

Aula 12 – BD1

Dependências Funcionais e Normalização

Profa. Elaine Faria

UFU - 2018

Refinamento de Esquema

- Problemas causados pela redundância
 - Armazenamento redundante
 - Algumas informações são armazenadas repetidamente
 - Anomalias de atualização
 - Se uma cópia dos dados redundantes é atualizada, é gerada uma inconsistência
 - Anomalias de inserção
 - Pode não ser possível armazenar certas informações, a não ser que outras informações não relacionadas sejam armazenadas
 - Anomalias de exclusão
 - Pode não ser possível excluir certas informações sem perder também algumas informações não relacionadas

Refinamento de Esquema

- Problemas causados pela redundância
 - Ex: Funcion_Horista (cpf, nome, vaga, avaliacao, salario_hora, horas_trabalhadas)
 - RI: salario_hora é determinado pela avaliação → é uma **dependência funcional**
 - Essa dependência funcional gera redundância

cpf	nome	vaga	avaliacao	salario_hora	horas_trabalhadas
123456	Artur	48	8	10	40
234567	Sandro	22	8	10	30
345678	Saulo	35	5	7	30
456789	Gustavo	35	5	7	32
567890	Manoel	35	8	10	40

Instância da relação Funcion_Horistas

- Se o mesmo valor aparece na coluna avaliacao de duas tuplas, a RI nos informa que o mesmo valor deve aparecer na coluna salario_hora
- Consequências negativas
 - Armazenamento redundante
 - Anomalias de atualização
 - Anomalias de inserção
 - Anomalias de exclusão

Refinamento de Esquema

- Valores Nulos
 - Podem tratar anomalias de inserção e de exclusão
 - Ex: Relação Funcion_Horistas
 - Uso de nulos na anomalia de inserção
 - Inserir uma tupla de funcionários com valores nulos no campo de salario_hora
 - Não pode-se registrar o salario_hora para uma avaliação a não ser que exista um funcionário com essa avaliação
→ cpf não pode ser nulo

Decomposições

- Redundância
 - Surge quando um esquema relacional impõe uma associação entre atributos que não é natural
 - Dependências funcionais podem ser usadas para identificar tais situações
 - Podem sugerir refinamentos para o esquema
- Ideia básica: muitos problemas provenientes da redundância podem ser resolvidos substituindo uma relação por um conjunto de relações “menores”

Decomposições

- Decomposição de um esquema de relação R
 - Consiste na substituição do esquema de relação por dois (ou mais) esquemas de relação, cada um contendo um subconjunto de atributos de R e, juntos, incluindo todos os atributos de R
 - Ex: Decompor Funcion_Horista em:

Funcion_Horista2(cpf, nome, vaga, avaliacao, horas_trabalhadas)

Salarios(avaliacao, salario_hora)

cpf	nome	vaga	avaliação	horas_trabalhadas
123456	Artur	48	8	40
234567	Sandro	22	8	30
345678	Saulo	35	5	30
456789	Gustavo	35	5	32
567890	Manoel	35	8	40

avaliacao	salario_hora
8	10
5	7

É possível:

- Registrar o salario_hora para qualquer avaliação
- Mudar o salario_hora associado a uma avaliacao apenas atualizando uma única tupla

Dependências Funcionais

- Um atributo Y de um esquema de relação R é funcionalmente dependente de um outro atributo X de R , se um valor de X determina um único valor de Y em qualquer momento
- Notação: $X \rightarrow Y$
- Se Y é funcionalmente dependente de X , então X determina funcionalmente Y

Dependências Funcionais

- São derivadas pelo projetista do BD na análise da especificação de requisitos
- Uma dependência funcional é uma *propriedade do esquema da relação R*, não de um estado particular válido da relação r de R
- $X \rightarrow Y$ diz que se duas tuplas concordam nos valores dos atributos X , elas também devem concordar nos valores dos atributos Y
Se $t1.X = t2.X$, então $t1.Y = t2.Y$
- Se $X \rightarrow Y$ em R , isso não implica necessariamente que $Y \rightarrow X$ em R

Dependências Funcionais

- Dependência Funcional (DF)
 - Certas DFs podem ser especificadas sem recorrer a uma relação específica, mas pelas propriedades de seus atributos
 - Os exemplos abaixo deveriam ser válidos para qualquer advogado ou engenheiro no Brasil:
 $\{\text{ESTADO, OAB}\} \rightarrow \text{NOME_ADVOGADO}$

 $\{\text{ESTADO, CREA}\} \rightarrow \text{NOME_ENGENHEIRO}$

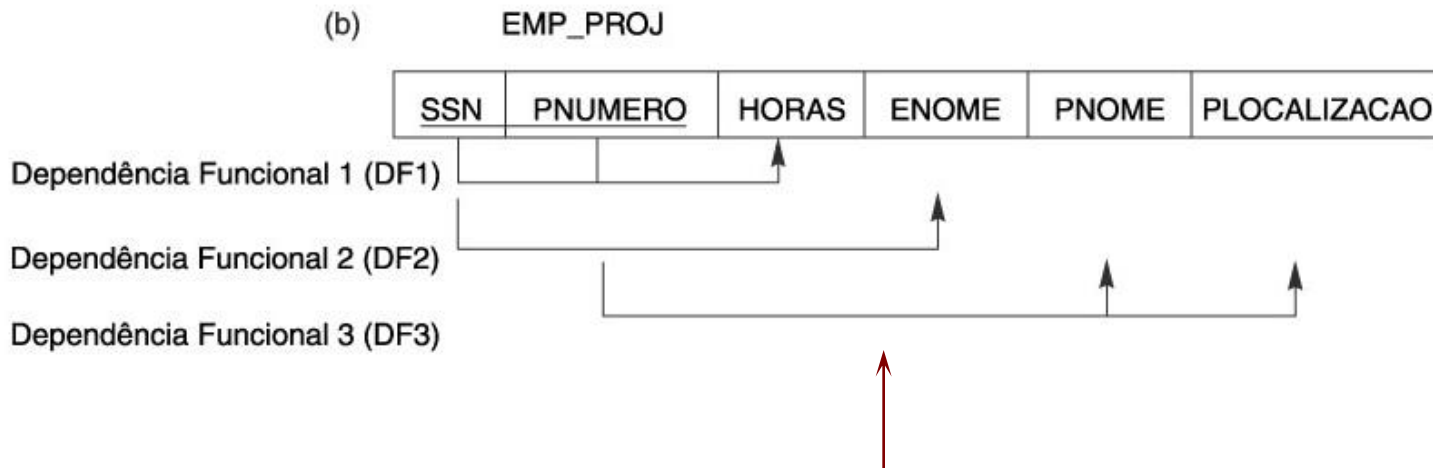
Dependências Funcionais

- Dependência Funcional (DF)
 - Também é possível que algumas DF possam deixar de existir
 - Exemplo:

PRIMEIROS_QUATRO_DÍGITOS → OPERADORA_CELULAR
com a portabilidade essa DF passou a não ser mais verdadeira

Dependência Funcional

- Exemplo:
 - {SSN, PNUMERO} → HORAS
 - SSN → ENOME
 - PNUMERO → {PNOME, PLOCALIZACAO}



Notação diagramática para DF

Dependência Funcional

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

As seguintes dependências são corretas?

- nro_cliente → nome → OK!
- nro_cliente → endereço → OK!
- nome → endereço
- endereço → nome Não!

Dependências Funcionais

DF: $AB \rightarrow C$

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2
a1	b2	c2	d1
a2	b1	c3	d1

Observações

- DF não é o mesmo que restrição de chave (vide tuplas 1 e 2)
- Se duas tuplas diferem no campo A ou B, elas podem diferir no C
- Se for adicionada a tupla (a1,b1,c2,d1), a DF é violada

Dependências Funcionais

- Restrição de chave primária
 - É um caso especial de DF
 - Os atributos na chave desempenham o papel de X e o conjunto de todos os atributos na relação desempenha o papel de Y

Obs.:

A definição de uma DF não exige que o conjunto X seja mínimo

Normalização

- Processo de Normalização
 - Inicia com um esquema de relação ou conjunto de esquemas de relação
 - Produz uma nova coleção de esquemas de relação
 - Equivalente à coleção original (representa a mesma informação)
 - Livre de problemas

Problemas relacionadas à decomposição

- É preciso decompor uma relação?
 - Várias formas normais foram propostas
 - Se a relação está em uma dessas formas normais, certos tipos de problemas não podem surgir
- Propriedades das decomposições
 - Junção sem perda → recuperar qualquer instância da relação decomposta
 - Preservação da dependência → garante que cada dependência funcional será representada em algum esquema de relação resultante da decomposição

Problemas relacionadas à decomposição

- Desempenho
 - Consultas na relação original podem exigir junção das relações decompostas → penalidade no desempenho
 - Se a maioria das consultas e atualizações examinam apenas uma das relações decompostas → pode melhorar o desempenho

Formas Normais

- Formas normais (FN)
 - Dado um esquema é necessário decidir se ele é um bom projeto ou se é preciso decompô-lo em relações menores
 - Se um esquema de relação está em uma dessas FNs sabe-se que certos tipos de problemas não podem surgir
- Formas Normais baseadas em DFs
 - Primeira forma normal (1FN)
 - Segunda forma normal (2FN)
 - Terceira forma normal (3FN)
 - Forma normal de *Boyce-Codd* (FNBC)

Revisão: Chaves e Superchaves

- Uma superchave de uma relação R é um conjunto de atributos S contido em R
 - no qual não haverá duas tuplas t_1 e t_2 cujo
$$t_1[S] = t_2[S]$$

Revisão: Chaves e Superchaves

- Uma chave K é uma superchave com a propriedade adicional de que a remoção de qualquer atributo da chave fará com que K não identifique mais unicamente cada tupla da relação
 - a diferença é que uma chave tem que ser mínima

Revisão: Chaves e Superchaves

EMPREGADO			chave estrangeira (f.k.)	
ENOME	<u>SSN</u>	DATANASC	ENDERECO	DNUMERO

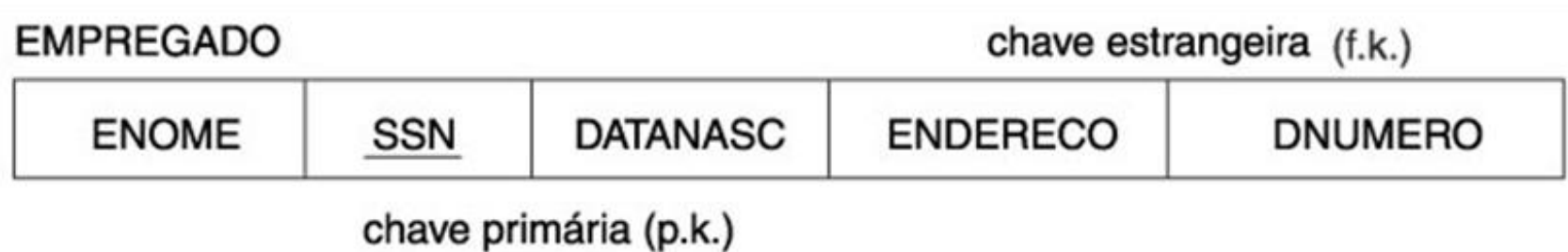
chave primária (p.k.)

- Exemplo:
 - {SSN} é uma chave de empregado
 - Superchaves
 - {SSN, Enome}
 - {SSN, Enome, Datanasc}
 - {SSN, Enome, Datanasc, Endereço}
 - {SSN, Enome, Datanasc, DNumero}

Revisão: Chaves e Superchaves

- Chave candidata:
 - Se um esquema de relação tiver mais de uma chave, cada uma delas é chamada chave candidata
 - Uma delas é arbitrariamente designada para ser chave primária
- Um atributo de um esquema de relação R é chamado atributo primário se for membro de alguma chave candidata

Revisão: Chaves e Superchaves



- Exemplo:
 - {SSN} é a única chave candidata de empregado, portanto também é a chave primária

Revisão: 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

Primeira Forma Normal (1FN)

- Uma relação está na primeira forma normal
 - Se todo campo contém apenas valores atômicos e monovalorados → nenhuma lista nem conjuntos
 - Requisito implícito na definição de modelo relacional usada nessa disciplina

Primeira Forma Normal (1FN)

- Exemplo

– cliente (nro_cli, nome, {end_entrega})

repetição

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

Cliente_nome

<u>nro_cli</u>	nome
124	João dos Santos
311	José Ferreira Neves

Cliente_entrega

<u>nro_cli</u>	<u>end_entrega</u>
124	Rua 10, 1024
124	Rua 24, 1356
311	Rua 46, 1344
311	Rua 98, 4456

Métodos para Corrigir o Problema

- Método 2
 - substituir o grupo de repetição pelo número máximo de valores estabelecido para o grupo
 - **abordagem menos genérica e que pode introduzir muitos valores *null***

<u>nro_cli</u>	nome	end_entrega1	end_entrega2
124	João dos Santos	Rua 10, 1024	Rua 24, 1356
311	José Ferreira Neves	Rua 46, 1344	Rua 98, 4456
025	Cecília Neves	Rua 77, 275	<i>null</i>

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, entrega1, entrega2)

Segunda forma normal (2FN)s

- Uma relação está na 2FN se
 - Está na 1FN
 - Todo atributo não chave deve ser totalmente dependentes da chave primária (dependente de toda chave e não de parte dela)

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

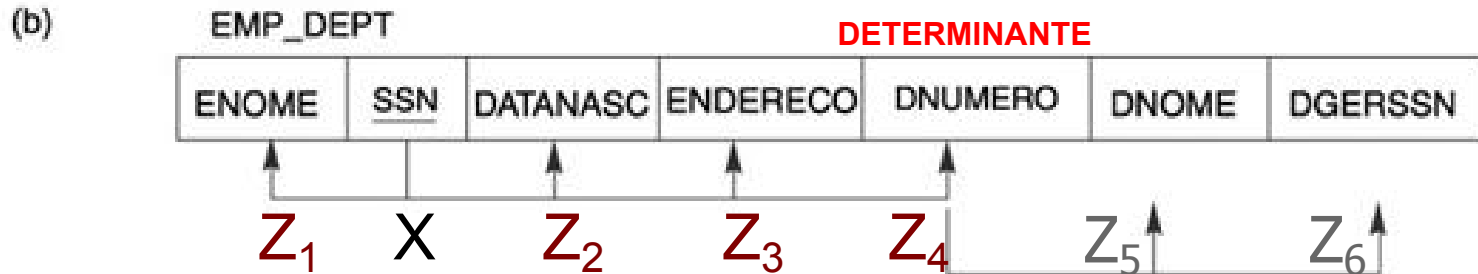
- Uma relação R está na 3FN se
 - Está na 2FN
 - Não existem atributos não chave que sejam dependentes de outros atributos não chave
 - Dependência transitiva

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



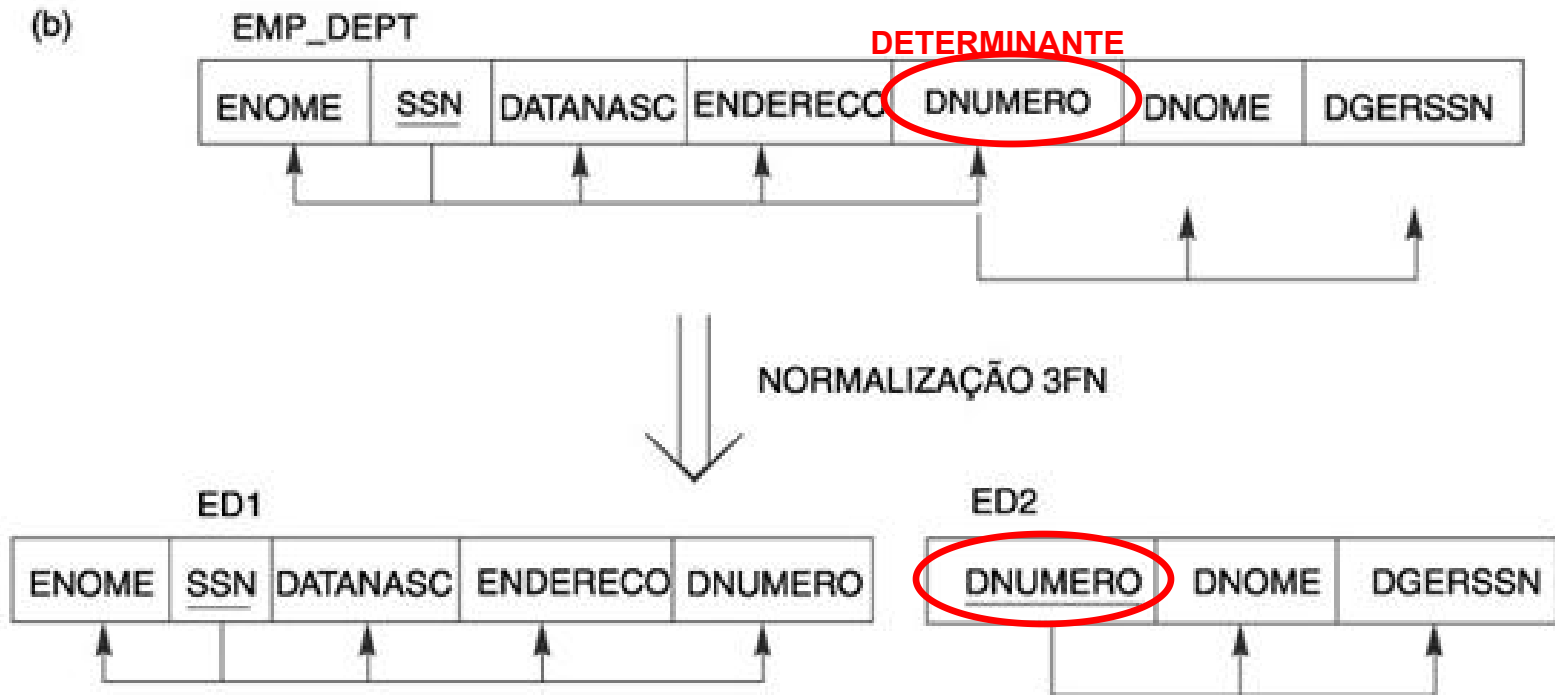
- DNOME e DGERSSN dependem funcionalmente de DNUMERO ($Z_4 \rightarrow \{Z_5, Z_6\}$)
- DNUMERO depende funcionalmente de SSN ($X \rightarrow \{Z_1, \dots, Z_4\}$)
 - DNUMERO não é chave, nem parte de chave
- DNOME e DGERSSN dependem transitivamente de SSN

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:



Terceira Forma Normal (3FN)

- Exemplo 2:
 - 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)

- 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)

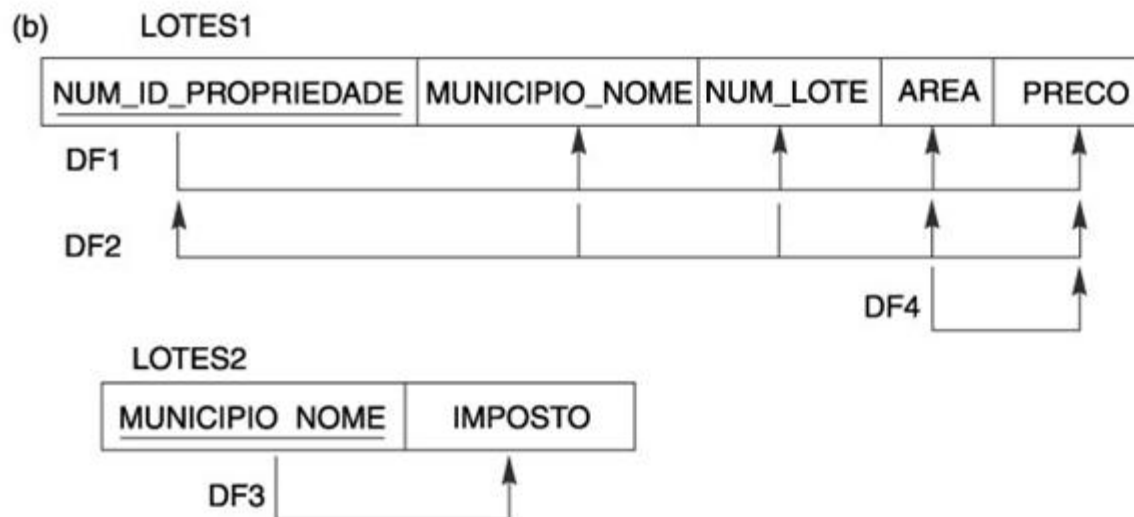
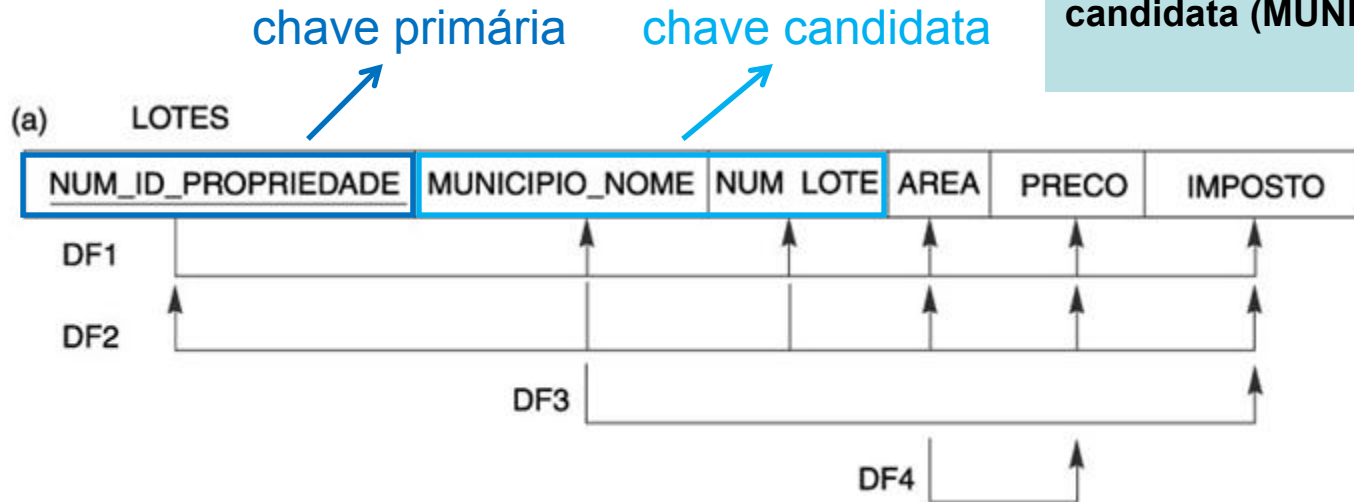
Definição Genérica

- 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 (chave primária ou chaves candidatas)

Definição Genérica

Segunda forma normal

Não está na 2FN: imposto depende de parte da chave candidata (MUNICIPIO_NOME)

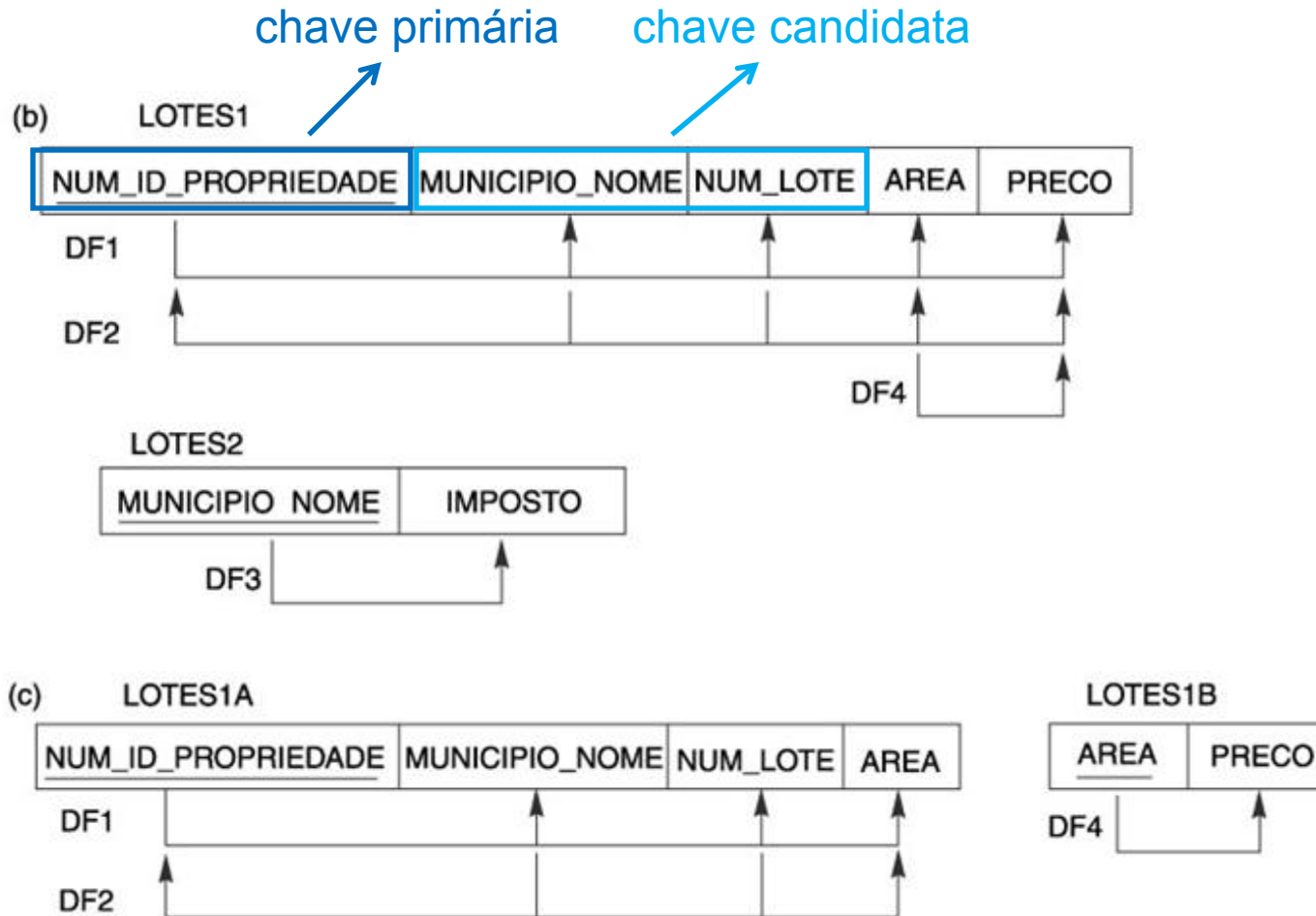


Definição Genérica

- 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

Definição Genérica

Terceira forma normal



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

Forma Normal de Boyce-Codd

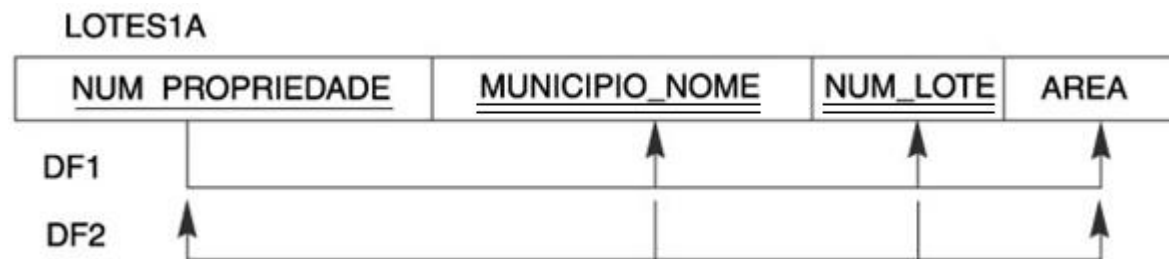
- R está na forma normal de Boyce-Codd (FNBC) se
 - Para toda DF $X \rightarrow A$ em F , X é uma superchave, isto é $X \rightarrow R$.

Forma Normal de Boyce-Codd

- Em uma relação na FNBC
 - As únicas dependências são aquelas em que uma chave determina algum(ns) atributo(s)
 - É garantido que nenhuma redundância pode ser detectada usando-se apenas as informações das DFs
 - É a forma normal mais desejável do ponto de vista de redundância

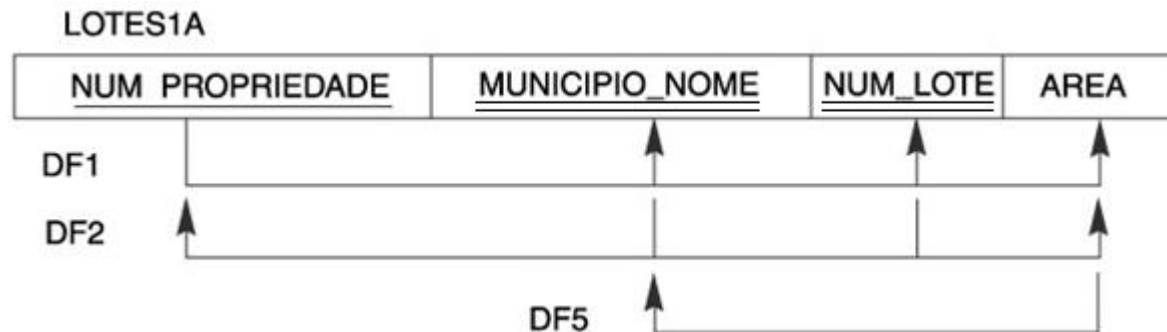
Exemplo

- Considere o esquema LOTES, que descreve lotes à venda em vários municípios
- Considere as chaves:
 - NUM_PROPRIEDADE
 - {MUNICIPIO_NOME, NUM_LOTE}



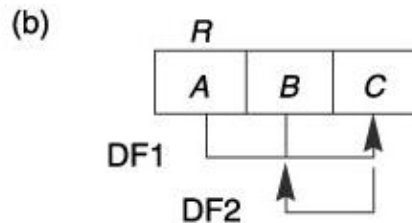
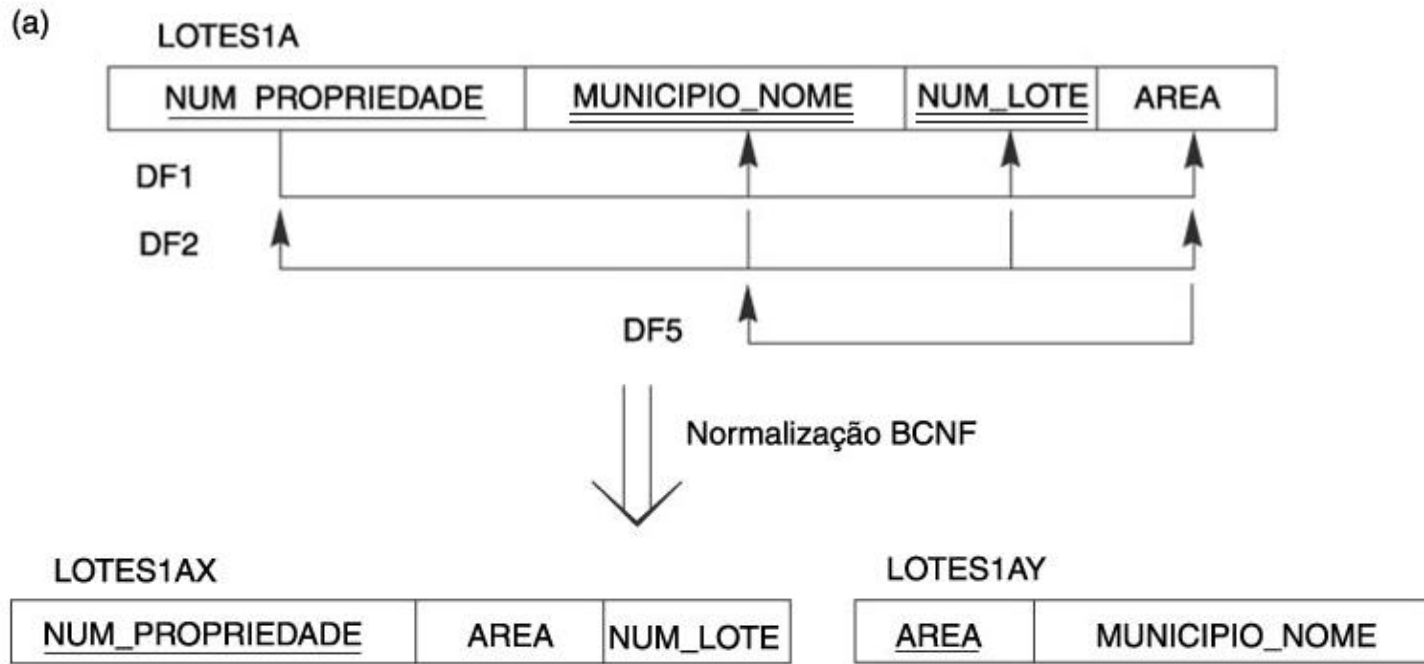
Exemplo

- Consideremos que há milhares de lotes, mas apenas nos municípios de Uberlândia e Araguari
 - em Uberlândia só existem lotes com 100, 200 e 300 m²
 - em Araguari só existem lotes com 150, 250 e 350 m²
- Nesse caso
 - DF5: AREA → MUNICIPIO_NOME



Forma Normal de Boyce-Codd

Exemplo



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

Forma Normal de Boyce-Codd

X	Y	A
---	---	---

x	y1	a
x	y2	?

- Suponha a DF $X \rightarrow A$
 - O que podemos inferir sobre a coluna A na 2ª tupla?
R: usando a DF, conclui-se que a 2ª tupla também tem o valor a na coluna A
 - Uma situação assim pode surgir em uma relação na FNBC?
R: Não! Se uma relação está na FNBC porque A é diferente de X, segue-se que X deve ser uma chave
Se X é uma chave, então $y1 = y2$, o que significa que as duas tuplas são idênticas

Forma Normal

- Cada FN engloba a FN anterior:
 - Toda relação em 2FN está na 1FN
 - Toda relação em 3FN está na 2FN
 - Toda relação em BCNF está na 3FN
- Existem relações que estão na 3FN mas não em BCNF
- A meta é alcançar a BCNF (ou 3FN) em todas as relações

Referências

- R. Ramakrishnan e J. Gehrke, *Database Management Systems*, 3a Edição, McGraw-Hill, 2003.