

```
// use("test");

db.students.find();
// 1. Display the details of the students whose branch starts with "C" or "M"
// and have at least 2 backlogs in 1st semester

db.students.find({
    Branch: { $regex: "^(C|M)", $options: "i" },
    Semester: 1,
    "Backlogs.1": { $gte: 2 }
});

// 2. display studentid,branch, and CoCurrAct of the students who are having
// exactly 3 co-curricular activities
// or 2nd co-curricular activity as Coding Club

db.students.find({
    $or: [{ "CoCurrAct": { $size: 3 } }, { "CoCurrAct.1": "Coding Club" }] // 2nd
item means at 1st index
}, // here we can use $expr(it allows to use aggregation expressions in normal
find)
//     $or: [
//         { $expr: { $eq: [{ $size: "$CoCurrAct" }, 3] } },
//         {"CoCurrAct.1": "Coding Club"}
// ]
{ _id: 0, StudentID: 1, Branch: 1, CoCurrAct: 1 });

// 3. display branch and average CGPA as "avgCGPA"(column alias) for branches other
// than "Civil Engineering" and "Computer Science" having
// an average CGPA more than 7 , arrange the records in reverse alphabetical order
// of Branch

db.students.aggregate([
    { $match: { Branch: { $nin: ["Civil Engineering", "Computer Science"] } } },
    { $group: { _id: "$Branch", avgCGPA: { $avg: "$CGPA" } } },
    { $match: { avgCGPA: { $gt: 7 } } },
    { $sort: { _id: -1 } }
]);

// 4. display studentid and total number of co-curricular activities as
// "NumOfCoCurraAct"(column alias),
// for each student having more than 2 co-curricular activities
// Average the records in increasing order of studentID

db.students.aggregate([
    {
        $project: {
            _id: "$StudentID",
            NumOfCoCurraAct: { $size: "$CoCurrAct" }
        }
    },
    {
        $group: {
            _id: "$StudentID",
            NumOfCoCurraAct: { $sum: 1 }
        }
    }
]);
```

```
{ $match: { NumOfCoCurraAct: { $gt: 2 } } },
{ $sort: { _id: 1 } }
]);
```