**暨南大学本科实验报告专用纸**

课程名称 《数据库系统》 成绩评定

实验项目名称 指导教师雷小林 、魏林锋

实验项目编号 实验项目类型 综合 实验地点 N117

学生姓名 金鸿正 学号 2021102560

学院 信息科学与技术学院 系 计算机 专业 软件工程

实验时间 年 月 日 午～ 月 日 午 温度 ℃湿度

2 安全性语言实验

2.1 自主存取控制实验

2.1.1 实验目的

掌握自主存取控制权限的定义和维护方法。

2.1.2 实验内容和要求

定义用户、角色，分配权限给用户、角色，回收权限，以相应的用户名登陆数据库验证权限分配是否正确。选择一个应用场景，使用自主存取控制机制设计权限分配。可以采用两种方案。

方案一：采用SYSTEM 超级用户登录数据库（对于 MYSQL(Linux) 则是 root），完成所有权限分配工作，然后用相应用户名登陆数据库以验证权限分配正确性；

方案二：采用 SYSTEM 用户登陆数据库创建三个部门经理用户，并分配相应的权限，然后分别用三个经理用户名登陆数据库，创建相应部门的 USER、ROLE ，并分配相应权限。

2.1.3 实验重点和难点

实验重点：定义角色，分配权限和回收权限。

实验难点：实验方案二实现权限的再分配和回收。

2.1.4 实验内容记录

本次实验中，为财务部（finance）创建一个管理员，部门的管理员只有管理自己部门内部的员工的权限以及访问部门表（departments）、部门管理员表（dept\_manager）的权限。

注：为了减少工作量，只为财务部门创建相关用户。其它部门的操作可以类比。

2.1.4.1 建立视图

数据库基本表没有区分不同的部门，因此通过视图来划分不同部门的员工。

先创建员工所在部门视图：

CREATE VIEW v\_emp\_dept(emp\_no, dept\_no, dept\_name) AS

SELECT de.emp\_no, d.dept\_no, d.dept\_name

FROM dept\_emp de, departments d

WHERE de.dept\_no=d.dept\_no;

然后为各个表创建部门视图：

CREATE VIEW v\_finance\_employees AS

SELECT e.\*

FROM employees e, v\_emp\_dept ved

WHERE e.emp\_no=ved.emp\_no

AND ved.dept\_name='Finance'

WITH CHECK OPTION;

CREATE VIEW v\_finance\_salaries AS

SELECT s.\*

FROM salaries s, v\_emp\_dept ved

WHERE s.emp\_no=ved.emp\_no

AND ved.dept\_name='Finance'

WITH CHECK OPTION;

CREATE VIEW v\_finance\_dept\_emp AS

SELECT de.\*

FROM dept\_emp de, v\_emp\_dept ved

WHERE de.emp\_no=ved.emp\_no

AND ved.dept\_name='Finance'

WITH CHECK OPTION;

CREATE VIEW v\_finance\_titles AS

SELECT t.\*

FROM titles t, v\_emp\_dept ved

WHERE t.emp\_no=ved.emp\_no

AND ved.dept\_name='Finance'

WITH CHECK OPTION;

2.1.4.2 建立角色

为财务部门建立一个经理角色（完全控制本部门数据，同时能查询其它部门的数据，能够为本部门职员分配权限），一个职员角色（可以插入和查询本部门数据）。

CREATE ROLE

role\_finance\_manager,

role\_finance\_staff;

为角色分配权限。

注：MYSQL 只有 WITH GRANT OPTION，并且行为是非级联权限收回。

先分配部门经理角色对本部门表的权限。

GRANT ALL

ON emplorees.v\_finance\_emplorees

TO role\_finance\_manager

WITH GRANT OPTION;

GRANT ALL

ON emplorees.v\_finance\_salaries

TO role\_finance\_manager

WITH GRANT OPTION;

GRANT ALL

ON emplorees.v\_finance\_dept\_emp

TO role\_finance\_manager

WITH GRANT OPTION;

GRANT ALL

ON emplorees.v\_finance\_titles

TO role\_finance\_manager

WITH GRANT OPTION;

分配部门经理对其它部门表的查询权限。

GRANT SELECT

ON employees.\*

TO role\_finance\_manager;

分配员工对本部门的权限。

GRANT SELECT, INSERT

ON employees.v\_finance\_employees

TO role\_finance\_staff;

GRANT SELECT, INSERT

ON employees.v\_finance\_salaries

TO role\_finance\_staff;

GRANT SELECT, INSERT

ON employees.v\_finance\_dept\_emp

TO role\_finance\_staff;

GRANT SELECT, INSERT

ON employees.v\_finance\_titles

TO role\_finance\_staff;

2.1.4.3 建立用户

为财务部门建立一个经理用户和一个职员用户。

CREATE USER user\_manager

IDENTIFIED BY '123456';

CREATE USER user\_staff

IDENTIFIED BY '123456';

将角色赋予用户。

GRANT role\_finance\_manager

TO user\_manager;

GRANT role\_finance\_staff

TO user\_staff;

2.1.4.4 查询角色和用户的权限

查询权限使用 SHOW GRANTS 。

查询角色的权限。

SHOW GRANTS FOR role\_finance\_manager;

SHOW GRANTS FOR role\_finance\_staff;

查询用户的权限。

SHOW GRANTS FOR user\_manager;

SHOW GRANTS FOR user\_staff;

.1.4.5 验证权限分配正确性

仅仅对职员用户进行尝试。

使用职员用户登录。

访问数据库时，会出现报错：

ERROR 1044 (42000): Access denied for user 'user\_staff'@'%' to database 'employees'

原因是角色未启用。 [1]

登录后，启用角色

SET ROLE role\_finance\_staff;

此时查询表，为

思考题:

* 请分析 WITH CHECK OPTION、WITH GRANT OPTION、WITH ADMIN OPTION 有何区别和联系。

WITH CHECK OPTION 是视图选项，用于达到对视图进行更新时检查视图的条件的要求。WITH GRANT OPTION 和 WITH ADMIN OPTION 在授权时使用，前者（WITH GRANT OPTION）在权限被撤销时，会级联撤销，后者不会级联撤销。

注：MYSQL 中只有 WITH GRANT OPTION ，且其行为类似 WITH ADMIN OPTION 。

* 请结合上述实验示例分析使用角色进行权限分配有何优缺点。

使用角色进行权限分配便于管理权限（一个角色对应一种权限，而不是一个用户对应一种权限）。

2.2 审计实验

注：SQL 标准中不包含审计语句。

暂略。

**暨南大学本科实验报告专用纸(附页)**