Examen

Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Bases de Datos II (IC 4302)
Primer Semestre 2023

Edgar André Araya Vargas **2020142856** **10/06/2023**

Instrucciones:

- Conteste todas las preguntas con el nivel mínimo y suficiente de detalle para demostrar su conocimiento del tema.
- No se evaluarán respuestas parciales o imprecisas.
- Es responsabilidad del estudiante garantizar que sus respuestas se entiendan, puede usar recursos como imágenes, diagramas, videos, etc.
- Si las respuestas no se entienden el profesor está en derecho de calificar con un 0 la respuesta.
- La nota máxima del examen es 100.
- El tiempo estimado para completar el examen en una clase presencial es de 120 minutos.
- El examen deberá ser entregado antes de las **05:00 pm del día 10 de junio del 2023**. El estudiante cuenta con más de 12 horas para elaborar el examen. La fecha en la cual se entregaría el examen fue notificada con más de 15 días de antelación y acordada con todo el grupo.
- Si el examen es entregado después de esta hora, no será revisado y se obtendrá una nota de 0.
- El examen deberá ser entregado al correo electrónico del profesor, debe seguir el formato especificado en el programa del curso.
- El nombre del archivo debe ser **ex.pdf**
- Cualquier indicio de copia será calificado con una nota de 0 y será procesado de acuerdo con el reglamento, esto incluye cualquier herramienta que genere textos mediante inteligencia artificial y cualquier producción parcial o total de algún documento sin su debido reconocimiento al autor o autora.
- Se puede utilizar cualquier recurso en Internet para elaborar sus respuestas, deben especificar referencias bibliográficas, se debe validar que sea una fuente confiable, herramientas de inteligencia artificial no se consideran una fuente confiable.
- Si la referencia bibliográfica NO es confiable el profesor está en derecho de calificar la respuesta con una nota de 0. Para verificar si la referencia el confiable, puede hacer las siguientes preguntas:
    - ¿Quién es el autor o autora?
    - ¿Cuál es el propósito de ese documento?
    - ¿Es posible que esté parcializado?
    - ¿Ha sido revisado o aprobado por expertos en ese campo de estudio?
- Las preguntas fuera del horario de clase se pueden hacer por medio de correo electrónico o al grupo oficial de Telegram, pueden darse retrasos en las respuestas a las preguntas, en especial las que se realizan a altas horas de la noche o madrugada, se recomienda realizar todas las consultas necesarias durante la clase del 9 de junio del 2023.
- El examen consta de 4 preguntas de desarrollo.

- **Es importante recalcar que las preguntas son de desarrollo, cada respuesta debe estar cuidadosamente desarrollada con explicaciones adecuadas.**
- El valor del examen es de un 10%.
- Es responsabilidad del estudiante completar todas las preguntas del examen, en caso de que se olvide responder alguna de ellas se obtendrá una nota de 0.

Pregunta 1 (60 pts)

Aproximadamente para el año 23651 de nuestra era y durante el apogeo del imperio galáctico, el matemático Hari Seldon ha desarrollado su teoría llamada Psicohistoria, mediante esta, ha podido predecir con un grado de confianza bastante alto la caída de la civilización seguida de un periodo de barbarie, con el fin de reducir este periodo de barbarie, este ha desarrollado un plan y como parte de este, se encuentra la conformación de la Enciclopedia Galáctica, la cual de acuerdo con el divulgador científico Carl Sagan es un sugerente proyecto del saber colectivo de las civilizaciones avanzadas del universo.

Usted ha sido escogido como líder técnico del equipo que se encargará de implementar la base de datos que mantendrá esta información con alta disponibilidad y con un mecanismo adecuado para navegar los datos y realizar búsquedas. Es importante mencionar:

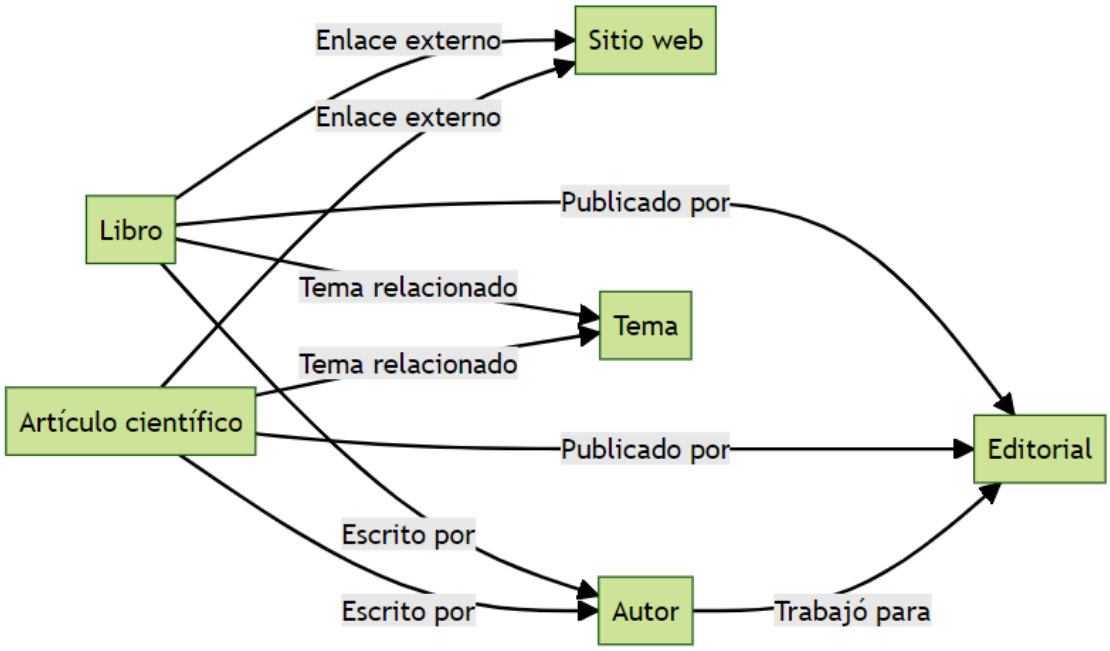
- La tecnología en bases de datos SQL y NoSQL no han cambiado desde el año 2023.
- No existe restricción en cuanto a dinero que se puede invertir en el proyecto.
- Los proveedores de Cloud siguen existiendo y ahora han expandido sus ubicaciones en prácticamente todo el universo conocido.
- Los productos ofrecidos en los proveedores de Cloud para el 2023 siguen siendo ofrecidos para el año 23651.
- Se tiene que permitir full text search sobre la información en la Enciclopedia Galáctica.
- Se tienen que establecer relaciones entre los diferentes elementos de información de forma tal que permita descubrir relaciones entre la información. Un excelente ejemplo de cómo funcionara la navegación es el sitio de Wikipedia.
- La Enciclopedia Galáctica presenta un alto número de lecturas contra un bajo número de escrituras (prácticamente 0).
- Para el año 23651, se han escrito:
    - 4 billones de libros con una media de 200 páginas.
    - 1 billón de artículos científicos con una media de 10 páginas.
    - 20 billones de sitios web con una media de 10 páginas cada uno.

En su calidad de líder técnico, usted debe presentar una propuesta para dar respuesta a las siguientes preguntas:

1. ¿Qué motor de base de datos utilizaría para implementar la navegación entre distintos elementos de información? ¿Es necesario que este motor de base de datos contenga todo el elemento de información o solo palabras clave que permitan establecer relaciones? Justifique su respuesta

mediante la elaboración de un pequeño modelo de datos y las relaciones que establecería entre los diferentes elementos de información, lo más importante es garantizar una navegación y que permita descubrir relaciones. (20 pts)

Para el caso específico de la Enciclopedia Galáctica de Hari Seldon se podría utilizar un motor de base de datos de grafos, específicamente el servicio de Microsoft Azure Cosmos DB. Mediante este motor podemos representar la información recopilada como nodos y las relaciones que existen entre ellos como arcos. En el caso del ejemplo y pensando en esta base presentada como única base de datos para el proyecto es necesario que Azure Cosmos DB contenga todo el elemento de información dentro de sus nodos para poder acceder a sus detalles relevantes. Al mismo tiempo es de extrema importancia que se incluyan claves o metadatos para generar relaciones entre ellas. Ahora, si tomamos en consideración la siguiente pregunta, y consideramos que los datos de los elementos de información se encuentran almacenados dentro de MongoDB, podemos solo generar la información de relaciones pertinente mediante metadatos y palabras clave, sin tener todo el documento guardado en la base.



2. ¿Qué motor de base de datos utilizaría para almacenar los elementos de información y garantizar full text search? Justifique su respuesta comentando: (20 pts)

La base de datos NoSQL de MondoDB Atlas sería ideal para almacenar los datos y garantizar el full text search.

   a. Capacidad del motor para implementar full text search.
   Mongo DB Atlas cuenta con la compatibilidad con búsqueda de texto completo utilizando el servicio de Atlas Full-Text Search. El motor permite realizar búsquedas especificas con análisis de texto y consultas avanzados. Con esta capacidad se pueden buscar términos específicos dentro de la gran librería de información de la Enciclopedia Galáctica.

   b. Particionamiento o sharding de datos.
   La base de datos de Mongo DB igual que otras bases de datos no relacionales maneja la partición y distribución de datos mediante sharding. Esto permite distribuir en diversos servidores la información, lo cual es de gran utilizada para no perder la valiosa información de la psicohistoria para predecir la caída de la civilización. Con la gran cantidad de lecturas con el sharding de Mongo podemos reducir su carga de consultas y

garantizar alta disponibilidad.

c. Representación de elementos de información en la base de datos (tablas, documentos, collections, etc.)

Como se sabe, Mongo DB utiliza una representación de JSON o BSON para la información almacenada, usualmente organizados dentro de colecciones donde se pueden guardar millones de documentos. Cada diferente elemento de información podría tener su propia colección en Mongo, "Libros", "Sitios Web… Los documentos podrían tener campos estructurados lo que facilita sus relaciones o atributos. Todo depende de como requiera su arquitectura de software Hari Seldon.

3. Describa la forma en la cual combinaría los dos motores anteriores (navegación y full text search) para crear un sistema simple de búsqueda y navegación de información similar al que tiene el sitio Wikipedia donde se busca un elemento de información y nos podemos mover entre términos. (5 pts)

Para desarrollar la Enciclopedia galáctica de manera similar a Wikipedia podemos almacenar los elementos de información dentro de Mongo DB. En esta plataforma hacemos la indexación de campos relevantes para full text search. Luego en Azure Cosmos almacenamos metadatos relacionados con los libros, sitios web, y artículos científicos. Es decir por cada elemento de información tendremos un JSON con los metadatos del elemento e incluso referencias cruzadas con otros elementos de información. Estos documentos JSON de metadatos se pueden almacenar en las mencionadas colecciones especificas en Azure Cosmos y relacionar con otros elementos de información de MongoDB mediante sus identificadores únicos. La plataforma funcionaria básicamente de esta manera: Un usuario realiza una búsqueda, y utilizamos el motor de búsqueda de texto completo de Mongo DB para realizar la búsqueda en los campos indexados. Se presentan los resultados y al usuario seleccionar una opción se utilizan los metadatos almacenados en Azure Cosmos para mostrar info adicional y enlaces internos para permitir fácil acceso a otros elementos de información dentro de la plataforma de Enciclopedia galáctica.

4. ¿De qué forma garantizaría alta disponibilidad de las bases de datos? (5 pts)

Es importante mantener datos tan importantes disponibles. Hay diferentes maneras de garantizar alta disponibilidad pero tenemos que asegurarnos que sea funcional para ambas bases. La réplica de datos ofrece gran disponibilidad y esta disponible para MongoDB y Azure Cosmos. Mediante la replicación de datos podemos asegurarnos que si un nodo falla los demás mantienen la replica del dato. La replica de datos también puede tener una amplia distribución geográfica para así garantizar alta disponibilidad e incluso mejor rendimiento.

5. ¿Cómo podría garantizar que las búsquedas siempre tengan un tiempo de respuesta constante? (5 pts)

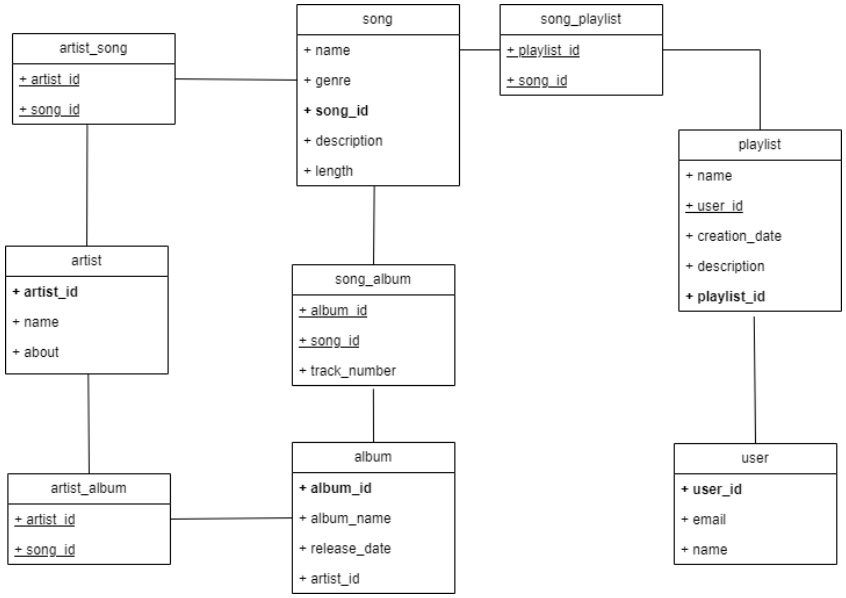
Para intentar garantizar un tiempo de respuesta constante podemos aplicar diversos cambios o técnicas en nuestras bases de datos. Recordemos la correcta indexación en Mongo DB nos llevara a un ahorro de tiempo. Un escalado horizontal de múltiples nodos como se menciono anteriormente puede hacer que las consultas se ejecuten en paralelo y así mejorar el tiempo de respuesta.

6. ¿Cómo el uso de caches y localidad podría mejorar el rendimiento del sistema? (5 pts)

Mediante el uso de caché se podrían almacenar las consultas más frecuentes para reproducir más fácilmente un resultado. También se pueden almacenar los índices en el cache para no necesitar su búsqueda en la nube o disco. Finalmente, la localidad de datos

Pregunta 2 (10 pts)

El siguiente diagrama representa una versión simplificada de un sistema de reproducción de música que utiliza una base de datos relacional:



Este sistema tiene varios vicios o problemas de normalización, así como el grave problema de que no tiene definidos índices, en conjunto esto ha causado que se esté experimentando muchos timeouts y la solución convencional de agregar más hardware se ha vuelto insostenible. Luego de un estudio del workload de la base de datos, se llegó a las siguientes conclusiones:

- Es necesario definir algunos índices fuera de los que son definidos automáticamente mediante llaves primarias y foráneas.
- Un motor de base de datos relacional no parece ser el más adecuado para el problema.
- El patrón de uso es muchas lecturas contra pocas escrituras.

Mediante los logs de acceso y los logs de slow queries, se ha encontrado que los siguientes queries son los más usuales y problemáticos en tiempo que tardan en ejecutarse:

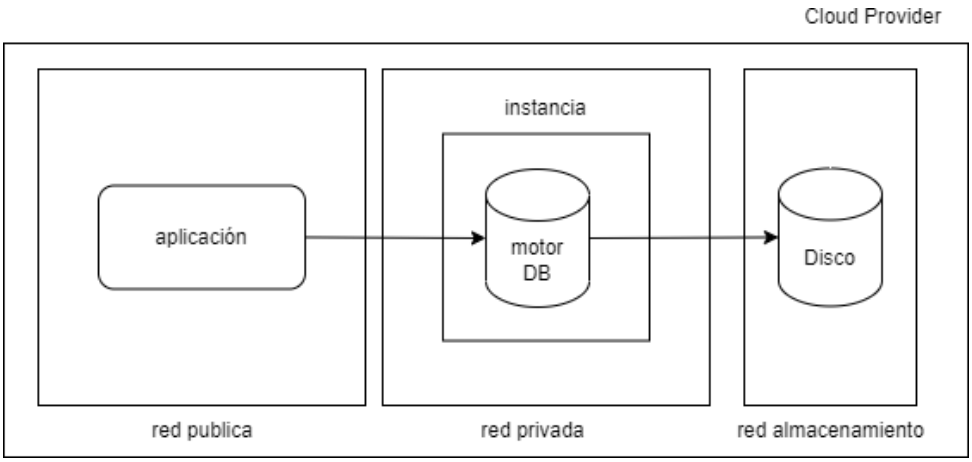
SELECT name FROM artist WHERE name like '%{text}%'
SELECT name FROM album WHERE name like '%{text}%'
SELECT a.name as artist_name, al.name as album_name, s.name, s.genre, sa.track_number, s.length, s.description FROM artist a INNER JOIN artist_song as ON a.artist_id = as.artist_id INNER JOIN song s ON as.song_id = s.song_id INNER JOIN song_album sa ON s_song_id = sa.song_id INNER JOIN album al ON sa.album_id = al_album_id WHERE a.name = '%{name}%' AND al.name = '%{name}%' and s.name = '%{name}%'

Como administrador o administradora de bases de datos, elabore respuestas a las siguientes preguntas:

- ¿Qué índices definiría para aumentar la velocidad de todo el sistema? Tome en cuenta todos los tipos de índices estudiados en el curso. (3 pts)
  Intentaría solucionar desarrollando un índice de texto completo en la columna "name" de las tablas "artista" y "álbum". Mediante este índice podemos mejorar el rendimiento al utilizar algoritmos que procesan el lenguaje y descomponen el texto en términos y palabras significativas, así haciendo más fácil buscar rápidamente los nombres de álbumes y artistas. Se pueden desarrollar índices compuestos para mejorar la velocidad de las consultas. En la tabla "artista_song" índice compuesto entre "artista_id" y "song_id", en la tabla "song_album" un índice compuesto de "song_id" y "álbum id", finalmente en la tabla "song_playlist" un índice compuesto de "playlist_id" y "song_id".
- ¿Qué base de datos SQL o NoSQL recomendaría para reemplazar la base de datos actual? Justifique su respuesta. (3 pts).
  Una base de datos funcional para este ejemplo podría ser nuevamente la base de datos no relacional de Mongo DB. Este base ayudaría a mejorar el rendimiento al ignorar problemas de la base actual como problemas de normalización, falta de índices y mala normalización. MongoDB permite una gran flexibilidad y no limita la información a tablas.
- ¿Existirá alguna otra forma de mejorar el rendimiento de la base de datos relacional en especial para la tercera consulta? Comente. (4 pts)
  Además de la solución mencionada anteriormente simplemente pensaría en una reestructuración de la base de datos, es decir, desnormalizar aun mas la base para duplicar ciertos datos en ciertas tablas para evitar la necesidad de múltiples por operaciones por consulta así acelerando la consulta en general.

Pregunta 3 (20 pts)

En los últimos 15 años, la forma en la cual se mantienen e instalan servidores de bases de datos ha cambiado considerablemente, la aparición del Cloud ha proporcionado muchas ventajas para la instalación y mantenimiento, pero ha inducido nuevos problemas de seguridad y nuevas soluciones, en el siguiente diagrama se muestra una arquitectura típica de una base de datos en un Cloud Provider



Tomando como referencia el diagrama anterior, ¿Cuáles son las buenas prácticas en términos de seguridad que se deben seguir cuando se instala un motor de base de datos en el Cloud? Fundamente su respuesta hablando de la seguridad de cada uno de los componentes que se exponen en el diagrama.

Inicialmente me parece importante generar reglas de firewall para controlar el tráfico de red entrante y saliente tanto en la red publica como en la red privada. Mediante firewalls en la red pública podemos restringir el acceso a la aplicación por dirección IP no deseadas y filtrar el trafico entrante a la aplicación para que funcione correctamente (permitir tráfico en puertos necesarios y bloquear en puertos no utilizados). También con Firewall en la red privada nos permite establecer políticas de seguridad para proteger los datos sensibles en el motor DB. Es importante recordar que se puede configurar una autenticación para acceder específicamente a la instancia del motor DB y darle otra capa de seguridad incluso a la instancia. Para añadir incluso otra capa de seguridad podemos limitar los privilegios de los usuarios dentro de la base, al darles roles específicos. Finalmente en la red de almacenamiento podemos utilizar encriptación y copias de seguridad para tener en un lugar seguro los datos almacenados. Tenemos seguridad de acceso hacia la información en el disco y podemos generar seguridad en la integridad de los datos.

Pregunta 4 (10 pts)

La Observabilidad es una gran herramienta que nos permite tener una visión en el tiempo de la forma en la cual se comportan sistemas computacionales, estos sistemas hacen uso extensivo de bases de datos de series de tiempo, una de las más utilizadas es Prometheus, pero existen soluciones que utilizan otras bases de datos o motores de búsqueda como Elasticsearch u OpenSearch. Como ingeniera o ingeniero a cargo de los sistemas de Observabilidad de una empresa, se le ha solicitado dar respuesta a las siguientes preguntas, con el fin de determinar la estrategia que seguirá la empresa en términos de Observabilidad en los siguientes años.

- ¿Por qué las bases de datos de series de tiempo son tan utilizadas en soluciones de Observabilidad? Realice un análisis desde el punto de vista de la naturaleza de los datos que se

recolectan. (2 pts)

Las bases de datos de series de tiempo están diseñadas para almacenar y gestionar los datos permitiendo consultas y análisis basado en **intervalos de tiempos específicos.** Esto es de extrema funcionalidad para la observabilidad ya que los datos se generan y registran a lo largo del tiempo por lo que necesitan su respectivo timestamp que será utilizado en la base.

- ¿Es posible utilizar BigTable como una base de datos de series de tiempo que se pueda utilizar como parte de una solución de Observabilidad? Justifique su respuesta desde el punto de vista de la naturaleza de la base de datos. (2 pts)

  BigTable es una base de datos creada por Google que ofrece gran disponibilidad y rendimiento en consultas aleatorias y escrituras en grandes conjuntos. Esta PUEDE almacenar y consultar datos temporales pero no proporciona las funcionalidades de análisis y consulta de datos de series de tiempo. Básicamente, puede ser utilizado pero debido a la naturaleza de la base de datos no debería ser específicamente utilizado para observabilidad.

- Suponiendo que tenemos una solución de Observabilidad que utiliza Elasticsearch, ¿Cómo podemos ahorrar dinero con información histórica? (2 pts)

  Observabilidad significa métricas, logs y rastreos que son unificados y pueden ser visualizados en un solo lugar dándole mayor facilidad a desarrolladores para encontrar errores o fugas de dinero en servicios o secciones innecesarias. Elastic Observability ha demostrado brindar a las empresas 10 veces mejor rendimiento con un ahorro de costos del 75 % [4]. Al tener estas métricas, logs y rastreos podemos aplicar estrategias como la compresión de los datos de indexación [5].

- Comente las ventajas y las desventajas de utilizar un servicio de Observabilidad on-premise (por ejemplo, Prometheus y Grafana) vs un Managed Service (como Datadog), justifique su respuesta con la experiencia obtenida en la tarea corta 1 de este curso. (4 pts)

  Al utilizar observabilidad on-premise con Prometheus y Grafana en la tarea 1 del curso pudimos ver que hay un control total sobre la configuración y gestión de la observabilidad de las bases de datos. Esto es una ventaja ya que permite fácilmente adaptar nuestro código para adaptarse al sistema de monitoreo pero al mismo tiempo presenta una mayor complejidad y posibles problemas de escalabilidad. El managed service por otro lado, seria solo contratar el servicio sin decisiones sobre la implementación. Es mas sencillo, pero puede generar complicaciones si tenemos problemas con el servicio o queremos hacer configuraciones manuales.

Referencias.

- [1]https://www.mongodb.com/docs/
- [2]https://learn.microsoft.com/en-us/azure/cosmos-db/
- [3]https://static.googleusercontent.com/media/research.google.com/es//archive/bigtable-osdi06.pdf
- [4]https://www.elastic.co/es/what-is/observability
- [5]https://www.elastic.co/es/blog/elastic-observability-clusters-upgrade-latest-release-save-money