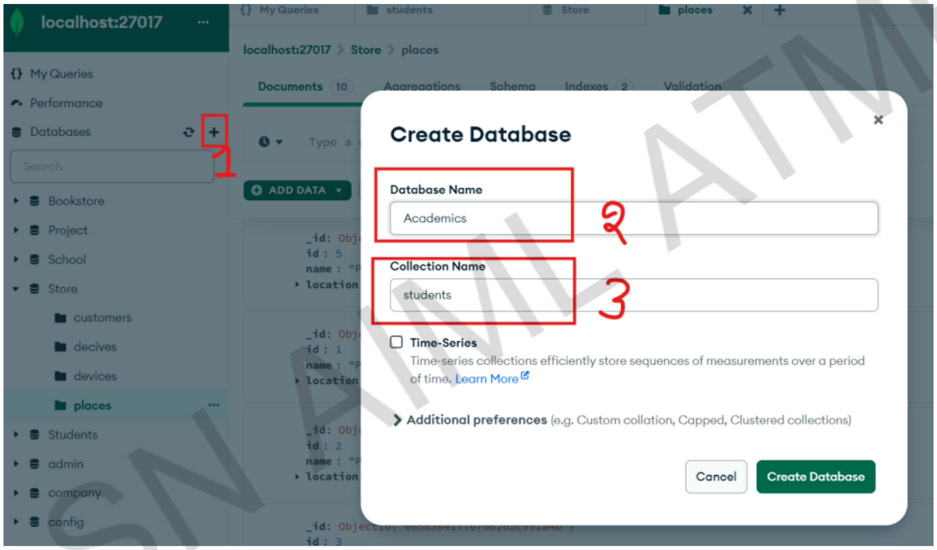
Program 5

Execute Aggregation operations ($avg, $min, $max, $push, $addToSet etc.). Encourage students to execute several queries to demonstrate various aggregation operators.

Create a database **Academics** and collection **students** in Mongo IDE.



Create Database

Database Name: Academics

Collection Name: students

☐ Time-Series

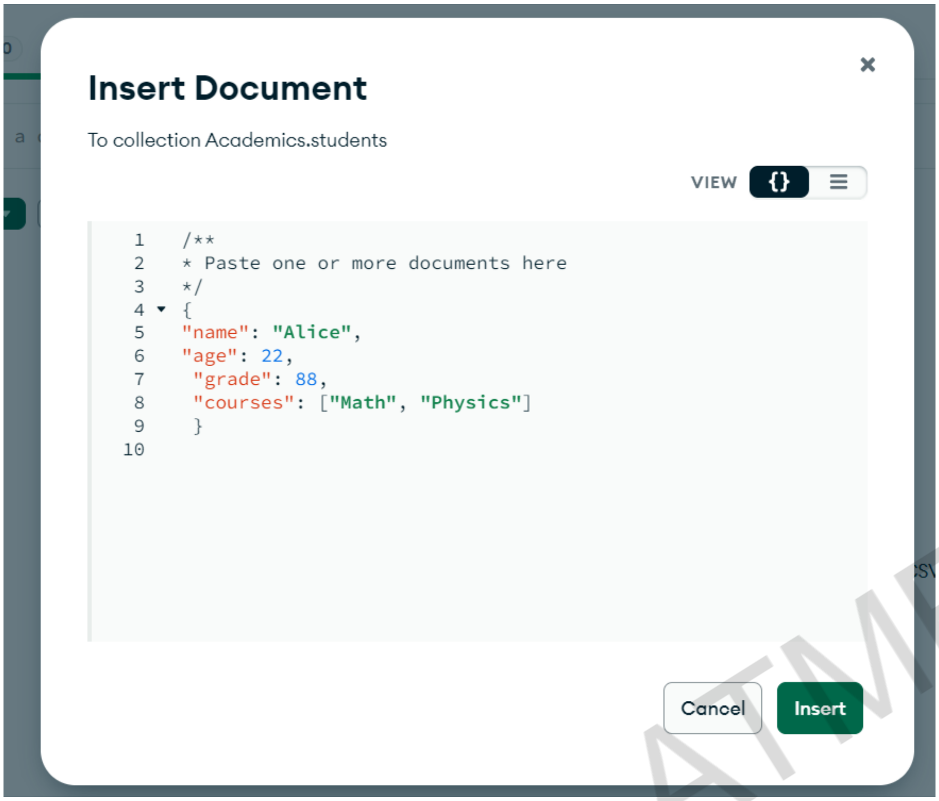
Time-series collections efficiently store sequences of measurements over a period of time. Learn More

Additional preferences (e.g. Custom collation, Capped, Clustered collections)

Cancel | Create Database

Add the following documents in the **students collection** in MongoDB IDE.

```
{
"name": "Alice",
"age": 22,
 "grade": 88,
 "courses": ["Math", "Physics"]
 }
```



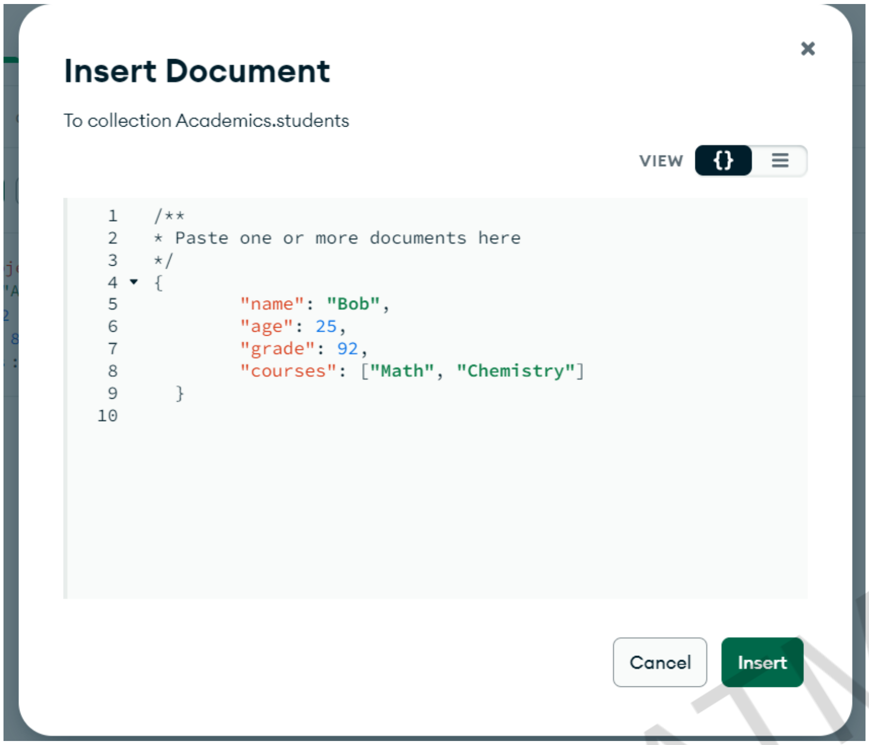
Insert Document

To collection Academics.students

```
/**
* Paste one or more documents here
*/
{
"name": "Alice",
"age": 22,
 "grade": 88,
 "courses": ["Math", "Physics"]
 }

```

```
{
"name": "Bob",
"age": 25,
"grade": 92,
"courses": ["Math", "Chemistry"]
}
```

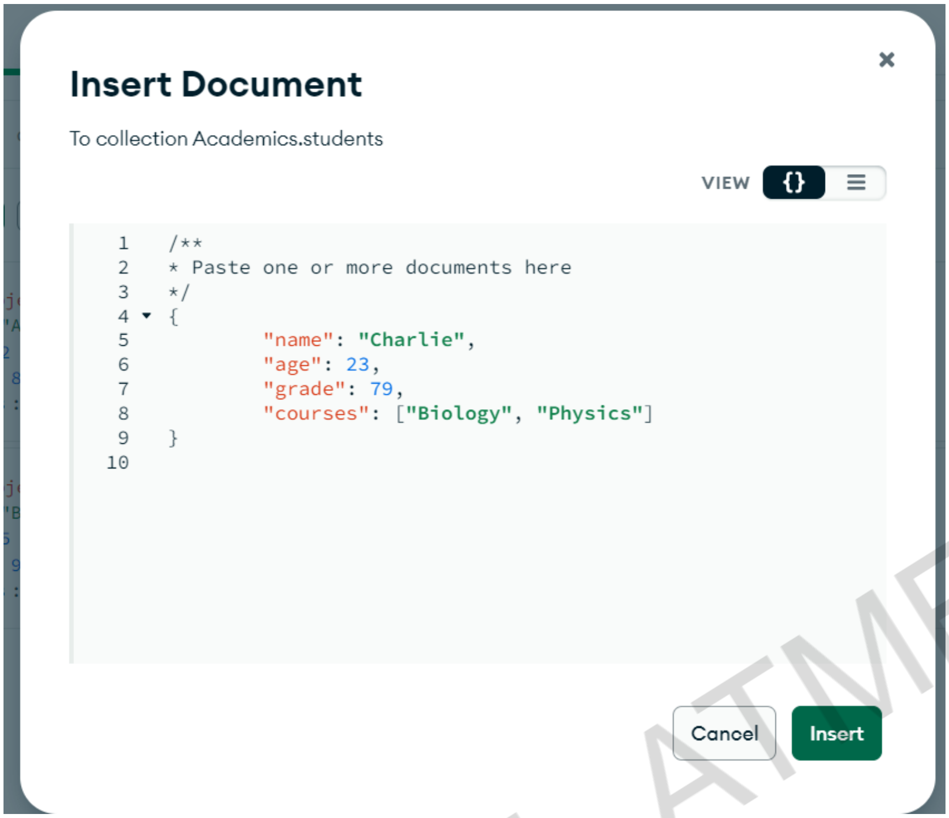


Insert Document

To collection Academics.students

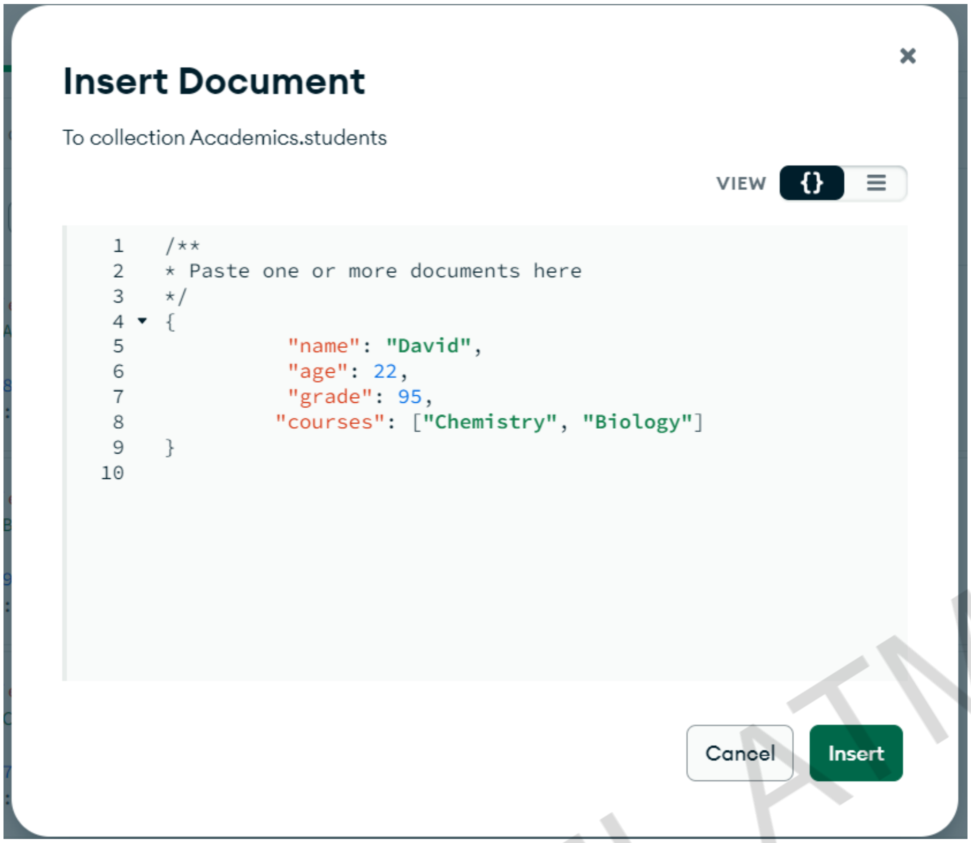
```
/**
 * Paste one or more documents here
 */
{
        "name": "Bob",
        "age": 25,
        "grade": 92,
        "courses": ["Math", "Chemistry"]
  }
```

```
{
    "name": "Charlie",
    "age": 23,
    "grade": 79,
    "courses": ["Biology", "Physics"]
}
```



```
/**
* Paste one or more documents here
*/
{
        "name": "Charlie",
        "age": 23,
        "grade": 79,
        "courses": ["Biology", "Physics"]
}
```

```
{
  "name": "David",
"age": 22,
"grade": 95,
 "courses": ["Chemistry", "Biology"]
 }
```

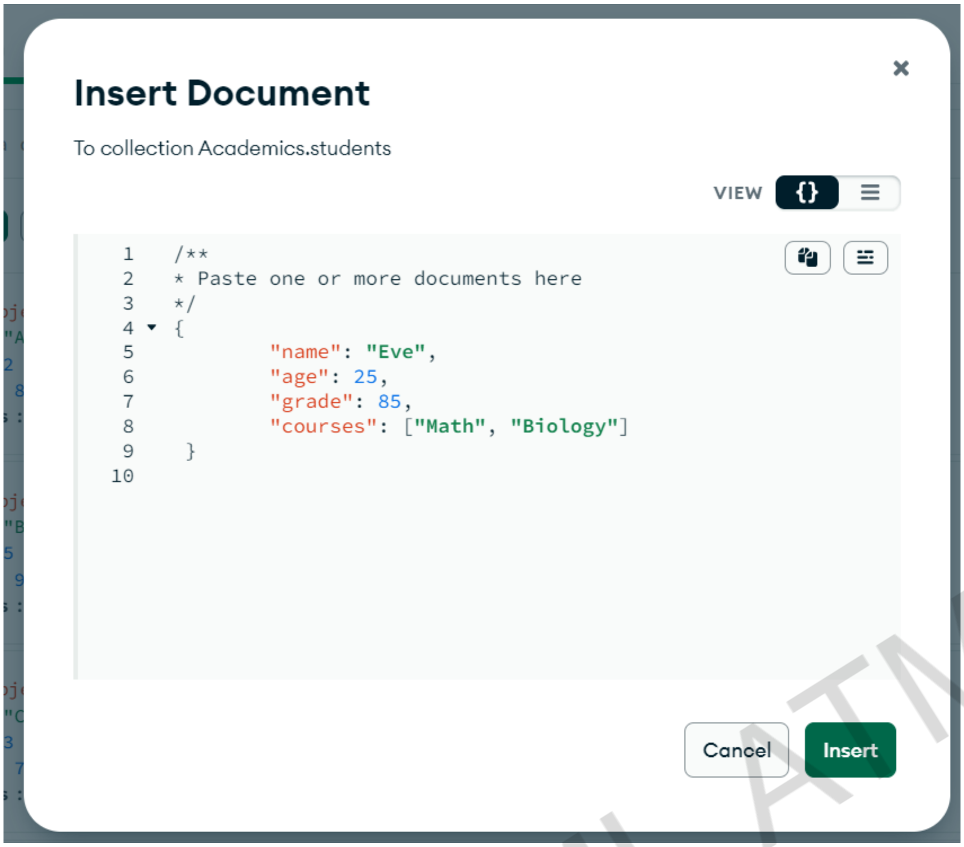


Insert Document

To collection Academics.students

```
/**
* Paste one or more documents here
*/
{
          "name": "David",
          "age": 22,
          "grade": 95,
         "courses": ["Chemistry", "Biology"]
}
```

```
{
     "name": "Eve",
     "age": 25,
     "grade": 85,
     "courses": ["Math", "Biology"]
 }
```



Insert Document

To collection Academics.students

```
/**
 * Paste one or more documents here
 */
{
        "name": "Eve",
        "age": 25,
        "grade": 85,
        "courses": ["Math", "Biology"]
 }
```

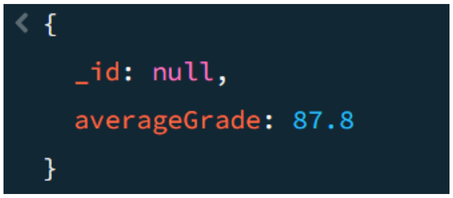
In MongoShell

>use Academics

1. **$avg** - Calculate the average grade of all students

```
> db.students.aggregate([
  {
    $group: {
      _id: null,
      averageGrade: { $avg: "$grade" }
    }
  }
])
```

Output:

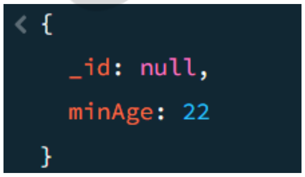


```
< {
    _id: null,
    averageGrade: 87.8
  }
```

2. **$min** - Find the minimum age of students

```
>db.students.aggregate([
  {
    $group: {
      _id: null,
      minAge: { $min: "$age" }
    }
  }
])
```

Output:

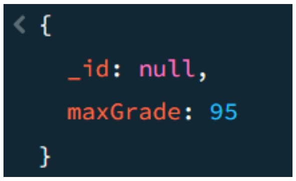


```
< {
    _id: null,
    minAge: 22
  }
```

3. **$max** - Find the maximum grade among students

```
>db.students.aggregate([
  {
    $group: {
      _id: null,
      maxGrade: { $max: "$grade" }
    }
  }
])
```

Output:

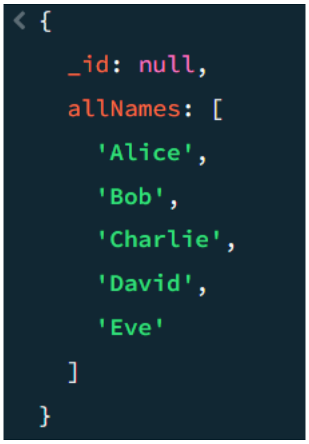


```
< {
    _id: null,
    maxGrade: 95
  }
```

4. **$push** - List all student names in an array

```
db.students.aggregate([
  {
    $group: {
      _id: null,
      allNames: { $push: "$name" }
    }
  }
])
```

Output:

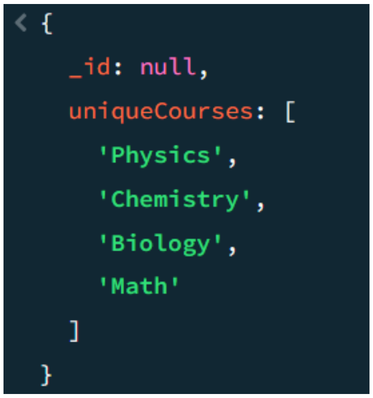


```
< {
    _id: null,
    allNames: [
      'Alice',
      'Bob',
      'Charlie',
      'David',
      'Eve'
    ]
  }
```

5. **$addToSet** - List all unique courses taken by students

```
>db.students.aggregate([
  {
    $unwind: "$courses"
  },
  {
    $group: {
      _id: null,
      uniqueCourses: { $addToSet: "$courses" }
    }
  }
])
```

Output:



```
{
  _id: null,
  uniqueCourses: [
    'Physics',
    'Chemistry',
    'Biology',
    'Math'
  ]
}
```