

Date:

USECASE: Indexing various device in IOT platform

Aim:

An IOT platform collects and manages data from multiple heterogeneous devices - sensors, meters, controllers. The data storage must be flexible, scalable and capable of index-based querying for high speed retrieval.

A JSON-based document database such as MongoDB is ideal for this situation.

Reason for choosing MongoDB:

Requirement	MongoDB Advantage
Flexible schema	stores data in JSON-like BSON documents, no fixed schema needed
Indexing on multiple fields	MongoDB supports compound indexes and secondary indexes for faster
Scalability	supports horizontal scaling to handle millions of device records
Fast retrieval	Indexed queries and aggregation pipelines make data retrieval fast.

Example SQL schema:

```
CREATE TABLE devices(device_id varchar(20) PRIMARY KEY,  
device_type varchar(50), manufacturer varchar(50));
```

```
CREATE TABLE locations(location_id varchar(20) PRIMARY KEY,  
city varchar(50), latitude DECIMAL(9,6), longitude DECIMAL(9,6));
```

```
CREATE TABLE sensor_data(record_id int auto-increment  
PRIMARY KEY, device_id varchar(20), location_id varchar(20),  
temperature DECIMAL(5,2), humidity DECIMAL(5,2),
```

Query INPUT:

device_id	location_id	temperature
D101	LOC01	31.5
D203	LOC05	32.1

timestamp DATETIME, FOREIGN KEY(device_id) REFERENCES devices(device_id), FOREIGN KEY(location_id) REFERENCES location(location_id));

Indexes for faster retrieval:

create index idx_device_location ON sensorData (device_id, location_id);

create index idx_temperature ON sensorData (temperature);

Normalization Justification:

INPUT: select device_id, location_id, temperature FROM sensorData WHERE temperature > 30;

All tables are normalized up to 3NF:

1NF: Atomic attribute values.

2NF: No partial dependencies.

3NF: No transitive dependencies

This ensures data integrity and avoids redundancy while supporting fast indexing.