

Conceptos y Comandos básicos del particionamiento en bases de datos NoSQL.

Actividad 3 – Unidad 3

Nombre de los estudiantes

Santiago Herrera Rocha

Docente

Jorge Isaac Castañeda Valbuena

CORPORACIÓN UNIVERSITARIA IBEROAMERICANA
INGENIERÍA DE SOFTWARE, FACULTAD DE INGENIERÍA
BASES DE DATOS AVANZADAS

Bogotá, D.C.

15 de diciembre de 2024

Requerimientos no funcionales y casos de prueba

1. Estrategia de Particionamiento

El particionamiento horizontal se realizará sobre la base de datos “torneo_futbol” creada en la Actividad 1. Las colecciones se fragmentarán por claves de partición específicas para optimizar el desempeño de las consultas y distribuir los datos equitativamente entre los nodos.

Claves (de Partición) para Crear por Colección

A continuación, se presentan las claves de partición que se van a crear por colección con el nombre de la clave y el motivo de creación de esta:

1. “equipos”:

- **Clave de partición:** “nombre”.
- **Motivo:** Cada equipo tiene un nombre único, ideal para distribuir los datos uniformemente.

2. “jugadores”:

- **Clave de partición:** “equipo”.
- **Motivo:** Las consultas suelen filtrar jugadores por equipo.

3. “arbitros”:

- **Clave de partición:** “nombre”.
- **Motivo:** El nombre es único y permite balancear los datos.

4. “Encuentros”:

- **Clave de partición:** “ronda”.
- **Motivo:** La mayoría de las consultas filtran por rondas específicas.

5. “tabla_posiciones”:

- **Clave de partición:** “equipo”.
- **Motivo:** Cada entrada corresponde a un equipo único.

2. Comandos para el Particionamiento Horizontal

Configuración del Entorno

1. Configurar el Config Server:

- Configurar el nodo que almacena la información de particionamiento.

```
mongod --configsvr --replSet cfgRS --dbpath configdb --port 27019
```

- Iniciar el replica set del config server:

```
mongo --port 27019
```

```
rs.initiate({id: "cfgRS", configsvr: true, members: [{ _id: 0, host:
"localhost:27019"}]});
```

2. Configurar los Shards:

- Cada shard es un nodo independiente. Ejemplo:

```
mongod --shardsvr --replSet shard1RS --dbpath shard1 --port 27018
```

```
mongod --shardsvr --replSet shard2RS --dbpath shard2 --port 27017
```

- Iniciar los sets de replicación de los shards:

```
mongo --port 27018
```

```
rs.initiate({ _id: "shard1RS", members: [{ _id: 0, host: "localhost:27018" }] });
```

```
mongo --port 27017
```

```
rs.initiate({ _id: "shard2RS", members: [{ _id: 0, host: "localhost:27017" }] });
```

3. Configurar el Router (o los mongos):

- Configurar el router que es el componente que coordina las operaciones entre shards.

```
mongos --configdb cfgRS/localhost:27019 --port 27020
```

4. Conectar al Router:

- Conectarse al router para administrar el particionamiento.

```
mongo --host localhost --port 27020
```

Configuración del Sharding

1. Añadir los Shards al Router:

```
sh.addShard("shard1RS/localhost:27018");
```

```
sh.addShard("shard2RS/localhost:27017");
```

2. Habilitar el Sharding para la Base de Datos:

```
sh.enableSharding("torneo_futbol");
```

3. Particionar las Colecciones:

- Para “equipos”:

```
sh.shardCollection("torneo_futbol.equipos", { nombre: 1 });
```

- Para “jugadores”:

```
sh.shardCollection("torneo_futbol.jugadores", { equipo: 1 });
```

- Para “arbitros”:

```
sh.shardCollection("torneo_futbol.arbitros", { nombre: 1 });
```

- Para “encuentros”:

```
sh.shardCollection("torneo_futbol.encuentros", { ronda: 1 });
```

- Para “tabla_posiciones”:

```
sh.shardCollection("torneo_futbol.tabla_posiciones", { equipo: 1 });
```

Resultados

Figura 1 – Ejecución del comando status en el router

```
[direct: mongos] test> sh.status();
shardingVersion
{ _id: 1, clusterId: ObjectId('6757841ebb3c9020d7922c77') }
---
shards
[
  {
    _id: 'shard1RS',
    host: 'shard1RS/localhost:27018',
    state: 1,
    topologyTime: Timestamp({ t: 1733789930, i: 10 }),
    replSetConfigVersion: Long('1')
  },
  {
    _id: 'shard2RS',
    host: 'shard2RS/localhost:27017',
    state: 1,
    topologyTime: Timestamp({ t: 1733789938, i: 9 }),
    replSetConfigVersion: Long('1')
  }
]
---
active mongoses
[ { '8.0.3': 1 } ]
---
autosplit
{ 'Currently enabled': 'yes' }
---
balancer
{
```

Figura 2 – segundo fragmento de la ejecución del comando status en el router

```
shardedDataDistribution
[
  {
    ns: 'torneo_futbol.encuentros',
    shards: [
      {
        shardName: 'shard2RS',
        numOrphanedDocs: 0,
        numOwnedDocuments: 0,
        ownedSizeBytes: 0,
        orphanedSizeBytes: 0
      }
    ]
  },
  {
    ns: 'torneo_futbol.tabla_posiciones',
    shards: [
      {
        shardName: 'shard2RS',
        numOrphanedDocs: 0,
        numOwnedDocuments: 0,
        ownedSizeBytes: 0,
        orphanedSizeBytes: 0
      }
    ]
  },
  {
    ns: 'config.system.sessions',
    shards: [
```

Figura 3 - tercer fragmento de la ejecución del comando status en el router

```
{
  ns: 'torneo_futbol.arbitros',
  shards: [
    {
      shardName: 'shard2RS',
      numOrphanedDocs: 0,
      numOwnedDocuments: 0,
      ownedSizeBytes: 0,
      orphanedSizeBytes: 0
    }
  ]
},
{
  ns: 'torneo_futbol.equipos',
  shards: [
    {
      shardName: 'shard2RS',
      numOrphanedDocs: 0,
      numOwnedDocuments: 0,
      ownedSizeBytes: 0,
      orphanedSizeBytes: 0
    }
  ]
},
{
  ns: 'torneo_futbol.jugadores',
  shards: [
    {
      shardName: 'shard2RS',
      numOrphanedDocs: 0,
```


Figura 4 - segundo fragmento de la ejecución del comando status en el router

```
{
  database: {
    _id: 'torneo_futbol',
    primary: 'shard2RS',
    version: {
      uuid: UUID('9a08505b-bbd2-4e9b-937d-787d4fb21e8b'),
      timestamp: Timestamp({ t: 1733789993, i: 5 }),
      lastMod: 1
    }
  },
  collections: {
    'torneo_futbol.arbitros': {
      shardKey: { nombre: 1 },
      unique: false,
      balancing: true,
      chunkMetadata: [ { shard: 'shard2RS', nChunks: 1 } ],
      chunks: [
        { min: { nombre: MinKey() }, max: { nombre: MaxKey() }, 'on shard': 'shard2RS', 'last modified': Timestamp({ t:
1, i: 0 }) }
      ],
      tags: []
    },
    'torneo_futbol.encuentros': {
      shardKey: { ronda: 1 },
      unique: false,
      balancing: true,
      chunkMetadata: [ { shard: 'shard2RS', nChunks: 1 } ],
      chunks: [
        { min: { ronda: MinKey() }, max: { ronda: MaxKey() }, 'on shard': 'shard2RS', 'last modified': Timestamp({ t:
1, i: 0 }) }
      ]
    }
  }
}
```