MongoDB Lab Report - Week 05

DengFeng Qiao Yi He MingYi Wang

October 13, 2024

Introduction

This report summarizes the tasks related to MongoDB completed as part of Week 05 lab session. The tasks are divided into basic, medium, and advanced categories, and cover MongoDB operations such as document relationships, schema flexibility, data queries, and MongoDB-specific operations equivalent to SQL commands.

Basic Tasks

Task 1: MongoDB Document Relationships

MongoDB handles relationships between documents by using either embedded documents or references to other documents. Embedded documents are stored within the parent document, allowing for denormalization, while references create a connection between separate documents similar to foreign keys in SQL.

Task 2: MongoDB Schema-less Nature

Unlike SQL, MongoDB does not require a predefined schema. Collections are created automatically when inserting documents, and each document can have a different structure. This makes MongoDB "schema-less," allowing for greater flexibility in data modeling, which is particularly useful in dynamic environments.

Task 3: MongoDB Dynamic Schema

MongoDB supports a dynamic schema where documents in the same collection can have different fields. This flexibility means documents can evolve over time without requiring structural changes to the database.

Medium Tasks

Task 4: SQL to MongoDB Translation

Here are some SQL statements and their equivalent MongoDB commands:
- **CREATE/INSERT:** SQL: `CREATE TABLE`, `INSERT INTO`
MongoDB: `db.collection.insertOne()` automatically creates the collection and inserts a document.

- **SELECT Queries:** SQL: `SELECT` with conditions
MongoDB: `db.collection.find()` with query filters and field projections.

For example:

```
db.employee.insertOne({empNo: 'E55', lastName: 'Carmen', firstName: 'Silva', hours:
db.employee.find({hours: {$gt: 400}})
```

Task 5: Big Data and MongoDB Variety

Big Data is often described by the three Vs: Volume, Velocity, and Variety. MongoDB efficiently handles the variety aspect of Big Data by supporting different types of data, including structured, semi-structured, and unstructured data, within a flexible document model.

Advanced Tasks

Task 6: MongoDB Queries on a Books Collection

Here are several MongoDB queries to perform various operations on a collection of books:

i) Find books published between 2019 and 2024:

```
db.books.find({"year": {$gte: 2019, $lte: 2024}})
```

ii) Find books where book_id is not equal to 552020:

```
db.books.find({"book_id": {$ne: "552020"}})
```

iii) Find books by D. Sullivan or ISBN 9780134023212:

```
db.books.find({$or: [{"author": "D. Sullivan"}, {"ISBN": "9780134023212"}]})
```

iv) Find books with ISBN 9876543210 or 0123456789:

```
db.books.find({"ISBN": {$in: ["9876543210", "0123456789"]}})
```

v) Find books with titles containing ”SQL”:

```
db.books.find({"title": /SQL/})
```

vi) Count books published by Addison-Wesley:

```
db.books.find({"publisher": "Addison-Wesley"}).count()
```

vii) Find books published in 2019 and titles containing ”Mortals”, sorted alphabetically:

```
db.books.find({"year": 2019, "title": /Mortals/}).sort({"title": 1})
```

viii) Add a ”subject” field to books published in 2019:

```
db.books.updateMany({"year": 2019}, {$set: {"subject": "computing"}})
```

Conclusion

The tasks in this lab session illustrated the differences between MongoDB and SQL in terms of schema flexibility, query operations, and data handling. MongoDB's document-based model and dynamic schema provide a flexible approach to managing diverse data types, making it a strong candidate for Big Data applications.