4. Databases:

Many small (personal computer) applications corporate information systems take a file-based approach to data storage. As an example of this approach if we take the customers information , the customers' names , addresses, products, and services , each of these information stored in a separate file and there is no relation between them , so another approach appear it's called data base approach, which all information stored under one data base.

A specialized system, the ***database management system (DBMS),*** will control all access to the database and be responsible for security, integrity, data formats, backups and other technical issues.

Data Structures :

1. **Database:** The totality of the data collected by an information system is called the *database*.

2. **File and Table:** Normally, the database will be structured into smaller logical or physical units: tables or files. The *table* is really a *logical* concept while the *file* is a *physical* way in which data is stored on the computer.

3. **Record:** Within a table (or file), information will be organized in the form of *records*. In the customer database, there may be a record containing all the data about customer (including his address, contact numbers, etc.), there will be another record for each of the customers.

4. **Field:** The information within a record is structured into separate *fields*. In practice, the customer name may be split into at least three separate fields: one for his surname, one for his first name, one for his designation "Mr. ". Similarly, the address will usually be split into several fields, the town and the street.

5. **Characters and Bytes:** Finally, the data inside a field will be encoded in a number of characters (for textual data) or bytes (for numerical and multimedia data).

Database Models:

Database model means the type and nature of the relationships that are allowed between different data tables.

1. **The Hierarchical Data Model:** The *hierarchical data (base) model* is the oldest and simplest model. This structure allows only for one type of relationship, the "parent-child" (or one-to-many) relationship. In addition, there can be only one single relationship between tables. This structure can also be or represented using the image of a tree. a customer (the "parent" entity or *root*) can place one or several orders (="child"). Each order will consist of one or several line items (the different products or Services that are ordered).

2. **The Network Model:** The *network data model* allows "many-to-many" relationships often found in the real world. In a university environment, each student can record for one or several courses, while each course will have many students. However, although the network model allows more complex relationships to be defined than the hierarchical model, it is significantly more complicated than the other models to design.

3. **The Relational Model:** all major DBMS in use today support the relational model almost in its entirety. The relational model requires that all data is stored in *data tables*, with unique key fields (the *primary* key) and all other data fields in the table entirely and fully dependent on the key fields. A typical example to illustrate the use of a relational database is that of student registrations at the university. There will be a *student* table containing the student number (the key field), surname, name, year of first registration. Similarly, there will be a *course* table which contains course codes (key field), course names, level of course (1st year, 2nd year etc.), type of course (first semester, second semester, full-year, summer term), etc.

4. **The Object-Oriented Model:** more and more applications are benefiting from the use of *object-oriented database management systems (OODMBS)*. Two particular strengths of the OO model are the capability to handle *multimedia* more easily and the feature of *inheritance*. Multimedia, such as pictures or sound, are more easily handled by OODBMS because it is much easier to store the methods, or procedures, to access and edit the data within the object than it is to include those methods permanently as part of the DBMS. As such, OODBMS are popular for engineering and scientific applications. The other advantage is that the OO model allows specialized objects to inherit object attributes and methods from their *parent*.

5. **Free-form databases:** Another attempt to deal with unstructured data and information is the *free-form database*. These databases feature powerful indexing and searching tools. In a way, one could consider the entire web to be one huge distributed freeform database.

Database Management:

The person who is responsible for looking after the computerized corporate database(s) called a ***Database Administrator (DBA)***, the main responsibility of this person is to define the standard data elements and structure of the database (i.e. table and field structure). He will also look after the integrity and safety of the *contents* of the database.

An important component of the DBMS is the ***data dictionary.*** This describes in detail all data elements in the database, i.e. all data tables, field names, field types (text, numeric values, monetary units, Boolean, dates), field lengths, key fields, relationships and dependencies between fields and tables, etc.

Database Processing:

1. **Centralized processing**

   In a *centralized processing* environment, the entire processing load is performed by the central computer, usually a mainframe, to which a number of terminals are connected. This central and powerful computer is therefore usually referred to as the *host* computer. The host controls the data storage and the communication. The terminal itself does almost no data processing. Although this model was very popular in the early days of mainframe computing

2. **Distributed processing**

   In the *distributed computing* environment, there is no single host computer that controls all of the organization's data. All data resides on individual workstations or, more often, on the file servers of the LANs of the functional departments or local branches that actually need the data. These departments' computers are then responsible for *all* the data processing. This is the exact opposite of centralized computing. Distributed processing is made possible by the development of powerful microcomputers with high-capacity hard disks.

3. **The client-server model**

   The *client-server* model is a compromise between the two extremes of centralized and distributed forms of processing. Generally, the central computer is responsible for runs the database management system and performs all related functions such as backup, data integrity, access control and security. Therefore it's called the *server*. The local workstation, the *client*, receives the data from the server and is responsible for the presentation of the data. An ideal example of the client-server model is the World-wide Web on the Internet.