

UNIT:3

Transaction Processing and Database Security

Prepared By: Prof. Meghna Bhatt

TOPICS TO BE DISCUSSED....

❖ Transaction Processing Concepts:

- ❖ Overview of Transaction Processing,
- ❖ Transaction States with State Transition Diagram
- ❖ ACID properties.

❖ Security Mechanism:

- ❖ Introduction,
- ❖ Discretionary Access Control (DAC)
- ❖ Mandatory Access Control (MAC),
- ❖ Public Key Encryption.



TRANSACTION


- ❖ The transaction is a set of logically related operation. It contains a group of tasks.
- ❖ A transaction is an action or series of actions.
- ❖ It is performed by a single user to perform operations for accessing the contents of the database.
- ❖ The operations performed in a transaction include one or more of database operations like insert, delete, update or retrieve data.
- ❖ Whenever a user reads from the database or writes to the database a transaction is created.



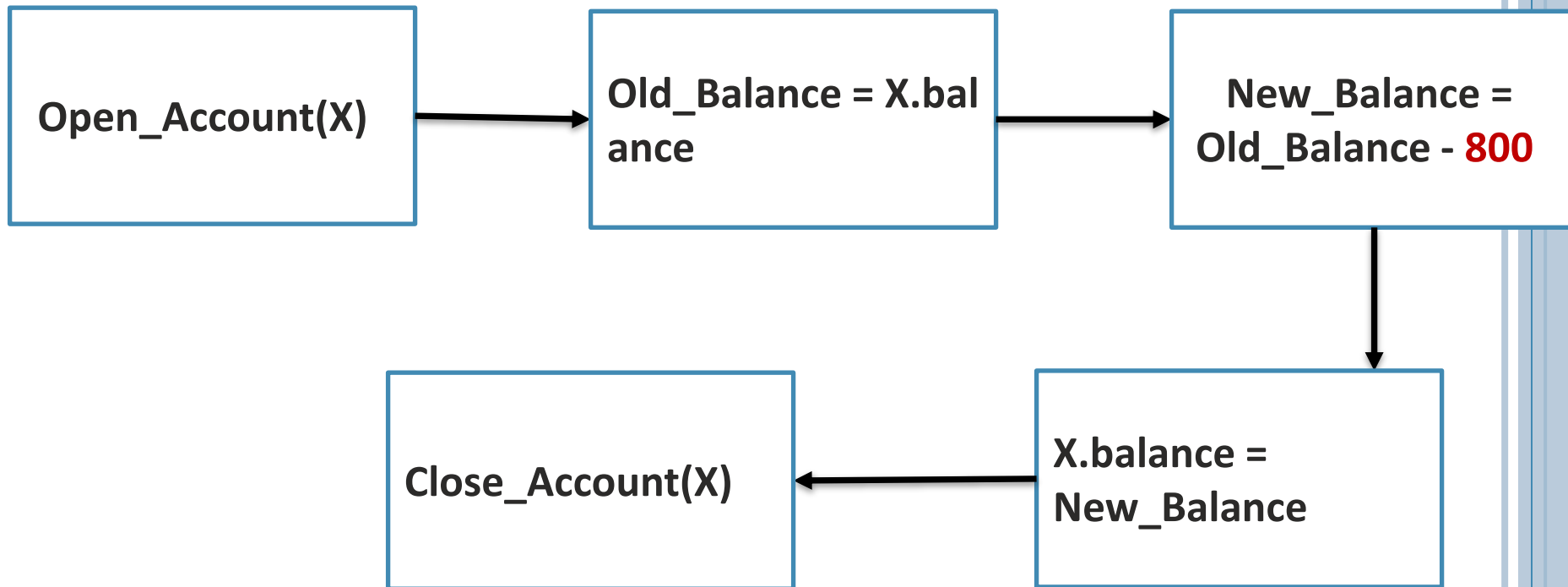
TRANSACTION PROCESSING

- A transaction is a series of read and write actions that are grouped together to perform the operation on the database.
- Whenever a user reads from the database or writes to the database a transaction is created.
- A transaction may consist of simple select operation or it may be consist of a series of related select and update command sequence.
- Example:

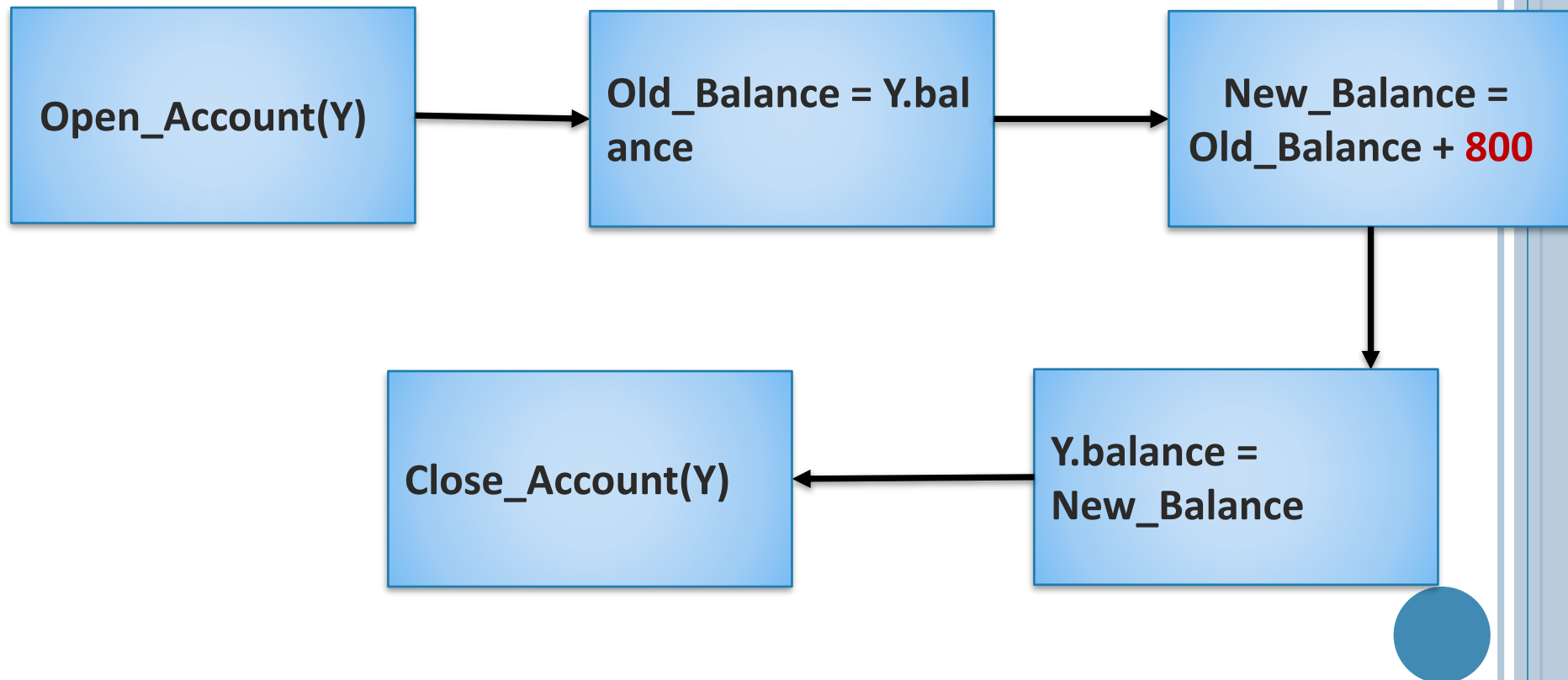
Suppose an employee of bank transfers Rs 800 from X's account to Y's account. This small transaction contains several low-level tasks:



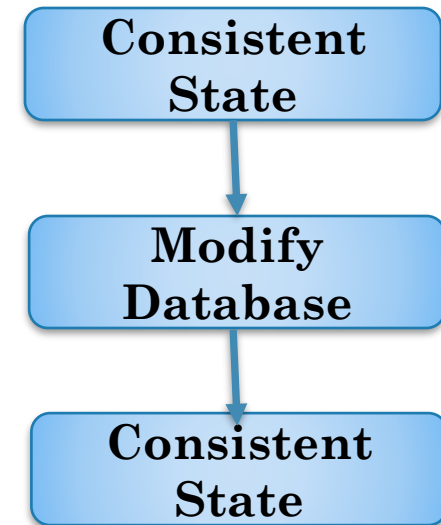
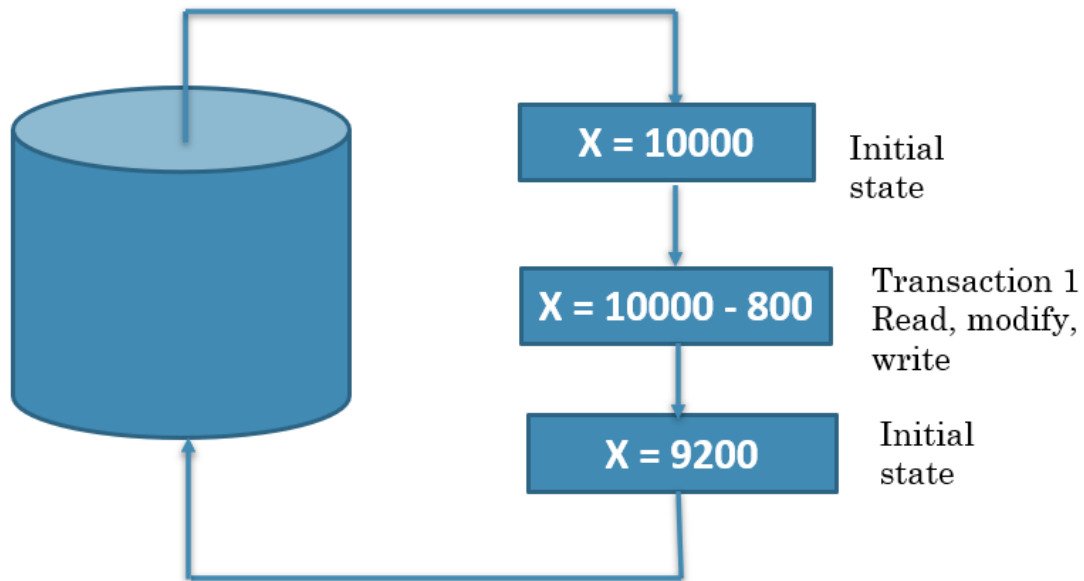
EXAMPLE



EXAMPLE



OVERVIEW OF TRANSACTION PROCESSING



State Transition Diagram

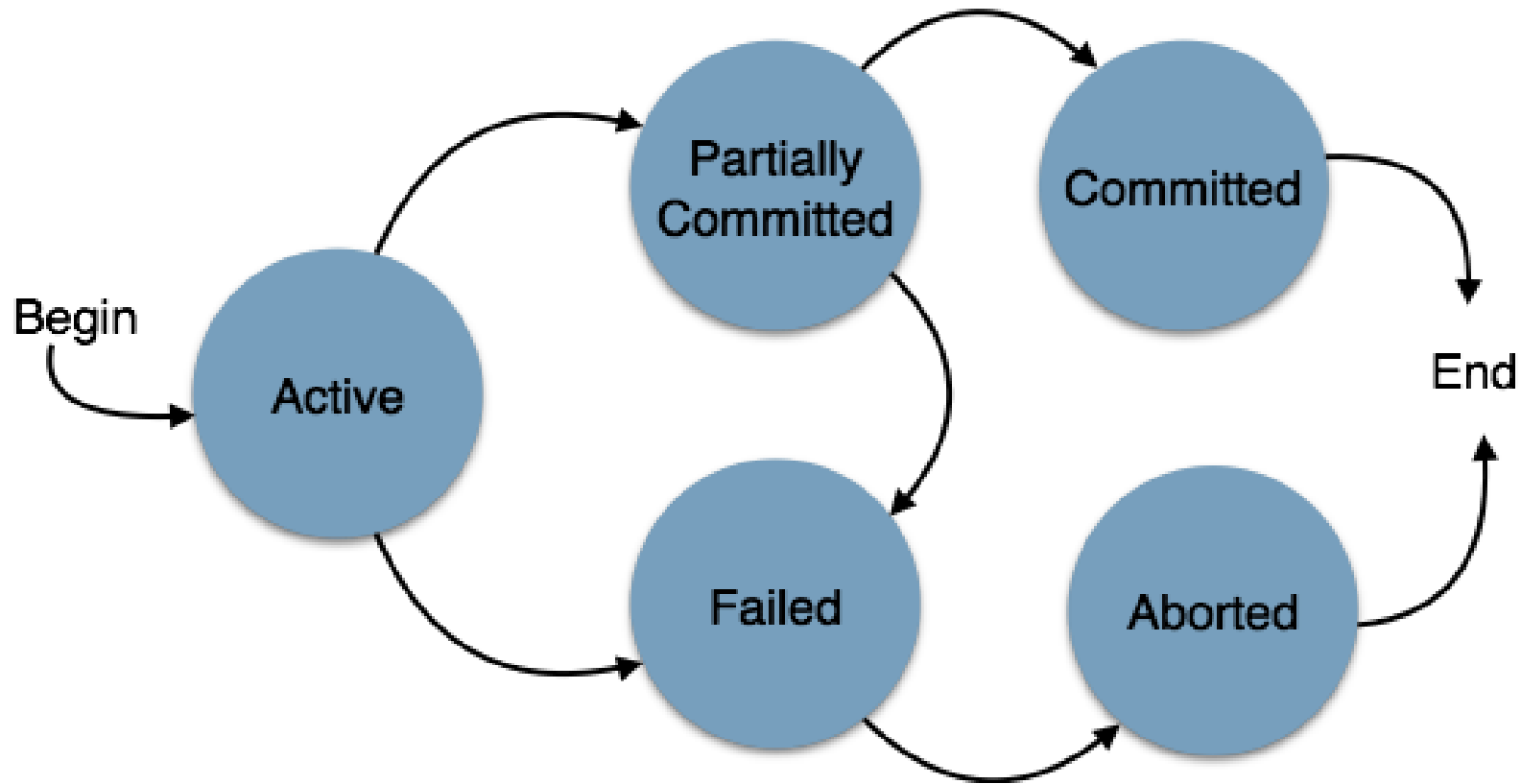


TRANSACTION STATES WITH STATE TRANSITION DIAGRAM

- ❖ A transaction is the sequence of one or more SQL statements that are combined together to form a single unit of work.
- ❖ A transaction goes through many different states throughout its life cycle. These states are called as transaction states
- ❖ Transaction states are as follows-
 - ❖ Active state
 - ❖ Partially committed state
 - ❖ Committed state
 - ❖ Failed state
 - ❖ Aborted state
 - ❖ Terminated state



STATE TRANSITION DIAGRAM



STATE TRANSITION DIAGRAM: EXAMPLE

Consider a banking application where a customer transfers money from one account to another.

The transaction involves two main steps: deducting the amount from the sender's account and crediting it to the recipient's account.

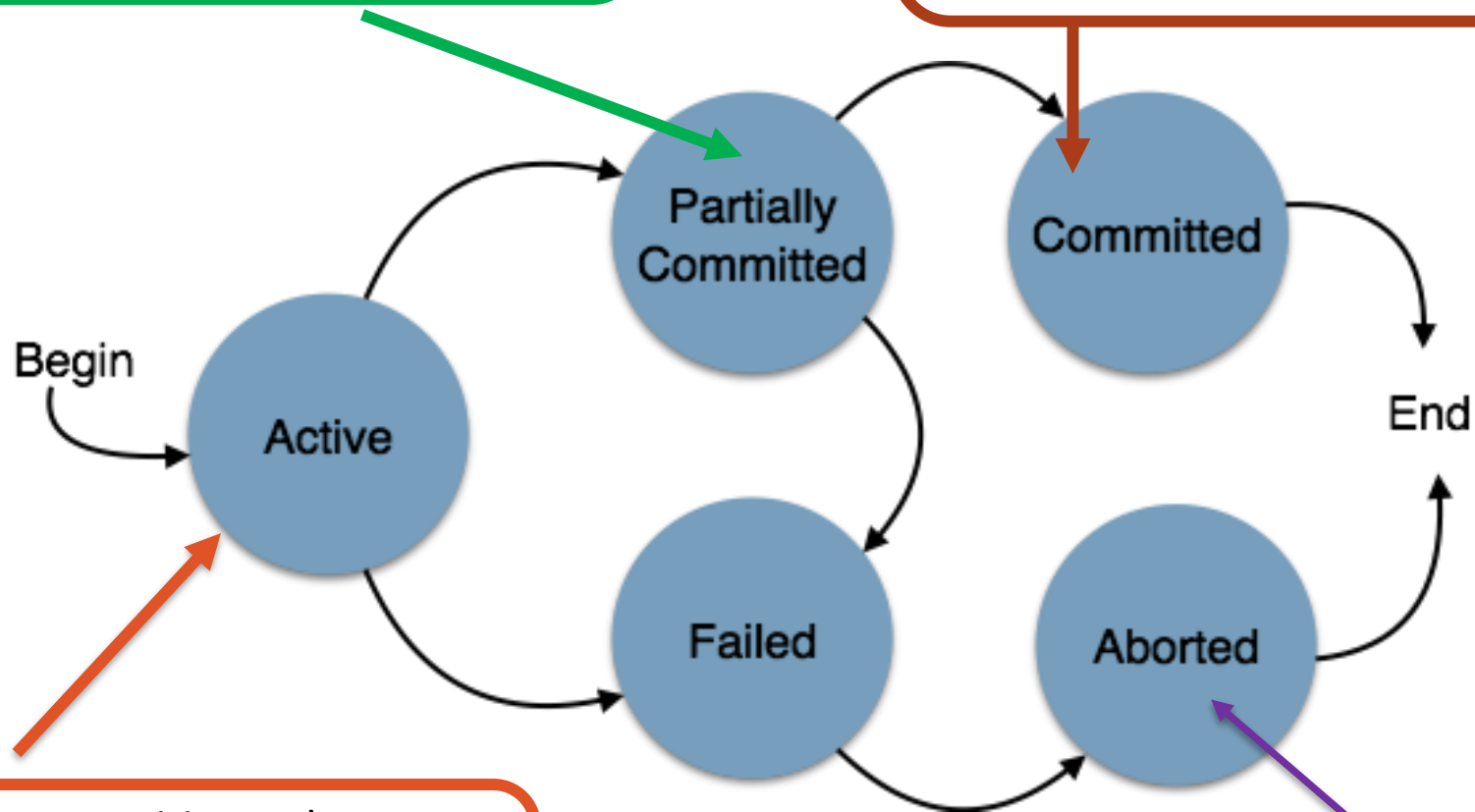


Deducting The Amount From The Sender's Account.

All modifications to the database but has not yet been confirmed.

Amount is successfully credited to the recipient's account.

All changes made by the transaction are now permanently saved in the database

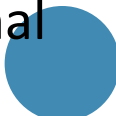


Customer Initiates The Transfer.

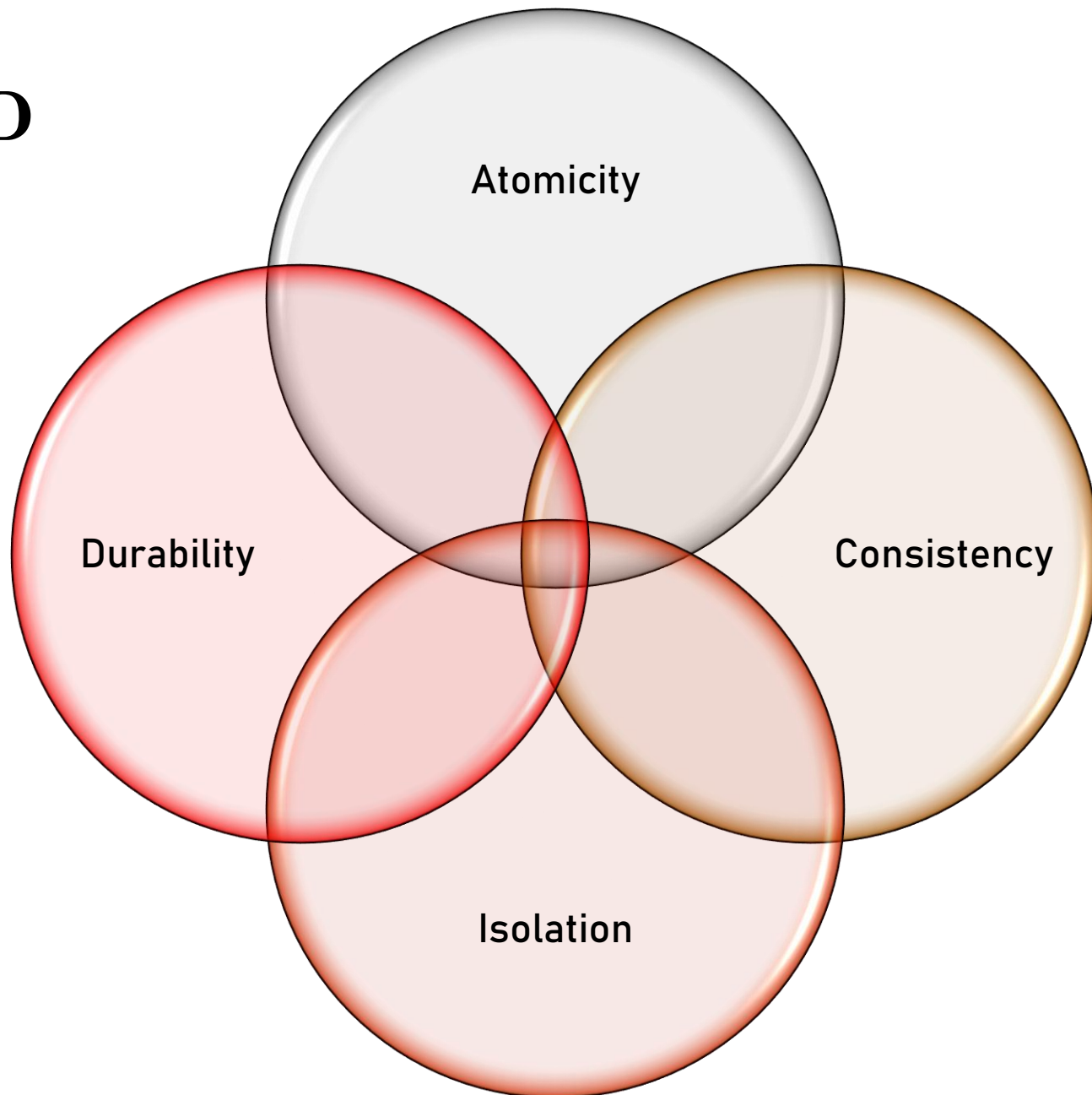
It Is Currently Executing And Modifying The Database.

Error occurs during the transaction process
transaction rolled back, and the database returns to original state.

STATE TRANSACTION EXAMPLE

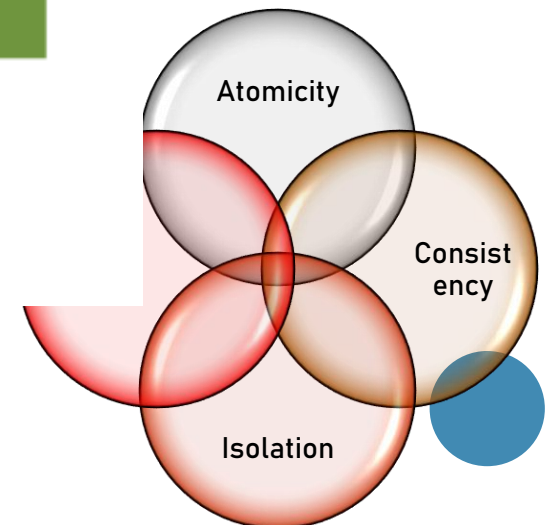
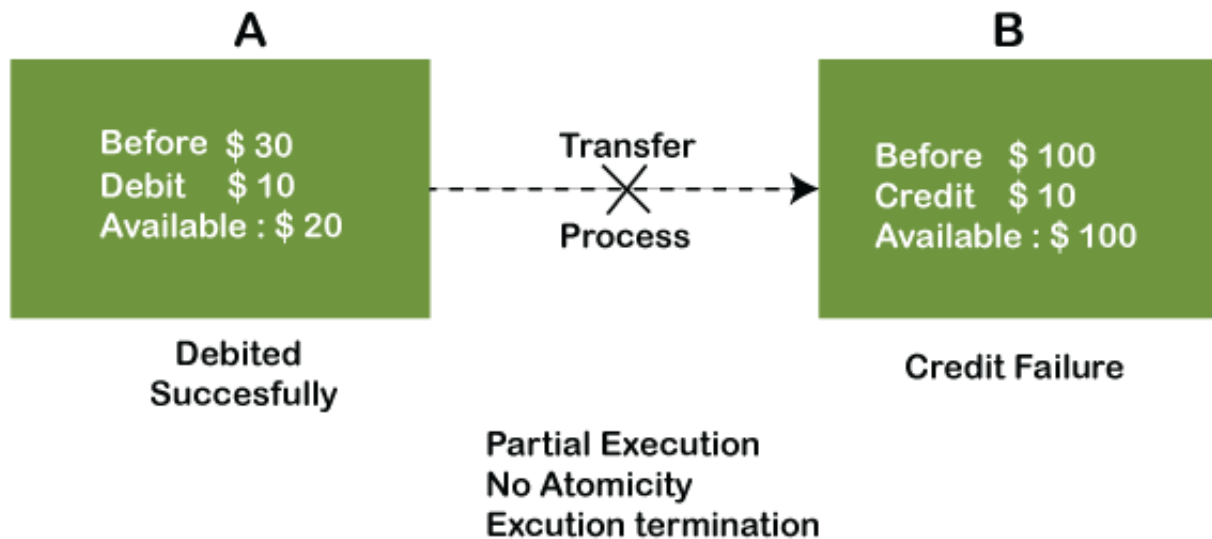
- **Active State:** The transaction starts in the active state when the customer initiates the transfer. It is currently executing and modifying the database.
 - **Partially Committed State:** After deducting the amount from the sender's account, the transaction enters the partially committed state. At this point, it has made all modifications to the database but has not yet been confirmed.
 - **Committed State:** Once the amount is successfully credited to the recipient's account, the transaction enters the committed state. All changes made by the transaction are now permanently saved in the database, and the transaction is considered successful.
 - **Aborted State:** If an error occurs during the transaction process, such as insufficient funds or a database failure, the transaction may enter the aborted state. In this state, any changes made by the transaction are rolled back, and the database returns to its original state.
- 

ACID

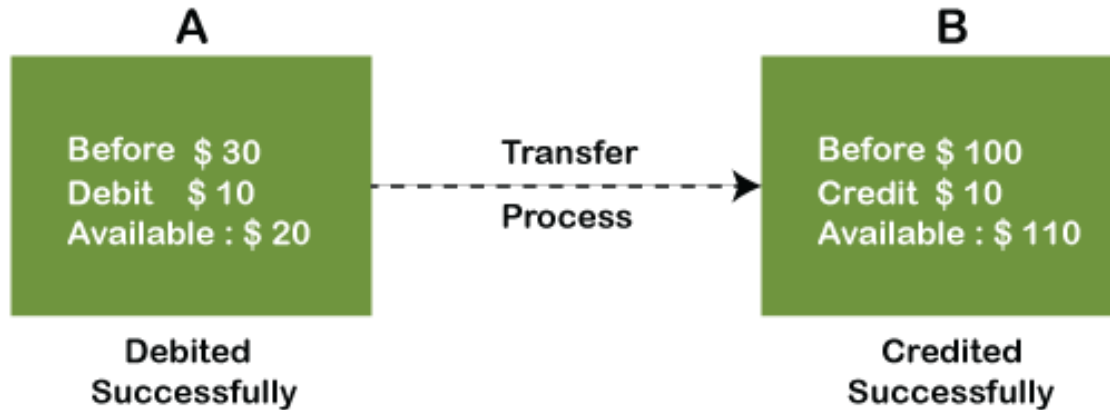


ATOMICITY

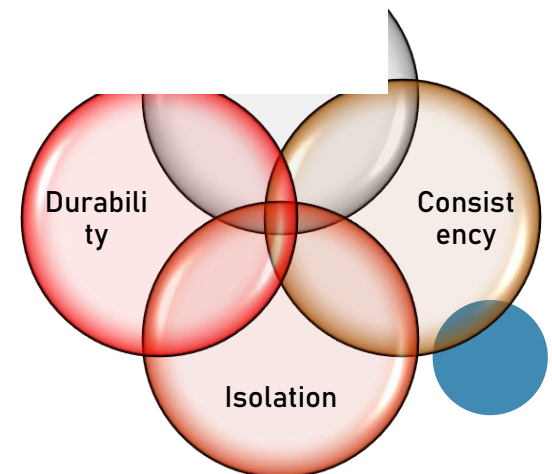
- ❖ Atomicity ensures that a transaction is treated as a single indivisible unit, either executing all its operations or none at all.



ATOMICITY

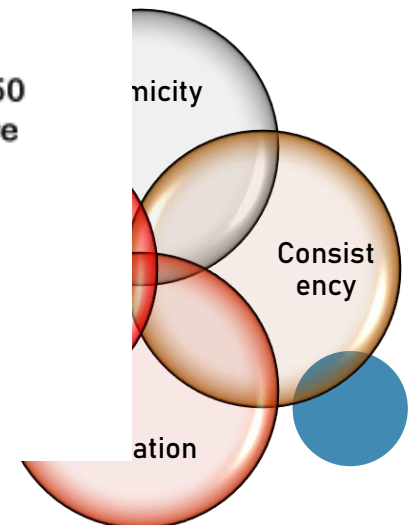
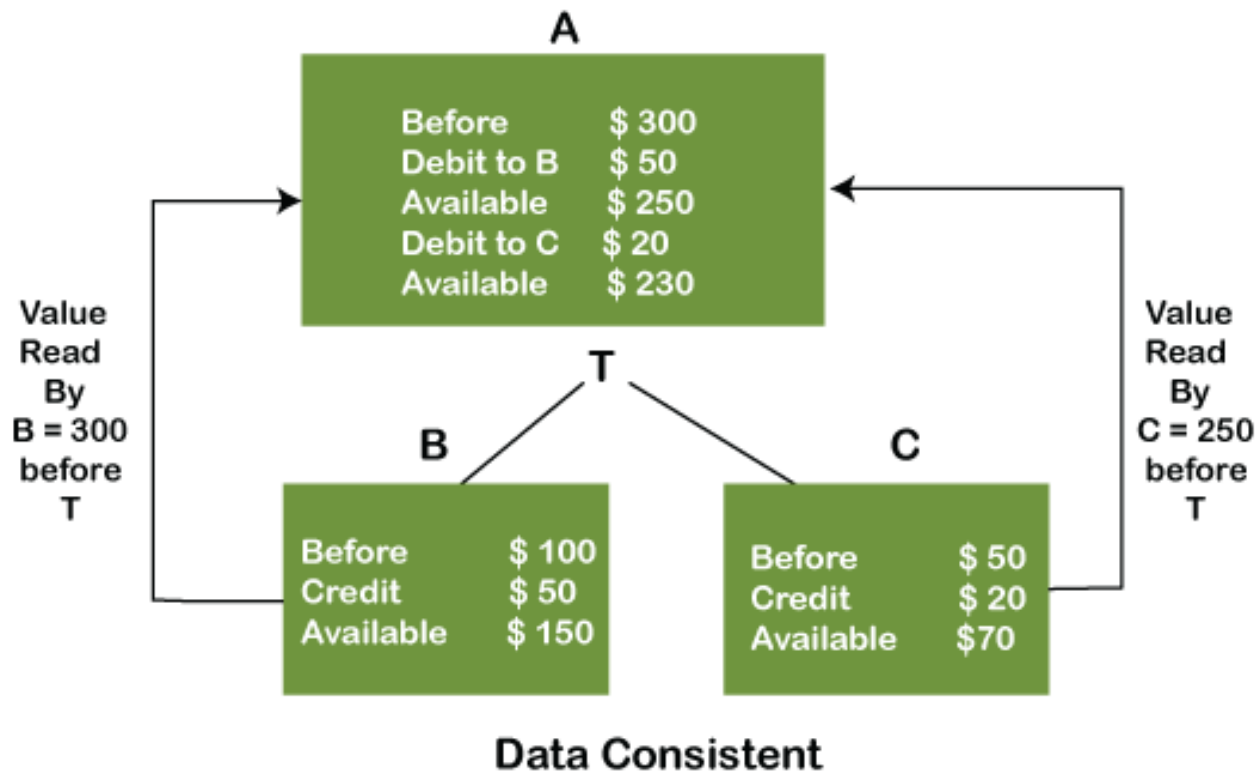


Complete Execution
Atomicity
Execution Successfull



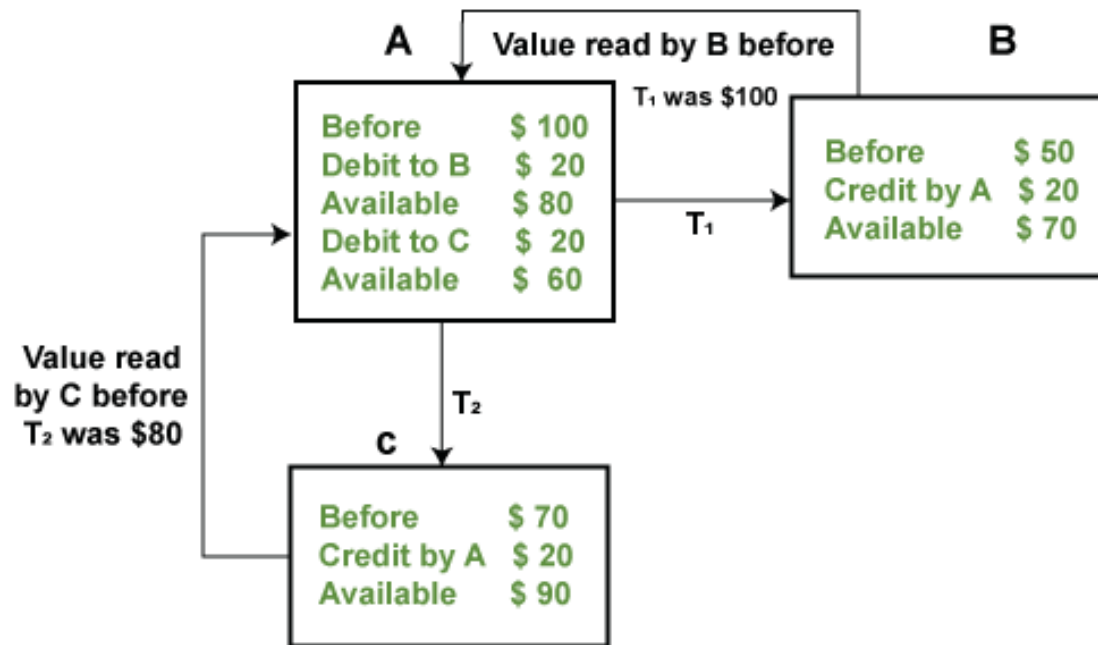
CONSISTENCY

- ❖ Consistency ensures that the database remains in a valid state before and after a transaction.

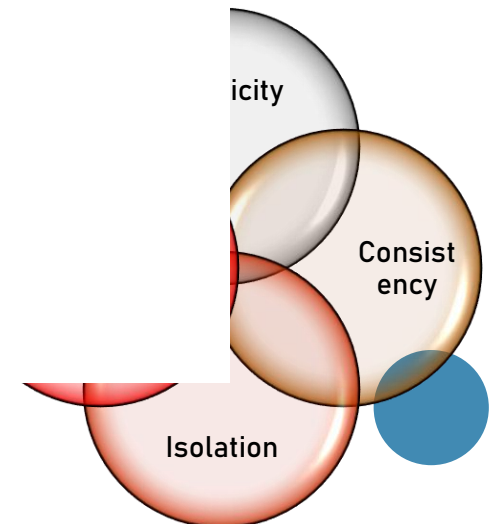


ISOLATION

- ❖ **Isolation** ensures that concurrent transactions do not interfere with each other, maintaining data integrity.

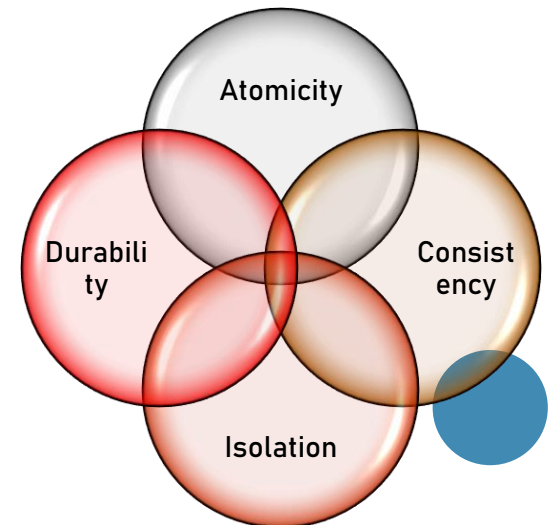


Isolation - Independent execution of T₁ & T₂ by A



DURABILITY

- ❖ **Durability** guarantees that once a transaction is committed, its effects are permanent and survive any system failures.
- ❖ Together, these properties ensure reliability and maintain data integrity in DBMS operations..



DATABASE SECURITY

- ❖ Database Security means keeping sensitive information safe and prevent the loss of data.
- ❖ Security of data base is controlled by Database Administrator (DBA).
- ❖ Managing database security has become more difficult and time consuming.
- ❖ Therefore, it important for the DBA to develop overall policies, procedures and appropriate controls to protect the database.



AUTHORIZATION

- ❖ Authorization is the process of a granting of right or privileges to the user to have a limited access to a system or objects of the system.
- ❖ It is an administrative policy of the organization, express as a set of rules that can be used to determine which user has what type of access to which portion of database.



AUTHENTICATION

- ❖ Authentication is a mechanism that determines whether a user is who he or she claims to be.
- ❖ In other words, an authentication checks whether a user operating upon the database is allow to doing so or not.
- ❖ It verifies the identity of the user.
- ❖ The simplest form of authentication is a simplest consists of a secret password which must be presented when a connection is open to database.



AUTHENTICATION

- ❖ Authorization and Authentication controls can be built into the software.
- ❖ Authorization rules are incorporated in DBMSs that restrict access to data and also restrict the action that people may take when they access data.
- ❖ Two types of access control techniques are used in database security system:
 - ❑ Discretionary Access Control.
 - ❑ Mandatory Access Control.



DAC

- ❖ DAC is based on the concept of privileges and mechanism for giving such privileges to user.
- ❖ It grant the privileges to user on different object, including capability to access specific data file, records or fields in specified mode, such as, read, insert, delete or update or combination.
- ❖ A user who creates a database object such as a table or view automatically gets all applicable privileges on that object.



DAC

- ❖ The DBMS keep track of how these privileges are granted to other users and it is very flexible.
- ❖ For Example, An unauthorized user into disclose of sensitive data.
 - ❖ Granting / Revoking Privileges
 - ❖ Audit Trail



USER

- ❖ Oracle relies on a mechanism that allows you to register a person, called a user.
- ❖ Each registered user has an access password, which must be provided in various situations.
- ❖ Each user is then assigned individual privileges or roles.



CREATE USER

- ❖ CREATE USER command is responsible for the creating of new user.
- ❖ It creates a user or an account through which you can connect to the database, and establish the means by which Oracle will allow the user access.



CREATE USER

```
CREATE USER user IDENTIFIED
{BY password | EXTERNALLY
  | GLOBALLY AS 'CN = user'}
[DEFAULT TABLESPACE tablespace
| TEMPORARY TABLESPACE tablespace
| QUOTA {integer [K | M] | UNLIMITED} ON tablespace
[QUOTA {integer [K | M] | UNLIMITED} ON
tablespace].....
| PROFILE profile
| PASSWORD EXPIRE
| ACCOUNT {LOCK | UNLOCK}.....
```



COMMON SYNTAX OF CREATE USER & EXAMPLE

SYNTAX:

```
CREATE USER user IDENTIFIED BY {password };
```

EXAMPLE:

```
CREATE USER MCA1 IDENTIFIED BY MCA123;
```



DELETING USER

- ❖ You can remove a database user with the DROP USER command.
- ❖ This command removes both the user and all the objects contained in this user's schema.
- ❖ You will need to specify the CASCADE clause of the command. Oracle also removes all the referential integrity associated to the objects of the removed user.

SYNTAX:

```
DROP USER <USER_NAME> [<CASCADE>];
```

EXAMPLE:

```
DROP USER MCA1;
```



DISPLAY ALL USERS

```
SELECT * FROM ALL_USERS;
```

USER NAME	USER_ID	CREATED
SYS	0	03/01/99
SYSTEM	5	03/01/99
OUTLN	11	03/01/99
DBSNMP	20	03/01/99
MTSSYS	28	03/01/99
AURORA\$ORB\$UNAUTHENTICATED	25	03/01/99
SCOTT	26	03/01/99
DEMO	27	03/01/99
ORDSYS	30	03/01/99
ORDPLUGINS	31	03/01/99
MDSYS	32	03/01/99
CTXSYS	35	03/01/99

PRIVILEGES

- ❖ A privilege is an authorization given to the user to access and manipulate a database object in a certain way.
- ❖ There are two types of privileges :
 - ❖ System Privileges
 - ❖ Object Privileges.




SYSTEM PRIVILEGES

- ❖ A system privilege is the right or permission to execute certain database actions in a specific type of database object.
- ❖ These are more than 70 types of privileges associated to the name of the action it executes.



SYSTEM PRIVILEGES

System Privilege	Type of Action
CREATE	Create an object in a user's own schema.
CREATE ANY	Create an object in another user's schema.
CREATE SESSION	Connect to the database.
DROP	Drop an object in user's own schema
DROP ANY	Drop an object in another user's schema
ALTER SYSTEM	Manipulate an instance
ALTER DATABASE	Manipulate the database
ALTER USER	Change user's role or password
ALTER/ CREATE/ DROP/ MANAGE TABLESPACE	Manipulate a tablespace



OBJECT PRIVILEGES

- ❖ An object privileges is the right to perform certain actions in a specific object, such as the right to include a row in a table.
- ❖ These privileges do not apply to all the objects of a database links, and clusters do not have any object privileges.
- ❖ The following is a list of the privileges:

PRIVILEGE	OBJECT
ALTER	Tables and sequences
DELETE	Tables and views
EXECUTE	Procedures
INDEX	Tables
INSERT	Tables and views
REFERENCE	Tables
SELECT	Tables, views, and sequences
UPDATE	Tables and views



GRANT

- ❖ To assign privileges this command will be used.
- ❖ The GRANT command has two distinct forms: - one to distribute system privileges and the other to distribute object privileges.
- ❖ Only the DBA can use the GRANT command to distribute system privileges.



GRANT

- ❖ The following types of privileges can be granted:
 - ❖ Assign role: connect, resource & DBA.
 - ❖ Insert data into a specific table.
 - ❖ Update data into a specific table.
 - ❖ Display data of any table.
 - ❖ Delete data from a specific table.



GRANT

❖ SYNTAX:

GRANT SYSTEM PRIVILEGE / ROLE TO USER / ROLE /
PUBLIC WITH ADMIN OPTION

ARGUMENTS	DESCRIPTION
PRIVILEGE	The name of the privilege to be assigned
USER / ROLE	The name of the user or role that is received the privilege.
WITH ADMIN OPTION	Allows a user or role that receives the privileges to assign it to other users, change it, or even delete it.

GRANT

ARGUMENT	DESCRIPTION
SELECT	To select data in a table or view.
INSERT	To insert rows in a table or view.
DELETE	To eliminate rows in a table or view
UPDATE	To update a table and, optionally, update only the specified columns
INDEX	To create or eliminate the indexes of a table
ALTER	To modify a table
ALL	To perform all the above privileges
Table_name	Names of existing tables (including the qualifier) in the database
View_name	Names of existing views (including the qualifier) in the database
Column_name	This is optional and determines the column of tables or views specified in the ON clause. The names of the columns must not be qualified.
USER	Name of the user to whom the privilege is assigned
PUBLIC	Means that all the current and new user have the privileges specified for the table or view
WITH ADMIN OPTION	Allows the user or role that receives the privilege to grant it to other users, modify it, or even delete it.

GRANT EXAMPLE

- ❖ GRANT CONNECT TO USER1;
- ❖ GRANT INSERT, UPDATE ON CUSTOMER TO USER1;



REVOKE

- ❖ This command use to remove privileges from a specific user or role from all users.
- ❖ The following types of privileges can be revoked:
 - ❖ Role: Connect resource & DBA.
 - ❖ To insert data into a specific table.
 - ❖ To update data into a specific table.
 - ❖ To Display data of any table.
 - ❖ To delete data from a specific table.



REVOKE

❖ SYNTAX:

```
REVOKE <privilege> | ROLE ON [<object name>] FROM  
<user name>;
```

❖ EXAMPLE:

```
REVOKE CONNECT FROM USER1;  
REVOKE INSERT, UPDATE, ON CUSTOMER FROM USER1;
```



ROLES

- ❖ A role is a group of privileges assigned to a name.
- ❖ Instead of granting eight privileges to a user, you can create a role that is assigned those eight privileges, and then assign that role to the user.
- ❖ They are mainly three types of oracles predefined Roles available for any user:
 - ❖ Connect Role.
 - ❖ Resource Role.
 - ❖ DBA Role.



Mac security



MAC

- ❖ Mandatory Access Control .
- ❖ MAC based on system-wide policies that cannot be changed by individual users.
- ❖ Its a central authority manages resource access according to classifications and clearance levels.
- ❖ It is used to enforced multilevel security by classifying the data and user into various security classes or levels and then implementing the appropriate security policy of the organization.



MAC

- ❖ To ensure the safety and security of critical information, MAC assigns labels and clearances to both users and objects.
- ❖ It uses a hierarchical model of classifications and clearances.
- ❖ Security administrators assign users matching clearance levels, determining what they can access.
 - ❖ For instance, someone with a Secret clearance can access Confidential data but is barred from top-secret information.



MAC

- ❖ The commonly used MAC technique for multilevel security is known as the **Bel- LaPadula model**.
- ❖ The **Bel-LaPadula model** is described in terms of Subject (Users, Accounts, Programs), Objects (Relations or Tables, Tuples, Attributes, Views, Operations) and the level of clearance for a particular user.
- ❖ This model classifies each subject and object into one of the security levels such as TS, S, C, U.
- ❖ The security classes in a system are organized according to a particular order, with a most secure class or level and a least secure class or level.



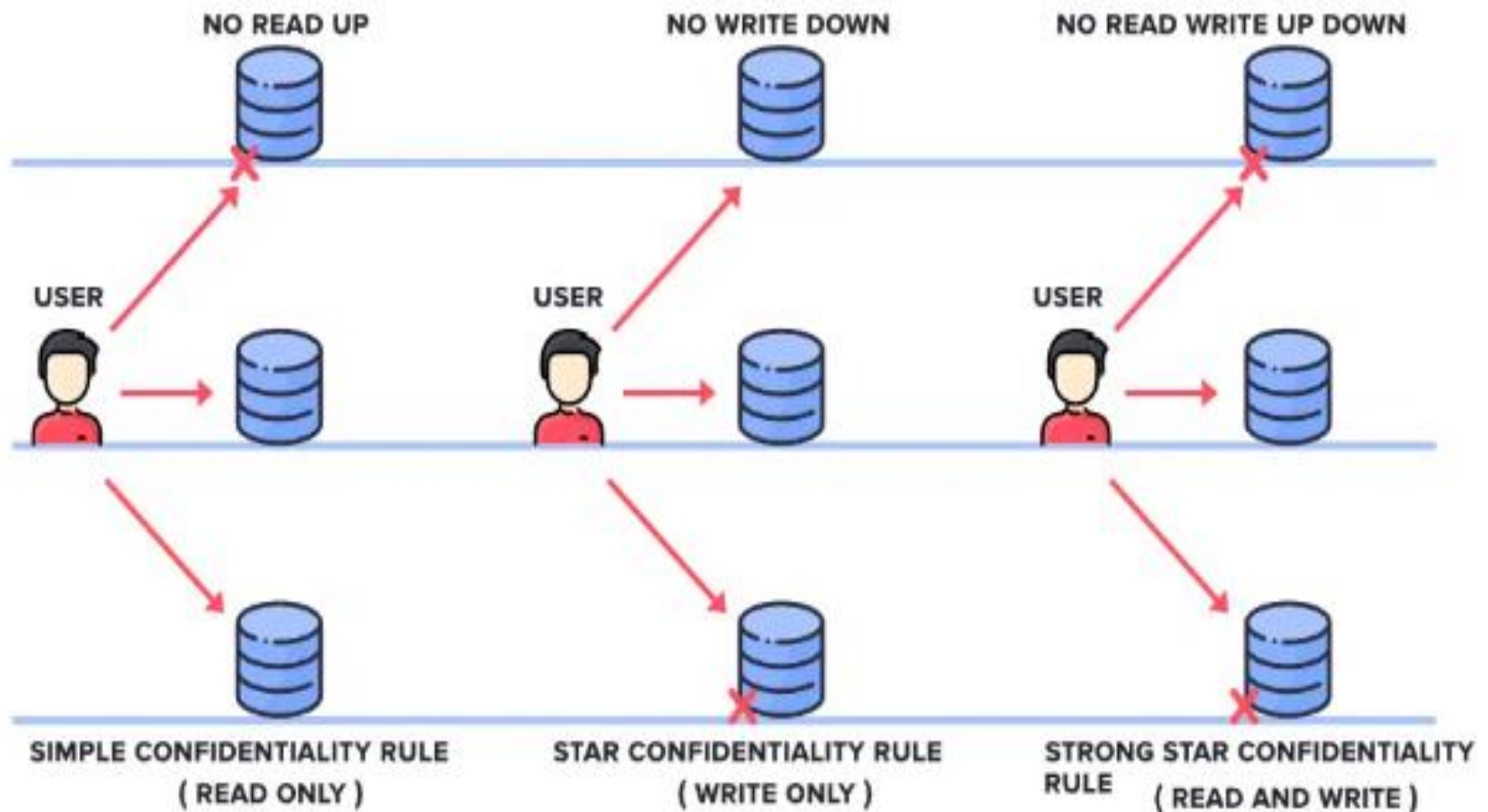
MAC

- ❖ The commonly used MAC technique for multilevel security is known as the **Bel- LaPadula model**.
- ❖ **This model “no read up, no write down” rule** – users can’t read data above their clearance level or write data to a lower level, guaranteeing tight control over information flow and preventing data leaks.

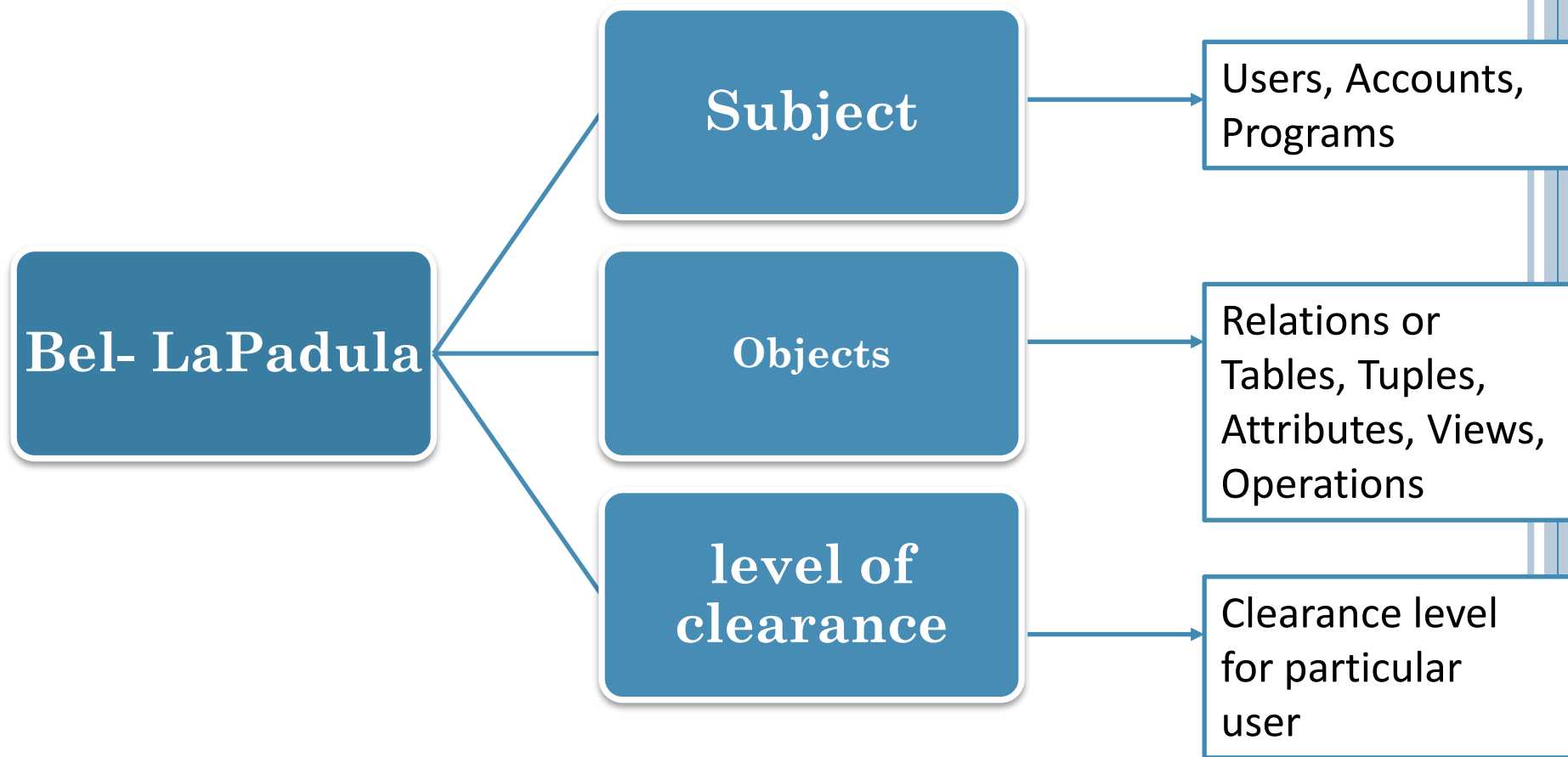


BEL- LAPADULA MODEL.

BELL - LAPADULA MODEL

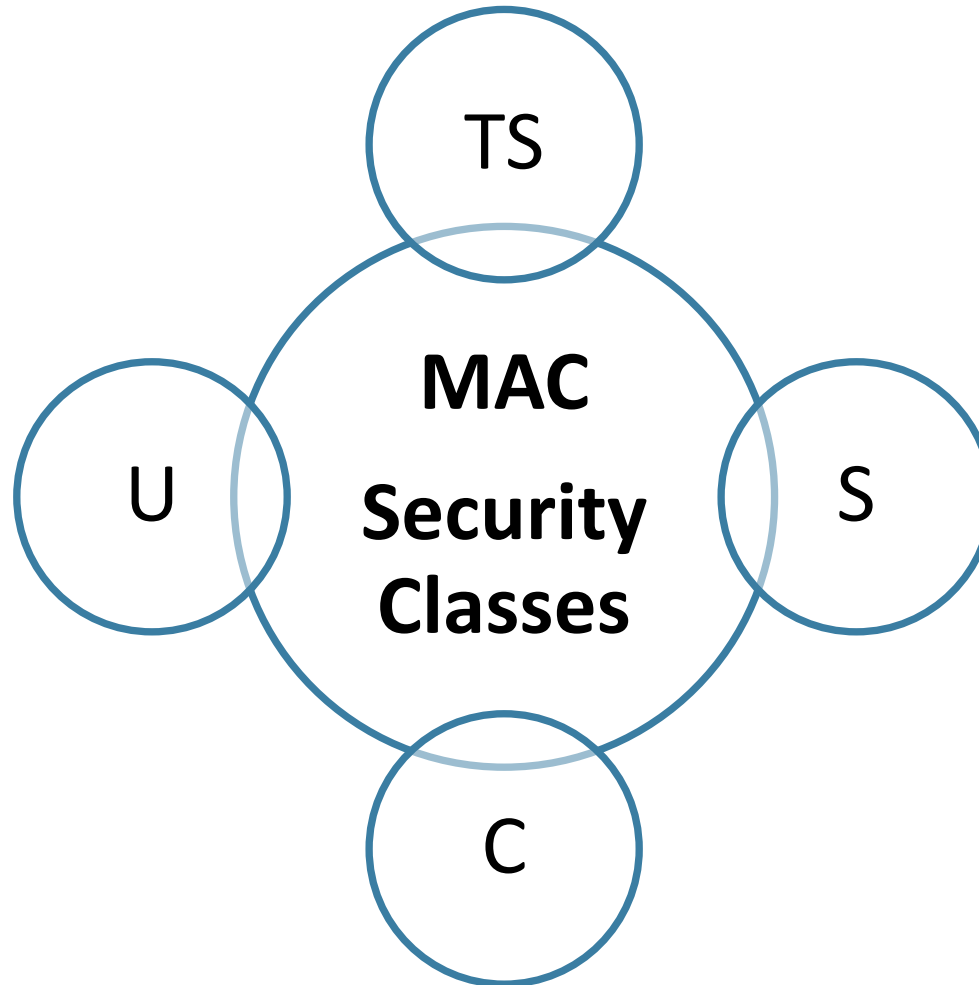


MAC



MAC

- ❖ MAC technique classifies data and users based on security classes.



MAC

- ❖ The security classes in a system are organized according to a particular order, with a most secure class or level and a least secure class or level.
- ❖ This model enforces following two restrictions on data access based on the subject and object classification.
- ❖ **Simple Security Property:**
 - ❖ In this case, a subject S is not allowed read access to an object O unless classification of subject S is greater than or equal to classification of object O .
 - ❖ In other word **$\text{class}(S) \geq \text{class}(O)$** .



MAC

❖ Star Security Property:

- ❖ In this case, a subject S is not allowed to write an object O unless classification of subject S is less than or equal to classification of an object O .
- ❖ In other word **$\text{class}(S) \leq \text{class}(O)$** .



ENCRYPTION AND PUBLIC KEY INFRASTRUCTURES

- ❖ Encryption is the conversion of data into a form, called a **ciphertext**, which cannot be easily understood by unauthorized persons.
- ❖ It enhances security and privacy when access controls are bypassed, because in cases of data loss or theft, encrypted data cannot be easily understood by unauthorized persons.



ENCRYPTION AND PUBLIC KEY INFRASTRUCTURES

- ❖ **Ciphertext:**

- ❖ Encrypted (enciphered) data.

- ❖ **Plaintext (or cleartext):**

- ❖ Intelligent data that has meaning and can be read or acted upon without the application of decryption.

- ❖ **Encryption:**

- ❖ The process of transforming plaintext into ciphertext.

- ❖ **Decryption:**

- ❖ The process of transforming ciphertext back into plaintext



THE DATA ENCRYPTION AND ADVANCED ENCRYPTION STANDARDS

- ❖ The Data Encryption Standard (DES) is a system developed by the U.S. government for use by the general public.
- ❖ It has been widely accepted as a cryptographic standard both in the United States and abroad. DES can provide end-to-end encryption on the channel between sender A and receiver B.
- ❖ The DES algorithm is a careful and complex combination of two of the fundamental building blocks of encryption: substitution and permutation (transposition).



THE DATA ENCRYPTION AND ADVANCED ENCRYPTION STANDARDS

- ❖ After questioning the adequacy of DES, the NIST introduced the Advanced Encryption Standard (AES).
- ❖ AES introduces more possible keys, compared with DES, and thus takes a much longer time to crack.



SYMMETRIC KEY ALGORITHMS

- ❖ A symmetric key is one key that is used for both encryption and decryption.
- ❖ By using a symmetric key, fast encryption and decryption is possible for routine use with sensitive data in the database.
- ❖ A message encrypted with a secret key can be decrypted only with the same secret key.
- ❖ Algorithms used for symmetric key encryption are called secret-key algorithms.
- ❖ Since secret-key algorithms are mostly used for encrypting the content of a message, they are also called content-encryption algorithms.



SYMMETRIC KEY ALGORITHMS

- ❖ A symmetric key is one key that is used for both encryption and decryption.
- ❖ By using a symmetric key, fast encryption and decryption is possible for routine use with sensitive data in the database.
- ❖ A message encrypted with a secret key can be decrypted only with the same secret key.
- ❖ Algorithms used for symmetric key encryption are called secret-key algorithms.
- ❖ Since secret-key algorithms are mostly used for encrypting the content of a message, they are also called content-encryption algorithms.



PUBLIC (ASYMMETRIC) KEY ENCRYPTION (PKI)

- ❖ In 1976, Diffie and Hellman proposed a new kind of cryptosystem, which they called public key encryption.
- ❖ Public key algorithms are based on mathematical functions rather than operations on bit patterns.
- ❖ They address one drawback of symmetric key encryption, namely that both sender and recipient must exchange the common key in a secure manner.



PUBLIC (ASYMMETRIC) KEY ENCRYPTION (PKI)

- ❖ In public key systems, two keys are used for encryption/decryption.
- ❖ The public key can be transmitted in a non-secure way, whereas the private key is not transmitted at all.
- ❖ The two keys used for public key encryption are referred to as the public key and the private key.



PUBLIC (ASYMMETRIC) KEY ENCRYPTION (PKI)

❖ A public key encryption scheme, or infrastructure, has six ingredients:

- 1. Plaintext**
- 2. Encryption algorithm**
- 3. and 4. Public and private keys**
- 5. Ciphertext**
- 6. Decryption algorithm**



THE ESSENTIAL STEPS ARE 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 the two keys in a public register or other accessible file. This is the public key. The companion key is kept private.
3. If a sender wishes to send a private message to a receiver, the sender encrypts the message using the receiver's public key.
4. When the receiver receives the message, he or she decrypts it using the receiver's private key. No other recipient can decrypt the message because only the receiver knows his or her private key.

