

# 安全综合案例实践

## 实验目的

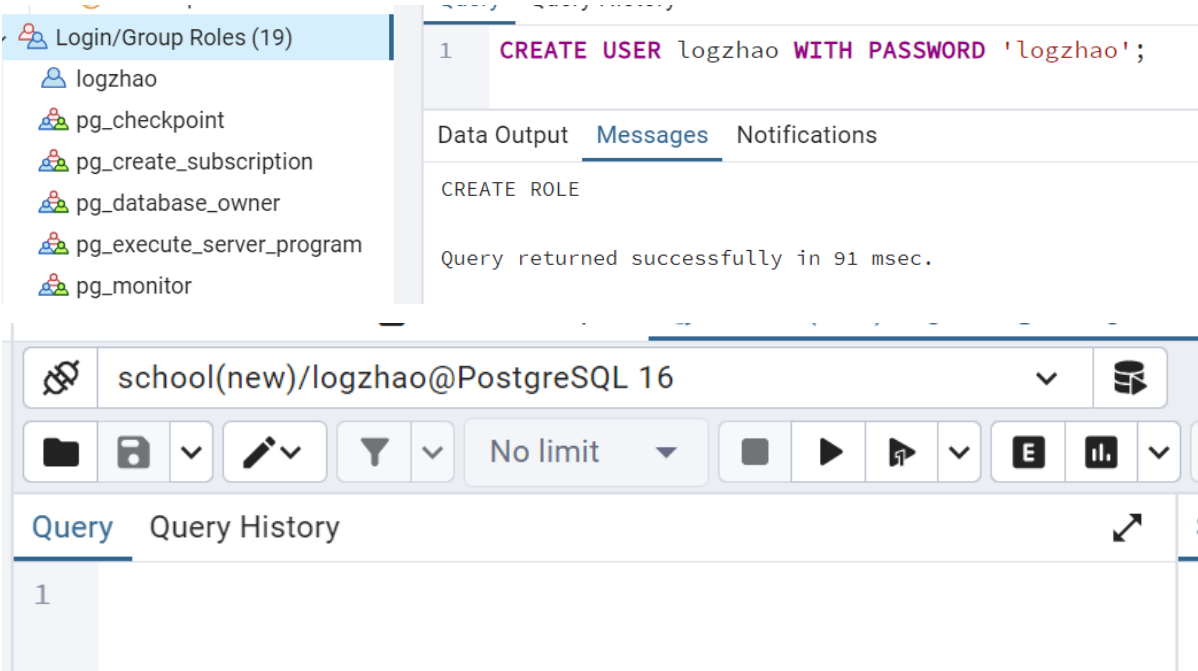
通过完成一个综合案例的实验，加深对数据库安全性控制的理解。

## 课内实验（遇到的问题用灰色字体表示）

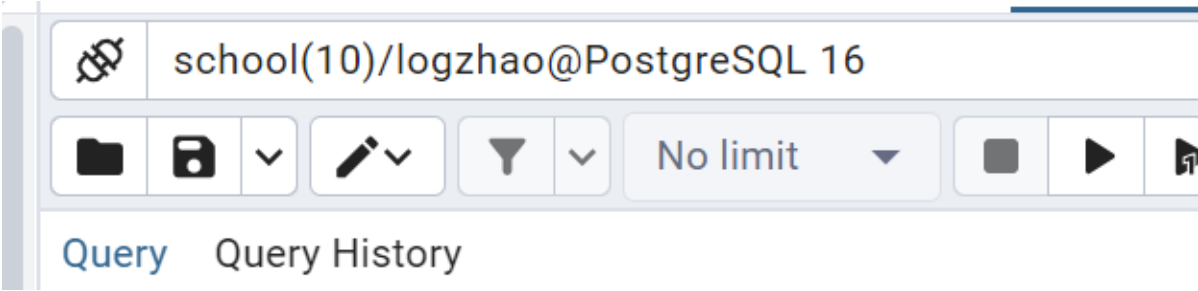
问题：赵老师当了2008级电子商务班的班主任，他要能查到全校的课程信息以及本班学生的选课信息，如何让他有权查到这些信息? 主要内容如下：

### 1. 登录管理

为新老师创建登录账号logzhao,验证该账号与数据库的连接访问是否正确？



我们看到可以登录此用户，说明该账号可以正常连接访问数据库



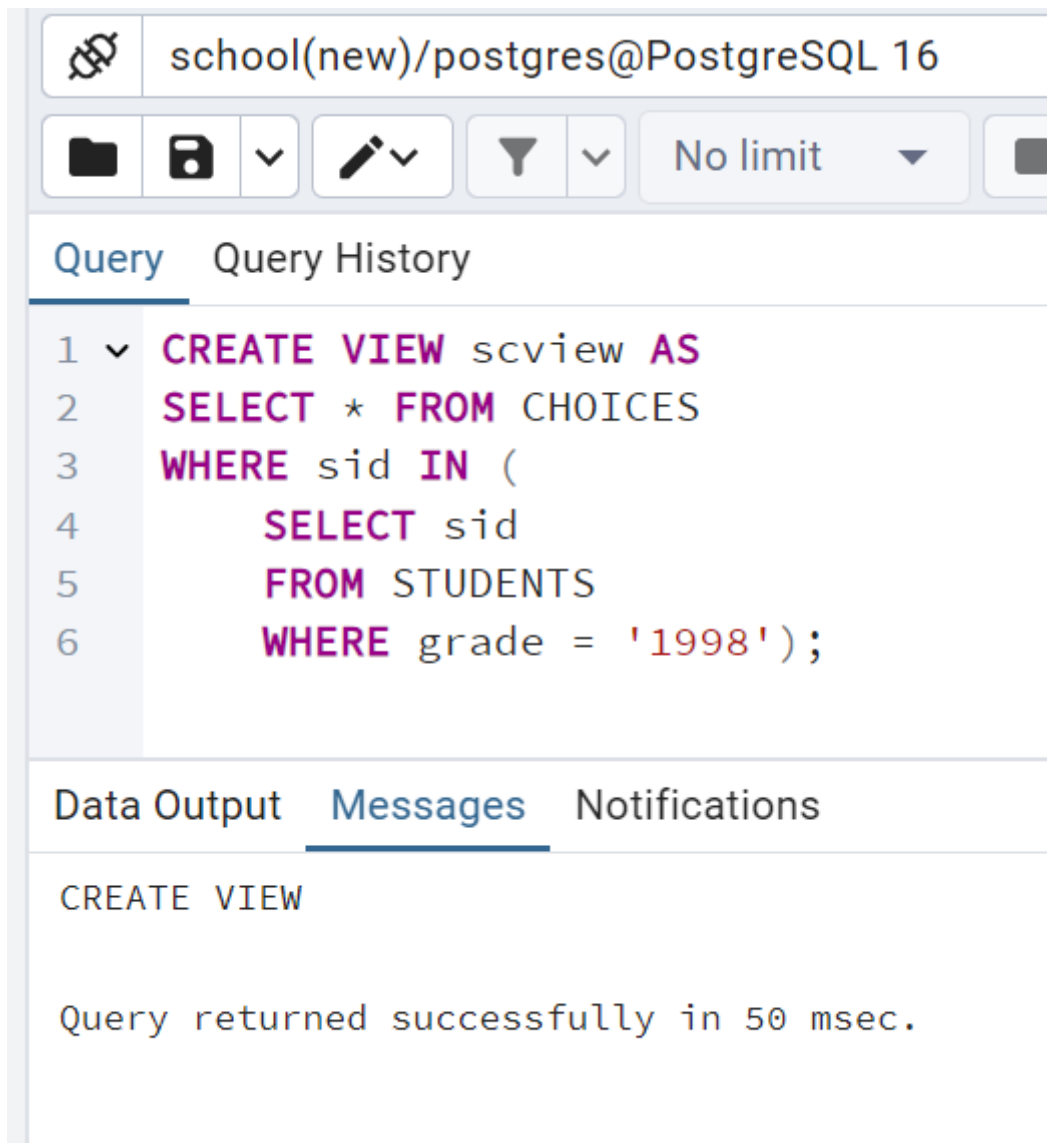
### 2. 对用户授权

#### 问题1:试解决赵老师能查询本年级学生的选课信息？

首先创建2008级学生选课信息的视图 scview,把访问该视图的权限授予赵老师，最后验证赵老师能否访问该视图？

由于没有2008级的学生，因此为了方便后面继续实验，我们把年级改为1998进行查询

创建scview成功：



The screenshot shows a web-based PostgreSQL interface. At the top, the connection string is 'school(new)/postgres@PostgreSQL 16'. Below this is a toolbar with icons for file operations, editing, and filtering. The main area is divided into two tabs: 'Query' and 'Query History'. The 'Query' tab is active, displaying a SQL query to create a view named 'scview'. The query selects all columns from the 'CHOICES' table where the 'sid' is in a subquery that selects 'sid' from the 'STUDENTS' table where the 'grade' is '1998'. Below the query editor, there are three tabs: 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is active, showing the message 'CREATE VIEW' and 'Query returned successfully in 50 msec.'

```
1 CREATE VIEW scview AS
2 SELECT * FROM CHOICES
3 WHERE sid IN (
4     SELECT sid
5     FROM STUDENTS
6     WHERE grade = '1998');
```

CREATE VIEW

Query returned successfully in 50 msec.

授权成功：

school(new)/postgres@PostgreSQL 16

No limit

Query
Query History

1
GRANT SELECT ON scview TO logzhao;

Data Output
Messages
Notifications

GRANT

Query returned successfully in 44 msec.

切换用户，验证赵老师能访问该视图：

Object Explorer

1.3 Sequences

Tables (7)

choices
course
courses
sc
stu\_union
students
teachers

Trigger Functions

Types

Views

Subscriptions

Login/Group Roles (19)

logzhao
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

SQL
Processes
CHOICES.sql\*
school(new)/logzhao@PostgreSQL 16\*

school(new)/logzhao@PostgreSQL 16

No limit

Query
Query History
Scratch Pad

1
SELECT \* FROM scview;

Data Output
Messages
Notifications

	no integer	sid character (9)	tid character (9)	cid character (5)	score integer
1	500004144	817636568	253205179	10047	60
2	500012952	835265794	244275084	10043	68
3	500015544	893821981	287021451	10015	84
4	500021670	855727911	201929763	10027	78
5	500027163	847843459	227295470	10025	[null]
6	500033812	820193911	238001793	10025	65
7	500040843	860579754	286542488	10020	53
8	500050662	833389259	257941400	10044	74
9	500054008	858506173	297235958	10048	61
10	500054941	835410506	264024214	10033	76
11	500056030	819331388	274499985	10022	51
12	500059622	838623090	209595691	10040	53
13	500065969	838938981	250737076	10003	[null]
14	500087616	833544211	243426525	10040	54

Total rows: 1000 of 19861
Query complete 00:00:00.321
Ln 1, Col 22

## 问题2:试解决让赵老师了解某课程的选课情况?

遇到的问题太多为了不影响实验报告观感我写在最后

首先创建能查询指定课程选课信息的存储过程 scpro:

```
CREATE OR REPLACE FUNCTION scpro(course_id CHAR)
RETURNS TABLE(no INT, sid CHAR, tid CHAR, cid CHAR, score INT) AS
$$
BEGIN
    RETURN QUERY
    SELECT *
    FROM scview c
    WHERE c.cid = course_id;
END;
$$
LANGUAGE plpgsql;
```

The screenshot shows a SQL IDE interface with a query editor and a messages pane. The query editor contains the SQL code for creating the scpro function. The messages pane shows the execution result: "CREATE FUNCTION" and "Query returned successfully in 93 msec."

Query	Query History
1	CREATE OR REPLACE FUNCTION scpro(course_id CHAR)
2	RETURNS TABLE(no INT, sid CHAR, tid CHAR, cid CHAR
3	\$\$
4	BEGIN
5	RETURN QUERY
6	SELECT *
7	FROM scview c
8	WHERE c.cid = course_id;
9	END;
10	\$\$
11	LANGUAGE plpgsql;

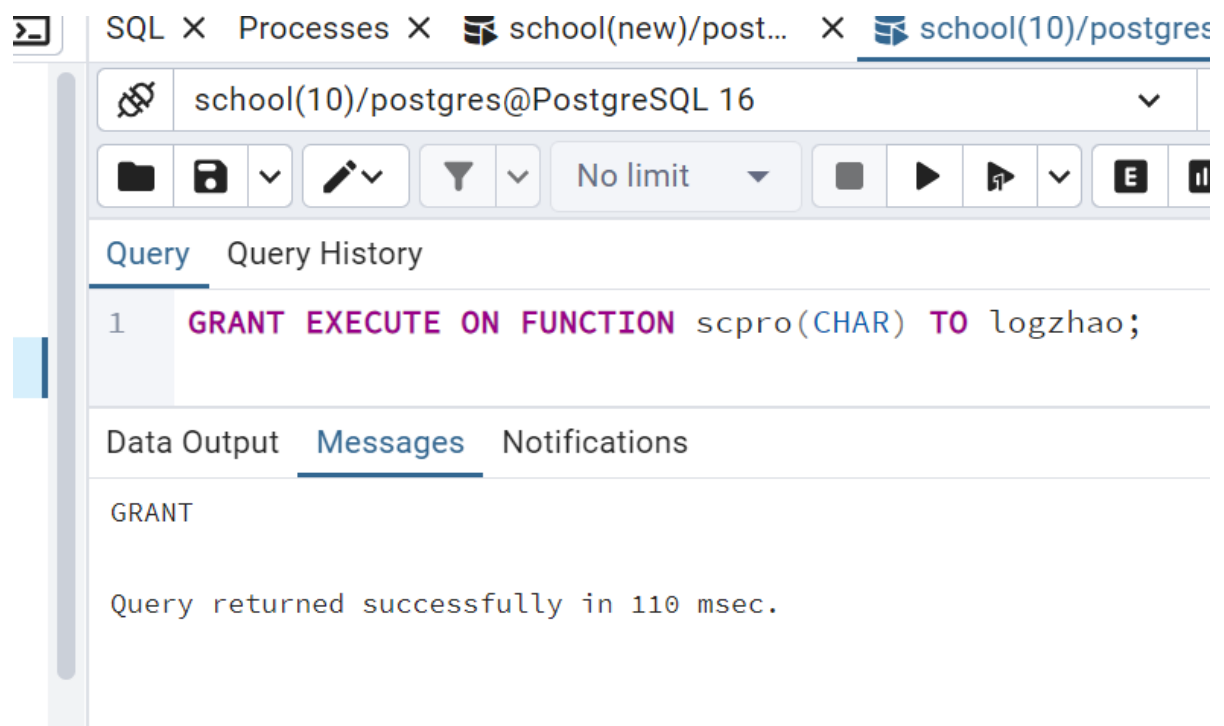
Data Output   **Messages**   Notifications

CREATE FUNCTION

Query returned successfully in 93 msec.

创建成功

把执行该存储过程的权限授予赵老师:













验证赵老师能否执行存储过程，可以看见能够执行：

school(10)/logzhao@PostgreSQL 16					
<div> <div> <div></div> <div></div> <div></div> <div></div> <div>No limit</div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> </div>					
Query Query History					
1 SELECT * FROM scpro('10021');					
Data Output Messages Notifications					
<div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div>SQL</div> </div>					
	no integer	sid character	tid character	cid character	score integer
1	500174588	883198563	262565909	10021	[null]
2	500473022	821637081	253107547	10021	[null]
3	500950603	827087891	269061913	10021	77
4	501119253	871750571	221687192	10021	64
5	501163200	861243120	289314721	10021	81
6	501429205	816674539	291717354	10021	72
7	501915071	849924318	208028576	10021	76
8	503225949	807428857	244086468	10021	[null]
9	503414869	873214688	252273969	10021	91
10	503648891	835861655	201697986	10021	92
11	504133032	886935711	241786771	10021	70

补充内容：撤销赵老师查询某课程的选课情况，再验证赵老师能否执行存储过程？

撤销成功：

 school(10)/postgres@PostgreSQL 16

 No limit 


[Query](#) [Query History](#)










1 **REVOKE SELECT ON** scview **FROM** logzhao;

[Data Output](#) [Messages](#) [Notifications](#)

REVOKE

Query returned successfully in 83 msec.

 school(10)/logzhao@PostgreSQL 16

 No limit 

[Query](#) [Query History](#)

1 **SELECT \* FROM** scpro('10021');

[Data Output](#) [Messages](#) [Notifications](#)

ERROR: 对视图 scview 权限不够  
CONTEXT: SQL 语句 "SELECT \*  
FROM scview c  
WHERE c.cid = course\_id"  
在RETURN QUERY的第3行的PL/pgSQL函数scpro(character)

错误: 对视图 scview 权限不够  
SQL state: 42501

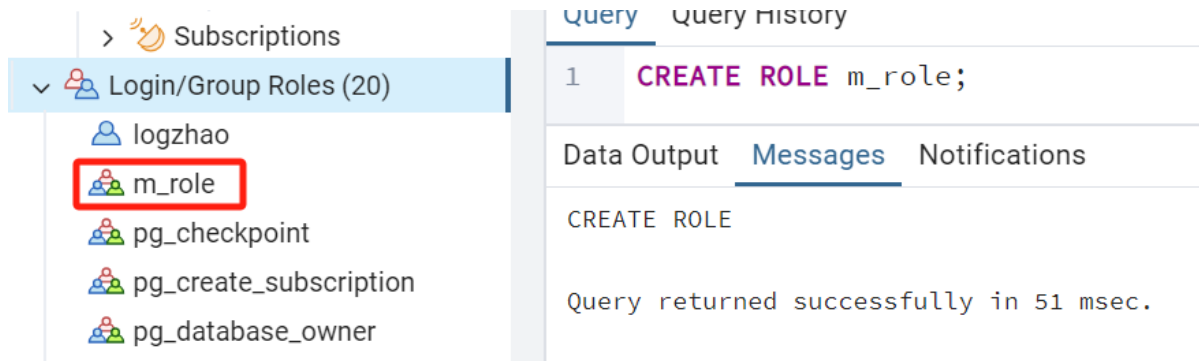
可以看到logzhao无法再执行存储过程。

### 3. 角色管理

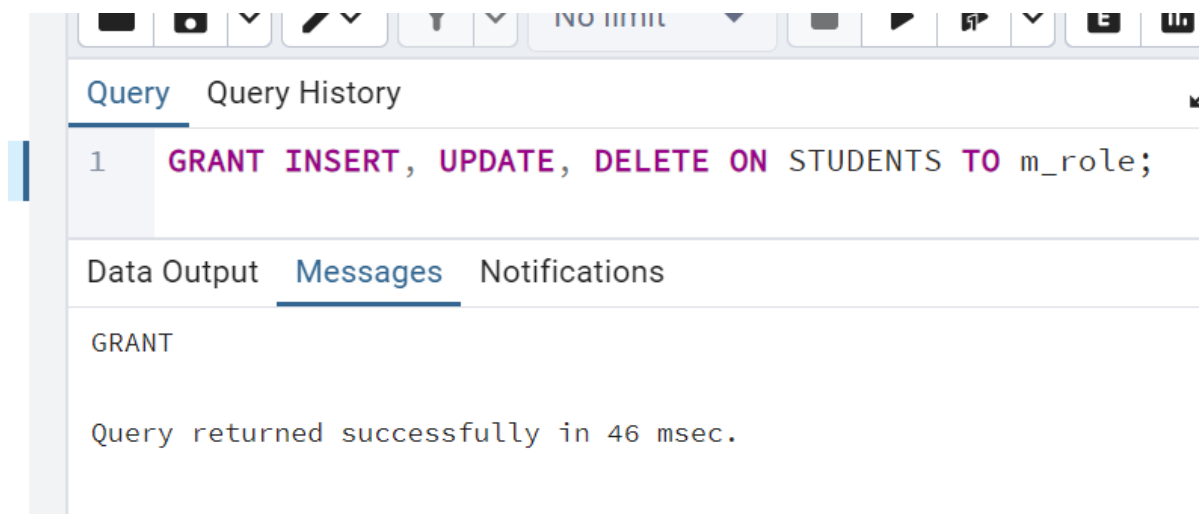
问题：假如学校新增10个辅导员，都要在student表中添加、修改和删除学生，要个个设置权限，方便吗？

可以考虑利用数据库的角色管理来实现：

首先创建辅导员角色m\_role
















然后对角色进行插入操作授权



再创建各个辅导员的登录以及对应的登录用户，使这些用户成为角色成员



 school(10)/postgres@PostgreSQL 16

 No limit 

Query

Query History

1

CREATE USER counselor1 WITH PASSWORD '123456';

2

GRANT m\_role TO counselor1;













Data Output

Messages

Notifications

GRANT ROLE

Query returned successfully in 40 msec.

 No limit 

Query

Query History

1

CREATE USER counselor2 WITH PASSWORD '123456';

2

GRANT m\_role TO counselor2;

Data Output

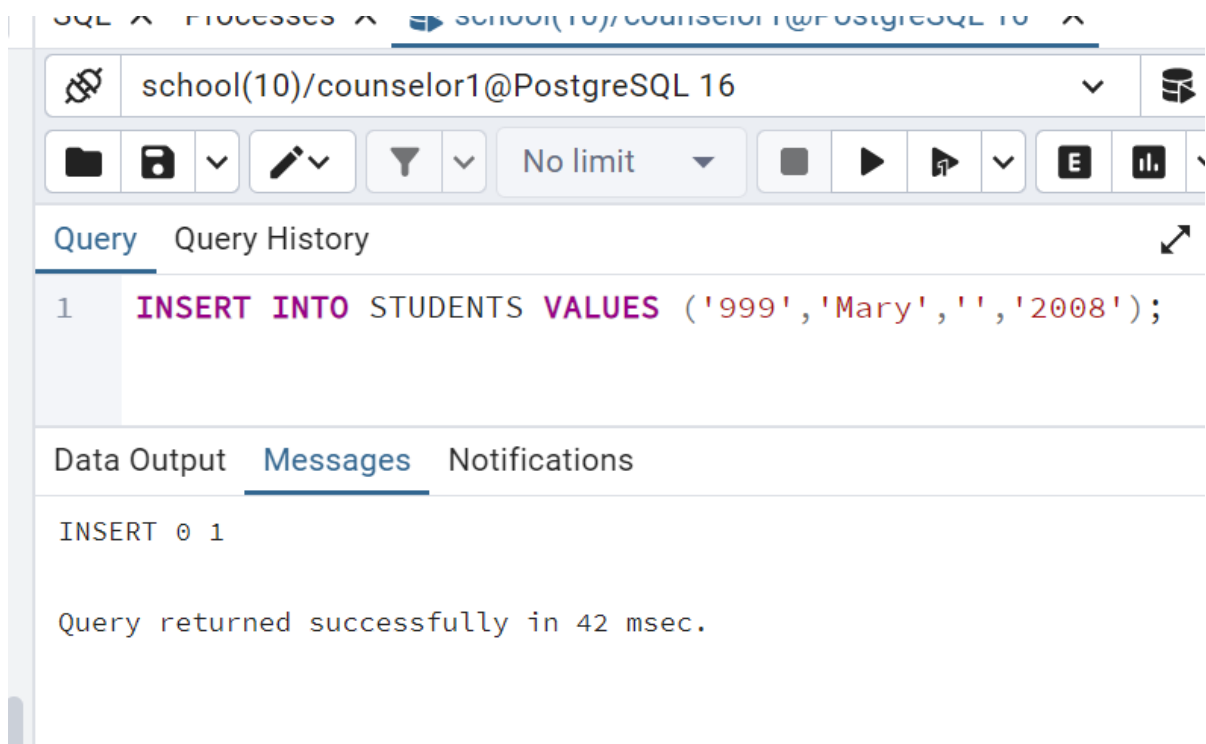
Messages

Notifications

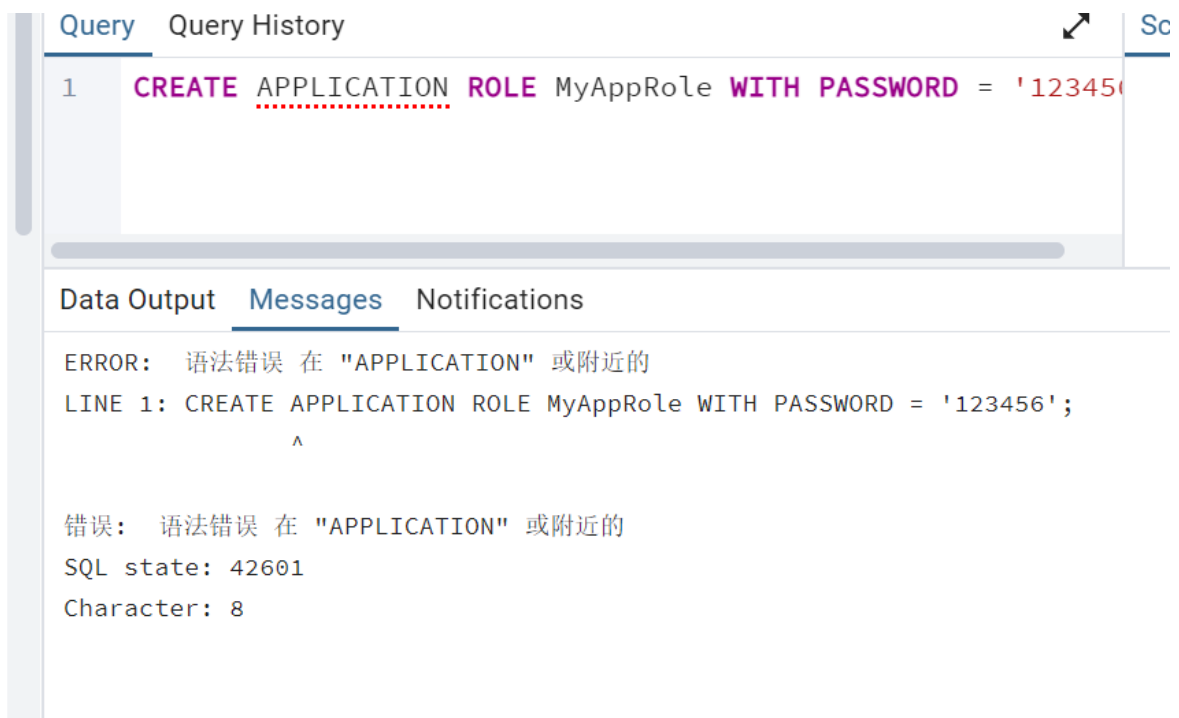
GRANT ROLE

Query returned successfully in 37 msec.

切换到其中一个辅导员，发现权限可以使用：



对于应用程序角色，暂时不知道具体是什么原因，导致无法使用 `CREATE APPLICATION ROLE` 命令，所以在此软件上无法使用应用程序角色进行授权



至此，实验结束。

## 遇到的问题

在问题二中，我刚开始是仿照报告给的代码创建存储过程；

school(new)/postgres@PostgreSQL 16

No limit

Query Query History

1

2

3

4

5

6

7

8

9

10

11

12

CREATE OR REPLACE PROCEDURE

scpro(IN course\_id CHAR)

AS

\$\$

BEGIN

SELECT c.\*, s.sid, s.sname, t.tid, t.tname

FROM CHOICES c

JOIN STUDENTS s ON c.sid = s.sid

JOIN TEACHERS t ON c.tid = t.tid

WHERE c.cid = course\_id;

END;

\$\$

LANGUAGE plpgsql;

Data Output

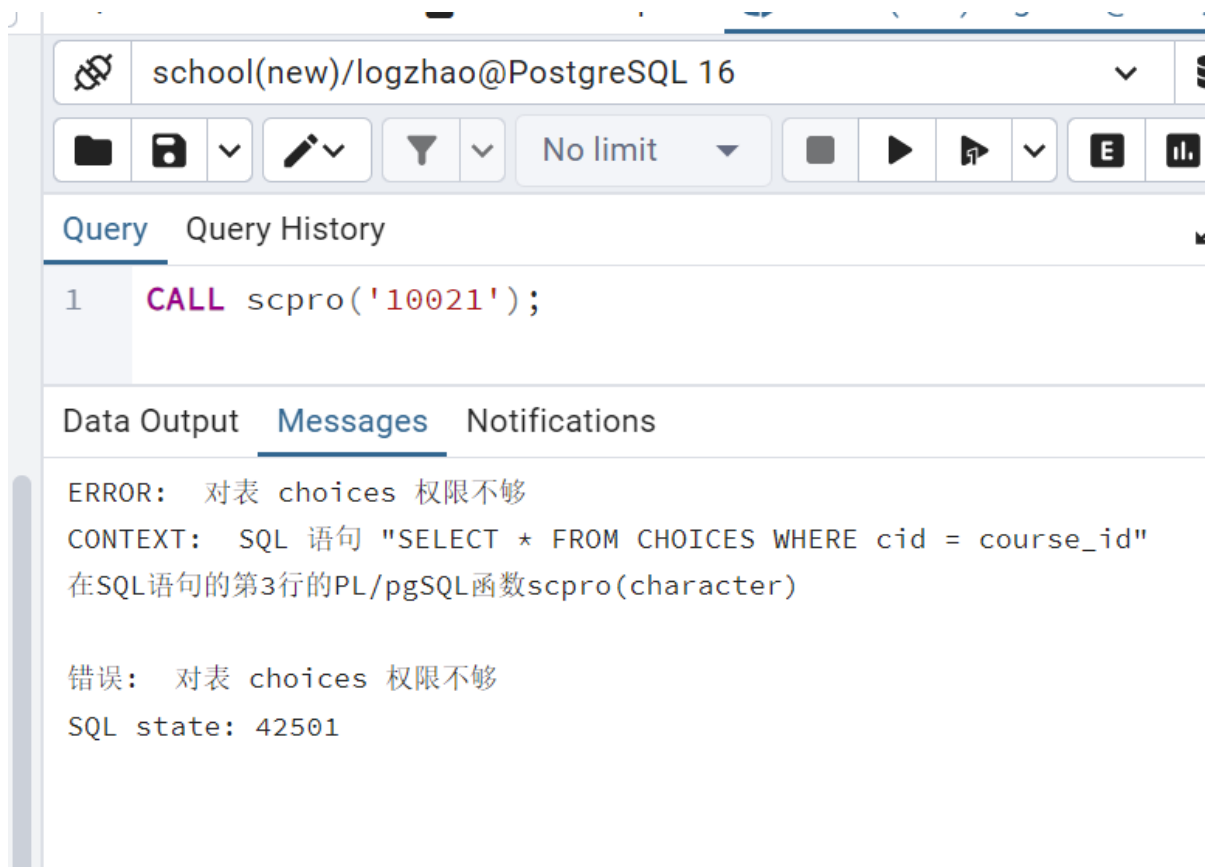
Messages

Notifications

CREATE PROCEDURE

Query returned successfully in 64 msec.

但后面发现对CHOICES 权限不够，原来存储过程相当于一个函数，logzhao用户调用这个函数时，若表格权限不够仍然无法调用：



前面我们已经把scview权限赋给logzhao，所以我们直接使用scview表格，修改代码：

SQL X Processes X CHOICES.sql\* X school(new)/postgres@l

school(new)/postgres@PostgreSQL 16

Query Query History

```
1 CREATE OR REPLACE PROCEDURE scpro(course_id CHAR)
2 AS
3 $$
4 BEGIN
5     SELECT * FROM scview WHERE cid = course_id;
6 END;
7 $$
8 LANGUAGE plpgsql;
```

Data Output Messages Notifications

CREATE PROCEDURE

Query returned successfully in 48 msec.

重复上面步骤：

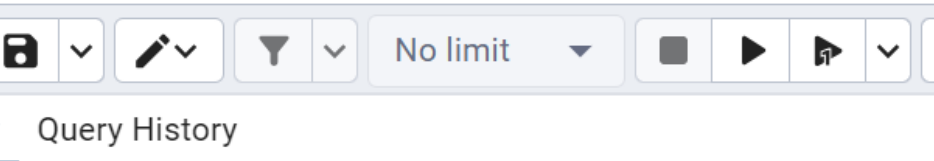
The screenshot shows a PostgreSQL client window titled "school(new)/logzhao@PostgreSQL 16". The interface includes a toolbar with icons for file operations, a query editor, and execution controls. The "Query" tab is active, displaying a single SQL statement: `CALL scpro('10021');`. Below the query editor, the "Messages" tab is selected, showing an error message:   
ERROR: 对于结果数据, 查询没有目标  
HINT: 如果您想要放弃SELECT语句的结果, 请使用PERFORM.  
CONTEXT: 在SQL语句的第3行的PL/pgSQL函数scpro(character)  
  
错误: 对于结果数据, 查询没有目标  
SQL state: 42601

于是再改代码, 把SELECT 改为PERFORM, 成功执行, 但我们看不到结果:

The screenshot shows the same PostgreSQL client window. The query editor now contains two lines: `call` and `scpro('10021')`. The "Messages" tab is selected, displaying the following output:   
CALL  
  
Query returned successfully in 247 msec.

因为PROCEDURE无法返回结果, 所以我们只能使用FUNCTION执行, 也就是实验报告中的代码

还有一个问题就是我执行了对logzhao执行存储过程scpro权限的撤销，却还是能够对scpro查询，目前没发现原因：



The screenshot shows a PostgreSQL client window titled 'school(10)/postgres@PostgreSQL 16'. The interface includes a toolbar with icons for file operations, a query editor, and execution controls. The 'Query' tab is active, displaying a single SQL statement: `REVOKE EXECUTE ON FUNCTION scpro(CHAR) FROM logzhao;`. Below the query, the 'Messages' tab is selected, showing the output: `REVOKE` and `Query returned successfully in 77 msec.`

school(10)/postgres@PostgreSQL 16

Query Query History

```
1 REVOKE EXECUTE ON FUNCTION scpro(CHAR) FROM logzhao;
```

Data Output Messages Notifications

REVOKE

Query returned successfully in 77 msec.

SQL > Processes > school(new)/post... > school(10)/logzhao

school(10)/logzhao@PostgreSQL 16

No limit

Query Query History

```
1 SELECT * FROM scpro('10021')
```

Data Output Messages Notifications

	no integer	sid character	tid character	cid character	score integer
1	500174588	883198563	262565909	10021	[null]
2	500473022	821637081	253107547	10021	[null]
3	500950603	827087891	269061913	10021	77
4	501119253	871750571	221687192	10021	64
5	501163200	861243120	289314721	10021	81
6	501429205	816674539	291717354	10021	72
7	501915071	849924318	208028576	10021	76
8	503225949	807428857	244086468	10021	[null]
9	503414869	873214688	252273969	10021	91
10	503648891	835861655	201697986	10021	92
11	504133032	886935711	241786771	10021	70

Total rows: 201 of 201    Query complete 00:00:00.100    Ln 1, Col 27