DBMS

UNIT-1

CONTENTS

- *Database Systems*
- *DBMS*
- *Database System environment*
- *Traditional file systems for storing data*
- *Advantages of DBMS over traditional file systems*
- *Describing and Storing data in DBMS*
- *Three schema Architecture*
- *Data independence*
- *Structure of a DBMS*
- *People who work with DBMS*

Introduction

- Databases and Systems to manage them have become significant components of any present day business of any nature.

- These databases help businesses to perform their day-to-day activities in an efficient and effective manner.

- Banking
- Travel ticket reservation
- Library catalog search

- Here some program access the database.
- Advances in technology have given raise to new concepts-

- ❑ Multimedia databases
- ❑ GIS
- ❑ Web data
- ❑ Data warehousing and mining

- *Data:* Known fact that can be recorded and that has implicit meaning.
- Ex. *Name*, *Tel_no*, *city* etc.

- This data can be stored in a file on a computer.

- *Database:* Is a collection of related data.

❖ It is a collection of logically related data.

❖ A database is designed, built and populated with data for a specific purpose

DBMS

- *DBMS:* Is a collection of programs that enables users to create and maintain databases in a convenient and effective manner.
- DBMS is a software system that facilitates the following:
- 1.*Defining the database:* This includes defining the structures, data types, constraints, indexes etc.
- Database catalog/Data dictionary/ called as *Meta-data*
- 2.*Constructing the database:* This means storing data into the database structures and storing on some storage medium.
- 3.*Manipulating database for various applications:* This encompasses activities like – *querying* the database*, inserting* new records into the database, *updating* some data items, and *deleting* certain items from the database.
- What is DBMS?
- What is a Database System?

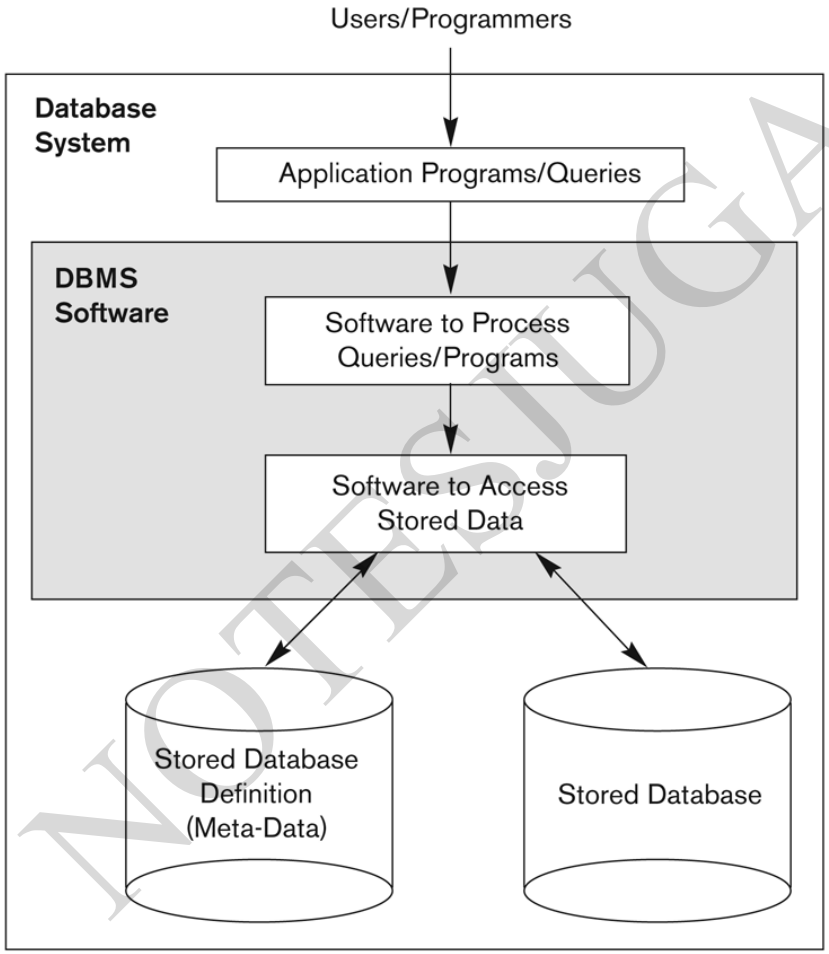


Figure 1.1
A simplified database system environment.

Traditional file systems for storing the data

- If we take the example of savings bank enterprise, information about customers and savings accounts etc. need to be stored.

- One way to keep the information on computers is to store in files provided by operating systems (OS).

- *Disadvantages of the above System*

❖ Difficulty in accessing data (possible operations need to be hard- coded in programs).
❖ Redundancy leading to inconsistency.
❖ Inconsistent changes made by concurrent users.
❖ No recovery on crash.
❖ The security provided by OS in the form of password is not sufficient.

Advantages of using DBMS

- Data independence
- Efficient data access
- Data integrity and security
- Data Administration
- Concurrent access and Crash recovery
- Reduced application development time

Disadvantages of DBMS

1. Extra cost due to SW, HW and training.
2. Not suitable or effective for certain applications (Real-time constraints; well-defined limited operations)
3. Data manipulation not supported by Query languages

Describing and storing data in DBMS

- ***Data model***

- Is a collection of high-level data description constructs that hide many
- low-level details.
- DBMS allows a user to define the data to be stored in terms of a data model.

- ***Semantic data models*****:** More abstract high-level data models that make it easier
- for a user to come up with a good initial description of the data in an enterprise.
- Contain wide variety of constructs that help describe a real-world enterprise data.
- Ex. ER model
- ***Representational / Implementation data models:***
- These are DBMS specific data models and are built around just few basic constructs.
- Ex. Relational data model, Object data model
-
- A database design in terms of a semantic model serves as a useful starting point and
- is subsequently translated into a database design in terms of the data model the DBMS supports.

- ***Relational Model***:

- The central data description construct in this model is a relation,
- which can be thought of as a set of records.

- Schema: Description of data in terms of a data model is called a schema.
- A relation schema specifies the name of the relation, field, type etc.

- Ex. *Student (sid: string; name: string; age: integer)*

- every row follows the schema of the relation.

- The following are some important representational data models (DBMS Specific)

- **1. *Network Model***: Though the basic structure is a record,
- the relationships are captured using links.
- The database can be seen as an arbitrary network of records connected by links.
- Ex.: GE's Integrated Data store (IDS), in Early 1960s
- ***2. Hierarchical Model***: The records containing data are organized
- as a collection of trees. Ex.: IBMs IMS (Information Management System),
- in late 1960s
- **3. *Relational Model***: (early 1970s)Data & relationships are captured as tables & keys.
- Ex.: Oracle, IBMs DB2, MySQL, Informix, Sybase, MS Access, Ingress, MySQL etc.
- The basic storage structure is a record.
- **4. *Object Data Model***: Objects created through object–oriented programs
- can be stored in database.
- Ex.: Object Store
- ***5. Object Relational Model***: Objects can be stores in tables.
- Ex.: Oracle, Informix

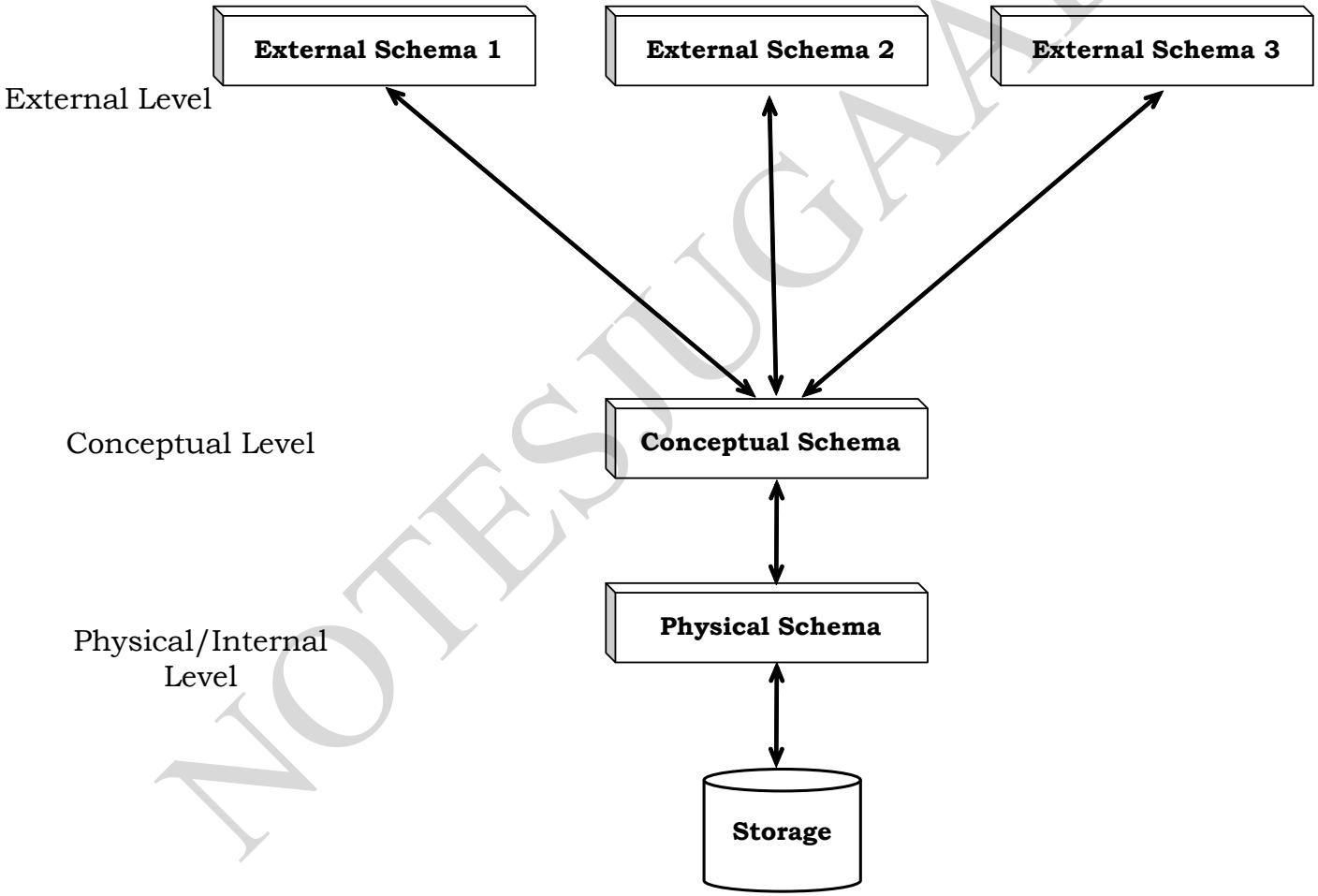
Database Schema

- *Database Schema*: Description of a database is called as *database Schema*

- *Three-Schema Architecture*
- A database can be described using three different levels of abstractions.
- Description at each level can be defined by a schema. For each abstraction we focus on one of the specific issues such as user views, concepts, storage etc.
-

1. *External schema*: Used to describe the database at external level.
- Also described in terms of the data model of that DBMS. This allows data access to be customized at the level of individual users/groups/applications.
- Any external schema has one or more views and relations from the conceptual schema. This schema design is guided by end user requirements.
2. *Conceptual schema* (logical schema) Describes the stored data in terms of the data model specific to that DBMS. In RDBMS conceptual schema describes all relations that are stored in the database. Arriving at good choice of relations, fields and constraints is known as conceptual database design.
- 3. *Physical schema*: Describes the physical storage strategy for the database.

Three Schema Architecture



Data Independence

- *Data Independence:*
- The three-level architecture which is the result of the three-level abstraction on database, leads to data independence.

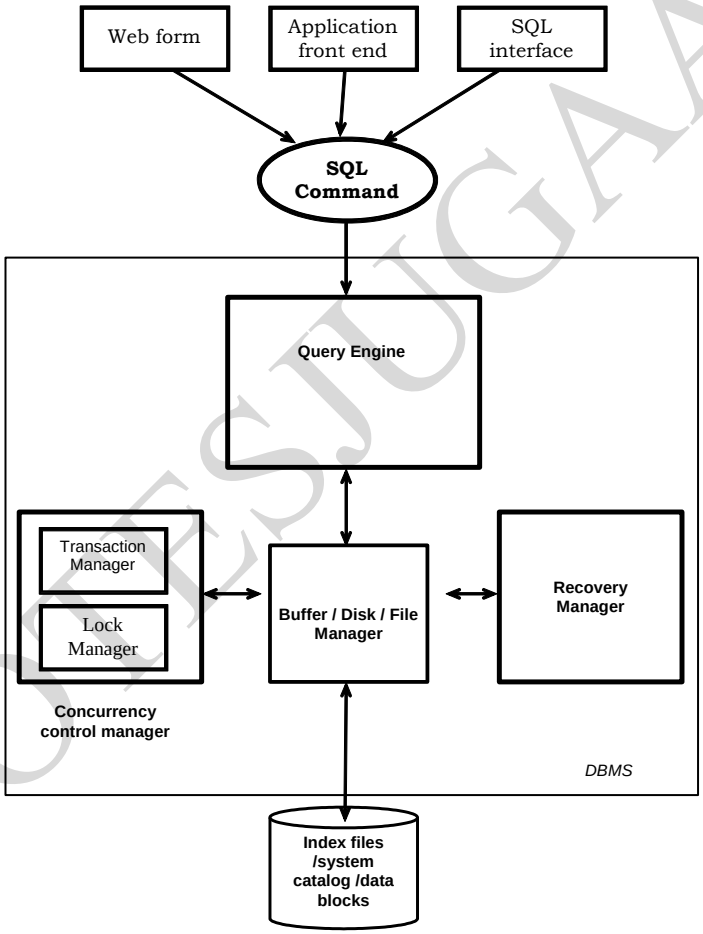
1. *Logical data independence:* changes in conceptual level schema should not affect the application level or external level schemas.

- 2. *Physical data independence:* The changes in physical features of storage, i.e., changes to the physical storage format should not affect schema at conceptual level.

- The above data independence is one of the important advantages of DBMS.

- The DBMS stores the description of schemas as System catalog

DBMS Structure



People who work with DBMS

- ***Database Implementers***
- ***End users***
- ***Application Programmers***
- ***Database administrator (DBA)***

- ***DBA's role:***
  1. ***Design of physical & Conceptual schemas***
  2. ***Security and authorization***
  3. ***Data availability , recovery and backup***
  4. ***Database tuning- modifying the schemas to meet the requirements***