Database & Storage Security

Professor Dr. Mohammad Abu Yousuf
yousuf@juniv.edu

Lecture – 2

Multilevel Security : Mandatory Access Control (MAC)

- Definition and need for MAC
  - Security Classification
  - Secrecy-Based Mandatory Policies: **Bell-LaPadula Model**
  - Integrity-based Mandatory Policies: **The Biba Model**
  - Limitation of Mandatory Policies

Multilevel Security : Mandatory Access Control (MAC)

- MACs are based on Security Labels associated with each data items and each user.
- A label on data item is called Security Classification.
- A label on an user is called Security Clearance.
- An user's Security Clearance must be greater than the Security Classification of the data item to be accessed.

Multilevel Security : Military & Govt. Organization

Two components:	
Hierarchical Component	Category Component
Top Secret(TS)	Nuclear
Secret(S)	Conventional
Confidential(C)	Navy
Unclassified	NATO

Multilevel Security : Military & Govt. Organization

- More formally, each object is associated with a security level of the form (classification level, set of categories).
- Each subject is also associated with a maximum and current security level, which can be changed dynamically. The set of classification levels is ordered by a $<$ relationship.
- For instance, it can be the set **top secret, secret, confidential, unclassified**, where **unclassified < confidential < secret < top-secret.**

Multilevel Security : Military & Govt. Organization

- Security of Military & Govt. sectors:
  - Label X is said to dominate Label Y provided that the hierarchical component of X is greater than that of Y and the categories of X contains all the categories of Y.

Multilevel Security (Example)

- Security of Military & Govt. sectors:
  - Example
    - X = {TS,(NUCLEAR,ARMY)} dominates Y={S,(ARMY)}

      because top-secret > secret and the set{Nuclear, Army} contains{Army}.
    - An user with Labe X can have access a file of Label Y.

Multilevel Security (Example)

- Security of Military & Govt. sectors:
  - Example
    - X = {S,(NUCLEAR,ARMY)} dominates Y={S,(NUCLEAR)}
    - X = {TS,(NUCLEAR)} cannot dominates Y={S,(ARMY

Definition and need for Multilevel Security

- Multilevel security involves a database in which the data stored has an associated classification and consequently constraints for their access
- MAC allows users with different classification levels to get different views from the same data
- MAC cannot allow **downward leaking**, meaning that a user with a lower classification views data stored with a higher classification

Bell – LaPadula Model

MAC formulated by Bell-LaPadula

- The **Bell-LaPadula Model (BLM)**, also called the multi-level model, was proposed by Bell and LaPadula for enforcing access control in government and military applications.
- **Bell-LaPadula** model was developed in 1973.
- This is an extension of the Access Matrix model with classified data
- This model has two components:
  - Classification
  - Set of categories
- **Bell-LaPadula model shows how to use Mandatory Access Control to prevent the Trojan Horse**

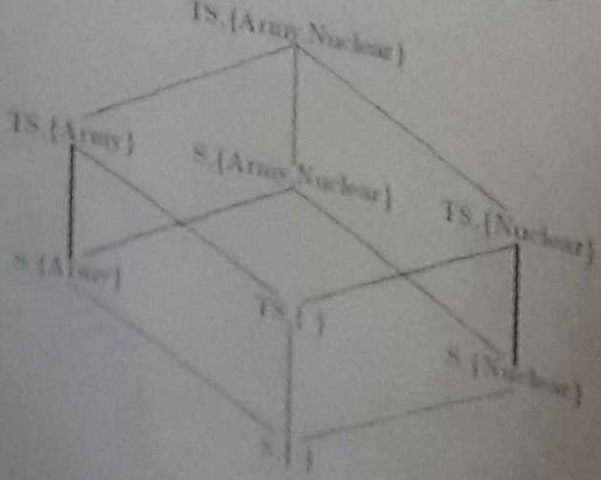
Bell – LaPadula Model

- Classification has four values {U, C, S, TS}
  - U = unclassified
  - C = confidential
  - S = secret
  - TS = top secret
- Classifications are ordered: TS > S > C > U
- Set of categories consists of the data environment and the application area, i.e., Nuclear, Army, Financial, Research

Example: In USA, a "SECRET" clearance involves checking FBI fingerprint files.

Bell – LaPadula Model

- An access class c1 **dominates** ≥ an access class c2 iff
  - Security level of c1 is **greater** than or equal to that of c2
  - The categories of c1 **include** those of c2



Bell – LaPadula Model

- Bell-LaPadula model is based on a subject-object paradigm
- Subjects are active elements of the system that execute actions
- Objects are passive elements of the system that contain information
- Subjects act on behalf of users who have a security level associated with them (indicating the level of system trust)

Bell – LaPadula Model

- In such applications, **subjects** and objects are often partitioned into different security levels. A subject can only access objects at certain levels determined by his security level.

- For instance, the following are two typical access specifications: ``**Unclassified personnel cannot read data at confidential levels**'' and ``**Top-Secret data cannot be written into the files at unclassified levels**''.

Bell – LaPadula Model

- Subjects execute access modes on objects
- Access modes are:
  - Read-only
  - Append (writing without reading)
  - Execute
  - Read-write (writing known data)
- Decentralized administration of privileges on objects

Bell – LaPadula Model

❑ Simple Security property:

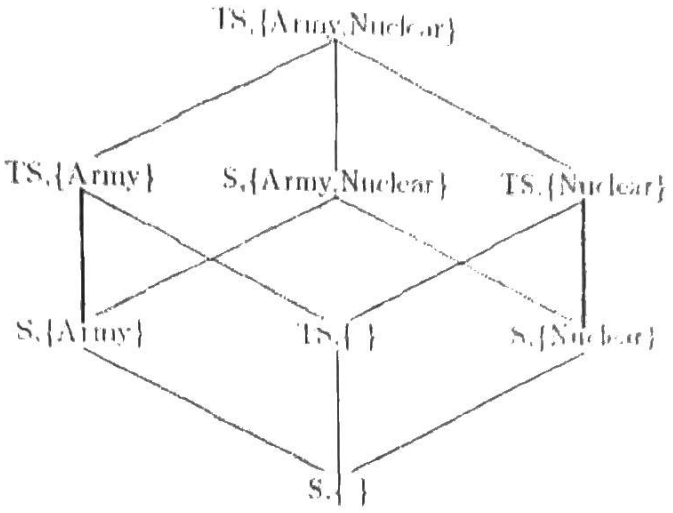
1) If access = read, then **level(Subj)** should dominate **level(Obj).**

2) if access = append, then **level(Obj)** should dominate **level(Subj);**

3) if acc = write, then **level(Obj)** should be equal to **level(Subj).**

Bell – LaPadula Model

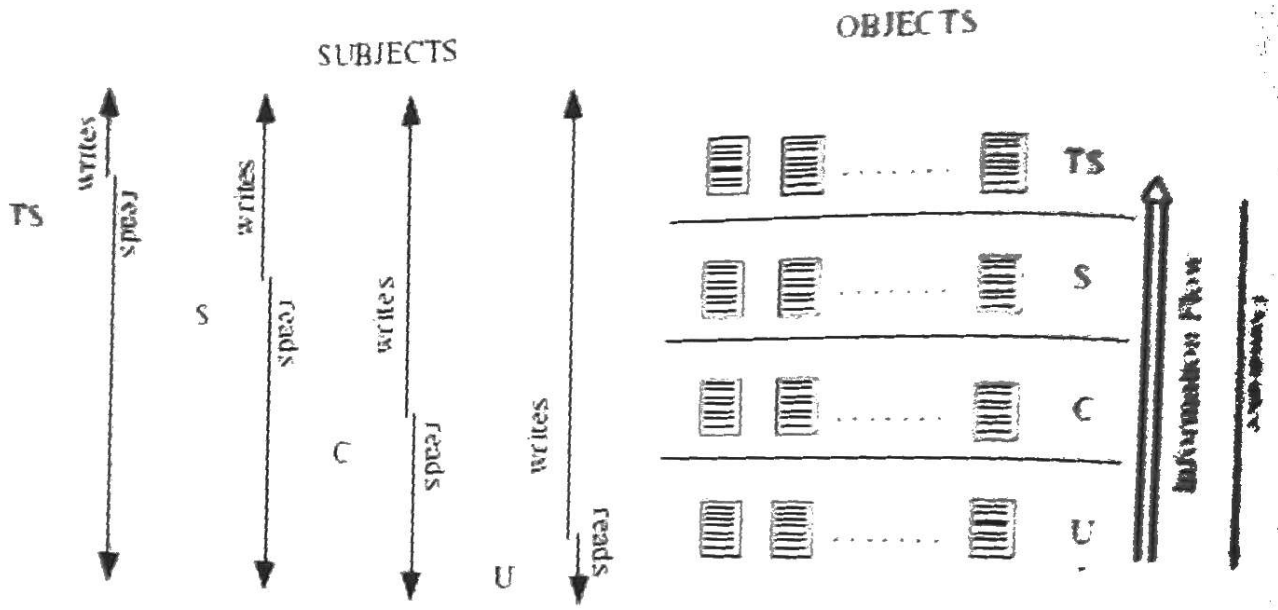
- **Control** direct and indirect **flows of information**
- Prevent leakage to unauthorized subjects
- User can connect to the system with any access class dominated by their clearance



Bell – LaPadula Model: Two Principles

- To protect information confidentiality
  - **No-read-up**, a subject is allowed a read access to an object only if the access class of the subject dominate the access class of the object
  - **No-write-down**, a subject is allowed a write access to an object only if the access class of the subject is dominated by the access class of the object

No-read-up & No-write-down



- Can TS subject write to S object?
- Can S subject write to U object?
- How to apply to the Trojan Horse case?

Bell – LaPadula Model

- Two main properties of this model for a secure system are:
  - Simple security property
  - Star property
- **Simple security** means: a subject at a given security level may not read an object at a higher security level **(no read-up)**.
- **Star property** means: a subject at a given security level must not write to any object at a lower security level **(no write-down)**.

Bell – LaPadula Model

- This model guarantees secrecy by preventing unauthorized release of information.
- This model does not protect from unauthorized modification of information

Bell – LaPadula Model

- **Example:** Trojan Horse
  - A Secret subject (a running program) can read Secret and Unclassified data but cannot write Unclassified data.
  - Trojan Horse can copy information from Secret data items (EMPLOYEE) but cannot write to Unclassified data items (COPY-OF-EMPLOYEE).

The Biba Model

- A model due to Ken Biba which is often referred to as "Bell-LaPadula upside down."
- It deals with integrity alone and ignores confidentiality entirely.
- Each subject and object in the system is assigned an integrity classification
  - Crucial
  - Important
  - Unknown

Integrity Level

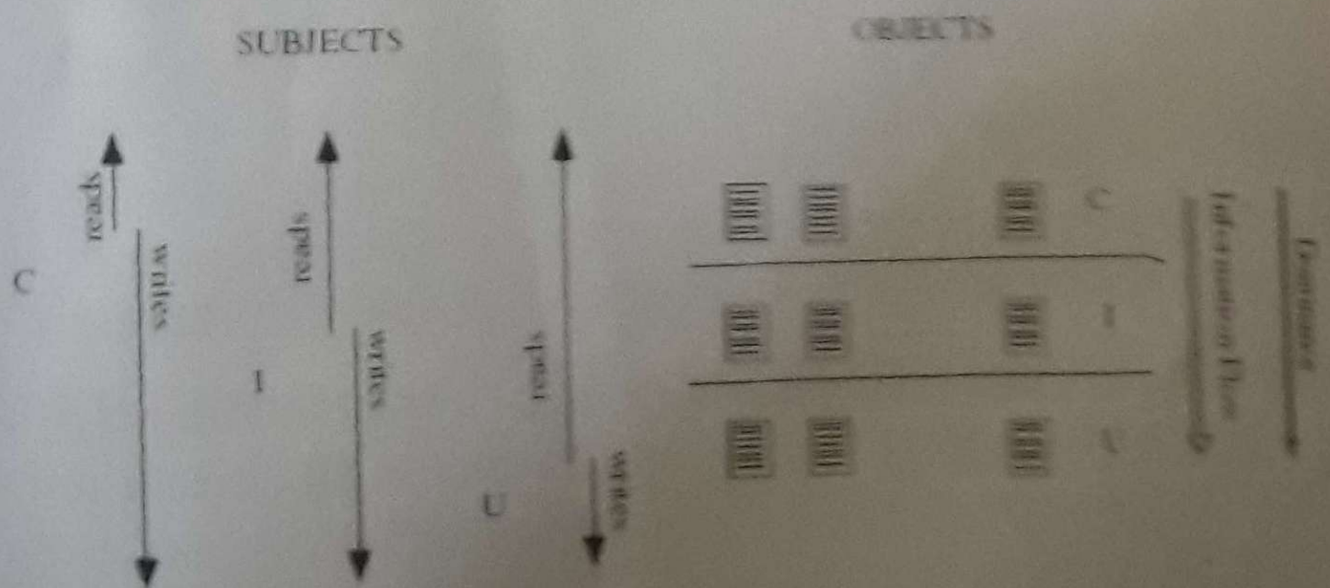
- Integrity level of a user reflects user's trustworthiness for inserting, modifying, or deleting information.
- Integrity level of an object reflects both the degree of trust that can be placed on the info stored in the object, and the potential damage could result from unauthorized modification of info

Two principles of Biba Model

- **No-read-down:** A subject is allowed a read access to an object only if the access class of the object dominates the access class of the subject.

- **No-write-up:** A subject is allowed a write access to an object only if the access class of the subject is dominated by the access class of the object

Two principles of Biba Model



Q: How to control both the secrecy and integrity?

Multilevel Security

MAC advantages :

- MAC provides tighter security because only a system administrator may access or alter controls.
- MAC policies reduce security errors.

Multilevel Security

❑ **Limitation of MAC**

- ❑MACs do not solve the Trojan Horse problem completely.
- ❑It is true that a program running at the Secret level is prevented from writing directly to Unclassified data item i.e. Trojan Horse table COPY-OF-EMPLOYEE
- ❑There are some other ways of communication to unclassified programs.

Multilevel Security

- Covert Channels
  - In a distributed networking environment, workstations and servers exchange data across WANs and LANs.
  - Covert Channel is a type of computer attack that steals information from the server to a workstation using Trojan horse in disguise of a valid data transfer from the server to the workstations.
  - It uses TCP/IP protocol.
  - It is a hidden attack and is not easily detectable.

Multilevel Security

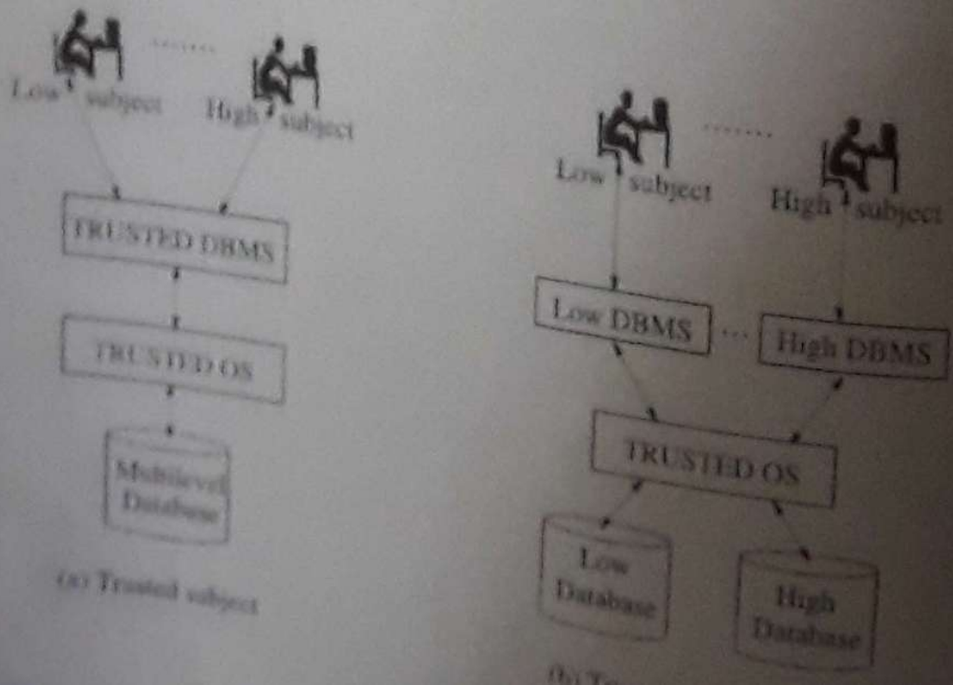
- Covert Channels
  - Exploitation may require the help of two Trojan horses. One runs at a high level and feeds high data into the channel, and the other Trojan horse runs at a lower level and reconstructs the high data for the malicious user from the signals received through the covert channel.
  - The low Trojan horse is not needed if the high one can send a straightforward signal that can be directly interpreted. Also, as we explain later, malicious users can exploit some special kinds of covert channels directly without using any Trojan horses at all.

Multilevel Security

- Covert Channels (Example)
  - A workstation is retrieving data from a DB server.
  - The Trojan horses at the DB server writes some ex bits of information in the data stream from Server workstation.
  - When the bits are coming to the workstation, the ext bits are isolated by another malicious program an reconstructs information by the Trojan Horse.

Multilevel DBMSs Architecture

- **Trusted subject:** The DBMS itself must be trusted to ensure mandatory policy
- **Trusted Computing Base:** Data are partitioned in different databases, one for each level



Reference

- Sushil Jajodia and Ravi S. Sandhu, "Toward a Multilevel Secure Relational Model", essay 20

Inference & Aggregation

- **Inference**
  - Inference is when you do multiple queries and you then have the ability to infer new information that normally you could not access directly. You infer new things from the data set you have consulted.
  - Users can draw inferences from the information they obtain from the database.
    - An unclassified user legitimately accesses unclassified information from which that user is able to deduce secrete information.
  - It presents a security breach if highly classified information can be inferred from less classified information.

Inference & Aggregation

Inference (Example):

- Imagine that you are the database administrator for a military transportation system.

- You have a table named cargo in your database that contains information on the various cargo holds available on each outbound airplane.

- Each row in the table represents a single shipment and lists the contents of that shipment and the flight identification number.

Inference & Aggregation

- The cargo table appears as follows:

Flight ID	Cargo Hold	Contents	Classification
1254	A	Boots	Unclassified
1254	B	Guns	Unclassified
1254	C	Atomic Bomb	Top Secret
1254	D	Butter	Unclassified

- **Suppose that General Jones (who has a Top Secret security clearance) comes along and requests information on the cargo carried by flight 1254. The general would (correctly) see all four shipments.**

Inference & Aggregation

- On the other hand, if Private Smith (who has no security clearance) requests the data, the private would see the following table:

Flight ID	Cargo Hold	Contents	Classification
1254	A	Boots	Unclassified
1254	B	Guns	Unclassified
1254	D	Butter	Unclassified

Inference & Aggregation

- Two important cases of the inference problem:
  - Aggregation problems occurs whenever there is a collection of data items that is classified at a higher level than the levels of individual data items.

Example:

- individual sales figure for branch offices might be less sensitive that the aggregate sales figure for the entire company.

Inference & Aggregation

- Two important cases of the inference problem:
  - Data association problem: occurs whenever two values seen together are classified at a higher level that the classification of either value individually.

Example:

- list consisting the names of all employees and the list containing all employee salaries are unclassified while a combined list giving employee names with their salaries is classified.

Inference Controls: Techniques

Four inference control techniques:

1) Appropriate Labeling
2) Query restriction
3) Polyinstantiation
4) Auditing

Inference Controls: Techniques

1. **Appropriate Labeling**

- If unclassified information x permits disclosure of secrete information y, reclassify all or part of information x such that it is no longer possible to derive y from the disclosed subset of x.
- Example: A + B <= 20;

  Where A = Unclassified Attribute, B = Classified Attribute.

  And the constraint is known to the unclassified user.

  Unclassified user can infer the values of secret information of B.

  So reclassify A as secret.

Inference Control: Techniques

2. **Query restriction**
   - Many inference violations arise as a result of a query which returns data at the user's level, but its evaluation requires accessing data above user's level.
   - An unclassified relation EP with attributes EMP-NAME & PRJ-NAME
   - A Secret relation PT with attributes PRJ-NAME & PRJ-TYPE
   - EMP-NAME & PRJ-NAME are the keys of 1st & 2nd relations

Inference Control: Techniques

2. **Query restriction(contd..)**

- An unclassified user makes the following query:

```
SELECT EP.PRJ-NAME FROM EP,PT
    WHERE EP.PRJ-NAME=PT.PRJ-NAME AND
        PT.PRJ-TYPE='NUCLEAR';
```

Secret information from unclassified data.

Query restriction ensures that all data used in the process of evaluating the query is dominated by the level of the user and therefore prevents such inference.

Inference Control: Techniques

3. **Polyinstantiation**

- Another technique to prevent inference violation.
- Polyinstantiation is a database technique that allows the database to contain multiple instances of the same data but with different classifications.
- Polyinstantiation occurs because of mandatory policy.
- In relational DBMS it is possible to have different tuples with the same key but with different classifications.

Inference Control: Techniques

3. **Polyinstantiation(contd..)**

- Polyinstantiation can affect relations, tuples and data elements.
- Polyinstantiation arises because subjects with different classes are allowed to operate on the same relations.

Inference Control: Techniques

3. **Polyinstantiation(contd..)**

Refer (**Polyinstantiation problem. pdf**)

Inference Control: Techniques

3. Polyinstantiation(contd..)

- Suppose an Unclassified user wants to enter a row in a relation in which each row is labeled as S (secret) or U (Unclassified).
- If the same key is already occurring in an S row, we cannot prevent the Unclassified user from inserting the U row without leaking of 1 bit of information by inference.

Inference Control: Techniques

3. Polyinstantiation (contd..)

STARTSHIP	DESTINATION	CLASS
Enterprise	Rigel	S
Enterprise	Mars	U

- Above Starship_Destination(SD) relation has the key STARSHIP, CLASS
- A secret user inserts the first row & later an unclassified user 2nd row.
- Unclassified user sees only one row and Secret user sees two rows.

Inference Control: Techniques

3. **Polyinstantiation(contd..)**
   - ❑Two rows may be interpreted in two different ways:
     - ❑There are two distinct STARSHIP name Enterprise going to two distinct destinations.
       - ❑Unclassified users know only of one. Secret users know both.
     - ❑There is a single STARSHIP named Enterprise.
       - ❑Its real destination is Regal which is known to Secret users.
       - ❑There is an unclassified cover story claiming that the destination is Mars.
       - ❑Doubtless, Secret users know which interpretation is intended.

Inference Control: Techniques

4. **Auditing**
   - ❑Can be used to control inferences.
   - ❑History can be kept of all queries that can be analyzed later to determine the proof of inference.
   - ❑If violation arises, query may be aborted.
   - ❑disadvantages:
     - ❑Too cumbersome
     - ❑Detects very limited types of inferences.

Integrity Principle & Mechanism

- Integrity
  - Concerned with improper modification of information.
  - Example:
    - Insertion of new information
    - Deletion of existing information
    - Changes of existing information
  - Authorization is only one piece of solution.
  - Integrity breaches can occur without authorization violation.
  - We must deal with the malicious user who exercises his/her authority improperly.

Integrity Principle & Mechanism

- Insider threat
  - A corrupt user can leak secrets by
    1) using the computer to legitimately access confidential information
    2) passing this information to an improper destination using telephone.
  - It is impossible for the computer to know whether step 1) was followed by step 2).

Integrity Principle & Mechanism

- Insider threat
  - A corrupt user can compromise integrity by
    1. Manipulating stored data or
    2. Falsifying source or output documents.
  - A computer can do little to solve the problem of false source or output documents.

Integrity Principle & Mechanism

- Insider threat
  - Military & Govt. sectors have established elaborate procedure for secrecy purpose whereas commercial sector is more informal.
  - We must rely on traditional techniques of paper-based manual systems.

Integrity Principles

- Seven integrity principles
  1. Well-formed transactions: Users should not manipulate data arbitrarily.
  2. Least Privilege: only necessary privilege.
  3. Separation of duties: Checker/maker
  4. Reconstruction of Events: Users are accountable for their actions. The ability to reconstruct what happened in the system may deter the users to violate integrity.

Integrity Principles

- Seven integrity principles
  5. Delegation of Authority: The procedure to delegate privilege must reflect the structure of the organization.
  6. Reality Checks: Cross-checks with external reality.
     - Physical inventory checks with the records in the database.
  7. Continuity of Operation: System operation should be maintained to some degree in the face of disaster which is beyond organization's control.

Integrity mechanisms

- Well-formed transactions

  Properties of a Transaction:

  - Correct state Transform: Consistent state before and after the transaction.
  - Serializability: net effect of executing a set of transactions is equivalent to executing them in some sequential order, even though they may actually be executed concurrently
  - Failure Atomicity: Either all or none updates of a transaction take effect.
  - Progress: No indefinite blocking.

Integrity mechanisms

- Well-formed transactions
  - Consistency constraints of relational database model
    - Entity integrity: stipulates that attributes in the primary key of a relation cannot have NULL values. This amounts to requiring that each entity represented in the database must be uniquely identifiable.
    - Referential integrity: references from one entity to another by foreign key. It requires that a foreign key either be all NULL or a matching tuple exists in the latter relation.

Integrity mechanisms

- Referential integrity: Referential integrity is a database concept that ensures that relationships between tables remain consistent.

  When one table has a foreign key to another table, the concept of referential integrity states that you may not add a record to the table that contains the foreign key unless there is a corresponding record in the linked table

Integrity mechanisms

- Least Privilege:
  - Fine-grained access control.
  - Read access: using views or query modification.
    - Extremely flexible and can be fine grained as desired.
  - Control of updates: neither one (view or query modification) provides the same flexibility.
- Cannot translate updates on views into update of base relations.
- As a result, authorization to control updates is often less sophisticated than authorization for read access.

Integrity mechanisms

- Least Privilege (Example: read access):

EMP	DEPT
Smith	Toy
Jones	Toy
Adams	Candy

DEPT	MANAGER
Toy	Brown
Candy	Baker

EMP	MANAGER
Smith	Brown
Jones	Brown
Adams	Baker

```
CREATE VIEW EMP-MANAGER AS

SELECT EMP,MANAGER FROM
EMP-DEPT a, DEPT-MANAGER b
WHERE a.DEPT=b.DEPT
```

Integrity mechanisms

- Least Privilege (Example: update control):

EMP	DEPT
Smith	Toy
Jones	Toy
Adams	Candy

DEPT	MANAGER
Toy	Brown
Candy	Baker

EMP	MANAGER
Smith	Brown (Green)
Jones	Brown (Green)
Adams	Baker

```
UPDATE VIEW EMP-MANAGER

SET MANAGER = 'Green'

WHERE EMP='Smith'
```

Integrity mechanisms

- Separation of duties (No concrete solution):
  - Issuance of checks
  - Entry of check information for withdrawal
  - Authorization by supervisor.

Integrity mechanisms

- Reconstruction of events
  - Audit trail of every transactions/activities
  - Event journal
  - It may deter improper behavior of the user.

Integrity mechanisms

- Delegation of Authority
  - A branch of a bank may have three business streams
    - General Banking
    - Credit
    - Trade Finance
  - Manager should have overall authority
  - Sectional heads should have authority only on their section.
  - SQL GRANT & REVOKE statement are not adequate.
  - Internal controls and checks should be maintained daily.

Integrity mechanisms

- Reality Checks
  - Requires activity outside of the DBMS.
  - External inspection to be conducted on an ad hoc on-demand basis.

Integrity mechanisms

- Continuity of Operation (disaster recovery)
  - Redundancy in various forms
    - DC/DRS concept
    - Application Clustering(RAC)
    - Virtualization
    - Recovery tools

Thank you