**实验3 DBMS的安装和使用**

**姓名：段皞一**

**学号：3190105359**

**专业：计算机科学与技术**

**一 实验目的**

1. 熟悉通过SQL进行数据完整性控制的方法。

**二 实验平台**

1. 操作系统： Windows
2. 数据库管理系统：SQL Server

**三 实验内容和要求**

**3.1定义若干表，其中包括primary key, foreign key和check的定义**

打开Microsoft SQL Server Management Studio, 连接自己的服务器，建立一个基于教材的详细的大学模式数据库。



在完成数据库University的建立后，在数据库的表选项中添加相应的一系列表。也可以使用SQL语言直接在新建查询中进行表的新建。

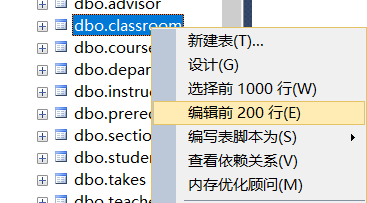
本次实验，采用SQL语句和图形界面结合的方式，进行表的创建。

**3.1.1创建表classroom**

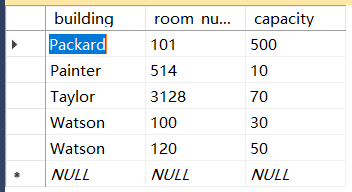
创建表“classroom”.相关代码如下所示：

|  |
| --- |
| create table classroom  (  building varchar(15),  room\_number varchar(7),  capacity numeric(4,0),  primary key (building, room\_number)  ); |

表建成以后，右键dbo.classroom单击选择“编辑前200行”选项，可以使用图形界面进行字段的添加。



完成添加字段。效果图如下：

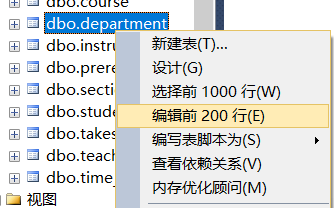


**3.1.2创建表department**

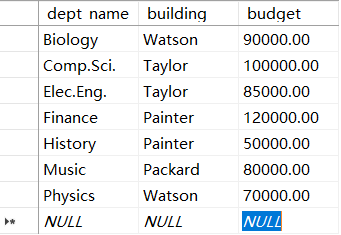
创建表”department”,相关SQL语句如下所示：

|  |
| --- |
| create table department  (  dept\_name varchar(20),  building varchar(20),  budget numeric(12,2) check (budget > 0),  primary key (dept\_name)  ); |

创建表成功后，可以在表的栏目下查找到”dbo.department”记录。右键以后，在弹出的窗口栏中选择“编辑前200行”选项，即可进行该表相应字段的添加。



该表字段添加完成以后的示意图如下所示：

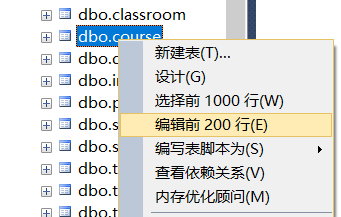


**3.1.3创建表course**

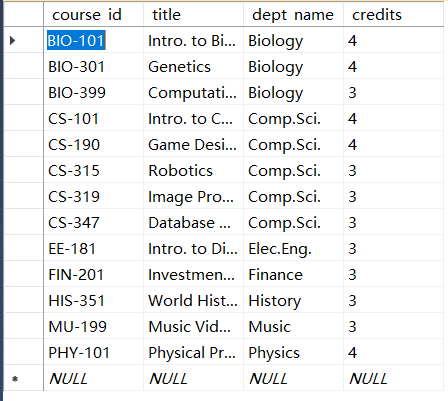
该表创建的相关代码如下所示：

|  |
| --- |
| create table course  (  course\_id varchar(8),  title varchar(50),  dept\_name varchar(20),  credits numeric(2, 0) check (credits > 0),  primary key (course\_id),  foreign key (dept\_name) references department  on delete set null  ); |

创建成功以后，可以在表窗口栏下查到dbo.course.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



添加成功后的效果图如下：

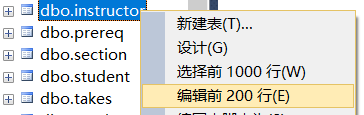


**3.1.4创建表instructor**

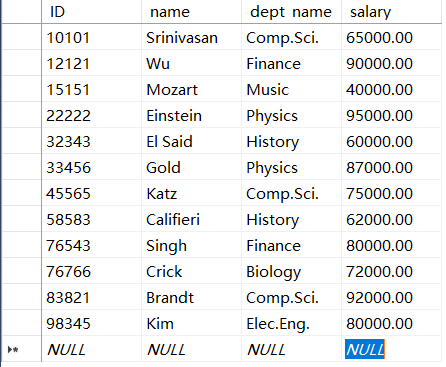
该表的创建代码如下所示：

|  |
| --- |
| create table instructor  (  ID varchar(5),  name varchar(20) not null,  dept\_name varchar(20),  salary numeric (8,2) check (salary > 29000),  primary key(ID),  foreign key (dept\_name) references department  on delete set null  ); |

创建成功以后，可以在表窗口栏下查到dbo.instructor.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



字段添加完成以后的效果图如下：

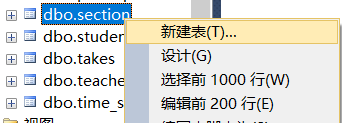


**3.1.5创建表section**

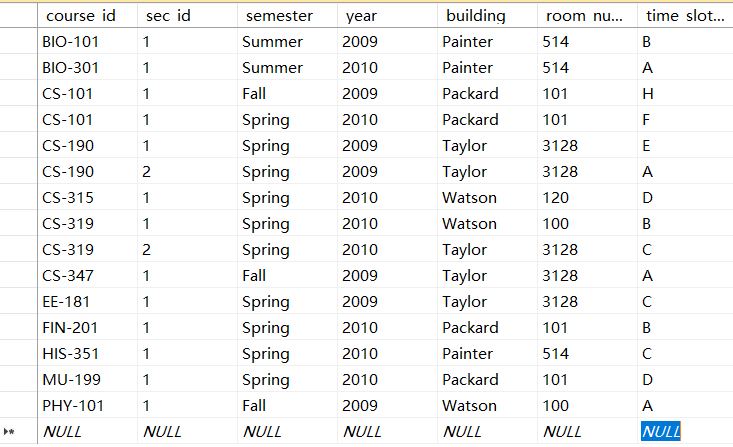
该表的创建的代码如下：

|  |
| --- |
| create table section  (  course\_id varchar(8),  sec\_id varchar(8),  semester varchar(6) check (semester in  ('Fall', 'Winter', 'Spring', 'Summer')),  year numeric(4, 0) check (year > 1701 and year < 2100),  building varchar(15),  room\_number varchar(7),  time\_slot\_id varchar(4),  primary key (course\_id, sec\_id, semester, year),  foreign key (course\_id) references course  on delete cascade,  foreign key (building, room\_number) references classroom  on delete set null  ); |

创建成功以后，可以在表窗口栏下查到dbo.section.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



添加字段以后的示意图如下：

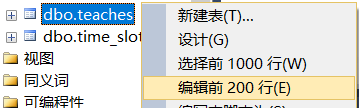


**3.1.6创建表teaches**

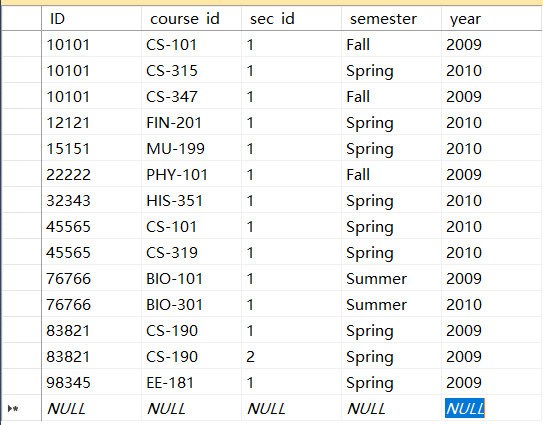
创建表的相关代码如下：

|  |
| --- |
| create table teaches  (  ID varchar(5),  course\_id varchar(8),  sec\_id varchar(8),  semester varchar(6),  year numeric(4, 0),  primary key (ID, course\_id, sec\_id, semester, year),  foreign key (course\_id, sec\_id, semester, year) references section  on delete cascade,  foreign key (ID) references instructor  on delete cascade  ); |

创建成功以后，可以在表窗口栏下查到dbo.teaches.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



添加字段完成以后的示意图如下：

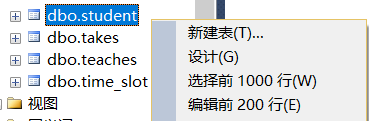


**3.1.7创建表student**

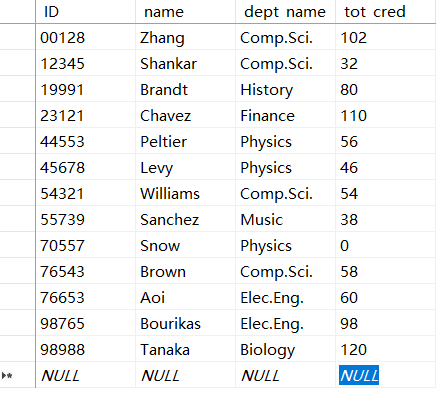
创建相关表的相应代码如下所示：

|  |
| --- |
| create table student  (  ID varchar(5),  name varchar(20) not null,  dept\_name varchar(20),  tot\_cred numeric(3, 0) check(tot\_cred >= 0),  primary key (ID),  foreign key (dept\_name) references department  on delete set null  ); |

创建成功以后，可以在表窗口栏下查到dbo.student.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



编辑字段成功后，可以看到相应的示意图如下：

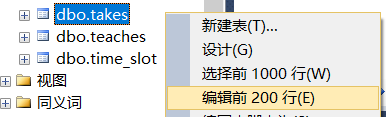


**3.1.8创建表takes**

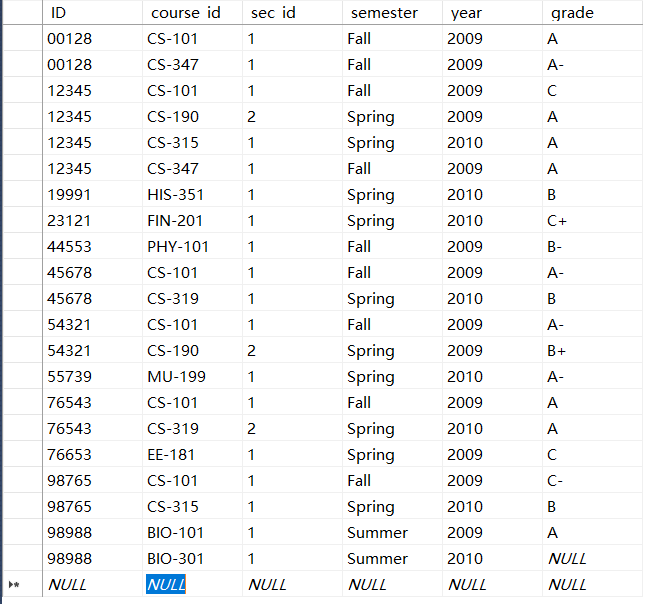
该表创建的相关代码如下所示：

|  |
| --- |
| create table takes  (  ID varchar(5),  course\_id varchar(8),  sec\_id varchar(8),  semester varchar(6),  year numeric(4, 0),  grade varchar(2),  primary key (ID, course\_id, sec\_id, semester, year),  foreign key (course\_id, sec\_id, semester, year) references section  on delete cascade,  foreign key(ID) references student  on delete cascade  ); |

创建成功以后，可以在表窗口栏下查到dbo.takes.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



完成字段的添加以后，相关示意图如下：

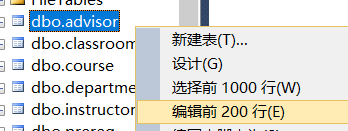


**3.1.9创建表advisor**

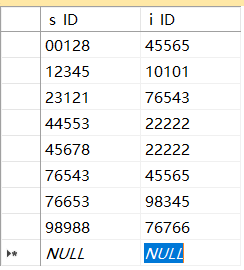
该表创建的相关代码如下：

|  |
| --- |
| create table advisor  (  s\_ID varchar(5),  i\_ID varchar(5),  primary key (s\_ID),  foreign key (i\_ID) references instructor(ID)  on delete set null,  foreign key (s\_ID) references student(ID)  on delete cascade  ); |

创建成功以后，可以在表窗口栏下查到dbo.advisor.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



完成字段的添加以后，相关示意图如下：

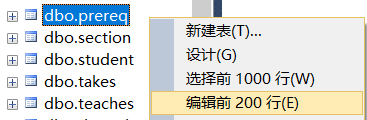


**3.1.10创建表prereq**

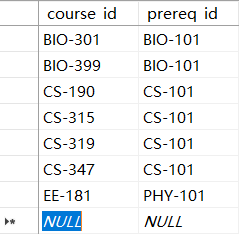
创建表的相关代码如下：

|  |
| --- |
| create table prereq  (  course\_id varchar(8),  prereq\_id varchar(8),  primary key (course\_id, prereq\_id),  foreign key (course\_id) references course  on delete cascade,  foreign key (prereq\_id) references course  ); |

创建成功以后，可以在表窗口栏下查到dbo.prereq.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



完成字段的添加以后，可以看到示意图如下：

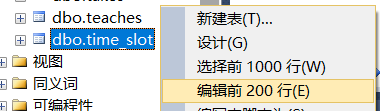


**3.1.11创建表time\_slot**

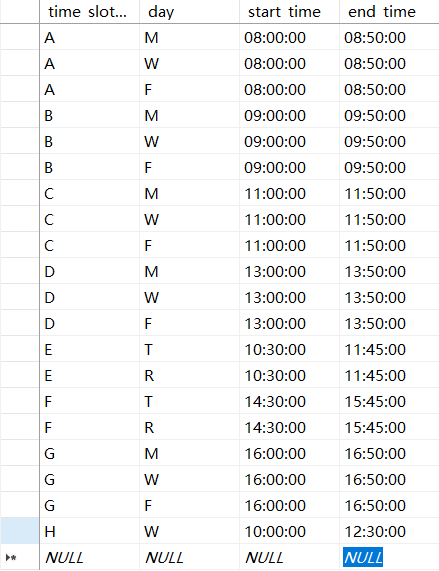
该表创建的相关代码如下：

|  |
| --- |
| create table time\_slot  (  time\_slot\_id varchar(4),  day varchar(1) check (day in ('M', 'T', 'W', 'R', 'F', 'S', 'U')),  start\_time time,  end\_time time,  primary key(time\_slot\_id, day, start\_time)  ); |

创建成功以后，可以在表窗口栏下查到dbo.time\_slot.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



完成该表字段的添加以后，可以看到相关的示意图如下：



**3.2让表中插入数据，考察primary key如何控制实体完整性**

新建查询，书写下面指令，加入学号为10101的Mike同学字段。相关代码如下：

|  |
| --- |
| insert student(ID, name, dept\_name, tot\_cred)  values (10101, 'Mike', 'Biology', 4) |

在下表中出现了新的字段Mike, 新建字段成功。



在书写以下代码，尝试加入学号同样为10101的Shan同学。相关代码如下：

|  |
| --- |
| insert student(ID, name, dept\_name, tot\_cred)  values (10101, 'Shan', 'Physics', 5) |

执行语句后，发现存在以下错误：

*消息 2627，级别 14，状态 1，第 1 行*

*违反了 PRIMARY KEY 约束“PK\_\_student\_\_3214EC2712D27239”。不能在对象“dbo.student”中插入重复键。重复键值为 (10101)。*

*语句已终止。*

**

**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**

尝试删除表course中的course\_id为‘BIO-101’的字段，输入以下代码：

|  |
| --- |
| delete from course  where course\_id = 'BIO-101' |

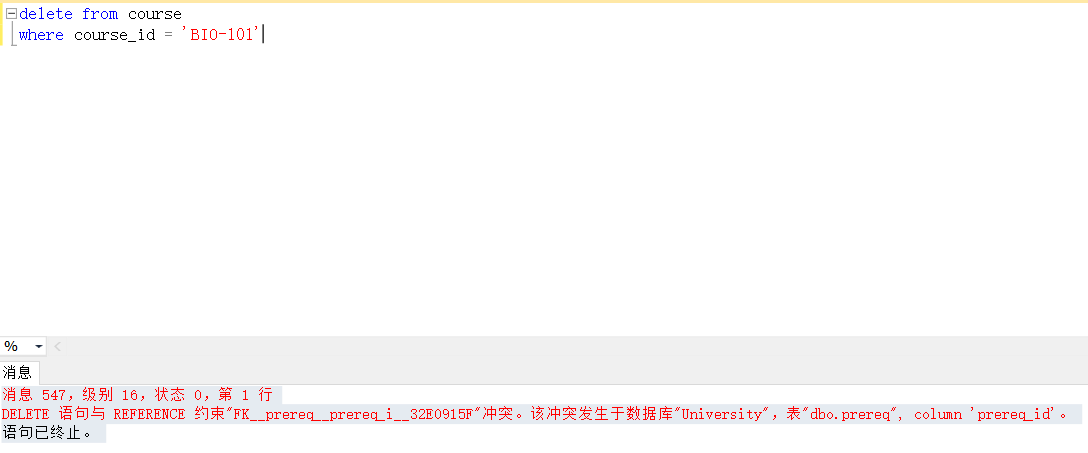
可以看到报出以下的错误，

*消息 547，级别 16，状态 0，第 1 行*

*DELETE 语句与 REFERENCE 约束"FK\_\_prereq\_\_prereq\_i\_\_32E0915F"冲突。该冲突发生于数据库"University"，表"dbo.prereq", column 'prereq\_id'。*

*语句已终止。*

示意图如下：



**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**

为了后续实验的方便，首先定义几个新的表。



|  |
| --- |
| CREATE TABLE customer  (customer\_name char(20),  customer\_street char(30),  customer\_city char(30),  primary key (customer\_name));  CREATE TABLE branch  (branch\_name char(15),  branch\_city char(30),  assets integer,  primary key (branch\_name),  CHECK (assets>=0));  CREATE TABLE account  (account\_number char(10),  branch\_name char(15),  balance integer,  primary key (account\_number),  foreign key (branch\_name) references branch,  CHECK(balance>=0));  CREATE TABLE depositor  (customer\_name char(20),  account\_number char(10),  primary key (customer\_name, account\_number),  foreign key (account\_number) references account,  foreign key (customer\_name) references customer); |

account表中的外码没有加入on update子句进行约束，所以书写以下语句进行account中branch\_name的修改是可以的。

但是当我们定义一下account\_new表示，注意加入了on update子句：

|  |
| --- |
| CREATE TABLE account  (account\_number char(10),  branch\_name char(15),  balance integer,  primary key (account\_number),  foreign key (branch\_name) references branch  on update cascade,  CHECK(balance>=0)); |

此时当输入以下语句，视图进行account\_new表中的branch\_name的修改：

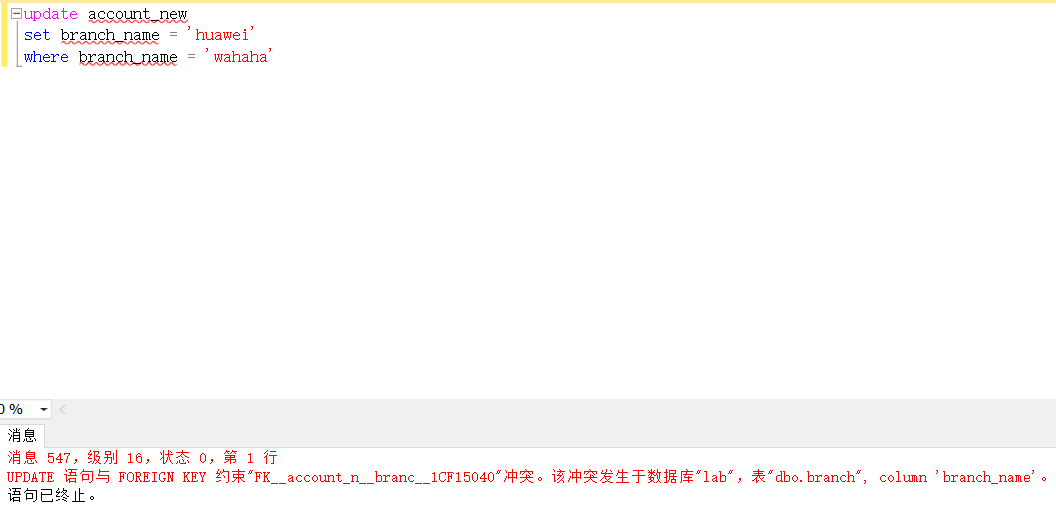
|  |
| --- |
| update account\_new  set branch\_name = 'huawei'  where branch\_name = 'wahaha' |

系统将会报出以下的错误：

*消息 547，级别 16，状态 0，第 1 行*

*UPDATE 语句与 FOREIGN KEY 约束"FK\_\_account\_n\_\_branc\_\_1CF15040"冲突。该冲突发生于数据库"lab"，表"dbo.branch", column 'branch\_name'。*

语句已终止。



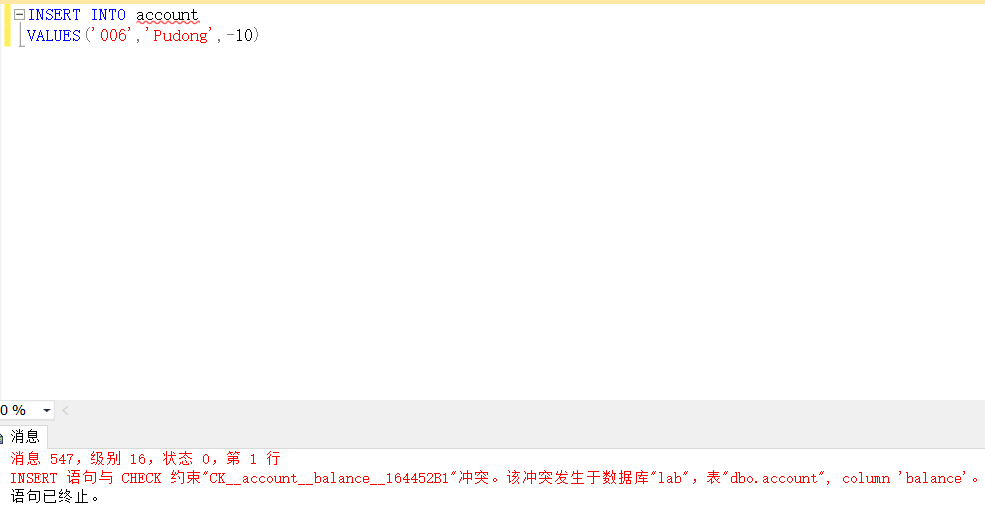
**3.5修改或插入表中数据，考察check子句如何控制校验完整性**

|  |
| --- |
| INSERT INTO account  VALUES('006','Pudong',-10) |

*消息 547，级别 16，状态 0，第 1 行*

*INSERT 语句与 CHECK 约束"CK\_\_account\_\_balance\_\_164452B1"冲突。该冲突发生于数据库"lab"，表"dbo.account", column 'balance'。*

*语句已终止。*



**3.6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**

下面是一个断言定义的示例：

|  |
| --- |
| CREATE ASSERTION assertion\_bal  CHECK (not exists(SELECT \* FROM account  WHERE balance>5000)); |

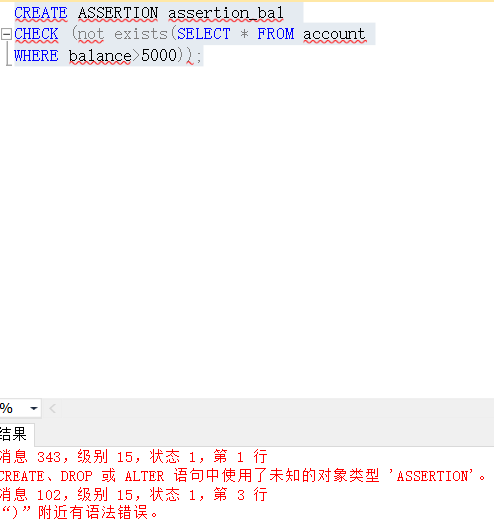
但是遗憾的是，本实验使用的SQL Server貌似不支持ASSERSION功能，所以不能在服务器上进行实验的验证。

*消息 343，级别 15，状态 1，第 1 行*

*CREATE、DROP 或 ALTER 语句中使用了未知的对象类型 'ASSERTION'。*

*消息 102，级别 15，状态 1，第 3 行*

*“)”附近有语法错误。*



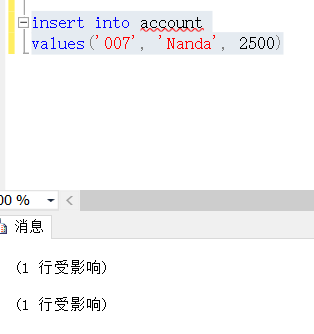
**3.7定义一个trigger, 并通过修改表中数据考察触发器如何起作用**

定义一个trigger, 并通过修改数据考察触发器是如何起作用的。

|  |
| --- |
| CREATE TRIGGER trig  ON account  AFTER INSERT AS  IF(SELECT count(\*)FROM account)=7  BEGIN  UPDATE account  SET balance=1.1\*balance  WHERE account\_number=(SELECT account\_number FROM INSERTED)  END |

定义好触发器以后，当插入新的一个字段，且字段总数已经有7个的时候，将触发触发器，对新插入的字段进行balance乘上1.1的操作，输入下面的代码，作为该表的第八个输入，

|  |
| --- |
| insert into account  values('007', 'Nanda', 2500) |



可以发现插入的字段，其balance增加了10%，从输入端的2500到输出端的2750。

