

Types of Distributed Databases Systems (Homogeneous and Heterogeneous)

BY: KASHISH SOOD
(2310993856)

**Chitkara University Institute of Engineering and Technology,
Chitkara University, Punjab**

A **distributed database** is a collection of multiple interconnected databases, which are spread physically across various locations and communicate via a computer network. Selecting the appropriate database architecture is crucial for effectively managing and organizing data across multiple nodes.

A distributed database management system (DDBMS) is a centralized software system that manages a distributed database in a manner as if it were all stored in a single location.

FEATURES:

- It is used to create, retrieve, update and delete distributed databases.
- It synchronizes the database periodically and provides access mechanisms by the virtue of which the distribution becomes transparent to the users.
- It ensures that the data modified at any site is universally updated.
- It is used in application areas where large volumes of data are processed and accessed by numerous users simultaneously.
- It is designed for heterogeneous database platforms.
- It maintains confidentiality and data integrity of the databases.

ADVANTAGES:

Modular Development – If the system needs to be expanded to new locations or new units, in centralized database systems, we can just add new computers and local data to the new site and connect them to the distributed system, with no interruption in current functions.

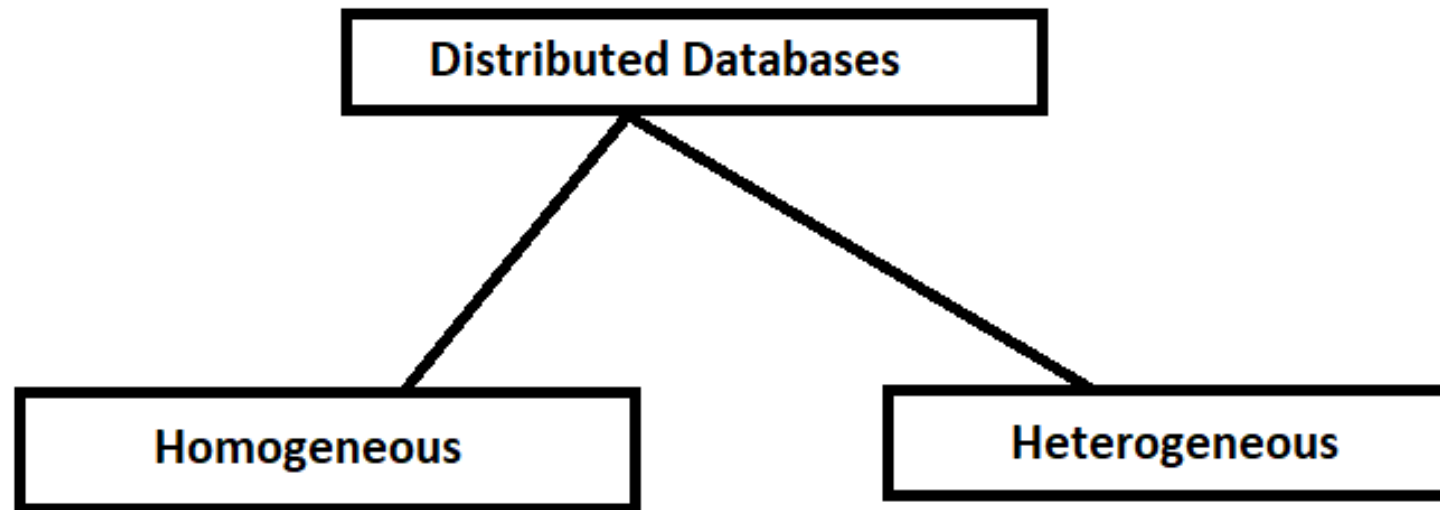
More Reliable – In distributed systems, when a component fails, the functioning of the system continues may be at a reduced performance.

Better Response – User requests can be met from local data itself, providing faster response.

Lower Communication Cost – In DDMS, if data is located locally where it is mostly used, then the communication costs for data manipulation can be minimized. This is not feasible in centralized systems.

CHALLENGES:

- **Need for complex and expensive software** – DDBMS demands complex and often expensive software to provide data transparency and co-ordination across the several sites.
- **Processing overhead** – Even simple operations may require a large number of communications and additional calculations to provide uniformity in data across the sites.
- **Data integrity** – The need for updating data in multiple sites pose problems of data integrity.
- **Overheads for improper data distribution** – Responsiveness of queries is largely dependent upon proper data distribution. Improper data distribution often leads to very slow response to user requests.



Distributed databases can be classified into homogeneous and heterogeneous databases.

Homogeneous databases are characterized by all participating nodes sharing the same DBMS and schema structure. These databases are designed to offer a unified and consistent view of data across all nodes.

Characteristics of Homogeneous Databases

- **Uniformity:** All nodes utilize the same DBMS software and possess identical database schemas.
- **Data Consistency:** Changes made to the database on one node are automatically propagated to other nodes, ensuring data consistency.
- **Simplicity:** Homogeneous databases are relatively easier to manage as the same DBMS software is employed throughout the system.

Steps to Set up a Homogeneous Database:

- **Select a DBMS:** Choose a DBMS that aligns with the distributed system's requirements, such as [MySQL](#), [PostgreSQL](#), or [Oracle](#).
- **Install the DBMS:** Install the chosen DBMS on each node within the distributed system.
- **Design the Schema:** Create a unified database [schema](#) to be shared across all nodes.
- **Establish Communication:** Configure network connectivity between the nodes to facilitate data replication and synchronization.
- **Implement Replication:** Set up replication mechanisms provided by the DBMS to ensure changes made on one node are propagated to others.

HOMOGENEOUS DATABASE SYSTEMS



Consider a distributed e-commerce system with multiple nodes handling customer orders, inventory management, and shipping. Employing a homogeneous database approach, all nodes share the same DBMS (e.g., MySQL) and adhere to a consistent schema. When an order is placed on one node, the system automatically synchronizes the order details and inventory updates across all nodes, enabling real-time visibility and consistency.

Heterogeneous databases allow nodes in a distributed system to use different DBMS software or possess varying database schemas. This approach caters to diverse requirements and facilitates seamless integration between nodes employing different technologies.

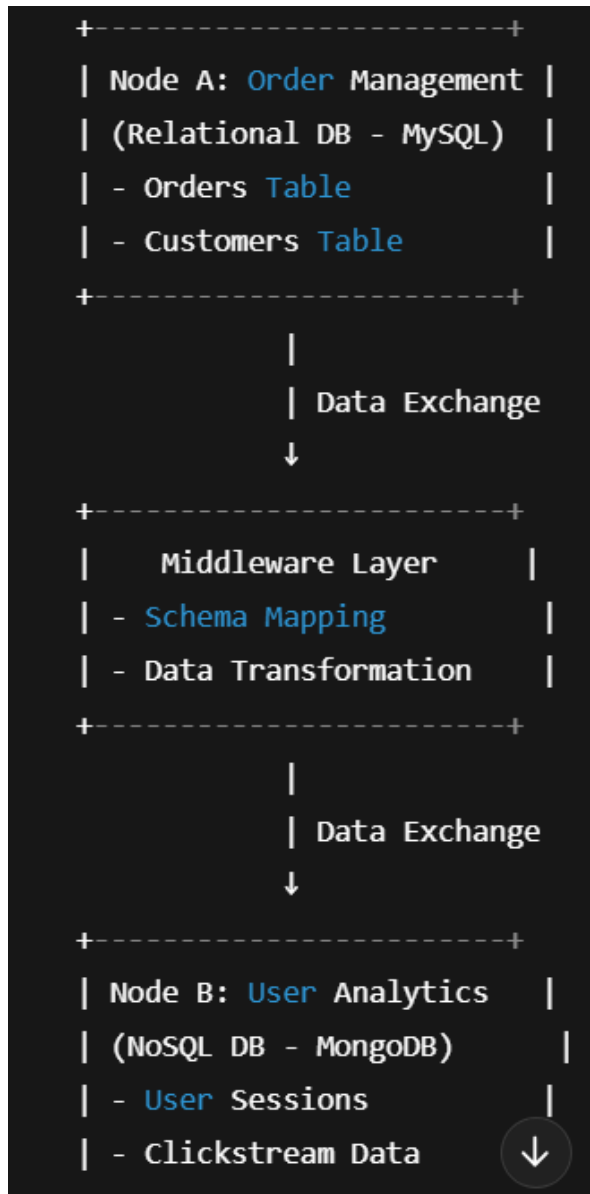
Characteristics of Heterogeneous Databases

- **Flexibility:** Nodes can employ different DBMS software, such as MySQL, [MongoDB](#), etc. based on their specific needs.
- **Schema Mapping:** They necessitate mapping between different schemas to ensure interoperability between nodes.
- **Data Transformation:** Data might need to be transformed or translated between different formats or encodings to maintain consistency.

Steps to Set up a Heterogeneous Database

- **Identify Diverse Requirements:** Understand the specific needs of each node in the distributed system and select the appropriate DBMS software accordingly.
- **Define Schema Mapping:** Analyze the differences in database schemas between nodes and establish mapping rules to convert data between schemas.
- **Implement Data Transformation:** Develop mechanisms or scripts to transform data from one format to another, ensuring seamless integration.
- **Establish Communication:** Configure network connectivity and establish communication channels between heterogeneous nodes.

HETEROGENEOUS DATABASE SYSTEMS



Imagine a distributed system where one node utilizes a traditional relational database (e.g., MySQL) for order management, while another node relies on a NoSQL database (e.g., MongoDB) for user analytics. Employing a heterogeneous database approach enables the two nodes to leverage their preferred DBMS technologies while facilitating data exchange through schema mapping and data transformation.

In the world of distributed databases, the choice between homogeneous and heterogeneous architectures shapes how data is managed across multiple nodes.

Homogeneous databases offer uniformity, simplicity, and data consistency by employing the same DBMS and schema across nodes. This approach is ideal for maintaining a consistent view of data.

Heterogeneous databases provide flexibility to cater to diverse node requirements. They allow different DBMS software and schemas, requiring schema mapping and data transformation. This approach facilitates integration across varying technologies.

*Thank
you!*

