数据库系统实验

实验二:安全性语言实验

16级计科教务2班

16337327

郑映雪

自主存取控制实验

实验目的

掌握自主存取控制权限的定义和维护方法。

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。

实验内容

定义用户、角色,分配权限给用户、角色,回收权限,以相应的用户名登录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,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。下面的实验报告示例,采用实验

方案一。验证权限分配之前,请备份好数据库;针对不同用户所具有的权限,分别设计相应的 SQL 语句加以验证。

实验操作和结果

(由于书上全是 kingbase 的语句,所以我查阅 SQL SERVER 的相同操作的语句)

创建用户

- ①为 David、Tom、Kathy 创建用户标识,要求具有创建用户或角色的权利。
- ②为 Jeffery、Jane、Mike 创建用户。

```
EXEC sp_addlogin 'David','123456','TPCH';

EXEC sp_addlogin 'Tom','123456','TPCH';

EXEC sp_adduser 'Tom','Tom'

EXEC sp_adduser 'David','David'

EXEC sp_adduser 'Cathy','Cathy'

EXEC sp_addlogin 'Mike','123456','TPCH';

EXEC sp_addlogin 'Jeffery','123456','TPCH';

EXEC sp_addlogin 'Jane','123456','TPCH';

EXEC sp_addlogin 'Jane','123456','TPCH';

EXEC sp_adduser 'Mike','Mike';

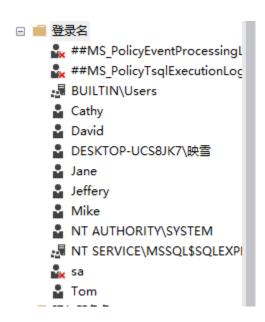
EXEC sp_adduser 'Jeffery','Jeffery';

EXEC sp_adduser 'Jeffery','Jeffery';

EXEC sp_adduser 'Jane','Jane';
```

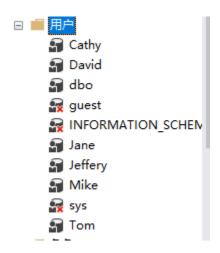
创建成功:

登录名:



数据库系统实验 - 2018 年 10 月 2018 年 2018 年 10 月 2018 年 201

用户:



创建角色并分配权限

①为各个部门分别创建一个查询角色,并分配相应的查询权限。

```
EXEC sp_addrole 'PurchaseQueryRole';

GRANT SELECT ON Part TO PurchaseQueryRole;

GRANT SELECT ON Supplier TO PurchaseQueryRole;

GRANT SELECT ON PartSupp TO PurchaseQueryRole;

EXEC sp_addrole 'SaleQueryRole';

GRANT SELECT ON Orders TO SaleQueryRole;

GRANT SELECT ON Lineitem TO SaleQueryRole;

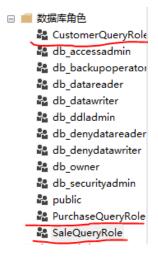
EXEC sp_addrole 'CustomerQueryRole';

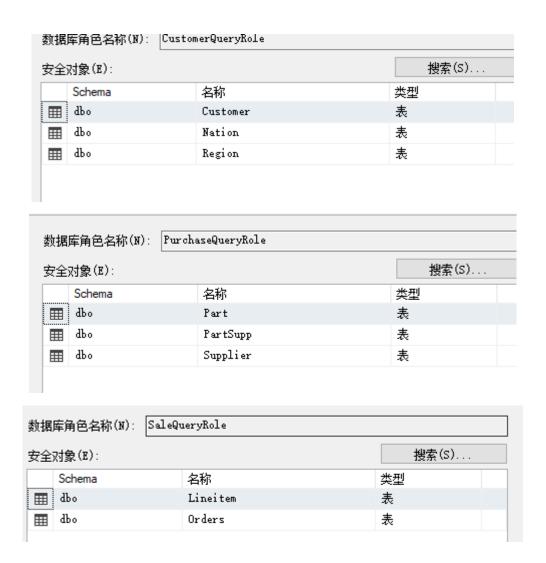
GRANT SELECT ON Customer TO CustomerQueryRole;

GRANT SELECT ON Nation TO CustomerQueryRole;

GRANT SELECT ON Region TO CustomerQueryRole;
```

角色创建成功,结果如下:





②为各部门创建职员角色,对本部门信息具有查看、插入权限。

```
EXEC sp_addrole 'PurchaseEmployeeRole';

GRANT SELECT,INSERT ON Part TO PurchaseEmployeeRole;

GRANT SELECT,INSERT ON Supplier TO PurchaseEmployeeRole;

GRANT SELECT,INSERT ON PartSupp TO PurchaseEmployeeRole;

EXEC sp_addrole 'SaleEmployeeRole';

GRANT SELECT,INSERT ON Orders TO SaleEmployeeRole;

GRANT SELECT,INSERT ON Lineitem TO SaleEmployeeRole;

EXEC sp_addrole 'CustomerEmployeeRole';

GRANT SELECT,INSERT ON Customer TO CustomerEmployeeRole;

GRANT SELECT,INSERT ON Nation TO CustomerEmployeeRole;

GRANT SELECT,INSERT ON Region TO CustomerEmployeeRole;
```

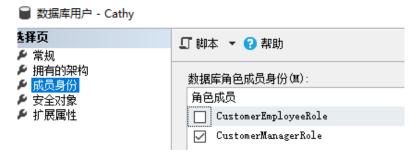
结果如下:

安全对象(E): 搜索(S) IIII dbo Customer 表 IIII dbo Nation 表 IIII dbo Region 表 IIII dbo Region 表 IIII dbo PurchaseEmployeeRole 安全对象(E): 搜索(S) IIII dbo Part Supp 表 IIII dbo Supplier 表 数据库角色名称(N): SaleEmployeeRole 安全对象(E): 搜索(S) Schema 名称 类型 IIII dbo Lineitem 表 IIII dbo Orders 表	数据库角色名称(N): CustomerEmployeeRole			
囲 dbo Customer 表 目 dbo Nation 表 目 dbo Region 表 数据库角色名称(N): PurchaseEmployeeRole 安全対象(E): 搜索(S) Schema 名称 类型	安全对象(E):		搜索(s)	
囲 dbo Customer 表 田 dbo Nation 表 田 dbo Region 表 数据库角色名称(N): PurchaseEmployeeRole 安全対象(E): 捜索(S) Schema 名称 类型 dbo Part 表 田 dbo PartSupp 表 田 dbo Supplier 表 数据库角色名称(N): SaleEmployeeRole 安全対象(E): 捜索(S) 数据库角色名称(N): SaleEmployeeRole 安全対象(E): 捜索(S)		名称	类型	
数据库角色名称(N): PurchaseEmployeeRole 安全对象(E): 搜索(S) Schema 名称 类型 dbo Part 表	1	Customer	表	
数据库角色名称(N): PurchaseEmployeeRole 安全对象(E): 搜索(S) Schema 名称 类型 曲 dbo Part 表 曲 dbo PartSupp 表 曲 dbo Supplier 表 数据库角色名称(N): SaleEmployeeRole 安全对象(E): 搜索(S) Schema 名称 类型 曲 dbo Lineitem 表	⊞ dbo	Nation	表	
安全对象(E): 搜索(S) Schema 名称 类型 田 dbo Part Supp 表 田 dbo Supplier 表 数据库角色名称(N): SaleEmployeeRole 安全对象(E): 搜索(S) Schema 名称 类型 田 dbo Lineitem 表	⊞ dbo	Region	表	
Schema 名称 类型	数据库角色名称(N): PurchaseEmployeeRole			
田 dbo Part 表 田 dbo PartSupp 表 田 dbo Supplier 表 数据库角色名称(N): SaleEmployeeRole 安全对象(E): 搜索(S) Schema 名称 类型 田 dbo Lineitem 表	安全对象(E):		搜索(s)	
世 dbo PartSupp 表 由 dbo Supplier 表 数据库角色名称(N): SaleEmployeeRole 安全对象(E): 搜索(S) Schema 名称 类型 由 dbo Lineitem 表	Schema	名称	类型	
数据库角色名称(N): SaleEmployeeRole 安全对象(E): 搜索(S) Schema 名称 类型 dbo Lineitem 表	⊞ dbo	Part	表	
数据库角色名称(N): SaleEmployeeRole 安全对象(E): 搜索(S) Schema 名称 类型 面 dbo Lineitem 表	⊞ dbo	PartSupp	表	
安全对象(E): 搜索(S) Schema 名称 类型 面 dbo Lineitem 表	⊞ dbo	Supplier	表	
安全对象(E): 搜索(S) Schema 名称 类型 面 dbo Lineitem 表				
Schema 名称 类型 面 dbo Lineitem 表	数据库角色名称(N): SaleEmployeeRole			
■ dbo Lineitem 表	安全对象(图):		搜索(s)	
	Schema	名称	类型	
Ⅲ dbo Orders 表	⊞ dbo	Lineitem	表	
	⊞ dbo	Orders	表	

③为各部门创建一个经理角色,相应角色对本部门的信息具有完全控制权限,对其它 部门的信息具有查询权。经理有权给本部门职员分配权限。

```
EXEC sp_addrolemember 'PurchaseManagerRole','David';
EXEC sp_addrolemember 'SaleManagerRole','Tom';
EXEC sp_addrolemember 'CustomerManagerRole','Cathy';
EXEC sp_addrolemember 'PurchaseEmployeeRole','Jeffery';
EXEC sp_addrolemember 'SaleEmployeeRole','Jane';
EXEC sp_addrolemember 'CustomerEmployeeRole','Mike';
```

权限分配结果如下:



■ 数据库用户 - David	
桂择页	□ 脚本 ▼ ② 帮助
▶ 常规▶ 拥有的架构	数据库角色成员身份(M):
成员身份 完全計算	角色成员
▶️ 安全对象 ▶️ 扩展属性	CustomerEmployeeRole
	CustomerManagerRole
	CustomerQueryRole
	db accessadmin
	db_backupoperator
	db_datareader
	db_datawriter
	db_ddladmin
	db_denydatareader
接	db_denydatawriter
 服务器:	db_owner
DESKTOP-UCS8JK7\SQLEXPRESS	db_securityadmin
连接:	PurchaseEmployeeRole
DESKTOP-UCS8JK7\映雪	✓ PurchaseManagerRole
γ₩ 查看连接属性	PurchaseQueryRole
■ 数据库用户 - Tom Analysis Services MDX 查试 择页 ■ 常规 ■ 拥有的架构 ■ <mark>成员身份</mark> ■ 安全对象 ■ 扩展属性	」 即本 ▼ ② 帮助 数据库角色成员身份(M): 角色成员 CustomerEmployeeRole
	CustomerManagerRole
	CustomerQueryRole
	db_accessadmin
	db_backupoperator
	☐ db_datareader
	☐ db_datawriter
	☐ db_ddladmin
L 13.	☐ db_denydatareader
接	☐ db_denydatawriter
服务器: DESKTOP-UCS8JK7\SQLEXPRESS	db_owner
连接:	☐ db_securityadmin☐ PurchaseEmployeeRole
DESKTOP-UCS8JK7\映雪	PurchaseManagerRole
y ₩ 查看连接属性	PurchaseQueryRole
	☐ SaleEmployeeRole
ě szás	✓ SaleManagerRole
<u>±度</u>	SaleQueryRole
就绪	

数据库系统实验 - 2018年 10月

6

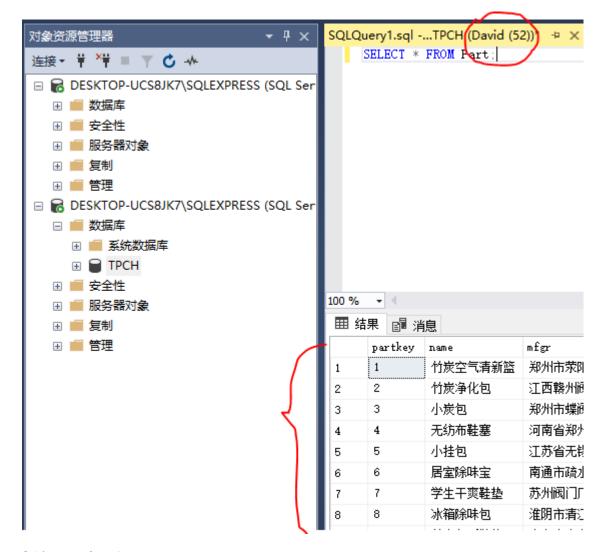
ѝ 数据库用户 - Jane 释页 □ 脚本 🔻 🕝 帮助 常规 拥有的架构 数据库角色成员身份(M): 成员身份 角色成员 安全对象 CustomerManagerRole 扩展属性 ☐ CustomerQueryRole ☐ db_accessadmin db_backupoperator db_datareader ☐ db_datawriter db_ddladmin ☐ db_denydatareader ☐ db_denydatawriter db_owner {务器: :SKTOP-UCS8JK7\SQLEXPRESS ☐ db_securityadmin ☐ PurchaseEmployeeRole :接: !SKTOP-VCS8JK7\映雪 PurchaseManagerRole ☐ PurchaseQueryRole ₩ 查看连接属性 ✓ SaleEmployeeRole SaleManagerRole ■ 数据库用户 - Jeffery 选择页 ☑ 脚本 🔻 🕝 帮助 ♪ 常规 🔑 拥有的架构 数据库角色成员身份(M): 成员身份安全対象 角色成员 ▶ 扩展属性 CustomerManagerRole ☐ CustomerQueryRole db_accessadmin db_backupoperator db_datareader ☐ db_datawriter ☐ db_ddladmin ☐ db_denydatareader ☐ db_denydatawriter 连接 db_owner 服务器: DESKTOP-UCS8JK7\SQLEXPRESS db_securityadmin PurchaseEmployeeRole 连接: ■ 数据库用户 - Mike 🖵 脚本 🔻 😯 帮助 🔑 常规 🔑 拥有的架构 数据库角色成员身份(M): 🔑 成员身份 角色成员 ▶️ 安全对象 🔑 扩展属性 CustomerEmployeeRole CustomerManagerRole

测试分配权限的效果

①登录 David, 输入查询语句。

SELECT * FROM Part;

查询结果如下,可以看到通过 David 的登录是可以成功查询的。



②输入删除语句。

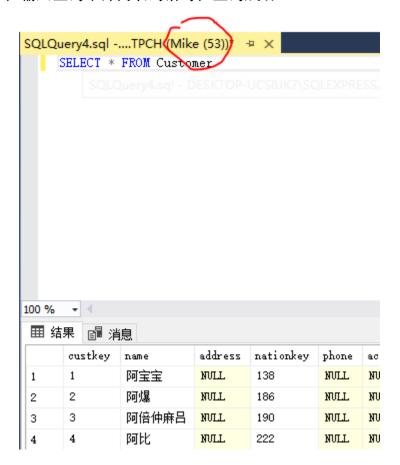
DELETE FROM Orders WHERE orderkey=1;

结果如下,没有权限。因为当初授权的时候只授予了插入和查找权限,没有授予删除。

数据库系统实验 - 2018 年 10 月

8

③登录 Mike,输入查询本部门表的语句,查询成功:



但是查询别的部门的表时显示:



9

说明 Mike 的权限也授权成功了,只能查询本部门的几个表。

收回权限

收回对 Mike 对 Customer 的查询权限:

REVOKE SELECT ON Customer FROM Mike;

再登录 Mike 进行上一步中的查询:

```
100 % ▼ ◀ ■ 消息 | 消息 | 消息 | 消息 | 229, 级别 | 14, 状态 | 5, 第 1 行 | 拒绝了对对象 'Customer' (数据库 'TPCH', 架构 'dbo')的 SELECT 权限。
```

由图可见验证成功。

实验总结

这个实验真的是一波三折……由于语句和理论书、实验书大都不一样,所以在查阅资料上花费了挺大一番功夫。听说使用 MySQL 的同学遇到了更大的麻烦,这样对比一下我当初选择使用 SQL Server 起码有与书上要求差不多的操作。

首先,书上明确说了这一节的语句为 KINGBASE 中的语句,不管是使用 sql server 还是 mysql 的同学都不能照搬书上的语句做实验了(其实这样也挺好的,可以在查资料的过程中对自己使用的 DBMS 的语句有一个熟悉的过程)。

在进行实验的过程中,主要出现了以下的一些困难:

①登录名和用户名的问题。在实验书上,只设置了用户名。而 SQL Server 用户名和登录名是独立分开的。需要用 EXEC sp_addlogin 进行设置,为了方便加上不混淆,我将用户名与登录名设置一致。

②语句差异。这个实验中的语句"不兼容性"更强了……不仅在语句上的表达不同,有些操作本身在 SQL Server 里也是不支持的(例如早已经不再被 SQL Server 推荐使用的 delete all),所以我将实验里的一些语句替换了,只力求实现本身要实现的效果。

③SQL Server 本身的操作不够娴熟。在更换登录用户的时候,由于我本地数据库我没有设置兼容性登录,所以一直登录不上 SQL 账户,我还一直以为是我代码写得有问题。后来在逐步排错之后才发现是我数据库本身没有设置好。

除了以上三个方面的问题,本次实验本身的难度还是不高的。在授权、收回权限的前后进行权限的测试是验证授权、收权是否成功的重要一步。通过这次实验,我对权限管理有了一个更深的认识,巩固了理论课上的内容。