1. Database Users

There are four different types of database-system users, differentiated by the way theyexpect to interact with the system. Different types of user interfaces have been designedfor the different types of users.

1. **Na¨ıve users** are unsophisticated users who interact with the system by using predefined user interfaces, such as web ormobile applications. The typical user interface for na¨ıve users is a forms interface, where the user can fill in appropriate fields of the form. Na¨ıve users may also view read *reports* generated from the database.

As an example, consider a student, who during class registration period, wishes
to register for a class by using a web interface. Such a user connects to a web
application program that runs at a web server. The application first verifies the
identity of the user and then allows her to access a form where she enters the
desired information. The form information is sent back to the web application
at the server, which then determines if there is room in the class (by retrieving
information from the database) and if so adds the student information to the class
roster in the database.

2. **Application programmers** are computer professionals who write application programs. Application programmers can choose from many tools to develop user interfaces.

3. **Sophisticated users** interact with the system without writing programs. Instead,they form their requests either using a database query language or by using tools such as data analysis software. Analysts who submit queries to explore data in the database fall in this category.

4. **The Data base Administrator (DBA)** who is like the super-user of the system.

Comment [S1]: This is not there in mam's notes. Only 3 users given.

2. Roles of DBA

- **Schema definition:** The DBA creates the original database schema by executing aset of data definition statements in the DDL.
- **Liaising with users**: The DBA needs to interact continuously with the users to understand the data in the system and its use.
- **Defining Security & Integrity checks:** The DBA finds about the access restrictions to be defined and defines security checks accordingly. Data Integrity checks are defined by the DBA.
- **Defining Backup/Recovery Procedures**: The DBA also defines procedures for backup and recovery. Defining backup procedure includes specifying what data is to be backed up, the periodicity of taking backups and also the medium and storage place to backup data.
- **Monitoring performance:** The DBA has to continuously monitor the performance of the queries and take the measures to optimize all the queries in the application.
- **Storage structure and access-method definition**. The DBA may specify some parameters pertaining to the physical organization of the data and the indices to be created.
- **Schema and physical-organization modification.** The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization, or to alter the physical organization to improve performance.
- **Granting of authorization for data access.** By granting different types of authorization, the database administrator can regulate which parts of the database various users can access. The authorization information is kept in a special system structure that the database system consults whenever a user tries to access the data in the system.
- **Routine maintenance**. Examples of the database administrator's routine maintenance activities are:

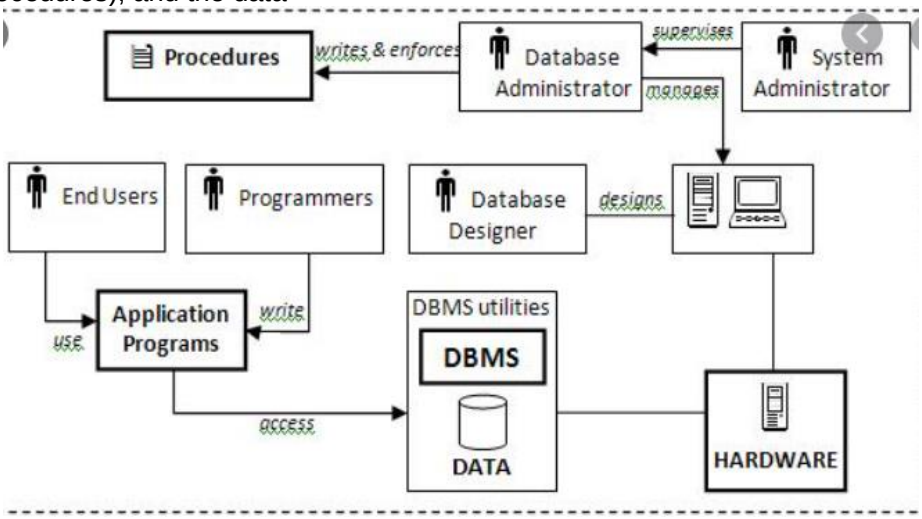
Comment [S2]: In book, not in mam notes

° Periodically backing up the database onto remote servers, to prevent loss of
data in case of disasters such as flooding.
° Ensuring that enough free disk space is available for normal operations, and
upgrading disk space as required.
° Monitoring jobs running on the database and ensuring that performance is not
degraded by very expensive tasks submitted by some users.

3. Database System Environment

A database environment is a collective system of components that comprise and regulates the group of data, management, and use of data

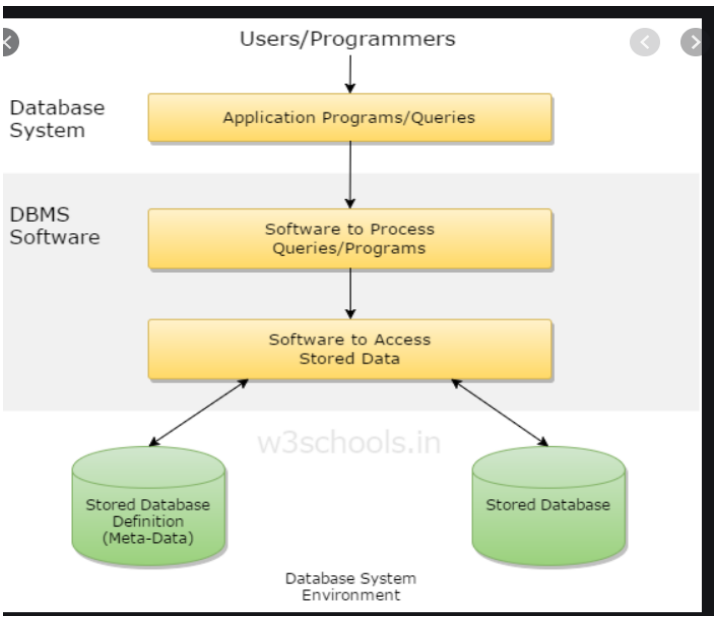
Consists of: software, hardware, people, techniques of handling database (procedures), and the data



1. **Hardware:** Hardware refers to all of the system's physical devices; for example, computers storage devices, printers, network devices and etc
2. **Software:** To make the database system work properly, three types of software are needed: operating system, DBMS software, and application programs.
   a) Operating system: It manages all hardware components and allows other software to run on the computers. Examples of operating system software include Windows, Linux and etc.
   b) DBMS software: It manages the database within the database system. Some examples of DBMS software include Oracle, Access, MySql and etc.
   c) Application programs: These are used to access and manipulate data in the DBMS and to manage the computer environment in which data access and manipulation take place. Application programs are most commonly used to access data to generate reports. Most of the application programs provide GUI.
3. **People:** This component includes all users of the database system. According to the job nature, five types of users can be identified: systems administrators, database administrators, database designers, systems analysts and programmers, and end users.

a) System administrators: They supervise the database system's general operations.
b) Database administrators: They are also known as DBAs. They manage the DBMS and ensure that the database is functioning properly.
c) Database designers: They design the database structure. They are the database architects. As this is very critical, the designer's job responsibilities are increased.
d) Systems analysts and programmers: They design and implement the application programs. They design and create the data entry screens, reports, and procedures through which end users can access and manipulate the data.
e) End users: They are the people who use the application programs to run the organization's daily operations. For example, sales-clerks, supervisors, managers are classified as end users.

4. **Procedures:** Procedures are the instructions and rules that supervise the design and use of the database system. Procedures are a critical component of the system. Procedures play an important role in a company because they enforce the standards by which business is conducted in an organization
5. **Data:** Data refers the collection of facts stored in the database. Because data are the raw material from which information is generated, no database can exist without database.



- A database management system (DBMS) is a collection of programs that enables users to create and maintain database.
- The DBMS is a general purpose software system that facilitates the process of defining, constructing, manipulating and sharing databases among various users and applications.
- Defining a database specifying the database involves specifying the data types, constraints and structures of the data to be stored in the database.
- The descriptive information is also stored in the database in the form database catalog or dictionary; it is called meta-data.
- Manipulating the data includes the querrying the database to retrieve the specific data. An application program accesses the database by sending the queries or requests for data to DBMS.
- The important function provided by the DBMS includes protecting the database and maintain the database.

4. Why a Spreadsheet Is Not a Database

While a spreadsheet allows for the creation of multiple tables, it does not support even the most basic database functionality such as support for self-documentation through metadata, enforcement of data types or domains to ensure consistency of data within a column, defined relationships among tables, or constraints to ensure consistency of data across related tables. Most users lack the necessary training to recognize the limitations of spreadsheets for these types of tasks

5. 3-Tier Architecture of DBMS

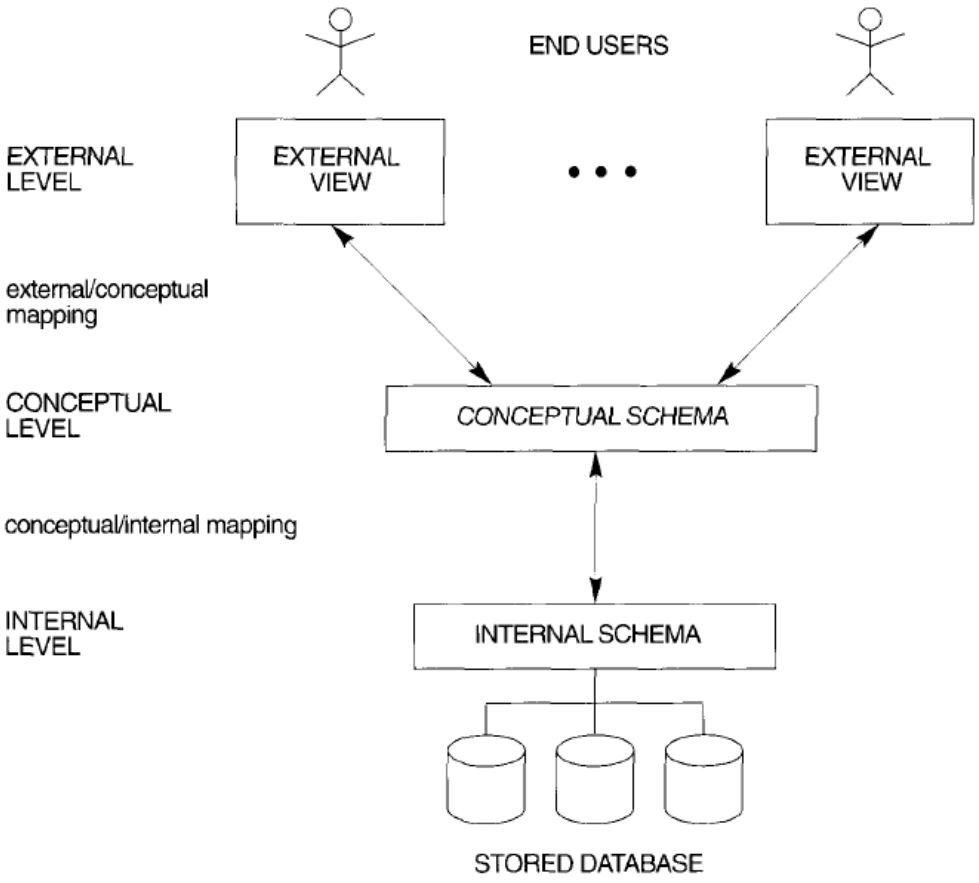
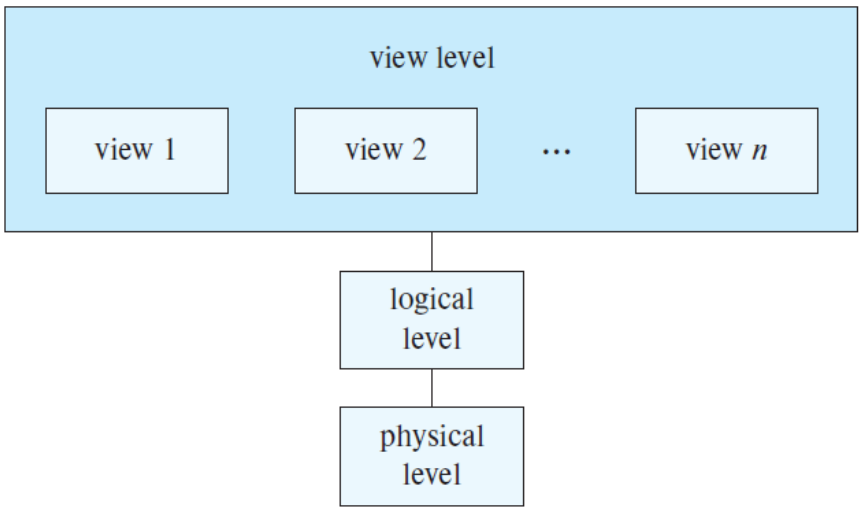


FIGURE 2.2 The three-schema architecture.



A commonly used views of data approach is the three-level architecture suggested by ANSI/SPARC (American National Standards Institute/Standards Planning and Requirements Committee).

The three levels of the architecture are three different views of the data:

External - individual user view:

Comment [S3]: •Comprises of a the things visible to usual

- It represents the user's view of the database.
- This view is often a restricted view of the database and the same database may provide a number of different views for different classes of users.
- It is closely related to the real world as perceived by each user.
- It is quite abstract in nature and closer to the way a programming language would model a record
- The view level exists to simplify the interaction of users with the system.
- It allows independent customized user views.
- It hides the physical storage details from users

- The external or view level includes a number of external schemas or user views. Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group
- s/w and h/w independent
- Software independence means that the model does not depend on the DBMS software used to implement the model, i.e, the view of data for a given user will be same, regardless of the software he uses.
- Hardware independence means that the model does not depend on the hardware used in the implementation of the model, that is it is unaffected by the choice of the computer on which the software is installed. Therefore, a change in storage devices or even a change in operating systems will not affect the internal model

Conceptual - community user view – logical level

Comment [S4]: Comprises of ent and their types, and relation

- Present a “community view”: the logical structure of the entire database
- The conceptual level is a way of describing what data is stored within the whole database and how the data is inter-related. It does not specify how the data is physically stored.
- This view is normally more stable than the other two views. [Change in physical level does not affect conceptual level]
- The conceptual view is defined by the conceptual schema which includes definitions of each of the various types of data.
- The conceptual model represents a global view of the entire database as viewed by the entire organization. That is, the conceptual model integrates all external views (entities, relationships, constraints, and processes) into a single global view of the data in the enterprise. Also known as a conceptual schema,
  The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints
- Show relationships among data including:

Constraints
Semantic information (e.g., business rules)
Security and integrity information

- Some facts: DBA works at this level. Describes the structure of all users. Only DBA can define this level. Global view of database.
- Because the internal model depends on specific database software, it is said to be software-dependent. Therefore, a change in the DBMS software requires that the internal model be changed to fit the characteristics and requirements of the implementation database model
- Hardware independence means that the model does not depend on the hardware used in the implementation of the model, that is it is unaffected by the choice of the computer on which the software is installed. Therefore, a change in storage devices or even a change in operating systems will not affect the internal model

Internal - physical or storage view

Comment [S5]: Comprises of wh how data is stored, encryption

- The internal view is the view about the actual physical storage of data. It tells us what data is stored in the database and how.
- At least the following aspects are considered at this level:
  1. Storage allocation e.g. B-trees, hashing etc.
  2. Access paths e.g. specification of primary and secondary keys, indexes and pointers and sequencing.
  3. Miscellaneous e.g. data compression and encryption techniques, optimization of the internal structures.
- provides information regarding the physical organisation of the data in the database
- The internal model is the representation of the database as “seen” by the database
- Provide concepts that describe the details of how data are stored in the computer’s memory
- deals with
  - Run-time performance
  - Storage utilization and compression

    - File organization and access methods
    - Data encryption
- The internal level has an internal schema, which describes the physical storage structure of the database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.
- the physical model is dependent on the DBMS, methods of accessing files, and types of hardware storage devices supported by the operating system. Therefore h/w s/w dependent.
- managed by the operating system (OS)

Example:

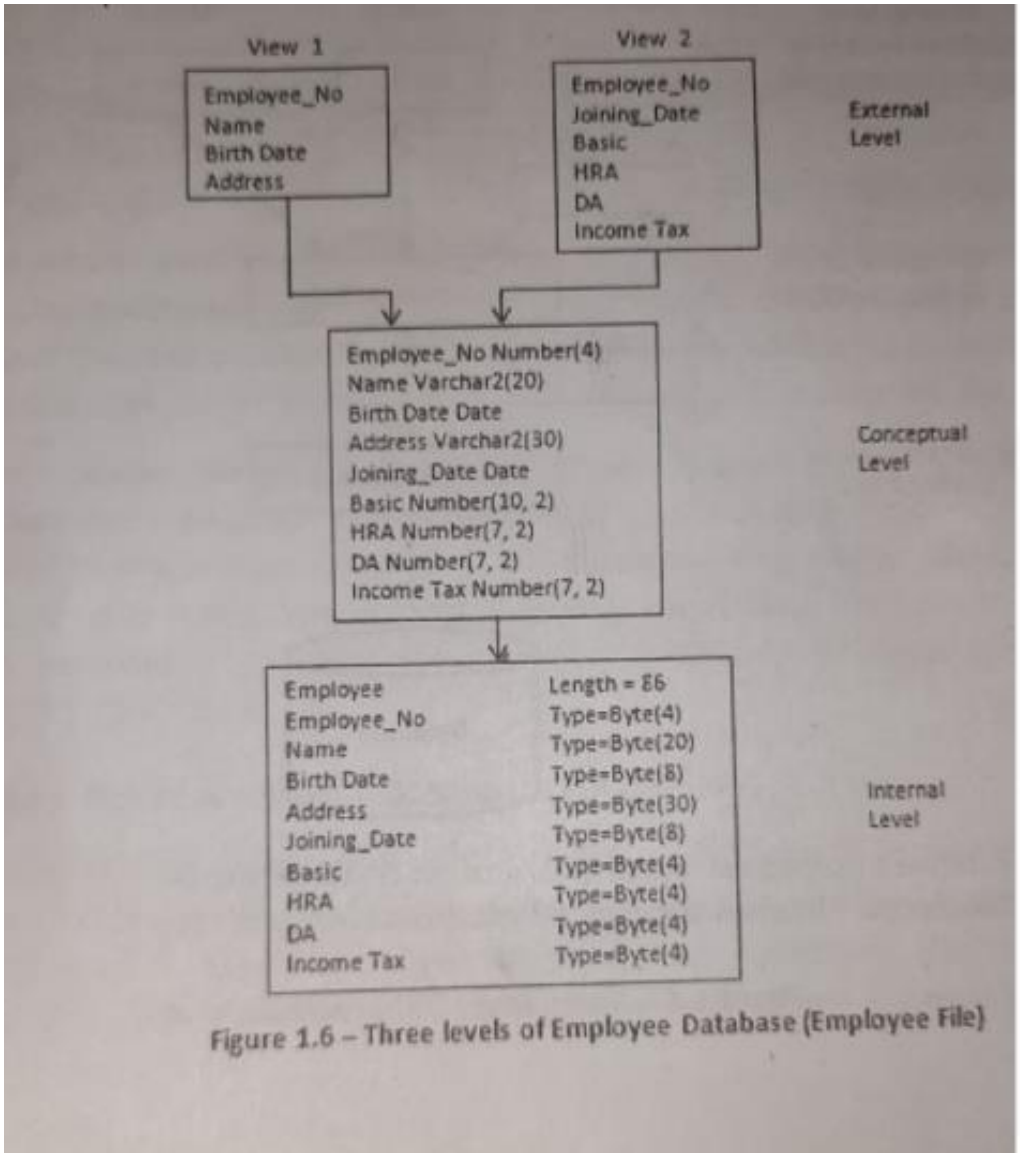


Figure 1.6 – Three levels of Employee Database (Employee File)

The three-schema architecture is a convenient tool with which the user can visualize the schema levels in a database system. Most DBMSs do not separate the three levels completely, but support the three-schema architecture to *some* extent. Some DBMSs may include physical-level details in the conceptual schema.

Mappings : the three schernas are only *descriptions* of data; the only data that *actually*
exists is at the physical level. In a DBMS based on the three-schema architecture, each
user group refers only to its own external schema. Hence, the DBMS must transform a
request specified on an external schema into a request against the conceptual schema, and
then into a request on the internal schema for processing over the stored database. If the
request is a database retrieval, the data extracted from the stored database must be
reformatted to match the user's external view. The processes of transforming requests and
results between levels are called **mappings**

6. Data Independence

Data independence can be defined as the capacity to change the schema at one level without changing the schema at next higher level.

Data Independence occurs because when the schema is changed at some level, the schema at the next higher level remains unchanged; only the mapping between the two levels is changed.

They are of two types:

1. **Logical data independence:**

- is the capacity to change the conceptual schema without having to change the external schema, In other words, changes to the logical schema (e.g., alterations to the structure of the database like adding a column or other tables) should not affect the function of the application (external views).
- Changes in entities, attributes, or relationships should not affect views not dealing directly with them
- We may change the conceptual schema to expand the database (by adding a record type or data item), to change constraints, or to reduce the database (by removing a record type or data item). In the last case, external schemas that refer only to the remaining data should not be affected.

- Only the view definition and the mappings need be changed in a DBMS that supports logical data independence. After the conceptual schema undergoes a logical reorganization, application programs that reference the external schema constructs must work as before.

2. Physical data independence:

- is the capacity to change the internal schema without changing the conceptual schema, Hence, the external schemas need not be changed as well.
- Changes to the internal schema may be needed because some physical files had to be reorganized-for example, by creating additional access structures-to improve the performance of retrieval or update. If the same data as before remains in the database, we should not have to change the conceptual schema.
- In other words, it is Immunity of the conceptual schema to changes in the internal schema.
- Changes in storage structures, hardware, indexes, hashing, defragmentation, etc. - Only effect might be a change in performance.