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：	卢佳盈
学　　号：	3180103570
电子邮箱：	ljy28501@163.com
联系电话：	18868703211
指导老师：	孙建伶

2020 年 3 月 21 日

目录

SQL数据完整性

一、实验目的

1. 熟悉通过SQL进行数据完整性控制的方法。

二、实验环境

1. 操作系统：Windows
2. 数据库管理系统： MySQL

三、实验流程

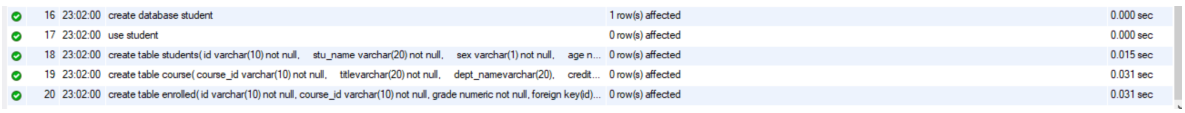
3.1 定义若干表

```
create database student;
use student;

create table students(
    id varchar(20) not null,
     stu_name varchar(20) not null,
     sex varchar(1) not null,
     age numeric,
     collage varchar(20),
      primary key(id),
      check (sex in ('M','F'))
);

create table course(
    course_id   varchar(10) not null,
     title       varchar(40) not null,
     dept_name   varchar(30),
     credits     numeric not null,
      primary key(course_id),
      check(credits>=0)
);

create table enrolled(
    id          varchar(20) not null,
    course_id   varchar(10) not null,
    grade       numeric not null,
    foreign key(id)     references student.students(id),
      foreign key(course_id) references student.course(course_id),
    check(grade>=0)
);
```



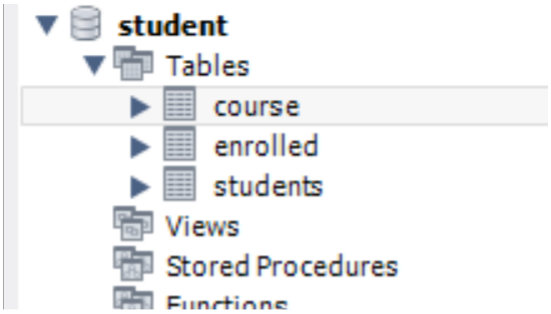


Table: enrolled

Columns:

id	varchar(20)
course_id	varchar(10)
grade	decimal(10,0)

Table: course

Columns:

course_id	varchar(10) PK
title	varchar(40)
dept_name	varchar(30)
credits	decimal(10,0)

Table: students

Columns:

id	varchar(20) PK
stu_name	varchar(20)
sex	varchar(1)
age	decimal(10,0)
collage	varchar(20)

3.2 表中插入数据

完整insert语句见附录1

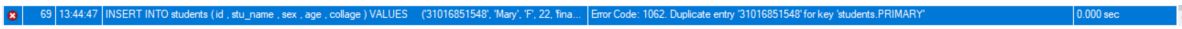
在执行下语句后

```
INSERT INTO students ( id , stu_name , sex , age , collage )
VALUES     ('31016851548', 'Ashley', 'F', 21, 'cs');
```

再插入一条主键id重复的记录：

```
INSERT INTO students ( id , stu_name , sex , age , collage )
VALUES     ('31016851548', 'Mary', 'F', 22, 'finance');
```

会报错　Error Code: 1062. Duplicate entry '31016851548' for key 'students.PRIMARY'



3.3 删除被引用表中的行

在原始数据插入时有以下记录：

```
INSERT INTO course ( course_id , title , dept_name , credits )
VALUES   ('19914', 'Chemical Enginerring', 'chemistry', 4);

INSERT INTO enrolled ( id , course_id , grade )
VALUES  ('30101218934', '19914', 83);
```

试图删除这门名为‘Chemical Enginerring’的课程，它的课程id为‘19914’

```
delete from course
where title='Chemical Enginerring';
```

报错　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.

此时报错是因为未关闭safe mode

重新执行删除语句：

```
-- 关闭safe mode
SET SQL_SAFE_UPDATES = 0;
-- 执行删除
delete from course
where title='Chemical Enginerring';
```

报错　Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`student.enrolled`, CONSTRAINT `enrolled_ibfk_2` FOREIGN KEY (`course_id`) REFERENCES `course` (`course_id`))

可以看出，除非将enrolled表中对应课程数据删除，否则无法删除父表数据

3.4 修改被引用表中的行的primary key

```
update students
set id='1990001000'
where stu_name='Ashley';
```

报错：　Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`student.enrolled`, CONSTRAINT `enrolled_ibfk_1` FOREIGN KEY (`id`) REFERENCES `students` (`id`))

3.5 修改或插入表中数据

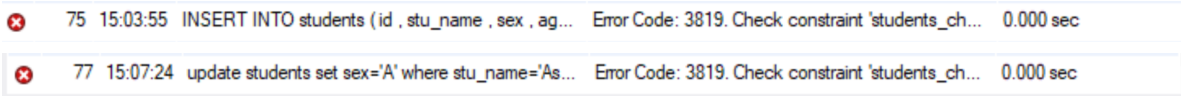
插入不符合check的数据

```
INSERT INTO students(id,stu_name,sex,age,collage)
VALUES      ('31016851000','May','A',25,'cs');
```

或将数据进行修改后不符合check子句

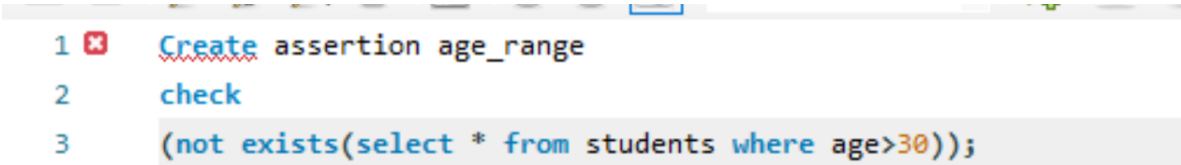
```
update students
set sex='A'
where stu_name='Ashley';
```

报错：　Error Code: 3819. Check constraint 'students_chk_1' is violated.



原因：在students表定义时，对sex字段进行了check子句约束，只能输入'M'、'F'值

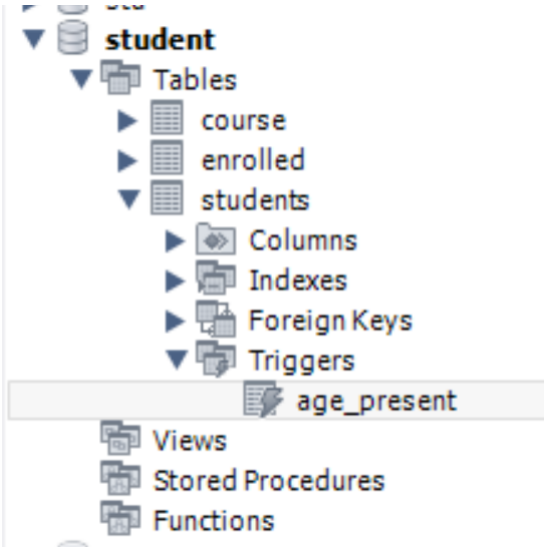
3.6 定义一个assertion



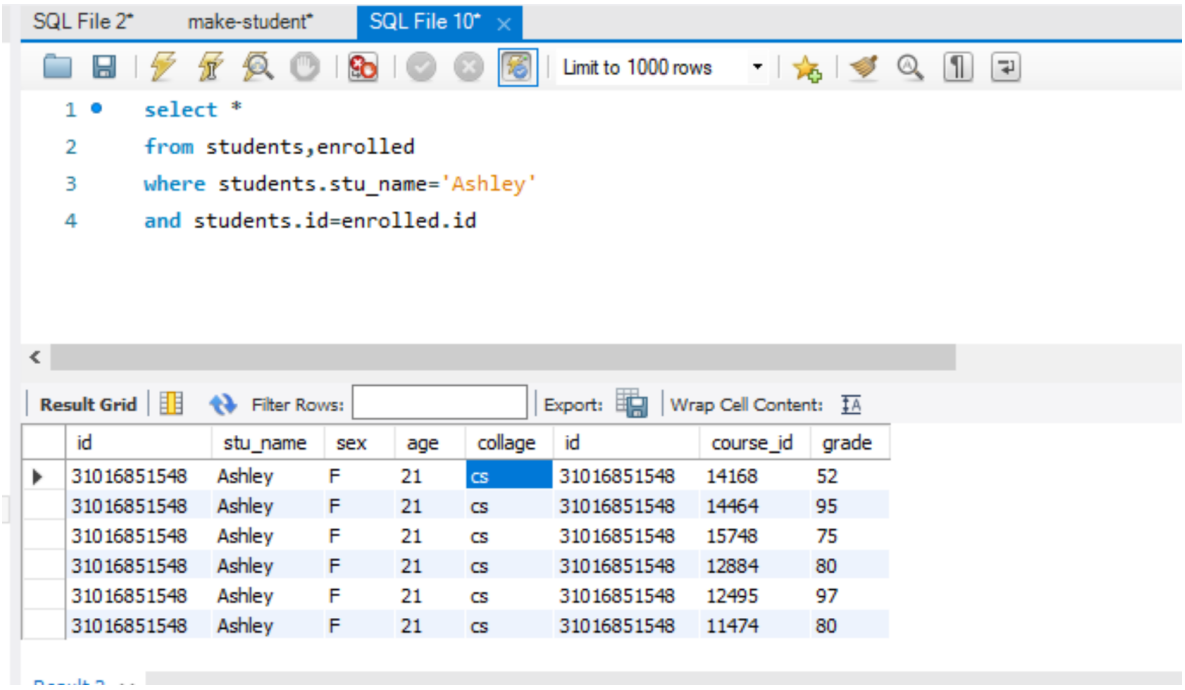
发现mysql不支持assertion语句

3.7 定义一个trigger

```
-- 在students表每次更新后，都对enrolled表进行修改，所有年龄在15岁以下的学生的所有成绩都改为100分
Delimiter $$
Create trigger age_present
    after update on students
      For each row
Begin
    Update enrolled
      set grade=100
      where enrolled.id in(select id from students where age<15);
end;$$
```



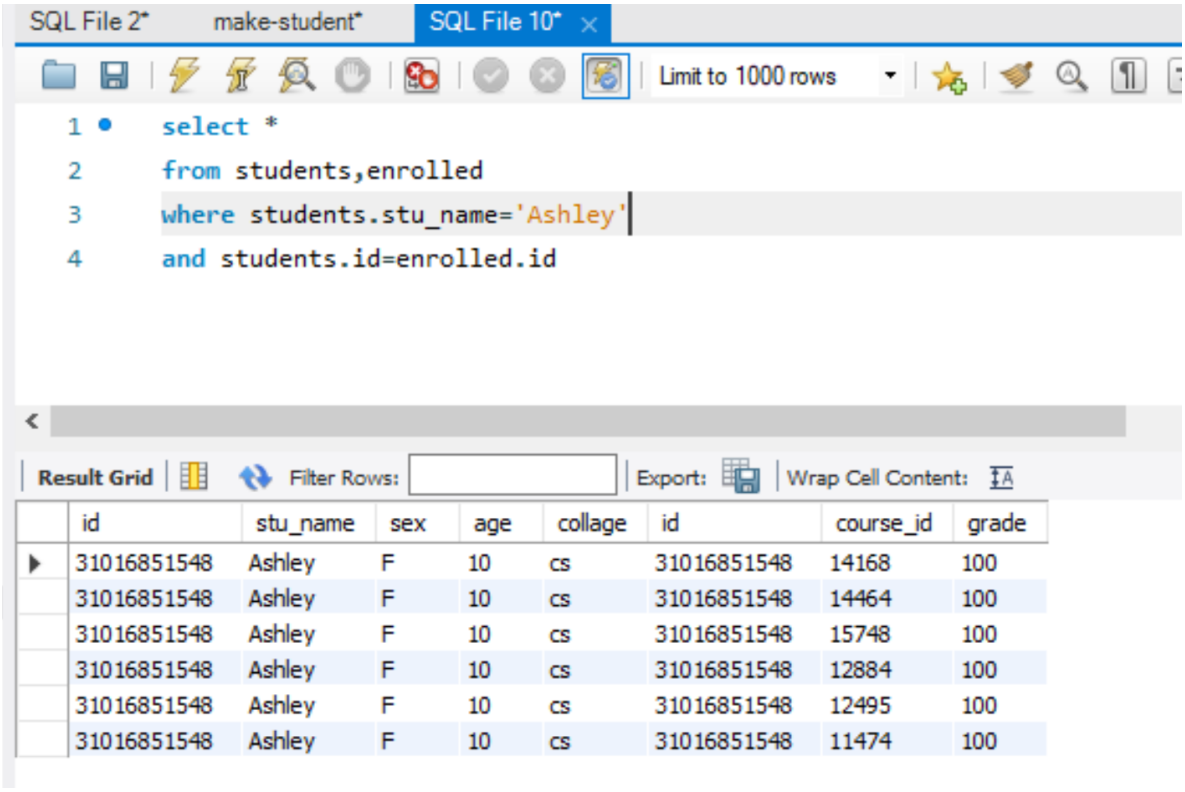
原数据：



id	stu_name	sex	age	collage	id	course_id	grade
31016851548	Ashley	F	21	cs	31016851548	14168	52
31016851548	Ashley	F	21	cs	31016851548	14464	95
31016851548	Ashley	F	21	cs	31016851548	15748	75
31016851548	Ashley	F	21	cs	31016851548	12884	80
31016851548	Ashley	F	21	cs	31016851548	12495	97
31016851548	Ashley	F	21	cs	31016851548	11474	80

对该同学的的年龄进行修改后：

```
update students
set age=10
where stu_name='Ashley';
```



发现该同学所有的成绩都变成了100

四、遇到的问题及解决方法

4.1 无法设置外键

```
create table course(
    course_id   varchar(10) not null,
     title       varchar(20) not null,
     dept_name   varchar(20),
     credits     numeric not null,
      primary key(course_id),
      foreign key(course_id) references enrolled(course_id),
      check(credits>=0)
);
```

错误提示

Error Code: 1822. Failed to add the foreign key constraint. Missing index for constraint 'course_ibfk_1' in the referenced table 'enrolled'

解决方案

设置外键的时候需要注意以下几点：

1. 外键是用于两个表的数据之间建立连接，可以是一列或者多列，即一个表可以有一个或多个外键。
2. 这个表里面设置的外键必须是另外一个表的主键！
3. 外键可以不是这个表的主键，但必须和另外一个表的主键相对应（字段的类型和值必须一样）。
4. 带有主键的那张表称为父表，含外键的是子表，必须先删除外键约束才能删除父表。

4.2 外键约束的设置具有顺序性

```
create table enrolled(
    id          varchar(10) not null,
    course_id   varchar(10) not null,
```

```
    grade       numeric not null,
    foreign key(id)     references students(id),
      foreign key(course_id) references course(course_id),
    check(grade>=0)
);

create table course(
    course_id   varchar(10) not null,
     title       varchar(20) not null,
     dept_name   varchar(20),
     credits     numeric not null,
      primary key(course_id),
      check(credits>=0)
);
```

错误提示

Error Code: 1824. Failed to open the referenced table 'course'

解决方案

先创建父表，再创建子表

4.3 有外键的子表对应的主表中没有数据

错误提示

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`student.enrolled`, CONSTRAINT `enrolled_ibfk_2` FOREIGN KEY (`course_id`) REFERENCES `course` (`course_id`))

解决方案

在父表中插入未能对应的外键值

五、总结

在这次实验中，我对于mysql中一些更加复杂的情况有了认识，比如说表格与表格间外键约束的要求。但是，在实验过程中，对于check与assertion的定义存在问题，经检验可以看出，尽管这些语法是SQL标准中的一部分，并被IBM DB2所支持，但在MySQL中不被支持，各个数据库系统对此都有不同的要求。

APPENDIX I 插入语句（原始数据）

```
use student;
INSERT INTO students ( id , stu_name , sex , age , collage )
VALUES
     ('31016851548', 'Ashley', 'F', 21, 'cs'),
     ('30753862502', 'Herman', 'M', 22, 'cs'),
     ('30278481255', 'Carl', 'M', 20, 'cs'),
     ('30548307063', 'Jessica', 'F', 19, 'cs'),
     ('30195312980', 'Sarah', 'F', 22, 'cs'),
     ('30260017728', 'Ives', 'M', 18, 'finance'),
     ('30664245771', 'Amanda', 'F', 21, 'finance'),
     ('30429686313', 'Charles', 'M', 24, 'finance'),
     ('30214513203', 'Brittany', 'F', 20, 'math'),
     ('30159019419', 'Felix', 'M', 23, 'math'),
     ('30111987529', 'Megan', 'F', 22, 'math'),
     ('30799855988', 'Jacob', 'M', 18, 'math'),
     ('31083056490', 'Lambert', 'M', 22, 'math'),
```

```
    ('30364533065', 'Jennifer', 'F', 22, 'sociology'),
    ('30192094390', 'Nicole', 'F', 19, 'sociology'),
    ('30101218934', 'Stephanie', 'F', 20, 'chemistry'),
    ('30292665702', 'Michael', 'M', 22, 'chemistry'),
    ('30351510431', 'Katherine', 'F', 18, 'physics'),
    ('30427841012', 'David', 'M', 22, 'sociology'),
    ('31057675425', 'Elaine', 'F', 22, 'physics'),
    ('30405511627', 'Hyman', 'M', 21, 'physics');

INSERT INTO course ( course_id , title , dept_name , credits )
VALUES
    ('11474', 'Military Theory', 'sociology', 2),
    ('12495', 'Advanced Mathematics', 'math', 5),
    ('12884', 'Linear Algebra', 'math', 4),
    ('12943', 'Functions of Complex Variables', 'math', 3),
    ('13829', 'Lab of General Physics', 'physics', 2),
    ('14168', 'Data Structure', 'cs', 3),
    ('14464', 'Digital Image Processing', 'cs', 3),
    ('14833', 'Basis of Software Technique', 'cs', 3),
    ('15748', 'C Language', 'cs', 3),
    ('16481', 'College Physics', 'physics', 4),
    ('16878', 'Optics', 'physics', 3),
    ('17120', 'Enterprise Management', 'finance', 4),
    ('17560', 'Microeconomics', 'finance', 3),
    ('17947', 'Bisic Accounting', 'finance', 3),
    ('19103', 'Organic Chemistry', 'chemistry', 4),
    ('19123', 'General Chemistry', 'chemistry', 3),
    ('19914', 'Chemical Enginerring', 'chemistry', 4);

INSERT INTO enrolled ( id , course_id , grade )
VALUES
    ('31016851548', '14168', 52),
    ('31016851548', '14464', 95),
    ('31016851548', '15748', 75),
    ('31016851548', '12884', 80),
    ('31016851548', '12495', 97),
    ('31016851548', '11474', 80),
    ('30753862502', '15748', 79),
    ('30753862502', '14168', 73),
    ('30753862502', '11474', 55),
    ('30278481255', '14168', 76),
    ('30278481255', '14464', 97),
    ('30278481255', '14833', 68),
    ('30278481255', '13829', 52),
    ('30278481255', '16481', 83),
    ('30548307063', '14464', 96),
    ('30548307063', '14833', 57),
    ('30548307063', '15748', 81),
    ('30548307063', '17560', 94),
    ('30195312980', '14168', 80),
    ('30195312980', '11474', 50),
    ('30195312980', '12495', 57),
    ('30195312980', '16481', 78),
    ('30260017728', '17120', 57),
    ('30260017728', '17560', 84),
```

```
    ('30260017728', '17947', 92),
    ('30260017728', '12495', 78),
    ('30260017728', '15748', 54),
    ('30664245771', '17560', 75),
    ('30664245771', '17947', 82),
    ('30664245771', '12495', 72),
    ('30664245771', '16481', 83),
    ('30429686313', '17120', 59),
    ('30429686313', '17947', 59),
    ('30214513203', '12495', 50),
    ('30214513203', '12884', 54),
    ('30214513203', '12943', 89),
    ('30159019419', '12495', 74),
    ('30159019419', '12884', 74),
    ('30159019419', '12943', 95),
    ('30159019419', '13829', 77),
    ('30111987529', '12495', 90),
    ('30111987529', '12884', 86),
    ('30111987529', '14464', 87),
    ('30111987529', '17947', 51),
    ('30799855988', '16481', 61),
    ('30799855988', '12495', 78),
    ('31083056490', '12884', 91),
    ('31083056490', '12943', 96),
    ('31083056490', '13829', 89),
    ('30364533065', '11474', 85),
    ('30364533065', '17560', 98),
    ('30192094390', '11474', 66),
    ('30101218934', '19103', 77),
    ('30101218934', '19123', 58),
    ('30101218934', '19914', 83),
    ('30101218934', '12495', 100),
    ('30292665702', '19103', 89),
    ('30292665702', '19123', 84),
    ('30292665702', '11474', 82),
    ('30292665702', '16481', 61),
    ('30351510431', '13829', 59),
    ('30351510431', '16878', 56),
    ('30351510431', '12943', 61),
    ('30427841012', '11474', 77),
    ('30427841012', '17120', 83),
    ('31057675425', '16481', 77),
    ('31057675425', '16878', 80),
    ('31057675425', '15748', 66),
    ('30405511627', '19103', 64),
    ('30405511627', '16481', 77),
    ('30405511627', '13829', 69);
```