22920192204097 吴雨娟 数据库实验

实验三 数据更新、视图、权限管理

实验目的：熟悉 SQL SERVER 2019 的数据更新、视图、权限管理相关语句

实验环境：WINDOWS10、SQL SERVER 2019

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

1)使用 SQL 语句向 STUDENTS 表中插入元组

（编号：12345678 名字：LiMing EMAIL: LM@gmail.com 年级：2002）

```
insert into STUDENTS values('12345678','LiMing','LM@gmail.com',2002);
select * from STUDENTS where sid ='12345678';
```

结果 消息

	sid	sname	email	grade
1	12345678	LiMing	LM@gmail.com	2002

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。

① 使用 SELECT INTO 实现

```
select cid,COUNT(*) as stNum,MAX(score) as MaxGrade
into C_INFO from CHOICES group by cid;
```

```sql
select * from C_INFO;
```

结果 消息

	cid	stNum	MaxGrade
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

查询已成功执行。 LAPTOP-N3QO3T48 (15.0 RTM) wyj (52) School 00:00:00 49 行

② 使用 INSERT INTO 实现

首先创建表 C_INFO2

```sql
create table C_INFO2(
    cid char(10) primary key,
    stNum int,
    MaxGrade int
);
```

然后进行插入：

```sql
insert into C_INFO2(cid,stNum,MaxGrade)
select cid,COUNT(*),MAX(score)
from CHOICES group by cid;
```

查看结果：

```sql
select * from C_INFO2;
```

结果 消息

	cid	stNum	MaxGrade
1	10001	5898	99
2	10002	6013	99
3	10003	5975	99
4	10005	6031	99
5	10006	6090	99
6	10007	5965	99
7	10008	5985	99
8	10009	5965	99
9	10010	6027	99
10	10011	6086	99
11	10012	5819	99
12	10013	5982	99
13	10014	5866	99

查询已成功执行。 LAPTOP-N3QO3T48 (15.0 RTM) wyj (52) School 00:00:00 49 行

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。

```
update STUDENTS set email = 'LM@qq.com'
where sname = 'LiMing';
select * from STUDENTS where sname = 'LiMing';
```

结果 消息

	sid	sname	email	grade
1	12345678	LiMing	LM@qq.com	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;
select * from TEACHERS;
```

结果 消息

	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	12cj@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	lmogxgh	if095@dxdm.org	8936

查询已成功执行。 LAPTOP-N3QO3T48 (15.0 RTM) wyj (52) School 00:00:00 15,001 行

5)将姓名为 waqcj 的学生的课程 C++的成绩加 10 分。

先查看学生 waqcj 原本 C++课程的分数：

```
select score from CHOICES where cid in(
select cid from COURSES where cname = 'C++')
and sid in(
select sid from STUDENTS where sname = 'waqcj');
```

结果 消息

	score
1	66

然后加 10 分：

```
update CHOICES set score = score +10 where
cid in(
select cid from COURSES where cname = 'C++')
and sid in(
select sid from STUDENTS where sname = 'waqcj');
```

再查看：

结果 消息

	score
1	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';
select * from STUDENTS where sname = 'LiMing';
```

结果 消息

sid	sname	email	grade

7)删除所有选修课程 C 的选课记录。

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

```
select * from CHOICES where cid in(
select cid from COURSES where cname = 'C');
```

结果 消息

no	sid	tid	cid	score

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。

首先查看课程的时间信息：

```
select * from COURSES;
```

结果 消息

	cid	cname	hour
1	10001	database	96
2	10002	operating system	88
3	10003	computer graphics	48
4	10004	java	48
5	10005	c++	60
6	10006	design pattern	48
7	10008	data structure	60
8	10010	software engineering	50
9	10014	computer network	60
10	10015	tcp/ip protocol	68
11	10017	algorithm	72
12	10028	architectonics	50
13	10029	compiling principle	62

查询已成功执行。 LAPTOP-N3QO3T48 (15.0 RTM) wyj (52) School 00:00:00 16 行

然后执行删除操作：

```
delete from COURSES where hour > 80;
select * from COURSES;
```

结果如下：

结果 消息

	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

查询已成功执行。 LAPTOP-N3QO3T48 (15.0 RTM) wyj (52) School 00:00:00 14 行

讨论该删除操作所受到的约束：

删除语句执行时，如果删除元组后会破坏参照完整性，那么元组就不能被删除。因为其他的表中也存在着相应元组的 cid 信息。但在本例中不存在这样的情况，所以删除语句可以成功执行，且显示正确的结果。

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。

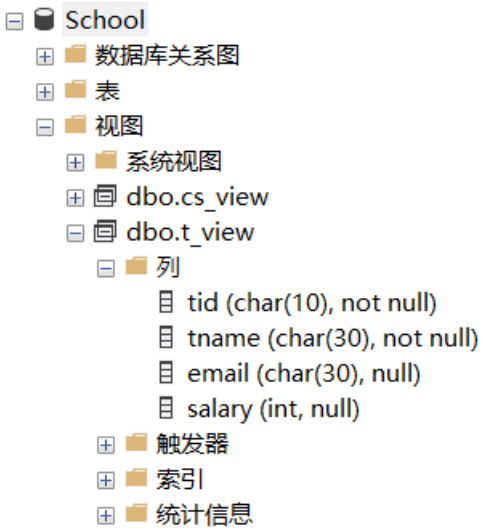
2 实验步骤

在数据库 School 上按下列要求进行视图的有关操作

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。

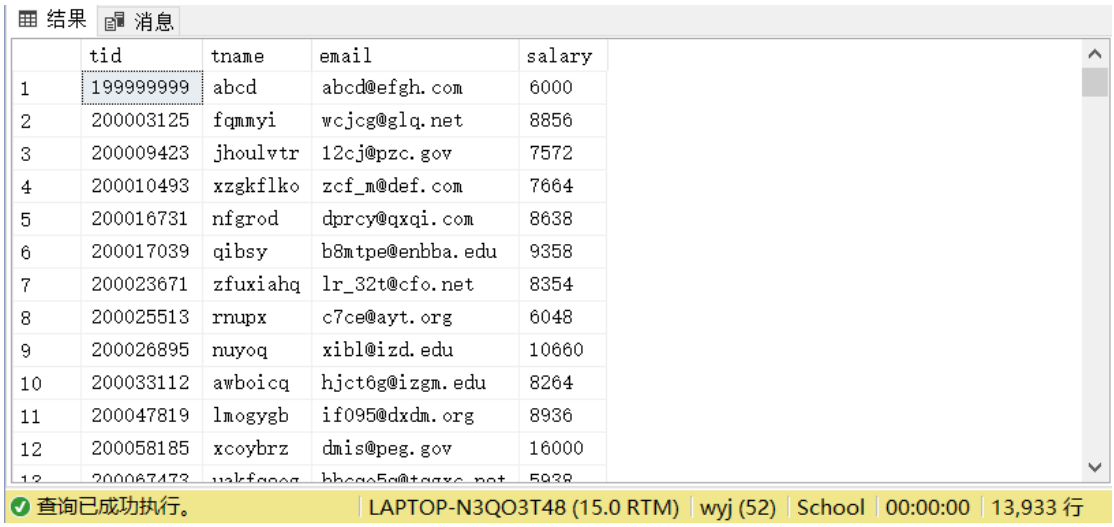
```
create view t_view as
select * from TEACHERS where salary>3000
with check option;
```

建立的视图如下：




```
select * from t_view;
```

视图中的信息如下：



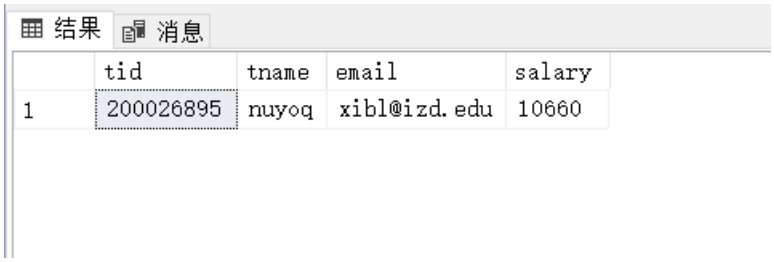
结果 消息

	tid	tname	email	salary
1	199999999	abcd	abcd@efgh.com	6000
2	200003125	fqmmyi	wcjcg@glq.net	8856
3	200009423	jhoulvtr	12cj@pzc.gov	7572
4	200010493	xzgkflko	zcf_m@def.com	7664
5	200016731	nfgrod	dprcy@qxqi.com	8638
6	200017039	qibsy	b8mtpe@enbba.edu	9358
7	200023671	zfuxiahq	lr_32t@cfo.net	8354
8	200025513	rnupx	c7ce@ayt.org	6048
9	200026895	nuyoq	xibl@izd.edu	10660
10	200033112	awboicq	hjct6g@izgm.edu	8264
11	200047819	lmogygb	if095@dxdm.org	8936
12	200058185	xcoybrz	dmis@peg.gov	16000

查询已成功执行。 LAPTOP-N3QO3T48 (15.0 RTM) wyj (52) School 00:00:00 13,933 行

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';
```



结果 消息

	tid	tname	email	salary
1	200026895	nuyoq	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);

select * from t_view;
```

插入的信息如下：

结果 消息

	tid	tname	email	salary
1	199999998	abc	abc@def.com	5000
2	199999999	abcd	abcd@efgh.com	6000
3	200003125	fqmmyi	wcjcg@glq.net	8856
4	200009423	jhoulvtr	12cj@pzc.gov	7572
5	200010493	xzgkflko	zcf_m@def.com	7664
6	200016731	nfgrod	dprcy@qxqi.com	8638
7	200017039	qibsy	b8mtpe@enbba.edu	9358
8	200023671	zfuxiahq	lr_32t@cfo.net	8354
9	200025513	rnupx	c7ce@ayt.org	6048
10	200026895	nuyoq	xibl@izd.edu	10660
11	200033112	awboicq	hjct6g@izgm.edu	8264
12	200047819	lmogygb	if095@dxdm.org	8936
13	200058185	xcoxbrz	dmjc@pcg.gov	16000

查询已成功执行。 LAPTOP-N3QO3T48 (15.0 RTM) wyj (52) School 00:00:00 13,934 行

4)在视图 t_view 中将编号为 200010493 的教师的薪水改为 6000。

```
update t_view set salary = 6000
where tid = '200010493';
select * from t_view where tid = '200010493';
```

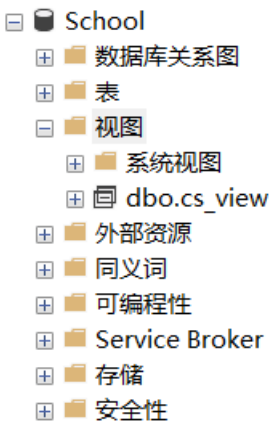
结果 消息

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

5)删除视图 t_view。

```
drop view t_view;
```

查看视图，发现视图 t_view 已经被删除：



3.3 用户标识与鉴别

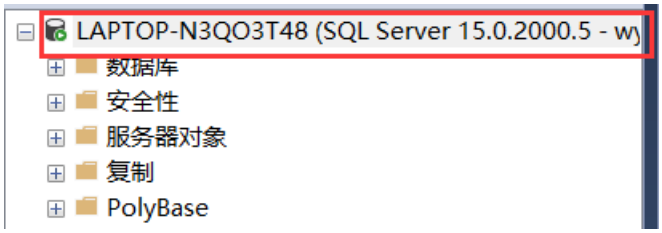
1 实验内容

由系统提供一定的方式让用户标识自己的名字和身份。

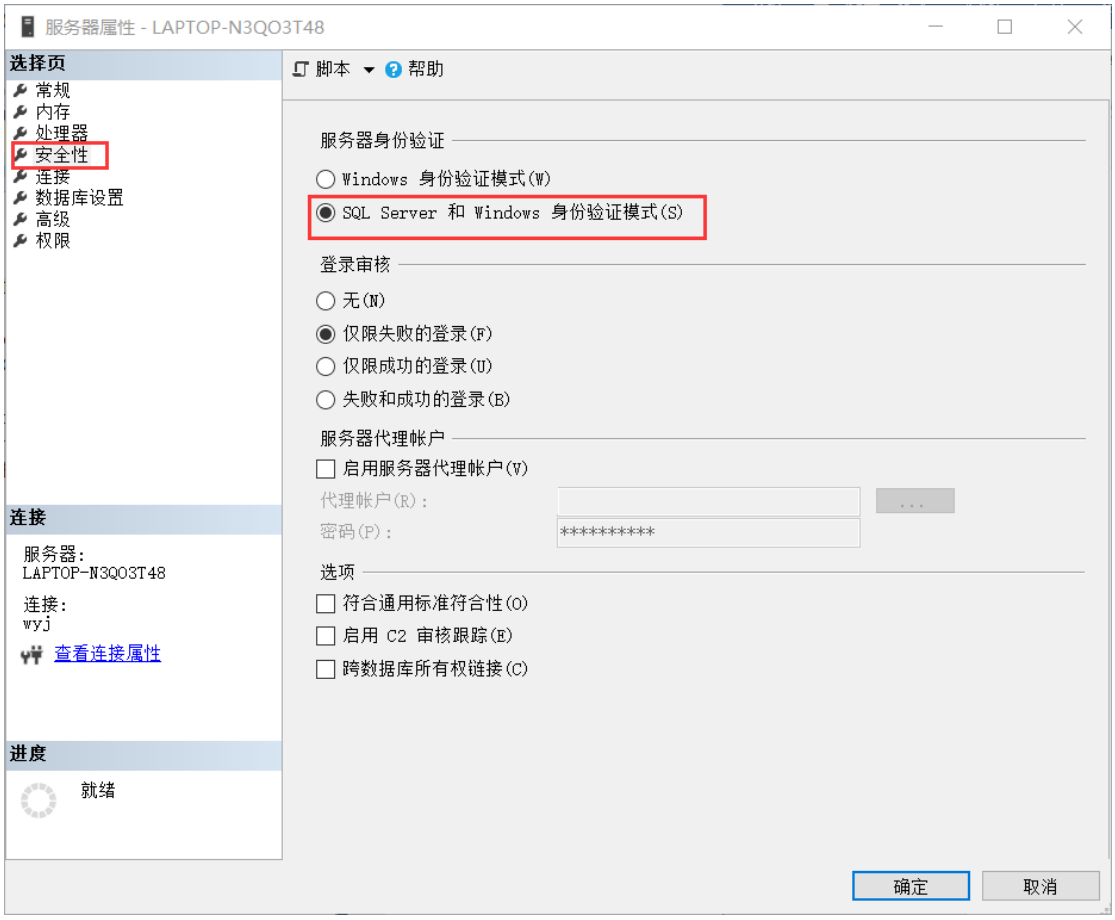
2 实验步骤

1)在 SSMS 中，设置 SQL Server 的安全认证模式。

在下图所示位置单击右键，点击属性。

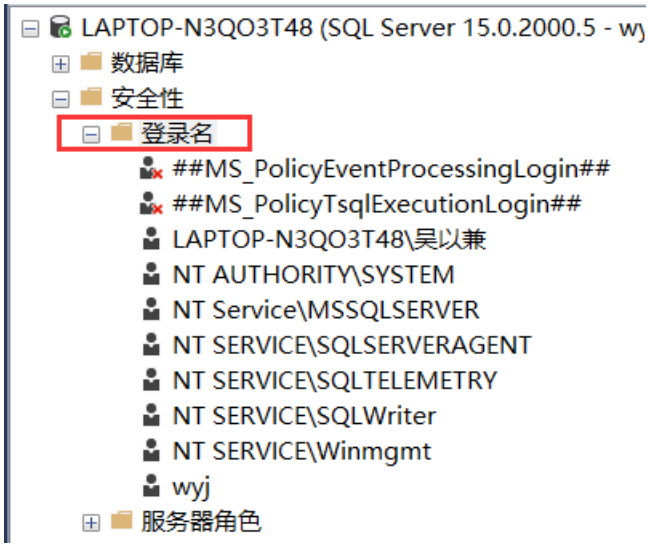


点击安全性，勾选 SQL-Server 和 Windows 身份验证模式。最后点击确定。

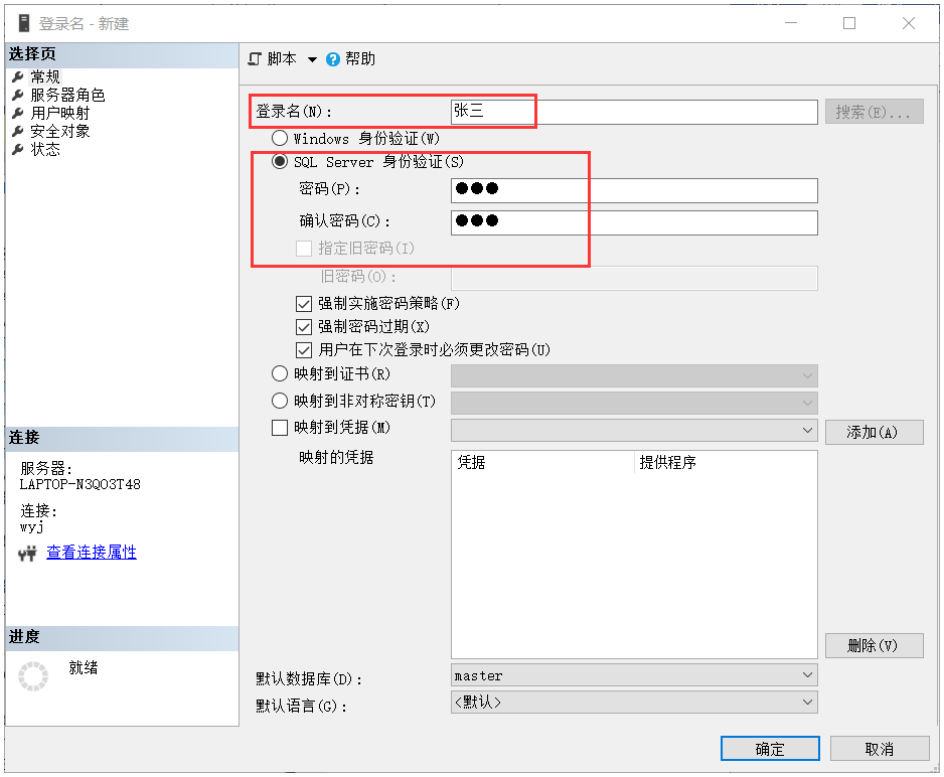


2)在 SSMS 中建立一个名为“张三”的登陆用户、School 数据库用户。

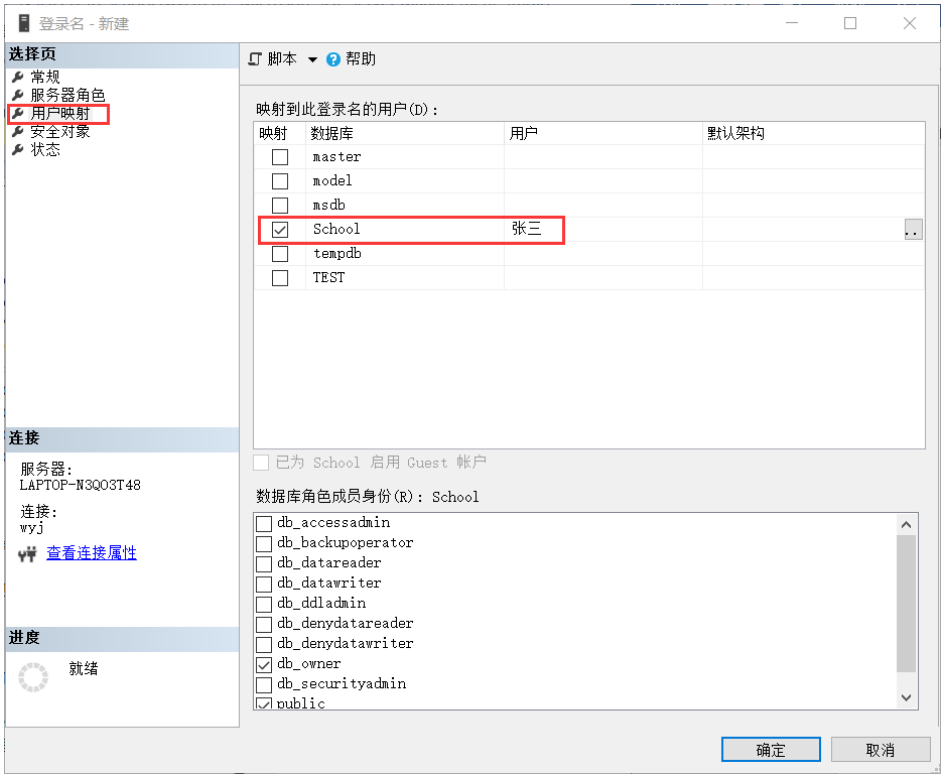
在安全性下，鼠标右键单击登录名，选择新建登录名。



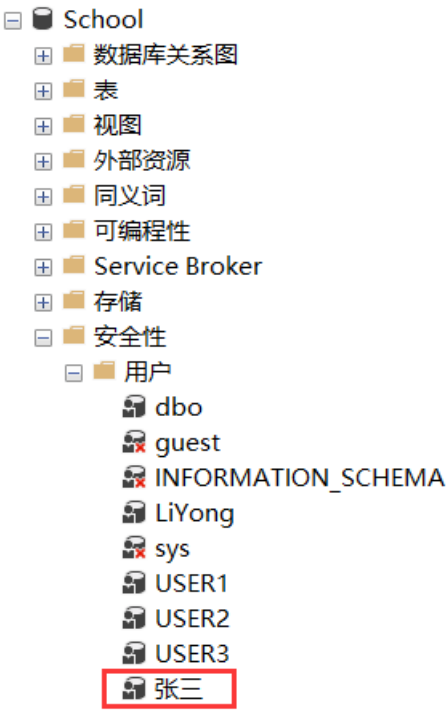
在跳出的页面中填写登录名‘张三’和密码。

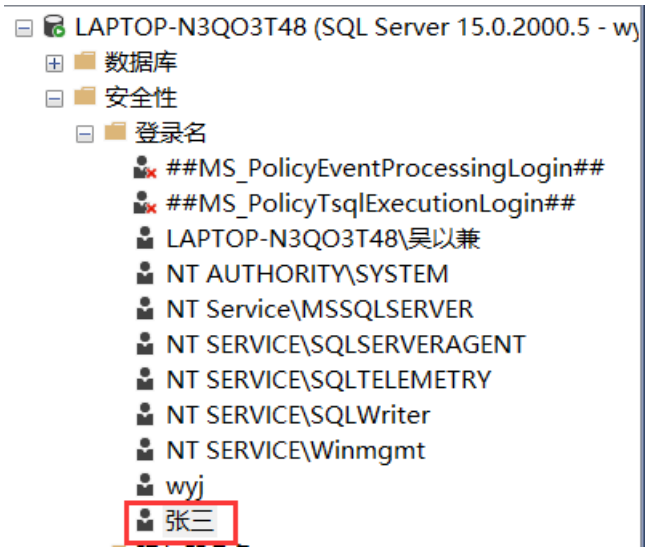


点击用户映射，勾选 School。点击确定。



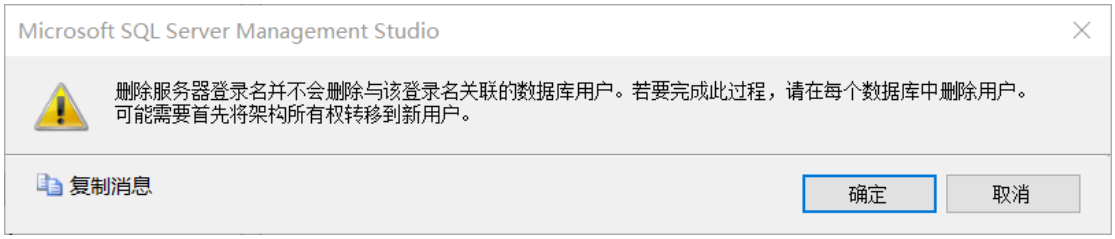
结果如下：

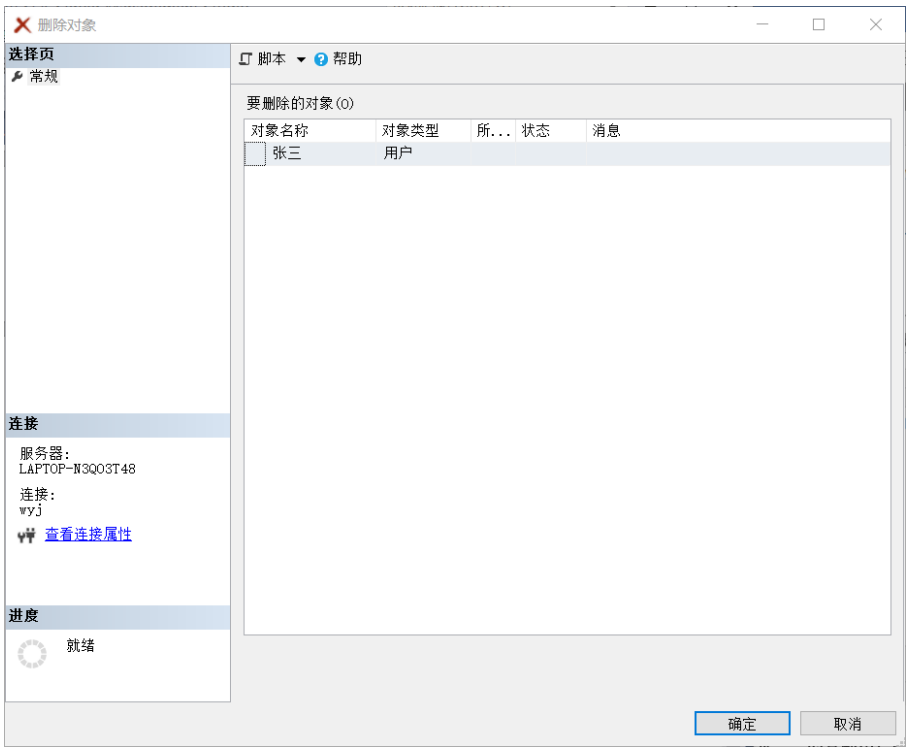




3)在 SSMS 中取消“张三”这个用户。

鼠标右键单击用户‘张三’，选择删除。根据提示消息，要在两个地方都将用户‘张三’删除。




删除对象
选择页
常规
脚本
帮助
要删除的对象(0)
对象名称
对象类型
所...
状态
消息
张三
用户
连接
服务器:
LAPTOP-N3QO3T48
连接:
wyj
查看连接属性
进度
就绪
确定
取消

删除后结果如下：

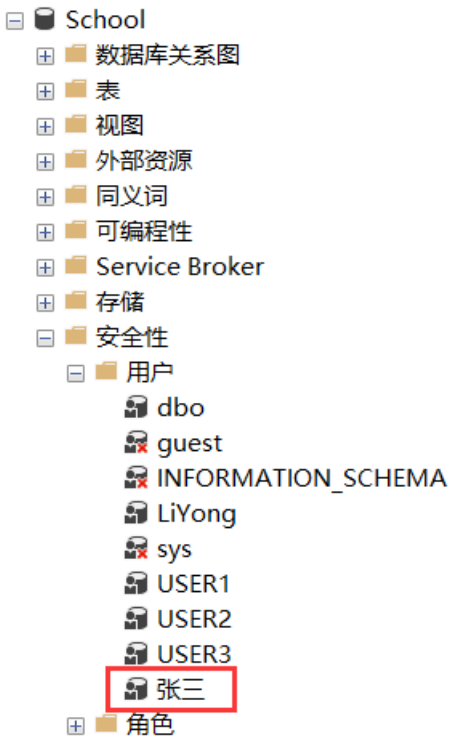
School
数据库关系图
表
视图
外部资源
同义词
可编程性
Service Broker
存储
安全性
用户
dbo
guest
INFORMATION_SCHEMA
LiYong
sys
USER1
USER2
USER3
角色

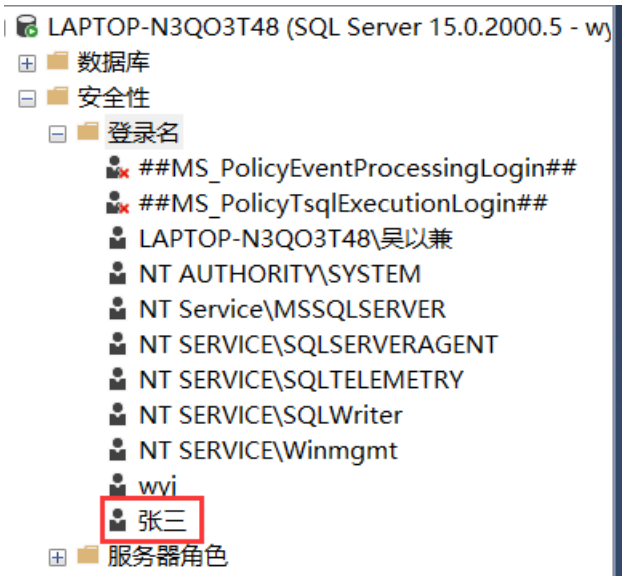
4)学习【参考命令】中的几个命令，了解利用命令实现上述步骤。

①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','111','School';
exec sp_adduser '张三';
```

结果如下：

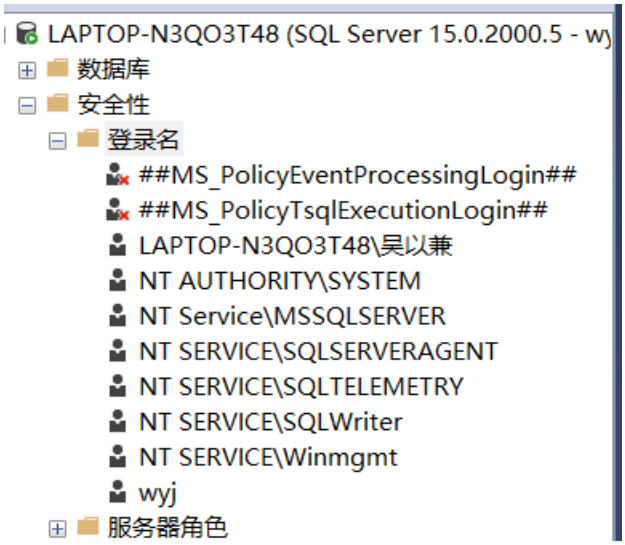


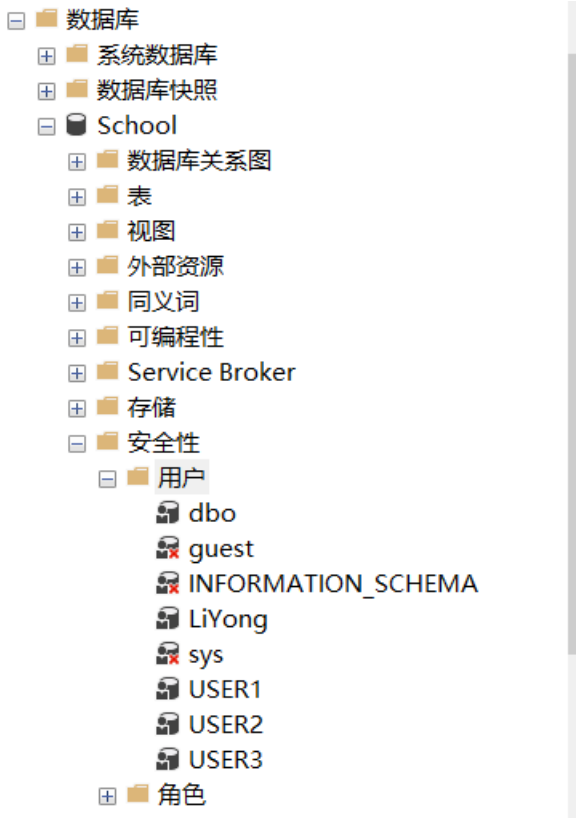


②在 SSMS 中取消“张三”这个用户。

```
exec sp_dropuser '张三';
exec sp_droplogin '张三';
```

结果如下：





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

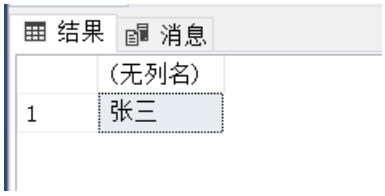
1） 在 SSMS 中建立一个名为“张三”的登陆用户、School 数据库的用户。参见实验 3.3 的试验步骤（2）

```
exec sp_addlogin '张三','111','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。

首先切换至用户‘张三’：

```
execute as user = '张三';
select user;
```



执行查询操作：

```
select * from STUDENTS;
```

结果如下：

```
select * from STUDENTS;
```

146 %

消息

消息 229，级别 14，状态 5，第 98 行
拒绝了对对象 'STUDENTS' (数据库 'School'，架构 'dbo')的 SELECT 权限。

完成时间：2022-04-21T15:06:04.3149092+08:00

所以“张三”这个用户名不具有对学生表的 SELECT 权限。

3） 将 School 数据库的操作权限赋予数据库用户张三。

```
revert select user;
grant all on STUDENTS to 张三;
grant all privileges on CHOICES to 张三;
grant all privileges on TEACHERS to 张三;
grant all privileges on COURSES to 张三;
execute as user = '张三';
select * from STUDENTS;
```

结果 消息

	sid	sname	email	grade
1	800001216	gfxrgs	hhce4@qhldj.gov	1992
2	800002933	vnbqzsvv	pvhxd4l@zqur.org	2002
3	800005753	waqcj	hlhq0h8@jdba.gov	1992
4	800006682	fiiluommh	ihzd6_k@kzvft.gov	1992
5	800006941	ogvmu	62sfbd@lrt.gov	1995
6	800007595	uxqqbkjn	cr8g@zrvgt.edu	1997
7	800008565	ehlycg	nach10@uic.com	1999
8	800009026	rcxaihj	4ul4kqb@hko.edu	2002
9	800009099	zapyv	jqmqn8@iwaiu.org	1992
10	800009249	zyuoh	8enjrcu@upfw.org	1991
11	800010666	uwphrw	emb7k@ipp.com	1992
12	800013889	nahhluoe	w6org6@maq.com	2000

查询已成功执行。 LAPTOP-N3QO3T48 (15.0 RTM) wyj (63) School 00:00:00 100,000 行

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','111','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）。

```
create view ch_view as
select * from CHOICES where cid = '10005'
with check option;

select * from ch_view;
```

结果 消息

	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
11	500156392	890918686	214209928	10005	NULL
12	500158288	861996649	268142256	10005	95
13	500216347	816364126	248477977	10005	82

查询已成功执行。 LAPTOP-N3QO3T48 (15.0 RTM) wyj (59) School 00:00:00 6,031 行

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;
```

结果 消息

	score
1	90
2	91
3	82
4	83
5	74
6	79
7	73
8	68
9	81
10	NULL
11	NULL
12	95

查询已成功执行。 LAPTOP-N3QO3T48 (15.0 RTM) wyj (59) School 00:00:00 6,031 行

5)以用户张三登陆查询分析器，对 ch_view 进行查询操作。

首先将对 ch_view 视图的权限赋给用户张三：

```
revert select SYSTEM_USER;
grant all on ch_view to 张三;
```

然后切换到用户张三，对 ch_view 进行查询操作：

```
execute as user = '张三';
select * from ch_view;
```

结果 消息

	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
11	500156392	890918686	214209928	10005	NULL
12	500158288	861996649	268142256	10005	95
13	500216347	816364126	248477977	10005	82

查询已成功执行。 LAPTOP-N3QO3T48 (15.0 RTM) wyj (59) School 00:00:00 6,031 行

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。

以用户张三登陆查询分析器：

```
execute as user = '张三';
select user;
```

结果 消息

	(无列名)
1	张三

查询修改前 no 为 500127998 的学生的成绩：

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

结果 消息

	score
1	81

修改为 90 分：

```
update ch_view set score = 90 where no=500127998;
select score from ch_view where no=500127998;
```

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

收回用户张三对视图 ch_view 的查询权限：

```
revert select SYSTEM_USER;
revoke select on ch_view from 张三;
```

切换到用户张三，对视图 ch_view 进行查询：

结果 消息

	(无列名)
1	张三

```
select * from ch_view;
```

146 %

消息

消息 229，级别 14，状态 5，第 33 行
拒绝了对对象 'ch_view' (数据库 'School'，架构 'dbo')的 SELECT 权限。

完成时间：2022-04-21T15:43:29.5314237+08:00

实验总结：

通过本次实验，对视图的建立、更新、删除操作更加熟练了。在本次实验中用到了不同用户的切换，使用了创建登录名和用户的 exec 语句。要注意的是创建新用户和对用户权限的赋值需要在管理员模式下进行，否则会导致失败。可以使用 revert 来切换到管理员用户。完成实验后，对数据库不同用户的切换和权限操作有了更具象的理解。