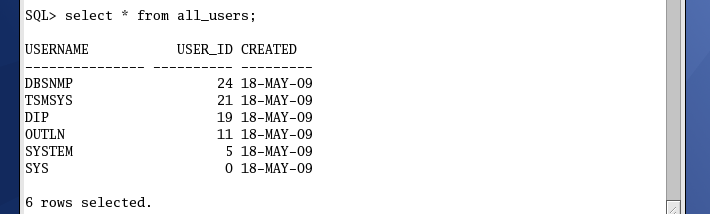
***USER MANAGEMENT AND SECURITY***

A database without user, there is no meaning. So there must be a user that can use the database. When you create a database by default some users automatically created. When you create a database manually the following are the six users created automatically.



Above users are created automatically. Among these users, two users SYS and SYSTEM are very important users,

SYS- SYS user is a database user and an owner of the oracle database.

SYSTEM- SYSTEM user is another database user, which have DBA role with admin option, and performs administrative tasks.

We cannot rename these two users, but we can change its password.

CREATE DBA\_USERS

ALTER

DROP ALL\_USERS

USER\_USERS

VIEWS

SQL

Network Authentication

AUTHENTICATION

Operating System Authentication

Database Authentication

Default

PROFILE

User defined

PRIVILEGES

System Privileges

Default

ROLE

Object Privileges

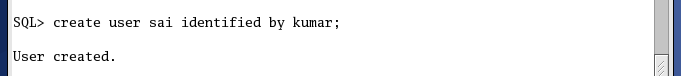
User defined

***1.How we create a database user***

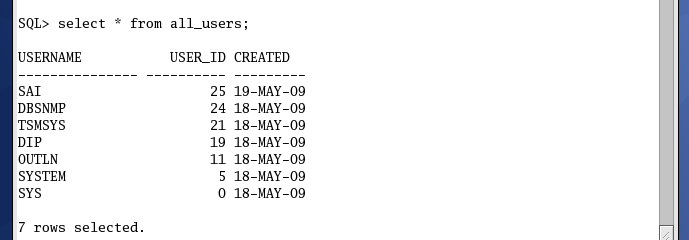
Syntax for creating a user:

SQL>> create user sai identified by kumar;

Here sai is a username kumar is password .



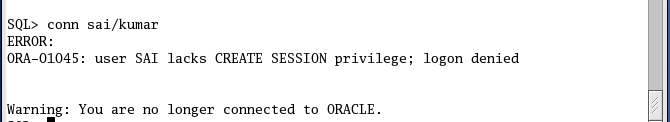
***Check whether user is created or not***



Yes, user is created.

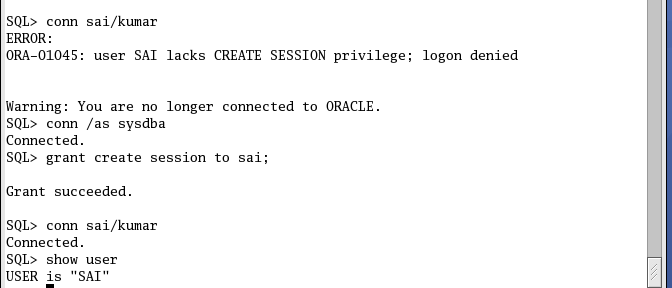
***GIVING PERMISSIONS TO THE USERS***

A newly created user doesn’t have any privilege to connect or to do anything in the database.



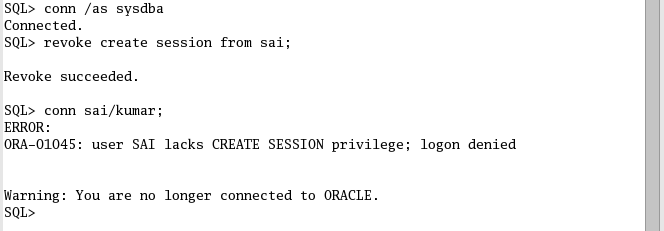
It is clearly showing user sai doesn’t have create session privilege.

With the help of GRANT command DBA people give the create session permission to the user as…



***Revoking permissions from users***

To revoke permissions from user user Revoke command



**Drop User**

To drop a user you must be a DBA or have drop user permission.

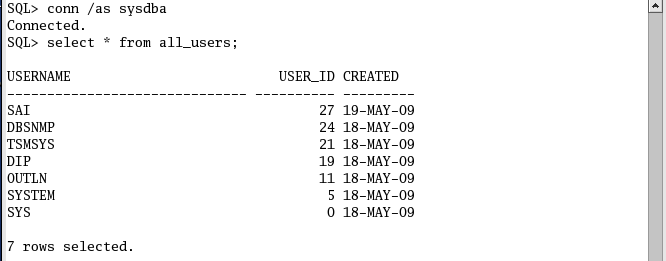
Syntax:

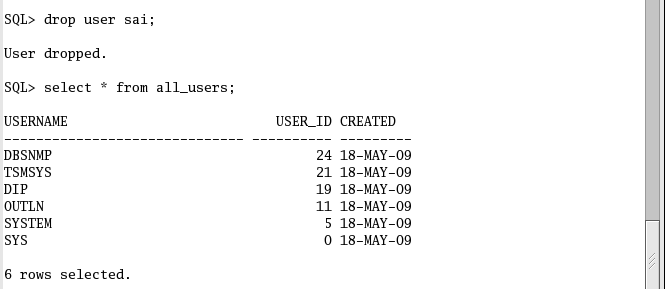
SQL>> Drop user sai;

Or

IF user contains any object use cascade option with drop command

SQL>> Drop user sai cascade;





User sai has been droped.

**What is Privilege**

Privilege means permissions to perform some actions (to execute SQL statement) against database or table.

Lack of permissions a simple user cannot do anything’s in the database even cannot connect in the database.

For security purposes we manage the privilege at different level. The following are the two types of privilege

1. SYSTEM PRIVILEGE
2. OBJECT PRIVILEGE

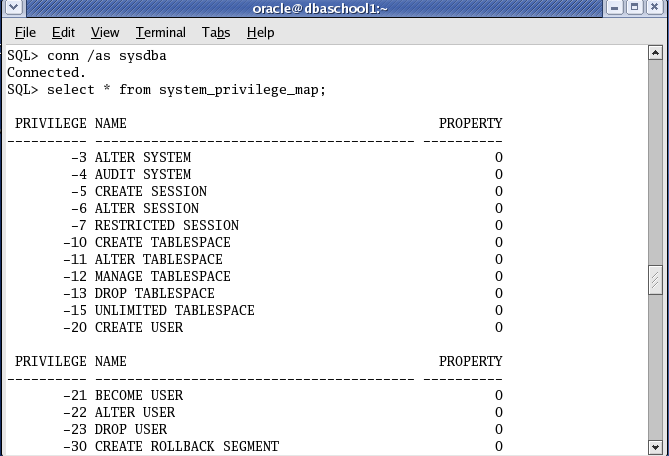
***SYSTEM PRIVILEGE***

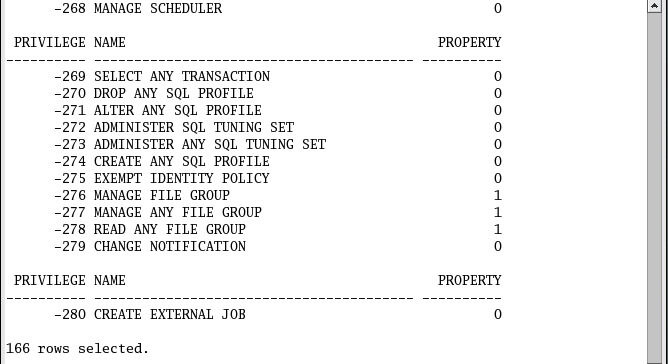
System privileges are those privileges which enables us to perform actions (to execute SQL statement) against the database. The following are the some system privileges…

1. Create table
2. Create tablespace
3. Create user
4. Create session
5. Drop user ….

*Use the following syntax to see the entire system privilege*

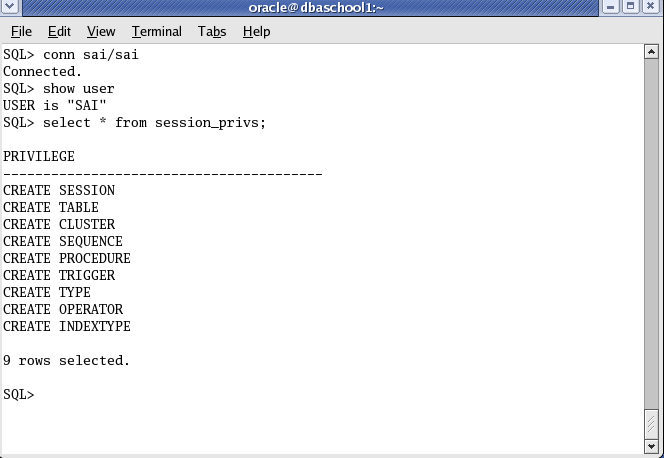
SQL>> select \* from system\_privilege\_map;



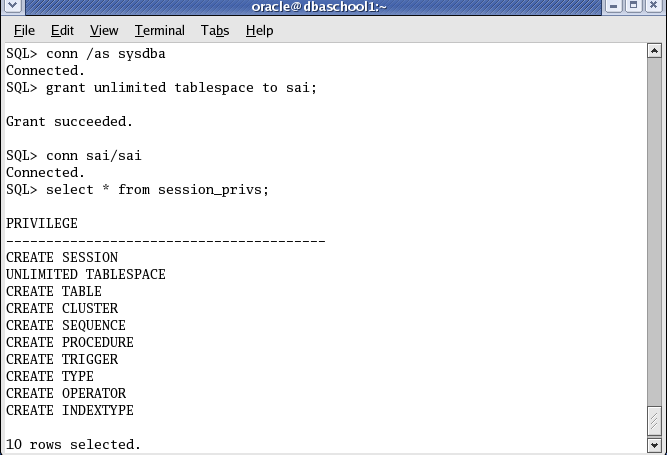


*To see all system privileges for a particular session.*

SQL>>select \* from session\_privs;



Use grant command to give a system level privilege to a user and user revoke command to return the granted privileges from the user



***OBJECT PRIVILEGE***

Object level privileges are those privileges which enables us to perform action against the database tables (i.e. to execute SQL statement against a table).

The following are the some object level privileges

1. Select
2. Insert
3. Update
4. Delete

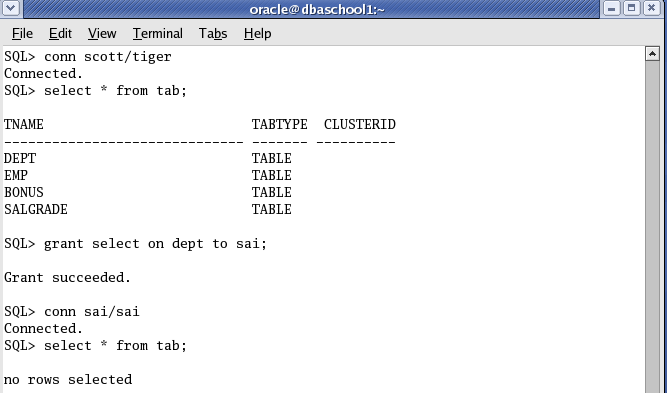
Only DBA people or Owner of the table can grant the object level privilege to another user.

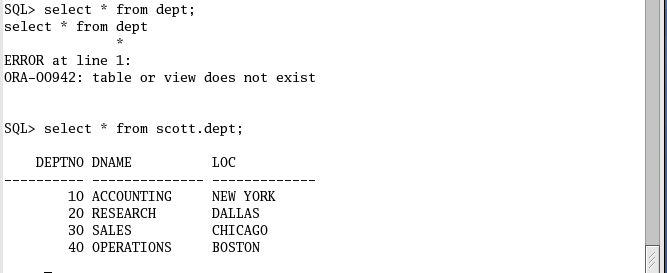
Example:

By owner,

SQL>>connect scott/tiger

SQLL>>grant select on dept to sai ;

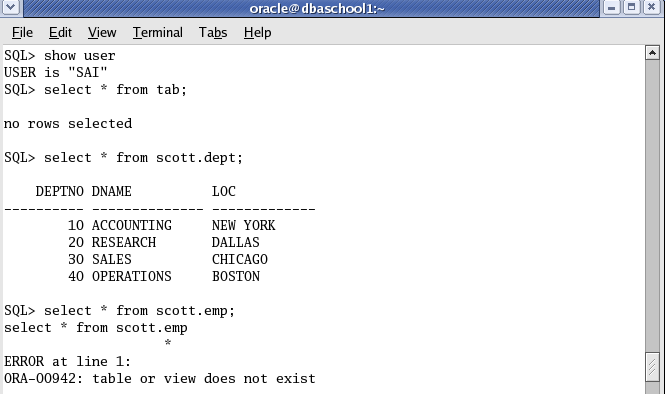


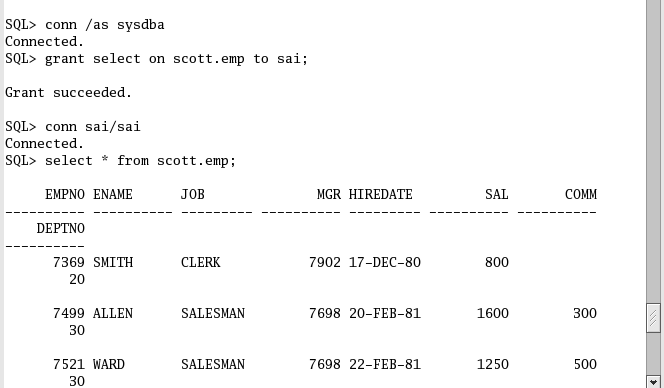


By DBA,

SQL>>connect / as sysdba

SQL>> grant select on scott.dept to sai;





Roles: A role is a database objects. It acts just like a logical container that contains a set of privilege.

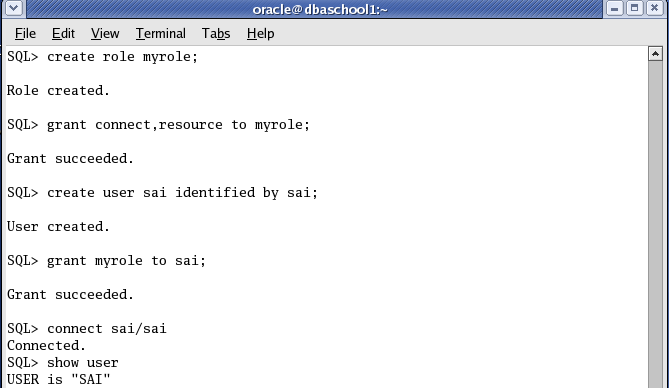
Why use role - For better management of privileges.

***How to create a role***

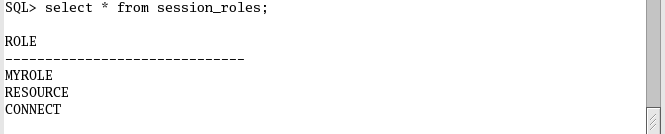
SQL>>Create role myrole;

SQL>>grant create session, create table , unlimited tablespace to myrole.

SQL>>grant myrole to sai ;



To see all roles granted to a particular user



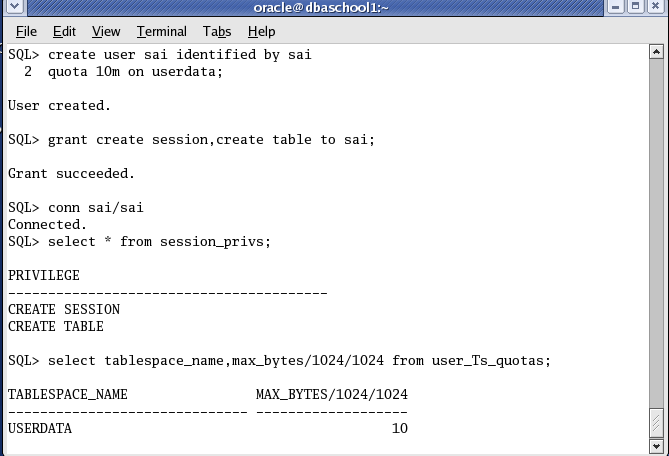
To drop a role



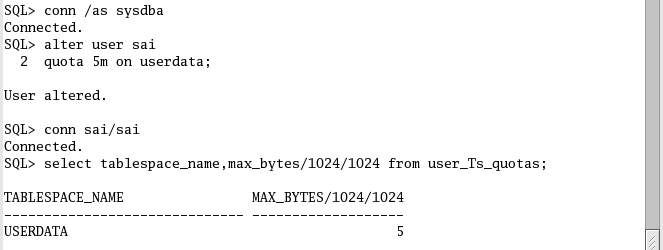
***QUOTA:*** By using quota we limit the certain amount of storage space in the tablespace. We can do this when we create a user or later by using alter user command.

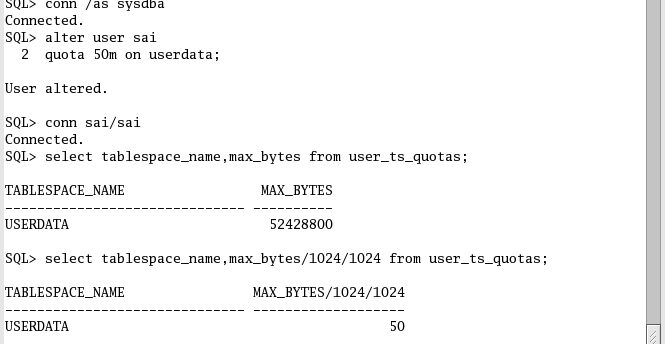
View related to quota

1. DBA\_TS\_QUOTAS
2. USER\_TS\_QUOTAS



***We can increase /decrease quota on the tablespace.***

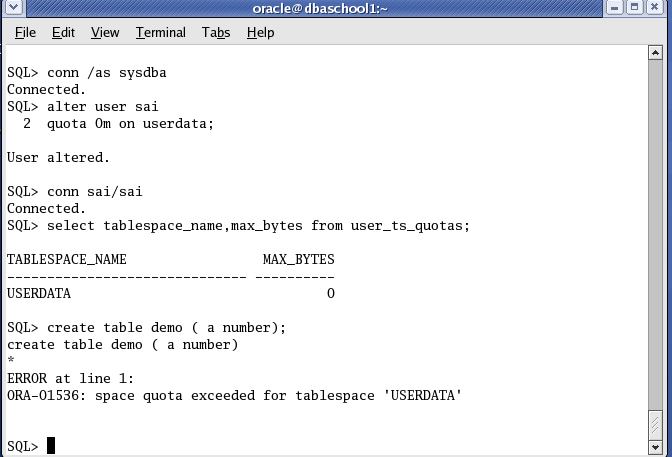




***To give unlimited quota on Tablespace***



***To return all quota from a user***



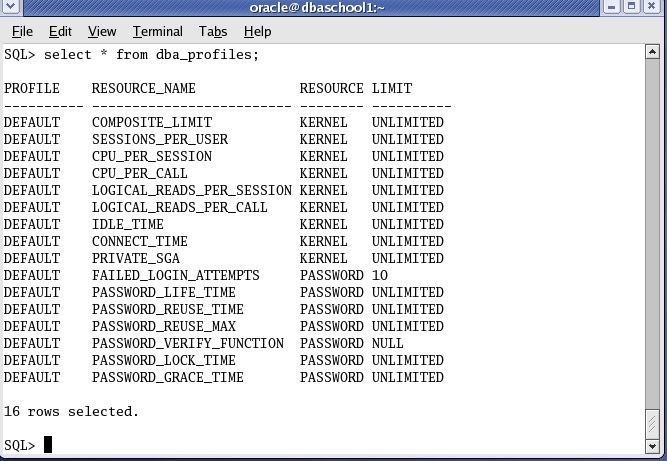
***PROFILE***

A Profile is a collection of resource usage and password related attributes which is to assign a user. It is used to manage and control the system resources which are used by a user and verify the password for security purposes.

When a user is created, it is automatically assigned to a default profile.

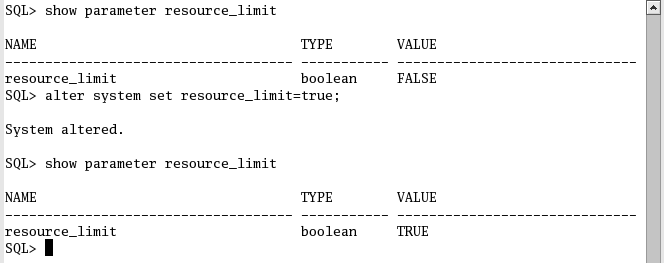
***View related to Profile***

DBA\_PROFILES



It is clear that two type of resource is maintained in profile kernel type and password type. To enable kernel type resource parameter. We must have to enable the following paramter

**resource\_limit=true**



***Resource Parameter***

SESSIONS\_PER\_USER:- It specifies maximum number of Concurrent sessions can be opened by a user.

CONNECT\_TIME:- It specifies total time a session may remain connected to the database (IN MINUTES).

IDLE\_TIME:- It specifies to logout automatically from a session, if session is idle with the specified limit (IN MINUTES).

CPU\_PER\_CALL: Limits the CPU time used per each call within a transaction( for the parse, execute, fetch)

CPU\_PER\_SESSION: Limits the total CPU time used during a session.

LOGICAL\_READS\_PER\_SESSIONS: It specifies the limits of the total number of data block read from the sga

LOGICAL\_READS\_PER\_CALL: It specifies the limits of the total number of data block read from the sga per each session call

PRIVATE\_SGA: It specifies session’s limit on the space it allocated in the shared pool component of the sga (applicable only to shared server architecture)

COMPOSITE\_LIMIT: It specifies overall limits on the resources.

***Password Parameter***

FAILED\_LOGIN\_ATTEMPTS: It specifies number of times a user can attempt to login in before being locked out.

PASSWORD\_LIFE\_TIME: It specifies the password will expire after the specified time.

PASSWORD\_REUSE\_TIME: Specifies the number of days must pass before you can reuse the same password

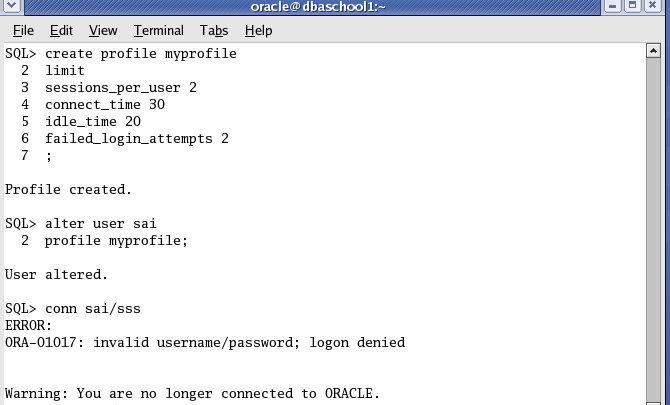
PASSWORD\_REUSE\_MAX: While there is a need to change your password, you can reuse the same password up to the limit

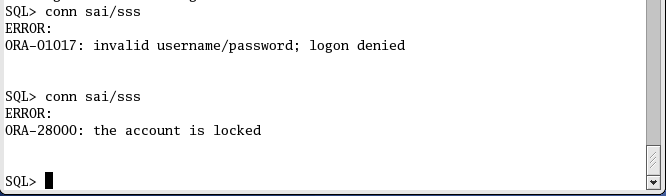
PASSWORD\_LOCK\_TIME: It specifies the number of days an account of a user will be locked

PASSWORD\_GRACE\_TIME: It specifies the number of days a user can use the expired password.

PASSWORD\_VERIFY\_FUNCTION: For verifying the password

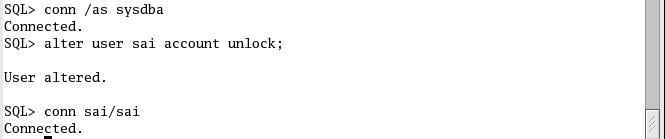
***HOW TO CREATE AND ASSIGN A PROFILE***



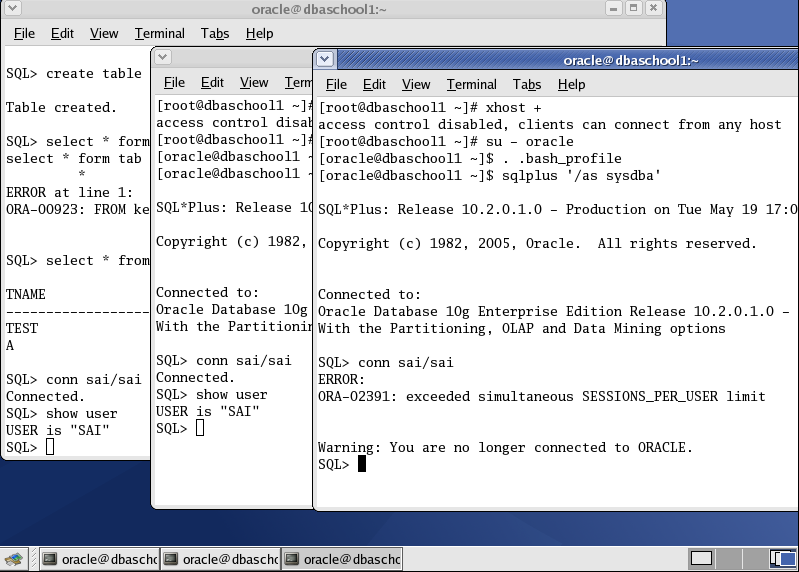


***User tried to more than 2 attempts to login before it is locked***

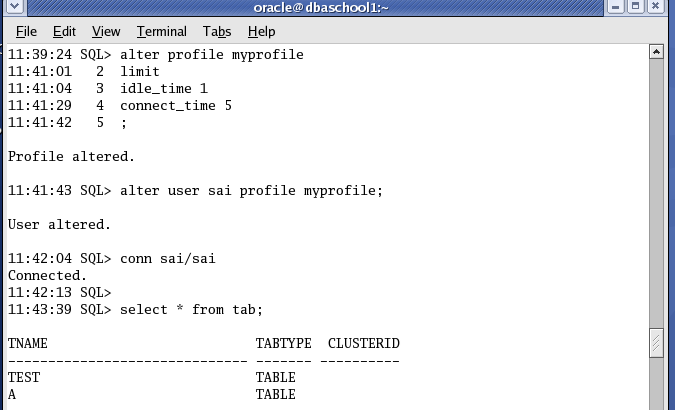
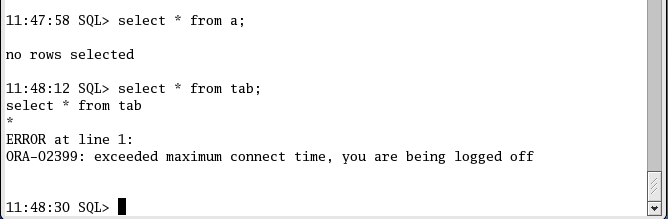
**To open locked account**



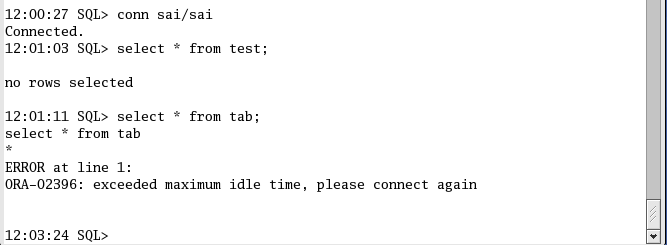
***EXAMPLE OF SESSIONS\_PER\_USER***

User cannot open concurrent session more than specified limit 

***EXAMPLE OF CONNECT\_TIME***

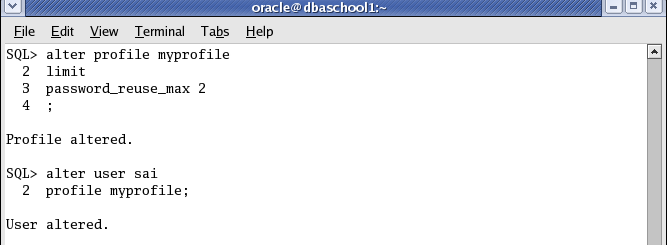
 

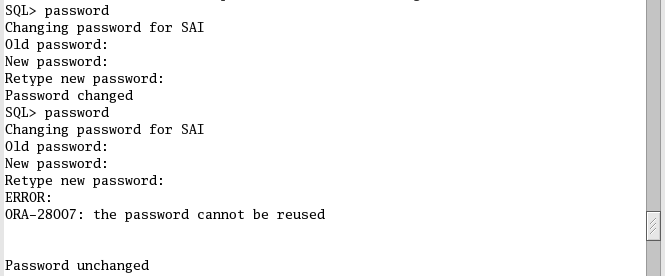
***EXAMPLE OF IDLE\_TIME***

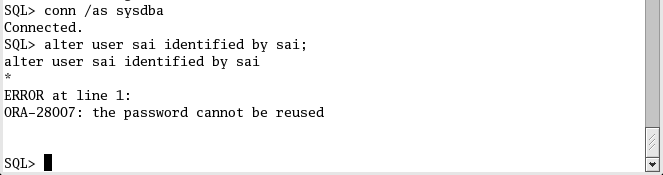


***EXAMPLE OF PASSWORD\_REUSE\_MAX***

While there is a need to change your password, you can reuse the same password up to the limit







***EXAMPLE OF PASSWORD\_REUSE\_TIME***

Specifies the number of days must pass before you can reuse the same password.

