|  |  |
| --- | --- |
| 成绩评定 |  |
| 教师签名 |  |

**嘉应学院 计算机学院**

**实验报告**

课 程　 数据库系统原理

实验地点 锡科407

指导老师　 陈旭东 实验名称　 完整性和触发器设计

班 级 1602

学 号　 161060009

学生姓名 黄佳俊

**一 实验目的**

通过实验掌握完整性设计方法及触发器的原理和基本设计方法。

二 **实验原理**

数据库的完整性是指数据的正确性和相容性，可以通过触发器来维护数据库的完整性。触发器是一种特殊的存储过程，在用户试图对指定的表执行指定的数据修改语句时自动执行。允许为任何给定的 INSERT、UPDATE 或 DELETE 语句创建多个触发器。

基本语法：create trigger 触发器名 on 表名 for insert/update/delete as T-sql

**三 实验环境**

VMware Workstation7

**四 内容和方法**

（一）完整性设计

方法一、在设计表时定义约束

删除数据库school,建立新数据库school1

1.定义约束

create table student(

Sno char(9), /\*主键约束 提示primary key \*/

Sname char(20), /\* 唯一约束 提示 unique \*/

Ssex char(2) , /\* 检查约束 提示 性别只能是男或女 check \*/

Sage int , /\* 检查约束 提示年龄介于1 到 100之间 check\*/

Sdept char(20) /\* 缺省 缺省值为CS, default（） \*/

);

create table Course(

Cno char(4), /\*主键约束 \*/

Cname char(40), /\* 唯一约束 \*/

Cpno char(4) , /\*非空约束 提示not null \*/

Ccredit int, /\* 缺省值为1 \*/

);

create table SC(

Sno char(9),

Cno char(4),

Grade int , /\* 约束取值为0 到100 \* check /

/\*定义(Sno,Cno)为主键 \*/

/\*定义sno为外键参考student表的主键sno,并且实现级联删除更新SC表中相应的元组 \*/

/\* 提示 foreign key .. references ... on delete cascade on update cascade,

/\*定义Cno为外键参考course表的主键Cno,并且实现级联删除SC表中相应的元组 \*/

);

2. 检验约束

插入下列记录

insert into student values ('95001','李勇','男',20,'CS')

insert into student values ('95002', '刘晨','女',19,'IS')

insert into student values('95003', '王敏', '女',18, 'MA')

insert into student values('95004', '张立', '男',19, 'IS')

insert into Course values(1,'数据库', 5,4)

insert into Course values(2, '数学', 3 ,2)

insert into Course values(3, '信息系统', 1, 4)

insert into Course values('4', '操作系统', '6', 3)

insert into Course values('5', ' 数据结构', '7', 4)

insert into Course values('6', '数据处理', '', 2)

insert into Course values('7', 'PASCAL语言', '6', 4)

insert into SC values('95001', '1', 92)

insert into SC values('95001', '2', 85)

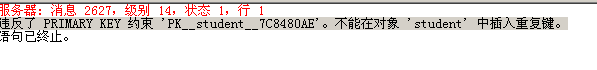
insert into SC values('95001', '3', 88)

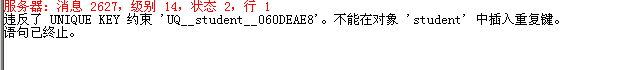
insert into SC values('95002', '2', 90)

insert into SC values('95002', '3', 80)

请设计系列测试用例逐一检查每一个定义的约束是否生效

例如：检查student表的主键约束是否生效,测试用例 insert into student values ('95001','测试','男',20,'CS')

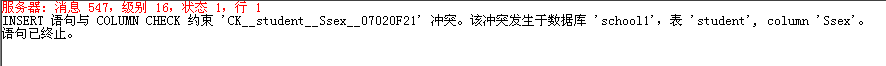




检查student表的唯一约束是否生效,测试用例 insert into student values ('95005','李勇','男',20,'CS')

检查student表的检查约束是否生效,测试用例 insert into student values ('95005','李刚','牛',20,'CS')

等



方法二、在表定义完成后，添加修改约束

建立新数据库school2

1. 定义表

create table student(

Sno char(9),

Sname char(20),

Ssex char(2) ,

Sage int ,

Sdept char(20)

);

create table Course(

Cno char(4),

Cname char(40),

Cpno char(4) ,

Ccredit int

);

create table SC(

Sno char(9),

Cno char(4),

Grade int,

);

2.添加修改约束

（1). 添加主键约束

alter table 表名

add constraint 约束名 primary key (主键)







(2). 添加外键约束

alter table 从表

add constraint 约束名

foreign key(属性名) references 主表(属性名)





(3). 外键约束 + 级联（删除/更新）

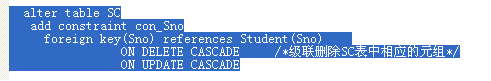
alter table 从表

add constraint 约束名

foreign key(属性名) references 主表(属性名)

ON DELETE CASCADE /\*级联删除SC表中相应的元组\*/

ON UPDATE CASCADE



(4). 添加唯一约束

alter table 表名

add constraint 约束名 unique (属性名)





(5). 添加默认约束

alter table 表名

add constraint 约束名 default ('默认内容') for 字段





(6). 添加检查check约束,

alter table 表名

add constraint 约束名 check (属性名 约束条件 )

根据方法一的约束要求，在数据库school2上为各个表添加完整性约束。





2.检查约束是否生效

参考方法一设计测试用例，逐一检查！

3. 删除约束

alter table 表名 drop constraint 约束名

设计测试用例，检查删除约束后是否生效？

添加多个约束，情况如何？

（二）触发器

删除触发器

drop trigger 触发器名

1. 先了解下列触发器的设计

例1：

USE school

IF EXISTS (SELECT name FROM sysobjects

WHERE name = 'trg3' AND type = 'TR')

DROP TRIGGER trg3

GO

CREATE TRIGGER trg3

ON student

FOR INSERT, UPDATE, DELETE

AS

EXEC master..xp\_sendmail 'teacher',

'HI，有人改动学生表！！'

GO

在STUDENT表中作任何操作有什么变化？请查看MASTER数据库中的扩展存储过程xp\_sendmail

2.触发器设计实验

（1）更新触发器设计

在pubs数据库中建立下列触发器：

在表sales中创建了更新触发器，在sales表中更新title\_id的销售数据(qty值) ，将自动在titles表中更新总计的数据。

deleted -old

inserted - new

use pubs

create trigger upd1

on sales

for update

as

begin

update titles

set ytd\_sales=ytd\_sales-(select sum(qty) from deleted

group by deleted.title\_id having titles.title\_id=deleted.title\_id)

where titles.title\_id in (select deleted.title\_id from deleted)

select \* from deleted

update titles

set ytd\_sales=ytd\_sales+(select sum(qty) from inserted

group by inserted.title\_id having titles.title\_id=inserted.title\_id)

where titles.title\_id in (select inserted.title\_id from inserted)

select \* from inserted

end

注意:deleted和inserted为临时表。

触发器中加入了 select \* from deleted，select \* from inserted

请用SQL语句在表sales中做更新操作,在titles中观察输出结果.

查询更新前的数据



select title\_id, ytd\_sales from titles where title\_id='bu1032'

记录查询结果

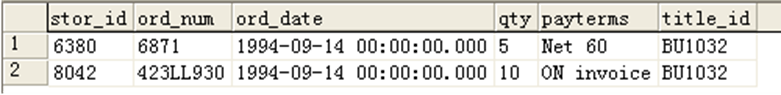
BU1032 4095

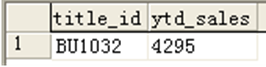
BU1032 4295

测试用例

update sales set qty=qty + 100 where title\_id='bu1032'

再次运行 select title\_id, ytd\_sales from titles where title\_id='bu1032'，记录结果，与前面的查询记录比较





测试用例

update sales set qty=qty - 100 where title\_id='bu1032'

再次运行 select title\_id, ytd\_sales from titles where title\_id='bu1032'，记录结果，与前面的查询记录比较





（2）插入触发器设计

在数据库school1中 创建两个新表

create table teacher1(

eno char(10) primary key,

ename varchar(20),

job char(10),

Sal numeric(7,2)

);

CREATE TABLE Sal\_logg

(eno char(10) references teacher1(eno),

Sal NUMERIC(7,2),

Username char(10),

Date TIMESTAMP

)

在此表创建一个触发器，确保输入的数据中，如果职称为教授，输入的工资数据如果低于4000，但将自动变为4000,同时自动在表Sal\_logg中添加相关记录。

在表Teacher1中，创建插入触发器

CREATE TRIGGER Insert\_trg ON Teacher1

for insert AS

BEGIN

IF ((select job from inserted)='教授')

AND ((select sal from inserted )<4000)

begin

select \* from inserted /\* 测试添加的临时记录 \*/

update teacher1 set Sal=4000

end

END

在表teacher1中，创建更新触发器

CREATE TRIGGER update\_trg ON Teacher1

for update as

declare @eno1 char(10), @sal1 numeric(7,2),@ename1 char(20)

BEGIN

select @eno1=eno from inserted

select @sal1=Sal from inserted

select @ename1= ename from inserted

INSERT INTO Sal\_logg (eno,sal,Username) VALUES(

@eno1,@sal1,@ename1)

END

请设计测试用例，检验触发器是否生效。

insert into teacher1 values('t001','李刚','教授',3500.35)



检查结果

select \* from teacher1

select \* from Sal\_logg





通过本次实验学会和掌握了完整性设计方法及触发器的原理和基本设计方法。