广州大学学生实验报告

开课实验室： 电子楼 418　　2019 年 5 月 16 日

学院	计算机科学与网络工程学院	年级、专业、班	软件 171	姓名	谢金宏	学号	1706300001
实验课程名称	数据库原理					成绩	
实验项目名称	实验四 用户权限管理					指导老师	张少宏

教师评语：

一、实验目的

1. 对 ORACLE 数据库系统的用户权限管理有感性认识。

二、实验环境

安装有 Oracle 11g 数据库软件的远程计算机和安装有 SQL Developer 软件的本地计算机。

三、实验内容

1. 进行理论学习，包括权限分类、系统权限管理、实体权限管理等内容的相关知识。
2. 进行正式操作，详见“实验步骤”小节。

四、实验步骤

本次实验中涉及的 SQL 语句：

代码.sql

1. 以 SYSTEM 身份连接到数据库，创建新的用户 Alice 并授权。

```
CREATE USER alice IDENTIFIED BY "alicepwd";
GRANT CONNECT, RESOURCE, DBA TO alice;
```

2. 以用户 Alice 的身份建立连接，并在此连接下执行后面的操作。

```
SELECT * from user_role_privs;
```

查询结果 ×

SQL | 提取的所有行：3，用时 0.03 秒

	USERNAME	GRANTED_ROLE	ADMIN_OPTION	DEFAULT_ROLE	OS_GRANTED
1	ALICE	CONNECT	NO	YES	NO
2	ALICE	DBA	NO	YES	NO
3	ALICE	RESOURCE	NO	YES	NO

图表 1 Alice 具有 Connect, DBA, Resource 三种权限

3. 如有必要，清理旧的数据表。
4. 建立表格及输入数据。

```
CREATE TABLE READER
(
    RNO VARCHAR2(4 CHAR) PRIMARY KEY,
    RNAME    VARCHAR2(10 CHAR) NOT NULL,
    RSEX     VARCHAR2(1 CHAR),
    RAGE     INTEGER,
    RBOSS    VARCHAR2(10 CHAR),
    RADDRESS     VARCHAR2(30 CHAR)
) ;
INSERT INTO READER (RNO,RNAME,RAGE,RSEX,RBOSS, RADDRESS) VALUES('R001','张三',20,'男','
李四','416');

CREATE TABLE BOOK
(
    BNO VARCHAR2(4 CHAR),
    BNAME    VARCHAR2(50 CHAR) NOT NULL,
    BAUTHOR VARCHAR2(50 CHAR),
    BPRESS   VARCHAR2(50 CHAR),
    BPRICE   NUMERIC(6,2),
    PRIMARY KEY(BNO)
);
INSERT INTO BOOK (BNO,BAUTHOR,BNAME, BPRESS, BPRICE) VALUES('B001','严蔚敏','数据结构','
清华大学出版社',NULL);
```

```
CREATE TABLE RB
(
    RNO VARCHAR2(4 CHAR),
    BNO VARCHAR2(4 CHAR),
    RBDATE DATE DEFAULT SYSDATE,
    PRIMARY KEY(RNO,BNO),
    FOREIGN KEY (RNO) REFERENCES READER(RNO),
    FOREIGN KEY (BNO) REFERENCES BOOK(BNO)
);
INSERT INTO RB (RNO,BNO) VALUES ('R001','B001');
```

5. 确认 orcl 数据库中有这三个数据表，以及相应的数据。
6. 查询用户 CC 的权限信息。

```
SELECT * FROM dba_role_privs;
SELECT * FROM dba_sys_privs;
SELECT * FROM role_sys_privs;
```

脚本输出 × 查询结果 ×

SQL | 提取的所有行：78，用时 0.243 秒

	GRANTEE	GRANTED_ROLE	ADMIN_OPTION	DEFAULT_ROLE
1	SYS	XDB_SET_INVOKER	YES	YES
2	SYS	XDBADMIN	YES	YES
3	SYS	IMP_FULL_DATABASE	YES	YES
4	DBA	SCHEDULER_ADMIN	YES	YES
5	DBA	DATAPUMP_IMP_FULL_DATABASE	NO	YES
6	SYSTEM	AQ_ADMINISTRATOR_ROLE	YES	YES
7	EXECUTE_CATALOG_ROLE	HS_ADMIN_EXECUTE_ROLE	NO	YES
8	HS_ADMIN_ROLE	HS_ADMIN_EXECUTE_ROLE	NO	YES
9	OEM_MONITOR	SELECT_CATALOG_ROLE	NO	YES
10	APEX_040000	RESOURCE	YES	YES
11	ALICE	CONNECT	NO	YES
12	SYS	APEX_ADMINISTRATOR_ROLE	YES	YES
13	SYS	RECOVERY_CATALOG_OWNER	YES	YES
14	SYS	DELETE_CATALOG_ROLE	YES	YES
15	DBA	DELETE_CATALOG_ROLE	YES	YES
16	DBA	EXECUTE_CATALOG_ROLE	YES	YES
17	HR	RESOURCE	NO	YES

图表 2 SELECT * FROM dba_role_privs;

脚本输出 × 查询结果 ×

SQL | 已提取 50 行，用时 0.076 秒

	GRANTEE	PRIVILEGE	ADMIN_OPTION
1	DBA	CREATE SESSION	YES
2	DBA	ALTER SESSION	YES
3	DBA	DROP TABLESPACE	YES
4	DBA	BECOME USER	YES
5	DBA	DROP ROLLBACK SEGMENT	YES
6	DBA	SELECT ANY TABLE	YES
7	DBA	INSERT ANY TABLE	YES
8	DBA	UPDATE ANY TABLE	YES
9	DBA	DROP ANY INDEX	YES
10	DBA	SELECT ANY SEQUENCE	YES
11	DBA	CREATE ROLE	YES
12	DBA	EXECUTE ANY PROCEDURE	YES
13	DBA	ALTER PROFILE	YES
14	DBA	CREATE ANY DIRECTORY	YES
15	DBA	CREATE ANY LIBRARY	YES
16	DBA	EXECUTE ANY LIBRARY	YES
17	DBA	ALTER ANY INDEXTYPE	YES

图表 3 SELECT * FROM dba_sys_privs;

脚本输出 × 查询结果 ×

SQL | 已提取 50 行，用时 0.141 秒

	ROLE	PRIVILEGE	ADMIN_OPTION
1	DBA	CREATE SESSION	YES
2	DBA	ALTER SESSION	YES
3	DBA	DROP TABLESPACE	YES
4	DBA	BECOME USER	YES
5	DBA	DROP ROLLBACK SEGMENT	YES
6	DBA	SELECT ANY TABLE	YES
7	DBA	INSERT ANY TABLE	YES
8	DBA	UPDATE ANY TABLE	YES
9	DBA	DROP ANY INDEX	YES
10	DBA	SELECT ANY SEQUENCE	YES
11	DBA	CREATE ROLE	YES
12	DBA	EXECUTE ANY PROCEDURE	YES
13	DBA	ALTER PROFILE	YES
14	DBA	CREATE ANY DIRECTORY	YES
15	DBA	CREATE ANY LIBRARY	YES
16	DBA	EXECUTE ANY LIBRARY	YES
17	DBA	ALTER ANY INDEXTYPE	YES
18	DBA	DROP ANY INDEXTYPE	YES

图表 4 SELECT * FROM role_sys_privs;

7. 查询用户创建的表。

```
SELECT table_name FROM user_tables;
```

脚本输出 × 查询结果 ×

SQL | 提取的所有行：3，用时 0.157 秒

	TABLE_NAME
1	READER
2	BOOK
3	RB

8. 尝试删除 RB 表。

```
DROP TABLE RB;
SELECT table_name FROM user_tables;
```

删除成功。

9. 以 SYSTEM 的身份连接，回收 Alice 的 DBA 和 Resource 权限。

```
REVOKE dba, RESOURCE FROM alice;
```

10. 尝试使用 Alice 创建数据表。

```
CREATE TABLE aa(cola INT);
```

执行出错，提示“ORA-01950: 对表空间 'SYSTEM' 无权限”。
11. SYS：将 Resource 权限授予 Alice。

```
GRANT RESOURCE TO alice;
```

12. 查询用户 Alice 的权限并尝试创建表。

```
SELECT * FROM user_role_privs;
CREATE TABLE aa(cola INT);
```

Alice 具有 Connect 和 Resource 两种权限，创建表成功。

13. 使用 SYS 尝试删除用户 Alice。

```
DROP USER alice;
```

无法删除，提示“ORA-01940: 无法删除当前连接的用户”。

14. 断开 Alice 链接，再次使用 SYS 尝试删除 Alice。
无法删除，提示“ORA-01922: 必须指定 CASCADE 以删除 'ALICE'”。
需要使用 CASCADE 选项将 Alice 所创建的数据库对象也删除之后才能删除 Alice 用户。

```
DROP USER alice CASCADE;
```

删除 Alice 用户之后就不能以 Alice 身份连接了。

15. 使用 SYS 创建用户 Alice 和 Bob。

```
CREATE USER alice IDENTIFIED BY alicepwd;
CREATE USER bob IDENTIFIED BY bobpwd;
GRANT RESOURCE, CONNECT, DBA TO alice, bob;

-- 只有 DBA 才能执行下面这条查询
SELECT * FROM dba_role_privs WHERE grantee IN ('ALICE','BOB');
```

	GRANTEE	GRANTED_ROLE	ADMIN_OPTION	DEFAULT_ROLE
1	ALICE	CONNECT	NO	YES
2	ALICE	DBA	NO	YES
3	ALICE	RESOURCE	NO	YES
4	BOB	RESOURCE	NO	YES
5	BOB	CONNECT	NO	YES
6	BOB	DBA	NO	YES

16. 分别以 Alice 和 Bob 身份执行下列操作。

```sql
-- Alice
CREATE TABLE from_alice(message CHAR(1));
INSERT INTO from_alice VALUES('a');
SELECT * FROM from_alice;

-- Bob

-- 显示 FROM_ALICE 表的创建者和表名

SELECT OWNER, table_name FROM all_tables WHERE table_name='FROM_ALICE';

-- 此时可见 alice 创建的表 from_alice


-- 尝试查询 FROM_ALICE 表中的数据

SELECT * FROM from_alice;  -- ORA-00942: 表或视图不存在


-- Alice
GRANT ALL ON from_alice TO bob;

-- Bob

-- 显示 FROM_ALICE 表的创建者和表名

SELECT OWNER, table_name FROM all_tables WHERE table_name='FROM_ALICE';

-- 可以查询到 from_alice 表


SELECT * FROM from_alice;  -- ORA-00942: 表或视图不存在

SELECT * FROM alice.from_alice;  -- 显示 Alice 在表中插入的信息‘a’

INSERT INTO alice.from_alice VALUES('d');

SELECT * FROM alice.from_alice;  -- 验证上面的操作插入成功
```

17. 分别以 Alice 或 Bob 身份执行下列操作

```sql
-- Alice
REVOKE INSERT ON from_alice FROM bob;

-- Bob
SELECT * FROM alice.from_alice;

-- 可以显示表中的内容


INSERT INTO alice.from_alice VALUES('y');
SELECT * FROM alice.from_alice;

-- 插入成功


-- SYS
REVOKE DBA, RESOURCE FROM bob;

-- Bob
SELECT * FROM alice.from_alice;

-- 能显示表中的内容


INSERT INTO alice.from_alice VALUES('z');
SELECT * FROM alice.from_alice;

-- 插入成功


-- Alice
REVOKE ALL ON from_alice FROM bob;
SELECT * FROM alice.from_alice;

-- 没有显示前面成功插入的记录'd'、'y'、'z'


-- Bob
SELECT * FROM alice.from_alice;

-- 显示前面成功插入的记录'd'、'y'、'z'


commit;  -- 成功提交


-- Alice
SELECT * FROM alice.from_alice;

-- 显示前面成功插入的记录'd'、'y'、'z'
```

18. 重新连接 Bob

```
-- Bob
SELECT * FROM alice.from_alice;
-- ORA-00942: table or view does not exist

SELECT * FROM user_role_privs;

-- 当前用户只有 Connect 权限


CREATE TABLE from_bob(message CHAR(1 CHAR));

-- 不能执行

-- ORA-01031: insufficient privileges

-- SYS
GRANT RESOURCE TO bob;


-- Bob (不重新连接 Bob)

CREATE TABLE from_bob(message CHAR(1 CHAR));

-- 不能执行

-- ORA-01031: insufficient privileges


-- 重新连接 Bob

CREATE TABLE from_bob(message CHAR(1 CHAR));

INSERT INTO from_bob VALUES('甲');

SELECT * FROM from_bob;  -- 可以查询到记录‘甲’


-- Alice
SELECT * FROM bob.from_bob;

-- 不能看到记录‘甲’
```

Bob 需要对 Alice 进行以下授权，才能使 Alice 对数据表有读权限或写权限。

```
-- 允许 Alice 查看 from_bob 表中的数据

GRANT SELECT ON from_bob TO alice;


-- 允许 Alice 插入、修改和删除 from_bob 表中的数据

GRANT INSERT, DELETE, UPDATE ON from_bob TO alice;
```

五、分析总结

完成了全部实验内容。
通过本次实验，我获悉了 Oracle 数据库对象权限管理的冰山一角。
正如许多其他数据库软件一样，Oracle 数据库会在用户连接时在内存中缓存用户的 Permissions。当一个持有连接的用户被赋予更多权限或者撤销已有的权限时，数据库的 Permissions 不会实时更新，用户仍然以旧的权限模式或角色访问和操作数据库；只有用户执行重新连接等 flush privileges 操作后，用户的权限模式或角色才会生效。
用户修改其他用户所创建的数据对象是，对数据对象的修改并不是实时的，而是隐式地启用了事务；只有当用户使用 COMMIT 提交事务后其他用户数据对象的修改才对其他用户可见（参考实验过程中的第 17 小节）。此处关于事务的内容是实验指导书所没有提到的。