**MONGO-DB-TASK-2**

**Database Schema Design:**

1. **Collection**: **Users**

{

user\_id (Primary Key, Integer, Auto-increment)

username (VARCHAR, Unique)

email (VARCHAR, Unique)

password\_hash (VARCHAR)

full\_name (VARCHAR)

role (ENUM: 'student', 'mentor', 'admin')

}

1. **Collection: CodeKata**

{

codekata\_id (Primary Key, Integer, Auto-increment)

title (VARCHAR)

description (TEXT)

difficulty (ENUM: 'easy', 'medium', 'hard')

created\_at (DATETIME)

updated\_at (DATETIME)

}

1. **Collection: Attendance**

{

attendance\_id (Primary Key, Integer, Auto-increment)

user\_id (Foreign Key, References Users(user\_id))

class\_date (DATE)

status (ENUM: 'present', 'absent', 'late')

Created\_At (DATE)

}

1. **Collection: Topics**

{

topic\_id (Primary Key, INT, Auto Increment)

title (VARCHAR)

description (TEXT)

created\_at (DATETIME)

updated\_at (DATETIME)

}

1. **Collection: Tasks:**

{

task\_id (Primary Key, INT, Auto Increment)

title (VARCHAR)

description (TEXT)

due\_date (DATE)

created\_at (DATETIME)

updated\_at (DATETIME)

}

1. **Collection: Company\_Drives:**

**{**

drive\_id (Primary Key, INT, Auto Increment)

title (VARCHAR)

link (VARCHAR)

description (TEXT)

uploaded\_by (Foreign Key to Users.user\_id, INT)

uploaded\_at (DATETIME)

**}**

1. **Collection: Mentors:**

**{**

mentor\_id (Primary Key, INT, Auto Increment)

user\_id (Foreign Key to Users.user\_id, INT)

specialization (VARCHAR)

biography (TEXT)

created\_at (DATETIME)

updated\_at (DATETIME)

**}**

**QUESTIONS OF MONGO DB**

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

**Query:**

db.topics\_tasks.find({ month: "October" });

db.topics\_topics.find({ month: "October" });

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

**Query:**

db.company\_drives.find({

date: {

$gte: ISODate("2020-10-15"),

$lte: ISODate("2020-10-31")

}

})

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

**Query:**

db.company\_drives.aggregate([

{

$lookup: {

from: "students",

localField: "\_id",

foreignField: "driveId",

as: "students"

}

}

])

Db.CompanyDrive.find() .populate({ path: 'students',

select: 'name'})

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

**Query:**

Db.Codekata.find({codekatacount:1})

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

**Query:**

db.mentor.aggregate([

{

$group: {

\_id: "$mentorId",

menteeCount: { $sum: 1 }

}

},

{

$match: {

menteeCount: { $gt: 15 }

}

}

])

db.mentor.aggregate([

{

$group: {

\_id: "$mentorId",

menteeCount: { $sum: 1 }

}

},

{

$match: {

menteeCount: { $gt: 15 }

}

},

{

$lookup: {

from: "mentors",

localField: "\_id",

foreignField: "mentorId",

as: "mentorDetails"

}

},

{

$unwind: "$mentorDetails"

},

{

$project: {

\_id: 0,

mentorId: "$\_id",

mentorName: "$mentorDetails.name",

menteeCount: 1

}

}

])

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

**Query:**

**//Attendance List**

db.attendance.aggregate([

{

$match: {

status: 'absent',

date: { $gte: ISODate('2020-10-15'), $lte: ISODate('2020-10-31') }

}

},

{

$group: { \_id: "$user\_id" }

}

]);

**//Submitted Task List**

db.tasks.aggregate([

{

$match: {

submission\_date: { $gte: ISODate('2020-10-15'), $lte: ISODate('2020-10-31') }

}

},

{

$group: { \_id: "$user\_id" }

}

]);

**// Attendance Aggregate:**

db.attendance.aggregate([

{

$match: {

status: 'absent',

date: { $gte: ISODate('2020-10-15T00:00:00Z'), $lte: ISODate('2020-10-31T23:59:59Z') }

}

},

{

$group: { \_id: "$user\_id" }

},

{

$lookup: {

from: "tasks",

localField: "\_id",

foreignField: "user\_id",

as: "tasks"

}

},

{

$match: {

"tasks": { $eq: [] } // No tasks submitted

}

},

{

$count: "absent\_and\_not\_submitted\_users"

}

]);