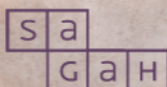


# MODELAGEM E DESENVOLVIMENTO DE BANCO DE DADOS

Fabrício Felipe Meleto Barboza



SOLUÇÕES  
EDUCACIONAIS  
INTEGRADAS



# Normalização de dados

## Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Identificar as dependências funcionais.
- Reconhecer as formas normais (1FN, 2FN, 3FN, e FN Boyce-Codd).
- Relacionar os processos de normalização.

## Introdução

A normalização de dados permite um melhor relacionamento e performance de dados dentro do próprio Sistema de Gerenciamento de Banco de Dados (SGBD).

Neste capítulo, você vai estudar como são feitas as tabelas nas formas normais, garantindo que estejam aderentes às regras, e vai ver como fazer a normalização de uma tabela, desde o início, passando pelas três formas normais (1FN, 2FN e 3FN) e identificando a mudança que ocorre em cada uma delas.

## Dependências funcionais

Você está administrando um banco de dados grandioso, com várias tabelas e *databases*. Ao inserir, excluir ou editar alguma informação do cadastro dos clientes, você precisaria manipular os dados da tabela “clientes”, que concentra essas informações. Bem rápido, fácil e estruturado, correto? Agora, pense que esse sistema utiliza a tabela “clientes” para montar o relatório gerencial dos cadastros, utiliza a tabela “clientes\_vendas” para o cadastro dos mesmos dados – mas para referenciar uma venda – e, ainda, a tabela “clientes\_aberto” para esses atributos serem exibidos na tela de clientes com contas em aberto.

Dessa forma, para editar o campo telefone de um cliente, você precisaria manipular as tabelas “clientes”, “clientes\_vendas” e “clientes\_aberto”. O trabalho foi triplicado – e isso para qualquer edição, inserção ou deleção! Em vez de a informação ser centralizada e única, ela fica em duplicidade e correndo grande risco de não estar igual nas tabelas. Assim, a redundância de dados, salvo em caso controlado pelo próprio SGBD, como em uma replicação, por exemplo, não deve ser feita.



### Saiba mais

Segundo Macêdo (2012, documento on-line),

a redundância consiste no armazenamento de uma mesma informação em locais diferentes, provocando inconsistências. Em um banco de dados, as informações só se encontram armazenadas em um único local, não existindo duplicação descontrolada dos dados. Quando existem replicações dos dados, elas são decorrentes do processo de armazenagem típica do ambiente cliente-servidor, totalmente sob controle do banco de dados.

Outra técnica, utilizada em menor escala, para quando a replicação de dados é necessária, seria a utilização de valores nulos (NULL) em campos não importantes para aquela cópia de dados. Quando é possível que o atributo receba esse valor, isso significa que esse atributo não possui nenhum dado atrelado a ele. Também é possível dizer que esse atributo, quando recebe a entrada NULL, diz que os dados estão ausentes naquele setor. Mesmo assim, recomenda-se cautela no uso de NULL para criar uma replicação dos valores.

## Identificando as dependências funcionais

As tabelas de bancos de dados devem seguir uma regra para que fiquem livres de dados incorretos ou duplicados. Para garantir essa regra, foram definidas as formas normais, que, em teoria, todas as tabelas devem obedecer. Além disso, para que as formas normais sejam respeitadas, é necessário entender as dependências funcionais.

Uma dependência funcional ocorre quando um valor A depende de um valor B, ou seja, uma coluna depende de outra coluna da mesma tabela.

As dependências funcionais se dividem em três: dependências funcionais parciais, dependências funcionais totais e dependências funcionais transitivas.

## Dependência funcional total

Quando houver uma chave primária concatenada, isto é, duas ou mais colunas são a chave primária de uma tabela, as demais colunas dependerão exclusivamente dessa ligação para que possam ser inseridas corretamente.

Exemplo:

- **Tabela ItemVenda**
  - CodVenda PRIMARY KEY
  - CodProduto PRIMARY KEY
  - Qtd

Repare que a coluna Qtd irá depender totalmente da chave primária concatenada entre CodVenda e CodProduto.

## Dependência funcional parcial

Ocorre quando um item da tabela depende de uma parte da chave primária concatenada da tabela, e não da chave toda.

Exemplo:

- **Tabela ItemVenda**
  - CodVenda PRIMARY KEY
  - CodProduto PRIMARY KEY
  - Qtd
  - PrecoProduto

A coluna PrecoProduto depende do valor da coluna CodProduto, que faz parte da chave primária da tabela ItemVenda. Portanto, PrecoProduto depende parcialmente da chave primária concatenada da tabela.

## Dependência funcional transitiva

Acontece quando uma coluna da tabela depende de outra coluna da tabela que não é chave dessa tabela.

Exemplo:

### ■ Tabela ItemVenda

- CodVenda PRIMARY KEY
- CodProduto PRIMARY KEY
- Qtd
- PrecoProduto
- TotalParcial

A coluna TotalParcial depende do resultado da multiplicação das colunas Qtd por PrecoProduto, e essas colunas não são chave da tabela.

## Reconhecendo as formas normais (1FN, 2FN, 3FN, e FN Boyce-Codd)

Para o estudo das formas normais, utilizaremos como base a planilha a seguir (Quadro 1), adaptada do exemplo de Massan (2007), a qual devemos transformar em tabelas para compor o sistema que será desenvolvido.

Quadro 1. Planilha com dados para estudo das formas normais

CodProj	Tipo	Descricao	Empregado					
			CodEmpregado	Nome	Categoria	Salario	DataInicio	TempoAlocado
PJ554	Desenvolvimento	Sistema de estoque	1	João	A1	6	01/01/2019	12
			2	Maria	A1	6	01/01/2019	12
			7	José	B2	4	01/01/2019	12
			8	Angelo	B3	2	30/03/2019	9
			9	Márcia	A2	5	15/04/2019	6
			10	Pedro	A2	5	15/04/2019	6
PJ529	Desenvolvimento	Sistema de compra	1	João	A1	6	01/01/2020	8
			2	Maria	A1	6	01/01/2020	3
			7	José	B2	4	01/01/2020	6
			8	Angelo	B3	2	01/01/2020	5
			9	Márcia	A2	5	15/11/2019	5
			10	Pedro	A2	5	15/11/2019	8
PF38	Correção	Problema no upload	3	Augusto	A1	6	01/01/2018	3
			4	Joana	B2	4	01/01/2018	1
			5	Alex	B3	2	01/01/2018	5
			6	Sophie	B3	2	01/01/2018	2
			7	José	A2	4	01/01/2020	6
			8	Angelo	B3	2	01/01/2020	5

## 1FN: primeira forma normal

A primeira forma normal comunica que a tabela estará na primeira forma normal se não tiver tabelas aninhadas uma dentro da outra. Conforme você pode reparar, podemos dividir a planilha de exemplo em mais de uma tabela, pois ela contempla exatamente um aninhamento de tabelas.

Para respeitar essa forma normal, podemos criar as tabelas **Projeto** (Quadro 2) e **AlocacaoEmpregado** (Quadro 3), conforme segue. As colunas em destaque são chave.

Tabela projeto

CodProj	Tipo	Descricao
PJ554	Desenvolvimento	Sistema de estoque
PJ529	Desenvolvimento	Sistema de compra
PF380	Correção	Problema no upload

Tabela AlocacaoEmpregado

CodProj	CodEmpregado	Nome	Categoria	Salario	DataInicio	TempoAlocado
J554	1	João	A1	6	01/01/2019	12
PJ554	2	Maria	A1	6	01/01/2019	12
PJ554	7	José	B2	4	01/01/2019	12
PJ554	8	Angelo	B3	2	30/03/2019	9
PJ554	9	Márcia	A2	5	15/04/2019	6
PJ554	10	Pedro	A2	5	15/04/2019	6
PJ529	1	João	A1	6	01/01/2020	8
PJ529	2	Maria	A1	6	01/01/2020	3
PJ529	7	José	B2	4	01/01/2020	6
PJ529	8	Angelo	B3	2	01/01/2020	5
PJ529	9	Márcia	A2	5	15/11/2019	5
PJ529	10	Pedro	A2	5	15/11/2019	8
PF380	3	Augusto	A1	6	01/01/2018	3
PF380	4	Joana	B2	4	01/01/2018	1
PF380	5	Alex	B3	2	01/01/2018	5
PF380	6	Sophie	B3	2	01/01/2018	2
PF380	7	José	A2	4	01/01/2020	6
PF380	8	Angelo	B3	2	01/01/2020	5

## 2FN: segunda forma normal

A regra de segunda forma normal diz que a tabela deve estar na primeira forma normal e não possui dependências parciais.



### Saiba mais

Segundo Siqueira (2018, documento on-line):

[...] quando uma coluna ou conjunto de colunas A depende de outra coluna B que faz parte da chave primária concatenada, dizemos que há uma dependência funcional parcial A em relação a B. Para cada valor da coluna B, existe um valor associado para a coluna A.

Assim, a tabela **AlocacaoEmpregado** deve ser segmentada em uma outra tabela, ou seja, uma terceira tabela: “empregado”. Observe nos Quadros 4 e 5.

Quadro 4. Exemplo da tabela “AlocacaoEmpregado”

CodEmpregado	Nome	Categoria	Salario
1	João	A1	6
2	Maria	A1	6
3	Augusto	A1	6
4	Joana	B2	4
5	Alex	B3	2
6	Sophie	B3	2
7	José	B2	4
8	Angelo	B3	2
9	Márcia	A2	5
10	Pedro	A2	5



**Quadro 5.** Resultado da divisão da tabela

<b>CodProj</b>	<b>CodEmpregado</b>	<b>DataInicio</b>	<b>TempoAlocado</b>
PJ554	1	01/01/2019	12
PJ554	2	01/01/2019	12
PJ554	7	01/01/2019	12
PJ554	8	30/03/2019	9
PJ554	9	15/04/2019	6
PJ554	10	15/04/2019	6
PJ529	1	01/01/2020	8
PJ529	2	01/01/2020	3
PJ529	7	01/01/2020	6
PJ529	8	01/01/2020	5
PJ529	9	15/11/2019	5
PJ529	10	15/11/2019	8
PF380	3	01/01/2018	3
PF380	4	01/01/2018	1
PF380	5	01/01/2018	5
PF380	6	01/01/2018	2
PF380	7	01/01/2020	6
PF380	8	01/01/2020	5

### 3FN: terceira forma normal

Para a terceira forma normal, a tabela precisa estar na segunda forma normal e também não pode ter dependências transitivas.

**Fique atento**

Para Siqueira (2018), “quando uma coluna ou conjunto de colunas A depende de outra coluna B, que não pertence à chave primária, dizemos que A é dependente transitivo de B”.

Repare que, na tabela *Empregado*, as colunas “Categoria” e “Salário” podem ser segmentadas em uma nova tabela. Dessa forma, temos (Quadros 6 e 7):

**Quadro 6.** Tabela *Empregado*

<b>CodEmpregado</b>	<b>Nome</b>	<b>Categoria</b>
1	João	A1
2	Maria	A1
3	Augusto	A1
4	Joana	B2
5	Alex	B3
6	Sophie	B3
7	José	B2
8	Angelo	B3
9	Márcia	A2
10	Pedro	A2

**Quadro 7.** Tabela *Categoria* após segmentação

<b>Categoria</b>	<b>Salario</b>
A1	6
A2	5
B2	4
B3	2

## Forma normal Boyce-Codd

Para a forma normal Boyce-Codd, temos que todo atributo da tabela deve depender única e exclusivamente da própria chave primária.



### Saiba mais

Carvalho (2018, documento on-line) menciona que “uma tabela está na BCNF se, e somente se, estiver na 3FN e todo atributo não chave depender funcionalmente diretamente da chave primária, ou seja, não há dependências entre atributos não chave.”

Portanto, para que a tabela esteja nessa forma normal, a dependência de qualquer coluna deve se limitar somente à chave primária da própria tabela. Veja um exemplo no Quadro 8.

**Quadro 8.** Tabela Categoria

Categoria	Salario
A1	6
A2	5
B2	4
B3	2

Repare que a coluna “Salario” depende exclusivamente da coluna “Categoria”; portanto, essa tabela está na forma normal Boyce-Codd.



## Referência

MASSAN, J. *Normalização de banco de dados*. 2007. Disponível em: <[infernatica.blogspot.com/2007/09/normalizacao-de-banco-de-dados.html](http://infernatica.blogspot.com/2007/09/normalizacao-de-banco-de-dados.html)>. Acesso em: 19 jul. 2018.

## Leituras recomendadas

CARVALHO, C. E. M. *Banco de dados: a forma normal Boyce Codd*. 06 abr. 2018. Disponível em: <<https://pt.linkedin.com/pulse/banco-de-dados-forma-normal-boyce-codd-medeiros-de-carvalho>>. Acesso em: 19 jul. 2018.

MACÊDO, D. *SGBD - Sistema de Gerenciamento de Banco de Dados*. 15 jan. 2012. Disponível em: <<https://www.diegomacedo.com.br/sgbd-sistema-de-gerenciamento-de-banco-de-dados/>>. Acesso em: 19 jul. 2018.

SIQUEIRA, F. *Normalização: parte 1*. 2018. Disponível em: <<https://sites.google.com/site/uniplibancodedados1/aulas/normalizacao>>. Acesso em: 19 jul. 2018.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

