Document databases are databases that use a hierarchical order of database > collection > document to form a logical structure and order to themselves. The documents themselves can be stored in XML, JSON or BSON depending on the application used for the database. The idea behind the documents is to be able to store information in key value pairs and to be able to have additional relationships between documents. There can also be many collections within a database that hold different documents of various data that may not have direct relationships with other documents under another collection within the same database. Document databases are great for Event Logging, Content Management, Web Analytics and E-Commerce however, they are not ideal for complex transactions spanning many different operations or queries against varying aggregate structures (Fowler, Sadalage pg. 98).

Collections are groupings of documents within a document database. It can be compared to a table in a RDBMS. Collections do not have to store the same data in documents due to them not requiring a schema. This creates a sort of flexibility in design in comparison to a RDBMS which requires schemas to define structure and organization of data within a database.

The main difference between a RDBMS and a NRDBMS is a relational database requires a defined structure through schemas. Schemas are a sort of skeleton that determines the structure of a database and in the case of relational database, one has to keep that structure and relationships. This is in contrast to a non-relational database that does not require schemas and uses a document structure to hold data. Due to the lack of schema, a non-relational database doesn’t require the same sort of structure as its counterpart which lends itself to higher adaptability and flexibility in nature.