DATABASE SECURITY, INTEGRITY AND RECOVERY
DATABASE SECURITY, INTEGRITY AND RECOVERY
Database Security and Integrity

- Definitions
- Threats to security and integrity
- Resolution of problems
DEFINITIONS

SECURITY:

✓ Protecting the database from unauthorized access, alteration or deletion.

INTEGRITY:

✓ It refers to accuracy or validation of the data.
THREATS to security and integrity......

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 may be intentional or accidental.

Given below are some database security threats....

• Data tampering
• Eavesdropping and data theft
• Falsifying User’s identities
• Password related threats
• Unauthorized access to data
• Lack of accountability
DEFENCE MECHANISMS

Generally four levels of defence are recognized for a database security:

- Physical security
- Human factors
- Operating system
- Database system
Data Security REQUIREMENTS

The basic security standards which technologies can assure are:

⭐ CONFIDENTIALITY

- **Access control** - Access to data is controlled by means of privileges, roles, and user accounts.

- **Authenticated users** – Authentication is a way of implementing decisions of whom to trust. It can be employ passwords, finger prints etc.

- **Secure storage of sensitive data** – It is required to prevent data from hackers who could damage 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 etc.
INTEGRITY

Integrity contributes to maintaining a secure database by preventing the data from becoming invalid and giving misleading results. It consists of following aspects:

- **System and object privileges control access to applications tables** and system commands so that only authorized users can change the data.

- **Integrity constraints** are applied to maintain the correctness and validity of the data in the database.

- **Database must be protected from viruses** so firewalls and anti-viruses should be used.

- Ensures that **access to the network is controlled** and data is not vulnerable to attacks during transmission across network.
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 number of aspects.

- **Ease of use** – Resources managed by users for working with databases should be effectively managed so that it is available all the time to valid users.

- **Flexibility** – Administrators must have all the relevant tools for managing user population.

- **Scalability** - System performance should not get affected by the increase in number of users or processes which require services from system.

- **Resistance** – User profiles must be defined and the resource used by any user should be limited.
IMPORTANT SECURITY FEATURES

- Views
- Authorization and controls
- User defined procedures or privileges
- Encryption procedures
AUTHORIZATION is a PROCESS OF PERMITTING USERS to perform certain operations on certain data objects in a shared database. For example: Let us consider the authorization that a salesperson undertakes;

<table>
<thead>
<tr>
<th>AUTHORIZATION</th>
<th>CUSTOMER RECORDS</th>
<th>ORDER RECORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>INSERT</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>MODIFY</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>DELETE</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

Where **N stands for NO and Y stands for YES** to authorization for salesperson
To explain the concept of view,

let us consider the example of a bank clerk who needs to know the names of customers of each branch but is not authorized to see specific loan information. 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 processor begins.
DATABASE INTEGRITY

Constraints:

It can be defined in 3 ways-

1) Business constraints
2) Entity constraints
3) Referential constraints
• **BUSINESS CONSTRAINTS** – A value in one column may be constrained by value of some another or by some calculation or formulae.

• **ENTITY CONSTRAINTS** – Individual columns of a table may be constrained eg. Not null.

• **REFRENTIAL CONSTRAINTS** – Sometimes referred to as key constraints. Eg. Table two depends upon table one.
• BENEFITS OF USING CONSTRAINTS

✓ Guaranteed integrity and consistency
✓ Defined as a part of table definition
✓ Applies across all applications
✓ Cannot be circumvented
✓ Application development and productivity
✓ Requires no special programming
✓ Easy to specify and maintain
✓ Defined once only
CONCURRENCY CONTROL

• What is it?
  The coordination of simultaneous requests for the same data, from multiple users.

• Why is it important?
  Simultaneous execution of transactions over a shared database may create several data integrity and consistency problems.
THREE MAIN INTEGRITY PROBLEMS ARE

• Lost updates
• Uncommitted data
• Inconsistent retrievals
The process of restoring database to a correct state in the case of failure. E.g.

- system crashes
- Media failures
- Application software errors
- Natural physical disasters
- carelessness
BASIC RECOVERY CONCEPTS

• **Backup mechanism** – it makes periodic backup copies of the database.

• **Logging concept** – that keeps the track of current state of transaction and the changes made in the database.

• **Check pointing mechanism** – that enables update to be made permanent.
The choice of the best possible strategy depends upon the
• Extent of damage that had occurred to the database.

• If there has been a physical damage like disk crash then the last backup copy of the data is restored.

• However if database has become inconsistent but not physically damaged then changes caused inconsistency must be undone. It may also be required to redo some transactions so as to ensure that the updates are reflected in the database.
THANK YOU