Semana 1: postagem 2 do fórum de discussão

Instruções

Responda à pergunta de discussão postada por seu instrutor abaixo e, em seguida, responda a pelo menos duas postagens de colegas (até domingo antes da meia-noite).

Leia a seção inicial do Capítulo 2 chamada "Projetando Tabelas" (pp. 32-34).

Defina, com suas próprias palavras, o que significa um banco de dados ser "normalizado". Discuta brevemente uma possível consequência quando um banco de dados não é normalizado.

**Projetando tabelas**

Até agora, temos apenas duas tabelas em nosso banco de dados e não falamos realmente sobre como decidimos o que vai em cada tabela, exceto na maneira muito informal de fazer o que parecia razoável. Esse design, que inclui tabelas, colunas e relacionamentos, é mais corretamente chamado de esquema.

Projetar um esquema de banco de dados com mais de duas dúzias de tabelas pode ser bastante desafiador se os dados forem complexos. Os designers de banco de dados ganham dinheiro sendo bons nessa difícil tarefa. Felizmente, para bancos de dados relativamente simples, com até talvez dez tabelas, é possível criar um design razoavelmente bom apenas aplicando algumas regras básicas, em vez de precisar aplicar regras de maneira mais formal.

Nesta seção, veremos o banco de dados de amostra simples que estamos começando a construir e descobriremos uma maneira de decidir de quais tabelas precisamos.

**Entendendo algumas regras básicas do polegar**

Quando um banco de dados é projetado, geralmente é normalizado; ou seja, um conjunto de regras é aplicado para garantir que os dados sejam divididos de maneira apropriada. No Capítulo 12, veremos o projeto de banco de dados de maneira formal. Para começar, tudo o que precisamos são algumas regras básicas simples. Essas regras são apenas para ajudá-lo a entender o banco de dados inicial, chamado “ *bpsimple ”* , que usaremos para explorar SQL e PostgreSQL neste e nos próximos capítulos. Nós sugerimos fortemente que você não apenas leia estas regras, e então corra para projetar um banco de dados com 20 tabelas. Trabalhe seu caminho através do livro – pelo menos até o Capítulo 12.

**Regra um: divida os dados em colunas**

A primeira regra é colocar apenas uma informação, ou atributo de dados, em cada coluna. Isso vem naturalmente para a maioria das pessoas, desde que pensem conscientemente sobre isso. Em nossa planilha original, já dividimos naturalmente as informações de cada cliente em colunas diferentes, então o nome foi separado do CEP, por exemplo.

Em uma planilha, essa regra apenas simplifica o trabalho com os dados; por exemplo, para classificar pelo CEP. Em um banco de dados, no entanto, é essencial que os dados sejam corretamente divididos em atributos.

Por que isso é tão importante em bancos de dados? Do ponto de vista prático, é difícil especificar que você deseja os dados entre o vigésimo nono e o trigésimo quinto caracteres de uma coluna de endereço, porque é onde reside o CEP. É provável que haja algum lugar onde a regra não se aplique e você obtenha os dados errados. Outra razão para que os dados sejam divididos corretamente é que todas as colunas em um banco de dados devem ter o mesmo tipo, ao contrário de uma planilha, que perdoa bastante os tipos de dados em uma coluna.

**Regra dois: tenha uma maneira única de identificar cada linha**

Você vai se lembrar que quando tentamos decidir como identificar cada linha no exemplo de planilha no início deste capítulo, tivemos o problema de não ter certeza do que seria único. Como foi mencionado, isso ocorreu porque não havia chave primária. Em geral, não precisa ser uma única coluna exclusiva; pode ser um par de colunas juntas ou, ocasionalmente, até mesmo a combinação de três colunas que identifica exclusivamente uma linha. É raro e provavelmente um erro se você precisar de mais de três colunas para identificar exclusivamente uma linha.

De qualquer forma, deve haver uma maneira de dizer, com absoluta certeza, se eu olhar o conteúdo de uma determinada coluna, ou grupo de colunas nesta linha, sei que ela terá um valor diferente de todas as outras linhas desta tabela . Se você não conseguir encontrar uma coluna, ou no máximo uma combinação de três colunas, que identifique exclusivamente cada linha, é hora de adicionar uma coluna extra para cumprir essa finalidade. Em nossa tabela de clientes, adicionamos uma coluna extra, “ customer\_id ”, para identificar cada linha.

**Regra três: remover informações repetidas**

Lembre-se de que, quando tentamos armazenar as informações do pedido na tabela do cliente, ela parecia bastante desordenada por causa dos grupos repetidos. Para cada cliente, repetimos as informações do pedido quantas vezes fossem necessárias. Isso significava que nunca poderíamos saber quantas colunas eram necessárias para os pedidos. Em um banco de dados, o número de colunas em uma tabela é efetivamente fixado pelo design. Portanto , devemos decidir antecipadamente quantas colunas precisamos, de que tipo são e nomear cada coluna antes de podermos armazenar quaisquer dados. Nunca tente armazenar grupos de dados repetidos em uma única linha.

A maneira de contornar essa restrição é fazer exatamente o que fizemos com nossos pedidos e dados de clientes: dividir os dados em tabelas separadas. Em seguida, você pode unir as tabelas quando precisar de dados de ambas as tabelas. Em nosso exemplo, usamos a coluna “ customer\_id ” para unir as duas tabelas.

Mais formalmente, o que tínhamos era um relacionamento de muitos para um; ou seja, pode haver muitos pedidos recebidos de um único cliente.

**Regra Quatro: Acerte no Nome**

Esta é ocasionalmente a regra mais difícil de implementar bem. O que chamamos de tabela ou coluna? Tabelas e colunas devem ter nomes curtos e significativos. Se você não pode decidir como chamar algo, geralmente é uma pista de que nem tudo está bem no design da sua tabela e coluna.

Além de criar nomes apropriados, a maioria dos designers de banco de dados tem suas próprias regras práticas, ou convenções de nomenclatura, que eles usam para garantir que a nomenclatura de tabelas e colunas em um banco de dados seja consistente. Não tenha alguns nomes de tabelas no singular e alguns no plural. Por exemplo, em vez de nomear uma mesa como escritório e os outros departamentos, use escritório e departamento. Se você decidir sobre uma regra de nomenclatura para uma coluna id—talvez o nome da tabela com um \_id anexado—mantenha essa regra. Se você usar abreviações, use-as sempre de forma consistente. Se uma coluna em uma tabela for uma chave para outra tabela (uma chave estrangeira, conforme explicado no Capítulo 12), tente dar a elas o mesmo nome base. Em um banco de dados complexo, pode ser muito irritante quando os nomes não são muito consistentes, como “ customer\_id ”, “ customer\_ident ”, “ cust\_id ” e “ cust\_no ”.

Alcançar esse objetivo aparentemente simples de acertar os nomes costuma ser surpreendentemente desafiador, mas as recompensas na manutenção simplificada são consideráveis.