CS5300
Database Systems

Integrity and Security

A.R. Hurson
323 CS Building
hurson@mst.edu

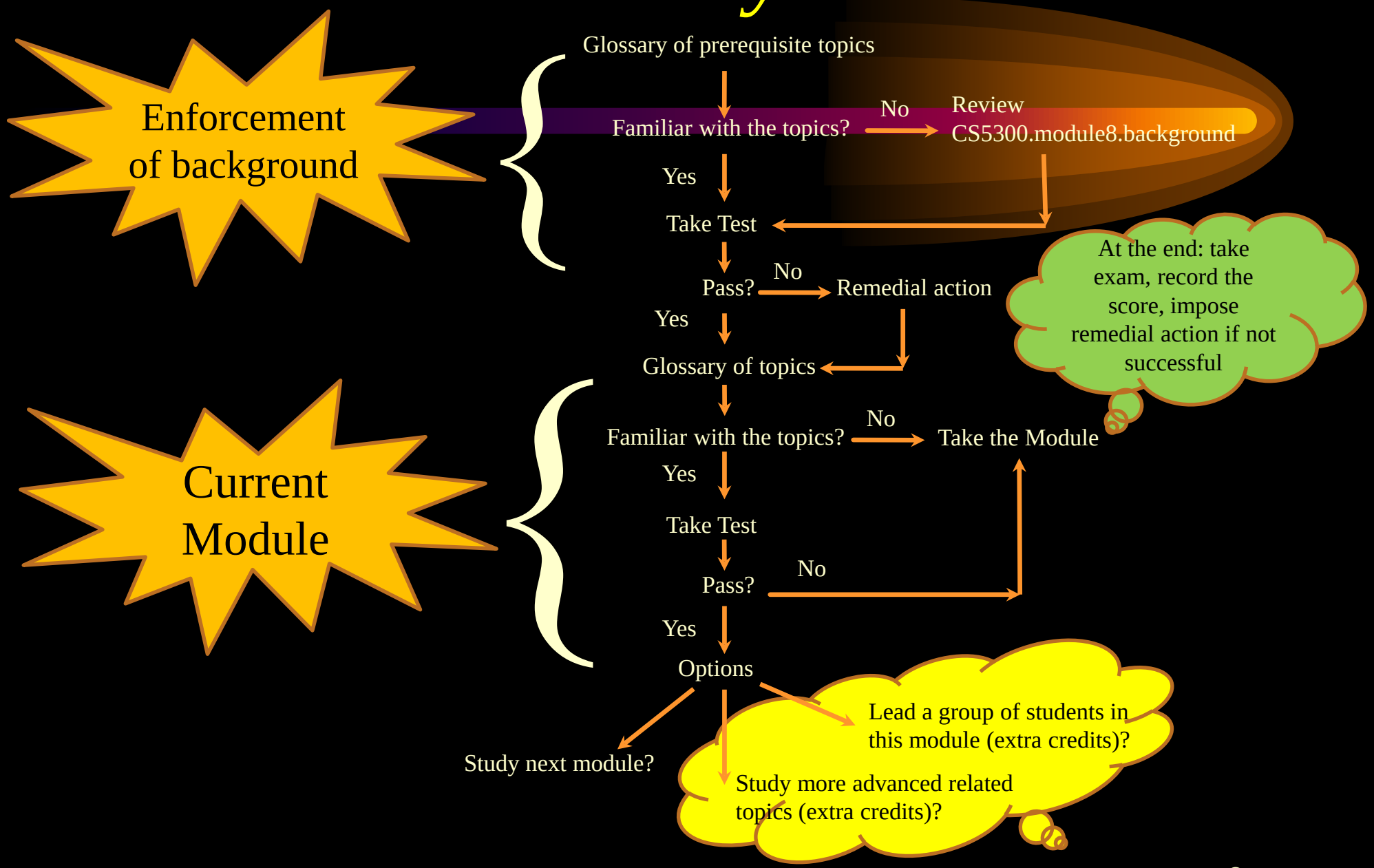
Database Systems

Note, this unit will be covered in three lectures. In case you finish it earlier, then you have the following options:

1) Take the early test and start CS5300.module9
2) Study the supplement module (supplement CS5300.module8)
3) Act as a helper to help other students in studying CS5300.module8

Note, options 2 and 3 have extra credits as noted in course outline.

Database Systems



Database Systems

- You are expected to be familiar with:
  - Relational database model,
  - SQL
  - Transaction processing and concurrency control
- If not, you need to study CS5300.module8.background

Database Systems

- New Concerns
  - It was estimated that 7000 new computer viruses were discovered in 2003 and the FBI approximated that computer viruses cost businesses $27 million during that time.
  - In 2003, unsolicited email, commonly known as spam, cost businesses $20 billion worldwide due to lost of productivity, system overhead, user support, and anti-spam software.

Database Systems

- Data stored in the database need to be protected from unauthorized access and malicious destruction or alteration.
- In addition, it should be protected against accidental introduction of inconsistency that integrity constraints provide.

Database Systems

- Integrity constraints ensure that accidental/malicious changes made to the database by authorized users do not result in a loss of data consistency. Therefore, integrity constraints guard against damage to the database.

Database Systems

- Type of integrity constraints supported by SQL are:
  - Domain constraints, and
  - Referential constraints

Database Systems

- Domain Constraints
  - Proper definition of domain constraints not only allows one to test values inserted in database, but also permits to test queries to ensure that the comparisons made make sense.

Database Systems

- Domain Constraints
  - Consider the following examples:

    Create domain Dollars Numeric (12,2)

    Create domain Pounds Numeric (12,2)

  - An attempt to assign a value of type Dollars to a variable of type Pounds would result a syntax error.

Database Systems

- Domain Constraints
  - The check clause in SQL permits the schema designer to specify a predicate that must be satisfied by any value assigned to a variable whose type is the domain.

Database Systems

- Domain Constraints

```
Create domain      Hourlywage      numeric (5,2)
    Constraint     wage-value-test check (value >=4.00)

Create domain      AccountNumber       char (10)
    Constraint     account-number-null-test      check
    (value not null)

Create domain      AccountType   char (10)
    Constraint     account-type-test      check   (value   in
    ('checking', 'saving'))
```

Database Systems

- Referential Constraints
  - Often, we want to ensure that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation. This constraint is called referential integrity (i.e., foreign key).

Database Systems

- Referential Constraints
  - Let $r_1(R_1)$ and $r_2(R_2)$ be relations with primary keys k_1 and k_2, respectively. We say a subset α of R_2 is a foreign key referencing k_1 in r_1 if it is required that for every $t_2 \varepsilon r_2$ there is a tuple $t_1 \varepsilon r_1$ such that $t_1[k_1] = t_2[\alpha]$ (referential integrity constraints or subset dependencies).

Database Systems

- Referential Constraints
  - Database modifications can cause violations of referential integrity:
  - Insert: If a tuple t_2 is inserted into r_2, then the system must ensure that there is a tuple $t_1 \varepsilon r_1$ such that $t_1[k_1] = t_2[\alpha]$.

Database Systems

- Referential Constraints
  - Delete: If a tuple t_1 is deleted from r_1, then the system must compute the set of tuples t_2 that reference it $\sigma_\varepsilon = t_1[k](r_2)$.
  - If this set is not empty then the delete command must be rejected or tuples that reference t_1 must be deleted (this could result in cascading deletions).

Database Systems

- Referential Constraints
  - Update: Two cases must be considered:
    - If a tuple $t_2 \in r_2$ is updated and update modifies the value of foreign key α, then a test similar to the insert case needs to be made.
    - If a tuple $t_1 \in r_1$ is updated and update modifies the value of primary key (k), then a test similar to the delete case needs to be made.

Database Systems

- Referential Integrity in SQL
  - Foreign key are specified as part of SQL create table statement.
  - When a referential integrity constraint is violated, the normal procedure is to reject the action that caused the violation.

Database Systems

- Referential Integrity in SQL
  - However, a foreign key clause can specify that if a delete or update action on the referenced relation violates the constraints, then instead of rejection, the system must take proper steps to change the tuple in the referencing relation to restore the constraint.

Database Systems

- Referential Integrity in SQL

```
Create table       account
        (….
        Foreign key (branch-name)       reference branch
                                On delete cascade
                                On update cascade,
        ….)
```

Database Systems

- Referential Integrity in SQL
  - SQL also allows the foreign key clause to specify actions other than cascade, if the constraint is violated: the referencing field can be set to null (using set null) or to the default value for the domain (using set default).

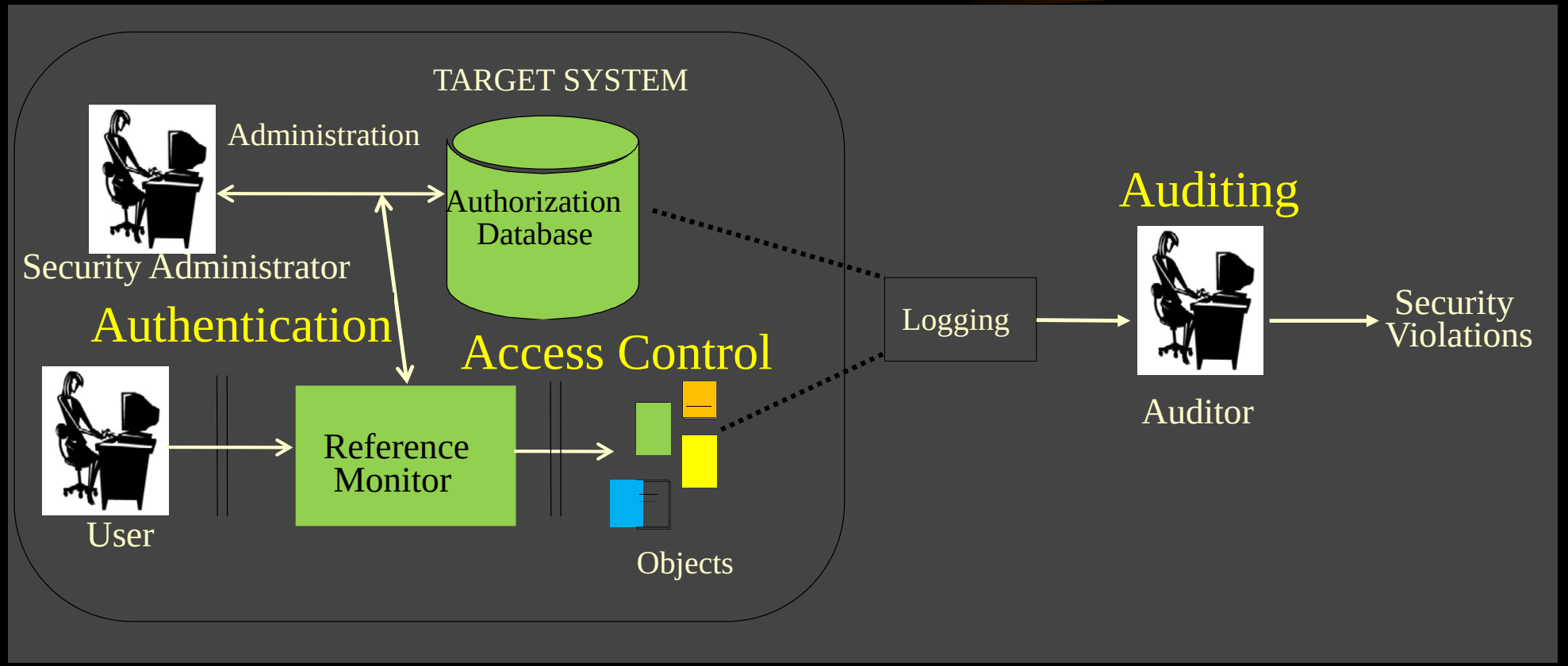
Database Systems

- Definition
  - Security is defined as the science of protecting computers, network resources, and information against unauthorized access, modification, and destruction.

Database Systems

- Three interrelated technologies can be used to achieve information confidentiality and integrity in traditional DBMSs:
  - Authentication,
  - Access control, and
  - Audit

Database Systems



Database Systems

- Authentication identifies one party to another. It ensures the true identity of a user, computer, or process.
- Once the identity is established, the party can access data under the guidance of the access control service.

Database Systems

- Authentication can be performed in both directions:
  - In a peer-to-peer network, peer computers need to identify each other.
  - In a client-server environment, the client should identify itself to the server to obtain services. Authenticating the server to the client is also important to prevent spoofing attacks and protection of sensitive client information.

Database Systems

- Authentication mechanisms are often based on one or more of the following techniques:
  - Knowledge-based authentication – something that "you know" to identify you,
  - Token-based authentication – something that "you possess" to identify you,
  - Biometric-based authentication – your physiological or behavioral characteristics to identify you, and
  - Recognition-based authentication – what you can recognize to identify you.

Database Systems

- Knowledge-based authentication
  - Because of its low cost and ease of implementation, password is the most common mechanism used in Knowledge-based authentication.

Database Systems

- Knowledge-based authentication
  - Several well-known problems associated with passwords are:
    - It is easy to discover,
    - It can be snooped,
    - It can be shared,
      -
      -
      -

Database Systems

- Knowledge-based authentication
  - Techniques have been proposed to overcome the shortcomings of password:
    - User Education in order to choose a strong password.
    - Random Password Generation to eliminate weak password choices.
    - Reactive Password Checking to check and crack a password and inform the user about the weak password.
    - Proactive Password Checking to scan weak password before being used.
    - One-Time Password where password changes randomly say minute by minute.

Token-based authentication

- SmartCards are credit card-sized cards with more memory than the traditional magnetic strip and an on-board embedded processor.
- SmartCards can be either a memory card or a processor-enabled card.
  - In memory card, memory is read only, so the information flows from the card to the card reader.
  - In processor-enabled card cryptography can be used to protect the stored information on the card.

Database Systems

- Biometric-based authentication
  - Both knowledge-based and token-based authentication schemes suffer from the fact that they may not be always available.
  - In addition, these systems cannot identify authorized and unauthorized person — an imposter can use them and hence can have access to the system.
  - Biometric-based scheme does not suffer from these disadvantages — voice recognition, finger prints, hand and finger geometry, retinal scan, iris scan, …

Database Systems

- Biometric-based authentication
  - Biometric technique can be used for two purposes:
    - Biometric identification identifies a person from the database of persons known to the system – Who am I?
    - Biometric verification authenticates a claimed identity – Am I who I claim I am?
  - Biometric-based authentication is a two step process:
    - Enrollment phase,
    - Identification phase.

Database Systems

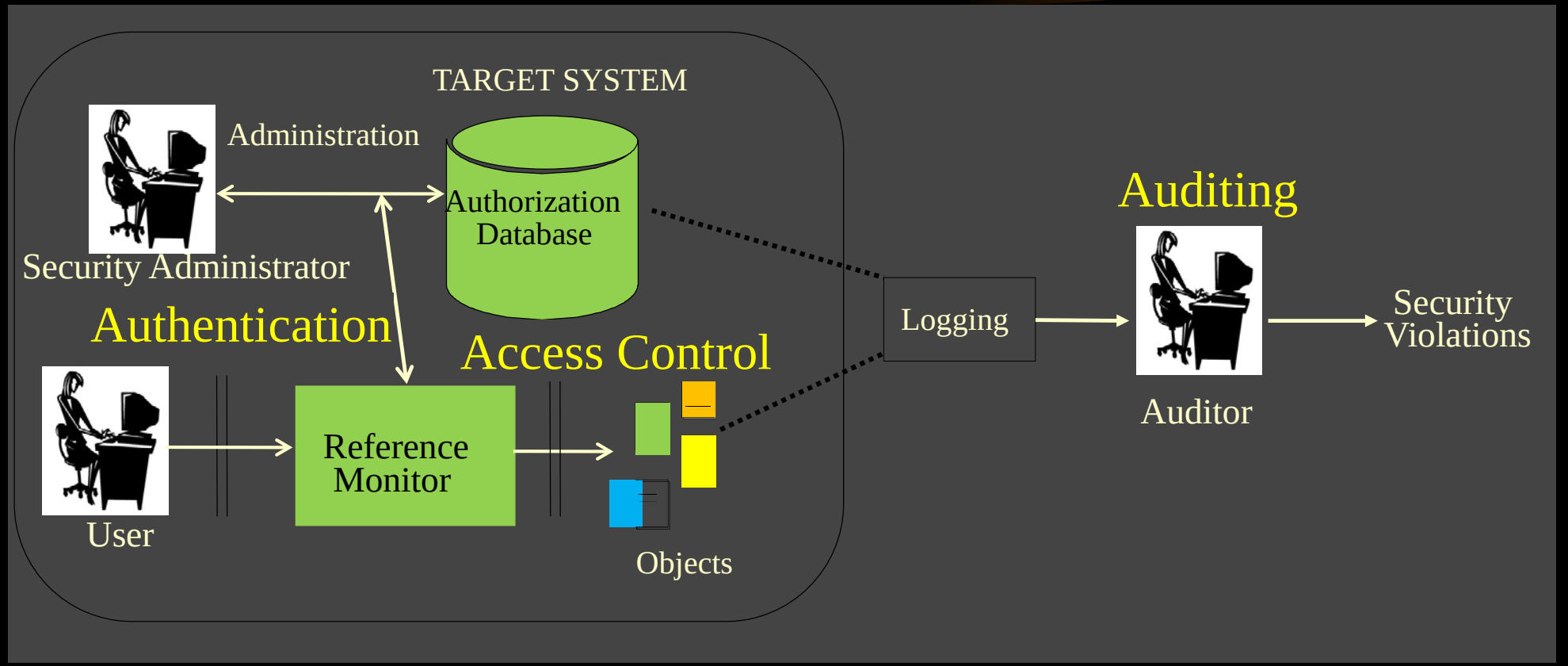
- Recognition-based authentication
  - This technique is based on the studies that have shown it is much easier to recognize something that to recall the same information from memory without help — Image recognition, graphical password.

Database Systems

Authentication Methods — Comparative analysis

Method	Based On	Advantage	Disadvantage
Knowledge-Based	"Something that you know"	• Simple	• Password reusability • Passwords can be forgotten, shared, or observed
Token-Based	"Something that you possess"	• Simple • Overcomes some of weaknesses of knowledge-based systems	• Tokens can be forgotten, lost, stolen, or duplicated
Biometric-Based	Physiological or behavioral characteristic	• Greater security than other methods • Can distinguish between authorized person and imposters	• Privacy issues • Social acceptance
Recognition-Based	Recognition of previously viewed images	• Can overcome the drawbacks of knowledge- and token-based systems	• Most current proposals are not truly recognition based

Database Systems



Database Systems

- Access control (authorization) regulations are set by the system security administrator and define who can access what data with which privileges.
  - The prerequisite of an access control system is successful user authentication.
  - Authentication and access control do not comprise a complete security solution — they must be complemented by an auditing service.

Database Systems

- Access control (authorization)
  - Malicious access comes in different forms:
    - Unauthorized reading of data,
    - Unauthorized modification of data
    - Unauthorized destruction of data

Database Systems

- Access Control
  - The tasks of an access control system include:
    - Defining access policy,
    - Determining and granting access rights for a specific user, and
    - Enforcing the execution of those rights.

Database Systems

- Access Control
  - It is necessary to distinguish the access control mechanism from the authorization policy.
    - Authorization Policies are high-level guidelines that determine how accesses are controlled and access decisions are determined. It is usually the job of the database security administrator to define such policies.
    - Access Control Mechanisms are low-level software and hardware functions that can be configured to implement a policy. The choice of different mechanisms is usually made by DBMS designers — Access control mechanisms should be flexible and comprehensive enough to enforce a variety of policies.

Database Systems

- We may assign a user several forms of access right on parts of the database. For example:
  - Read authorization,
  - Insert authorization,
  - Update authorization,
  - Delete authorization
- We may assign the user all, none, or a combination of these types of access right.

Database Systems

- We may also grant a user authorization to modify the database schema. For example:
  - Index authorization (to allow creation and deletion of indices),
  - Resource authorization (to allow creation of new relations),
  - Alteration authorization (to allow addition or deletion of attributes in a relation),
  - Drop authorization (to allow deletion of relations)

Database Systems

- Access Control
  - Subjects and objects are the two basic principals in an access control system.
    - Subjects are the entities that initiate activities in the system. Subjects can be users or processes executing on behalf of users.
    - Objects are the entities behind the protection of the system. In a DBMS environment, an object can be a field, a record, a table, a view, etc.
  - There is a many-to-many relationship between subjects and users: a user can log on to the system as different subjects and a subject can represent different users.

Database Systems

- Access Control
  - Operations that subjects can perform on objects (access rights) are usually defined by the security administrator and represented in the form of an access matrix, or one of its alternatives.
  - For databases, the typical access rights include own, read, and write.
  - Normally, the owner of a database is authorized to grant and revoke access rights to the database.

Database Systems

- Access Control – Access Matrix
  - An access matrix is a common approach to modeling the access rights of subjects with respect to objects. Each row in the matrix represents a subject and a column corresponds to each object. Each entry in this matrix defines the access rights of a subject over an object.

Database Systems

- Access Control – Access Matrix
  - The following shows an access matrix example – *O* represents ownership, *R* denotes the read right, and *W* represents the write right.

Object / Subject	Table1	Table2	Table3	Table4
Alice	O, R, W		O, R, W	R
Bob		O, R, W		R
Cathy	R			O, R, W

  - There are three subjects: Alice, Bob, and Cathy and four objects, Tables 1 – 4. Alice is the owner of Table 1 and Table 3, and therefore, she has the read and write rights to those tables. In addition, the matrix also specifies that Alice has the read right on Table 4, but she does not have access rights to Table 2.

Database Systems

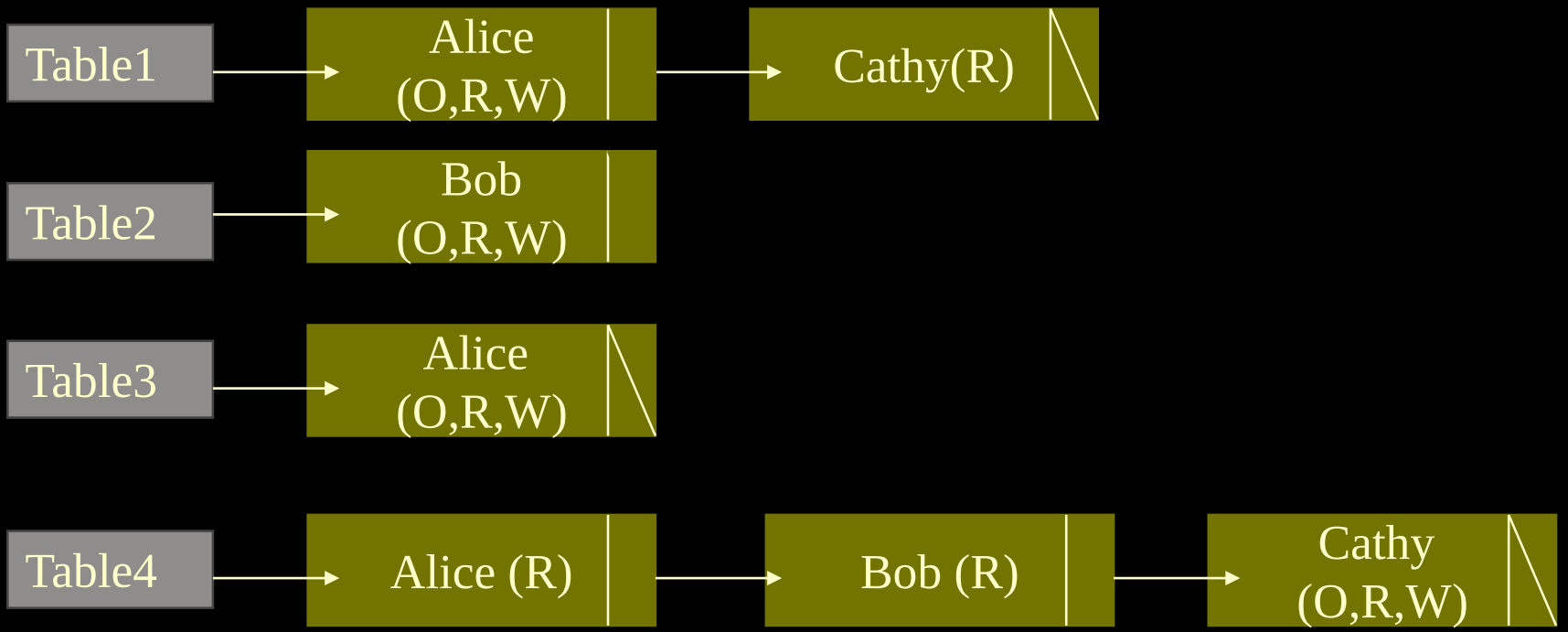
- Access Control – Access Matrix
  - When a user, say Bob, requests a read operation on Table 3, the request is sent to a program called the reference monitor. The reference monitor checks the access matrix to find out whether Bob has the read access right to Table 3.
  - In previous example, Bob does not have any access rights to Table 3. Thus, the Bob's read request will be rejected.

Database Systems

- Access Control – Access Control List
  - Each object is associated with a list of subjects and their access rights to that object – Columns of access control Matrix are represented as link lists to eliminate the empty entries.
  - This approach:
    - Provides a convenient way to find out who can access an object.
    - Allows one to easily modify the access rights associated with each object.
    - Allows one to easily revoke access to an object from all subject.

Database Systems

- Access Control – Access Control List

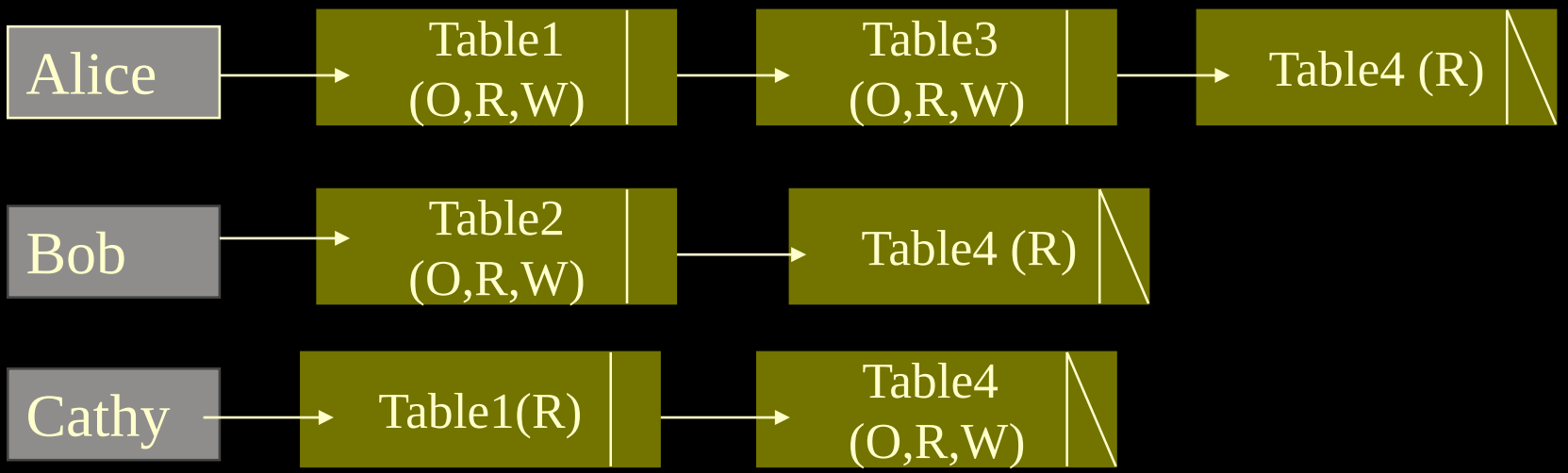


Database Systems

- Access Control – Capability List
  - A capability List is the list of objects that a subject can operate on along the access rights associated with each object that subject has – it stores access matrix by rows.
    - This approach allows one to easily modify or revoke the access of a subject to all objects.

Database Systems

- Access Control – Capability List



Database Systems

- Access Control – Combinatory Approach
  - This approach is intended to get advantage of both Access Control List and Capability List – Access Matrix is represented as an authorization relation.
  - In this relation, each tuple is a triplet:
    - Subject
    - Access Right
    - Object

Database Systems

Access Control – Combinatory Approach

Authorization relation

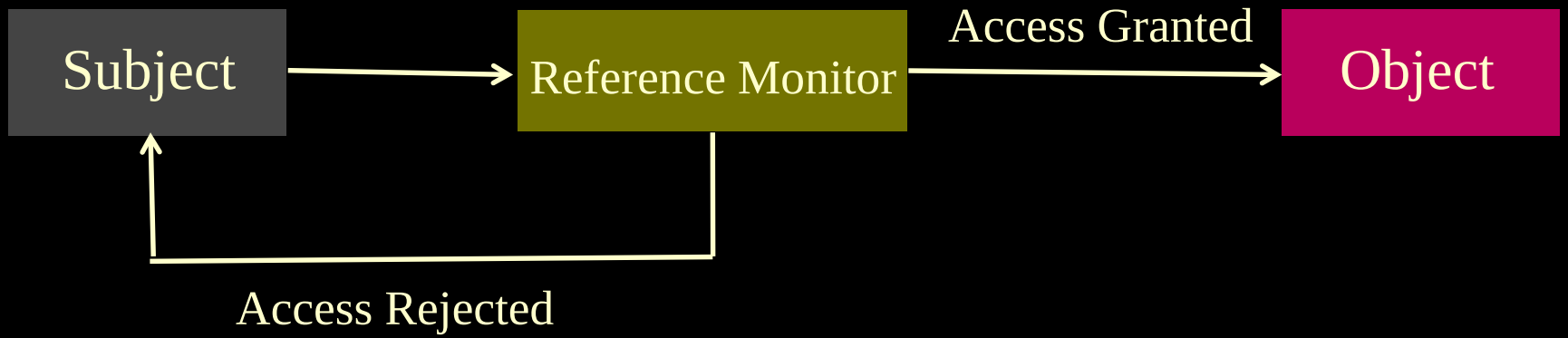
Subject	Access Right	Object
Alice	O	Table1
Alice	R	Table1
Alice	W	Table1
Alice	O	Table3
Alice	R	Table3
Alice	W	Table3
Alice	R	Table4
Bob	O	Table2
Bob	R	Table2
Bob	W	Table2
Bob	R	Table4
Cathy	R	Table1
Cathy	O	Table4
Cathy	R	Table4
Cathy	W	Table4

If sorted by subjects it is
a collection of capability lists

If sorted by objects it is
a collection of access control lists

Database Systems

- Access Control – Access Matrix



Database Systems

- Access Control Policy
  - Generally three classes of access control policies are commonly used:
    - Discretionary Access Control (DAC) Policies,
    - Mandatory Access Control (MAC) Policies, and
    - Role-Based Access Control (RBAC) Policies.

Database Systems

- Discretionary Access Control (DAC) Policies govern subject's access to the object based on the subject's identity and authorization rules.
- Discretionary access control policies cannot control the flow of information.

Database Systems

- Discretionary Access Control Policies can be classified further as:
  - Closed discretionary policies assume a denial of access unless specified otherwise by the authorization rules.
  - Open discretionary policies assume that an access request can be granted unless stated explicitly otherwise by the rules.

Database Systems

- Mandatory access control (MAC) policies define authorization rules based on the classification of subjects and objects in the system.
  - Hierarchical classifications are made within the domains of confidentiality, integrity, and sensitivity of the object.
  - System security is achieved by enforcing a set of read/write rules among the hierarchies.

Database Systems

- Mandatory access control policies
  - The BLP Model: Within the scope of information confidentiality (secrecy), each subject and object is assigned a security label:
    - A subject's security label, called security clearance, reflects the subject's trustworthiness,
    - An object's security label, called security classification, reflects the sensitivity of the object,
    - One example could be; TS (top secret), S (secret), C (confidential), and U (unclassified) ,

Database Systems

- Mandatory access control policies (BLP Model)
  - Let λ denote the security label of a subject or object. The mandatory access rules specified in the BLP model are as follows:
    - Simple-Security Property (read rule): Subject *s* can read object *o* only if $\lambda(s) \geq \lambda(o)$.
    - *-Property (write rule): Subject *s* can write object *o* only if $\lambda(s) \leq \lambda(o)$.

Database Systems

- Mandatory access control policies
  - The Biba Model: Within the scope of information integrity, each subject and object is assigned an integrity label.
    - A subject's integrity level reflects the subject's trustworthiness for modifying information.
    - An object's integrity level indicates the reliability of the information stored in that object, and the potential damage that could result from unauthorized modification.
    - One example of the integrity hierarchy could be C (crucial), I (important), and U (unknown).

Database Systems

- Mandatory access control policies (Biba Model)
  - Let ω denote the integrity label of a subject or object. The mandatory access rules specified in the Biba model are as follows:
    - Simple-Integrity Property (read rule): Subject *s* can read object *o* only if $\omega(s) \leq \omega(o)$.
    - Integrity *-Property (write rule): Subject *s* can write object *o* only if $\omega(s) \geq \omega(o)$.

Database Systems

- Mandatory access control policies (Composite Model)
  - In this model, each subject and object is assigned two labels: a confidentiality label λ and an integrity label ω.
  - The mandatory access rules can be stated as follows:
    - Read Rule: Subject s can read object o only if $\lambda(s) \geq \lambda(o)$ and $\omega(s) \leq \omega(o)$.
    - Write Rule: Subject s can write object o only if $\lambda(s) \leq \lambda(o)$ and $\omega(s) \geq \omega(o)$.

Database Systems

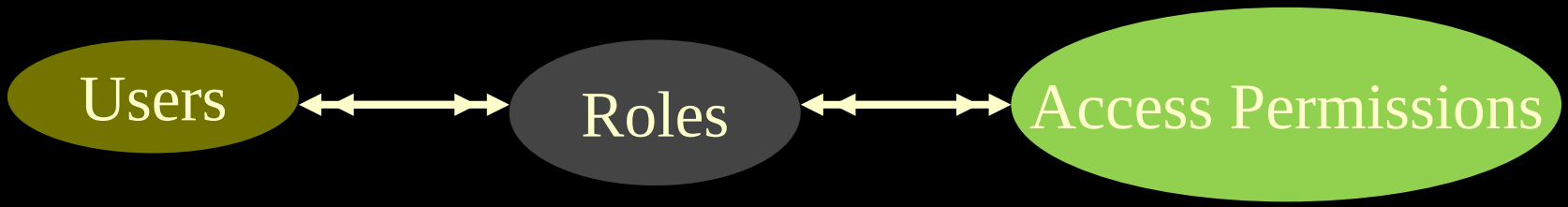
- Access Control Policy (Role-based Model)
  - Role-Based Access Control (RBAC) Policies establish permissions based on the functional roles in the enterprise.
  - Roles can represent tasks and responsibilities within an enterprise: a set of roles must be identified and users are assigned to an appropriate role or a set of roles. Instead of defining access rights to all objects for each user, the RBAC policies specify these rights on a role basis.
  - A role acts as a mediator between a group of users and a set of tasks/responsibilities associated with them.

Database Systems

- Access Control Policy (Role-based Model)
  - Roles in an enterprise are generally persistent and many users can be represented by a single role. Thus, RBAC approaches can reduce the administration complexity, cost, and potential errors.

Database Systems

- Access Control Policy (Role-based Model)
  - Role Based Access Control is based on three set of entities: users, roles, and access permissions.



Database Systems

- Access Control Policy (Role-based Model)
  - The RBAC model can be expressed as many-to-many relations:
    - A user can be a member of several roles and a role can be assigned to several users.
    - A role can have a number of access permissions and a set of permissions can be given to multiple roles.

Database Systems

- Access Control Policy (Role-based Model)
  - Roles can form a role hierarchy to reflect lines of authority. As a result, a role may inherit permissions of another role in the hierarchy.
  - Relationships among roles in the hierarchy may be in a partial order even though some pairs of roles may not be comparable.
  - Three different forms of role hierarchy have been defined:
    - *isa* role hierarchy
    - Activity role hierarchy
    - Supervision role hierarchy

Database Systems

- Access Control Policy (Role-based Model)
  - *isa* role hierarchy is based on generalization.
  - Roles are defined by qualification where one role is more specific than its *isa* role; thus, the role inherits all permissions assigned to its *isa* role. The role may have its own extra permissions.

Database Systems

- Access Control Policy (Role-based Model)
  - Activity role hierarchy is based on aggregation where a role is defined as an aggregation of its component roles. As a result, the role inherits all permissions given to its components.

Database Systems

- Access Control Policy (Role-based Model)
  - Supervision role hierarchy is based on organizational hierarchy of positions where a role at a higher position inherits all permissions of roles at its lower positions.

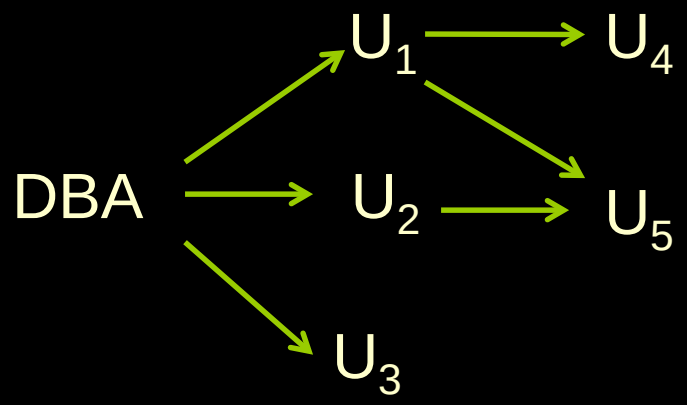
Database Systems

- A user who has been granted some form of authorization may be allowed to pass on this authorization to other users.
- However, we must be careful how authorization may be passed among users, to ensure that such authorization can be revoked at some future time.

Database Systems

- Assume that the database administrator grants update authorization on a relation r to users U_1, U_2, U_3, who may in turn pass on this authorization to other users. This process can be represented by a directed graph, i.e., authorization graph.

Database Systems



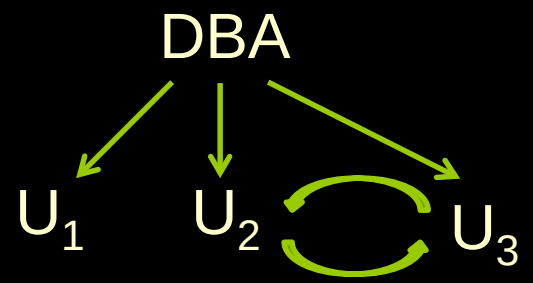
- A user has an authorization if and only if there is a path from the root of the authorization graph down to the node representing the user.

Database Systems

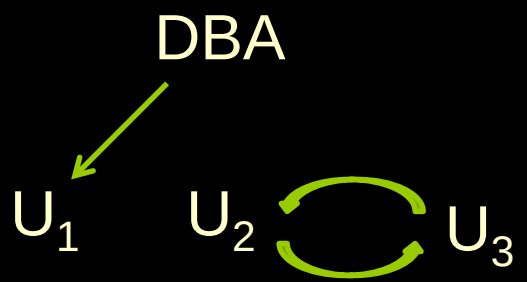
- Assume database administrator decides to revoke the authorization of U_1. Since U_4 has authorization from U_1, that authorization should be revoked as well. However, U_5 had authorization from U_1 and U_2. Since authorization of U_2 has not been revoked. U_5 retains his update authorization.
- A pair of users might attempt to defeat the rules of revocation of authorization by granting authorization to each other.

Database Systems

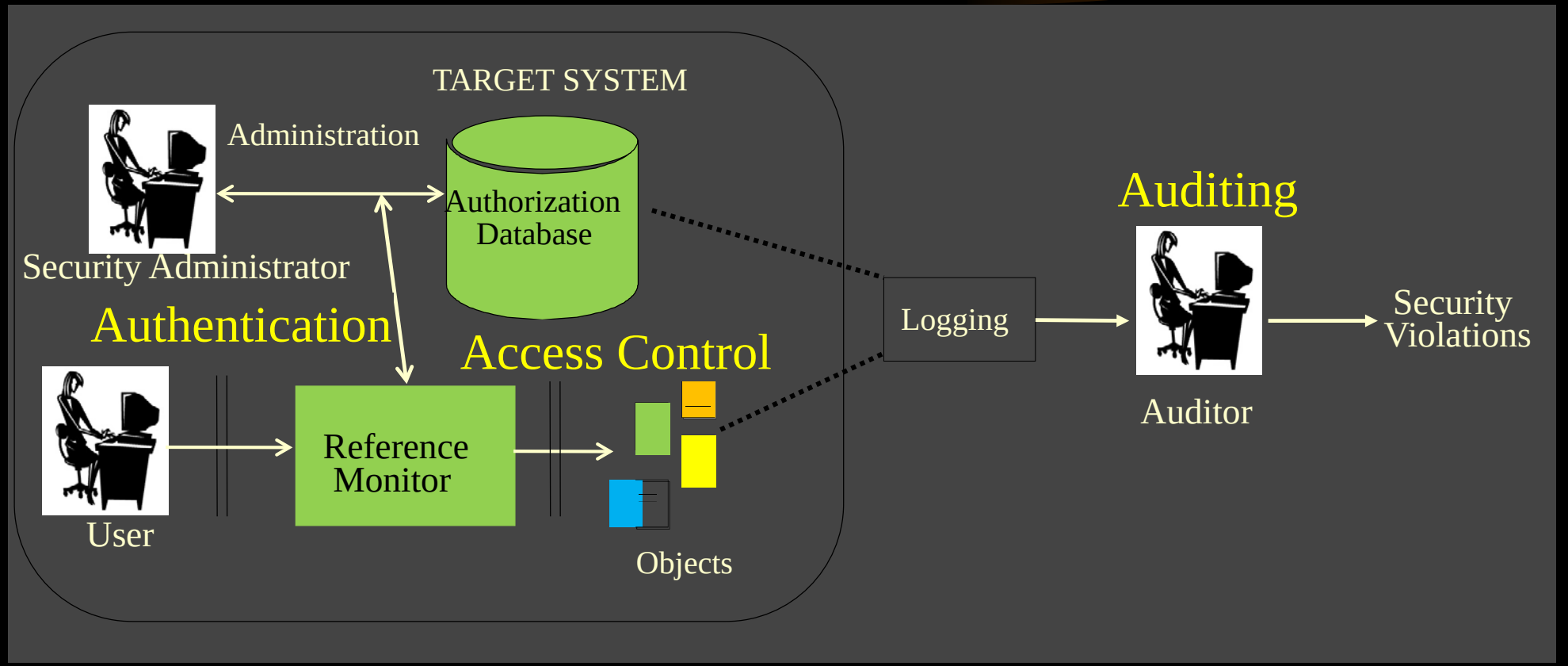
- Consider the following scenario



- Now assume that the DBA revokes authorization from U_2 and U_3. Then we have:



Database Systems



Database Systems

Auditing

- An auditing service records important user activities in system logs for real-time or a posteriori analysis.
- Real-time auditing is often referred to as intrusion detection.
- An audit service protects a system in three ways:
  - Detecting actual security violations;
  - Assisting the security administrator in discovering attempted attacks by recognizing abnormal activity patterns; and
  - Detecting possible system security flaws.

Database Systems

- Auditing
  - Auditing consists of examining the history of events in a system to determine if and how security violations have occurred or been attempted.
  - Auditing is required for all system activities:
    - The subject requesting access,
    - The object to be accessed,
    - The operation requested,
    - Permissions granted or rejected,
    - The resource consumption,
    - The outcome of the request, etc.
  - Audit data is stored in an audit log.

Database Systems

- Auditing
  - Audit data can be collected at different level of granularities:
    - Fine grain,
    - Coarse grain.
  - Intrusion detection systems can be divided into:
    - Passive systems, and
    - Active systems.
  - Passive systems perform a posteriori analysis and bring security violations to the auditor's attention.
  - Active systems perform analysis in real time. Once a violation is detected or suspected, the intrusion detection system alerts the auditor and may take immediate measures for system protection.

Database Systems

- Security in SQL
  - The concept of views was introduced as a means to provide a personalized model of the database. Views provides both simplicity and security.
  - A combination of relational-level security and view-level security limits a user's access to precisely the data that the user needs.

Database Systems

- Consider the following example:

```
create view cust-loan as
    (select  branch-name, customer-name
    from     borrower, loan
    where    borrower.loan-number=loan.loan-number)
```

  - Assume that the clerk issues the following query

```
select  *
from    cust-loan
```

Database Systems

- Clearly the clerk is authorized to see the result of this query. However, this query produces a query on borrower and loan relations. As a result, the system must check authorization of the clerk's query before processing it.

Database Systems

- Authorization in SQL
  - SQL includes the delete, insert, select (read), and update privileges.
  - SQL also grants references privilege on foreign key.

Database Systems

- Authorization in SQL
  - The grant statement as part of data definition language is used to confer authorization.

    grant <privilege list> on <relation name or view name> to <user/role list>

    grant select on account to U_1, U_2, U_3

grant update (amount) on loan to U_1, U_2, U_3

List of Attributes

grant references (branch-name) on branch to U_1,

- By default, a user/role that is granted a privilege is not authorized to grant that privilege to another user/role. Unless we append the with grant option clause to the appropriate grant command.

  grant select on branch to U_1 with grant option

- Revoke statement revokes an authorization.

  revoke <privilege list> on <relation name or view name> from <user/role list> [restrict | cascade]

  revoke select on branch from U_1, U_2, U_3

  revoke update (amount) on loan from U_1, U_2, U_3

  List of Attributes

Database Systems

- Security terminology
  - Plaintext is a message in its unencrypted form, either before the encryption transformation has been applied, or after the corresponding decryption transformation is complete.
  - Ciphertext is the encrypted form of a message — the output of the encryption transformation.
  - Encryption is the process of transforming data into a form that cannot be understood without applying a second transformation. The transformation is affected by an encryption key in such a manner that the second transformation can only be applied by someone in possession of the corresponding decryption key.

Database Systems

- Security terminology
  - A secret-key cryptosystem (symmetric cryptosystem), as defined by the Data Encryption Standard (DES), uses a single key for both encryption and decryption. Such an encryption key is called a secret key.
  - A public-key cryptosystem (asymmetric cryptosystem), such as RSA, uses different keys for encryption and decryption. One of the keys in the pair can be publicly known while the other must be kept private. These keys are referred to as public and private keys, respectively.

Database Systems

- Security terminology
  - The spoofing attack refers to the situation where one computer masquerades as another.
  - The brute force attack (or, exhaustive key space attack) cracks the password by performing an exhaustive search over the entire key space.
  - The dictionary attack tries a list of words, possible weak passwords, and their simple transformations, such as capitalizing, prefixing, suffixing or reversing a word, until the hashed value of the candidate matches a password hash.