

学号：XXXXXXX 姓名：XXX EMAIL: XXX 学院：数学与计算机学院 专业：XXX

《数据库应用实践》

实验二：数据库管理系统的维护与管理

实验目的：

掌握 DBMS 提供的数据库完整性功能；理解数据库安全性，掌握 DBMS 的用户和权限管理；理解存储过程概念，掌握存储过程的使用方法；掌握触发器的使用；掌握备份、还原的基本技术。

实验环境：

操作系统：Windows 7 SP1 X64

数据库管理系统：Microsoft SQL Server 2008 R2

实验内容：

(一) 数据库安全性

1. 添加登录帐号与数据库用户。

首先需要将系统身份验证模式改为混合验证模式

- ✓ SQL Server实例上右击鼠标，从弹出的菜单中选择“属性”命令弹出“服务器属性”窗口
- ✓ 在该窗口左边的“选择页”上，单击“安全性”选项，
- ✓ 在显示窗口的“服务器身份验证”部分，将验证模式改为混合验证模式。

之后修改 SA 的登录密码，并将该帐号改为启用方式


- ✓ 在 SSMS 的对象资源管理器中，依次展开“安全性”，在“登录名”节点选“SA 登录名”节点上右击鼠标，从弹出的菜单中选择“属性”命令弹出“登录属性”窗口，在“常规选择页”上，修改密码，在状态选择页”上，将“SA 登录名”设置为启用状态。

最后 重新启动服务，以 SA 身份登录系统，完成后续操作

1) 创建登录名并设定登陆密码

```
create login sqltest with password='sqltest123'
```

可登陆，但无访问该数据库的权限。

 charkey (SQL Server 10.50.2500 - sqltest)

2) 添加用户

```
use student  
create user sqltest for login sqltest
```

注意执行这语句的必须是 SA 或具有相关权限的登录名

2 对用户授予权限；收回权限。

注意执行以下语句的必须是 SA 或具有相关权限的数据库用户，数据库及相关数据对象是已经存在的。

1) 授予权限

授予对 student 数据库中 student 表的 select 权限

```
use student  
grant select on student to sqltest
```

可以用 sqltest 登录系统，验证是否具有查询权限，是否不具有其它权限。

2) 收回权限

收回对 student 数据库中 student 表的 select 权限

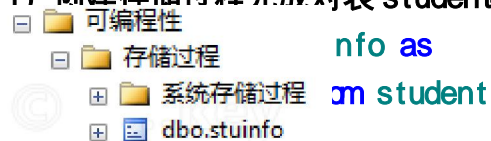
```
use student  
revoke select on student to sqltest
```

再次用 sqltest 登录系统，验证是否已经不能查询该表格。

(二) 触发器，存储过程的使用

1. 创建存储过程

1) 创建存储过程完成对表 student 中所有学生学号和姓名的查询



```
use student  
create procedure stuinfo as  
select Sno, Sname from student
```

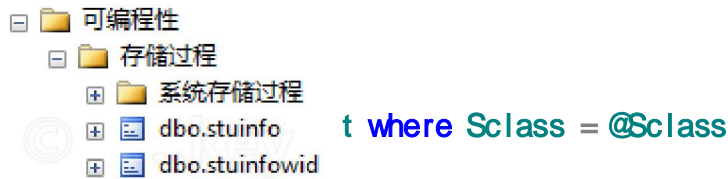
执行所创建的存储过程

```
exec stuinfo
```

	Sno	Sname
1	221002130	郝静
2	031002222	刘滔滔
3	031002543	吕珊珊
4	031002617	天涯
5	031002314	田磊
6	121003444	王欢
7	221002210	王丽
8	121003310	夏宇
9	121003212	杨静蕾
10	221002314	张峰
11	221002208	张伟楠
12	121001623	赵本伟

2) 创建带参数的存数过程完成对表 student 中指定班级学生信息的查询

```
create procedure stuinfo
```



执行所创建的存储过程

```
exec stuinfo
```

```
@Sclass = 6
```

	Sno	Sname	Ssex	Sbirth	Sclass	Specialty	Tel
1	031002617	天涯	男	1991-11-11 00:00:00.000	6	计算机应用技术	NULL
2	121001623	赵本伟	男	1991-04-26 00:00:00.000	6	信息安全	NULL
3	121001624	赵本伟1	男	1991-04-26 00:00:00.000	6	信息安全	NULL

3) 创建具有返回值的存储过程完成对表 student 中年龄大于给定值的人数

```
create procedure stuinfowidreturn
```

```
@Sage int, @studentNum smallint output
```

```
as
```

```
set @studentNum = (  
    select COUNT(*) from student  
    where year(getdate())-year(Sbirth) > @Sage)
```

```
print @studentNum
```

执行所创建的存储过程

```
declare @Sage int, @studentNum smallint
```

```
set @Sage = 22
```

```
exec stuinfowidreturn
```

```
@Sage,@studentNum
```

返回值：3

2. 修改存储过程

```
alter procedure stuinfo as
```

```
select * from student where Sclass = 6
```

执行所创建的存储过程

	Sno	Sname	Ssex	Sbirth	Sclass	Specialty	Tel
1	031002617	天涯	男	1991-11-11 00:00:00.000	6	计算机应用技术	NULL
2	121001623	赵本伟	男	1991-04-26 00:00:00.000	6	信息安全	NULL
3	121001624	赵本伟1	男	1991-04-26 00:00:00.000	6	信息安全	NULL

3. 删除存储过程

```
drop procedure stuinfo
```

4. 创建触发器

创建 DM 触发器的一般形式：

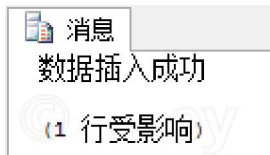
```
create trigger trigger_name
on {table|view}
{for|after|instead of}
{[insert],[update],[delete]}
[with encryption]
as
if UPDATE(column_name)
[ {and|or} update (column name)... ]
sql_statements
```

1) 创建 insert/delete 触发器

a) 简单触发器：

```
create trigger insertinfo
on student
for insert as
print '数据插入成功'
```

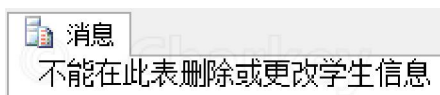
当插入一条记录，系统显示



b) 使用 instead of 关键字

```
create trigger updateinstead
on SignCourse
instead of delete, update
as print '不能在此表删除或更改学生信息'
```

如果想删除或者更改，则出现如下提示：



2) 创建 update 触发器

创建触发器不允许修改学生成绩一栏：

```
create trigger updatereollback
on SignCourse
for update
```

```
as if UPDATE(Grade)
rollback transaction
```

尝试更改学生成绩，提示如下：



3) 修改触发器

修改触发器的具体语法如下：

```
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)...]
sql_statements
```

4) 删除触发器

```
drop trigger trigger_name
```

如：drop trigger updatereollback则删除了刚才创建的触发器

5) 禁止或启动触发器

```
alter trigger table_name
{enable|disable} trigger
{all|trigger_name[,...trigger_name]}
```

另外，DDL触发器是一种特殊的触发器，也是通过事件来激活的，并执行其中的SQL语句。但是与DM触发器不同的是，DDL触发器是响应create alter drop等开头的语句的。

(三) 数据库备份还原

1) 完整备份

```
use master
```

```
go
```

```
backup database student to disk='d:\studentBackup.dat'
```

已为数据库 'student', 文件 'student' (位于文件 1 上)处理了 176 页。
已为数据库 'student', 文件 'student_log' (位于文件 1 上)处理了 4 页。
BACKUP DATABASE 成功处理了 180 页, 花费 0.583 秒(2.518 MB/秒)。

在D盘可以看到该数据文件。

2) 差异备份

```
use student
insert into Student(Sno,Sname,Ssex,Sbirth,Sclass,Specialty)
values('121001700','AAAA','男','1991-04-26',6,'信息安全')

use master
go
backup database student to disk='c:\studentBackupDiff.dat'
with differential
```

已为数据库 'student', 文件 'student' (位于文件 1 上)处理了 48 页。
已为数据库 'student', 文件 'student_log' (位于文件 1 上)处理了 2 页。
BACKUP DATABASE WITH DIFFERENTIAL 成功处理了 50 页, 花费 0.323 秒(1.247 MB/秒)。

在C盘可以看到该数据文件。

3) 日志备份

```
use master
go
BACKUP LOG student TO disk='c:\studentlog.dat'
```

在C盘可以看到该数据文件。

4) 数据库还原

```
use master
go
restore database[student]
from disk='d:\studentBackup.dat'
WITH NORECOVERY
GO
restore database[student]
from disk='c:\studentBackupDiff.dat'
WITH NORECOVERY
GO
Restore LOG [student]
From disk='c:\studentlog.dat'
```

已为数据库 'student', 文件 'student' (位于文件 1 上)处理了 176 页。
已为数据库 'student', 文件 'student_log' (位于文件 1 上)处理了 4 页。
RESTORE DATABASE 成功处理了 180 页, 花费 0.108 秒(13.596 MB/秒)。
已为数据库 'student', 文件 'student' (位于文件 1 上)处理了 48 页。
已为数据库 'student', 文件 'student_log' (位于文件 1 上)处理了 2 页。
RESTORE DATABASE 成功处理了 50 页, 花费 0.049 秒(8.223 MB/秒)。
RESTORE LOG 成功处理了 0 页, 花费 0.003 秒(0.000 MB/秒)。