

Introdução

Trabalhar com dados é algo cada vez mais necessário, considerando nossa realidade predominantemente conectada ao mundo virtual. Produzimos uma grande quantidade de dados todos os dias e essa quantidade aumentou significativamente nas últimas décadas.

Os bancos de dados não estão restritos às empresas, as pessoas também têm seu próprio armazenamento! Por exemplo, ao manter uma planilha de gastos pessoais está armazenando dados significativos para seu controle financeiro. Logo, faz sentido ter um conhecimento melhor de como manter esses valores e ter boas práticas para manipulá-los.

Normalização

A normalização é focada na prevenção de problemas com repetição e atualização de dados, assim como o cuidado com a integridade dos dados. Este conceito foi apresentado originalmente em um artigo científico publicado pela IBM de autoria do matemático [Edgar F. Codd](#), intitulado "Um modelo de dados relacionais para grandes bancos de dados compartilhados" (1970). Codd se centra nos valores dos elementos relacionados no banco de dados, não em ligações ou agrupamentos específicos.

Este modelo resultou em um processo flexível e menos custoso para o armazenamento e processamento de dados. Foi algo tão notório que seu autor ganhou o [prêmio Turing](#) em 1981 e a Forbes em 2002 marcou este modelo relacional como uma das principais inovações dos últimos 85 anos.

Mediante a decomposição das relações presentes no banco de dados, este processo busca anomalias, isto é, repetições e redundâncias entre os dados. Ao identificar anomalias, o conjunto de regras da normalização tentará eliminá-las e redefinir as relações afetadas, para que tudo se encaixe em seu devido lugar depois das alterações.

Formas Normais

Seguindo o conceito de padronização, temos estas regras estruturadas e agrupadas em três níveis que são utilizadas para ajudar as tabelas do banco de dados. Estes grupos são denominados formas normais e neste artigo serão apresentadas quatro formas que são

utilizadas. Cada forma normal segue requisitos da forma anterior, ou seja, se mantém uma herança de requisitos, com exceção da primeira forma que não possui uma antecessora.

Primeira Forma

Nesta primeira forma tratamos as repetições, e também nos certificamos que os atributos estão sendo armazenados de forma única, isto é, não há nenhum outro atributo com os valores da mesma linha na tabela.

Vemos a chave primária da tabela, e se é necessário criar outra, associamos a tabela original com a segunda precisamente por essa chave. Vamos criar um exemplo (informações fictícias) com informação sobre duas pessoas:

Código	Nome	Localização	Telefone
1	José	Curitiba	(44) 91234-5678
		Paraná	(44) 94834-8348
2	Arturo	Recife	(81) 91234-8765
		Pernambuco	(81) 91328-7811

A princípio, identificamos a chave primária, neste caso, é o atributo de código. Considere que entre os atributos estão os valores associados a chave, então precisamos fragmentar estes valores, desta forma é necessário criar uma nova tabela.

Código	Nome	Cidade	Estado
1	José	Curitiba	Paraná
2	Arturo	Recife	Pernambuco

Criamos os atributos cidade e estado, porque estas informações estavam em um único atributo, sendo que é mais útil ter essas informações separadas, para filtrar os dados por exemplo.

Código	Telefone
1	(44) 91234-5678
1	(44) 94834-8348
2	(81) 91234-8765
2	(81) 91328-7811

Esta nova tabela foi criada para poder relacionar telefones com o atributo código, que na tabela principal é a chave primária, sendo definida como chave estrangeira. Deixamos todos os dados definidos individualmente, ainda assim relacionados.

Resumindo, os atributos e valores posteriores a primeira forma são atômicos, isto é, são dados que não podem ser modificados nem divididos pois estão em sua forma mínima. Além disso, as tabelas devem conter a chave primária como nula, para efeitos de identificação e relação entre dados e tabelas.

Resumindo, os atributos e valores posteriores a primeira forma são atômicos, isto é, são dados que não podem ser modificados nem divididos pois estão em sua forma mínima. Além disso, as tabelas devem conter a chave primária como nula, para efeitos de identificação e relação entre dados e tabelas.

Segunda Forma

A segunda forma trabalha focada nas possíveis redundâncias nas tabelas, em especial, define se os atributos da tabela dependem inteiramente da chave primária. Os atributos que não dependem ou dependem parcialmente da chave são associados a uma outra tabela, agora com uma relação clara com a chave primária da tabela original. Em outras palavras, a chave primária é convertida em chave estrangeira (ou externa) na nova tabela.

Vamos seguir com outro exemplo, similar a tabela anterior.

Código	Nome	Código Voo	Origem	Destino
1	José	101	Santiago	São Paulo
2	Arturo	102	Bogotá	Buenos Aires

Considere que os campos de origem e destino não têm relação direta com o campo de código, mas têm uma relação direta com o código de voo, já que são informações relacionadas a uma viagem aérea, por exemplo. Assim, podemos mover essas informações a uma nova tabela sem que os dados percam as relações originais.

Código Voo	Origem	Destino
101	Santiago	São Paulo
102	Bogotá	Buenos Aires

Por isso, a tabela original elimina os dados que não são necessários nela, porém eles seguem relacionados em uma tabela secundária.

Código	Nome	Código Voo
1	José	101
2	Arturo	102

Vale a pena lembrar que a segunda forma normal está de acordo com as regras da primeira forma normal, e assim sucessivamente.

Terceira Forma

Na terceira forma normal trabalhamos precisamente com a organização dos atributos que dependem uns dos outros, porém que não são atributos chaves (primárias ou estrangeiras). Caso necessário, é criada uma tabela secundária para reestruturar a relação de dependência entre os atributos. Essas tabelas devem ter a chave primária ou estrangeira.

Usamos agora um exemplo referente a modelos de carros:

Placa	Modelo	Ano	Código Fabricação	Nome Fábrica
123X	Modelo 1	2016	1	A
456Y	Modelo 2	2012	2	B
789Z	Modelo 3	2012	3	C

Tenha em conta que existe uma dependência entre o atributo 'nome de fábrica' e 'ano' com o 'código de fábrica', entretanto, estes atributos não dependem da chave principal da tabela que é 'placa'.

Código Fabricação	Nome Fábrica	Ano
1	A	2016
2	B	2012
3	C	2012

Neste caso, criamos uma nova tabela para relacionar o nome da fábrica com seu código, e também removemos as relações de dependência entre atributos que não são chaves da tabela original.

Placa	Modelo	Código Fabricação
123X	Modelo 1	1
456Y	Modelo 2	2
789Z	Modelo 3	3

Sempre é bom recordar que para uma tabela está em terceira forma normal, antes disso, deve estar definida de acordo com a primeira e segunda formas normais. Estas foram as três formas normais principais. Existe uma quarta forma, não considerada principal porém útil, que é apresentada na sequência.

Quarta Forma

A quarta e última forma normal é focada em eliminar dependências multivariadas entre os atributos chave, isto é, se há mais atributos (além das chaves primárias ou estrangeiras) que se repetem na tabela. Se isso ocorrer, geramos novas tabelas para eliminar tal redundância e manter as relações entre os atributos.

Música	Artista	Álbum
Música 1	Artista A	Álbum 1
Música 1	Artista B	Álbum 2
Música 2	Artista A	Álbum 1

Leve em conta que o campo 'música' está relacionado com 'artista' e 'álbum', porém, o artista e o álbum podem não estar relacionados, porque sabemos que a mesma canção pode estar em vários álbuns diferentes e também pode ser cantada por diferentes artistas. Por isso, o ideal é que se produza a divisão desta tabela e assim eliminar repetições entre os dados.

Música	Álbum
Música 1	Álbum 1
Música 1	Álbum 2
Música 2	Álbum 1

E agora o campo de música é relacionado ao campo de artista.

Música	Artista
Música 1	Artista A
Música 1	Artista B
Música 2	Artista A

Conclusão

A normalização é um passo importante para quem está modelando um banco de dados relacional, e certamente resulta em uma maior eficiência na hora de abstrair o banco e seus atributos.

Fonte: Alura