

SOLUTION FOR PROBLEM STATEMENT 4:

mongo

```
// Use or create the 'university' database
use university;

// Drop the collection if it already exists (for clean setup)
db.Student_Data.drop();

// Create the 'Student_Data' collection and insert sample data
db.Student_Data.insertMany([
  { Student_ID: 1, Student_Name: "Alice", Department: "CS", Marks: 85 },
  { Student_ID: 2, Student_Name: "Bob", Department: "CS", Marks: 90 },
  { Student_ID: 3, Student_Name: "Charlie", Department: "Math", Marks: 88 },
  { Student_ID: 4, Student_Name: "David", Department: "Math", Marks: 92 },
  { Student_ID: 5, Student_Name: "Eve", Department: "Physics", Marks: 95 },
  { Student_ID: 6, Student_Name: "Frank", Department: "Physics", Marks: 87 }
]);

// 1. Display all students based on their departments along with the average marks
// of each department
print("1. Display all students with their departments and the average marks per
department:");
db.Student_Data.aggregate([
  { $group: { _id: "$Department", avgMarks: { $avg: "$Marks" }, students:
{ $push: "$$ROOT" } } }
]).forEach(printjson);

// 2. Display the number of students associated with each department
print("2. Display the number of students associated with each department:");
db.Student_Data.aggregate([
  { $group: { _id: "$Department", numOfStudents: { $sum: 1 } } }
]).forEach(printjson);

// 3. Display a list of students with the highest marks in each department, in
// descending order of marks
print("3. Display students with the highest marks in each department, in descending
order of marks:");
db.Student_Data.aggregate([
  { $sort: { Department: 1, Marks: -1 } },
  { $group: { _id: "$Department", highestMarks: { $first: "$Marks" }, topStudent:
{ $first: "$$ROOT" } } },
  { $replaceRoot: { newRoot: "$topStudent" } },
  { $sort: { Marks: -1 } }
]).forEach(printjson);

// 4. Create an index on the Student_ID field
print("4. Creating index on field Student_ID:");
db.Student_Data.createIndex({ Student_ID: 1 });

// 5. Create a compound index on Student_Name and Department fields
print("5. Creating compound index on fields Student_Name and Department:");
db.Student_Data.createIndex({ Student_Name: 1, Department: 1 });
```

```
// 6. Drop the index on the Student_ID field
print("6. Dropping index on field Student_ID:");
db.Student_Data.dropIndex({ Student_ID: 1 });

// 7. Drop the compound index on Student_Name and Department fields
print("7. Dropping compound index on fields Student_Name and Department:");
db.Student_Data.dropIndex({ Student_Name: 1, Department: 1 });
```