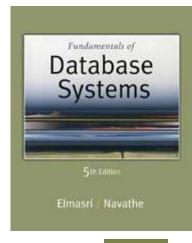


5th Edition

Elmasri / Navathe

## Chapter 23

Database Security and Authorization





#### **Chapter Outline**

- 1 Database Security and Authorization
  - 1.1 Introduction to Database Security Issues
  - 1.2 Types of Security
  - 1.3 Database Security and DBA
  - 1.4 Access Protection, User Accounts, and Database Audits
- 2 Discretionary Access Control Based on Granting Revoking Privileges
  - 2.1 Types of Discretionary Privileges
  - 2.2 Specifying Privileges Using Views
  - 2.3 Revoking Privileges
  - 2.4 Propagation of Privileges Using the GRANT OPTION
  - 2.5 Specifying Limits on Propagation of Privileges

### Chapter Outline (contd.)

- 3 Mandatory Access Control and Role-Based Access Control for Multilevel Security
  - 3.1 Comparing Discretionary Access Control and Mandatory Access Control
  - 3.2 Role-Based Access Control
  - 3.3 Access Control Policies for E-Commerce and the Web
- 4 Introduction to Statistical Database Security

## 1 Introduction to Database Security Issues

- Types of Security
  - Legal and ethical issues
  - Policy issues
  - System-related issues
  - The need to identify multiple security levels

# Introduction to Database Security Issues (2)

- Threats to databases
  - Loss of integrity
  - Loss of availability
  - Loss of confidentiality
- To protect databases against these types of threats four kinds of countermeasures can be implemented:
  - Access control
  - Inference control
  - Flow control
  - Encryption

# Introduction to Database Security Issues (3)

 A DBMS typically includes a database security and authorization subsystem that is responsible for ensuring the security portions of a database against unauthorized access.

- Two types of database security mechanisms:
  - Discretionary security mechanisms
  - Mandatory security mechanisms

#### Introduction to Database Security Issues (4)

- The security mechanism of a DBMS must include provisions for restricting access to the database as a whole
  - This function is called access control and is handled by creating user accounts and passwords to control login process by the DBMS.

# Introduction to Database Security Issues (5)

The security problem associated with databases is that of controlling the access to a statistical database, which is used to provide statistical information or summaries of values based on various criteria.

 The countermeasures to statistical database security problem is called inference control measures.

#### Introduction to Database Security Issues (6)

 Another security is that of flow control, which prevents information from flowing in such a way that it reaches unauthorized users.

Channels that are pathways for information to flow implicitly in ways that violate the security policy of an organization are called covert channels.

# Introduction to Database Security Issues (7)

- A final security issue is data encryption, which is used to protect sensitive data (such as credit card numbers) that is being transmitted via some type communication network.
- The data is encoded using some encoding algorithm.
  - An unauthorized user who access encoded data will have difficulty deciphering it, but authorized users are given decoding or decrypting algorithms (or keys) to decipher data.

#### 1.2 Database Security and the DBA

- The database administrator (DBA) is the central authority for managing a database system.
  - The DBA's responsibilities include
    - granting privileges to users who need to use the system
    - classifying users and data in accordance with the policy of the organization
- The DBA is responsible for the overall security of the database system.

#### 1.2 Database Security and the DBA (2)

- The DBA has a DBA account in the DBMS
  - Sometimes these are called a system or superuser account
  - These accounts provide powerful capabilities such as:
    - 1. Account creation
    - 2. Privilege granting
    - 3. Privilege revocation
    - 4. Security level assignment
  - Action 1 is access control, whereas 2 and 3 are discretionarym and 4 is used to control mandatory authorization

### 1.3 Access Protection, User Accounts, and Database Audits

- Whenever a person or group of person s need to access a database system, the individual or group must first apply for a user account.
  - The DBA will then create a new account id and password for the user if he/she deems there is a legitimate need to access the database
- The user must log in to the DBMS by entering account id and password whenever database access is needed.

# 1.3 Access Protection, User Accounts, and Database Audits(2)

- The database system must also keep track of all operations on the database that are applied by a certain user throughout each login session.
  - To keep a record of all updates applied to the database and of the particular user who applied each update, we can modify system log, which includes an entry for each operation applied to the database that may be required for recovery from a transaction failure or system crash.

# 1.3 Access Protection, User Accounts, and Database Audits(3)

- If any tampering with the database is suspected,
   a database audit is performed
  - A database audit consists of reviewing the log to examine all accesses and operations applied to the database during a certain time period.
- A database log that is used mainly for security purposes is sometimes called an audit trail.

# Discretionary Access Control Based on Granting and Revoking Privileges

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

### 2.1Types of Discretionary Privileges

#### ■ The account level:

 At this level, the DBA specifies the particular privileges that each account holds independently of the relations in the database.

#### The relation level (or table level):

 At this level, the DBA can control the privilege to access each individual relation or view in the database.

#### 2.1Types of Discretionary Privileges(2)

- The privileges at the account level apply to the capabilities provided to the account itself and can include
  - the CREATE SCHEMA or CREATE TABLE privilege, to create a schema or base relation;
  - the CREATE VIEW privilege;
  - the ALTER privilege, to apply schema changes such adding or removing attributes from relations;
  - the DROP privilege, to delete relations or views;
  - the MODIFY privilege, to insert, delete, or update tuples;
  - and the SELECT privilege, to retrieve information from the database by using a SELECT query.

#### 2.1Types of Discretionary Privileges(3)

- The second level of privileges applies to the relation level
  - This includes base relations and virtual (view) relations.
- The granting and revoking of privileges generally follow an authorization model for discretionary privileges known as the access matrix model where
  - The rows of a matrix M represents subjects (users, accounts, programs)
  - The columns represent objects (relations, records, columns, views, operations).
  - Each position M(i,j) in the matrix represents the types of privileges (read, write, update) that subject i holds on object j.

#### 2.1Types of Discretionary Privileges(4)

- To control the granting and revoking of relation privileges, each relation R in a database is assigned and owner account, which is typically the account that was used when the relation was created in the first place.
  - The owner of a relation is given <u>all</u> privileges on that relation.
  - In SQL2, the DBA can assign and owner to a whole schema by creating the schema and associating the appropriate authorization identifier with that schema, using the CREATE SCHEMA command.
  - The owner account holder can pass privileges on any of the owned relation to other users by granting privileges to their accounts.

#### 2.1Types of Discretionary Privileges(5)

- 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.
  - MODIFY privileges on R:
    - This gives the account the capability to modify tuples of R.
    - In SQL this privilege is further divided into **UPDATE**, **DELETE**, and **INSERT** privileges to apply the corresponding SQL command to R.
    - In addition, both the INSERT and UPDATE privileges can specify that only certain attributes can be updated by the account.

#### 2.1Types of Discretionary Privileges(6)

- In SQL the following types of privileges can be granted on each individual relation R (contd.):
  - REFERENCES privilege on R:
    - This gives the account the capability to **reference** relation R when specifying integrity constraints.
    - The privilege can also be restricted to specific attributes of R.
- Notice that to create a view, the account must have SELECT privilege on all relations involved in the view definition.

#### 2.2 Specifying Privileges Using Views

- The mechanism of views is an important discretionary authorization mechanism in its own right. For example,
  - If the owner A of a relation R wants another account B to be able to <u>retrieve only some fields</u> of R, then A can create a view V of R that includes <u>only those attributes</u> 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.

### 2.3 Revoking Privileges

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

## 2.4 Propagation of Privileges using the GRANT OPTION

- Whenever the owner A of a relation R grants a privilege on R to another account B, privilege can be given to B with or without the GRANT OPTION.
- If the **GRANT OPTION** is given, this means that 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 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 <u>A now revokes</u> the privilege granted to B, <u>all the privileges that B propagated based</u> on that privilege should automatically <u>be revoked</u> by the system.

#### 2.5 An Example

- Suppose that the DBA creates four accounts
  - A1, A2, A3, A4
- and wants only A1 to be able to create base relations.
   Then the DBA must issue the following GRANT command in SQL

**GRANT** CREATETAB TO A1;

■ In SQL2 the same effect can be accomplished by having the DBA issue a **CREATE SCHEMA** command as follows:

CREATE SCHAMA EXAMPLE AUTHORIZATION A1;

### 2.5 An Example(2)

- User account <u>A1 can create tables</u> under the schema called **EXAMPLE**.
- Suppose that A1 creates the two base relations
   EMPLOYEE and DEPARTMENT
  - A1 is then owner of these two relations and hence <u>all the</u> relation privileges on each of them.
- Suppose that A1 wants to grant A2 the privilege to insert and delete tuples in both of these relations, but A1 does not want A2 to be able to propagate these privileges to additional accounts:

```
GRANT INSERT, DELETE ON

EMPLOYEE, DEPARTMENT TO A2;
```

#### 2.5 An Example(3)

#### **EMPLOYEE**

Name	Ssn	Bdate	Address	Sex	Salary	Dno
				0011	- c	

#### DEPARTMENT

Dnumber Dname Mgr\_ssn

#### Figure 23.1

Schemas for the two relations EMPLOYEE and DEPARTMENT.

### 2.5 An Example(4)

- Suppose that A1 wants to allow A3 to retrieve information from either of the two tables and also to be able to propagate the SELECT privilege to other accounts.
- A1 can issue the command:
  - GRANT SELECT ON EMPLOYEE, DEPARTMENT TO A3 WITH GRANT OPTION;
- A3 can grant the SELECT privilege on the EMPLOYEE relation to A4 by issuing:
  - GRANT SELECT ON EMPLOYEE TO A4;
    - Notice that A4 can't propagate the SELECT privilege because GRANT OPTION was not given to A4

### 2.5 An Example(5)

Suppose that A1 decides to revoke the SELECT privilege on the EMPLOYEE relation from A3; A1 can issue:

REVOKE SELECT ON EMPLOYEE FROM A3;

■ The DBMS must now automatically revoke the SELECT privilege on EMPLOYEE from A4, too, because A3 granted that privilege to A4 and A3 does not have the privilege any more.

### 2.5 An Example(6)

- Suppose that A1 wants to give back to A3 a limited capability to SELECT from the EMPLOYEE relation and wants to allow A3 to be able to propagate the privilege.
  - The limitation is to retrieve only the NAME, BDATE, and ADDRESS attributes and only for the tuples with DNO=5.
- A1 then create the 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 ASEMPLOYEE TO AS WITH GRANT OPTION;
```

### 2.5 An Example(7)

- Finally, suppose that A1 wants to allow A4 to update only the SALARY attribute of EMPLOYEE;
- A1 can issue:

```
GRANT UPDATE ON EMPLOYEE (SALARY) TO A4;
```

- The UPDATE or INSERT privilege can specify particular attributes that may be updated or inserted in a relation.
- Other privileges (SELECT, DELETE) are not attribute specific.

# 2.6 Specifying Limits on Propagation of Privileges

- Techniques to limit the propagation of privileges have been developed, although they have not yet been implemented in most DBMSs and are not a part of SQL.
  - Limiting horizontal propagation to an integer number i means that an account B given the GRANT OPTION can grant the privilege to at most i other accounts.
  - Vertical propagation is more complicated; it limits the depth of the granting of privileges.

### 3 Mandatory Access Control and Role-Based Access Control for Multilevel Security

- The discretionary access control techniques of granting and revoking privileges on relations has traditionally been the main security mechanism for relational database systems.
- This is an all-or-nothing method:
  - A user either has or does not have a certain privilege.
- In many applications, and additional security policy is needed that classifies data and users based on security classes.
  - This approach as mandatory access control, would typically be combined with the discretionary access control mechanisms.

### 3 Mandatory Access Control and Role-Based Access Control for Multilevel Security(2)

- Typical security classes are top secret (TS), secret (S), confidential (C), and unclassified (U), where TS is the highest level and U the lowest: TS ≥ S ≥ C ≥ U
- The commonly used model for multilevel security, known as the Bell-LaPadula model, classifies each **subject** (user, account, program) and **object** (relation, tuple, column, view, operation) into one of the security classifications, T, S, C, or U:
  - Clearance (classification) of a subject S as class(S) and to the classification of an object O as class(O).

## 3 Mandatory Access Control and Role-Based Access Control for Multilevel Security(3)

- Two restrictions are enforced on data access based on the subject/object classifications:
  - Simple security property: A subject S is not allowed read access to an object O unless class(S) ≥ class(O).
  - A subject S is not allowed to write an object O unless class(S) ≤ class(O). This known as the star property (or \* property).

## 3 Mandatory Access Control and Role-Based Access Control for Multilevel Security(4)

- To incorporate multilevel security notions into the relational database model, it is common to consider attribute values and tuples as data objects.
- Hence, each attribute A is associated with a classification attribute
   C in the schema, and each attribute value in a tuple is associated with a corresponding security classification.
- In addition, in some models, a tuple classification attribute TC is added to the relation attributes to provide a classification for each tuple as a whole.
- Hence, a multilevel relation schema R with n attributes would be represented as
  - $R(A_1,C_1,A_2,C_2, ..., A_n,C_n,TC)$
- where each Ci represents the classification attribute associated with attribute A<sub>i</sub>.

## 3 Mandatory Access Control and Role-Based Access Control for Multilevel Security(5)

- The value of the **TC** attribute in each tuple t which is the highest of all attribute classification values within t provides a general classification for the tuple itself, whereas each C<sub>i</sub> provides a finer security classification for each attribute value within the tuple.
  - The apparent key of a multilevel relation is the set of attributes that would have formed the primary key in a regular(single-level) relation.

## 3 Mandatory Access Control and Role-Based Access Control for Multilevel Security(6)

- A multilevel relation will appear to contain different data to subjects (users) with different clearance levels.
  - In some cases, it is possible to store a single tuple in the relation at a higher classification level and produce the corresponding tuples at a lower-level classification through a process known as **filtering**.
  - In other cases, it is necessary to store two or more tuples at different classification levels with the same value for the apparent key.
- This leads to the concept of polyinstantiation where several tuples can have the same apparent key value but have different attribute values for users at different classification levels.

## 3 Mandatory Access Control and Role-Based Access Control for Multilevel Security(7)

- In general, the entity integrity rule for multilevel relations states that all attributes that are members of the apparent key must not be null and must have the same security classification within each individual tuple.
- In addition, all other attribute values in the tuple must have a security classification greater than or equal to that of the apparent key.
  - This constraint ensures that a user can see the key if the user is permitted to see any part of the tuple at all.

## 3 Mandatory Access Control and Role-Based Access Control for Multilevel Security(8)

Other integrity rules, called null integrity and interinstance integrity, informally ensure that if a tuple value at some security level can be filtered (derived) from a higher-classified tuple, then it is sufficient to store the higher-classified tuple in the multilevel relation.

# 3.1 Comparing Discretionary Access Control and Mandatory Access Control

- Discretionary Access Control (DAC) policies are characterized by a high degree of flexibility, which makes them suitable for a large variety of application domains.
  - The main drawback of **DAC** models is their vulnerability to malicious attacks, such as Trojan horses embedded in application programs.

# 3.1 Comparing Discretionary Access Control and Mandatory Access Control(2)

- By contrast, mandatory policies ensure a high degree of protection in a way, they prevent any illegal flow of information.
- Mandatory policies have the drawback of being too rigid and they are only applicable in limited environments.
- In many practical situations, discretionary policies are preferred because they offer a better trade-off between security and applicability.

#### 3.2 Role-Based Access Control

- Role-based access control (RBAC) emerged rapidly in the 1990s as a proven technology for managing and enforcing security in large-scale enterprisewide systems.
- Its basic notion is that permissions are associated with roles, and users are assigned to appropriate roles.
- Roles can be created using the CREATE ROLE and DESTROY ROLE commands.
  - The GRANT and REVOKE commands discussed under DAC can then be used to assign and revoke privileges from roles.

#### 3.2 Role-Based Access Control(2)

- RBAC appears to be a viable alternative to traditional discretionary and mandatory access controls; it ensures that only authorized users are given access to certain data or resources.
- Many DBMSs have allowed the concept of roles, where privileges can be assigned to roles.
- Role hierarchy in RBAC is a natural way of organizing roles to reflect the organization's lines of authority and responsibility.

### 3.2 Role-Based Access Control(3)

- Another important consideration in RBAC systems is the possible temporal constraints that may exist on roles, such as time and duration of role activations, and timed triggering of a role by an activation of another role.
- Using an RBAC model is highly desirable goal for addressing the key security requirements of Web-based applications.
- In contrast, discretionary access control (DAC) and mandatory access control (MAC) models lack capabilities needed to support the security requirements emerging enterprises and Web-based applications.

## 3.3 Access Control Policies for E-Commerce and the Web

- E-Commerce environments require elaborate policies that go beyond traditional DBMSs.
  - In an e-commerce environment the resources to be protected are not only traditional data but also knowledge and experience.
  - The access control mechanism should be flexible enough to support a wide spectrum of heterogeneous protection objects.
- A related requirement is the support for contentbased access-control.

## 3.3 Access Control Policies for E-Commerce and the Web(2)

- Another requirement is related to the heterogeneity of subjects, which requires access control policies based on user characteristics and qualifications.
  - A possible solution, to better take into account user profiles in the formulation of access control policies, is to support the notion of credentials.
  - A credential is a set of properties concerning a user that are relevant for security purposes
    - For example, age, position within an organization
  - It is believed that the XML language can play a key role in access control for e-commerce applications.

# 4 Introduction to Statistical Database Security

- Statistical databases are used mainly to produce statistics on various populations.
- The database may contain confidential data on individuals, which should be protected from user access.
- Users are permitted to retrieve statistical information on the populations, such as averages, sums, counts, maximums, minimums, and standard deviations.

# 4 Introduction to Statistical Database Security(2)

- A population is a set of tuples of a relation (table) that satisfy some selection condition.
- Statistical queries involve applying statistical functions to a population of tuples.

# 4 Introduction to Statistical Database Security(3)

- For example, we may want to retrieve the *number* of individuals in a **population** or the *average income* in the population.
  - However, statistical users are not allowed to retrieve individual data, such as the income of a specific person.
- Statistical database security techniques must prohibit the retrieval of individual data.
- This can be achieved by prohibiting queries that retrieve attribute values and by allowing only queries that involve statistical aggregate functions such as COUNT, SUM, MIN, MAX, AVERAGE, and STANDARD DEVIATION.
  - Such queries are sometimes called statistical queries.

# 4 Introduction to Statistical Database Security(4)

- It is DBMS's responsibility to ensure confidentiality of information about individuals, while still providing useful statistical summaries of data about those individuals to users. Provision of **privacy protection** of users in a statistical database is paramount.
- In some cases it is possible to infer the values of individual tuples from a sequence statistical queries.
  - This is particularly true when the conditions result in a population consisting of a small number of tuples.

#### 5 Introduction to Flow Control

- Flow control regulates the distribution or flow of information among accessible objects.
- A flow between object X and object Y occurs when a program reads values from X and writes values into Y.
  - Flow controls check that information contained in some objects does not flow explicitly or implicitly into less protected objects.
- A flow policy specifies the channels along which information is allowed to move.
  - The simplest flow policy specifies just two classes of information:
    - confidential (C) and nonconfidential (N)
  - and allows all flows except those from class C to class N.

### Summary

- 1 Database Security and Authorization
- 2 Discretionary Access Control
- 3 Mandatory Access Control and Role-Based Access Control for Multilevel Security
- 4 Statistical Database Security
- 5 Flow Control
- 6 Encryption and Public Key Infrastructures