Three level Architecture

Database Management System

- Database Management system is a software which control the data in database and users of database.
- DBMS is a software that interacts with the users, application programs and the database.
- A database management is a software that is used to create, maintain, and provide controlled access to user databases.
- DBMS provides an environment that is both convenient and efficient to use.

Component of DBMS Environment

- Hardware
- Software
- Data
- Procedure
- People

Functions of DBMS

- Data Processing
- User Accessible Catalog
- Concurrency Control Services
- Recovery Services
- Authorization Services
- Support for data Communication
- Integrity Services

Functions of DBMS

Data Processing

- DBMS perform different operation about data like
  - Creation of database
  - Putting data into databases
  - Arranging data into different form
  - Accessing data
- All of these activities are included at processing of data and all these activities are performed by DBMS.

Functions of DBMS

User Accessible Catalog

- Catalog mean structure of database or meta data.
- Catalog is a place or repository where information describing in database are store.
- Catalog stores
  - Data about all users of database
  - Architecture of database
- DBMS must provide a catalog in which description of data items are stored and which is accessible to users.
-

Functions of DBMS

Concurrency Control Services

- Multiple user access the database at the same time
- Concurrent approach is good but very sensitive it make database **inconsistent**
- DBMS provide concurrent services to the users of database and provide the proper control so that interference cannot occur.
-

Functions of DBMS

Recovery Services

- A database can be damaged or destroyed in many ways, e.g database can end abnormally during an update, hardware problem can occur; and so on. When these event has occurred, database become inconsistent.
- DBMS must provide methods to recover a database to a contestant state.

Functions of DBMS

Authorization Services

- DBMS must provide a mechanism to ensure that only authorized users can access the database. It also provide the protection of database against unauthorized access.

Functions of DBMS

Support for data Communication

- DBMS must be capable of integrating with communication software.

Functions of DBMS

Integrity Services

- Database Integrity mean correctness of stored data.
- Integrity is considered in the term of **constraints**.
- Place different checks on data which ensures the correctness of database. So responsibility of DBMS is to take care of those checks.

DBMS User Environment

- Two main type of DBMS user Environment
  - **Single User DBMS Environment**
  - **Multi-User DBMS Environment**
    - **Teleprocessing**
    - **File Server**
    - **Client Server**

DBMS User Environment

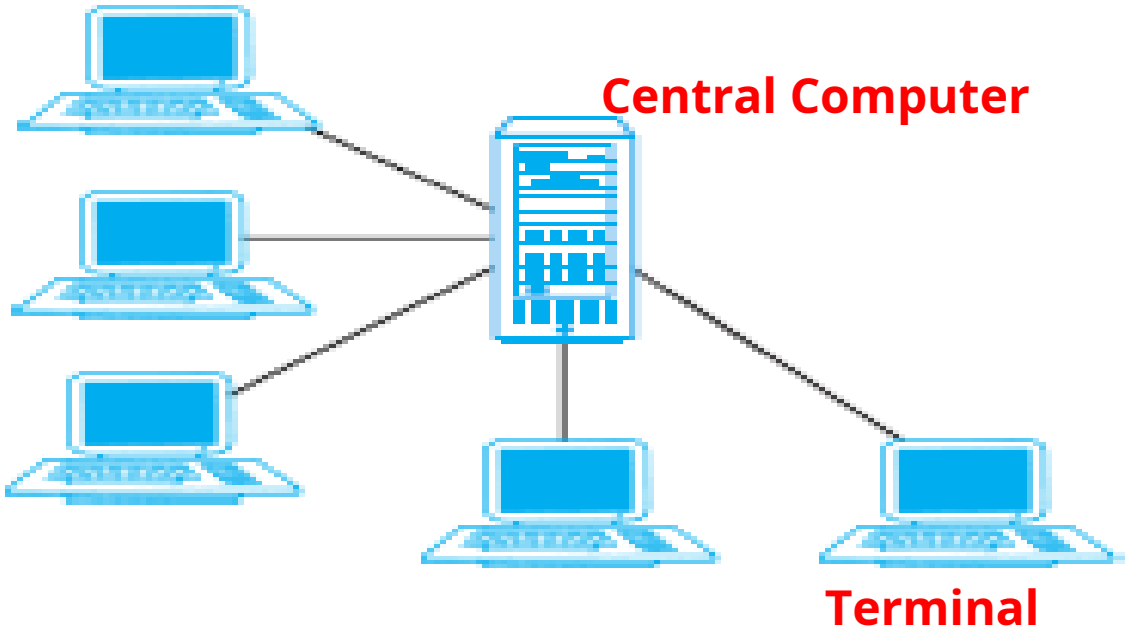
Single User DBMS Environment

- Normally called Desktop DBMS
- Only Single user can access DBMS or environment can support one user at a time.

DBMS User Environment

Multi-User DBMS Environment

- **Teleprocessing**



DBMS User Environment

Multi-User DBMS Environment

- **Teleprocessing**
  - Very old Architecture of multi-user environment
  - All processing is performed at only a central computer
  - The attached devices are called Terminal or **Dumb terminal**
  - Dumb terminal are only use to view the result or typing some thing.
  - All the processing by DBMS and application seen on dumb terminal are run on at central computer

DBMS User Environment

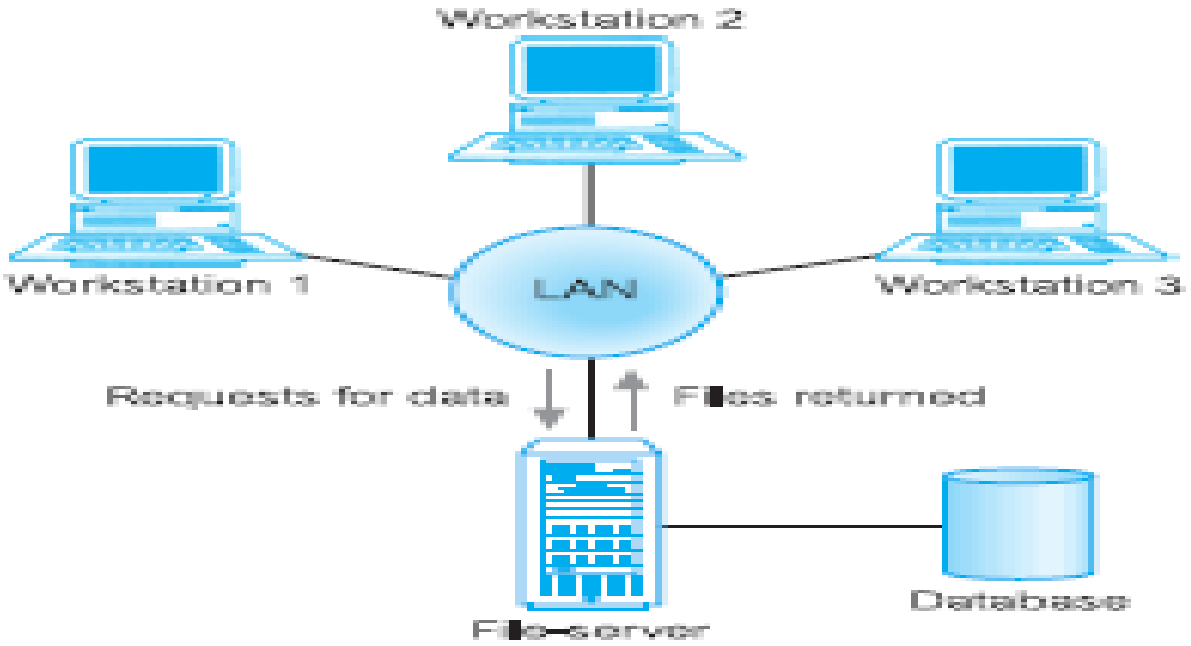
Multi-User DBMS Environment

- **Teleprocessing**
- **Drawback:**
  - Too much burden on central computer
  - Central computer become slow down
- **Solution**
  - To overcome the drawback of teleprocessing, File server environment was introduced.

DBMS User Environment

Multi-User DBMS Environment

- **File Server Environment**



DBMS User Environment

Multi-User DBMS Environment

- **File Server Environment**
  - The processing is distributed about the network.
  - Data storage was the responsibility of File server.
  - Application and copy of DBMS run on each workstation.
  - When user need any data it request from workstation to file server, the file server rather sending only that data the complete file was send to the workstation. All the files are store at file server and all the processing is done at workstation.

DBMS User Environment

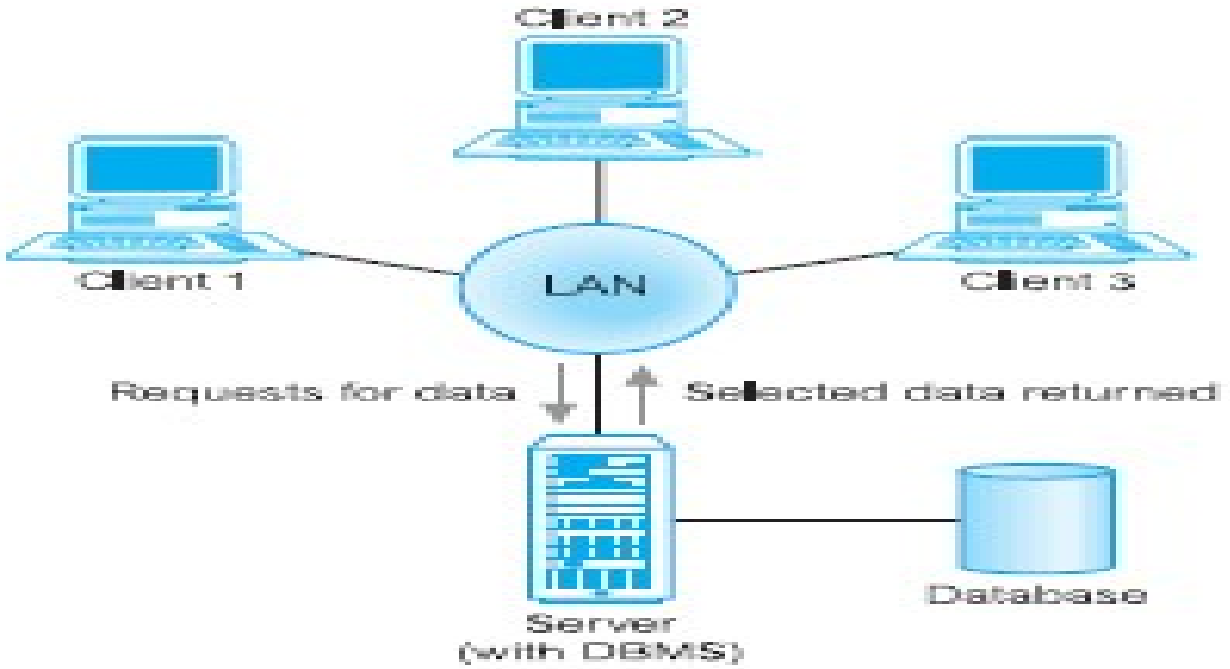
Multi-User DBMS Environment

- **File Server Environment**
- **Drawback**
  - Network Over burden
  - Full copy of DBMS is required on each workstation.
  - Concurrency control become very complex
- **Solution**
  - To overcome the drawback of File server, Client Server Architecture was developed.

DBMS User Environment

Multi-User DBMS Environment

- **Client Server Environment**



DBMS User Environment

Multi-User DBMS Environment

- **Client Server Environment**
  - DBMS run only on server machine and no copy of DBMS is on the client computer
  - Client computer manages the user interface and application programs, which means that application program are run on client computers.
  - If user want to access any data, 1st query is checked at local machine then send it to server where the DBMS check either request is valid or not and then specific result are send back to client.

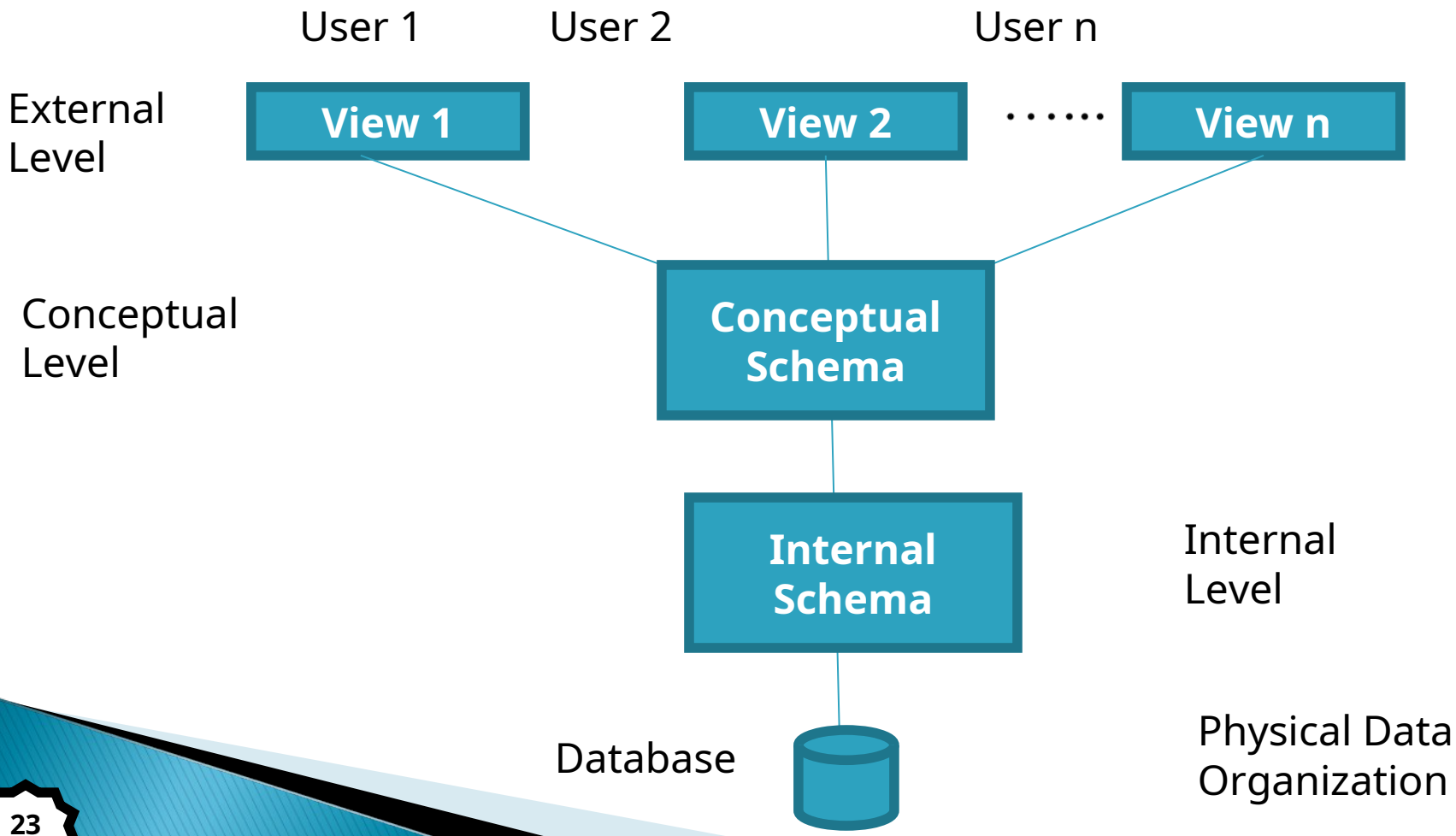
DBMS User Environment

Multi-User DBMS Environment

- **Client Server Environment**
- **Advantages**
  - Concurrency control
  - Increase performances
  - Reduced network traffic

Three Level Architecture

Three Schema Component



Three Level Architecture

Schemas Mapping

- Link between two level is establish by schema mapping.
- Data item or object of one level interlink with the object of other level is called schema mapping
- Why???

Three Level Architecture

Schemas Mapping

- Two type of schema mapping
  - Conceptual Internal Mapping
  - External Conceptual Mapping

Three Level Architecture

Schemas Mapping

- **External Conceptual Mapping**
- Create link between external level and conceptual level.
- Specifies mapping between objects in the external view to those in the logical model.

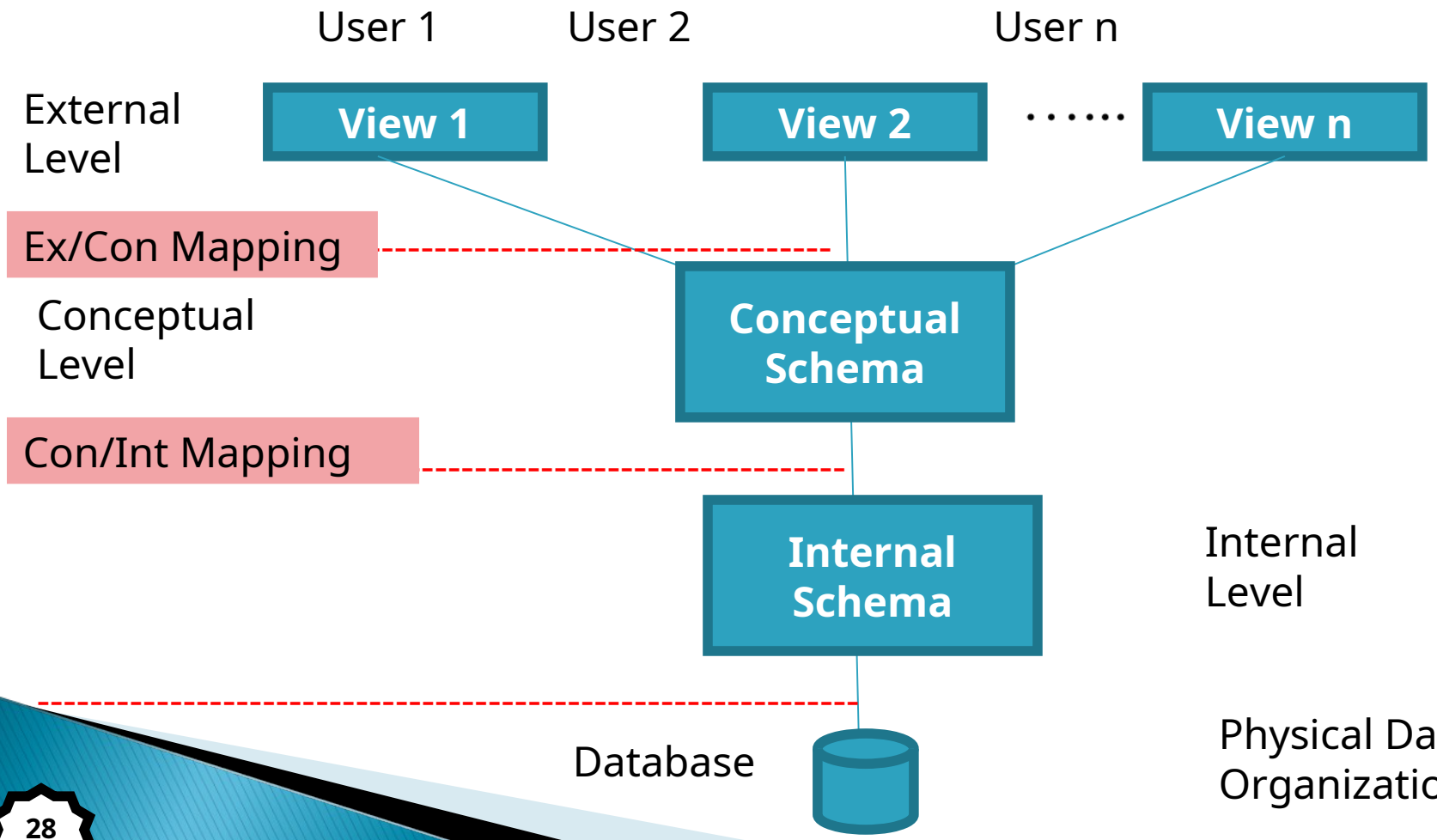
Three Level Architecture

Schemas Mapping

- **Conceptual Internal Mapping**
- Specifies mapping between objects in the logical model to those in the physical data model.

Three Level Architecture

Three Schema Component



Data Independence

- Outcome of three level architecture is data independence.
- Change in lower level do not affect the upper level.
- Type of data Independence
  - **Logical Data Independence**
  - **Physical data Independence**

Data Independence

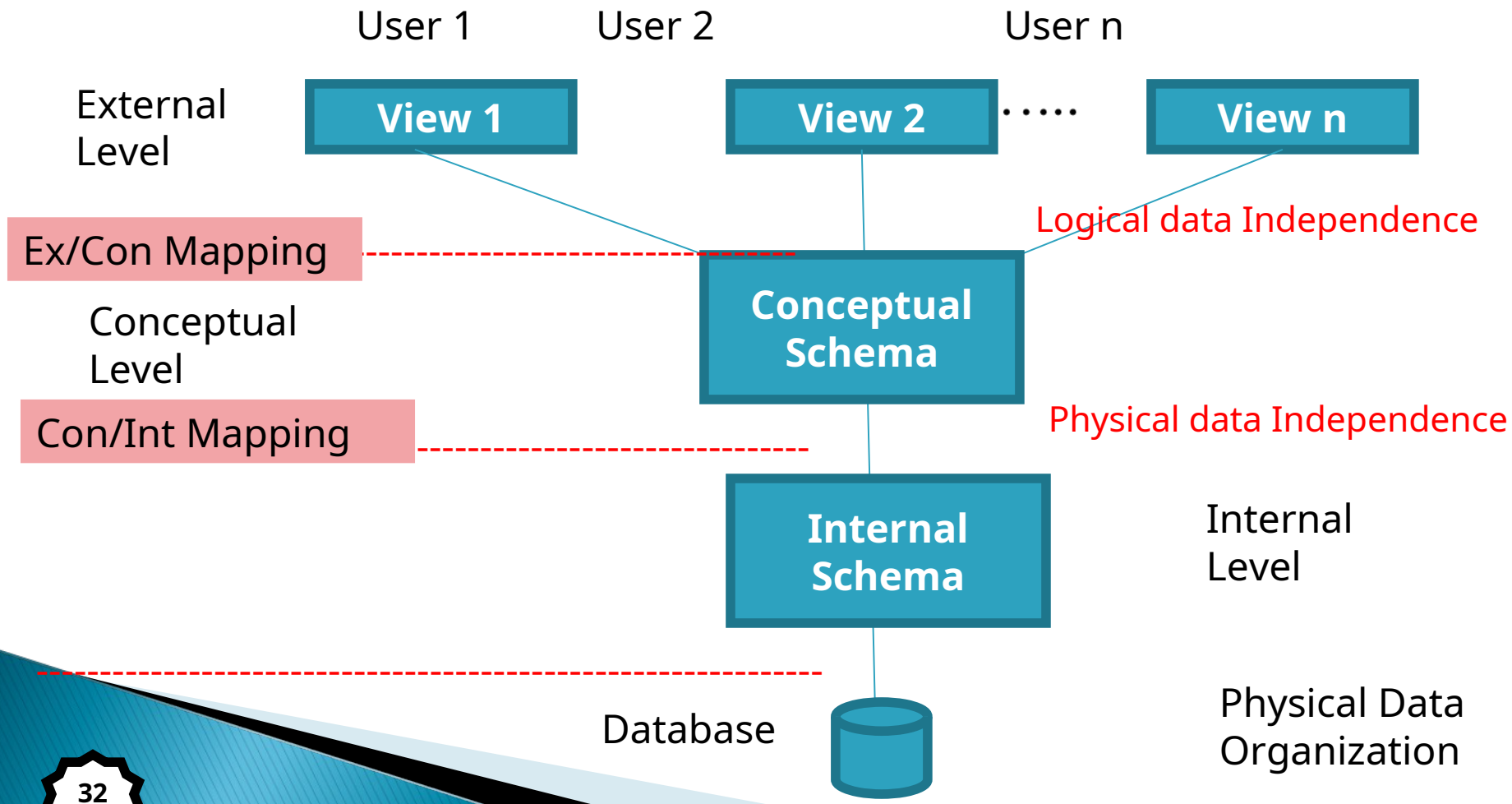
- **Logical Data Independence**
- Changes in conceptual model do not affect external view.
- Logical data independence is the Immunity/resistance of the external schema to changes in the conceptual schema.
- **Example**
  - Adding new file
  - Adding new fields
  - Change the type of Attribute
  - Delete the Attribute

Data Independence

- **Physical Data Independence**
- Changes in Internal model do not affect conceptual model.
- Physical data independence is the Immunity/resistance of the conceptual schema to changes in the Internal schema.
- **Example**
  - Changing in file organization
  - Index Implementation

Three Level Architecture

Three Schema Component



Database Schema

- It is the skeleton structure that represents the logical view of the entire database.
- It defines how the data is organized and how the relations among them are associated.
- It defines all the constraints that are to be applied on the data.

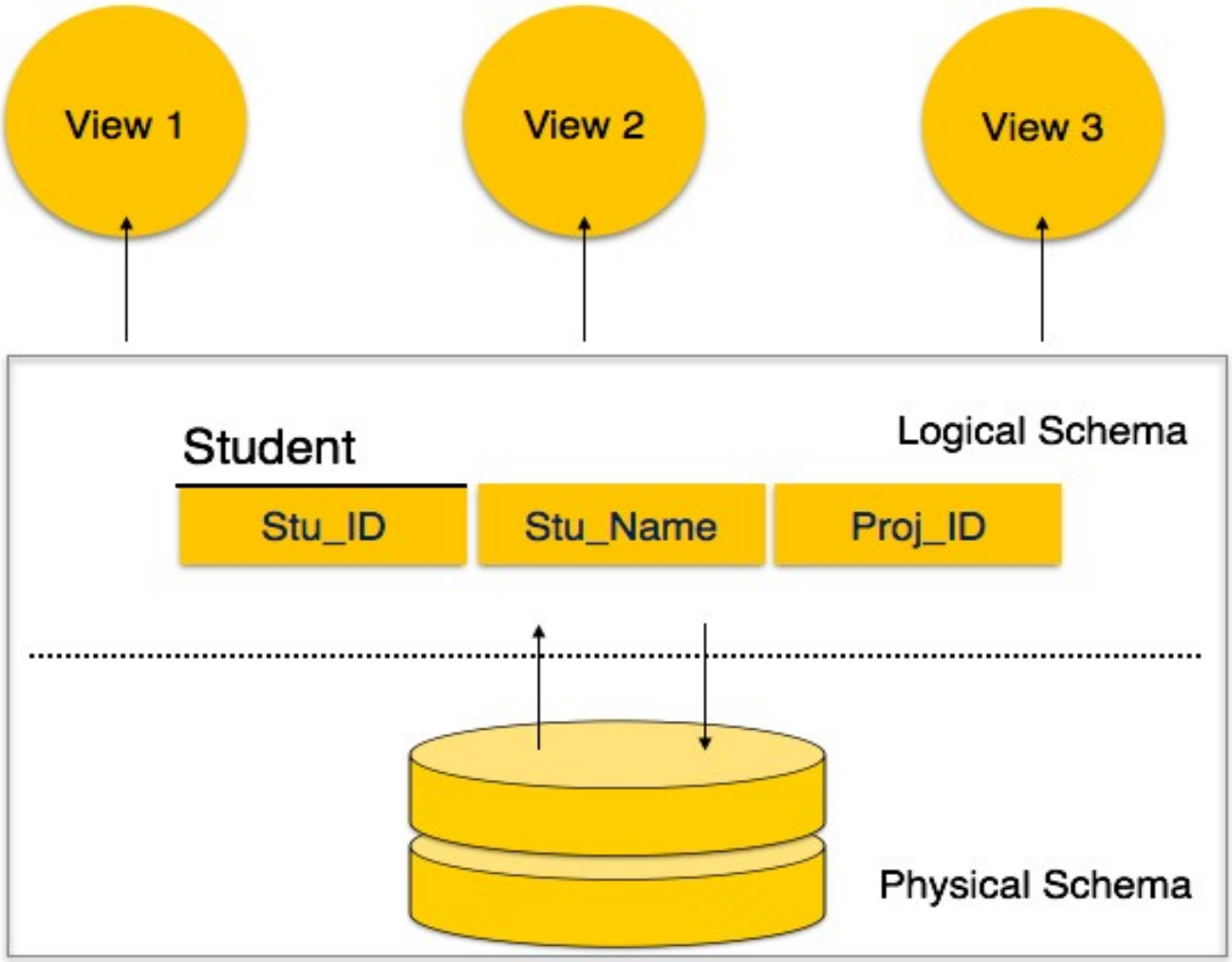
Database Schema

- A database schema defines its entities and the relationship among them.
- The database designers who design the schema to help programmers understand the database and make it useful.

Database Schema

- A database schema can be divided broadly into two categories:
- **Physical Database Schema**
  - This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.
- **Logical Database Schema**
  - This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

Database Schema



Database Schema

- Database schema is the skeleton of database.
- It is designed when the database doesn't exist at all.
- A database schema does not contain any data or information.

Database Instance

- A database instance is a state of operational database with data at any given time.
- It contains a snapshot of the database.
- Database instances tend to change with time. A DBMS ensures that its every instance (state) is in a valid state.

Database Instance

- A database instance describes a complete database environment, including the RDBMS software, table structure, stored procedures and other functionality.
- Oracle instance can access only one database at a time, while an Oracle database can be accessed by multiple instances.
- For example, if you sign into an Oracle database, your login session is an instance. If you log off or shut down your computer, your instance disappears

Database Instance

Analogy

	Real World	Database
Scheme	 Plan for a Standard-House	P-ID | Name | Prename Template for a Table
Instances	 Built Standard-Houses	P-ID | Name | Prename 102356 | Smith | John 102357 | Potter | Harry Prename William 523646 | Wood | Lucinda Data-filled Tables

Database Design

- Database design or database model
- Database design represent the **logical structure** of database
- The process of creating a logical representation of the structure of database is called database design.
- It should be representing graphically
- The goal is to identifying the fact to be store in database

Database Design

- Data modeling involves users and analyst
- Database design is created using a **data model**.
- Data models define how data is connected to each other and how they are processed and stored inside the system.
- Data models define how the logical structure of a database is modeled.
- Data Models are fundamental entities to introduce **abstraction in a DBMS.**

Data Model

- Data model is a set of tools that are used to develop a database design.
- Data model is used to develop a database design
- Graphical system used to capture the nature and relationship among data

Data Model

- Data models are created at **enterprise level** as well as **project levels**
- Data model organizes data for various users
- Every DBMS is based on Data Model

Data Model

Component of Data Model

- **Structure Part**
- **Manipulation Part**
- **Integrity Constraints Supports**

Data Model

Type of Data Model

- Semantic Data Model
- Records Based Data Model
- Physical Data Model

Data Model

Semantic Data Model

- The model which provide more tool
- They only deal at External Level and Conceptual Level

- Type of Semantic Data Model
  - Object Based Data Model
  - Entity Relationship Data Model

Data Model

Records Based Data Model

- The database consists of a number of fixed-format records and each record type having fixed number of fields.
- Type of Records Based data model are
  - Relational Data Model
  - Network Data Model
  - Hierarchical Data Model.

Data Model

Records Based Data Model

- **Relational Data Model**
  - Based on mathematical relation
  - **Tables** are used to represent Data and Relationships

Data Model

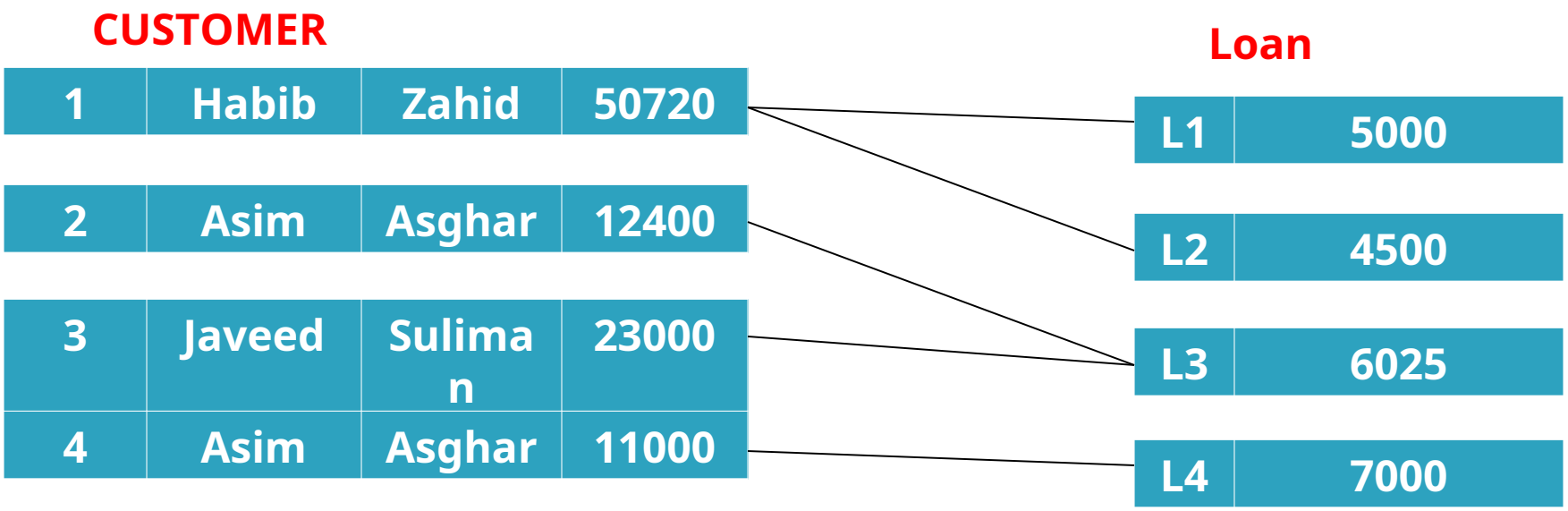
Records Based Data Model

- **Network Data Model**
  - Sometimes called the DBTG model.
  - Data is stored in records; each record consists of a group of related data values**.**
  - Each record can have several child and each child record can have more then one parent records.
  - The IDS database architecture based on network data model.

Data Model

Records Based Data Model

- **Network Data Model**



Data Model

Records Based Data Model

- **Network Data Model**
  - Parent Record is called Owner
  - Child Record is called Member
  - Support 1-1, 1-M & M-M relationship
  - Support **recursive relationship**

Data Model

Records Based Data Model

- **Network Data Model**
  - **Disadvantages**
  - Structure should must be defined in advanced which is difficult task.
  - It is very complex to understand and developed

Data Model

Records Based Data Model

- **Hierarchical Data Model**
  - Use two data structuring concepts: **records** and **parent-child relationships**
  - A record is a collection of field values that provide information on an entity or a relationship instance.

Data Model

Records Based Data Model

- **Hierarchical Data Model**
  - Use two data structuring concepts: **records** and **parent-child relationships**
  - A record is a collection of field values that provide information on an entity or a relationship instance.
  - A parent-child relationship type (PCR type) is a 1:M relationship between two record types.

Data Model

Records Based Data Model

- **Hierarchical Data Model**
  - Hierarchical data model is **tree like structure** database and data is store in the form of branches.
  - The IMS database architecture based on Hierarchical data model.
  - Another popular hierarchical DBMS is MRI's System-2000

Data Model

Records Based Data Model

- **Hierarchical Data Model**
  - In hierarchical data model, high level record is called parent
  - The parent at top most position is called root.
  - A lower level record is called child
  - A parent may have more then one child but a child can have only one parent.

Data Model

Records Based Data Model

- **Hierarchical Data Model**
  - This model can support 1:1 and 1:M relationship directly
  - It can not support M:M relationship and recursive relationship directly.
  - The IMS database architecture based on Hierarchical data model.
  - Another popular hierarchical DBMS is MRI's System-2000

Data Model

Records Based Data Model

- **Hierarchical Data Model**
  - **Advantages**
  - Flexible for changing the structure
  - Insertion, deletion and modification is easy
  - Little skill is required to developed and understand
  - Accessing and updating is very fast

Data Model

Records Based Data Model

- **Hierarchical Data Model**
  - **Disadvantages**
  - Structure should must be defined in advanced which is difficult task.
  - Adding new records to database need to redefined the whole database
  - No relationship between child records

Have a Nice Day