

# 数据库系统原理实验报告

课程名称:数据库系统原理

专业:竺可桢学院工科平台

学号:3080030049

姓名:王可宇

## 实验3 SQL 数据完整性

**实验目的:**

1. 熟悉通过 SQL 进行数据完整性控制的方法。

**实验平台:**

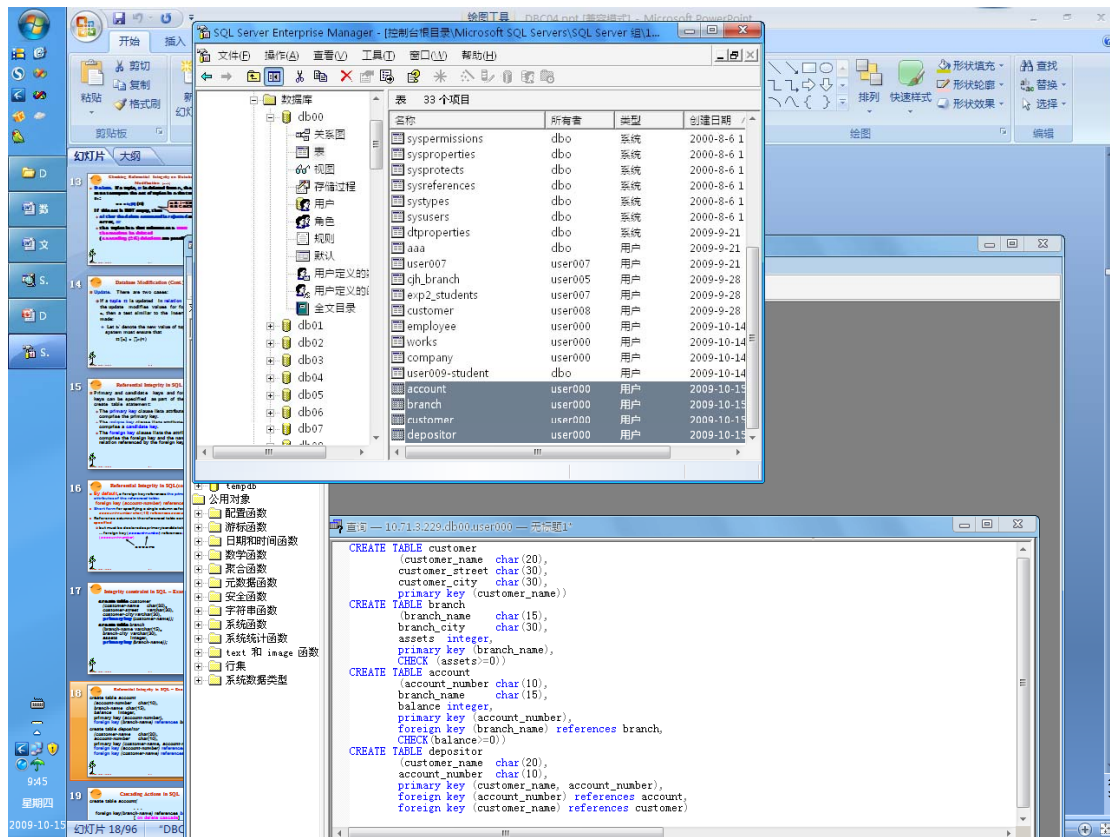
1. 数据库管理系统: SQL Server 2000 或 MySQL

**实验内容和要求:**

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。

**实验过程:**

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



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);

```

让表中插入数据，考察 primary key 如何控制实体完整性：

```
INSERT customer
```

```
VALUES('Henry','Tianmushan','Hangzhou')
```

```
INSERT customer
```

```
VALUES('Rock','Nanjing','Hangzhou')
```

之后再执行

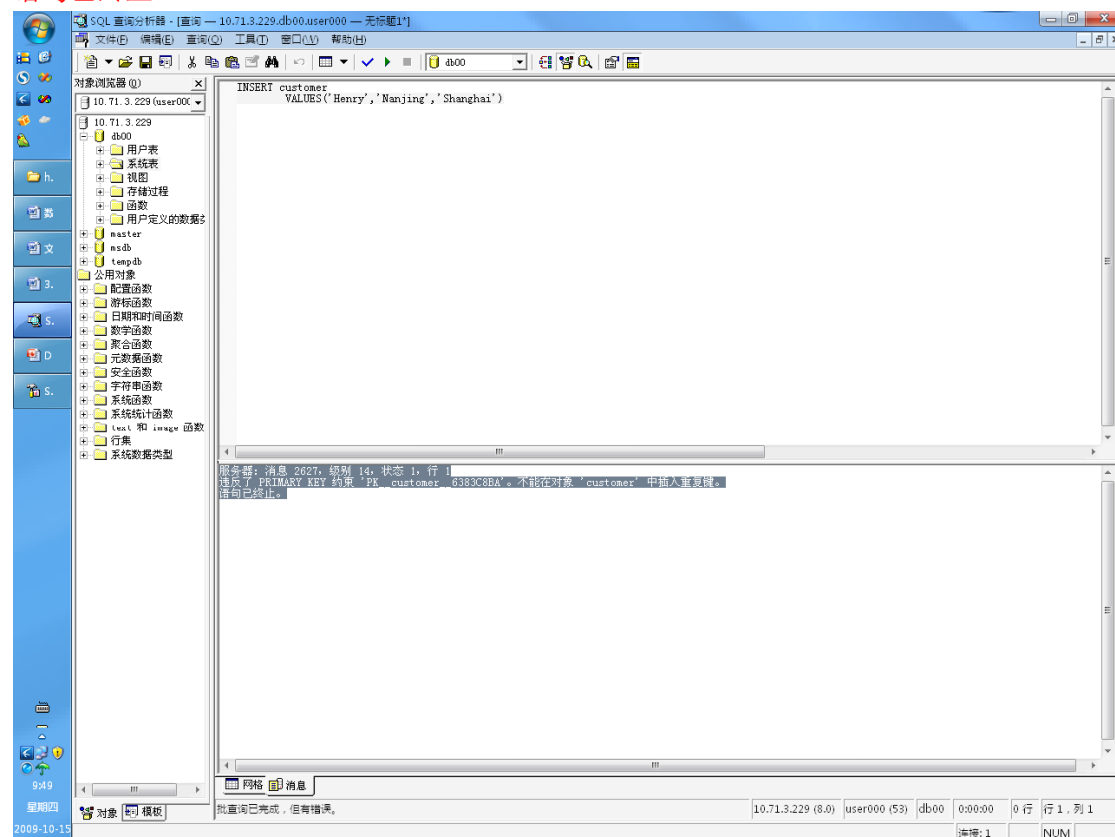
```
INSERT customer
```

```
VALUES('Henry','Nanjing','Shanghai')
```

服务器：消息 2627，级别 14，状态 1，行 1

违反了 PRIMARY KEY 约束 'PK\_customer\_6383C8BA'。不能在对象 'customer' 中插入重复键。

语句已终止。



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：

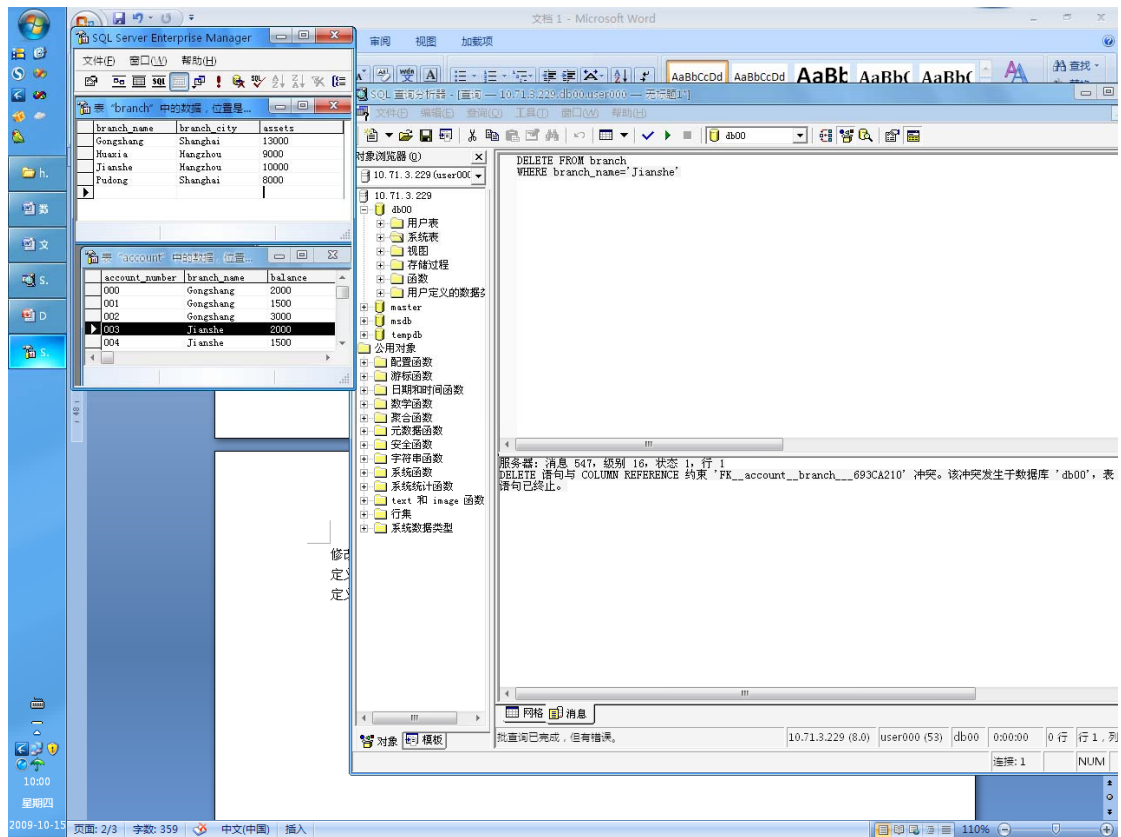
```
DELETE FROM branch
```

```
WHERE branch_name='Jianshe'
```

服务器：消息 547，级别 16，状态 1，行 1

DELETE 语句与 COLUMN REFERENCE 约束 'FK\_account\_branch\_\_693CA210' 冲突。该冲突发生于数据库 'db00'，表 'account'，column 'branch\_name'。

语句已终止。



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:

UPDATE branch

SET branch\_name='Guangda'

WHERE branch\_name='Jiaotong'

服务器: 消息 547, 级别 16, 状态 1, 行 1

UPDATE 语句与 COLUMN REFERENCE 约束 'FK\_\_account\_\_branch\_\_693CA210' 冲突。该冲突发生于数据库 'db00', 表 'account', column 'branch\_name'。

语句已终止。

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)

服务器: 消息 547, 级别 16, 状态 1, 行 1

INSERT 语句与 COLUMN CHECK 约束 'CK\_\_account\_\_balance\_\_6A30C649' 冲突。该冲突发生于数据库 'db00', 表 'account', column 'balance'。

语句已终止。

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:

CREATE ASSERTION assertion\_bal CHECK

(not exists(SELECT \* FROM account

WHERE balance>5000))

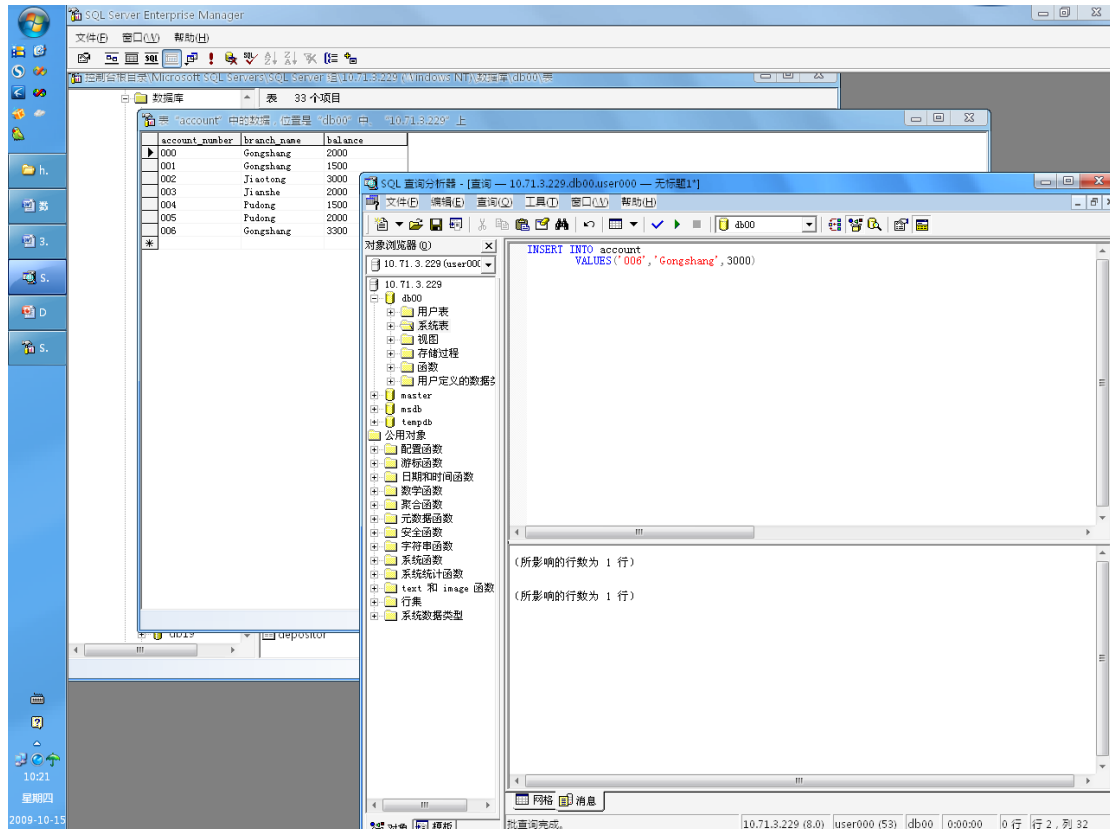
SQL Server 似乎不支持此功能

定义一个 trigger, 并通过修改表中数据考察触发器如何起作用:

```

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

```



```

INSERT INTO account
VALUES('006','Gongshang',3000)

```

(所影响的行数为 1 行)

(所影响的行数为 1 行)