**\*\*\*\*\*\*\*\*\*\*\*MySQL键值（key）\*\*\*\*\*\*\*\*\*\*\*\*\***

设置在字段上，约束如何给字段赋值。

**普通索引：index \***

**主键：primary key \***

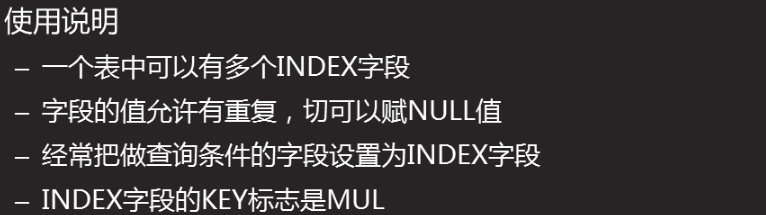
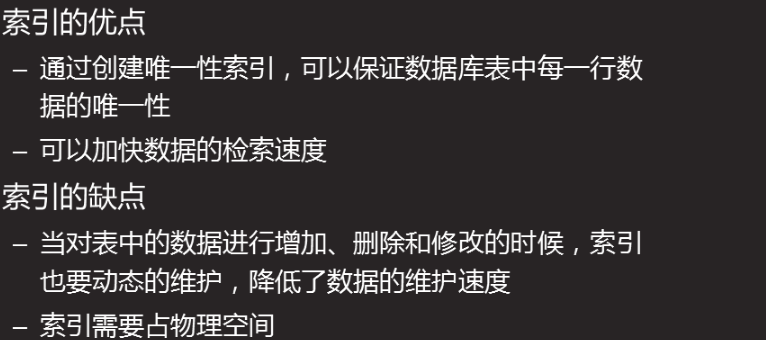
**外键：foreign key \***

唯一索引：unique

全文索引：fulltext

1. **【普通索引 index】什么是索引，及其优缺点？**

相当于书的目录

**1.1查看表中是否由索引字段**

**index普通索引**

**desc 表名；**

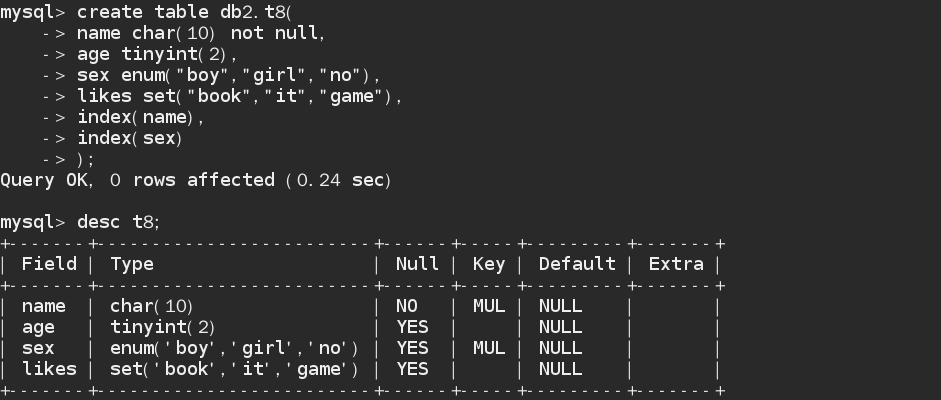
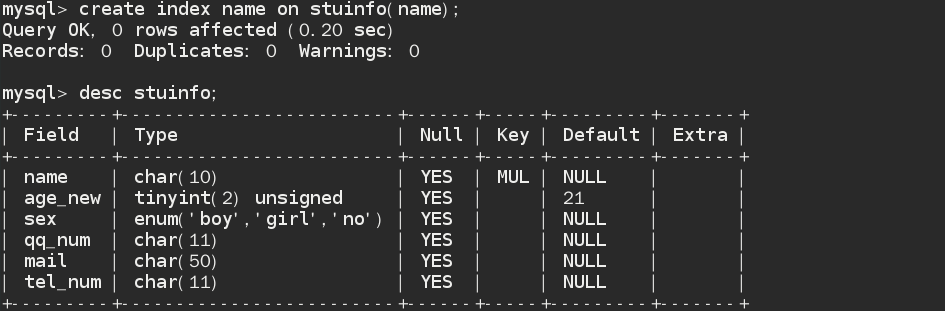
**show index from 表名；**

**1.2创建索引字段**

建表 创建索引字段

把已有表中的字段设置为索引字段

**create index 索引名称 on 表名（字段名）；**

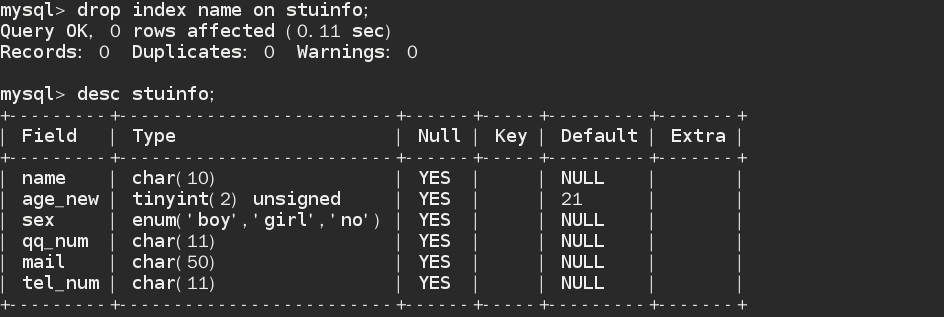


**在已有的表中创建索引字段**

**建表的时候同时创建索引**

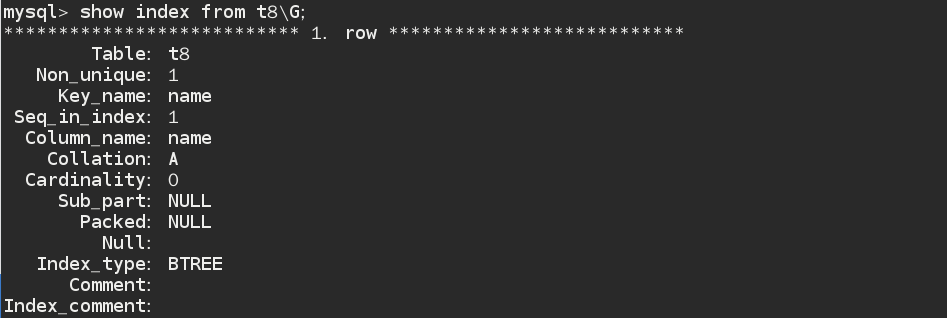
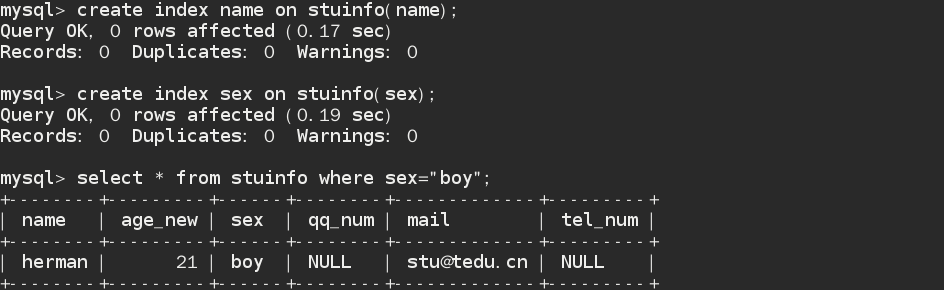
**1.3删除索引字段**

**drop index 索引名 on 表名；**



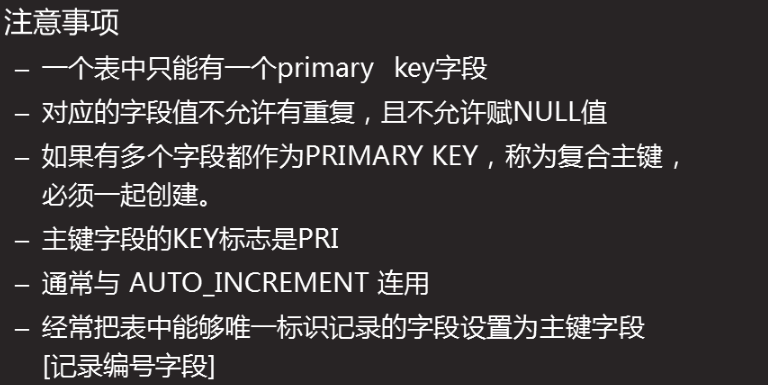
**1.4查询索引**

**select \* from 表名 where 索引名=“想要查询的名称”；**



**B树类型，即二叉树，其他索引类型还有bash，B+TREE等**

1. **【主键：primary key】**



**2.1 建表时创建主键字段**

**create table 表名（**

**字段名 类型 primary key，**

**字段名 类型，**

**……**

**); #复合主键不可这样写**

**create table 表名（**

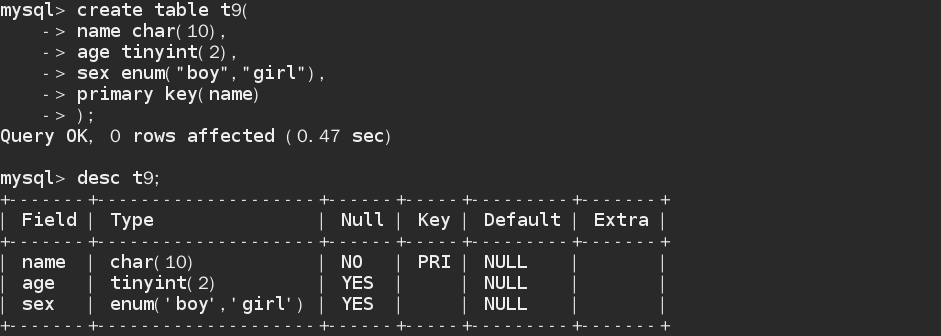
**字段名 类型，**

**字段名 类型，**

**……**

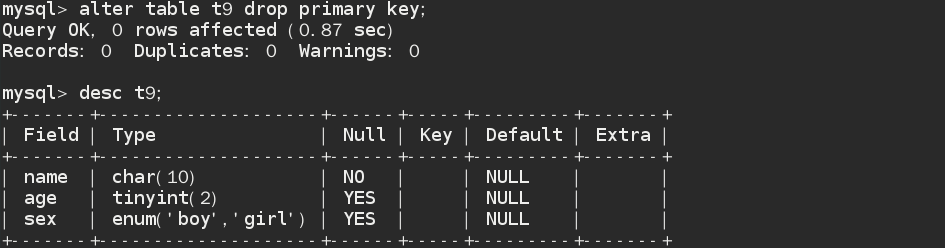
**primary key(name)**

**);**



**2.2 删除主键**

**alter table 库名.表名 drop primary key;**



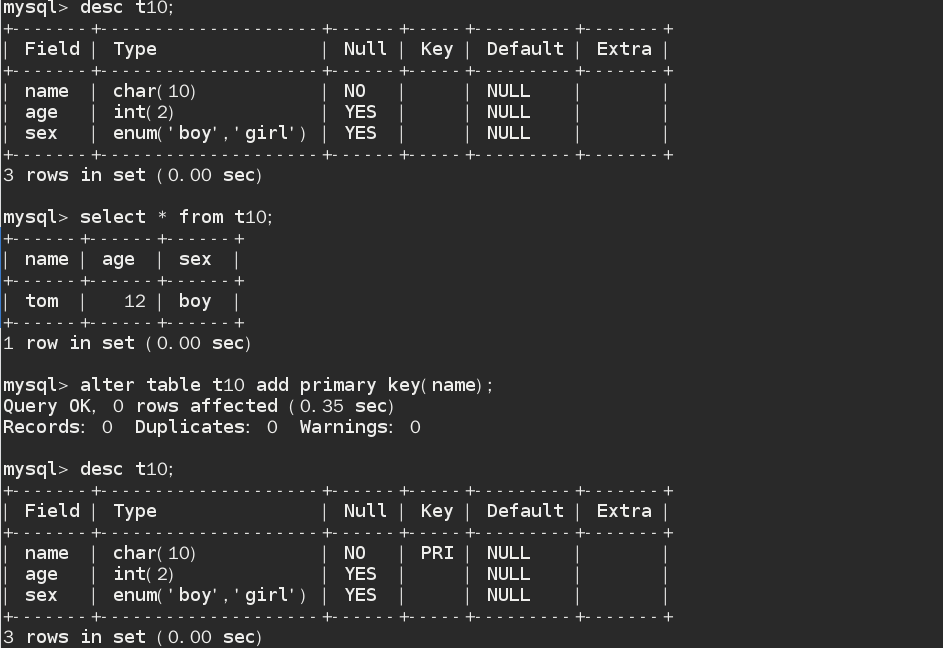


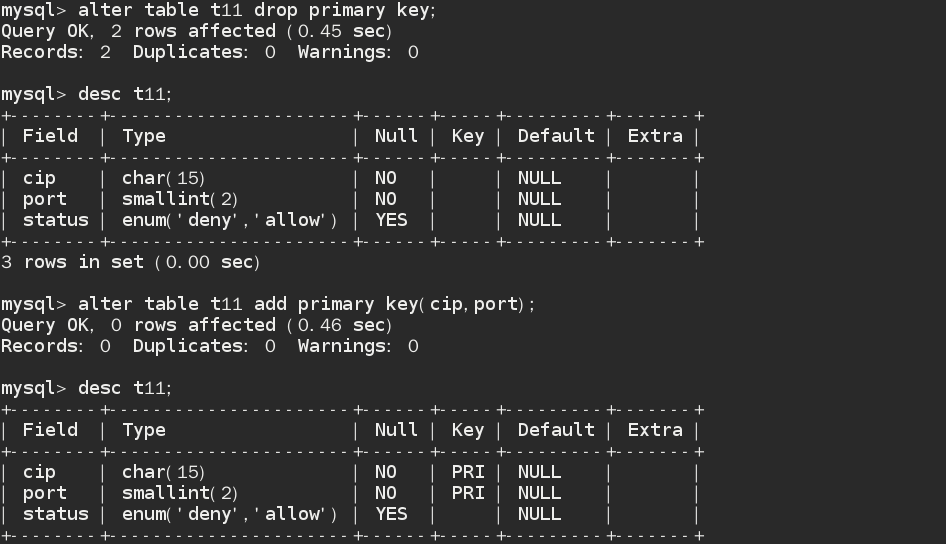
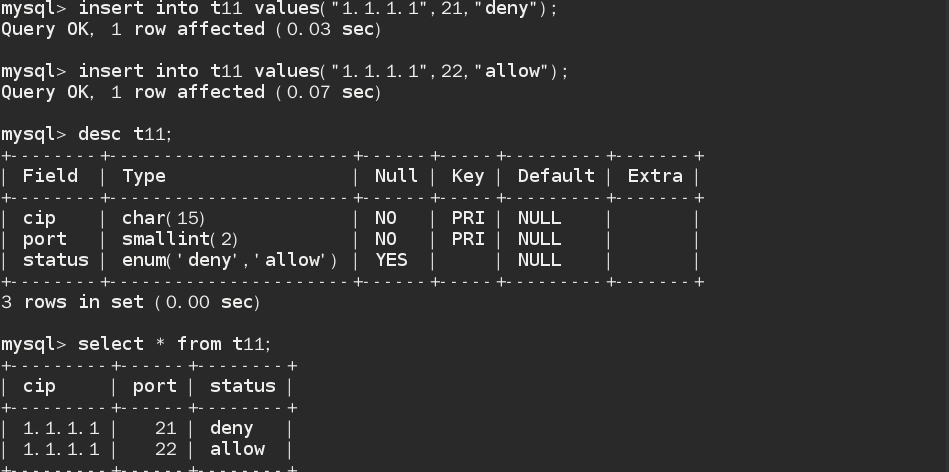
**#验证规则**

**#创建主键之后，对应的字段值不允许重复，不能为null**

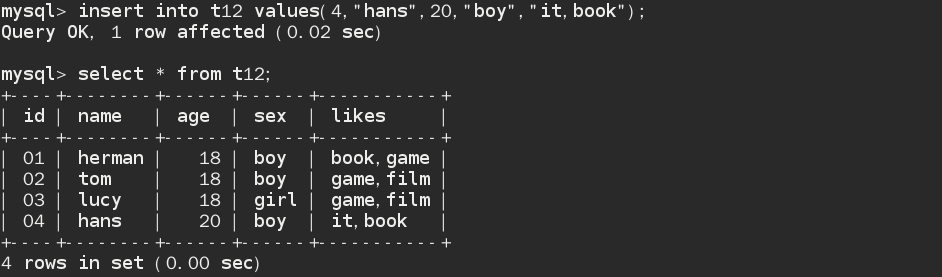
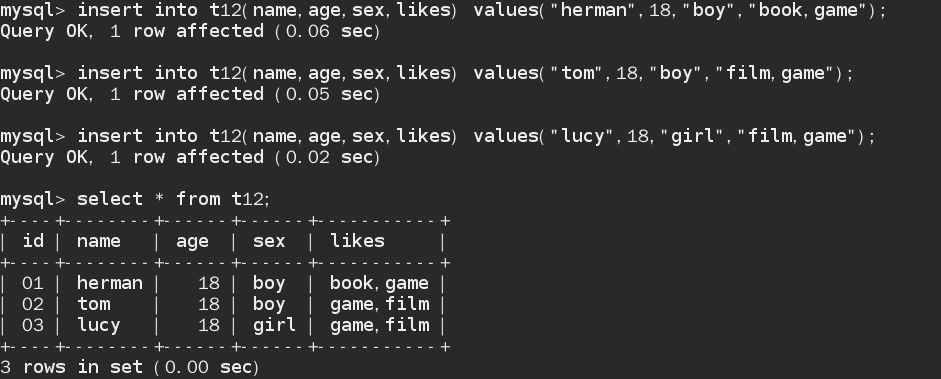
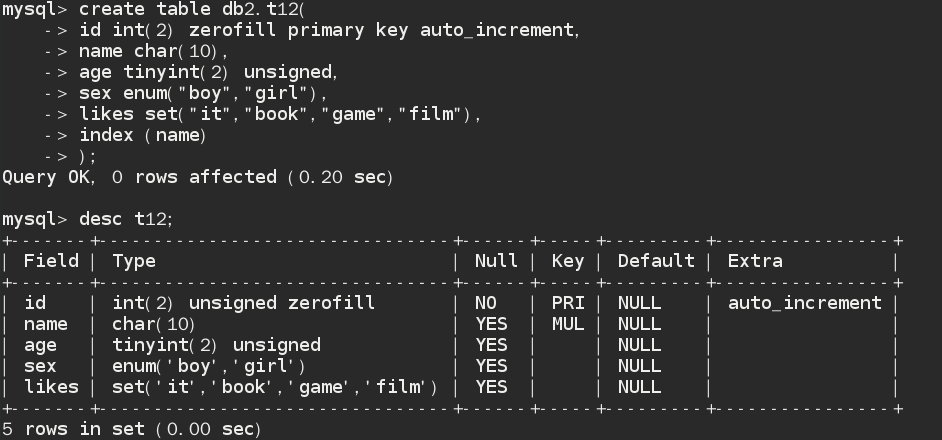
**2.3 把已有表中的字段设置为主键字段**

**alter table 库名.表名 add primary key(字段);**

**2.4 创建复合主键（复合主键字段的值不允许同时相同）（同时创建同时删除）**

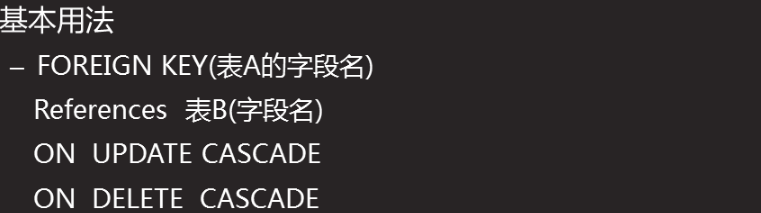
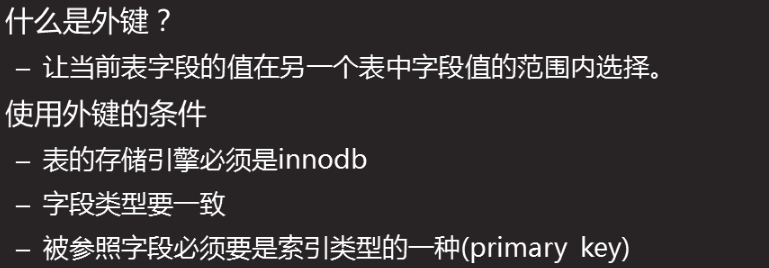


**主键通常与auto\_increment（自动增长，即自增）连用**



**#手动设置，前提是该id之前没有**

**3 【外键：foreign key】让字段的值在指定的范围内选择**



**3.1 创建外键的命令格式**

**create table 库.表(**

**字段列表，**

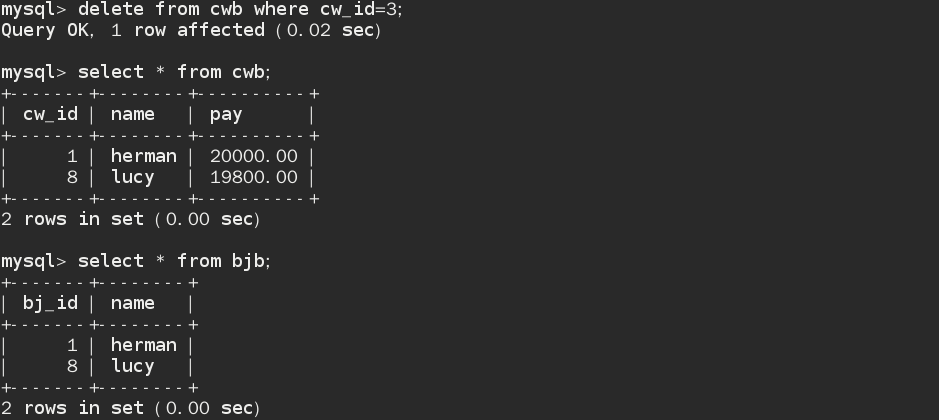
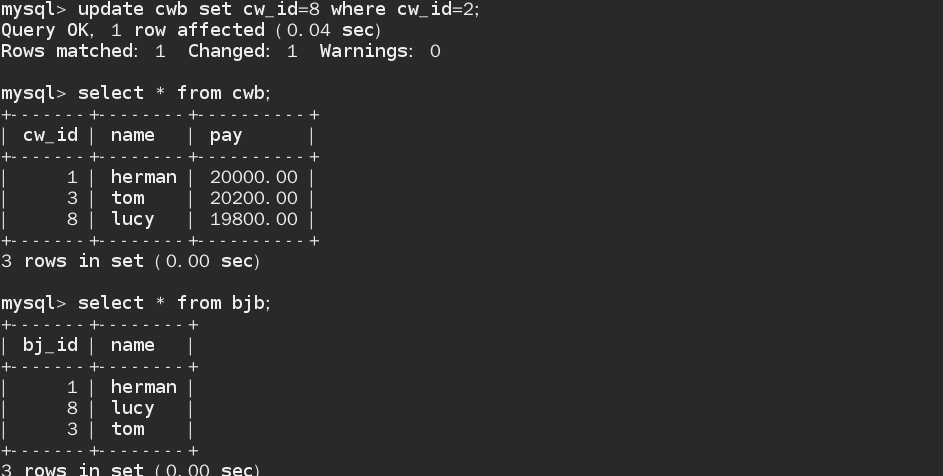
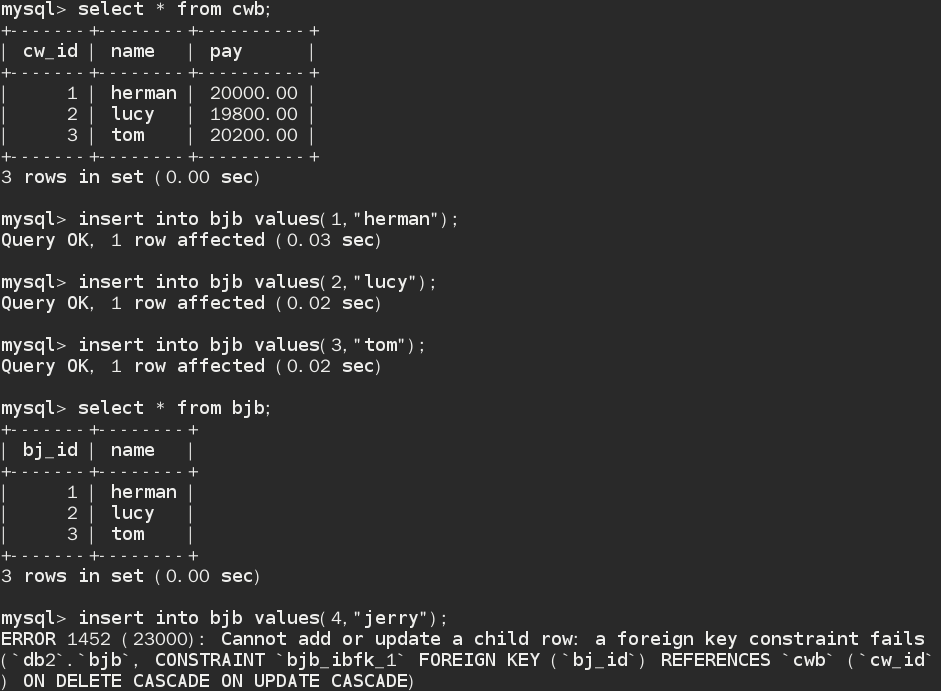
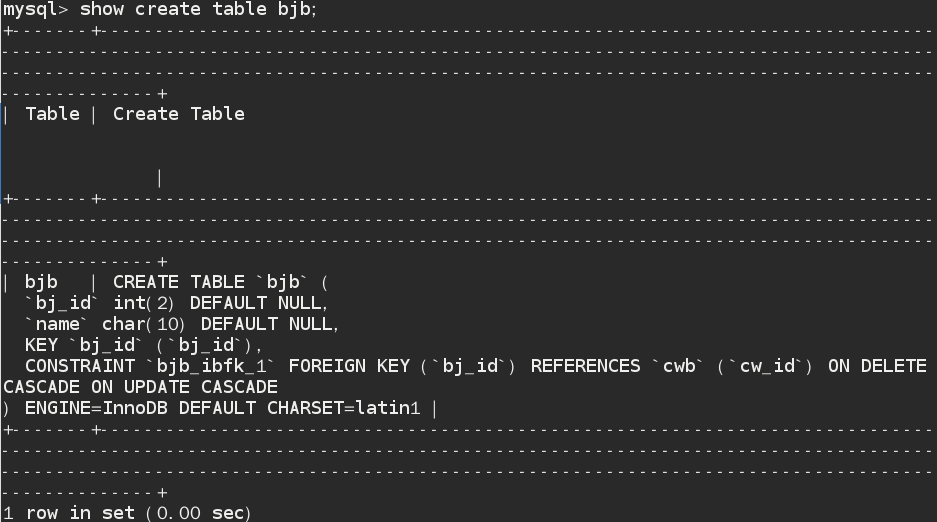
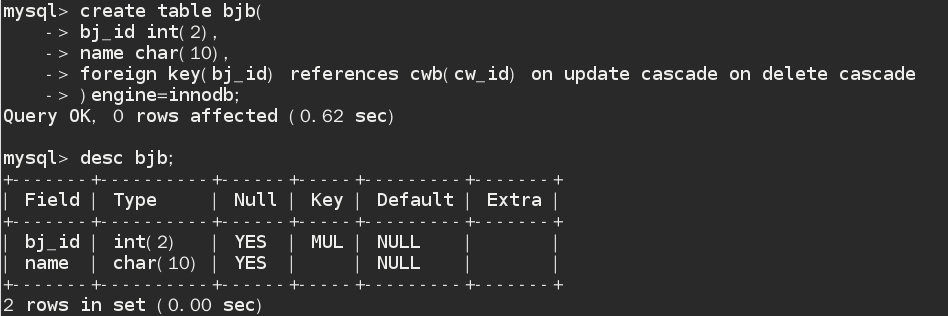
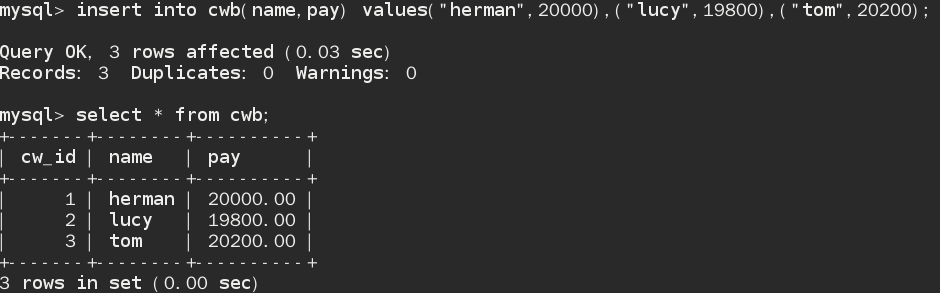
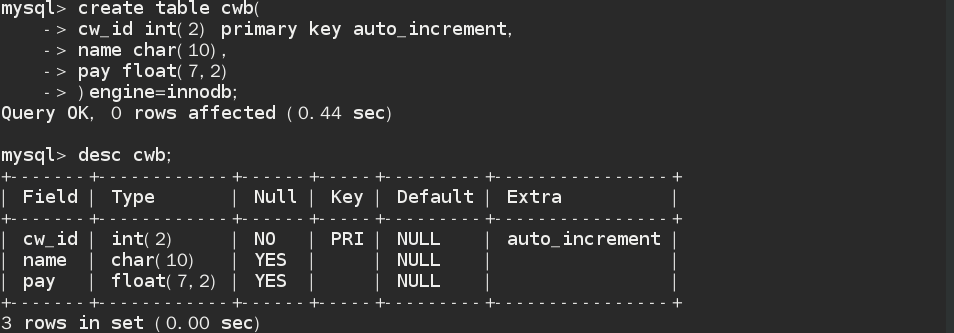
**foreign key(字段名) references 表名（字段名） on update cascade on delete cascade**

**)engine=innodb;**

**#同步删除**

**#同步更新**

MySQL5.7默认的存储引擎是innodb，5.7以前的版本默认的存储引擎不是innodb。



**#此处看不出是否是外键**

**#创建班级表**

**#在财务表添加记录**

**#创建财务表**

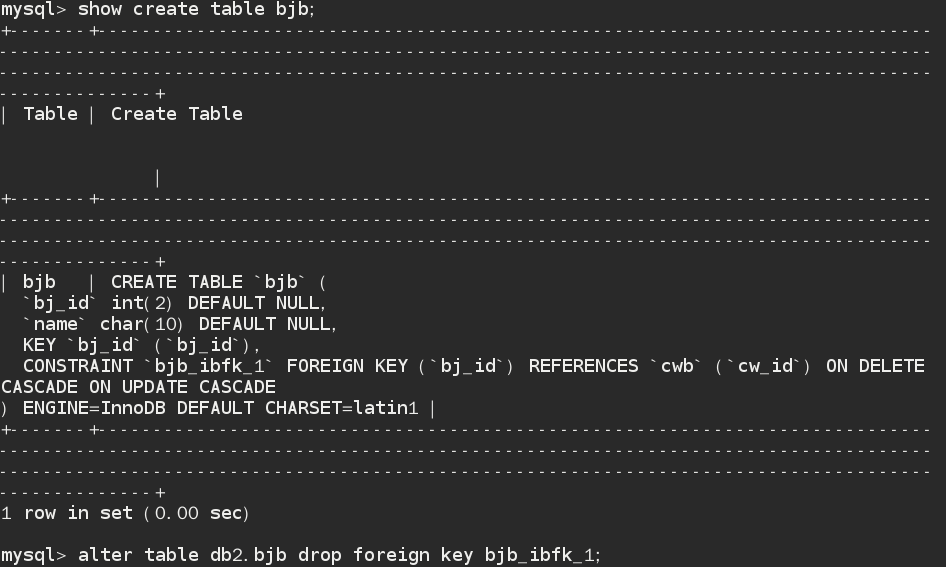
**#让当前字段的值在另一个表中字段值的范围内选择**

**#通过查看建表命令，判断是否时外键**

**#同步删除**

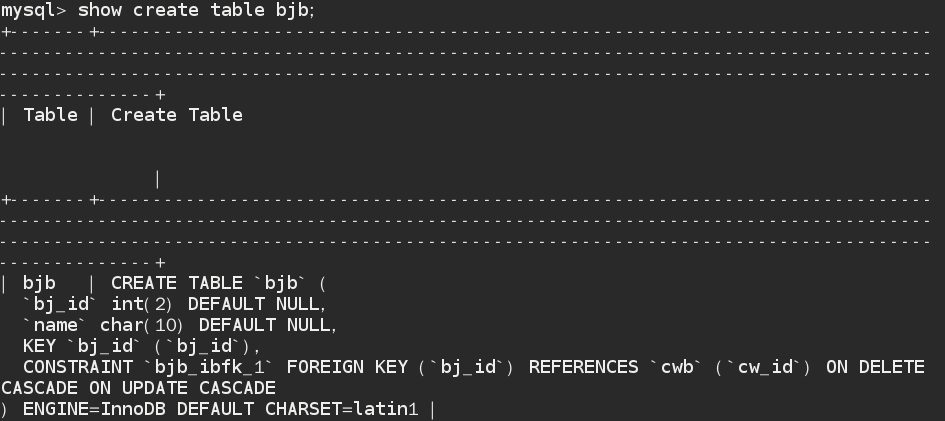
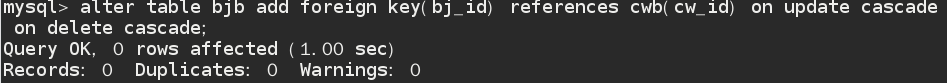
**#同步修改**

**3.2 删除外键**

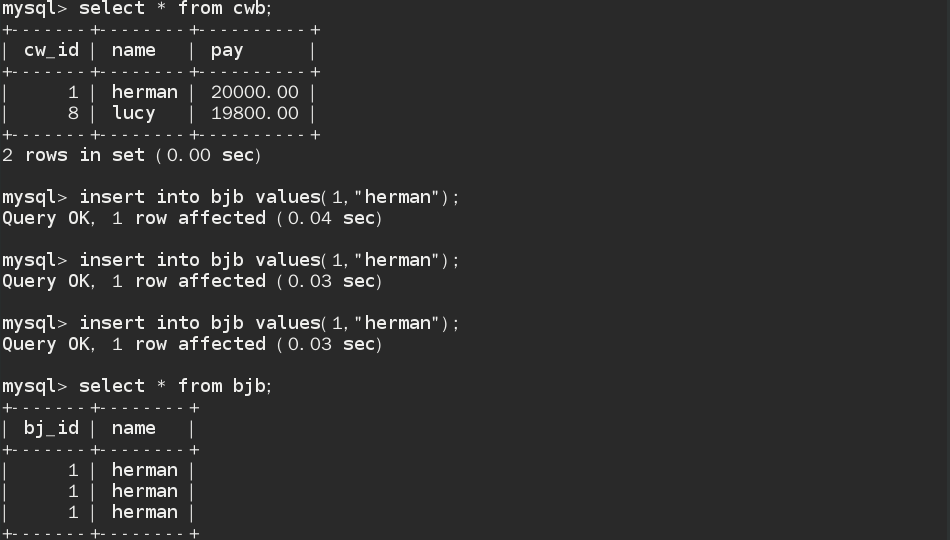


删除外键之后，bjb就不再受cwb的约束了。

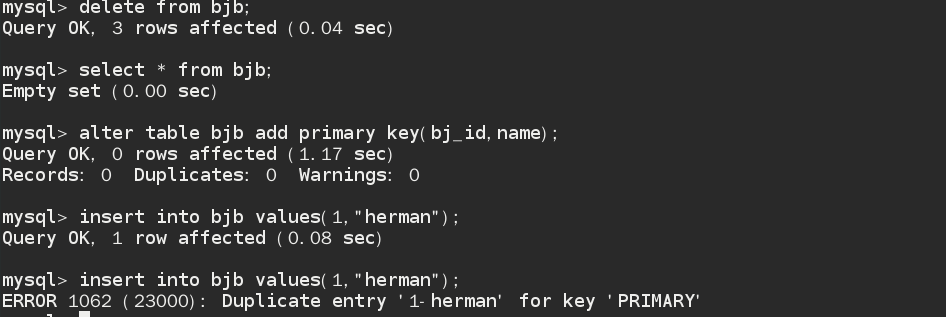
**3.3 在已有表中添加外键字段**



**3.4 优化**



**#只要时父表中有的，不做优化的话，子表可以重复添加**



**为bjb表添加复合主键，来确保不重复**

如果做了同步修改同步删除，并且做了参考，那么父表删除字符列表时，子表也会随之删除，解除参考关系后，子表不受父表的影响了。

**\*\*\*\*\*MySQL存储引擎**

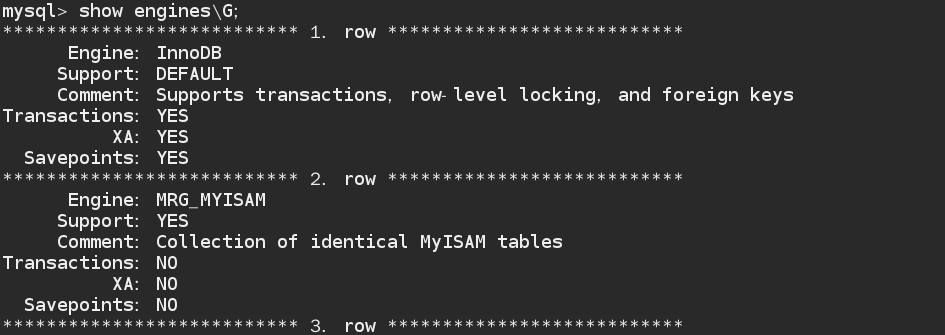
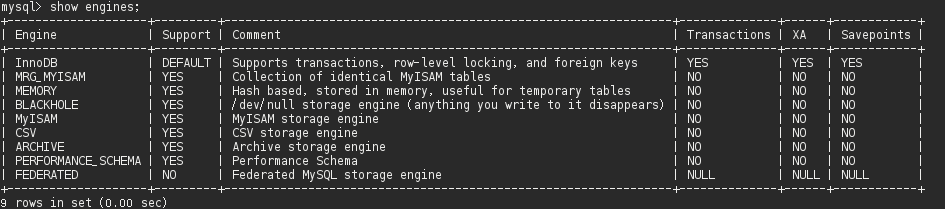
1. **什么是存储引擎**

**是数据库服务软件自带的功能程序，是表的处理器；**

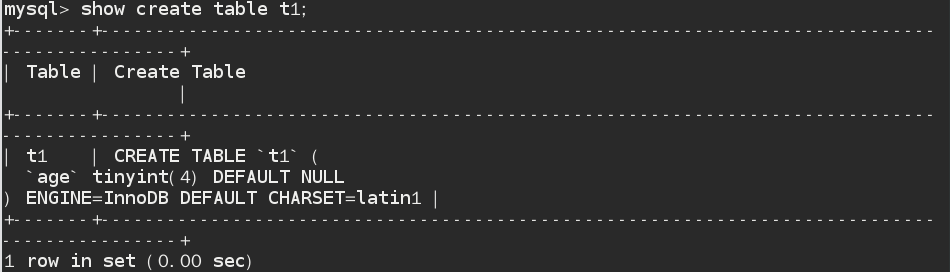
**不同的存储引擎有不同数据存储方式和功能，**

1. **查看数据库服务支持哪些存储引擎**

**show engines;**



1. **查看表使用的存储引擎**



**建表时指定表使用的存储引擎**

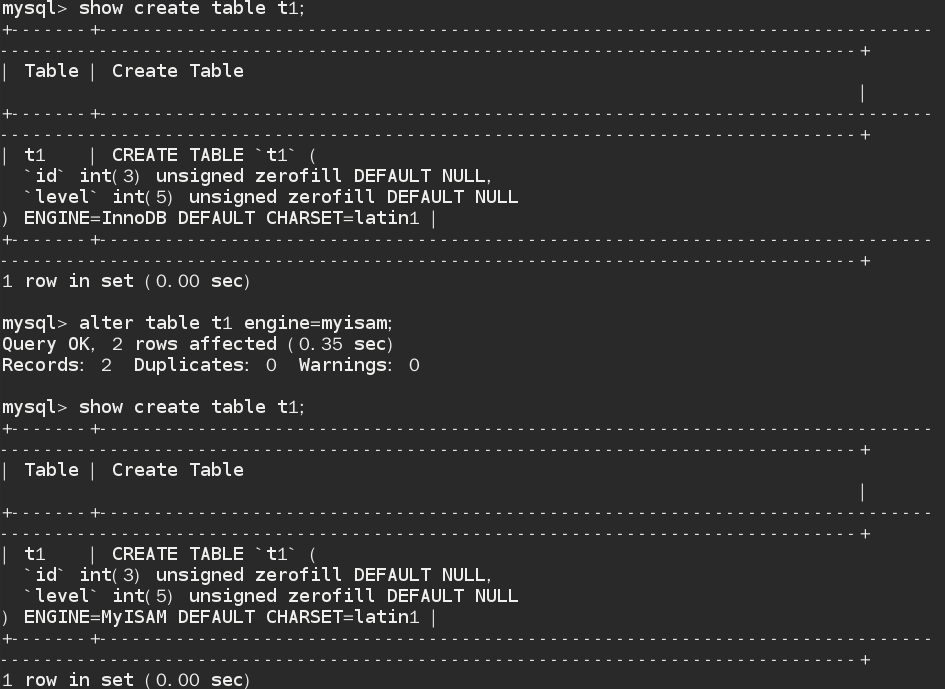
***create table 表名（***

***字段名列表***

***）engine=存储引擎；***

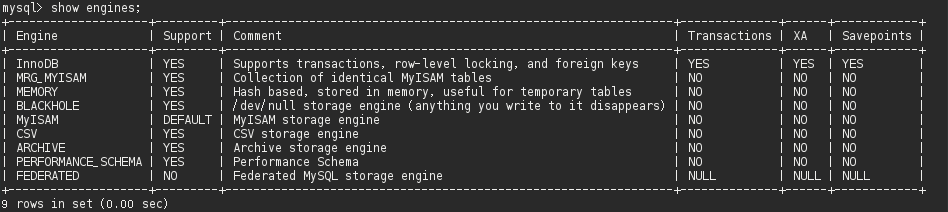
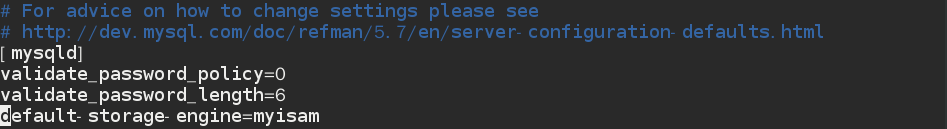
1. **修改服务和表使用的存储引擎**

***alter table 表名 engine=存储引擎；***



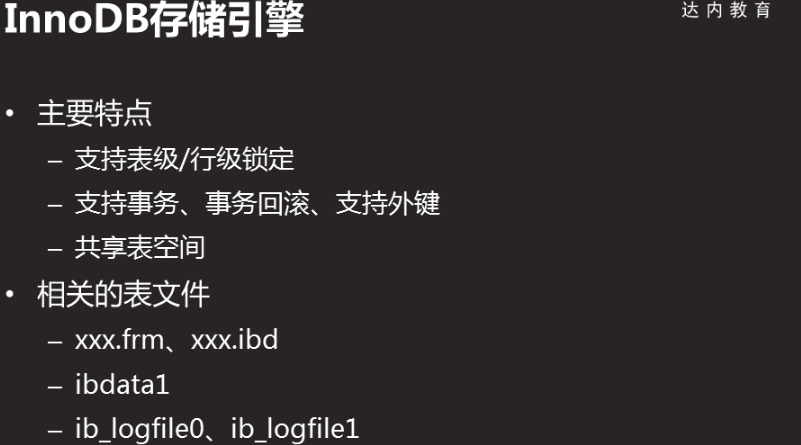
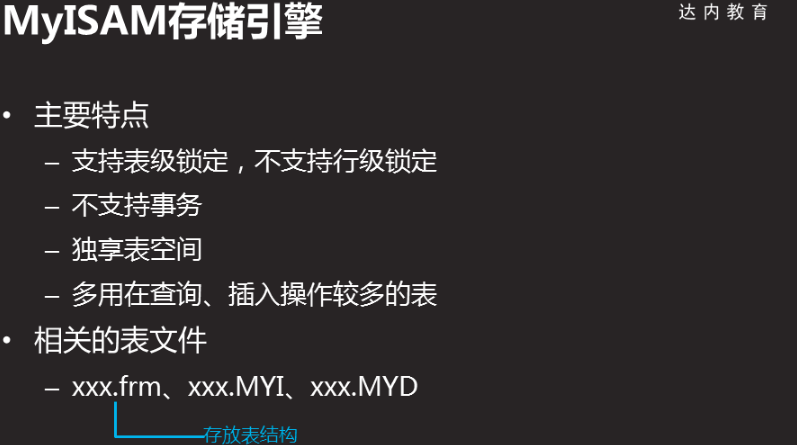
**修改数据库服务默认的存储引擎**





**#systemctl restart mysqld**

1. **生产环境中常用哪种存储引擎，有什么特点**



**支持事务、外键、事务回滚**

**支持行级锁**

**每个表对应两个表文件，分别是：**

**表名.frm 表名.ibd**

**表结构 索引+数据**

**编辑多的表适用**

**事务：一次数据访问从建立连接 操作数据到断开连接的过程叫做事务。**

**事务回滚：一次事务执行过程中，任何一步操作失败都会恢复之前所有的操作。**

**事务日志文件：**

**不支持事务、外键、事务回滚**

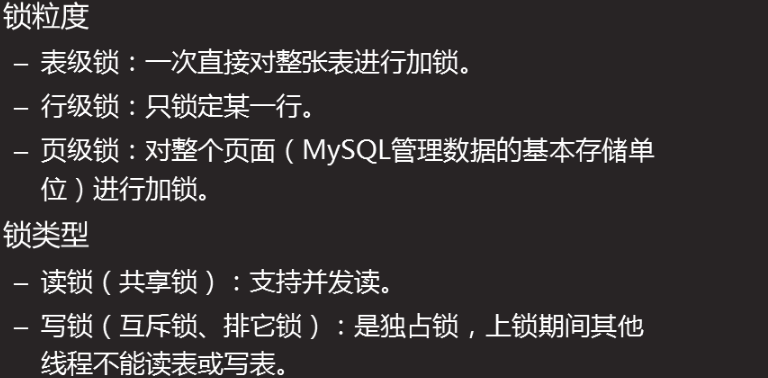
**支持表级锁**

**每个表对应三个表文件，分别是：**

**表名.frm 表名.MYI 表名.MYD**

**表结构 索引 数据**

**查询多的表适用**



**MySQL锁机制**

1. **如何决定表使用哪种存储引擎**

**select -----> myisam**

**insert / update / delete -----> innodb**