Time to Complete Task

10 hours

Why MongoDB is an Appropriate Choice for AggieFit

MongoDB provides several features that make it a better fit for AggieFit than a traditional relational database.

MongoDB is a NoSQL database that uses a document-based data model. Document-based databases have a high degree of flexibility in what data types and fields can be stored in the the database. In AggieFit's case, this is especially useful because employees may opt to include or not include different fields, such as their age, weight, or other information. Document-based databases also map more easily to objects in object-oriented programming languages, making it easier to develop applications that use the stored data in the future. MongoDB in particular uses a data format called BSON that is very similar to JSON.

MongoDB is also has a robust query language that supports CRUD operations. It also queries over all fields stored in documents and offers primary and secondary indexing over all fields to improve performance. These features enable MongoDB to perform with comparable speed and power to a traditional relational database.

MongoDB also supports APIs for major languages including Java, C++, Python, JavaScript, Ruby, and more. This support allows the development of large range of applications across different platforms. This would provide AggieFit with flexibility in developing future uses for the data stored in the database.

Finally, MongoDB has strong commercial support, making it a safe and reliable choice for a NoSQL document-based database. It's highly available and has an active community developing new tools for it.

Possible Improvements for AggieFit DB

The main disadvantage of MongoDB for AggieFit is that there is no way to set a required field, such as employee ID. This means that any future applications developed for AggieFit will have to make sure to provide checking and validation on the application side.

How to run code

The queries were added to the queries.py file. I added a few constants to the constants.py file, so the updated constants file will need to be in the same directory as queries.py and mongo_connect.py. To run the queries, use the command "python queries.py".