Aim:

To study distributed database design.

Theory :

The strategies can be broadly divided into replication and fragmentation. However, in most cases, a combination of the two is used.

1. Data Replication

Data replication is the process of storing separate copies of the database at two or more sites. It is a popular fault tolerance technique of distributed databases.

Advantages of Data Replication

* **Reliability** − In case of failure of any site, the database system continues to work since a copy is available at another site(s).
* **Reduction in Network Load** − Since local copies of data are available, query processing can be done with reduced network usage, particularly during prime hours. Data updating can be done at non-prime hours.
* **Quicker Response** − Availability of local copies of data ensures quick query processing and consequently quick response time.
* **Simpler Transactions** − Transactions require less number of joins of tables located at different sites and minimal coordination across the network. Thus, they become simpler in nature.

Disadvantages of Data Replication

* **Increased Storage Requirements** − Maintaining multiple copies of data is associated with increased storage costs. The storage space required is in multiples of the storage required for a centralized system.
* **Increased Cost and Complexity of Data Updating** − Each time a data item is updated, the update needs to be reflected in all the copies of the data at the different sites. This requires complex synchronization techniques and protocols.
* **Undesirable Application – Database coupling** − If complex update mechanisms are not used, removing data inconsistency requires complex co-ordination at application level. This results in undesirable application – database coupling.

Some commonly used replication techniques are −

* Snapshot replication
* Near-real-time replication
* Pull replication

1. Fragmentation

Fragmentation is the task of dividing a table into a set of smaller tables. The subsets of the table are called **fragments**. Fragmentation can be of three types: horizontal, vertical, and hybrid (combination of horizontal and vertical). Horizontal fragmentation can further be classified into two techniques: primary horizontal fragmentation and derived horizontal fragmentation.

Fragmentation should be done in a way so that the original table can be reconstructed from the fragments. This is needed so that the original table can be reconstructed from the fragments whenever required. This requirement is called “reconstructiveness.”

Advantages of Fragmentation

* Since data is stored close to the site of usage, efficiency of the database system is increased.
* Local query optimization techniques are sufficient for most queries since data is locally available.
* Since irrelevant data is not available at the sites, security and privacy of the database system can be maintained.

Disadvantages of Fragmentation

* When data from different fragments are required, the access speeds may be very high.
* In case of recursive fragmentations, the job of reconstruction will need expensive techniques.
* Lack of back-up copies of data in different sites may render the database ineffective in case of failure of a site.

2.1 Vertical Fragmentation

In vertical fragmentation, the fields or columns of a table are grouped into fragments. In order to maintain reconstructiveness, each fragment should contain the primary key field(s) of the table. Vertical fragmentation can be used to enforce privacy of data.

For example, let us consider that a University database keeps records of all registered students in a Student table having the following schema.

STUDENT

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Regd\_No | Name | Course | Address | Semester | Fees | Marks |

Now, the fees details are maintained in the accounts section. In this case, the designer will fragment the database as follows −

CREATE TABLE STD\_FEES AS

SELECT Regd\_No, Fees

FROM STUDENT;

2.2 Horizontal Fragmentation

Horizontal fragmentation groups the tuples of a table in accordance to values of one or more fields. Horizontal fragmentation should also confirm to the rule of reconstructiveness. Each horizontal fragment must have all columns of the original base table.

For example, in the student schema, if the details of all students of Computer Science Course needs to be maintained at the School of Computer Science, then the designer will horizontally fragment the database as follows −

CREATE COMP\_STD AS

SELECT \* FROM STUDENT

WHERE COURSE = "Computer Science";

2.3 Hybrid Fragmentation

In hybrid fragmentation, a combination of horizontal and vertical fragmentation techniques are used. This is the most flexible fragmentation technique since it generates fragments with minimal extraneous information. However, reconstruction of the original table is often an expensive task.

Hybrid fragmentation can be done in two alternative ways −

* At first, generate a set of horizontal fragments; then generate vertical fragments from one or more of the horizontal fragments.
* At first, generate a set of vertical fragments; then generate horizontal fragments from one or more of the vertical fragments.