**Scripts para el Particionamiento y Resultados Obtenidos**

**1. Creación de la Base de Datos y Colección**

use TorneoFutbol2024;

// Crear una colección para partidos

db.createCollection("partidos");

// Insertar datos de ejemplo

db.partidos.insertMany([

{ \_id: 1, equipoLocal: "Equipo A", equipoVisitante: "Equipo B", fecha: new Date("2024-12-01"), resultado: { local: 2, visitante: 1 } },

{ \_id: 2, equipoLocal: "Equipo C", equipoVisitante: "Equipo D", fecha: new Date("2024-12-02"), resultado: { local: 1, visitante: 1 } },

{ \_id: 3, equipoLocal: "Equipo A", equipoVisitante: "Equipo C", fecha: new Date("2024-12-03"), resultado: { local: 3, visitante: 2 } },

]);

2. Creación de un Índice para Particionamiento

db.partidos.createIndex({ fecha: 1 });

**3. Particionamiento Horizontal (Sharding)**

**3.1. Habilitar Sharding en la Base de Datos**

sh.enableSharding("TorneoFutbol2024");

3.2. Configurar la Colección para el Sharding

sh.shardCollection("TorneoFutbol2024.partidos", { fecha: 1 });

**4. Resultados Obtenidos**

**4.1. Verificación del Estado de Sharding**

Para verificar el estado de sharding, se puede utilizar el siguiente comando:

sh.status();

**Resultados Esperados**:

* La colección partidos estará repartida entre los shards definidos.
* Los índices deben estar activos y optimizados para consultas rápidas sobre las fechas de los partidos.

5. Conclusiones

La implementación de particionamiento (sharding) permite una gestión eficiente de los datos en un torneo deportivo, mejorando el rendimiento y la disponibilidad del sistema.