Distributed Database

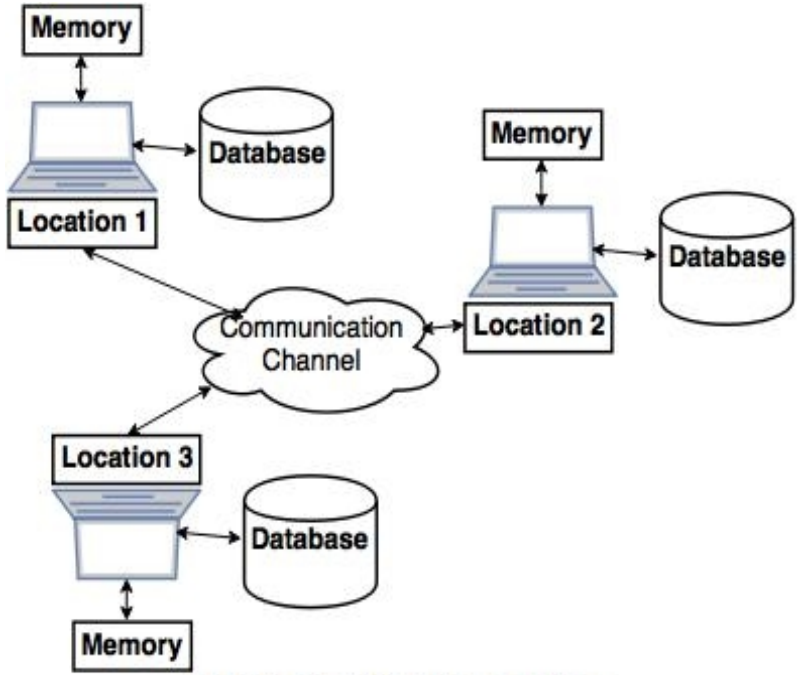
By,
Harshil T. Kanakia

Outline

- Overview
- Types of Distributed Database
- Data Fragmentation
- Data Replication
- Query Processing
- Concurrency Control

Distributed Database

A distributed database is a database that is not limited to one system, it is spread over different sites, i.e, on multiple computers or over a network of computers. A distributed database system is located on various sites that don't share physical components. This may be required when a particular database needs to be accessed by various users globally. It needs to be managed such that for the users it looks like one single database.



Distributted Dtatabase system

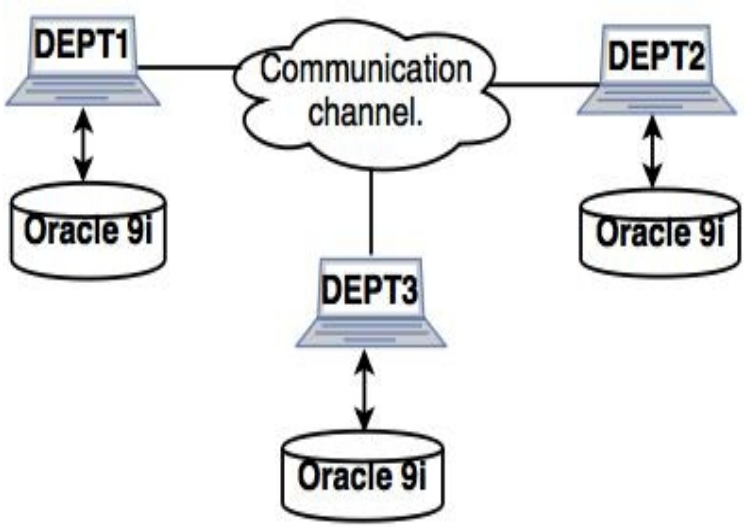
Types of Distributed Database

- Homogeneous Distributed Database

- Heterogeneous Distributed Database

Homogeneous Distributed Database

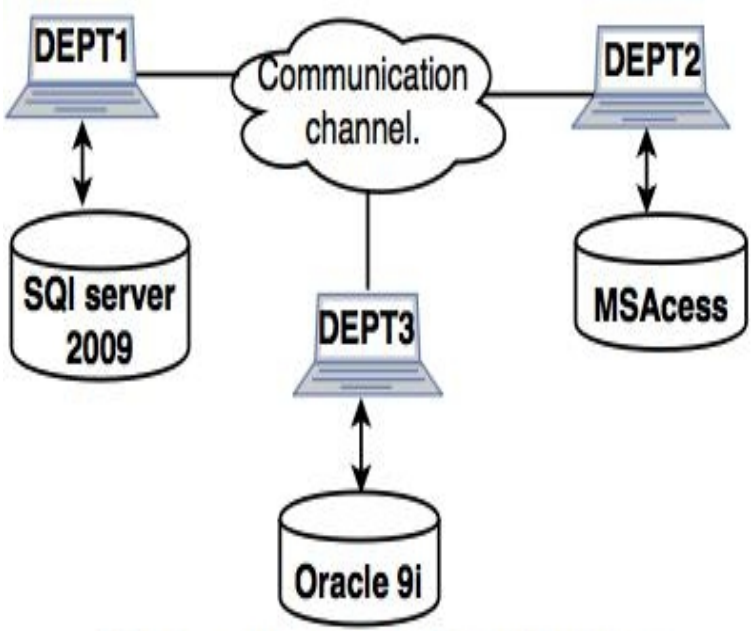
In a homogeneous database, all different sites store database identically. The operating system, database management system and the data structures used – all are same at all sites. Hence, they're easy to manage.



Homogeneous distributed system

Heterogeneous Distributed Database

In a heterogeneous distributed database, different sites can use different schema and software that can lead to problems in query processing and transactions. Also, a particular site might be completely unaware of the other sites. Different computers may use a different operating system, different database application. They may even use different data models for the database. Hence, translations are required for different sites to communicate.



Heterogeneous distributed system

Data Fragmentation

- In this approach, the relations are fragmented (i.e., they're divided into smaller parts) and each of the fragments is stored in different sites where they're required. It must be made sure that the fragments are such that they can be used to reconstruct the original relation (i.e, there isn't any loss of data).
- Fragmentation is advantageous as it doesn't create copies of data, consistency is not a problem.
- Three Ways:
  Horizontal fragmentation
  Vertical fragmentation
  Hybrid fragmentation

Horizontal Fragmentation - Splitting by rows

Horizontal fragmentation refers to the process of dividing a table horizontally by assigning each row or (a group of rows) of relation to one or more fragments. These fragments are then be assigned to different sides in the distributed system. Some of the rows or tuples of the table are placed in one system and the rest are placed in other systems. The rows that belong to the horizontal fragments are specified by a condition on one or more attributes of the relation.

Horizontal Fragmentation - Splitting by rows Example

This is global Table

CustomerID	CustomerName	City	Gender
1	Bob	Mumbai	Male
2	Alice	Bangalore	Female
3	Milind	Agra	Male
4	Jaya	Pune	Female

Horizontal Fragment

Fragment 1

CustomerID	CustomerName	City	Gender
1	Bob	Mumbai	Male
3	Milind	Agra	Male

Fragment 2

CustomerID	CustomerName	City	Gender
2	Alice	Bangalore	Female
4	Jaya	Pune	Female

Vertical Fragmentation - Splitting by columns

Vertical fragmentation refers to the process of decomposing a table vertically by attributes are columns. In this fragmentation, some of the attributes are stored in one system and the rest are stored in other systems. This is because each site may not need all columns of a table. In order to take care of restoration, each fragment must contain the primary key field(s) in a table. The fragmentation should be in such a manner that we can rebuild a table from the fragment by taking the natural JOIN operation and to make it possible we need to include a special attribute called Tuple-id to the schema.

Vertical Fragmentation - Splitting by columns Example

This is global Table

CustomerID	CustomerName	City	Gender
1	Bob	Mumbai	Male
2	Alice	Bangalore	Female
3	Milind	Agra	Male
4	Jaya	Pune	Female

Vertical Fragment

Fragment 1

CustomerID	CustomerName
1	Bob
2	Alice
3	Milind
4	Jaya

Fragment 2

CustomerID	City	Gender
1	Mumbai	Male
2	Bangalore	Female
3	Agra	Male
4	Pune	Female

Hybrid Fragmentation

- The combination of vertical fragmentation of a table followed by further horizontal fragmentation of some fragments is called mixed or hybrid fragmentation.
- Mixed fragmentation can be done in two different ways:

1. The first method is to first create a set or group of horizontal fragments and then create vertical fragments from one or more of the horizontal fragments.
2. The second method is to first create a set or group of vertical fragments and then create horizontal fragments from one or more of the vertical fragments.

- The original relation can be obtained by the combination of JOIN and UNION operations.

Data Replication

- In this approach, the entire relation is stored redundantly at 2 or more sites. If the entire database is available at all sites, it is a fully redundant database. Hence, in replication, systems maintain copies of data.
- This is advantageous as it increases the availability of data at different sites. Also, query requests can be processed in parallel.
- However, it has certain disadvantages as well. Data needs to be constantly updated. Any change made at one site needs to be recorded at every site that relation is stored or else it may lead to inconsistency. This is a lot of overhead. Also, concurrency control becomes way more complex as concurrent access now needs to be checked over a number of sites.

Types of Data Replication

- Synchronous Replication:

The replica will be modified immediately after some changes are made in the relation table. So there is no difference between original data and replica.

- Asynchronous replication

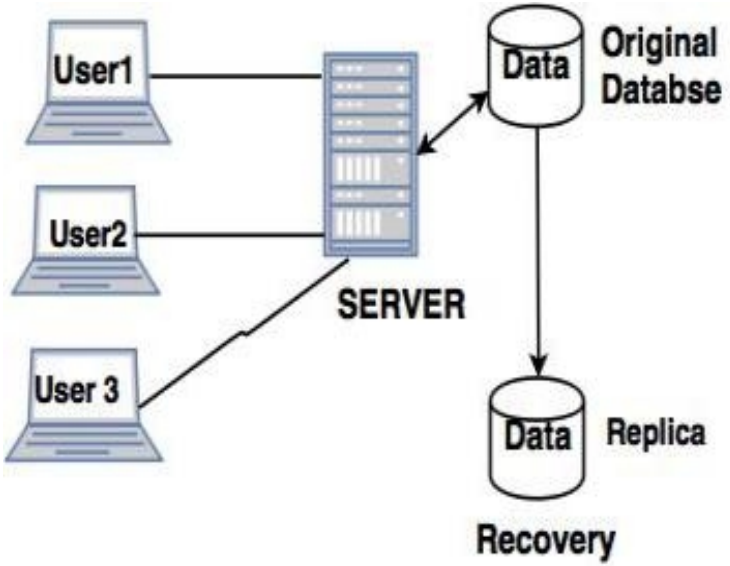
The replica will be modified after commit is fired on to the database.

Replication Scheme

- Full Replication
- No Replication
- Partial Replication

Full Replication

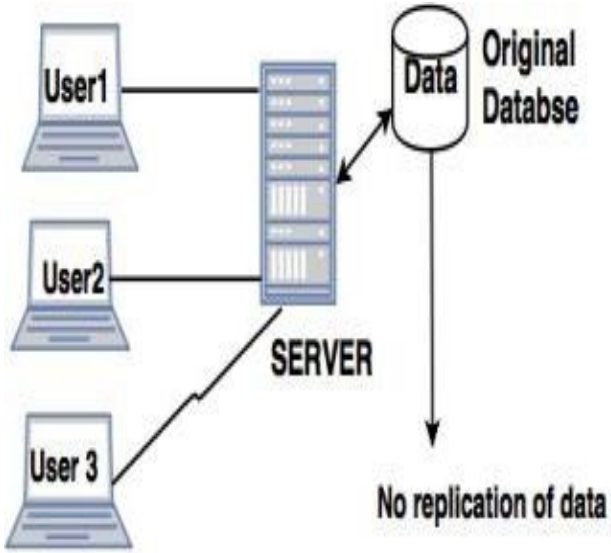
- The database is available to almost every location.
- Advantages:
  1. High availability of data.
  2. Faster execution of queries.
- Disadvantages:
  1. Concurrency Control is difficult.
  2. Update operation is slower.



Full Replication Process In Distributed System

No Replication

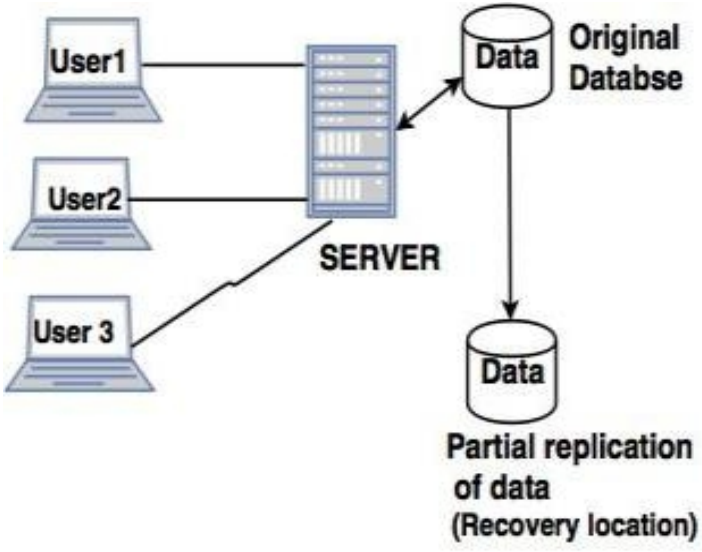
- No replication means, each fragment is stored exactly at one location.
- Advantages:
  1. Concurrency can be minimized.
  2. Easy recovery of data.
- Disadvantages:
  1. Poor availability of data.
  2. Slows down the query execution process, as multiple clients are accessing the same server.



No Replication Process in Distributed Databases

Partial Replication

Partial replication means only some fragments are replicated from the database.



Partial Replication Process In Distributed System