Object- Oriented Databases and Web Databases: A Review of Concepts and Applications

Musa Ibrahim A.
Department of Computer Science
University of Nigeria, Nsukka

Okoronkwo Mathew C.
Department of Computer Science
University of Nigeria, Nsukka

Nnenaya Ogbonnaya
Department of Computer Science
University of Nigeria, Nsukka

Ekle Francis A.
Department of Computer Science
University of Nigeria, Nsukka

ABSTRACT
Databases form part of the web as well as mobile applications and are ubiquitous in data centre. Over the years, databases have had a magnified importance because of their needs to store and access information over the World Wide Web and in data intense applications. The functions of databases revolve around the design of such databases using specific models. Object-oriented paradigm has gained tremendous popularity in application programming in recent times owing to its features like abstraction, encapsulation, inheritance and polymorphism which has made software design and development easier, faster, less expensive and extensible. In this paper, we examined the Object-oriented concepts and their application in object-oriented database system and web databases. The underlying concepts and evolutions were discussed as well as the practical applications with a view to enhancing thorough understanding of the subject matter. Literature reviewed showed that the advent of the object- oriented database has proved to be relevant for cases in which designs are complex, the need for constant evolution of data exists and also in cases where collaborative development is required; furthermore, web databases are quite important in the modern management of information and web data.

General Terms
Object- Oriented Databases

Keywords
Object-oriented, Databases, web databases, web data.

1. INTRODUCTION
The emergence and constant innovation in the field of information systems and management is an integral element of the modern world and is based squarely on the interaction of different kinds of data. Research shows that information management practices—such as acquiring, classifying, storing, sharing, and timely using various data sources—have been found to directly and positively affect a firm's innovative performance, financial performance, and growth (Azevedo, 2024). Moreover, in the AI domain, there has been a shift toward data-centric AI, which emphasizes not just algorithms or models, but the quality, structure, and dynamism of data itself as central to system performance (Jarrahi et al., 2022). Together, these findings support the claim that evolution and persistent innovation in information systems and management derive largely from how different kinds of data interact (structured, unstructured, internal vs. external, high vs. low quality) and how systems and organizations adapt to harness that interaction effectively. Databases serve as the foundational infrastructure of modern information systems: nearly every web or mobile application depends on some form of database to function, and database management systems are critical from both business and technological perspectives (Akhtar, 2023). In data centers, whether on-premises or cloud-based, databases are omnipresent and they are used to store diverse kinds of records—such as educational, medical, enterprise, governmental, and public service data—ensuring accuracy, reliability, and accessibility (Akhtar, 2023; Deng, 2024). Over time, databases have become essential repositories for information pieces that are both interrelated and heterogeneous, due to the growing need to store and access diverse data types via the World Wide Web and in data-intensive applications. These systems are valued not only for their ability to manage varied data formats and interconnected data entities (e.g., relational, document, semi-structured), but also for their ease of maintenance and strong performance, which have driven much of the growth in database technologies (Putrama, 2024; Huang, Qin, Zhang, Tu, & Cui, 2023; Taipalus, 2023).

The operation of database systems is fundamentally based on the design of the database using specific data models, which enable interaction among all system components through an understanding of schema and data. These models include the Entity-Relationship (E-R) model, the Hierarchical model, the Network model, semi-structured data models, the Relational model (RDBMS), object-based databases (OBDMS), object-oriented databases (OODBMS), and object-relational databases (ORDBMS) (Hevo Data, 2024; GeeksforGeeks, 2025).

This paper examines object- oriented databases in the context of web data. The underlying concepts and evolution were discussed as well as the practical applications.

2. THE CONCEPT OF OBJECT-ORIENTED PROGRAMMING
Object-oriented programming (OOP) is a programming paradigm in which software design is organized around data or "objects" rather than functions and logic (Gillis & Lewis, 2024). According to (TechTarget, 2024),
Object-oriented programming (OOP) revolves around using classes and objects to build models that reflect the real world. In this paradigm, a class serves as a blueprint defining both the data (attributes) and behaviors (methods) that its objects (instances) will have. An object-oriented program is composed of many such objects, which interact by sending and receiving

messages (method calls) to request services or information from one another.

Complete object-oriented programs are developed by decomposing functionality into distinct classes, defining the interfaces for each class, and establishing connections and interactions between components through these interfaces (Alsaeedi & Khan, 2019; Xu et al., 2021). This methodology promotes the separation of program concerns, allowing developers to focus on individual components independently, which simplifies the development process and enhances collaboration within a software team (Adewumi et al., 2020). Object-oriented programming (OOP) models a system as a set of objects that can be controlled and manipulated in a modular manner, enabling reusability, maintainability, and scalability (Pérez-Pons et al., 2022). By building self-contained modules or objects that can be easily replaced or modified, OOP supports abstraction and encapsulation, which improve code understanding and long-term maintenance (Sharma & Tiwari, 2021). Because OOP mirrors real-world entities and interactions, it continues to gain popularity for designing complex software systems where objects interact much like entities in the physical world (Xu et al., 2021). The core concept of OOP revolves around classes and objects, forming the foundation for modern software engineering practices (Alsaeedi & Khan, 2019).

i. Class

Object-oriented programming (OOP) is centered on the concept of data objects and the functionality associated with each object, with the class serving as the primary unit of programming in this paradigm (Gamma et al., 2022; Sebesta, 2024). A class encapsulates data elements (also known as attributes or members) and functions (called methods) that operate on those data (Larman, 2021). Classes provide the fundamental development structure of any OOP program by defining the properties and behaviors of objects and enabling information sharing between them (Meyer, 2022). Within a class, methods implement user-defined operations, while attributes define the characteristics of the data (Booch et al., 2021). The hierarchical organization of classes allows programmers to manage execution priorities and maintain clear, modular software designs (Gamma et al., 2022). Consequently, the class is regarded as the key element in software modeling, monitoring, and the overall design process in OOP (Sebesta, 2024).

ii. Objects

In object-oriented programming (OOP), an object is a component of a program that encapsulates both data (its attributes) and the operations (methods) that act on that data; it interacts with other parts of the program via well-defined interfaces. Objects are instances of classes, which serve as blueprints defining what attributes and behaviors (methods) those instances will have (Real Python, 2024; Techtarget, 2024). Encapsulation (i.e., restricting direct access to an object's internal state, often via private data) is essential for ensuring integrity and security in OOP systems by preventing unauthorized or unintended interference with the internal data of objects (Real Python, 2024).

3. THE CONCEPT OF OBJECT-ORIENTED DATABASES AND WEB DATABASES

At the core of the development and use of object- oriented databases is the trend in computer science towards the object-oriented paradigm which is essentially an approach to the design and implementation of programs based on the usage of a collection of entities referred to as objects (Costa & Santos, 2017). The development of object-oriented databases resulted from a bid to imitate the way data is processed in the real world. While data is stored traditionally in databases in the form of strings, real numbers and integers, object- oriented databases are more sophisticated in that they are a combination of attributes and methods. Alzahrani (2016) conceptualized attributes as the data which defines that characteristics of an object and which may be defined in terms of a lexical class such as integers or strings or non- lexical class such as an entity-valued attribute. Methods as described by Damesha (2015) are the procedures and functions which guide the behavior of an object. The implication of this is that an object-oriented database contains both executable data and code. This combination is perceived among database designers and managers as responsible for characteristics such as custom data models, capability for modelling complex data structures, faster access time and ease of extensibility which position object- oriented databases as better than other types of database systems.

In evolutionary terms, object- oriented database systems were introduced in 1985 and their development was sped up by the widespread criticism of the relational model as being relatively inadequate in terms of meeting the demands for new applications (Damesha, 2015). This early period marked the adoption of characteristics and modes which would later define the use of object- oriented databases all over the world. As reported by de Sousa and del Val Cura (2018) the early period of developing object- oriented databases marked the incorporation of two criterion as essential before defining a database as object- oriented.

The first criterion is the incorporation of the five features; persistence, concurrency, data recovery, ad hoc query facility and secondary storage management as essential aspect of such databases (Cheng & Hurson, 1991; Dietrich & Urban, 2010).The second criterion revolves around the idea that an object- oriented database should exhibit requisite properties such as polymorphism, inheritance, extensibility and encapsulation (Damesha, 2015). By 2004, the first free open-source object- oriented database management system, db4o was released implementing native queries in languages such as C+ and

Java and LINQ and DLINQ have also been introduced by Microsoft to support native queries.

Web databases also form an important component of the 21st century schema for data management. According to Caprita and Mazilescu (2005), web database is described as a database which exists entirely on a dedicated or set of dedicated internet servers. For such databases, the access to information revolves around the utilization of a password system and a hierarchical mode of privileges stipulating who can access different kinds of information (Cheung et al, 2014). Historically, the development of web databases was triggered by the need to have better platforms for collecting, delivering and sharing of information among points which are geographically dispersed globally. This overtime led to the development of tools which would allow computer users in different geographical locations to utilize client-server database systems to access computer systems containing legacy data.

By the late 1990s and early 2000s, web databases are advanced and helped by increasing capabilities of the internet leading to the development of such databases connector tools like MySQL, Front Page, Active Server, Java Servlets and Dream Weaver among others (Cheung et al, 2014). In contrast with object- oriented databases which are based on the defining features of persistence, concurrency, data recovery, ad hoc query facility and secondary storage management, web databases are geared towards features such as concurrency, resource sharing, fault- tolerance, scalability, openness and transparency.

4. THE MERITS AND DEMERITS OF OBJECT- ORIENTED DATABASES

Of importance in considering the utilization of object-oriented databases in information systems is the rationale behind their fast adoption. Ogunlere and Idowu (2015) and Costas and Santos (2017) expressed that these reasons include the limitations of RDBMS, the increased popularity of the object-oriented paradigm and the need for advanced applications. The limitations of the RDBMS include the emphasis on normalization, poor integrity support, poor schema which impedes long duration transactions and easy navigation and the overloading of semantic structures. The fact that these drawbacks are addressed by object-oriented databases serve as the basis for its merits. Below are some of the merits attributed to the use of object- oriented databases.

1. OODBs are supportive of user- defined data types.

2. OODBs rely on the use of object identity unlike other systems which use keys

3. OODBs are supportive in terms of the creation of new relationships

4. OODBS reduce significantly the need for joins

5. OODBs are designed for nested and long- duration transactions

6. Control mechanisms for versioning which are not supported by RDBMS are supported by OODBs

7. OODBs have higher level of performance than RDBMs

While OODBs and their merits as stated above have impacted significantly the processing of information, there are also a few weaknesses which have been associated with OODBs. The first weakness associated with OODBs is that they are poor in terms of the optimization of query expression. Ogunlere and Idowu (2015) explained that OODBs have difficulty in optimizing queries which are important for performance purposes because of issues such as multiple object identities, user- defined data types, the nested language nature of object query language, the changing variety of data types and the process of encapsulation. Also, another limitation of object- oriented databases is that query optimization is limited because there is no fixed query algebra for object- oriented databases (Damesha, 2015). Other limitations include the fact that OODBs do not have full-fledged query systems, poor level of coherency with relational databases, poor implementation of consistency constraints mechanisms, lack of basic features such as metadata management and triggers (Dittrich & Dittrich, 1995; Uduak et al, 2012). Lastly, a key weakness of OODBs relates to security. In information systems, the concept of security is related to authorization, authentication and accounting, however OODBs do not have the facility for authorization which generates different security problems (Alzahrani, 2016).

Web databases have quite differentiating features from the OOBDs and as such have different set of merits for justifying their use. Generally, web databases have password access and privilege- based restrictions and can contain images, links and hyperlinks to other pages in different databases (Ramachandra, 2015). Also, based on their unique architecture web databases can be readily used to download database files which can be read by a local computer even when the downloaded pages are dynamically personalized. The main merits of web databases thus include features such as security, modularity, reusability, maintenance and updating and the embedded capability for the distribution of data updates (Caprita & Mazilescu, 2005).

Web data has close interrelationship with the object- oriented approach. Web data is conceptualized as the different kinds of data generated on the World Wide Web and includes documents, applications, and media. As explained by Martinez-Mosquera et al (2020) web data can be delineated into structured, semi- structured and unstructured in nature and exist in a myriad of languages. Further, the myriad languages in which web data exist necessitated the development of data exchange formats which can then be used in translating web data into a form which is utilized in object- oriented databases (Uduak et al, 2012). On the one hand, object-oriented databases serve as repository for documents, query answers and mediators while feedback from the web and its constant evolution serve to expand these databases. In recent times however, there have been concerns over the fact that web data is increasingly becoming schema- less (irregular and heterogeneous) hereby negating the usefulness of a database model such as the object- oriented database system (Dell' Aglio, Balduini & Della Valle, 2015).

5. APPROACHES AND APPLICATIONS OF OBJECT- ORIENTED DATABASES AND WEB DATA

An important facet of object-oriented databases in terms of their use for information processing is the approaches adopted by information designers and data processing practitioners. These approaches are pure OODBMS, object/relational DBMS and relational extension-based DBMS. Object-oriented databases which are based on pure OODBMS are characterized by the fact that they have followed no standards (Costas & Santos, 2017). According to this approach an object-oriented database is built based on a model designed to capture the semantics of objects encapsulated in an object- based programming design. The relational extension-based DBMS was the earliest object- oriented database approach and it works by extending the features of a relational database to an object-oriented one. Alzahrani (2016) reported that the design techniques for this approach revolves around the definition of new data types and operations, creation of new access methods and the optimization of query processing. The merit of this approach was that it allowed for the supporting of variable length and undefined data values which allowed for the support of data types such as LONG, LONGRAW and RAW (de Sousa and del Val Cura, 2018).

The object/relational approach is an amalgamation of relational and object- based paradigms and has a mode of operation which involves the building of an object layer on top of a system which can be defined as relational such as ODAPTER and Open ODB. Important features of this approach which allows for its wide applicability includes its basis on Iris DBMS and client/server architectures as well as the use of OSQL as data manipulation language, the fact that clients can use Application Program Interface (API) to access information and the advantage that data and applications can be shared by user applications (Barry, 1997; Hamouda & Zaino, 2017).

The approaches defined above and the features underlying them provide basis for the myriad applications to which object-oriented databases have been put. The need for advanced applications which have varying a higher level of complexity and the realization that these applications cannot be built on relational databases have contributed to the widespread adoption of object- oriented databases. A key application of OODBs is Computer Aided Design (CAD). Dietrich and Urban (2010) and Damesha (2015) explained that CAD applications require relevant data about integrated circuits, airplanes, buildings and other structures. The use of object- oriented databases for these applications allow for the storage of large designs and the need to cope with ever- evolving designs through updates. Also, CAD designs require features such as the management of configuration, version control and complex objects which might be related to each other. These features are best based on object- oriented databases which also support collaborative engineering and long duration transactions (Ogunlere & Idowu, 2015).

Another important application of objective- oriented databases relates to their use in Computer Aided Manufacturing (CAM) and Computer Aided Software Engineering (CASE). The utilization of OODBs for these applications is premised on their requirements of discrete production like CAD, real- time events and the linking of algorithms and rules to specific situations (Cheng & Hurson, 1991). Also, these applications are based on designs that are often large, requiring collaboration, configuration management, versioning and the cogent need of maintaining dependencies among various components (Martinez- Mosquera et al, 2020). Object-oriented databases also have demonstrated application in network management systems, geographic information systems, multimedia systems and office information systems. As explained by Alzahrani (2016), their use in network management is premised on their amenability to network planning, problem management and network path management.

For web databases, the approaches to their design often revolve around the choice of a two- tier or three- tier application. The use of architecture is premised on the consideration that a web database is an amalgam of a single huge database and multiple data sources. As explained by Caprita and Mazilescu (2005), the two- tier architecture is the minimal spatial configuration upon which a web database can be built resembling the traditional client- server paradigm. This architecture is built using technologies such as XML, SQL and JDBC. The more complex three- tier architecture is made up of components such as the client server, the application server and the data server. This is more complex than the two- tier model and it involves a complex relationship between components and functions such as client- side APIs, screen logic and handling, validation functions and the translation and mapping mechanisms (Cheung et al, 2014).

Based on these architectures, web-based catalogues are applied extensively in the creation of product catalogues, maintenance of client and user details, time- bound and frequent communications and the back- end for e- commerce functions, online learning, design and implementation of collaborative projects and more predominantly social media communication.

In terms of application, there is a close linkage between the object- oriented approach, object- oriented databases and web data. The utilization of web data in building systems is based on the sequence of requirements analysis, architectural design, application design and analysis. However, achieving this relies on the use of object- oriented databases which addresses the three germane and pervasive problems of flexibility, productivity and quality (Uduak et al, 2012). The use of object-oriented approaches and databases also has applications in situations in which there are a need to reduce errors and increase flexibility. Further, while web data have important applications especially in this age where data mining is required as a basis for generating insights, object-oriented database is instrumental in managing the different programming relationships on which such applications are based (Dietrich & Urban, 2010).

6. CONCLUSIONS

The importance of databases and the web cannot be over-emphasized in the modern information age. Essentially, the transmission of information from one point to another necessitates the development of databases which would basically act as the repository for such information. Databases can be described as the nuts and bolts which hold together modern information systems. Databases form parts of web and mobile applications and are ubiquitous in data centers where they are utilized in maintaining various types of records. The advent of the object-oriented database has proved to be relevant for cases in which designs are complex, the need for constant evolution of data exists and in cases where collaborative development is required.

The utilization of object- oriented databases has also developed in order to remedy the limitations of relational database systems. While object- oriented databases have weaknesses revolving around certain protocols and security, the main issue with their use is the uncertainty of their relevance in an age when data from web is increasingly becoming schema- less, incongruous and heterogeneous in nature. Evidence also suggests that web databases are quite important in the modern management of information. Factors such as the high level of internet penetration, the shift towards digital handling and management of information even among traditional institutions, globalization and the development of business models that are best described as web-based have increased the importance attached to web databases and this importance will become more magnified in the future. This will in turn led to continuous improvement in the architecture of web databases.

7. REFERENCES

[1] Adhikari, B. (2016). Object Oriented Programming Vs Procedural Programming. https://doi.org/10.13140/RG.2.2.33443.45604

[2] Bouktir, T., Slimani, L., & Mahded, B. (n.d.). Object-Oriented Software of Power System Analysis.

[3] Butler, S., et al. (2011). Mining java class naming conventions in Software Maintenance(ICSM).

[4] D'Andrea, R. J. and R. G. G. (1990). Object-Oriented programming: concepts and languages in Aerospace and Electronics.

[5] Heung Eeok, C. and K. Y. R. (1998). A Cohesion measure for classes in object-oriented systems. in software Metrics Symposium.

[6] Hiremath, S., & Tavade, C. M. (2016). Review Paper on Inheritance and issues in Object Oriented Languages. 2. www.ijaetmas.com/wp-content/uploads/2016/05/NCICT-150.pdf

[7] Jackson, M. A. G. and M. S. (1989). Introduction to Object-Oriented Programming Pure Object-Oriented Languages. Information and Technology, 1–24.

[8] Kölling, M., & Rosenberg, J. (1996). An object-oriented program development environment for the first programming course. ACM Sigcse Bulletin, 28, 83–87. https://doi.org/10.1145/236462.236514

[9] M, A. R., D, K. M., J, P. S., & M, S. C. (2016). Object-Oriented Programming and its Concepts. IJSRD - International Journal for Scientific Research & Development|, Vol. 5(Issue 09).

[10] Maney, H. . (1994). An object-oriented synthetic aperture rader procesor implemented using C++.

[11] Mclntyre, S. C. and H. (1998). Object-Oriented Systems Analysis and Design: Methology and Application.

[12] Omar, N. and N. A. R. (2008). Determining the basic elements of object oriented programming using natural language processing in Information Technology.

[13] Rani, A., & Parvathy, M. K. M. D. (2017). Object-Oriented Programming and its Concepts GSSS Institute of Engineering and Technology For Women , Mysuru. 5(09), 840–842.

[14] Seban, R. R. (1994). An overview of object-oriented design and C++. Proceedings. In Aerospace Applications Conference, 1994.

[15] Urdhwareshe, A. (2016). Object-Oriented Programming and its Concepts. International Journal of Innovation and Scientific Research, 26, n(1).

[16] Alzahrani, H. (2016). Evolution of object- oriented database systems. Global Journal of Computer Science and Technology: Software & Data Engineering, 16(3), 1-5.

[17] Barry, D. K. (1997). The truth about object databases. Database Programming & Design, July, 44- 48.

[18] Caprita, A. D. & Mazilescu, V. (2005). Web- based database distributed systems. Economy Informatics, 1-4, 126- 131.

[19] Cheng, J. R. & Hurson, A. R. (1991). Effective clustering of complex objects in object- oriented databases. Proceedings of International Conference on Management of Data (Sigmod Record), 20(2), 22- 31.

[20] Cheung, A., Arden, O., Madden, S., Solar- Lezama, A. & Myers, A. (2014). Using program analysis to improve database applications. IEEE Data Engineering Bulletin, 37(1), 48- 59.

[21] Costa, C. & Santos, Y. (2017). Big Data: State of the art concepts, techniques, technologies, modelling approaches and research challenges. International Journal of Computer Science, 44, 1- 17.

[22] Damesha, H. S. (2015). Object- oriented database management systems: Concepts, advantages, limitations and comparative study with relational database management systems. Global Journal of Computer Science and Technology: Software & Data Engineering, 15(3), 1- 9.

[23] De. Sousa, V. & del Val Cura, L. M. (2018). Multimedia Big Data Analytics: A survey. ACM Computer Survey, 51(10), 1-12.

[24] Dell' Aglio, D., Balduini, M. & Della Valle, E. (2015). Applying semantic interoperability principles in data stream management. Data Management in Pervasive Systems, 135- 166.

[25] Derrett, N., Kent, W. & Lyngbaek, P. (1985). Some aspects of operations in an object- oriented database. Database Engineering, 8(4), 1- 10.

[26] Dietrich, S. & Urban, S. (2010). Fundamentals of Object Databases: Object- oriented and object- related design. New York, NY: Morgan& Claypool.

[27] Dittrich, K. A. & Dittrich, K. R. (1995). Where object-oriented DBMs should do better: A critique based on early experiences. In Kim, W (ed.) Modern Database Systems: The Object Model, Interoperability and Beyond, 238- 252.

[28] Hamouda, S. & Zaino, Z. (2017). Document- oriented data schema for relational database migration to NoSQL. In Proceedings of the 2017 International Conference on Big Innovations and Applications, Prague, Czech Republic, August, 2018, 43- 50.

[29] Martinez- Mosquera, D., Navarrete, R. & Lujan- Mora, S. (2020). Modeling and Management Big Data in Databases- A systematic literature review. Sustainability, 12(634), 1- 41.

[30] Ogunlere, S. O. & Idowu, S. A. (2015). Comparison analysis of object- based databases, object- oriented databases and object relational databases. Asian Journal of Computer & Information Systems, 3(2), 52- 57.

[31] Ramachandra, K. (2015). Program transformations for asynchronous and batched query submission. IEEE Transactions on Knowledge and Data Engineering, 531-544.

[32] Uduak, U. A., Nwachukwu, E. O., Umoh, A. A. &Isong, E. B. (2012). A web- based database system: An industrial application. Research Journal in Engineering & Applied Sciences, 1(3), 203- 208.