

1、约束是什么

创建表的时候可以给表的字段添加相应的约束，目的是为了保证表中数据的合法性、有效性以及完整性。

--常见约束有：

非空约束（**not null**）：约束的字段中数据不能为null

唯一约束（**unique**）：约束的字段中数据不能重复

主键约束（**primary key**）：约束的字段中的数据既不能为null也不能重复，简称PK

外键约束（**foreign key**）：...简称FK

检查约束（**check**）：Oracle数据库中有该约束，MySQL中目前还没有

2、非空约束 (not null)

```
drop table if exists t_user;
create table t_user(
  id int,
  username varchar(255) not null,
  password varchar(255)
);
```

```
mysql> drop table if exists t_user;
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

```
mysql> create table t_user(
  -> id int,
  ->   username varchar(255) not null,
  ->   password varchar(255)
  -> );
Query OK, 0 rows affected (0.40 sec)
```

```
mysql> desc t_user;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	YES		NULL	
username	varchar(255)	NO		NULL	
password	varchar(255)	YES		NULL	

3 rows in set (0.00 sec)--可以看到表中的username中的null是NO

insert into t_user (id,password) values (1,'123');--报错，因为username不能为null，该字段不添加数据底层又无法给其一个null，但是又需要赋值，报没有设定默认值的错误

```
mysql> insert into t_user (id,password) values (1,'123');
ERROR 1364 (HY000): Field 'username' doesn't have a default value
```

2、唯一性约束 (unique)

unique修饰的字段不可以重复，但可以为null且多个null

```
drop table if exists t_user;
create table t_user(
  id int,
  username varchar(255) unique,
  password varchar(255)
```

```
);
insert into t_user values(1,'zz','123');
insert into t_user values(2,'zz','123');--报错

mysql> create table t_user(
    -> id int,
    ->     username varchar(255) unique,
    ->     password varchar(255)
    -> );
Query OK, 0 rows affected (0.34 sec)

mysql> insert into t_user values(1,'zz','123');
Query OK, 1 row affected (0.06 sec)

mysql> insert into t_user values(2,'zz','123');
ERROR 1062 (23000): Duplicate entry 'zz' for key 'username'

--但可以这样
insert into t_user(id) values(3);
insert into t_user(id) values(4);
mysql> select * from t_user;
+-----+-----+-----+
| id   | username | password |
+-----+-----+-----+
| 1    | zz      | 123      |
| 3    | NULL    | NULL     |
| 4    | NULL    | NULL     |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

3、给两个列或多个列字段添加unique

```
drop table if exists t_user;
create table t_user(
    id int,
    username varchar(255),
    password varchar(255),
    unique(username,password)--表级约束
);
--上面这种方式表示username和password这两个字段的数据联合拼接出来的字符不能出现重复，联合添加约束
--可以写成这样
insert into t_user values(1,'zz','111');
insert into t_user values(2,'zz','222');

--下面这种方式表示username和password单独插入数据不能重复，是分别添加约束
drop table if exists t_user;
create table t_user(
    id int,
    username varchar(255) unique,
    password varchar(255) unique--列级约束
);
```

4、主键约束

给一张表添加主键

```

drop table if exists t_user;
create table t_user(
    id int primary key,--列级约束
    username varchar(255),
    password varchar(255)
);
insert into t_user(id,username,password) values(1,'zs','@zs.com');
insert into t_user(id,username,password) values(2,'ls','@ls.com');
mysql> select * from t_user;
+----+-----+-----+
| id | username | password |
+----+-----+-----+
| 1 | zs      | @zs.com  |
| 2 | ls      | @ls.com  |
+----+-----+-----+
2 rows in set (0.00 sec)
--下面插入数据出现错误
insert into t_user(id,username,password) values(1,'ww','@ww.com');
mysql> insert into t_user(id,username,password) values(1,'ww','@ww.com');
ERROR 1062 (23000): Duplicate entry '1' for key 'PRIMARY'

```

由上面可知：id字段是主键字段，因为添加了主键约束primary key，所以该字段中的数据既不能为null，也不能重复。这是主键字段的特点。

--主键相关术语

主键字段：被主键约束primary key修饰的字段

主键值：主键字段中的值

主键约束：primary key

--主键的作用

表的设计三范式中有要求，第一范式就要求每张表中必须有主键。

作用：主键值是这张表中的某行记录的唯一标识，就是说无论哪两行记录的除了主键字段外的所有字段的值都相同，只要主键值不同，这就是两行不同的记录（就像人的身份证号一样）

--主键的分类

根据主键字段的数量来划分：

单一主键（推荐且是常用）

复合主键（多个字段联合起来使用一个主键）-不建议使用，违反三范式原则

根据主键性质划分：

自然主键：主键值最好是一个于业务没有任何关系的自然数

业务主键：主键值和系统的业务挂钩，例如用银行卡号或者身份证号做主键值，这是不推荐使用的。

原因是往后业务一旦改变，这些主键也需要随之变动，但是主键变动可能会导致主键重复，但主键是不能重复的，这对于业务影响较大。

--一张表中只能有一个主键约束

--使用表级约束定义主键

```

drop table if exists t_user;
create table t_user(
    id int ,
    username varchar(255),
    password varchar(255),
    primary key(id)
);

```

--mysql提供主键值自增，插入数据时没有给主键字段数据主键会自动按照主键值递增的方式赋值

```
--id字段自动维护一个自增的数字，从1开始，以1递增
drop table if exists t_user;
create table t_user(
    id int primary key auto_increment,
    username varchar(255),
    password varchar(255)
);
```

5、外键约束-foreign key

--外键相关术语

外键约束: foreign key

外键字段: 添加有外键约束的字段

外键值: 外键字段的每一个值

--业务背景

设计数据库表，用来维护学生和班级的信息

第一种方案: 设计一张表存储所有数据，缺点是太冗余了

第二种方案: 设计两张表，学生表和班级表

t_class 班级表

cno(PK) cname

101	西电1602052
102	西电1602051

t_student 学生表

sno(PK) sname classno(FK)

1	zs	101
2	ls	102
3	ww	102
4	cl	101

--由于设计两张表，我们可能不知道学生是哪个班级的，所以将学生表中的classno字段添加一个外键约束，用班级表中的cno字段(要求该字段是主键字段)里的数据来约束classno字段要填的值，即classno字段值只能填班级表cno字段中的值，填其他值会报错。这样就可以知道学生是哪个班级了。

--实现上面的建表语句

由于学生表中的某个字段只能填另一个表中的字段的值，填其他会报错，所以就有父表和子表之分，且建表、插入数据以及删表的顺序有先后

t_student中的classno字段引用t_class表中的cno字段，此时t_student叫做子表，t_class表叫做父表

--顺序如下

删除数据，先删子表，再删父表

添加数据，先添加父表，后添加子表

删除表，先删子表，再删父表

创建表，先创父表，后创子表

--references 表示引用的意思

```
drop table if exists t_student;
drop table if exists t_class;
create table t_class(
    cno int,
    cname varchar(255),
    primary key(cno)
);
```

```

create table t_student(
    sno int,
    sname varchar(255),
    classno int,
    foreign key (classno) references t_class(cno)
);
insert into t_class values(101,'xxxxxxxx');
insert into t_class values(102,'yyyyyyyy');
insert into t_student values(1,'zs',101);
insert into t_student values(2,'ls',102);
insert into t_student values(3,'ww',102);
insert into t_student values(4,'cl',101);
select * from t_class;
select * from t_student;

```

--将上面的sql语句放进记事本改为sql脚本就可以了

--由于开始没有将班级表中的cno设为主键，在创建学生表时无法添加外键约束导致了建学生表失败

```

mysql> drop table if exists t_student;
Query OK, 0 rows affected (0.24 sec)

```

```

mysql> drop table if exists t_class;
Query OK, 0 rows affected (0.13 sec)

```

```

mysql> create table t_class(
-> cno int,
->     cname varchar(255),
->     primary key(cno)
-> );
Query OK, 0 rows affected (0.26 sec)

```

```

mysql> create table t_student(
-> sno int,
->     sname varchar(255),
->     classno int,
->     foreign key (classno) references t_class(cno)
-> );
Query OK, 0 rows affected (0.28 sec)

```

```

mysql> insert into t_class values(101,'xxxxxxxx');
Query OK, 1 row affected (0.12 sec)

```

```

mysql> insert into t_class values(102,'yyyyyyyy');
Query OK, 1 row affected (0.03 sec)

```

```

mysql> insert into t_student values(1,'zs',101);
Query OK, 1 row affected (0.08 sec)

```

```

mysql> insert into t_student values(2,'ls',102);
Query OK, 1 row affected (0.01 sec)

```

```

mysql> insert into t_student values(3,'ww',102);
Query OK, 1 row affected (0.09 sec)

```

```

mysql> insert into t_student values(4,'cl',101);
Query OK, 1 row affected (0.03 sec)

```

```

mysql> select * from t_class;
+-----+-----+

```

```
| cno | cname |
+----+-----+
| 101 | xxxxxxxx |
| 102 | yyyyyyyy |
+----+-----+
2 rows in set (0.00 sec)
```

```
mysql> select * from t_student;
```

```
+-----+-----+-----+
| sno | sname | classno |
+-----+-----+-----+
| 1 | zs | 101 |
| 2 | ls | 102 |
| 3 | ww | 102 |
| 4 | cl | 101 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

--这时我们往t_student表中插入数据，当classno字段不是班级表中主键字段cno中的值时报错

```
insert into t_student values(5,'dw',103);
```

```
mysql> insert into t_student values(5,'dw',103);--子记录外键约束添加失败
```

```
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint
fails (`mysqlstudy`.`t_student`, CONSTRAINT `t_student_ibfk_1` FOREIGN KEY
(`classno`) REFERENCES `t_class` (`cno`))
```

--外键值是可以为null的

--外键字段引用其他表中的某个字段时，被引用的字段不一定是主键字段，但是至少具有unique约束，就是说不能重复