

INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA  
EN INGENIERIA Y TECNOLOGÍAS AVANZADAS

BASES DE DATOS DISTRIBUIDAS

GRUPO: 3TM3

*PRÁCTICA 2*

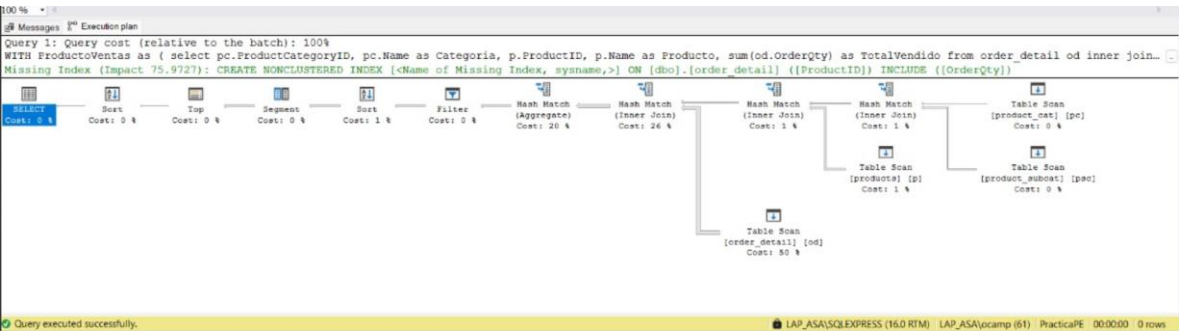
PROFESOR: DE LA CRUZ SOSA CARLOS

INTEGRANTES EQUIPO 2:

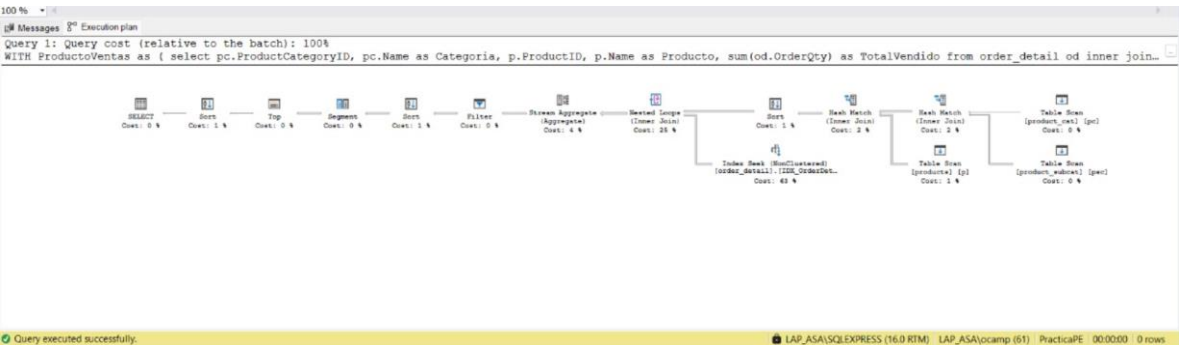
OCAMPO CABRERA ASael	2022640070
ROBERT ROA KARLA GUADALUPE	2022640710
SANCHEZ VILLAGRANA OSMAR ROBERTO	2022640514

<b>Número de Consulta</b>	Número 1: Listar el producto más vendido de cada una de las categorías registradas en la base de datos.
<b>Requisitos</b>	<ul style="list-style-type: none"><li>- El índice sugerido es: CREATE NONCLUSTERED INDEX IDX_OrderDetail_Composite ON order_detail (ProductID, SalesOrderID) INCLUDE (OrderQty);</li><li>- Ejecutar la consulta con CTE para calcular el total vendido por producto y categoría.</li><li>- Obtener el producto con mayor venta dentro de cada categoría.</li></ul>
<b>Significado de los valores de los catálogos</b>	<ul style="list-style-type: none"><li>• <b>ProductCategoryID:</b> Identificador único de cada categoría de producto.</li><li>• <b>Categoría:</b> Nombre descriptivo de la categoría.</li><li>• <b>ProductID:</b> Identificador único del producto.</li><li>• <b>Producto:</b> Nombre descriptivo del producto.</li><li>• <b>TotalVendido:</b> Suma total de la cantidad vendida por producto.</li></ul>
<b>Responsable de la consulta</b>	Karla Guadalupe Robert Roa
<b>Comentarios</b>	<ul style="list-style-type: none"><li>✓ La creación del índice mejora significativamente el rendimiento, evitando escaneos completos en order_detail.</li><li>✓ La consulta usa varias uniones y agregaciones para determinar el producto líder en ventas por categoría.</li><li>✓ El uso del índice compuesto y columnas incluidas permite acelerar la suma y el agrupamiento de los datos.</li></ul>

Consulta a) Antes de los Índices

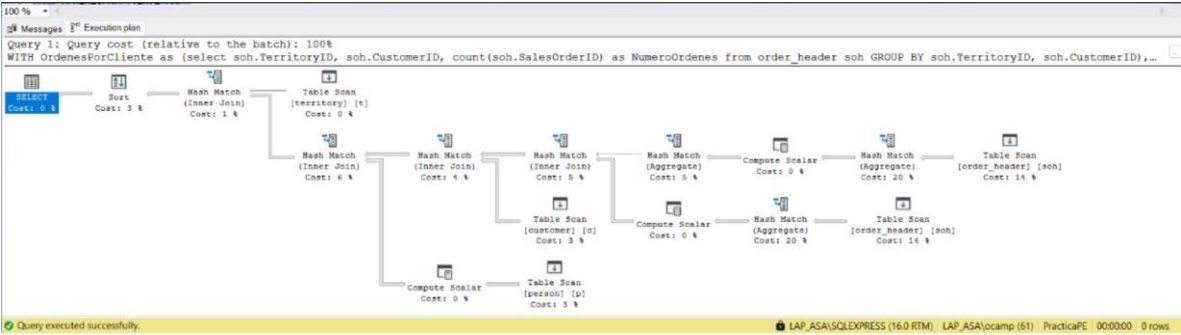


Consulta a) Después de los Índices

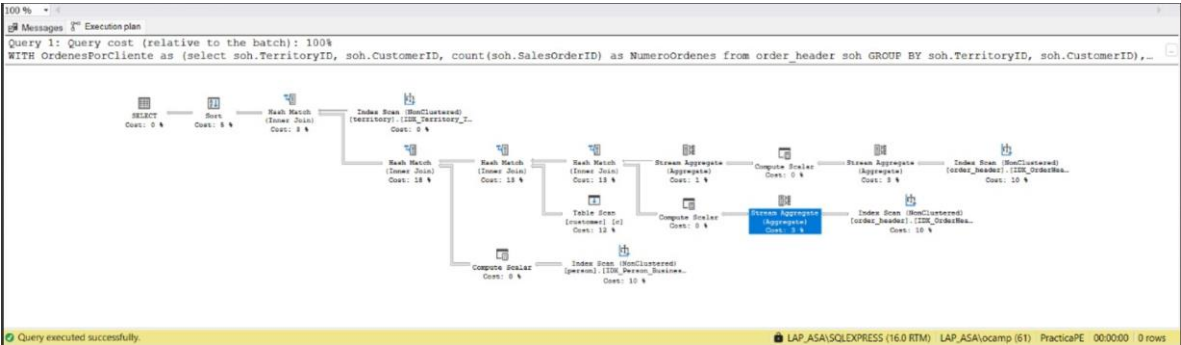


<b>Número de Consulta</b>	Número 2: Listar el nombre de los clientes con más órdenes por cada uno de los territorios registrados en la base de datos.
<b>Requisitos</b>	<ul style="list-style-type: none"> <li>- Índice compuesto en order_header (TerritoryID, CustomerID) con columna incluida SalesOrderID.</li> <li>- Índice simple en customer (CustomerID).</li> <li>- Índice compuesto en person (BusinessEntityID) con columnas incluidas FirstName, LastName.</li> <li>- Índice compuesto en territory (TerritoryID) con columna incluida Name.</li> </ul>
<b>Significado de los valores de los catálogos</b>	<ul style="list-style-type: none"> <li>• <b>TerritoryID:</b> Identificador único de territorio.</li> <li>• <b>CustomerID:</b> Identificador único de cliente.</li> <li>• <b>SalesOrderID:</b> Identificador único de orden de venta.</li> <li>• <b>FirstName, LastName:</b> Nombre y apellido del cliente.</li> <li>• <b>Name:</b> Nombre del territorio.</li> </ul>
<b>Responsable de la consulta</b>	Osmar Roberto Sánchez Villagrana
<b>Comentarios</b>	<ul style="list-style-type: none"> <li>✓ Los índices propuestos permiten acelerar la agrupación, conteo y unión de tablas.</li> <li>✓ La consulta obtiene clientes con el mayor número de órdenes en cada territorio y muestra nombres completos y territorios.</li> <li>✓ La inclusión de columnas en índices mejora el acceso y evita lecturas adicionales.</li> </ul>

### Consulta b) Antes de los Índices

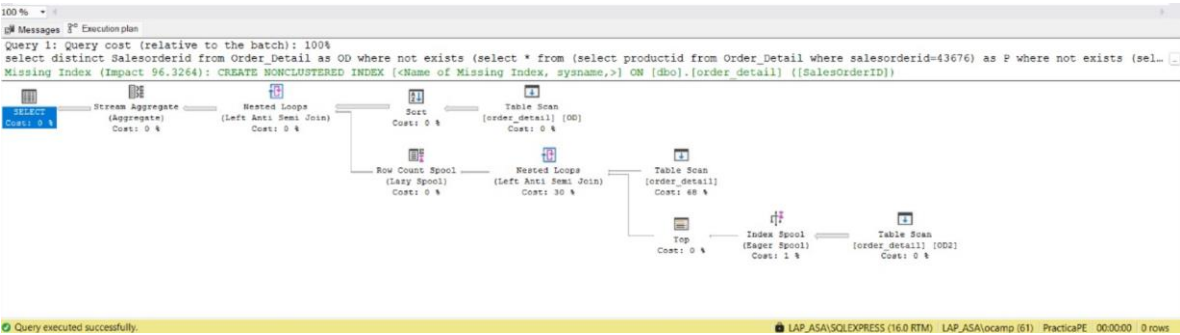


### Consulta b) Después de los Índices

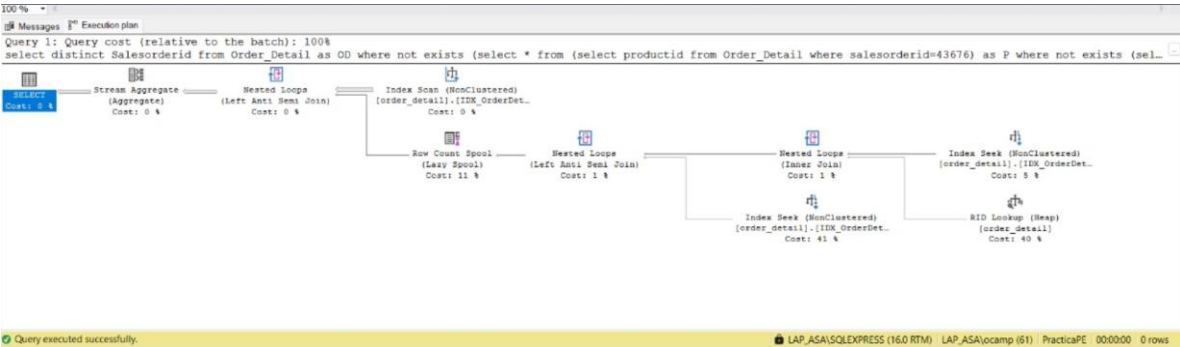


<b>Número de Consulta</b>	Número 3: Listar los datos generales de las ordenes que tengan al menos los mismos productos de la orden con <b>salesorderid = 43676</b> .
<b>Requisitos</b>	<ul style="list-style-type: none"> <li>- Crear un índice no clusterizado en la tabla <b>order_detail</b> sobre la columna <b>SalesOrderID</b> para acelerar búsquedas por orden.</li> <li>- La consulta utiliza la cláusula <b>NOT EXISTS</b> anidada para filtrar órdenes que contienen todos los productos de la orden 43676.</li> <li>- Se emplea la estructura de subconsultas para comprobar que no falte ningún producto de la orden base en las otras órdenes.</li> </ul>
<b>Significado de los valores de los catálogos</b>	<ul style="list-style-type: none"> <li>• <b>SalesOrderID</b>: Identificador único de cada orden de venta.</li> <li>• <b>ProductID</b>: Identificador único de cada producto.</li> </ul> <p>La consulta busca órdenes que sean un superconjunto de productos de una orden específica.</p>
<b>Responsable de la consulta</b>	Asael Ocampo Cabrera
<b>Comentarios</b>	<ul style="list-style-type: none"> <li>✓ La creación del índice mejora el acceso a las filas correspondientes a cada orden, evitando escaneos completos de la tabla.</li> <li>✓ La consulta puede ser costosa en tablas grandes debido a la naturaleza de NOT EXISTS anidado, pero el índice mitiga este impacto.</li> <li>✓ Se recomienda mantener estadísticas actualizadas para un óptimo plan de ejecución.</li> </ul>

### Consulta c) Antes de los Índices



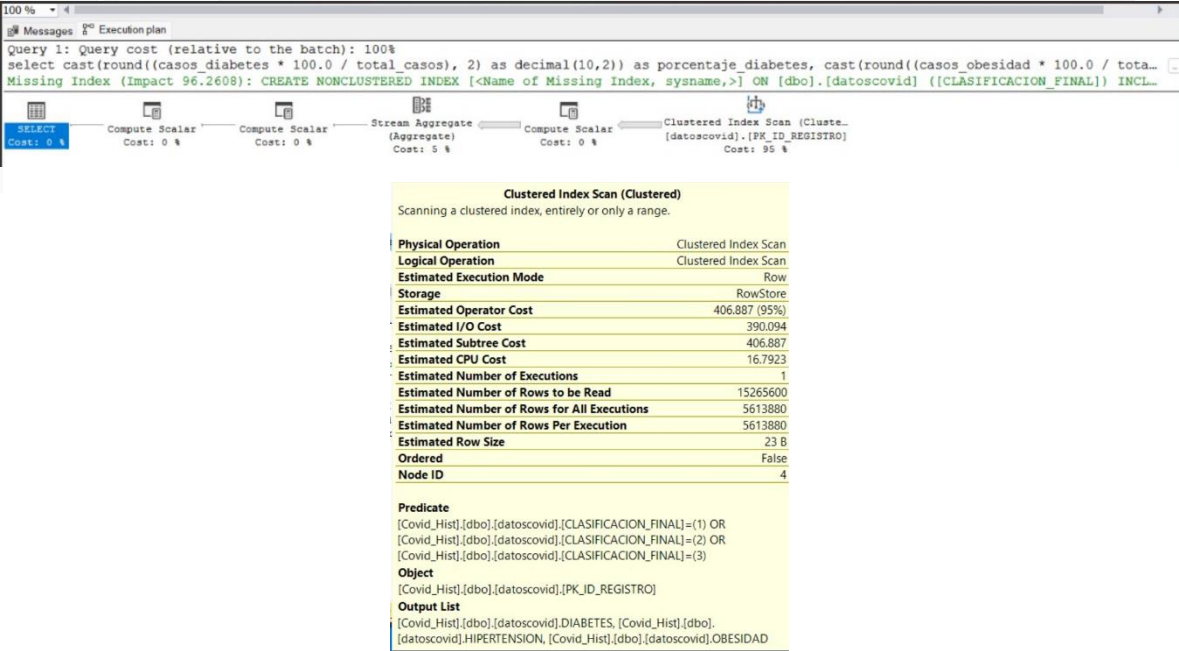
### Consulta c) Después de los Índices



6. Generar los planes de ejecución de las consultas 3, 4 y 5 de la práctica de consultas en la base de datos Covid y proponer índices para mejorar el rendimiento.

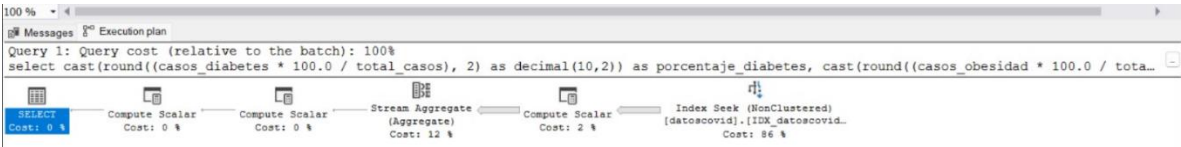
Consulta 3

Sin índices



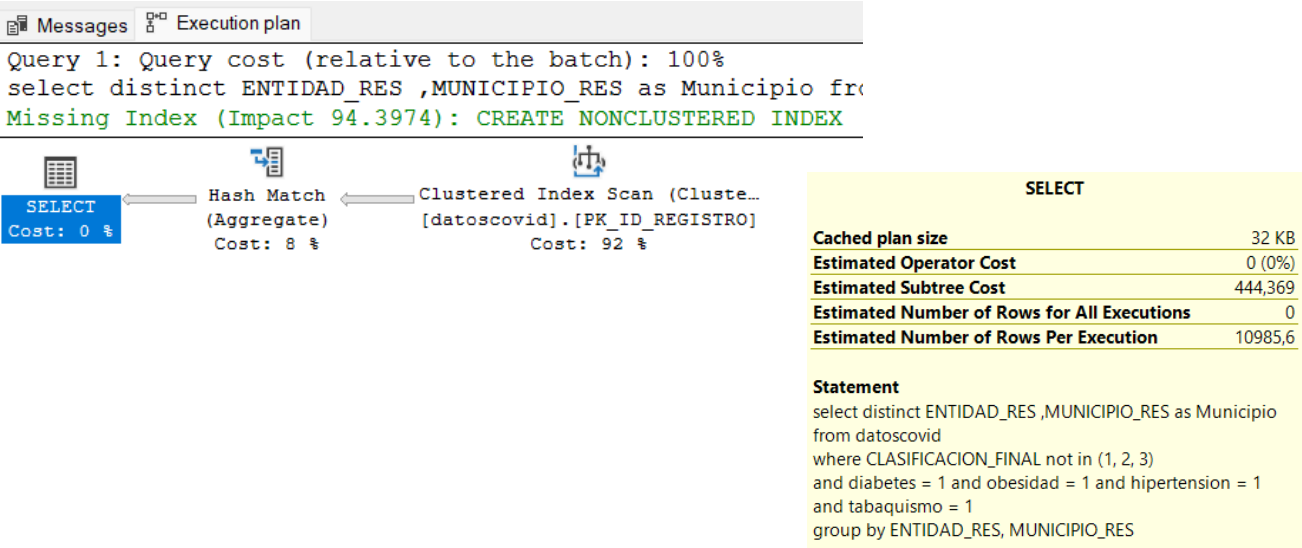
Con índices

CREATE NONCLUSTERED INDEX IDX\_datoscovid\_CLASIF\_MORBILIDADES  
ON datoscovid (CLASIFICACION\_FINAL)  
INCLUDE (diabetes, obesidad, hipertension, tabaquismo);



Consulta 4

Sin índices



Hash Match	
Use each row from the top input to build a hash table, and each row from the bottom input to probe into the hash table, outputting all matching rows.	
Physical Operation	Hash Match
Logical Operation	Aggregate
Estimated Execution Mode	Row
Estimated Operator Cost	37,482 (8%)
Estimated I/O Cost	0
Estimated Subtree Cost	444,369
Estimated CPU Cost	1,15077
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	10985,6
Estimated Number of Rows for All Executions	10985,6
Estimated Row Size	23 B
Node ID	0
<b>Output List</b>	
[Covid_Historico].[dbo].[datoscovid].[ENTIDAD_RES];	
[Covid_Historico].[dbo].[datoscovid].[MUNICIPIO_RES]	
<b>Build Residual</b>	
[Covid_Historico].[dbo].[datoscovid].[ENTIDAD_RES] =	
[Covid_Historico].[dbo].[datoscovid].[ENTIDAD_RES] AND	
[Covid_Historico].[dbo].[datoscovid].[MUNICIPIO_RES] =	
[Covid_Historico].[dbo].[datoscovid].[MUNICIPIO_RES]	

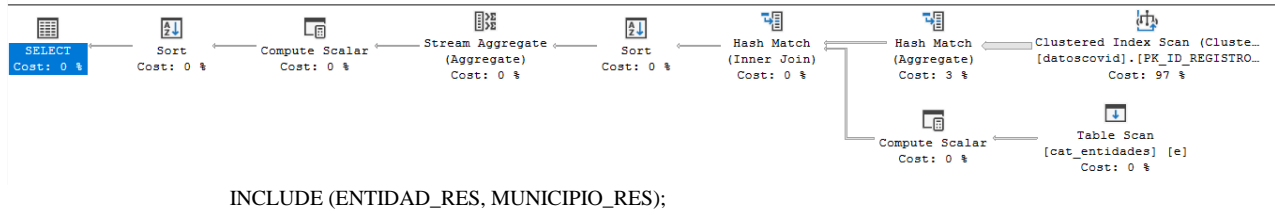
Clustered Index Scan (Clustered)	
Scanning a clustered index, entirely or only a range.	
Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Estimated Execution Mode	Row
Storage	RowStore
Estimated Operator Cost	406,887 (92%)
Estimated I/O Cost	390,094
Estimated Subtree Cost	406,887
Estimated CPU Cost	16,7923
Estimated Number of Executions	1
Estimated Number of Rows to be Read	15265600
Estimated Number of Rows for All Executions	112641
Estimated Number of Rows Per Execution	112641
Estimated Row Size	43 B
Ordered	False
Node ID	1
<b>Predicate</b>	
[Covid_Historico].[dbo].[datoscovid].[CLASIFICACION_FINAL] <> (1) AND	
[Covid_Historico].[dbo].[datoscovid].[CLASIFICACION_FINAL] <> (2) AND	
[Covid_Historico].[dbo].[datoscovid].[CLASIFICACION_FINAL] <> (3) AND	
[Covid_Historico].[dbo].[datoscovid].[DIABETES] = (1) AND	
[Covid_Historico].[dbo].[datoscovid].[OBESIDAD] = (1) AND	
[Covid_Historico].[dbo].[datoscovid].[HIPERTENSION] = (1) AND	
[Covid_Historico].[dbo].[datoscovid].[TABAQUISMO] = (1)	
<b>Object</b>	
[Covid_Historico].[dbo].[datoscovid].[PK_ID_REGISTRO]	
<b>Output List</b>	
[Covid_Historico].[dbo].[datoscovid].[ENTIDAD_RES]; [Covid_Historico].[	
[dbo].[datoscovid].[MUNICIPIO_RES]	

## Con índices

CREATE NONCLUSTERED INDEX IDX\_DATOSCOVID\_MUNICIPIOS\_MORBILIDADES  
ON datoscovid (CLASIFICACION\_FINAL, diabetes, obesidad, hipertension, tabaquismo)

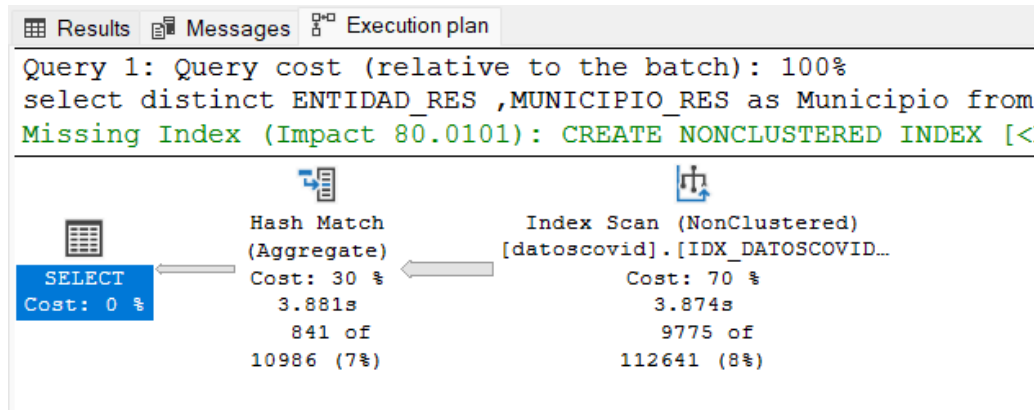
Query 1: Query cost (relative to the batch): 100%

select e.entidad as Estado, count(\*) as Total\_con\_neumonia from datoscovid d join cat\_entidades e on d.ENTIDAD\_UM = e.clave where n  
Missing Index (Impact 96.3097): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([NEUMONIA]) IN



## Consulta 5

### Sin índices

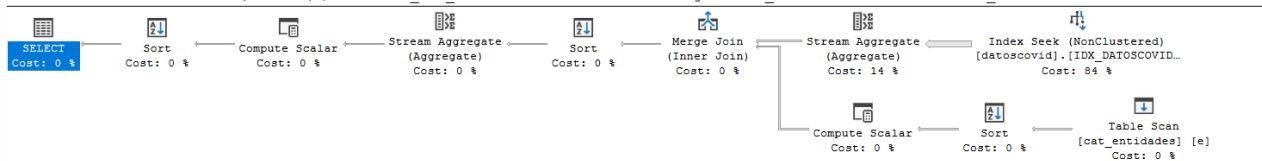


### Con índices

CREATE NONCLUSTERED INDEX IDX\_DATOSCOVID\_NEUMONIA  
ON datoscovid (neumonia, ENTIDAD\_UM);

Query 1: Query cost (relative to the batch): 100%

select e.entidad as Estado, count(\*) as Total\_con\_neumonia from datoscovid d join cat\_entidades e on d.ENTIDAD\_UM = e.clave where neumonia =...



## **7. Comparar los planes de ejecución del punto 6 con los planes de ejecución de otro equipo.**

Para este punto, se realizó la comparación entre los planes de ejecución generados por nuestras consultas y los presentados por el equipo 6. A pesar de que ambos equipos implementaron índices para optimizar las consultas sobre la tabla principal datoscovid, observamos que el costo de ejecución sigue concentrado en el escaneo clustered de la tabla, lo cual indica que los índices no están siendo completamente aprovechados por el optimizador. Esto puede deberse a que los índices creados no son suficientemente específicos o no cubren de manera óptima las columnas utilizadas en los filtros y agregaciones de las consultas. Por lo tanto, ambos planes presentan costos elevados y patrones similares de acceso a datos, reflejando la necesidad de mejorar la selección y diseño de índices, por ejemplo, usando índices filtrados o compuestos que reduzcan el volumen de datos escaneados y mejoren el rendimiento general. Esta comparación nos permitió entender la importancia de ajustar los índices a las consultas específicas para obtener mejoras reales en la ejecución.

## **8. Conclusiones**

Durante el desarrollo de la práctica, nuestro equipo se enfocó en optimizar el rendimiento de las consultas mediante la implementación de índices adecuados basados en el análisis de los planes de ejecución. La estrategia principal fue crear índices compuestos y cubrir columnas utilizadas en filtros, joins y agregaciones, con el objetivo de reducir escaneos completos de tablas y acelerar el acceso a los datos.

En la consulta que lista el producto más vendido por categoría, se diseñó un índice compuesto en la tabla order\_detail sobre ProductID y SalesOrderID, incluyendo la columna OrderQty. Este índice permitió realizar agrupaciones y sumas de manera eficiente, mejorando el tiempo de respuesta de la consulta. Para la consulta que obtiene los clientes con más órdenes en cada territorio, se crearon índices en las tablas order\_header, customer, person y territory, que facilitaron los joins y las agrupaciones necesarias, disminuyendo el costo de ejecución y mejorando el acceso a los datos relacionados.

Para la consulta que identifica órdenes que contienen al menos los mismos productos que una orden específica, se implementó un índice sobre SalesOrderID en la tabla order\_detail. Esto permitió optimizar la comparación de productos entre órdenes mediante subconsultas anidadas, acelerando la ejecución y evitando búsquedas completas innecesarias. En general, la selección de índices se basó en las columnas más usadas en condiciones y uniones, y en incluir columnas que cubren completamente las consultas para evitar accesos adicionales a las tablas base. Esta metodología permitió mejorar significativamente el rendimiento de las consultas, tal como se reflejó en los planes de ejecución.

En el análisis de las consultas sobre la base de datos Covid, al comparar los planes de ejecución con y sin índices, se observó que, aunque la implementación de índices específicos mejoró ligeramente el rendimiento, el costo principal seguía concentrado en el escaneo clustered de la tabla datoscovid. Esto indica que los índices actuales no son completamente aprovechados por el optimizador debido a su falta de especificidad o a que no cubren de forma óptima las columnas utilizadas en los filtros y agregaciones. Por lo tanto, se concluye que es necesario diseñar índices más específicos y efectivos, como índices filtrados o compuestos, que reduzcan el volumen de datos escaneados y optimicen realmente el rendimiento de las consultas. La comparación con otro equipo evidenció patrones similares, reafirmando la importancia de ajustar y afinar la selección de índices según las consultas específicas para lograr mejoras significativas en los tiempos de ejecución y eficiencia general.