



Dependências Funcionais e Normalização

Prof. Humberto Luiz Razente
Bloco B - sala 1B144

Roteiro

- ◆ Dependências Funcionais
- ◆ Introdução aos conceitos de normalização
 - 1 FN
 - 2 FN
 - 3 FN
 - BCNF

Dependência Funcional e Normalização

- ◆ Cada esquema de relação consiste de um número de atributos e o esquema do banco de dados relacional consiste de um número de esquemas de relação
- ◆ É necessário uma maneira formal para mensurar o quanto um esquema de relação é melhor do que outro
 - Dependências funcionais
 - Formas normais

Dependência Funcional e Normalização

◆ Processo de normalização

- oferece mecanismos para analisar o projeto do BD
 - ◆ identificação de erros
- oferece métodos para corrigir problemas

◆ Erros encontrados

- repetição de informação
- perda de informações
- inability de representar certas informações

Dependência Funcional

- ◆ É uma **restrição** entre dois conjuntos de atributos de um esquema de relação R
 - restrição \rightarrow impõe uma limitação nos valores possíveis de tuplas que formem um estado da relação r de R
- ◆ É uma propriedade da **semântica** ou do **significado dos atributos**
 - são derivadas pelo projetista do BD na análise da especificação de requisitos

Dependência Funcional

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

atributo $A \rightarrow$ atributo B

determina
funcionalmente

1 valor de A
(não existe outro)

somente um
valor do atributo B

lado à esquerda

lado à direita

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
- ◆ Notação: $A \rightarrow B$
- ◆ Se B é funcionalmente dependente de A, então A determina funcionalmente B

Observações

- ◆ As dependências funcionais são informações semânticas fornecidas pelo projetista
- ◆ 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

Dependência Funcional

- ◆ Certas DF 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:
 - ◆ $\{\text{ESTADO, OAB}\} \rightarrow \text{NOME_ADVOGADO}$
 - ◆ $\{\text{ESTADO, CREA}\} \rightarrow \text{NOME_ENGENHEIRO}$

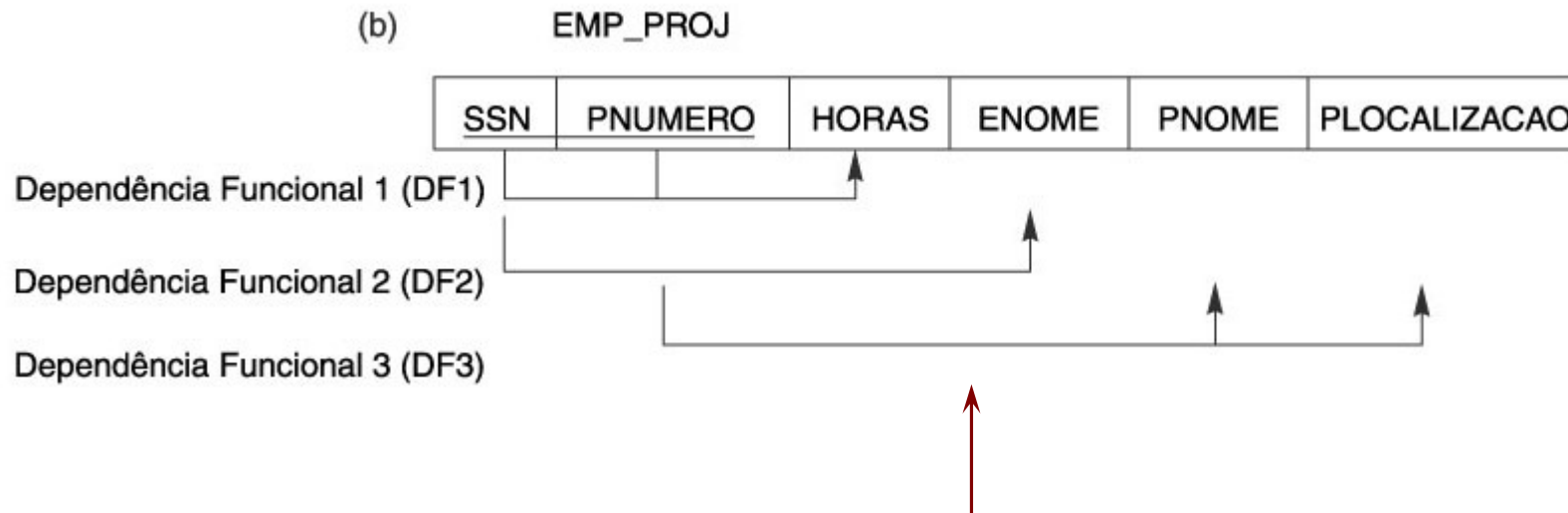
Dependência Funcional

- ◆ Também é possível que algumas DF possam deixar de existir
- ◆ Exemplo: telefonia celular nos anos 90 e início dos anos 2000
 - ◆ PRIMEIROS_QUATRO_DÍGITOS → OPERADORA_CELULAR
 - ◆ com a portabilidade numérica a partir de 2008 no Brasil, essa DF passou a não ser mais verdadeira

Dependência Funcional

◆ Exemplo:

- ◆ $SSN \rightarrow ENOME$
- ◆ $PNUMERO \rightarrow \{PNOME, PLOCALIZACAO\}$
- ◆ $\{SSN, PNUMERO\} \rightarrow HORAS$





Notação diagramática para DF

Exercício 1

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

As seguintes dependências são corretas?

- ◆ $\text{nro_cliente} \rightarrow \text{nome}$  OK!
- ◆ $\text{nro_cliente} \rightarrow \text{endereço}$  OK!
- ~~◆ $\text{nome} \rightarrow \text{endereço}$~~
- ~~◆ $\text{endereço} \rightarrow \text{nome}$~~ Não!

Exercício 2

- ◆ 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 2

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

◆ As seguintes dependências são verdadeiras?

- ~~nro_pedido → qtidade_comprada~~
- ~~nro_peça → qtidade_comprada~~
- ~~nro_pedido → preço_cotado~~
- ~~nro_peça → preço_cotado~~

Não!

OK!

- {nro_pedido, nro_peça} → qtidade_comprada
- {nro_pedido, nro_peça} → preço_cotado
- {nro_pedido, nro_peça} → {qtidade_comprada, preço_cotado}

Dependência Funcional e Normalização

◆ Vantagens:

- garante relações sem redundância
- oferece fácil recuperação das informações

◆ Formas normais:

- Primeira Forma Normal (1FN)
- Segunda Forma Normal (2FN)
- Terceira Forma Normal (3FN)
- Forma Normal de Boyce-Codd (FNBC)
- 4FN, 5FN...

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, Datnasc}
 - ◆ {SSN, Enome, Datnasc, Endereço}
 - ◆ {SSN, Enome, Datnasc, 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

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

◆ 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 R está na 1FN se:

- todo valor em R for atômico e monovalorado
- ou seja, R não contém grupos de repetição

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

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>

E se Cecília tivesse 3 endereços de entrega?

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)

- ◆ 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

Segunda Forma Normal (2FN)

◆ Exemplo:

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

nro-pedido → data

nro-peça → descrição

{nro-pedido, nro-peça} → {quantidade_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)

Exercícios

- ◆ Diga em que forma normal (Nenhuma, 1 FN, 2 FN) está cada relação abaixo, justificando sua resposta. Depois, se necessário, indique os passos que devem ser realizados para normalizar para a 2ª forma normal.

- ◆ LIVROS = {Título, Autor, Tipo, Preço, {FiliaçãoDoAutor}, Editora}
 - Título → {Editora, Tipo}
 - Tipo → Preço
 - Autor → FiliaçãoDoAutor

- ◆ FORNECEDOR = {CNPJ, RazãoSocial, NomeFantasia, Contato}
 - CNPJ → {RazãoSocial, NomeFantasia, Contato}

- ◆ CLIENTE = {CPF, Nome, NroAgência, NroConta, TipoConta}
 - CPF → {Nome, NroAgência, NroConta, TipoConta}
 - {NroAgência, NroConta} → {CPF, Nome, TipoConta}
 - TipoConta → NroAgência

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)

Uma relação está na **3ª FN** se já estiver na 2ª e ...

TODOS os atributos que **NÃO** fazem parte da chave primária **NÃO** possuírem nenhuma dependência entre si.

OU SEJA,

Na 2ª FN verifica-se a dependência em relação aos atributos que fazem parte da CHAVE PRIMARIA

Na 3ª FN verifica-se a dependência em relação aos atributos que **NÃO** fazem parte da CHAVE PRIMARIA

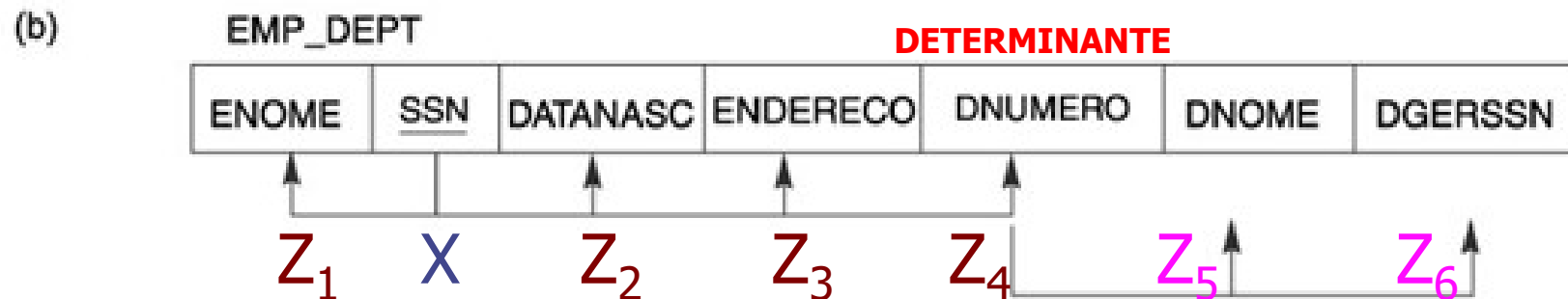
Terceira Forma Normal (3FN)

◆ 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)

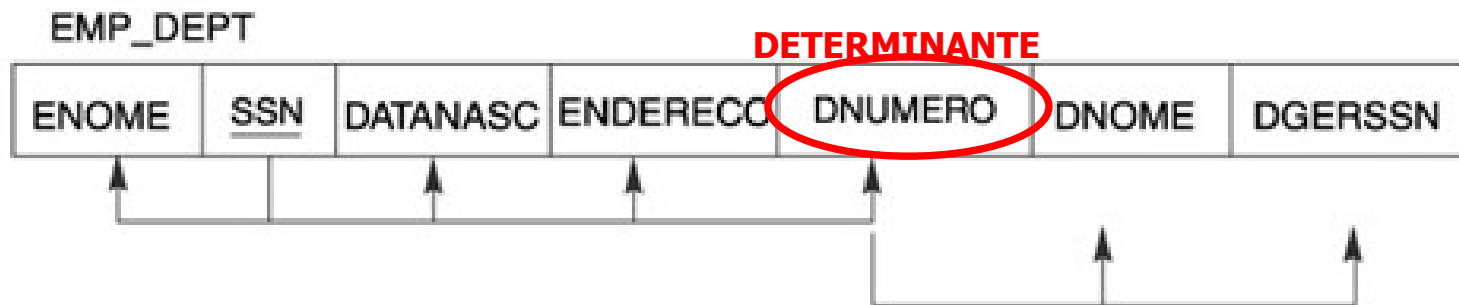
◆ Método para corrigir o problema:

- 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:

(b)



NORMALIZAÇÃO 3FN



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)

Exercícios

- ◆ Diga em que forma normal (Nenhuma, 1 FN, 2 FN ou 3 FN) está cada relação abaixo, justificando sua resposta. Depois, se necessário, indique os passos que devem ser realizados para normalizar para a forma normal mais restrita possível.
- ◆ LIVROS = {Título, Autor, Tipo, Preço, {FiliaçãoDoAutor}, Editora}
 - Título → {Editora, Tipo}
 - Tipo → Preço
 - Autor → FiliaçãoDoAutor
- ◆ FORNECEDOR = {CNPJ, RazãoSocial, NomeFantasia, Contato}
 - CNPJ → {RazãoSocial, NomeFantasia, Contato}
- ◆ CLIENTE = {CPF, Nome, NroAgência, NroConta, TipoConta}
 - CPF → {Nome, NroAgência, NroConta, TipoConta}
 - {NroAgência, NroConta} → {CPF, Nome, TipoConta}
 - TipoConta → NroAgência

Exercícios

◆ LIVROS = {Título, Autor, Tipo, Preço, {FiliaçãoDoAutor}, Editora}

■ DF:

- ◆ Título → {Editora, Tipo}
- ◆ Tipo → Preço
- ◆ Autor → FiliaçãoDoAutor

1) Não está na 1FN (atributo multivalorado)

LIVROS = {Título, Autor, Tipo, Preço, Pai, Mãe, Editora}

2) Não está na 2FN (existência de dependência parcial)

- LIVROS = {Título, Autor}
- DADOSLIVROS = {Título, Tipo, Preço, Editora}
- AUTOR = {Autor, Pai, Mãe}

3) Não está na 3FN (existência de dependência transitiva)

- LIVROS = {Título, Autor}
- DADOSLIVROS = {Título, Tipo, Editora}
- TIPO_LIVRO = {Tipo, Preço}
- AUTOR = {Autor, Pai, Mãe}

Exercícios

- ◆ FORNECEDOR = {CNPJ, RazãoSocial, NomeFantasia, Contato}
 - DF: CNPJ \rightarrow {RazãoSocial, NomeFantasia, Contato}
 - 1FN, ok! Todos atributos atômicos
 - 2FN, ok! A chave primária é composta de apenas 1 atributo
 - 3FN, ok! Não existe dependência transitiva

Exercícios

◆ CLIENTE = {CPF, Nome, NroAgência, NroConta, TipoConta}

- $CPF \rightarrow \{Nome, NroAgência, NroConta, TipoConta\}$
- $\{NroAgência, NroConta\} \rightarrow \{CPF, Nome, TipoConta\}$
- $TipoConta \rightarrow NroAgência$

- 1FN, ok! Todos atributos atômicos
- 2FN, ok! Não existe dependência parcial
- 3FN, ok! Não existe dependência transitiva

Considerações Finais

- ◆ A normalização para as FN apoiadas em DF se atinge com a separação dos atributos em duas ou mais relações
 - Isso aumenta o número de relações
 - Requer operações de junção na recuperação de informações
- ◆ Normalizar evita inconsistências nas relações, porém obriga a execução de operações de junção nas consultas

Considerações Finais

◆ Normalizar ou não uma relação?

- O que é mais importante
 - ◆ garantir a eliminação de inconsistências no banco de dados ou a eficiência de acesso?
- Se a consistência não for um fator fundamental pode-se abrir mão da normalização
 - ◆ em casos muito especiais
 - por exemplo em relações com uma quantidade crítica de linhas

Exercícios

1. Diga em que forma normal (Nenhuma, 1FN, 2FN, 3FN) está cada relação abaixo, justificando sua resposta. Depois, se necessário, indique os passos que devem ser realizados para normalizar para a forma normal mais restrita possível.

a) CARROSVENDIDOS = {Carro, DataVenda, Vendedor, Comissão, Desconto}

■ DF:

- ♦ Carro → DataVenda
- ♦ DataVenda → Desconto
- ♦ Vendedor → Comissão

b) FILIAL = {CodF, País, Cidade, Continente, Língua, NomeGerente, FusoHorário, Nível}

■ DF:

- ♦ CodF → {País, Cidade, NomeGerente, Nível, FusoHorário}
- ♦ País → {Continente, Língua}

Exercícios

c. ProjetoEmpresa = {CodProj, Tipo, Descr, CodEmp, Nome, Cat, Sal, DataIni, TempAl}

■ DF:

- ♦ { CodProj, CodEmp } → {DataIni, TempAl}
- ♦ CodProj → {Tipo, Descr}
- ♦ CodEmp → {Nome, Cat, Sal}
- ♦ Cat → Sal

d. Matricula = {CodAluno, CodTurma, CodDisciplina, CargaHoraria, Oferecimento, NomeDisciplina, NomeAluno, CodLocalNascAluno, NomeLocalNascAluno, Nota, Faltas}

■ DF:

- ♦ CodAluno → {NomeAluno, CodLocalNascAluno, NomeLocalNascAluno}
- ♦ CodLocalNascAluno → NomeLocalNascAluno
- ♦ CodTurma → {CodDisciplina, NomeDisciplina, CargaHoraria, Oferecimento}
- ♦ CodDisciplina → {NomeDisciplina, CargaHoraria}
- ♦ {CodAluno, CodTurma} → {Nota, Faltas}

Bibliografia e leitura complementar para casa

- ◆ Elmasri, Ramez; Navathe, Shamkant B. Sistemas de banco de dados, 6ª edição.
 - Capítulo 15: “Fundamentos de dependências funcionais e normalização para bancos de dados relacionais”