****Oracle 权限****  
权限允许用户访问属于其它用户的对象或执行程序，ORACLE系统提供三种权限：Object 对象级、System 系统级、Role 角色级。这些权限可以授予给用户、特殊用户public或角色，如果授予一个权限给特殊用户"Public"（用户public是oracle预定义的，每个用户享有这个用户享有的权限）,那么就意味作将该权限授予了该数据库的所有用户。  
对管理权限而言，角色是一个工具，权限能够被授予给一个角色，角色也能被授予给另一个角色或用户。用户可以通过角色继承权限，除了管理权限外角色服务没有其它目的。权限可以被授予，也可以用同样的方式撤销。  
权限分类  
1、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)。  
2、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）。  
****系统权限管理****   
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。  
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]...;  
注: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;

查询用户拥有哪里权限：   
SQL> select \* from dba\_role\_privs;  
SQL> select \* from dba\_sys\_privs;  
SQL> select \* from role\_sys\_privs;   
查自己拥有哪些系统权限  
SQL> select \* from session\_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  
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。  
系统权限回收：系统权限只能由DBA用户回收  
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。   
****实体权限管理****   
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。  
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)  
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;   
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。  
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 角色管理****   
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。系统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';   
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。  
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。  
EXP\_FULL\_DATABASE， IMP\_FULL\_DATABASE：这两个角色用于数据导入导出工具的使用。  
AQ\_USER\_ROLE， AQ\_ADMINISTRATOR\_ROLE：AQ:Advanced Query。这两个角色用于oracle高级查询功能。  
SNMPAGENT：用于oracle enterprise manager和Intelligent Agent  
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》  
HS\_ADMIN\_ROLE：A DBA using Oracle's heterogeneous services feature needs this role to access appropriate tables in the data dictionary.  
****管理角色****   
建一个角色  
sql>create role role1;   
授权给角色  
sql>grant create any table,create procedure to role1;   
授予角色给用户  
sql>grant role1 to user1;   
查看角色所包含的权限  
sql>select \* from role\_sys\_privs;   
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;  
修改角色：是否需要口令   
sql>alter role role1 not identified;  
sql>alter role role1 identified by password1;   
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。  
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

与权限安全相关的数据字典表有:  
ALL\_TAB\_PRIVS  
ALL\_TAB\_PRIVS\_MADE  
ALL\_TAB\_PRIVS\_RECD  
DBA\_SYS\_PRIVS  
DBA\_ROLES  
DBA\_ROLE\_PRIVS  
ROLE\_ROLE\_PRIVS  
ROLE\_SYS\_PRIVS  
ROLE\_TAB\_PRIVS  
SESSION\_PRIVS  
SESSION\_ROLES  
USER\_SYS\_PRIVS  
USER\_TAB\_PRIV  
****oracle的系统和对象权限列表****   
alter any cluster 修改任意簇的权限  
alter any index 修改任意索引的权限  
alter any role 修改任意角色的权限  
alter any sequence 修改任意序列的权限  
alter any snapshot 修改任意快照的权限  
alter any table 修改任意表的权限  
alter any trigger 修改任意触发器的权限  
alter cluster 修改拥有簇的权限  
alter database 修改数据库的权限  
alter procedure 修改拥有的存储过程权限  
alter profile 修改资源限制简表的权限  
alter resource cost 设置佳话资源开销的权限  
alter rollback segment 修改回滚段的权限  
alter sequence 修改拥有的序列权限  
alter session 修改数据库会话的权限  
alter sytem 修改数据库服务器设置的权限  
alter table 修改拥有的表权限  
alter tablespace 修改表空间的权限  
alter user 修改用户的权限  
analyze 使用analyze命令分析数据库中任意的表、索引和簇  
audit any 为任意的数据库对象设置审计选项  
audit system 允许系统操作审计  
backup any table 备份任意表的权限  
become user 切换用户状态的权限  
commit any table 提交表的权限  
create any cluster 为任意用户创建簇的权限  
create any index 为任意用户创建索引的权限  
create any procedure 为任意用户创建存储过程的权限  
create any sequence 为任意用户创建序列的权限  
create any snapshot 为任意用户创建快照的权限  
create any synonym 为任意用户创建同义名的权限  
create any table 为任意用户创建表的权限  
create any trigger 为任意用户创建触发器的权限  
create any view 为任意用户创建视图的权限  
create cluster 为用户创建簇的权限  
create database link 为用户创建的权限  
create procedure 为用户创建存储过程的权限  
create profile 创建资源限制简表的权限  
create public database link 创建公共数据库链路的权限  
create public synonym 创建公共同义名的权限  
create role 创建角色的权限  
create rollback segment 创建回滚段的权限  
create session 创建会话的权限  
create sequence 为用户创建序列的权限  
create snapshot 为用户创建快照的权限  
create synonym 为用户创建同义名的权限  
create table 为用户创建表的权限  
create tablespace 创建表空间的权限  
create user 创建用户的权限  
create view 为用户创建视图的权限  
delete any table 删除任意表行的权限  
delete any view 删除任意视图行的权限  
delete snapshot 删除快照中行的权限  
delete table 为用户删除表行的权限  
delete view 为用户删除视图行的权限  
drop any cluster 删除任意簇的权限  
drop any index 删除任意索引的权限  
drop any procedure 删除任意存储过程的权限  
drop any role 删除任意角色的权限  
drop any sequence 删除任意序列的权限  
drop any snapshot 删除任意快照的权限  
drop any synonym 删除任意同义名的权限  
drop any table 删除任意表的权限  
drop any trigger 删除任意触发器的权限  
drop any view 删除任意视图的权限  
drop profile 删除资源限制简表的权限  
drop public cluster 删除公共簇的权限  
drop public database link 删除公共数据链路的权限  
drop public synonym 删除公共同义名的权限  
drop rollback segment 删除回滚段的权限  
drop tablespace 删除表空间的权限  
drop user 删除用户的权限  
execute any procedure 执行任意存储过程的权限  
execute function 执行存储函数的权限  
execute package 执行存储包的权限  
execute procedure 执行用户存储过程的权限  
force any transaction 管理未提交的任意事务的输出权限  
force transaction 管理未提交的用户事务的输出权限  
grant any privilege 授予任意系统特权的权限  
grant any role 授予任意角色的权限  
index table 给表加索引的权限  
insert any table 向任意表中插入行的权限  
insert snapshot 向快照中插入行的权限  
insert table 向用户表中插入行的权限  
insert view 向用户视图中插行的权限  
lock any table 给任意表加锁的权限  
manager tablespace 管理（备份可用性）表空间的权限  
references table 参考表的权限  
restricted session 创建有限制的数据库会话的权限  
select any sequence 使用任意序列的权限  
select any table 使用任意表的权限  
select snapshot 使用快照的权限  
select sequence 使用用户序列的权限  
select table 使用用户表的权限  
select view 使用视图的权限  
unlimited tablespace 对表空间大小不加限制的权限  
update any table 修改任意表中行的权限  
update snapshot 修改快照中行的权限  
update table 修改用户表中的行的权限  
update view 修改视图中行的权限

参考至：http://blog.csdn.net/junmail/article/details/4381287

             http://read.newbooks.com.cn/info/48990.html

             http://rocolex.blog.163.com/blog/static/6844641020101199262373/

本文原创，转载请注明出处、作者

如有错误，欢迎指正

邮箱:czmcj@163.com