Oracle 数据库权限、角色和用户总结

前言：

ORACLE 数据库系统预先定义了 CONNECT 、RESOURCE、 DBA、 EXP_FULL_DATABASE、IMP_FULL_DATABASE 五个角色。
CONNECT 具有创建表、视图、序列等特权；(alter session create cluster)
RESOURCE 具有创建过程、触发器、表、序列等特权、
DBA 具有全部系统特权；
EXP_FULL_DATABASE、 IMP_FULL_DATABASE 具有卸出与装入数据库的特权。

权限管理

一、权限分类

系统权限：系统规定用户使用数据库的权限。（系统权限是对用户而言)。

实体权限：某种权限用户对其它用户的表或视图的存取权限。（是针对表或视图而言的）。

二、系统权限管理：

1、系统权限分类：
DBA: 拥有全部特权，是系统最高权限，只有 DBA 才可以创建数据库结构。

RESOURCE:拥有 Resource 权限的用户只可以创建实体，不可以创建数据库结构。

CONNECT:拥有 Connect 权限的用户只可以登录 Oracle，不可以创建实体，不可以创建数据库结构。

对于普通用户：授予 connect, resource 权限。
对于 DBA 管理用户：授予 connect，resource, dba 权限。

2、系统权限授权命令：
[系统权限只能由 DBA 用户授出：sys, system(最开始只能是这两个用户)]
授权命令：SQL> grant connect, resource, dba to 用户名 1 [,用户名 2]...;
[普通用户通过授权可以具有与 system 相同的用户权限，但永远不能达到与 sys 用户相同的权限，system 用户的权限也可以被回收。]
例：

SQL> connect system/manager
SQL> Create user user50 identified by user50;
SQL> grant connect, resource to user50;
查询用户拥有哪些权限：
SQL> select * from dba_role_privs;
SQL> select * from dba_sys_privs;
SQL> select * from role_sys_privs;
删除用户：SQL> drop user 用户名 cascade; //加上 cascade 则将用户连同其创建的东西全部删除

3、系统权限传递：
增加 WITH ADMIN OPTION 选项，则得到的权限可以传递。
SQL> grant connect, resorce to user50 with admin option; //可以传递所获权限。

4、系统权限回收：系统权限只能由 DBA 用户回收
命令：SQL> Revoke connect, resource from user50;
说明：
1）如果使用 WITH ADMIN OPTION 为某个用户授予系统权限，那么对于被这个用户授予相同权限的所有用户来说，取消该用户的系统权限并不会级联取

消这些用户的相同权限。
2）系统权限无级联，即 A 授予 B 权限，B 授予 C 权限，如果 A 收回 B 的权限，C 的权限不受影响；系统权限可以跨用户回收，即 A 可以直接收回 C 用户的权限

。

三、实体权限管理(对象权限)

1、实体权限分类：select, update, insert, alter, index, delete, all //all 包括所有权限
execute //执行存储过程权限
user01:
SQL> grant select, update, insert on product to user02;
SQL> grant all on product to user02;
user02:
SQL> select * from user01.product;
// 此时 user02 查 user_tables，不包括 user01.product 这个表，但如果查 all_tables 则可以查到，因为他可以访问。

2. 将表的操作权限授予全体用户：
SQL> grant all on product to public; // public 表示是所有的用户，这里的 all 权限不包括 drop。
[实体权限数据字典]:
SQL> select owner, table_name from all_tables; // 用户可以查询的表
SQL> select table_name from user_tables; // 用户创建的表
SQL> select grantor, table_schema, table_name, privilege from all_tab_privs; // 获权可以存取

的表（被授权的）
SQL> select grantee, owner, table_name, privilege from user_tab_privs; // 授出权限的表(授出的权限)

查看用户拥有哪些系统权利
Sql> select grantee， privilege from dba_sys_privs where grantee=‘SCOTT’;
查看用户拥有哪些对象权利
Sql> select grantee，privilege，owner，table_name from dba_tab_privs where grantee=‘SCOTT’;

3. DBA 用户可以操作全体用户的任意基表(无需授权，包括删除)：
DBA 用户：
SQL> Create table stud02.product(
id number(10),
name varchar2(20));
SQL> drop table stud02.emp;
SQL> create table stud02.employee
as
select * from scott.emp;

4. 实体权限传递(with grant option)：
user01:
SQL> grant select, update on product to user02 with grant option; // user02 得到权限，并可以传递。

5. 实体权限回收：
user01:
SQL>Revoke select, update on product from user02; //传递的权限将全部丢失。
说明
1）如果取消某个用户的对象权限，那么对于这个用户使用 WITH GRANT OPTION 授予权限的用户来说，同样还会取消这些用户的相同权限，也就是说取

消授权时级联的。

Oracle 用户管理

一、创建用户的 Profile 文件
SQL> create profile student limit // student 为资源文件名
FAILED_LOGIN_ATTEMPTS 3 //指定锁定用户的登录失败次数
PASSWORD_LOCK_TIME 5 //指定用户被锁定天数
PASSWORD_LIFE_TIME 30 //指定口令可用天数

二、创建用户

SQL> Create User username
Identified by password
Default Tablespace tablespace_name
Temporary Tablespace tablespace_name
Profile profile_name
Quota integer/unlimited on tablespace;
例:
SQL> Create user acc01
identified by acc01　　// 如果密码是数字，请用双引号括起来
default tablespace account
temporary tablespace temp
profile default
quota 50m on account;
SQL> grant connect, resource to acc01;
查询用户缺省表空间、临时表空间
SQL> select username, default_tablespace, temporary_tablespace from dba_users;
查询系统资源文件名：
SQL> select * from dba_profiles;
资源文件类似表，一旦创建就会保存在数据库中。
SQL> select username, profile, default_tablespace, temporary_tablespace from dba_users;
SQL> create profile common limit
failed_login_attempts 5
idle_time 5;
SQL> Alter user acc01 profile common;

三、修改用户：
SQL> Alter User 用户名
Identified 口令
Default Tablespace tablespace
Temporary Tablespace tablespace
Profile profile
Quota integer/unlimited on tablespace;
1、修改口令字：
SQL>Alter user acc01 identified by "12345";
2、修改用户缺省表空间：
SQL> Alter user acc01 default tablespace users;
3、修改用户临时表空间
SQL> Alter user acc01 temporary tablespace temp_data;
4、强制用户修改口令字：
SQL> Alter user acc01 password expire;
5、将用户加锁
SQL> Alter user acc01 account lock;　// 加锁
SQL> Alter user acc01 account unlock;　// 解锁

四、删除用户

SQL>drop user 用户名; //用户没有建任何实体

SQL> drop user 用户名 CASCADE; // 将用户及其所建实体全部删除

*1. 当前正连接的用户不得删除。

五、监视用户：

1、查询用户会话信息：

SQL> select username, sid, serial#, machine from v$session;

2、删除用户会话信息：

SQL> Alter system kill session 'sid, serial#';

3、查询用户 SQL 语句：

SQL> select user_name, sql_text from v$open_cursor;

Oracle 角色管理

一、何为角色？

我在前面的篇幅中说明权限和用户。慢慢的在使用中你会发现一个问题：如果有一组人，他们的所需的权限是一样的，当对他们的权限进行管理

的 时候会很不方便。因为你要对这组中的每个用户的权限都进行管理。

有一个很好的解决办法就是：角色。角色是一组权限的集合，将角色赋给一个用户，这个用户就拥有了这个角色中的所有权限。那么上述问题就

很 好处理了，只要第一次将角色赋给这一组用户，接下来就只要针对角色进行管理就可以了。

以上是角色的一个典型用途。其实，只要明白：角色就是一组权限的集合。下面分两个部门来对 oracle 角色进行说明。

二、系统预定义角色

预定义角色是在数据库安装后，系统自动创建的一些常用的角色。下介简单的介绍一下这些预定角色。角色所包含的权限可以用以下语句查询：

sql>select * from role_sys_privs where role='角色名';

1．CONNECT, RESOURCE, DBA

这些预定义角色主要是为了向后兼容。其主要是用于数据库管理。oracle 建议用户自己设计数据库管理和安全的权限规划，而不要简单的使用这些预

定角色。将来的版本中这些角色可能不会作为预定义角色。

2．DELETE_CATALOG_ROLE， EXECUTE_CATALOG_ROLE， SELECT_CATALOG_ROLE

这些角色主要用于访问数据字典视图和包。

3．EXP_FULL_DATABASE， IMP_FULL_DATABASE

这两个角色用于数据导入导出工具的使用。

4．AQ_USER_ROLE， AQ_ADMINISTRATOR_ROLE
AQ:Advanced Query。这两个角色用于 oracle 高级查询功能。
5． SNMPAGENT
用于 oracle enterprise manager 和 Intelligent Agent
6．RECOVERY_CATALOG_OWNER
用于创建拥有恢复库的用户。关于恢复库的信息，参考 oracle 文档《Oracle9i User-Managed Backup and Recovery Guide》
7．HS_ADMIN_ROLE
A DBA using Oracle's heterogeneous services feature needs this role to access appropriate tables in the data dictionary.

三、管理角色
1.建一个角色
sql>create role role1;
2.授权给角色
sql>grant create any table,create procedure to role1;
Sql>grant create session，create table to role1;
Sql>revoke create session，create table from role1;

3.授予角色给用户
sql>grant role1 to user1;
Sql>grant role1 to user1 with admin option;

revoke role1 from user1;

查看系统中所有的角色
Sql>select * from dba_roles;

查看用户被授予了哪些角色
Sql>select grantee，granted_role from dba_role_privs where grantee=‘SCOTT’;

查看角色中包含了哪些系统权限
Sql>select role，privilege from role_sys_privs where role=‘ROLE1’;
查看角色中包含了哪些对象权限
Sql>select role，privilege，from role_tab_privs where role=‘ROLE1’;

将角色授予角色
Sql>grant role1 to role2;

4.查看角色所包含的权限
sql>select * from role_sys_privs;
5.创建带有口令以角色(在生效带有口令的角色时必须提供口令)
sql>create role role1 identified by password1;
6.修改角色：是否需要口令

sql>alter role role1 not identified;
sql>alter role role1 identified by password1;
7.设置当前用户要生效的角色
(注：角色的生效是一个什么概念呢？假设用户 a 有 b1,b2,b3 三个角色，那么如果 b1 未生效，则 b1 所包含的权限对于 a 来讲是不拥有的，只有角色生效

了，角色内的权限才作用于用户，最大可生效角色数由参数 MAX_ENABLED_ROLES 设定；在用户登录后，oracle 将所有直接赋给用户的权限和用户默认

角色中的权限赋给用户。）
sql>set role role1;//使 role1 生效
sql>set role role,role2;//使 role1,role2 生效
sql>set role role1 identified by password1;//使用带有口令的 role1 生效
sql>set role all;//使用该用户的所有角色生效
sql>set role none;//设置所有角色失效
sql>set role all except role1;//除 role1 外的该用户的所有其它角色生效。
sql>select * from SESSION_ROLES;//查看当前用户的生效的角色。
8.修改指定用户，设置其默认角色
sql>alter user user1 default role role1;
sql>alter user user1 default role all except role1;
详见 oracle 参考文档
9.删除角色
sql>drop role role1;
角色删除后，原来拥用该角色的用户就不再拥有该角色了，相应的权限也就没有了。
说明:
1)无法使用 WITH GRANT OPTION 为角色授予对象权限
2)可以使用 WITH ADMIN OPTION 为角色授予系统权限,取消时不是级联

查看角色中还包含哪些角色
Sql>select role，granted_role from role_role_privs where role=‘DBA’;
备注：授予用户 DBA、RESOURCE 这俩个角色后系统会自动再授予用户 unlimited tablespace