实验2 SQL数据定义和操作

Zhang Yichi 3180103772

实验目的：

1. 掌握关系数据库语言SQL的使用。
2. 使所有的SQL作业都能上机通过。

实验平台：

1. 数据库管理系统：MySQL
    - 实验基于Mysql workbench进行
2. 操作系统: Windows 10

实验内容和要求：

1. 建立数据库。

- 建立名为“student”的数据库

```
create database student;
```

- 创建的结果为`Query OK, 1 row affected (0.01 sec)`
- 使用该数据库

```
use student;
```

2. 数据定义： 表的建立/删除/修改; 索引的建立/删除；视图的建立/删除

- 表的建立：建立了名为student_info的表

```
create table student_info(
    stu_name varchar(20),
    stu_id varchar(20),
    age int,
    phone varchar(20),
    grade1 numeric(3,1),
    grade2 numeric(3,1),
    grade3 numeric(3,1),
    primary key (stu_id));
```

- 建立的结果可以用DESC语句查询到

Field	Type	Null	Key	Default	Extra
stu_name	varchar(20)	YES		NULL	
stu_id	varchar(20)	NO	PRI	NULL	
age	int	YES		NULL	
phone	varchar(20)	YES		NULL	
grade1	decimal(3,1)	YES		NULL	
grade2	decimal(3,1)	YES		NULL	
grade3	decimal(3,1)	YES		NULL	

- 表的删除与修改：再建立一个实验性的表student_info2对其进行修改和删除

```
create table student_info2(
    stu_name varchar(20),
    stu_id varchar(20),
    age int,
    phone varchar(20),
    grade1 numeric(3,1),
    grade2 numeric(3,1),
    grade3 numeric(3,1),
    primary key (stu_id));
```

```
alter table student_info2
add column stu_rank int; //添加一列名为stu_rank

alter table student_info2
drop column age;//删除名为age的一列

alter table student_info2
rename student_info3;//对表进行重命名
```

修改后的结果为

Field	Type	Null	Key	Default	Extra
stu_name	varchar(20)	YES		NULL	
stu_id	varchar(20)	NO	PRI	NULL	
phone	varchar(20)	YES		NULL	
grade1	decimal(3,1)	YES		NULL	
grade2	decimal(3,1)	YES		NULL	
grade3	decimal(3,1)	YES		NULL	
stu_rank	int	YES		NULL	

7 rows in set (0.00 sec)

```
drop table student_info3;//删除这个表
```

- 索引的建立和删除

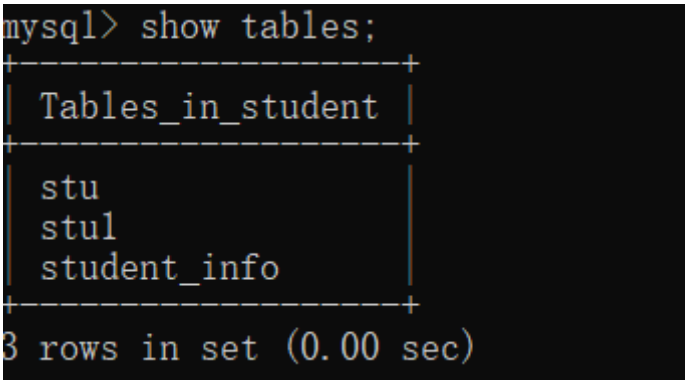
```
create index age_index on student_info(age);
create index name_index on student_info(stu_name);
```

```
drop index age_index on student_info;
```

- 视图的建立和删除

```
create view stu(stu_name,stu_id,age)
as select stu_name,stu_id,age
from student_info;
```

查询可查看的视图，发现已经建立成功。



删除多建立的视图stu1，

```
drop view stu1;
```

使用show table语句之后发现stu1已经被删除

3. 数据更新： 用 insert/delete/update命令插入/删除/修改表数据。

- 插入若干条数据

```
insert into student_info
values('zhangyichi','3180103772',20,'18888913487',90.0,87.0,88.0);
insert into student_info values('zhangerchi','3180101234',21,'10086',65,77,53);
insert into student_info values('zhangsanchi','3180102345',21,'10000',65,77,53);
insert into student_info
values('zhangsichi','3180103456',23,'4008823823',67,98,100);
insert into student_info values('zhangwuchi','3180104567',22,'10086',89,65,0);
insert into student_info values('randomstar','3180105678',19,'10086',95,81,79);
```

stu_name	stu_id	age	phone	grade1	grade2	grade3
zhangerchi	3180101234	21	10086	65.0	77.0	53.0
zhangsanchi	3180102345	21	10000	65.0	77.0	53.0
zhangsichi	3180103456	23	4008823823	67.0	98.0	95.0
zhangyichi	3180103772	20	18888913487	90.0	87.0	88.0
zhangwuchi	3180104567	22	10086	89.0	65.0	0.0
randomstar	3180105678	19	10086	95.0	81.0	79.0

6 rows in set (0.00 sec)

- 删除/修改数据

```
delete from student_info where stu_id='3180104567';
```

删除后的结果为

```
+-------------+------------+------+-------------+--------+--------+--------+
| stu_name    | stu_id     | age  | phone       | grade1 | grade2 | grade3 |
+-------------+------------+------+-------------+--------+--------+--------+
| zhangerchi  | 3180101234 |   21 | 10086       |   65.0 |   77.0 |   53.0 |
| zhangsanchi | 3180102345 |   21 | 10000       |   65.0 |   77.0 |   53.0 |
| zhangsichi  | 3180103456 |   23 | 4008823823  |   67.0 |   98.0 |   95.0 |
| zhangyichi  | 3180103772 |   20 | 18888913487 |   90.0 |   87.0 |   88.0 |
| randomstar  | 3180105678 |   19 | 10086       |   95.0 |   81.0 |   79.0 |
+-------------+------------+------+-------------+--------+--------+--------+
5 rows in set (0.00 sec)
```

```
update student_info set stu_name='zyc',age=18 where stu_id='3180103456';
```

修改后的结果为

```
mysql> select * from student_info;
+-------------+------------+------+-------------+--------+--------+--------+
| stu_name    | stu_id     | age  | phone       | grade1 | grade2 | grade3 |
+-------------+------------+------+-------------+--------+--------+--------+
| zhangerchi  | 3180101234 |   21 | 10086       |   65.0 |   77.0 |   53.0 |
| zhangsanchi | 3180102345 |   21 | 10000       |   65.0 |   77.0 |   53.0 |
| zyc         | 3180103456 |   18 | 4008823823  |   67.0 |   98.0 |   95.0 |
| zhangyichi  | 3180103772 |   20 | 18888913487 |   90.0 |   87.0 |   88.0 |
| randomstar  | 3180105678 |   19 | 10086       |   95.0 |   81.0 |   79.0 |
+-------------+------------+------+-------------+--------+--------+--------+
5 rows in set (0.00 sec)
```

4. 数据查询： 单表查询，多表查询， 嵌套子查询等。

- 查询年龄大于等于20的学生信息(单表查询)

```
select * from student_info
where age>=20;
```

```
+-------------+------------+------+-------------+--------+--------+--------+
| stu_name    | stu_id     | age  | phone       | grade1 | grade2 | grade3 |
+-------------+------------+------+-------------+--------+--------+--------+
| zhangerchi  | 3180101234 |   21 | 10086       |   65.0 |   77.0 |   53.0 |
| zhangsanchi | 3180102345 |   21 | 10000       |   65.0 |   77.0 |   53.0 |
| zhangyichi  | 3180103772 |   20 | 18888913487 |   90.0 |   87.0 |   88.0 |
+-------------+------------+------+-------------+--------+--------+--------+
3 rows in set (0.00 sec)
```

- 查询总分大于200的学生的姓名和id(单表查询)

```
select stu_name,stu_id
from student_info
where grade1+grade2+grade3>200;
```

```
+-------------+------------+
| stu_name    | stu_id     |
+-------------+------------+
| zyc         | 3180103456 |
| zhangyichi  | 3180103772 |
| randomstar  | 3180105678 |
+-------------+------------+
3 rows in set (0.00 sec)
```

- 多表查询(为此新建了一张数据表)，一张表是原本的student_info，另一张表是grade1的成绩组成情况，下面的查询实现了查询grade1不低于90分的同学的成绩组成的查询

```
select stu_name,stu_id,grade.grade1,final,lab,homework
from student_info,grade
where student_info.stu_name=grade.stuname and student_info.grade1>=90;
```

```
+------------+------------+--------+-------+------+----------+
| stu_name   | stu_id     | grade1 | final | lab  | homework |
+------------+------------+--------+-------+------+----------+
| randomstar | 3180105678 |   95.0 |    90 |   98 |       92 |
| zhangyichi | 3180103772 |   90.0 |    85 |   93 |       94 |
+------------+------------+--------+-------+------+----------+
2 rows in set (0.00 sec)
```

- 查询grade1成绩最高的学生的姓名和id(嵌套子查询)

```
select stu_name,stu_id
from student_info
where grade1>=all(
    select grade1
    from student_info);
```

```
+------------+------------+
| stu_name   | stu_id     |
+------------+------------+
| randomstar | 3180105678 |
+------------+------------+
1 row in set (0.00 sec)
```

- 查询所有成绩1比成绩2高的同学的姓名(嵌套子查询)

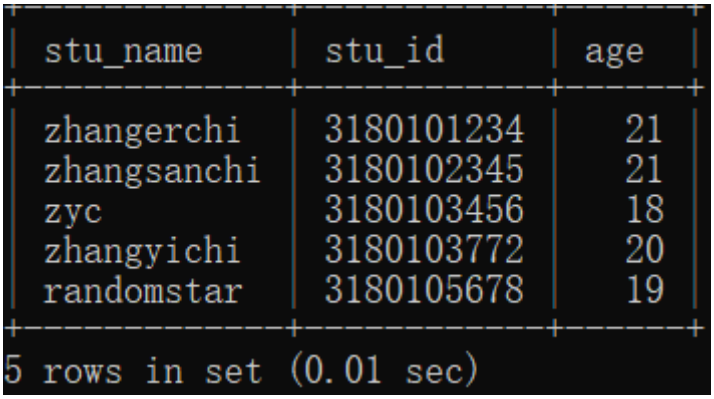
```
select distinct stu_name
from(select stu_name,grade1 from student_info) a,
(select grade2 from student_info) b
where a.grade1>b.grade2;
```

```
+------------+
| stu_name   |
+------------+
| zhangyichi |
| randomstar |
+------------+
2 rows in set (0.00 sec)
```

5. 视图操作：通过视图的数据查询和数据修改

- 通过视图进行数据查询，选用刚才建立的视图stu

```
select * from stu;
```

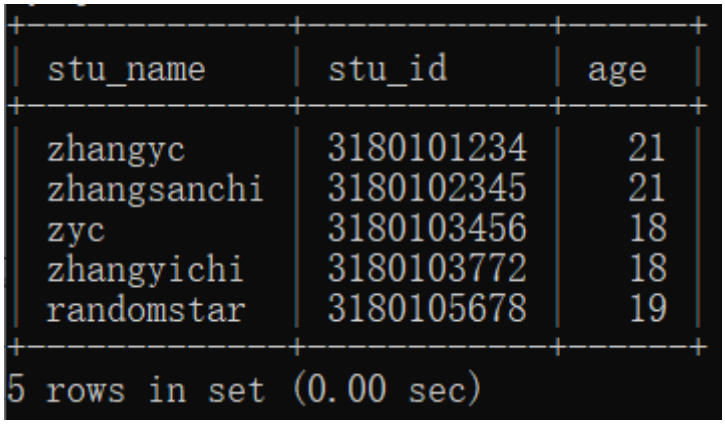


stu_name	stu_id	age
zhangerchi	3180101234	21
zhangsanchi	3180102345	21
zyc	3180103456	18
zhangyichi	3180103772	20
randomstar	3180105678	19

5 rows in set (0.01 sec)

- 通过视图进行数据的修改

```
update stu set age=18 where stu_name='zhangyichi';
update stu set stu_name='zhangyc' where stu_id='3180101234';
```



stu_name	stu_id	age
zhangyc	3180101234	21
zhangsanchi	3180102345	21
zyc	3180103456	18
zhangyichi	3180103772	18
randomstar	3180105678	19

5 rows in set (0.00 sec)

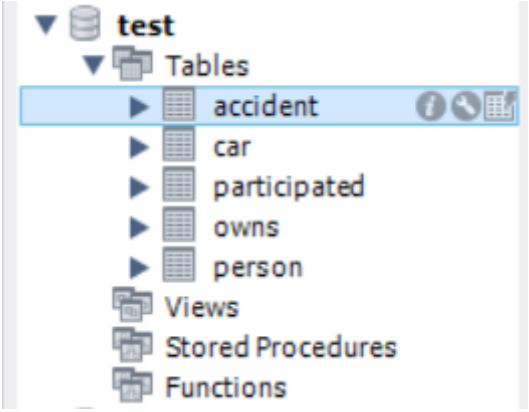
6.完成实验报告。

7.所有SQL作业上机通过

- 3.13建立数据表

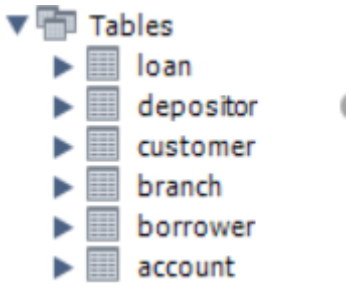
```
use test;
create table person(
    driver_id       varchar(10),
    name            varchar(20),
    address         varchar(50),
    primary key(driver_id),
    foreign key(driver_id) references  participated);
create table car(
    lisence         varchar(20),
    model           varchar(20),
    year            int,
    primary key(license),
    foreign key(license) references participated);
create table accident(
    report_number   int,
    date            date,
    location        varchar(50),
    primary key(report_number),
    foreign key(report_number) references participated);
create table owns(
    driver_id       varchar(10),
    lisence         varchar(20),
    primary key(driver_id),
    foreign key(driver_id) references owns );
create table participated(
    report_number   int,
    license         varchar(20),
    driver_id       varchar(10),
    damage_amount   numeric(10,2),
```

建立完成后发现成功建立了这些表格



3.8

- 先建立如图所示的数据表并插入一些数据



- 3.8.a

Mysql中不支持except语句，我们换用union语句代替，得到的结果为

	customer_name
▶	zyc
	zyc2

- 3.8.b，结果为，与输入的数据相符合

	customer_name
▶	zyc
	Smith
	zyc2

- 3.8.c 结果如下，符合要求

```
insert into account values('10086','CB','1');
insert into customer values ('zhangyichi','xc','Harrison');
insert into depositor values ('zhangyichi','10086');
select distinct branch_name
from account natural join customer natural join depositor
where customer_city='Harrison'
```

Result Grid | Filter

	branch_name
▶	CB