CHAPTER

7

Security

Objectives

- To know the differences between system security and data security.
- To learn how to create and drop users, grant and revoke system privileges, create and drop roles, and grant and revoke roles.
- To learn how to grant and revoke privileges alter, delete, insert, select, update, index, and references on database objects such as tables and views.
- To know what is auditing and how to specify auditing options.

7.1 Introduction

Data stored in a database need protection from unauthorized access and manipulation. Multiuser database systems, such as Oracle, provide security features that control how a database is accessed and used. The control mechanism used in Oracle is called *discretionary access control* that regulates all user activities through privileges. A privilege is permission to perform an operation or access the data in the database in a prescribed manner; for example, permission to create a table, query a table or update a table. Privileges may be granted to users at the discretion of other users—hence the term discretionary access control.

Database security can be classified into two categories: *system security* and *data security*.

- *System security* controls the access and use of the database at the system level. It can grant or revoke privileges for the user to create session, create table, etc, and it can assign or alter CPU time, disk space for a user.
- *Data security* controls the access and use of the database at the schema object level. It can grant or revoke access to a specific schema object and the

specific types of actions allowed for each user on the schema object (e.g. the user can issue select statements on the Student table, but not insert, update, and delete statements).

7.1 System Security

System security includes creating users, authenticating users, allocating system resource to users, granting users with the system privileges to perform operations such as creating tables.

7.1.1 Creating Users

To access a database, you have to log on as a user. Each database user has a username identified by a password. DBA can create a user using the following command:

```
create user username identified by password;
```

> NOTE: The rules for naming user name and password are the same as for naming table names and view names. The first character cannot be a digit.

To drop a user, use the command:

```
drop user username [cascade];
```

The following statement drops user scott and all associated objects and foreign keys that depend on the tables owned by scott.

```
drop user scott cascade;
```

A user can change its password using the following command:

```
alter user username identified by newpassword;
```

7.1.2 Granting System Privileges

By default, a user has no privilege to use the database. An authorized user such as DBA grants system privileges to a user using the grant command:

```
grant privilege1 [, privilege2...] to username1 [, username2...];
```

For example, the following statement grants create table, create view, and create session privilege to users scott and liang:

```
grant create table, create view, create session
to scott, liang;
```

To revoke privileges, use the following syntax:

```
revoke privilege1 [, privilege2...] from username1 [, username2...];
```

> NOTE: There are more than 100 system privileges. For a complete list, including descriptions, of system privileges, see Tables 16-1 on Page 16-38 in *Oracle9i SQL Reference*.

7.1.3 Using Roles to Grant Privileges

For convenience, you can group the system privileges into *roles* and assign roles to the users. Here is an example of creating a role, adding privileges to the role, and granting the role to the users:

```
create role student_role;
grant create table, create view to student_role;
grant student_role to scott;
```

To revoke a role from a user, use the following syntax:

```
revoke rolename1 [, rolename2...] from username1 [, username2...];
```

To drop a role, use the following syntax:

```
drop role rolename [cascade];
```

Oracle provides several predefined roles. Two frequently used roles are *connect* and *dba*. The connect role contains the privileges create session, alter session, create table, create view, etc. You may grant the connect role to an ordinary user. The dba role contains the privileges that are granted to a DBA. For a complete list, including descriptions, of roles, see Tables 16-2 on Page 16-47 in *Oracle9i SQL Reference*.

> TIP: I recommend creating your own roles to meet your security requirements rather than relying on Oracle's predefined roles.

> NOTE: You can grant system privileges or roles to a user with admin option as in the following example:

```
grant role1 to scott with admin option;
```

> A grantee with admin option has additional privileges:
>
> - The grantee can grant or revoke the system privilege or role to or from *any* user or

other role in the database. Users cannot revoke a role from themselves.

- The grantee can further grant the system privilege or role with the admin option.

- The grantee of a role can alter or drop the role.

TIP: The DBA can view the privileges made to the roles and users by querying the DBA_SYS_PRIVS data dictionary view.

NOTE: To grant a system privilege, you must either have been granted the system privilege with the admin option or have been granted the grant any privilege system privilege.

To grant a role, you must either have been granted the role with the admin option or have been granted the grant any role system privilege, or you must have created the role.

7.1.4 Using Profiles to Set Resource Limit

In addition to system privileges, you can also set resource limits for the users using a *profile*. A user's profile limits database usage and instance resources as defined in the profile. Here is an example of creating a profile:

```
create profile student_profile limit
  sessions_per_user 20
  cpu_per_session unlimited
  cpu_per_call 6000
  logical_reads_per_session unlimited
  logical_reads_per_call 100
  idle_time 30
  connect_time unlimited;
```

A profile can be dropped using the following syntax:

```
drop profile profileName [cascade];
```

You can assign a profile to a user when a user is created using the create user statement or using the alter user statement. If a user is not assigned a profile, DBMS assigns a default profile to the user. Here is an example of assigning a new profile to a user using the alter user statement:

```
alter user liang
  profile student_profile;
```

NOTE: For profiles to take effect, resource limits must be turned on for the database as a whole. A profile can be created, assigned to users, altered, and dropped at any time by any authorized database user, but the resource limits set for a profile are enforced only when you enable resource limitation for the database. You can enable resource limitation enforcement using the following command:

```
alter system set resource_limit = true;
```

or disable it by using

```
alter system set resource_limit = false;
```

The alter system command does not permanently set the system properties. To permanently set resource_limit, add the line resource_limit = true or resource_limit = false in the init file for the database. The file for the database named liangora9i is at \oralce_home\admin\liangora9i\pfile\ora.init.

******End of NOTE***

7.2 Data Security

Data security includes the mechanisms that control the access to and use of the database at the object level. A user must have appropriate object privileges in order to access and manipulate a database object owned by another user. An authorized user can grant object privileges to other users using the following syntax:

```
grant objectprivileges [(columns)] | all
on objectname to user | role | public
[with grant option]
```

- Each type of object has different privileges associated with it. The privileges associated with a table are alter, delete, insert, select, update, index, and references. The references privilege enables the user to create a constraint that refers to the table.

  Note: For a complete list of objects and associated privileges, see Table 16-4, "Object Privileges and the Operations" on Page 16-49 in *Oracle9i SQL Reference*.

- For a table or view object, you can specify columns on which the insert and update privileges are to be granted. If columns are not specified, the insert and update privileges are granted to all columns. The select and delete privileges must be applied to all columns in a table or view.
- You can specify all in the syntax to grant all available object privileges for an object.
- Object privileges can be granted to users, roles, and public. Public means to all users.
- The with grant option clause can be used to grant object privileges to user and enables the user to grant the object privileges to other users and roles. This clause cannot be used to grant object privileges to roles.

NOTE: To grant an object privilege, you must be a DBA or own the object or the owner of the object must have granted you the object privileges with the grant option.

The following statement grants the select privilege and update privilege on the phone on the Student table to the role student_role and user Liang.

```
grant select, update (phone) on Student
to student_role, Liang;
```

The following statement grants all privilege on the Student table to the role student_role and user liang.

```
grant all on Student to student_role, Liang;
```

To revoke object privileges, use the following syntax:

```
revoke objectprivileges [(columns)] | all
on objectname from user | role | public
```

The following statement revokes all privileges on Student from role student_role and user liang.

```
revoke all on Student from student_role, liang;
```

7.3 Database Auditing

Auditing is to trace database activities for security purpose. A database system maintains an audit trail, which logs all changes (inserts/deletes/updates) to the database along with information such as which user performed the change and when the change was performed.

7.3.1 Specifying Audit Options

You can specify auditing options using the audit statement. The audit statement allows you to set audit options at three levels:

Statement: Causes auditing of specific SQL statements or groups of statements that affect a particular type of database object. For example, audit table command audits the actions on tables.

Privilege: Audits SQL statements that are authorized by the specified system privilege. For Example, audit create database link audits statements issued using the create database link system privilege.

Object: Audits specific statements on specific objects, such as alter table on the Faculty table.

The following statement sets the option to audit connections to and disconnections from the database.

```
audit session;
```

You can set this option selectively for individual users also, as in the next example:

```
audit session by scott, liang;
```

The following statement sets the option to audit privileges on table and views.

```
audit grant;
```

The following statement sets the option to audit select on Faculty.

```
audit select on Faculty;
```

You can remove the audit options using the noaudit command. For example, the following statement remove the audit session option.

```
noaudit session;
```

To remove all audit option, use

```
noaudit all;
```

> NOTE: Any authorized database user can set statement, privilege, and object auditing options at any time, but Oracle does not generate and store audit records in the audit trail unless database auditing is enabled. To enable database auditing, set the AUDIT_TRAIL initialization parameter to db in the database's initialization parameter file. By default, auditing is not enabled.

7.3.2 Viewing Audit Trail

The database audit trail is a single table named SYS.AUD$ in each Oracle database's data dictionary. Oracle provides many meaningful views to help you auditing information in this table, for example, you can use DBA_Audit_Trail to display all audit trail entries, use User_Audit_Trail to display all audit trail entries related to current user. You can use DBA_audit_Object to display all audit trail records for all objects in the system, and User_audit_Object to display all audit trail records for statements concerning objects that are accessible to the current user. You can use DBA_Audit_Session to list all audit trail records concerning connection to and disconnection from the database, and DBA_Audit_Session to display all audit trail records concerning connection to and disconnection from the database for current users.

Chapter Summary

This chapter introduced the concept of database security. You learned about system security and how to create database user, roles, and grant user privileges and roles. You also learned about data security and how to grant select, insert, delete, and update privileges on the tables or views to other users.

Review Questions

7.1 What is system security? What is data security?

7.2 What is a role? How do you create roles?

7.3 How do you grant or revoke system privileges to users? How do you grant or revoke data access privilege to users?