**Task 1: Design the Data Model**

Design a data model for the blogging platform.

Please refer to the [best practices for schema design in MongoDB](https://www.mongodb.com/developer/products/mongodb/mongodb-schema-design-best-practices/).

You should model the following entities:

* users
* blogs
* posts
* comments

**My Schema Design**

* A user is unique (one-to-one)
* Assume a single user can have multiple blogs (one-to-few or one-to-many)
* A single blog can have multiple posts (one-to-squillions)
* A single post can have multiple comments (one-to-squillions)

Due to the unlimited amount of possible posts and comments we use referencing instead of embedding to avoid exceeding the MongoDB document limit of 16MB.

We include referencing in the individual documents of e.g. user id, blog id, post id, and comment id in order to be able to join related comments, posts, blogs and users.

**The User Document (one per user)**

The user is unique and has unique contact information as e.g. email. Here we use key-value pairs on the MongoDB document:

* “\_id”: ObjectId(“id\_someone”), automatically assigned if not specified
* “username”: “Someone”,
* “email”: “[someone@gmail.com](mailto:someone@gmail.com)”,
* “blogs”: [ObjectId(“blog1”), ObjectId(“blog2”)] Reference to the users unique blog ids

**The Blog Document (one per blog per user)**

Each blog is unique. Again, we use key-value pairs and references to the blog owner (the user) and all posts on the blog:

* “\_id”: ObjectId(“blog1”),
* “user\_id”: ObjectId(“id\_someone”), Reference to the unique user id
* “title”: “Blog about Someones life”,
* “description”: “This blog is about …”,
* “Posts”: [ObjectId(“post1”), ObjectId(“post2”)] Reference to the unique posts on the blog

**The Post Document (one per post)**

Each post on a blog is unique. Again, we use key-value pairs and references to the blog owner (the user), the blog id, and also reference to all comments added to this post:

* “\_id”: ObjectId(“post1”),
* “user\_id”: ObjectId(“id\_someone”),
* “blog\_id”: ObjectId(“blog1”),
* “title”: “Typical Monday…”,
* “context”: “Bla bla bla”,
* “comments”: [ObjectId(“comment1”), ObjectId(“comment2”),

ObjectId(“comment3”)] Reference to the unique comment ids of this post

**The Comment Document (One per comment)**

Each comment is unique. Again we use key-value pairs and references to the blog owner (the user), the blog id and post id:

* “\_id”: ObjectId(“comment1”),
* “user\_id”: ObjectId(“id\_someone”),
* “blog\_id”: ObjectId(“blog1”),
* “post\_id”: ObjectId(“post1”),
* “content”: “someone commented on Someones blog post”,
* “created\_at”: ISODate(“…”)

Note, in e.g. the comment document we do not need to include all the referencing ids. We could as example leave out the blog id, as we can get this from the post document via the post\_id reference. However, including the reference to everything (and it is still only 3 lines) we may make lookups and other information seeking faster and easier as we (likely) need less joining of the different documents.