

Normalização de Bancos de Dados

Adaptação de

<http://www.blogdati.com.br/index.php/2010/03/normalizacao-em-banco-de-dados/>

É a aplicação de algumas regras a todas as tabelas do banco de dados com a finalidade de tornar o banco mais eficiente, eliminando redundâncias e que uma única tabela contenha mais que um assunto, entre outros.

Primeira Forma Normal – 1FN

Uma tabela está na 1FN se nenhum dos seus campos contiver informações repetidas nem tiver mais de um valor.

Como fazer uma tabela ficar na 1FN:

- Identificar a chave primária da tabela (o campo que nunca se repete),
- Identificação do campo que contém informações duplicadas ou agrupadas e movê-lo para outra tabela,
- Criação de uma nova tabela com chave primária para conter os dados repetidos acima,
- Criar um relacionamento entre a tabela principal e a secundária

Exemplo:

id	nome	endereço	telefone
1	José da Silva	R Justiniano, 3300	2342-2342/ 2242-1210
2	Emanuel Bezerra	Travessa Quinze, 200	6952-4231
3	João dos Santos	Av. Brasil, 220	6758-3422/ 3452-4576 / 4592-3422
4	Maria Mariana	Av. Bandeirantes, 2987	9990-3022

Observe que o campo telefone está agrupando informações, o que denota a criação de uma nova tabela, para que cada telefone venha em apenas um único campo.

Veja como devem ficar as novas tabelas:

cliente

id	nome	endereço
1	José da Silva	R Justiniano, 3300
2	Emanuel Bezerra	Travessa Quinze, 200
3	João dos Santos	Av. Brasil, 220
4	Maria Mariana	Av. Bandeirantes, 2987

cliente_telefones

id	cliente_id	telefone
1	1	2342-2342
2	1	2242-1210
3	2	6952-4231
4	3	6758-3422
5	3	3452-4576
6	3	4592-3422
7	4	9990-3022

Segunda Forma Normal – 2FN

Uma tabela está na 2FN somente se estiver antes na 1FN e todos os campos diferentes da chave primária dependerem unicamente da chave primária. Se um campo depende apenas em parte da chave primária isso viola a 2FN.

Passos para deixar a tabela na 2FN:

- Identificar campos que não são funcionalmente dependentes da chave primária
- Remover a coluna da tabela criando uma nova tabela

Exemplo:

id	data	produto	nome	qtd	valor	total
1	01/06/2016	1234	HD de 450GB da SS	2	100	200
2	02/06/2016	1235	HD de 180GB da HP	1	60	60
3	03/06/2016	1235	HD de 180GB da HP	4	60	240
4	04/06/2016	1234	HD de 450GB da SS	6	100	600

Esta tabela armazena dados de vendas. Veja que temos um campo (total), que depende somente em parte da chave primária. É um chamado campo calculado, o resultado de dois outros campos da tabela. Normalmente o que se faz é remover o campo total e quando precisar dele efetuar uma consulta multiplicando os dois campos. Além disso veja que temos campos com valores duplicados, então devemos mover estes campos para outra tabela, deixando na primária apenas o código do produto.

A tabela final na 2FN será esta:

id	data	produto	qtd	valor
1	01/06/2016	1234	2	100
2	02/06/2016	1235	1	60
3	03/06/2016	1235	4	60
4	04/06/2016	1234	6	100

produto_id	produto	nome
1	1234	HD de 450GB da SS
2	1235	HD de 180GB da HP
3	1235	HD de 180GB da HP
4	1234	HD de 450GB da SS

Então relacionar as duas tabelas.

Terceira Forma Normal – 3FN

Uma tabela está na 3FN somente se estiver na 2FN e todos os campos não chave forem mutuamente independentes, isto é, não deve haver dependência funcional entre eles e todos os campos dependem única e exclusivamente da chave primária de forma irredutível.

Passos para deixar uma tabela na 3FN:

- Identificar os campos que são funcionalmente dependentes de outros campos não chave
- Remover estes campos

Exemplo

id	data	produto	qtde	valor	total
1	01/06/2016	1234	2	100	200
2	02/06/2016	1235	1	60	60
3	03/06/2016	1235	4	60	240
4	04/06/2016	1234	6	100	600

Como podemos ver acima existe o campo total não depende inteiramente da chave primária, pois é um campo calculado, portanto sem a necessidade de ser armazenado. A qualquer momento que tenhamos necessidade dele podemos efetuar uma consulta multiplicando os dois campos.

Resultado:

id	data	produto	qtd	valor
1	01/06/2016	1234	2	100
2	02/06/2016	1235	1	60
3	03/06/2016	1235	4	60
4	04/06/2016	1234	6	100

Tomando estes cuidados estaremos tornando o resultado do nosso trabalho mais eficiente e consequentemente menos sujeito a erros.

É bom lembrar que cada caso é um caso e precisará de experiência para identificar o que fazer em cada tabela.