Design database for Zen class programme

* users
* codekata
* attendance
* topics
* tasks
* company\_drives
* mentors

// Insert data into users collection

db.users.insertMany([

{ "\_id": ObjectId("5ff6c36a3620624a22e33da3"), "username": "Dinesh", "email": "dinesh@example.com", "userType": "student" },

{ "\_id": ObjectId("5ff6c36a3620624a22e33da4"), "username": "Saravanan", "email": "saravanan@example.com", "userType": "student" },

{ "\_id": ObjectId("5ff6c36a3620624a22e33da5"), "username": "sanjay", "email": "sanjay@example.com", "userType": "mentor" },

{ "\_id": ObjectId("5ff6c36a3620624a22e33da6"), "username": "prasanth", "email": "prasanth@example.com", "userType": "mentor" },

{ "\_id": ObjectId("5ff6c36a3620624a22e33da7"), "username": "admin", "email": "admin@example.com", "userType": "admin" }

]);

// Insert data into Codekata collection

db.codekata.insertMany([

{ "\_id": ObjectId("5ff6c4b73620624a22e33da8"), "title": "Kata1", "description": "Description for Kata1", "createdAt": new Date("2020-10-05") },

{ "\_id": ObjectId("5ff6c4b73620624a22e33da9"), "title": "Kata2", "description": "Description for Kata2", "createdAt": new Date("2020-10-10") },

{ "\_id": ObjectId("5ff6c4b73620624a22e33daa"), "title": "Kata3", "description": "Description for Kata3", "createdAt": new Date("2020-11-01") }

]);

// Insert data into Attendance collection

db.attendance.insertMany([

{ "\_id": ObjectId("5ff6c5d73620624a22e33dab"), "userID": ObjectId("5ff6c36a3620624a22e33da3"), "date": new Date("2020-10-15"), "status": "present" },

{ "\_id": ObjectId("5ff6c5d73620624a22e33dac"), "userID": ObjectId("5ff6c36a3620624a22e33da4"), "date": new Date("2020-10-15"), "status": "absent" },

{ "\_id": ObjectId("5ff6c5d73620624a22e33dad"), "userID": ObjectId("5ff6c36a3620624a22e33da5"), "date": new Date("2020-10-16"), "status": "present" }

]);

// Insert data into Topics collection

db.topics.insertMany([

{ "\_id": ObjectId("5ff6c6a73620624a22e33dae"), "title": "Topic1", "description": "Description for Topic1", "createdAt": new Date("2020-10-05") },

{ "\_id": ObjectId("5ff6c6a73620624a22e33daf"), "title": "Topic2", "description": "Description for Topic2", "createdAt": new Date("2020-10-12") }

]);

// Insert data into Tasks collection

db.tasks.insertMany([

{ "\_id": ObjectId("5ff6c7773620624a22e33db0"), "userID": ObjectId("5ff6c36a3620624a22e33da3"), "title": "Task1", "description": "Description for Task1", "dueDate": new Date("2020-10-20"), "submissionStatus": "submitted" },

{ "\_id": ObjectId("5ff6c7773620624a22e33db1"), "userID": ObjectId("5ff6c36a3620624a22e33da4"), "title": "Task2", "description": "Description for Task2", "dueDate": new Date("2020-10-25"), "submissionStatus": "not submitted" }

]);

// Insert data into Company Drives collection

db.company\_drives.insertMany([

{ "\_id": ObjectId("5ff6c8163620624a22e33db2"), "title": "Drive1", "date": new Date("2020-10-18"), "location": "CityA" },

{ "\_id": ObjectId("5ff6c8163620624a22e33db3"), "title": "Drive2", "date": new Date("2020-11-02"), "location": "CityB" },

{ "\_id": ObjectId("5ff6c8163620624a22e33db4"), "title": "Drive3", "date": new Date("2020-10-25"), "location": "CityC" }

]);

// Insert data into Mentors collection

db.mentors.insertMany([

{ "\_id": ObjectId("5ff6c8aa3620624a22e33db5"), "userID": ObjectId("5ff6c36a3620624a22e33da5"), "specialization": "Programming"},

{ "\_id": ObjectId("5ff6c8aa3620624a22e33db6"), "userID": ObjectId("5ff6c36a3620624a22e33da6"), "specialization": "Data Science",” mentorName”:”ajay” }

]);

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

db.topics.aggregate([

{

$match: {

createdAt: {

$gte: new Date('2020-10-01'),

$lt: new Date('2020-11-01')

}

}

}

]);

db.tasks.aggregate([

{

$match: {

dueDate: {

$gte: new Date('2020-10-01'),

$lt: new Date('2020-11-01')

}

}

}

]);

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

db.company\_drives.aggregate([

{

$match: {

date: {

$gte: new Date('2020-10-15'),

$lte: new Date('2020-10-31')

}

}

}

]);

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

db.company\_drives.aggregate([

{

$lookup: {

from: 'attendance',

localField: '\_id',

foreignField: 'companyDrive.id',

as: 'attendances'

}

},

{

$match: {

'attendances.status': 'present'

}

}

]);

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

db.codekata.aggregate([

{

$lookup: {

from: 'tasks',

localField: '\_id',

foreignField: 'codekata.id',

as: 'tasks'

}

},

{

$group: {

\_id: '$userID',

totalProblemsSolved: { $sum: { $size: '$tasks' } }

}

}

]);

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

db.mentors.aggregate([

{

$lookup: {

from: 'users',

localField: '\_id',

foreignField: 'mentors.id',

as: 'mentors'

}

},

{

$group: {

\_id: '$\_id',

mentorName: { $first: '$mentors.mentorName' },

menteesCount: { $sum: { $size: '$mentors' } }

}

},

{

$match: {

menteesCount: { $gt: 15 }

}

}

]);

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

db.users.aggregate([

{

$lookup: {

from: 'attendance',

localField: '\_id',

foreignField: 'userID',

as: 'attendance'

}

},

{

$lookup: {

from: 'tasks',

localField: '\_id',

foreignField: 'userID',

as: 'tasks'

}

},

{

$match: {

'attendance.status': 'absent',

'tasks.submissionStatus': 'not submitted',

'attendance.date': {

$gte: new Date('2020-10-15'),

$lte: new Date('2020-10-31')

}

}

}

]);