Asian Journal of Information Technology 14 (4): 122-128, 2015
ISSN: 1682-3915

Dynamic Database Schema for Hospital Management System

[1]Ali H. Al-Fatlawi, [2]Basheer A. Al-Assadi and [2]Mohammed H. Jabardi
[1]Artificial Intelligence Division, [2]Programming Division,
Information Technology Research and Development Center,
University of Kufa, Kufa, Iraq

Abstract: The transition in the treatment and the hospitals' care from a single process into continuous and recorded progress has highlighted the importance of the clinical management systems. Although, the paper-based recording operations can manage clinical data and save them they are complicated and unreliable. Therefore, computerizing these processes in a central database is crucial in order to extract beneficial information and follow the patient's status. The difficulty in defining requirements of the medical organization previously has brought a difficulty in building a stable system with the conventional database schemas. In this regard, the proposed system challenges these difficulties through presenting a reliable and flexible solution for such organizations to manage their data efficiently by a dynamic database schema that can manage multiple hospitals centrally with high flexibility. For this purpose, Entity-Attribute-Value (EAV) modeling technique has been adopted to build a flexible database with the schema that satisfying demands of the hospitals and their ever-changing requirements. In this approach, data from unlimited number of organizations (hospitals or medical centers) can be added to the system with different entities in the same database. Moreover, centralizing the data repositories of clinical operations has a significant role in enhancing the hospitals' care and improving the treatment's quality in different ways. This research introduces a flexible system that is capable of overcoming the requirements' variation and supporting the research field via building a practical and central data repository for the researchers.

Key words: Dynamic database, entity-attribute-value, EAV, clinical data, management systems, hospitals, medical centers, clinical research, clinical repositories, flexible database

INTRODUCTION

During the recent decades, the technological advances in the medical and bioinformatics fields have highlighted the importance of collecting clinical data which have been considered as key corporate resources in developing clinical researches and medical services. The idea of transferring clinical data into useful information has been promoted to help hospitals in tracking the patients' history and analyzing their treatment progress. The transition in data collection from the traditional methods (paper-based methods) into electronic capturing has benefited these hospitals in saving cost and time through computerizing manual operations. Additionally, digitizing patients' records and sharing them with authorized people (inside or outside the organization) can improve the quality of their services and support activities of the clinical researchers in gathering information and examining them. In this regard, the effectiveness of the electronic management systems relies on their flexibility in satisfying the demands of the users in meeting all of their requirements that can be changing continuously. Recently, there is an increase in the need for releasing flexible, reliable and high-quality systems to manage clinical data because of the crucial role of the clinical research in generating information and valuable statistical from clinical trials. These records could include patients' personal data such as name and age, in addition to their medical history to serve many purposes. Thus, analyzing these data and housing them in real datasets has a significant contribution in supporting the research field. As an example for one of these purposes, evaluating and following the patients' status and their responses for short or long time. In that many researchers and specialists have recognized the significance of the integration in clinical data and building central, flexible and reliable data repositories. Although, many companies and developers have adopted various systems and techniques to satisfy the demands of hospitals and medical centers by using relational databases, most of these systems are not flexible enough to overcome the problem of ever-changing requirements and the continuous needs for improvements and updates. The databases of the conventional hospitals management systems are built to facilitate specific tasks according to

Corresponding Author: Ali H. Al-Fatlawi, Artificial Intelligence Division,
Information Technology Research and Development Center, University of Kufa, Kufa, Iraq

predefined attributes. Therefore, minor updates may bring major changes in the database and they can be expensive for the organizations.

This study introduces a flexible and dynamic hospital management system by using the Entity-Attribute-Value (EAV) modeling technique. With this system, any update or development can be handling without having to expand the original database or change its attributes. For this and other reasons, it is not crucial to define and analyze requirements of the hospital's functions previously because the proposed system is adjustable, adaptable and upgradeable at any time.

Background: Generally, databases can be described as collections of data that are integrated in a logical manner to store and manage information efficiently to fulfill the users' needs (Frawley *et al.*, 1992). The term of clinical repositories has been defined by Johnson (1996) as an accurate representation for the patients' information that are gathered through the treatment procedures and arranged in conceptual schema (Johnson, 1996). In the medical field, Hammond (1980) highlights the difficulty of collecting the patients' data and complexity of constructing and managing their databases. In that, the paper-based data is highly noisy and complicated in extracting knowledge and get useful information. The electronic processing for such data is very important and crucial in order to overcome difficulties of managing these data and analyzing them efficiently through defining a data model (Batra and Sachdeva, 2014). Studies such as Connolly, Begg and Collen have detailed the process of implementation computer-based management systems to replace the file-based systems. In the study, the writer points out that the relational data model seems to be simple in its operations but it can be more complicated with increase the functional relationships between subcomponents of the system during the updates processes. As it explained previously, the development of the healthcare techniques has change the medical treatment from isolated processes into monitored and continuous procedures (Lenza *et al.*, 2007). Thus, many hospitals have adopted various systems and techniques to manage their data and record the patients' status in addition to support the research in the medical field through building a practical and reliable data repositories that can be available for many researchers. In this direction, the study (Lenza *et al.*, 2007) categorized informative systems of healthcare process through distinguishing between the technical and the functional integrations. It described health informative systems but with less regarding the importance of the flexibility in these systems as they assumed that the requirements of such systems were already well known and predefined.

Another way to map the treatment process is a document-based workflow which is demonstrated by Christoph and Richard. This method is based on collecting data about the treatment processes from several institutions that are engaged in the procedure via singular document to represent the whole details. Although, this study details the treatment process and monitors the patients' history electronically, it lacks the centralization and the self-containing in the recording process. The fundamental of the preceding study has been established according to collecting information about the patients from multiple systems which have different cores via exporting information to XML files. Therefore, this requires providing a new mapping to the essential XML format. In conclusion, it is clear from the previous studies that the major challenge in the medical databases is their diversity and homogeneity. Consequently, the notion of the Entity-Attribute-Value (EAV) databases has been promoted recently to introduce an optimal solution to overcome all of the problems and the difficulties that have been indicated in this study.

The concept of the Entity-Attribute-Value (EAV) Model is introduced to present a solution for the highly heterogeneous data that are arranged in related tables. The EAV databases can substitute the conventional modeling for the databases through housing various types of data that are generated from diverse sources in few tables comparing with the traditional relational method (Dinu and Nadkarni, 2007; Brandt *et al.*, 2002). With the EAV modeling approach, each row in a table refers to a fact and represents an entity (Dinu and Nadkarni, 2007) and it has a key value that identifies its entity and it has attributes to contain the descriptive data. The proceeding method aims to simplify the modeling of the databases which have large number of attributes and to handle them efficiently. As the databases of the hospitals have millions of parameters and records that are accumulated over years, the relational databases can be incompetent in managing this vast number of records. Furthermore, this can cause problems in upgrading the system's structure and satisfying the new demands of the medical field. Therefore, this study adopts the EAV modeling technique to build a dynamic, flexible and reliable solution to handle the medical data of multiple departments for one or many hospitals and medical centers.

MATERIALS AND METHODS

Data structure: This study presents a web-based system to manage one or many hospitals or medical centers with a dynamic database centrally. This database has been built and structured according to the Entity-Attribute-Value (EAV) Modeling Method. It is

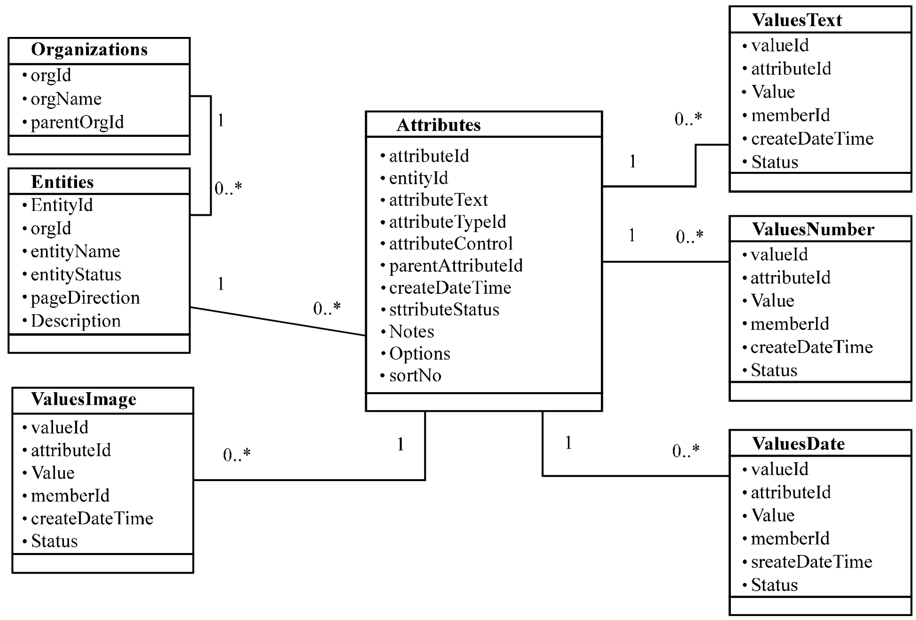

Fig. 1: The physical schema of the proposed database

described as a dynamic because of its flexibility toward any expansion or development. An authorized user or an administrator of a hospital can add new portions to the system within the structure of his organization. He also can add, remove and update attributes of his sections easily without affect divisions of other organizations or sub-organizations. The construction and the reconstruction processes in the system's structure are implemented virtually through adjusting data inside the database. Therefore, this will keep the physical schema of the origin database without any change.

The physical schema of the proposed database is shown in Fig. 1. As a logical construction, it starts with the class 'organizations' that is included to grant the system its ability in managing multiple hospitals and medical centers with a united database. The class 'organizations' identifies each organization (hospital, center or department) in the system. In this class, the attribute 'orgId' is a unique identifier for each organization and sub-organization in the system while the identifier 'parentOrgId' refers to the parent-organization of that center or department. In the other words, this field defines the sub-organizations and their parent-organization. In this field, the value "0" means that this structure does not have any parent (highest-level organization) while any other integer number refers to its parent. In order to explain this idea, Table 1 shows an example of rough data for three organizations (Florida Hospital, Washington Hospital and Sanford Medical Center) which has multiple centers and departments. In this example, Florida Hospital, Washington Hospital and Sanford Medical Center are main organizations and do not have any parent because the field 'parentOrgId' has the value '0'. On the other hand, "Heart diseases department" is sub-organization because the field 'parentOrgId' has the value "1" which refers to the organization 'Florida Hospital' as a parent. In the like manner, "Center for physical therapy" is part of 'Florida Hospital'; "Children department" is part of Sanford medical center and so on. In the other words, the organization "Florida Hospital" is the parent of the two sub-organizations "Heart diseases Dept." and "Center for physical therapy" and this is identified in the field 'parentOrgId' of the two sub-organizations.

Table 1: Example data for the 'Organizations' class

orgId	orgName	parentOrgId	Description
1	Florida hospital	0	Hospital
2	Washington hospital	0	Hospital
3	Sanford medical center	0	Medical center
4	Heart diseases department	1	Department in Florida Hospital
5	Center for physical therapy	1	Center in Florida Hospital

Each organization (whether it is a parent-organization or a sub-organization) has zero, one or many entities that are assigned in the class 'Entities'. This class includes general information about the entities and their status (active, suspended or removed). The administrator can suspend an entity temporarily or remove it from its page by update the attribute 'status' to be "suspended" or "removed" instead of "active". He also can set the direction of the desired page to be "Left to Right" (LTR) or "Right to Left" (RTL) according to language of the required page. The data-type of attributes can come with different formats according to the type of their values (string, number, date or image file). Therefore, their values are grouped according to their data-type into four different classes: 'ValuesNumber', 'ValuesText', 'ValuesDate' and 'ValuesImage'. Then, the string values will be stored in the class 'ValuesText' while the numerical values in the class 'ValuesNumber' and so on. In additions to the values, these four classes include some details about the attributes such as their status (active, removed, suspended) and their creation date. As shown Fig. 1, the class 'Attributes' is of particular interest as it is the central class in the database and the associative unit between the attributes and their values. This class suits the situation where each entity can be connected with its attributes and then with its values and it permits relationships between them. Then, the structure of each entity is specified through the included attributes and their values in the associated classes. Finally, Table 2 illustrates a dictionary for attributes of the class 'Attributes' to presents a short description about each one.

One link with a dynamic schema: The idea of designing client side page with one link (URL) is to provide a simple way to access the system without generating a link for each entity. Thus, the content of that page and its attributes are changing dynamically according to the related entity. This can be considered as another feature to grant the system more flexibility in the design and implementation phases. For this function, the login process defines attributes and operations of that entity to provide the client side page with a suitable interfacing. As a result, it is not crucial to understanding and analyzing the requirements previously to set the links. Figure 2 illustrates the general structure of the system and identifies the flow of the data starting from the process of accessing the database until to granting a particular design to the client side. With the EAV Method and the singular URL technique, the knowledge that are stored in the database generate the schema of the client page and

Table 2: A dictionary for the attributes of the class 'attributes'

Attributes	Data type	Description
attributeId	Integer	Unique identifier for each attribute
entityId	Integer	Identifier of the entity that the attribute is belonging to
attributeText	String	The text that appear at the user's page to represent the attribute
attributeTypeId	Integer	Data type of the attribute such as number, string, date and so on
attributeControl	String	Type of the required control for the user's page such as check-box text-box or list-box
Options	String	Options that user can choose from them if there are specific choices for that attribute and it can be null
Notes	String	General notes to describe the function of that attribute
parentAttributeId	Integer	If this attribute is part of another, this will identify its parent (its main attribute)
createDateTime	Date time	The date and time of creating this attribute
attributeStatus	String	Status of the attribute (active, suspended or removed)
sortNo	Integer	The sequence of showing the attribute on the user's page (its sequence)

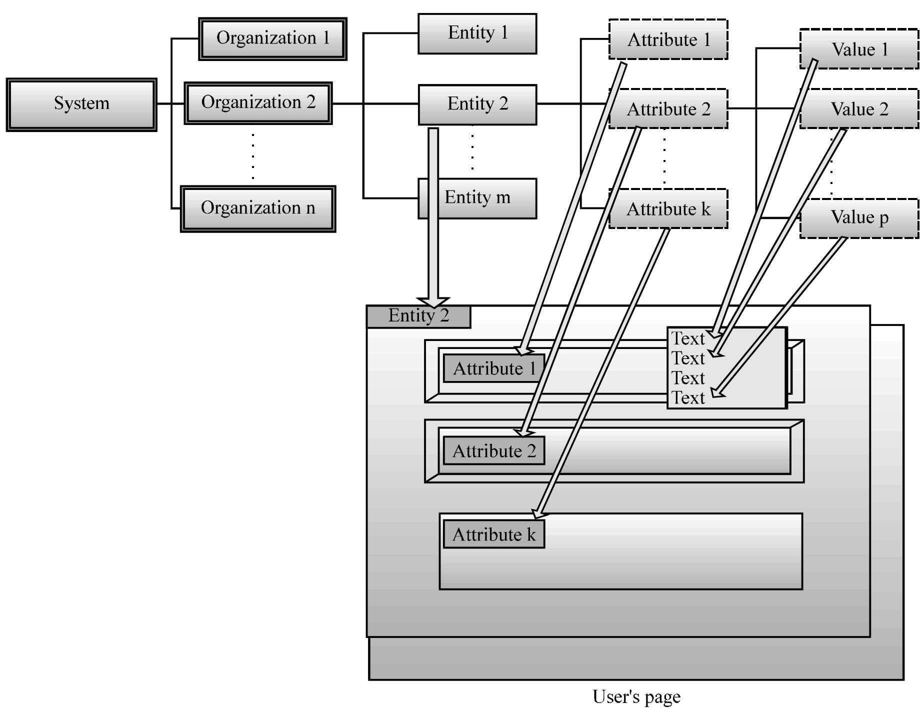

Fig. 2: General structure of the proposed system

feed it. In the other words, the entities and the attributes are connected and used in conjunction to represent structure and values of the entities.

Example of implementation: The proposed system has been implemented and tested with a dataset for a medical center. This dataset includes records for several entities spread over groups of attributes. The database has been implemented by using Microsoft SQL Server 2008 (MSSQL) and programmed as Asp.net webpages.

For privacy reasons, the name of the medical center and names of its patients have been hidden. In this study, the center has been called as "MKV" as a reference. Firstly, MKV medical center is independent and does not have any parent organization as it identified in the field 'parentOrgId'. It has two departments: "Dental Dept." and "GP Dept.". Therefore, the data of the class "Organizations" are shown in Table 3. As illustrated in this table, the MKV organization is a parent for these two departments.

In this organization, the database contains information about two entities: 'patientsInfo' (EN0001) and 'patientsFeedback'(EN0002) as shown in Table 4. The first entity 'patientsInfo' (EN0001) contains the following fourteen attributes: 'patientName', 'weight', 'height', 'gender', 'birthdate', 'maritalStatus', 'isDrinking', 'isSmoking', 'homeAddress', 'phone', 'emergeContaName', 'emergeContaPhone', 'allergies' and 'bloodClass'. Secondly, the entity 'patientsFeedback' (EN0002) has been created to record the patients' feedback and it contains the following five attributes: 'treatmentQuality', 'meetPateintsNeeds', 'patientAdvices', 'recommedToOthers' and 'comments'. As a result, the general structure this system will be arranged as shown in Fig. 3 and the data distribution is shown in Table 5-8.

Table 3: Data of "Organizations" class for MKV medical center

orgId	orgName	parentOrgId	Description
1	MKV	0	Medical center
2	Dental dept.	1	Dental department
3	GP dept.	1	General practitioner department

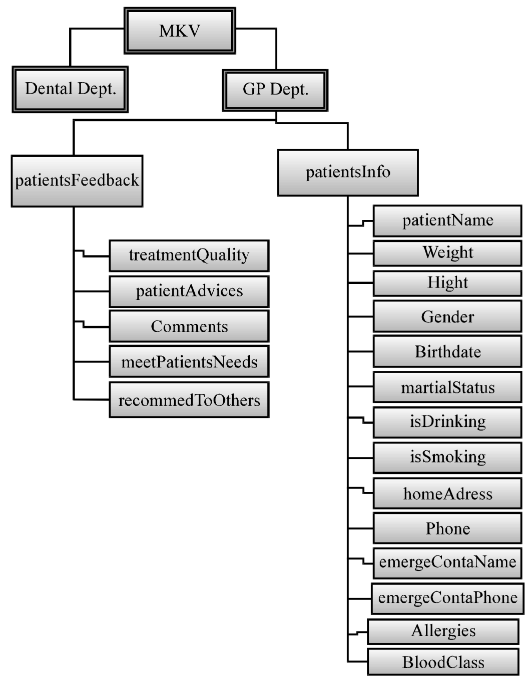

Fig. 3: Structure of the identified example

Table 4: Data of "entity class" for MKV medical center

entityId	orgId	EntityName	Status	pageDir	Description
EN0001	2	patientsInfo	Active	LTR	Entity for doctors' information
EN0002	2	patientsFeedback	Active	LTR	Entity for patients' feedbacks

Table 5: Sample of data for 'attributes' table in MKV medical center

attributeId	entityId	attributeName	Attribute TypeId	attributeControl	Options	Parent attributeId	creationDate	Status	sortNo.
1	EN0002	treatmentQuality	1	Combobox	Excellence, good, average, bad, too bad	0	4/1/2015	Active	1
2	EN0002	meetPateintsNeeds	1	Combobox	Yes, no	0	4/1/2015	Active	2
3	EN0002	patientAdvices	1	Textbox		0	4/1/2015	Active	3
4	EN0002	recommedToOther	1	Combobox	Yes, no, maybe	0	4/1/2015	Active	4
5	EN0002	Comments	1	Textbox		0	4/1/2015	Active	5
6	EN0001	patientName	1	Textbox		0	4/1/2015	Active	1
7	EN0001	isSomking	1	Combobox	Yes, no	0	4/1/2015	Active	2
8	EN0001	Weight	2	Textbox		0	4/1/2015	Active	3
9	EN0001	Height	2	Textbox		0	4/1/2015	Active	4
10	EN0001	isDrinking	1	Combobox	Yes, no	0	4/1/2015	Active	5
11	EN0001	Gender	1	Combobox	Male, female	0	4/1/2015	Active	6
12	EN0001	Birthdate	3	DateTimePicker		0	4/1/2015	Active	7
13	EN0001	maritalStatus	1	Combobox	Single, married, divorced, widow, other	0	4/1/2015	Active	8
14	EN0001	homeAddress	1	Textbox		0	4/1/2015	Active	9
15	EN0001	Phone	1	Textbox		0	4/1/2015	Active	10
16	EN0001	emergeContaName	1	Textbox		0	4/1/2015	Active	11
17	EN0001	emergeContaPhone	1	Textbox		0	4/1/2015	Active	12
18	EN0001	Allergies	1	Textbox		0	4/1/2015	Active	13
19	EN0001	BloodClass	1	Combobox	AB, B-, B+, A+, A-	0	4/1/2015	Active	14

Table 6: Sample of data for "ValueText" table in MKV medical center

valueId	memberId	attributeId	attributeValue	createDate
1	5003	1	Excellence	8/1/2015
2	5003	2	Yes	8/1/2015
3	5003	3		8/1/2015
4	5003	4	Yes	8/1/2015
5	5003	5	-	8/1/2015
6	5003	6	p1	8/1/2015
7	5003	7	No	8/1/2015
8	5003	10	No	8/1/2015
9	5003	11	Male	8/1/2015
10	5003	13	Married	8/1/2015
11	5003	14	Iraq-Najaf	8/1/2015
12	5003	15	964-xxxxxxxxx	8/1/2015
13	5003	16	P2	8/1/2015
14	5003	17	964-xxxxxxxxx	8/1/2015
15	5003	18	No	8/1/2015
16	5003	19	A+	8/1/2015
17	5004	1	Good	8/1/2015
18	5004	2	Yes	8/1/2015
19	5004	3		8/1/2015
20	5004	4	Yes	8/1/2015
21	5004	5	-	8/1/2015
22	5004	6	P3	8/1/2015
23	5004	7	No	8/1/2015
24	5004	10	No	8/1/2015
25	5004	11	Female	8/1/2015
26	5004	13	Single	8/1/2015
27	5004	14	Iraq-Karbala	8/1/2015
28	5004	15	964-985546565	8/1/2015
29	5004	16	ppp	8/1/2015
30	5004	17	964-654864645	8/1/2015
31	5004	18	No thing	8/1/2015
32	5004	19	O+	8/1/2015
33	5005	1	bad	8/1/2015
34	5005	2	No	8/1/2015
35	5005	3		8/1/2015
36	5005	4	No	8/1/2015
37	5005	5	-	8/1/2015
38	5005	6	P4	8/1/2015
39	5005	7	Yes	8/1/2015
40	5005	10	No	8/1/2015
41	5005	11	Male	8/1/2015
42	5005	13	Single	8/1/2015
43	5005	14	Iraq-Bagdad	8/1/2015
44	5005	15	964-xxxxxxx	8/1/2015
45	5005	16	p	8/1/2015
46	5005	17	964-xxxxxxxx	8/1/2015
47	5005	18	No thing	8/1/2015
48	5005	19	AB	8/1/2015

Table 7: Sample of data for "ValueNumber" table in MKV medical center

valueId	membertId	attributeId	attributeValue	createDate
1	5003	8	65	8/1/2015
2	5003	9	170	8/1/2015
3	5004	8	70	8/1/2015
4	5004	9	165	8/1/2015
5	5005	8	58	8/1/2015
6	5005	9	163	8/1/2015

Table 8: Sample of data for "ValueDate" table in MKV medical center

valueId	membertId	attributeId	attributeValue	createDate
1	5003	12	5/2/1980	8/1/2015
2	5004	12	4/26/1993	8/1/2015
3	5005	12	6/11/1986	8/1/2015

RESULTS AND DISCUSSION

Flexibility: The significant idea that has been adopted in this study is managing clinical data of hospitals without a need to define the system's requirements upfront. Therefore, the database has been optimized to present high flexibility in the construction and reconstruction processes which can overcome the problem of the ever-changing requirements. In this approach, there is no arbitrary limit on the number of attributes for each entity. Besides, with grow the database, any number of attributes can be added without affect the schema. Therefore, this design has an important contribution in saving cost and time for the medical organizations and building a sophisticated and flexible system that can manage data of these organizations for a long time and record the treatment procedures of the patients and their responses.

The centralization and data integrity: Another advantage of using EAV modeling technique in managing the hospitals' and medical centers' data is its simplicity in integrating the data from multiple sources. With the proposed system, many hospitals and medical centers create their entities in the same tables without affect each other. This results in creating one universal and central database with few classes for many organizations. Creating a central database containing data of many hospitals and medical centers can provide real data repositories to be accessible by the researchers who are interesting in the medical and bioinformatics fields. Therefore, this has a considerable role in supporting the research field through recording the patients' responses and their treatment processes. This facilitates many operations for researchers such as data-mining operations.

Efficiency of processing highly sparse data: This approach can be an optimal solution for databases that have very large number of classes and numerous classes have a very modest number of instances. With the highly sparse data (heterogeneous data), the proposed system is very efficient in optimizing the storage because there is no need to reserve space for attributes whose values are null.

However, querying data is more complicated compared with the relational databases. Despite this fact, the system's simplicity in the data integration and its flexibility in the updates and upgrades processes offer significant advantages over the conventional and contribute in maintaining the data and keeping them without any lost.

CONCLUSION

The broad investigation for previous studies and management systems shows that the Entity-Attribute-Value (EAV) modeling technique is an optimal solution to manage clinical data. The main concern in this article is the flexibility and reliability of the system in managing clinical data and its wellness to cope the ever-changed requirements and the continued needs for enhancements and development. This study proposes a dynamic database to manage data of hospitals and medical centers. The dynamic schema of the database in the presented system allows the administrators to model any entities easily even if they do not have any programming experience or background. This approach demonstrates very high flexibility with different types of data management systems. It also allows storing data from various sources a compact way and with high efficiency in processing highly sparse data.

REFERENCES

Batra, S. and S. Sachdeva, 2014. Suitability of Data Models for Electronic Health Records Database. In: Big Data Analytics. Srinivasa, S. and S. Mehta, (Eds.). Springer International Publishing, New Delhi, India, ISBN: 978-3-319-13819-0, pp: 14-32.

Brandt, C.A., R. Morse, K. Matthews, K. Sun and A.M. Deshpande *et al*., 2002. Metadata-driven creation of data marts from an EAV-modeled clinical research database. Int. J. Med. Inf., 65: 225-241.

Dinu, V. and P. Nadkarni, 2007. Guidelines for the effective use of entity-attribute-value modeling for biomedical databases. Int. J. Med. Inf., 76: 769-779.

Frawley, W.J., G.Piatetsky-Shapiro and C.J. Matheus, 1992. Knowledge discovery in databases: An overview. AI Magazine, 13: 57-70.

Hammond, W.E., 1980. The growth of medical information systems in the United States. Health serv. Res., 15: 314-315.

Johnson, S.B., 1996. Generic data modeling for clinical repositories. J. Am. Med. Inf. Assoc., 3: 328-339.

Lenz, R., M. Beyer and K.A. Kuhn, 2007. Semantic integration in healthcare networks. Int. J. Med. Inf., 76: 201-207.