

Laboratório Prático: Normalização, Chaves e Restrições em Banco de Dados Relacional



Tempo estimado necessário: 25 minutos

Instruções

Neste laboratório, você explorará normalização, chaves e restrições no Datasette. Inicialmente, você aprenderá a minimizar a redundância e a inconsistência de dados em um banco de dados normalizando tabelas. Em seguida, você aprenderá a usar chaves para identificar de forma única um registro em uma tabela, estabelecer um relacionamento entre tabelas e identificar a relação entre elas. Por fim, você aprenderá sobre diferentes tipos de restrições do modelo relacional que ajudam a manter a integridade dos dados em um modelo de dados relacional.

Objetivos

Após concluir este laboratório, você será capaz de:

- Minimizar a redundância e a inconsistência de dados em um banco de dados usando normalização.
- Usar chaves para identificar de forma única um registro em uma tabela, estabelecer um relacionamento entre tabelas e identificar a relação entre elas.
- Manter a integridade dos dados em um modelo de dados relacional usando restrições.

Pré-requisitos

Software Usado neste Laboratório

Neste laboratório, você usará [Datasette](#), uma ferramenta multiuso de código aberto para explorar e publicar dados.

Conjunto de dados usado neste Laboratório

Neste laboratório, você usará um conjunto de dados **BookShop**.

Exercício 1: Normalização

Neste exercício, você aprenderá sobre a primeira forma normal (1NF) e implementará a segunda forma normal (2NF).

Tarefa A: Primeira forma normal (1NF)

Nesta tarefa de normalização, você trabalhará com a tabela **BookShop**. A imagem a seguir mostra a tabela **BookShop**:

BOOK_ID	TITLE	AUTHOR_NAME
B101	Introduction to Algorithms	Thomas H. Cormen
B201	Structure and Interpretation of Computer Programs	Harold Abelson
B301	Deep Learning	Ian Goodfellow
B401	Algorithms Unlocked	Thomas H. Cormen
B501	Machine Learning: A Probabilistic Perspective	Kevin P. Murphy

Você responderá algumas perguntas para determinar se a tabela acima está em 1NF.

1. A tabela acima possui linhas únicas?
 - ▶ Dica
 - ▶ Resposta
2. Cada célula da tabela acima tem um valor único/atômico?

- ▶ Dica
 - ▶ Resposta
3. Por definição, uma tabela está em 1NF se cada atributo nessa relação contém dados de valor único e cada tupla nessa relação é única. A tabela acima está na primeira forma normal?
- ▶ Dica
 - ▶ Resposta
4. Se sua resposta à pergunta 3 for Não, como você pode normalizar a tabela para garantir a primeira forma normal?
- ▶ Dica
 - ▶ Resposta

Tarefa B: Segunda forma normal (2NF)

1. Baixe o script BookShop-CREATE-INSERT.sql abaixo, copie e cole-o no laboratório Datasette e execute-o. O script irá remover qualquer tabela BookShop anterior que exista, criar a nova tabela BookShop e preenchê-la com os dados de exemplo necessários para este laboratório.

BOOK_ID	TITLE	AUTHOR_ID
B101	Introduction to Algorithms	Thomas H
B201	Structure and Interpretation of Computer Pro...	Harold Ab
B301	Deep Learning	Ian Goodt
B401	Algorithms Unlocked	Thomas H
B501	Machine Learning: A Probabilistic Perspective	Kevin P. M

- [BookShop-CREATE-INSERT.sql](#)
- ▶ Clique aqui para ver as consultas dentro do script
- Dica:** Se você não tem certeza de como fazer o upload e executar o script no Datasette, siga os passos dados:
- Baixe o arquivo do script para o seu computador:
[BookShop-CREATE-INSERT.sql](#)
 - Abra o arquivo do script usando **Bloco de Notas** ou qualquer **editor de texto**.
 - **Copie** o conteúdo do arquivo **script** e cole na **área de texto** do datasette.
 - Clique no botão **Enviar consulta**.
 - As consultas serão executadas.
 - Por definição, uma relação está na segunda forma normal se já está em 1NF e não contém dependências parciais. Se você olhar para a tabela BookShop, verá que cada coluna na tabela é de valor único ou atômico, mas possui vários livros do mesmo autor. Isso significa que os detalhes de AUTHOR_ID, AUTHOR_NAME e AUTHOR_BIO para BOOK_ID B101 e B401 são os mesmos. À medida que o número de linhas na tabela aumenta, você estará armazenando desnecessariamente mais e mais ocorrências dessas mesmas informações. E se um autor atualizar sua biografia, você deve atualizar todas essas ocorrências.

BOOK_ID	TITLE	AUTHOR
B101	Introduction to Algorithms	Thomas H
B201	Structure and Interpretation of Computer Pro...	Harold Ab
B301	Deep Learning	Ian Goodf
B401	Algorithms Unlocked	Thomas H
B501	Machine Learning: A Probabilistic Perspective	Kevin P. M

3. Nesse cenário, para impor 2NF, você pode retirar as informações do autor, como AUTHOR_ID, AUTHOR_NAME e AUTHOR_BIO, da tabela BookShop para outra tabela, por exemplo, uma tabela chamada **BookShop_AuthorDetails**. Você então vincula cada livro na tabela BookShop à linha relevante na tabela BookShop_AuthorDetails, usando uma coluna comum única, como AUTHOR_ID, para vincular as tabelas. Para criar a nova tabela **BookShop_AuthorDetails**, copie o código abaixo e cole na área de texto do dataset. Clique no botão **Enviar consulta**.

```
CREATE TABLE BookShop_AuthorDetails
(AUTHOR_ID INTEGER NOT NULL, AUTHOR_NAME VARCHAR(30) NOT NULL,
AUTHOR_BIO VARCHAR(250), PRIMARY KEY (AUTHOR_ID)) ;
-----Inserir os registros da Bookshop nesta tabela.
insert into BookShop_AuthorDetails select DISTINCT AUTHOR_ID, AUTHOR_NAME, AUTHOR_BIO FROM BookShop;
select * from BookShop_AuthorDetails;
```

4. Agora você está armazenando as informações do autor apenas uma vez por autor e só precisa atualizá-las em um lugar; reduzindo a redundância e aumentando a consistência dos dados. Assim, a 2NF é garantida.

Exercício 2: Chaves

Neste exercício, você aprenderá como utilizar uma chave primária para identificar de forma exclusiva registros em uma tabela, usar uma chave estrangeira para estabelecer relacionamentos entre tabelas e discernir as relações entre elas.

Tarefa A: Chave Primária

- Por definição, uma chave primária é uma coluna ou grupo de colunas que identifica exclusivamente cada linha em uma tabela. Uma tabela não pode ter mais de uma chave primária. As regras para definir uma chave primária incluem:
 - Nenhuma duas linhas podem ter um valor de chave primária duplicado.
 - Cada linha deve ter um valor de chave primária.
 - Nenhum campo de chave primária pode ser nulo.
- Para identificar exclusivamente cada linha nas tabelas BookShop e BookShop_AuthorDetails, você criará uma chave primária. Defina a coluna BOOK_ID da tabela BookShop e a coluna AUTHOR_ID da tabela BookShop_AuthorDetails como chaves primárias para suas respectivas tabelas. Ambas as colunas foram declaradas como NOT NULL quando as tabelas foram criadas (verifique isso no script SQL ou na definição da tabela). A tabela BookShop_AuthorDetails herda tipos de dados e restrições de coluna, incluindo NOT NULL, da tabela pai BookShop).
- Para definir a coluna BOOK_ID da tabela BookShop como uma chave primária para cada uma das tabelas, copie o código abaixo e cole na **área de texto** do dataset. Clique no botão **Enviar consulta**.

```
--Remover a tabela.
DROP TABLE IF EXISTS BookShop;
-----Recriar com Chave Primária -----
CREATE TABLE BookShop (
BOOK_ID VARCHAR(4) NOT NULL,
```

```
TITLE VARCHAR(100) NOT NULL,
AUTHOR_NAME VARCHAR(30) NOT NULL,
AUTHOR_BIO VARCHAR(250),
AUTHOR_ID INTEGER NOT NULL,
PUBLICATION_DATE DATE NOT NULL,
PRICE_USD DECIMAL(6,2) CHECK(Price_USD>0) NOT NULL,PRIMARY KEY (BOOK_ID));
INSERT INTO BookShop VALUES
('B101', 'Introdução a Algoritmos', 'Thomas H. Cormen', 'Thomas H. Cormen é coautor de Introdução a Algoritmos, junto com Charles Leiserson
('B201', 'Estrutura e Interpretação de Programas de Computador', 'Harold Abelson', 'Harold Abelson, Ph.D., é Professor da Classe de 1922 de
('B301', 'Deep Learning', 'Ian Goodfellow', 'Ian J. Goodfellow é um pesquisador que trabalha em aprendizado de máquina, atualmente empregac
('B401', 'Algoritmos Desbloqueados', 'Thomas H. Cormen', 'Thomas H. Cormen é coautor de Introdução a Algoritmos, junto com Charles Leiserson
('B501', 'Aprendizado de Máquina: Uma Perspectiva Probabilística', 'Kevin P. Murphy', '', 157, '2012-08-24', 46);
-- Recuperar todos os registros da tabela
SELECT * FROM BookShop;
```

4. Para definir a coluna `AUTHOR_ID` da tabela `BookShop_AuthorDetails` como uma chave primária para cada uma das tabelas, copie o código abaixo e cole na **área de texto** do dataset. Clique no botão **Enviar consulta**.

```
--Remover a tabela.
DROP TABLE IF EXISTS BookShop_AuthorDetails;
-----Recriar outra tabela BookShop_AuthorDetails com o id do autor como chave primária
CREATE TABLE BookShop_AuthorDetails(AUTHOR_ID INTEGER NOT NULL,AUTHOR_NAME VARCHAR(30) NOT NULL,AUTHOR_BIO VARCHAR(250),PRIMARY KEY (AUTHOR_ID))
-----Inserir os registros da Bookshop nesta tabela.
insert into BookShop_AuthorDetails select DISTINCT AUTHOR_ID, AUTHOR_NAME, AUTHOR_BIO FROM BookShop;
select * from BookShop_AuthorDetails;
```

5. Após adicionar a chave primária, vamos tentar adicionar o mesmo registro na tabela `BookShop`, copie o código abaixo e cole na **área de texto** do dataset. Clique no botão **Enviar consulta**.

```
INSERT INTO BookShop VALUES
('B101', 'Introdução a Algoritmos', 'Thomas H. Cormen', 'Thomas H. Cormen é coautor de Introdução a Algoritmos, junto com Charles Leiserson
---Você receberá uma violação de restrição única.
select * from BookShop;
```

6. Agora você pode usar a coluna `BOOK_ID` para identificar exclusivamente cada linha na tabela `BookShop` e a coluna `AUTHOR_ID` para identificar exclusivamente cada linha na tabela `BookShop_AuthorDetails`.

Tarefa B: Chave Estrangeira

1. Por definição, uma chave estrangeira é uma coluna que estabelece um relacionamento entre duas tabelas. Ela atua como uma referência cruzada entre duas tabelas, pois aponta para a chave primária de outra tabela. Uma tabela pode ter múltiplas chaves estrangeiras referenciando chaves primárias de outras tabelas. Regras para definir uma chave estrangeira:

- o Uma chave estrangeira na tabela de referência deve corresponder à estrutura e ao tipo de dados da chave primária existente na tabela referenciada.
- o Uma chave estrangeira só pode ter valores presentes na chave primária referenciada.
- o Chaves estrangeiras não precisam ser únicas. Na maioria das vezes, não são.
- o Chaves estrangeiras podem ser nulas.

2. Para criar uma chave estrangeira para a tabela `BookShop`, defina sua coluna `AUTHOR_ID` como uma chave estrangeira, para estabelecer um relacionamento entre as tabelas `BookShop` e `BookShop_AuthorDetails`. Copie o código abaixo e cole na **área de texto** do dataset. Clique no botão **Enviar consulta**.

```
-----Adição de chave estrangeira na bookshop
DROP TABLE IF EXISTS BookShop;
CREATE TABLE BookShop (
  BOOK_ID VARCHAR(4) NOT NULL,
  TITLE VARCHAR(100) NOT NULL,
  AUTHOR_NAME VARCHAR(30) NOT NULL,
  AUTHOR_BIO VARCHAR(250),
  AUTHOR_ID INTEGER NOT NULL,
  PUBLICATION_DATE DATE NOT NULL,
```

```
PRICE_USD DECIMAL(6,2) CHECK(Price_USD>0) NOT NULL,PRIMARY KEY (BOOK_ID),  
FOREIGN KEY (AUTHOR_ID)  
REFERENCES BookShop_AuthorDetails(AUTHOR_ID)  
ON UPDATE NO ACTION  
ON DELETE NO ACTION);
```

Nota: A cláusula ON DELETE junto com a chave estrangeira é usada para configurar ações que ocorrem ao excluir linhas da tabela de referência. A cláusula ON UPDATE junto com a chave estrangeira é usada para tomar as ações definidas ao modificar os valores da chave de referência de linhas existentes.

NO ACTION simplesmente significa que, quando uma chave pai é atualizada, modificada ou excluída do banco de dados, nenhuma ação especial será tomada.

Se a ação configurada for definida como RESTRICT, então a aplicação estará proibida de excluir e modificar uma chave pai onde uma ou mais chaves filhas já estão presentes.

Ao configurar a ação como SET NULL, quando uma chave pai é excluída ou atualizada, a coluna de todas as chaves filhas que estão mapeadas para a chave pai será definida para conter valores SQL NULL.

SET DEFAULT é semelhante a SET NULL, exceto que as colunas das chaves filhas serão definidas para conter um valor padrão em vez de nulo.

3. Agora que você criou o relacionamento, cada livro na tabela BookShop está vinculado à linha relevante na tabela BookShop_AuthorDetails através do AUTHOR_ID.

Exercício 3: Restrições

Neste exercício, você revisará diferentes tipos de restrições do modelo relacional que são cruciais para manter a integridade dos dados em um modelo de dados relacional.

1. **Restrição de Integridade da Entidade:** A integridade da entidade garante que não existam registros duplicados dentro de uma tabela e que a coluna que identifica cada registro na tabela não seja duplicada e não seja nula. A existência de uma chave primária nas tabelas BookShop e BookShop_AuthorDetails satisfaz essa restrição de integridade, pois uma chave primária impõe a restrição NOT NULL e garante que cada linha na tabela tenha um valor que denota exclusivamente a linha.
2. **Restrição de Integridade Referencial:** A integridade referencial garante a existência de um valor referenciado se um valor de uma coluna de uma tabela referencia um valor de outra coluna. A existência da chave estrangeira (AUTHOR_ID) na tabela BookShop satisfaz essa restrição de integridade porque existe uma relação de referência cruzada entre as tabelas BookShop e BookShop_AuthorDetails. Como resultado dessa relação, cada livro na tabela BookShop está vinculado à linha relevante na tabela BookShop_AuthorDetails através das colunas AUTHOR_ID.
3. **Restrição de Integridade de Domínio:** A integridade de domínio garante clareza do propósito da coluna e consistência dos valores válidos. A tabela BookShop adere a essa restrição através da especificação de tipos de dados, comprimento, formato de data, restrições de verificação e restrições de nulo em sua instrução CREATE. Essa abordagem abrangente garante que os valores em cada coluna não sejam apenas válidos, mas também estejam em conformidade com as restrições de domínio especificadas.

▼ Click here to view queries inside the script

```
-- Drop the table in case it exists
```

```
DROP TABLE BookShop;
```

```
-- Create the table
```

```
CREATE TABLE BookShop (  
    BOOK_ID VARCHAR(4) NOT NULL,  
    TITLE VARCHAR(100) NOT NULL,  
    AUTHOR_NAME VARCHAR(30) NOT NULL,  
    AUTHOR_BIO VARCHAR(250),  
    AUTHOR_ID INTEGER NOT NULL,  
    PUBLICATION_DATE DATE NOT NULL,  
    PRICE_USD DECIMAL(6,2) CHECK(Price_USD>0) NOT NULL  
);
```

Parabéns! Você completou este laboratório e está pronto para o próximo tópico.

Author: [Pratiksha Verma](#)

© IBM Corporation. Todos os direitos reservados.