

CSE 4125: Distributed Database Systems Chapter – 3

Levels of Distributed Transparency.
(part – A)

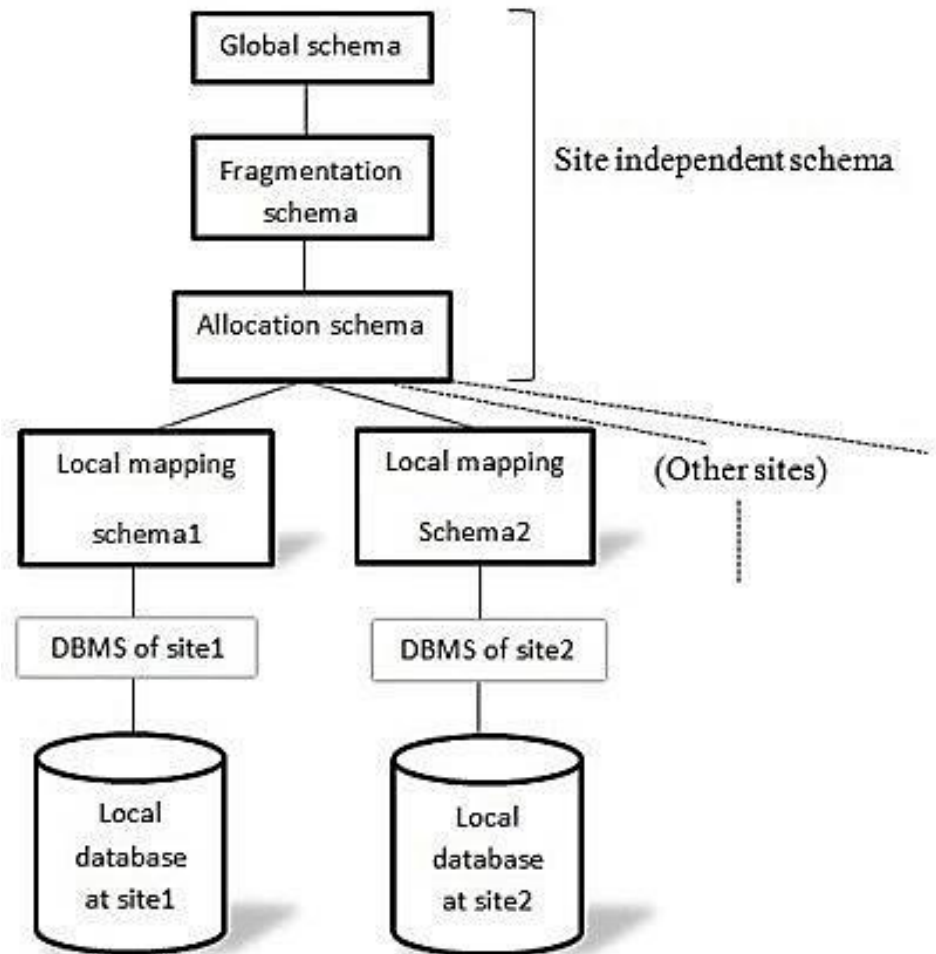
Outline

- The Reference Architecture of DDB.
- Introduction to Distribution Transparency.
- Different Levels of Distribution Transparency.

Reference Architecture for DDB

- Represents the organization of any DDB.
- Not explicitly implemented in all DDBs.
 - But conceptually relevant in order to understand the working mechanism of DDB.

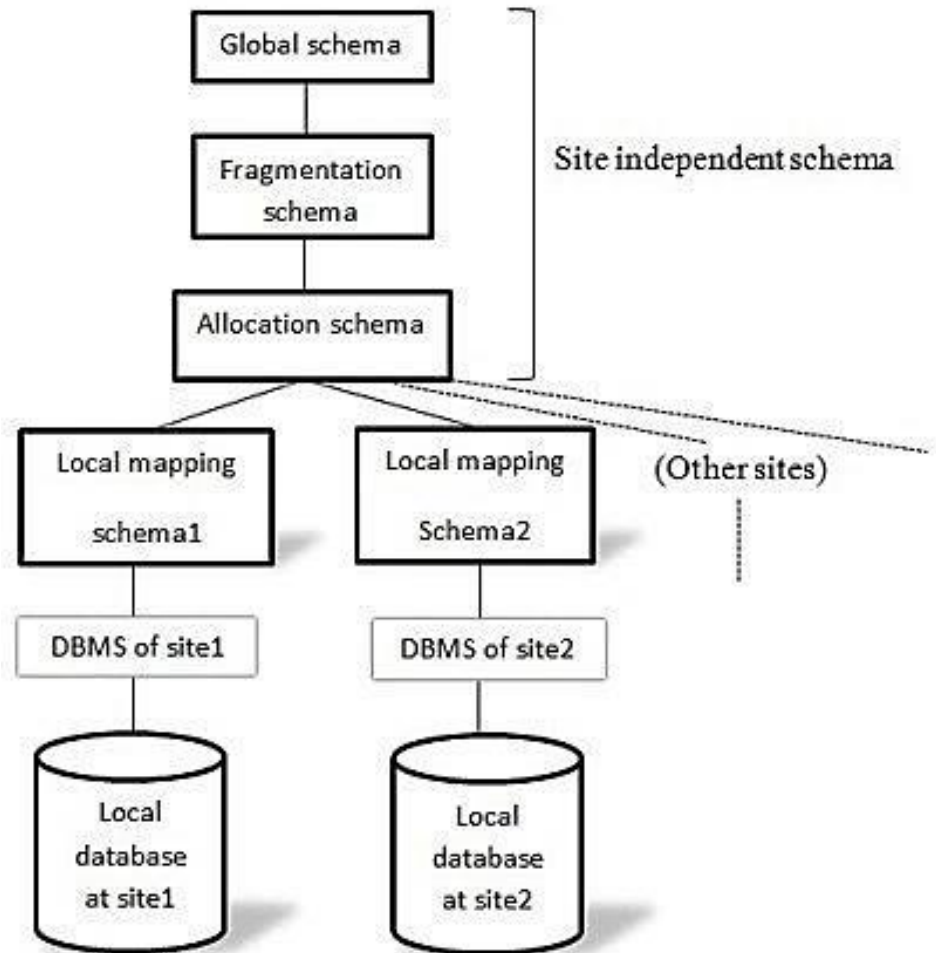
Reference Architecture for DDB (cont.)



Reference Architecture for DDB (cont.)

Components:

1. Global Schema.
2. Fragmentation Schema.
3. Allocation Schema.
4. Local Mapping Schema.

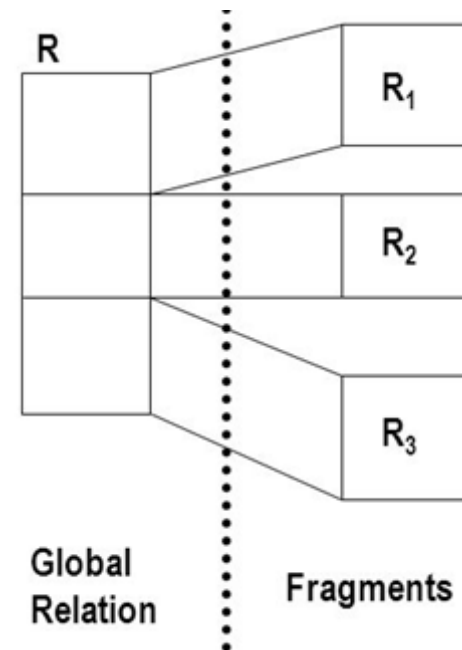


Global Schema

- Global schema defines all the data which are contained in the distributed database.
 - *Conceptual view** of the database.
 - As if the database were not distributed at all.
- Global schema defines a set of *global relations*.

Fragmentation Schema

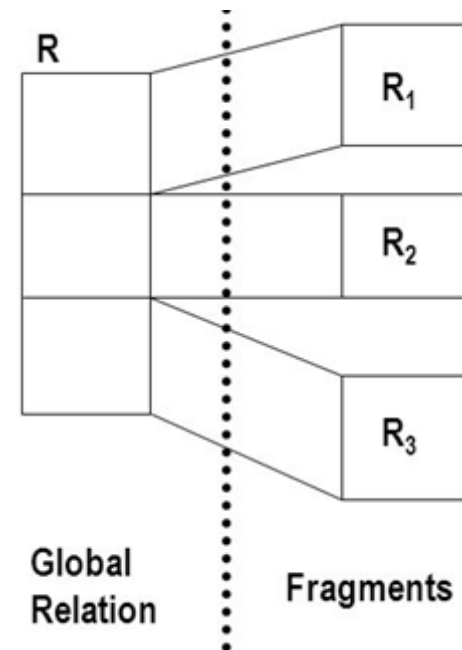
- Each global relation (R) can be split into several non-overlapping portions which are called *fragments*.
 - Logical portions of R .
- Example –
 R can be partitioned into R_1, R_2 and R_3



Fragmentation Schema (cont.)

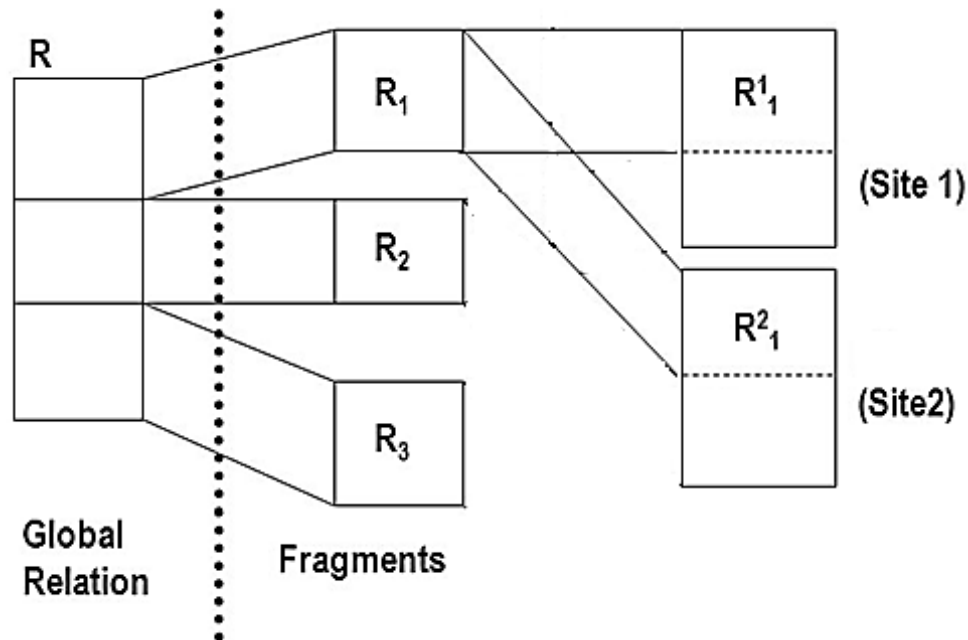
- The mapping between global relations and fragments is defined in the *fragmentation schema*.

R_i indicates i^{th} fragment of global relation R .



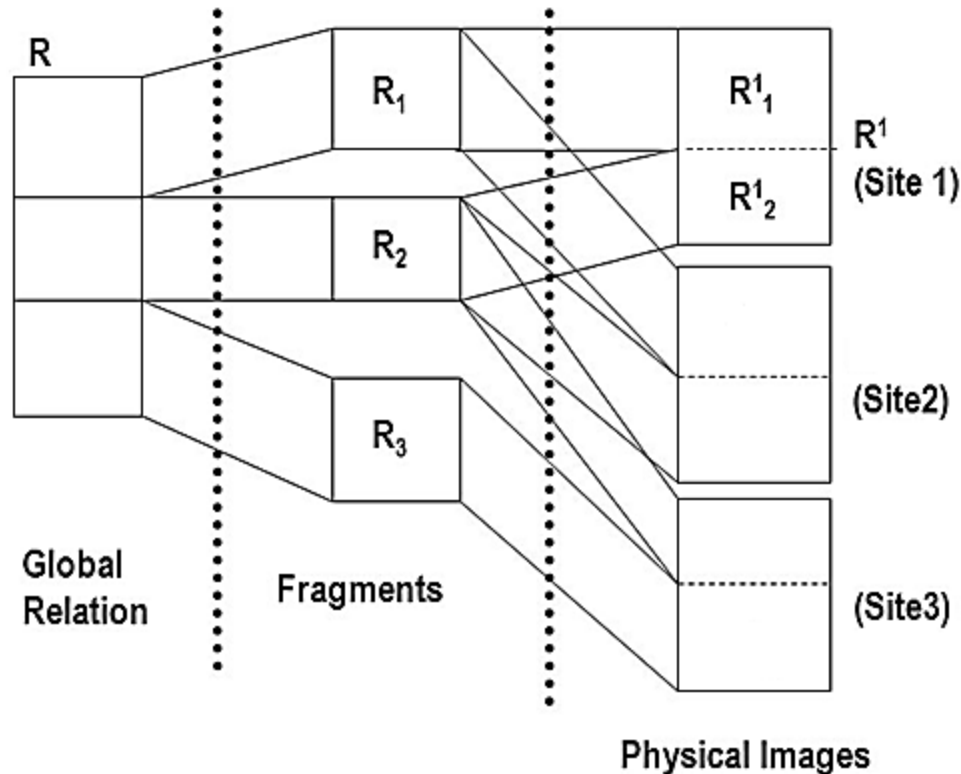
Allocation Schema

- Allocation schema defines at which site(s) a fragment is located.



Allocation Schema (cont.)

- Fragments from R creates *physical image* (R^j) of R at site j .



Local Mapping Schema

- Mapping physical images to database objects which are manipulated by local DBMS.
- Depends on the types of the local DBMS.
 - Example: if local DBMS is Oracle, the physical images must be mapped so that Oracle can understand

Advantage of Fragmentation*

- Usage:
 - In general, applications work with *views* (subset of relation) rather than entire relations.
 - In data distribution, it seems appropriate to work with subsets of relation as the unit of distribution.
- Efficiency:
 - Data is stored close to where it is most frequently used. In addition, data that is not needed by local applications is not stored.

Advantage of Fragmentation*

(cont.)

- Parallelism:
 - A transaction can be divided into several sub queries that operate on fragments. This should increase the degree of concurrency by allowing transactions to execute in parallel.
- Security:
 - Data not required by local applications is not stored, and consequently not available to unauthorized users.

Disadvantage of Fragmentation*

- Performance:
 - The performance of global application that requires data from several fragments located at different sites may be slower.
- Integrity:
 - Integrity control may be more difficult if data and functional dependencies are fragmented and located at different sites.

Distribution Transparency

- The property of DDB by which the internal details of the distribution are hidden from the users (i.e. application programmer).*
 - A transparent system “*hides*” the implementation details from users.
 - The advantage of a *fully transparent* DDB is the high level of support that it provides for the development of complex applications.

Distribution Transparency (cont.)

- Levels of Distribution Transparency:
 - Levels at which an application programmer view the DDB, depending on how much distribution transparency is provided by the DDBMS.
- Important levels are –
 - Level-1*: Fragmentation transparency.
 - Level-2*: Location transparency.
 - Level-3*: Local mapping transparency.

Distribution Transparency (cont.)

Level-1: Fragmentation Transparency:

- Programmer works on global relation.
- Fragmentation information is hidden.
 - Enables Programmer to query upon any relation as if it *were not fragmented*.

	Global relation	Fragmentation	Location	Local mapping
Availability to programmer	yes	n/a	n/a	n/a

Distribution Transparency (cont.)

Level-2: Location Transparency:

- Fragmentation information is provided. Programmer works on fragments.
- Location (i.e. site name) information is hidden.
 - Enables Programmer to query upon fragments as if they were stored *locally* in the user's site.

	Global relation	Fragmentation	Location	Local mapping
Availability to programmer	yes	yes	n/a	n/a

Distribution Transparency (cont.)

Level-3: Local Mapping Transparency:

- Location information is provided. Programmer works on fragmentation at specific location (site).
- Local DBMS information is hidden.
 - Enables Programmer to query upon fragments at a site as if the local DBMS is “known” (i.e. Oracle or MySQL) .

	Global relation	Fragmentation	Location	Local mapping
Availability to programmer	yes	yes	yes	n/a

Who Should Provide Transparency?*

Three layers:

- User language (code).
 - Compiler, interpreter.
- Operating system.
 - Distributed environment (Network management)
- Local DBMS.

Additional Reading

- Copy of a fragment.
- Replication transparency.

Questions

- a) Do you think two physical images can be identical? Give an example with diagram.
- b) What do you understand by the notation – $STUDENT_5^3$?
- c) According to you, what could be the possible difficulties in local mapping schema for a heterogeneous DDBMS?