实验报告11

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。

1. 访问控制

1.1 创建登录用户和密码

首先，创建数据库的登录用户，并为其分配密码：

SQL

```sql
CREATE USER your_user WITH PASSWORD 'your_password';
```

1.2 授予权限

根据需要，授予用户相应的权限，例如：

```sql
GRANT SELECT, INSERT, UPDATE, DELETE ON your_table TO your_user;
```

2. 角色管理

2.1 创建角色

使用角色进行权限管理是一种有效的方式。创建角色并授予权限：

```sql
CREATE ROLE data_admin;
GRANT data_admin TO your_user;
```

2.2 分配角色

将角色分配给用户，以便用户继承角色的权限：

```sql
GRANT data_admin TO your_user;
```

3. SSL加密

启用SSL加密以确保数据在传输过程中的安全：

ssl = on

4. 存储过程和函数的权限控制

4.1 创建存储过程

```sql
CREATE OR REPLACE PROCEDURE your_procedure()
AS
$$
BEGIN
   -- Your logic here  (e.x. SELECT * FROM your_table WHERE condition;)
END;
$$
LANGUAGE plpgsql;
```

4.2 授予权限

为用户授予执行存储过程的权限：

```sql
GRANT EXECUTE ON PROCEDURE your_procedure() TO your_user;
```

5. 行级安全

使用行级安全策略限制用户对数据的访问：

```sql
ALTER TABLE your_table ENABLE ROW LEVEL SECURITY;
CREATE POLICY your_policy
   USING (your_condition)
   FOR ALL
   USING (true);
```

6. 审计日志

启用审计日志以跟踪数据库活动：

```sql
logging_collector = on
log_statement = 'all'
log_directory = '/var/log/postgresql/'
```

以上只是一个入门级的教程，实际上，数据库安全性控制涉及到更多方面，包括定期备份、更新数据库软件、监控异常活动等。在实际应用中，应根据具体需求和环境进行更详细的安全性配置。

课内实验

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

```sql
CREATE USER logzhao WITH PASSWORD '1111';
```

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

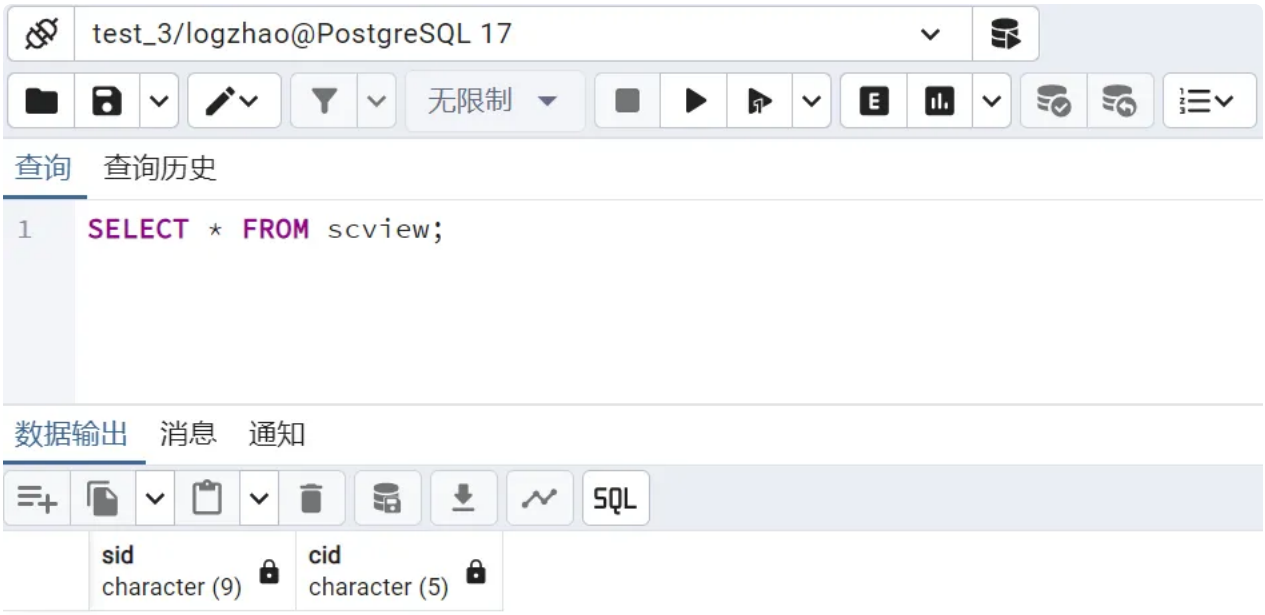
最后 验证赵老师能否访问该视图?

```sql
CREATE VIEW scview AS
SELECT sid,cid
FROM choices
WHERE sid in
(selecy sid from students where grade='2008')


GRANT SELECT ON scview TO logzhao;
```

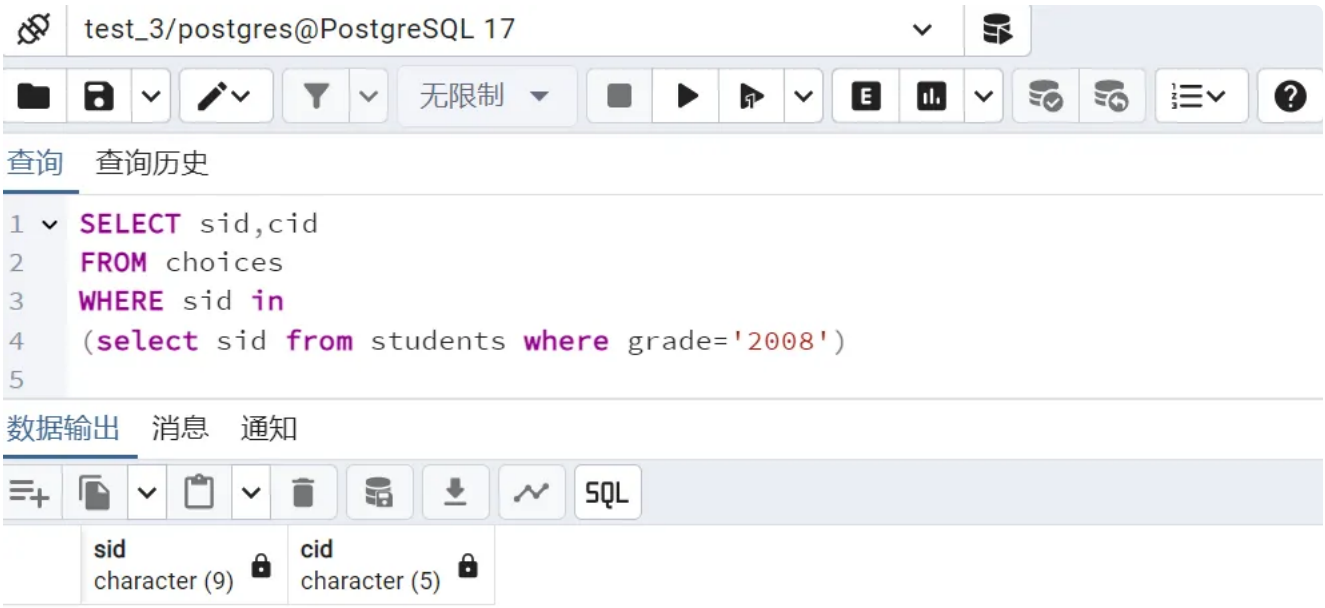
验证查询sql语句

```sql
SELECT * FROM scview;
```



赵老师去查询结果为0

返回在公共账号中查询结果也为空集，所以答案正确



问题2:试解决让赵老师了解某课程的选课情况?

首先创建能查询指定课程选课信息的存储过程 scpro,把执行该存储过程的权限授予赵 老师，最后验证赵老师能否执行存储过程?

```sql
CREATE OR REPLACE PROCEDURE scpro()
RETURNS TABLE(sid character) AS $$
    SELECT sid
    FROM chOICES
    WHERE cid = '10001';
$$ LANGUAGE sql;
```

```sql
GRANT EXECUTE ON FUNCTION scpro() TO logzhao
```

数据输出 消息 通知

```
错误:  对表 choices 权限不够
CONTEXT:  SQL 函数 "scpro" 语句 1

错误:  对表 choices 权限不够
SQL 状态: 42501
```

查询结果正常

查询 查询历史

```
1  select * from scpro()
```

数据输出 消息 通知

	sid character
1	870899566
2	818285935
3	896389791
4	827173338
5	830131870
6	869944480
7	806090255
8	841438368
9	876914321
10	842457951
11	856916681
12	886292532
13	877210408
14	834169057
15	893991860
16	815417168
17	828657260
18	822750091
19	804312075
20	877952697
21	850385507

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？

SQL

```sql
1  REVOKE SELECT ON TABLE choices FROM logzhao;
2
```

不能执行，因为没有该表的权限

数据输出 消息 通知

```
错误:  对表 choices 权限不够
CONTEXT:  SQL 函数 "scpro" 语句 1

错误:  对表 choices 权限不够
SQL 状态: 42501
```

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，再验证用户是否有插入操作的权限?

SQL

```sql
CREATE ROLE m_role;
GRANT insert,delete,update on students TO m_role;

CREATE USER fu1 WITH PASSWORD '1111';
GRANT m_role TO fu1;
```

SQL

```sql
insert into students(sid,sname,grade)
values(22336122,'lixi',2022)

```

数据输出 消息 通知

```
INSERT 0 1

耗时47 毫秒 成功返回查询。
```

还可以考虑应用程序角色来实现：

```sql
create role app role with login;
grant insert on students to app role;
set session authorization 'app role';


insert into students values('123456788','lx','lovecc',2022);
reset session authorization;
select*from students where email='lovecc';
```

创建应用程序角色，激活该角色，对其进行插入操作的授权，验证是否具有该操作的 权限?

实验总结

在实际操作中，我掌握了创建登录用户、授予权限、创建视图和存储过程等关键技能。同时，利用角色管理来简化用户权限管理，提高了工作效率。