**练习五 索引与视图的创建**

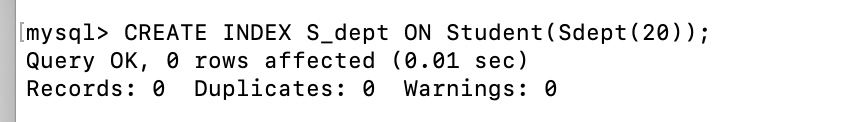
学号2351610105 姓名方泽宇

一、建立索引

1、使用CREATE INDEX和ALTER TABLE命令为Student表创建一个名为S\_dept的普通索引，以系别排序。

命令如下：

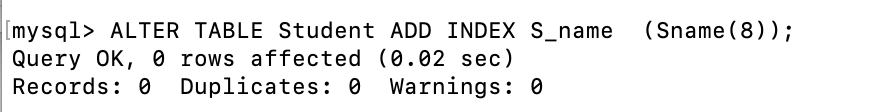
CREATE INDEX S\_dept ON Student(Sdept(20));



2、使用ALTER TABLE命令为Student表创建一个名为S\_name的普通索引，以姓名排序。

命令如下：

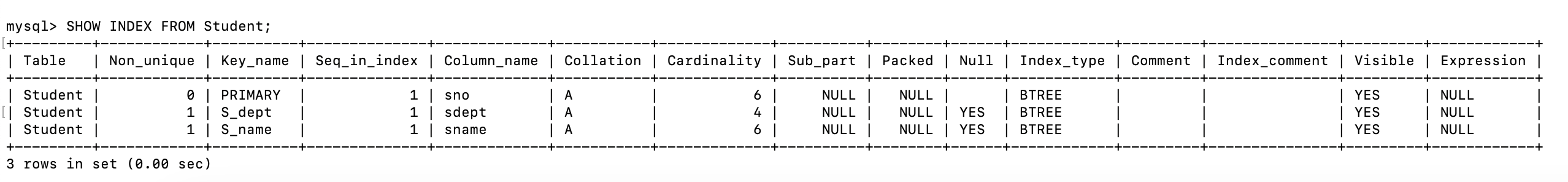
ALTER TABLE Student ADD INDEX S\_name  (Sname(8));



3、使用SHOW INDEX命令查看Student表的所有索引。

命令如下：

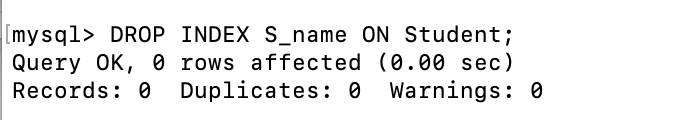
SHOW INDEX FROM Student;



4、为Student表删除S\_name索引。

命令如下：

DROP INDEX S\_name ON Student;

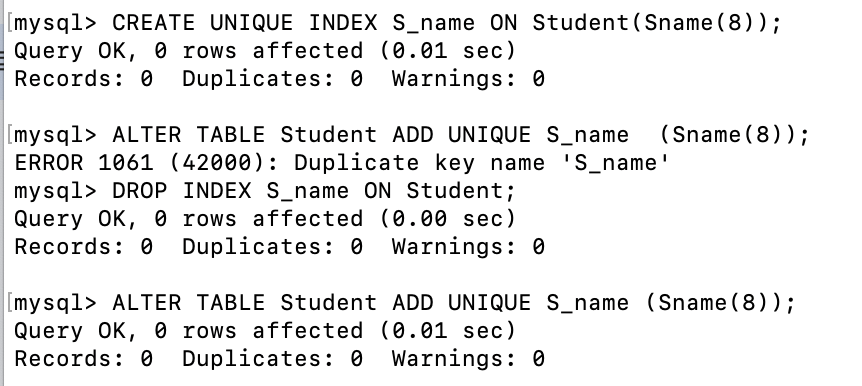


5、分别使用CREATE UNIQUE INDEX 和ALTER TABLE命令为Student表创建一个名为S\_name的唯一索引，以Sname排序。

命令如下：

CREATE UNIQUE INDEX S\_name ON Student(Sname(8));

ALTER TABLE Student ADD UNIQUE S\_name  (Sname(8));

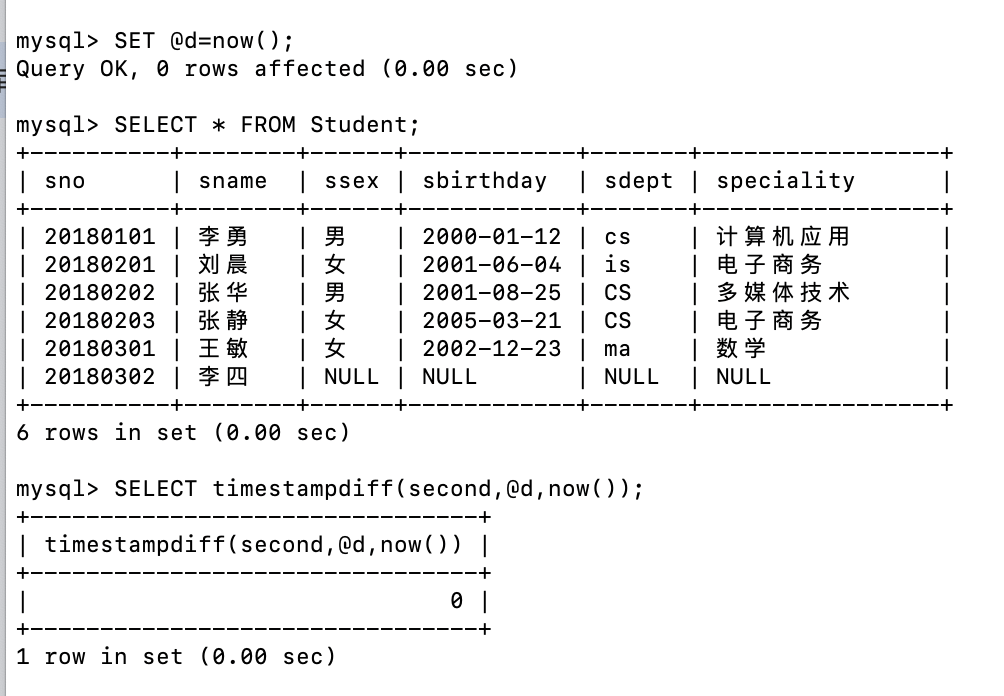


6、测试select查询花费时间。

SET @d=now();

SELECT \* FROM Student;

SELECT timestampdiff(second,@d,now());



TIMESTAMPDIFF(interval,datetime\_expr1,datetime\_expr2)。

说明：

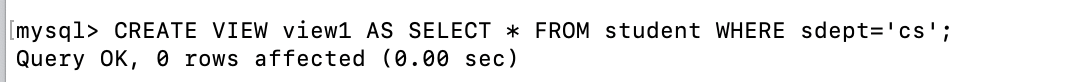
返回日期或日期时间表达式datetime\_expr1 和datetime\_expr2the 之间的整数差。其结果的单位由interval 参数给出。该参数必须是以下值的其中一个：

1. FRAC\_SECOND。表示间隔是毫秒
2. SECOND。秒
3. MINUTE。分钟
4. HOUR。小时
5. DAY。天
6. WEEK。星期
7. MONTH。月
8. QUARTER。季度
9. YEAR。年

二、视图的创建与使用

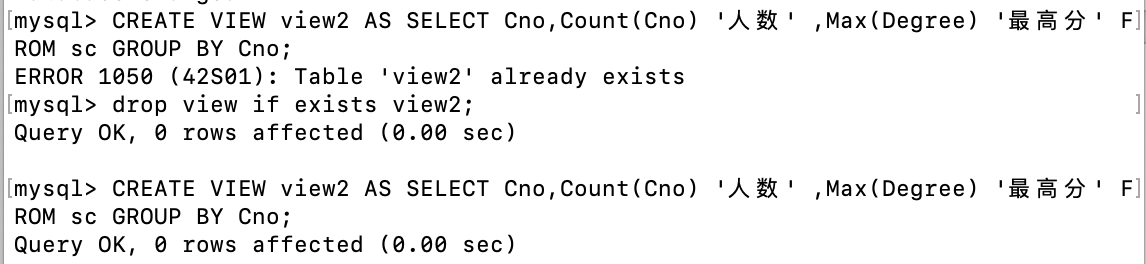
1、创建一个简单视图view1，查询“计算机系”学生的信息

CREATE VIEW view1 AS SELECT \* FROM student WHERE sdept=’cs’;



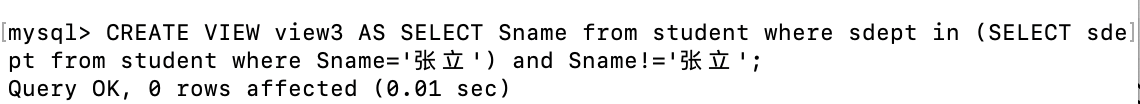
1. 创建一个简单视图view2，统计每门课程的选课人数和最高分

命令如下：



1. 创建一个复杂视图view3，查询与“张立”同学同一系别的学生信息

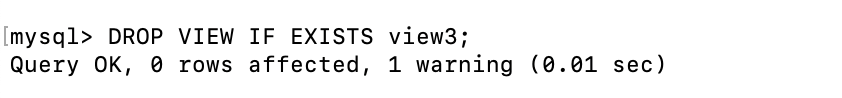
命令如下：



1. 删除视图View3

命令如下：

DROP VIEW IF EXISTS view3;



三、存储过程与触发器

1、创建store表格，命令如下：

CREATE TABLE IF NOT EXISTS `store` (  `id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(20) NOT NULL,

`count` int(11) NOT NULL DEFAULT '1',    PRIMARY KEY (`id`)

 ) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO\_INCREMENT=7;

INSERT INTO `store` (`id`, `name`, `count`) VALUES  (1, 'android', 15),

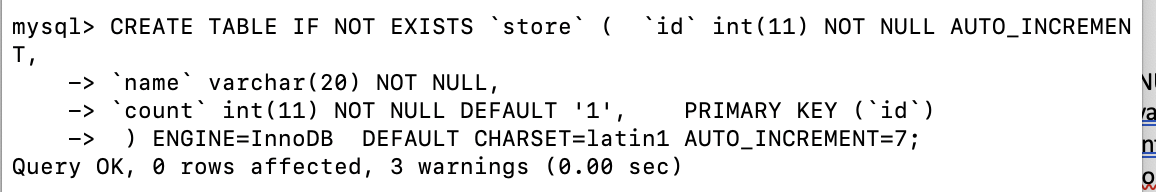
(2, 'iphone', 14),

(3, 'iphone', 20),

(4, 'android', 5),

(5, 'android', 13),

(6, 'iphone', 13);



2、编写存储过程StatisticStore，统计iphone的总库存是多少，并把总数输出到控制台。

delimiter //   --默认情况下，delimiter是分号; delimiter // 定义结束符为”//”

drop procedure if exists StatisticStore;   --如果存储过程StatisticStore已经存在就删除。

CREATE PROCEDURE StatisticStore()   --创建存储过程

BEGIN

    declare c int;

    declare n varchar(20);

    declare total int default 0;

    declare done int default false;   --声明c,n,total和done变量。其中done为结束标志变量

    declare cur cursor for select name,count from store where name = 'iphone';   --声明cur游标

    declare continue HANDLER for not found set done = true;   --指定游标循环结束时的返回值

    set total = 0;   --设置初始值

    open cur;   --打开游标

    fetch cur into n,c;   --游标指向当前数据

    while(not done) do   --判断游标的循环是否结束

        set total = total + c;   --获取一条数据时，将total值进行累加操作，这里可以做任意你想做的操作，

        fetch cur into n,c;     --推进游标

    end while;

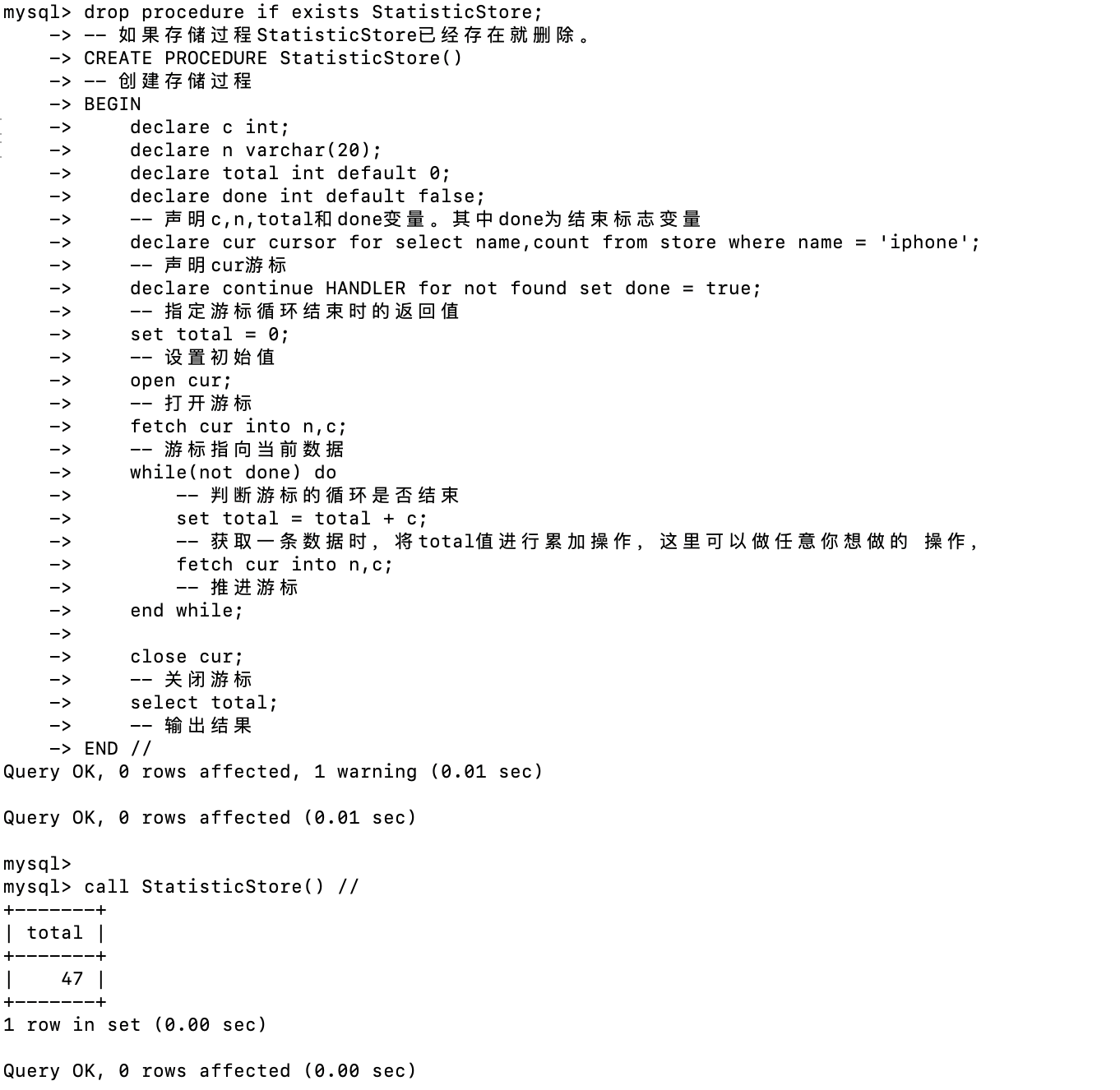
    close cur;   --关闭游标

    select total;   --输出结果

END;

call StatisticStore1();  // --调用存储过程

delimiter ;



Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

+-------+

| total |

+-------+

| 47 |

+-------+

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

3、查看jxgl数据库的存储过程。

SELECT name,db,type FROM mysql.proc WHERE db='jxgl' AND type='procedure';

结果如下：

4、触发器的使用

（1）创建学生表student\_info和学生数目统计表student\_count：

触发器是一种与表操作有关的数据库对象，当触发器所在表上出现指定事件时，将调用该对象，即表的操作事件触发表上的触发器的执行。

CREATE TABLE student\_info (stu\_no INT(11) NOT NULL AUTO\_INCREMENT,

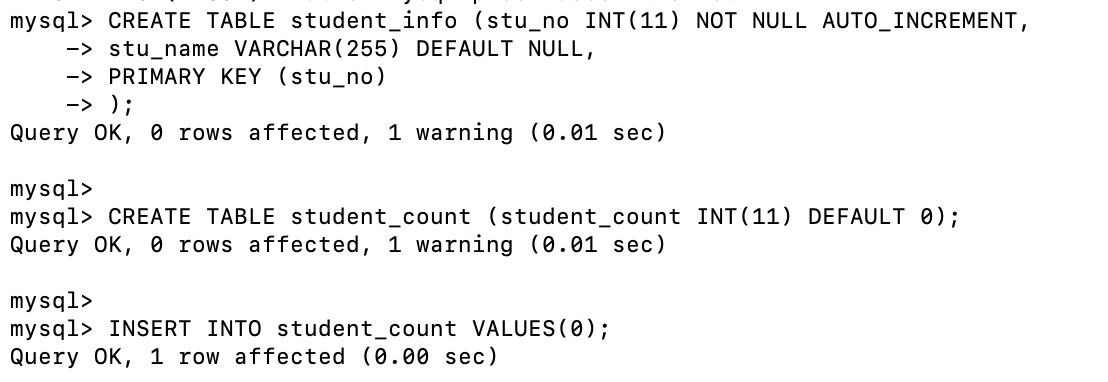
stu\_name VARCHAR(255) DEFAULT NULL,

PRIMARY KEY (stu\_no)

);

CREATE TABLE student\_count (student\_count INT(11) DEFAULT 0);

INSERT INTO student\_count VALUES(0);



（2）创建简单触发器，在向学生表INSERT数据时，学生数增加，DELETE学生时，学生数减少。

CREATE TRIGGER trigger\_student\_count\_insert

AFTER INSERT ON student\_info FOR EACH ROW

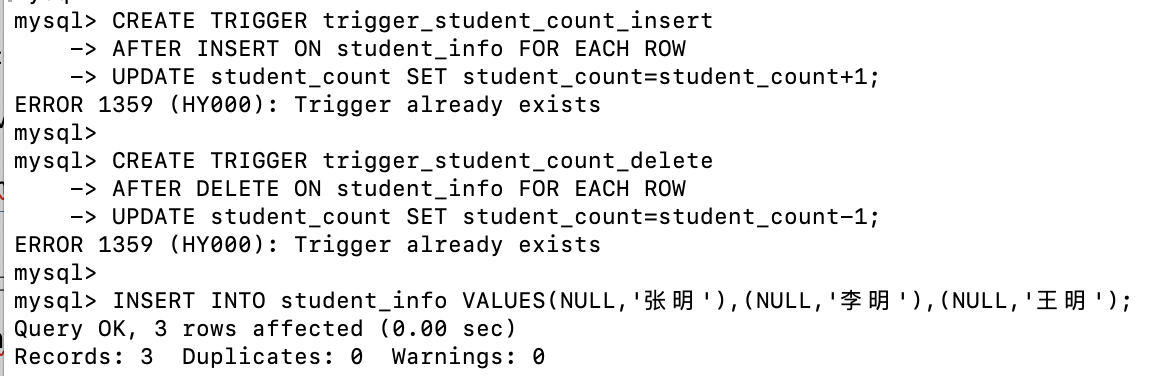
UPDATE student\_count SET student\_count=student\_count+1;

CREATE TRIGGER trigger\_student\_count\_delete

AFTER DELETE ON student\_info FOR EACH ROW

UPDATE student\_count SET student\_count=student\_count-1;

INSERT INTO student\_info VALUES(NULL,'张明'),(NULL,'李明'),(NULL,'王明');



--设置中文编码

set names utf8；

--设置数据库字符集：

alter database jxgl character set utf8;

--设置表格默认字符集：

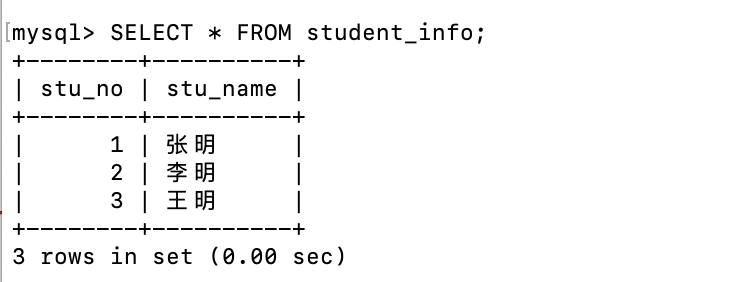
ALTER TABLE student\_info CONVERT TO CHARACTER SET utf8;

--设置字段编码格式

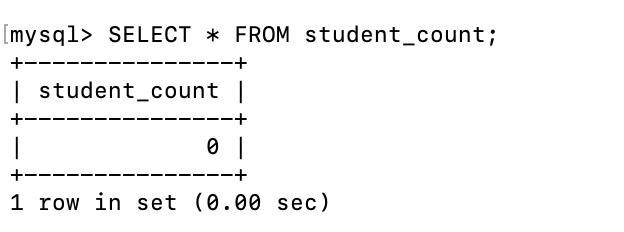
alter table student change Sname sname varchar(8) character set utf8;

（3）执行如下命令，写出结果并理解触发器功能：

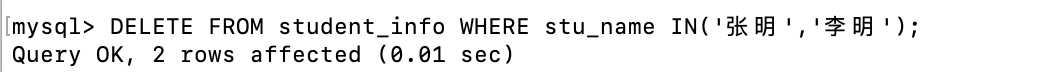
SELECT \* FROM student\_info;



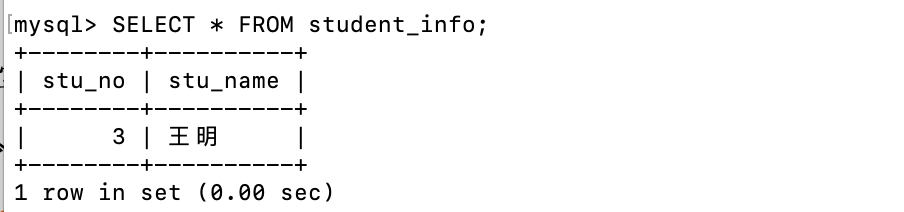
SELECT \* FROM student\_count;



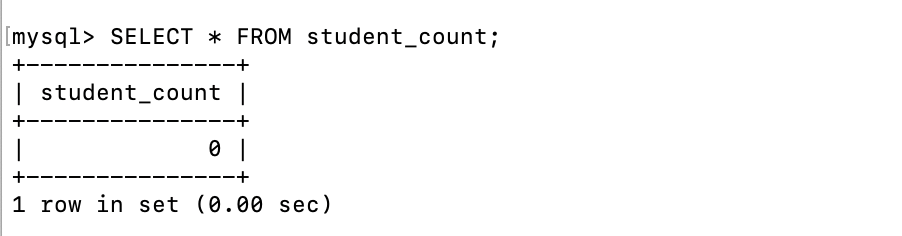
DELETE FROM student\_info WHERE stu\_name IN('张明','李明');



SELECT \* FROM student\_info;



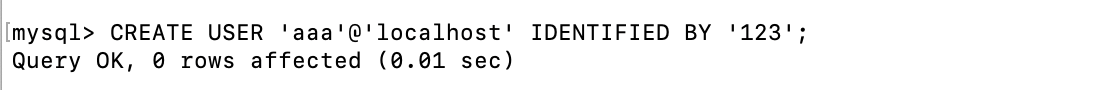
SELECT \* FROM student\_count;



四、用户管理（请将命令写在空白处）

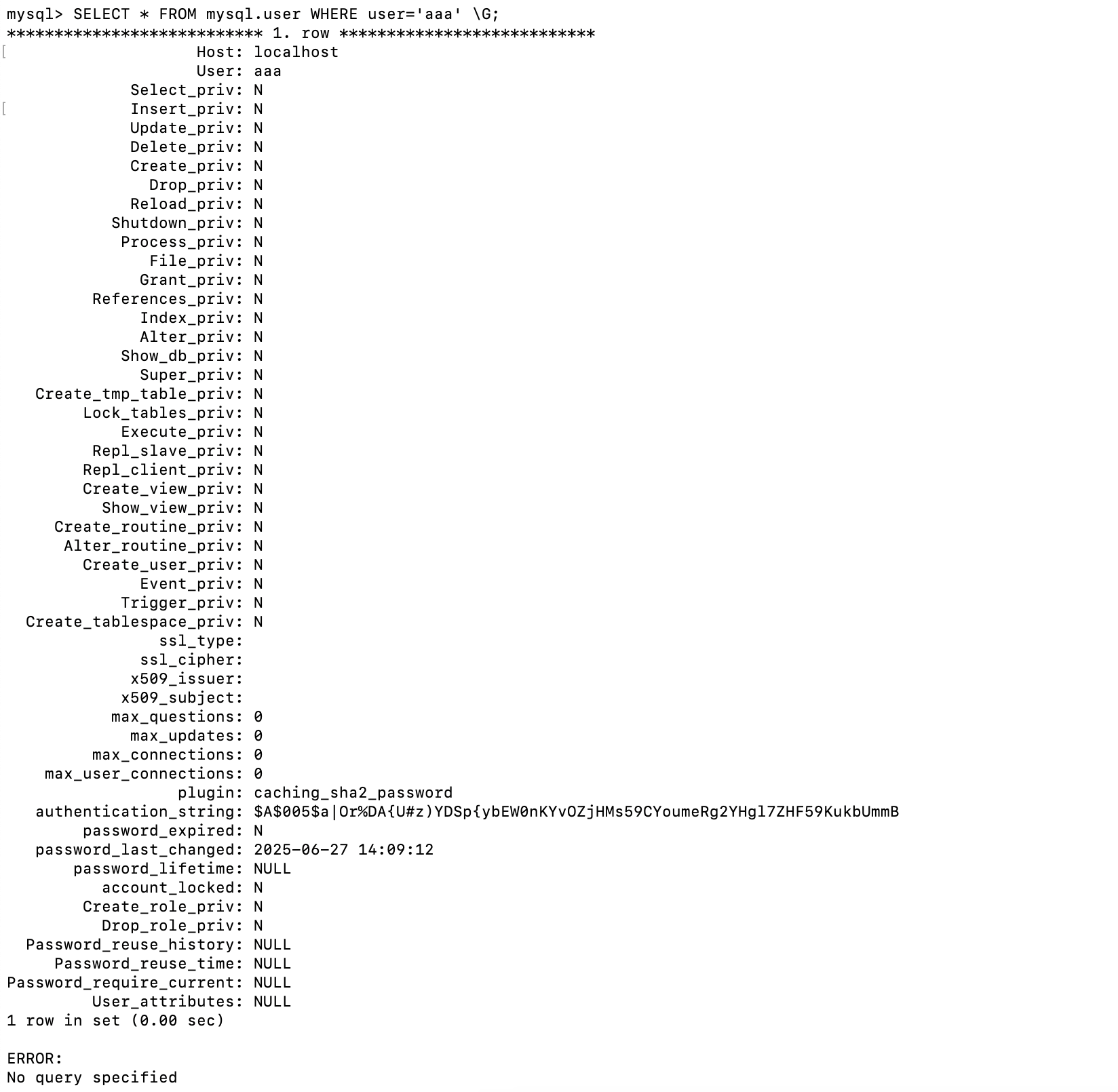
1、使用CREATE USER命令创建用户’aaa’，密码为’123’。

CREATE USER 'aaa'@'localhost' IDENTIFIED BY '123';

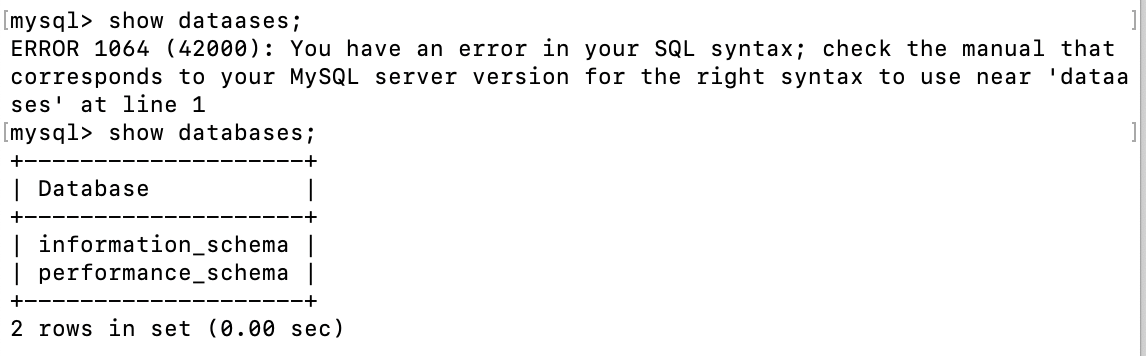


2、查看用户’aaa’的权限。（此时aaa用户几乎所有权限为”N”）。

SELECT \* FROM mysql.user WHERE user='aaa' \G;

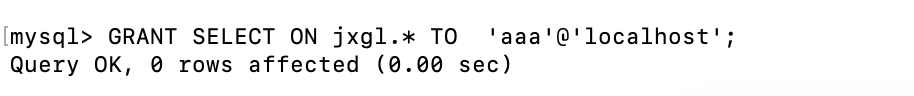


3、新建一个会话，并以’aaa’用户登陆MySQL服务器，使用SHOW DATABASES命令查看所有数据库。（aaa用户下只有information\_schema库）

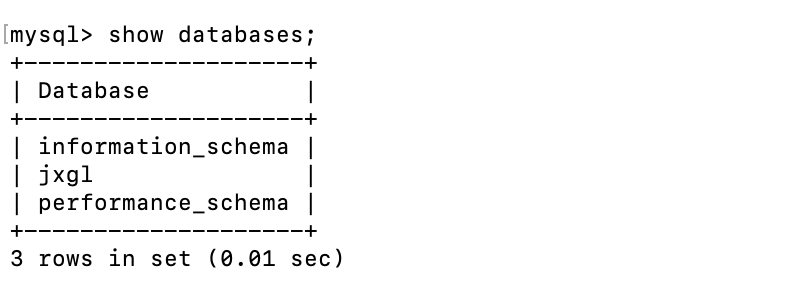


4、在root用户下，使用GRANT命令为’aaa’用户分配查询jxgl数据库的权限。

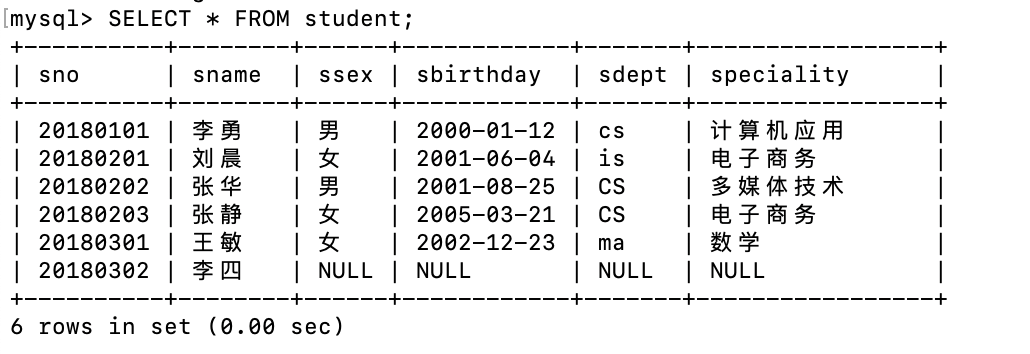
GRANT SELECT ON jxgl.\* TO 'aaa'@'localhost';



5、在aaa用户下，使用SHOW DATABASES命令查看所有数据库。（此时可以看到jxgl数据库）



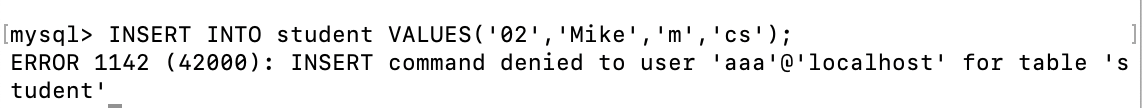
6、执行命令：SELECT \* FROM student; 写出结果。



7、执行命令：

INSERT INTO student VALUES('02','Mike','m','cs');

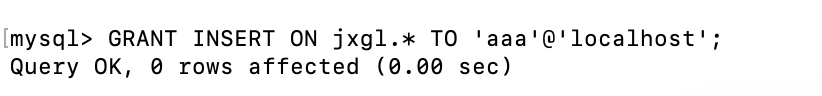
写出结果，并分析原因。



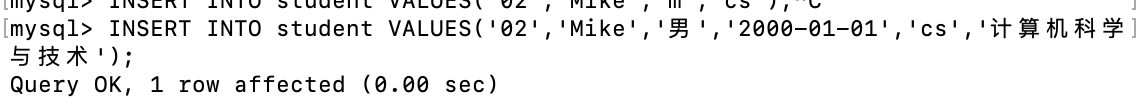
原因：aaa用户没insert权限

8、使用root用户为aaa用户分配INSERT权限。

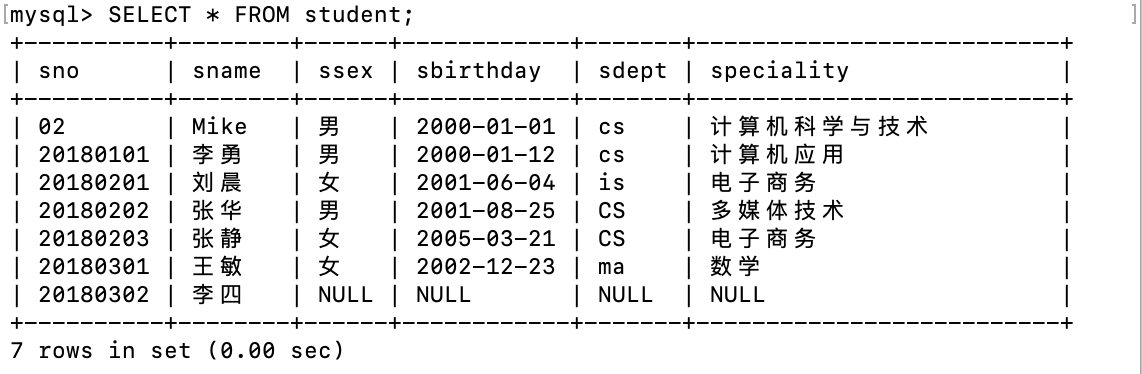
GRANT INSERT ON jxgl.\* TO 'aaa'@'localhost';



9、在aaa用户下执行命令：INSERT INTO student VALUES('02','Mike','m','cs');

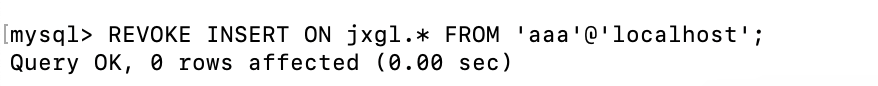


10、查询student表格所有记录。



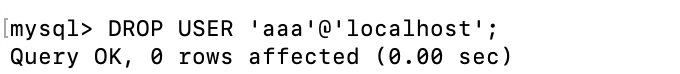
11、使用root收回aaa用户的INSERT权限。

REVOKE INSERT ON jxgl.\* FROM 'aaa'@'localhost';



12、删除aaa用户。

DROP USER 'aaa'@'localhost';



* **思考题**

1. 使用索引为什么能提高查询速度？
2. 减少全盘扫描
3. 减少I/O次数
4. 索引经常被建立在哪些字段上？
5. 主键
6. 外键
7. 被经常频繁查询的字段

3．聚集索引与其他索引有什么不同？

聚焦查询优化了查询速度，但是会影响修改性能，存储了表的数据行，其顺序决定了表中行的物理顺序。