Chapter 1: Introduction

- **Data:** Known facts that can be recorded and that have implicit meaning
- **Information**:-processing of data.

- **Data Base:-** Collection of related data.
- Eg. the names, telephone numbers and addresses of all the people you know. Database are organized by fields, records and files

Benefits of database approach(Data base system) over file system

- Redundancy can be reduced
- Inconsistency can be avoided
- Data can be shared
- Standards can be enforced
- Security restrictions can be applied
- Integrity & quality can be maintained
- Data independence can be provided
- Backup and Recovery.
- Data retrieval become efficient.
- Concurrency control.
- Transactional Problem can be removed.
- Economic on scale

Database Applications:

- Banking: all transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions

- Databases touch all aspects of our lives

COMPONENTS OF DATABASE SYSTEM

- A database system is composed of four components;
- • Data
- • Hardware
- • Software
- • Users

- . **Data** -. A data is known as the rowfact which is stored in computers memory and used by the user.

- **Data may be of different types.**
- • **User Data** - It consists of a table(s) of data called Relation(s) where Column(s) are called fields of attributes and rows are called Records for tables. A Relation must be structured properly.

- • **Metadata** - A description of the structure of the database is known as Metadata. It basically means "data about data". System Tables store the Metadata which includes.
- - Number of Tables and Table Names
- - Number of fields and field Names
- - Primary Key Fields

- • **Application Metadata** - It stores the structure and format of Queries, reports and other applications components.

- **2. Hardware** - The hardware consists of the secondary storage devices such as magnetic disks (hard disk, zip disk, floppy disks), optical disks (CD-ROM), magnetic tapes etc. on which data is stored together with the Input/Output devices (mouse, keyboard, printers), processors, main memory etc. which are used for storing and retrieving the data in a fast and efficient manner.

- **3. Software** - The Software part consists of DBMS which acts as a bridge between the user and the database or in other words, software that interacts with the users, application programs, and database and files system of a particular storage media (hard disk, magnetic tapes etc.) to insert, update, delete and retrieve data. For performing these operations such as insertion, deletion and updation we can either use the Query Languages like SQL, QUEL, Gupta SQL or application softwares such as Visual Basic, SQLYog Developer etc.

- **4. Users** - Users are those persons who need the information from the database to carry out their primary business responsibilities i.e. Personnel, Staff, Clerical, Managers, Executives etc. On the basis of the job and requirements made by them they are provided access to the database totally or partially.

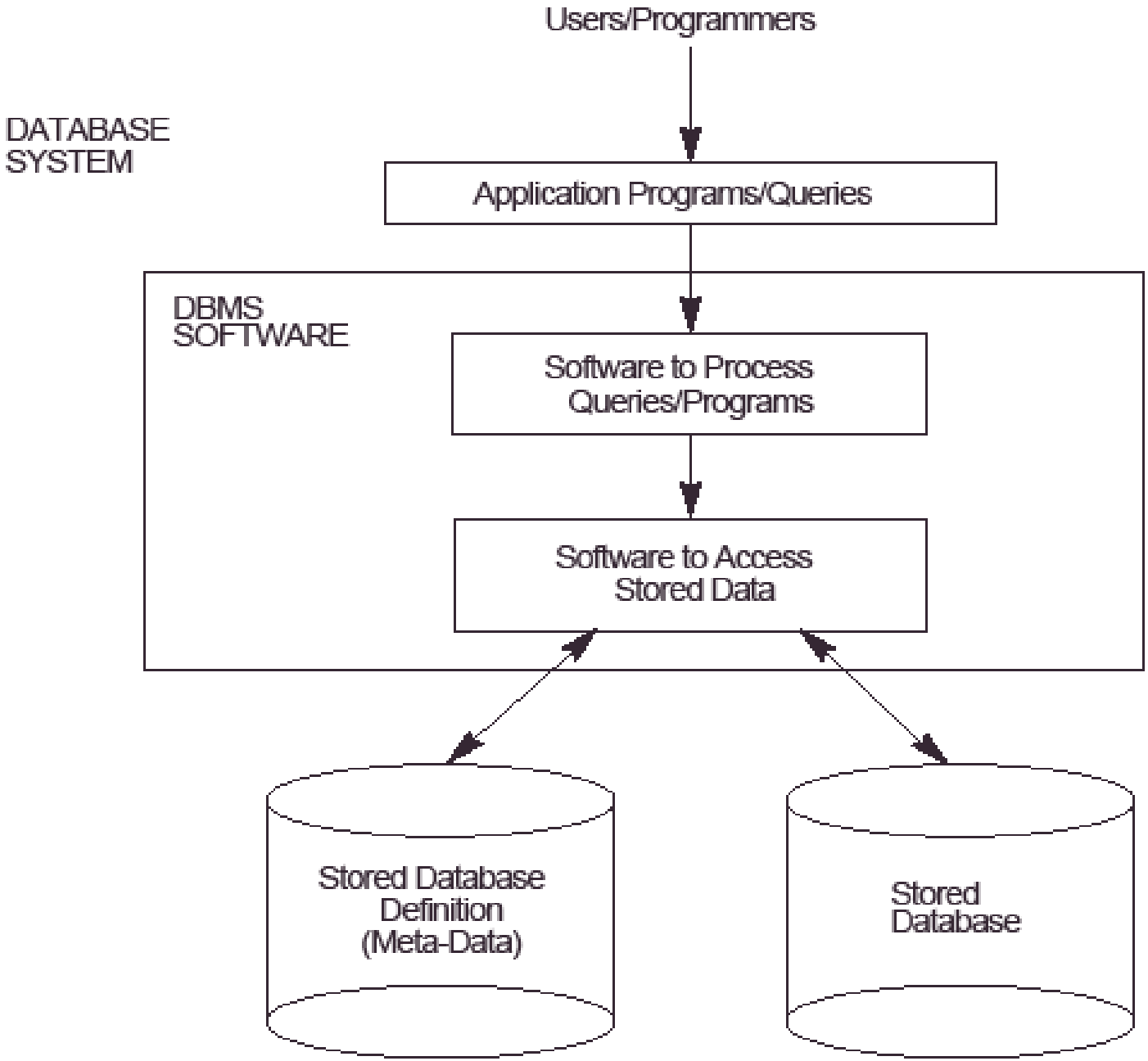
The various types of users which can access the database are:-

  - Database Administrators (DBA)
  - Database Designers
  - End Users
  - Application Programmers

Disadvantage of Database system

- Complexity increase
- Additional cost of hardware
- Cost of conversion
- Need of additional and specialized manpower
- Need for backup and recovery.
- More installation and management cost

concepts and terminology discussed in Section 1.1.



DBMS

- **Database Management System (DBMS)**: *A* software package/ system to facilitate the creation and maintenance of a computerized database

Typical DBMS Functionality

- **Define a database** : in terms of data types, structures and constraints
- **Construct or Load** the Database on a secondary storage medium
- Manipulating the database : querying, generating reports, insertions, deletions and modifications to its content
- Concurrent Processing and Sharing by a set of users and programs – yet, keeping all data valid and consistent

Typical DBMS Functionality

Other features:

- Protection or Security measures to prevent unauthorized access
- Presentation and Visualization of data

Main Characteristics of the Database Approach

- Self-describing nature of a database system: A DBMS **catalog** stores the *description* of the database. The description is called **meta-data**. This allows the DBMS software to work with different databases.
- Insulation between programs and data: Called **program-data independence**. Allows changing data storage structures and operations without having to change the DBMS access programs.

Contd…

- Data Abstraction: A **data model** is used to hide storage details and present the users with a *conceptual view* of the database.
- Support of multiple views of the data: Each user may see a different view of the database, which describes *only* the data of interest to that user.

Contd..

- Sharing of data and multiuser transaction processing : allowing a set of concurrent users to retrieve and to update the database. Concurrency control within the DBMS guarantees that each **transaction** is correctly executed or completely aborted. OLTP (Online Transaction Processing) is a major part of database applications.

Database Users

Users may be divided into those who actually use and control the content (called "Actors on the Scene") and those who enable the database to be developed and the DBMS software to be designed and implemented (called "Workers Behind the Scene").

Actors on the scene

1. **Database administrators:** responsible for

Deciding the conceptual schema or content of Db. Deciding the internal Schema of structure of physical storage.

- Deciding users.
- Deciding userview(external schema)
- Granting of authorities.
- Deciding constraints
- Security.
- Monitoring the performance.
- Backup .

Contd..

2. **End-users:** they use the data for queries, reports and some of them actually update the database content.

Categories of End-users

- **Casual** : access database occasionally when needed
- **Naïve or Parametric** : they make up a large section of the end-user population. They use previously well-defined functions in the form of "canned transactions" against the database. Examples are bank-tellers or reservation clerks who do this activity for an entire shift of operations.

Contd..

- **Sophisticated** : these include business analysts, scientists, engineers, others thoroughly familiar with the system capabilities. Many use tools in the form of software packages that work closely with the stored database.
- **Stand-alone** : mostly maintain personal databases using ready-to-use packaged applications. An example is a tax program user that creates his or her own internal database.

Worker Behind The Scene

- DBMS Designer & Implementers.
- Tool Developers
- Operators & maintenance Personal

1. DBMS system designers and implementer

- DBMS system designers and implementers are persons who design and implement the DBMS modules and interfaces as a software package. A DBMS is a complex software system that consists of many components, **including components for implementing the catalog, query language, interface processors, data access, concurrency control, recovery, and security.** The DBMS must interface with other system software, such as the operating system and compilers for various programming languages.

2. Tool developers design and implement tools—the software packages that facilitate database modeling and design , and improved performance. Tools are optional packages that are often purchased separately. They include packages for database design, performance monitoring, natural language or graphical interfaces, prototyping, simulation, and test data generation.

3. Operators and maintenance personnel (the system administration personnel) are responsible for the actual running and maintenance of the hardware and software environment for the database system.

Advantages of Using the Database Approach

- Controlling redundancy in data storage and in development and maintenance efforts.
- Sharing of data among multiple users.
- Restricting unauthorized access to data.
- Providing Storage Structures for efficient

Contd..

- Providing backup and recovery services.
- Providing multiple interfaces to different classes of users.
- Representing complex relationships among data.
- Enforcing integrity constraints on the database.
- Drawing Inferences and Actions using rules

Additional Implications of Using the Database Approach

- Potential for enforcing standards: this is very crucial for the success of database applications in large organizations Standards refer to data item names, display formats, screens, report structures, meta-data (description of data) etc.
- Reduced application development time: incremental time to add each new application is reduced.

Contd..

- Flexibility to change data structures: database structure may evolve as new requirements are defined.
- Availability of up-to-date information – very important for on-line transaction systems such as airline, hotel, car reservations.
- Economies of scale: by consolidating data and applications across departments wasteful overlap of resources and personnel can be avoided.

Historical Development of Database Technology

- **Early Database Applications:** The Hierarchical and Network Models were introduced in mid 1960's and dominated during the seventies. A bulk of the worldwide database processing still occurs using these models.
- **Relational Model based Systems:** The model that was originally introduced in 1970 was heavily researched and experimented with in IBM and the universities. Relational DBMS Products emerged in the 1980's.

Contd..

- **Object-oriented applications:** OODBMSs were introduced in late 1980's and early 1990's to cater to the need of complex data processing in CAD and other applications. Their use has not taken off much.
- **Data on the Web and E-commerce Applications:** Web contains data in HTML (Hypertext markup language) with links among pages. This has given rise to a new set of applications and E-commerce is using new standards like XML (eXtended Markup Language).

When not to use a DBMS

- **Main inhibitors (costs) of using a DBMS**:
  - High initial investment and possible need for additional hardware.
  - Overhead for providing generality, security, concurrency control, recovery, and integrity functions.
- **When a DBMS may be unnecessary:**
  - If the database and applications are simple, well defined and not expected to change.
  - If there are stringent real-time requirements that may not be met because of DBMS overhead.
  - If access to data by multiple users is not required

Contd..

- **When no DBMS may sufficient:**
  - If the database system is not able to handle the complexity of data because of modeling limitations
  - If the database users need special operations not supported by the DBMS.

Component of DBMS

- DATA DEFINATION LANGUAGE (DDL)
- DATA MANIPULATION LANGUAGE(DML)
- SOFTWARE FOR CONTROLLED ACCESS OF DATABASE

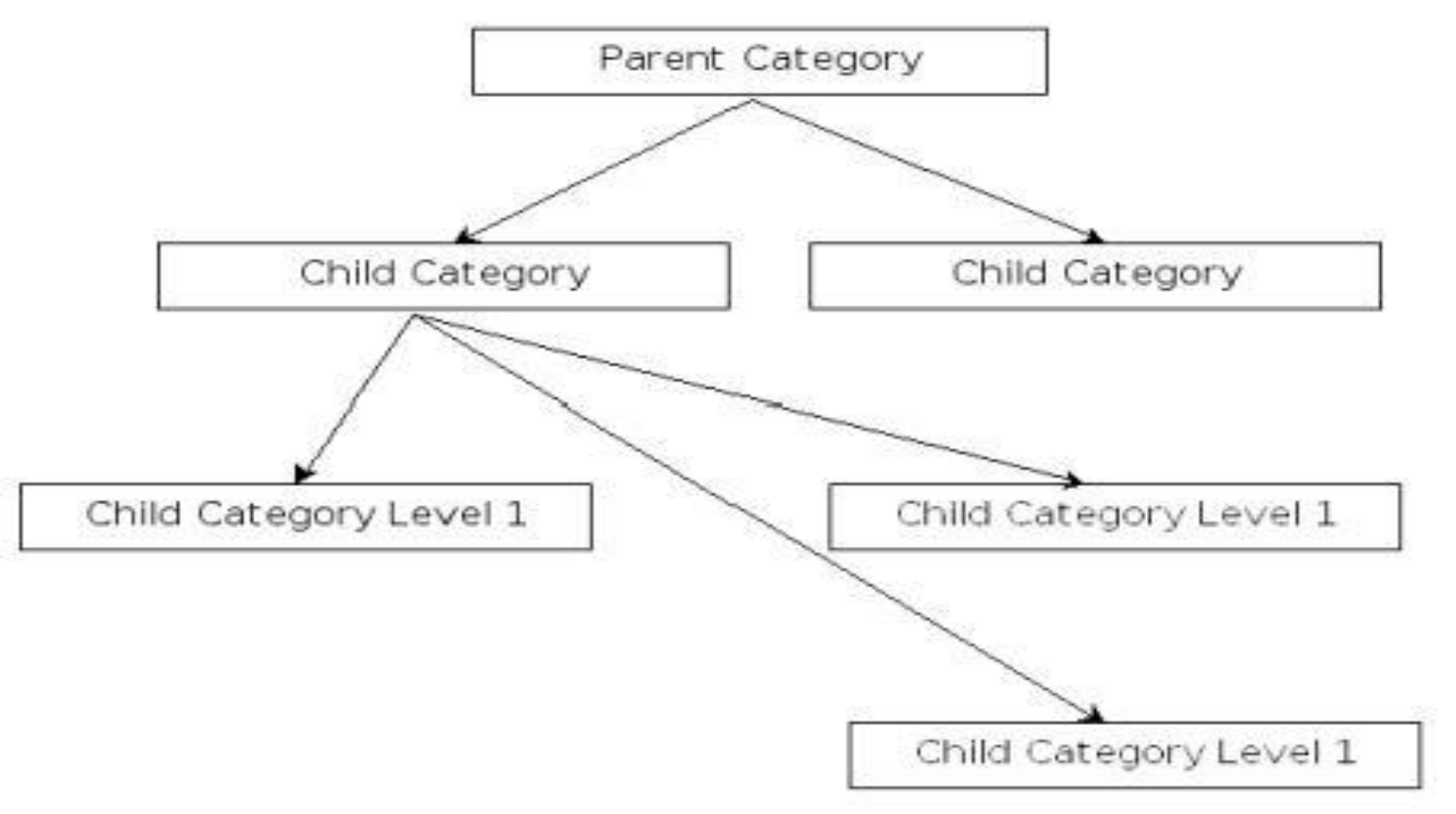
Database System Concepts and Architecture

- **Data Model**: A set of concepts to describe the *structure* of a database, and certain *constraints* that the database should obey

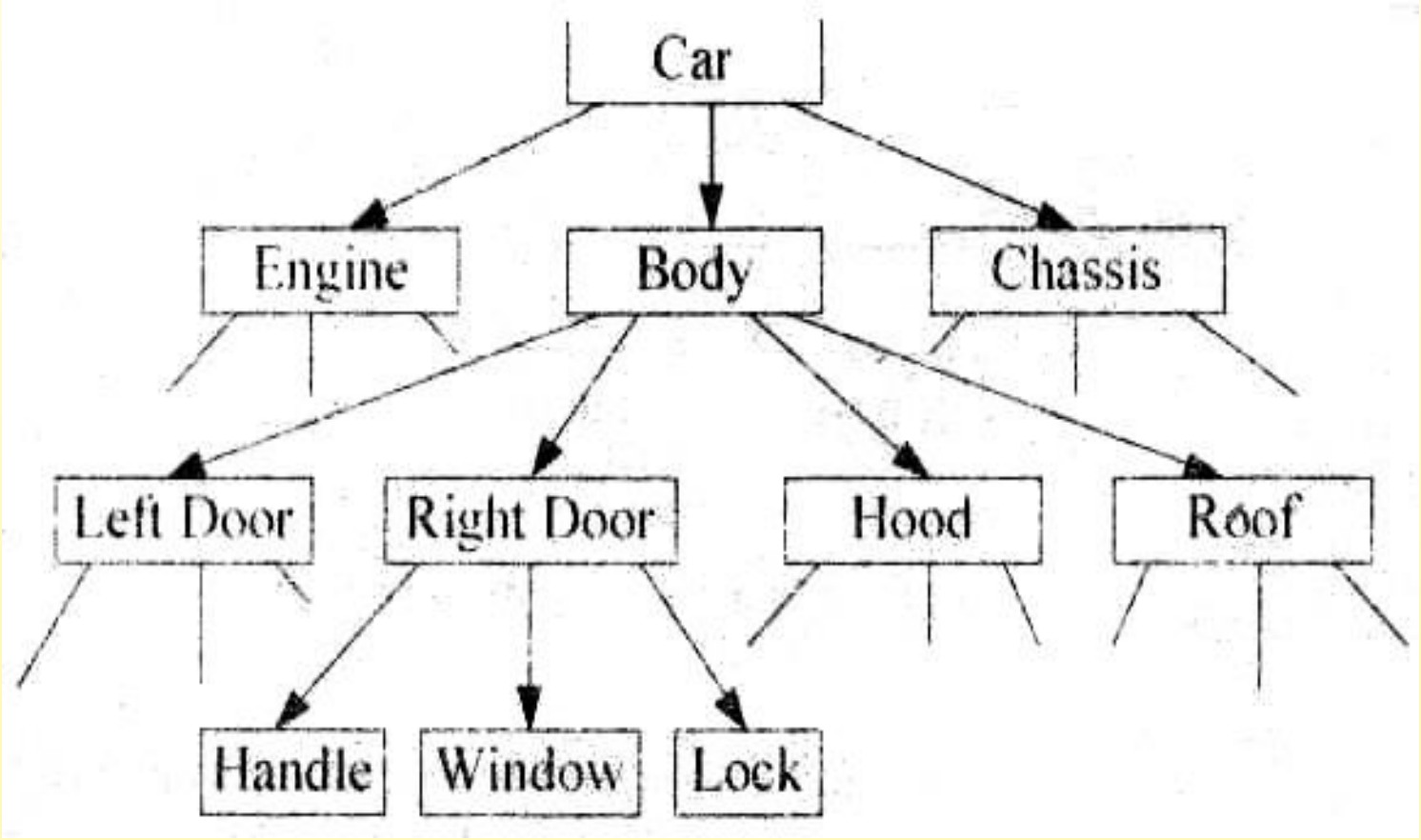
1. Hierarchical Model
2. Network Model
3. Relational Model

Hierarchical Model

- Hierarchical Model is based on tree structure. A Hierarchical Db consists of collection of records, that are connected to each other by links. The root node is dummy node or an empty node



Hierarchical Model



Hierarchical Model

- ADVANTAGES:
  - Hierarchical Model is simple to construct and operate on
  - Corresponds to a number of natural hierarchically organized domains - e.g., assemblies in manufacturing, personnel organization in companies

Contd..

- DISADVANTAGES:
  - It can not represent all the relationship of entire world
  - Maintaining the Database is very difficult task.
  - Little scope for "query optimization"
  - Wastage of storage space
  - Inconsistency during updation of database because when parent node is deleted that results in deletion of child node force fully.
  - Not flexible.

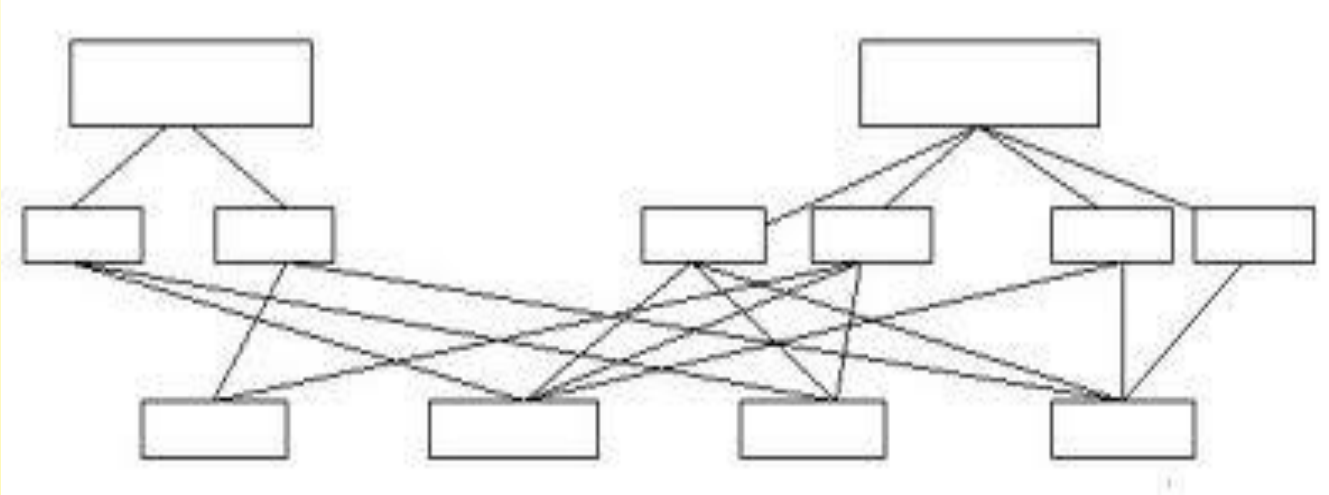
Contd..

Commercially available Hierarchical Database system:

- IBM's information management system
- MRI's system 2000
- IMS informatics Mark IV
- Time –shared Data management System of SDC

Network Model

- Network model is based on graph structure. A Network Db consists of collection of records, that are connected to each other by links.



ADVANTAGES

- **Provide very efficient "High-speed" retrieval**

- **Simplicity**
- The network model is conceptually simple and easy to design.

- **Ability to handle more relationship types**
- The network model can handle the one-to-many and many-to-many relationships.

- **Ease of data access**
- In the network database terminology, a relationship is a set. Each set comprises of two types of records.- an owner record and a member record, In a network model an application can access an owner record and all the member records within a set.

- **Data Independence**
- The network model draws a clear line of demarcation between programs and the complex physical storage details. The application programs work independently of the data. Any changes made in the data characteristics do not affect the application program.

- **Flexible than hierarchical model.**

DISADVANTAGES

- **Complex to design than relational model**
- **Maintenance is not easy.**
- **Flexibility and efficiency are less than relational model**
- **Lack of Structural independence.**
- Making structural modifications to the database is very difficult in the network database model as the data access method is navigational. Any changes made to the database structure require the application programs to be modified before they can access data. Though the network model achieves data independence, it still fails to achieve structural independence.

Relational Model

- The relational model used the basic concept of a relation or table. The columns or fields in the table identify the attributes such as name, age, and so. A tuple or row contains all the data of a single instance of the table such as a person named Doug. In the relational model, every tuple must have a unique identification or key based on the data. In this figure, a social security account number (SSAN) is the key that uniquely identifies each tuple in the relation.

Books

Book_ID	Book_Title	Author
Win007	Cloud Street	Tim Winton
Win008	Cloud Street	Tim Winton
Gri001	Grimm's Fairytales	Brothers Grimm

Borrowers

Borrower_ID	Lastname	Phone
Adam0098	Adamson	9567 9999
Albe0076	Alberti	9567 1234
Asto0034	Aston	9568 4567

Loans

Loan_ID	Book_ID	Borrower_ID
00000123	Win007	Albe0076
00000124	Gri001	Albe0076
00000125	Win008	Adam0098

ADVANTAGES:-

- Ease of use: The revision of any information as tables consisting Of rows and columns is quite natural and therefore even first time users find it attractive.

- Flexibility: Different tables from which information has to be linked and extracted can be easily manipulated by operators such as project and join to give information in the form in which it is desired.

- Precision: The usage of relational algebra and relational calculus in the manipulation of the relations between the tables ensures that there is no ambiguity.

- **Security**: Security control and authorization can also be implemented more easily by moving sensitive attributes in a given table into a separate relation with its own authorization controls.
- structural Independence
- Easy to Design

Disadvantage:-

- A major constraint and therefore disadvantage in the use of relational database system is machine performance. If the number of tables between which relationships to be established are large and the tables themselves are voluminous, the performance in responding to queries is definitely degraded.

- Need more powerful computing H/W and data storage devices that increase the cost and H/W overhead.

Schemas versus Instances

- **Database Schema**: The *description* of a database. Includes descriptions of the database structure and the constraints that should hold on the database.
- **Database Instance**: The actual data stored in a database at a *particular moment in time*. Also called **database state** (or **occurrence**).

- **Distinction**
  - The **database schema** changes *very infrequently*. The **database state** changes *every time the database is updated.*
  - **Schema** is also called **intension**, whereas **state** is called **extension**.
- **Data dictionary / repository**:
  - Used to store schema descriptions and other information such as design decisions, application program descriptions, user information, usage standards, etc.

Three-Schema Architecture

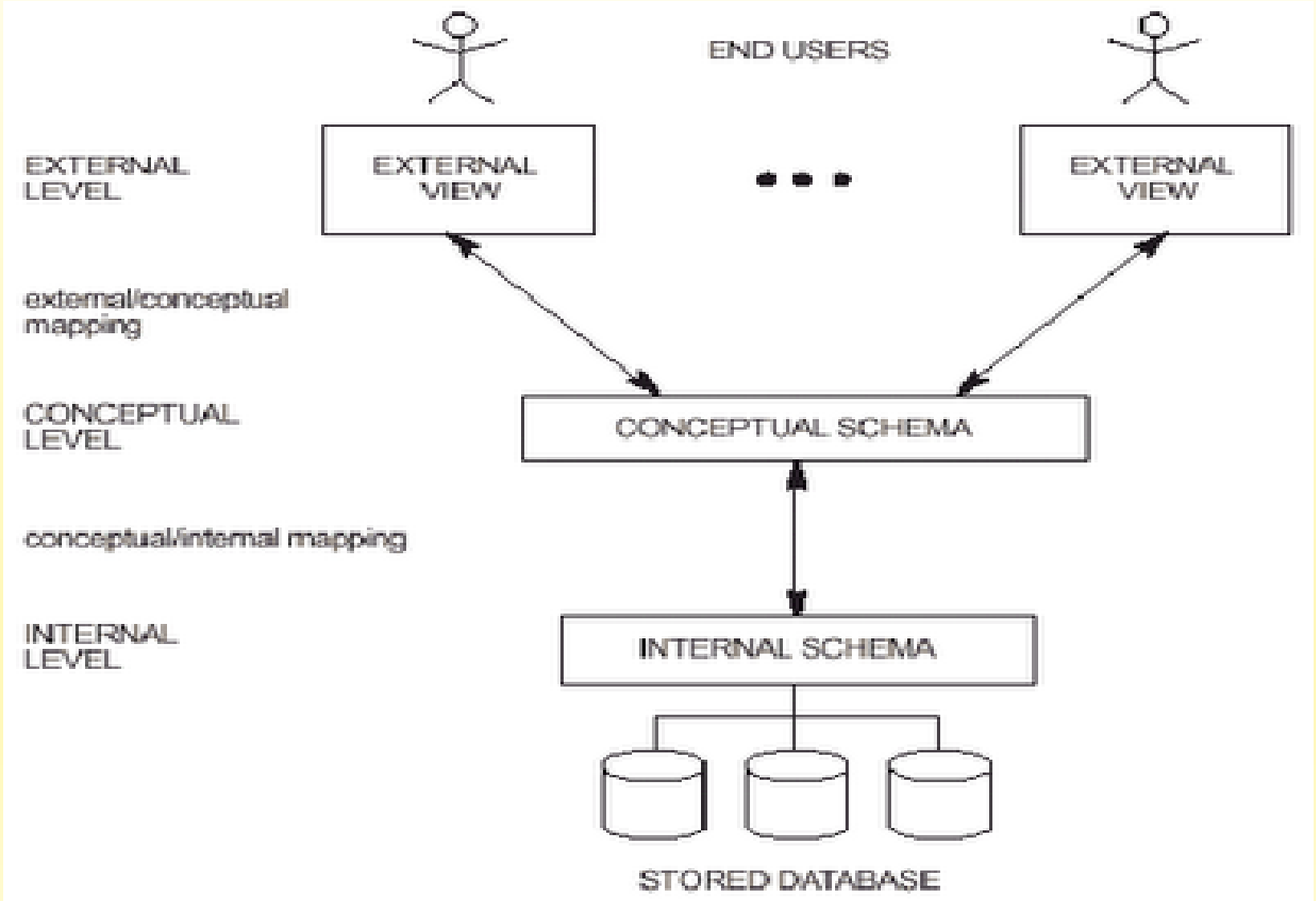
- Defines DBMS schemas at *three levels*:
  - **Internal schema** at the internal level to describe physical storage structures and access paths. Typically uses a *physical* data model. It describes how a record (e.g., customer) is stored
  - **Conceptual schema** describes data stored in database, and the relationships among the data.
  - **type** *customer* = **record**

    customer_id : string;
    customer_name : string;
    customer_street : string;
    customer_city : string;

    .
  - **External schemas** at the external level to describe the various user views.. application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

Three-Schema Architecture

- An architecture for a database system.



Three-Schema Architecture

Advantage :-

1. Logical data Independence .

2. Physical data Independence

This division into levels allows both developers and users to work on their own levels –They do not need to know the details of the other levels –and they do not have to know anything about changes in the other levels

.

Data Independence

- **It is defined as the characteristics of a Db system to change the schema at on level without having to change the schema at the next higher level.**

- **Logical Data Independence**: The capacity to change the conceptual schema without having to change the external schemas and their application programs.
- **Physical Data Independence**: The capacity to change the internal schema without having to change the conceptual schema

Data Independence

- **Mappings** among schema levels are needed to transform requests and data. Programs refer to an external schema, and are mapped by the DBMS to the internal schema for execution.

Data Independence

The conceptual/internal mapping:

- The conceptual-internal mapping defines the correspondence between the conceptual view and the internal view, i.e. database stored on the physical storage device. It describes how conceptual records are stored and retrieved to and from the storage device. This means that conceptual-internal mapping tells the DBMS that how the conceptual! records are physically represented. If the structure of the stored database is changed, then the mapping must be changed accordingly. It is the responsibility of DBA to manage such changes.

Data Independence

An external/conceptual mapping:

- An external-conceptual mapping defines the correspondence between a particular external view and the conceptual view. The external-conceptual mapping tells the DBMS which objects on the conceptual level correspond to the objects requested on a particular user's external view. If changes are made to either an external view or conceptual view, then mapping must be changed accordingly.

- A change to the storage structure definition means that the conceptual/internal mapping must be changed accordingly, so that the conceptual schema may remain invariant, achieving physical data independence.
- A change to the conceptual definition means that the conceptual/external mapping must be changed accordingly, so that the external schema may remain invariant, achieving logical data independence.

Data Independence

- When a schema at a lower level is changed, only the **mappings** between this schema and higher-level schemas need to be changed in a DBMS that fully supports data independence. The higher-level schemas themselves are *unchanged*. Hence, the application programs need not be changed since they refer to the external schemas

DBMS Languages

- **Data Definition Language** (**DDL**): it is used to 1) Create 2) Alter 3) Drop .IT is Used by the DBA and database designers to specify the *conceptual schema* of a database.

- In many DBMSs, the DDL is also used to define internal and external schemas (views). In some DBMSs, separate **storage definition language** (**SDL**) and **view definition language** (**VDL**) are used to define internal and external schemas. DDL statements -Data Definition Language
-

- **Data abstraction**
- - one fundamental characteristic of the database approach
- - hides details of data storage that are not needed by most
- database users and applications

DBMS Languages

- **DML:-**
- Language for accessing and manipulating the data organized by the appropriate data model. Queries like insert ,update delete are exmple.
  - DML also known as query language
- Two classes of languages
  - **Procedural (Record at a time)** – user specifies what data is required and how to get those data ex. relational Algebra, INSERT, UPDATE, DELETE, SELECT
  - **Declarative (nonprocedural/High Level/ set at a time)** – user specifies what data is required without specifying how to get those data.ex-Relational calculus
- SQL is the most widely used query language
- .

- **DCL statements**-Data Control Language It is used to create roles, permissions, and referential integrity as well it is used to control access to database by securing it. 1) GRANT 2) REVOKE

- **TCL-Transactional Control Language** It is used to manage different transactions occurring within a database. 1) COMMIT 2) ROLLBACK 3) Save point
-