用户管理

创建

- 命令格式：

```
create user 用户名
identified {by 口令 | externally}
      [default tablespace 默认表空间名]
      [temporary tablespace 临时表空间名]
        [quota {整数[K | M]| unlimited} on 表空间名]
        [quota {整数[K | M]| unlimited} on 表空间名]
            [password expire]
                [account {lock | unlock}]
                [profile {概要文件名 | default}]
```

```
create user u1
    identified by 123456//密码
    default tablespace users
    temporary tablespace temp
    quota unlimited on users
```

u1
123456
ORACL
NORMAL
使用上述信息，无连接数据库权限

指定权限

修改

```
alter user u1 password expire
```
u1口令修改为过期状态
再次登录需要指定新的密码

删除

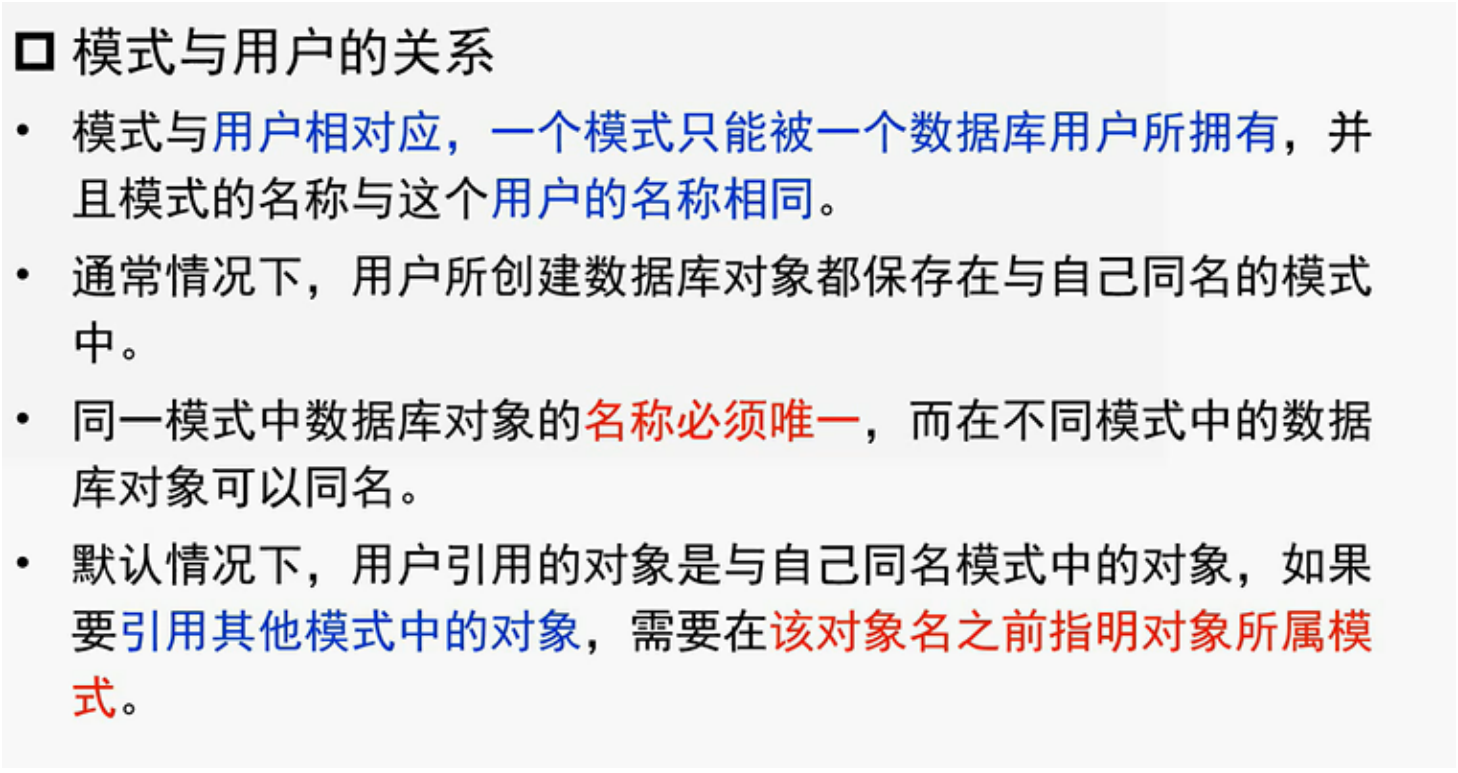
```
drop user Yonghuming [cascade]
```
指定cascade将会删除该用户的所有对象
Table下的所有对象将被删除
若不指定将会先提醒

模式

用户权限

模式：一系列逻辑数据结构或对象的集合

模式与用户的关系:



模式切换与选择

模式选择与切换

- 如果用户以NORMAL身份登录，则进入同名模式。
- 若以SYSDBA身份登录，则进入SYS模式。
- 如果以SYSOPER身份登录，则进入PUBLIC模式。

- Oracle用户是Oracle数据库中的重要概念，与用户密切相关的另一个概念就是模式(schema)。
- 用户主要连接数据库和访问数据库对象，用户是用来连接数据库和访问数据库。
- 模式是数据库对象的集合。模式对象是数据库数据的逻辑结构，把数据库对象用模式分开成不同的逻辑结构。
- 用户是用来连接数据库对象。而模式是用来创建和管理对象的。模式跟用户在oracle数据库中是一对一的关系。
- 一个用户一般对应一个模式，该用户的模式名等于用户名，并作为该用户的缺省模式。
- Oracle数据库中不能通过create schema语句来新创建一个模式，要想创建一个模式，只能通过创建一个用户的方法解决，Oracle中虽然有create schema语句，但它并不是用来创建模式的。

权限管理

- 一个新的用户被创建后，该用户还无法操作数据库，还需要为用户授予相关的权限。
- 权限是指在数据库中执行某种操作的权利。
- Oracle有两种类型的权限：系统权限和对象权限。
- 系统权限：允许用户在数据库的任何模式上执行特定操作所需要的权限称为系统权限，这些操作包括建立、修改和删除表、视图、索引、表空间、触发器、用户、角色等。

- 一个新的用户被创建后，该用户还无法操作数据库，还需要为用户授予相关的权限。
- 权限是指在数据库中执行某种操作的权利。
- Oracle有两种类型的权限：系统权限和对象权限。
- 系统权限：允许用户在数据库的任何模式上执行特定操作所需要的权限称为系统权限，这些操作包括建立、修改和删除表、视图、索引、表空间、触发器、用户、角色等。

系统权限	说明
Create session	连接数据库
Create tablespace	创建表空间
Alter tablespace	修改表空间
Drop tablespace	删除表空间
Create user	创建用户
Alter user	修改用户
Drop user	删除用户
Create table	创建表
Create any table	在任何用户模式中创建表
Drop any table	删除任何用户模式中的表
Alter any table	修改任何用户模式中的表
Select any table	查询任何用户模式中基本表的记录
Insert any table	向任何用户模式中的表插入记录

- 授予系统权限：可以使用Grant语句把系统权限授予一个用户或一个角色，语句格式：

```
grant  { 系统权限 | 角色 } [, { 系统权限 | 角色 } ]…
To  { 用户 | 角色 | public} [, { 用户 | 角色 | public}] …
    [with admin option]
```

查询所有系统权限
```
select * from system_privilege_map
```

为用户u1授予在任何用户模式下创建表的权限和查询任何模式中基本表中数据的权限
1.首先授予u1用户连接数据库的权限
```
grant Creat session
to u1
```

2.完成之后可以点击u1单击右键选择编辑，之后查看系统权限，将会发现CREATE SESSION

3.之后便可使用用户u1进行登录
用户名：u1
密码：123456
ORACL
NORMAL

4.继续操作
```
grant Creat any table,
 Select any table
to u1
```

5.当前情况下可以建表
```
create table ban.test//在ban用户下建表
(a number,
b date)
```

权限回收

- 回收系统权限：使用REVOKE语句可以从用户或角色中回收系统权限。语句格式：

```
revoke { 系统权限 | 角色 } [, { 系统权限 | 角色 } ]…
from  { 用户 | 角色 | public} [, { 用户 | 角色 | public}]
 …
```

对象权限

基本定义

- 对象权限：允许用户访问一个特定对象并对特定对象执行特定操作所需要的权限称为对象权限。
- 对象权限针对不同模式的对象，如表、视图、序列、过程、函数等
- 对象权限可以是对数据的查询(select)、修改(update)、删除(delete) 、插入(insert)或引用(references)，也可以是是否可以执行(execute)程序的权限，或修改对象结构(alter)的权限。

授予对象权限

- 授予对象权限：使用grant语句授予对象权限，语句格式：

```
grant  { 对象权限 [,对象权限 ]…| all  [ privileges ]}
on [模式.] 对象名
to  { 用户 | 角色 | public} [, { 用户 | 角色 | public}]
…
   [with grant option]
```

```
授予u1, u2对对象s的所有权限
grant all privileges
on s
to u1,u2

//把查询s表和修改学生学号的权限授予u3.并允许他再将次权限授予其他用户
grant update(sno), select
on s
to u3
with grant option//并允许他再将次权限授予其他用户
```

执行例5后，U3不仅拥有了对表S的select权限和对学号列的修改权限，还可以传播此权限：

```
GRANT UPDATE(Sno), SELECT ON s TO U4
    WITH GRANT OPTION;
```

同样，U4还可以将此权限授予U5：

```
GRANT UPDATE(Sno), SELECT ON s TO U5
```

但U5不能再传播此权限。

回收对象权限：使用Revoke语句回收对象权限，语句格式：

```
revoke  { 对象权限 [,对象权限 ]…| all  [ privileges ]}
on [模式.] 对象名
from  { 用户 | 角色 | public} [, { 用户 | 角色 |
      public}] …
       [cascade constraints]
```

练习

1. 用系统帐户sys登录数据库，分别创建数据库用户lisa和tom，初始帐户口令都为“888888”，要求将密码设置为过期
```
create user lisa
    identified by 888888//密码
    default tablespace users//默认表空间为users
    temporary tablespace temp
    quota unlimited on users//表空间中使用大小不受限制
        password expire//密码设置为过期状态

create user tom
    identified by 888888//密码
    default tablespace users//默认表空间为users
    temporary tablespace temp
    quota unlimited on users//表空间中使用大小不受限制
        password expire//密码设置为过期状态
```
2. 为了使lisa和tom能够登录数据库，请为其授予相应的权限。
```
grant Creat session
to lisa
grant Creat session
to tom
```
3. 授予用户lisa在自己模式下创建表的权限，在任何模式下删除表的权限，授予用户tom可以在任何模式下创建表的权
```
grant create tablespace,
drop tablespace to lisa

grant create any tablespace,
select any tablespace,
create any view
to tom
```

4. 分别用lisa和tom登录，写出相应的SQL语句验证为其授予的权限。（如果建立的表中有主键约束，需要预先授予lis
```
create table tomTable1
(
    a date,
    b varchar(20)
)

create table lisaTable1
(
    a date,
    b varchar(20)
)
(lisa在自己模式下创建表的权限，在任何模式下删除表的权限)
create table lisaTable2
(
    a date,
    b varchar(20)
)
insert into lisaTable2 (a,b) values (1,"bb");
drop lisaTable1;
drop tom.tomTable1;
```

(tom可以在任何模式下创建表的权限，查询任何模式下表中数据的权限和在任何模式下创建视图的权限)

```
create table tomTable2
(
    a date,
    b varchar(20)
)
create table lisa.tomTable3
(
    a date,
    b varchar(20)
)
select * from lisa.lisaTable2;
```

5. 用系统帐户sys登录数据库，创建用户user1_admin，将角色权限DBA授予用户user1_admin，并将S、P、J、SPJ四张
6. 用user_admin登录，完成以下授权。（要求授完权后，在lisa和tom用户下执行相应的SQL语句验证授权是否成功，
（1）把对表s的insert权力授予用户lisa，并允许他再将此权限授予其他用户。
（2）用户tom对s，p，j三个表有select和insert权力
（3）用户lisa对spj表有delete权力，对qty字段具有update权力。
（4）收回lisa对s表的插入权限。
7. 把对用户tom授予的所有权限收回，只保留登录权限。（系统权限和对象权限应该分别写语句收回）
8. 用系统帐户sys登录数据库，创建用户user2_admin，将角色权限DBA授予此用户，在user2_admin的模式下导入Sud
9. 使用user2_admin登录，创建角色school_manager，将user2_admin模式下五张表的插入、删除、修改和查询数据的
10. 对于通过school_manager角色授出的权限，在tom用户下执行相应的SQL语句对权限进行验证。