Contenido

Introducción

COMO perfil analista de datos, **QUIERO** realizar **consultas** (queries) de forma **sencilla** (con habilidades comunes en el mercado), **eficiente** (código sencillo y rápido) e **interactiva** (analizar, ver resultados, volver a analizar... con tiempos de respuesta razonables), **PARA** analizar los datos, conocerlos y extraer patrones.

Como **perfil analista de datos** para explotar (analizar, conocer, extraer patrones, etc.) los datos almacenados en bases de datos NO relacionales utilizando operaciones CRUD (Create, Read, Update, Delete) tendremos **acceso** a dichos datos de alguna (o varias) de las siguientes formas:

- **entorno interactivo** que nos permitirá ejecutar operaciones CRUD directamente y mostrar el resultado,
- acceso programático via **API** que nos permitirá ejecutar código (pej. Python) en el que pej. podremos incrustar nuestros comandos,
- otras formas en las que accedemos a los datos con ciertas limitaciones en cuanto a **visibilidad** (pej. no podemos ver todos los datos), **recursos** (pej. solo podemos recuperar cierta cantidad de datos), **permisos** (pej. no solo podemos leer ciertas tablas, o no podemos hacer ciertas operaciones)

En cualquier caso lo habitual (y deseable por seguridad y por otros muchos motivos) es que NO actuemos como administradores de las bases de datos (perfil profesional DBA = DataBase Administrator) ni tengamos sus conocimientos.

Como **perfil analista de datos** debemos conocer las principales operaciones CRUD para explotar los datos pero no es nuestro objetivo instalar, configurar y operar una BD NoSQL. Ciertamente esos conocimientos de **DBA** son útiles para poder montar nuestro entorno de pruebas y es relativamente sencillo instarlar en tu PC una BD NoSQL como MongoDB para “jugar”

Por tanto vamos a abordar los fundamentos de las operaciones CRUD en BD NoSQL desde el punto de vista de un perfil analista de datos y desde un punto de vista muy práctico.

A este nivel introductorio no se persigue conocer en detalle un lenguaje de alguna BD NoSQL para realizar operaciones CRUD si no que conozcas los fundamentos y las herramientas que te permitan realizar una explotación de datos y poder seguir avanzando de forma autónoma.

Normalizacion y modelado de datos en BD SQL y NoSQL

Recordemos el proceso de modelado de datos en bases de datos relacionales (BD SQL, RDBMS):

- **Modelo Conceptual:** También llamado modelo Entidad-Relación. Persigue modelar (representar) gráficamente los objetos y conceptos del mundo real, así como las relaciones entre ellos. Buscamos la estructura, la arquitectura, la forma que tiene la información. Es la representación de los datos en un lenguaje informal. Deduciremos el modelo conceptual de una base de datos a partir de la recogida de requisitos utilizando una descripción textual, una entrevista, examen de diagramas de flujo de datos, etc.

- **Modelo Lógico:** Definimos cómo queremos los datos. Se representan de una forma estructurada y estableciendo relaciones entre ellos. Es en esta etapa cuando se aplica la **normalización** según el proceso que hemos visto (formas normales).

- **Modelo Físico:** Es la representación de los datos y su estructuración en un lenguaje que pueda ser "entendido" por una máquina. Pasar de los esquemas del modelo lógico a tipos (enteros, cadenas…) y estructuras de datos (tablas…) es usar el modelo físico.

Cuando hablamos de realizar un "diseño" (pej. "diseño conceptual") nos referimos a "modelar" de forma que utilizando cierto "modelo" (pej. modelo entidad-relación) para desarrollar dicho diseño y obtener como resultado/entregable (salida) un "esquema" o un "diagrama" (en papel, en PDF, en una herramienta de diseño…). Este entregable (salida) nos permite transmitir esa información a otras personas y utilizarla como entrada de otros procesos (pej. en la definición de una arquitectura de datos).

En las BD NoSQL, tomando como ejemplo referencia la conocida BD NoSQL de tipo documental MongoDB, el concepto de **normalización** de las BD SQL se relaja y se permite la redundancia de datos = "incrustar" los datos en las estrucuturas.

Para comprender los fundamentos del modelado de datos y sobre cómo estructurar los documentos en MongoDB puedes revisar el artículo https://www.mongodb.com/docs/manual/core/data-model-design/ : *The key consideration for the structure of your documents is the decision to embed or to use references.*

Lenguaje MQL

MQL (MongoDB Query Language) es un lenguaje específico de consulta y manipulación de datos para MongoDB. Aunque MQL está basado en la sintaxis de JSON, no se basa en Python ni en ningún otro lenguaje de programación de alto nivel.

Es importante tener en cuenta que MQL es un lenguaje de consulta y manipulación de datos específico de MongoDB, y su objetivo es proporcionar una forma eficiente de interactuar con la base de datos MongoDB. MQL proporciona una serie de operaciones y comandos específicos de MongoDB para realizar tareas de consulta y manipulación de datos en la base de datos.

Aunque se puede interactuar con MongoDB desde otros lenguajes de programación de alto nivel como Python, Java o Node.js, esto se realiza utilizando drivers o bibliotecas específicas que proporcionan una interfaz de programación para interactuar con MongoDB. En este caso, se utilizan los lenguajes de programación de alto nivel para escribir aplicaciones que interactúan con MongoDB a través de estos drivers o bibliotecas, pero no se utiliza MQL directamente.

IMPORTANTE: A este nivel introductorio NO se persigue conocer en detalle el lenguaje MQL si no que conozcas los fundamentos y las herramientas que te permitan realizar una explotación de datos y poder seguir avanzando de forma autónoma en el conocimiento del lenguaje MQL.

Interoperabilidad no garantizada:

La interoperabilidad entre los lenguajes de consulta y manipulación de datos de las bases de datos NoSQL no está garantizada, debido a que cada sistema de base de datos NoSQL puede tener su propia sintaxis y gramática para estas operaciones.

Por tanto el código que desarrolles probablemente NO será transportable de una BD NoSQL a otra. Ten esto en cuenta cuando utilices diferentes motores de BD ya sea en real o en las herramientas que veremos (simuladores para ejecutar MQL).

Existen herramientas y bibliotecas que pueden ayudar a facilitar la interoperabilidad entre sistemas de bases de datos NoSQL y otros sistemas o herramientas de software. Por ejemplo, existen bibliotecas de software que permiten a los desarrolladores interactuar con múltiples sistemas de bases de datos NoSQL utilizando una única API. También existen herramientas de integración de datos que pueden ayudar a transferir datos entre sistemas de bases de datos NoSQL y sistemas de bases de datos relacionales, o entre sistemas de bases de datos NoSQL y otras aplicaciones.

Algunas bases de datos NoSQL, como MongoDB, ofrecen la posibilidad de conectarse a través de **drivers** de diferentes lenguajes de programación, lo que facilita la interoperabilidad con aplicaciones desarrolladas en diferentes lenguajes. Por ejemplo, MongoDB cuenta con drivers para Java, Python, C#, Node.js, entre otros.

Referencias sobre MQL

IMPORTANTE: estas referencias NO son pre-requisitos para nuestros objetivos si no fuentes de información a modo de complemento por si quieres ampliar tu conocimiento sobre lenguaje MQL.

A parte de las que estamos viendo existe multitud de fuentes, te indico algunas para que las revises de forma crítica y valores cual te es más útil según tus objetivos. Existen diferentes objetivos, algunos de ellos:

- aprender/estudiar un manual/tutorial/libro de forma estructurada sobre MQL,
- practicar MQL usando un tutorial a modo de guía,
- consultar un manual o la documentación del fabricante a modo de referencia sobre uso MQL.

Fuentes:

- https://www.mongodb.com/docs/manual/ : **documentación oficial de MongoDB, clicando sobre el icono de buscar (una lupa) se puede buscar la referencia de un comando MQL.**

- https://www.mongodb.com/docs/manual/crud/#std-label-crud : referencia en concreto para operaciones CRUD

- https://www.mongodb.com/docs/manual/tutorial/getting-started/ : **tutorial** para practicar la inserción de datos de prueba en una base de datos MongoDB y la consulta de esos datos mediante el **shell web integrado (embedded web shell).** No se necesita instalar MongoDB para completar este tutorial. Esta shell permite pegar los comandos (pej. en Windows con CTRL+V) pero NO permite copiar el texto para extraer la salida.

- https://nosqlzoo.net/ : análogo a https://sqlzoo.net/ , **simulador** muy útil pensado para practicar comandos sencillos sobre un dataset. Trabajas con datos reales de forma interactiva y puedes ver el resultado de tu sentencia. No te muestra la solución pero sí los datos correctos que debería retornar.

- https://www.w3schools.com/mongodb/mongodb_exercises.php : **ejercicios** donde NO trabajas con datos, se trata de escribir/completar la sentencia correcta en MQL, y puedes ver la solución.

Utilidades para ejecutar lenguaje MQL

IMPORTANTE: el objetivo es poner en práctica lo que estás aprendiendo sobre MQL desde un punto de vista de perfil analista de datos ejecutando operaciones CRUD sobre un conjunto de datos (dataset) almacenado en una BD NoSQL.

Como perfil analista de datos a priori no tenemos conocimientos de DBA y no queremos dedicar tiempo a instalar una BD NoSQL en tu PC o utilizar una en el cloud. Para ambos casos (instalación local o servicio en el cloud) hay opciones gratuitas y es fácil encontrar información para empezar a usarlas. Es una opción válida pero requiere dedicar cierto tiempo a montar un entorno de pruebas. Como alternativa más rápida y práctica existen numerosas herramientas para ejecutar MQL online simulando que estamos trabajando sobre una BD NoSQL como MongoDB.

Cada herramienta tiene sus pequeñas diferencias y particularidades.

Revisa estas herramientas para ejecutar código NoSQL:

- https://www.mongodb.com/docs/manual/tutorial/getting-started/ : es el **shell web integrado (embedded web shell)**, un simulador integrado en el tutorial. Recuerda esta shell permite pegar los comandos (pej. en Windows con CTRL+V) pero NO permite copiar el texto para por ejemplo extraer la salida.

- https://www.mycompiler.io/online-mongodb-editor
- https://www.mycompiler.io/new/mongodb

- https://onecompiler.com/mongodb

- https://mongoplayground.net/ (only find(), aggregate(), update() and explain() are supported)

Familiarizate con algunas de ellas y prueba a ejecutar alguno de los comandos básicos de MQL aprendidos. Ten en cuenta estas consideraciones y funcionalidades interesantes:

- algunas permiten ejecutar código MQL de manera interactiva y realista como si estuvieras trabajando en una BD real
- no simulan completamente una BD NoSQL MongoDB pero la mayoría sirven para crear tu dataset y ejecutar sentencias MQL sobre tus datos
- algunas herramientas permiten crear tu cuenta de usuario
- normalmente si se puede crear una cuenta de usuario se dispondrá de funcionalidades adicionales como poder guardar tus scripts MQL para ejecutarlos en otro momento
- es interesante poder exportar tu código MQL, se crea un fichero de texto, que puedes ejecutar en otra BD MongoDB

- algunos tienen salida de los comandos MQL con formato fácilmente legible (estilo JSON formateado), y otros muestran un estilo menos legible.
- algunos se comportan como un editor con ciertas funciones tipo programador (desarrollo de código) como pej. auto-completar comandos (con sentencias MQL o nombres de campos que hayas usado antes en el código).

A partir de las utilidades vistas te propongo realizar el siguiente ejercicio práctico:

Ejercicio práctico propuesto: operaciones CRUD con MQL

A partir de las referencias y utilidades vistas en los apartados anteriores te propongo un ejercicio practico mediante las siguientes sentencias.

Es un ejercicio OPCIONAL, NO es evaluable, NO tiene peso en la evaluación. Debes realizarlo de forma autónoma y para ello se incluyen ayudas y explicaciones adicionales:

1. **crea explicitamente la colección "libros":** si una colección no existe MongoDB la crea cuando insertas datos por primera vez en dicha colección, también podemos crear una colección de forma explícita con db.createCollection(). Como ayuda revisa el artículo https://www.mongodb.com/docs/manual/core/databases-and-collections/
    - Busca en la documentación oficial información sobre este comando, indica la página de referencia y algún ejemplo de dicha página.
    - indica el comando para crear explicitamente la colección "libros" y la salida

2. inserta un documento en la colección "libros" con dos campos "titulo" y "autor", cada uno con su respectivo valor "NoSQL databases" y "Alice" respectivamente.

Ayuda: la sintaxis del comando es:

- db: hace referencia a la base de datos actual en la que se está trabajando.
- libros: es el nombre de la colección en la que se va a insertar el documento.
- insertOne(): es el método utilizado para insertar un documento en la colección.
- El objeto {titulo: "NoSQL databases", autor: "Alice"} es el documento que se va a insertar en la colección "libros".

El resto de operaciones se harán sobre la colección creada (libros).

El **ObjectId** es un tipo de dato que se utiliza en MongoDB para identificar de manera única los documentos en una colección. La salida de los comandos devolverán un objeto que contiene un campo _id (o insertedId) con un valor ObjectId generado automáticamente por MongoDB. El ObjectId se utiliza como clave primaria en MongoDB y consta de 12 bytes, que se dividen en los siguientes componentes:

- Timestamp: los primeros 4 bytes representan un timestamp de la fecha y hora en que se creó el ObjectId.
- Machine ID: los siguientes 3 bytes representan un identificador único de la máquina que creó el ObjectId.
- Process ID: los siguientes 2 bytes representan un identificador único del proceso que creó el ObjectId.
- Counter: los últimos 3 bytes son un contador que se incrementa cada vez que se crea un nuevo ObjectId en el mismo proceso y en el mismo milisegundo.

Por tanto ObjectId es un valor único generado automáticamente por MongoDB que se utiliza para identificar de manera única los documentos en una colección.

3. inserta dos documentos con los mismos campos y con los valores titulo: "RDBMS by design", autor: "Tom" y titulo: "Estructuras de datos", autor: "Bob"

4. busca y muestra los documentos de la colección. ¿A que sentencia SQL equivale?

Operadores de comparación en MongoDB:

```
MongoDB | BD SQL
----------------
$eq     | ==
$gt     | >
$gte    | >=
$lt     | <
$lte    | <=
$ne     | !=, <>
```

En el resto de comandos añade el comando **print** para escribir texto descriptivo en pantalla, pej:

print("---------- CRUD - READ - FIND ----------");

5. busca y muestra los documentos de la colección cuyo autor es “Bob”. ¿A que sentencia SQL equivale?

6. actualiza el autor del libro “RDBMS by design” con el valor “Alice”

7. borra el documento con titulo "NoSQL databases"

8. busca y muestra los documentos de la colección