ArangoDB is hailed as a native multi-model database by its developers. This is unlike other NoSQL databases. In this database, the data can be stored as documents, key/value pairs or graphs. And with a single declarative query language, any or all of your data can be accessed. Moreover, different models can be combined in a single query. And, owing to its multi-model style, one can make lean applications, which will be scalable horizontally with any or all of the three data models.

Layered vs. Native Multi-Model Databases

In this section, we will highlight a crucial difference between native and layered multimodel databases.

Many database vendors call their product “multi-model,” but adding a graph layer to a key/value or document store does not qualify as native multi-model.

With ArangoDB, the same core with the same query language, one can club together different data models and features in a single query, as we have already stated in previous section. In ArangoDB, there is no “switching” between data models, and there is no shifting of data from A to B to execute queries. It leads to performance advantages to ArangoDB in comparison to the “layered” approaches.

The Need for Multimodal Database

Interpreting the [Fowler’s] basic idea leads us to realize the benefits of using a variety of appropriate data models for different parts of the persistence layer, the layer being part of the larger software architecture.

According to this, one might, for example, use a relational database to persist structured, tabular data; a document store for unstructured, object-like data; a key/value store for a hash table; and a graph database for highly linked referential data.

However, traditional implementation of this approach will lead one to use multiple databases in the same project. It can lead to some operational friction (more complicated deployment, more frequent upgrades) as well as data consistency and duplication issues.

The next challenge after unifying the data for the three data models, is to devise and implement a common query language that can allow data administrators to express a variety of queries, such as document queries, key/value lookups, graphy queries, and arbitrary combinations of these.

By **graphy queries**, we mean queries involving graph-theoretic considerations. In particular, these may involve the particular connectivity features coming from the edges. For example, **ShortestPath, GraphTraversal**, and **Neighbors**.

Graphs are a perfect fit as data model for relations. In many real-world cases such as social network, recommendor system, etc., a very natural data model is a graph. It captures relations and can hold label information with each edge and with each vertex. Further, JSON documents are a natural fit to store this type of vertex and edge data.

ArangoDB ─ Features

There are various notable features of ArangoDB. We will highlight the prominent features below −

* Multi-model Paradigm
* ACID Properties
* HTTP API

ArangoDB supports all popular database models. Following are a few models supported by ArangoDB −

* Document model
* Key/Value model
* Graph model

A single query language is enough to retrieve data out of the database

The four properties **Atomicity, Consistency, Isolation**, and **Durability** (ACID) describe the guarantees of database transactions. ArangoDB supports ACID-compliant transactions.

ArangoDB allows clients, such as browsers, to interact with the database with HTTP API, the API being resource-oriented and extendable with JavaScript.