Database Security

- **Database Security** means keeping sensitive information safe and prevent the loss of data. Security of data base is controlled by Database Administrator (DBA).

- Database security is the technique that protects and secures the database against intentional or accidental threats.
- Security concerns will be relevant not only to the data resides in an organization's database: the breaking of security may harm other parts of the system, which may ultimately affect the database structure.
- Consequently, database security includes hardware parts, software parts, human resources, and data.
- To efficiently do the uses of security needs appropriate controls, which are distinct in a specific mission and purpose for the system.
- The requirement for getting proper security while often having been neglected or overlooked in the past days; is now more and more thoroughly checked by the different organizations.

- Data centres, as well as your own database, are **susceptible to insider threats and physical attacks by outsiders**. Even physical hardware such as laptops and other mobile devices need to be kept secure. If unauthorised users gain access to your physical database server, they can corrupt, steal, or leak your data.

- We consider database security about the following situations:
- Theft and fraudulent.
- Loss of confidentiality or secrecy.
- Loss of data privacy.
- Loss of data integrity.
- Loss of availability of data.

- A **privilege** is a right to execute a particular type of SQL statement or to access another user's object. Some examples of privileges include the right to:
- Connect to the database (create a session)
- Create a table
- Select rows from another user's table
- Execute another user's stored procedure

- We can give privilege to the

a. User

b. Role

SQL Authorization

- When the SQL standard authorization mode is enabled, object owners can use the GRANT and REVOKE SQL statements to set the user privileges for specific database objects or for specific SQL actions. They can also use roles to administer privileges.

- The GRANT statement is used to grant specific privileges to users or to roles, or to grant roles to users or to roles. The REVOKE statement is used to revoke privileges and role grants. The grant and revoke privileges are:
- DELETE
- EXECUTE
- INSERT
- SELECT
- REFERENCES
- TRIGGER
- UPDATE

Granting of privileges:

i. A system privilege is the right to perform a particular action,or to perform an action on any schema objects of a particular type.
ii. An authorized user may pass on this authorization to other users.This process is called as ganting of privileges.
iii. Syntax:

GRANT <privilege list> ON<relation name or view name> TO<user/role list>

iv. Example:
The following grant statement grants user U1,U2 and U3 the select privilege on Emp_Salary relation:

GRANT select ON Emp_Salary TO U1,U2 and U3.

Revoking of privileges:

i. We can reject the privileges given to particular user with help of revoke statement.
ii. To revoke an authorization, we use the revoke statement.
iii. Syntax:
REVOKE <privilege list> ON<relation name or view name> FROM <user/role list>[restrict/cascade]
iv. Example:
The revocation of privileges from user or role may cause other user or roles also have to loose that privileges.This behavior is called cascading of the revoke.
Revoke select ON Emp_Salary FROM U1,U2,U3.

- Write a sql command to create role “Manager” who can update , delete and view records of the table name “Employee”. Assign user Ram and Shyam to role Manager.

ROLES

- When there are many users in a database it becomes difficult to grant or revoke privileges to each users.
- A role is a collection of privileges that can be granted to a group of users or other roles.
- Roles help you grant and manage sets of privileges for various categories of users, rather than grant those privileges to each user individually.
- Any authorization that can be granted to a user can be granted to a role.
- Examples of roles could include instructor, teaching_assistant, student, dean, and department_chair.

Role

- A role is created to ease setup and maintenance of the security model. It is a named group of related privileges that can be granted to the user. When there are many users in a database it becomes difficult to grant or revoke privileges to users. Therefore, if you define roles:
- You can grant or revoke privileges to users, thereby automatically granting or revoking privileges.
- You can either create Roles or use the system roles pre-defined.

System Roles	Privileges granted to the Role
Connect	Create table, Create view, Create synonym, Create sequence, Create session etc.
Resource	Create Procedure, Create Sequence, Create Table, Create Trigger etc. The primary usage of the Resource role is to restrict access to database objects.
DBA	All system privileges

ROLES

- Roles can be created in SQL as

```
CREATE ROLE instructor;
```

- Roles can then be granted privileges just as the users can :

```
GRANT SELECT ON department TO instructor;
```

- Roles can be granted to user, as well as to other roles:

```
CREATE ROLE dean;
GRANT instructor TO dean;
GRANT dean TO Asha;
```

Creating and Assigning a Role –

First, the (Database Administrator)DBA must create the role. Then the DBA can assign privileges to the role and users to the role.

Syntax –

```
CREATE ROLE manager; Role created.
```

In the syntax:

'manager' is the name of the role to be created.

- Now that the role is created, the DBA can use the GRANT statement to assign users to the role as well as assign privileges to the role.
- It's easier to GRANT or REVOKE privileges to the users through a role rather than assigning a privilege directly to every user.
- If a role is identified by a password, then GRANT or REVOKE privileges have to be identified by the password.

Grant privileges to a role –

```
GRANT create table, create view TO manager; Grant succeeded.
```

Grant a role to users

```
GRANT manager TO SAM, STARK; Grant succeeded.
```

Revoke privilege from a Role :

```
REVOKE create table FROM manager;
```

Drop a Role :

```
DROP ROLE manager;
```

- **Explanation –**
  Firstly it creates a manager role and then allows managers to create tables and views. It then grants Sam and Stark the role of managers. Now Sam and Stark can create tables and views. If users have multiple roles granted to them, they receive all of the privileges associated with all of the roles. Then create table privilege is removed from role ‘manager’ using Revoke.The role is dropped from the database using drop.

How database can be made secure :

1. New security patches must be updated .
2. Add privilege to the database relation
3. Whitelisted users only allowed to use software
4. Regular update of operating system
5. Regular audit of database security
6. Using latest antivirus and firewalls.

Why database security is needed ?

- 1. Critical information protection
- 2. To prevent data misuse
- 3. to protect data integrity
- 4. to keep company reputation
- 5. to prevent unauthorized access
- 6. to maintain database a viability