Contato}
  - DF: CNPJ  $\rightarrow$  {RazaoSocial, NomeFantasia, Contato}
- ◆ CLIENTE = {CPF, Nome, NroAgencia, NroConta, TipoConta}
  - DF: CPF  $\rightarrow$  {Nome, NroAgencia, NroConta, TipoConta}, {NroAgencia, NroConta}  $\rightarrow$  {CPF, Nome, TipoConta}, TipoConta  $\rightarrow$  NroAgencia

# Exercícios complementares

## Normalização

1. Diga em que forma normal (Nenhuma, 1 FN, 2 FN, 3 FN ou FNBC) está cada relação abaixo, justificando sua resposta. Depois, se necessário, indique os passos que devem ser realizados para normalizar para a forma normal mais restrita possível.
  - ◆ CARROSVENDIDOS = {Carro, DataVenda, Vendedor, Comissao, Desconto}
    - DF: Carro → DataVenda, DataVenda → Desconto, Vendedor → Comissao
  - ◆ FILIAL = {CodF, Pais, Cidade, Continente, Lingua, NomeGerente, FusoHorario, Nivel}
    - DF: CodF → {Pais, Cidade, NomeGerente, Nivel, FusoHorario}, Pais → {Continente, Lingua}
2. Os seguintes exercícios dos Cap. 15 e 16 do Navathe: 15.2, 15.4, 15.5, 15.13, 16.8 e 16.25

# Bibliografia

- ◆ Elmasri, Ramez; Navathe, Shamkant B. **Sistemas de banco de dados**. 4 ed. São Paulo: Addison Wesley, 2005, 724 p. Bibliografia: p. [690]-714.
- ◆ Garcia-Molina, H.; Ullman J. D.; Widow, J. Database Systems – The Complete Book. Prentice-Hall, 2002.
- ◆ Ramakrishnan, R. Sistemas de bancos de dados. 3 ed. São Paulo: McGraw-Hill, 2008.
- ◆ Material Didático produzido pelos professores Cristina Dutra de Aguiar Ciferri e Caetano Traina Júnior

# Leitura complementar para casa

- ◆ Capítulo 15 do livro: Elmasri, Ramez; Navathe, Shamkant B. Sistemas de banco de dados. 6ª Edição.

# Conteúdo Primeira Prova

- ◆ Introdução ao SGBD
- ◆ Modelos ER e ERX
- ◆ Modelo Relacional
- ◆ Mapeamento MER e MERX para o MRel

**Data da Prova: 13/09 (Nos dois primeiros horários – em sala de aula)**

# Sugestão de revisão das matérias anteriores...

- ◆ Revisar as notas de aula e as suas anotações
- ◆ Leituras complementares para casa
  - Fazem parte do conteúdo da prova!!!
  - Ao final de cada capítulo existem as seções “Resumo” e “Perguntas de Revisão” que resumizam os principais conceitos abordados
- ◆ Revisar as listas de exercícios
- ◆ Fazer exercícios adicionais disponíveis nos livros da bibliografia da disciplina



# Sugestão de revisão das matérias anteriores...

## ◆ MATÉRIA INTRODUTÓRIA

- Capítulos 1 e 2 do livro: Elmasri, Ramez; Navathe, Shamkant B. Sistemas de banco de dados. (2011)
- Capítulo 1 do livro: Garcia-Molina, Hector; Ullman, Jeff; Widom, Jennifer. Database Systems: The Complete Book.

## ◆ MODELO ENTIDADE RELACIONAMENTO

- Capítulo 7 do livro: Elmasri, Ramez; Navathe, Shamkant B. Sistemas de banco de dados. (2011)
- Capítulo 8 do livro: Elmasri, Ramez; Navathe, Shamkant B. Sistemas de banco de dados. (2011)
- O livro do professor Carlos Heuser também é uma boa referência para esse tópico: *Heuser, C. A., Projeto de Banco de Dados. Porto Alegre: Sagra .*

# Sugestão de revisão das matérias anteriores...

## ◆ MODELO RELACIONAL

- Capítulo 3 do livro: Elmasri, Ramez; Navathe, Shamkant B. Sistemas de banco de dados. (2011)
- Capítulo 9 do livro: Elmasri, Ramez; Navathe, Shamkant B. Sistemas de banco de dados. (2011)