

实验八 数据库安全性

实验目的

熟悉数据库用户管理和权限管理，能够使用SQL语句来向用户授予和收回权限。

实验内容

- 1. 使用GRANT语句对用户授权，对单个用户和多个用户授权，或使用保留字PUBLIC对所有用户授权。对不同的操作对象包括数据库、视图、基本表等进行不同权限的授权。
- 2. 使用WITH GRANT OPTION子句授予用户传播该权限的权利。
- 3. 在授权时发生循环授权，考察DBS能否发现这个错误。如果不能，结合取消权限操作，查看DBS对循环授权的控制。
- 4. 使用REVOKE子句收回授权，取消授权的级联反应。

课内实验

要求：

在数据库school中建立三个用户USER1,USER2和USER3,它们在数据库中的角色是PUBLIC。请按以下要求，分别以管理员身份或这三个用户的身份登录到数据库中，进行操作。

> Subscriptions

> school(new)

✓ Login/Group Roles (18)

pg_checkpoint

pg_create_subscription

pg_database_owner

pg_execute_server_program

pg_monitor

pg_read_all_data

pg_read_all_settings

pg_read_all_stats

pg_read_server_files

pg_signal_backend

pg_stat_scan_tables

pg_use_reserved_connections

pg_write_all_data

pg_write_server_files

postgres

user1

user2

user3

1 CREATE ROLE USER1 WITH LOGIN PASSWORD '123456';

2 CREATE ROLE USER2 WITH LOGIN PASSWORD '123456';

3 CREATE ROLE USER3 WITH LOGIN PASSWORD '123456';

Data Output

Messages

Notifications

CREATE ROLE

Query returned successfully in 44 msec.

我刚开始并没有设密码，就会在转换用户时出现问题，所以必须要设置密码

1. 授予所有用户对表COURSES的查询权限。

Query

Query History

1

GRANT SELECT ON COURSES TO PUBLIC;

Data Output

Messages

Notifications

GRANT

Query returned successfully in 71 msec.

2. 授予USER1对表STUDENTS插入和更新的权限，但不授予删除权限，并且授予USER1传播这两个权限的权利。

Query

Query History

1

GRANT INSERT, UPDATE ON STUDENTS TO USER1

2

WITH GRANT OPTION;

Data Output

Messages

Notifications

GRANT

Query returned successfully in 68 msec.

3. 允许USER2在表CHOICE中插入元组，更新的SCORE列，可以选取除了SID以外的所有列。

Query

Query History

1

GRANT INSERT, UPDATE(score) ON TABLE CHOICES TO USER2;

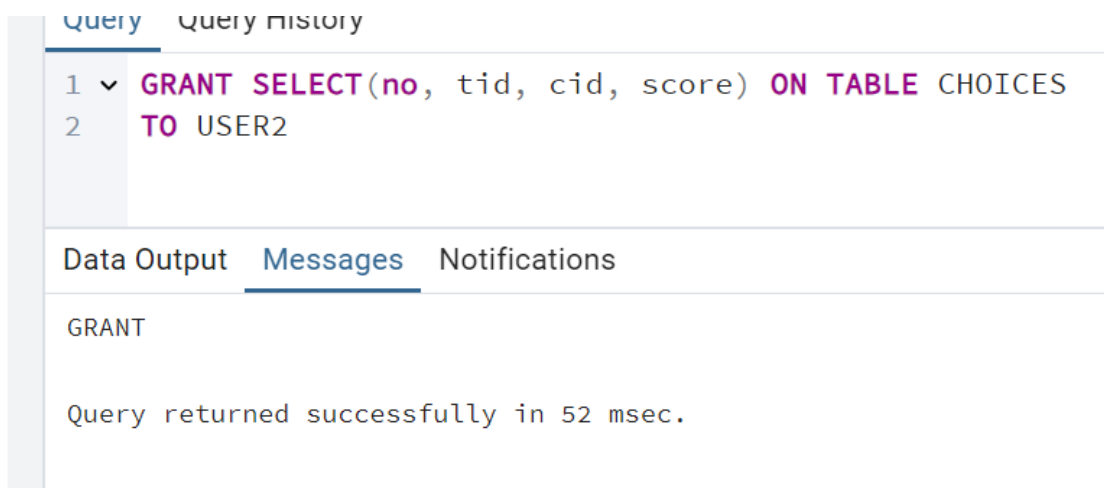
Data Output

Messages

Notifications

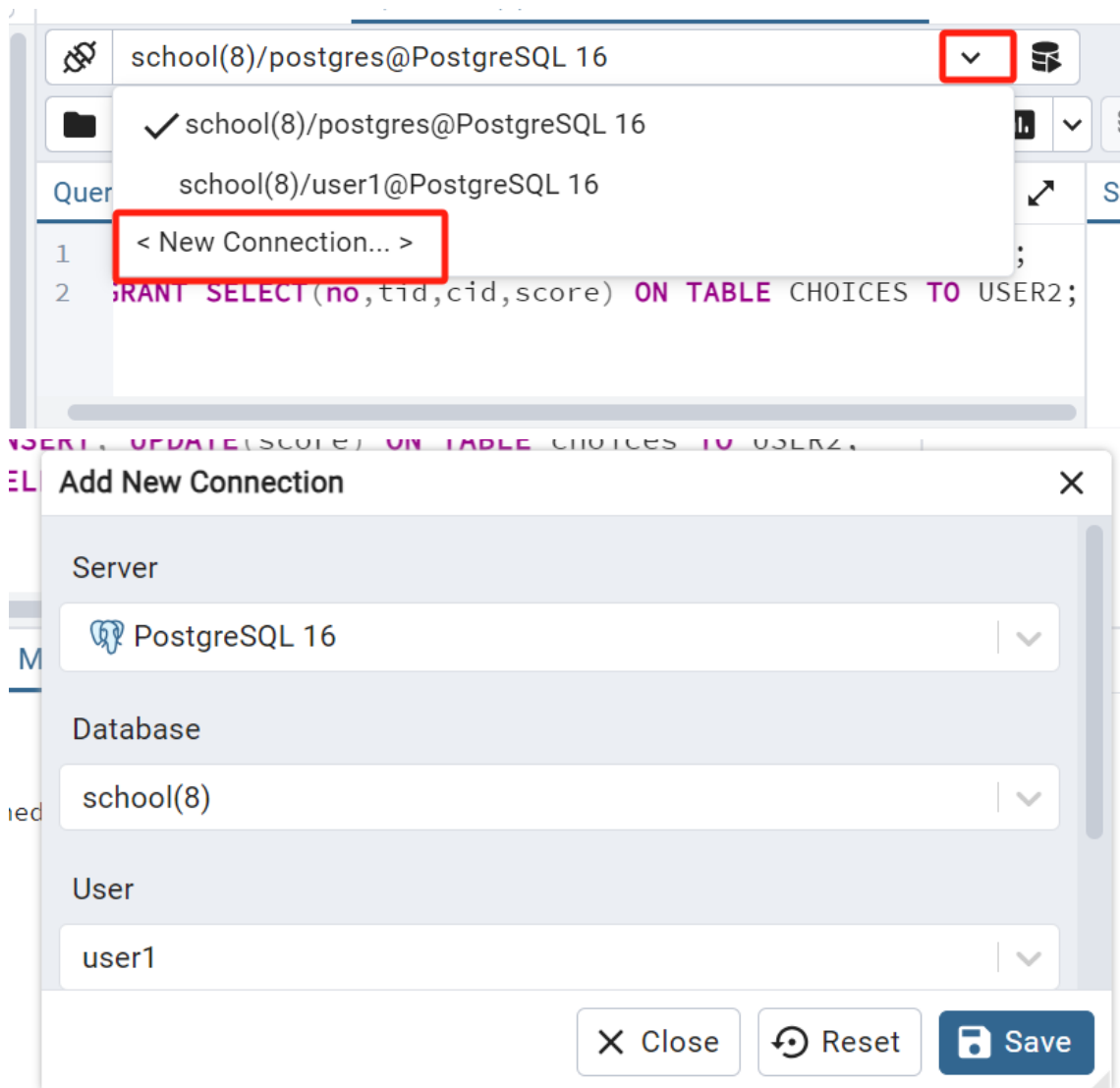
GRANT

Query returned successfully in 42 msec.

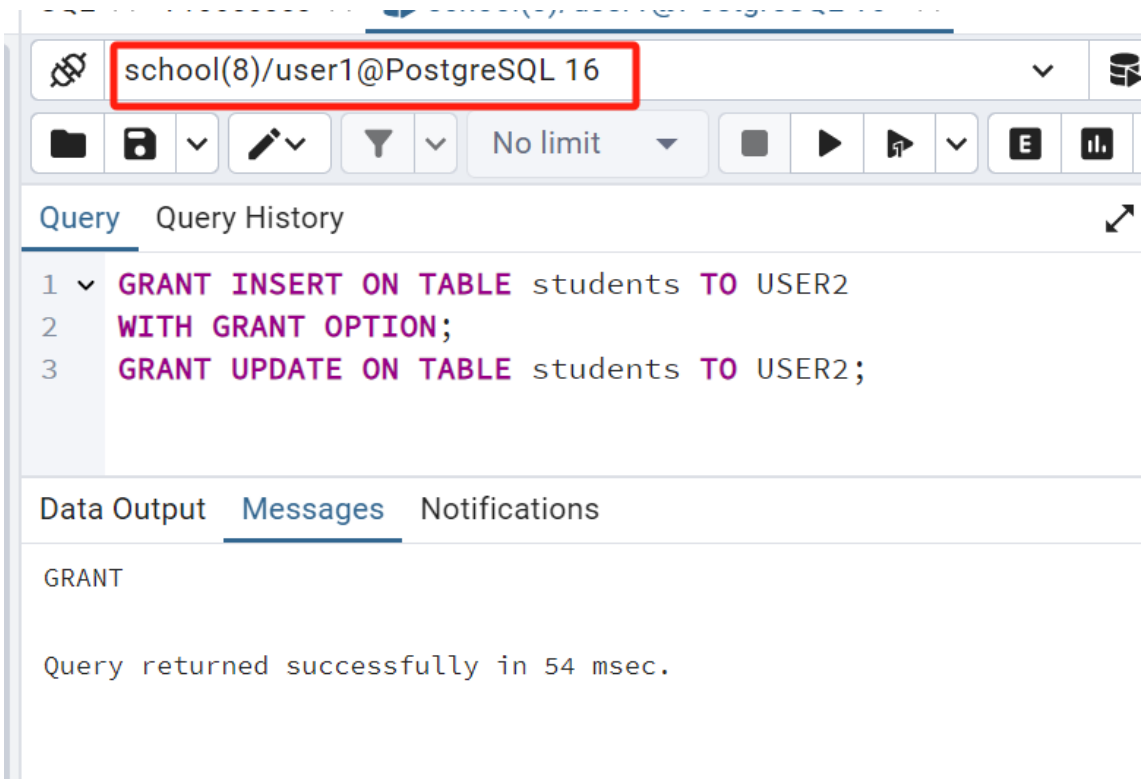


4. USER1授予USER2对表STUDENTS插入和更新的权限，并且授予USER2传播插入操作的权利。

我们先将用户转变过来，点击小三角



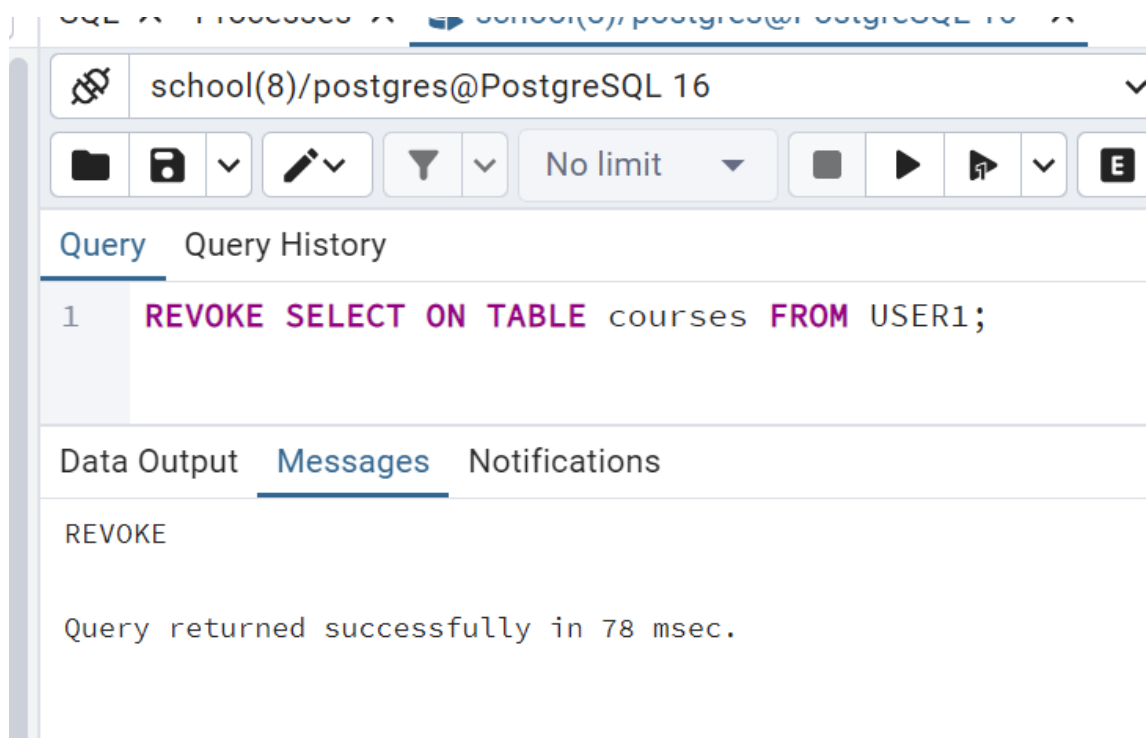
输入密码后就可以转换过来了，如下图。（这里如果不设置密码将无法登录）



也就能用USER1授予USER2的权限了

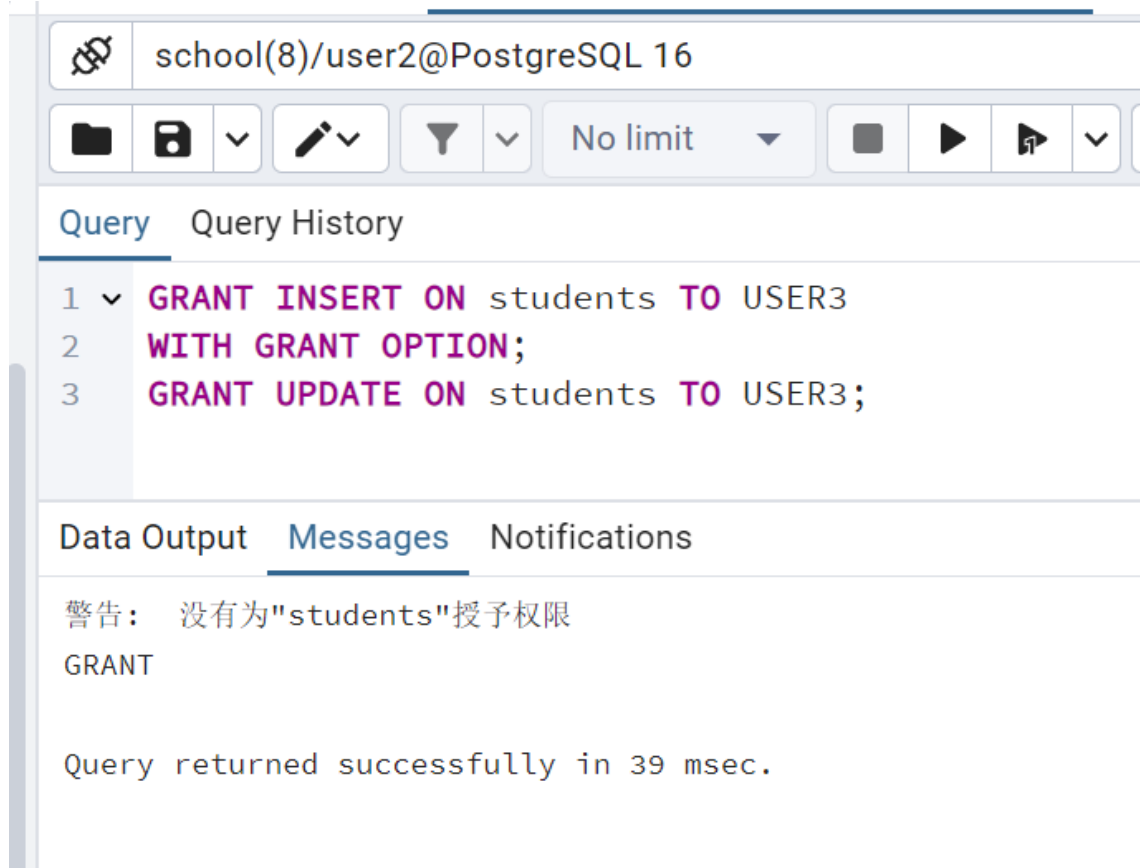
5. 收回对USER1对表COURSES查询权限的授权。

用户切换回管理员，收回USER1的权限



6. 由上面2. 和4. 的授权，再由USER2对USER3授予表STUDENTS插入和更新的权限，并且授予USER3传播插入操作的权利。这时候，如果由USER3对USER1授予表STUDENTS的插入和更新权限是否能得到成功?如果能够成功，那么如果由USER2取消USER3的权限，对USER1会有什么影响?如果再由DBA取消USER1的权限，对USER2有什么影响?

同样的，先将身份切换为USER2,对USER3进行授权



The screenshot shows a PostgreSQL client interface. At the top, the connection string is 'school(8)/user2@PostgreSQL 16'. Below the connection bar, there are icons for file operations and a 'No limit' dropdown. The main area is divided into 'Query' and 'Query History' tabs. The 'Query' tab is active, showing a SQL statement with line numbers 1, 2, and 3. The statement is:
1 GRANT INSERT ON students TO USER3
2 WITH GRANT OPTION;
3 GRANT UPDATE ON students TO USER3;
Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is active, displaying a warning message: '警告: 没有为"students"授予权限 GRANT'. Below the warning, it says 'Query returned successfully in 39 msec.'

经过检查，发现出现警告的原因是因为在第四步实验中，USER1只为USER2授予传播插入操作的权利，因此USER2在为USER3授予UPDATE操作时就会警告，此时也就说明USER2不存在传播更新操作的权力。

SQL X Processes X school(8)/user3@PostgreSQL 16

school(8)/user3@PostgreSQL 16

Folder Save Filter No limit

Query Query History

```
1 GRANT INSERT ON students TO USER1;
```

Data Output Messages Notifications

GRANT

Query returned successfully in 36 msec.

school(8)/user3@PostgreSQL 16

Folder Save Filter No limit

Query Query History

```
1 GRANT UPDATE ON students TO USER1;
```

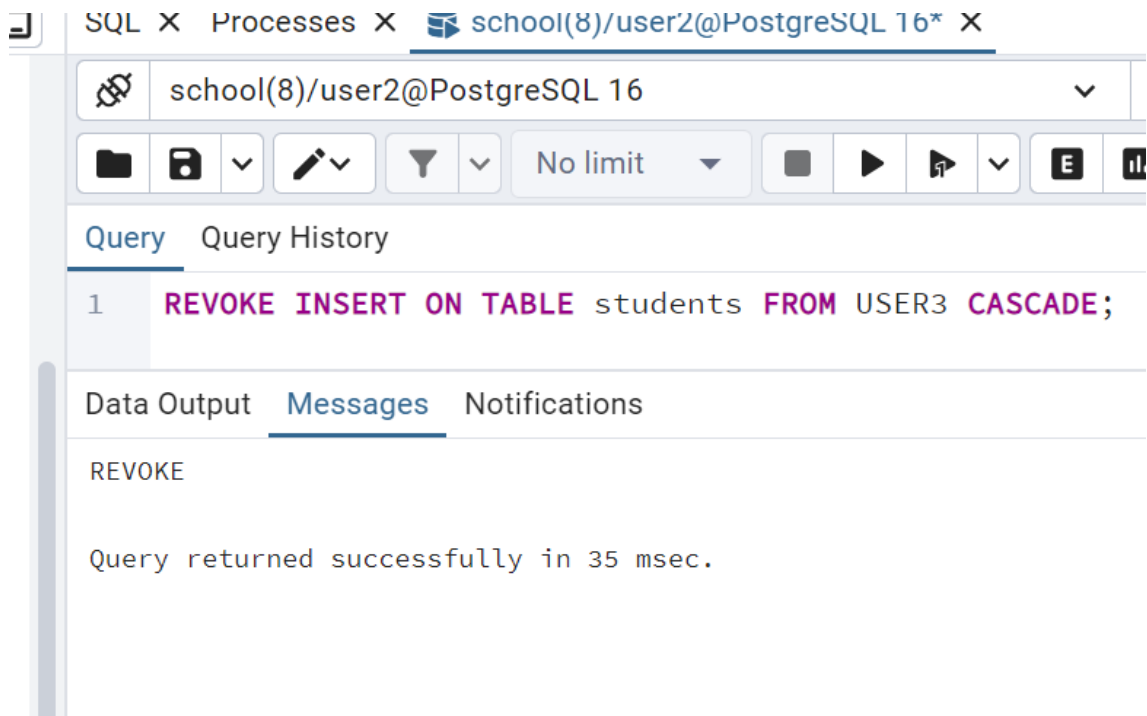
Data Output Messages Notifications

警告： 没有为"students"授予权限
GRANT

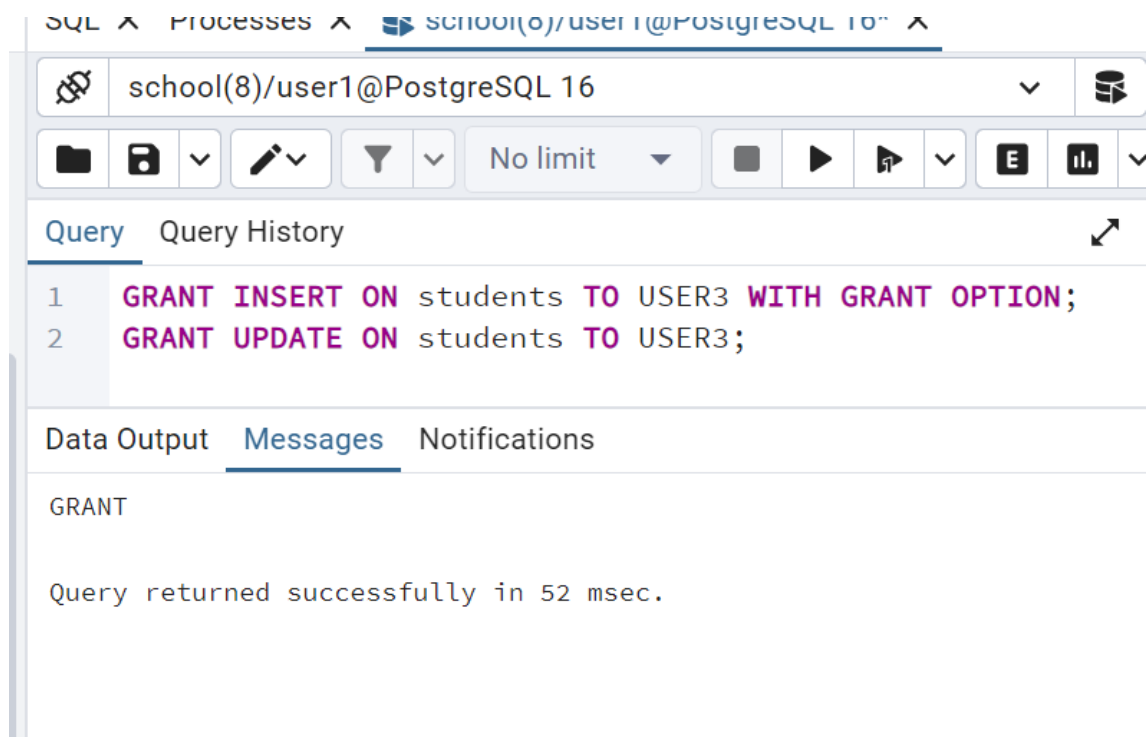
Query returned successfully in 110 msec.

因此，USER3只能向USER1授予插入功能，而无法授予更新功能


而通过USER2删除USER3的权限时，必须使用CASCADE，因为此时USER3已经将权限赋予USER1，有绑定。








这时候我们使用USER1对USER3传播权限，可以看到能够进行传播，说明USER2对USER3权限的删除不会影响USER1，因为USER1本身就被赋予了权限






但如果由DBA收回USER1的权限，USER2也就没有这些权限了，因为它的权限依赖于USER1





 school(8)/postgres@PostgreSQL 16










No limit





Query Query History

1


2




REVOKE INSERT, UPDATE ON TABLE students FROM USER1
CASCADE;



Data Output Messages Notifications



REVOKE


Query returned successfully in 49 msec.





 school(8)/user2@PostgreSQL 16









No limit





Query Query History

1

GRANT INSERT, UPDATE ON TABLE students TO USER1

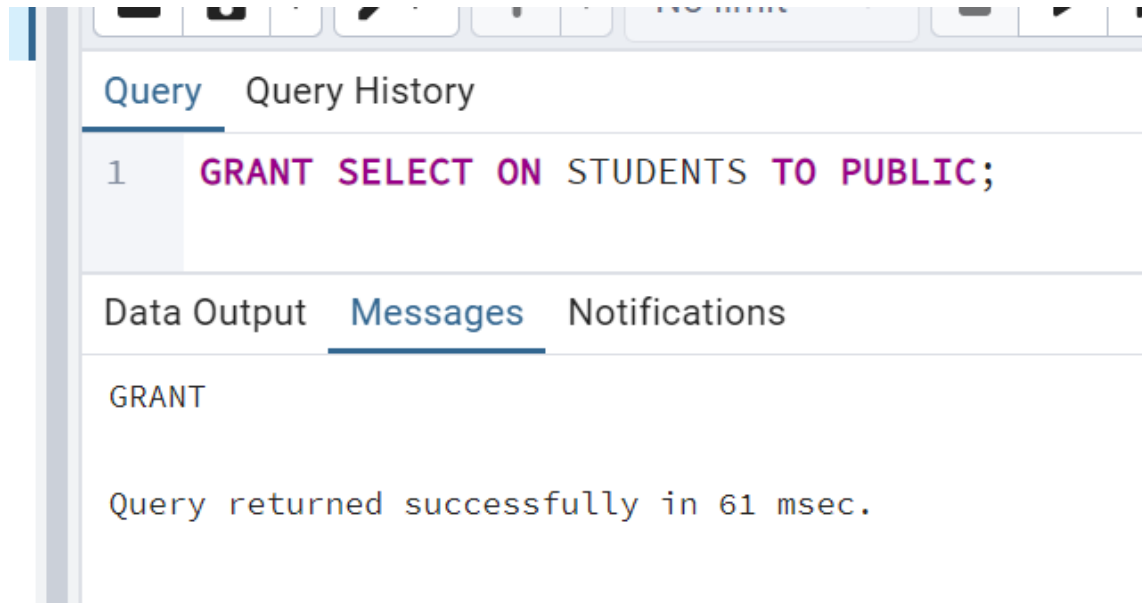
Data Output Messages Notifications

ERROR: 对表 students 权限不够

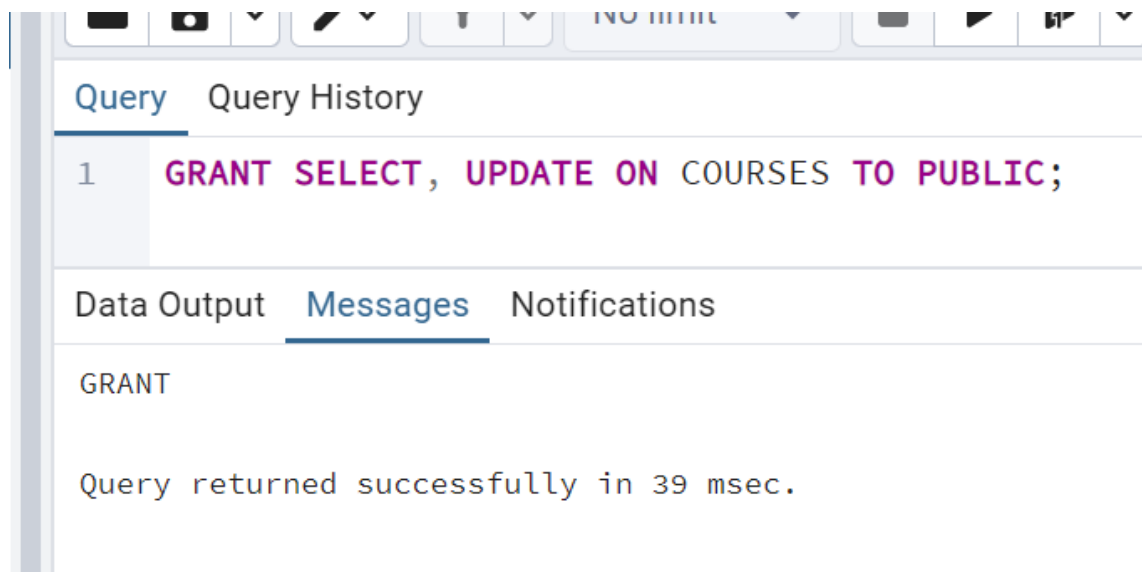
错误: 对表 students 权限不够
SQL state: 42501

自我实践

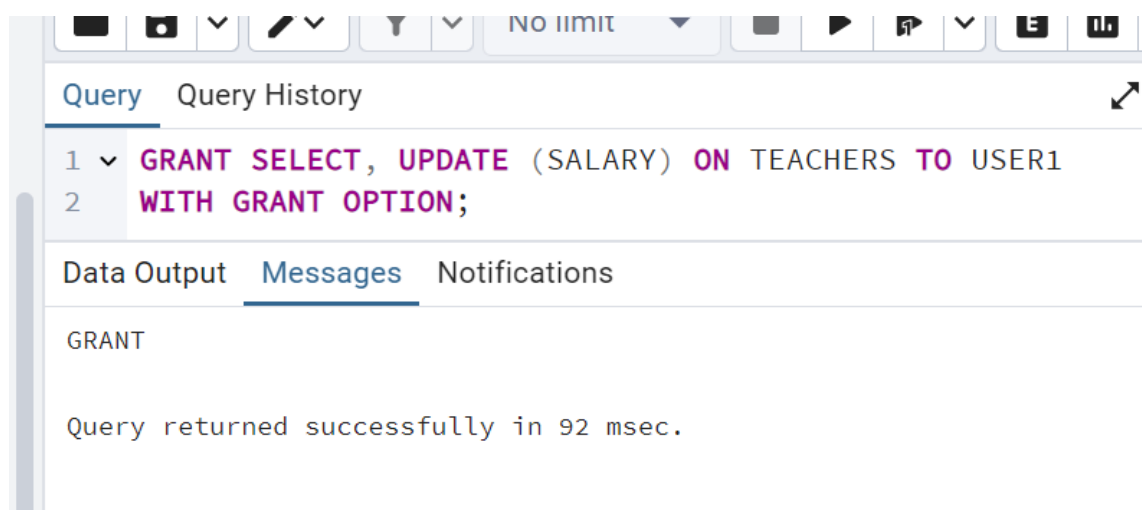
1. 授予所有用户对表STUDENTS的查询权限。



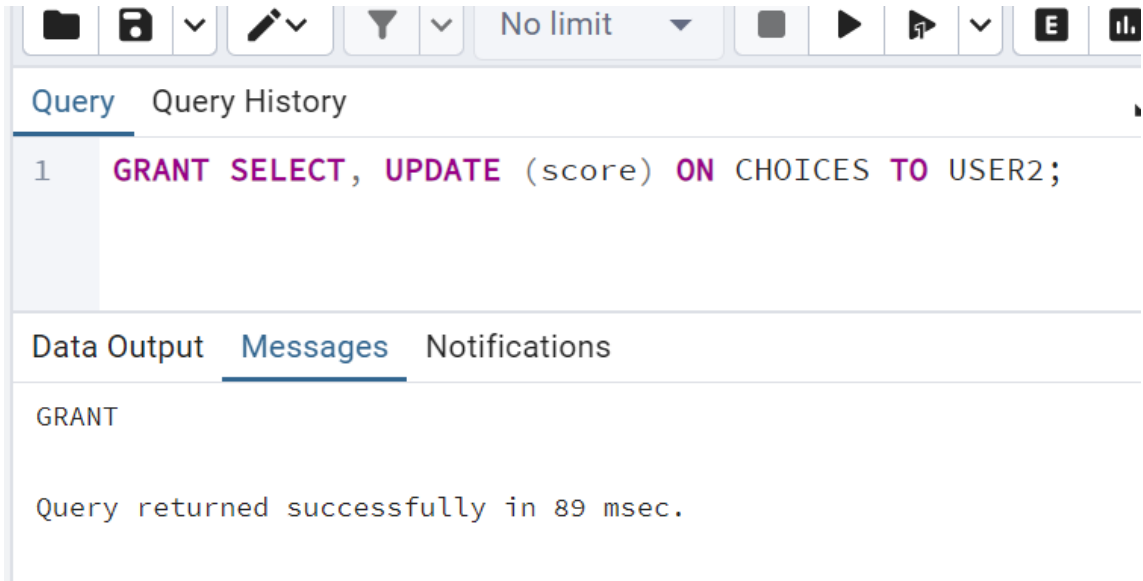
2. 授予所有用户对表COURSES的查询和更新权限。



3. 授予USER1对表TEACHERS的查询，更新工资的权限，且允许USER1可以传播这些权限。

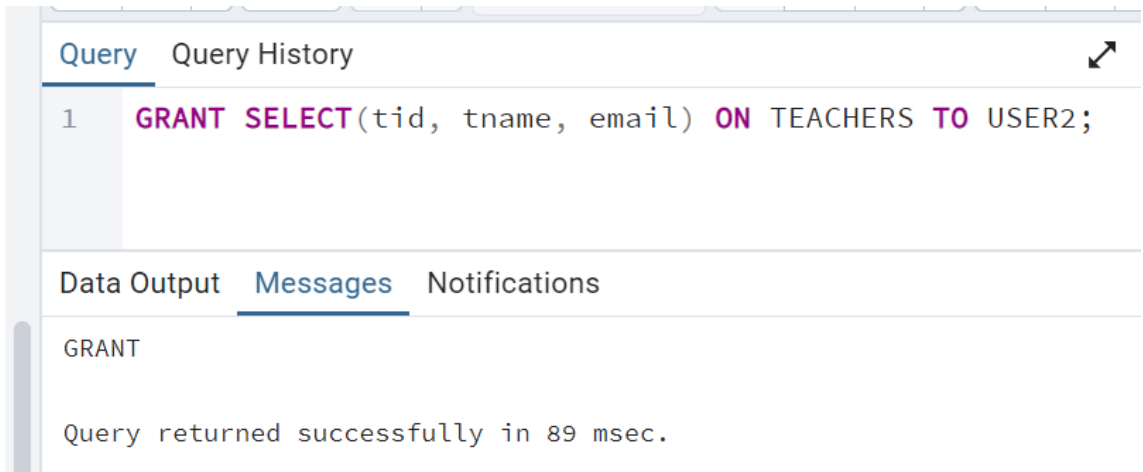


4. 授予USER2对表CHOICES的查询，更新成绩的权限。



The screenshot shows a SQL IDE window with a toolbar at the top containing icons for file operations, editing, and execution. Below the toolbar, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying a single SQL statement: `1 GRANT SELECT, UPDATE (score) ON CHOICES TO USER2;`. Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is active, showing the output: `GRANT` and `Query returned successfully in 89 msec.`

5. 授予USER2对表TEACHERS的除了工资之外的所有信息的查询。



The screenshot shows a SQL IDE window with a toolbar at the top. Below the toolbar, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying a single SQL statement: `1 GRANT SELECT(tid, tname, email) ON TEACHERS TO USER2;`. Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is active, showing the output: `GRANT` and `Query returned successfully in 89 msec.`

6. 由USER1授予USER2对表TEACHERS的查询权限和传播的此项权限的权利。

和课内实验一样，先切换用户，再执行命令

SQL X Processes X school(8)/user1@PostgreSQL 16* X

school(8)/user1@PostgreSQL 16

Query Query History

```
1 GRANT SELECT ON TEACHERS TO USER2
2 WITH GRANT OPTION;
```

Data Output Messages Notifications

GRANT

Query returned successfully in 104 msec.

7. 由USER2授予USER3对表TEACHERS的查询权限，和传播的此项权限的权利。再由USER3授予USER2上述权限，这样的SQL语句能否成功得到执行？

USER2->USER3

SQL ^ PROCESSES ^ school(8)/user2@PostgreSQL 16

school(8)/user2@PostgreSQL 16

Query Query History

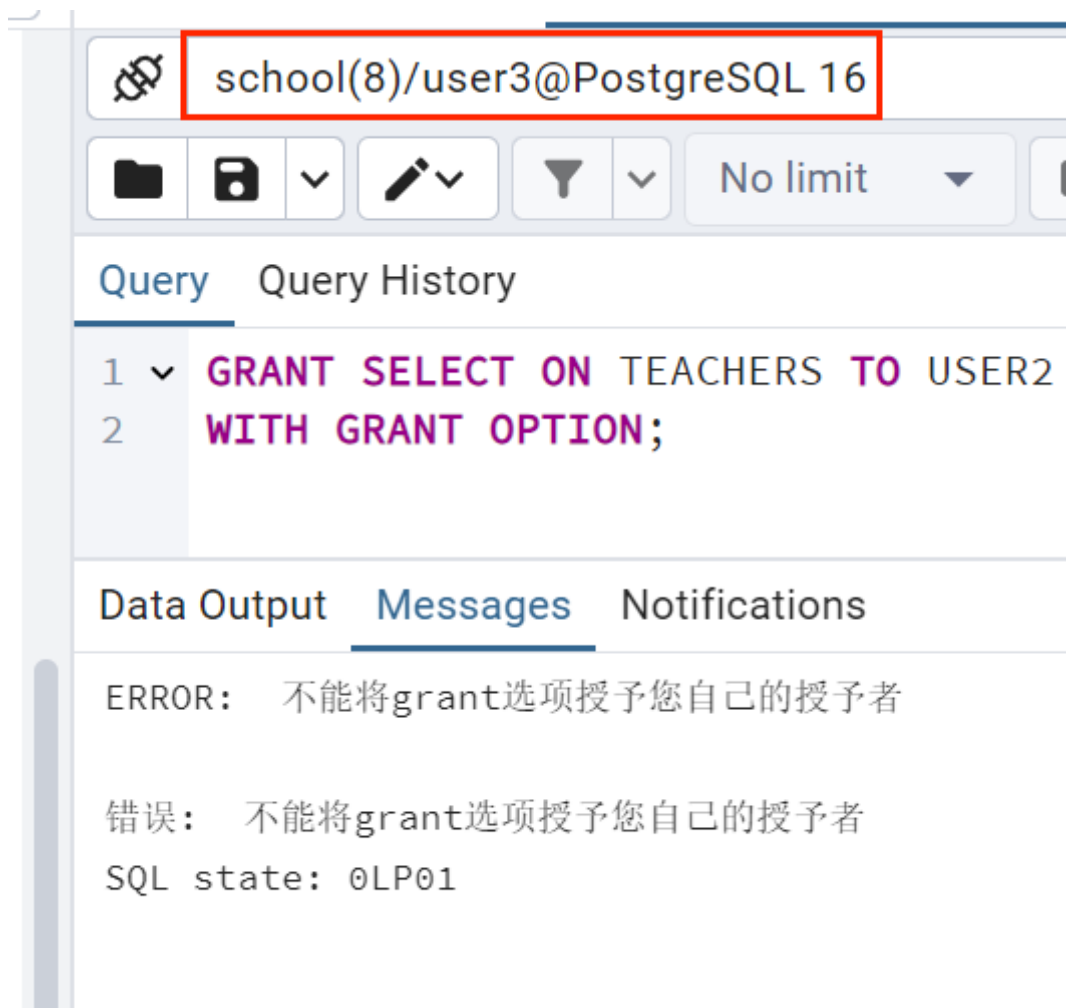
```
1 GRANT SELECT ON TEACHERS TO USER3
2 WITH GRANT OPTION;
```

Data Output Messages Notifications

GRANT

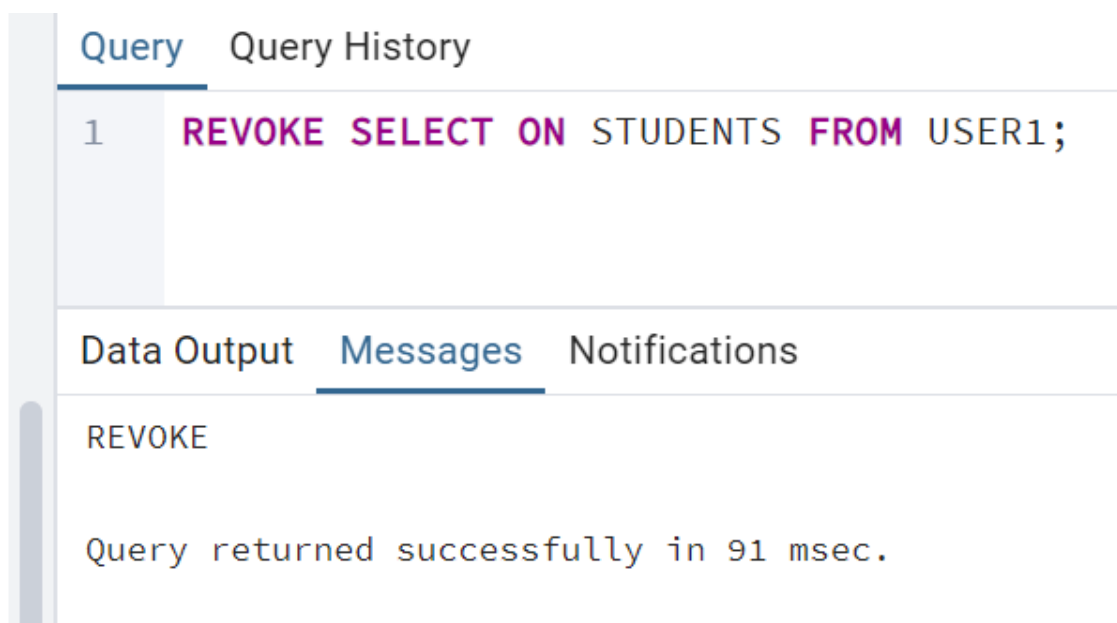
Query returned successfully in 45 msec.

USER3->USER2



可以看到该操作无法正常完成

8. 取消USER1对表STUDENTS的查询权限，考虑由USER2的身份对表STUDENTS进行查询，操作能否成功?为什么?



school(8)/user2@PostgreSQL 16

Query Query History Scratch P

```

1 SELECT *
2 FROM STUDENTS

```

Data Output Messages Notifications

	sid [PK] character (9)	sname character varying (30)	email character varying (30)	grade integer
1	800001216	gfxrgs	hhce4@qhldj.gov	1992
2	800002933	vnbqz%svv	pvhxd4l@zqur.org	2002
3	800005753	waqcj	hlhq0h8@jdba.gov	1992
4	800006682	fiiluommh	ihzd6_k@kzvft.gov	1992
5	800006941	ogvmu	62sfbdlrt.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	jmqn8@iwaiu.org	1992
10	800009249	zyuoh	8enjrcu@upfw.org	1991
11	800010666	uwphrw	emb7k@ipp.com	1992
12	800013889	nahhluoe	w6org6@maq.com	2000
13	800014004	aoaahudi	ftl0oci@fits.edu	1994
14	800014678	fnvxgrisa	nikk@ccchc.com	1996

可以看出能成功。因为第一步实验中是管理员对所有用户授予可查询STUDENTS的权限，用户之间并不互相依赖，所以即使取消了USER1的权限，USER2和USER3的权限依然保留。

9. 取消USER1和USER2的关于表COURSES的权限。

The screenshot shows a PostgreSQL query editor window. The title bar indicates the connection is to 'school(8)/postgres@PostgreSQL 16'. The interface includes a toolbar with icons for file operations, a filter icon, and a 'No limit' dropdown. Below the toolbar, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying two lines of SQL code: '1 REVOKE ALL PRIVILEGES ON COURSES FROM USER1;' and '2 REVOKE ALL PRIVILEGES ON COURSES FROM USER2;'. Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is active, showing the output 'REVOKE' and a status message 'Query returned successfully in 105 msec.'

school(8)/postgres@PostgreSQL 16

Query Query History

```
1 REVOKE ALL PRIVILEGES ON COURSES FROM USER1;
2 REVOKE ALL PRIVILEGES ON COURSES FROM USER2;
```

Data Output Messages Notifications

REVOKE

Query returned successfully in 105 msec.

注意：以上各题目，若无特别指明，均指由表的所有者授权或取消授权。