Question 2

Answer the following questions in MongoDB using your model from question 1.

Modifying data

Use MongoDB CRUD methods to execute the following scenarios. If nothing else is stated, assume you know the object ids of the objects involved.

1. **Sell a book to a customer.**

```
db.orders.insertOne({
        _id: 1,
        date: 2020/01/15,
        order_state: “ordered”,
        book: [{
                ISBN: NumberInt(A1),
                price: NumberDecimal(“35.50”),
                quantity: NumberInt(1)
        }]
})
db.customer.updateOne(
{$and: [{“name”: “Markus”}, {order_id: []}]},
{$addToSet: {_id: 1}}
)
```

2. **Change the address of a customer.**

```
db.customer.updateOne(
{$and: [{“name”: “Markus”}, {“address”: “71 Fakegade”}]},
{$set: {“address”: “98 Fakevej”}}
)
```

3. **Add an existing author to a book.**

```
db.book.updateOne(
{{author_id: []}},
{$addToSet: {_id: ObjectId("5e906db8a76d96ee3479d975")}}
)
```

4. **Retire the "Space Opera" category and assign all books from that category to the parent category. Don't assume you know the id of the parent category.**

```
db.book.updateMany(
{{category_id: [ObjectId("5e906fdda76d96ee3479d97f")]}},
{$set: {name: “sciencefiction”)}}
)

db.category.deleteOne(
{name: “spaceopera”}
)
```

5. **Change a book from Non-fiction to fiction, or vice versa.**

```
db.book.updateOne(
{$and: [{title: “Sit Aret”}, {category_id: [ObjectId("5e906fdda76d96ee3479d979")]}]},
{$set: {category_id: [ObjectId("5e906fdda76d96ee3479d97a")]}}
)
```

6. **Retire the "Fantasy and Science Fiction" category and just use either "Fantasy" or "Science Fiction"**

```
db.book.updateMany(
{category_id: [ObjectId("5e906fdda76d96ee3479d97d")]},
{$set: {category_id: [ObjectId("5e906fdda76d96ee3479d97c")]}}
)
db.category.deleteOne(
{name: “sciencefictionandfantasy”}
)
```

7. **Sell 3 copies of one book and 2 of another in a single order**

```
db.orders.createMany({
        _id: 2,
        date: 2020/02/11,
        order_state: “ordered”,
        book: [{
                ISBN: “A1”,
                price: NumberDecimal(“35.50”),
                quantity: NumberInt(3)
        },
        {
                ISBN: “B2”,
                price: NumberDecimal(“42.00”),
                quantity: NumberInt(2)
        }]
});

db.customer.updateOne(
{$and: [{“name”: “Amy”}, {order_id: []}]},
{$addToSet: {_id: 2}}
)
```

Querying data

Write find(), aggregate(), or mapreduce() statements to return the following data

1. **All books by an author**

db.author.find({name: "Luka"})
db.book.find(author_id: [ObjectId of Luka]).pretty()

2. **Total price of an order**

db.order.find({_id: 1})

3. **Total sales to a customer**

db.customer.find({order_id: []}).size()

4. **Books that are categorized as neither fiction nor non-fiction**

db.book.find({category_id: {$nin: [objectId of fiction, objectId of non-fiction]} })

5. **Average page count by genre**

db.book.aggregate([{$group: {category_id: [ObjectId of category], pageAverage: {$avg: "$pages"} } }])

6. **Categories that have no sub-categories**

db.category.find({}, {parent: "null"})

7. **ISBN numbers of books with more than one author**

db.book.find({author_id: {$gt: 1}})

8. **ISBN numbers of books that sold at least X copies (you decide the value for X)**

db.order.aggregate([{$unwind: "$book"}])
db.order.aggregate([{ $group: {_id: "$ISBN" , total: {$sum: "$quantity"}} }, {$match: {"total": {$gte: 3} }}
])

9. **Number of copies of each book sold – unsold books should show as 0 sold copies.**

db.book.find({ISBN})
db.order.aggregate([{$unwind: "$book"}])

10. **Best-selling books: The top 10 selling books ordered in descending order by number of sales.**

db.order.aggregate([{$unwind: "$book"}])
var descending = db.order.aggregate([{ $group: {_id: "$ISBN" , total: {$sum: "$quantity"}} }])
db.order.find()._descending("$orderby", {_id: -1})

11. **Best-selling genres: The top 3 selling genres ordered in descending order by number of sales.**

```
db.order.aggregate([{$unwind: “$book”}])
collect all isbns into a set
db.book.aggregate([{ $group: {_id: “$ISBN” , category: {“$category_id”}} }])
db.order.find()._descending(“$orderby”, {_id: -1})
```

12. **All science fiction books (Hint: Google "Recursive with")**

```
db.book.aggregate( [
  {$graphLookup: {
       from: "book",
       startWith: "$category_id",
       connectFromField: "category_id"
       connectToField: "ISBN",
       as: "SciFiBooks"}
  }] )
```

13. **Characters used in science fiction books**

```
db.category.aggregate([{$match: {parent: “sciencefiction”}, {name: “sciencefiction”}}])
```

14. **Number of books in each category**

```
var categoryList = db.book.aggregate([{$unwind: “$category_id”}])
db.book.aggregate([{ $group: {_id: null, category: {$addToSet: “$category_id”}}
```