



UNIDAD 7

- **Bases de datos no relacionales.**
 - **Conceptos básicos y tipos.**
 - **Modelo de datos.**
 - **JSON.**
 - **Algunos motores y sus características.**
 - **MongoDB**

MongoDB Documentation - <https://www.mongodb.com/docs/>

JSON - <https://www.json.org/json-es.html>



Bases de datos no relacionales.

Introducción

Bases de Datos Relacionales

Las bases de datos relacionales son aquellas bases de datos que cumplen con el modelo relacional, que se fundamenta en el uso de relaciones. Una relación no es más que un vínculo entre dos entidades de la base de datos. La información contenida en las bases de datos relacionales y las relaciones de dependencia que se establecen entre sí, interpretados, persiguen describir el mundo real.

Características del modelo de datos relacional:

El modelo de datos relacional es un modelo matemático de datos. El componente principal del modelo relacional son las relaciones. Una relación es una interconexión entre los datos, que a su vez se almacenan en tablas. Así, en sentido inverso, se dice que dos tablas se relacionan entre sí por la relación que existe entre sus datos.



Bases de datos no relacionales.

Introducción

Bases de Datos Relacionales

Los componentes básicos de una base de datos relacional son:

- **Tablas:** Una base de datos relacional puede contener una o varias tablas donde se estructurarán los datos. Las tablas no pueden repetirse dentro de la misma base de datos.
- **Columnas:** Subdivisiones lógicas que conforman una tabla.
- **Registros:** Una tabla está formada por el conjunto de registros (también conocidos como filas o tuplas), es decir un conjunto de valores definidos para las columnas definidas en una tabla concreta.



Bases de datos no relacionales.

Introducción

Bases de Datos Relacionales

Los componentes básicos de una base de datos relacional son:

- **Relaciones entre tablas:** Las relaciones entre tablas se establecen mediante claves primarias y claves ajenas.
- **Clave primaria:** Cada registro posee una única clave primaria, y son claves que identifican unívocamente a un registro, (es decir son irrepetibles en toda la tabla).
- **Clave foránea:** Las claves foráneas son referencias colocadas en las tablas hijas de las relaciones entre tablas. Contienen el mismo valor que la clave primaria del registro padre.



Bases de datos no relacionales.

Introducción

Bases de Datos Relacionales

Finalmente, se puede decir que una base de datos relacional es un conjunto de una o más tablas, estructuradas en registros o tuplas y que se relacionan entre ellas mediante diversas claves.

Además, los principios del álgebra relacional y la teoría de conjuntos nos permitirán tratar la información que se quiera almacenar.



Bases de datos no relacionales.

Introducción

Bases de Datos No Relacionales o NoSQL

NoSQL es un término general para cualquier sistema alternativo a las bases de datos SQL tradicionales. A veces, cuando decimos sistemas de gestión NoSQL, nos referimos a cualquier base de datos que no utilice un modelo relacional. Las bases de datos NoSQL usan un modelo de datos que tiene una estructura diferente a la estructura de la tabla de filas y columnas que se usa con RDBMS.

Las bases de datos NoSQL son diferentes entre sí. Hay cuatro tipos de esta base de datos: bases de datos de documentos, almacenes de clave-valor, bases de datos orientadas a columnas y bases de datos de gráficos.



Bases de datos no relacionales.

Introducción

Bases de Datos No Relacionales o NoSQL

Tipos de sistemas de gestión de bases de datos NoSQL

Los desarrolladores necesitan soluciones que se alineen con las realidades de los datos modernos y las prácticas iterativas de desarrollo de software.

Las bases de datos NoSQL han surgido en los últimos años como respuesta a las limitaciones de las bases de datos relacionales tradicionales y para proporcionar el rendimiento, la escalabilidad y la flexibilidad que requieren las aplicaciones modernas.



Bases de datos no relacionales.

Introducción

Bases de Datos No Relacionales o NoSQL

La mayoría de los aspectos de estas tecnologías NoSQL varían mucho y tienen poco en común, excepto por el hecho de que no utilizan un modelo de datos relacional. **Hay cuatro tipos** de sistemas de gestión de bases de datos NoSQL :

- Almacenes de clave-valor

Este es el tipo más simple de base de datos NoSQL . Cada elemento se almacena como un par clave-valor que consta de un nombre de atributo ("clave") y un valor. Esta base de datos es como un RDBMS con dos columnas: el nombre del atributo (como "estado") y el valor (como "Alaska"). Los casos de uso de las bases de datos NoSQL incluyen carritos de compras, preferencias de usuario y perfiles de usuario.



Bases de datos no relacionales.

Introducción

Bases de Datos No Relacionales o NoSQL

- **Orientadas a columnas**

Mientras que un RDBMS almacena datos en filas y los lee fila por fila, las bases de datos orientadas a columnas se organizan como un conjunto de columnas. Cuando desee ejecutar análisis en una pequeña cantidad de columnas en la red, puede leer esas columnas directamente sin consumir memoria con datos no deseados. Las columnas son del mismo tipo y se benefician de una compresión más eficiente, lo que hace que las lecturas sean aún más rápidas. Una base de datos orientada a columnas puede agregar el valor de una columna dada (sumando las ventas del año, por ejemplo). Los casos de uso de una base de datos orientada a columnas incluyen análisis.



Bases de datos no relacionales.

Introducción

Bases de Datos No Relacionales o NoSQL

- **Orientadas a columnas**

Si bien las bases de datos orientadas a columnas son excelentes para el análisis, la forma en que escriben los datos dificulta que sean consistentes, ya que las escrituras de todas las columnas en la base de datos orientada a columnas requieren múltiples eventos de escritura en el disco. Las bases de datos relacionales no sufren este problema ya que los datos de las filas se escriben de forma contigua en el disco.



Bases de datos no relacionales.

Introducción

Bases de Datos No Relacionales o NoSQL

- **Bases de datos de grafos**

Una base de datos de grafos se centra en la relación entre los elementos de datos. Cada elemento está contenido como un nodo. Las conexiones entre los elementos de la base de datos se denominan vínculos o relaciones. Las conexiones son elementos de primera clase de la base de datos, almacenados directamente.



Bases de datos no relacionales.

Introducción

Bases de Datos No Relacionales o NoSQL

- **Bases de datos de grafos**

Una base de datos de grafos está optimizada para capturar y buscar las conexiones entre elementos, superando la sobrecarga asociada con UNIR varias tablas en SQL. Muy pocos sistemas comerciales del mundo real pueden sobrevivir únicamente en bases de datos de grafos. Como resultado, las bases de datos de gráficos generalmente se ejecutan junto con bases de datos más tradicionales.

Los casos de uso incluyen detección de fraude y redes sociales.



Bases de datos no relacionales.

Introducción

Bases de Datos No Relacionales o NoSQL

- **Bases de datos de documentos**

Una base de datos de documentos almacena datos en documentos JSON, BSON o XML. Los documentos en la base de datos se pueden anidar. Los elementos particulares se pueden indexar para consultas más rápidas.

Puede acceder, almacenar y recuperar documentos de su red en una forma mucho más parecida a los objetos de datos utilizados en las aplicaciones, lo que significa que se requiere menos traducción para usar y acceder a los datos en una aplicación. Los datos SQL a menudo se deben ensamblar y desensamblar cuando se mueven entre aplicaciones, almacenamiento o más de una red.



Bases de datos no relacionales.

Introducción

Bases de Datos No Relacionales o NoSQL

- **Bases de datos de documentos**

Son populares entre los desarrolladores porque ofrecen la flexibilidad de modificar sus estructuras de documentos según sea necesario para adaptarse a sus aplicaciones. La flexibilidad de esta base de datos acelera el desarrollo: los datos se convierten en código y están bajo el control de los desarrolladores. Pueden acceder y utilizarlo más fácilmente. En las bases de datos SQL, es posible que se requiera la intervención de los administradores de la base de datos para cambiar la estructura de una base de datos.



Bases de datos no relacionales.

Introducción

Bases de Datos No Relacionales o NoSQL

- **Bases de datos de documentos**

Las bases de datos de documentos generalmente se implementan con una arquitectura escalable, lo que proporciona un camino hacia la escalabilidad de los volúmenes y el tráfico de datos. Los casos de uso incluyen plataformas de comercio electrónico, plataformas comerciales y desarrollo de aplicaciones móviles.



Bases de datos no relacionales.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript)

Es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo.

Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999.

JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros.

Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.



Bases de datos no relacionales.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript)

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.



Bases de datos no relacionales.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript)

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.



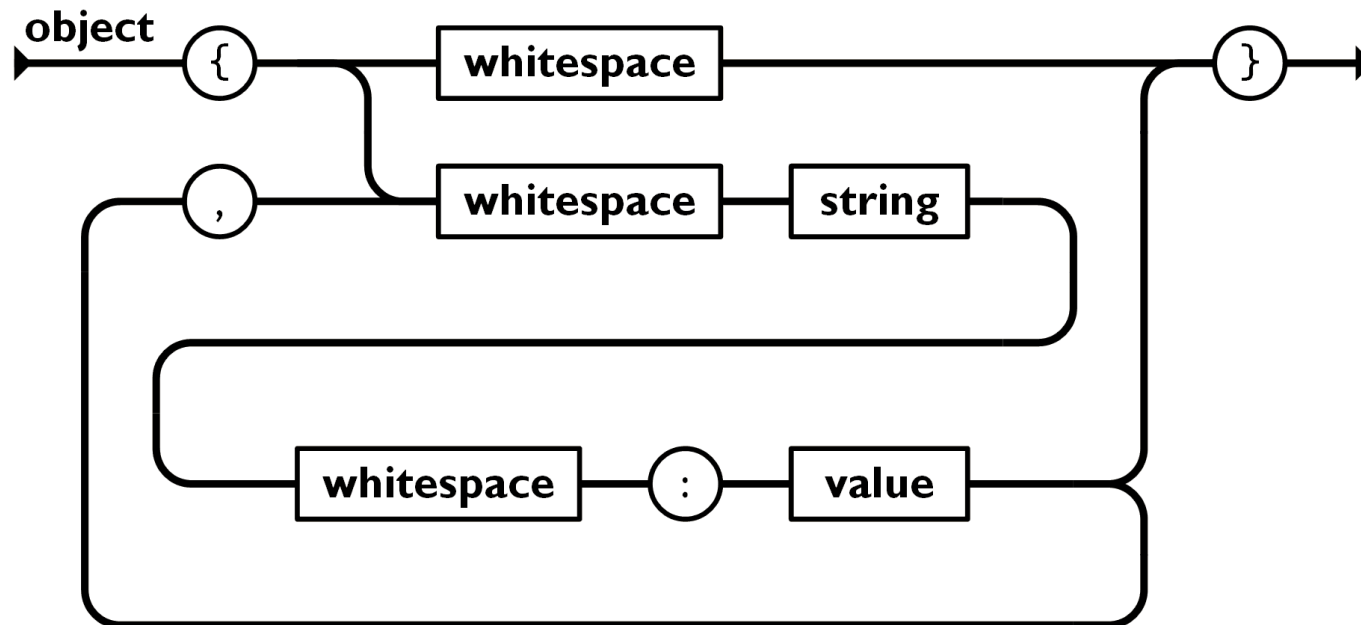
Bases de datos no relacionales.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript)

En JSON, se presentan de estas formas:

Un **objeto** es un conjunto desordenado de pares nombre/valor.

Un objeto comienza con {llave de apertura y termine con }llave de cierre. Cada nombre es seguido por :dos puntos y los pares nombre/valor están separados por ,coma.



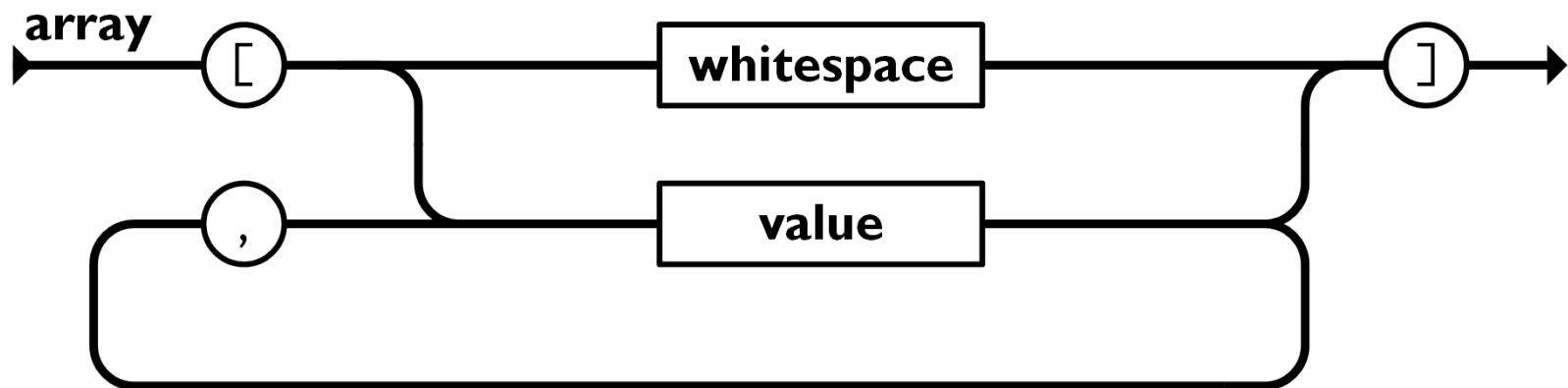


Bases de datos no relacionales.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript)

En JSON, se presentan de estas formas:

Un **arreglo** es una colección de valores. Un arreglo comienza con [corchete izquierdo y termina con]corchete derecho. Los valores se separan por ,coma.



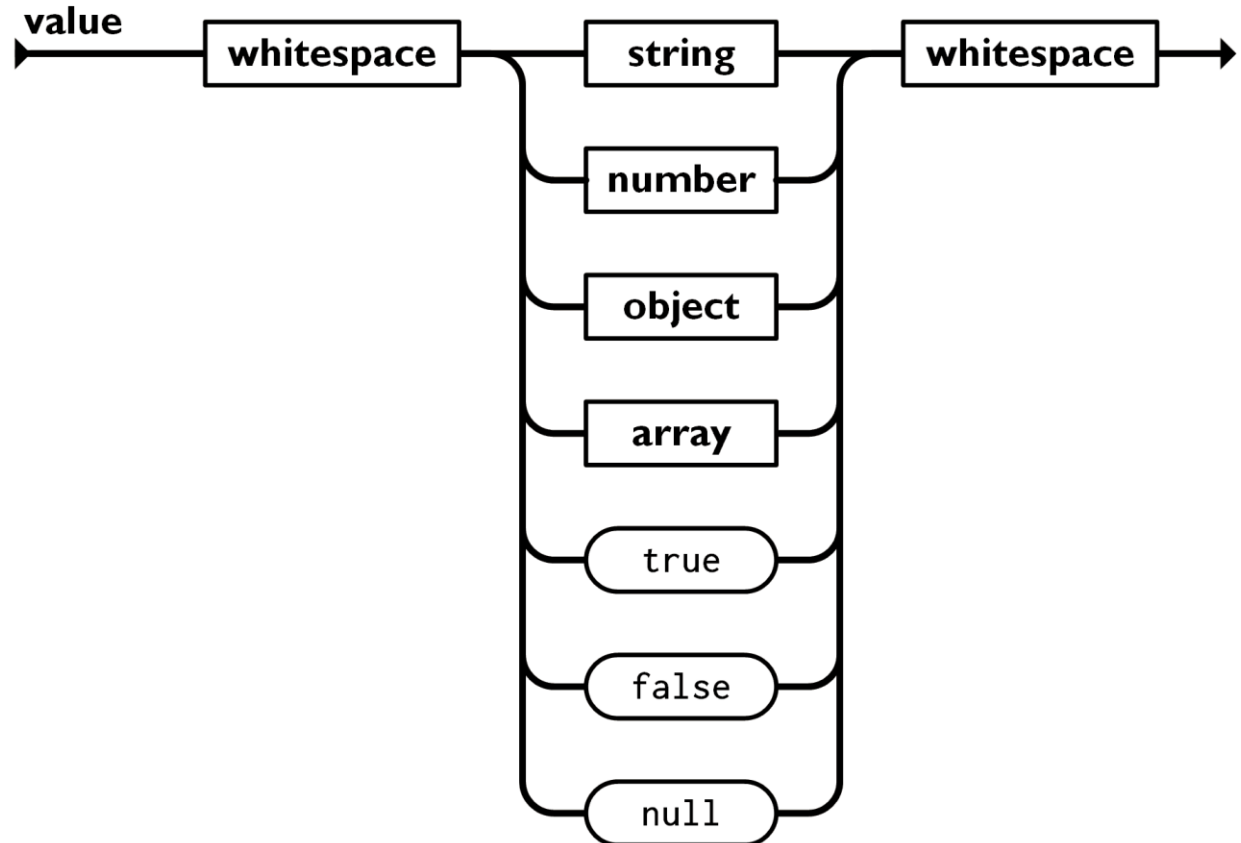


Bases de datos no relacionales.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript)

En JSON, se presentan de estas formas:

Un **valor** puede ser una
cadena de caracteres
con comillas dobles,
o un número,
o true
o false
o null,
o un objeto
o un arreglo.
Estas estructuras
pueden anidarse.





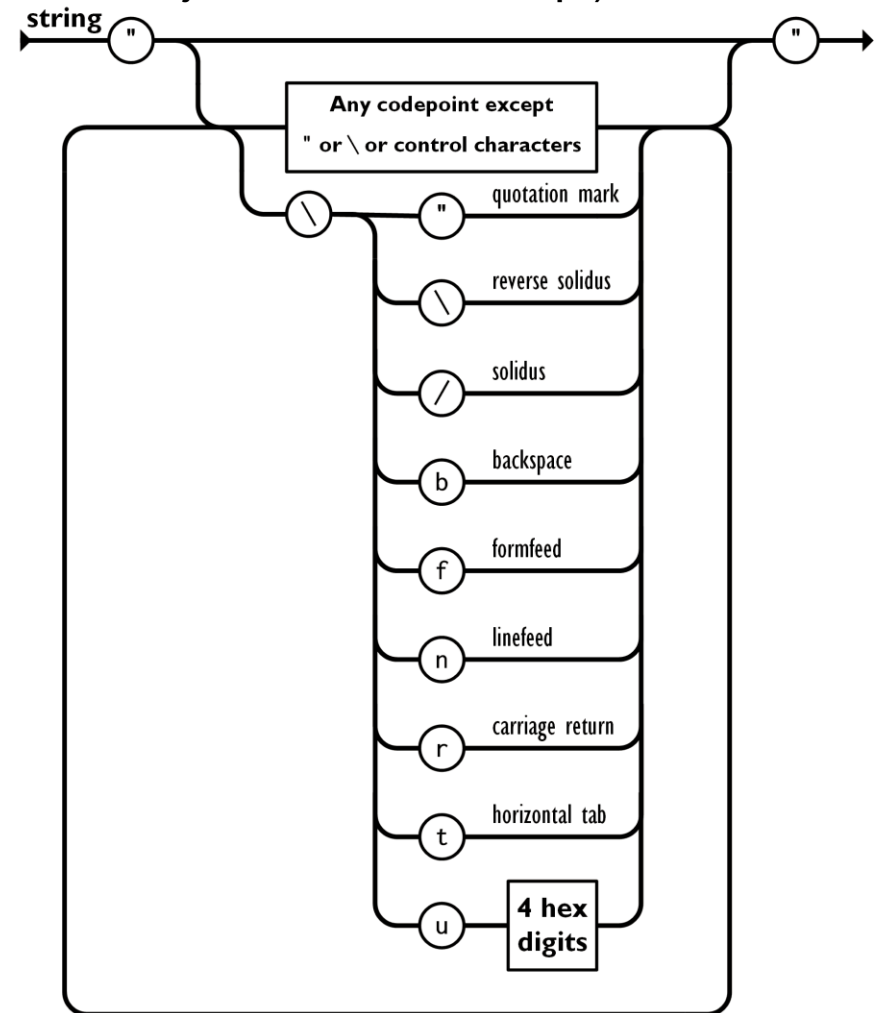
Bases de datos no relacionales.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript)

En JSON, se presentan de estas formas:

Una **cadena de caracteres**

es una colección de cero o más caracteres Unicode, encerrados entre comillas dobles, usando barras divisorias invertidas como escape. Un carácter está representado por una cadena de caracteres de un único carácter. Una cadena de caracteres es parecida a una cadena de caracteres C o Java.



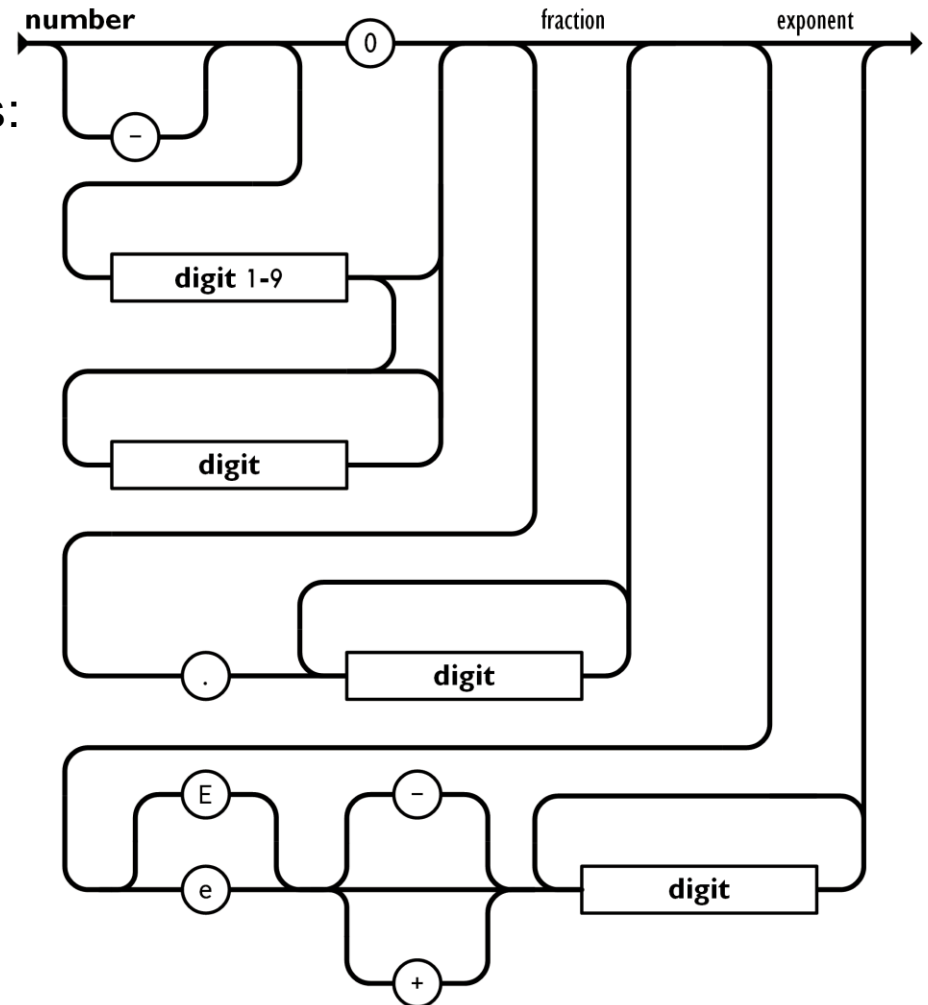


Bases de datos no relacionales.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript)

En JSON, se presentan de estas formas:

Un **número** es similar a un número C o Java, excepto que no se usan los formatos octales y hexadecimales.





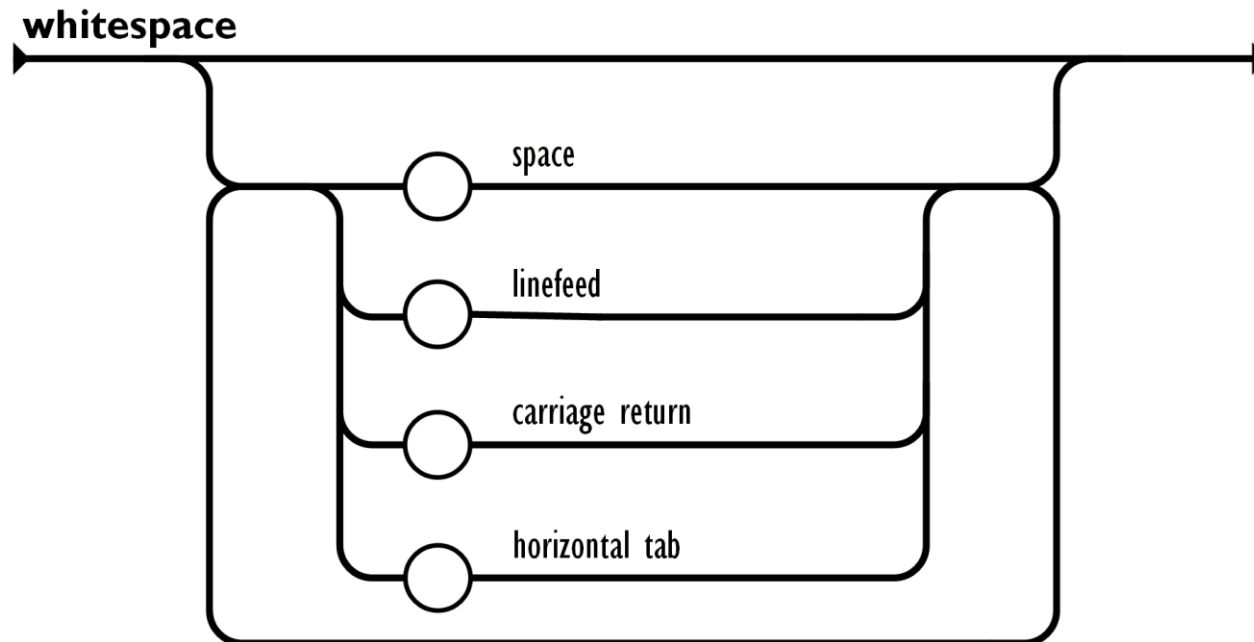
Bases de datos no relacionales.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript)

En JSON, se presentan de estas formas:

Los **espacios en blanco** pueden insertarse entre cualquier par de símbolos.

Exceptuando pequeños detalles de encoding, esto describe completamente el lenguaje.





Bases de datos no relacionales.

JSON y BSON

JSON y BSON son primos cercanos, como lo implican sus nombres casi idénticos, pero no lo sabrías mirándolos uno al lado del otro. JSON, o notación de objetos de JavaScript, es el estándar muy popular para el intercambio de datos en la web, en el que **se basa BSON (Binary JSON)**. Echaremos un vistazo a cada uno y, arrojaremos algo de luz sobre el misterio de JSON frente a BSON:

¿cuál es la diferencia y por qué es importante?



Bases de datos no relacionales.

JSON y BSON

JSON y BSON son primos cercanos, como lo implican sus nombres casi idénticos, pero no lo sabrías mirándolos uno al lado del otro. JSON, o notación de objetos de JavaScript, es el estándar muy popular para el intercambio de datos en la web, en el que **se basa BSON (Binary JSON)**. Echaremos un vistazo a cada uno y, arrojaremos algo de luz sobre el misterio de JSON frente a BSON:

¿cuál es la diferencia y por qué es importante?



Bases de datos no relacionales.

¿Qué es JSON?

JSON, o notación de objetos de JavaScript, es un formato de intercambio de datos legible por humanos, especificado a principios de la década de 2000. Aunque JSON se basa en un subconjunto del estándar del lenguaje de programación JavaScript, es completamente independiente del lenguaje.

Los objetos JSON **son contenedores asociativos**, en los que una clave de cadena se asigna a un valor (que puede ser un número, una cadena, un valor booleano, una matriz, un valor vacío, nulo o incluso otro objeto).

Casi todos los lenguajes de programación tienen una implementación para esta estructura de datos abstracta: objetos en JavaScript, diccionarios en Python, tablas hash en Java y C#, matrices asociativas en C++, etc. Los objetos JSON son fáciles de entender para los humanos y para que las máquinas los analicen y generen:



Bases de datos no relacionales.

¿Qué es JSON?

```
{
  "_id": 1,
  "name": { "first" : "John", "last" : "Backus" },
  "contribs": [ "Fortran", "ALGOL", "Backus-Naur Form", "FP" ],
  "awards": [
    {
      "award": "W.W. McDowell Award",
      "year": 1967,
      "by": "IEEE Computer Society"
    }, {
      "award": "Draper Prize",
      "year": 1993,
      "by": "National Academy of Engineering"
    }
  ]
}
```



Bases de datos no relacionales.

¿Qué es JSON?

A medida que JavaScript se convirtió en el lenguaje líder para el desarrollo web, JSON comenzó a cobrar vida propia. En virtud de ser legible por humanos y máquinas, y comparativamente fácil de implementar en otros lenguajes, JSON se movió rápidamente más allá de la página web y en el software en todas partes.

Hoy, JSON aparece en muchos casos diferentes:

- API
- Archivos de configuración
- Registrar mensajes
- Almacenamiento de base de datos



Bases de datos no relacionales.

La conexión MongoDB-JSON

MongoDB fue diseñado desde su inicio para ser una base de datos enfocada en brindar una excelente experiencia de desarrollo. La omnipresencia de JSON lo convirtió en la opción obvia para representar estructuras de datos en el modelo de datos de documentos de MongoDB .

Sin embargo, hay varios problemas que hacen que JSON no sea ideal para usar dentro de una base de datos.

1. JSON solo admite una cantidad limitada de tipos de datos básicos. En particular, **JSON carece de soporte para fechas y datos binarios.**
2. Los **objetos y las propiedades JSON no tienen una longitud fija**, lo que hace que el recorrido sea más lento.



Bases de datos no relacionales.

La conexión MongoDB-JSON

MongoDB fue diseñado desde su inicio para ser una base de datos enfocada en brindar una excelente experiencia de desarrollo. La omnipresencia de JSON lo convirtió en la opción obvia para representar estructuras de datos en el modelo de datos de documentos de MongoDB .

Sin embargo, hay varios problemas que hacen que JSON no sea ideal para usar dentro de una base de datos.

1. JSON solo admite una cantidad limitada de tipos de datos básicos. En particular, **JSON carece de soporte para fechas y datos binarios.**
2. Los **objetos y las propiedades JSON no tienen una longitud fija**, lo que hace que el recorrido sea más lento.



Bases de datos no relacionales.

La conexión MongoDB-JSON

Para hacer que MongoDB JSON sea primero, pero aún de alto rendimiento y propósito general, se inventó BSON para cerrar la brecha: una representación binaria para almacenar datos en formato JSON, optimizada para velocidad, espacio y eficiencia. No es diferente de otros formatos de intercambio binario como Protocol Buffers o Thrift, en términos de enfoque.



Bases de datos no relacionales.

¿Qué es BSON?

BSON significa "**JSON binario**", y eso es exactamente lo que se inventó para ser. La estructura binaria de BSON codifica información de tipo y longitud, lo que permite recorrerla mucho más rápido en comparación con JSON.

BSON agrega algunos tipos de datos que no son nativos de JSON, como fechas y datos binarios, sin los cuales a MongoDB le habría faltado un soporte valioso.

Los siguientes son algunos objetos JSON de ejemplo y sus representaciones BSON correspondientes.



Bases de datos no relacionales.

¿Qué es BSON?

```
{"hello": "world"} →  
\x16\x00\x00\x00          // total document size  
\x02                      // 0x02 = type String  
hello\x00                 // field name  
\x06\x00\x00\x00world\x00 // field value  
\x00                      // 0x00 = type E00 ('end of object')  
  
{"BSON": ["awesome", 5.05, 1986]} →  
\x31\x00\x00\x00  
  \x04BSON\x00  
  \x26\x00\x00\x00  
  \x02\x30\x00\x08\x00\x00\x00awesome\x00  
  \x01\x31\x00\x33\x33\x33\x33\x33\x33\x14\x40  
  \x10\x32\x00\xc2\x07\x00\x00  
  \x00  
  \x00
```



Bases de datos no relacionales.

¿MongoDB usa BSON o JSON?

MongoDB almacena datos en formato BSON tanto internamente como a través de la red, pero eso no significa que no pueda pensar en MongoDB como una base de datos JSON. Cualquier cosa que pueda representar en JSON puede almacenarse de forma nativa en MongoDB y recuperarse con la misma facilidad en JSON.

Cuando usa el controlador MongoDB para su lenguaje de programación favorito, trabaja con las estructuras de datos nativas para ese lenguaje. El controlador se encargará de convertir los datos a BSON y viceversa al consultar la base de datos.



Bases de datos no relacionales.

¿MongoDB usa BSON o JSON?

A diferencia de los sistemas que almacenan JSON como valores codificados en cadena o blobs codificados en binario, MongoDB utiliza BSON para ofrecer potentes funciones de indexación y consulta además del formato de datos más popular de la web.

Por ejemplo, MongoDB permite a los desarrolladores consultar y manipular objetos mediante claves específicas dentro del documento JSON/BSON, incluso en documentos anidados en muchas capas en un registro, y crear índices de alto rendimiento en esas mismas claves y valores.



Bases de datos no relacionales.

¿MongoDB usa BSON o JSON?

En primer lugar, los documentos BSON pueden contener objetos de fecha o binarios que no se pueden representar de forma nativa en JSON puro. En segundo lugar, cada lenguaje de programación tiene su propia semántica de objetos. Los objetos JSON tienen claves ordenadas, por ejemplo, mientras que los diccionarios de Python (la estructura de datos nativos más cercana que es análoga a los objetos de JavaScript) están desordenados, mientras que las diferencias en los tipos de datos numéricos y de cadena también pueden entrar en juego. En tercer lugar, BSON admite una variedad de tipos numéricos que no son nativos de JSON y muchos lenguajes los representan de manera diferente.



Bases de datos no relacionales.

JSON frente a BSON

JSON

BSON

	JSON	BSON
Codificación	Cadena UTF-8	Binario
Soporte de datos	Cadena, booleano, número, matriz, objeto, nulo	Cadena, booleano, número (entero, flotante, largo, decimal 128...), matriz, nulo, fecha, BinData
Legibilidad	Humano y Máquina	Solo máquina



Bases de datos no relacionales.

JSON frente a BSON

De hecho, JSON y BSON son primos cercanos por diseño. BSON está diseñado como una representación binaria de datos JSON, con extensiones específicas para aplicaciones más amplias y optimizado para el almacenamiento y el recorrido de datos. Al igual que JSON, BSON admite la incorporación de objetos y matrices.

Una forma particular en la que BSON se diferencia de JSON es en su compatibilidad con algunos tipos de datos más avanzados. JSON, por ejemplo, no diferencia entre números enteros (que son números redondos) y números de coma flotante (que tienen varios grados de precisión decimal).

La mayoría de los lenguajes de programación del lado del servidor tienen tipos numéricos más sofisticados (los estándares incluyen números enteros, número de punto flotante de precisión regular, también conocido como "flotante", punto flotante de precisión doble, también conocido como "doble" y valores booleanos), cada uno con su propio uso óptimo para operaciones matemáticas eficientes. operaciones.



Bases de datos no relacionales.

Flexibilidad de esquema y gobierno de datos

Una de las grandes atracciones para los desarrolladores que utilizan bases de datos con modelos de datos JSON y BSON es el esquema dinámico y flexible que proporcionan en comparación con los modelos de datos tabulares rígidos que utilizan las bases de datos relacionales.

En primer lugar, los documentos de MongoDB son polimórficos: los campos pueden variar de un documento a otro dentro de una sola colección (análogo a las tablas en una base de datos relacional). Esta flexibilidad facilita el modelado de datos de cualquier estructura y la adaptación del modelo a medida que cambian los requisitos.

En segundo lugar, no es necesario declarar la estructura de los documentos en la base de datos: los documentos se describen a sí mismos. Los desarrolladores pueden comenzar a escribir código y conservar los objetos a medida que se crean.



Bases de datos no relacionales.

Flexibilidad de esquema y gobierno de datos

En tercer lugar, si es necesario agregar un nuevo campo a un documento, se puede crear sin afectar a todos los demás documentos de la colección, sin actualizar un catálogo del sistema central y sin desconectar la base de datos. Cuando necesita realizar cambios en el modelo de datos, la base de datos de documentos continúa almacenando los objetos actualizados sin necesidad de realizar ALTER TABLE operaciones costosas o, lo que es peor, tener que rediseñar el esquema desde cero.

A través de estas ventajas, la flexibilidad del modelo de datos de documentos se adapta bien a las demandas de las prácticas modernas de desarrollo de aplicaciones.

Si bien un esquema flexible es una característica poderosa, hay situaciones en las que es posible que desee tener más control sobre la estructura de datos y el contenido de sus documentos. La mayoría de las bases de datos de documentos devuelven la aplicación de estos controles al desarrollador para que los implemente en el código de la aplicación. Sin embargo, las bases de datos de documentos más avanzadas brindan validación de esquemas, utilizando enfoques como el estándar IETF JSON Schema adoptado por MongoDB.