Design Strategies

In the last chapter, we had introduced different design alternatives. In this chapter, we will study the strategies that aid in adopting the designs. The strategies can be broadly divided into replication and fragmentation. However, in most cases, a combination of the two is used.

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

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

1. Permits a number of transactions to executed concurrently
2. Results in parallel execution of a single query
3. Increases level of concurrency, also referred to as, intra query concurrency
4. Increased System throughput.
5. Since data is stored close to the site of usage, efficiency of the database system is increased.
6. Local query optimization techniques are sufficient for most queries since data is locally available.
7. Since irrelevant data is not available at the sites, security and privacy of the database system can be maintained.

Disadvantages

1. Applications whose views are defined on more than one fragment may suffer performance degradation, if applications have conflicting requirements.
2. Simple tasks like checking for dependencies, would result in chasing after data in a number of sites
3. When data from different fragments are required, the access speeds may be very high.
4. In case of recursive fragmentations, the job of reconstruction will need expensive techniques.
5. Lack of back-up copies of data in different sites may render the database ineffective in case of failure of a site.

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.

Grouping

- Starts by assigning each attribute to one fragment
    - At each step, joins some of the fragments until some criteria is satisfied.
- Results in overlapping fragments

Splitting

- Starts with a relation and decides on beneficial partitioning based on the access behavior of applications to the attributes
- Fits more naturally within the top-down design
- Generates non-overlapping fragments

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

```
CREATE TABLE STD_FEES AS
SELECT Regd_No, Fees
FROM STUDENT;
```

fragment

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.

- Primary horizontal fragmentation is defined by a selection operation on the owner relation of a database schema.
- Given relation R_i, its horizontal fragments are given by

$R_i = \sigma_{Fi}(R), \quad 1 <= i <= w$

Fi selection formula used to obtain fragment R_i

The example mentioned in slide 20, can be represented by using the above formula as

$$Emp_1 = \sigma_{Sal <= 20K}(Emp)$$

$$Emp_2 = \sigma_{Sal > 20K}(Emp)$$

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";
```

Derived Horizontal Fragmentation

- Defined on a member relation of a link according to a selection operation specified on its owner.
- Link between the owner and the member relations is defined as equi-join
- An equi-join can be implemented by means of semijoins.
- Given a link L where owner (L) = S and member (L) = R, the derived horizontal fragments of R are defined as

$R_i = R \alpha S_i, \quad 1 <= I <= w$

Where,

$$S_i = \sigma F_i (S)$$

w is the max number of fragments that will be defined on

F_i is the formula using which the primary horizontal fragment S_i is defined

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.