

COMP90050: Advanced Database Systems

Semester 2, 2024

NoSQL Lab

NoSQL (also called Not Only SQL) is an approach to database design that enables storage and querying of data outside of the traditional approach of relational databases. It provides a more flexible mechanism as compared to relational databases to store information with limited constraints on the underlying data schema.

NoSQL databases come in a variety of types as discussed in class. As part of this lab, we use MongoDB, which is a document-oriented NoSQL database.

MongoDB Installation Instructions:

The general installation instructions are located here:

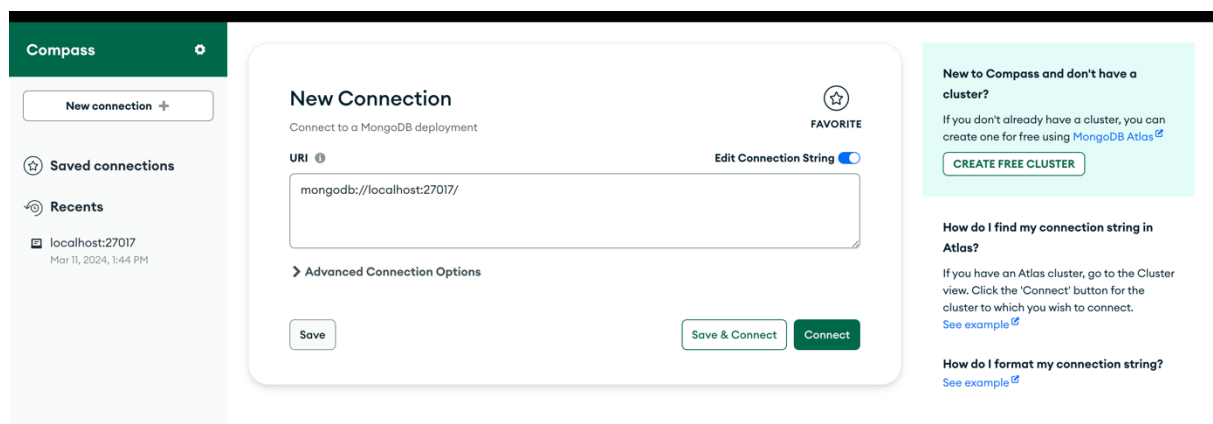
<https://www.mongodb.com/docs/manual/installation/>

Depending on your OS, you can follow the relevant instructions which keep getting updated as newer versions release. Alternatively, if you are familiar with containerisation, you could simply run MongoDB as a Docker container.

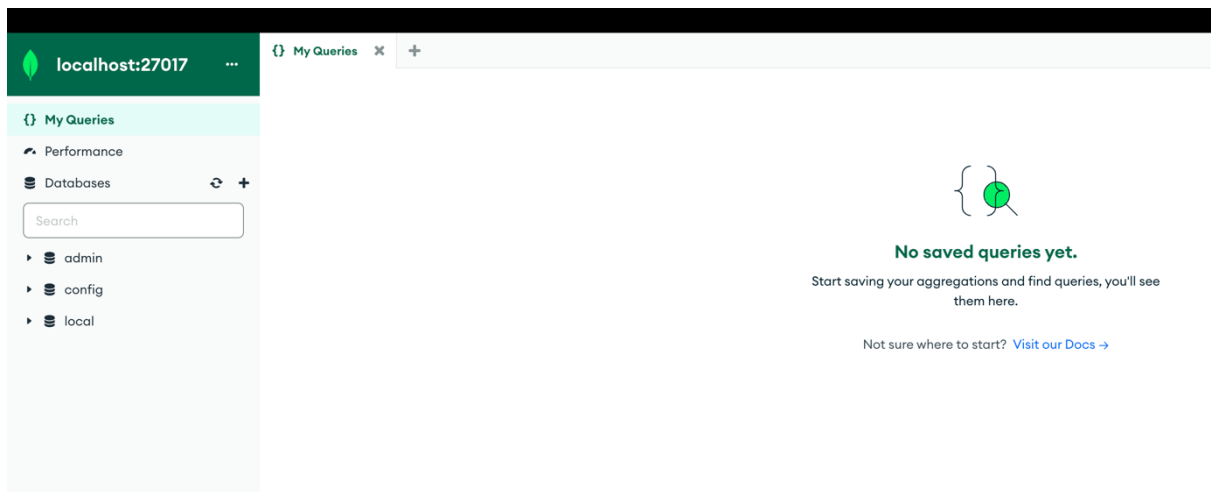
After installing the MongoDB database engine, it would be best to install a User Interface (UI) to interface with the database engine. There are numerous UI tools out there for MongoDB. We recommend installing MongoDB Compass:

<https://www.mongodb.com/products/tools/compass>

Once you install the MongoDB database server and Compass, open Compass. The view should look similar to as seen below:



The URI section should point to the database server location and port number to connect to (similar to what we did in MySQL Workbench). When you click connect, you should be able to see some databases appear on the left side of the UI as seen below:



MySQL to MongoDB Terminology Mapping:

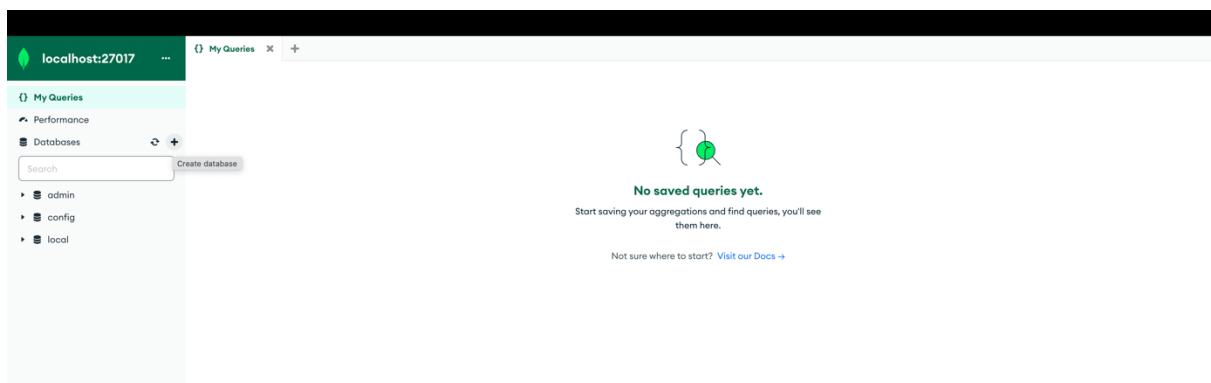
The underlying database structure and classifications in MySQL are similar to MongoDB with a few differences in terminology and the underlying implementation. These differences are documented here: <https://www.mongodb.com/docs/manual/reference/sql-comparison/>

The main terminology to be familiar of are as follows:

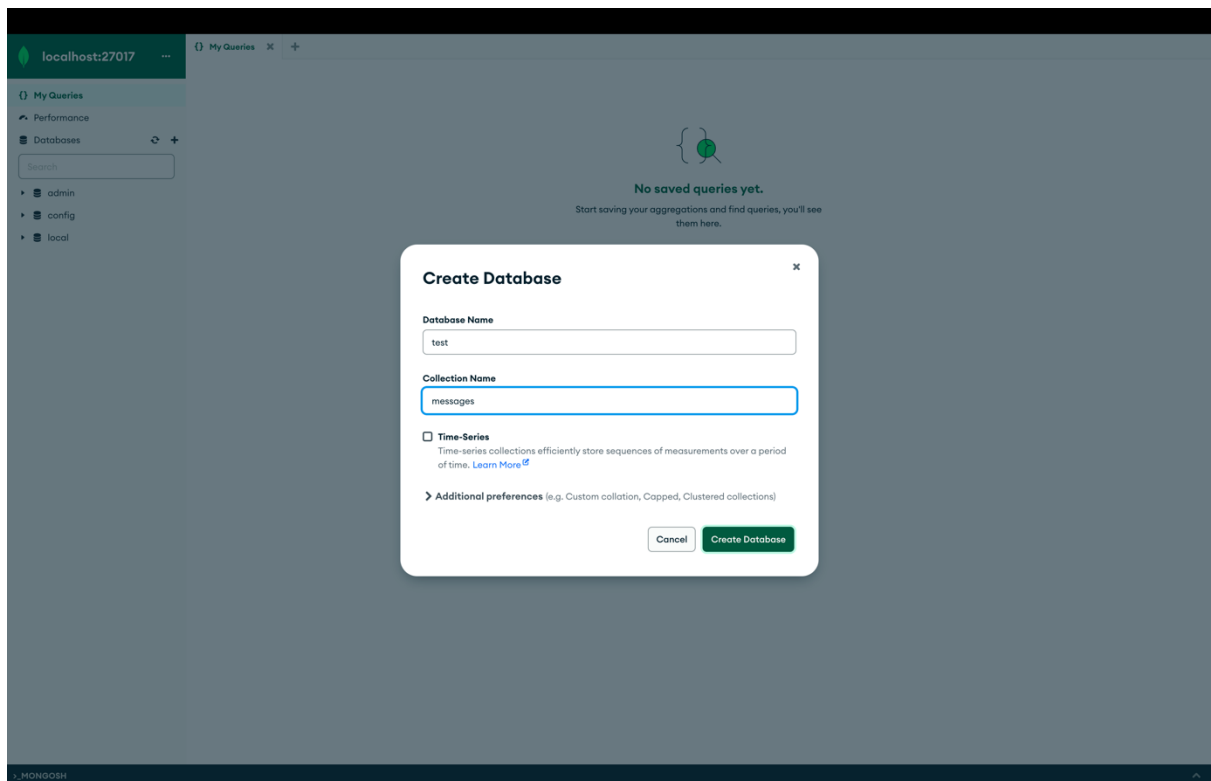
| MySQL | MongoDB |
|----------|------------|
| Database | Database |
| Table | Collection |
| Row | Document |

Create a Database:

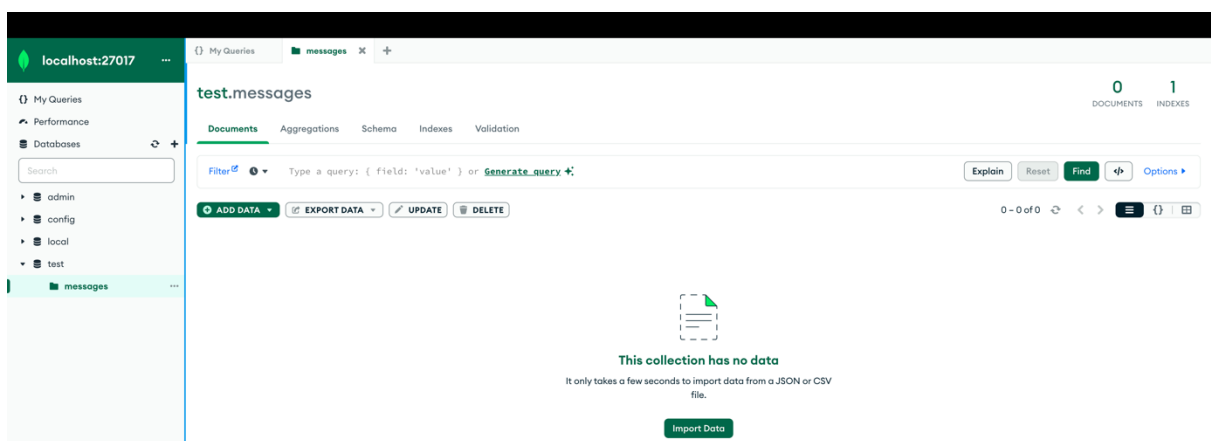
After connecting to the Database server, we will start by creating a Database. For this, click on the “+” sign on the left next to the Databases sign as seen in the screenshot below:



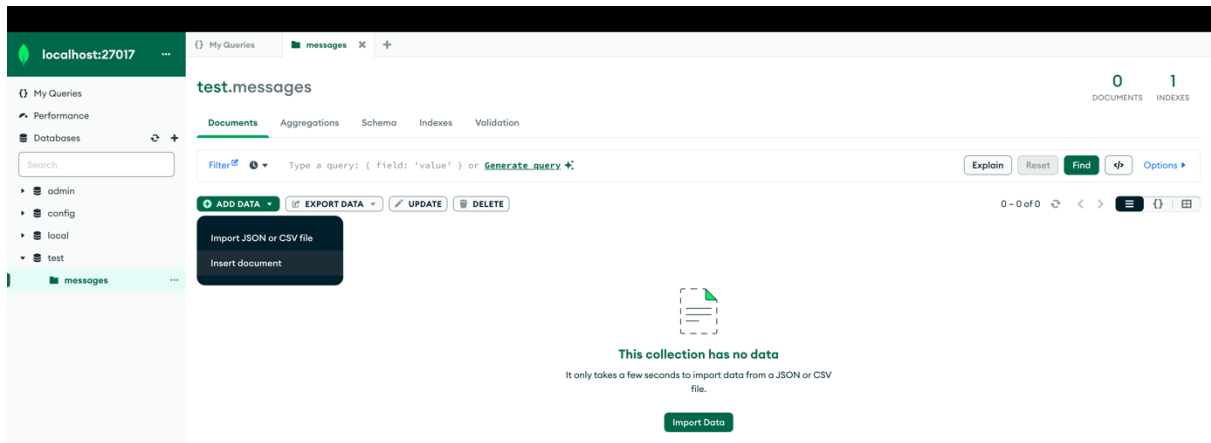
In the MySQL Locking Lab, we were running our operations on the “messages” table in the “test” schema/database. To create a similar structure, we can enter the database name as “test” and the collection name as “messages” as seen in the screenshot below:



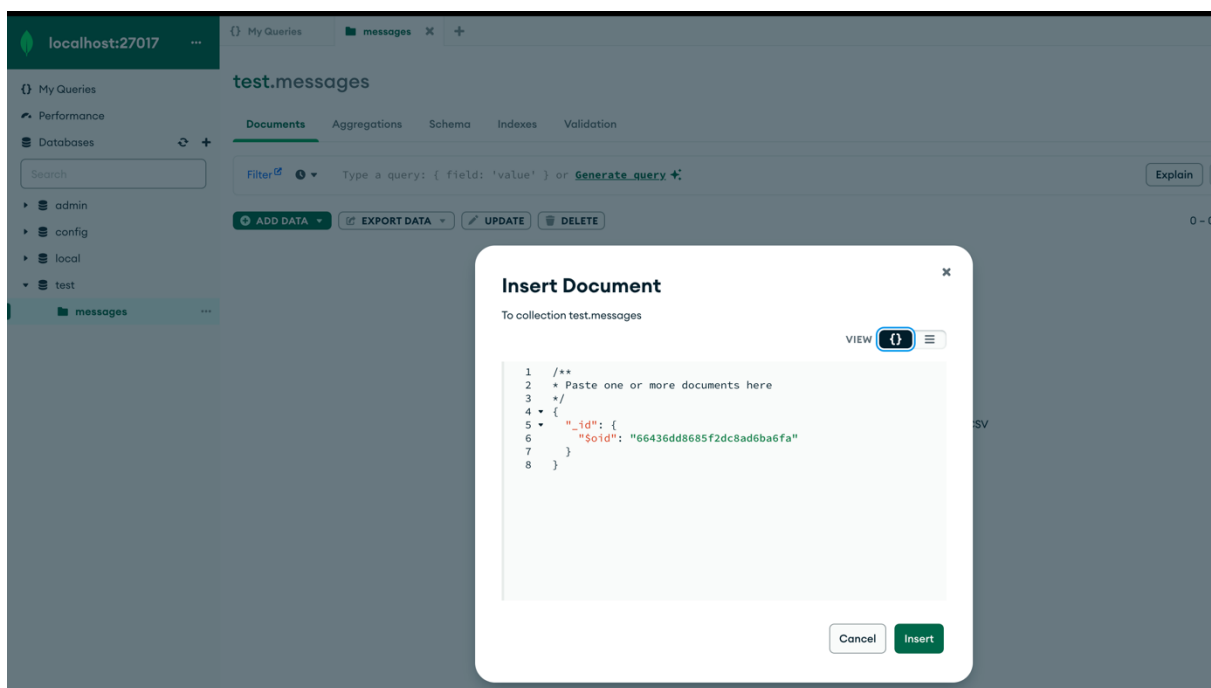
Click on the “Create Database” button and you will observe that a database called “test” has been created on the left side pane of the UI. This contains a Collection called “messages”.



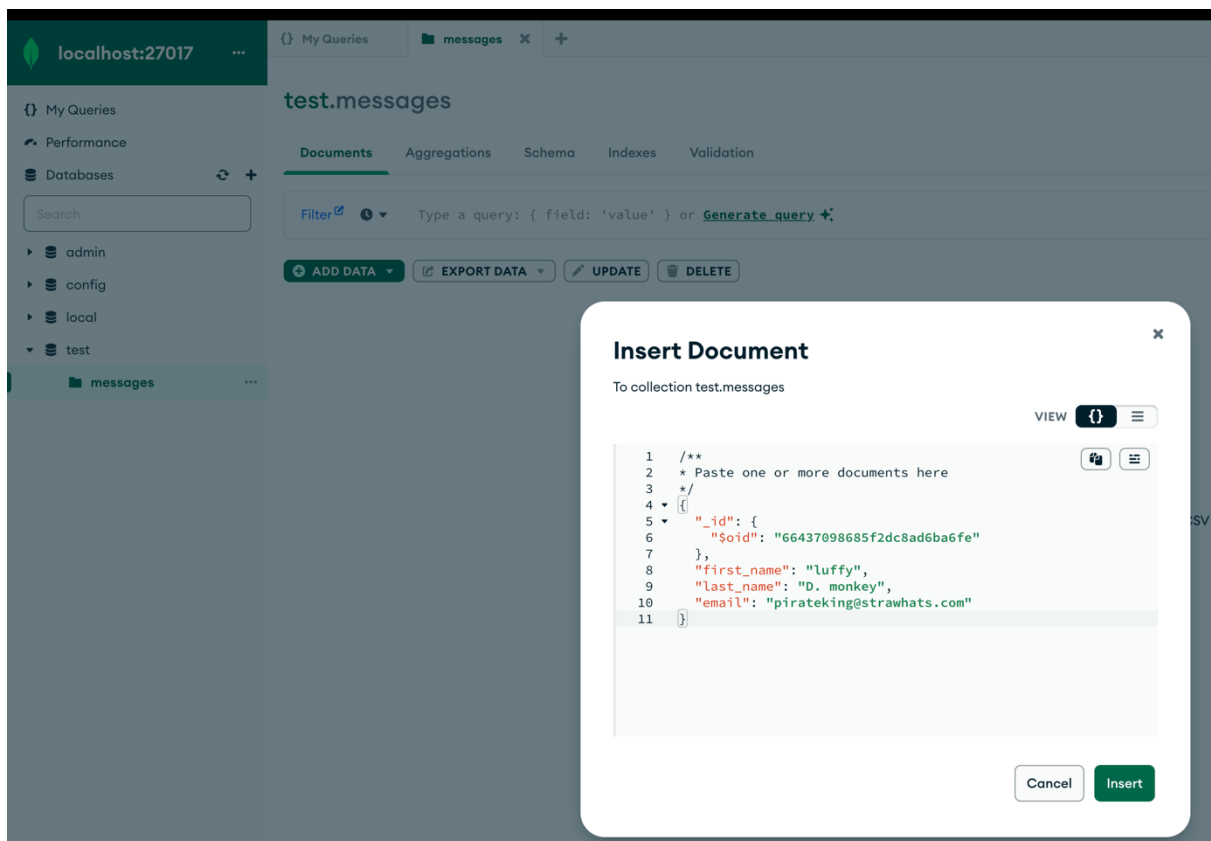
To insert data, we can click on the “Add Data” button and choose “Insert Document” as seen in the screenshot shown below:



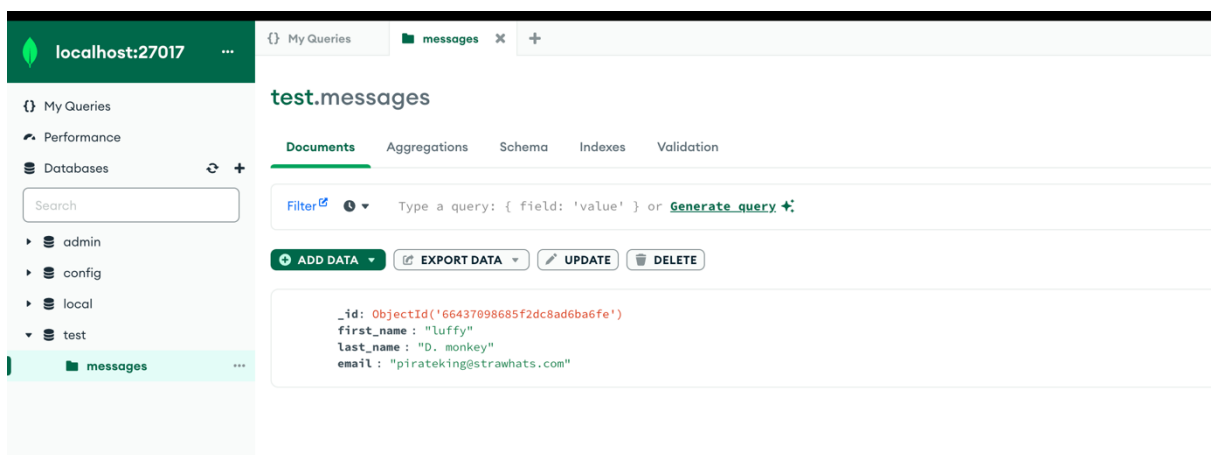
You will observe a pane with a blank template as seen below:



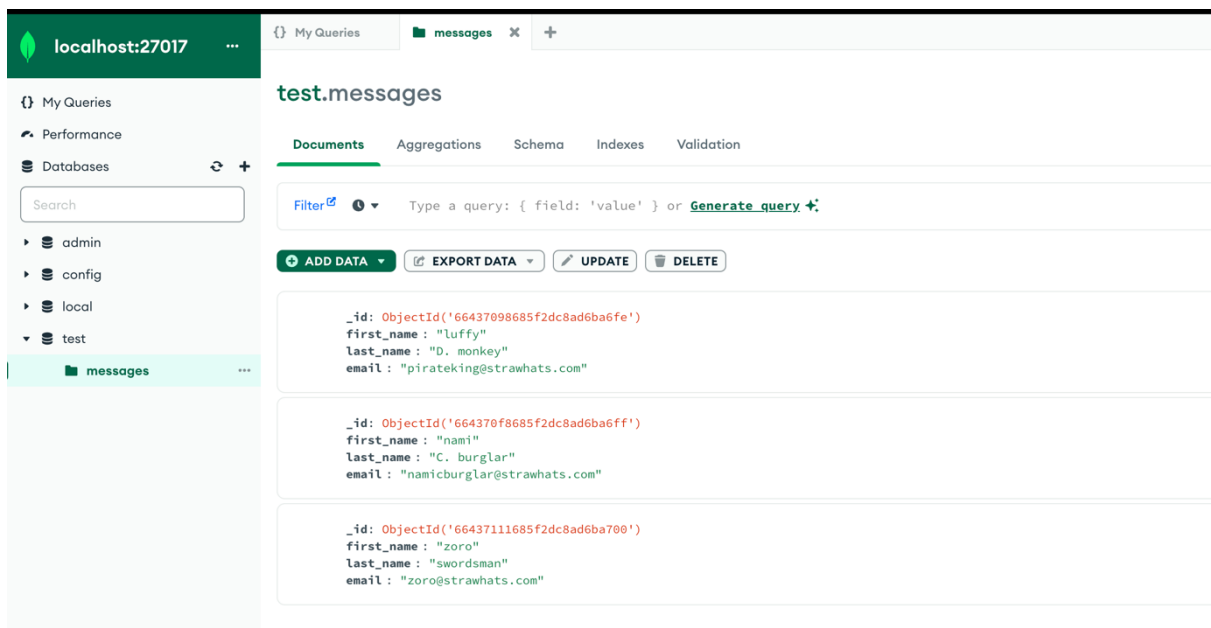
To insert data, we add information as a set of Key-value pairs as follows:



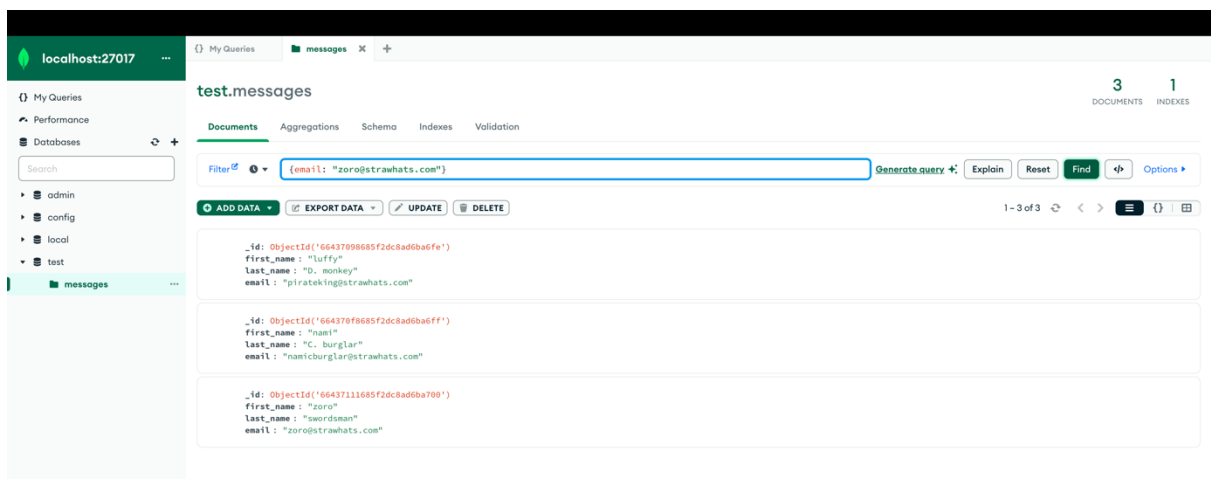
After writing the entries needed, we click on the “Insert” button. We can then observe that a data object has been inserted into the “messages” collection in the “test” database.



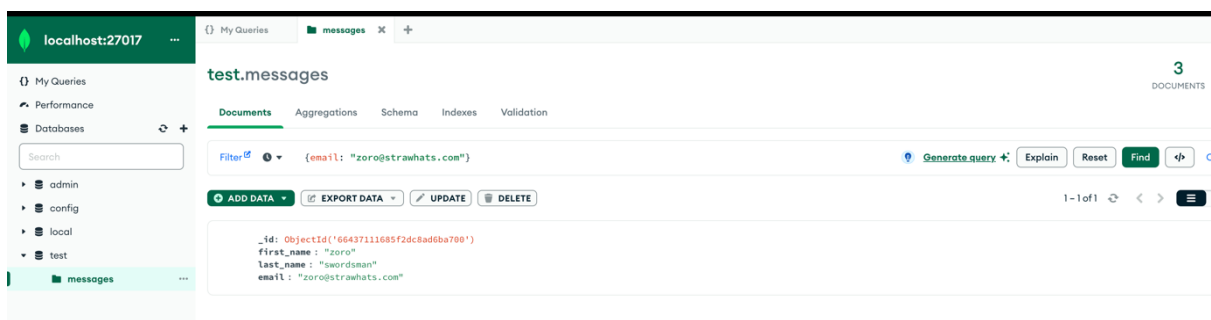
Similarly, insert a few more rows as seen below:



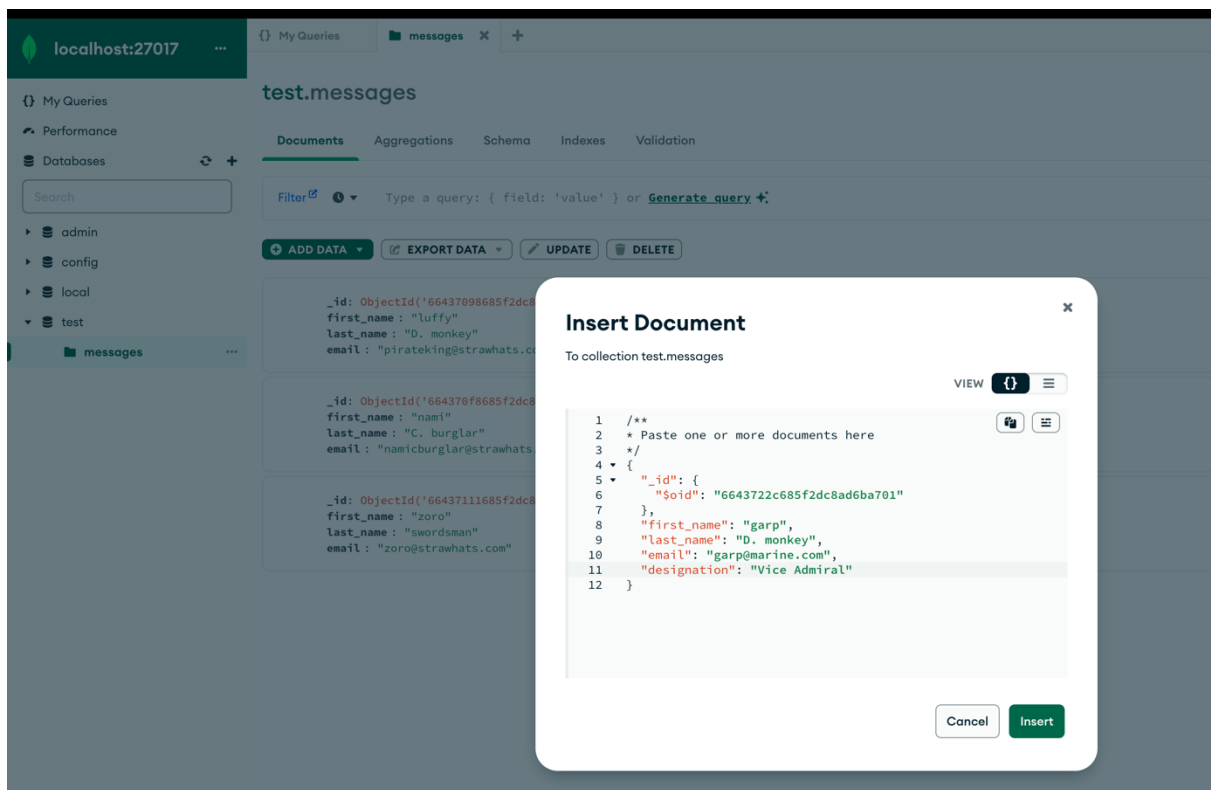
To run queries, we can type the query in the search bar on top and click on the “Find” button as seen below:



This will filter out the data returning the relevant document matching the query as seen below:



One of the unique features of NoSQL in comparison to SQL is that the schema is flexible. This means that if we try to insert a Document with a slightly different schema as seen below, it will be accepted and stored:



If you click “Insert” you will observe that this entry has been accepted and is visible in the “messages” collection.

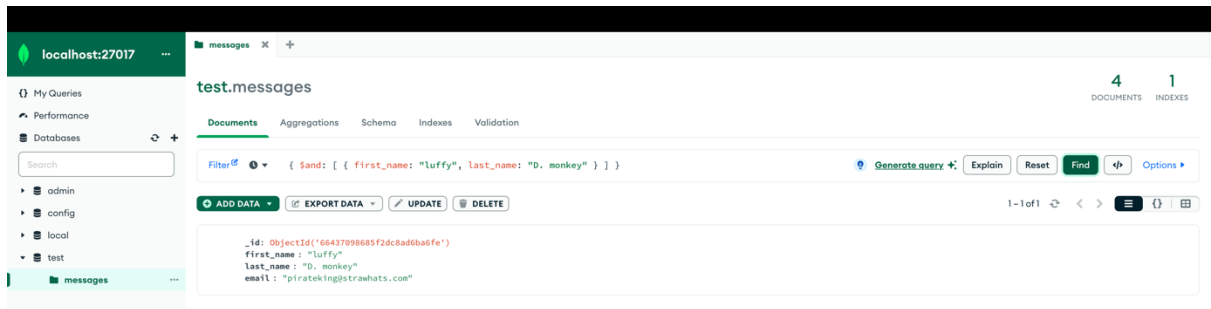
The screenshot shows the MongoDB Compass interface at localhost:27017. The left sidebar lists the database structure: My Queries, Performance, Databases, and a search bar. Under 'Databases', there are folders for 'admin', 'config', 'local', and 'test'. The 'test' database is expanded, showing a 'messages' collection. The main panel displays the 'test.messages' collection with tabs for Documents, Aggregations, Schema, Indexes, and Validation. The 'Documents' tab is active, showing a list of four documents. Each document contains fields: _id (ObjectId), first_name, last_name, and email. The documents are as follows:

- Document 1: `{ "_id": "ObjectId('66437098685f2dc8ad6ba6fe')", "first_name": "luffy", "last_name": "D. monkey", "email": "pirateking@strawhats.com" }`
- Document 2: `{ "_id": "ObjectId('664370f8685f2dc8ad6ba6ff')", "first_name": "nami", "last_name": "C. burglar", "email": "namicburglar@strawhats.com" }`
- Document 3: `{ "_id": "ObjectId('6643711685f2dc8ad6ba700')", "first_name": "zoro", "last_name": "swordsman", "email": "zoro@strawhats.com" }`
- Document 4: `{ "_id": "ObjectId('6643722c685f2dc8ad6ba701')", "first_name": "garp", "last_name": "D. monkey", "email": "garp@marine.com", "designation": "Vice Admiral" }`

If we wish to filter based on multiple parameters then we can do this as seen below:

The screenshot shows the MongoDB Compass interface with a filter applied to the 'test.messages' collection. The filter bar contains the JSON query: `{ $and: [{ first_name: "luffy", last_name: "D. monkey" }] }`. The 'Find' button is highlighted, and the documents are filtered to show only the first document. The interface also shows the number of documents (4) and indexes (1) for the collection.

The `$and` function accepts your conditionals in a JSON format, specifying the key and expected value. When you click on the “Find” button on the right, you will observe the filtered document as seen below:



There is an excellent Blog post/tutorial on mapping terms from SQL to NoSQL from MongoDB with step-by-step examples and a sample use-case. Please review this and follow along to understand the nitty-gritties of the two approaches:

<https://www.mongodb.com/developer/products/mongodb/map-terms-concepts-sql-mongodb/>