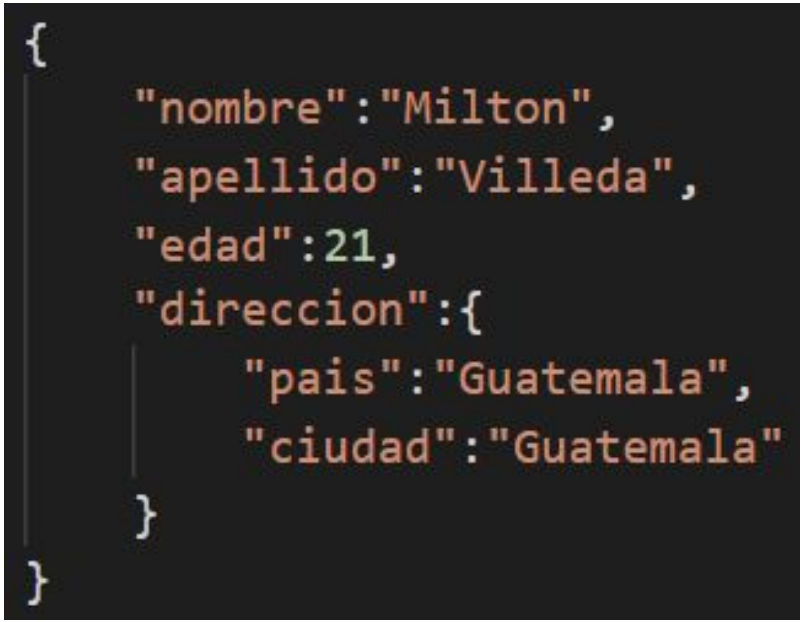
Base de datos NoSQL Documentales

Grupo 16

MongoDB

MongoDB es un sistema de base de datos NoSQL orientado a documentos que ofrece una gran escalabilidad y flexibilidad, y un modelo de consultas e indexación avanzado

Documentos


{
"nombre":"Milton",
"apellido":"Villeda",
"edad":21,
"direccion":{
"pais":"Guatemala",
"ciudad":"Guatemala"
}
}

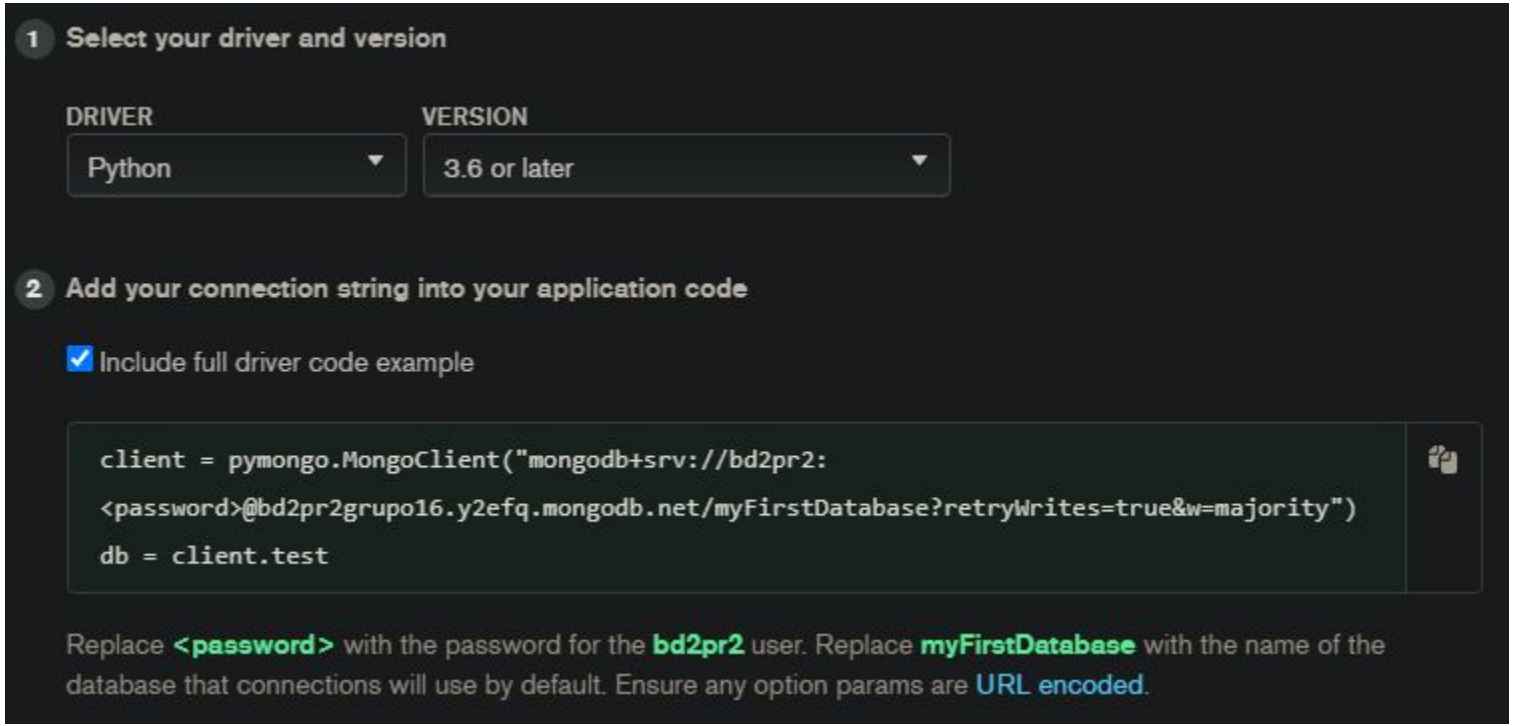
Características

- Alta disponibilidad
- Escalabilidad horizontal
- Seguridad integral
- Validación nativa de documentos
- Herramienta de gestión de automatización, monitorización y respaldo
- Completamente elástica

Fácil desarrollo

- Fácil de aprender y usar
- Drivers para lenguajes

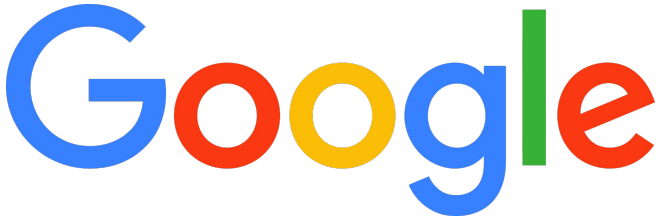

Fácil conexión al cluster



Casos de uso

- Registro de eventos
- Documentos y contenido
- Comercio electrónico
- Juegos
- Altos volúmenes de lectura
- App móvil
- Datos operacionales de sitios web

Empresas que lo utilizan



MySQL vs MongoDB

Conceptos clave

MySQL	MongoDB
Base de datos	Base de datos
Tabla	Colección
Fila	Documento
Columna	Campo
Llave primaria	_id
Joins de tabla	Lookup
Aggregation functions	Aggregation pipelines
Union all	Union with

Creación y Alteración

Crear tabla

MySQL

```
create table persona(
    id int not null,
    user_id varchar(30),
    edad number,
    estado char(1),
    primary key (id)
);
```

MongoDB

```
db.create_collection('persona')
```

Alterar una tabla: añadir campo

MySQL

```
alter table persona add fecha_union Datetime;
```

MongoDB

```
db.persona.update_many(
    {},
    { $set: { fecha_union: new Date() } }
)
```

Alterar una tabla: eliminar campo

MySQL

```
alter table persona
drop column fecha_union;
```

MongoDB

```
db.persona.update_many(
    {},
    { $unset: { fecha_union: "" } }
)
```

Eliminar tabla

MySQL

```
drop table persona;
```

MongoDB

```
db.persona.drop()
```

Inserción y selección

Insertar un registro

MySQL

```mysql
insert into
persona (id,user_id,edad,estado)
values (1,"MJA",21,"A");
```

MongoDB

```mongodb
db.persona.insert_one(
    {
        id: 1,
        user_id: "MJA",
        edad: 21,
        estado: "A"
    }
)
```

Selección de todos los registros

MySQL

```
select * from persona;
```

MongoDB

```
db.persona.find()
```

Selección de llave primaria y campos

MySQL

```mysql
select id, user_id, status
from persona;
```

MongoDB

```mongodb
db.persona.find(
    {},
    { user_id: 1, status: 1 }
)
```

Selección de campos sin llave primaria

MySQL

```
select user_id, status
from persona;
```

MongoDB

```
db.persona.find(
    {},
    { user_id: 1, status: 1, _id: 0 }
)
```

Selección de todos los registros con cláusula where

MySQL

```
select * from persona
where status = "A";
```

MongoDB

```
db.persona.find(
    { status: "A" }
)
```

Selección de campos con cláusula where

MySQL

```mysql
select user_id, status
from persona
where status = "A";
```

MongoDB

```mongodb
db.persona.find(
    { status: "A" },
    { user_id: 1, staus: 1, _id: 0 }
)
```

MySQL

```
select * from persona
where status != "A";
```

MongoDB

```
db.persona.find(
    { status: { $ne: "A"} }
)
```

Actualización

Actualizar registros

MySQL

```
update persona
set status = "C"
where edad > 25;
```

MongoDB

```
db.persona.updateMany(
    { age: { $gt: 25 }},
    { $set: { status: "C"}}
)
```

Actualizar registros

MySQL

```
update persona
set edad = edad + 3
where status = "A";
```

MongoDB

```
db.persona.updateMany(
    { status: "A"},
    { $int: { age: 3}}
)
```

Eliminación

Eliminar registros con cláusula where

MySQL

```
delete from persona
where status = "D";
```

MongoDB

```
db.persona.deleteMany(
    { status: "D"}
)
```

Eliminar registros sin cláusula where

MySQL

```
delete from persona;
```

MongoDB

```
db.persona.deleteMany({})
```

Ejemplo práctico

Sistema de base de datos documental en la Nube

¿Por qué una base de datos documental en la nube?

- Son escalables
- Mantienen la integridad de la información
- Ahorro de espacio físico
- Evitan problemas
- Aumentan la seguridad
- Alta disponibilidad

Son escalables

El proveedor de servicios puede aumentar los recursos o proporcionar más espacio de almacenamiento en función de las necesidades del cliente.

Mantienen la integridad de la información

Este tipo de almacenamiento de información puede replicar instancias de las bases de datos, permitiendo así el acceso concurrente de usuarios sin que los datos pierdan integridad.

Ahorro de espacio físico

Las empresas que contratan este tipo de servicio no necesitan instalar infraestructuras en su empresa, todo queda almacenado en los servidores del proveedor.

Evitan Problemas

Algunos de ellos muy típicos en las bases de datos tradicionales, como la imposibilidad de acceder a la información si se caen los servidores locales.

Aumentan la Seguridad

Normalmente los proveedores de almacenamiento en la nube son empresas de reputación probada en el mundo digital e incorporan soluciones para evitar brechas de seguridad o pérdidas de información.

Aumentan la Seguridad

Normalmente los proveedores de almacenamiento en la nube son empresas de reputación probada en el mundo digital e incorporan soluciones para evitar brechas de seguridad o pérdidas de información.


amazon
DynamoDB

Disney

Disney+ captura mil millones de acciones de clientes diarios para mejorar la experiencia del espectador »

Dropbox

Dropbox se ahorró millones de dólares en costos de expansión con un nuevo sistema de almacenamiento »

Snap Inc. redujo la latencia promedio en un 20 % con Amazon DynamoDB »

zoom

Zoom Video Communications, Inc. administró el incremento de 10 millones a 300 millones de participantes diarios en reuniones »

Características de Amazon DynamoDB

Amazon DynamoDB es una base de datos NoSQL que admite modelos de datos de documentos y clave valor.

DynamoDB se ha diseñado para ejecutar aplicaciones de alto rendimiento a escala de Internet que sobrecargaron las bases de datos relacionales tradicionales.

Table

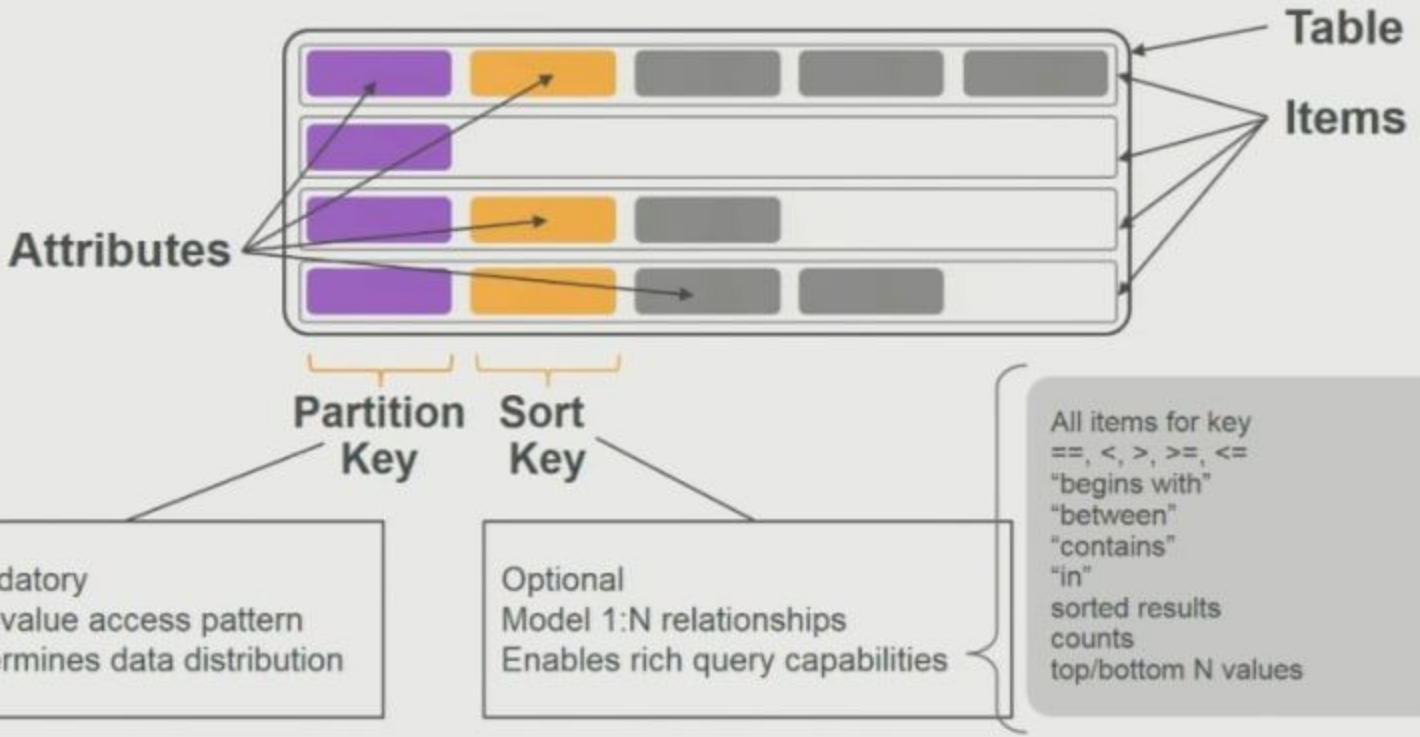


Table
Items
Attributes
Partition Key
Sort Key
Mandatory
Key-value access pattern
Determines data distribution
Optional
Model 1:N relationships
Enables rich query capabilities
All items for key
==, <, >, >=, <=
"begins with"
"between"
"contains"
"in"
sorted results
counts
top/bottom N values

Ejemplo Práctico

Big Data

Big data es un término que describe el gran volumen de datos – estructurados y no estructurados – que inundan una empresa todos los días.

- Volumen
- velocidad
- variedad

Volumen

Las organizaciones recopilan datos de diversas fuentes, como transacciones comerciales, dispositivos inteligentes (IO), equipo industrial, vídeos, medios sociales y más. En el pasado, su almacenamiento habría sido un problema - pero el almacenamiento más barato en plataformas como los data lakes y el Hadoop han aliviado la carga.

Velocidad

Con el crecimiento del Internet de las Cosas, los datos llegan a las empresas a una velocidad sin precedentes y deben ser manejados de manera oportuna. Las etiquetas RFID, los sensores y los medidores inteligentes están impulsando la necesidad de manejar estos torrentes de datos en tiempo casi real.

Variedad

Los datos se presentan en todo tipo de formatos: desde datos numéricos estructurados en bases de datos tradicionales hasta documentos de texto no estructurados, correos electrónicos, vídeos, audios, datos de teletipo y transacciones financieras.

Video