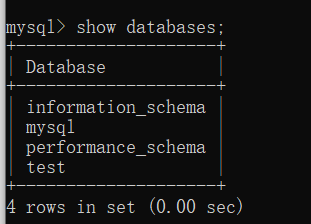
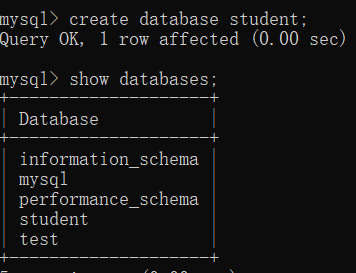
1.查看库

show databases;



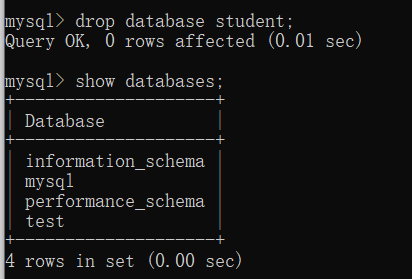
2.创建数据库

create database student;



3.删除数据库

drop database student;

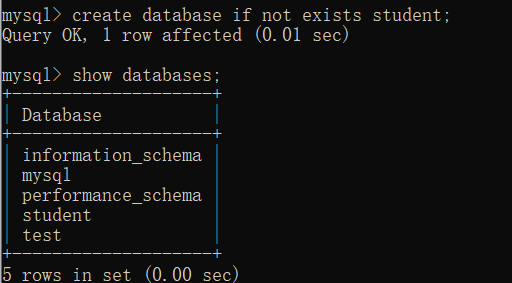


4.通过判断删除

drop database if exists student;

5.判断创建数据库

create database if not exists student;



6.选择数据库

use student;

7.查看库中的表

show tables;

8.mysql 数据类型

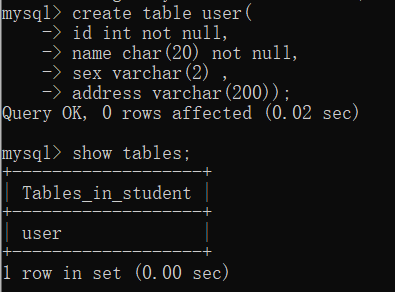
int decimal float double

datetime time date

varchar char text longtext

9.创建 mysql 中的表

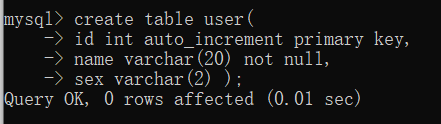
创建表 注意字段为null 或者 not null

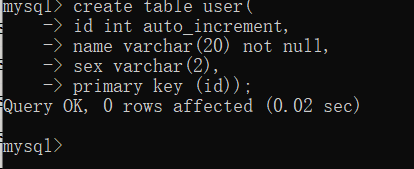


1. 表里面的自增列

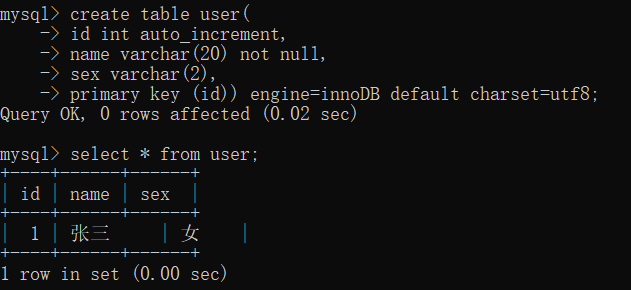
Auto\_increment 设置自增列 一般自增列都为主键 （因为自增列都是唯一的 不能为null 的）

创建表 直接设置某个列为 自增 + 主键





如果乱码 set names gbk;



1. 删除表

Drop table user;

Drop table if exists user;

1. 修改自增列的初值

Alter table user auto\_increment=10000;

1. 表的主键 外键 唯一键   
    主键 值不能为空 值不能重复 一般一个表对应一个主键（也可以设置多个主键）

外键是引用键 外键引用时主键

唯一键 unique 值可以为空 值不能重复

1. 表内添加主键 看上

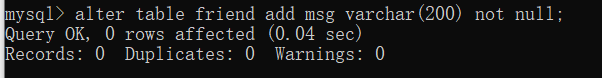
表外添加主键

Alter table friend add primary key (id);

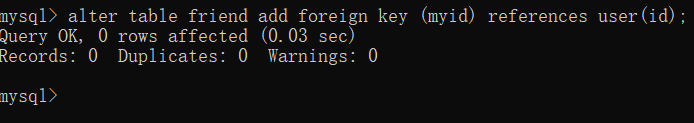
修改id列自增

Alter table friend change id id int auto\_increment;

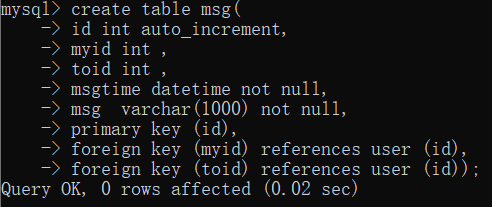
表外添加列



表外添加外键



表内创建 自增 主键 外键



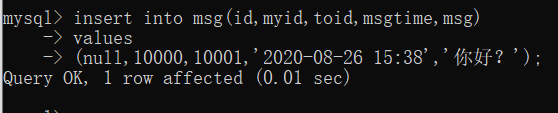
Engine=innoDB default charset=utf8;

1. 数据库常规操作

增删改查

增加：

下面这种写法 (扩号里面 为空的可以不写该列，自增列也可以不写，下面给值上面多少列，下面给多少值，数据类型统一)

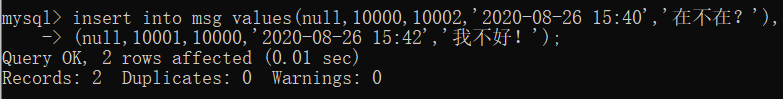


下面这种时表全列式写法 自增列直接给null



一次性增加多条数据

上面两种用法随便使用 多写几组values



查找 ，修改 ，删除 ，需要搭配where条件语句使用

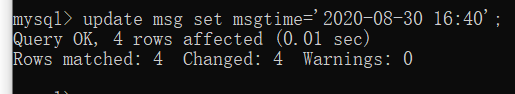
Where 后面写多条件 使用 and 或者 or

使用的符号 > < >= <= != <>

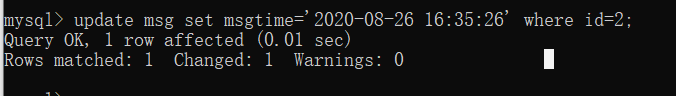
修改：

Update msg set colume=value where \*\*

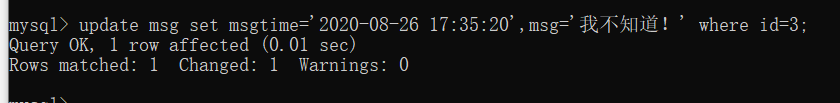
不加where 条件 是全部修改



根据条件修改



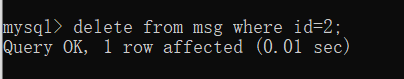
根据条件修改多列



删除语句

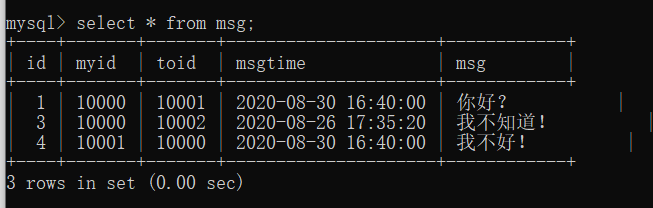
Delete from msg where \*\*\*；

不写条件表的数据全部删除

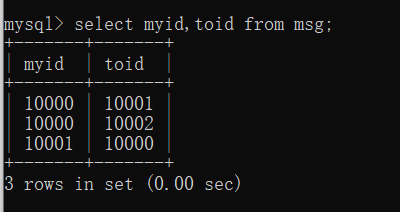


查找语句：(可写条件 可不写)

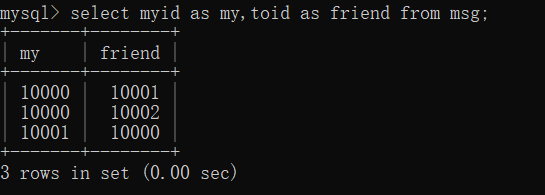
Select \* from msg; \*指所有的列



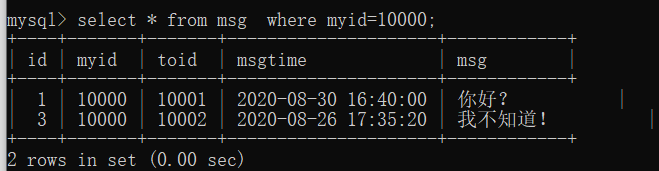
可以根据列查询



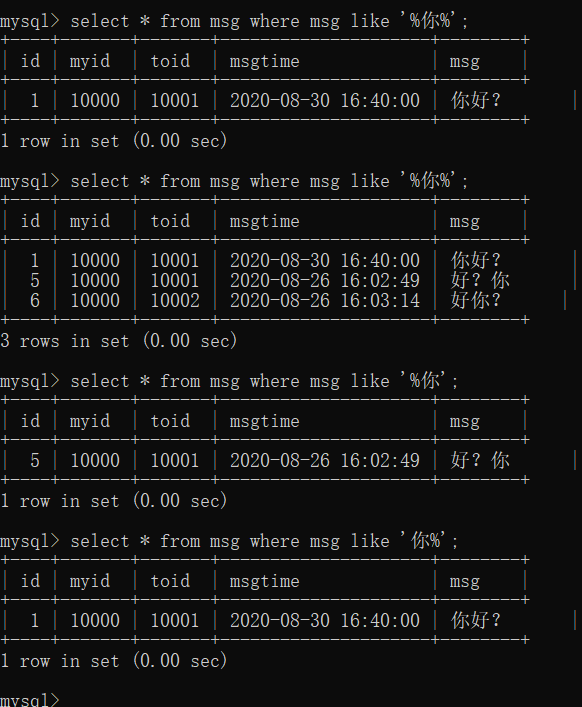
给列起别名 as 关键字



Where 条件查询



模糊查找



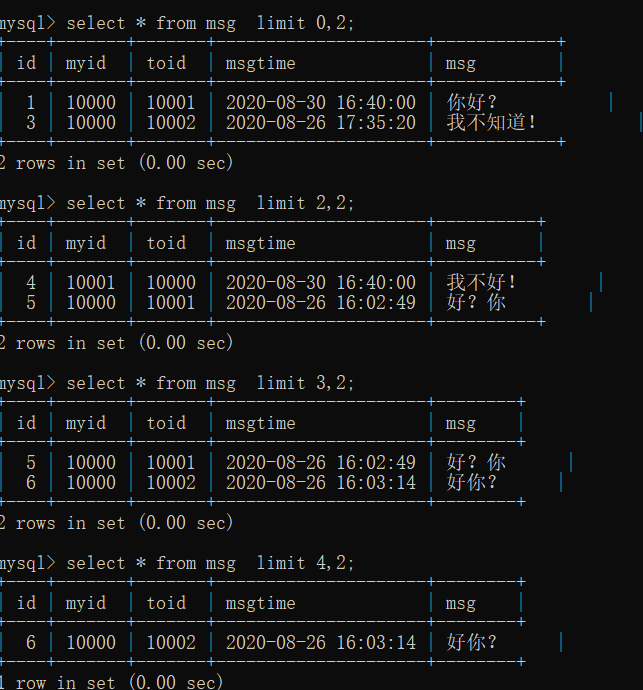
limit查找 可以做数据分页

Limt 0,5 从0开始 取5条

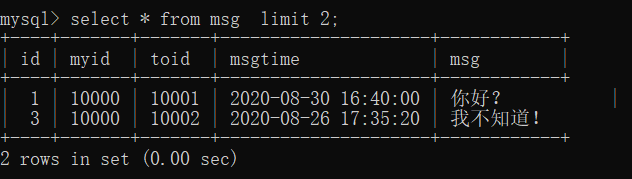
Nowpage=1

Pagenum=5;

(nowpage-1)\*pagenum

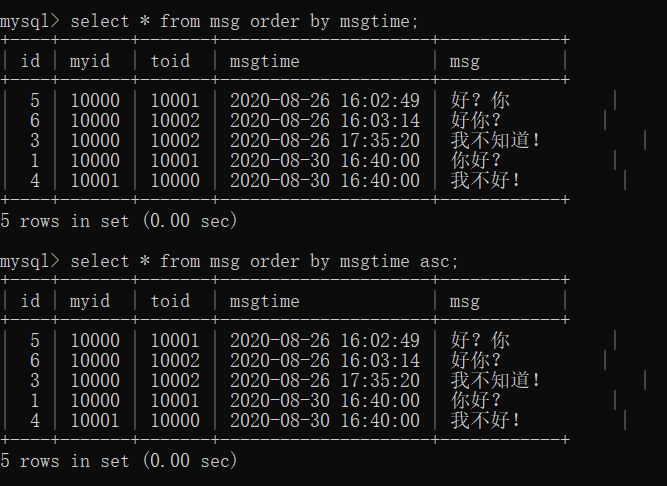


Limit 5 写一个参数 取前几条

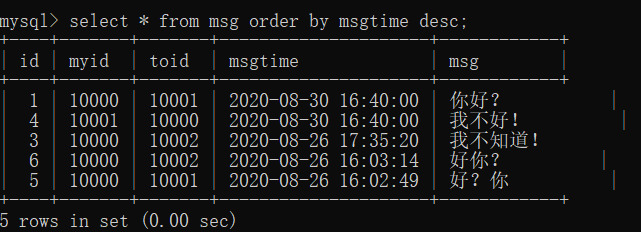


查询的排序 orderby 倒序 desc 升序 asc （默认）

根据列升序排序



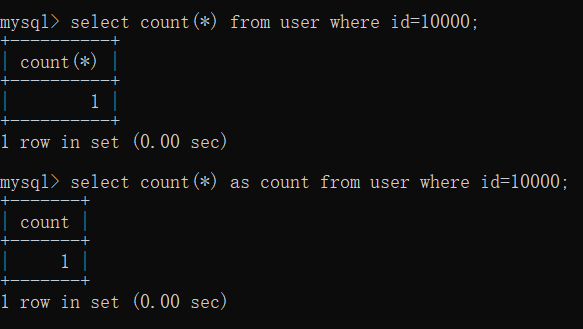
根据列倒序



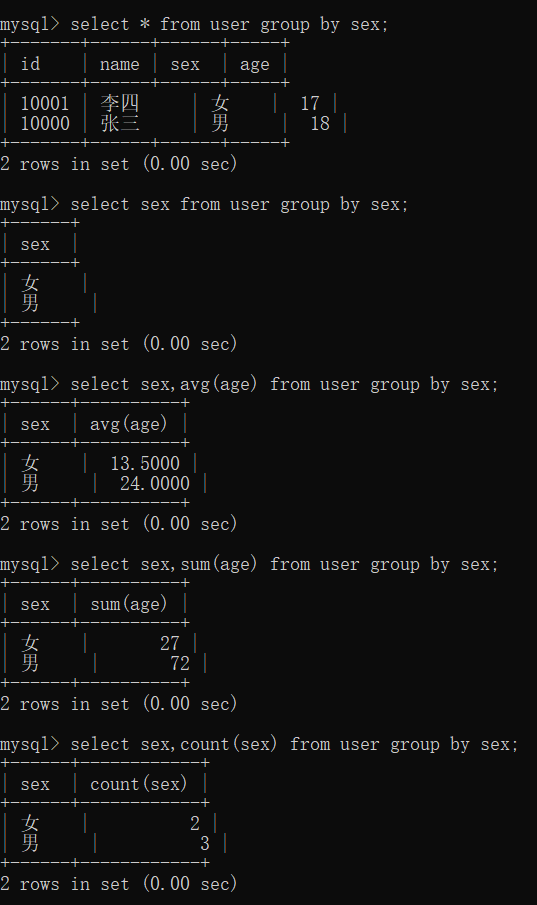
分组查询 group by sex

可以和聚合函数使用COUNT, SUM, AVG,等函数。

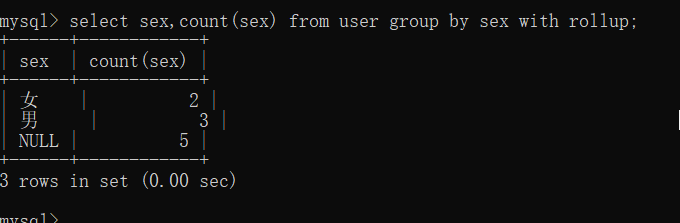
Count() 计算数据条数



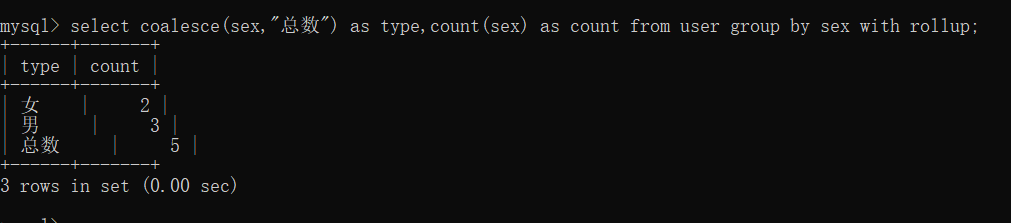
分组查询



分组之后 总统计 with rollup



列里面的null 值替换 coalesce(sex,”总数”)



高级查询

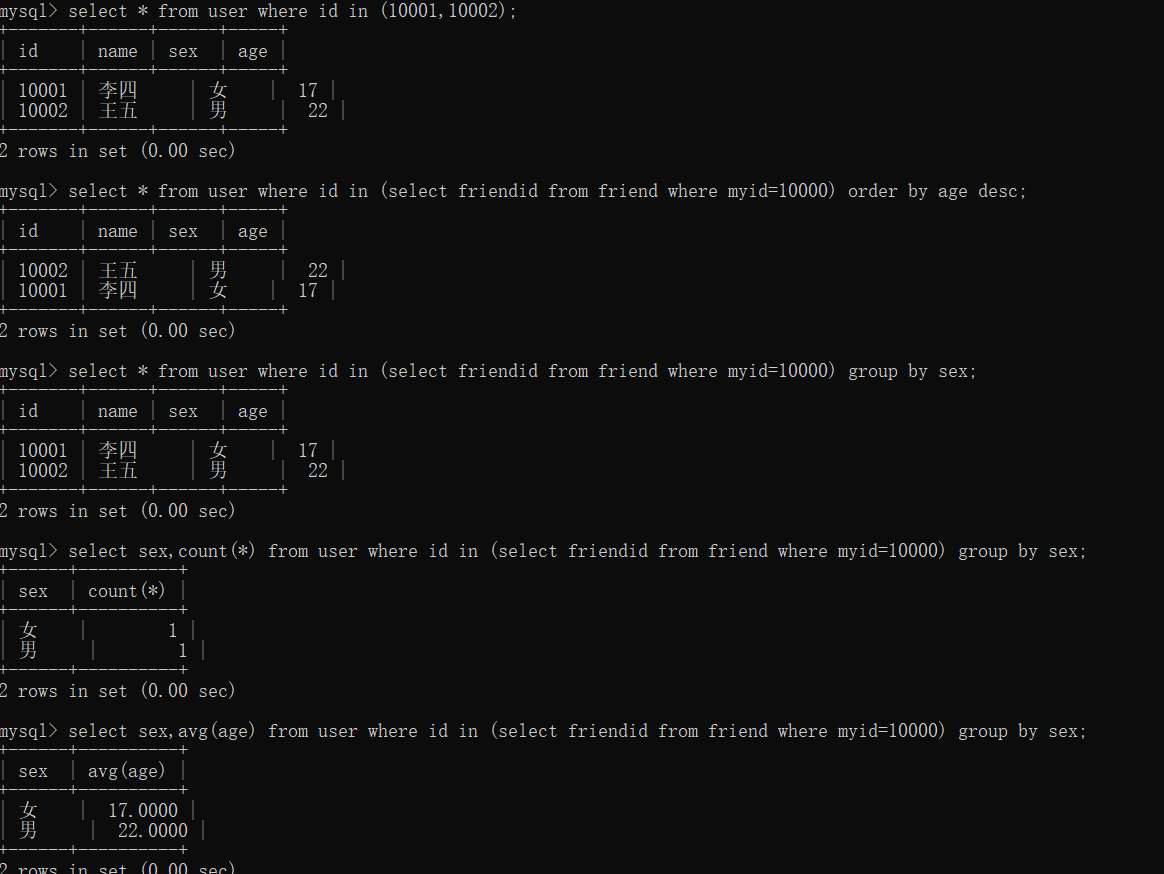
子查询

In 查询

根据我的id 查找我的好友的信息

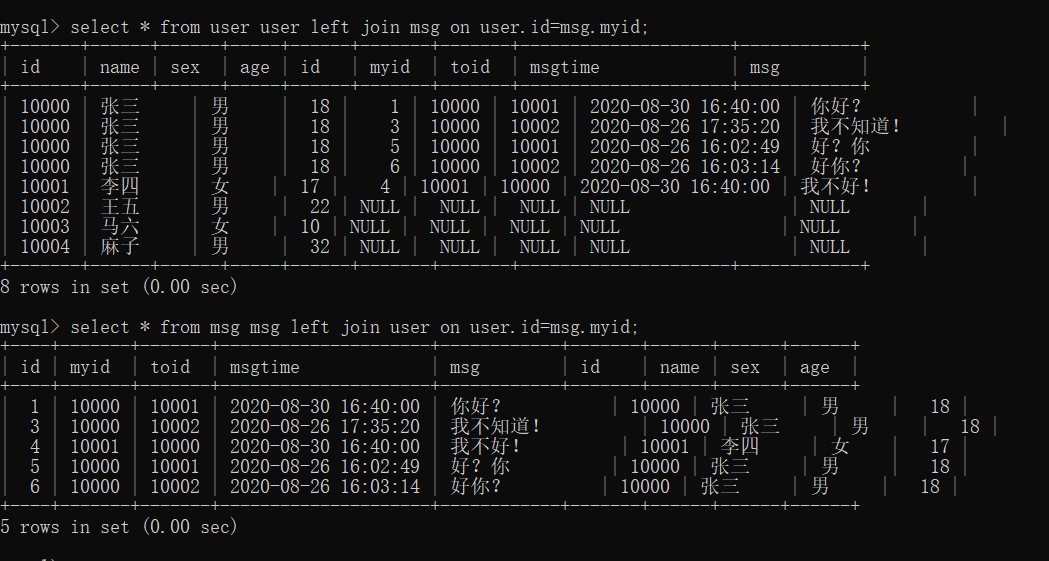
Select \* from user where id in (select friendid from friend where myid=10000);

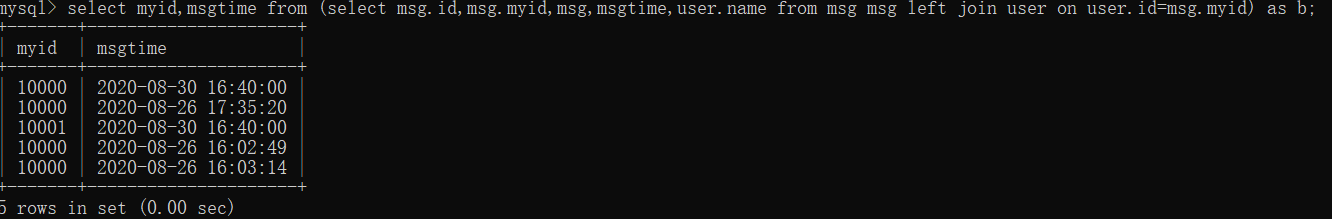
在子查询上写排序 分组 聚合 之类



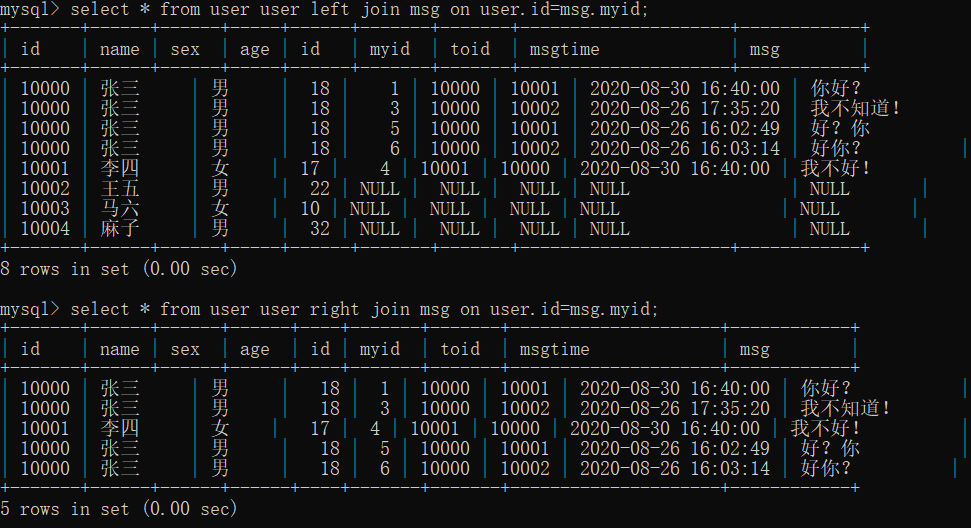
左连接 查询

Left join on



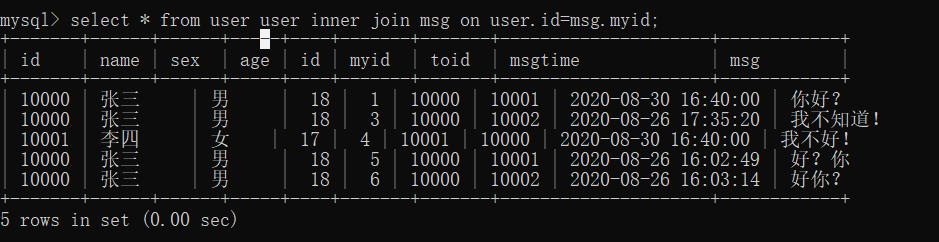


右连接 查询



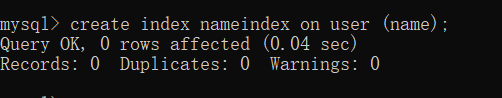
内连查询

Inner join on 根据条件返回两张表的交集



1. 给表中的列建立索引

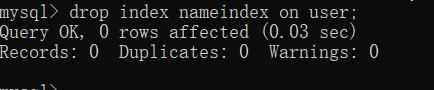
Create index indexname on tablename (colume);



修改表结构的方式添加索引

Alter table user add index nameindex (name);

删除指定的索引



1. 临时表 tempopary

创建临时表和普通表一致 只是加关键字

Crate tempopary table name()

1. mysql里面的事务

操作数据量大 复杂高 用事务

维护数据的完整性的

只有 innodb 数据库引擎的数据库和表 才可以使用事务

四大特性： 原子行，一致性，隔离性，持久性

创建事务的关键字

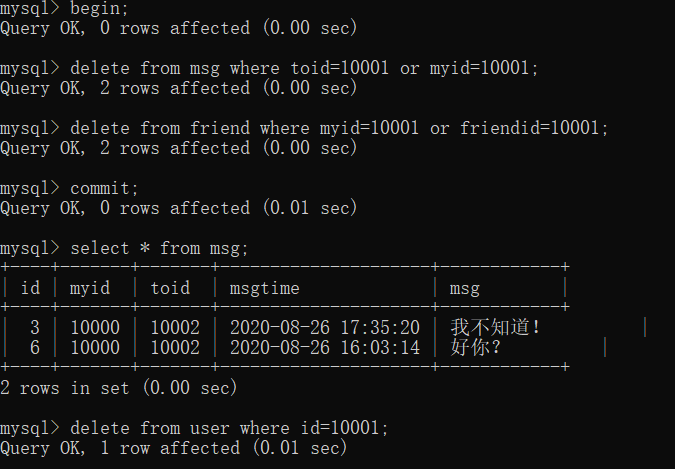
Begin; 开始事务

Rollback 事务的回滚

Commit 事务提交

Mysql 设置

* **SET AUTOCOMMIT=0** 禁止自动提交
* **SET AUTOCOMMIT=1** 开启自动提交



1. 存储过程

创建存储过程 可以将复杂的sql语句 封装存在mysql ,后台代码直接通过调用过程名称传递参数 执行sql

