Practice 4: MongoDB Basics

Big Data System Design

Part 1

PRACTICE

Database

- Database is a physical container for collections
  - Commands
    - use DATABASE_NAME
    - show dbs
    - db.dropDatabase()

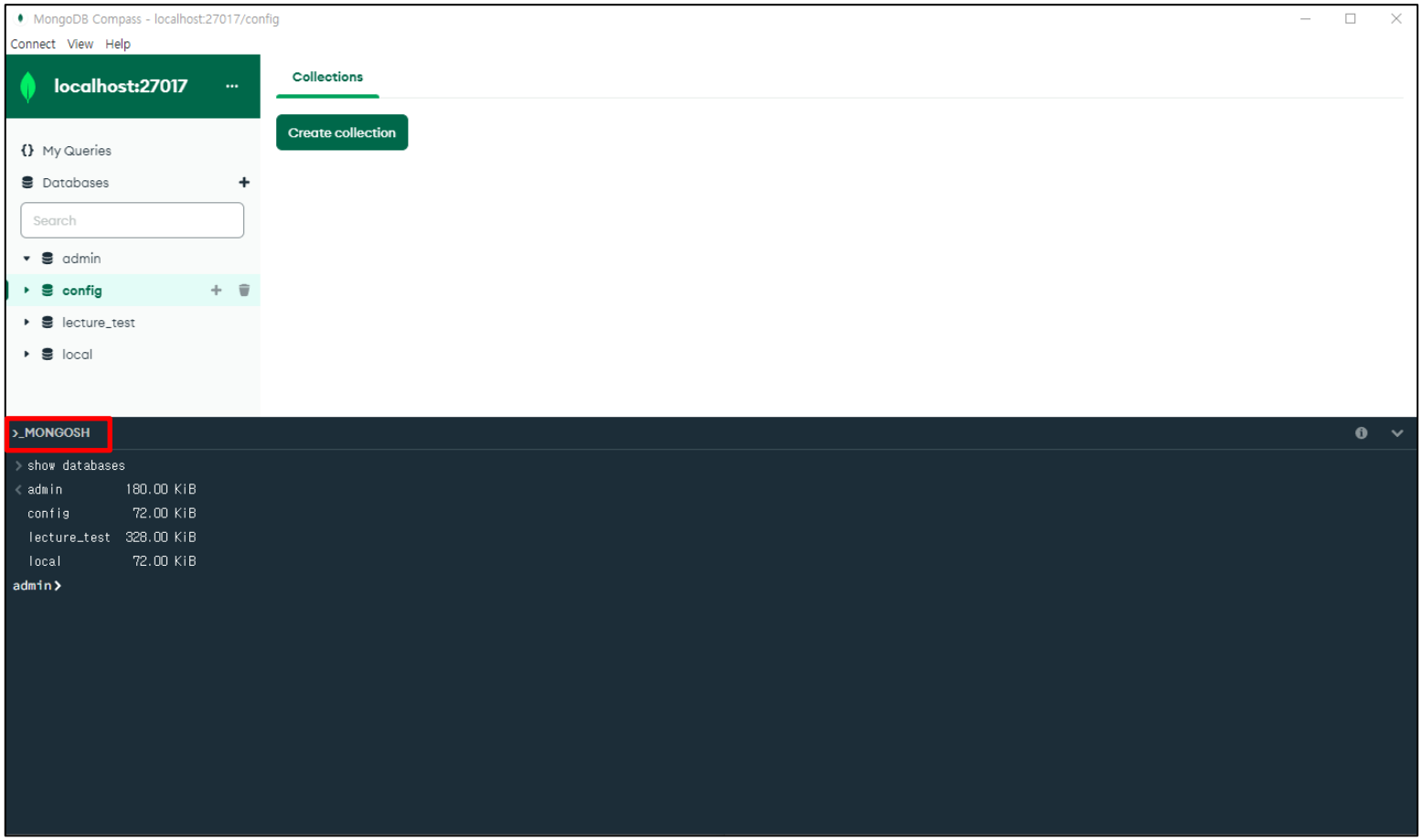
- Practice 1
  - **show dbs**

- Practice 2
  - **use blogging**

- Practice 3
  - **db**

Database

- MongoDB Compass
  - You can use Compass Shell (Mongosh) in the lower left corner

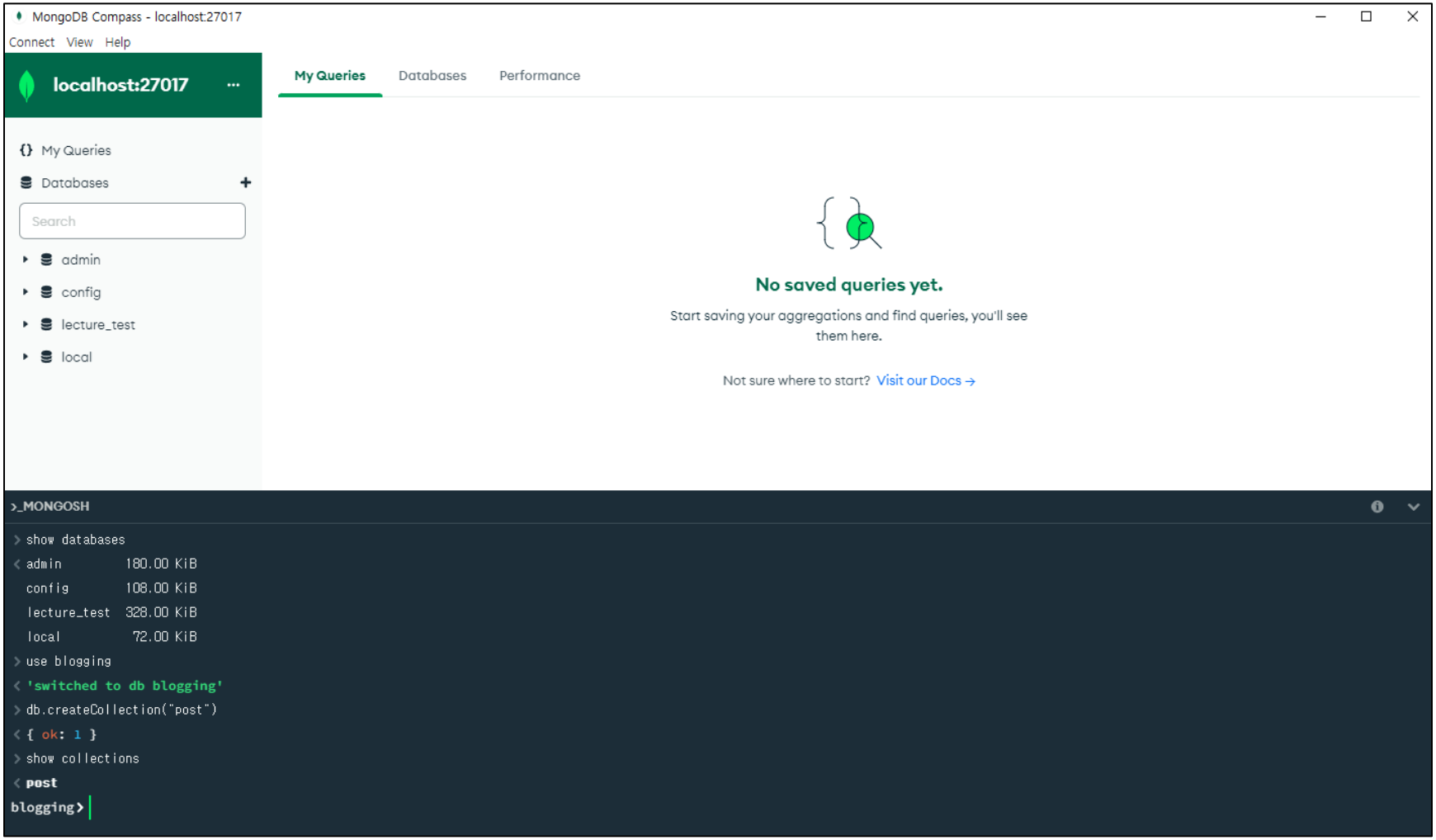


Collection

- Collection is a group of MongoDB documents similar to tables of RDBMS
  - Commands
    - db.createCollection(name, options)
    - show collections
    - db.COLLECTION_NAME.drop()
  - Practice 4
    - **show collections**
  - Practice 5
    - **db.createCollection("post")**

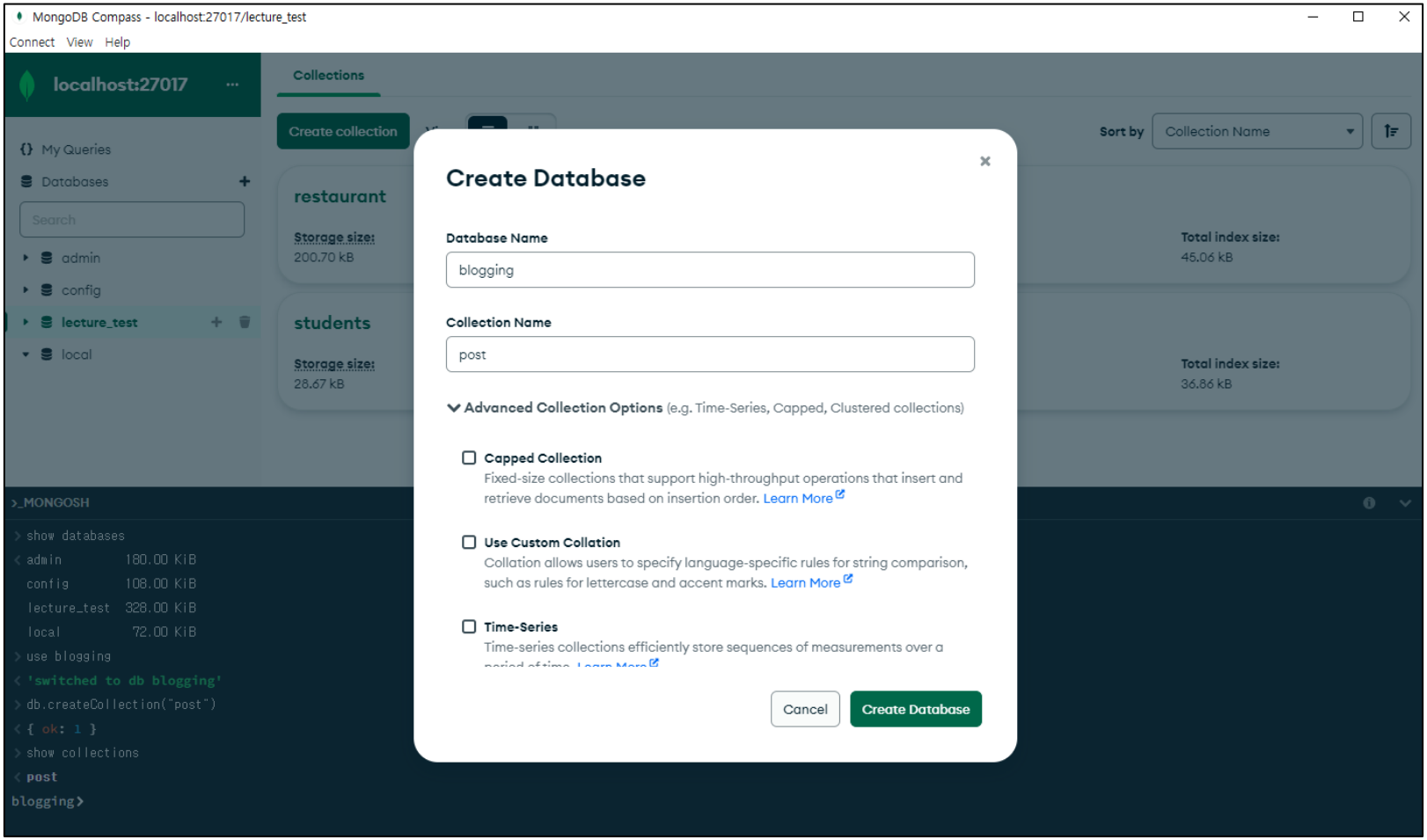
Collection

❖ Result of Practice 4 and Practice 5



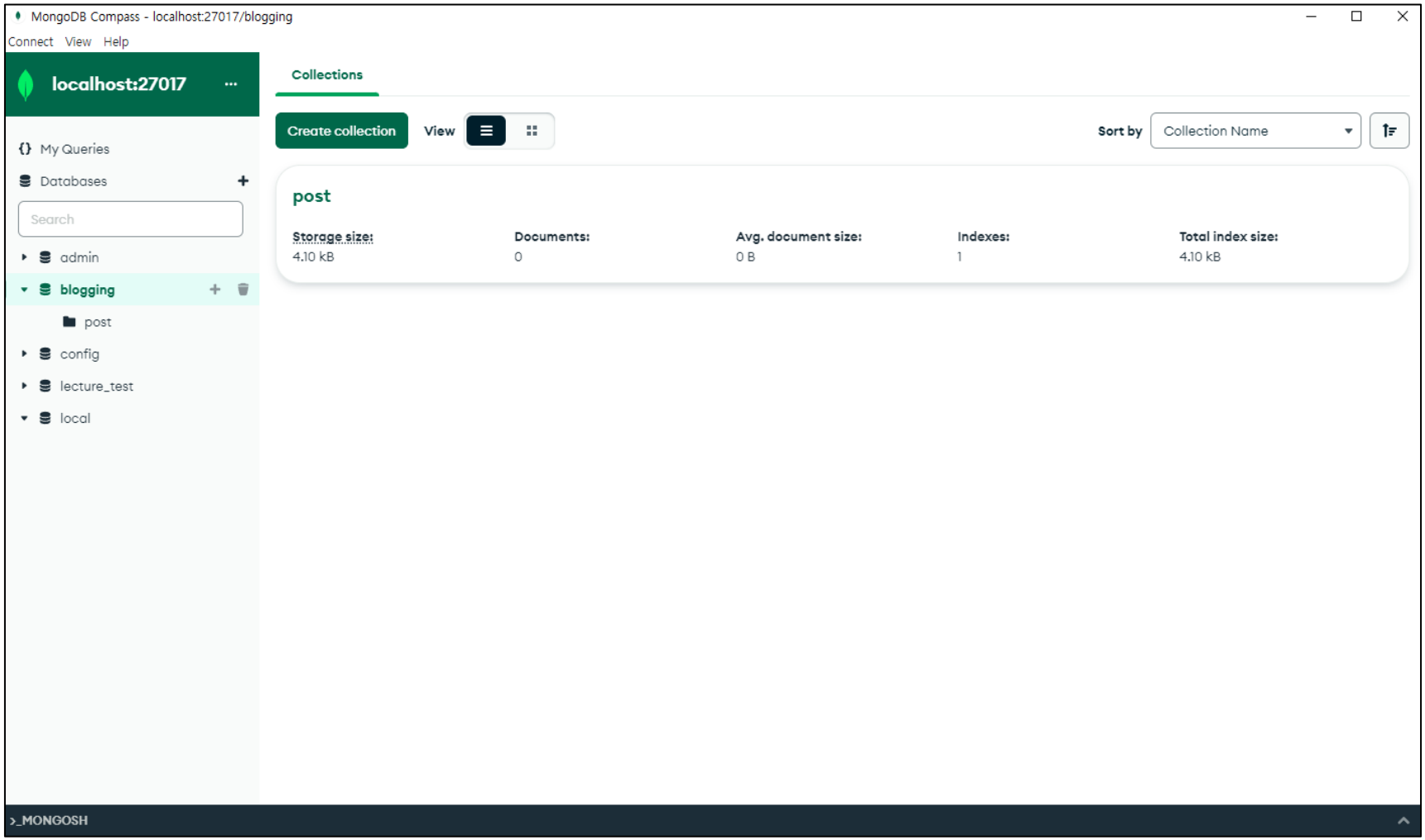
Collection

❖ Creating database and collection in MongoDB Compass



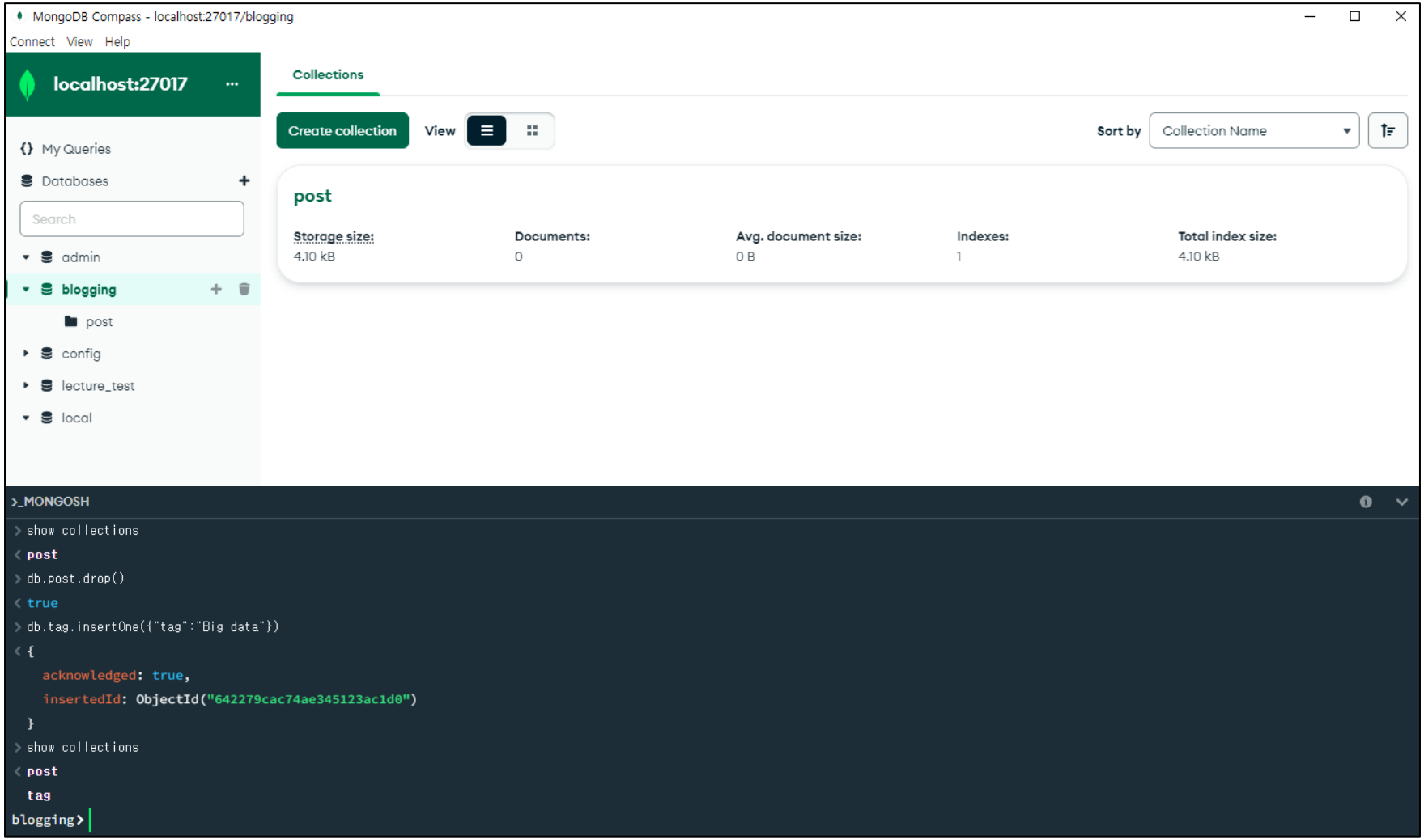
Collection

❖ New database with POST collection is created



Collection

❖ New TAG collection is created by inserting a document



Collection

- Explore your schema
  - You can easily visualize your schema to understand the frequency, types and ranges of fields in your data

  - Practice 6

① Create new ‘restaurant’ collection

② Download ‘restaurant’ JSON dataset from eCampus

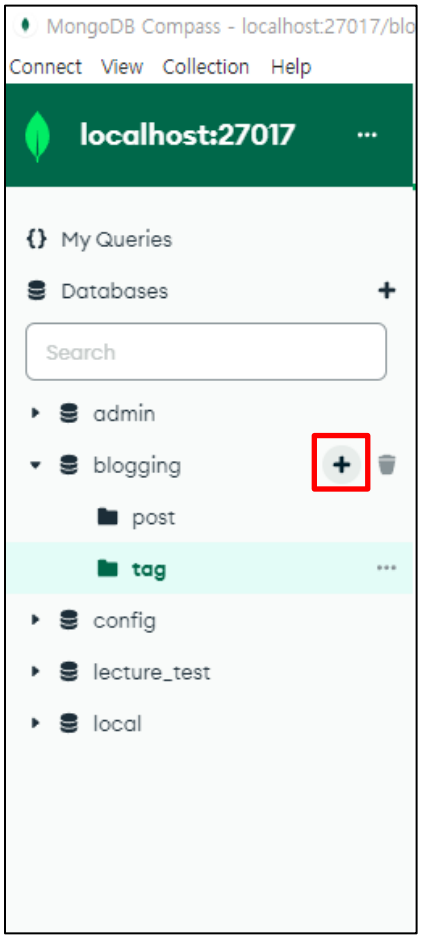
③ Add JSON data on ‘restaurant’ collection

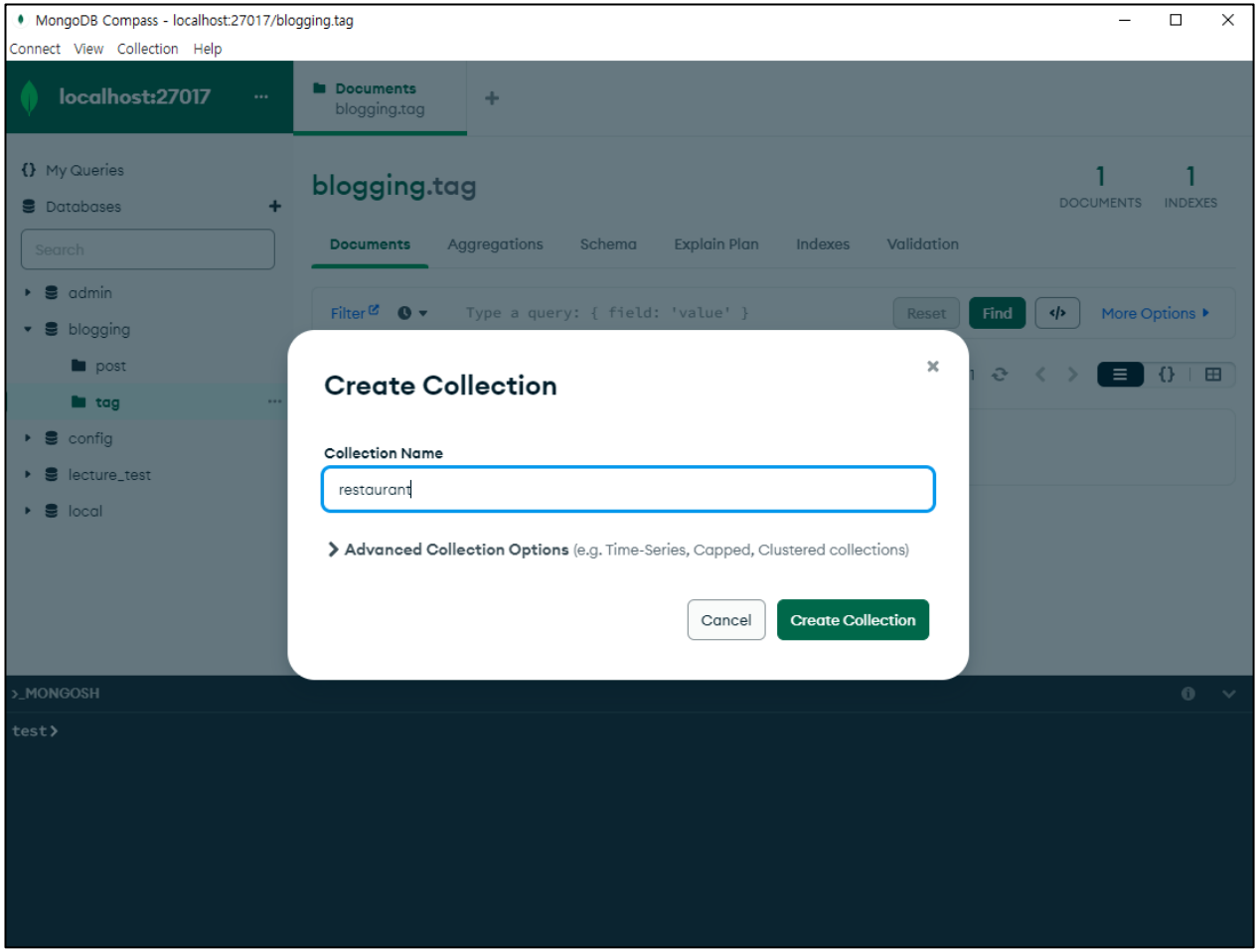
④ Click ‘schema’ option and run analysis

⑤ Check the results

Collection

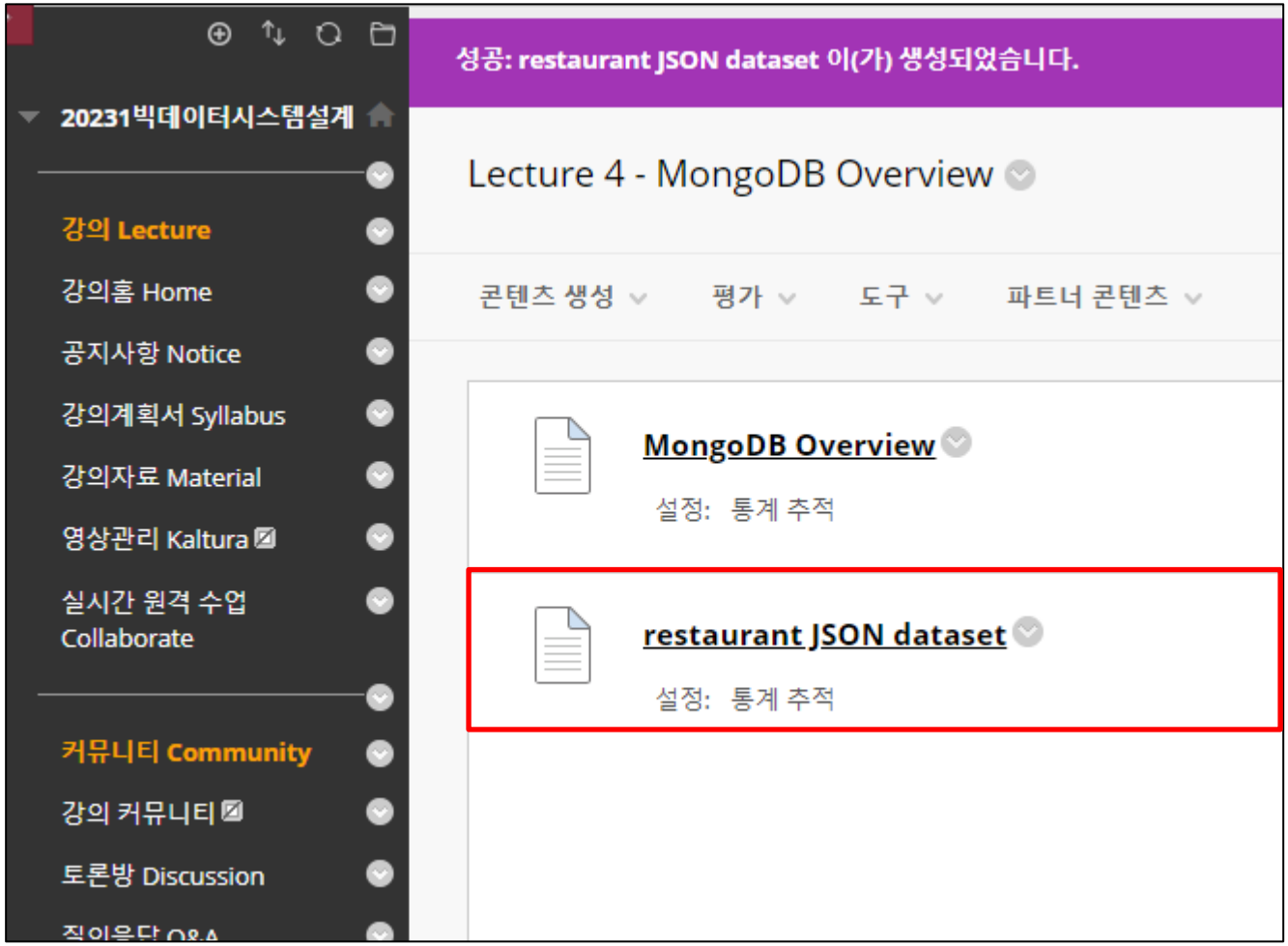
❖ Explore your schema





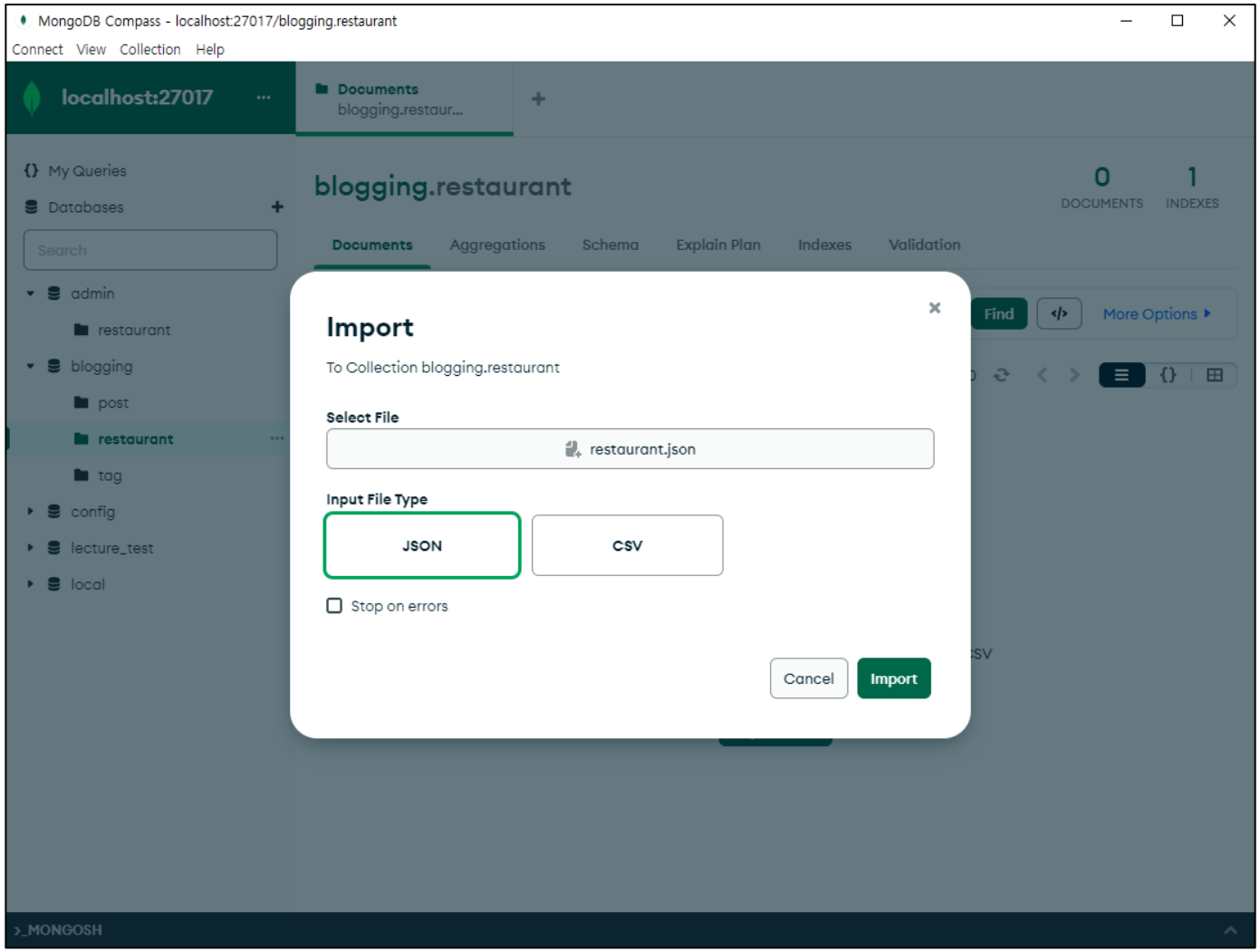
Collection

❖ Explore your schema



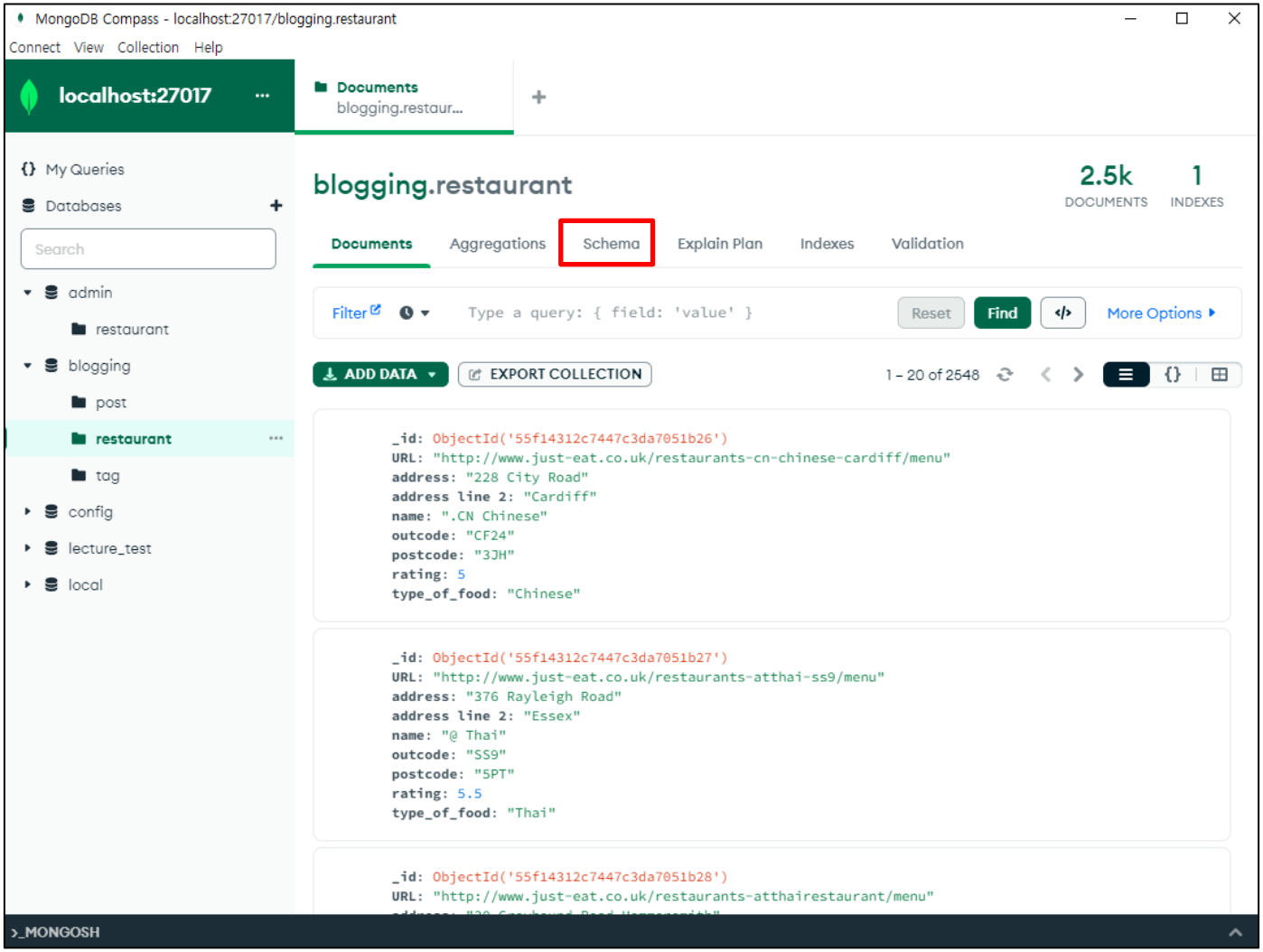
Collection

❖ Explore your schema



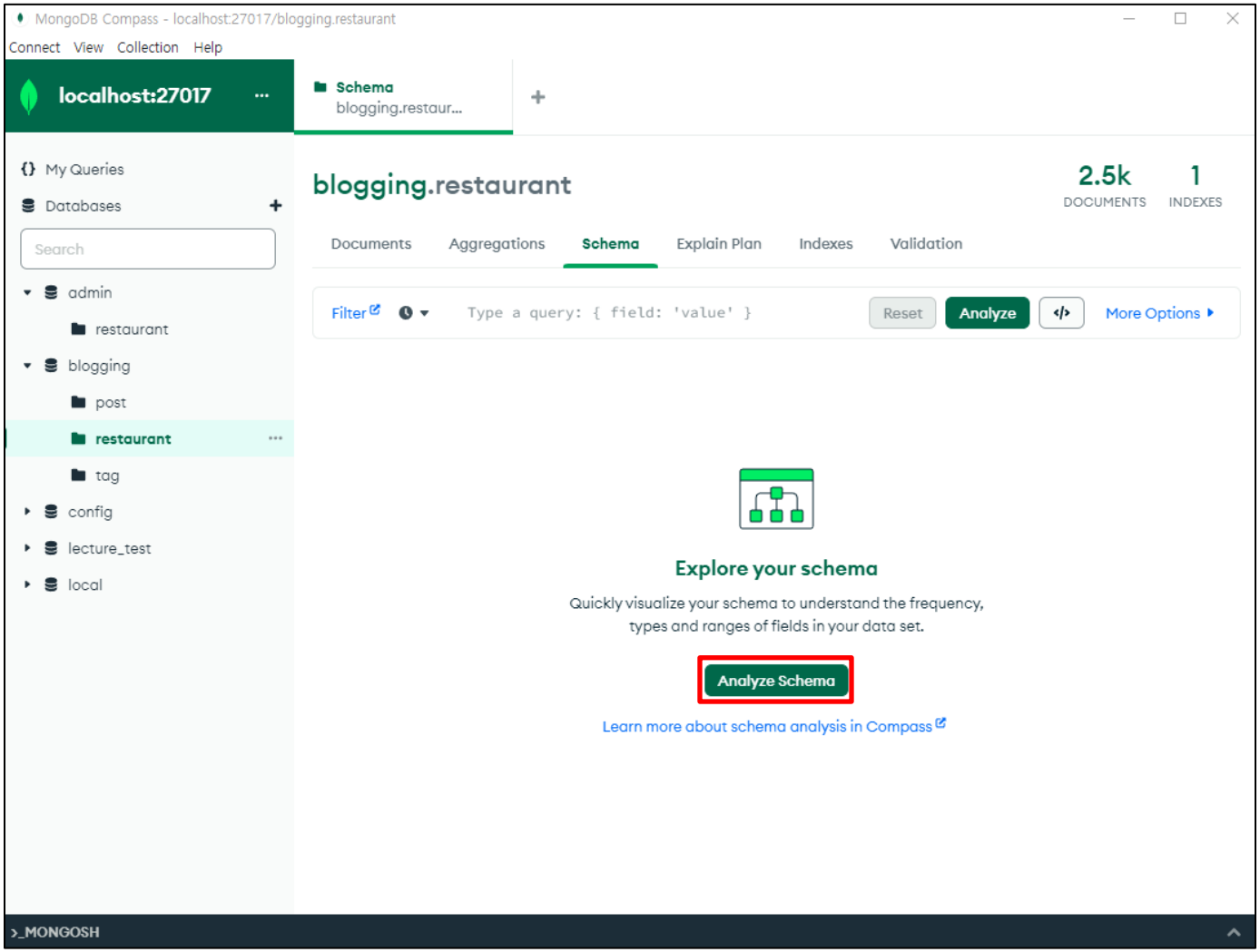
Collection

❖ Explore your schema



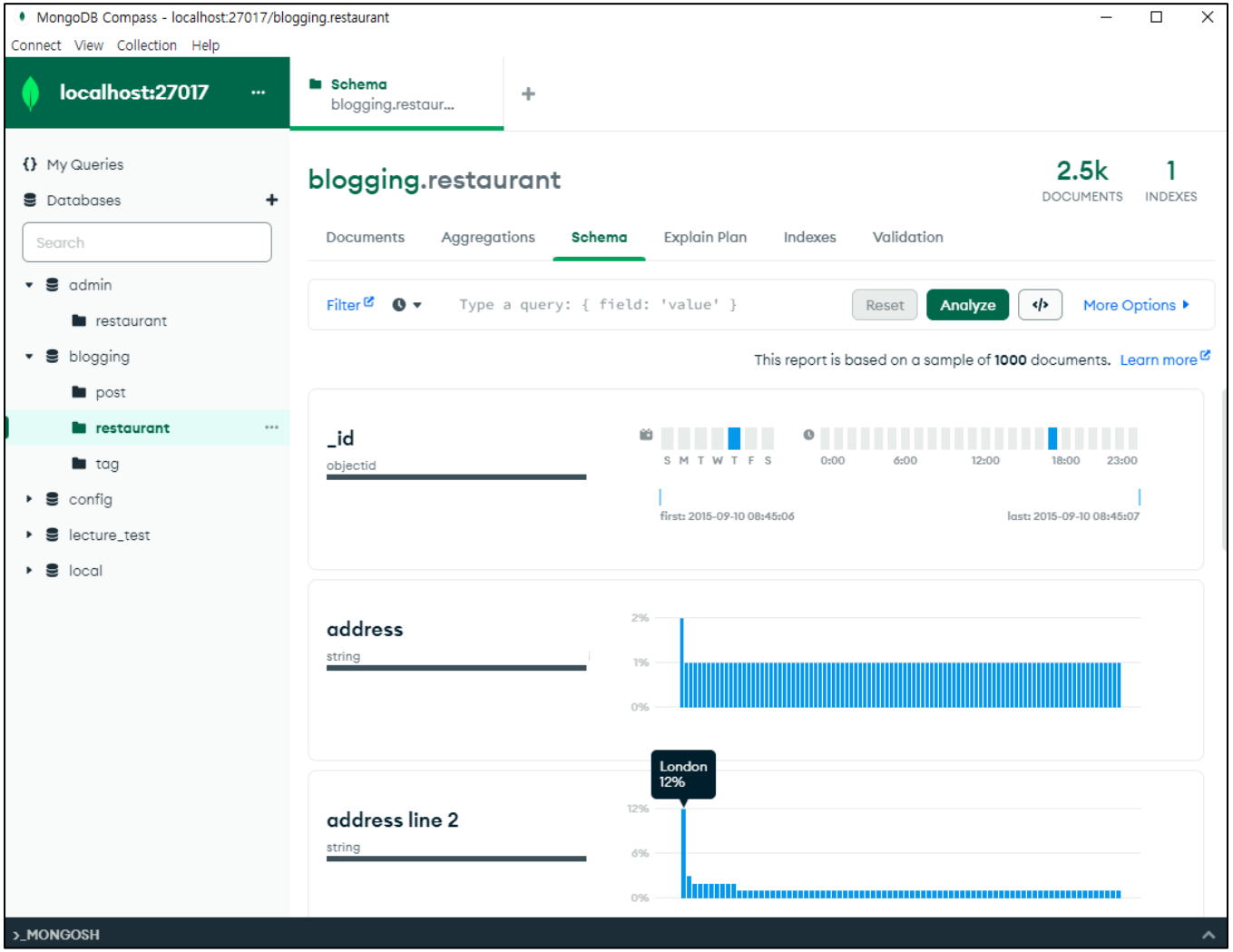
Collection

❖ Explore your schema



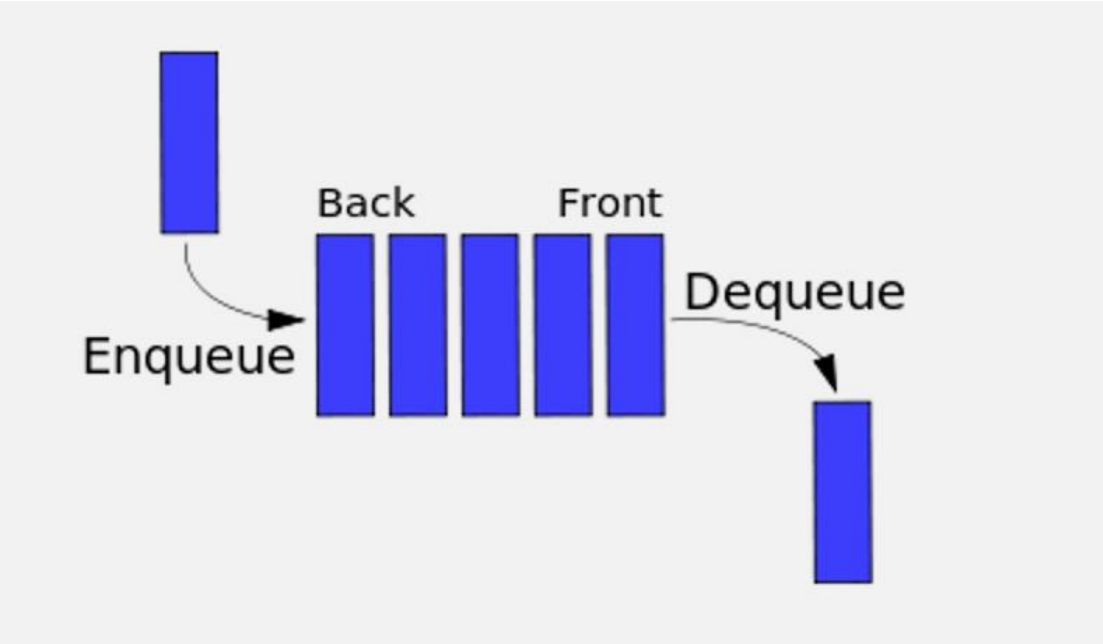
Collection

❖ Explore your schema



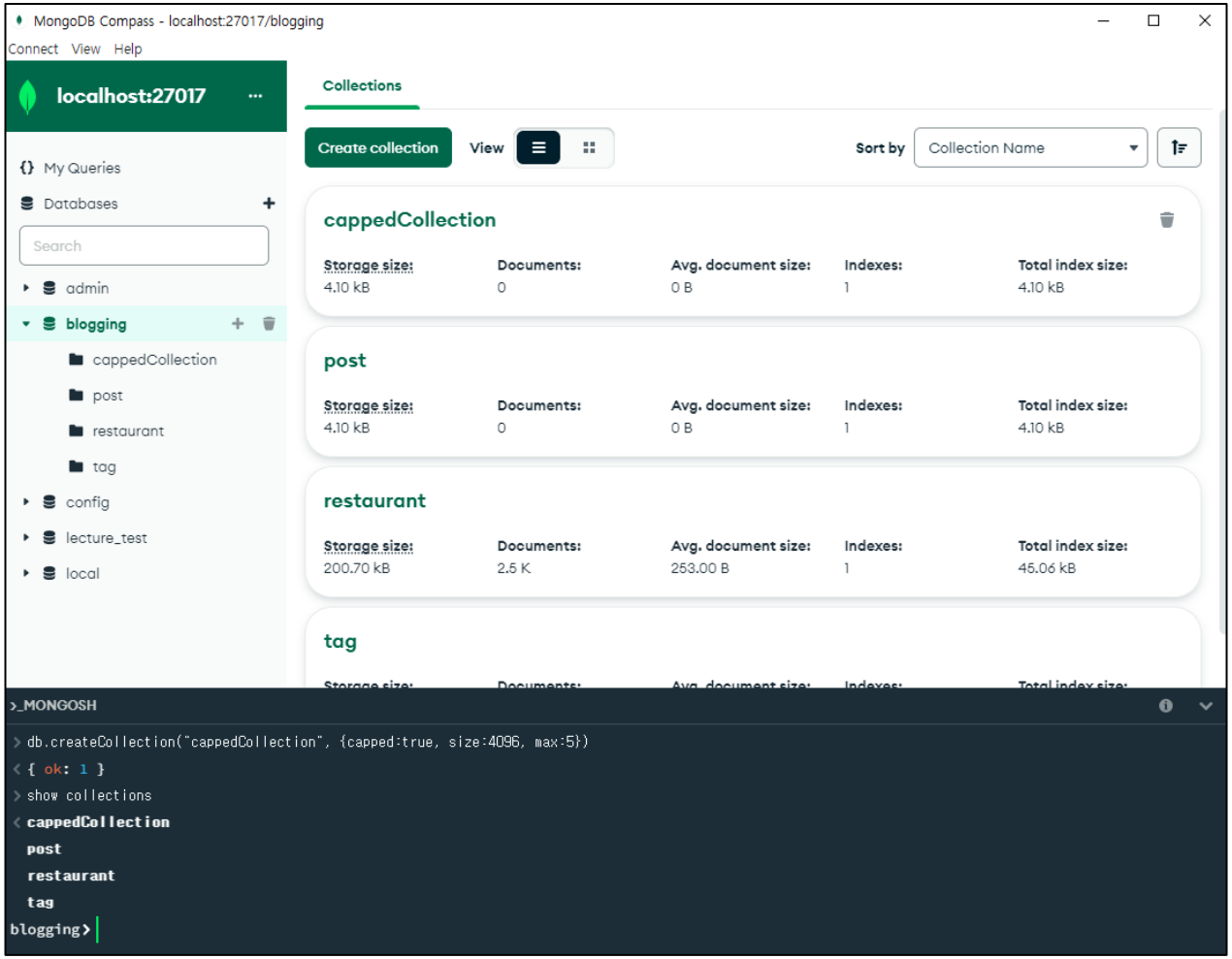
Collection

- Capped collection
  - To create a capped collection (Practice 7)
    - **db.createCollection("cappedCollection",{capped:true, size:4096, max: 5})**



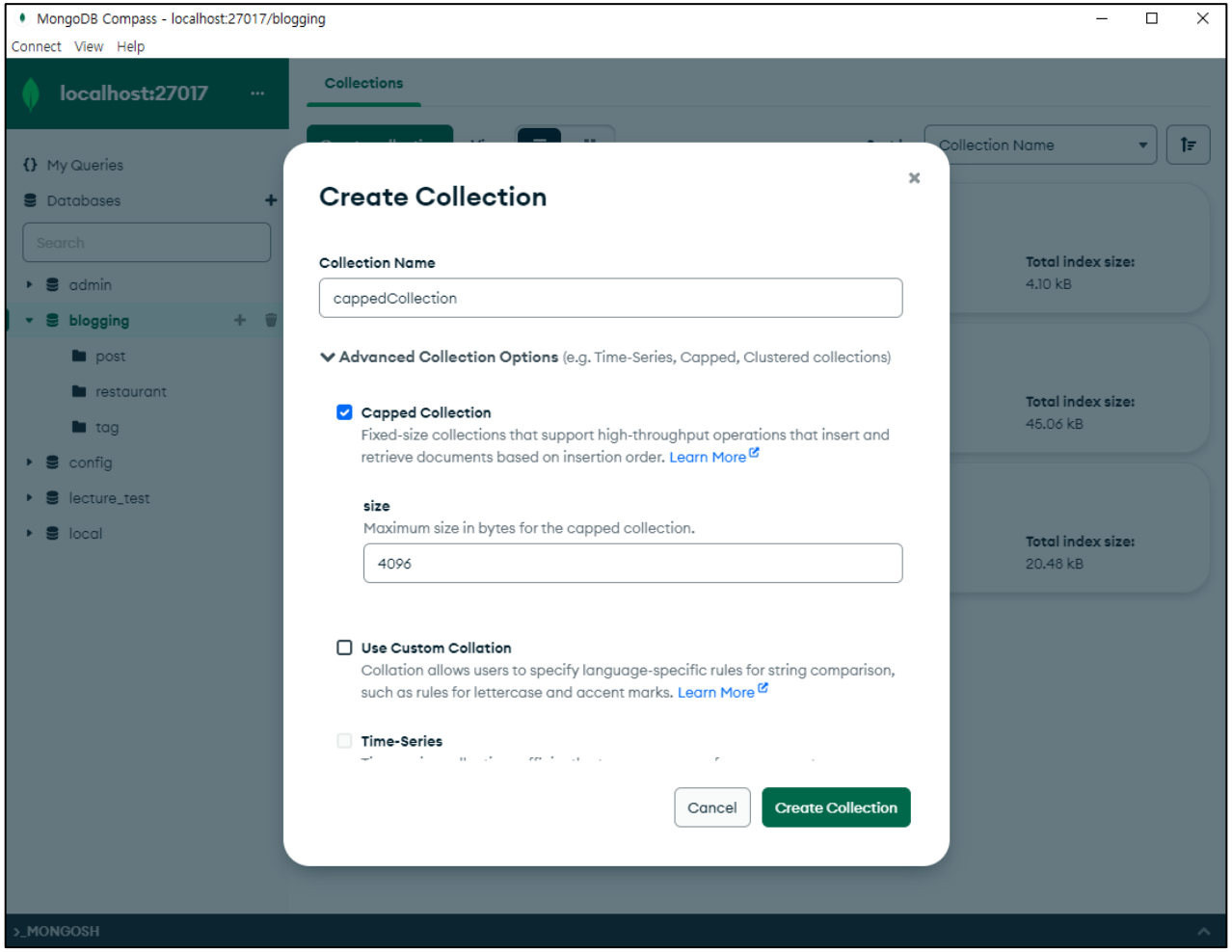
Collection

- Capped collection
  - You can create a capped collection from Mongo Shell



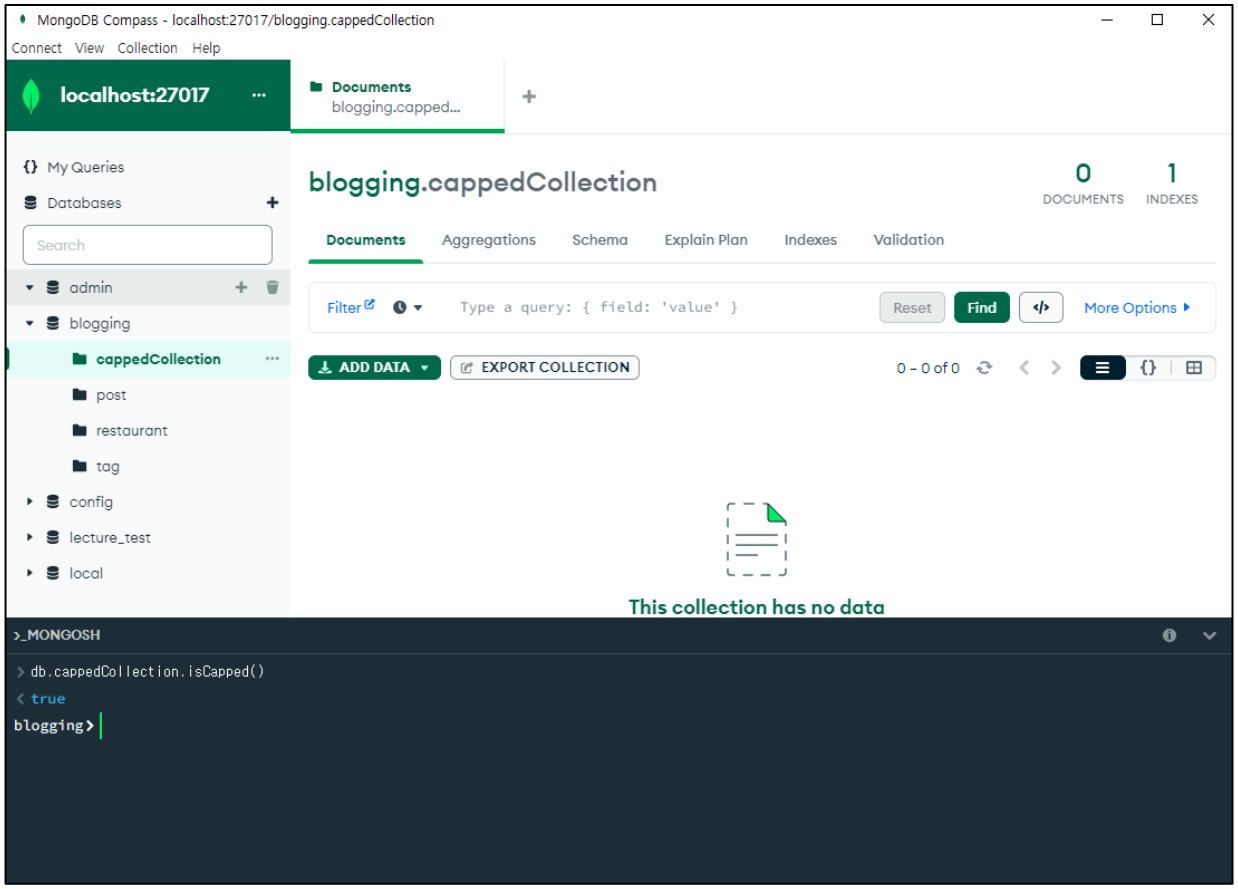
Collection

- Capped collection
  - You can also use Compass GUI for creating the capped collections



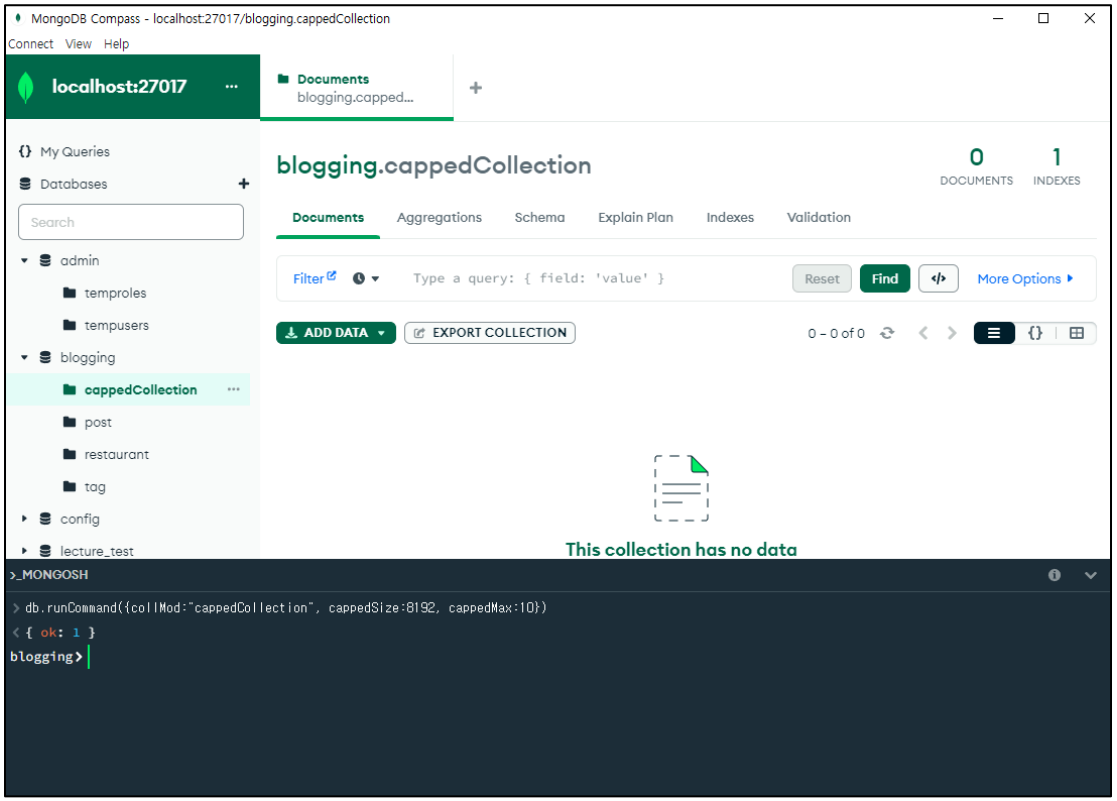
Collection

- Capped collection
  - Check capped collection
    - **db.cappedCollection.isCapped()**



Collection

- Capped collection
  - Change options of a capped collection
    - **db.runCommand({collMod:"cappedCollection", cappedSize: 8192, cappedMax: 10})**



Document

- A document is a set of key-value pairs
  - Commands
    - insertOne()
    - insertMany()

  - Practice 8
    - **db.post.insertOne({**

      title: 'Why Does Data Science Need to Be Successful?',

      description: 'Big Data challenges',

      by: 'Sherry Tiao',

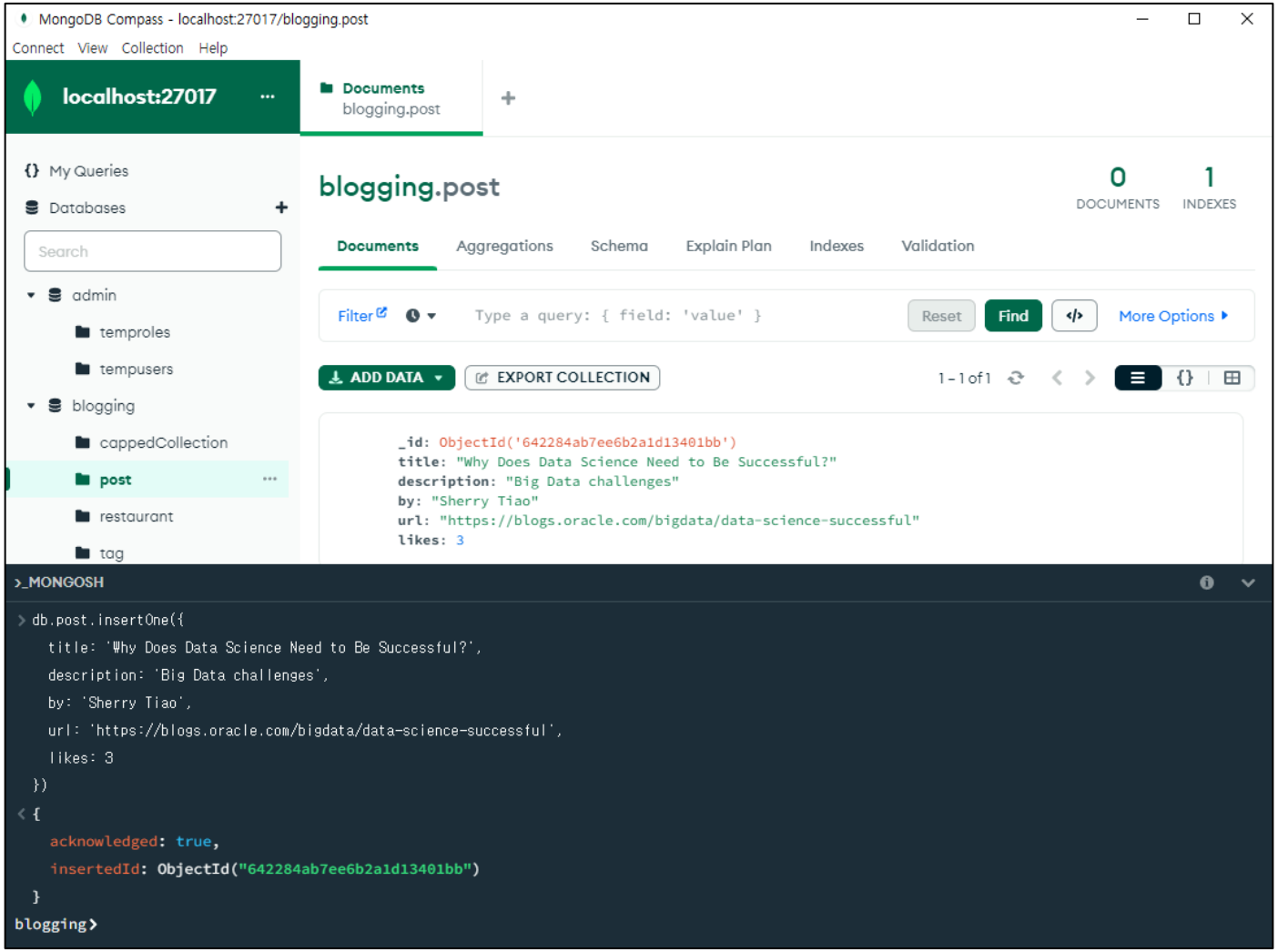
      url: 'https://blogs.oracle.com/bigdata/data-science-successful',

      likes: 3

      })

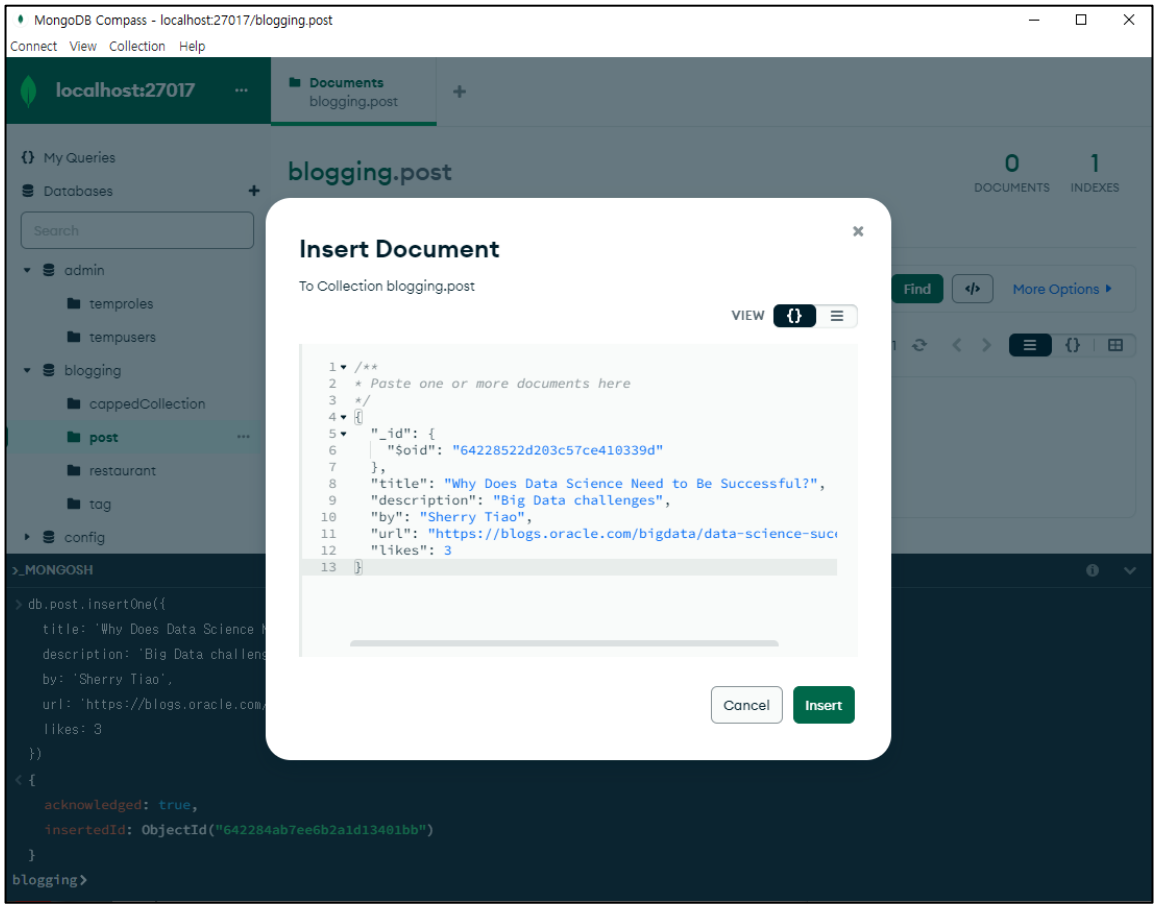
Document

❖ Result of Practice 8



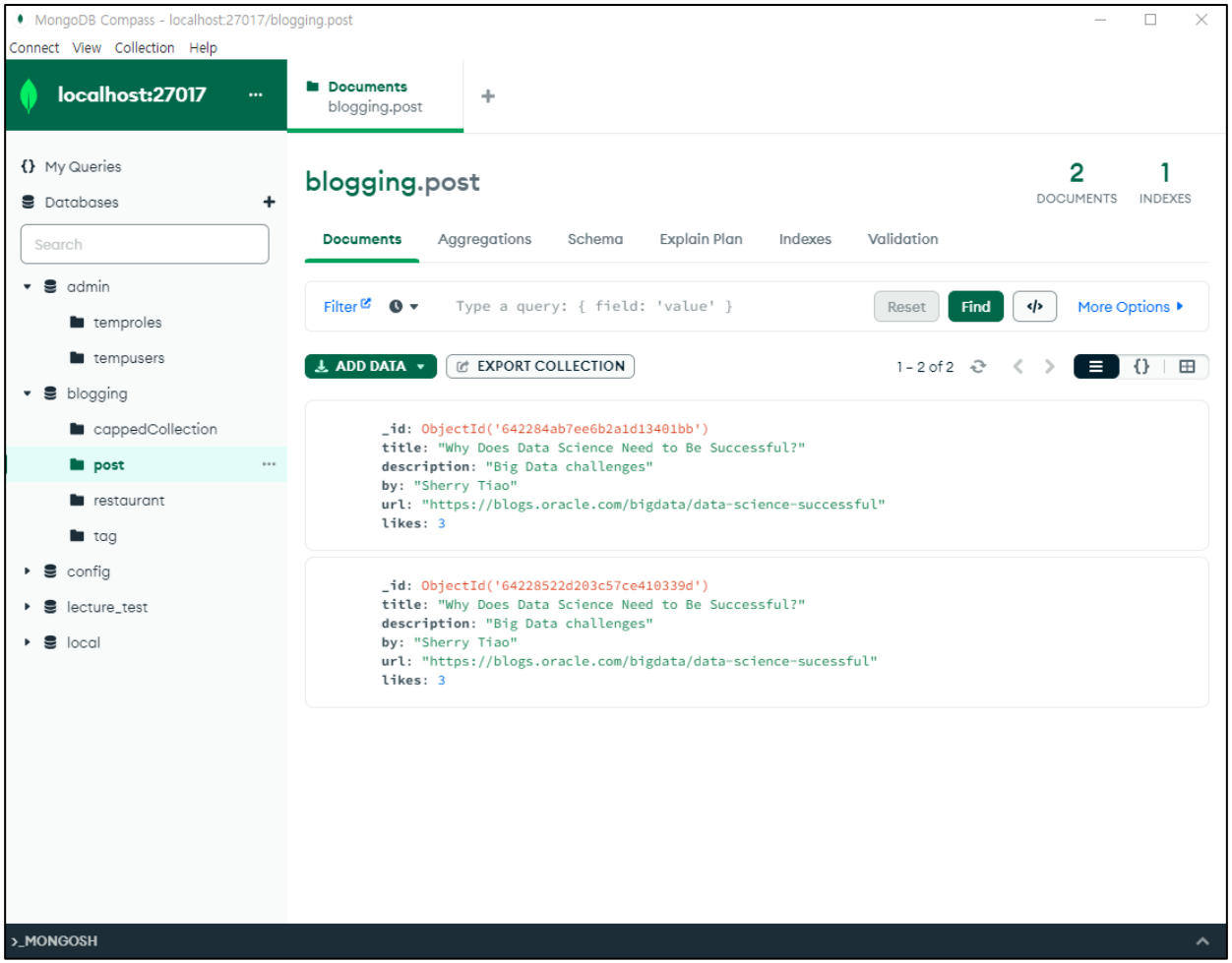
Document

- Use Compass GUI to insert the document
  - ADD DATA -> Insert Document
    - NOTE: Pay attention to quotations



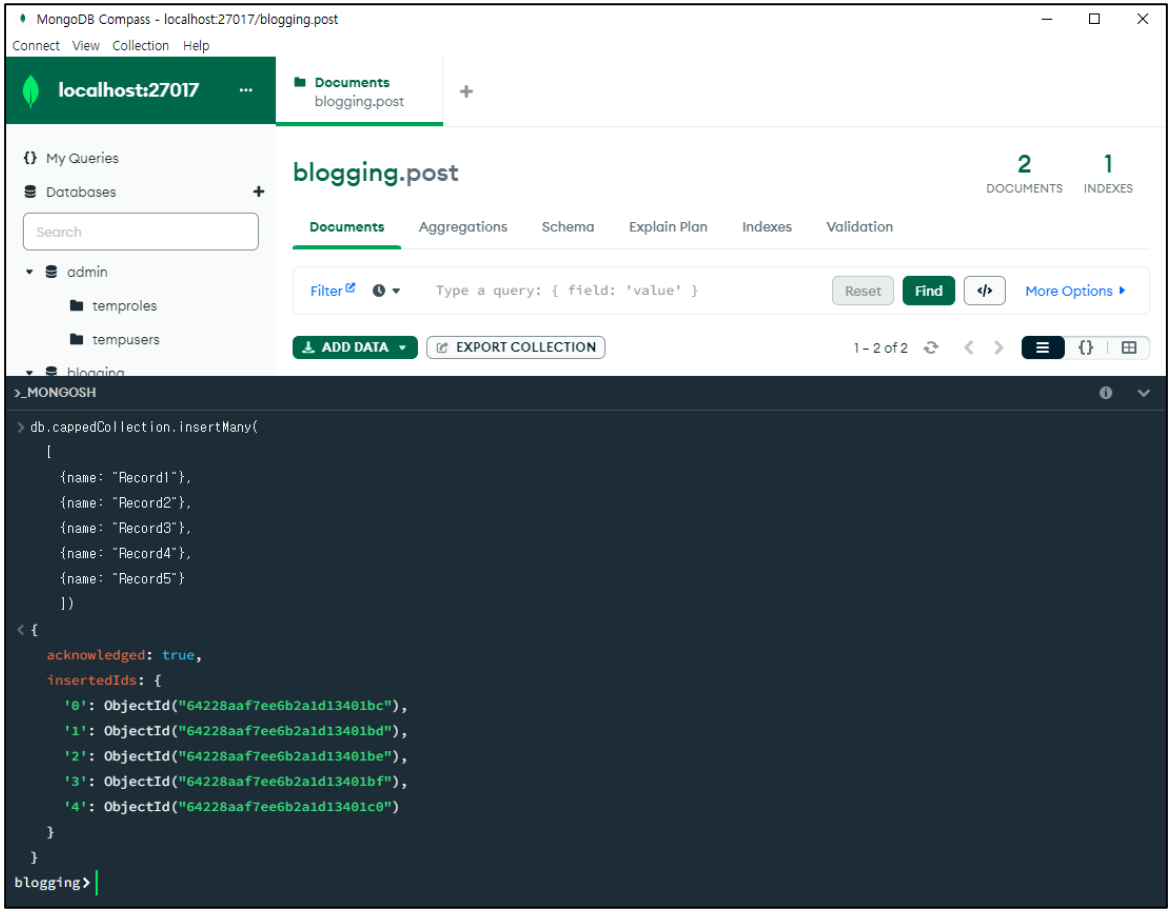
Document

- Use Compass GUI to insert the document
  - Check out the result



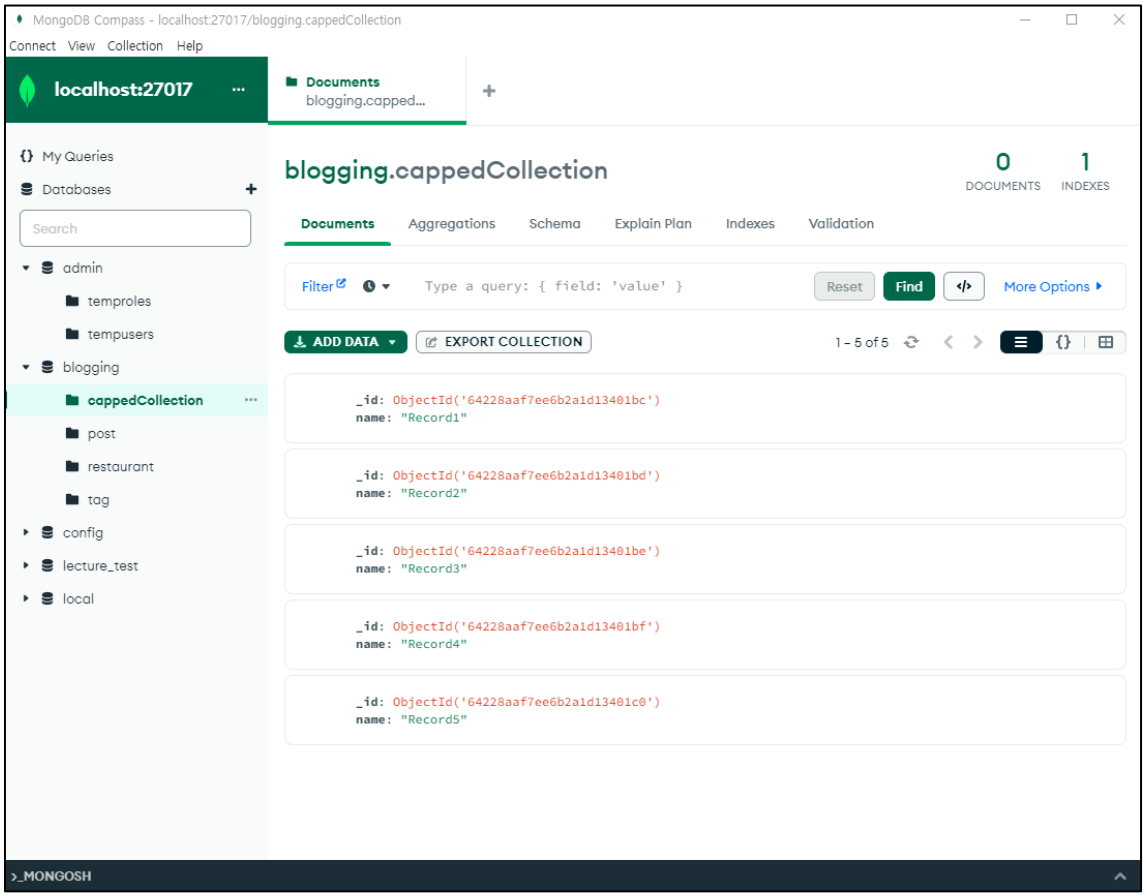
Document

- insertMany()
  - As the name suggests, you can use insertMany() to insert multiple documents



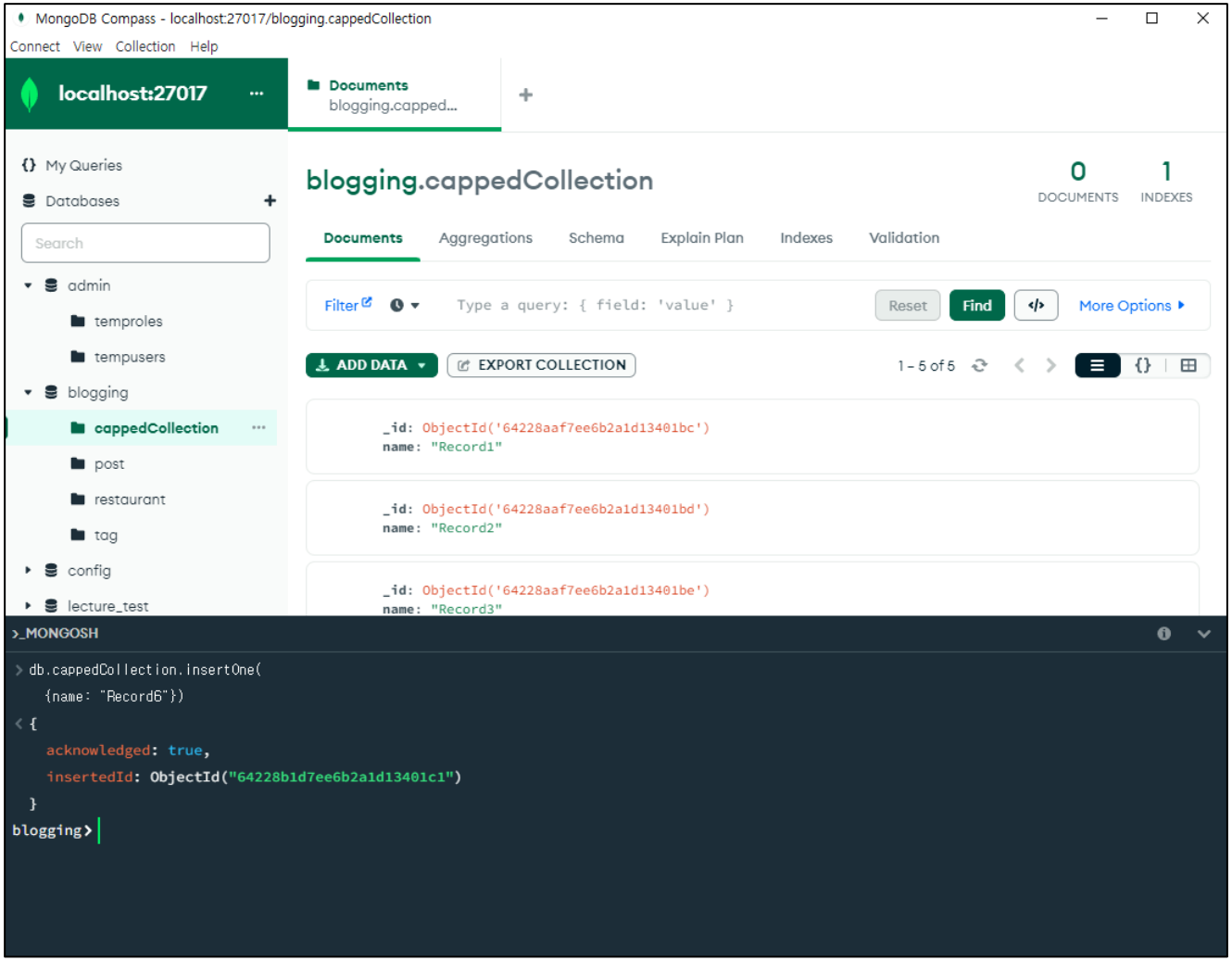
Document

- insertMany()
  - As the name suggests, you can use insertMany() to insert multiple documents



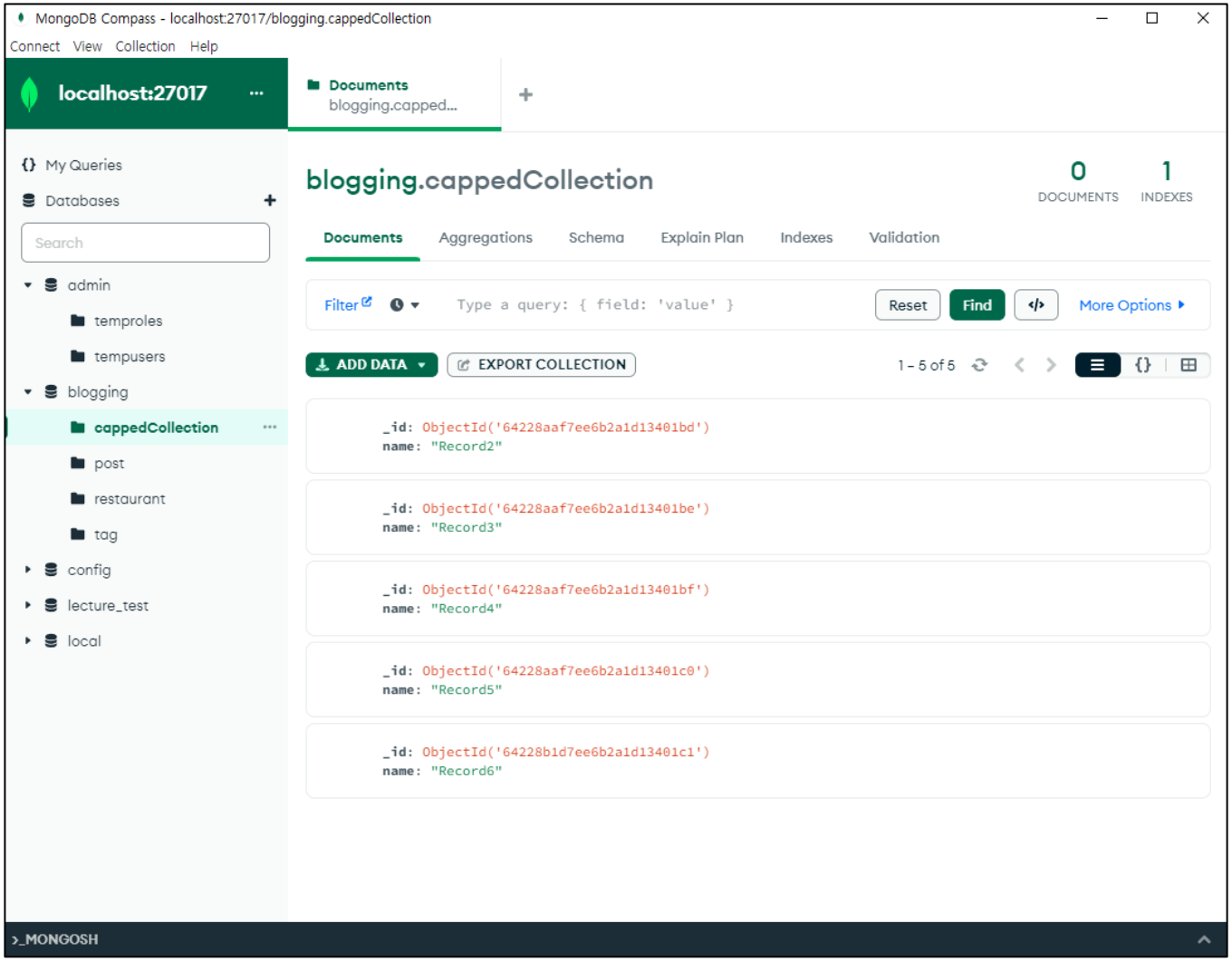
Document

❖ Test Capped Collection by inserting Record6



Document

❖ Test Capped Collection by inserting Record6

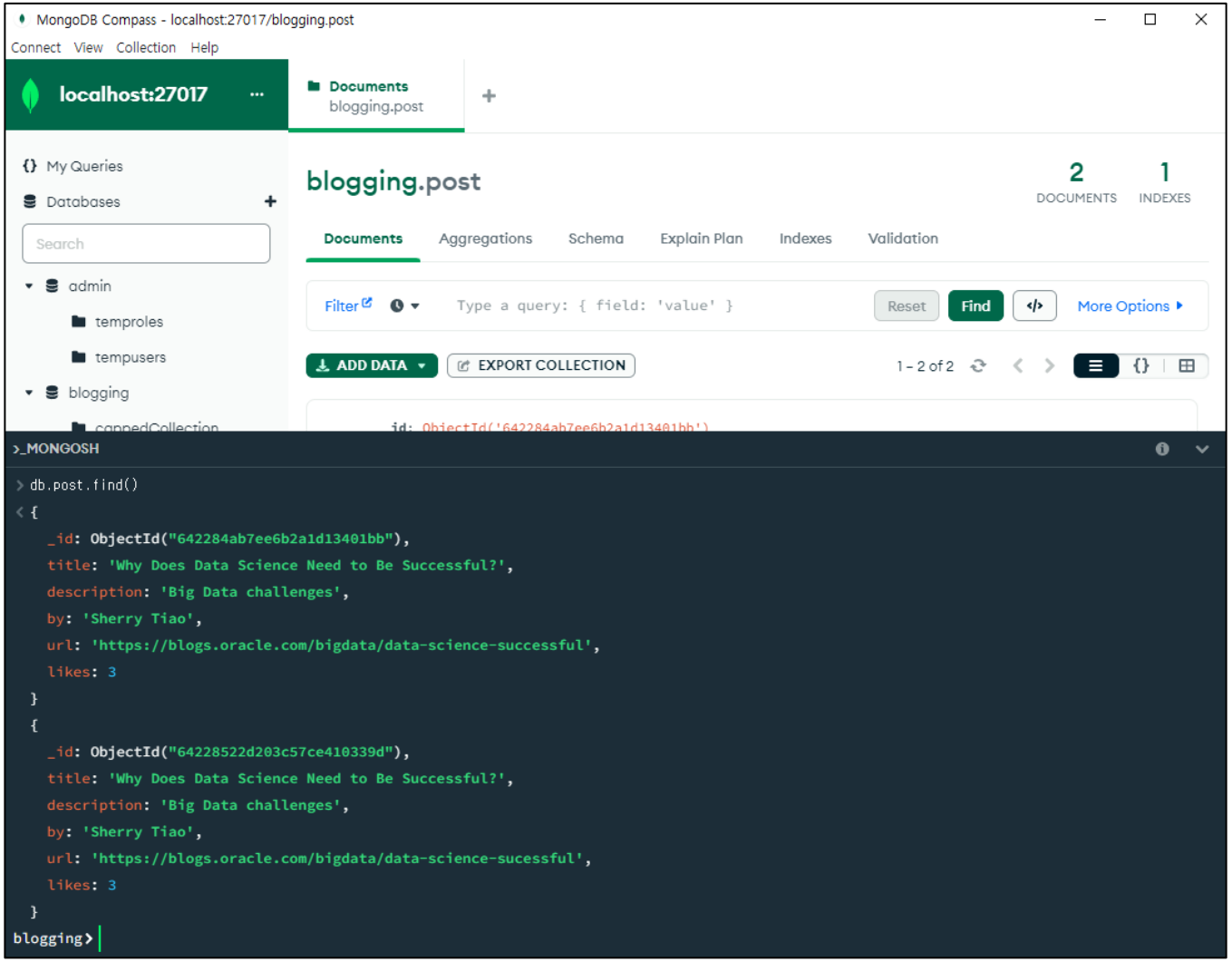


Query Document

- Select all documents in a collection
  - To query the post collection, simply define the following select criteria
    - **db. post.find()**
  - To display the results in a formatted way, you can use pretty() method
    - *db.COLLECTION_NAME.find().pretty()*
    - **db. post.find().pretty()**
      - Compare difference between them
  - This operation corresponds to the following SQL statement
    - SELECT * FROM post

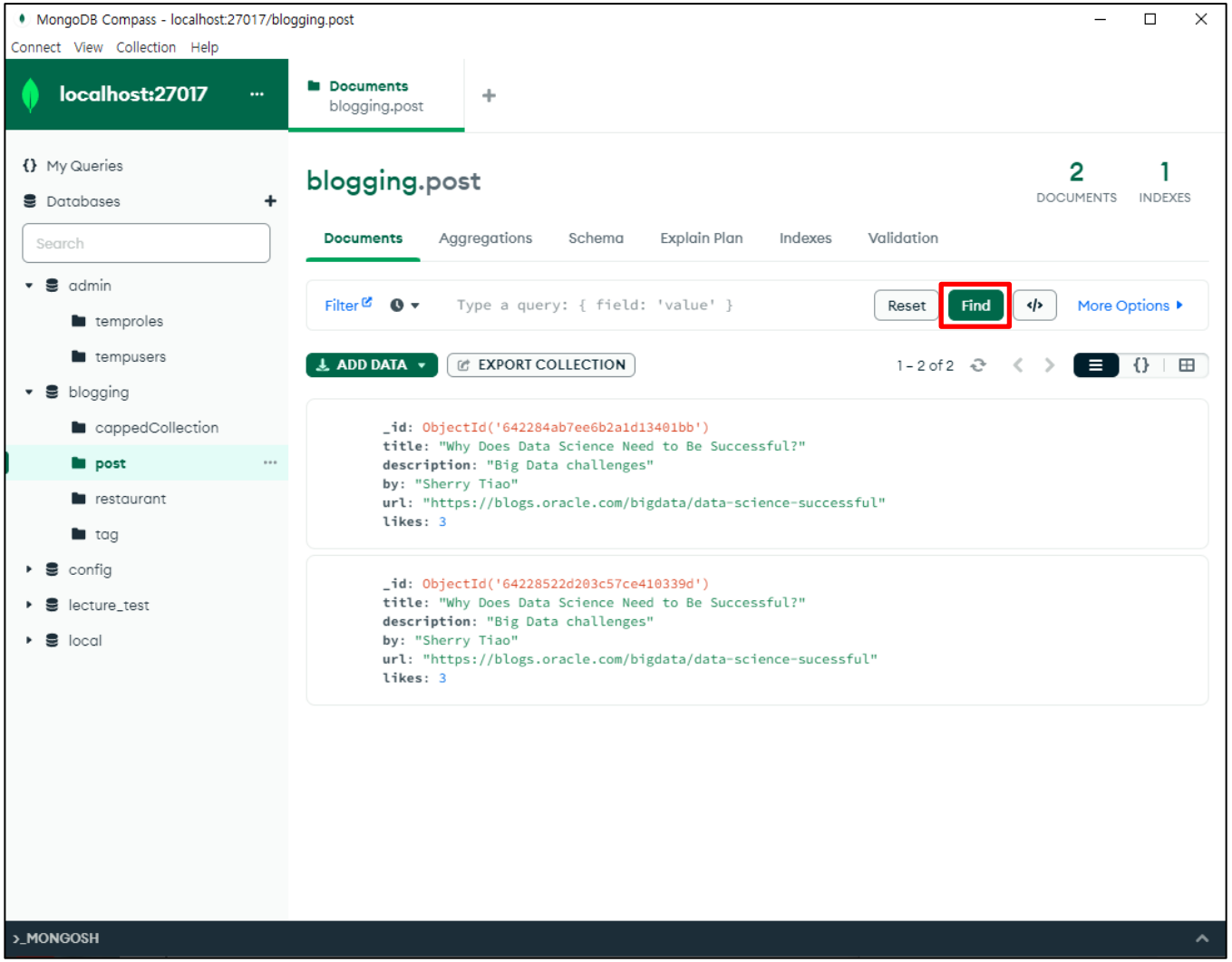
Query Document

❖ Result of querying POST collection in MongoDB Shell



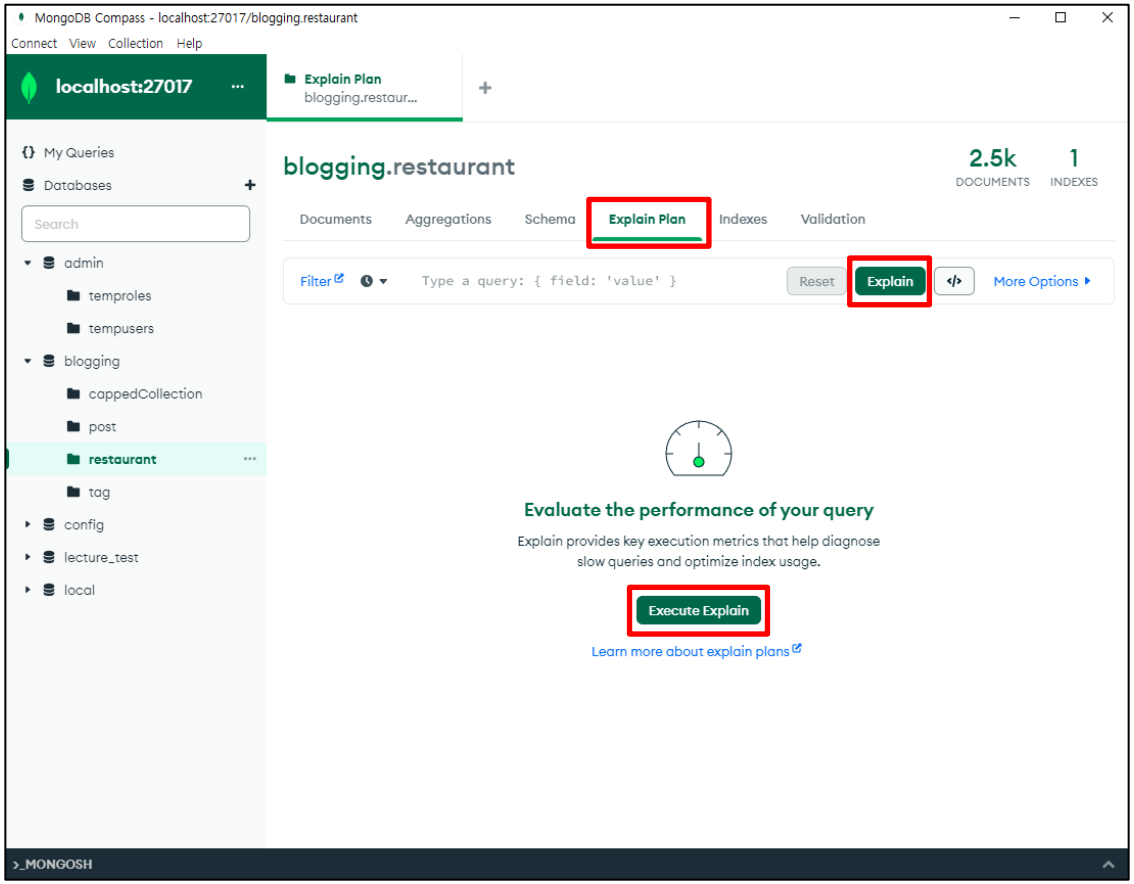
Query Document

❖ You can also avoid writing complex queries by Compass GUI



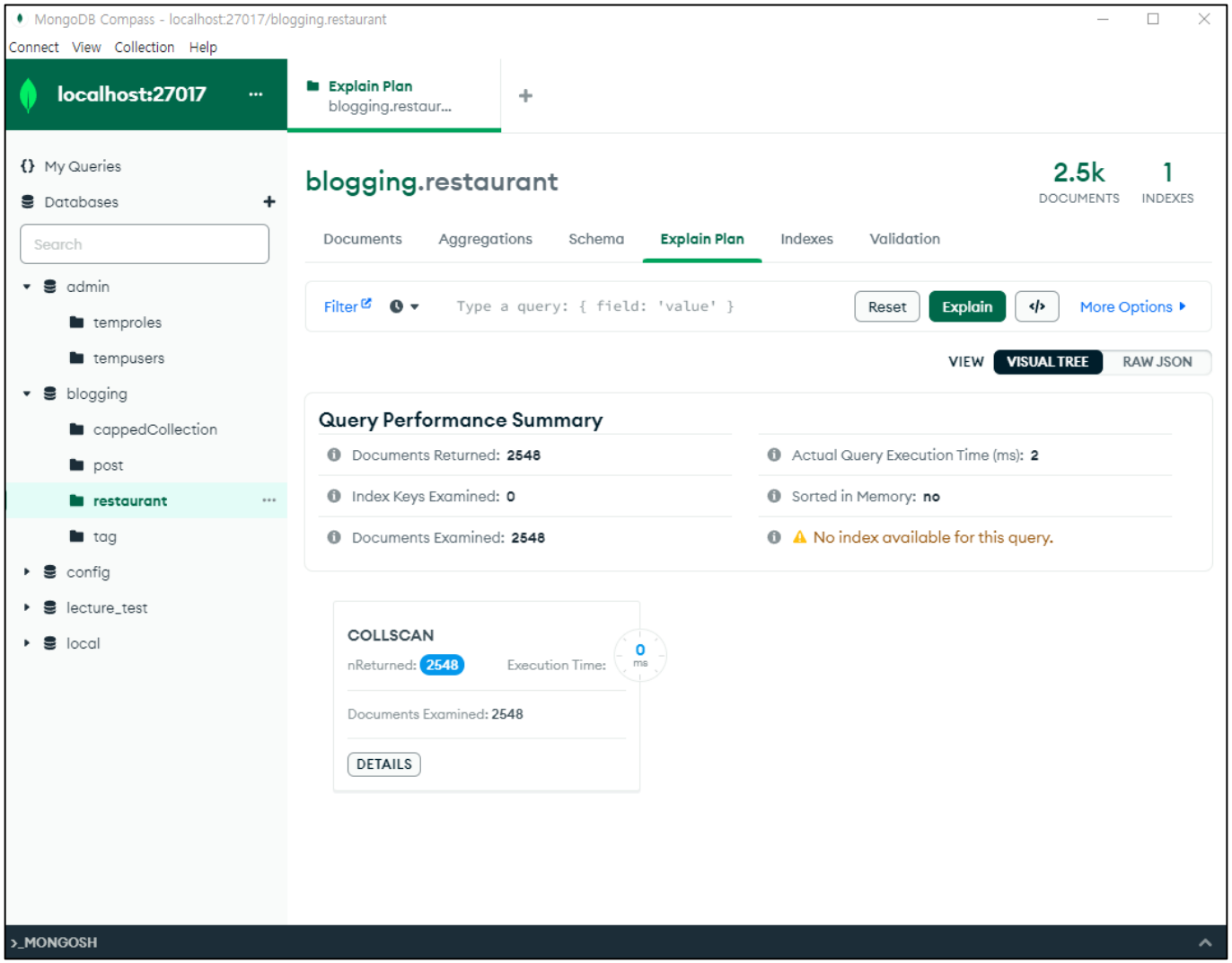
Query Document

- Evaluate the performance of your query
  - Explain provides key execution metrics that help diagnose slow queries and optimize index usage (use restaurant collection)



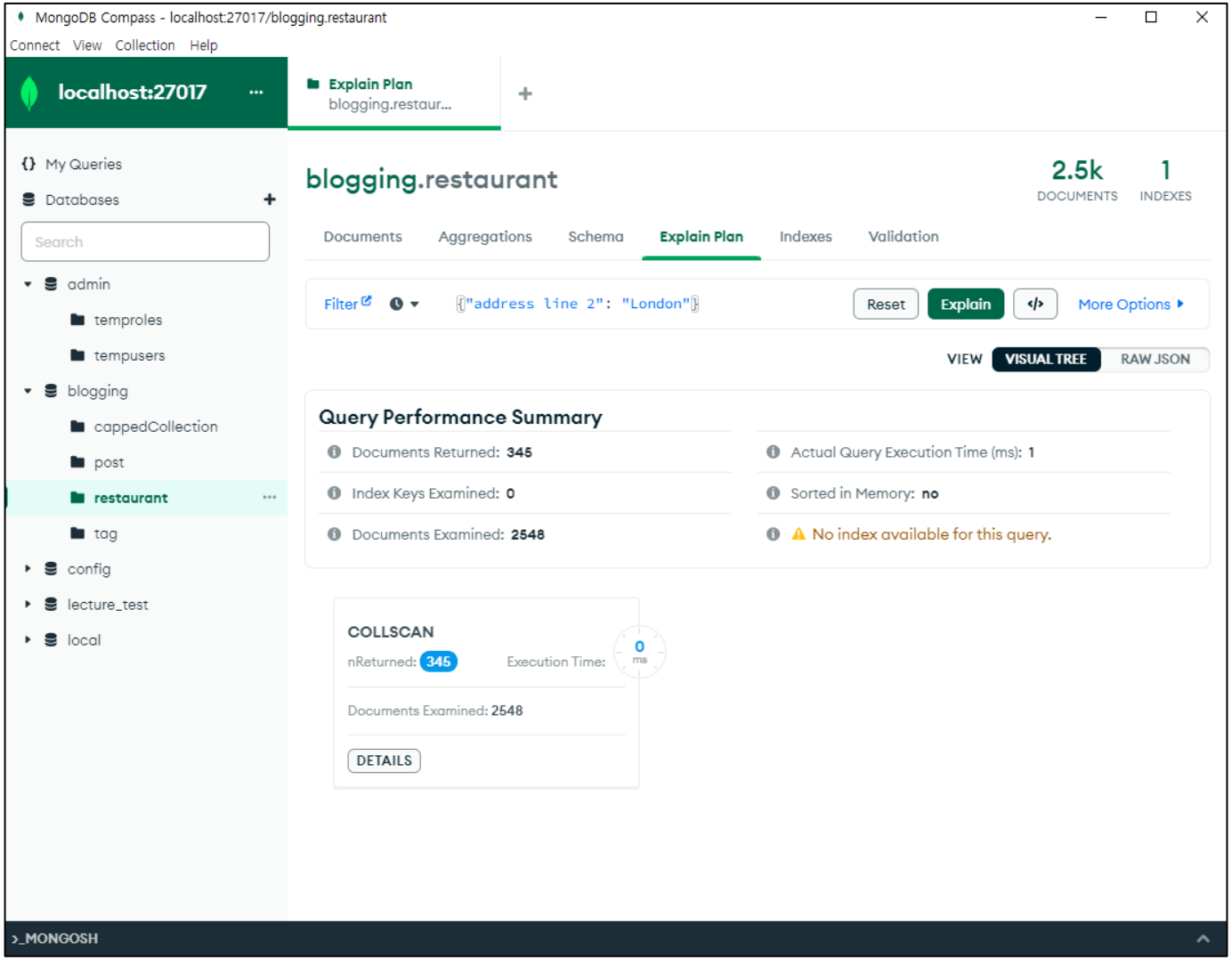
Query Document

- Evaluate the performance of your query



Query Document

- Evaluate the performance of your query



Final Task

- Do the following tasks for the final task
  1. Clean up the post collection
  2. Change the post collection into capped collection
     - With maximum number of documents 5
  3. Test the post collection by inserting at least 5 blogs
     - Use insertMany() function
  4. Insert one more document into the post collection
     - Use insertOne() function
     - Check the behavior of the post collection
  5. Perform queries and check out the performance using ‘restuarants’ dataset

Questions?

SEE YOU NEXT TIME!