## UNIT-3

# 1. A FRAMEWORK FOR DISTRIBUTED DATABASE

We apply following steps in the framework of distributed database

1. Designing the "conceptual schema" – it describes the integrated database, in which all the data used by database applications
2. Designing the "physical database" – it maps the conceptual schema to different storage areas.
3. Designing the "fragmentation" – it determines how the global schema are divided into horizontal, vertical, or mixed fragmentation
4. Designing the "allocation of fragments" – determine how fragments are mapped to physical images

First two steps show the design of the global schema and design of the local databases at each site in distributed databases. Step 3 and 4 shows the characterization of design of data distribution, i.e. in detailed fragmentation of global relation and physical placement of the data at various sites.

**Application requirements may also influence the schema design**. Application requirements are included as follows:

1. The site from which the application is issued, know as **site of origin** of application
2. The frequency of activation of the application
3. The number, type, and the statistical information of data objects of application

## Objectives of the Design of Data Distribution

### 1. Processing Locality

Place the data near to the application site where the application is to run is known as processing locality. We maximize the processing locality by increasing the local references and remote references to each candidate fragmentation.

Applications are completely executed at their sites of origin is known as **complete locality**.

### 2. Availability and Reliability of Distributed Data

A high degree of availability for read-only applications achieved by storing multiple copies of the same information.

Reliability is also achieved by storing multiple copies of the same information for recovering of data during either crashes or physical destruction

### 3. Workload Distribution

Distributing the workload over the sites is an important feature of distributed computing systems. It maximizes the degree of parallelism of execution of applications.

### 4. Storage costs and availability

Database distribution should reflect the cost and availability of storage at the different sites. It is possible to have specialized sites in the network for data storage.

**Top-Down and Bottom-Up Approaches to the Design of Data Distribution**

**Top-Down Approach**

In the top-down approach, we start the designing the global schema and we proceed by designing the fragmentation of the database, and then allocating the fragments to the sites, creating the physical images.

(explain the horizontal and vertical fragmentation also)

**Bottom-Up Approach**

When the distributed database is developed as the aggregation of existing databases is known as bottom-up approach. This approach is based on the **integration** of existing schemata into a single, global schema.

The bottom-up design of a distributed database requires the following developments

1. The **selection** of a **common database model** for describing the global schema of the database
2. The **translation** of each local schema into the common data model
3. The **integration** of the local schemata into a common global schema

**2. THE DESIGN OF DATABASE FRAGMENTATION**

The purpose of fragmentation design is to determine "non-overlapping fragments", which are "logical units of allocation".

Designing of fragmentation consists of grouping tuples as fragments in horizontal fragmentation and grouping the attributes as fragments in vertical fragmentation.

Each group of tuples or attributes having the "same properties", constitute as fragments.

Example:

**Consider the horizontal fragmentation** for a global relation EMP.

**Distributed database application: Determine employees are members of projects from EMP**

Let each department be a site of the distributed database and application can be issued at any department. Existing employees are distributed in different departments.

Each project is made in one department. In our application, we expect the members of project are in the same department. Thus, tuples of each department are collected by determining the employees who work at the same department.

**Vertical fragmentation for relation EMP**

Assume that attributes SAL and TAX are used only by administrative applications, which always use these attributes together. Thus, pair of SAL and TAX are showing appropriate vertical fragmentation.

### 2.1 Horizontal Fragmentation

Two types of horizontal fragmentation are called as primary and derived. Derived fragmentation is derived in terms of primary fragmentation.

Horizontal fragmentation deals with following

1. Logical properties of the data – fragmentation predicates
2. Statistical properties of the data – define number of references of applications to fragments

#### 2.1.1 Primary Fragmentation

Primary horizontal fragments are defined using selections on global relations, in which each tuple can be selected in one and only one fragments. Determining the primary fragmentation of a global relation requires a set of disjoint and complete selection predicates. Therefore, we use following definitions:

1. A simple predicate is a type of the predicate
   Attribute = value
2. A minterm predicate y for a set of P of simple predicates is the conjunction of all predicates appearing in P
   $y = \wedge_{p_i \in p} p_i^*$, where ($p_i^* = p_i$ or $p_i^* = NOT\ p_i$) and y≠false
3. A fragment is the set of all tuples for which a minterm predicate holds.

**Example:**

**Determine the employee's information who are the members of projects and also require the data of employees who are the programmers**

Let assume that there are only two departments 1 and 2 and required project members are in DEPT = 1 only.

DEPT = 1 is same as DEPT ≠ 2 and also vice-versa

Simple predicate of give example is DEPT =1 and JOB = "P"

Possible Minterm Predicates are

DEPT =1 and JOB = "P"

DEPT =1 and JOB ≠ "P"

DEPT ≠ 1 and JOB = "P"

DEPT ≠ 1 and JOB ≠ "P"

Let P = {p1, p2, .........pn} be a set of simple predicates. If fragmentation is efficient and correct if and only if P must be complete and minimal.

**Complete** - we say the predicate P is **complete** if and only if any two tuples belong to same fragment are referenced with same fragment by any application

**Minimal** – we say that P is **minimal** if all its predicates are relevant

**Example**

P1={DEPT = 1} is not complete

P2 = {DEPT = 1, JOB = "P"} is complete and minimal

P3 = {DEPT = 1, JOB = "P", SAL >50 } is complete but not minimal

**General Example 1:**

Consider the distributed database for a company in California having three sites at San Francisco (site 1), Fresno (site 2), and Los Angeles (site 3). Fresno is located between site 1 and site 3. There are 30 departments, in which first 10 are close to site 1, 11 to 20 are close to site 2, and 21 to 30 are close to site 3.

Global schema : EXAMPLE_DDB

Other Relations : EMP, DEPT, SUPPLIER, SUPPLY

Database Application : Names of suppliers with a given SNUM for relation SUPPLIER (SNUM, NAME, CITY)

Select NAME

from SUPPLIER

where SNUM = $X

This distributed application is issued at any site.

If it is issued at site 1, it references SUPPLIERS whose CITY is San Francisco with 80% probability

If it is issued at site 2, it references SUPPLIER of San Francisco (SF) and Loss Angeles (LA) with equal probability

If it is issued at site 3, it references SUPPLIER of Loss Angeles (LA) with 80% probability.

We can apply this application for producing the possible simple predicates p1 and p2 as per follows

p1: CITY = "SF"

p2: CITY ="LA"

the set {p1, p2} is now complete and minimal.

**General Example 2:**

Global Relation : DEPT (DEPTNO, NAME, AREA, MGRNUM)

Database Application : Administrative application is issued at sites 1 and 3 for the respective departments

Possible set of Predicates :

p1: DEPTNO ≤ 10

p2 : 10 < DEPTNO ≤ 10

p3 : DEPTNO > 20

p4 : AREA = "North"

P5 : AREA = "South"

There are implications between AREA and DEPTNO, thus reduced fragmentation is defined as follows:

y1: DEPTNO ≤ 10

y2 : (10 < DEPTNO) ≤ 10 and (AREA="North")

y3 : (10 < DEPTNO) ≤ 10 and (AREA="South")

y4 : DEPTNO > 20

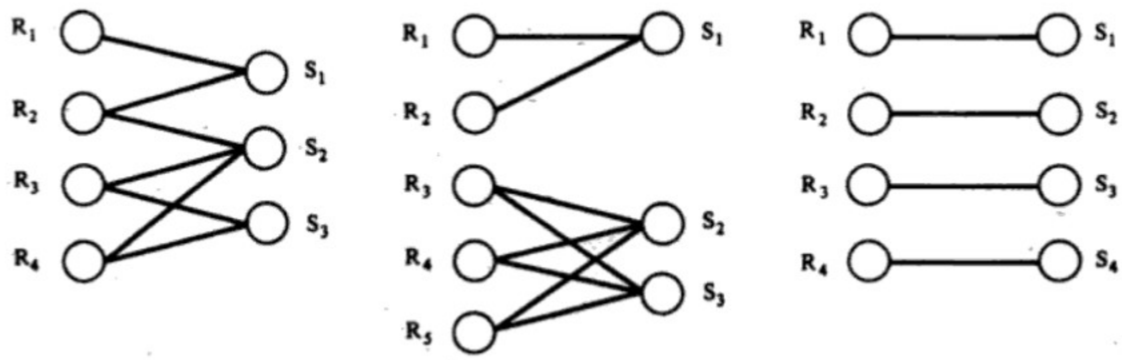
Therefore, final fragmentation of relation DEPT is shown in following table.

| | p4: AREA = "North" | p5: AREA = "South" |
|---|---|---|
| p1: DEPTNO ≤ 10 | y1 | FALSE |
| p2 : 10 < DEPTNO ≤ 10 | y2 | y3 |
| p3 : DEPTNO > 20 | FALSE | y4 |

### 2.1.2 Derived Horizontal Fragmentation

The derived horizontal fragmentation of a global relation R is not based n properties of its own attributes, but it is derived from the horizontal fragmentation of another result. Derive fragmentation is used to facilitate join between fragments.

A **distributed join** is a join between horizontally fragmented relations. A distributed join is represented using **join graphs.** The join graph G of the distributed join R JN S is a graph < N, E>, where nodes N represent fragments of R and S and non-directed edges between nodes represent joins between fragments. The join graphs are shown in following figure.




(a) Join graph. (b) Partitioned join graph. (c) Simple join graph.

### 2.1.3 Vertical Fragmentation

Determining the vertical fragmentation of a global relation R requires grouping the attributes as set of fragments. The following approaches are used for vertical fragmentation

1. **The split approach**- in which global relations are progressively split into fragments
2. **The grouping approach** – in which attributes are progressively aggregating to constitute fragments.

In above cases, formulas are used to indicate which is the best splitting or grouping

General Example

Global Relation : EMP (EMPNO, NAME, SAL, TAX, MGRNUM, DEPTNO)

Vertical Fragmentation Results:

EMP1(EMPNO, NAME, SAL, TAX)

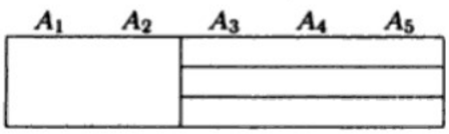
EMP2(EMPNO, MGRNUM, DEPTNO)

### 2.1.4 Mixed Fragmentation

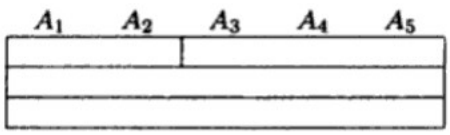
The mixed fragmentation consists of :

1. Applying horizontal fragmentation to vertical fragments
2. Applying vertical fragmentation to horizontal fragments

The following figures shows the mixed fragmentation




(a) Vertical fragmentation followed by horizontal fragmentation




(b) Horizontal fragmentation followed by vertical fragmentation