Oracle 利用create user、create role、grant命令控制数据访问。

* Database system privilege
* Database object privilege

**SYS**用户拥有Oracle用来管理数据库的核心内部表；

**SYSTEM**用户则拥有其他的表和视图 - 可以使用SYSTEM用户登录，创建其他用户。

create user username identified  
            {by password | externally | globally as 'extnm'};

-- 创建用户  
create user Dora identified by avocado;  
-- 修改密码  
alter user Dora identified by psyche;  
-- 不给用户CREATE SESSION权限，则用户无法登录  
grant CREATE SESSION to Dora;

* 密码会过期 - ***由分配给账户的配置文件决定***
* 用户有可能会由于多次连续失败而被锁定
* 修改密码之后，需要维护密码历史记录，防止重用以前的密码

配置文件由DBA管理 - create profile命令创建。

关于密码和账户的访问，配置文件规定了一下内容：

* 密码的生存期 Lifetime，决定了多长时间必须更改密码
* 密码的“过期日期”后的宽限期，在此期间可以修改密码
* 在账户自动“锁定”之前允许连续失败的次数
* 账户保持锁定状态的天数
* 在重用一个密码前必须经过的天数
* 在重用一个密码前必须经历的更改次数

数据库管理员可以利用password命令修改任何用户的密码；其他用户只能修改自己的密码。

在SQL\*Plus中使用password命令修改密码，输入的新密码不会显示在屏幕上。

SQL> conn wpeng/exigen@server03  
Connected.  
SQL> password  
Changing password for WPENG  
Old password:  
New password:  
Retype new password:  
**Password changed**

可以使用配置文件管理密码的过期、重用和复杂性。

可以设定一个密码的生存期，锁定密码使用太久的账户，降低密码的复杂性，将多次失败登陆的账户锁定...

**密码过期机制**

-- 创建LIMITED\_PROFILE配置文件  
create profile LIMITED\_PROFILE limit  
FAILED\_LOGIN\_ATTEMPTS **5**;  
-- 交由JANE使用  
create user JANE identified by EYRE  
profile LIMITED\_PROFILE;  
grant create SEESION to JANE;

-- JANE 5次连续失败  
connect jane/eyre  
ERROR:  
ORA-**28000**: the account is locked  
-- 解锁  
alter user JANE account unlock;  
-- 锁住指定账户  
alter user JANE account lock;  
  
-- 在超过配置文件的PASSWORD\_LOCK\_TIME值，自动解锁  
  
-- 通过PASSWORD\_LIFE\_TIME设置密码的最大寿命  
-- 强制30天修改一次密码  
alter profile LIMITED\_PROFILE limit  
PASSWORD\_LIFE\_TIME **30**;  
  
-- 为到期密码设置一个宽限期 PASSWORD\_GRACE\_TIME

"锁定"账户，可以在一定时间之后，自动解锁；  
“过期”账户，则需要DBA手工设置才能重新生效。

SQL> alter user wpeng password expire;  
User altered.  
  
SQL> conn wpeng/exige@server03  
ERROR:  
ORA-**28001**: the password has expired  
  
Changing password for wpeng  
New password:  
Retype new password:  
Password changed  
Connected.

可以强制用户在第一次登陆的时候，修改自己的密码：

-- 使用create user 的子句 passowed expire  
create user \*\*\* identified by \*\*\*  
password expire;

可以通过查询DBA\_USERS视图的EXPIRY\_DATE，得到任意账户密码的到期日期。

**密码重用机制**  
为了防止密码被重用，可以使用配置文件中的两个参数（相互排斥的，设置其中一个，另一个必须被设置为UNLIMITED）：

* PASSWORD\_REUSE\_MAX - 指定一个密码重用前密码必须变化的次数
* PASSWORD\_REUSE\_TIME - 指定一个密码可以重用前必须间隔的天数

alter profile LIMITED\_PROFILE limit  
PASSWORD\_REUSE\_MAX **3**  
PASSWORD\_REUSE\_TIME UNLIMITED;  
  
-- 试图使用一个近期的密码，则：  
ORA-**280007**: the password cannot be reused.

**标准角色**  
应用程序中用户通过角色来获得权限。  
可以根据应用程序的需要，将系统权限和对象访问融入到角色当中。  
  
CONNECT, RESOURCE, DBA是为了向后兼容，所以不应该再使用。  
为了替换CONNECT, RESOURCE, DBA，应当创建自己的角色以便有权限执行特定的系统权限。

-- 可以将任何系统权限赋予其他用户和角色，或则授予public  
-- with admin option 可以保证被授权者可以将权限或角色授予其他用户或角色  
-- all子句授予用户或角色除SELECT ANY DICTIONARY系统权限以外的所有权限  
grant {system privilege | role | all [privileges] }   
[, {system privilege | role | all [privileges]} ...]  
to {user | role} [, {user | role}]  ...  
[identified by password]  
[with admin option]

-- 撤销用户或角色权限  
revoke {system privilege | role | all [privileges] }   
[, {system privilege | role | all [privileges]} http://www.blogjava.net/Images/dot.gif]  
from {user | role} [, {user | role}]  http://www.blogjava.net/Images/dot.gif

-- 删除一个用户  
-- cascade可以删除一个用户及其拥有的所有对象，其中包括参照完整性约束  
-- 使用引用已删除用户模式中的对象的视图、同义词、存储过程、函数或程序包失效  
  
-- 如果不是用cascade，并且用户拥有的对象仍然存在，则Oracle不会删除该用户，相反返回一个错误信息  
drop user username [cascade];

用户可以给他所拥有的任何对象授权；  
数据库管理员可以授予任何系统权限。

-- Dora 拥有COMFORT表并且是数据库管理员  
-- ======系统权限=====  
create user Judy identified by sarah;  
grant create SESSION to Judy;  
  
create user Bob identified by carolyn;  
grant create SEESION, create TABLE, create VIEW, create SYNONYM to Bob;  
  
alter user Bob  
default tablespace users  
quota 5m on users;  
  
-- ======对象权限=====  
-- with grant option，则Bob可以将从Dora的表上得到的权限再授予其他用户。  
grant select on COMFORT to Bob;

如果打算基于另外一个用户的表创建视图，并且把视图的访问权限授予其他用户，则必须拥有对基表的with grant option权限。

除非同义词，否则表名必须跟在表的拥有者的用户名之后；  
否则Oracle会认为这个表不存在。

select \* from Dora.COMFORT;

可以使用任何约束或任何计算来创建视图，然后把访问视图的而不是访问基表的权限，授予其他用户。

创建同义词  
创建一个视图，使它包含其他用户的整个表或视图的另一种方法就是创建同义词。

create synonym LITTLECONFORT for Dora.SOMECOMFORT;

**创建角色**  
**为了创建角色，必须拥有CREATE ROLE系统权限**

create role role\_name  
[not identified  
| identified {by password | using [schema.]package   
|externally | globally}];

create role CLERK;  
create role MANAGER;

grant select on COMFORT to CLERK;  
  
grant CREATE SESSION to CLERK;  
grant CREATE SESSION, CREATE VIEW to MANAGER;  
-- 不可以进行循环授权  
grant CLERK to MANAGER;

使用with admin option子句，那么被授权者有权将这个角色授予其他的用户或角色。  
被授权者也能更改和删除这个角色。  
  
**角色可以看做命名的一组权限。**  
对视图、过程、函数、程序包或者外键进行操作时，不能使用通过角色授予用户权限。  
在创建这些类型的数据库对象后，必须依靠所需权限直接的授权。

grant CLERK to Bob;  
  
grant MANAGER to Dora with admin option;

**为角色添加密码**  
使用alter role命令达到：更改启用角色所需的权限。  
在默认情况下，角色不具有相关联的密码。为了使一个角色就有安全性，应该在alter role命令中使用identified关键字。

alter role MANAGER identified by cygnusxi;  
-- 角色也可以绑定到操作系统权限。  
alter role MANAGER identified externally;

-- 删除角色密码  
alter role MANAGER no identified;

在用户的账户被更改时，可以通过alter user命令的default role子句来为该用户创建默认角色列表。这个命令的默认的动作是：将用户的所有角色设置为默认角色，每当用户登录启用所有的角色。  
在任何时候，用户可以启用的角色最大数目是通过MAX\_ENABLED\_ROLES数据库初始化参数设置的。

-- Bob在登录时，启用CLERK角色  
alter user Bob  
default role CLERK;  
-- 启用一个非默认角色  
set role CLERK;  
  
-- =====查看当前用户，已经启用的角色，可以从SESSION\_ROLES数据字典视图中选相关信息  
-- =====查询SEESION\_PRIVS可以知道当前启动的系统权限  
  
set role all;  
set role all except CLERK;  
  
-- 如果一个角色有相关密码，则必须通过identified by 指定  
set role MANAGER identified by cygnusxi;  
  
-- 在会话中，禁用某一个角色 - 此命令在当前会话中，禁用所有角色  
set role none;

-- 撤销角色的权限  
revoke SELECT on COMFORT from CLERK;  
  
-- 删除角色  
drop role CLERK;  
drop role MANAGER;

系统和对象权限的grant和revoke操作会立即生效。  
而角色的grant和revoke操作仅当前用户发出set role或启用一个新的用户会话时才生效。

-- =====给指定的列授予UPDATE权限  
grant update(Noon, Midnight) on COMFORT to Judy;

**revoke all**: 将删除以前列出的所有权限，从SELECT到INDEX；

创建一个其where子句中包含User（伪列）的一个视图：

-- 当查询MY\_CHECKOUT时，将依赖于伪列User的记录  
create or replace view MY\_CHECKOUT as  
  select \* from BOOKSHELF\_CHECKOUT  
    where SUBSTR(name, **1**, INSTR(name, ' ')-**1**) = User;

给公众授予访问权：

grant select on MY\_CHECKOUT to public;

创建public同义词，可以简化使用，不必使用模式所有者

create public synonym MY\_CHECKOUT for Practice.MY\_CHECKOUT;

-- 有限资源的授权  
-- ===== create user 或者 alter user参数  
alter user Bob  
quota 100M on USERS;  
  
-- =====想撤销一个用户的空间限额，将UNLIMITED TABLESPACE系统权限授予该用户