1. **List and explain types of databases.**

**Hierarchical databases**

This type of database organises data in a tree-like structure. It stores data in the form of parent-children relationship nodes. A single object “parent” has one or multiple objects beneath it “the child”. No child can have more than one parent. It has easy access and quick querying time.

**Relational databases**

They store data in the form of rows and columns, which form a table (relation). A relation database uses SQL for storing, manipulating, as well as maintaining the data. Each table in the database carries a key that makes the data unique from others. Each column within the table has a name and a datatype. Each row represents and individual record or data item within the table, which contains values for each other column.

Relational databases are a choice for many applications because the applications generate well-ordered structured data.

**Centralized databases**

In centralized databases, data is stored at a centralized location and the users from different locations can access this data. The centralized database contains application procedures that help the users to access the data even from remote locations. The applications contain an authentication process to let users access data securely.

**Distributed databases**

A distributed database refers to a system in which the data is stored across multiple physical or logical locations. It involves dividing the database into different parts, called fragments or partitions, and distributing those fragments across multiple nodes or servers in a network. In a distributed database, each node or server holds a portion of the data, and they work together to provide a unified view of the database to users and applications.

**Cloud databases**

They refer to databases that are hosted and operated in a cloud computing environment. Instead of being stored on-premises or on dedicated servers, the database and its associated components are deployed and managed on cloud infrastructure provided by a cloud service provider. Cloud databases offer flexibility, scalability, high availability, and reduced operational overhead compared to traditional on-premises databases.

**Network databases**

They are based on a hierarchical structure that represents relationships between records as a network of interconnected nodes. Network database entries can have more than one parent which allows them to model more complex relationships. Network databases can be represented by a generalized graph structure which outlays what each node and each relationship represents.

**Object-oriented databases**

Object-oriented databases are designed to store and manage objects as the primary data structure. Unlike traditional relational databases that store data in tables, an OODB stores data as objects. They seek to bridge the gap between the representations used by object-oriented programming languages and databases.

**NoSQL databases**

NoSQL can stand for either “non-SQL” or “not only SQL”, they are also known as non-relational databases. These databases do not use the tabular schema of rows and columns found in relational databases. They are used for storing a wide range of data sets, NoSQL databases can be divided into;

1. **Key-value stores**

They are one of the simplest database types. Key-value stores work by storing arbitrary data accessible through a specific key. To store data, you provide a key and the data you wish to store such as JSON object, an image, or a plain text. Popular uses of key-value databases are to store configuration values and application variables.

1. **Document-oriented databases**

They store data structured formats called documents often using formats such as JSON, BSON, or XML. Though data within documents is organized within a structure, document databases do not prescribe any specific format or schema.

1. **Graph databases**

They establish connections using the concepts of nodes, edges, and properties. They represent data as individual nodes which can have any number of properties associated with them, between these nodes, edges (relationships) are established to represent different types of connections. The database encodes information about data items within the nodes and information about their relationship in the edges connecting the nodes. Most commonly social media websites use graph databases.

1. **Wide-column databases**

These are databases that store data using a column-oriented model, allowing for efficient storage and retrieval of large amounts of structured and semi-structured data.

**2. List and explain types of database management systems.**

**Relation database management system**

A Relational Database Management System (RDBMS) is a type of database management system (DBMS) that organizes data into tables (relations) and allows for the creation of relationships between them. The data in an RDBMS is stored in a structured format, with each table consisting of rows and columns. Each row represents a single record, and each column represents a field of data within that record.

One of the key features of an RDBMS is the use of a primary key and foreign keys to establish relationships between tables. A primary key is a unique identifier for each record in a table, and a foreign key is a reference to the primary key of another table. This allows for the creation of relationships between tables, such as one-to-one, one-to-many, and many-to-many.

Some popular examples of RDBMSs include MySQL, Oracle Database, and Microsoft SQL Server. They are well suited for structured data and support SQL (Structured Query Language) which is a standard language for interacting with relational databases.

**Object-oriented database management system (OODBMS)**

An Object-oriented Database Management System (OODBMS) is a type of DBMS that organizes data into objects and allows for the creation of classes and inheritance. In an OODBMS, data is stored in a format that is similar to objects in object-oriented programming languages, such as Java or C++. Each object has its own properties, methods, and behaviours. One of the key features of an OODBMS is the ability to model complex relationships and hierarchies within the data.

OODBMSs are well suited for handling complex, unstructured, or semi-structured data, and are often used in applications such as engineering, geographic information systems, and multimedia. Examples of OODBMSs include MongoDB, Apache Cassandra, and ObjectDB.

**Hierarchical database management system (HDBMS)**

A Hierarchical Database Management System (HDBMS) is a type of DBMS that organizes data in a hierarchical tree-like structure. The data is represented as a series of records, with each record having one parent record and one or more child records. The data is stored in a tree-like structure, with each node representing a record and each branch representing a relationship between records. This structure allows for easy traversal of data, but can make it more difficult to represent more complex relationships between records.

HDBMSs were popular in the past, especially for applications such as data modelling for manufacturing systems. IBM’s Information Management System (IMS) is an example of a HDBMS.

**Network database management system (NDBMS)**

A Network Database Management System (NDBMS) is a type of DBMS that organizes data in a network structure. In a NDBMS, data is represented as a series of records, with each record having multiple parents and children. This creates a many-to-many relationship between records, with records being connected to other multiple records, creating a web-like structure. The data is stored in a web-like structure where each node represents a record and each edge represents a relationship between records. This structure allows for easy traversal of data, and can represent complex relationships between records. Examples of NDBMS are, Integrated Data Store (IDS) and Integrated Data Store II (IDS II) developed by Integrated Data Systems.

**NoSQL database management system**

NoSQL (Not Only SQL) Database Management Systems (DBMS) refer to a type of DBMS that do not use the traditional relational model and SQL (Structured Query Language) for data storage and retrieval. Instead, they use alternative data models and query languages that are more suitable for handling large and unstructured data sets. They are often used in web, mobile, gaming, and social media applications, as well as real-time analytics, and machine learning applications.

Common types of NoSQL database management systems include;

* **Document database management system**

They store data in the form of documents and allow for rich, nested data structures. MongoDB, Azure Cosmos DB, and Couchbase are some prominent examples of document database management systems.

* **Columnar database management system**

This type of DBMS is used to manage columnar databases that store data in the form of columns rather than rows and are optimized for high read and write performance. Some databases that use columnar format are Apache Cassandra and Apache HBase.

* **Graph database management system**

They store data as nodes and edges in a graph and are optimized for traversing complex relationships.  Neo4j and OrientDB are examples of graph database management system.

**In-memory database management system**

An In-memory Database Management System is a type of DBMS that stores all of its data in the memory, as opposed to storing it on disk. This allows for faster data access and processing, since data does not need to be read from disk and can be accessed directly from memory. In-memory DBMSs are designed to handle real-time, high-concurrency, and high-throughput workloads. They are often used in applications such as online transaction processing (OLTP), real-time analytics, and high-frequency trading, where low latency and high performance are critical. In-memory DBMSs are becoming more popular as the cost of memory has decreased and the need for real-time data processing has increased. Some examples of in-memory databases are SAP HANA, Oracle TimesTen, and IBM DB2 BLU.