Use case II

Indexing various services 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 document oriented database like mongoDB is preferred. Mongo DB stores data in flexible JSON-like documents, allowing each device type to store its own data attributes. The makes the system adaptable and scalable.

Why mongo DB?

→ 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": "100",
"timestamp": "2025-11-03T10:45:00Z",
"data": {
"temperature": 2605,
"humidity": 70
},
"status": "active"
}
```

Energy meter Example:

```
{
"device_id": "E-2001",
"device_type": Energy meter",
"location_id": "1001",
"timestamp": "2025-11-03 T10:46:00Z",
"data": {
"voltage": 220,
"current": 1.5,
"power": 330
},
"status": "active"
}
```

Index creation in mongo DB:-

"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":"1001"});

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

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

3. Retrieve temperature reading from all devices

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

Result:- using mongo DB allows the IoT platform to store and retrieve data from devices without modifying the database structure. Indexes helps 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.