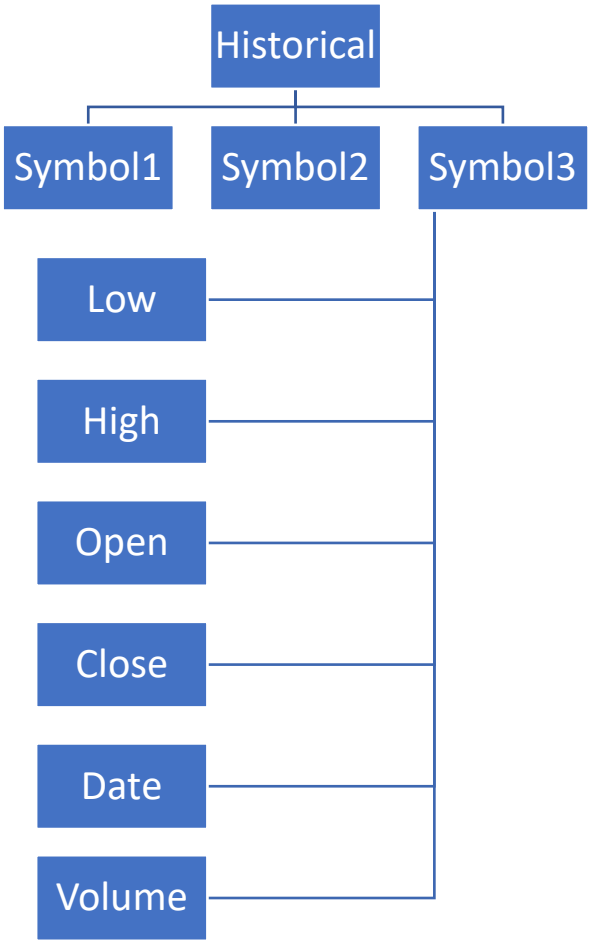
Schema for the database:

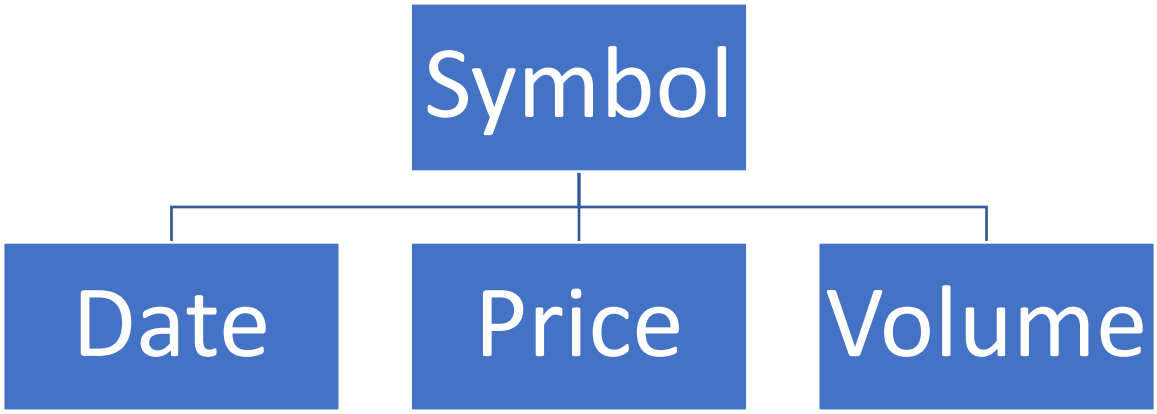
We used a cloud mongoDB service to facilitate data storage. Having the database in the cloud ensures that we don't need to set it up on multiple computers as we work and that data replication is not an issue.

Since we are using mongoDB, our database does not have strong Entity Relations like a SQL database does since mongoDB is a NoSQL database.

The reason we chose a NoSQL database is because of the set of tools we are using to develop. Our database is in the cloud so we fetch data using REST calls which are returned as JSON strings. NodeJS is quite robust in processing JSON strings so it made sense to use all of these together. This would mean that we didn't have to change the structure of the data at any stage(with MYSQL we would have to rearrange it every time we insert or fetch) therefore we end up saving on processing power!

Below we provide a pictorial representation of the mongoDB collections. We have 1 document called "historical" that contains all data for all companies. Then we have 10 documents for storing real time data for each of the 10 companies.





Storing data in this kind of a hierarchy is convenient for us and allows us to fetch and insert data rapidly therefore this design is suitable for our needs and we decided to use it.