十二、用户管理

# 创建新用户

创建用户必须要有管理员权限，可以使用sys或system用户来创建新用户；

|  |
| --- |
| **CREATE USER 用户名 IDENTIFIED BY 用户密码;** |

**注：**Oracle中，用户密码必须以字母开头；

1. 创建一个用户，用户名叫superman；

|  |
| --- |
| CREATE USER superman IDENTIFIED BY superman; |
|  |

当使用新建用户superman登录时，发现无法登录，并报出一下错误信息：

|  |
| --- |
|  |

这是因为新建立的用户没有创建session的权限，也就不能获取数据库的会话；需要数据库管理员为该用户授予CREATE SESSION权限；

|  |
| --- |
| GRANT CREATE SESSION TO superman; |
|  |

若此时为superman表创建一张名为super\_power的表，则会报出一下错误：

|  |
| --- |
| CREATE TABLE super\_power (  power\_id NUMBER(5) PRIMARY KEY,  power\_name VARCHAR2(20) NOT NULL,  power\_description VARCHAR2(500)  ); |
|  |

要解决上述问题，首先需要授予用户创建表CREATE TABLE的权限；但会报出以下错误：

|  |
| --- |
| GRANT CREATE TABLE TO superman; |
|  |

**原因：**Oracle中数据表的保存方式：在oracle中数据表都保存在硬盘上，但并不是每张表都单独保存在硬盘上，而是将表空间保存在硬盘上，然后将数据表放在表空间中。如果把硬盘比作图书馆的话，每个表空间就是一个书架；因此，要想创建表，还要赋予用户操作表空间的权限；

|  |
| --- |
| ALTER USER superman QUOTA UNLIMITED ON users; |
|  |

# 修改用户密码

**格式1：**

|  |
| --- |
| **PASSWORD 用户名;** |

**格式2：**

|  |
| --- |
| **ALTER USER 用户名 IDENTIFIED BY 密码;** |
|  |

当管理员为用户重置密码后，往往希望用户在第一次登录时修改密码，所以可以通过如下语句让用户密码失效：

|  |
| --- |
| ALTER USER superman PASSWORD EXPIRE; |
|  |

# 删除用户

|  |
| --- |
| **DROP USER 用户名 [CASCADE];** |

**注**：

* 用户自己不能删除自己；
* 若要删除的用户已经创建了数据表，那么删除时要加CASCADE，级联删除数据表；

# 4. 用户权限管理

## 4.1 权限

新创建的用户没有任何权限，需要系统管理员为其指定相应权限；在Oracle中已经事先定义好了140多种权限，分为

1. **系统权限**：描述用户对数据库访问的权限，控制用户可以执行一个或一组数据库操作。例如建库，建表，登录，创建用户等;

常用系统权限

|  |  |
| --- | --- |
| **CREATE SESSION** | 连接数据库 |
| **CREATE TABLE** | 创建数据表 |
| **CREATE VIEW** | 创建视图 |
| **CREATE INDEX** | 创建索引 |
| **CREATE PUBLIC SYNONYM** | 创建同义词 |
| **CREATE PROCEDURE** | 创建过程、函数、包 |
| **CREATE TRIGGER** | 创建触发器 |
| **CREATE CLUSTER** | 创建簇 |
| **CREATE ANY TABLE** | 为任何方案建表 |
| **ALL** | 要访问数据对象的所有权限； |

查看系统权限：数据字典system\_privilege\_map

|  |
| --- |
| SELECT \* FROM system\_privilege\_map ORDER BY name; |
|  |

1. **对象权限**：用户对其他用户数据对象操作的权限，例如访问其他用户的数据表；

查看对象权限：数据字典dba\_tab\_privs

|  |
| --- |
| SELECT \* FROM dba\_tab\_privs; |
|  |

常见对象权限：

|  |  |
| --- | --- |
| **SELECT** | 查询权限 |
| **INSERT** | 插入权限 |
| **UPDATE** | 更新权限 |
| **DELETE** | 删除权限 |

指定对象权限：

|  |
| --- |
| **GRANT 对象权限1[,对象权限2…] ON 要操作数据对象 TO 用户;** |

**注**：数据对象所属用户也可以为其他用户指定对象权限；

1. 让fzx用户可以访问scott的emp表；

|  |
| --- |
| GRANT SELECT ON emp TO fzx; |

## 4.2 方案

当用户创建时，系统会为用户创建一套方案；

## 4.3 角色

角色就是相关权限的命令集合，如果将权限逐个授予用户，其操作会很麻烦。为了简化权限管理，Oracle引入了角色这一数据对象，每个角色包含了一种到多种权限，用户可以根据需要指定相应的角色，从而简化操作；

查看角色：数据字典dba\_roles

|  |
| --- |
| SELECT \* FROM dba\_roles; |
|  |

1. 查看角色具有的系统权限：role\_sys\_privs

|  |
| --- |
| SELECT privilege,admin\_option FROM role\_sys\_privs WHERE role='角色名'; |
|  |

1. 查询角色具有的对象权限：dba\_tab\_privs

|  |
| --- |
| SELECT \* FROM dba\_tab\_privs WHERE grantor='角色名'; |

1. 显示用户具有的角色及默认角色：dba\_role\_privs

|  |
| --- |
| SELECT \* FROM dba\_role\_privs WHERE grantee='用户名'; |
|  |

**注意：角色名和用户名等字段都区分大小写，一般都为大写；**

角色分为两种：

1. **预定义角色**：Oracle数据库自带的角色；
2. **自定义角色**：用户根据需求自己定义角色，并赋予权限；

自定义角色语法：

|  |
| --- |
| **CREATE ROLE 角色名 NOT IDENTIFIED | IDENTIFIED BY 密码；** |

**注**：若使用NOT IDENTIFIED,则表示角色没有密码，如果不希望角色被其他用户修改，可以使用IDENTIFIED BY为角色加密码；

自定义的角色没有任何权限，需要使用GRANT语句为角色授予系统权限和对象权限；语法和为用户授予权限的方式一样，但授权角色需要加上WITH ADMIN OPTION 或 WITH GRANT OPTION，因为角色要被授予其他用户或权限；

指定权限用GRANT命令；

|  |
| --- |
| **GRANT 角色1[,角色2…] TO 用户名;** |

常见角色：

|  |  |
| --- | --- |
| **connect** | 用户可以登录数据库； |
| **resource** | 用户可以在任意表空间建表； |
| **dba** | 用户具有数据库管理员权限； |

1. 为用户fzx指定数据库连接权限；

|  |
| --- |
| GRANT connect TO fzx; |

1. 为用户fzx指定在任意表空间建表的权限；

|  |
| --- |
| GRANT resource TO fzx; |

删除角色语法：

|  |
| --- |
| **DROP ROLE 角色名;** |

**注：**一旦角色被删除，则被授予权限的用户也没有相应的权限；

## 4.4 授予系统权限

授予系统权限，一般有DBA来完成。如果为用户指定某种权限之后，还希望该用户可以把这种权限授予给其他用户，则语法如下：

对象权限：

|  |
| --- |
| **GRANT 对象权限1[,对象权限2…] ON 数据表(字段1[,字段2…])|角色 TO 用户 [WITH GRANT OPTION];** |

**注：**若被授予对象为角色，则不能加WITH GRANT OPTION;

系统权限：

|  |
| --- |
| **GRANT 角色1[,角色2…] TO 用户名 [WITH ADMIN OPTION];** |

**注**：若加上WITH GRANT OPTION 或WITH ADMIN OPTION，则被授予权限的用户也可以授予其他用户此权限；若用户的权限被收回，则该用户指定给其他用户的权限也一并收回；

## 4.5 收回权限

若要收回权限，可以用REVOKE命令；

|  |
| --- |
| **REVOKE 角色1[,角色2…] FROM 用户名;** |

收回对象权限：

|  |
| --- |
| **REVOKE 对象权限1[,对象权限2…] ON 要操作的数据对象 FROM 用户名;** |

## 4.6 精细访问控制

通过函数、策略实现更加细微的安全访问控制，可以使不同的用户访问相同表时返回不同的数据信息；

# 5. 锁定与解锁用户

## 5.1 锁定用户

使该用户失效，无法登录；

|  |
| --- |
| **ALTER USER 用户名 ACCOUNT LOCK;** |
|  |

## 5.2 解锁用户：

使锁定的用户解锁，例如初始时，scott是锁定的需要解锁后才能登录。

|  |
| --- |
| **ALTER USER 用户名 ACCOUNT UNLOCK;** |
|  |

# 6. profile管理用户**口令**

配置(profile)是口令限制，资源限制的命令集合，当建立数据库时，Oracle会自动建立名为default的profile；当建立用户没有指定profile选项，则Oracle会将default分配给用户；

## 6.1 账户锁定

指定账户登录时最多可以输入密码的次数，也可以指定用户锁定的时间（单位：天），一般用数据库管理员身份执行该命令；

创建profile:

|  |
| --- |
| **CREATE PROFILE 配置名 LIMIT**  **FAILED\_LOGIN\_ATTEMPTS 3**  **PASSWORD\_LOCK\_TIME 2;** |

为用户指定profile:

|  |
| --- |
| **ALTER USER 用户名 PROFILE 配置名;** |

1. 指定用户fzx最多只能尝试3次登录，若登录失败则锁定用户，锁定时间为2天；

|  |
| --- |
| CREATE PROFILE lock\_account LIMIT  FAILED\_LOGIN\_ATTEMPTS 3  PASSWORD\_LOCK\_TIME 2;  ALTER USER fzx PROFILE lock\_account; |
|  |

若此时登录3次错误，则锁定fzx用户：

|  |
| --- |
|  |

## 6.2 账户解锁

若用户被锁定，则可以用下面命令解锁：

|  |
| --- |
| **ALTER USER 用户名 ACCOUNT UNLOCK;** |

|  |
| --- |
|  |

## 6.3终止口令

为了让用户定期修改密码，可以使用终止口令来完成，这个命令同样需要数据库管理员身份来操作；

创建profile:

|  |
| --- |
| **CREATE PROFILE 配置名 LIMIT**  **PASSWORD\_LIFE\_TIME 10**  **PASSWORD\_GRACE\_TIME 2;** |

为用户指定profile:

|  |
| --- |
| **ALTER USER 用户名 PROFILE 配置名;** |

1. 为fzx创建一个profile文件，要求该用户每隔10天要修改自己的登录密码，宽限期为2天，否则锁定；

|  |
| --- |
| CREATE PROFILE myprofile LIMIT  PASSWORD\_LIFE\_TIME 10  PASSWORD\_GRACE\_TIME 2;  ALTER USER fzx PROFILE myprofile; |
|  |

## 6.4 历史口令

如果希望用户在修改密码时，不能使用以前用过的密码，可以使用口令历史，这样Oracle就会将口令修改的信息存放到数据字典中，这样当用户修改密码时，Oracle就会对新旧密码进行比较，当发现新旧密码一样时，就提示用户重新输入密码；

创建profile:

|  |
| --- |
| **CREATE PROFILE 配置名 LIMIT**  **PASSWORD\_LIFE\_TIME 10**  **PASSWORD\_GRACE\_TIME 2;**  **--指定口令不能重用的时间，n天之后即可重用密码**  **PASSWORD\_REUSE\_TIME n;** |

为用户指定profile:

|  |
| --- |
| **ALTER USER 用户名 PROFILE 配置名;** |

## 6.5 删除profile

当不需要某个配置文件时，即可删除该文件；但删除文件后，指定该配置的用户也将被释放，不再受配置约束；

|  |
| --- |
| **DROP PROFILE 配置名 [CASCADE];** |