ISSN: 2321-7782 (Online)
Volume 3, Issue 6, June 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Comparison of HDBMS, NDBMS, RDBMS and OODBMS

Dr. Ranjana Ingolikar[1]
HOD, Dept. of Computer Science
SFS College, RTM Nagpur University
Nagpur, India

Rasika Khandal[2]
Assit. Professor, MCA Department
SRPCE, RTM Nagpur University
Nagpur, India

Dr. Rahul Mohare[3]
Asst. Professor, MBA Department
DMIMS, RTM Nagpur University
Nagpur, India

Abstract: ***The similarities and differences between HDBMS, NDBMS, RDBMS and OODBMS give the clear cut idea to the user or database designer of selecting better model for storing information. In this paper we have discussed the mandatory parameters used for comparing the different database models. The comparison is done to bring out the structural as well as operational differences between different database models. The differences are presented in the form of grid view so that audience can drill into and review each difference in the grid with respect to the selected parameters.***

Keywords: HDBMS, NDBMS, RDBMS, OODBMS, Parameterized Comparison

I. INTRODUCTION

A Database Management System or DBMS is software designed to assist in maintaining and utilizing large collections of data, and the need for such systems, as well as their use, is growing rapidly. A DBMS is piece of software that is designed to make the tasks easier. By storing data in a DBMS, rather than as a collection of database files, we can use the DBMS's features to manage the data in a robust and efficient manner. As the volume of data and number of users grow, hundreds of gigabytes of data and thousands of users are common in current corporate databases. DBMS contains a number of Data Models. A Data Model is a collection of high-level data description constructs that hide many low-level storage details. A DBMS allows user to define the data to be stored in terms of data model.

a) Basic Fundamentals of HDBMS and NDBMS

The hierarchical data model organizes data in a tree structure. There is a hierarchy of parent and child data segments. This structure implies that a record can have repeated information, generally in child data segments. The hierarchical model employs two main data structuring concepts: records and parent-child relationships. A record is a collection of field values that provide information on an entity or a relationship instance. Records of the same type are grouped into record types. A record type is given a name, and its structure is defined by a collection of named fields or data items. Each field has a certain data type, such as integer, real, or string.

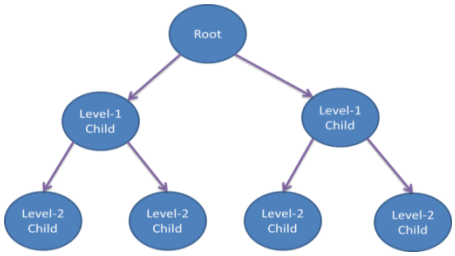

Fig. 1 Hierarchical Model

The hierarchical database model depicts a set at one-to-many (l:M) relationships between a parent and its children. In addition, a many-to-many (M:N) relationship formed by using the concept of Virtual (or Pointer) Record Type or Virtual Parent-Child Relationship (VPCR).

The principles behind the hierarchical model are derived from Information Management System (IMS), which is the dominant hierarchical system in use today by a large number of banks, insurance companies, and hospitals as well as several government agencies. Another popular hierarchical DBMS is MRI 's System-2000 (which was later sold by SAS Institute).

The next version of database management system is Network Database Management System (NDBMS). The network model makes an attempt to deal with many of the hierarchical model's limitations. A network database consists of a collection of records connected to one another through links. Each record is a collection of fields (attributes), each of which contains only one data value. A link is an association between precisely two records.

The model had its origins in the Conference on Data Systems Languages (CODASYL) which had created the Data Base Task Group to explore and design a method to replace the hierarchical model. CODASYL produced standard network specifications for a network schema, subschemas, and a data management language. The data management language contains three basic components: a Data Definition Language (DDL) used to define schema components, a subschema DDL, and a Data Manipulation Language (DML) designed to manipulate the database contents.

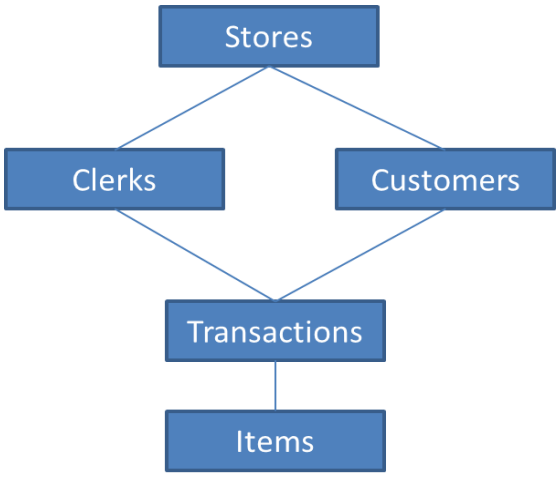

Fig. 2 Network Model

A network database model implementation, like the hierarchical database model, may be represented by a tree structure in which 1:M relationships are maintained. However, the network model easily handles complex multiparent relationships without resorting to the creation of logical (as opposed to physical) database links.

The network model requires the database administrator to pay close attention to the model's physical environment. In turn, application programmers must take note of the model's physical detail.

b) Basic Fundamentals of RDBMS and OODBMS

RDBMS is based on the relational model and data in a RDBMS are stored in the form of related tables. Each column corresponds to an attribute of the relation and each row corresponds to a record that consists of data values for an entity. The tables are linked together by using the concept of primary key (one or more fields or columns that uniquely identify each record or row in the table) and a foreign key in another table with the same data value. The relation schema is as shown in following fig.3.

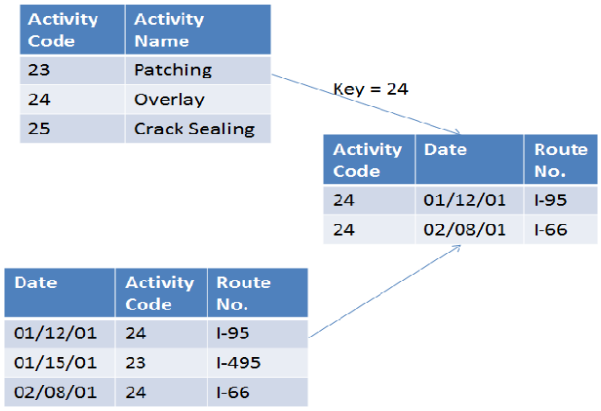

Fig. 3 Relational Data Model

The relationship between tables are one-to-one (1:1), one-to-many (1:M) and many-to-many (M:N). Main elements of a RDBMS are the concepts of relational integrity and normalization. These concepts are based on the 13 rules for a relational system developed by Ted Codd. Currently most of the dominant DBMSs such as IBM's DB2 family, Oracle, Microsoft's Access and SQL Server are RDBMS.

An Object-Oriented Database Management System (OODBMS), sometimes referred as Object Database Management System (ODMS) is a new Database Management System (DBMS) that supports modeling and creation of data as objects. OODBMS is a DBMS which allows information to be represented in the form of objects as used in object-oriented programming. OODBMSs were developed in 1980s to overcome the limitations in RDBMSs such as handling large and complex data.

Each object is being uniquely assigned an identity, i.e. object ID (OID) which will never change throughout the life of the object. This OID is used to link different objects dynamically when objects interact with each other, in contrast to the permanent primary and foreign key relationship in RDBMS. OODBMSs enforce object oriented programming concepts such as encapsulation, polymorphism and inheritance as well as database management concepts such as Atomicity, Consistency, Isolation and Durability. Object-oriented languages such as Java, C#, Visual Basic .NET and C++ can work well with OODBMSs. Since both the programming language and OODBMS use the same object-oriented model, the programmers can maintain the consistency easily between the two environments.

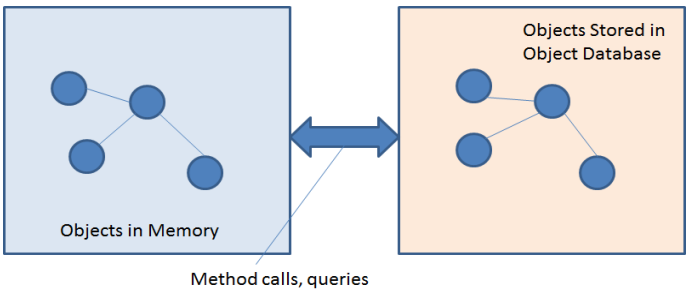

Fig. 4 Object-Oriented Data Model Storage Structure

Object Data Management Group (ODMG) is an organization standard for storing objects into OODBMS. The ODMG standard consists of the object query language (OQL), object model, object defining language (ODL) and the bindings to OOP language (i.e. to enable OQL to work closely with OOP language such as Java, C++ and Smalltalk). Examples of OODBMS products are DB4O developed by Versant, Ardent™ developed by Ardent Software and ObjectStore™ developed by Object Design Inc. etc.

II. Need Of Comparison Between Different Data Models

The presentation of the similarities and differences between different data models is of great importance both for database designers and database users. Comparison between models helps us for giving ideas of choosing the best one for storing information. We have used the above data models in our research study and defined the parameterized comparison between these considered data models.

III. Parameterized Comparison Between HDBMS, NDBMS, RDBMS And OODBMS Data Models

This paper proposes the IND-OCPA-P model to analyze the security of the proposed EOB and the encryption schemes supporting an efficient range query over encrypted data.

Sr.No.	Considered Parameters	HDBMS	NDBMS	RDBMS	OODBMS
1	Defining Standard	IMS (Information Management System)	DBTG (Database Task Group)	SQL2 (Structured Query Language)	ODMG3.0 (Object Data Management Group)
2	Data Structure	Family tree like structure of database	Database structure look like a network form	Data is stored in the form of table structure (rows & columns)	Data is stored in the form of objects by using object-oriented programming language
3	Establishing Relationship	1:1, 1:M Parent-child relationship is established	1:1, 1:M and Multiple parent-child relationship is established	1:1, 1:M, M:N relationship is established between tables	M:N relationship between objects and pointers are used to link objects to established relationship
4	Query Language	It does not support independent stand-alone query interface. HDML used for manipulating hierarchical database	CODASYL is used in NDBMS. They also have different host language. The original DBTG report used COBOL as a host language. Another popular host language is PL/1, Pascal	The powerful language called SQL or query by example are used to manipulate data stored in the database	OQL is a query language based on SQL 92 and equivalent to SQL. ODL used to define database object schema and OIF used to dump and load the object databases state to and from the files
5	Data Integrity Constraints	We cannot insert the information of child who does not have any parent. Deletion	No member can exist without an owner. A user must therefore first define the owner	Data integrity constraints are ensured by using key constraint like	Data integrity is controlled by the application. Object references created

		of parent record automatically results in deletion of all its children and descendent record	record and then the member record. This ensures the integrity.	primary key, foreign key and domain constraints.	in the application are mentioned in the database
6	Data Independence	The Hierarchical data model enables users to design more applications which will access the same data in a database.	The application programs work independently of the data. Any changes made in the data characteristics do not affect the application program	Ensuring data independence from application programs. Data can be reorganized and modified without affecting the mode of using them.	Data independence is formed by using class structure. Class can be reorganized without affecting the mode of using them.
7	Data Access	Data access is easy via the key attribute, but difficult for other attributes. e.g. In the geographical case, easy to find record gives its geographical level (state, county, city, census tract), but difficult to find it given any other attribute	An application can access an owner record and all the member records within a set.	SQL language is used to access the data. DDL and DML commands are used for retrieve data from database	Data is accessed by initiating the class object. Object Query Language (OQL) is a declarative (nonprocedural) language for querying and updating objects.
8	Supporting Data Types	It support all basic data type with their default values and field width but it is inconvenient for storage of complex object data type like audio/video, graphics etc.	It support all basic data type with their default values and field width but it is inconvenient for storage of complex object data type like audio/video, graphics etc.	Only primitive data type like text and numerical data to be stored in the database with their default values and field width. E.g. for "Numeric" data type field width is long number and for "Text" data type field width is 255 character long.	All basic primitive data types with their default values and field width. It also supports Complex object data types such as graphics, audio/video or geographic information etc.
9	Supporting Mathematical	Does not support	Does not support	Mathematical functions are	User defined functions and

	Functions			supported by RDBMS. E.g. sqrt(), min(), max(), sum(), count() etc.	methods are created in the class body. It also support built in functions depending on which programming language is used
10	Data Decomposition	Data decomposition is not possible	Data decomposition is not possible	Data decomposition is possible. After decomposition to establish the relationship between tables by using foreign key constraints.	Decomposition can be performed by creating classes. Relationship is established between classes by using the characteristics of inheritance.
11	Language Binding	IMS (Information Management System) database work with traditional mainframe languages like COBOL, Assembler, PL/1, C and Pascal. It also uses Java to access an IMS database.	CODASYL data model supports COBOL, PL/1, Pascal.	Relational data model support Object Oriented Programming Language like C, C++, Java, .Net etc. but impedance mismatch is found in fetched queries	In Object Oriented data model no impedance mismatch is found because data is stored in the form of objects in files and fetched by using Object Query Language (OQL). It support Java, .Net, C++ etc.
12	Data Storage Limitations	Data storage limitation depends on which data model product to be used. E.g. for IMS data model limits a VSAM dataset to 4 GB (and OSAM to 8 GB.	Data storage limitation depends on which data model product to be used.	Data storage limitation depends on which data model product used by the user for storing information. It is not supported for storing of big data in the database. Problems will occur when retrieving data from the database. E.g. Microsoft Access limits to store 2GB	There is no physical limit, storing of number of objects in database files. E.g. DB4O limits to memory and file size. Maximum database file size is 254 GB/database file.

				data for 100,000 records per table.	
13	Example of Database Model	An IMS database is made up of segments arranged in hierarchical order. An example of hierarchical arrangement of data is in XML.	NDBMS proposed by DBTG (Data Base Task Group) of the programming language committee of the "Conference on Data Systems Language" (CODASYL) data model. The organization responsible for the definition of the COBOL programming language.	Relational databases are Microsoft Access, DB2, Oracle, MySQL and PostgreSQL etc.	Object Oriented databases are DB4O, Versant ODBMS, POET ODBMS, Objectivity/DB ODBMS, ObjectStore ODBMS etc.
14	Database Security	Security does not apply on the data present in the database. Some of the Hierarchical data model like IMS applies security on database by providing the facility of logging to user.	Security does not apply on the data present in the database. Some of the Network data model like CODASYL data model applies security on database by providing the facility of logging to user.	In relational database model views are valuable in context of database security. Views give a group of user access to the information that they allowed to see. Relational model also give a security on database by allowing username and password for that particular database product.	It provides high level of security to the data. Data or information is stored in the form of object into the database file. The files contain objects which are in encrypted format that are not easily readable by any user. OODBMS model also gives a security on database by allowing username and password for that particular database product.

TABLE I Comparison of Database Management System

IV. CONCLUSION

From the above parameterized comparison we come to the conclusion that the following points regarding to HDBMS, NDBMS, RDBMS and OODBMS:

1. Although the hierarchical and network data models are powerful and efficient, they are complex, requiring users to understand data structures and access paths to data. From the above comparison we came to know that they were designed for use with programs rather than for interactive access by users. In these models ad hoc queries are not supported. In IMS DBA can assign passwords to programs or to a data file limiting security to very superficial level whereas DBTG system use an extensive system lock and keys to limit access. The hierarchical database and network database has lots of limitations and it is inefficient for storage of any kind of information.
2. Many of the problems associated with flat-file systems were partially addressed with the introduction of the IMS database model. Database task group (the DBTG) addresses database standards by developing a set of rules, or standards, for database management systems. The CODASYL DBTG developed what is called the Network Model for databases.
3. In RDBMS complex data relationships are difficult to model whereas ODBMS can handle arbitrary complexity because user can write methods on any structure. Table structures used for data storage relatively easy to understand in RDBMS whereas Objects are used to store data in ODBMS only good programmers can understand.
4. Except ODBMS the integrity constraints of other database models are based on the keys of the schema where as in ODBMS it is controlled by the application creating the objects.
5. The relational database is capability to handle lots of data in the form of tables but it is inefficient for the storage of complex data in the database. Also there is an impedance mismatch is found when it works with object oriented programming language.
6. Object-oriented database is well suited and works with high speed in accordance with object oriented programming language. Unlike RDBMS it doesn't form impedance mismatch.
7. Object-oriented databases have as their main objective encapsulation, being stored together with the data and the methods. They are inseparable. It is said that we have to encapsulate the data and the methods with an independence of classes and not with an independence of data.
8. There is again one big advantage of object oriented database is to handle complex object data type with no physical limit of storage that will not happen in any of the database system.
9. Object oriented database system gives the data security. E.g. an object stored into database file is in encrypted format. It also provides database security like other database system.

It seems likely that the future need of users for large database systems will drive them toward database systems based on the mentioned data models. Subsequent research and papers should focus on quantifiable advancements and specific benefits derived from the use of different database software.

Acknowledgment

I would like to thanks Dr. Ranjana Ingolikar and Dr. Rahul Mohare for their valuable guidance and support.

References

1. Raghu Ramakrishnan/Johnnes Gehrke, "DatabaseManagement Systems", Second Edition, McGraw-Hill Higher Education.
2. Gheorghe SABAU, "Comparison of RDBMS, OODBMS and ORDBMS", Bucharest, Romania.

3. "Db4o Open Source Database" Sponsored by Versant Corporation, db4o Product Information 8.1.
4. A. Juozapavicius, "Introducing to Database", http://www.mif.vu.lt/cs2/courses/infos2.pdf
5. Roshan Ragel, http://www.differencebetween.com/difference-between-rdbms-and-vs-oodbms, April 22, 2011.
6. http://www.wech.com.my/assignment/ormap/2Chap-literature%20review.pdf
7. "IMS/DB Introduction and Structure",www.sysed.com/DnLoads/.../IMSControlBlock/04_Chapter01_IPA1.pdf.
8. "Network CODASYL Data Model" http://coronet.iicm.edu/is/scripts/lesson03.pdf.
9. http://en.wikipedia.org/wiki/IBM_Information_Management_System.

AUTHOR(S) PROFILE

Dr. Ranjana Ingolikar, received Ph.D. degree from RTMNU, Nagpur. She is working as H.O.D. of Department of Computer Science in St. Francis De Sales College, Nagpur.

Mrs. Rasika V. Khandal, received MCA degree from RTMNU, Nagpur in 2008. Since 2009 she is working as Asst. Lecturer in SRPCE, Nagpur in MCA Department.

Dr. Rahul Mohare, received Ph.D. degree from RTMNU, Nagpur. He is working as Asst. Lecturer in DMIMS, Nagpur in MBA Department.

