1. What are document databases?

A document database uses existing, data-centric file types to store data. JSON, XML/TSV, and CSV are a few of my favorites! Yes, I have favorite file types and I’m proud of it. These file types were designed to give visual and logical structure to stored information. In this setup, should need for either arise, the data can be readily viewed and manipulated through either direct (at least screen/eyes/keyboard) human interaction, or applications. This will easily be my comfort zone. I do wonder though how something like “Pandas” DataFrames can play into this. Maybe that is more table based? I guess we’ll see.

1. What are collections?

Collections are groupings of documents, whose structure and dataset is similar to one. This provides similar functionality to schemas in Relational Databases. I may get unreasonably excited as I speculate on how this structure can allow an application to evolve over time, without loosing reverse compatibility. But the developers should definitely take care to not create fields with the same names which do not actually hold the same kind of information. On the flip side, it would be best to avoid having multiple names for fields which store the same data (i.e. heart rate & bpm). “+1” to the value of good developers!

1. What are the key differences between relational and non-relational database structures?

So far, I like to think of relational databases as rigid, thoroughly designed, and less prone to “id-10T” errors (at least on the user end). “A place for everything and everything in it’s place” comes time mind, and I may even add, “a way for everything to be in its place” in there. A non-relational database is like my desk. Everyone that uses my desk (just me, but still…) knows where everything is. However if we (I) decide to move something, that would be just fine. There is however an “implicit place for everything”. If someone (like my son) moves something on the desk without advising all other users, the system starts to break down. Each type of database has its own strengths and weaknesses.