**新建项目**

**计算机学院（软件学院）实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **学生姓名** | 徐可可 | **学号** | 171530425 | **实验成绩** |  |
| **专业** | 软件工程 | **班级** | 软工四班 | **实验日期** | 19年11月19日 |
| **课程名称** | 数据库原理 | | | **任课教师** | 马丽 |
| **实验名称** | 触发器 | | | **实验序号** | 5 |
| **实验地点** | S411 | **实验台号** | 25 | **指导教师** | 马丽 |
| **一、实验目的及要求**  1. 使用T-SQL语句熟悉触发器的创建和查看方法；  2. 熟悉触发器的修改和删除方法。 | | | | | |
| **二、实验内容（或实验原理、实验拓扑）**  1. 利用触发器实现更为复杂的数据完整性约束和业务逻辑。  2. 写出建立触发器的代码。  3. 写出测试用例。 | | | | | |
| **三、实验设备与环境**  1. 阿里云服务器：Intel(R) Xeon(R) Platinum 8269CY CPU @ 2. 50GHz、2G内存、40G高效云盘、Windows Server 2019 数据中心版、Microsoft SQL Server 2008 R2  2. 高配笔记本：Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz  、16G内存、48TB分布式硬盘、Windows 10 1909政府版、SQL Server Management Studio 18.4  3. 实验室电脑：英特尔 酷睿2 双核 T7700 @ 2.40GHz、3G内存、128 G固态硬盘、Windows 7 旗舰版 32位 SP1、Microsoft SQL Server 2008 R2 | | | | | |
| **四、实验设计方案（包括实验步骤、设计思想、算法描述或开发流程等）**  对于School数据库，删除SC表上的外键约束。   1. 向SC表插入或修改一个记录时，通过触发器检查记录的SNO值在Student表中是否存在，若不存在，则取消插入或修改操作。 插入或修改数据，查看效果。 2. 修改Student表“SNO”字段值时，该字段在SC表中的对应值也做相应修改。 3. 删除Student表中记录的同时删除该记录“SNO”字段值在SC表中对应的记录。 4. 创建INSTEAD OF触发器，当向SC表中插入记录时，先检查CNO列上的值在Course中是否存在，如果存在则执行插入操作，如果不存在则提示“课程编号不存在”。 5. 创建DDL（数据定义语言）触发器，当删除School数据库的一个表时，提示“不能删除表”，并回滚删除表的操作。 6. 若没有删除SC表的外键约束，如果在SC表中插入的“学号”和“课程号”，分别在Student表和Course表中存在，则允许SC表中插入一条数据，如果插入的学号或课程号，不存在，则显示：课程号、学号不在范畴之列，添加不成功！ | | | | | |

|  |
| --- |
| **五、实验结果（包括设计效果、测试数据、运行结果等）**  对于School数据库，删除SC表上的外键约束。   1. 向SC表插入或修改一个记录时，通过触发器检查记录的SNO值在Student表中是否存在，若不存在，则取消插入或修改操作。 插入或修改数据，查看效果。 在这里插入图片描述 在这里插入图片描述 2. 修改Student表“SNO”字段值时，该字段在SC表中的对应值也做相应修改。 在这里插入图片描述 3. 删除Student表中记录的同时删除该记录“SNO”字段值在SC表中对应的记录。 在这里插入图片描述 4. 创建INSTEAD OF触发器，当向SC表中插入记录时，先检查CNO列上的值在Course中是否存在，如果存在则执行插入操作，如果不存在则提示“课程编号不存在”。 在这里插入图片描述 5. 创建DDL（数据定义语言）触发器，当删除School数据库的一个表时，提示“不能删除表”，并回滚删除表的操作。 在这里插入图片描述 6. 若没有删除SC表的外键约束，如果在SC表中插入的“学号”和“课程号”，分别在Student表和Course表中存在，则允许SC表中插入一条数据，如果插入的学号或课程号，不存在，则显示：课程号、学号不在范畴之列，添加不成功！ 在这里插入图片描述 |
| **六、实验小结（包括收获、心得体会、注意事项、存在问题及解决办法、建议等）**  操作过程中，注意：设置完整性后，在维护数据时，要注意数据类型、键和数据约束的限制。  Microsoft SQL Server 2000 提供了两种主要机制来强制业务规则和数据完整性：约束和触发器。触发器是一种特殊类型的存储过程，它在指定的表中的数据发生变化时自动生效。唤醒调用触发器以响应 INSERT、UPDATE 或 DELETE 语句。触发器可以查询其它表，并可以包含复杂的Transact-SQL语句。将触发器和触发它的语句作为可在触发器内回滚的单个事务对待。如果检测到严重错误（例如，磁盘空间不足），则整个事务即自动回滚。  约束主要被用于强制数据的完整性，约束也能提供比触发器更好的性能。然而在所能完成的操作，以及完成工作时所能使用约束是有限制的。触发器则常被用于验证业务规则，或是更复杂的数据验证，然而可以对数据的其他地方的数据完成更深入的更新，约束只能对其所在表中的数据，或是在设计时输入的特定数据进行验证。这同触发器形成对比，触发器可以跨越数据库甚至服务器，可以对任何在设计时设置的数据，或从任何表上的其他行为所收集的数据进行检查。如果所需的访问权限被给予所有包含的对象，就可以使用触发器的这些功能。 简单地来说，触发器可以实现约束的一切功能。但是在考虑数据一致性问题的时候，首先要考虑通过约束来实现。如果约束无法完成的功能，则再通过触发器来解决。两者从功能上来说，他们的关系如下图所示： 在这里插入图片描述 |

|  |
| --- |
| **七、附录（包括作品、流程图、源程序及命令清单等）**   * **sc1.sql** 带约束 * if exists(select top 1 \* from sys.databases where name = 'sc') * begin * use master; * drop database sc; * end * CREATE DATABASE sc * -- 设置工作数据库 * use sc * -- 删除表 * if exists (select \* from sysobjects where id = object\_id(N'Student') and OBJECTPROPERTY(id, N'IsUserTable') = 1) * drop table Student; * -- 创建表 * CREATE TABLE Student( * SNo CHAR(9) PRIMARY Key, * SName CHAR(20) UNIQUE, * SDept CHAR(20), * SB datetime, * SSex CHAR(2), * ); * --插入数据 * --INSERT Student(SNo, SName, SSex, SAge, SDept) * --VALUES('201215121', '李勇', '男', 20, 'CS'); * --INSERT Student VALUES('201215122', '刘晨', '女', 19, 'CS'); * --INSERT Student VALUES('201215123', '王敏', '女', 18, 'MA'); * --INSERT Student VALUES('201215125', '张立', '男', 19, 'IS'); * INSERT INTO student VALUES('s01','王玲','计算机','1986-03-01','男') * INSERT INTO student VALUES('s02','李想','计算机','1985-04-01','女') * INSERT INTO student VALUES('s03','罗军','数学','1986-03-01','男') * INSERT INTO student VALUES('s04','李爱民','英语','1987-06-01','女') * INSERT INTO student VALUES('s05','季然','英语','1986-02-01','女') * INSERT INTO student VALUES('s06','王明','数学','1987-06-01','男') * SELECT \* FROM Student; * -- 设置工作数据库 * use sc * -- 删除表 * if exists (select \* from sysobjects where id = object\_id(N'Course') and OBJECTPROPERTY(id, N'IsUserTable') = 1) * drop table Course; * -- 创建表 * CREATE TABLE Course( * CNo CHAR(4) PRIMARY Key, * CName CHAR(40) NOT null, * CPno CHAR(4), * --CCredit SMALLINT, * --FOREIGN KEY(CPno)REFERENCES Course(CNo) * --表级完整性约束条件，CPno是外码，被参照表是Course，被参照列是CNo * ); * --插入数据 * --INSERT Course(CNo, CName, CPno, CCredit) * --VALUES('1', '数据库', 5, 4); * --INSERT Course VALUES('2', '数学', null, 2); * --INSERT Course VALUES('3', '信息系统', 1, 4); * --INSERT Course VALUES('4', '操作系统', 6, 3); * --INSERT Course VALUES('5', '数据结构', 7, 4); * --INSERT Course VALUES('6', '数据处理', null, 2); * --INSERT Course VALUES('7', 'PASCAL语言', 6, 4); * INSERT INTO course VALUES('c01','高等数学',null) * INSERT INTO course VALUES('c02','数据结构',null) * INSERT INTO course VALUES('c03','操作系统','c02') * INSERT INTO course VALUES('c04','数据库','c03') * INSERT INTO course VALUES('c05','作战指挥','c04') * INSERT INTO course VALUES('c06','离散数学','c01') * INSERT INTO course VALUES('c07','信息安全','c06') * INSERT INTO course VALUES('c08','大学英语',null) * INSERT INTO course VALUES('c09','商贸英语','c08') * INSERT INTO course VALUES('c10','大学物理',null) * INSERT INTO course VALUES('c11','网络',null) * INSERT INTO course VALUES('c12','C程序',null) * SELECT \* FROM Course; * -- 设置工作数据库 * use sc * -- 删除表 * if exists (select \* from sysobjects where id = object\_id(N'SC') and OBJECTPROPERTY(id, N'IsUserTable') = 1) * drop table SC; * -- 创建表 * CREATE TABLE SC( * SNo CHAR(9), * CNo CHAR(4), * Grade SMALLINT, * PRIMARY KEY(SNo, CNo), * FOREIGN KEY(Sno)REFERENCES Student(Sno), * FOREIGN KEY(Cno)REFERENCES Course(CNo) * ); * --插入数据 * --INSERT SC(SNo, CNo, Grade) * --VALUES('201215121', '1', 92); * --INSERT SC VALUES('201215121', '2', 85); * --INSERT SC VALUES('201215121', '3', 88); * --INSERT SC VALUES('201215122', '2', 90); * --INSERT SC VALUES('201215122', '3', 80); * INSERT INTO sc VALUES('s01','c01',80.0) * INSERT INTO sc VALUES('s01','c02',98.0) * INSERT INTO sc VALUES('s01','c03',85.0) * INSERT INTO sc VALUES('s01','c04',80.0) * INSERT INTO sc VALUES('s02','c07',89.0) * INSERT INTO sc VALUES('s02','c05',88.0) * INSERT INTO sc VALUES('s02','c06',78.0) * INSERT INTO sc VALUES('s03','c04',89.0) * INSERT INTO sc VALUES('s03','c01',88.0) * INSERT INTO sc VALUES('s03','c03',78.0) * INSERT INTO sc VALUES('s04','c07',77.0) * INSERT INTO sc VALUES('s04','c02',null) * INSERT INTO sc VALUES('s04','c09',83.0) * INSERT INTO sc VALUES('s05','c10',75.0) * INSERT INTO sc VALUES('s05','c11',90.0) * INSERT INTO sc VALUES('s05','c03',94.0) * INSERT INTO sc VALUES('s06','c09',89.0) * INSERT INTO sc VALUES('s06','c10',88.0) * INSERT INTO sc VALUES('s06','c11',null) * SELECT \* FROM SC; * **sc2.sql** 不带约束 * if exists(select top 1 \* from sys.databases where name = 'sc') * begin * use master; * drop database sc; * end * CREATE DATABASE sc * -- 设置工作数据库 * use sc * -- 删除表 * if exists (select \* from sysobjects where id = object\_id(N'Student') and OBJECTPROPERTY(id, N'IsUserTable') = 1) * drop table Student; * -- 创建表 * CREATE TABLE Student( * SNo CHAR(9) PRIMARY Key, * SName CHAR(20) UNIQUE, * SDept CHAR(20), * SB datetime, * SSex CHAR(2), * ); * --插入数据 * --INSERT Student(SNo, SName, SSex, SAge, SDept) * --VALUES('201215121', '李勇', '男', 20, 'CS'); * --INSERT Student VALUES('201215122', '刘晨', '女', 19, 'CS'); * --INSERT Student VALUES('201215123', '王敏', '女', 18, 'MA'); * --INSERT Student VALUES('201215125', '张立', '男', 19, 'IS'); * INSERT INTO student VALUES('s01','王玲','计算机','1986-03-01','男') * INSERT INTO student VALUES('s02','李想','计算机','1985-04-01','女') * INSERT INTO student VALUES('s03','罗军','数学','1986-03-01','男') * INSERT INTO student VALUES('s04','李爱民','英语','1987-06-01','女') * INSERT INTO student VALUES('s05','季然','英语','1986-02-01','女') * INSERT INTO student VALUES('s06','王明','数学','1987-06-01','男') * SELECT \* FROM Student; * -- 设置工作数据库 * use sc * -- 删除表 * if exists (select \* from sysobjects where id = object\_id(N'Course') and OBJECTPROPERTY(id, N'IsUserTable') = 1) * drop table Course; * -- 创建表 * CREATE TABLE Course( * CNo CHAR(4) PRIMARY Key, * CName CHAR(40) NOT null, * CPno CHAR(4), * --CCredit SMALLINT, * --FOREIGN KEY(CPno)REFERENCES Course(CNo) * --表级完整性约束条件，CPno是外码，被参照表是Course，被参照列是CNo * ); * --插入数据 * --INSERT Course(CNo, CName, CPno, CCredit) * --VALUES('1', '数据库', 5, 4); * --INSERT Course VALUES('2', '数学', null, 2); * --INSERT Course VALUES('3', '信息系统', 1, 4); * --INSERT Course VALUES('4', '操作系统', 6, 3); * --INSERT Course VALUES('5', '数据结构', 7, 4); * --INSERT Course VALUES('6', '数据处理', null, 2); * --INSERT Course VALUES('7', 'PASCAL语言', 6, 4); * INSERT INTO course VALUES('c01','高等数学',null) * INSERT INTO course VALUES('c02','数据结构',null) * INSERT INTO course VALUES('c03','操作系统','c02') * INSERT INTO course VALUES('c04','数据库','c03') * INSERT INTO course VALUES('c05','作战指挥','c04') * INSERT INTO course VALUES('c06','离散数学','c01') * INSERT INTO course VALUES('c07','信息安全','c06') * INSERT INTO course VALUES('c08','大学英语',null) * INSERT INTO course VALUES('c09','商贸英语','c08') * INSERT INTO course VALUES('c10','大学物理',null) * INSERT INTO course VALUES('c11','网络',null) * INSERT INTO course VALUES('c12','C程序',null) * SELECT \* FROM Course; * -- 设置工作数据库 * use sc * -- 删除表 * if exists (select \* from sysobjects where id = object\_id(N'SC') and OBJECTPROPERTY(id, N'IsUserTable') = 1) * drop table SC; * -- 创建表 * CREATE TABLE SC( * SNo CHAR(9), * CNo CHAR(4), * Grade SMALLINT, * PRIMARY KEY(SNo, CNo), * ); * --插入数据 * --INSERT SC(SNo, CNo, Grade) * --VALUES('201215121', '1', 92); * --INSERT SC VALUES('201215121', '2', 85); * --INSERT SC VALUES('201215121', '3', 88); * --INSERT SC VALUES('201215122', '2', 90); * --INSERT SC VALUES('201215122', '3', 80); * INSERT INTO sc VALUES('s01','c01',80.0) * INSERT INTO sc VALUES('s01','c02',98.0) * INSERT INTO sc VALUES('s01','c03',85.0) * INSERT INTO sc VALUES('s01','c04',80.0) * INSERT INTO sc VALUES('s02','c07',89.0) * INSERT INTO sc VALUES('s02','c05',88.0) * INSERT INTO sc VALUES('s02','c06',78.0) * INSERT INTO sc VALUES('s03','c04',89.0) * INSERT INTO sc VALUES('s03','c01',88.0) * INSERT INTO sc VALUES('s03','c03',78.0) * INSERT INTO sc VALUES('s04','c07',77.0) * INSERT INTO sc VALUES('s04','c02',null) * INSERT INTO sc VALUES('s04','c09',83.0) * INSERT INTO sc VALUES('s05','c10',75.0) * INSERT INTO sc VALUES('s05','c11',90.0) * INSERT INTO sc VALUES('s05','c03',94.0) * INSERT INTO sc VALUES('s06','c09',89.0) * INSERT INTO sc VALUES('s06','c10',88.0) * INSERT INTO sc VALUES('s06','c11',null) * SELECT \* FROM SC; * **CreateTrigger.sql** * --对于School数据库，删除SC表上的外键约束。 * use sc; * --1. 向SC表插入或修改一个记录时，通过触发器检查记录的SNO值在Student表中是否存在，若不存在，则取消插入或修改操作。 * IF (EXISTS(SELECT \* FROM sysobjects WHERE id=object\_id(N'SCSno') AND OBJECTPROPERTY(id, N'IsTrigger') = 1)) * DROP TRIGGER SCSno; * go * CREATE TRIGGER SCSno ON SC FOR INSERT,UPDATE AS * BEGIN * IF((SELECT sno FROM inserted) NOT IN(SELECT sno FROM Student)) * begin * print 'Coco: 插入学号不在学生表中，插入失败'; * ROLLBACK TRANSACTION; * print 'Coco: 回滚完毕'; * end * else * print 'Coco: 插入成功'; * END * --插入或修改数据，查看效果。 * delete from sc where sno='s19' * insert into sc values('s19','c04',92) * delete from sc where sno='s01' * insert into sc values('s01','c04',93) * --2. 修改Student表“SNO”字段值时，该字段在SC表中的对应值也做相应修改。 * IF (EXISTS(SELECT \* FROM sysobjects WHERE id=object\_id(N'SCSno') AND OBJECTPROPERTY(id, N'IsTrigger') = 1)) * DROP TRIGGER SCSno; * -- 注意：需要先删除1.中建立的触发器，否则会报错。 * IF (EXISTS(SELECT \* FROM sysobjects WHERE id=object\_id(N'StudentUpdate') AND OBJECTPROPERTY(id, N'IsTrigger') = 1)) * DROP TRIGGER StudentUpdate; * go * CREATE TRIGGER StudentUpdate ON Student FOR UPDATE AS * BEGIN * UPDATE SC SET sno=(SELECT sno FROM inserted) WHERE sno=(SELECT sno FROM deleted) * --修改(UPDATE)记录 * --inserted表存放更新后的记录，deleted表存放更新前的记录 * END * update Student set sno='s19' where sno='s02'; * --3. 删除Student表中记录的同时删除该记录“SNO”字段值在SC表中对应的记录。 * IF (EXISTS(SELECT \* FROM sysobjects WHERE id=object\_id(N'StudentDelete') AND OBJECTPROPERTY(id, N'IsTrigger') = 1)) * DROP TRIGGER StudentDelete; * go * CREATE TRIGGER StudentDelete ON Student FOR DELETE AS BEGIN * DELETE FROM SC WHERE SNO=(SELECT SNO FROM deleted) * END * delete from Student where sno='s02' * --4. 创建INSTEAD OF触发器，当向SC表中插入记录时，先检查CNO列上的值在Course中是否存在，如果存在则执行插入操作，如果不存在则提示“课程编号不存在”。 * IF (EXISTS(SELECT \* FROM sysobjects WHERE id=object\_id(N'Course\_EXISTS') AND OBJECTPROPERTY(id, N'IsTrigger') = 1)) * DROP TRIGGER Course\_EXISTS; * go * CREATE TRIGGER Course\_EXISTS ON SC INSTEAD OF INSERT AS BEGIN * DECLARE @CourseID char(6) * SET @CourseID=(SELECT CNO FROM inserted) * IF(@CourseID IN(SELECT CNO FROM Course)) * INSERT INTO SC SELECT \* FROM inserted * ELSE * PRINT '课程编号不存在' * END * --5. 创建DDL（数据定义语言）触发器，当删除School数据库的一个表时，提示“不能删除表”，并回滚删除表的操作。 * IF (EXISTS(SELECT \* FROM sysobjects WHERE id=object\_id(N'table\_delete') AND OBJECTPROPERTY(id, N'IsTrigger') = 1)) * DROP TRIGGER table\_delete; * go * CREATE TRIGGER table\_delete ON DATABASE AFTER DROP\_TABLE AS BEGIN * PRINT '不能删除该表' * ROLLBACK TRANSACTION * END * if exists (select \* from sysobjects where id = object\_id(N'SC') and OBJECTPROPERTY(id, N'IsUserTable') = 1) * drop table SC; * --6. 若没有删除SC表的外键约束，如果在SC表中插入的“学号”和“课程号”，分别在Student表和Course表中存在，则允许SC表中插入一条数据，如果插入的学号或课程号，不存在，则显示：课程号、学号不在范畴之列，添加不成功！ * IF (EXISTS(SELECT \* FROM sysobjects WHERE id=object\_id(N'tr4\_insert\_sc') AND OBJECTPROPERTY(id, N'IsTrigger') = 1)) * DROP TRIGGER tr4\_insert\_sc; * go * create trigger tr4\_insert\_sc on SC instead of insert as begin * if exists(select \* from inserted where sno in(select sno from Student) and cno in (select cno from Course)) * print '数据添加成功！' * else * begin * print '课程号、学号不在范畴之列，添加不成功！' * Rollback transaction * end * end   insert into SC values('s19','c04',96) |