Framework DataBase (FDB): A NoSQL Database Model for structured and semi-structured data

EVANGELIA N. PETRAKI
Department of Economics
National and Kapodistrian University of Athens
Athens, GREECE

Abstract: —The massive collection of data that has been taking place in recent years, in multiple ways and at an unimaginably fast pace, has led to the need to store and retrieve information from both structured and semi-structured or unstructured data. While the relational database model is widely used and provides efficient ways to organize structured data, manage it, and search for information through SQL commands, it has limitations and is not a suitable model for organizing and storing semi-structured data or data that lacks structure. This paper briefly presents the FDB model, a NoSQL database model that provides flexibility in defining any database schema, overcomes the limitations of the relational database model, while at the same time offering the possibility of multilingual conceptual searches in the database data, utilizing the information derived from multilingual thesauri. This paper presents the architecture of the FDB model, the thesaurus management and conceptual search algorithms, the process of converting a relational database schema to the corresponding FDB database schema, the advantages of the model, the ability to host semi-structured databases, as well as conclusions and future directions.

Key-words:—Framework Database, FDB, relational model, NoSQL, thesaurus, conceptual search

Received: May 26, 2025. Revised: June 29, 2025. Accepted: August 9, 2025. Published: December 31, 2025.

1 Introduction

A huge volume of data is constantly being generated from multiple sources, such as social media, mobile devices, sensors, cameras, etc. The format and the data type of the data vary depending on its nature and origin. In the literature, the term **big data** is used to refer to the huge and continuous accumulation of data of different types. Various definitions have been proposed for the term big data, which emphasize the fact that it is data that is very large in volume or complicated to analyze and manage with conventional methods, data characterized by great variety, variability and enormous size, data that is complex and it is necessary to use special methods to process and manage it [1], [2], [3],[4],[5].

The characteristics used to describe big data which are briefly referred to as the 5Vs, are a) Volume, b) Velocity, c) Variety, d) Veracity, and e) Value [2],[4] while other scientific reports focus on only the first three characteristics, the 3Vs [3]. Volume refers to the amount of data generated every second and is related to the size and scale of the data. Velocity refers to the speed at which data is produced, but it also concerns the time and delay in handling it. Variety represents the structural heterogeneity of different types and sources of data. Veracity describes the completeness, accuracy, and quality of the data and is related to its reliability and consistency. Value is about how important the data is. Unlike the previous V's that have more to do with big data challenges, the value is what organizations are looking for - business value - and focuses on the usefulness of data in decision-making.

For the storage, organization, analysis, and management of data, its structure plays a crucial role. According to the degree of structure, data is divided into structured, semi-structured, and unstructured. Structured data management and representation are done in a strict and well-defined way using a data model, mainly the relational model. In semi-structured data, the structure can be partial, undefined, or determined by the data itself. In unstructured data, there is no information or very little information about the format and data type.

The relational data model was proposed by Codd, and it is based on mathematical foundations but has its own terminology [6]. The relational model is the most established in the field of databases, and it is ideal for structured data management. The 4th generation language, SQL, is used for defining and manipulating relational databases. The major disadvantage of the model is that any change or addition of new requirements to the database takes time, as the changes require modifications to the database schema.

The relational data model is not the ideal model in the case of semi-structured and unstructured data for which NoSQL databases are appropriate. In the modern era of big data and cloud computing, applications require database technologies that allow quick and easy database schema scalability, a capability that non-relational databases offer. NoSQL refers to Not Only SQL databases and provides a mechanism for storing and retrieving unstructured, or non-relational, data. These databases do not require a predefined table format and are horizontally scalable while allowing the coexistence of relational and non-relational databases.

They are designed to ensure automatic data distribution and fault tolerance.

In NoSQL databases, the data model is simple, the database schema is not fixed, and there is the possibility of scalability by dynamically adding new properties. They are reliable and support distributed environments with high availability. For this reason, Google, Facebook and Amazon, eBay, and others use NoSQL environments as their main databases [7].

The types of NoSQL databases that have been developed over the years are [7], [8], [9]:

A) Document-Oriented Databases (DOD): Store data in documents and encapsulate them using a standard internal format. The document model (Figure 1) can be considered a superset of other data models. It is used mainly for storing data in XML, JSON, and BSON type files. MongoDB [10], Apache CouchDB [11], Terrastore [12], OrientDB [13], and Amazon DocumentDB (with MongoDB compatibility) [14] are examples of document-oriented databases.

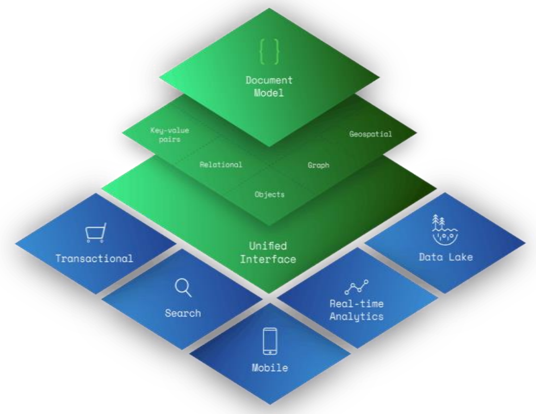

Fig. 1. Document model source: [15]

B) Key-Value databases (Figure 2): It is among the most popular NoSQL databases. They provide mechanisms for storing data as pairs (key, value) across multiple servers. Amazon DynamoDB [16] is an example of key-value database.

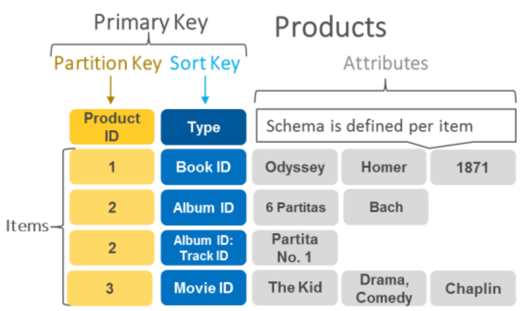

Fig. 2. An example of data stored as key-value pairs source: [17]

C) Column-oriented databases (Figure 3): Data is represented and stored in the form of column families. Apache Cassandra is an example of a partitioned wide-column storage model. It is an open-source distributed NoSQL database and it was initially designed for Facebook [18].

Row-oriented

ID	Name	Grade	GPA
001	John	Senior	4.00
002	Karen	Freshman	3.67
003	Bill	Junior	3.33

Column-oriented

Name	ID
John	001
Karen	002
Bill	003

Grade	ID
Senior	001
Freshman	002
Junior	003

GPA	ID
4.00	001
3.67	002
3.33	003

Fig. 3. Column-oriented vs row oriented database source: [19]

D) Graph Databases (Figure 4): They facilitate the storage of entities (or nodes) and the connections or relationships between them. OrientDB [13], Neo4j [20], and ArangoDB[21] are examples of Graph Databases.

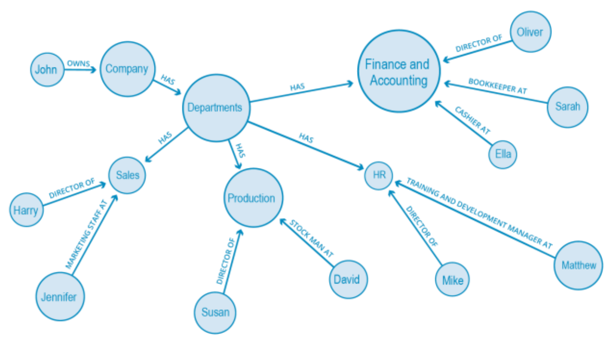

Fig. 4. A graph database using nodes and relationships source: [22]

Several of the above NoSQL databases, like OrientDB, are multi-model and support graph, document, key-value, etc models.

The next section briefly presents the Framework DataBase (FDB) model, a set-theoretic data model that overcomes the limitations of the relational data model as it suggests a universal schema that allows the evolution of the logical schema of any database without affecting the data already existing in the database and provides the possibility of conceptual searches through its multilingual thesauri.

2 The FDB model

The Framework Database (FDB) model (Figure 5) is a set-theoretic model that supports dynamically evolving Database environments [23]. It is based on a fully parametric structure that allows modifications to any database schema to be managed without requiring modifications to the existing universal schema of the framework. FDB is based on a unified and universal schema that can host any database. The basic structural components of the FDB model are based on the mathematical theory of unordered sets and consist of [23]:

a) frame entity: the unordered set of entities that participate in the logical schema

b) frame object: a set of all frame entities' objects

c) Tag: the set of attributes that describes a frame entity.

d) subtag: the set of simple atomic attributes that constitute existing complex tags

e) domain, which is used to define the set of all data domains

f) language, message, vocabulary: sets of strings that are used to define languages and corresponding messages

The definition of any FDB database is carried out by specifying the appropriate metadata, while the changes that may occur in the schema do not affect the existing structure (the universal schema), but concern the definition of the corresponding metadata. Thus, FDB offers a suitable and flexible database management environment. The universal schema of FDB allows the definition of any database by defining the appropriate metadata.

Additionally:

- It is possible to dynamically define objects without requiring reorganization of the logical schema.
- Objects with a variable number of fields (tags) and subfields (subtags) can be managed.
- Variable-length fields and subfields can be defined.
- Repeated fields and subfields are defined in a controlled way.
- Relationships between entities are supported.
- Redundancy of data is avoided, as well as the use of "null" in fields and subfields.
- Multilingual databases can be defined at both the data level and interface level.

Within the framework of the FDB model, one or more multilingual or monolingual thesauri can be defined, which can then form the basis for the implementation of conceptual searches in the FDB Databases [24]. With this feature, the FDB model provides the ability to perform searches both in the traditional way using keywords, as well as conceptual searches through multilingual thesauri.

Three thesaurus administration algorithms have been proposed for the FDB model: The first algorithm automatically correlates the FDB database frame objects (records) with terms derived from the thesaurus(s). The association of the records of a database with thematic terms allows the enrichment of search queries in the database with new terms derived from the thesaurus(s), achieving efficient conceptual searches. The correlation of the database frame objects with thesaurus terms is carried out to enhance searches in an FDB database, as it allows the enrichment of search queries in the database, with new terms derived from the thesaurus(es), achieving efficient conceptual searches. The second algorithm is used to enrich an existing thesaurus with terms derived from the FDB database, and the third one allows the creation of a set of terms that could form the core for a new thesaurus.

In conjunction with the correlation of the FDB database records (frame objects) with terms derived from multilingual thesauri, an advanced conceptual search algorithm is provided by the FDB model that allows queries to be enriched with synonyms, related terms, broader terms, or narrower terms derived from the FDB multilingual thesauri. At the same time, the algorithm allows the exploitation of opposite terms to limit the search results to records that are not relevant to what is being searched for. [25]

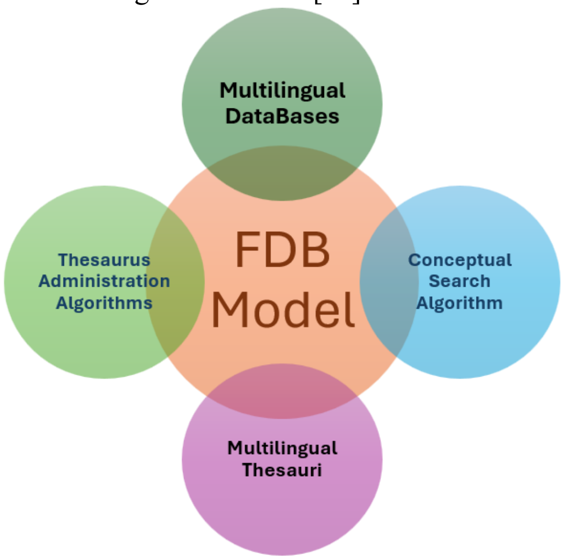

Fig. 5. The FDB model source: created by the author

The purpose of the current research is to present the process of converting a relational database into an FDB one. The steps that should be followed are presented in the next section.

3 Converting a relational database to the corresponding FDB one

As already mentioned above, the relational data model is widely used for the organization and management of structured data in an organization, and the SQL query language is used for performing queries in any relational database. The relational data model has many advantages, but also several limitations that were mentioned above. The FDB model is a NoSQL data model that overcomes the limitations of the relational model and provides the ability to define any multilingual database using the appropriate metadata. In addition, it offers the possibility of performing conceptual searches using the information provided by multilingual thesauri that can be defined in the same integrated FDB schema. The process of converting a relational database to the corresponding FDB can be summarized in a set of specific steps.

All the information that guides us through the process of converting a relational database schema to its corresponding FDB one is provided by the relational database data dictionary. The relational database dictionary, often called the data dictionary, is a set of metadata that provides information about the structure and organization of a relational database schema. The data dictionary describes all the database objects starting from the tables (relations), the fields they consist of, the data type of each field, the primary key of each table, any foreign keys, as well as any field constraints.

The basic steps of the conversion are numbered below:

1. An entity must be defined in the FDB model for each separate relational table. The entities that should be defined at the beginning of the conversion process are the tables for which there are no foreign keys.
2. For each field of each table, a new TAG must be defined, as well as its properties: data type, length, if it is mandatory, repeated, and if it refers to the TAG of another entity, in case it is a foreign key.
3. Finally, foreign keys are also defined in the AUTHORITY_LINKS object.

The conversion algorithm that follows takes as input the description of the relational database schema provided by the data dictionary and, as processing proposes the creation of metadata in the appropriate objects of the FDB model. A new element (RDBToFDB_Dict) is used to store the name of each table from the given relational database and the corresponding entity number it receives during the conversion, as well as for each field, the table name and the field name, along with the entity number and tag number it receives during the conversion.

Algorithm RDB_To_FDB
// input **T**(ables), **F**(ields), **C**(onstraints) in data dictionary
For each t ∈ T //for each table in the relational database
    Create an entity **e** in the ENTITIES object
    For each f ∈ F //for each field of table t
        Create a **TAG** in TAG_ATTRIBUTES
        // update of RDFToFDB dictionary
        Set RDBToFDB_Dict.table_name= **t**
        Set RDBToFDB_Dict.entity_number= **e**
        Set RDBToFDB_Dict.field_name= **f**
        Set RDBToFDB_Dict.TAG_number= **t1**
        // update of TAG_ATTRIBUTES
        Entity=**e** //the entity that the TAG belongs is e
        TAG= **t1** //a new tag number
        If f is mandatory, **then**
            t1.Occurrence = M
        Else
            t1.Occurrence=O
        End if
        If f is repeatable, **then**
            t1.Repetetion= Y
        Else
            t1.Repetetion=N
        End if
        If f is a Foreign Key **then**
            t1.Authority= Y
            Insert a row into AUTHORITY_LINKS:
            From_Entity = e
            From_Tag = t1
            To_Entity = e2 //entity number of related table
            To_Tag = t2 //tag number of related field
        Else
            t1.Authority=N
        End if
    End Loop //For each field
    End Loop //For each table
// data transfer
For each t ∈ T // for each table in the relational database
    For each f ∈ F //for field of table t
        Get entity and TAG number from RDBToFDB_Dict
        INSERT INTO FDB TAG_DATA object
        - entity_number
        - TAG number
        - Language
        - Data
    End Loop
End Loop

The above algorithm describes the process of converting the logical schema of a relational database into an FDB database schema, as well as the process of importing data into the new FDB database, which can be done seamlessly if there is a mapping list of relational tables - FDB entities and relational table fields – FDB TAGs.

When importing data into the FDB, the language of the data is also specified, while in the case of multilingual databases, this specific property distinguishes in a very simple way the data records of each different language.

In Figure 6 it is presented the data frame object with the frame object number 1, which stores in the FDB model the title (Digital Design and Computer Architecture), the abstract (Graduate-level courses …), and the author (David Harris) of a book. The frame object is in English (EN) and the FDB model allows the storage of the corresponding information regarding the specific book in other languages, e.g., in Greek (GR). The figure also denotes that the specific book is also related to three thesaurus terms (thesaurus term object numbers 1,3, and 8).

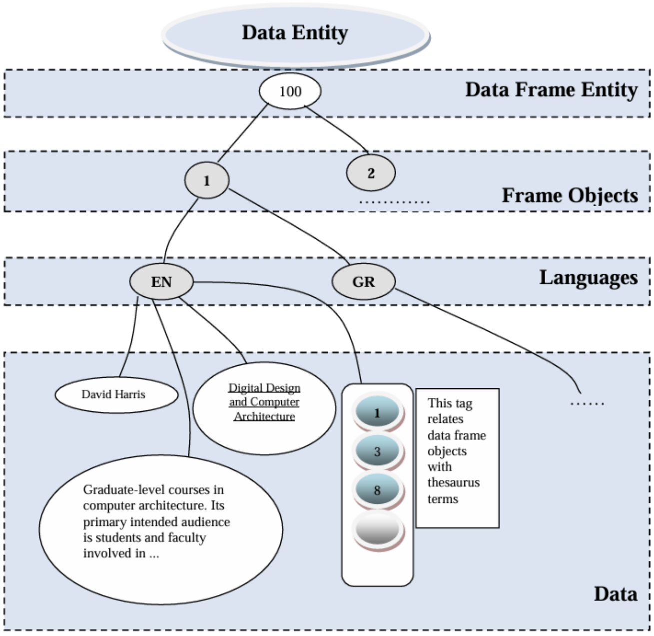

Fig. 6. Multilingual database instance in the FDB model source: [24]

During the conversion process, the RDFToFDB_Dict object is gradually updated by mapping relational tables and fields to FDB frame entity numbers and TAGs. The importance of this specific information is great both when defining the metadata of the FDB database and when transferring the data from the relational to the FDB database.

The complexity of the above conversion algorithm is linear, O(n), when n is the number of relational tables, as it depends on the number of tables in the relational database.

In the case of semi-structured data, the information that may exist about any structure and organization of the data, which is found within each data file, is important. Both the structuring of data in JSON files and in XML ones provide information about their structure through the appropriate tags: records and data fields. In the case of XML files, useful information is also provided by the XML Schema and can be used in the process of converting XML semi-structured data into FDB databases.

4 Conclusions

The relational data model has prevailed in the field of databases and is suitable for managing structured information and organizing data in relational databases. SQL query language is a standard and effective way to retrieve any information from relational databases. However, the relational data model has many limitations and lacks flexibility, a characteristic that is necessary, especially in the modern era, where huge volumes of data are collected in multiple ways and at very fast rates, either through the information systems of organizations or through social media, IoT devices, cloud services, etc.

The current research briefly presents the FDB model, a data model that does not have the limitations of the relational data model, and provides a universal schema that allows the definition of any multilingual database as well as any multilingual thesaurus. The FDB model provides advanced algorithms for the management of multilingual thesauri, the automated linking of data frame objects (data records) with terms derived from conceptual thesauri, as well as the possibility of conceptual searches through the thesauri. This paper presents the process of converting a relational logical database schema into an FDB one, as well as explores the possibility of storing semi-structured data from the JSON and XML formats in the FDB. Any change to the logical schema of the database in the FDB is only equivalent to defining the appropriate metadata while not affecting the data already stored in the database. At the same time, the ability to effectively organize multilingual data and perform conceptual searches on it provides a powerful and flexible environment for managing any database.

Further study is required to determine the algorithm for converting semi-structured databases to FDB, as well as to address specific issues related to semi-structured databases.

References

[1]. H. Basim Alwan, K.R. Ku-Mahamud, “Big data: definition, characteristics, life cycle, applications, and challenges, IOP Conference Series: Materials Science and Engineering. Volume 769, No 1, p. 012007, 2020, doi:10.1088/1757-899X/769/1/012007

[2]. A. Gandomi, M. Haider, “Beyond the hype: Big data concepts, methods, and analytics”, International Journal of Information Management, Volume 35, Issue 2, pages 137-144, 2015, ISSN 0268-4012, https://doi.org/10.1016/j.ijinfomgt.2014.10.007.

[3]. B.Furht, F. Villanustre, “Big Data Technologies and Applications”, 2016, Springer International Publishing, ISBN 978-3-319-44548-9, DOI 10.1007/978-3-319-44550-2

[4]. D. Gupta, R. Rani, “A study of big data evolution and research challenges”, Journal of Information Science, Volume 45, Issue 3, pages 322–340, 2019, https://doi.org/10.1177/0165551518789880

[5]. Oxford Learner’s Dictionaries, big data. https://www.oxfordlearnersdictionaries.com/definition/english/big-data?q=big+data (accessed 3 May 2025)

[6]. E. F. Codd, “A Relational Model of Data for Large Shared Data Banks”, Communications of the ACM, Volume 13, Issue 6, pages 377-387, 1970, https://doi.org/10.1145/362384.362685

[7]. S. A. Jowan, R. F. Swese , A. Y. Aldabrzi, M. S. Shertil, “Traditional RDBMS to NoSQL database: new era of the databases for big data”, Journal of Humanities and Applied Science (JHAS), Issue 29, 2016

[8]. S.Islam, “Techniques for Converting Big Data from SQL to NoSQL Databases”, Available online: https://www.academia.edu/29793008 (accessed on 3 May 2025).

[9]. K.Rajaram, P. Sharma, S.Selvakumar, “DLoader: Migration of Data from SQL to NoSQL Databases”, In: Kumar, A., Ghinea, G.,

Merugu, S., Hashimoto, T. (eds) Proceedings of the International Conference on Cognitive and Intelligent Computing. Cognitive Science and Technology. Springer, Singapore. 2023. https://doi.org/10.1007/978-981-19-2358-6_19

[10].MongoDB, "What is MongoDB?", https://www.mongodb.com/docs/manual/ (accessed on 4 May 2025).

[11].Apache CouchDB, "Introduction", https://docs.couchdb.org/en/stable/intro/index.html (accessed on 4 May 2025).

[12].Terrastore, https://code.google.com/archive/p/terrastore/ (accessed on 4 May 2025).

[13].OrientDB, https://github.com/orientechnologies (accessed on 4 May 2025).

[14].Amazon DocumentDB (with MongoDB compatibility) https://aws.amazon.com/documentdb/ (accessed on 4 June 2025).

[15].MongoDB, What is a Document Database? https://www.mongodb.com/resources/basics/databases/document-databases (accessed on 4 June 2025).

[16].Amazon DynamoDB https://aws.amazon.com/dynamodb/ (accessed on 4 June 2025).

[17].Understanding Amazon NoSQL data stores: Types of NoSQL databases https://docs.aws.amazon.com/whitepapers/latest/choosing-an-aws-nosql-database/types-of-nosql-databases.html (accessed on 4 June 2025).

[18].Apache Cassandra, https://cassandra.apache.org/doc/latest/cassandra/architecture/overview.html, (accessed on 4 May 2025).

[19].Column-Oriented Databases, Explained https://www.kdnuggets.com/2021/02/understanding-nosql-database-types-column-oriented-databases.html (accessed on 4 June 2025).

[20].Neo4j, https://neo4j.com/ (accessed on 4 May 2025).

[21].ArangoDB, https://arangodb.com/ (accessed on 4 May 2025).

[22].Neo4j: Graph Databases for Beginners: Wait, What Do You Mean by "Graph"? https://neo4j.com/blog/graph-visualization/graph-databases-for-beginners-wait-what-do-you-mean-by-graph/ (accessed on 4 June 2025).

[23].E.J. Yannakoudakis, C.X.Tsionos, and C.A. Kapetis C.A, "A new framework for dynamically evolving database environments", Journal of Documentation, Volume 55, Issue 2, pages 144-158, 1999.

[24].E. Petraki, C. Kapetis, E. J. Yannakoudakis, "Conceptual Database Retrieval through Multilingual Thesauri", Computer Science and Information Technology Volume 1, Issue 1, pages 19 – 32, 2013, DOI: 10.13189/csit.2013.010103.

[25].E.N.Petraki, E.J.Yannakoudakis, C.A.Kapetis, "Conceptual Search Algorithms For FDB Databases", Springer Proceedings in Business and Economics, in: Androniki Kavoura & Damianos P. Sakas & Petros Tomaras (ed.), Strategic Innovative Marketing, pages 471-477, 2017. Springer Cham. https://doi.org/10.1007/978-3-319-56288-9_62 .

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed to the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or the Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.