

## Prueba Corta # 3 y #4

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



Fecha de entrega: **11/04/23 antes de las 11:59 pm**

Forma de entrega: **Email al profesor siguiendo los lineamientos del programa de curso, adjuntando documento y link al repositorio.**

Formato: **Markdown**

Nombre Archivo: **pc34.md**

---

**Estudiante: Gerald Núñez Chavarría – Carné: 2021023226**

### 1. Comente de que forma un query como:

**SELECT col1, col2, ..., colN FROM table;  
N es menor al (número total de columnas /2).**

**Se puede ver beneficiado por un columnar storage (25 pts).**

**R/** En un almacenamiento de tipo columnar storage, las columnas de la tabla se almacenan de forma contigua en el disco, y cada columna contiene todos los valores para esa columna en todas las filas. Esto significa que cuando una consulta SQL selecciona un subconjunto de columnas, el motor de la base de datos solo tiene que leer las columnas seleccionadas y no todas las columnas de todas las filas, lo que puede reducir significativamente la cantidad de lecturas de disco necesarias.

Si el número de columnas seleccionadas en la consulta es menor que la mitad del número total de columnas de la tabla ( $N < \#columnas/2$ ), entonces es más probable que una gran proporción de las columnas restantes no se utilicen en la consulta. En un almacenamiento de filas, el motor de la base de datos aún tendría que leer todas estas columnas innecesarias de todas las filas para realizar la consulta, lo que resultaría en un mayor tiempo de procesamiento y mayor uso de recursos. Sin embargo, en un almacenamiento de columnas, estas columnas innecesarias no se leerán y se puede lograr una mejora significativa en el rendimiento.

### 2. Una base de datos MariaDB o MySQL se encuentra bajo mucha presión debido a un alto número de lecturas (heavy read), de qué forma se podría lograr bajar esta presión sobre la base de datos, justifique su respuesta (25 pts).

**R/** A continuación se mencionan tres técnicas que se pueden implementar para reducir la presión que se está ejerciendo en la base de datos, basándose en que esta presión es generada por un heavy read.

2.1 Una técnica muy utilizada en bases de datos es el caching. Se basa en almacenar los resultados de las consultas más frecuentes en memoria caché, para así tener un tiempo de respuesta más rápido y que no genera carga sobre la base de datos, ya que no se necesita acceder a esta, solucionando en gran medida el problema de recibir un volumen muy alto de lecturas.

2.2 El uso de índices bien configurados para poder aliviar la carga de operaciones de lectura en una base de datos es otra solución importante. Un índice bien configurado puede ayudar a que la búsqueda y recuperación de un dato sea más eficiente.

2.3 La tercera técnica es realizar un escalamiento horizontal en la base de datos, que consiste en agregar más nodos a la base de datos para distribuir la carga de trabajo, permitiendo que las operaciones de lectura no lleguen a un solo nodo sino a varios, ayudando a reducir la carga de trabajo.

3. Durante la semana santa 2023 un grupo de estudiantes de Bases de Datos 2 decidieron desarrollar un juego retro de la plataforma Atari llamado Breakout, el mismo en su versión original es de un sólo jugador, pero se toma la decisión de implementar una versión multi-jugador, el backend se encuentra almacenado en una base de datos NoSQL que se encuentra ubicada en Azure Cloud en una región en Estados Unidos, el juego se volvió muy popular en menos de una semana, tanto que una gran cantidad de usuarios en otros continentes se han estado quejando de retrasos en el juego lo cual causa una mala experiencia, gracias a las herramientas de observabilidad configuradas, se ha descubierto que la naturaleza del workload es pocas escrituras y muchas lecturas a los mismos datos, presente dos recomendaciones (con su debida explicación) desde el punto de vista de base de datos que daría para mejorar el rendimiento de la misma (50 pts).

R/ A continuación se presenta las dos soluciones explicadas:

**3.1** Se puede considerar utilizar replicación, con esto se puede mejorar el rendimiento de la base de datos NoSQL, ya que se va a distribuir la carga de trabajo entre los múltiples nodos o instancias de la base. Esto ayudaría a reducir los retrasos y mejorar la experiencia del usuario en otros continentes, ya que cada instancia de la base de datos estaría procesando una parte del workload y no estarían compitiendo por los mismos recursos. Además, la replicación permitiría mejorar la disponibilidad y la tolerancia a fallos, ya que, si una instancia falla, las demás pueden tomar el relevo y garantizar el servicio.

**3.2** Una técnica de caché es una solución, esto le va a permitir a la base de datos NoSQL almacenar los resultados de las consultas más solicitadas por el usuario en memoria caché. Esto permitiría reducir el número de lecturas a la base de datos y mejorar la latencia de respuesta, ya que los datos estarían disponibles en memoria y no tendrían que ser recuperados de la base de datos. Esto ayuda a solucionar el problema, ya que, por ejemplo, si un usuario en Europa solicita información sobre el puntaje más alto registrado en el juego, y ese puntaje ya está almacenado en caché en un servidor cercano a Europa, el servidor puede responder rápidamente con esa información, sin tener que hacer una consulta a la base de datos en la nube en Estados Unidos. Esto reduce el tiempo de respuesta y mejora la experiencia del usuario.