

Documentação - Sistema de Empréstimos de Biblioteca

Descrição Geral

Este script SQL cria e organiza um sistema de gerenciamento de empréstimos de livros em uma biblioteca, com três principais entidades:

- **Membros:** Pessoas cadastradas na biblioteca que podem pegar livros emprestados.
- **Livros:** Obras disponíveis para empréstimo.
- **Empréstimos:** Registros de quando um livro foi emprestado a um membro e suas respectivas datas de devolução.

Estrutura do Script

1. Configurações Iniciais

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;  
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;  
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

Descrição:

- Desativamos temporariamente as verificações de chaves únicas (`UNIQUE_CHECKS`) e chaves estrangeiras (`FOREIGN_KEY_CHECKS`) para evitar conflitos durante a criação das tabelas.
- Alteramos o `SQL_MODE` para facilitar o processo de criação sem restrições que poderiam bloquear a execução de alguns comandos.

2. Criação do Schema (Banco de Dados)

```
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8  
USE `mydb` ;
```

Descrição:

- O comando `CREATE SCHEMA` cria o banco de dados `mydb` com a codificação UTF-8, que permite armazenar uma grande variedade de caracteres.
 - O comando `USE` define `mydb` como o banco de dados que será utilizado para a criação das tabelas subsequentes.
-

3. Criação da Tabela `membro`

```
CREATE TABLE IF NOT EXISTS `mydb`.`membro` (  
  `id_domembro` INT NOT NULL,  
  `nome` VARCHAR(45) NULL,  
  `endereco` VARCHAR(45) NULL,  
  `telefone` INT NOT NULL,  
  `data_deassociacao` DATE NULL,  
  PRIMARY KEY (`id_domembro`)  
)  
ENGINE = InnoDB;
```

Descrição:

- **Tabela `membro`:** Armazena os dados dos membros cadastrados na biblioteca.
 - `id_domembro`: Identificador único do membro, definido como chave primária (`PRIMARY KEY`).
 - `nome`: Nome do membro.
 - `endereco`: Endereço do membro.
 - `telefone`: Número de telefone do membro.
 - `data_deassociacao`: Data em que o membro se associou à biblioteca.
 - **Motor de Armazenamento:** `InnoDB` é usado para garantir a integridade transacional e a integridade referencial com suporte a chaves estrangeiras.
-

4. Criação da Tabela `livro`

```
CREATE TABLE IF NOT EXISTS `mydb`.`livro` (  
  `id_dolivro` INT NOT NULL,
```

```

`titulo` VARCHAR(90) NULL,
`autor` VARCHAR(45) NULL,
`ano_publicado` DATE NULL,
`genero` VARCHAR(45) NULL,
PRIMARY KEY (`id_dolivro`)
)
ENGINE = InnoDB;

```

Descrição:

- **Tabela `livro`**: Armazena as informações dos livros da biblioteca.
 - `id_dolivro`: Identificador único de cada livro, definido como chave primária (`PRIMARY KEY`).
 - `titulo`: Título do livro.
 - `autor`: Nome do autor do livro.
 - `ano_publicado`: Ano de publicação do livro.
 - `genero`: Gênero literário do livro (por exemplo, ficção, biografia, etc.).
- **Motor de Armazenamento**: Novamente, usamos `InnoDB` para suportar as futuras integrações com outras tabelas.

5. Criação da Tabela `emprestimo`

```

CREATE TABLE IF NOT EXISTS `mydb`.`emprestimo` (
  `id_doemprestimo` INT NOT NULL,
  `id_dolivro` INT NOT NULL,
  `id_domembro` INT NOT NULL,
  `data_deemprestimo` DATE NULL,
  `data_dedevolucao` DATE NULL,
  PRIMARY KEY (`id_doemprestimo`),
  FOREIGN KEY (`id_dolivro`) REFERENCES `mydb`.`livro` (`id_dolivro`),
  FOREIGN KEY (`id_domembro`) REFERENCES `mydb`.`membro` (`id_domembro`)
)

```

```
)  
ENGINE = InnoDB;
```

Descrição:

- **Tabela `emprestimo`** : Registra cada empréstimo realizado pelos membros.
 - `id_doemprestimo` : Identificador único de cada empréstimo (chave primária).
 - `id_dolivro` : Referência ao livro emprestado, que é uma chave estrangeira que aponta para a tabela `livro` .
 - `id_domembro` : Referência ao membro que tomou o livro emprestado, que é uma chave estrangeira que aponta para a tabela `membro` .
 - `data_deemprestimo` : Data em que o empréstimo foi feito.
 - `data_dedevolucao` : Data prevista para a devolução do livro.
- **Relacionamentos:**
 - A chave estrangeira (`FOREIGN KEY`) `id_dolivro` referencia o identificador único da tabela `livro` .
 - A chave estrangeira `id_domembro` referencia o identificador único da tabela `membro` .

6. Restauração das Configurações Iniciais

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Descrição:

- Após a criação das tabelas, as verificações de unicidade (`UNIQUE_CHECKS`), integridade referencial (`FOREIGN_KEY_CHECKS`), e o modo SQL original são restaurados ao estado anterior para garantir a validação de futuras operações.

1. Relacionamento entre `membro` e `emprestimo`

- **Cardinalidade: 1:N** (Um membro pode realizar vários empréstimos, mas um empréstimo está associado a apenas um membro).

- **Descrição:**

- Um **membro** da biblioteca pode realizar múltiplos empréstimos de livros ao longo do tempo. Assim, a tabela `membro` tem uma relação de um para muitos com a tabela `emprestimo`.
- A tabela `emprestimo` contém o campo `id_domembro`, que é uma chave estrangeira referenciando o `id_domembro` na tabela `membro`. Isso significa que cada registro em `emprestimo` está vinculado a um único membro.

- **Representação:**

- `membro` (1) ——— (N) `emprestimo`
-

2. Relacionamento entre `livro` e `emprestimo`

- **Cardinalidade: 1:N** (Um livro pode ser emprestado várias vezes, mas um empréstimo se refere a um único livro).

- **Descrição:**

- Um **livro** pode ser emprestado várias vezes, mas cada registro na tabela `emprestimo` refere-se a um único livro em um dado momento. Isso cria uma relação de um para muitos entre a tabela `livro` e a tabela `emprestimo`.
- A tabela `emprestimo` contém o campo `id_dolivro`, que é uma chave estrangeira referenciando o `id_dolivro` na tabela `livro`. Isso significa que cada registro em `emprestimo` está vinculado a um único livro.

- **Representação:**

- `livro` (1) ——— (N) `emprestimo`
-

3. Relacionamento entre `emprestimo`, `livro`, e `membro`

- **Descrição:**

- A tabela `emprestimo` atua como uma tabela de junção, conectando as tabelas `livro` e `membro`. Para cada empréstimo, há um membro

específico (quem está tomando o livro emprestado) e um livro específico (o item sendo emprestado).

- Assim, **emprestimo** forma um relacionamento **N:M** (muitos para muitos) entre **membros** e **livros**, através de dois relacionamentos de cardinalidade 1:N.
- **Representação Completa:**



Resumo das Cardinalidades

- **membro-emprestimo:** Um membro pode realizar vários empréstimos, mas um empréstimo é realizado por apenas um membro (**1:N**).
- **livro-emprestimo:** Um livro pode ser emprestado várias vezes, mas um empréstimo se refere a apenas um livro (**1:N**).