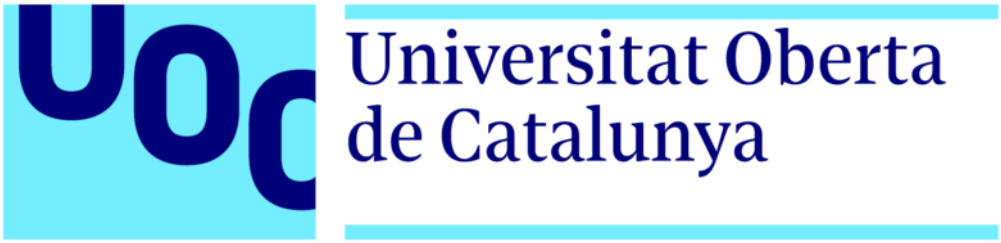


Máster universitario de Ciencia de Datos

Prueba de Evaluación Continua – PEC1

Arquitecturas de bases de datos no tradicionales – Conceptos básicos y modelo de datos.

Autor:

Mario Ubierna San Mamés

Índice de Contenido

1. Enunciado

1.1. Ejercicio 1

A partir de la lectura de los apuntes (locuciones de los vídeos) de los temas I y II se pide responder de manera concisa (una página y media en total) a las siguientes preguntas:

1.1.1. ¿Qué significa que las bases de datos NoSQL orientadas hacia agregados favorecen los esquemas de crecimiento horizontal (escalabilidad horizontal)?

Lo primero de todo es determinar qué es la escalabilidad horizontal, la podemos definir como la capacidad que tiene nuestro sistema de aumentar la capacidad de cómputo y almacenamiento añadiendo más nodos u ordenadores.

Con las bases de datos NoSQL se busca priorizar la disponibilidad de los datos frente a la consistencia de los mismos, es decir, se tiene una gran cantidad de datos que están altamente distribuidos, es por ello que la bases de datos NoSQL orientadas hacia agregados se comportan mejor en estos casos, ya que el agregado se puede replicar y distribuir por los diferentes nodos.

1.1.2. Explica las ventajas del modelo de procesamiento map-reduce.

MapReduce es un *framework* que permite procesar de forma eficiente información de los agregados que hay en la base de datos, gracias a la programación paralela. Esto lo hace dividiendo una consulta compleja en otras más simples, de tal forma que cada consulta simple se ejecuta en cada nodo que contenga datos útiles y luego se unen para convertir los datos en información.

Esto tiene dos ventajas, la primera, minimiza el tiempo de respuesta (al dividir las consultas de forma paralela es como si tuviéramos diferentes hilos ejecutándose de un mismo programa, de tal forma que cada hilo nos proporciona la información que queremos. Si esta tecnología fuera secuencial, esta ventaja no la obtendríamos). La

segunda, minimiza el transporte de datos innecesarios (como las consultas se ejecutan de forma paralela, accedemos a los datos de forma local en cada nodo, entonces a la hora de transmitir la información por la red solo se envía lo que se ha solicitado).

1.1.3. Explica qué es la persistencia políglota y cómo se refleja en un proyecto.

La persistencia políglota no es más que el uso de diferentes tecnologías de bases de datos para hacer frente a diferentes problemáticas, cuando hablamos de persistencia políglota no nos estamos refiriendo a hacer uso exclusivo de una sola tecnología, sino que dentro de una misma organización y proyecto se pueden usar varias.

Dentro de un proyecto puede verse reflejada esta persistencia cuando se usan diferentes tecnologías de almacenamiento de datos, por ejemplo si somos *Twitter*, podemos tener una base de datos orientada a grafos para almacenar los datos generados en la interacción de usuarios, y por otro lado tener una base de datos relacional, en la cual se almacena la gestión de usuarios, la gestión de compañías publicitarias…

1.1.4. Explica las razones por las que un modelo relacional no es buena opción para procesar flujos de datos.

Los sistemas que trabajan con un gran flujo de datos se caracterizan porque prima la disponibilidad a la consistencia de los mismos, es decir, características que son opuestas en los modelos relacionales.

Una de las razones por lo que un modelo relacional no es buena idea, es por el hecho de que estos modelos necesitan primero almacenar los datos en la base de datos para luego ser consultados, mientras que los modelos que tienen un elevado flujo de datos permiten consultar los datos antes de almacenarlos en la base de datos, esto se debe a que los puede almacenar en la memoria, con esto quiero decir que hay casos en los que los datos se tienen y necesitan en tiempo real (mercado bursátil, control de tráfico, líneas de producción…). Otra razón, es que a veces no es necesario almacenar todo el flujo de datos, es decir, hay una estructura variable, y lo modelos relacionales no soportan dicha variación.

1.1.5. ¿Diferencias entre el modelo relacional y el modelo en grafo en cuanto a la representación de las relaciones?

La principal diferencia entre el modelo relacional y el modelo en grafo respecto a las relaciones es que, en el modelo relacional las relaciones representan conceptos (equipo de fútbol) y las filas son instancias de los conceptos (Real Madrid), es decir, son construcciones distintas. Sin embargo, en el modelo en grafo los conceptos y las instancias se representan de la misma forma, con los nodos.

1.2. Ejercicio 2

A partir de la lectura de los apuntes (locuciones de los vídeos) de los temas I y II indica si te parecen ciertas o falsas las siguientes afirmaciones.

1.2.1. Afirmación 1

En un modelo de agregación, la estructuración de agregados de un mismo tipo no puede variar en ningún caso. Es decir, todos los agregados del mismo tipo deben seguir la misma estructura.

Esta afirmación es falsa, según el PDF "*B2_T3_2_ModelosAgregacionCaracterísticas*" en la página 7 y segundo párrafo afirma lo contrario, es decir, que la estructuración de agregados de un mismo tipo puede variar. Por ejemplo, podemos tener un agregado de cliente con una dirección y otro agregado con tres direcciones. Los modelos de agregación se caracterizan porque tienen un esquema flexible.

1.2.2. Afirmación 2

El modelo en grafo es igual de fácil de escalar que los modelos agregados.

Esta afirmación es falsa, según el PDF "*B2_T4_1_ModelosEnGrafo*" en la página 17 y último párrafo indica todo lo contrario. Un modelo en grafo no es tan fácil de escalar como un modelo de agregación, esto se debe a que en los modelos en grafo los datos están altamente relacionados, es decir, la distribución de los datos por cada uno de los nodos de la red es más compleja y por lo tanto más complicada.

1.2.3. Afirmación 3

En un proyecto donde se debe priorizar la disponibilidad frente a la consistencia de los datos, son igual de recomendables una base de datos relacional que una base de datos NoSQL.

Esta afirmación es falsa, en el PDF "*B1_T2_PersistenciaPoliglota*" página 16 y último párrafo podemos ver que la afirmación no es correcta, es decir, tanto las bases de datos relacionales como las NoSQL permiten la distribución de los datos pero, si lo que se busca es mejorar la disponibilidad son más recomendables las bases de datos NoSQL debido a la escalabilidad horizontal, gracias a esto se pueden tener más réplicas de los datos en la red de nodos. Si por el contrario, buscamos una mayor consistencia es mejor una base de datos relacional.

1.2.4. Afirmación 4

Los modelos de agregación son una buena elección en el caso de modelos conceptuales donde predominen asociaciones jerárquicas, asociaciones 1 a 1 y relaciones de tipo parte- todo.

Esta afirmación es verdadera, según el PDF "*B2_T3_2_ModelosAgregacionCaracteristicas*" página 10 y penúltimo párrafo, es decir, los modelos de agregación son una buena elección siempre y cuando no existan relaciones complejas, en otras palabras, cuando las asociaciones son jerárquicas, 1 a 1 y parte-todo.

1.2.5. Afirmación 5

El modelo documental es un caso particular del modelo clave-valor, pero más flexible que el clave-valor.

Esta afirmación es falsa, según el PDF "*B2_T3_3_ModelosAgregacionTipos*" página 6 y segundo párrafo, el modelo documental sí que es una extensión del modelo clave-valor, pero no es más flexible, sino que todo lo contrario, el modelo documental incluye una estructura interna la cual reduce esta flexibilidad.

1.3. Ejercicio 3

La biblioteca de la UOC ha diseñado un sistema para conocer los hábitos de las personas que solicitan préstamos de sus libros. Para ello se quiere explotar la información que se almacena cada vez que un usuario solicita el préstamo de un recurso de la biblioteca.

Actualmente la información se encuentra almacenada en una base de datos relacional de tipo Oracle. La dirección de la biblioteca ha solicitado ayuda al equipo docente de la asignatura de Arquitectura de bases de datos no tradicionales, y de acuerdo al tipo de consultas que les gustaría realizar, el equipo docente ha recomendado realizar una migración de la información a una base de datos de tipo documental.

Además, se ha acordado que los estudiantes de esta asignatura ayudarán a diseñar los tipos de documentos más eficientes para las siguientes dos consultas que desean realizar:

Consulta 1: Teniendo en cuenta la titulación en la que están matriculados los estudiantes, interesa conocer información acerca de los estudiantes que solicitaron un préstamo por cada mes y titulación. El objetivo es hacer un estudio de estacionalidad, por lo tanto, no consideraremos el año, sólo consideraremos los meses. Concretamente

para cada mes y por cada titulación, se desea recuperar los datos (DNI, nacionalidad, país donde vive) de los estudiantes de esa titulación, así como, por cada estudiante, la lista de recursos que tomaron en préstamo. De cada recurso se quiere obtener el título, ISBN, tipo de recurso y asignatura de la titulación para la cual está recomendado el recurso prestado (se supone que todo los préstamos que solicitan los estudiantes están relacionados con alguna asignatura de la titulación que están cursando).

Consulta 2: También interesa conocer información de los recursos prestados agrupados por las asignaturas y titulación en las que están recomendados. Concretamente para cada titulación y por cada asignatura (una asignatura queda especificada en el plan de estudios indicando su nombre y el curso en el que se imparte) que forman parte del plan de estudios de la misma, se desea recuperar los datos principales de cada recurso prestado: título, ISBN, tipo de recurso y una lista con la información (DNI, nacionalidad, país donde vive) de cada uno de los estudiantes que solicitaron el préstamo del recurso.

Cada consulta requiere un tipo concreto de documento. En este sentido, se pide indicar:

- *Una representación gráfica del documento propuesto utilizando un diagrama de cajas anidadas como el que se explica en los apuntes sobre "Diseño de agregados" suministrado en el aula virtual. Como alternativa se puede presentar el documento en formato JSON pero se debe elegir entre una representación u otra.*

- *Una breve explicación de la estructura del documento y una justificación del porqué de su estructura.*

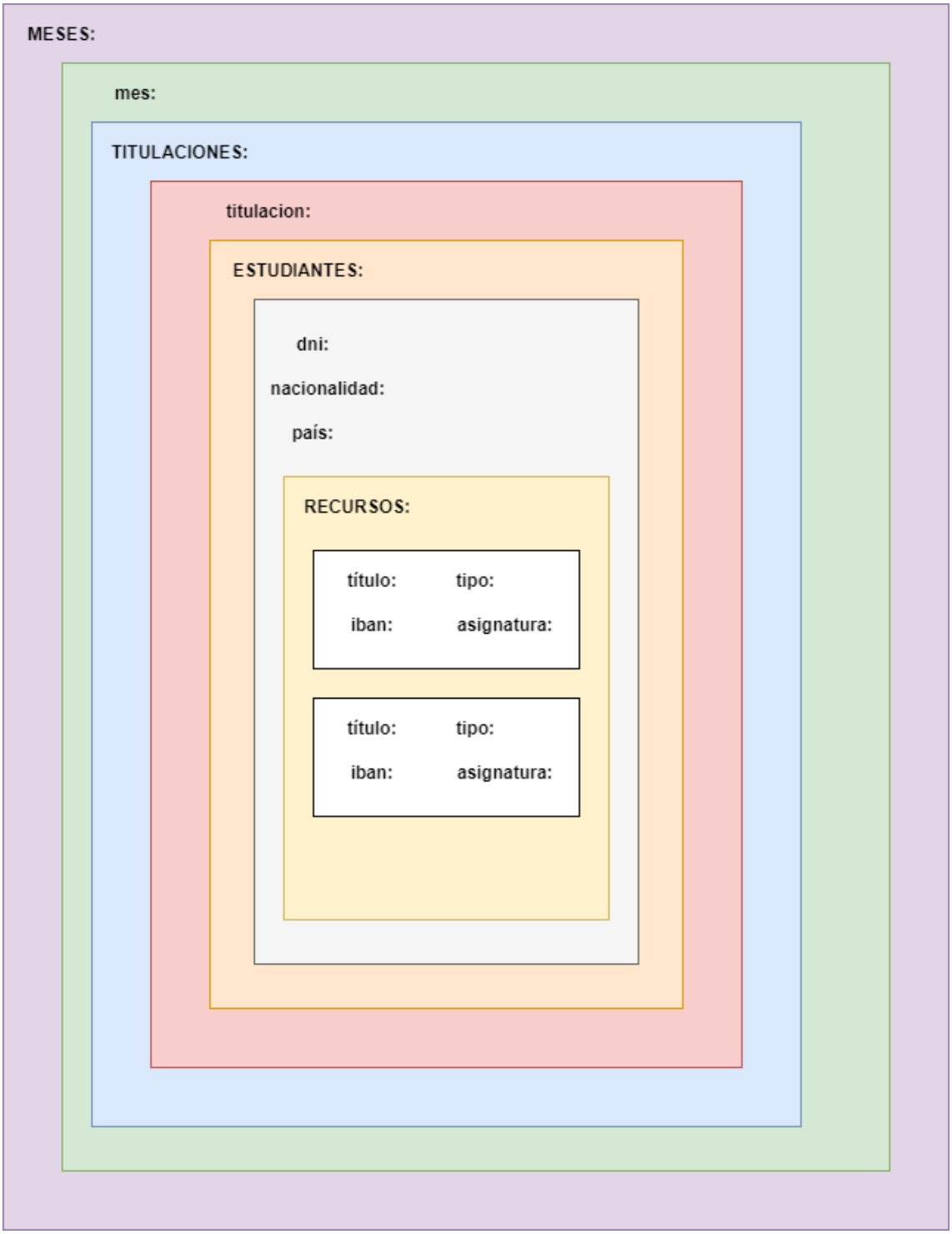
1.3.1. Consulta 1

Esta primera consulta tiene como objetivo obtener la información sobre los estudiantes que solicitaron un préstamo por cada mes y titulación, es decir, para cada mes y por cada titulación se tiene que recuperar los datos de los estudiantes, y a su vez por cada estudiante los recursos utilizados.

Teniendo esto claro, se ha definido un campo denominado “MESES” que contiene una colección de agregados para cada mes (lo forman el campo “mes” y “TITULACIONES”), “TITULACIONES” contiene una colección de agregados por cada una de las titulaciones de la universidad, cuyos atributos son “titulacion” y una nueva colección de estudiantes para cada titulación denominada “ESTUDIANTES”. Dentro de “ESTUDIANTES” tenemos los atributos que queremos obtener, es decir, el dni, la nacionalidad y el país, y tenemos otro campo que es una colección denominado “RECURSOS”, el cual contiene los recursos asociados para cada estudiante en una

determinada titulación y en un determinado mes. “RECURSOS” contiene la información que se quiere obtener sobre los recursos, es decir, el título, el iban, el tipo y la asignatura.

Se ha elegido esta estructura porque es la que mejor se adapta a la consulta que se quiere realizar, es decir, tal y como se ha definido en el primer párrafo el objetivo de esta consulta, podemos ver que la mejor forma de hacerla es encadenando cada uno de los meses, con las titulaciones, éstas a su vez con los estudiantes que cursan dichas titulaciones y los préstamos realizados por cada uno de los estudiantes.



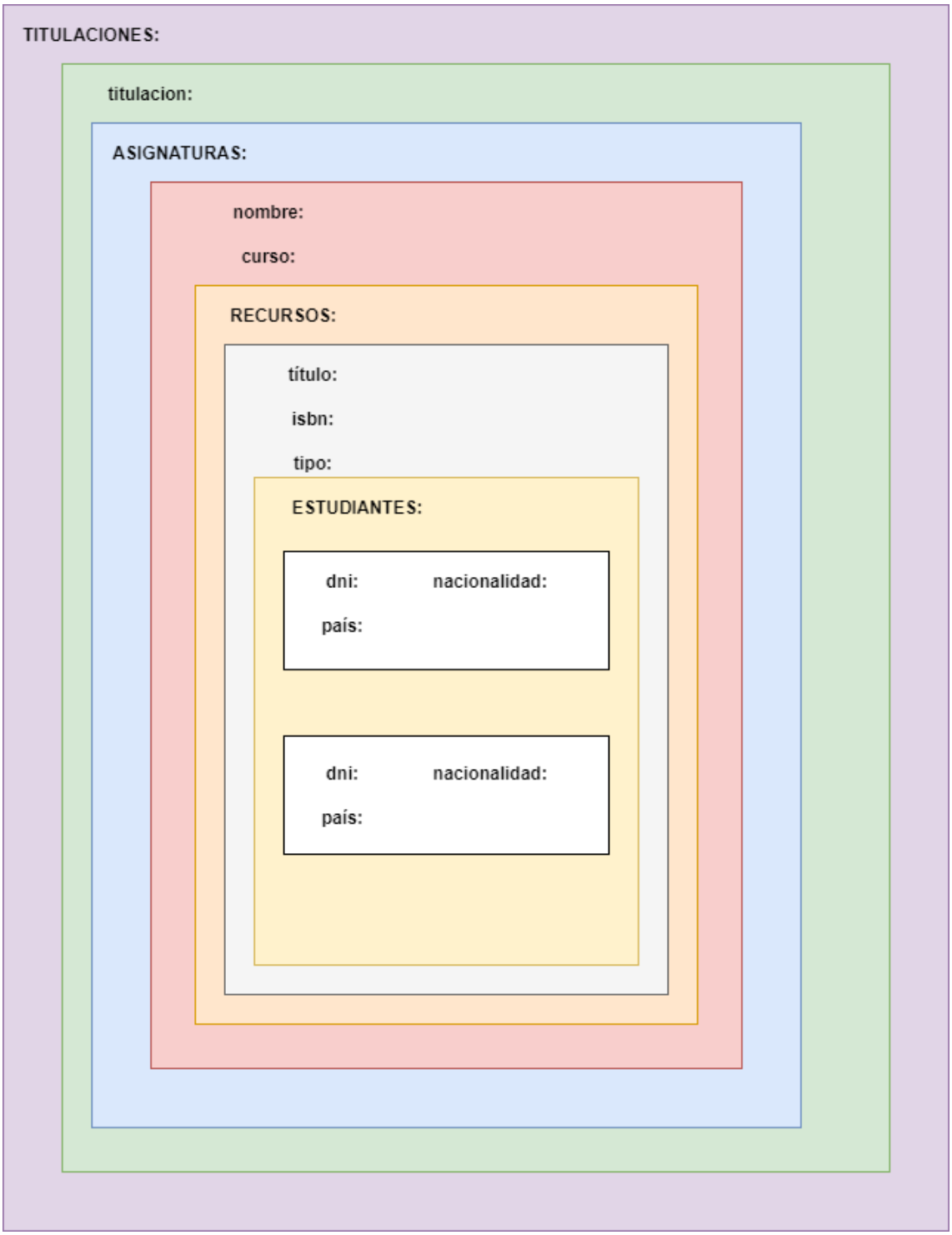
Con el objetivo de simplificar el diagrama se han eliminado otros agregados, es decir, dentro de “MESES” vamos a tener cada uno de los meses (en este caso solo se muestra uno), en “TITULACIONES” va a estar cada una de la titulaciones, en “ESTUDIANTES” vamos a tener cada uno de los estudiantes para cada titulación y mes, y en “RECURSOS” tenemos todos los recursos para cada estudiante bajo una titulación y mes.

1.3.2. Consulta 2

Esta consulta tiene como objetivo conocer la información de los recursos prestados agrupados por las asignaturas y titulación en las que están recomendados, es decir, para cada titulación y por cada asignatura se desea saber los datos de cada recurso prestado.

Teniendo esto claro, se ha definido un campo denominado “TITULACIONES” que contiene una colección de agregados para cada titulación (lo forman el campo “titulación” y “ASIGNATURAS”). “ASIGNATURAS” contiene una colección de agregados por cada una de las asignaturas que hay en cada titulación, cuyos atributos son “nombre”, “curso” y una nueva colección de recursos para cada asignatura y titulación denominada “RECURSOS”. Dentro de “RECURSOS” tenemos los atributos que queremos obtener, es decir, el título, el isbn y el tipo, y tenemos otro campo que es una colección denominado “ESTUDIANTES”, el cual contiene los estudiantes asociados para cada recurso en una determinada asignatura y en una determinada titulación. “ESTUDIANTES” contiene la información que se quiere obtener sobre los estudiantes, es decir, el dni, la nacionalidad y el país.

Se ha elegido esta estructura para poder satisfacer el objetivo de esta consulta, es decir, se busca obtener información de los recursos prestados a partir de las asignaturas y las titulaciones, por lo que encadenar las titulaciones con cada una de las asignaturas, y esto a su vez con los recursos asociados junto con los estudiantes que han solicitado dichos recursos, conseguimos obtener toda la información detallada.



Con el objetivo de simplificar el diagrama se han eliminado otros agregados, es decir, dentro de “TITULACIONES” vamos a tener cada una de las titulaciones de la universidad, en “ASIGNATURAS” van a estar cada una de las asignaturas asociadas a una titulación, en “RECURSOS” vamos a tener cada uno de los recursos para cada asignatura y titulación, y en “ESTUDIANTES” tenemos todos los estudiantes asociados a un determinado recurso de una determinada asignatura de una determinada titulación.

1.4. Ejercicio 4

Las limitaciones que presenta el modelo relacional fue una de las razones que hicieron que surgieran las bases de datos NoSQL. Sin embargo, las limitaciones de los sistemas NoSQL también motivaron otras bases de datos denominadas NewSQL, que son sistemas que adoptan el modelo relacional para ofrecer algunas de sus ventajas junto con algunas de las mejoras que proporcionan las bases de datos NoSQL. Para saber más de las bases de datos NewSQL se propone leer los apartados 1, 2 y 3 del artículo titulado "What's Really New with NewSQL?" y los apartados 1, 3 y 4 del artículo titulado "NewSQL Through the Looking Glass".

Una vez leídos los artículos de referencia, se propone buscar un caso de aplicación de una base de datos NoSQL y otro caso de aplicación de una base de datos NewSQL. A continuación, contesta a las siguientes preguntas por cada caso de aplicación (1 página como máximo para cada caso):

1. *Indica el enlace al caso analizado.*
2. *Describe el problema de persistencia de datos que se ha resuelto.*
3. *Justifica las razones por las que es recomendable la base de datos que se ha utilizado como solución.*
4. *Justifica las razones por las que no sería recomendable utilizar otro tipo de bases de datos diferentes a la utilizada en la solución.*
5. *Indica las referencias extra utilizadas para desarrollar el ejercicio.*

Las referencias extra se han documentado a lo largo del ejercicio y se puede acceder a ellas en el apartado "Bibliografía".

1.4.1. Caso de Salesforce a través de Streamsets

Enlace del caso analizado

En este caso vamos a analizar un modelo en grafo que se ha usado para obtener y almacenar los datos de los tickets (casos) que se producen en *Salesforce Service Cloud* [1], con el fin de hacer análisis posteriores.

Cabe destacar que este caso no es de *Salesforce* directamente, sino que una compañía intermedia *Streamsets*, la cual se encarga de la integración continua de los datos, hace uso de *Salesforce* y la tecnología *Neo4j* (software de base de datos orientado a grafos) [2].

El enlace de este caso es el siguiente, también se puede consultar en el apartado de bibliografía:

https://streamsets.com/blog/visualizing-analyzing-salesforce-data-neo4j/

Descripción del problema de persistencia de datos

El problema que se plantea es analizar los tickets o casos que se generan al hacer uso de *Salesforce Service Cloud*, es decir, hacer un análisis continuo de los datos que se generan al hacer uso de la nube para así mejorar la experiencia al usuario, identificar qué tickets están repetidos, cuáles son los más importantes, los que se deben de analizar antes… Cuando hablamos de tickets nos estamos refiriendo a las incidencias que se producen al hacer uso de un sistema, en nuestro caso de *Salesforce Service Cloud*.

Tal y como podemos ver, este caso tiene muy presente tanto los datos como las relaciones en sí, es por ello que hacer uso de una base de datos de grafos nos permite representar y analizar mejor los mismos.

Justifica por lo que es recomendable la base de datos usada

En primer lugar, es recomendable hacer uso de una base de datos NoSQL frente a una relacional ya que el volumen de datos crece muy rápido, escala mejor horizontalmente (es más fácil añadir clústeres y distribuir los datos), no se necesita de una gran cantidad de recursos para hacer uso de la base de datos.

En segundo lugar, hacer uso de una base de datos en grafo respecto a modelos de agregación se debe a que es más fácil realizar un análisis en un grafo que en un documento o una tabla, ya que nos permite a nivel visual encontrar patrones. Además, estos datos están muy interconectados, si hacemos uso de una base de datos clave-valor/orientada a documentos no podemos llegar al potencial de análisis que obtenemos con un modelo en grafos.

Justifica por lo que no es recomendable otras bases de datos

Este punto está altamente relacionado con el anterior, si hacemos uso de una base de datos relacional ganamos consistencia de los datos pero no disponibilidad, además, es más difícil escalar horizontalmente y a mayor volumen de datos más costes vamos a tener debido a la escalabilidad vertical.

Por otro lado, si hacemos uso de una base de datos clave-valor o orientada a documentos estaría muy bien ya que seguiríamos manteniendo los datos, pero al estar éstos interrelacionados lo mejor es una base de datos en grafo, ya que podemos ver a nivel visual como cada clave está relacionada con uno o varios nodos, permitiéndonos así ver de forma fácil patrones en los datos, es decir, ver qué incidencias están relacionadas, por qué hay una relación en dichas incidencias, si hay una incidencia única o hay un conjunto de ellas…

1.4.2. Caso de Kellogg

Enlace del caso analizado

El caso analizado en este caso es el de la compañía de cereales *Kellogg* [3], su objetivo era reducir los procesos ETL (extraer, transformar y cargar los datos) y la explotación de los datos.

El enlace al caso es el siguiente:

https://www.singlestore.com/blog/kellogg-case-study/

Descripción del problema de persistencia de datos

La compañía de cereales presentaba un gran problema con los datos que tenía y con los que generaba, haciendo que los procesos de ETL y explotación de datos fueran muy lentos.

Para solucionar este problema *Kellogg* hace uso de *AWS* (la nube de Amazon) y de *SingleStore* (anteriormente conocida como *MemSQL*, es una base de datos del tipo *NewSQL*, ya que es relacional, distribuida y permite la escalabilidad horizontal). Haciendo uso de estas tecnologías, consigue reducir todo el proceso de horas a minutos, además, al hacer uso de *SingleStore* consigue que la explotación de datos sea muy rápida para mejorar la toma de decisiones.

Justifica por lo que es recomendable la base de datos usada

Lo bueno de hacer uso de una base de datos *NewSQL* es que consigues tener los mejor de ambos mundos, es decir, al ser relacional permite la consistencia de los datos (ACID – atomicidad, consistencia, aislamiento y durabilidad), al ser distribuida y permitir la escalabilidad horizontal se consigue también la disponibilidad de los mismos, de esta forma estamos obteniendo unos datos que son seguros y los podemos obtener rápida y eficiente, por lo que se consigue mejorar la toma de decisiones de la empresa.

Justifica por lo que no es recomendable otras bases de datos

No es recomendable en este caso hacer uso de una base de datos relacional clásica, porque si se quiere obtener los datos de una forma rápida necesitamos la disponibilidad de los mismos, eso se consigue con la distribución de los datos (ventaja que se tiene en las bases de datos NoSQL).

Por otro lado, no es recomendable hacer uso de una base de datos NoSQL, ya que se quiere tener una consistencia de los datos, es decir, garantizar las operaciones ACID y esto lo conseguimos con las base de datos relaciones.

2. Bibliografía

[1] «Visualizing and Analyzing Salesforce Data with Neo4j», *StreamSets*, may 16, 2017. https://streamsets.com/blog/visualizing-analyzing-salesforce-data-neo4j/ (accedido oct. 12, 2021).

[2] «Neo4j», *Wikipedia, la enciclopedia libre*. ago. 06, 2019. Accedido: oct. 12, 2021. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Neo4j&oldid=118034594

[3] K. White, «How Kellogg Reduced 24-Hour ETL to Minutes and Boosted BI Speed by 20x». https://www.singlestore.com/blog/kellogg-case-study/ (accedido oct. 12, 2021).