

Use Case II

## Indexing Various Devices in IoT Platform

Aim:-

To design and implement a data storage mechanism for an IoT Platform that can efficiently store, retrieve, and index data from a wide range of heterogeneous devices. The mechanism should support fast querying and flexible schema to handle new device types easily.

Explanation:-

A generic IoT Platform collects data from different types of devices such as temperature sensors, humidity sensors, and energy meters. Each device generates data in various formats and at different intervals. A traditional relational database would require a fixed schema, making it difficult to handle new or variable device data structures.

To overcome this issue, a JSON-based documents-oriented database like MongoDB is preferred. MongoDB stores data in flexible JSON-like documents, allowing each device type to store its own data attributes. This makes the system adaptable and scalable.

Why MongoDB?

- Schema-less structure supports new device types dynamically
- Documents are stored in JSON format, suitable for IoT data
- Supports indexing on single and multiple fields

→ Allows fast retrievals of data for specific devices or parameters

→ Scalable and efficient for handling large volumes of real-time data.

### Example JSON Documents:

#### Temperature Sensor Example:

```
{  
    "device_id": "D1001",  
    "device_type": "Temperature Sensor",  
    "location_id": "L001",  
    "timestamp": "2025-11-03T10:45:00Z",  
    "data": {  
        "temperature": 28.5,  
        "humidity": 70  
    },  
    "status": "active"  
}
```

#### Energy Meter Example:

```
{  
    "device_id": "E2001",  
    "device_type": "Energy Meter",  
    "location_id": "L001",  
    "timestamp": "2025-11-03T10:46:00Z",  
    "data": {  
        "Voltage": 220,  
        "Current": 1.5  
    }  
}
```

```
"power": 330
```

```
},
```

```
"status": "active"
```

```
}
```

## Index Creation in MongoDB:-

|| Create single and compound indexes.

```
db.iot-devices.createIndex({ "device_id": 1});
```

```
db.iot-devices.createIndex({ "device_id": 1, "location_id": 1});
```

```
db.iot-devices.createIndex({ "data.temperature": 1});
```

## Sample Queries:-

1. Retrieve all records of a particular device:

```
db.iot-devices.find({ "device_id": "D1001" });
```

2. Retrieve data for a device in a particular location:

```
db.iot-devices.find({ "device_id": "1", "location_id": 1 });
```

3. Retrieve temperature reading from all devices?

```
db.iot-devices.find({ "data.temperature": { $exists: true } });
```

Result:- Using MongoDB allows the IoT platform to store and retrieve data from devices without modifying the database structure. Indexes help in fast query execution especially when searching data by device location or parameters like temperature. The system remains scalable and adaptable to new device types introduced in the future.