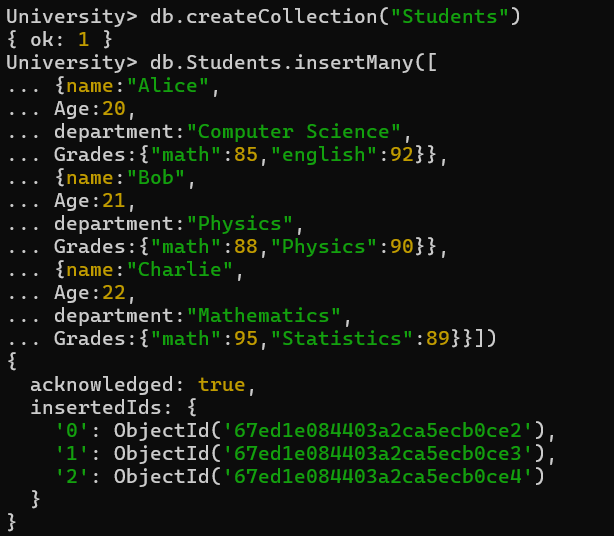
**Basic Of Mongo Db**

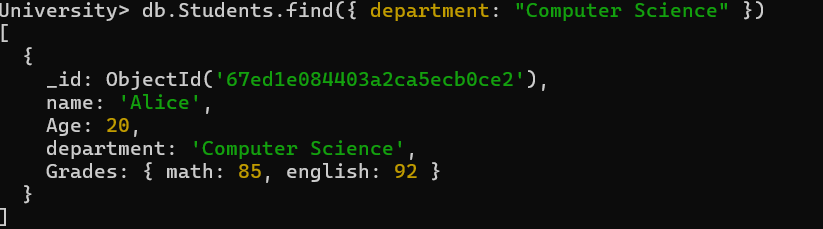
1. *Create a database named university and a collection named students. Insert multiple student documents with fields: name, age, department, and grades.*

**OUTPUT: -**



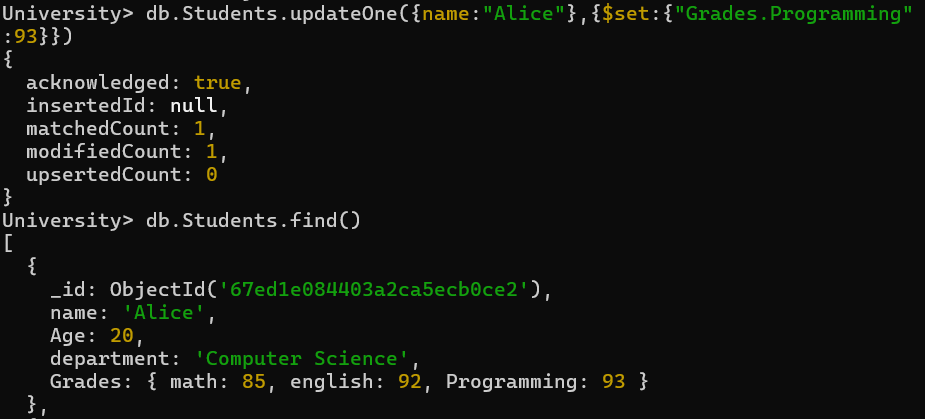
1. *Write a query to display all students who are in the Computer Science department.*

**OUTPUT: - db.Students.find({ department: "Computer Science" })**



1. *Write a query to update the grades of a student named Alice by adding a new subject programming with a grade of 93.*

**OUTPUT: - db.Students.updateOne({name:"Alice"},{$set:{"Grades.Programming":93}})**



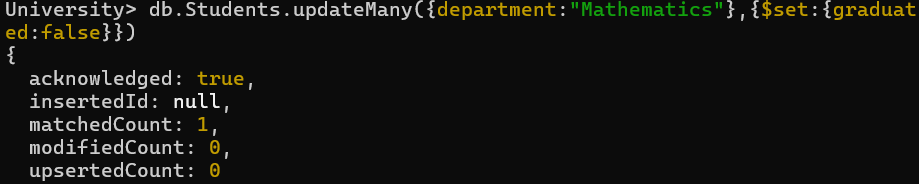
1. *Write a query to increment the age of all students by 1*.

**OUTPUT: - db.Students.updateMany({}, { $inc: { Age: 1 } })**



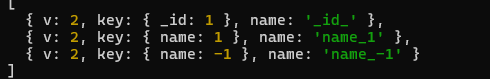
1. *Write a query to delete all students who are 23 years old*.

**OUTPUT: - db.Students.updateMany({department:"Mathematics"},{$set:{graduated:false}})**



1. *Write a query to create an index on the name field of the students collection*.

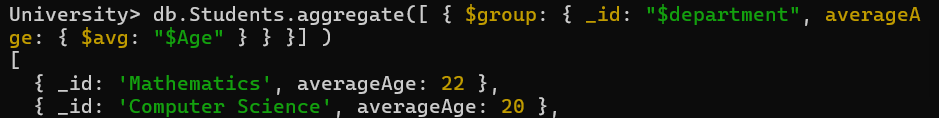
**OUTPUT**: - **db.Students.createIndex({ name: 1 })**



1. *Write an aggregation query to group students by their department and calculate the average age in each department*.

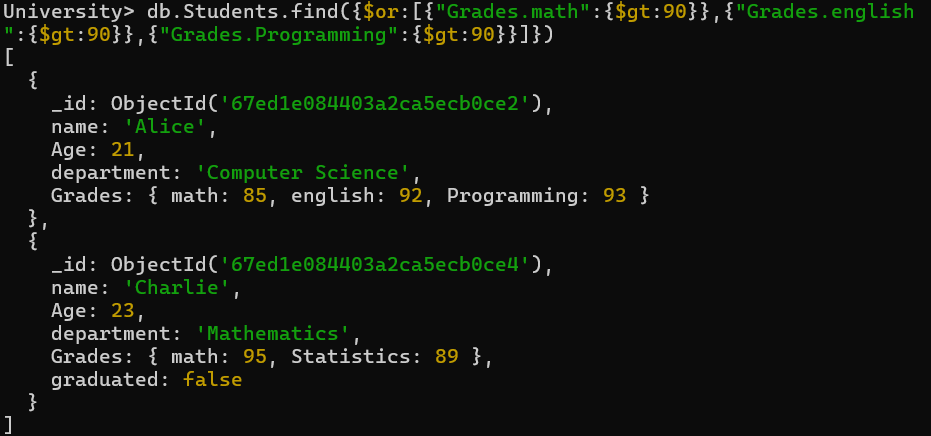
**OUTPUT**: -

**db.Students.aggregate([ { $group: { \_id: "$department", averageAge: { $avg: "$Age" } } }] )**



1. *Write a query to find all students who have scored more than 90 in any subject.*

**OUTPUT: - db.Students.find({$or:[{"Grades.math":{$gt:90}},{"Grades.english":{$gt:90}},{"Grades.Programming":{$gt:90}}]})**



1. *Write a query to add a new field graduated set to false for all students who are in the Mathematics department.*

**OUTPUT**: - **db.Students.deleteMany({age:{$eq:23}})**



1. *How can you retrieve only the name and department fields for all students, excluding the \_id field?*

**OUTPUT**: - **db.Students.find({},{name:1,department:1,\_id:0})**

