

项目10 创建使用存储过程和触发器

任务1

任务2

实训

项目10 创建使用存储过程和触发器

项目10 创建使用存储过程和触发器

任务1

任务2

实训

【能力目标】

- 能理解存储过程和触发器的概念与分类
- 能创建、执行、修改与删除存储过程
- 能定义、修改与删除触发器

【项目描述】

按照需求为XS数据库创建存储过程和触发器。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

【项目分析】

在学生数据库XS的实际应用中，常需要重复执行一些数据操作。比如，要查询某个系的学生情况，也存在如某学生新增加一门课程的及格成绩时，在学生档案XSDA表中自动在总学分中增加该课程的学分等。为了方便用户，也为了提高执行效率，SQL Server 2008中的用户自定义函数、存储过程、触发器可以用来满足这些应用需求。它们是SQL Server程序设计的灵魂，掌握和使用好它们对数据库的开发与应用非常重要。本项目主要介绍存储过程和触发器的使用。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

【任务设置】

任务1.....创建和使用存储过程

任务2.....创建和使用触发器

实训十.....为sale数据库创建存储过程和触发器

项目10 创建使用存储过程和触发器

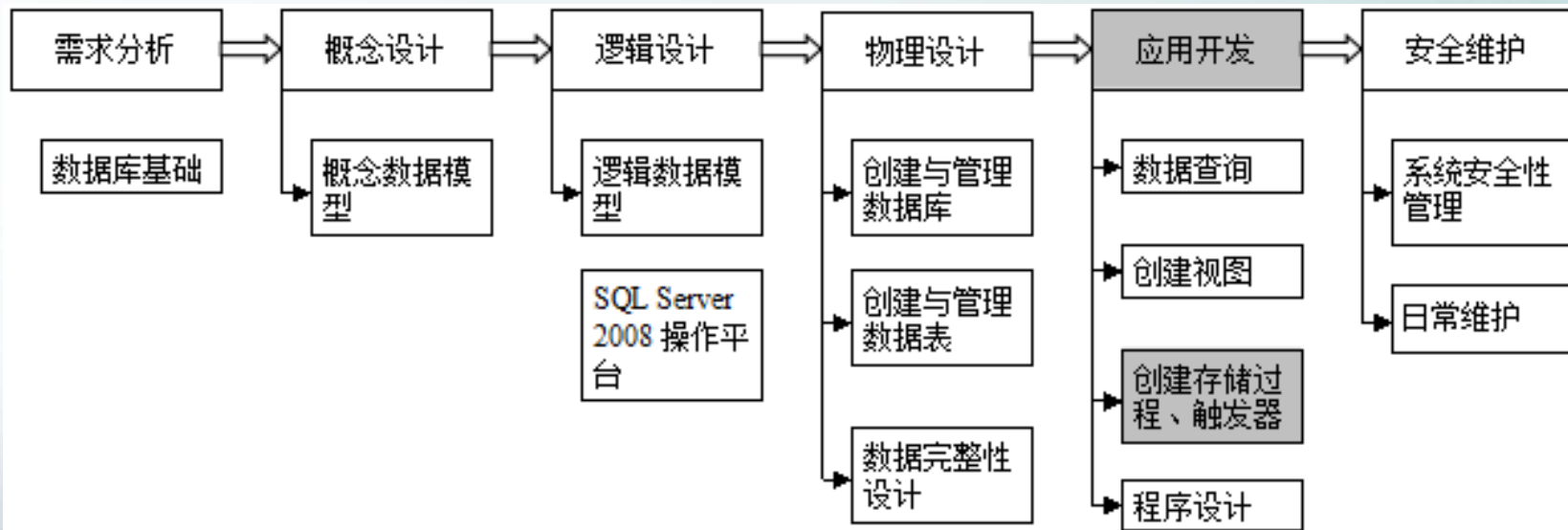
任务1

任务2

实训

【项目定位】

数据库系统开发



项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务1 创建和使用存储过程

【任务目标】

- 理解存储过程的作用
- 学会根据需要创建、修改、删除存储过程（带输入、输出参数）
- 在实际应用开发时能够灵活运用存储过程以提高开发效率

项目10 创建使用存储过程和触发器

任务1

任务2

实训

【任务描述】

按需求在XS数据库为其创建下面的存储过程。

1. 创建存储过程xsda_xhxm_in，从XS数据库的XSDA表中查询信息系男生的学号、姓名。
2. 创建存储过程xsda_query，从XS数据库的三个表中查询某人指定课程的成绩和学分。
3. 创建存储过程xsda_xm，从XSDA表中查询学生姓名的含‘红’字的学生的姓名，性别和系。
4. 创建存储过程xsda_avg，计算指定学生的各科的平均成绩。
5. 创建存储过程p_CourseNum，能根据用户给定的课程名称统计报名人数。
6. 创建存储过程p_CourseName，能根据用户给定的课程名称显示所有学生姓名及所在班级。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

【任务分析】

存储过程是T-SQL语句和流控语句集合，存储过程能被编译和优化。为了理解什么是存储过程，先看下面的例子。为了查询XS数据库中信息系的学生所修总学分信息，可以使用下列查询：

```
USE XS
```

```
SELECT 学号, 姓名, 系名, 总学分 FROM XSDA
```

```
WHERE 系名='信息'
```

```
ORDER BY 总学分 DESC
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

尽管这个查询不大，只有四行文本，但如果网络上有5000个用户执行同一查询，则从客户机通过网络向服务器发送这个查询需要增加大量网络通信流，可能造成拥塞。要避免拥塞，让网络全速运行，应减少从客户机通过网络向服务器发送的代码量，从而减少网络通信流。为此，最好将代码以一定的形式存放在服务器中，而不是在客户机上。若将上面的这段代码存放在服务器上，并取名为EX，那么执行该查询时，只需发送EXEC EX就可以了。

执行上述存储过程，查看执行结果。比较一下存储过程与用户自定义函数有何异同？

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务1-1 介绍存储过程概述

1. 存储过程的概念

存储过程就是在SQL Server数据库中存放的查询，是存储在服务器中的一组预编译过的T-SQL语句，而不是在客户机上的前端代码中存放的查询。

存储过程除减少网络通信流之外，还有如下**优点**：

(1) 存储过程在服务器端运行，执行速度快。存储过程是预编译过的，当第一次调用以后，就驻留在内存中，以后调用时不必再进行编译，因此，它的运行速度比独立运行同样的程序要快。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

(2) 简化数据库管理。例如，如果需要修改现有查询，而查询存放在用户机器上，则要在所有的用户机器上进行修改。而如果在服务器中集中存放查询并作为存储过程，则只需要在服务器上改变一次。

(3) 提供安全机制，增强数据库安全性。通过授予对存储过程的执行权限而不是授予数据库对象的访问权限，可以限制对数据库对象的访问，在保证用户通过存储过程操纵数据库中数据的同时，可以保证用户不能直接访问存储过程中涉及的表及其他数据库对象，从而保证了数据库数据的安全性。另外，由于存储过程的调用过程隐藏了访问数据库的细节，也提高了数据库中的数据安全性。

(4) 减少网络流量。如果直接使用T-SQL语句完成一个模块的功能，那么每次执行程序时都需要通过网络传输全部T-SQL。若将其组织成存储过程，这样用户仅仅发送一个单独的语句就实现了一个复杂的操作，则需要通过网络传输的数据量将大大减少。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

2. 存储过程的分类

在SQL Server中存储过程主要分为两类：系统存储过程、扩展存储过程和用户自定义存储过程。

系统存储过程主要存储在master数据库中并以sp_为前缀，在任何数据库中都可以调用，在调用时不必在存储过程前加上数据库名。

扩展存储过程提供从SQL Server到外部程序的接口，以便进行各种维护活动，并以xp_为前缀。使用方法与系统存储过程相似。如想返回指定目录下的文件列表，则

```
Use master
```

```
go
```

```
EXEC xp_cmdshell 'dir c:\'
```

用户自定义存储过程由用户自己根据需要而创建，是用来完成某项特定任务的存储过程。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务1-2 创建存储过程

简单的存储过程类似于给一组SQL语句起个名字，然后就可以在需要时反复调用；复杂一些的则需要输入和输出参数。

创建存储过程前，应注意下列几个事项：

- (1) 存储过程只能定义在当前数据库中。
- (2) 存储过程的名称必须遵循标识符的命名规则。
- (3) 不要创建任何使用sp_作为前缀的存储过程。

语法格式：

```
CREATE PROC[EDURE] procedure_name  
[@parameter data_type [=default] [OUTPUT]] [, ...]  
AS sql_statement
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

说明：

- (1) `procedure_name`: 存储过程的名称，并且在当前数据库中必须唯一。
- (2) `@parameter`: 存储过程的形参名，必须以@开头，参数名必须符合标识符的规则，`data_type` 用于说明形参的数据类型。
- (3) `default`: 存储过程输入参数的默认值。如果定义了`default`值，则无须指定此参数值，即可执行存储过程。默认值必须是常量或NULL。如果存储过程使用带LIKE关键字的参数，则可包含通配符：`%`、`_`、`[]`和`[^]`。
- (4) `OUTPUT`: 指定输出参数。此选项的值可以返回给调用EXECUTE的语句。
- (5) `sql_statement`: 要包含在存储过程中的任意数量的T-SQL语句。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

1. 创建简单的存储过程

【例10-1】创建一个存储过程stu_inf，从XSDA表中查询管理系总学分大于55分的学生信息。

```
use XS
go
create proc stu_inf
as
select * from XSDA where 系名='管理' and 总学分>55
go
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

2. 使用输入参数

在上面的例子中，该存储过程只能查询管理系，为了提高程序的灵活性，可使用输入参数。下面的两个例子是有输入参数时，存储过程的创建。

【例10-2】使用输入参数，创建存储过程stu_per，根据学生姓名查询该生的个人信息。

```
use XS
go
create proc stu_per @name char(8)
as
select * from XSDA where 姓名=@name
go
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

3. 使用带默认值的输入参数

【例10-3】创建一个存储过程xscj_inf，查询指定学生的学号，姓名，所选课程名称及该课程的成绩。默认为查询王姓学生的学习情况。

```
create proc xscj_inf @name varchar(8)='王%'
```

```
as
```

```
select xsda.学号,姓名,课程名称,成绩 from xsda,xscj,kcxx
```

```
where xsda.学号=xscj.学号 and xscj.课程编号=kcxx.课程编号 and 姓名 like @name
```

```
go
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

4. 使用输出参数

输出参数用于在存储过程中返回值，使用OUTPUT声明输出参数。

【例10-4】创建一个存储过程kc_avg查询所有学生所学指定课程的平均成绩，并将该平均成绩返回。

```
create proc kc_avg @kcname char(20),@kcavg decimal(3,1) output  
as
```

```
select @kcavg=avg(成绩) from xscj join kcxx on xscj.课程编号=kcxx.课程编号  
where 课程名称=@kcname
```

```
go
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务1-3 执行存储过程

存储过程创建成功后，保存在数据库中。在SQL Server中可以使用EXECUTE命令来直接执行存储过程。

语法格式：

```
[EXEC[UTE]] procedure_name [value|@variable OUTPUT] [, ...]
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

说明：

- (1) EXECUTE：执行存储过程的命令关键字，如果此语句是批处理的第一条语句，可以省略此关键字。
- (2) procedure_name：指定存储过程的名称。
- (3) value为输入参数提供实值，@variable为一个已定义的变量，OUTPUT紧跟在变量后，说明该变量用于保存输出参数返回的值。
- (4) 当有多个参数时，彼此用逗号分隔。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

【例10-5】执行【例10-1】创建的存储过程stu_inf。

use XS

exec stu_inf

【例10-6】执行【例10-2】中创建的存储过程stu_per，查询王红同学的个人信息。

exec stu_per '王红'

项目10 创建使用存储过程和触发器

任务1

任务2

实训

【例10-7】执行【例10-3】中创建的存储过程xscj_inf。

xscj_inf--使用输入参数的默认值，查询的是姓王的学生

xscj_inf '李%'--查询李姓的学生

xscj_inf '王红'--查询“王红”

xscj_inf '% '---查询所有学生

项目10 创建使用存储过程和触发器

任务1

任务2

实训

【例10-8】执行【例10-4】中创建的存储过程kc_avg，查询《数据结构》课程的平均分。

```
declare @avg decimal(3,1)
exec kc_avg '数据结构',@avg output
select @avg as '数据结构'
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务1-4 修改存储过程

存储过程的修改是由ALTER语句来完成的。

语法格式：

```
ALTER PROC[EDURE] procedure_name
```

```
[@parameter data_type [=default] [OUTPUT]] [, ...]
```

```
AS sql_statement
```

其中，各参数的含义与CREATE PROCEDURE相同。

【例10-9】修改存储过程stu_inf，将它修改为查询指定系的学生信息。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务1-4 修改存储过程

```
use XS
```

```
go
```

```
alter proc stu_inf @xb_name char(10)='管理'
```

```
as
```

```
select * from xsda where 系名=@xb_name and 总学分>55
```

```
go
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务1-5 删除存储过程

当不再使用一个存储过程时，就需要把它从数据库中删除。

1. 使用SQL Server Management Studio删除存储过程

使用SQL Server Management Studio删除存储过程十分简单。下面以删除XS数据库中的一个存储过程为例了解一下删除存储过程的操作步骤。

(1) 在【对象资源管理器】中展开XS数据库后，然后展开【可编程性】|【存储过程】，右击需要删除的存储过程，在弹出的快捷菜单上选择【删除】，如图10-1所示。

(2) 打开【删除对象】窗口后，单击【确定】按钮即可删除。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

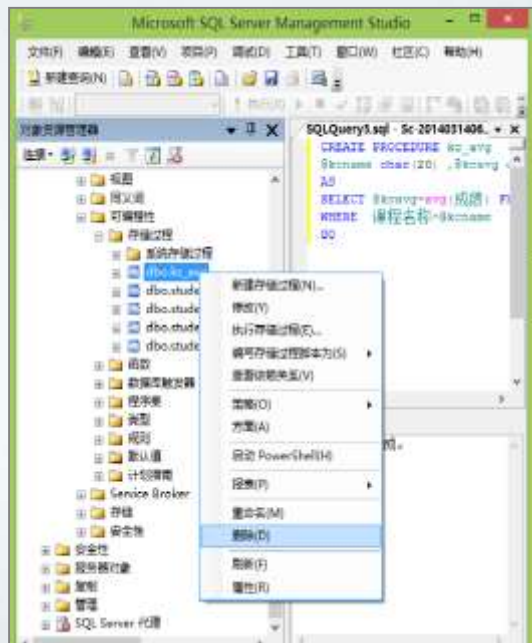


图10-1 选择要删除的存储过程

项目10 创建使用存储过程和触发器

任务1

任务2

实训

2. 使用T-SQL语句删除存储过程

删除存储过程也可以使用T-SQL中的DROP命令，DROP命令可以将一个或多个存储过程从当前数据库中删除。

语法格式：

```
DROP PROC[EDURE] procedure_name [, ...]
```

【例10-10】删除数据库XS中的stu_inf存储过程。

```
drop proc stu_inf
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务1-6 完成综合任务

1. 创建存储过程xsda_xhxm_in，从XS数据库的XSDA表中查询信息系男生的学号、姓名。

```
CREATE PROCEDURE xsda_xhxm_in  
AS  
SELECT 姓名, 性别  
FROM XSDA  
WHERE 系名='信息' and 性别='男'  
GO
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

2. 创建存储过程xsda_query, 从XS数据库的三个表中查询某人指定课程的成绩和学分。

```
CREATE PROCEDURE xsda_query @课程 char (20)
AS
SELECT 姓名, 成绩, 学分
FROM XSDA, KCXX, XSCJ
WHERE 课程名称=@课程 AND XSDA. 学号 =XSCJ. 学号
      AND XSCJ. 课程编号 =KCXX. 课程编号
GO
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

3. 创建存储过程xsda xm, 从XSDA表中查询学生姓名的含‘红’字的学生的姓名, 性别和系。

```
CREATE PROCEDURE xsda_xm
AS
SELECT 姓名, 性别, 系名
FROM XSDA
WHERE 姓名 like '%红%'
GO
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

4. 创建存储过程xsda_avg, 计算指定学生的各科的平均成绩。

```
CREATE PROCEDURE xsda_avg @学生 char (8)
```

```
AS
```

```
SELECT XSCJ. 课程编号, avg(成绩) 平均分
```

```
FROM XSDA, KCXX, XSCJ
```

```
WHERE 姓名=@学生 AND XSDA. 学号 =XSCJ. 学号 AND XSCJ. 课程编号 =KCXX. 课程编号
```

```
GROUP BY XSCJ. 课程编号
```

```
GO
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

5. 创建存储过程p_CourseNum, 能根据用户给定的课程名称统计学习这门课程的人数。

```
CREATE PROCEDURE p_CourseNum
@课程 char (20), @kccou int output
AS
SELECT @kccou=COUNT(xsda. 姓名)
FROM XSDA, KCXX, XSCJ
WHERE 课程名称=@课程 AND XSDA. 学号 =XSCJ. 学号 AND XSCJ. 课程编号 =KCXX. 课程编号
GROUP BY XSCJ. 课程编号
GO
DECLARE @kccou int
exec p_CourseNum '数据结构', @kccou OUTPUT
SELECT @kccou AS '数据结构'
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

6. 创建存储过程p_CourseName，能根据用户给定的课程名称显示所有学习这门课程的学生姓名及所在系部。

```
CREATE PROCEDURE p_CourseName
@课程 char(20)
AS
SELECT XSDA.姓名, 系名
FROM XSDA, KCXX, XSCJ
WHERE 课程名称=@课程 AND XSDA.学号 =XSCJ.学号
      AND XSCJ.课程编号 =KCXX.课程编号
GO
exec p_CourseName '数据结构'
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

【任务分析】

触发器是一类特殊的存储过程，它是在执行某些特定的T-SQL语句时可以自动执行的一种存储过程。根据任务要求定义触发器的条件，查看触发器的效用。比较存储过程与触发器有何异同。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务2 创建和使用触发器

【任务目标】

- 理解触发器的作用
- 能熟练创建、修改、删除触发器
- 在实际应用开发时能够灵活运用触发器完成业务规则以达到简化系统整体设计的目的

项目10 创建使用存储过程和触发器

任务1

任务2

实训

【任务描述】

按需求在XS数据库创建以下触发器：

1. 定义一个触发器xsda_update，一旦对XSDA表进行任何更新操作，这个触发器都将显示一个语句” Stop update xsda, now!”，并取消所作修改。
2. 在XS数据库上定义一个触发器xscj_insert，当在XSCJ表中的插入一条记录时，检查该记录的学号在XSDA表中是否存在，若该记录的学号在XSDA表中不存在，则不允许插入操作，并提示“错误代码1，违背数据的一致性，不允许插入！”。
3. 在数据库XS中创建一触发器kcxx_delete，当从KCXX表中删除一条记录时，检查该记录的课程编码在XSCJ表中是否存在，若该记录的课程编码在XSCJ表中存在，则不允许删除操作，并提示“错误代码2，违背数据的一致性，不允许删除！”。
4. 修改触发器xsda_update，使其起到只是不允许更新学号字段的作用。
5. 阐述何谓触发器？简述其作用与分类。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务2-1 触发器概述

1. 触发器的概念

触发器是一类特殊的存储过程，它作为一个对象存储在数据库中。触发器为数据库管理人员和程序开发人员提供了一种保证数据完整性的方法。触发器是定义在特定的表或视图上的。当有操作影响到触发器保护的数据时，例如，数据表发生了INSERT、UPDATE或DELETE操作时，如果该表有对应的触发器，这个触发器就会自动激活执行。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

2. 触发器的功能

SQL Server 2008提供了两种方法来**保证数据的有效性和完整性**：约束和触发器。触发器是针对数据库和数据表的特殊存储过程，它在指定的表中的数据发生改变时自动生效，并可以包含复杂的T-SQL语句，用于处理各种复杂的操作。SQL Server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，则整个事务即自动回滚，恢复到原来的状态。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

3. 触发器的类型

在SQL Server 2008中，根据激活触发器执行的T-SQL语句类型，可以把触发器分为两类：一类是DML触发器，另一类是DDL触发器。

(1) DML触发器

DML触发器是当数据库服务器中发生数据操作语言DML事件时执行的特殊存储过程，如INSERT、UPDATE或DELETE等。

DML触发器根据其引发的时机不同又可以分为AFTER触发器和INSTEAD OF触发器两种类型。

① AFTER触发器。在执行了INSERT、UPDATE或DELETE语句操作之后执行AFTER触发器。它主要用于记录变更后的处理或检查，一旦发现错误，也可以使用ROLLBACK TRANSACTION语句来回滚本次的操作。

② INSTEAD OF触发器。这类触发器一般是用来取代原本要进行的操作，是在记录变更之前发生的，它并不去执行原来的SQL语句里的操作，而是去执行触发器本身所定义的操作。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

(2) DDL触发器

DDL触发器是当数据库服务器中发生**数据定义语言**DDL事件时执行的特殊存储过程，如CREATE、ALTER等。DDL触发器一般用于执行数据库中的管理任务，如审核和规范数据库操作，防止数据库表结构被修改等。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

4. inserted表和deleted表

每个触发器有两个特殊的表：inserted表和deleted表。这两个表建在数据库服务器的内存中，是由系统管理的逻辑表，而不是真正存储在数据库中的物理表。对于这两个表，用户只有读取的权限，没有修改的权限。

这两个表的结构与触发器所在数据表的结构是完全一致的，当触发器的工作完成之后，这两个表也将从内存中删除。

- inserted表里存放的是更新前的记录：对于插入记录操作来说，inserted表里存储的是要插入的数据；对于更新记录操作来说，inserted表里存放的是要更新的记录。
- deleted表里存放的是更新后的记录：对于更新记录操作来说，deleted表里存放的是更新前的记录；对于删除记录操作来说，deleted表里存放的是被删除的旧记录。

由此可见，在进行INSERT操作时，只影响inserted表；进行删除操作时，只影响deleted表；进行UPDATE操作时，既影响inserted表也影响到deleted表。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

4. inserted表和deleted表

修改操作	inserted表	deleted表
增加(INSERT)记录	存放新增的记录	-----
删除(DELETE)记录	-----	存放被删除的记录
修改(UPDATE)记录	存放更新后的记录	存放更新前的记录

inserted表和deleted表存放的信息

项目10 创建使用存储过程和触发器

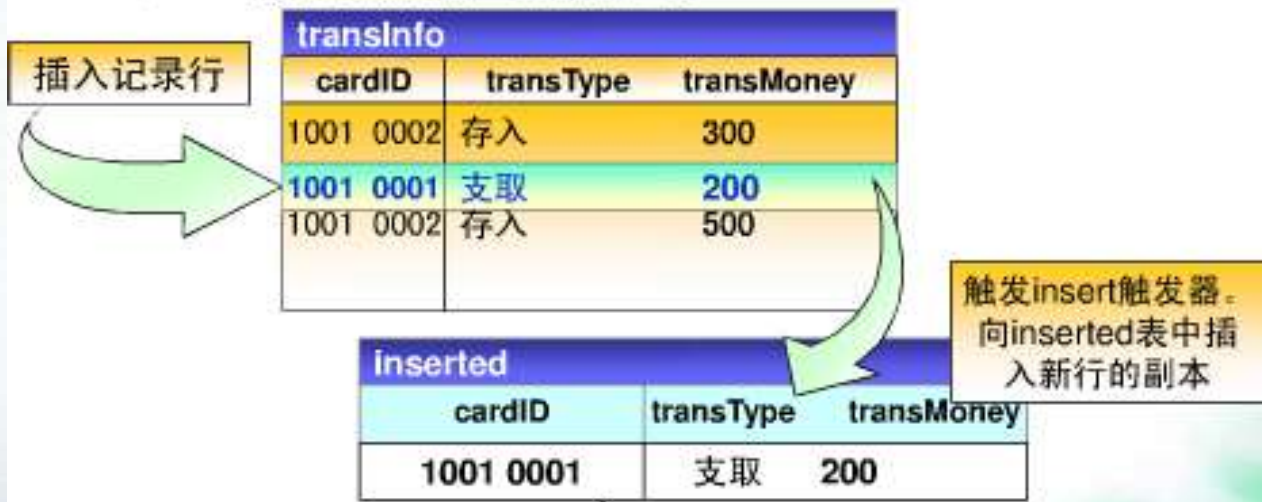
任务1

任务2

实训

4. inserted表和deleted表

□ INSERT触发器的工作原理:



项目10 创建使用存储过程和触发器

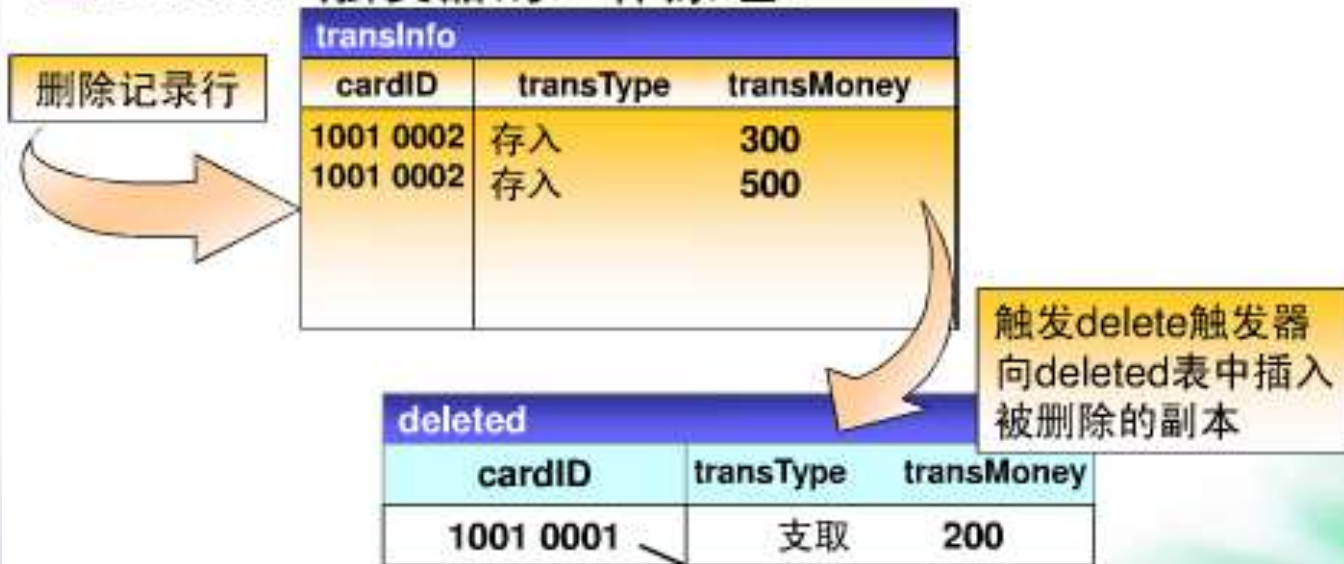
任务1

任务2

实训

4. inserted表和deleted表

□ DELETE触发器的工作原理:



项目10 创建使用存储过程和触发器

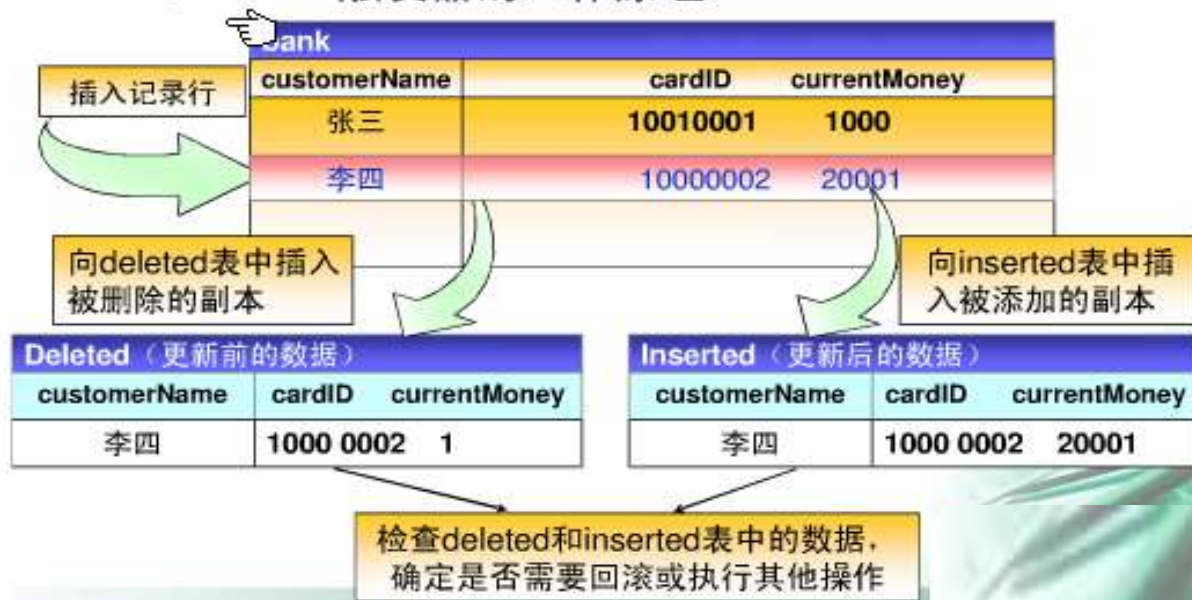
任务1

任务2

实训

4. inserted表和deleted表

□ UPDATE触发器的工作原理:



项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务2-2 创建触发器

1. 使用T-SQL语句创建触发器

触发器与表（视图）是紧密相关的。在创建触发器时，需要指定触发器的名称、包含触发器的表、引发触发器的条件以及当触发器启动后要执行的语句等内容。

语法格式：

```
CREATE TRIGGER trigger_name
ON { table | view }
{[FOR|AFTER]|INSTEAD OF}
{[INSERT ][, ][UPDATE] [, ][DELETE]}
[WITH ENCRYPTION]
AS
[IF UPDATE (column_name) [{AND|OR} UPDATE (column_name) ][...n]]
SQL_statement
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

说明：

- (1) trigger_name: 触发器的名称。每个 trigger_name 必须遵循标识符规则，但 trigger_name 不能以 # 或 ## 开头，不与其它数据库对象同名。
- (2) table|view: 触发器表（或触发器视图）的名称，指出触发器何处触发。
- (3) FOR|AFTER| INSTEAD OF: FOR与AFTER均指定为AFTER触发器，INSTEAD OF指定为INSTEAD OF触发器。该项指出了何时触发。
- (4) INSERT 指定了为INSERT触发器，UPDATE指定为UPDATE触发器，DELETE指定为DELETE触发器。该项指出了何种操作触发。可以为多种操作（如INSERT和UPDATE）定义相同的触发器操作。
- (5) WITH ENCRYPTION: 对CREATE TRIGGER语句的文本进行加密。使用 WITH ENCRYPTION 可以防止将触发器作为 SQL Server 复制的一部分进行发布。
- (6) AS 引导触发器的主体，SQL_statement为一条或多条SQL语句，指出触发器完成的功能。当触发器为INSERT或UPDATE时，还可以通过IF UPDATE（列名1）[{AND|OR} UPDATE（列名2）进一步哪些列数据修改时触发。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

注意，创建触发器有下列**限制**：

- (1) CREATE TRIGGER必须是批处理中的第一条语句，并且只能应用于一个表（视图）。
- (2) 触发器只能在当前的数据库中创建，但触发器可以引用当前数据库的外部对象。
- (3) 不能在视图上定义 AFTER触发器。在表或视图上，每个INSERT、UPDATE 或DELETE语句最多可以定义一个INSTEAD OF 触发器。
- (4) 创建DML触发器的权限默认分配给表的所有者，且不能将该权限转给其他用户。
- (5) 在触发器内可以指定任意的SET语句。选择的SET选项在触发器执行期间保持有效，然后恢复为原来的设置。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

【例10-11】创建INSERT触发器：在数据库XS中创建一触发器，当向XSCJ表插入一记录时，检查该记录的学号在XSDA表是否存在，检查课程编号在KCXX表中是否存在，若有一项为否，则不允许插入。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

```
USE XS
GO
CREATE TRIGGER check_trig
ON XSCJ
FOR INSERT
AS
IF EXISTS(SELECT * FROM inserted
WHERE 学号 NOT IN (SELECT 学号 FROM XSDA) OR 课程编号 NOT IN (SELECT
课程编号 FROM KCXX))
BEGIN
    RAISERROR ('违背数据的一致性.', 16, 1)
    ROLLBACK TRANSACTION
END
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

说明：

(1) Inserted表：存放要插入的数据

(2) 测试触发器：

```
INSERT XSCJ
```

```
VALUES('110110','110',99)
```

(3) 思考：ROLLBACK TRANSACTION加与不加有何区别？

(4) RAISERROR的作用及用法

项目10 创建使用存储过程和触发器

任务1

任务2

实训

【例10-12】创建UPDATE触发器：在数据库XS中创建一触发器，当在XSDA中修改学号字段时，XSCJ中对应学号随之修改。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

```
USE XS
GO
CREATE TRIGGER xsdaxh_trig
ON XSDA
FOR UPDATE
AS
IF UPDATE(学号)
UPDATE XSCJ
SET XSCJ.学号=(SELECT 学号 FROM INSERTED)
WHERE XSCJ.学号=(SELECT 学号 FROM DELETED)
GO
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

【例10-13】创建DELETE触发器：当从XS数据库的XSDA表中删除一个学生的记录时，相应的从XSCJ表中删除该学生对应的所有记录。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

USE XS

GO

CREATE TRIGGER delete_trig

ON XSDA

AFTER DELETE

AS

DELETE FROM XSCJ

WHERE 学号=(SELECT 学号 FROM DELETED)

GO

项目10 创建使用存储过程和触发器

任务1

任务2

实训

【例10-14】创建INSTEAD OF触发器：在XS数据库中创建表、视图和触发器，以说明INSTEAD OF INSERT 触发器的使用。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

USE XS

GO

--创建表goods

CREATE TABLE goods

(goodsKey int IDENTITY(1,1),
goodsName nvarchar(10) NOT NULL,
goodsColor nvarchar(10) NOT NULL,
goodsDescription AS (goodsName+goodsColor),
goodsPrice float)

GO

--创建以goods表为基表的视图goods_view, 该视图包含基表的所有列

CREATE VIEW goods_view

AS

SELECT goodsKey, goodsName, goodsColor, goodsDescription, goodsPrice
FROM goods

GO

项目10 创建使用存储过程和触发器

任务1

任务2

实训

```
--在视图goods_view上创建INSTEAD OF INSERT  
CREATE TRIGGER  instead_insert  
ON goods_view  
INSTEAD OF INSERT  
AS  
BEGIN  
    INSERT INTO goods(goodsName, goodsColor, goodsPrice)  
    SELECT goodsName, goodsColor, goodsPrice FROM inserted  
END  
GO  
--向goods表中插入数据的正确语句如下:  
INSERT INTO goods(goodsName, goodsColor, goodsPrice)  
Values('Apple', '红色', 1.25)  
GO
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

2. 使用SQL Server Management Studio创建触发器

用户也可以使用图形界面的方式建立触发器，如图10-3所示，但还需要采用SQL脚本编辑的方式，请读者自学完成。



图10-3 菜单方式建立触发器

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务2-3 修改触发器

可以使用T-SQL语句修改触发器的定义，但同其它数据库对象的修改类似，这里只讨论修改触发器的定义。

修改触发器语法格式与创建时类似，**只需将关键字CREATE改为ALTER。**

语法格式：

```
ALTER TRIGGER trigger_name
ON { table | view }
{[FOR|AFTER]|INSTEAD OF}
{[INSERT ][, ][UPDATE] [, ][DELETE]}
[WITH ENCRYPTION]
AS
[IF UPDATE (column_name) [{AND|OR} UPDATE (column_name) ][...n]]
SQL_statement
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

说明：

`trigger_name`: 为要修改的已存在于数据库中的一个触发器的名称。其它参数含义请参考创建触发器命令部分。

【例10-15】修改XS数据库的触发器`xздаupd_trig`，将它的功能改为当修改或删除XSDA中的一条记录时，如果在XSCJ表存在有相同学号的记录，则禁止修改或删除。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

```
USE XS
GO
ALTER TRIGGER xsdaupd_trig
ON xsda
FOR UPDATE, DELETE
AS
IF EXISTS (SELECT * FROM XSCJ WHERE 学号=(SELECT 学号 FROM DELETED) )
BEGIN
    PRINT 'Cannot do!'
    PRINT 'Transaction has been cancelled'
    ROLLBACK
END
GO
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

--验证

```
DELETE XSDA WHERE 学号='200801'
```

显示:

Cannot do!

Transaction has been cancelled

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务2-4 删除触发器

同其它数据库对象一样，删除触发器也可以使用多种方式。

1. 使用T-SQL语句删除触发器

语法格式：

```
DROP TRIGGER trigger_name[, ...]
```

说明：trigger_name为需要删除的触发器名称，当一次删除多个触发器时，各个触发器名称之间用逗号隔开。

【例10-16】为了保证触发器能成功创建，可以在CREATE TRIGGER语句前先加一条判断语句，判断数据库中是否有这样一个触发器，如果有，就删除掉。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务2-4 删除触发器

```
IF EXISTS (SELECT * FROM SYSOBJECTS WHERE NAME='xsdaupd_trig' AND  
TYPE='TR' )  
DROP TRIGGER XSDAUPD_TRIG  
go
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

2. 使用SQL Server Management Studio删除触发器

进入如图10-3所示界面后，在【名称】对应的下拉列表中选择要删除的触发器名，然后单击【删除】即可。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务2-5 完成综合任务

1. 定义一个触发器xsda_update，一旦对XSDA表进行任何更新操作，这个触发器都将显示一个语句” Stop update xsda, now!”，并取消所作修改。

```
CREATE TRIGGER xsda_update
ON XSDA
INSTEAD OF UPDATE
AS
    PRINT 'Stop update xsda, now!'
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务2-5 完成综合任务

2. 在XS数据库上定义一个触发器xscj_insert，当在XSCJ表中的插入一条记录时，检查该记录的学号在XSDA表中是否存在，若该记录的学号在XSDA表中不存在，则不允许插入操作，并提示“错误代码1，违背数据的一致性，不允许插入！”

项目10 创建使用存储过程和触发器

任务1

任务2

实训

任务2-5 完成综合任务

```
USE XS
GO
CREATE TRIGGER xscj_insert
ON XSCJ
FOR INSERT
AS
IF EXISTS(SELECT * FROM inserted
WHERE 学号 NOT IN (SELECT 学号 FROM XSDA) )
BEGIN
    RAISERROR ('错误代码1, 违背数据的一致性, 不允许插入!', 16, 1)
    ROLLBACK TRANSACTION
END
GO
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

3. 在数据库XS中创建一触发器kcxx_delete，当从KCXX表中删除一条记录时，检查该记录的课程编码在XSCJ表中是否存在，若该记录的课程编码在XSCJ表中存在，则不允许删除操作，并提示“错误代码2，违背数据的一致性，不允许删除！”。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

```
USE XS
GO
CREATE TRIGGER kcxx_delete
ON KCXX
FOR DELETE
AS
    IF EXISTS (SELECT * FROM deleted
               WHERE 课程编号 IN (SELECT 课程编号 FROM XSCJ) )
    BEGIN
        RAISERROR ('错误代码2, 违背数据的一致性, 不允许删除!', 16, 1)
        ROLLBACK TRANSACTION
    END
GO
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

4. 修改触发器xsda_update, 使其起到只是不允许更新学号字段的作用。

```
USE XS
```

```
GO
```

```
ALTER TRIGGER xsda_update
```

```
ON xsda
```

```
FOR UPDATE
```

```
AS
```

```
BEGIN
```

```
    update xsda set xsda.学号=deleted.学号 from xsda,deleted where xsda.姓名  
=deleted.姓名 and xsda.学号!=deleted.学号
```

```
END
```

```
GO
```

项目10 创建使用存储过程和触发器

任务1

任务2

实训

实训十 为sale数据库创建存储过程和触发器

1. 创建存储过程p_Sale1，显示每种产品的销售量和销售金额。
2. 创建存储过程p_Sale2，能够根据指定的客户统计汇总该客户购买每种产品的数量和花费金额。
3. 创建存储过程p_Sale3，能够根据指定的产品编号和日期，以输出参数的形式得到该日期该产品的销售量和销售金额。
4. 创建触发器，实现即时更新每种产品的库存数量。
5. 使用IF UPDATE(column)尽可能优化上题中触发器以提高系统效率。

项目10 创建使用存储过程和触发器

任务1

任务2

实训

小结

本项目主要介绍了存储过程和触发器的使用。存储过程是存储在服务器上的一组预编译的SQL语句的集合，而触发器可以看成是特殊的存储过程。触发器在数据库开发过程中，在对数据库的维护和管理等任务中，特别是在维护数据完整性等方面具有不可替代的作用。

1. 存储过程的创建、执行、修改与删除
2. 触发器的创建、修改与删除