

# SQL

## Exclusão de tabelas em banco de dados

Você sabia que seu material didático é interativo e multimídia? Isso significa que você pode interagir com o conteúdo de diversas formas, a qualquer hora e lugar. Na versão impressa, porém, alguns conteúdos interativos ficam desabilitados. Por essa razão, fique atento: sempre que possível, opte pela versão digital. Bons estudos!

Nesta webaula você vai ver o processo de exclusão de tabelas, além do processo de restrições e de manutenção da integridade das tabelas contra alterações indevidas e/ou não autorizadas.

## Exclusão de tabelas

Toda tabela deve ter uma chave primária, que identifica um registro. Já uma chave estrangeira é a responsável por unir duas tabelas. Esta chave estrangeira é o campo que se refere a uma chave primária em outra tabela. É **importante** observar que a tabela que contém a chave estrangeira é chamada de tabela filha e a tabela que contém a chave candidata (seja ela primária ou não) é chamada de tabela pai ou referenciada ( [CARDOSO E CARDOSO, 2013](#) ).

## Restrição FOREIGN KEY

Assim, a restrição FOREIGN KEY é usada para impedir ações que destruam links entre tabelas, além de impedir que dados inválidos sejam inseridos na coluna de chave estrangeira, porque ela deve ser um dos valores contidos na tabela para a qual ela aponta.

Veja a seguir um exemplo de criação de tabelas para atribuição e cálculo de notas.

```
1  CREATE TABLE aluno (  
2      id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
3      nome CHAR (50) NOT NULL  
4  );  
5  CREATE TABLE curso (  
6      id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
7      nome CHAR (50) NOT NULL  
8  );  
9  CREATE TABLE nota (  
10     aluno_id INT NOT NULL,  
11     curso_id INT NOT NULL,  
12     dataavaliacao DATE NOT NULL,  
13     nota DOUBLE NOT NULL,  
14     PRIMARY KEY(aluno_id, curso_id, dataavaliacao),  
15     INDEX i2 (curso_id),  
16     FOREIGN KEY (aluno_id) REFERENCES aluno(id) ON DELETE  
17         CASCADE,  
18     FOREIGN KEY (curso_id) REFERENCES curso (id) ON DELETE  
19         RESTRICT  
20 );
```

Na estrutura, a tabela nota detém duas restrições, tendo como referência as tabelas:

- **Aluno** (linhas 1 a 4) - que contém as informações acerca dos alunos da instituição, e pode ser alterada para conter ainda mais dados de registro desse aluno, caso necessário.
- **Curso** (linhas 5 a 8) - que contém informações acerca do curso em que os alunos poderão estar matriculados.

Estas duas tabelas estão cedendo suas chaves primárias para se relacionar com a tabela “nota”.

O comando CREATE TABLE cria a tabela nomeada:

```
CREATE TABLE nome_tabela;
```

E o comando SHOW CREATE TABLE criada uma tabela de nome “nota”, com exibição dos resultados:

```
SHOW CREATE TABLE nota;
```

Resultado da instrução após execução

Table	Create Table
nota	CREATE TABLE `nota` ( `aluno_id` int(11) NOT NULL, `curso_id` int(11) NOT NULL, `dataavaliac

Fonte: elaborada pelo autor, captura de tela no software MySQL Workbench.

Na tela de resultado da instrução, ao clicar com o botão direito do mouse sobre a linha acima na opção COPY FIELD UNQUOTED teremos como resultado a instrução a seguir.

### Criação de uma tabela com restrições (*constraints*)

```
1 CREATE TABLE `nota` (
2     `aluno_id` INT (11) NOT NULL,
3     `curso_id` INT (11) NOT NULL,
4     `dataavaliacao` DATE NOT NULL,
5     `nota` DOUBLE NOT NULL
6     PRIMARY KEY(`aluno_id`, `curso_id`, `dataavaliacao`),
7     KEY `i2` (`curso_id`),
8     CONSTRAINT `nota_ibfk_1` FOREIGN KEY (`aluno_id` REFERENCES
9         `aluno`(`id`) ON DELETE CASCADE,
10    CONSTRAINT `nota_ibfk_2` FOREIGN KEY (`curso_id`) REFERENCES
11        `curso`(`id`)
12    ENGINE=InnoDB DEFAULT CHARSET=LATIN1
```

O conjunto de instruções entre as linhas 6 (inclusive) e 13 (inclusive) deve sempre ser utilizado para determinar as restrições utilizadas nas tabelas que poderão ser alteradas ou excluídas.

Analisando esta instrução podemos nos certificar que a tabela “nota” é a filha da tabela “aluno” e “curso”. A tentativa de exclusão da tabela aluno ou curso está restringida, ou seja, não será possível.

## Instrução DROP TABLE

Quando deseja-se remover uma ou mais tabelas, a instrução DROP TABLE deve ser utilizada. Sua sintaxe envolve mencionar as tabelas que devem sofrer a ação.

```
DROP TABLE [IF EXISTS] nome_tabela [, nome_tabela] ...
```

### ATENÇÃO

Esta instrução remove a definição da tabela e todos os dados da tabela.

## IF EXISTS

Se quaisquer tabelas nomeadas na lista de argumentos não existirem, a instrução falhará, com um erro indicando por nome quais tabelas não existentes não foram possíveis descartar, não sendo realizada nenhuma alteração. Existe uma maneira de evitar este erro: se você utilizar a cláusula IF EXISTS em vez de um erro, um alerta é gerado para cada tabela inexistente.

Isso também pode ser útil para eliminar tabelas em circunstâncias incomuns sob as quais há uma entrada no dicionário de dados. Porém, esta cláusula não deleta nenhuma tabela gerenciada pelo mecanismo de armazenamento.

Para utilizar a cláusula IF EXISTS no exemplo, averiguando a existência da tabela “aluno” antes de executar o comando DROP, utilize a instrução a seguir:

```
DROP TABLE IF EXISTS aluno;
```

Como resultado, aparecerá uma mensagem de alerta identificando que há uma restrição de chave estrangeira que impede esta exclusão.

Mas, vamos supor que essa ação de exclusão da tabela “aluno” fosse inevitável e, após sua análise esta tabela não fosse mais necessária em seu banco de dados. Para poder excluí-la é necessário excluir antes a restrição.

## Excluir restrição

Para excluir a restrição na tabela “nota” que faz referência à tabela “aluno” (pai):

```
ALTER TABLE nota DROP FOREIGN KEY nota_ibfk_1;
```

Após esta instrução, é possível executar a exclusão com sucesso.

É fundamental reforçar a importância da inclusão de restrições adequadas para evitar erros graves quando da execução do comando DROP TABLE. A exclusão de tabelas implica na exclusão de todos os dados. A instrução DROP TABLE não deve ser confundida com a instrução que apagará todo o conteúdo de uma tabela, ou seja, esvaziar sua tabela.

# Instrução TRUNCATE TABLE

A instrução para esvaziar uma tabela completamente é a TRUNCATE TABLE e sua sintaxe é:

```
TRUNCATE [TABLE] nome_tabela;
```

Logicamente, TRUNCATE TABLE é semelhante a uma instrução DELETE (que exclui todas as linhas) ou uma sequência de instruções DROP TABLE e CREATE TABLE.

- A instrução TRUNCATE TABLE falha para uma tabela se houver alguma restrição FOREIGN KEY de outras tabelas que a referenciam, Já restrições de chaves estrangeiras entre colunas da mesma tabela são permitidas. Contando que a definição da tabela seja válida, a tabela pode ser recriada como uma tabela vazia com TRUNCATE TABLE, mesmo que os dados ou arquivos de índice tenham sido corrompidos.
- O campo AUTO\_INCREMENT é o campo que vai automaticamente sendo incrementado em cada registro, funcionando naturalmente como chave primária, caso o usuário assim desejar. Qualquer valor AUTO\_INCREMENT é redefinido para seu valor inicial.

## SAIBA MAIS

**Existe também um recurso que deve ser utilizado com cuidado!**

Podemos instruir ao MySQL que em determinadas circunstâncias especiais que mesmo que exista restrições em seu banco de dados, estas sejam ignoradas:

```
SET FOREIGN_KEY_CHECKS = 1;
```

Ao executar esta instrução o MySQL irá verificar antes da exclusão quais são as restrições impostas na estrutura do banco de dados.

- Se definido como 1 (o padrão), as restrições de chave estrangeira para tabelas são verificadas.
- Se definido como 0, as restrições de chave estrangeira serão ignoradas, com algumas exceções.

Ao recriar uma tabela que foi descartada, um erro será retornado se a definição da tabela não estiver de acordo com as restrições de chave estrangeira que fazem referência à tabela. Da mesma forma, uma operação ALTER TABLE retornará um erro se uma definição de chave estrangeira for formada incorretamente.

## ATENÇÃO

Normalmente, essa configuração fica ativada durante a operação normal para impor a integridade referencial.

No exemplo, uma eventual tentativa de excluir a tabela “curso” não seria bem-sucedida, pois existe uma restrição na tabela “nota” fazendo sua referência e, assim, impedindo sua exclusão. Mas, ao executar a instrução a seguir, o MySQL será instruído a ignorar quaisquer restrições que estejam criadas.

```
SET FOREIGN_KEY_CHECKS = 0;
```

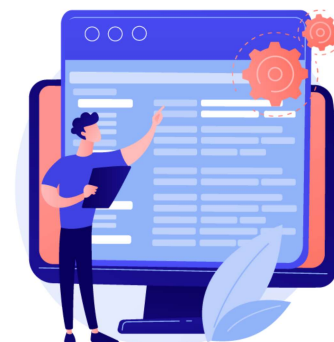
Posteriormente, ao executar a instrução a seguir, a exclusão da tabela “curso” ocorrerá com sucesso.

```
DROP TABLE IF EXISTS curso;
```

Então, após isso, pode-se fazer com que o MySQL volte ao seu estado padrão, utilizando todas as restrições existentes para mudanças de estrutura, ao executar a instrução:

```
SET FOREIGN_KEY_CHECKS = 1;
```

Aqui, você viu como é processo de exclusão de tabelas e como os comandos de deleção devem ser cuidadosamente compostos, sob pena de gerarem problemas enormes para o banco de dados.



Fonte: Shutterstock.

Para visualizar o vídeo, acesse seu material digital.