Security

As it was mentioned, DBMS enforces security measures for access control. There are various aspects of this enforcement. These are as follows.

1. Like any multi-user environment, only valid users can interact with the database. Database Administrator (DBA) will create the user accounts by executing the command
   CREATE USER user-id IDENTIFIED BY pwd
   It creates an account with the given use-id and password.
   DBA can remove an account also by executing the command:
   DROP USER user-id

2. What a user can do with the database it depends on **system level** and **object level privilege**.
   System level privilege:
   DBA can grant such privileges to the users by executing the command:
   GRANT privilege1, privilege2, …
   TO userid1, userid2, …
   Privileges may be CONNECT, RESOURCE, DBA. CONNECT privilege is the lowest level privilege one must have. It allows a user to connect with the database and work with the objects to which he has been given privilege (will see next in object level privilege). So, having this privilege is a must. RESOURCE privilege allows one to create his own objects (like, table, view, index). Having DBA privilege means that user can act as DBA. In order to take away the privilege, DBA executes the command:
   REVOKE privilege1, privilege2, …
   FROM userid1, userid2, …
   If CONNECT privilege is taken away then user can not work till it is granted again. So temporarily one can be blocked by taking away CONNECT privilege.
   Object level privilege:
   Object may be a table, index etc. One who creates the object is the owner of it. An owner can grant certain privileges on its objects to others. Thus others may have only CONNECT privilege, still they can work with the objects owned by others subjected to the privileges granted to him/her. Owner can grant the privileges to other users by executing the command:
   GRANT privilege1, privilege2, … |ALL
   ON objectname
   TO userid1, userid2, …. | PUBLIC
   [WITH GRANT OPTION] ← this clause is optional and applicable in Oracle

Privileges may be SELECT, INSERT, DELETE, UPDATE, REFERNCE, ALTER etc. Object may be table, view etc. If it is a view then possible privileges are SELECT, INSERT, DELETE, UPDATE (provided view is such that INSERT, DELETE, UPDATE are possible). Instead of specifying the individual privileges, one can say ALL which stands for all privileges. If privileges are to be granted to every user then one can PUBLIC instead of mentioning the user ids. Oracle allows the clause WITH GRANT OPTION. In that case, the user getting the privileges can also grant privileges (restricted to those privileges he/she has been granted) to others. In order to take away the privileges from users, owners will execute the command:

REVOKE privilege1, privilege2, … |ALL
ON objectname
FROM userid1, userid2, …

Access to tablespace [SPECIFIC TO ORACLE ONLY]:

Try to remember that we discussed about tablespace. In oracle, database is logically partitioned into tablespace. DBA creates the tablespace using CREATE TABLESPACE command and specifies the corresponding filenames in the disk. Between user and tablespace there is a many to many relation. A user can have access to multiple tablespaces and a tablespace can have multiple users accessing this. Data of the objects like table are stored into tablespace (physically goes to the corresponding file(s)). Thus, the tablespace (corresponding file(s)) stores data from multiple objects and those are from multiple users. But, how a user can get access to a tablespace? DBA grants it as follows.

GRANT RESOURCE (10M) ON tablespace name TO user-id1, user-id2,….

Thus, the user gets a quota of 10 megabytes on the said tablespace. Quota may be specified as integer value followed by K or M denoting kilobyte or megabyte respectively. This way, user may be granted certain amount of space in one or more tablespace.

How a user will put an object to a tablespace? One way is to specify the tablespace at the time of object creation. For example,

CREATE TABLE tablename (column1 decription, column2 description,….., table constraints) TABLESPACE tablespace name

Suppose, TABLESPACE clause is not specified (that’s what we did in lab so far) then it goes to the default tablespace. Default tablespace for the user can be specified by DBA as follows.

CREATE USER userid IDENTIFIED BY password DEFAULT TABLESPACE tablespace name

Or

ALTER USER userid DEFAULT TABLESPACE tablespace name

If default tablespace is not specified by DBA, then Oracle assumes default tablespace is SYSTEM. This tablespace in Oracle is a special one which is automatically created at the time of installation and it contains the data dictionary (remember, a set of tables keeping the details of objects, user, constraints, accessibility information and consulted every time for any query. These are maintained by DBMS itself). Hence, it is advisable that DBA should make other tablespaces as the default one for the users.
DBA can withdraw the access permission on a tablespace from a user as follows.
REVOKE RESOURCE ON tablespace name FROM user-id1, user-id2, …
or
GRANT RESOURCE (0) ON tablespace name TO user-id1, user-id2, …
Please note that, logical partitioning in the form of TABLESPACE allows DBA to put different projects to different partitions to provide better maintenance in the form of backup etc. Related users can be placed in same partition; independent users can be isolated by putting them in different partitions. All these help in better management.

3. **Security in the form of view:** Remember one can create views based on one or multiple tables. In order to restrict the access to all the data of a table, one can create a view from the table(s) and grant suitable privileges on the view to the target users (follow the command for granting object level privileges). So, users can not access other data of the table. This way view also supplements the security.

==========

Following are not related to security. But I may miss it later. There are certain useful features of oracle that one should be aware of.

There is a utility in Oracle called sql loader that enables inserting rows in tables by reading data from a text file.

There is export import facility to transfer database from one installation to another.

Export creates a dump file of the database. One can copy this and use it in another installation through import utility. Thus, export can be used for backup purpose also. One can import data from latest dump.

One can export particular tablespace(s) or it may be user specific etc. If a project is limited to certain tablespaces, one can have a dump version of the project. Similarly, one can dump data of specific user. Instead of having a complete dump of whole database/tablspace/user-specific objects, one can export only the definitions, say only the schemas. It has a good use. Developers can take the export version of schemas and import the same into client site without creating all those huge number of objects along with constraints afresh. Unwanted errors in the process can be avoided. Whenever there is version upgradation of the software then also data and objects can be imported using export version of the old database. Thus, export-import are strong utilities.