An Architectural Framework for a Medical Image Database Management System

Seyyedeh Leila Seyed Bagheri Roudbari 1*, Fatemeh Ahmadi Abkenari 2

1 Master Student in Software Engineering, Department of Computer Engineering, Rasht Branch, Islamic Azad University, Rasht, Iran.
2 Assistant Professor, Faculty of Computer Engineering and Information Technology, Payame Noor University of Iran (PNU).

* Email of Corresponding Author: Bagheri_Leila27@yahoo.com

ABSTRACT

Medical image database management systems are such systems that because of involving with data of huge sizes need special consideration in satisfying performance requirements such as retrieval speeds and storage spaces. Although there are such systems that are available as commercial applications such as MEDIMAN, PACS, KMed and MIAPS but there are no comparisons or suggestions on best database management based approaches in literature to handle different inputs for managing such a database.

The aim of this paper is to develop an architectural framework for a medical database management system as an image database management platform. To this end, at first a brief review is carried out on the main characteristics of the existed medical information management systems such as MEDIMAN, PACS, MIAPS and KMed. The proposed system consists of several layers and a Web-based structure with a platform independent behavior. The system was tested with real pathology images in order to assess the performance. To evaluate and measure the speed of the proposed information retrieval system, different storage engines in MySQL including InnoDB, Archive and … as well as different modes of logical database design approaches were tested.

Keywords: Database architecture, Medical image database management, Medical information system, Medical image storage.

1. INTRODUCTION

Medical information system is a computer-based system which records, creates and deletes patient data records. The purpose of medical information systems is to support hospital activities at practical, scientific and strategic levels [2]. The fact is that, nowadays, hospitals cannot still adhere to the conventional process of recording patient’s dossiers and archiving operations including medical images, access to a part of patients’ information which may be different at times, outpatients, hospitalization information and critical care services. Therefore, it is necessary to create a medical information system in any hospital for increasing accuracy, expediting service provision to patients, profiling, retrieving patients’ dossier for various purposes including research and study by relevant students, and facilitating the used of medical documents. Special considerations should be given to several factors while creating a medical image management system; for instance:

✓ The proposed system should help the diagnosis of disease.
An Architectural Framework for a Medical Image Database Management System

✓ Improving information retrieval is one of the main objectives of information management systems.
✓ The proposed system should have the ability to make connections with other information systems of the hospital so that the medical images such as CT-Scan, MRI and Radiography can be incorporated into the patients’ dossier even though the companies offering the software believe that it is impossible to establish such connections due to the differences in data volume and data disorder.
✓ System implementation should be reasonable and affordable.
✓ The system should conform to standards.
✓ The utilities of software should have the attribute of auto-hide. That is, they should not only be available to the users but also be hidden and not interfere with images.

The architecture of the proposed image database system is multi-layer to allow the facile development and maintenance. The system is a Web-based platform for retrieving any set of image file from any location. It is capable of processing and storing any medical image format like TIFE, DICOM or any common digital imaging. The proposed system includes a layout for arranging database along with a comprehensive architecture for the assessment of diabetic retinopathy image database which can record up to 5000 samples with a proper production program.

2. Related Works

Many medical information systems have been developed since 1964 [6]. Several medical image management systems which have been studies or implemented throughout the world are discussed hereunder.

MEDIMAN

MEDIMAN is a Web information system (WIS) to manage and process medical images which is currently being used for clinical trials and research by neuroscientists and physicians in many medical and research centers in Spain [4].

PACS

PACS is today one of the most advanced medical image management systems, connected to the hospital information system (HIS), which provides the accomplishment of medical profiling. PACS is currently being used in many hospitals in Iran [11].

MIPAS

MIPAS is a Web-based system designed for remotely accessing and presenting DICOM images. It is accessed through a Web browser via internet. MIAPS provides four features including the DICOM image retrieval, maintenance, presentation and output [10].

KMeD

The objectives of KMeD are querying medical multimedia distributed database, creating an alphanumeric content, modeling temporal and spatial files, formulating the queries based on medical terms and concepts, and creating a high-level query language [1]. The first group includes storage resource broker (SRB), supporting a sharing network, which can be distributed to multifaceted organizations and heterogeneous storage systems. It is one of the most famous examples of middleware used for creating a data grid in direct support of using DICOM storage servers. SRB is a general type of middleware for integrating different data sources and is mainly used for committing various PACSs to DICOM resources through a special driver [3]. Furthermore, BIRN is a virtual community specifically used in the field of neurological imaging to develop a protocol for sharing research resources between neuroscientists and medical scientists. The main objective of BIRN is developing a multi-institutional information management system to support biomedical

---

3Picture Archiving and Communication System
4Medical Image Access and Presentation System
5A Knowledge-Based Multimedia Medical Distributed Database System
6Biomedical Informatics Research Network
research. To this end, any participating institution should store a database of their experimental and computational data; then, BIRN applies the conceptual integrative systems to the databases so that the researchers can separately fulfill their data analysis in wider and larger datasets than the available datasets.

The second group is related to LONI Pipeline which is a plain graphical environment for complicated scientific data analysis. The pipeline provides a visual intuitive interface in data analysis and simultaneously allows the distributed applications to interact uniformly. Moreover, it allows the researcher to easily share their data analysis methods and provides the community with a plain platform for the new distributed applications such as updates. The environment also exploits super-computational environments through automatic parallelization of data-independent applications in a data analysis process at any possible time [7].

The third group consists of a set of systems associated with content-based image retrieval. CBIR is an active field of study referring to a set of techniques and algorithms which makes the query of image database possible based on the image content including color, texture, objects, and their geometry instead of textual attributes such as image name or other keywords. CBIR has the potential to provide a powerful resource for helping physicians make accurate medical diagnosis. To name some CBIR systems, QBIC and IBM are known as the commencement of content-based image retrieval [8].

3. Definition of Requirements

There exist various technologies to record medical images and support clinical decision. However, not many efforts have been done to develop intelligent management systems for visual information retrieval. The required system should support the medical workflows including clinical, scientific, and research tasks. Figure 1 presents an overview of the system requirements which should support image capture, storage and methods for similar questions.

3.1. Practical Requirements

There are different methods and devices to record images in a medical center. A medical sample may require only an image or various images. Whatever the number of images is for storage, the system should provide a standard interface to allow any number of images to be uploaded. Furthermore, each image can have various annotations or textual descriptions which should be stored in combination with visual raw data. Like the attributes of DICOM, the system should also allow the incorporation of textual metadata and questions by means of keywords.

---

7Laboratory of Neuro Imaging
8Content-Based Image Retrieval

Figure 1: The Architecture of Image Database System

Once the images are stored, the physicians may explore a set of specific images for the retrieval of simple or special occasions to get an overview of
a general collection. The discovery can be made by metadata through filtering some textual conditions such as the date or quality of image. Since a high number or results may be delivered in the process of exploration, the images should be presented with a certain number of items per page. The details of image should be displayed to the users on request. The system should allow to upload images with pertinent metadata. Considering that medical images are described by certain visual patterns, the system should support displaying a set of images according to a given image pattern to be able to present a set of similar images. Therefore, displaying images by analogy is an appropriate and useful way for physicians to find images with similar patterns. CBIR systems [5] have many merits in medical environments. They are considered as a computer-aided diagnostic tool. The system should provide tools for automatic image processing and, if necessary, specialized image analysis to present images by analogy. It is worth noting that if the analogous images are considered in retrieval process, to what extent the speed of retrieval is decreased and the storage space is occupied.

3.2. Technical Requirements

In designing such an image database system, extensibility and scalability should be taken into account. Extensibility is possible through modular architecture. The system should be able to manage any digital image format and perform needs assessment for any medical specialty after initial medical requirements. For instance, in dermatology, in which the resolution is less important, the most common method is using a digital image, usually saved as JPEG format. In pathology, a digital camera with a microscope allows users to capture images in BMP, JPEG and TIFF formats [4] whereas the standard format used in radiology is DICOM. The important point is that image database system should not be developed as a single-user desktop software. The main features of the system should be implemented in a high-performance computing platform which provides users with simultaneous access via network services. Additionally, since medical information and images should not be exposed to unauthorized access, the system must provide a single authentication security. The system should be designed to allow Web access. However, with regard to the fact that the main applications of medical image database system are to support scientific activities and diagnosis, the process of searching should be simple; that is, the system should consider the retrieval mechanisms in accordance with the model of information storage [2].

4. Medical Care Database

Selecting a database for medical operations is a controversial issue. Considering the properties of each database, a comparison can be made between different database management systems. Both MySQL and Sqlserver databases are suggested for the proposed system architecture. According to [9], MySQL database system can have a better performance than Sqlserver for medical care systems. In addition, MySQL’s open-source feature makes its interior architecture illustrative and clear in forms of major and minor modules for users. The structure of this database consists of different modules for performing different tasks as well as a standard of an all-purpose database engine. Some of the main components in MySQL include:

4.1. Connection Management

The control and restriction in the number of connections between the client and database server is one of the main tasks and modules of MySQL. Indeed, the database engine plays a crucial role in optimal use of available connections by establishing automatic management and sharing between clients. The module tries to avoid a large number of idle connections to the database by creating a shared space in order to be effective in using less memory space and CPU processing.

4.2. Commands and Key Elements

The tasks of deployed modules are creating database tables for indices, views, stored procedures, functions and triggers, and supporting
SQL commands for making these elements or reading, inserting, deleting and modifying their data.

4.3. Memory Management
One of the most important tasks of a database engine is the allocation, management and processing of memory used in the database as well as creation of required space for input and output buffering operations and caching operation to increase information readability speed. These activities are better done in MySQL. Moreover, at the time of creating and maintaining a database, different parameters are adjustable by the systems administrator using MySQL specific tools. These tools have been used for simulation as discussed in the section of assessment in the present study.

4.4. Information Storage Engine
MySQL possess various tolls for information storage in different ways. Therefore, when creating a database, MySQL addresses specific questions about the type of database application, and storage engine. This part is further discussed in the section of assessment.

5. The Proposed Architecture
The architecture of the proposed image database system is based on the commercial version of .NET which offers a multifaceted or distributed model. In this model, each layer has a certain task for facile repair and maintenance, and extensibility. Each layer provides the other layers with different services.

5.1. Components and Layers
The system architectures consisted of four main layers. The strategy which allows the delegation of processing responsibilities to each layer is called database management. The main focus of the architecture in on data layer. As noted earlier, each layer has a certain task and the interaction between the layers will result in a system with all the functionalities.

5.1.1. Client Tier
The client layer includes the graphical user interface (GUI) which allows users to the interact with system and displays images and information. This layer has two clients; a Web client who uses an internet browser to access the system through the Web layer and a standalone client who has a remote application using .NET Remote to access .NET server.

5.1.2. Server Tier
The server layer consists of three sublayers including Web, active directory and management-executive. The Web sublayer has .ASPNET server pages which processes the queries and responses based on the results of commercial layer. It uses a local interface to handle requests. The active directory is the Microsoft’s technology for managing network resources and is managed by a domain controller. It is fundamentally an integrated distributable database which has been developed for windows-based servers. Without active directory, the resource management requires the single management of database in solitary whereas with active directory, resource management is integrated [12]. The management-executive sublayer is, in fact, the core of system and is composed by .NET Library and .NET classes. This sublayer has also a metadata management unit to capture images and extract contextual information such as name, size and features of DICOM.

5.1.3. Data Tier
The data layer provides tools for having access to the file system, in which images and relevant small pictures have been stored, and database containing metadata and features of images. The main purpose of the present study was to design data layer for medical image applications which is associated with the overall architectural design.

5.1.4. Security Tier
The security layer controls the access to applications based on the assigned role by security applications which are configured in a standard XML descriptor. The security model authenticates the role of users and either allows or refuses them
to have access to some functions and domains. The security layer offers three security levels i.e. application security, host security and network security. The application security involves input validation, authentication, access control, configuration management, sensitive data, session management, encryption, manipulation of parameters, exceptions management, auditing and logging add-ons. The host security is associated with patches and updates, services, protocols, files and directories, sharing, ports, registry, auditing and log add-ons. Finally, the network security includes routers, switches and firewalls. All these items were summarized into two groups of Web access and .NET Library.

5.2. Connections
A single system is composed of a set of interrelated components following the same target. The main purpose of designing the proposed architecture is to expedite medical information retrieval. Considering that medical database structure includes textual and multimedia information (i.e. images), the plans should be designed so that they take the shortest time to process the interval between the user’s request and response. The overall architecture is illustrated in Figure 2. The double arrows indicate the interaction between layers and sections.

Figure 2: The Overall System Architecture.

The medical center operator can send queries to the server using the client tier through Web pages, plugins, embedded programs in browser or standalone programs. For instance, if the query is to record an image, the image is, first, uploaded on the server and, then, the features of image are removed. The separated features of image are saved with the patient’s information. As shown in data tier, several centers are interacting with each other. A database whose tables are designed interactionally is placed at the highest level which maintains the recently saved or retrieved data. This database intersects with the other three databases at the second level. The three databases at the second level include FullText DB with textual information, ImageDB with medical images and related samples, and Meta Data DB which maintains the extracted features of medical images. All three databases store a part of the total system data. But they are stored in Static DB as an archive to refer to, if necessary. The arrangement of databases in data tier is displayed in Figure 3.
An Architectural Framework for a Medical Image Database Management System

According to Figure 3, all data have been stored in Static DB. The Static DB maintains its information in one site or several closely related sites. The selected storage engine configured for the tables of Static DB is ARCHIVE. Data are stored in their most compact mode due to the features of ARCHIVE storage engine. Actually, storage uses less space than disc. The amount of space is to the extent that if the storage engine is changed to Inno DB, the used space of the disc will be increased to more than twice as much as the storage for storing table’s data. The configured types of storage engines for Image DB is Inno DB, for Full Text DB is My ISAM and for Meta Data DB is either My ISAM or Inno DB. Selecting My ISAM storage engine for both Full Text DB and Meta Data DB is due to using more speed in textual retrieval or searching. Since Meta Data DB may store the image features of many patients, it would be better to use Inno DB storage engine. Also, Inno DB storage engine is selected for Image DB due to the large size of the stored files. The My ISAM storage engine does not allow the users the storage of more than 4 GB. Therefore, the image tables are built with Inno DB storage engine. The Transactional DB acts as a temporary event-oriented database. On the one hand, it is temporary because a small part of these data are interacting with other layers in this database. On the other hand, it is event-oriented because it gives an appropriate response to the query if necessary. The tables of these databases are configured based on Inno DB storage engine which execute the commands in transaction mode. Transactional DB is an interface between data tier and server tier.

6. Case Study

Two architectures were discussed in the present study. The first architecture was an outline of the relationship between the units whereas the second was in line with the main objective i.e. access to data. Accordingly, achieving a practical application of the proposed architectural approach was sought. To this end, a Windows or Linux server was launched with at least 8 GB hosting space, 4 GB unshared main memory and MySQL database management system. The scripts were written in PHP language applicable to both Windows and Linux servers. The dedicated speed of the processor in the server was Core i5 2.30GH Intel along with 3 GB cache memory. To conduct the test, there required a hypothetical database. To this end, three hypothetical tables were developed for diabetics’ eye impairment. The tables are presented in Figure 4. In the table of Patients, basic information is addressed; Logs presents the accomplished processes and caseHistory stores the main image of each patient in the field of Color Image and other side images in the fields of hardExudates, softExudates, Microaneurysms, and Hemorrhages.
If the Database X was prepared for medical health program, it would be placed in one or more sites to fulfill insertion and retrieval operations. Retrieving or navigating the data is costly. Almost 5000 records were inserted in designed tables. Figure 5 shows the first records of patient’s caseHistory table.

**Figure 4:** Designed Tables from the Database X

<table>
<thead>
<tr>
<th>caseID</th>
<th>patientID</th>
<th>caseDate</th>
<th>color/image</th>
<th>hardExudates</th>
<th>softExudates</th>
<th>Microaneurysms</th>
<th>Hemorrhages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2015-04-10 19:09:24</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2015-04-10 19:09:26</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2015-04-10 19:09:27</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2015-04-10 19:09:29</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2015-04-10 19:09:30</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
<td>18.08</td>
</tr>
</tbody>
</table>

**Figure 5:** The caseHistory Table from Database X

All the images were stored in binary mode in the database. The format of image fields were Blob in tables. Figure 6 displays a binary code sample of Hemorrhages field (one of the four stored side images) for patient’s sample.

**Figure 6:** The Binary Code of Hemorrhages Field in MySQL Workbench.

To provide performance assessment, 6 databases were designed with the same structure. The differences of databases are presented in Table 1.

<table>
<thead>
<tr>
<th>Database</th>
<th>Storage Engine</th>
<th>Inter-Table Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>Db1</td>
<td>InnoDB</td>
<td>Yes</td>
</tr>
<tr>
<td>Db2</td>
<td>MyISAM</td>
<td>No</td>
</tr>
<tr>
<td>Db3</td>
<td>ARCHIVE</td>
<td>No</td>
</tr>
<tr>
<td>Db4</td>
<td>InnoDB</td>
<td>No</td>
</tr>
<tr>
<td>Db5</td>
<td>InnoDB</td>
<td>No</td>
</tr>
<tr>
<td>Db6</td>
<td>InnoDB</td>
<td>No</td>
</tr>
</tbody>
</table>

**Table 1:** The List of Studied Databases

According to Table 1, four databases of db1, db4, db5 and db6 were similar in terms of the storage engine structure. The db4 did not have any inter-table relationship; db5 was partitioned vertically into two tables. Finally, db6 was partitioned both vertically and horizontally. Figure 7 shows the vertical partitions based on which the caseHistory table is divided into two two tables with fewer columns. The speed would be enhanced considerably if Joint was not used and only one table was required.
7. System Assessment

Designing a database needs pre-requisite information like any other tasks. For instance, the transactions, in some processing, demand a secure status for tables. Nevertheless, making such a request requires costly runtime, sufficient memory consumption and disk space. The proposed architecture of data tier did not deal with tables design and their relationship. However, the performance of the conventional model gets improved through new arrangement. The assessment was proceeded based on factors such as consumed space, run speed, transaction support and the overall enhancement of system performance. Moreover, Mysqlslap was used for model simulation since it is the most flexible tool for benchmarking MySQL database and is a new tool in MySQL 5.1.4. The load simulator allows the developer to test a specific query or table’s engine under load conditions. In other words, Mysqlslap has been developed as an identification program for simulating user load in MySQL server and reporting each stage. The parameter of Concurrency is the number of connections or clients which should be simulated during the select command. Without this command, the queries will be run as a single-threaded command. If the command was written as Concurrency = 1, 2, 4, 8, 100, the same test would be increased for different concurrencies in each run.

As displayed in Figure 8, a query was assessed for 10, 100, and 1000 records in 6 databases. Figure 9 shows the command to retrieve 1000 records for db1 with 30 concurrent users.

The command of (P4) in caseHistory Partition refers to the information/data located in Partition 4 obtained from vertical partitioning. A set of 1000 records was retrieved from atotal of 4000 to 5000 records through partition 4 in caseHistory table. In the next stage, two indices were assigned to the fields of name in Patients’ table and main image in caseHistory. Once the assessment was fulfilled, all the items were recorded in Table 2.
An Architectural Framework for a Medical Image Database Management System

Table 2: The Observed Values of Simulation.

According to Table 2, Plan 3, using Archive storage engine, had longer retrieval time which is incomparable with other plans. It has less storage space than other engines. Therefore, it is removed to make a better comparison between the engines; then, the other different plans were compared based on their number of records (Table 3).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Client (10 Rec)</td>
<td>1.421</td>
<td>0.015</td>
<td>1.287</td>
<td>1.281</td>
<td>0.343</td>
<td>0.009</td>
</tr>
<tr>
<td>1 Client (100 Rec)</td>
<td>2.17</td>
<td>0.032</td>
<td>11.28</td>
<td>1.687</td>
<td>1.748</td>
<td>0.187</td>
</tr>
<tr>
<td>1 Client (1000 Rec)</td>
<td>27.072</td>
<td>0.23</td>
<td>-</td>
<td>12.193</td>
<td>16.456</td>
<td>6.182</td>
</tr>
<tr>
<td>30 Clients (10 Rec)</td>
<td>1.141</td>
<td>0.124</td>
<td>-</td>
<td>1.343</td>
<td>0.406</td>
<td>0.39</td>
</tr>
<tr>
<td>30 Clients (100 Rec)</td>
<td>3.638</td>
<td>1.858</td>
<td>-</td>
<td>1.952</td>
<td>3.108</td>
<td>1.687</td>
</tr>
<tr>
<td>30 Clients (10 Rec) + colorImage Index</td>
<td>0.14</td>
<td>0.187</td>
<td>-</td>
<td>0.39</td>
<td>0.421</td>
<td>0.25</td>
</tr>
<tr>
<td>30 Clients (100 Rec) + colorImage Index</td>
<td>2.373</td>
<td>1.936</td>
<td>-</td>
<td>0.874</td>
<td>0.344</td>
<td>1.671</td>
</tr>
<tr>
<td>30 Clients (10 Rec) + Name Index</td>
<td>0.156</td>
<td>0.125</td>
<td>-</td>
<td>1.109</td>
<td>0.421</td>
<td>0.109</td>
</tr>
<tr>
<td>30 Clients (100 Rec) + Name Index</td>
<td>2.17</td>
<td>0.967</td>
<td>-</td>
<td>3.545</td>
<td>1.936</td>
<td>0.624</td>
</tr>
<tr>
<td>30 Clients (1000 Rec) + Name Index</td>
<td>27.775</td>
<td>16.441</td>
<td>-</td>
<td>29.228</td>
<td>19.329</td>
<td>12.318</td>
</tr>
</tbody>
</table>

Table 3: The Comparison between the Capacities of Storage Engines.

As illustrated in Figure 10, plan 2 showed a better performance in concurrent access to single user and 30 users for retrieving 10 records in comparison to other plans. The results indicated that MyISAM had a better performance in low-record retrievals and even in concurrent access. Plan 6, which was partitioned vertically and horizontally, had a relatively good performance after MyISAM. Furthermore, the performance of Plans 1 and 4 was poor. On the basis of the assessment, indexation decreased the run time. Remarkably, indexation had a better performance on image than textual field.

Figure 10: The Comparison between Different Plans for Retrieving 10 Records.

Seyyedeh Leila Seyed Bagheri Roudbari, et al. 462
Figure 11 displays the plans for retrieving 100 records. Accordingly, Plan 6 showed a better performance than Plan 2. Comparing these two plans, one would find out that MyISAM storage engine in Plan 2 had a better performance than InnoDB in the retrieval of lower records for single user load. Furthermore, indexation acts differently in different plans. For instance, In Plan 6, which had a relatively better performance than other plans at this stage, textual indexation or indexation on name showed a better performance than indexation on image.

Figure 11: The Comparison between Different Plans for Retrieving 100 Records.

According to Figure 12, Plan 6 still showed a better performance compared with Plan 2 in retrieving 1000 records. With respect to the concurrent query run for 30 clients in Plans 2 and 6, the concurrent access to database showed, by far, a better performance using InnoDB storage engine than MyISAM storage engine for higher number of users. Furthermore, the indexation on patient’s image and name in Plan 6 acted better than Plan 2. If the single-user section was removed in Plans 4 and 5, the remaining sections would have almost the same performance similar to Plan 2.

Figure 12: The Comparison between Different Plans for Retrieving 1000 Records.

The format of MyISAM in MYSQL table has been optimized to increase its speed and reliability. It runs compression to conserve the disc space; Therefore, the size of stored data is far less in MyISAM than InnoDB. Likewise, the format of Archive requires less space for storage. With regard to these concepts, Figure 3 is explicated. The plan 6, with all its performances assessed, can be selected as the best database which directly interacts with the user. In other words, Transaction DB is database whose main tables have been composed of vertical and horizontal partitioning along with InnoDB storage engine, and its indexation has been applied to name and image fields. Image DB acts like Plan 4. The Plan 4 has a better performance than Plan 1; and if the indexation on its name field is removed, it will show a performance similar to Plan 5. The
partitioning of Plan 5 was vertical which may increase the cost of join. Thus, Plan 4 is the most appropriate plan. Both the Full Text DB and Meta Data DB are completely textual which are compatible with textual compression and text-based retrieval approaches; hence, Plan 2 is the best option. The purpose of Static DB is archiving information. According to the results of Table 3, the Archive storage engine benefits from a considerable compression which is very vital for data archive.

8. CONCLUSION
The present article proposed the design, development and assessment of a medical image database system which reduced the query execution time, increased the speed of information retrieval and enhanced the performance of medical databases. The propose system possesses some features which distinguish it from the conventional image management systems. Its multilayered architecture facilitates the Web access to image collections. The results of system assessment indicated that MyISAM storage engine has a better performance for retrieving information with lower records or asynchronous accesses than InnoDB storage engine. InnoDB shows a better performance in non-relationship mode with higher records than MyISAM. The operations of removing columns and rows, in other words, vertical and horizontal partitioning can help expedite the retrieval speed. There is a direct relationship between indexation and the number of records and field size. The best effectiveness of light retrievals is in placing the index on image. Increasing the number of indexation records on light textual fields would result in better efficiency. In general, indexation optimizes the system. However, in optimal mode, the size of the data to be indexed should be specified in order to determine whether the index can expedite the retrieval speed. Obtaining this threshold value to decide on indexation can be considered as one of the optimization processing for database servers. By comparing several assessments, a series of databases was designed at several layers. Data are relocating between these layers to be accessed on request.

REFERENCES
