**What is data replication?**

Data replication is the process of making multiple copies of data and storing them at different locations to improve their overall accessibility across a network. Similar to [data mirroring](https://www.manageengine.com/device-control/file-shadowing.html), data replication can be applied to both individual computers and servers. The data replicates can be stored within the same system, on-site and off-site hosts, and cloud-based hosts.

## Benefits of data replication

1. Improve the availability of data
2. Increase the speed of data access
3. Enhance server performance
4. Accomplish disaster recovery

### Improve the availability of data

When a particular system experiences a technical glitch due to malware or a faulty hardware component, the data can still be accessed from a different site or node. Data replication enhances the resilience and reliability of systems by storing data at multiple nodes across the network.

### Increase data access speed

In organizations where there are multiple branch offices spread across the globe, users may experience some latency while accessing data from one country to another. Placing replicas on local servers provides users with faster data access and query execution times.

### Enhance server performance

Database replication effectively reduces the load on the primary server by dispersing it among other nodes in the distributed system, thereby improving network performance. By routing all read-operations to a replica database, IT administrators can save the primary server for write-operations that demand more processing power.

### Accomplish Disaster recovery

Businesses are often susceptible to data loss due to a data breach or hardware malfunction. During such a catastrophe, the employees' valuable data, along with client information can be compromised. Data replication facilitates the recovery of data which is lost or corrupted by maintaining accurate backups at well-monitored locations, thereby contributing to enhanced data protection.

## How does data replication work?

Modern day applications use a distributed database in the back end, where data is stored and processed using a cluster of systems, instead of relying on one particular system for the same.

Let us assume that a user of an application wishes to write a piece of data to the database. This data gets split into multiple fragments, with each fragment getting stored on a different node across the distributed system. The database technology is also responsible for gathering and consolidating the different fragments when a user wants to retrieve or read the data.

In such an arrangement, a single system failure can inhibit the retrieval of the entire data. This is where data replication saves the day. Data replication technology can store multiple fragments at each node to streamline read and write operations across the network.

Data replication tools ensure that complete data can still be consolidated from other nodes across the distributed system during the event of a system failure.

## Types of data replication

Depending on data replication tools employed, there are multiple types of replication practiced by businesses today. Some of the popular replication modes are as follows

1. Full table replication
2. Transactional replication
3. Snapshot replication
4. Merge replication
5. Key-based incremental replication

### Full table replication

Full table replication means that the entire data is replicated. This includes new, updated as well as existing data that is copied from source to the destination. This method of replication is generally associated with higher costs since the processing power and network bandwidth requirements are high.

However, full table replication can be beneficial when it comes to the recovery of hard-deleted data, as well as data that do not possess replication keys - discussed further down this article.

### Transactional replication

In this method, the data replication software makes full initial copies of data from origin to destination following which the subscriber database receives updates whenever data is modified. This is more efficient mode of replication since fewer rows are copied each time data is changed. Transactional replication is usually found in server-to-server environments.

### Snapshot replication

In Snapshot replication, data is replicated exactly as it appears at any given time. Unlike other methods, Snapshot replication does not pay attention to the changes made to data. This mode of replication is used when changes made to data tends to be infrequent; for example performing initial synchronizations between publishers and subscribers

### Merge replication

This type of replication is commonly found in server-to-client environments and allows both the publisher and subscriber to make changes to data dynamically. In merge replication, data from two or more databases are combined to form a single database thereby contributing to the complexity of using this technique.

### Key-based incremental replication

Also called key-based incremental data capture, this technique only copies data changed since the last update. Keys can be looked at as elements that exist within databases that trigger data replication. Since only a few rows are copied during each update, the costs are significantly low.

However, the drawback lies in the fact that this replication mode cannot be used to recover hard deleted data, since the key value is also deleted along with the record.

## Data replication schemes in DBMS

Data replication in distribution servers can be carried out using a suitable replication scheme. The widely-adopted replication schemes are as follows:

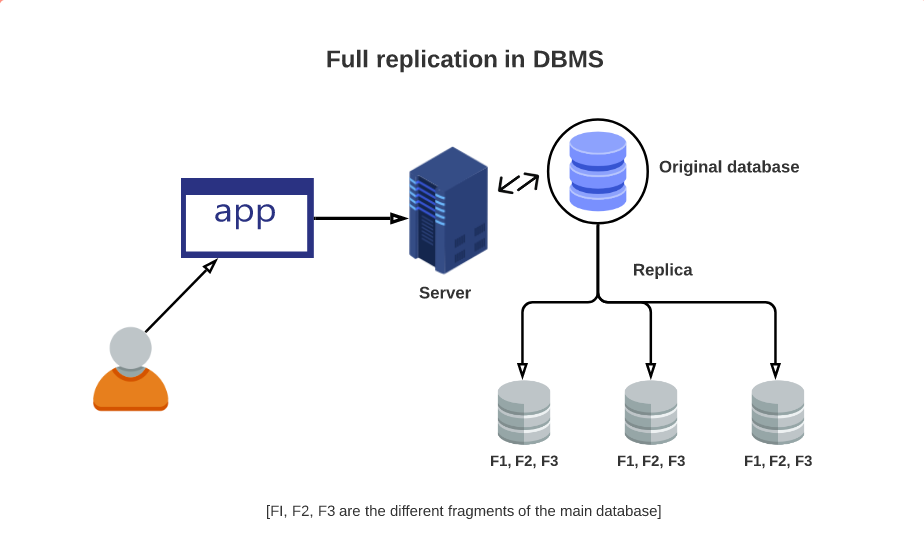
1. Full data replication
2. Partial data replication
3. No replication

### Full data replication

Full replication means that the complete database is replicated at every site of the distributed system. This scheme maximizes data availability and redundancy across a wide area network.

For example, users in a cross-country network have access to the complete database from an Asia based server if the European or North American server experiences a technical difficulty.

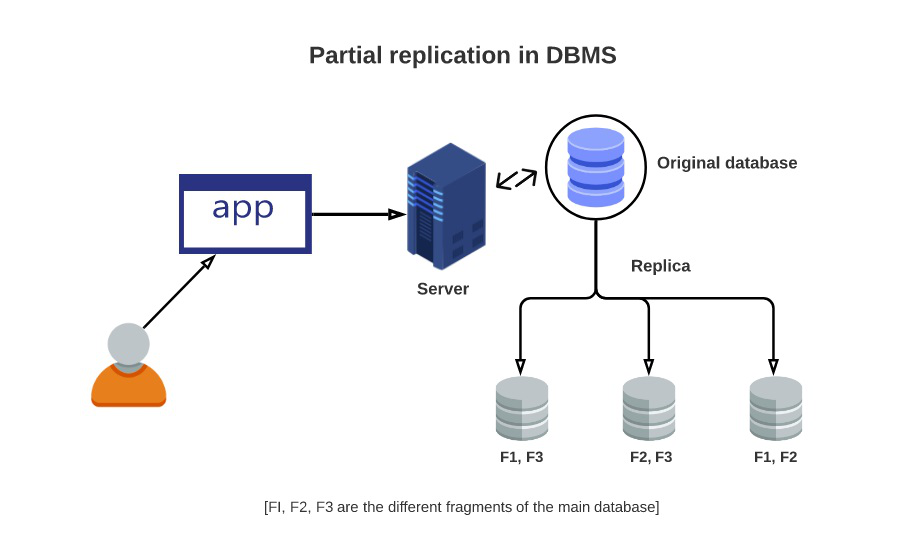
Full replication also contributes to faster execution of global queries as the results can be obtained from any local server.The disadvantage of full replication is that the update process tends to be on the slower side. This makes keeping up-to-date copies of data at every location quite challenging.



### Partial data replication

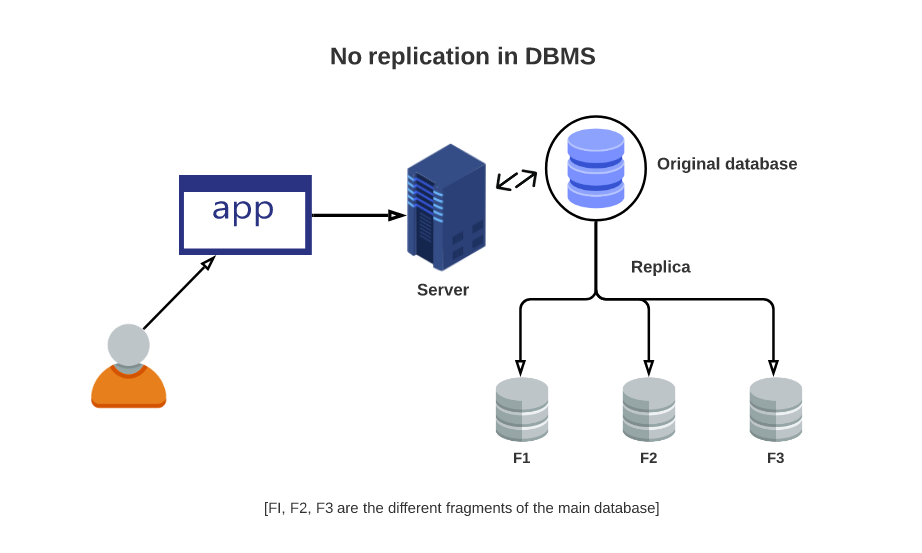
Partial replication occurs when only certain fragments of the database are replicated based on the importance of data at each location. Here, the number of copies can range from one to the total number of nodes in the distributed system.

In an enterprise environment, this mode of replication can be useful for members of sales and marketing teams where a partial database is stored on personal computers and regularly synced with the main server.



### No replication

In this mode of replication, only one fragment exists on each site of the distributed system. While no replication can be attributed to the ease of data recovery, it can have an adverse effect on the speed of execution of queries since multiple users access the same server. Compared to other replication schemes, no replication provides poor availability of data.



Some **master-slave distribution models** have single points of failure that might impact your system availability, so you might need to take special care when configuring these systems.

Distribution models get to the heart of the question who’s in charge here? There are two ways to answer this question: **one node** or **all nodes**. In the master-slave model, one node is in charge (master). When there’s no single node with a special role in taking charge, you have a peer-to-peer distribution model.

**Peer-to-peer systems distribute** the responsibility of the master to each node in the cluster. In this situation, testing is much easier since you can remove any node in the cluster and the other nodes will continue to function. The disadvantage of peer-to-peer networks is that there’s an increased complexity and communication overhead that must occur for all nodes to be kept up to date with the cluster status.