Universidad Técnica Federico Santa María
Campus Santiago San Joaquín
Bases de datos avanzadas
Segundo Semestre 2024

Tarea 3: Cassandra

Profesor: Jose Luis Martí Lara
Ayudante: Daniela Sánchez Nizza

1. Introducción

Las bases de datos columnares corresponden a un tipo de base de datos NOSQL que gestionan los datos de manera diferente, almacenándolos en columnas en lugar de filas. Este tipo de bases de datos está optimizado para manejar grandes volúmenes de datos en grupos de columnas, permitiendo leer solo aquellas necesarias en una consulta, reduciendo la cantidad de datos a procesar.

Para esta tarea se utilizará Cassandra como gestor de base de datos.

2. Set Up

Para empezar, es necesario instalar Nodejs y Docker desktop. Una vez configurados, en el buscador de docker, sección images buscar "cassandra" y luego hacer pull al primer resultado. Después, abrir una terminal y colocar `docker run -name cassandra -d -p 9042:9042 cassandra`, para correrlo. Una vez realizado lo anterior, es posible configurar la conexión desde javascript. En la carpeta donde se realizará la tarea, es necesario correr por consola `npm install cassandra-driver`.
Nota: es posible realizar instalación sin docker, descargando Cassandra de la página oficial pero a mi me funcionaron los pasos anteriores, busquen el método que mas les acomode.

3. Descripción del caso

LibrePost es la empresa lider de comercio de estampillas en línea, debido a su crecimiento exponencial en los ultimos años, comenzó a tener problemas con su base de datos relacional y sus CEO'S, Claudio y Claudia, requieren que los ingenieros de la UTFSM los ayuden diseñando una base de datos columnar usando Cassandra. Ambos son abogados, por lo que no están familiarizados con el modelado de las bases de datos ni con la creación de consultas. Requieren una aplicación simple en Javascript con conexión a Cassandra que permita realizar diversos requerimientos. Deben implementar un menú que realice cada requerimiento al seleccionar la opción determinada, no debe realizarse una página web.

Para esta tarea, se cuenta con la siguiente información:

1. Dado que LibrePost tiene una amplia variedad de estampillas, el modelo de datos debe ser lo suficientemente flexible para acomodar información diversa sobre las estampillas:

   a) **título**: El nombre de la estampilla

   b) **country**: El país de emisión de la estampilla.

   c) **year**: Indica el año en el que la estampilla fue emitida.

   d) **serie**: Define la serie a la que pertenece la estampilla (para efectos de esta tarea será un grupo de estampillas que tienen un tema en común y fueron puestas en circulación al mismo tiempo).

e) **stamp_id**: Es el identificador único de cada estampilla.

f) **design**: Contiene la descripción visual o textual del diseño de la estampilla.

g) **face_value**: Representa el valor nominal de la estampilla al momento de su emisión, en dólares (float).

h) **condition**: Indica el estado físico de la estampilla, como "nuevo", "usado", o "dañado".

i) **status**: Describe el estado actual de la estampilla, por ejemplo, "disponible", "vendido", o "reservado".

j) **seller**: Almacena el nombre del usuario que actualmente posee la estampilla (único).

k) **transaction_history**: Se debe registrar el historial de transacciones de la estampilla, en pares fecha y acción (por ejemplo, venta, intercambio, etc.).

l) **tags**: Es un conjunto de etiquetas asociadas a la estampilla, descriptores como por ejemplo "rara","naturaleza","segunda guerra mundial", etc. (similares a hashtags en instagram).

m) **time_value**: Es un conjunto de valores por fecha de la estampilla, son pares fecha, valor.

2. A continuación se presenta un conjunto de 4 consultas, sólo para ellas, se requiere el esquema de navegación de las consultas, el modelo de datos lógico y el modelo de datos físico (presentación 3.3 Bases de datos de columnas diapositivas 24-31) (20 puntos, deben realizar todos los pasos):

   a) Buscar estampillas por country y year. Filtrando por tags.

   b) Encontrar la estampilla más cara por condición y año.

   c) Consultar el historial de transacciones de una estampilla específica.

   d) Verificar las estampillas de un vendedor específico.

3. En esta prestigiosa plataforma, la mayor parte del tiempo los usuarios realizan búsquedas por diversos filtros. Debido a la cantidad de registros, el año y el país son solicitados siempre. El buscador debe estar diseñado para ingresar un año determinado, país, y, opcionalmente el usuario puede buscar por tags (hasta 3) y por condition. La búsqueda debe retornar título, stamp_id, status y time_value (10 puntos).

4. Para reflejar la apreciación o depreciación de las estampillas, se almacena un conjunto de pares fecha-monto (time_value). Para una estampilla, el conjunto debe poder actualizarse cuando sea necesario, añadiendo un nuevo registro, también un elemento determinado puede ser modificado (10 puntos).

5. La plataforma permite la compra y venta. Se requiere simular la compra de una estampilla, siempre y cuando esté disponible. El proceso de compra debe ser el siguiente: Se debe ingresar el título y seller_id de aquella que se desea comprar, verificar que efectivamente está disponible, confirmar la compra y por último, cambiar el status de la estampilla a vendido, agregar el registro a transaction_history y cambiar el seller_id al valor de "D". (Los 3 ultimos cambios descritos deben realizarse usando BATCH, realizar cada uno por separado tendrá puntaje máximo 5). (15 puntos)

6. Debido a la economía actual, se tiene la creencia de que realizar una tabla en específico para el siguiente requerimiento es un despropósito, por lo que debe realizar una VISTA MATERIALIZADA para lo siguiente: Para efectos de reportería, se requiere saber cual es la estampilla mas cara para cada valor del campo "condition" (nuevo, usado, dañado) para un año específico. (10 puntos)

7. Debe ser posible agregar una estampilla en la base de datos. (5 puntos) (si el nombre es creativo, +1 punto de bonus).

8. El directorio ha descubierto que Cassandra tiene algunas dificultades para buscar rangos de valores, pero sabe de la existencia de los índices SASI e índices SAI, curiosos por el funcionamiento de ellos, se les solicita buscar la estampilla mas barata para cada uno de los status dado un rango de años sin iterar manualmente en cada año. usen uno de los dos tipos de indices (15 puntos).

9. Realizar muchas tablas en Cassandra dependiendo del patrón de acceso a los datos puede llevar a pérdida de consistencia de los datos, describan 2 técnicas para garantizar la consistencia entre tablas. (10 puntos)

4. Sobre la entrega

1. La tarea se debe realizar en parejas utilizando el software mencionado. Cualquier problema con su compañero debe informarle a su profesor.

2. Es necesario entregar un informe en pdf que contenga las respuestas a cada punto y tablas definidas con capturas que muestren el proceso, explicando cada paso brevemente. El informe debe tener nombre, rol y paralelo de los integrantes, puede ser realizado en word o latex.

3. El punto 2 no puede ser realizado con vistas desde una gran tabla, debe realizarse como el ejemplo de la diapositiva.(presentación 3.3 Bases de datos de columnas diapositivas 24-31)

4. Una estampilla será vendida según el precio que indique time_value en su registro mas actual.

5. Para tener todo el puntaje es necesario que se incluyan imagenes de los pasos que se realizaron y el resultado de estos como prueba, si la cantidad de registros retornados es alta basta con colocar una captura que muestre unos cuantos.

6. Deben entregar además los archivos necesarios para correr la tarea junto con un readme que tenga nombre, rol y consideraciones en caso de ser necesarias.

7. Para probar su trabajo, añadan unos cuantos registros a la base de datos para que se vea reflejado el correcto funcionamiento de cada punto. No es necesario añadir una gran cantidad de registros solo los suficientes. Habrá un descuento de 10 puntos si una funcionalidad no tiene suficientes registros para comprobar su funcionalidad. Por ejemplo, en un hipotético caso de que se solicite un top 10, no es lógico añadir 10 registros porque no hay registros sin seleccionar, en cambio, es mejor añadir al menos 11, para ver que efectivamente se dejan fuera elementos.

8. Debido al fin de semestre, se sugiere empezar con tiempo, no es posible dar plazo extra.

9. La fecha de entrega es hasta las **23:59 del lunes 11 de Noviembre de 2024**. Consultas sobre la tarea pueden ser realizadas a mi correo: daniela.sanchezn@usm.cl, no se responderán consultas 24 horas antes de la fecha de entrega. Errores en el formato del documento tendrán un descuento de 5 puntos, se descontarán 10 puntos por hora de atraso o fracción.