Instituto Politécnico Nacional

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERIA Y TECNOLOGIAS AVANZADAS.

BASES DE DATOS DITRIBUIDAS

PRACTICA 2

Equipo 4:

Alan González Morales

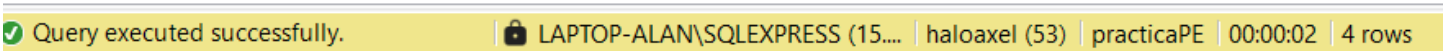
Axel Iván Rossano Medina

Erik Bravo Pérez

Docente: Carlos De La Cruz Sosa

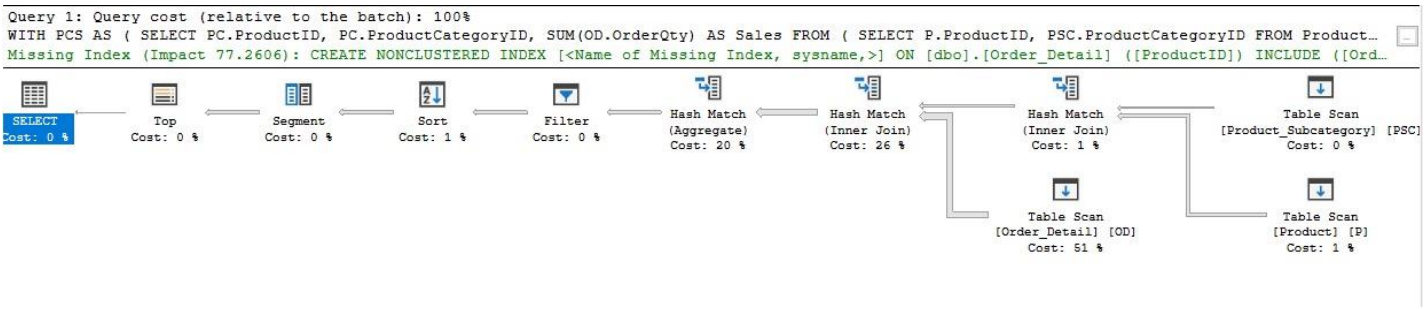
No. consulta	1
Responsable	Alan González Morales
Descripción	Listar el producto más vendido de cada una de las categorías registradas.
Requisitos	N/A
Significado de los valores de los catálogos	ProductCategoryID: Categoria del producto ProductID: ID del producto más vendido en la categoría Sales: Cantidad de ventas del producto
Comentarios	La sentencia 'WITH' asigna un nombre a una consulta, haciendo más legible la lógica ejecutada.
Propuesta de indices	Indices agrupados: - Product (ProductID): Ayuda en la selección más rápida de productos al ordenarlos a manera de clave primaria. - Order_Detail (ProductID): Nos ayuda a acelerar el JOIN y los SUM. Índices no agrupados: - Product_Category (ProductCategoryID): Ayuda con el auto JOIN.

	ProductCategoryID	ProductID	Sales
1	4	870	6815
2	3	712	8311
3	2	738	1581
4	1	782	2977



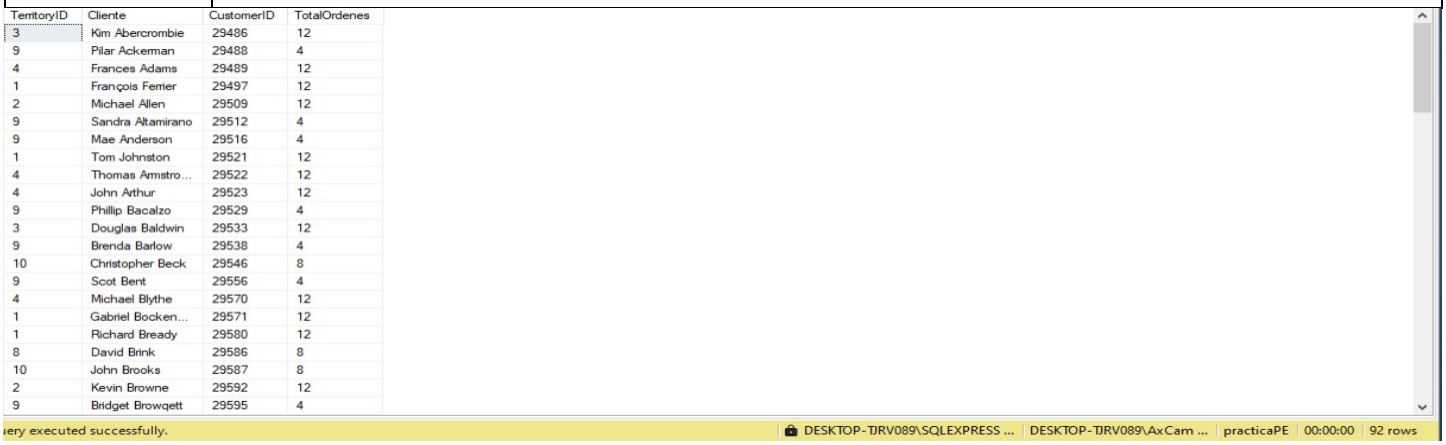
Planes de ejecución

Sin índices:



Con índices:

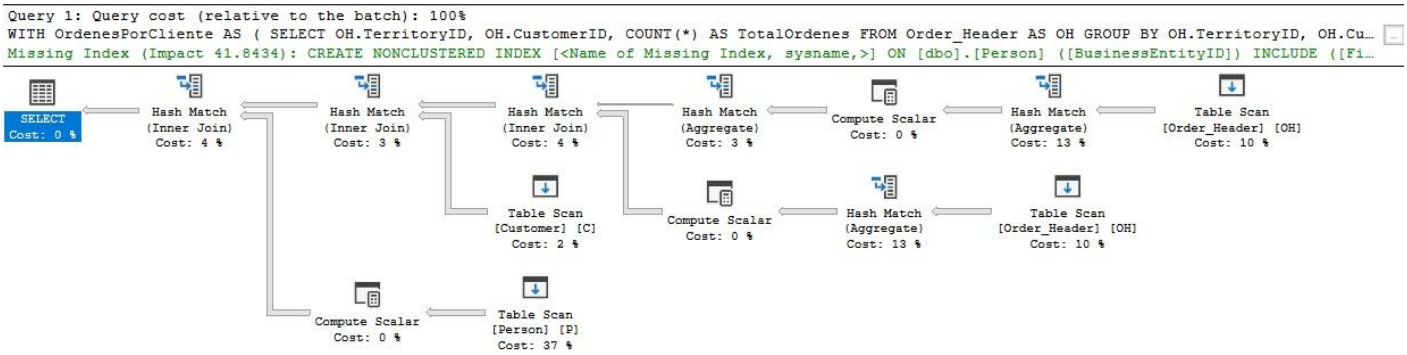
No. consulta	2
Responsable	Axel Iván Rossano Medina
Descripción	Listar el municipio con más casos confirmados recuperados por estado y por año.
Requisitos	N/A
Significado de los valores de los catálogos	CLASIFICACION_FINAL: - 1: Order_Header (información general de la venta del producto). - 2: Customer (información general del usuario y la relación). - 3: Person (información general del customer).
Comentarios	La sentencia 'WITH' asigna un nombre a una consulta, haciendo más legible la lógica ejecutada.
Propuesta de indices	Índices no agrupados: - Person (businessentityID): Mejora el JOIN y accede a las columnas FirstName y LastName para evitar acceder a toda la tabla usando un include. - Order_Header (TerritoryID, CustomerID): Acelera los COUNT y los GROUP BY. - Customer (customerID): Sirve como llave primaria.



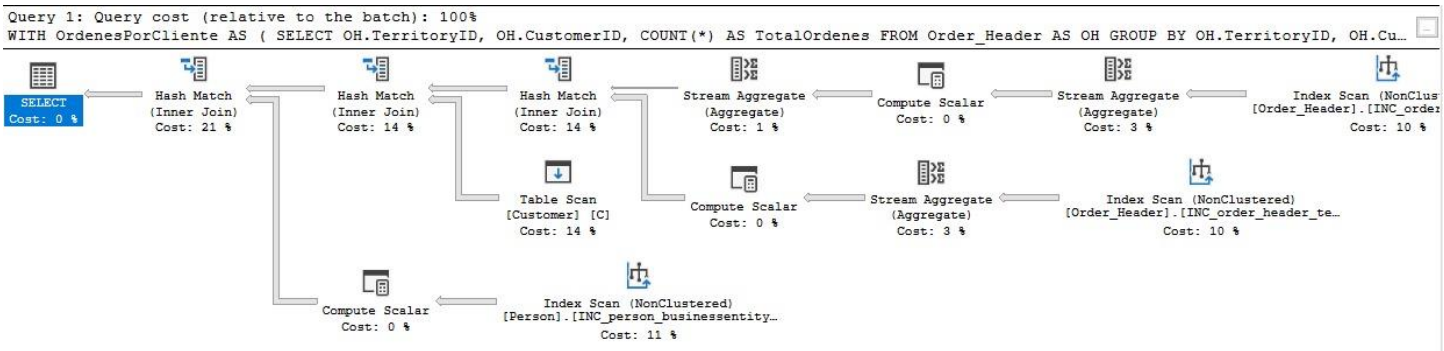
TerritoryID	Cliente	CustomerID	TotalOrdenes
3	Kim Abercrombie	29486	12
9	Pilar Ackerman	29488	4
4	Frances Adams	29489	12
1	François Ferrier	29497	12
2	Michael Allen	29509	12
9	Sandra Altamirano	29512	4
9	Mae Anderson	29516	4
1	Tom Johnston	29521	12
4	Thomas Armstro...	29522	12
4	John Arthur	29523	12
9	Phillip Bacalzo	29529	4
3	Douglas Baldwin	29533	12
9	Brenda Barlow	29538	4
10	Christopher Beck	29546	8
9	Scot Bent	29556	4
4	Michael Blythe	29570	12
1	Gabriel Bocken...	29571	12
1	Richard Bready	29580	12
8	David Brink	29586	8
10	John Brooks	29587	8
2	Kevin Browne	29592	12
9	Bridget Browqett	29595	4

Planes de ejecución

Sin índices:



Con índices:

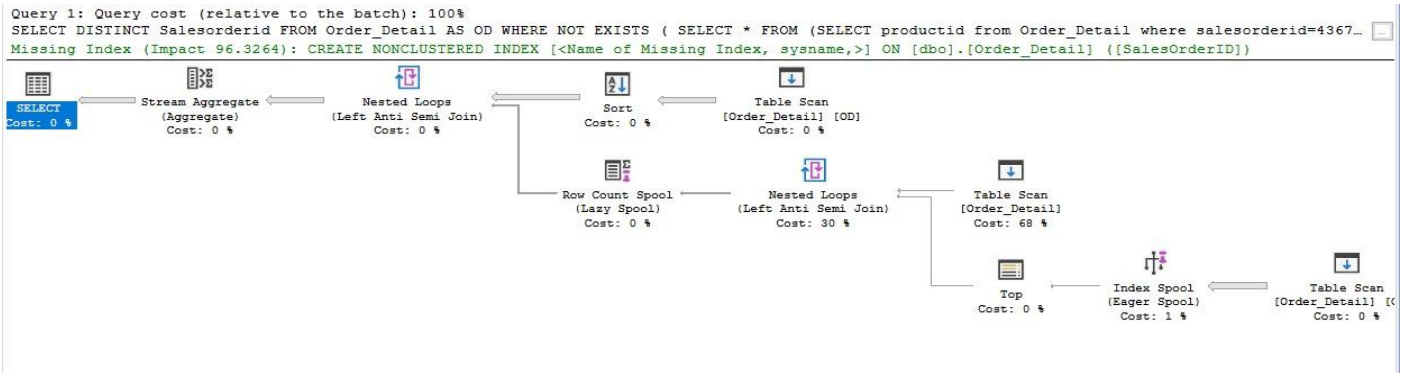


No. consulta	3
Responsable	Erik Bravo
Descripción	Listar el porcentaje de casos confirmados en cada una de las siguientes morbilidades a nivel nacional: diabetes, obesidad e hipertensión.
Requisitos	N/A
Significado de los valores de los catálogos	CLASIFICACION_FINAL: - 1: Order_Detail (información general de las ordenes que hace cada cutomer).
Comentarios	**NOT EXIST:** Ayuda a descartar y dar true a los registros que queremos y descarta los registros que existen en la tabla que estamos comparando
Propuesta de indices	Índices no agrupados: - Order_Detail(SalesOrderID): Mejora la eficiencia del WHERE y NOT EXISTS

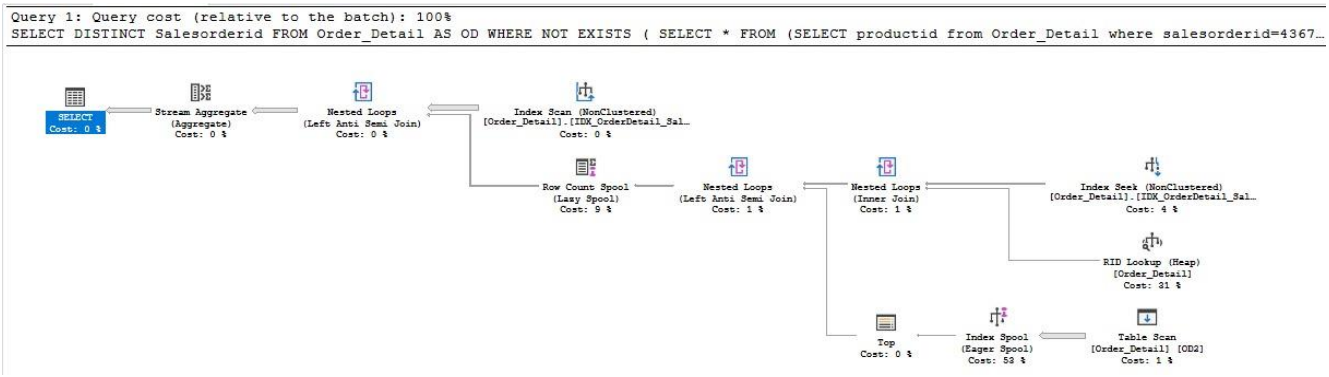
	Salesorderid
1	43676
2	43891
3	43894
4	43900
5	44075
6	44285
7	44523
8	44549
9	44567
10	44779
11	44783
12	44792
13	45785
14	45796
15	46039
16	46052
17	46332

Planes de ejecución

Sin índices:



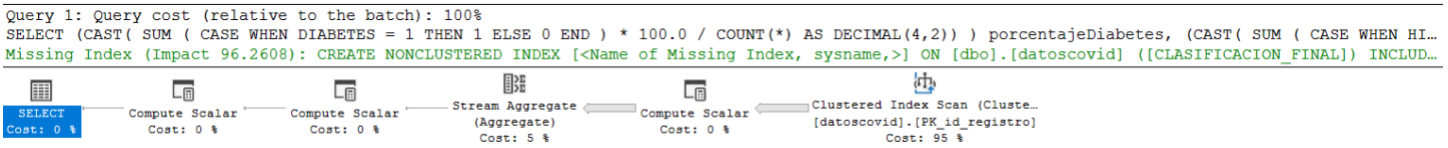
Con índices:



CONSULTAS A LA BASE DE DATOS COVID:

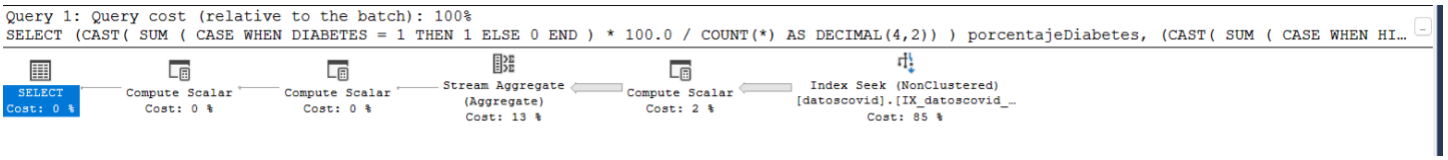
CONSULTA 3:

Sin índices:

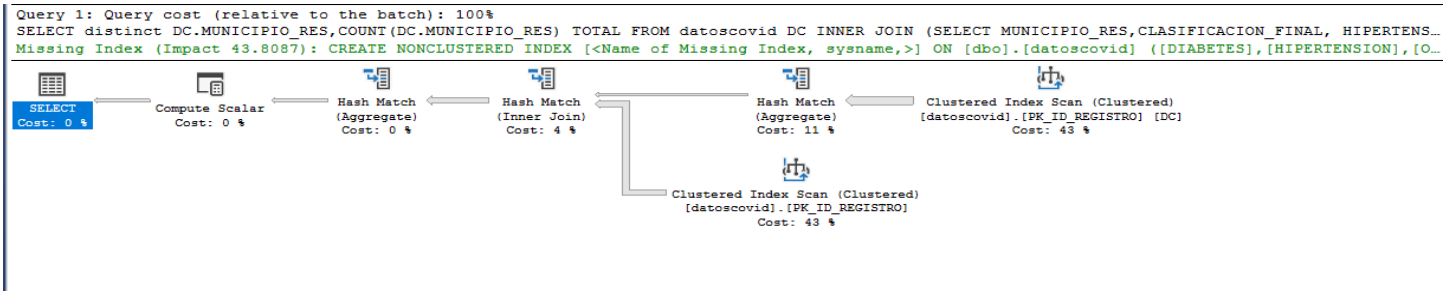


Con índices:

datoscovid (CLASIFICACION_FINAL) INCLUDE (DIABETES, HIPERTENSION, OBESIDAD);



CONSULTA 4:



Con índices:

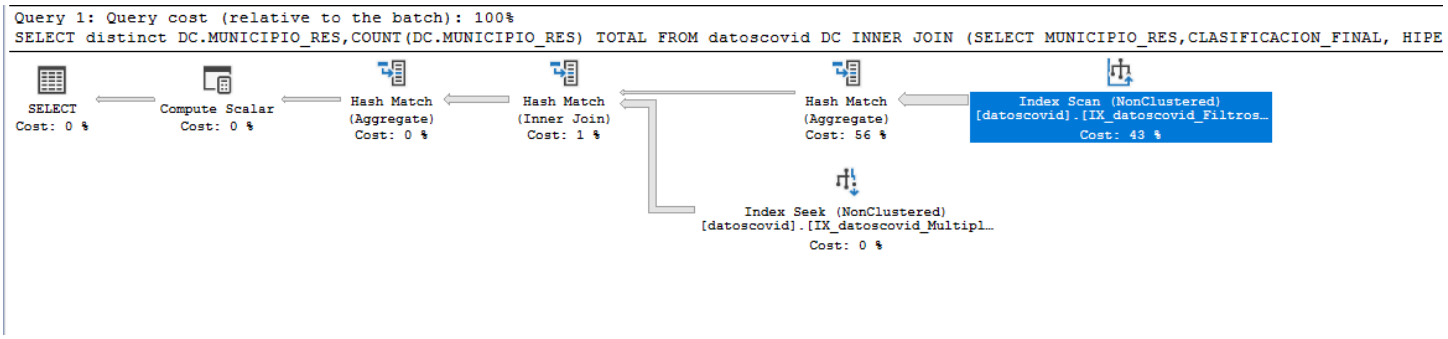
CREATE NONCLUSTERED INDEX IX_datoscovid_Filtros

ON datoscovid (CLASIFICACION_FINAL, HIPERTENSION, OBESIDAD, DIABETES, TABAQUISMO)

INCLUDE (MUNICIPIO_RES);

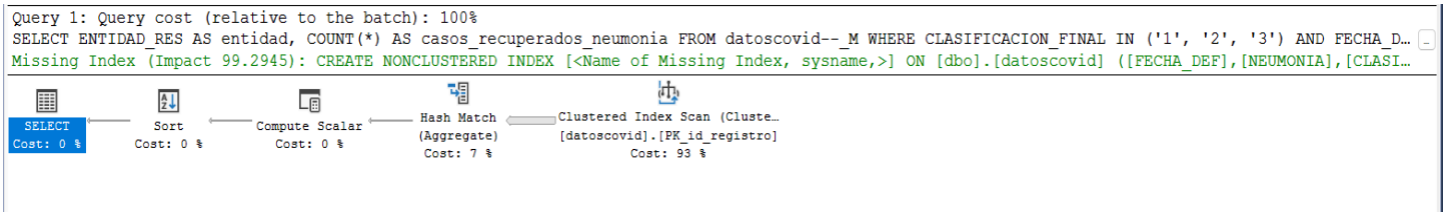
CREATE NONCLUSTERED INDEX IX_datoscovid_MultipleFiltros

ON datoscovid (HIPERTENSION, OBESIDAD, DIABETES, TABAQUISMO, CLASIFICACION_FINAL, MUNICIPIO_RES);



CONSULTA 5:

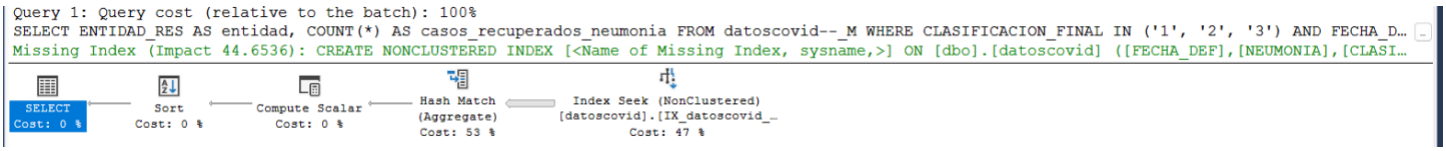
Sin índices:



Con índices:

Índices agrupados:

- datoscovid (CLASIFICACION_FINAL, FECHA_DEF, NEUMONIA, ENTIDAD_RES)



CONCLUSIONES

Para llevar a cabo esta práctica, el equipo acordó que sería más eficiente y rápido analizar las consultas en conjunto. De esta manera, se abordó mejor el proponer los índices una vez hechas las consultas con sus respectivos planes de ejecución, lo que permitió abordar la tarea de forma organizada y colaborativa.

Se estableció una estrategia que consistió en dividir el problema en subconjuntos de datos, lo cual permitió trabajar de manera modular y resolver la consulta paso a paso. Durante el desarrollo de las consultas, se implementaron varias estrategias clave para optimizar el análisis de datos relacionados con la base de datos AdventureWorks. Se utilizo en la mayoría de las consultas subconsultas, with para mejorar la lectura y el análisis de los datos y las diferentes variantes de la sentencia JOIN.

Los índices clustered ordenan físicamente los datos en la tabla según la clave del índice, por lo que solo puede haber uno por tabla. Son ideales para consultas que retornan rangos de datos o buscan valores ordenados. En cambio, los índices nonclustered crean una estructura separada que contiene las claves del índice y punteros a las filas de la tabla, permitiendo múltiples índices por tabla. Para optimizar consultas, se pueden crear índices nonclustered simples o compuestos, y también incluir columnas adicionales (INCLUDE) para cubrir consultas sin necesidad de acceder a la tabla base. Esto mejora el rendimiento al reducir operaciones costosas como los "Lookup". Los parámetros clave para los índices son las columnas usadas en filtros o joins (como SalesOrderID y ProductID), que deben estar en la clave o incluidas para un acceso eficiente.

Por ejemplo, la segunda consulta de la base de datos AdventureWorks, SalesOrderID incluye todos los productos de un pedido específico, realizando comparaciones entre múltiples filas y subconsultas. Para optimizarla, un índice nonclustered sobre SalesOrderID porque permite realizar búsquedas rápidas por número de orden, reemplazando el Table Scan por un Index Seek.

Sin embargo, el plan de ejecución muestra que, aunque el índice sobre SalesOrderID ayuda, el sistema realiza operaciones de Lookup para obtener columnas adicionales (como ProductID) que no están en el índice, lo que implica acceso extra a la tabla base y eleva el costo.

Los indices propuestos son ejemplos de como SQL Server puede cubrir consultas específicas sin tener que acceder a la toda la tabla, mejorando el rendimiento. Por ejemplo, el índice INC_person_businessentityID permite buscar rápidamente por BusinessEntityID y obtener el nombre completo (FirstName, LastName) directamente del índice sin buscar en la tabla principal.

De manera similar, un índice compuesto o con columnas incluidas en Order_Detail que abarque SalesOrderID y ProductID evitaría el Lookup y permitiría que toda la información necesaria estuviera disponible en el índice. Esto se traduciría en un plan de ejecución con más Index Seek y menos acceso directo a la tabla, reduciendo el costo total.

En conclusión, el correcto diseño y aplicación de índices en SQL Server es fundamental para optimizar el rendimiento de las consultas, especialmente en bases de datos con grandes volúmenes de información como AdventureWorks. La combinación adecuada entre índices clustered y nonclustered, junto con el uso estratégico de columnas incluidas (INCLUDE), permite minimizar operaciones costosas como los escaneos completos de tabla y los lookups, favoreciendo que el motor de base de datos utilice accesos más eficientes como Index Seek. Esto no solo mejora la velocidad de respuesta, sino que también reduce la carga en recursos del servidor, facilitando un mejor aprovechamiento de la infraestructura.

Finalmente, el trabajo colaborativo y modular llevado a cabo permitió un análisis profundo y ordenado, garantizando que cada consulta fuera revisada con detenimiento y optimizada con los índices más adecuados. Este enfoque sistemático y basado en evidencia de planes de ejecución es una práctica recomendada para cualquier proyecto de gestión de bases de datos, ya que asegura resultados eficientes y escalables que responden a las necesidades reales de consulta y procesamiento de datos. Así, esta práctica contribuye al desarrollo de competencias esenciales en administración y optimización de bases de datos, que son clave para la gestión eficiente de sistemas.