RT 801 - Security in Computing

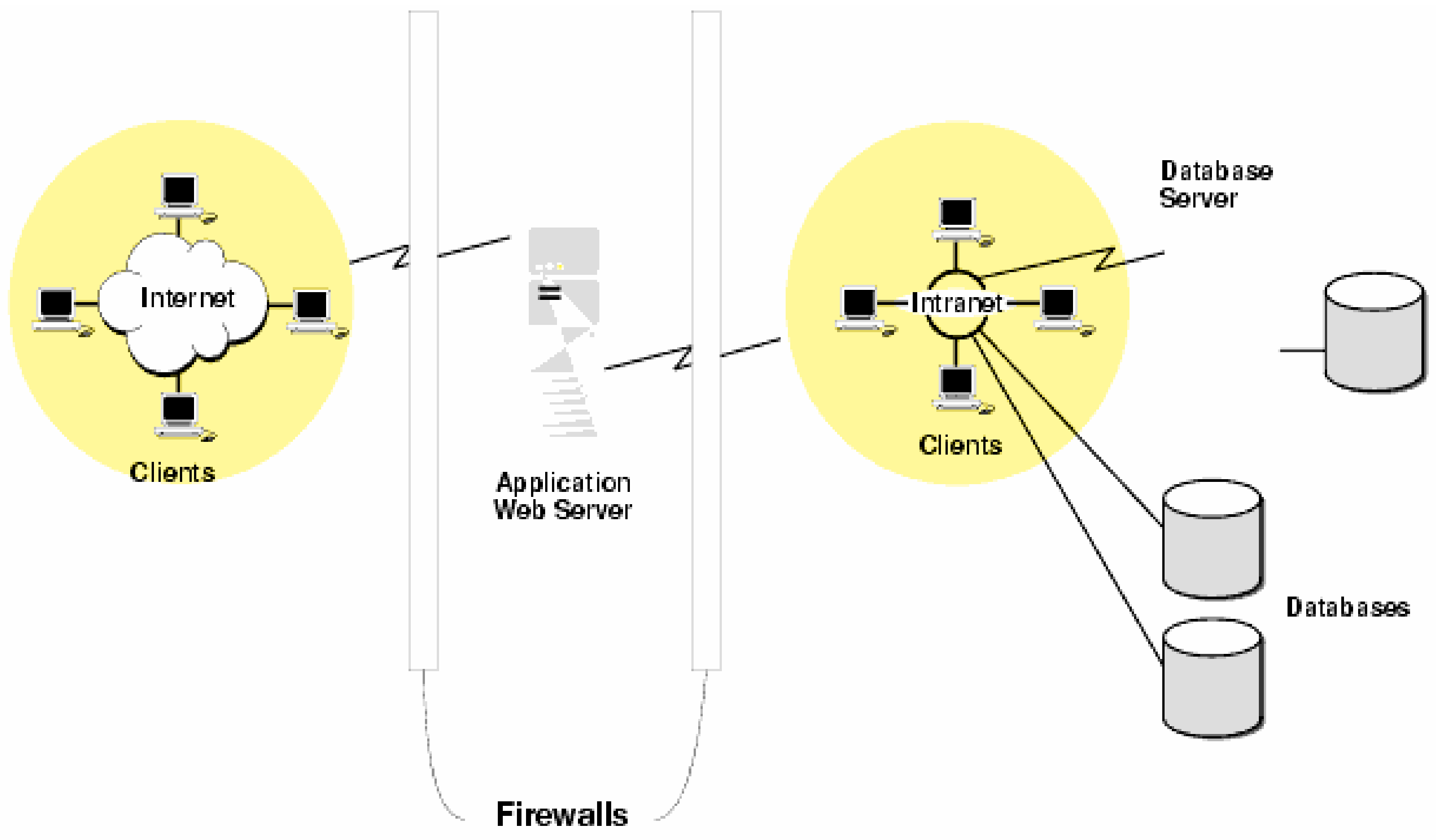
Module V

DATABASE SECURITY

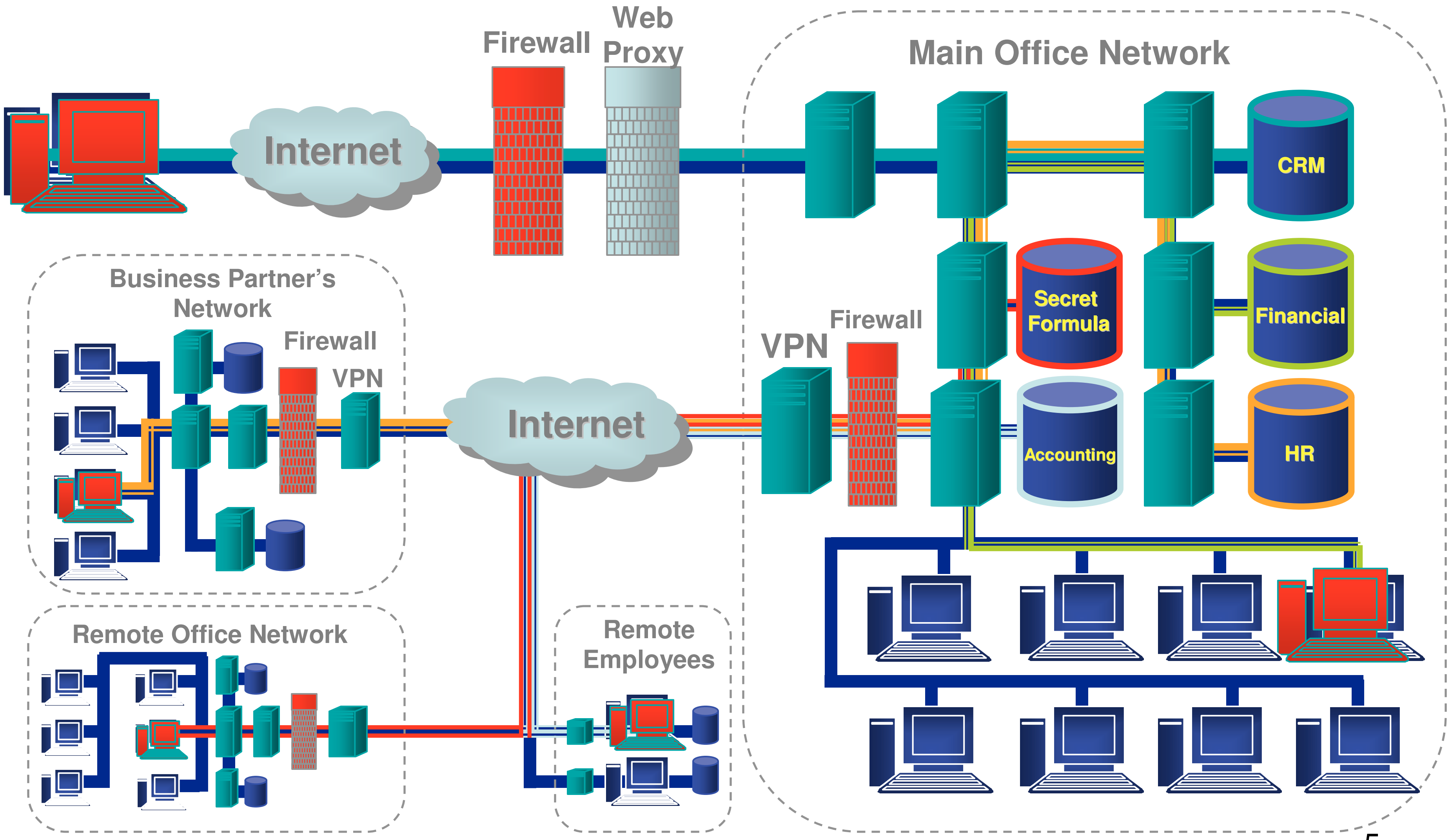
Instructor: Dr. SABU M THAMPI, Rajagiri School of Engineering and Technology, Kochi, India

- Module 5
  - Database Security: - Security issues – SQL security DAC based on granting & revoking privileges – MAC for multilevel security – Statistical database security.

- A decade ago, databases were
  - Physically secure
  - Housed in central data centers – not distributed
  - External access mediated
  - Security issues rarely reported
- Now, databases are externally accessible
  - Suppliers directly connected
  - Customers directly connected
  - Customers and partners directly sharing data
- Data is the most valuable resource in application stack
  - Value increases with greater integration and aggregation
  - But so does the threat of data theft, modification, or destruction


Internet
Clients
Application
Web Server
Firewalls
Intranet
Clients
Database
Server
Databases

Barrier Defense Is No Longer Enough



- **Database security** is the system, processes, and procedures that protect a database from unintended activity.
- Unintended activity can be categorized as authenticated misuse, malicious attacks or inadvertent mistakes made by authorized individuals or processes.
- *Database security* is also a specialty within the broader discipline of computer security.

- Database security requirements arise from the need to protect data:
  - From accidental loss and corruption, and
  - From deliberate unauthorized attempts to access or alter that data.
  - Secondary concerns include protecting against undue delays in accessing or using data.
  - The global costs of such security breaches run to billions of dollars annually, and the cost to individual companies can be severe, sometimes catastrophic.

Why is database security important?

- If the loss of system or data integrity is not corrected, continued use of the contaminated system or corrupted data could result in inaccuracy, fraud, or erroneous decisions.
- In addition, unauthorized, unanticipated, or unintentional disclosure could result in loss of public confidence, embarrassment, or legal action against the organization.

Security Issues by Category

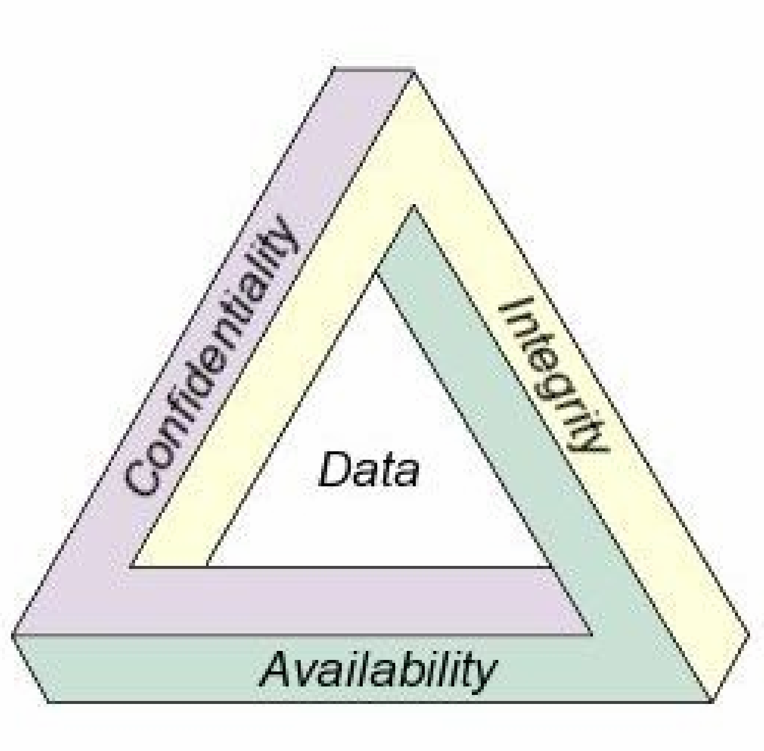
- **Procedural**
  - The procedures and policies used in the operation of your system must assure reliable data.
  - It is often wise to separate out users' functional roles in data management.
    - For example, one person might be responsible for database backups. Her only role is to be sure the database is up and running.
    - Another person might be responsible for generating application reports involving payroll or sales data. His role is to examine the data and verify its integrity.
  - Further, you can establish policies that protect tables and schemas against unauthorized, accidental, or malicious usage.

Security Issues by Category

- **Technical**
  - Storage, access, manipulation, and transmission of data must be safeguarded by technology that enforces your particular information control policies.
- **Physical**
  - Computers must be made physically inaccessible to unauthorized users by keeping them in a secure physical environment.
- **Personnel**
  - The people responsible for your site's physical security, system administration, and data security must be reliable.
  - Performing background checks on DBAs before making hiring decisions is a wise protective measure.

- Database security begins with physical security for the computer systems that host the DBMS.
  - No DBMS is safe from intrusion, corruption, or destruction by people who have physical access to the computers.
- After physical security has been established, database administrators must protect the data from unauthorized user and from unauthorized access by authorized users.
- There are three main objects when designing a secure database application
  - Anything prevents from a DBMS to achieve these goals would be consider a threat to Database Security.

CIA Triad



What are the Basic Facets of Database Security?

- Database security can be defined as a system or process by which the "Confidentiality, Integrity, and Availability," or CIA, of the database can be protected.
- Unauthorized entry or access to a database server signifies a loss of confidentiality
- Unauthorized alteration to the available data signifies loss of integrity
- and lack of access to database services signifies loss of availability
- Loss of one or more of these basic facets will have a significant impact on the security of the database.

Illustration

- Imagine that the website of a company contains information like who they are, what they do, and what prospective customers have to do to contact them for their queries.
  - the availability of the database services is more important when compared with other factors like the confidentiality or integrity of the database security.
- For a company that sells products or goods online
  - confidentiality and integrity are more important as customers use their credit cards to buy goods online only when the site is available.

Confidentiality

- In database security concepts, Confidentiality comes first.
  - Confidentiality can be enforced by encrypting the data stored in the database.
  - Encryption is a technique or process by which data is encoded in such a way only authorized users be able to read the data.
  - In other words, encryption means rendering sensitive data unreadable to unauthorized users.
- Encryption can be done at two different levels: data-in-transit and data-at-rest.

Data-in-transit

- This refers to data that is moving within the network.
  - Sensitive data, for example, that is sent through network layers or through the Internet.
- A hacker can gain access to this sensitive data by eavesdropping.
  - When this happens, the confidentiality of the data is compromised.
- Encrypting data-in-transit avoids such compromises.

Data-at-rest

- Possible for a hacker to hack the data that is stored in the database.
- The data-at-rest can be made secured by providing two level of security:
  - controlling the access to the data by Access control and Encryption.
- Different encryption algorithms are available, which includes Data Encryption Standards (DES), Triple DES or 3DES, and Advanced Encryption Standards (AES).

Integrity

- Integrity can be enforced by setting User Access Controls (UAC) that define which users have to be given what permissions in the database.
- For example, data related to employee information is stored in a database.
  - An employee may have permission for viewing the records and altering only part of information like his contact details, whereas a person in the human resources department will have more privileges.
  - Students may be allowed to see their grades, yet not allowed to modify it.

Availability

- Databases must not have unplanned downtime.
  - Downtime can equate to loss of revenue and/or loss of production.
- To ensure this, following steps have to be taken:
  - Limit the number of concurrent sessions made available to each database user.
  - Backup the data at periodic intervals to ensure data recovery.
  - Databases should be secured against security vulnerabilities.
  - To ensure high availability, use database clusters

Types of threats to database security

- Privilege abuse
- Operating System vulnerabilities
- Weak authentication
- Weak audit trails

Privilege abuse

- When database users are provided with privileges that exceeds their day-to-day job requirement
  - these privileges may be abused **intentionally or unintentionally.**
- For instance, a database administrator in a financial institution.
  - What will happen if he create bogus accounts?
    - He will be able to transfer money from one account to another thereby abusing the excessive privilege intentionally.
- A university administrator whose job requires only the ability to change student contact information
  - may take advantage of excessive database update privileges to change grades.

Privilege abuse

- **How privilege can be abused unintentionally?**
  - A company is providing a “work from home” option to its employees and the employee takes a backup of sensitive data to work on from his home.
    - Violates the security policies of the organization
    - May result in data security breach if the system at home is compromised.

Operating System vulnerabilities

- Vulnerabilities in underlying operating systems like Windows, UNIX, Linux, etc., and the services that are related to the databases could lead to unauthorized access.
  - Lead to a Denial of Service (DoS) attack.
    - Can be prevented by updating the operating system related security patches as and when they become available.

- vulnerabilities in the new Windows 7 operating system
- There is a current vulnerability in Windows Explorer which contains all of the files in your PC's operating system.
- Windows Explorer in the Windows 7 operating system hides file extensions by default.
- Since Windows Explorer hides file extensions by default, the malware can present itself as "destructive_malware.txt.exe."
  - When you view your files in Windows Explorer the extra file extension ".exe" is not visible.

Weak authentication

- Weak authentication models allow attackers to employ strategies such as brute force and social engineering to obtain database login credentials and assume the identity of legitimate database users.
  - A brute force attack consists of trying every possible password until you find the right one.

Social Engineering

- Social engineering describes a non-technical kind of intrusion that relies heavily on human interaction and often involves tricking other people to break normal security procedures.
  - For example, a person using social engineering to break into a computer network would try to gain the confidence of someone who is authorized to access the network in order to get them to reveal information that compromises the network's security.

Social Engineering Example

Mr. Smith: Hello?

Caller: Hello, Mr. Smith. This is Fred Jones in tech support. Due to some disk space constraints, we're going to be moving some user's home directories to another disk at 8:00 this evening. Your account will be part of this move, and will be unavailable temporarily.

Mr. Smith: Uh, okay. I'll be home by then, anyway.

Caller: Good. Be sure to log off before you leave. I just need to check a couple of things. What was your username again, Smith?

Mr. Smith: Yes. It's smith. None of my files will be lost in the move, will they?

Caller: No sir. But I'll check your account just to make sure. What was the password on that account, so I can get in to check your files?

Mr. Smith: My password is tuesday, in lower case letters.

Caller: Okay, Mr. Smith, thank you for your help. I'll make sure to check you account and verify all the files are there.

Mr. Smith: Thank you. Bye.

- Social engineering preys on qualities of human nature:
  - the desire to be helpful
  - the tendency to trust people
  - the fear of getting into trouble

- The sign of a truly successful social engineer is they receive information without raising any suspicion as to what they are doing.

- Social engineering is the hardest form of attack to defend against because it cannot be defended with hardware or software alone.
- A successful defense depends on having good policies in place ensuring that all employees follow them.

Weak audit trails

- A weak audit logging mechanism in a database server represents a critical risk to an organization especially in retail, financial, healthcare, and other industries with stringent regulatory compliance.
- Logging of sensitive or unusual transactions happening in a database must be done in an automated manner for resolving incidents.
- Audit trails act as the last line of database defense.
  - Audit trails can detect the existence of a violation that could help trace back the violation to a particular point of time and a particular user.

Countermeasures to database security threats

- Flow Control
- Encryption
- Access Control
- Inference control

Flow Control

- Flow control regulates the distribution or flow of information among accessible objects.
- A flow between object X and object Y occurs when a program reads values from X and writes values into Y.
- Flow controls check that information contained in some objects does not flow explicitly or implicitly into less protected objects.
  - Thus, S user cannot get indirectly in Y what he or she cannot get directly from X.

Flow control example

- Preventing a service program from leaking a customer’s confidential data
  - The bank receives periodic requests from each customer
    - e.g., to withdraw or deposit money.
  - Each request should be able to observe only information that is owned by that customer, and none of the bank’s private data.
- Blocking the transmission of secret military data to an unknown classified user.

- There are two types of information flows: *indirect read* and *indirect write*.
- Users may indirectly read an unauthorized relation through a sequence of access operations.
  - For example, authorized users may read data from sensitive tables and then write to a table shared with unauthorized users.
  - In this way, unauthorized users can access the sensitive data by reading data from the shared table.
    - That is, unauthorized users can indirectly read sensitive data from authorized users.

- Users may also indirectly write to an unauthorized relation through a sequence of access operations.
  - Unauthorized users may write data to shared tables and then authorized users read the data from the shared tables and write into the sensitive tables.
  - In this way, users can indirectly write data to unauthorized tables.

Flow Policy

- A **flow policy** specifies the channels along which information is allowed to move.
  - – The simplest flow policy specifies just two classes of information:
    - confidential (C) and non-confidential (N)
  - – and allows all flows except those from class C to class N.
  - – For example an income-tax computing service might be allowed to retain a customer’s address and the bill for services rendered, but not a customer's income or deductions.

Access Control Vs. Flow Control

- Access control mechanisms
  - Check users’ authorizations for resource access
  - Only granted operations are executed
- Flow controls can be enforced by an extended access control mechanism
  - Involves assigning a security class (clearance) to each running program

Access Control Vs. Flow Control

  - The program is allowed to read a particular memory segment only if its security class is as high as that of the segment.
  - It is allowed to write in a segment only if its class is as low as that of the segment
- Ensures no information transmitted by the person can move from a higher to a lower class.
  - A military program with a secret clearance can only read from objects that are unclassified and confidential and can only write into objects that are secret or top secret.

Encryption

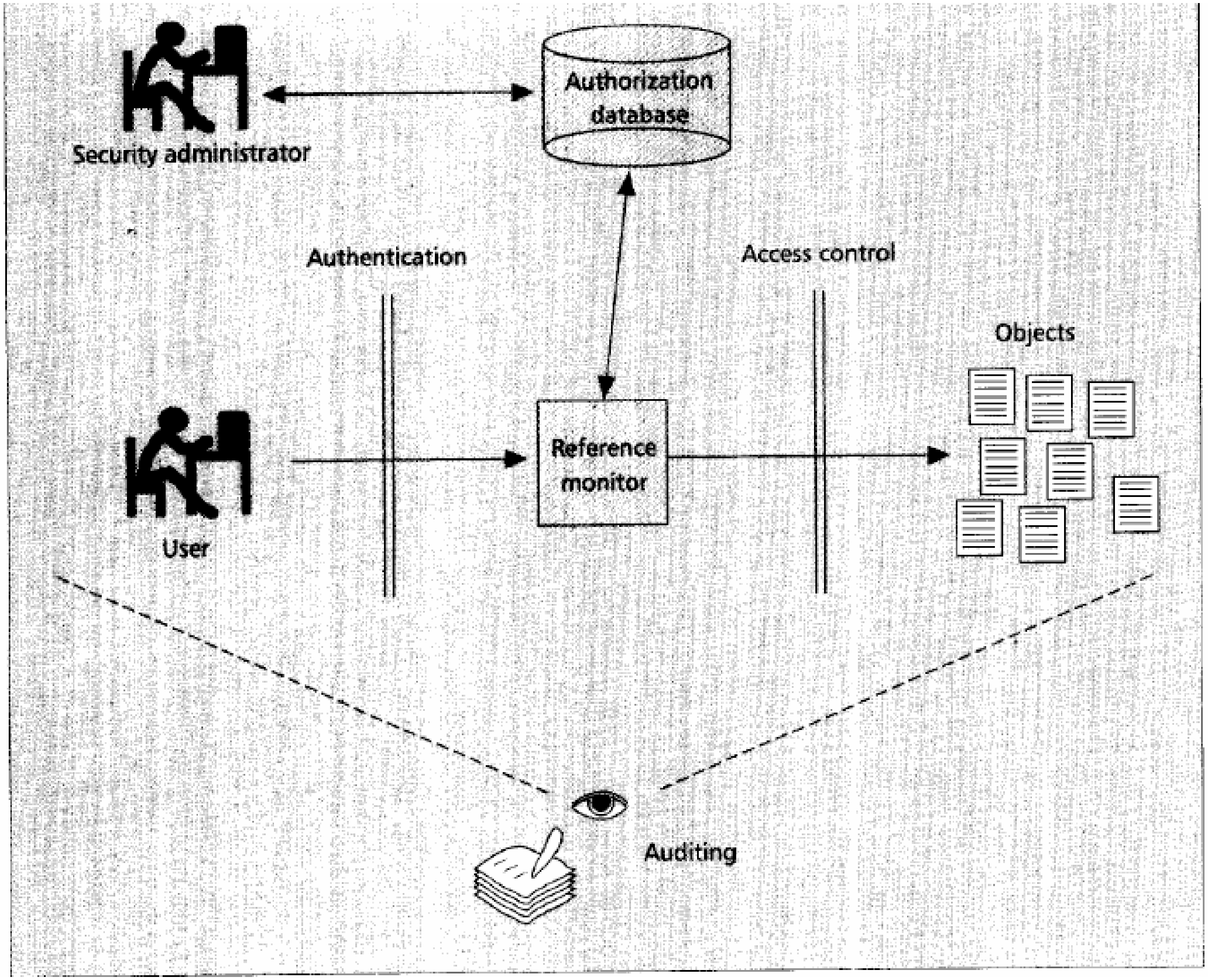
- The idea behind encryption is to apply an encryption algorithm to the data, using a user-specified or DBA-specified encryption key.
- The output of the algorithm is the encrypted version of the data.
- There is also a decryption algorithm, which takes the encrypted data and a decryption key as input and then returns the original data.

Authentication and Access Control

- Correctly establishing the identity of the user is the responsibility of the authentication service.
- Access control assumes that authentication of the user has been successfully verified prior to enforcement of access control via a reference monitor.
- The effectiveness of the access control rests on a proper user identification and on the correctness of the authorizations.

Auditing

- Access control is not a complete solution for securing a system.
- It must be coupled with auditing.
- Auditing controls concern an analysis of all the requests and activities of users in the system.
- Auditing requires the registration (logging) of all user requests and activities for their later analysis.
- Auditing can be useful for determining possible flaws in the security system
- Auditing is essential to ensure that authorized users do not misuse their privileges.



Access control is not a complete solution for securing a system; it must be coupled with auditing.

- **Reference monitor**
  - A means of checking that a particular user is allowed access to a specified object in a computing system.
    - Also known as access-control mechanism; reference validation mechanism.
  - The reference monitor verifies the nature of the request against a table of allowable access types for each process/object on the system.

Subjects

- Activity in the system is initiated by entities known as subjects
- Subjects are typically users or programs executing on behalf of users.
- A user may sign on to the system as different occasions, depending on which privileges the user wishes to exercise in a given session.
  - For example, a user working on two different projects may sign on for purpose of working on one project or the other.
    - We then have two subjects corresponding to this user, depending on the project the user is currently working on.

- A subject can create additional subjects in order to accomplish its task.
- The children subjects may be executing on various computers in a network.
- The parent subject will usually be able to suspend or terminate its children as appropriate.

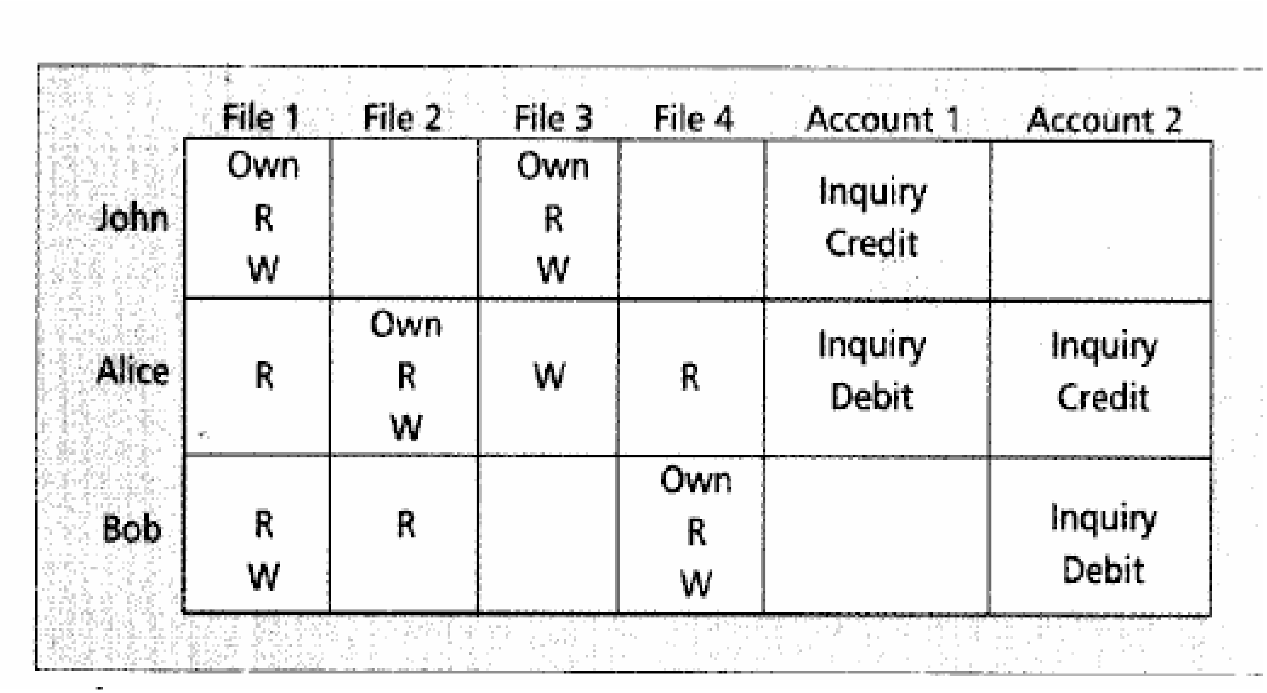
- Subjects initiate actions or operations on objects.
- These actions are permitted or denied in accord with the authorizations established in the system.
- Authorizations are expressed in terms of access rights or access modes.
- The meaning of access rights depends upon the object in question.
  - For files the typical access rights are read, write, execute and own.

- Ownership is concerned with controlling who can change the access permissions for the file.
- An object such as bank account may have access rights inquiry, credit and debit corresponding to the basic operations that can be performed an account.
  - These operations would be implemented by application programs whereas for a file the operations would typically be provided by the operating system.

Access Matrix

- It is a conceptual model that specifies the rights that each subject possesses for each object.
- There is a row in this matrix for each subject and a column for each object.
- Each cell of the matrix specifies the access authorized for the subject in the row to the object in the column.
- The task of access control is to ensure that only those operations authorized by the access matrix actually get executed..
  - This is achieved by means of a reference monitor, which is responsible for mediating all attempted operations by subjects on objects.

An Access Matrix



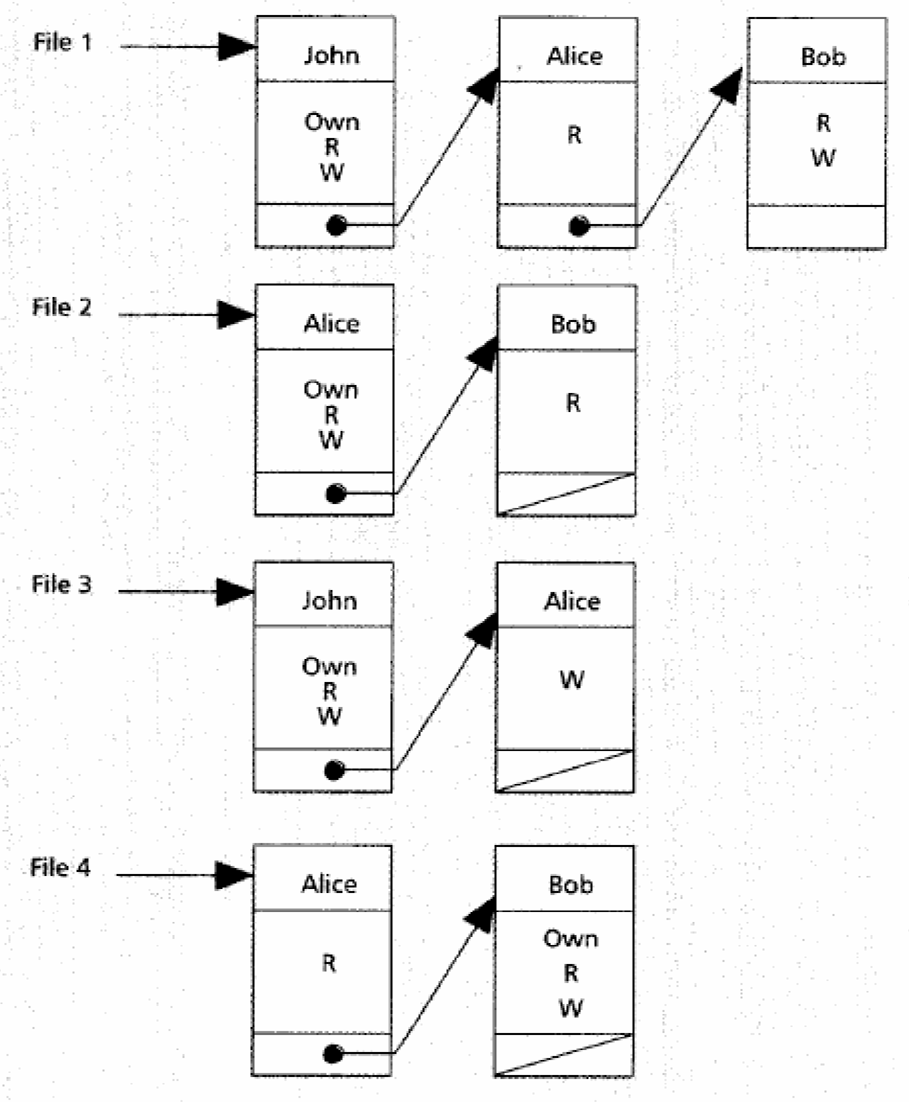
	File 1	File 2	File 3	File 4	Account 1	Account 2
John	Own R W		Own R W		Inquiry Credit	
Alice	R	Own R W	W	R	Inquiry Debit	Inquiry Credit
Bob	R W	R		Own R W		Inquiry Debit

John is the owner of file 3and he has no access to File 2 or File 4. The owner of a file is authorized to grant other users access to the file as well as revoke access.

- In a large system the access matrix will be enormous in size and most of its cells are likely to be empty.

Access Control Lists

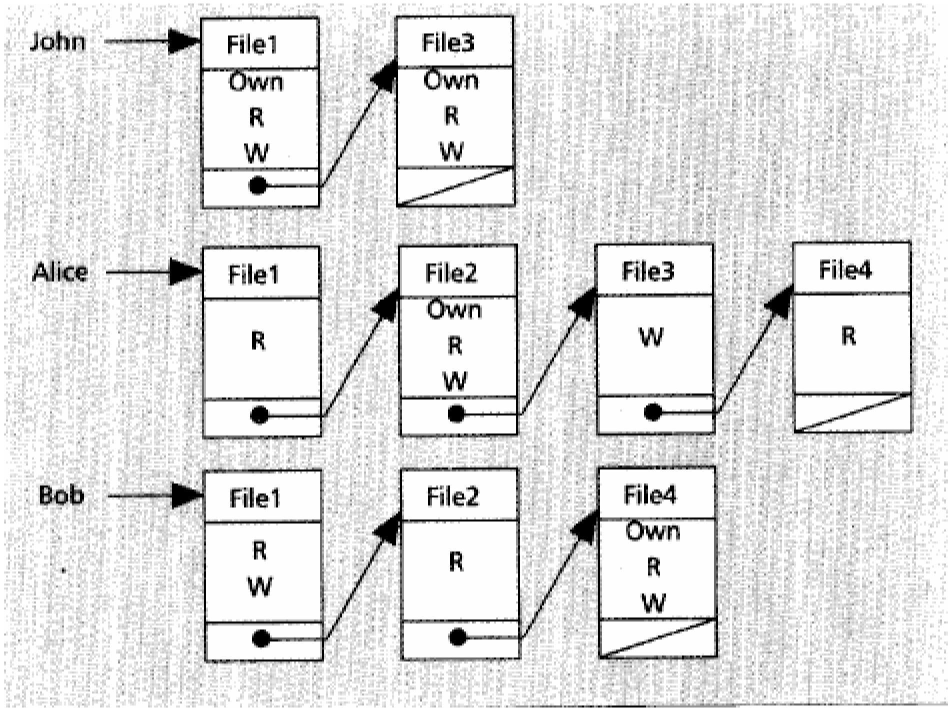
- A popular approach to implementing the access matrix
- Each object is associated with an ACL indicating for each subject in the system the accesses the subject is authorized to execute on the object.



- ACL is very convenient to use.
  - By looking at an object’s ACL it is very easy to determine which modes of access subjects are currently authorized for that object.
- Also easy to revoke all accesses to an object by replacing the existing ACL with an empty one.
- If all accesses of a subject need to be revoked all ACLs must be visited one by one.
  - In practice revocation of all accesses of a subject is often done by deleting the user account corresponding to that subject. This is acceptable if a user is leaving an organization.
  - However, if a user is reassigned within the organization it would be more convenient to retain the account and change its privileges to reflect the changed assignment of the user.

Capability Lists

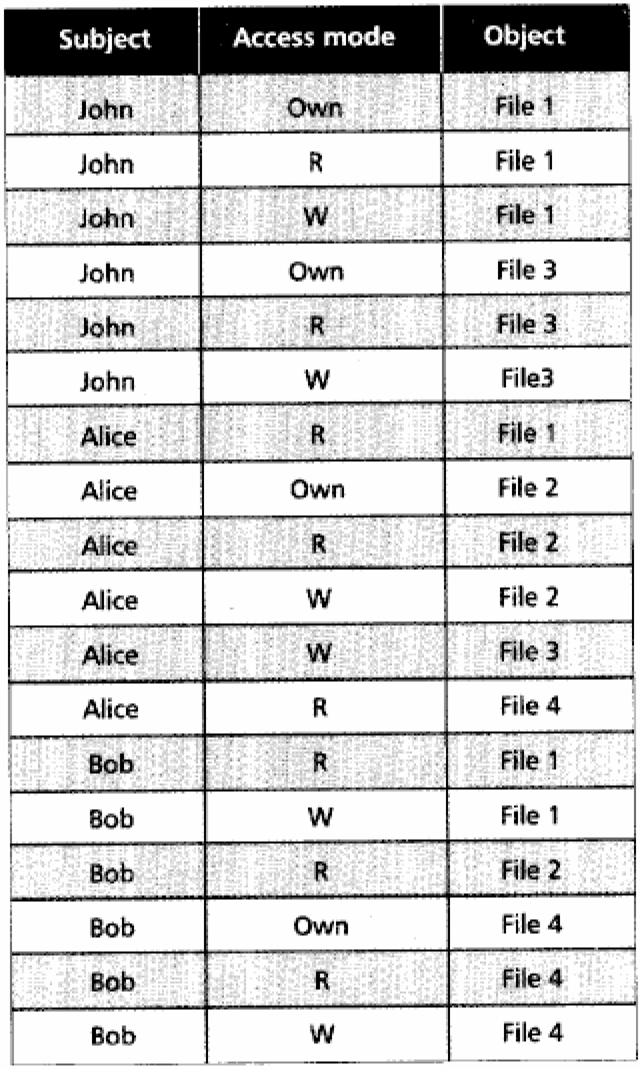
- Capabilities are a dual approach to ACLs.
- Each subject is associated with a list that indicates , for each object in the system, which accesses the subject is authorized to execute on the object.



- In a capability list approach, it is very easy to review all accesses that a subject is authorized to perform, by simply examining the subject’s capability list.

Authorization Relations

- The access matrix can be represented by an authorization relation
- Each tuple of the table specifies one access right of a subject to an object.
  - – Thus, John’s accesses to File1 require three rows.
- If this table is sorted by subject, we get the effect of capability lists.
- If it is sorted by object, we get the effect of ACLs.
- RDBMSs typically use such a representation.

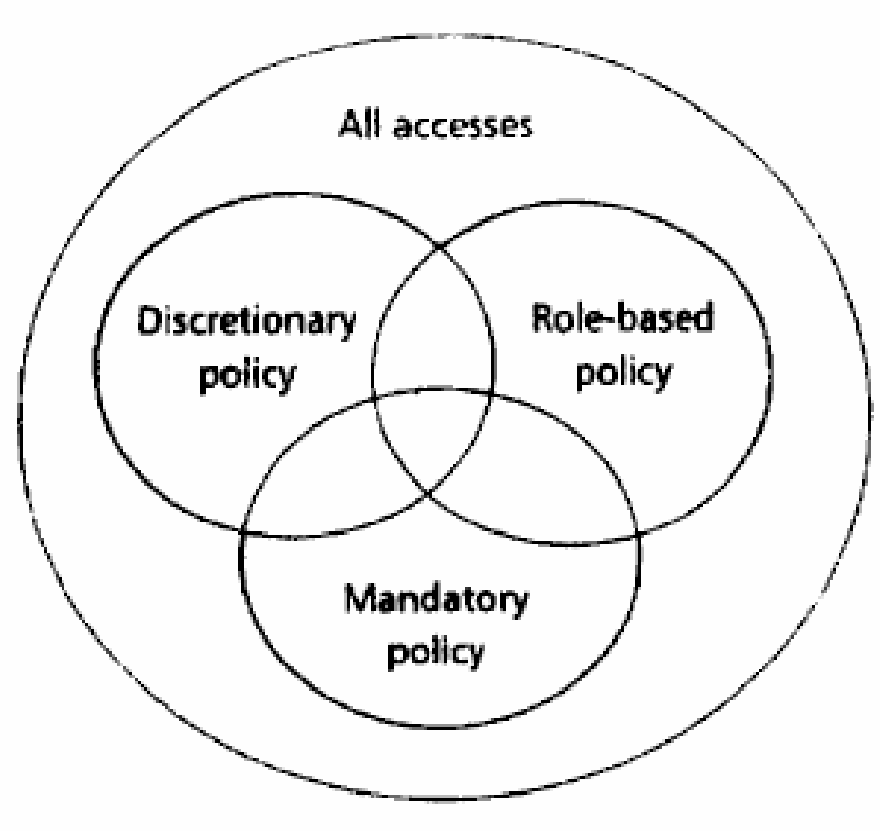


Subject	Access mode	Object
John	Own	File 1
John	R	File 1
John	W	File 1
John	Own	File 3
John	R	File 3
John	W	File3
Alice	R	File 1
Alice	Own	File 2
Alice	R	File 2
Alice	W	File 2
Alice	W	File 3
Alice	R	File 4
Bob	R	File 1
Bob	W	File 1
Bob	R	File 2
Bob	Own	File 4
Bob	R	File 4
Bob	W	File 4

Access Control Policy

- Policies are higher level guidelines that determine how accesses are controlled and access decisions determined.
- Mechanisms are low-level software and hardware functions that can be configured to implement a policy.
- Security researchers have sought to develop access control mechanisms that are largely independent of the policy for which they could be used.
  - Helps to reuse of mechanisms that serve a variety of security purposes.

Multiple Access Control Policies

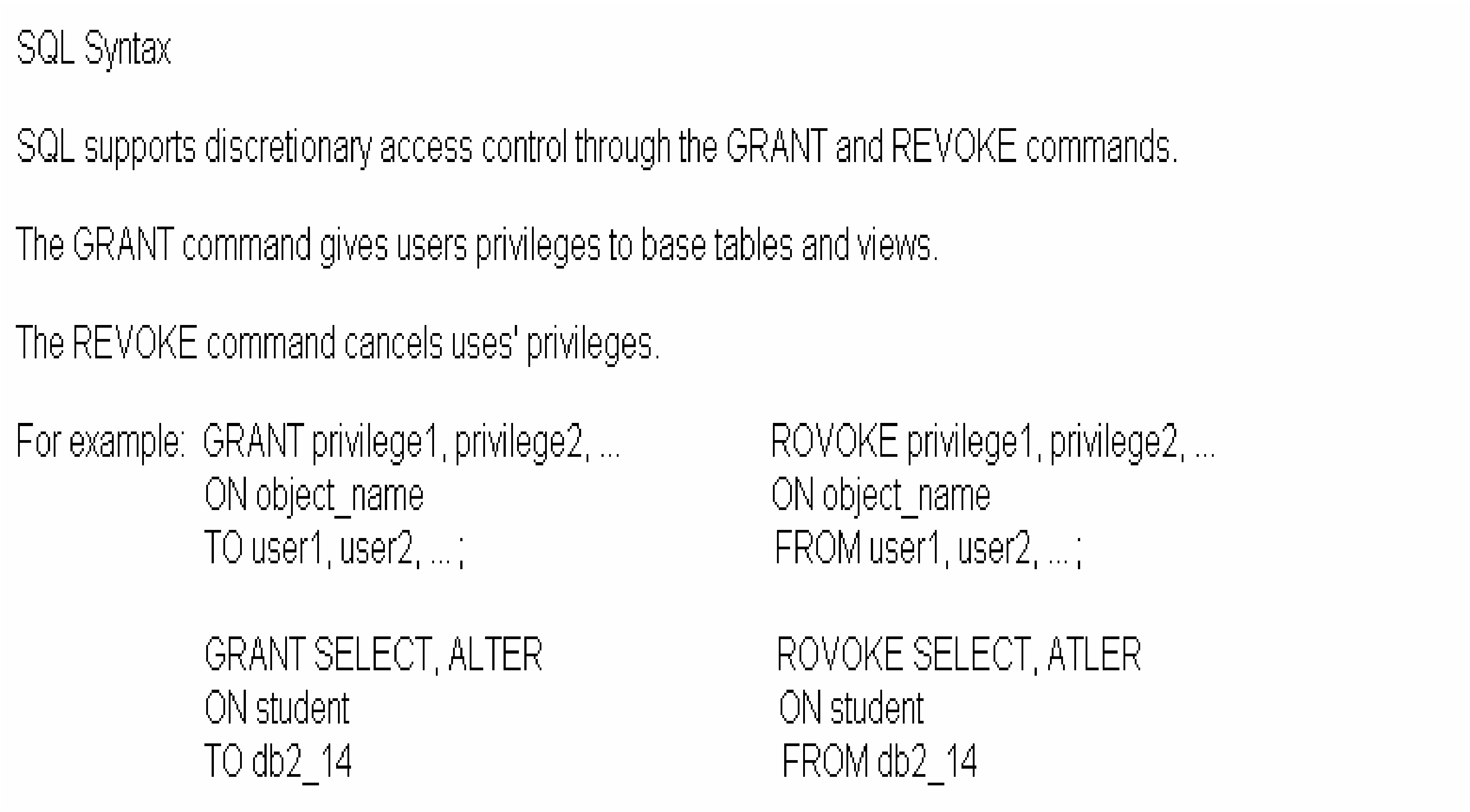


Different policies can be combined to provide a more suitable protection system.

Discretionary Access Control

- Discretionary access control is based on the idea of access rights, or privileges, and mechanisms for giving users such privileges.

- A user who creates data object such as a table or a view automatically gets all applicable privileges on that object
  - and the user can also propagate privileges using "**Grant Option**".
- The DBMS keeps track of how these privileges are granted to other users and ensures that at all times only users with the necessary privileges can access an object.



SQL Syntax

SQL supports discretionary access control through the GRANT and REVOKE commands.

The GRANT command gives users privileges to base tables and views.

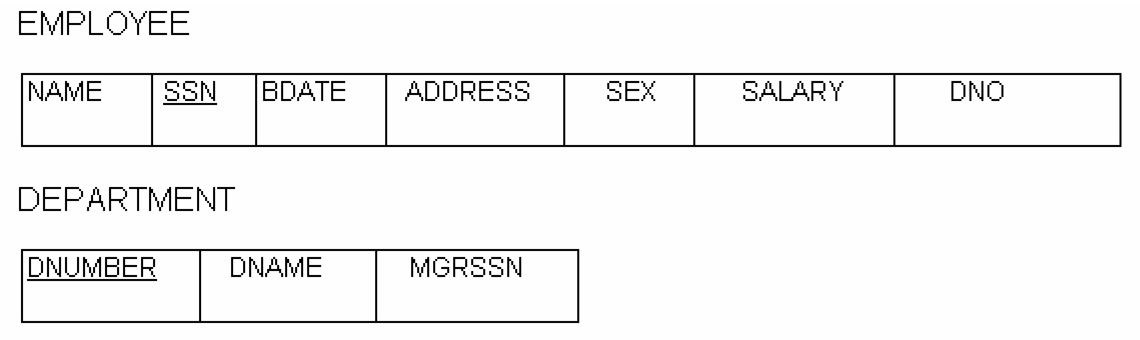
The REVOKE command cancels uses' privileges.

```
For example:  GRANT privilege1, privilege2, ...          ROVOKE privilege1, privilege2, ...
              ON object_name                             ON object_name
              TO user1, user2, ... ;                     FROM user1, user2, ... ;

              GRANT SELECT, ALTER                        ROVOKE SELECT, ATLER
              ON student                                 ON student
              TO db2_14                                  FROM db2_14
```

Example

- Suppose that A1 creates the two base relations EMPLOYEE and DEPARTMENT



EMPLOYEE

NAME	SSN	BDATE	ADDRESS	SEX	SALARY	DNO

DEPARTMENT

DNUMBER	DNAME	MGRSSN

- A1 is then the owner of these two relations and hence has all the relation privileges on each of them.
- A1 wants to grant to account A2 the privilege to insert and delete tuples in both of these relations
  - GRANT INSERT, DELETE ON EMPLOYEE, DEPARTMENT TO A2;
- A2 cannot grant INSERT and DELETE privileges on the EMPLOYEE and DEPARTMENT tables, because A2 was not given the *GRANT OPTION* in the preceding command.
  - GRANT SELECT ON EMPLOYEE, DEPARTMENT TO A3 with GRANT OPTION;

- The clause WITH GRANT OPTION means that A3 can now propagate the privilege to other accounts by using GRANT.
  - For example, A3 can grant the SELECT privilege on the EMPLOYEE relation to A4 by issuing the following command:
    - GRANT SELECT ON EMPLOYEE TO A4;
- Now suppose that A1 decides to revoke the SELECT privilege on the EMPLOYEE relation from A3; A1 then can issue this command:
  - REVOKE SELECT ON EMPLOYEE FROM A3;
- The DBMS must now automatically revoke the SELECT privilege on EMPLOYEE from A4, too, because A3 granted that privileges to A4 and A3 does not have the privilege any more.

Pros and Cons of discretionary access control

- Advantages:
  - Being easy to implement for various types of systems and application like commercial and industrial environments.
- Disadvantages:
  - Not imposing any restriction on the usage of information once it is obtained by a user and makes system vulnerable to attacks.
    - For example, a user who is able to read data can pass it to other users not authorized to read it without the cognizance of the owner.

Mandatory Access control

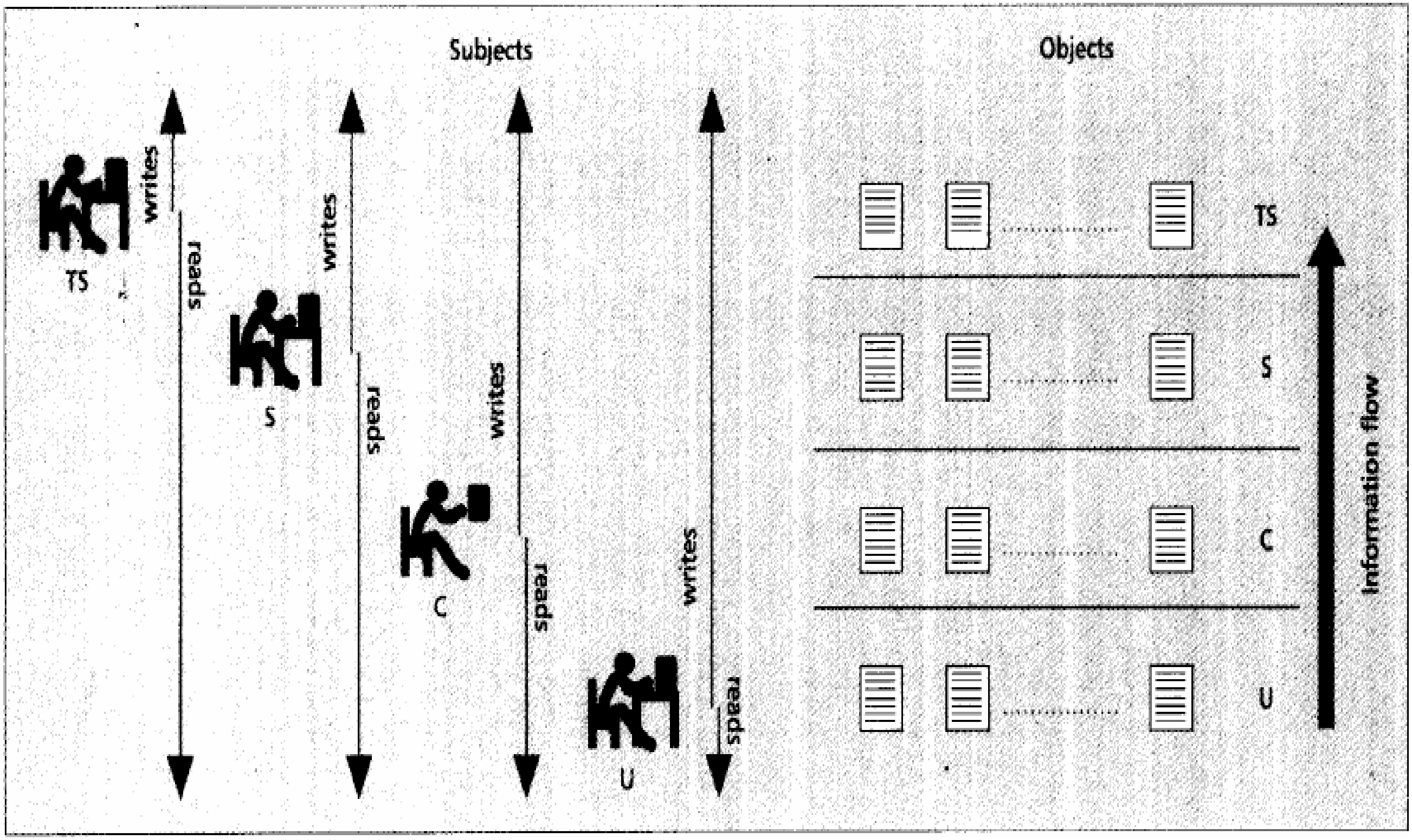
- Mandatory access control are aimed at addressing the loopholes in discretionary access control.
- The popular model for mandatory access control - Bell-LaPadula model
  - described in terms of objects, subjects, security classes, and clearances.
- Each database object is assigned a security class, and each subject is assigned clearance for a security class.

- The **security level** is an element of a hierarchical ordered set.
- In the military and civilian government arenas, the hierarchical set consists of TopSecret (TS), Secret (S), Confidential (C), and Unclassified (U), where TS>S>C>U.
- Access to an object by a subject is granted only if some relationship is satisfied between the security levels associated with the two.

- The following two principles are required to hold:
  - Read down – A subject’s clearance must dominate the security level of the **subject being read**.
  - Write up – A subject’s clearance must be dominated by the security level of the **object being written**.
- See Figure slide 70
- Satisfaction of these principles prevents information in high-level objects (i.e. more sensitive) to flow to objects at lower levels.
- In such a system the information can only flow upwards or within the same security class.

- The Bell-LaPadula model imposes two restrictions on all reads and writes of database objects:
  - **Subject S is allowed to read object O only if class(S)≥ class(O).**
    - For example, a user with TS (top secret) clearance can read a table with C(confidential) clearance, but a user with C(Confidential) clearance is not allowed to read a table with TS (top secret) classification.
  - **Subject S is allowed to write object O only if class(S)≤ class(O).**
    - For example, a user with S (secret) clearance can write only objects with S (secret) or TS (top secret) classification.

Controlling Information Flow for Secrecy



Important to understand the relationship between users and subjects

- Let us say that the human user Jane is cleared to S and assume she always signs on to the system as an S subject (i.e. a subject with clearance S)
- Jane’s subjects are prevented from reading TS objects by the read-down rule.
- Jane’s subjects can write a TS object.
    - They can overwrite existing TS data and therefore destroy it. Due to this integrity concern, many systems for mandatory access control do not allow write up.
        - At the same time, write up does allow Jane’s S subjects to send e-mail to TS subjects.

- Jane’s subjects can not write C or U data.
  - For example, Jane can never send e-mail to C or U users.
  - But, Jane signs to the system as a C or U
  - During these sessions she can send e-mail to C or U and C subjects respectively.
- In other words, a user can sign on to the system as a subject at any level dominated by the user’s clearance.

Why then bother to impose write-up rule

- To prevent malicious software from leaking secrets downward from S to U
- Users are trusted not to leak such information, but the programs they execute do not merit the same degree of trust.
  - For example, Jane signs onto the system at the level in which her subjects cannot read S objects, and thereby cannot leak data from S to U.
- The write up rule also prevents users from inadvertently leaking information from high to low.

- **Advantages:** Mandatory policies ensure a high degree of protection.
  - Suitable for military types of applications, which require a high degree of protection.
- **Disadvantages**: Applicable to very few environment for being too rigid.

- **DAC** permits the granting and revoking of access control privileges to be left to the discretion of the individual users.
  - A DAC mechanism allows users to grant or revoke access to any of the objects under their control.
- **MAC** is “a means of restricting access to objects based on the sensitivity of the information contained in the objects and the formal authorization (i.e. clearance) of subjects to access information of such sensitivity."

DAC and MAC

- Not particularly well suited to the requirements of government and industry organizations that process unclassified but sensitive information.
  - In these environments, security objectives often support higher-level organizational policies which are derived from existing laws, ethics, regulations, or generally accepted practices.
- Such environments usually require the ability to control actions of individuals.

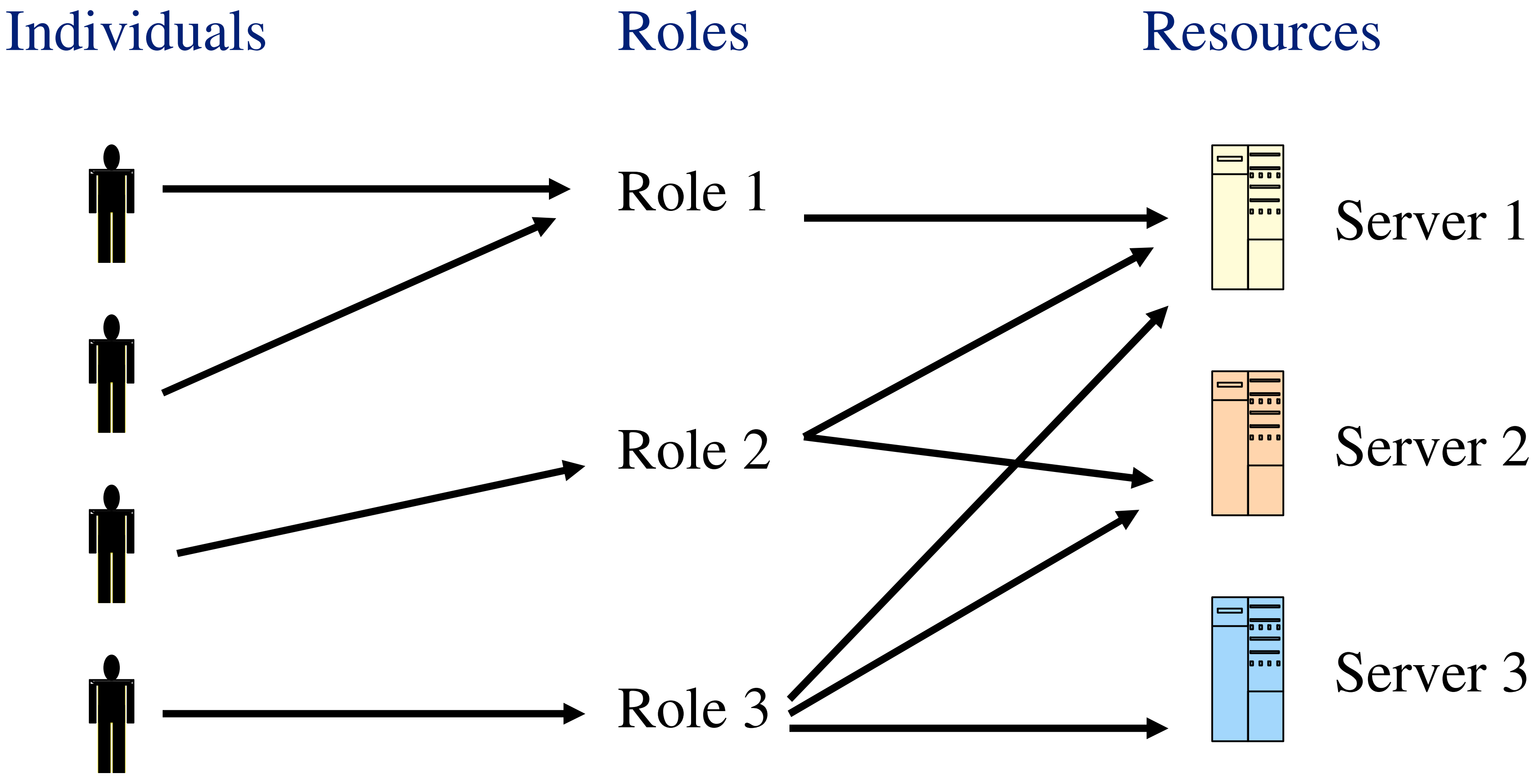
What is Role-Based Access Control?

- With role-based access control, access decisions are based on the roles that individual users have as part of an organization.
  - Users take on assigned roles (such as doctor, nurse, teller, manager).
- The process of defining roles
  - Based on a thorough analysis of how an organization operates and should include input from a wide spectrum of users in an organization.

Role-Based Access Control

- Role-Based Access Control emerged rapidly in the 1990s and it's adopted by most DBMS since then.
- Its basic concept is that privileges are associated with roles, and users are assigned to appropriate roles.
- Roles can then be granted to users and other roles.
- Roles can be created and destroyed using the **CREATE ROLE and DROP ROLE** commands.
- RBAC - a viable alternative to traditional discretionary and mandatory access controls
  - it ensures that only authorized users given access to certain data or resources.

Role-Based AC



User's change frequently, Roles don't

- Access rights are grouped by role name, and the use of resources is restricted to individuals authorized to assume the associated role.
  - For example, within a hospital system the role of doctor can include
    - operations to perform diagnosis, prescribe medication, and order laboratory tests; and
    - the role of researcher can be limited to gathering anonymous clinical information for studies.

Users and Roles

- Users are granted membership into roles based on their competencies and responsibilities in the organization.
- The operations that a user is permitted to perform are based on the user's role.
- User membership into roles can be revoked easily and new memberships established as job assignments dictate.
- Role associations can be established when new operations are instituted, and old operations can be deleted as organizational functions change and evolve.
  - This simplifies the administration and management of privileges
  - Roles can be updated without updating the privileges for every user on an individual basis.

Roles and Operations

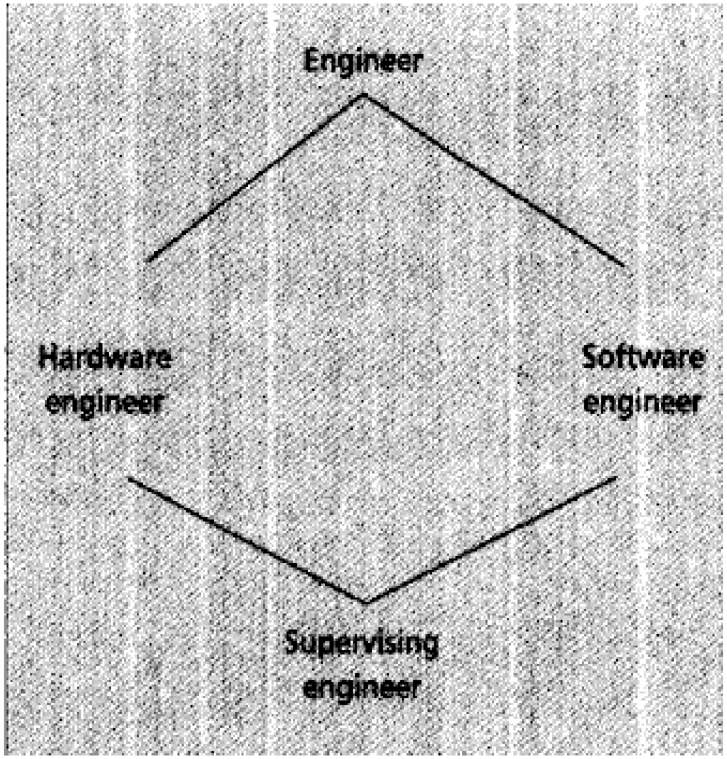
- Organizations can establish the rules for the association of operations with roles.
  - For example, a healthcare provider may decide that the role of clinician must be constrained to post only the results of certain tests but not to distribute them where routing and human errors could violate a patient's right to privacy.
- Operations can also be specified in a manner that can be used in the demonstration and enforcement of laws or regulations.
  - For example, a pharmacist can be provided with operations to dispense, but not to prescribe, medication.

- For example, there are differences between the access needs of a teller and an accounting supervisor in a bank.
  - An enterprise defines a teller role as being able to perform a savings deposit operation.
    - This requires read and write access to specific fields within a savings file.

  - An enterprise may also define an accounting supervisor role that is allowed to perform correction operations.
    - These operations require read and write access to the same fields of a savings file as the teller.
    - However, the accounting supervisor may not be allowed to initiate deposits or withdrawals but only perform corrections after the fact.
  - The teller is not allowed to perform any corrections once the transaction has been completed.
- The difference between the two roles is the operations that are executed by the different roles and the values that are written to the transaction log file.

Advantages of RBAC

- Hierarchical roles:
  - In many applications there is a natural hierarchy of roles
  - For example, the roles of hardware and software engineer are specializations of the engineer role.
  - A user assigned to the role of software engineer (or hardware engineer) will also inherit privileges assigned to the more general role of engineer.
  - The role of supervising engineer similarly inherits privileges from both software engineer and hardware engineer roles.



Advantages of RBAC

- **Authorization Management**
  - Two parts in specifying user authorizations
    - One which assigns users to roles
    - One which assigns access rights for objects to roles.
  - For instance, suppose a user responsibilities change say due to promotion
    - The user’s current roles can be taken away and new roles assigned as appropriate for the new responsibilities.
    - Assign new rights and revoke existing rights.

Countermeasures to database security threats

- Flow Control
- Encryption
- Access Control
- **Inference control**

- Access control mechanisms such as discretionary and mandatory access control, *prevent unauthorized direct access* to data.
  - However, they are unable to protect against *indirect data access*, when unauthorized information is obtained via inference channels.

Inference

- Inference occurs when users are able to piece together information at one security level to determine a fact that should be protected at a higher security level.

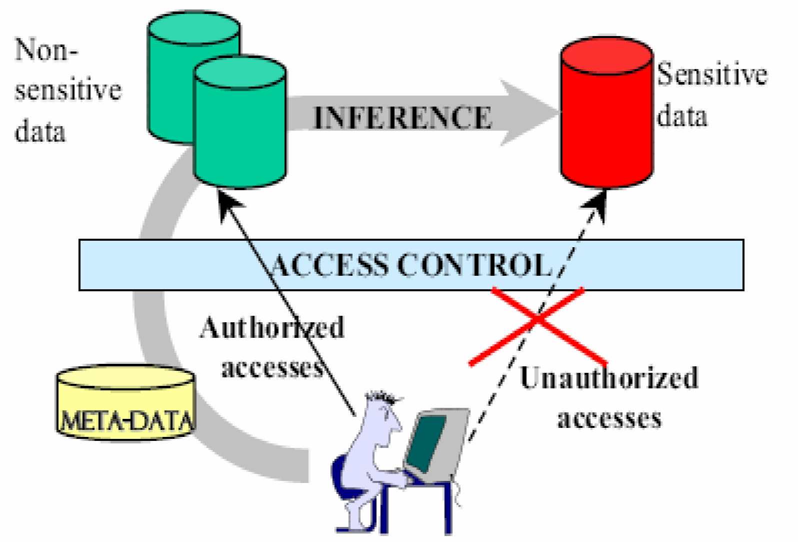
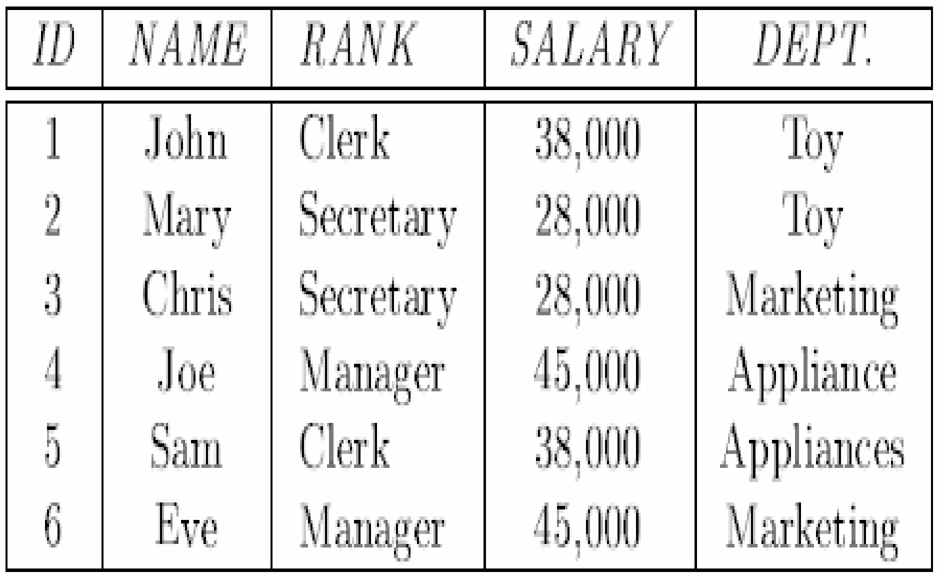


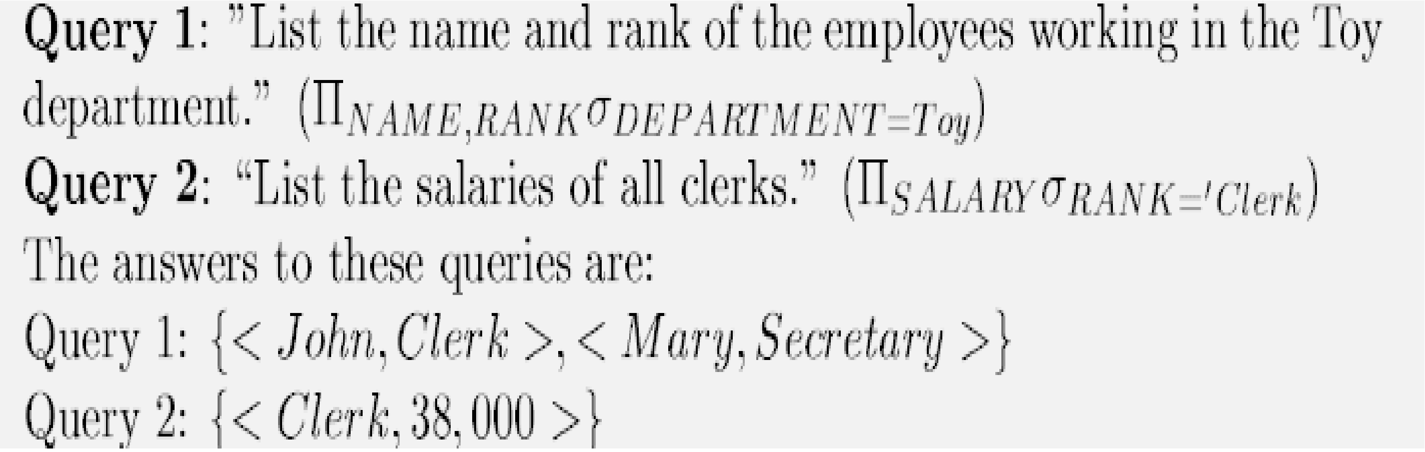
Figure 1: Indirect information access via inference channels



ID	*NAME*	*RANK*	*SALARY*	*DEPT.*
1	John	Clerk	38,000	Toy
2	Mary	Secretary	28,000	Toy
3	Chris	Secretary	28,000	Marketing
4	Joe	Manager	45,000	Appliance
5	Sam	Clerk	38,000	Appliances
6	Eve	Manager	45,000	Marketing

Original

- Employees’ salaries should be kept confidential
- NAME and SALARY can only be accessed by authorized users.
- To increase data availability, unauthorized users are allowed to access values for NAME and SALARY separately.
- Suppose an unauthorized user submits the following two queries:



Query 1: "List the name and rank of the employees working in the Toy department." $(\Pi_{NAME,RANK}\sigma_{DEPARTMENT=Toy})$

Query 2: “List the salaries of all clerks.” $(\Pi_{SALARY}\sigma_{RANK='Clerk})$

The answers to these queries are:

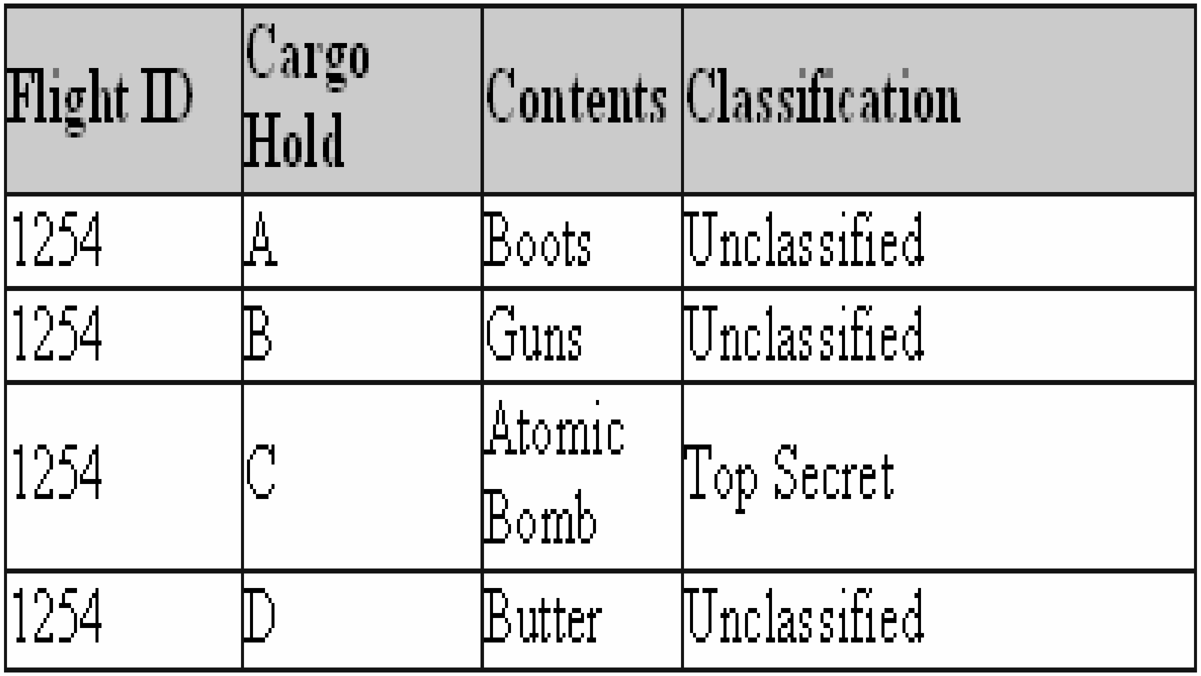
Query 1: $\{<John, Clerk>, <Mary, Secretary>\}$

Query 2: $\{<Clerk, 38,000>\}$

•The answers reveal that John’s salary is $38,000.

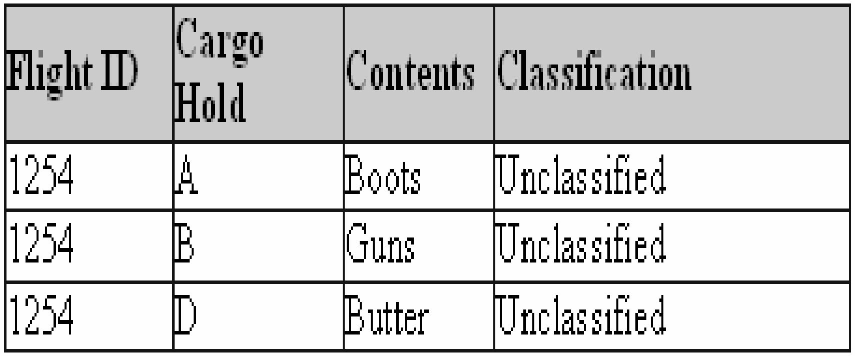
Military transportation system

- 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.
- The flight identification number may be cross-referenced with other tables to determine the origin, destination, flight time and similar data.



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 see all four shipments.
- 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

- When Private Jones sees that nothing is scheduled for hold C on flight 1254, he might attempt to insert a new record to transport some vegetables on that flight.
- However, when he attempts to insert the record, his insert will fail due to the unique constraint.
- At this point, Private Jones has all the data he needs to infer that there is a secret shipment on flight 1254.
- He could then cross-reference the flight information table to find out the source and destination of the secret shipment and various other information.

- **What can you do about inference?**
- Include the classification column in the unique constraint.
  - This technique, known as **polyinstantiation**
    - Polyinstantiation is the ability of a database to maintain multiple records with the same key to prevent inference attacks.
  - Private Jones would never learn of the Top Secret shipment.

How do polyinstantiated elements arise ?

- A subject updates what appears a null element in a tuple, but which actually hides data with a higher (or incomparable) security level
- Problem:
  - Subject cannot be informed about existence of higher security level data
  - Overwriting the old value allows “low” users to unintentionally destroy “high” data
- Insertion must be accepted

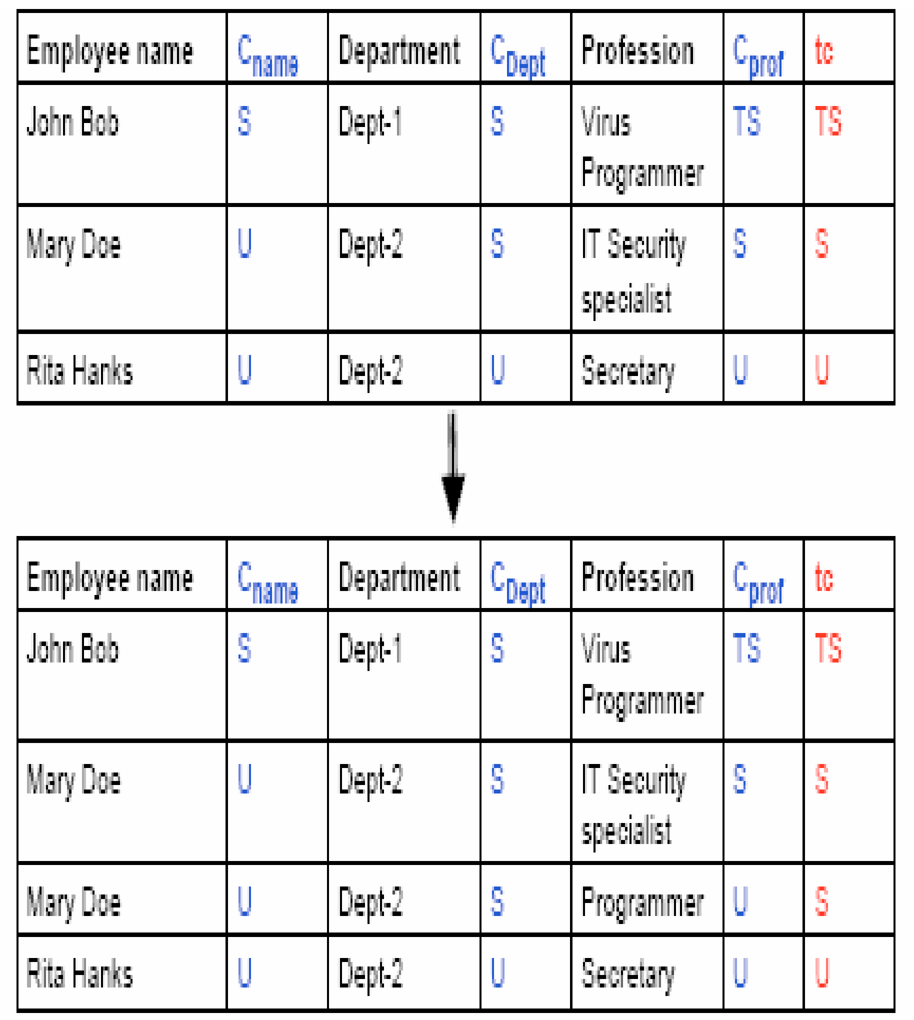
- **Polyinstantiation:**
  - Several tuples might exist for the same primary key
- **Polyinstantiated elements:**
  - Elements of an attribute which have different security levels, but are associated with the same primary key.

Polyinstantiation- Example

Primary key: Employee Name

Unclassified Subject requests the following operation:

Update employee
SET profession = “Programmer”
WHERE name = “Mary Doe”



Employee name	C_{name}	Department	C_{Dept}	Profession	C_{prof}	tc
John Bob	S	Dept-1	S	Virus Programmer	TS	TS
Mary Doe	U	Dept-2	S	IT Security specialist	S	S
Rita Hanks	U	Dept-2	U	Secretary	U	U

Employee name	C_{name}	Department	C_{Dept}	Profession	C_{prof}	tc
John Bob	S	Dept-1	S	Virus Programmer	TS	TS
Mary Doe	U	Dept-2	S	IT Security specialist	S	S
Mary Doe	U	Dept-2	S	Programmer	U	S
Rita Hanks	U	Dept-2	U	Secretary	U	U

Overview of Virtual Private Databases

SQL Views

- A view can be thought of as either a virtual table or a stored query.
- The data accessible through a view is not stored in the database as a distinct object.
- The result set of the SELECT statement forms the virtual table returned by the view.
- A user can use this virtual table by referencing the view name

A view is used to do any or all of these functions

- Restrict a user to specific rows in a table.
  - For example, allow an employee to see only the rows recording his or her work in a labor-tracking table.
- Restrict a user to specific columns.
  - For example, allow employees who do not work in payroll to see the name, office, work phone, and department columns in an employee table, but do not allow them to see any columns with salary information or personal information.
- Join columns from multiple tables so that they look like a single table.
- Aggregate information instead of supplying details.
  - For example, present the sum of a column, or the maximum or minimum value from a column.

- Views are created by defining the SELECT statement that retrieves the data to be presented by the view.
- The data tables referenced by the SELECT statement are known as the base tables for the view.
- You can then reference v_customer in statements in the same way you would reference a table:
  - SELECT * FROM V_Customer

```
CREATE VIEW V_Customer
AS SELECT First_Name,
Last_Name, Country
FROM Customer
```

VPD

Virtual Private Database

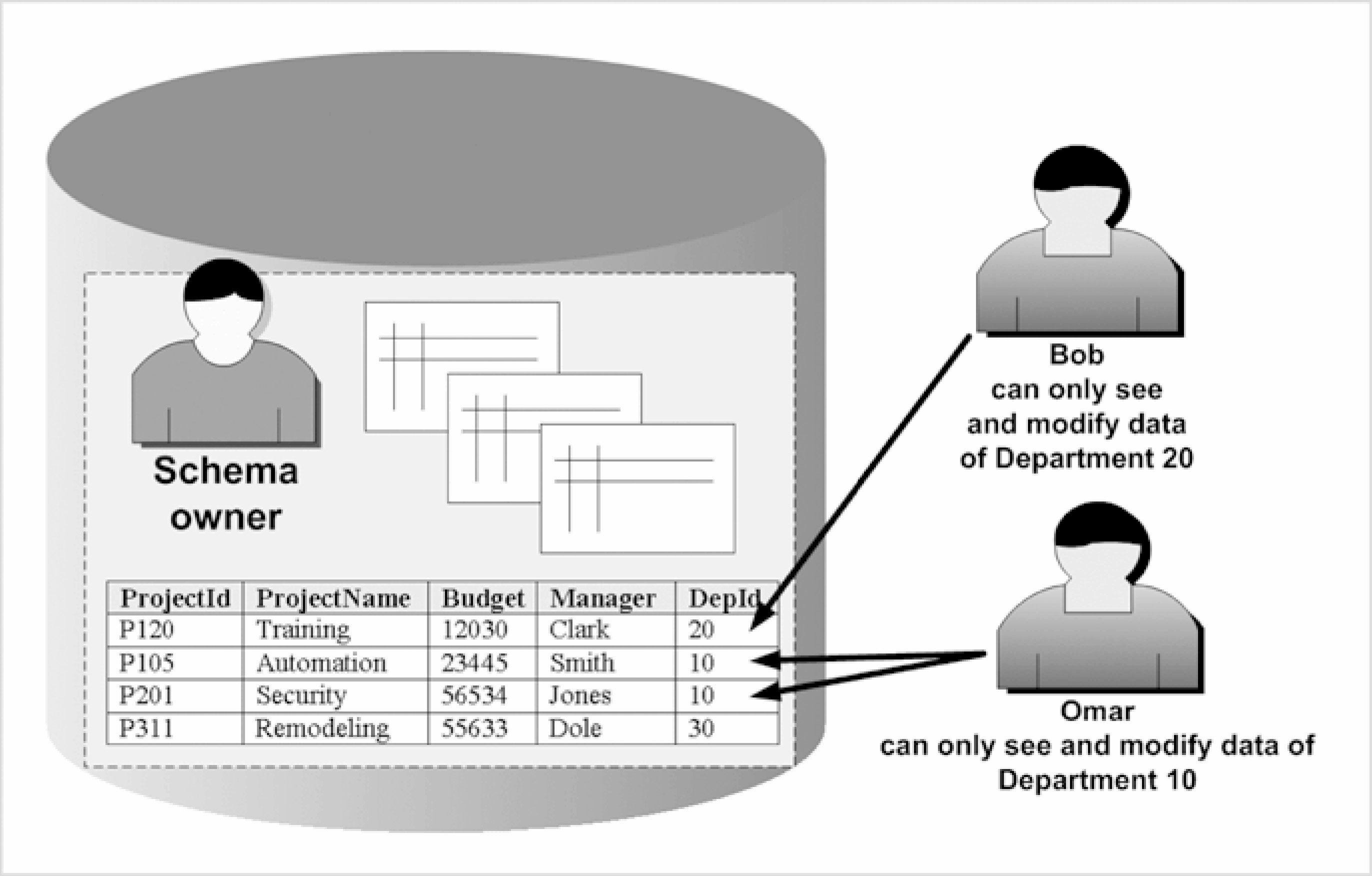
Why VPD?

- Scalability
  - Table Customers contains 1,000 customer records. Suppose we want customers to access their own records only. Using views, we need to create 1,000 views. Using VPD, it can be done with a single policy function.
- Simplicity
  - Say, we have a table T and many views are based on T. Suppose we want to restrict access to some information in T.
  - Without VPD, all view definitions have to be changed.
  - Using VPD, it can be done by attaching a policy function to T
    - as the policy is enforced in T, the policy is also enforced for all the views that are based on T.
- Security
  - Because the VPD provides server-enforced security, it cannot be bypassed by users accessing data directly, or using another application
    - Server enforces the security policy and controls access to information.

What is VPD?

- Protect confidential and secret information
- Virtual Private Databases (VPD) allow multiple users to access a single schema whilst preventing them from accessing data that is not relevant to them
- Sometimes referred to as Oracle Row-Level Security (RLS) or Fine Grained Access Control (FGAC)
  - Allows to define which rows users may have access to

Virtual Private Databases



ProjectId	ProjectName	Budget	Manager	DepId
P120	Training	12030	Clark	20
P105	Automation	23445	Smith	10
P201	Security	56534	Jones	10
P311	Remodeling	55633	Dole	30

FIGURE 6-2 Virtual private database example

How does it work?

- When a user accesses a table which is protected by a VPD policy (function),
  - The Oracle server invokes the policy function.
  - The policy function returns a predicate, based on session attributes or database contents.
  - The server dynamically rewrites the submitted query by appending the returned predicate to the WHERE clause.
  - The modified SQL query is executed.

Example

- Suppose Alice has the following table.
  - my_table(owner varchar2 (30), data varchar2(30));
- Users can access only the data of their own.
- But Admin should be able to access any data without restrictions.

Create a policy function

```
Create function sec_function(p_schema varchar2, p_obj varchar2)
Return varchar2
As
    user VARCHAR2(100);
Begin
    if ( SYS_CONTEXT (‘userenv’, ‘ISDBA’) ) then
            return ‘ ’;
    else
            user := SYS_CONTEXT (‘userenv’, ‘SESSION_USER’);
            return ‘owner =‘|| user;
    end if;
End;
```

SYS_CONTEXT returns the value of *parameter* associated with the context *namespace*. *If the namespace of 'USERENV' is used, attributes describing the current Oracle session can be returned.*
ISDBA returns 'TRUE' if the user has been authenticated as having DBA privileges
SESSION_USER returns the user name of the current context in the current database.

Attach the policy function to my_table

```
execute dbms_rls.add_policy (object_schema => ‘Alice’,
          object_name => ‘my_table’,
          policy_name => ‘my_policy’,
          function_schema => ‘Alice’,
          policy_function => ‘sec_function’,
          statement_types => ‘select, update, insert’);
```

- object_schema => ‘Alice’ : Specifies the schema that you want to protect.
- object_name => ‘my_table : Specifies the object within the schema to protect.
- policy_name => ‘my_policy : Names this policy my_policy.
- function_schema => ‘Alice’ : Specifies the schema in which the sec_function was created.
- policy_function => ‘sec_function : Specifies a function to enforce the policy.
- statement_types => ‘select, update, insert : Specifies the operations to which the policy applies.
  - In this example, the policy applies to all SELECT, INSERT, UPDATE, statements the user may perform.

Bob accesses my_table

```
select * from my_table;
    => select * from my_table where owner = ‘bob’;
    : only shows the rows that owner is ‘bob’
insert into my_table values(‘Some data’, ‘bob’); OK!
insert into my_table values(‘Other data’, ‘alice’); NOT OK!
    = because of the check option.
```

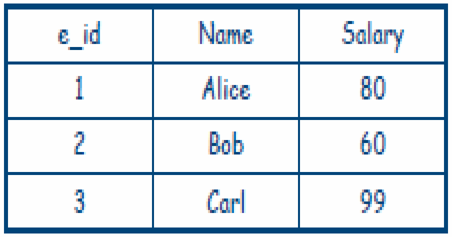
Column-level VPD

- Instead of attaching a policy to a whole table or a view, attach a policy only to security-relevant columns
  - *Default behavior*: restricts the number of rows returned by a query.
  - *Masking behavior*: returns all rows, but returns NULL values for the columns that contain sensitive information.

Column-level VPD: Example

Suppose Alice has the following table.

Employees(e_id number(2), name varchar2(10), salary number(3));



e_id	Name	Salary
1	Alice	80
2	Bob	60
3	Carl	99

Users can access e_id’s and names without any restriction. But users can access only their own salary information.

Create a policy function

```
Create function sec_function(p_schema varchar2,
   p_obj varchar2)
Return varchar2
As
    user VARCHAR2(100);
Begin
    user := SYS_CONTEXT(‘userenv’, ‘SESSION_USER’);
    return ‘name = ‘ || user;
    end if;
End;
```

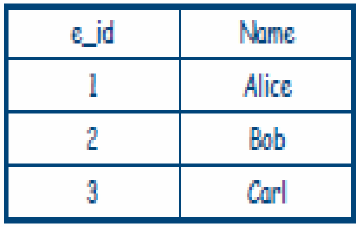
Attach the policy function to Employees (default behavior)

```
execute dbms_rls.add_policy
    (object_schema => ‘Alice’,
    object_name => ‘employees’,
    policy_name => ‘my_policy’,
    function_schema => ‘Alice’,
    policy_function => ‘sec_function’,
    sec_relevant_cols=>’salary’);
```

sec_relevant_cols allows you to display all rows but hide the restricted rows containing values of the specified columns

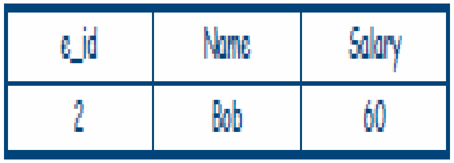
Bob accesses table Employees (default behavior)

select e_id, name from Employee;



e_id	Name
1	Alice
2	Bob
3	Carl

select e_id, name, salary from Employee;



e_id	Name	Salary
2	Bob	60

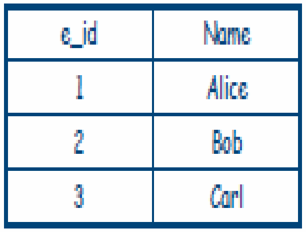
Attach the policy function to Employees (masking behavior)

```
execute dbms_rls.add_policy (object_schema => ‘Alice’,
    object_name => ‘employees’,
    policy_name => ‘my_policy’,
    function_schema => ‘Alice’,
    policy_function => ‘sec_function’,
    sec_relevant_cols=>’salary’,
    sec_relevant_cols_opt=>dbms_rls.ALL_ROWS);
```

Column masking behaviour is implemented by using the "sec_relevant_cols_opt => DBMS_RLS.ALL_ROWS" parameter. This allows you to display all rows but mask the values of the specified columns for the restricted row

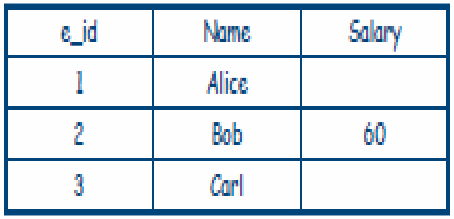
Bob accesses table Employees (masking behavior)

select e_id, name from Employee;



e_id	Name
1	Alice
2	Bob
3	Carl

select e_id, name, salary from Employee;



e_id	Name	Salary
1	Alice	
2	Bob	60
3	Carl	

Statistical Database Security

- A **statistical database (SDB)** typically contains information about n individuals where n is very large.
- A **statistical database** system gives users the ability to both obtain **statistical** information (like average, median, count) and preserve the privacy of any individual.
- Examples include census and medical databases.

- Many government agencies, businesses, and nonprofit organizations need to collect, analyze, and report data about individuals in order to support their short-term and long-term planning activities.
- SDBs contain confidential information such as income, credit ratings, type of disease, or test scores of individuals.

- A hospital database
  - In the hospital environment, physicians may be given access to patients' entire medical records,
  - whereas statistical researchers may only be allowed to obtain aggregate statistics for subsets of the patient population.

Accuracy vs. Confidentiality

Accuracy –

Researchers want to extract accurate and meaningful data

Confidentiality –

Patients, laws and database administrators want to maintain the privacy of patients and the confidentiality of their information

- The system should be secure enough to guard against a user’s ability to infer any confidential information related to a specific individual represented in the database.

- Example of a hospital database
- The database contains data about patients:
  - (Age, Sex, Employer, Social Security Number, Diagnosis Type)

- **In the hospital environment:**
  - A subset of patients whose data are included in the computation of the response to a query is referred to as the query set.
  - Statistics are calculated for subsets of patients having common attribute values (e.g., Age = 42 and Sex = male).
  - Such a subset can be specified by a characteristic formula, C

- For example, C = (Age = 42) & (Sex = Male) & (Employer = ABC)
  - is a characteristic formula that specifies the subset of male patients, age 42, employed by the ABC company.

- Suppose there is a malicious researcher who wants to obtain information about the diagnosis type of a given patient, Mr. X.
- A malicious user who wants to compromise the database is referred to as a **snooper**.
- Assume that the snooper knows the age and employer of Mr. X.
  - He can then issue the query
    - QI: COUNT (Age = 42) & (Sex = Male) & (Employer = ABC).

- If the answer is 1, the snooper has located Mr. X and can then issue such queries as
- Q2: COUNT (Age = 42) & (Sex = Male) & (Employer = ABC) & (Diagnosis Type = Schizophrenia).
  - If the answer to Q2 is 1, the database is said to be positively compromised and the user is able to infer that Mr. X has the diagnosis type schizophrenia.
  - If the answer is 0, the database is said to be *partially compromised*, because the user was able to infer that the diagnosis type of Mr. X is not schizophrenia.

- Partial compromise refers to the situation in which some inference about a confidential attribute of an entity can be made, even if the exact value cannot be determined.

Statistical Database Security

- Focuses on the protection of confidential individual values stored in ***statistical databases*** and used for statistical purposes.
  - Example
    - Detailed phone call records, statistically analyzed by phone companies in order to improve their services.

Security in Statistical DBs

Goal:

- Allow arbitrary *aggregate* SQL queries
- Hide confidential data

```
SELECT name
FROM    Patients
WHERE age=42
        and sex=‘M’
        and diagnostic=‘schizophrenia’
```

Not OK

```
SELECT count(*)
FROM    Patients
WHERE age=42
        and sex=‘M’
        and diagnostic=‘schizophrenia’
```

OK

Types of Statistical Databases

- **Static** – a static database is made once and never changes
- Example: U.S. Census

- **Dynamic** – changes continuously to reflect real-time data
- Example: most online research databases

Security Methods

- Access Restriction
- Microaggregation
- Data Perturbation
- Output Perturbation
- Auditing

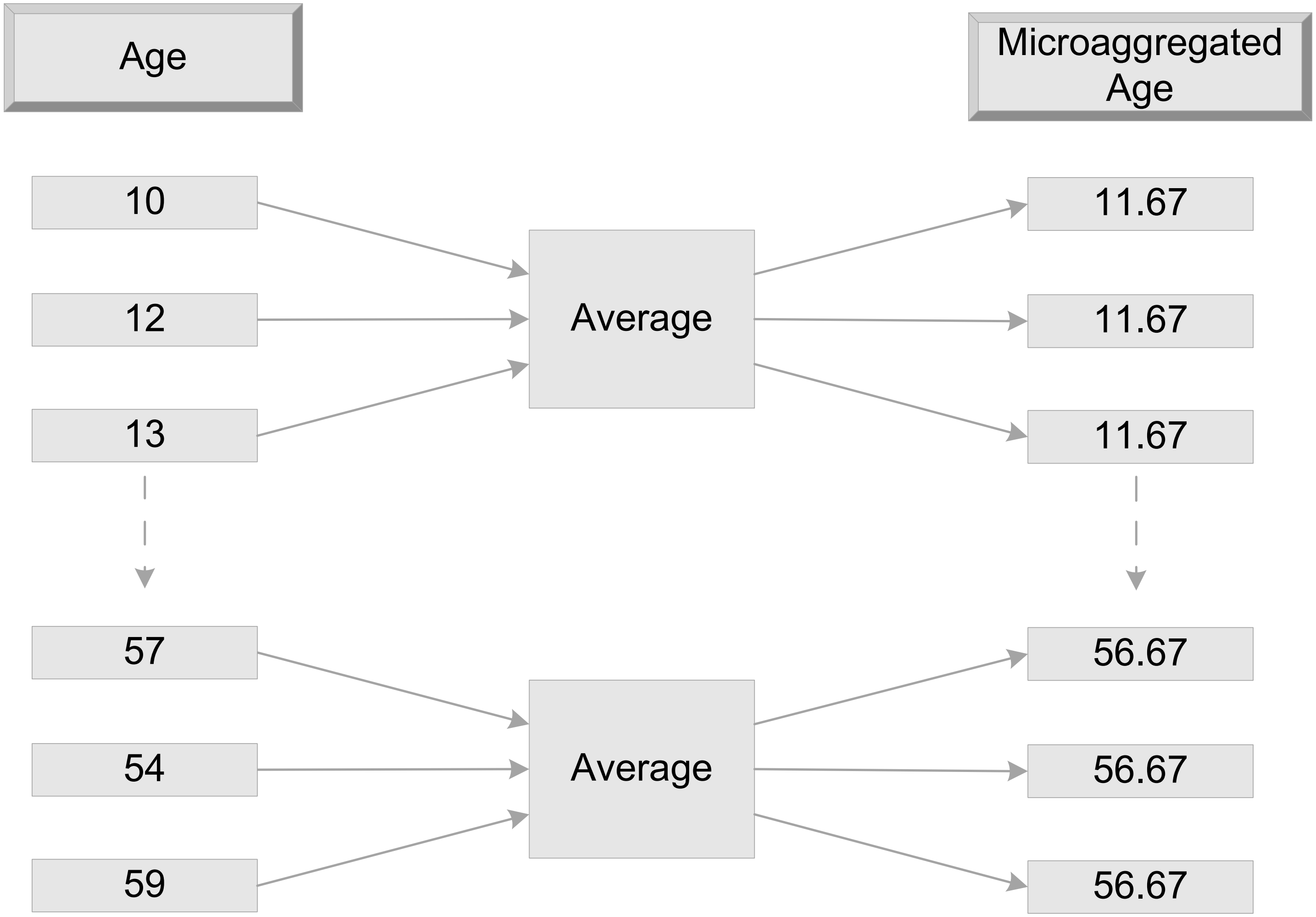
Access Restriction

- Databases normally have different access levels for different types of users
- User ID and passwords are the most common methods for restricting access
  - In a medical database:
    - Doctors/Healthcare Representative – full access to information
    - Researchers – only access to partial information (e.g. aggregate information)

Microaggregation

- Raw (individual) data is grouped into small aggregates before publication
- The average value of the group replaces each value of the individual
- Data with the most similarities are grouped together to maintain data accuracy
- Helps to prevent disclosure of individual data

Microaggregation



Age

10

12

13

57

54

59

Average

Average

Microaggregated Age

11.67

11.67

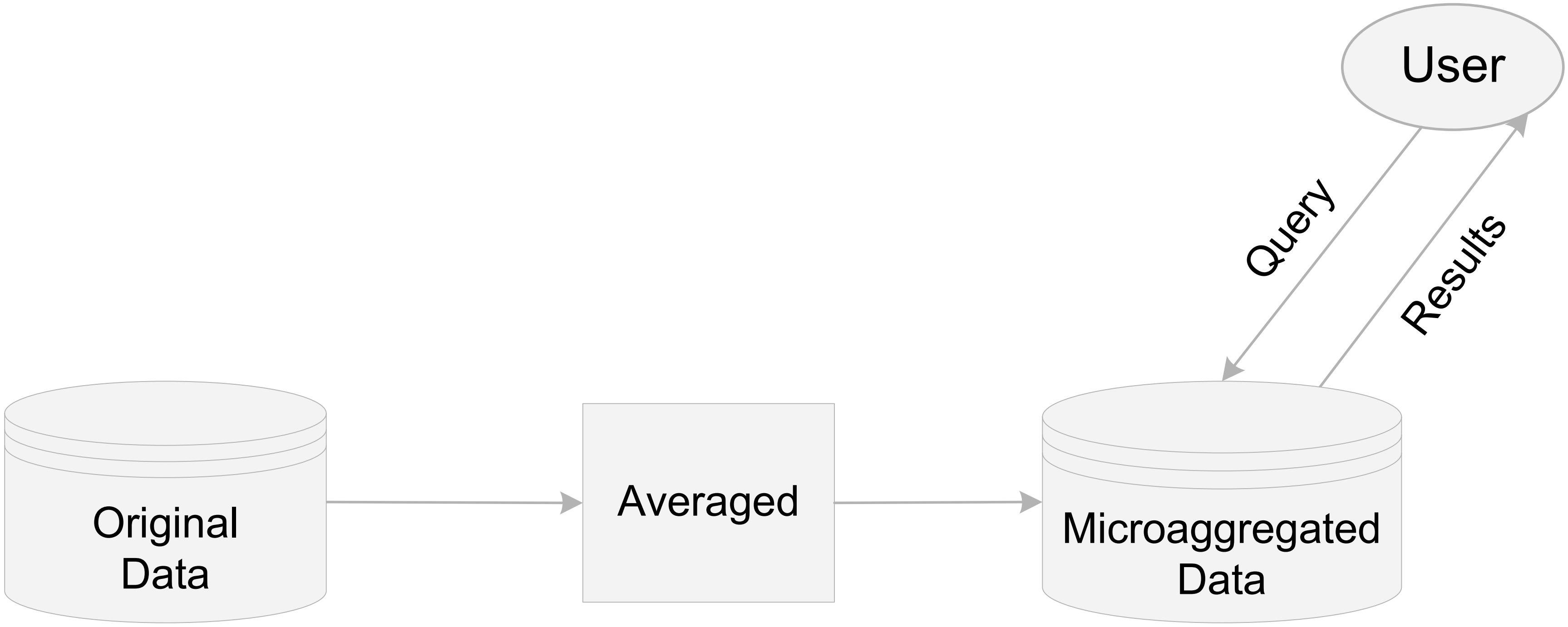
11.67

56.67

56.67

56.67

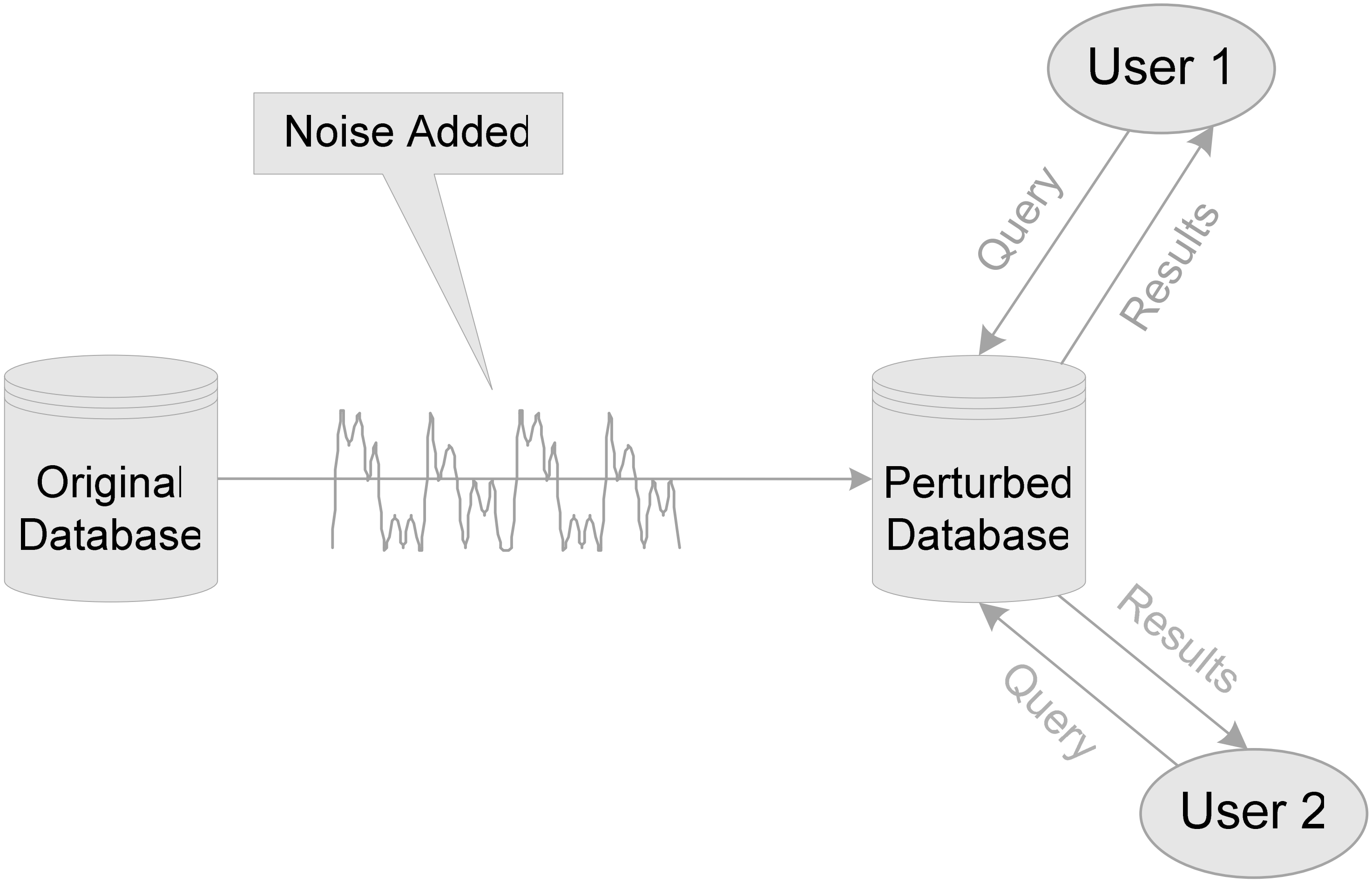
Microaggregation



Data Perturbation

- Perturbed data is raw data with noise added
- **Pro**: With perturbed databases, if unauthorized data is accessed, the true value is not disclosed
- **Con**: Data perturbation runs the risk of presenting biased data

Data Perturbation



Noise Added

Original Database

Perturbed Database

User 1

User 2

Query

Results

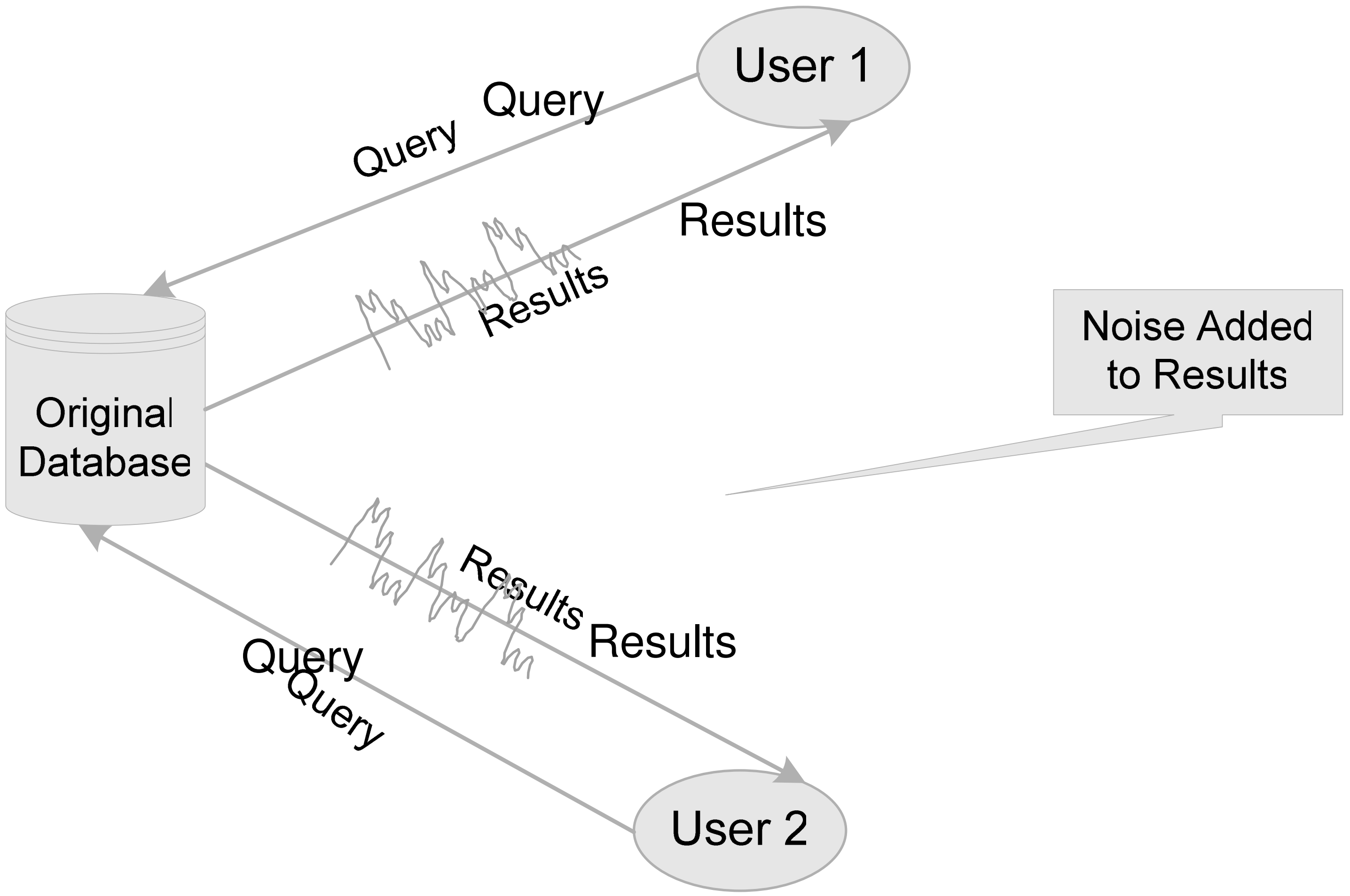
Query

Results

Output Perturbation

- Instead of the raw data being transformed as in Data Perturbation, only the output or query results are perturbed
- The bias problem is less severe than with data perturbation

Output Perturbation



Auditing

- Auditing is the process of keeping track of all queries made by each user
- Usually done with up-to-date logs
- Each time a user issues a query, the log is checked to see if the user is querying the database maliciously

Cryptography?