**Types of DBMS**

**Hierarchical DBMS**

The hierarchical data model organizes data in a tree structure. There is a hierarchy of parent and child data segments. This structure implies that a record can have repeating information, generally in the child data segments. Hierarchical data model was developed by IBM in 1968 and introduced in I.M.S. (Information Management System).This model is like a structure of a tree with the records forming the nodes and fields forming the branches of the tree. In the hierarchical model, records are linked in the form of an organization chart. A tree structure may establish one-to-many relationship. To create links between these record types, the hierarchical model uses Parent Child Relationships. For example, an organization might store information about an employee, such as name, employee number, department, salary. The organization might also store information about an employee's children, such as name and date of birth. The employee and children data forms a hierarchy, where the employee data represents the parent segment and the children data represents the child segment. If an employee has three children, then there would be three child segments associated with one employee segment. In a hierarchical database the parent-child relationship is one to many. This restricts a child segment to having only one parent segment.

**Network DBMS**

Network DBMS are not in use these days due to its complexity. A DBMS is said to be a Network DBMS if the relationships among data in the database are of type many-to-many. The relationships among many to many appear in the form of a network. Thus the structure of a network database is extremely complicated because of these many-to-many relationships in which one record can be used as a key of the entire database. A network database is structured in the form of a graph that is also a data structure. Though the structure of such a DBMS is highly complicated however it has two basic elements i.e. records and sets to designate many-to-many relationships

**Relational DBMS**

(RDBMS - relational database management system) A database based on the relational model developed by E.F. Codd. A relational database allows the definition of data structures, storage and retrieval operations and integrity constraints. In such a database the data and relations between them are organized in tables. A table is a collection of records and each record in a table contains the same fields.

**Properties of Relational Tables:**

Values Are Atomic

Each Row is Unique

Column Values Are of the Same Kind

The Sequence of Columns is Insignificant

The Sequence of Rows is Insignificant

Each Column Has a Unique Name

Certain fields are designated as keys

**Object-Oriented Model**

The object-oriented database (OODB) paradigm is the combination of object-oriented programming language (OOPL) systems and persistent systems. The power of the OODB comes from the seamless treatment of both persistent (permanent) data, as found in databases, and transient (temporary) data, as found in executing programs. There is one-to-one mapping of object programming language objects to database objects. It has two benefits over other storage approaches: it provides higher performance management of objects, and it enables better management of the complex interrelationships between objects. This makes object DBMSs better suited to support applications such as financial risk analysis systems, telecommunications service applications, World Wide Web document structures, design and manufacturing systems, and hospital patient record systems, which have complex relationships between data.

**Semi-structured DBMS**

Semi-structured data model permits the specification of data where individual data items of the same type may have different set of attributes. In Semi-structured data model, information that is normally associated with a schema is contained within the data, which is sometimes called ``self-describing''. XML language provides a way to store and retrieve information in nested structure.

<Employee Name> Ramesh </Employee Name>

<Post> Officer </Post>

<Salary> 20,000 </Salary>