**Database Management System**

**Practical No : 10(a)**

**Aim :**  Perform DCL commands using grant and revoke

* To access the database, a user must specify a valid databse user account and successfully authenticate as required by the user account.
* Each database user has a unique database account.
* Oracle recommends this to avoid potential security holes and provide meaningful data for certain audit activity.

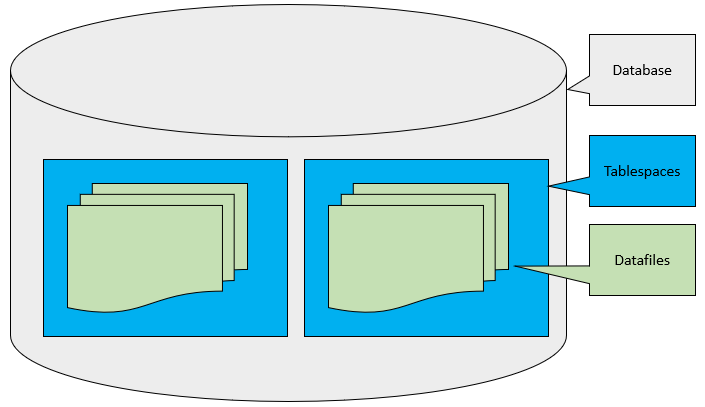
**Each database user has:**

* A unique username
* An authentication method
* A default tablespace
* A temporary tablespace
* A user profile
* An account status.

**Predefined Accounts:**

|  |  |
| --- | --- |
| **SYS** | Owns all internal tables |
| **SYSTEM** | Has additional tables and views |
| **SYSMAN** | Use by OEM to monitor and gather performance stats, which are stored in the ***sysaux*** tablespace |
| **DBSNMP** | Same as *sys* but for the OEM, owns all internal tables in the ***sysaux*** tablespace. |

Oracle divides a database into one or more logical storage units called tablespaces. Each tablespace consists of one or more files called datafiles. A datafile physically stores the data objects of the database such as tables and indexes on disk



The SYSTEM account is one of a handful of predefined administrative accounts generated automatically when Oracle is installed. SYSTEM is capable of most administrative tasks, but the task we’re particularly interested in is account management.

## **Creating a User**

**The below command is to check all the users inside the database:**

SQL> select username, account\_status, default\_tablespace from dba\_users;

**Check current user:**

SQL> show user;

**Lock/Unlock user:**

SQL> alter user scott account unlock;

SQL> alter user scott account lock;

To create user, first connect as SYSTEM, and simply issue the CREATE USER command to generate a new account.

create user username identified by password

PROFILE DEFAULT

DEFAULT TABLESPACE USERS

TEMPORARY TABLESPACE TEMP

;

CREATE USER books\_admin IDENTIFIED BY MyPassword;

Here we’re simply creating a books\_admin account that is IDENTIFIED or authenticated by the specified password.

## **Example:**

* create user ram identified by ram;(by default 125Mb in temporary tblspc)
* create user ravan identified by ravan default tablespace lanka;
* create user ravan identified by ravan default tablespace lanka quota 5m on lanka;

**Changing user password:**

* alter  user ravan identified by ravan1;

**Ulter space with limited and unlimited quota**

* alter  user ravan quota 25m on lanka;
* alter user ravan quota unlimited on lanka;

**View Quota allocation(DBA level and User level)**

* select \* from  dba\_ts\_quotas;
* select \* from user\_ts\_quotas;

***CREATE TABLESPACE tbs1 DATAFILE 'tbs1\_data.dbf' SIZE 100m;***

## **The Grant Statement**

With our new books\_admin account created, we can now begin adding privileges to the account using the [GRANT](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_9013.htm) statement. GRANT is a very powerful statement with many possible options, but the core functionality is to manage the privileges of both users and roles throughout the database.

## **Providing Roles**

Typically, you’ll first want to assign privileges to the user through attaching the account to various roles, starting with the CONNECT role:

GRANT CONNECT TO books\_admin;

In some cases to create a more powerful user, you may also consider adding the RESOURCE role (allowing the user to create named types for custom schemas) or even the DBA role, which allows the user to not only create custom named types but alter and destroy them as well.

GRANT CONNECT, RESOURCE, DBA TO books\_admin;

## **Assigning Privileges**

Next you’ll want to ensure the user has privileges to actually connect to the database and create a session using GRANT CREATE SESSION. We’ll also combine that with all privileges using GRANT ANY PRIVILEGES.

GRANT CREATE SESSION GRANT ANY PRIVILEGE TO books\_admin;

## **Table Privileges**

While not typically necessary in newer versions of Oracle, some older installations may require that you manually specify the access rights the new user has to a specific schema and database tables.

For example, if we want our books\_admin user to have the ability to perform SELECT, UPDATE, INSERT, and DELETE capabilities on the books table, we might execute the following GRANT statement:

GRANT

SELECT,

INSERT,

UPDATE,

DELETE

ON

schema.books

TO

books\_admin;

This ensures that books\_admin can perform the four basic statements for the books table that is part of the schema schema.

**Privilege  Description**

SELECT                Privilege to perform SELECT statements on the table.

INSERT                Privilege to perform INSERT statements on the table.

UPDATE              Privilege to perform UPDATE statements on the table.

DELETE               Privilege to perform DELETE statements on the table.

REFERENCES     Privilege to create a constraint that refers to the table.

ALTER                 Privilege to perform ALTER TABLE statements to change the table definition.

INDEX                 Privilege to create an index on the table with the create index statement.

ALL                      All privileges on table.

**Example:**

* SELECT username FROM dba\_users;
* Drop user a cascade;
* CREATE TABLESPACE tbs1 DATAFILE 'tbs1\_data.dbf' SIZE 100m;
* create user a identified by a default tablespace tbs1;
* alter user a quota 1m on tbs1;
* grant connect to a;
* grant create session, create table to a;

similarly, we create user **b** and **c.**

|  |  |  |
| --- | --- | --- |
| **System** | **a** | **b** |
| Above all stmt;  grant create view to a; | Create table t(c1 int);  Grant all on t to **b**; | Insert into a.t values(1);  Commit; then only everyone can see.  grant all on a.t to c; |

**Summary:**

|  |  |
| --- | --- |
| Creating | create user vallep identified by password; create user vallep identified by password default tablespace users temporary tablespace temp quota 100m on users; |
| Remove | drop user vallep; drop user vallep cascade;  Note: the cascade option will remove all the users objects as well. |
| Alter | alter user vallep idenitfied by newpassword; alter user vallep quota 200m on users; |
| Password options | alter user vallep identified by password; alter user vallep identified externally; alter user vallep identified globally as extname;  Note: identified by - the password will be kept in the data dicitonary identified externally - authenication will be performed by the O/S idenitified globally as extname - authenication will be performed by external app i.e radius  a user is only allowed to change is his/her password |
| Expire password | alter user vallep password expire; |
| Lock/unlock | alter user vallep account lock; alter user vallep account unlock; |
| Connecting | grant create session to vallep;  Note: this allows the user to connect to the database |
| Revoke access | revoke create session from vallep; |
| Quota | alter user vallep quota 100m on users; alter user vallep quota unlimited on users; grant unlimited tablespace to vallep; |

## **Exercise**

Create a Student table with following fields:

StudentCode, Name, Dateofbirth, Course, RollNo, Batch, Result.

Perform the following:

1. User F has retrieve over entire table
2. User S has insert and delete on entire table.
3. Every user has retrieve over his/her record only.
4. User N has retrieve over entire table and update on Course and RollNo only
5. User T has retrieve over Name, StudentCode and Result only.
6. User W has retrieve as T and update as N
7. User P has all the privileges for BE-IT student’s records.
8. User J has delete on records for student of Batch B2
9. User B has update and delete on students record of courses where there are no more than 5 student
10. User K has retrieve for Eldest and youngest student
11. Create a tablespace **user\_space** of 100Mb.
12. Change default tablespace of all users to ‘user\_space’
13. Allocate 10Mb Quota to each user.