MongoDB Day 2 Task

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

```
> db.topics.aggregate([{ $match: { introduction_date: { $gte: "2020-10-01", $lte: "2020-10-31" } } },
                       { $lookup: { from: "tasks", localField: "topic_id", foreignField: "topic_id", as: "tasks" } },
                       { $unwind: "$tasks" },
                       { $project: { _id: 0, topic_name: 1, introduction_date: 1, task_name: "$tasks.task_name", task_assigned_date: "$tasks.task_assigned_date" } }
                      ]).toArray();
< [
    {
      topic_name: 'React',
      introduction_date: '2020-10-01',
      task_name: 'React Component Development',
      task_assigned_date: '2020-10-01'
    },
    {
      topic_name: 'SQL',
      introduction_date: '2020-10-11',
      task_name: 'SQL Database Design Project',
      task_assigned_date: '2020-10-11'
    },
    {
      topic_name: 'MongoDB',
      introduction_date: '2020-10-21',
      task_name: 'MongoDB CRUD Operations',
      task_assigned_date: '2020-10-21'
    },
    {
      topic_name: 'Node.js',
```

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

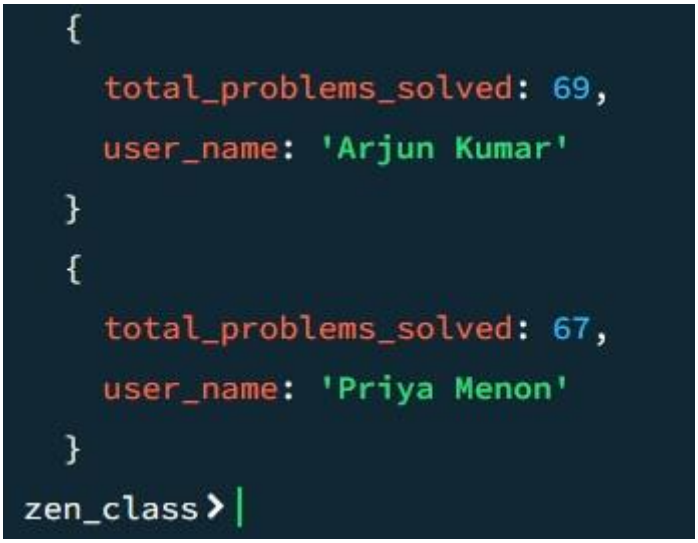
```
> db.company_drives.find({
    drive_date: {
      $gte: "2020-10-15",
      $lte: "2020-10-31"
    }
  }, {
    _id: 0,
    company_name: 1,
    drive_date: 1
  });
< {
    company_name: 'Tech Solutions Inc.',
    drive_date: '2020-10-20'
  }
  {
    company_name: 'Data Analytics Corp.',
    drive_date: '2020-10-25'
  }
zen_class>
```

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

```
> db.company_drives.aggregate([{ $unwind: "$students_appeared" },
                               { $match: { "students_appeared.status": "attended" } },
                               { $lookup: { from: "users", localField: "students_appeared.user_id", foreignField: "id", as: "student" } },
                               { $unwind: "$student" },
                               { $project: { _id: 0, company_name: 1, drive_date: 1, student_name: "$student.name", status: "$students_appeared.status" } }]);
< {
    company_name: 'Tech Solutions Inc.',
    drive_date: '2020-10-20',
    student_name: 'Aruna Iyer',
    status: 'attended'
  }
  {
    company_name: 'Tech Solutions Inc.',
    drive_date: '2020-10-20',
    student_name: 'Ananya Reddy',
    status: 'attended'
  }
  {
    company_name: 'Tech Solutions Inc.',
    drive_date: '2020-10-20',
    student_name: 'Sudha Krishnan',
    status: 'attended'
  }
  {
    company_name: 'Data Analytics Corp.',
    drive_date: '2020-10-25',
```

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

```
> db.codekata.aggregate([{ $lookup: { from: "users", localField: "user_id", foreignField: "id", as: "user" } },
                         { $unwind: "$user" },
                         { $group: { _id: "$user.name", total_problems_solved: { $sum: "$problems_solved" } } },
                         { $project: { user_name: "$_id", total_problems_solved: 1, _id: 0 } }]);
< {
    total_problems_solved: 32,
    user_name: 'Rajesh Nair'
  }
  {
    total_problems_solved: 55,
    user_name: 'Aruna Iyer'
  }
  {
    total_problems_solved: 34,
    user_name: 'Manoj Krishnan'
  }
  {
    total_problems_solved: 41,
    user_name: 'Divya Patel'
  }
  {
    total_problems_solved: 34,
    user_name: 'Vijay Raju'
  }
  {
    total_problems_solved: 69,
```



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

```
> db.mentors.aggregate([{ $match: { mentees_count: { $gt: 15 } } }, { $project: { _id: 0, mentor_name: 1, mentees_count: 1 } }]);
< {
    mentor_name: 'Karthik Subramanian',
    mentees_count: 18
  }
  {
    mentor_name: 'Rajeshwari Iyengar',
    mentees_count: 20
  }
  {
    mentor_name: 'Suresh Menon',
    mentees_count: 16
  }
  {
    mentor_name: 'Anjali Raghavan',
    mentees_count: 22
  }
zen_class>
```

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

```
> const attendanceResult = db.attendance.aggregate([{ $match: { attended: false, date: { $gt: "2020-10-15", $lt: "2020-10-31" } } },
                                                     { $group: { _id: null, user_ids: { $addToSet: "$user_id" } } }]).toArray();
  const tasksResult = db.tasks.aggregate([{ $unwind: "$assigned_to" },
                                          { $match: { "assigned_to.submitted": false, "task_assigned_date": { $gt: "2020-10-15", $lt: "2020-10-31" } } },
                                          { $group: { _id: null, user_ids: { $addToSet: "$assigned_to.user_id" } } }]).toArray();
  const mergedUserIds = Array.from(new Set([...attendanceResult[0].user_ids, ...tasksResult[0].user_ids]));
  const userCount = mergedUserIds.length;
  const result = { user_ids_count: userCount, task_status: "Not submitted", attendance_status: "Absent" };
  printjson(result);
< {
    user_ids_count: 4,
    task_status: 'Not submitted',
    attendance_status: 'Absent'
  }
zen_class>
```