

Normalizar um banco de dados

A. Para normalizar, geralmente escolhe-se dividir as tabelas. Neste caso, mantenha a chave na tabela original e copie-a para a nova tabela, junto com os atributos relacionados. Na nova tabela, defina uma chave. Se nenhum atributo sozinho puder ser a chave, use uma chave composta (formada por mais de um atributo).

B. Excluir atributo redundante.

1. Primeira Forma Normal (1FN): Se algum campo tiver mais de um valor, crie uma nova tabela para isso use a regra A.
 - 1.1. Resolvido a 1FN em todas as tabelas? Se não, repita o processo (2).
2. Segunda Forma Normal (2FN): Se houver uma chave composta, veja se cada parte dela define outros atributos. Se sim, use a regra A.
 - 2.1. Resolvido 2FN em todas as tabelas? Se não, repita o processo.
3. Terceira Forma Normal (3FN): Se algum atributo que não é chave depender de outro atributo que também não é chave, uso da regra A ou B.
 - 3.1. Resolvido a 3FN em todas as tabelas? Se não, repita o processo.

Exemplo resolvido. Normalize a tabela abaixo.

Pedido_ID	Data_Pedido	Cliente_ID	Cliente_Nome	Endereço_Cliente	Produto_ID	Produto_Nome	Quantidade	Preço_Unitário	Total_Pedido	email
1	01/10/2024	101	João Silva	Rua A, 123	301	Teclado	2	100.00	200.00	joao@email.com, joaos@email.com
2	02/10/2024	102	Maria Souza	Rua B, 456	302	Mouse	3	50.00	150.00	maria@email.com, marias@email.com
3	01/10/2024	101	João Silva	Rua A, 123	303	Monitor	1	500.00	50.000	joao@email.com, joaos@email.com

Definir a chave



<u>Pedido_ID</u>	<u>Data_Pedido</u>	<u>Cliente_ID</u>	<u>Cliente_Nome</u>	<u>Endereço_Cliente</u>	<u>Produto_ID</u>	<u>Produto_Nome</u>	<u>Quantidade</u>	<u>Preço_Unitário</u>	<u>Total_Pedido</u>	<u>email</u>
1	01/10/2024	101	João Silva	Rua A, 123	301	Teclado	2	100.00	200.00	joao@email.com, joaos@email.com
2	02/10/2024	102	Maria Souza	Rua B, 456	302	Mouse	3	50.00	150.00	maria@email.com, marias@email.com
3	01/10/2024	101	João Silva	Rua A, 123	303	Monitor	1	500.00	50.000	joao@email.com, joaos@email.com

1FN

Cliente => email

<u>Cliente_ID</u>	<u>email</u>
101	joao@email.com, joaos@email.com
102	maria@email.com, marias@email.com
101	joao@email.com, joaos@email.com



<u>Cliente_ID</u>	<u>email</u>
101	joao@email.com
101	marias@email.com
102	joao@email.com, joaos@email.com
102	joaos@email.com

2FN

Pedido_id => { data_pedido, Cliente_id, quantidade, preço_unitário, total_pedido }

Cliente_id => { cliente_nome, endereço_cliente }

Produto_id => { produto_nome }

Pedido_ID	Data_Pedido	Cliente_ID	Cliente_Nome	Endereço_Cliente	Produto_ID	Produto_Nome	Quantidade	Preço_Unitário	Total_Pedido
1	01/10/2024	101	João Silva	Rua A, 123	301	Teclado	2	100.00	200.00
2	02/10/2024	102	Maria Souza	Rua B, 456	302	Mouse	3	50.00	150.00
3	01/10/2024	101	João Silva	Rua A, 123	303	Monitor	1	500.00	50.000



Pedido_ID	Data_Pedido	Cliente_ID	Produto_ID	Quantidade	Total_Pedido
1	01/10/2024	101	301	2	200.00
2	02/10/2024	102	302	3	150.00
3	01/10/2024	101	303	1	50.000

Faz-se a 1ª divisão

Cliente_ID	Cliente_Nome	Endereço_Cliente
101	João Silva	Rua A, 123
102	Maria Souza	Rua B, 456

Faz-se a 2ª divisão

Produto_ID	Produto_Nome	Preço_Unitário
301	Teclado	100.00
302	Mouse	50.00
303	Monitor	500.00

3FN

{ quantidade } => { Total_pedido }

Neste caso como total_pedido pode ser obtido dinamicamente, pode-se eliminá-lo sem criar uma tabela nova.

Se um pedido pode ter mais de um produto e somente um cliente, usa-se pedido_id e produto_id como chave.

Pedido_ID	Data_Pedido	Cliente_ID	Produto_ID	Quantidade	Total_Pedido
1	01/10/2024	101	301	2	200.00
2	02/10/2024	102	302	3	150.00
3	01/10/2024	101	303	1	50.000