NOSQL PROJECT

Daire O`Neill

IT3

Horsepower Fitness

# 

Contents

[Introduction 2](#_Toc132851203)

[What is NoSQL? 2](#_Toc132851204)

[Scope out the Project 2](#_Toc132851205)

[Collections 2](#_Toc132851206)

[Database Design Using the DOM Technique 3](#_Toc132851207)

[Download, install and test a Mongo database environment. 3](#_Toc132851208)

[Creating a NoSQL Database CRUD 4](#_Toc132851209)

[Creating Back-ups and Deploying the Database into a Suitable Cloud Environment 5](#_Toc132851210)

[Creating Aggregation pipelines 8](#_Toc132851212)

[Conclusion 10](#_Toc132851213)

[All the Data 10](#_Toc132851214)

# Introduction

The goal of this project is to develop a NoSQL database for a gym management system that will have a minimum of three collections. The database will store information about gym members, workout plans, and equipment. The NoSQL database will provide the necessary scalability and flexibility to handle large amounts of data and dynamic changes in the structure of the data.

# What is NoSQL?

NoSQL is a type of database management system that offers a more flexible and scalable approach to storing and retrieving data compared to traditional relational databases. It provides a non-tabular, distributed architecture that supports unstructured and semi-structured data. NoSQL databases do not have a fixed schema or data model, allowing for dynamic data management and better scalability. There are different types of NoSQL databases, including document-oriented, key-value stores, graph databases, and column-family stores, which are often used for web applications, real-time analytics, and big data processing. Although NoSQL databases have some limitations, they offer benefits such as improved scalability, performance, and availability.

# Scope out the Project

# Collections

Members collection: This collection will store information about the gym members. Each member document will contain fields such as member ID, name, date of birth, gender, contact information, membership plan, medical conditions.

Membership Plan collection: This collection will store information about the membership plan for each gym member. Each membership plan document will contain fields such as membership plan ID, name, length, cost.

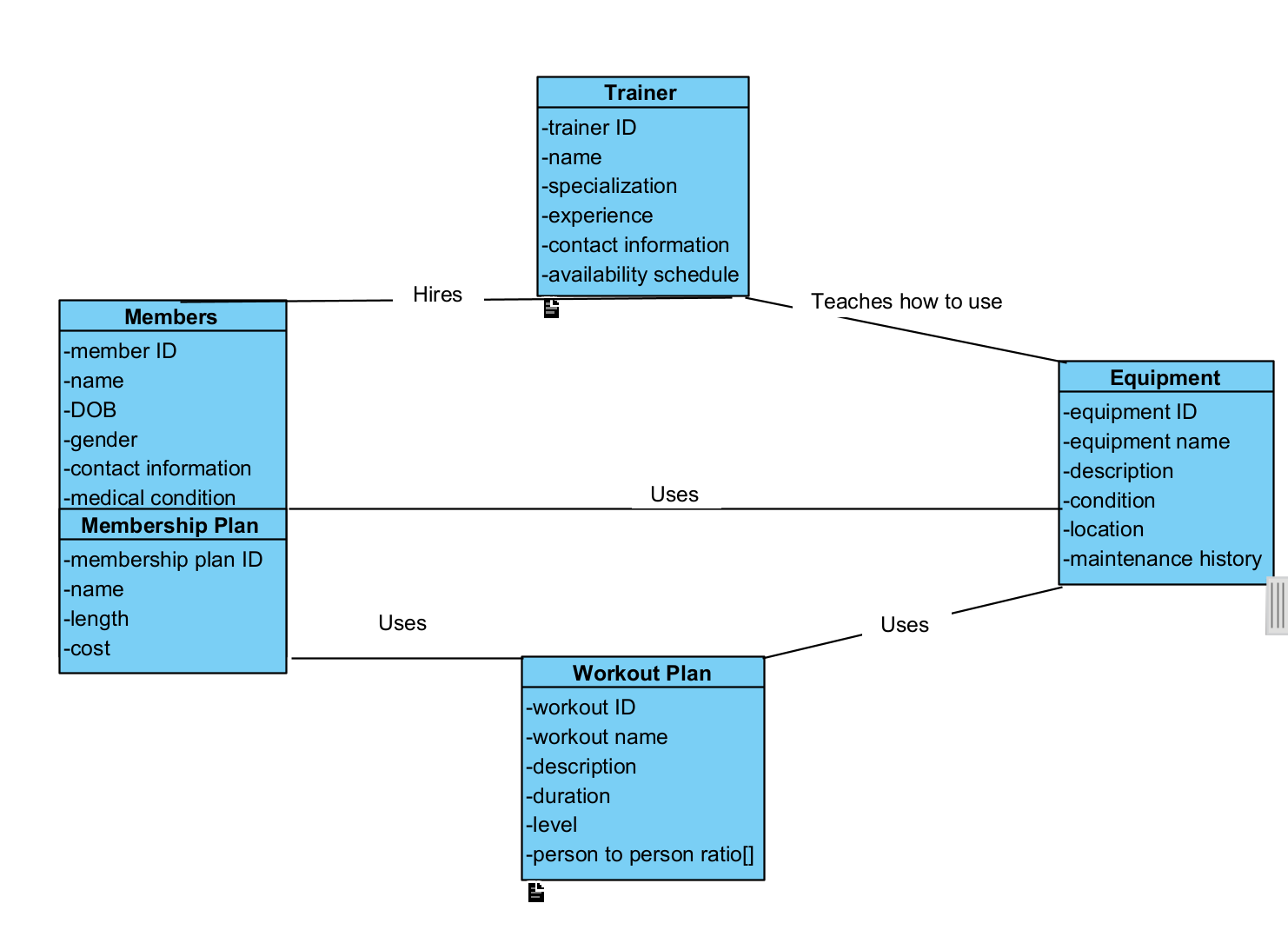
Workouts collection: This collection will store information about the workout plans offered by the gym. Each workout document will contain fields such as workout ID, workout name, description, duration, level, and equipment required.

Equipment collection: This collection will store information about the gym equipment. Each equipment document will contain fields such as equipment ID, equipment name, description, condition, location, and maintenance history.

Trainers collection: This collection will store information about the gym's trainers. Each trainer document will contain fields such as trainer ID, name, specialization, experience, contact information, and availability schedule. This collection can be used to manage trainer schedules, assign trainers to specific members, and track trainer performance. It can also be used to generate reports on the utilization and effectiveness of the gym's trainers.

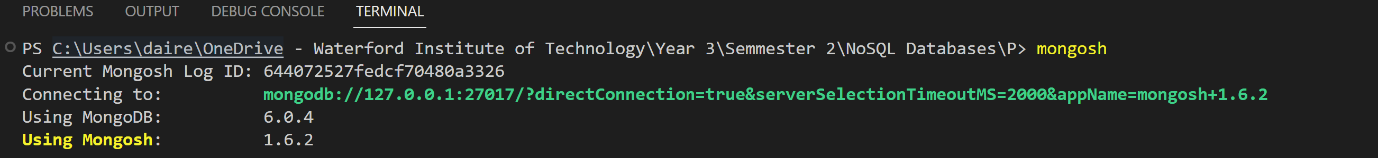
# Database Design Using the DOM Technique

The database will be designed using a NoSQL data model, which will provide the flexibility to handle dynamic changes in the structure of the data. The database will be designed to handle large amounts of data and provide fast retrieval times. It will focus on identifying the key entities in the business domain, defining their attributes and relationships, and mapping them to tables and columns in the database schema.



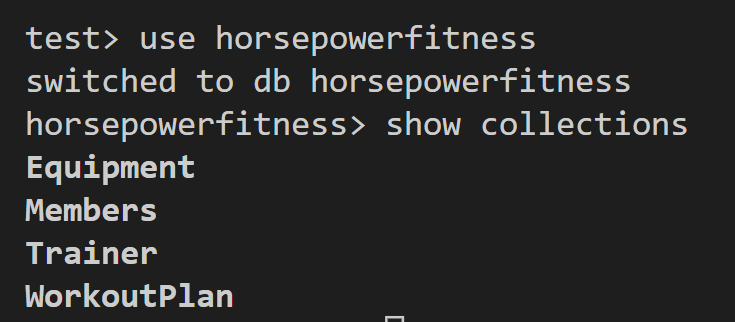
# Download, install and test a Mongo database environment.

I downloaded mongoDB server, as well as a shell (mongosh) using MSI. I then ran both programs, added them to paths in my environmental elements, launched the server, then ran the shell in order to access the server and found everything working. I then installed Visual Studio and ran everything again thought it, which I used exclusively for the rest of my project.

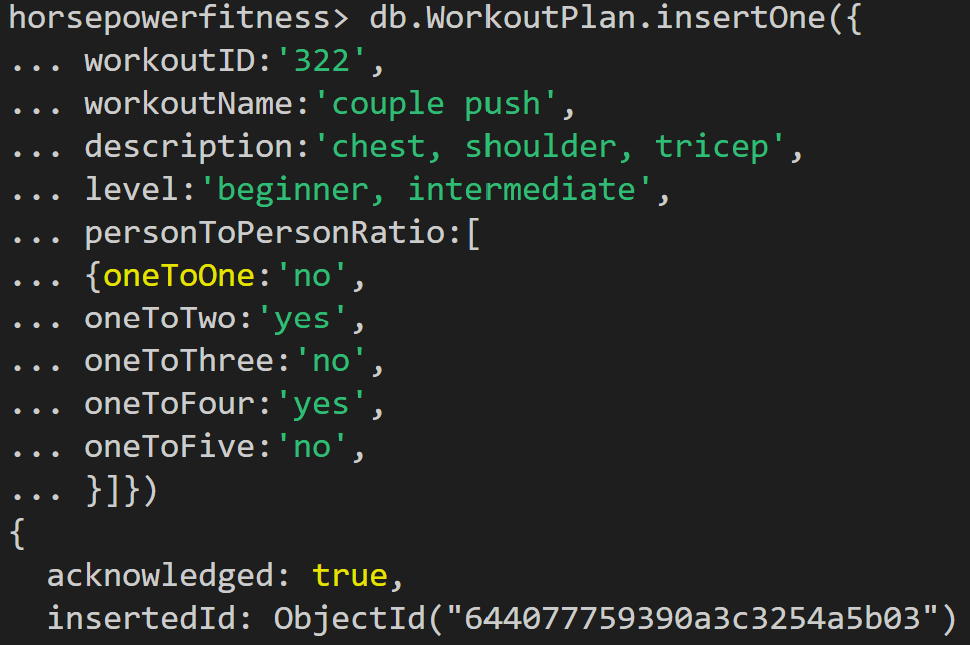


# Creating a NoSQL Database CRUD

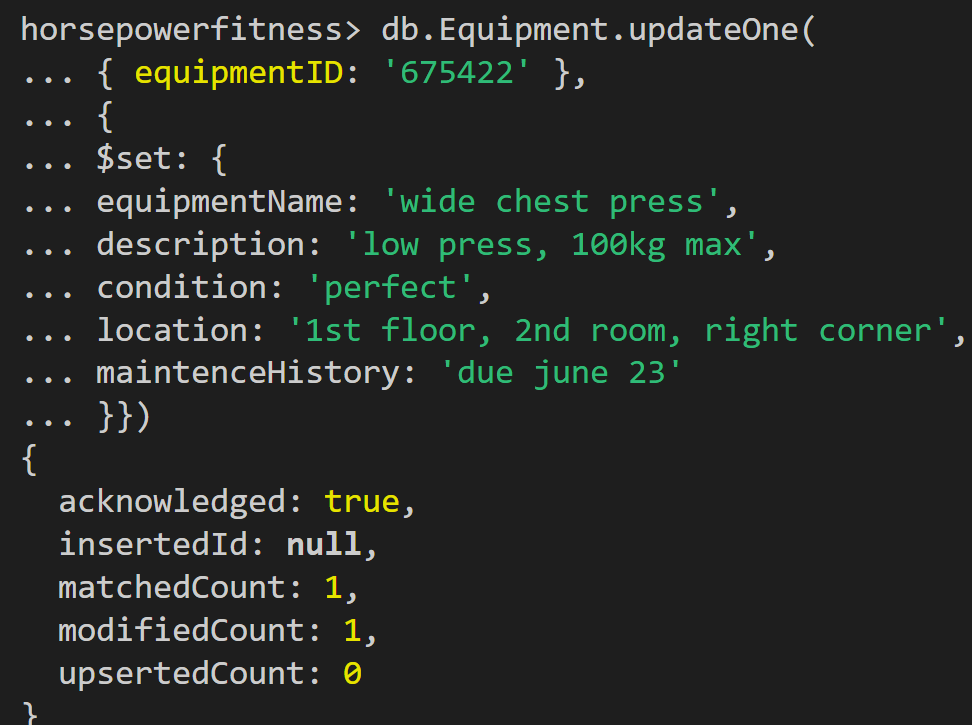
I created the database horsepowerfitness and added collections: Equipment, Members, Trainer and Workoutplan. Members had Membership plan as a sub collection and workoutplan had an array.



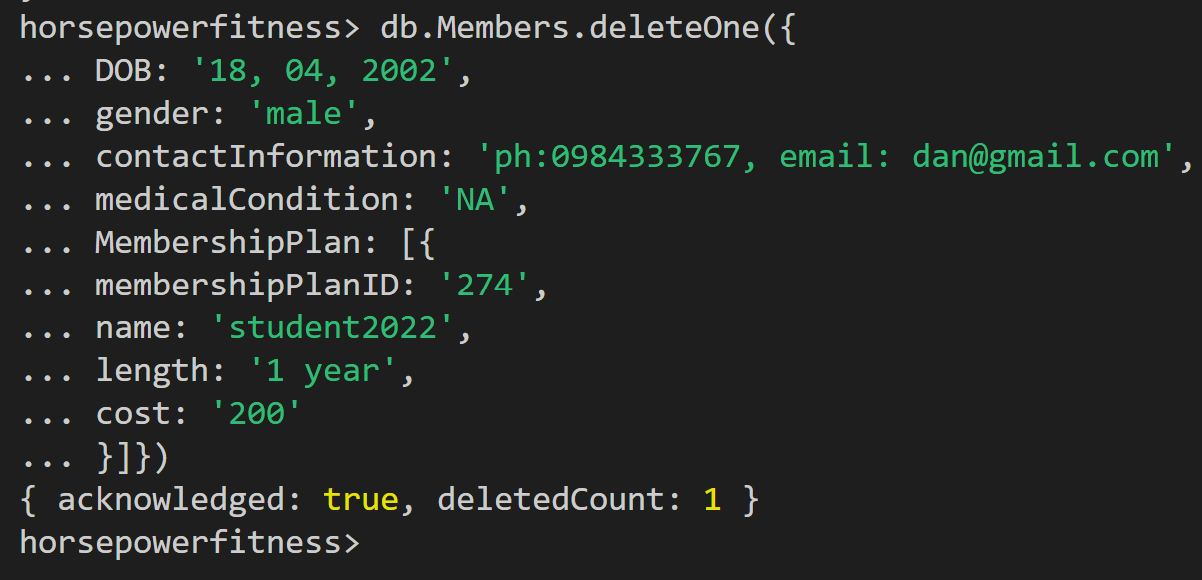
Here is an example of how I created and added data into the database.



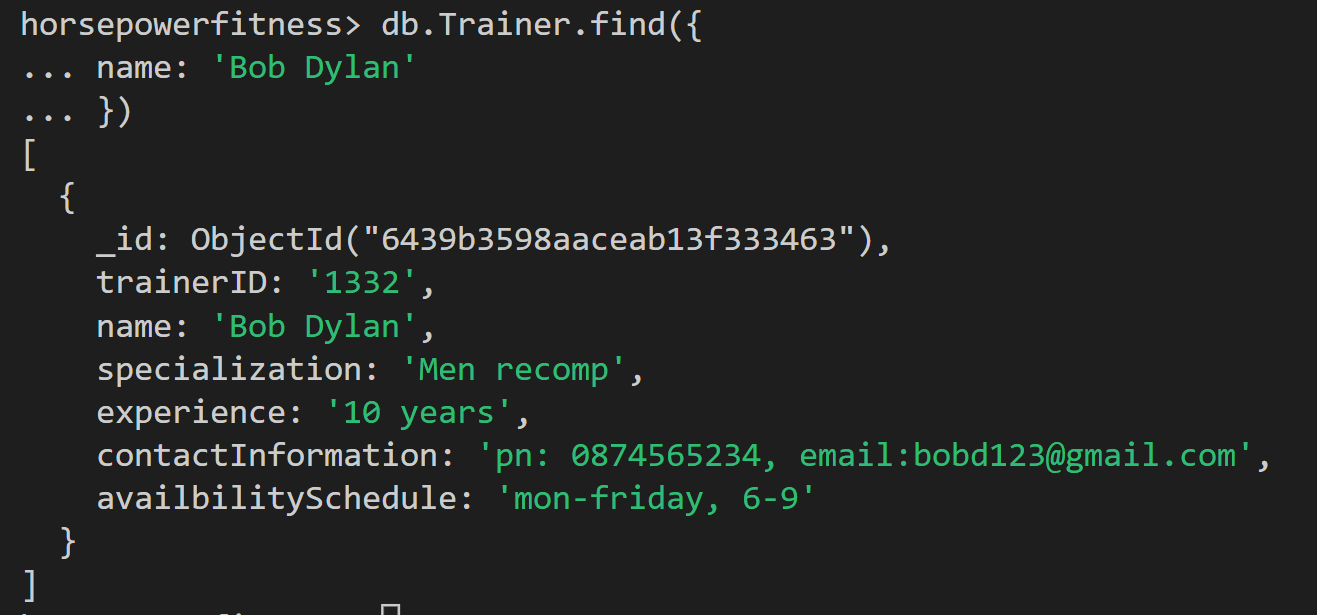
Here is an example of me updating the database.



Here is an example of me deleting from the database.

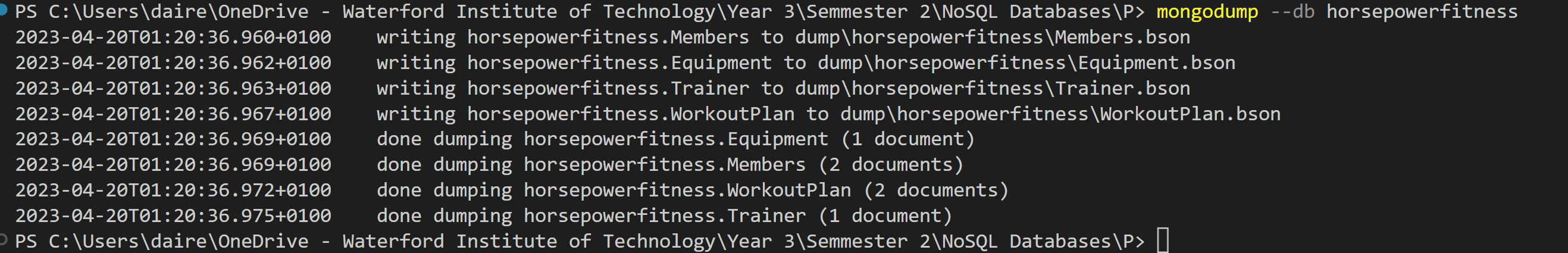


And here is an example of me reading the database.



# Creating Back-ups and Deploying the Database into a Suitable Cloud Environment

Here I used mongodump to create a back-up of my database and store it in on my one drive.



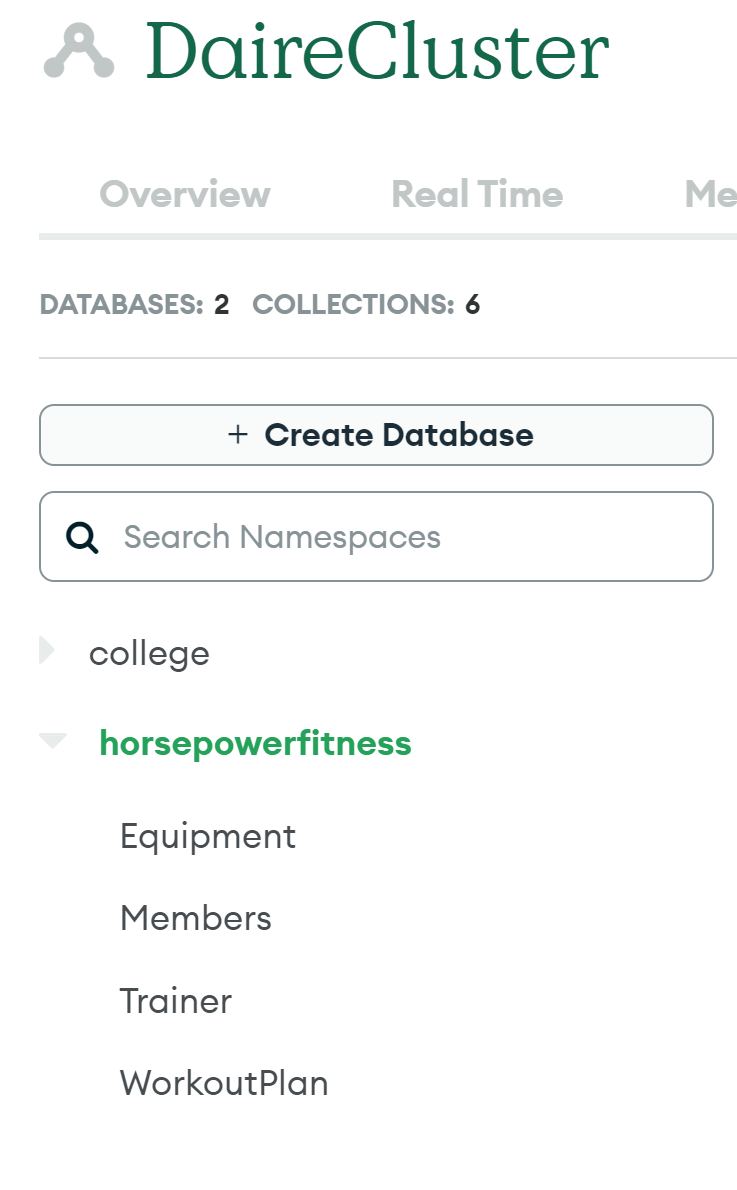
Here I dropped the database.

# 

Then I used mongorestore dump to restore the database.



I decided to use Atlas for as a suitable cloud environment. I deployed a cluster, then used mongoDB for VS Code to connect my database and my cluster. I then copy and pasted mongorestore --uri mongodb+srv://20019346:<PASSWORD>@dairecluster.riyhnvd.mongodb.net into cmd where my database is stored in order to migrate my data to Atlas. In Atlas I can now see my collection. The database has been deployed to a suitable cloud environment.



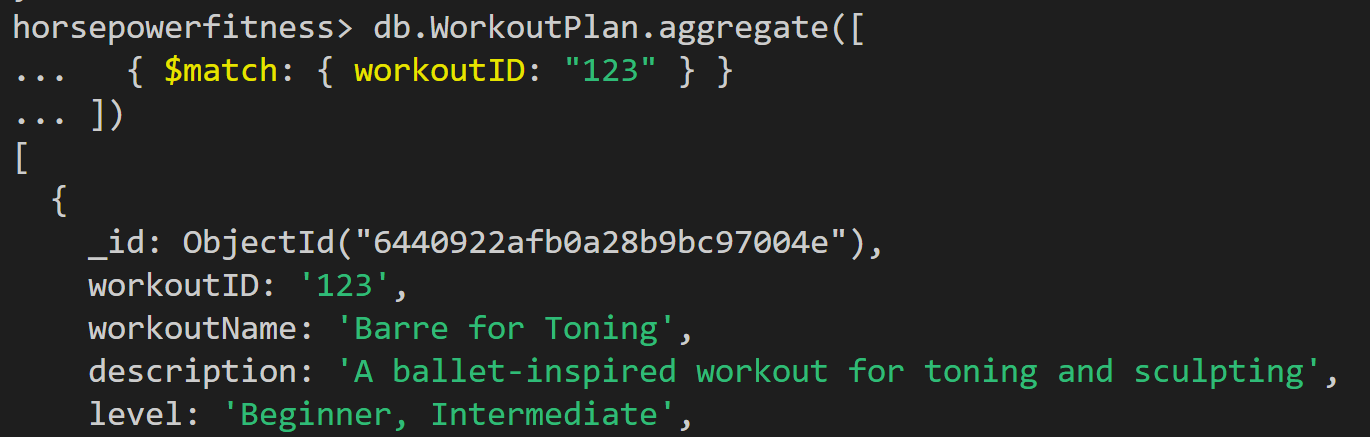
# Creating Aggregation pipelines

Here I used %match to find all intermediate plans.

db.WorkoutPlan.aggregate([

  { $match: { workoutID: "123" } }

])

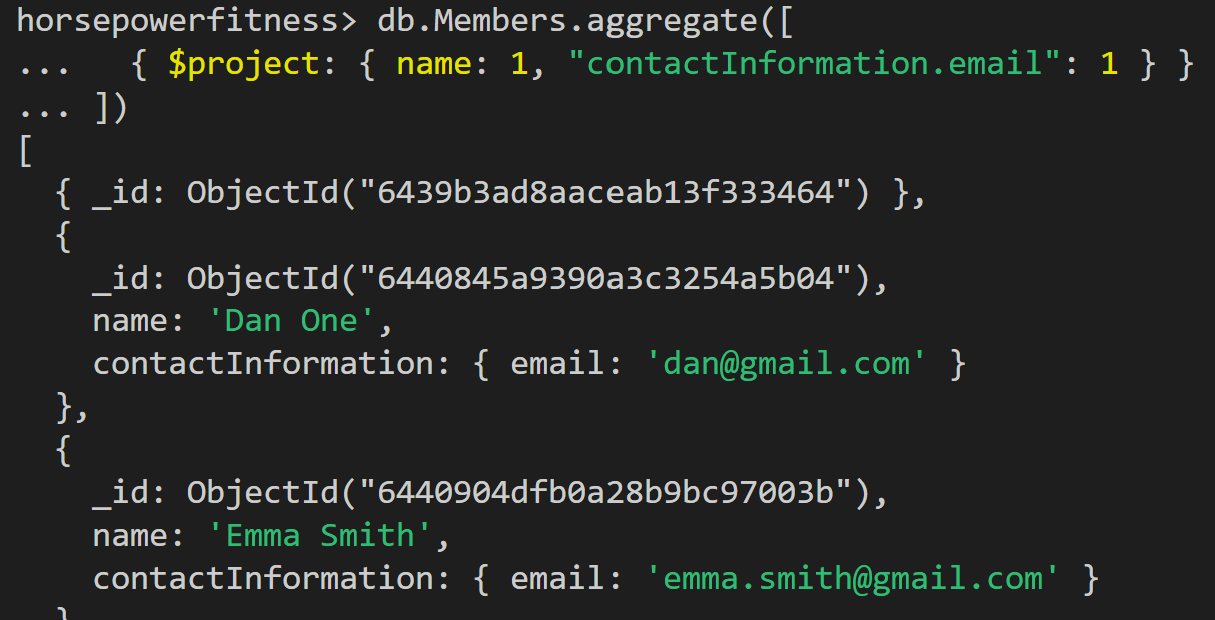


Here I used $project to find specific data from members.

db.Members.aggregate([

  { $project: { name: 1, "contactInformation.email": 1 } }

])



Here I used $count to count how many trainers with more than 5 years experience.

db.Trainer.aggregate([

  {

    $match: {

      experience: { $gt: "5 years" }

    }

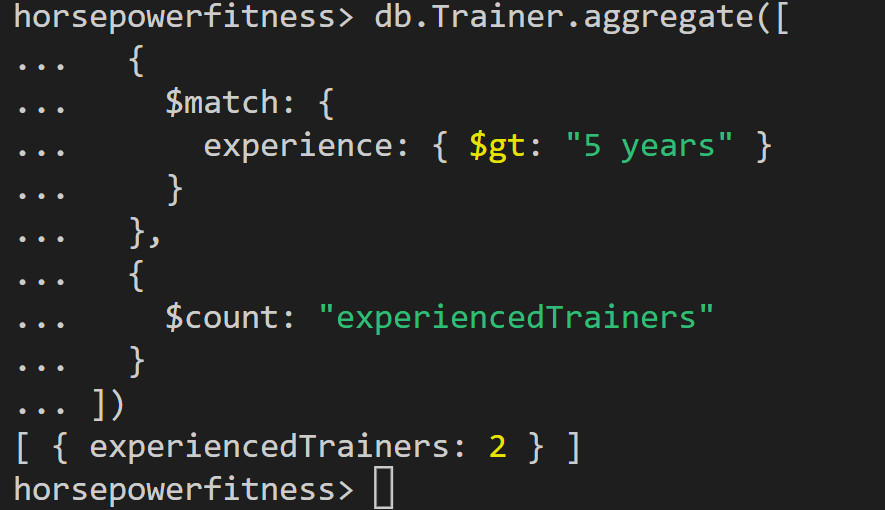
  },

  {

    $count: "experiencedTrainers"

  }

])



Here I used sample to sample one document from members.

db.Members.aggregate([

  {

    $sample: { size: 4 }

  }

])



Here I used $group the equipment based on location and find how many there were in that location.

db.Equipment.aggregate([

  {

    $group: {

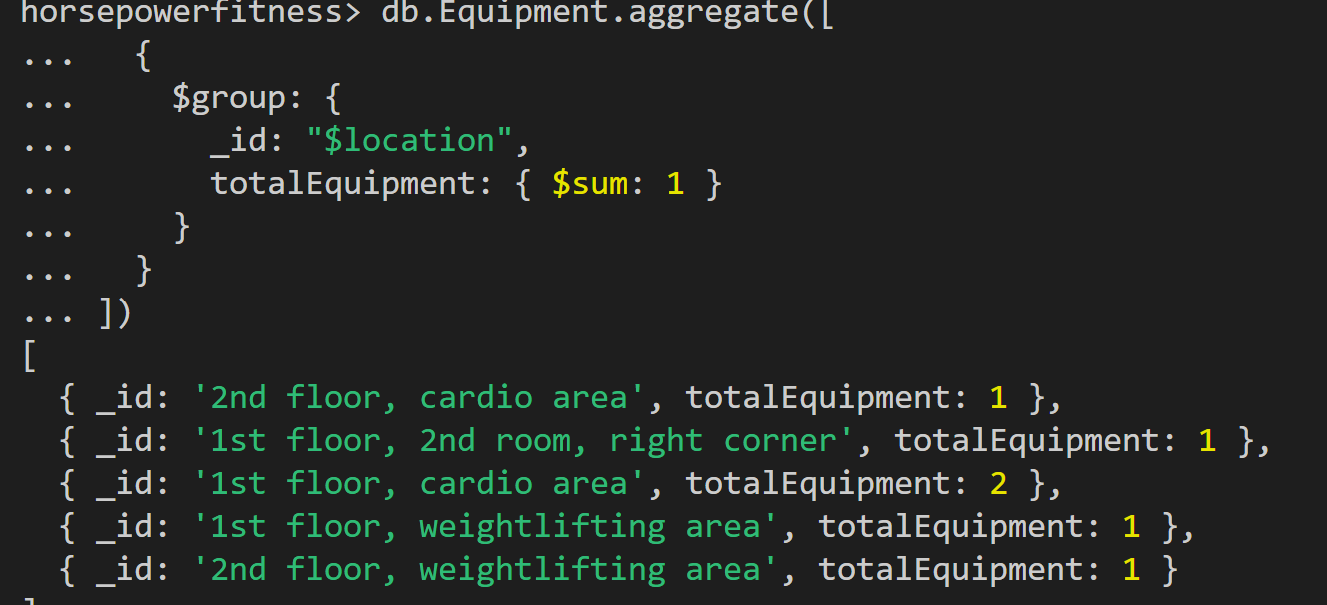
      \_id: "$location",

      totalEquipment: { $sum: 1 }

    }

  }

])



# Conclusion

The NoSQL database designed for horsepower fitness will provide the necessary scalability and flexibility to handle large amounts of data and dynamic changes in the structure of the data. The database will provide fast retrieval times and a secure way to access the data. This will enable the gym to manage its members, workout plans, and equipment efficiently, providing a seamless and convenient experience for its members.

# All the Data

Members:

db.Members.insertOne({

memberID:’123231’,

name:’Dan One’,

DOB:'18, 04, 2002',

contactInformation: {

phone: '0984333767',

email: 'dan@gmail.com'

 },

MedicalConditions:'NA',

MembershipPlan:{

membershipPlanID: '274',

Name:'student2022',

length:'1 year',

cost:'200'

}})

db.Members.insertOne({

    memberID: '789012',

    name: 'Liam Johnson',

    DOB: '08, 07, 1985',

    contactInformation: {

        phone: '0765432198',

        email: 'liam.johnson@hotmail.com'

    },

    MedicalConditions: 'Diabetes',

    MembershipPlan: {

        membershipPlanID: '678',

        Name: 'premium',

        length: '2 years',

        cost: '400'

    }

})

db.Members.insertOne({

    memberID: '234567',

    name: 'Olivia Garcia',

    DOB: '01, 01, 2000',

    contactInformation: {

        phone: '0678901234',

        email: 'olivia.garcia@yahoo.com'

    },

    MedicalConditions: 'NA',

    MembershipPlan: {

        membershipPlanID: '901',

        Name: 'family',

        length: '1 year',

        cost: '300'

    }

})

db.Members.insertOne({

    memberID: '345678',

    name: 'Noah Martinez',

    DOB: '12, 06, 1998',

    contactInformation: {

        phone: '0543219876',

        email: 'noah.martinez@gmail.com'

    },

    MedicalConditions: 'NA',

    MembershipPlan: {

        membershipPlanID: '123',

        Name: 'student2023',

        length: '1 year',

        cost: '250'

    }

})

db.Members.insertOne({

    memberID: '456789',

    name: 'Sophia Brown',

    DOB: '09, 02, 1996',

    contactInformation: {

        phone: '0765432109',

        email: 'sophia.brown@hotmail.com'

    },

    MedicalConditions: 'Allergies',

    MembershipPlan: {

        membershipPlanID: '456',

        Name: 'premium plus',

        length: '3 years',

        cost: '800'

    }

})

Trainers:

db.Trainer.insertOne({

  trainerID: '1332',

  name: 'Bob Dylan',

  specialization: 'Men recomp',

  experience: '10 years',

  contactInformation: {

    phone: '0874565234',

    email: 'bobd123@gmail.com'

  },

  availabilitySchedule: 'mon-fri, 6-9'

})

db.Trainer.insertOne({

    trainerID: '2456',

    name: 'Jane Smith',

    specialization: 'Strength training',

    experience: '5 years',

    contactInformation: {

        phone: '0789567342',

        email: 'janesmithfit@gmail.com'

    },

    availabilitySchedule: 'mon-wed, 2-6'

})

db.Trainer.insertOne({

    trainerID: '7890',

    name: 'Alex Rodriguez',

    specialization: 'Functional training',

    experience: '8 years',

    contactInformation: {

        phone: '0765432987',

        email: 'arodfit@yahoo.com'

    },

    availabilitySchedule: 'tue-thu, 8-12'

})

db.Trainer.insertOne({

    trainerID: '1234',

    name: 'Maria Martinez',

    specialization: 'Yoga',

    experience: '3 years',

    contactInformation: {

        phone: '0654321789',

        email: 'mariayoga@gmail.com'

    },

    availabilitySchedule: 'mon-fri, 7-11'

})

db.Trainer.insertOne({

    trainerID: '5678',

    name: 'John Lee',

    specialization: 'Boxing',

    experience: '6 years',

    contactInformation: {

        phone: '0987654321',

        email: 'johnleefit@hotmail.com'

    },

    availabilitySchedule: 'wed-sat, 4-8'

})

db.Trainer.insertOne({

    trainerID: '9012',

    name: 'Samantha Brown',

    specialization: 'Cardio',

    experience: '2 years',

    contactInformation: {

        phone: '0765432198',

        email: 'samanthabfit@gmail.com'

    },

    availabilitySchedule: 'mon-thu, 3-7'

})

Equipment:

db.Equipment.insertOne({

equipmentID:'675422',

equipmentName:'wide chest press',

description:'low press, 80kg max',

condition:'perfect',

location:'1st floor, 1st room, left corner',

maintenceHistory:'due june 23'

})

db.Equipment.insertOne({

    equipmentID: '345687',

    equipmentName: 'dumbbell set',

    description: '5-50 lbs set',

    condition: 'good',

    location: '2nd floor, weightlifting area',

    maintenceHistory: 'due september 23'

})

db.Equipment.insertOne({

    equipmentID: '891234',

    equipmentName: 'treadmill',

    description: 'commercial grade',

    condition: 'excellent',

    location: '1st floor, cardio area',

    maintenceHistory: 'due december 23'

})

db.Equipment.insertOne({

    equipmentID: '456789',

    equipmentName: 'rowing machine',

    description: 'water resistance',

    condition: 'fair',

    location: '2nd floor, cardio area',

    maintenceHistory: 'due july 23'

})

db.Equipment.insertOne({

    equipmentID: '012345',

    equipmentName: 'leg press machine',

    description: 'plate loaded',

    condition: 'good',

    location: '1st floor, weightlifting area',

    maintenceHistory: 'due october 23'

})

db.Equipment.insertOne({

    equipmentID: '678912',

    equipmentName: 'elliptical machine',

    description: 'full body workout',

    condition: 'excellent',

    location: '1st floor, cardio area',

    maintenceHistory: 'due november 23'

})

WorkoutPlan:

db.WorkoutPlan.insertOne({

workoutID:'322',

workoutName:'couple push',

description:'chest, shoulder, tricep',

level:'beginner, intermediate',

personToPersonRatio:[

{oneToOne:'no',

oneToTwo:'yes',

oneToThree:'no',

oneToFour:'yes',

oneToFive:'no',

}]})

db.WorkoutPlan.insertOne({

    workoutID: '867',

    workoutName: 'Full Body HIIT',

    description: 'High intensity interval training',

    level: 'Intermediate, Advanced',

    personToPersonRatio: [

        { oneToOne: 'yes',

          oneToTwo: 'no',

          oneToThree: 'no',

          oneToFour: 'no',

          oneToFive: 'no'

        }]

})

db.WorkoutPlan.insertOne({

    workoutID: '420',

    workoutName: 'Yoga for Flexibility',

    description: 'A gentle yoga practice for stretching and relaxation',

    level: 'Beginner',

    personToPersonRatio: [

        { oneToOne: 'yes',

          oneToTwo: 'yes',

          oneToThree: 'no',

          oneToFour: 'no',

          oneToFive: 'no'

        }]

})

db.WorkoutPlan.insertOne({

    workoutID: '999',

    workoutName: 'Pilates for Core Strength',

    description: 'A low-impact workout for building core strength and flexibility',

    level: 'Intermediate',

    personToPersonRatio: [

        { oneToOne: 'no',

          oneToTwo: 'yes',

          oneToThree: 'yes',

          oneToFour: 'no',

          oneToFive: 'no'

        }]

})

db.WorkoutPlan.insertOne({

    workoutID: '555',

    workoutName: 'Cardio Bootcamp',

    description: 'A high-energy workout combining cardio and strength training',

    level: 'Intermediate, Advanced',

    personToPersonRatio: [

        { oneToOne: 'no',

          oneToTwo: 'no',

          oneToThree: 'yes',

          oneToFour: 'yes',

          oneToFive: 'no'

        }]

})

db.WorkoutPlan.insertOne({

    workoutID: '123',

    workoutName: 'Barre for Toning',

    description: 'A ballet-inspired workout for toning and sculpting',

    level: 'Beginner, Intermediate',

    personToPersonRatio: [

        { oneToOne: 'yes',

          oneToTwo: 'yes',

          oneToThree: 'yes',

          oneToFour: 'no',

          oneToFive: 'no'

        }]

})