

A normalização de dados é um processo realizado dentro das tabelas para evitar anomalias.

Com base no conteúdo e nos links sugeridos do hipertexto 3 do banco de dados I, descreva o processo de normalização citando as formas normais apresentadas.

# Resposta

Normalização é o processo de organização de dados em um banco de dados. Isso inclui a criação de tabelas e o estabelecimento de relações entre essas tabelas de acordo com as regras projetadas para proteger os dados e tornar o banco de dados mais flexível, eliminando a redundância e a dependência inconsistente.

Dados redundantes desperdiçam espaço em disco e criam problemas de manutenção. Se os dados existentes em mais de um local devem ser alterados, eles devem ser alterados exatamente da mesma maneira em todos os locais. Uma alteração de endereço do cliente é muito mais fácil de implementar se esses dados são armazenados apenas na tabela Clientes e em nenhum outro lugar no banco de dados.

O que é uma "dependência inconsistente"? Embora seja intuitivo para um usuário procurar na tabela Clientes o endereço de um determinado cliente, talvez não faça sentido procurar o salário do funcionário que chama esse cliente. O salário do funcionário está relacionado ou depende do funcionário e, portanto, deve ser movido para a tabela Funcionários. As dependências inconsistentes podem dificultar o acesso aos dados porque o caminho para encontrar os dados pode estar ausente ou quebrado.

Há algumas regras para normalização do banco de dados. Cada regra é chamada de "formulário normal". Se a primeira regra for observada, o banco de dados deverá estar na "primeira forma normal". Se as três primeiras regras forem observadas, o banco de dados será considerado como "terceiro formulário normal". Embora outros níveis de normalização sejam possíveis, a terceira forma normal é considerada o nível mais alto necessário para a maioria dos aplicativos.

Assim como muitas regras e especificações formais, os cenários do mundo real nem sempre permitem a conformidade perfeita. Em geral, a normalização requer tabelas adicionais e alguns clientes acham isso complicado. Se você decidir violar uma das três primeiras regras de normalização, certifique-se de que seu aplicativo preveja qualquer problema que possa ocorrer, como dados redundantes e dependências inconsistentes.

As descrições a seguir incluem exemplos.

### **Primeiro formulário normal**

- Eliminar grupos repetidos em tabelas individuais.
- Crie uma tabela separada para cada conjunto de dados relacionados.
- Identifique cada conjunto de dados relacionados com uma chave primária.

Não use vários campos em uma única tabela para armazenar dados semelhantes. Por exemplo, para rastrear um item de inventário que pode vir de duas fontes possíveis, um registro de inventário pode conter campos para o Código do Fornecedor 1 e o Código do Fornecedor 2.

O que acontece quando você adiciona um fornecedor terceiro? Adicionar um campo não é a resposta; exige modificações de programa e tabela e não acomoda facilmente um número dinâmico de fornecedores. Em vez disso, coloque todas as informações do fornecedor em uma tabela separada chamada Fornecedores e vincule o inventário a fornecedores com uma chave de número de item ou fornecedores para “inventariá-los” com uma chave de código de fornecedor.

### **Segundo formulário normal**

- Crie tabelas separadas para conjuntos de valores que se aplicam a vários registros.
- Relacione essas tabelas com uma chave estrangeira.

Os registros não devem depender de nada além da chave primária de uma tabela (uma chave composta, se necessário). Por exemplo, considere o endereço de um cliente em um sistema de contabilidade. O endereço é necessário para a tabela Clientes, mas também para as tabelas Pedidos, Envio, Faturas, Contas a Receber e Coleções. Em vez de armazenar o endereço do cliente como uma entrada separada em cada uma dessas tabelas, armazene-o em um só lugar, na tabela Clientes ou em uma tabela Endereços separada.

### **Terceira forma normal**

- Elimine campos que não dependem da chave.

Os valores em um registro que não fazem parte da chave deste registro não pertencem à tabela. Em geral, sempre que o conteúdo de um grupo de campos pode se aplicar a mais de um único registro na tabela, considere colocar esses campos em uma tabela separada.

Por exemplo, em uma tabela de Recrutamento de Funcionários, o nome da universidade e o endereço de um candidato podem ser incluídos. Mas você precisa

de uma lista completa de universidades para envio de mensagens em grupo. Se as informações da universidade forem armazenadas na tabela Candidatos, não haverá como listar universidades sem candidatos atuais. Crie uma tabela de Universidades separada e vincule-a à tabela Candidatos com uma chave de código da universidade.

**EXCEÇÃO:** a adesão à terceira forma normal, embora teoricamente desejável, nem sempre é prática. Se você tiver uma tabela Clientes e quiser eliminar todas as possíveis dependências entre campos, deverá criar tabelas separadas para cidades, CEPs, representantes de vendas, classes de clientes e qualquer outro fator que possa ser duplicado em vários registros. Em teoria, vale a pena buscar a normalização. No entanto, muitas tabelas pequenas podem degradar o desempenho ou exceder as capacidades abertas de arquivo e memória.

Pode ser mais viável aplicar a terceira forma normal apenas aos dados que mudam com frequência. Se alguns campos dependentes permanecerem, projete seu aplicativo para exigir que o usuário verifique todos os campos relacionados quando qualquer um deles for alterado.

### **Outros formulários de normalização**

A quarta forma normal, também chamada de BCNF (Boyce Codd Normal Form), e a quinta forma normal existem, mas raramente são consideradas no design prático. Ignorar essas regras pode resultar em não mais que um design de banco de dados perfeito, mas não deve afetar a funcionalidade.