MONGODB CODING ASSESSMENT

UNDERSTANDING

RELATIONSHIP

LIBRARY MANAGEMENT

AUTHORS

```
[
  { "_id": 1, "name": "J.K. Rowling", "nationality": "British" },
  { "_id": 2, "name": "George R.R. Martin", "nationality": "American" },
  { "_id": 3, "name": "J.R.R. Tolkien", "nationality": "British" }
]
```

BOOKS

```
[
  { "_id": 101, "title": "Harry Potter", "author_id": 1, "genre": "Fantasy" },
  { "_id": 102, "title": "Game of Thrones", "author_id": 2, "genre": "Fantasy" },
  { "_id": 103, "title": "The Hobbit", "author_id": 3, "genre": "Adventure" }
]
```

BORROWERS

```
[
  { "_id": 1, "name": "Alice", "borrowed_books": [ { "book_id": 101, "borrowed_date": "2025-07-20" } ] },
  { "_id": 2, "name": "Bob", "borrowed_books": [ { "book_id": 102, "borrowed_date": "2025-07-22" }, { "book_id": 103, "borrowed_date": "2025-07-23" } ] }
]
```

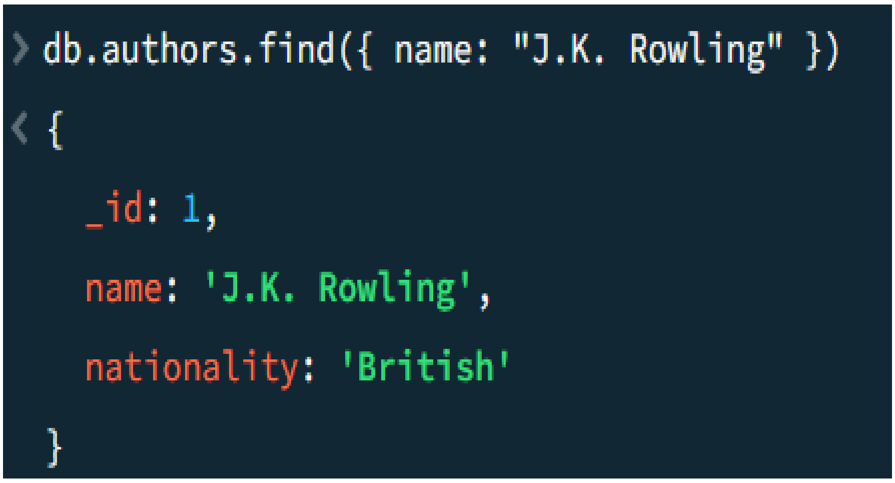
QUESTIONS:

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

Relationship: One-to-Many (authors → books)

Step 1: Find the author's _id

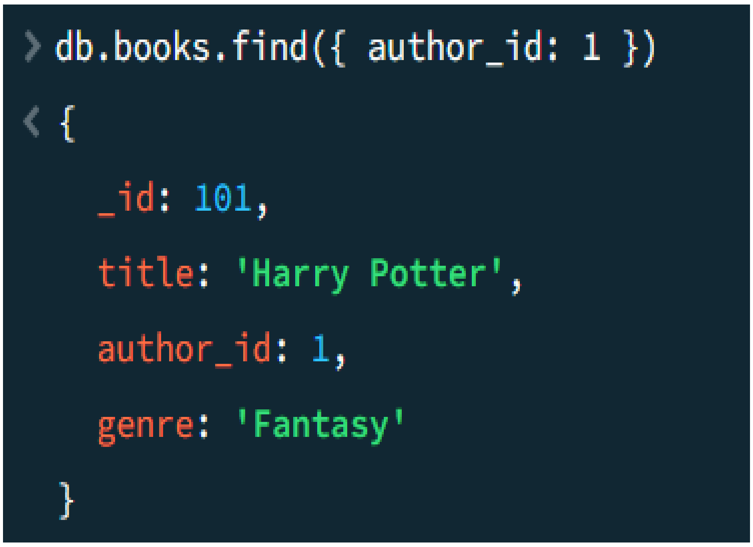
db.authors.find({ name: "J.K. Rowling" })



```
> db.authors.find({ name: "J.K. Rowling" })
< {
    _id: 1,
    name: 'J.K. Rowling',
    nationality: 'British'
  }
```

Step 2: Find all books with that author_id

db.books.find({ author_id: 1 })



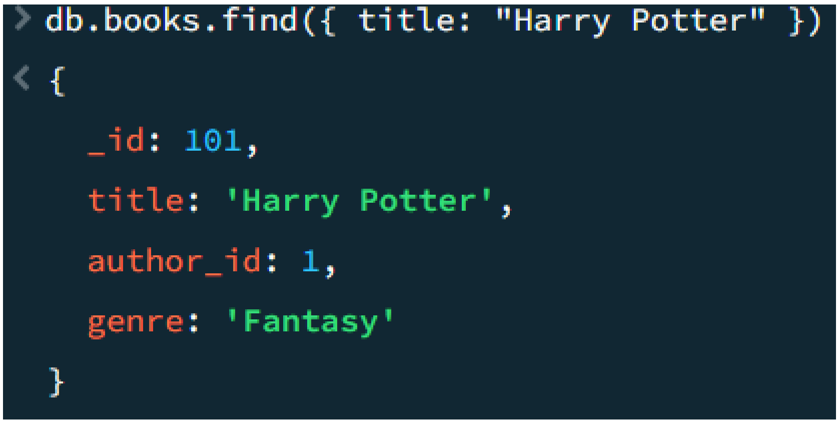
```
> db.books.find({ author_id: 1 })
< {
    _id: 101,
    title: 'Harry Potter',
    author_id: 1,
    genre: 'Fantasy'
  }
```

2. Get borrower details who borrowed 'Harry Potter'

Relationship: Many-to-Many (books ↔ borrowers)

Step 1: Find book _id for "Harry Potter"

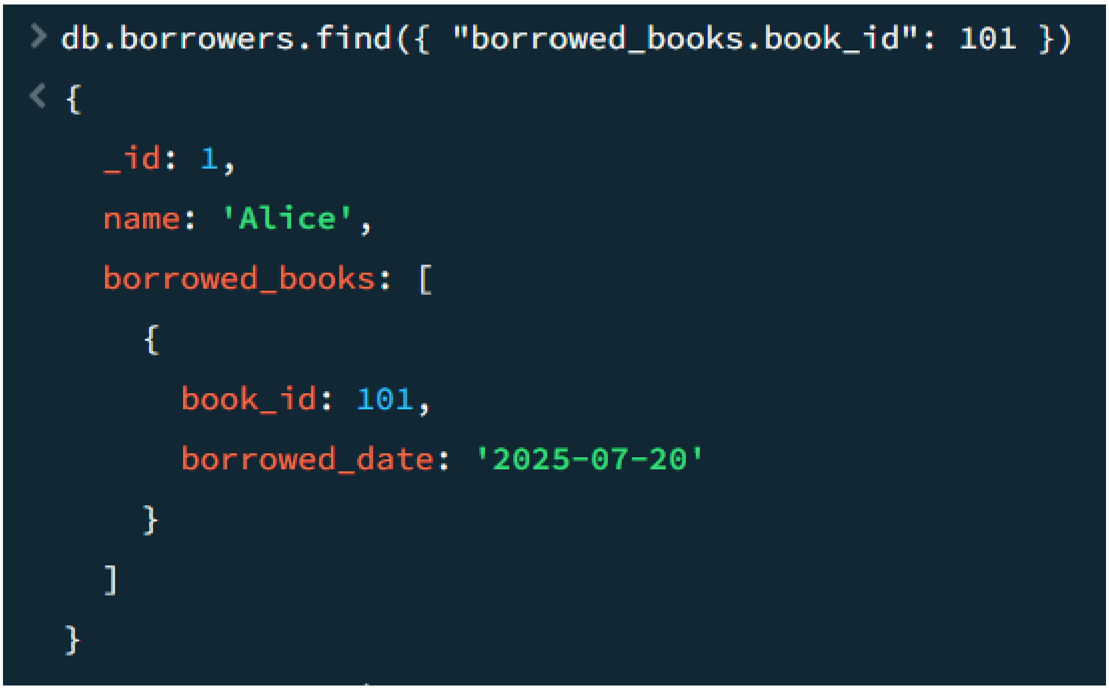
db.books.find({ title: "Harry Potter" })



```
> db.books.find({ title: "Harry Potter" })
< {
    _id: 101,
    title: 'Harry Potter',
    author_id: 1,
    genre: 'Fantasy'
  }
```

Step 2: Find borrower with that book_id in borrowed_books

db.borrowers.find({ "borrowed_books.book_id": 101 })

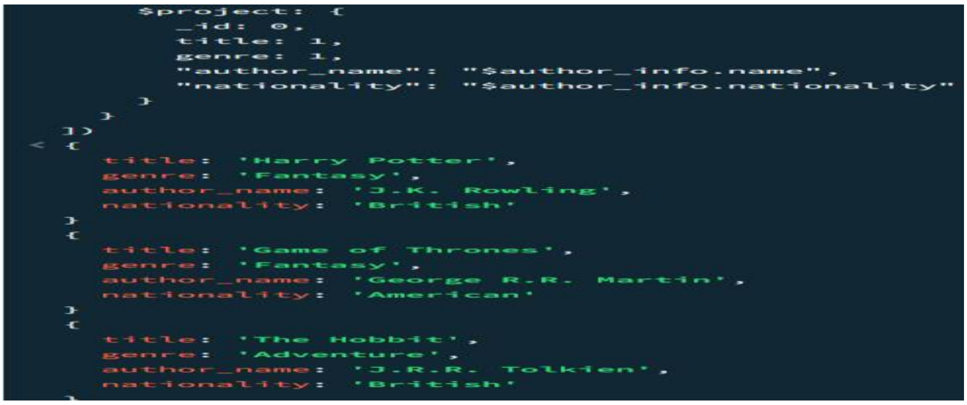


```
> db.borrowers.find({ "borrowed_books.book_id": 101 })
< {
    _id: 1,
    name: 'Alice',
    borrowed_books: [
      {
        book_id: 101,
        borrowed_date: '2025-07-20'
      }
    ]
  }
```

3. List all books and include author name using $lookup

Relationship: One-to-Many

```
db.books.aggregate([
  {
    $lookup: {
      from: "authors",
      localField: "author_id",
      foreignField: "_id",
      as: "author_info"
    }
  },
  { $unwind: "$author_info" },
  {
    $project: {
      _id: 0,
      title: 1,
      genre: 1,
      "author_name": "$author_info.name",
      "nationality": "$author_info.nationality"
    }
  }
])
```



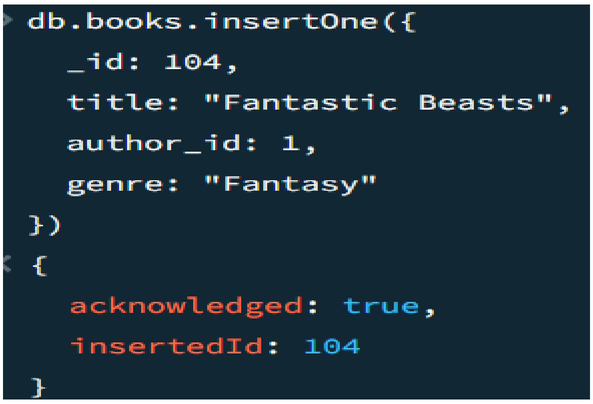
```
        $project: {
          _id: 0,
          title: 1,
          genre: 1,
          "author_name": "$author_info.name",
          "nationality": "$author_info.nationality"
        }
      }
    ])
<   {
      title: 'Harry Potter',
      genre: 'Fantasy',
      author_name: 'J.K. Rowling',
      nationality: 'British'
    }
    {
      title: 'Game of Thrones',
      genre: 'Fantasy',
      author_name: 'George R.R. Martin',
      nationality: 'American'
    }
    {
      title: 'The Hobbit',
      genre: 'Adventure',
      author_name: 'J.R.R. Tolkien',
      nationality: 'British'
```

4. Add a new book and a borrower who borrowed it

Relationship: Many-to-Many (books ↔ borrowers)

Insert a new book:

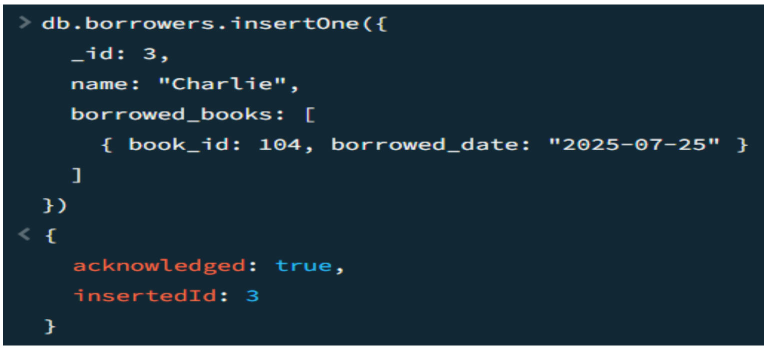
```
db.books.insertOne({
  _id: 104,
  title: "Fantastic Beasts",
  author_id: 1,
  genre: "Fantasy"
})
```



```
> db.books.insertOne({
    _id: 104,
    title: "Fantastic Beasts",
    author_id: 1,
    genre: "Fantasy"
  })
< {
    acknowledged: true,
    insertedId: 104
  }
```

Add a borrower who borrowed it:

```
db.borrowers.insertOne({
  _id: 3,
  name: "Charlie",
  borrowed_books: [
    { book_id: 104, borrowed_date: "2025-07-25" }
  ]
})
```

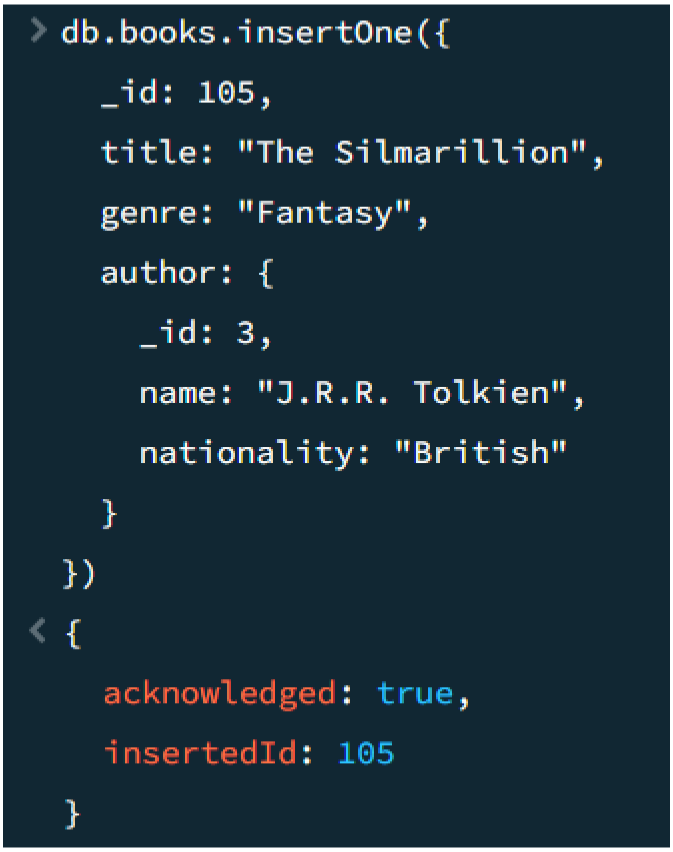


```
> db.borrowers.insertOne({
    _id: 3,
    name: "Charlie",
    borrowed_books: [
      { book_id: 104, borrowed_date: "2025-07-25" }
    ]
  })
< {
    acknowledged: true,
    insertedId: 3
  }
```

5. Embed author info inside a book document (alternate model)

Relationship: One-to-One (embedded)

```
db.books.insertOne({
  _id: 105,
  title: "The Silmarillion",
  genre: "Fantasy",
  author: {
    _id: 3,
    name: "J.R.R. Tolkien",
    nationality: "British"
  }
})
```



```
> db.books.insertOne({
    _id: 105,
    title: "The Silmarillion",
    genre: "Fantasy",
    author: {
      _id: 3,
      name: "J.R.R. Tolkien",
      nationality: "British"
    }
  })
< {
    acknowledged: true,
    insertedId: 105
  }
```