

HW 10.06.24 General Tech

1. Создайте пустую коллекцию students.

```
db.createCollection("students")
```

2. Заполните ее данными о студентах.

```
db.students.insertMany([
  { name: "Alice", age: 20, major: "Computer Science", gpa: 3.5 },
  { name: "Bob", age: 22, major: "Mathematics", gpa: 3.8 },
  { name: "Charlie", age: 21, major: "Physics", gpa: 3.2 },
  { name: "David", age: 23, major: "Biology", gpa: 3.9 },
  { name: "Emma", age: 20, major: "Chemistry", gpa: 3.6 },
  { name: "Lucy", age: 21, major: "Physics", gpa: 3.4 },
  { name: "Maya", age: 22, major: "Biology", gpa: 3.7 },
  { name: "Tom", age: 19, major: "Chemistry", gpa: 3.4 },
  { name: "Lucas", age: 20, major: "Physics", gpa: 3.6 },
  { name: "Bill", age: 22, major: "Mathematics", gpa: 3.7 }
])
```

3. Найдите количество студентов с GPA выше 3.5.

```
db.students.countDocuments({ gpa: { $gt: 3.5 } })
```

4. Найдите количество студентов младше 21 года.

```
db.students.countDocuments({ age: { $lt: 21 } })
```

5. Найдите всех студентов, у которых major - Physics, и установите им возраст 18 лет.

```
db.students.updateMany({ major: "Physics" }, { $set: { age: 18 } })
```

6. Увеличьте возраст студентов, у которых GPA ниже 3.5, на 1 год.

```
db.students.updateMany({ gpa: { $lt: 3.5 } }, { $inc: { age: 1 } })
```

7. Обновите GPA для студента "Charlie" на 3.4.

```
db.students.updateOne({ name: "Charlie" }, { $set: { gpa: 3.4 } })
```

8. Удалите студента с именем "David" из коллекции

```
db.students.deleteOne({ name: "David" })
```

9. Используя агрегацию, найдите средний GPA всех студентов

```
db.students.aggregate([  
  { $group: { _id: null, averageGpa: { $avg: "$gpa" } } }  
])
```

10. Используя агрегацию, найдите сумму возрастов всех студентов.

```
db.students.aggregate([  
  { $group: { _id: null, totalAge: { $sum: "$age" } } }  
])
```

11. Используя агрегацию, отсортируйте студентов по возрасту по убыванию и верните первых двух.

```
db.students.aggregate([  
  { $sort: { age: -1 } },  
  { $limit: 2 }  
])
```

12. Используя агрегацию, найдите средний GPA среди студентов, возраст которых меньше 22 лет.

```
db.students.aggregate([  
  { $match: { age: { $lt: 22 } } },  
  { $group: { _id: null, averageGpa: { $avg: "$gpa" } } }  
])
```

13. Используя агрегацию, верните только имена студентов, чьи GPA выше 3.7.

```
db.students.aggregate([  
  { $match: { gpa: { $gt: 3.7 } } },  
  { $project: { _id: 0, name: 1 } }  
])
```

14. Используя агрегацию, найдите средний возраст студентов в каждом мажоре, пропустив первые два результата.

```
db.students.aggregate([  
  { $group: { _id: "$major", averageAge: { $avg: "$age" } } },
```

```
{ $skip: 2 }  
])
```

15. Используя агрегацию, найдите и верните только имя студента с самой высокой GPA.

```
db.students.aggregate([  
  { $sort: { gpa: -1 } },  
  { $limit: 1 },  
  { $project: { _id: 0, name: 1 } }  
])
```