



Dependência Funcional e Normalização

Prof. Bruno Travençolo



Roteiro

- ◆ Dependência Funcional
- ◆ Introdução aos conceitos de normalização
 - 1 FN
 - 2 FN
 - 3 FN

Dependência Funcional

relação R (atributoA, atributoB, atributoC)

atributoA \rightarrow atributoB

determina
funcionalmente

1 valor de A
(não existe outro)

somente um
valor do atributo B

lado à esquerda

lado à direita

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

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

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.

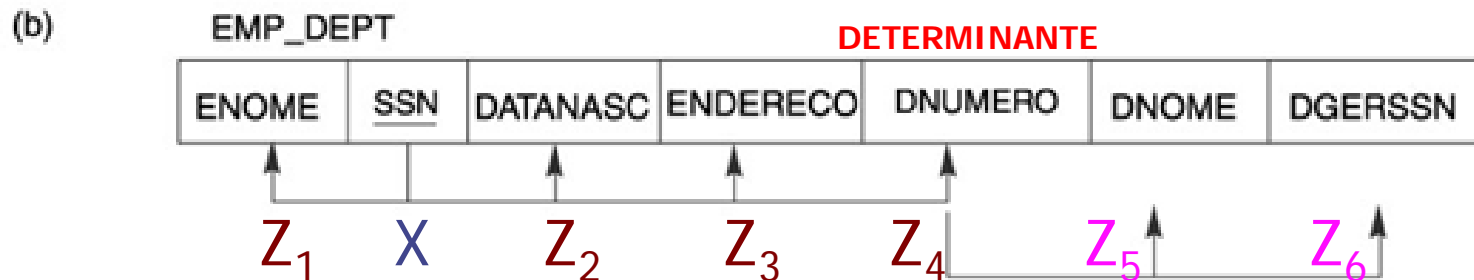
OU SEJA,

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

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

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

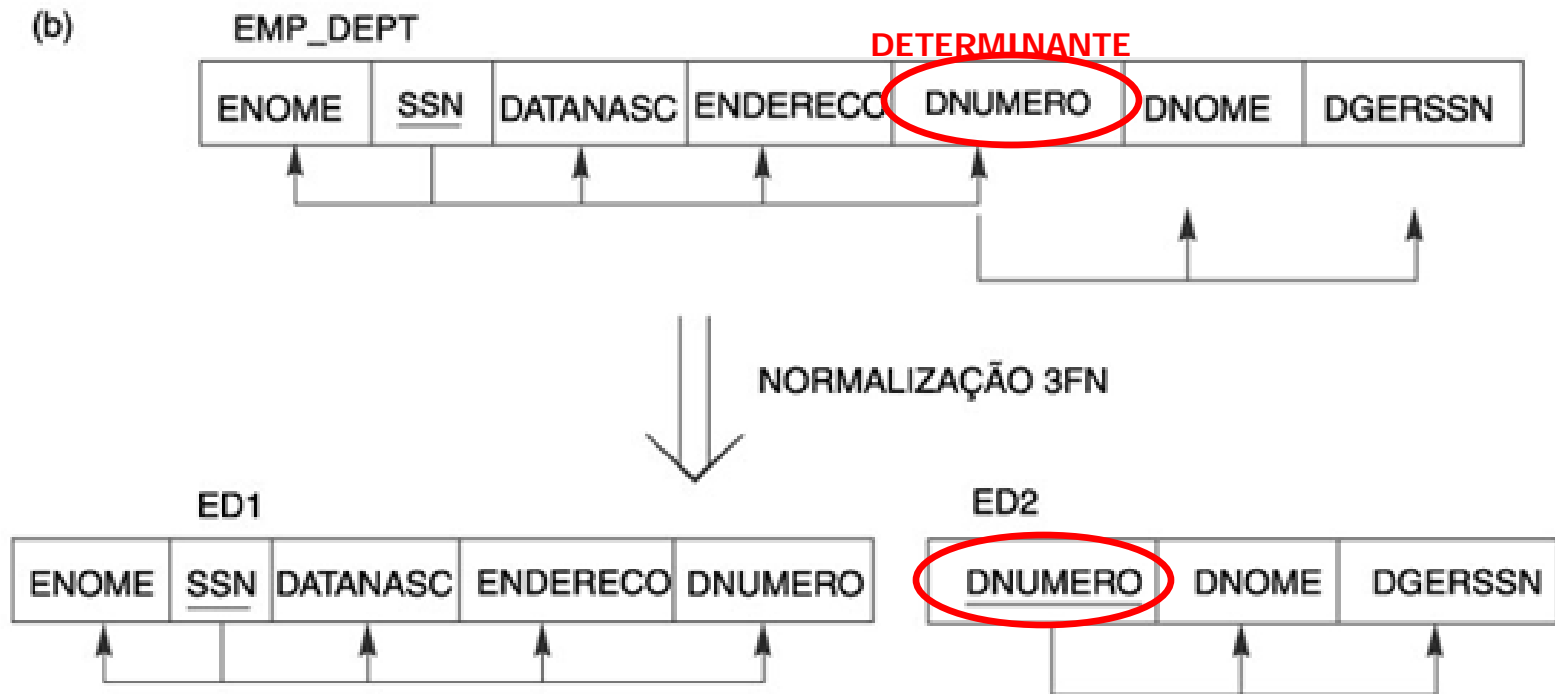
Terceira Forma Normal (3FN)

◆ 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)

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

- ◆ Slides: Prof. Humberto Luiz Razente
- ◆ 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”

