

Aula: Chaves Primárias, Chaves Estrangeiras, Identificadores Únicos e Restrições (Constraints) no MySQL

Objetivos:

1. Compreender o conceito e a importância das chaves primárias e estrangeiras.
2. Aprender a definir e utilizar identificadores únicos e restrições em tabelas MySQL.
3. Aplicar conceitos de integridade referencial e regras de integridade de dados em bancos de dados relacionais.

1. Introdução às Chaves

1.1 Chave Primária (Primary Key)

- **Definição:** A chave primária é um identificador exclusivo para cada linha de uma tabela. Garante que cada registro seja único e não nulo.
- **Características:**
 - Única para cada linha.
 - Não pode conter valores nulos.
 - Pode ser composta por uma ou mais colunas (chave primária composta).

Exemplo:

sql

Copiar código

```
CREATE TABLE clientes (  
    id INT AUTO_INCREMENT,  
    nome VARCHAR(100),  
    email VARCHAR(100),  
    PRIMARY KEY (id)  
);
```

1.2 Chave Estrangeira (Foreign Key)

- **Definição:** A chave estrangeira é uma coluna ou um conjunto de colunas que estabelece um relacionamento com a chave primária de outra tabela. Garante a integridade referencial entre tabelas.
- **Características:**
 - Cria um vínculo entre tabelas.
 - Pode ser usada para garantir que o valor em uma tabela exista em outra tabela.
 - Pode ser usada para definir ações de atualização e exclusão em cascata.

Exemplo:

sql

Copiar código

```
CREATE TABLE pedidos (  
    id INT AUTO_INCREMENT,  
    cliente_id INT,  
    data_pedido DATE,  
    PRIMARY KEY (id),  
    FOREIGN KEY (cliente_id) REFERENCES clientes(id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

2. Identificadores Únicos e Restrições

2.1 Identificador Único (Unique Key)

- **Definição:** A restrição de identificador único assegura que todos os valores em uma coluna sejam diferentes. Diferente da chave primária, pode permitir valores nulos (dependendo da versão do MySQL).
- **Características:**
 - Garante a unicidade dos valores em uma ou mais colunas.
 - Pode haver múltiplas restrições **UNIQUE** em uma tabela.

Exemplo:

sql

Copiar código

```
CREATE TABLE usuarios (  
    id INT AUTO_INCREMENT,  
    username VARCHAR(50),  
    email VARCHAR(100),  
    PRIMARY KEY (id),  
    UNIQUE (username),  
    UNIQUE (email)  
);
```

2.2 Restrições (Constraints)

- **Definição:** Restrições são regras aplicadas às colunas de uma tabela para garantir a integridade dos dados. Incluem restrições de chave primária, chave estrangeira, identificador único, NOT NULL e CHECK.

Restrições Comuns:

NOT NULL: Garante que uma coluna não possa ter valores nulos.

sql

Copiar código

```
CREATE TABLE produtos (  
    id INT AUTO_INCREMENT,  
    nome VARCHAR(100) NOT NULL,  
    preco DECIMAL(10,2) NOT NULL,  
    PRIMARY KEY (id)  
);
```

•

CHECK: Garante que os valores em uma coluna atendam a uma condição específica (disponível a partir do MySQL 8.0.16).

sql

Copiar código

```
CREATE TABLE funcionarios (  
    id INT AUTO_INCREMENT,  
    nome VARCHAR(100),  
    salario DECIMAL(10,2),  
    CHECK (salario > 0),  
    PRIMARY KEY (id)  
);
```

•

3. Exercícios Práticos

Exercício 1: Criar Tabelas com Restrições

1. Crie uma tabela **produtos** com as seguintes colunas e restrições:
 - **id** (INT, auto incremento, chave primária)
 - **nome** (VARCHAR(100), não nulo)
 - **preco** (DECIMAL(10,2), não nulo)
 - **categoria_id** (INT, chave estrangeira referenciando a tabela **categorias**)

sql

Copiar código

```
CREATE TABLE categorias (  
    id INT AUTO_INCREMENT,  
    nome VARCHAR(100),  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE produtos (  
    id INT AUTO_INCREMENT,
```

```
    nome VARCHAR(100) NOT NULL,  
    preco DECIMAL(10,2) NOT NULL,  
    categoria_id INT,  
    PRIMARY KEY (id),  
    FOREIGN KEY (categoria_id) REFERENCES categorias(id)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE  
);
```

2.

Exercício 2: Adicionar e Alterar Restrições

1. Adicione uma restrição de identificador único à coluna `email` na tabela `usuarios`.

Modifique a tabela `funcionários` para adicionar uma restrição `CHECK` que garanta que o salário seja maior que 0.

sql

Copiar código

```
ALTER TABLE usuarios  
ADD UNIQUE (email);
```

```
ALTER TABLE funcionários  
ADD CONSTRAINT chk_salario CHECK (salario > 0);
```

2.

Exercício 3: Analisar e Corrigir Violação de Integridade

1. Crie um cenário onde uma violação de integridade referencial pode ocorrer e, em seguida, corrija o problema.
2. Descreva a importância de manter a integridade referencial e como as restrições ajudam a garantir a qualidade dos dados.

Conclusão

- **Revisão dos Conceitos:** As chaves primárias e estrangeiras são essenciais para a integridade e relacionamento entre tabelas. Identificadores únicos e restrições adicionais garantem a validade dos dados.
- **Importância das Restrições:** Garantem a integridade e a consistência dos dados, prevenindo erros e mantendo a qualidade do banco de dados.

Referências

- Documentação MySQL - Chaves Primárias e Estrangeiras
- Documentação MySQL - Restrições e Integridade
- Tutorial MySQL - Integridade Referencial

Perguntas e Discussão

- Dúvidas sobre a implementação de chaves e restrições no MySQL?
- Experiências práticas e desafios encontrados ao trabalhar com integridade de dados?