

NoSQL

Características de las BBDD No relacionales

1. **Flexibilidad de esquema:** Las bases de datos NoSQL no requieren un esquema fijo, lo que permite una estructura de datos más flexible y adaptable.
2. **Escalabilidad horizontal:** Se diseñan para escalar agregando más servidores, en lugar de mejorar el hardware de un solo servidor.
3. **Alto rendimiento:** Optimizadas para operaciones rápidas de lectura y escritura.
4. **Disponibilidad y particionamiento:** Diseñadas para ser altamente disponibles y manejar particiones de datos de manera eficiente.
5. **Modelos de datos variados:** Soportan diferentes modelos de datos como documentos, clave-valor, columnas y grafos.

Diferencias entre BBDD Relacionales y BBDD NoSQL

1. **Estructura de datos:**
 - **Relacionales:** Usan tablas con filas y columnas, y requieren un esquema fijo.
 - **NoSQL:** Utilizan estructuras como documentos, clave-valor, columnas y grafos, y no requieren un esquema fijo.
2. **Escalabilidad:**
 - **Relacionales:** Generalmente escalan verticalmente.
 - **NoSQL:** Escalan horizontalmente.
3. **Integridad de datos:**
 - **Relacionales:** Enfatizan la integridad y las relaciones a través de claves primarias y foráneas.
 - **NoSQL:** Ofrecen flexibilidad en la integridad y consistencia a favor de la disponibilidad y particionamiento.
4. **Lenguaje de consulta:**
 - **Relacionales:** Usan SQL (Structured Query Language).
 - **NoSQL:** Varían según el tipo, con lenguajes específicos o APIs para consultas.

Tipos de BBDD No relacionales

1. **Clave-valor:** Almacenan datos como pares clave-valor. Ejemplo: Redis, DynamoDB.
2. **Documentales:** Almacenan datos en documentos (como JSON). Ejemplo: MongoDB, CouchDB.
3. **Columnares:** Almacenan datos en columnas en lugar de filas. Ejemplo: Cassandra, HBase.
4. **Grafos:** Almacenan datos en nodos y aristas, optimizados para relaciones y grafos. Ejemplo: Neo4j, OrientDB.

Elementos de las BBDD No relacionales

1. **Claves y valores:** Fundamento de las bases de datos clave-valor.

2. **Documentos:** Estructuras de datos semi-estructurados en bases de datos documentales.
3. **Columnas:** Almacenamiento basado en columnas para bases de datos columnares.
4. **Nodos y aristas:** Componentes fundamentales en bases de datos de grafos.

Sistemas gestores de BBDD No relacionales

1. **MongoDB:** Base de datos documental.
2. **Cassandra:** Base de datos columnar.
3. **Redis:** Base de datos clave-valor.
4. **Neo4j:** Base de datos de grafos.
5. **Couchbase:** Combina características de bases de datos clave-valor y documentales.

Herramientas de los Sistemas gestores de BBDD No relacionales para la gestión de la información almacenada

1. **Lenguajes de consulta:** MongoDB usa MongoDB Query Language (MQL), Neo4j usa Cypher.
 1. **APIs:** La mayoría ofrece APIs en múltiples lenguajes de programación para interactuar con la base de datos.
2. **Herramientas de administración:** Interfaces gráficas como MongoDB Compass para MongoDB, o herramientas de línea de comandos.
3. **Monitoreo y administración:** Herramientas para monitorear el rendimiento y la integridad, como Prometheus y Grafana para bases de datos NoSQL.
4. **Integración y replicación:** Herramientas para facilitar la integración con otras plataformas y la replicación de datos para alta disponibilidad.

Práctica

- **Crear una base de datos denominada *mongoTienda*.**

```
use mongoTienda
```

Salida:

```
test> use mongoTienda
switched to db mongoTienda
```

- Insertar al menos 6 documentos, con información sobre los artículos comercializados, en una colección que llamaremos *Articulos*.

```
db.Articulos.insertMany([
  { nombre: "Gatito", precio: 3, categoria: "Animales" },
  { nombre: "Bambas Nike", precio: 50, categoria: "Calzado" },
  { nombre: "Pantalones", precio: 30, categoria: "Ropa" },
  { nombre: "Reloj", precio: 62, categoria: "Accesorios" },
  { nombre: "Rai", precio: 777, categoria: "Aniamles" },
```

```
{ nombre: "Mochila", precio: 40, categoria: "Accesorios" }  
}]
```

Salida:

```
mongoTienda> db.Articulos.insertMany([  
  { nombre: "Bambas Nike", precio: 50, categoria: "Calzado" },  
...   { nombre: "Gatito", precio: 3, categoria: "Animales" },  
...   { nombre: "Bambas Nike", precio: 50, categoria: "Calzado" },  
...   { nombre: "Pantalones", precio: 30, categoria: "Ropa" },  
...   { nombre: "Reloj", precio: 62, categoria: "Accesorios" },  
...   { nombre: "Rai", precio: 777, categoria: "Aniamles" },  
...   { nombre: "Mochila", precio: 40, categoria: "Accesorios" }  
... ])  
{  
  acknowledged: true,  
  insertedIds: {  
    '0': ObjectId('66503e06b4d148cc4e2202d8'),  
    '1': ObjectId('66503e06b4d148cc4e2202d9'),  
    '2': ObjectId('66503e06b4d148cc4e2202da'),  
    '3': ObjectId('66503e06b4d148cc4e2202db'),  
    '4': ObjectId('66503e06b4d148cc4e2202dc'),  
    '5': ObjectId('66503e06b4d148cc4e2202dd')  
  }  
}
```

- Mostrar todos los documentos almacenados en la colección.

```
db.Articulos.find()
```

Salida:

```
mongoTienda> db.Articulos.find()
[
  {
    _id: ObjectId('66503e06b4d148cc4e2202d8'),
    nombre: 'Gatito',
    precio: 3,
    categoria: 'Animales'
  },
  {
    _id: ObjectId('66503e06b4d148cc4e2202d9'),
    nombre: 'Bambas Nike',
    precio: 50,
    categoria: 'Calzado'
  },
  {
    _id: ObjectId('66503e06b4d148cc4e2202da'),
    nombre: 'Pantalones',
    precio: 30,
    categoria: 'Ropa'
  },
  {
    _id: ObjectId('66503e06b4d148cc4e2202db'),
    nombre: 'Reloj',
    precio: 62,
    categoria: 'Accesorios'
  },
  {
    _id: ObjectId('66503e06b4d148cc4e2202dc'),
    nombre: 'Rai',
    precio: 777,
    categoria: 'Aniamles'
  },
  {
    _id: ObjectId('66503e06b4d148cc4e2202dd'),
    nombre: 'Mochila',
    precio: 40,
    categoria: 'Accesorios'
  }
]
```

- **Realizar una búsqueda de algún artículo concreto.**

```
db.Articulos.findOne({ nombre: "Rai" })
```

Salida:

```
mongoTienda> db.Articulos.findOne({ nombre: "Rai" })
{
  _id: ObjectId('66503e06b4d148cc4e2202dc'),
  nombre: 'Rai',
  precio: 777,
  categoria: 'Aniamles'
}
```

- **Modificar algún dato de un artículo**

```
db.Articulos.updateOne({ nombre: "Gatito"}, {$set: {precio: 0.5} })
```

Salida:

```
mongoTienda> db.Articulos.updateOne({ nombre: "Gatito"}, {$set: {precio: 0.5} })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

- **Borrar el documento perteneciente a un artículo**

```
db.Articulos.deleteOne({ nombre: "Mochila" })
```

Salida:

```
mongoTienda> db.Articulos.deleteOne({ nombre: "Mochila" })
{ acknowledged: true, deletedCount: 1 }
```

- **Mostrar la información de todos los documentos**

```
db.Articulos.find()
```

```
mongoTienda> db.Articulos.find()
[
  {
    _id: ObjectId('665039db9f8fd2b0f32202d8'),
    nombre: 'Gatito',
    precio: 0.5,
    categoria: 'Animales'
  },
  {
    _id: ObjectId('665039db9f8fd2b0f32202d9'),
    nombre: 'Bambas Nike',
    precio: 50,
    categoria: 'Calzado'
  },
  {
    _id: ObjectId('665039db9f8fd2b0f32202da'),
    nombre: 'Pantalones',
    precio: 30,
    categoria: 'Ropa'
  },
  {
    _id: ObjectId('665039db9f8fd2b0f32202db'),
    nombre: 'Reloj',
    precio: 62,
    categoria: 'Accesorios'
  },
  {
    _id: ObjectId('665039db9f8fd2b0f32202dc'),
    nombre: 'Rai',
    precio: 777,
    categoria: 'Aniamles'
  }
]
```