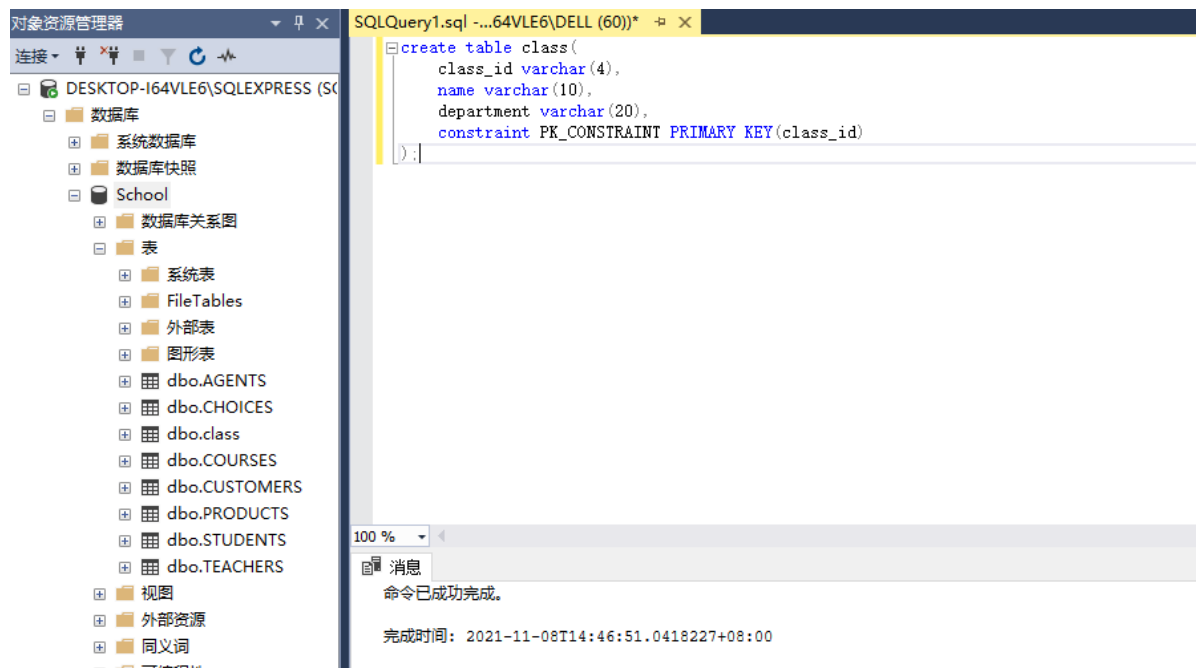


# 第九次实验

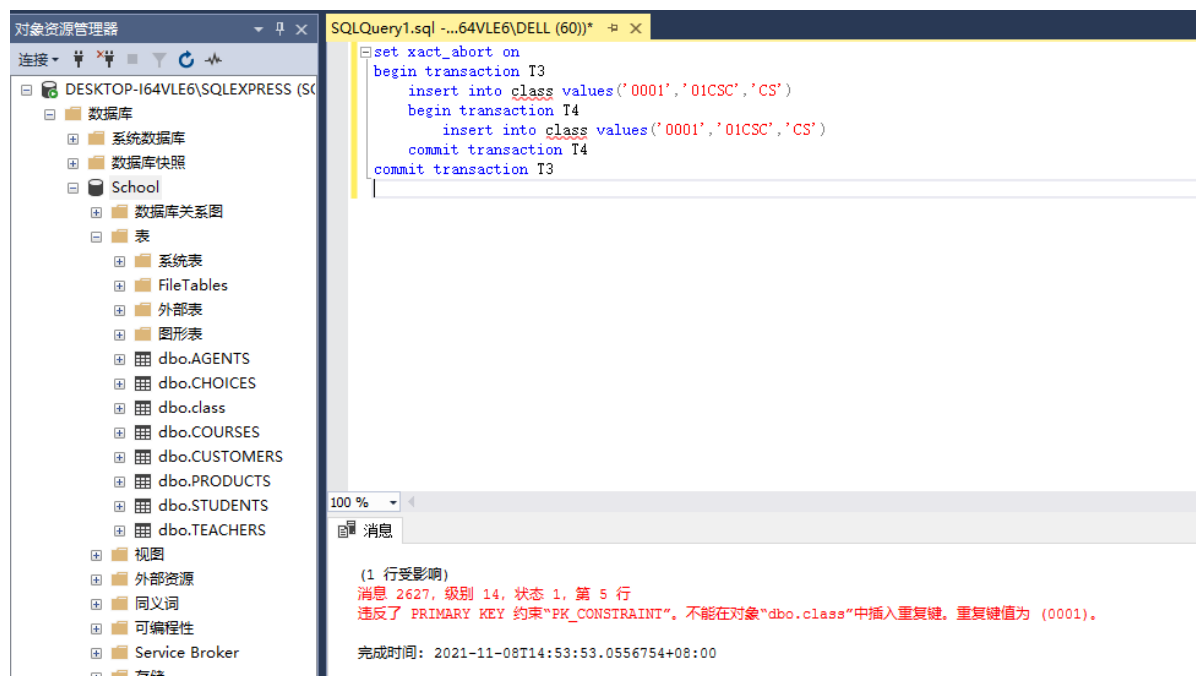
(1)在school数据库中建立一张新表class，包括class\_id(varchar(4)), name(varchar(10)), department(varchar(20))三个列，并约束class\_id为主键。

```
1 create table class(  
2     class_id varchar(4),  
3     name varchar(10),  
4     department varchar(20),  
5     constraint PK_CONSTRAINT PRIMARY KEY(class_id)  
6 );
```

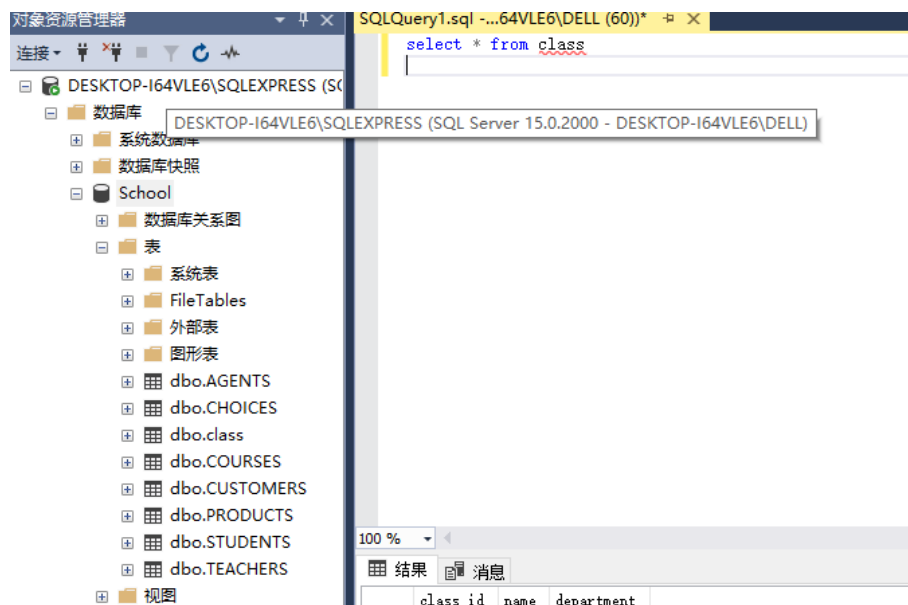


(2)创建事务T3，在事务中插入一个元组 ('0001', '01CSC','CS') ,并在T3中嵌套创建事务T4，T4也插入和T3一样的元组，编写代码测试，查看结果。

```
1 set xact_abort on  
2 begin transaction T3  
3     insert into class values('0001','01CSC','CS')  
4     begin transaction T4  
5         insert into class values('0001','01CSC','CS')  
6     commit transaction T4  
7 commit transaction T3
```



```
1 select * from class
```



第二句话出错了因为主键约束0001已经存在了。嵌套事务T4出错以后T3所做的插入操作也被回滚了，这主要是由于SET XACT\_ABORT ON语句的使用。只有SET XACT\_ABORT off也就是取消了回滚，这样第二个插入语句执行不了，第一个插入成功。

(3)在表class中，尝试设置name='01CSC'的记录的class\_id 为NULL，查看结果

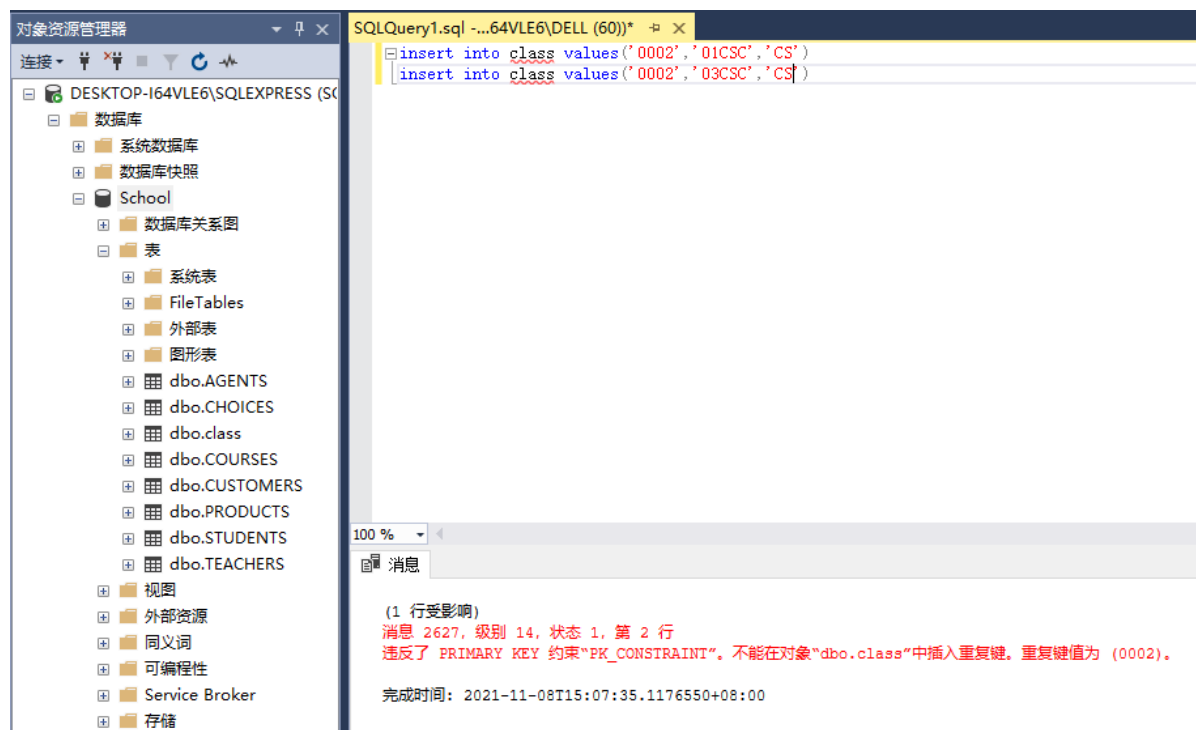
```
1 update class set class_id=NULL where name='01CSC'
```



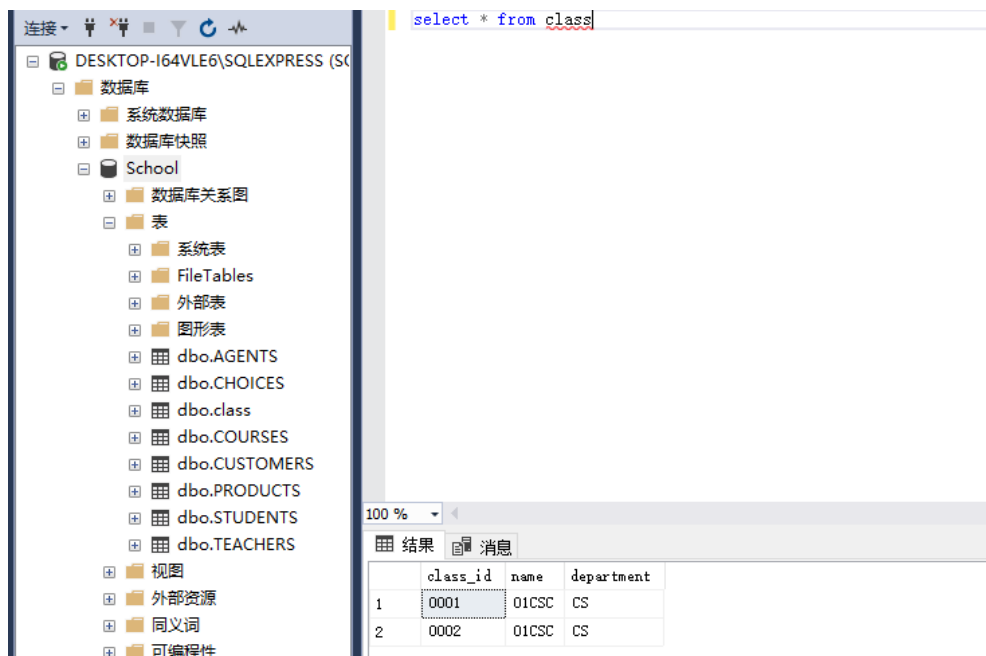
## 主键不能为NULL

(4)在表class中，不创建事务，插入两个元组（'0002'，'01CSC'，'CS'），（'0002'，'03CSC'，'CS'），然后查看表中有几条记录，为什么？

```
1 insert into class values('0002','01CSC','CS')
2 insert into class values('0002','03CSC','CS')
```



```
1 select * from class
```



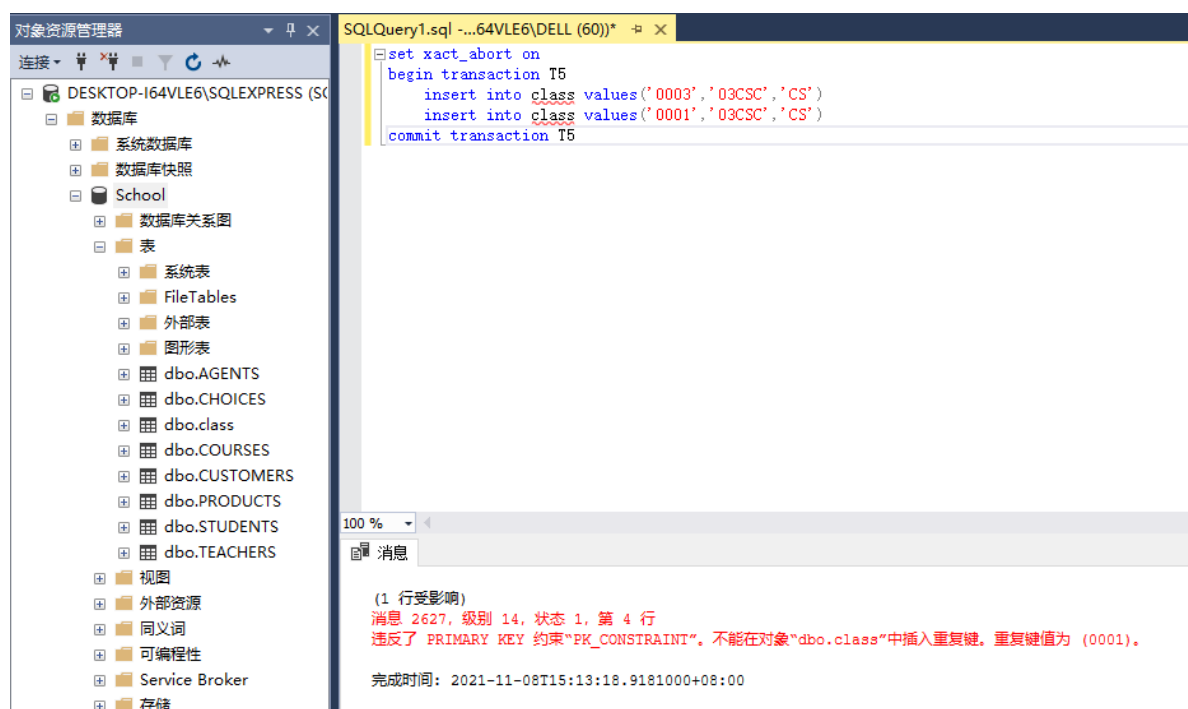
有2条记录，1条是之前插入的。不能重复插入重复的主键。

(5)在表class中，创建事务，并设置开启回滚，然后插入两个元组 ('0003', '03CSC', 'CS')， ('0001', '03CSC', 'CS')，查看结果，表中几条记录？

```

1 set xact_abort on
2 begin transaction T5
3     insert into class values('0003','03CSC','CS')
4     insert into class values('0001','03CSC','CS')
5 commit transaction T5

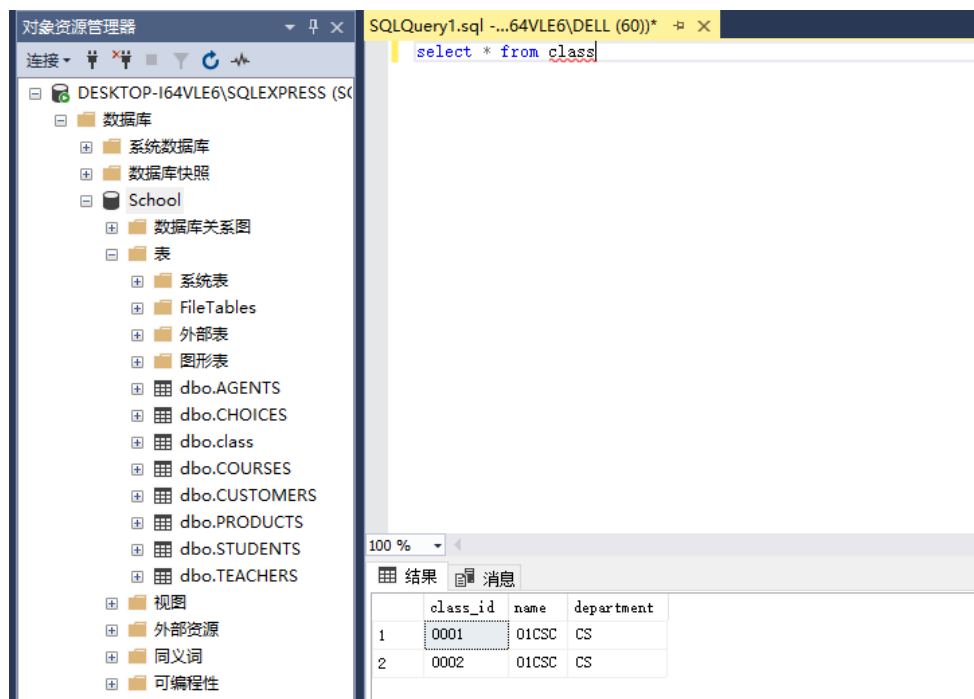
```



```

1 select * from class

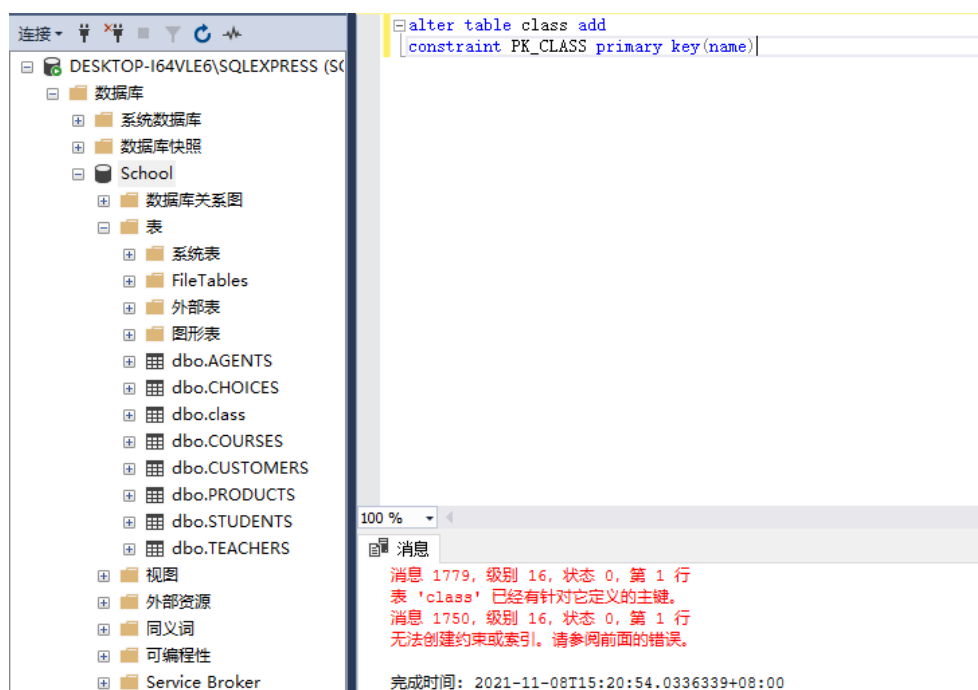
```



有2条记录。由于SET XACT\_ABORT ON语句的使用，当插入重复主键的values时，整个事务终止并回滚，2条都插入失败。

(6)在完成上面几步的前提下，尝试设置'name'为主键，看能否成功，并思考原因。

```
1 alter table class add
2 constraint PK_CLASS primary key(name)
```



不能。因为表class中的2条记录的name属性都是'01CSC'。如果设为主键，违反了主键的唯一性属性，将破坏实体完整性，所以不能成功。