

# 约束

---

## 1、为什么需要约束

数据是需要完整性，也可以是数据的精准性，可靠性，做约束就是为了防止数据库中存在不符合语义定义的数据和防止错误信息的输入输出造成的无效操作或错误信息的提出

## 2、什么是约束

约束就表的强制规定，可以在创建表的时候规定约束，，或者在创建之后通过alter table语句规定约束

## 3、约束的分类

根据约束的范围：

- 列级约束：只作用在一个字段（列）上面，在创建表的时候可以在列的后面定义
- 表级约束：可以作用在多个字段上面

根据约束的字的段的限制：

- 单列约束：每个约束只能约束一列
- 多列约束：每个约束可以约束多列

根据约束的作用，可以分为以下几种约束：

- not null 非空约束，规定某个字段不能为null
- unique 唯一约束，规定某个字段唯一
- primary key 主键约束，非空且唯一
- foreign key 外键约束
- check 检查约束
- default 默认值约束

注意：

mysql不支持check约束，可以使用，但是没有效果，传说mysql8以上的版本可以用，但是没试出效果

### 3.1、非空约束

限制某一个字段的值不允许为null

特点：

- 默认所有的类型的值都可以是null，包括null
- 非空约束只能出现在字段上面，只能某个字段单独限制非空，不能组合给
- 一个表可以有多个字段限制非空
- 0不是null，空字符串也不是null

添加非空约束：

语法：

```
create table 表名称(  
    字段名 数据类型 not null,  
    字段名 数据类型 not null,  
    字段名 数据类型  
)
```

举例：

```
create table students  --创建表，sname和cardid为非空约束  
(  
    sid int,  
    sname varchar(20) not null,  
    tel char(11),  
    cardid char(18) not null  
)  
  
insert into students  
values(1,'zhangsan','13111111111','123123123123123123');  
  
insert into students values(1,'zhangsan','13111111111',null); --  
失败，身份证不能为空
```

建表之后修改非空约束

```
alter table 表名称 modify 字段名 数据类型 not null;
```

删除非空约束

```
alter table 表名称 modify 字段名 数据类型;
```

## 3.2、唯一约束

用来限制某个字段的值不能重复

特点：

- 同一个表可以多个唯一约束
- 唯一约束可以是某一个字段数据的唯一，也可以是多个字段数据组合的唯一
- 唯一约束的值可以是null
- 在创建唯一约束的时候，可以给约束取名，如果不取默认与字段同名
- mysql会给唯一约束的字段上默认创建一个唯一索引

添加唯一约束：

语法：

```
create table 表名称
(
    字段名 数据类型 unique,
    字段名 数据类型 unique key
)

create table 表名称
(
    字段名 数据类型,
    字段名 数据类型,
    字段名 数据类型,
    [constraint 约束名] unique key(字段名,字段名)
)
```

举例：

```
create table student
(
    sid int,
    sname varchar(20),
    tel char(11) unique,
    cardid char(18) unique key
);

create table user
(
    id int not null,
    name varchar(20),
    cardid char(18),
    CONSTRAINT uk_name_cid unique(name,cardid)
```

)

#列独立约束

```
insert into student values(1,'张三','1311111111','123213123123123');
```

```
insert into student values(1,'张三','1311111112','123213123123123');
```

#组合约束

```
insert into user values(1,'zhangsan','999999') #成功
```

```
insert into user values(2,'zhangsan','999999') #失败
```

```
insert into user values(2,'zhangsan','888888') #成功
```

建表后添加组合唯一约束

```
alter table 表名称 add unique key(字段列表)
```

```
alter table 表名称 add CONSTRAINT 约束名 unique(字段列表)
```

```
alter table 表名称 modify 字段名 数据类型 unique;
```

关于复合唯一约束的情况:

#学生表

```
create table student(  
    sid int,  
    sname varchar(20),  
    tel char(11) unique key,  
    cardid char(18) unique key  
);
```

#课程表

```
create table course(  
    cid int,  
    cname varchar(20)  
);
```

#分数表

```
create table student_course(  
    id int,  
    sid int,
```

```

    cid int,
    score int,
    unique key(sid,cid)
);

# 张三 语文 80
# 张三 语文 90

insert into student values(2,'李四',17711112211,12121331121),
(3,'王五',17711444411,43211221121);

insert into course values(1,'前端'),(2,'JAVA');

insert into student_course values
(1,1,1,80),
(2,1,2,89),
(3,2,1,79),
(4,2,2,70)

```

分析:

这里我们通过对分数表中的sid（学生），cid（课程）做了一个组合约束的唯一约束，避免了出现一个学生在同一门课上出现不同分数的情况，对数据的完整性做了约束

删除复合唯一约束

```
alter table user drop index uk_name_cid
```

### 3.3、主键约束

用来标记表中的唯一数据，使用关键字primary key

特点:

- 主键约束相当于非空约束+唯一约束的组合，主键列不允许重复，不允许为null
- 一个表最多只能有一个主键约束，建立表的时候主键约束可以写在字段上也可以写在表上
- 主键约束可以对应表中的一个字段或多个字段（复合主键）
- 如果是多字段的复合主键约束，那么这些列都不允许为null，并且组合的值不允许重复
- mysql的主键名总是primary，就算自己命名也没有用
- 注意，不要修改主键字段的值，因为主键是数据记录的唯一标识，如果修改了主键的值，就有可能破坏数据的完整性

## 添加主键约束

语法:

```
create table 表名称(  
    字段名 数据类型 primary key,  
    字段名 数据类型,  
    字段名 数据类型  
)  
  
create table 表名称(  
    字段名 数据类型,  
    字段名 数据类型,  
    字段名 数据类型,  
    [constraint 约束名] primary key(字段列表) #表级添加, 约束名设置了也没  
用  
)
```

举例:

```
create table temp(  
    id int primary key,  
    name varchar(20)  
)  
  
#添加数据  
insert into temp values(1, '张三'), (2, '李四');  
  
insert into temp values(1, '王五'); #失败  
  
insert into temp values(null, '赵六'); #失败
```

制作一个复合主键 的表

#错误演示，主键重复定义

```
create table temp1(  
    id int primary key,  
    name varchar(20) primary key  
)
```

#正确演示

```
create table temp1(  
    id int,  
    name varchar(20),  
    pwd varchar(10),  
    constraint temp1_id_name primary key(id,name)  
)
```

## 复合主键的使用

```
create table student(  
    sid int primary key,  
    sname varchar(20)  
);
```

```
create table course(  
    cid int primary key,  
    cname varchar(20)  
);
```

```
create table student_course(  
    sid int,  
    cid int,  
    score int,  
    primary key(sid,cid)  
);
```

```
insert into student values(1,'zhangsan'),(2,'lisi');  
insert into course values(1,'javascript'),(2,'mysql');
```

```
insert into student_course values  
(1,1,60),  
(1,2,50),  
(2,1,70),  
(2,2,90)
```

## 删除主键约束

```
alter table 表名称 drop primary key
```

### 3.4、自增列

某个字段的值自己增加，使用关键字auto\_increment

特点：

- 一个表只能有一个自增列
- 自增列必须是唯一或主键约束的列
- 自增列的数据类型必须是整数型

错误演示：

```
create table emp1(  
    eid int auto_increment #错误，eid不是唯一或主键  
)  
  
create table emp2(  
    ename varchar(20) unique key auto_increment #错误，数据类型错误  
)
```

语法：

```
create table 表名称(  
    字段名 数据类型 primary key auto_increment  
)  
  
create table 表名称(  
    字段名 数据类型 unique key auto_increment  
)
```

删除自增列

```
alter table 表名称 modify 字段名 数据类型
```

### 3.5、外键约束

限定一个表中的某一个字段的引用完整性

比如：员工表中的所在部门，必须是部门表中能找到的部门，使用关键字foreign key



## 主表与从表（父表与子表）

- 主表：被引用的表，被参考的表
- 从表：引用别人的表，参考别人的表

比如：员工表的员工所在部门这个字段的值要参考部门表，所以部门表是主表，员工表是从表

比如：学生表、课程表、分数表、其中分数表的学生字段和课程字段分别参考学生表和课程表，所以课程表和学生表就是主表，分数表是从表

特点：

- 从表的外键列，必须引用主表的主键列或者唯一约束列
- 在创建外键约束的时候，如果不给外键约束命名，默认名不是列名，而是自动生成一个外键名，也可以指定外键名
- 如果在创建表的时候就指定外键约束的话，先创建主表，再创建从表
- 删除表的时候，先删从表，再删主表
- 当主表中的记录被从表参照之后，主表的记录将不被允许删除，如果要删除数据，需要先在从表中删除依赖该记录的数据，然后再删主表的
- 在从表中指定外键约束列，并且一个表是可以创建多个外键约束列的
- 从表的外键列和主表的参考列的字段名可以是不一样的，但是数据类型必须是一样的

添加外键约束：

语法：

```
create table 主表名称(  
    字段名 数据类型 primary key,  
    字段名 数据类型  
)  
  
create table 从表名称(  
    字段名 数据类型 primary key,  
    字段名 数据类型  
    [constraint 外键约束名] foreign key (从表的某个字段) references 主  
    表名(被引用字段)  
)
```

举例：

建表时创建外键约束

```

create table dept(           #主表
    did int primary key,     #部门编号--主键约束
    dname varchar(20)       #部门名称
);

create table emp(           #从表
    eid int primary key,     #员工编号
    ename varchar(10),      #员工姓名
    deptid int,             #所属部门
    foreign key (deptid) REFERENCES dept(did)
    #在从表中指定外键约束emp表中的deptid和主表dept表的did字段进行引用
);

```

### 建表后创建外键约束

```

alter table 从表名称 add [constraint 约束名] foreign key (从表的某个字段) references 主表名(被引用字段)

```

### 问题演示1:

```

create table dept(           #主表
    did int,                 #部门编号
    dname varchar(20)       #部门名称
);

create table emp(           #从表
    eid int primary key,     #员工编号
    ename varchar(10),      #员工姓名
    deptid int,             #所属部门
    foreign key (deptid) REFERENCES dept(did)
    #在从表中指定外键约束emp表中的deptid和主表dept表的did字段进行引用
);
#错误原因: 主表的did不是主键约束或唯一约束列

```

### 问题演示2:

```

create table dept(          #主表
    did int primary key,    #部门编号
    dname varchar(20)      #部门名称
);

create table emp(          #从表
    eid int primary key,    #员工编号
    ename varchar(10),     #员工姓名
    deptid char(4),        #所属部门
    foreign key (deptid) REFERENCES dept(did)
    #在从表中指定外键约束emp表中的deptid和主表dept表的did字段进行引用
);
#错误原因：主键与外键数据类型不统一

```

在添加数据的时候需要注意的问题：

```

insert into dept values(1001, '捞鱼部门'), (1002, '贴膜部门');

insert into emp values(1, '张三', 1001); #成功
insert into emp values(2, '李四', 1005); #失败，因为部门表中没有1005这个部门

update emp set deptid = 1002 where eid = 1; #成功
update emp set deptid = 1004 where eid = 1; #失败，因为部门表中没有1004这个部门

```

总结：

- 约束关系是针对双方的
- 添加外键约束之后，主表的修改和删除受约束
- 添加外键约束之后，从表的添加和修改受约束
- 在从表上建立外键，要求主表必须存在
- 删除主表的时候，要先删除从表，或者把从表中的引用关系删除掉

## 约束等级

- cascade方式：当父表上update / delete，同步update / delete 掉子表的匹配记录
- set null方式：当父表上update / delete，将子表上匹配记录的列设置成null
- no action方式：如果子表中有匹配的记录，则不允许父表对键列进行update / delete
- restrict方式：立即检查外键约束，同no action
- set default方式：父表有变动的时候，子表将外键列设置成一个默认值

如果没有设置等级，默认restrict

## 删除外键约束

```
alter table 从表名 drop foreign key 外键约束名
```

### 3.6、default约束

给某个字段指定默认值，一旦设置了默认值，在插入数据的时候，如果该字段没有赋值，则赋值默认值，使用关键字default

语法：

```
create table 表名称(  
    字段名 数据类型 primary key,  
    字段名 数据类型 unique key not null,  
    字段名 数据类型 default 默认值 not null  
)  
#说明：默认值约束一般不在主键列和唯一列上添加
```

举例：

```
create table tempd(  
    tid int primary key,  
    tname varchar(10) not null,  
    gender char default '男',  
    tel char(11) not null default ''  
)
```

删除修改默认值约束

```
alter table 表名 modify 字段名 数据类型 ....
```