**Module: Node - Node with Mongodb**

**Q1. What is MongoDB.**

**Ans:**

* **Flexible Storage:** Instead of forcing your data into rigid tables like in traditional databases, MongoDB uses a more adaptable format called "documents," which are like JSON objects. This allows you to store data in a way that fits your needs without worrying about strict structures.
* **Easy to Scale:** MongoDB is designed to handle a lot of data and can easily grow as your needs increase. It’s like having a filing cabinet that can magically expand to fit more documents.
* **Fast and Efficient:** It’s built to quickly find and manage your data, even if you have lots of it, making it great for applications that need quick access to information.
* **No Fixed Schema:** You don’t need to define the structure of your data ahead of time. You can store different types of data in the same place without having to change the whole setup.

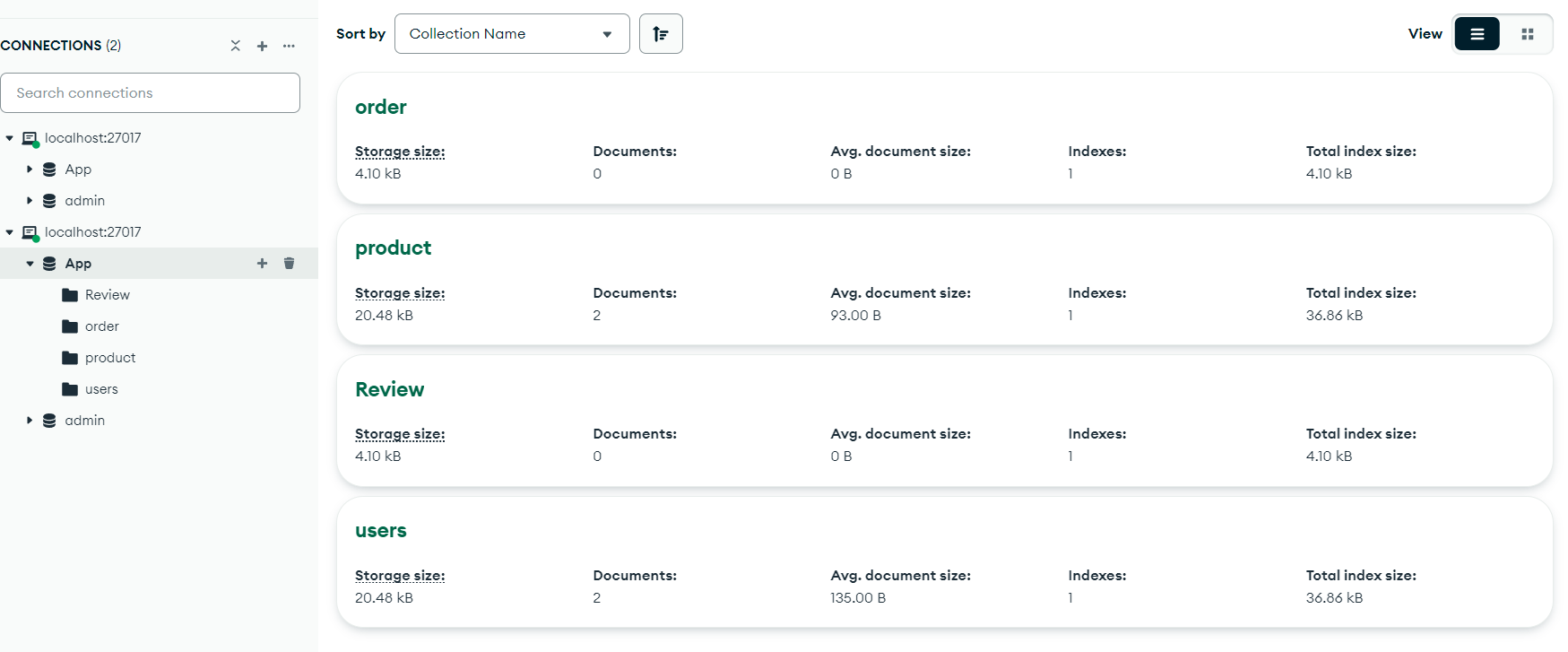
**Q2. What is difference between mongodb and SQL**

**Ans:**

|  |  |
| --- | --- |
| **MongoDB** | **SQL** |
| * MongoDB is an open-source database developed by MongoDB, Inc. MongoDB stores data in JSON-like documents that can vary in structure. It is a popular NoSQL database. | * MySQL is a popular open-source relational database management system (RDBMS) that is developed, distributed and supported by Oracle Corporation. |
| * In MongoDB, each individual records are stored as ‘documents’. | * In MySQL, each individual records are stored as ‘rows’ in a table. |
| * MongoDB is what is called a NoSQL database. | * MySQL as the name suggests uses Structured Query Language (SQL) |
| * C, C++, Java, Python, JavaScript, PHP, Ruby, Perl | * C, C++ |
| * Document-oriented non-relational database (key-value structure) | * Relational database with tables format. |

**Q3. Create Require collections for online shopping app and documents.**

**i.User ii. Product category iii. Product iv. Order v. Review**

Ans: 

**Q4. Write command to show all data from product collections and sort in ascending order.**

**Ans:**

* Show all data
* Sort in ascending order

db.product.find().sort({Price:1})

{

\_id: ObjectId('66e3d37396a2df937fb0ddd7'),

id: '2',

Title: 'women',

Description: 'women t-shirt',

Price: '2000'

}

{

\_id: ObjectId('66e3d37396a2df937fb0ddd7'),

id: '2',

Title: 'women',

Description: 'women t-shirt',

Price: '2000'

}

**Q5. Update product price for particular product.**

**Ans:** db.products.updateOne(

{ \_id: ObjectId("PRODUCT\_ID") }, // Replace with your product ID

{ $set: { price: 29.99 } } // Set the new price

);

**Q6. Write command to delete particular document and collection.**

**Ans:** db.delete("/deleteproduct", async (req, resp) => {

let data = await productModel.deleteOne(req.body);

resp.send(data);

})

db.product.drop()