



Banco de Dados

Dependência Funcional, Dependência Multivalorada e Normalização

* Slides Baseados no material elaborado pelos professores Eduardo R. Hruschka, Cristina D.A. Ciferri e Elaine Parros Machado

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

(cont)...

Exemplo...

- **Emp_Dept**={ Nome, CPF, DataNasc, End, Dnum, Dnome, DGerCPF}
 - 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

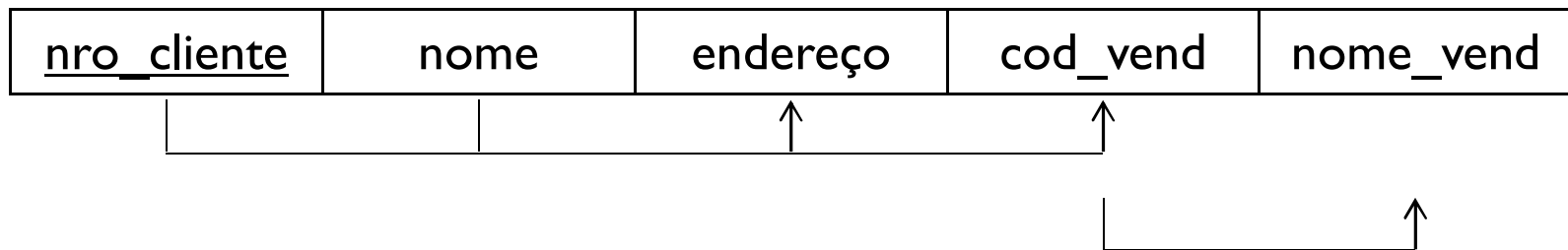
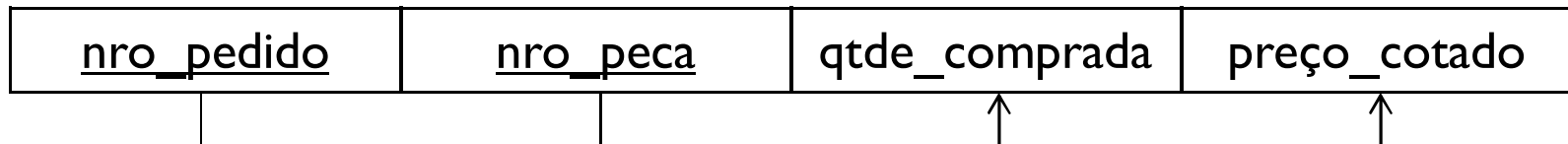
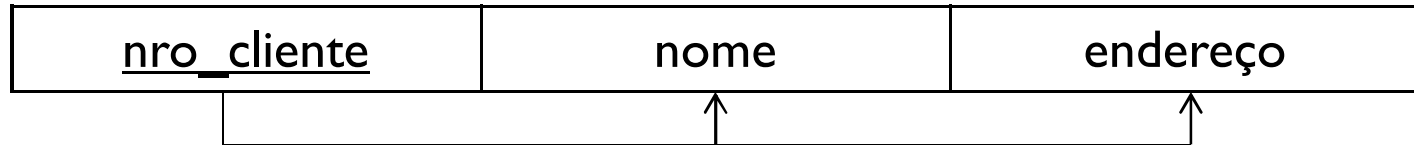
Qualidade do Projeto Lógico ...

- **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 **A** \rightarrow **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:
 - $\{\#UFF\} \rightarrow \{\text{Nome, Idade, Curso}\}$
 - $\{\text{Sigla, Sala, Hora}\} \rightarrow \{\text{CódigoTurma, Professor}\}$
 - $\{\text{Sigla}\} \rightarrow \{\text{NomeDisciplina, NCréditos}\}$

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, Eng. Prod. 22>,
 <Almir, Enf., 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, Eng. Prod. 22>,
<Almir, Enf., 22>,
<Marta, Comp., 21>,
<Vânia, Eletr., 22>

Exercícios

- Dada a relação Cliente ($n_cliente$, nome, endereço), as seguintes dependências são corretas?
 - $n_cliente \rightarrow nome$;
 - $n_cliente \rightarrow endereço$;
 - $nome \rightarrow endereço$;
 - $endereço \rightarrow nome$.
- 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

- nro_pedido → qtidade_comprada;
- nro_peça → qtidade_comprada;
- nro_pedido → preço_cotado;
- nro_peça → preço_cotado;
- {nro_pedido, nro_peça} → qtidade_comprada;
- {nro_pedido, nro_peça} → preço_cotado;
- {nro_pedido, nro_peça} → {qtidade_comprada, 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 \Rightarrow 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 ⇒ **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.

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
- IFN 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 I:
 - 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 = {NUFF, Idade, ~~Disciplinas~~}

Aluno = {NUFF, Idade}

Disciplinas = {NUFF, Disciplina}

Exercício

- Considere a relação emp_proj (nro_emp, nome_emp, {projeto (nro_proj, nome_proj) }).
Como normalizá-la para a 1FN?
 - { } 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 $(\mathbf{X} - \{\mathbf{B}\})$ 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, qtidade_comprada, preço_cotado)
 - nro-pedido \rightarrow data
 - nro-peça \rightarrow descrição
 - {nro-pedido, nro-peça} \rightarrow {qtidade_comprada, preço_cotado}
 - Obs: Caso XY e XZ forem chaves candidatas, Y pode determinar Z ...

* Atributo é dito primário se é membro de uma chave candidata

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, qtidade_comprada, preço_cotado)
 - pedido (nro-pedido, data)
 - peça (nro_peça, descrição)
 - pedido_peça (nro_pedido, nro_peça, qtidade_comprada, preço_cotado)

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 $\mathbf{X} \rightarrow \mathbf{Y}$ é não transitiva, então não pode haver no conjunto de DFs: $\mathbf{X} \rightarrow \mathbf{Z}$ e $\mathbf{Z} \rightarrow \mathbf{Y}$.
- Exemplo:
 - cliente (nro-cliente, nome-cliente, end-cliente, nro- nrovendedor, 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, nomevendedor):
 - 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 dos 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} .

Definição geral de 3FN

- Um esquema de relação R está na 3FN se para cada dependência funcional $\mathbf{X} \rightarrow \mathbf{A}$, \mathbf{X} é uma superchave de R ou \mathbf{A} é um atributo primário de R .
- Alternativamente, um esquema de relação 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:

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 R;
 - Identifique e justifique se R encontra-se ou não na forma normal em questão; e
 - Caso R sendo analisada não se encontre na forma normal em questão, normalize-a, especificando as relações originadas.

Exercício I

- 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.

Exercício 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:
 - $\text{cod_depto} \rightarrow \{\text{nome_depto}, \text{sigla_depto}\};$
 - $\text{cod_orient} \rightarrow \{\text{nome_orient}, \text{fone_orient}\};$
 - $\text{nro_aluno} \rightarrow \{\text{cod_depto}, \text{cod_orient}, \text{cod_curso}\};$
 - 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.

Exercício 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 \rightarrow nome_aluno;
 - nro_curso \rightarrow descrição_curso;
 - nro_depto \rightarrow nome_depto;
 - {nro_aluno, nro_curso} \rightarrow ano_ingresso;
 - nro_curso \rightarrow 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 $\mathbf{X} \rightarrow \mathbf{A}$, \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.

Possível instância de Filmes1

nome	ano	duração	nomeEstúdio	moradaEstúdio
Star Wars	1977	124	Fox	10 Elm St., Los Angeles
Empire Strikes Back	1980	143	Fox	10 Elm St., Los Angeles
Gone With the Wind	1939	181	Paramount	44 Pine St., Los Angeles
Lion King	1994	124	Disney	56 Oak St., Los Angeles
Return of the Jedi	1983	165	Fox	10 Elm St., Los Angeles
Pocahontas	1995	115	Disney	56 Oak St., Los Angeles

Problemas

1. Redundância

- ▶ a morada de um estúdio aparece repetida várias vezes.

2. Pode dar origem a anomalias em “updates”.

- ▶ ex: actualizar a morada no 1º tuplo e não o fazer no 2º e 5º tuplo.

3. Pode dar origem a anomalias em “deletes”.

- ▶ ex: se apagarmos o filme “Gone With the Wind”, o estúdio “Paramount” desaparece da BD!

Exemplo

Filmes1(nome ano duração nomeEstúdio moradaEstúdio)

nome	ano	duração	nomeEstúdio	moradaEstúdio
Star Wars	1977	124	Fox	10 Elm St., Los Angeles
Empire Strikes Back	1980	143	Fox	10 Elm St., Los Angeles
Gone With the Wind	1939	181	Paramount	44 Pine St., Los Angeles
Lion King	1994	124	Disney	56 Oak St., Los Angeles
Return of the Jedi	1983	165	Fox	10 Elm St., Los Angeles
Pocahontas	1995	115	Disney	56 Oak St., Los Angeles

- ▶ Pode ser decomposto em:
 - ▶ Filmes2(nome ano duração nomeEstúdio)
 - ▶ Filmes3(nomeEstúdio moradaEstúdio)

Exemplo (cont.)

Filmes2:

nome	ano	duração	nomeEstúdio
Star Wars	1977	124	Fox
Empire Strikes Back	1980	143	Fox
Gone With the Wind	1939	181	Paramount
Lion King	1994	124	Disney
Return of the Jedi	1983	165	Fox
Pocahontas	1995	115	Disney

Filmes3:

nomeEstúdio	moradaEstúdio
Fox	10 Elm St., Los Angeles
Paramount	44 Pine St., Los Angeles
Disney	56 Oak St., Los Angeles

Forma normal de Boyce-Codd (BCNF)

Uma relação R está em **BCNF** sse:

- ▶ para todas as DF's de R não triviais $X \rightarrow Y$,
 X for superchave de R .

(basta existir uma DF em que X não seja superchave, para R não estar em BCNF. Se isso acontecer, diz-se que essa DF viola a condição de BCNF).

Exemplo: Filmes1

Filmes1(nome ano duração nomeEstúdio moradaEstúdio)

com DF's:

nome ano \rightarrow duração

nome ano \rightarrow nomeEstúdio

nome ano \rightarrow moradaEstúdio

nomeEstúdio \rightarrow moradaEstúdio

- ▶ A chave é {nome,ano} porque:
 - ▶ nome ano \rightarrow duração nomeEstúdio moradaEstúdio
- ▶ Filmes1 não está em BCNF porque:
 - ▶ {nomeEstúdio} não é superchave de Filmes1.
- ▶ nomeEstúdio \rightarrow moradaEstúdio, viola a condição BCNF.

Outro exemplo: Filmes2

Filmes2(nome ano duração nomeEstúdio)

com DF's:

nome ano \rightarrow duração

nome ano \rightarrow nomeEstúdio

- ▶ A chave é {nome,ano} porque:
 - ▶ nome ano \rightarrow duração nomeEstúdio
- ▶ Filmes2 está em BCNF porque em todas as DF's, o lado esquerdo contém a chave.
 - ▶ {nome,ano} é superchave de Filmes2.

Outro exemplo: Filmes3

Filmes3(nomeEstúdio moradaEstúdio)

com DF's:

nomeEstúdio \rightarrow moradaEstúdio

- ▶ A chave é {nomeEstúdio} porque:
 - ▶ nomeEstúdio \rightarrow moradaEstúdio
- ▶ Filmes3 está em BCNF porque {nomeEstúdio} é superchave de Filmes3.

Possíveis instâncias de Filmes2 e Filmes3

Filmes2:

nome	ano	duração	nomeEstúdio
Star Wars	1977	124	Fox
Empire Strikes Back	1980	143	Fox
Gone With the Wind	1939	181	Paramount
Lion King	1994	124	Disney
Return of the Jedi	1983	165	Fox
Pocahontas	1995	115	Disney

Filmes3:

nomeEstúdio	moradaEstúdio
Fox	10 Elm St., Los Angeles
Paramount	44 Pine St., Los Angeles
Disney	56 Oak St., Los Angeles

Decomposição de relações para BCNF

- ▶ Não é qualquer decomposição que resolve o problema.
- ▶ Exemplo: Decompor Filmes1 em:
 - ▶ Filmes4(nome ano duração)
 - ▶ Filmes5(duração nomeEstúdio moradaEstúdio)

Resultado

Filmes4:

nome	ano	duração
Star Wars	1977	124
Empire Strikes Back	1980	143
Gone With the Wind	1939	181
Lion King	1994	124
Return of the Jedi	1983	165
Pocahontas	1995	115

Filmes5:

duração	nomeEstúdio	moradaEstúdio
124	Fox	10 Elm St., Los Angeles
143	Fox	10 Elm St., Los Angeles
181	Paramount	44 Pine St., Los Angeles
124	Disney	56 Oak St., Los Angeles
165	Fox	10 Elm St., Los Angeles
115	Disney	56 Oak St., Los Angeles

Qual o problema?

1. Filmes5 continua sem estar em BCNF.
2. Mais grave: se juntarmos Filmes4 com Filmes5, vamos obter tuplos que não pertenciam à relação original!

nome	ano	duração	nomeEstúdio	moradaEstúdio
Star Wars	1977	124	Fox	10 Elm St., Los Angeles
Star Wars	1977	124	Disney	56 Oak St., Los Angeles
Empire Strikes Back	1980	143	Fox	10 Elm St., Los Angeles
Gone With the Wind	1939	181	Paramount	44 Pine St., Los Angeles
Lion King	1994	124	Disney	56 Oak St., Los Angeles
Lion King	1994	124	Fox	10 Elm St., Los Angeles
Return of the Jedi	1983	165	Fox	10 Elm St., Los Angeles
Pocahontas	1995	115	Disney	56 Oak St., Los Angeles

Exercício

- Colocar na FNBC

ABASTECIMENTO
cd_Bomba
qt_Capacidade_Bomba
cd_Funcionário
nm_Funcionário
dt_Abastecimento
hr_Abastecimento
cd_Placa_Veículo
ds_Veículo
qt_Kilometragem_Veículo
qt_Litros_Abastecido

Chaves Candidatas

cd_Placa_Veículo + dt_Abastecimento+ hr_Abastecimento
cd_Bomba+ dt_Abastecimento+ hr_Abastecimento

Considerações sobre DFs e Normalização...

- Normalização:
 - uma relação por vez;
 - FN de uma relação
 - forma normal mais restrita atendida;
 - Decompor relações, criando outras relações;
- Propriedades desejáveis:
 - decomposição sem perda de junção (sem geração de tuplas ilegítimas);
 - decomposição com preservação de dependências.
 - aumenta consistência, mas reduz desempenho (junções).

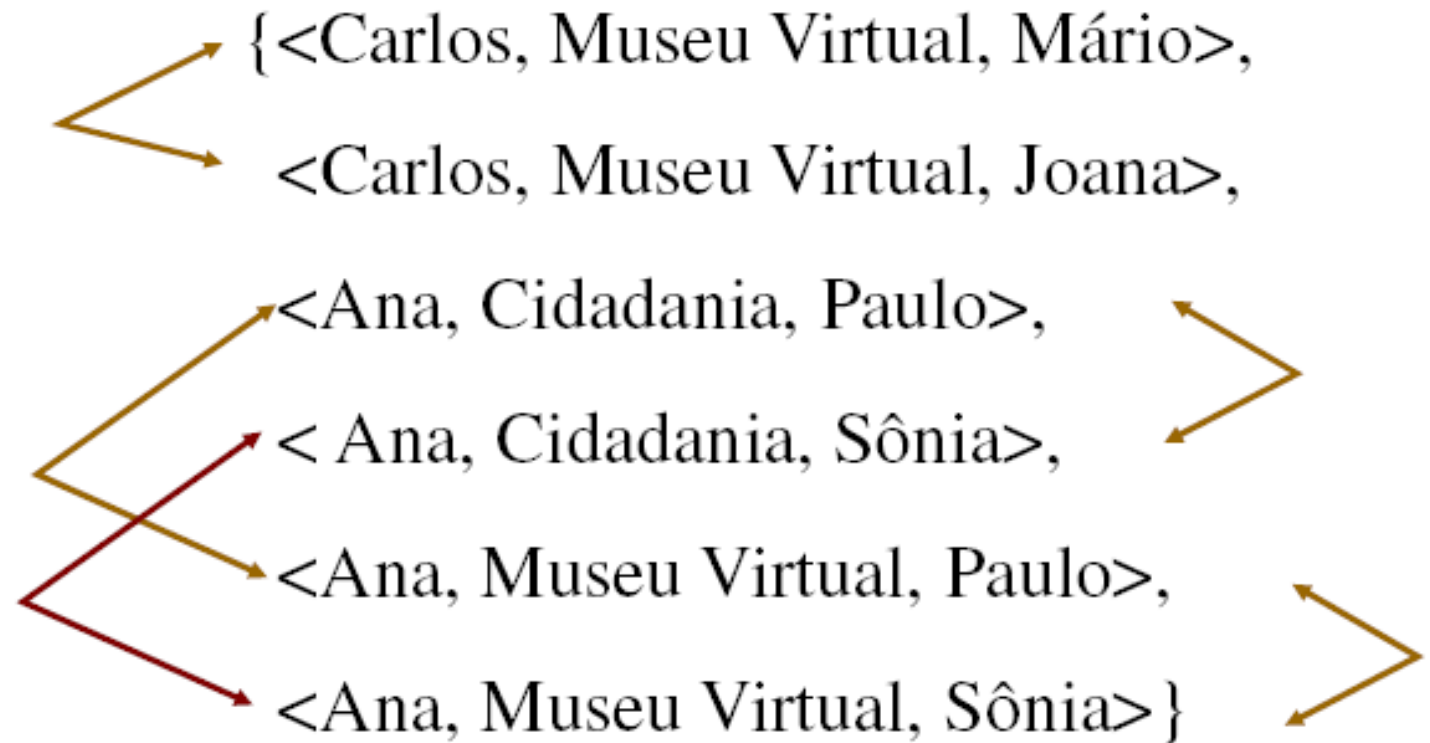
Dependência Multivalorada e Normalização

- 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**)
 - 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 IFN
- 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....

4^a Forma Normal (4FN)

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

 - Empregado={Nome, Projeto, Dependente}
 - Nome ->> Projeto
 - Nome ->> Dependente

4a Forma Normal (4FN) ...

- Colocando a relação na 4FN...

Nome \twoheadrightarrow Projeto

Nome \twoheadrightarrow 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
 - 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

- FN, 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: decomposição de relações:
 - aumenta consistência;
 - reduz desempenho \Rightarrow operações de junção.

5FN – 5ª Forma Normal

- Estar na 4FN
- Não deve haver dependência de junção
 - Evitar a criação de relação PK/FK quando não for estritamente necessário

Exemplo:

Produto (idProduto, descrição, idFornecedor(FK), quantidade)

Fornecedor (idFornecedor, nome)

Nota (idNota, idVendedor(FK), idProduto(FK), idFornecedor(FK))



5FN – 5ª Forma Normal

- Solução

Produto (idProduto, descrição, idFornecedor(FK), quantidade)

Fornecedor (idFornecedor, nome)

Nota (idNota, idVendedor(FK), idProduto(FK))