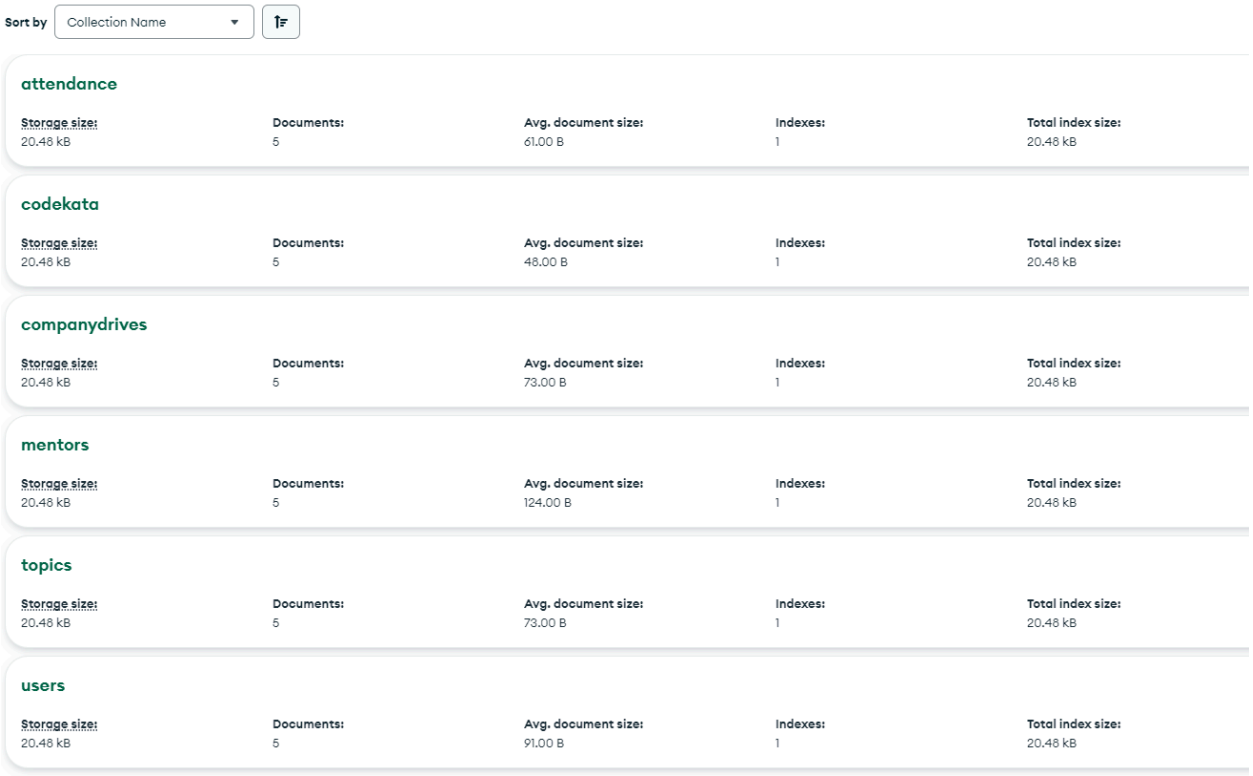
Initialize collections

Overview of collections



Create DB

use zen_class

Users collection

```
db.createCollection("users");
db.users.insertMany([
    { userid: 1, name: "John Doe", email: "john.doe@example.com" },
    { userid: 2, name: "Jane Smith", email: "jane.smith@example.com" },
    { userid: 3, name: "Michael Brown", email: "michael.brown@example.com" },
    { userid: 4, name: "Emily Davis", email: "emily.davis@example.com" },
    { userid: 5, name: "William Johnson", email: "william.johnson@example.com" }
]);
```

Codekata collection

```
db.createCollection("codekata");
db.codekata.insertMany([
   { userid: 1, problems: 45 },
   { userid: 2, problems: 72 },
   { userid: 3, problems: 58 },
   { userid: 4, problems: 83 },
   { userid: 5, problems: 67 }
]);
```

Attendance collection

```
db.createCollection("attendance");
db.attendance.insertMany([
   { userid: 1, topicid: 2, hasAttended: true },
   { userid: 2, topicid: 1, hasAttended: false },
   { userid: 3, topicid: 5, hasAttended: true },
   { userid: 4, topicid: 3, hasAttended: false },
   { userid: 5, topicid: 4, hasAttended: true }
]);
```

Topics collection

```
db.createCollection("topics");
db.topics.insertMany([
   { topicid: 1, topic: "HTML", topic_date: new Date("2020-01-15") },
   { topicid: 2, topic: "CSS", topic_date: new Date("2020-03-22") },
   { topicid: 3, topic: "JavaScript", topic_date: new Date("2020-10-20") },
   { topicid: 4, topic: "ReactJS", topic_date: new Date("2020-10-25") },
   { topicid: 5, topic: "NodeJS", topic_date: new Date("2020-09-04") }
]);
```

Task collection

```
db.createCollection("tasks");
db.tasks.insertMany([
   { taskid: 1, topicid: 1, userid: 1, task: "HTML Task", due_date: new Date("2020-01-31"),
submitted_date: new Date("2020-01-30") },
   { taskid: 2, topicid: 2, userid: 2, task: "CSS Task", due_date: new Date("2020-03-31"),
submitted_date: new Date("2020-03-30") },
   { taskid: 3, topicid: 3, userid: 3, task: "JavaScript Task", due_date: new Date("2020-10-31"),
submitted_date: new Date("2020-10-30") },
```

```
    { taskid: 4, topicid: 4, userid: 4, task: "React Task", due_date: new Date("2020-10-31") },
    { taskid: 5, topicid: 5, userid: 5, task: "NodeJS Task", due_date: new Date("2020-09-30") }
]);
```

Companydrive collection

```
db.createCollection("companydrives");
db.companydrives.insertMany([
    { userid: 1, drive_date: new Date("2020-10-17"), company: "Apple" },
    { userid: 2, drive_date: new Date("2020-02-22"), company: "Amazon" },
    { userid: 3, drive_date: new Date("2020-03-25"), company: "TCS" },
    { userid: 4, drive_date: new Date("2020-04-30"), company: "Flipkart" },
    { userid: 5, drive_date: new Date("2020-10-29"), company: "Zomato" }
]);
```

Mentors collection

```
db.createCollection("mentors");
db.mentors.insertMany([
    { mentorid: 1, mentorname: "John Doe", mentor_email: "john.doe@example.com",
mentee_count: 20 },
    { mentorid: 2, mentorname: "Emily Davis", mentor_email: "emily.davis@example.com",
mentee_count: 18 },
    { mentorid: 3, mentorname: "Laura Martinez", mentor_email: "laura.martinez@example.com",
mentee_count: 30 },
    { mentorid: 4, mentorname: "Michael Brown", mentor_email: "michael.brown@example.com",
mentee_count: 15 },
    { mentorid: 5, mentorname: "Sophia Lee", mentor_email: "sophia.lee@example.com",
mentee_count: 12 }
]);
```

1. Find all the topics and tasks which are taught in the month of October

```
db.tasks.aggregate([
    {
        $match: {
            due_date: { $gte: new Date("2020-10-01"), $lte: new Date("2020-10-31") }
        }
    },
```

```
        // Join with Topics collection
    {
        $lookup: {
            from: "topics",
            localField: "topicid",
            foreignField: "topicid",
            as: "topic_details"
        }
    },
    {
        $unwind: "$topic_details"
    },
    {
        $project: {
            _id: 0,
            taskid: 1,
            task: 1,
            due_date: 1,
            "topic_details.topic": 1,
            "topic_details.topic_date": 1
        }
    }
]).toArray();
```

2. Find all the company drives which appeared between 15 Oct 2020 and 31 Oct 2020

```
db.companydrives.find({
   drive_date: {
      $gte: new Date("2020-10-15"),
      $lte: new Date("2020-10-31")
   }
})
```

```
> db.companydrives.find({
      drive_date: {
          $gte: new Date("2020-10-15"),
          $lte: new Date("2020-10-31")
      }
  })
< {
    _id: ObjectId('66b9a869c058beaa0902e37d'),
    userid: 1,
    drive_date: 2020-10-17T00:00:00.000Z,
    company: 'Apple'
  }
  {
    _id: ObjectId('66b9a869c058beaa0902e381'),
    userid: 4,
    drive_date: 2020-10-29T00:00:00.000Z,
    company: 'Zomato'
  }
```

3. Find all the company drives and students who are appeared for the placement

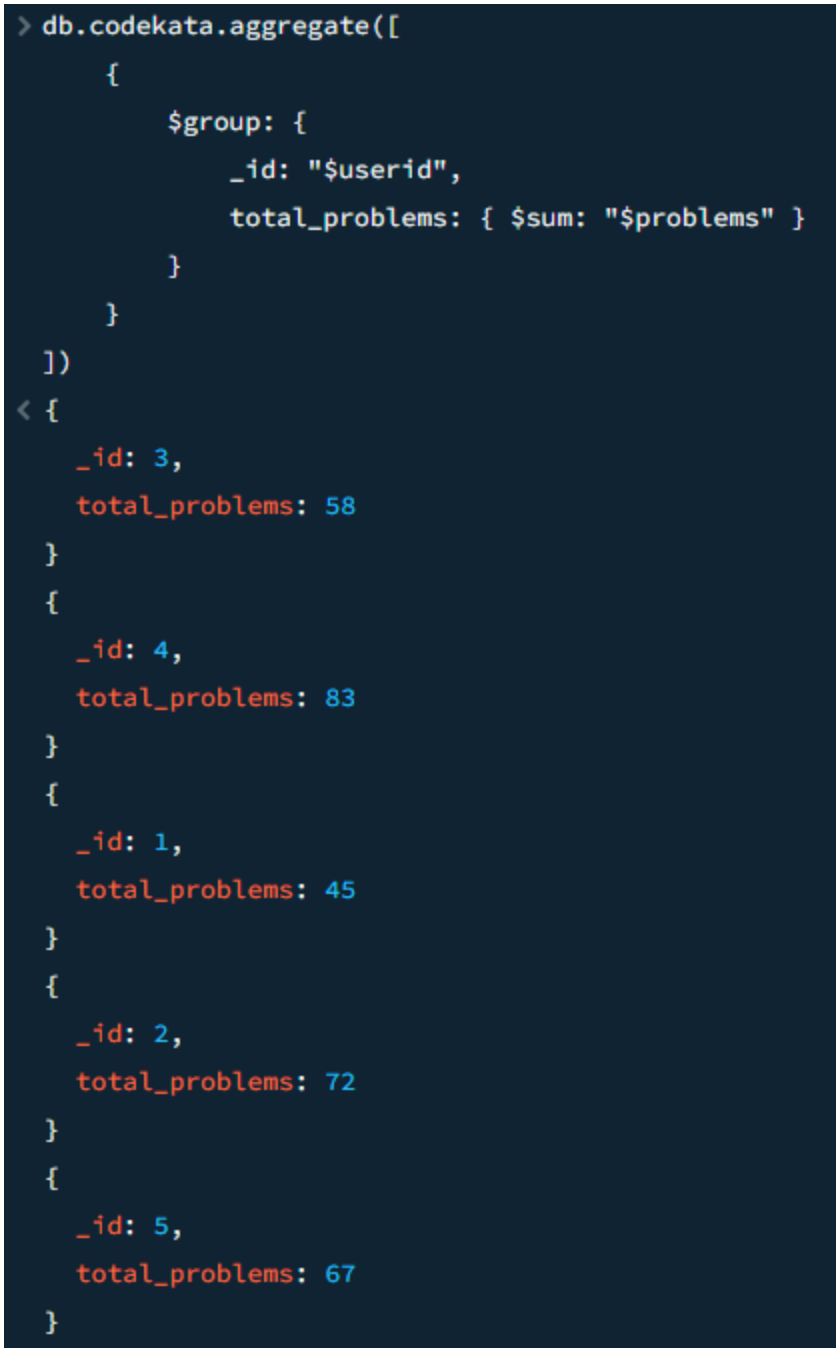
```
db.companydrives.aggregate([
        // Join with Users collection
   {
      $lookup: {
         from: "users",
         localField: "userid",
         foreignField: "userid",
```

```
            as: "student_details"
        }
    },
    {
        $unwind: "$student_details"
    },
            // To include columns needed
    {
        $project: {
            _id: 0,
            drive_date: 1,
            company: 1,
            student_details: {
                userid: 1,
                name: 1,
                email: 1
            }
        }
    }
])
```

```
{
  drive_date: 2020-10-17T00:00:00.000Z,
  company: 'Apple',
  student_details: {
    userid: 1,
    name: 'John Doe',
    email: 'john.doe@example.com'
  }
}
{
  drive_date: 2020-02-22T00:00:00.000Z,
  company: 'Amazon',
  student_details: {
    userid: 2,
    name: 'Jane Smith',
    email: 'jane.smith@example.com'
  }
}
{
  drive_date: 2020-03-25T00:00:00.000Z,
  company: 'TCS',
  student_details: {
    userid: 3,
    name: 'Michael Brown',
    email: 'michael.brown@example.com'
  }
}
{
  drive_date: 2020-04-30T00:00:00.000Z,
  company: 'Flipkart',
  student_details: {
    userid: 4,
    name: 'Emily Davis',
    email: 'emily.davis@example.com'
  }
```

4. Find the number of problems solved by the user in codekata

```
db.codekata.aggregate([
  {
    $group: {
      _id: "$userid",
      total_problems: { $sum: "$problems" }
    }
  }
])
```



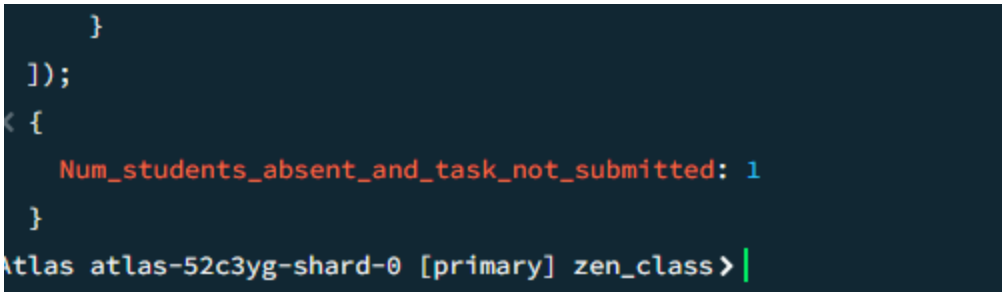
5. Find all the mentors with who has the mentee's count more than 15

```
db.mentors.find({
   mentee_count: { $gt: 15 }
})
```

6. Find the number of users who are absent and task is not submitted between 15 Oct 2020 and 31 Oct 2020

```
db.attendance.aggregate([
   // Join with topics collection
   {
      $lookup: {
         from: "topics",
         localField: "topicid",
         foreignField: "topicid",
```

```
            as: "topics_details"
        }
    },
    {
                    $unwind: "$topics_details"
            },
    // Join with tasks collection
    {
        $lookup: {
            from: "tasks",
            localField: "userid",
            foreignField: "userid",
            as: "task_details"
        }
    },
    {
                    $unwind: "$task_details"
            },
    // Match documents based on conditions
    {
        $match: {
            hasAttended: false,
            $or: [
                { "task_details.submitted_date": { $lt: new Date("2020-10-15") } },
                { "task_details.submitted_date": { $gt: new Date("2020-10-31") } }
            ]
        }
    },
    // count number of distinct users
    {
        $group: {
            _id: "$userid"
        }
    },
    {
        $count: "Num_students_absent_and_task_not_submitted"
    }
]);
```


}
]);
{
Num_students_absent_and_task_not_submitted: 1
}
tlas atlas-52c3yg-shard-0 [primary] zen_class>