CHAPTER 4:

DATABASE SECURITY

LECTURE OUTLINE

- Security Requirements
- Threats and countermeasures
- Integrity of Database
- Access control mechanisms
- SQL’s grant and revoke
- Role of views

Security Requirements

Physical database integrity. The data of a database are immune to physical problems, such as power failures, and someone can reconstruct the database if it is destroyed through a catastrophe.

Logical database integrity. The structure of the database is preserved. With logical integrity of a database, a modification to the value of one field does not affect other fields.

Element integrity. The data contained in each element are accurate.

Auditability. It is possible to track who or what has accessed (or modified) the elements in the database.

Access control. A user is allowed to access only authorized data, and different users can be restricted to different modes of access (such as read or write).

User authentication. Every user is positively identified, both for the audit trail and for permission to access certain data.

Availability. Users can access the database in general and all the data for which they are authorized.

THREATS

What are the threats?

Loss of integrity

- Changing data values for reasons of sabotage
- Data must not be altered in transit, and steps must be taken to ensure that unauthorized people cannot alter data (for example, in a breach of confidentiality).
- E.g. Student changing grades for a class they're taking

Loss of confidentiality

- Data or system cannot be accessed.

Loss of availability

- "Denial of service"
- The ability for authorized parties to access information on a consistent and timely basis is referred to as "availability."

- ❑ Who's trying to mess with us?
  - ❑ Outsiders
  - ❑ Corporate competitors
  - ❑ Organized crime
  - ❑ Government "cyber-warriors"
  - ❑ Terrorists / activists
- ❑ Insiders
  - ❑ Disgruntled, bribed, or naïve employees
- ❑ Accidental mis-use

HOW TO SECURE THE DB?

Prevent it

Deter it

Deflect it

Detect it

Recover from it

LEGISLATIVE ASPECTS OF DB SECURITY

Legal and ethical compliance / Business rules

- Requirements to maintain accurate information
- Requirements to disclose information to appropriate people
- Requirements to *not* disclose information to *in*appropriate people

Where will security be enforced?

- by the physical environment?
  - by locked doors? by armed guards?
- by the hardware?
- by the software?
  - by the OS? by the DBMS? by applications programs?
  - DBMS includes **security subsystem**

LEGISLATIVE ASPECTS OF DB SECURITY

Levels of security

- Access / no access
- Partial access
  - Limited authorizations
  - Authorizations based on user role, time of day, location, etc.
- Emergency access

COUNTERMEASURES

Database and web servers should be kept apart

- Store the DB secure.
- Locked environment with authorized access.

Use firewalls for web applications and databases

- Denies access to traffic and protect the DB server from threats.

Limit the number of people to access

- Strong passwords
- Encrypted password should be stored.
- After three or four failed logins, accounts should be locked.
- Protocol should be established for staffs leaving the job.
- User access to the database is secured.

Update the operating system and patches on a regular basis

- Important when dealing with databases that are linked to a large number of third-party applications, each of which requires its own set of patches.

Database activity should be audited and monitored on a regular basis

Validate the safety of your database

Data and backups must be encrypted

Integrity of the Database

Two situations can affect the integrity of a database

- When the whole database is damaged (as happens, for example, if its storage medium is damaged)
- When individual data items are unreadable

Integrity of the database as a whole is the responsibility of:

- The DBMS
- The operating system
- (human) computing system manager

Element Integrity

The integrity of database elements is their correctness or accuracy.

This corrective action can be taken in three ways:

- The DBMS can apply **field checks,** activities that test for appropriate values in a position.
- Access control
- change log for the database. it lists every change made to the database; it contains both original and modified values.

Auditability

For some applications it may be desirable to generate an audit record of all access (read or write) to a database. Such a record can help to maintain the database's integrity, or at least to discover after the fact who had affected which values and when.

That users can access protected data incrementally; that is, no single access reveals protected data, but a set of sequential accesses viewed together reveals the data, much like discovering the clues in a detective novel.

ACCESS CONTROL MECHANISMS

What's that access control?

- Individuals are **authenticated** and **authorized to access** the information they are permitted **to see and use through access controls**.

Discretionary Access Control (DAC)

- **Providing access to a specific piece of data to a specific user** in a specific way.
- For example, “allow John to enter employees data into the Employee table.”

Mandatory Access Control (MAC)

- This is a security model in which a **central authority manages access rights based on multiple levels of security.**
- Used in Military applications.

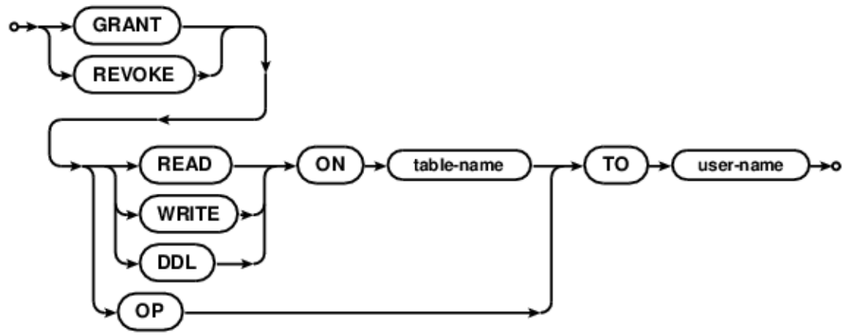
Role Based Access Control (RBAC)

- **Users assigned roles**.
- Roles entitled to **specific permissions on specific data**
- E.g., “emergency physician can update any patient record”

SQL’S GRANT AND REVOKE

What's that access control?

- DCL commands are used to enforce database security in a multiple user database environment.
- Two types of DCL commands are **GRANT** and **REVOKE**.
- Only **Database Administrators** or **database object owners** have the ability to **grant or revoke privileges on a database object**.



SQL'S GRANT

- The SQL Grant command is used to **grant permissions to database objects to a user**.
- Users can also grant permissions to other users using this command.

```
grant privilege_name on object_name to {user_name | public | role_name}
```

- Here, **privilege name** is the **permission to be granted**, **object name** is the **database object's name**, **user name** is the user to **whom access should be granted**, and **public is used to allow access to all users**.

REVOKE

- If any user privileges on database objects have been granted, the **revoke command will remove them**.
- It **performs operations in the opposite direction of the Grant command**.

```
revoke privilege_name on object_name from {user_name | public | role_name}
```