**实验八——事务管理——补充材料1**

**一、**SQL Server数据库事务基础知识

**1．事务的概念( Transaction )**

所谓事务是用户定义的一个数据库操作序列，这些操作要么都做，要么都不做，是一个不可分割的工作单位。

关系数据库中，事务可以是一条SQL语句、一组SQL语句。

在SQL语言中，定义事务的语句有三条：

Begin Transaction 开始

Commit 结束

Rollback 回滚

**2．事务开始：BEGIN TRANSACTION**

标记一个显式本地事务的起始点。BEGIN TRANSACTION将 @@TRANCOUNT 加 1。

**语法结构：**

BEGIN TRAN [ SACTION ] [ transaction\_name | @tran\_name\_variable

[ WITH MARK [ 'description' ] ] ]

**参数说明：**

transaction\_name：是给事务分配的名称。transaction\_name 必须遵循标识符规则，但是不允许标识符多于 32 个字符。仅在嵌套的 BEGIN...COMMIT 或 BEGIN...ROLLBACK 语句的最外语句对上使用事务名。

@tran\_name\_variable：是用户定义的、含有有效事务名称的变量的名称。必须用 char、varchar、nchar 或 nvarchar 数据类型声明该变量。

WITH MARK ['description']：指定在日志中标记事务。Description 是描述该标记的字符串。

如果使用了 WITH MARK，则必须指定事务名。WITH MARK 允许将事务日志还原到命名标记。

**4．事务提交：COMMIT TRANSACTION**

标志一个成功的隐性事务或用户定义事务的结束。如果 @@TRANCOUNT 为 1，COMMIT TRANSACTION 使得自从事务开始以来所执行的所有数据修改成为数据库的永久部分，释放连接占用的资源，并将 @@TRANCOUNT 减少到 0。如果 @@TRANCOUNT 大于 1，则 COMMIT TRANSACTION 使 @@TRANCOUNT 按 1 递减。

**语法结构：**

COMMIT [ TRAN [ SACTION ] [ transaction\_name | @tran\_name\_variable ] ]

**参数说明：**

transaction\_name：Microsoft SQL Serve 忽略该参数。transaction\_name 指定由前面的 BEGIN TRANSACTION 指派的事务名称。transaction\_name 必须遵循标识符的规则，但只使用事务名称的前 32 个字符。通过向程序员指明 COMMIT TRANSACTION 与哪些嵌套的 BEGIN TRANSACTION 相关联，transaction\_name 可作为帮助阅读的一种方法。

@tran\_name\_variable：是用户定义的、含有有效事务名称的变量的名称。必须用 char、varchar、nchar 或 nvarchar 数据类型声明该变量。

**5．事务回滚：ROLLBACK TRANSACTION**

将显式事务或隐性事务回滚到事务的起点或事务内的某个保存点。

**语法结构：**

ROLLBACK [ TRAN [ SACTION ]

[ transaction\_name | @tran\_name\_variable

| savepoint\_name | @savepoint\_variable ] ]

**参数说明：**

transaction\_name：是给 BEGIN TRANSACTION 上的事务指派的名称。transaction\_name 必须符合标识符规则，但只使用事务名称的前 32 个字符。嵌套事务时，transaction\_name 必须是来自最远的 BEGIN TRANSACTION 语句的名称。

@tran\_name\_variable：是用户定义的、含有有效事务名称的变量的名称。必须用 char、varchar、nchar 或 nvarchar 数据类型声明该变量。

savepoint\_name：是来自 SAVE TRANSACTION 语句的 savepoint\_name。savepoint\_name 必须符合标识符规则。当条件回滚只影响事务的一部分时使用 savepoint\_name。

@savepoint\_variable：是用户定义的、含有有效保存点名称的变量的名称。必须用 char、varchar、nchar 或 nvarchar 数据类型声明该变量。

二、SQL Server数据库事务创建

**1．事务的创建**

（1）在查询分析器中执行以下语句，创建一个名为t\_InsUpdate简单的事务，并使它正常提交。

Begin transaction t\_InsUpdate --t\_InsUpdate为事务名

Use CPXS

Insert into CP(产品编号,产品名称,价格,库存量)

Values('100021','宝马汽车',456780,39)

Update XSS

Set 负责人='张飞'

Where 客户编号='000003'

Commit transaction t\_InsUpdate --事务提交结束，t\_InsUpdate为事务名

**测试：**

执行语句“select \* from xss; ”，看数据是否添加到表中？

请给出测试结果：

（2）在查询分析器中执行以下语句，创建一个简单的事务，并使它回滚

BEGIN TRANSACTION

Use CPXS

select \* from xss;

update xss

set 客户名称='厦门理工学院'

where 客户编号='000002';

select \* from xss;

Rollback;

**测试：**

select \* from xss;

**2．事务的存储点：**

事务的存储点可以使事务在发生回滚的情况下，存储点前的操作结果得以保存。执行以下语句，创建一个名为t\_InsertCP，其中包含一个存储点。

Begin transaction t\_InsertCP

--Use CPXS

Insert into CP(产品编号,产品名称,价格,库存量)

Values('100028','天山雪莲',456,57)

Save transaction t\_InsertCP --存储点

Update CP

Set 产品名称='云南白药'

Where 产品编号='208729' --此为一个不存在的编号，目的是使插入操作出错

if @@error!=0 --@@error为系统全局变量，错误号

rollback transaction t\_InsertCP

else

commit transaction t\_InsertCP

**测试：**

使用查询语句查询表CP中数据，观查查询结果，看存储点前的操作结果是否确实得以保存。