* Data → Data means raw facts and figures that can be recorded and have implicit meaning.

* Information → processed data is called inf^n.

* Database (DB) → Collection of related data together is called DB.

* DBMS → A S/W package / System that facilitates the users to create, use and manipulate the database which include defining, constructing, manipulating and sharing.

or

A S/W package / System that can be used to store, manage and retreive data from databases.

* DB system (DBS) :→ Collection of DBMS S/W along with Database is DBS

* Difference b/w file System and DBMS.

1). Files are used to store data in form of records.	DBMS is a collection of related data.
2). Tasks such as storange and retreiving the data is done manually.	In DBMS, automated methods are used.
3). problems like data integrity, data searching, data redundancy is inefficient.	In DBMS, it is very efficient.
4) file System has one user at a time.	multiuser at a time.

5)	Backup is not there.	Backup is there.
6).	Less Security	Security is provided.

Characteristics of DBS:

1). Self describing nature of DB.
2) Insulation b/w program and data, and data abstraction.
3). Support of multiple views of the data.
4) Sharing of data and multiuser processing

1). Self describing nature of DB

:→ The DB system contains the complete definition or description of DB structure and constraints. This definition is stored in the DBMS catalog, which contains information such as the structure of each file, the type and storage format of each data item, and various constraints on the data.

- The inf^n stored in catalog is called metadata, and it describes the structure of the primary database.

2). Insulation b/w prog^s and data, and Data abstraction

:→ The structure of data files is stored in the DBMS catalog seprately from the access programs. Hence, changes to the data or structure of the file does not require changes in program, this property is called data independance.

The characteristic that allows

program data independance and program-operation independence is called data abstraction. Hiding the details that how the data is stored and implemented from users, is called data abstraction

3). Support of multiple views of the data

:→ A database typically has many users, each of whom may require a different view of the database. A view may be a subset of the database or it may contain virtual data that is derived from the database files but is not explicitely stored. ~~or derived~~. The multiuser DBMS provide fascilities for defining multiple views.

4). Sharing of data and multiuser transaction processing:

→ A multiuser DBMS, as its name implies, must allow multiple users to access the database at the same time. This is essential if data for multiple appl^n is to be integrated and maintained in a single Database. The DBMS must include Concurrency Control S/W to ensure that several users trying to update the same data do so in a controlled manner so that the results of the updates is correct.

Actors on the scene :→ The people whose jobs involve the day to day use of a large database ; are called actors on the scene.

1). DB Administrator :→ The DBA is responsible for authorizing the access to the DB, coordinating and monitoring its use, and acquiring S/W and H/W resources as needed

2). DB Designers :→ DB Designers are responsible for identifying the data to be stored in the DB and choosing appropriate structures to represent and store this data. The final Database Design designed by the DB Designer must be capable of supporting the requirements of all user groups.

3). End Users → End users are the people whose jobs require access to the database for querying, updating and generating reports; The DB primarily exists for their use.

The several categories of end users are -

(i) Casual End users → They occasionally access the DB, but they may need different inf^{n} each time

(ii) parametric End users :→ They use the DB mostly and keep it updating.

(iii) Sophisticated End users :→ Sophisticated end users include Doctor, Engineer, Scientists, Business Analysts and Others who thoroughly familiarize themselves with the facilities of the

DBMS in order to implement their own appl^ns to meet their complex requirements.

(iv) Standalone users :→ They maintain their personal databases by using ready-made programs packages that provide easy to use menu based or graphics based interface.
Ex - Tax package that stores variety of personal financial data for tax purposes.

4). System Analysts :→ They determine the requirement of end users and develop specifications for standard canned transaction that meet these requirements.

5) Appl^n programmer :→ Appl^n programmers implement the specifications analysed by system analyst as programs. They test, debug, document and maintain these canned transactions.

6. Actors behind the scene :→ In addition to those who design, use, and administer a database, others are associated with the design, development and oper^n of the DBMS S/W and system environment. These persons are typically not interested in the DB contents itself, are called Actors behind the scene.

(1) DBMS System Designer and Implementer
(2) Tool Developer
operators and ~~perso~~ maintainance personnel.

(1) DBMS System Designer and Implementers:

:→ They Design and implement the DBMS modules and interfaces as a S/W package. The DBMS must interface with other System S/W such as the OS and Compilers for various programming languages.

(2) Tool Developer → They Design and implement tools — the S/W package that facilitate database modeling and design, database System design, and improved performance. Tools are optional package that are often purchased separately. They include package for DB design, performance monitoring, natural language or graphical interfaces, prototyping, Simulation and test data generation.

(3) Operators and maintanance personnel:

→ They are also called System administration personnel and are responsible for the actual running and maintanance of the H/W and S/W environment for the DB System.

☆ Advantages of DBMS approach:

1). Controlling redundancy :— Storing same data on various places in a Database is called data redundancy/duplicacy. This redundancy leads to several problems such as duplication of efforts, wastage of storage space

Since the data is stored on various places in files and if update in applied to a single places the remaining places data remain as it is. Hence, the best advantage of DBMS approach is controlling redundancy.

2) Restricting unauthorized access → When multiple users share a large DB, it is likely that most users will not be authorized to access all the inf" in the DB. Some users may only be permitted to retrieve data, whereas others are allowed to retreive and update. Hence the type of access operations - retreival or update must also be controlled.

3) providing persistent storage for program objects:

→ DB can be used to provide persistent storage for program objects and data structure. This is one of the main reasons for object-oriented DBS. A Complex object in C++ can be stored permanently in an object-oriented DBMS. Such an object is said to be persistent, since it services the termination of prg execution and can later be directly retreived by C++ progs. The pessistent storage of prg objects and DS is an important fun" of DBSs.

4) providing Storage structure and Search techniques for efficient query processing.

→ The DBS must provide capability for efficient executing and queries and updates. Because, the DB is typically stored on disk,

the DBMS must provide Specialized data Structures and techniques to Speed up disk search for the desired records.

5). providing Backup and Recovery :→ The DBMS must provide the facilities for recovering from H/w or S/w failure. The backup and recovery subsystem of DBMS is responsible for recovery.

6). providing multiple user interfaces :→ The DBMS must be able to provide multiple views a/c to the requirements of the users.

7). Representing Complex relationship among data.

:→ A DBMS must have the capability to represent a variety of complex relationships among data, to define new relationships as they arise, and to retreive and update related data easily and efficiently.

8). Enforcing integrity Constraints :— Most DBMS appl^ns have certain integrity Constraints. A DBMS must provide capabilities for defining and enforcing these Constraints. The simplest type of integrity constraints involves specifying the data type for each data item.

). Permitting inferencing and Actions using rules

(vi). Additional implication of using the database approach : →

Data Models:

→ The Data model is nothing but a logical structure or conceptual structure. The models provide the necessary means to acheive the abstraction.

Categories of data model :—

1). High-Level or Conceptual Data model :— End user level data explanation is explained in this model. Only the basic things are described, not the details.

→ It provides concepts close to the way users preceive the data.

2). Implementational or Representational data model :

→ It is also called middle level data model. It is partial ~~level~~ explanation of high level and low level.

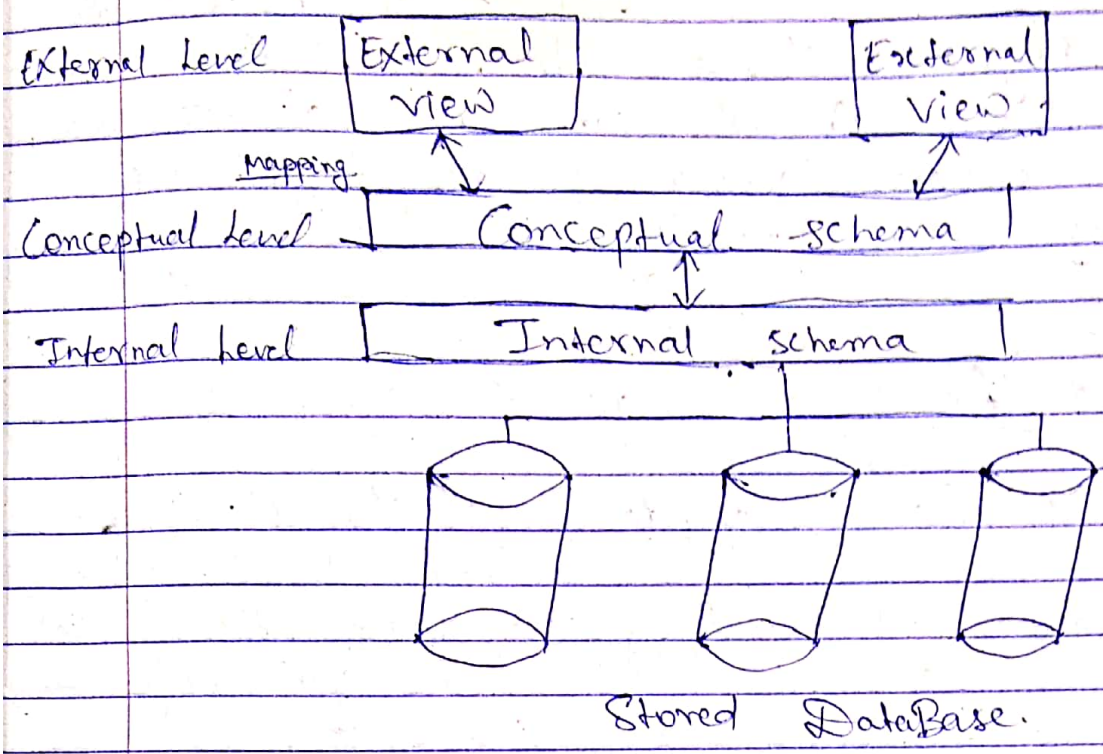
3). physical or low level data model :→

It describes how the data is stored in files by representing record formats, record ordering, and access paths.

Schema

Schema → It is the description of the DB It includes description of DB structure data types and constraints.

Three Schema Architecture:



Mapping → It converts requests into data and connect different levels of schema.

External Level → It includes external schemas or user views. Each external schema describes the part of the DB that a particular users group is interested in and hides the rest of the DB from that user group.

Conceptual Level :→ The Conceptual level schema has a Conceptual schema which describes the ~~part~~ structure of the DB for users. It hides the details of the physical Storage structure and Concentrates on describing entities, data types, relationships, user operations and constrains. Usually a representational data model is used to describe the Conceptual schema.

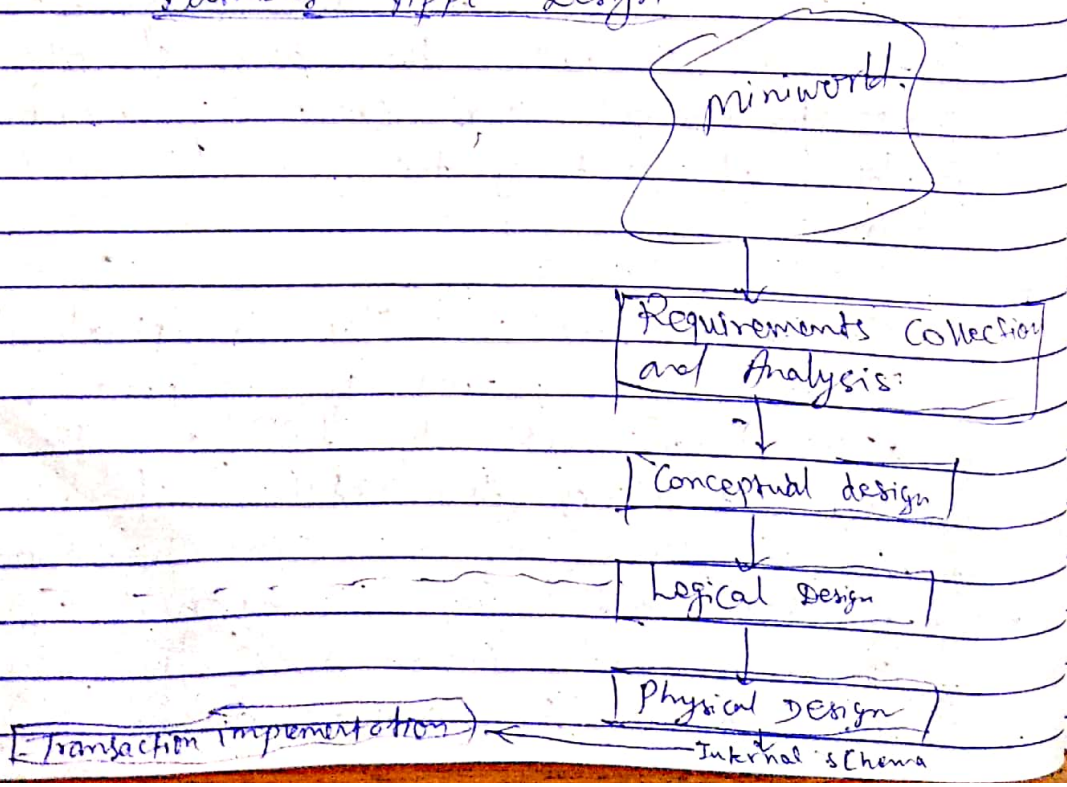
The internal Level :→ It has an internal schema which describes the physical storage structure of the DB. It uses a physical data model and describes the complete details of data storage and access paths for the DB.

Data ~~Inden~~ Independence :→

1) Logical Data Independence → The capacity to change the conceptual schema without having to change the external schema and their associated appl^n programs is called logical data independence.

2) Physical Data Independance :→ The capacity to change the internal schema without changing the Conceptual schema.

Database Appl^n Design



Entity – Relationship model

Entity – Entity is a real time object from miniworld.

Attributes – properties of ~~en~~ entities are called attributes.

Instance → Complete row (Single) is called an instance in a particular entity.

Attributes.

Simple Attributes. → The attributes that are not divisible, are called Simple or atomic ~~model~~ attributes.

Composite Attributes :→ Composite attributes are ~~called~~ can be divided into smaller subparts which represents more basic attributes ~~which~~ with independent meaning.

Single-valued Attributes :→ ~~most~~ Attributes having a single value for a particular entity, such ~~entities~~ ~~are~~ attributes are called Single-valued attributes.

Multi-Valued attributes :→ Attributes having multiple values are called multi-valued attributes.

Stored and Derived attribute :→ An attribute that doesn't exist in real but are derived from ~~on~~ a stored attribute are called derived attributes.