

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 average of total marks in TOC.  

```
db.student.aggregate([
  { $match: { Subject: "TOC" } },
  { $group: { _id: null, avgMarks: { $avg: "$Marks" } } }
]);
```
2. Find the number of students division wise.  

```
db.student.aggregate([
  { $group: { _id: "$Div", count: { $sum: 1 } } }
]);
```
3. Find students of div B who scored min marks in DBMS.  

```
db.student.aggregate([
  { $match: { Div: "B", Subject: "DBMS" } },
  { $group: { _id: null, minMarks: { $min: "$Marks" } } }
]);
```
4. Find total sum of marks in SPOS of students staying pune  

```
db.student.aggregate([
  { $match: { Subject: "SPOS", "Address.City": "Pune" } },
  { $group: { _id: null, totalMarks: { $sum: "$Marks" } } }
]);
```
5. Find the division wise count of students whose DBMS marks.  

```
db.student.aggregate([
  { $match: { Subject: "DBMS" } },
  { $group: { _id: "$Div", count: { $sum: 1 } } }
]);
```
6. Find the count of each city.  

```
db.student.aggregate([
  { $group: { _id: "$Address.City", count: { $sum: 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: "TOC", 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: "DBMS", 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: "SPOS", Marks: 65, Address: { City: "Pune", State: "MH" } }
```

```
]);
```

### **1. Find the average marks of students in the subject "TOC".**

javascript

Copy code

```
db.student.aggregate([
  { $match: { Subject: "TOC" } },
  { $group: { _id: null, avgMarks: { $avg: "$Marks" } } }
]);
```

### **2. Find the number of students in each division.**

javascript

Copy code

```
db.student.aggregate([
  { $group: { _id: "$Div", count: { $sum: 1 } } }
]);
```

### **3. Find the student(s) of division "B" who scored the minimum marks in "DBMS".**

javascript

Copy code

```
db.student.aggregate([
  { $match: { Div: "B", Subject: "DBMS" } },
  { $group: { _id: null, minMarks: { $min: "$Marks" } } }
]);
```

### **4. Find the total sum of marks in "SPOS" for students staying in Pune.**

javascript

Copy code

```
db.student.aggregate([
  { $match: { Subject: "SPOS", "Address.City": "Pune" } },
  { $group: { _id: null, totalMarks: { $sum: "$Marks" } } }
])
```

**5. Find the division-wise count of students in the "DBMS" subject.**

javascript

Copy code

```
db.student.aggregate([
  { $match: { Subject: "DBMS" } },
  { $group: { _id: "$Div", count: { $sum: 1 } } }
])
```

**6. Find the count of students in each city.**

javascript

Copy code

```
db.student.aggregate([
  { $group: { _id: "$Address.City", count: { $sum: 1 } } }
])
```

### Indexing Example

To optimize the above queries, especially those involving matching by subject or city, you can create indexes. For example, you can create indexes on the "Subject" and "Address.City" fields:

javascript

Copy code

```
db.student.createIndex({ Subject: 1 });
db.student.createIndex({ "Address.City": 1 });
```