ST. XAVIER'S COLLEGE

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Database Management System**

**LAB ASSIGNMENT # 3**

**SUBMITTED BY:**

Ronak Agrawal  
013BSCCSIT033

4th Semester

**SUBMITTED TO:**

|  |  |
| --- | --- |
| **Er. Sanjay kr. Yadav**  Lecturer,  Department of Computer Science  St. Xavier’s College |  |

**Date of Submission:** 3rd August, 2015

### Database system components

A database management system (DBMS) consists of several components. Each component plays very important role in the database management system environment. The major components of database management system are:

* Software
* Hardware
* Data
* Procedures
* Database Access Language
* User

**Software:** The main component of a DBMS is the software. It is the set of programs used to handle the database and to control and manage the overall computerized database

DBMS software itself, is the most important software component in the overall system

* Operating system including network software being used in network, to share the data of database among multiple users.
* Application programs developed in programming languages such as C++, Visual Basic that are used to to access database in database management system. Each program contains statements that request the DBMS to perform operation on database.
* The operations may include retrieving, updating, deleting data etc . The application program may be conventional or online  workstations or terminals.

**Hardware:** Hardware consists of a set of physical electronic devices such as computers (together with associated I/O devices like disk drives), storage devices, I/O channels, electromechanical devices that make interface between computers and the real world systems etc, and so on. It is impossible to implement the DBMS without the hardware devices, In a network, a powerful computer with high data processing speed and a storage device with large storage capacity is required as database server.

**Data:** Data is the most important component of the DBMS. The main purpose of DBMS is to process the data. In DBMS, databases are defined, constructed and then data is stored, updated and retrieved to and from the databases. The database contains both the actual (or operational) data and the metadata (data about data or description about data).

**Procedures:** Procedures refer to the instructions and rules that help to design the database and to use the DBMS. The users that operate and manage the DBMS require documented procedures on hot use or run the database management system. These may include.

Procedure to install the new DBMS.

* To log on to the DBMS.
* To use the DBMS or application program.
* To make backup copies of database.
* To change the structure of database.
* To generate the reports of data retrieved from database.

**Database Access Language:** The database access language is used to access the data to and from the database. The users use the database access language to enter new data, change the existing data in database and to retrieve required data from databases. The user write a set of appropriate commands in a database access language and submits these to the DBMS. The DBMS translates the user commands and sends it to a specific part of the DBMS called the Database Jet Engine. The database engine generates a set of results according to the commands submitted by user, converts these into a user readable form called an Inquiry Report and then displays them on the screen. The administrators may also use the database access language to create and maintain the databases.

The most popular database access language is SQL (Structured Query Language). Relational databases are required to have a database query language.

**Users:** The users are the people who manage the databases and perform different operations on the databases in the database system.There are three kinds of people who play different roles in database system

* Application Programmers
* Database Administrators
* End-Users

**Database system utilities**

* **Loading:** A loading utility is used to load existing data files-such as text files or sequential files-into the database. Usually, the current (source) format of the data file and the desired (target) database file structure are specified to the utility, which then automatically reformats the data and stores it in the database. With the proliferation of DBMSs, transferring data from one DBMS to another is becoming common in many organizations. Some vendors are offering products that generate the appropriate loading programs, given the existing source and target database storage descriptions (internal schemas). Such tools are also called conversion tools.
* **Backup:** A backup utility creates a backup copy of the database, usually by dumping the entire database onto tape. The backup copy can be used to restore the database in case of catastrophic failure. Incremental backups are also often used, where only changes since the previous backup are recorded. Incremental backup is more complex but saves space.
* **File Reorganization:** This utility can be used to reorganize a database file into a different file organization to improve performance.
* **Performance Monitoring:**Such a utility monitors database usage and provides statistics to the DBA. The DBA uses the statistics in making decisions such as whether or not to reorganize files to improve performance.

**Classification of DBMS**

A database management system (DBMS) is a [computer software](https://en.wikipedia.org/wiki/Computer_software) application that interacts with the user, other applications, and the database itself to capture and analyze data. A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases. Well-known DBMSs include [MySQL](https://en.wikipedia.org/wiki/MySQL" \o "MySQL), [PostgreSQL](https://en.wikipedia.org/wiki/PostgreSQL" \o "PostgreSQL),[Microsoft SQL Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server), [Oracle](https://en.wikipedia.org/wiki/Oracle_Database), [Sybase](https://en.wikipedia.org/wiki/Sybase) and [IBM DB2](https://en.wikipedia.org/wiki/IBM_DB2).

The database management systems can be classified based on several criteria.

**BASED ON DATA MODEL**

The most popular data model in use today is the relational data model. Well known DBMSs like Oracle, MS SQL Server, DB2, MySQL support this model. Other traditional models can be named hierarchical data model, or network data model. In the recent years, we are getting familiar with object-oriented data models but these models have not had widespread use. Some examples of Object-oriented DBMSs are O2, ObjectStore or Jasmine.

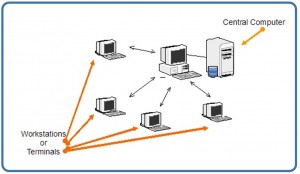
**BASED ON THE NUMBER OF USERS**

We can have a single user database system which supports one user at a time or multiuser systems which support multiple users concurrently.

**BASED ON THE WAYS DATABASE IS DISTRIBUTED**

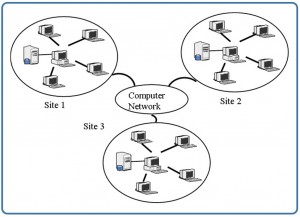
* **Centralized Systems**

With centralized database systems, the system is stored at a single site.



##### Distributed database system

Actual database and DBMS software are distributed in various sites connected by a computer network.



##### Homogeneous distributed Database Systems

Use the same DBMS software at multiple sites.  Data is exchanged between various sites and can be handled easily.

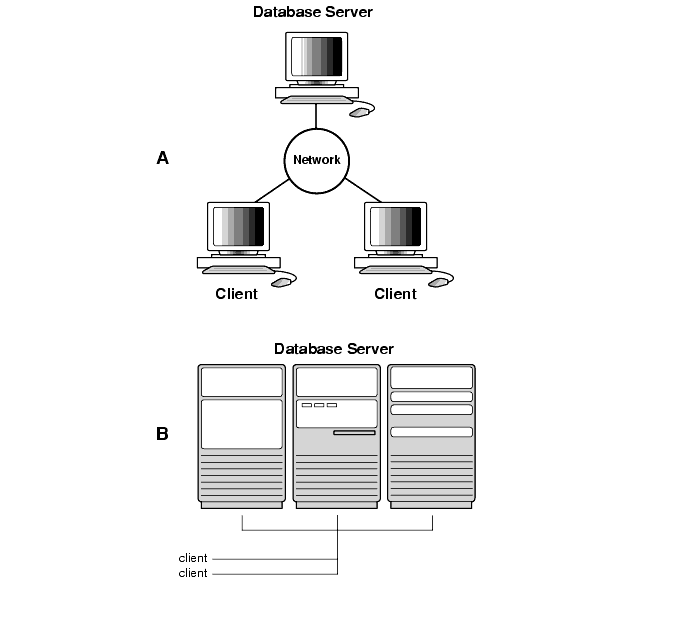
##### Heterogeneous distributed Database Systems

Different sites might use different DBMS software.  There is additional software to support data exchange between sites.

**Variation of distributed environments**

Distributed processing is the use of more than one processor to perform the processing for an individual task. Examples of distributed processing in Oracle database systems appear in figure.

* In Part A of the figure, the client and server are located on different computers; these computers are connected via a network. The server and clients of an Oracle database system communicate via Net8, Oracle's network interface.
* In Part B of the figure, a single computer has more than one processor, and different processors separate the execution of the client application from Oracle.



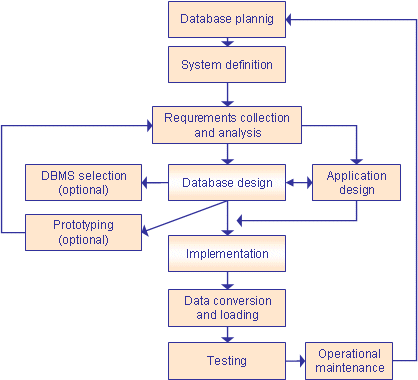
**Fig: client server architecture and distributed Processing**

Oracle client/server architecture in a distributed processing environment provides the following benefits:

* Client applications are not responsible for performing any data processing. Rather, they request input from users, request data from the server, and then analyze and present this data using the display capabilities of the client workstation or the terminal (for example, using graphics or spreadsheets).
* Client applications are not dependent on the physical location of the data. If the data is moved or distributed to other database servers, the application continues to function with little or no modification.
* Oracle exploits the multitasking and shared-memory facilities of its underlying operating system. As a result, it delivers the highest possible degree of concurrency, data integrity, and performance to its client applications.
* Client workstations or terminals can be optimized for the presentation of data (for example, by providing graphics and mouse support) and the server can be optimized for the processing and storage of data (for example, by having large amounts of memory and disk space).
* In networked environments, you can use inexpensive client workstations to access the remote data of the server effectively.
* If necessary, Oracle can be scaled as your system grows. You can add multiple servers to distribute the database processing load throughout the network (horizontally scaled), or you can move Oracle to a minicomputer or mainframe, to take advantage of a larger system's performance (vertically scaled). In either case, all data and applications are maintained with little or no modification, since Oracle is portable between systems.
* In networked environments, shared data is stored on the servers, rather than on all computers in the system. This makes it easier and more efficient to manage concurrent access.
* In networked environments, client applications submit database requests to the server using SQL statements. Once received, the SQL statement is processed by the server, and the results are returned to the client application. Network traffic is kept to a minimum because only the requests and the results are shipped over the network.

**Database system life cycle**

A database is usually a fundamental component of the information system, especially in business oriented systems. Thus database design is part of system development. The following picture shows how database design is involved in the system development lifecycle.  
  
The phases in the middle of the picture (Database Design, Database Implementation) are the phases that you concentrate on in the Database Design course. The other phases are briefly described. They are part of the contents of the Systems Analysis and Design courses, for example.  
  
There are various methods of how the different phases of information system design, analysis and implementation can be done. Here the main tasks or goals are described but no method is introduced.



**Database Planning**

The database planning includes the activities that allow the stages of the database system development lifecycle to be realized as efficiently and effectively as possible. This phase must be integrated with the overall Information System strategy of the organization.  
  
The very first step in database planning is to define the mission statement and objectives for the database system. That is the definition of:  
- the major aims of the database system  
- the purpose of the database system  
- the supported tasks of the database system  
- the resources of the database system

**Systems Definition**

In the systems definition phase, the scope and boundaries of the database application are described. This description includes:  
- links with the other information systems of the organization  
- what the planned system is going to do now and in the future  
- who the users are now and in the future.  
  
The major user views are also described. i.e. what is required of a database system from the perspectives of particular job roles or enterprise application areas.

**Requirements Collection and Analysis**

During the requirements collection and analysis phase, the collection and analysis of the information about the part of the enterprise to be served by the database are completed. The results may include eg:  
- the description of the data used or generated  
- the details how the data is to be used or generated  
- any additional requirements for the new database system

**Database Design**

The database design phase is divided into three steps:  
- conceptual database design  
- logical database design  
- physical database design  
  
In the conceptual database design phase, the model of the data to be used independent of all physical considerations is to be constructed. The model is based on the requirements specification of the system.  
  
In the logical database design phase, the model of the data to be used is based on a specific data model, but independent of a particular database management system is constructed. This is based on the target data model for the database e.g. relational data model.  
  
In the physical database design phase, the description of the implementation of the database on secondary storage is created. The base relations, indexes, integrity constraints, security, etc. are defined using the SQL language.

**Database Management System Selection**

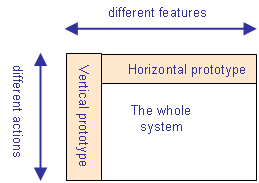
This in an optional phase. When there is a need for a new database management system (DBMS), this phase is done. DBMS means a database system like Access, SQL Server, MySQL, Oracle.  
  
In this phase the criteria for the new DBMS are defined. Then several products are evaluated according to the criteria. Finally the  
recommendation for the selection is decided.

**Application Design**

In the application design phase, the design of the user interface and the application programs that use and process the database are defined and designed.

**Protyping**

The purpose of a prototype is to allow the users to use the prototype to identify the features of the system using the computer.  
  
There are horizontal and vertical prototypes. A horizontal prototype has many features (e.g. user interfaces) but they are not working. A vertical prototype has very few features but they are working. See the following picture.



**Implementation**

During the implementation phase, the physical realization of the database and application designs are to be done. This is the programming phase of the systems development.

**Data Conversion and Loading**

This phase is needed when a new database is replacing an old system. During this phase the existing data will be transferred into the new database.

**Testing**

Before the new system is going to live, it should be thoroughly tested. The goal of testing is to find errors! The goal is not to prove the software is working well.

**Operational Maintenance**

The operational maintenance is the process of monitoring and maintaining the database system.  
Monitoring means that the performance of the system is observed. If the performance of the system falls below an acceptable level, tuning or reorganization of the database may be required.  
  
Maintaining and upgrading the database system means that, when new requirements arise, the new development lifecycle will be done. Source: Connolly, Begg. 2005. Database Systems. A Practical Approach to Design, Implementation, and Management. Addison Wesley. Chapter 9. Database Planning, Design and Administration.