

主键约束(实体完整性)

主键 (PRIMARY KEY) 的完整称呼是“主键约束”，是 MySQL 中使用最为频繁的约束。一般情况下，为了便于 DBMS 更快地查找到表中的记录，都会在表中设置一个主键。主键本质上是一个排序，即索引。

主键分为单字段主键和多字段联合主键，本节将分别讲解这两种主键约束的创建、修改和删除。

使用主键应注意以下几点：

- ◆ 每个表只能定义一个主键。
- ◆ 主键值必须唯一标识表中的每一行，且不能为 NULL，即表中不可能存在有相同主键值的两行数据。这是唯一性原则。
- ◆ 一个字段名只能在联合主键字段表中出现一次。
- ◆ 联合主键不能包含不必要的多余字段。当把联合主键的某一字段删除后，如果剩下的字段构成的主键仍然满足唯一性原则，那么这个联合主键是不正确的。这是最小化原则。

1、在创建表时设置主键约束

在创建数据表时设置主键约束，既可以为表中的一个字段设置主键，也可以为表中多个字段设置联合主键。但是不论使用哪种方法，在一个表中主键只能有一个。下面分别讲解设置单字段主键和多字段联合主键的方法。

- ◆ 设置单字段主键

在 CREATE TABLE 语句中，通过 PRIMARY KEY 关键字来指定主键。

在定义字段的同时指定主键，语法格式如下：

```
<字段名> <数据类型> PRIMARY KEY [默认值]
```

或者是在定义完所有字段之后指定主键，语法格式如下：

```
[CONSTRAINT <约束名>] PRIMARY KEY [字段名]
```

- ◆ 在创建表时设置联合主键

所谓的联合主键，就是这个主键是由一张表中多个字段组成的。

主键由多个字段联合组成，语法格式如下：

```
PRIMARY KEY [字段 1, 字段 2, ..., 字段 n]
```

注意：当主键是由多个字段组成时，不能直接在字段名后面声明主键约束。

【例 5-1】为 Lib 数据库创建 school_bk 表，表结构参照 school 表。在 MySQL 的命令行格式中，输入以下命令：

```
CREATE TABLE school_bk (  
    schoolNo char(4) NOT NULL,  
    schoolName varchar(20) NOT NULL,  
    PRIMARY KEY (schoolNo)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

或者以下命令：

```
CREATE TABLE school_bk (  
    schoolNo char(4) NOT NULL PRIMARY KEY,  
    schoolName varchar(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

2、在修改表时添加主键约束

主键约束不仅可以在创建表的同时创建，也可以在修改表时添加。但是需要注意的是，设置成主键约束的字段中不允许有空值。

在修改数据表时添加主键约束的语法格式如下：

```
ALTER TABLE <数据表名> ADD PRIMARY KEY(<字段名>);
```

通常情况下，当在修改表时要设置表中某个字段的主键约束时，要确保设置成主键约束的字段中值不能够有重复的，并且要保证是非空的。否则，无法设置主键约束。

【例 5-2】为 Lib 数据库创建 bookClass_bk 表，表结构参照 bookClass 表。分两步，先创建无主键的表，再用 ALTER TABLE 实现添加主键。

```
CREATE TABLE bookclass_bk (  
    classNo char(4) NOT NULL,  
    className varchar(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
ALTER TABLE bookclass_bk ADD PRIMARY KEY(classNo);
```

3、删除主键约束

当一个表中不需要主键约束时，就需要从表中将其删除。删除主键约束的方法要比创建主键约束容易的多。

删除主键约束的语法格式如下所示：

```
ALTER TABLE <数据表名> DROP PRIMARY KEY;
```

由于主键约束在一个表中只能有一个，因此不需要指定主键名就可以删除一个表中的主键约束。

【例 5-3】删除 bookClass_bk 的主键。在 MySQL 的命令行格式中，输入以下命令：

```
ALTER TABLE bookClass_bk DROP PRIMARY KEY;
```

4、自增长主键

在 MySQL 中，当主键定义为自增长后，这个主键的值就不再需要用户输入数据了，而由数据库系统根据定义自动赋值。每增加一条记录，主键会自动以相同的步长进行增长。

AUTO_INCREMENT 关键字，可以在添加一条记录时，自动为该行添加数值，而且是自动增长的，一般被作为主键。通过给字段添加 AUTO_INCREMENT 属性来实现主键自增长。语法格式如下：

```
字段名 数据类型 AUTO_INCREMENT;
```

默认情况下，AUTO_INCREMENT 的初始值是 1，每新增一条记录，字段值自动加 1。

一个表中只能有一个字段使用 AUTO_INCREMENT 约束，且该字段必须有唯一索引，以避免序号重复（即为主键或主键的一部分）。

AUTO_INCREMENT 约束的字段必须具备 NOT NULL 属性。

AUTO_INCREMENT 约束的字段只能是整数类型（TINYINT、SMALLINT、INT、BIGINT 等）。

AUTO_INCREMENT 约束字段的最大值受该字段的数据类型约束，如果达到上限，AUTO_INCREMENT 就会失效。

【例 5-4】在 Lib 数据库中，borrow 表中的 borrowNo 是自增长主键。请创建 borrow 表。在 MySQL 的命令行格式中，输入以下命令：

```
CREATE TABLE borrow (  
    borrowNo int(11) NOT NULL AUTO_INCREMENT,  
    bookNo char(4) NOT NULL,  
    readerNo char(4) NOT NULL,  
    borrowDate datetime NOT NULL,  
    returnDate datetime NOT NULL,
```

```
re_new char(2) NOT NULL DEFAULT '否',
re_turn char(2) NOT NULL DEFAULT '否',
PRIMARY KEY (borrowNo),
KEY FK_Borrow_Reader (readerNo),
KEY FK_Borrow_Book (bookNo),
CONSTRAINT FK_Borrow_Book FOREIGN KEY (bookNo) REFERENCES book (bookNo),
CONSTRAINT FK_Borrow_Reader FOREIGN KEY (readerNo) REFERENCES reader
(readerNo)
) ENGINE=InnoDB AUTO_INCREMENT=500 DEFAULT CHARSET=utf8;
```

当前 borrowNo 是自增长主键，AUTO_INCREMENT 的初始值是 500，每新增一条记录，字段值自动加 1。