MySQL操作

一、创库创表语句（DDL）

1.创建school库

```
create database school;
```

```
mysql> create database school;
Query OK, 1 row affected (0.00 sec)
```

2.创建student表,要求表结构带有(id,name,age,gradname)

```
# 进入school库
use school
# 创建表
create table student(id tinyint(4),name varchar(20),age tinyint(4),gradname enum('first','secind','third'));
# 查看结果
desc school
```

```
mysql> use school
Database changed
mysql> create table student(id tinyint(4),name varchar(20),age tinyint(4),gradname
 enum('first','secind','third'));
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+-----------------+
| Tables_in_school |
+-----------------+
| student         |
+-----------------+
1 row in set (0.00 sec)

mysql> desc student;
+----------+-------------------------------+------+-----+---------+-------+
| Field    | Type                          | Null | Key | Default | Extra |
+----------+-------------------------------+------+-----+---------+-------+
| id       | tinyint(4)                    | YES  |     | NULL    |       |
| name     | varchar(20)                   | YES  |     | NULL    |       |
| age      | tinyint(4)                    | YES  |     | NULL    |       |
| gradname | enum('first','secind','third') | YES  |     | NULL    |       |
+----------+-------------------------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql>
```

二、alter修改表（DDL）

1.使用关键字alter修改表结构

修改表名SQL语句： `alter table 表名 rename to 新表名;`

```
# 修改表结构
alter table student rename to student_new;
# 查看修改结果
show tables;
```

```
mysql> alter table student rename to student_new;
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
+-----------------+
| Tables_in_school |
+-----------------+
| student_new      |
+-----------------+
1 row in set (0.00 sec)

mysql>
```

2.使用关键字alter修改字段名

修改字段SQL语句： `alter table biao_name change name new_name data_type;`（name为要修改的字段名，new_name为新的字段名，注意：字段名后面一定要加字段类型）

```
# 修改字段名
alter table student_new change age age_new int;
# 查看结果
desc student_new;
```

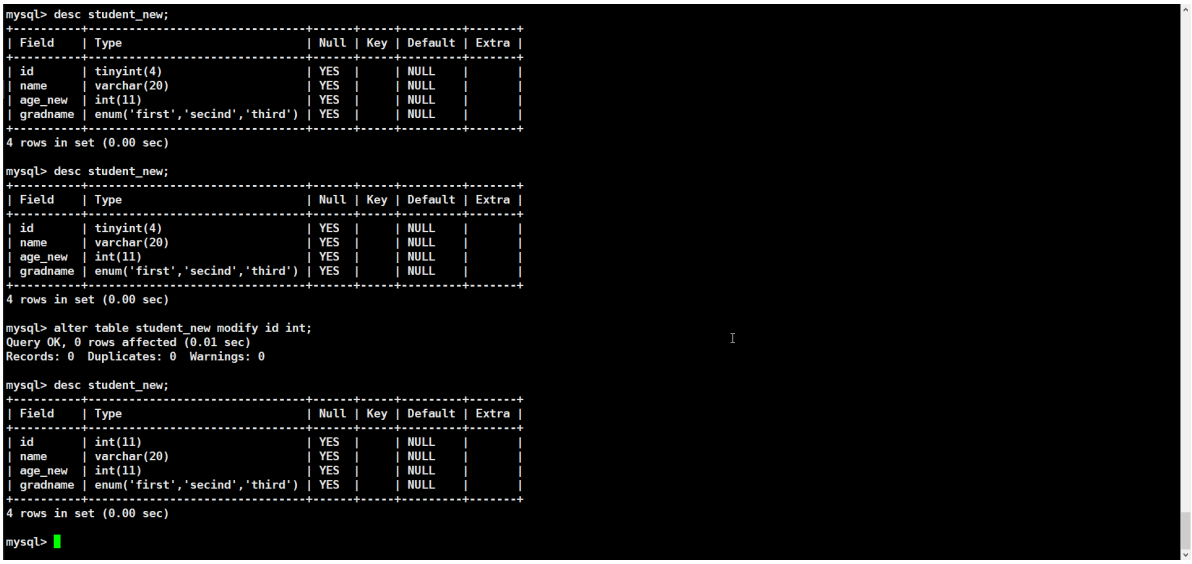
```
mysql> desc student_new;
+----------+-------------------------------+------+-----+---------+-------+
| Field    | Type                          | Null | Key | Default | Extra |
+----------+-------------------------------+------+-----+---------+-------+
| id       | tinyint(4)                    | YES  |     | NULL    |       |
| name     | varchar(20)                   | YES  |     | NULL    |       |
| age_new  | int(11)                       | YES  |     | NULL    |       |
| gradname | enum('first','secind','third') | YES  |     | NULL    |       |
+----------+-------------------------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql>
```

3.使用关键字alter修改字段类型

修改字段类型SQL语句： `alter table tb_name modify field_name data_type;`（注意修改字段类型时一定要注意不要超出范围。）

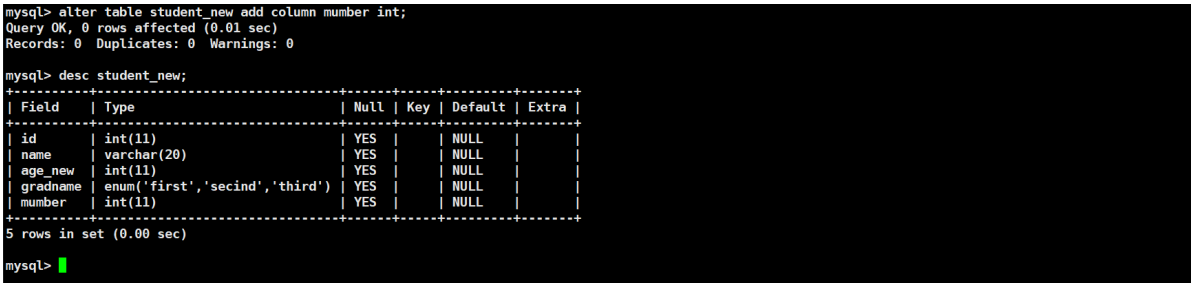
```
# 查看目前字段id的类型
desc student_new;
# 修改字段类型
alter table student_new modify id int;
```



4.使用关键字alter添加字段

添加字段SQL语句： `alter table tb_name add [column] field_name data_type;`（column可加可不加。）

```
# 添加字段
alter table student_new add column mumber int;
# 查看结果
desc student_new;
```



拓展：将字段添加到指定位置（指定字下面）

```
alter table student_new add columm sex char(2) after
name;
```

```
mysql> alter table student_new add column sex char(2) after name;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc student_new;
+----------+--------------------------------+------+-----+---------+-------+
| Field    | Type                           | Null | Key | Default | Extra |
+----------+--------------------------------+------+-----+---------+-------+
| id       | int(11)                        | YES  |     | NULL    |       |
| name     | varchar(20)                    | YES  |     | NULL    |       |
| sex      | char(2)                        | YES  |     | NULL    |       |
| age_new  | int(11)                        | YES  |     | NULL    |       |
| gradname | enum('first','secind','third') | YES  |     | NULL    |       |
| mumber   | int(11)                        | YES  |     | NULL    |       |
+----------+--------------------------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql>
```

将字段添加到第一行，则将after id改为first即可。

```
# 添加字段到第一行
alter table student_new add columm test int first;
# 查看结果
desc student_new;
```

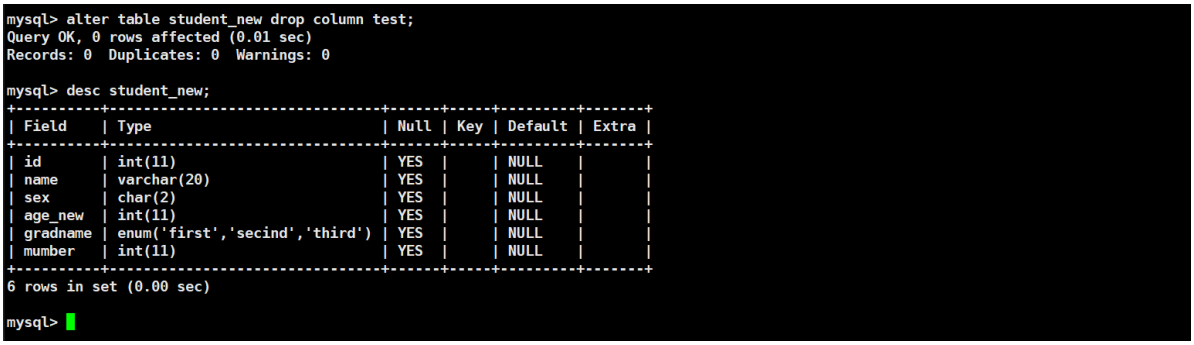
```
mysql> alter table student_new add column test int first;
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc student_new;
+----------+--------------------------------+------+-----+---------+-------+
| Field    | Type                           | Null | Key | Default | Extra |
+----------+--------------------------------+------+-----+---------+-------+
| test     | int(11)                        | YES  |     | NULL    |       |
| id       | int(11)                        | YES  |     | NULL    |       |
| name     | varchar(20)                    | YES  |     | NULL    |       |
| sex      | char(2)                        | YES  |     | NULL    |       |
| age_new  | int(11)                        | YES  |     | NULL    |       |
| gradname | enum('first','secind','third') | YES  |     | NULL    |       |
| mumber   | int(11)                        | YES  |     | NULL    |       |
+----------+--------------------------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

5.使用关键字alter删除字段

删除字段SQL语句： `alter table tb_name drop [column] field_name;（column可加可不加。）`

```
# 删除字段test
alter table student_new drop column test;
# 查看结果
desc student_new;
```



```
mysql> alter table student_new drop column test;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc student_new;
+----------+------------------------------+------+-----+---------+-------+
| Field    | Type                         | Null | Key | Default | Extra |
+----------+------------------------------+------+-----+---------+-------+
| id       | int(11)                      | YES  |     | NULL    |       |
| name     | varchar(20)                  | YES  |     | NULL    |       |
| sex      | char(2)                      | YES  |     | NULL    |       |
| age_new  | int(11)                      | YES  |     | NULL    |       |
| gradname | enum('first','secind','third') | YES  |     | NULL    |       |
| mumber   | int(11)                      | YES  |     | NULL    |       |
+----------+------------------------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql>
```

三、表关系（涉及insert插入操作）

通过外键的方式来实现表的主键连接另外一张表的主键。（给主键的字段连接外键）

> 实操引入：假设有两张表，一张是学生表student，学生表中有学号、姓名、学院，但学生还有些比如电话，家庭住址等比较私密的信息，这些信息不会放在学生表当中，会新建一个学生的详细信息表data1来存放。这时的学生表和学生的详细信息表两者的关系就是一对一的关系，因为一个学生只有一条详细信息。用主键加主键的方式来实现这种关系。

第一步：创建学生表

```
# 创建学生表
create table student(id int primary key, name varchar(20));
# 创建学生详细表
insert into student value(1,'xiaopeng');
insert into student value(2,'xiaoming');
insert into student value(3,'xiaoliang');
insert into student value(4,'xiaohong');
# 查看表情况
select * from student;
# 查看表结构
desc student;
```

```
mysql> create table student(id int primary key, name varchar(20));
Query OK, 0 rows affected (0.01 sec)

mysql> insert into student value(1,'xiaopeng');
Query OK, 1 row affected (0.00 sec)

mysql> insert into student value(2,'xiaoming');
Query OK, 1 row affected (0.00 sec)

mysql> insert into student value(3,'xiaoliang');
Query OK, 1 row affected (0.00 sec)

mysql> insert into student value(4,'xiaohong');
Query OK, 1 row affected (0.00 sec)

mysql> select * from student;
+----+----------+
| id | name     |
+----+----------+
|  1 | xiaopeng |
|  2 | xiaoming |
|  3 | xiaoliang |
|  4 | xiaohong |
+----+----------+
4 rows in set (0.00 sec)

mysql> desc student;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(11)     | NO   | PRI | NULL    |       |
| name  | varchar(20) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

第二步：学生详细信息表的创建：

注意：通过外键关联两张表，必须连接另外一张表里面的主键，则另外一张表（student）就是主 表 ,本表data1就是副表。

```
create table data(d_id int primary key, number int,
foreign key(d_id) references student(id));
#
insert into data value(1,222222);
insert into data value(2,333333);
insert into data value(3,444444);
insert into data value(4,555555);
# 查看数据信息
select * from data;
```

```
mysql> create table data(d_id int primary key, number int, foreign key(d_id) references student(id));
Query OK, 0 rows affected (0.00 sec)

mysql> insert into data value(1,222222);
Query OK, 1 row affected (0.01 sec)

mysql> insert into data value(2,333333);
Query OK, 1 row affected (0.00 sec)

mysql> insert into data value(3,444444);
Query OK, 1 row affected (0.00 sec)

mysql> insert into data value(4,555555);
Query OK, 1 row affected (0.00 sec)

mysql> select * from data;
+------+--------+
| d_id | number |
+------+--------+
|    1 | 222222 |
|    2 | 333333 |
|    3 | 444444 |
|    4 | 555555 |
+------+--------+
4 rows in set (0.00 sec)
```

第三步：通过外连接实现一对一（这里还体现了外键约束）：

```
select * from student left join data on student.id =
data.d_id;
```

```
mysql> select * from student left join data on student.id = data.d_id;
+----+----------+------+--------+
| id | name     | d_id | number |
+----+----------+------+--------+
|  1 | xiaopeng |    1 | 222222 |
|  2 | xiaoming |    2 | 333333 |
|  3 | xiaoliang |    3 | 444444 |
|  4 | xiaohong |    4 | 555555 |
+----+----------+------+--------+
4 rows in set (0.00 sec)
```

注意：删除数据的时候：
要先删除副表的，再删除主表的，才能删掉 。如果直接删除主表的将报错。

```sql
# 删除主表id
delete from data where d_id = 1;
# 删除副表id
delete from student where id = 1;
# 查看结果
select * from student;
```

```
mysql> delete from student where id = 1;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint
fails (`school`.`data`, CONSTRAINT `data_ibfk_1` FOREIGN KEY (`d_id`) REFERENCES `s
tudent` (`id`))
mysql> delete from data where d_id = 1;
Query OK, 1 row affected (0.00 sec)

mysql> delete from student where id = 1;
Query OK, 1 row affected (0.00 sec)

mysql> select * from student;
+----+----------+
| id | name     |
+----+----------+
|  2 | xiaoming |
|  3 | xiaoliang |
|  4 | xiaohong |
+----+----------+
3 rows in set (0.00 sec)
```

(2)多对一（或者称一对多）关系—非主键连接主键

自己的这张表的非主键连接另外一张表的主键,通过外键的方式实现。(给非主键的字段连接外键)

第一步：创建一个没有主键的表：

没有主键的这张表的非主键连接另外一张有主键的表的主键。则实现了多对一。

因为：非主键可以重复，主键唯一。

```
# 创建一个没有主键的表
create table data2(d_id int, number int, foreign key(d_id) references student(id));
# 关闭主键检查
SET FOREIGN_KEY_CHECKS = 0;
否则会出现 : ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
# 创建数据
insert into data2 value(1,232332);
insert into data2 value(1,241241);
insert into data2 value(1,290832);
```

第二步：创建一个有主键的表：

```
# 创表
create table stub(id int primary key, name varchar(20));
# 创数据
insert into stub value(1,'xiaohuang');
insert into stub value(2,'xiaolv');
insert into stub value(3,'xiaomei');
insert into stub value(4,'xiaozhi');
# 看结果
select * from student;
```

```
mysql> create table student(id int primary key, name varchar(20));
ERROR 1050 (42S01): Table 'student' already exists
mysql> create table stub(id int primary key, name varchar(20));
Query OK, 0 rows affected (0.01 sec)

mysql> insert into stub value(1,'xiaohuang');
Query OK, 1 row affected (0.01 sec)

mysql> insert into stub value(1,'xiaolv');
ERROR 1062 (23000): Duplicate entry '1' for key 'PRIMARY'
mysql> insert into stub value(2,'xiaolv');
Query OK, 1 row affected (0.01 sec)

mysql> insert into stub value(3,'xiaomei');
Query OK, 1 row affected (0.01 sec)

mysql> insert into stub value(4,'xiaozhi');
Query OK, 1 row affected (0.01 sec)
```

第三步：通过外连接查看多对一：

```
select * from stub left join data2 on student.id = data2.d_id;
```

```
mysql> select * from stub left join data2 on stub.id = data2.d_id;
+----+----------+------+--------+
| id | name     | d_id | number |
+----+----------+------+--------+
|  1 | xiaohuang |    1 | 232332 |
|  1 | xiaohuang |    1 | 241241 |
|  1 | xiaohuang |    1 | 290832 |
|  2 | xiaolv    | NULL |   NULL |
|  3 | xiaomei   | NULL |   NULL |
|  4 | xiaozhi   | NULL |   NULL |
+----+----------+------+--------+
6 rows in set (0.00 sec)
```

(3)多对多关系:

思路是非主键连接非主键，但是外键必须连接主键，所以必须有一个中间表，负责结合两张表的主键

联合主键：　(1,1)

添加联合主键：alter table 表名 add primary key(字段1，字段2);
实操引入：学生要报名选修课，一个学生可以报名多门课程，一个课程有很多的学生报名，那么学生表和课程表两者就形成了多对多关系。

下面是实操讲解（主要目的是讲解，所以我将两张表的字段做了简化）：

第一步：创建第一个表：

```
# 创表
create table stud(id int primary key, name varchar(20));
# 添加数据
insert into stud value(1,'xiaot');
insert into stud value(2,'xiaoy');
insert into stud value(3,'xiaod');
insert into stud value(4,'xiaof');
# 查询
select * from stud;
# 查看表结构
desc stud;
```

```
mysql> create table stud(id int primary key, name varchar(20));
Query OK, 0 rows affected (0.01 sec)

mysql> insert into stud value(1,'xiaot');
Query OK, 1 row affected (0.00 sec)

mysql> insert into stud value(2,'xiaoy');
Query OK, 1 row affected (0.01 sec)

mysql> insert into stud value(3,'xiaod');
Query OK, 1 row affected (0.00 sec)

mysql> insert into stud value(4,'xiaof');
Query OK, 1 row affected (0.00 sec)

mysql> select * from stud;
+----+-------+
| id | name  |
+----+-------+
|  1 | xiaot |
|  2 | xiaoy |
|  3 | xiaod |
|  4 | xiaof |
+----+-------+
4 rows in set (0.00 sec)

mysql> desc stud;
+-------+------------+------+-----+---------+-------+
| Field | Type       | Null | Key | Default | Extra |
+-------+------------+------+-----+---------+-------+
| id    | int(11)    | NO   | PRI | NULL    |       |
| name  | varchar(20) | YES  |     | NULL    |       |
+-------+------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

第二步：创建第二个表：

```
# 创表
create table stuc(c_id int primary key, name varchar(20));
# 添加数据
insert into stuc value(1,'C');
insert into stuc value(2,'C++');
insert into stuc value(3,'PHP');
insert into stuc value(4,'JAVA');
# 查询
select * from stuc;
# 查看表结构
desc stuc;
```

```
mysql> create table stuc(c_id int primary key, name varchar(20));
Query OK, 0 rows affected (0.01 sec)

mysql> insert into stuc value(1,'C');
Query OK, 1 row affected (0.00 sec)

mysql> insert into stuc value(2,'C++');
Query OK, 1 row affected (0.00 sec)

mysql> insert into stuc value(3,'PHP');
Query OK, 1 row affected (0.00 sec)

mysql> insert into stuc value(4,'JAVA');
Query OK, 1 row affected (0.01 sec)

mysql> select * from stuc;
+------+------+
| c_id | name |
+------+------+
|    1 | C    |
|    2 | C++  |
|    3 | PHP  |
|    4 | JAVA |
+------+------+
4 rows in set (0.00 sec)

mysql> desc stuc;
+-------+------------+------+-----+---------+-------+
| Field | Type       | Null | Key | Default | Extra |
+-------+------------+------+-----+---------+-------+
| c_id  | int(11)    | NO   | PRI | NULL    |       |
| name  | varchar(20) | YES  |     | NULL    |       |
+-------+------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

第三步：上面已经创建了两张表，还需要一张中间表：

这张中间表就将上面两张表都连接起来了，实现了多对多：

```
create table link(l_id int, c_id int, primary key(l_id,c_id), foreign key(l_id) references stud(id), foreign key(c_id) references link(l_id));
```

上面这张图的primary key(id,c_id)是创建联合主键

> foreign key(l_id) references stud(id)是关联学生的id
> foreign key(c_id) references link(l_id)是关联stuc的c_id

但这样查看不直观，所以用连接查询看：

```
select * from stud left join link on stud.id = link.l_id
left join stuc on stuc.tuc.c_id = link.c_id;
```

```
mysql> select * from stud left join link on stud.id = link.l_id left join stuc on stuc.
tuc.c_id = link.c_id;
+----+-------+------+------+------+------+
| id | name  | l_id | c_id | c_id | name |
+----+-------+------+------+------+------+
|  1 | xiaot | NULL | NULL | NULL | NULL |
|  2 | xiaoy | NULL | NULL | NULL | NULL |
|  3 | xiaod | NULL | NULL | NULL | NULL |
|  4 | xiaof | NULL | NULL | NULL | NULL |
+----+-------+------+------+------+------+
4 rows in set (0.00 sec)
```

4.拓展：

1.视图：（把查询出来的结果变成一张表）

特点：是一张虚表，可以引用多张表。

优点：

> 1.简单化，数据所见即所得。
> 2.安全性，用户只能查询或者修改他们看得到的数据。
> 3.逻辑独立性，可以屏蔽真实表结构变化带来的影响。比如：如果给一张有主键的表建立视图，那么这张表的视图是没有主键的。

缺点：

> 1.性能较差，简单的查询也会变得复杂
> 2.修改不方便，复杂的聚合视图基本无法修改

创建视图：`create view 视图名字 as select * from 表名;`

> 1.一张表形成的视图是可以进行增删改查的；
>
> 2.对视图进行修改，对应的原表也会跟着修改。但是多张表（通过连接查询生成的视图）形成的视图不能进行修改操作。

删除视图：`drop view 视图名字;`

四、约束条件

约束是一种限制条件，通过对表结构做出限制，来确保表的完整性和唯一性。

约束类型:	默认	非空	唯一	自增长	主键	外
关键字:	default	not null	unique key	auto_increment	primary key	fore ke

1.默认约束（default)

插入数据的时候，如果没有明确为字段赋值，则自动赋予默认值,在没有设置默认值的情况下，默认值为NULL.

默认约束的SQL语句为： `create table tearcher( id int default ‘a’ , name varchar(20));`

```
# 创建tercher库并添加约束
create table teacher(id int default 0, name varchar(20));
# 添加成员
insert into teacher(name) values('wanglaoshi');
# 查看约束结果
select * from teacher;
```

```
mysql> create table teacher(id int default 0, name varchar(20));
Query OK, 0 rows affected (0.01 sec)

mysql> insert into teacher(name) values('wanglaoshi');
Query OK, 1 row affected (0.02 sec)

mysql> select * from teacher;
+------+------------+
| id   | name       |
+------+------------+
|    0 | wanglaoshi |
+------+------------+
1 row in set (0.00 sec)

mysql>
```

```
mysql> desc teacher;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(11)     | YES  |     | 0       |       |
| name  | varchar(20) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

主动使用默认值添加用户： `insert into 表名 value(default,default);`

```sql
# 添加实验成员liulaoshi
insert into teacher value (default,'liulaoshi');
# 查看约束结果
select * from teacher;
```

```
mysql> insert into teacher value (default,'liulaoshi');
Query OK, 1 row affected (0.00 sec)

mysql> select * from teacher;
+------+------------+
| id   | name       |
+------+------------+
|    0 | wanglaoshi |
|    0 | liulaoshi  |
+------+------------+
2 rows in set (0.00 sec)

mysql>
```

使用modify修改字段类型的命令修改默认值:`alter table 表名 modify 字段名 字段类型 default 新的默认值;`

```sql
# 修改默认值
alter table teacher modify id int default 1;
# 查看结果
desc teacher;
```

```
mysql> alter table teacher modify id int default 1;
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc teacher
    -> ;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(11)     | YES  |     | 1       |       |
| name  | varchar(20) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

2.非空约束（not null）

限制一个字段的值不能为空，Insert的时候必须为该字段赋值。不赋值，则使用默认值null，但规定非空，即不能为null，所以会报错。

注意：空字符不等于NULL。

非空约束的SQL语句为:`create table tb( id int not null, name varchar(20));`

```sql
# 创建stu库并添加约束条件
create table stu(id int not null,name varchar(20));
# 测试添加一个id为空的sql语句是否能成功
insert into stu(name) value('xiaoming');
```

```
mysql> create table stu(id int not null,name varchar(20));
Query OK, 0 rows affected (0.00 sec)

mysql> insert into stu(name) value('xiaoming');
ERROR 1364 (HY000): Field 'id' doesn't have a default value
```

删除非空约束，在后面加上null： `alter table tb_name modify field_name data_type null;`

添加非空约束: `alter table tb_name modify field_name data_type not null;`

```sql
# 取出id字段的 not null 约束
alter table stu modify id int null;
# 测试添加一个id为空的sql语句是否能成功
insert into stu(name) value('xiaoming');
# 查看结果
select * from stu;
```

```
mysql> alter table stu modify id int null;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> insert into stu(name) value('xiaoming');
Query OK, 1 row affected (0.00 sec)

mysql> select * from stu;
+------+----------+
| id   | name     |
+------+----------+
| NULL | xiaoming |
+------+----------+
1 row in set (0.00 sec)
```

3.唯一约束（unique key）

限制一个字段的值不重复，该字段的数据不能出现重复的。

作用：确保字段中值的唯一。

唯一约束的SQL语句为 `create table tb2( id int unique key, name varchar(20));`

```sql
# 创建stu2库并添加约束条件
create table stu2(id int unique key,name varchar(20));
# 添加实验语句
insert into stu2 values(1,'xiaohong'),(1,'xiaogang');
```

```
mysql> create table stu2(id int unique key,name varchar(20));
Query OK, 0 rows affected (0.01 sec)

mysql> insert into stu2 values(1,'xiaohong'),(1,'xiaogang');
ERROR 1062 (23000): Duplicate entry '1' for key 'id'
```

删除唯一约束： `drop index 字段名 on 表名;`

```
# 删除约束
drop index id on stu2;
# 添加实验语句
insert into stu2 values(1,'xiaohong'),(1,'xiaogang');
# 查看结果
select * from stu2;
```

4.主键约束（primary key）

作用： 通常每张表都需要一个主键来体现唯一性，每张表里面只能有一个主键。

特点： 主键 = 非空 + 唯一。

主键约束SQL语句： `create table stu3(id int primary key, name varchar(20));`

```
# 创建测试表
create table stu3(id int primary key, name
varchar(20));
# 添加测试语句1
insert into stu3 value(1,'xiaopeng');
# 添加测试语句2验证主键的唯一性
insert into stu3 value(1,'xiaopeng');
# 添加测试语句3验证主键的非空性
insert into stu3(name) value('test');
# 查看stu3库
desc stu3;
```

```
mysql> create table stu3(id int primary key, name varchar(20));
Query OK, 0 rows affected (0.00 sec)

mysql> insert into stu3 value(1,'xiaopeng');
Query OK, 1 row affected (0.00 sec)

mysql> insert into stu3 value(1,'xiaopeng');
ERROR 1062 (23000): Duplicate entry '1' for key 'PRIMARY'
mysql> insert into stu3(name) value('test');
ERROR 1364 (HY000): Field 'id' doesn't have a default value
mysql> desc stu3;
    -> ;
ERROR 1146 (42S02): Table 'cc.stu3; ' doesn't exist
mysql> desc stu3;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(11)     | NO   | PRI | NULL    |       |
| name  | varchar(20) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

删除主键： `alter table 表名 drop primary key;`

```
alter table stu3 drop primary key;
```

```
mysql> desc stu3;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(11)     | NO   | PRI | NULL    |       |
| name  | varchar(20) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> alter table stu3 drop primary key;
Query OK, 1 row affected (0.02 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql> desc stu3;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(11)     | NO   |     | NULL    |       |
| name  | varchar(20) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

增加主键（三种方式）：

```
1.alter table 表名 add primary key(字段名);

2.alter table tb_name change name new_name data_type primary key;

3.alter table tb_name modify field_name data_type primary key;
```

5.自增长约束（auto_increment)

默认值从1开始，每次在最后一个值的基础上增加1，自动编号，和主键组合使用，一个表里面只能有一个自增长(原因：auto_increment 要求用在主键上,一张表就一个主键)

自增长约束SQL语句`create table ta( id int primary key auto_increment, name varchar(20));`

```
# 创建测试表
create table stu4(id int primary key auto_increment, name varchar(20));
# 查看表结构
desc stu4;
```

```
mysql> create table stu4(id int primary key auto_increment, name varchar(20));
Query OK, 0 rows affected (0.00 sec)

mysql> desc stu4;
+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| id    | int(11)     | NO   | PRI | NULL    | auto_increment |
| name  | varchar(20) | YES  |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)
```

设置自增长的增长幅度（默认为1）： set @@auto_increment_increment = 指定步长;

```
# 设置自增长的幅度为2
set @@auto_increment_increment=2;
# 添加两个成员
insert into stu4 value(1,'x');
insert into stu4(name) value('z');
# 查看id增长
select * from stu4;
```

```
mysql> set @@auto_increment_increment=2;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into stu4 value(1,'x');
Query OK, 1 row affected (0.00 sec)

mysql> insert into stu4(name) value('z');
Query OK, 1 row affected (0.01 sec)

mysql> select * from stu4;
+----+------+
| id | name |
+----+------+
|  1 | x    |
|  3 | z    |
+----+------+
2 rows in set (0.00 sec)
```

6.外键约束（foreign key）

外键：通过外键关联两张表，必须连接另外一张表里面的主键，另外一张表就是主表，本表就是副表。

保持数据的一致性，主表和副表是因为使用外键连接表才产生的概念。

创建外键约束： `alter table 表名1 add constraint 表名2 foreign key(id1) references 表名3(id2);`

删除外键约束： `alter table h2 drop foreign key fk_name;`

注意：

1. 外键名字可以随便取，因为如果你创建表的时候就增加外键，系统默认会有一个外键名字，可以通过`show create table 表名;`查看。
2. 自己的这张表连接另外一张表的主键（注意连接的一定要是主键），连接的主表里面没有的数据，添加外键的这张表里也一定不能有。

外键约束代码操作：

```sql
# 创建测试表1
create table fk(id int primary key, name varchar(20));
# 插入数据
insert into fk value(1,'jiejie');
insert into fk value(2,'weiwei');
insert into fk value(3,'mingming');
insert into fk value(4,'lele');
insert into fk value(5,'kele');
# 查询数据
select * from fk;
```

```
mysql> create table fk(id int primary key, name varchar(20));
Query OK, 0 rows affected (0.01 sec)

mysql> insert into fk value(1,'jiejie');
Query OK, 1 row affected (0.01 sec)

mysql> insert into fk value(2,'weiwei');
Query OK, 1 row affected (0.00 sec)

mysql> insert into fk value(3,'mingming');
Query OK, 1 row affected (0.01 sec)

mysql> insert into fk value(4,'lele');
Query OK, 1 row affected (0.00 sec)

mysql> insert into fk value(5,'kele');
Query OK, 1 row affected (0.01 sec)

mysql> select * from fk;
+----+----------+
| id | name     |
+----+----------+
|  1 | jiejie   |
|  2 | weiwei   |
|  3 | mingming |
|  4 | lele     |
|  5 | kele     |
+----+----------+
5 rows in set (0.00 sec)
```

```sql
# 创建测试表2
create table fk2(id int, number int);
# 查询fk2表结构
desc fk2;
```

```
mysql> create table fk2(id int, number int);
Query OK, 0 rows affected (0.02 sec)

mysql> desc fk2
    -> ;
+--------+---------+------+-----+---------+-------+
| Field  | Type    | Null | Key | Default | Extra |
+--------+---------+------+-----+---------+-------+
| id     | int(11) | YES  |     | NULL    |       |
| number | int(11) | YES  |     | NULL    |       |
+--------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

第一张表有主键，第二张表没主键且没在创建表的时候就增加外键约束，所以下面增加外键：

```
# 创建外键约束
alter table fk2 add constraint fk foreign key(id) references fk(id);
# 查看约束
show create table fk2\G;
```

```
mysql> alter table fk2 add constraint fk foreign key(id) references fk(id);
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show create table fk2\G;
*************************** 1. row ***************************
       Table: fk2
Create Table: CREATE TABLE `fk2` (
  `id` int(11) DEFAULT NULL,
  `number` int(11) DEFAULT NULL,
  KEY `fk` (`id`),
  CONSTRAINT `fk` FOREIGN KEY (`id`) REFERENCES `fk` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.00 sec)
```

删除外键约束：

```
# 删除外键约束
alter table fk2 drop foreign key fk;
# 查看
show create table fk2\G;
```

```
mysql> alter table fk2 drop foreign key fk;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show create table fk2\G;
*************************** 1. row ***************************
       Table: fk2
Create Table: CREATE TABLE `fk2` (
  `id` int(11) DEFAULT NULL,
  `number` int(11) DEFAULT NULL,
  KEY `fk` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.00 sec)
```