
修改表

修改数据表的前提是数据库中已经存在该表。修改表指的是修改数据库中已经存在的数据表的结构。修改数据表的操作也是数据库管理中必不可少的，在 MySQL 中可以使用 ALTER TABLE 语句来改变原有表的结构，例如增加或删除列、更改原有列类型、重新命名列或表等。

其语法格式如下：

```
ALTER TABLE <表名> [修改选项]
```

修改选项的语法格式如下：

```
{ ADD COLUMN <列名> <类型>
```

```
| CHANGE COLUMN <旧列名> <新列名> <新列类型>
```

```
| ALTER COLUMN <列名> { SET DEFAULT <默认值> | DROP DEFAULT }
```

```
| MODIFY COLUMN <列名> <类型>
```

```
| DROP COLUMN <列名>
```

```
| RENAME TO <新表名>
```

```
| CHARACTER SET <字符集名>
```

```
| COLLATE <校对规则名> }
```

```
| ENGINE={ INNODB | MYISAM | MEMOERY }
```

4.4.1 修改表名

MySQL 通过 ALTER TABLE 语句来实现表名的修改，语法规则如下：

```
ALTER TABLE <旧表名> RENAME [TO] <新表名>;
```

其中，TO 为可选参数，使用与否均不影响结果。

【例 4-13】修改 book 表的表名为 bookNew，在命令行窗口中输入如下命令。

```
ALTER TABLE book RENAME bookNew;
```

4.4.2 修改表字符集

MySQL 通过 ALTER TABLE 语句来实现表字符集的修改，语法规则如下：

```
ALTER TABLE <表名> [DEFAULT] CHARACTER SET <字符集名> [DEFAULT] COLLATE <校对规则名>;
```

其中，DEFAULT 为可选参数，使用与否均不影响结果。

【例 4-14】将 reader 表的字符集改成“gb2312”，在命令行窗口中输入如下命令。

```
ALTER TABLE reader CHARACTER SET gb2312;
```

4.4.3 增加新字段

MySQL 数据表是由行和列构成的，通常把表的“列”称为字段(Field)，把表的“行”称为记录(Record)。随着业务的变化，可能需要在已有的表中添加新的字段。

MySQL 允许在开头、中间和结尾处添加字段。

1) 在末尾添加字段

一个完整的字段包括字段名、数据类型和约束条件。MySQL 添加字段的语法格式如下：

```
ALTER TABLE <表名> ADD <新字段名><数据类型>[约束条件];
```

对语法格式的说明如下：

-
- <表名> 为数据表的名字;
 - <新字段名> 为所要添加的字段的名字;
 - <数据类型> 为所要添加的字段能存储数据的数据类型;
 - [约束条件] 是可选的, 用来对添加的字段进行约束。

这种语法格式默认在表的最后位置(最后一列的后面)添加新字段。

2) 在开头添加字段

MySQL 默认在表的最后位置添加新字段, 如果希望在开头位置(第一列的前面)添加新字段, 那么可以使用 FIRST 关键字, 语法格式如下:

```
ALTER TABLE <表名> ADD <新字段名> <数据类型> [约束条件] FIRST;
```

FIRST 关键字一般放在语句的末尾。

3) 在中间位置添加字段

MySQL 除了允许在表的开头位置和末尾位置添加字段外, 还允许在中间位置(指定的字段之后)添加字段, 此时需要使用 AFTER 关键字, 语法格式如下:

```
ALTER TABLE <表名> ADD <新字段名> <数据类型> [约束条件] AFTER <已经存在的字段名>;
```

AFTER 的作用是将新字段添加到某个已有字段后面。

【例 4-15】在 reader 表中, 在“phoneNumber”列后面添加新字段“email”, 在命令行窗口中输入如下命令。

```
ALTER TABLE reader ADD email varchar(100) AFTER phoneNumber;
```

4.4.4 修改字段名

MySQL 中修改表字段名的语法规则如下:

```
ALTER TABLE <表名> CHANGE <旧字段名> <新字段名> <新数据类型>;
```

其中, 旧字段名指修改前的字段名; 新字段名指修改后的字段名; 新数据类型指修改后的数据类型, 如果不需要修改字段的数据类型, 可以将新数据类型设置成与原来一样, 但数据类型不能为空。

【例 4-16】在 reader 表中, 把“phoneNumber”列名改为“cellPhone”, 在命令行窗口中输入如下命令。

```
ALTER TABLE reader CHANGE phoneNumber cellPhone varchar(12);
```

4.4.5 修改字段数据类型

修改字段的数据类型就是把字段的数据类型转换成另一种数据类型。在 MySQL 中修改字段数据类型的语法规则如下:

```
ALTER TABLE <表名> MODIFY <字段名> <数据类型>;
```

其中, 表名指要修改数据类型的字段所在表的名称; 字段名指需要修改的字段; 数据类型指修改后字段的新数据类型。

【例 4-17】在 reader 表中, 把“readerName”列的数据类型改成 VARCHAR(20), 在命令行窗口中输入如下命令。

```
ALTER TABLE reader MODIFY readerName varchar(20);
```

4.4.6 删除字段

删除字段是将数据表中的某个字段从表中移除, 语法格式如下:

```
ALTER TABLE <表名> DROP <字段名>;
```

其中，“字段名”指需要从表中删除的字段名称。

【例 4-18】在 reader 表中，把“email”列删除，在命令行窗口中输入如下命令。

```
ALTER TABLE reader DROP email;
```

4.4.7 修改表的存储引擎

修改表的存储引擎是将建表时或者数据库默认的存储引擎进行修改，语法格式如下：

```
ALTER TABLE <表名> ENGINE=INNODB|MYISAM|MEMOERY;
```

修改时，如果当前表本来的存储引擎是 INNODB 且存在外键约束，则会报错“ERROR 1217 (23000): Cannot delete or update a parent row: a foreign key constraint fails”。所以，创建表很关键，建表之后尽量不要修改存储引擎。

【例 4-18】将 book 表的存储引擎改为 MYISAM。由于 book 表中的 classNo 外键参照约束于 bookClass 表的 classNo，borrow 表中的 bookNo 外键参照约束于 book 表中的 bookNo。要想修改当前的 book 表的存储引擎，必须先将相关外键约束删除。在命令行窗口中输入如下命令。

```
ALTER TABLE book DROP FOREIGN KEY FK_Book_BookClass;
```

```
ALTER TABLE book DROP INDEX FK_Book_BookClass;
```

```
ALTER TABLE borrow DROP FOREIGN KEY FK_Borrow_Book;
```

```
ALTER TABLE borrow DROP INDEX FK_Borrow_Book;
```

```
ALTER TABLE book ENGINE=INNODB;
```