Problem Statement

Create table with location field. Create 3 servers-Mumbai, Delhi, and Kolkata. Create global view from three tables and fire query as follows.

"select * from global view"

What is Distributed database?

distributed database is a database that is under the control of a central database management system (DBMS) in which storage devices are not all attached to a common CPU. It may be stored in multiple computers located in the same physical location, or may be dispersed over a network of interconnected computers.

Collections of data (e.g. in a database) can be distributed across multiple physical locations. A distributed database can reside on network servers on the Internet, on corporate intranets or extranets, or on other company networks. Replication and distribution of databases improve database performance at end-user worksites. [1]

To ensure that the distributive databases are up to date and current, there are two processes: replication and duplication. Replication involves using specialized software that looks for changes in the distributive database. Once the changes have been identified, the replication process makes all the databases look the same. The replication process can be very complex and time consuming depending on the size and number of the distributive databases. This process can also require a lot of time and computer resources. Duplication on the other hand is not as complicated. It basically identifies one database as a master and then duplicates that database. The duplication process is normally done at a set time after hours. This is to ensure that each distributed location has the same data. In the duplication process, changes to the master database only are allowed. This is to ensure that local data will not be overwritten. Both of the processes can keep the data current in all distributive locations.[2]

Besides distributed database replication and fragmentation, there are many other distributed database design technologies. For example, local autonomy, synchronous and asynchronous distributed database technologies. These technologies’ implementation can and does depend on the needs of the business
and the sensitivity/confidentiality of the data to be stored in the database, and hence the price the business is willing to spend on ensuring data security, consistency and integrity.

What are Important considerations?

Care with a distributed database must be taken to ensure the following:

The distribution is transparent — users must be able to interact with the system as if it were one logical system. This applies to the system's performance, and methods of access among other things.

Transactions are transparent — each transaction must maintain database integrity across multiple databases. Transactions must also be divided into subtransactions, each subtransaction affecting one database system.

What are advantages

- Management of distributed data with different levels of transparency.
- Increase reliability and availability.
• Easier expansion.
• Reflects organizational structure — database fragments are located in the departments they relate to.
• Local autonomy — a department can control the data about them (as they are the ones familiar with it.)
• Protection of valuable data — if there were ever a catastrophic event such as a fire, all of the data would not be in one place, but distributed in multiple locations.
• Improved performance — data is located near the site of greatest demand, and the database systems themselves are parallelized, allowing load on the databases to be balanced among servers. (A high load on one module of the database won't affect other modules of the database in a distributed database.)
• Economics — it costs less to create a network of smaller computers with the power of a single large computer.
• Modularity — systems can be modified, added and removed from the distributed database without affecting other modules (systems).
• Reliable transactions - Due to replication of database.
• Hardware, Operating System, Network, Fragmentation, DBMS, Replication and Location Independence.
• Continuous operation.

• Distributed Query processing.
• Distributed Transaction management.