PROGRAMING LAB 3

ASSIGNMENT NO. – 8

NAME : AKASH RAKESH METKARI.

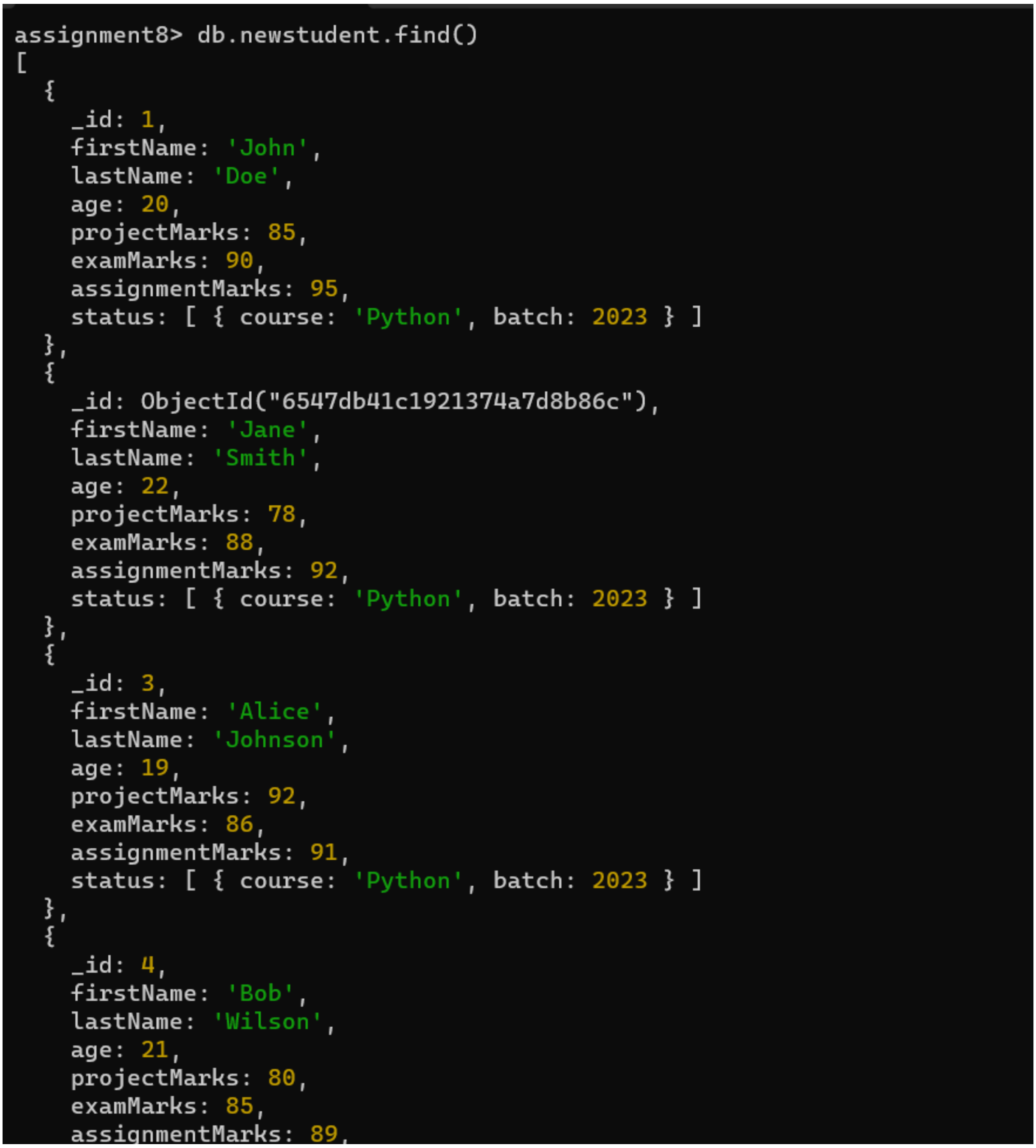
PRN : 22520012.

BATCH : T3.

Study of MongoDB

Creating a database :

```
db.newstudent.insertMany([
  {
    "_id": 1,
    "firstName": "John",
    "lastName": "Doe",
    "age": 20,
    "projectMarks": 85,
    "examMarks": 90,
    "assignmentMarks": 95,
    "status": [
      { "course": "Python", "batch": 2023 }
    ]
  },
  {
    "_id": 2,
    "firstName": "Jane",
    "lastName": "Smith",
    "age": 22,
  "projectMarks": 78,
  "examMarks": 88,
  "assignmentMarks": 92,
  "status": [
    { "course": "Python", "batch": 2023 }
  ]
},
{
  "_id": 3,
  "firstName": "Alice",
  "lastName": "Johnson",
  "age": 19,
  "projectMarks": 92,
  "examMarks": 86,
  "assignmentMarks": 91,
  "status": [
    { "course": "Python", "batch": 2023 }
  ]
},
{
  "_id": 4,
  "firstName": "Bob",
  "lastName": "Wilson",
  "age": 21,
  "projectMarks": 80,
  "examMarks": 85,
  "assignmentMarks": 89,
    "status": [
      { "course": "Python", "batch": 2023 }
    ]
  },
  {
    "_id": 5,
    "firstName": "Eva",
    "lastName": "Brown",
    "age": 23,
    "projectMarks": 88,
    "examMarks": 92,
    "assignmentMarks": 87,
    "status": [
      { "course": "Python", "batch": 2023 }
    ]
  }
]);
```



```
assignment8> db.newstudent.find()
[
  {
    _id: 1,
    firstName: 'John',
    lastName: 'Doe',
    age: 20,
    projectMarks: 85,
    examMarks: 90,
    assignmentMarks: 95,
    status: [ { course: 'Python', batch: 2023 } ]
  },
  {
    _id: ObjectId("6547db41c1921374a7d8b86c"),
    firstName: 'Jane',
    lastName: 'Smith',
    age: 22,
    projectMarks: 78,
    examMarks: 88,
    assignmentMarks: 92,
    status: [ { course: 'Python', batch: 2023 } ]
  },
  {
    _id: 3,
    firstName: 'Alice',
    lastName: 'Johnson',
    age: 19,
    projectMarks: 92,
    examMarks: 86,
    assignmentMarks: 91,
    status: [ { course: 'Python', batch: 2023 } ]
  },
  {
    _id: 4,
    firstName: 'Bob',
    lastName: 'Wilson',
    age: 21,
    projectMarks: 80,
    examMarks: 85,
    assignmentMarks: 89,
```

Performing following operations on it:

o Group by a Single Field in MongoDB.

o Group by Multiple Fields in MongoDB

o Group by the Multiple Expressions

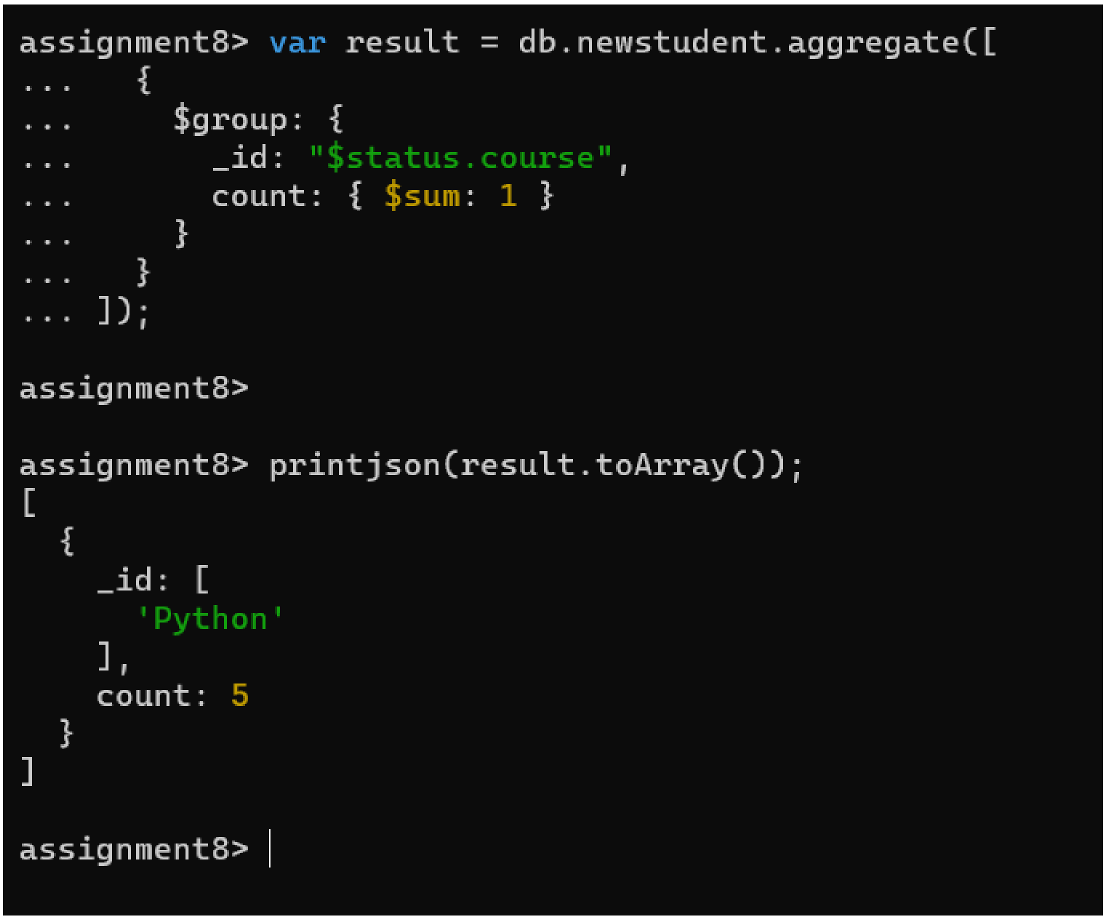
o Group by the Conditional Statements in MongoDB

o Group by a Nested Field in MongoDB

1) Group by a Single Field:

To group by a single field, you can use the $group stage with the _id key set to the field you want to group by. In this example, I'll group by the "course" field:

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

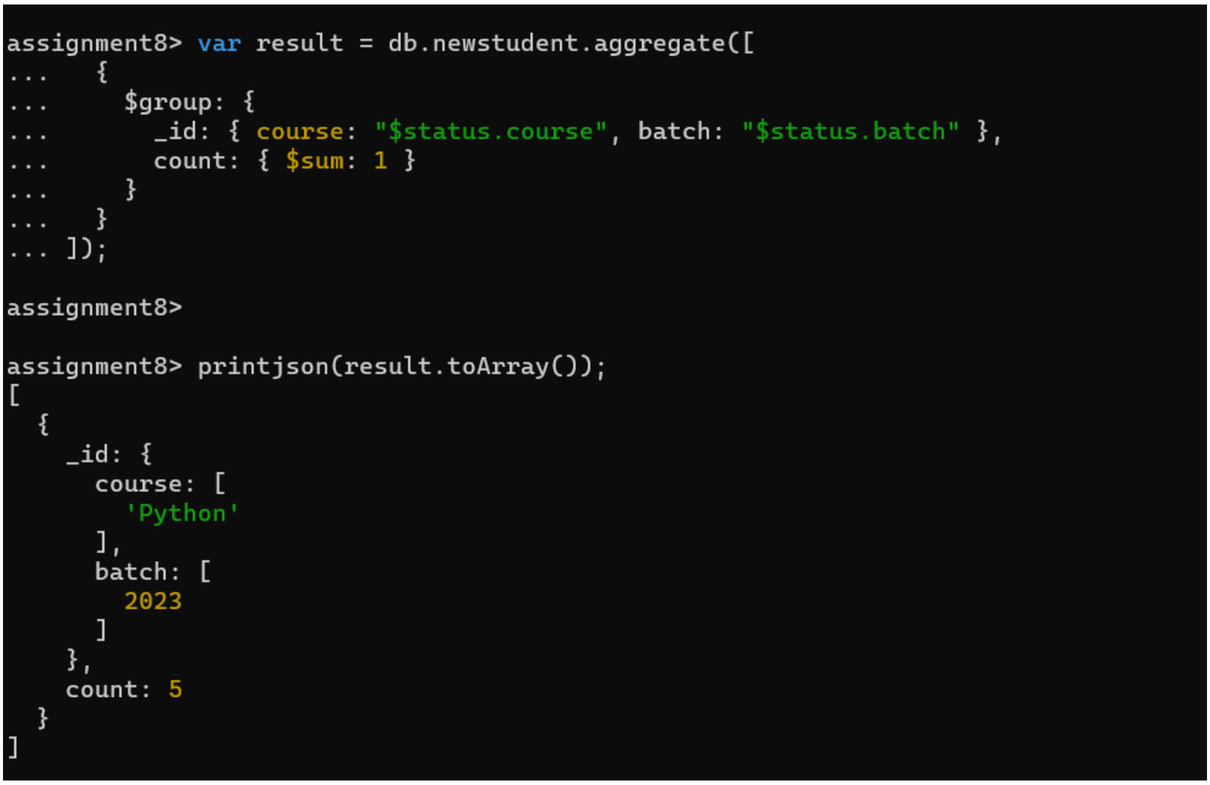


```
assignment8> var result = db.newstudent.aggregate([
...   {
...     $group: {
...       _id: "$status.course",
...       count: { $sum: 1 }
...     }
...   }
... ]);

assignment8>

assignment8> printjson(result.toArray());
[
  {
    _id: [
      'Python'
    ],
    count: 5
  }
]

assignment8> 
```

Group by Multiple Fields:

To group by multiple fields, set the _id key to an object containing those fields. Here, I'll group by "course" and "batch":

```
db.newstudent.aggregate([
 {
  $group: {
   _id: { course: "$status.course", batch: "$status.batch" },
   count: { $sum: 1 }
  }
 }
]);
```

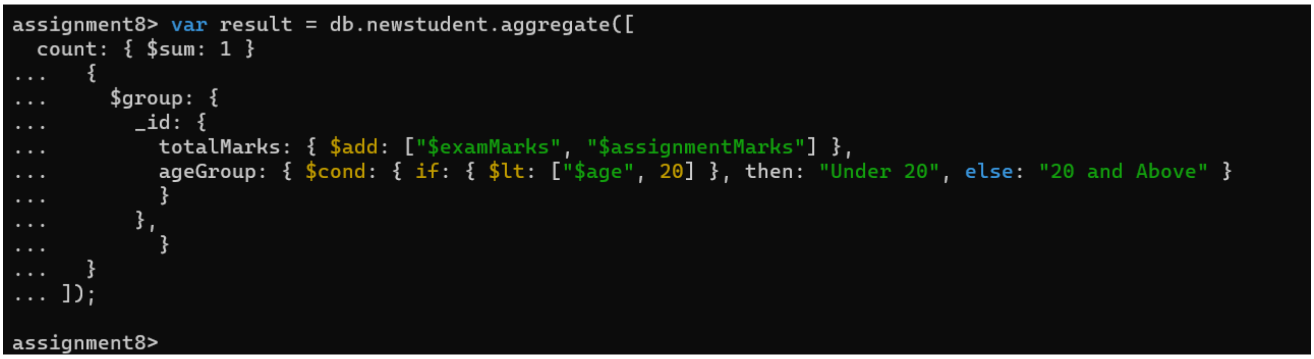


```
assignment8> var result = db.newstudent.aggregate([
...   {
...     $group: {
...       _id: { course: "$status.course", batch: "$status.batch" },
...       count: { $sum: 1 }
...     }
...   }
... ]);

assignment8>

assignment8> printjson(result.toArray());
[
  {
    _id: {
      course: [
        'Python'
      ],
      batch: [
        2023
      ]
    },
    count: 5
  }
]
```

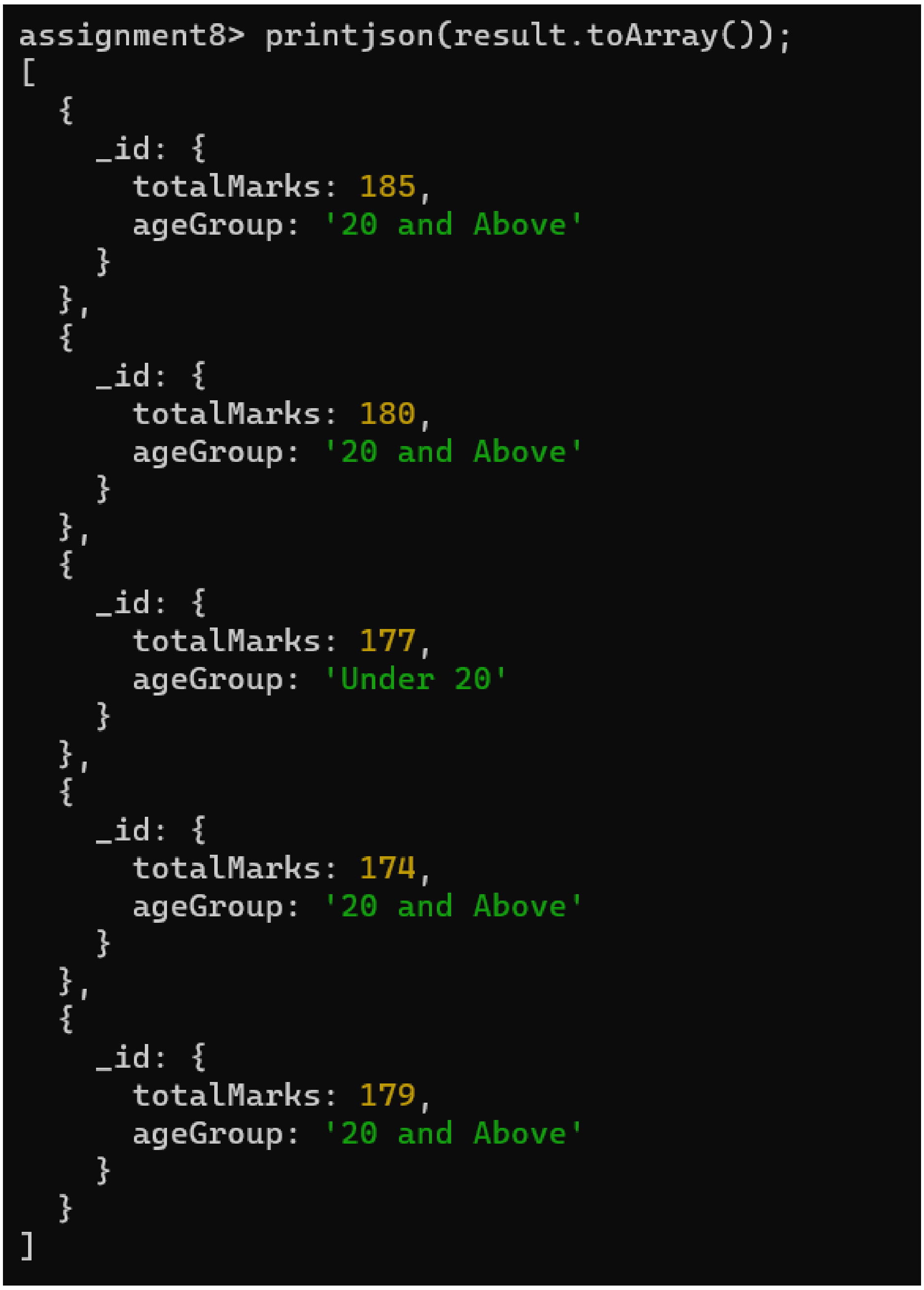
Group by Multiple Expressions:

You can group by the result of expressions. For example, you can group by the sum of "examMarks" and "assignmentMarks":

```
db.newstudent.aggregate([
  {
    $group: {
      _id: { totalMarks: { $add: ["$examMarks", "$assignmentMarks"] } },
      count: { $sum: 1 }
    }
  }
]);
```



```
assignment8> var result = db.newstudent.aggregate([
  count: { $sum: 1 }
...   {
...     $group: {
...       _id: {
...         totalMarks: { $add: ["$examMarks", "$assignmentMarks"] },
...         ageGroup: { $cond: { if: { $lt: ["$age", 20] }, then: "Under 20", else: "20 and Above" }
...         }
...       },
...         }
...   }
... ]);

assignment8>
```

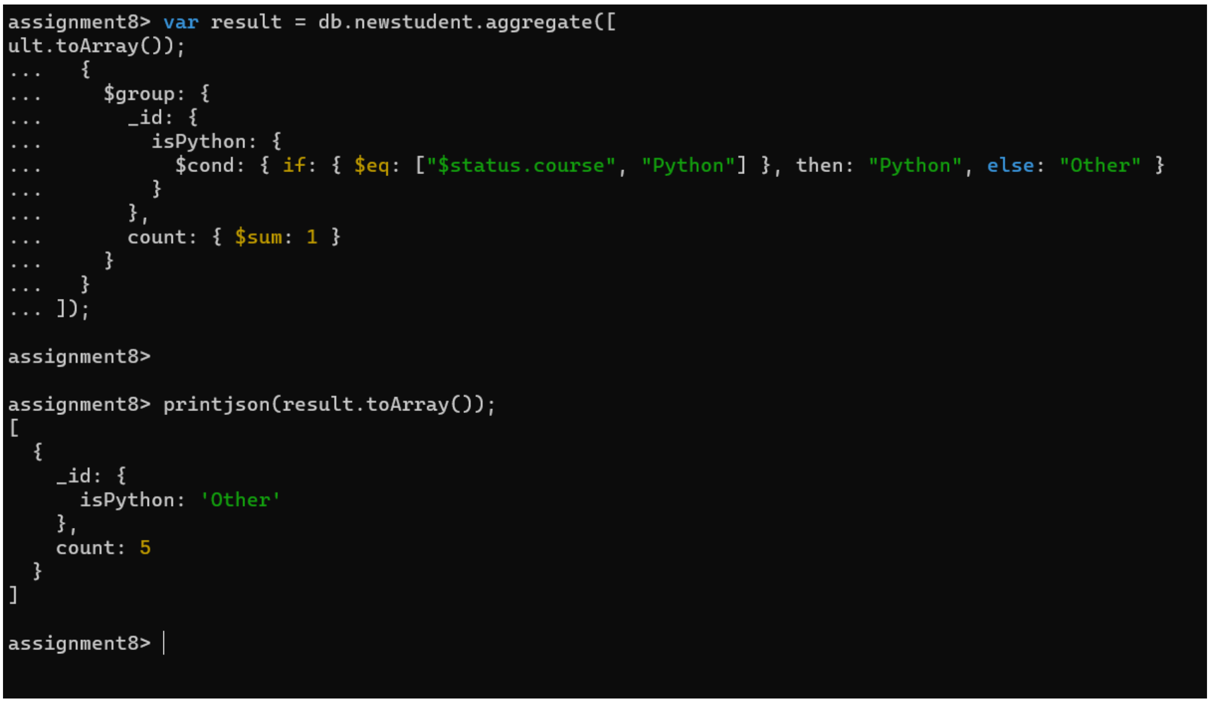


```
assignment8> printjson(result.toArray());
[
  {
    _id: {
      totalMarks: 185,
      ageGroup: '20 and Above'
    }
  },
  {
    _id: {
      totalMarks: 180,
      ageGroup: '20 and Above'
    }
  },
  {
    _id: {
      totalMarks: 177,
      ageGroup: 'Under 20'
    }
  },
  {
    _id: {
      totalMarks: 174,
      ageGroup: '20 and Above'
    }
  },
  {
    _id: {
      totalMarks: 179,
      ageGroup: '20 and Above'
    }
  }
]
```

Group by Conditional Statements:

You can group documents based on conditions. Here, I'll group by the "status" field where "course" is "Python":

```
db.newstudent.aggregate([
 {
   $group: {
     _id: {
       isPython: {
         $cond: { if: { $eq: ["$status.course", "Python"] }, then: "Python", else: "Other" }
       }
     },
     count: { $sum: 1 }
   }
 }
]);
```

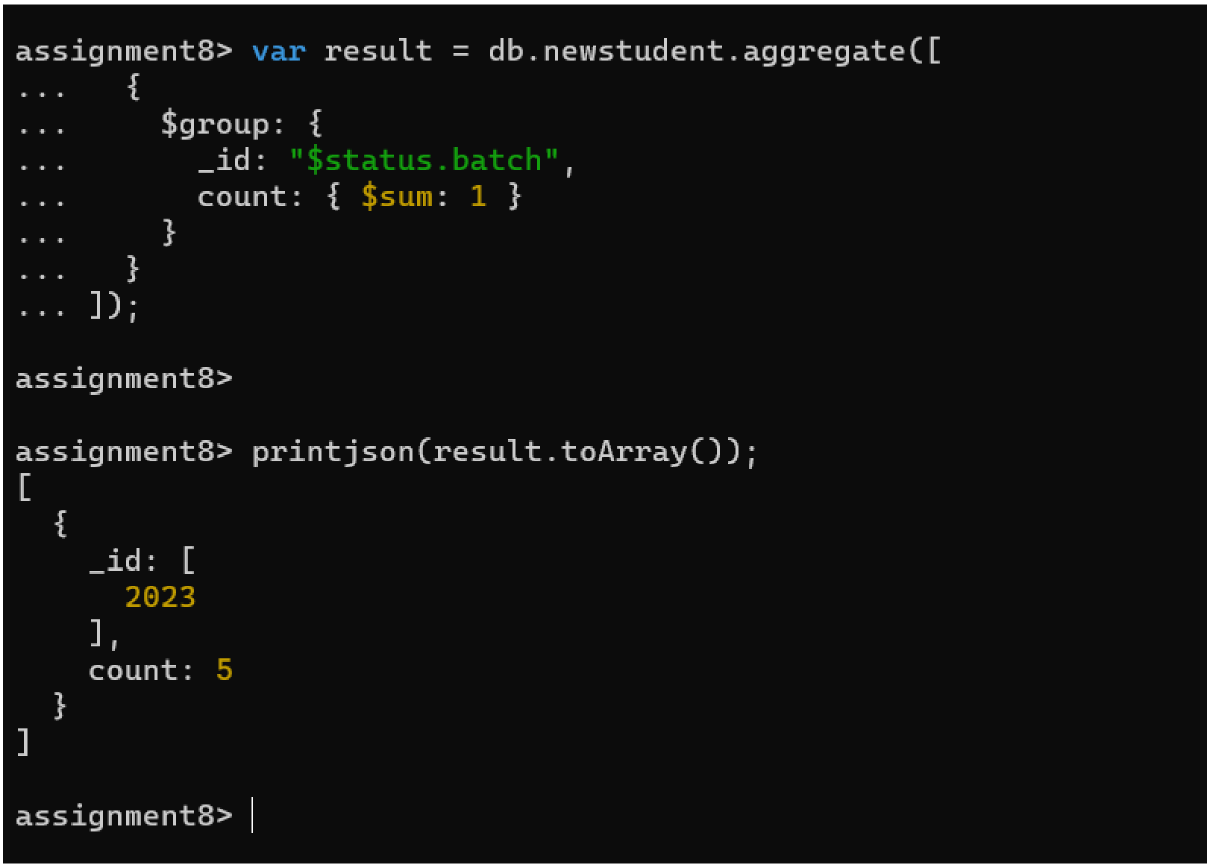


```
assignment8> var result = db.newstudent.aggregate([
ult.toArray());
...   {
...     $group: {
...       _id: {
...         isPython: {
...           $cond: { if: { $eq: ["$status.course", "Python"] }, then: "Python", else: "Other" }
...         }
...       },
...       count: { $sum: 1 }
...     }
...   }
... ]);

assignment8>

assignment8> printjson(result.toArray());
[
  {
    _id: {
      isPython: 'Other'
    },
    count: 5
  }
]

assignment8> |
```

Group by a Nested Field:

To group by a nested field, you can use dot notation. In this example, I'll group by the "batch" field within "status":

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



```
assignment8> var result = db.newstudent.aggregate([
...   {
...     $group: {
...       _id: "$status.batch",
...       count: { $sum: 1 }
...     }
...   }
... ]);

assignment8>

assignment8> printjson(result.toArray());
[
  {
    _id: [
      2023
    ],
    count: 5
  }
]

assignment8>
```