

Prueba Corta #5 y #6

Erick Madrigal Zavala 2018146983

1)

Un clustered index es un tipo de índice en una base de datos que ordena físicamente las filas de una tabla en el disco según los valores de una o más columnas. Esto significa que las filas con valores similares en las columnas del índice se almacenan juntas en el disco. Por otro lado, un índice non-clustered no ordena físicamente las filas de la tabla, sino que crea una estructura de datos separada que contiene una copia de las columnas del índice y un puntero a la fila correspondiente en la tabla. La principal diferencia entre un clustered index y un índice non-clustered que utiliza INCLUDE para agregar columnas al índice es que el primero ordena físicamente las filas de la tabla, mientras que el segundo no lo hace. Además, un índice non-clustered con INCLUDE puede incluir columnas adicionales en el índice que no están en la clave del índice, lo que puede mejorar el rendimiento de las consultas que utilizan estas columnas.

2)

El memory footprint es la cantidad de memoria principal que un programa utiliza o referencia mientras se ejecuta. En el contexto de la creación de índices, el memory footprint puede afectar el rendimiento de la base de datos, ya que los índices consumen memoria para almacenar la estructura de datos que se utiliza para acelerar las consultas. Si el memory footprint de los índices es demasiado alto, puede provocar que la base de datos tenga que paginar a disco con más frecuencia ya que esta no puede entrar por completo en la memoria principal, lo que puede ralentizar el rendimiento de las consultas.

3)

Ya que la compañía FASTantic no tiene estos 40 billones de registros almacenados en una base de datos se podría empezar por utilizar herramientas para poder migrar estos datos a una. Debido a la gran cantidad de registros una opción sería utilizar bases de datos distribuidas como MongoDB o Cassandra y las herramientas pueden ser mongoimport para mongo o copy command para Cassandra. Se escogen estas bases de datos ya que permiten la recuperación y almacenamiento de grandes cantidades de datos distribuidos en múltiples nodos y con esto podríamos aumentar la escalabilidad y la tolerancia a fallos.

Otra opción sería utilizar una tabla hash en la cual cada registro se almacene en una posición determinada en su clave construida por país, ciudad y fecha. Esto con el fin de poder tener una búsqueda mas rápida de los registros. Un inconveniente de esta implementación es que podría haber colisión de los datos