Vem Ser Tech - Dados (5) | #1104

Matheus Muniz
Id: 1104005
MM

Formas normais

As formas normais são um conjunto de regras para garantir que um banco de dados relacional esteja organizado de maneira coerente e sem redundâncias desnecessárias. As formas normais são importantes porque elas ajudam a evitar problemas como inconsistências de dados, dificuldade de manutenção e desempenho ruim do banco de dados.

Existem várias formas normais, sendo as mais comuns:

> Primeira Forma Normal (1FN) A 1FN exige que todas as tabelas em um banco de dados relacional tenham dados atômicos, ou seja, que não haja campos que contenham mais de um valor. Além disso, cada campo em uma tabela deve ter um nome único e cada registro deve ter um identificador exclusivo (chave primária).
>
> Segunda Forma Normal (2FN) A 2FN exige que todas as tabelas em um banco de dados relacional estejam na 1FN e que cada campo em uma tabela dependa totalmente da chave primária da tabela. Isso significa que não deve haver campos que dependam apenas de parte da chave primária.
>
> Terceira Forma Normal (3FN) A 3FN exige que todas as tabelas em um banco de dados relacional estejam na 2FN e que não haja dependências transitivas entre os campos de uma tabela. Isso significa que um campo deve depender apenas da chave primária da tabela e não de outros campos.
>
> Forma Normal de Boyce-Codd (BCNF) A BCNF é uma forma normal mais rigorosa do que a 3FN. Ela exige que cada dependência funcional em uma tabela seja uma chave candidata da tabela. Isso garante que não haja redundância de dados em uma tabela.

Além dessas formas normais, existem outras, como a Quarta Forma Normal (4FN), a Quinta Forma Normal (5FN) e a Forma Normal de Domínio/Elementar (DKNF). Essas formas normais são usadas em situações específicas e são menos comuns do que as quatro primeiras.

Para exemplificar, faremos um caso de uso.

Vamos supor que temos uma tabela chamada "Pedidos" que contém informações sobre os pedidos que os clientes fizeram em uma loja virtual:

| **Pedido** | **Cliente** | **Endereço** | **Cidade** | **Estado** | **CEP** | **Produto** | **Quantidade** | **Preço** | |:----------:|:----------:|:----------:|:----------:|:----------:|:----------:|:----------:|:----------:|:----------:| | 1 | Ana | Rua A | São Paulo | SP | 01000-000 | Camiseta | 2 | 50,00 | | 2 | Bruno | Rua B | Rio de Janeiro | RJ | 02000-000 | Camiseta | 1 | 50,00 | | 3 | Carla | Rua C | São Paulo | SP | 01000-000 | Calça | 3 | 80,00 |

Nesta tabela, temos informações que não estão atomicamente armazenadas, como o "Endereço", "Cidade", "Estado" e "CEP". Além disso, há uma dependência transitiva entre "Produto", "Quantidade" e "Preço", já que o preço pode ser calculado a partir do produto e da quantidade.

Para normalizar essa tabela, podemos dividir em três tabelas: "Pedidos", "Clientes" e "Produtos".

Tabela Pedidos: | **Pedido** | **ClienteID** | **ProdutoID** | **Quantidade** | |:----------:|:----------:|:----------:|:----------:| | 1 | 1 | 1 | 2 | | 2 | 2 | 1 | 1 | | 3 | 3 | 2 | 3 |

Tabela Clientes:

| **ClienteID** | **Nome** | **Endereço** | **Cidade** | **Estado** | **CEP** | |:----------:|:--------:|:----------:|:----------:|:----------:|:----------:| | 1 | Ana | Rua A | São Paulo | SP | 01000-000 | | 2 | Bruno | Rua B | Rio de Janeiro | RJ | 02000-000 | | 3 | Carla | Rua C | São Paulo | SP | 01000-000 |

Tabela Produtos:

| **ProdutoID** | **Nome** | **Preço** | |:----------:|:--------:|:--------:| | 1 | Camiseta | 50,00 | | 2 | Calça | 80,00 |

Na tabela "Pedidos", a chave primária é a combinação de "Pedido", "ClienteID" e "ProdutoID". Na tabela "Clientes", a chave primária é "ClienteID" e na tabela "Produtos", a chave primária é "ProdutoID". Com essa estrutura, todas as informações estão atomicamente armazenadas, e não há dependência transitiva entre os campos de cada tabela.

Com isso, temos uma estrutura normalizada que está na 3NF, e que nos permite gerenciar os dados de forma mais eficiente e segura. Note que este exemplo é apenas ilustrativo e que o processo de normalização pode ser mais complexo em casos reais.

Referências e materiais complementares

Normalização em Bancos de Dados

NORMALIZAÇÃO DE DADOS E AS FORMAS NORMAIS

Normalização de Bancos de Dados Relacionais

Normalizando um banco de dados por meio das 3 principais formas

Próximo Tópico