ADVANCED DATABASE ACCESS PROTOCOLS

Module II

DATABASE SECURITY

- Securing databases against a variety of threats.
- It also presents schemes of providing access privileges to authorized users.
- **Types of Security:**
- Various legal and ethical issues regarding the right to access certain information *e.g. some information may be deemed to be private and cannot be accessed legally by unauthorized organizations or persons.*
- Policy issues at the governmental, institutional, or corporate level as to what kinds of information should not be made publicly available *e.g. credit ratings and personal medical records.*
- System-related issues such as the *system levels* at which various security functions should be enforced e.*g. whether a security function should be handled at the physical hardware level, the operating system level, or the DBMS level.*
- The need in some organizations to identify multiple *security levels* and to categorize the data and users based on these classifications *e.g.top secret, secret, confidential, and unclassified*

THREATS TO DATABASES.

- Threats to databases can result in the loss or degradation of some or all of the following commonly accepted security goals: integrity, availability, and confidentiality.
- **Loss of integrity:**
- Database integrity refers to the requirement that information be protected from improper modification.
- Modification of data includes creation, insertion, updating, changing the status of data, and deletion.
- Integrity is lost if unauthorized changes are made to the data by either intentional or accidental acts.
- If the loss of system or data integrity is not corrected, continued use of the contaminated system or corrupted data could result in inaccuracy, fraud, or erroneous decisions.

- **Loss of availability:**
- Database availability refers to making objects available to a human user or a program to which they have a legitimate right.

- **Loss of confidentiality:**
- Database confidentiality refers to the protection of data from unauthorized disclosure.

- The impact of unauthorized disclosure of confidential information can range from violation of the Data Privacy Act to the jeopardization of national security.

- Unauthorized, unanticipated, or unintentional disclosure could result in loss of public confidence, embarrassment, or legal action against the organization.

DATABASE SECURITY MECHANISMS

- Responsible for ensuring the security of portions of a database against unauthorized access.
- It is now customary to refer to two types of database security mechanisms:

- **Discretionary Security Mechanisms**

- **Mandatory Security Mechanisms**

DISCRETIONARY ACCESS CONTROL BASED ON GRANTING AND REVOKING PRIVILEGES

- To grant privileges to users, including the capability to access specific data files, records, or fields in a specified mode (such as read, insert, delete, or update).

- The typical method of enforcing **discretionary access control** in a database system is based on the granting and revoking of **privileges**

- Main idea is to include statements in the query language that allow the DBA and selected users to grant and revoke privileges.

TYPES OF DISCRETIONARY PRIVILEGES

- **The account level:**
- DBA specifies the particular privileges that each account holds independently of the relations in the database. e.g. CREATE SCHEMA or CREATE TABLE, ALTER, DROP, SELECT privilege

- **The relation (or table) level:**
- DBA can control the privilege to access each individual relation or view in the database.
- whether they are base relations or virtual (view) relations.
- Privileges at the relation level specify for each user the individual relations on which each type of command can be applied

- The owner account holder can pass privileges on any of the owned relations to other users by **granting** privileges to their accounts.
- *In SQL the following types of privileges can be granted on each individual relation R:*

- **SELECT (retrieval or read) privilege on *R:*** Gives the account retrieval privilege.
- In SQL this gives the account the privilege to use the SELECT statement to retrieve tuples from *R*.
- **Modification privileges on *R:*** Gives the account the capability to modify the tuples of *R*. In SQL this includes three privileges: UPDATE, DELETE, and INSERT.
- These correspond to the three SQL commands for modifying a table *R*. Additionally, both the INSERT and UPDATE privileges can specify that only certain attributes of *R* can be modified by the account.
- ***References privilege on R:*** Gives the account the capability to *reference* (or refer to) a relation *R* when specifying integrity constraints.
- This privilege can also be restricted to specific attributes of *R*.

SPECIFYING PRIVILEGES THROUGH THE USE OF VIEWS

- The mechanism of **views** is an important *discretionary authorization mechanism* in its own right.

- e.g., if the owner *A* of a relation *R* wants another account *B* to be able to retrieve only some fields of *R,* then *A* can create a view *V* of *R* that includes only those attributes and then grant SELECT on *V* to *B*.

- The same applies to limiting *B* to retrieving only certain tuples of *R;* a view *V* can be created by defining the view by means of a query that selects only those tuples from *R* that *A* wants to allow *B* to access

REVOKING OF PRIVILEGES

- In some cases it is desirable to grant a privilege to a user temporarily.

- e.g. the owner of a relation may want to grant the SELECT privilege to a user for a specific task and then revoke that privilege once the task is completed. Hence, a mechanism for **revoking** privileges is needed.

- In SQL a REVOKE command is included for the purpose of canceling privileges.

PROPAGATION OF PRIVILEGES USING THE GRANT OPTION

- Whenever the owner *A* of a relation *R* grants a privilege on *R* to another account *B*, the privilege can be given to *B* *with* or *without* the **GRANT OPTION**.

- If the GRANT OPTION is given, *B* can also grant that privilege on *R* to other accounts. Suppose that *B* is given the GRANT OPTION by *A* and that *B* then grants the privilege on *R* to a third account *C*, also with the GRANT OPTION.

- In this way, privileges on *R* can **propagate** to other accounts without the knowledge of the owner of *R*.

- If the owner account *A* now revokes the privilege granted to *B*, all the privileges that *B* propagated based on that privilege *should automatically be revoked* by the system.

AN EXAMPLE TO ILLUSTRATE GRANTING AND REVOKING OF PRIVILEGES

- **GRANT** CREATETAB **TO** A1;
- **CREATE SCHEMA** EXAMPLE **AUTHORIZATION** A1;
- **GRANT** INSERT, DELETE **ON** EMPLOYEE, DEPARTMENT **TO** A2;
- **GRANT** SELECT **ON** EMPLOYEE, DEPARTMENT **TO** A3 **WITH GRANT OPTION**;
- **GRANT** SELECT **ON** EMPLOYEE **TO** A4;
- **REVOKE** SELECT **ON** EMPLOYEE **FROM** A3;

- The limitation is to retrieve only the Name, Bdate, and Address attributes and only for the tuples with Dno = 5. A1 then can create the following view:

```
CREATE VIEW A3EMPLOYEE AS
SELECT Name, Bdate, Address
FROM EMPLOYEE
WHERE Dno = 5;
```

- After the view is created, A1 can grant SELECT on the view A3EMPLOYEE to A3 as follows:
- **GRANT** SELECT **ON** A3EMPLOYEE **TO** A3 **WITH GRANT OPTION**;

- Finally, suppose that A1 wants to allow A4 to update only the Salary attribute of EMPLOYEE; A1 can then issue the following command:
- **GRANT** UPDATE **ON** EMPLOYEE (Salary) **TO** A4;

MANDATORY ACCESS CONTROL (MAC)

- Allocation of access privileges depends on the hierarchy of employees and personnel in organization.
- When a user tries to access a resource, the system automatically checks whether or not they are allowed access and their assigned category.
- Users must fulfil both security and category in order to access data.
- Organization must first put in time and effort to understand the information flow properly and map it out.

- Typical **security classes** are top secret (TS), secret (S), confidential (C), and unclassified (U), where TS is the highest level and U the lowest.

- Other more complex security classification schemes exist, in which the security classes are organized in a lattice.

- The system with four security classification levels

- where TS ≥ S ≥ C ≥ U, to illustrate our discussion

- like a flow chart of information and the person in middle will have access to ground level data but not any level further up.

TYPES OF MANDATORY ACCESS CONTROL

- **Multilevel security systems:** This system consists of a vertical structure of security levels, making it a simple form.

- Users can only access the information up to their security level clearance, so in the same or lower levels of the vertical structure.

- **Multilateral security systems:** As in the name, multilateral means not only vertical but also horizontal security systems.

- This is more complex as the assignment of security clearance is based on segments.

ROLE-BASED ACCESS CONTROL (RBAC)

- It also known as role-based security
- Access control method that creates permissions to end-users depends on their role within the organization.
- The notion of securely managing access by creating and restricting user access depends on clearly established roles.
- Organizations rely on RBAC to put solid, pre-defined, and pre-approved access policies in place that recognize which access privileges each user required and which access to grant or delete.
- These access privileges can be cross-system, cross platform, or cross-software, and they can continue on premise, in the cloud, or both.
- Roles can be created using the CREATE ROLE and DESTROY ROLE commands.
- RBAC can be used with traditional discretionary and mandatory access controls

- The RBAC methodology is based on a group of three primary rules that govern access to secured systems

- **Role Assignment:** Each transaction or operation can only be carried out if the user has assumed the suitable role. An operation is represented as some action taken with respect to a system or network object that is secured by RBAC. Roles can be assigned by an independent party or selected by the user attempting to implement the action.

- **Role Authorization:** The objective of role authorization is to provide that users can only consider a role for which they have been given the suitable authorization. When a user consider a role, they should do so with authorization from an administrator.

- **Transaction Authorization:** An operation can only be done if the user trying to complete the transaction possesses the suitable role.

REMOTE DATABASE ACCESS (RDA) PROTOCOL

- Remote Database Access provides standard protocols for establishing a remote connection between a database client and a database server.

- The client is acting on behalf of an application program while the server is interfacing to a process that controls data transfers to and from a database.

- The goal is to promote the interconnection of database applications among heterogeneous environments.

An RDA client is an application-process, within an open system, that requests database services from another application-process called a database server.

A database server is an application-process, within the same or another open system, that supplies database storage facilities and provides, through OSI communication, database services to RDA clients.
An RDA client and a database server communicate by means of the RDA Service, supported by an RDA service- provider.

The part of the database server that uses the RDA service-provider to communicate with an RDA client is called an RDA server.
The RDA client has the ability to initiate RDA service requests, while the RDA server can only issue RDA service responses to reply to such requests.

An RDA operation models a request by an RDA client that is transferred to an RDA server for processing.

RDA operations enable an RDA client to request any of five types of RDA services:

a) RDA Dialogue Management services, to start and end RDA dialogues;
b) RDA Transaction Management services, to start and end RDA transactions;
c) RDA Control services, to report the status or cancel existing operations;
d) Resource Handing services, to enable or disable access by RDA clients to data resources;
e) Database Language services, to access and modify data resources.

OVERVIEW OF ADVANCED DATABASE MODELS- MOBILE DATABASE

- A Mobile Database is a type of database that can be accessed by a mobile network and connected to a mobile computing device (or wireless network). A wireless connection between the client and the server.

- A remote worker may be present at the "office" in the form of a laptop, desktop, **PDA** (Personal Digital Assistant), or another Internet-accessing device.

- Mobile users will soon be able to access any data from any location at any time because of the rapid development of the mobile network, wireless media, and satellite communications.

- A remedy to some of these limitations or issues is provided by mobile databases.

MOBILE DATABASE ENVIRONMENT COMPONENTS

- For storing the corporate and providing the corporate applications, a Corporate Database Server and DBMS is used.

- For storing the mobile data and providing the mobile application, a Remote Database and server are used.

- There is always a two-way communication link present between the Mobile DBMS and Corporate DBMS.

FEATURES OF MOBILE DATABASE

- To prevent frequent transactions from being missed due to connection failure, a cache is kept.
- Mobile Databases and the main database server are physically independent.
- Mobile gadgets hosted Mobile Databases.
- Mobile Databases can communicate with other mobile clients or a centralized database server from distant locations.
- Due to unreliable or nonexistent connections, mobile users need to be able to operate without a wireless connection with the aid of a Mobile Database (disconnected)
- Information on mobile devices is analyzed and managed using a Mobile Database

MOBILE DATABASE CONSISTS OF THREE PARTIES

- **Fixed Hosts:**
  With the aid of database servers, it handles transactions and manages data.

- **Mobile Units:**
  These are mobile, transportable computers, and the cell tower they utilize to connect to base stations is a part of that geographical area.

- **Base Stations:**
  These two-way radios, which are installed in fixed places, allow communication between the stationary hosts and the mobile units.

TEMPORAL DATABASES

- Database that needs some aspect of time for the organization of information.
- Each tuple in relation is associated with time. It stores information about the states of the real world and time.
- Temporal database does store information about past states it only stores information about current states.
- Whenever the state of the database changes, the information in the database gets updated
- In many fields, it is very necessary to store information about past states. *e.g., a stock database must store information about past stock prizes for analysis. Historical information can be stored manually in the schema.*

Terminologies in the temporal database:

- **Valid Time:** The valid time is a time in which the facts are true with respect to the real world.

- **Transaction Time:** The transaction time of the database is the time at which the fact is currently present in the database.

- **Decision Time:** Decision time in the temporal database is the time at which the decision is made about the fact.

APPLICATIONS OF TEMPORAL DATABASES

- **Finance:** It is used to maintain the **stock price** histories.
- It can be used in **Factory Monitoring System** for storing information about current and past readings of sensors in the factory.
- **Healthcare:** The histories of the patient need to be maintained for giving the right treatment.
- **Banking:** For maintaining the credit histories of the user.

TEMPORAL RELATION

- A temporal relation is defined as a *relation in which each tuple in a table of the database is associated with time, the time can be either transaction time or valid time.*

Types of Temporal Relation

1. Uni-Temporal Relation: The relation which is associated with valid or transaction time is called Uni-Temporal relation. It is related to only one time.

2. Bi-Temporal Relation: The relation which is associated with both valid time and transaction time is called a Bi-Temporal relation. Valid time has two parts namely start time and end time, similar in the case of transaction time.

3. Tri-Temporal Relation: The relation which is associated with three aspects of time namely Valid time, Transaction time, and Decision time called as Tri-Temporal relation.

FEATURES OF TEMPORAL DATABASES

- The temporal database provides built-in support for the time dimension.
- Temporal database stores data related to the time aspects.
- A temporal database contains Historical data instead of current data.
- It provides a uniform way to deal with historical data.

CHALLENGES OF TEMPORAL DATABASES

- **Data Storage:** In temporal databases, each version of the data needs to be stored **separately**.
- As a result, storing the data in temporal databases requires more storage as compared to storing data in non-temporal databases.

- **Schema Design:** The temporal database schema must accommodate the **time dimension.**
- Creating such a schema is more difficult than creating a schema for non temporal databases.
- **Query Processing:** Processing the query in temporal databases is **slower** than processing the query in non-temporal databases due to the additional complexity of managing temporal data.

SPATIAL DATABASES

- Spatial databases incorporate functionality that provides support for databases that keep track of objects in a multidimensional space.

- A general-purpose database (often a relational database) which has been improved to contain spatial information that represents objects specified in a geometric space as well as tools for searching and analyzing such data, is known as a **Spatial Database**.

- The depiction of basic geometric objects like points, lines, and polygons is supported by the majority of Spatial Databases.

- **Geographical Information Systems (GIS)**, and they are used in areas such as environmental applications, transportation systems, emergency response systems, and battle management. Other databases, such as meteorological databases for weather information, are 3D, since temperatures and other meteorological information are related to 3D spatial points.

GEODATABASE

- A Geographic Database, sometimes known as a Geodatabase.
- A Georeferenced Spatial Database that is used to store and modify geodata or information about a specific place on Earth.
- Additionally, the term "geodatabase" can refer to a collection of exclusive geographic database formats called **Geodatabase.**
- For instance, a city might connect and use datasets from common spatial databases for its wastewater department, land registry, transportation, and fire services.

CHARACTERISTICS OF SPATIAL DATABASE

- One or more spatial data types that enable the recording of spatial data as values in a table are the fundamental capability that a spatial extension to a database adds.
- Based on the vector data model, a single spatial value is often a geometric primitive (points, lines, polygon, etc.).
- Open Geospatial Consortium OGC Simple Features definition for describing geometric primitives serves as the foundation for most spatial databases data types.
- Spatial Databases must support the tracking and manipulation of coordinate systems since every geographic place must be described using a spatial reference system.
- coordinate system is included when a spatial column is defined in a table. This choice is made from a list of possible systems that are kept in a lookup table.

SEVERAL TYPES OF OPERATIONS

- **Measurement:**
  Computes geometry distance, polygon area, line length, etc.

- **Geoprocessing:**
  Create new features by changing existing ones, for as by surrounding them with a buffer or by intersecting features.

- **Geometry Constructors:**
  Specifies the vertices (points or nodes) that define the form to create new geometries.

- **Observer Functions:**
  Queries that give detailed answers on a feature, like the location of a circle's center.

- **Predicates:**
  True/false questions about the spatial relationships between geometries are permissible.

SPATIAL INDEX

- To improve database efficiency, geographical databases employ something similar to a distinct index known as a Spatial Index.

- A system must be able to obtain data from a vast collection of items without actually searching them all.

- Spatial Indexing is crucial. In addition to filtering, it ought to better allow connections between objects from various classes.

- In addition to indexes, geographical databases also provide spatial data types in their query language and data model.

SPATIAL QUERY

- A unique kind of SQL query supported by spatial databases, especially geodatabases, is known as a Spatial Query.

- The queries have a number of significant differences from non-spatial SQL queries.

- The usage of geometry data types, including points, lines, and polygons, as well as the fact that these queries take the spatial relationship between these geometries into account, are two of the most crucial features.