

Case Study: Zen Class MongoDB Analysis

1. Project Setup

Database Name:

```
test> use zen_class;
switched to db zen_class
zen_class>
```

2. Collection Creation & Data Insertion

Now that you've created and switched to the zen_class database, the next step is to create all collections and insert the data provided in the .txt files.

File Name	Collection Name
mentorsDB.txt	mentors
userDB.txt	users
attendanceDB.txt	attendance
CodekataDB.txt	codekata
companydrivesDB.txt	companydrives
topicsDB.txt	topics
tasksDb.txt	tasks
task_completionDB.txt	task_completion

1. mentorsDB.txt → mentors Collection

```
mongosh mongodb://127.0.0.1:27017/test -
```

```
test> show dbs
admin 40.00 KiB
config 84.00 KiB
local 40.00 KiB
productDB 80.00 KiB
ragavidb 48.00 KiB
test 24.00 KiB
trainingdb 72.00 KiB
test> use zen_class
switched to db zen_class
zen_class> db.createCollection("mentors");
{ ok: 1 }
zen_class> db.mentors.insertMany([
...   { mentor_id: 1, mentor_name: 'Sathish', mentees: 30 },
...   { mentor_id: 2, mentor_name: 'Sudar', mentees: 13 },
...   { mentor_id: 3, mentor_name: 'Thillan', mentees: 50 },
...   { mentor_id: 4, mentor_name: 'Kulunthan', mentees: 70 },
...   { mentor_id: 5, mentor_name: 'Mani', mentees: 16 },
...   { mentor_id: 6, mentor_name: 'Balu', mentees: 10 },
...   { mentor_id: 7, mentor_name: 'Guru', mentees: 13 },
...   { mentor_id: 8, mentor_name: 'Parthiban', mentees: 32 },
...   { mentor_id: 9, mentor_name: 'Nesamani', mentees: 14 },
...   { mentor_id: 10, mentor_name: 'Vijay', mentees: 35 }
... ]);
{
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('6881c1535898fc7896eec4a9'),
    '1': ObjectId('6881c1535898fc7896eec4aa'),
    '2': ObjectId('6881c1535898fc7896eec4ab'),
    '3': ObjectId('6881c1535898fc7896eec4ac'),
    '4': ObjectId('6881c1535898fc7896eec4ad'),
    '5': ObjectId('6881c1535898fc7896eec4ae'),
    '6': ObjectId('6881c1535898fc7896eec4af'),
    '7': ObjectId('6881c1535898fc7896eec4b0'),
    '8': ObjectId('6881c1535898fc7896eec4b1'),
    '9': ObjectId('6881c1535898fc7896eec4b2')
  ]
}
zen_class> |
```

2. userDB.txt → users Collection

```
zen_class> db.createCollection("users");
{ ok: 1 }
zen_class> db.users.insertMany([
... { user_id: 1, name: 'Abi', email: 'abi@email.com', batch: 'B45WE' },
... { user_id: 2, name: 'Bala', email: 'bala@email.com', batch: 'B45WE' },
... { user_id: 3, name: 'Chandru', email: 'chandru@email.com', batch: 'B45WE' },
... { user_id: 4, name: 'Dhanush', email: 'dhanush@email.com', batch: 'B45WE' },
... { user_id: 5, name: 'Ezhil', email: 'ezhil@email.com', batch: 'B45WE' }
... ]);
{
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('6881c1f75898fc7896eec4b3'),
    '1': ObjectId('6881c1f75898fc7896eec4b4'),
    '2': ObjectId('6881c1f75898fc7896eec4b5'),
    '3': ObjectId('6881c1f75898fc7896eec4b6'),
    '4': ObjectId('6881c1f75898fc7896eec4b7')
  ]
}
zen_class> |
```

3. attendanceDB.txt → attendance Collection

```
zen_class> db.createCollection("attendance");
{ ok: 1 }
zen_class> db.attendance.insertMany([
... { userid: 1, Class_date: ISODate("2023-10-01"), attendance: true },
... { userid: 2, Class_date: ISODate("2023-10-01"), attendance: true },
... { userid: 3, Class_date: ISODate("2023-10-01"), attendance: false },
... { userid: 4, Class_date: ISODate("2023-10-01"), attendance: false },
... { userid: 5, Class_date: ISODate("2023-10-01"), attendance: true }
... ]);
{
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('6881c23f5898fc7896eec4b8'),
    '1': ObjectId('6881c23f5898fc7896eec4b9'),
    '2': ObjectId('6881c23f5898fc7896eec4ba'),
    '3': ObjectId('6881c23f5898fc7896eec4bb'),
    '4': ObjectId('6881c23f5898fc7896eec4bc')
  ]
}
zen_class> |
```

4. CodekataDB.txt → codekata Collection

```
zen_class> db.createCollection("codekata");
{ ok: 1 }
zen_class> db.codekata.insertMany([
... { user_id: 1, codekata_solved: 70, webkata_solved: 10 },
... { user_id: 2, codekata_solved: 50, webkata_solved: 5 },
... { user_id: 3, codekata_solved: 20, webkata_solved: 7 },
... { user_id: 4, codekata_solved: 90, webkata_solved: 9 },
... { user_id: 5, codekata_solved: 60, webkata_solved: 6 }
... ]);
{
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('6881c27e5898fc7896eec4bd'),
    '1': ObjectId('6881c27e5898fc7896eec4be'),
    '2': ObjectId('6881c27e5898fc7896eec4bf'),
    '3': ObjectId('6881c27e5898fc7896eec4c0'),
    '4': ObjectId('6881c27e5898fc7896eec4c1')
  ]
}
zen_class> |
```

5. companydrivesDB.txt → companydrives Collection

```
zen_class> db.createCollection("companydrives");
{ ok: 1 }
zen_class> db.companydrives.insertMany([
... { drive_id: 1, company: "TCS", drive_date: ISODate("2023-10-20") },
... { drive_id: 2, company: "Infosys", drive_date: ISODate("2023-10-24") },
... { drive_id: 3, company: "Wipro", drive_date: ISODate("2023-10-10") },
... { drive_id: 4, company: "HCL", drive_date: ISODate("2023-11-01") },
... { drive_id: 5, company: "Zoho", drive_date: ISODate("2023-10-18") }
... ]);
{
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('6881c2bb5898fc7896eec4c2'),
    '1': ObjectId('6881c2bb5898fc7896eec4c3'),
    '2': ObjectId('6881c2bb5898fc7896eec4c4'),
    '3': ObjectId('6881c2bb5898fc7896eec4c5'),
    '4': ObjectId('6881c2bb5898fc7896eec4c6')
  ]
}
zen_class> |
```

6. topicsDB.txt → topics Collection

```
zen_class> db.createCollection("topics");
{ ok: 1 }
zen_class> db.topics.insertMany([
... { topic_id: 1, topic_name: "HTML", start_date: ISODate("2023-10-01") },
... { topic_id: 2, topic_name: "CSS", start_date: ISODate("2023-10-05") },
... { topic_id: 3, topic_name: "JavaScript", start_date: ISODate("2023-10-10") },
... { topic_id: 4, topic_name: "React", start_date: ISODate("2023-10-20") },
... { topic_id: 5, topic_name: "Node.js", start_date: ISODate("2023-11-01") }
... ]);
{
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('6881ca635898fc7896eec4c7'),
    '1': ObjectId('6881ca635898fc7896eec4c8'),
    '2': ObjectId('6881ca635898fc7896eec4c9'),
    '3': ObjectId('6881ca635898fc7896eec4ca'),
    '4': ObjectId('6881ca635898fc7896eec4cb')
  ]
}
zen_class> |
```

7. tasksDb.txt → tasks Collection

```
zen_class> db.createCollection("tasks");
{ ok: 1 }
zen_class> db.tasks.insertMany([
... { task_id: 1, topic_id: 1, task_name: "HTML Page", due_date: ISODate("2023-10-03") },
... { task_id: 2, topic_id: 2, task_name: "CSS Layout", due_date: ISODate("2023-10-08") },
... { task_id: 3, topic_id: 3, task_name: "JS Calculator", due_date: ISODate("2023-10-15") },
... { task_id: 4, topic_id: 4, task_name: "React Todo", due_date: ISODate("2023-10-25") },
... { task_id: 5, topic_id: 5, task_name: "Node API", due_date: ISODate("2023-11-05") }
... ]);
{
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('6881cab55898fc7896eec4cc'),
    '1': ObjectId('6881cab55898fc7896eec4cd'),
    '2': ObjectId('6881cab55898fc7896eec4ce'),
    '3': ObjectId('6881cab55898fc7896eec4cf'),
    '4': ObjectId('6881cab55898fc7896eec4d0')
  ]
}
zen_class> |
```

8. task_completionDB.txt → task_completion Collection

```
zen_class> db.createCollection("task_completion");
{ ok: 1 }
zen_class> db.task_completion.insertMany([
... { userid: 1, task_id: 1, task_completion: true },
... { userid: 2, task_id: 2, task_completion: false },
... { userid: 3, task_id: 3, task_completion: true },
... { userid: 4, task_id: 4, task_completion: false },
... { userid: 5, task_id: 5, task_completion: true }
... ]);
{
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('6881caf55898fc7896eec4d1'),
    '1': ObjectId('6881caf55898fc7896eec4d2'),
    '2': ObjectId('6881caf55898fc7896eec4d3'),
    '3': ObjectId('6881caf55898fc7896eec4d4'),
    '4': ObjectId('6881caf55898fc7896eec4d5')
  ]
}
zen_class> |
```

⇒ We have created Collections and added necessary documents in it.

```
zen_class> show collections
attendance
codekata
companydrives
mentors
task_completion
tasks
topics
users
```

Q1: Find all the topics and tasks which are taught in the month of October.

```
zen_class> db.topics.find({
...   start_date: {
...     $gte: ISODate("2023-10-01"),
...     $lt: ISODate("2023-11-01")
...   }
... });
[{
  _id: ObjectId('6881ca635898fc7896eec4c7'),
  topic_id: 1,
  topic_name: 'HTML',
  start_date: ISODate('2023-10-01T00:00:00.000Z')
},
{
  _id: ObjectId('6881ca635898fc7896eec4c8'),
  topic_id: 2,
  topic_name: 'CSS',
  start_date: ISODate('2023-10-05T00:00:00.000Z')
},
{
  _id: ObjectId('6881ca635898fc7896eec4c9'),
  topic_id: 3,
  topic_name: 'JavaScript',
  start_date: ISODate('2023-10-10T00:00:00.000Z')
},
{
  _id: ObjectId('6881ca635898fc7896eec4ca'),
  topic_id: 4,
  topic_name: 'React',
  start_date: ISODate('2023-10-20T00:00:00.000Z')
}
]
```

```
zen_class> db.tasks.find({
...   due_date: {
...     $gte: ISODate("2023-10-01"),
...     $lt: ISODate("2023-11-01")
...   }
... });
[{
  _id: ObjectId('6881cab55898fc7896eec4cc'),
  task_id: 1,
  topic_id: 1,
  task_name: 'HTML Page',
  due_date: ISODate('2023-10-03T00:00:00.000Z')
},
{
  _id: ObjectId('6881cab55898fc7896eec4cd'),
  task_id: 2,
  topic_id: 2,
  task_name: 'CSS Layout',
  due_date: ISODate('2023-10-08T00:00:00.000Z')
},
{
  _id: ObjectId('6881cab55898fc7896eec4ce'),
  task_id: 3,
  topic_id: 3,
  task_name: 'JS Calculator',
  due_date: ISODate('2023-10-15T00:00:00.000Z')
},
{
  _id: ObjectId('6881cab55898fc7896eec4cf'),
  task_id: 4,
  topic_id: 4,
  task_name: 'React Todo',
  due_date: ISODate('2023-10-25T00:00:00.000Z')
}
]
```

Q2: Find all the company drives which appeared between 15 Oct 2023 and 31 Oct 2023.

```
zen_class> db.companydrives.find({
...   drive_date: {
...     $gte: ISODate("2023-10-15"),
...     $lte: ISODate("2023-10-31")
...   }
... });
[{
  _id: ObjectId('6881c2bb5898fc7896eec4c2'),
  drive_id: 1,
  company: 'TCS',
  drive_date: ISODate('2023-10-20T00:00:00.000Z')
},
{
  _id: ObjectId('6881c2bb5898fc7896eec4c3'),
  drive_id: 2,
  company: 'Infosys',
  drive_date: ISODate('2023-10-24T00:00:00.000Z')
},
{
  _id: ObjectId('6881c2bb5898fc7896eec4c6'),
  drive_id: 5,
  company: 'Zoho',
  drive_date: ISODate('2023-10-18T00:00:00.000Z')
}
]
```

Q3. Find the number of problems solved by each user in CodeKata (codekata + webkata).

```
zen_class> db.codekata.aggregate([
...   {
...     $project: {
...       user_id: 1,
...       total_problems_solved: {
...         $add: ["$codekata_solved", "$webkata_solved"]
...       }
...     }
...   }
... ]);
[{
  _id: ObjectId('6881c27e5898fc7896eec4bd'),
  user_id: 1,
  total_problems_solved: 80
},
{
  _id: ObjectId('6881c27e5898fc7896eec4be'),
  user_id: 2,
  total_problems_solved: 55
},
{
  _id: ObjectId('6881c27e5898fc7896eec4bf'),
  user_id: 3,
  total_problems_solved: 27
},
{
  _id: ObjectId('6881c27e5898fc7896eec4c0'),
  user_id: 4,
  total_problems_solved: 99
},
{
  _id: ObjectId('6881c27e5898fc7896eec4c1'),
  user_id: 5,
  total_problems_solved: 66
}]
zen_class>
```

Q4. Find all the mentors who have mentee count more than 15.

```
zen_class> db.mentors.find({ mentees: { $gt: 15 } });
[{
  _id: ObjectId('6881c1535898fc7896eec4a9'),
  mentor_id: 1,
  mentor_name: 'Sathish',
  mentees: 30
},
{
  _id: ObjectId('6881c1535898fc7896eec4ab'),
  mentor_id: 3,
  mentor_name: 'Thillan',
  mentees: 50
},
{
  _id: ObjectId('6881c1535898fc7896eec4ac'),
  mentor_id: 4,
  mentor_name: 'Kulunthan',
  mentees: 70
},
{
  _id: ObjectId('6881c1535898fc7896eec4ad'),
  mentor_id: 5,
  mentor_name: 'Mani',
  mentees: 16
},
{
  _id: ObjectId('6881c1535898fc7896eec4b0'),
  mentor_id: 8,
  mentor_name: 'Parthiban',
  mentees: 32
},
{
  _id: ObjectId('6881c1535898fc7896eec4b2'),
  mentor_id: 10,
  mentor_name: 'Vijay',
  mentees: 35
}]
zen_class>
```

Q5: Find the number of users who are absent and didn't submit tasks between 15 Oct 2023 and 31 Oct 2023

Step 1: Get absentees between 15-31 Oct

```
zen_class> const absentees = db.attendance.find({
...   Class_date: {
...     $gte: ISODate("2023-10-15"),
...     $lte: ISODate("2023-10-31")
...   },
...   attendance: false
... }).toArray();
```

⇒ Grabs all users who were absent in the date range.

Step 2: Extract distinct user IDs

```
zen_class> const absentUserIds = [...new Set(absentees.map(a => a.userid))];
```

⇒ Extracts unique user IDs from the absentee list using JS Set.

Step 3: Find users who also didn't complete tasks

```
zen_class> const incompleteUsers = db.task_completion.find({
...   userid: { $in: absentUserIds },
...   task_completion: false
... }).toArray();
```

⇒ Filters users from the above list who also didn't complete their tasks.

Step 4: Count the users

```
zen_class> incompleteUsers.length;
```

⇒ Gets the final count of users who match both conditions.

Conclusion

- Explored MongoDB using the mongosh command-line interface.
 - Created the zen_class database and added 8 collections using JavaScript insert commands.
 - Verified inserted data using MongoDB shell commands like .find() and .aggregate().
 - Wrote and executed 5 real-world MongoDB queries involving Date filtering, Aggregation, Conditional matching and Simulated joins using \$lookup and scripting logic.
-