MongoDB - Aggregation and Indexing: Design and Develop MongoDB Queries using aggregation and indexing with suitable example using MongoDB.

Create collection student{ Rollno ,Name, Class, Div, Subject, Marks, Address} and enter 6 entries or more. And perform the following:

1. Find the maximum marks of student in DS who stay in the same city
   Db.student.aggreged(
   $match:{subject:"ds"}
   $group:{_id:"address.city",maxmarks:{$max:"$marks"}})
2. Calculates the average of given marks.
   db.student.aggregate([
     {
       $group: {
         _id: null,
         avgMarks: { $avg: "$Marks" }
       }
     }
   ]);
3. Inserts the any value to an array in the resulting document.
   db.student.aggregate([
     {
       $group: {
         _id: "$Div",
         students: { $push: "$Name" }
       }
     }
   ]);
4. Create a compound index on name and class.
db.student.createIndex({ Name: 1, Class: 1 });
5. Create single index on name.
   db.student.createIndex({ Name: 1 });
6. Delete index of name.
   db.student.dropIndex({ Name: 1 });

ANS =
Step 1: Create the "student" collection and insert sample data

javascript

Copy code

db.student.insertMany([

 { Rollno: 1, Name: "Aman", Class: "TE", Div: "A", Subject: "DS", Marks: 85, Address: { City: "Pune", State: "MH" }},

 { Rollno: 2, Name: "Bhavna", Class: "TE", Div: "B", Subject: "DBMS", Marks: 72, Address: { City: "Mumbai", State: "MH" }},

{ Rollno: 3, Name: "Chetan", Class: "TE", Div: "A", Subject: "SPOS", Marks: 90, Address: { City: "Pune", State: "MH" } },

{ Rollno: 4, Name: "Divya", Class: "TE", Div: "B", Subject: "DS", Marks: 60, Address: { City: "Bangalore", State: "KA" } },

{ Rollno: 5, Name: "Ekta", Class: "TE", Div: "C", Subject: "TOC", Marks: 78, Address: { City: "Pune", State: "MH" } },

{ Rollno: 6, Name: "Farhan", Class: "TE", Div: "B", Subject: "DS", Marks: 65, Address: { City: "Pune", State: "MH" } }

]);

1. Find the maximum marks of students in "DS" who stay in the same city.

javascript

Copy code

```
db.student.aggregate([

 { $match: { Subject: "DS" } },

 { $group: { _id: "$Address.City", maxMarks: { $max: "$Marks" } } }

]);
```

2. Calculate the average of marks for all students.

javascript

Copy code

```
db.student.aggregate([

 {

  $group: {

   _id: null,

   avgMarks: { $avg: "$Marks" }

  }

 }

]);
```

3. Insert names into an array grouped by division.

javascript

Copy code

```
db.student.aggregate([

 {
```

```
   $group: {
     _id: "$Div",
     students: { $push: "$Name" }
   }
 }
]);
```

4. Create a compound index on Name and Class.

javascript

Copy code

```
db.student.createIndex({ Name: 1, Class: 1 });
```

5. Create a single index on Name.

javascript

Copy code

```
db.student.createIndex({ Name: 1 });
```

6. Delete the index on Name.

javascript

Copy code

```
db.student.dropIndex({ Name: 1 });
```