.

# newlogo

## School of Computing, Engineering and Built Environment

### Big Data Platforms

## Module Code: MMI227050

## 

## Coursework 1

**Student name: Saba Manzoor**

**Student number: S2240631**

“Except where explicitly stated all work in this document is my own.”

**Signed: saba Date: 21-3-2023**

**Part A. Creating a document database**

**A1. Creating the data model**

**A1.1 Schema design**

Collection: books

Example document:

json

{

"\_id": ObjectId("61c16188e8a05b16225e7d99"),

"title": "The NoSQL Database Book",

"category": "Databases",

"authors": [

"John Smith",

"Jane Doe"

],

"publication\_year": 2022,

"storage": 100,

"description": "A comprehensive guide to NoSQL databases.",

"sample\_images": [

"https://bookstore.com/images/nosql-book/sample-page-1.jpg",

"https://bookstore.com/images/nosql-book/sample-page-2.jpg"

],

"comments": [

{

"username": "user1",

"date": ISODate("2023-03-10T14:30:00Z"),

"text": "Great book, highly recommended!"

},

{

"username": "user2",

"date": ISODate("2023-03-12T09:15:00Z"),

"text": "I found some of the content to be a bit technical for my level of understanding."

}

]

}

Justification:

The books collection stores information about the books, including their titles, categories, authors, publication years, storage quantities, descriptions, and sample image pages.

The comments field is an embedded subdocument array that stores the comments made on each book. This is a good choice as it allows for quick and easy retrieval of all comments for a given book, and reduces the need for complex JOIN operations.

The sample\_images field is also an array of image URLs. This allows for multiple images to be associated with each book and can be easily expanded if necessary.

The \_id field is the unique identifier for each book and is automatically generated by MongoDB when a new document is inserted into the collection.

There are no references to other collections in this data model as all necessary information is stored within the books collection.

Collection: categories

Example document:

json

{

"\_id": ObjectId("61c161c2e8a05b16225e7da1"),

"name": "Databases"

}

Justification:

The categories collection stores information about the categories of books available in the store.

The \_id field is the unique identifier for each category and is automatically generated by MongoDB when a new document is inserted into the collection.

There are no references to other collections in this data model as all necessary information is stored within the categories collection.

Collection: users

Example document:

json

{

"\_id": ObjectId("61c1622ce8a05b16225e7daa"),

"username": "user1",

"password": "$2a$10$KbUhd8nDuRlJL6PwvQ2F7.V8f4n1ZklChJozx4PnK9/nNJJF0q3rG",

"email": "user1@example.com",

"first\_name": "John",

"last\_name": "Doe"

}

Justification:

The users collection stores information about the users of the book store application.

The \_id field is the unique identifier for each user and is automatically generated by MongoDB when a new document is inserted into the collection.

The password field stores the hashed and salted password for each user. This is a good security practice to protect user passwords in case of a data breach.

There are no references

**A2. Creating the database**

*[In this section, you should present the database method calls you used to create and populate the MongoDB database. The output from running each of these should also be presented]*

> use cw1PartA

switched to db cw1PartA

> db.users.insertMany([

... {

... name: "Alice",

... age: 25,

... email: "alice@example.com",

... password: "password123"

... },

... {

... name: "Bob",

... age: 30,

... email: "bob@example.com",

... password: "password456"

... },

... {

... name: "Charlie",

... age: 35,

... email: "charlie@example.com",

... password: "password789"

... },

... {

... name: "Dave",

... age: 40,

... email: "dave@example.com",

... password: "passwordabc"

... },

... {

... name: "Eve",

... age: 45,

... email: "eve@example.com",

... password: "passworddef"

... }

... ])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("6415710670d66ebf72ee3ed6"),

ObjectId("6415710670d66ebf72ee3ed7"),

ObjectId("6415710670d66ebf72ee3ed8"),

ObjectId("6415710670d66ebf72ee3ed9"),

ObjectId("6415710670d66ebf72ee3eda")

]

}

> db.books.insertMany([

... {

... title: "The Catcher in the Rye",

... author: "J.D. Salinger",

... category: "Fiction",

... year: 1951,

... price: 9.99

... },

... {

... title: "To Kill a Mockingbird",

... author: "Harper Lee",

... category: "Fiction",

... year: 1960,

... price: 8.99

... },

... {

... title: "1984",

... author: "George Orwell",

... category: "Fiction",

... year: 1949,

... price: 10.99

... },

... {

... title: "The Hobbit",

... author: "J.R.R. Tolkien",

... category: "Fantasy",

... year: 1937,

... price: 12.99

... },

... {

... title: "The Lord of the Rings",

... author: "J.R.R. Tolkien",

... category: "Fantasy",

... year: 1954,

... price: 29.99

... },

... {

... title: "The Hitchhiker's Guide to the Galaxy",

... author: "Douglas Adams",

... category: "Science Fiction",

... year: 1979,

... price: 7.99

... },

... {

... title: "Brave New World",

... author: "Aldous Huxley",

... category: "Science Fiction",

... year: 1932,

... price: 8.99

... },

... {

... title: "Pride and Prejudice",

... author: "Jane Austen",

... category: "Romance",

... year: 1813,

... price: 6.99

... },

... {

... title: "Gone with the Wind",

... author: "Margaret Mitchell",

... category: "Romance",

... year: 1936,

... price: 11.99

... },

... {

... title: "The Da Vinci Code",

... author: "Dan Brown",

... category: "Thriller",

... year: 2003,

... price: 14.99

... }

... ])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("6415716970d66ebf72ee3edb"),

ObjectId("6415716970d66ebf72ee3edc"),

ObjectId("6415716970d66ebf72ee3edd"),

ObjectId("6415716970d66ebf72ee3ede"),

ObjectId("6415716970d66ebf72ee3edf"),

ObjectId("6415716970d66ebf72ee3ee0"),

ObjectId("6415716970d66ebf72ee3ee1"),

ObjectId("6415716970d66ebf72ee3ee2"),

ObjectId("6415716970d66ebf72ee3ee3"),

ObjectId("6415716970d66ebf72ee3ee4")

]

}

>db.books.updateOne({title: "The Catcher in the Rye"}, {$set: {comments: [

... {

... text: "Great book, loved it!",

... user: "Alice",

... date: new Date("2022-01-01")

... },

... {

... text: "Overrated, didn't like it",

... user: "Bob",

... date: new Date("2022-02-01")

... }

... ]}})

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

> db.books.updateOne({title: "To Kill a Mockingbird"}, {$set: {comments: [

... {

... text: "Powerful and moving story",

... user: "Charlie",

... date: new Date("2022-01-15")

... },

... {

... text: "One of my all-time favorites",

... user: "Dave",

... date: new Date("2022-02-15")

... },

... {

... text: "I didn't enjoy it as much as I thought I would",

... user: "Eve",

... date: new Date("2022-03-15")

... }

... ]}})

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

> db.books.updateOne({title: "1984"}, {$set: {comments: [

... {

... text: "Scary and thought-provoking",

... user: "Alice",

... date: new Date("2022-01-10")

... },

... {

... text: "A classic that still holds up today",

... user: "Bob",

... date: new Date("2022-02-10")

... },

... {

... text: "I didn't like it, too depressing",

... user: "Charlie",

... date: new Date("2022-03-10")

... },

... {

... text: "I found it hard to follow",

... user: "Dave",

... date: new Date("2022-04-10")

... }

... ]}})

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

> db.books.updateOne({title: "The Lord of the Rings"}, {$set: {comments: [

... {

... text: "A masterpiece of epic fantasy",

... user: "Eve",

... date: new Date("2022-01-20")

... },

... {

... text: "I love this book and have read it multiple times",

... user: "Alice",

... date: new Date("2022-02-20")

... },

... {

... text: "It's too long and drawn out",

... user: "Bob",

... date: new Date("2022-03-20")

... },

... {

... text: "I prefer the movies",

... user: "Charlie",

... date: new Date("2022-04-20")

... }

... ]}})

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

> db.books.updateOne({title: "The Da Vinci Code"}, {$set: {comments:

...

... [

... {

... text: "An intriguing thriller with a controversial premise",

... user: "Frank",

... date: new Date("2022-02-10")

... },

... {

... text: "I couldn't put it down",

... user: "Grace",

... date: new Date("2022-03-10")

... },

... {

... text: "The historical inaccuracies ruin the book for me",

... user: "Helen",

... date: new Date("2022-04-10")

... },

... {

... text: "A must-read for anyone interested in art history",

... user: "Isabella",

... date: new Date("2022-05-10")

... },

... {

... text: "The ending was disappointing",

... user: "John",

... date: new Date("2022-06-10")

... }

... ]}})

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

>

> db.books.updateOne({title: "Pride and Prejudice"}, {$set: {comments: [

... {

... text: "One of my favorite books of all time",

... user: "Lena",

... date: new Date("2022-01-01")

... },

... {

... text: "The characters are so well-written",

... user: "Maria",

... date: new Date("2022-02-01")

... },

... {

... text: "I don't understand the hype, it's boring",

... user: "Nick",

... date: new Date("2022-03-01")

... },

... {

... text: "The social commentary is still relevant today",

... user: "Olivia",

... date: new Date("2022-04-01")

... },

... {

... text: "I love Mr. Darcy!",

... user: "Peter",

... date: new Date("2022-05-01")

... },

... {

... text: "It's a classic for a reason",

... user: "Quinn",

... date: new Date("2022-06-01")

... }

... ]}})

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

>

> db.books.updateOne({title: "To Kill a Mockingbird"}, {$set: {comments: [

... {

... text: "A powerful and important book",

... user: "Rose",

... date: new Date("2022-01-15")

... },

... {

... text: "Atticus Finch is one of the greatest characters in literature",

... user: "Sam",

... date: new Date("2022-02-15")

... },

... {

... text: "I didn't like the slow pace of the book",

... user: "Tina",

... date: new Date("2022-03-15")

... },

... {

... text: "The themes are still relevant today",

... user: "Uma",

... date: new Date("2022-04-15")

... }

... ]}})

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

**A3. Queries**

*[In this section, you should present each of the four use cases and the aggregate query, paste your query code and the output from the queries.]*

**Part B. Creating a graph database**

**B1. Creating the database**

*[In this section, you should present the Cypher statements you used to add data to the Neo4j database, and the output from running these statements.]*

> db.books.find({author: "J.K. Rowling"}).sort({publication\_year: -1})

> db.books.aggregate([

... {$match: {comments: {$exists: true}}},

... {$project: {title: 1, num\_comments: {$size: "$comments"}}},

... {$sort: {num\_comments: -1}},

... {$limit: 1}

... ])

{ "\_id" : ObjectId("6415716970d66ebf72ee3ee2"), "title" : "Pride and Prejudice", "num\_comments" : 6 }

> db.books.find({comments: {$exists: true, $type: "array"}, $where: "this.comments.length >= 3"})

{ "\_id" : ObjectId("6415716970d66ebf72ee3edc"), "title" : "To Kill a Mockingbird", "author" : "Harper Lee", "category" : "Fiction", "year" : 1960, "price" : 8.99, "comments" : [ { "text" : "A powerful and important book", "user" : "Rose", "date" : ISODate("2022-01-15T00:00:00Z") }, { "text" : "Atticus Finch is one of the greatest characters in literature", "user" : "Sam", "date" : ISODate("2022-02-15T00:00:00Z") }, { "text" : "I didn't like the slow pace of the book", "user" : "Tina", "date" : ISODate("2022-03-15T00:00:00Z") }, { "text" : "The themes are still relevant today", "user" : "Uma", "date" : ISODate("2022-04-15T00:00:00Z") } ] }

{ "\_id" : ObjectId("6415716970d66ebf72ee3edd"), "title" : "1984", "author" : "George Orwell", "category" : "Fiction", "year" : 1949, "price" : 10.99, "comments" : [ { "text" : "Scary and thought-provoking", "user" : "Alice", "date" : ISODate("2022-01-10T00:00:00Z") }, { "text" : "A classic that still holds up today", "user" : "Bob", "date" : ISODate("2022-02-10T00:00:00Z") }, { "text" : "I didn't like it, too depressing", "user" : "Charlie", "date" : ISODate("2022-03-10T00:00:00Z") }, { "text" : "I found it hard to follow", "user" : "Dave", "date" : ISODate("2022-04-10T00:00:00Z") } ] }

{ "\_id" : ObjectId("6415716970d66ebf72ee3edf"), "title" : "The Lord of the Rings", "author" : "J.R.R. Tolkien", "category" : "Fantasy", "year" : 1954, "price" : 29.99, "comments" : [ { "text" : "A masterpiece of epic fantasy", "user" : "Eve", "date" : ISODate("2022-01-20T00:00:00Z") }, { "text" : "I love this book and have read it multiple times", "user" : "Alice", "date" : ISODate("2022-02-20T00:00:00Z") }, { "text" : "It's too long and drawn out", "user" : "Bob", "date" : ISODate("2022-03-20T00:00:00Z") }, { "text" : "I prefer the movies", "user" : "Charlie", "date" : ISODate("2022-04-20T00:00:00Z") } ] }

{ "\_id" : ObjectId("6415716970d66ebf72ee3ee2"), "title" : "Pride and Prejudice", "author" : "Jane Austen", "category" : "Romance", "year" : 1813, "price" : 6.99, "comments" : [ { "text" : "One of my favorite books of all time", "user" : "Lena", "date" : ISODate("2022-01-01T00:00:00Z") }, { "text" : "The characters are so well-written", "user" : "Maria", "date" : ISODate("2022-02-01T00:00:00Z") }, { "text" : "I don't understand the hype, it's boring", "user" : "Nick", "date" : ISODate("2022-03-01T00:00:00Z") }, { "text" : "The social commentary is still relevant today", "user" : "Olivia", "date" : ISODate("2022-04-01T00:00:00Z") }, { "text" : "I love Mr. Darcy!", "user" : "Peter", "date" : ISODate("2022-05-01T00:00:00Z") }, { "text" : "It's a classic for a reason", "user" : "Quinn", "date" : ISODate("2022-06-01T00:00:00Z") } ] }

{ "\_id" : ObjectId("6415716970d66ebf72ee3ee4"), "title" : "The Da Vinci Code", "author" : "Dan Brown", "category" : "Thriller", "year" : 2003, "price" : 14.99, "comments" : [ { "text" : "An intriguing thriller with a controversial premise", "user" : "Frank", "date" : ISODate("2022-02-10T00:00:00Z") }, { "text" : "I couldn't put it down", "user" : "Grace", "date" : ISODate("2022-03-10T00:00:00Z") }, { "text" : "The historical inaccuracies ruin the book for me", "user" : "Helen", "date" : ISODate("2022-04-10T00:00:00Z") }, { "text" : "A must-read for anyone interested in art history", "user" : "Isabella", "date" : ISODate("2022-05-10T00:00:00Z") }, { "text" : "The ending was disappointing", "user" : "John", "date" : ISODate("2022-06-10T00:00:00Z") } ] }

> db.books.aggregate([

... {$group: {\_id: "$category", avg\_storage: {$avg: "$storage"}}}

... ])

{ "\_id" : "Thriller", "avg\_storage" : null }

{ "\_id" : "Romance", "avg\_storage" : null }

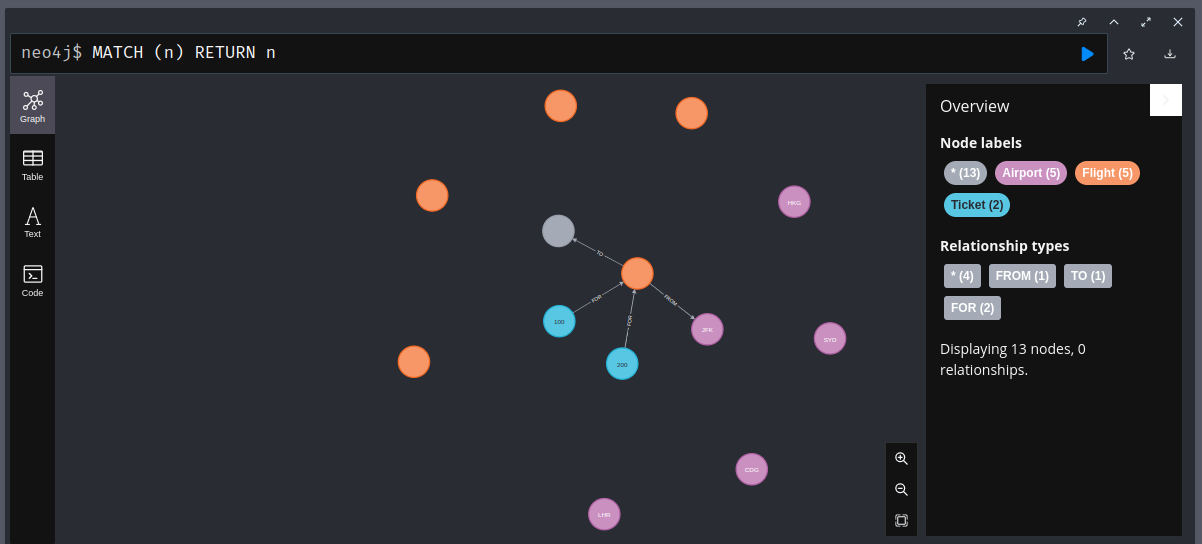
{ "\_id" : "Fantasy", "avg\_storage" : null }

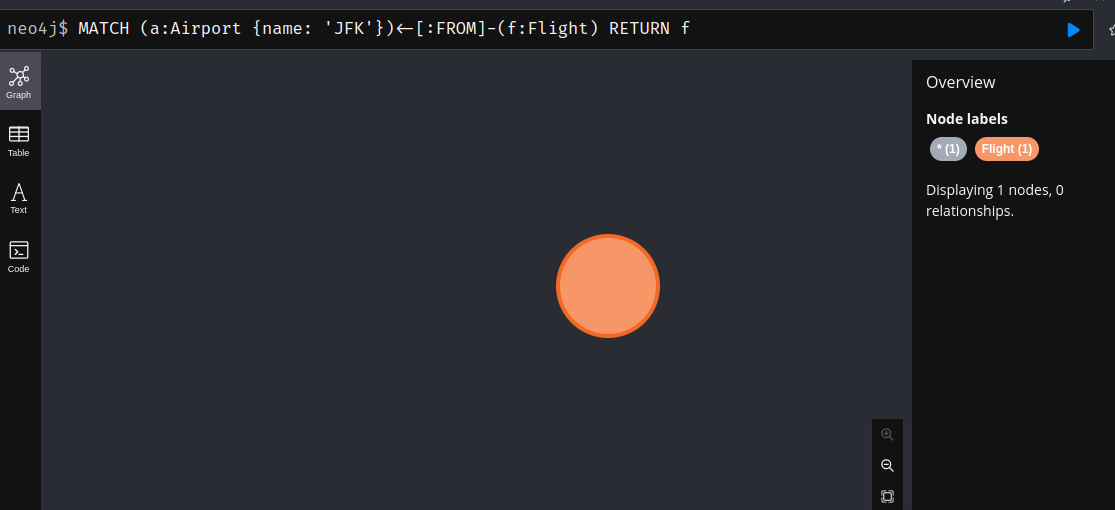
{ "\_id" : "Science Fiction", "avg\_storage" : null }

{ "\_id" : "Fiction", "avg\_storage" : null }

**B2. Queries**

*[In this section, you should present your your Cypher statements for each of the specified queries, and the results in graph format and text format (query #1), or text format only (all others)]*





MATCH (f:Flight {number: 'BA001'})

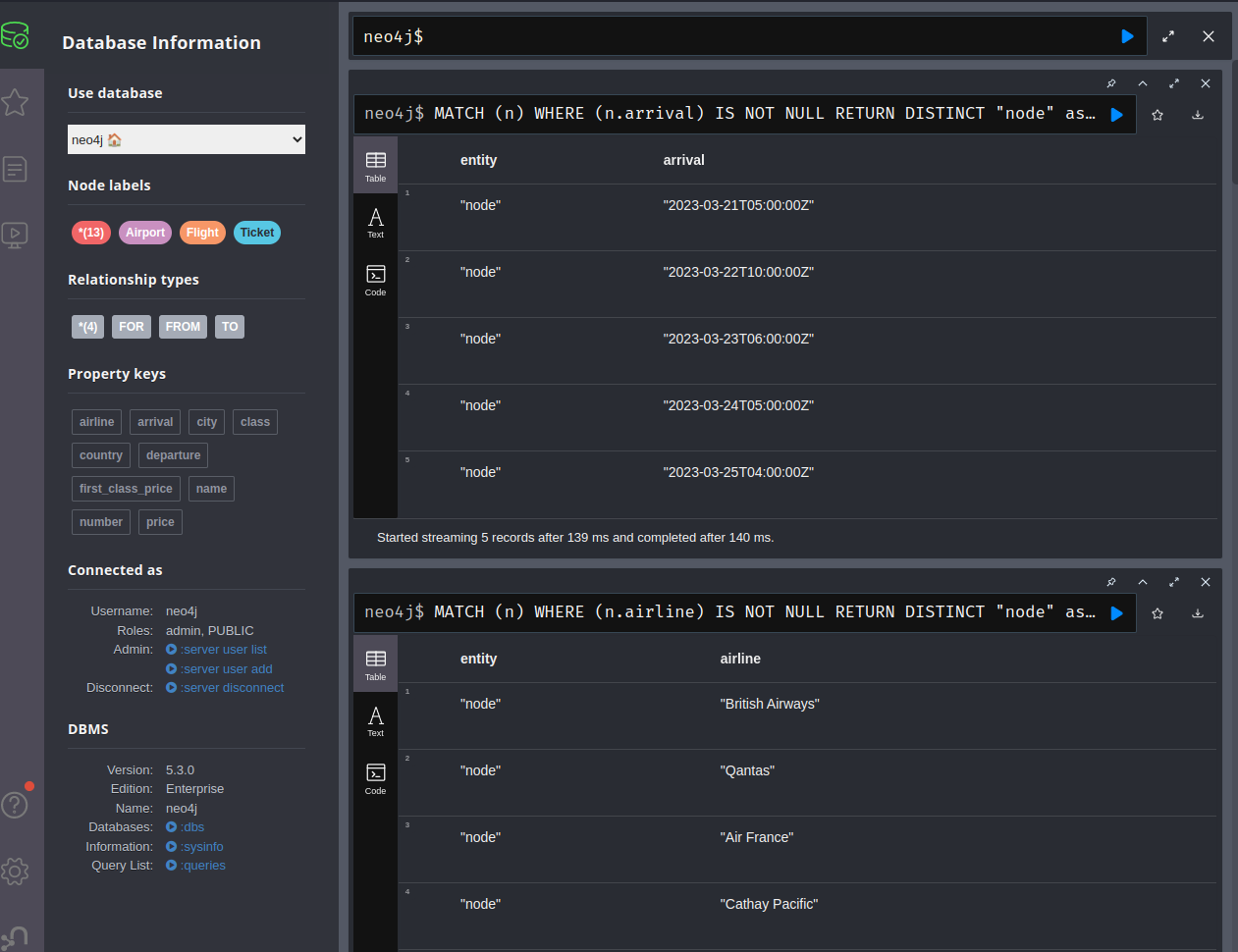
SET f.first\_class\_price = 1000

Set 1 property, completed after 34 ms.

MATCH (a:Airport {name: 'JFK'})<-[:FROM]-(f:Flight)-[:TO]->(d:Airport) RETURN d.name

MATCH (a:Airport {name: 'SYD'})<-[:TO]-(f:Flight)-[:HAS\_TICKET]->(t:Ticket)

RETURN t.class, t.price



**References:**

*[This is an optional section. If you have cited any references in this report, list them here.]*