## 实验五 数据库的完整性约束与触发器实验

##### 一、实验目的：

(1) 了解SQL Server 中实体完整性、参照完整性和用户自定义完整性的实现手段。

(2) 理解关系数据模型的三类完整性约束。

(3) 掌握SQL中实现数据完整性的方法。

(4) 掌握触发器的操作。

##### 二、实验内容：

(1) 用SQL Server Management Studio对数据库中数据表的属性列设置主键、创建默认约束，创建唯一性约束。

(2) 用SQL语句对数据库中数据表的属性列设置主键、创建默认约束，创建唯一性约束。

(3) 建立和测试触发器

(4) 管理触发器

##### 三、实验步骤：

**1、创建、删除约束**

（1）在执行创建产品信息表的操作时，指定产品编号为主键值，并且创建一个聚簇索引。

create table products(

id char(10) not null，

name char(20) not null,

price money default 0.01,

quantity smallint null,

constraint pk\_id primary key clustered (id))

（2）创建一个学生信息表，其中输入性别字段值时，只能接受“F”或者“M”，并且为phonenum字段创建检查约束，限制只能输入类似01080798654之类的数据，而不能随意输入其他数据。

create table student(

Id char(8),

name char(8),

sex char(2),

phonenum int,

constraint chk\_sex check（sex in （’F’，’M’））,

Constraint chk\_phonenum check

（phonenum like ‘(010)[0-9][0-9][0-9][0-9] [0-9][0-9][0-9][0-9]’）

)

（3）删除学生信息表中的约束chk\_phonenum。

**2、建立和测试触发器**

(1)建立和测试DELETE触发器

目的：将book表中删除的行备份在book\_bak表中。

①建立测试表

CREATE TABLE book(

bno INT PRIMARY KEY,

bname VARCHAR(20) NOT NULL,

bprice MONEY NOT NULL

)

②建立测试备份表

CREATE TABLE book\_bak(

bno INT PRIMARY KEY,

bname VARCHAR(20) NOT NULL,

bprice MONEY NOT NULL

)

③插入测试数据

INSERT INTO book VALUES(1,'数据库系统概论',10)

INSERT INTO book VALUES(2,'计算机网络',20)

INSERT INTO book VALUES(3,'数据结构',30)

INSERT INTO book VALUES(4,'计算机组成原理',40)

④建立DELETE触发器

CREATE TRIGGER book\_delete

ON book FOR DELETE AS

INSERT INTO book\_bak SELECT \* FROM deleted

⑤测试DELETE触发器

DELETE FROM book WHERE bno=1

DELETE FROM book WHERE bno=2

DELETE FROM book WHERE bno=3

DELETE FROM book WHERE bno=4

SELECT \* FROM book

SELECT \* FROM book\_bak

(2)建立和测试INSERT触发器

目的：将book表中插入的行同时插入book\_bak表中。

①建立INSERT触发器

CREATE TRIGGER book\_insert

ON book FOR INSERT AS

INSERT INTO book\_bak SELECT \* FROM inserted

②测试INSERT触发器

INSERT INTO book VALUES(5,'C程序设计',10)

INSERT INTO book VALUES(6,'C++程序设计',10)

INSERT INTO book VALUES(7,'Java程序设计',10)

INSERT INTO book VALUES(8,'C#程序设计',10)

SELECT \* FROM book

SELECT \* FROM book\_bak

(3)建立和测试UPDATE触发器

目的：检查价格变化，若上下浮动超过20%则不能修改。

①建立UPDATE触发器

CREATE TRIGGER book\_update

ON book FOR UPDATE AS

DECLARE @old\_price MONEY,@new\_price MONEY

SELECT @old\_price=bprice FROM deleted

SELECT @new\_price=bprice FROM inserted

IF(@new\_price>@old\_price\*1.2) OR (@new\_price<@old\_price\*0.8)

BEGIN

PRINT 'ROLLBACK!'

ROLLBACK

END

ELSE

BEGIN

PRINT 'UPDATE!'

END

②测试UPDATE触发器

UPDATE book SET bprice=11 WHERE bno=5 //允许修改

UPDATE book SET bprice=9 WHERE bno=6 //允许修改

UPDATE book SET bprice=20 WHERE bno=7 //不能修改

UPDATE book SET bprice=5 WHERE bno=8 //不能修改

SELECT \* FROM book

**3、管理触发器**

(1)查看触发器

sp\_helptrigger book

sp\_helptext book\_update

(2)删除触发器

DROP TRIGGER book\_insert

(3)禁用触发器

ALTER TABLE book DISABLE TRIGGER book\_delete

(4)启用触发器

ALTER TABLE book ENABLE TRIGGER book\_delete