Chapter 12

Distributed Database Management Systems

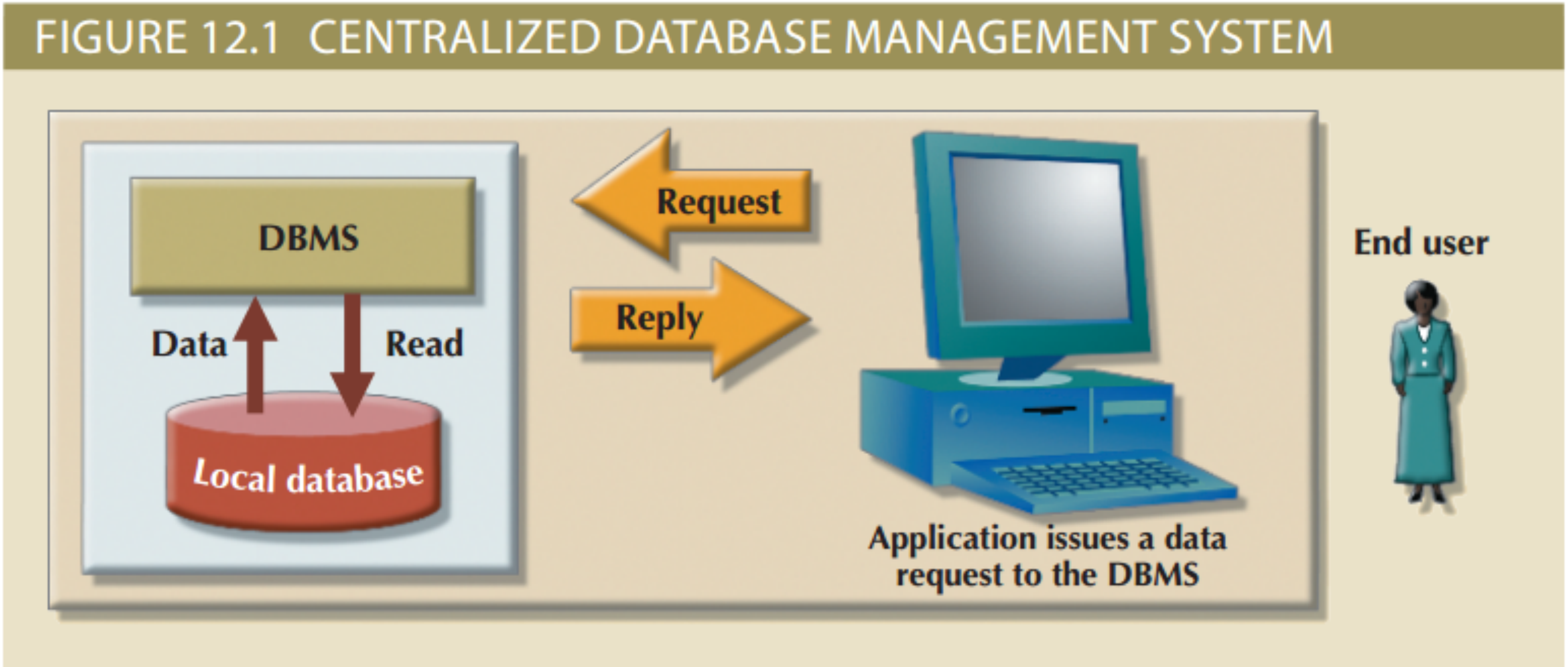
Learning Objectives

- In this chapter, you will learn:
  - About distributed database management systems (DDBMSs) and their components
  - How database implementation is affected by different levels of data and process distribution
  - How transactions are managed in a distributed database environment
  - How distributed database design draws on data partitioning and replication to balance performance, scalability, and availability
  - About the trade-offs of implementing a distributed data system

Evolution of Database Management Systems

- **Distributed database management system (DDBMS)** governs storage and processing of logically related data over interconnected computer systems
  - Data and processing functions are distributed among several sites
- Centralized database management system
  - Required that corporate data be stored in a single central site
  - Data access provided through dumb terminals

Figure 12.1 - Centralized Database Management System



Changes that Affected the Nature of Systems

- Globalization of business operation
- Market needs for an on-demand transaction style, based on web-based services
- Rapid social and technological changes fueled by low-cost smart mobile devices
- Data realms converging in the digital world
- Advent of social media to reach new customers and markets

Database Requirements in a Dynamic Business Environment

- Rapid ad hoc data access
  - Crucial in the quick-response decision-making environment
- Distributed data access
  - Needed to support geographically dispersed business units

Factors that Influenced DDBMS

- Acceptance of Internet as a platform for business
- Mobile wireless revolution
- Growth of use of “application as a service”
- Focus on mobile business intelligence
- Emphasis on Big Data analytics

Potential Centralized DBMS Problems

- Performance degradation
- High costs
- Reliability problems
- Scalability problems
- Organizational rigidity

Table 12.1 – Distributed DDBM Advantages and Disadvantages

TABLE 12.1

DISTRIBUTED DBMS ADVANTAGES AND DISADVANTAGES

ADVANTAGES	DISADVANTAGES
Data is located near the site of greatest demand. The data in a distributed database system is dispersed to match business requirements.	***Complexity of management and control.*** Applications must recognize data location, and they must be able to stitch together data from various sites. Database administrators must have the ability to coordinate database activities to prevent database degradation due to data anomalies.
Faster data access. End users often work with only the nearest stored subset of the data.	***Technological difficulty.*** Data integrity, transaction management, concurrency control, security, backup, recovery, and query optimization must all be addressed and resolved.
Faster data processing. A distributed database system spreads out the system's workload by processing data at several sites.	***Security.*** The probability of security lapses increases when data is located at multiple sites. The responsibility of data management will be shared by different people at several sites.
Growth facilitation. New sites can be added to the network without affecting the operations of other sites.	***Lack of standards.*** There are no standard communication protocols at the database level. For example, different database vendors employ different and often incompatible techniques to manage the distribution of data and processing in a DDBMS environment.
Improved communications. Because local sites are smaller and located closer to customers, local sites foster better communication among departments and between customers and company staff.	***Increased storage and infrastructure requirements.*** Multiple copies of data are required at different sites, thus requiring additional storage space.
Reduced operating costs. It is more cost-effective to add nodes to a network than to update a mainframe system. Development work is done more cheaply and quickly on low-cost PCs than on mainframes.	***Increased training cost.*** Training costs are generally higher in a distributed model than they would be in a centralized model, sometimes even to the extent of offsetting operational and hardware savings.
User-friendly interface. PCs and workstations are usually equipped with an easy-to-use graphical user interface (GUI). The GUI simplifies training and use for end users.	***Costs.*** Distributed databases require duplicated infrastructure to operate, such as physical location, environment, personnel, software, and licensing.
Less danger of a single-point failure. When one of the computers fails, the workload is picked up by other workstations. Data is also distributed at multiple sites.	
Processor independence. The end user can access any available copy of the data, and an end user's request is processed by any processor at the data location.	

Distributed Processing and Distributed Databases

- **Distributed processing**: Database's logical processing is shared among two or more physically independent sites via network
- **Distributed database**: Stores logically related database over two or more physically independent sites via computer network
  - **Database fragments**: Database composed of many parts in distributed database system

Figure 12.2 – Distributed Processing Environment

FIGURE 12.2 DISTRIBUTED PROCESSING ENVIRONMENT

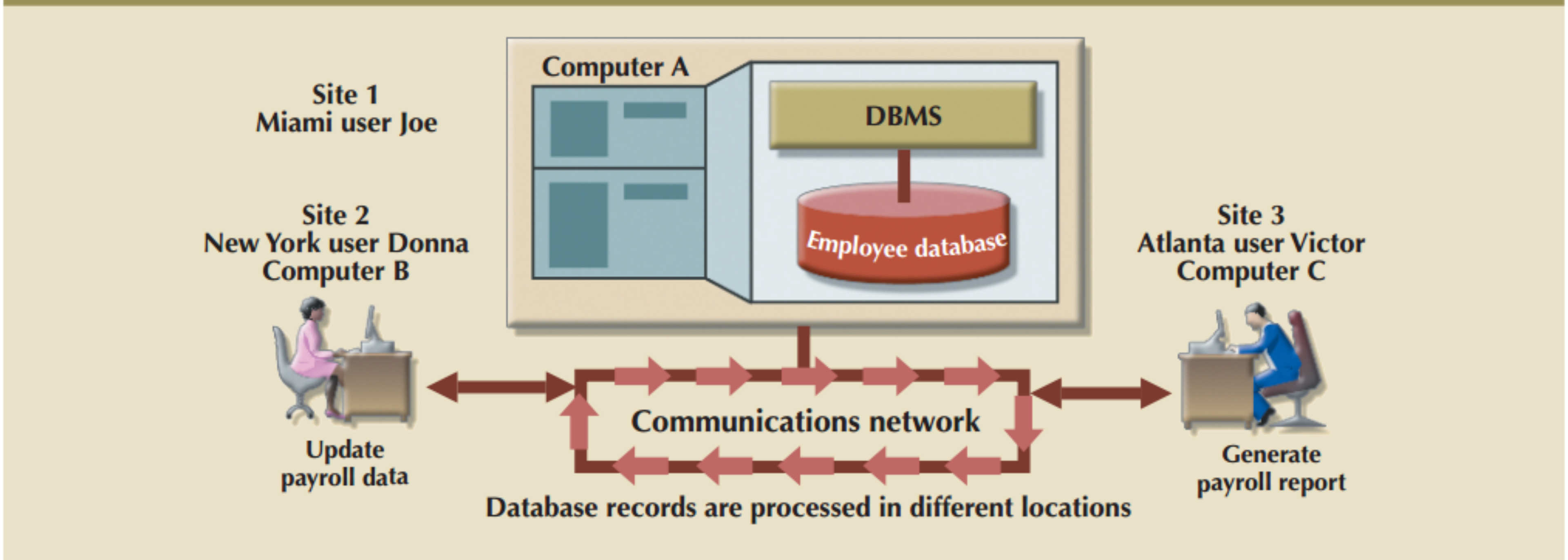
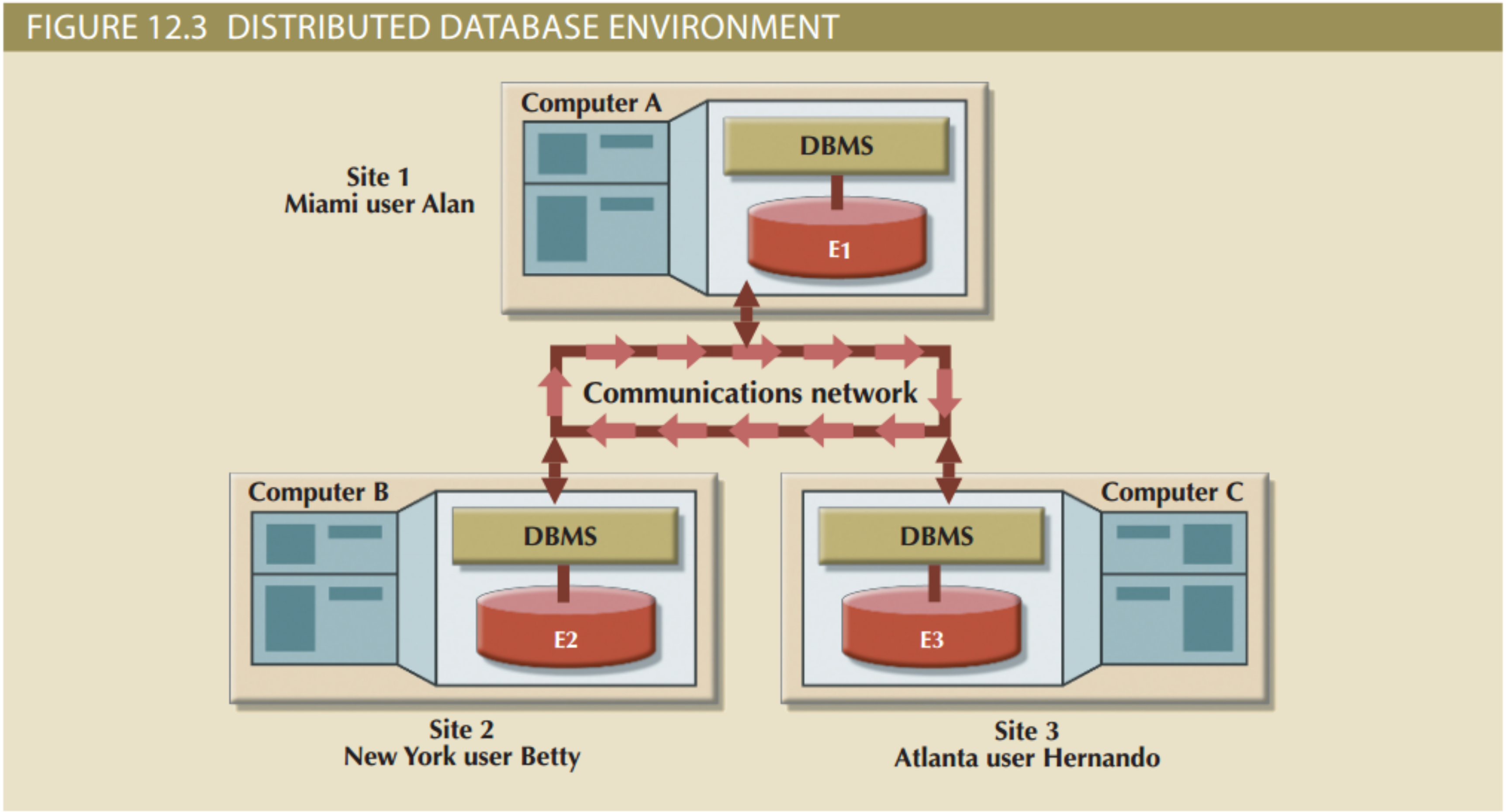


Figure 12.3 – Distributed Database Environment



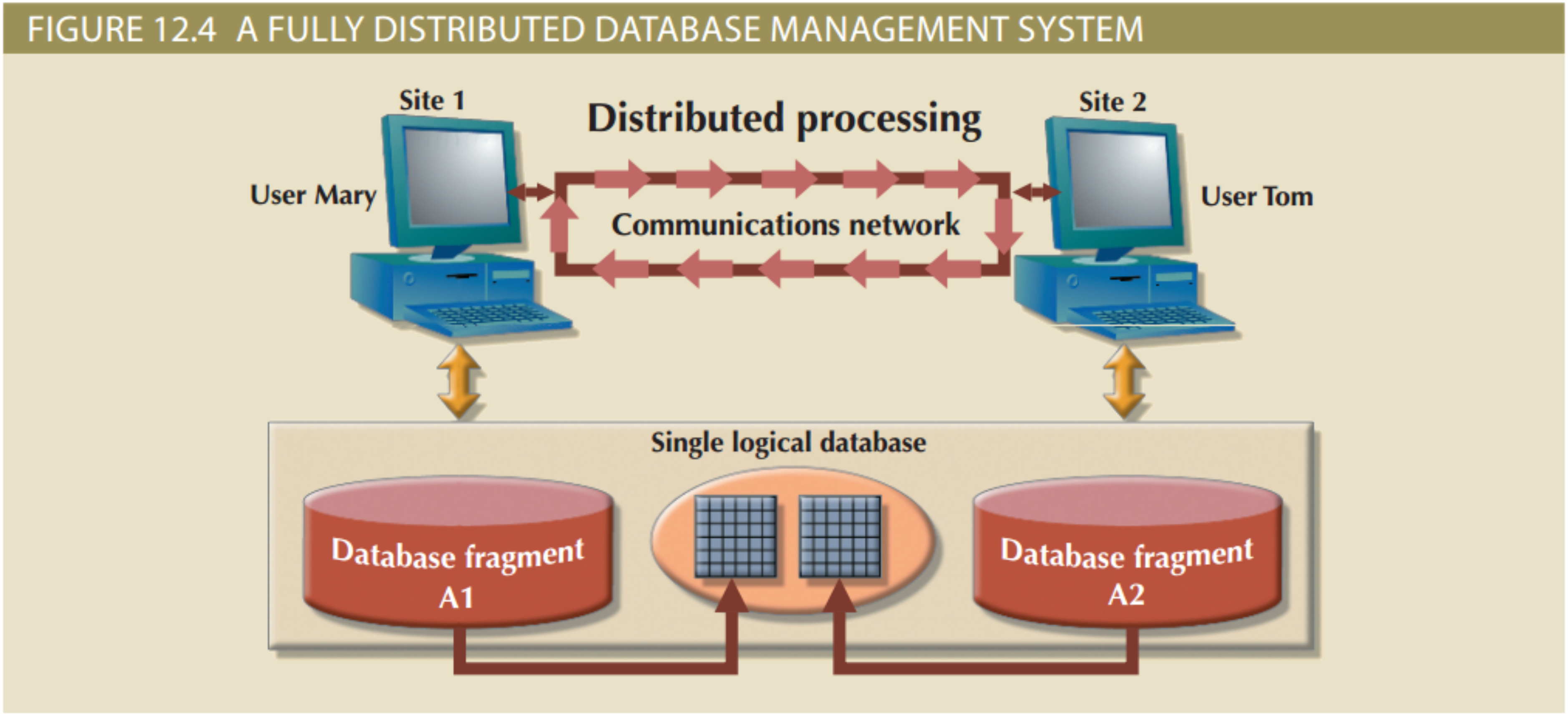
Characteristics of Distributed Management Systems

Application interface	Validation	Transformation	Query optimization
Mapping	I/O interface	Formatting	Security
Backup and recovery	DB administration	Concurrency control	Transaction management

Functions of Fully Distributed DBMS

- Receive the request of an application or end user
- Validate, analyze, and decompose the request
- Map request's logical-to-physical data components
- Decompose request into several I/O operations
- Search, locate, read and validate data
- Ensure database consistency, security, and integrity
- Validate data for conditions specified by request
- Present data in required format
- Handle all necessary functions transparently to user

Figure 12.4 - A Fully Distributed Database Management System

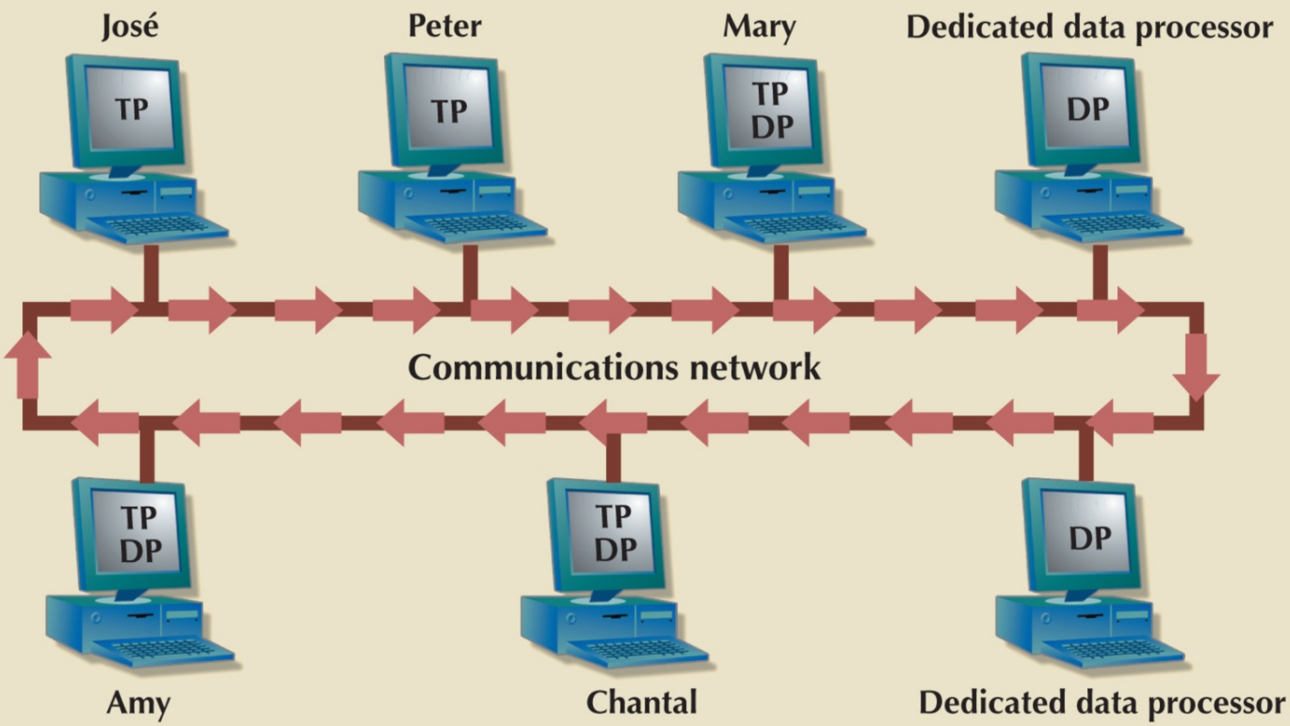


DDBMS Components

- Computer workstations or remote devices
- Network hardware and software components
- Communications media
- **Transaction processor (TP):** Software component in each computer or device that requests data
  - Known as **transaction manager (TM)** or **application processor (AP)**
- **Data processor (DP)** or **data manager (DM)**
  - Software component on each computer or device that stores and retrieves data from located at the site

Figure 12.5 - Distributed Database System Components

FIGURE 12.5 DISTRIBUTED DATABASE SYSTEM COMPONENTS

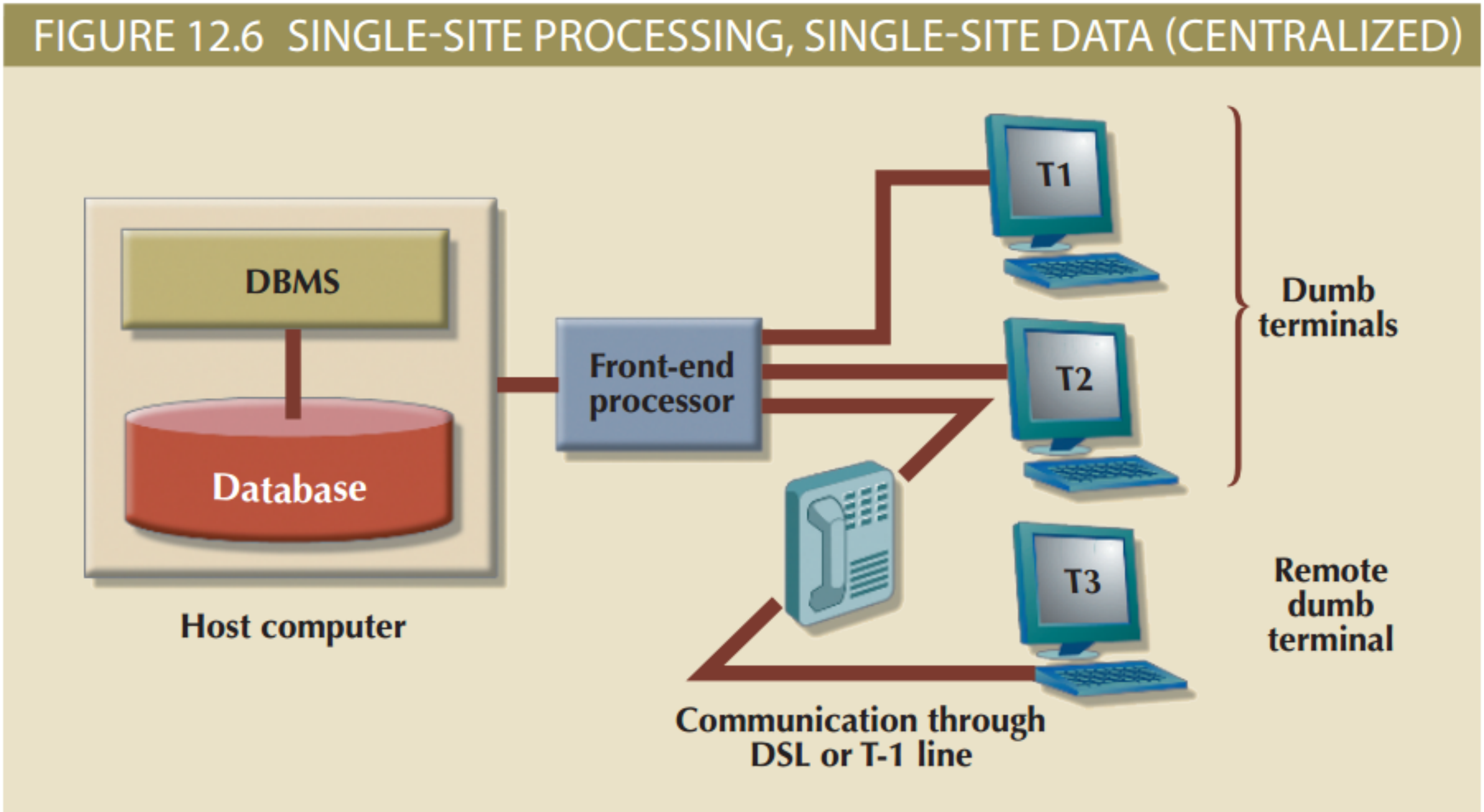


Note: Each TP can access data on any DP, and each DP handles all requests for local data from any TP.

Single-Site Processing, Single-Site Data (SPSD)

- Processing is done on a single host computer
- Data stored on host computer's local disk
- Processing restricted on end user's side
- DBMS is accessed by dumb terminals

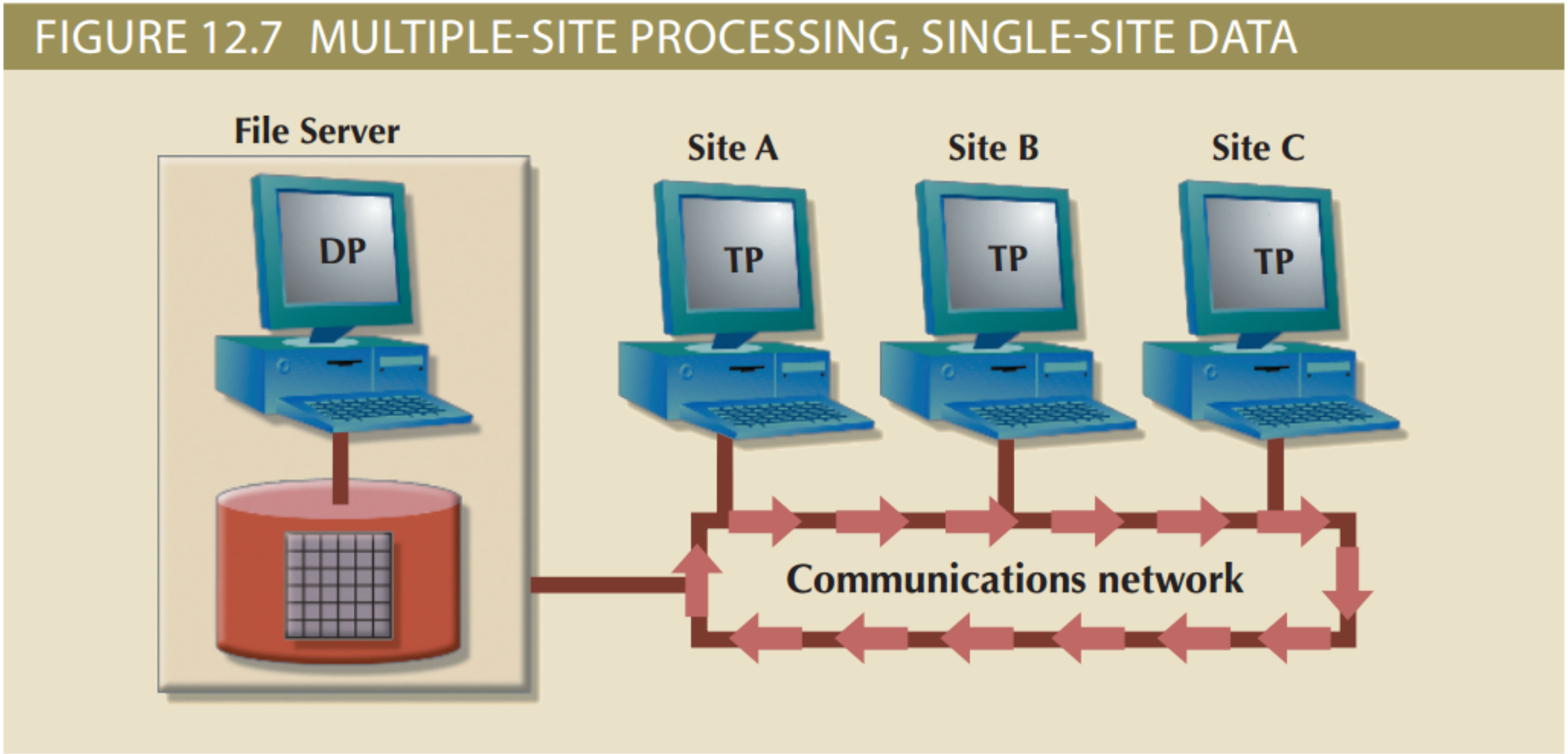
Figure 12.6 - Single-Site Processing, Single-Site Data (Centralized)



Multiple-Site Processing, Single-Site Data (MPSD)

- Multiple processes run on different computers sharing a single data repository
- Require network file server running conventional applications
  - Accessed through LAN
- **Client/server architecture**
  - Reduces network traffic
  - Processing is distributed
  - Supports data at multiple sites

Figure 12.7 - Multiple-Site Processing, Single-Site Data



Multiple-Site Processing, Multiple-Site Data (MPMD)

- Fully distributed database management system
  - Support multiple data processors and transaction processors at multiple sites
- Classifications:
  - **Homogeneous**: Integrate multiple instances of same DBMS over a network
  - **Heterogeneous**: Integrate different types of DBMSs over a network
  - **Fully heterogeneous**: Support different DBMSs, each supporting different data model running under different computer systems

Restrictions of DDBMS

- Remote access is provided on a read-only basis
- Restrictions on the number of remote tables that may be accessed in a single transaction
- Restrictions on the number of distinct databases that may be accessed
- Restrictions on the database model that may be accessed

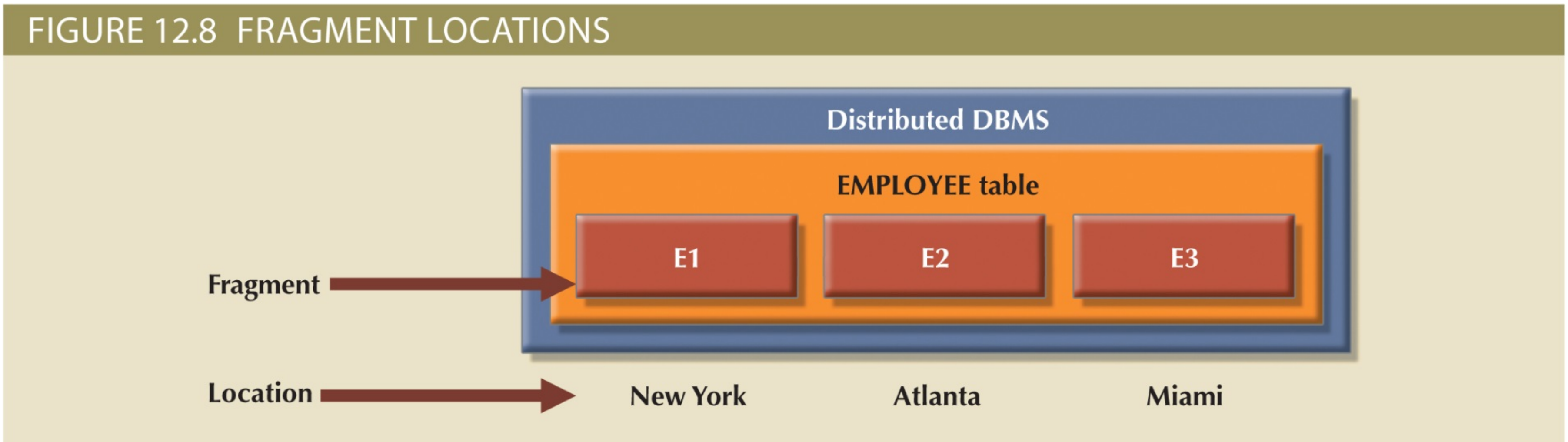
Distributed Database Transparency Features

- Distribution transparency
- Transaction transparency
- Failure transparency
- Performance transparency
- Heterogeneity transparency

Distribution Transparency

- Allows management of physically dispersed database as if centralized
  - Levels: **Fragmentation, location and local mapping**
- **Unique fragment**: Each row is unique, regardless of the fragment in which it is located
- Supported by **distributed data dictionary (DDD)** or **distributed data catalog (DDC)**
  - DDC contains the description of the entire database as seen by the database administrator
- **Distributed global schema**: Common database schema to translate user requests into subqueries

Figure 12.8 – Fragment Locations



Distributed Database Transparency Features

- Distribution transparency
- Transaction transparency
- Failure transparency
- Performance transparency
- Heterogeneity transparency

Transaction Transparency

- Ensures database transactions will maintain distributed database's integrity and consistency
- Ensures transaction completed only when all database sites involved complete their part
- Distributed database systems require complex mechanisms to manage transactions

Distributed Requests and Distributed Transactions

Remote request

- Single SQL statement accesses data processed by a single remote database processor

Remote transaction

- Accesses data at single remote site composed of several requests

Distributed transaction

- Requests data from several different remote sites on network

Distributed request

- Single SQL statement references data at several DP sites

Figure 12.9 – Remote Request

FIGURE 12.9 A REMOTE REQUEST

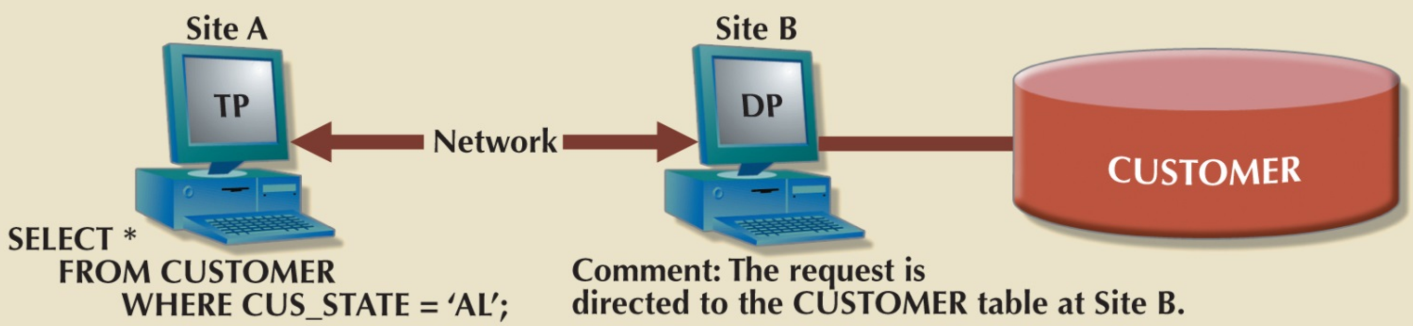
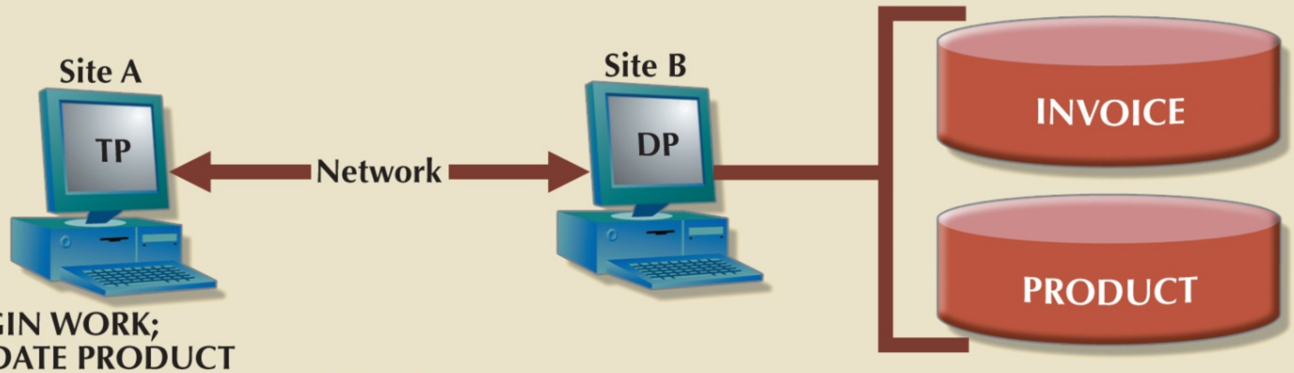


Figure 12.10 – Remote Transaction

FIGURE 12.10 A REMOTE TRANSACTION




```
BEGIN WORK;
UPDATE PRODUCT
      SET PROD_QTY = PROD_QTY – 1
          WHERE PROD_NUM = ‘231785’;
INSERT INTO INVOICE (CUS_NUM, INV_DATE, INV_TOTAL)
       VALUES ‘100’, ‘15-FEB-2016’, 120.00;
COMMIT WORK;
```

Figure 12.12 – Distributed Request

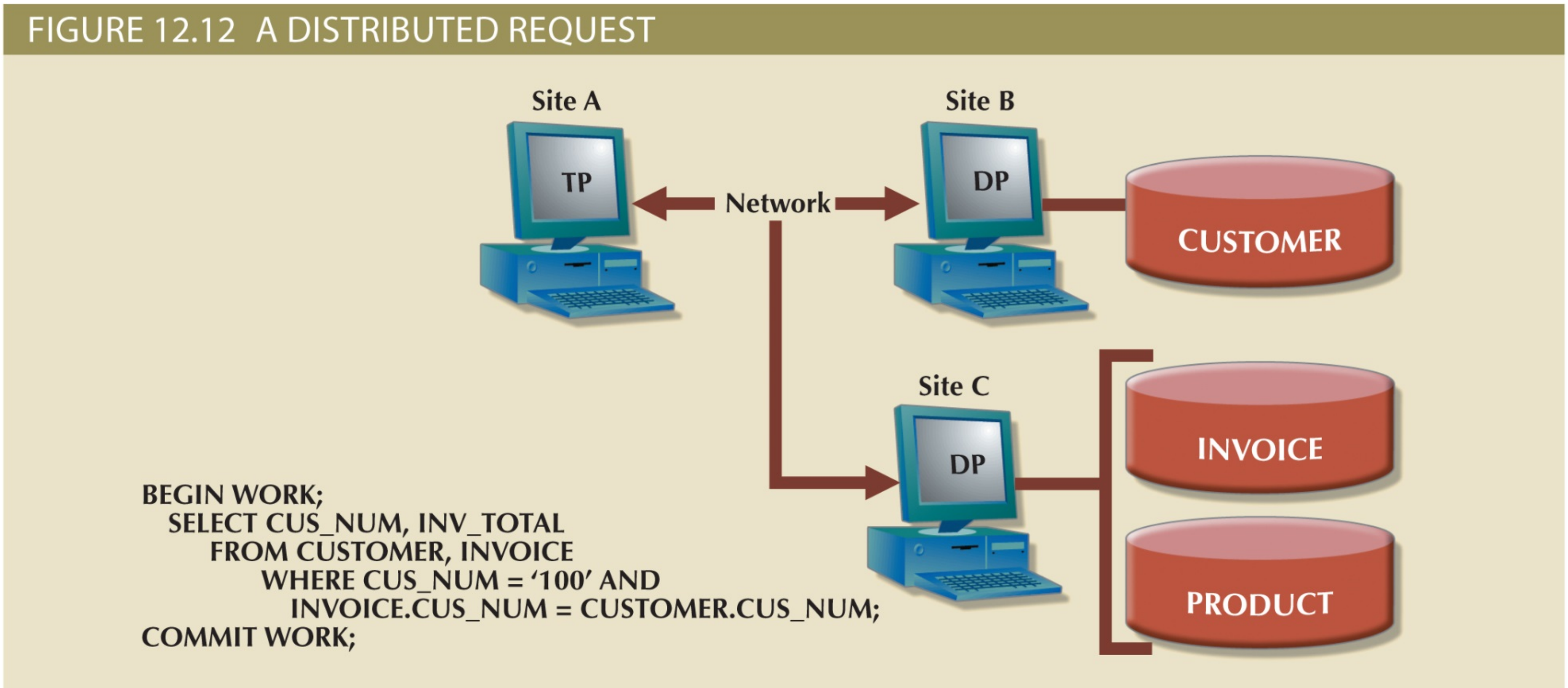


Figure 12.13 – Another Distributed Request

FIGURE 12.13 ANOTHER DISTRIBUTED REQUEST

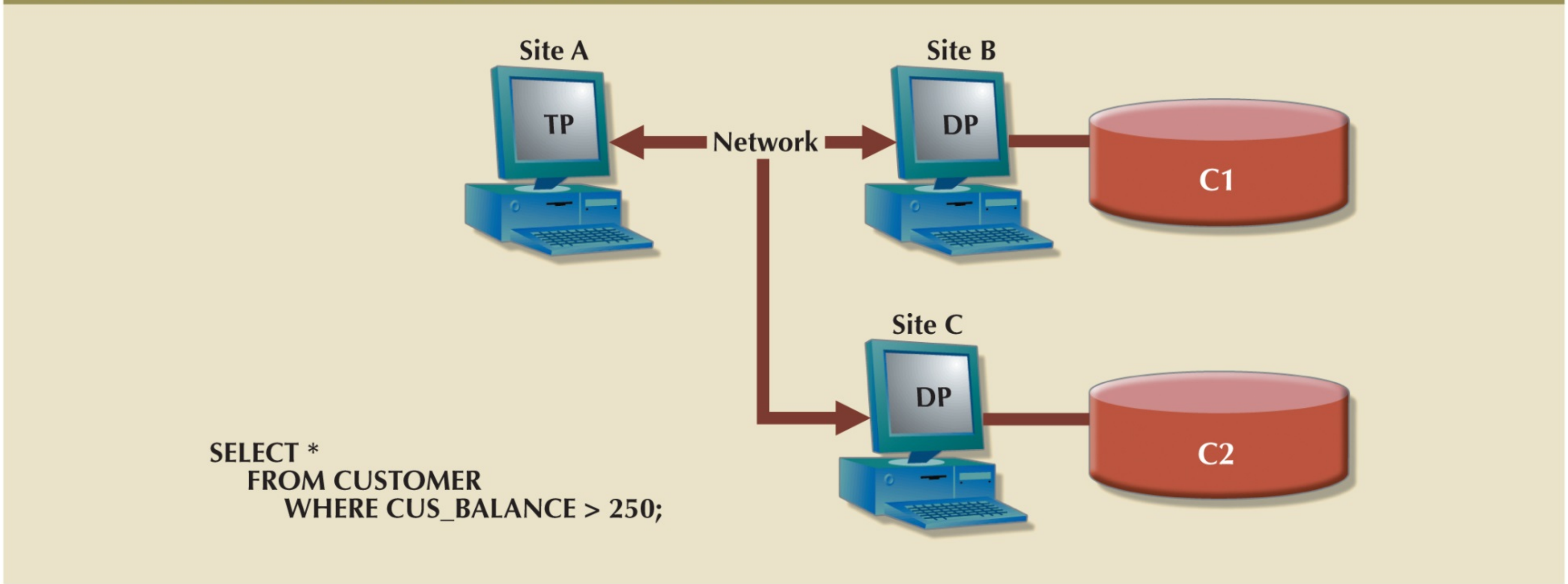
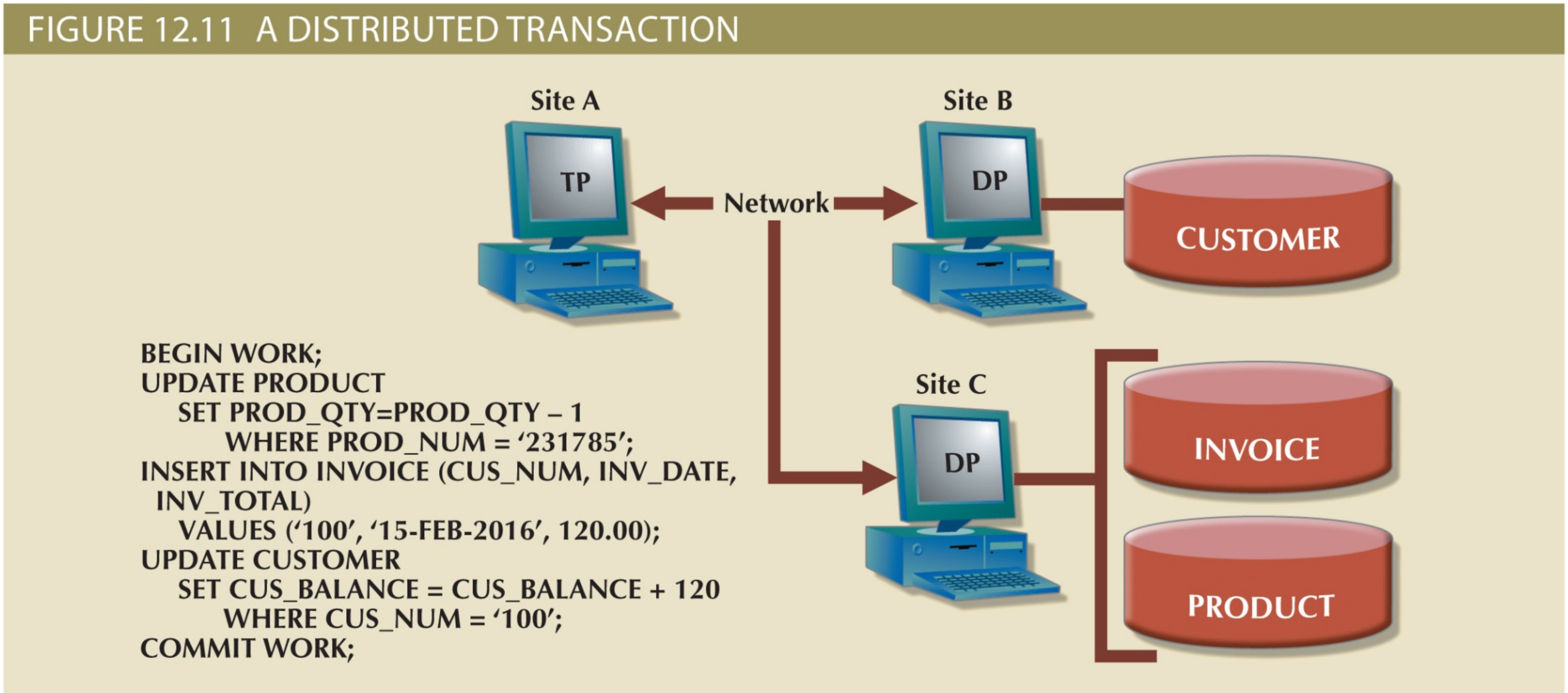


Figure 12.11 – Distributed Transaction

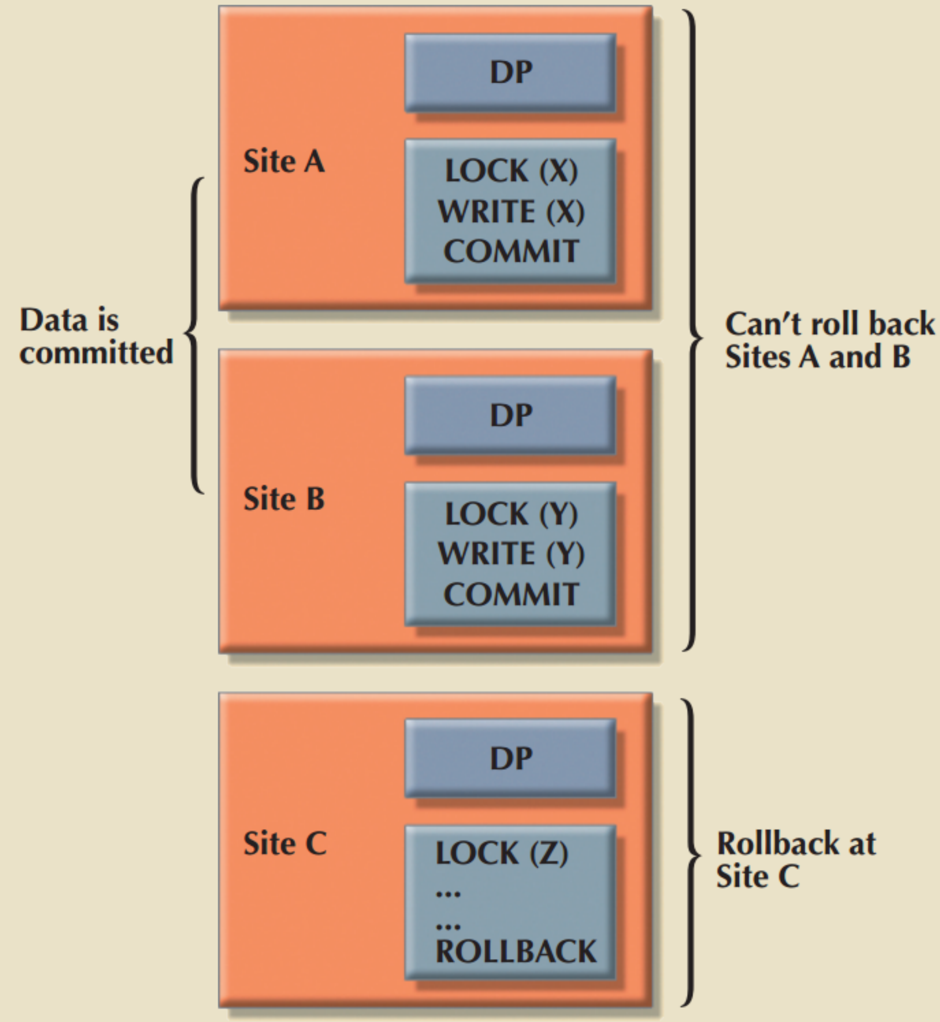


Distributed Concurrency Control

- Concurrency control is especially important in distributed databases environment
  - Multi-site, multiple-process operations are more likely to create inconsistencies and deadlocked transactions
- Solution to inconsistent database is a two-phase commit protocol

Figure 12.14 - The Effect of Premature COMMIT

FIGURE 12.14 THE EFFECT OF A PREMATURE COMMIT



Two-Phase Commit Protocol (2PC)

- Guarantees if a portion of a transaction operation cannot be committed, all changes made at the other sites will be undone
  - To maintain a consistent database state
- Requires that each DP's transaction log entry be written before database fragment is updated
- **DO-UNDO-REDO protocol**: Roll transactions back and forward with the help of the system's transaction log entries

Two-Phase Commit Protocol (2PC)

- **Write-ahead protocol**: Forces the log entry to be written to permanent storage before actual operation takes place
- Defines operations between **coordinator** and **subordinates**
- Phases of implementation
  - Preparation
  - The final COMMIT

Distributed Database Transparency Features

- Distribution transparency
- Transaction transparency
- Failure transparency
- Performance transparency
- Heterogeneity transparency

Performance and Failure Transparency

- Performance transparency allows a DDBMS to perform as if it were a centralized database
- Failure transparency ensures the system will operate in case of network failure
- Objective of query optimization is to minimize total costs which are a function of:
  - Access time (I/O) cost involved in accessing data from multiple remote sites
  - Communication costs associated with data transmission
  - CPU time cost associated with the processing overhead

Performance and Failure Transparency

- Considerations for resolving data requests in a distributed data environment:
  - Data distribution and data replication
    - **Replica transparency**: DDBMS's ability to hide multiple copies of data from the user
  - Network and node availability
    - **Network latency**: Delay imposed by the amount of time required for a data packet to make a round trip
    - **Network partitioning**: Delay imposed when nodes become suddenly unavailable due to a network failure

Distributed Database Design

Data fragmentation

- How to partition database into fragments

Data replication

- Which fragments to replicate

Data allocation

- Where to locate those fragments and replicas

Data Fragmentation

- Breaks a single object into two or more segments
  - Information is stored in distributed data catalog (DDC)
- Strategies
  - **Horizontal fragmentation**: Division of a relation into subsets (fragments) of tuples (rows)
  - **Vertical fragmentation**: Division of a relation into attribute (column) subsets
  - **Mixed fragmentation**: Combination of horizontal and vertical strategies

Figure 12.15 – Customer Table

FIGURE 12.15 A SAMPLE CUSTOMER TABLE

Table name: CUSTOMER **Database name: Ch12_Text**

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

Figure 12.16 – Fragments in Three Locations

FIGURE 12.16 TABLE FRAGMENTS IN THREE LOCATIONS

Database name: Ch12_Text

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	Mynux 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

Figure 12.15 – Vertically Fragmented Table Contents

FIGURE 12.17 VERTICALLY FRAGMENTED TABLE CONTENTS

Database name: Ch12_Text

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	6000.00	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

Figure 12.18 –Table Contents after Mixed Fragmentation

FIGURE 12.18 TABLE CONTENTS AFTER THE MIXED FRAGMENTATION PROCESS

Database name: Ch12_Text

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 Allocation Strategies

Centralized data allocation

- Entire database stored at one site

Partitioned data allocation

- Database is divided into two or more disjoined fragments and stored at two or more sites

Replicated data allocation

- Copies of one or more database fragments are stored at several sites

Data Replication

- Storage of data copies at multiple sites served by a computer network
- **Mutual consistency rule** requires all copies of data fragments be identical
- Styles of replication
  - Push replication focuses on maintaining data consistency
  - Pull replication focuses on maintaining data availability and allows for temporary data inconistencies

Data Replication Scenarios

Fully replicated database

- Stores multiple copies of each database fragment at multiple sites

Partially replicated database

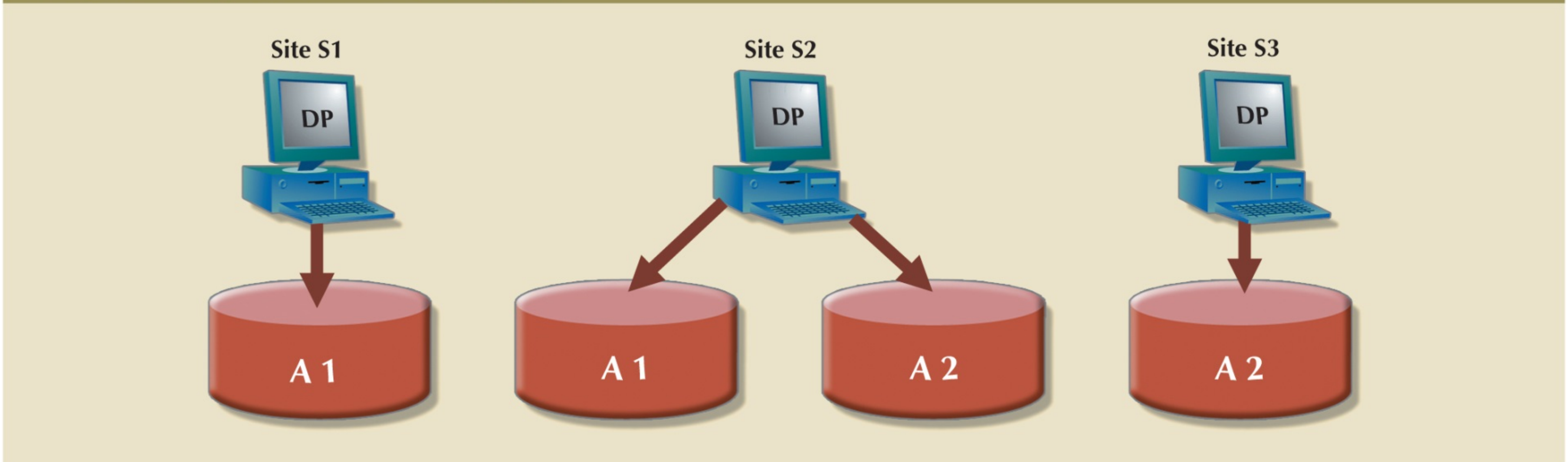
- Stores multiple copies of some database fragments at multiple sites

Unreplicated database

- Stores each database fragment at a single site

Figure 12.19 –Data Replication

FIGURE 12.19 DATA REPLICATION



The CAP Theorem

- CAP stands for:
  - Consistency
  - Availability
  - Partition tolerance
- Trade-off between consistency and availability generated in a new system in which data is **basically available, soft state, eventually consistent (BASE)**
  - Data changes are not immediate but propagate slowly through the system until all replicas are consistent

Table 12.8 - Distributed Database Spectrum

TABLE 12.8

DISTRIBUTED DATABASE SPECTRUM

DBMS TYPE	CONSISTENCY	AVAILABILITY	PARTITION TOLERANCE	TRANSACTION MODEL	TRADE-OFF
Centralized DBMS	High	High	N/A	ACID	No distributed data processing
Relational DBMS	High	Relaxed	High	ACID (2PC)	Sacrifices availability to ensure consistency and isolation.
NoSQL DDBMS	Relaxed	High	High	BASE	Sacrifices consistency to ensure availability

Table 12.9 - C. J. Date's 12 Commandments for Distributed Databases

TABLE 12.9

C. J. DATE'S 12 COMMANDMENTS FOR DISTRIBUTED DATABASES

RULE NUMBER	RULE NAME	RULE EXPLANATION
1	*Local-site independence*	Each local site can act as an independent, autonomous, centralized DBMS. Each site is responsible for security, concurrency control, backup, and recovery.
2	*Central-site independence*	No site in the network relies on a central site or any other site. All sites have the same capabilities.
3	*Failure independence*	The system is not affected by node failures. The system is in continuous operation even in the case of a node failure or an expansion of the network.
4	*Location transparency*	The user does not need to know the location of data to retrieve that data.
5	*Fragmentation transparency*	Data fragmentation is transparent to the user, who sees only one logical database. The user does not need to know the name of the database fragments to retrieve them.
6	*Replication transparency*	The user sees only one logical database. The DDBMS transparently selects the database fragment to access. To the user, the DDBMS manages all fragments transparently.

Table 12.9 - C. J. Date's 12 Commandments for Distributed Databases (cont'd.)

TABLE 12.9

C. J. DATE'S 12 COMMANDMENTS FOR DISTRIBUTED DATABASES

RULE NUMBER	RULE NAME	RULE EXPLANATION
7	*Distributed query processing*	A distributed query may be executed at several different DP sites. Query optimization is performed transparently by the DDBMS.
8	*Distributed transaction processing*	A transaction may update data at several different sites, and the transaction is executed transparently.
9	*Hardware independence*	The system must run on any hardware platform.
10	*Operating system independence*	The system must run on any operating system platform.
11	*Network independence*	The system must run on any network platform.
12	*Database independence*	The system must support any vendor's database product.