NSQ1 S23 Course Assignment 2

# Question 1 – Revise model

If necessary, revise the model from course assignment 1. Otherwise, use the same model.

# Question 2 – Model database

Design a MongoDB model for the bookstore model from the 1st course assignment. Make note of the choices you make and why. You can document the model using example documents.

Design the model to be used with the queries from the 1st course assignment.

# Question 3 – Create schemas

Design and create schemas for the collections in your model.

# Question 4 – Work with data

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

## Modifying data

Use MongoDB CRUD methods to execute the following scenarios. Use transactions where necessary.

If nothing else is stated, assume you know the object ids of the objects involved.

1. Sell a book to a customer.
2. Change the address of a customer.
3. Add an existing author to a book.
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.
5. Sell 3 copies of one book and 2 of another in a single order

## Querying data

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

1. All books by an author
2. Total price of an order
3. Total sales (in £) to a customer
4. Books that are categorized as neither fiction nor non-fiction
5. Average page count by genre
6. Categories that have no sub-categories
7. ISBN numbers of books with more than one author
8. ISBN numbers of books that sold at least X copies (you decide the value for X)
9. Number of copies of each book sold – unsold books should show as 0 sold copies.
10. Best-selling books: The top 10 selling books ordered in descending order by number of sales.
11. Best-selling genres: The top 3 selling genres ordered in descending order by number of sales.
12. All science fiction books. Note: Books in science fiction subcategories like cyberpunk also count as science fiction. Don’t use your knowledge of the concrete category structure. (Hint: $graphLookup)
13. Characters used in science fiction books. Note from (12) applies here as well.
14. For each category: Number of books in the category including books in its subcategories.

# Question 5 – Cloud

1. Create a cluster and a database in MongoDB Atlas. Create a collection for each of the collections in Question 2.
2. Create a new collection, orderLog, and a trigger on creation of orders. The trigger should add objects to the orderLog whenever an order is inserted or updated.
   1. When inserted: The object should have a single property, “inserted”, with the order object as a value.
   2. When updated: The object should have an “updated” property with the new order, and an “updateDescription” property with the [update description](https://www.mongodb.com/docs/manual/reference/change-events/update/).
3. In MongoDB Realm set up the schemas from question 3.
4. Run the following Queries in GraphQL
   1. Data of a book including authors and categoriesGraphical user interface, text, application

      Description automatically generated
   2. All books by an authorGraphical user interface, text, application

      Description automatically generated
   3. The details of an order Graphical user interface, text, application

      Description automatically generated

Document this with

1. The code of the trigger function
2. The GraphQL Schema
3. The three GraphQL Queries

# Question 6 – Report

Write a report on the experience gained by completing Question 1 and 2 above. The report should contain answers to the questions

* What were the decisions taken in the modelling?
* Why were these decisions taken?
* What were the consequences of these decisions?
* What were the difficult and easy parts of the exercise?
* How does that compare to relational databases?
* What are the advantages and disadvantages of MongoDB over relational databases for this exercise?

# Rules

* Make the exercise in groups of 2 – 4
* Hand in to itslearning no later than 26 March

Assignment 2, Handin 2, Report 2

1. What were the decisions taken in the modelling?

The decisions taken in consideration were the webpage of amazon.co.uk and the different querying & modifying requirements from Question 3 & Question 4. Feedback from assignment 1 was considered and ER Diagram altered. Then the schema was first created. The schema was really important in order to model this MongoDB exercise.

1. Why were these decisions taken?

In order to satisfy Question 2 & Question 3 the schema had to be defined very precisely. Otherwise, I was running into risks of inconsistent data, faulty data or incomplete data when working with MongoDB. Decisions such as subcategories having a recursive relationship was taken from the feedback of Assignment 1 whereas decision such as customer\_number having the type of int was totally my decision ( now that I look back at it this can be a way to break the database if not proper checks are added in the backend or frontend of an application that would use this model with customer\_number having int type )

1. What were the consequences of these decisions?

Consequences can be leading down rabbit holes on later development stages if poor modelling decisions are made. The reviewer of this assignment should also take in consideration I am not entirely taking in consideration some potential flaws because I am not developing a fullstack application from this database here. As mentioned in Answer 2 I could have changed the address\_number if I am seeing a potential data inconsistency from example if a future user might enter a number starting with 0 in the frontend of potentially implemented full-stack system on top of this MongoDB database. But as it is of right now – even with a couple of small flaws, I had looked and fixed many potentially bigger modelling flaws of this assignment.

1. What were the difficult and easy parts of the exercise?

First 3 parts of this assignment were easy or sort to say – definitely easier than question 4 where modifying and querying had to be done. Reason is – working with aggregate and aggregation pipeline was not a piece of cake. I also had some banging against the wall when it came to the Question 5 GraphQL query implementation as I was not at all fond of GraphQL query language. To be honest even the data insertion was a troublesome experience because not all of the inserted Ids were stored in the session until I realized I did not have a session and an open transaction. That being said, only the editing of my ER and the defining of my schema were a breeze, other 3 parts of this assignment definitely took some commits to do (as of right now I have 42 commits made in Github for parts 2-4)

1. How does that compare to relational databases?

It does not. Or at least definitely rarely touching ends. Defining schema was something I have previously worked on with mongoose but never with jsonSchema and validator. That and of course sharing the same ER diagram is comparably close to assignment 1. Inserting data, modifying data, querying data & all this fancy triggers, graphQL schemas and queries, in my opinion, are by far more intuitive, easy to work with and even sort to say – interesting for me to work. Hence the reason why I signed this course even though I do not really like Data Modelling. I just think that MongoDb, GraphQL are a must in the kit of a web developer and not that much SQL DBs

1. What are the advantages and disadvantages of MongoDB over relational databases for this exercise?

Could not say I find MongoDB disadvantages in any way possible. Not having a schema in ways even let’s a user/develop have more freedom on their hands. Of course, that leads to as said in Answer 1 – data inconsistency and hence why people came with the schema validators, third parties such as mongoose and etc. Once that schema is set and connected in my opinion MongoDB is a really powerful DB and I cannot, as of right now, think of any disadvantages. That conclusion, of course, can be lead by the fact I am not a fan of SQL databases and by the way MongoDB have everything for different users – from the MongoDB Atlas (web browser based) to MongoDB Compass (application) to MongoDB shell for the more experienced.  
  
Github repo link for everything else: (ER Diagram, Answers to questions 2-4, GraphQL Schema):  
<https://github.com/NoHop3/via-sem6-nsq-assignment2>