SVJ

(b) **Make Use of 'AFTER' trigger :** If use a trigger is must for you, then it is best to avoid the mutating table error by using 'after' trigger, to avoid the currency issues associated with a mutating table. For example, using a trigger "after update on salary", the original update has completed and the table will not be mutating.

(c) **Re-work the trigger syntax :** We can avoid mutating tables with a combination of row-level and statement-level triggers.

(d) **Use autonomous transactions :** You can avoid the mutating table error by marking your trigger as an autonomous transaction, making it independent from the table that calls the procedure.

2.9 Security :

MU - Dec. 2010

1. Introduction

A DBMS system always has a separate system for security which is responsible for protecting database against accidental or intentional loss, destruction or misuse.

2. Objectives CIA

(a) **Confidentiality :**

- Data in database should be given to only authorized users.
- For e.g. in HR department employee's personnel data should be accessible to that particular employee and the HR person only.

(b) **Integrity :**

- Only authorized users should be allowed to modify data.
- For e.g. only account department can change financial details of company.

(c) **Availability :**

- Authorized users can be able to accesses data any time he wants.
- For e.g. Employee should be able to accesses own salary any time.

3. Security Levels

(a) **Database level :** DBMS system should ensure that the authorization restriction needs to be there on users.

(b) **Operating system Level :** Operating system should not allow unauthorized users to enter in system.

(c) **Network Level :** Database is at some remote place and it is accessed by users through the network so security is required.

4. Security Mechanisms

(a) Access Control (Authorization) :

- Which identifies valid users who may have any access to the valid data in the Database and which may restrict the operations that the user may perform e.g. ROLE function in SQL
- **For Example** The movie database might designate two roles : "users" (query the data only) and "designers" (add new data) user must be assigned to a role to have the access privileges given to that role.
- Access privileges are assigned to users and roles.
- Each application is associated with a specified role. Each role has a list of authorized users who may execute / Design /administers the application.

(b) Authenticate the User :

- Which identify valid users who may have any access to the data in the Database
- Restrict each user's view of the data in the Database
- This may be done with help of concept of Views in Relational databases.

(c) Cryptographic control / Data Encryption :

- Encode data in a cryptic form (Coded) so that although data is captured by unintentional user still he can't be able to decode the data.
- Used for sensitive data, usually when transmitted over communication links but also may be used to prevent by passing the system to gain access to the data.

(d) Inference Control :

- Ensure that confidential information can't be retrieved even by deduction.
- Prevent disclosure of data through statistical summaries of confidential data.

(e) Flow control or Physical Protection :

- Prevents the copying of information by unauthorized person.
- Computer systems must be physically secured against any unauthorized entry.

(f) Virus control :

- At user level authorization should be done to avoid intruder attack through humans.
- There should be mechanism for providing protection against data virus.

(g) **User defined control :**

- Define additional constraints or limitations on the use of database.
- These allow developers or programmers to incorporate their own security procedures in addition to above security mechanism.

2.10 Authorization in SQL (Access Control) :

MU - Dec. 2010

Introduction :

- Authorization is finding out if the person, once identified, is permitted to have the resource.
- Authorization explains that what you can do and is handled through the DBMS unless external security procedures are available.
- This is usually determined by finding out if that person is a part of a particular group, if that person has paid admission, or has a particular level of security clearance.
- Authorization is equivalent to checking the guest list at an exclusive party, or checking for your ticket when you go to the opera.
- Database management system allows DBA to give different access rights to the users as per their requirements.
- In SQL Authorization can be done by using Read, Insert, Update or Delete privileges

Basic Authorizations we can use any one form or combination of the following basic forms of authorizations :

(a) **Resource authorization :** Authorization to access any system resource. e.g. sharing of database, printer etc.

(b) **Alternation Authorization :** Authorization to add attributes or delete attributes from relations

(c) **Drop Authorization :** Authorization to drop a relation

Database administrator :

- The main authority of database system is database administrator (DBA).
- The SQL standard specifies modification to the schema can be done only by the database owner of schema or DBA of schema.
- The DBA may authorize new users; restructure the database and so on.
- It is analogous to that of **superuser** or operator for operating systems.

4. **Granting of privileges :**

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

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

- **Example :**
  - o Consider an example for granting update authorization to the Emp_Salary relation of the company database. Assume that, initially that the DBA grants update authorization on Emp_Salary to other users U1, U2 and U3, who may in turn pass on this authorization to other users.
  - o This passing of authorization from one user to other users is called **authorization graph**. (Note: The nodes of graph are the users.)
- 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
```

- Following **grant** statement gives all users all authorization on the amount attributes of the Emp_Salary relation using public keyword;

```
GRANT ALL
ON Emp_Salary
TO PUBLIC
```

- To view privileges given to table

```
SELECT GRANTEE, OWNER, GRANTOR, PRIVILEGE, GRANTABLE
FROM DBA_TAB_PRIVS
WHERE TABLE_NAME = 'Emp_Salary';
```

- **Some other types of Privileges :**

(a) Reference privileges :

- o SQL permits a user to declare foreign keys while creating relations.

- **Example** Allow user U1 to create relation that references key 'Eid' of Emp_Salary relation.

```
GRANT REFERENCES (Eid)
ON Emp_Salary
TO U1
```

(b) **Execute privileges :**

- This privilege authorizes a user to execute a function or procedure.
- Thus, only user who has execute privilege on a function **Create_Acc()** can call function.

```
GRANT EXECUTE
ON Create_Acc
TO U1
```

5. **Role :**

- In real time database applications having multiple users with same access rights, needs to have same grant privileges. For Example there are many users who are of type managers and having same access rights on system in a company.
- Instead of giving grant to each and every Manager user we can create a generalized set of grant privileges with name manager and this can be assigned to all employees belongs to category manager. Such concept is called as Role
- A set of role can be created in databases and assign to any new user entered in database system as per authorization required.
- **Syntax :**

```
CREATE ROLE <Role_name>
```

- **Example :**

We can create role manager as follows

```
CREATE ROLE manger
CREATE ROLE Tech_manager
```

Role can be granted to users as follows

```
GRANT manger
TO mahesh
GRANT manger
TO Tech_manger
```

- It is possible to have chain of roles like role manager is assigned to Tech_manager as above.

6. **Revoking of privileges :**

- We can reject the privileges given to particular user with help of revoke statement.
- To revoke an authorization, we use the **revoke** statement.
- **Syntax :**

```
REVOKE <privilege list>
ON <relation name or view name>
From <user/role list> [restrict/ cascade]
```

- **Example :**
  - o The revocation of privileges from user or role may cause other user or roles also have to loose that privilege. This behavior is called cascading of the revoke.

```
Revoke select
ON Emp_Salary
FROM U1, U2, U3

Revoke update (amount)
ON Emp_Salary
FROM U1, U2, U3

Revoke references (amount)
ON Emp_Salary
FROM U1
```

  - o The revoke statements may alternatively specify **restrict** if we don't wa cascade behavior

```
REVOKE select
ON Emp_Salary
FROM U1, U2, U3 restrict
```

  - o The following **revoke** statement revokes only the grant option, rather tha the actual select privileges

```
REVOKE grant option select
ON Emp_Salary
FROM U1
```

2.11 Authentication in DBMS :

1. Introduction :

- Authentication is any process by which you verify that someone is who they claim they are.
- Authentication basically checks who you are and is typically handled via the operating system and not the DBMS.
- This usually involves a username and a password, Knowledge of the password is assumed to guarantee that the user is authentic.
- The weakness in this system for transactions that are significant (such as the exchange of money) is that passwords can often be stolen, accidentally revealed, or forgotten.
- So we can also use other method of demonstrating identity, such as a smart card, retina scan, voice recognition or fingerprints to protect identity.
- Authentication is equivalent to showing your drivers license at the ticket counter at the airport.
- Internet business and many other transactions require a more stringent authentication process. The use of digital certificates issued and verified by a Certificate Authority (CA) as part of a public key infrastructure is considered likely to become the standard way to perform authentication on the Internet.
- Authentication is distinct from authorization, which is the process of giving individuals access to system objects based on their identity.
- Logically, authentication precedes authorization (although they may often seem to be combined).

Review Questions

Q. 1 What do you mean by Assertion and when are they used in database.

Q. 2 Explain working of triggers with help of example. In which conditions Triggers can not be used in database.

Q. 3 Explain various security mechanisms in databases.

Q. 4 Explain all types of integrity with help of examples.

Q. 5 Write in brief about Referential Integrity problems.

Q. 6 Write short note about Delete update rules in Referential integrity.

Q. 7 Explain the term 'Referential Integrity' and its relation with foreign key.