**CODING CHALLENGE**

**Understanding Relationships in MongoDB**

Scenario: **Learning Management System**

We will model a simple Learning Management System with the following collections:

1. **students**
2. **profiles**
3. **courses**
4. **enrollments**

DB: LMS

db.createCollection(“students”)

db.students.insertMany([

{ \_id: 1, name: "Alice" },

{ \_id: 2, name: "Bob" },

{ \_id: 3, name: "Charlie" },

{ \_id: 4, name: "Diana" }

])

db.createCollection("profiles")

db.profiles.insertMany([

{ \_id: 101, student\_id: 1, bio: "Loves Math and Coding" },

{ \_id: 102, student\_id: 2, bio: "Enjoys Physics and Astronomy" },

{ \_id: 103, student\_id: 3, bio: "Passionate about Chemistry" },

{ \_id: 104, student\_id: 4, bio: "Interested in Biology and Art" }

]);

db.createCollection("courses")

db.courses.insertMany([

{ \_id: "C1", title: "Math 101" },

{ \_id: "C2", title: "Physics 101" },

{ \_id: "C3", title: "Biology 101" }

]);

db.createCollection("enrollments")

db.enrollments.insertMany([

{ student\_id: 1, course\_id: "C1" }, // Alice - Math

{ student\_id: 1, course\_id: "C2" }, // Alice - Physics

{ student\_id: 2, course\_id: "C2" }, // Bob - Physics

{ student\_id: 3, course\_id: "C3" }, // Charlie - Biology

{ student\_id: 4, course\_id: "C1" }, // Diana - Math

{ student\_id: 4, course\_id: "C3" } // Diana - Biology

]);

**Questions:**

1. **Find the bio of the student named "Diana". (One-to-One)**

db.students.aggregate([

{

$match: { name: "Diana" }

},

{

$lookup: {

from: "profiles",

localField: "\_id",

foreignField: "student\_id",

as: "profile"

}

},

{

$project: {

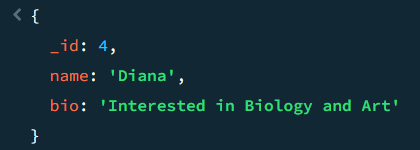
name: 1,

bio: { $arrayElemAt: ["$profile.bio", 0] }

}

}

]);



1. **List all courses that Alice is enrolled in. (One-to-Many)**

db.students.aggregate([

{ $match: { name: "Alice" } },

{

$lookup: {

from: "enrollments",

localField: "\_id",

foreignField: "student\_id",

as: "enrollments"

}

},

{

$lookup: {

from: "courses",

localField: "enrollments.course\_id",

foreignField: "\_id",

as: "course"

}

},

{

$project: {

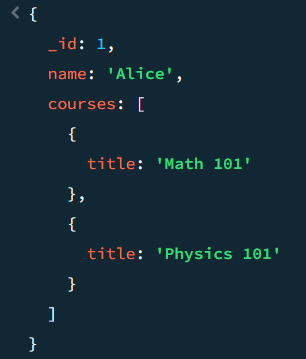
name: 1,

"courses.title": 1

}

}

]);



1. **Find all students enrolled in "Biology 101". (One-to-Many)**

db.courses.aggregate([

{ $match: { title: "Biology 101" } },

{

$lookup: {

from: "enrollments",

localField: "\_id",

foreignField: "course\_id",

as: "enrollments"

}

},

{ $unwind: "$enrollments" },

{

$lookup: {

from: "students",

localField: "enrollments.student\_id",

foreignField: "\_id",

as: "student"

}

},

{

$project: {

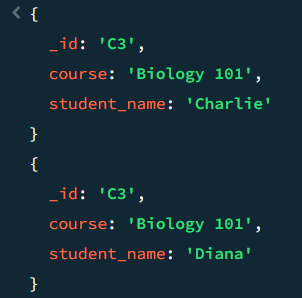
course: "$title",

student\_name: { $arrayElemAt: ["$student.name", 0] }

}

}

]);



1. **Use $lookup to join students with their profiles. (Join for One-to-One)**

db.students.aggregate([

{

$lookup: {

from: "profiles",

localField: "\_id",

foreignField: "student\_id",

as: "profile"

}

},

{

$project: {

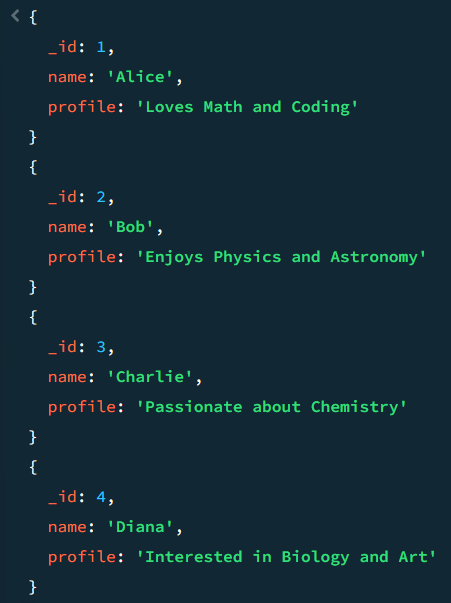
name: 1,

profile: { $arrayElemAt: ["$profile.bio", 0] }

}

}

]);



1. **Show each student's name along with the titles of all courses they are enrolled in. (Join for Many-to-Many)**

db.students.aggregate([

{

$lookup: {

from: "enrollments",

localField: "\_id",

foreignField: "student\_id",

as: "enrollments"

}

},

{ $unwind: "$enrollments" },

{

$lookup: {

from: "courses",

localField: "enrollments.course\_id",

foreignField: "\_id",

as: "course"

}

},

{

$project: {

name: 1,

course: { $arrayElemAt: ["$course.title", 0] }

}

}

]);