

Análisis de los planes de ejecución de la base de datos COVID 19“Listado de consultas a programar para analizar planes de ejecución”

Configuración de la base de datos:

A partir del backup proporcionado por el profesor en la tabla original implemente los tres índices que vi conveniente, mientras que realice una copia de la tabla que no cuenta con índices por lo que en dicha tabla analizo los planes de ejecución sin índices/llaves.

```
IX_DatosCovid_CLASIFICACION_FINAL_ENTIDAD (Non-Unique, N  
IX_DatosCovid_FECHA_DEF (Non-Unique, Non-Clustered)  
PK_datoscovid_ID_Registro (Clustered)
```

Dichos índices iniciales son:

1. Un índice primario
2. Un índice de la fecha de defunción
3. Un índice que combina la clasificación final con la entidad y municipio de residencia

Estos índices iniciales están definidos con base en las combinaciones comunes de columnas durante el desarrollo de las consultas

Para la descripción de los análisis elegí en la mayoría de los casos tomar la consulta que considere más practica y sobre ese marque índice y no índice para diferenciar

1.Listar los casos positivos por entidad de residencia

Consulta 1:

```
select ENTIDAD_RES, count(*) total_confimado from dbo.datoscovid  
where CLASIFICACION_FINAL between 1 and 3  
group by ENTIDAD_RES  
order by ENTIDAD_RES
```

Consulta 2:

```
select ENTIDAD_RES, count(*) total_confirmado  
from dbo.datoscovid  
where CLASIFICACION_FINAL between 1 and 3  
group by ENTIDAD_RES  
order by ENTIDAD_RES
```

Sin índices:

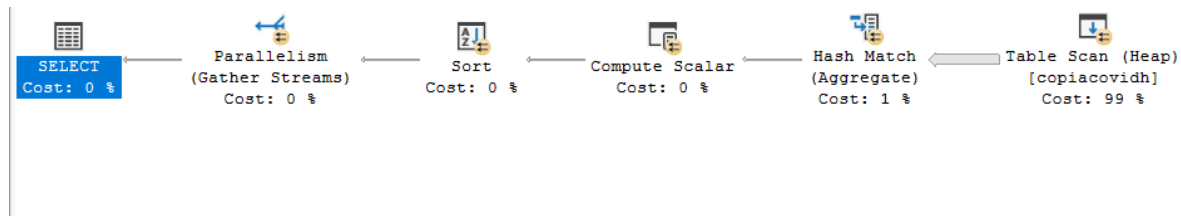
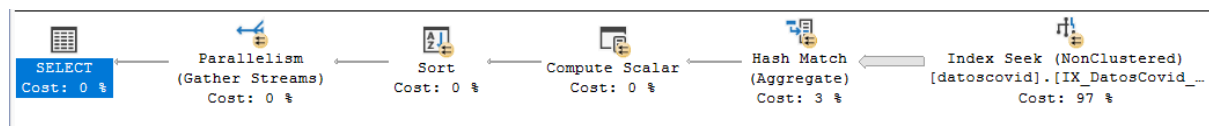


Table Scan (Heap)	
Scan rows from a table.	
Physical Operation	Table Scan
Logical Operation	Table Scan
Estimated Execution Mode	Batch
Storage	RowStore
Estimated Operator Cost	406.658 (99%)
Estimated I/O Cost	403.859
Estimated Subtree Cost	406.658
Estimated CPU Cost	2.79872
Estimated Number of Executions	1
Estimated Number of Rows to be Read	15265600
Estimated Number of Rows for All Executions	5597090
Estimated Number of Rows Per Execution	5597090
Estimated Row Size	19 B
Ordered	False
Node ID	4
Predicate	
[covidHistorico].[dbo].[copiacovidh].[CLASIFICACION_FINAL] >=(1) AND [covidHistorico].[dbo].[copiacovidh].[CLASIFICACION_FINAL] <=(3)	
Object	
[covidHistorico].[dbo].[copiacovidh]	
Output List	
[covidHistorico].[dbo].[copiacovidh].ENTIDAD_RES	

Con índices:



Index Seek (NonClustered)	
Scan a particular range of rows from a nonclustered index.	
Physical Operation	Index Seek
Logical Operation	Index Seek
Estimated Execution Mode	Batch
Storage	RowStore
Estimated Operator Cost	21.4957 (97%)
Estimated I/O Cost	20.4735
Estimated Subtree Cost	21.4957
Estimated CPU Cost	1.02222
Estimated Number of Executions	1
Estimated Number of Rows to be Read	5575610
Estimated Number of Rows for All Executions	5575610
Estimated Number of Rows Per Execution	5575610
Estimated Row Size	15 B
Ordered	True
Node ID	4
Object	
[covidHistorico].[dbo].[datos covid]. [IX_DatosCovid_CLASIFICACION_FINAL_ENTIDAD]	
Output List	
[covidHistorico].[dbo].[datos covid].ENTIDAD_RES	
Seek Predicates	
Seek Keys[1]: Start: [covidHistorico].[dbo]. [datos covid].CLASIFICACION_FINAL >= Scalar Operator((1)), End: [covidHistorico].[dbo].[datos covid].CLASIFICACION_FINAL <= Scalar Operator((3))	

Análisis:

A partir de esta consulta llegue al desarrollo del índice que involucra la clasificación final, la entidad y municipio de residencia que resulta útil en varias de las consultas ya que ocupan dicho índice en lugar de los múltiples table scan con alto costo de ejecución que maneja la tabla sin índices.

2. Listar los casos sospechosos por entidad

Consulta propia:

```
select ENTIDAD_UM, ENTIDAD_RES, count(*) total_sospechosos from dbo.datoscovid
where CLASIFICACION_FINAL = 6
group by ENTIDAD_RES, ENTIDAD_UM
order by ENTIDAD_UM
```

Consulta 2:

```
Select ENTIDAD_UM, ENTIDAD_RES, count(*) total_sospechosos
from dbo.datoscovid
where CLASIFICACION_FINAL = 6
group by ENTIDAD_UM, ENTIDAD_RES
order by ENTIDAD_UM
```

Sin índices:

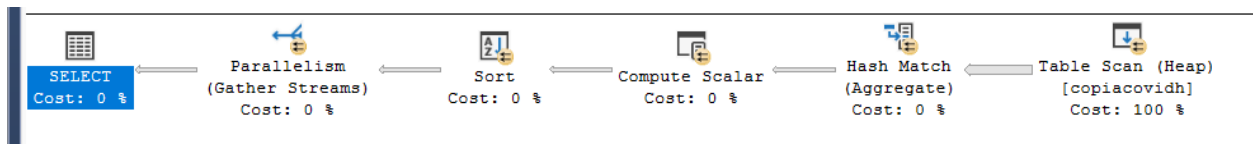
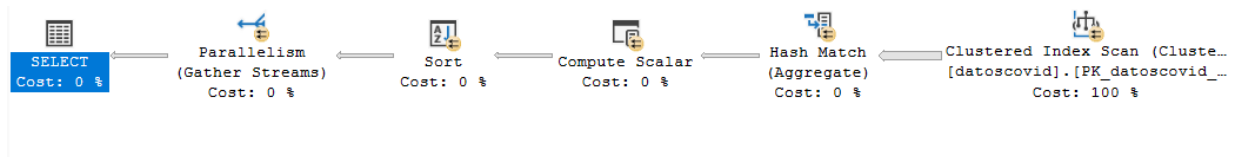


Table Scan (Heap)	
Scan rows from a table.	
Physical Operation	Table Scan
Logical Operation	Table Scan
Estimated Execution Mode	Batch
Storage	RowStore
Estimated Operator Cost	406.658 (100%)
Estimated I/O Cost	403.859
Estimated Subtree Cost	406.658
Estimated CPU Cost	2.79872
Estimated Number of Executions	1
Estimated Number of Rows to be Read	15265600
Estimated Number of Rows for All Executions	570493
Estimated Number of Rows Per Execution	570493
Estimated Row Size	25 B
Ordered	False
Node ID	4
Predicate	
[covidHistorico].[dbo].[copiacovidh].[CLASIFICACION_FINAL]=(6)	
Object	
[covidHistorico].[dbo].[copiacovidh]	
Output List	
[covidHistorico].[dbo].[copiacovidh].ENTIDAD_UM, [covidHistorico].[dbo].[copiacovidh].ENTIDAD_RES	

Con índices:



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	Batch
Storage	RowStore
Estimated Operator Cost	406.657 (100%)
Estimated I/O Cost	403.858
Estimated Subtree Cost	406.657
Estimated CPU Cost	2.79872
Estimated Number of Executions	1
Estimated Number of Rows to be Read	15265600
Estimated Number of Rows for All Executions	572785
Estimated Number of Rows Per Execution	572785
Estimated Row Size	25 B
Ordered	False
Node ID	4
Predicate	
[covidHistorico].[dbo].[datoscovid].[CLASIFICACION_FINAL]=(6)	
Object	
[covidHistorico].[dbo].[datoscovid].[PK_datoscovid_ID_Registro]	
Output List	
[covidHistorico].[dbo].[datoscovid].ENTIDAD_UM, [covidHistorico].[dbo].[datoscovid].ENTIDAD_RES	

Análisis:

A diferencia de la consulta 1, esta tabla utilizada en lugar de la combinación de clasificación final con entidad y municipio de residencia cambia al municipio por la entidad de la unidad médica, un índice similar podría ser creado para reducir costos de ejecución, pero debido a que esta es la única referencia a entidad_um que realizo en mis consultas

he decidido omitirla, aun así, observamos un nivel similar en el costo del operador respecto a la ejecución sin ninguna clase de índice.

3. Listar el top 5 de municipios por entidad con el mayor número de casos reportados, indicando casos sospechosos y casos confirmados.

Consulta propia:

```
select cc.ENTIDAD_RES, cc.MUNICIPIO_RES, cc.confirmado, cs.sospechoso
from (select ENTIDAD_RES, MUNICIPIO_RES, count(*) as sospechoso
      from dbo.datoscovid where CLASIFICACION_FINAL = 6
      group by ENTIDAD_RES, MUNICIPIO_RES
    ) cs
inner join
(select ENTIDAD_RES, MUNICIPIO_RES, count (*) as confirmado
 from dbo.datoscovid where CLASIFICACION_FINAL between 1 and 3
 group by ENTIDAD_RES, MUNICIPIO_RES) cc
on cc.ENTIDAD_RES = cs.ENTIDAD_RES and cs.MUNICIPIO_RES = cc.MUNICIPIO_RES
order by cc.ENTIDAD_RES
```

Sin índice :

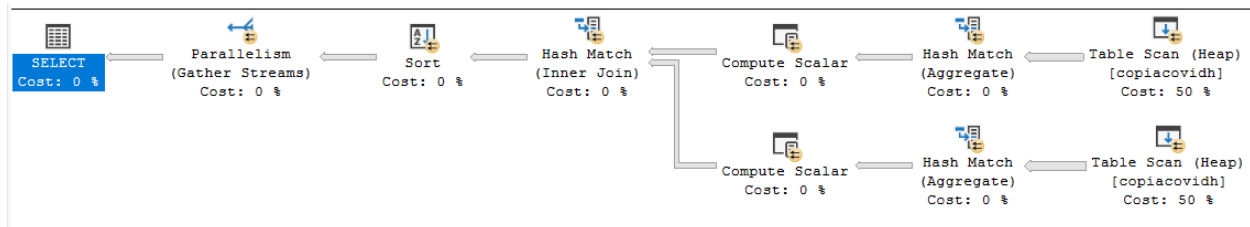
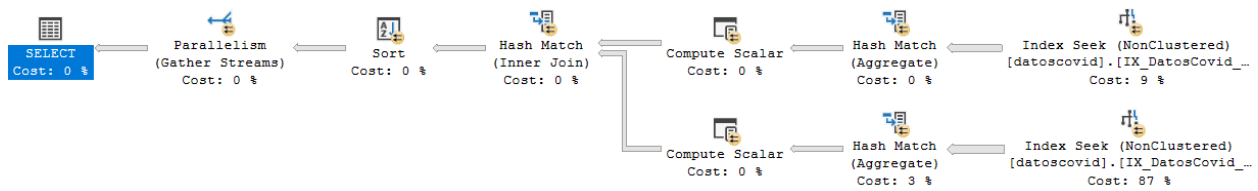


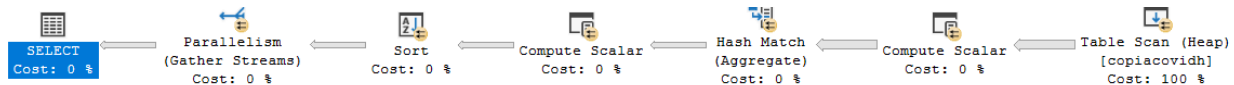
Table Scan (Heap)	
Scan rows from a table.	
Physical Operation	Table Scan
Logical Operation	Table Scan
Estimated Execution Mode	Batch
Storage	RowStore
Estimated Operator Cost	406.658 (50%)
Estimated I/O Cost	403.859
Estimated Subtree Cost	406.658
Estimated CPU Cost	2.79872
Estimated Number of Executions	1
Estimated Number of Rows to be Read	15265600
Estimated Number of Rows for All Executions	570493
Estimated Number of Rows Per Execution	570493
Estimated Row Size	27 B
Ordered	False
Node ID	5
Predicate	
[covidHistorico].[dbo].[copiacovidh].[CLASIFICACION_FINAL]=(6)	
Object	
[covidHistorico].[dbo].[copiacovidh]	
Output List	
[covidHistorico].[dbo].[copiacovidh].ENTIDAD_RES,	
[covidHistorico].[dbo].[copiacovidh].MUNICIPIO_RES	

Con índice:



Consulta 2:

```
select ENTIDAD_RES, MUNICIPIO_RES, count(*) as reportados, count(case CLASIFICACION_FINAL when 1 then CLASIFICACION_FINAL
when 2 then CLASIFICACION_FINAL
when 3 then CLASIFICACION_FINAL
end) as confirmado,
count(case CLASIFICACION_FINAL when 6 then CLASIFICACION_FINAL end) as sospechoso
from dbo.datoscovid
group by ENTIDAD_RES, MUNICIPIO_RES
order by ENTIDAD_RES, reportados desc
```



Análisis:

Siguiendo con la tendencia de enfocarme en la primera consulta, (aunque considero el uso del case para esta consulta un tanto más práctico) hemos pasado del uso de dos table scan simultáneos con costos en el rango de 400 a index seeks de 2 y 21 aproximadamente, gracias al índice de clasificación final con entidad y municipio de residencia

4. Determinar el municipio con el mayor número de defunciones en casos confirmados.

Consulta propia:

```
select top 1 ENTIDAD_RES, MUNICIPIO_RES, count(*) as Difuntos from dbo.datoscovid
where FECHA_DEF != '9999-99-99' and CLASIFICACION_FINAL between 1 and 3
group by ENTIDAD_RES, MUNICIPIO_RES
order by Difuntos desc
```

Sin índice:

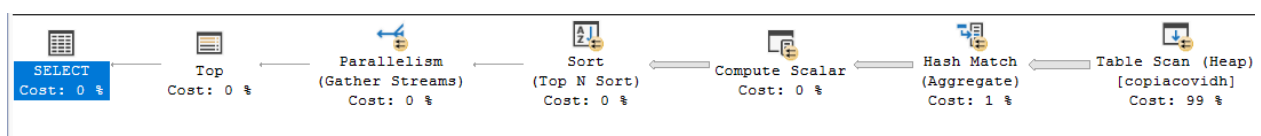
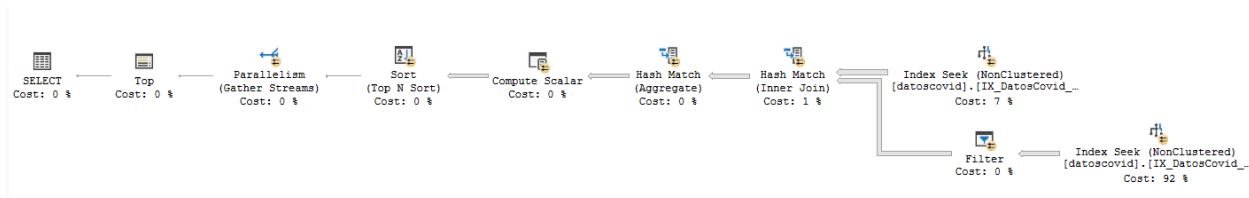


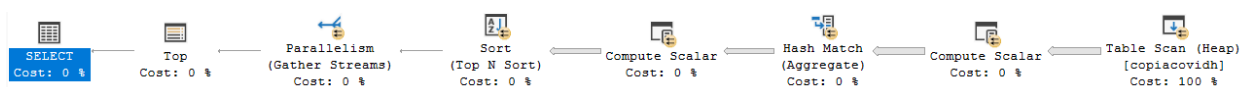
Table Scan (Heap)	
Scan rows from a table.	
Physical Operation	Table Scan
Logical Operation	Table Scan
Estimated Execution Mode	Batch
Storage	RowStore
Estimated Operator Cost	406.658 (99%)
Estimated I/O Cost	403.859
Estimated Subtree Cost	406.658
Estimated CPU Cost	2.79872
Estimated Number of Executions	1
Estimated Number of Rows to be Read	15265600
Estimated Number of Rows for All Executions	286192
Estimated Number of Rows Per Execution	286192
Estimated Row Size	49 B
Ordered	False
Node ID	5
Predicate	
[covidHistorico].[dbo].[copiacovidh].[CLASIFICACION_FINAL]>=(1) AND [covidHistorico].[dbo].[copiacovidh].[CLASIFICACION_FINAL] <=(3) AND [covidHistorico].[dbo].[copiacovidh].[FECHA_DEF] <>'9999-99-99'	
Object	
[covidHistorico].[dbo].[copiacovidh]	
Output List	
[covidHistorico].[dbo].[copiacovidh].ENTIDAD_RES, [covidHistorico].[dbo].[copiacovidh].MUNICIPIO_RES	

Con índice:



Consulta 2:

```
Select TOP 1 ENTIDAD_RES,MUNICIPIO_RES, count(case CLASIFICACION_FINAL when 1 then CLASIFICACION_FINAL
when 2 then CLASIFICACION_FINAL
when 3 then CLASIFICACION_FINAL
end) as Confirmados_fallecidos
from dbo.datoscovid
where FECHA_DEF != '9999-99-99'
GROUP BY ENTIDAD_RES,MUNICIPIO_RES
ORDER BY Confirmados_fallecidos DESC;
```



Análisis:

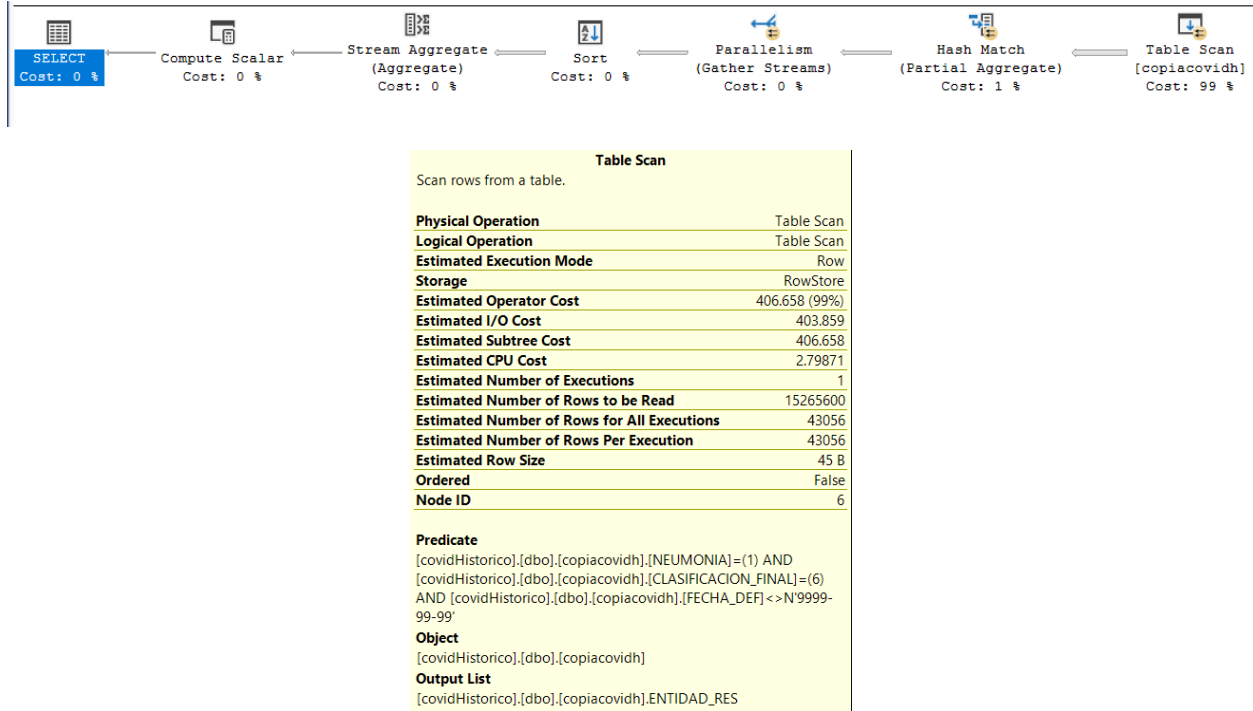
Otra consulta que se ve beneficiada de la inclusión de los índices que se crearon, a pesar de a simple vista aparentar que el plan de ejecución se vio afectado por tener más componentes, pero analizándola podemos apreciar que los costos de todos esos pasos juntos resultan menores al table scan que realiza la consulta sin índices

5. Determinar por entidad, si de casos sospechosos hay defunciones reportadas asociadas a neumonía.

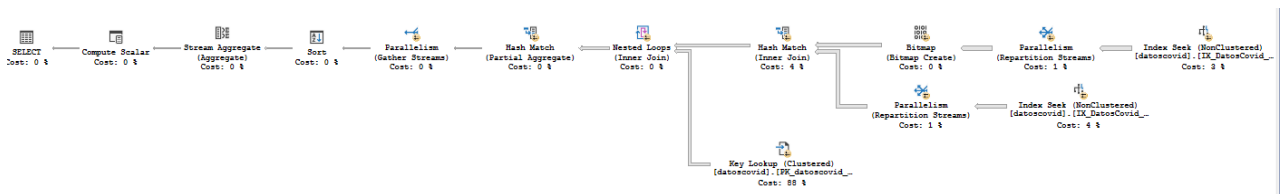
Consulta propia:

```
select ENTIDAD_RES, count(*) from dbo.datoscovid
where neumonia=1 and CLASIFICACION_FINAL=6 and FECHA_DEF != '9999-99-99'
group by ENTIDAD_RES
```

Sin índice:

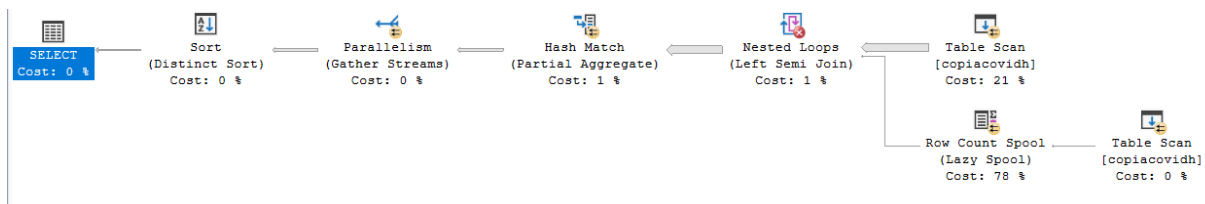


Con índice:



Consulta 2:

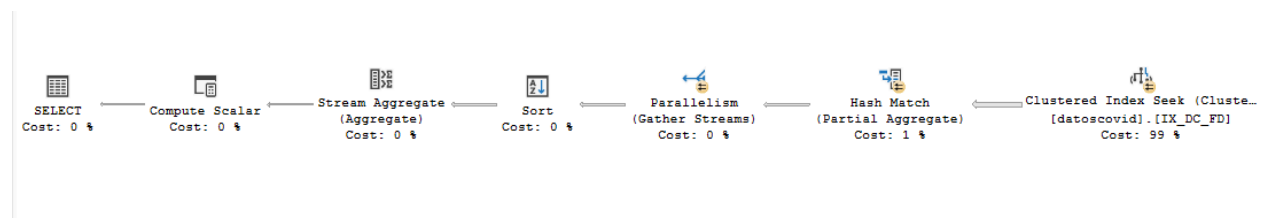
```
SELECT DISTINCT ENTIDAD_RES from dbo.datoscovid
where EXISTS(Select *from dbo.datoscovid Where NEUMONIA = 1 and FECHA_DEF != '9999-99-99')
GROUP BY ENTIDAD_RES
```



Análisis:

Otra consulta en la que los índices juegan a nuestro favor en este caso la fecha de defunción, que podría ser optimizado cambiándolo por la fecha de defunción en conjunto con la entidad de residencia como en el de clasificación final, en cuanto a la clave de neumonía sería innecesario agregar un índice en esa columna, con todo y las deficiencias encontradas, el índice establecido nos sigue proporcionando un coste menor de ejecución al table scan por defecto

Sobre esta consulta decidí implementar el CLUSTERED_INDEX, eliminando así el índice agrupado anterior que estaba sobre los ID pero sobre ninguna consulta se vio utilizado, decidí crear un índice compuesto por las columnas de FECHA_DEF y CLASIFICACION_FINAL, esto basado en la cantidad de veces que se utilizan dichas columnas y por las ventajas del índice agrupado que particularmente afectan en este caso respecto a que se va a organizar todas las fechas de defunción, o al menos ese era el plan inicial, pero después de investigar más y realizar pruebas, se llega a que no se puede añadir una columna para un índice agrupado, por lo que el índice agrupado se construye únicamente sobre FECHA_DEF e inmediatamente podemos ver una mejora sobre el plan de ejecución y los costos de ejecución como se muestra a continuación:



Costo:

Clustered Index Seek (Clustered)	
Scanning a particular range of rows from a clustered index.	
Physical Operation	Clustered Index Seek
Logical Operation	Clustered Index Seek
Estimated Execution Mode	Row
Storage	RowStore
Estimated Operator Cost	11.3541 (99%)
Estimated I/O Cost	11.2787
Estimated Subtree Cost	11.3541
Estimated CPU Cost	0.0754477
Estimated Number of Executions	1
Estimated Number of Rows Per Execution	37374.2
Estimated Number of Rows to be Read	411390
Estimated Number of Rows for All Executions	37374.2
Estimated Row Size	23 B
Ordered	True
Node ID	6
Predicate	
[covidHistorico].[dbo].[datoscovid].[NEUMONIA]=(1) AND	
[covidHistorico].[dbo].[datoscovid].[CLASIFICACION_FINAL]=(6)	
Object	
[covidHistorico].[dbo].[datoscovid].[IX_DC_FD]	
Output List	
[covidHistorico].[dbo].[datoscovid].ENTIDAD_RES	
Seek Predicates	
[1] Seek Keys[1]: End: [covidHistorico].[dbo].[datoscovid].FECHA_DEF <	
Scalar Operator(N'9999-99-99'), [2] Seek Keys[1]: Start: [covidHistorico].	
[dbo].[datoscovid].FECHA_DEF > Scalar Operator(N'9999-99-99')	

6. Listar por entidad el total de casos sospechosos, casos confirmados, total de defunciones en los meses de marzo a agosto 2020 y de diciembre 2020 a mayo 2021.

Consulta propia:

```
select ENTIDAD_RES, count(*) as difuntos, count(case CLASIFICACION_FINAL when 1 then CLASIFICACION_FINAL
                                                    when 2 then CLASIFICACION_FINAL
                                                    when 3 then CLASIFICACION_FINAL end)
as confirmado, count(case CLASIFICACION_FINAL when 6 then CLASIFICACION_FINAL end) as sospechoso
from dbo.datoscovid
where (FECHA_DEF between '2020-03-01' and '2020-08-01') or (FECHA_DEF between '2020-12-01' and '2021-05-01')
GROUP BY ENTIDAD_RES
```

Sin índice:

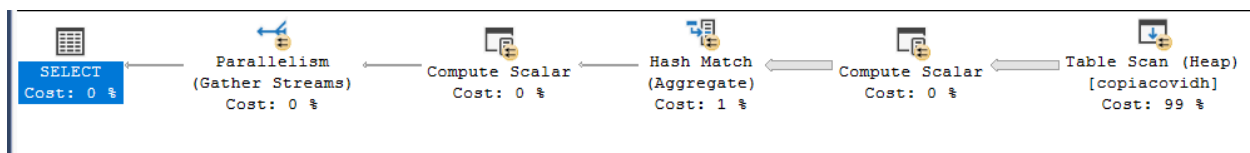
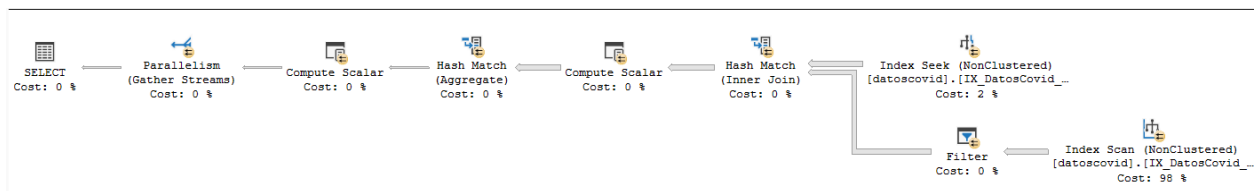


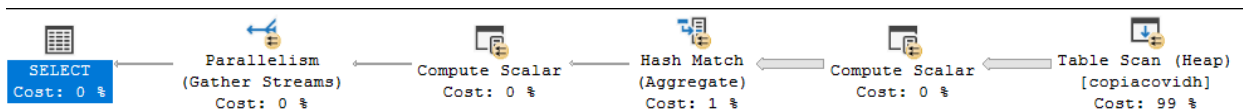
Table Scan (Heap)	
Scan rows from a table.	
Physical Operation	Table Scan
Logical Operation	Table Scan
Estimated Execution Mode	Batch
Storage	RowStore
Estimated Operator Cost	406.658 (99%)
Estimated I/O Cost	403.859
Estimated Subtree Cost	406.658
Estimated CPU Cost	2.79872
Estimated Number of Executions	1
Estimated Number of Rows to be Read	15265600
Estimated Number of Rows for All Executions	234495
Estimated Number of Rows Per Execution	234495
Estimated Row Size	41 B
Ordered	False
Node ID	4
Predicate	
[covidHistorico].[dbo].[copiacovidh].[FECHA_DEF]>=N'2020-03-01' AND [covidHistorico].[dbo].[copiacovidh].[FECHA_DEF]<=N'2020-08-01' OR [covidHistorico].[dbo].[copiacovidh].[FECHA_DEF]>=N'2020-12-01' AND [covidHistorico].[dbo].[copiacovidh].[FECHA_DEF]<=N'2021-05-01'	
Object	
[covidHistorico].[dbo].[copiacovidh]	
Output List	
[covidHistorico].[dbo].[copiacovidh].ENTIDAD_RES, [covidHistorico].[dbo].[copiacovidh].CLASIFICACION_FINAL	

Con índice:



Consulta 2:

```
select ENTIDAD_RES, count(*) as fallecidos, count(case CLASIFICACION_FINAL when 1 then CLASIFICACION_FINAL
when 2 then CLASIFICACION_FINAL
when 3 then CLASIFICACION_FINAL
end) as confirmado,
count(case CLASIFICACION_FINAL when 6 then CLASIFICACION_FINAL end) as sospechoso
from dbo.datoscovid
where (FECHA_DEF between '2020-03-01' and '2020-08-31') or (FECHA_DEF between '2020-12-01' and '2021-05-31')
GROUP BY ENTIDAD_RES
```



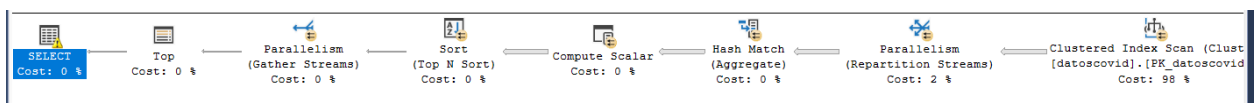
Análisis:

Otra consulta optimizable si junto dos de los índices que tiene la tabla (FECHA_DEF y [Clasificación_final con ENTIDAD y MUNICIPIO]), para reducir uno de los pasos que realiza el plan de ejecución, debido a que ya hay índices con esas columnas el costo que representa esta consulta es bajo, sumando los dos index scan obtenemos un valor menor a 6`

7. Listar los 5 municipios con el mayor número de casos confirmados en niños menos de 13 años con alguna comorbilidad reportada y cuantos de esos casos fallecieron.

Consulta 1:

```
select top 5 ENTIDAD_RES, MUNICIPIO_RES, COUNT(*) as dcom
from dbo.datoscovid
where EDAD<13 and FECHA_DEF !='9999-99-99' and (DIABETES=1 OR epoc=1 or asma=1 or INMUSUPR=1 or HIPERTENSION=1 or OTRA_COM=1)
group by ENTIDAD_RES, MUNICIPIO_RES order by dcom desc
```



Análisis:

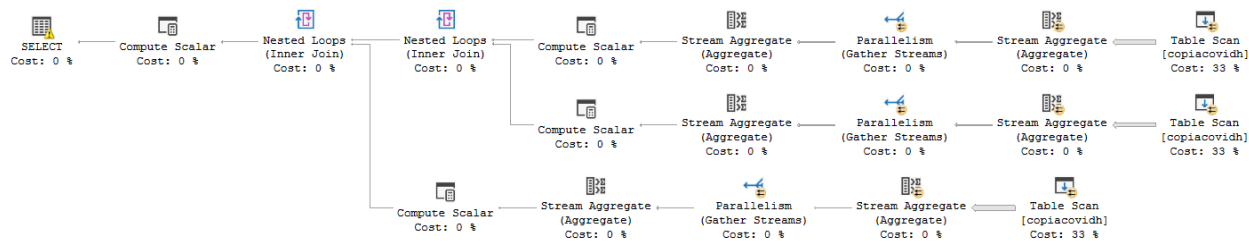
La consulta 1 es como comprendí comorbilidad inicialmente, después de una serie de intentos de combinaciones de padecimientos, no logré llegar a una solución satisfactoria a la consulta

8. Determinar si en el año 2020 hay una mayor cantidad de defunciones menores de edad que en el año 2021 y 2022.

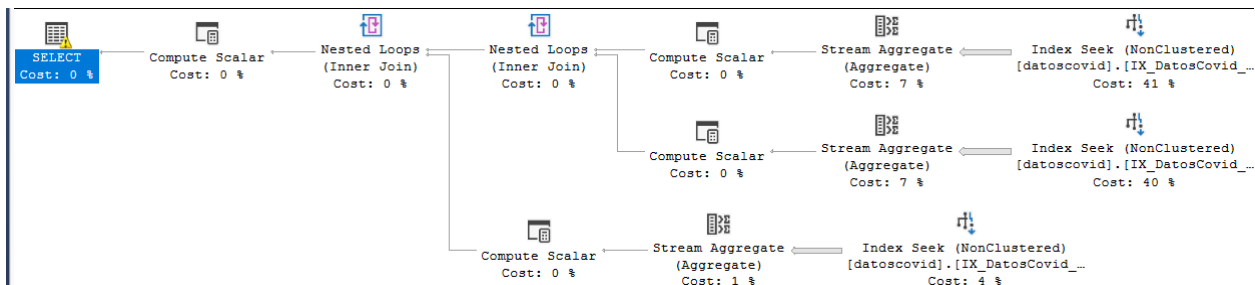
Consulta propia:

```
select
case when sqrd20.DIFUNTOS2020>sqrd21.DIFUNTOS2021+sqrd22.DIFUNTOS2022 then 'fue mayor'
else 'no fue mayor'
end
from (select count(*) as DIFUNTOS2020 from dbo.datoscovid where edad<18 and FECHA_DEF between '2020-01-01' and '2020-12-31') sqrd20,
(select count(*) as DIFUNTOS2021 from dbo.datoscovid where edad<18 and FECHA_DEF between '2021-01-01' and '2021-12-31') sqrd21,
(select count(*) as DIFUNTOS2022 from dbo.datoscovid where edad<18 and FECHA_DEF between '2022-01-01' and '2022-12-31') sqrd22
```

Sin índice:

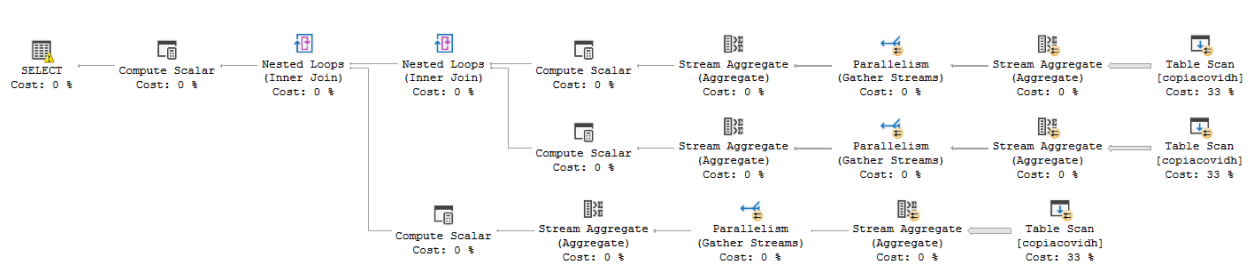


Con índice:



Consulta 2

```
select Defunciones_2020.d_20,Defunciones_2021.d_21,Defunciones_2022.d_22,
case when Defunciones_2020.d_20>Defunciones_2021.d_21+Defunciones_2022.d_22 then 'Es mayor a 2021+2022'
else 'Es menor a 2021+2022'
end as vs_SUMA_2122, case when Defunciones_2020.d_20>Defunciones_2021.d_21 then 'Es mayor a 2021'
else 'Es menor a 2021'
end as vs_2021, case when Defunciones_2020.d_20>Defunciones_2022.d_22 then 'Es mayor a 2022'
else 'Es menor a 2022'
end as vs_2022
from (select count(*) as d_20 from dbo.datoscovid where edad<18 and FECHA_DEF between '2020-01-01' and '2020-12-31') Defunciones_2020,
(select count(*) as d_21 from dbo.datoscovid where edad<18 and FECHA_DEF between '2021-01-01' and '2021-12-31') Defunciones_2021,
(select count(*) as d_22 from dbo.datoscovid where edad<18 and FECHA_DEF between '2022-01-01' and '2022-12-31') Defunciones_2022
```



Análisis:

Originalmente con los índices anteriores se continuaba con el alto costo de ejecución de esta consulta, lo que me llevo a desarrollar un índice adicional, este índice resulta útil para todas las consultas que necesiten relacionar la edad con la fecha de defunción (que además cumple la función de determinar si alguien falleció o no)

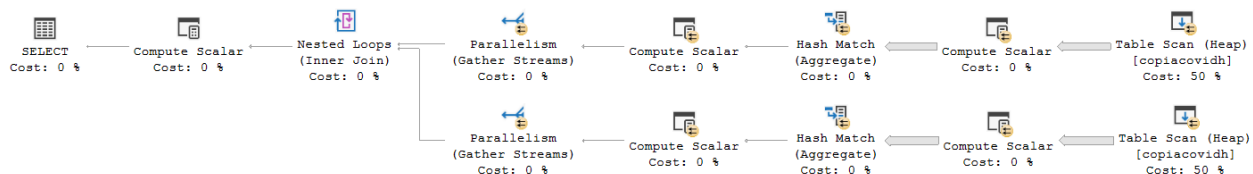
```
CREATE NONCLUSTERED INDEX [IX_DatosCovid_EDAD_DEF]
ON [dbo].[datoscovid] ([FECHA_DEF])
INCLUDE ([EDAD])
```

9. Determinar si en el año 2021 hay un porcentaje mayor al 60 de casos reportados que son confirmados por estudios de laboratorio en comparación al año 2020.

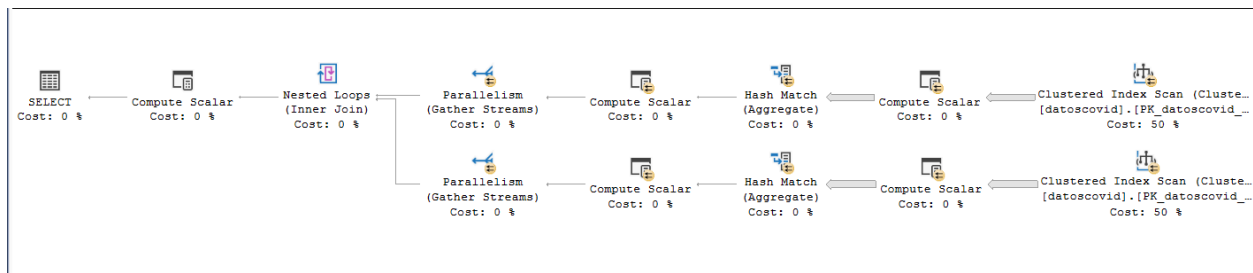
Consulta :

```
Select t1.Confirmados_2020,t2.Confirmados_2021,(((t2.Confirmados_2021-t1.Confirmados_2020)*100)/Confirmados_2020)as Porcentaje
from(select count(case CLASIFICACION_FINAL when 1 then CLASIFICACION_FINAL
when 2 then CLASIFICACION_FINAL
when 3 then CLASIFICACION_FINAL
end) as Confirmados_2020
from dbo.datoscovid where FECHA_SINTOMAS between '2020-01-01' and '2020-12-31') as t1,
(select count(case CLASIFICACION_FINAL when 1 then CLASIFICACION_FINAL
when 2 then CLASIFICACION_FINAL
when 3 then CLASIFICACION_FINAL
end) as Confirmados_2021
from dbo.datoscovid where FECHA_SINTOMAS between '2021-01-01' and '2021-12-31') as t2
```

Sin índice:



Con índice:



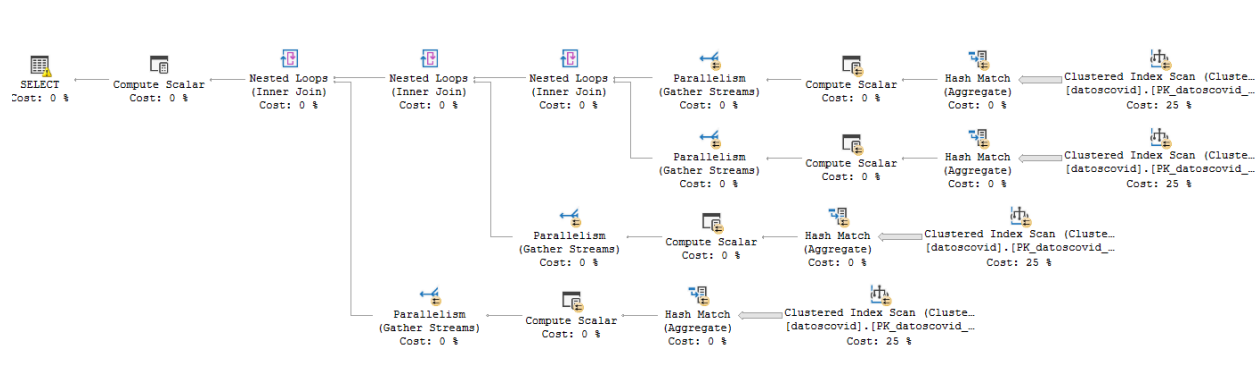
Análisis:

En este caso la consulta no se ve beneficiada de ningún índice, previo por lo que considero que la solución sería reescribir la consulta haciendo uso de subconsultas, en este caso, no pude llegar a dicho resultado de una consulta más optimizada.

10. Determinar en qué rango de edad: menor de edad, 19 a 40, 40 a 60 o mayor de 60 hay más casos reportados que se hayan recuperado.

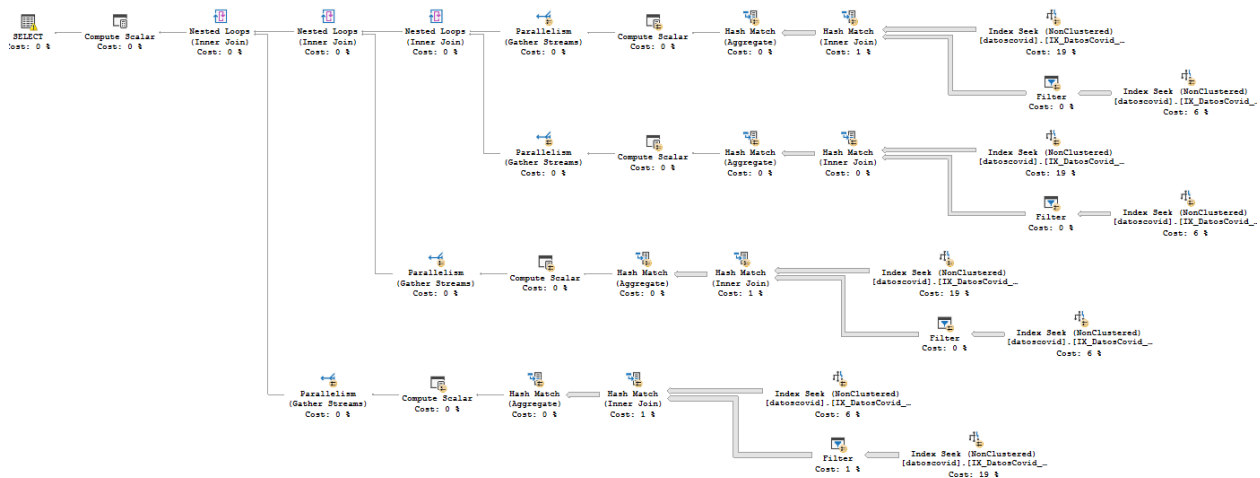
Consulta propia:

```
select
case when ran1.Recuperado>ran2.Recuperado and ran1.Recuperado>ran3.Recuperado and ran1.Recuperado>ran4.Recuperado
then 'El range de menor de edad tiene mas recuperados'
when ran2.Recuperado>ran1.Recuperado and ran2.Recuperado>ran3.Recuperado and ran2.Recuperado>ran4.Recuperado
then 'El rango de 18 a 40 tiene mas recuperados'
when ran3.Recuperado>ran1.Recuperado and ran3.Recuperado>ran2.Recuperado and ran3.Recuperado>ran4.Recuperado
then 'El rango de 18 a 40 tiene mas recuperados'
when ran4.Recuperado>ran1.Recuperado and ran4.Recuperado>ran2.Recuperado and ran4.Recuperado>ran3.Recuperado
then 'El rango de 18 a 40 tiene mas recuperados'
else 'error'
end
from
(select count(*) as Recuperado from dbo.datoscovid where FECHA_DEF = '9999-99-99' and (CLASIFICACION_FINAL= '1'
or CLASIFICACION_FINAL = '3') and EDAD<18) ran1,
(select count(*) as Recuperado from dbo.datoscovid where FECHA_DEF = '9999-99-99' and (CLASIFICACION_FINAL= '1'
or CLASIFICACION_FINAL = '3') and EDAD between '18' and '40') ran2,
(select count(*) as Recuperado from dbo.datoscovid where FECHA_DEF = '9999-99-99' and (CLASIFICACION_FINAL= '1'
or CLASIFICACION_FINAL = '3') and EDAD between '41' and '60' ) ran3,
(select count(*) as Recuperado from dbo.datoscovid where FECHA_DEF = '9999-99-99' and (CLASIFICACION_FINAL= '1'
or CLASIFICACION_FINAL = '3') and EDAD >'61' ) ran4
```



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	Batch
Storage	RowStore
Estimated Operator Cost	406.657 (25%)
Estimated I/O Cost	403.858
Estimated Subtree Cost	406.657
Estimated CPU Cost	2.79872
Estimated Number of Executions	1
Estimated Number of Rows to be Read	15265600
Estimated Number of Rows for All Executions	1035200
Estimated Number of Rows Per Execution	1035200
Estimated Row Size	39 B
Ordered	False
Node ID	12
Predicate [covidHistorico].[dbo].[datoscovid].[FECHA_DEF]=N'9999-99-99' AND [covidHistorico].[dbo].[datoscovid].[EDAD]>N'61' AND ([covidHistorico]. [dbo].[datoscovid].[CLASIFICACION_FINAL]=(1) OR [covidHistorico]. [dbo].[datoscovid].[CLASIFICACION_FINAL]=(3))	
Object [covidHistorico].[dbo].[datoscovid].[PK_datoscovid_ID_Registro]	

Con índice:

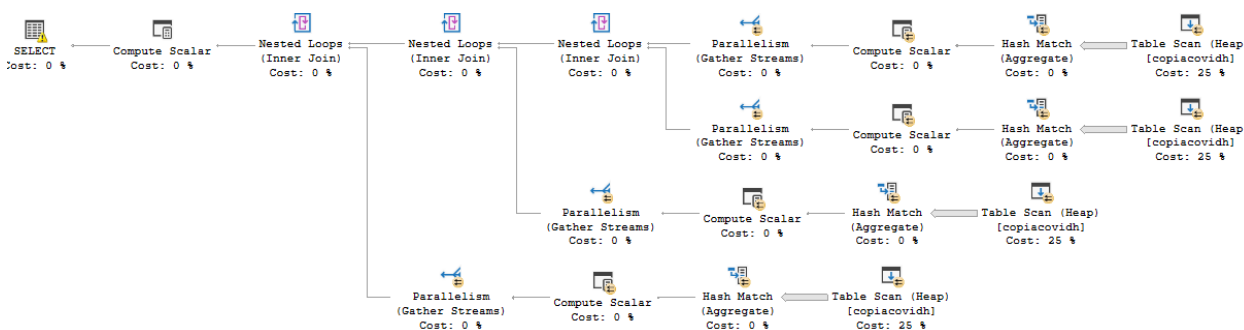


Consulta 2:

```

select
  case when R1.P_recuperadas>R2.P_recuperadas and R1.P_recuperadas>R3.P_recuperadas and R1.P_recuperadas>R4.P_recuperadas
    then 'Menores de edad se recuperan mas'
  when R2.P_recuperadas>R1.P_recuperadas and R2.P_recuperadas>R3.P_recuperadas and R2.P_recuperadas>R4.P_recuperadas
    then '19 a 40 años se recuperan mas'
  when R3.P_recuperadas>R1.P_recuperadas and R3.P_recuperadas>R2.P_recuperadas and R3.P_recuperadas>R4.P_recuperadas
    then '41 a 60 años se recuperan mas'
  when R4.P_recuperadas>R1.P_recuperadas and R4.P_recuperadas>R2.P_recuperadas and R3.P_recuperadas>R4.P_recuperadas
    then 'Mayores de 60 se recuperan mas'
end as Supervivencia
from
  (select count(*) as P_recuperadas from dbo.datoscovid where FECHA_DEF = '9999-99-99' and (CLASIFICACION_FINAL= '1'
  or CLASIFICACION_FINAL = '3') and (EDAD<=18) AND TIPO_PACIENTE!=2) R1,
  (select count(*) as P_recuperadas from dbo.datoscovid where FECHA_DEF = '9999-99-99' and (CLASIFICACION_FINAL= '1'
  or CLASIFICACION_FINAL = '3') and (EDAD between '19' and '40') AND TIPO_PACIENTE!=2) R2,
  (select count(*) as P_recuperadas from dbo.datoscovid where FECHA_DEF = '9999-99-99' and (CLASIFICACION_FINAL= '1'
  or CLASIFICACION_FINAL = '3') and (EDAD between '41' and '60') AND TIPO_PACIENTE!=2 ) R3,
  (select count(*) as P_recuperadas from dbo.datoscovid where FECHA_DEF = '9999-99-99' and (CLASIFICACION_FINAL= '1'
  or CLASIFICACION_FINAL = '3') and (EDAD >'60') AND TIPO_PACIENTE!=2) R4

```



Análisis:

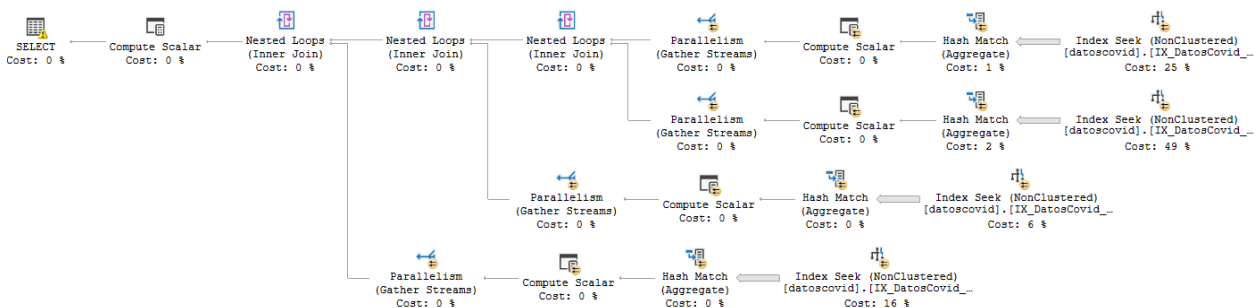
Una consulta que se ve afectada por los índices anteriores, agregando complejidad, ya que hay índices con similitud a lo que esta consulta necesita para reducir sus costos, basado en lo visto en consultas anteriores se llega implementar el siguiente índice:

```

CREATE NONCLUSTERED INDEX [IX_DatosCovid_DEF_EDAD_CF]
ON [dbo].[datoscovid] ([FECHA_DEF],[EDAD],[CLASIFICACION_FINAL])

```

Que nos da el siguiente plan de ejecución:



Como podemos observar reduce el plan a una forma más digerible (similar a cuando no tiene índices la tabla), y gracias al nuevo índice nos encontramos con costo total menor a 200 aproximadamente

Conclusiones

En este ejercicio fue un poco más compleja la decisión de establecer índices iniciales que ayuden a lograr resultados optimizados para las soluciones que se llegaron de las consultas, debido a la forma en la que los mismos están almacenados en conjunto y la adición de un documento externo en el cual se tenían que consultar el significado de algunos valores de las filas, resultó una tarea con mayor dificultad en comparación con las antes vistas en clase, esto principalmente a que la información se manejaba en distintas tablas, por lo que las diferencias eran más notorias a simple vista, resulto interesante el proponer optimizaciones a través de los índices, ya que también ayudan a encontrar fallas en las consultas propuestas