Exercício Simples – tabelas de livros – entidades – relacionamentos – Código inicial em Python e SQL

Prof. José Alfredo F. Costa (Julho / 2024)

Tabela com dez livros, incluindo os autores, títulos, editoras, datas de aquisição, estados dos livros, status de leitura e observações.

| **ID** | **Autor 1** | **Autor 2** | **Título** | **Editora** | **Data de Aquisição** | **Estado do Livro** | **Lido** | **Observação** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | João Silva | Maria Oliveira | Gestão de Processos | Editora Alfa | 2023-01-15 | Novo | Sim | Com a pessoa |
| 2 | Carlos Pereira | Ana Souza | Inteligência Artificial | Editora Beta | 2022-06-20 | Usado | Não | Emprestado a João |
| 3 | João Silva | Pedro Lima | Engenharia de Produção | Editora Alfa | 2021-11-05 | Novo | Sim | Com a pessoa |
| 4 | Rafael Mendes | Fernanda Costa | Direito e Tecnologia | Editora Gamma | 2020-03-10 | Novo | Não | Com a pessoa |
| 5 | Ana Souza | Lucas Martins | Ética em IA | Editora Delta | 2022-09-25 | Usado | Sim | Com a pessoa |
| 6 | Pedro Lima | Maria Oliveira | Pesquisa Operacional | Editora Beta | 2023-05-12 | Novo | Não | Emprestado a Ana |
| 7 | Fernanda Costa | João Silva | Automação Industrial | Editora Gamma | 2019-12-18 | Usado | Não | Com a pessoa |
| 8 | Lucas Martins | Rafael Mendes | Inovação e Sustentabilidade | Editora Delta | 2021-07-07 | Novo | Sim | Com a pessoa |
| 9 | Carlos Pereira | Pedro Lima | Robótica Avançada | Editora Alfa | 2022-11-30 | Usado | Não | Emprestado a Rafael |
| 10 | Maria Oliveira | Ana Souza | Big Data e Análise de Dados | Editora Beta | 2023-02-22 | Novo | Sim | Com a pessoa |

Esta tabela inicial contém informações detalhadas sobre os livros, permitindo uma análise de como organizar e gerenciar um banco de dados de livros. Os autores repetidos e as diferentes editoras podem ajudar na discussão sobre a estrutura de dados ideal para essa aplicação.

Para organizar a tabela de livros de forma eficiente utilizando Bancos de Dados Relacionais, podemos aplicar técnicas de normalização e modelagem de entidade-relacionamento. A normalização ajuda a eliminar redundâncias e inconsistências nos dados, dividindo a tabela original em várias sub-tabelas que estão logicamente relacionadas entre si. A modelagem de entidade-relacionamento nos permite visualizar essas relações de forma clara.

**1. Identificação das Entidades**

Primeiro, identificamos as principais entidades envolvidas na tabela de livros:

* **Autor**
* **Livro**
* **Editora**

**2. Criação das Sub-Tabelas**

**Tabela de Autores**

Cada autor deve ser armazenado apenas uma vez, com um identificador único.

| **ID\_Autor** | **Nome** |
| --- | --- |
| 1 | João Silva |
| 2 | Maria Oliveira |
| 3 | Carlos Pereira |
| 4 | Ana Souza |
| 5 | Pedro Lima |
| 6 | Rafael Mendes |
| 7 | Fernanda Costa |
| 8 | Lucas Martins |

**Tabela de Editoras**

Cada editora deve ser armazenada apenas uma vez, com um identificador único.

| **ID\_Editora** | **Nome** |
| --- | --- |
| 1 | Editora Alfa |
| 2 | Editora Beta |
| 3 | Editora Gamma |
| 4 | Editora Delta |

**Tabela de Livros**

Cada livro deve ter um identificador único e referenciar as editoras e autores.

| **ID\_Livro** | **Título** | **ID\_Editora** | **Data de Aquisição** | **Estado do Livro** | **Lido** | **Observação** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Gestão de Processos | 1 | 2023-01-15 | Novo | Sim | Com a pessoa |
| 2 | Inteligência Artificial | 2 | 2022-06-20 | Usado | Não | Emprestado a João |
| 3 | Engenharia de Produção | 1 | 2021-11-05 | Novo | Sim | Com a pessoa |
| 4 | Direito e Tecnologia | 3 | 2020-03-10 | Novo | Não | Com a pessoa |
| 5 | Ética em IA | 4 | 2022-09-25 | Usado | Sim | Com a pessoa |
| 6 | Pesquisa Operacional | 2 | 2023-05-12 | Novo | Não | Emprestado a Ana |
| 7 | Automação Industrial | 3 | 2019-12-18 | Usado | Não | Com a pessoa |
| 8 | Inovação e Sustentabilidade | 4 | 2021-07-07 | Novo | Sim | Com a pessoa |
| 9 | Robótica Avançada | 1 | 2022-11-30 | Usado | Não | Emprestado a Rafael |
| 10 | Big Data e Análise de Dados | 2 | 2023-02-22 | Novo | Sim | Com a pessoa |

**Tabela de Autor\_Livro**

Essa tabela é necessária para representar a relação muitos-para-muitos entre autores e livros.

| **ID\_Autor** | **ID\_Livro** |
| --- | --- |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 2 |
| 1 | 3 |
| 5 | 3 |
| 6 | 4 |
| 7 | 4 |
| 4 | 5 |
| 8 | 5 |
| 5 | 6 |
| 2 | 6 |
| 7 | 7 |
| 1 | 7 |
| 8 | 8 |
| 6 | 8 |
| 3 | 9 |
| 5 | 9 |
| 2 | 10 |
| 4 | 10 |

**3. Modelagem de Entidade-Relacionamento (ER)**

A seguir, um modelo ER básico para representar a relação entre as entidades.

* **Livro** possui uma relação com **Editora** (muitos-para-um).
* **Livro** possui uma relação com **Autor** através de uma tabela associativa **Autor\_Livro** (muitos-para-muitos).

**4. Normalização**

A normalização garante que nossos dados estejam livres de redundâncias e inconsistências. As tabelas acima estão normalizadas nas formas normais até a 3NF:

* **1NF (Primeira Forma Normal):** Todos os valores das colunas são atômicos.
* **2NF (Segunda Forma Normal):** Remoção de dependências parciais (cada atributo não chave é totalmente dependente da chave primária).
* **3NF (Terceira Forma Normal):** Remoção de dependências transitivas (nenhum atributo não chave depende de outro atributo não chave).
* **Dicas:** Utilizando essas sub-tabelas normalizadas, podemos manter um banco de dados relacional eficiente e escalável. A modelagem de entidade-relacionamento nos ajuda a entender e implementar essas relações de maneira clara, facilitando a manutenção e a consulta dos dados.

**Conexões entre as Entidades**

Para entender as conexões entre as entidades no banco de dados, vamos detalhar as relações e criar diagramas que representem essas conexões.

**Entidades e suas Relações**

1. **Livro**:
   * Relaciona-se com a **Editora** através de uma relação muitos-para-um.
   * Relaciona-se com **Autor** através de uma tabela associativa **Autor\_Livro** que representa uma relação muitos-para-muitos.
2. **Editora**:
   * Relaciona-se com **Livro** em uma relação um-para-muitos (uma editora pode publicar muitos livros).
3. **Autor**:
   * Relaciona-se com **Livro** através de uma tabela associativa **Autor\_Livro** que representa uma relação muitos-para-muitos.

**Diagrama Entidade-Relacionamento (ER)**

Vamos construir um diagrama ER básico para visualizar essas relações.

1. **Entidade Livro**:
   * Atributos: ID\_Livro, Título, ID\_Editora, Data de Aquisição, Estado do Livro, Lido, Observação
2. **Entidade Editora**:
   * Atributos: ID\_Editora, Nome
3. **Entidade Autor**:
   * Atributos: ID\_Autor, Nome
4. **Entidade Autor\_Livro**:
   * Atributos: ID\_Autor, ID\_Livro

O diagrama ER seria algo como:

plaintext

Copiar código

+----------------+ +----------------+

| Autor | | Editora |

|----------------| |----------------|

| ID\_Autor (PK) |1 M| ID\_Editora (PK) |

| Nome |----------------->| Nome |

+----------------+ +----------------+

|

|M

|

|

|1

+------------------+

| Autor\_Livro |

|------------------|

| ID\_Autor (FK,PK) |

| ID\_Livro (FK,PK) |

+------------------+

|M

|

|1

|

+----------------+

| Livro |

|----------------|

| ID\_Livro (PK) |

| Título |

| ID\_Editora (FK)|

| Data\_Aquisição |

| Estado\_Livro |

| Lido |

| Observação |

+----------------+

**Descrição das Relações**

1. **Relação Autor e Autor\_Livro**:
   * Um autor pode escrever muitos livros, e um livro pode ter muitos autores. A tabela associativa **Autor\_Livro** gerencia essa relação muitos-para-muitos.
   * A tabela **Autor\_Livro** possui chaves estrangeiras **ID\_Autor** e **ID\_Livro** que são chaves primárias compostas.
2. **Relação Editora e Livro**:
   * Uma editora pode publicar muitos livros, mas cada livro é publicado por apenas uma editora. Isso cria uma relação um-para-muitos entre **Editora** e **Livro**.
3. **Relação Livro e Autor\_Livro**:
   * Cada livro pode ter muitos autores e, inversamente, cada autor pode escrever muitos livros. Isso é representado na tabela associativa **Autor\_Livro**.

**Benefícios da Normalização e Modelagem ER**

* **Redução de Redundância**: A normalização ajuda a reduzir a redundância de dados. Por exemplo, as informações sobre autores e editoras são armazenadas apenas uma vez.
* **Integridade dos Dados**: A modelagem ER ajuda a manter a integridade dos dados, garantindo que as relações sejam mantidas corretamente.
* **Facilidade de Manutenção**: As tabelas menores e bem definidas são mais fáceis de manter e atualizar.
* **Consultas Eficientes**: A estrutura relacional permite consultas eficientes usando SQL.

**Conclusão**

Com a modelagem entidade-relacionamento e a normalização das tabelas, podemos criar um banco de dados relacional eficiente e organizado. O diagrama ER serve como um guia visual para entender as conexões entre as diferentes entidades, facilitando a implementação e a manutenção do banco de dados.

**Python**

Para criar um programa em Python que permita a inserção de dados em uma base de dados MySQL, você pode usar a biblioteca mysql-connector-python para se conectar ao MySQL e manipular o banco de dados. Vou fornecer um exemplo de como criar um formulário de entrada de dados para inserir um novo livro, juntamente com o código para criar as tabelas no MySQL.

**1. Configuração do Banco de Dados MySQL**

Primeiro, vamos criar as tabelas necessárias no MySQL.

CREATE DATABASE biblioteca;

USE biblioteca;

CREATE TABLE Editora (

ID\_Editora INT AUTO\_INCREMENT PRIMARY KEY,

Nome VARCHAR(100) NOT NULL

);

CREATE TABLE Autor (

ID\_Autor INT AUTO\_INCREMENT PRIMARY KEY,

Nome VARCHAR(100) NOT NULL

);

CREATE TABLE Livro (

ID\_Livro INT AUTO\_INCREMENT PRIMARY KEY,

Título VARCHAR(255) NOT NULL,

ID\_Editora INT,

Data\_Aquisição DATE,

Estado\_Livro VARCHAR(50),

Lido BOOLEAN,

Observação TEXT,

FOREIGN KEY (ID\_Editora) REFERENCES Editora(ID\_Editora)

);

CREATE TABLE Autor\_Livro (

ID\_Autor INT,

ID\_Livro INT,

PRIMARY KEY (ID\_Autor, ID\_Livro),

FOREIGN KEY (ID\_Autor) REFERENCES Autor(ID\_Autor),

FOREIGN KEY (ID\_Livro) REFERENCES Livro(ID\_Livro)

);

**2. Programa em Python para Inserção de Dados**

Aqui está um exemplo de programa em Python para inserir um livro na base de dados MySQL.

**Instalação da biblioteca mysql-connector-python**

Você pode instalar a biblioteca usando o pip:

pip install mysql-connector-python

**Código em Python**

import mysql.connector

from mysql.connector import Error

def inserir\_dados():

try:

# Conectar ao banco de dados

conexao = mysql.connector.connect(

host='localhost',

database='biblioteca',

user='seu\_usuario',

password='sua\_senha'

)

if conexao.is\_connected():

cursor = conexao.cursor()

# Entrada de dados para a tabela Livro

titulo = input("Digite o título do livro: ")

id\_editora = int(input("Digite o ID da editora: "))

data\_aquisicao = input("Digite a data de aquisição (AAAA-MM-DD): ")

estado\_livro = input("Digite o estado do livro (Novo/Usado): ")

lido = input("O livro foi lido? (Sim/Não): ").lower() == 'sim'

observacao = input("Digite alguma observação: ")

# Inserção na tabela Livro

livro\_sql = """INSERT INTO Livro (Título, ID\_Editora, Data\_Aquisição, Estado\_Livro, Lido, Observação)

VALUES (%s, %s, %s, %s, %s, %s)"""

livro\_dados = (titulo, id\_editora, data\_aquisicao, estado\_livro, lido, observacao)

cursor.execute(livro\_sql, livro\_dados)

conexao.commit()

id\_livro = cursor.lastrowid

# Entrada de dados para a tabela Autor\_Livro

while True:

id\_autor = int(input("Digite o ID do autor (ou 0 para terminar): "))

if id\_autor == 0:

break

autor\_livro\_sql = """INSERT INTO Autor\_Livro (ID\_Autor, ID\_Livro)

VALUES (%s, %s)"""

autor\_livro\_dados = (id\_autor, id\_livro)

cursor.execute(autor\_livro\_sql, autor\_livro\_dados)

conexao.commit()

print("Livro inserido com sucesso!")

except Error as e:

print("Erro ao conectar ao MySQL", e)

finally:

if conexao.is\_connected():

cursor.close()

conexao.close()

print("Conexão ao MySQL encerrada")

if \_\_name\_\_ == "\_\_main\_\_":

inserir\_dados()

**Explicação do Código**

1. **Conexão ao Banco de Dados**:
   * O código se conecta ao banco de dados MySQL usando a biblioteca mysql.connector.
   * Substitua 'seu\_usuario' e 'sua\_senha' pelas credenciais do seu banco de dados.
2. **Entrada de Dados**:
   * O usuário insere os dados do livro, como título, ID da editora, data de aquisição, estado do livro, se foi lido e observações.
   * O programa insere esses dados na tabela Livro.
3. **Inserção de Relacionamentos Autor\_Livro**:
   * O usuário pode inserir múltiplos autores para o livro inserido.
   * O programa insere cada autor na tabela associativa Autor\_Livro, associando o autor ao livro recém-criado.
4. **Fechamento da Conexão**:
   * A conexão com o banco de dados é fechada após a inserção dos dados.

**Dicas finais:** Este programa em Python, juntamente com a estrutura de tabelas no MySQL, permite a inserção eficiente de novos livros na base de dados, mantendo a integridade relacional e a normalização dos dados. A implementação pode ser expandida para incluir outras operações, como atualização, exclusão e consulta de dados.