

# Notopedia

## Self Learning Study Material(BIBLIOPHILY)

<sup>1</sup>Anjali S.Jibhakate, <sup>2</sup>Ankita J. Gakare, <sup>3</sup>Rinku C. Nag, <sup>4</sup>Prof.Rahul V. Bambodkar

<sup>1, 2, 3, 4</sup>Department of (C.S.E)

<sup>1, 2, 3</sup>Datta Meghe College of Engineering, Wardha

<sup>1</sup>ranipopy.83@gmail.com, <sup>2</sup>ankitagakare99@gmail.com, <sup>3</sup>rinkunag629@gmail.com, <sup>4</sup>rahulbambodkar1@gmail.com

**Abstract**— This project aims at addressing the formal system of students academic, technical activities and notices management system by implementation of an online system for these purposes. Colleges follow a manual system to provide students with their notices, notes and other important documents that are needed. This current system provides students with the bibliophilic data by means of printed notes and documentation. Notices are provided on a notice board which is accessible to students in college premises only. This current analogue system requires a lot of paperwork and cost to provide students with the basic documentation like notes and question papers. Also this analogue system fails at providing the students the flexibility required in terms of time and location, to access these documentation. The concept of notice board in the college premises is restricted by strictly location requiring students to travel for getting access to the notices Displayed for them. Moreover this analogue technique is responsible for consumption of more time than is really required for such a short process. Development of this system ensures at addressing these drawbacks of the current system of flexibility and time management. This system uses a web based approach to address and solve these issues by creating an online management system to manage the notes, notices, question papers and student data, unlike formal system. Also databases have been kept distributed so as to increase the security of data in the system.

**Keywords** —: *Bibliophily, Interactive Paper, Distributed Databases, Design & use of Web Technology*

\*\*\*\*\*

### INTRODUCTION

It is book love that enables us to perceive whatever is true and beautiful in books, and it is a passport to the purest and the perfect pleasures possible to men. We are never really well-bred until we have attained ability to know and to love real books: it is almost all a matter of education "of self-education; and the completer the culture, the deeper-rooted the appreciation and the greater the influence. Book love is a mark of refinement, and we are only fractions of men without it.

Book love supplies each day and each hour with an endless stream of independent and rational pleasure, and we need not hope for anything really worthy of a Christian or an American from the man who does not at times love to stay in his own room in the enabling company of the great men who live in books. We all are made or marred by the company we keep, whether of men or of books. No darkness from without can ever obscure the light and the sweetness within, which is forever the portion of the man who loves books.

To become entirely worthy of our Aicmaen manhood, we must Uve among books and live lovingly among them: the soul of man has nowhere else so stamped his image as in this world of books; and, to be fair of head and heart, we must find a home within this world.

The plain-people are the pillars of every democracy what they are decides the strength or the weakness of our nation; and nothing can conduce more to their being somebody and to their doing something, than to help them early to become acquainted with the good and the better and the best among books, to inspire them to make books their steadfast friends and lifelong companions, and to establish in them the conviction that even illiteracy is preferable to bad, worse, worst.

A web portal is most often one specially-designed Webpage at a website which brings information together from diverse sources in a uniform way. Usually, each information source gets its dedicated area on the page for displaying information (a port let); often, the user can configure which ones to display. Variants of portals include Mash up (web application hybrid) and intranet "dashboards" for executives and managers.

The extent to which content is displayed in a "uniform way" may be more or less depending on the intended user and the intended purpose as well as the diversity of the content. Very often design emphasis is on a certain "metaphor" for configuring and customizing the presentation of the content and the chosen implementation framework

577

and/or code libraries. In addition, the role of the user in an organization may determine which content can be added to the portal or deleted from the portal configuration.

### 1.1 Distributed Databases:

Centralized distributed database management system (DDBMS) manages the database as if it were all stored on the same computer. The DDBMS synchronizes all the data periodically and, in cases where multiple users must access the same data, ensures that updates and deletes performed on the data at one location will be automatically reflected in the data stored elsewhere.

The users and administrators of a distributed system, should, with proper implementation, interact with the system as if the system was centralized. This transparency allows for the functionality desired in such a structured system without special programming requirements, allowing for any number of local and/or remote tables to be accessed at a given time across the network.

The different types of transparency sought after in a DDBMS are data distribution transparency, heterogeneity transparency, transaction transparency, and performance transparency. Data distribution transparency requires that the user of the database should not have to know how the data is fragmented (fragmentation transparency), know where the data they access is actually located (location transparency), or be aware of whether multiple copies of the data exist (replication transparency).

Heterogeneity transparency requires that the user should not be aware of the fact that they are using a different DBMS if they access data from a remote site. The user should be able to use the same language that they would normally use at their regular access point and the DDBMS should handle query language translation if needed. Transaction transparency requires that the DDBMS guarantee that concurrent transactions do not interfere with each other (concurrency transparency) and that it must also handle database recovery (recovery transparency). Performance transparency mandates that the DDBMS should have a comparable level of performance to a centralized DBMS. Query optimizers can be used to speed up response time.

The Mariposa distributed database system addresses a fundamental problem in the standard approach to distributed data management. We argue that the underlying assumptions traditionally made while implementing distributed data

managers do not apply to today's wide-area network (WAN) environments.

A distributed database system allows applications to access data from local and remote databases. In a homogenous distributed database system, each database is an Oracle database. In a heterogeneous distributed database system, at least one of the databases is a non-Oracle database. Distributed databases use client/server architecture to process information requests.

This section contains the following topics:

- Homogenous Distributed Database Systems
- Heterogeneous Distributed Database Systems
- Client/Server Database Architecture

### 1.2 Heterogeneous Distributed Database Systems

In a heterogeneous distributed database system, at least one of the databases is a non-Oracle system. To the application, the heterogeneous distributed database system appears as a single, local, Oracle database. The local Oracle database server hides the distribution and heterogeneity of the data.

The Oracle database server accesses the non-Oracle system using Oracle Heterogeneous Services in conjunction with an agent. If you access the non-Oracle data store using an Oracle Transparent Gateway, then the agent is a system-specific application. For example, if you include a Sybase database in an Oracle distributed system, then you need to obtain a Sybase-specific transparent gateway so that the Oracle databases in the system can communicate with it.

### 1.3 Client/Server Database Architecture

A database server is the Oracle software managing a database, and a client is an application that requests information from a server. Each computer in a network is a node that can host one or more databases. Each node in a distributed database system can act as a client, a server, or both, depending on the situation.

The host for the hq database is acting as a database server when a statement is issued against its local data (for example, the second statement in each transaction issues a statement against the local depttable), but is acting as a client when it issues a statement against remote data (for example, the first statement in each transaction is issued against the remote table emp in the sales database).

Vertical partitioning is applied in three contexts: a database stored on devices of a single type, a database stored in different memory levels, and a distributed database. In a two-level memory hierarchy, most transactions should be processed using the fragments in primary memory. In distributed databases, fragment allocation should maximize the amount of local transaction processing.

**1.4 PHP Code Design:**

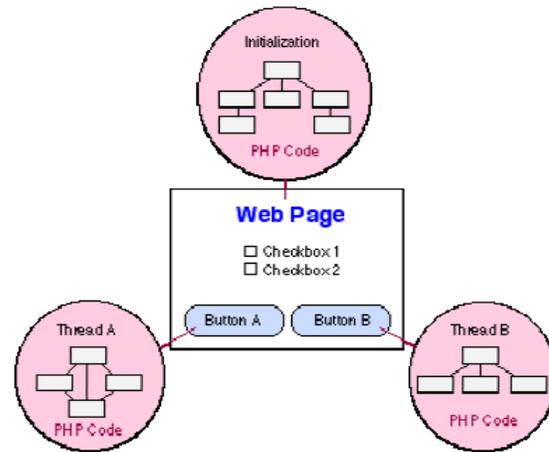
PHP is a server-side, open-source, HTML-embedded scripting language used to drive many of the world’s most popular web sites. All major web servers support PHP enabling normal HTML pages to embed code fragments that get interpreted before being served up to requesting users. PHP is often used with MySQL, a popular open-source database engine.

PHP is a modern language for web site development with both procedural and object-oriented programming and integrated database support. It often serves as the glue code to connect web pages to a server-side database. An interactive web site is event driven. Each user click, menu selection or keyboard action can initiate a thread of execution that runs many lines of code.

Many web projects start as small code snippets gathered from open source sites. As the web site grows to accommodate new data interfaces, web pages and more user features, new developers are added to the team. Without design and documentation tools, the project often becomes unmanageable, unreliable, insecure and costly to expand.

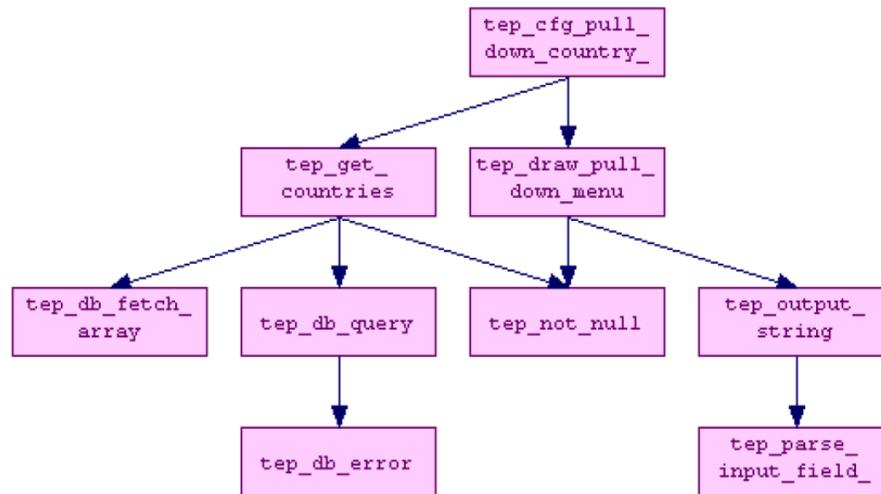
This paper shows techniques used to organize and communicate the structure of existing PHP projects and design

new enhancements. Software models provide the framework that enables a development team to work together productively and understand, evaluate integrate or implement new source code for a PHP project.



*Server-Side PHP Threads of Execution for a Web Page*

Each thread of execution is represented with a Structure Chart diagram. At the top of the diagram, the root module is shown as a named box. Each called module is shown as a named box connected with a line and arrowhead pointing to it.



*Structure Chart for PHP Thread of Execution*

Procedural programs define data structures and functions that access and manipulate that data. As a program grows in size and complexity, the structural hierarchy of its modules also grows. It becomes increasingly difficult to keep

track of which functions or threads of execution are manipulating which data. Seemingly small changes can ripple through the software system with unintended affects. An object-oriented program organizes related data and functions into classes. To reduce the ripple effect of program changes,

the actual data structures are often hidden from the other classes or functions that access that data through methods. The physical data structure can change while the interface of the access method and its callers are unaffected. Classes are reusable, self-contained, conceptual units of a program's code and data.

PHP supports object-oriented programming with classes, interfaces, attributes, operations, constructors and destructors. A class can inherit another class, then reuse or override its methods. Attribute and operation members of a class can have public, private and protected access to limit accessibility of class members to other parts of the program.

One could expect an MCCS-DM to maintain a very large number of smaller databases. In addition, the MCCS-DM should access the larger permanent databases located at sites such as hospitals, police stations, and ground stations. Information from these larger databases would often be cached and stored in the smaller databases.

There are several, storage management issues that need to be investigated. First of all, how often should the information be cached. What is the size of each cache? What sort of storage structures should be used. Traditional databases use structure such as B-Tress. Are these structures suitable for MCCS-DM? What sort of index strategies and access methods should be supported? Are special strategies ended to support multimedia data? What is the relationship the storage manager has to the other modeule such as query processing, update/transaction manager, and metadata manager? What is the impact of data distribution on storage management techniques. Much research is needed on storage management issues.

## CONCLUSION

Mobile computing and communication systems will require effective management, retrieval, and integration of databases which are possibly heterogeneous in nature. So the comparative study of the Distributed Database systems and the PHP code designing helps to develop a platform or web technology from where the normal person can interact the world around him. And which is getting more and more efficient with the concepts of Distributed Databases.

## REFERENCES

- [1] BABAD, M.J. A record and file partitioning model. Commun ACM 20, 1 (Jan. 1977).
- [2] BERNSTEIN, P.A., GOODMAN, N., WONG, E., REEVE, C.L., AND ROTHNIE, J.B., JR. Query processing in a system for distributed databases (SDD-1). ACM Trans. Database Syst. 6,4 (Dec.1981), 602-625.
- [3] CERI, S., NEGRI, M., AND PELAGATTI, G. Horizontal data partitioning in database design. In Proceedings ACM SIGMOD International Conference on Management of Data (Orlando, Fla., 1982), ACM, New York.
- [4] CERI, S., NAVATHE, S.B., AND WIEDERHOLD, G. Distribution design of logical database schemas. IEEE Trans. Softw. Eng. SE-g, 3 (July 1983).
- [5] CERI, S., AND PELAGATTI, G. The allocation of operations in distributed database access. IEEE Trans. Comput. TC-32,2 (1982).
- [6] Banerjea A Mah BA (1991) The real-time channel administration protocol. In: Proc 2nd Int Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, Germany, November
- [7] Bernstein PA, Goodman N, Wong E, Reeve CL, Rothnie J (1981) Query processing in a system for distributed databases (SDD-1). ACM Trans Database Syst 6:602–625
- [8] Bitton D, DeWitt DJ, Turbyfill C (1983) Benchmarking data base systems: a systematic approach. In: Proc 9th Int Conf on Very Large Data Bases, Florence, Italy, November
- [9] Cheriton D, Mann TP (1989) Decentralizing a global naming service for improved performance and fault tolerance. ACM Trans Comput Syst 7:147–183
- [10] [CERI84] Ceri, S. and G Pelagatti, "Distributed Databases, Principles and Systems," McGraw Hill, NY,1984.
- [11] [1:M1E92] Irnielinski, T. and B. Badrinath, "Querying in Highly Mobile Distributed Environments," Proceedings of the 18th Very Large Database Conference, Vancouver, B.C., August 1992.
- [12] [SHETH90] Sheth, A. and J. Larson, September 1990, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," ACM Computing Surveys, Vol. 22, #3.
- [13] [SIGN93] Small Mobile Devices Link Army Combined Arms Team, SIGNAL, November 1993.
- [14] [THUR93] Thuraisingham, B., October 1993, "Developments and Challenges in Distributed Database Systems," Handbook of Local Area Network, 1993 (Ed: J. Slone and A. Drinan).