➢ USER MANAGEMENT

- Introduction
- Privileges
- Roles
- Tables related to user management
- Create User

➢ PROFILE MANAGEMENT

- Profiles
- Create profile
- Resource parameters
- Password parameters

➢ AUDITING

➢ Practical on User Management
➢ Practical on Profile and Session Management
➢ Practical on auditing
➢ ORACLE NETWORK CONFIGURATION

➢ USER MANAGEMENT

1. Introduction

In order to connect to the database a user must specify a valid user name, that has been previously defined, and information of the account stored in data dictionary. When you create a database user the following information should be specified

- User name.
- Authentication method.
- Default tablespace.
- Temporary tablespace.
- Other tablespaces and quotas.
- User profile.

In order to control user access to data and the types of SQL statements the user can execute privileges and roles are assigned to the user by administrator. Administrator has the following responsibilities

- Evaluate database server hardware.
- Installation of oracle software.
- Creation, dropping, startup, shutdown of the database.
- Backing up of database.
- Creating and dropping the users and assigning the role and privileges.
- Tuning up the database.

There are following types of users

- Database administrators
- Network administrators
- Applications developers
- Applications administrators
- Database users
- Security officers

2. Privileges

A user privilege is a right to execute a particular type of SQL statement, or a right to access another user's object, execute a PL/SQL package, and so on. The types of privileges are defined by Oracle.

3. Roles

Roles are created by users (usually administrators) to group together privileges or other roles. They are a means of facilitating the granting of multiple privileges or roles to users.

4. Tables related to user management

TABLE NAME	INFORMATION
dba_profiles	Describes all profiles and their limits
dba_users	Describes all users in database
dba_sys_privs	Describes system privileges and roles
dba_tab_privs	Describes all object grants in database
dba_role_privs	Describes all roles granted to all users and roles in database

5. Create User

Use the CREATE USER statement to create and configure a database user, which is an account through which you can log in to the database, and to establish the means by which Oracle Database permits access by the user.

You can issue this statement in an Automatic Storage Management cluster to add a user and password combination to the password file that is local to the ASM instance of the current node. Each node's ASM instance can use this statement to update its own password file. The password file itself must have been created by the ORAPWD utility.

You can enable a user to connect to the database through a proxy application or application server.

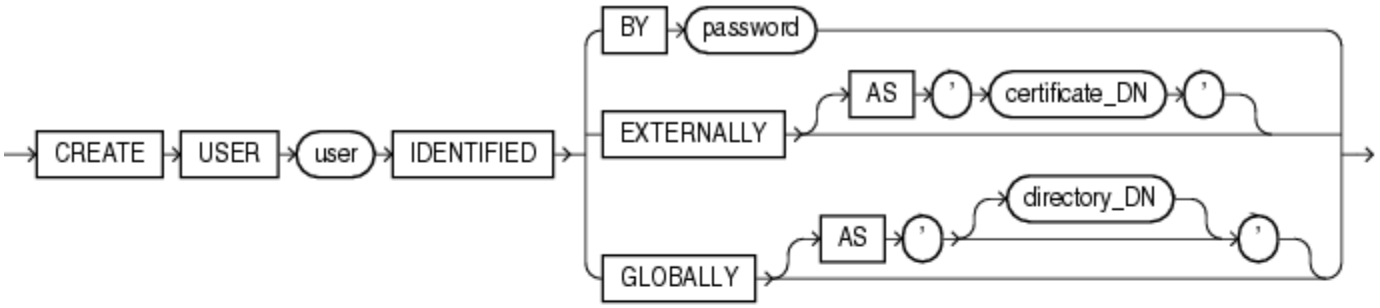
Prerequisites

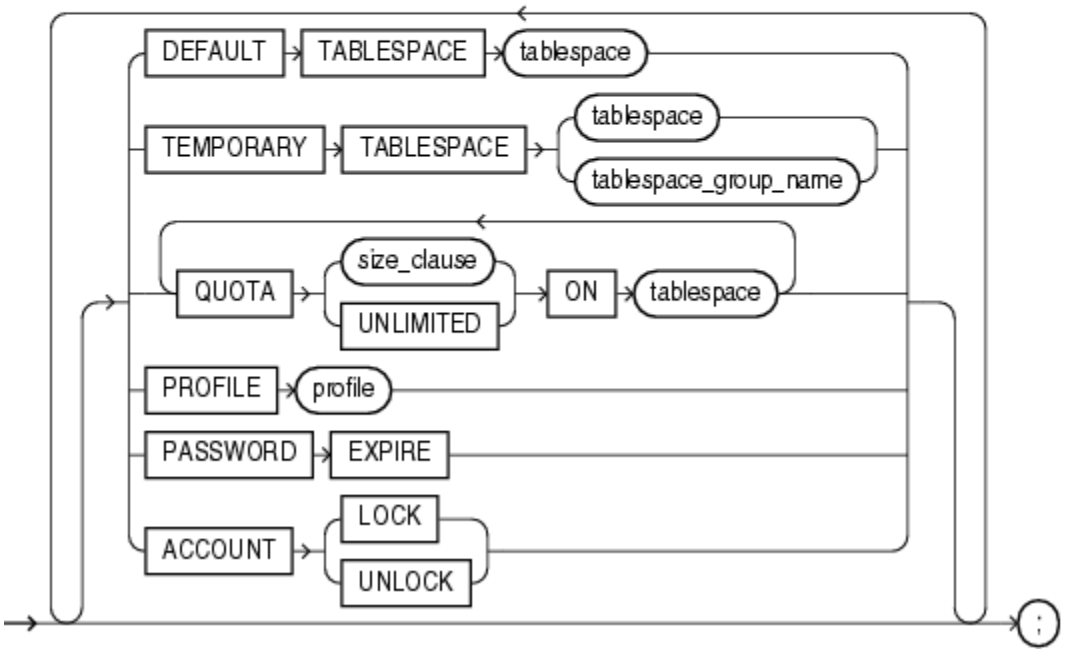
You must have the CREATE USER system privilege. When you create a user with the CREATE USER statement, the user's privilege domain is empty. To log on to Oracle Database, a user must have the CREATE SESSION system privilege. Therefore, after creating a user, you should grant the user at least the CREATESESSION system privilege.

Only a user authenticated AS SYSASM can issue this command to modify the Automatic Storage Management instance password file.

Syntax

Create user:-





Semantics

User

Specify the name of the user to be created. This name can contain only characters from your database character. Oracle recommends that the user name contain at least one single-byte character regardless of whether the database character set also contains multibyte characters.

Note:
Oracle recommends that user names and passwords be encoded in ASCII or EBCDIC characters only, depending on your platform.

IDENTIFIED Clause

The IDENTIFIED clause lets you indicate how Oracle Database authenticates the user.

BY password

The BY password clause lets you creates a local user and indicates that the user must specify password to log on to the database. Passwords are case sensitive. Any subsequent CONNECT string used to connect this user to the database must specify the password using the same case (upper, lower, or mixed) that is used in this CREATE USER statement or a subsequent ALTER USER statement. Passwords can contain any single-byte, multi-byte, or special characters, or any combination of these, from your database character set.

EXTERNALLY Clause

Specify EXTERNALLY to create an external user. Such a user must be authenticated by an external service, such as an operating system or a third-party service. In this case, Oracle Database relies on authentication by the operating system or third-party service to ensure that a specific external user has access to a specific database user.

AS 'certificate_DN':-This clause is required for and used for SSL-authenticated external users only. The certificate_DN is the distinguished name in the user's PKI certificate in the user's wallet.

Caution:

Oracle strongly recommends that you do not use IDENTIFIED EXTERNALLY with operating systems that have inherently weak login security.

GLOBALLY Clause

The GLOBALLY clause lets you create a global user. Such a user must be authorized by the enterprise directory service (Oracle Internet Directory).

The directory_DN string can take one of two forms:

- The X.509 name at the enterprise directory service that identifies this user. It should be of the form CN=username ,other_attributes, where other_attributes is the rest of the user's distinguished name (DN) in the directory. This form creates a **private global schema**.
- A null string (' ') indicating that the enterprise directory service will map authenticated global users to this database schema with the appropriate roles. This form is the same as specifying the GLOBALLY keyword alone and creates a **shared global schema**.

You can control the ability of an application server to connect as the specified user and to activate that user's roles using the ALTER USER statement.

DEFAULT TABLESPACE Clause

Specify the default tablespace for objects that the user creates. If you omit this clause, then the user's objects are stored in the database default tablespace. If no default tablespace has been specified for the database, then the user's objects are stored in the SYSTEM tablespace.

Restriction on Default Tablespaces You cannot specify a locally managed temporary tablespace, including an undo tablespace, or a dictionary-managed temporary tablespace, as a user's default tablespace.

TEMPORARY TABLESPACE Clause

Specify the tablespace or tablespace group for the user's temporary segments. If you omit this clause, then the user's temporary segments are stored in the database default temporary tablespace or, if none has been specified, in the SYSTEM tablespace.

- Specify tablespace to indicate the user's temporary tablespace.
- Specify tablespace_group_name to indicate that the user can save temporary segments in any tablespace in the tablespace group specified by tablespace_group_name.

Restrictions on Temporary Tablespace This clause is subject to the following restrictions:

- The tablespace must be a temporary tablespace and must have a standard block size.
- The tablespace cannot be an undo tablespace or a tablespace with automatic segment-space management.

QUOTA Clause

Use the QUOTA clause to specify the maximum amount of space the user can allocate in the tablespace.

A CREATE USER statement can have multiple QUOTA clauses for multiple tablespaces.

UNLIMITED lets the user allocate space in the tablespace without bound.

Restriction on the QUOTA Clause You cannot specify this clause for a temporary tablespace.

PROFILE Clause

Specify the profile you want to assign to the user. The profile limits the amount of database resources the user can use. If you omit this clause, then Oracle Database assigns the DEFAULT profile to the user.

PASSWORD EXPIRE Clause

Specify PASSWORD EXPIRE if you want the user's password to expire. This setting forces the user or the DBA to change the password before the user can log in to the database.

ACCOUNT Clause

Specify ACCOUNT LOCK to lock the user's account and disable access. Specify ACCOUNT UNLOCK to unlock the user's account and enable access to the account.

Examples

All of the following examples use the example tablespace, which exists in the seed database and is accessible to the sample schemas.

Creating a Database User: Example If you create a new user with PASSWORD EXPIRE, then the user's password must be changed before the user attempts to log in to the database. You can create the user sidney by issuing the following statement:

SQL>create user sidney identified by out_standing1 default tablespace example quota 10m on example temporary tablespace temp quota 5m on system profile app_user password expire;

The user sidney has the following characteristics:

- The password out_standing1
- Default tablespace example, with a quota of 10 megabytes
- Temporary tablespace temp
- Access to the tablespace SYSTEM, with a quota of 5 megabytes
- Limits on database resources defined by the profile app_user
- An expired password, which must be changed before sidney can log in to the database

Creating External Database Users: Examples The following example creates an external user, who must be identified by an external source before accessing the database:

SQL>crezte user apps_user1 identified externally default tablespace example quota 5m on example profile app_user;

The user app_user1 has the following additional characteristics:

- Default tablespace example
- Default temporary tablespace example
- 5M of space on the tablespace example and unlimited quota on the temporary tablespace of the database
- Limits on database resources defined by the app_user profile

To create another user accessible only by an operating system account, prefix the user name with the value of the initialization parameter OS_AUTHENT_PREFIX. For example, if this value is "ops$", then you can create the externally identified user external_user with the following statement:

SQL>create user ops$external_user identified externally default tablespace example quota 5m on example profile app_user;

Creating a Global Database User: Example The following example creates a global user. When you create a global user, you can specify the X.509 name that identifies this user at the enterprise directory server:

SQL>create user global_user identified globally as'CN=analyst, OU=division1, O=oracle, C=US' default tablespace example quota 5m on example;

PROFILE MANAGEMENT

Profiles

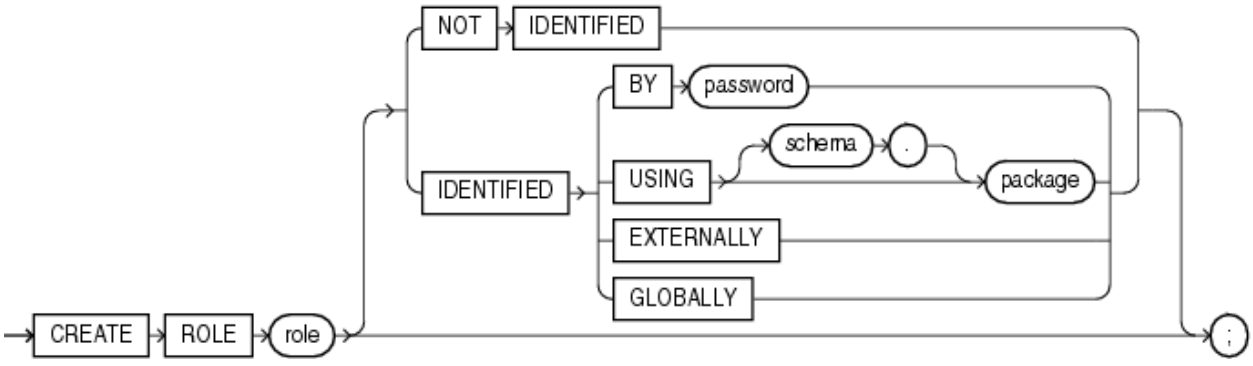
- Profiles define resource limits imposed upon a user account. The default profile sets all resource limits to unlimited.There are three types of privileges and roles
- **System privilege:-**it is a system defined privilege usually only granted to administrators. Performed at dblevel.
- **Object privilege**:-it is a system defined privilege that controls access to specific objects. Performed at object (table space) level.
- **Role**:-it is a collection of privileges and roles

These privileges and roles can only be granted by users who have privilege. All the actions done by all users including administrator can be monitored.

The privileges and roles can be given to the users using GRANT command and taken away using REVOKE command.

SQL>grant privilege-name to user;

SQL>revoke privilege-name to user;



Semantics

Role

Specify the name of the role to be created. Oracle recommends that the role contain at least one single-byte character regardless of whether the database character set also contains multibyte characters. The maximum number of user-defined roles that can be enabled for a single user at one time is 148.

Some roles are defined by SQL scripts provided on your distribution media.

NOT IDENTIFIED Clause

Specify NOT IDENTIFIED to indicate that this role is authorized by the database and that no password is required to enable the role.

IDENTIFIED Clause

Use the IDENTIFIED clause to indicate that a user must be authorized by the specified method before the role is enabled with the SET ROLE statement.

BY password

The BY password clause lets you create a local role and indicates that the user must specify the password to the database when enabling the role. The password can contain only single-byte characters from your database character set regardless of whether this character set also contains multibyte characters.

USING package

The USING package clause lets you create an application role, which is a role that can be enabled only by applications using an authorized package. If you do not specify schema, then the database assumes the package is in your own schema.

NOTE:

When you grant a role to a user, the role is granted as a default role for that user and is therefore enabled immediately upon logon. To retain the security benefits of an application role, you must ensure that the role is not a default role. Immediately after granting the application role to a user, issue an ALTER USER statement with the DEFAULT ROLE ALL EXCEPT role clause, specifying the application role. Doing so will enforce the rule that, in subsequent logons by the user, the role will not be enabled except by applications using the authorized package.

EXTERNALLY

Specify EXTERNALLY to create an external role. An external user must be authorized by an external service, such as an operating system or third-party service, before enabling the role.

Depending on the operating system, the user may have to specify a password to the operating system before the role is enabled.

GLOBALLY

Specify GLOBALLY to create a global role. A global user must be authorized to use the role by the enterprise directory service before the role is enabled at login.

If you omit both the NOT IDENTIFIED clause and the IDENTIFIED clause, then the role defaults to NOT IDENTIFIED.

Create profile

A profile can be created using CREATE PROFILE statement. It has set of limits on database resources. If you assign the profile to a user, then that user cannot exceed these limits. Profile can be enabled or disabled.

Prerequisites

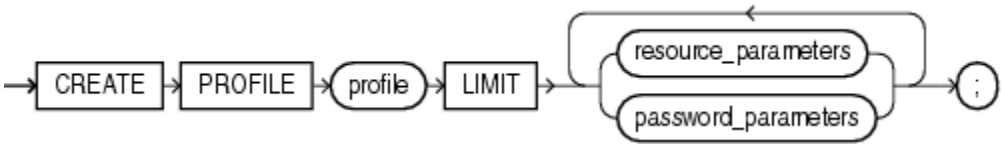
To create a profile, you must have the CREATE PROFILE system privilege.

To specify resource limits for a user, you must:

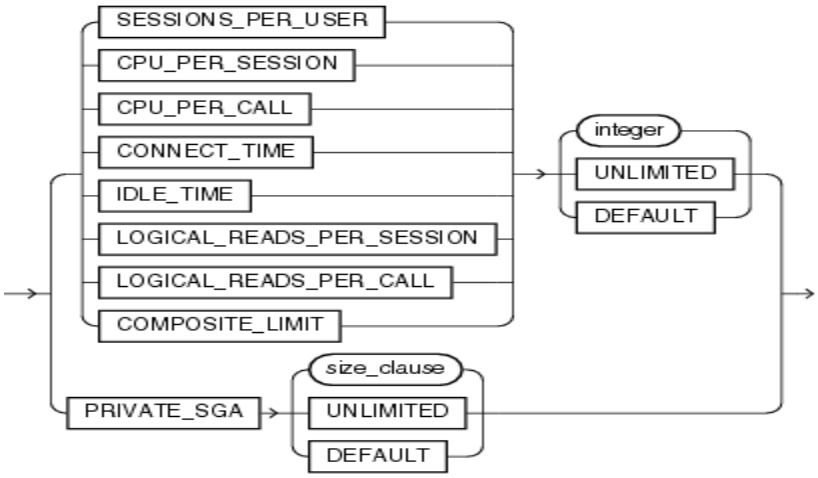
- Enable resource limits dynamically with the ALTER SYSTEM statement or with the initialization parameter RESOURCE_LIMIT. This parameter does not apply to password resources. Password resources are always enabled.
- Create a profile that defines the limits using the CREATE PROFILE statement
- Assign the profile to the user using the CREATE USER or ALTER USER statement

Syntax

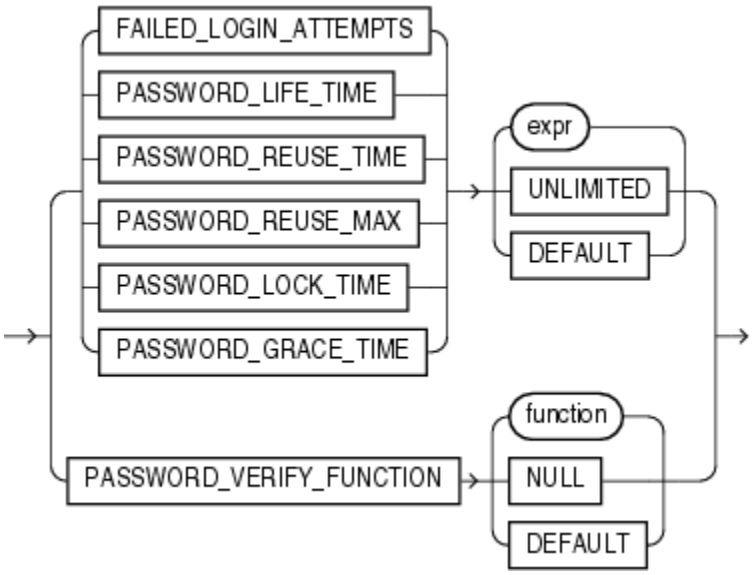
Create profile:-



RESOURCE PARAMETERS:-



Password parameters:-



Semantics

Profile

Specify the name of the profile to be created. Use profiles to limit the database resources available to a user for a single call or a single session.

Oracle Database enforces resource limits in the following ways:

- If a user exceeds the CONNECT_TIME or IDLE_TIME session resource limit, then the database rolls back the current transaction and ends the session. When the user process next issues a call, the database returns an error.
- If a user attempts to perform an operation that exceeds the limit for other session resources, then the database aborts the operation, rolls back the current statement, and immediately returns an error. The user can then commit or roll back the current transaction, and must then end the session.
- If a user attempts to perform an operation that exceeds the limit for a single call, then the database aborts the operation, rolls back the current statement, and returns an error, leaving the current transaction intact.

Notes:

- You can use fractions of days for all parameters that limit time, with days as units. For example, 1 hour is 1/24 and 1 minute is 1/1440.
- You can specify resource limits for users regardless of whether the resource limits are enabled. However, Oracle Database does not enforce the limits until you enable them.

UNLIMITED

When specified with a resource parameter, UNLIMITED indicates that a user assigned this profile can use an unlimited amount of this resource. When specified with a password parameter, UNLIMITED indicates that no limit has been set for the parameter.

DEFAULT

Specify DEFAULT if you want to omit a limit for this resource in this profile. A user assigned this profile is subject to the limit for this resource specified in the DEFAULT profile. The DEFAULT profile initially defines unlimited resources. You can change those limits with the ALTER PROFILE statement.

Any user who is not explicitly assigned a profile is subject to the limits defined in the DEFAULT profile. Also, if the profile that is explicitly assigned to a user omits limits for some resources or specifies DEFAULT for some limits, then the user is subject to the limits on those resources defined by the DEFAULT profile.

Profile management is divided into

a. **Resource Parameters**
b. **Password Parameters**

Resource parameters

The following parameters used for resource

SESSIONS_PER_USER:-Specify the number of concurrent sessions to which you want to limit the user.

CPU_PER_SESSION:-Specify the CPU time limit for a session, expressed in hundredth of seconds.

CPU_PER_CALL:-Specify the CPU time limit for a call (a parse, execute, or fetch), expressed in hundredths of seconds.

CONNECT_TIME:-Specify the total elapsed time limit for a session, expressed in minutes.

IDLE_TIME:-Specify the permitted periods of continuous inactive time during a session, expressed in minutes. Long-running queries and other operations are not subject to this limit.

LOGICAL_READS_PER_SESSION:-Specify the permitted number of data blocks read in a session, including blocks read from memory and disk.

LOGICAL_READS_PER_CALL:-Specify the permitted number of data blocks read for a call to process a SQL statement (a parse, execute, or fetch).

PRIVATE_SGA:-Specify the amount of private space a session can allocate in the shared pool of the system global area (SGA).

This limit applies only if you are using shared server architecture. The private space for a session in the SGA includes private SQL and PL/SQL areas, but not shared SQL and PL/SQL areas.

COMPOSITE_LIMIT:-Specify the total resource cost for a session, expressed in service units. Oracle Database calculates the total service units as a weighted sum of CPU_PER_SESSION, CONNECT_TIME, LOGICAL_READS_PER_SESSION, and PRIVATE_SGA.

Password parameters

Use the following clauses to set password parameters. Parameters that set lengths of time are interpreted in number of days. For testing purposes you can specify minutes (n/1440) or even seconds (n/86400). The following are parameters for password policy management

FAILED_LOGIN_ATTEMPTS:-Specify the number of failed attempts to log in to the user account before the account is locked. If you omit this clause, then the default is 10 days.

PASSWORD_LIFE_TIME:-Specify the number of days the same password can be used for authentication. If you also set a value for PASSWORD_GRACE_TIME, then the password expires if it is not changed within the grace period, and further connections are rejected. If you omit this clause, then the default is 180 days.

PASSWORD_REUSE_TIME and PASSWORD_REUSE_MAX:-These two parameters must be set in conjunction with each other. PASSWORD_REUSE_TIME specifies the number of days before which a password cannot be reused. PASSWORD_REUSE_MAX specifies the number of password changes required before the current password can be reused. For this parameter to have any effect, you must specify an integer for both of them.

- If you specify an integer for both of these parameters, then the user cannot reuse a password until the password has been changed the password the number of times specified for PASSWORD_REUSE_MAX during the number of days specified for PASSWORD_REUSE_TIME.

  For example, if you specify PASSWORD_REUSE_TIME to 30 and PASSWORD_REUSE_MAX to 10, then the user can reuse the password after 30 days if the password has already been changed 10 times.

- If you specify an integer for either of these parameters and specify UNLIMITED for the other, then the user can never reuse a password.
- If you specify DEFAULT for either parameter, then Oracle Database uses the value defined in the DEFAULT profile. By default, all parameters are set to UNLIMITED in the DEFAULT profile. If you have not changed the default setting of UNLIMITED in the DEFAULT profile, then the database treats the value for that parameter as UNLIMITED.
- If you set both of these parameters to UNLIMITED, then the database ignores both of them. This is the default if you omit both parameters.

PASSWORD_LOCK_TIME:-Specify the number of days an account will be locked after the specified number of consecutive failed login attempts. If you omit this clause, then the default is 1 day.

PASSWORD_GRACE_TIME:-Specify the number of days after the grace period begins during which a warning is issued and login is allowed. If you omit this clause, then the default is 7 days.

PASSWORD_VERIFY_FUNCTION:-The PASSWORD_VERIFY_FUNCTION clause lets a PL/SQL password complexity verification script be passed as an argument to the CREATE PROFILE statement. Oracle Database provides a default script, but you can create your own routine or use third-party software instead.

- For function, specify the name of the password complexity verification routine.
- Specify NULL to indicate that no password verification is performed..

Examples

Creating a Profile:

SQL>create profile new_profile limit password_reuse_max 10 password_resuse_time 30;

Setting Profile Resource Limits:

SQL>create profile app_user limit sessions_per_user unlimited cpu_per_session unlimited cpu_per_call 3000 connect_time 45 logical_reads_per_session default logical_reads_per_call 1000 private_sga 15k composite_limit 5000000;

If you assign the app_user profile to a user, then the user is subject to the following limits in subsequent sessions:

- The user can have any number of concurrent sessions.
- In a single session, the user can consume an unlimited amount of CPU time.
- A single call made by the user cannot consume more than 30 seconds of CPU time.
- A single session cannot last for more than 45 minutes.
- In a single session, the number of data blocks read from memory and disk is subject to the limit specified in the DEFAULT profile.
- A single call made by the user cannot read more than 1000 data blocks from memory and disk.
- A single session cannot allocate more than 15 kilobytes of memory in the SGA.
- In a single session, the total resource cost cannot exceed 5 million service units. The formula for calculating the total resource cost is specified by the ALTER RESOURCE COST statement.
- Since the app_user profile omits a limit for IDLE_TIME and for password limits, the user is subject to the limits on these resources specified in the DEFAULT profile.

Setting Profile Password Limits:

SQL>create profile app_user2 limit failed_login_attempts 5 password_life_time 60 password_reuse_time 60 password_reuse_max 5 password_verify_function verify_function password_lock_time 1/24 password_grace_time 10;

This example uses the default Oracle Database password verification function.

AUDITING

1. It is the process of recording user actions in the database
2. Auditing is enabled by setting AUDIT_TRAIL to
    a. None – no auditing enabled (default)
    b. DB – audited information will be recorded in AUD$ table and can be viewed using DBA_AUDIT_TRAIL view
    c. OS – audited information will be stored in the form of trace files at OS level. For this we need to set AUDIT_FILE_DEST parameter
    d. DB, Extended – it is same as DB option but will still record info like SQL_TEXT, BIND_VALUE etc
    e. XML – it will generated XML files to store auditing information
    f. XML, Extended – same as XML but will record much more information
3. By default some activites like startup & shutdown of database, any structural changes to database are audited and recorded in alert log file
4. If auditing is enabled with DB, then we need to monitor space in SYSTEM tablespace as there is a chance of getting full when more and more information is keep on recorded
5. SYS user activities can also be captured by setting AUDIT_SYS_OPERATIONS to TRUE
6. Auditing should use following scopes
    a. Whenever successful / not successful
    b. By session / By access
7. Disadvantage of auditing is, it will not capture changed values. For that DBA will use triggers. This is replaced with Fine Grained Auditing (FGA) which will capture old and new values when a record is modified
8. FGA can be initiated using DBMS_FGA and by creating and setting audit policies. Information that is captured can be viewed using DBA_FGA_AUDIT_TRAIL view
9. Enabling auditing at database level will have adverse impact on the database performance

Practical on User Management

1. Creating the user

SQL> Create user <username> identified by <password> default tablespace hz_ts

temporary tablespace temp

quota unlimited on hz_ts

quota 100m on hz_demo_ts

(hz_ts,temp,hz_demo_ts: tablespce names)

2. Assigning Tablespace Quotas

You can assign each user a tablespace quota for any tablespace (except a temporary tablespace). Assigning a quota does two things:

- Users with privileges to create certain types of objects can create those objects in the specified tablespace.
- Oracle limits the amount of space that can be allocated for storage of a user's objects within the specified tablespace to the amount of the quota.

By default, a user has no quota on any tablespace in the database. If the user has the privilege to create a schema object, you must assign a quota to allow the user to create objects. Minimally, assign users a quota for the default tablespace, and additional quotas for other tablespaces in which they can create objects.

You can assign a user either individual quotas for a specific amount of disk space in each tablespace or an unlimited amount of disk space in all tablespaces. Specific quotas prevent a user's objects from consuming too much space in the database.

You can assign a user's tablespace quotas when you create the user, or add or change quotas later. If a new quota is less than the old one, then the following conditions hold true:

- If a user has already exceeded a new tablespace quota, the user's objects in the tablespace cannot be allocated more space until the combined space of these objects falls below the new quota.
- If a user has not exceeded a new tablespace quota, or if the space used by the user's objects in the tablespace falls under a new tablespace quota, the user's objects can be allocated space up to the new quota.

2.1. Assigning the quota to the user on other tablespace

SQL> alter user hz quota 10m on tbs1;

3. To unlock the user account

SQL> alter user hz account unlock;

4. To force user to change the Password

SQL> alter user hz password expire;

SQL>grant connect,resource to tz identified by tz**;**

5. Granting privileges to users

SQL> grant create session, create table, create sequence to hz;

SQL> conn hz/hz;

SQL> select * from session_privs;

SQL> select * from role_sys_privs;

6. Creating and assigning the custom roles:

SQL> create role role1;

SQL> grant create session, create table, create sequence to role1;

SQL> grant role1 to hz;

SQL> select role,privilege from role_sys_privs where role='ROLE1';

sql

SQL>select * from dba_roles_privs where granted_role='ROLE1'

7. Assingning object privileges on a table to other users

SQL> connect hz/hz

SQL> grant select,insert,update on emp to tz; //emp:table

SQL> connect tz/tz;

SQL> select * from hz.emp;

SQL>select grantee,owner,grantor,table_name,privilege from dba_tab_privs where owner='HZ';

8. Checking the privileges for the user

SQL> connect / as sysdba;

SQL> select owner,role,privilege from role_tab_privs where owner='HZ';

SQL> select grantee,owner,table_name from dba_tab_privs where grantee='HZ2';

9. Revoking the privileges from user

SQL> revoke insert,update on hz.emp from hz2; //hz.emp=>user.table

10. Some the key seeded (predefined) roles:

dba, connect,resource, exp_full_database, imp_full_database etc

11. Dropping the user Account

Specify the user to be dropped. Oracle Database does not drop users whose schemas contain objects unless you specify CASCADE or unless you first explicitly drop the user's objects.

SQL> drop user hz;

11.1. Cascade:

Specify **CASCADE** to drop all objects in the user's schema before dropping the user. You must specify this clause to drop a user whose schema contains any objects.

- If the user's schema contains tables, then Oracle Database drops the tables and automatically drops any referential integrity constraints on tables in other schemas that refer to primary and unique keys on these tables.
- If this clause results in tables being dropped, then the database also drops all domain indexes created on columns of those tables and invokes appropriate drop routines.

SQL> drop user hz cascade;

12. To check default parameter tablespace and temporary tablespace :

SQL> select PROPERTY_NAME,PROPERTY_VALUE from database_properties where property_nmae like 'DEFAULT';

13. To change default permanent tablespace :

SQL>alter database default temporary tablespace emp;

14. To change default temporary tablespace :

SQL>alter database default temporary tablespace mytemp;

15. To check system privileges for a user :

SQL>select privilege from dba_sys_privs where grantee='SCOTT';

16. Views:

- dba_users;
- dba_role_privs;
- user_users;
- dba_sys_privs;
- all_users;
- dba_tab_privs;
- dba_ts_quotas;
- role_tab_privs;
- user_ts_quots
- dba_roles;
- role_sys_privs;
- role_sys_privs;

Practical on Profile and Session Management

1. Profile Management

1.1. Creating profile

Sql> create profile prof1 limit //prof1:name of profile
Failed_login_attempts 3 (no of attempts)
Password_lock_time 1 (no of days) 1/24 (for 1 hour)
Password_life_time 7 (no of days)
Sessions_per_user 5 (no of total sessions)
Idle_time 1 (in minutes)
Connect_time 600; (10 hours)

1.2. Assigning profile

Sql> create user hz idenfied by hz profile prof1; //prof1:name of profile
Sql> alter user hz profile prof1;
SQL>select * from dba_profiles where profile='prof1';
SQL>select username,profile from dba_users where profile='PROF1';
SQL>alter profile prof1 limit password_lock_time 2;
SQL>select * from dba_profiles where profile='PROF1';

- To enforce kernel/resource parameters the following parameter must be set
  Sql> alter system set resource_limit=true scope=both;

1.3. Applying password restriction in profile:

Sql> @$ORACLE_HOME/rdbms/admin/utlpwdmg.sql;
SQL>alter profile default limit password_verify_function null;
Sql> alter profile prof1 limit Password_verify_function verify_function;
Sql> alter profile prof1 limit Password_verify_function null;

- **By default password is case sensitive, to disable it set the following parameter to false**

Sec_case_sensitive_logon=false;

SQL>alter system set sec_case_sensitive_logon=true scope=both;

2. Session Management

2.1. Monitoring

sql> select count(*) from v$session Where username is not null;

Sql> select username,sid,serial# from v$session whrere username is not null;

2.2. To get session details:

Sql> select username,sid,serial#,status,logon_time from v$session;

2.3. Userswise sessions:

Sql> select username,count(*) from v$session group by username;

2.4. Joining v$process & v$session

Sql>select a.sid,a.serial#,a.username,b.pid,b.spid from v$session a,v$process b where a.paddr=b.addr and a.username =’HZ’;

Sql>select username,to_char(logon_time,’hh24:mi:ss dd-mon-yy’),sid,serial# from v$session where sid=’146’;

2.5. Killing session at database level :

Syntax:

alter system kill session (‘SID,SERIAL#’)

ex. Sql>Alter system kill session ‘146,4’;

2.6. Killing session from os level :

At os level:

$ps –ef |grep 3657

Kill -9 3657

3. Important v$ views:

- V$process
- v$session
- v$sqlarea
- v$sqltext
- v$lock
- V$session_wait
- V$sess_io

Practical on auditing

 # To enable auditing

SQL> alter system set audit_trail=DB/OS/XML scope=spfile;

Note: AUDIT_TRAIL parameter is static and require a restart of database before going to be effective

To audit what is required

SQL> Audit update on table SALARY;

SQL> Audit all by scott;

SQL> Audit session by scott;

SQL> Audit all privileges;

To turn off auditing

SQL> Noaudit session;

SQL> Noaudit update on table SALARY;

SQL> Noaudit all privileges;

ORACLE NETWORK CONFIGURATION

NETWORKING WITH ORACLE

1. We need to have oracle client software installed on client machine in order to connect to server
2. The following files are required to establish a successful connection to the server
    a. Client – TNSNAMES.ORA and SQLNET.ORA
    b. Server – LISTENER.ORA, TNSNAMES.ORA and SQLNET.ORA
3. TNSNAMES.ORA file contains the description of the database to which connection should establish
4. SQLNET.ORA will define the type of connection between client and server
5. Apart from using tnsnames.ora we can also use EZCONNECT, LDAP, bequeath protocol etc to establish connection to server
6. These files will reside $ORACLE_HOME/network/admin
7. LISTENER service will run based on LISTENER.ORA file and we can manage listener using below commands
    a. $ lsnrctl start / stop / status / reload
8. We can have host string different from service name i.e instance name or SID and even it can be different from database name. This is to provide security for the database.
9. Tnsping is the command to check the connectivity to the database from client machine
10. Do create seperate listeners for multiple databases
11. If the connections are very high, create multiple listeners for the same database
12. Any network related problem should be resolved in the following steps
    a. Check whether listener is up and running or not on server side
    b. Check the output is ok for tnsping command
    c. If still problem exist, check firewall on both client and server. If not known take the help of network admin
13. We can know the free port information from netstat command
14. We need to set password for listener so as to provide security
15. TNSNAMES.ORA and SQLNET.ORA files can also be seen on server side because server will act as client when connecting to another server
16. If listener is down, existing users will not have any impact. Only new users cannot be able to connect to instance
17. From 10g, SYSDBA connection to database will use bequeath protocol as it doesn't require any protocol like TCP/IP

Error 1: Connect failed because target host or object does not exist

Solution : goto run->drivers->etc->hosts file and open it with notepad. Then do add a entry

Note : If it is unix machine, check in /etc/hosts

Error 2 : protocol adapter error

Solution : Generally it will occur if network connectivity is not there. So ping the server and if its working fine then check if firewall is enabled either at client side or server side.

Error 3 : connection description doesn't found

Solution : check tns entry in tnsnames.ora for any syntax error

Note : port number in listener.ora and tnsnames.ora should be same

Creating new Listener

1. Copy the existing entry to end of the listener.ora file

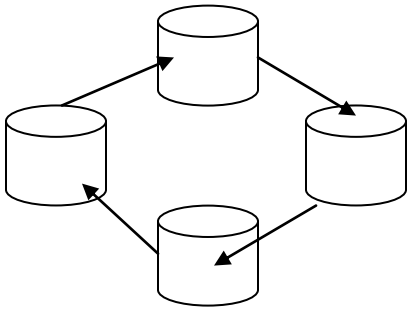
```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /u02)
      (PROGRAM = extproc)
    ))
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = server1.kanna.com)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC0))
    ))
```

2. Remove the line mentioned “protocol=IPC” in listener description in listener.ora
3. Change listener name, host and port no to appropriate values
4. Change the listener name in the first line of SID_LIST
5. Change SID_NAME and ORACLE_HOME and remove EXTPROC line, when finished it should look like below

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = prod)
      (ORACLE_HOME = /u02)
    ))
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = server1.kanna.com)(PORT = 1521))
    ))
```

Distributed Database Management

1. Having multiple databases connected to each other is called distributed database management system



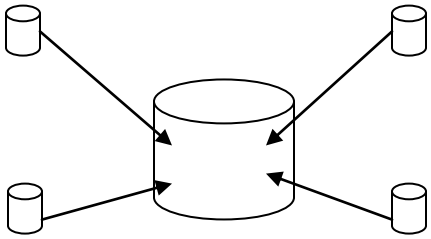
2. Even though manageability is complex, having multiple small databases will give more benefit
3. DDMS will use two phase commit mechanism i.e. a transaction should be committed in both the databases before the data was made permanent
4. DDMS can have different databases like oracle, db2, sql server etc. In this case they will talk each other using oracle gateway component (which need to be configured seperately)

Database Links

1. It is the object which will pull remote database data to local database
2. While creating dblink, we need to know username and password of remote database
3. Following is the example for creating db link
    a. Sql> create database link dev.com
        Connect to scott identified by tiger
        Using 'dev';
4. Following query will show how to retrieve data from remote database
    a. Sql> select * from scott.emp@dev.com;
5. Apart from username and password of remote db, we need to have tns entry in tnsnames.ora of local db and tnsping command should work

Materialized Views

1. It is an object used to pull remote database's data frequently in specified time which is called as refreshing the data using materialized views



2. Snapshot is the object which used to do the same till 8i, but the disadvantage is time constraint in pulling huge no.of rows
3. MV uses MV log to store information of already transferred rows. MVLOG will store rowid's of table rows which helps in further refresh
4. MV should be created in the database where we store the data MVLOG will be created automatically in remote database
5. MVLOG is a table not a file
6. MV refresh can happen in following three modes
    a. Complete – pulling entire data
    b. Fast – pulling non transferred rows
    c. Force – it will do fast refresh and in case any failure, it will go for complete refresh
7. When MV refresh happening very slow, check the size of table and compare that with MVLOG size
8. If MVLOG size is more than table size, then drop and recreate only MVLOG

Note: A complete refresh is required after the recreation of MVLOG