

**Facultad de Ciencias Exactas y Naturales Agrimensura**

**Catedra: Base de Datos I**

**Año: 2024**

**Informe:**

**Proyecto Integrador de Bases de Datos I**

Alumnos:

-Almada, Tomas Emanuel DNI: 44876943

-Gimeno, Daniel Eduardo DNI:37327946

-

-

**I – Introducción**

El presente documento tiene la finalidad de exponer problemas comunes de las bases datos y su solución mediante el uso de las buenas practicas, haciendo uso de conceptos teóricos y demostraciones practicas.

**Optimización de consultas a través de índices.**

En esta sección abordaremos el problema de rendimiento de las consultas SQL de tipo selección al momento de intentar recuperar información de las tablas contenidas en las bases de datos, asi mismo se buscaran soluciones que permitan mejorar dicho rendimiento mediante el uso de los índices.

Objetivos Generales:

Se espera que el lector una vez finalizada la lectura del presente documento, tenga la capacidad de comprender los problemas presentes en las bases de datos al momento de realizar las consultas para obtener información, producto de una mala gestión de la misma o uso inadecuado de los recursos provistos.

Objetivos Especificos:

Se espera que el lector una vez concluida la lectura del presente documento, adquiera la capacidad de identificar posibles problemas de rendimiento en las consultas SQL y pueda solucionarlas aplicando las buenas practicas que se presentaran en dicho documento.

**II – Marco Conceptual O Referencial**

**III – Metodología Utilizada**

La presente investigación de los temas aquí expuestos, fueron llevados a cabo utilizando una metodología de desarrollo (SCRUM).

1. **Descripción de cómo se realizó el trabajo practico:**

Durante la observación de diversas bases de datos utilizadas en diferentes ámbitos, se pudo observar el problema frecuente que presentaban al momento de obtener la información solicitada, una vez observado el problema en cuestión, se realizo un análisis de dichas bases de datos, donde se pudo observar que las mismas al momento de realizar las consultas no utilizaban índices para mejorar la búsqueda o realizaban uniones entre tables usando columnas que no eran candidatas a ser utilizadas para la indexación de los registros, produciendo esto una demora excesiva en la lectura y obtencion de los datos solicitados.

1. Herramientas

La información aquí recabada fue obtenida de fuentes de origen corporativo, siendo las mismas mediante entrevistas a DBA, donde se presenta el problema y la solución planteada propuesta por dicho profesional que permitió mitigar el problema mencionado previamente.

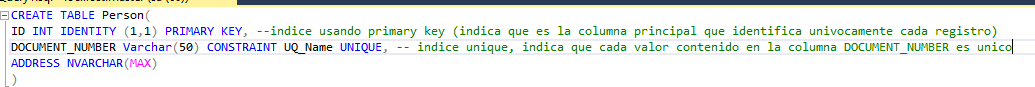
**IV – Desarrollo del tema**

A continuación, se exponen los diferentes tipos de índices que son utilizados por las bases de datos.

**Índice Único**

Un índice único es usado para mantener la integridad de la información de las columnas sobre las cuales es creado al asegurar que no hay valores duplicados en el índice clave, y la filas de la tabla, sobre la cual el índice es creado. Esto aseguro que la información será única basada en el índice clave, dependiendo de las características de la información que es almacenada en la columna del índice clave o la lista de columnas. Si el índice Único clave consiste en una columna, el SQL Server garantizará que cada valor en el índice clave es único. Por otro lado, el índice Único clave consiste en múltiples columnas, cada combinación de valores en ese índice clave debería ser único.

Se presenta un ejemplo de código utilizando este tipo de índices al momento de la creación de un tabla.



Índice filtrado

Un índice Filtrado es un índice No Agrupado optimizado, introducido en SQL Server 2008, que usa un filtro predicado para mejorar el rendimiento de las consultas que recuperan bien definidos subgrupos de filas de la tabla, al indexar la única porción de las filas de la tabla. El tamaño más pequeño del índice Filtrado, que consume una pequeña cantidad del espacio del disco, comparado con la tamaño de la tabla completa de índice, y mientras más precisas las estadísticas filtradas, que cubren las filas del índice filtrado con solo costo mínimo de mantenimiento, ayuda a mejorar el rendimiento de las consultas al generar una más óptima ejecución del plan.

Un ejemplo de buenos sub-grupos de información, que se pueden beneficiar de las ganancias del rendimiento del índice Filtrado, son las columnas Dispersas con un gran número de valores NULL.

Índice Espacial

Tipos de información especial de Geometría y Geografía fueron introducidos la primera vez en el SQL Server 2008. El tipo de información de **geometría** es usado para almacenar información de planificación geométrica como puntos, líneas y polígonos, donde el tipo de información **geográfica** es usado para representar geográficamente objetos en el área de la superficie de la Tierra, como las coordinadas de latitud y longitud de GPS.

Un índice Espacial es un tipo especial de índice, creado en las columnas que almacenan información espacial, para mejorar el rendimiento de las operaciones realizadas en las columnas espaciales, al reducir el número de objetos en los cuales costosamente las operaciones espaciales necesitan ser aplicadas. Toma en consideración que creando índices Espaciales requiere que la tabla tenga PRIMARY KEY agrupada.

**Índice Hash:**

Este es un tipo especial de índice usado en tablas de Memoria Optimizada, para acceder a la información a través de una tabla in-memory hash, consumiendo una cantidad fija de memoria, especificada por la cuenta.

**Índice de texto entero:**

Este es un tipo especial de índice con base funcional token, que es construido y mantenido por el Motor SQL Server Texto Entero, para mejorar el rendimiento de la búsqueda de cadena de caracteres de información.

**Índice de Almacenaje en columna:**

El índice de almacenaje en columna es una característica en la cual la información será físicamente organizada en un formato de información en columna, no como la tecnología tradicional de almacenamiento en fila, con el formato por fila, que es usado para almacenar, administrar y recuperar grande información usando un formato de información en columna. Un índice de almacenaje en columna funciona bien para cargas de trabajo de almacenaje de información que realiza pesos voluminosos y consultas de solo lectura, llegando a 7x de compresión de la información del tamaño de la información no comprimida.