

SISTEMAS DE BANCO DE DADOS 1

AULA 8

Normalização

Vandor Roberto Vilardi Rissoli



APRESENTAÇÃO

- Dependência Funcional (revisando)
- Normalização
 - Primeira
 - Segunda
 - Terceira
 - *Boyce Codd*
- Referências



Dependência Funcional

As dependências funcionais exploradas em estudo anterior contribuem com a definição de algumas das **Formas Normais**, a fim de garantir um nível de consistência na modelagem proposta como solução de um problema.

a) Representação da Dependência Funcional

$A \rightarrow B$ (indica que A **determina funcionalmente** B ou também que B **depende de** A).

b) Representação da Dependência Multivalorada

$A \twoheadrightarrow B$ (indica que A **determina muitos** B's).



Dependência Funcional

A Normalização também permite ao projetista controlar o quanto de CONSISTÊNCIA SERÁ GARANTIDA pela maneira de construção do sistema (estrutura), e quanto deve ser de responsabilidade dos aplicativos e/ou do SGBD.

Porém, se deve analisar até que ponto a Normalização deve chegar, pois

Normalizar DEMAIS pode diminuir a eficiência dos aplicativos que utilizam a base de dados (BD);

Normalizar de MENOS facilita as possibilidades das inconsistências na BD.



Dependência Funcional

O processo de Normalização consiste em analisar cada uma das relações (tabelas) que compõe o projeto de banco de dados elaborado e verificar a qual forma normal cada tabela atende.

De acordo com o número mais alto que todas as tabelas do projeto atendam, então será classificado que o projeto de banco de dados proposto atende a esta determinada **Forma Normal**.

Geralmente, o processo de normalizar um projeto irá provocar a decomposição de esquemas insatisfatórios de algumas relações em novas relações que tenham cuidados com as situações já estudadas para a elaboração e implementação de um projeto mais eficiente e que evite as **ANOMALIAS**.



Normalização

As Formas Normais, processo chamado de Normalização, são identificadas por meio de números, que serão estudadas a primeira, segunda e terceira forma normal no decorrer da disciplina.

No Modelo Relacional a normalização principal é a primeira, pois é considerada parte da própria definição de uma relação.

PRIMEIRA - (1 FN)

Uma relação se encontra na Primeira Forma Normal se todos os domínios de atributos possuem apenas valores atômicos (simples e indivisíveis). Com isso a relação é construída sem atributos compostos e multivalorados em suas tuplas.

Exemplo:

CURSO(curso, periodo, idCurso, (nome, matricula)) – não 1FN

Normalização

Para levar uma relação a atender as exigências da primeira forma normal (normalizar) tem-se que realizar a aplicação de algumas “**regras**” para as seguintes situações:

- 1- Cada um dos atributos compostos (ou não atômicos) devem ser “divididos” em seus componentes
- 2- Para os atributos multivalorados tem-se duas alternativas:

A) Pequena quantidade de valores possíveis, substitui-se o atributo multivalorado por um conjunto de atributos de mesmo domínio, cada um mono valorado representando uma ocorrência do valor;

B) Quantidade de possíveis valores desconhecida, grande ou muito variada, retira-se da relação o atributo multivalorado e cria-se uma nova relação que tem o mesmo conjunto de atributos chave, mais o atributo multivalorado **também como chave**, mas tomado como mono valorado.

Normalização

A 1 FN é uma das maneiras de controlar a consistência por meio da própria estrutura do sistema, sendo também fundamental para a conceituação do sistema.

SEGUNDA - (2 FN)

Uma relação esta na Segunda Forma Normal quando ela está na 1 FN e todos os atributos que não participam da chave primária são dependentes diretos de toda a chave primária.

Para se normalizar uma relação para a 2 FN as “**regras**” seguintes devem ser seguidas.



Normalização

A normalização para um relação estar na 2 FN deve:

- 1- Verificar os grupos de atributos que dependem da mesma parte da chave primária;
- 2- Retirar da relação todos os atributos de um desses grupos;
- 3- Criar uma nova relação contendo esse grupo como atributo não chave, e os atributos que determinam esse grupo como chave primária;
- 4- Repetir o procedimento para cada grupo, até que a relação toda somente contenha atributos que dependam da chave primária.



Normalização

Esta normalização evita a inconsistência devido a duplicidade de informações, além da perda de dados em operações de remoção ou de alterações na relação (**anomalias**).

Note que a normalização de relações é realizada, na grande maioria das vezes, decompondo-se uma relação em duas ou mais relações.

Exemplo: não está na 2 FN

EMPR_PROJ (matrEmpr, codProj, horas, nomeEmpr,
nomeProj, localProj)



Normalização

continuação do exemplo...

EMPR_PROJ (matrEmpr, codProj, horas, nomeEmpr,
nomeProj, localProj)

- Dependências funcionais parciais em relação à chave primária:
matrEmpr \rightarrow nomeEmpr
codProj \rightarrow nomeProj, localProj
- Dependências funcionais totais em relação à chave primária:
(matrEmpr, codProj) \rightarrow horas

Esquema que atende a 2 FN:

EMPREGADO (matrEmpr, nomeEmpr)

PROJETO(codProj, nomeProj, localProj)

EMPR_PROJ(matrEmpr,codProj, horas)

Normalização

TERCEIRA - (3 FN)

Uma tabela está na Terceira Forma Normal se estiver na 2 FN e não possuir dependências transitivas (ou indireta). Uma dependência transitiva ocorre quando

$$A \rightarrow B \text{ e } B \rightarrow C, \text{ então } A \rightarrow C.$$

Em outras palavras, deve-se evitar que qualquer atributo não chave seja dependente funcional de outro atributo não chave.

Exemplo: não está na 3 FN

EMPR_DEPTO (matrEmpr, nomeEmpr, dataNasc,
codDep, nomeDep)



Normalização

continuação do exemplo...

EMPR_DEPTO (matrEmpr, nomeEmpr, dataNasc,
codDep, nomeDep)

Dependências transitivas (indiretas):

matrEmpr \rightarrow codDep \rightarrow nomeDep

Esquema que atende a 3 FN:

EMPR (matrEmpr, nomeEmpr, dataNasc, codDep)

DEPTO (codDep, nomeDep)

Restrição **EMPR_DEPTO_FK** de chave estrangeira (codDep)
que Referencia DEPTO(codDep)

→ Da mesma forma que a 2 FN, a normalização para a 3 FN evita a inconsistência devido a duplicidade de dados, além da perda de dados em operações de remoção ou de alterações na relação.

Normalização

Um outro exemplo no processo de normalização envolvendo, diretamente, a dependência transitiva seria:

ENFERMEIRO (codigo, nome, dtFormacao,
dtNascimento, sexo, idade)

Exemplo: não está na 3 FN

ENFERMEIRO (codigo, nome, dtFormacao,
dtNascimento, sexo, ~~idade~~)



O atributo **idade** tem dependência direta em relação ao atributo **dtNascimento** e poderia ser responsável pela inconsistência na base de dados se um enfermeiro tivesse nascido a exatos 20 anos atrás, mas a idade armazenada fosse 51.

Qual idade em anos completos seria a correta para o enfermeiro?

Normalização

Assim, em um projeto de banco de dados que esteja na **3FN** (terceira forma normal) não deverá existir a descrição do atributo derivado no ME-R, mas ele estará no DE-R com a expressão explícita (**derivado**) como comentário no respectivo atributo, pois tal dado é do interesse do usuário. Por exemplo :

ME-R

ENFERMEIRO (codigo, nome, dtFormacao, dtNascimento, sexo)

DE-R



→ No **Dicionário de Dados** este atributo, geralmente desejado pelo usuário, também deverá estar completamente documentado (descrito).

Normalização

RELAÇÃO (ou Tabela)
NÃO NORMALIZADA

Normalizar Projeto Lógico
Banco de Dados

Remoção dos Atributos
Multivalorados e Compostos

1ª FN

Remoção das Dependências PARCIAIS
(atributos dependentes da PK)

2ª FN

Remoção das Dependências TRANSITIVAS
(atributos dependentes da PK)

3ª FN



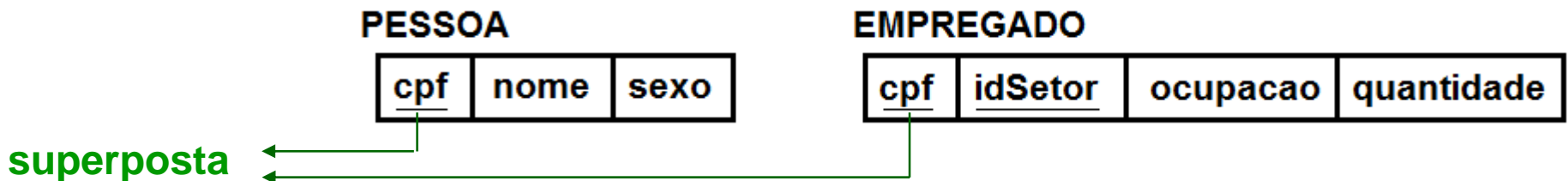
Normalização

BOYCE CODD - (FNBC)

A Forma Normal de Boyce Codd (FNBC) consiste na **3FN** mais “forte” ou rigorosa, a fim de evitar algumas **anomalias** na base de dados.

Na normalização a FNBC deve ser aplicada sobre as tabelas que já estão respeitando a **3FN** e possuam mais de uma chave candidata (**primária também é candidata antes de ser PK**), sendo pelo menos uma dessas chaves composta e com **superposição**.

CHAVE SUPERPOSTA: duas chaves são superpostas quando pelo menos uma delas é composta e existe ao menos um atributo em comum entre elas.



Normalização

Assim, para uma tabela estar na FNBC ela deverá estar na 3 FN e sem possuir nenhum outro atributo que seja determinado por outro(s) atributo(s) que NÃO é (sejam) uma chave candidata da tabela.

Se isso acontecer a solução será conduzir a tabela para a 3 FN e levar esses atributos para uma outra tabela, aplicando o conceito de **decomposição sem perdas**, por exemplo:

CLIENTE	AGENCIA	GERENTE
00185	Norte	Paulo
01018	Norte	Ana
05270	Norte	Paulo
04857	Norte	Ana
00854	Central	Maria
05879	Central	Carlos

- A tabela está na 3 FN;
- Note que existe outra chave candidata e que ela pode ser sobreposta à chave primária;
- **CLIENTE** estará nas duas chaves candidatas compostas, efetuando a sobreposição (**sem FNBC**).



<u>CLIENTE</u>	<u>AGENCIA</u>
----------------	----------------

<u>CLIENTE</u>	<u>GERENTE</u>
----------------	----------------

→ **Anomalia na atualização, pois a agência se repete para o mesmo gerente**

Normalização

Chave primária { ~~CLIENTE~~, AGENCIA } → { GERENTE }

Chave candidata { ~~CLIENTE~~, ~~GERENTE~~ } → { AGENCIA }

Dessa forma, será necessário decompor a tabela para que ela esteja na FNBC, no qual o atributo determinado por um campo que **NÃO** é chave candidata vai para uma outra tabela.

- Note que há outro **determinante** que **NÃO** é chave candidata, sendo superposta e fazendo com que esta solução **respeite a FNBC**.

GERENTE	AGENCIA
Paulo	Norte
Ana	Norte
Maria	Central
Carlos	Central

CLIENTE	GERENTE
00185	Paulo
01018	Ana
05270	Paulo
04857	Ana
00854	Maria
05879	Carlos

CLIENTE	AGENCIA	GERENTE
00185	Norte	Paulo
01018	Norte	Ana
05270	Norte	Paulo
04857	Norte	Ana
00854	Central	Maria
05879	Central	Carlos

- Tabelas na **3FN** que **NÃO** possuem superposição de chaves já estão ou respeitam a **FNBC**.

Normalização

Realizando a Normalização

- A normalização é aplicada em uma relação por vez;
- Uma relação vai sendo dividida criando outras relações, quando necessário;
- Inicia-se das formas normais menos rígidas (1 FN→2 FN→...) até chegar em uma normalização considerada satisfatória.

A **decisão de normalizar** ou NÃO uma relação é um compromisso entre garantir a eliminação de inconsistências no BD e a eficiência de acesso. A normalização para formas apoiadas em dependências funcionais evita inconsistências, usando para isso a própria construção do banco de dados.

→ Se a consistência NÃO for um fator fundamental para um projeto, pode-se abrir mão da normalização.

Exercício de Fixação

- 1) A partir do estudo deste material sobre Normalização, todos o próximos exercícios que solicitam um projeto de banco de dados deverão atender no mínimo a Terceira Forma Normal (**3FN**). Assim, o **Exercício 2 da Aula 7** (Farmácia **AI-AI**) deverá respeitar também a exigência descrita a seguir em sua novas evoluções solicitadas pela disciplina/turma.

Baseado no enunciado definido na aula anterior, elabore o projeto de banco de dados a partir do ME-R até a implementação física de possível solução à **Farmácia AI-AI**, a fim dos esquemas de todas as tabelas estarem na **Terceira Forma Normal (3FN)** para a proposta entregue poder ser considerada uma possível solução ao problema, caso contrário nem será considerada uma proposta por não estar na **3FN**.

Referência de Criação e Apoio ao Estudo

Material para Consulta e Apoio ao Conteúdo

- ELMASRI, R. e NAVATHE, S. B., Fundamentals of Database Systems, Addison-Wesley, 3rd edition, 2000
 - Capítulo 14
- SILBERSCHATZ, A. & KORTH, H. F., Sistemas de Banco de Dados.
 - Capítulo 7
- HEUSER, C. A., Projeto de Banco Dados, 2001.
 - Capítulo 6
- Universidade de Brasília (UnB Gama)
 - <https://sae.unb.br/cae/conteudo/unbfga/>
(escolha a opção **Sistemas Banco Dados 1** seguida da opção **Mod.Entid.Relacionamento**)