ASSIGNMENT 03

01. Database Name: **Library**

02. Collections: **books, libraries, authors**

03. Data insertion

db.**books**.insertMany(

[{

\_id: ObjectId("60d5fc5ef295f5e2c4d5a8a3"),

title: "To Kill a Mockingbird",

library\_id: ObjectId("60d5fbb2f295f5e2c4d5a8a1"),

author\_ids: [ObjectId("60d5fd3ef295f5e2c4d5a8a4")]

},

{

\_id: ObjectId("60d5fc5ef295f5e2c4d5a8a5"),

title: "1984",

library\_id: ObjectId("60d5fbb2f295f5e2c4d5a8a2"),

author\_ids: [ObjectId("60d5fd3ef295f5e2c4d5a8a6")]

},

{

\_id: ObjectId("60d5fc5ef295f5e2c4d5a8a7"),

title: "Harry Potter and the Philosopher's Stone",

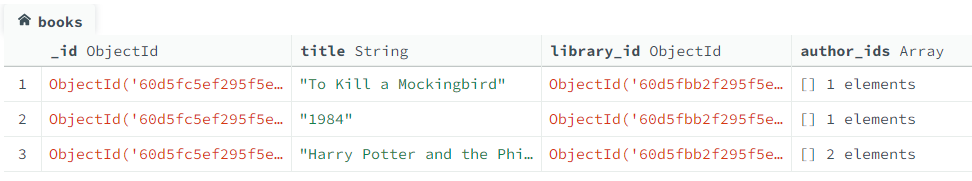
library\_id: ObjectId("60d5fbb2f295f5e2c4d5a8a1"),

author\_ids: [ObjectId("60d5fd3ef295f5e2c4d5a8a8"),

ObjectId("60d5fd3ef295f5e2c4d5a8a9")]

}

]);



db.libraries.insertMany([

{

\_id: ObjectId("60d5fbb2f295f5e2c4d5a8a1"),

name: "Central Library",

location: "Downtown"

},

{

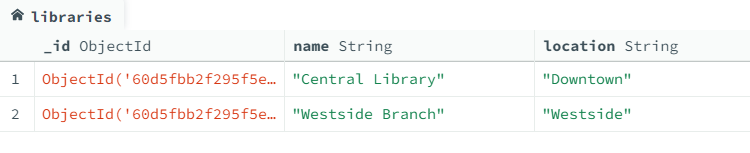
\_id: ObjectId("60d5fbb2f295f5e2c4d5a8a2"),

name: "Westside Branch",

location: "Westside"

}

]);



db.authors.insertMany([

{

\_id: ObjectId("60d5fd3ef295f5e2c4d5a8a4"),

name: "Harper Lee",

birth\_year: 1926

},

{

\_id: ObjectId("60d5fd3ef295f5e2c4d5a8a6"),

name: "George Orwell",

birth\_year: 1903

},

{

\_id: ObjectId("60d5fd3ef295f5e2c4d5a8a8"),

name: "J.K. Rowling",

birth\_year: 1965

},

{

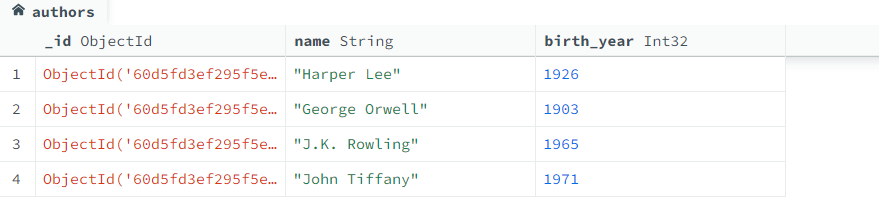
\_id: ObjectId("60d5fd3ef295f5e2c4d5a8a9"),

name: "John Tiffany",

birth\_year: 1971

}

]);



4. Find all books in the "Central Library".

db.books.aggregate([

{

$lookup: {

from: "libraries",

localField: "library\_id",

foreignField: "\_id",

as: "library"

}

},

{ $unwind: "$library" },

{ $match: { "library.name": "Central Library" } ]);

5. List all libraries along with the number of books they have.

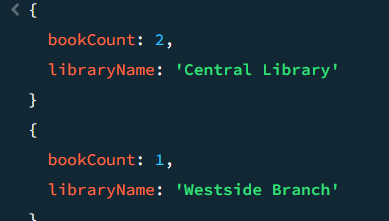
db.books.aggregate([

{

$group: {

\_id: "$library\_id",

bookCount: { $sum: 1 }

 }

},

{

$lookup: {

from: "libraries",

localField: "\_id",

foreignField: "\_id",

as: "library"

}

},

{ $unwind: "$library" },

{

$project: {

\_id: 0,

libraryName: "$library.name",

bookCount: 1

}

}

]);

6. Find all books written by "J.K. Rowling".

db.books.aggregate([

{

$lookup: {

from: "authors",

localField: "author\_ids",

foreignField: "\_id",

as: "authors"

}

},

{ $match: { "authors.name": "J.K. Rowling" } }

]);

7. List all authors along with the books they have written.



8. Find all books with their authors' details

db.books.aggregate([

{

$lookup: {

from: "authors",

localField: "author\_ids",

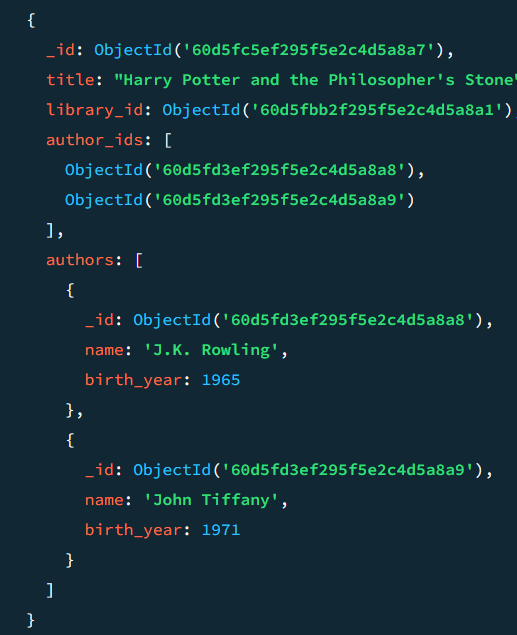
foreignField: "\_id",

as: "authors"

}

}

]);

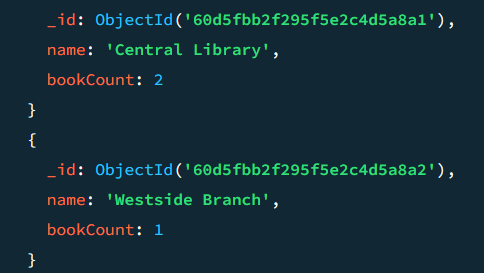


9. List all libraries with number of books (include libraries with no books)

db.libraries.aggregate([

{

$lookup: {

 from: "books",

localField: "\_id",

foreignField: "library\_id",

as: "books"

}

},

{

$project: {

name: 1,

bookCount: { $size: "$books" }

}

}

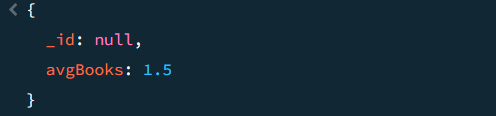
]);

10. Calculate the average number of books per library.

db.books.aggregate([

{

$group: {

 \_id: "$library\_id",

count: { $sum: 1 }

}

},

{

$group: {

\_id: null,

avgBooks: { $avg: "$count" }

}

}

]);

11. Find all authors who have written more than one book.

db.books.aggregate([

{ $unwind: "$author\_ids" },

{

$group: {

\_id: "$author\_ids",

bookCount: { $sum: 1 }

}

},

{ $match: { bookCount: { $gt: 1 } } },

{

$lookup: {

from: "authors",

localField: "\_id",

foreignField: "\_id",

as: "author"

}

},

{ $unwind: "$author" },

{

$project: {

\_id: 0,

name: "$author.name",

bookCount: 1

}

}

]);

12. Retrieve all books along with their authors' names and the library they belong to.

db.books.aggregate([

{

 $lookup: {

from: "authors",

localField: "author\_ids",

foreignField: "\_id",

as: "authors"

}

},

{

$lookup: {

from: "libraries",

localField: "library\_id",

foreignField: "\_id",

as: "library"

}

},

{ $unwind: "$library" },

{

$project: {

title: 1,

libraryName: "$library.name",

authorNames: "$authors.name"

}

}

]);

13. List all authors who have not written any books.

db.authors.aggregate([

{

$lookup: {

from: "books",

localField: "\_id",

foreignField: "author\_ids",

as: "books"

}

},

{ $match: { books: { $size: 0 } } }

]);