Database-System Experiment-13

21307289 刘森元

1. 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。

2. 实验环境

Macbook Pro, 14 inchs, 2021

Apple M1 Pro

macOS Sonoma 14.1.1

psql (PostgreSQL) 15.4 (Homebrew)

3. 实验内容

1. 登录管理

1.1. 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?

创建登录账号 logzhao

```
CREATE USER logzhao WITH PASSWORD 'logzhao'
```

```
school=# \du
                                   角色列表
  角色名称   |                         属性                         | 成员属于
-------------+------------------------------------------------------+----------
 logzhao     |                                                      | {}
 qiu_nangong | 超级用户, 建立角色, 建立 DB, 复制, 绕过RLS | {}
```

验证该账号与数据库的连接访问是否正确

```
 @qiu_nangong ▶ psql --username=logzhao school
口令:
psql (15.4 (Homebrew))
输入 "help" 来获取帮助信息.

school=>
```

可见能成功访问

2. 对用户授权

2.1. 试解决赵老师能查询本年级学生的选课信息?

首先创建选课信息视图

```
-- 创建视图
CREATE VIEW scview AS
SELECT s.sid, s.sname, c.cname, c.cid, t.tname, t.tid
FROM CHOICES ch
JOIN STUDENTS s ON s.sid = ch.sid
JOIN COURSES c ON ch.cid = c.cid
JOIN TEACHERS t ON t.tid = ch.tid
WHERE s.grade = 2008;
```

授予视图访问权限

```
-- 授予权限
GRANT SELECT ON scview TO logzhao;
```

验证能否访问视图

```
school=> SELECT COUNT(*) FROM scview;
 count
-------
 20050
(1 行记录)
```

可见能够成功访问

2.2. 试解决让赵老师了解某课程的选课情况

首先创建存储过程 scpro，并授予权限

```
CREATE OR REPLACE PROCEDURE scpro(course_id CHARACTER(5))
AS
$$
BEGIN
    PERFORM *
    FROM scview
    WHERE cid = course_id;
END;
$$
LANGUAGE plpgsql;

-- 授予权限
GRANT EXECUTE ON PROCEDURE scpro(CHAR) TO logzhao;
```

验证其能否执行

```
school=> CALL scpro('10001');
CALL
```

可见能够成功执行

撤销其查询权限

```
REVOKE SELECT ON scview FROM logzhao;
```

```
school=> CALL scpro('10001');
ERROR:  permission denied for view scview
背景:  SQL statement "SELECT *
    FROM scview
    WHERE cid = course_id"
PL/pgSQL function scpro(character) line 3 at PERFORM
```

可见其不能够再执行该过程

3. 角色管理

3.1. 利用数据库的角色管理实现

创建辅导员角色m_role，并对角色进行插入操作的授权。

```
-- 创建角色
CREATE ROLE m_role;

-- 授予权限
GRANT INSERT, UPDATE, DELETE ON STUDENTS TO m_role;
```

创建各个辅导员的登录用户，并使这些用户成为角色成员，最后验证用户是否具有插入操作的权限。

```
-- 创建辅导员登录用户
CREATE USER instructor1 WITH PASSWORD 'password1';
CREATE USER instructor2 WITH PASSWORD 'password2';
-- ... 创建其他辅导员登录用户

-- 将用户添加到角色中
GRANT m_role TO instructor1;
GRANT m_role TO instructor2;
-- ... 将其他辅导员用户添加到角色中

-- 验证权限
-- 连接数据库使用辅导员用户进行插入操作
INSERT INTO STUDENTS (sid, sname, email, grade)
VALUES ('114514191', 'John', 'john@example.com', 2008);
```

```
school=> INSERT INTO STUDENTS (sid, sname, email, grade)
VALUES ('114514191', 'John', 'john@example.com', 2008);
INSERT 0 1
```

可见能够成功插入

3.2. 应用程序角色实现

创建应用程序角色，并激活该角色。

```
-- 创建应用程序角色
CREATE ROLE app_role;

-- 激活角色
SET ROLE app_role;
```

对应用程序角色进行插入操作的授权，验证是否具有该操作的权限。

```
-- 授予权限
GRANT INSERT ON STUDENTS TO app_role;

-- 验证权限
-- 连接数据库使用应用程序角色进行插入操作
INSERT INTO STUDENTS (sid, sname, email, grade)
VALUES (1, 'John', 'john@example.com', '2008');
```

```
school=> INSERT INTO STUDENTS (sid, sname, email, grade)
VALUES ('114514191', 'John', 'john@example.com', 2008);
INSERT 0 1
```

可见能够成功插入