项目7 存储过程、触发器的创建和使用

学习目标

- 了解存储过程的优点。
- 掌握常用的系统存储过程。
- 掌握如何创建存储过程。
- 掌握如何调用存储过程。

存储过程、触发器的创建和使用

知识重点

- 创建存储过程。
- 调用存储过程。
- 掌握如何使用Inserted表和deleted表。
- 掌握如何创建INSERT触发器、UPDATE触发器、DELETE触发器。

知识难点

- 带输入参数和输出参数的存储过程的创建和调用。
- 使用Inserted表和deleted表。

7.1 知识准备

7.1.1 存储过程概述

- 存储过程（procedure）类似于C语言中的函数，是SQL语句和控制流语句的预编译集合。
- 用来执行管理任务或应用复杂的业务规则。
- 存储过程可以带参数，也可以返回结果。

存储过程相当于C语言中的函数

```
int sum(int a,int b)
{
   int s;
   s =a+b;
   return s ;
}
```

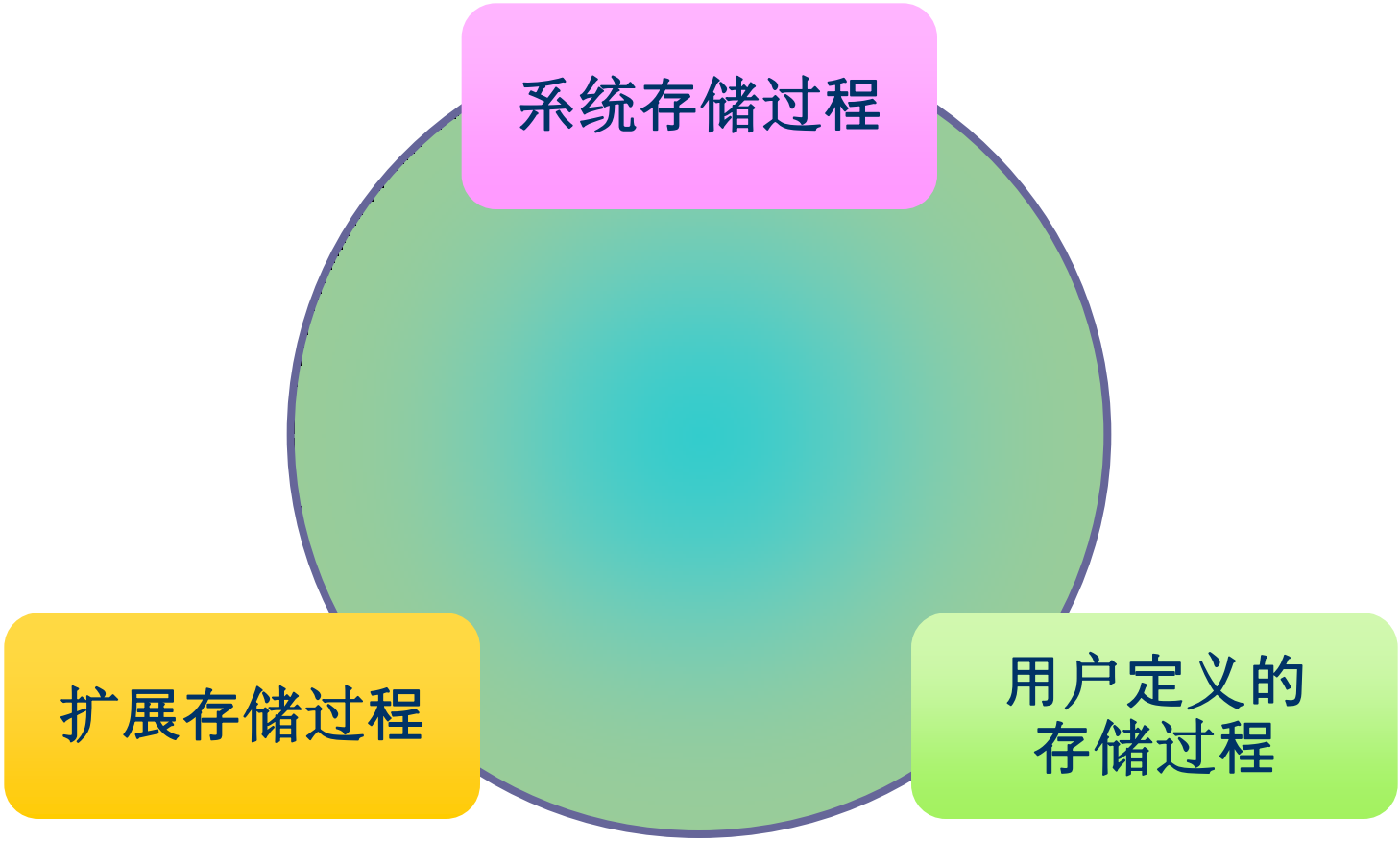
7.1.1 存储过程概述

使用存储过程的优点

- 允许模块化程序设计，实现代码重用。
- 封装复杂操作
- 加快系统运行速度
- 减少网络流量
- 可作为安全机制使用

7.1.1 存储过程概述

存储过程的分类



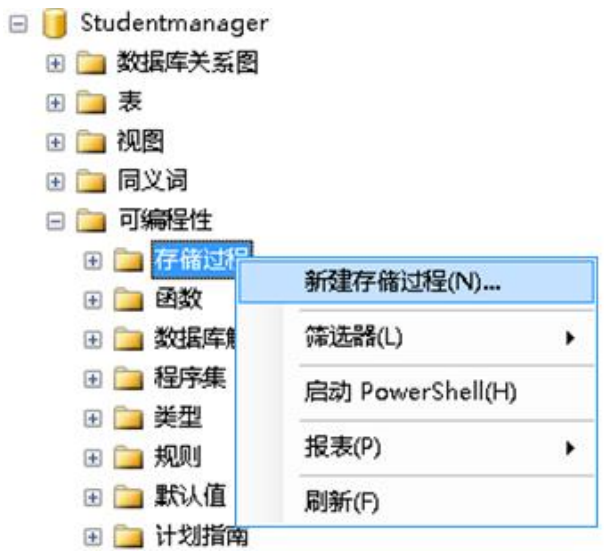
常用的系统存储过程

系统存储过程	说明
sp_databases	列出服务器上的所有数据库
sp_helpdb	报告有关指定数据库或所有数据库的信息
sp_renamedb	更改数据库的名称
sp_tables	返回当前环境下可查询的对象的列表
sp_columns	回某个表列的信息
sp_help	查看某个表的所有信息
sp_helpconstraint	查看某个表的约束
sp_helpindex	查看某个表的索引
sp_stored_procedures	列出当前环境中的所有存储过程
sp_password	添加或修改登录帐户的密码
sp_helptext	显示默认值、未加密的存储过程、用户定义的存储过程、触发器或视图的实际文本

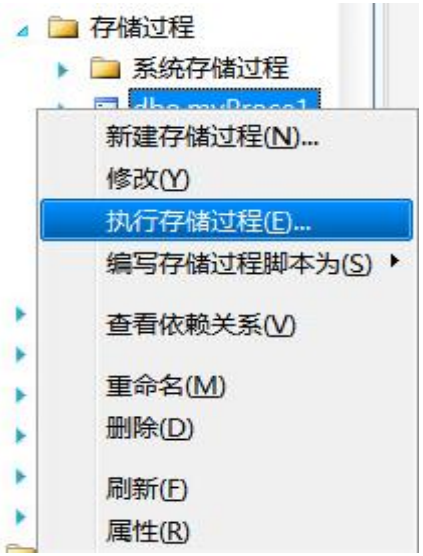
7.2 项目实施

7.2.1 任务1：创建存储过程

子任务1：使用SQL Server Management Studio创建存储过程



在SQL Server Management Studio中创建存储过程



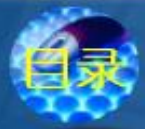
执行存储过程

子任务2：使用T-SQL创建存储过程

创建存储过程的T-SQL语法为：

```
CREATE   PROC[EDURE]   存储过程名
@参数1   数据类型 [= 默认值] [OUTPUT],
……
@参数n   数据类型 [= 默认值] [OUTPUT]
AS
SQL语句
```

特别注意：如果没有output表示该参数为输入参数



存储过程的执行

Exec 存储过程的名称
@参数=值

存储过程实例

1. 创建一个带有输入参数的存储过程pro_stu1，要求根据输入学生的学号来查询相应的学生信息

```
create procedure Pro_stu1
@studid varchar(15)
as
select * from student_tab
where stud_id=@studid
```

测试：

- exec Pro_stu1
- @studid='101'

存储过程实例

2. 创建带有两个参数的存储过程PRO_score0，一个参数用于表示课程的门数，一个参数表示该课程的成绩。要求：参数1的值为2，参数2的值为:0-100的整数，查询有2（参数1）门以上的课程是参数2以上的学生学号

- create procedure Pro_score0
- @course_count int,
- @grade int
- as
- select stud_id as 学生学号
- from score_tab
- where score>=@grade
- group by stud_id
- having COUNTstud_id)>=@course_count
- 测试：
- exec pro_score0
- @course_count=2,
- @grade=75

存储过程实例

3. 创建一个带有输入参数和输出参数的存储过程 pro_teacher，其中输入参数用于接收教师工号，输出参数用于显示该教师的姓名、性别、职称和所在的系部

```
●create procedure pro_teacher
●@teacherid varchar(15),
●@name varchar(20) output,
●@xibi varchar(2) output,
●@ziche varchar(20) output,
●@xibu varchar(20) output
●as
● SELECT @name=teacher_tab.teacher_name,
@xibi=gender_tab.gender_name,@ziche=title_tab.title_name,@xibu=depart_tab.depart_name
● FROM gender_tab INNER JOIN teacher_tab ON gender_tab.gender_id =teacher_tab.gender_id INNER JOIN title_tab ON teacher_tab.title_id =title_tab.title_id INNER JOIN depart_tab ON teacher_tab.depart_id =depart_tab.depart_id
● where teacher_tab.teacher_id=@teacherid
```

存储过程实例

```
--测试
declare
      @teacherid varchar(15),
      @name varchar(20),
      @xibi varchar(2),
      @ziche varchar(20),
      @xibu varchar(20)
set @teacherid='302'
execute pro_teacher
 @teacherid=@teacherid,
 @name= @name output,
 @xibi=@xibi output,
 @ziche= @ziche output,
 @xibu=@xibu output
SELECT  @name as '姓名',
                @xibi as '性别',
                @ziche as '职称',
                @xibu as '系部'
```

存储过程实例

4.创建带有一个输入参数的存储过程pro_course，根据输入课程号来查询该课程的详细信息，如果存在则显示，反之，提示信息“不存在该课程”，测试数据：100101。（和T-SQL结合）

```
create procedure pro_course
@cid char(6)
as
if exists(select * from course_tab where course_id=@cid)
select * from course_tab where course_id=@cid
else
print'不存在该课程 ‘
```

测试：

```
exec pro_course
@cid='100101'
```

存储过程的修改

```
Alter proc 存储过程的名称
As
Sql语句
```

删除存储过程

使用T-SQL语句删除存储过程

DROP {PROC|PROCEDURE}{[schema_name.]procedure} [,...n]

其中一些参数的含义如下：

schema_name：过程所属架构的名称。

procedure：要删除的存储过程或存储过程组的名称。

删除存储过程proc_stu的代码如下：

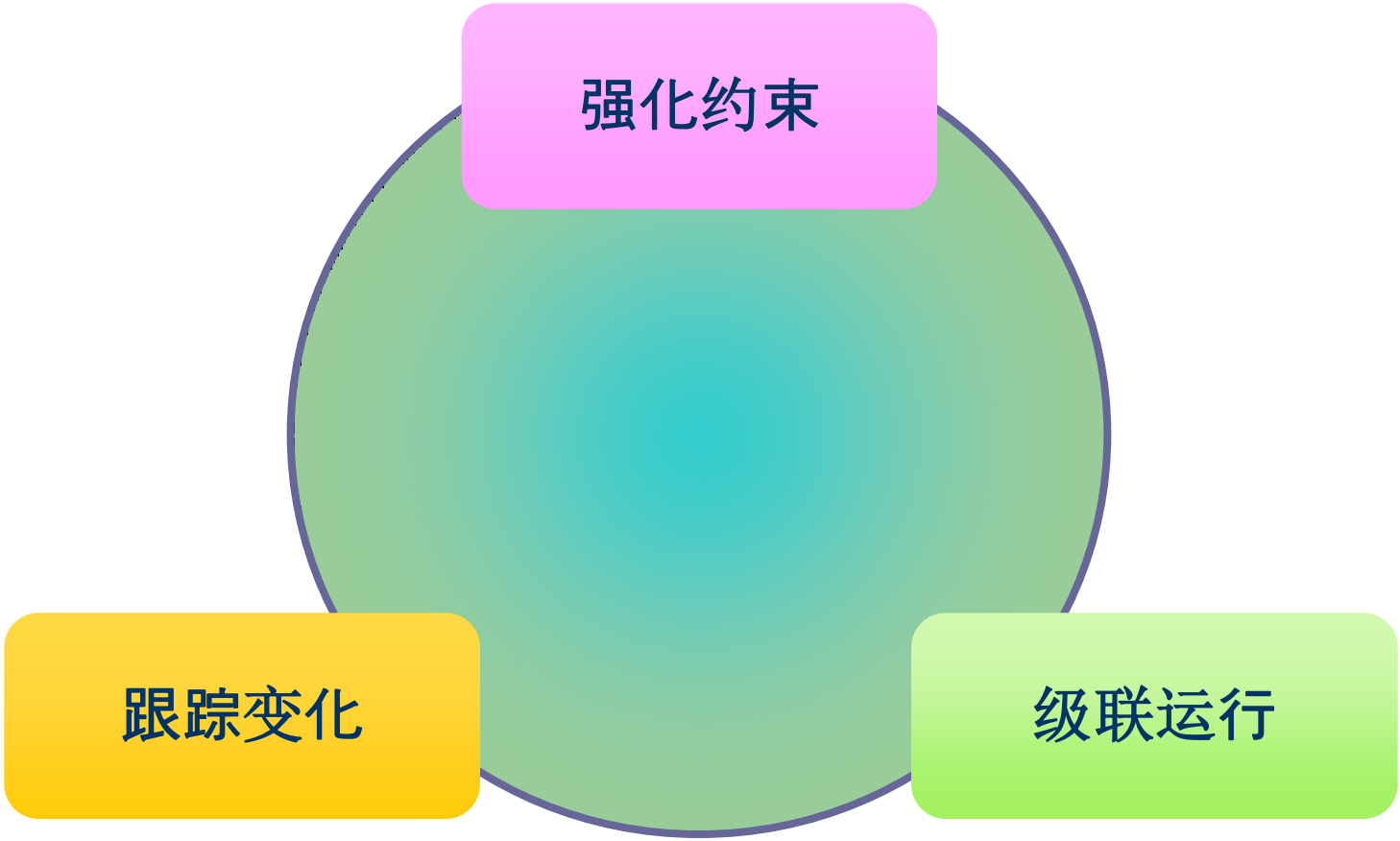
```
DROP PROCEDURE proc_stu
```

7.3 触发器概述

- 触发器是在对表进行插入、更新或删除操作时自动执行的存储过程
- 触发器通常用于强制业务规则
- 触发器是一种高级约束，可以定义比用CHECK约束更为复杂的约束
  - 可执行复杂的SQL语句（if/while/case）
  - 可引用其它表中的列

7.3 触发器概述

触发器的功能



7.3 触发器概述

DDL 触发器

当服务器和数据库中发生数据定义语言（DDL）事件时将调用DDL触发器，一般用于在数据库中执行管理任务 。

DML 触发器

当数据库服务器中发生数据操作语言（**DML**）事件时将调用**DML**触发器，**MNL**触发器包括INSERT、UPDATE 和DELETE 触发器。

7.3 触发器概述

触发器的工作原理

- 当对某一表进行诸如UPDATE、INSERT、DELETE操作时，如果在这些操作上定义了触发器，SQL Server就会自动执行触发器（执行触发器所定义的SQL语句），从而确保对数据的处理必须符合由这些SQL语句所定义的规则。
- inserted 表
  - 临时保存了插入或更新后的记录行
  - 可以从inserted表中检查插入的数据是否满足业务需求
  - 如果不满足，则向用户报告错误消息，并回滚插入操作
- deleted 表
  - 临时保存了删除或更新前的记录行
  - 可以从deleted表中检查被删除的数据是否满足业务需求
  - 如果不满足，则向用户报告错误消息，并回滚插入操作

7.3 触发器概述

inserted表和deleted表存放的信息

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

7.4 任务1：创建触发器

子任务1：在SQL Server Management Studio中创建触发器

（1） 单击【开始】/【所有程序】/【Microsoft SQL Server 2008】/【SQL Server Management Studio】，启动SQL Server Management Studio工具。

（2） 在“对象资源管理器中”中，连接到SQL Server 2008数据库引擎实例。

（3） 选择数据库studentmanager，找到要创建触发器的数据表展开表目录，右击“触发器”节点，选择“新建触发器”命令。

（4） 在SQL Server Management Studio管理工具中，系统将打开查询编辑器，并按照触发器的格式显示编码。

（5） 在查询编辑器中，用户根据需要修改触发器名称，添加触发器内容，完成如下触发器的编码，再单击“执行”按钮，在出现“命令已成功完成”的提示后，即完成创建。

使用T-SQL创建触发器

创建触发器的T-SQL语法为：

```
CREATE  trigger   触发器名
on 表名 for 类型(insert、update、delete)
   AS
   SQL语句
```

检测触发器是否存在，
触发器存放在系统表sysobjects中
代码：

```
if exists(select name from sysobjects
where name=‘触发器名名称’ and type=’tr’)
drop trigger 触发器名称
```

项目7 子任务2：创建INSERT触发器

创建insert触发器insert_score_tri，实现向成绩表中添加记录时，首先检查该行的学号列在学生表中和课程表中是否存在，如有一项不成立,则不允许插入，并提示"学生表或课程表中没有相关的记录“

```
--检测触发器是否存在，触发器存放在系统表sysobjects中
if exists(select name from sysobjects where
name='insert_score_tri' and type='tr')
drop trigger insert_score_tri
go
```

项目7 子任务2：创建INSERT触发器

```
create trigger insert_score_tri  on score_tab  for insert
as
   if exists(select * from inserted where inserted.stud_id
not in (select stud_id from student_tab)or
inserted.course_id not in (select course_id from
course_tab))
       begin
         print('学生表或课程表中没有相关的记录')
         rollback
       end
insert into score_tab values('205','100105',98)
```

子任务3：创建update触发器

创建update触发器update_student_tri，当更新了学生表中某位学生的学号时，自动更新成绩表中的学号

```
--检测触发器是否存在，触发器存放在系统表sysobjects中
if exists(select name from sysobjects where name='update_student_tri' and type='tr')
drop trigger update_student_tri
go
create trigger update_student_tri  on student_tab
for update
as
```

子任务3：创建update触发器

创建update触发器update_student_tri，当更新了学生表中某位学生的学号时，自动更新成绩表中的学号

```
 declare @new_Stud_id varchar(15) --声明变量， 更新后的学号
  declare @Stud_id varchar(15)     --声明变量， 未更新的学号
select @Stud_id=deleted.stud_id,@new_Stud_id=inserted.stud_id
from deleted,inserted
where deleted.stud_name=inserted.stud_name
update score_tab
set stud_id=@new_Stud_id
where stud_id=@Stud_id
--提示信息
print '已将成绩表中的学生学号为'+@stud_id+'更新为' +@new_stud_id
  测试：
update student_tab set stud_id='205' where stud_name='罗小红'
```

项目7 子任务4：创建DELETE触发器

创建delete触发器del_student_tri，当删除学生表中的学生信息时，自动将成绩表的中学生成绩删除

```
 if exists(select name from sysobjects where name='del_student_tri' and
type='tr')
drop trigger del_student_tri
go
create trigger del_student_tri  on student_tab
for delete
as
 declare @Stud_id varchar(15)
 select @Stud_id=deleted.stud_id from deleted
delete from score_tab  where stud_id in (select stud_id from deleted)
print'已删除学号为'+@stud_id+'的所有成绩信息'
```

测试：

```
delete from student_tab where stud_name='张海'
```

子任务4：创建DELETE触发器

首先删除表与表的关系（如果存在），然后创建一个del_tri触发器，当删除course表中的数据时，先判断score表中是否存在该门课程的数据，如果不存在，删除成功，反之提示“存在该门课程的信息，不能删除”，测试数据：测试数据：cid='100101'。

```
if exists(select name from sysobjects where name='del_tri' and type='tr')
drop trigger del_tri
go
create trigger del_tri on course_tab
for delete
as
```

子任务4：创建DELETE触发器

首先删除表与表的关系（如果存在），然后创建一个del_tri触发器，当删除course表中的数据时，先判断score表中是否存在该门课程的数据，如果不存在，删除成功，反之提示“存在该门课程的信息，不能删除”，测试数据：测试数据：cid='100101'。

```
●declare @courseid char(6)
●select @courseid=deleted.course_id from deleted
●if exists(select * from score_tab where course_id=@courseid)
● begin
●  rollback transaction
●  print'存在该门课程的信息，不能删除'
●  end
●else
●delete from course_tab where course_id=@courseid
●--测试
●delete from course_tab where course_id='100101'
```

项目7 7.3.3 修改、删除DML触发器

修改：ALTER TRIGGER 触发器名称

当不再需要某个触发器时，可将其删除，删除触发器有两种方法，一种是使用SQL Server Management Studio，步骤与禁用触发器类似，只是在快捷菜单中选择“删除”选项，打开“删除对象”窗口，在其中单击“确定”按钮即可。

另一种是通过T-SQL的DROP TRIGGER语句，其语法格式如下：

DROP TRIGGER schema_name.trigger_name [,...n][;]

其中，参数trigger_name表示要删除的触发器的名称。

7.4 项目总结

存储过程是一组预编译的SQL语句。存储过程可以包含数据操纵语句、逻辑控制语句和调用函数等。

存储过程可加快查询的执行速度，提高数据的访问速度，帮助实现模块化编程，保持数据的一致性和提高安全性。

存储过程可分为两种：系统存储过程和用户定义的存储过程。

EXECUTE语句用于调用存储过程。

存储过程的参数分为输入参数和输出参数，输入参数用来向存储过程中传入（输入）值，输出参数则从存储过程中返回（输出）值，并且后面跟随OUTPUT关键字。

触发器是对表进行插入、更新或删除操作时自动执行的存储过程。触发器通常用于强制业务规则。

触发器一般都需要使用临时表，即deleted表和inserted表，它们保存了被删除或插入的记录行副本。

从触发的条件来分，触发器包括INSERT触发器、UPDATE触发器和DELETE触发器。