

# 厦门大学计算机科学系本科生课程

## 《数据库系统原理》实验三

### 数据更新、视图、权限管理

学号：22920202204570 姓名：李波

## 实验 3.1 数据更新

### 1 实验内容

- (1) 使用 `INSERT INTO` 语句插入数据，包括插入一个元组或将子查询的结果插入到数据库中两种方式。
- (2) 使用 `SELECT INTO` 语句，产生一个新表并插入数据。
- (3) 使用 `UPDATE` 语句可以修改制定表中满足 `WHERE` 子句条件的元组，有三种修改的方式：修改某个元组的值；修改多个元组的值；带子查询的修改语句。
- (4) 使用 `DELETE` 语句删除数据：删除某一个元组的值；删除多个元组的值；带子查询的删除语句。

### 2 实验步骤

在数据库 School 上按下列要求进行数据更新

可在 SQL 代码前加下面两句 SQL 语句，用于确保当前使用的是 School 数据库

```
Use School
```

```
go
```

- (1) 使用 SQL 语句向 STUDENTS 表中插入元组（编号：12345678 名字：LiMing EMAIL: [LM@gmail.com](mailto:LM@gmail.com) 年级：2002）。

```
insert into STUDENTS values('700045678', 'LIMING', 'LX@cdemg.com', 1992)
```

|   | sid       | sname     | email             | grade |
|---|-----------|-----------|-------------------|-------|
| 1 | 700045678 | LIMING    | LX@cdemg.com      | 1992  |
| 2 | 800001216 | gfxrgs    | hhce4@qhldj.gov   | 1992  |
| 3 | 800002933 | vnbnqzsvv | pvhxd4l@zqur.org  | 2002  |
| 4 | 800005753 | waqcj     | hlhq0h8@jdba.gov  | 1992  |
| 5 | 800006682 | fiiluommh | ihzd6_k@kzvft.gov | 1992  |
| 6 | 800006941 | ogvmti    | 62sfhd@lrt.gov    | 1995  |

(2) 对每个课程,求学生的选课人数和学生的最高成绩,并把结果存入数据库。使用 `SELECT INTO` 和 `INSERT INTO` 两种方法实现。

`SELECT INTO` 方法:

```
select cid, count(cid) as cnt, max(score) as maxScore into COURSE1 from CHOICES group by cid
```

| 结果 消息 |       |      |          |
|-------|-------|------|----------|
|       | cid   | cnt  | maxScore |
| 1     | 10008 | 5985 | 99       |
| 2     | 10019 | 6074 | 99       |
| 3     | 10018 | 5969 | 99       |
| 4     | 10040 | 6102 | 99       |
| 5     | 10011 | 6086 | 99       |
| 6     | 10028 | 6042 | 99       |
| 7     | 10035 | 6104 | 99       |
| 8     | 10021 | 5916 | 99       |
| 9     | 10046 | 6069 | 99       |
| 10    | 10032 | 6073 | 99       |
| 11    | 10005 | 6031 | 99       |
| 12    | 10050 | 6077 | 99       |
| 13    | 10034 | 6000 | 99       |

`INSERT INTO` 方法:

```
create table COURSE2
(
    cid char(10),
    cnt int,
    maxScore int
);
insert into COURSE2 select cid, count(cid), max(score) from CHOICES group by cid
```

| 结果 消息 |       |      |          |
|-------|-------|------|----------|
|       | cid   | cnt  | maxScore |
| 1     | 10008 | 5985 | 99       |
| 2     | 10019 | 6074 | 99       |
| 3     | 10018 | 5969 | 99       |
| 4     | 10040 | 6102 | 99       |
| 5     | 10011 | 6086 | 99       |
| 6     | 10028 | 6042 | 99       |
| 7     | 10035 | 6104 | 99       |
| 8     | 10021 | 5916 | 99       |
| 9     | 10046 | 6069 | 99       |
| 10    | 10032 | 6073 | 99       |
| 11    | 10005 | 6031 | 99       |
| 12    | 10050 | 6077 | 99       |
| 13    | 10034 | 6000 | 99       |
| 14    | 10015 | 5990 | 99       |
| 15    | 10036 | 6043 | 99       |

(3) 在 STUDENTS 表中使用 SQL 语句将姓名为 LiMing 的学生的 EMAIL 改为 [LM@qq.com](mailto:LM@qq.com)。

```
update STUDENTS set email = 'LM@qq.com' where sname = 'LiMing'
```

|   | sid       | sname    | email            | grade |
|---|-----------|----------|------------------|-------|
| 1 | 700045678 | LIMING   | LM@qq.com        | 1992  |
| 2 | 800001216 | gfxrgs   | hhce4@qhldj.gov  | 1992  |
| 3 | 800002933 | vnbqzsvv | pvhxd41@zqur.org | 2002  |

(4) 在 TEACHERS 表中使用 SQL 语句将所有教师的工资翻倍。

```
update TEACHERS set salary = salary * 2
```

|    | tid       | tname    | email            | salary |
|----|-----------|----------|------------------|--------|
| 1  | 199999999 | abcd     | abcd@efgh.com    | 6000   |
| 2  | 200003125 | fqmmyi   | wcjcg@glq.net    | 8856   |
| 3  | 200005322 | udjom    | gd5l8@ppmlf.gov  | 2746   |
| 4  | 200009423 | jhoulvtr | l2cj@pzc.gov     | 7572   |
| 5  | 200010493 | xzgkflko | zcf_m@def.com    | 7664   |
| 6  | 200011899 | lwujzim  | s4zrck@pew.net   | NULL   |
| 7  | 200016731 | nfgrod   | dprcy@qxqi.com   | 8638   |
| 8  | 200017039 | qibsy    | b8mtpe@enbba.edu | 9358   |
| 9  | 200023671 | zfuxiahq | lr_32t@cfo.net   | 8354   |
| 10 | 200025513 | rnupx    | c7ce@ayt.org     | 6048   |
| 11 | 200026895 | nuyoq    | xibl@izd.edu     | 10660  |
| 12 | 200033112 | awboicq  | hjct6g@izgm.edu  | 8264   |
| 13 | 200047819 | lmogygb  | if095@dxdm.org   | 8936   |

(5) 将姓名为 waqcj 的学生的课程 C++ 的成绩加 10 分。

```
update CHOICES set score = score + 10 where no =
(select no from CHOICES, STUDENTS, COURSES
where CHOICES.sid = STUDENTS.sid
and CHOICES.cid = COURSES.cid
and STUDENTS.sname = 'waqcj'
and COURSES.cname = 'C++')
```

修改前成绩:

|   | no        | sid       | tid       | cid   | score |
|---|-----------|-----------|-----------|-------|-------|
| 1 | 502014878 | 800005753 | 215295550 | 10005 | 66    |

修改后成绩:

|   | no        | sid       | tid       | cid   | score |
|---|-----------|-----------|-----------|-------|-------|
| 1 | 502014878 | 800005753 | 215295550 | 10005 | 76    |

(6) 在 STUDENTS 表中使用 SQL 语句删除姓名为 LiMing 的学生信息。

```
delete from STUDENTS where sname = 'LiMing'
```

The screenshot shows a SQL query editor with the text: `select * from STUDENTS where sname = 'LiMing'`. Below the editor, there is a tab labeled "结果" (Results) and a sub-tab labeled "消息" (Messages). The "消息" sub-tab is active, displaying the execution results. The results are shown in a table with columns: sid, sname, email, and grade. The table is currently empty, indicating that the deletion was successful.

(7) 删除所有选修课程 C 的选课记录。

删除前:

|   | no        | sid       | tid       | cid   | score |
|---|-----------|-----------|-----------|-------|-------|
| 1 | 500019253 | 826800279 | 254743764 | 10042 | 52    |
| 2 | 500026725 | 828743446 | 264652927 | 10042 | 85    |
| 3 | 500055883 | 841348939 | 212523885 | 10042 | 94    |
| 4 | 500077819 | 832815482 | 241749288 | 10042 | 92    |
| 5 | 500103431 | 878941549 | 268548571 | 10042 | 73    |
| 6 | 500115088 | 800575453 | 213781494 | 10042 | 78    |
| 7 | 500119959 | 893003636 | 214156099 | 10042 | 88    |
| 8 | 500130427 | 857435877 | 201045902 | 10042 | 52    |
| 9 | 500134392 | 845075819 | 276174453 | 10042 | 89    |

```
delete from CHOICES where cid =  
(select cid from COURSES where cname = 'C')
```

删除后:

The screenshot shows a SQL query editor with the text: `select * from CHOICES where cid = (select cid from COURSES where cname = 'C')`. Below the editor, there is a tab labeled "结果" (Results) and a sub-tab labeled "消息" (Messages). The "消息" sub-tab is active, displaying the execution results. The results are shown in a table with columns: no, sid, tid, cid, and score. The table is currently empty, indicating that the deletion was successful.

(8) 对 COURSES 表做删去时间>80 的元组的操作, 讨论该删除操作所受到的约束。

删除前:

|   | cid   | cname             | hour |
|---|-------|-------------------|------|
| 1 | 10001 | database          | 96   |
| 2 | 10002 | operating system  | 88   |
| 3 | 10003 | computer graphics | 48   |
| 4 | 10004 | java              | 48   |

```
delete from COURSES where hour > 80
```

删除后:

|    | cid   | cname                | hour |
|----|-------|----------------------|------|
| 1  | 10003 | computer graphics    | 48   |
| 2  | 10004 | java                 | 48   |
| 3  | 10005 | c++                  | 60   |
| 4  | 10006 | design pattern       | 48   |
| 5  | 10008 | data structure       | 60   |
| 6  | 10010 | software engineering | 50   |
| 7  | 10014 | computer network     | 60   |
| 8  | 10015 | tcp/ip protocol      | 68   |
| 9  | 10017 | algorithm            | 72   |
| 10 | 10028 | architectonics       | 50   |
| 11 | 10029 | compiling principle  | 62   |
| 12 | 10033 | real-time system     | 48   |
| 13 | 10042 | c                    | 48   |
| 14 | 10047 | computer interface   | 48   |

该表并未定义外键，所以该删除操作没有约束。

## 实验 3.2 视图操作

### 1 实验内容

- (1) 使用 `CREATE VIEW` 命令建立视图。
- (2) 对视图进行查询
- (3) 更新视图
- (4) 使用 `DROP VIEW` 命令删除视图。

### 2 实验步骤

在数据库 School 上按下列要求进行视图的有关操作

*可在 SQL 代码前加下面两句 SQL 语句，用于确保当前使用的是 School 数据库*

```
Use School
```

```
go
```

- (1) 建立薪水大于 3000 的教师的视图 t\_view，并要求进行修改和插入操作时仍需保证该视图只有薪水大于 3000 的教师信息。

```
create view t_view as
select * from TEACHERS where salary > 3000
with check option
```

(2) 在视图 t\_view 中查询邮件地址为 xibl@izd.edu 的教师的相关信息。

```
select * from t_view where email = 'xibl@izd.edu'
```

|   | tid       | tname  | email        | salary |
|---|-----------|--------|--------------|--------|
| 1 | 200026895 | nuyyoq | xibl@izd.edu | 10660  |

(3) 向视图 t\_view 中插入一个新的教师记录，其中教师编号为 199999998，姓名为 abc，邮件地址为 abc@def.com，薪水为 5000。

```
insert into t_view values('199999998', 'abc', 'abc@def.com', 5000)
```

|   | tid       | tname | email       | salary |
|---|-----------|-------|-------------|--------|
| 1 | 199999998 | abc   | abc@def.com | 5000   |

(4) 在视图 t\_view 中将编号为 200010493 的教师的薪水改为 6000。

修改前：

|   | tid       | tname    | email         | salary |
|---|-----------|----------|---------------|--------|
| 1 | 200010493 | xzgkflko | zcf_m@def.com | 7664   |

```
update t_view set salary = 6000 where tid = '200010493'
```

修改后：

|   | tid       | tname    | email         | salary |
|---|-----------|----------|---------------|--------|
| 1 | 200010493 | xzgkflko | zcf_m@def.com | 6000   |

(5) 删除视图 t\_view。

删除前：

- 视图
  - 系统视图
  - dbo.cs\_view
  - dbo.t\_view
  - 已删除账本视图

```
drop view t_view
```

删除后：

- 视图
  - 系统视图
  - dbo.cs\_view
  - 已删除账本视图

# 实验 3.3 用户标识与鉴别

## 1 实验内容

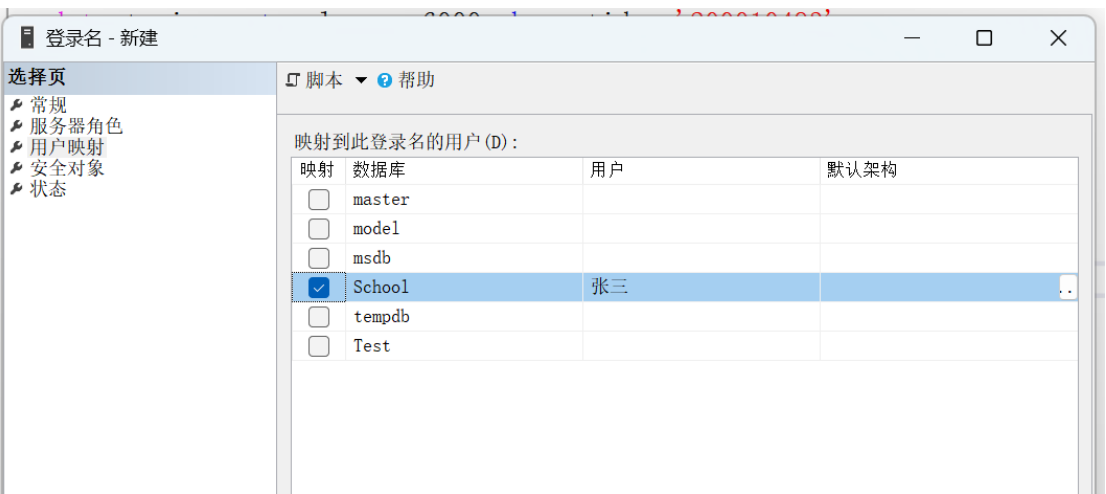
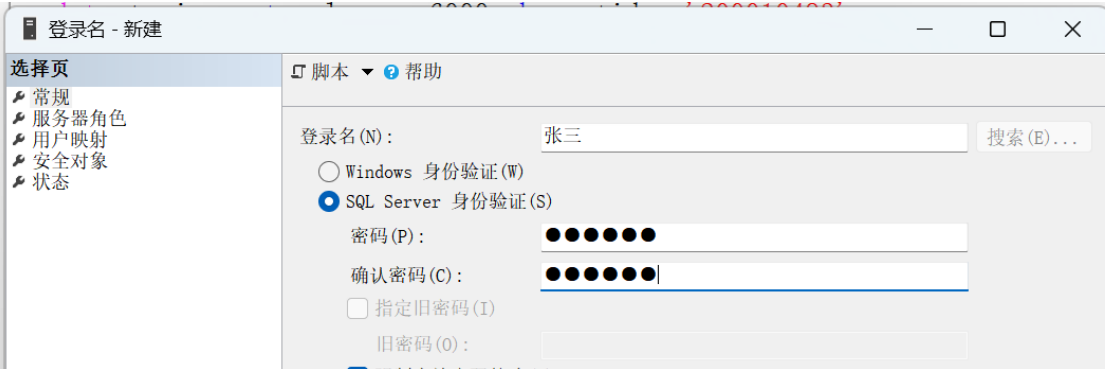
由系统提供一定的方式让用户标识自己的名字和身份。

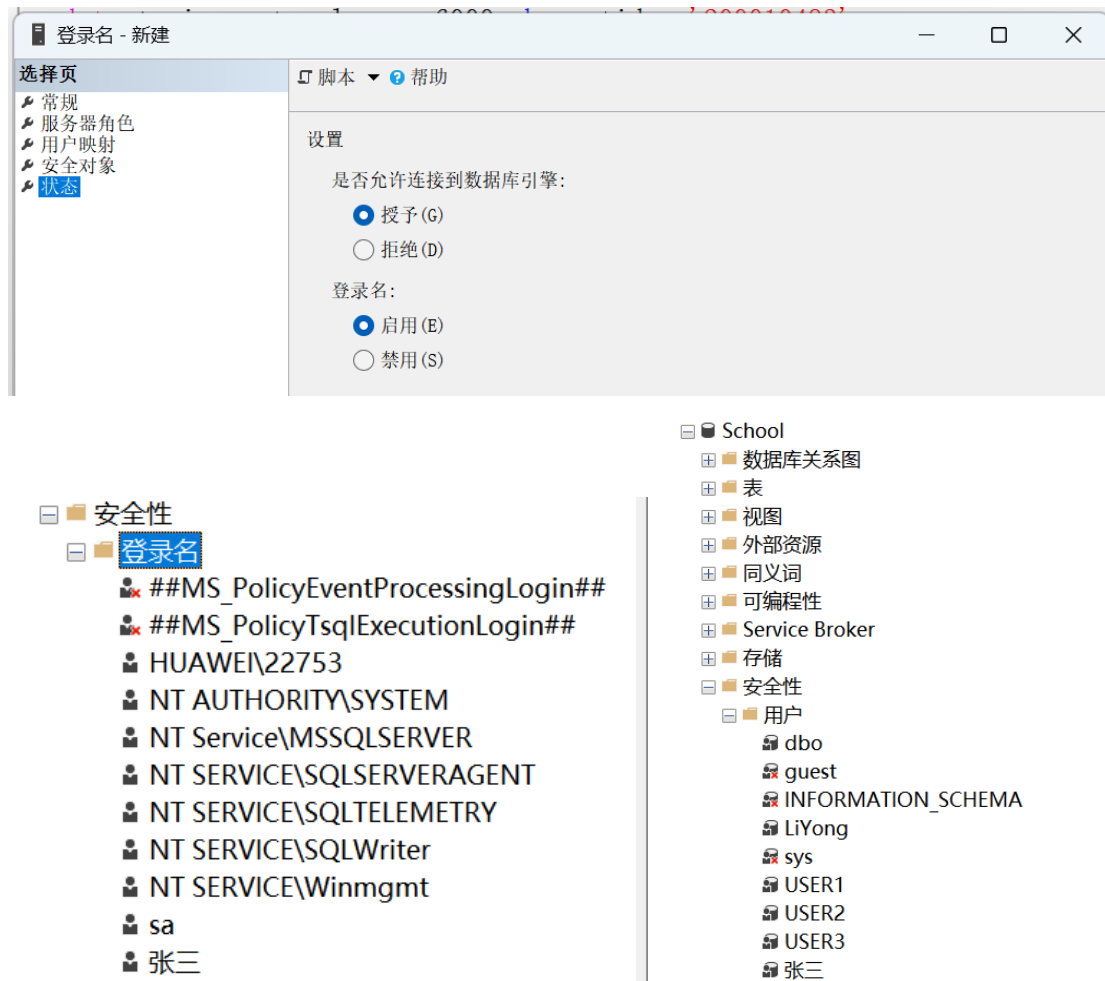
## 2 实验步骤

(1) 在 SSMS 中，设置 SQL Server 的安全认证模式。

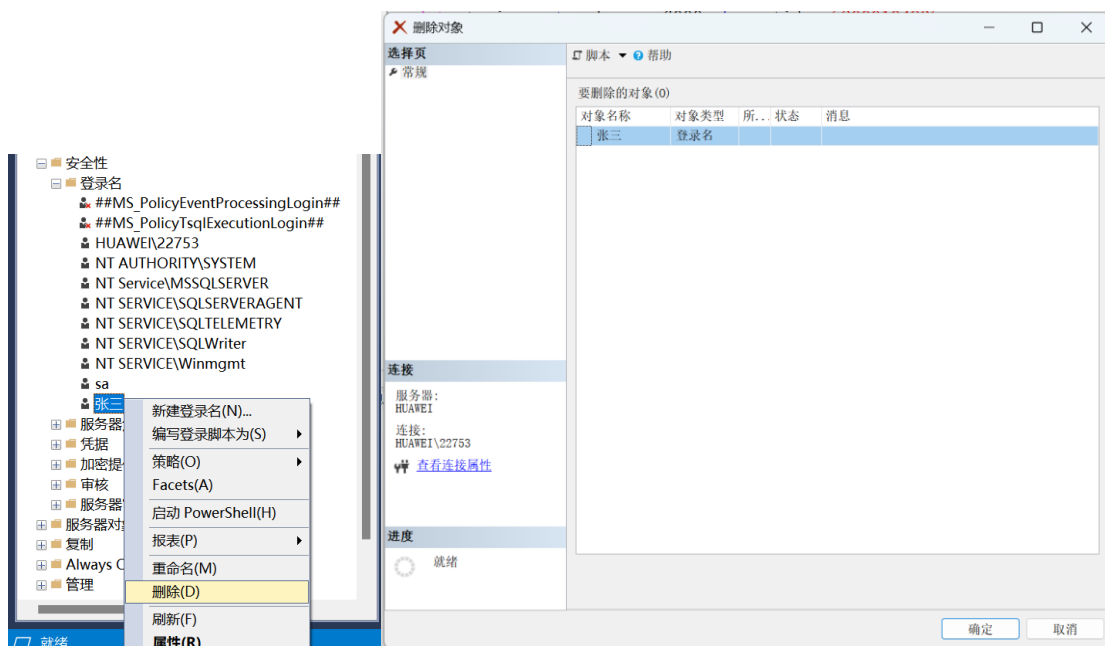


(2) 在 SSMS 中建立一个名为“张三”的登陆用户、School 数据库用户。





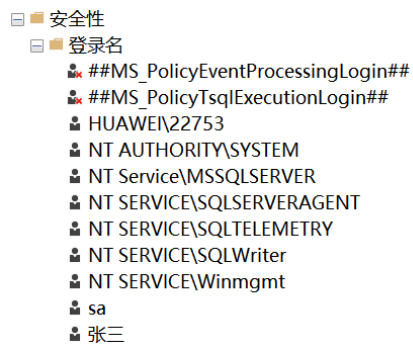
(3) 在 SSMS 中取消“张三”这个用户。





(4) 学习【[参考命令](#)】中的几个命令，了解利用命令实现上述步骤。  
在 SSMS 中建立一个名为“张三”的登陆用户、School 数据库用户。

```
exec sp_addlogin '张三', '123456', 'School'
```

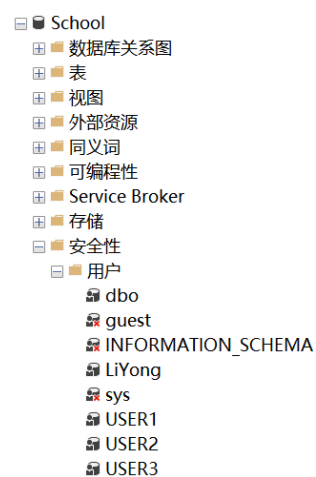


```
exec sp_adduser '张三'
```



在 SSMS 中取消“张三”这个用户。

```
exec sp_dropuser '张三'
```



```
exec sp_droplogin '张三'
```



## 实验 3.4 自主存取控制

### 1 实验内容

定义用户权限，将用户权限登记到数据字典中，并对用户进行合法权限检查。

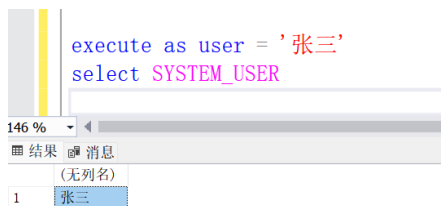
### 2 实验步骤

(1) 在 SSMS 中建立一个名为“张三”的登陆用户、School 数据库的用户。参见实验 3.3 的试验步骤 (2)

```
exec sp_addlogin '张三', '123456', 'School'
exec sp_adduser '张三'
```

(2) 使用查询验证“张三”这个用户名是否具有对学生表的 `SELECT` 权限。

切换用户



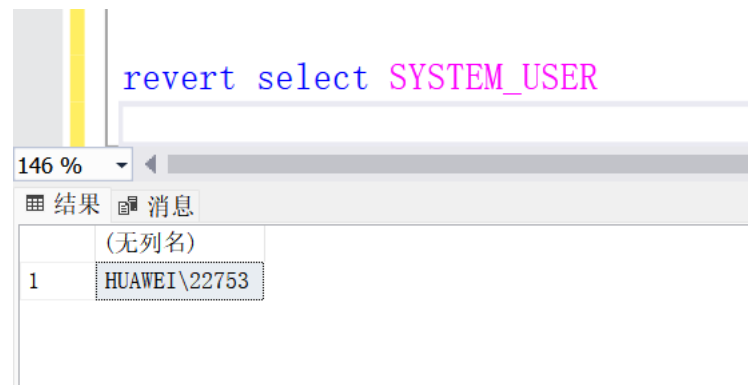
```
select * from STUDENTS
```

消息 229, 级别 14, 状态 5, 第 69 行  
拒绝了对对象 'STUDENTS' (数据库 'School', 架构 'dbo') 的 `SELECT` 权限。

完成时间: 2023-04-17T16:22:09.2588369+08:00

(3) 将 School 数据库的操作权限赋予数据库用户张三。

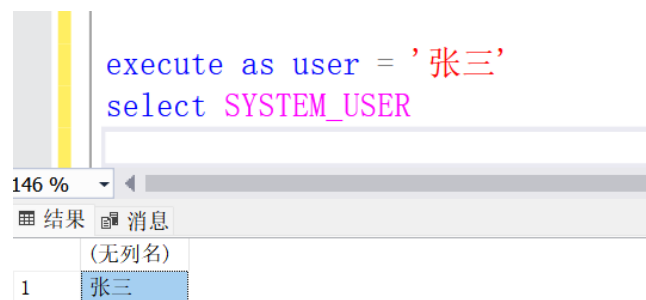
切换管理员身份



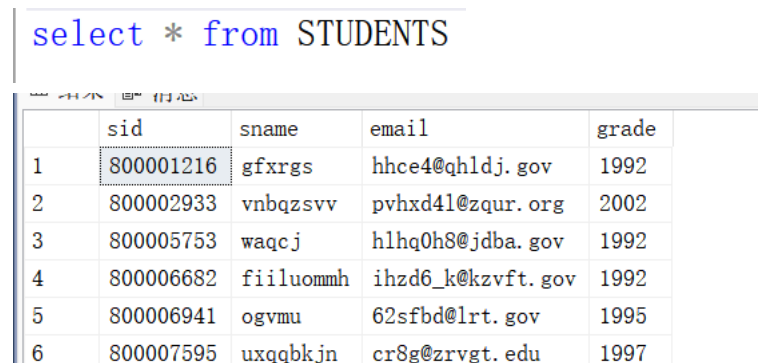
赋予张三权限

```
grant all privileges on STUDENTS to 张三
grant all privileges on CHOICES to 张三
grant all privileges on COURSES to 张三
grant all privileges on TEACHERS to 张三
```

切换回用户张三



查询



# 实验 3.5 视图机制在自主存取控制上的应用

## 1 实验内容

在视图机制上定义用户权限，将用户权限登记到数据字典中，并对用户进行合法权限检查。

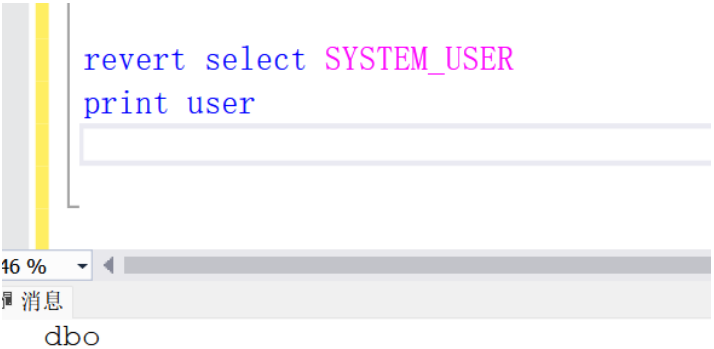
## 2 实验步骤

(1) 在数据库 School 上创建用户“张三”，具体操作参见实验 3.3 中的试验步骤 (2)。

```
exec sp_addlogin '张三', '123456', 'School'
exec sp_adduser '张三'
```

(2) 新建查询，用管理员身份登陆数据库。在 choices 表上创建视图 ch\_view，并显示其内容（选课课程号为 10005）。

切换管理员身份



创建视图 ch\_view

```
create view ch_view as
select * from CHOICES where cid = '10005'
```

```
select * from ch_view
```

|   | no        | sid       | tid       | cid   | score |
|---|-----------|-----------|-----------|-------|-------|
| 1 | 500003262 | 826310502 | 267846042 | 10005 | 90    |
| 2 | 500029445 | 880275978 | 245622362 | 10005 | 91    |
| 3 | 500030717 | 883794999 | 220037007 | 10005 | 82    |
| 4 | 500039740 | 852880400 | 230467893 | 10005 | 83    |
| 5 | 500056410 | 812844702 | 233198108 | 10005 | 74    |
| 6 | 500070692 | 891133170 | 215152459 | 10005 | 79    |

(3) 在视图 ch\_view 上给用户张三赋予 INSERT 的权限。

```
grant insert on ch_view to 张三
```

(4) 将视图 ch\_view 上 score 列的权限赋予用户张三。

```
grant all on ch_view(score) to 张三
```

```
execute as user = '张三'  
select score from ch_view
```

|   | score |
|---|-------|
| 1 | 90    |
| 2 | 91    |
| 3 | 82    |
| 4 | 83    |
| 5 | 74    |
| 6 | 79    |
| 7 | 73    |
| 8 | 68    |
| 9 | 81    |

(5) 以用户张三登陆查询分析器，对 ch\_view 进行查询操作。

```
revert select SYSTEM_USER  
grant all on ch_view to 张三  
execute as user = '张三'  
select * from ch_view
```

|    | no        | sid       | tid       | cid   | score |
|----|-----------|-----------|-----------|-------|-------|
| 1  | 500003262 | 826310502 | 267846042 | 10005 | 90    |
| 2  | 500029445 | 880275978 | 245622362 | 10005 | 91    |
| 3  | 500030717 | 883794999 | 220037007 | 10005 | 82    |
| 4  | 500039740 | 852880400 | 230467893 | 10005 | 83    |
| 5  | 500056410 | 812844702 | 233198108 | 10005 | 74    |
| 6  | 500070692 | 891133170 | 215152459 | 10005 | 79    |
| 7  | 500098913 | 837089679 | 208131015 | 10005 | 73    |
| 8  | 500110951 | 879230818 | 246435150 | 10005 | 68    |
| 9  | 500127998 | 804706477 | 237142778 | 10005 | 81    |
| 10 | 500139090 | 845947855 | 297644124 | 10005 | NULL  |

(6) 以用户张三登陆查询分析器，对 no 为 500127998 的学生的成绩进行修改，改为 90 分。

```
select score from ch_view where no = '500127998'
```

146 %

结果 消息

|   | score |
|---|-------|
| 1 | 81    |

```
update ch_view set score = 90 where no = '500127998'
```

```
select score from ch_view where no = '500127998'
```

146 %

结果 消息

|   | score |
|---|-------|
| 1 | 90    |

(7) 收回对用户张三对视图 ch\_view 查询权限的授权

```
revert select SYSTEM_USER  
revoke select on ch_view from 张三  
execute as user = '张三'  
select * from ch_view
```

(1 行受影响)

消息 229, 级别 14, 状态 5, 第 104 行  
拒绝了对对象 'ch\_view' (数据库 'School', 架构 'dbo') 的 SELECT 权限。

完成时间: 2023-04-17T16:45:53.4317983+08:00

# 实验总结

本次实验主要涉及到数据更新、视图操作以及权限管理三个方面的内容。

在数据更新实验中，我主要练习了select into、insert into、update和delete等命令的使用，对这些命令的使用越发熟练。通过实验，我深刻地认识到了数据更新操作的重要性。在实际应用中，数据更新操作是非常常见的，我们需要通过对数据进行增删改查等操作来维护数据库的数据完整性和一致性。同时，在进行数据更新操作时，由于操作不当，我出现了误删的操作，提醒我需要注意保证操作的安全性，避免误操作等问题。

在视图操作实验中，我主要学习了如何创建视图、修改视图和删除视图等操作。通过实验，我深刻地认识到了视图操作的重要性，感受到了视图和表的区别和联系。在实际应用中，视图操作可以帮助我们更加方便地访问和管理数据库中的数据，同时也可以提高我们的工作效率。

在权限管理实验中，我主要学习了用户标识与鉴别以及用户对表和视图的权限控制等操作。通过对实验指导后面的参考命令和网络资料的学习，我对数据库系统的用户创建和删除有了基本的了解，并学会了有关的基本操作。通过对用户的权限分配练习，我对grant和revoke命令的使用方式有了更多的了解。在实际应用中，权限管理可以帮助我们更好地控制数据库中的数据访问和操作权限，保证数据库的安全性和完整性。