[MySQL的环境变量配置 2](#_Toc529535972)

[完整性约束条件 3](#_Toc529535973)

[主键约束 4](#_Toc529535974)

[单字段主键 4](#_Toc529535975)

[多字段主键 5](#_Toc529535976)

[外键约束 6](#_Toc529535977)

[其它完整性约束 8](#_Toc529535978)

[对表的修改 9](#_Toc529535979)

[修改表名 9](#_Toc529535980)

[修改字段的数据类型 9](#_Toc529535981)

[修改字段名称 10](#_Toc529535982)

[增加一个字段 11](#_Toc529535983)

[从已有表中删除一个字段 13](#_Toc529535984)

[修改表的存储引擎. 13](#_Toc529535985)

[删除表的外键约束 13](#_Toc529535986)

[删除表的主键 15](#_Toc529535987)

[对已有表添加主键约束 15](#_Toc529535988)

[对已有表添加外键约束 15](#_Toc529535989)

[各种查询语句 16](#_Toc529535990)

[查询语句的基本语法 16](#_Toc529535991)

[单表查询 17](#_Toc529535992)

[聚合函数查询 17](#_Toc529535993)

[多表查询 17](#_Toc529535994)

[子查询 17](#_Toc529535995)

[合并查询 17](#_Toc529535996)

[别名查询 17](#_Toc529535997)

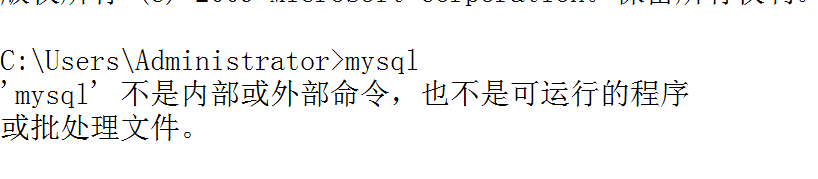
[正则表达式查询 17](#_Toc529535998)

[Limit限制查询 17](#_Toc529535999)

[Navicat连接MySQL8.0 17](#_Toc529536000)

# MySQL的环境变量配置

我们使用的是解压方式来安装的数据库(如果是用安装向导安装的,在安装过程会自动添加环境变量.当然也有不添加的时候),需要我们自己手工配置环境变量.不然会产生如下错误



解决方案就是:把MySQL的bin目录添加到Path中.记得当修改了环境变量.CMD是要重启的.

# 完整性约束条件

完整性约束条件是对字段进行限制.要求用户对该属性(字段)的操作符合特定的要求.如果不满足完整性约束,数据库系统将不会执行任何用户操作.

完整性约束是为了保护数据库中数据的完整性和参照完整性.

MySQL中有6种完整性约束

|  |  |
| --- | --- |
| 约束条件 | 说明 |
| Primary key | 标识该字段(属性)为该表的主键.可以唯一标识某条记录 |
| Foreign key | 标识该字段(属性)为该表的外键.与它联系的是某表的主键. |
| Not Null | 标识该字段(属性)不能为空.非空 |
| Unique | 标识该字段(属性)的值是唯一的 |
| Default | 为该字段(属性)设置默认值 |
| Auto\_Increment | 标识该字段(属性)的值是自动增长的.这个用法是MySQL的特色语句 |

对于其他SQL来说还有一种约束叫做:Check受查约束.MySQL是不支持的.

自动自增在不同的SQL中有不同的写法

|  |  |
| --- | --- |
| SQL | 自动增长写法 |
| MySQL | Auto\_increment |
| Oracle | Sequence |
| SQL Server | Identity(x,y) |

## 主键约束

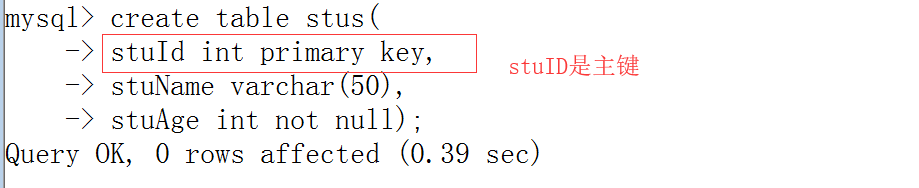
主键是表的一个或若干个特殊字段,它可以唯一标识表中的每条信息.主键和记录的关系,如同我们每个人和我们的身份证号码关系.

主键的设置方式有两种,可以在建表时设置,也可以对已有表进行修改,然后添加主键.

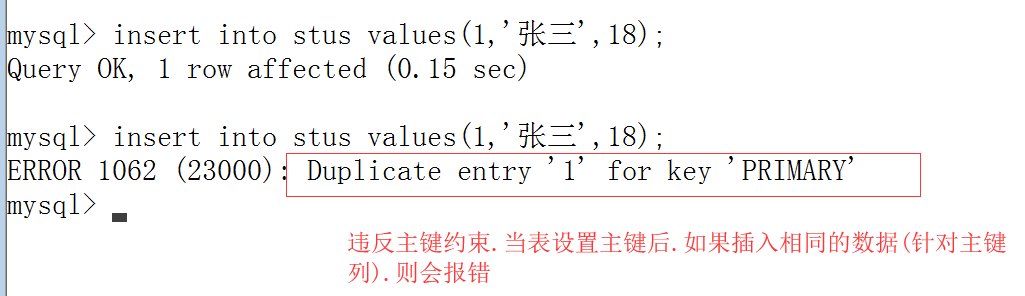
主键可以是单一的字段,也可以由多个字段组合.

### 单字段主键

该类型的主键是由一个字段构成.我们可以在建表时,直接在字段的后面添加private key来设置.具体的语法:属性名 数据类型 private key.

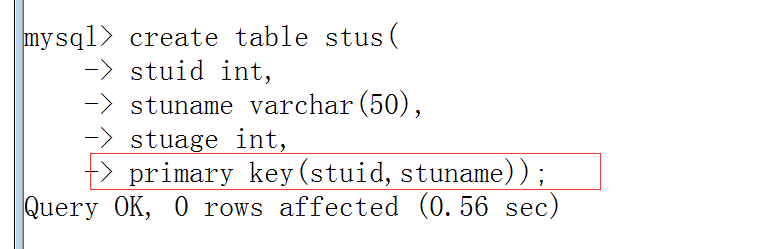


主键的是保证数据唯一.即主键本身的取值就是唯一的.我们可以进行简单测试.对主键列插入重复数据.如果成功代表创建失败,如果报错代表主键创建成功!

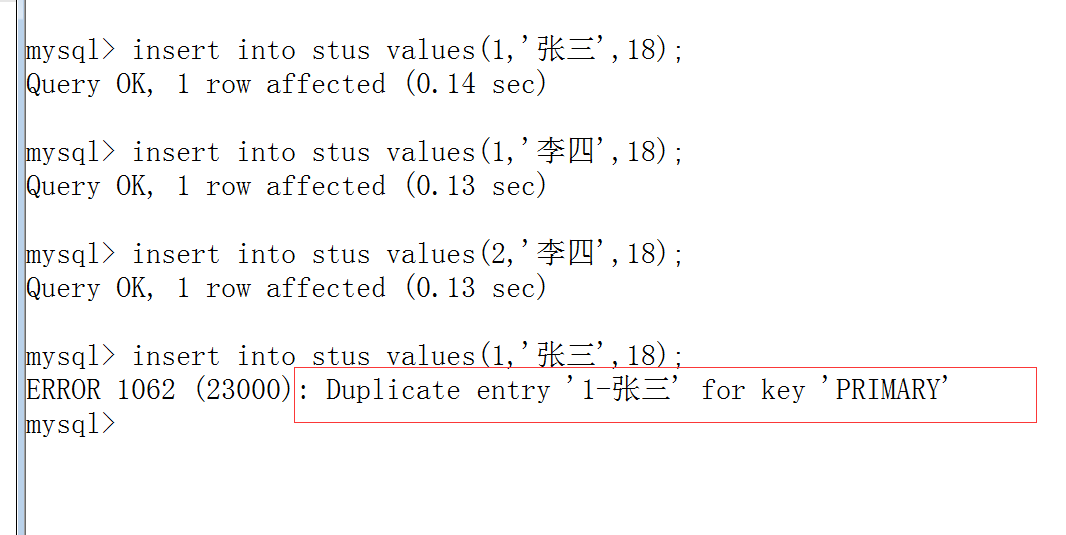


### 多字段主键

该主键是由多个属性(字段)组合而成.单个字段可以重复但是组合后一定不可以重复.在建表时,但给我们设置完所有字段后再来设置组合主键写法是:private key(字段1,字段2,…,字段n);



我们也进行要给简单的测试



## 外键约束

外键是表的一个特殊字段.如果有字段X是表A的属性,且依赖于B表的主键Y.那么表B称之为父表,表A称之为子表,X是表A的外键.通过X字段将表父表B和子表A建立联系.子表A中字段X的所有值都必须是在父表B中出现过的.

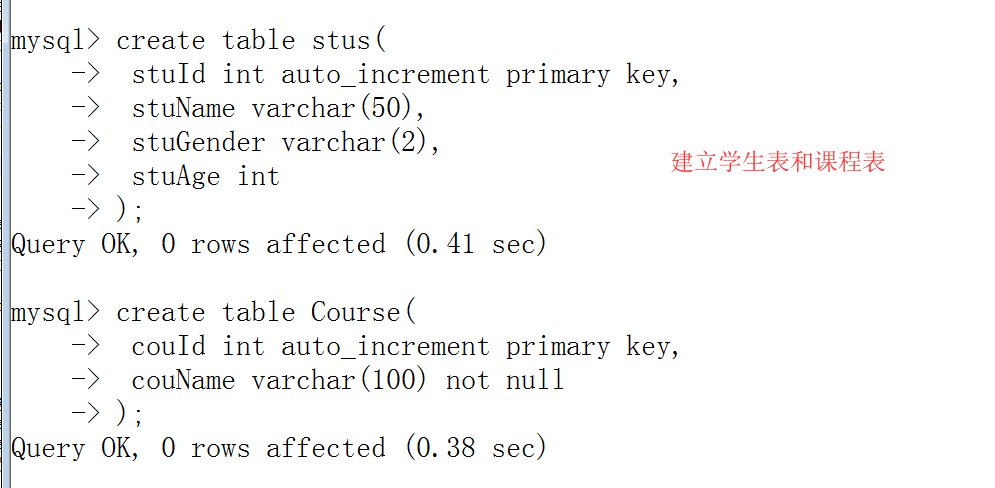
设置外键的原则就是必须依赖数据库中已经存在的父表的主键,外键的值可以为空.

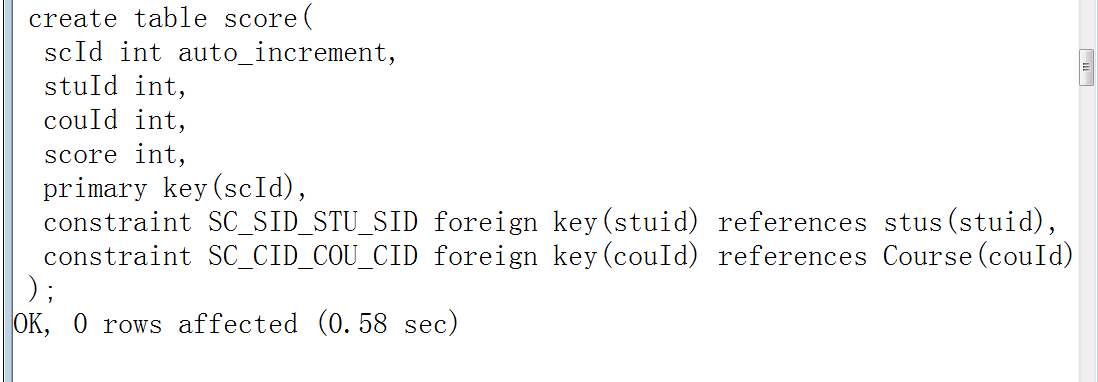
设置外键的目的就是建立子表和父表之间的关联关系.如果父表中删除了某条信息,那么子表中与之对应的信息也要有相应的更改.

设置外键的语法:

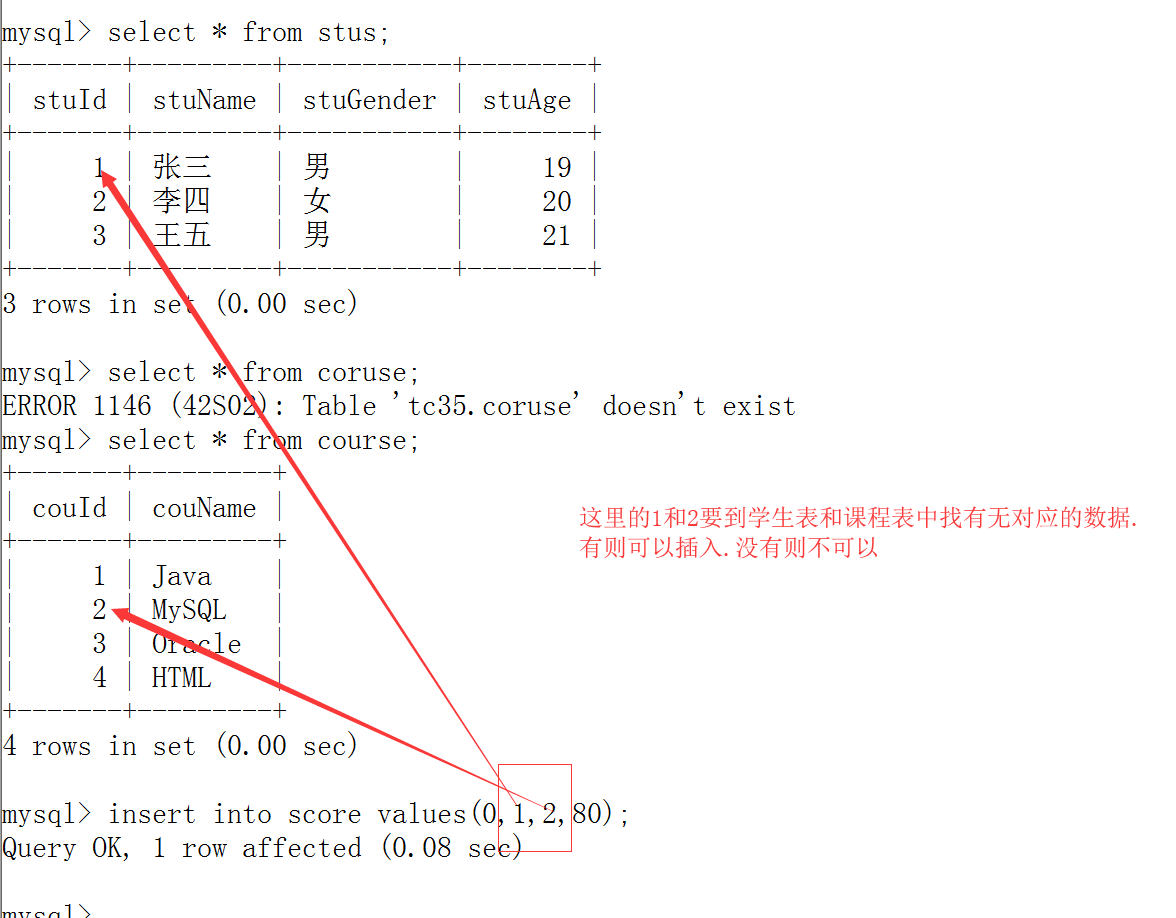
Constraint 外键别名 foreign key(字段1,字段2,…,字段n) references 父表(字段1,字段2,…,字段n);

这里我们建立学生表和成绩表还有课程表.具体的建表语句如下

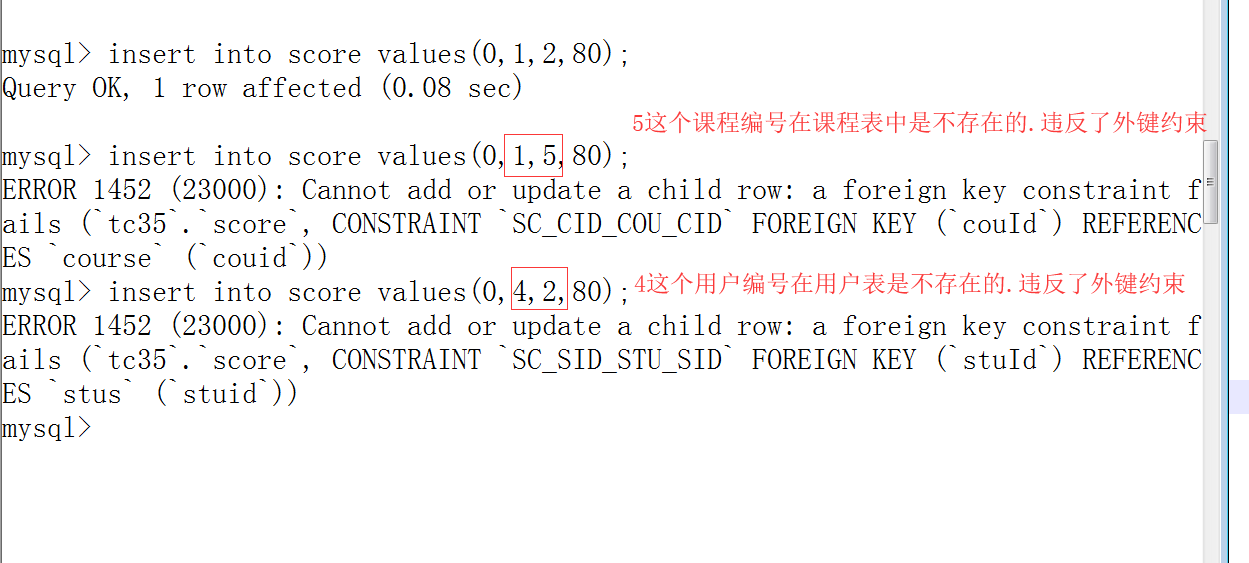




我们进行测试



下面是失败的案例



## 其它完整性约束

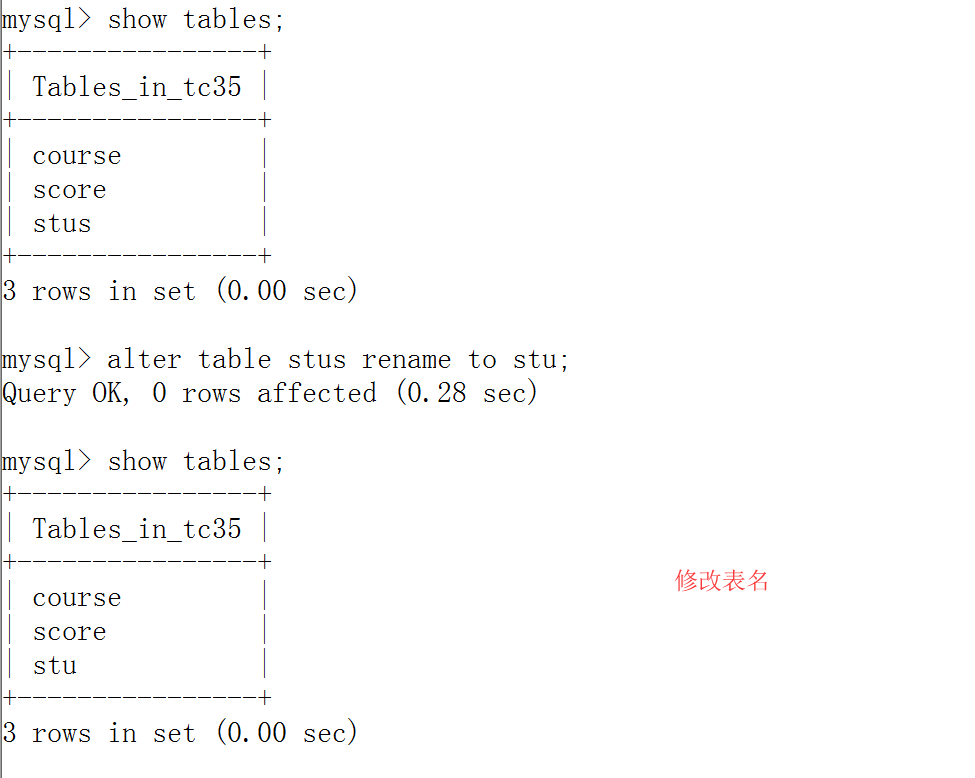
剩下的是就是非空,唯一和默认值.它们的使用比较简单.直接用在列的后面就可以.

# 对表的修改

## 修改表名

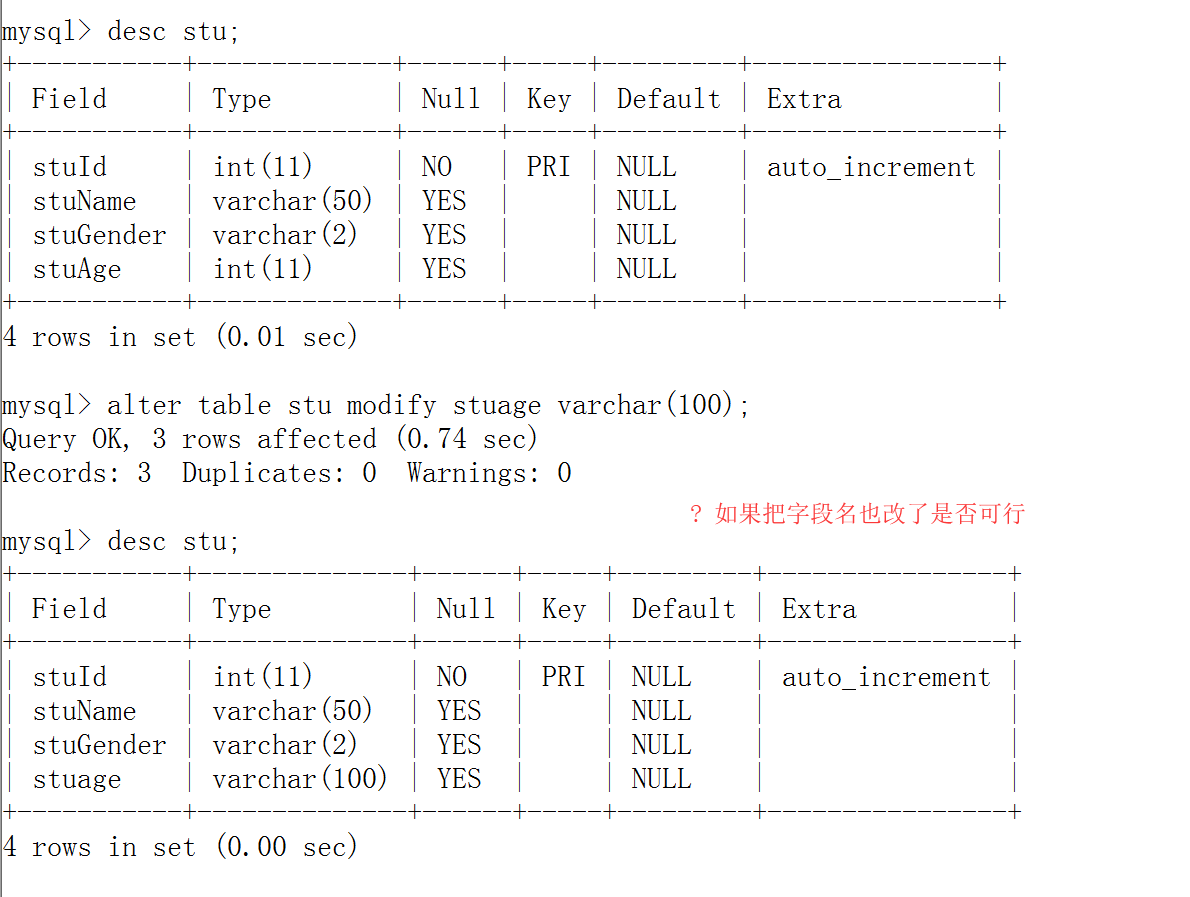
表名是一个库中唯一确定一张表的存在.数据库系统通过表名来区分不同的表.MySQL中所有的修改语句都是通过alter来实现.修改表名的语法是:

Alter table 原表名 rename [to] 新表名;



## 修改字段的数据类型

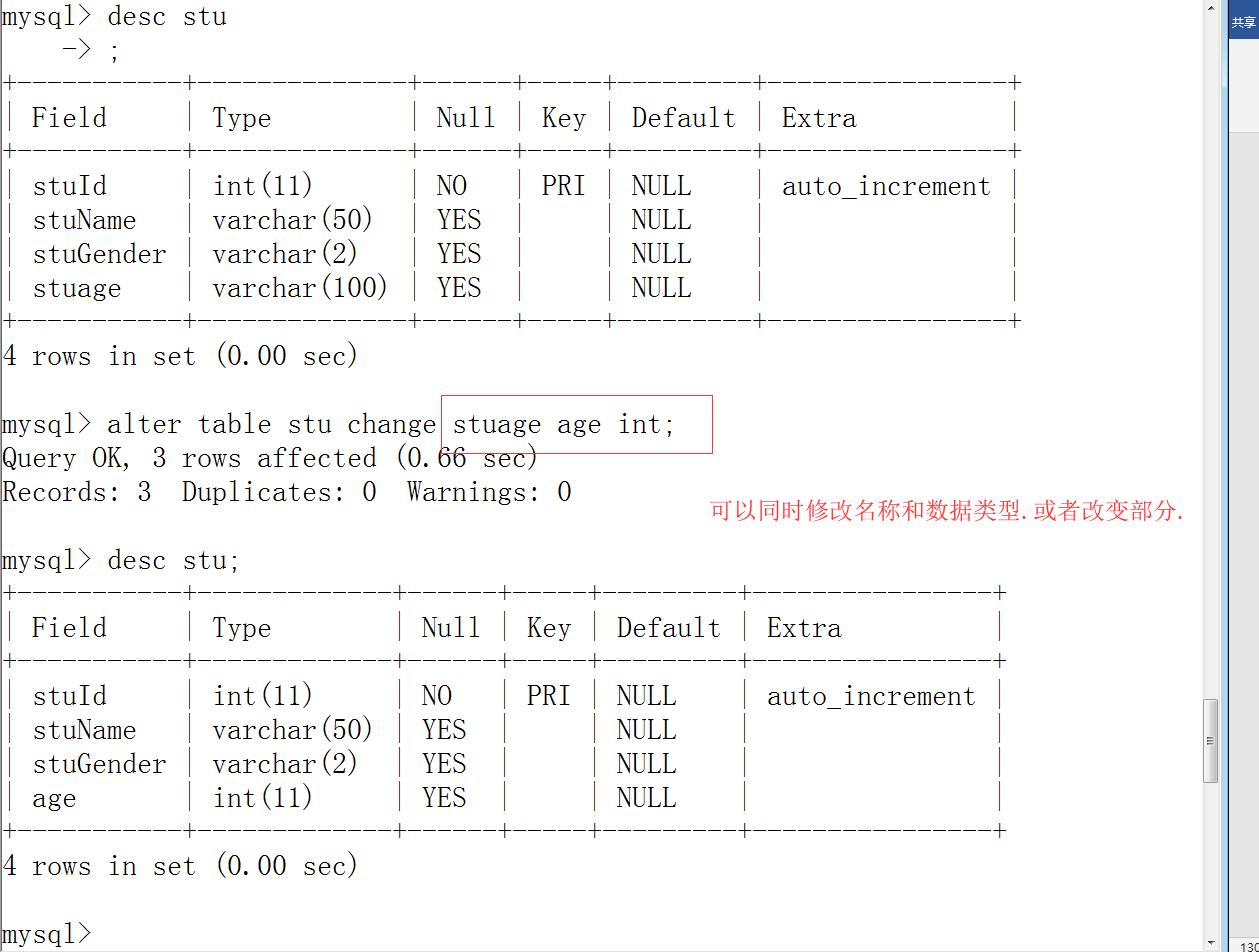
语句:alter table 表名 modify 属性名 新的数据类型



## 修改字段名称

修改字段名称和修改类型类似.或者说使用修改字段名称可以同时修改名称和类型.它的语法格式:

Alter table 表名 change 原字段名 新字段名 新数据类型

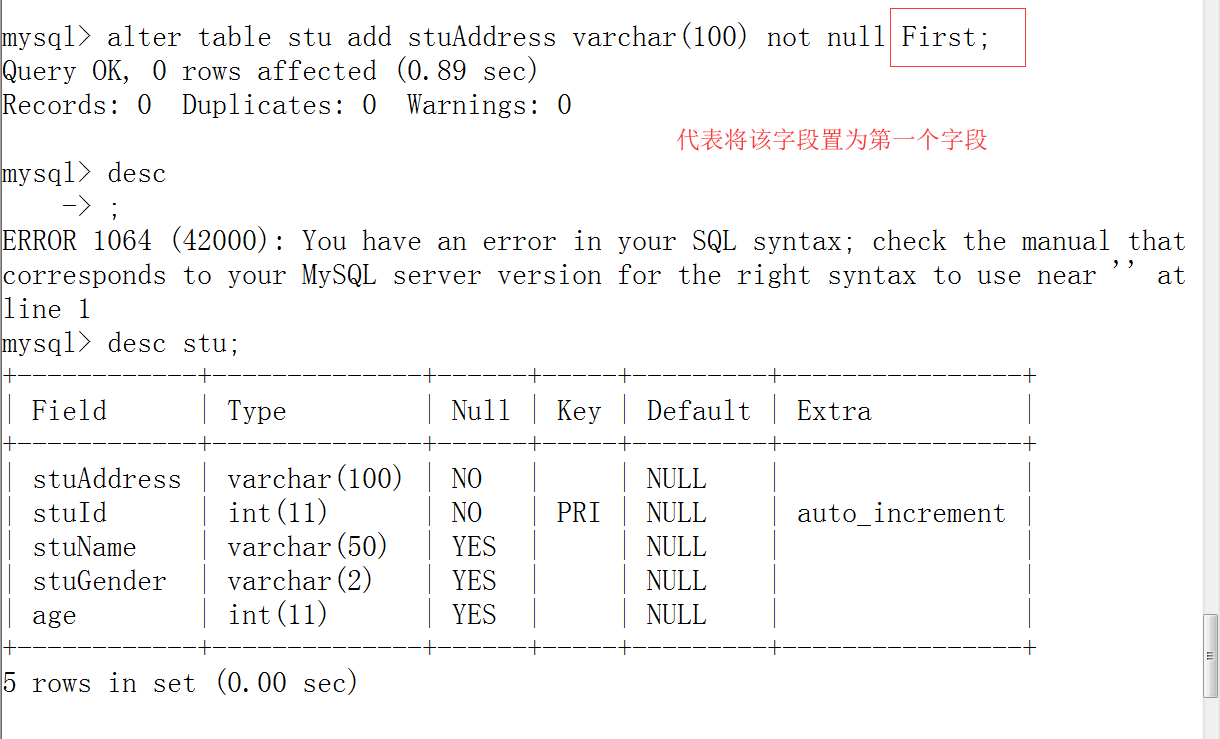


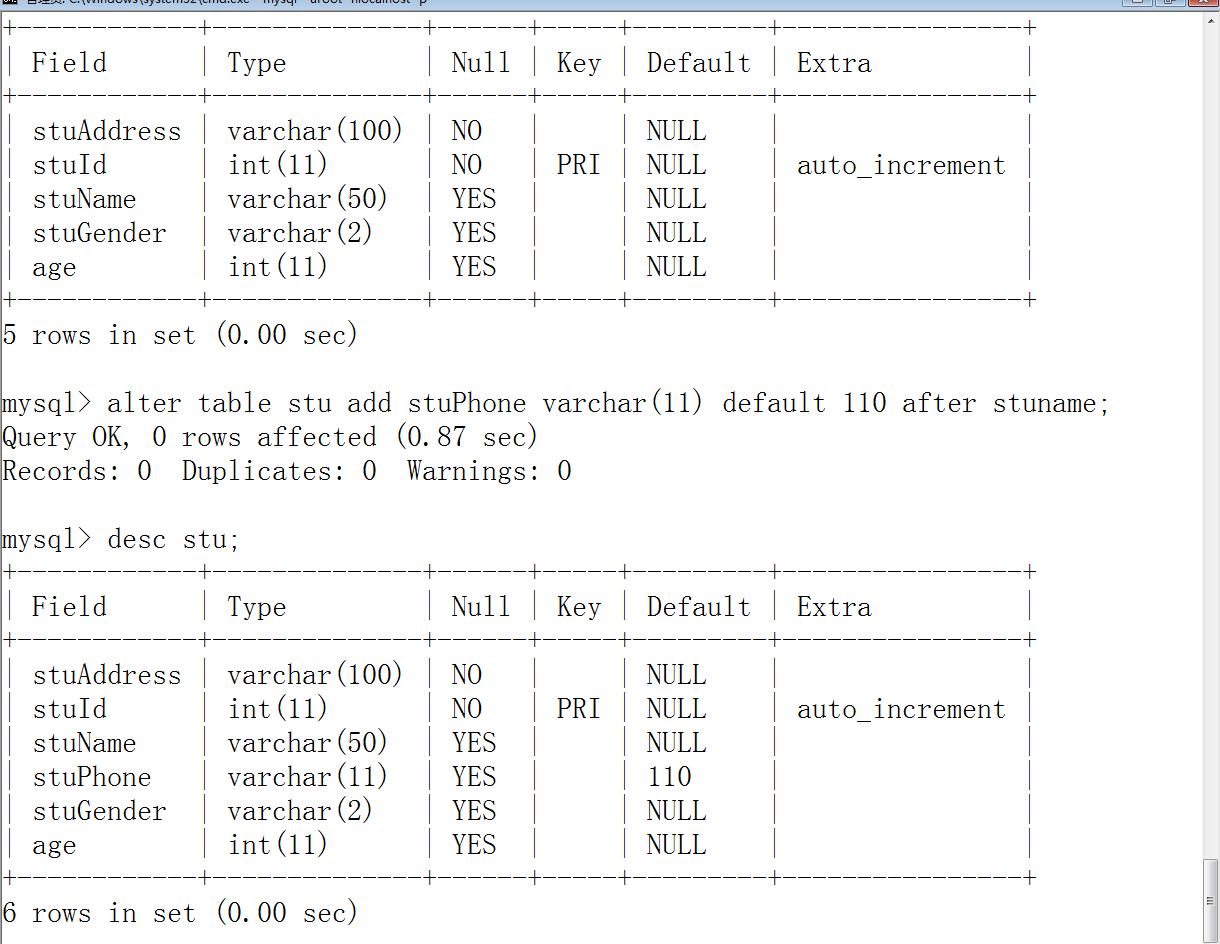
## 增加一个字段

向已有表中添加一个字段.语法格式

Alter table 表名 add 属性名1 数据类型 [完整性约束] [First | After 属性名2];

* 属性名1 是要增加的字段名称.
* 数据类型 是被增加的字段数据类型
* 完整性约束条件 是可选参数,用来设置新增字段的完整性约束
* First 是可选参数 将该字段设置为表的第一个字段
* After 属性名2 是可选参数.将新增的字段放置在属性名2后.

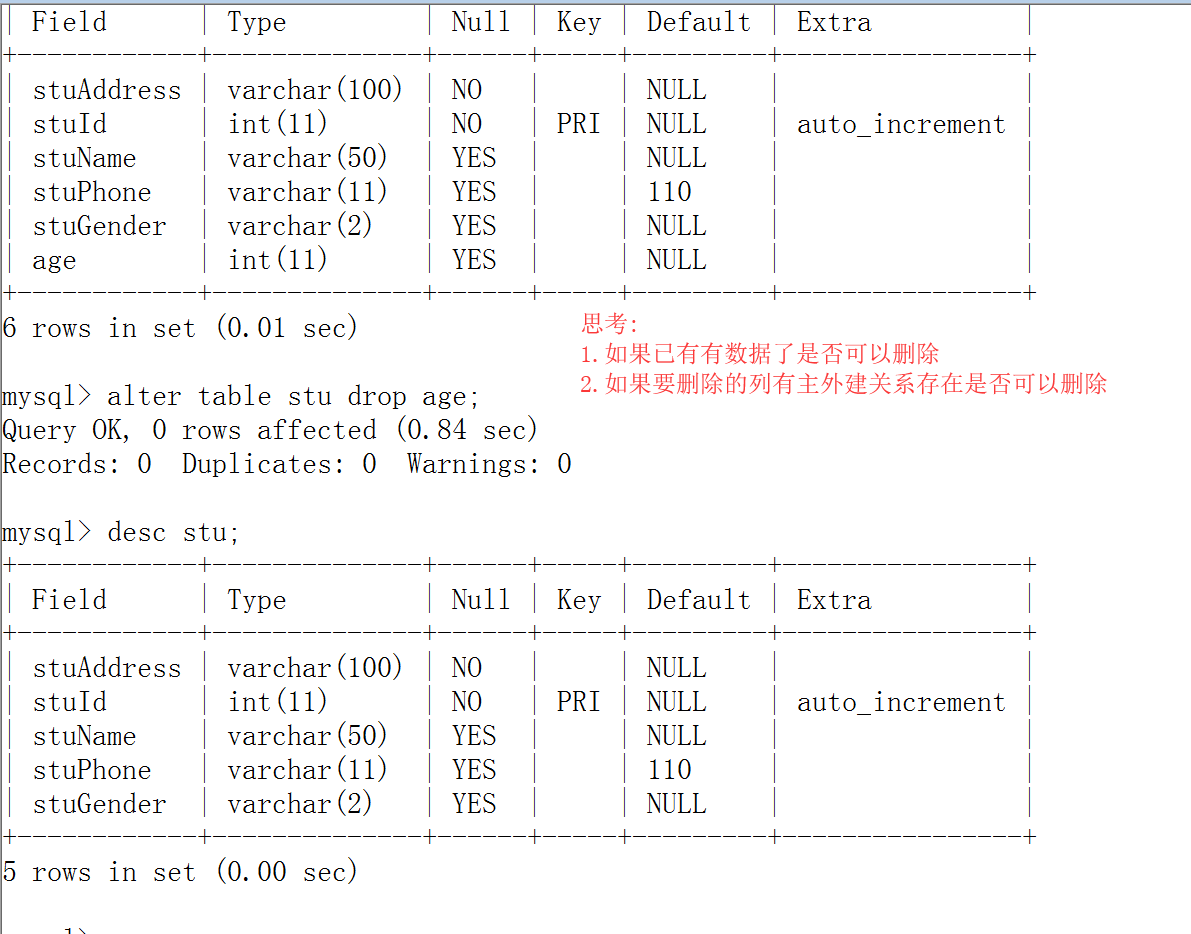




## 从已有表中删除一个字段

删除和增加正好相反.使用的是drop关键字.语法是

Alter table 表名 drop 属性



## 修改表的存储引擎.

我们在建表时是可以指定存储引擎的.同时我们也可以在建表后更改存储引擎:

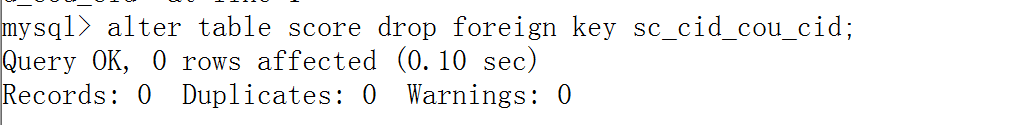
Alter table 表名 engine = 新的存储引擎.

## 删除表的外键约束

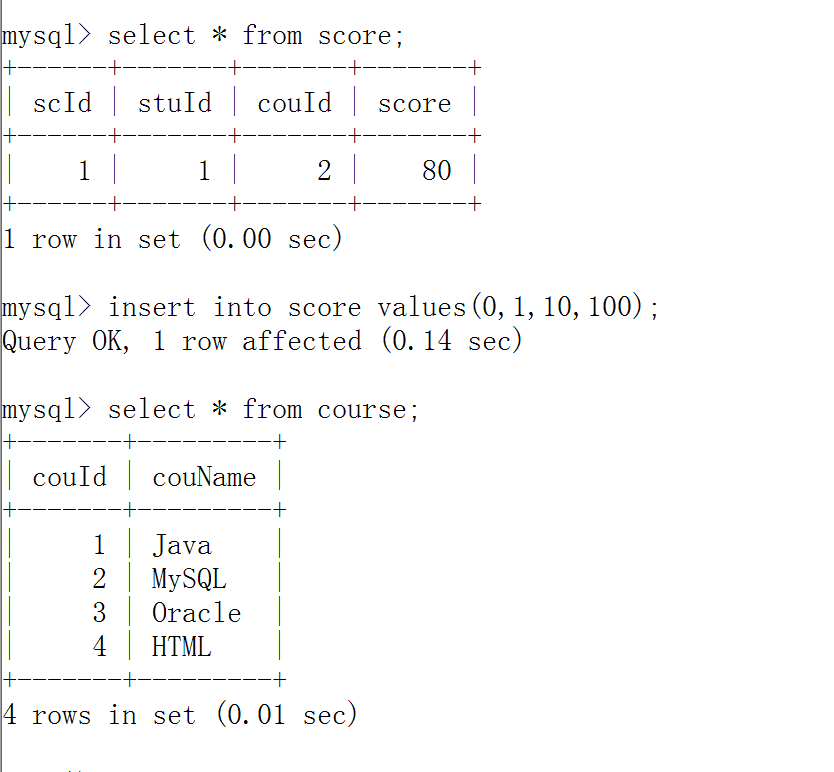
有些时候会因为一些特殊需要,我们要删除已有的省外键约束.它的语法格式

Alter table 表名 drop foreign key 外键别名;

因为这个删除语句.有个问题.一张表可以有几个外键.数量应该是N个.那这些外键的别名可以一样吗?外键别名不可以一样.



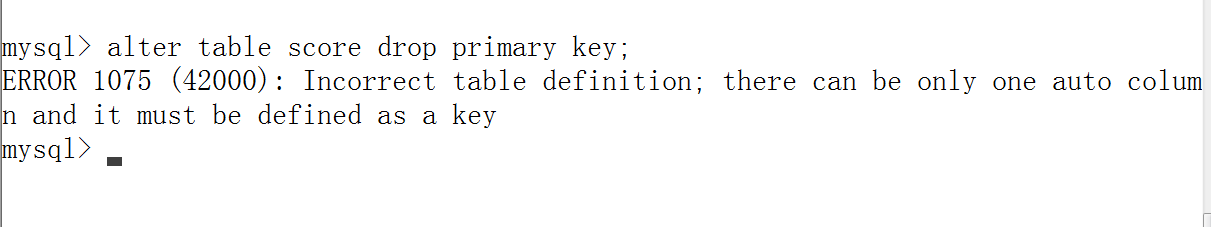
插入一条违反约束的语句进行测试



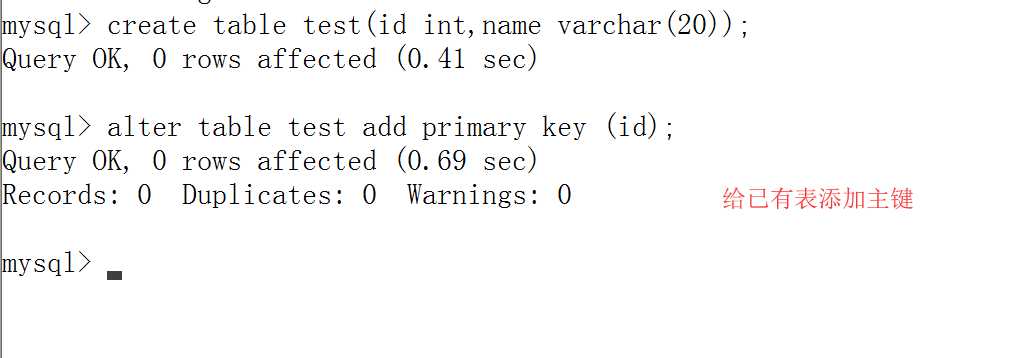
课程编号为10在课程表中明显不存在.但是因为我们把外键删除了,所以这条语句可以插入.

## 删除表的主键

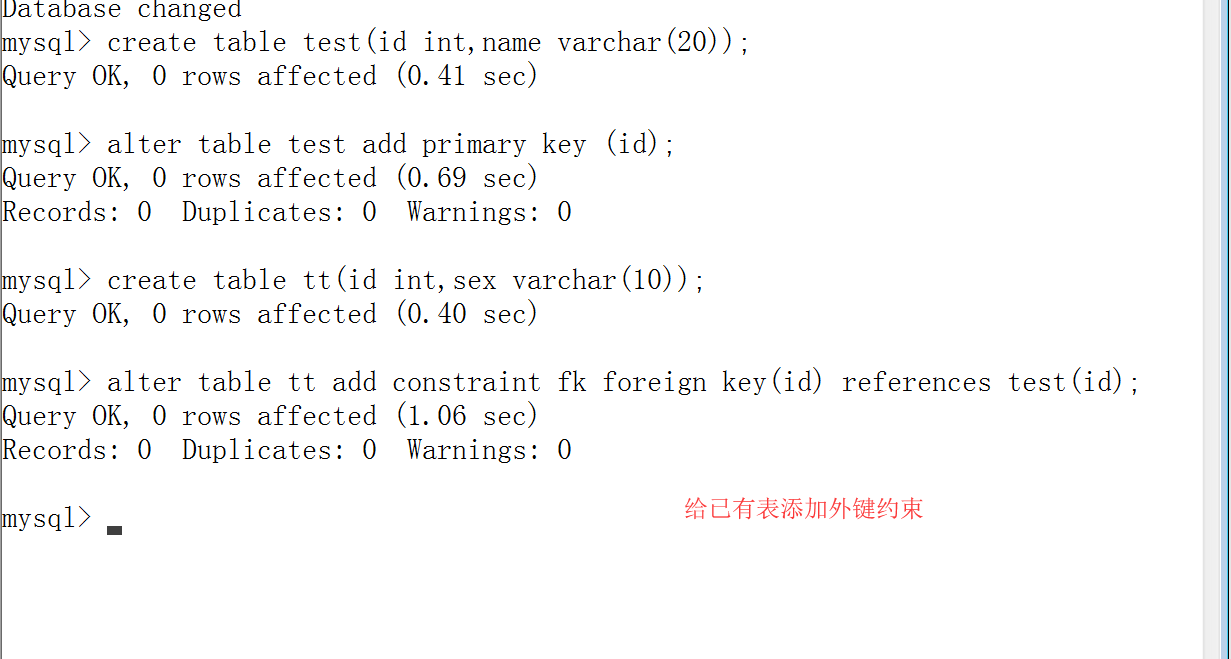
在MySQL中是不允许进行删除主键.



## 对已有表添加主键约束



## 对已有表添加外键约束



# 各种查询语句

查询就是从数据库表中获取所需要的数据,查询是数据库最基本也是最重要的操作.

在所有的SQL中查询都会使用select关键字配合一些子语句来实现各种查询效果.

## 查询语句的基本语法

Select的基本语法:

Select 属性列表 from 表/视图(数据源) [where 条件子句][group by 字段 [having 条件子句]] [order by 字段 [ASC|DESC]];

* 属性列表.需要查询的字段名
* 表/视图 从指定的表或者视图中获取数据,就是数据源.数据源可以有多个
* Where子句 条件
* Group by 分组子句
* Having 筛选 having只能用于group by之后
* Order by 排序子句 ASC或者什么都不写是升序,DESC是降序

## 单表查询

## 聚合函数查询

常见的聚合函数有5个

|  |  |
| --- | --- |
| 函数 | 含义 |
| Count | 汇总个数 |
| Max | 求最大值 |
| Min | 求最小值 |
| Avg | 求平均值 |
| Sum | 求和 |

聚合函数一般都是配合分组语句使用的.

## 多表查询

## 子查询

## 合并查询

## 别名查询

## 正则表达式查询

## Limit限制查询

# Navicat连接MySQL8.0

mysql 8.0 默认使用 caching\_sha2\_password 身份验证机制 —— 从原来的 mysql\_native\_password 更改为 caching\_sha2\_password。   
从 5.7 升级 8.0 版本的不会改变现有用户的身份验证方法，但新用户会默认使用新的 caching\_sha2\_password 。

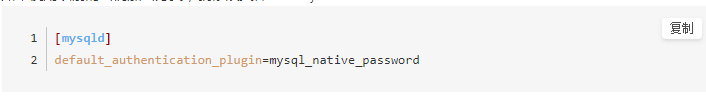
客户端不支持新的加密方式。

方法之一，修改用户的密码和加密方式

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY '123456';

mysql8.\*的新特性 caching\_sha2\_password 密码加密方式

以前版本的mysql密码加密使用的是 mysql\_native\_password  
新添加的用户密码默认使用的 caching\_sha2\_password  
如果在以前mysql基础上升级的 就得用户使用的密码加密使用的是 mysql\_native\_password  
如果使用以前的密码加密方式，就修改文件 /my.ini



修改完成然后记得要刷新FLUSH PRIVILEGES;

