**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.