# A PRACTICAL EXAMPLE OF IMPLEMENTING A DISTRIBUTED DATABASE APPLICATION

**Runceanu Adrian, Assist. Prof., PhD**
**Gilca Gheorghe, Assist. Prof., Eng. PhD**

*"Constantin Brâncuși" University from Târgu Jiu, ROMANIA*

***ABSTRACT:*** *In this article we propose the practical implementation of a distributed database using the SQL query language. Our example consists in presenting a didactic application that can be used in the management of books in a student library.*

**KEYWORDS: Availability, Computer network, Distributed database, Reliability, Implementation**

## 1. INTRODUCTION

In centralized databases, data is stored together with all its attributes that define tuples. When this data is queried, all attributes are loaded into main memory, even if only part of them is needed. If the number of pages of queries is high, then the ratio of unused attributes to those attributes that are used is very high. If we had in tuples only those attributes that are required by the query, then the number of retrievals would be lower.

Therefore, if the file is partitioned in such a way that the attributes that are queried more, are placed together, then the number of pages accessed would be lower. This fragmentation of attributes, called vertical fragmentation (partitioning), can improve the performance of centralised databases. Moreover, the greatest effect of this partitioning can be seen in distributed databases.

According to [5], "distributed databases are a multiple collection of logically related databases arranged along a network". It is important to note two aspects of this definition: distributed database systems must be able to communicate between users (network), and data stored in different nodes must be related.

If a query from one node retrieves tuples from another node, the cost of having unwanted (unneeded) attributes in the tuple is higher than for centralised databases because there is a need to send these tuples across the network.

The cost of communication can be low if proper vertical fragmentation is applied.

Distributed databases are defined as databases located on different machines in the same location or in different locations that look like a single centralised database to the end user.

Thus, instead of having a single centralized database carrying the entire load, it is shared by a collection of machines / computer. It is actually a set of server machines working in synchronization to serve the needs of multiple users.

These machines in a distributed system are connected to each other either by wireless connection or by various means of communication that serve to transfer data at high speed. The machines do not have a shared memory nor do they share a clock.

Processors in a distributed system can range from microcomputers to workstations to mini-computers to computers used in everyday life.

Distributed database is very useful now because of the promising feature of managing a database.

.

Data is the most important property of any organization to ensure secure and proper management of organizational data

The distributed database can take this general form:

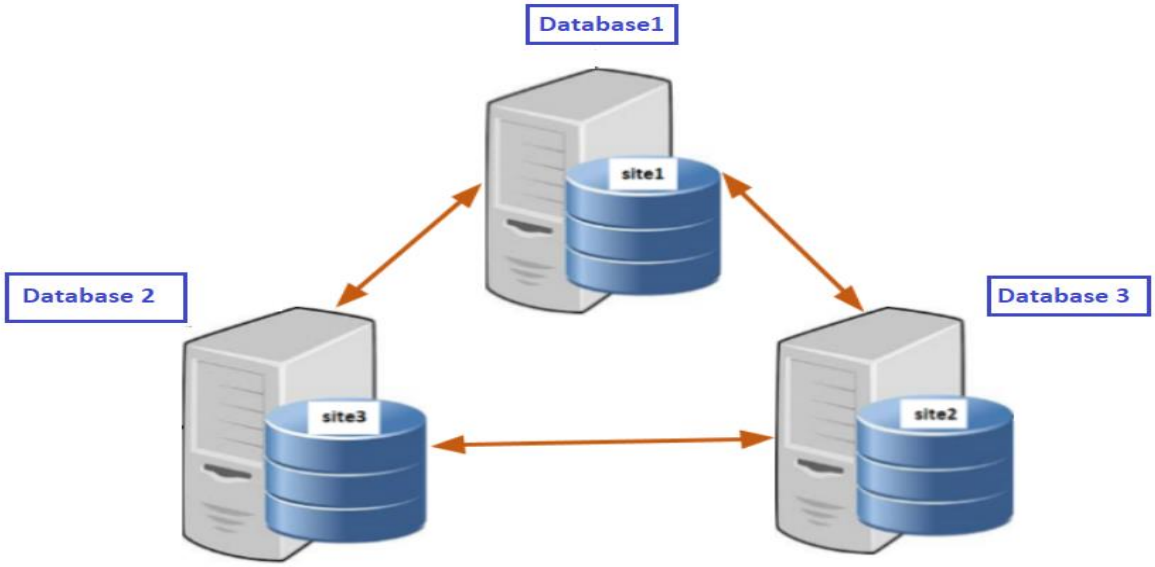


**Fig. 1 Distributed database**

## 2. ADVANTAGES OF THE DISTRIBUTED DATABASES

The advantages of the distributed database are the following:

a) Availability: data is replicated in multiple locations. If the local server is unavailable for any reason, the data can be retrieved from the other available server.

b) Robustness: the whole system becomes more robust as more servers are involved in data management. Thus, the failure of one system does not lead to the failure of the whole system.

c) Sharing: data from multiple sites is shared by users from different sites.

d) Performance: leads to improved performance as more machines are involved, the load is distributed. The database is divided into database chunks so that local queries can be resolved by rather than all queries being directed to a centralised database. Thus, query processing time is reduced and performance is increased.

e) Ease of growth: adding more clients to such a system is quite easy, as overload is never a problem.

f) Distributed data management with different levels of transparency, hardware, operating system, network and location independence. Provides continuous operation. No longer dependent on the central site.

## 3. DISADVANTAGES

a) Complexity: database administrators may have to go to extra lengths to ensure that the distributed nature of the system is transparent. They also have to do extra work to maintain several disparate systems instead of one big one. Additional work also needs to be done on database design to account for the disconnected nature of the database, e.g. joins become prohibitive when performed across multiple systems.

b) Economics: increased complexity and more extensive infrastructure means additional labour costs.

c) Security: remote database fragments need to be secured, and they are not centralised, so sites also need to be secured. The

infrastructure must also be secure (e.g. by encrypting network links between remote sites.
d) Difficult to maintain integrity: but in a distributed database, enforcing integrity across a network might require too many network resources to be feasible.
e) Inexperience: distributed databases are difficult to work with, and in such a young field there is not much experience readily available in "proper" practice.
f) Lack of standards: there are still no tools or methodologies to help users convert a centralised DBMS to a distributed DBMS
g) More complex database design: in addition to traditional database design challenges, the design of a distributed database must take into account data fragmentation, allocation of fragments to specific sites and data replication
h) Additional software is required
i) The operating system must support the distributed environment

## 4. PROBLEM AREAS OF DISTRIBUTED DATABASES

The following are distributed database problem domains.
1) **Distributed Concurrency Control**: distributed concurrency control specifies that synchronization of access to the distributed database so that the integrity of the database is maintained. In order to maintain concurrency in distributed databases, different locating techniques should be used that rely on mutual exclusion of data access. An algorithm is also used to mark the time when transactions are executed in a particular order [1].
2) **Distributed deadlock management**: in distributed database, multiple users request resources from the database if the resources are available at that time , then the database grants resources to that user, if they are not available, the user has to wait until the resources are released by other users. Sometimes resources are not released by users and are blocked by another user. This situation is known as blocking. Distributed blocking is handled using different algorithm and techniques such as avoidance and detection algorithm.
3) **Replication control**: replication is a technique that is only applicable to distributed systems. A database is said to be replicated if the whole database or a part of it (a table, some abele, one or more fragments, etc.) is copied and the copies are stored in different places. The problem with having more than one copy of a database is maintaining consistency between copies - ensuring that all copies have identical schema and data content [2].
4) **Operating environment**: to implement a distributed database environment a particular operating system is required depending on the organizational needs. The operating system also plays an important role in managing the distributed database. At some times, the operating system is not supported for the distributed database.
5) **Transparent management**: transparent data management is one of the main issues. In distributed database. In the distributed database, the data is in multiple locations and is used by multiple users. To maintain the integrity of the database, transparent data management is important.
6) **Security and privacy**: how to apply security policies to the interdependent system is a big issue in the distributed system. Because distributed systems deal with sensitive data and information, so the system must have a strong security and privacy metric. Protection of distributed system assets, including core resources, storage, communications, and user interface I/O, as well as higher-level components of these resources, such as processes, files, messages, display windows, and other more complex objects, are important issues in distributed systems
7) **Resource manage**ment: iIn distributed systems, objects consisting of resources are located in different places. Routing is an issue at the network layer of the distributed system and at the application layer. Resource management in a distributed system will interact with its heterogeneous nature.

## 5. IMPLEMENTATION OF A DISTRIBUTED DATABASE

Consider a distributed database for a bookstore with 3 sites called site1, site2 and site3. Consider the global schema consists of following relations:

Books (ISBN, Author, Topic, TotalStock, Price)
BookStore (Storeno, City, State, ZipCode, InventoryValue)
Stock (Storeno, ISBN, Quantity)

Total Stock is the total number of books in stock.

Now here in this distributed database for bookstore, we have fragmented the books according to the ISBN numbers into:

F1: books: ISBN from 978-973-16-10000 to 978-973-16-90000
F2: books: ISBN from 978-973-13-10000 to 978-973-13-90000
F3: books: ISBN from 978-973-10-10000 to 978-973-10-90000

Similarly, Book Stores are divided according to their store number into:

Site1: BookStore: Storeno from 101 to 199
Site2: BookStore: Storeno from 201 to 299
Site3: BookStore: Storeno from 301 to 399

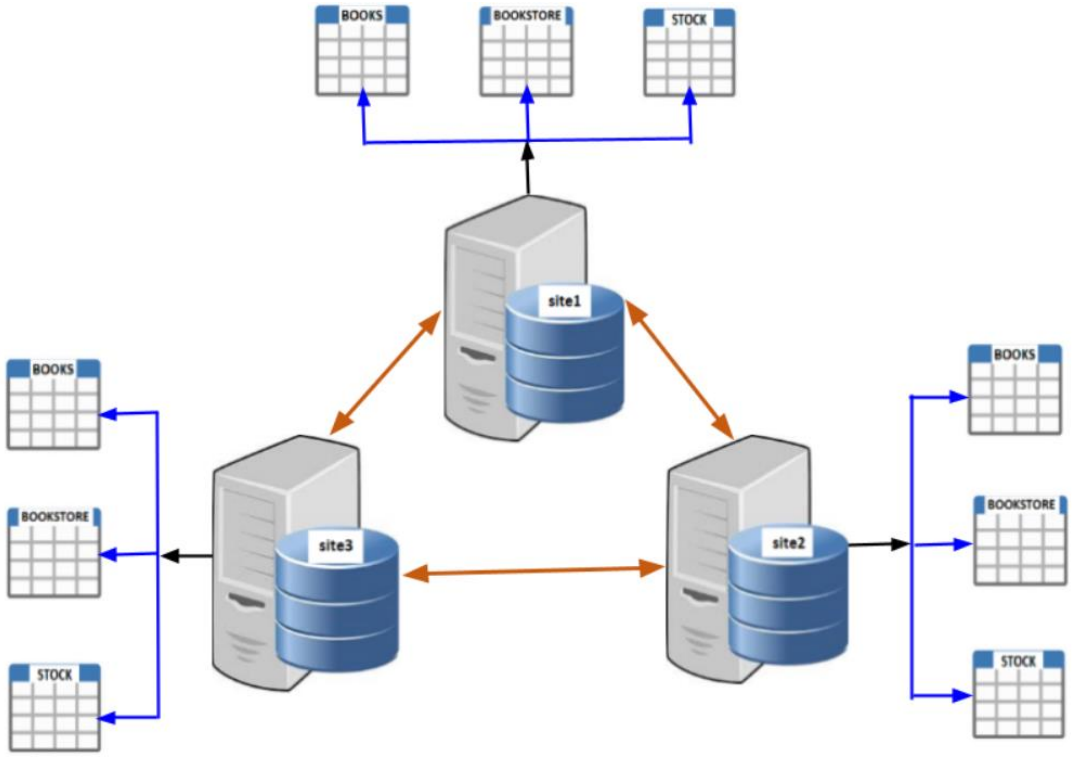


**Fig.2 Distributed database diagram for BOOKSTORE**

```
CREATE DATABASE site1;
USE site1;
CREATE TABLE BOOKS (ISBN int, AUTHOR varchar2(20), TOPIC varchar2(100), TOTALSTOCK int, PRICE int)
INSERT INTO SITE1.BOOKS VALUES (101, "Adrian Runceanu", "BAZE DE DATE", 50, 25);
INSERT INTO SITE1.BOOKS VALUES (102, "Octavian Dogaru", "PROGRAMAREA CALCULATOARELOR", 30, 55);
INSERT INTO SITE1.BOOKS VALUES (103, "Gheorghe Gilca", "APLICATII JAVA", 20, 15);
INSERT INTO SITE1.BOOKS VALUES (104, "Adrian Runceanu", "LIMBAJUL C++. APLICATII", 30, 44);
INSERT INTO SITE1.BOOKS VALUES (105, "Adrian Runceanu", "ALGORITMI. CULEGERE DE PROBLEME", 20, 22);
INSERT INTO SITE1.BOOKS VALUES (106, "Octavian Dogaru", "TEHNICI DE PROGRAMARE IN LIMBAJUL C++", 20, 25);
CREATE TABLE BOOKSTORE (STORENO int, CITY varchar2(30), STATE varchar2(100), ZIPCODE int, INVENTORYVALUE int)
```

INSERT INTO SITE1.BOOKSTORE VALUES (1, "Targu-Jiu", "GORJ", 210160, 7654);
INSERT INTO SITE1.BOOKSTORE VALUES (2, "Targu-Jiu", "GORJ", 210123, 1212);
INSERT INTO SITE1.BOOKSTORE VALUES (3, "Targu-Jiu", "GORJ", 210267, 1234);
INSERT INTO SITE1.BOOKSTORE VALUES (4, "Targu-Jiu", "GORJ", 210563, 4321);
INSERT INTO SITE1.BOOKSTORE VALUES (5, "Targu-Jiu", "GORJ", 210876, 9876);
INSERT INTO SITE1.BOOKSTORE VALUES (6, "Targu-Jiu", "GORJ", 210163, 8765);
CREATE TABLE STOCK (STORENO int, ISBN varchar2(20), QUANTITY int)
INSERT INTO SITE1.STOCK VALUES (1, "978-973-16-12345", 12);
INSERT INTO SITE1.STOCK VALUES (2, "978-973-16-54321", 10);
INSERT INTO SITE1.STOCK VALUES (3, "978-973-16-12222", 24);
INSERT INTO SITE1.STOCK VALUES (4, "978-973-16-54333", 65);
INSERT INTO SITE1.STOCK VALUES (5, "978-973-16-12444", 34);
INSERT INTO SITE1.STOCK VALUES (6, "978-973-16-55555", 20);

CREATE DATABASE site2;
USE site2;
CREATE TABLE BOOKS (ISBN int, AUTHOR varchar2(20), TOPIC varchar2(100), TOTALSTOCK int, PRICE int)
INSERT INTO SITE2.BOOKS VALUES (201, "Cristea Traian", "SISTEME DE OPERARE", 10, 60);
INSERT INTO SITE2.BOOKS VALUES (202, "Andrew Tanenbaum", "RETELE DE CALCULATOARE", 20, 40);
INSERT INTO SITE2.BOOKS VALUES (203, "Marian Popescu", "AUTOMATIZARI", 20, 70);
INSERT INTO SITE2.BOOKS VALUES (204, "Victor Cretu", "STRUCTURI DE DATE SI ALGORITMI", 20, 40);
INSERT INTO SITE2.BOOKS VALUES (205, "Nicholas Wirth", "ALGORITHMS AND DATA STRUCTURES", 40, 20);
INSERT INTO SITE2.BOOKS VALUES (206, "Liviu Negrescu", "LIMBAJELE C SI C++ PENTRU INCEPATORI. VOL. II",10, 30);
CREATE TABLE BOOKSTORE (STORENO int, CITY varchar2(30), STATE varchar2(100), ZIPCODE int, INVENTORYVALUE int)
INSERT INTO SITE2.BOOKSTORE VALUES (11, "Craiova", "DOLJ", 123456, 9876);
INSERT INTO SITE2.BOOKSTORE VALUES (12, "Craiova", "DOLJ", 654321, 3434);
INSERT INTO SITE2.BOOKSTORE VALUES (13, "Craiova", "DOLJ", 122226, 9888);
INSERT INTO SITE2.BOOKSTORE VALUES (14, "Craiova", "DOLJ", 654441, 2434);
INSERT INTO SITE2.BOOKSTORE VALUES (15, "Craiova", "DOLJ", 122225, 9121);
INSERT INTO SITE2.BOOKSTORE VALUES (16, "Craiova", "DOLJ", 321654, 5454);
CREATE TABLE STOCK (STORENO int, ISBN varchar2(70), QUANTITY int)
INSERT INTO SITE2.STOCK VALUES (11, "978-973-13-12345", 12);
INSERT INTO SITE2.STOCK VALUES (12, "978-973-13-54321", 10);
INSERT INTO SITE2.STOCK VALUES (11, "978-973-13-34212", 11);
INSERT INTO SITE2.STOCK VALUES (12, "978-973-13-67811", 20);
INSERT INTO SITE2.STOCK VALUES (11, "978-973-13-19875", 33);
INSERT INTO SITE2.STOCK VALUES (12, "978-973-13-58881", 40);

CREATE DATABASE site3;
USE site3;
CREATE TABLE BOOKS (ISBN int, AUTHOR varchar2(20), TOPIC varchar2(100), TOTALSTOCK int, PRICE int)
INSERT INTO SITE3.BOOKS VALUES (301, "Tudor Sorin", "BAZELE PROGRAMARII IN JAVA", 20, 30);
INSERT INTO SITE3.BOOKS VALUES (302, "Doina Logofatu", "ALGORITMI FUNDAMENTALI IN JAVA", 10, 50);
INSERT INTO SITE3.BOOKS VALUES (303, "Horia Georgescu", "INTRODUCERE IN UNIVERSUL JAva", 20, 25);
INSERT INTO SITE3.BOOKS VALUES (304, "Mihai Fotache", "ORACLE 9i", 20, 40);
INSERT INTO SITE3.BOOKS VALUES (305, "Valentn Velea", "INTEROGAREA BAZELOR DE DATE", 50, 20);
INSERT INTO SITE3.BOOKS VALUES (306, "Adrian Runceanu", "INTERNET SI INTRANET", 40, 40);
CREATE TABLE BOOKSTORE (STORENO int, CITY varchar2(30), STATE varchar2(100), ZIPCODE int, INVENTORYVALUE int)
INSERT INTO SITE3.BOOKSTORE VALUES (311, "Bucuresti", "Bucuresti", 123456, 9876);
INSERT INTO SITE3.BOOKSTORE VALUES (312, "Bucuresti", "Bucuresti", 654321, 3434);
INSERT INTO SITE3.BOOKSTORE VALUES (313, "Bucuresti", "Bucuresti", 111111, 4446);
INSERT INTO SITE3.BOOKSTORE VALUES (314, "Bucuresti", " Bucuresti", 633333, 3111);
INSERT INTO SITE3.BOOKSTORE VALUES (315, " Bucuresti", " Bucuresti", 999999, 9111);
INSERT INTO SITE3.BOOKSTORE VALUES (316, " Bucuresti", " Bucuresti", 888888, 2233);
CREATE TABLE STOCK (STORENO int, ISBN varchar2(70), QUANTITY int)

```
INSERT INTO SITE3.STOCK VALUES (311, "978-973-10-12345", 12);
INSERT INTO SITE3.STOCK VALUES (312, "978-973-10-54321", 10);
INSERT INTO SITE3.STOCK VALUES (313, "978-973-10-12222", 12);
INSERT INTO SITE3.STOCK VALUES (314, "978-973-10-55436", 10);
INSERT INTO SITE3.STOCK VALUES (315, "978-973-10-62745", 12);
INSERT INTO SITE3.STOCK VALUES (316, "978-973-10-51221", 10);
```

Now from Site 3, we want to check on the total number of books available at each site.

```
use site3;
SELECT SUM(QUANTITY) FROM site3.STOCK;
SELECT SUM(QUANTITY) FROM site2.STOCK;
SELECT SUM(QUANTITY) FROM site1.STOCK;
```

We are on site1 and we want to access the books on site3, site 2. Now it is possible using distributed database.

```
use site1;
SELECT * FROM site3.BOOKS;
UPDATE site3.BOOKS SET PRICE=123 WHERE ISBN="978-973-10-54321";
SELECT * FROM site3.BOOKS;
SELECT * FROM site2.BOOKS;
UPDATE site2.BOOKS SET PRICE=999 WHERE ISBN="978-973-13-54321";
SELECT * FROM site2.BOOKS;
```

From site 2, we want to check the available copies of particular book with ISBN number in the bookstore. According to the ISBN number in which fragment it belongs, search in the database.

```
use site2;
SELECT STORENO, QUANTITY FROM site3.STOCK WHERE ISBN="978-973-10-12345";
```

Get the list of all the books available in the bookstore from any site.

```
use site2;
SELECT * FROM site1.BOOKS
UNION
SELECT * FROM site2.BOOKS
UNION
SELECT * FROM site3.BOOKS;
```

Get the list of all the stores from any site.

```
use site3;
SELECT * FROM site1.BOOKSTORE
UNION
SELECT * FROM site2.BOOKSTORE
UNION
SELECT * FROM site3.BOOKSTORE
```

## 6. CONCLUSIONS

The example of a distributed database implementation in a library book management application is built for teaching purposes only. However it can be used as a good practical method of implementing the notion of a distributed database.

## 7. REFERENCES

[1] Andrew S Tanenbaum, Maarten Steen. *Distributed Systems: Principles and Paradigms,* Upper Saddle River, NJ: Pearson Prentice Hall, 2007

[2] Stefano Ceri, Giuseppe Pelagatti, *Distributed Databases: Principles and System,* McGraw-Hill Book Company 1984, ISBN 0-07-010829-3

[3] Sharad Maheshwari, Ruchin Jain, *DBMS Complete Practical Approach,* 2nd Edition, Published 2006 by Laxmi Publications

[4] Coulouris, George F. *Distributed Systems: Concepts and Design*, Boston: Addison-Wesley, 2012

[5] M.T. Ozru and P. Valduriez. *Principles of distributed Database Systems.* Prentice Hall, Englewood Cliffs, NJ, 1999.

[6] Saeed K. Rahimi, Frank S.Haug, *A Distributed Database Management System a practical approach,* Wiley Publication, 2010