oracle 管理安全性

在 oracle 数据库中，涉及安全方面内容包括：用户和模式，系统权限，对象权限，数据库角色，访问粒度控制等。

用户：connect system/shine 其中 shine 是 system 这个 dba 用户的密码

create user oracle_admin identified by oracle_admin;创建一个用户名和密码为 oracle_admin 的用户。

grant careat session,dba to oracle_admin;create session 是提供用户连接数据库的能力。dba 是一个具有 100 多个系统权限的角色。

1. 重置密码

alter user oracle_admin identified by dba_admin;修改了 oracle_admin 用户的 密码为 dba_admin；

锁定账户以及解除账户锁定：alter user <username> account[lock|unlock]有权限的用户锁定其他用户，导致这个用户不能连接上 oracle。

2. 删除用户

drop user oracle_admin;如果 oracle_admin 下面拥有一个对象需要使用：drop user oracle_admin cascade；

删除表：drop table table_name;

3. 系统权限

为了查看用户的 oracle 数据库中可以使用的权限集合。可以查看 dba_sys_privs 数据库图。某个用户使用的权限集合可以使用 user_sys_privs 来查看。

如：desc dba_sys_privs。

4. 常用的系统权限

create session 连接到数据库上

create sequence 创建序列，序列是一系列的数字，通常用来自动填充主键列。

create synonym 创建同义词。同义词用于引用其他模式中的表

create table 创建表

create any table 在任何模式中创建表

drop table 删除表

drop any table 删除任何模式中表

create procedure 创建存储过程

create any procedure 创建任何存储过程

create user 创建用户

drop user 删除用户

create view 创建视图

5. 向用户赋予系统权限的基本语法如下

grant system_privilege to username[with admin option]

6. 与赋予权限相对应的是删除权限 ,从数据库中删除权限的基本语法格式如下

revoke system_privilege from username ;

注意的是：取消系统权限的数据库用户不需要是最初授予系统权限的用户。任何具有 admin option 系统权限的用户都能够取消其他用户的系统权限。

在 dba 用户下创建一个新用户 , 如果让新用户能够具有创建表这个功能。需要 3 点：

1：需要 create session 权限以致能连接数据库。2：需要 create table 权限功能。 3：需要是用表空间的权限.

如果创建一个用户 dropme , 并且该用户赋予了 1 , 2 条件 , 以 dropme 身份连接数据库 , 如果这个时候执行创建表的操作。会出现错误。错误为 : 对表空间 user 无权限。

为了允许 dropme 账户可以创建表，可以使用 alter user 命令来为该账户赋予 sysaux 表空间的配额。在 dba 状态下：alter user dropme default tablespace sysaux quota

10M sysaux 这个时候如果以 dropme 连接到数据库，创建就可以成功。

7. 对象权限

对于数量众多的系统权限来说，对象权限比较少，并且好理解。对象权限如下:

1.select　2.insert,update,delete 3.execute 可以应用与 PL/SQL 过程，函数，程序包以及其他可执行元素。execute 权限可以允许终端用户直接执行和编译过程，函数等。4.index 和 references 只能应用与表。5.alter

属于对象权限的语法格式如下：

grant object_privilege on object_name to username[wiht option];为了授予数据库用户指定的对象权限，并同时赋予其相同的权限赋予其他用户的权限，需要在用户的 grant 语句中包括 with grant option。

在 athos 用户下把其下的表 mytale 的 select 对象权限赋予用户 scott。connect athos/athos　grant　select on mytalbe to scott;

在 scott 用户下。不具传授对象的权利，grant select athos.mytable to otheruser；会出现错误。此时如果想传授对象的权利，要在 grant select athos.mytable to otheruser 后面加上 with grant option。

只用授权者才能够撤销为其他用户赋予的权限。因此 athos 可以收回为用户 scott 授予 athos.mytable 上的 select 权限。

revoke select on mytable from scott ;

为了检查 oracle 数据库中已有的表的权限，可以查询 user_tab_privs，all_tab_privs，或者 dba_tab_privs 数据数视图

desc user_tab_privs ;

select * from user_tab_privs ;

8. 数据库角色

数据库角色就是权限的命名集合。使用角色可以大大降低用户权限的维护负担，角色可以是对象或系统权限的命名集合。

创建数据库角色的基本语法格式如下：

create role role_name;

create role role_name identified by role_password 当数据库角色创建之后，其功能和实际的数据库用户类似。为角色赋予权限与为用户赋予权限使用的 grant 语句大体相同，取消数据库角色的权限与 reboke 语句也大致相同，也可以创建带有口令的角色、

在 system 下 create role 权限授予 athos。接着以 athos 连接数据库。使用 create role 命令来创建 assembly_line 的新的数据库角色。

大致过程如下 :在 system 下 ,grant create role to athos ;在 athos 下 ,create role assembly_line；创建了一个数据库角色。grant select，insert，update on mytbale to assembly_line；

此时可以通过检索 user_tab_privs 数据字典来获得有关数据库角色授权的信息。

select * from user_tab_privs ;可以看出在权限中显示 select ,insert ,update。

在 athos 用户下：grant assembly_line to scott；

在 scott 下就可以针对 athos.mytable 执行 select，insert，update 这 3 想操作。

默认情况下，在将角色授予用户时就为该用户启动了该角色，也就是说，在用户连接到数据库上时，就可以自动使用了这个角色。为了增强安全性，也可以默认地禁用一个角色。在这种情况下，当用户连接到数据库上是，在使用这个禁用的之前，必须先启动这个角色，如果有口令，必须输入口令；

简单实例：在 system 用户下：create role test identified by test；创建角色 grant test to scott；alter user scott default role all except test；禁用了 test 角色；

9. 启用 test 角色

在 scott 状态下。set role test identified by test;如果不需要某个角色了，可以使用 drop role role_name;