

---

**DISTRIBUTED DATABASE DESIGN**

The design of a distributed database introduces three new issues:

- How to partition the database into fragments.
- Which fragments to replicate.
- Where to locate those fragments and replicas.

**DATA FRAGMENTATION**

Data fragmentation allows you to break a single object into two or more segments or fragments. The object might be a user's database, a system database, or a table. Each fragment can be stored at any site over a computer network.

Information about data fragmentation is stored in the distributed data catalog (DDC), from which it is accessed by the TP to process user requests.

There are three types of data fragmentation strategies:

- **Horizontal fragmentation** refers to the division of a relation into subsets (fragments) of tuples (rows). Each fragment is stored at a different node, and each fragment has unique rows. However, the unique rows all have the same attributes (columns). In short, each fragment represents the equivalent of a SELECT statement, with the WHERE clause on a single attribute.
- **Vertical fragmentation** refers to the division of a relation into attribute (column) subsets. Each subset (fragment) is stored at a different node, and each fragment has unique columns—with the exception of the key column, which is common to all fragments.
- **Mixed fragmentation** refers to a combination of horizontal and vertical strategies. In other words, a table may be divided into several horizontal subsets (rows), each one having a subset of the attributes (columns).

To illustrate the fragmentation strategies, let's use the CUSTOMER table for the XYZ Company, depicted in Figure below.

Table name: CUSTOMER

CUS_NUM	CUS_NAME	CUS_ADDRESS	CUS_STATE	CUS_LIMIT	CUS_BAL	CUS_RATING	CUS_DUE
10	Sinex, Inc.	12 Main St.	TN	3500.00	2700.00	3	1245.00
11	Martin Corp.	321 Sunset Blvd.	FL	6000.00	1200.00	1	0.00
12	Mynux Corp.	910 Eagle St.	TN	4000.00	3500.00	3	3400.00
13	BTBC, Inc.	Rue du Monde	FL	6000.00	5890.00	3	1090.00
14	Victory, Inc.	123 Maple St.	FL	1200.00	550.00	1	0.00
15	NBCC Corp.	909 High Ave.	GA	2000.00	350.00	2	50.00

### Horizontal Fragmentation

Suppose XYZ Company's corporate management requires information about its customers in all three states, but company locations in each state (TN, FL, and GA) require data regarding local customers only. Based on such requirements, you decide to distribute the data by state.

FRAGMENT NAME	LOCATION	CONDITION	NODE NAME	CUSTOMER NUMBERS	NUMBER OF ROWS
CUST_H1	Tennessee	CUS_STATE = 'TN'	NAS	10, 12	2
CUST_H2	Georgia	CUS_STATE = 'GA'	ATL	15	1
CUST_H3	Florida	CUS_STATE = 'FL'	TAM	11, 13, 14	3

Table name: CUST_H1		Location: Tennessee		Node: NAS			
CUS_NUM	CUS_NAME	CUS_ADDRESS	CUS_STATE	CUS_LIMIT	CUS_BAL	CUS_RATING	CUS_DUE
10	Sinex, Inc.	12 Main St.	TN	3500.00	2700.00	3	1245.00
12	Myriux Corp.	910 Eagle St.	TN	4000.00	3500.00	3	3400.00

Table name: CUST_H2		Location: Georgia		Node: ATL			
CUS_NUM	CUS_NAME	CUS_ADDRESS	CUS_STATE	CUS_LIMIT	CUS_BAL	CUS_RATING	CUS_DUE
15	NBCC Corp.	909 High Ave.	GA	2000.00	350.00	2	50.00

Table name: CUST_H3		Location: Florida		Node: TAM			
CUS_NUM	CUS_NAME	CUS_ADDRESS	CUS_STATE	CUS_LIMIT	CUS_BAL	CUS_RATING	CUS_DUE
11	Martin Corp.	321 Sunset Blvd.	FL	6000.00	1200.00	1	0.00
13	BTBC, Inc.	Rue du Monde	FL	6000.00	5890.00	3	1090.00
14	Victory, Inc.	123 Maple St.	FL	1200.00	550.00	1	0.00

### Vertical Fragmentation

suppose the company is divided into two departments: the service department and the collections department. Each department is located in a separate building, and each has an interest in only a few of the CUSTOMER table's attributes

FRAGMENT NAME	LOCATION	NODE NAME	ATTRIBUTE NAMES
CUST_V1	Service Bldg.	SVC	CUS_NUM, CUS_NAME, CUS_ADDRESS, CUS_STATE
CUST_V2	Collection Bldg.	ARC	CUS_NUM, CUS_LIMIT, CUS_BAL, CUS_RATING, CUS_DUE



Table name: CUST_V1				Location: Service Building		Node: SVC
CUS_NUM	CUS_NAME	CUS_ADDRESS	CUS_STATE			
10	Sinex, Inc.	12 Main St.	TN			
11	Martin Corp.	321 Sunset Blvd.	FL			
12	Mynux Corp.	910 Eagle St.	TN			
13	BTBC, Inc.	Rue du Monde	FL			
14	Victory, Inc.	123 Maple St.	FL			
15	NBCC Corp.	909 High Ave.	GA			

Table name: CUST_V2				Location: Collection Building		Node: ARC
CUS_NUM	CUS_LIMIT	CUS_BAL	CUS_RATING	CUS_DUE		
10	3500.00	2700.00	3	1245.00		
11	8000.00	1200.00	1	0.00		
12	4000.00	3500.00	3	3400.00		
13	6000.00	5890.00	3	1090.00		
14	1200.00	550.00	1	0.00		
15	2000.00	350.00	2	50.00		

### Mixed Fragmentation

Mixed fragmentation requires a two-step procedure. First, horizontal fragmentation is introduced for each site based on the location within a state (CUS\_STATE). The horizontal fragmentation yields the subsets of customer tuples (horizontal fragments) that are located at each site. Because the departments are located in different buildings, vertical fragmentation is used within each horizontal fragment to divide the attributes, thus meeting each department's information needs at each sub site. Mixed fragmentation yields the results displayed in Table below.

FRAGMENT NAME	LOCATION	HORIZONTAL CRITERIA	NODE NAME	RESULTING ROWS AT SITE	VERTICAL CRITERIA ATTRIBUTES AT EACH FRAGMENT
CUST_M1	TN-Service	CUS_STATE = 'TN'	NAS-S	10, 12	CUS_NUM, CUS_NAME CUS_ADDRESS, CUS_STATE
CUST_M2	TN-Collection	CUS_STATE = 'TN'	NAS-C	10, 12	CUS_NUM, CUS_LIMIT, CUS_BAL, CUS_RATING, CUS_DUE
CUST_M3	GA-Service	CUS_STATE = 'GA'	ATL-S	15	CUS_NUM, CUS_NAME CUS_ADDRESS, CUS_STATE
CUST_M4	GA-Collection	CUS_STATE = 'GA'	ATL-C	15	CUS_NUM, CUS_LIMIT, CUS_BAL, CUS_RATING, CUS_DUE
CUST_M5	FL-Service	CUS_STATE = 'FL'	TAM-S	11, 13, 14	CUS_NUM, CUS_NAME CUS_ADDRESS, CUS_STATE
CUST_M6	FL-Collection	CUS_STATE = 'FL'	TAM-C	11, 13, 14	CUS_NUM, CUS_LIMIT, CUS_BAL, CUS_RATING, CUS_DUE

Table name: CUST_M1				Location: TN-Service		Node: NAS-S
CUS_NUM	CUS_NAME	CUS_ADDRESS	CUS_STATE			
10	Sinex, Inc.	12 Main St.	TN			
12	Mynux Corp.	910 Eagle St.	TN			

Table name: CUST_M2				Location: TN-Collection		Node: NAS-C
CUS_NUM	CUS_LIMIT	CUS_BAL	CUS_RATING	CUS_DUE		
10	3500.00	2700.00	3	1245.00		
12	4000.00	3500.00	3	3400.00		

Table name: CUST_M3				Location: GA-Service		Node: ATL-S
CUS_NUM	CUS_NAME	CUS_ADDRESS	CUS_STATE			
15	NBCC Corp.	909 High Ave.	GA			

Table name: CUST_M4				Location: GA-Collection		Node: ATL-C
CUS_NUM	CUS_LIMIT	CUS_BAL	CUS_RATING	CUS_DUE		
15	2000.00	350.00	2	50.00		

Table name: CUST_M5				Location: FL-Service		Node: TAM-S
CUS_NUM	CUS_NAME	CUS_ADDRESS	CUS_STATE			
11	Martin Corp.	321 Sunset Blvd.	FL			
13	BTBC, Inc.	Rue du Monde	FL			
14	Victory, Inc.	123 Maple St.	FL			

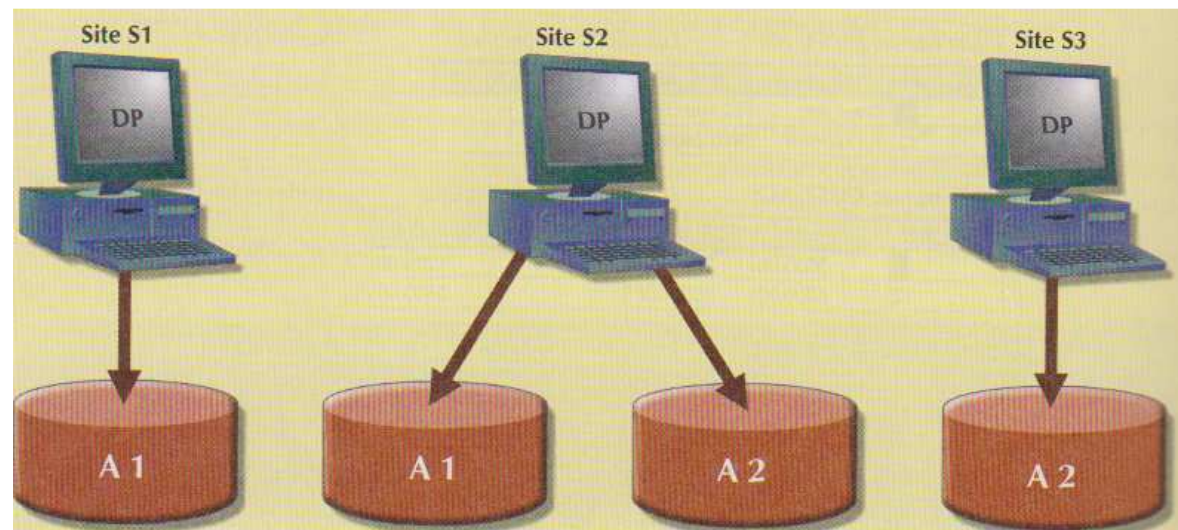
  

Table name: CUST_M6				Location: FL-Collection		Node: TAM-C
CUS_NUM	CUS_LIMIT	CUS_BAL	CUS_RATING	CUS_DUE		
11	6000.00	1200.00	1	0.00		
13	6000.00	5890.00	3	1090.00		
14	1200.00	550.00	1	0.00		

## DATA REPLICATION

Data replication refers to the storage of data copies at multiple sites served by a computer network. Fragment copies can be stored at several sites to serve specific information requirements.





Replicated data are subject to the mutual consistency rule. The **mutual consistency rule** requires that all copies of data fragments be identical. Therefore, to maintain data consistency among the replicas, the DDBMS must ensure that a database update is performed at all sites where replicas exist.

- replication has some benefits (such as improved data availability, better load distribution, improved data failure-tolerance, and reduced query costs).
- it also imposes additional DDBMS processing overhead— because each data copy must be maintained by the system. Furthermore, because the data are replicated at another site, there are associated storage costs and increased transaction times (as data must be updated at several sites concurrently to comply with the mutual consistency rule).
- To illustrate the replica overhead imposed on a DDBMS. Consider the processes that DDBMS
  - If the database is fragmented, the DDBMS must decompose a query into sub queries to access the appropriate fragments.
  - If the database is replicated, the DDBMS must decide which copy to access. A READ operation selects the nearest copy to satisfy the transaction. A WRITE operation requires that all copies be selected and to satisfy the mutual consistency rule.
  - The TP sends a data request to each selected DP for execution.
  - The DP receives and executes each request and sends the data back to the TP.
  - The TP assembles the DP responses

Three replication scenarios exist :

- A **fully replicated** database stores multiple copies of each database fragment at multiple sites. In this case , all database fragments are replicated. A fully replicated database can be impractical due to the overhead it imposes on the system.
- A **partially replicated** database stores multiple copies of some database fragments at multiple sites. Most DDBMSs are able to handle the partially replicated database well.
- An **unreplicated** database stores each database fragment at a single site. Therefore,

=====

there are no database fragments.

**Several factors influence the decision to use data replication:**

- Database size. The amount of data replicated will have an impact on the storage requirements and a r data transmission costs. Replicating large amounts of data requires a window of time and high I bandwidth that could affect other applications.
- Usage frequency. The frequency of data usage determines how frequently the data needs to be updated. Frequently used data needs to be updated more often, for example, than large data sets that are used only every quarter.
- Costs, including those for performance, software overhead, and management associated with synchronizing transactions and their components vs. fault-tolerance benefits that are associated with replicated data.
- Data replication information is stored in the distributed data catalog (DDC), Whose contents are used by the TP to decide which copy of a database fragment to access.

-

**DATA ALLOCATION**

Data allocation describes the process of deciding where to locate data. Data allocation strategies are as follows:

- With **centralized data allocation**, the entire database is stored at one site.
- With **partitioned data allocation**, the database is divided into two or more disjointed parts (fragments) and stored at two or more sites.
- With **replicated data allocation**, copies of one or more database fragments are stored at several sites.