Certified Tech Developer
The Ultimate Degree

Base de datos II

Modelado en bases de datos NoSQL

NoSQL o 'Not Only SQL' es un modelo de datos que difiere totalmente de las expectativas tradicionales de SQL.

La principal diferencia es que NoSQL enfatiza el **design flexible**. La **falta de requisitos** para un esquema hace que el diseño sea un proceso **mucho más simple y económico.** Esto no quiere decir que no pueda usar un esquema por completo, sino que el diseño esquemático es muy flexible.

Otra característica muy importante de los **modelos de datos NoSQL** es que están diseñados para una alta eficiencia y velocidad en términos de creación de **millones de consultas por segundo**. Esto se logra a través de menos solicitudes al motor de base de datos, creando documentos con más información, evitando consultas innecesarias.

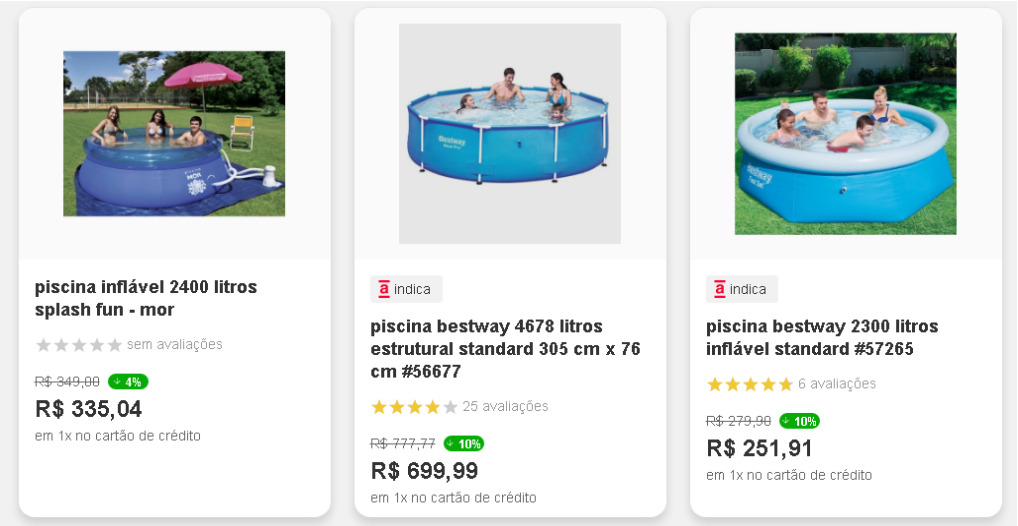
El modelado de datos NoSQL a menudo comienza con consultas específicas de la aplicación. Esto se opone al modelado relacional:

- El **modelado relacional** normalmente se guía por la estructura de los datos disponibles. El tema principal del diseño es **"¿Qué respuestas tengo?"**

- El **modelado de datos NoSQL** a menudo está impulsado por **patrones de acceso específicos de la aplicación**, es decir, los tipos de consultas que se admiten. El tema principal del diseño es **"¿Qué preguntas tengo?"**

Por lo tanto, para comenzar a modelar una base de datos NoSQL, primero estudiamos los resultados que queremos obtener y mostrar en nuestra aplicación. Analizamos cómo responder a lo que se preguntaba y luego decidimos el mejor modelo.

Como ejemplo, imaginen que queremos crear una base de datos para una tienda online. Observamos la aplicación e identificamos qué información será necesaria para que la aplicación funcione. Miren la imagen de abajo.



En la imagen de arriba, podemos notar que necesitamos mostrar la imagen del producto, una descripción, capacidad, calificación y valor.

Ahora que sabemos cómo funciona la aplicación, será mucho más práctico crear un único documento que reciba toda esta información. Así, cuando el usuario busque el producto, estará consultando un único documento, o mejor dicho, necesitará una sola consulta para traer toda la información

```
{
        _id: <ObjectID01>,
        nombre: "splash fun-mar",
        desc: "piscina inflável",
        capacidad: "2400 litros"
        valoración: 3,
        imagen: "piscina01.jpg"
        valor: 335.04
}
```

Los modelos de datos efectivos respaldan las necesidades de las aplicaciones. La principal consideración para la estructura de sus documentos es la decisión de incorporar o utilizar referencias.

Supongamos que en esta tienda, al acceder a la página del producto, también se muestra la categoría del producto. Esta información puede estar en el mismo documento o en otro. Entonces, si elige incorporar esta información, el documento se vería así:

```
{
    _id:<ObjectID01>,
    nombre: "splash fun-mar",
    desc: "piscina inflavel",
    capacidad: "2400 litros",
    valoracion: 3,
    imagen: "piscina01.jpg",
    valor: 335.04 ,
    catergoria:
                {
                    idCategoria: 1236,
                    nombre: piscina
                }
}
```

Tenga en cuenta que la categoría es otro documento, incorporado en el documento del producto.

Los modelos de datos integrados permiten que las aplicaciones almacenen información relacionada en el mismo registro de la base de datos. Como resultado, las aplicaciones deben emitir menos consultas y actualizaciones para completar operaciones comunes

En general, conviene que utilicen modelos de datos integrados cuando:

- Tengan la relación **uno-a-uno** entre entidades.

- Tengan la relación **uno-a-muchos** entre entidades. En esa relación, los documentos "muchos" o hijos siempre aparecen como son vistos en contexto de "uno" o documento padre.

La incorporación proporciona un mejor rendimiento para las operaciones de lectura, así como la capacidad de solicitar y recuperar datos relacionados en una única operación de base de datos. Los modelos de datos integrados permiten actualizar los datos relacionados en una sola operación de grabación atómica.

Pero si eligen usar la referencia, el modelo se verá como el que se usa en SQL.

```
{
    _id:<ObjectID01>,
    nombre: "splash fun-mar",
    desc: "piscina inflavel",
    capacidad: "2400 litros",
    valoracion: 3,
    imagen: "piscina01.jpg",
    valor: 335.04 ,
    idCatergoria:<ObjectID02>
}
```

```
"categoria": {
                idCategoria: <ObjectID02>,
                nombre: piscina
            }
```

En la imagen de arriba, tenemos dos documentos: productos y categoría. Y en la tabla de productos solo hacemos referencia a la identificación de categoría.

Utilicen referencia cuando:

- La incorporación daría como resultado la duplicación de datos, pero no proporcionaría suficientes ventajas de rendimiento de lectura para superar las implicaciones de la duplicación.

- Para representar la relación muchos-a-muchos más compleja.

- Para modelar grandes conjuntos de datos jerárquicos.

Para más información y futuras consultas, utilicen la documentación oficial.