

# Normalização em bancos de dados

## 1. Anomalia de inserção, exclusão e modificação

Anomalias são mudanças em dados que podem gerar uma inconsistência no banco de dados relacional. Uma inconsistência é geralmente representada por situações em que dados que deveriam ser iguais, apresentam valores diferentes em várias tabelas do banco de dados.

### 1.1. Anomalia de inclusão

Não deve ser possível adicionar um dado a não ser que outro dado esteja disponível. Por exemplo, não deve ser permitido cadastrar um dependente para um funcionário que não esteja cadastrado. Caso seja possível, temos uma anomalia de inclusão.

### 1.2. Anomalia de exclusão

Ao excluirmos um registro, os registros relacionados em outras tabelas devem também ser excluídos. Por exemplo, ao excluirmos um funcionário, os registros dos dependentes relacionados a esse funcionário também devem ser excluídos. Caso isso não ocorra, temos uma anomalia de exclusão.

### 1.3. Anomalia de modificação

Ao modificarmos um dado em uma tabela, os dados relacionados em outras tabelas também devem ser modificados. Por exemplo, se o código do funcionário for atualizado na tabela de funcionários, esse código também deve ser atualizado na tabela de dependentes. Caso isso não ocorra, temos uma anomalia de modificação.

## 2. Normalização

A normalização consiste em um processo de análise de uma relação para assegurar que ela seja bem formada, ou seja, em uma relação normalizada podemos inserir, excluir ou modificar registros sem criar anomalias. A normalização consiste em decompor as relações que contém anomalias para produzir outras relações menores e melhor estruturadas.

O processo de normalização foi proposto por Dr. E. F. Codd em 1972 como uma forma de evitar as anomalias descritas anteriormente. Assim, o objetivo da normalização é remover a duplicação de dados e, consequentemente, minimizar a redundância. A remoção da duplicação de dados permite:

- Reduzir o espaço físico necessário para armazenar o banco de dados;
- Melhorar a organização dos dados;
- Reduzir o impacto de atualizações, inserções e remoções nos dados do banco de dados.

### 2.1. Primeira Forma Normal (1FN)

Uma tabela está na Primeira Forma Normal (1FN) se, e somente se, quando todos os atributos são atômicos e monovalorados. Um **atributo atômico** é aquele que não pode ser divisível como gênero ou cor dos olhos. Um **atributo monovalorado** é aquele que pode assumir apenas um valor, como estado civil ou estatura.

A seguir, temos um exemplo de normalização de uma tabela para a 1FN.

Antes da normalização:

clientes			
id_cliente (pk)	nome_cliente	endereco	telefone
1	Marcelo	Rua Prudente de Moraes, 12 - Curitiba	3605-5223, 3605-5141
2	João	Avenida Hermes da Fonseca, 15 - São Paulo	4225-5889
3	Pedro	Rua Engenheiro Roberto Freire, 20 - Porto Alegre	5223-8556, 5223-8557

Nesse exemplo, temos um atributo não atômico (endereço) e um atributo multivalorado (telefone).

Processo de normalização:

- **Se houver atributos compostos:** gerar atributos atômicos a partir dos atributos compostos;
- **Se houver atributos multivalorados:** criar uma tabela para esses atributos e relacionar a nova tabela com a tabela original.

Após a normalização:

clientes					
id_cliente (pk)	nome_cliente	tipo_logradouro (fk)	logradouro	numero	id_cidade (fk)
1	Marcelo	2	Prudente de Moraes	12	1
2	João	1	Hermes da Fonseca	15	2
3	Pedro	2	Roberto Freire	20	3

telefones		
id_telefone (pk)	id_cliente (fk)	telefone
1	1	3605-5223
2	1	3605-5141
3	2	4225-5889
4	3	5223-8556
5	3	5223-8557

logradouros	
id_logradouro (pk)	tipo_logradouro
1	Avenida
2	Rua

cidades	
id_cidade (pk)	nome_cidade
1	Curitiba
2	São Paulo
3	Porto Alegre

## 2.2. Segunda Forma Normal (2FN)

Dependências funcionais:

- **Dependência funcional:** existe em um relacionamento entre dois ou mais atributos de forma que o valor de um atributo identifique o valor para cada um dos outros atributos.
- **Dependência funcional parcial:** existe quando a chave primária é composta e os atributos não chave dependem de parte dessa chave primária.

Uma tabela está na Segunda Forma Normal (2FN) se, e somente se, ela estiver na 1FN e não possuir dependência funcional parcial. **Importante:** se a tabela não possuir uma chave primária composta, o processo de normalização 2FN não se faz necessário pois não existirão dependências funcionais parciais.

A seguir, temos um exemplo de normalização de uma tabela para a 2FN.

Antes da normalização:

peças				
id_peça (pk)	id_fornecedor (pk)	nome_fornecedor	telefone_fornecedor	qtde_estoque
0009	121	ABC Materiais	2365-6532	512
0023	122	XYZ Materiais	4465-8632	263
0065	121	ABC Materiais	2365-6532	196
0071	123	Hidra Materiais Hidráulicos	2956-8653	89
0073	122	XYZ Materiais	4465-8632	296

**Chave primária composta:** id\_peça + id\_fornecedor.

Nesse exemplo, a tabela está em sua 1FN, não possuindo atributos não atômicos ou multivalorados, mas existe uma dependência funcional parcial onde os atributos "nome\_fornecedor" e "telefone\_fornecedor" estão vinculados a uma parte da chave primária composta (id\_fornecedor).

Processo de normalização:

- **Se houver dependência funcional parcial:** (1) Criar uma nova tabela; (2) Remover os atributos que não dependam integralmente da chave primária; (3) Inserir esses atributos na nova tabela.

Após a normalização:

peças		
id_peça (pk)	id_fornecedor (fk)	qtde_estoque
0009	121	512
0023	122	263
0065	121	196
0071	123	89
0073	122	296

**Chave primária composta:** id\_peça + id\_fornecedor.

fornecedores		
id_fornecedor (pk)	nome_fornecedor	telefone_fornecedor
121	ABC Materiais	2365-6532
122	XYZ Materiais	4465-8632
123	Hidra Materiais Hidráulicos	2956-8653

### 2.3. Terceira Forma Normal (3FN)

- **Dependência funcional transitiva:** ocorre quando um campo não chave depende de outro campo também não chave.

Uma tabela está na Terceira Forma Normal (3FN) se, e somente se, ela estiver na 1FN e na 2FN e não possuir dependência funcional transitiva.

A seguir, temos um exemplo de normalização de uma tabela para a 3FN.

Antes da normalização:

vendas				
nota_fiscal (pk)	id_vendedor	nome_vendedor	id_produto	quantidade_vendida
15326	002	Leila	132	10
15327	006	Ana	153	12
15328	002	Leila	143	11
15329	007	Renato	132	9
15330	009	Fábio	153	12

Nesse exemplo, a tabela está em sua 1FN, não possuindo atributos não atômicos ou multivalorados e como não existe uma chave primária composta, a análise da 2FN não se faz necessária. Mas existe um caso de dependência funcional transitiva onde o atributo "nome\_vendedor" é dependente de um campo não chave (id\_vendedor).

Processo de normalização:

- **Se houver dependência funcional transitiva:** (1) Criar uma nova tabela; (2) Inserir os atributos que participam da dependência funcional transitiva nessa nova tabela.

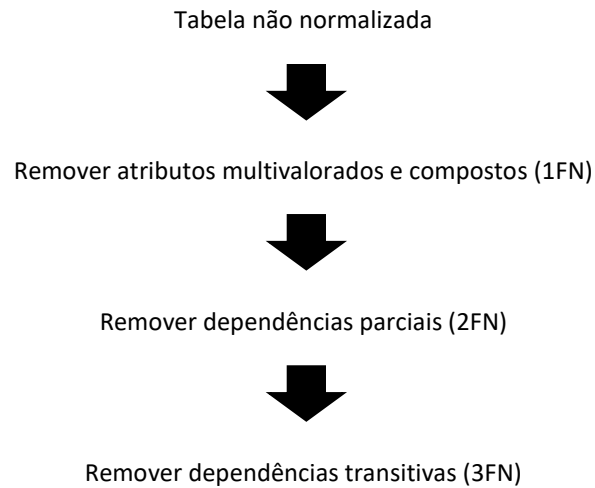
Após a normalização:

vendas			
nota_fiscal (pk)	id_vendedor (fk)	id_produto	quantidade_vendida
15326	002	132	10
15327	006	153	12
15328	002	143	11
15329	009	132	9
15330	007	153	12

vendedores	
id_vendedor (pk)	nome_vendedor
002	Leila
006	Ana
007	Renato
009	Fábio

## 2.4. Resumo

*"O processo de normalização compreende o uso de um conjunto de regras, chamados de formas normais. Ao analisarmos o banco de dados e verificarmos que ele respeita as regras da primeira forma normal, então podemos dizer que o banco está na "primeira forma normal". Caso o banco respeite as primeiras três regras, então ele está na "terceira forma normal". Mesmo existindo mais conjuntos de regras para outros níveis de normalização, a terceira forma normal é considerada o nível mínimo necessário para grande parte das aplicações". [Microsoft 2007]*



## ATIVIDADES

1. Verifique se a tabela a seguir está normalizada. Caso ela não esteja, identifique a forma normal necessária e realize o processo de normalização da tabela.

pessoas			
id_pessoa (pk)	nome	endereço	telefone
1	Ary	Avenida Getúlio Vargas, 1000 – Centro – Patos de Minas – MG	(34) 3821-0000 (34) 9979-0000 (34) 9964-0000
2	Tatiana	Avenida Brasil, 966 – Centro – Belo Horizonte – MG	(34) 3822-0000 (34) 9976-0000
3	Ana	Rua Minas Gerais, 100 – Bairro Brasil – Recife – PE	(31) 3184-0000
4	Luís	Praça da Liberdade, 27 – Bairro Esperança – São Paulo – SP	(11) 3257-0000

Resolução:

A normalização necessária é a 1FN, visto que a tabela possui atributos não atômicos (endereço) e atributos multivalorados (telefone).

pessoas							
id_pessoa (pk)	nome	id_logradouro (fk)	logradouro	numero	bairro	id_cidade (fk)	id_uf (fk)
1	Ary	1	Getúlio Vargas	1000	Centro	1	1
2	Tatiana	1	Brasil	966	Centro	2	1
3	Ana	2	Minas Gerais	100	Brasil	3	2
4	Luís	3	Liberdade	27	Esperança	4	3

telefones			
id_telefone (pk)	id_pessoa (fk)	ddd	telefone
1	1	34	3821-0000
2	1	34	9979-0000
3	1	34	9964-0000
4	2	34	3822-0000
5	2	34	9976-0000
6	3	31	3184-0000
7	4	11	3257-0000

logradouros	
id_logradouro (pk)	tipo_logradouro
1	Avenida
2	Rua
3	Praça

cidades	
id_cidade (pk)	nome_cidade
1	Patos de Minas
2	Belo Horizonte
3	Recife
4	São Paulo

estados	
id_uf (pk)	sigla_uf
1	MG
2	PE
3	SP

2. Verifique se a tabela a seguir está normalizada. Caso ela não esteja, identifique a forma normal necessária e realize o processo de normalização da tabela.

vendas				
id_venda (pk)	id_produto (pk)	nome_produto	valor_unitario	quantidade_vendida
1	1	Sabão em pó	25.50	2
2	2	Sabonete	2.7	5
3	1	Sabão em pó	25.50	3
4	2	Sabonete	2.7	2

**Chave primária composta:** id\_venda + id\_produto.

Resolução:

A tabela encontra-se em 1FN, não possuindo atributos não atômicos ou multivalorados. Mas faz-se necessária a normalização 2FN visto que a tabela possui atributos com dependência funcional parcial por conta da chave primária composta "id\_venda + id\_produto": os atributos "nome\_produto" e "valor\_unitario" estão vinculados ao atributo "id\_produto", que é parte da chave primária composta.

vendas		
id_venda (pk)	id_produto (fk)	quantidade_vendida
1	1	2
2	2	5
3	1	3
4	2	2

**Chave primária composta:** id\_venda + id\_produto.

produtos		
id_produto (pk)	nome_produto	valor_unitario
1	Sabão em pó	25.50
2	Sabonete	2.7

3. Verifique se a tabela a seguir está normalizada. Caso ela não esteja, identifique a forma normal necessária e realize o processo de normalização da tabela.

funcionarios				
id_funcionario (pk)	nome_funcionario	id_cargo	nome_cargo	salario
1	Ary	1	Professor	4.500
2	Tatiana	2	Advogado	6.900
3	Ana	3	Secretaria	1.150
4	Luis	4	Analista	6.000
5	Rodrigo	1	Professor	4.500

Resolução:

A tabela encontra-se em 1FN, não possuindo atributos não atômicos ou multivalorados. Como não existe uma chave primária composta para a tabela, a normalização 2FN não se faz necessária. Mas faz-se necessária a normalização 3FN visto que existe uma dependência funcional transitiva, onde os atributos "nome\_cargo" e "salario" são dependentes de um campo não chave (id\_cargo).

funcionarios		
id_funcionario (pk)	nome_funcionario	id_cargo (fk)
1	Ary	1
2	Tatiana	2
3	Ana	3
4	Luis	4
5	Rodrigo	1

cargos		
id_cargo (pk)	nome_cargo	salario
1	Professor	4.500
2	Advogado	6.900
3	Secretaria	1.150
4	Analista	6.000



**FONTES**

METROPOLE DIGITAL. **Normalização avançada**. Disponível em:  
<<https://materialpublic.imd.ufrn.br/curso/disciplina/3/45/7>>. Acesso em: 27 out. 2021.

METROPOLE DIGITAL. **Normalização básica**. Disponível em:  
<<https://materialpublic.imd.ufrn.br/curso/disciplina/3/45/6>>. Acesso em: 27 out. 2021.

MICROSOFT. **Descrição dos conceitos de normalização banco de dados básicos**. Disponível em:  
<<http://support.microsoft.com/kb/283878/pt-br>>. Acesso em: 28 out. 2011.

WIKIPEDIA. **Normalização de dados**. Disponível em: < [https://pt.wikipedia.org/wiki/Normalização\\_de\\_dados](https://pt.wikipedia.org/wiki/Normalização_de_dados)>. Acesso em: 27 out. 2021.