# 实验七 使用触发器实现数据完整性实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号： | 1181103114 | 姓名： | 章磊 |
| 班级： | 计算机1181 | 指导老师： | 冯万利 |
| 实验时间： | 5.20 | 实验地点： | 家 |

# 一、实验目的和要求

要求学生理解触发器实现数据完整性的重要性,掌握用触发器创建与应用的方法;掌握用触发器实现数据参照完整性的方法;理解触发器与约束的不同。要求完成利用SQL Server中创建和管理触发器，并能进行简单的应用

# 二、实验内容

(1)为表建立触发器,激活触发器进行验证。  
(2)为表建立级联更新的触发器,实现参照完整性，并激活触发器进行验证。

(3)比较约束与触发器的执行顺序。

# 三、实验步骤

## 实验6.1

## 创建JXGL数据库表S的插人触发器triI\_NSERT\_S,插入记录的年龄仅在15~30。 (1)打开SQL Server Management Studio窗口。 (2)单击“标准”工具栏上的“新建查询”按钮,打开“查询编辑器”窗口。 (3)在窗口内直接输入以下CREATE TRIGGER语句创建触发器。 (4)为JXGL数据库表S创建一个INSERT触发器，当插入的新行中年龄的值不在15~30时就激活该触发器,撤销该插入操作。

USE JXGL

GO

CREATE TRIGGER tri\_INSERT\_S ON S

FOR INSERT

AS

DECLARE @S\_age tinyint

SELECT @S\_age=S.age

FROM S

IF @S\_age NOT BETWEEN 15 AND 30

ROLLBACK TRANSACTION

GO

1. 单击“SQL编辑器”工具栏上的“分析”按钮,检查输人的T-SQL语句是否有语法错误，如果有语法错误则进行修改,直到没有语法错误为止。  
   (6)确保无语法错误后单击“SQL编辑器”工具栏上的“执行”按钮,完成触发器的创建。

(7)在表中分别插人两行记录以激活该触发器,第一行年龄的值在15~30以内,第二行年龄的值在15~30以外。  
USE JXGL

GO

INSERT INTO S

VALUES('S15','王晓杰','F',21,'MA')

INSERT INTO S

VALUES('S16','邵庆国','M',13,'IS')

GO

## 实验6.2

## 创建JXGL数据库表S的一个DELETE触发器tgr\_s\_delete,当删除S表中的记录时触发该触发器。

USE JXGL

GO

CREATE TRIGGER tgr\_s\_delete

ON S

FOR DELETE

AS

PRINT'备份数据中...'

IF(object\_id('S\_Backup','U')IS NOT NULL)

INSERT INTO S\_Backup SELECT sno,sname FROM deleted

ELSE

SELECT \* INTO S\_Backup FROM deleted

PRINT'备份数据成功！'

GO  
USE JXGL

GO

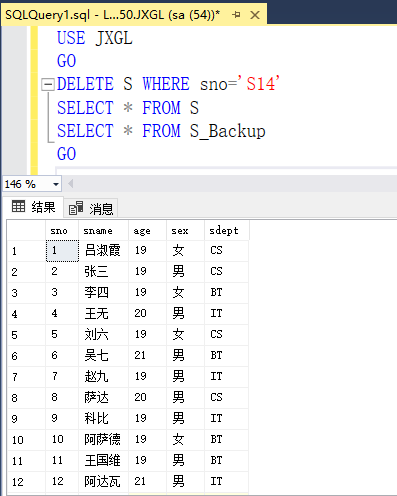
DELETE S WHERE sno='S14'

SELECT \* FROM S

SELECT \* FROM S\_Backup

GO

object\_id()函数是根据对象名称返回该对象的id。

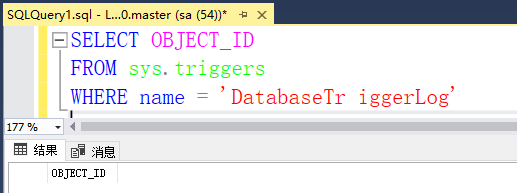


删除表S中的记录

1. 返回DDL触发器的对象标识号SELECT OBJECT\_ID

FROM sys.triggers

WHERE name = 'DatabaseTr iggerLog'



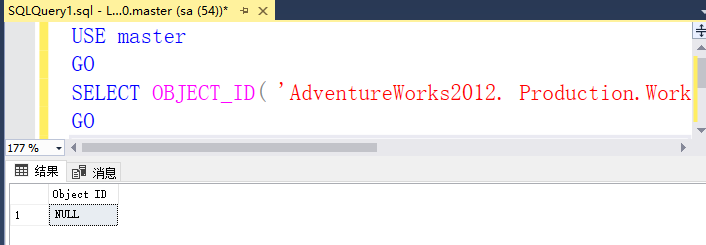
返回DDL触发器的对象标识号

(2)返回AdventureWorks数据库中Production.WorkOrder表的对象ID。如图6.2.3所示  
USE master

GO

SELECT OBJECT\_ID( 'AdventureWorks2012. Production.WorkOrder' )AS '0bject ID';

GO



返回表的对象ID

(3)通过验证表是否具有对象ID来检查指定表的存在性。如果该表存在，则将其删除;如果该表不存在，则不执行DROP TABLE语句。如图6.2.4所示  
USE master

GO

IF OBJECT\_ID('dbo.AWBuildVersion','U') IS NOT NULL

DROP TABLE dbo.AWBuildVersion

GO

(4)使用sys.dm\_db\_index\_operational\_stats函数返回AdventureWorks数据库中Person.Address表的所有索引和分区信息。  
 使用T-SQL函数DB\_ID()OBJECT\_ID()返回参数值时要确保返回有效的ID。如果找不到数据库或对象的名称,例如相应名称不存在或拼写不正确,则两个函数都会返回NULL.sys.dm\_db\_index\_operational\_stats()函数将NULL解释为指定所有数据库或所有对象的通配符值。

USE master

GO

DECLARE @db\_id int

DECLARE @object\_id int

SET @db\_id = DB\_ID('AdventureWorks2008')

SELECT object\_id = OBJECT\_ID('AdventureWorks2008.Person.Address')

IF @db\_id IS NULL

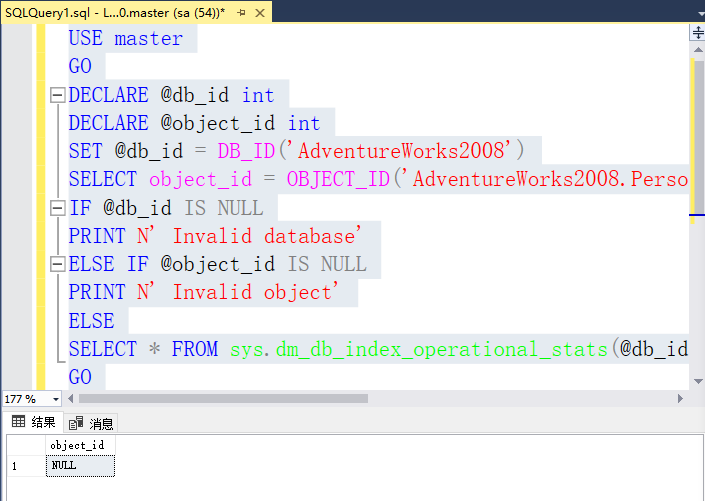
PRINT N' Invalid database'

ELSE IF @object\_id IS NULL

PRINT N' Invalid object'

ELSE

SELECT \* FROM sys.dm\_db\_index\_operational\_stats(@db\_id, @object\_id, NULL, NULL)

GO

## 实验6.3

在数据库JXGL中有3个表S、SC和C,表SC的字段sno作为外键与表S连接。如果要删除表S中的记录需要创建触发器,先删除表SC中与要删除记录级联的全部记录,再删除表S中的记录。

USE JXGL

GO

CREATE TRIGGER delete\_sc\_s ON S

INSTEAD OF DELETE

AS

DECLARE @s\_no char(9)

SELECT @s\_no = sno FROM deleted

DELETE FROM SC

WHERE sno= @s\_no

DELETE FROM S

WHERE sno= @s\_no

执行下述语句时需要启用触发器delete\_sc\_s同时删除表S和SC中级联的记录。  
USE JXGL

GO

DELETE S WHERE sno='S13'

GO

## 实验6.4

在教学管理数据库JXGL中,在删除S表中的元组时将删除的这个元组的前两个属性值插入到bian\_s表中。

CREATE TRIGGER tr\_s\_delete

ON S

FOR DELETE

AS

DECLARE @s\_no char(8),@s\_name char(4)

SELECT @s\_no=sno,@s\_name= sname FROM deleted

INSERT INTO bian\_s VALUES(@s\_no,@s\_name)

GO  
DELETE FROM S WHERE sno= 'S30'

DELETE FROM S WHERE sno= 'S22'

## 实验6.5

为学生表S创建-一个UPDATE触发器,当更新某同学的姓名时激活该触发器,并使用PRINT语句返回一个提示信息。

USE JXGL

GO

CREATE TRIGGER tgr\_s\_update

ON S

FOR UPDATE

AS

DECLARE @oldName char(8), @newName char(8)

SELECT @oldName = sname FROM deleted

IF (exists (SELECT \* FROM S WHERE sname LIKE'%'+ @oldName+'%'))

BEGIN

SELECT @newName = sname FROM inserted

UPDATE S SET sname = replace(sname,@oldName,@newName)WHERE sname LIKE'%'+ @oldName+'%'

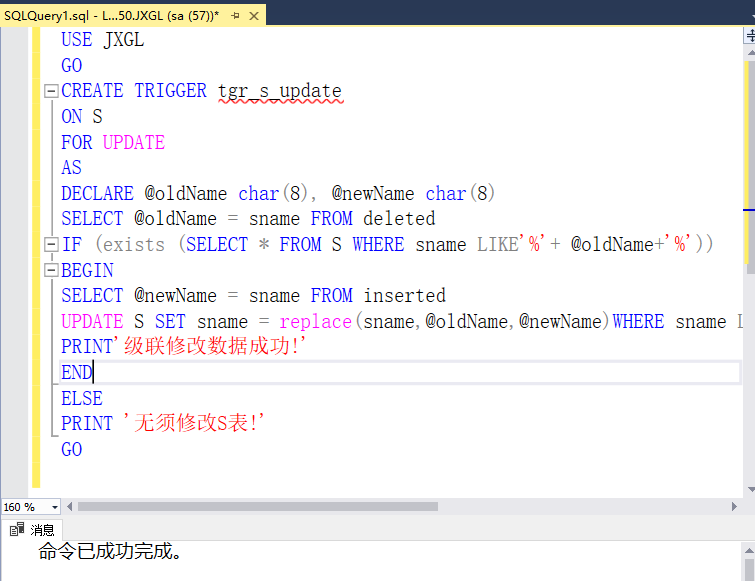
PRINT'级联修改数据成功!'

END

ELSE

PRINT '无须修改S表!'

GO



创建-一个UPDATE触发器

首先查询学生表S的信息,再修改学生的姓名。

USE JXGL

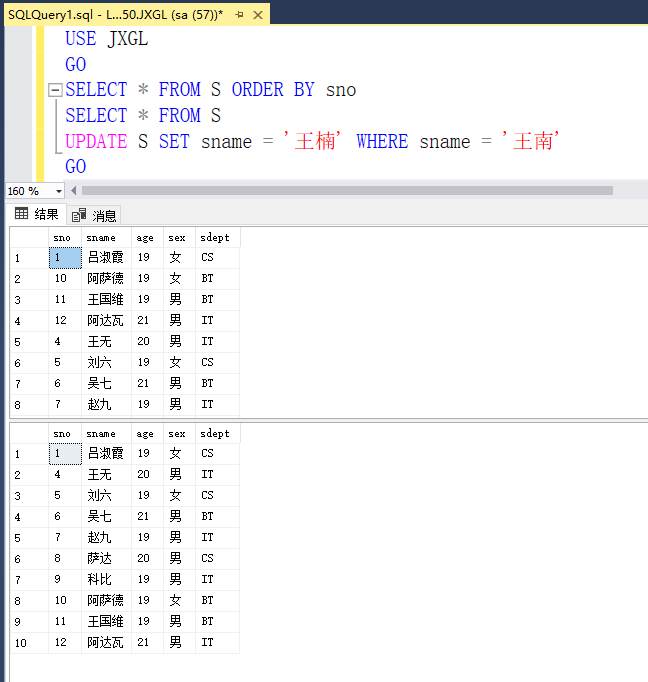
GO

SELECT \* FROM S ORDER BY sno

SELECT \* FROM S

UPDATE S SET sname = '王楠' WHERE sname = '王南'

GO



查询学生表S的信息

## 实验6.6

创建图书销售数据库TSXS,并创建如下两个数据表。

图书销售表

(  
图书编号varchar(20) PRIMARY KEY NOT NULL,

图书名称varchar(40)，  
销售数量int NULL,  
销售单价money NULL,  
销售金额money NULL

）  
其中,销售金额=销售数量\*销售单价。  
图书库存表

(  
图书编号varchar(20) PRIMARY KEY NOT NULL,

库存数量int NULL,  
库存单价money NULL,  
库存金额money NULL

)  
其中,库存金额=库存数量\*库存单价。

(1)创建触发器。说明:每当“图书库存表”发生INSERT动作时触发该触发器。触发器的功能:实现业务规则。  
业务规则:如果销售“图书编号”的书不存在于库存中或者库存为零,则返回错误,否则自动减少“图书库存表”中对应“图书编号”的“库存数量”和“库存金额”。

USE TSXS

GO

IF EXISTS ( SELECT NAME FROM SYSOBJECTS WHERE XTYPE = 'TR'

AND NAME ='t\_sales' )

DROP TRIGGER t\_sales

GO

CREATE TRIGGER t\_sales

ON 图书销售表

FOR INSERT

AS

BEGIN TRANSACTION

IF NOT EXISTS

(

SELECT 库存数量

FROM 图书库存表

WHERE 图书编号 IN (SELECT 图书编号 FROM inserted)

)

BEGIN RAISERROR('错误!该图书不存在库存,不能销售.',16,1)

ROLLBACK

RETURN

END

IF EXISTS (

SELECT 库存数量

FROM 图书库存表

WHERE 图书编号 IN (SELECT 图书编号 FROM inserted) AND 库存数量<= 0)BEGIN

RAISERROR('错误!该图书库存小于等于0,不能销售.',16,1)

RETURN

END

UPDATE 图书销售表

SET 销售金额=销售数量\*销售单价

WHERE 图书编号 IN (SELECT 图书编号 FROM inserted)

DECLARE @图书编号 varchar(20)

SET @图书编号=(SELECT 图书编号 FROM inserted)

DECLARE @销售数量 MONEY

SET @销售数量=(SELECT 销售数量 FROM inserted)

UPDATE 图书库存表

SET 库存数量 = 库存数量 - @销售数量,

库存金额=(库存数量 - @销售数量) \* 库存单价

WHERE 图书编号= @图书编号

COMMIT TRANSACTION

GO

利用不同数据测试“图书库存表”和“图书销售表”的数据变化。

①针对“图书销售表"插人第一条测试数据  
INSERT INTO 图书销售表(图书编号,图书名称,销售数量,销售单价,销售金额)

SELECT 'JSJ\_ QH\_ S120023','数据库原理及应用',10,12,1200

GO

②针对“图书销售表”插入第二条测试数据,该数据“销售金额”不等于“销售单价\*销售数量"。此时触发器将自动更正数据,使“销售金额”等于“销售单价\*销售数量”。  
  
INSERT INTO 图书销售表( 图书编号,图书名称,销售数量,销售单价,销售金额 )

SELECT 'JSJ\_QH\_S120023','数据库原理及应用',10,22,2200

GO

③针对“图书销售表”插入第三条测试数据,该数据中的“图书编号”在“图书库存表"中找不到对应,此时触发器将报错。  
INSERT INTO 图书销售表(图书编号,图书名称,销售数量,销售单价,销售金额)

SELECT 'JSJ\_ GJ\_ C203934', '操作系统',10,60,600

GO

④针对“图书销售表”插人第三条测试数据,该数据中的“图书编号”在“图书库存表”中库存为0,此时触发器将报错。  
INSERT INTO 图书销售表(图书编号,图书名称,销售数量,销售单价,销售金额)

SELECT 'JSJ\_ QH\_ W403011','物联网实用技术',10,30,300

GO

## 实验6.7

触发器中的其他操作。

①触发器中常用消息丽数raisError()的应用。  
USE JXGL

GO

CREATE TRIGGER tgr\_message

ON S

AFTER INSERT,UPDATE

AS raisError('tgr\_message 触发器被触发',16,10)

GO

验证消息函数的提示消息。

USE JXGL

GO

UPDATE S SET sex=18 WHERE sname='张晓梅'

SELECT \* FROM S ORDER BY sno

②启用、禁用触发器。

禁用触发器:  
DISABLE TRIGGER tgr\_message ON S

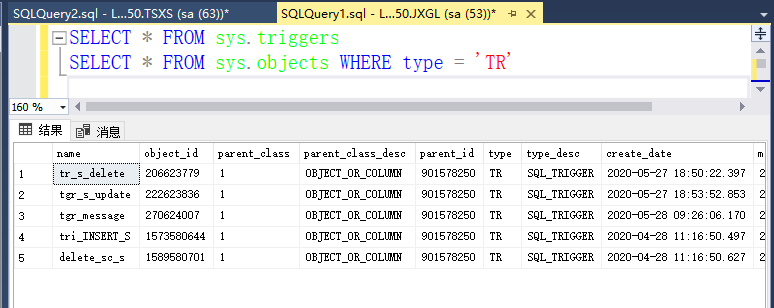
启用触发器:  
ENABLE TRIGGER tgr\_message ON S

③查询创建的触发器信息。

查询已存在的触发器:

SELECT \* FROM sys.triggers

SELECT \* FROM sys.objects WHERE type = 'TR'



查询已存在的触发器

④查看触发器的触发事件。

USE JXGL

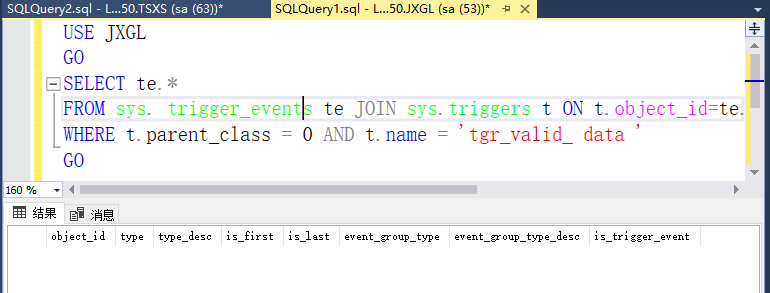
GO

SELECT te.\*

FROM sys. trigger\_events te JOIN sys.triggers t ON t.object\_id=te.object\_id

WHERE t.parent\_class = 0 AND t.name = 'tgr\_valid\_ data '

GO



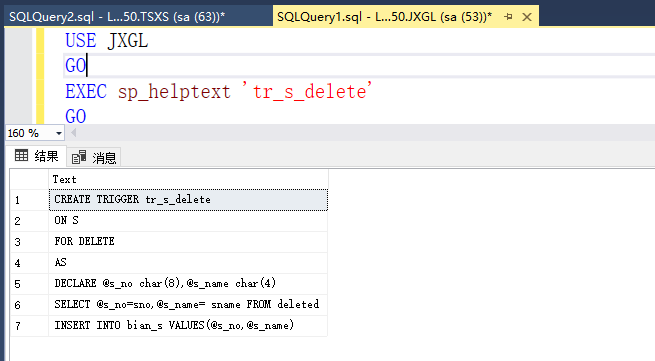
查看触发器的触发事件

⑤查看创建触发器语句USE JXGL

GO

EXEC sp\_helptext 'tr\_s\_delete'

GO



查看创建触发器语句

四、总结与体会

本次实验让我掌握了触发器的一系列操作，以下是我总结的几点要素

触发器有4个要素：名称，触发器有一个符合标志符命名规则的名称。定义的目标：触发器必须定义在表或者视图上。触发条件：是UPDATE、INSERT还是DELETE语句。触发逻辑：触发之后如何处理。　触发器是一种特殊的存储过程。通常用于实现强制业务规则和数据完整性。触发器是通过事件触发而由系统自动执行。主要体现在它在插入、删除或修改指定表中的数据时自动触发执行，以保持数据完整性、检查数据有效性、实现数据库管理任务和相关功能。在sqlserver中，在服务器端实现数据完整性主要有两种手段：一种是在创建表时定义数据完整性，主要分为：实体完整性、域完整性、和级联参照完整性；实现的手段是创建主键约束、唯一键约束、检查约束、默认值约束和各种级联完整性约束。另一种是通过编写触发器语句来实现，通过定义触发条件和编写触发后执行语句，来实现对数据表操作的各种约束。

在实验过程遇到了许多的难题，也使我感到理论知识的重要性。但是我并没有气垒，在实验中发现问题，自己看书，独立思考，最终解决问题，从而也就加深我对课本理论知识的理解，达到了“双赢”的效果。