

Modelo de datos NoSQL-ACTIVIDADES.

Explicación de la Tarea:

En el ámbito de la gestión de datos, especialmente en aplicaciones que requieren almacenamiento y consulta eficiente de grandes volúmenes de información, los modelos de bases de datos NoSQL han ganado relevancia. Estos modelos ofrecen flexibilidad, escalabilidad y rendimiento en escenarios donde los modelos de bases de datos relacionales tradicionales pueden resultar limitados.

En este contexto, nos enfocaremos en un caso de uso específico: la gestión de datos de generación y consumo de energía por barrio. Este caso de uso es relevante en aplicaciones de monitorización y gestión energética, donde se requiere almacenar y consultar datos de manera eficiente para optimizar el uso de recursos y mejorar la eficiencia energética a nivel local.

Para abordar este caso de uso, proponemos la aplicación de varios patrones de diseño de MongoDB, una base de datos NoSQL ampliamente utilizada y conocida por su capacidad para manejar grandes volúmenes de datos de manera eficiente. Estos patrones están diseñados para mejorar el rendimiento de las consultas, la eficiencia del almacenamiento y la facilidad de mantenimiento de la base de datos.

Patrones de Diseño MongoDB para Optimizar la Estructura de la Base de Datos

1. Patrón de Documentos Atributos (Attribute Pattern)

Descripción: Útil cuando los datos de un documento varían o cuando hay atributos presentes en un pequeño subconjunto de los datos, como variaciones en la producción y consumo de energía.

Aplicación: Incluir un campo "atributos" en los documentos de energía para almacenar información variante como tipo de día (laboral, fin de semana, festivo), condiciones climáticas, etc.

2. Patrón de Resumen (Summary Pattern)

Descripción: Evita cálculos costosos al mantener totales acumulados, como energía generada y consumida para un barrio, evitando recálculos en cada consulta.

Aplicación: Agregar campos de resumen al nivel del barrio en cada documento de energía o en documentos separados para almacenar totales diarios de energía generada, consumida y necesitada de la red para cada barrio.

3. Patrón de Bucket (Bucket Pattern)

Descripción: Agrupa datos que se recogen o se utilizan juntos, especialmente útil para datos temporales como mediciones de energía.

Aplicación: Crear "buckets" o "cubos" de datos por día para cada barrio, donde cada "bucket" contiene todas las mediciones de energía de las viviendas en ese barrio para ese día, facilitando la recuperación de medidas acumuladas.

4. Patrón de Árbol (Tree Pattern)

Descripción: Almacena estructuras jerárquicas, como la ubicación de viviendas dentro de los barrios y las ciudades.

Aplicación: Utilizar un enfoque jerárquico para organizar documentos, donde cada ciudad contiene barrios, y cada barrio contiene viviendas, facilitando las consultas por ubicación.

Estos patrones de diseño de MongoDB ayudarán a optimizar la estructura de la base de datos para el caso de uso descrito, mejorando el rendimiento de las consultas, la eficiencia del almacenamiento y la facilidad de mantenimiento.

Estructura Propuesta de la Base de Datos

Para optimizar la estructura de la base de datos para el caso de uso descrito, propongo la siguiente estructura que combina varios patrones de diseño de MongoDB para facilitar la recuperación de los totales acumulados de energía por barrio

```
{
  "_id": ObjectId("6316fc1597eb703de2add36e"),
  "fecha": ISODate("2022-10-26T00:00:00.000Z"),
  "barrio": "Carrús",
  "ciudad": "Elche",
  "resumen_diario": {
    "total_consumo": 11000, // kW acumulados durante el día
    "total_generado": 6000,
    "total_necesidad_red": 5000
  },
  "detalles": [
    {
      "propietario_id": ObjectId("6317048697eb703de2add36f"),
      "nombre": "Aitor Medrano",
      "direccion": "Calle Secreta, 123",
      "consumo": 11, // kW en una hora específica
      "generado": 6,
      "necesidad_red": 5
    }
  ]
  // Más documentos de viviendas en el mismo barrio para ese día
}
```

Explicación:

- **Patrón de Resumen:** `resumen_diario` proporciona un resumen a nivel de barrio para evitar cálculos en cada consulta.
- **Patrón de Bucket:** Agrupa los datos de energía de todas las viviendas de un barrio en un día específico.

- **Patrón de Árbol:** Organiza los datos de manera jerárquica (ciudad > barrio > vivienda).

Esta estructura permite consultas eficientes para recuperar los totales diarios por barrio y facilita la implementación del sistema de bonos basado en la producción de energía excedente a nivel de barrio.