In this chapter, we will discuss the basic concepts and terminologies for ArangoDB. It is very important to have a knowhow of the underlying basic terminologies related to the technical topic we are dealing with.

The terminologies for ArangoDB are listed below −

* Document
* Collection
* Collection Identifier
* Collection Name
* Database
* Database Name
* Database Organization

From the perspective of data model, ArangoDB may be considered a document-oriented database, as the notion of a document is the mathematical idea of the latter. Document-oriented databases are one of the main categories of NoSQL databases.

The hierarchy goes like this: Documents are grouped into collections, and Collections exist inside databases

It should be obvious that Identifier and Name are two attributes for the collection and database.

Usually, two documents (vertices) stored in document collections are linked by a document (edge) stored in an edge collection. This is ArangoDB's graph data model. It follows the mathematical concept of a directed, labeled graph, except that edges don't just have labels, but are full-blown documents.

Having become familiar with the core terms for this database, we begin to understand ArangoDB's graph data model. In this model, there exist two types of collections: document collections and edge collections. Edge collections store documents and also include two special attributes: first is the **\_from** attribute, and the second is the **\_to** attribute. These attributes are used to create edges (relations) between documents essential for graph database. Document collections are also called vertex collections in the context of graphs (see any graph theory book).

Let us now see how important databases are. They are important because collections exist inside databases. In one instance of ArangoDB, there can be one or many databases. Different databases are usually used for multi-tenant setups, as the different sets of data inside them (collections, documents, etc.) are isolated from one another. The default database **\_system** is special, because it cannot be removed. Users are managed in this database, and their credentials are valid for all the databases of a server instance.