

1 Introdução

A normalização de banco de dados é o processo de organizar os dados em um banco de dados relacional de tal forma que se minimiza a redundância e a dependência de dados. É uma etapa crucial no design de um banco de dados relacional, pois ajuda a garantir a integridade e a consistência dos dados.

2 Primeira Forma Normal (1FN)

A primeira forma normal (1FN) é a forma mais básica de normalização. Em 1FN, cada tabela no banco de dados tem uma chave primária e todos os dados são atômicos, ou seja, não podem ser divididos ainda mais. Por exemplo, um campo que contém o nome de uma pessoa deve ser dividido em dois campos separados, um para o primeiro nome e outro para o último nome.

Um exemplo de uma tabela que não está em 1FN seria uma tabela que contém informações do cliente com um campo chamado "Nome" que contém o primeiro e o último nome do cliente. Essa tabela não estaria em 1FN, porque os dados no campo "Nome" não são atômicos. Para levá-la à 1FN, o campo "Nome" precisaria ser dividido em campos separados para "Primeiro Nome" e "Último Nome".

3 Segunda Forma Normal (2FN)

A segunda forma normal (2FN) constrói sobre os princípios da 1FN removendo as dependências parciais. Uma dependência parcial ocorre quando um campo que não é a chave primária é dependente apenas de uma parte da chave primária. Na 2FN, todos os campos que não são chave primária devem ser dependentes da chave primária inteira.

Um exemplo de uma tabela que não está em 2FN seria uma tabela "Pedidos" com campos para "ID do Pedido" (a chave primária), "ID do Cliente", "ID do Produto" e "Preço do Produto". Nessa tabela, o campo "Preço do Produto" é dependente apenas da parte "ID do Produto" da chave primária "ID do Pedido" e "ID do Produto". Para levá-la à 2FN, o campo "Preço do Produto" precisaria ser removido e colocado em uma tabela separada "Produtos" com "ID do Produto" como sua chave primária.

4 Terceira Forma Normal (3FN)

A terceira forma normal (3FN) constrói sobre os princípios da 2FN removendo as dependências transitivas. Uma dependência transitiva ocorre quando um campo não-chave primária é dependente de outro campo não-chave primária. Na 3FN, todos os campos não-chave primária devem ser diretamente dependentes da chave primária.

Um exemplo de uma tabela que não está na 3FN seria uma tabela "Funcionários" com campos para "ID do Funcionário" (a chave primária), "ID do Departamento", "ID do Gerente" e "Salário do Gerente". Nesta tabela, o campo "Salário do Gerente" é dependente do campo "ID do Gerente", que não é a chave primária. Para trazer essa tabela para 3FN, o campo "Salário do Gerente" precisaria ser removido e colocado em uma tabela separada "Gerentes" com "ID do Gerente" como sua chave primária.

5 Exemplo de Transformação para a Primeira Forma Normal (1NF)

A seguir, vamos ver um exemplo de como transformar uma tabela não normalizada em uma tabela normalizada na primeira forma. Considere a tabela abaixo, que armazena informações sobre clientes:

ID	Nome	Endereço	Telefone	Email
1	João Silva	Rua A, 123	(11) 1234-5678	joao.silva@email.com
2	Maria Santos	Rua B, 456	(11) 8765-4321	maria.santos@email.com

Nesta tabela, o campo "Nome" contém o nome completo do cliente, o que viola a regra da atômidade de dados. Para tornar a tabela normalizada na primeira forma, precisamos dividir o campo "Nome" em dois campos distintos: "Primeiro Nome" e "Sobrenome". Além disso, precisamos adicionar uma chave primária à tabela, no caso, o campo "ID". A tabela ficaria assim:

ID	Primeiro Nome	Sobrenome	Endereço	Telefone	Email
1	João	Silva	Rua A, 123	(11) 1234-5678	joao.silva@email.com
2	Maria	Santos	Rua B, 456	(11) 8765-4321	maria.santos@email.com

Com essas alterações, a tabela está agora normalizada na primeira forma, pois cumpre as regras da atômidade de dados e possui uma chave primária.

6 Exemplo de Transformação para a Segunda Forma Normal (2NF)

Imagine que temos uma tabela chamada "Vendas", com os seguintes campos: "ID da Venda" (chave primária), "ID do Cliente", "ID do Produto", "Preço do Produto" e "Desconto". Neste exemplo, "Preço do Produto" e "Desconto" são dependentes apenas de "ID do Produto", e não da chave primária completa, "ID da Venda" e "ID do Produto".

Para transformar essa tabela para a segunda forma normal (2NF), podemos criar uma nova tabela chamada "Produtos", com os campos "ID do Produto" (chave primária) e "Preço". Então, removeríamos o campo "Preço do Produto"

da tabela "Vendas" e adicionaríamos uma chave estrangeira "ID do Produto" que se relaciona com a tabela "Produtos".

A tabela "Vendas" agora teria os campos: "ID da Venda" (chave primária), "ID do Cliente", "ID do Produto" (chave estrangeira) e "Desconto". A tabela "Produtos" teria os campos "ID do Produto" (chave primária) e "Preço". Desta forma, garantimos que todos os campos na tabela "Vendas" estão dependentes da chave primária completa e removemos a dependência parcial.

Vendas		[0.5ex]	
ID da Venda	ID do Cliente	ID do Produto	
1	1001	50	
2	1002	51	
3	1003	52	

Produtos		[0.5ex]	
ID do Produto	Preço		
50	\$10.00		
51	\$20.00		
52	\$30.00		

7 Transformando uma tabela para Terceira Forma Normal (3NF)

Uma tabela pode ser transformada para a terceira forma normal (3NF) ao remover as dependências transitivas. Uma dependência transitiva ocorre quando um campo não-chave primário é dependente de outro campo não-chave primário. Na 3NF, todos os campos não-chave primários devem ser diretamente dependentes da chave primária.

Como exemplo, considere uma tabela "Funcionários" com campos "ID do Funcionário" (chave primária), "ID do Departamento", "ID do Gerente" e "Salário do Gerente". Neste caso, o campo "Salário do Gerente" é dependente do campo "ID do Gerente", que não é a chave primária. Para transformar esta tabela para a 3NF, o campo "Salário do Gerente" deve ser removido e colocado em uma tabela separada "Gerentes" com "ID do Gerente" como sua chave primária.