

浙江大学实验报告

课程名称： 数据库系统原理 实验类型：
实验项目名称： SQL 数据完整性
学生姓名： 彭子帆 专业： 软件工程 学号： 3170105860
同组学生姓名： 指导老师： 周波
实验地点： 紫金港机房 实验日期： 2019 年 4 月 3 日

一、 实验目的和要求

1. 熟悉通过 SQL 进行数据完整性控制的方法。

二、 实验内容和原理

1. 定义若干表，其中包括 primary key, foreign key 和 check 的定义。
2. 让表中插入数据，考察 primary key 如何控制实体完整性。
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。
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。
5. 修改或插入表中数据，考察 check 子句如何控制校验完整性。
6. 定义一个 assertion，并通过修改表中数据考察断言如何控制数据完整性。
7. 定义一个 trigger，并通过修改表中数据考察触发器如何起作用。

三、 主要仪器设备

1. 操作系统: Windows 10 & 64 bits
2. 数据库管理系统 : MySQL Community 8.0.15

四、 操作方法与实验步骤

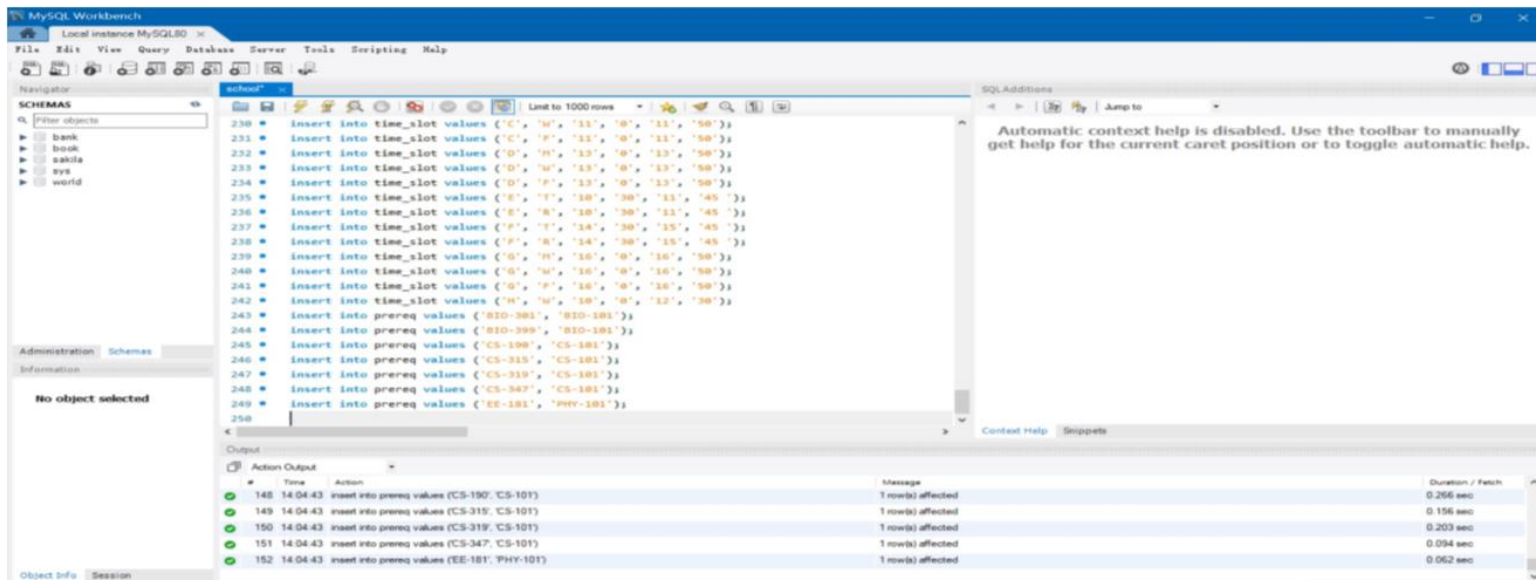
1. 4.1 数据表定义 数据表定义

在本次实验中，我们以教材配套网站上给出的学校数据库为操作对象。由于实验环境，选用网上提供的 DDL-My SQL.sql 和 smallRelationInsertFile.sql 两个脚本文件来创建小型数据库。我们分析其中的 department 和 instructor 表的约束条件。

```
1. create table department (  
2. dept_name varchar(20),  
3. building varchar(15),  
4. budget numeric(12,2) check (budget > 0),  
5. primary key (dept_name) );  
6.  
7. create table instructor (  
8. ID varchar(5),  
9. name varchar(20) not null,  
10. dept_name varchar(20),  
11. salary numeric(8,2) check (salary > 29000), primary key (ID),  
12. foreign key (dept_name) references department(dept_name)  
13. on delete set null  
14. on update cascade );
```

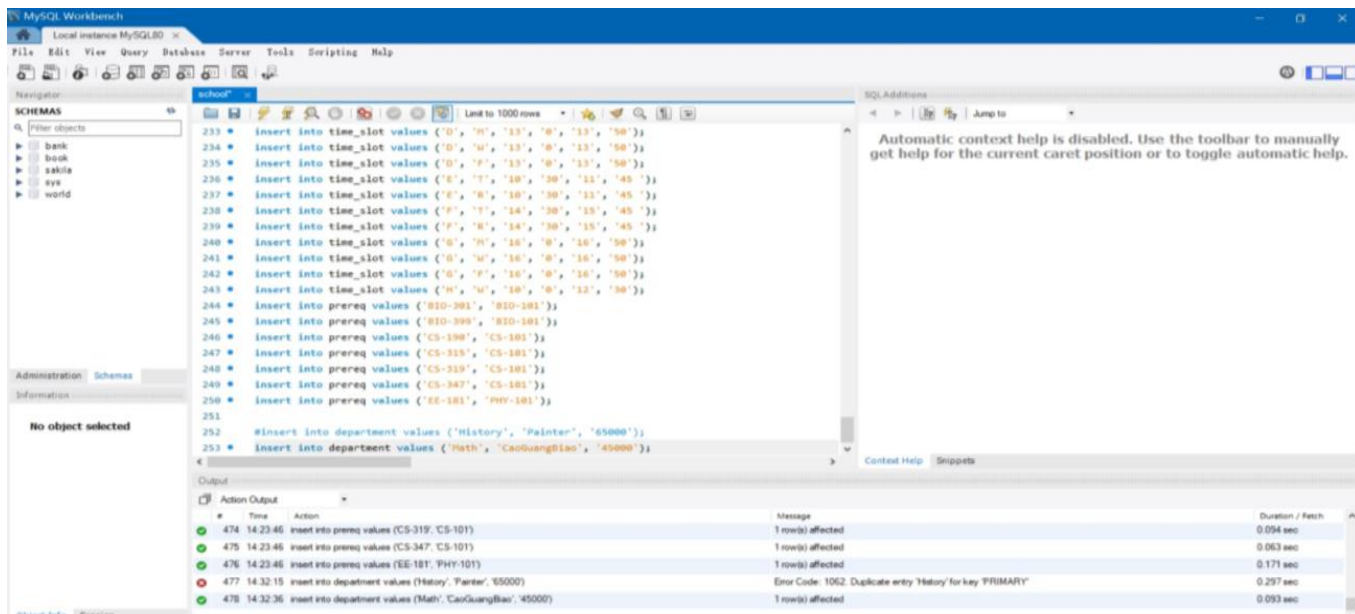
department 表中定义了主键 dept_name，instructor 表中定义了主键 ID，说明了两张表中，每个 dept_name 和 ID 均只能出现在一行中。而外码声明说明了 department 是 instructor 的参照关系，故 instructor 中的每个必须在 department 之中出现过才可以，因此在执行数据插入操作时，我们应先插入 department 中的元素，再插入 instructor 中的元素更合理。instructor 中的 on delete 和 on update 子句规定了当 department 中的某个 dept_name 属性被删除时，instructor 中取该值的 dept_name 应被置为 null；当

department 中的某个 dept_name 属性被更改时, instructor 中取该值的 dept_name 应当进行同步修改。两张表的 check 子句规定了 budget 和 salary 两个属性的限定取值范围。表定义与数据插入结果如下图所示, 可见操作成功。



2. 考察 check 子句

结果显示, 两条数据的插入都成功了, 这不符合我们的预期。经过查询得知: MySQL 所有的存储引擎均不支持 check 约束, MySQL 会对 check 子句进行分析, 但是在插入数据时会忽略, 因此 check 并不起作用。想要在 MySQL 中实现 check 约束的效果可以考虑使用 enum 约束或者触发器 trigger 来实现。在 4.7 中, 我们尝试使用 trigger 来实现等效的 check 效果。



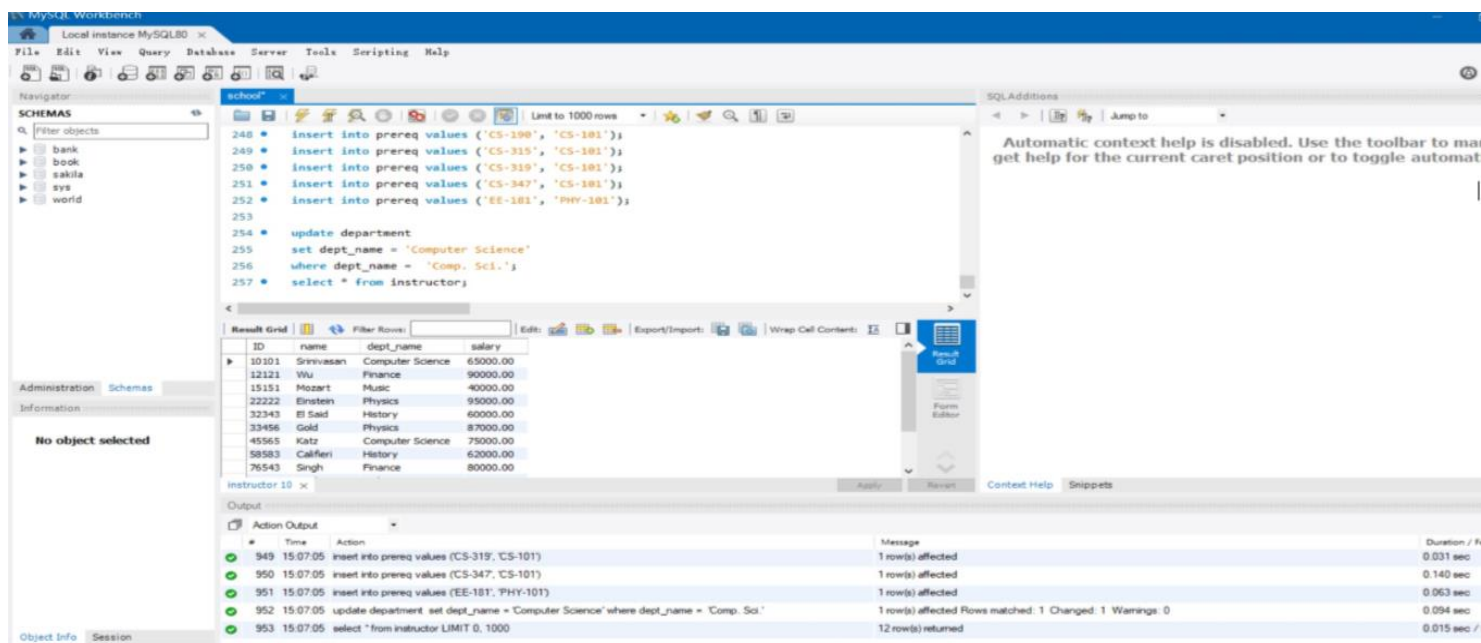
3. 删除行，考察 on delete on delete 控制参照完整性 控制参照完整性

仍旧以上面分析过的 department 和 instructor 两张表为例。首先我们在 instructor 表中查询计算机系的教授信息，之后我们在 department 表中删除计算机系，再次进行上述查询。

操作结果如下图所示。由两次查询结果的对比可见，在删除计算机系之时，由于 on delete 子句的缘故，instructor 中系别为计算机系的教授的院系属性都被设置为了 null 因此执行与删除前的查询完全形同的查询得不到结果，由第三张图的结果也可见，三位计算机系教授的院系已被置为了 null。

4. 修改 primary key，考察 on update 控制参照完整

给两外两张表添加了 on update cascade 之后重复操作，发现这次的操作成功了，各个教授的院系属性成功的得到了更新：



5. 考察 check 子句

经过查询得知：MySQL 所有的存储引擎均不支持 check 约束，MySQL 会对 check 子句进行分析，但是在插入数据时会忽略，因此 check 并不起作用。想要在 MySQL 中实现 check 约束的效果可以考虑使用 enum 约束或者触发器 trigger 来实现。在 4.7 中，我们尝试使用 trigger 来实现等效的 check 效果。

6. asseration 断言定义与相关操作

同样，MySQL 同样不支持 asseration 的功能，甚至 MySQL 中没有规定 asseration 为关键字，因此我们甚至不能定义一个 asseration。不过我们知道，断言的本质是较为

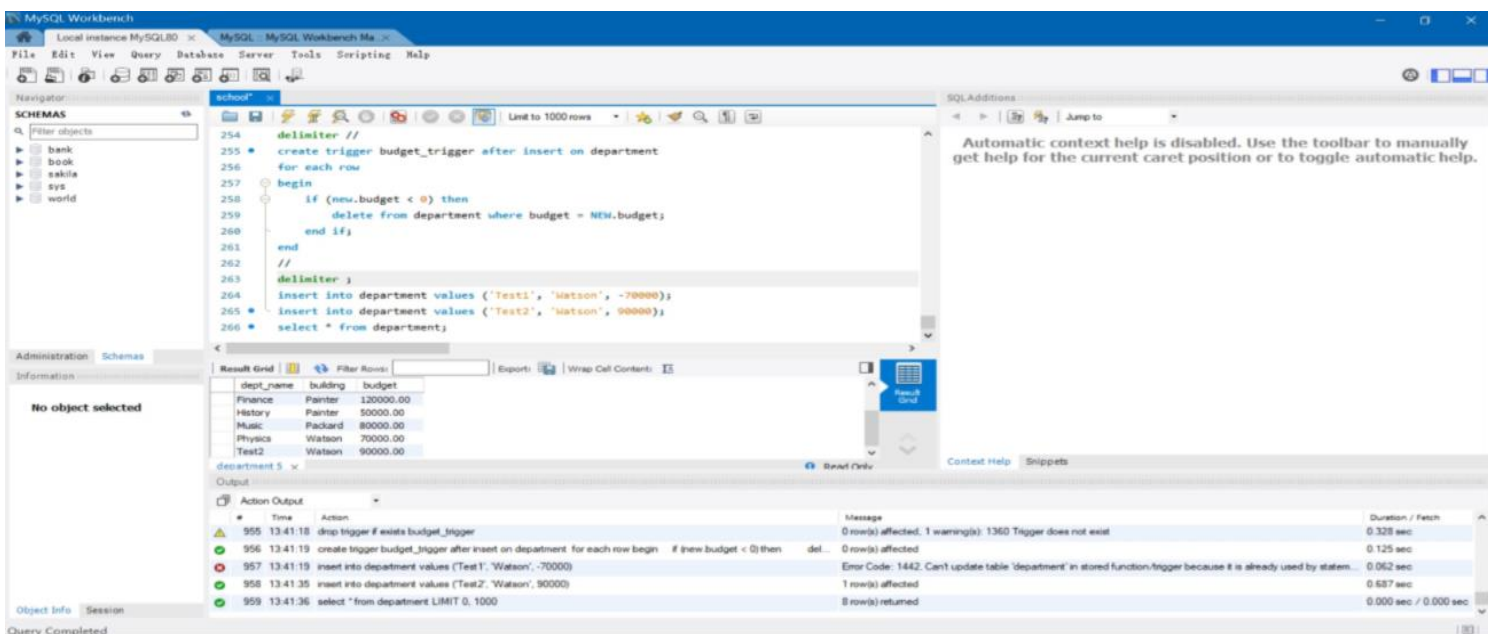
复杂的 check，因此要实现类似的功能，我们也可以通过 trigger 来实现。有这种利用 trigger 的方式替代 asseration 的功能，这或许也是 MySQL 没有引入 asseration 语句的原因

7. trigger 触发器定义与相关操作

在 MySQL 中，默认的句子结束符号是分号。由于在触发器的内部定义中可能存在 delete, update 等等子操作，其中是存在分号的，这会中止当前语句的定义，从而使触发器的定义不完整而执行错误。因此，我们需要使用 delimiter 命令来暂时更改语句结束符号，在这之后再次调用该语句使语句结束符号变回分号：

```
1. create trigger budget_trigger after insert on department
2. for each row
3. begin
4.     if (new.budget < 0) then
5.         delete from department where budget = new.budget;
6.     end if;
7. end
```

分析这个触发器可知：这个触发器会在数据插入之后触发，检测新插入的行的 budget 属性是否符合为正的要求， 如果不符合则会删除这一行。通过下方的运行截图可以看到，触发器的定义成功了，插入不符合要求数据的行为会失败，而插入其他数据则不受影响，通过查询结果也可得知插入 Test2 一行是成功。



五、 讨论、心得

在本次实验中，我们进行了 MySQL 数据完整性的测试，包括了 primary key ， foreign key ， on delete ， on update 子句，并利用 trigger 实现了 check 和 asseration 的功能。通过本实验，我们可以认识到数据库设计并非易事，当 数据表增多的时候，各张表之间的关系会变得愈发复杂，难以辨明，因此约束数据完整性的控制至关重要，今后在利 用数据库，处理 SQL 问题时，应当更加小心谨慎

六、 附录：数据表与插入数据

下载地址: https://www.db-book.com/db6/lab-dir/sample_tables-dir/index.html