

The Object-Oriented Database Management System versus the Relational Database Management System: A Comparison

Mounika Mandapuram^{1*}, Md. Faruk Hosen²

¹Cognizant Technology Solutions, Teaneck, New Jersey, USA

²Department in accounting, Nandina Sheikh Anwar Hossain College, Jamalpur, BANGLADESH

*Corresponding Contact:

Email: mounikamandapuram09@gmail.com

ABSTRACT

Both object-oriented and relational database management systems are referred to as DBMS. The web-scale is expanding at an incredible rate, and with it comes an increase in mobile applications, technologies, and social media platforms. The rapid growth of these technologies produces a variety of structured and unstructured data in various forms, each of which has various consequences for data security. Using RDBMS and OODBMS, this study's methodology outlines the development, installation, and management process of multiple and complicated data sets. This study gives a documented study comparing different kinds of data stored in databases by employing two types of database techniques: relational database method and object-oriented database technique. We use a different fingerprint database for the web and mobile devices. This database is also managed using different technologies, such as MySQL, and it accommodates data generated using distinct methods. In addition, this study explores the benefits and advantages of the various databases and determines the ideal database management system for organizations. According to the survey's findings, RDBMS stores data and the relationships between them in tables, but OODBMS stores data in objects, in a manner analogous to Object-Oriented Programming (OOP). According to the study's findings, the primary distinction between relational database management systems (RDBMS) and object-oriented database management systems (OODBMS) is that RDBMS is a Database Management System based on the relational model. In contrast, OODBMS is a Database Management System that facilitates creating and modeling data as objects.

Key Words: OODBMS, RDBMS, Contrast, Involvedness, Legacy

INTRODUCTION

Data management programs are software applications that give modeling services, manage and manage a massive amount of relevant information shared by various users over an extended period, and manage this information for an extended time. A relational

database management system (RDBMS) is a type of database management system that organizes its data storage in connected tables. It does this by establishing a clear connection between the various database tables. Structured Query Language, or SQL, is a tool the programmer can use to manage the data stored in RDBMS. On the other hand, an object-oriented database management system (OODBMS) is a type of database management system that, like object-oriented programming, represents data in the form of objects (Ya-Hui & Raschid, 2000). Finally, it ensures that there is a greater degree of coherence between the database and the programming language.

The Database Management System allows the company's database to be utilized effectively and efficiently. Any business, whether large or small, must deal with enormous volumes of data, including information about customers or clients, user information, and other internal and external information (Mandapuram, 2017a). Without the assistance of a powerful database management system, it is not simple to generate, update, or convert massive amounts of data technically. The database is easier to handle and more productive thanks to the DBMS, which enables new and more intelligent systems and provides more detail.

In the following section of this paper, we will conduct an in-depth analysis of RDBMS and then compare it to OODBMS. Relational Database Management System is what is meant by the abbreviation RDBMS. It is a database management system based on the relational model, which means that the data and relationships are portrayed by a set of tables connected to one another. It is a database management system (DBMS) that allows the user to control the databases by letting them create, update, administer, and interact with a relational database (De et al., 2001). Object-Oriented Database Management System, or OODBMS for short, is the acronym that describes this type of database. Object-oriented programs use the database management system (DBMS), which stores data expressed as objects. The Object-Oriented Database (OODB) uses various object-centric principles, including object categories, object ownership, polymorphism, encapsulation, and inheritance. Finally, an object-oriented database can manage significantly more detailed data than a related database. For example, the Versant Object Database, the Objectivity / DB, the Object Store, and the ZODB (Zope Object Database) are a few other instances of OODBMS.

Relational and object-oriented databases are engaged in a never-ending struggle for supremacy in database administration. This conflict is fought over storage capacity. Let us look at the world of relational and object-oriented databases to determine which suits your data management requirements (Leitheiser & March, 1996). Databases are valuable tools for storing and manipulating data in an efficient and well-organized way. Relational and object-oriented databases are only two examples of the many kinds of databases available today. Relational and object-oriented databases are two types of databases that can be highly effective tools for managing data; nevertheless, they approach the task of dealing with information in quite different ways.

CONCEPTUAL FRAMEWORK

A relational database management system (RDBMS) is a type of database management system that organizes its data storage in connected tables. It does this by establishing a clear connection between the various database tables. Structured Query Language, or SQL, is a tool the programmer can use to manage the data stored in RDBMS. On the other hand, an object-oriented database management system (OODBMS) is a type of database

management system that, like object-oriented programming, represents data in the form of objects (Leitheiser & March 1996). Finally, it ensures that there is a greater degree of coherence between the database and the programming language.

What exactly is an RDBMS?

A database management system (DBMS) is a piece of software or an application on a computer that can be used to construct and manage databases. It was first presented to the public in 1960. The relational database management system, RDBMS, was initially developed in the 1970s. The relational model serves as its foundation. It contributes to the storage and management of data being done in a more effective manner (Dietrich & Urban, 2010). RDBMS makes use of a variety of restrictions to establish relationships between the tables in the database. Tables are also referred to as entities in some contexts. For example, a single entry is represented by a row, while each attribute is displayed in its column.

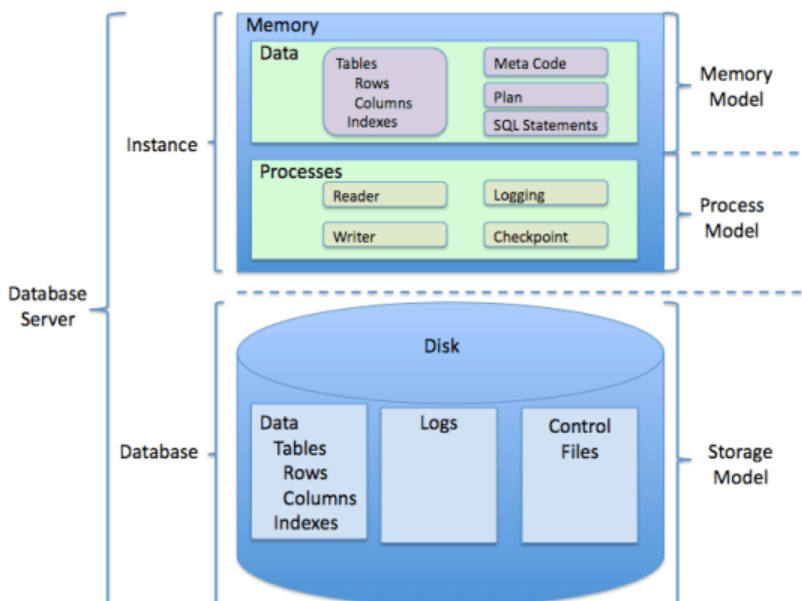


Figure 1: Demonstrates the fundamental architecture of a relational database.

Utilizing an RDBMS can provide a variety of benefits to an organization. RDBMS are capable of performing the process of normalization, which is the act of removing data anomalies in order to make the database consistent. It lessens the amount of redundant data. In addition, it can make transitions while preserving qualities such as the atomicity of data, its isolation, and its durability. In addition, numerous users can view and perform operations on the data simultaneously. Another crucial aspect is that RDBMS prevents unauthorized data access, which is a significant safeguard. In order to keep unauthorized parties from accessing the data, it is feasible to set up individual user accounts with varied permission levels (Mandapuram, 2016).

What exactly are OODBMSs?

An object-oriented database management system (OODBMS) is a type of database management system that combines the capabilities of traditional database management

systems with those of object-oriented programming languages (Thuraisingham et al., 1994). In an OODBMS, the programmer can use object-oriented ideas such as inheritance, polymorphism, and encapsulation and database features like atomicity, consistency, isolation, and durability. These databases offer benefits like the extendability of systems, the reusability of code, and the maintainability of their applications.

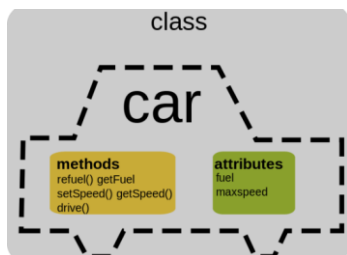


Figure 2: Objects of OODBMS

Object-oriented database management systems (OODBMS) are compatible with various object-oriented programming languages, including Python, JavaScript, Perl, Ruby, Java, C#, Visual Basic, and C++. Object-oriented programming languages serve as a model for this type of language. Due to the proliferation of web-based technologies, most commercial enterprises now employ OODBMS to represent their complex data. Because it depicts data in the form of objects, businesses that use computer-aided design (CAD) and other multimedia technology can benefit from its implementation.

When choosing between relational and object-oriented databases for your project, it is essential to have a solid understanding of the project's goals and the limits imposed by your budget (Mandapuram, 2017b). Understanding these criteria can assist you in making an educated decision regarding which sort of system is most suited for the requirements of your project (Lungu et al., 2009). Different jobs call for the use of different kinds of methods in order to process results efficiently.

ADVANTAGES AND DISADVANTAGES

Advantages of Object-Oriented Database

Due to their data structure, object-oriented databases (OODB) can store, manage, and retrieve vast volumes of complex data efficiently. Complex objects like multimedia, films, photos, and documents benefit from OODBs. Object-oriented databases have several benefits:

- Faster access speeds and more efficient storage when managing vast volumes of data due to object transactions that work faster than relational databases that break down object transactions.
- More accessible interface with other systems because it can handle complex objects without significant development changes.
- Creating and storing real-world items lets the database easily reflect hierarchical data relationships.
- Comparison queries simplify searching and retrieval.
- Cost-effective scalability by adding processing power or memory capacity as needed.
- Unlike relational databases, queries retrieve a complete object at once.

Disadvantages of Object-Oriented Database

Compared to Relational Database management systems, OODBMS has some restrictions. OODBMS performance could be better because relational databases can index subset tables to increase query latency. ORACLE found that OODBMS query results are often much slower than relational databases. Object-oriented databases utilize navigational methods, while relational databases require SQL queries, which can be a strength or a weakness depending on the user's SQL skills (Purao et al., 2002). Object-oriented databases may be costly. Even though hardware and software costs may be lower than relational options, the extra resources and knowledge needed for program development and integration may exceed relational system maintenance costs. Due to implementation complexity, it may only be suitable for some enterprise network users. Finally, object-oriented technologies can be incompatible when integrated into their environment or connected to non-object systems. For example, data exchange between outside sources may need extra work to complete commercial transactions.

Advantages of Relational Database

Relational databases are the most popular database type and are considered an essential component of many different types of organizations. They are widely used because of the reliability of their data structures, the consistency of their implementation, and their adaptability. The following are some of the benefits that come with using a relational database:

- Data Integrity—Referential integrity keeps related fields in multiple tables synced when changed or deleted. This ensures accurate results when joined querying many tables.
- Storage efficiency - Relational databases employ tables to store recurring data once, saving much space.
- Structured data — Relational databases store data in rows and columns in tables, making it easier to query and manipulate.
- Multi-User Accessibility — Locking mechanisms prevent concurrent users from damaging data.
- Scalability - Relational databases may be scaled by adding resources or physical components like servers or hard drives.

Disadvantages of Relational Database

Few Options: Relational databases can only store one type of data per field or record, making sophisticated data storage challenging. When making changes across many tables, they must all be updated to stay consistent, which can lead to more effort and mistakes if done incorrectly.

Data Complexity: Relational databases feature complicated data structures that require expertise to build, manage, and update. Due to their complexity, new users and programmers with database design knowledge may find them easier to learn and maintain.

Performance Issues: Relational databases might need to improve speed when maintaining massive datasets or conducting analytical queries against them because the database structure does not necessarily scale well with rising demands on memory and processing capacity.

DIFFERENCE BETWEEN RDBMS AND OODBMS

Object-Oriented Database Management System (OODBMS). It is an object-oriented DBMS. OODB supports classes, object identity, polymorphism, encapsulation, and inheritance. Object-oriented databases store detailed data. Versant, Objectivity/DB, ObjectStore, Caché, and ZODB are OODBMS. Relational Database Management System (RDBMS). A relational database management system stores data and relationships in interrelated tables. It lets users create, edit, manage, and interact with relational databases. RDBMS underpins SQL and current database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access (Ma, 2004).

Difference between OODBMS and RDBMS

BASIS OF COMPARISON	OBJECT ORIENTED DATABASE (OODBMS)	OBJECTED RELATIONAL DATABASE (ORDBMS)
Data Storage	The information is kept in the form of objects during storage.	Tables, which have both rows and columns, are used to store information in many forms.
Data Storage Structure	It finds object disk pages via indexing. They can persist in complex-structured objects.	Since each base relation is a separate file, it cannot provide persistent storage for complex-structured items.
Way of storing data	Stores data as Objects.	Entities, considered tables that hold technical information, are where the data is stored.
Quantity Of Data	Manages more extensive and detailed data than RDBMS systems.	Deals with data that is relatively straightforward.
Type Of Data	Object-oriented databases manage many data types.	Relational databases store one type of data.
Constraints	Each system supports different limitations.	Keys, entity, and referential integrity.
Data Complexity	Manages more extensive and detailed data than RDBMS systems.	Deals with data that is relatively straightforward.
Performance	Complex object-oriented queries are processed more quickly by OODBMS.	Complex queries that involve numerous tables can be handled quickly and easily using RDBMS.

Flexibility	OODBMS' object-oriented design makes it adaptable.	RDBMS schema limits flexibility.
Data Persistence	Object-Oriented Database Management Systems store data in objects in memory or disk.	The data is saved in tables on the disk using RDBMS.
Grouping	Class is a group of objects with similar relationships, behaviors, and features.	Entity types are entities with a standard definition.
Data Handling	Contains both the data and the procedures for using it.	RDBMS is used to store data.
Scalability	OODBMS has Extremely scalable as a result of its flexible schema.	Because of its inflexible schema, RDBMS has a limited capacity for scaling.
Concurrency Control	OODBMS is equipped with an optimistic kind of concurrency control.	RDBMS is capable of providing fine-grained locking.
Data Relationships	Complex object-oriented queries in OODBMS are quicker.	RDBMS tables and foreign keys store relational data.
Examples	Versant, db4o, Objectivity/DB	SQL Server, MySQL, Oracle

CONCLUSION

In conclusion, relational and object-oriented databases each have their own benefits and drawbacks, all of which need to be considered before determining which kind of database would be most appropriate for a specific endeavor. Relational databases are highly structured and offer excellent searching capabilities, whereas object-oriented databases allow developers to represent more complicated items and their interactions. Relational databases are the more common type of database. Ultimately, the choice should be determined by the application's requirements; other factors, including cost, scalability, and ease of development, will also play a role in the decision. RDBMS stands for Relational Database Management System, which is a Database Management System that is based on the relational model. On the other hand, OODBMS is for an Object-Oriented Database Management System, which is a Database Management System that facilitates the generation and modeling of data as objects. RDBMS operates under the premise of the relational model, whereas OODBMS operates under the assumption of the object-oriented model.

REFERENCES

- De, P., Sinha, A. P., & Vessey, I. (2001). An empirical investigation of factors influencing object-oriented database querying, *Information Technology and Management*, 2(1), 71–93.
- Dietrich, S., & Urban, S. (2010). *Fundamentals of object databases: Object-oriented and object-relational design*, Springer International Publishing AG.
- Leitheiser, R. L., & March, S. T. (1996). The influence of database structure representation on database system learning and use: *JMIS. Journal of Management Information Systems*, 12(4), 187.
- Leitheiser, R. L., & March, S. T. (1996). The influence of database structure representation on database system learning and use: *JMIS. Journal of Management Information Systems*, 12(4), 187.
- Lungu, I., Velicanu, M., & Botha, I. (2009). Database Systems - Present and Future. *Informatica Economica*, 13(1), 84-99.
- Ma, Z. (Ed.). (2004). *Advances in fuzzy object-oriented databases: Modeling and applications*. Idea Group, Inc.
- Mandapuram, M. (2016). Applications of Blockchain and Distributed Ledger Technology (DLT) in Commercial Settings. *Asian Accounting and Auditing Advancement*, 7(1), 50–57. Retrieved from <https://4ajournal.com/article/view/76>
- Mandapuram, M. (2017a). Security Risk Analysis of the Internet of Things: An Early Cautionary Scan. *ABC Research Alert*, 5(3), 49–55. <https://doi.org/10.18034/ra.v5i3.650>
- Mandapuram, M. (2017b). Application of Artificial Intelligence in Contemporary Business: An Analysis for Content Management System Optimization. *Asian Business Review*, 7(3), 117–122. <https://doi.org/10.18034/abr.v7i3.650>
- Purao, S., Jain, H. K., & Nazareth, D. L. (2002). An approach to the distribution of object-oriented applications in loosely coupled networks: *JMIS. Journal of Management Information Systems*, 18(3), 195–234.
- Thuraisingham, B., Sandhu, R., & Ting, T. C. (Eds.). (1994). *Security for object-oriented systems: Proceedings of the OOPSLA-93 conference workshop on security for object-oriented systems, Washington dc, USA, 26 September 1993*. Springer London, Limited.
- Ya-Hui, C., & Raschid, L. (2000). Producing Interoperable Queries for Relational and Object-Oriented Databases. *Journal of Intelligent Information Systems*, 14(1), 51.

--0--