

# Banco de Dados I

## 04 - Normalização

---

Marcos Roberto Ribeiro



Instituto Federal Minas Gerais - Campus Bambuí

2018

# Introdução

- Sempre que trabalhamos com banco de dados não desejamos que haja redundância, visto que isto pode causar armazenamento desnecessário e inconsistências;
- O armazenamento desnecessário de informações pode levar a problemas de desempenho devido ao grande número de dados armazenados em duplicidade;
- As inconsistências levam a certas **anomalias** que podem prejudicar profundamente as informações do banco de dados

# Anomalia em Bancos de Dados não Normalizados

**Anomalia de alteração** Se um dado em duplicidade é alterado, o banco de dados apresentará inconsistências a menos que todas as cópias do dados sejam atualizadas;

**Anomalia de inclusão** Não é possível incluir uma informação no banco de dados, a não ser que outra informação não relacionada também seja incluída;

**Anomalia de exclusão** Pode não ser possível apagar uma informação sem perder outra informação não relacionada.

# Reengenharia de Banco de Dados

- Quando projetamos um banco de dados utilizando um processo de modelagem adequado restarão poucas redundâncias no esquema relacional resultante, mas ainda sim podem haver redundâncias que devem ser eliminadas;
- Outro processo que gera bancos de dados com informações redundantes é a reengenharia<sup>1</sup>. No caso da reengenharia a quantidade de informação redundante pode ser extremamente grande.
- A normalização é um processo de modificação que aplica **formas normais** sobre as tabelas de um banco de dados com o intuito de eliminar a redundância de informações;

---

<sup>1</sup>Reengenharia de banco de dados é a obtenção de um novo banco de dados utilizando técnicas de engenharia reversa sobre um banco de dados existente.

# Exemplo de Banco de Dados não Normalizado

NP(número do pedido), PE (prazo de entrega), CL (cliente) , CNPJ (Código Nacional de Pessoa Jurídica), E (Endereço), CI (Cidade) , UF (Unidade Federativa) , CV (Código do Vendedor), V (Vendedor), CP (Código do Produto), DP (Descrição do Produto), UP (Unidade do Produto) , QP (Quantidade do Produto), VUP (Valor Unitário do Produto), VT (Valor Total do Produto), VTP (Valor Total do Pedido)

NP	PE	CL	CNPJ	E	CI	UF	CV	V	CP	DP	UP	QP	VUP	VT	VTP
1	10	C	003	cc3	Yyy	YY	25	W	101	ff	UN	2,0	10,0	20,0	232,0
									102	gg	UN	7,0	20,0	140,0	
									103	hh	KG	2,4	30,0	72,0	
2	15	D	004	dd4	Yyy	YY	26	Z	102	gg	UN	6,0	20,0	120,0	931,0
									103	hh	KG	4,3	30,0	129,0	
									104	ii	KG	5,8	40,0	232,0	
									105	jj	L	9,0	50,0	450,0	
3	30	A	001	aa1	Xxx	XX	25	W	101	ff	UN	8,0	10,0	80,0	80,0
4	21	A	001	aa1	Xxx	XX	25	W	102	gg	UN	10,0	20,0	200,0	280,0
									104	ii	KG	2,0	40,0	80,0	
5	10	C	003	cc3	Yyy	YY	26	Z	102	gg	UN	3,0	20,0	60,0	204,0
									104	ii	KG	2,6	40,0	94,0	
									105	jj	L	1,0	50,0	50,0	
6	30	B	002	bb2	Xxx	XX	25	W	104	ii	KG	1,8	40,0	72,0	472,0
									105	jj	L	2,0	50,0	100,0	
									106	kk	L	5,0	60,0	300,0	
7	15	A	001	aa1	Xxx	XX	25	W	102	gg	UN	3,0	20,0	60,0	135,0
									103	hh	KG	2,5	30,0	75,0	

## Anomalias no Exemplo

- Além do problema de redundância a tabela anterior apresenta as seguintes anomalias de atualização:
  - Anomalia de alteração** Se o preço de um produto for alterado, todas as linhas devem ser varridas para alteração.
  - Anomalia de inclusão** Na inclusão de um novo cliente, o mesmo deve estar relacionado a uma venda;
  - Anomalia de exclusão** Na exclusão de um cliente, os dados dos pedidos relacionados são perdidos;

## Primeira Forma Normal (1FN)

- A 1FN visa eliminar colunas multi-valoradas;
- No nosso exemplo o conjunto de colunas relacionados com os produtos são multi-valorados, ou seja, um pedido possui vários produtos;
- Devido a este problema a tabela possui várias linhas com dados duplicados para que um pedido com todos os seus produtos seja armazenado;
- Para obtermos tabelas obedecendo a 1FN devemos decompor a tabela não normalizada em várias tabelas, sendo que deve existir uma tabela para cada conjunto de colunas multi-valoradas. As tabelas resultantes possuirão como chave primária a chave primária da tabela original mais uma coluna identificadora do conjunto de colunas multi-valoradas.

## Exemplo com a 1FN

- Para o nosso exemplo teremos as seguintes tabelas obedecendo a 1FN:

```
Pedido(número pedido, prazo entrega, cliente, CNPJ, endereço,  
        cidade, UF, código vendedor, vendedor, valor total pedido)  
Item-de-pedido(*número pedido, código produto,  
        descrição produto, unidade produto, quantidade produto,  
        valor unitário produto, valor total produto)  
*Item-de-pedido.número pedido: Pedido.número pedido
```



## Primeira Forma Normal (1FN) I

Pedido(número pedido, prazo entrega, cliente, CNPJ, endereço, cidade, UF, código vendedor, vendedor, valor total pedido)

NP	PE	CI	CNP	E	Ci	UF	CV	V	VTP
1	10	C	003	cc3	Yyy	YY	25	W	232,0
2	15	D	004	dd4	Yyy	YY	26	Z	931,0
3	30	A	001	aa1	Xxx	XX	25	W	80,0
4	21	A	001	aa1	Xxx	XX	25	W	280,0
5	10	C	003	cc3	Yyy	YY	26	Z	204,0
6	30	B	002	bb2	Xxx	XX	25	W	472,0
7	15	A	001	aa1	Xxx	XX	25	W	135,0

## Primeira Forma Normal (1FN) II

Item-de-pedido(  
  \*número pedido,  
  código produto,  
  descrição produto,  
  unidade produto,  
  quantidade produto,  
  valor unitário produto,  
  valor total produto)  
\*Item-de-pedido.  
  número pedido:  
  Pedido.número pedido

NP	CP	DP	UP	QP	VUP	VT
1	101	ff	UN	2,0	10,0	20,0
1	102	gg	UN	7,0	20,0	140,0
1	103	hh	KG	2,4	30,0	72,0
2	102	gg	UN	6,0	20,0	120,0
2	103	hh	KG	4,3	30,0	129,0
2	104	ii	KG	5,8	40,0	232,0
2	105	jj	L	9,0	50,0	450,0
3	101	ff	UN	8,0	10,0	80,0
4	102	gg	UN	10,0	20,0	200,0
4	104	ii	KG	2,0	40,0	80,0
5	102	gg	UN	3,0	20,0	60,0
5	104	ii	KG	2,6	40,0	94,0
5	105	jj	L	1,0	50,0	50,0
6	104	ii	KG	1,8	40,0	72,0
6	105	jj	L	2,0	50,0	100,0
6	106	kk	L	5,0	60,0	300,0
7	102	gg	UN	3,0	20,0	60,0
7	103	hh	KG	2,5	30,0	75,0

# Dependência Funcional I

- Até o momento estudamos a 1FN, para estudarmos as demais formas normais precisamos entender o conceito de **dependência funcional**;
- Um conjunto de colunas não vazio  $A$  depende funcionalmente de um conjunto de colunas não vazio  $B$  (denotado por  $B \rightarrow A$ ) quando o valor de  $B$  determina o valor de  $A$ . Isto é, para cada valor do **determinante**  $B$  existe um único valor de  $A$ ;
- No nosso exemplo temos  $NP \rightarrow PE$ , ou seja, **prazo de entrega** depende funcionalmente de **número pedido**;

# Dependência Funcional II

## Dependência Funcional Total

- A dependência funcional total ocorre quando uma tabela possui uma chave primária composta (formada por várias colunas) e o valor de uma coluna (que não está na chave primária) depende de todas as colunas da chave primária;
- Na tabela **item-de-pedido** de nosso exemplo temos  $NP, CP \rightarrow QP$ , isto é, o coluna **quantidade** depende de forma total das colunas **número do pedido** e **código do produto**;

# Dependência Funcional III

## Dependência Funcional Parcial

- A dependência funcional parcial acontece quando o valor de uma coluna depende de parte de uma chave primária composta;
- Em nosso exemplo temos  $CP \rightarrow UP$ , a coluna **unidade** depende parcialmente da coluna **código do produto** na tabela **item-de-pedido**;

## Dependência Funcional Transitiva

- Quando uma coluna  $A$  depende de uma coluna  $B$  que por sua vez depende da chave primária  $C$ , então  $A$  depende transitivo de  $C$ ;
- Voltando ao nosso exemplo, na entidade **pedido** temos  $NP \rightarrow V$ , pois  $NP \rightarrow CV$  e  $CV \rightarrow V$ . Isto significa que **vendedor** depende transitivamente do **número do pedido**

## Segunda Forma Normal (2FN) I

- A 2FN consiste em eliminar as dependências funcionais parciais;
- Esta eliminação gera novas tabelas para os conjuntos de colunas com dependência funcional parcial;
- Cada tabela gerada possui como chave primária as colunas das quais havia a dependência funcional parcial;

## 2FN Aplicada ao Exemplo

- Considerando a tabela **item-do-pedido** de nosso exemplo, existe uma dependência funcional parcial que pode ser eliminada gerando a tabela **produto**<sup>2</sup>;
- Neste momento temos o seguinte esquema relacional:

Pedido(número pedido, prazo entrega, cliente, CNPJ, endereço, cidade, UF, código vendedor, vendedor, valor total pedido)

Produto(código produto, descrição produto, unidade produto)

Item-de-pedido(\*número pedido, \*código produto, quantidade produto, valor unitário produto, valor total produto)

\*Item-de-pedido.número pedido: Pedido.número pedido

\*Item-de-pedido.código produto: produto.código produto

---

<sup>2</sup>Em nosso exemplo a coluna **valor unitário** ficou na tabela **Item-de-pedido**. Apesar de criarmos uma certa redundância, poderíamos ter esta coluna também na tabela **Produto**. Desta maneira o **valor unitário** de **Produto** armazena o valor atual dos produtos e o **valor unitário** de **Item-de-pedido** armazena o valor unitário em cada pedido.

## Segunda Forma Normal (2FN) I

Produto(código produto, descrição produto, unidade produto)

CP	DP	UP
101	ff	UN
102	gg	UN
103	hh	KG
104	ii	KG
105	jj	L
106	kk	L



## Segunda Forma Normal (2FN) II

Item-de-pedido(\*número pedido,  
\*código produto,  
quantidade produto,  
valor unitário produto,  
valor total produto)  
\*Item-de-pedido.número pedido:  
Pedido.número pedido  
\*Item-de-pedido.código produto:  
produto.código produto

NP	CP	QP	VUP	VT
1	101	2,0	10,0	20,0
1	102	7,0	20,0	140,0
1	103	2,4	30,0	72,0
2	102	6,0	20,0	120,0
2	103	4,3	30,0	129,0
2	104	5,8	40,0	232,0
2	105	9,0	50,0	450,0
3	101	8,0	10,0	80,0
4	102	10,0	20,0	200,0
4	104	2,0	40,0	80,0
5	102	3,0	20,0	60,0
5	104	2,6	40,0	94,0
5	105	1,0	50,0	50,0
6	104	1,8	40,0	72,0
6	105	2,0	50,0	100,0
6	106	5,0	60,0	300,0
7	102	3,0	20,0	60,0
7	103	2,5	30,0	75,0

## Terceira Forma Normal (3FN)

- A 3FN tem como objetivo eliminar as dependências transitivas;
- Na retirada de uma dependência transitiva, cria-se uma nova tabela contendo as colunas com dependência transitiva. A chave primária desta tabela será a coluna que *intermediou* a dependência transitiva;

## 3FN Aplicada ao Exemplo

- Em nosso exemplo, na tabela **Pedido** temos que **vendedor** depende transitivamente de **número pedido** por intermédio de **código vendedor**;
- Além da dependência transitiva anterior temos uma dependência transitiva envolvendo as colunas do cliente. Neste caso, podemos tomar o **CNPJ** como chave primária da tabela a ser criada;
- Aplicando a 3FN nas duas situações levantadas temos:

Vendedor(código vendedor, vendedor)

Cliente(CNPJ, cliente, endereço, cidade, UF)

Pedido(número pedido, prazo entrega, \*CNPJ, \*código vendedor,  
valor total pedido)

\*Pedido.CNPJ: Cliente.CNPJ

\*Pedido.código vendedor: Vendedor.código vendedor

## Terceira Forma Normal (3FN)

Vendedor(código vendedor,  
vendedor)

CV	V
25	Y
26	Z

Cliente(CNPJ, cliente,  
endereço, cidade, UF)

CNP	CI	E	Ci	UF
001	A	aa1	Xxx	XX
002	B	bb2	Xxx	XX
003	C	cc3	Yyy	YY
004	D	dd4	Yyy	YY

Pedido(número pedido,  
prazo entrega, \*CNPJ,  
\*código vendedor,  
valor total pedido)

\*Pedido.CNPJ: Cliente.CNPJ

\*Pedido.código vendedor:

Vendedor.código vendedor

NP	PE	CNP	CV	VTP
1	10	003	25	232,0
2	15	004	26	931,0
3	30	001	25	80,0
4	21	001	25	280,0
5	10	003	26	204,0
6	30	002	25	472,0
7	15	001	25	135,0

## Terceira Forma Normal (3FN)

- A tabela **Cliente** ainda não está na 3FN, pois o **UF** depende transitivamente de **CNPJ** por intermédio de **cidade**;
- Poderíamos criar a tabela **Cidade(cidade, UF)** para resolver esta dependência, porém se incluirmos o coluna **código cidade** temos uma chave primária melhor do que o nome da cidade.

# Esquema Relacional Normalizado

Finalmente chegamos a um esquema relacional que atende a 3FN:

Vendedor(código vendedor, vendedor)

Cidade(código cidade, cidade, UF)

Cliente(CNPJ, cliente, endereço, \*código cidade, UF)

\*Cliente.código cidade: Cidade.código cidade

Pedido(número pedido, prazo entrega, \*CNPJ, \*código vendedor,  
valor total pedido)

\*Pedido.CNPJ: Cliente.CNPJ

\*Pedido.código vendedor: Vendedor.código vendedor

Produto(código produto, descrição produto, unidade produto)

Item-de-pedido(\*número pedido, \*código produto,  
quantidade produto, valor unitário produto,  
valor total produto)

\*Item-de-pedido.número pedido: Pedido.número pedido

\*Item-de-pedido.código produto: produto.código produto

# Exemplo de Banco de Dados Normalizado I

Cliente(CNPJ, cliente, endereço, \*código cidade)

CNP	CI	E	CC
001	A	aa1	001
002	B	bb2	001
003	C	cc3	002
004	D	dd4	002

Pedido(número pedido, prazo entrega, \*CNPJ, \*código vendedor, valor total pedido)

NP	PE	CNP	CV	VTP
1	10	003	25	232,0
2	15	004	26	931,0
3	30	001	25	80,0
4	21	001	25	280,0
5	10	003	26	204,0
6	30	002	25	472,0
7	15	001	25	135,0

Vendedor(código vendedor, vendedor)

CV	V
25	Y
26	Z

Cidade(código cidade, cidade, UF)

CC	Ci	UF
001	Xxx	XX
002	Yyy	YY

# Exemplo de Banco de Dados Normalizado

Produto(código produto,  
descrição produto, unidade  
produto)

CP	DP	UP	VUP
101	ff	UN	10,0
102	gg	UN	20,0
103	hh	KG	30,0
104	ii	KG	40,0
105	jj	L	50,0
106	kk	L	60,0

Item-de-pedido(\*número pedido,  
\*código produto, quantidade  
produto, valor unitário  
produto, valor total produto)

NP	CP	QP	VUP	VT
1	101	2,0	10,0	20,0
1	102	7,0	20,0	140,0
1	103	2,4	30,0	72,0
2	102	6,0	20,0	120,0
2	103	4,3	30,0	129,0
2	104	5,8	40,0	232,0
2	105	9,0	50,0	450,0
3	101	8,0	10,0	80,0
4	102	10,0	20,0	200,0
4	104	2,0	40,0	80,0
5	102	3,0	20,0	60,0
5	104	2,6	40,0	94,0
5	105	1,0	50,0	50,0
6	104	1,8	40,0	72,0
6	105	2,0	50,0	100,0
6	106	5,0	60,0	300,0
7	102	3,0	20,0	60,0
7	103	2,5	30,0	75,0





Elmasri, R. and Navathe, S. B. (2011).

***Sistemas de banco de dados.***

Pearson Addison Wesley, São Paulo, 6 edition.



Ramakrishnan, R. and Gehrke, J. (2008).

***Sistemas de gerenciamento de banco de dados.***

McGrawHill, São Paulo, 3 edition.