



TEMA-13-ACCESO-A-DATOS.pdf



user_4383848



Acceso a datos



1º Desarrollo de Aplicaciones Multiplataforma



Estudios España

antes



**Descarga sin publi
con 1 coin**



Después

WUOLAH



TODOS LOS LADOS DE LA CAMA

14 NOVIEMBRE
SOLO EN CINES

TEMA 13 - ACCESO A DATOS EXPLOTACIÓN DE BD NO-SQL

1. CONCEPTO Y CARACTERÍSTICAS PRINCIPALES DE LOS ÍNDICES:

Características de los índices:

- Ofrecen grandes mejoras en las operaciones de lectura.
- Pueden penalizar las operaciones de escritura.
- Se almacenan en **memoria**.
- Por defecto se crea un índice único sobre el campo `_id`.
- No sólo se indican los campos del índice sino el orden del mismo.
- Los operadores de negación no utilizan índices.
- Los índices los emplean las queries y las "agregaciones".

2. TIPOLOGÍAS DE LOS ÍNDICES EN MONGODB:

- **Monoclave:** Sólo indexan por un campo de los documentos de la colección
`db.collection.createIndex({ <field>: <-1|1> })`

Características:

- **Sólo afectan a un campo de búsqueda.** Indicar -1 o 1 muestra si el índice es ascendente o descendente.
- **Índices sobre entidades embebidas:** sólo se produce un acierto en el índice si las entidades son exactamente iguales, incluyendo orden.
- **Compuestos:** Indexan por varios campos de los documentos de la colección. El -1 indica ordenación descendente y la 1 ordenación ascendente.
- **Hashed:** Indexan de forma clave-valor. Son índices en los que un elemento del índice hace referencia un campo que tiene valores de array. Pueden ser índices de un único campo o compuestos. Se crean tantas entradas del índice como valores contenga el array. Solo uno de los elementos del índice puede tener valores de array.
- **Text:** Índices de texto.
- **Geoespaciales:** Indexan por coordenadas espaciales
- **Unique:** Sólo puede haber una correspondencia entre entrada del índice y documento. Indica que cada valor del índice debe corresponder con un único valor. Se puede utilizar tanto para índices de un solo campo o multicampo.
- **Sparse:** No indexa campos con valores nulos. Es un índice de valores únicos que admite valores nulos. Los valores nulos no se introducen en el índice. Una búsqueda que utilice este índice no devolverá documentos que no estén en el índice
- **Background:** Más lento, pero no bloquea a los lectores/escritores.
- **TTL:** Elimina documentos de la colección cuando pasa cierto tiempo o cuando alcanza la fecha de expiración.
- **Partial:** Indexa aquellos documentos que cumplan una condición.

WUOLAH

3. OPERACIONES CON LOS ÍNDICES:

- **Covered query:** Es una consulta que se resuelve contra un índice sin consultar datos de la colección.

Collection Query Criteria Projection

```
db.users.find( { score: { "$lt": 30 } }, { score: 1, _id: 0 } )
```

- **Borrado de índice:** El borrado de índice lo realizaríamos de la siguiente forma

```
db.accounts.dropIndex( { "tax-id": 1 } )
```

- **Regeneración índice:** Para volver a crear un índice ya creado previamente usaremos:

```
db.accounts.reIndex()
```

- **Listado de índices:** Para listar los índices asociados a una colección usaremos:

```
db.accounts.getIndexes()
```

4. CREACIÓN DE USUARIOS Y ROLES:

Mongo DB emplea el control de acceso basado en roles (RBAC), para determinar el acceso de los usuarios. A un usuario se le otorgan uno o más roles que determinan el acceso o los privilegios del usuario.

Un usuario debe tener solo el conjunto mínimo de privilegios necesarios para garantizar un sistema de privilegios mínimos. Cada aplicación y usuario de un sistema Mongo debe asignarse a un usuario distinto. Este aislamiento de acceso facilita la revocación del acceso y el mantenimiento continuo del usuario. Para crear un usuario en Mongo DB podemos ejecutar el comando "**db.createUser()**". Si el usuario es creado no devolverá nada, si el usuario existe previamente, nos devolverá un error de duplicado.

4.1. ACCESO DE CONTROL BASADO EN ROLES (RBAC):

MongoDB emplea el control de acceso basado en roles (RBAC) para controlar el acceso a un sistema MongoDB, además, a un usuario se le otorgan uno o más roles que determinarán el acceso del usuario a los recursos y operaciones de la base de datos, lo que quiere decir, que fuera de las asignaciones de funciones, el usuario no tiene acceso al sistema.

MongoDB no habilita el control de acceso basado en roles por defecto. **Se puede habilitar la autorización mediante la configuración "--auth" o a través de la configuración "security.authorization"**. Si se habilita la autenticación interna también se habilita la del cliente. Una vez que se habilita el control de acceso, los usuarios deben autenticarse.

Cada privilegio se especifica explícitamente en el rol o se hereda de otro rol. Un recurso puede ser una base de datos, una colección, un conjunto de ellas, o incluso el clúster.

Podemos usar el comando "rolesInfo" para visualizar los privilegios de un rol con el parámetro "showPrivileges" y "showBuiltinRoles" a true.

4.2. EJEMPLO CREACIÓN DE ROL:

- **Primero** nos conectaremos a nuestra base de datos MongoDB con los privilegios pertinentes.

```
mongo --port 27017 -u user -p '1234' --authenticationDatabase 'admin'
```

- **Segundo** crearemos un nuevo rol para administrar las operaciones actuales. Crearemos un rol llamado **"manRole"**:

```
use admin
db.createRole(
{
  role: "manRole",
  privileges: [
    { resource: { cluster: true }, actions: [ "killop",
"inprog" ] },
    { resource: { db: "", collection: "" }, actions: [
"killCursors" ] }
  ],
  roles: []
})
```

En la operación anterior hemos creado un rol que garantiza **permisos para hacer "kill" de cualquier operación**.

5. IMPORTACIÓN DE DATOS:

Proceso de importación de datos a través de Compass.

- El primer paso sería **conectarnos a nuestra base de datos mongoDB** como hemos estudiado previamente.
- Después haríamos **clic en el botón "Add Data"** y seleccionaríamos la opción **"Import File"**.
- La aplicación nos mostrará un cuadro de diálogo donde deberemos de **indicar la ruta del fichero que queremos introducir a nuestra base de datos**.
- **Seleccionaremos el tipo de fichero que vamos a introducir: JSON o CSV**. Si importamos un fichero CSV deberemos especificar los campos que vamos a importar y los tipos de los mismos. El tipo de datos por defecto es String.
- Tendremos que **configurar las opciones de importación acorde a nuestro caso**.
- Una **barra de progreso** mostrará el estado actual de la importación. Una vez finalizado el proceso, la aplicación mostrará los datos importados.



6. EXPORTACIÓN DE DATOS:

Realizaremos una exportación de datos con la aplicación **Compass**.

- El primer paso sería **conectarnos a nuestra base de datos MongoDB** y navegar hasta encontrar la información, colección/es que deseemos exportar.

- Haremos clic en el menú **"Collection"** de nuestra aplicación y a continuación en la opción **"Export Collection"** y Compass nos mostrará un cuadro de dialogo.

El cuadro muestra la **"query"** por medio de la que se va a realizar la operación. Si queremos ignorar la **"query"** y exportar directamente la colección completa podemos seleccionar el radio button **"Export Full Collection"** y hacer clic en **"Select Fields"**.

- En esta parte se nos muestra otro cuadro de diálogo donde **seleccionaremos los campos a exportar**. Si nuestra aplicación no nos detecta algún campo podremos añadirlo manualmente con el botón de **"Add Field"**. Si todo está correcto hacemos clic en **"Select Output"**.

- Básicamente en esta parte deberemos de **seleccionar el formato del fichero** que queremos exportar: JSON o CSV son las opciones disponibles.

- Hacemos clic en **Export**.