

Dependência Funcional e Primeira, Segunda e Terceira Formas Normais

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: $A \twoheadrightarrow B$

Dependência Funcional

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

atributo A  atributo B



1 valor de A
(não existe outro)



lado à esquerda



somente um único
valor do atributo B







lado à direita

Exercício

- ♦ Dada a seguinte relação
cliente (nro_cliente, nome, endereço)

As seguintes dependências são corretas?








- ♦ nro_cliente  nome
- ♦ nro_cliente  endereço
- ♦ nome  endereço
- ♦ endereço  nome

Exercício

- ♦ Dada a seguinte relação (e suas instâncias)
linha_pedido (nro_pedido, nro_peça
qtidade_comprada, preço_cotado)

nro_pedido	nro_peça	qtidade_ comprada	preço_cotado
101	P01	3	30,00
101	P02	4	70,00
102	P01	8	80,00
102	P02	3	20,00

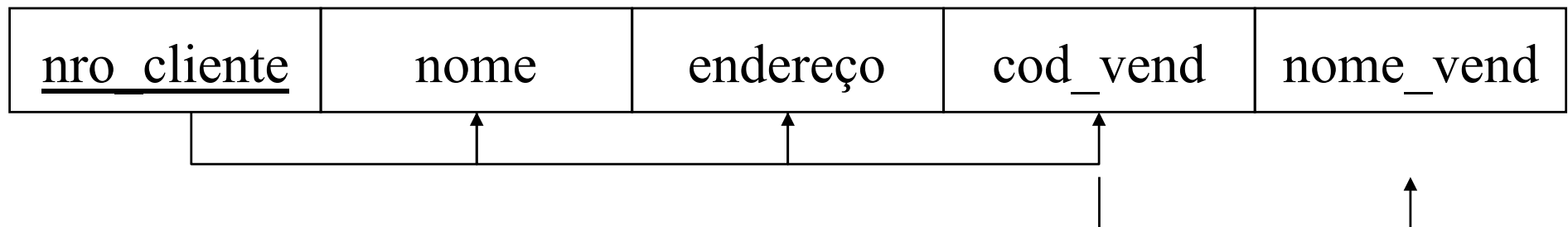
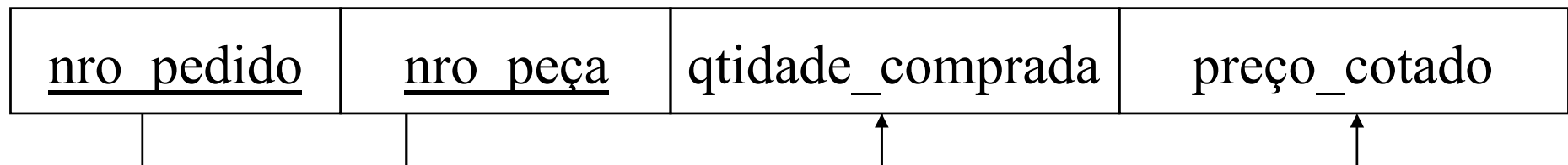
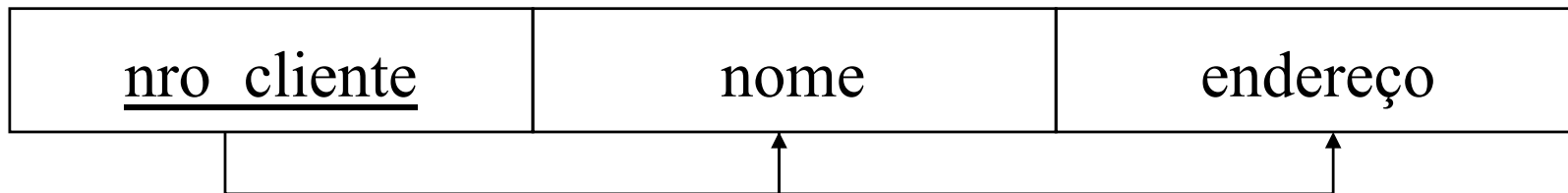
Exercício

- ♦ As seguintes dependências são verdadeiras?
- nro_pedido  qtidade_comprada
 - nro_peça  qtidade_comprada
 - nro_pedido  preço_quotado
 - nro_peça  preço_quotado
 - {nro_pedido, nro_peça}  qtidade_comprada
 - {nro_pedido, nro_peça}  preço_cotado
 - {nro_pedido, nro_peça}  {qtidade_comprada, preço_cotado}

Observações

- ♦ Uma dependência funcional é uma *propriedade do esquema da relação R* , não de um estado particular válido da relação r de R
- ♦ Se $X \rightarrow Y$ em R , isso não implica necessariamente que $Y \rightarrow X$ em R

Notação Diagramática para DF



Dependência Funcional e Normalização

♦ Vantagens:

- garante relações sem redundância desnecessária
- oferece fácil recuperação das informações

♦ Tipos de formas normais:

- Primeira Forma Normal (1FN)
- Segunda Forma Normal (2FN)
- Terceira Forma Normal (3FN)

Dependência Funcional e 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
- ♦ Significado: os novos esquemas de relação estarão, pelo menos, na 3FN

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

Chave Primária

- ♦ Um atributo A (ou coleção de atributos) é a chave primária para um esquema de relação R se
 - todos os atributos em R são funcionalmente dependentes de A
 - não existe um subconjunto próprio de A que determina funcionalmente os atributos em R
- ♦ Exercício
 - Considere as relações cliente e linha_pedido do exercício anterior. Qual a chave primária de cada relação? Por quê?

Primeira Forma Normal (1FN)

♦ Uma relação R está na 1FN se:

- todo valor em R for atômico

⇨ ou seja, R não contém grupos de repetição

♦ Considerações:

- geralmente considerada parte da definição formal de uma relação

- não permite atributos multivalorados, compostos ou suas combinações

Primeira Forma Normal (1FN)

♦ Exemplo

– cliente (nro_cli, nome, {end_entrega})

nro_cli	nome	end_entrega
124	João dos Santos	Rua 10, 1024 Rua 24, 1356
311	José Ferreira Neves	Rua 46, 1344 Rua 98, 4456

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

Métodos para Corrigir o Problema

♦ Método 1

- 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
- abordagem mais genérica e que não causa redundância

Métodos para Corrigir o Problema

♦ Método 2

- remover o grupo de repetição
- expandir a chave primária
- abordagem que causa redundância

♦ Método 3

- substituir o grupo de repetição pelo número máximo de valores estabelecido para o grupo
- abordagem menos genérica e que pode introduzir muitos valores *null*

Primeira Forma Normal (1FN)

♦ Problema

- cliente (nro_cli, nome, {end_entrega})

Corrigindo o problema ...

♦ Solução 1

- cliente_nome (nro_cli, nome)
- cliente_entrega (nro_cli, end_entrega)

♦ Solução 2

- cliente (nro_cli, nome, end_entrega)

♦ Solução 3

- cliente (nro_cli, nome, entrega1, entrega2)

Primeira Forma Normal (1FN)

♦ Exemplo mais genérico

- emp_proj (nro_emp, nome_emp,
 { projeto (nro_proj, nome_proj) })

♦ Representação

- { } indica que o atributo projeto é multivalorado
- {projeto ()} indica os atributos componentes do atributo multivalorado projeto

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
- ♦ Dependência funcional total $X \twoheadrightarrow Y$
 - se a remoção de qualquer atributo A de X implicar que a dependência não mais será assegurada

Segunda Forma Normal (2FN)

♦ Exemplo:

- pedido (nro-pedido, data, nro-peça, descrição, qtdade_comprada, preço_cotado)

nro-pedido  data

nro-peça  descrição

{nro-pedido, nro-peça}  {qtdade_comprada, preço_cotado}

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

Segunda Forma Normal (2FN)

- ♦ Problema: pedido (nro-pedido, data, nro-peça, descrição, qtdade_comprada, preço_cotado)

Corrigindo o problema ...

- ♦ Solução:

pedido (nro-pedido, data)

peça (nro_peça, descrição)

pedido_peça (nro_pedido, nro_peça,
qtdade_comprada, preço_cotado)


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

Terceira Forma Normal (3FN)

♦ Exemplo:

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

nro-cliente  nome-cliente, end-cliente,
nro_vendedor

nro-vendedor  nome_vendedor

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)

- ♦ 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 $X \stackrel{\text{W}}{\twoheadrightarrow} A$, X é uma superchave de R ou A é um atributo primário de R

Forma Normal de Boyce-Codd

♦ BCNF

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

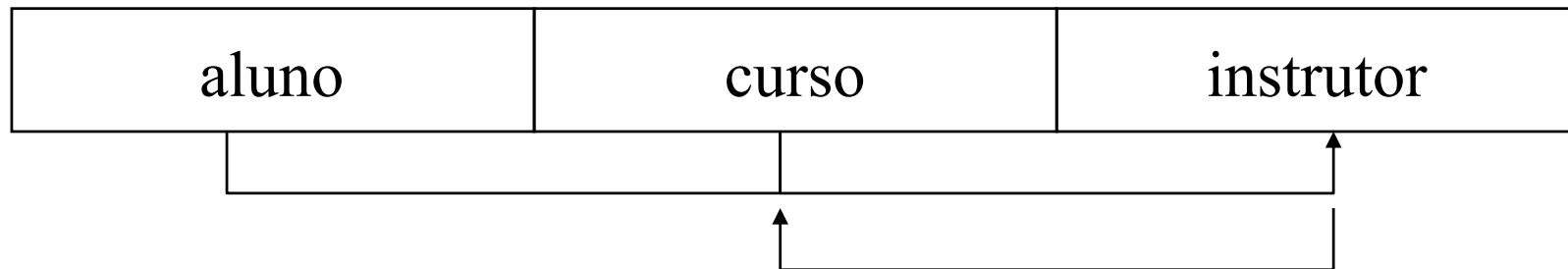
♦ BCNF e 3FN

- relação está na BCNF \twoheadrightarrow relação está na 3FN
- relação está na 3FN $\not\rightarrow$ relação está na BCNF

♦ Prática

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

Forma Normal de Boyce-Codd



♦ Dependências Funcionais

– {aluno, curso} \twoheadrightarrow instrutor

– instrutor \twoheadrightarrow 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

♦ 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)

melhor solução:
não gera tuplas
ilegítimas