Case Study: MongoDB Implementation for Zen Class Data Analysis

Prepared By: Ruthravarshan S
Date: July 23, 2025

Database and Collection Setup

Collections Created:

1. **users** – Contains user profile data.
2. **topics –** Tracks topics taught and their schedule.
3. **tasks –** Holds task assignments and deadlines.
4. **task_completion –** Indicates task submission status per user.
5. **companydrives –** Logs user participation in company drives.
6. **codekata –** Records coding problems solved by users.
7. **mentors –** Contains mentor and mentee statistics.
8. **attendance –** Tracks user class attendance by date**.**

All collections were populated using insertMany() commands in MongoDB Shell.

Database Name: zen_class

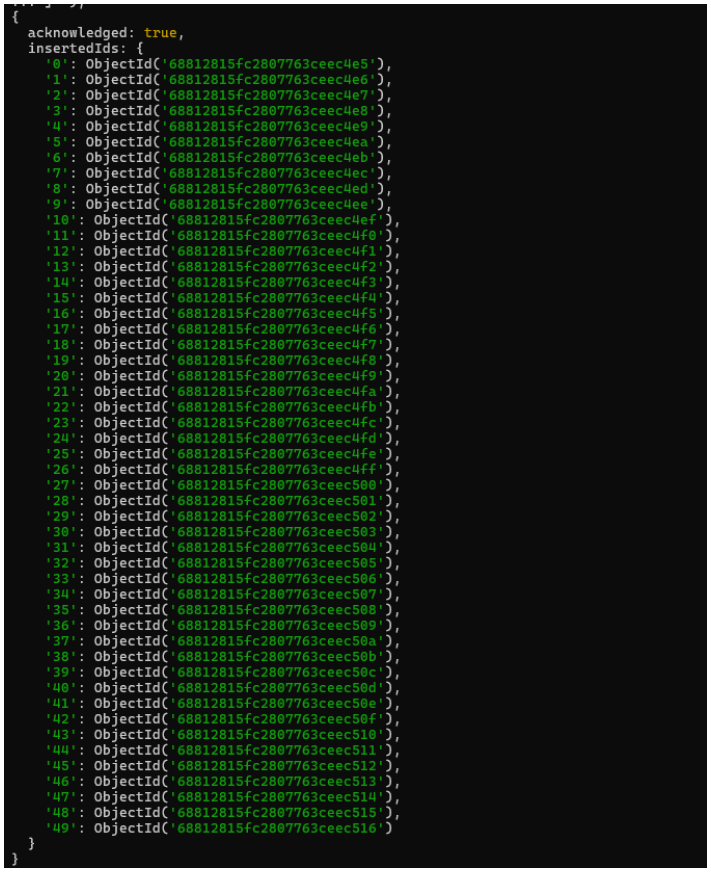
CREATE DB

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

Create Collections

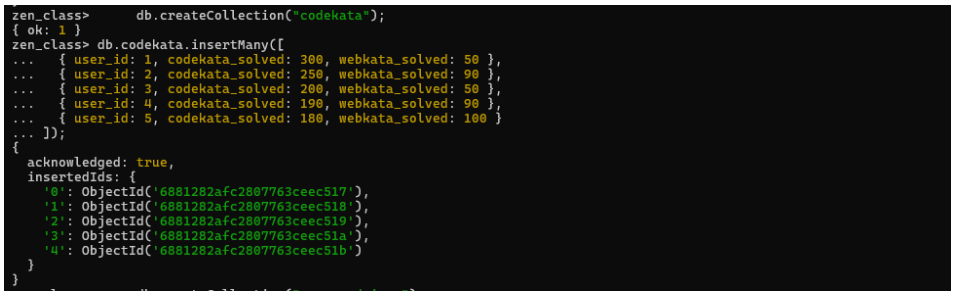
attendance

- Inserted using attendanceDB.txt
- Fields: userid, Class_date, attendance



codekata

- Inserted using CodekataDB.txt
- Fields: user_id, codekata_solved, webkata_solved



companydrives

- Inserted using companydrivesDB.txt
- Fields: userid, drive_date, company

```
zen_class>      db.createCollection("companydrives");
{ ok: 1 }
zen_class> db.companydrives.insertMany([
...    {
...        userid:1,
...        drive_date:ISODate("2023-10-16"),
...        company:"TCS"
...    },
...     {
...        userid:1,
...        drive_date:ISODate("2023-10-19"),
...        company:"HCL"
...    },
...     {
...        userid:2,
...        drive_date:ISODate("2023-10-22"),
...        company:"Amazon"
...    },
...     {
...        userid:3,
...        drive_date:ISODate("2023-10-28"),
...        company:"Wipro"
...    },
...     {
...        userid:4,
...        drive_date:ISODate("2023-11-01"),
...        company:"ITC"
...    }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68812839fc2807763ceec51c'),
    '1': ObjectId('68812839fc2807763ceec51d'),
    '2': ObjectId('68812839fc2807763ceec51e'),
    '3': ObjectId('68812839fc2807763ceec51f'),
    '4': ObjectId('68812839fc2807763ceec520')
  }
}
```

mentors

- Inserted using mentorsDB.txt
- Fields: mentor_id, mentor_name, mentees

```
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('68812847fc2807763ceec521'),
    '1': ObjectId('68812847fc2807763ceec522'),
    '2': ObjectId('68812847fc2807763ceec523'),
    '3': ObjectId('68812847fc2807763ceec524'),
    '4': ObjectId('68812847fc2807763ceec525'),
    '5': ObjectId('68812847fc2807763ceec526'),
    '6': ObjectId('68812847fc2807763ceec527'),
    '7': ObjectId('68812847fc2807763ceec528'),
    '8': ObjectId('68812847fc2807763ceec529'),
    '9': ObjectId('68812847fc2807763ceec52a')
  }
}
```

task_completion

- Inserted using task_completionDB.txt
- Fields: userid, task_completion, pending_tasks_id

```
zen_class> db.createCollection("task_completion");
{ ok: 1 }
zen_class> db.task_completion.insertMany([
...   {
...     "userid":1,
...     "task_completion":true,
...     "pending_tasks_id":[]
...   },
...   {
...     "userid":2,
...     "task_completion":true,
...     "pending_tasks_id":[]
...   },
...   {
...     "userid":3,
...     "task_completion":false,
...     "pending_task_id":[1,2]
...   },
...   {
...     "userid":4,
...     "task_completion":true,
...     "pending_tasks_id":[]
...   },
...   {
...     "userid":5,
...     "task_completion":false,
...     "pending_tasks_id":[2]
...   }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68812856fc2807763ceec52b'),
    '1': ObjectId('68812856fc2807763ceec52c'),
    '2': ObjectId('68812856fc2807763ceec52d'),
    '3': ObjectId('68812856fc2807763ceec52e'),
    '4': ObjectId('68812856fc2807763ceec52f')
  }
}
```

tasks

- Inserted using tasksDb.txt
- Fields: topic_id, task_id, topic, description, deadline_

```
zen_class> db.createCollection("tasks");
{ ok: 1 }
zen_class> db.tasks.insertMany([
...   {
...     "topic_id":1,
...     "task_id":1,
...     "topic":"HTML",
...     "description":"Create a content using HTML",
...     "deadline_date":ISODate("2023-10-09")
...   },
...   {
...     "topic_id":2,
...     "task_id":2,
...     "topic":"CSS",
...     "description":"Style - created HTML content",
...     "deadline_date":ISODate("2023-10-16")
...   },
...   {
...     "topic_id":3,
...     "task_id":3,
...     "topic":"Javascript",
...     "description":"Add logics to the created content",
...     "deadline_date":ISODate("2023-10-23")
...   },
...   {
...     "topic_id":4,
...     "task_id":4,
...     "topic":"React",
...     "description":"Create same website (created by using of html,css & js) using react",
...     "deadline_date":ISODate("2023-10-30")
...   },
... {
...   "topic_id": 5,
...   "task_id": 5,
...   "topic": "Bootstrap",
...   "description": "Create same website using bootstrap",
...  "deadline_date":ISODate("2023-11-06")
... }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68812864fc2807763ceec530'),
    '1': ObjectId('68812864fc2807763ceec531'),
    '2': ObjectId('68812864fc2807763ceec532'),
    '3': ObjectId('68812864fc2807763ceec533'),
    '4': ObjectId('68812864fc2807763ceec534')
  }
}
```

topics

- Inserted using topicsDB.txt
- Fields: topic_id, task_id, topic, description, start_date, end_date

```
zen_class> db.createCollection("topics");
{ ok: 1 }
zen_class> db.topics.insertMany([
...   {
...     "topic_id":1,
...      "task_id":1,
...     "topic":"HTML",
...     "description":"Master Html",
...     "start_date":ISODate("2023-10-07"),
...     "end_date":ISODate("2023-10-08")
...   },
...   {
...     "topic_id":2,
...      "task_id":2,
...     "topic":"CSS",
...     "description":"Master Css",
...     "start_date":ISODate("2023-10-14"),
...     "end_date":ISODate("2023-10-15")
...   },
...   {
...     "topic_id":3,
...      "task_id":3,
...     "topic":"Javascript",
...     "description":"Master Javascript",
...     "start_date":ISODate("2023-10-21"),
...     "end_date":ISODate("2023-10-22")
...   },
...   {
...     "topic_id":4,
...      "task_id":4,
...     "topic":"React",
...     "description":"Master React",
...     "start_date":ISODate("2023-10-28"),
...     "end_date":ISODate("2023-10-29")
...   },
...  {
...   "topic_id": 5,
...    "task_id":5,
...   "topic": "Bootstrap",
...   "description": "Master Bootstrap",
...   "start_date":ISODate("2023-11-05"),
...   "end_date":ISODate("2023-11-06")
... }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68812871fc2807763ceec535'),
    '1': ObjectId('68812871fc2807763ceec536'),
    '2': ObjectId('68812871fc2807763ceec537'),
    '3': ObjectId('68812871fc2807763ceec538'),
    '4': ObjectId('68812871fc2807763ceec539')
  }
}
```

users

- Inserted using userDB.txt
- Fields: userid, name, email, mentor

```
zen_class> db.createCollection("users");
{ ok: 1 }
zen_class> db.users.insertMany([
...   {
...     userid: 1,
...     name: 'Boobathi',
...     email: 'boobathi@gmail.com',
...     mentor: 'Sathish'
...   },
...   {
...     userid: 2,
...     name: 'Mani',
...     email: 'mani@gmail.com',
...     mentor: 'Nagaraj'
...   },
...   {
...     userid: 3,
...     name: 'Karthi',
...     email: 'karthi@gmail.com',
...     mentor: 'Sathish'
...   },
...   {
...     userid: 4,
...     name: 'Dharshini',
...     email: 'dharshini@gmail.com',
...     mentor: 'Rupan'
...   },
...   {
...     userid: 5,
...     name: 'Balu',
...     email: 'balu@gmail.com',
...     mentor: 'Sathish'
...   }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6881287dfc2807763ceec53a'),
    '1': ObjectId('6881287dfc2807763ceec53b'),
    '2': ObjectId('6881287dfc2807763ceec53c'),
    '3': ObjectId('6881287dfc2807763ceec53d'),
    '4': ObjectId('6881287dfc2807763ceec53e')
  }
}
```

Created Collections and added necessary documents in it.

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

Query 1: 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"),
...     $lte: ISODate("2023-10-31")
...   }
... });
...
[
  {
    _id: ObjectId('68812871fc2807763ceec535'),
    topic_id: 1,
    task_id: 1,
    topic: 'HTML',
    description: 'Master Html',
    start_date: ISODate('2023-10-07T00:00:00.000Z'),
    end_date: ISODate('2023-10-08T00:00:00.000Z')
  },
  {
    _id: ObjectId('68812871fc2807763ceec536'),
    topic_id: 2,
    task_id: 2,
    topic: 'CSS',
    description: 'Master Css',
    start_date: ISODate('2023-10-14T00:00:00.000Z'),
    end_date: ISODate('2023-10-15T00:00:00.000Z')
  },
  {
    _id: ObjectId('68812871fc2807763ceec537'),
    topic_id: 3,
    task_id: 3,
    topic: 'Javascript',
    description: 'Master Javascript',
    start_date: ISODate('2023-10-21T00:00:00.000Z'),
    end_date: ISODate('2023-10-22T00:00:00.000Z')
  },
  {
    _id: ObjectId('68812871fc2807763ceec538'),
    topic_id: 4,
    task_id: 4,
    topic: 'React',
    description: 'Master React',
    start_date: ISODate('2023-10-28T00:00:00.000Z'),
    end_date: ISODate('2023-10-29T00:00:00.000Z')
  }
]
zen_class>
```

Query 2: 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('68812839fc2807763ceec51c'),
    userid: 1,
    drive_date: ISODate('2023-10-16T00:00:00.000Z'),
    company: 'TCS'
  },
  {
    _id: ObjectId('68812839fc2807763ceec51d'),
    userid: 1,
    drive_date: ISODate('2023-10-19T00:00:00.000Z'),
    company: 'HCL'
  },
  {
    _id: ObjectId('68812839fc2807763ceec51e'),
    userid: 2,
    drive_date: ISODate('2023-10-22T00:00:00.000Z'),
    company: 'Amazon'
  },
  {
    _id: ObjectId('68812839fc2807763ceec51f'),
    userid: 3,
    drive_date: ISODate('2023-10-28T00:00:00.000Z'),
    company: 'Wipro'
  }
]
zen_class>
```

Query 3: Find the number of problems solved by the user in codekata

```
zen_class> db.codekata.aggregate([
...   {
...     $group: {
...       _id: null,
...       total_codekata_solved: { $sum: "$codekata_solved" }
...     }
...   }
... ]);
...
[ { _id: null, total_codekata_solved: 1120 } ]
zen_class>
```

Query 4: Find all the mentors who have mentee count more than 15

```
zen_class> db.mentors.find({
...   mentees: { $gt: 15 }
... });
...
[
  {
    _id: ObjectId('68812847fc2807763ceec521'),
    mentor_id: 1,
    mentor_name: 'Sathish',
    mentees: 30
  },
  {
    _id: ObjectId('68812847fc2807763ceec523'),
    mentor_id: 3,
    mentor_name: 'Thillan',
    mentees: 50
  },
  {
    _id: ObjectId('68812847fc2807763ceec524'),
    mentor_id: 4,
    mentor_name: 'Kulunthan',
    mentees: 70
  },
  {
    _id: ObjectId('68812847fc2807763ceec525'),
    mentor_id: 5,
    mentor_name: 'Mani',
    mentees: 16
  },
  {
    _id: ObjectId('68812847fc2807763ceec528'),
    mentor_id: 8,
    mentor_name: 'Parthiban',
    mentees: 32
  },
  {
    _id: ObjectId('68812847fc2807763ceec52a'),
    mentor_id: 10,
    mentor_name: 'Vijay',
    mentees: 35
  }
]
zen_class>
```

Query 5: Find the number of users who are absent and task is not submitted between 15 Oct 2023 and 31 Oct 2023

Run these step-by-step using MongoDB Shell inside Compass (or your application).

Step 1: Find absentees

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

Step 2: Extract distinct user IDs from absentees

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

zen_class>
```

Step 3: Find users from that list who didn't submit tasks

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

zen_class>
```

Step 4: Count the users

```
zen_class> incompleteUsers.length;
2
zen_class>
```

Conclusion:

This project successfully demonstrates the use of **MongoDB Shell** for managing and analyzing educational data for the Zen Class program. Using shell commands, we were able to write and execute five meaningful queries that provide valuable insights into topic delivery schedules, company drives, user problem-solving stats, mentor effectiveness, and user participation. This proves that MongoDB Shell is a powerful and efficient tool for backend data analysis in educational platforms.