Programming Lab 3

Assignment 08

Operations on MongoDB

Name : Ayush Sachin Doshi

PRN : 21510120

Batch : T-1

Problem Statement 01:

Create Database collection and do the following operations:

o Group by a Single Field in MongoDB.

o Group by Multiple Fields in MongoDB

o Group by the Multiple Expressions

o Group by the Conditional Statements in MongoDB

o Group by a Nested Field in MongoDB

Ans :

Commands given to Mongosh:

1]use Student

2]db.createCollection("student")

3]db.insertMany([{ id:1,firstname:"Candice"

])Insert command.

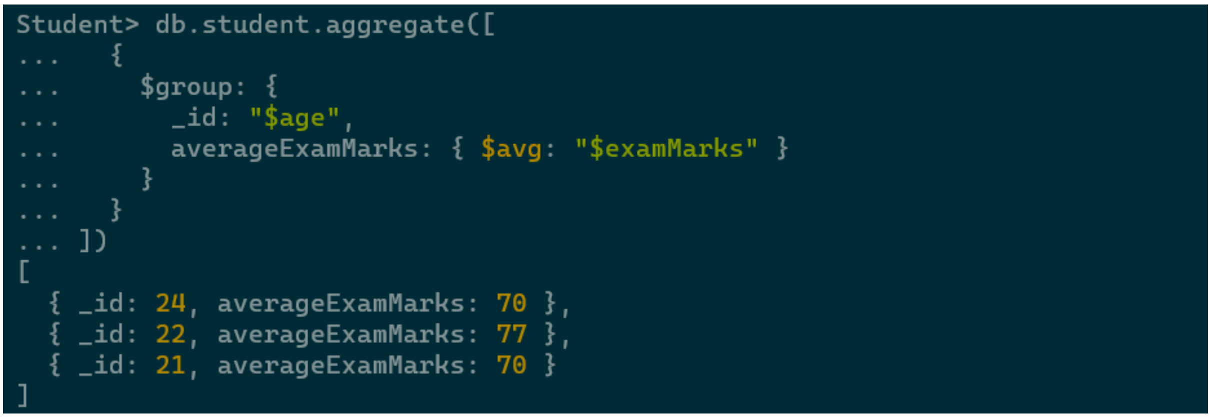
Operations:

1]Group by single field:

Command:

```
db.student.aggregate([
 {
   $group: {
     _id: "$age",
     averageExamMarks: { $avg: "$examMarks" }
   }
 }
])
```

Output:



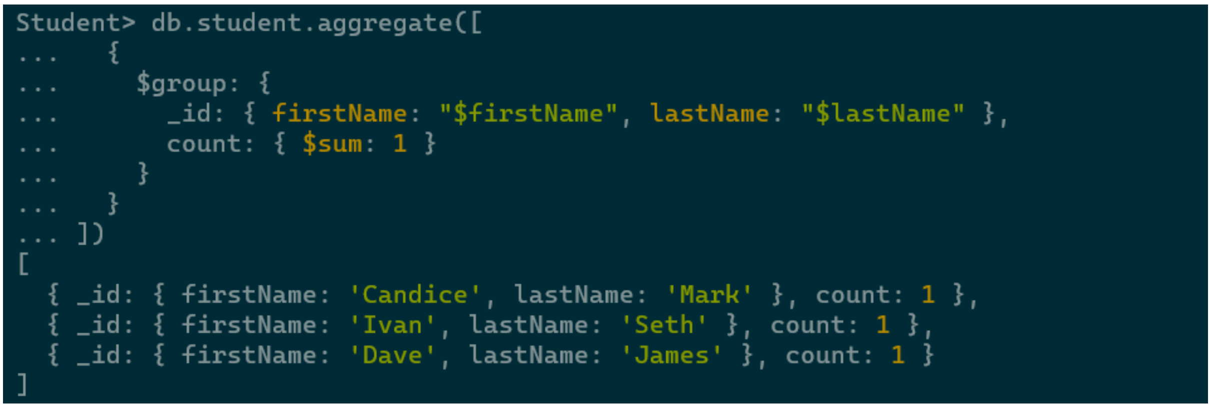
```
Student> db.student.aggregate([
...   {
...     $group: {
...       _id: "$age",
...       averageExamMarks: { $avg: "$examMarks" }
...     }
...   }
... ])
[
  { _id: 24, averageExamMarks: 70 },
  { _id: 22, averageExamMarks: 77 },
  { _id: 21, averageExamMarks: 70 }
]
```

1]Group by multiple fields:

Command:

```
db.student.aggregate([
 {
   $group: {
     _id: { firstName: "$firstName", lastName: "$lastName" },
     count: { $sum: 1 }
   }
 }
])db.student.aggregate([
 {
   $group: {
     _id: "$age",
     averageExamMarks: { $avg: "$examMarks" }
   }
 }
])
```

Output:



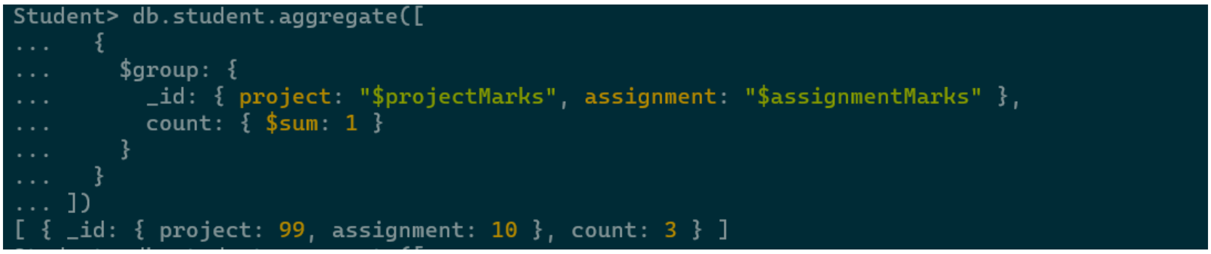
```
Student> db.student.aggregate([
...   {
...     $group: {
...       _id: { firstName: "$firstName", lastName: "$lastName" },
...       count: { $sum: 1 }
...     }
...   }
... ])
[
  { _id: { firstName: 'Candice', lastName: 'Mark' }, count: 1 },
  { _id: { firstName: 'Ivan', lastName: 'Seth' }, count: 1 },
  { _id: { firstName: 'Dave', lastName: 'James' }, count: 1 }
]
```

1]Group by Multiple Expressions:

Command:

```
db.student.aggregate([
 {
   $group: {
     _id: { project: "$projectMarks", assignment: "$assignmentMarks" },
     count: { $sum: 1 }
   }
 }
])db.student.aggregate([
 {
   $group: {
     _id: "$age
",
     averageExamMarks: { $avg: "$examMarks" }
   }
 }
])
```

Output:



```
Student> db.student.aggregate([
...   {
...     $group: {
...       _id: { project: "$projectMarks", assignment: "$assignmentMarks" },
...       count: { $sum: 1 }
...     }
...   }
... ])
[ { _id: { project: 99, assignment: 10 }, count: 3 } ]
```

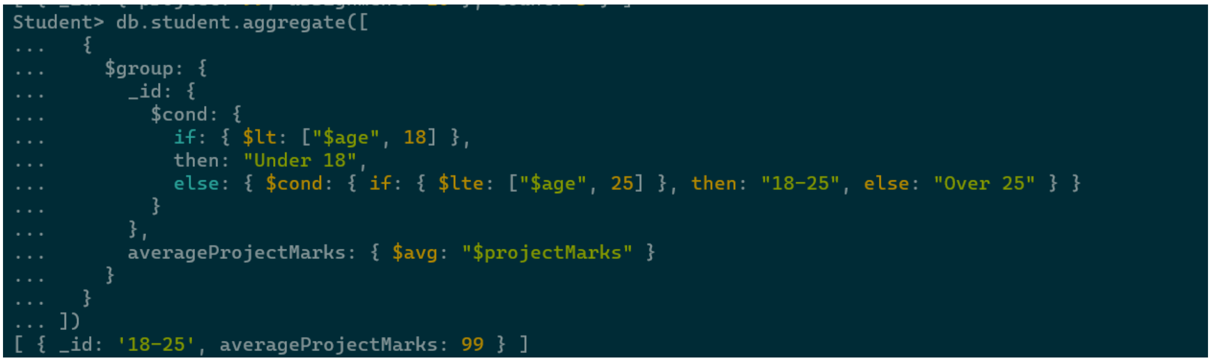
1]Group by conditional statements:

Command:

db.student.aggregate([

```
 {
  $group: {
   _id: {
    $cond: {
     if: { $lt: ["$age", 18] },
     then: "Under 18",
     else: { $cond: { if: { $lte: ["$age", 25] }, then: "18-25", else: "Over 25" } }
    }
   },
   averageProjectMarks: { $avg: "$projectMarks" }
  }
 }
])db.student.aggregate([
 {
  $group: {
   _id: "$age",
   averageExamMarks: { $avg: "$examMarks" }
  }
 }
])
```

Output:



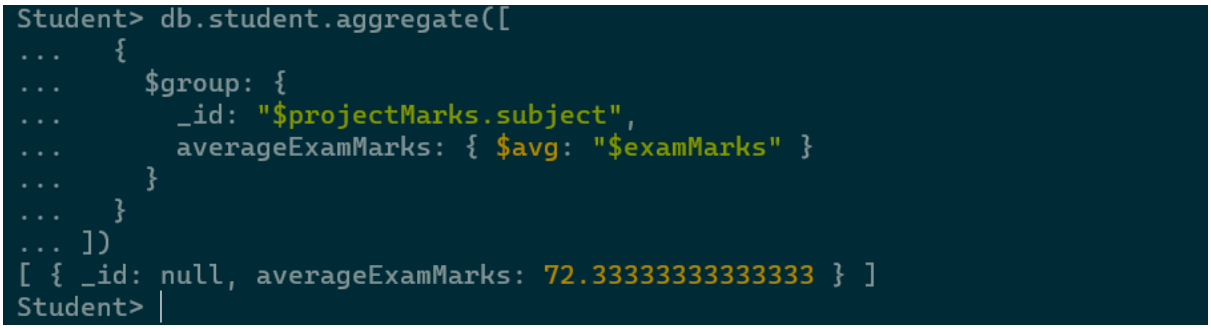
```
Student> db.student.aggregate([
...   {
...     $group: {
...       _id: {
...         $cond: {
...           if: { $lt: ["$age", 18] },
...           then: "Under 18",
...           else: { $cond: { if: { $lte: ["$age", 25] }, then: "18-25", else: "Over 25" } }
...         }
...       },
...       averageProjectMarks: { $avg: "$projectMarks" }
...     }
...   }
... ])
[ { _id: '18-25', averageProjectMarks: 99 } ]
```

1]Group by Nested field:

Command:

```
db.student.aggregate([
 {
   $group: {
     _id: "$projectMarks.subject",
     averageExamMarks: { $avg: "$examMarks" }
   }
 }
])
```

Output:



```
Student> db.student.aggregate([
...   {
...     $group: {
...       _id: "$projectMarks.subject",
...       averageExamMarks: { $avg: "$examMarks" }
...     }
...   }
... ])
[ { _id: null, averageExamMarks: 72.33333333333333 } ]
Student> 
```

End of Assignment…