Moving from Heterogeneous Data Sources to Big Data: Interoperability and Integration Issues

Mohamed Osman Hegazi[1]
College of Computer Engineering and Sciences
Prince Sattam Bin Abdulaziz University
Al-Kharj, Saudi Arabia

Dinesh Kumar Saini[2], Kashif Zia[3]
Faculty of Computing and Information Technology
Sohar University, Sohar
Sultanate of Oman

***Abstract*—Heterogeneous databases are now facing an emerging challenge of moving towards big data. These databases are adhoc polyglot systems, complex, and NoSQL tools which are semantically annotated. Integration of these heterogeneous databases are becoming very challenging because big data analytics is integrating human and machines contexts. In this paper, an attempt is made to study heterogeneous databases and their interoperability issues and integration issues, and their impact on analysis of data. The data science has grown exponentially and a new paradigm has emerged which is of integration of heterogeneous data to big data. Information, knowledge and decision making become easier but the size of databases has grown and it became big data.**

Keywords*—Heterogeneous databases; interoperability; integration; big data; analytics and intelligence

I. Introduction

This paper presents a new novel approach for integrating database systems and unifying processing while preserving heterogeneity among databases. Instead of using interface techniques this approach uses database connection techniques, which is more flexible and available. Since 1980th till now, several heterogeneous distributed database models have been developed. The early models, the models that were generated before 1990th, such as R* [1] SDD-[2], SIRIUS-DELTA [3], Distributed INGRE [4], ADDS [5], IMDAS [6], and MERMAID [7], presented good ideas, but they were never commercially successful [8; 9]. The new models, designed after 1990, such as Mariposa [10], DB2 UDB [11] and Sybase for extensible data management [12], in additional to the new generation of the early models, are trying to realize the value of the new advantages of the computers, the networks, and the DBMSs. They are also providing heterogeneous distributed systems and unified processing, which have become more complex, because of the continuous growth of several commercial hardware and software products.

1) The limitation of the current models: Most of these models focus on providing heterogeneous distributed systems under certain components. No model can provide standard approach for developing the distributed system under the rapid changes we face today. AS we can see there is a large body of applications, data, and enterprises that run on variety of computers and networks, which cannot be unified by a model that is designed to solve a certain kind of heterogeneity under a certain concept i.e. a limited solution for a global problem. So these models cannot be considered as standard solutions for the distributed processing unless all the heterogeneous and the unified components are static, which is not the case in the situation we face today.

In a sense most of all these models are projects that were generated to satisfy a certain purpose; so they look like an independent particular solution, which cannot be extended to cover integrated systems.

Most of these models do not address the principles introduced by distributed database such as fragmentation, and replication techniques.

2) The gap between research and application: Researchers have developed techniques for handling distributed problems. A large body of research work has been produced in areas such as distributed queries processing and optimization (as example: [13; 14]); distributed location (as example: [15; 16]); distributed replication (as example: [17; 18; 19]); and distributed concurrency control (as example: [20; 21]). Today the situation became more complex; a large body of enterprises has been distributed on the networks, which generate tremendous amounts of data. Applications and computerized products are generated so fast and distributed on different sites using variety of software and hardware, which provide difficulties in compatibility and load balancing among various data marts, especially as the number of systems is growing continuously. This situation causes more complexity in heterogeneous and unified processing. Furthermore, new problems are coming up, the problems of cocupling and integrating data, enterprises, and information, which can be difficult for the existing models to solve. Therefore, new approach is needed to handle this situation. The approach which, can build a heterogeneous distributed models that can provide a standard solution to the distributed processing, the solution which can work on all kinds of data, application, and enterprises, with ability to integrate several systems and enterprises, and ability to facilitate the continuously growing computing advantages and complexity.

Thus, this paper is to present new approach for the heterogeneous distributed database systems integration and shows the role of this approach in moving from Heterogeneous Data Sources to Big Data, the approach:

- acts as a standard approach for developing integrated distributed system under complexity and heterogeneous components,
- provides unified processing,
- permits the use of distributed database techniques such as fragmentation, replication and allocation.
- provides the heterogeneous distributed database capabilities such as schema integration, distributed query processing, and distributed transaction management.
- and provide a frame work for applying big data processing.

II. The Proposed Approach

This section describes the proposed two Approaches for Heterogonous Distributed Database Systems Integration **(**AHDDI**)**, which are top down and bottom up approach.

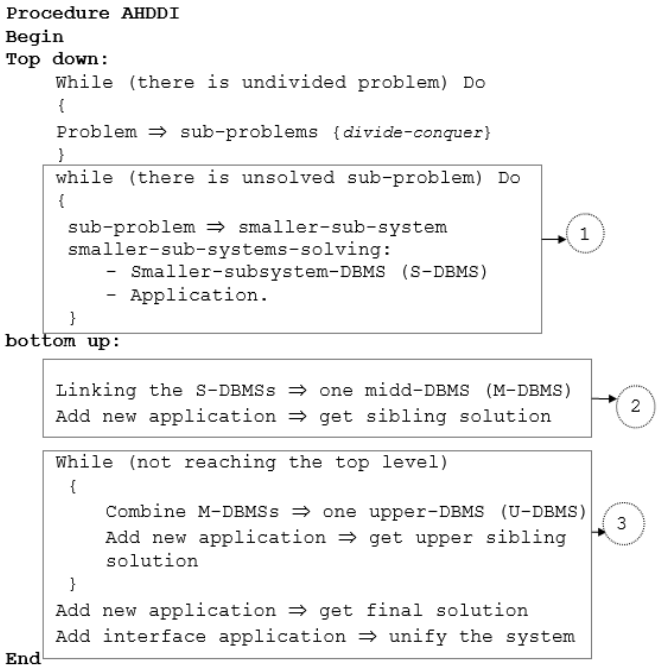

Fig. 1. The Methodology.

The AHDDI algorithm:

Figure (1) illustrates the general idea of AHDDI algorithm. It gives the logical process and it explains the use of DBMS in linking the system components. While the numbers (1), (2) and (3) refer to the steps in which the distributed database techniques are provided, hence in (1) vertical fragmentation and allocation are provided, in (2) horizontal fragmentation and replication are provided, and in (3) hybrid fragmentation and replication are provided.

III. Divide-Conquer Algorithm

Divide and conquer approach is applied, where AHDDI splits the problem into several smaller sub-instances. Then it independently solves each of the sub-instances and then combines the sub-instance solutions yielding a solution for the original instance.

Divide-Conquer algorithm guarantees that a final solution is obtained by combining the independent solutions of the components. [22] [23]

AHDDI uses the following Divide-Conquer algorithm:

Re Divide an instance of size n into r sub-instances each of size n/k. (where k is the divide factor)

Solve these sub-instances.

Combine these sub-solutions.

So the recursion algorithm can follow this rule (*equation 1*):

$$a_n = ca_{n/k} + f(n) \quad c \neq k \text{ [22][23]} \tag{1}$$

n : the size of the problem.(total number of the parts).

an : total number of steps needs to obtain the n-parts.

c : is a constant indicating the weight of the step

k : the divide factor (n/k: the size of the part).

f(n) : the function(the problem).

For example: if we want to divide each instance into 2 sub-instances (k=2), and if we have certain problem (f(n)=1) with equal levels (c=1) then our equation can be as follow (*equation2*) :

$$a_n = (2a_{n/2}) + 1 \tag{2}$$

We can recursively solving this problem as follows:

$$(2a_{n/2}) = [2^2 a_{n/2^2}] + 2$$

$$[2^2 a_{n/2^2}] = 2^3 a_{n/2^3} + 2^2$$

$$2^{k-1} a_{n/2^{k-1}} = 2^k a_{n/2^k} + 2^{k-1}$$

Assuming large n and summing:

$$a_n = na_1 + \sum_{i=0}^{k-1} 2^i$$

According to Divide-Conquer Rule if we put $a_1 = 1$ as initial conditions then: [23].

$$\text{Then } a_n = n + 1 \tag{3}$$

Which is means we need n+1 steps for recursively dividing instants of size n to n sub-instances with the same size of parameters. This means n+1 tasks are required to solve problems of size n (*equation 3*)

IV. Software Architecture

AHDDI is generated using the bottom up technique. This is done by using divide-conquer rule to reach the lowest level systems (smallest subsystems). When we reach the smallest subsystems then we start working in these subsystems, which are located in bottom of the system, and then go up combining

subsystems to obtain finally the whole system. All subsystems at the same level, which have the same parent, are combined to automatically obtain the solution for the parent system. Two steps did the combination of these systems (figure 2).

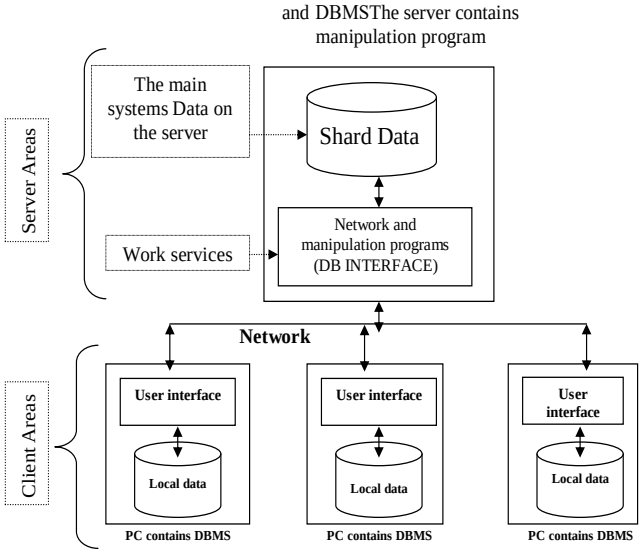

Fig. 2. AHDDI First Level Structure.

V. AHDDI Data Replication

It is important, for some applications, to replicate the data in different sites. For example, when we apply the AHDDI in the education environment we find that the academic subjects and syllabuses are needed to be used in different systems, such as, the students' academic record, the departments information system, and the registrations system, while these kinds of data are not frequency updated it will be better for the system to replicate these data in each user's local machine. Replication can be one of the successful factors in distributed models.

It is easy for AHDDI designer to implement all replication mechanisms (Eager or/and Lazy) because no more than one user can (simultaneously) create or update the same data. This is due to the fact hat AHDDI implements the bottom-up technique built on divide conquer rule.

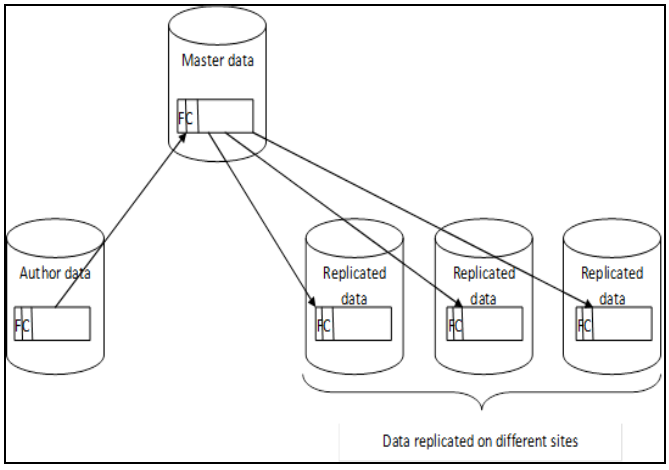

Fig. 3. AHDDI Data Replication.

AHDDI recommended that: any static data, not frequently updated data and archive data, must be replicated on any system that uses this data. This replication can be handled easy because; in AHDDI any data is actually master data in only one subsystem constitutes assistant data to the other subsystems. As example in educational system the academic subjects are master data in academic subject subsystem and its assistance data for all the college sub system such as, the timetable and the registration system. Therefore, this data is administrated by one user or one subsystem administrator and the replication of this data on other subsystem can be done by putting the master data on the global area and make a copy of it on each other subsystem data. So the author user's transaction can then access this data directly from its local machine (figure 3).

VI. Application Layer for Global Processing

The main function of this layer is to map the main system data on the subsystems machines, so this layer is available on each subsystem server, and through this layer the AHDDI queries can make access to main system data.

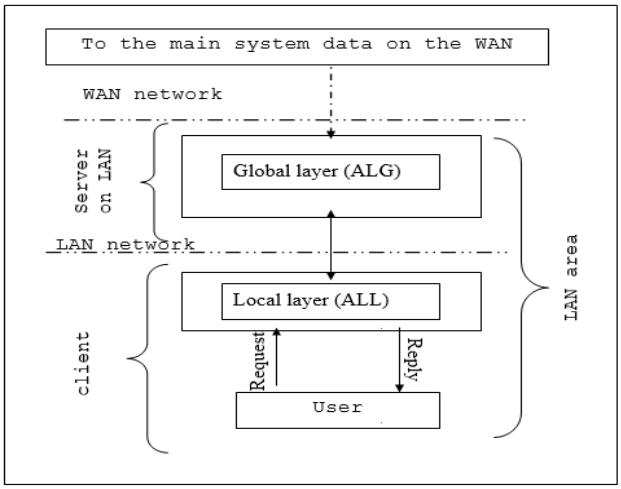

Fig. 4. AHDDI Communication.

```
{Example: linking Oracle DBMS}
Private cn As ADODB.Connection {ADO connection unit}

Public Sub condb()
{Connect to Oracle database}
Set cn = New
ADODB.Connection With cn
        .ConnectionTimeout = 3 { the wait time till the  30 sec. original time is finished}
        .CursorLocation = adUseClient
End With
{open table, using the ODBC DSN }
cn.Open "DSN=ORACLE;UID=SCOTT;PWD=tiger;"
End Sub

{Example: linking SQL-Server DBMS}
Private Sub Form_Load()
Dim cn1 As Connection
Dim r1, r2 As Recordset
Dim s1 As String
Set cn1 = New Connection {Connect SQL-
Server DBMS } cn1.Open "DSN=pubs"
{Specified an ODBC DSN } Set r1 = new
Recordset
r1.Open "Emp", cn1 , adOpenForwardOnly, adLockReadonly
{open table} do Until r1.EOF   s1=r1("name")
        …
        …
loop
End Sub
```

Fig. 5. Data Linking Algorithm (algorithm that used for linking the two layers and make data available in the place of the user application).

The application layer for global processing establishes a link to the main system's DBMS by bringing the name of a server (directory) on each subsystem server and letting the location of the main system DBMS available to each AHDDI queries (figure 4).

These two layers cause limitation to the work space and allow the manipulation of the queries looks like one machine process (transparent), hence these two layers make all the AHDDI data be available in the place of the user application. The algorithm that used for handling this process is illustrated in figure (5).

VII. Big Data Architecture

There was a shift in paradigms from RDBMS to AHDDI. Now data management and processing becoming more challenging because of nature of data, type of data, volume of data, and data become big data. Now complex NoSQL tools with semantically annotations are becoming popular to handle this big data. Architecture and design of databases are changing accordingly with technologies available to handle big data. Data variety, volume and access pattern, data migration from existing systems and integration in the new systems also very challenging task. Data analytics and analytical tools need to be integrated in this big data architectures and design because data modelling and management need them consistently.

Data Acquisition, storage, and analysis of huge data is becoming very complex and expensive process so most of the big companies working on their project called big data. It provides solution in terms of data value, velocity, variety and volume. Requirement of running web services over cloud has tremendously increased and it also facilitated the growth of big data. The sources of data generation is increasing like IoT and it compels big data platforms to grow. MPI, MapReduce and Dryad are the major big data tools that used for handling big data and the comparison will make it clear which tool is better than the other. Deployment, Resource management, scheduling, level of programming support, data storage, task handling, portioning, communication and fault tolerance are some of the issue which are solved by big data tools. In figure (6), an example of Oman is given, it shown how various heterogeneous data sources creating data and now it is moving towards Big Data.

TABLE I. Various Components of Big Data

Value Traditional data service Resource Pool Numeric Data types Content format Data **consumers**	**Velocity** Inquiry analysis Data Collections Analysis Processing
Variety Text, graphics, image, sound and animation data Real time Batch Oriented On demand	**Volume** Virtual Resource Pool Distributed Storage Meta Data Master Data Historical Data

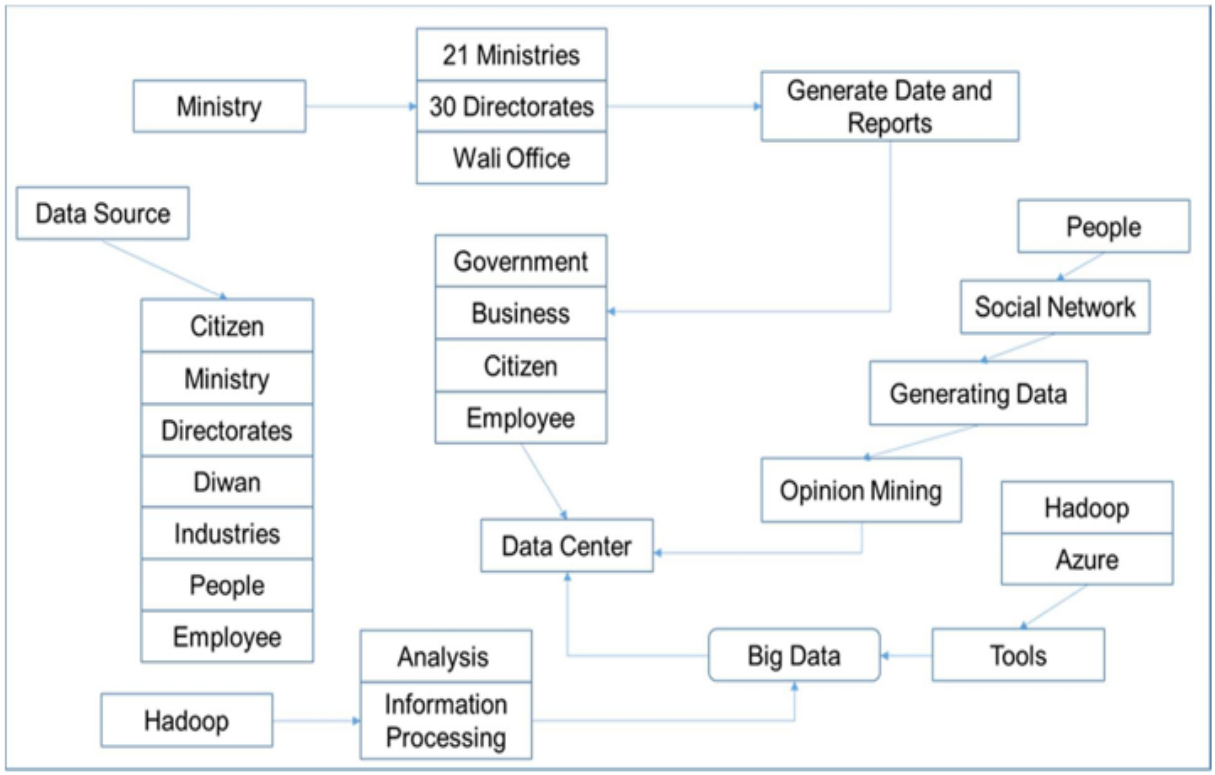

Fig. 6. From Heterogeneous Databases to Big Data (Data Origination and Management).

Figure 6:Data Origination and Management from Heterogeneous Databases to Big Data

Table 1 given below shows various components of big data and various operations happening in these components

VIII. Results and Discussion

It is proposed to use the DBMS to combine these sub-tableinstances, and because it start from the bottom of the system, fragmentations are generated automatically. Thus, the top systems databases carry the combined data that accumulated from several sub systems. Accordingly, in our model all fragmentations are obtained within the model methodology. Moreover, this is done without any conspicuous difficulties. In this approach the work is handled on the smaller subsystems, in the bottom of the system, and each one of these subsystems concern a specific task. Thus most of the queries are processed in the local machine, only the updating process and some synchronous work affect the remote data. Therefore, these techniques minimize the network traffic. In this approach only the owner subsystem is allowed to update its data, because each task is handled by a particular subsystem. This results in the simplification of the query processing and in the elimination of the updating problems. It is demonstrated that this approach can provide both kinds of the replication mechanisms, Eager and Lazy. In addition to that we can combine these two replication mechanisms. This provides more flexibility in AHDDI models, by applying the suitable technique for the suitable data. In this paper we propose that this approach can build an expandable model. The expandable model that can add new systems or application to the existing model or/and can be developed using ready application.

This approach can solve the integration problems by: combining applications of organization and result unified system and by the ability of integrating applications that do not belong to subsystems. This combination provides integration of the organization data as well as the integration of the applications because this is done by using the DBMS as tools to link the application data. This will guarantee that the new technologies are provided on these combined applications because this methodology can be applied as long as there is DBMS and there are protocols that can connected several DBMS, to provide heterogeneous database system, such as ODBC and JDBC , which is make avail of several kind of DBMS. An integrated model with integrated data. An information network, by linking more than one information system to form a WAN. The traditional data was kept in files, then it moved to networked, then hierarchal, then relational, object oriented, object relational, warehouses, then mining now it is becoming big data. Processing, values, consumers, data sources all changed with due course of time. Now big data is real time processing, on demand and continuous.

IX. Conclusion

In this paper, we claim that AHDDI is a suitable model for generating information networks of various sizes. Examples of such networks that arise in practice are: unity between universities, national library system, public sector departments, and national or international organizations. Hence, using AHDDI technique it will be easy to link any kind of application's database in one WAN network, and then users can analyzing and accessing this data using any big data tools, which is indicate that this approach can be a suitable way for moving from heterogeneous data sources to big data.

The paper presents a conceptual solution to the movement from heterogonous database to big data, that can be used for solving difficulties of moving from databases to big data, and whether databases can address the big data problems. Most of those who have taken this aspect believe that databases cannot handle massive data problems for example [24] and [25], therefore most of the previous solutions try to provide new database that can be applicable to handle bid data problems such as [26] and [27]

References

[1]- Thomas, G. Thompson, G., Chung, C., Barkmerer, E., Carter, F., Templeton, M., Fox, S., and Hartman, B. 1990. "Heterogeneous Distributed Database systems for Production use". ACM computer Surveys, Vol. 22, No. 3, pp.237-266 Sept.

[2]- Bernstein, P., Goodman, N., Wong, E., Reeve, C., and Rothnie, J. 1981. "Query processing in a system for distributed database systems(SDD-1)". ACM Trans. Database Sys.6, 4 (Dec.), pp. 602-625.

[3]- Kossmann, D. 2000. "The State of the Art in Distributed Query Processing", ACM Computing Surveys; Dec; Vol. 32,4; ABI/INFORM Global, pp. 422-469.

[4]- Srinivasa, R. 2002. "Network-Aided Concurrency Control in Distributed Databases". Ph.D. thesis, University of Virginia.

[5]- Breitbart, Y., Olson, P., and Thompson, G. 1986. "Database integration in a distributed heterogeneous database system". In Proceedings of the International Conference on Data Engineering (Los Angeles, CA, Feb. 5-7). IEEE, Washington, D.C., pp. 301-310.

[6]- Barkmeyer, E., Mitchell, M., Mikkilineni, K., Su, S. Y. W., and Lam, H. 1986. "An architecture for an integrated manufacturing data administration system". NBSIR 863312, Md. 86, National Bureau of Standards, Gaithersburg.

[7]- Templeton, M., WARD, P., and LUND, E. 1987. "Pragmatics of access control in Merand". Q. Bull. IEEE Comput. SOC. Tech. Committee Data Eng. 10, 3, pp. 33-38, Sept. International Conference on High Performance Computing, pp. 126-131.

[8]- Stonebraker, M., Aoki, M., Pfeffer, W., Sah, A., Sidell, J., Staelin, C. and Yu, A. 1996. "Mariposa: A wide-area distributed database system" The VLDB Journal, Springer-Verlag, 5: pp.48-63.

[9]- Kossmann, D., Franklan, M., and Drach, G. 2000."Cache Investment: Integrating query optimization and dynamic data placement". ACM Trans. Data System.

[10]- Andrew S. Tanenbaum and Herbert Bos 2014. "Modern operating systems". PEARSON.

[11]- Carey M., Laura M., James K., and Berthold R. 1998. "Data Access Interoperability in the IBM Database Family". Bulletin of the Technical Committee on Data Engineering, Special 38 Issue on Interoperability, Vol 21 No.3, September.

[12]- Litwin, W. et al. 1982. "SIRIUS System for Distributed Data Management," in Distributed Databases". H.J. Schneider (ed.), North-Holland.

[13]- Kemme, B., Alonso, G. 2000. "A New Approach to Developing and Implementing Eager Database Replication Protocols". ACM Transactions on Database Systems, v25 i3, p333.

[14]- Ives, Z., Florescu, D., Friedman, M., Levy, A., and Weld, D. 1999. "An adaptive query execution engine for data integration". In Proceedings of the ACM SIGMOD Conference on Management of Data (Philadelphia, PA, USA), pp.299-310, June.

[15]- Chin A. 2001. "Incremental Data Allocation and ReAllocation in Distributed Database Systems". Journal of Database Management, Jan-March v12 i1 pg. 35.

[16]- Hegazi, Mohamed Osman Ali. "A conceptual foundation and integration database designing model." Journal of Computer Science 10.3 (2014): 376-381.

[17]- EL Abbadi, A. and Toueg, S. 1989. "Maintaining availability in partitioned replicated 17 databases". ACM Trans. Database Syst. 14, 2 (June), pp. 264-290.

[18]- Agrawal, D. and EL Abbadi, A., and Steinke. R. 1997. "Epidemic algorithms in replicated databases (extended abstract)". In Proceedings of the 16th ACM SIGACT-SIGMOD- SIGART Symposium on Principles of Database Systems (PODS '97, Tucson, AZ, May 12 14), A. Mendelzon and Z. M. Ozsoyoglu, Chairs. ACM Press, New York, NY, pp. 161-172.

[19]- Hegazi, Mohamed Osman Ali. "An Approach for Designing and Implementing Eager and lazy Data Replication." methodology 110.6 (2015).

[20]- Olson, S., Oledereder, R., Shaw P., and Yach, D. 1998. "The Sybase Architecture for Extensible Data Management". Bulletin of the Technical Committee on Data Engineering, 15 Special Issue on Interoperability, Vol. 21 No.3, Sep.

[21]- Gray J. and Reuter A. 1992. "Transaction Processing: Concepts and Facilities". Morgan-Kaufmann. 41,4; ABI/INFORM Global. pg.578.

[22]- Rosen, K. Discrete Mathematics and Its Applications 4/E, 2003. http://people.cs.vt.edu/~irchen/4004/pdf/discreteMath/chapt53.pdf,

[23]-Tucker A. 1995. "Applied Combinatory". 3rd ed. John Wiley & Sons, Inc., New York, U.S.A.

[24]- Madden, Sam. "From databases to big data." *IEEE Internet Computing* 3 (2012): 4-6.

[25]- Labrinidis, Alexandros, and Hosagrahar V. Jagadish. "Challenges and opportunities with big data." *Proceedings of the VLDB Endowment* 5.12 (2012): 2032-2033.

[26]- Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009.

[27]- Moniruzzaman, A. B. M., and Syed Akhter Hossain. "Nosql database: New era of databases for big data analytics-classification, characteristics and comparison." arXiv preprint arXiv:1307.0191 (2013).