Q1: Database & Collection Creation

1. Create a database named studentDB.

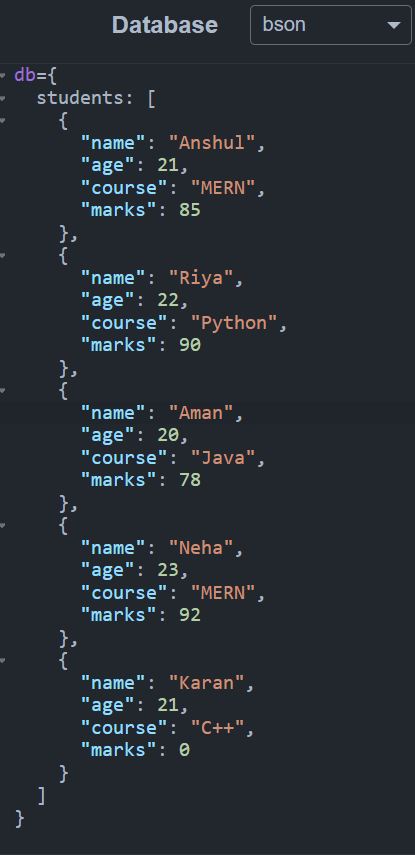
2. Inside it, create a collection named students

db={

students: []

}

Q2: Insert Operations Insert the following documents into the students collection: { "name": "Anshul", "age": 21, "course": "MERN", "marks": 85 } { "name": "Riya", "age": 22, "course": "Python", "marks": 90 } { "name": "Aman", "age": 20, "course": "Java", "marks": 78 } { "name": "Neha", "age": 23, "course": "MERN", "marks": 92 } { "name": "Karan", "age": 21, "course": "C++", "marks": 80



Q3: Read Operations

1. Display all students.

QUERY

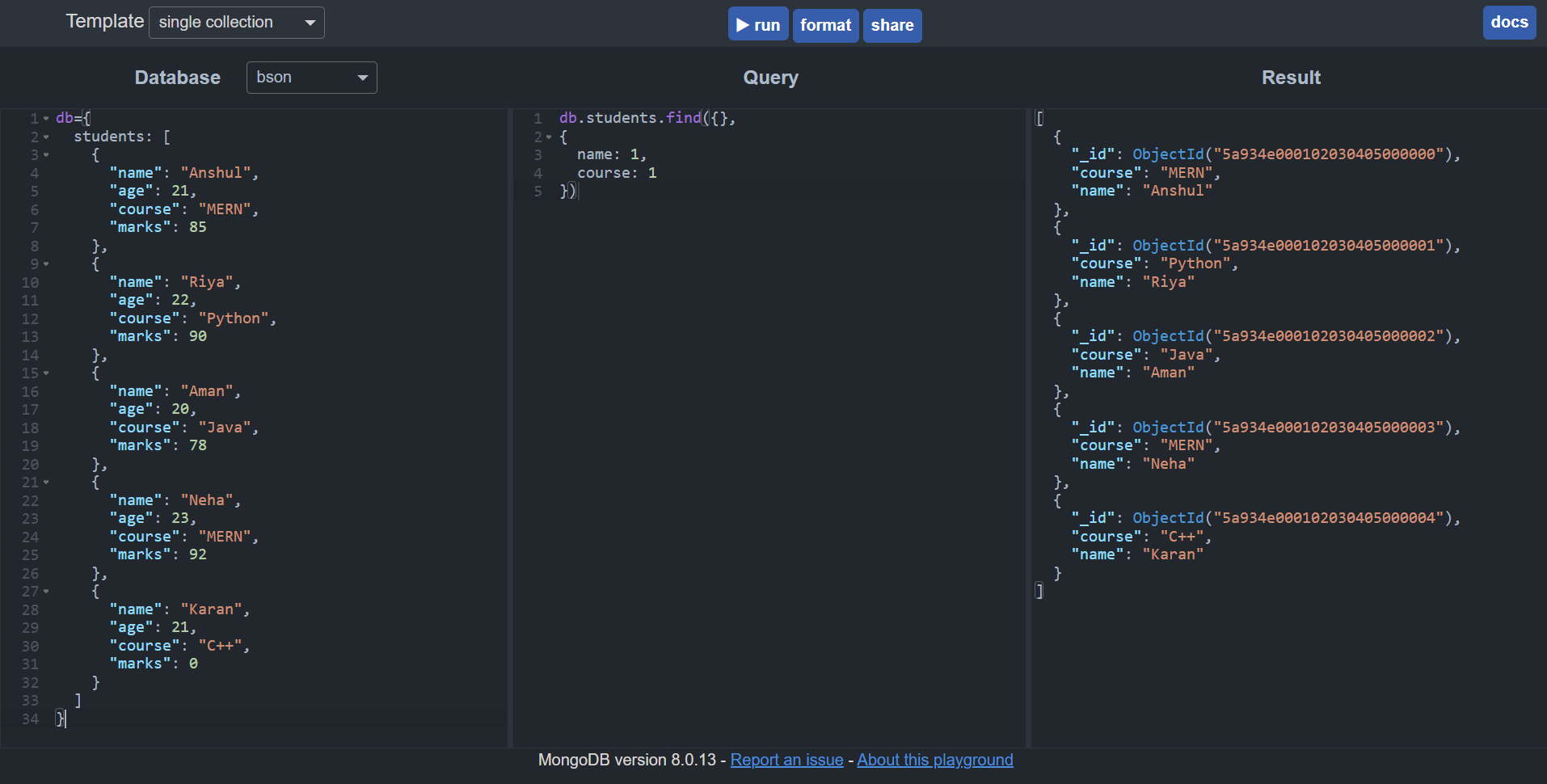
db.students.find({},

{

name: 1,

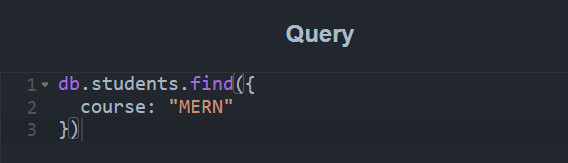
course: 1

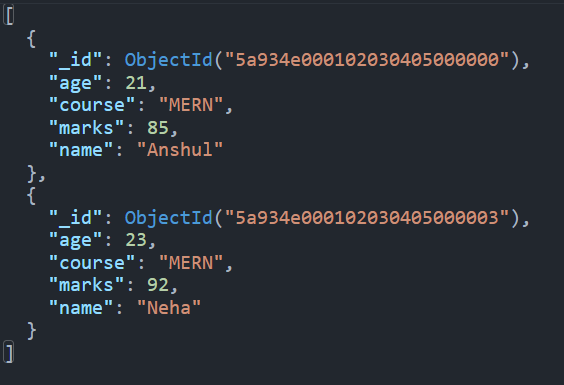
})



1. Display students enrolled in MERN.

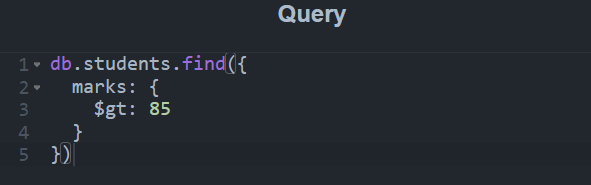
QUERY



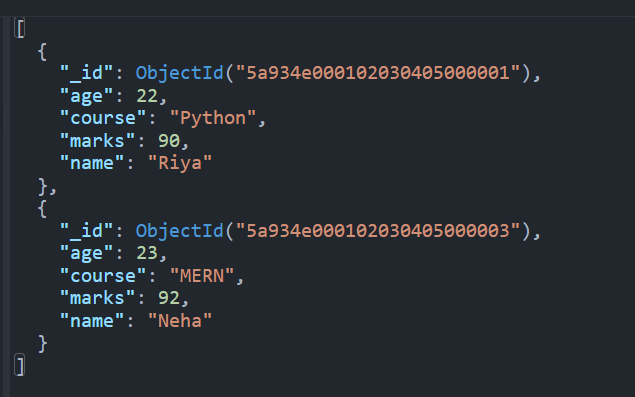
RESULT

1. Find students with marks greater than 85.

query

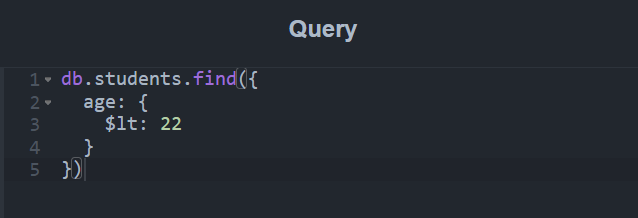


Result



4.Find students whose age is less than 22

query

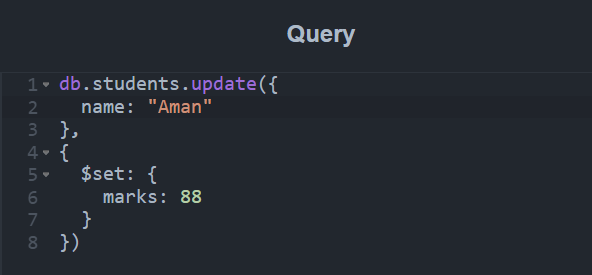
 result



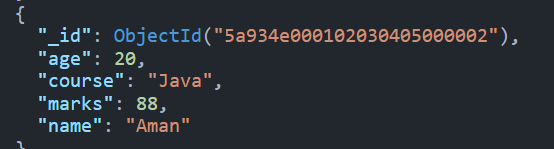
Q4: Update Operations

1. Update Aman's marks to 88

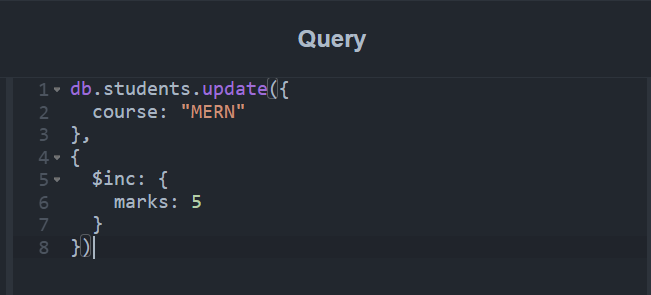
query



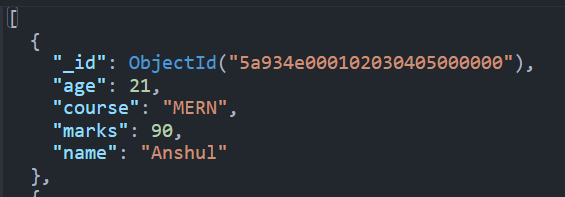
Result



1. Increase marks by 5 for all students in the MERN course

query

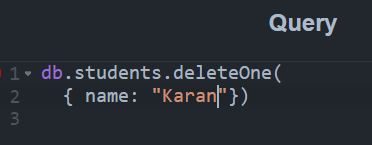
result



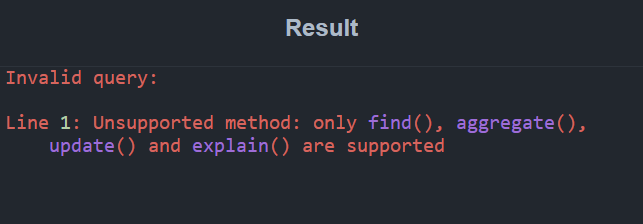
Q5: Delete Operations

1. Delete the student named Karan

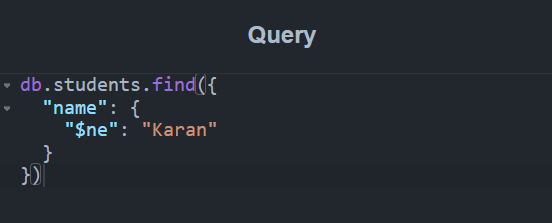
query



result



Using find()

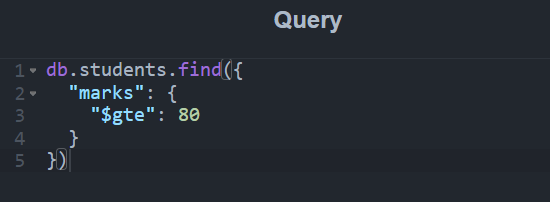
query

result



2.Delete all students with marks less than 80

query



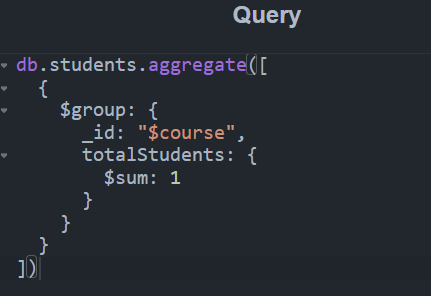
Result



Q6: Advanced Queries

1. Count the number of students in each course

query

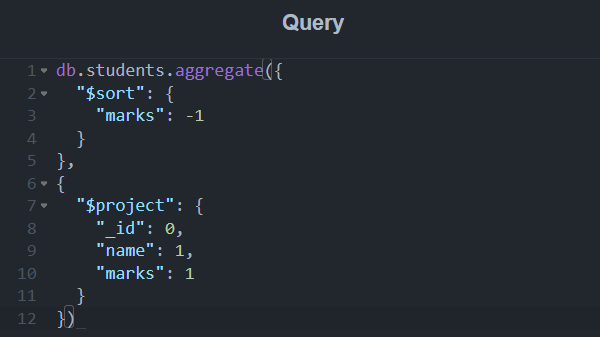


Result

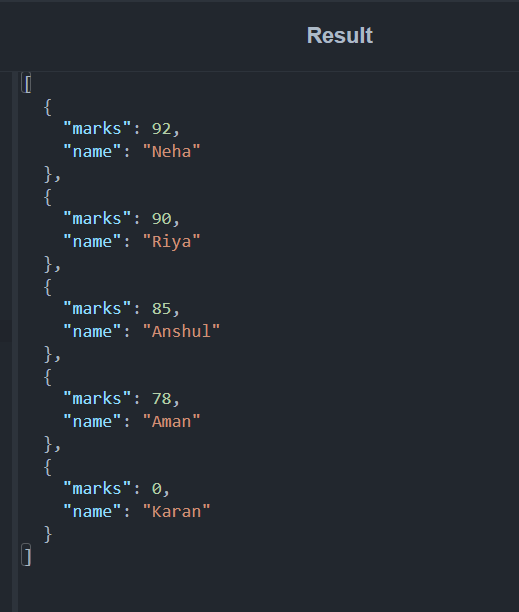


2.Sort students by marks in descending order

Query

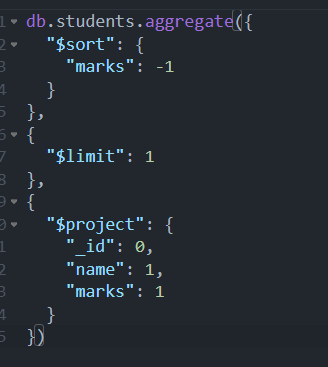


Result

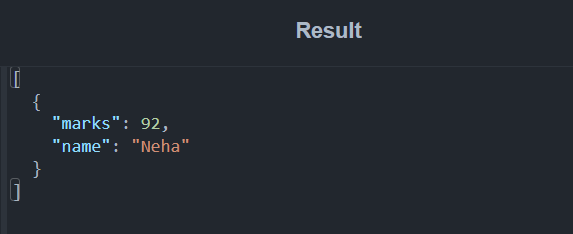


3.Find the student with the highest marks.

query

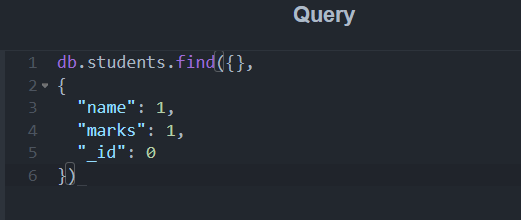


result



1. Display only name and marks of all students.

Query



Results

