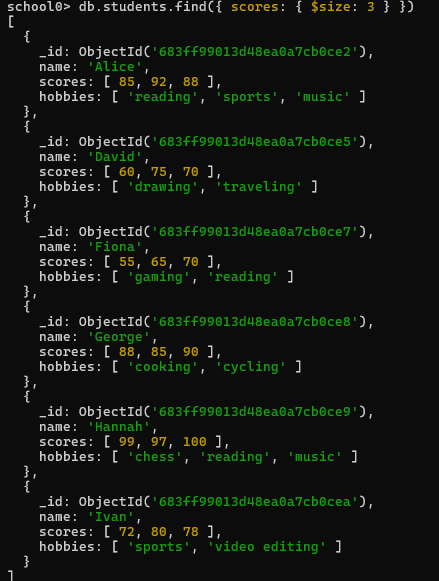
**Mongo DB**

**(Array Function)**

1. **Find students who have exactly 3 exam scores.**

Operator Used: $size

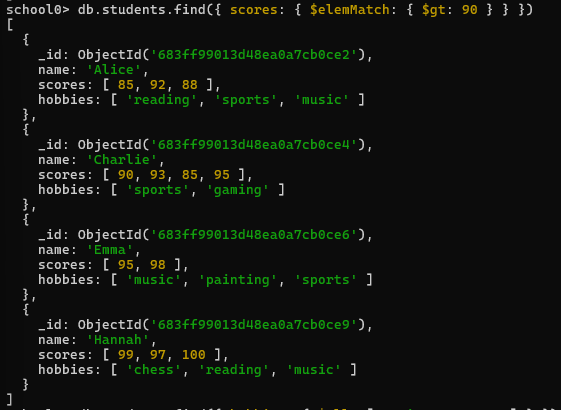
**db.students.find({ scores: { $size: 3 } })**

****

1. **Find students who scored above 90 in at least one exam.**

Operator Used: $elemMatch

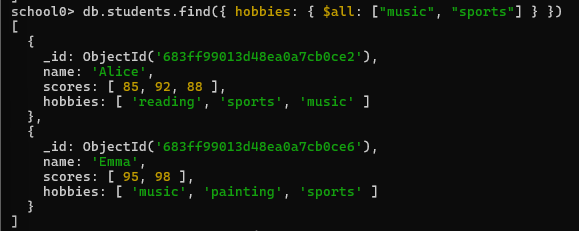
**db.students.find({ scores: { $elemMatch: { $gt: 90 } } })**

****

**3. Find students whose hobbies include both "music" and "sports".**

Operator Used: $all

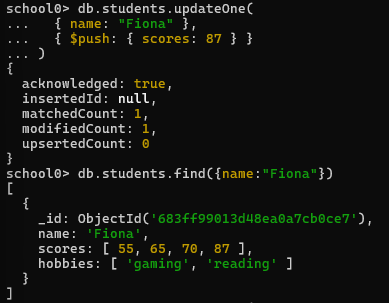
**db.students.find({ hobbies: { $all: ["music", "sports"] } })**

****

**4. Add a new score 87 to Fiona's scores array.**

Operator Used: $push

**db.students.updateOne({ name: "Fiona" },{ $push: { scores: 87 } })**

****

**5. Add multiple scores [90, 91] to Emma’s scores, sort them descending, and keep only top 3 scores.**

Operators Used: $push, $each, $sort, $slice

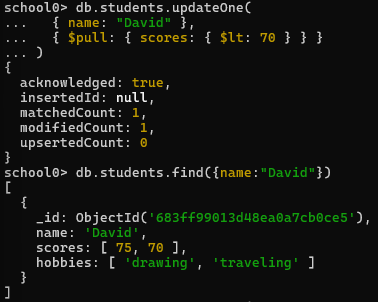
**db.students.updateOne({ name: "Emma" },{$push: {scores: {$each: [90,91],$sort: -1,$slice: 3}}})**

****

**6. Remove all scores less than 70 from David’s record.**

Operator Used: $pull

**db.students.updateOne({ name: "David" },{ $pull: { scores: { $lt: 70 } }})**

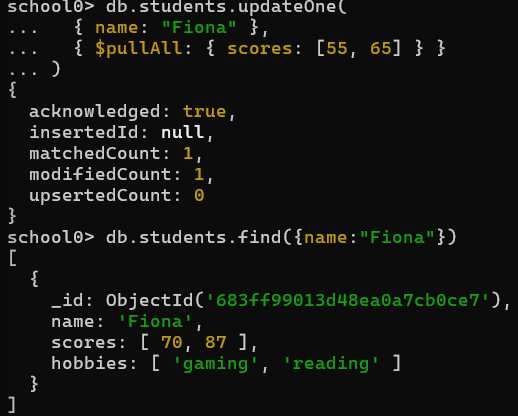
****

**7. Remove all scores 55 and 65 from Fiona’s scores.**

Operator Used: $pullAll

**db.students.updateOne({ name: "Fiona" },**

**{ $pullAll: { scores: [55, 65] } })**

****

**8. Project only the first hobby of each student.**

Operator Used: $slice (projection)

**db.students.find({}, { name: 1, hobbies: { $slice: 1 } })**

****

**9. Project only the hobby "music" if it exists.**

Operator Used: $elemMatch (projection)

**db.students.find({ hobbies: "music" },**

**{ name: 1, hobbies: { $elemMatch: { $eq: "music" } } })**

****

**10. Use aggregation to show each student's number of exam scores.**

Operator Used: $size

**db.students.aggregate([{$project: {name: 1,**

**numberOfScores: { $size: "$scores" }}}])**

****

**11. Show each student's highest score.**

Operator Used: $max

**db.students.aggregate([{$project: {name: 1,**

**highestScore: { $max: "$scores" }}}])**

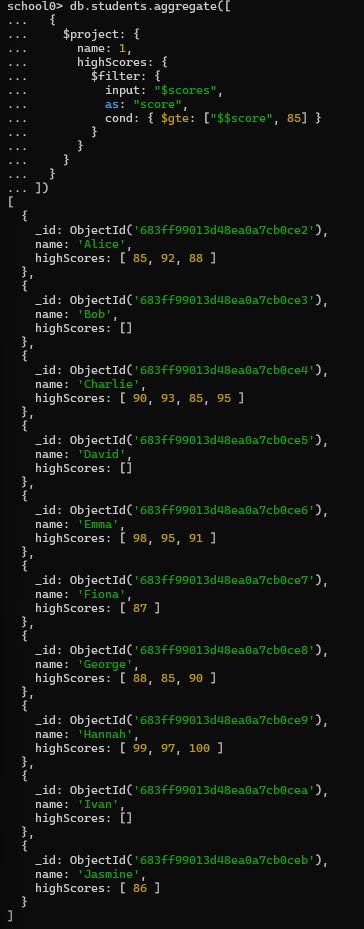
****

**12. Return all scores ≥ 85 for each student.**

Operator Used: $filter

**db.students.aggregate([$project: {name: 1,highScores: {$filter: {**

**input: "$scores",as: "score",cond: { $gte: ["$$score", 85] }}}}}])**

****

**13. Show score at index 1 for each student (2nd exam).**

Operator Used: $arrayElemAt

**db.students.aggregate([{$project: {name: 1,secondScore: { $arrayElemAt: ["$scores", 1] }}}])**

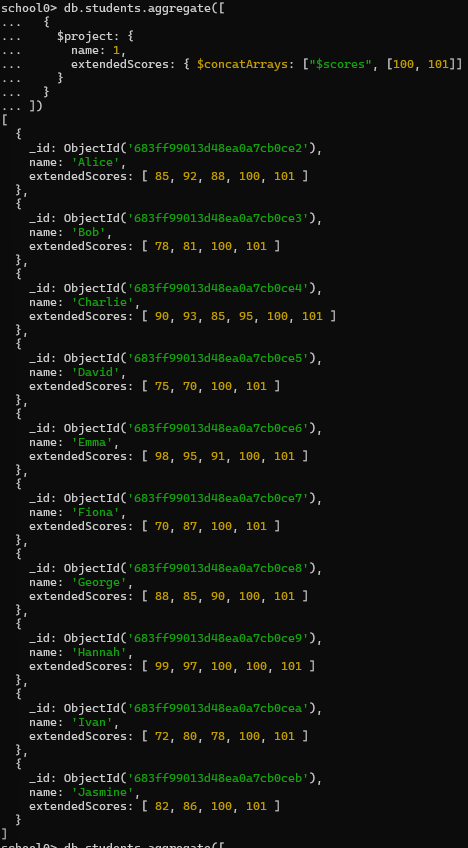
****

**14. Combine scores with a new array [100, 101].**

Operator Used: $concatArrays

**db.students.aggregate([{$project: {name: 1,**

**extendedScores: { $concatArrays: ["$scores", [100, 101]] }}}])**

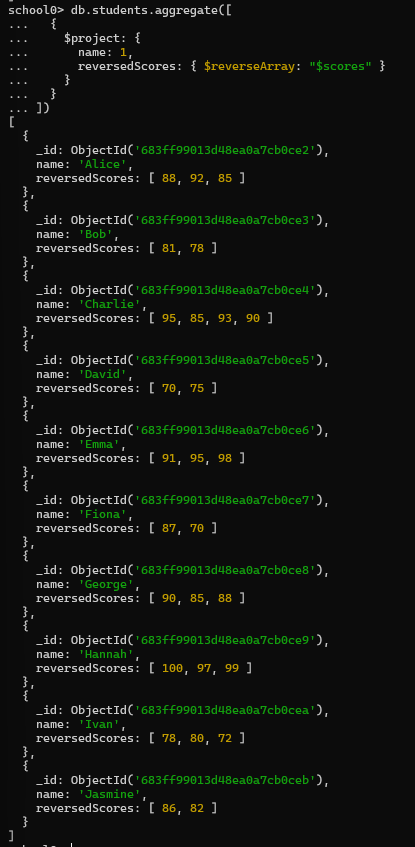
****

**15. Reverse the scores array for each student.**

Operator Used: $reverseArray

**db.students.aggregate([{$project: {name: 1,reversedScores:**

**{ $reverseArray: "$scores" }}}])**

****