LEVELS OF DISTRIBUTION TRANSPARENCY

UNIT-2

2.1 Reference Architecture for Distributed Databases

Reference architecture is not explicitly defined in all the distributed databases; however, its levels are conceptually relevant to organization of any distributed database. Following Fig. 2.1 shows a reference architecture for a distributed database.

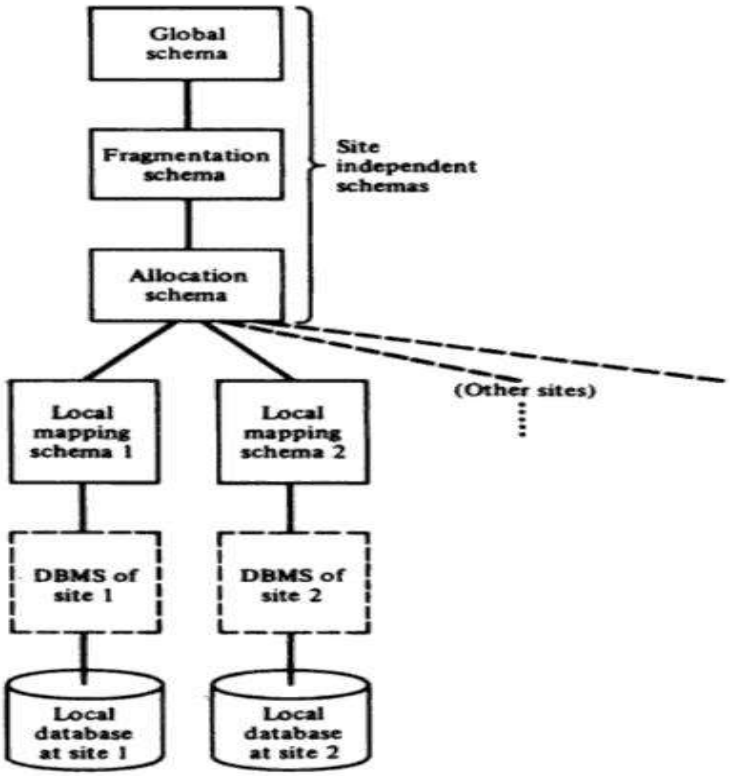


Fig. 2.1: A Reference Architecture for Distributed Databases

The **global schema** defines all the data which are contained in the distributed database as if the database were not distributed at all. We use relational model for defining the data model of the global schema. Global schema consists of the definition of a set of global relations.

Each global schema can be split into several non-overlapping portions are called **fragments.** The relation between global schema and fragments is defined in the **fragmentation schema**. Several fragments are corresponding to a single global relation. Hence, it is one to many. Fragments are indicated by a global relation name with an index (fragment index); For example, R_i indicates the i^{th} fragment of global relation R.

Allocation schema defines at which site(s) a fragment is located. **Type of mapping** in the allocation schema determines the whether the distributed database is redundant or non-redundant. In the redundant case, type of mapping is one to many, whereas as if it is non-redundant, then type of mapping is one to one.

All the fragment which corresponds to same global relation R and are located at the same site j constitute physical image of global relation at site j.

Example is shown in Fig 2.2, in which global relation is split into four fragment R_1 ,R_2 , R_3 , and R_4 . These four fragments are allocated redundantly at three sites of the computer network and building images are R^1, R^2, R^3. Copy of the fragment is denoted by R_i^j , i.e., copy of the fragment R_i is located at site j.

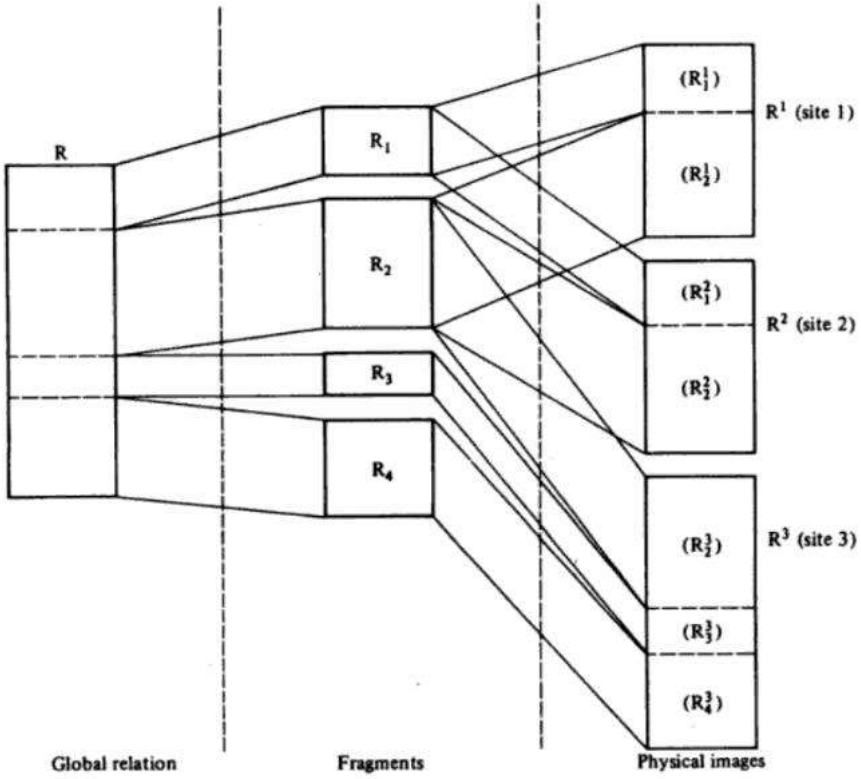


Fig. 2.2 Fragments of Physical Relation for a Global Relation

At lower level, it is necessary to map the physical images to the objects which are manipulated by the local DBMS. This mapping is called as local mapping schema and depends on the type of local DBMS.

Following three most important objectives are motivating the features of this architecture are data fragmentation and allocation, control of redundancy, and independency from local DBMS.

1. Separating the concept of data fragmentation from the concept of data allocation – this separation allows us to distinguish two different levels of distribution transparency, namely **fragmentation transparency** and **location transparency**. Fragmentation transparency is the highest degree of transparency for the fact that the user or application programs works on global relations. Location transparency is the lower degree of transparency for the fact of the user or applications to work on fragments instead of global relations.
2. **Explicit control of redundancy** The reference architecture provides the explicit control of redundancy at fragments level. For example, Fig. 2.2 shows that two physical images R^2 . and R^3 are overlapping. i.e., they contain common data.
3. **Independence from local DBMS** This feature called **local mapping transparency** , allows to study several problems of distributed database management without having to take account the specific models of local DBMSs.

2. 2 Types of Data Fragmentation

Decomposition of global relations into fragments can be performed by applying two different types of fragmentation: **horizontal fragmentation** and **vertical fragmentation.** A fragment can be expressed by a relational language. Following rules are applied during the fragmentation.

1. Completeness condition
2. Reconstruction condition
3. Disjoint condition

In completeness condition, all the data of global relation must be mapped into fragments

In reconstruction method, it must always possible to reconstruct the global relation from its fragments.

In disjoint condition, the fragments must be disjoint. So that the replication of the data can be controlled explicitly at the allocation level.

2.2.1 Horizontal Fragmentation

Horizontal fragmentation consists of partitioning the tuples of a global relation into subsets. It can be defined by the 'select' operation on the global relation

$$SUPPLIER(SNUM, NAME, CITY)$$

Then the horizontal fragmentation can be defined in the following way:

$$SUPPLIER_1 = \mathbf{SL}_{CITY = \text{"SF"}} SUPPLIER$$
$$SUPPLIER_2 = \mathbf{SL}_{CITY = \text{"LA"}} SUPPLIER$$

The above fragmentation satisfies the completeness condition if "SF" and "LA" are the only possible values of the *CITY* attribute; otherwise we would not know to which fragment the tuples with other *CITY* values belong.

The reconstruction condition is easily verified, because it is always possible to reconstruct the *SUPPLIER* global relation through the following operation:

$$SUPPLIER = SUPPLIER_1 \ \mathbf{UN} \ SUPPLIER_2$$

The disjoint property is also clearly verified.

We will call the predicate which is used in the selection operation which defines a fragment its **qualification.**

Example:

$q_1 : CITY = \text{"SF"}$
$q_2 : CITY = \text{"LA"}$

2.2.2 Derived Horizontal Fragmentation

In some cases, the horizontal fragmentation of a relation cannot be based on a property of its own attributes, but is derived from the horizontal fragmentation of another relation.

$$SUPPLY(SNUM, PNUM, DEPTNUM, QUAN)$$

where *SNUM* is a supplier number. It is meaningful to partition this relation so that a fragment contains the tuples for suppliers which are in a given city. However, city is not an attribute of the *SUPPLY* relation, it is an attribute of the *SUPPLIER* relation considered in the above example. Therefore we need a semi-join operation in order to determine the tuples of *SUPPLY* which correspond to the suppliers in a given city. The derived fragmentation of *SUPPLY* can be therefore defined as follows:

$$SUPPLY_1 = SUPPLY \, \mathbf{SJ}_{SNUM=SNUM} \, SUPPLIER_1$$
$$SUPPLY_2 = SUPPLY \, \mathbf{SJ}_{SNUM=SNUM} \, SUPPLIER_2$$

The effect of the semi-join operations is to select from *SUPPLY* the tuples which satisfy the join condition between $SUPPLIER_1$ or $SUPPLIER_2$ and *SUPPLY*, thus determining those tuples of *SUPPLY* which refer to suppliers in San Francisco or Los Angeles, respectively.

2.2.3 Vertical Fragmentation

The vertical fragmentation of a global relation is the sub-division of its attributes into groups; fragments are obtained by projecting the global relation over each group.

Example

$$EMP(EMPNUM, NAME, SAL, TAX, MGRNUM, DEPTNUM)$$

A vertical fragmentation of this relation can be defined as

$$EMP_1 = \mathbf{PJ}_{EMPNUM,NAME,MGRNUM,DEPTNUM} \, EMP$$
$$EMP_2 = \mathbf{PJ}_{EMPNUM,SAL,TAX} \, EMP$$

This fragmentation could, for instance, reflect an organization in which salaries and taxes are managed separately. The reconstruction of relation *EMP* can be obtained as

$$EMP = EMP_1 \, \mathbf{JN}_{EMPNUM=EMPNUM} \, EMP_2$$

because *EMPNUM* is a key of *EMP*. In general, the inclusion of a key of the global relation into each fragment is the most straightforward way to guarantee that the reconstruction through a join operation is possible. An alternative way to provide the reconstruction property is to generate **tuple identifiers** which are used as system-controlled keys. This can be convenient in order to avoid the replication of large keys; moreover, tuple identifiers cannot be modified by users.

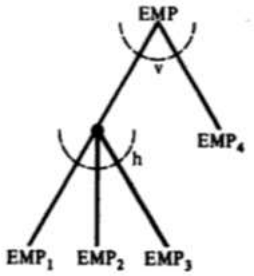
2.2.4 Mixed Fragmentation

We use both select and project operations as per defined horizontal and vertical fragmentations in mixed fragmentation. The reconstruction can be obtained by applying the reconstruction rules in inverse order.

EMP(EMPNUM, NAME, SAL, TAX, MGRNUM, DEPTNUM)

The following is a mixed fragmentation which is obtained by applying the vertical fragmentation of the previous example, followed by a horizontal fragmentation on *DEPTNUM*:

$$EMP_1 = \mathbf{SL}_{DEPTNUM \leq 10} \mathbf{PJ}_{EMPNUM,NAME,MGRNUM,DEPTNUM}\ EMP$$
$$EMP_2 = \mathbf{SL}_{10 < DEPTNUM \leq 20} \mathbf{PJ}_{EMPNUM,NAME,MGRNUM,DEPTNUM}\ EMP$$
$$EMP_3 = \mathbf{SL}_{DEPTNUM > 20} \mathbf{PJ}_{EMPNUM,NAME,MGRNUM,DEPTNUM}\ EMP$$
$$EMP_4 = \mathbf{PJ}_{EMPNUM,NAME,SAL,TAX}\ EMP$$



The reconstruction of relation *EMP* is defined by the following expression:

$$EMP = \mathbf{UN}\,(EMP_1, EMP_2, EMP_3)\mathbf{JN}_{EMPNUM=EMPNUM}\ \mathbf{PJ}_{EMPNUM,SAL,TAX}\ EMP_4$$

Mixed fragmentation can be conveniently represented by a **fragmentation tree**. In a fragmentation tree, the root corresponds to a global relation, the leaves correspond to the fragments, and the intermediate nodes correspond to the intermediate results of the fragment-defining expressions. The set of nodes which are sons of a given node represent the decomposition of this node by a fragmentation operation (vertical or horizontal). For example, Figure 3.3 shows the fragmentation tree of relation *EMP*. The root (relation *EMP*) is vertically fragmented into two portions: one portion corresponds to a leaf node of the tree (EMP_4); the other portion is horizontally partitioned, thus generating the other three leaves, corresponding to fragments EMP_1, EMP_2, and EMP_3.

The EXAMPLE_DDB

Figure 3.4 shows the global and fragmentation schemata of EXAMPLE_DDB which will be used in the rest of this book for the development of examples. Most of the global relations of EXAMPLE_DDB and their fragmentation have been already introduced. A *DEPT* relation, horizontally fragmented into three fragments on the value of the *DEPTNUM* attribute, is added. The features of EXAMPLE_DDB will be discussed when they will be used to exemplify specific topics. Note, as a

Global schema

EMP(EMPNUM, NAME, SAL, TAX, MGRNUM, DEPTNUM)

DEPT(DEPTNUM, NAME, AREA, MGRNUM)

SUPPLIER(SNUM, NAME, CITY)

SUPPLY(SNUM, PNUM, DEPTNUM, QUAN)

Fragmentation schema

$$EMP_1 = \mathbf{SL}_{DEPTNUM \leq 10} \mathbf{PJ}_{EMPNUM,NAME,MGRNUM,DEPTNUM}(EMP)$$

$$EMP_2 = \mathbf{SL}_{10 < DEPTNUM \leq 20} \mathbf{PJ}_{EMPNUM,NAME,MGRNUM,DEPTNUM}(EMP)$$

$$EMP_3 = \mathbf{SL}_{DEPTNUM > 20} \mathbf{PJ}_{EMPNUM,NAME,MGRNUM,DEPTNUM}(EMP)$$

$$EMP_4 = \mathbf{PJ}_{EMPNUM,NAME,SAL,TAX}(EMP)$$

$$DEPT_1 = \mathbf{SL}_{DEPTNUM \leq 10}(DEPT)$$

$$DEPT_2 = \mathbf{SL}_{10 < DEPTNUM \leq 20}(DEPT)$$

$$DEPT_3 = \mathbf{SL}_{DEPTNUM > 20}(DEPT)$$

$$SUPPLIER_1 = \mathbf{SL}_{CITY = \text{"SF"}}(SUPPLIER)$$

$$SUPPLIER_2 = \mathbf{SL}_{CITY = \text{"LA"}}(SUPPLIER)$$

$$SUPPLY_1 = SUPPLY \, \mathbf{SJ}_{SNUM = SNUM} SUPPLIER_1$$

$$SUPPLY_2 = SUPPLY \, \mathbf{SJ}_{SNUM = SNUM} SUPPLIER_2$$