

目录

1. MySQL 主键 primary key:	1
1.1 主键的作用: 限制字段赋值	1
1.2 主键的使用规则	1
1.3 主键与 auto_increment 连用	1
1.4 创建复合主键	3
1.4.1 建表时创建	3
1.4.2 删除表的复合主键	3
1.4.3 在已有表里创建	4
2. 外键 foreign key	4
2.1 创建外键	4
2.1.1 使用规则	4
2.1.2 创建外键	5
2.1.3 删除表	7
2.1.4 删除外键	8

1. MySQL 主键 primary key:

1.1 主键的作用：限制字段赋值

1.2 主键的使用规则

具体要求如下：

- ✓ 字段值不允许重复，且不允许赋NULL值
- ✓ 一个表中只能有一个primary key字段
- ✓ 多个字段都作为主键，称为复合主键，必须一起创建。
- ✓ 主键字段的标志是PRI
- ✓ 主键通常与 auto_increment 连用
- ✓ 通常把表中唯一标识记录的字段设置为主键 [记录编号字段]

创建表时，表中存在类似身份证号，编号等时，将表中的该字段设置为主键，让其不能重复，可以自动增长。

1.3 主键与 auto_increment 连用

命令格式

- ✓ 字段名 类型 primary key auto_increment;

功能

- ✓ 字段通过自加1方式赋值

auto_increment 自增长(如果不给自增长字段赋值,该字段每次自加 1 给自己赋值，例如：1, 2, 3, 4, 5 等)，自增长通常和主键一同使用

创建 t1 表，字段有：编号(id)，姓名(name)，年龄(age)，班级(class)，将 id 字段设置为主键和自增长

```
mysql> create table t1(id int primary key auto_increment, name char(10), age tinyint, class char(10));
```

```
mysql> desc t1;
```

向 t1 表中插入数据，不给 id 字段赋值

```
mysql> insert into t1(name,age,class) values("bob", 19, "nsd2000");  
mysql> select * from t1;          #查看t1表中的数据，值为1  
mysql> insert into t1(name,age,class) values('tom',29,'nsd11');  
mysql> insert into t1(name,age,class) values('tom',29,'nsd11');  
mysql> select * from t1;
```

字段设置为主键和自增长后，自己也可以给字段赋值，下次该字段自增长时，从新的数字自加 1 赋值

在 t1 表中插入一条数据，指定 id 号为 7

```
mysql> insert into t1 values(7,'lucy',18,'nsd2311');  
mysql> select * from t1;
```

再次在 t1 表中插入数据，不指定 id 的值，采用自增长赋值

```
mysql> insert into t1(name,age,class) values('tomA',29,'nsd2111');  
查看t1表中的所有数据，id值为8，从7开始增长  
mysql> select * from t1;
```

主键的值不能为 NULL，也不能重复；当给一个字段添加了主键，并设置自增长，在赋值时，如果该字段的值为 null,也是可以成功插入数据，因为自增长属性会自动加 1 插入数据

t1 表中插入数据，id 值为 null,插入成功

```
mysql> insert into t1 values(null,'john',27,'nsd2111');  
mysql> select * from t1;
```

设置了自增长字段 id 的表，当删除所有数据后，重新插入数据如果不指定 id 字段的值，默认还是从以前的 id 值继续增长

删除 t1 表中的所有数据

```
mysql> delete from t1;
```

向t1表中插入数据，id字段的值不用定义，采用自增长

```
mysql> insert into t11(name,age,class) values('AAA',25,'nsd2111');
```

```
mysql> select * from t11;      #查看t1表中的所有数据，id值为10，验证成功
```

1.4 创建复合主键

具体如下：

- ✓ 表中多个字段一起做主键
- ✓ 插入记录时，不允许主键字段值同时重复

pri	pri	
客户端地址	服务端端口	状态
1.1.1.1	21	no
2.1.1.1	21	yes
2.1.1.1	22	no

1.4.1 建表时创建

语法格式：primary key(字段名列表);

```
mysql> create table t2 (cip char(15), port int, status enum("yes", "no"), primary \
key(cip,port));      \为换行符
```

```
mysql> desc t2;
```

在t2表中插入表记录

```
mysql> insert into t2 values("1.1.1.1", 21, "yes"); #成功
```

```
mysql> insert into t2 values("1.1.1.1", 21, "yes"); #失败，cip字段和port字段同时重复
```

```
mysql> insert into t2 values("2.1.1.1", 21, "yes"); #port字段值相同,cip字段值不同,成功
```

```
mysql> insert into t2 values("2.1.1.1", 22, "yes"); #cip字段值相同,port字段值不同,成功
```

```
mysql> select * from t2;
```

1.4.2 删除表的复合主键

```
mysql> alter table t2 drop primary key;
```

```
mysql> desc t2;
```

没有复合主键以后，插入记录就允许重复,不允许赋空值，约束条件NULL限制

```
mysql> insert into t2 values("2.1.1.1", 22, "yes");
```

1.4.3 在已有表里创建

语法格式: alter table 表名 add primary key(字段名列表);

```
mysql> alter table t2 add primary key(cip,port);          #创建失败, 表中的记录有重复
mysql> delete from t2;                                     #删除表中所有数据(也可以修改表记录)
mysql> alter table t2 add primary key(cip,port);          #成功设置
mysql> desc t2;
mysql> insert into t2 values("2.2.2.2", 22,"no");        #成功
```

2. 外键 foreign key

2.1 创建外键

2.1.1 使用规则

外键功能

✓ 插入表记录时, 字段值在另一个表字段值范围内选择。

使用规则

- ✓ 表存储引擎必须是innodb
- ✓ 字段类型要一致
- ✓ 被参照字段必须要是索引类型的一种(primary key)

yg(员工表, 参考)

gz(工资表)

yg_id(员工编号) name(姓名)

gz_id(员工编号)【外键】 pay(工资)

1

bob

1

2w

2

tom

2

3w

1》【员工表】记录的是员工信息;

2》【工资表】记录的是员工的工资信息;

3》给员工发工资必须在员工还在公司上班的基础上;

4》创建外键, 将【工资表】的 gz_id 设置为【员工表】yg_id 的外键, 即: 【工资表】的数据是随着【员工表】中的数据变化的, 当【员工表】的数据发生变化时, 【工资表】里的数据也会发生变化

2.1.2 创建外键

创建 yg(员工表), 表字段: yg_id(员工编号), name(员工姓名)

因为要创建外键, 将字段 yg_id 设置为主键并且是自增长的, 创建外键表的引擎必须是 innodb

```
mysql> create table yg(yg_id int primary key auto_increment, name \
char(15))engine=innodb; \为换行符
```

```
mysql> desc yg;
```

给yg(员工表)插入两条记录, 因为yg_id(员工编号)为自增长的, 只插入name值即可

```
mysql> insert into yg(name) values("bob"),("tom");
```

```
mysql> select * from yg;
```

创建 gz 表(工资表) ——》指定 gz 表中的字段 gz_id 为 yg 表字段 yg_id 的外键

```
mysql> create table gz( gz_id int, pay float(7,2), foreign key(gz_id) references \
yg(yg_id) on update cascade on delete cascade )engine=innodb; \为换行符
```

通过查看gz(工资表)建表过程, 来查看表是否创建外键

```
mysql> show create table gz \G;
```

外键设置成功之后, gz(工资表)插入数据时,编号必须在yg(员工表)的yg_id范围之内
查看yg(员工表)中的记录

```
mysql> select * from yg;
mysql> insert into gz values(1,5000);      #在gz(工资表)中插入记录成功, 编号1在yg(员工表)编号范围之内
mysql> insert into gz values(2,6000);      #在gz(工资表)中插入记录成功, 编号2在yg(员工表)编号范围之内
mysql> select * from gz;                    #查看gz(工资表)的表记录
mysql> insert into gz values(3,5000);      #在gz(工资表)中插入记录失败, 编号必须在yg(员工表)编号范围之内

在yg表里插入一条记录, 用户名为john, 编号采用自增长
mysql> insert into yg(name) values('john');
mysql> select * from yg;
mysql> insert into gz values(3,60000);      # gz(工资表)重新插入记录时, 插入成功
mysql> select * from gz;
```

当主表(yg 表)的记录被删除时, 外键所在表的记录也会被删除(同步删除)

删除 yg(员工表)中编号为 3 的记录信息

```
mysql> delete from yg where yg_id=3;
mysql> select * from gz;                    #gz(工资表)中的记录, 编号为3的记录也被删除
```

当主表的记录字段被修改时, 外键所在表的记录字段也会被修改(同步更新)

```
mysql> update yg set yg_id=6 where name="tom";      #找到(员工表)中name为tom的记录, 将yg_id修改为6
mysql> select * from gz;                      #查看gz(工资表)中的记录信息, 也被修改(同步更新)
```

设置成外键的表字段，也必须将其设置成主键，否则会出现对于同一个编号，可以插入多次数据的情况，也会出现编号为 null,插入数据同样成功的情况

```
mysql> select * from gz; #查看gz(工资表)的记录信息
mysql> insert into gz values(1,60000); #在gz(工资表)中，为编号1的员工再次发工资
mysql> insert into gz values(6,60000); #在gz(工资表)中，为编号6的员工再次发工资
mysql> insert into gz values(null,60000); #在gz(工资表)中，为编号null的员工发工资，
null指的是没有数据，插入成功
mysql> select * from gz; #查看gz(工资表)的记录信息
```

将表中作为外键的表字段，约束设置成主键，解决上述问题

```
mysql> delete from gz; #删除gz(工资表)中的所有记录，解决数据重复或为NULL的问题
mysql> select * from gz; #查看gz(工资表)中的所有记录
```

为已经存在的表gz(工资表)添加主键，作为主键的字段为gz_id

```
mysql> alter table gz add primary key(gz_id);
mysql> select * from yg; #查看yg(员工表)中的记录，gz(工资表)插入时，编号只能是1,6
mysql> insert into gz values(1,60000); #在gz(工资表)中，为编号1的员工发工资
mysql> insert into gz values(6,60000); #在gz(工资表)中，为编号6的员工发工资
mysql> insert into gz values(1,60000); #为编号1的员工重复发工资时，插入数据失败
mysql> insert into gz values(6,60000); #为编号6的员工重复发工资时，插入数据失败
mysql> insert into gz values(null,60000); #为编号为null的员工发工资时，插入数据失败
mysql> insert into gz values(7,60000); #插入编号不再1, 6范围内的数据时，失败
```

2.1.3 删除表

当一个表被其他表所依赖时，该表则不可以被删除

删除方法两种：1.删除 gz(工资表)的外键；2.先删除 gz(工资表)

```
mysql> drop table yg; #直接删除，失败
```


2.1.4 删除外键

语法格式： alter table 表名 drop foreign key 名称;

```
mysql> show create table gz\G;    #外键编号为: gz_ibfk_1
.....
CONSTRAINT `gz_ibfk_1` FOREIGN KEY (`gz_id`) REFERENCES `yg` (`yg_id`) ON
DELETE CASCADE
mysql> alter table gz drop foreign key gz_ibfk_1; #根据外键编号删除gz(工资表)的外键
mysql> show create table gz\G;    #查看gz(工资表)的建表过程，确认外键是否被删除
mysql> drop table yg;             #删除成功
```