SGDI: Sistema de Gestión de Datos y de la Información
Sesión guiada 2

AUTORES: Aitor Cayón Ruano, Javier Cortés Tejada

9 de enero de 2019

Aitor Cayón Ruano y Javier Cortés Tejada declaramos que esta solución es fruto exclusivamente de nuestro trabajo personal. No hemos sido ayudados por ninguna otra persona ni hemos obtenido la solución de fuentes externas, y tampoco hemos compartido nuestra solución con nadie. Declaramos además que no hemos realizado de manera deshonesta ninguna otra actividad que pueda mejorar nuestros resultados ni perjudicar los resultados de los demás.

EVALUACIÓN DE CONSULTAS

Ejercicio 1

¿Cuál es el plan ganador para la misma consulta ordenada por nombre de usuario?

```
db.users.find({"year":1980}).sort({"username":1})
```

Solución

```
> exp.find({year:1980}).sort({username:1})
{
        "winningPlan" : {
                "stage" : "SORT",
                "sortPattern" : {
                        "username" : 1
                },
                "inputStage" : {
                        "stage" : "SORT_KEY_GENERATOR",
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "filter" : {
                                        "year" : {
                                                "$eq" : 1980
                                        }
                                },
                                "direction" : "forward"
                        }
                }
        }
}
```

En este caso tenemos 3 etapas dentro del **winningPlan**:

- **SORT**: indica el campo a la hora de ordenar los resultados.
- **SORT_KEY_GENERATOR**: ordena en memoría los resultados de las etapas hijo.
- **COLLSCAN**: escanea todos los elementos de la colección.

Ejercicio 2

Considerando el índice *year_1*, ¿cuáles son las etapas del plan ganador para la consulta de la pregunta anterior?

```
db.users.find({"year":1980}).sort({"username":1})
```

Solución

```
> exp.find({year:1980})
{

        "winningPlan" : {
                "stage" : "FETCH",
                "inputStage" : {
                        "stage" : "IXSCAN",
                        "keyPattern" : {
                                "year" : 1
                        },
                        "indexName" : "year_1",
                        "isMultiKey" : false,
                        "multiKeyPaths" : {
                                "year" : [ ]
                        },
                        "isUnique" : false,
                        "isSparse" : false,
                        "isPartial" : false,
                        "indexVersion" : 2,
                        "direction" : "forward",
                        "indexBounds" : {
                                "year" : [
                                        "[1980.0, 1980.0]"
                                ]
                        }
                }
        }
}
```

Es este caso, al igual que en el enunciado, el *winningPlan* tiene una etapa **IXNCAN** donde usa el índice *year_1* para haber la búsqueda seguida de una etapa **FETCH** para obtener los datos correspondientes.

Ejercicio 3

Considerando el mismo índice *year_1*, ¿cuáles son las etapas del plan ganador para la siguiente consulta?¿Cuántos documentos y claves se examinan? ¿Qué rango de claves se recorre?

```
db.users.find({"like":"deportes"}).sort({"year":1})
```

Solución

```
> exp.find({like:'deportes'}).sort({year:1})
{

        "winningPlan" : {
                "stage" : "FETCH",
                "filter" : {
                        "like" : {
                                "$eq" : "deportes"
                        }
                },
                "inputStage" : {
                        "stage" : "IXSCAN",
                        "keyPattern" : {
                                "year" : 1
                        },
                        "indexName" : "year_1",
                        ...
                        "indexBounds" : {
                                "year" : [
                                        "[MinKey, MaxKey]"
                                ]
                        }
                }
        }
        ...
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 0,
                "executionTimeMillis" : 0,
                "totalKeysExamined" : 0,
                "totalDocsExamined" : 0,
                ...
}
```

Al igual que en el apartado 2 tenemos etapas **IXSCAN** y **FETCH**. Además no se examinan ningún documento ni clave y el rango de claves oscila entre *Minkey* y *Maxkey*, es decir, el valor más bajo y más alto respectivamente que puede tomar *year*.

Ejercicio 4

Observa la evaluación de la consulta anterior tras crear el índice *year_1__id_1* y responde a las siguientes cuestiones:

1. **¿Cuál es el plan ganador? ¿Qué índice usa?**
2. **¿Cuántas claves y cuantos documentos consulta el plan ganador?**
3. **¿Cuáles son los planes rechazados?¿Qué índices usan?**

```
> exp.find({year:1980},{_id:1})
{

        "winningPlan" : {
                "stage" : "PROJECTION",
                "transformBy" : {
                        "_id" : 1
                },
                "inputStage" : {
                        "stage" : "IXSCAN",
                        "keyPattern" : {
                                "year" : 1,
                                "_id" : 1
                        },
                        "indexName" : "year_1__id_1",
                        ...
                }
                ...
                "rejectedPlans" : [
                        {
                                "stage" : "PROJECTION",
                                "transformBy" : {
                                        "_id" : 1
                                },
                                "inputStage" : {
                                        "stage" : "FETCH",
                                        "inputStage" : {
                                                "stage" : "IXSCAN",
                                                "keyPattern" : {
                                                        "year" : 1
                                                },
                                                "indexName" : "year_1",
                                                ...
                                }
                ]
        }


```

```
39          "executionStats" : {
40                  "executionSuccess" : true,
41                  "nReturned" : 0,
42                  "executionTimeMillis" : 0,
43                  "totalKeysExamined" : 0,
44                  "totalDocsExamined" : 0,
45                  ...
46          }
47  }
```

En este caso tenemos que el plan ganador cuenta con una etapa **IXSCAN** que usa el índice *year_1__id_1* seguida de una etapa **PROJECTION**, y en este caso al igual que en el anterior no se han examinado ni claves ni documentos. Por último tenemos que el plan rechazado cuenta con las mismas etapas del ejercicio 3 a lo que se añade una etapa **PROJECTION**, además usa el índice *year_1*.

REPLICA SETS

Ejercicio 5

¿Qué servidor se ha convertido en primario?

Tras lanzar el servidor con puerto 27102, según se indica en el enunciado, y haberlo inicializado se ha convertido en secundario. Tras haber añadido el servidor con puerto 27101 vemos que s2 se ha convertido en primario.

Ejercicio 6

¿Los servidores secundarios han sufrido algún cambio? Si es así, ¿a qué es debido ese cambio?

Tras haber matado el proceso 7288 del servidor s2 se ha convertido en primario el servidor s3. Esto es así porque teníamos levantados los tres servidores al mismo tiempo, de forma que se ha votado para elegir al primario, que ha resultado ser s3. En caso de haber tenido solo dos servidores activos s2 y s3 (como se indicaba en la práctica) si hubiésemos matado el proceso de s2 el servidor s3 se hubiese quedado como secundario pues para decidir el nuevo primario es necesario llevar a cabo una *votación* donde se requieren mínimo 2 votos para convertirse en primario, cosa que en este caso no es posible al haber solo un servidor activo.

Ejercicio 7

¿Qué servidores quedan activos y en qué estado están?

Tras hacer rs.status() en s1 y s3 ambos quedan marcados como secundario y primario respectivamente, mientras que s2 está caído. (Si solo tuviésemos 2 servidores como en el apartado anterior el restante sería secundario).