Document databases are nonrelational databases where data is stored and queried as documents which resemble JSON documents. These are great for developers because they often work with JSON documents, and this format requires minimal changes on their part to work with their applications. They work particularly well with catalogs, user profiles, and content management systems because they allow for flexible indexing, powerful ad hoc queries, and analytics as they change and evolve over time. (MENEGASSO, 2018) MongoDB lists the intuitive data model that is easy for developers to work with, the flexible schema that allows the database to change and evolve, and the ability to scale out horizontally among the top features of these databases. When you couple that with the widely used JSON document structure, these databases become easier to interpret, query, and work with. (MongoDB, 2022)

Collections then, are groupings of documents within a database. Although these collections may all have different fields, when viewed as a whole, the resemble the tables you would see in a relational database. Collections allow you to manage documents as a group through actions such as importing or exporting documents, analyzing your entire schema, or setting up validation rules. (MongoDB Compass, 2022) The primary difference to keep in mind between tables in relational databases and documents in collections, is that with collections, you don’t have to define what columns or attributes will be involved in advance. (ArangoDB Documentation, 2022)

What are the real differences then between a relational and a nonrelational database? Well, a relational database is a rigid way of sorting data into fields which make up rows and columns, very much you would find in a spreadsheet. Nonrelational databases are much more freeform, usually because the information in them continues to grow, but not in a set manner that fits into neat rows and columns. One comparison is that it is more like a Word document, where all the information is there, but you must use something like the **Find** command to search for the particular information you are seeking. (Pawlan, 2022)

References:

*Collections*. Collections - MongoDB Compass. (n.d.). Retrieved June 10, 2022, from https://www.mongodb.com/docs/compass/current/collections/

*Databases, collections and documents: Getting started: Manual: Arangodb documentation*. Getting Started | Manual | ArangoDB Documentation. (n.d.). Retrieved June 10, 2022, from https://www.arangodb.com/docs/stable/getting-started-databases-collections-documents.html

*Document database - nosql*. MongoDB. (n.d.). Retrieved June 10, 2022, from https://www.mongodb.com/document-databases

MENEGASSO, Andre Eduardo (2018). *NOSQL*. Amazon. Retrieved June 10, 2022, from https://aws.amazon.com/nosql/document/

Pawlan, D. (2022, January 18). *Relational vs. Non-Relational Database: Pros & Cons: The aloa blog*. RSS. Retrieved June 10, 2022, from https://aloa.co/blog/relational-vs-non-relational-database-pros-cons