

# Final Report

## 1. Project Description

The Library Management System (LMS) is an application designed to streamline and automate various operations of a library. The system covers multiple aspects such as user management, librarian activities, branch details, book inventories, and borrowing processes. The system uses MongoDB, a NoSQL document database, providing flexibility and scalability to manage complex data relationships effectively.

The LMS equips librarians and users with comprehensive tools for seamless interaction with library resources. The main functionalities include:

### Librarian Capabilities:

- **Book Management:** Librarians can add new books, update or remove outdated book records.
- **Branch Management:** Librarians manage branches by adding new ones and updating branch information.

### User-Friendly Borrowing System:

- **Borrowing Books:** Users can borrow books, with the system logging all borrowing details, including user information, book details, borrow date, and expected return date.

### User Account Management:

- **Account Information:** Registered users can view and update their personal details and access their borrowing history.

## 2. Database Description

The Library Management System database is designed to manage the diverse and interconnected components of library operations, including books, copies, branches, transactions, users, and administrators. The schema leverages MongoDB for its flexibility, scalability, and efficient querying capabilities.

### Collections:

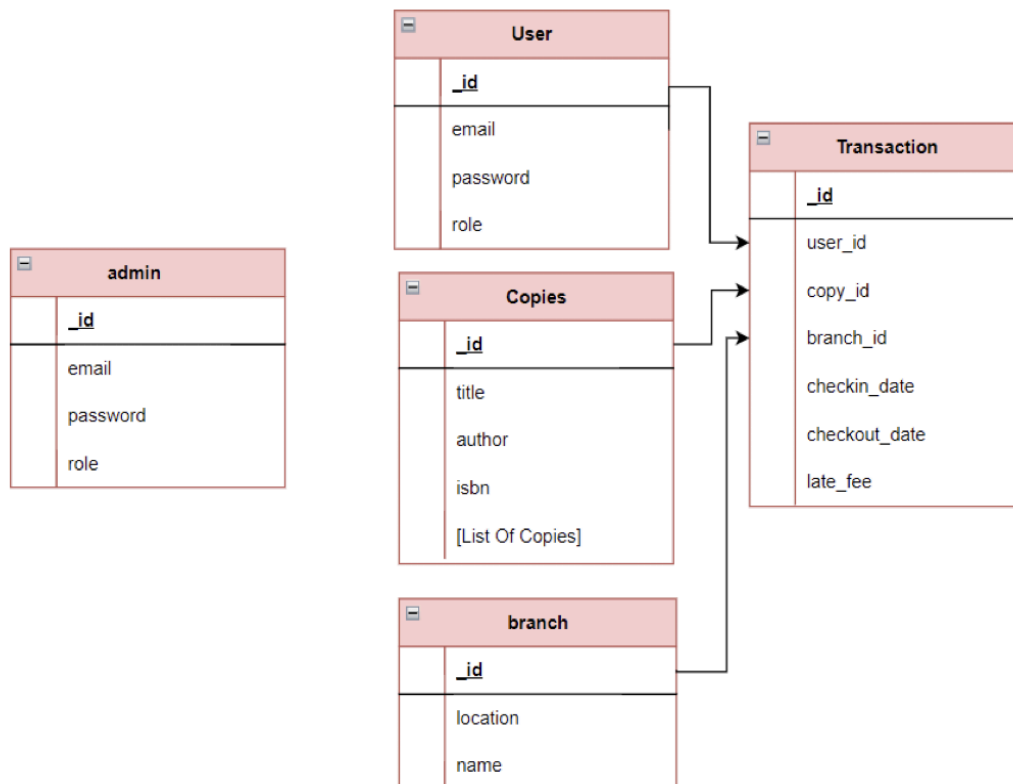
- **Branch Collection:** Stores information about library branches.
  - Attributes: `branch_id`, `name`, `location`
- **Admin Collection:** Contains details about administrators.
  - Attributes: `admin_id`, `email`, `password`

- **User Collection:** Maintains data of registered users.
  - Attributes: `user_id`, `email`, `password`
- **Copies Collection:** Houses comprehensive book information and manages book copies across branches.
  - Attributes: `copy_id`, `ISBN`, `title`, `author`, `genre`, `status`, `branch_id`
- **Transaction Collection:** Records book borrowings, linking users with book copies.
  - Attributes: `transaction_id`, `user_id`, `copy_id`, `branch_id`, `checkin_date`, `checkout_date`, `late_fee`

#### Relations:

- **User to Transaction:** One user can have multiple transactions (One-to-Many).
- **Copy to Branch:** Each copy is associated with one branch (Many-to-One).
- **Copy to Transaction:** One copy can have multiple transactions (One-to-Many).
- **Branch to Copy:** One branch can have multiple copies (One-to-Many).
- **Branch to Transaction:** One branch can be involved in multiple transactions (One-to-Many).

### 3. Database Diagram



#### 4. Data Dictionary

Collection	Field	Type	Description
Branch	branch_id	ObjectId	Unique identifier for the branch
	name	String	Name of the branch
	location	String	Location of the branch
Admin	admin_id	ObjectId	Unique identifier for the admin
	email	String	Email address of the admin
	password	String	Encrypted password of the admin
User	user_id	ObjectId	Unique identifier for the user
	email	String	Email address of the user
	password	String	Encrypted password of the user
Copies	copy_id	ObjectId	Unique identifier for the book copy
	ISBN	String	International Standard Book Number
	title	String	Title of the book
	author	String	Author of the book
	genre	String	Genre of the book
	status	String	Status of the book copy (e.g., Available)
	branch_id	ObjectId	Identifier of the branch containing the copy
Transaction	transaction_id	ObjectId	Unique identifier for the transaction
	user_id	ObjectId	Identifier of the user involved
	copy_id	ObjectId	Identifier of the book copy involved
	branch_id	ObjectId	Identifier of the branch involved
	checkin_date	Date	Date the book was checked in
	checkout_date	Date	Date the book was checked out

	late_fee	Number	Late fee incurred, if any
--	----------	--------	---------------------------

## 5. Sample Data

### Admin:

```
{
  "_id": {"$oid": "666e9f0942c094290f1d0ec2"},
  "email": "admin1@lms.com",
  "password": {"$binary": {"base64":
"JDJiJDEyJFNpL3liRFRtSk55OFQ3RzguZDc1Yi5QMDUvNi9PZ1dVWmdMc04yM0Y3VU94Sjk5QmM1c
Wcy", "subType": "00"}},
  "role": "Admin"
}
```

### Branch:

```
{
  "_id": {"$oid": "666bcf16eb3ce9a67fe300ce"},
  "location": "100 W 105 ST Johnson, Overland Park",
  "name": "Branch 1"
}
```

### Copies:

```
{
  "_id": {"$oid": "6670f5e41fae275325b5210a"},
  "title": "Dolls",
  "author": "Suellen Phaup",
  "isbn": "076704205-0",
  "copies": [
    {
      "branchName": "Branch 2",
      "copiesDetails": [
        {
          "copyNumber": 4955,
          "status": "Borrowed"
        }
      ]
    }
  ]
}
```

### Transaction:

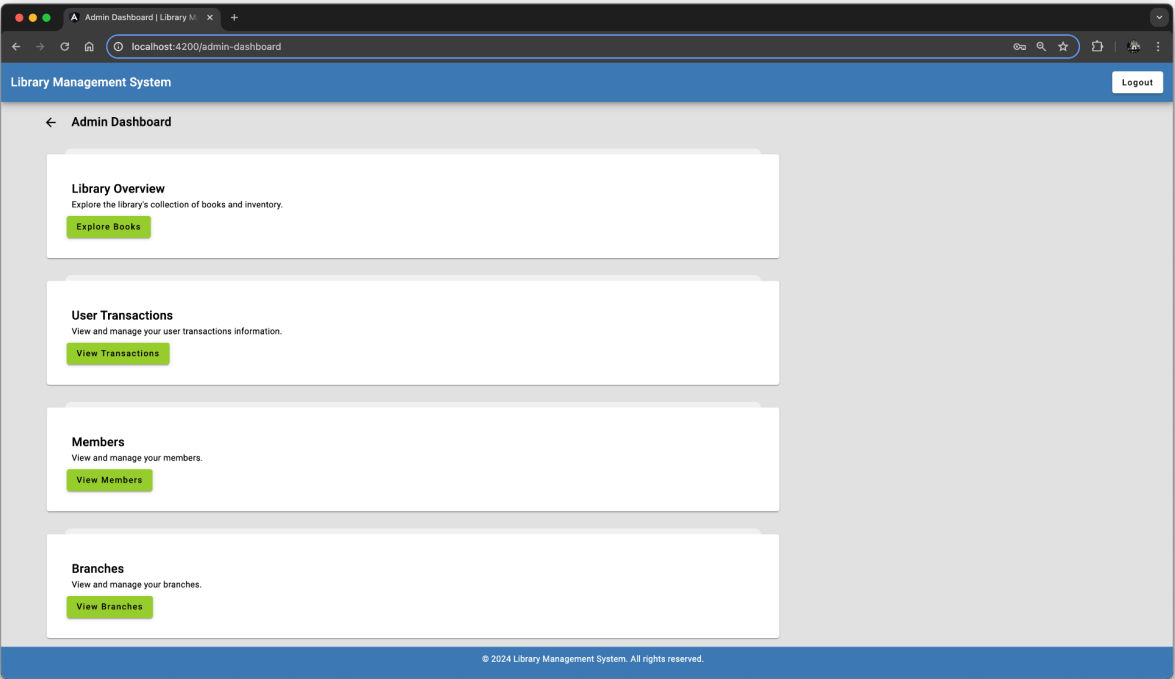
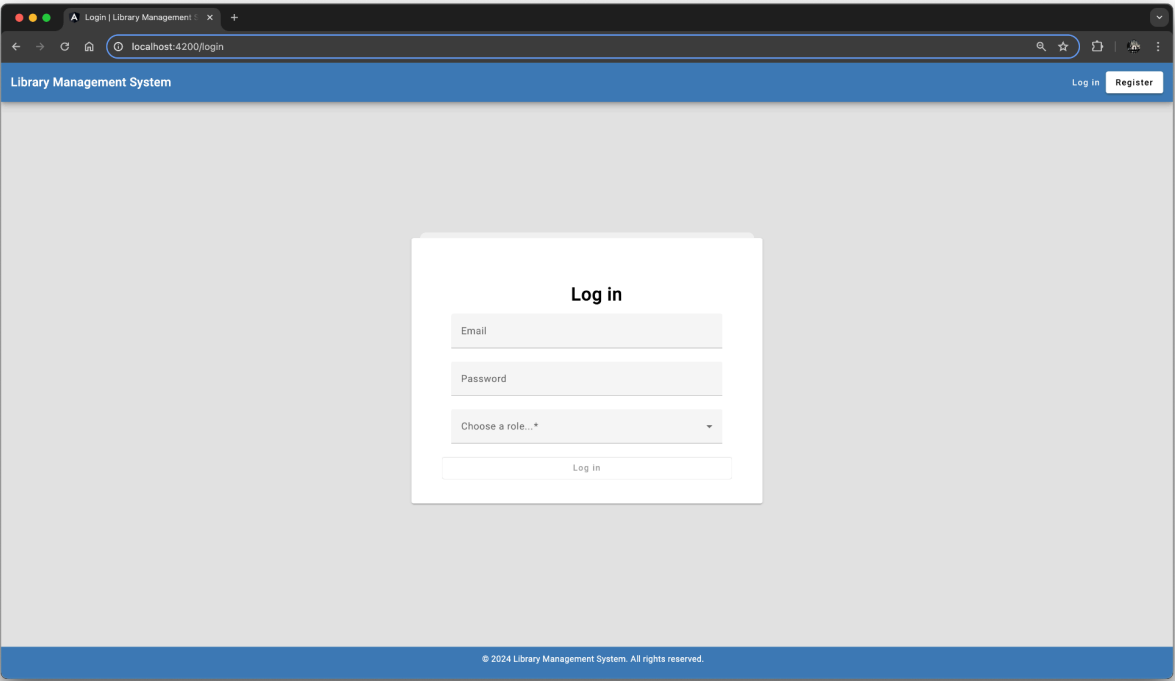
```
{
  "_id": {"$oid": "666c7455bacd7ac1290526ca"},
  "user_id": {"$oid": "666bcd155eef3cb4c036f1ab"},
  "copy_id": "5656",
  "checkout_date": {"$date": "2024-06-14T16:48:21.670Z"},
  "checkin_date": {"$date": "2024-06-14T16:53:35.578Z"},
  "late_fee": 0,
  "branch_name": "Branch 1"
}
```

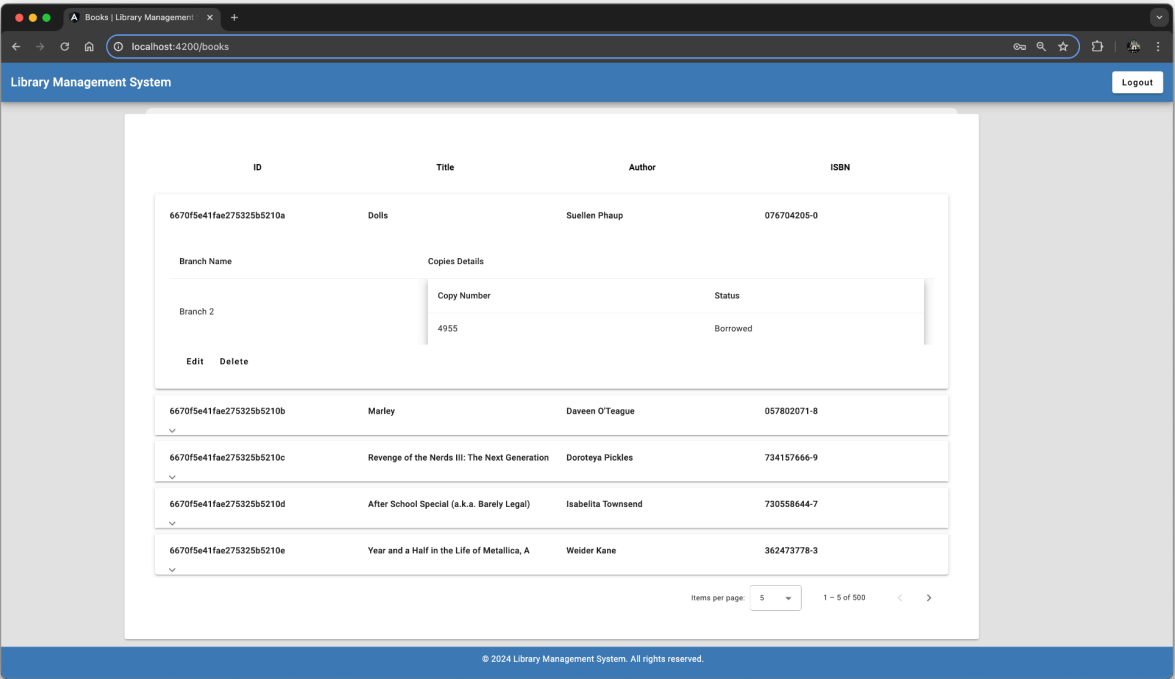
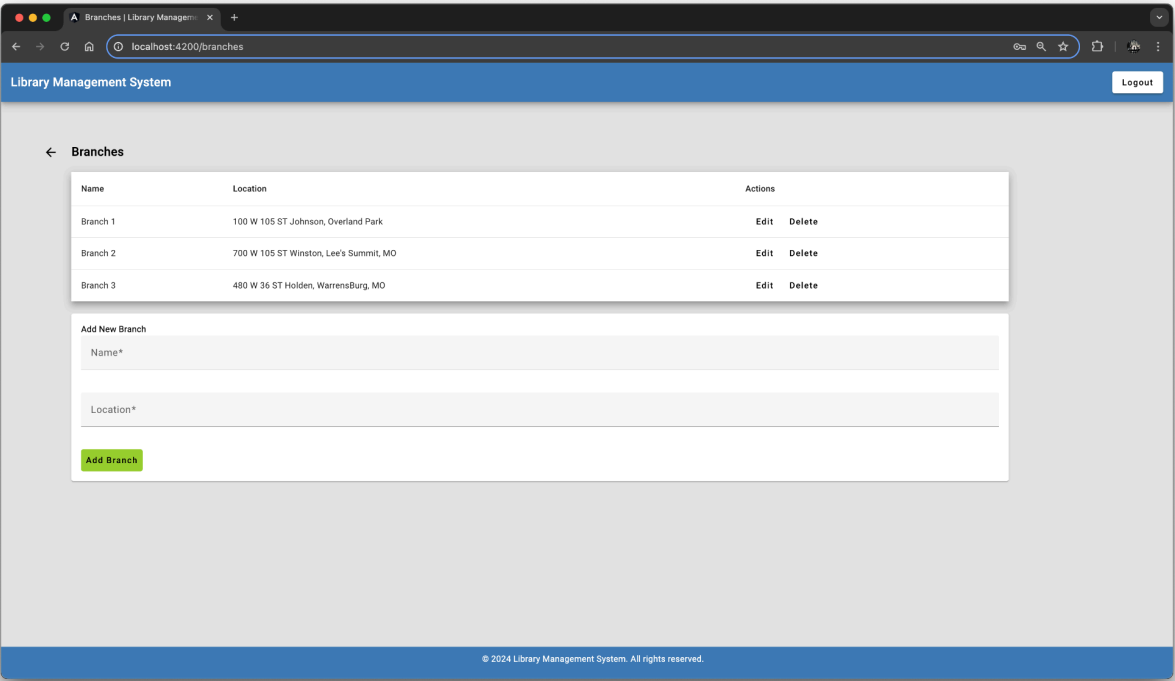
### User:

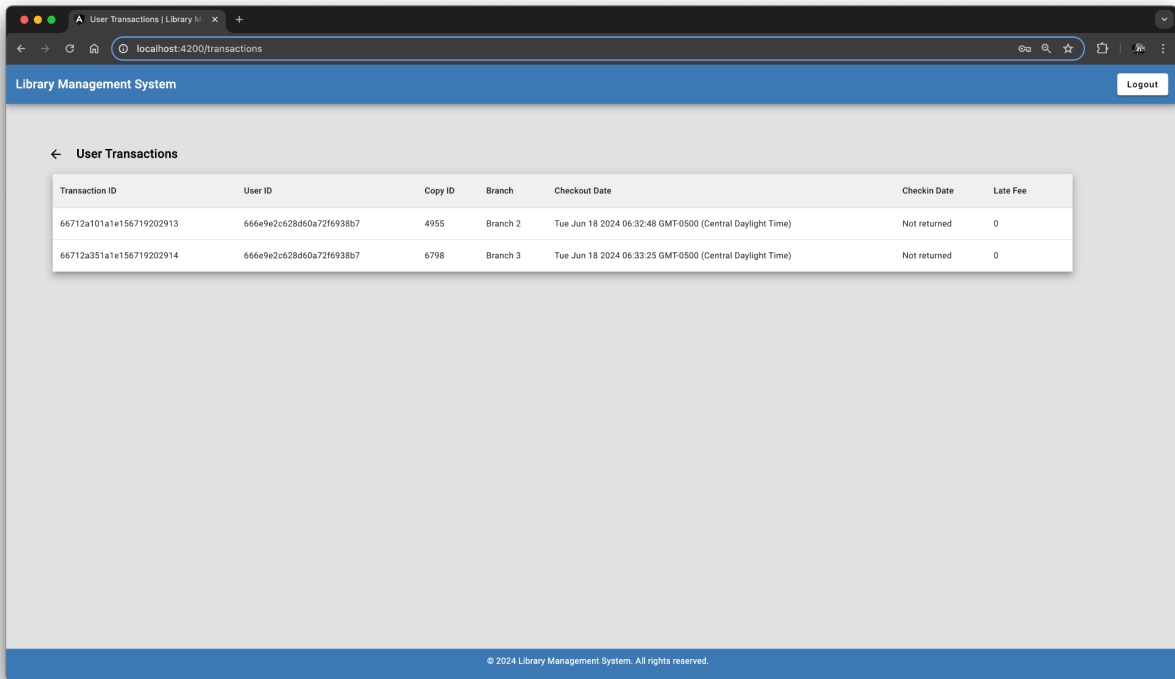
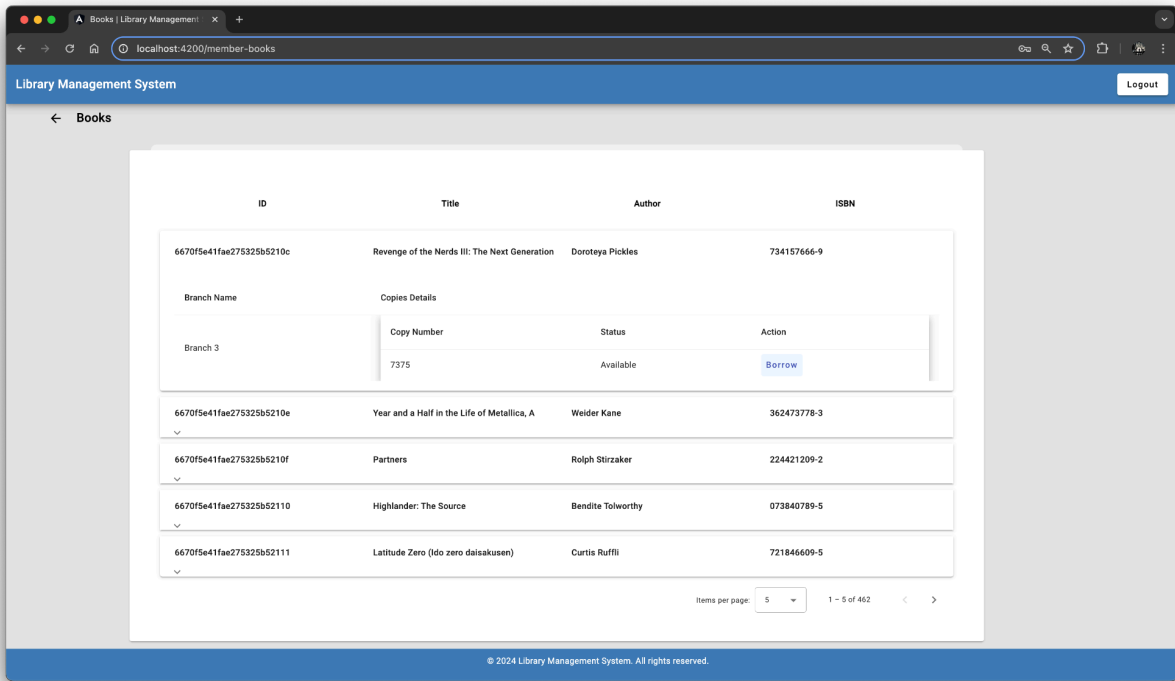
```
{
  "_id": {"$oid": "666e9e2c628d60a72f6938b7"},
  "email": "chandan.dy1995@gmail.com",
  "password": {"$binary": {"base64":
"JDJiJDEyJHdrlM1lwTW5XQ0VkVUxxMWY0aE9GYU9uWTBPFpFanhDTDYxMW1UWDNYa0ZGSGJkSTRvY
3Jp", "subType": "00"}},
  "role": "Member"
}
```

## 6. User Interfaces and Forms

The screenshot displays a web browser window with the address bar showing 'localhost:4200/register'. The page title is 'Library Management System'. In the top right corner, there are links for 'Log in' and 'Register'. The main content area features a white 'Register' form with the following fields: First Name, Last Name, Date of Birth (with a calendar icon), Email, Password, Confirm Password, Zip Code, Phone Number, and a dropdown menu labeled 'Choose a role...'. A 'Register' button is located at the bottom of the form. The footer of the page contains the text '© 2024 Library Management System. All rights reserved.'







## 7. Source Code

<https://github.com/chandan5614/LMS/tree/main>



## 8. NoSQL Commands

### Retrieving Branch Information for a Copy:

```
db.copies.findOne({ copy_id: "5656" }, { branch_id: 1 });
```

### Tracking Transactions by User:

```
db.transactions.find({ user_id: ObjectId("666bcd155eef3cb4c036f1ab")
});
```

### Managing Book Copies Across Branches:

```
db.copies.find({ branch_id: ObjectId("666bcf16eb3ce9a67fe300ce") });
```

### Borrow a copy:

```
db.copies.update_one(
    {'_id': book_id, 'copies.copiesDetails.copyNumber': copy_id},
    {'$set': {'copies.$[outer].copiesDetails.$[inner].status':
'Borrowed'}},
    array_filters=[{'outer.branchName': data['branch_id']},
{'inner.copyNumber': copy_id}]
);
```

## Conclusion

The Library Management System provides a comprehensive solution for managing library operations efficiently. By leveraging MongoDB, the system ensures scalability and flexibility to handle complex data relationships, making it a robust tool for libraries of varying sizes. This project demonstrates the effective application of NoSQL databases in real-world scenarios, highlighting the advantages of MongoDB in managing and querying interconnected datasets.