BASES DE DATOS NO SQL

NoSQL reúne conceptualmente a todas las bases de datos que no siguen los principios y conceptos de las bases relacionales es decir NoSQL es un conjunto de tecnologías que representa diferentes principios para el almacenamiento abre el almacenamiento y manipulación masiva de datos desde los años 80 la mayoría de los sistemas de información almacena sus datos en gestores relacionales básicamente estos recogen procesos transaccionales que gestionan datos simples y estructurados por ejemplo otras acciones bancarias compras ventas etc la introducción de NoSQL en el mercado fue una respuesta para llenar los espacios donde los sistemas gestores de bases de datos relacionales ya no podían ser útiles por la cantidad enorme de datos y por la velocidad requerida de almacenamiento y recuperación paralelamente los avances en virtualización condujeron a la construcción de nodos de computación en la nube los cloud computing que a su vez requerían nodos de almacenamiento cloud storage para procesar de forma paralela cantidades masivas de datos Google creó un sistema de archivos distribuidos una base de datos orientada a columnas un sistema gestor distribuido y un algoritmo que llamó mapreduce y que da soporte al procesamiento paralelo de datos masivos fue implementado inicialmente por hadoop este algoritmo junta todos los pares con la misma clave de todas las listas y los agrupa creando un grupo por cada una de las diferentes claves generadas para luego transformar la lista de pares clave valor en una lista de valor el término big data se refiere a esa cantidad masiva de datos.

El concepto big DAta refiere a conjuntos de datos cuyo tamaño excede la capacidad de los típicos sistemas gestores de base de datos para capturarlos almacenarlos gestionarlos y analizar el motor de un avión por ejemplo es capaz de generar más de 10 terabytes de datos en solo 30 minutos de vuelo Netflix utiliza modelos predictivos para informar de forma personalizada las novedades y recomendaciones clasificando los datos del pasado y los programas vistos o marcados como favoritos y que representan una cantidad también enorme de datos hay cada vez más dispositivo hay cada vez más dispositivos conectados a internet recopilando datos sobre patrones de uso de clientes o redes sociales y eficacia de productos y por qué no usarlos para que las computadoras aprendan por sí solas así nació el aprendizaje automático o machine learning el objetivo que persiguen las tecnologías big data es el de tratar de obtener información que guardan enormes cantidades de datos esto debe lograrse a través del análisis profundo de esos datos.

En BIG DATA hay 3 tipos de datos estructurados semi estructurados y no estructurados en cada uno de estos tipos hay mucha información útil que se puede extraer para usarla en distintos proyectos los datos estructurados tienen un formato fijo y a Menudo son numéricos así que en muchos casos los gestionan las computadoras y no los humanos este tipo de datos esa información que ya está ordenada en bases de datos de tipo sql y almacenes de datos data Waterhouse . los datos no estructurados son información que está desorganizada y no está en un formato predeterminado es el caso por ejemplo de los datos recopilados de fuentes de redes sociales y pueden convertirse en archivos de documentos de texto almacenados en sistemas no SQL.

Semi Estructurados pueden contener ambas formas de datos como registros de servidores web o datos de sensores incluye siempre múltiples fuentes y la mayor parte del tiempo es de distintos tipos también así que no siempre es fácil saber cómo integrar todas las herramientas para trabajar con distintos tipos de datos la necesidad de gestionar datos también requiere una infraestructura estable y bien organizada habrá que procesar rápidamente grandes volúmenes y distintos tipos de datos y esto puede sobrecargar un único servidor.

Big Data se caracteriza por las llamadas 6 V volumen Google procesa diariamente 20 petabytes un petabyte son 1000 terabytes velocidad deben ser almacenados y procesados de modo que el sistema responda casi en tiempo real a la demanda variedad la mayoría de las veces en formato multimedia son datos no estructurados a los que hay que agregar además los metadatos veracidad se refiere a lo exacto que son los datos de ese conjunto los datos inciertos pueden dar lugar a análisis imprecisos y llevar a tomar decisiones equivocadas valor es necesario que hable ser una forma de valorar y limpiar los datos y aceptar sólo los que son relevantes para el propósito del análisis y variabilidad refiere a los datos cuyo valor o significado cambian constantemente dependiendo del fin para los que son analizados y utilizados.

según el teorema de brewer no es posible que un sistema distribuido pueda garantizar simultáneamente estas 3 características consistencia en el sistema distribuido habitualmente se dice que se encuentra en un estado consistente si después de una operación de escritura todas las operaciones de lectura posteriores son capaces de ver las actualizaciones. Disponibilidad implica que el sistema ha sido diseñado e implementado de modo que se pueda continuar operando lecturas y escrituras incluso después de que un nodo no esté activo o que algunas partes del hardware tuvieran que ser desconectadas debido a errores o actualizaciones y tolerancia a particiones que es la habilidad de un sistema de tener diferentes regiones o divisiones lógicas en la red y de ser capaz de seguir funcionando aunque una de estas partes queda inaccesible durante un tiempo el teorema indica que en un ambiente distribuido la caída de un nodo es un hecho probable y frente a este hecho el sistema deberá elegir entre dar una respuesta al cliente pero que no garantice mantener la consistencia no dar una respuesta al cliente bajando la disponibilidad pero manteniendo la consistencia es decir se pueden garantizar sólo 2 y los sistemas gestores de base de datos cumplen siempre 2 de ellas la base documental pongo por ejemplo cumple con los principios de coherencia y tolerancia particiones mientras que la base columnar cassandra por su parte cumple con los principios de disponibilidad y tolerancia a las particiones.

En cuanto a la arquitectura en general proveen una consistencia débil de datos en generar una consistencia eventual emplean una arquitectura distribuida donde los datos están almacenados de forma redundante en varios servidores generalmente ofrecen estructura de datos simples como arrays asociativos o estructura de tipo llave de valor las consultas se realizan exclusivamente por las llaves o índices y las consultas complejas se realiza mediante una infraestructura de procesamiento externo tal como el algoritmo Map Reduce.

El mundo de las bases de datos relacionales está familiarizado con las transacciones ácido atomicidad consistencia aislamiento y durabilidad estas transacciones son más estrictas que lo que en muchos casos el dominio de la aplicación requiere por eso no se huele debilita alguno de estos requisitos. Ya pases no principio y se rigen por el modelo llamado base que abarca los siguientes conceptos disponibilidad básica básica bailado garantiza la disponibilidad de los datos la mayoría del tiempo habrá una respuesta a cualquier solicitud pero también puede haber fallas degradación o soft state el estado del sistema puede cambiar con el tiempo y consistencia eventual eventual consiste así el sistema eventualmente se volverá consistente una vez que deje de recibir información los datos en las diferentes réplicas no tienen que ser mutuamente consistentes en todo momento.

Dentro del concepto de bases no SQL existen varios modelos implementados vamos a enfocarnos en 2 de ellos las bases documentales y las bases columnas en las bases documentales este tipo de bases de datos no es un sistema administrador de documentos como se puede llegar a pensar sino que refiere principalmente a conjuntos estructurados de datos en documentos publicaciones impresas documentos audiovisuales gráficos y sonoros documentos de archivo documentos electrónicos etcétera que pueden ser conformado con un conjunto en un área específica además de ser indexados no sólo con base a un identificador primario sino también Asus atributos las codificaciones más habituales de estos documentos suelen ser en XML o jason pero también se pueden almacenar en formato Word o PDF permiten almacenar textos complejos por ejemplo leyes o expedientes y explotar esta información a través de consultas de tipo full text en forma eficiente las consultas full text permiten al usuario recuperar los documentos que contengan palabras o frases dadas en cualquier lugar del documento habitualmente los documentos se estructuran dentro de una serie de contenedores llamados colecciones que son proporcionados por el sistema gestor documental los documentos que son almacenados dentro de la base de datos con una clave única dentro del repositorio y por esa clave son también recuperados posteriormente también existirá un índice principal por esta clave primaria por cada uno de los documentos del conjunto en la base de mongo es un ejemplo de este tipo de base no SQL.

En las bases columnaris las unidades de datos ya no son insertadas como filas convirtiéndolas en pares de llaves y valores dando la definición de cada una de las columnas son parejas de valores identificadas por una llave y los arrays son una de las estructuras más usadas en las cuales se almacena más de un valor para una llave se estructuran almacenando la información como si se tratara de un diccionario el diccionario en este caso contiene tuplas clave valor llave valor los clientes añaden y solicitan valores a partir de una clave o llave asociada que conocen de antemano las comparativas de rendimiento entre los sistemas de gestión por filas y los basados en columnas vienen principalmente de la mano de la eficiencia de los accesos que realizan al disco los accesos a posiciones consecutivas se producen en una cantidad de tiempo sensiblemente menor que los accesos que se producen a posiciones aleatorias en disco este modelo de columnas es utilizado con más frecuencia en ambientes de esta web para el análisis olap entre las bases de datos más conocidas tenemos eh casandra hyper table o hide base.

Tenemos 2 ejemplos en el primero según el modelo relacional cada registro contiene todos los atributos definidos para un cierto identificador nombre fecha de nacimiento salario etcétera al transformarse al modelo columnar un registro contiene todos los nombres otro contiene todas las fechas de nacimiento etcétera en el segundo ejemplo vemos una vista lógica del modelo columnar.

Crear un nuevo registro significa que debe existir una forma de identificar que el registro ya existe sino éste debe ser actualizado y no creado nuevamente en las bases de datos relacionales los registros son almacenados según una clave primaria que identifica a cada registro como único cuando es necesario realizar una inserción se realiza una búsqueda por medio de esta clave para determinar si el valor existe o no las bases datos columnaris no están definidas el concepto de relaciones ya que las tecnlogias NoSQL evitan las uniones entre los datos estas bases de datos almacenan los datos de forma desnormalizada recurren entonces a la dimensión tiempo para almacenar los datos esto significa que la base de datos diferentes versiones del valor asociado en la familia de columnas por lo que el valor mostrado al momento de ser seleccionado es siempre el más reciente las versiones anteriores pueden ser consultadas ya que también son almacenadas la forma más fácil y eficiente de ejecutar consultas en la base de datos columnares es por medio un indicador que direccione al dato que se desea consultar, por lo que siempre es necesario indicar una llave o clave que identifique cada uno de los registros consultados.

Su uso es adecuado para aquellas aplicaciones que manejan grandes volúmenes de datos tienes una respuesta alta de accesos de lectura y escritura gestionan cambios frecuentes en los esquemas de datos y no requieren la consistencia ACID.

También hay que considerar los inconvenientes que presentan las bases de datos no SQL.

tienen una capacidad de consulta limitada

son un modelo estandarizado eso trae problemas de portabilidad entre diferentes entornos

el desarrollo de aplicaciones clientes es mas complejo

carecen de herramientas de control de acceso las cuestiones de seguridad son débiles.