**Complex MongoDB queries**

**Work Sheet**

**Usage of commands**

**1. $regex (Regular Expression)**

We can use $regex to search for documents where a specific field matches a regular expression pattern.

This can be case-sensitive or case-insensitive depending on the flags used.

**1: Case-insensitive search for names containing "john**

db.employees.find({ name: { "$regex": "john", "$options": "i" } })

**2: Match names that start with "A"**

db.employees.find({ name: { "$regex": "^A" } })

**3: Match names containing digits**

db.employees.find({ name: { "$regex": "[0-9]" } })

**2.$all (Matches All Elements of an Array)**

$all matches an array field that contains all the specified elements, and it doesn't matter in which order they appear.

**1: Find employees who have both "Java" and "MongoDB" in their skills**

db.employees.find({ skills: { "$all": ["Java", "MongoDB"] } })

**3. $size (Matches Array Length)**

$size checks the length of an array field and returns documents where the length of the array matches the specified value.

**1: Find employees with exactly 3 skills**

db.employees.find({ skills: { "$size": 3 } })

**4.$nin (Not in a Set of Values)**

$nin matches documents where the field's value is not in the list of values provided.

**1: Find employees who do not have "Java" in their skills**

db.employees.find({ skills: { "$nin": ["Java"] } })

**2: Find projects not related to "Python" or "Ruby"**

db.projects.find({ technologies: { "$nin": ["Python", "Ruby"] } })

**5.count()**

the **count** function is used to **find the number of documents** that match a specific query. It is useful when you want to know how many documents meet certain conditions without retrieving all the documents.

**1.Finds how many employees work in the Research department.**

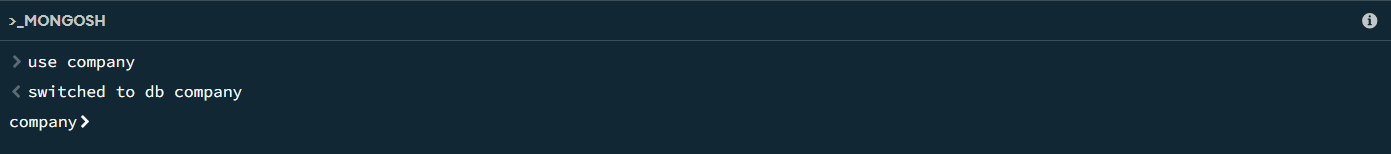
db.employees.find({ department: "Research" }).count()

**Step-by-Step Process**

**Step 1: Create a Database (company)**

This command switches to the company database. If it doesn't exist, MongoDB will create it when data is added.

**use company**



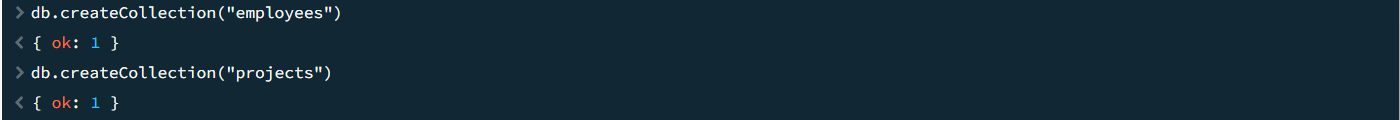
**Step 2: Create Collections (employees and projects)**

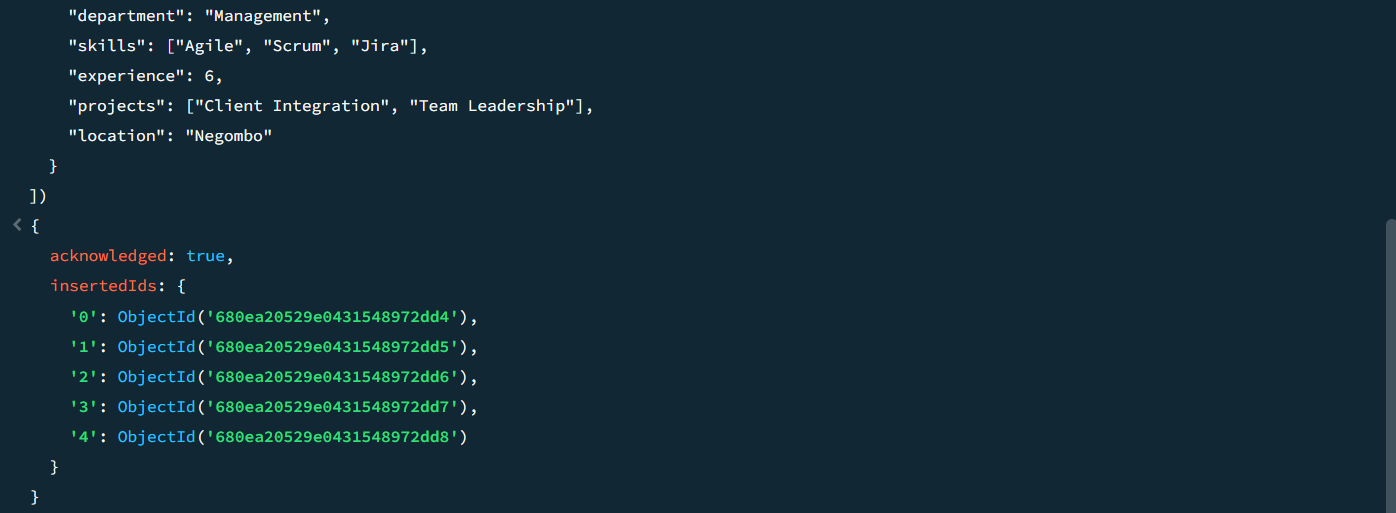
This command creates the employees collection where employee data will be stored. In MongoDB, collections are created automatically when you insert data, but we can create them manually using createCollection.

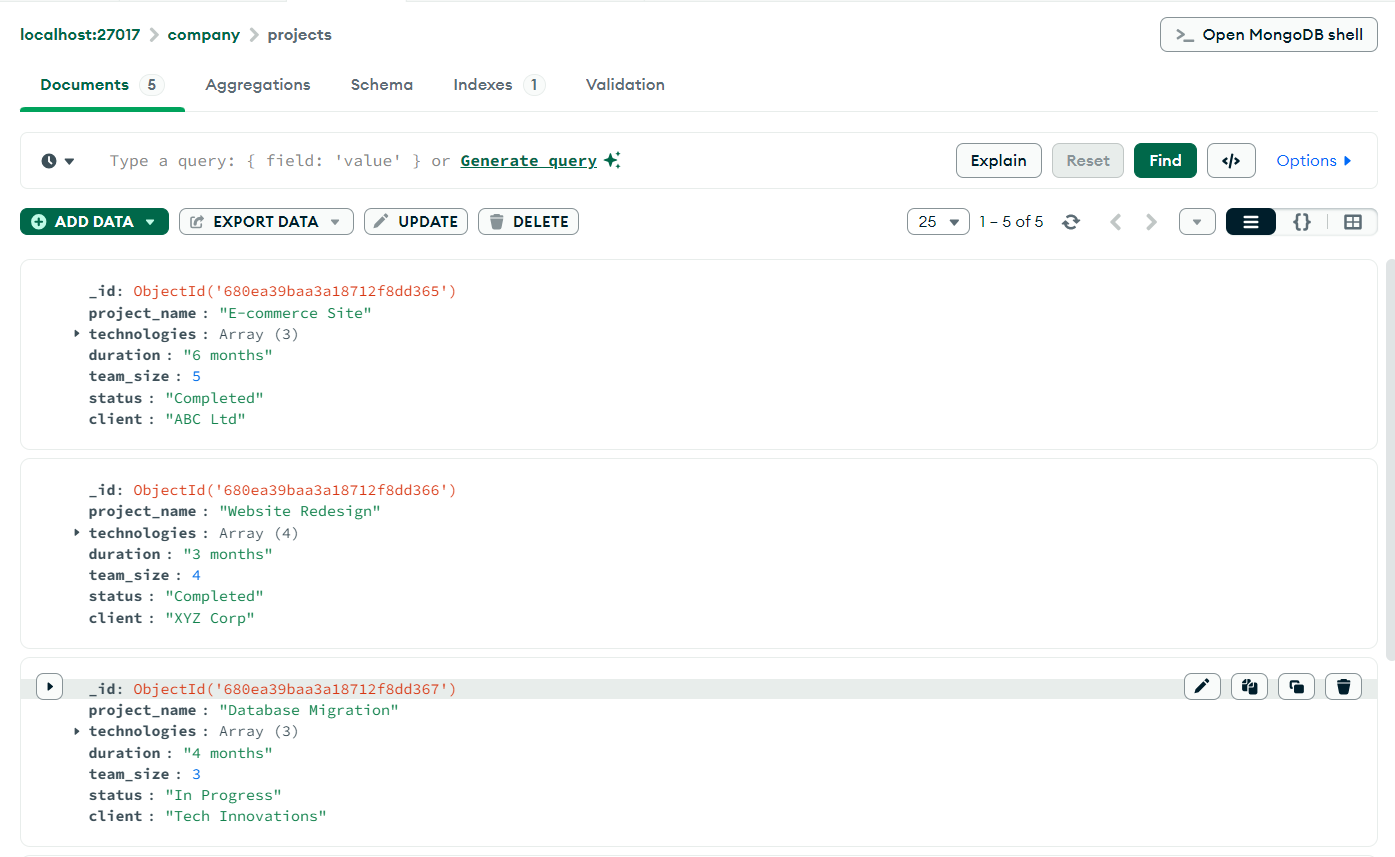
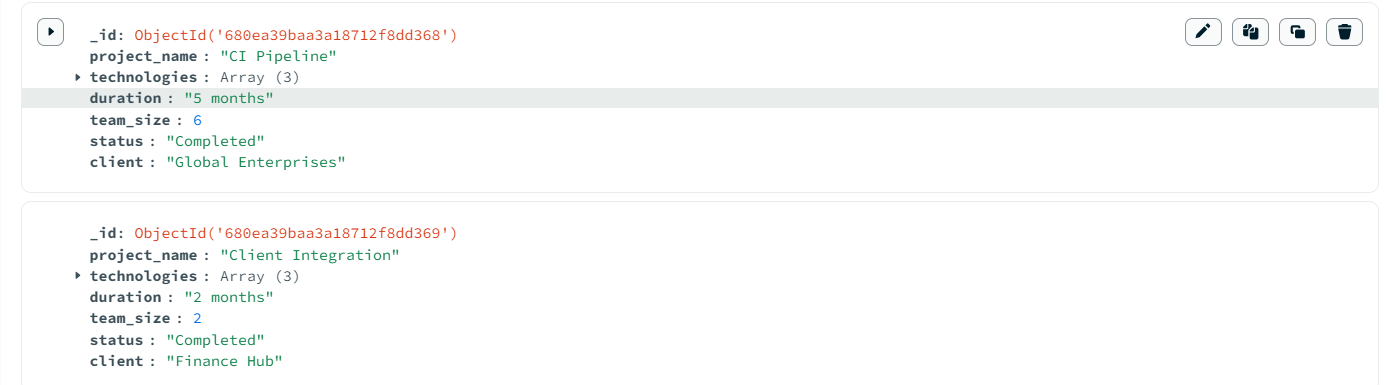
**db.createCollection("employees")**

Similarly, this command creates the projects collection where project data will be stored.

**db.createCollection("projects")**



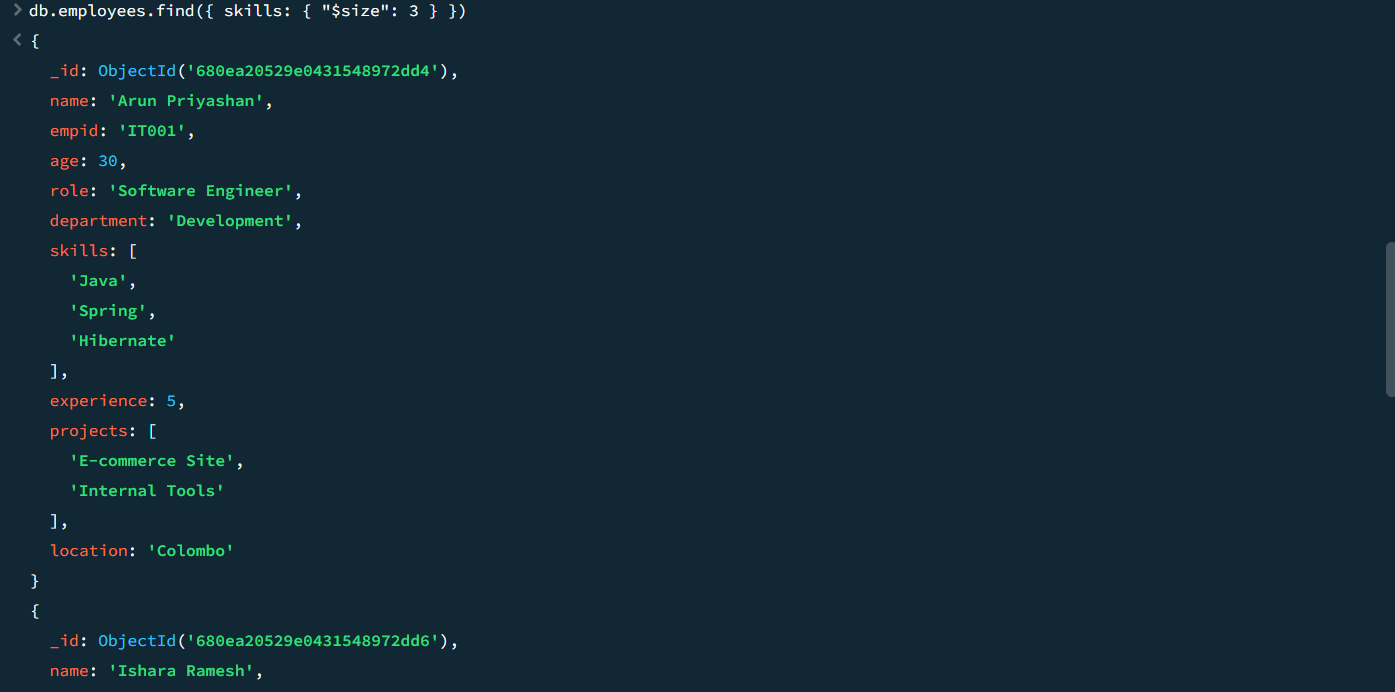
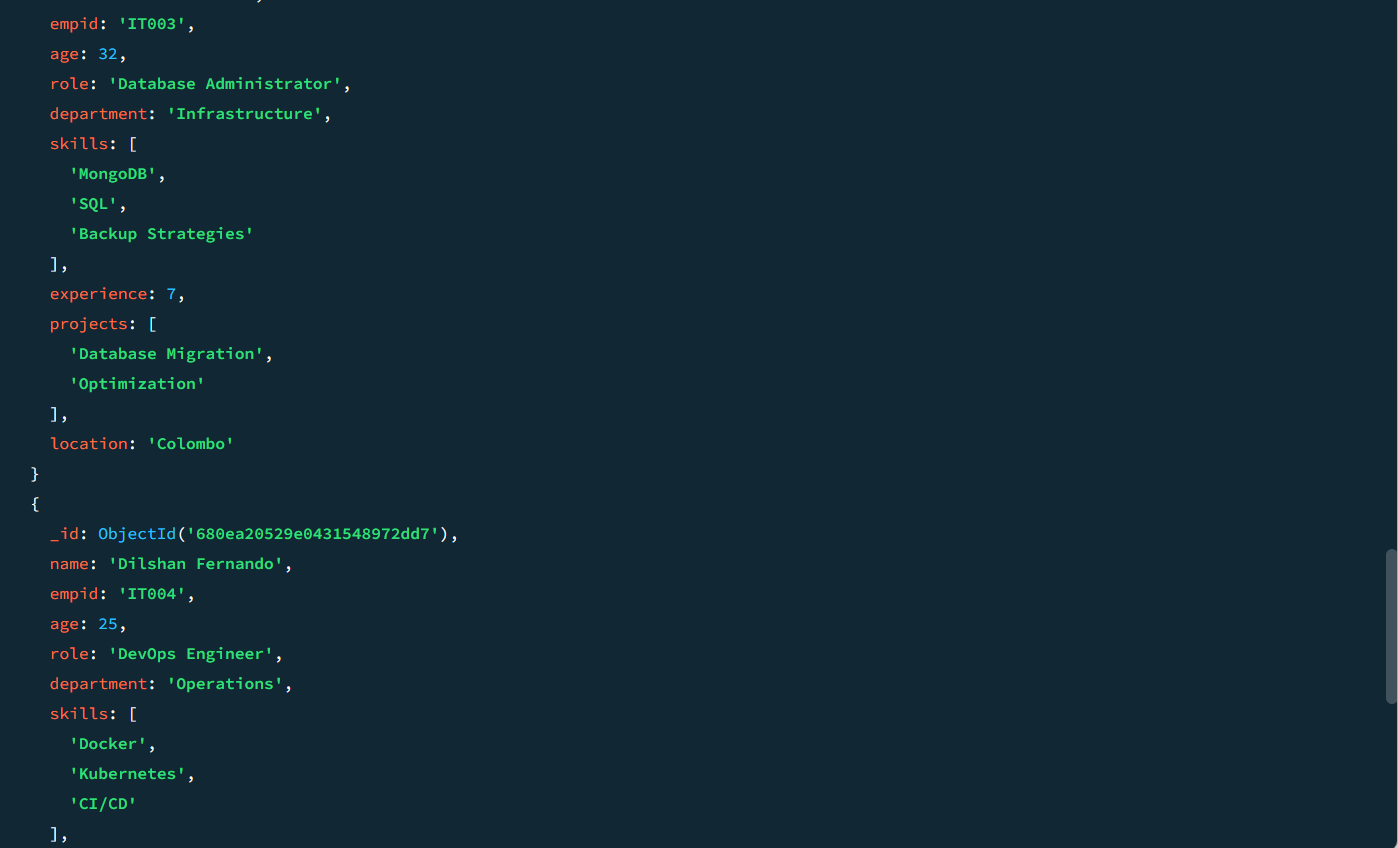
**Step 3: Insert Sample Data into employees Collection (Using Console)** 

**Step 4: Insert Sample Data Into Projects Collection (Using Compass)** 

**Step 5: Perform Complex Queries Using MongoDB Operators**

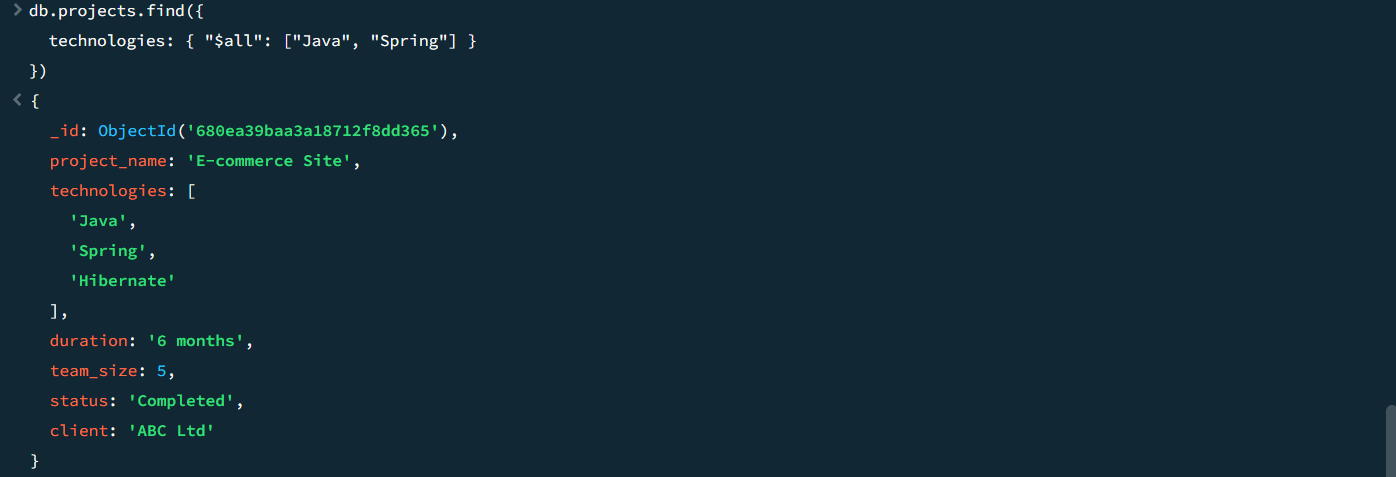
1. Find Employees with Exactly 3 Skills

**db.employees.find({ skills: { "$size": 3 } })**

2. Find Projects Using Both "Java" and "Spring"

**db.projects.find({ technologies: { "$all": ["Java", "Spring"] }})**



3.Find Employees Who Have "MongoDB" in Their Skills and Name Starts with "A"

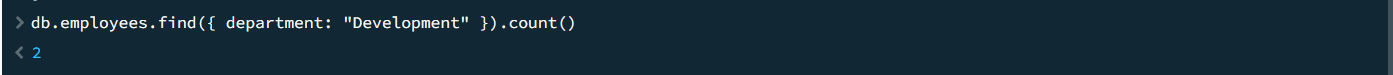
**db.employees.find({**

**skills: { "$in": ["MongoDB"] },**

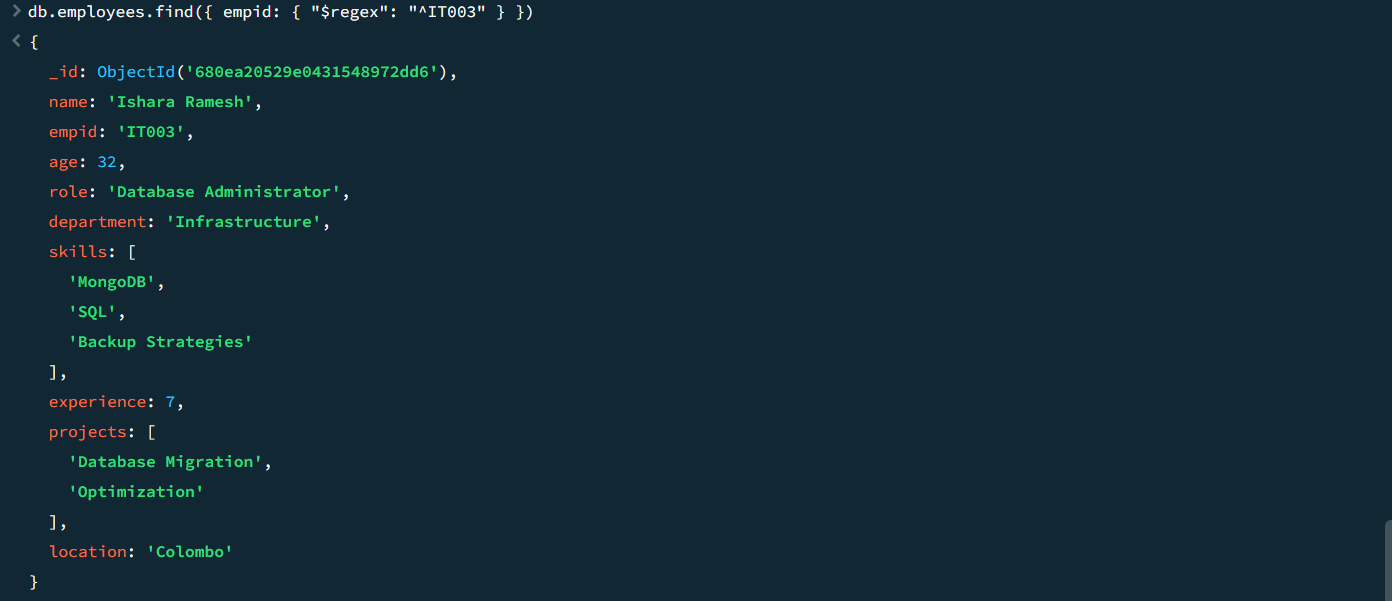
**name: { "$regex": "^A", "$options": "i" } })**

4. Finds how many employees work in the Research department.

**db.employees.find({ department: "Development" }).count()**

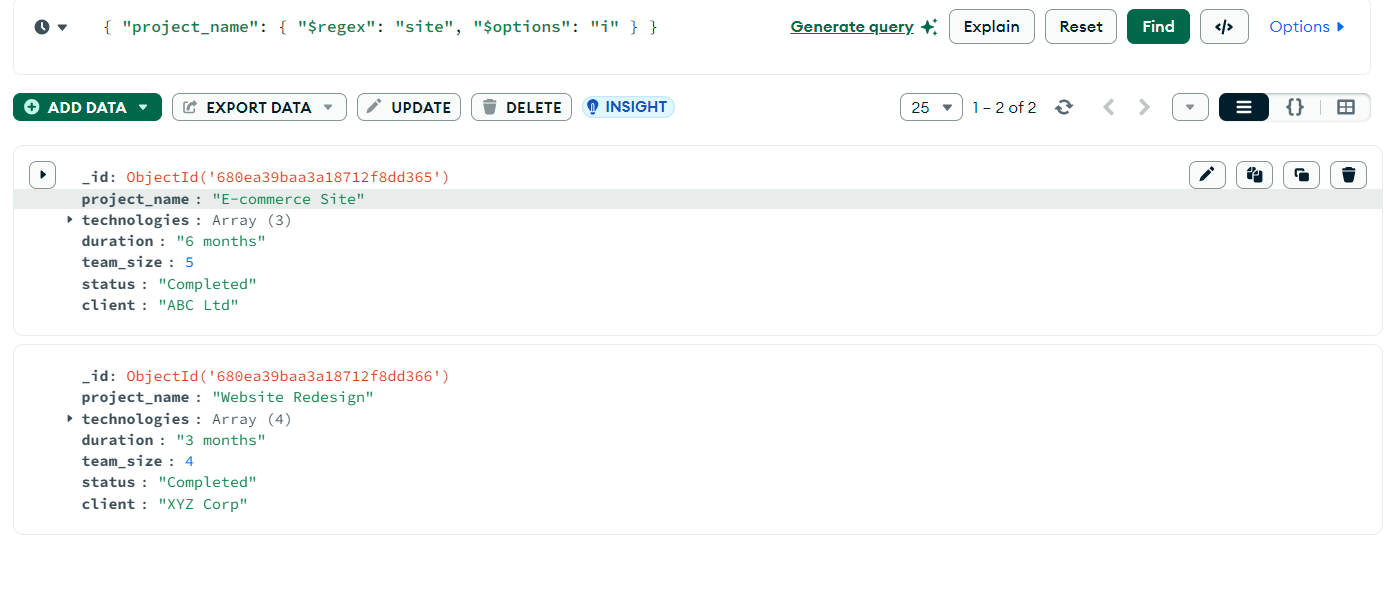


5.Finds employees whose empid starts with "IT003".

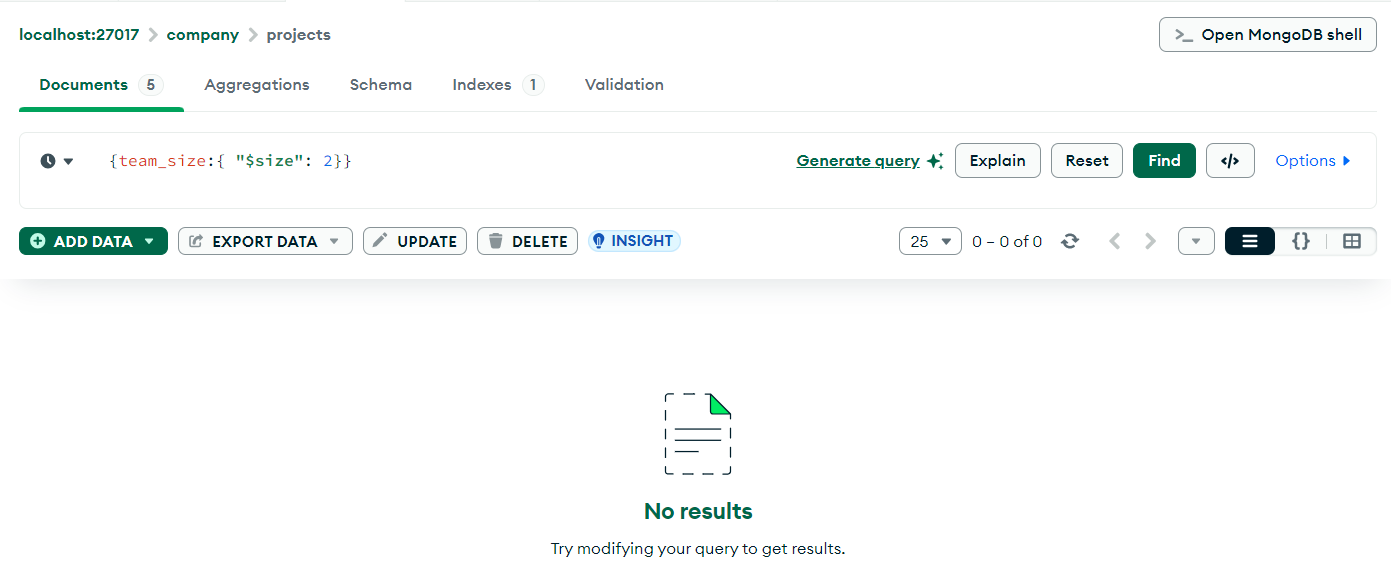
**db.employees.find({ empid: { "$regex": "^IT003" } })**

6.Find what are the project names containing "site"

**db.projects.find({ project\_name: { "$regex": "site", "$options": "i" } })**

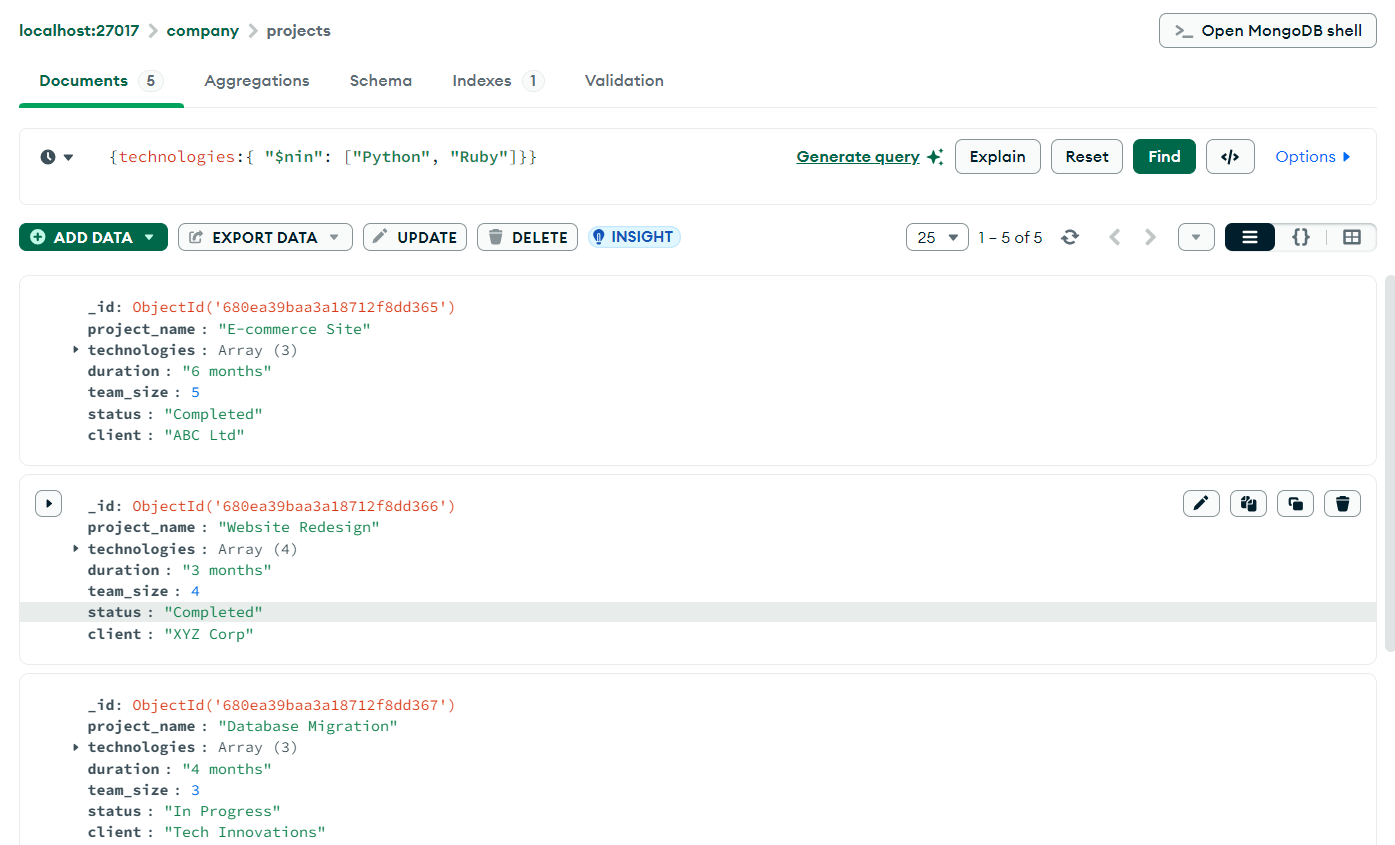


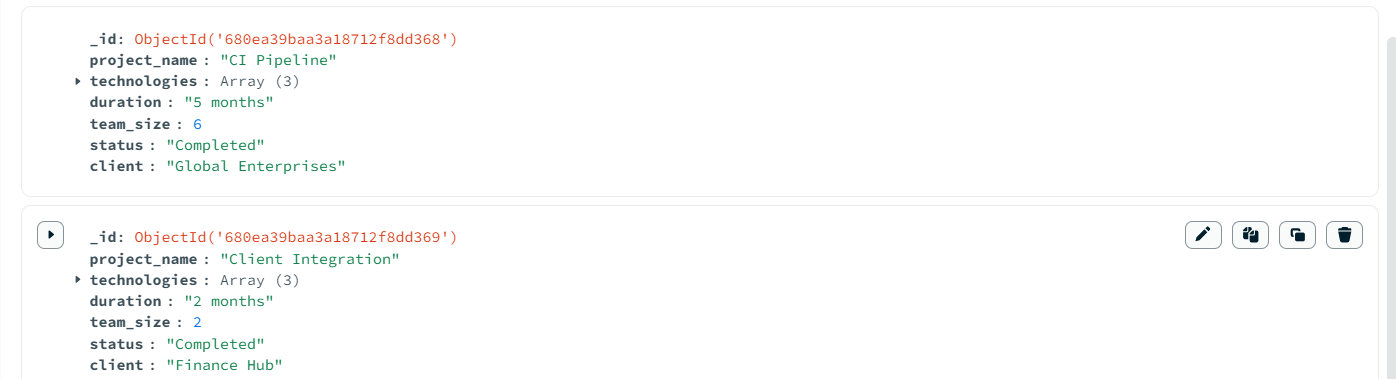
7. Find projects that use exactly 3 as team\_size

**db.projects.find({ team\_size: { "$size": 3 } })**

8. Find projects that do not use "Python" or "Ruby"

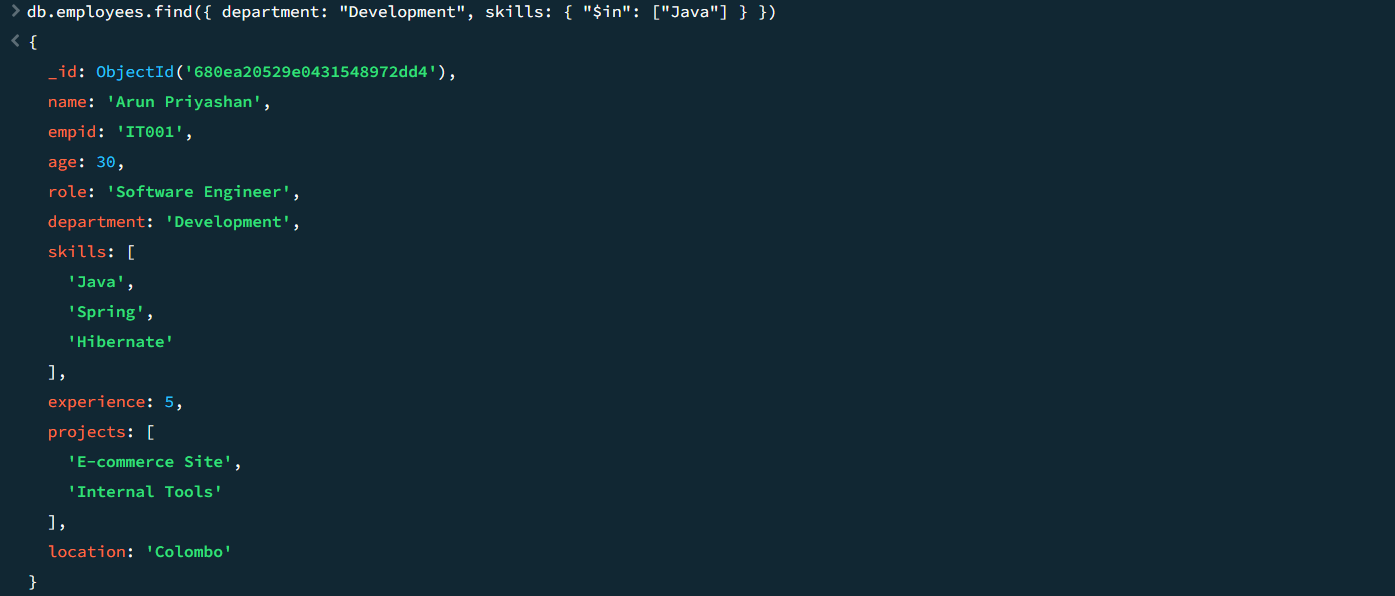
**db.projects.find({ technologies: { "$nin": ["Python", "Ruby"] } })**





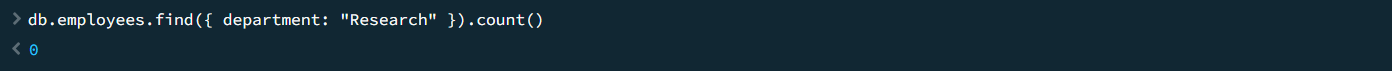
9.Finds employees in the Development department with "Java" skill.

**db.employees.find({ department: "Development", skills: { "$in": ["Java"] } })**

****

10.Finds how many employees work in the Research department.

**db.employees.find({ department: "Research" }).count()**



**Conclusion for the MongoDB Complex Queries Worksheet:**

The **step-by-step process** you outlined for working with **MongoDB collections** (such as employees and projects) is well-structured and provides a solid foundation for understanding how to use MongoDB operators like $regex, $all, $size, $nin, and count for complex queries. These commands are essential for filtering, matching, and analyzing data efficiently.