

MySQL: Introdução ALTER TABLE

Este tutorial do MySQL explica como usar a instrução ALTER TABLE do MySQL, para adicionar uma coluna, modificar uma coluna, excluir uma coluna, renomear uma coluna ou renomear uma tabela (com sintaxe e exemplos).

Descrição

O comando ALTER TABLE do MySQL é usado para adicionar, modificar ou eliminar / excluir colunas em uma tabela. O comando ALTER TABLE do MySQL também é usado para renomear uma tabela.

Adicionar uma coluna na tabela

Sintaxe

```
ALTER TABLE table_name  
  ADD new_column_name column_definition  
  [ FIRST | AFTER column_name ];
```

table_name

O nome da tabela a ser modificada.

new_column_name

O nome da nova coluna a ser adicionada à tabela.

column_definition

O tipo de dados e a definição da coluna (NULL ou NOT NULL etc).

FIRST | AFTER column_name

Opcional. Ele diz ao MySQL onde criar a coluna na tabela. Se este parâmetro não for especificado, a nova coluna será adicionada ao final da tabela.

Exemplo

```
ALTER TABLE contacts  
  ADD last_name VARCHAR(40) NOT NULL  
  AFTER contact_id;
```

Este exemplo ALTER TABLE do MySQL adicionará uma coluna chamada last_name à tabela de contatos. Ele será criado como uma coluna NOT NULL e aparecerá após o campo contact_id na tabela.

Adicionar várias colunas na tabela

Sintaxe

```
ALTER TABLE table_name  
  ADD new_column_name column_definition  
    [ FIRST | AFTER column_name ],  
  ADD new_column_name column_definition  
    [ FIRST | AFTER column_name ],  
  ...  
;
```

table_name

O nome da tabela a ser modificada.

new_column_name

O nome da nova coluna a ser adicionada à tabela.

column_definition

O tipo de dados e a definição da coluna (NULL ou NOT NULL etc).

FIRST | AFTER column_name

Opcional. Ele diz ao MySQL onde criar a coluna na tabela. Se este parâmetro não for especificado, a nova coluna será adicionada ao final da tabela.

Exemplo

```
ALTER TABLE contacts  
  ADD last_name varchar(40) NOT NULL  
    AFTER contact_id,  
  ADD first_name varchar(35) NULL  
    AFTER last_name;
```

Este exemplo ALTER TABLE irá adicionar duas colunas à tabela de contatos

- *last_name* e *first_name*.

O campo last_name será criado como uma coluna VARCHAR(40) NOT NULL e aparecerá após a coluna contact_id na tabela. A coluna first_name será criada como uma coluna VARCHAR(35) NULL e aparecerá após a coluna last_name na tabela.

Modificar coluna na tabela

Sintaxe

```
ALTER TABLE table_name  
  MODIFY column_name column_definition  
  [ FIRST | AFTER column_name ];
```

table_name

O nome da tabela a ser modificada.

column_name

O nome da coluna a ser modificada na tabela.

column_definition

O tipo de dados e a definição da coluna (NULL ou NOT NULL etc).

FIRST | AFTER column_name

Opcional. Ele diz ao MySQL onde criar a coluna na tabela. Se este parâmetro não for especificado, a nova coluna será adicionada ao final da tabela.

Exemplo

```
ALTER TABLE contacts  
  MODIFY last_name varchar(50) NULL;
```

Este exemplo ALTER TABLE modificará a coluna chamada last_name para ser um tipo de dados de VARCHAR(50) e forçará a coluna a permitir valores NULL.

Modificar várias colunas na tabela

Sintaxe

```
ALTER TABLE table_name  
  MODIFY column_name column_definition  
  [ FIRST | AFTER column_name ],  
  MODIFY column_name column_definition  
  [ FIRST | AFTER column_name ],  
  ...  
  ;
```

table_name

O nome da tabela a ser modificada.

column_name

O nome da coluna a ser modificada na tabela.

column_definition

O tipo de dados e a definição da coluna (NULL ou NOT NULL etc).

FIRST | AFTER column_name

Opcional. Ele diz ao MySQL onde criar a coluna na tabela. Se este parâmetro não for especificado, a nova coluna será adicionada ao final da tabela.

Exemplo.

```
ALTER TABLE contacts  
  MODIFY last_name varchar(55) NULL  
    AFTER contact_type,  
  MODIFY first_name varchar(30) NOT NULL;
```

Este exemplo ALTER TABLE irá modificar duas colunas para a tabela de contatos

- *last_name* e *first_name*.

O campo last_name será alterado para uma coluna varchar (55) NULL e aparecerá após a coluna contact_type na tabela. A coluna first_name será modificada para uma coluna VARCHAR(30) NOT NULL (e não mudará a posição na definição da tabela de contatos, pois não há FIRST | AFTER especificado).

Excluir uma coluna na tabela

Sintaxe

```
ALTER TABLE table_name  
  DROP COLUMN column_name;
```

table_name

O nome da tabela a ser modificada.

column_name

O nome da coluna a ser excluída da tabela.

Exemplo

```
ALTER TABLE contacts  
DROP COLUMN contact_type;
```

Este exemplo ALTER TABLE excluirá a coluna chamada contact_type da tabela chamada contacts.

Renomear coluna na tabela

Sintaxe

```
ALTER TABLE table_name  
CHANGE COLUMN old_name new_name  
column_definition  
[ FIRST | AFTER column_name ]
```

table_name

O nome da tabela a ser modificada.

old_name

A coluna a ser renomeada.

new_name

O novo nome da coluna.

column_definition

O tipo de dados e a definição da coluna (NULL ou NOT NULL etc).

FIRST | AFTER column_name

Opcional. Ele diz ao MySQL onde criar a coluna na tabela. Se este parâmetro não for especificado, a nova coluna será adicionada ao final da tabela.

Exemplo

```
ALTER TABLE contacts  
CHANGE COLUMN contact_type ctype  
varchar(20) NOT NULL;
```

Este exemplo ALTER TABLE do MySQL renomeará a coluna chamada contact_type para ctype. A coluna será definida como uma coluna varchar (20) NOT NULL.

Renomear tabela

Sintaxe

```
ALTER TABLE table_name  
  RENAME TO new_table_name;
```

table_name

A tabela a ser renomeada.

new_table_name

O novo nome da tabela a ser usado.

```
ALTER TABLE contacts  
  RENAME TO people;
```

Este exemplo ALTER TABLE renomeará a tabela de contatos para pessoas.