# Database Security

Professor Dr. Mohammad Abu Yousuf
yousuf@juniv.edu

#Lecture - 1

2019.

## Introduction

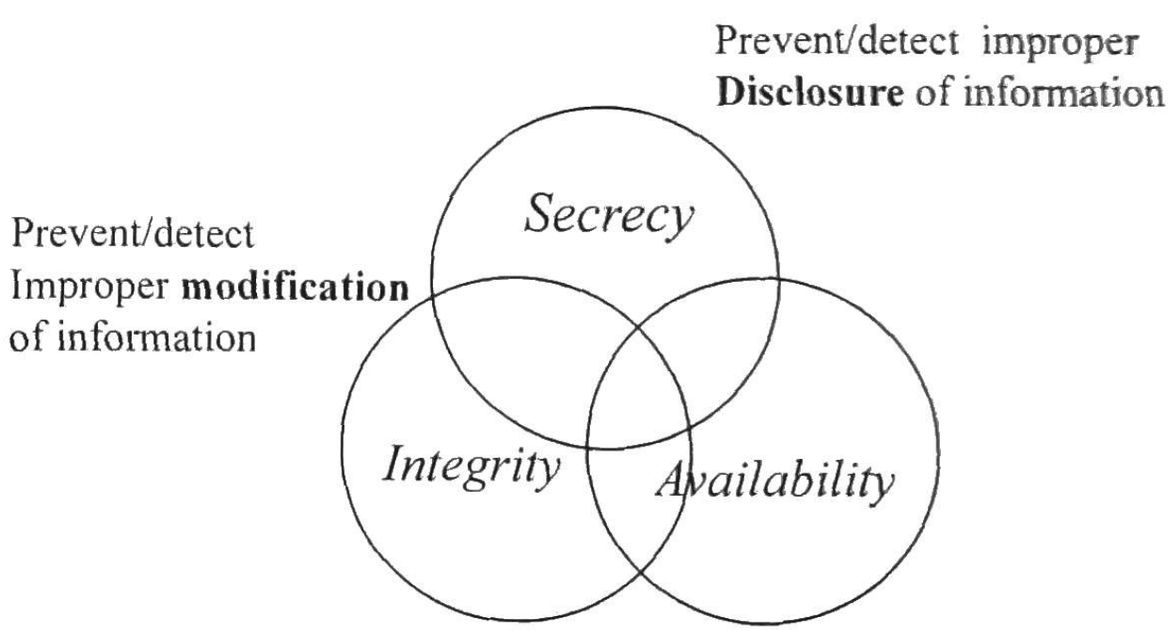
- Security violations and attacks are increasing globally at an annual average rate of **20%.**
- You serve as a database administrator to enforce security policies. Responsibilities can be:
  - Design and implement a new DB security policy.
  - Enforce a stringent security policy.
  - Implement functional specification of a module, i.e. encrypt the stored data, replace sensitive data using the data masking pack.

# Introduction

- **Security measures**
  - Prevent physical access to the servers where the data resided.
  - Operating systems require authentication of the identity of computer users.
  - Implement security models that enforce security measures.
- **DBA should manage databases and implement security policies to protect the data (assets).**



# Security Objectives

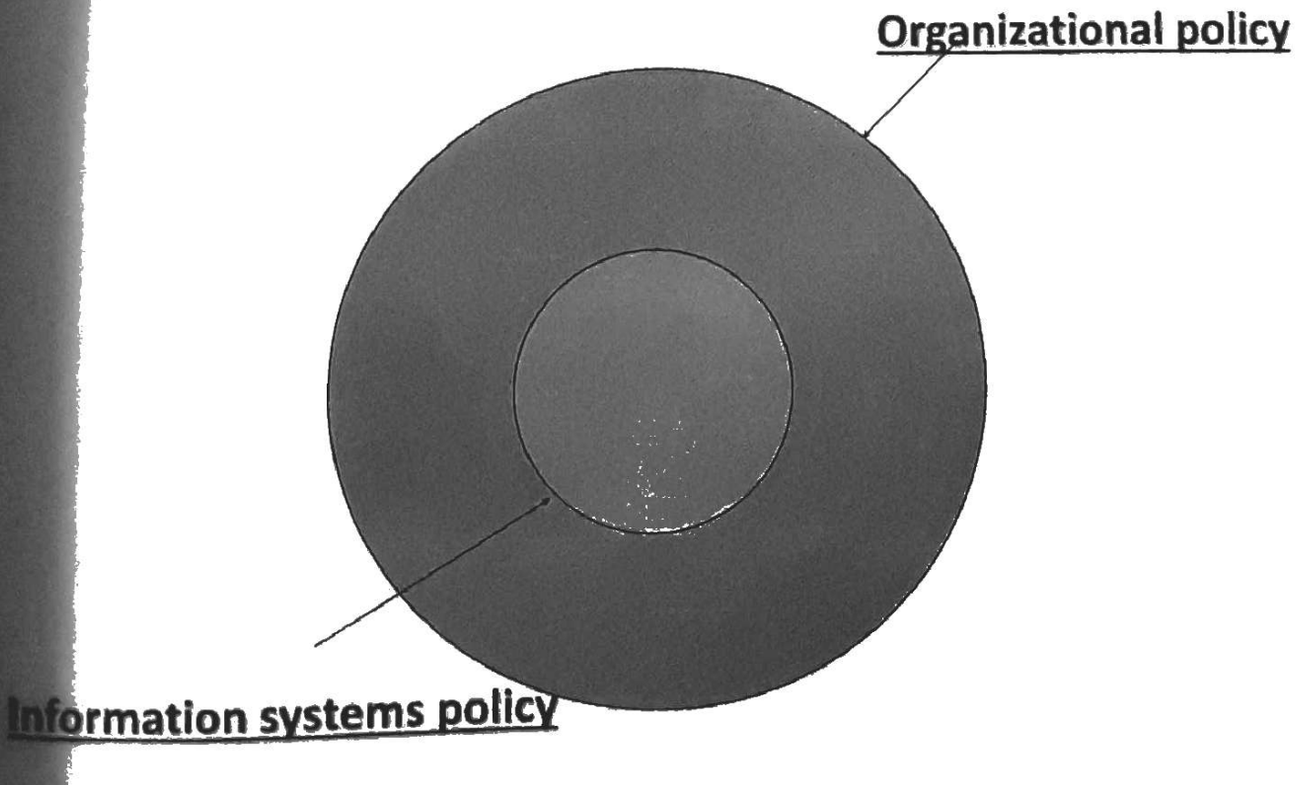



A denial-of-service attack is a security event that occurs when an attacker takes action that prevents legitimate users from accessing targeted computer systems, devices or other network resources.

# Security

- **Database security:** degree to which data is fully protected from tampering or unauthorized acts.
- Comprises information system and information security concepts

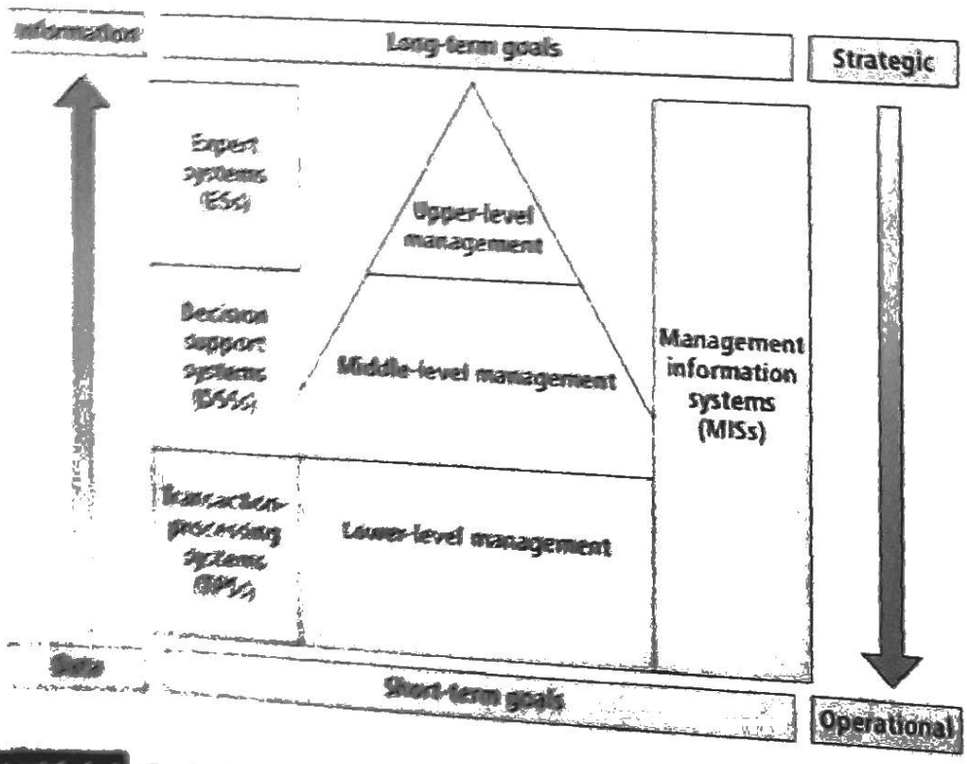
# Policy



# Information Systems

- Wise decisions require:
  - Accurate and timely information
  - Information integrity
- Information system: comprised of components working together to produce and generate accurate information
- Categorized based on usage: low-level, mid-level and high-level

# Information Systems (continued)




Typical use of system applications at various management levels

# Information Systems (continued)

**TABLE 1-1** Characteristics of information system categories

| Category | Acronym | Characteristics | Typical Application System |
|---|---|---|---|
| Transaction-processing system | TPS | ▪ Also known as online transaction processing (OLTP)<br>▪ Used for operational tasks<br>▪ Provides solutions for structured problems<br>▪ Includes business transactions<br>▪ Logical component of TPS applications (derived from business procedures, business rules, and policies) | ▪ Order tracking<br>▪ Customer service<br>▪ Payroll<br>▪ Accounting<br>▪ Student registration<br>▪ Car sales |



# Information Systems (continued)

**TABLE 1-1** Characteristics of information system categories (continued)

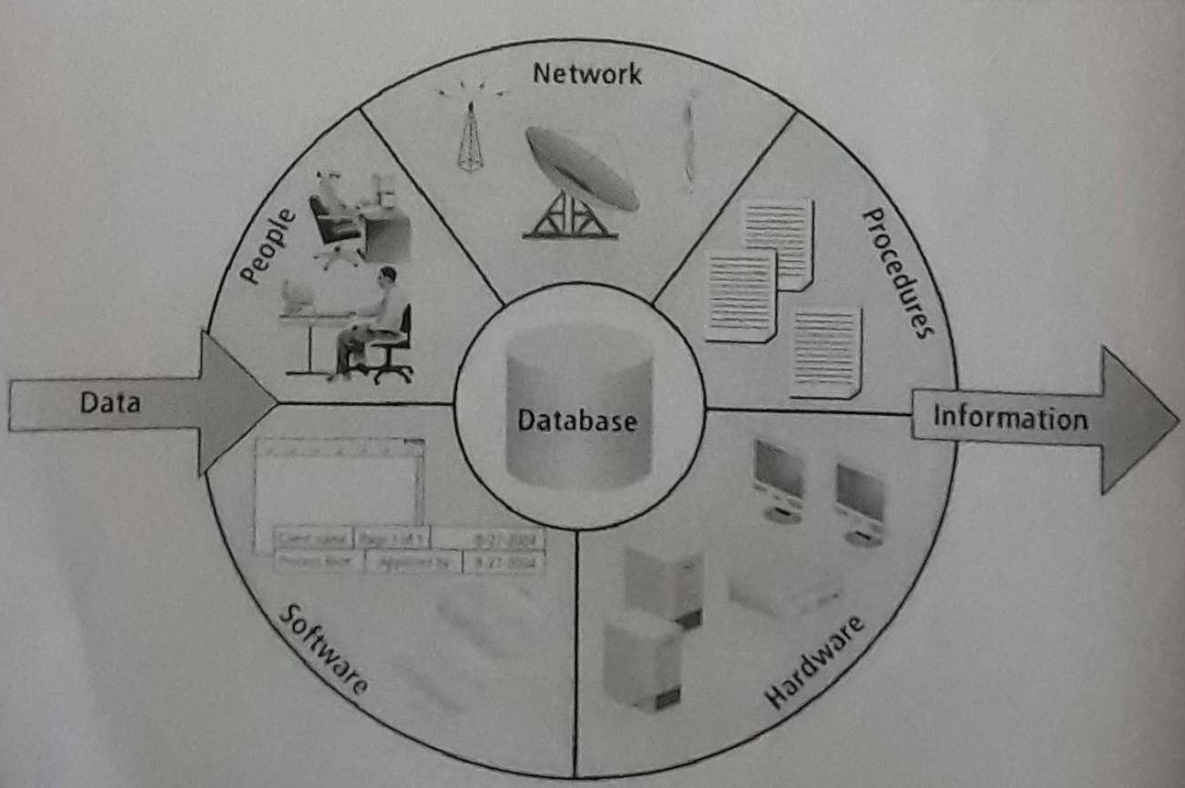
| Category | Acronym | Characteristics | Typical Application System |
|---|---|---|---|
| Decision support system | DSS | ▪ Deals with nonstructured problems and provide recommendations or answers to solve these problems<br>▪ Is capable of performing "What-if?" analysis<br>▪ Contains a collection of business models<br>▪ Is used for tactical management tasks | ▪ Risk management<br>▪ Fraud detection<br>▪ Sales forecasting<br>▪ Case resolution |
| Expert system | ES | ▪ Captures reasoning of human experts<br>▪ Executive expert systems (ESSs) are a type of expert system used by top-level management for strategic management goals<br>▪ A branch of artificial intelligence within the field of computer science studies<br>▪ Software consists of:<br>  ▪ Knowledge base<br>  ▪ Inference engine<br>  ▪ Rules<br>▪ People consist of:<br>  ▪ Domain experts<br>  ▪ Knowledge engineers | ▪ Virtual university simulation<br>▪ Financial enterprise<br>▪ Statistical trading<br>▪ Loan expert<br>▪ Market analysis |

# Information Systems (continued)

- Information system components include:
  - Data
  - Procedures
  - Hardware
  - Software
  - Network
  - People



# Information Systems (continued)



## Databases

- Collection of
  - Interrelated data and
  - set of programs to access the data
- Convenient and efficient processing of data

## Database Management

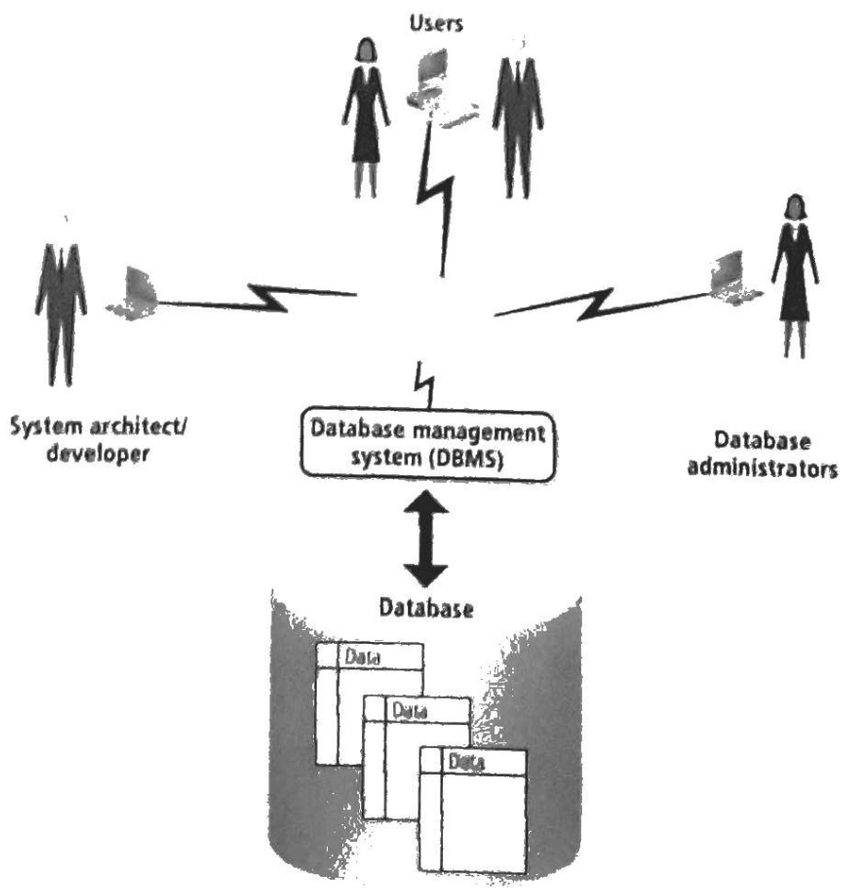
- Essential to success of information system
- DBMS functionalities:
  - Organize data
  - Store and retrieve data efficiently
  - Manipulate data (update and delete)
  - Enforce referential integrity and consistency
  - Enforce and implement data security policies and procedures
  - Back up, recover, and restore data

# Database Management (continued)

- DBMS components include:
  - Data
  - Hardware
  - Software
  - Networks
  - Procedures
  - Database servers

15

# Database Management (continued)




16

FIGURE 1-4 Database and DBMS environment

# Basic Security Concepts

The objectives of data security can be divided into:

- Confidentiality or Secrecy of data
- Integrity of data
- Availability of data

* What is CIA triad?

## Confidentiality (Secrecy)

- Need to ensure that confidential data is only available to correct people.
- Need to ensure that entire database is secured from external and internal system breaches.
- Need to provide for reporting on who has accessed what data and what they have done with it.
- Protected through the use of authentication **and** access control.

# Confidentiality (continued)

- **Addresses two aspects of security:**
  - Prevention of unauthorized access
  - Information disclosure based on classification
- **Classify company information into levels:**
  - Each level has its own security measures
  - Usually based on degree of confidentiality necessary to protect information

# Confidentiality (continued)

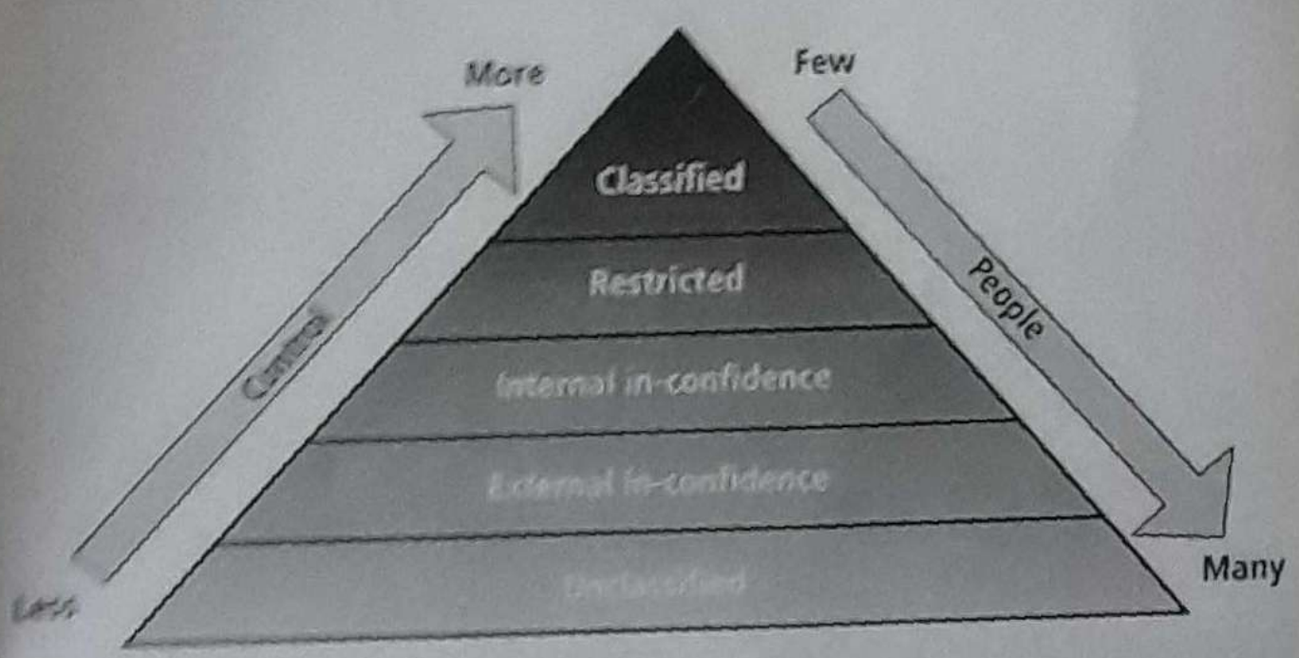



**FIGURE 1-6** Confidentiality classification

## Privacy vs. Confidentiality

- Privacy is sometimes mixed with confidentiality but there is a difference.
  - Confidentiality: withholding data access.
  - Privacy: Obtain and record the consents of users to disclose information. The data should be used only for the purpose sanctioned by the users and not misused for other purposes.

## Integrity

- refers to the prevention of unauthorized and improper data modification.
- Refers to the reliability, accuracy and consistency of the data stored within and retrieve from the database.
- Protected by preventing both unauthorized and authorized modifications whether accidental or deliberate, that might cause the database storage or retrieval to be unreliable and inconsistent.

# Integrity (continued)

**TABLE 1-2** Degradation of data integrity

| Type of Data Degradation | Description | Reasons for Data Losing Integrity |
|---|---|---|
| Invalid data | Indicates that not all the entered and stored data is valid without exception; checks and validation processes (known as database constraints) that prevent invalid data are missing. | ▪ User enters invalid data mistakenly or intentionally.<br>▪ Application code does not validate inputted data. |
| Redundant data | Occurs when the same data is recorded and stored in several places; this can lead to data inconsistency and data anomalies. | ▪ Faulty data design that does not conform to the data normalization process. (Normalization is a database design process used to reduce and prevent data anomalies and inconsistencies.) |



# Integrity (continued)

**TABLE 1-2** Degradation of data integrity

| Type of Data Degradation | Description | Reasons for Data Losing Integrity |
|---|---|---|
| Inconsistent data | Occurs when redundant data, which resides in several places, is not identical. | ▪ Faulty database design that does not conform to the data normalization process. |
| Data anomalies | Exists when there is redundant data caused by unnormalized data design; in this case, data anomalies occur when one occurrence of the repeated data is changed and the other occurrences are not. | ▪ Faulty data design that does not conform to the data normalization process. |

## Availability

- Data needs to be available at all necessary times
- Data needs to be available to only the appropriate users
- Need to be able to track who has access to and who has accessed what data

## Availability

- Identify those things that pose a threat to the availability of the database.
- Assess the level of threat and plan an intervention.
- Common potential threats:
  - Hardware failure
  - Software failure
  - Disaster
  - Intrusion etc.

## Examples of Security concern

- Database of a Payroll system.
  - Secrecy: Salaries should not be disclosed.
  - Integrity: Preventing an employee to change his/her salary.
  - Availability: Timely salary.

## Example of Security concern

- Database of a Web site of an airline company
  - Secrecy: Customer reservation to be available to the customers only.
  - Integrity: It should not be arbitrarily modified.
  - Availability: Flight information and reservation information to be available always.

## Secrecy, Integrity, Availability

- They co-exists in every information system.
- Relative importance differs from system to system
- High integrity is needed for both military and commercial sectors.
- Secrecy and availability requirements are more stringent in military sector than that of commercial sectors.

## Security Policy

- Purpose is to elaborate the three generic security objectives in the context of a particular system.
- It defines the guidelines to achieve the three security objectives
- Law of the land may be mentioned for breach of security rules by an individual.
- It should be determined by the organization itself rather than by regulatory agency.

# Database Security

**Database Security:**

- A set of established procedure, standard, policies and tools that is used to protect data from theft, misuse and unwanted intrusion, activities and attacks.
- It deals with the permission and access to the data structure and the data contained within it.



# Database Security (Continued)

- Enforce security at all database levels.
- **Security access point:** place where database security must be protected and applied.
- Data requires highest level of protection; data access point must be small

# Database Security (Continued)

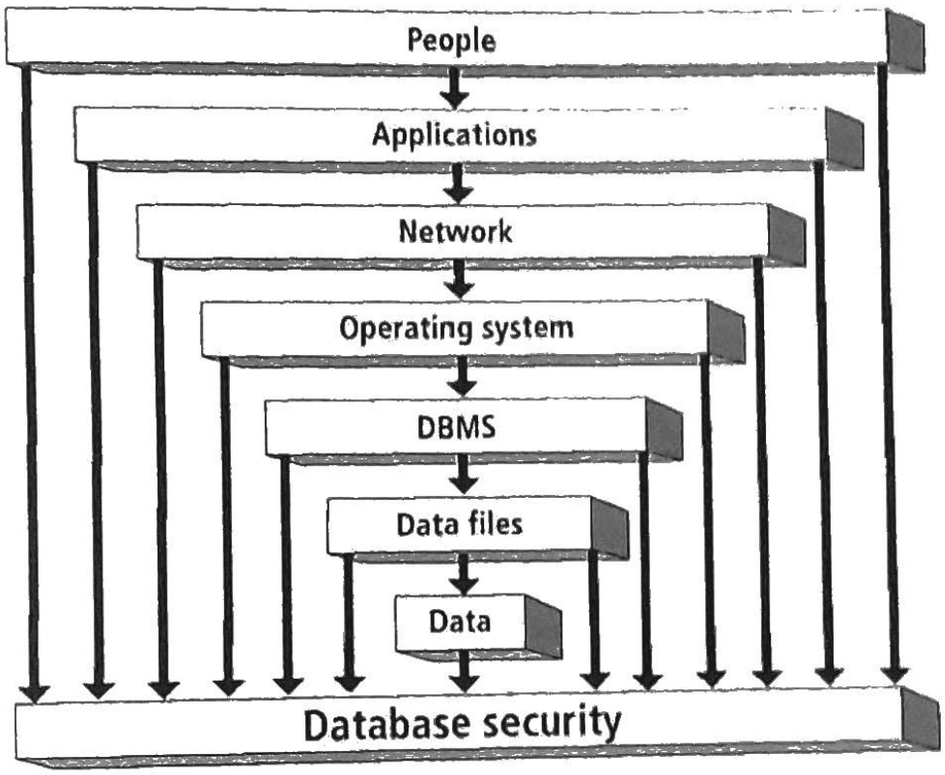





**FIGURE 1-8** Database security access points

# Database Security (Continued)

- Reducing access point size reduces security risks
- Security gaps: points at which security is missing.
- Vulnerabilities: twists in the system that can become threats.
- Threat: security risk that can become a system breach

# Database Security (continued)

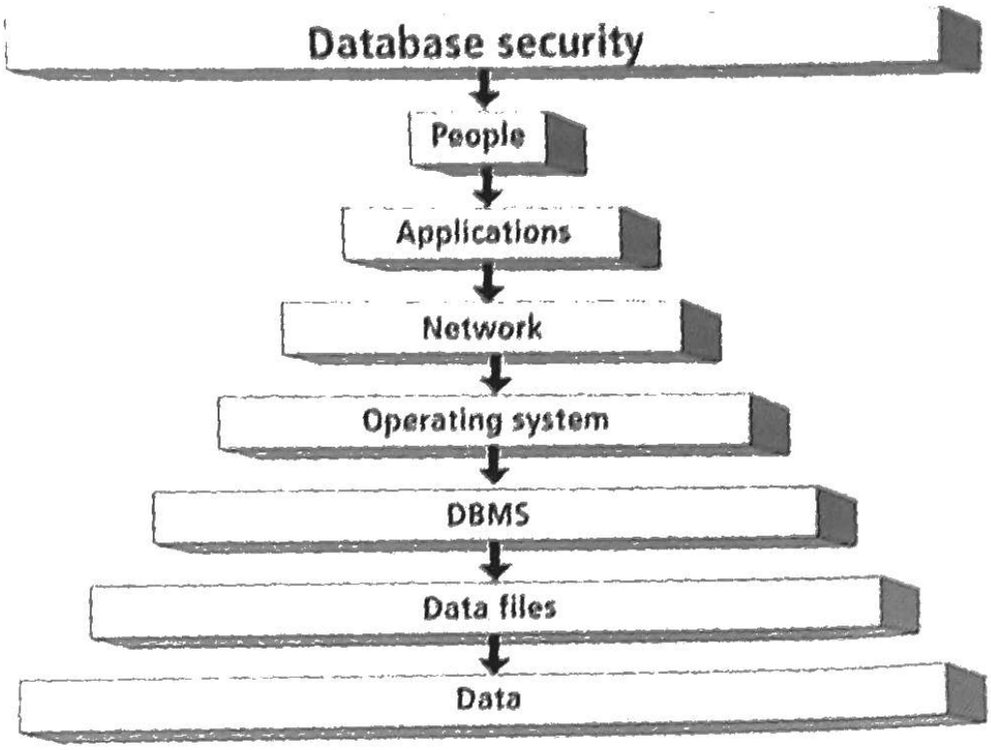



**FIGURE 1-9** Database security enforcement



# Database Security (continued)

- People: individuals who have been granted privileges and permissions to access applications, networks, servers, databases, data files and data.
- Applications: application design and implementation, which includes privileges and permissions granted to people. Be cautious because too loose permission results in violation of data access, and too strict permission compromises availability.
- Network is the most sensitive security access point. Use best effort to protect the network.

# Database Security (continued)

- Operating system: the authentication to the system and the gateway to the data.
- DBMS: logical structure of the database, include memory, executables, and other binaries.
- Data files: to be protected through the use of permissions and encryption.
- Data: need to enforce data integrity, and necessary privileges.



# Database Security (continued)

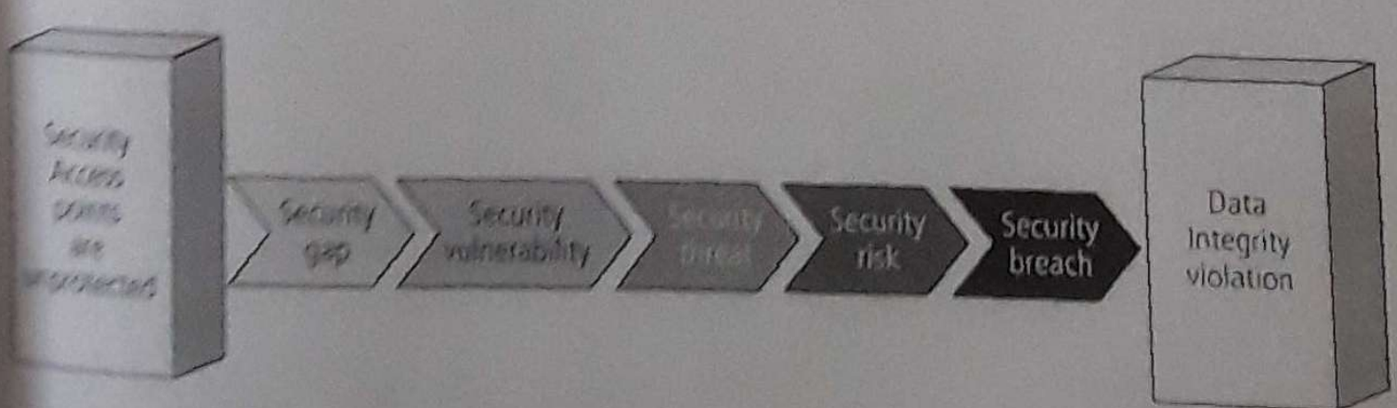


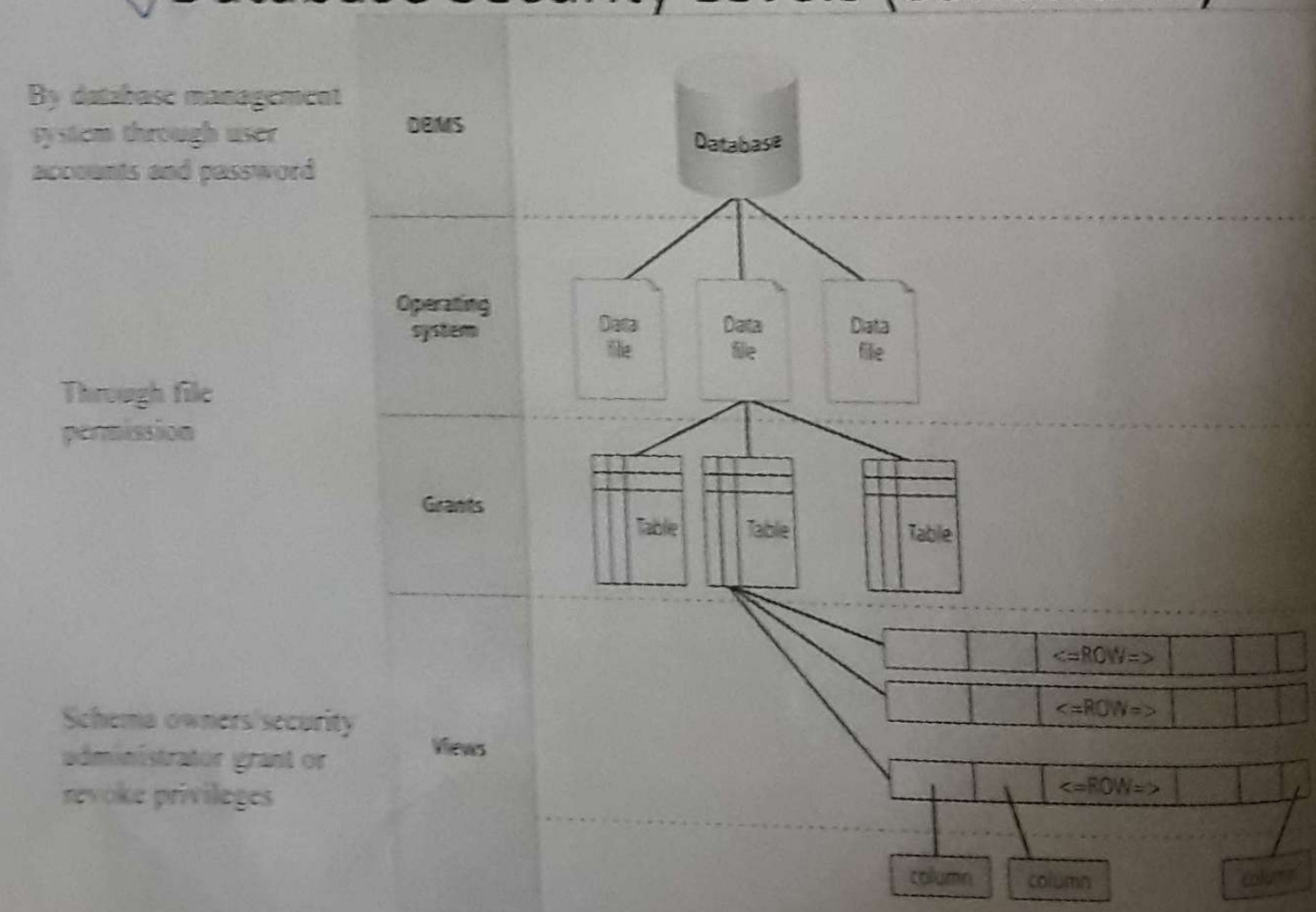

FIGURE 1-10 Data integrity violation process

# Database Security Levels

- Relational database: collection of related data files
- Data file: collection of related tables
- Table: collection of related rows (records)
- Row: collection of related columns (fields)

39

# Database Security Levels (continued)




40

**FIGURE 1-11** Levels of database security

# Threats to Databases

- Security vulnerability: a weakness in any information system component

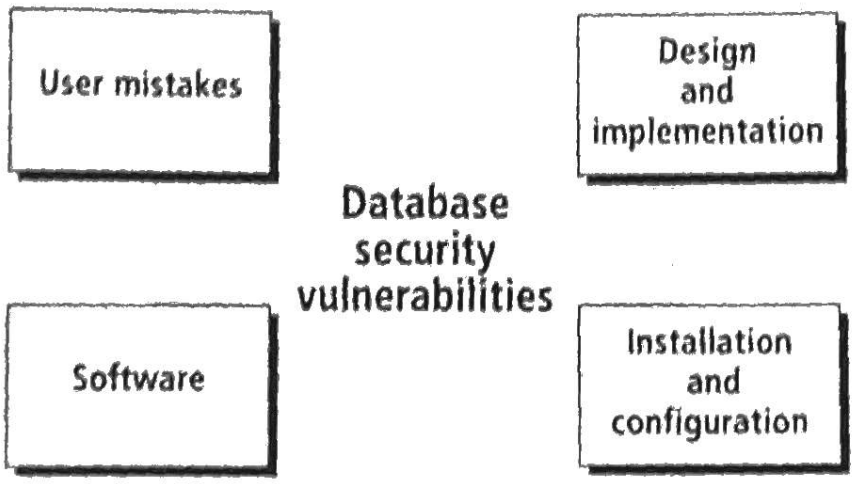



**FIGURE 1-12** Categories of database security vulnerablilities



# Threats to Databases (Continued)

- Security threat: a security violation or attack that can happen any time because of a security vulnerability.

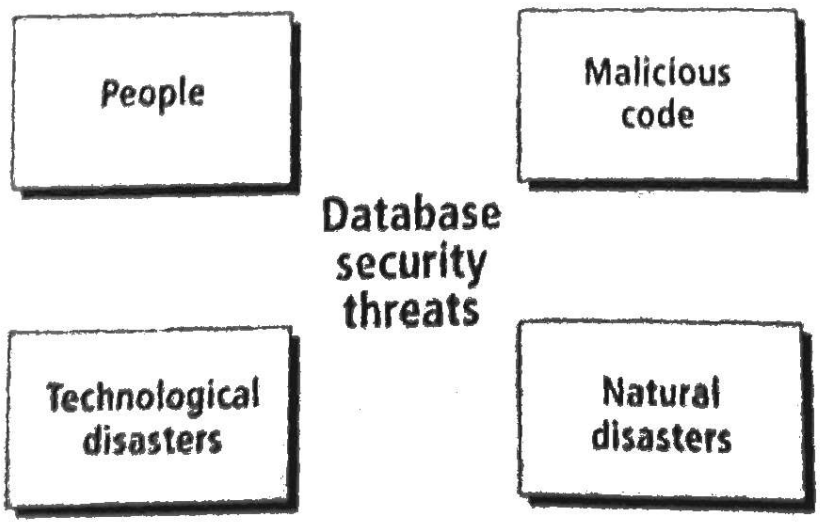



**FIGURE 1-13** Categories of database security threats

# Threats to Databases (Continued)

- Security risk: a known security gap left open.

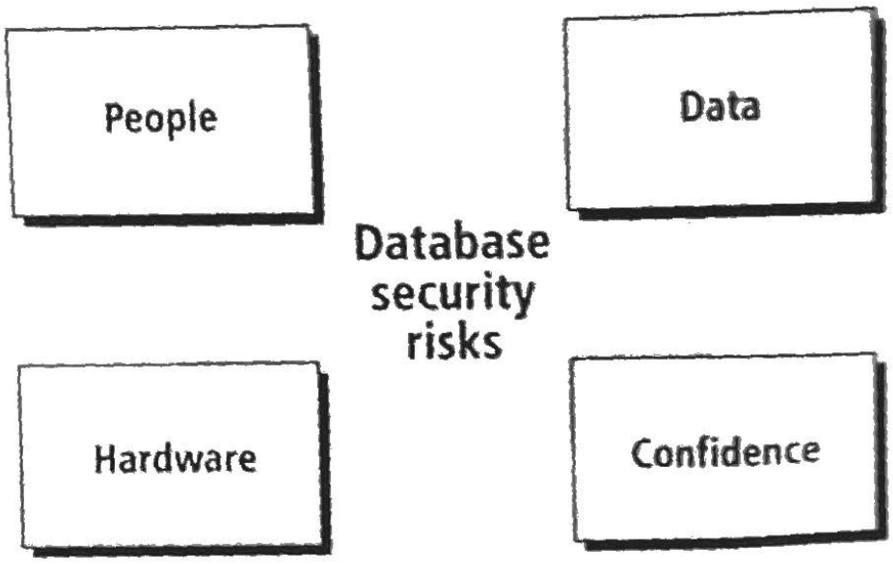



**FIGURE 1-14** Categories of database security risks



# Threats to Databases (Continued)

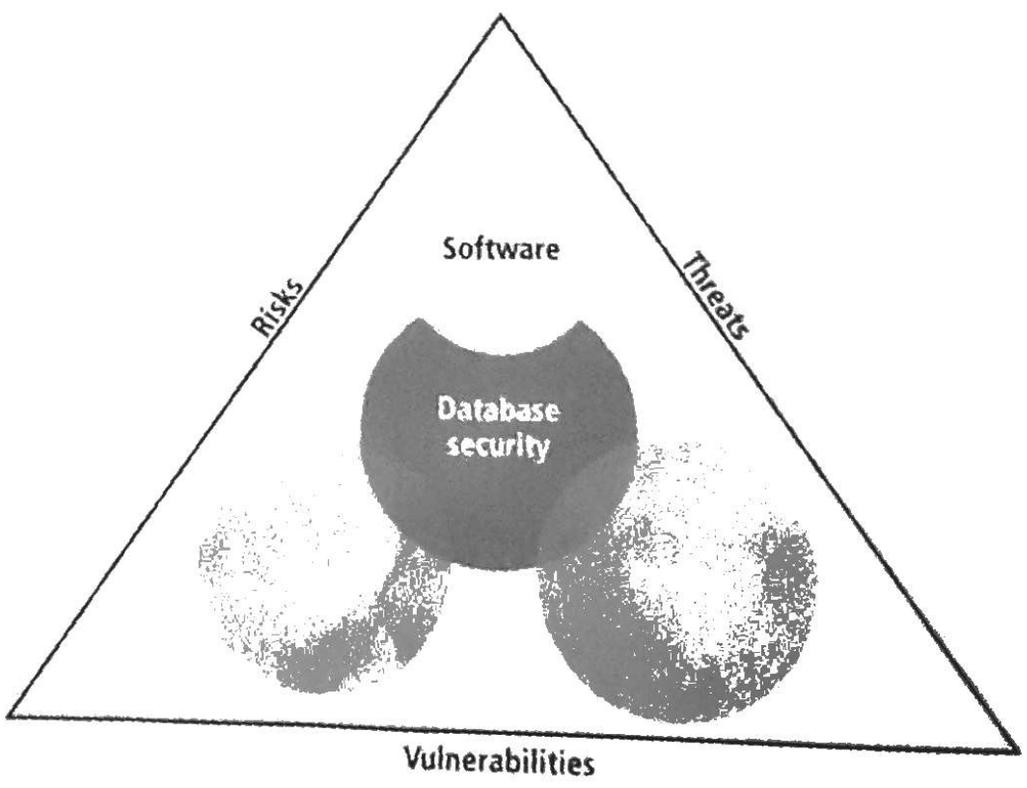



**FIGURE 1-15** Integration of security vulnerabilities, threats, and risks in a database environment

# Prevention, Detection & Tolerance

- The objective of data security can be approached in two distinct and mutually supportive, ways:
  - Prevention: ensures that security breaches can not occur. System examines every action and checks its with the security policy before allowing it to occur.
  - Detection: ensures that sufficient history of the activity in the system is recorded in an audit trail so that a security breach can be detected after the fact-auditing.

# Prevention, Detection & Tolerance

- Tolerance – a 3rd technique
  - In which the potential for some security breaches is tolerated because either these breaches are too expensive to prevent or likelihood of its occurrence is extremely low.
  - Every practical system tolerates some degree of risk to potential security breaches.

# Database Security Methodology

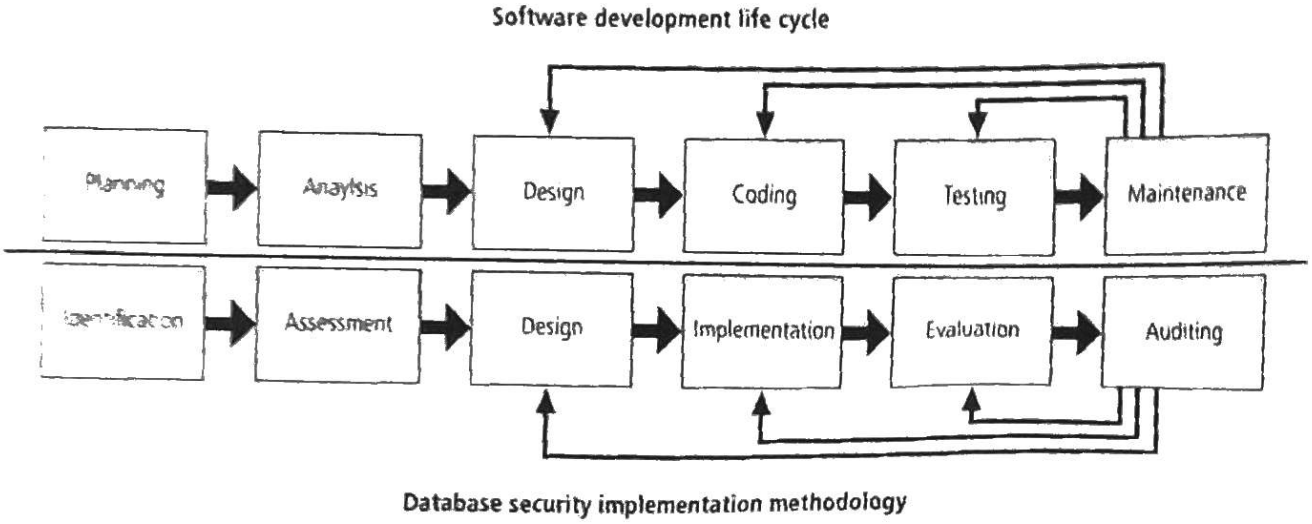



**FIGURE 1-16** Database security methodology



# Summary

- Security: level and degree of being free from danger and threats
- Database security: degree to which data is fully protected from unauthorized tampering
- Information systems: backbone of day-to-day company operations

# Summary (continued)

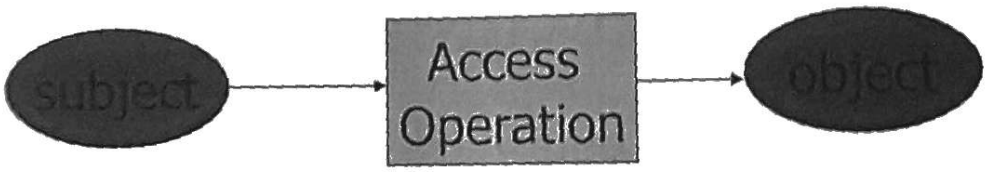
- DBMS: programs to manage a database
- C.I.A triangle:
  - Confidentiality
  - Integrity
  - Availability
- Secure access points
- Security vulnerabilities, threats and risks
- Information security architecture
  - Model for protecting logical and physical assets
  - Company's implementation of a C.I.A. triangle
- Enforce security at all levels of the database



# Access Controls in Current Systems

# Introduction

- "Access control" is where security engineering meets computer science.
- Its function is to control which (active) subject have access to a which (passive) object with some specific access operation.






# Introduction

- The purpose of access controls is to ensure that a user is only permitted to perform those operations on the database for which that user is authorized.

- Protects against accidental and malicious threats by regulating the read, write and execution of data and programs.

# Introduction

- **Authentication** typically requires the user to supply his or her claimed identity (e.g., user name, operator number, etc.) along with a password or some other authentication token.

- **Authentication** may be performed by the Operating System, the Database Management System, a special Authentication Server, or some combination thereof.

## Access Controls in Current Systems

## How to specify access control?

# How to specify access control?

- Access control components:
  - Access control policy: specifies the authorized accesses of a system
  - Access control mechanism: implements and enforces the policy

# How to specify access control?

Access Control Policies

- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)
- Role-Based Access Control (RBAC)

# Modes of Access Control

- AC can be imposed at various degrees of granularity in the system.
  - The entire Database.
  - Some Collection of relations(Tables)
  - One relation(Table).
  - Some columns of one relation
  - Some rows of one relation.
  - Some columns of some rows of one relation.

# Modes of Access Control

- Subject active entity that requests access to an object
  - e.g., user or program
- Object: passive entity accessed by a subject
  - e.g., record, relation, file
- Access right (privileges): how a subject is allowed to access an object
  - e.g., subject *s* can read object *o*

# Modes of Access Control

- Access Control Modes are expressed in terms of basic SQL statements:
  - SELECT can be specified by relation by relation basis. Finer granularity of authorization can be provided by views:
  - INSERT can be specified by relation by relation basis.
  - DELETE can be specified by relation by relation basis.
  - UPDATE can be restricted to certain columns of a relation.

# Discretionary Access Control

- Access to data objects (files, directories, etc.) is permitted based on the identity of users.

- Explicit access rules that establish who can, or cannot, execute which actions on which resources.

- Discretionary: users can be given the ability of passing on their privileges to other users, where **granting** and **revocation** of privileges is regulated by an administrative policy.

# Discretionary Access Control

- is a type of access control in which a user has complete control over all the programs(subject) it owns.
- Owner of the subject determines the access level of that subject.
- User Authentication: Username / Password /biometrics
- Authorized to perform specific operations on the Database.
- Managed by Granting / Revoking various kinds of privileges.

# Mandatory Access Control

- is a type of access control in which only the administrator manages the access controls.
- System determines the access level.
- The administrator defines the usage and access policy, which cannot be modified or changed by users, and the policy will indicate who has access to which programs and files.
- MAC is most often used in systems where priority is placed on confidentiality.

# Difference between DAC and MAC

- The main difference between them is in how they provide access to users.

- With MAC, admins creates a set of levels and each user is linked with a specific access level. He can access all the resources that are not greater than his access level.

- In contrast, each resource in DAC has a list of users who can access it. DAC provides access by identity of the user and not by permission level.

# Discretionary Access Control

- Discretionary Access Control (DAC)
  - Access Control Matrix Model
  - Implementation of the Access Matrix
  - Vulnerabilities of the Discretionary Policies
  - Additional features of DAC

# Discretionary Access Control

- Access control matrix
  - Describes protection state precisely
  - Matrix describing rights of subjects
  - State transitions change elements of matrix
- State of protection system
  - Describes current settings, values of system relevant to protection



# Discretionary Access Control

- Discretionary Access Control
  - Access Control Matrix Model
  - Implementation of the Access Matrix
  - Vulnerabilities of the Discretionary Policies
  - Additional features of DAC

# Access Control Matrix Model

- Access control matrix
  - Firstly identify the objects, subjects and actions.
  - Describes the protection state of a system.
  - State of the system is defined by a triple (S, O, A)
    - S is the set of subject,
    - O is the set of objects,
    - A is the access matrix
  - Elements indicate the access rights that subjects have on objects
    - Entry A[s, o] of access control matrix is the privilege of s on o



# Description

objects (entities)

subjects

| | $o_1$ | ... | $o_m$ | $s_1$ | ... | $s_n$ |
|---|---|---|---|---|---|---|
| $s_1$ | | | | | | |
| $s_2$ | | | | | | |
| ... | | | | | | |
| $s_n$ | | | | | | |

- Subjects $S = \{ s_1,...,s_n \}$
- Objects $O = \{ o_1,...,o_m \}$
- Rights $R = \{ r_1,...,r_k \}$
- Entries $A[s_i, o_j] \subseteq R$
- $A[s_i, o_j] = \{ r_x, ..., r_y \}$ means subject $s_i$ has rights $r_x, ..., r_y$ over object $o_j$

| | File 1 | File 2 | File 3 | Program 1 |
|---|---|---|---|---|
| Ann | own read write | read write | | execute |
| Bob | read | | read write | |
| Carl | | read | | execute read |

# Access Control

- Discretionary Access Control
  - Access Matrix Model
  - Implementation of the Access Control Matrix
  - Vulnerabilities of the Discretionary Policies
  - Additional features of DAC



# ACM Implementation

- ACM is an **abstract** model
  - Rights may vary depending on the object involved
- ACM is implemented primarily in three ways
  - Authorization Table
  - Capabilities (rows)
  - Access control lists (columns)

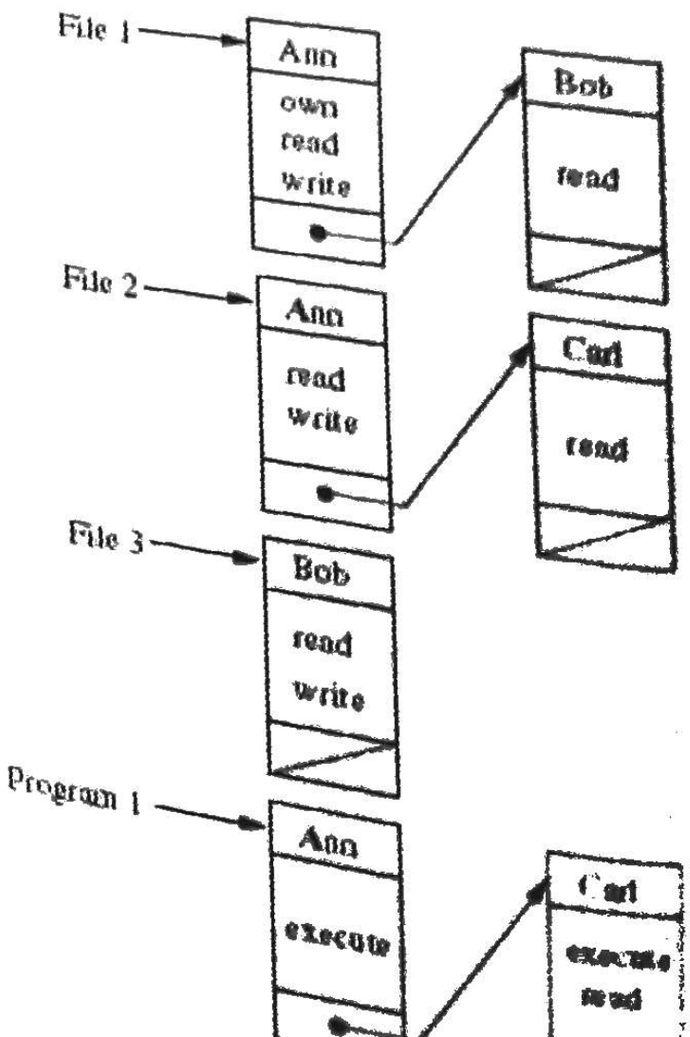
# Authorization Table

- Three columns: subjects, actions, objects
- Generally used in DBMS systems

| User | Access Mode | Object |
|---|---|---|
| Ann | own | File 1 |
| Ann | read | File 1 |
| Ann | write | File 1 |
| Ann | read | File 2 |
| Ann | write | File 2 |
| Ann | execute | Program 1 |
| Bob | read | File 1 |
| Bob | read | File 3 |
| Bob | write | File 3 |
| Carl | read | File 2 |
| Carl | execute | Program 1 |
| Carl | read | Program 1 |

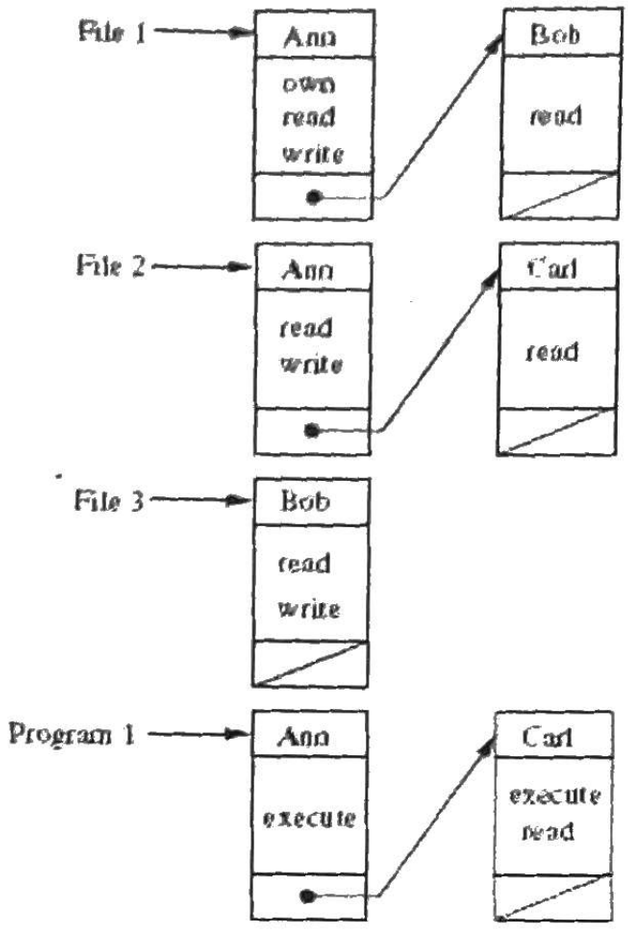
# Access Control List (ACL)

- Matrix is stored by column.
- Each object is associated with a list
- Indicate for each subject the actions that the subject can exercise on the object



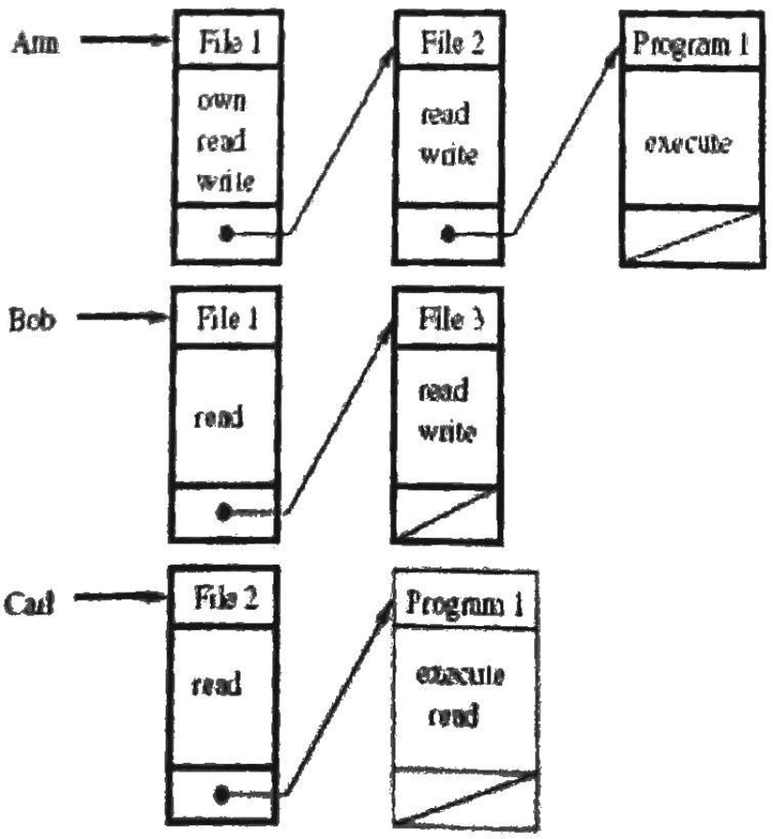
# Access Control List (ACL)

| USER | ACCESS MODE | OBJECT |
|---|---|---|
| Ann | own | File 1 |
| Ann | read | File 1 |
| Ann | write | File 1 |
| Ann | read | File 2 |
| Ann | write | File 2 |
| Ann | execute | Program |
| Bob | read | File 1 |
| Bob | read | File 3 |
| Bob | write | File 3 |
| Carl | read | File 2 |
| Carl | execute | Program |
| Carl | read | Program |



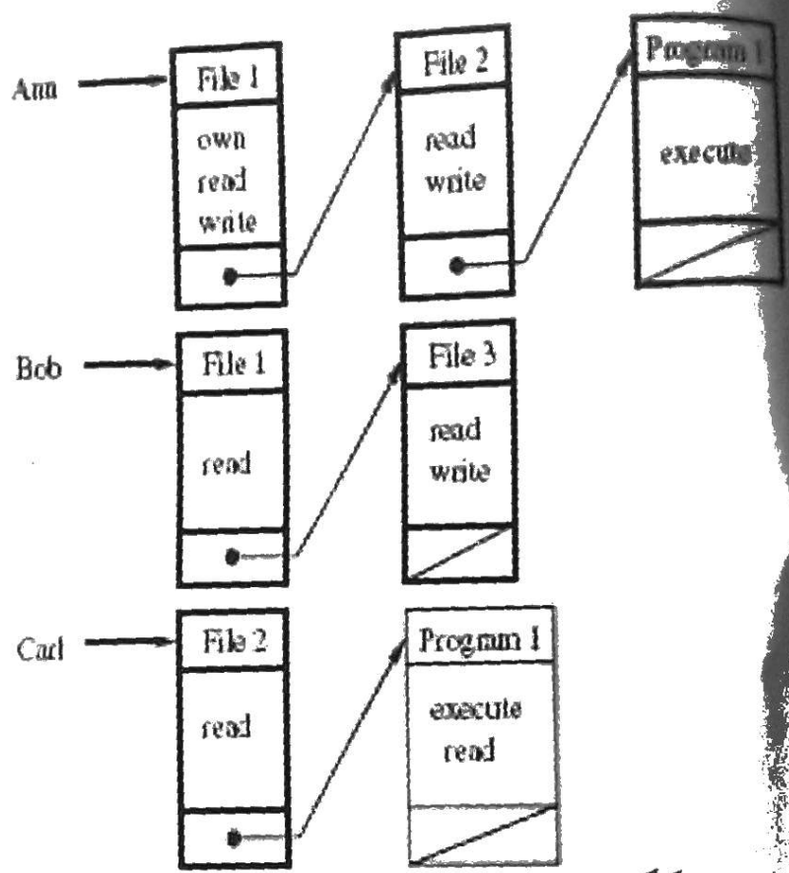

# Capability List

- Matrix is stored by row
- Each user is associated with a capability list
- Indicating for each object the access that the user is allow to exercise on the object



# Capability List

| User | Access Mode | Object |
|---|---|---|
| Ann | own | File 1 |
| Ann | read | File 1 |
| Ann | write | File 1 |
| Ann | read | File 2 |
| Ann | write | File 2 |
| Ann | execute | Program 1 |
| Bob | read | File 1 |
| Bob | read | File 3 |
| Bob | write | File 3 |
| Carl | read | File 2 |
| Carl | execute | Program 1 |
| Carl | read | Program 1 |




# ACLs vs Capability List

- Immediate to check the authorization holding on an object with ACLs. (subject?)
- Immediate to determine the privileges of a subject with Capability lists. (object?)
- Distributed system,
  - authenticate once, access various servers
  - choose which one?
- Limited number of groups of users, small bit vectors, authorization specified by owner.
  - Which one?

# Basic Operations in Access Control

- Grant permissions
  - Inserting values in the matrix's entries
- Revoke permissions
  - Remove values from the matrix's entries
- Check permissions
  - Verifying whether the entry related to a subject *s* and an object *o* contains a given access mode

# Access Control

## Discretionary Access Control

- Access Matrix Model
- State of Protection System
- Implementation of the Access Matrix
- Vulnerabilities of the Discretionary Policies
- Additional features of DAC

# Vulnerabilities of the Discretionary Policies

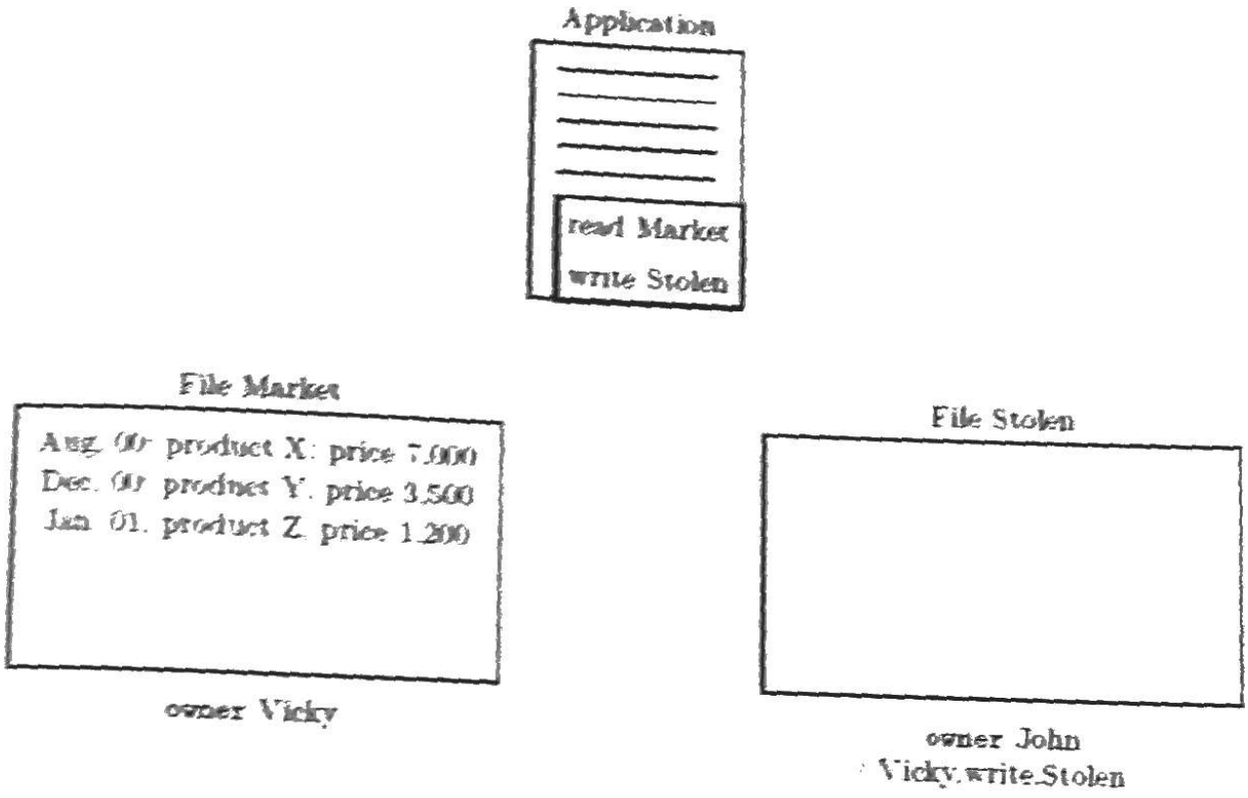
- No control on the flow the information
- Malicious code, i.e., Trojan horse



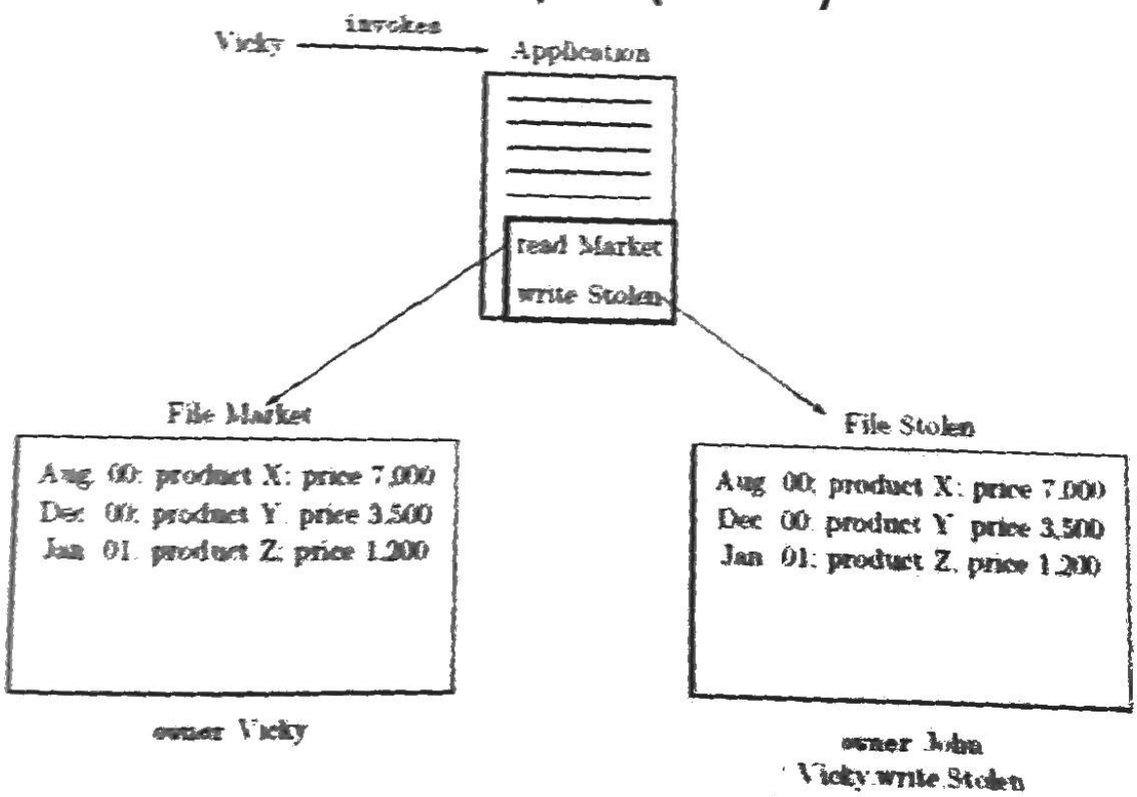
## Example

- Vicky, a top-level manager
- A file Market on the new products release
- John, subordinate of Vicky
- A file called "Stolen"
- An application with two hidden operations
  - Read operation on file Market
  - Write operation on file Stolen

# Example (cond)




# Example (cond)




- Restriction should be enforced on the operations that processes themselves can **execute**.
- Mandatory policies provide a way to enforce **information flow control** through the use of **labels**

# Access Control

- Discretionary Access Control
  - Access Matrix Model
  - State of Protection System
  - Implementation of the Access Matrix
  - Vulnerabilities of the Discretionary Policies

## DAC – additional features and recent trends

- Flexibility is enhanced by supporting different kinds of permissions
  - Positive vs. negative
  - Implicit vs. explicit
  - Content-based

## Positive and Negative Permissions

- Positive permissions → Give access
- Negative permissions → Deny access
- Useful to specify exceptions to a given policy and to enforce stricter control on particular crucial data items

## Implicit and Explicit Permissions

- Some models support implicit permissions
- Implicit permissions can be derived:
  - by a set of *propagation rules* exploiting the subject, object, and privilege hierarchies
  - by a set of user-defined *derivation rules*

# Derivation Rules: Example

- Ann can read file F1 from a table if Bob has an explicit denial for this access.
- Tom has on file F2 all the permissions that Bob has.
- Derivation rules are a way to concisely express a set of security requirements.

# Derivation Rules

- Derivation rules are often expressed according to logic programming
- Several research efforts have been carried out to compare the expressive power of such languages
- We need languages based on SQL and/or XML

# Content-based Permissions

- Content-based access control conditions the access to a given object based on its content
- This type of permissions are mainly relevant for database systems
- As an example, in a RDBMS supporting content-based access control **it is possible to authorize a subject to access information only of those employees whose salary is not greater than 30K**

# Content-based Permissions

- Two most common approaches to enforce content-based access control in a DBMS are done:
  - by associating a predicate (or a Boolean combination of predicates) with the permission
  - by defining a *view* which selects the objects whose content satisfies a given condition, and then granting the permission on the view instead of on the basic objects

# Discretionary Access Control

Data Dependent Access Control

- Database Access Controls are often data dependent.
- Some users may be allowed to see the salary up to Tk.20000
- A manager may be allowed to see the salary of the employees of his/her own department.
- **Three techniques**:
  - Security through Views
  - Query modification
  - Grant and Revoke

# Discretionary Access Control

- View based Access Controls
  - A view is a virtual relation (table)
  - The DB stores its definition and materializes the view as needed.
  - Useful mechanism for specifying data-dependent authorization for data retrieval.
  - Example: Employee Table
    - COLUMNS :

      EMPID, ENAME,SALARY,MANAGER, DEPTNO

# Discretionary Access Control

*View:*

- *Views* provide a valuable tool in enforcing security policies.

- A *view* is a table whose rows are not explicitly stored in the database but are computed as needed from a view definition.

# Discretionary Access Control

❑View based Access Controls (EMPLOYEE TABLE)

| EMPID | ENAME | SALARY | MANAGER | DEPTNO |
|---|---|---|---|---|
| 0001 | Smith | 10000 | Bretlee | Sales |
| 0002 | Jones | 12000 | Bretlee | Sales |
| 0003 | Baker | 15000 | Milton | Production |
| 0004 | Adams | 20000 | Bretlee | Sales |
| 0005 | Hares | 25000 | Milton | Production |
| 0006 | Milton | 70000 | NULL | Production |
| 0007 | Polard | 35000 | Bretlee | Sales |
| 0008 | Bretlee | 60000 | NULL | Sales |

# Discretionary Access Control

- View based Access Controls
  - Create a view for employee list of sales department:

```
CREATE VIEW EMP_SALES
    AS SELECT EMPID,ENAME, SALARY,MANAGER FROM EMPLOYEE
    WHERE DEPTNO='Sales';
```

# Discretionary Access Control

- View based Access Controls(EMP_Sale view)

| EMPID | ENAME | SALARY | MANAGER |
|---|---|---|---|
| 0001 | Smith | 10000 | Bretlee |
| 0002 | Jones | 12000 | Bretlee |
| 0004 | Adams | 20000 | Bretlee |
| 0007 | Polard | 35000 | Bretlee |
| 0008 | Bretlee | 60000 | NULL |

| EMPID | ENAME | SALARY | MANAGER | DEPTNO |
|---|---|---|---|---|
| 0001 | Smith | 10000 | Bretlee | Sales |
| 0002 | Jones | 12000 | Bretlee | Sales |
| 0003 | Baker | 15000 | Milton | Production |
| 0004 | Adams | 20000 | Bretlee | Sales |
| 0005 | Hares | 25000 | Milton | Production |
| 0006 | Milton | 70000 | NULL | Production |
| 0007 | Polard | 35000 | Bretlee | Sales |
| 0008 | Bretlee | 60000 | NULL | Sales |

# Discretionary Access Control

- View based Access Controls for Statistical Information

```
CREATE VIEW   AVSAL(DEPT,AVG)
              , AVG(SALARY)
              GROUP BY DEPTNO;
```

| DEPTNO | AV(SALARY) |
|---|---|
| Sales | 35,000 |
| Production | 36,000 |

# Discretionary Access Control

**Query Modification**

- Another technique for enforcing data-dependent access controls for retrieval.
- A *privilege* allows a user to access some data object in a certain manner (e.g., to read or to modify).
- SQL-92 supports discretionary access control through **GRANT** and **REVOKE** commands.

# Discretionary Access Control

- Query Modification
  - A query submitted by the user is modified to include further restrictions as determined by the DBA.
  - Example: GRANT SELECT ON EMPLOYEE

    TO Thomas

    WHERE DEPTNO="Sales";

# Discretionary Access Control

- Query Modification
  - If Thomas executes the query:

    SELECT * FROM EMPLOYEE;

Data from entire EMPLOYEE table would be returned.

But if the GRANT privilege is applied, DBMS will automatically modify the query to retrieve data only from Sales Department.

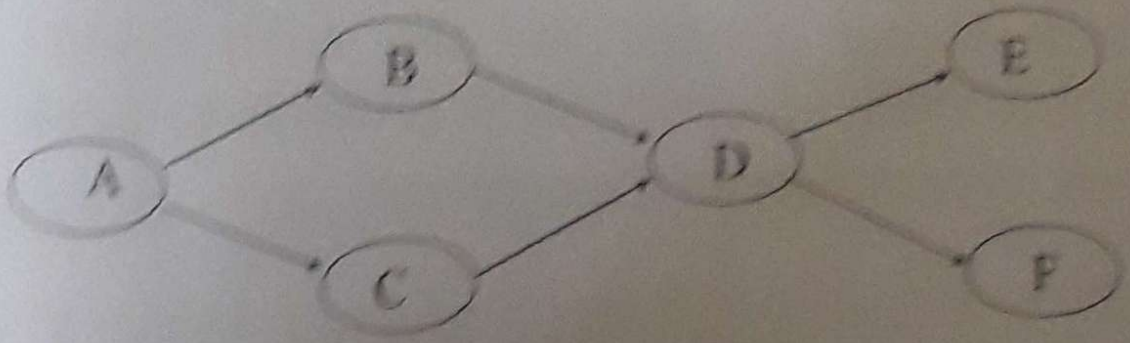
# Discretionary Access Control

- Granting and revocation of Access
  - Granting Access:

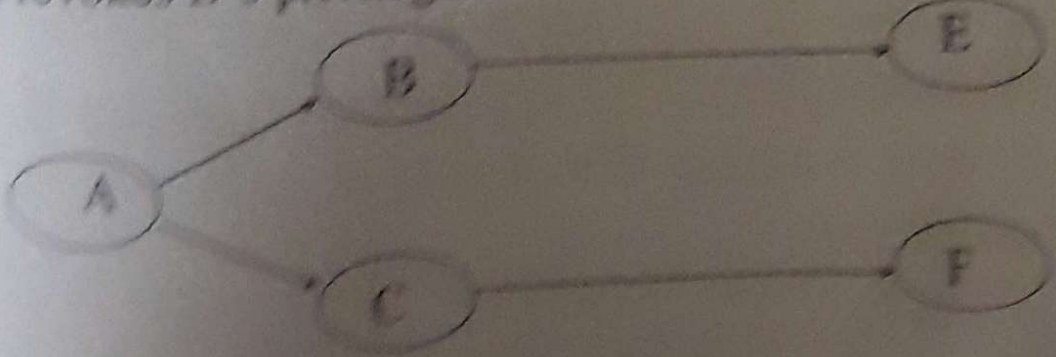
    GRANT privilege
    [ON relation]
    TO users
    [WITH GRANT OPTION]
  - Revocation of Access:

    REVOKE privilege
    [ON relation]
    FROM users
    [WITH GRANT OPTION]

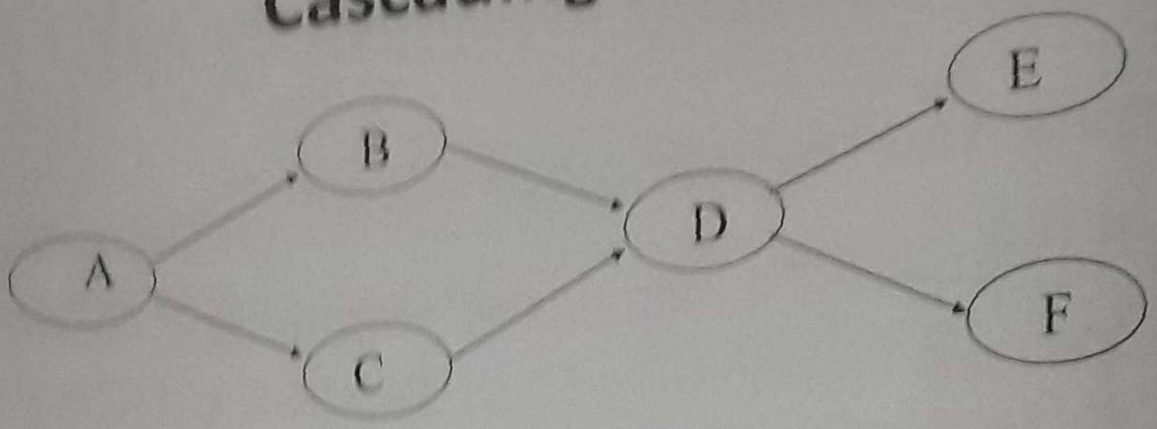
# Non-cascading Revoke

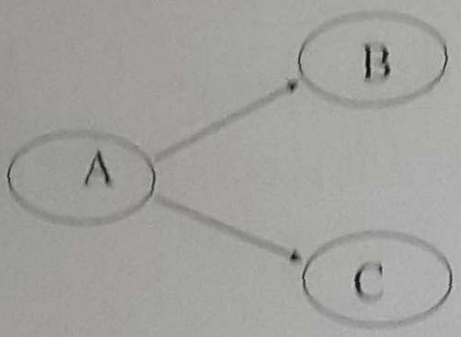



A revokes D's privileges



# Cascading Revoke




If you use **RESTRICT** keyword, the privilege will be revoked only from the specified user. If the specified user granted had the WITH GRANT OPTION and granted the same privilege to other users, they will retain the privilege.

If you use **CASCADE**, it will revoke the privilege and any dependent privileges as a result of your grant. A dependent privilege is one that could exist, if you granted the privilege that you're trying to revoke

# Discretionary Access Control

Granting and Revocation of Access(Examples):

- GRANT SELECT ON EMPLOYEE TO TOM;
- GRANT SELECT,UPDATE(SALARY) ON EMPLOYEE TO TOM;
- GRANT INSERT,DELETE ON EMPLOYEE TO TOM,RICK,HARRY;
- GRANT SELECT ON EMPLOYEE TO TOM WITH GRANT OPTION;
- GRANT DBA TO JILL WITH GRANT OPTION;

# Discretionary Access Control

Granting and revocation of Access (Examples):

- REVOKE SELECT ON EMPLOYEE FROM TOM;
- REVOKE SELECT,UPDATE(SALARY) ON EMPLOYEE FROM TOM;
- REVOKE INSERT,DELETE ON EMPLOYEE FROM TOM,RIK;

# Discretionary Access Control

Granting and revocation of Access(Examples):

- RIK: GRANT SELECT ON EMPLOYEE TO TOM;
- RIK: REVOKE SELECT ON EMPLOYEE TO TOM;

- RIK: GRANT SELECT ON EMPLOYEE TO TOM;
- HARRY: GRANT SELECT ON EMPLOYEE TO TOM;
- RIK: REVOKE SELECT ON EMPLOYEE TO TOM;

TOM continues to retain SELECT privilege due the grant by Harry.

# Discretionary Access Control

Cascading Revocation(Example):

- RIK: GRANT SELECT ON EMPLOYEE TO JOE WITH GRANT OPTION;
- JOE: GRANT SELECT ON EMPLOYEE TO TOM;
- RIK: REVOKE SELECT ON EMPLOYEE TO JOE;

RIK revokes SELECT privilege from JOE. Granting SELECT to TOM by JOE will be revoked also.

# Discretionary Access Control

**Granting and revocation of Access(Example):**

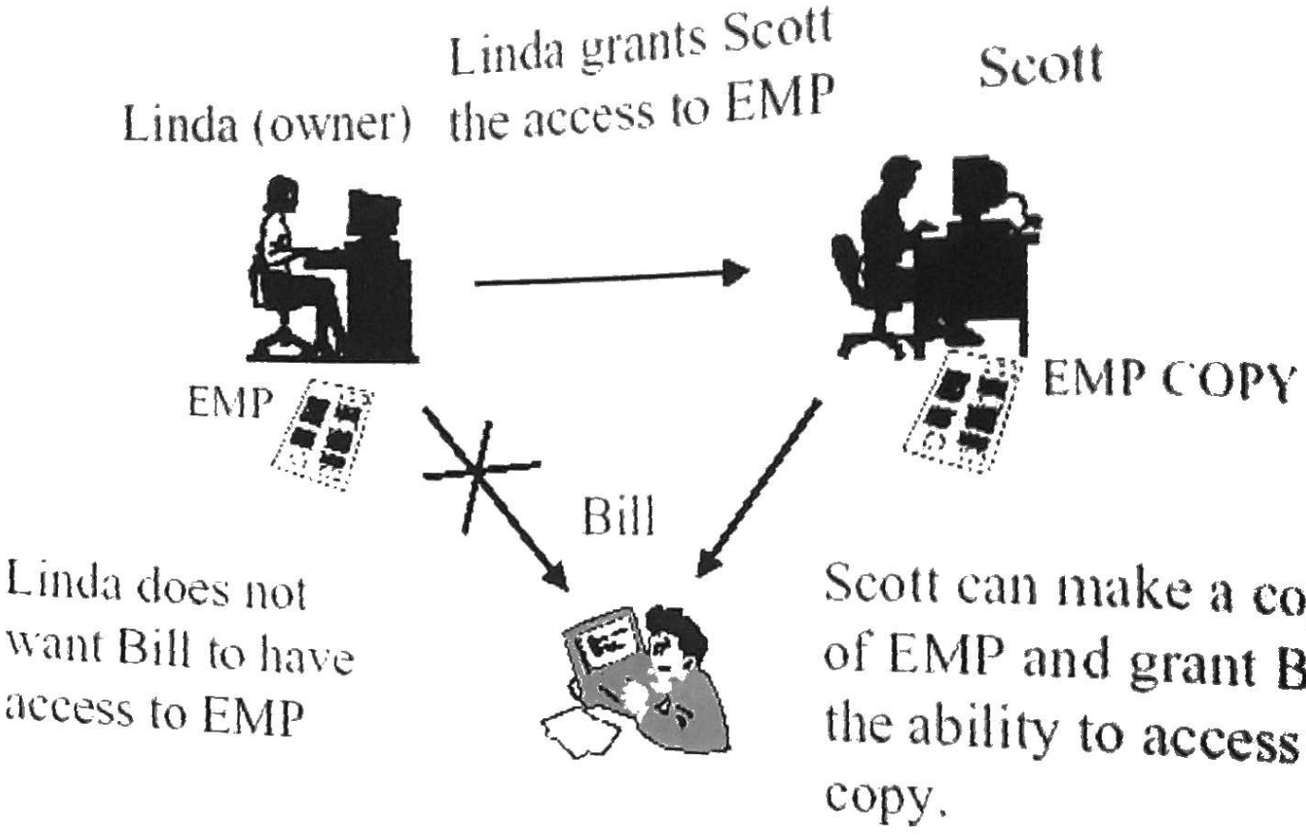
- RIK: GRANT SELECT ON EMPLOYEE TO JOE WITH GRANT OPTION;
- HARRY: GRANT SELECT ON EMPLOYEE TO JOE WITH GRANT OPTION;
- JOE: GRANT SELECT ON EMPLOYEE TO TOM;
- RIK: REVOKE SELECT ON EMPLOYEE FROM JOE;

RIK revokes SELECT privilege from JOE. JOE and TOM continue to retain the SELECT privilege due to the grant by Harry.

# Limitations of DAC

- Granting of access is under user control.
- Users who possess a privilege with the grant option are free to grant it to whoever they choose to.
- Serious limitations with respect to secrecy requirements

# Limitations of DAC




# Another Limitations of DAC (Example)

- TOM: GRANT SELECT ON EMPLOYEE TO RIK;

  No grant option is given to RIK

  Still RIK can pass information to another user say HARRY

- RIK can create a COPY-OF-EMPLOYEE RELATION and can grant privilege to others (Harry) on this relation(table).

Harry has access to all the information in the EMPLOYEE table.

# Another Limitations of DAC (Example)

- Suppose Rik is a trusted confident of Tom and would not deliberately subvert Tom's intentions regarding EMPLOYEE table.
- Rik uses a fancy text editor supplied to him by Harry.
- This editor provides all editing needs that Rik needs.

# Another Limitations of DAC (Example)

- Harry has also programmed it to create COPY-OF-EMPLOYEE relation and execute the following grant operation.
- RIK: GRANT SELECT ON COPY-OF-EMPLOYEE TO HARRY
- Such S/W is said to be a Trojan Horse.

# Hands-on Projects (10 minutes)

You are a security officer working for a medium-sized research company. You have been assigned to guard a back entrance checkpoint. One day, a well-known manager walks out with a box of papers. A day after you are summoned to the security office by your manager and the security director for questioning about the manager who had been terminated the day before. The manager had walked out with highly confidential information.

1. Outline briefly what types of security measures were violated and how to avoid those violations.
2. Describe how this incident may result in security violations.



Thank you !!