Database Systems LAB # 11

Data Control Language Commands and Views in RDBMS

DCL:

Data Control Language Statements are used to grant privileges on tables, views, sequences, synonyms, procedures to other users or roles.

set linesize 100 pagesize 50

The DCL statements are

GRANT :Use to grant privileges to other users or roles.
REVOKE :Use to take back privileges granted to other users and roles.

Privileges are of two types :

- System Privileges
- Object privileges

System Privileges are normally granted by a DBA to users. Examples of system privileges are CREATE SESSION, CREATE TABLE, CREATE USER etc.

System Privileges are normally granted by a DBA to users. Examples of system privileges are CREATE SESSION, CREATE TABLE, CREATE USER etc.

Object privileges means privileges on objects such as tables, views, synonyms, procedure. These are granted by owner of the object.

Object Privileges are

ALTER	Change the table definition with the ALTER TABLE statement.
DELETE	Remove rows from the table with the DELETE statement. Note: You must grant the SELECT privilege on the table along with the DELETE privilege.
INDEX	Create an index on the table with the CREATE INDEX statement.

INSERT	Add new rows to the table with the INSERT statement.
REFERENCES	Create a constraint that refers to the table. You cannot grant this privilege to a role.
SELECT	Query the table with the SELECT statement.
UPDATE	Change data in the table with the UPDATE statement.
	Note: You must grant the SELECT privilege on the table along with the UPDATE privilege.

Special Queries;

1. Create user username identified by password;
2. Alter user username account unlock;
3. Alter user username identified by new password;

Grant

Grant is use to grant privileges on tables, view, procedure to other users or roles

Examples

1. Grant create session to username;
2. Grant create table to username;
3. Grant insert any table to username;
4. Grant select any table to username;
5. Grant update any table to username;
6. Grant delete any table to username;
7. Grant alter any table to username;
8. Grant drop any table to username;
9. Grant select, update, insert on emp to sami;
10. Grant all on emp to sami;
11. Grant select on yaseen.t1 to yaseen;
12. Grant select on emp to yaseen; and to access the table after grant select * from sys.emp;
13. Grant dba to Yaseen;

Suppose you want to grant select privilege on emp to all other users of the database. Then

Grant select on emp to public;

Suppose you want to grant update and insert privilege on only certain columns not on all the columns then include the column names in grant statement. For example you want to grant update privilege on ename column only and insert privilege on empno and ename columns only. Then give the following statement

Grant update (ename), insert (empno, ename) on emp to sami;

To grant select statement on emp table to sami and to make sami be able further pass on this privilege you have to give WITH GRANT OPTION clause in GRANT statement like this.

Grant select on emp to sami with grant option;

REVOKE

Use to revoke privileges already granted to other users.

For example to revoke select, update, insert privilege you have granted to Sami then give the following statement.

revoke select, update, insert on emp from sami;

To revoke select statement on emp granted to public give the following command.

revoke select on emp from public;

To revoke update privilege on ename column and insert privilege on empno and ename columns give the following revoke statement.

revoke update, insert on emp from sami;

Note: You cannot take back column level privileges. Suppose you just want to take back insert privilege on ename column then you have to first take back the whole insert privilege and then grant privilege on empno column.

ROLES

A role is a group of Privileges. A role is very handy in managing privileges, Particularly in such situation when number of users should have the same set of privileges.

For example you have four users :Sami, Scott, Ashi, Tanya in the database. To these users you want to grant select ,update privilege on emp table, select,delete privilege on dept table. To do this first create a role by giving the following statement.

create role clerks

Then grant privileges to this role.

grant select,update on emp to clerks;
grant select,delete on dept to clerks;

Now grant this clerks role to users like this

grant clerks to sami, scott, ashi, tanya ;

Now Sami, Scott, Ashi and Tanya have all the privileges granted on clerks role.

Suppose after one month you want grant delete on privilege on emp table all these users then just grant this privilege to clerks role and automatically all the users will have the privilege.

grant delete on emp to clerks;

If you want to take back update privilege on emp table from these users just take it back from clerks role.

revoke update on emp from clerks;

To Drop a role

Drop role clerks;

LISTING INFORMATION ABOUT PRIVILEGES

To see which table privileges are granted by you to other users.

SELECT * FROM USER_TAB_PRIVS_MADE

To see which table privileges are granted to you by other users

SELECT * FROM USER_TAB_PRIVS_RECD;

To see which column level privileges are granted by you to other users.

SELECT * FROM USER_COL_PRIVS_MADE

To see which column level privileges are granted to you by other users

SELECT * FROM USER_COL_PRIVS_RECD;

To see which privileges are granted to roles

SELECT * FROM USER_ROLE_PRIVS;

Views in RDBMS:

A view is nothing more than a SQL statement that is stored in the database with an associated name. A view is actually a composition of a table in the form of a predefined SQL query.

A view can contain all rows of a table or select rows from a table. A view can be created from one or many tables which depends on the written SQL query to create a view.

Views, which are kind of virtual tables, allow users to do the following:

- Structure data in a way that users or classes of users find natural or intuitive.
- Restrict access to the data such that a user can see and (sometimes) modify exactly what they need and no more.
- Summarize data from various tables which can be used to generate reports.

Creating Views:

Database views are created using the **CREATE VIEW** statement. Views can be created from a single table, multiple tables, or another view.

To create a view, a user must have the appropriate system privilege according to the specific implementation.

The basic CREATE VIEW syntax is as follows:

```
CREATE VIEW view_name AS
SELECT column1, column2.....
FROM table_name
WHERE [condition];
```

You can include multiple tables in your SELECT statement in very similar way as you use them in normal SQL SELECT query.

Example:

Consider the CUSTOMERS table having the following records:

```
+----+----------+-----+-----------+----------+
| ID | NAME     | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|  1 | Ramesh   |  32 | Ahmedabad |  2000.00 |
|  2 | Khilan   |  25 | Delhi     |  1500.00 |
|  3 | kaushik  |  23 | Kota      |  2000.00 |
|  4 | Chaitali |  25 | Mumbai    |  6500.00 |
|  5 | Hardik   |  27 | Bhopal    |  8500.00 |
|  6 | Komal    |  22 | MP        |  4500.00 |
|  7 | Muffy    |  24 | Indore    | 10000.00 |
+----+----------+-----+-----------+----------+
```

Now, following is the example to create a view from CUSTOMERS table. This view would be used to have customer name and age from CUSTOMERS table:

```
SQL > CREATE VIEW CUSTOMERS_VIEW AS
SELECT name, age
FROM  CUSTOMERS;
```

Now, you can query CUSTOMERS_VIEW in similar way as you query an actual table. Following is the example:

```
SQL > SELECT * FROM CUSTOMERS_VIEW;
```

This would produce the following result:

```
+----------+-----+
| name     | age |
+----------+-----+
| Ramesh   |  32 |
| Khilan   |  25 |
| kaushik  |  23 |
| Chaitali |  25 |
```

```
| Hardik   | 27 |
| Komal    | 22 |
| Muffy    | 24 |
+----------+-----+
```

The WITH CHECK OPTION:

The WITH CHECK OPTION is a CREATE VIEW statement option. The purpose of the WITH CHECK OPTION is to ensure that all UPDATE and INSERTs satisfy the condition(s) in the view definition.

If they do not satisfy the condition(s), the UPDATE or INSERT returns an error.

The following is an example of creating same view CUSTOMERS_VIEW with the WITH CHECK OPTION:

```
CREATE VIEW CUSTOMERS_VIEW AS
SELECT name, age
FROM  CUSTOMERS
WHERE age IS NOT NULL
WITH CHECK OPTION;
```

Updating Views:

```
UPDATE CUSTOMERS_VIEW
      SET AGE = 35
      WHERE name='Ramesh';
```

Inserting Rows into a View:

Same as table;

Deleting Rows into a View:

```
DELETE FROM CUSTOMERS_VIEW
```

```
    WHERE age = 22;
```

Dropping Views:

Obviously, where you have a view, you need a way to drop the view if it is no longer needed. The syntax is very simple as given below:

```
DROP VIEW view_name;
```