# TASK – Use Case 2

## Indexing Various Devices in IoT Platform

### AIM:

To design and implement a MongoDB document database for indexing various IoT devices based on their attributes such as device type, location, and temperature. The goal is to efficiently retrieve and analyze device records using single and multiple indexes.

### STEPS:

Step 1: Install MongoDB and Mongo Shell from the official MongoDB website.

Step 2: Open the Mongo Shell or MongoDB Compass.

Step 3: Create a new database named IoTPlatform.

Step 4: Create a collection named DeviceRecords.

Step 5: Insert multiple IoT device records into the collection.

Step 6: Create indexes on key attributes like DeviceID, LocationID, and Temperature.

Step 7: Perform CRUD and indexing operations to demonstrate efficient querying.

### QUERIES:

#### Create a new database

use IoTPlatform

#### Create a new collection

db.createCollection("DeviceRecords")

#### Insert multiple IoT device documents

db.DeviceRecords.insertMany([  
 {  
 DeviceID: "D001",  
 DeviceType: "Temperature Sensor",  
 LocationID: "L001",  
 Location: "Chennai",  
 Temperature: 30.5,  
 Humidity: 70,  
 Status: "Active"  
 },  
 {  
 DeviceID: "D002",  
 DeviceType: "Humidity Sensor",  
 LocationID: "L002",  
 Location: "Bangalore",  
 Temperature: 28.0,  
 Humidity: 80,  
 Status: "Active"  
 },  
 {  
 DeviceID: "D003",  
 DeviceType: "Motion Detector",  
 LocationID: "L003",  
 Location: "Coimbatore",  
 Temperature: 26.5,  
 Humidity: 60,  
 Status: "Inactive"  
 },  
 {  
 DeviceID: "D004",  
 DeviceType: "Light Sensor",  
 LocationID: "L004",  
 Location: "Madurai",  
 Temperature: 32.1,  
 Humidity: 65,  
 Status: "Active"  
 },  
 {  
 DeviceID: "D005",  
 DeviceType: "Smoke Detector",  
 LocationID: "L005",  
 Location: "Trichy",  
 Temperature: 33.8,  
 Humidity: 55,  
 Status: "Active"  
 }  
])

#### Creating Indexes

db.DeviceRecords.createIndex({ DeviceID: 1 })  
db.DeviceRecords.createIndex({ LocationID: 1 })  
db.DeviceRecords.createIndex({ Temperature: 1, Humidity: 1 })  
db.DeviceRecords.getIndexes()

#### Querying Using Indexes

db.DeviceRecords.find({ Location: "Chennai" }).pretty()  
db.DeviceRecords.find({ Temperature: { $gt: 30 } }).pretty()  
db.DeviceRecords.find({ Temperature: { $gte: 28 }, Humidity: { $lte: 70 } }).pretty()  
db.DeviceRecords.find().sort({ Temperature: 1 })  
db.DeviceRecords.aggregate([ { $group: { \_id: "$Status", count: { $sum: 1 } } } ])

#### Updating and Deleting Documents

db.DeviceRecords.updateOne(  
 { DeviceID: "D003" },  
 { $set: { Status: "Active" } }  
)  
db.DeviceRecords.deleteOne({ Status: "Inactive" })

### RESULT:

Thus, the MongoDB document database for IoT device indexing was successfully designed and implemented. Indexes on attributes such as DeviceID, LocationID, and Temperature improved query performance, enabling efficient retrieval and analysis of IoT device data.