MongoDB Data Challenges

The following challenges aim to give you opportunities to wrangle data with MongoDB. They can be done in any order, and you can expand upon any of the ideas to suit your interest and follow your train of thought. The optional challenges are just that – if you find you have time or fancy more examples, feel free to give them a go, but in the time available, we are not expecting you to do them.

All these challenges are achievable using MongoDB Shell but you can use Compass too. There is no expectation of any coding other than MongoDB, but if you like, feel free to apply some Python or JavaScript.

These are all exercises for you to do yourself but of course you can chat to others about them. We recommend that you skim through them all and then decide which you'd like to try first.

There is no formal assessment of these challenges, but we would love some of you to show your work. We like to see unfinished/experimental stuff as well as finished solutions, so if you've got some way down the road, show us what you have!

Challenge A: Basic Queries, Updates and Data Types

A bunch of users have been chatting online. This resulted in the text files 'users.json', 'posts.json' and 'comments.json'. Import these as separate collections into a database called 'wiki'.

- How many users, posts and comments do we have?
- Is there a particularly prolific poster?
- What are the emails of the 5 people who commented on post number 42?
- Which post did Elian@matilda.me and Elda@orval.name both comment on?
- Find the user located at longitude '21.8984'.
- Find which user has the username 'Delphine'.

The same users have generated a lot of photos which they have sorted into albums, as recorded in 'photos.json' and 'albums.json'. Import these as separate collections into the wiki database.

- For each user, find how many albums they have.
- It turns out that user 3 has been plagiarizing photos taken by other users. In particular, the first five albums attributed to user 3 actually belong to user 1, and the last five belong to user 9. Make changes to your database to reflect this. Make sure user 3 ends up with zero albums attributed to them.
- Remove all posts made by user 3 then remove user 3 from the database.
- Insert a new user (e.g. yourself or a made up character) with some suitable data.

Optional: We need to be able to find users by their geographical location (lat and lng). Currently these are stored as strings. Take a look at https://docs.mongodb.com/manual/tutorial/model-monetary-data/

- Find a way to create new fields representing the latitude and longitude as decimal values.
- How many users live west of -40?