

数据库系统课程实验报告

|  |  |
| --- | --- |
| 实验名称： | 数据库的安全性 |
| 实验日期： | 2023.05.05 |
| 实验地点： | 文宣楼B311 |
| 提交日期： |  |
|  | |
| 学号： | 32420212202930 |
| 姓名： | 陈澄 |
| 专业年级： | 软工2021级 |
| 学年学期： | 2022-2023学年第二学期 |

1. 实验目的

 理解数据库系统用户(user)、权限(privilege)和角色(role)的概念和作用

 熟练掌握用户的管理：创建、查看、删除和权限的授予与回收

 熟练掌握通过数据字典查看用户权限、表和视图权限的方法

 熟练掌握使用 Grant 命令给用户、角色授权的方法

 熟练掌握使用 Revoke 命令回收已授权限的方法

 熟练掌握角色定义、重命名和删除的方法

 熟练掌握修改角色中权限的方法

 理解视图的安全性作用

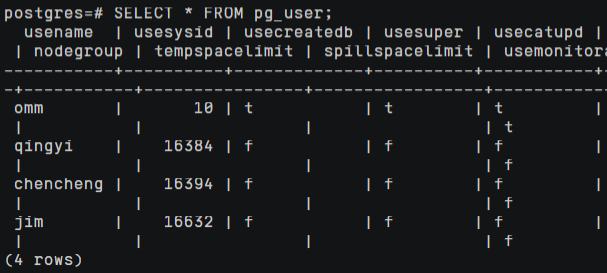
1. 实验内容和步骤

学习并完成 https://bokai.blog.csdn.net/article/details/117912175 的内容。

创建用户jim，登录密码为Bigdata@123。



查看用户列表



为用户jim追加有创建角色的CREATE ROLE权限



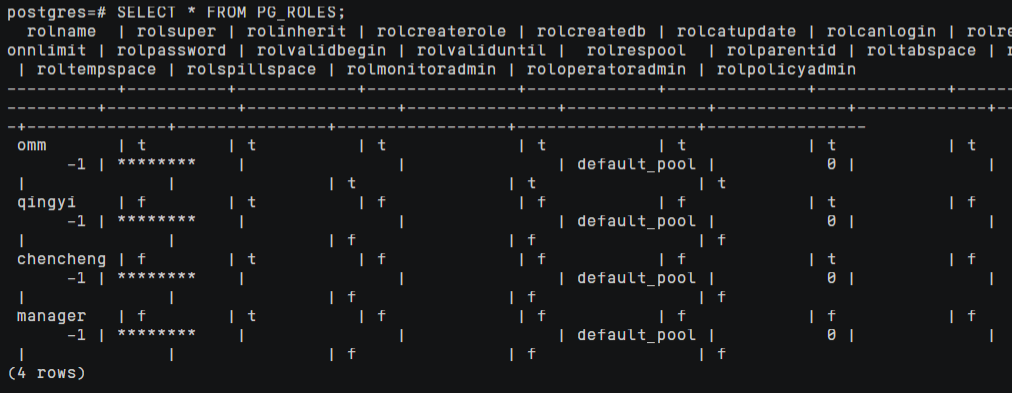
删除用户



创建一个角色，名为manager，密码为Bigdata@123



查看角色



修改角色manager的密码为abcd@123。



修改角色manager为系统管理员



删除角色manager



创建名为joe的用户:



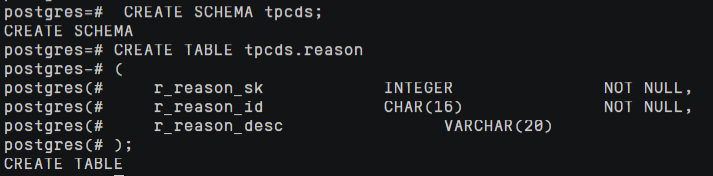
撤销joe用户的sysadmin权限



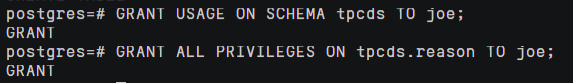
创建tpcds模式



tpcds模式下创建一张reason表



将模式tpcds的使用权限和表tpcds.reason的所有权限授权给用户joe



将tpcds.reason表中r\_reason\_sk、r\_reason\_id、r\_reason\_desc列的查询权限，r\_reason\_desc的更新权限授权给joe



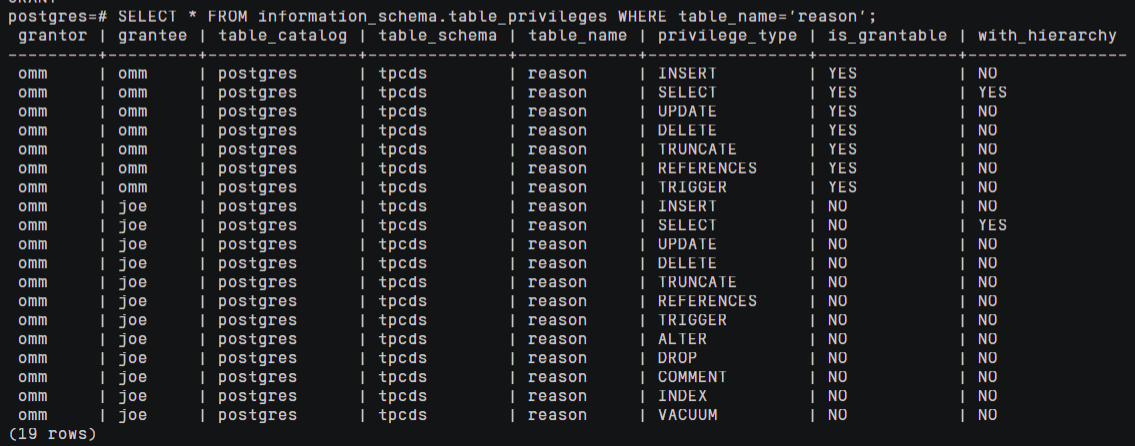
创建角色tpcds\_manager



将模式tpcds的访问权限授权给角色tpcds\_manager，并授予该角色在tpcds下创建对象的权限，不允许该角色中的用户将权限授权给其人



查看表reason权限：



创建角色manager



将joe的权限授权给manager，并允许该角色将权限授权给其他人



创建用户senior\_manager



将用户manager的权限授权给该用户



逐步回收manager权限

删除manager用户

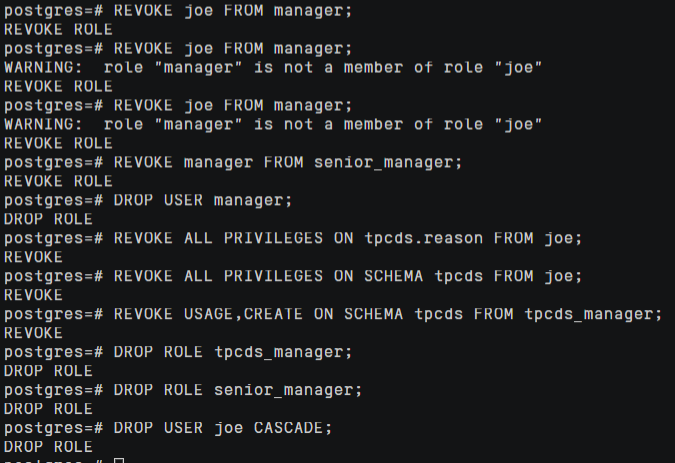
逐步回收joe权限

逐步回收tpcds\_manager权限

删除 tpcds\_manager用户

删除senior\_manager用户

删除joe用户

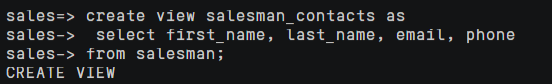


1. 创建视图 salesman，该视图只保存 employees 表中所有 job\_title 为'Sales Representative'的雇员。

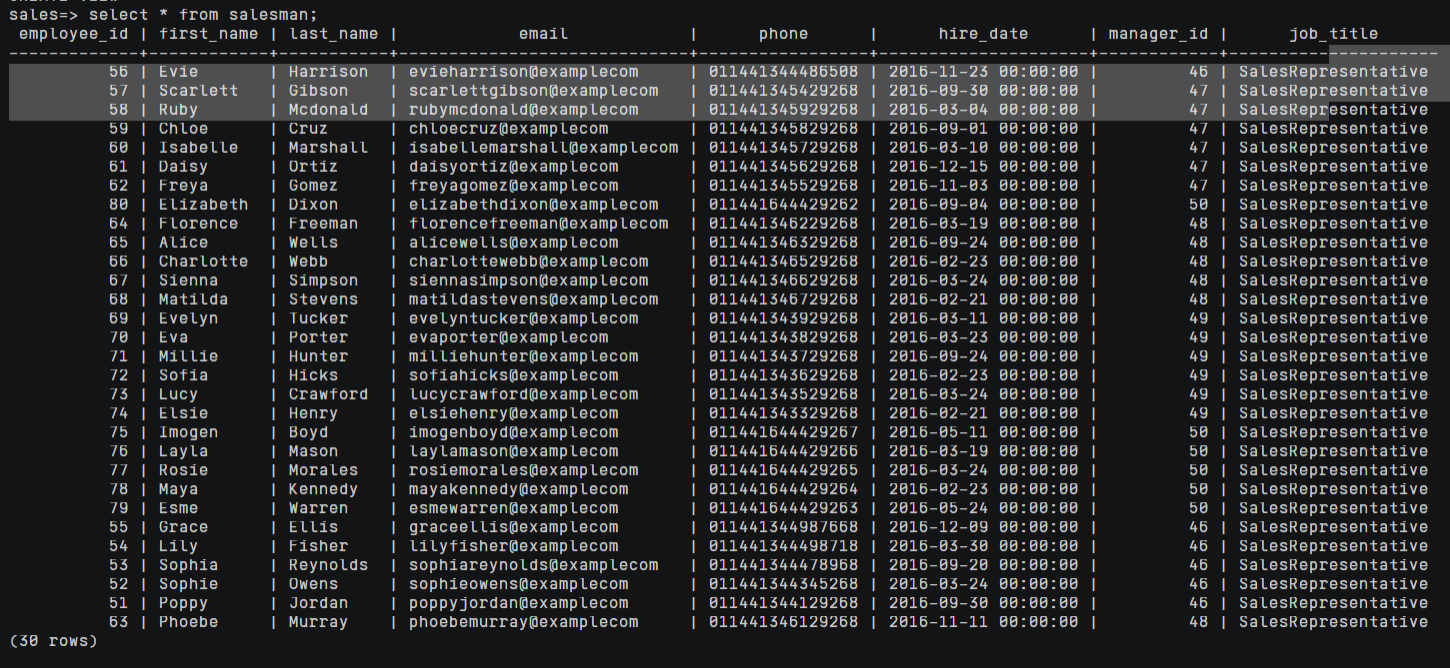


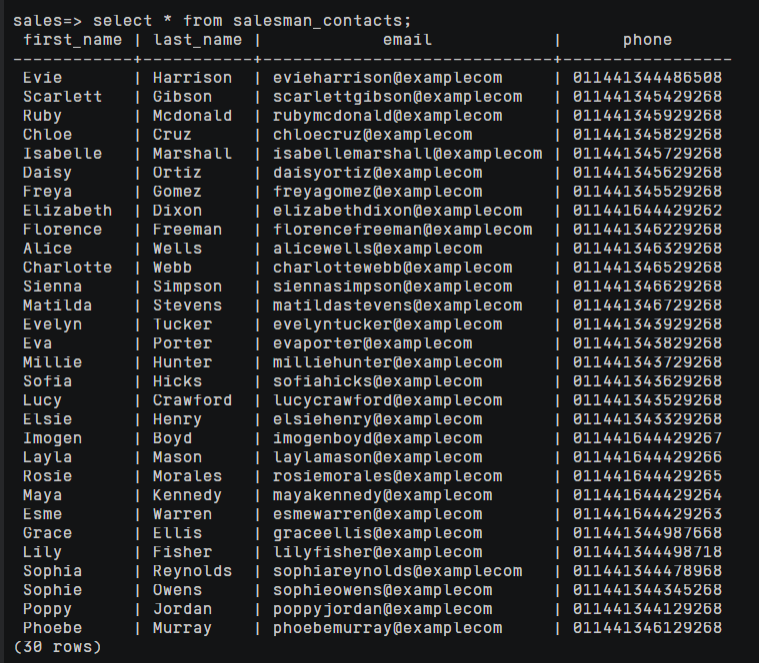
（3）创建基于 salesman 的视图 salesman\_contacts(first\_name,last\_name,email,phone)，该视图存储的

salesman 的联系方式。

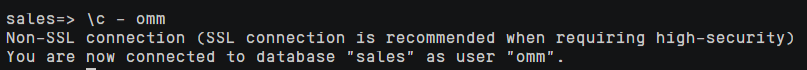


（4）查询视图 salesman 和 salesman\_contacts。





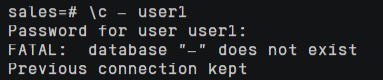
1. 在当前窗口输入命令：\c - omm 切换到 omm 用户。

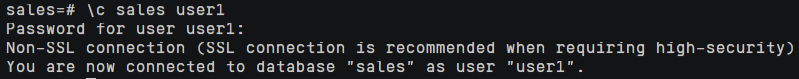


1. 创建新用户 user1。



1. 在当前窗口输入命令：\c – user1 切换到 user1 用户。

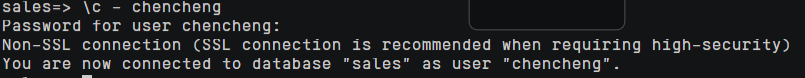




1. 发布查询命令：select \* from salesman\_contacts;观察结果。



1. 发布命令：\c – whj 切换到 whj 用户—此处的 whj 应替换为你们自己创建的用户。



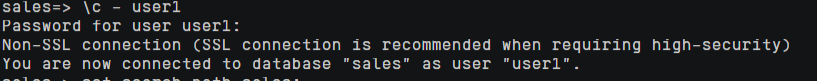
1. 在当前 whj 用户下输入命令：grant select on alesman\_contacts to user1; 实现授权操作。

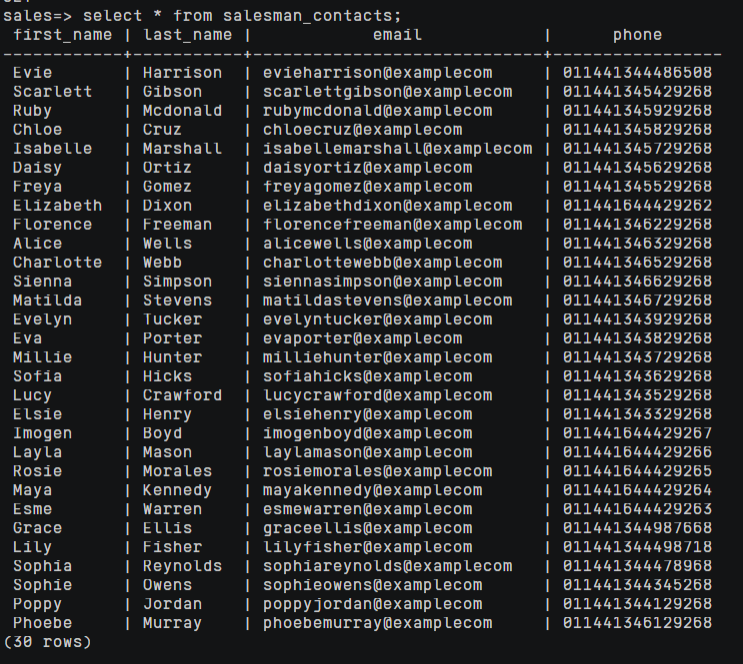


（11）依次重复步骤（7）和（8），比较两次查询的结果。

/\*说明：步骤（2）-（11）的主要目的是用于验证视图的作用：被授权用户只能查询在权限范围内

的数据，范围外的数据不可访问\*/

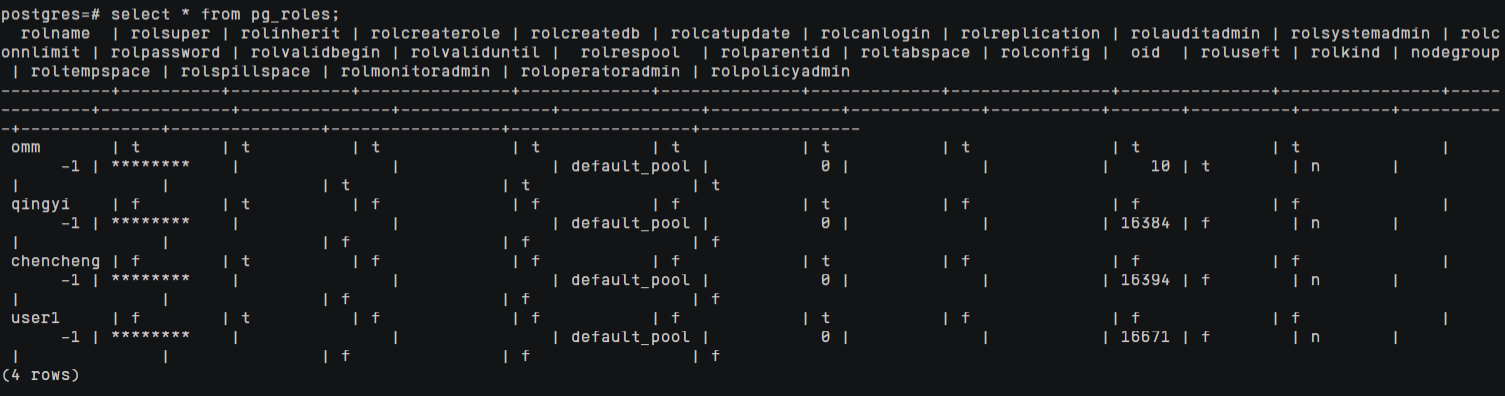


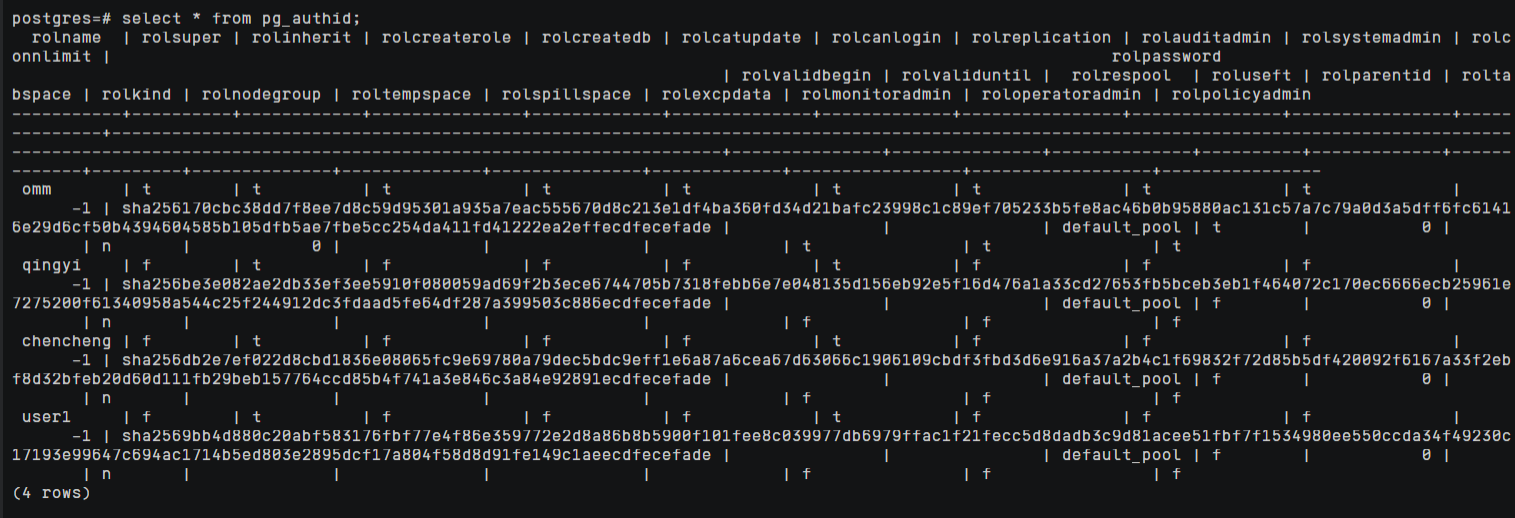


第一次查询时user1用户并没有被授予select的权限，所以报错

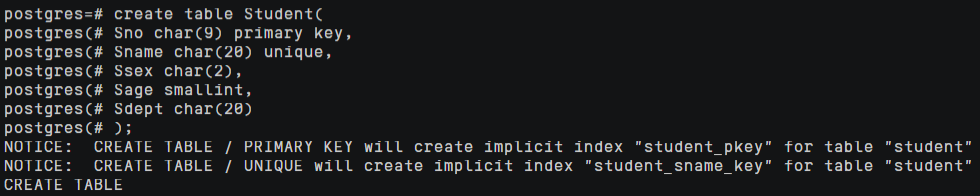
第二次查询时user1拥有了来自于chencheng授予的select权限，因此可以得到查询结果

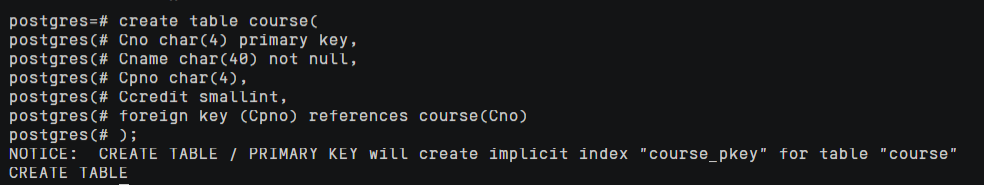
（12）查看与角色、权限相关的系统表和系统视图：pg\_roles，pg\_authid。

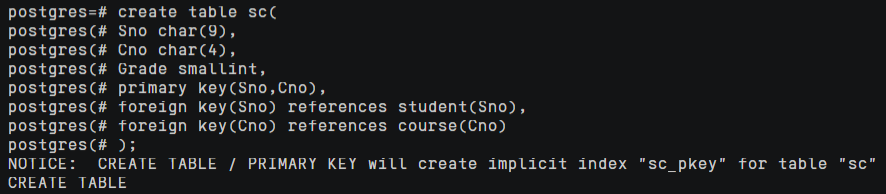


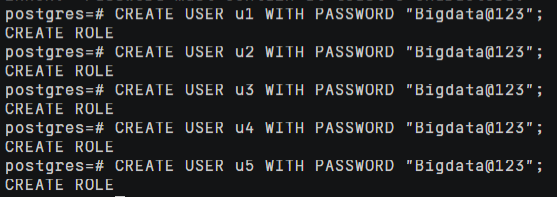


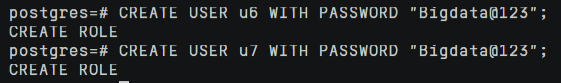
1. 在完成（1）的基础上，重做教材中的[例 4.1-例 4.13]，因为 openGauss 的语法与教材上的不完全一致，可以通过以上实操加深对 openGauss 安全性控制机制的理解。











[例4.1]把查询Student表的权限授给用户U1。



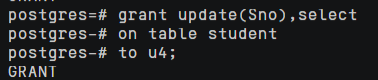
[例4.2]把对 Student表和 Course表的全部操作权限授予用户U2和U3。



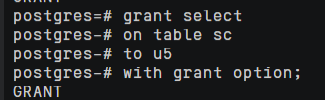
[例4.3] 把对表SC的查询权限授予所有用户。



[例4.4]把查询Student表和修改学生学号的权限授给用户U4。



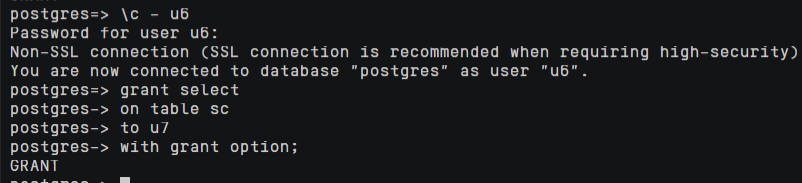
[例4.5]把对表SC的 INSERT权限授予U5用户，并允许将此权限再授予其他用户。



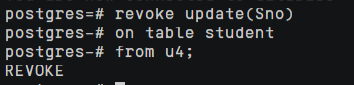
[例4.6]U5将此权限授予U6



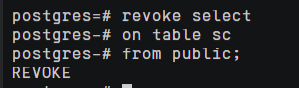
[例4.7]U6还可以将此权限授予U7



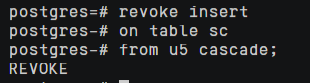
[例4.8]把用户U4修改学生学号的权限收回。



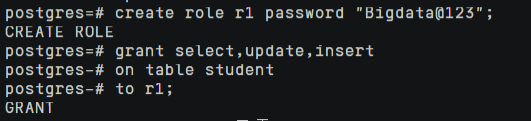
[例4.9]收回所有用户对表SC的查询权限。



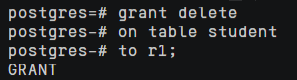
[例4.10]把用户U5对SC表的INSERT权限收回。



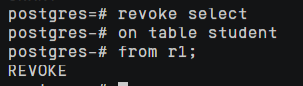
[例4.11]通过角色来实现将一组权限授予一个用户。



[例4.12]角色的权限修改。使角色R1在原来的基础上增加了Student表的DELETE权限。



[例4.13]使R1减少了SELECT权限。



1. 实验总结

3.1 完成的工作

1.创建用户/角色

2.修改用户密码

3.赋予用户/角色权限

4.将用户/角色设置为管理员

5.收回用户/角色的管理员权限

6.删除用户、角色

7.切换用户

8.创建视图

9.在用户拥有/没有查询权限的情况下分别查看视图

3.2 对实验的认识

学会了角色/用户的创建，修改与删除

学会了如何赋予/收回权限

3.3 遇到的困难及解决方法

无