

浙江大学



《数据库系统》 实验报告

作业名称 :	SQL数据完整性
姓 名 :	王晓宇
学 号 :	3220104364
电子邮箱 :	3220104364@zju.edu.cn
联系电话 :	19550222634
授课教师 :	孙建伶

2024 年 3 月 18 日

实验名称 SQL数据完整性

1 实验目的

1. 熟悉通过SQL进行数据完整性控制的方法。

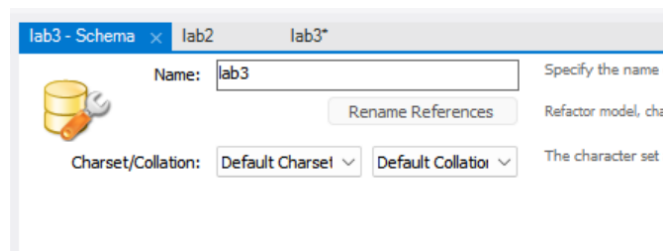
2 实验环境

1. 操作系统: Windows 11 23H2
2. 数据库管理系统: MySQL 8.0.36
3. 工具: MySQL workbench 8.0

3 实验流程

在实验过程中，截图所用命令和实验结果

- 创建‘lab3’数据库，以下操作均在lab3中进行。



3.1 定义若干表

包括primary key, foreign key 和check的定义

```
use lab3;
```

```
# 课程表
```

```
CREATE TABLE course(  
  c_id VARCHAR(20),  
  c_name VARCHAR(20) ,  
  PRIMARY KEY(c_id)  
);
```

```
# 学生表
```

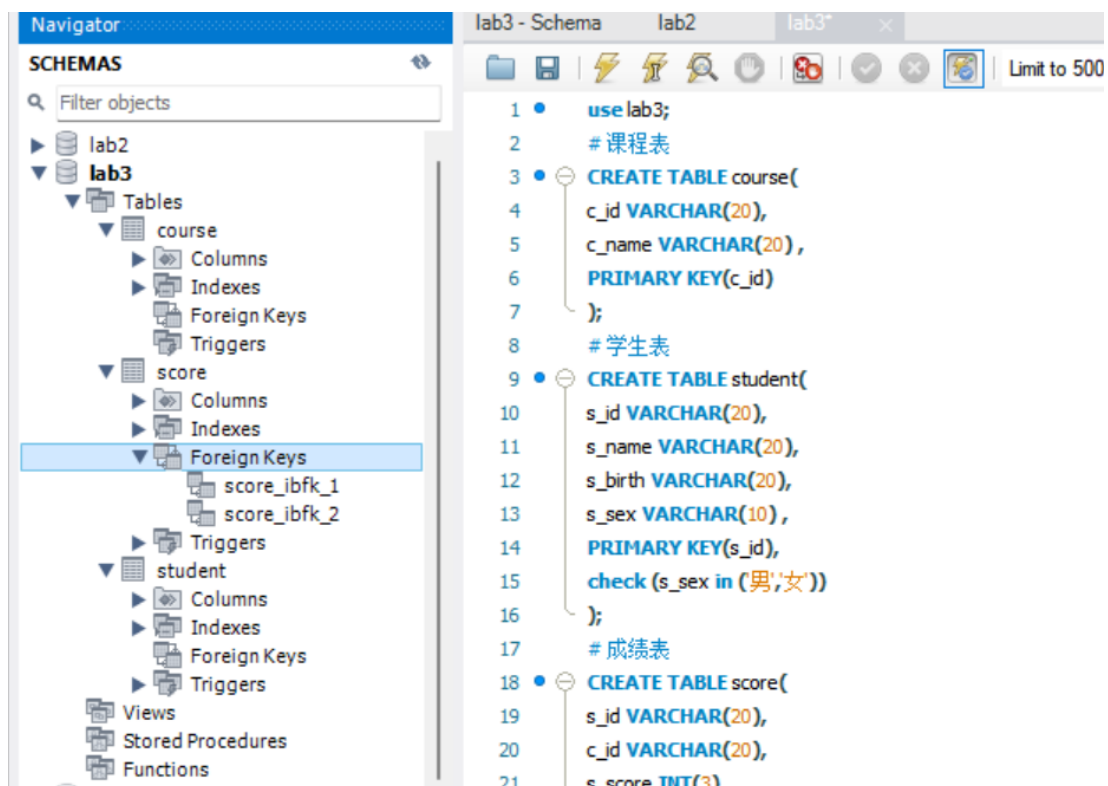
```
CREATE TABLE student(  
  s_id VARCHAR(20),  
  s_name VARCHAR(20),
```

```

s_birth VARCHAR(20),
s_sex VARCHAR(10) ,
PRIMARY KEY(s_id),
check (s_sex in ('男','女'))
);
# 成绩表
CREATE TABLE score(
s_id VARCHAR(20),
c_id VARCHAR(20),
s_score INT(3),
PRIMARY KEY(s_id,c_id),
check (s_score >0),
foreign key(s_id) references student(s_id),
foreign key(c_id) references course(c_id)
);

```

可以在左面看到生成的表(包括primary key, foreign key 和check的定义)



3.2 表中插入数据

考察primary key如何控制实体完整性。

首先插入主键不重复的数据

插入学生表测试数据

```
insert into student values('01' , '赵雷' , '1990-01-01' , '男');
insert into student values('02' , '钱电' , '1990-12-21' , '男');
insert into student values('03' , '孙风' , '1990-05-20' , '男');
insert into student values('04' , '李云' , '1990-08-06' , '男');
insert into student values('05' , '周梅' , '1991-12-01' , '女');
insert into student values('06' , '吴兰' , '1992-03-01' , '女');
insert into student values('07' , '郑竹' , '1989-07-01' , '女');
insert into student values('08' , '王菊' , '1990-01-20' , '女');
```

课程表测试数据

```
insert into course values('01' , '语文' );
insert into course values('02' , '数学' );
insert into course values('03' , '英语' );
```

成绩表测试数据

```
insert into score values('01' , '01' , 80);
insert into score values('01' , '02' , 90);
insert into score values('01' , '03' , 99);
insert into score values('02' , '01' , 70);
insert into score values('02' , '02' , 60);
insert into score values('02' , '03' , 80);
insert into score values('03' , '01' , 80);
insert into score values('03' , '02' , 80);
insert into score values('03' , '03' , 80);
insert into score values('04' , '01' , 50);
insert into score values('04' , '02' , 30);
insert into score values('04' , '03' , 20);
insert into score values('05' , '01' , 76);
insert into score values('05' , '02' , 87);
insert into score values('06' , '01' , 31);
insert into score values('06' , '03' , 34);
insert into score values('07' , '02' , 89);
```

```
insert into score values('07' , '03' , 98);
```

现在向学生表插入一条主键重复数据

```
insert into student values('08' , 'Ironman' , '1990-01-20' , '男');
```

##现表中已经有一条s_id为08的信息, primary key 会导致其无法插入

65	15:05:37	insert into student values('08', 'Ironman', '1990-01-20', '男')	Error Code: 1062. Duplicate entry '08' for key 'student.PRIMARY'
----	----------	--	--

Error Code: 1062. Duplicate entry '08' for key 'student.PRIMARY'

可以看到由于主键的作用无法插入这条数据

3.3 删除被引用表中的行

考察foreign key中on delete子句如何控制参照完整性

`score`中的s_id指向`student`表中的s_id,删除`student`表中被指向数据会导致触发外键约束

```
delete from student  
where s_id = '01';
```

产生报错:

#	Time	Action	Message
98	15:12:59	insert into score values('06' , '01' , 31)	1 row(s) affected
99	15:12:59	insert into score values('06' , '03' , 34)	1 row(s) affected
100	15:12:59	insert into score values('07' , '02' , 89)	1 row(s) affected
101	15:12:59	insert into score values('07' , '03' , 98)	1 row(s) affected
102	15:13:06	delete from student where s_id = '01'	Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (lab3.`score`, CONSTRAINT `score_ibfk_1` FOREIGN KEY (s_id) REFERENCES `student` (s_id))

Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (lab3.`score`, CONSTRAINT `score_ibfk_1` FOREIGN KEY (s_id) REFERENCES `student` (s_id))

3.4 修改被引用表中的行的primary key

考察foreign key中on update子句如何控制参照完整性

`score`中的s_id指向`student`表中的s_id,更新`student`表中被指向数据会导致触发外键约束

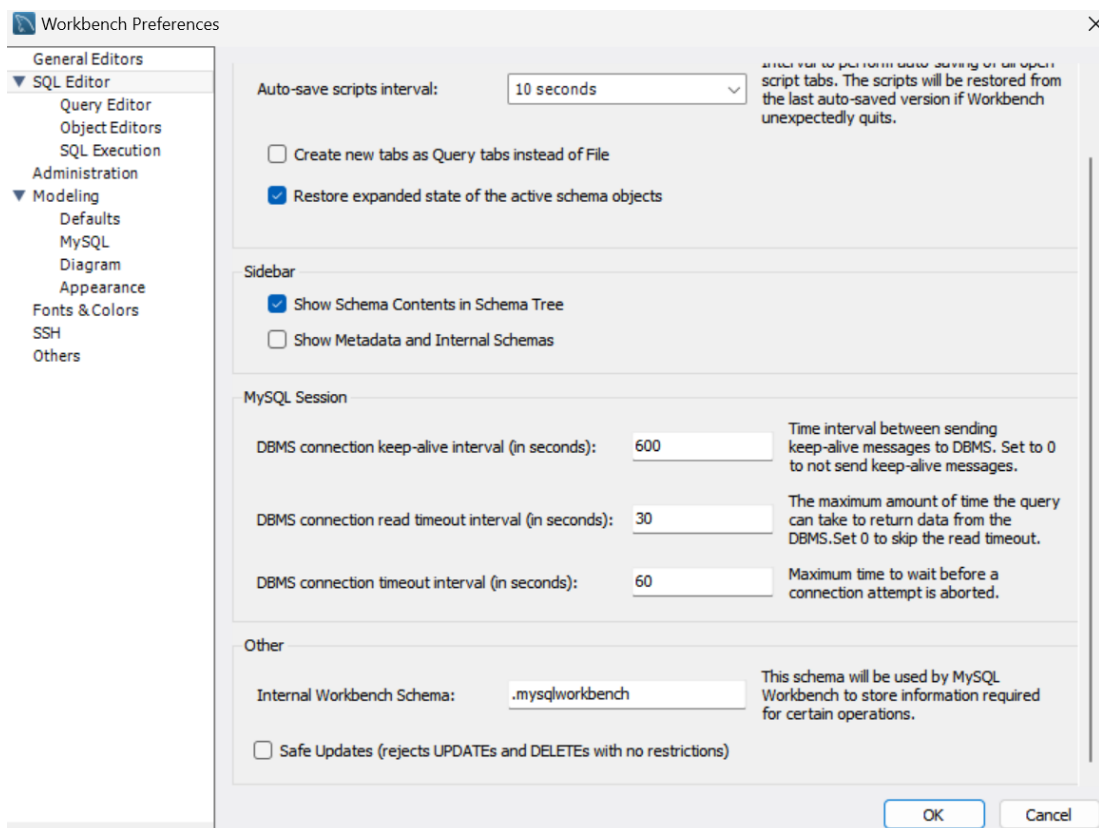
```
update student
set s_id = '09'
where s_name = '赵雷';
```

但是产生如下报错：

108 15:19:15 update student set s_id = '09' where s_name = '赵雷' Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses a KEY column. To disable safe mode, toggle the option in Preferences -> SQL Editor and re

Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses a KEY column. To disable safe mode, toggle the option in Preferences -> SQL Editor and reconnect.

解决这个问题的方法是关闭safe mode，并重连数据库



继续尝试更新：

53 15:35:58 delete from student where s_id = '01' Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (lab3.`score`, CONSTRAINT `score_ibfk_1` FOREIGN KEY (s_id) REFERENCES student (s_id))
54 15:36:07 update student set s_id = '09' where s_name = '赵雷' Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (lab3.`score`, CONSTRAINT `score_ibfk_1` FOREIGN KEY (s_id) REFERENCES student (s_id))

Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (lab3.`score`, CONSTRAINT `score_ibfk_1` FOREIGN KEY (s_id) REFERENCES student (s_id))

`score` 中的 `s_id` 指向 `student` 表中的 `s_id`, 更新 `student` 表中被指向数据会导致触发外键约束

3.5 修改或插入表中数据

考察 `check` 子句如何控制校验完整性

现向 `student` 表中插入一条数据, `check` 内容是性别为男或女, 我这里插入一条性别为武装直升机的数据

```
insert into student values('09' , 'Ironman' , '1990-01-20' , '武装直升机');
```



Error Code: 3819. Check constraint 'student_chk_1' is violated.

得到报错, 说明 `Check` 语句起作用了

3.6 定义一个 `assertion`

通过修改表中数据考察断言如何控制数据完整性

```
Create assertion score_range
check
(not exists (select * from score
where s_score>100));
```

```
/
8 Create assertion score_range
9 check
10 (not exists (select * from score
11 Where s_score>100));
12
```

这条语句在 `MySQL` 中报错, 说明 `MySQL` 并不支持断言语句, 这里应该使用触发器 `Trigger` 来控制。

3.7 定义一个 `trigger`

通过修改表中数据考察触发器如何起作用

`score` 原数据:

	s_id	c_id	s_score
	03	03	80
	04	01	50
	04	02	30
	04	03	20
	05	01	59
▶	05	02	59
	06	01	31
	06	03	34
	07	02	89
	07	03	98

##定义一个触发器，使得更新学生名单时，如果学生的成绩低于60分，则触发“help”捞捞，将学生的成绩自动更新为60

```
delimiter //
create trigger help
after insert on student for each row
begin
    update score
    set s_score = 60
    where s_score<60;
end;//
delimiter ;
```

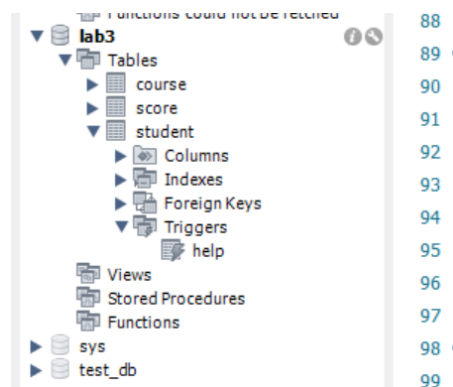
##默认情况下，不可能等到用户把这些语句全部输入完之后，再执行整段语句。

##因为 mysql一遇到分号，它就要自动执行。

##即，在语句RETURN ';'时，mysql解释器就要执行了。

##这种情况下，就需要 事先把delimiter换成其它符号，如//或\$\$。

```
update student
set s_sex = '女'
where s_name = '赵雷';
```



可以看到 `student` 增加了一个新的触发器 `help`

接下来开始更新 `student` 数据：

Result Grid				
Filter Rows:				
	s_id	s_name	s_birth	s_sex
▶	01	赵雷	1990-01-01	女
	02	钱电	1990-12-21	男
	03	孙风	1990-05-20	男
	04	李云	1990-08-06	男
	05	周梅	1991-12-01	女
	06	吴兰	1992-03-01	女
	07	郑竹	1989-07-01	女
	08	王菊	1990-01-20	女
✱	NULL	NULL	NULL	NULL

更新之后，看一下 `score` 表中的低于斩杀线的同学成绩：

Result Grid			
Filter Rows:			
	s_id	c_id	s_score
	03	01	80
	03	02	80
	03	03	80
	04	01	60
	04	02	60
	04	03	60
▶	05	01	60
	05	02	60
	06	01	60
	06	03	60
	07	02	89
	07	03	98

score 5 ✕

可以看到05号同学被捞捞了

4 遇到的问题及解决方法

```

delimiter //
create trigger help
after insert score for each row
begin
    update score
    set s_score = 60
    where s_score < 60;
end; //
delimiter ;

```

在执行插入语句对表修改时就产生了报错：

```

Error Code: 1442. Can't update table 'score' in stored function/trigger
because it is already used by statement which invoked this stored
function/trigger.

```

在定义触发器的时候还可以定义，但是一旦条件触发就会出错，命令执行停止：我本意是想通过触发器对同一个表进行监测和更新，但是好像不行。在MySQL中，触发器不允许在触发过程中对触发器所在的表进行修改，因为这可能会导致递归触发，产生无限循环的情况，只能在触发器中修改其他表。

如果真要处理这种情况，可以考虑使用存储过程。(网上搜的，我不会存储过程。)

5 总结

本次实验是对理论课上讲的sql语句的进一步实操，大部分是操作的约束性要求，包括但不限于外键约束、主键约束、触发器约束、**check**约束等语句，通过约束语句的操作可以更了解数据库中的逻辑关系，以便更好地学习数据库系统。同时非常感谢助教无私的帮助。