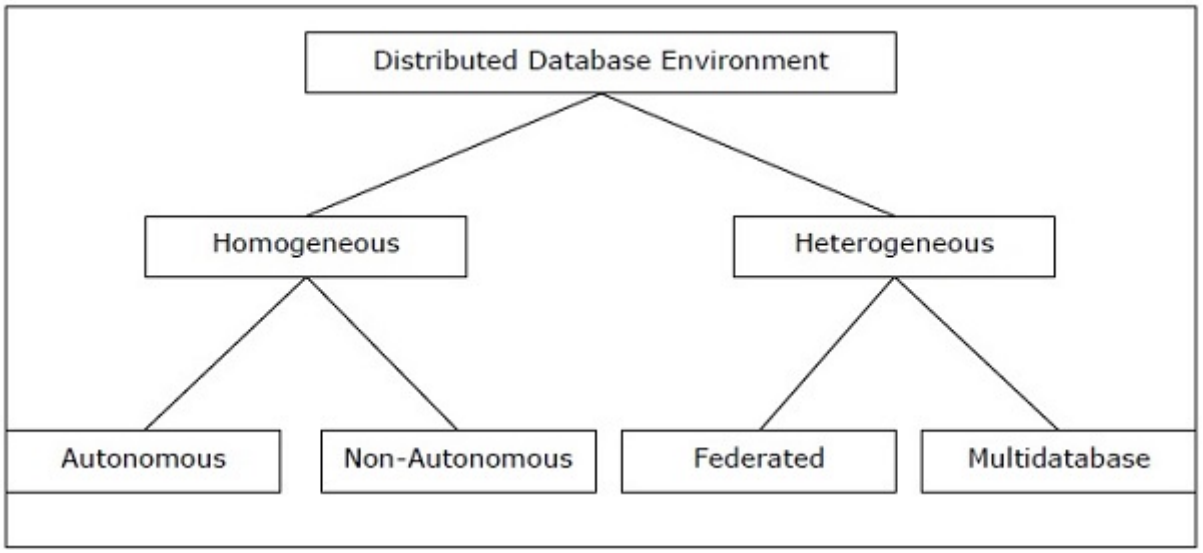
Distributed DBMS - Database Environments

In this part of the tutorial, we will study the different aspects that aid in designing distributed database environments. This chapter starts with the types of distributed databases. Distributed databases can be classified into homogeneous and heterogeneous databases having further divisions. The next section of this chapter discusses the distributed architectures namely client – server, peer – to – peer and multi – DBMS. Finally, the different design alternatives like replication and fragmentation are introduced.

Types of Distributed Databases

Distributed databases can be broadly classified into homogeneous and heterogeneous distributed database environments, each with further sub-divisions, as shown in the following illustration.



Homogeneous Distributed Databases

In a homogeneous distributed database, all the sites use identical DBMS and operating systems. Its properties are –

- The sites use very similar software.
- The sites use identical DBMS or DBMS from the same vendor.
- Each site is aware of all other sites and cooperates with other sites to process user requests.

The database is accessed through a single interface as if it is a single database.

Types of Homogeneous Distributed Database

There are two types of homogeneous distributed database –

Autonomous – Each database is independent that functions on its own. They are integrated by a controlling application and use message passing to share data updates.

Non-autonomous – Data is distributed across the homogeneous nodes and a central or master DBMS co-ordinates data updates across the sites.

Heterogeneous Distributed Databases

In a heterogeneous distributed database, different sites have different operating systems, DBMS products and data models. Its properties are –

Different sites use dissimilar schemas and software.

The system may be composed of a variety of DBMSs like relational, network, hierarchical or object oriented.

Query processing is complex due to dissimilar schemas.

Transaction processing is complex due to dissimilar software.

A site may not be aware of other sites and so there is limited co-operation in processing user requests.

Types of Heterogeneous Distributed Databases

Federated – The heterogeneous database systems are independent in nature and integrated together so that they function as a single database system.

Un-federated – The database systems employ a central coordinating module through which the databases are accessed.

Distributed DBMS Architectures

DDBMS architectures are generally developed depending on three parameters –

Distribution – It states the physical distribution of data across the different sites.

Autonomy – It indicates the distribution of control of the database system and the degree to which each constituent DBMS can operate independently.

Heterogeneity – It refers to the uniformity or dissimilarity of the data models, system components and databases.

Architectural Models

Some of the common architectural models are –

Client - Server Architecture for DDBMS

Peer - to - Peer Architecture for DDBMS

Multi - DBMS Architecture

Client - Server Architecture for DDBMS

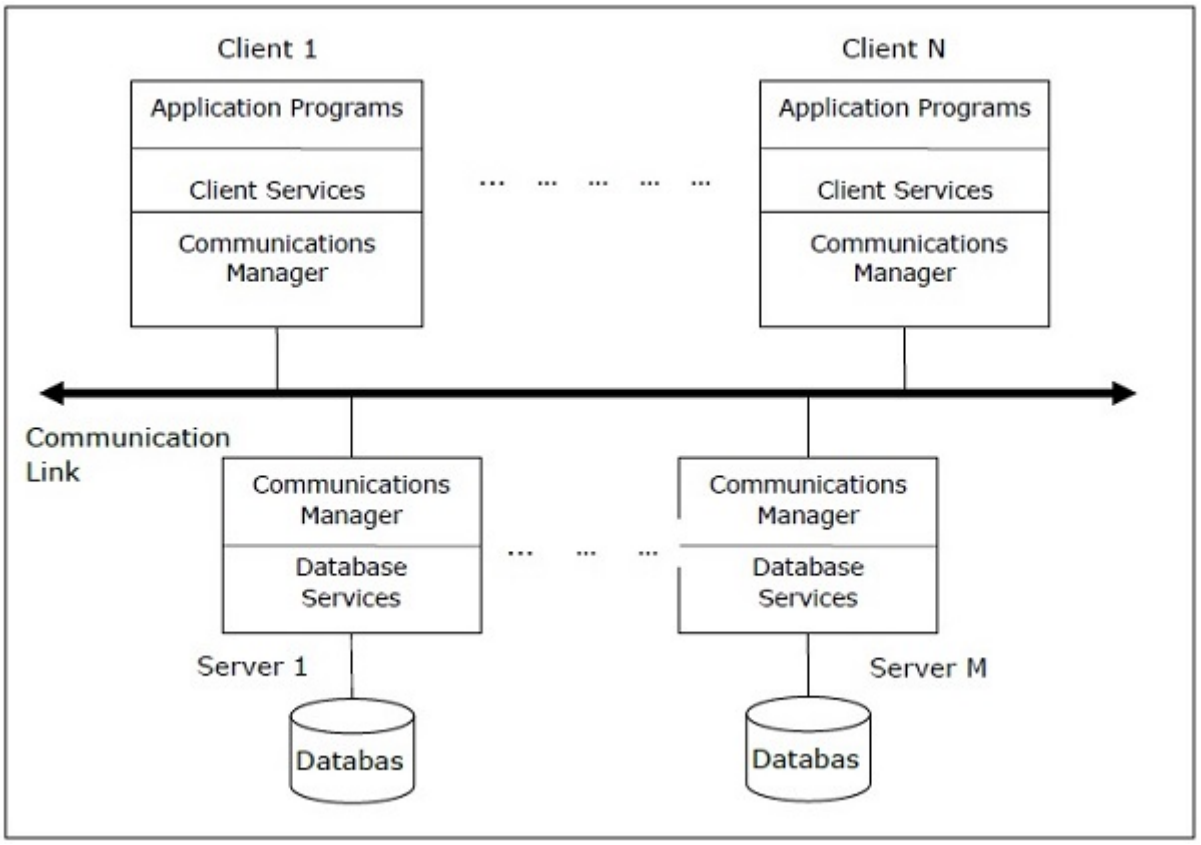
This is a two-level architecture where the functionality is divided into servers and clients. The server functions primarily encompass data management, query processing, optimization and transaction management. Client functions include mainly

user interface. However, they have some functions like consistency checking and transaction management.

The two different client - server architecture are –

Single Server Multiple Client

Multiple Server Multiple Client (shown in the following diagram)



Peer- to-Peer Architecture for DDBMS

In these systems, each peer acts both as a client and a server for imparting database services. The peers share their resource with other peers and co-ordinate their activities.

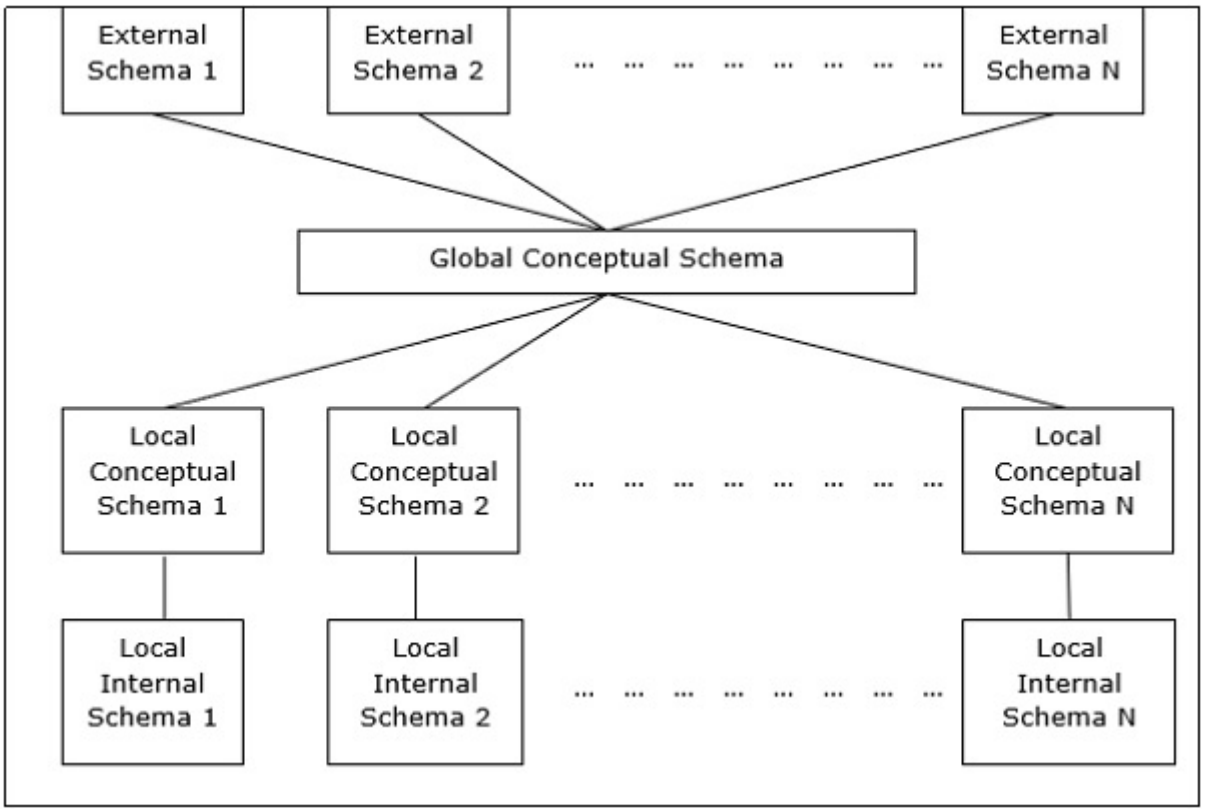
This architecture generally has four levels of schemas –

Global Conceptual Schema – Depicts the global logical view of data.

Local Conceptual Schema – Depicts logical data organization at each site.

Local Internal Schema – Depicts physical data organization at each site.

External Schema – Depicts user view of data.



Multi - DBMS Architectures

This is an integrated database system formed by a collection of two or more autonomous database systems.

Multi-DBMS can be expressed through six levels of schemas –

Multi-database View Level – Depicts multiple user views comprising of subsets of the integrated distributed database.

Multi-database Conceptual Level – Depicts integrated multi-database that comprises of global logical multi-database structure definitions.

Multi-database Internal Level – Depicts the data distribution across different sites and multi-database to local data mapping.

Local database View Level – Depicts public view of local data.

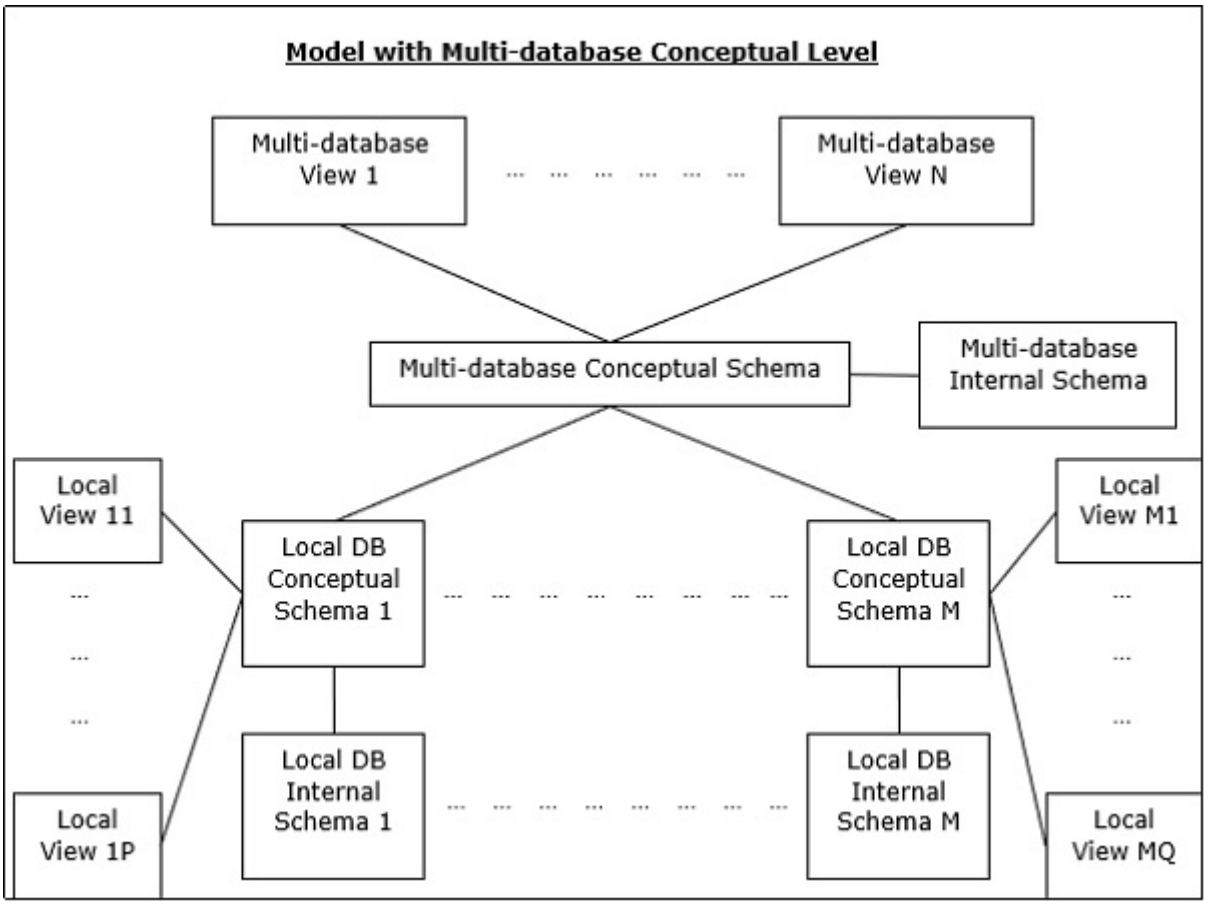
Local database Conceptual Level – Depicts local data organization at each site.

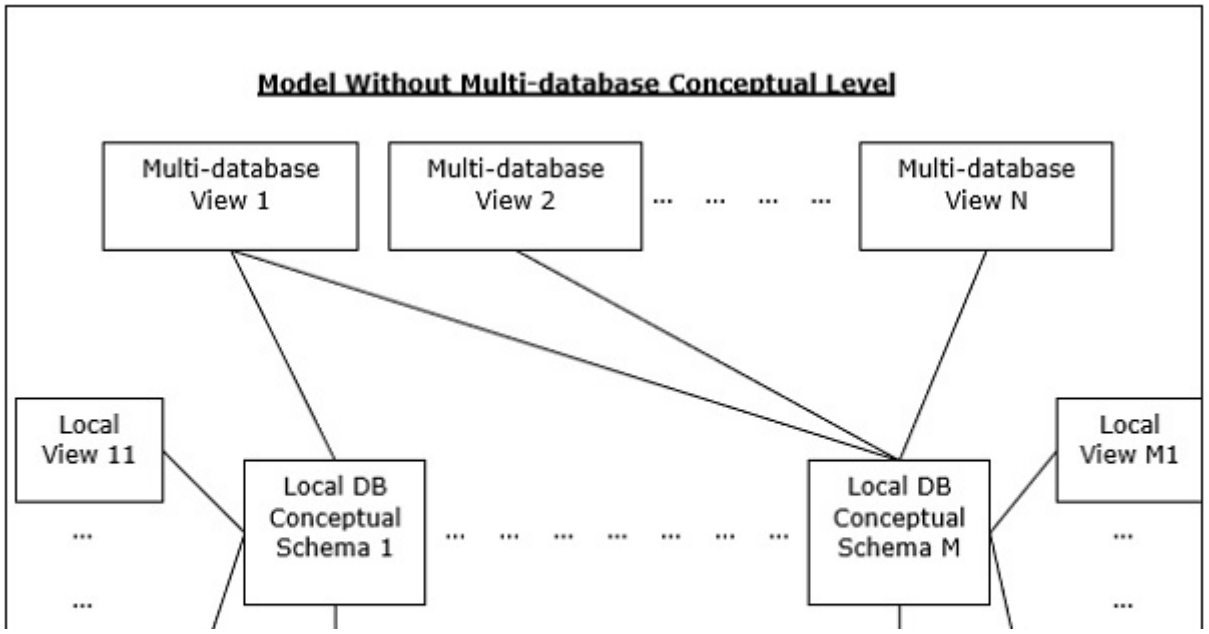
Local database Internal Level – Depicts physical data organization at each site.

There are two design alternatives for multi-DBMS –

Model with multi-database conceptual level.

Model without multi-database conceptual level.





Design Alternatives

The distribution design alternatives for the tables in a DDBMS are as follows –

- Non-replicated and non-fragmented
- Fully replicated
- Partially replicated
- Fragmented
- Mixed

Non-replicated & Non-fragmented

In this design alternative, different tables are placed at different sites. Data is placed so that it is at a close proximity to the site where it is used most. It is most suitable for database systems where the percentage of queries needed to join information in tables placed at different sites is low. If an appropriate distribution strategy is adopted, then this design alternative helps to reduce the communication cost during data processing.

Fully Replicated

In this design alternative, at each site, one copy of all the database tables is stored. Since, each site has its own copy of the entire database, queries are very fast requiring negligible communication cost. On the contrary, the massive redundancy in data requires huge cost during update operations. Hence, this is suitable for systems where a large number of queries is required to be handled whereas the number of database updates is low.

Partially Replicated

Copies of tables or portions of tables are stored at different sites. The distribution of the tables is done in accordance to the frequency of access. This takes into consideration the fact that the frequency of accessing the tables vary considerably from site to site. The number of copies of the tables (or portions) depends on how frequently the access queries execute and the site which generate the access queries.

Fragmented

In this design, a table is divided into two or more pieces referred to as fragments or partitions, and each fragment can be stored at different sites. This considers the fact that it seldom happens that all data stored in a table is required at a given site. Moreover, fragmentation increases parallelism and provides better disaster recovery. Here, there is only one copy of each fragment in the system, i.e. no redundant data.

The three fragmentation techniques are −

- Vertical fragmentation
- Horizontal fragmentation
- Hybrid fragmentation

Mixed Distribution

This is a combination of fragmentation and partial replications. Here, the tables are initially fragmented in any form (horizontal or vertical), and then these fragments are partially replicated across the different sites according to the frequency of accessing the fragments.