

# Normalização

Banco de Dados

Profa. Dra. Cristina Dutra de Aguiar Ciferri

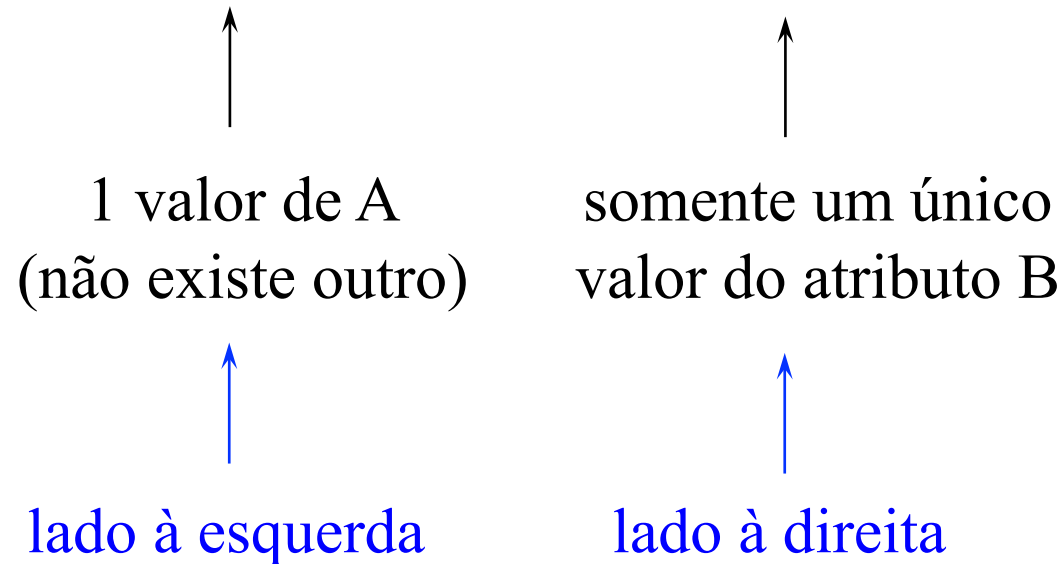
# Dependência Funcional

- ◆ Um atributo B de um esquema de relação R é funcionalmente dependente de um outro atributo A de R se um valor para A determina um único valor para B em qualquer momento.
- ◆ Se B é funcionalmente dependente de A, então A determina funcionalmente B.
- ◆ Notação:  $A \rightarrow B$

# Dependência Funcional

relação R (atributo A, atributo B, atributo C)

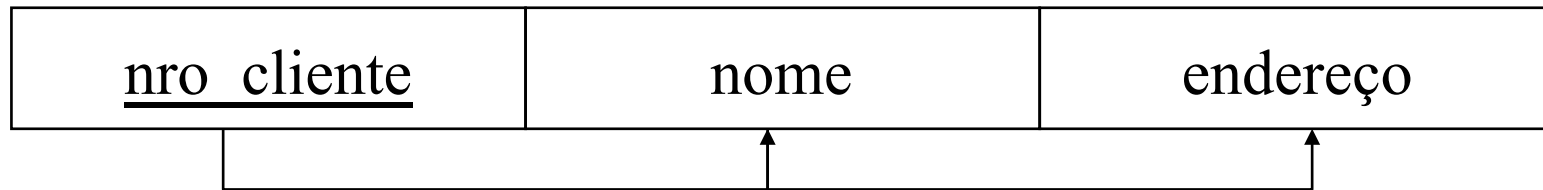
atributo A  $\rightarrow$  atributo B



# Exemplo

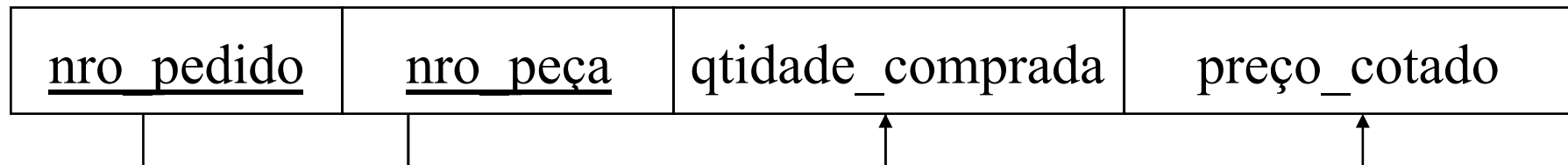
cliente (nro\_cliente, nome, endereço)

nro\_cliente → {nome, endereço}



pp (nro\_pedido, nro\_peça, qtidade\_comprada, preço\_cotado)

{nro\_pedido, nro\_peça} → {qtidade\_comprada, preço\_cotado}

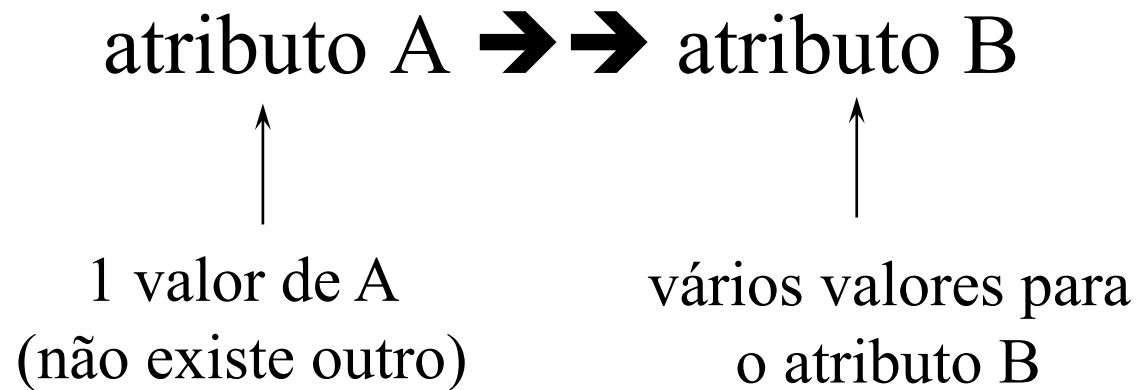


# Dependência Multivalorada

- ♦ 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})



**independentemente** dos valores do atributo C

# Exemplo

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

nro\_vend →→ cliente (nro\_cli, nome\_cli)

nro\_vend →→ filho (nome\_filho\_vend, parentesco)

# Normalização

- ◆ Processo de normalização:
  - inicia com um esquema de relação ou coleção de esquemas de relação
  - produz uma nova coleção de esquemas de relação
    - ◆ equivalente à coleção original (representa a mesma informação)
    - ◆ livre de problemas



# Normalização por Decomposição

## ◆ Propriedades

- junção sem perda ou junção não aditiva
  - ◆ garante que o problema de tuplas ilegítimas não ocorra nos esquemas de relação criados após a decomposição
- preservação da dependência
  - ◆ garante que cada dependência funcional será representada em algum esquema de relação individual resultante da decomposição

# Primeira Forma Normal (1FN)

- ◆ Uma relação R está na 1FN se:
  - todo valor em R for atômico
  - ou seja, R não contém nenhum grupo de repetição/dependência multivalorada
- ◆ Considerações:
  - geralmente considerada parte da definição formal de uma relação
  - não permite atributos multivalorados, compostos ou suas combinações
  - **um único grupo de repetição**

# Método para Corrigir o Problema

## ♦ Passos

- gerar uma nova relação contendo o grupo de repetição e a chave primária da relação original
- determinar a chave primária da nova relação, a qual será a concatenação do atributo chave primária da relação original com o atributo chave para o grupo de repetição

# Exemplo

emp\_proj (nro\_emp, nome\_emp,  
          { projeto ( nro\_proj, nome\_proj ) })

## ◆ Dependências funcionais

– nro\_emp → nome\_emp

## ◆ Dependências multivaloradas

– nro\_emp →→ projeto (nro\_proj, nome\_proj)

## ◆ Solução

– emp (nro\_emp, nome\_emp)

– emp\_proj (nro\_emp, nro\_proj, nome\_proj)



# 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 primária
- dependência funcional total

# Segunda Forma Normal (2FN)

- ◆ Método para corrigir o problema:
  - para cada sub-conjunto do conjunto de atributos que constitui a chave primária, começar uma relação com esse sub-conjunto como sua chave primária
  - incluir os atributos da relação original na relação correspondente à chave primária apropriada, isto é, colocar cada atributo junto com a coleção mínima da qual ele depende, atribuindo um nome a cada relação

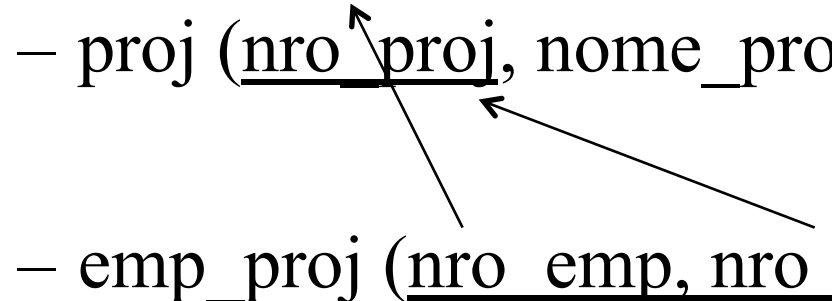
# Exemplo

emp\_proj (nro\_emp, nro\_proj, nome\_proj)

## ◆ Dependências funcionais

- nro\_emp →
- nro\_proj → nome\_proj
- {nro\_emp, nro\_proj} →

## ◆ Solução

- emp (nro\_emp, ...) *relação já existente*
  - proj (nro\_proj, nome\_proj)
  - emp\_proj (nro\_emp, nro\_proj)
- 

# 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



# 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

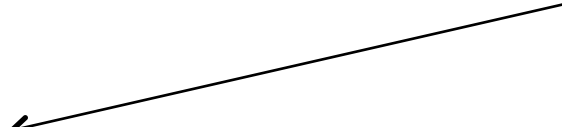
# Exemplo

cliente (nro\_cliente, nome\_cliente,  
nro\_vendedor, nome\_vendedor)

## ◆ Dependências funcionais

- nro\_cliente → {nome\_cliente, nro\_vendedor, nome\_vendedor}
- nro\_vendedor → nome\_vendedor

## ◆ Solução

- cli (nro\_cliente, nome\_cliente, nro\_vendedor)
  - vend (nro\_vendedor, nome\_vend)
- 

# Quarta Forma Normal (4 FN)

- ◆ Uma relação R está na 4FN se:
  - todo valor em R for atômico
  - ou seja, R não contém nenhum grupo de repetição/dependência multivalorada
- ◆ Considerações:
  - geralmente considerada parte da definição formal de uma relação
  - não permite atributos multivalorados, compostos ou suas combinações
  - **vários grupos de repetição**

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

## ◆ Dependências funcionais

nro\_vend  $\rightarrow$  nome\_vend

## ◆ Dependências multivaloradas

nro\_vend  $\twoheadrightarrow$  cliente {nro\_cli, nome\_cli}

nro\_vend  $\twoheadrightarrow$  filho {nome\_filho, parentesco}

# Exemplo

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

## ◆ Solução

– vendedor (nro\_vend, nome\_vend)

– vendCli (nro\_vend, nro\_cli, nome\_cli)

– vendFilho (nro\_vend, nome\_filho, parentesco)