Pregunta 01: Manejo de relaciones en MongoDB

MongoDB maneja las relaciones entre documentos mediante dos enfoques principales: embebidos y referenciados.

1. **Enfoque embebido:** Los datos relacionados se almacenan dentro del mismo documento en forma de subdocumentos o arreglos.
   - Ventajas:
     - Acceso rápido a la información sin necesidad de realizar un *join*.
     - Adecuado cuando los datos relacionados se consultan siempre en conjunto.
   - Desventajas:
     - Genera documentos muy grandes y con redundancia si los datos se repiten.
     - Dificultad de mantenimiento cuando la relación crece demasiado.
2. **Enfoque referenciado:** Los documentos se almacenan en colecciones distintas y se enlazan mediante un identificador único (`ObjectId`).
   - Ventajas:
     - Más flexible y escalable en aplicaciones de gran tamaño.
     - Evita la duplicidad de datos y facilita su reutilización.
   - Desventajas:
     - Las consultas pueden requerir múltiples accesos (*join manual*).
     - Aumenta la complejidad de la lógica en la aplicación.

En comparación con PostgreSQL, este último utiliza claves foráneas que garantizan la integridad referencial de manera nativa, mientras que MongoDB prioriza la flexibilidad y el rendimiento, sacrificando la consistencia estricta de las relaciones.

Pregunta 02: Ventajas de NoSQL sobre bases de datos relacionales

Las bases de datos NoSQL, como MongoDB, presentan varias ventajas frente a las bases de datos relacionales tradicionales (RDBMS). Usando como ejemplo una liga deportiva, se destacan los siguientes puntos:

- **Flexibilidad de esquema:** Permite agregar nuevos atributos a jugadores o equipos sin necesidad de modificar una estructura rígida de tablas.
- **Escalabilidad horizontal:** MongoDB puede distribuir los datos entre múltiples servidores mediante *sharding*, mientras que en un RDBMS esto resulta más costoso.
- **Velocidad en operaciones:** Es ideal para registrar resultados en tiempo real de partidos, sin necesidad de transacciones complejas.
- **Representación natural de datos:** Los equipos pueden contener directamente un arreglo de jugadores dentro de su documento, sin requerir múltiples tablas intermedias.

En este contexto, NoSQL se adapta mejor a escenarios dinámicos y con alta variabilidad en los datos, como el registro de temporadas, jugadores y estadísticas deportivas.

Pregunta 03: Creación de base y colección MongoDB

```
test> use mi_bd
switched to db mi_bd
mi_bd> db.productos.insertMany([
...   { nombre: "Laptop", precio: 1200, stock: 10 },
...   { nombre: "Teléfono", precio: 450, stock: 25 }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68d8ac23cab8ac2cbbce5f47'),
    '1': ObjectId('68d8ac23cab8ac2cbbce5f48')
  }
}
```

Primero se ejecuta el comando `use mi_bd` para seleccionar la base de datos o en su lugar crearla en caso de que no exista y se crea la colección sobre la que se estarán haciendo inserciones de registros.

Pregunta 04: Consulta MongoDB

```
mi_bd> db.productos.find({ precio: { $gt: 500 } })
...
[
  {
    _id: ObjectId('68d8ac23cab8ac2cbbce5f47'),
    nombre: 'Laptop',
    precio: 1200,
    stock: 10
  }
]
```

Se realiza la consulta sobre la base de datos con el comando `find` sobre la colección creada. En este caso como nos interesa selecionar los valores mayores que 500 se utiliza `$gt` (greater than) para traer los registros que cumplan la condición y el campo sobre el que se desea realizar la consulta (precio).

Pregunta 05: Actualización y elimincación de registros MongoDB

Se usa el comando `updateOne` para hacer la actualización de un registros, en este caso deseamos incrementar en 100 el precio de un producto, en nuestro ejemplo sobre el teléfono así que ejecutames `$inc` para indicar el incremento, el aumento que deseamos (100), el campo sobre el cual hacer la actualización (precio) y el producto (teléfono). Posterior, validamos que se haya actualizado el registro con el comando `find`, se elimina un valor, en este caso Laptop y para ello se utiliza `deleteOne` y nuevamente con `find` validamos los cambios.

Referencias

1. Chodorow, K. (2013). *MongoDB: The Definitive Guide*. O'Reilly Media.
2. MongoDB Inc. (2023). *Data Modeling Concepts*. Recuperado de: https://www.mongodb.com/docs/manual/core/data-modeling-introduction/
3. MongoDB Inc. (2023). *CRUD Operations*. Recuperado de: https://www.mongodb.com/docs/manual/crud/
4. Stonebraker, M., & Hellerstein, J. M. (2015). *What Goes Around Comes Around*. Communications of the ACM, 58(7), 10–13.
5. Silberschatz, A., Korth, H., & Sudarshan, S. (2019). *Database System Concepts* (7th ed.). McGraw-Hill.