Q20. What is the purpose of checkpointing techniques ?
Q21. What are the steps performed during checkpointing?
Q22. How does the frequency of checkpointing affect :
a) System performance when no failure occurs ?
b) Time it takes to recover from system crash ?
Q23. What do you understand by Transaction Rollback ?
Q24. Why is backward scanning preferred in transaction rollback instead of forward scanning ?
Q25. Explain the different recovery techniques ?
Q26. What do you mean by deferred update technique ? How is it performed ? Give proper examples ?
Q27. What do you mean by Immediate update technique ? How is it performed ? Give proper examples ?
Q28. Explain the Undo/Redo Logging Scheme ?
Q29. What is Shadow Paging ? How it is performed ?
Q30. What are the advantages and disadvantages of Shadow Paging ?
Q31. How are databases recovered from catastrophic failures ?
Q32. What is the difference between Current Page Table and Shadow Page Table ?
Q33. Compare the Shadow Paging Recovery Scheme with the log based recovery schemes in terms of ease of implementation and overhead cost ?
Q34. What is meant by forward and backward recovery technique ?
Q35. Explain the recovery procedure that needs to take place after a disk crash ?
Q36. What do understand by term "Fuzzy Checkpoints?
Q37. Explain undo/redo logging recovery for the following log as it appears at three instances of time :

[Start_transaction, T1]	[Start_transaction, T1]	[Start_transaction, T1]
[Write, T1, A, 4, 5]	[Write, T1, A, 4, 5]	[Write, T1, A, 4, 5]
[Start_transaction, T2]	[Start_transaction, T2]	[Start_transaction, T2]
[Commit, T1]	[Commit, T1]	[Commit, T1]
[Write, T2, B, 9, 10]	[Write, T2, B, 9, 10]	[Write, T2, B, 9, 10]
	[Checkpoint_start, T2]	[Checkpoint_start, T2]
	[Write, T2, C, 14, 15]	[Write, T2, C, 14, 15]
	[Start_transaction, T3]	[Start_transaction, T3]
	System Failure	System Failure
		[Checkpoint end]
		[Commit, T2]
		System Failure

UNIT 9

DATABASE SECURITY AND INTEGRITY

Data is a valuable resource for an organization and therefore should be kept secure and confidential. The database security is a crucial issue in the database management. The term "security" refers to the protection of the database against any unauthorized access that may be either intentional or accidental. Security in a database involves both policies and mechanisms that protect data from any unauthorized access.

The need for database security has often being neglected in the past but now with the use of databases in multiuser environments it has become a necessity.

Most of the database management systems are operated by multiple concurrent users. Some of them may be adding some new information and some of them may be deleting the existing information. Access by an unauthorized user might corrupt the database and make the database incorrect. So this may lead to the following questions :-

- How does DBMS recognize these users ?
- How will it distinguish authorized users from the unauthorized users ?
- How will it reject the request made by unauthorized users?

To answer all these questions, DBMS provides the security mechanism which involves allowing and disallowing them from performing actions on the database.

Integrity implies that any authorized access, updation, or deletion of data in the database doesn't change the validity of the data. In other words, Integrity ensures that changes made to the database by authorized users don't result in loss of data consistency.

Data security and data integrity concepts though distinct, but are related. Security refers to protection of data against unauthorized access, alteration or deletion. Integrity refers to accuracy or validity of that data.

To sum up,

- Security means protecting data against unauthorized users.
- Integrity means protecting data against authorized users.

9.1 SECURITY AND INTEGRITY THREATS

Database often lie within the organization's boundary so it is the foremost duty of the organization to protect its data from unauthorized users. An organization should identify all the risks factors and weak elements from the database security point of view. It should also be able to find solutions to handle all possible threats.

A *threat* is any situation, event or personnel that will adversely effect the database security and smooth and efficient functioning of the organization. Threat to database can be intentional or accidental. People can exploit loopholes or abuse privileges to intentionally or maliciously gain access to the confidential data.

Attempts have been made to protect the data through the use of firewalls and data encryption. So due to this organizations have not considered database security as a serious issue. It is a serious misconception that causes substantial financial and reputational loss. Access control becomes more important is an environment, where data resources are shared and not all users are privileged to access and modify all data. Shareability demands some mechanism to control who does what to what data. Access control mechanism will not avoid all unauthorized accesses. Infact over half of all reported abuse involving a computer resulted from insiders abusing their access lights. Hence it is important to control internal security breaches.

Given below are some database security threats.

- **Data Tampering** - Confidentiality and Integrity of information during transmission is essential and vital. Distributed environment bring with them the possibility that an unauthorized third party can perform a computer crime by tampering with data as it moves between site. In a data modification attack, an unauthorized party on the network intercepts data in transit and changes parts of that data before retransmitting it.
- **Eavesdropping and data theft** - As stated earlier, data must be stored and transmitted securely over the Internet and in WAN's environment. Both public carriers and private network owners often root portions of their network through insecure landlines, extremely vulnerable satellite links or a number of servers. Network taps throughout the buildings are live and unprotected. An attacker with a laptop computer can easily penetrate and monitor the network. The company that has no controls or policies on modems thus allow any user to setup a private PPP connection to bypass the firewall.
- **Falsifying User's Identities** - In a distributed environment it becomes increasingly possible for a user to counterfeit an identity to gain access to sensitive and important information. In addition, criminals can hijack connections. Identity theft is becoming a greatest threat to individuals in the Internet environment. Non repudiation is another concern i.e. how can a person's digital signature be protected.
- **Password related threats** - In large systems, user must remember multiple passwords for different services that they are entitled to use. Users typically respond to the problem of managing multiple passwords by selecting easy to guess passwords such as a name, a fictional character or a word found in a dictionary. Users choose the standardized passwords so that they are the same on all machines or websites and sometimes goto the extent of writing them down where an attacker can easily find them.

  All of these strategies sacrifice password secrecy and service availability.
- **Unauthorized access to data** - Database may contain confidential tables or confidential columns in a table which should not be available indiscriminately to all users who are authorized to access the database. It should be possible to protect data on a column level, Similarly, certain data rows may contain confidential information which should not be available indiscriminately to users who are authorized to access the table.
- **Lack of accountability :** There must be some reliable way to monitor users. In large scale environments, the burden of managing user accounts and passwords makes the system vulnerable to errors and attacks. Appropriate logs must enabled to monitor accountability. These problems become particularly complex in a multitier system.

9.2 DEFENCE MECHANISMS

Security considerations not only apply to data held in the database, but others parts of the system may be effected , which in turn could effect the database.

Security therefore encompasses hardware, software, people and data. The need for security often neglected in the past is increasingly recognized by the organizations, because of increasing amount of data stored and acceptance that loss or unavailability could prove disastrous.

Generally four levels of defence are recognized for database security which are given below.

- **Physical security** - Physical security mechanisms include appropriate locks and keys and entry logs to computing facility and terminals. Security of physical storage devices such as hard disks, magnetic tapes etc. must be maintained in case of power failures, natural disasters and sabotage by antisocial elements. Also, the computers and other equipments must be physically unaccessible to unauthorized users.

  User Identification and passwords have to be kept confidential otherwise unauthorized users can steal the passwords of a more privileged user and can make malicious changes in some portion of the database.
- **Human Factors** - Since a large number of people (such as database administrators, network administrator, application administrators, developers, security officers, database users etc.) interact with the database so there is a possibility that they may corrupt the database. So a formal clearance procedure should be performed on such persons dealing with the sensitive data by testing their reliability and trust they have earned while working in the previous and the present organization. Users must be authorized carefully to reduce the change of any user giving access to an intruder in exchange for a bribe or other favors .

- **Operating system** - Security of the operating system is entirely dependent on the platform it is running. It is extremely important to secure the OS from attacks. If the operating system is not secure then hackers could cause damage to the database files or configuration files of the database. So if the operating system used for handling the database is not provided proper security then it may serve as a means of unauthorized access to the database no matter how secure the database system is. Proper user identification and passwords should be provided by the operating system so that only the authorized users can access the database using the given operating system.
- **Database system** - In the database system, a large number of users access the data of the database, so what information will be accessible to what class of users and the type of the access that will be allowed to this class (i.e. whether he/she can insert, delete, modify and view data) is managed by the DBMS. It is the responsibility of the database management system to check that these restrictions should not be violated.

9.3 DATA SECURITY REQUIREMENTS

Data is the most important part of the organization so we need to secure it from the various types of threats, data tampering, eavesdropping and data theft etc. Appropriate technologies are used to resolve these security issues. The basic security standards which technologies can ensure are confidentiality, integrity and availability.

1. **CONFIDENTIALITY** - A secure system ensures the confidentiality of data. This means that users are provided access to specific type of confidential information as part of the performance of their assigned duties as well as specifying the types of actions that are prohibited. Confidentiality have several aspects which are discussed below:

- **Access control** - An important aspect of database security is who can access what data. Access control is the ability to hide portions of the database so that access to a data doesn't become an all or nothing proposition. Access to data is controlled by means of privileges, roles and user accounts.

  A *privilege* is a permission to access a named object in a prescribed manner. Privileges are grouped to form a role which can be then assigned to users. For example : A clerk in the HR department might need some access to the EMPLOYEE table but should not be allowed to access salary information for the entire company.

  An important issue is that a user must be given atleast privilege required to accomplish a task, unnecessary privileges given to the user must be revoked.
- **Authenticated users** - Before making any request for the data the user has to identify herself/himself to the system and authenticate the identification to conform that the user is infact the correct person. So authentication is a way of implementing decisions of whom to trust. The following methods can be used for the purpose of authentication.

a) Something known only by the user (passwords etc.)

b) Something which only user processes (Encoded card, key etc.)

c) Physical characteristics of the user (Finger prints, Voice prints etc.)

- **Secure storage of sensitive data** - Once the data has been collected you must ensure that it should remain private i.e. once confidential data is entered, its integrity and privacy must be protected on the databases and servers where it resides. It is extremely important to secure the storage media on which the sensitive data is stored, to be prevented from attacks, otherwise the hackers could cause damage to the sensitive data.
- **Privacy of communication** - The DBMS should be capable of controlling the spread of confidential personal information from unauthorized people such as credit cards, employment information etc. It should also keep the corporate data such as proprietary information about products, competitive analysis, trade secrets, marketing and sales plans away from the unauthorized people.

2. **INTEGRITY** - Integrity contributes to maintaining a secure database by preventing the data from becoming invalid and giving misleading results. i.e. data should be protected from corruption and deletion both while it resides within the database and while it is being transmitted over the network. It consists of the following aspects.

- System and object privileges control access to application tables and system commands so that only authorized users can change data. It is the responsibility of the DBA to grant privileges to the users who need to use the system and classify users and data in accordance with the policy of the organization.
- Integrity constraints are applied to maintain the correctness and validity of the data in the database.
- Database must be protected from viruses. So proper technologies such as firewalls, antiviruses etc. should be used to handle these problems.
- Ensure that access to the network is controlled and data is not vulnerable to attacks during transmission across the network.

3. **AVAILABILITY** - Data should always be made available for the authorized user by the secure system without any delays. Availability is often thought of as a continuity of service assuring that database is available. Denial of service attacks are attempts to block authorized users ability to access and use the system when needed. It has a number of aspects.

- **Ease of use** - Security should be managed properly. Resources managed by the users for working with databases should be effectively managed so that database is available all the time to all valid users. If the resources are not used effectively, a certain increase in database usage might crash the database leading to denial of service to valid users also.
- **Flexibility** - Administrators must have all the relevant tools necessary for managing the user population (Information about the users and their access rights etc.). For this purpose, the administrator use the directory.
- **Scalibility** - The system performance should not be effected with the increase of the number of users or processes which require services from the system .Data should be

available at all the time without any degradation in the performance of the system on increase in the number of users.

- **Resistance** - There must be facilities available within the database to prohibit run away queries. For this purpose, user profiles must be defined and the resources used by any user should be limited. In this way, system can be protected from the users consuming too much memory so that other users can easily do their work.

9.4 DISCRETIONARY AND MANDATORY ACCESS CONTROL

In a multiuser database system, the DBMS must provide techniques to enable certain users or users groups to access selected portions of the database without gaining access to the rest of the database. For example, sensitive information such as employee salaries should be kept confidential from most of the database system users and a DBMS typically includes a database security and authorization subsystem for this purpose.

Two approaches are used for database security known as *discretionary* and *mandatory* access control.

Discretionary access control is used to grant privileges to users including the capability to access specific data files or fields in a specific mode. For example, Read, insert, delete and update. In case of discretionary access control a given user typically has different access rights on different objects. Thus discretionary schemes are very flexible.

Mandatory access control is used to enforce multilevel security by classifying the data and users into various security classes. For example, to permit users at a certain clearance level see only a certain type of data. This control mechanism is applicable to databases in which the data has a rather static and rigid classification structure like in certain military and government organizations.

These access control security mechanism helps to protect the data in the database from unauthorized users.

9.5 PROTECTING THE DATA WITHIN THE DATABASE

Confidentiality, integrity and availability are the hallmarks of the database security. The following questions arises when you protect the data within the database.

1. Who should have rights to access data ?
2. What portion of the data should a particular user be able to access?
3. What operations should an authorized user be able to perform on data.?

The answers to above questions leads to the concept of authorization.

9.5.1 AUTHORIZATION

Authorization is a process of permitting users (whose identity has been authenticated) to perform certain operations on certain data objects in a shared database. The person who is

incharge of specifying the authorization is the *Authorizer* (one who owns the data.) In most cases authorizer is DBA.

In the authorization, the granting of a right or a privilege that enables a subject (i.e. A user) to have some rights to access to a system or a system object. Authorization control are built into the software. They govern what systems objects (for example : Database tables, views, procedure, triggers) a specified user can access and what the user may do with them.Procedure of authorization involves authentication of subjects (potential user or program) requesting access to objects.

A user may have several forms of authorization on part of the database using the various manipulation operations such as read, insert, delete and update.

- **Read Authorization** - Allows only reading of existing data but no modifications are allowed.
- **Insert Authorization** - Allows inserting new data . For insertion, user may be allowed to read the existing data as well but the modifications of the existing data is not allowed.
- **Delete Authorization** - Allows deleting an existing data. If the user deletes all the tuples of a relation, the relation still exists but it is empty.
- **Update Authorization -** Allows a user to make updations on existing data but no deletion of data is allowed. However some data items such as primary key attributes may not be modified.

A user may be assigned all, none, or a combination of these types of authorization. In addition to the above mentioned authorizations for assessing the data, the user may be authorized to perform control operations to modify database schema such as :-

- **Resource Authorization** - Allows the user to add new relations, records and set types.
- **Drop Authorization** - Allows the user to drop or delete existing relations from the database. Unlike deletion, on dropping a relation it will no longer exist.
- **Alter Authorization** - Allows the user to add new data-items or attributes to an existing relation. It also allows the user to drop the existing data-items or attributes from existing relations.
- **Index Authorization** - Allow the users in creation and deletion of indexes. Thus the users get the fast access of the data on the basis of some primary key field.

Several users might issue several grant permissions and revoke (denial of permission) can be represented using the authorization graph, which consists of following steps.

- Each user in the graph is represented by a node.
- An edge from U to W labelled P means U has granted privilege P to W and removed if the privilege is revoked later.

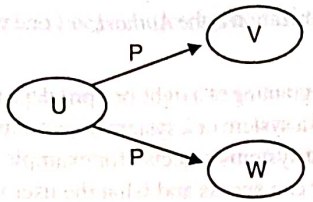

Fig. 9.1

Here U has granted privileges P to users V and W

9.6 PRIVILEGES

A *privilege* is a right to execute a named object (i.e. Database tables, views, procedures, Triggers) in a prescribed manner. For Example : A permission to create a table, right to select rows from another user's table, right to connect to the database (creating a session), permission to query a table etc.

The privileges are granted to users in a limited fashion so as not to compromise with the database security. Granting of the privileges, is done by the highest authority (in most cases, DBA) and helps the users to accomplish their required tasks.

A privilege can be granted to the user in the following two ways.

1. Explicitly by giving the right to the user to insert records into the table.
2. By granting privileges to the roles (which are basically a named group of privileges) and then granting roles to the users.

There are two distinct categories of privileges.

a) System Privileges
b) Object Privileges

System Privilege : A system privilege is also known as account level privilege in which the DBA specifies the particular privileges that each account holds independently of the tables in the database. In other words, a system privilege is a right to perform a particular action, or to perform a particular action on a particular type of object. The examples of system privileges are

- Create Table, Session.
- Create view
- Alter any procedure, role
- Drop privileges

The system privileges can be granted/revoked to users in Oracle using GRANT and REVOKE statements. Users who are granted system privileges using WITH ADMIN option can grant or revoke system privileges.

Object Privilege : An object privilege is also known as table level privilege which is used to access each individual relation , view, procedure, sequence, function etc. in the database. In other words, object level privileges are those guaranteed from a user to access or manipulate database objects. For example- A database user who want to insert a row into the " employee" table of user 'Rohit' must have granted a specific privilege to do this. Some of the object privileges are as follows :-

- *Select* applies to tables, views, sequences.

  This privilege allows a user to issue a query against a table or view or select a value from a sequence.
- *Insert*, *update* and *delete*- All these apply to tables and views. These privileges enable the user to insert new rows, delete existing rows, update existing rows in a table or view.
- *Execute* applies to procedures, functions and packages etc.
- *References* applies only to tables.

Difference between object privilege and system privilege.

1. Object privileges are used to control access to specific database objects whereas the system privileges control access to various system level facilities.
2. Object privileges are used to permit or prevent execution of database manipulation statements whereas the system privileges are used to permit or to prevent database definition language statements.
3. Object privileges are granted with the ability to grant the privileges to others using WITH GRANT option whereas the system privileges are granted with an ability to grant privilege to others using WITH ADMIN option.

9.7 DATABASE ROLES

Database applications are often composed of a large number of different database objects (tables, views , procedures, functions etc.). A DBA will quickly become helpless if they have to ensure each object privilege is granted or revoked explicitly from every user of the application. So in order to overcome these problems a mechanism known as roles which are used to provide authorization is used.

The database roles are named collection of privileges which can be either object or system privileges or both. It significantly reduces the burden of privilege maintenance for users. The DBA simplify need to create distinct roles which reflect the security privileges of the organization and grant these roles to users rather than granting discrete privileges. One should remember that a user can access to several roles and several users can be assigned the same roles.

The following properties of the roles allow easier privilege management within database.

1. **Reduce privilege administration** - Rather than granting discrete privileges to multiple users, the DBA can grant the privilege for a group of users to a role and grant this role to users which significantly reduce the burden of privilege maintenance for users.
2. **Dynamic privilege management** - If any changes are made to the database roles then they are effective immediately. There is no need to terminate all users sessions and have them to login again for the changed privileges in the roles to take effect.

3. **Selective availability of privilege** - Roles assigned to the user can be enabled or disabled selectively. This allows specific control of a user's privileges in any given situation.

4. **Application-specific security** - An additional level of security can be gained by providing password protection to a database role. Use of a password protected role is often convenient when you wish to temporarily suspend access to a collection of database objects without have to revoke privileges. Users cannot enable the role if they don't know the password. Due to a large number of advantages offered, generally privileges should be granted to roles and not to specific users.

9.8 VIEWS

A view is a dynamic result of one or more relational operations with the base table which produce another table. It is a virtual table i.e. it does not exist in the database but is produced upon request by a given user at a particular time. The view is a powerful and flexible security mechanism as it hides parts of databases from certain users.

Users can be granted permission on views without being given any permission on the base table used in the view definition. Ability of views to hide data serves both to simplify usage of system and to enhance security by allowing users access only to data they need for their job. Thus, a combination of relational level security and view level security can be used to limit a users access to precisely the data that users needs. Views being a shadow of the original base table always displays the table's data. Dropping the base table destroys the view of the table. If anyone tries to access that view an error is generated.

To explain the concept of a view, let us consider the example of a banking system. Suppose a bank clerk needs to know the names of the customers of each branch but is not authorized to see specific loan information. So for this purpose, the customer loan view is defined which consists only of the names of customers and branches at which they have a loan, and grant this view to the bank clerk and deny direct access to the loan base table. The view is defined as follows:

```
CREATE VIEW CUST_LOAN AS SELECT BRANCHNAME,
CUSTOMER_NAME FROM BORROWER, LOAN
where BORROWER.LOAN_NO = LOAN.LOAN_NO;
```

Since the clerk is authorized to see this view so clerk can execute a query to see the result.

```
SELECT * From CUST_LOAN ;
```

When the query processor translates the result into a query on actual base table in the database we obtain a query on BORROWER and LOAN tables. This permission must be checked on clerk's query before query processor begins.

SSION ON VIEWS

1 of view doesn't require resource authorization since no real table is being

2. Creator of a view gets only those privileges that provide no additional authorization beyond that he already had. For example, if creator of view CUST_LOAN had only read permission on BORROWER and LOAN tables, then he gets only read authorization on CUST_LOAN.

9.9 DATABASE SECURITY AND DBA

The Database Administrator (DBA) plays an important role in enforcing the security related aspects of a database design. He is responsible for overall security of the database system. The DBA's responsibilities includes granting privileges to users who need to use the system and classifying users and data in accordance with the policy of the organization. The DBA performs the following types of actions.

1. **Creating new accounts** - The database administrator provides the new UserId and password to the user or group of users so that they can access the DBMS.
2. **Privilege granting and revoking** - DBA can grant or revoke the privileges from a user or group of users according to the policy of the organization.
3. **Security level assignment** - It consists of assigning user accounts to the appropriate security classification level. This action is used to control mandatory access control.

9.10 DATA ENCRYPTION

Data encryption and access control are the internal computer techniques used to protect the data from unauthorized disclosure, alteration or destruction. This access control mechanism will not avoid all unauthorized accesses. Infact over half of all reported abuses invoking the computer resulted from insiders abusing their access authority.

The access control technique is ineffective if

1. Password are written down and found.
2. Off-line backup files are stolen.
3. Someone taps on a communication line.
4. Confidential data is left in the main memory after the job has completed.

So to counteract the possibility that either active or passive intruders obtain unauthorized access to sensitive data it is desirable to hide the meaning of the data accessed. The technique of data encryption is used for protecting the sensitive data against all unauthorized users.

Data encryption is a technique in which sensitive data is first encoded and then transmitted. At receiving end this encoded data is then decoded. It thus provides confidentiality for transmitted data which makes it difficult to extract information content by unauthorized user. Although this technique provides confidentiality of transmitted data but there is a degradation in performance as time is in decoding the encrypted data.

9.10.1 PROPERTIES OF GOOD ENCRYPTION TECHNIQUE

- Relatively simple for authorized users to encrypt and decrypt data.
- Encryption scheme depends not on the secrecy of the algorithm but on the secrecy of a parameter of the algorithm called the *encryption key*.
- Extremely difficult for an intruder to determine the encryption key.

9.10.2 CONVENTIONAL ENCRYPTION

Prior to the introduction of public-key encryption in 1970's, a conventional encryption technique, also referred to as *symmetric encryption* or *single-key encryption* was introduced. It is more widely used in present day diplomatic, military and commercial usage.

The conventional encryption scheme has the following parts :-

- **Plaintext** - This is the original message or data that is fed into the algorithm as input.
- **Encryption algorithm** - This algorithm performs various substitutions and transformations on the plaintext.
- **Secret key** - It is also input to the encryption algorithm. The exact substitutions and transformations performed by the algorithm depends on the key.
- **Ciphertext** - This is scrambled message produced as the output. It depends upon the plain text and the secret key. For a given message, two different keys will produce two different ciphertexts.

  Original Phrase - database management
  Scrambled Message - ADATABES MANAGEMENT
- **Decryption Algorithm** - It is just like running the encryption algorithm in reverse order. It takes the ciphertext and the secret key and produces the original plaintext.

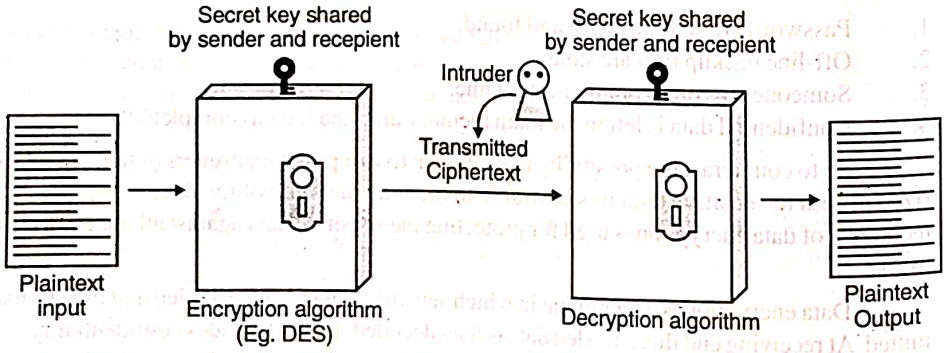

Fig. 9.2 Model of Data Encryption.

An encryption algorithm transform the senders plaintext message (input) using a secret key to produce the *ciphertext*. The plaintext is in the form which is recognizable by

humans or computers. The encrypted message or data can be transmitted through an insecure channel or stored in an insecure area. Using the same key, the decryption algorithm applies an inverse transformation to the ciphertext to reconstruct the original message i.e. plaintext (output). Transmission of the secret key to the decrypter must be kept secure since, knowing the decryption key and encryption algorithm, makes it easy to decrypt a message, thereby not disclosing sensitive information.

9.10.3 TECHNIQUES USED FOR ENCRYPTION

The traditional methods used for data encryption are the following :

- Transposition ciphers
- Substitution ciphers

9.10.3.1 TRANSPOSITION CIPHERS

Transposition ciphers retain the identity of the original characters of the plaintext but change their position. For example : If the transposition rule is to transpose each consecutive pair of characters.

Original Phrase : *data encryption*
it appears on transposition as
Ciphertext : *adatebcnyrtpoin* (Here 'b' is blank space)

Here the each consecutive pair of characters starting from left is interchanged with each other i.e. 'da' interchanged with 'ad' and 'ta' interchanged with 'at'.

However, this rule is not very secure. So other rules can also be devised and one of them is to take permutation to form blocks of four characters and permute 1234 to 3124. So the above original phrase now becomes. Here the blank characters are padded at end **ciphertext** : *tdaanbecprytnio* (Here 'b' is blank space) with the correct guess of the length of the permutation block only a few trails are needed to break the code.

9.10.3.2 SUBSTITUTION CIPHERS

Substitution ciphers retain the relative position of the characters in the original plain text, but hide their identity in the cipher text. In the substitution cipher each letter or group of letters is replaced by another letter or a group of letters to disguise it. One of the oldest known cipher is the Ceaser Cipher, in which 'a' becomes 'D', small 'b' becomes 'E' and so on.

For Example :

Plaintext : *encryption* becomes

Cyphertext : *HQFUBSWLRQ*

Such a cipher is very easy to break. So some other improved ciphers used are mono-alphabetic substitution ciphers, poly-alphabetic substitution ciphers etc.

9.10.4 DATA ENCRYPTION STANDARD (DES)

In 1977, the US National Bureau of Standards adopted a standard for data encryption called DES (Data Encryption Standard). It was widely adopted by the Industry for the use in security products.

The DES is an iterative product cipher repeatingly applying both transposition and substitution operations to blocks of data or text on the basis of the encryption key which is provided to the authorized users via a secure mechanism.

9.10.5 PUBLIC KEY ENCRYPTION

Public Key Encryption is another encryption technique proposed by *Diffie* and *Hallman* in 1976. Public key algorithms are based on mathematical functions rather than on simple operations on bit patterns. Public key cryptography is asymmetric involving the use of two separate keys in contrast to the symmetric conventional encryption which uses only one key.

A public key encryption includes the followings :

1. Plaintext
2. Encryption algorithm
3. Public key and Private key : Each user has two keys.
   - **Public Key :** Publicly published key used to encrypt data, but cannot be used to decrypt data.
   - **Private Key :** Key known only to individual user, and used to decrypt data. Need not be transmitted to the site doing encryption.
4. Ciphertext
5. Decryption algorithm

The following diagram shows the public key encryption scheme.

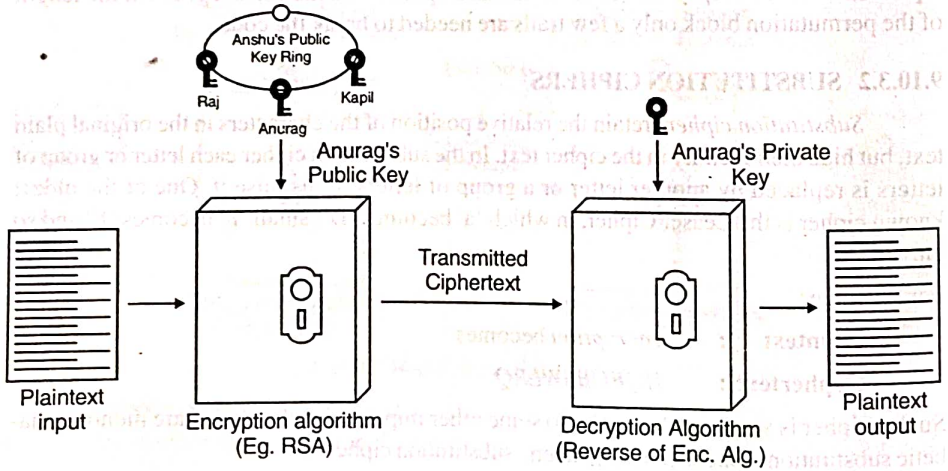

Fig. 9.3 - Model of Public Key Encryption

The public key encryption works as follows -

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of two keys in a public register known as the public key and the other key is known as private key. Each user maintains a collection of public keys obtained from others.
3. If Anshu wants to sends private message to Anurag then Anshu encrypts the message using Anurag's public key.
4. When anurag receives the message he decrypts it using its private key, since only Anurag knows his private key. No other recipient can decrypt the message.

The big advantage of public key encryption is that it doesn't require secure transmission of keys between sender and receiver. There need be no prior private communication between them before sending encrypted messages except to authenticate each others identity.

9.10.5.1 RSA PUBLIC KEY ENCRYPTION ALGORITHM

RSA public key encryption algorithm was developed in 1977 by *Ron Rivest, Adi Shamir* and *Len Adleman*. It is the most widely accepted public key encryption technique. It is based on the hardness of factoring a very large number (100's of digits) into its prime components.

9.10.6 DISADVANTAGES OF ENCRYPTION

The following are the disadvantages of encryption :

- Encrypting data gives rise to serious technical problems at the level of physical storage organization. For example : Indexing over data, which is stored in the encrypted form can be very difficult.
- Keeping keys secret is a problem. As long as a user protects his/her private key incoming communication is secure otherwise an intruder can decrypt it.
- Breaking a cipher and discovering a key is often easier if the intruder can obtain some plaintext and its corresponding cipher text.

9.11 STATISTICAL DATABASE SECURITY

A *statistical database* contains confidential information about individuals which is used to answer statistical queries concerning averages, totals with certain characteristics. The database may contain confidential data on individuals which should be protected from user access. The main objectives of statistical database is to maximize the sharing of statistical information, yet preserve the privacy of the individuals.

Statistical database security technique must prevent the retrieval of individual data. However users are permitted to retrieve statistical information on the population. A population is a set of tuples of a relation or table that satisfy some selection condition. For example: Consider a EMPLOYEE relation with attributes EMP_Id, ENAME, SALARY, POST, DEPT_NO. We can retrieve the number of employees in the EMPLOYEE table or the average income of the employees but statistical users are not allowed to retrieve individual data such as a salary of a particular employee.

In statistical database, security can be enforced in following ways :

1. **Threshold** - Disallow queries on populations less than the given threshold value.
2. **Sequence** - Disallow sequences that yield private data.
3. **Noise** - Deliberately distort data for low population.
4. **Audit trail** - Produce an audit trails of activities on the database. This trail will maintain the identity of the users and their interaction with the database.

9.12 DATA INTEGRITY

Security constraints help us to protect the data contained in the database against unauthorized or accidental access, modification or destruction whereas integrity constraints ensure that any properly authorized access, alteration, insertion or deletion of the data in the database doesn't change the consistency and validity of the data. So the term *Data Integrity* refers to the accuracy or the correctness of the data in the database. Since now a days most of the databases are shared by multiple users so it becomes necessary to preserve the data integrity in the presence concurrent operations, errors in the user operations and application programs, failure in hardware/software etc. When the contents of the database are changed using the insert, delete or update operations, the data integrity may be effected in many different ways. Some of these are

- Power failures or system crash may lead to loss of changes made to the database.
- Incorrect data may be entered into the database. For example : The age may be entered as 200 years instead of 20 years.
- Partial completed transaction may lead to inconsistency of data base.

So Integrity constraints must be applied to the database so that operations (insert, update, delete etc.) performed on the database should not make the dat abase state inconsistent.

The data integrity constraints which maintain consistency and validity can be divided into following categories :

- Domain Integrity rules
- Base Table constraints
- General constraints

9.12.1 DOMAIN INTEGRITY RULES

A *domain* is a set of atomic values. By *atomic* we mean that each value in the domain is indivisible. Domain integrity rule specify that value assigned to each attribute must be an atomic value from the domain of that attribute. The data types associated with domains includes NUMERIC data type for real and integers, CHARACTER data types, date, time etc. The definition of the type of a domain must be as precise as possible in order to avoid violation of domain integrity rule.

Some examples of domains are

1. The domain of attribute sex can either only be M (Male) or F (Female).
2. The age for a government employees must be between 18 years to 60 years.
3. The final grade assigned to the student can only have one of A, B, C, D, E, F, G, H values.
4. The domain can be composite. As in case of DATE_OF_BIRTH attribute in the STUDENT relation is constraint to the form MM/DD/YYYY, where MM is month and is restricted to 1 to 12 and similarly for DD (Day) and YYYY (year).

Domain Integrity constraints are performed by the Database Management System. However, some types of errors cannot be detected even using these rules. For example - If a teacher may incorrectly assign a grade 'G' instead of 'H' to a student because these two keys are next to each other. The domain integrity rules doesn't provide checks on these types of errors. Thus, Domain Integrity mechanism can only ensure that the data is in the specified domain.

9.12.2 BASE TABLE CONSTRAINTS

The base table constraints can be any of the following :-

- Candidate key definition
- Foreign key definition
- Check constraint definition

9.12.2.1 CANDIDATE KEY DEFINITION

A candidate key is any attribute or combination of attributes that uniquely identify a record and the attributes that form the key cannot be further reduced. Since a null value is not guaranteed to be unique so no data value of the key should contain a null value, but they should contain unique values. A table can have any number of candidate keys. Out of these, one candidate key is designated as a primary key, and the remaining keys are called the alternate keys.

In SQL, a candidate key definition can take the following form

[CONSTRAINT constraint_name] {Primary key|UNIQUE}
(column_comma_list)

In the given definition, the **column_comma_list** must not be empty. The keyword **Primary Key** is used to define the primary key of the table and unique is used to define the alternate keys of the table. A table can have atmost one primary key specification but any number of unique specification.

For example : In EMP relation, a candidate key consists of the columns EMP_ID and PAN_NO. If EMP_ID is taken as primary key then PAN_NO is specified with UNIQUE keyword.

9.12.2.2 FOREIGN KEY DEFINITION

Different tables in a relational database can be related by common columns and the rules that govern the relationship of the columns must be maintained. So the foreign key integrity constraints guarantees that these relationships are preserved.

A *foreign key* is an attribute or a set of attributes of a table whose values are required to match some primary key value in some table and they can also contain null values.

9.12.2.3 CHECK CONSTRAINT DEFINITION

A Check Integrity constraint on a column or a set of columns requires that a specified condition be true for every row of the table. Check constraints enable you to enforce very specific integrity rules by specifying a check condition. Once these constraints are defined the DBMS will automatically evaluate, the check constraint expression whenever the operations is being performed. If the check condition evaluates to be TRUE then the changes are allowed to the data in the table otherwise it remain unchanged.

It can take the following form :-

CHECK (conditional expression)

For example - If you want to check whether the salary is between Rs. 12000 and Rs. 30000 then it can be written as

CHECK (salary between 12000 and 30000)

9.12.3 GENERAL CONSTRAINTS

A General constraint is one that applies to combinations of columns in combinations of base tables. For example :

1. The HRA of all employees in department D3 should be twice the basic pay.
2. No shipment has a total weight greater than 25000 kg.
3. The salary should be greater than Rs. 30000.
4. Age of the eligible voter should not be less than 18.
5. The hours entered will entering time should not be greater than 24.

SUMMARY

- Database security is the protection of the database against intentional or unintentional threats that may be computer based or non-computer based controls.
- A Threat is any situation, event or personnel that will adversely effect the database security and smooth and efficient functioning of the organization. Threat to a database can be intentional or accidental.
- Some common database security threats are
  - Data Tampering
  - Eavesdropping and data theft
  - Falsifying User's Identities
  - Password related threats
  - Unauthorized access to data
  - Lack of accountability.
- Various levels of mechanisms required for database security include.
  - Physical Security
  - Human Factors
  - Operating System
  - Database System
- Database Security requirements include
  - **Confidentiality** - It deals with following aspects - Access control, Authenticated users, secure storage of sensitive data, privacy of communication.
  - **Integrity** - It consists of following aspects - System and object privileges control, Integrity constraints, access to network.
  - **Availability** - It deals with the following aspects - Ease of Use, Flexibility, Scalibility, Resistance.
- Discretionary access control are used to grant privileges to users including the capability to access specific data files or fields in a specific mode.
- Mandatory access control are used to enforce multilevel security by classifying the data and users into various security classes.
- Authorization is a process of permitting users to perform certain operations on certain data objects in a shared database.
- A privilege is a right to execute a named object in a prescribed manner. The privileges are granted to users in a limited fashion so as not to compromise with the database security. System privileges and object privileges are two distinct categories of privileges.
- The database roles are named collection of privileges which can be either object or

system privileges or both. It significantly reduces the burden of privilege maintenance for users.

- A view is dynamic result of one or more relational operations with base table which produce another table. It is a virtual table as it doesn't exist in the database but is produced upon the request by a given user at a particular time.
- Data encryption is a technique for providing confidentiality for transmitted data or text before transmission or storage which makes it more difficult to extract information content or meaning.
- In a statistical database, the objective is to maximize sharing statistical information and yet preserve privacy of individual record. This security problem cannot be solved by normal access control strategy since the aim of the database is to allow users full access to the data.
- Data integrity refers to the accuracy or the correctness of the data in the database. The integrity constraints ensure that any properly authorized access, alteration, insertion or deletion of the data in the database doesn't change the consistency and validity of the date.
- Domain Integrity rule specify that value assigned to each attribute must be an atomic value from the domain of that attribute. A domain is a set of atomic values. By atomic we mean that each value in the domain is indivisible.

Frequently Asked Questions (FAQs')

Q1. What is database security ? Why is database security important for an organization ?
Q2. What is a threat ? What are the potential threats in a database system ?
Q3. What do you mean by eavesdropping and data theft ?
Q4. What is a data tampering ?
Q5. How are user's identity falsified ?
Q6. How do people gain unauthorized access to database objects ?
Q7. Explain the different types of defence mechanisms ?
Q8. What are the basic data security requirements ?
Q9. What do you mean by confidentiality of data ?
Q10. How is confidentiality of data ensured in a database system ?
Q11. What do you mean by access control ?
Q12. What do you mean by database availability and what are its different aspects ?
Q13. How can you protect data within a database ?
Q14. What are the differences between discretionary access control and mandatory access control ?
Q15. What do you understand by authorization ? How are users authenticated to the database?
Q16. What is a privilege ? What are the different types of privileges ?

Q17. What is the difference between the object privilege and a system privilege ?
Q18. What are roles ? What are advantages of roles ?
Q19. What is a view ? How does it helps in security of data ?
Q20. How does DBA ensures database security ?
Q21. What is Data Encryption ? What are the different techniques used for encryption of data?
Q22. How is the data encrypted in a conventional technique i.e. DES ?
Q23. What are the different techniques used for data encryption ?
Q24. Explain the public key encryption technique ?
Q25. What is a statistical database ? How can the statistical database security be enforced?
Q26. What do you mean by database integrity ?
Q27. What are the Integrity constraints ? What are the different types of integrity constraints.
Q28. Differentiate between database security and database integrity ?
Q29. Explain the terms :-
a) Domain Integrity Rules
b) Plaintext
c) Public Key
d) Private Key
e) Domain