

MySQL ALTER命令

当我们需要修改数据表名或者修改数据表字段时，就需要使用到MySQL ALTER命令。

开始本章教程前让我们先创建一张表，表名为：testalter_tbl。

```
root@host# mysql -u root -p password;
Enter password:*****
mysql> use RUNOOB;
Database changed
mysql> create table testalter_tbl
-> (
-> i INT,
-> c CHAR(1)
-> );
Query OK, 0 rows affected (0.05 sec)
mysql> SHOW COLUMNS FROM testalter_tbl;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| i     | int(11) | YES  |     | NULL    |       |
| c     | char(1) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

删除，添加或修改表字段

如下命令使用了 ALTER 命令及 DROP 子句来删除以上创建表的 i 字段：

```
mysql> ALTER TABLE testalter_tbl DROP i;
```

如果数据表中只剩余一个字段则无法使用DROP来删除字段。

MySQL 中使用 ADD 子句来向数据表中添加列，如下实例在表 testalter_tbl 中添加 i 字段，并定义数据类型：

```
mysql> ALTER TABLE testalter_tbl ADD i INT;
```

执行以上命令后，i 字段会自动添加到数据表字段的末尾。

```
mysql> SHOW COLUMNS FROM testalter_tbl;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
```

```
| c      | char(1) | YES  |      | NULL  |      |
| i      | int(11) | YES  |      | NULL  |      |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

如果你需要指定新增字段的位置，可以使用MySQL提供的关键字 FIRST (设定位第一列)，AFTER 字段名（设定位于某个字段之后）。

尝试以下 ALTER TABLE 语句, 在执行成功后，使用 SHOW COLUMNS 查看表结构的变化：

```
ALTER TABLE testalter_tbl DROP i;
ALTER TABLE testalter_tbl ADD i INT FIRST;
ALTER TABLE testalter_tbl DROP i;
ALTER TABLE testalter_tbl ADD i INT AFTER c;
```

FIRST 和 AFTER 关键字可用于 ADD 与 MODIFY 子句，所以如果你想重置数据表字段的位置就需要先使用 DROP 删除字段然后使用 ADD 来添加字段并设置位置。

修改字段类型及名称

如果需要修改字段类型及名称, 你可以在ALTER命令中使用 MODIFY 或 CHANGE 子句。

例如，把字段 c 的类型从 CHAR(1) 改为 CHAR(10)，可以执行以下命令：

```
mysql> ALTER TABLE testalter_tbl MODIFY c CHAR(10);
```

使用 CHANGE 子句, 语法有很大的不同。在 CHANGE 关键字之后，紧跟着的是你要修改的字段名，然后指定新字段名及类型。尝试如下实例：

```
mysql> ALTER TABLE testalter_tbl CHANGE i j BIGINT;
```

```
mysql> ALTER TABLE testalter_tbl CHANGE j j INT;
```

ALTER TABLE 对 Null 值和默认值的影响

当你修改字段时，你可以指定是否包含值或者是否设置默认值。

以下实例，指定字段 j 为 NOT NULL 且默认值为100。

```
mysql> ALTER TABLE testalter_tbl
-> MODIFY j BIGINT NOT NULL DEFAULT 100;
```

如果你不设置默认值，MySQL会自动设置该字段默认为 NULL。

修改字段默认值

你可以使用 ALTER 来修改字段的默认值，尝试以下实例：

```
mysql> ALTER TABLE testalter_tbl ALTER i SET DEFAULT 1000;
mysql> SHOW COLUMNS FROM testalter_tbl;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| c     | char(1) | YES  |     | NULL    |       |
| i     | int(11) | YES  |     | 1000    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

你也可以使用 ALTER 命令及 DROP子句来删除字段的默认值，如下实例：

```
mysql> ALTER TABLE testalter_tbl ALTER i DROP DEFAULT;
mysql> SHOW COLUMNS FROM testalter_tbl;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| c     | char(1) | YES  |     | NULL    |       |
| i     | int(11) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
Changing a Table Type:
```

修改数据表类型，可以使用 ALTER 命令及 TYPE 子句来完成。尝试以下实例，我们将表 testalter_tbl 的类型修改为 MYISAM：

注意：查看数据表类型可以使用 SHOW TABLE STATUS 语句。

```
mysql> ALTER TABLE testalter_tbl ENGINE = MYISAM;
mysql> SHOW TABLE STATUS LIKE 'testalter_tbl'\G
***** 1. row *****
      Name: testalter_tbl
      Type: MyISAM
  Row_format: Fixed
        Rows: 0
  Avg_row_length: 0
   Data_length: 0
Max_data_length: 25769803775
   Index_length: 1024
      Data_free: 0
  Auto_increment: NULL
   Create_time: 2007-06-03 08:04:36
   Update_time: 2007-06-03 08:04:36
```

```
Check_time: NULL
Create_options:
Comment:
1 row in set (0.00 sec)
```

修改表名

如果需要修改数据表的名称，可以在 ALTER TABLE 语句中使用 RENAME 子句来实现。

尝试以下实例将数据表 testalter_tbl 重命名为 alter_tbl：

```
mysql> ALTER TABLE testalter_tbl RENAME TO alter_tbl;
```

ALTER 命令还可以用来创建及删除MySQL数据表的索引，该功能我们会在接下来的章节中介绍。

[← MySQL 事务](#)[MySQL 索引 →](#)**1 篇笔记****写笔记**

alter其他用途：

修改存储引擎：修改为myisam

```
alter table tableName engine=myisam;
```

删除外键约束：keyName是外键别名

```
alter table tableName drop foreign key keyName;
```

修改字段的相对位置：这里name1为想要修改的字段，type1为该字段原来类型，first和after二选一，这应该显而易见，first放在第一位，after放在name2字段后面

```
alter table tableName modify name1 type1 first|after name2;
```

Vladimir 2年前 (2017-07-17)