



# A Comparative Study of Data Management Systems

Samah Waleed Abu Salim<sup>1</sup>, Mohd Zainuri Saringat<sup>1\*</sup>, Salama A. Mostafa<sup>1</sup>

<sup>1</sup>Faculty of Computer Science and Information Technology,  
Universiti Tun Hussein Onn Malaysia, 86400 Johor, MALAYSIA

\*Corresponding Author

DOI: <https://doi.org/10.30880/jscdm.2020.01.01.002>

Received 20<sup>th</sup> December 2020; Accepted 10<sup>th</sup> February 2020; Available online 1<sup>st</sup> March 2020

**Abstract:** Databases play an important role in all areas that use computers such as industry, agriculture, trade, engineering, medicine, etc. Where relevant data is stored in a database. Information systems programmers have long been building their programs on data files and files have played an important role in storing information. In the data file processing system, each program is designed to perform a specific purpose and has its own files without a general framework linking all the programs and possibly repeated data in more than one file. The use of these files many problems and disadvantages such as data redundancy, heterogeneity or data compatibility, limited participation, and the difficulty of maintenance. This paper introduces the overview of Database Management System (DBMS) such as its components, purposes, and data models which are describing and manipulating data, relationships between data, and constraints on the data in an organization.

**Keywords:** Database Management System (DBMS), Relation Database Systems (RDBMS), Object-Oriented Database Management System (OODBMS), Network Database Model, Hierarchical Database Management System (HDBMS)

## 1. Introduction

Database Management System (DBMS) is an extensive set of interrelated programs used to generate, store, retrieve and manage data in the database effectively [1]. It is also responsible for access, security, storage, manage a large set of structural data, and provides data access to multiple concurrent users while maintaining data integrity [2]. Database management systems can be considered as interfaces between databases and end-users, or between databases and application programs. Database management systems rely on special programming languages called query languages [3]. The idea of database management systems began in the second half of the last century, where the leading IBM company in several fields, especially information systems, issued the first database management system and named IMS [4].

### 1.1 Advantages of DBMS

The database has a great importance in various works, and its importance is highlighted in many points, including:

- i. Storing a vast amount of data beyond human potential.
- ii. Data storage in an integrated way.
- iii. Monitor changes in stored data and make necessary adjustments.
- iv. Complete confidentiality of stored data.
- v. High speed in storage and retrieval of data.

\*Corresponding author: [zainuri@uthm.edu.my](mailto:zainuri@uthm.edu.my)

2020 UTHM Publisher. All rights reserved.  
[penerbit.uthm.edu.my/ojs/index.php/jscdm](http://penerbit.uthm.edu.my/ojs/index.php/jscdm)

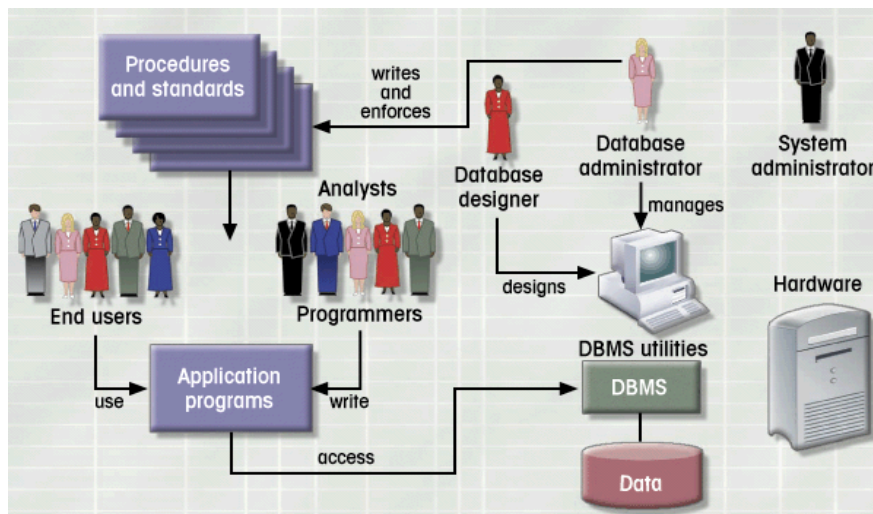
## 1.2 Purposes of DBMS

DBMS has been created to manage all the disadvantages of standard operating systems supporting typical file processing systems. Some of the goals DBMS achieves are data redundancy and inconsistency, data access difficulties, information isolation, integrity issues, update atomicity, concurrent access by multiple users and security issues. And more reasons to use DBMS as follows:

- i. Control data replication and restrict users who do not have certain privileges.
- ii. Provide a storage environment that does not lose data and allow the extraction of information from existing data.
- iii. Extreme flexibility in using and modifying data and providing highly updated data.
- iv. Provide multiple interfaces for user interaction with data and represent complex relationships between data easily.
- v. The possibility of applying security and confidentiality restrictions and maintaining data integrity.

## 1.3 Components of DBMS

Database systems consist of several components [5], forming an integrated and interconnected system as follows:



**Fig. 1 - DBMS components and its environment**

Figure 1 shows the components and environment for database that include data which is the point that all components of database system are centered. Data must be integrative, non-repetitive and participatory, Hardware where the databases are stored on magnetic disks. The middle layer between the data stored in the files on the top of the disks and among the users of the database called Software that used to manage control and the overall database, there are four sections of users who use the databases and execute different operations on the databases: (1) Databases Programmer who write database applications program in some of the programming languages such as (Cobol, Java, Visual Basic, or C++) and implementing the program to make sure there are no errors. (2) Database Administrator who responsible for determining the database requirements of programs and equipment, operating the database, defining security and confidentiality conditions, and determining the terms of use. (3) End-Users who deal with the database and perform different operations such as insert, update, and delete data etc. And Procedures that includes instructions and rules governing the design and use of the database system.

## 2. Data Model

A data model also called database model [6]. It represents the framework that describe the structure of the database. This framework describes how the data will be stored in the database, this description includes the data type, relations between the tables, conditions, constrains, and tables themselves. The data itself will literally be stored within this framework on a database. So, building a data model is a critical step in the design of the data warehouse. There are number of different data models.

### 2.1 Relational Model

Relational Database Management Systems (RDBMS) started in 1980 and became the first option for data storage in databases mostly used for financial records, manufacturing information and so on. RDBMSs are based on a schema-defined relational model. Two concepts are used in this model: table and relationship. A relational table is a well- defined set of rows and columns and the relationship is formed between the rows of the tables [7]. Relational data can be queried and manipulated using SQL query language [8].

The rows form the records, the columns (table attributes) form the fields, and these tables have (one-to-one) (1:1), one-to-many (1:M) and many-to-many (M:N) relationship between them that enables the user to reach Data from various parts of the database. Tables are associated together by using the primary key concept (one or more fields or columns that define each record or row in the table uniquely) and a foreign key in another table with the same data value. Fig 2 shows the relational database schema [9].

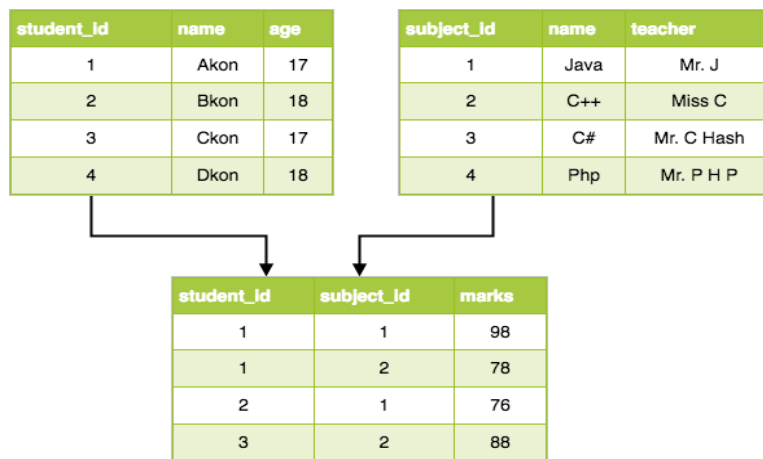


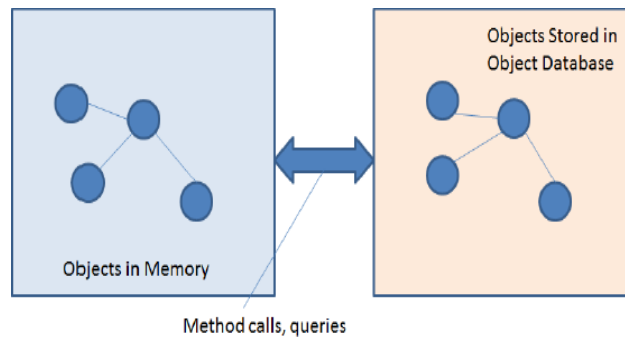
Fig 2 - Relational Data Model

**Advantages of RDBM:** The Relational model structures data in a manner that avoids complexity, it is more flexible than other types to collect data from different sources. Relational database offers reliability, simple indexing to organize data and maintaining data integrity by setting up defined constraints [10]. Relational model has special tools and characteristics and can be used with all programming languages.

**Disadvantages of RDBM:** Relation Database Systems (RDBMS) are used in nearly every area that requires data storage. Relational database management system is based on a solid mathematical basis and is very powerful in handling a lot of data that helps them to appear wherever software systems are. However, in some applications, RDBMS appears to be highly inefficient in processing big amounts of complicated information such as picture collections, video stream collections, etc. The relational data model has a restricted amount of data types that are not adequate to represent real-world objects, and also it is not enough to represent data and data relationships [11].

### 2.2 Object Oriented Model

Object Database (also called Object Oriented Database) is a database concept in which information is presented in the form of an object as it is used in object-oriented programming. The objects encapsulate the attributes and associated methods or member functions [12]. When a different object of the same type is grouped together, they form a class. This model is primarily used for multimedia applications as well as data with complex relationships. Object-Oriented Database Management Systems (OODBMS) were developed in the 1980s to overcome restrictions on RDBMSs such as handling large and complex data. An example of a program that follows this sample is FastObject.Net, etc. It works well with Object oriented languages such as Java, C #, Visual Basic .NET, and C ++. Fig 3 shows the Object-Oriented database schema [9].



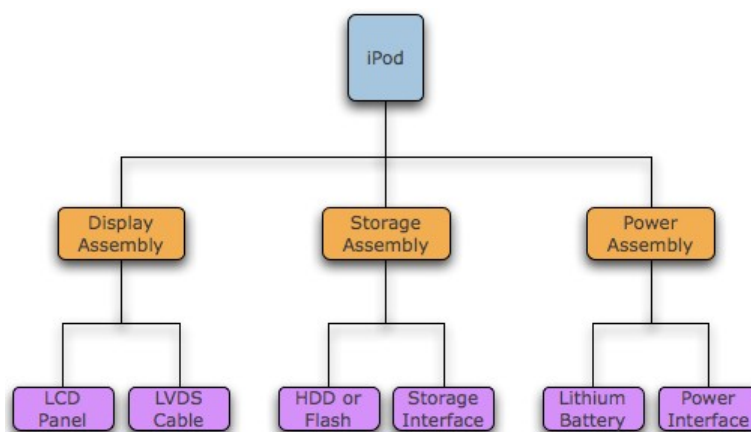
**Fig 3 - Object-Oriented Data Model**

**Advantages of OODB:** Unlike traditional databases (such as hierarchical, network or relational), the object-oriented database is capable of storing different types of data, for example, pictures, voice video, including text, numbers and so on. One of the most significant features of Object-oriented database has the ability to reference objects of complicated constructions, making them ideal for complicated data presentation [12]. Areas of Computer Integrated Manufacturing (CIM), Computer Aided Design (CAD), Computer Aided Manufacturing (CAM), Computer Aided Software Engineering (CASE) and advanced office automation system use object-oriented database system to handle complex graphical data and hypermedia data [11]. Also, it enables protection and security systems to be based on the notion of object.

**Disadvantages of OODB:** When compared this model to the more common relational database model. This gave it very little time to mature, which resulted in a restricted number of skilled programmers and support. The object-oriented model focuses on storing complex data structures referred to as objects. This makes it less effective when used with simple relationships on simple data records [13]. Another drawback of this model is that pure implementation is hard. This is true because object-oriented models, unlike relational models, use complicated data types that require easy structures such as tables and associated indexes. Relational databases are a result of mathematical set theory, but there is no powerful mathematical foundation for object-oriented designs.

### 2.3 Hierarchy databases Model

This model appeared at the beginning of 1970. It is a data model in which the data is arranged in a tree-like structure. This structure allows the representation of information using parent / child relationships. Each child can have only one parent, whereas each parent record can have more child or no child at bottom level as you see in fig 4. All attributes of a specific record are listed under an entity type. In a database, an entity type is the equivalent of a table; each individual record is represented as a row and an attribute as a column. Entity types are related to each other using 1:N mapping, also known as one to many relationships [14].



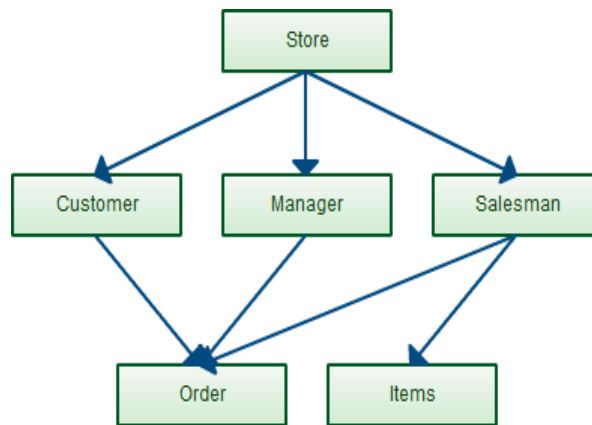
**Fig 4 - Hierarchy Databases Model**

**Advantages of HDBM:** Navigating between records in a hierarchical database is very quick as parent/ child relationships are implemented with pointers from one record to another, promotes data sharing and database security is provided.

**Disadvantages of HDBM:** The hierarchical database's primary drawback is its rigid structure. If you want to add a field to a table, a new table must be created for large records by the database management system. Unlike a SQL database, there is no ALTER TABLE command in the hierarchical model. In addition, if you want to add a new relationship, you will need to construct a new and potentially redundant database structure and you will need to know the features of physical data storage.

## 2.4 Network databases Model

Charles Buckman developed a network database model in 1969 [16] to enhance the hierarchical database model [15] [14]. This model was created to increase flexibility, easier understanding of relationships by using direct indicators to illustrate relationships between records, and overcome the problems of hierarchal model. The network model allows each record to have multiple parent and child records, forming a lattice structure. Fig 5 shows the Object-Oriented database schema.



**Fig 5 - Network Databases Model**

**Advantages of NDBM:** The network model is conceptually simple and easy to design, just like the hierarchical model, it can handle the one to many and many to many relationships which is real help in modeling the real life situations, the data access is easier and flexible than the hierarchical model, and the network model is better than the hierarchical model in isolating the programs from the complicated physical storage details. Also, based on the mathematical set theory, this model is articulated around related whole of recordings [14].

**Disadvantages of NDBM:** All records are kept using pointers and thus the entire structure of the database becomes very complicated. Inserting, deleting and updating any record operations require a large number of pointer adjustments, and it is very difficult to make structural changes to the database.

**Table 1 - Comparison between Database Model**

	<b>RDBMS</b>	<b>OODBMS</b>	<b>HDBMS</b>	<b>NDBMS</b>
Defining Standard	SQL (Structured Query Language)	ODMG (Object Data Management Group)	IMS (Information Management System)	DBTG (Data Base Task Group)
Data Organization	Tables	Objects	Files, Records	File, Records
Establishing Relationships	One to One (1:1), One to Many (1: N), and Many to Many (M: N) relationships can be implemented.	Many to Many (M:N) relationships are used between objects to link object with established relationship.	One to One (1:1) and One to Many (1:M) Parent-child is established, but Many to Many (M: N) relationship cannot established in this model.	One to One (1:1), One to Many (1: N), and Many to Many (M:N) relationships can be implemented.
Support for Mathematical Function	Support many functions such as sqrt (), min (), Max (), and sum ().	Support math functions based on the programming language that used.	Not support.	Not support.
Data Integrity	<ul style="list-style-type: none"> <li>• Does not suffer from any Insert eccentricity.</li> <li>• Free from update eccentricity.</li> <li>• Free from delete eccentricity.</li> </ul>	Data integrity is controlled by the application.	<ul style="list-style-type: none"> <li>• Cannot insert the information of a child who does not have any parent.</li> <li>• Inconsistency problem happen during the update operation.</li> <li>• Deletion of parent results in deletion of child records.</li> </ul>	<ul style="list-style-type: none"> <li>• Does not suffer from any Insert eccentricity.</li> <li>• Free from update eccentricity.</li> <li>• Free from delete eccentricity.</li> </ul>

### 3. Discussion and results

From the above comparison we found that the hierarchical and network data models are effective, but they are complex, inflexible and difficult to manage and maintain. In RDBMS is difficult to model complex data relationships whereas can do it easily in ODBMS because user can write methods on any structure. RDBMS use table structure for data storage and it easy to understand whereas ODBMS used object in store data and only a perfect developer can understand that. The object-oriented database is fully suitable and operates at high speed in accordance with object-oriented programming language.

### 4. Conclusion

Finally, databases management system was appeared to solve the problem of large files, which contain many information is difficult to access, modify, and search, in addition to the problem of repetition, which makes data less accurate, and occupies more space in the memory of the computer, This is the concept of databases management system, which is a system that enables the arrangement of data serially and coherently, without duplication, with the possibility to modify data, add and delete, and set conditions Limitations of State information and confidentiality, and to identify people who have access to it. This paper proposed the idea of database management system, components and various type of data model with their advantages and disadvantages.

### Acknowledgement

The authors would like to acknowledge the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400 Johor, Malaysia.

## References

- [1] Sharma, M., Singh, G., & Singh, R. (2016). Design and analysis of stochastic DSS query optimizers in a distributed database system. *Egyptian Informatics Journal*, 17(2), 161–173. doi:10.1016/j.eij.2015.10.003.
- [2] Er.Anshuman Sharma, Mr. Anurag Gupta, Jagmohan Mago (2011). *Fundamentals of DBMS*. Lakhanpal Publishers, fourth edition.
- [3] Dignös, A., Böhlen, M. H., Gamper, J., & Jensen, C. S. (2016). Extending the Kernel of a Relational DBMS with Comprehensive Support for Sequenced Temporal Queries. *ACM Transactions on Database Systems*, 41(4), 1– 46. doi:10.1145/2967608.
- [4] James P. Fry and Edgar H. Sibley (1976). *Evolution of Data-Base Management Systems*. *Computing Surveys*, Vol. 8, No, I, March 1976.
- [5] Prabhjot, and Neha Sharma (2017). Overview of the Database Management System. *International Journal of Advanced Research in Computer Science (ijarcs)*. Volume 8, No. 4, May 2017.
- [6] A. B. Silberschatz, H. F. Korth, and S. Sudarshan (2011). *Database System Concepts*. McGraw-Hill, sixth edition.
- [7] Stanescu, Liana & Brezovan, Marius & Burdescu, Dumitru. (2017). An algorithm for mapping the relational databases to mongodb - a case study. *International Journal of Computer Science and Applications*. 14. 65-79.
- [8] Krisciunas, A. (2014). Benefits of NoSQL: <https://www.devbridge.com/articles/benefits-of-nosql/>
- [9] Ranjana Ingolikar, and Rasika Khandal (2015). Comparison of HDBMS, NDBMS, RDBMS and OODBMS. *IJARCSM*, Volume 3, Issue 6, June 2015.
- [10] Song, E., Haw, S.-C., & Chua, F.-F. (2018). Handling XML to Relational Database Transformation Using Model-Based Mapping Approaches. 2018 IEEE Conference on Open Systems (ICOS). doi:10.1109/icos.2018.8632805.
- [11] Vipin Saxena and Ajay Pratap. (2013), Performance Comparison between Relational and Object-Oriented Databases, *International Journal of Computer Applications (0975 – 8887)* Volume 71– No.22, June 2013.
- [12] Sonal Kanungo and Rustom. D Morena. (2017), Semantic model for Complex object in Object oriented database. *International Journal of Advanced Research in Computer Science*, Volume 8, No. 5, May-June 2017.
- [13] S. O. Ogunlere, S. A. Idowu. Comparison Analysis of Object-Based Databases, Object Oriented Databases, and Object Relational Databases. *Asian Journal of Computer and Information Systems (ISSN: 2321 – 5658)* Volume 03– Issue 02, April 2015.
- [14] Thiam, A., Xu, D., Miao, H., & Niu, Z. (2018). Comparison of Query Languages for Semantic Database Models. *Proceedings of the 2018 International Conference on Computing and Data Engineering-ICCDE 2018*. doi:10.1145/3219788.3219806
- [15] Gaurav Jindal and Simmi Bali. (2012). Hierarchical Model Leads To the Evolution of Relational Model. *International Journal of Engineering and Management Research*, Vol. 2, Issue-4, August 2012.
- [16] Richard Williams. (2019). Data Management and Data Documentation. In book: *Data Management and Data Description*, pp.75-114.