

---

# Bancos (Bases) de Dados

Aula #6 – Dependência Funcional  
Dependência Multivalorada  
Normalização

---

**Prof. Eduardo R. Hruschka**

\* Slides baseados no material elaborado pelas professoras:

**Cristina D. A. Ciferri**

**Elaine P. M. de Souza**

---

# Qualidade do Projeto Lógico

- Como avaliar a qualidade do esquema da relação?
  - Semântica;
  - Implementação/desempenho.
- Análise informal:
  - *Princípios* para um bom projeto.
- Análise formal:
  - Dependência funcional;
  - Normalização.

---

# Qualidade do Projeto Lógico ...

- Análise Informal (princípios):
  - Semântica de atributos;
  - Redução de redundância em tuplas:
    - prevenção de anomalias de inserção;
    - prevenção de anomalias de remoção;
    - prevenção de anomalias de alteração.
  - Redução de valores nulos;
  - Prevenção de geração de tuplas espúrias (ilegítimas).

---

# Exemplo:

- **Emp\_Dept** = { Nome, CPF, DataNasc, End, Dnum, Dnome, DGerCPF }
- Combina informações de tipos diferentes de entidades;
  - Problema semântico;
- Redundância em relação às informações armazenadas;
  - Dados do departamento (Dnome e DGerCPF);
- Inserção:
  - Para inserir um empregado, é necessário cadastrar informações sobre o departamento (ou *nulls*). Tais informações podem gerar dados inconsistentes sobre o departamento.

---

# Exemplo...

- ❑ Exclusão:
  - Apagar um empregado pode significar apagar as informações do departamento.
- ❑ Atualização:
  - Mudar o valor de um atributo de uma tupla de *Emp\_Dept* pode implicar em ter de alterar outros valores correspondentes.
    - ❑ Ex.: mudar *Dnum*.
- ❑ Valores *null*:
  - Se muitos atributos não se aplicarem a muitas tuplas da relação, poderemos desperdiçar espaço de armazenamento. Ex:
    - ❑ Incluir nº escritório na relação “empregados”, sendo que somente 10% destes possuem de fato um escritório.

---

# Qualidade do Projeto Lógico ...

- **Análise Formal:**

- **Dependências Funcionais:**

- Restrições entre atributos:

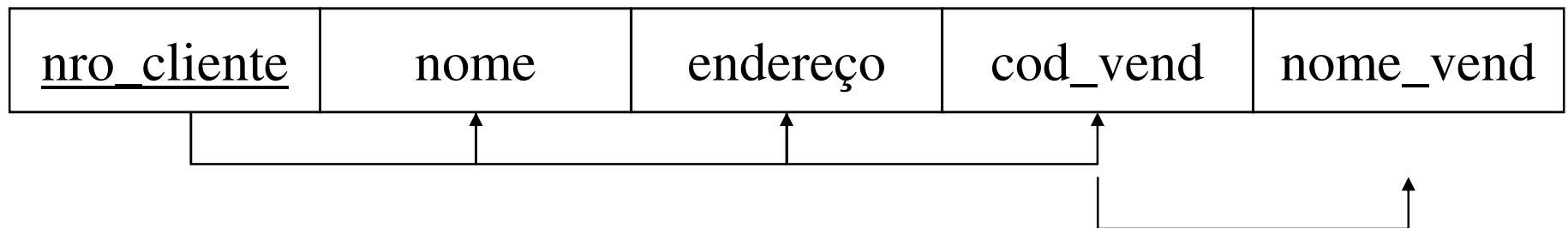
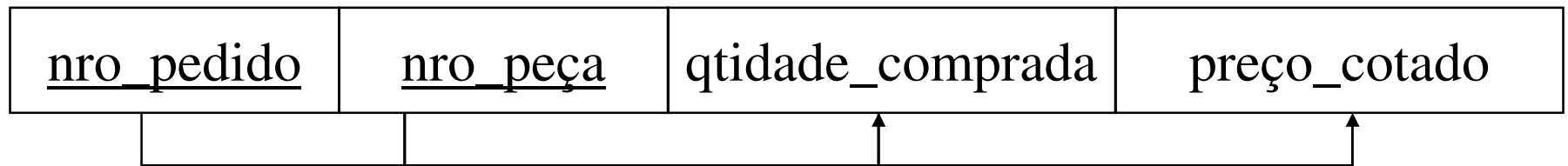
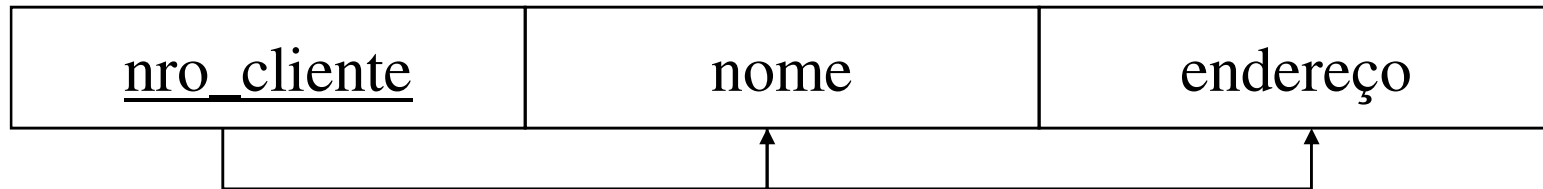
- Avaliação da qualidade dos esquemas de relação;
      - Garantia de consistência da base de dados.

---

# Dependência Funcional (DF)

- É uma restrição entre dois subconjuntos de atributos (**A** e **B**) de  $\mathcal{R}$ , sendo denotada por  $\mathbf{A} \rightarrow \mathbf{B}$ .
- Especifica uma restrição nas possíveis tuplas de  $\mathbf{R}(\mathcal{R})$ :  
Se  $t_i[\mathbf{A}] = t_j[\mathbf{A}]$  então  $t_i[\mathbf{B}] = t_j[\mathbf{B}]$  para quaisquer  $i, j$ .
- Neste caso diz-se que **A** determina funcionalmente **B** (ou alternativamente que **B** depende funcionalmente de **A**).
- Alguns exemplos:  
 $\{\text{NUSP}\} \rightarrow \{\text{Nome, Idade, Curso}\};$   
 $\{\text{Sigla, Sala, Hora}\} \rightarrow \{\text{CódigoTurma, Professor}\}$   
 $\{\text{Sigla}\} \rightarrow \{\text{NomeDisciplina, NCréditos}\}$

# Notação Diagramática para DF





---

# Dependência Funcional (DF) ...

- Propriedade semântica, identificada pelo projetista da(o) BDs;
- Pode ser verificada na instância do BDs mas não é definida a partir dela.

- Exemplo: Seja a relação **Alunos** = {Nome, Curso, Idade} e um de seus possíveis estados:

{ <Mario, Comp., 21>,  
<Paulo, Eletr., 22>,  
<Almir, Fisio., 22>,  
<Marta, Comp., 21>,  
<Vânia, Eletr., 22> }

---

# Dependência Funcional ...

- A relação **Alunos** atende às seguintes DFs?

- Nome  $\rightarrow$  Curso;
- Nome  $\rightarrow$  Idade;
- Curso  $\rightarrow$  Idade;
- Idade  $\rightarrow$  Curso;

**Alunos:**

<Mario, Comp., 21>,  
<Paulo, Eletr., 22>,  
<Almir, Fisio., 22>,  
<Marta, Comp., 21>,  
<Vânia, Eletr., 22>

---

# Exercícios

1. Dada a relação Cliente (n\_cliente, nome, endereço), as seguintes dependências são corretas?
    - a)  $n\_cliente \rightarrow nome$ ;
    - b)  $n\_cliente \rightarrow endereço$ ;
    - c)  $nome \rightarrow endereço$ ;
    - d)  $endereço \rightarrow nome$ .
  
  2. Dada a seguinte relação, deseja-se saber se as dependências listadas são verdadeiras:
-

---

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

- a)  $\text{nro\_pedido} \rightarrow \text{qtidade\_comprada}$ ;
  - b)  $\text{nro\_peça} \rightarrow \text{qtidade\_comprada}$ ;
  - c)  $\text{nro\_pedido} \rightarrow \text{preço\_cotado}$ ;
  - d)  $\text{nro\_peça} \rightarrow \text{preço\_cotado}$ ;
  - e)  $\{\text{nro\_pedido}, \text{nro\_peça}\} \rightarrow \text{qtidade\_comprada}$ ;
  - f)  $\{\text{nro\_pedido}, \text{nro\_peça}\} \rightarrow \text{preço\_cotado}$ ;
  - g)  $\{\text{nro\_pedido}, \text{nro\_peça}\} \rightarrow \{\text{qtidade\_comprada}, \text{preço\_cotado}\}$ .
-

---

# Dependência Funcional ...

- Controle de consistência:
  - Necessário conhecer todas as dependências funcionais - informação semântica fornecida pelo projetista;
  - Algumas dependências funcionais (DFs) podem ser inferidas a partir de DFs existentes ⇒ regras de inferência.

# Dependência Funcional ...

- Regras de Inferência de DFs:
  - Reflexiva: se  $B \subseteq A \Rightarrow A \rightarrow B$  (**DF trivial**)
  - Aumentativa: se  $A \rightarrow B \Rightarrow AC \rightarrow BC$
  - Decomposição: se  $A \rightarrow BC \Rightarrow A \rightarrow B, A \rightarrow C$
  - Aditiva: se  $A \rightarrow B, A \rightarrow C \Rightarrow A \rightarrow BC$
  - Transitiva: se  $A \rightarrow B, B \rightarrow C \Rightarrow A \rightarrow C$
  - Pseudo-Transitiva: se  $A \rightarrow B, BC \rightarrow D \Rightarrow AC \rightarrow D$

Observação:  $AB$  representa  $\{A, B\}$ .

---

## Controlando a consistência

- Na construção de um SGBD baseado no modelo relacional:
  - Definição das relações baseada na análise de DFs;
  - Formas normais;
  - Uma relação está em uma determinada *forma normal* quando satisfaz certas propriedades baseadas nas DFs;
  - Colocar uma relação em uma forma normal  $\Rightarrow$  **Normalização**.

---

# Normalização

- Normalização de Relações:
  - Baseada nas DFs;
  - Garante consistência na construção do sistema:
    - redução de anomalias.
    - redução de redundância;
- Formas Normais (FNs) baseadas em DFs:
  - baseadas em chave primária: 2ª FN, 3ª FN;
  - baseadas em chaves candidatas: FN de Boyce-Codd (FNBC ou, em Inglês, BCNF).
- FN baseada em dependências multivaloradas:
  - 4ª FN.



# Definições iniciais

- Dados os conjuntos de atributos  $\mathbf{X}$  e  $\mathbf{Y}$ , e um atributo  $A \in \mathbf{X}$  :
  - $\mathbf{X} \rightarrow \mathbf{Y}$  é dependência funcional parcial se  $(\mathbf{X} - \{A\}) \rightarrow \mathbf{Y}$ .
  - $\mathbf{X} \rightarrow \mathbf{Y}$  é dependência funcional total se  $(\mathbf{X} - \{A\}) \twoheadrightarrow \mathbf{Y}$ .
  - $\mathbf{X} \rightarrow \mathbf{Y}$  é uma **dependência funcional trivial** se  $\mathbf{Y} \subseteq \mathbf{X}$ .
  - $\mathbf{X} \rightarrow \mathbf{Y}$  é uma **dependência funcional transitiva** se existe  $\mathbf{X} \rightarrow \mathbf{Z}$  e  $\mathbf{Z} \rightarrow \mathbf{Y}$ , e  $\mathbf{Z}$  não é parte da chave primária.
  - Atributo primário: atributo que faz parte de alguma chave candidata em  $\mathcal{R}$ .

---

# 1ª Forma Normal (1FN)

- $\mathcal{R}$  está na 1FN se:
  - todo valor em  $\mathcal{R}$  for atômico;
    - $\mathcal{R}$  não contém grupos de repetição.
  
- Considerações:
  - geralmente considerada parte da definição de  $\mathcal{R}$ ;
  - não permite atributos multivalorados, compostos ou suas combinações.

---

## Atributos Multivalorados e Compostos (lembrete)

- Atributos multivalorados:
  - cor do carro;
  - título acadêmico, etc.
- Atributos compostos:
  - endereço {rua, número, ap.}, etc.
- *1FN* não permite tais atributos, nem suas combinações.

# 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

---

# 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:
      - {chave primária da relação original, 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*.
-

---

# Métodos para corrigir o problema ...

- Voltando ao caso em estudo:
  - cliente (nro\_cli, nome, {end\_entrega})

## *Corrigindo o problema ...*

- Solução 1:
    - cliente\_nome (nro\_cli, nome);
    - cliente\_entrega (nro\_cli, rua, numero).
  - Solução 2:
    - cliente (nro\_cli, nome, rua, numero).
  - Solução 3:
    - cliente (nro\_cli, nome, rua1, numero1, rua2, numero2).
-

## Outros exemplos

Aluno = {Nome, Idade, ~~DataNasc.~~, ~~DataMatricula~~}

Aluno = {Nome, Idade, DiaN, MesN, AnoN, DiaM, MesM, AnoM}

Aluno = {NUSP, Idade, Disciplinas}

Aluno = {NUSP, Idade}

Disciplinas = {NUSP, Disciplina}



---

## Exercício

1. Considere a relação emp\_proj (nro\_emp, nome\_emp, { projeto ( nro\_proj, nome\_proj ) }). Como normalizá-la para a 1FN?

□ Representação:

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

## 2ª Forma Normal (2FN)

- **Definição.** O esquema de relação  $\mathcal{R}$  está na 2FN se todo atributo não primário\*  $A$  em  $\mathcal{R}$  tem dependência funcional total da chave primária de  $\mathcal{R}$ .
  - 1FN;
  - $\mathbf{X} \rightarrow \mathbf{A}$  é dependência funcional total se  $(\mathbf{X} - \{\mathbf{B}\})$  não determina funcionalmente  $\mathbf{A}$  para qualquer atributo  $\mathbf{B} \in \mathbf{X}$ .
  - “Teste para 2FN”: verificar se atributos do lado esquerdo das DFs fazem parte da chave primária. Exemplos:

Pedido (nro-pedido, data, nro-peça, descrição, qtdade\_comprada, preço\_cotado)

- nro-pedido  $\rightarrow$  data
- nro-peça  $\rightarrow$  descrição
- {nro-pedido, nro-peça}  $\rightarrow$  {qtdade\_comprada, preço\_cotado}

---

\*Atributo é dito *primário* quando é membro de uma chave candidata.

## Slide 26

---

**i1** Caso XY e XZ forem chaves candidatas, Y pode determinar Z...  
imagem-ICMC; 17/10/2008

## 2FN ...

- Para corrigir o problema:
  - Para cada sub-conjunto de atributos da chave primária, gerar uma relação com esse sub-conjunto como sua chave primária;
  - Incluir os atributos da relação original na relação correspondente à chave primária apropriada:
    - colocar cada atributo junto com a coleção mínima da qual ele depende, atribuindo um nome a cada relação.
- Levando em conta nosso exemplo anterior:
  - Pedido (nro-pedido, data, nro-peça, descrição, qtdade\_comprada, preço\_cotado)
    - pedido (nro-pedido, data)
    - peça (nro\_peça, descrição)
    - pedido\_peça (nro\_pedido, nro\_peça, qtdade\_comprada, preço\_cotado)

---

## 2FN ...

- Outro exemplo:

- DFs identificadas pelo desenvolvedor:

- {Professor, Sigla} → LivroTexto;
    - {NúmeroT, Sigla} → Sala;
    - Sigla → No.Horas;
    - LivroTexto → LivroExerc.

- Ministra={Professor, Sigla, LivroTexto, LivroExerc}

- Está na 2FN, mesmo que LivroTexto → LivroExerc.

- Turma={NúmeroT, Sigla, Sala, No.Horas}

- Viola a 2FN, pois Sigla → No. Horas.
-

---

## 2FN ...

- Corrigindo o problema para atender à 2FN:

- Turma={NúmeroT, Sigla, Sala, No.Horas};

- {NúmeroT, Sigla} → Sala;

- Sigla → No.Horas;

Então:

- Turma = {NúmeroT, Sigla, Sala};

- Disciplina = {Sigla, No.Horas}.

- 2FN evita:

- Inconsistência e anomalias causadas por redundância de informação;

- Perda de informação em operações de remoção/alteração na relação.

---

---

## 3ª Forma Normal (3FN)

- **Definição.**  $\mathcal{R}$  está na 3FN se:
  - (i) Está na 2FN;
  - (ii) Nenhum atributo não primário de  $\mathcal{R}$  for transitivamente dependente da chave primária.
  
- Dependência transitiva:
  - Dependência transitiva  $\mathbf{X} \rightarrow \mathbf{Y}$  em  $\mathcal{R}$  acontece se:
    - (i)  $\mathbf{X} \rightarrow \mathbf{Z}$  e  $\mathbf{Z} \rightarrow \mathbf{Y}$  e;
    - (ii)  $\mathbf{Z}$  não for chave candidata nem subconjunto de qualquer chave de  $\mathcal{R}$ .

---

## 3FN ...

- Em outras palavras, todos os atributos não primários devem possuir dependência total, não transitiva, da chave primária.
- Se  $X \rightarrow Y$  é não transitiva, então não pode haver no conjunto de DFs:  $X \rightarrow Z$  e  $Z \rightarrow Y$ .
- Exemplo:
  - cliente (nro-cliente, nome-cliente, end-cliente, nro-vendedor, nome-vendedor);
    - nro-vendedor  $\rightarrow$  nome\_vendedor.



## 3FN ...

### ■ Corrigindo 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.
- ❑ Levando em conta nosso exemplo anterior:
  - cliente (nro-cliente, nome-cliente, end-cliente, nro-vendedor, nome-vendedor):
    - ❑ cliente (nro-cliente, nome-cliente, end-cliente, nro-vendedor);
    - ❑ vendedor (nro-vendedor, nome-vendedor).

Chave  
estrangeira

---

## 3FN ...

- Assim como a 2FN, a 3FN evita:
    - Inconsistência e anomalias causadas por redundância de informações;
    - Perda de informação em operações de remoção/alterações na relação.
-

---

## Definições Gerais de 2FN e 3FN

- Definição de 1FN não é diretamente dependente do conceitos de chaves e de DFs;
  - 2FN e 3FN discutidas até agora desaprovam somente dependências parciais e transitivas em relação à chave primária;
  - Definições gerais levam em conta todas as chaves candidatas de uma relação.
-

## Definição geral de 2FN

- $\mathcal{R}$  está na 2FN se cada atributo não primário\* de  $\mathcal{R}$  não for parcialmente dependente de nenhuma chave em  $\mathcal{R}$ .
- Alternativamente:  $\mathcal{R}$  está na 2FN se todo atributo não primário  $A$  de  $\mathcal{R}$  possuir dependência funcional total de cada chave do esquema  $\mathcal{R}$ .

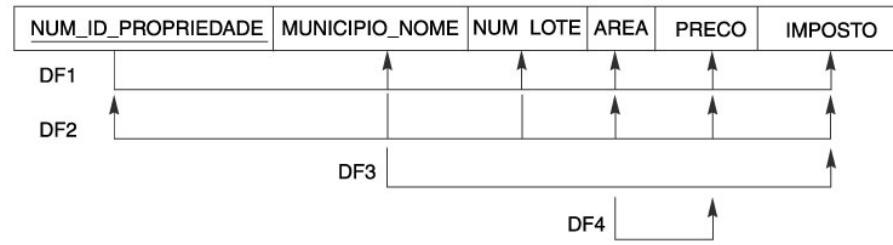
\* Atributo primário é um atributo que faz parte de qualquer chave candidata.

---

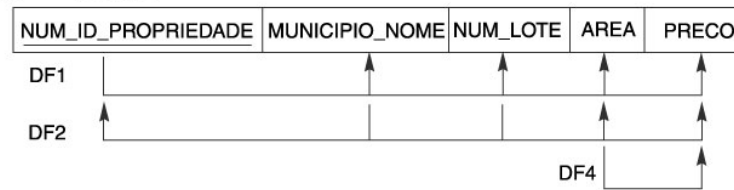
# Definição geral de 3FN

- Um esquema de relação  $\mathcal{R}$  está na 3FN se para cada dependência funcional  $\mathbf{X} \rightarrow A$ ,  $\mathbf{X}$  é uma superchave de  $\mathcal{R}$  ou  $A$  é um atributo primário de  $\mathcal{R}$ .
- Alternativamente, um esquema de relação  $\mathcal{R}$  está na 3FN se todo atributo não primário apresentar ambas as seguintes condições:
  - Ter dependência funcional total para todas as chaves (2FN);
  - Não ser transitivamente dependente de nenhuma chave.
- Ilustrando as definições gerais de 2FN e 3FN:

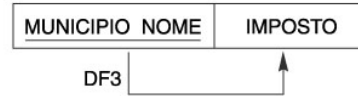
(a) LOTES



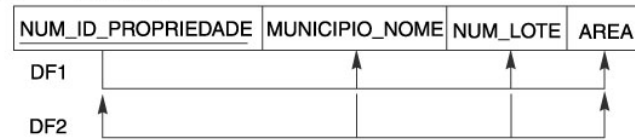
(b) LOTES1



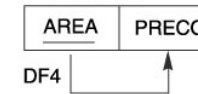
LOTES2



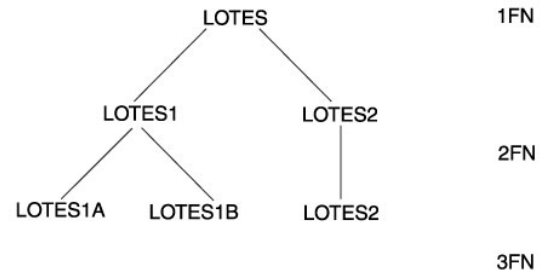
(c) LOTES1A



LOTES1B



(d)



## Exercícios

Nos exercícios seguintes, normalize as relações de forma que todas as relações resultantes estejam na forma normal mais restrita. Considere a 1FN, a 2FN e a 3FN. Para cada FN:

- ❑ Se necessário, identifique quais as dependências funcionais que se aplicam sobre  $\mathcal{R}$ ;
- ❑ Identifique e justifique se  $\mathcal{R}$  encontra-se ou não na forma normal em questão; e
- ❑ Caso  $\mathcal{R}$  sendo analisada não se encontre na forma normal em questão, normalize-a, especificando as relações originadas.

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

As seguintes dependências funcionais devem ser garantidas na normalização:

- ❑ nro\_vend  $\rightarrow$  nome\_vend;
- ❑ nro\_cli  $\rightarrow$  nome\_cli.

Observação: considere que um vendedor pode atender diversos clientes, e um cliente pode ser atendido por diversos vendedores.

---

2) aluno ( nro\_aluno, cod\_depto, nome\_depto, sigla\_depto, cod\_orient, nome\_orient, fone\_orient, cod\_curso )

As seguintes dependências funcionais devem ser garantidas na normalização:

- ❑ cod\_depto → {nome\_depto, sigla\_depto};
- ❑ cod\_orient → {nome\_orient, fone\_orient};
- ❑ nro\_aluno → {cod\_depto, cod\_orient, cod\_curso};

Observações adicionais:

- ❑ um aluno somente pode estar associado a um departamento;
- ❑ um aluno cursa apenas um único curso;
- ❑ um aluno somente pode ser orientado por um único orientador.



---

3) aluno ( nro\_aluno, nome\_aluno, { curso (nro\_curso, descrição\_curso, ano\_ingresso, nro\_depto, nome\_depto) })

As seguintes dependências funcionais devem ser garantidas na normalização:

- ❑ nro\_aluno → nome\_aluno;
- ❑ nro\_curso → descrição\_curso;
- ❑ nro\_depto → nome\_depto;
- ❑ {nro\_aluno, nro\_curso} → ano\_ingresso;
- ❑ nro\_curso → nro\_depto.

Observações adicionais:

- ❑ um aluno pode cursar mais do que um curso;
- ❑ um curso somente pode ser oferecido por um único departamento.

---

## Forma Normal de Boyce-Codd (FNBC)

- **Definição.**  $\mathcal{R}$  está na FNBC se para cada dependência funcional  $\mathbf{X} \rightarrow A$ ,  $\mathbf{X}$  é uma superchave de  $\mathcal{R}$ .
  - Diferença entre FNBC e 3FN:
    - 3FN permite A primário – não se aplica à FNBC.
    - Se  $\mathcal{R}$  está na FNBC  $\rightarrow \mathcal{R}$  está na 3FN;
    - Se  $\mathcal{R}$  está na 3FN, não necessariamente  $\mathcal{R}$  está na FNBC.
  - Na prática, a maioria dos esquemas de relação que está na 3FN também está na FNBC.
-

# Re-visitando exemplo anterior (lotes):

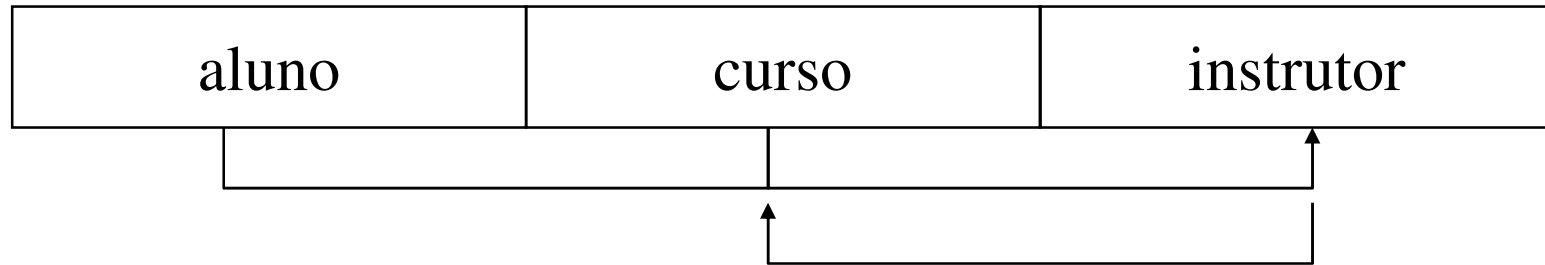
- Lotes1A{num\_id\_propriedade, município\_nome, num\_lote, area}.
- Lotes1B{area, preço}
  - Supor haver milhares de lotes de 2 municípios ( $x$  e  $y$ ) e;
  - $x$ : área  $[0,1]$  e  $y$ : área  $(1,2]$ ;
  - Atende 3FN, mesmo com a nova DF:  
área→município\_nome. /\* município\_nome é primário \*/
  - Nova tabela com áreas e número do município economizaria espaço.
  - → FNBC.

---

## Normalizando pela FNBC ...

- Lotes1AX{num\_id\_propriedade, área, num\_lote};
  - Lotes1AY{área, município\_nome};
  - Lotes1B{área, preço}
- 
- Decompor relações;
  - Reunir Lotes1AY e Lotes1B causaria um desperdício de espaço de armazenamento, pois a maioria das tuplas desta relação possuiria somente dois valores para município\_nome, a saber:  $x$  e  $y$ .

## Outro exemplo de FNBC ...



- $\mathcal{R}(\underline{\text{aluno}}, \underline{\text{curso}}, \text{instrutor})$
- DFs:
  - $\{\text{aluno}, \text{curso}\} \rightarrow \text{instrutor}$ ;
  - $\text{instrutor} \rightarrow \text{curso}$ .
    - essa dependência, que representa que cada instrutor ministra um curso, é uma restrição particular da aplicação;
    - “instrutor” não é superchave. Logo essa DF viola FNBC.

---

## FNBC ...

### ■ 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

---

---

## Considerações sobre DFs e Normalização...

- Normalização:

- uma relação por vez;
- FN de uma relação  $\Rightarrow$  formal normal mais restrita atendida;
- Decompor relações, criando outras relações;

$\Rightarrow$  Propriedades desejáveis:

- decomposição sem perda de junção (sem geração de tuplas ilegítimas);
- decomposição com preservação de dependências.
- aumenta consistência, mas reduz desempenho (junções).

\* Outras formas normais: 4FN, 5FN (Elmasri e Navathe, Sistemas de Banco de Dados, 2005)

---

# Slides Extras

- Dependência multivalorada;
- 4FN;



---

# Dependência Multivalorada e Normalização

- DF: mecanismo formal para definição de restrições e garantia de consistência em bases de dados relacionais;
- Entretanto, algumas restrições não podem ser especificadas com DFs.
  - Exemplo: informação sobre empregados de uma empresa. {nome do empregado, projetos, dependentes}.
  - Semanticamente:
    - um conjunto de valores de projeto é determinado por um valor de nome, e somente por nome.
    - projeto e dependente não têm relação alguma.

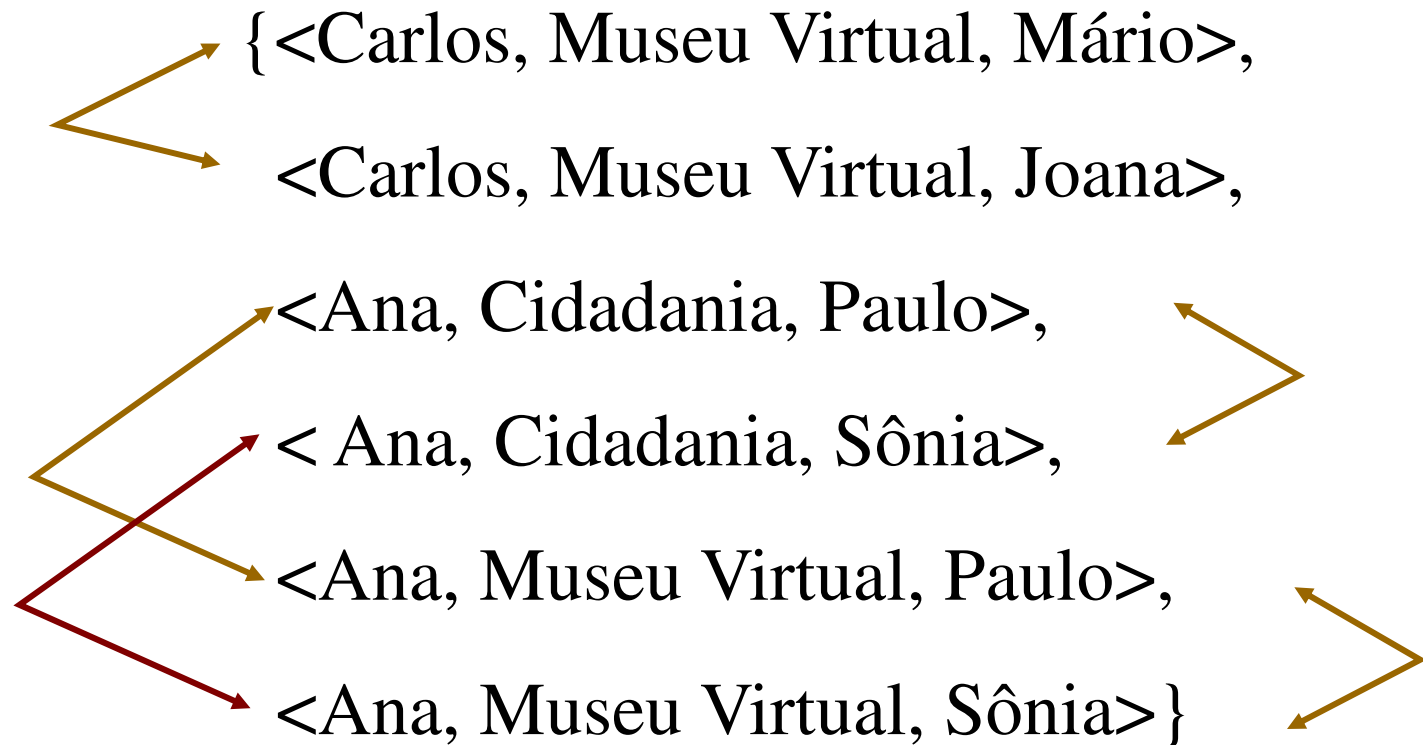
---

## Dependência Multivalorada ...

- Dependência Multivalorada (DM): restrição entre dois conjuntos de atributos.
- **A** multidetermina **B** (ou **B** é multidependente de **A**)  $\Rightarrow$  o conjunto de valores de **B** é determinado pelo valor de **A**, e somente pelo valor de **A**.
- Exemplo para Empregado={ Nome, Projeto, Dependente };
  - Carlos trabalha no projeto Museu Virtual e tem dois dependentes: Mário e Joana;
  - Ana trabalha nos projetos Museu Virtual e Cidadania, e tem dois dependentes: Paulo e Sônia;
    - Como armazenar os dados na relação Empregado de maneira a manter a semântica?

---

Empregado = { Nome, Projeto, Dependente }



**Dependência Multivalorada**

---

---

## Dependência Multivalorada ...

- Ocorrem quando atributos multivalorados são desmembrados em múltiplas ocorrências de tuplas por causa da 1FN;
- Identificadas pelo projetista da base de dados;
- Problemas:
  - Redundância nas tuplas;
  - Como garantir consistência?
  - Exemplo:
    - Empregado={Nome, Projeto, Dependente}
    - Está na FNBC, mas ainda vulnerável a inconsistências....

---

## 4ª Forma Normal (4FN)

- Um esquema de relação está na 4FN se:
  - todas as DMs são triviais **ou**;
  - para cada DM não-trivial  $\mathbf{A} \twoheadrightarrow \mathbf{B}$ ,  $\mathbf{A}$  é uma superchave em  $\mathcal{R}$ .
- Exemplos:
  - Empregado={Nome, Projeto}
  - Nome  $\twoheadrightarrow$  Projeto (trivial)
  
  - Empregado={Nome, Projeto, Dependente}
  - Nome  $\twoheadrightarrow$  Projeto
  - Nome  $\twoheadrightarrow$  Dependente

## 4ª Forma Normal (4FN) ...

- Colocando a relação na 4FN....

Nome  $\rightarrow$  Projeto

Nome  $\rightarrow$  Dependente

~~Empregado = { Nome, Projeto, Dependente }~~

Dependentes = { Nome, Dependente }

Projetos = { Nome, Projeto }

---

## 4ª Forma Normal (4FN) ...

### **Outro exemplo:**

Professor = { Nome, Programa, Orientado }

Nome -> Programa

Nome -> Orientado

Programa = { Nome, Programa }

Orientação = { Nome, Orientado }

---

## 4ª Forma Normal (4FN) ...

- Evita redundância nas tuplas  $\Rightarrow$  evita inconsistências causadas por inclusão/remoção/alteração de tuplas;
- Normalização é importante quando atributos multivalorados independentes são misturados na mesma relação:
  - Reduz espaço de armazenamento;
  - Mais restrita que FNBC;
  - Propriedade desejada: decomposição sem perda de junção.



---

## Considerações Finais - Normalização

- 1FN, 2FN, 3FN, BCNF e 4FN são consideradas para cada relação:
  - BD é considerada normalizada para uma determinada FN quando todas as suas relações estiverem nessa FN;
- Normalização  $\Rightarrow$  decomposição de relações:
  - aumenta consistência;
  - reduz desempenho  $\Rightarrow$  operações de junção.