Core MongoDB Concepts

This lesson briefly touches upon the structure of MongoDB and goes over the MongoDB documents.

WE'LL COVER THE FOLLOWING

In the previous chapter – Introduction to NoSQL and Polyglot Persistence, we talked about what NoSQL databases are, how they can be used, and what are the benefits of using one in your system. Different types of NoSQL databases were also mentioned, as well as their most popular representatives.

One of these types is *Document NoSQL databases*.

Document databases are probably the closest thing to relational databases of all the databases in the NoSQL ecosystem.

Documents databases store documents in collections, a lot like relational databases store rows in tables; but, we'll get more into that later.

One of the most popular representatives of Document databases and one of the leading NoSQL databases, is MongoDB.

A Basic Introduction

Before we see all the different ways that MongoDB can be used, let's go through some of the basic concepts of this database.

Guys from MongoDB are very proud of what they call *Nexus Architecture*. This architecture gives the ability to combine proven concepts and abilities of relational databases with NoSQL innovations.

MongoDB Documents

MongoDB is a *document-oriented* database and, as previously mentioned, it has certain similarities to relational databases.

Instead of rows, MongoDB has *documents*. Unlike relational databases, where information for a given record is spread across many tables, all of the data for a given record is stored within a *single* document.

Under the hood, documents are *BSON* files, which are the binary-encoded serialization of JSON files. Nevertheless, from a programmer's point of view, MongoDB manipulates pure JSON files.

For example, we can represent `user` like this:

```
{
    "_id" : ObjectId("58e28d41b1ad7d0c5cd27549"),
    "name" : "Nikola Zivkovic",
    "blog" : "rubikscode.net",
    "numberOfArticles" : 10,
    "Adress" : [
        "street" : "some street",
        "city" : "Novi Sad",
        "country" : "Serbia"
        ],
    "company" : "Vega IT Sourcing",
    "expertise" : [".NET", "JavaScript", "NoSQL", "Node.js"]
}
```

As you can see:

- there is an `_id` field at the beginning of this JSON document.

This field is unique and is generated by MongoDB for every document in the database.

This way MongoDB keeps one of the important properties of relational databases – *strong consistency*.

In this lesson, we barely touched upon the structure of MongoDB. In the next lesson, we will discuss it more in-depth when we discuss *Collections* in MongoDB.