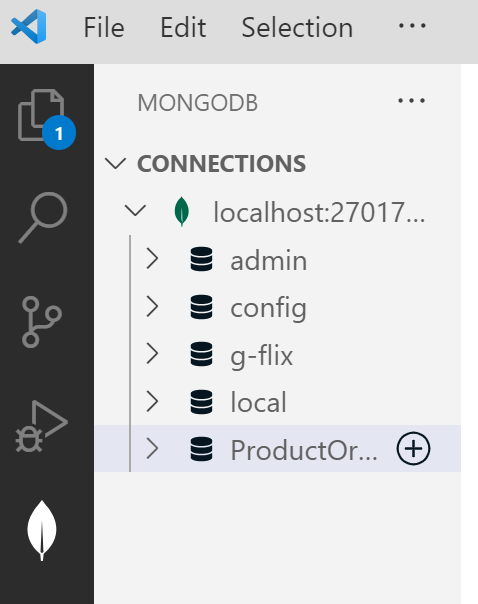
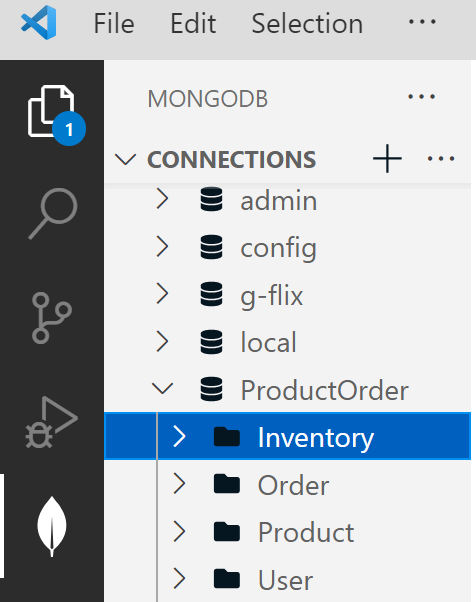
1. Open VS Code and connect to MONGODB



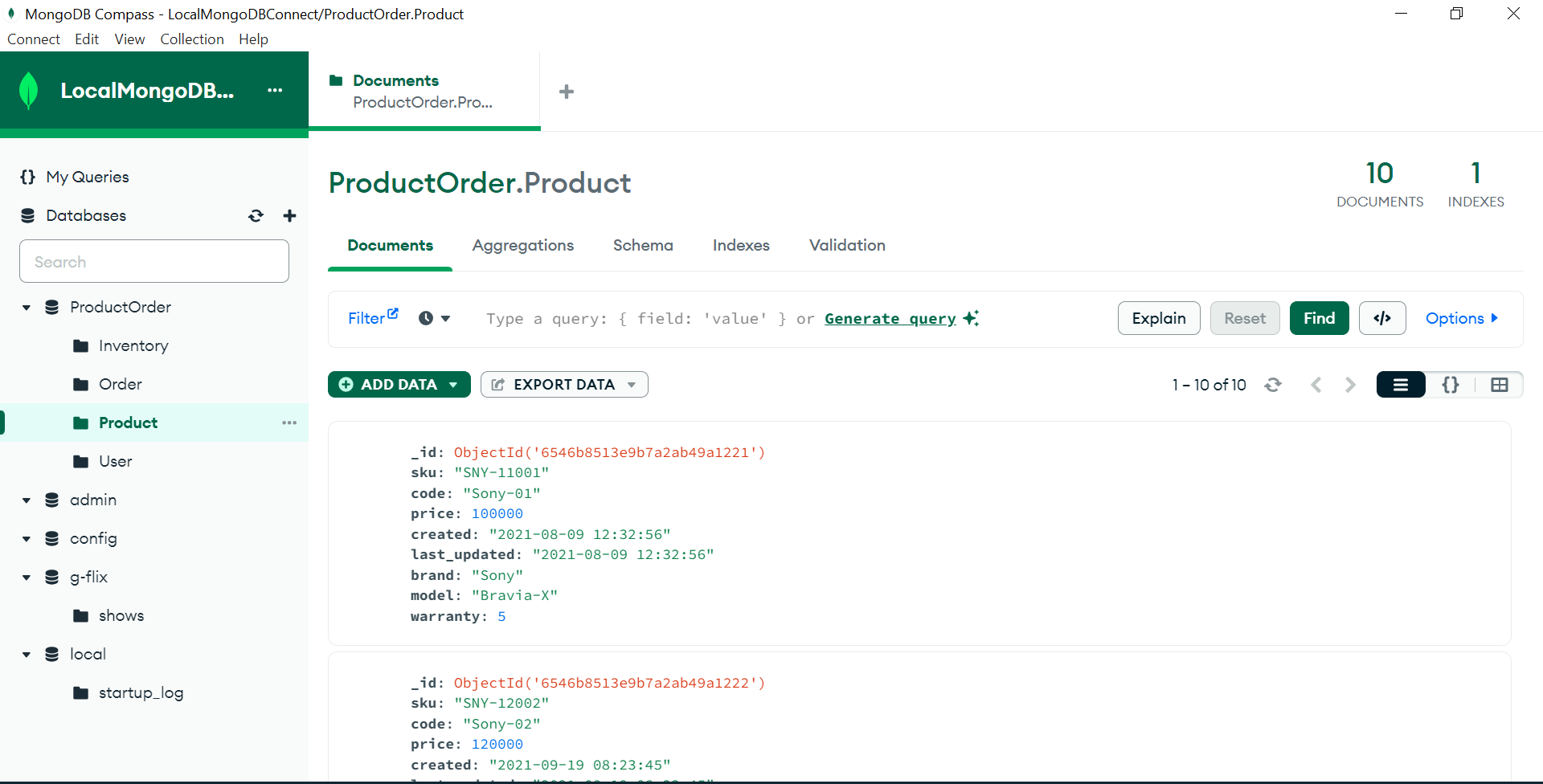
2. 2. Create a database "ProductOrder" and create collections

"Product","Inventory","User", and "Order" in it.

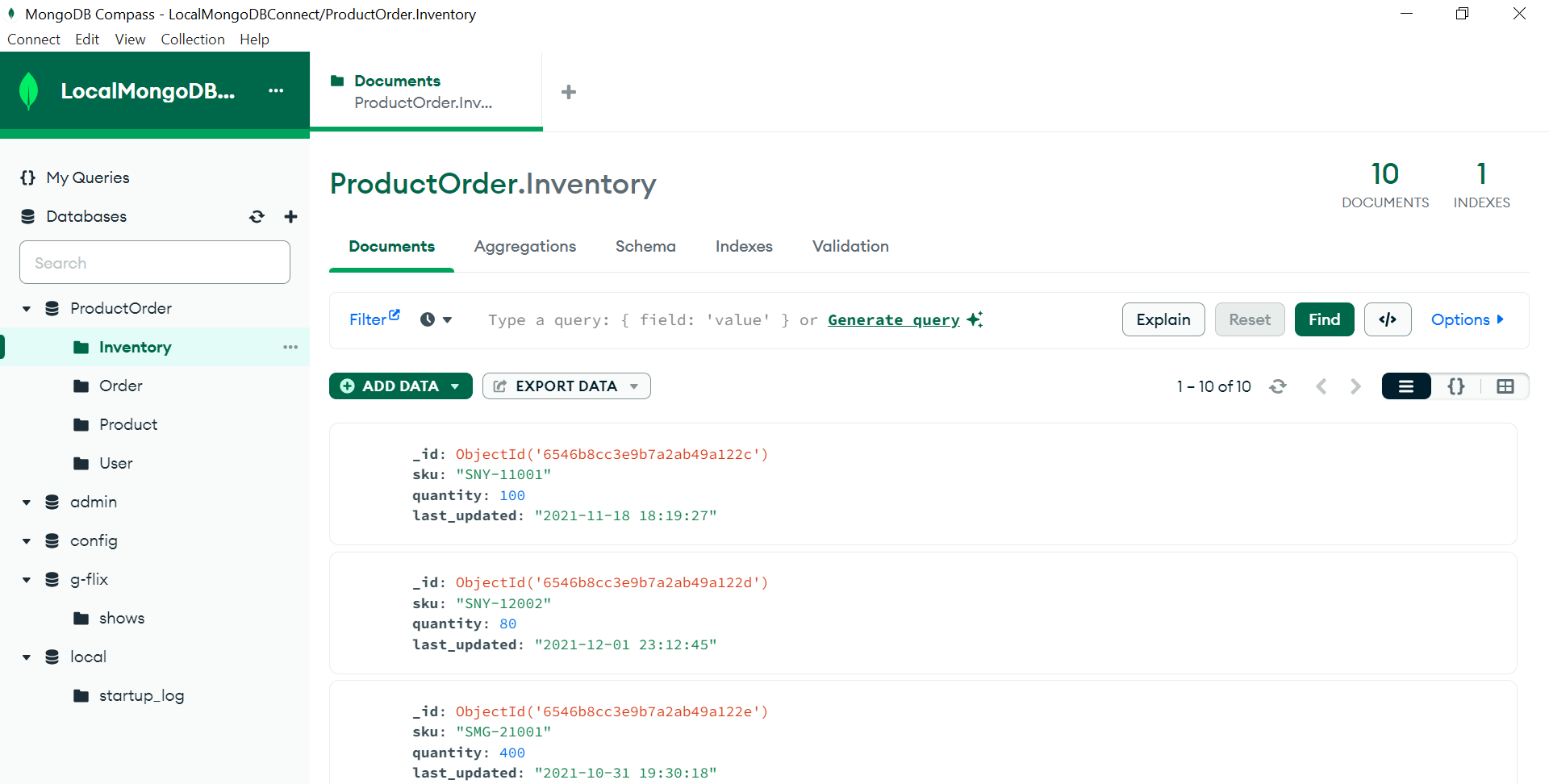


3. 3. Open MongoDBCompass and navigate to the "ProductOrder" database. (2)

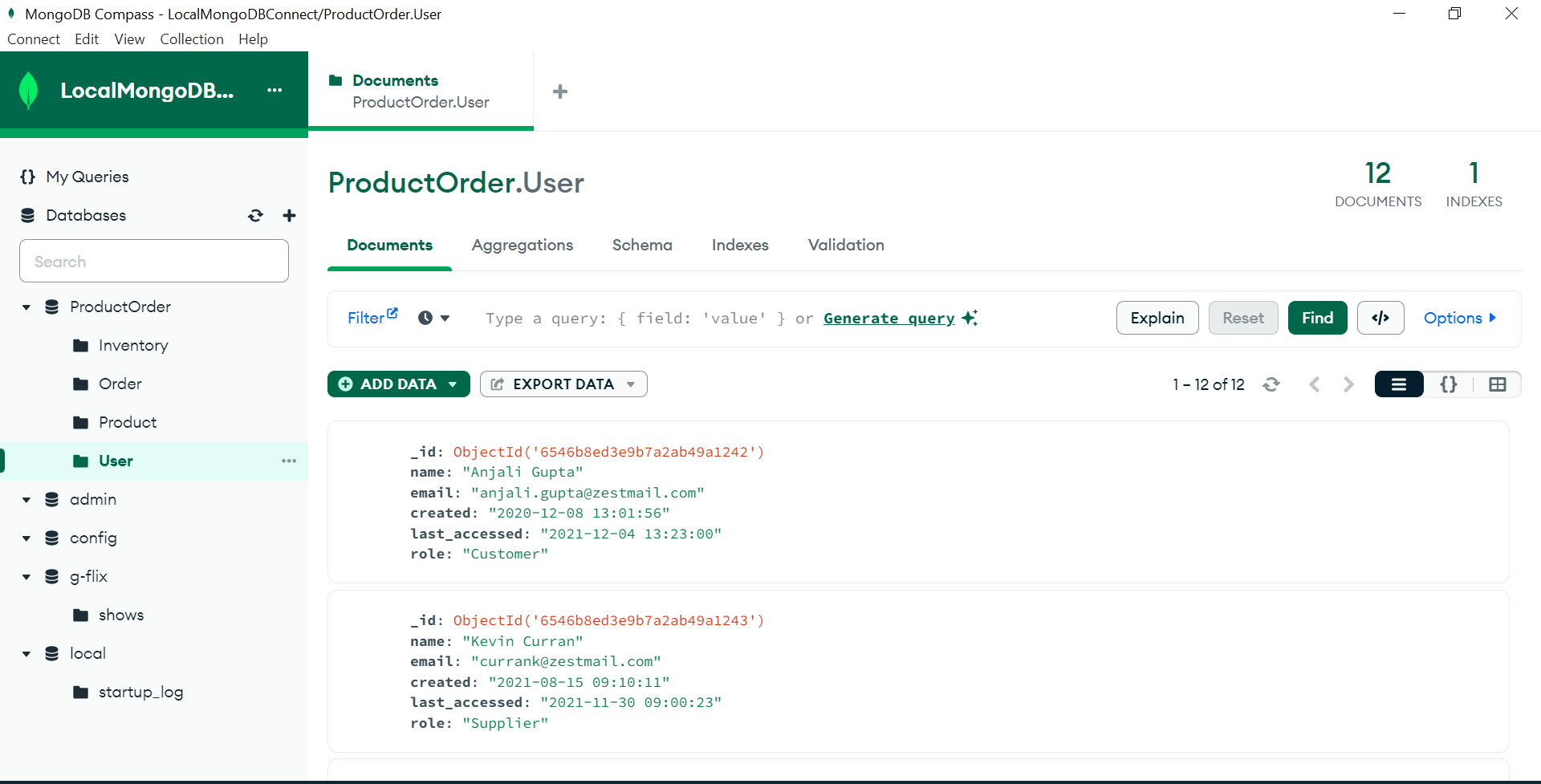
i) Add "Product\_info.json" file into the "Product" collection.



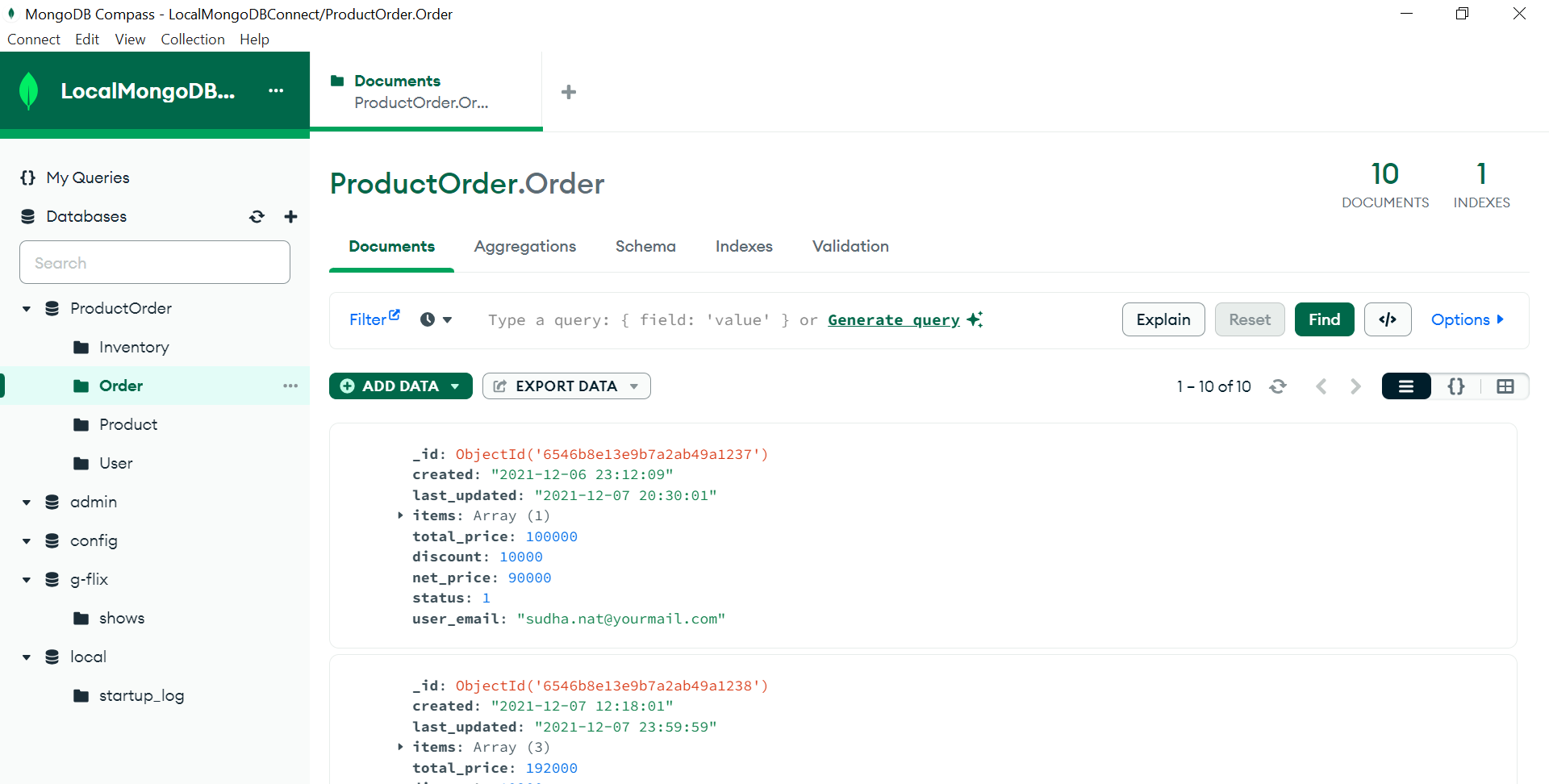
ii) Add "Inventory\_info.json" file into "Inventory" collection.



iii) Add "User\_info.json" file into the "User" collection.



iv) Add "Order\_info.json" file into "Order" collection.



4. Display the first 5 rows of product, inventory, user, and order collection.

// [1] display the first 5 rows of Product,Inventory,Order and the User collection

use ('ProductOrder');

db.Product.find().limit(5);

db.Inventory.find().limit(5);

db.Order.find().limit(5);

db.User.find().limit(5);

print()

5. Display the Unique Brand and Model names from the Product collection.

//Display the Unique Brand and Model names from the Product collection

use ('ProductOrder');

db.product.distinct("brand")

db.product.distinct("model")

6. Find the maximum and minimum price of the given products.

db.product.aggregate([{ $group: { \_id: null, maxPrice: { $max: "$price" } } }])

db.product.aggregate([{ $group: { \_id: null, minPrice: { $min: "$price" } } }])

7.Display the quantity and last\_updated date and time for sku "SNY-11001".

//Display the quantity and last\_updated date and time for sku "SNY-11001".

db.inventory.find(

    { "sku": "SNY-11001" },

    { "quantity": 1, "last\_updated": 1, "\_id": 0 }

  )

8. List down the count of the total number of users whose role is identified as 'Supplier' from User collection

//List down the count of the total number of users whose role is identified as 'Supplier' from User collection

db.User.countDocuments({ "role": "Supplier" }) or

db.User.count({ "role": "Supplier" })

9. Display 'sku', 'code', 'price', 'brand' and 'warranty' information for the model 'Bravia-X'

db.product.find(

    { "model": "Bravia-X" },

    { "sku": 1, "code": 1, "price": 1, "brand": 1, "warranty": 1, "\_id": 0 }

  )

10.Find all the information of Sony products which have an Price greater than 1 lakh

db.product.find({

    "brand": "Sony",

    "price": { "$gt": 100000 }

  })

11. Find the total no of products by each Brand and sort them in descending order.

db.product.aggregate([

    {

      $group: {

        \_id: "$brand", // Group by the "brand" field

        totalProducts: { $sum: 1 } // Count the number of products for each brand

      }

    },

    {

      $sort: {

        totalProducts: -1 // Sort the results by "totalProducts" in descending order

      }

    }

  ])

12. Find the total no of users by each role, sort them is descending order and save the results in the temporary collection

db.User.aggregate([

    {

      $group: {

        \_id: "$role", // Group by the "role" field

        totalUsers: { $sum: 1 } // Count the number of users in each group

      }

    },

    {

      $sort: {

        totalUsers: -1 // Sort the groups by "totalUsers" in descending order

      }