Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas

16-6-2022

EQUIPO 1

- CORDOVA GALVÁN ROBERTO.
- PEDROZA VELARDE LUIS RODRIGO.
- TORRES SÁNCHEZ EDGAR.

Profesor:
CARLOS DE LA CRUZ SOSA.

3TM3

PRACTICA 5: PLANES DE EJECUCIÓN.

Consultas para realizar:

/*Práctica de optimización de consultas

Listado de consultas a programar para analizar planes de ejecución

1. Listar los casos positivos por entidad de residencia

2.-Listar los casos sospechosos por entidad

3.- Listar el Top 5 de municipios por entidad con el mayor numero

de casos reportados, indicando casos sospechosos y casos confirmados

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

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

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.

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.

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

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

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.

*/

Solución.

- Primero hicimos una copia de la tabla original, en la cual no existía ningún tipo de llave.
- Se implemento un índice agrupado en la columna entidad_res, esto con la finalidad se optimizar las consultas que teníamos sobre la tabla original
- Se analizaron y compararon los nuevos planes de ejecución con los anteriores.

```
use covidHistorico;
select* into copiacovid from datoscovid
create clustered index idx_entidad_res on dbo.copiacovid(entidad_res);
```

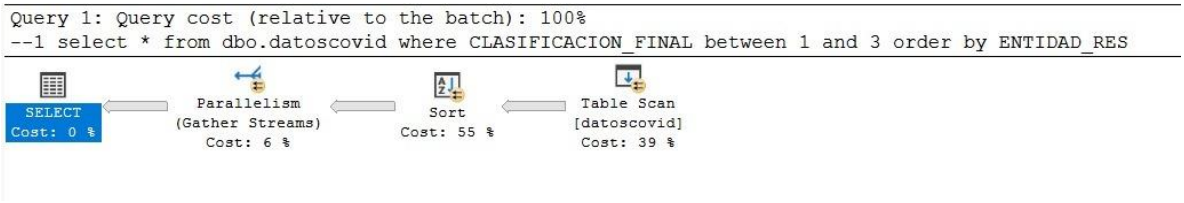
Consultas Principales:

```
--1

select * from dbo.datoscovid where

CLASIFICACION_FINAL between 1 and 3

order by ENTIDAD_RES;
```

Resultado de la Consulta.

	FECHA_ACTUALIZACION	ID_REGISTRO	ORIGEN	SECTOR	ENTIDAD_UM	SEXO	ENTIDAD_NAC	ENTIDAD_RES	MUNICIPIO_RES	TIPO_PACIENTE	FECHA_INGRESO	FECHA_SINTOMAS	FECHA_DEF	INTUBADO	NEUMONIA	EDAD	NACIONALIDA
1	2022-03-08	1b5816	2	12	01	2	01	01	007	1	2020-07-01	2020-06-30	9999-99-99	97	2	7	1
2	2022-03-08	3e9118	1	4	01	2	01	01	001	1	2020-12-01	2020-11-28	9999-99-99	97	2	55	1
3	2022-03-08	a8de63	1	4	01	2	01	01	001	1	2021-07-27	2021-07-21	9999-99-99	97	2	18	1
4	2022-03-08	83b0a0	2	12	01	1	01	01	001	1	2022-01-03	2021-12-30	9999-99-99	97	2	13	1
5	2022-03-08	4d4823	2	4	01	1	01	01	001	1	2022-01-13	2022-01-09	9999-99-99	97	2	43	1
6	2022-03-08	45963a	1	12	32	1	01	01	002	2	2020-11-29	2020-11-28	9999-99-99	2	2	62	1
7	2022-03-08	39bfb7	2	12	01	2	01	01	001	1	2020-11-30	2020-11-20	9999-99-99	97	2	45	1
8	2022-03-08	67e2a7	1	6	01	2	01	01	001	2	2022-01-26	2022-01-22	9999-99-99	2	1	64	1
9	2022-03-08	0acd28	2	4	01	2	01	01	001	1	2020-07-01	2020-06-28	9999-99-99	97	2	27	1
10	2022-03-08	143690	1	4	01	2	01	01	001	1	2020-08-28	2020-08-20	9999-99-99	97	2	28	1
11	2022-03-08	d5f13f	1	4	01	1	01	01	001	1	2021-07-27	2021-07-25	9999-99-99	97	2	24	1
12	2022-03-08	ca1c59	2	4	01	2	01	01	001	1	2022-01-06	2022-01-06	9999-99-99	97	2	34	1
13	2022-03-08	49b98f	2	12	01	1	09	01	001	1	2022-01-20	2022-01-17	9999-99-99	97	2	60	1

Plan de ejecución.



Lo primero que se realiza en este plan de ejecución es un recorrido de toda la tabla, para esto se usa un Table Scan, como la tabla carece de índices se buscan las filas que se están solicitando, si se agrega un índice es posible que se logre hacer que la consulta se realice en menos tiempo. Esta acción tiene un coste de 39% del total de la consulta ya que lee más de 15 millones de registros.

Posteriormente se muestra una operación de tipo sort, que está dada en la consulta por la instrucción order by, la columna que se requiere ser ordenada es ENTIDAD_RES la cual carece de índice por lo cual la instrucción de orden lleva más tiempo y por lo tanto baja el rendimiento de la consulta por lo cual su coste es de 55% del total.

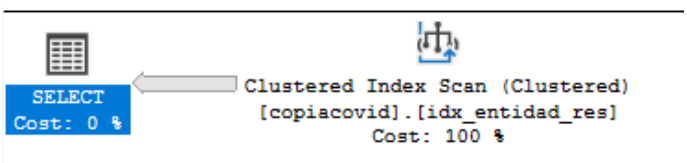
La parte de parallelism consume múltiples flujos de entrada, los une y los entrega en una sola salida. El coste de esta operación es de 6%.

Finalmente, el select se encarga de mostrar los datos.

Resultado con índice Agrupado.

	FECHA_ACTUALIZACION	ID_REGISTRO	ORIGEN	SECTOR	ENTIDAD_UM	SEXO	ENTIDAD_NAC	ENTIDAD_RES	MUNICIPIO_RES	TIPO_PACIENTE	FECHA_INGRESO	FECHA_SINTOMAS	FECHA
1	2022-03-08	b072b0	2	12	01	1	01	01	005	1	2021-06-30	2021-06-25	9999-9
2	2022-03-08	ea0bfb	2	12	01	1	01	01	005	1	2021-06-30	2021-06-25	9999-9
3	2022-03-08	5323a5	1	4	01	1	01	01	001	1	2021-06-29	2021-06-28	9999-9
4	2022-03-08	63095e	2	4	01	2	15	01	005	1	2021-06-30	2021-06-28	9999-9
5	2022-03-08	ec9ee3	2	4	01	2	01	01	001	1	2021-06-30	2021-06-28	9999-9
6	2022-03-08	a21c64	2	9	09	2	01	01	001	1	2021-04-13	2021-04-12	9999-9

Plan de ejecución.

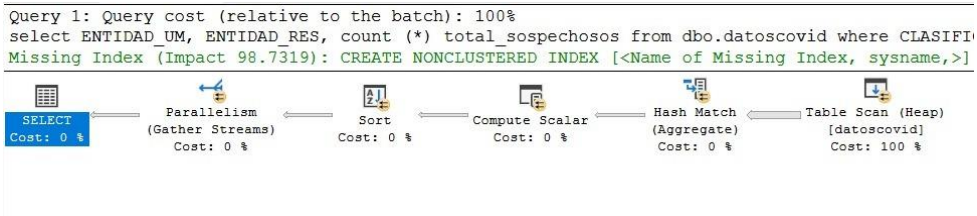


```
--2

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;
```

Resultado de la Consulta.

	ENTIDAD_UM	ENTIDAD_RES	total_sospechosos
1	01	07	1
2	01	03	1
3	01	28	1
4	01	24	1
5	01	21	2
6	01	15	7
7	01	30	1
8	01	32	39
9	01	19	1
10	01	16	2
11	01	10	1
12	01	23	1
13	01	26	1
14	01	27	1

Plan de ejecución.



En esta consulta tenemos un table scan ya que el costo es del 100% y los datos los muestra de una manera ordenada que es por entidad municipal y además no hace uso de un índice. El costo es del 100% ya que lee todos los datos y además lo muestra de una manera ordenada.

Después tenemos un operador hash match aquí aparece ya que al hacer la cuenta de los totales sospechosos y al ser una gran cantidad de datos que tiene que leer para hacer la cuenta optimiza la cuenta para no demorar mucho. El costo es del 0% ya que al crear una tabla temporal no consume muchos recursos

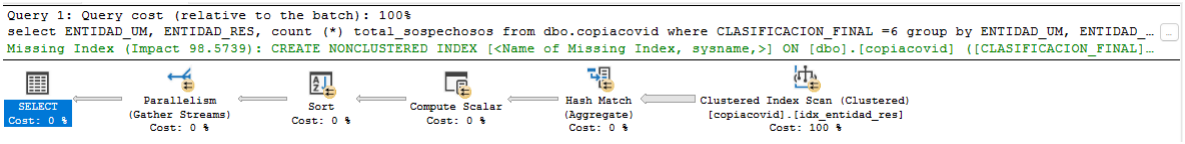
El computer scalar hace las operaciones para hacer los cálculos requeridos y poder realizar las operaciones correspondientes

Al tener un order by nos aparece una operación de tipo sort ya que ordenamos por la tabla ENTIDAD_UM pero este tiene un costo del 0% ya que al no tratarse de una gran cantidad de datos ya que estan siendo filtrados no consume muchos recursos.

Resultado con índice Agrupado.

	ENTIDAD_UM	ENTIDAD_RES	total_sospechosos
1	01	16	2
2	01	07	1
3	01	19	1
4	01	09	4
5	01	26	1
6	01	15	7
7	01	23	1
8	01	12	1

Plan de ejecución.



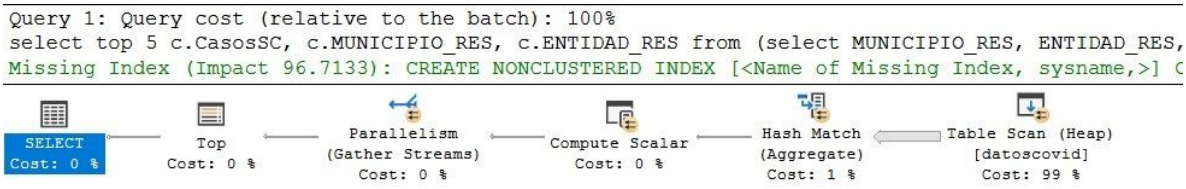
3--

```
select top 5 c.CasosSC, c.MUNICIPIO_RES, c.ENTIDAD_RES from (select MUNICIPIO_RES, ENTIDAD_RES,
count(CLASIFICACION_FINAL) CasosSC from dbo.datoscovid

where CLASIFICACION_FINAL between 1 and 3 or CLASIFICACION_FINAL=6

group by ENTIDAD_RES, MUNICIPIO_RES) as c;
```

Resultado de la Consulta.

	CasosSC	MUNICIPIO_RES	ENTIDAD_RES
1	137621	012	09
2	213	023	32
3	7024	024	24
4	58	006	28
5	11	107	21

Plan de ejecución.



En este caso se realiza un table scan, la operación tiene un coste grande en la consulta ya que los datos que se analizan fila por fila no se encuentran ordenados y además no se tiene un indice, esto genera que la consulta tarde un tiempo considerable para procesar los datos, su coste es de 99% del total de la consulta ya que lee demasiados datos.

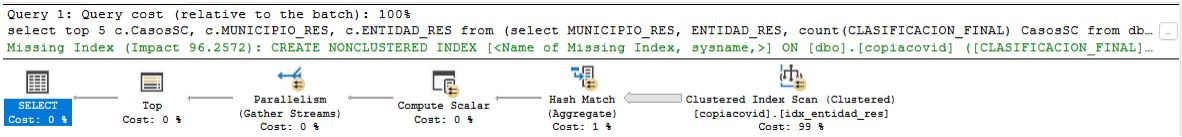
A continuación, tenemos una operación Hash Match (Aggregate), esta operación se utiliza porque se procesan tablas muy largas que no están ordenadas y que no usan un índice, se genera una tabla en la memoria, calcula el valor hash para cada registro y escanea todos los demás registros para buscar el hash. Así se asegura que solo hay un registro para cada grupo de datos, además optimiza y hace más rápida la función de agregación que se encuentra en la consulta, la cual sería el count. Esta operación tiene un coste de 1%.

Compute scalar es una operación que calcula nuevos valores a partir de un valor de fila existente mediante una operación.

Resultado con índice Agrupado.

	CasosSC	MUNICIPIO_RES	ENTIDAD_RES
1	347	007	04
2	625	077	07
3	22093	101	07
4	218	050	07
5	1388	007	01

Plan de ejecución.



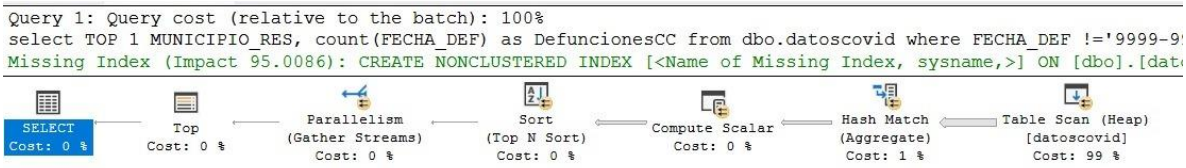
4--

```
select TOP 1 MUNICIPIO_RES, count(FECHA_DEF) as DefuncionesCC from dbo.datoscovid

where FECHA_DEF !='9999-99-99' and CLASIFICACION_FINAL between 1 and 3

group by MUNICIPIO_RES order by DefuncionesCC desc;
```

Resultado de la Consulta.

	MUNICIPIO_RES	DefuncionesCC
1	039	14256

Plan de ejecución.



En esta consulta aparece un table scan ya que al procesar igualmente una gran cantidad de datos lo lee fila por fila y al mostrar solamente el top 1 filtra los datos de las filas y muestra el que es mayor con todos y además no usa índices de ningún tipo y el costo es del 99% porque lee una gran cantidad de datos.

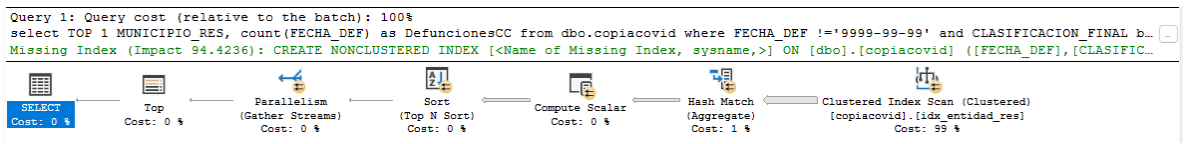
Al generar un índice temporal para la lectura de datos, aparece un operador de tipo hash match y su costo es del 1% y al estar contando los registros por la fecha de defunción y hacer una filtración entre el 1 y el 3 hace que los datos sean más rápidos de leer y se optimice mejor la consulta

El sort aparece ya que organizamos los datos por el municipio de residencia.

Resultado con índice Agrupado.

	MUNICIPIO_RES	DefuncionesCC
1	039	14256

Plan de ejecución.

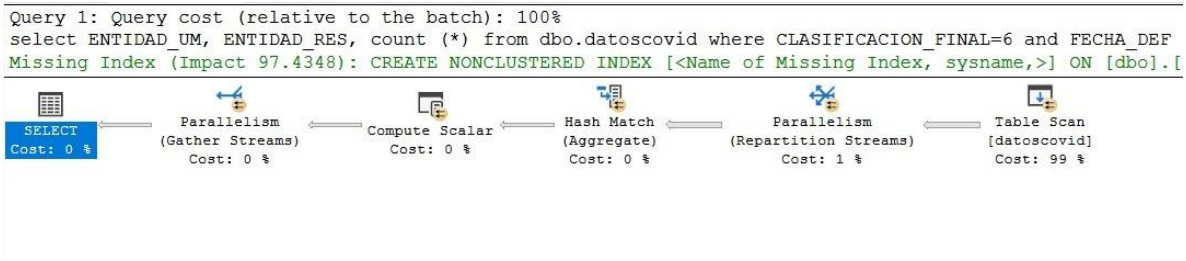


```
--5

select ENTIDAD_UM, ENTIDAD_RES, count (*)

from dbo.datoscovid

where CLASIFICACION_FINAL=6 and FECHA_DEF! = '9999-99-99' and NEUMONIA=1

group by ENTIDAD_UM, ENTIDAD_RES
```

Resultado de la Consulta.

	ENTIDAD_UM	ENTIDAD_RES	(No column name)
1	05	05	248
2	27	20	1
3	11	11	103
4	24	32	1
5	08	10	1
6	14	06	1
7	09	12	16
8	09	22	6
9	23	23	30
10	26	25	1
11	08	08	193
12	14	18	1
13	14	23	1
14	15	27	1

Plan de ejecución.



Nuevamente se realiza una operación de recorrido en toda la tabla, donde no hay un índice que haga más fácil la operación, para encontrar los datos que necesitamos y que se requieren para la consulta. Este tiene un 99% de coste del total de la consulta ya que es una operación que es bastante pesada.

El parallelism (Repartition Streams) se encarga de intercambiar filas entre diferentes flujos para optimizar la consulta que deseamos, podría ser para adaptarse a los operadores que siguen. Tiene un coste del 1%.

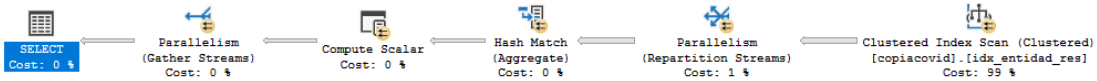
La operación Hash Match en esta ocasión optimiza la función de agregación que se encuentra en la consulta (count), con ayuda del Compute Scalar que calcula nuevos valores a partir de las filas ya existentes se genera una salida.

Resultado con índice agrupado

	ENTIDAD_UM	ENTIDAD_RES	(No column name)
1	30	27	1
2	01	32	1
3	15	17	2
4	09	21	23
5	20	30	1
6	29	13	1
7	09	20	8
8	01	14	2
9	32	32	2
10	10	10	25
11	16	16	74
12	19	08	1
13	30	20	1
14	15	12	1
15	05	32	1
16	20	20	72
17	25	24	1

Plan de ejecución

```
Query 1: Query cost (relative to the batch): 100%
select ENTIDAD_UM, ENTIDAD_RES, count (*) from dbo.copiacovid where CLASIFICACION_FINAL=6 and FECHA_DEF! = '9999-99-99' and NEUMO
Missing Index (Impact 97.1262): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[copiacovid] ([NEUMONIA],[
```

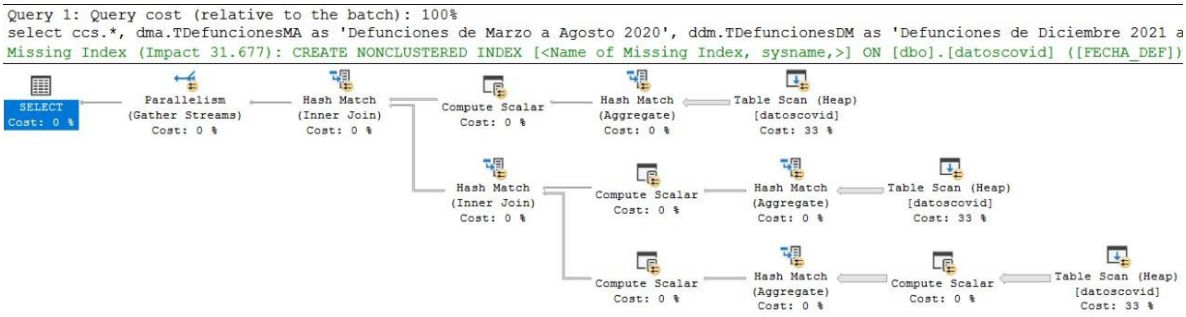


```
--6

select ccs.*, dma.TDefuncionesMA as 'Defunciones de Marzo a Agosto 2020',

ddm.TDefuncionesDM as 'Defunciones de Diciembre 2021 a Mayo 2021' from

(select ENTIDAD_RES,

--Casos Confirmados

count(case CLASIFICACION_FINAL when 1 then CLASIFICACION_FINAL

when 2 then CLASIFICACION_FINAL

when 3 then CLASIFICACION_FINAL

end) as Confirmados,

--Casos Sospechosos

count(case CLASIFICACION_FINAL when 6 then CLASIFICACION_FINAL end)as sospechosos

from dbo.datoscovid group by ENTIDAD_RES) ccs

inner join

--Casos confirmados de MARZO A AGOSTO 2020

(select ENTIDAD_RES, count(FECHA_DEF) TDefuncionesMA from dbo.datoscovid where FECHA_DEF between
'2020-03-01' and '2020-08-31' group by ENTIDAD_RES)

dma

on dma.ENTIDAD_RES=ccs.ENTIDAD_RES

join

--Casos confirmados de DICIEMBRE 2020 A MAYO 2021

(select ENTIDAD_RES, count(FECHA_DEF) TDefuncionesDM from dbo.datoscovid where FECHA_DEF between
'2020-12-01' and '2021-05-31' group by ENTIDAD_RES) ddm

on dma.ENTIDAD_RES=ddm.ENTIDAD_RES;
```

Resultado de la Consulta.

	ENTIDAD_RES	Confirmados	sospechosos	Defunciones de Marzo a Agosto 2020	Defunciones de Diciembre 2021 a Mayo 2021
1	10	67737	4397	737	1355
2	01	61705	10855	613	1766
3	17	66703	7764	1388	2917
4	03	101286	1940	601	942
5	22	140248	4626	990	3258
6	21	167075	19710	5519	7459
7	12	97719	3391	2527	2388
8	18	57733	9166	830	1087
9	06	54225	3159	568	604
10	19	309540	14707	3796	6699
11	32	67332	4876	752	1634
12	11	277113	21517	3596	7689
13	14	234286	20370	4096	9092
14	02	130194	41021	4716	4491

Plan de ejecución.



Aquí tenemos a 2 procesos hijos ya que están siendo separados por los casos confirmados, casos sospechosos y casos confirmados de diciembre 2020 y mayo del 2021. Los tres hijos contienen un table scan con un costo del 33% cada uno ya que al ser 3 se divide el total del 100%.

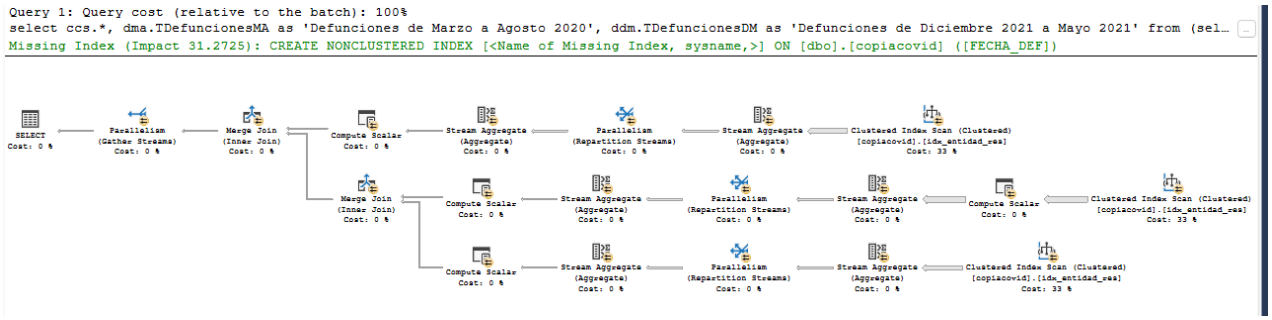
De la misma manera los 2 contienen un hash match optimiza los procesos hijos y no se demoren mucho en realizar las operaciones.

Después tenemos un hash match (inner join), este aparece ya que une las diferentes consultas que se hacen en los procesos hijos y las muestras en una sola.

Resultado con índice agrupado

	ENTIDAD_RES	Confirmados	sospechosos	Defunciones de Marzo a Agosto 2020	Defunciones de Diciembre 2021 a Mayo 2021
1	11	277113	21517	3596	7689
2	01	61705	10855	613	1766
3	10	67737	4397	737	1355
4	15	528556	95922	18666	23562
5	28	141284	6962	3575	2672
6	05	143702	12276	2302	3573
7	22	140248	4626	990	3258
8	27	189393	9187	3259	1397
9	31	108481	14540	2517	1983

Plan de ejecución.

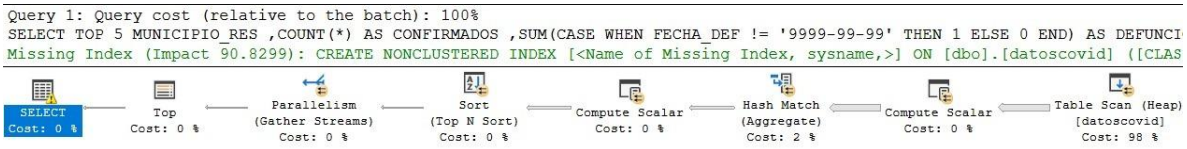


```
--7

SELECT TOP 5

         MUNICIPIO_RES

        ,COUNT(*) AS CONFIRMADOS

        ,SUM(CASE WHEN FECHA_DEF != '9999-99-99' THEN 1 ELSE 0 END) AS DEFUNCIONES

FROM dbo.datoscovid

WHERE (EDAD <= 13 AND CLASIFICACION_FINAL BETWEEN 1 AND 3)              -- Casos confirmados menores
de 13

AND (   DIABETES = 1 OR
        -- Comorbilidades

                EPOC = 1 OR

                ASMA = 1 OR

                INMUSUPR = 1 OR

                HIPERTENSION = 1 OR

                OTRA_COM = 1 OR

                OBESIDAD = 1 OR

                RENAL_CRONICA = 1)

GROUP BY MUNICIPIO_RES

ORDER BY CONFIRMADOS DESC;
```

Resultado de la Consulta.

	MUNICIPIO_RES	CONFIRMADOS	DEFUNCIONES
1	005	674	8
2	002	643	10
3	010	607	6
4	007	590	13
5	003	461	9

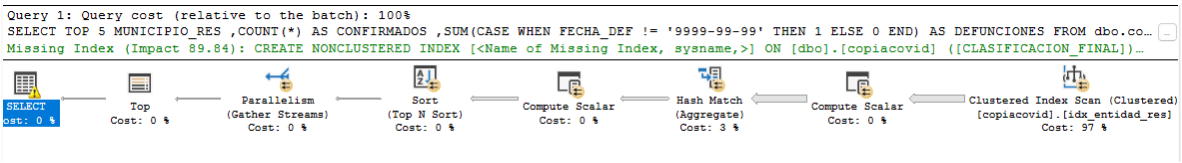
Plan de ejecución.



Resultado con índice agrupado

	MUNICIPIO_RES	CONFIRMADOS	DEFUNCIONES
1	005	674	8
2	002	643	10
3	010	607	6
4	007	590	13
5	003	461	9

Plan de ejecución

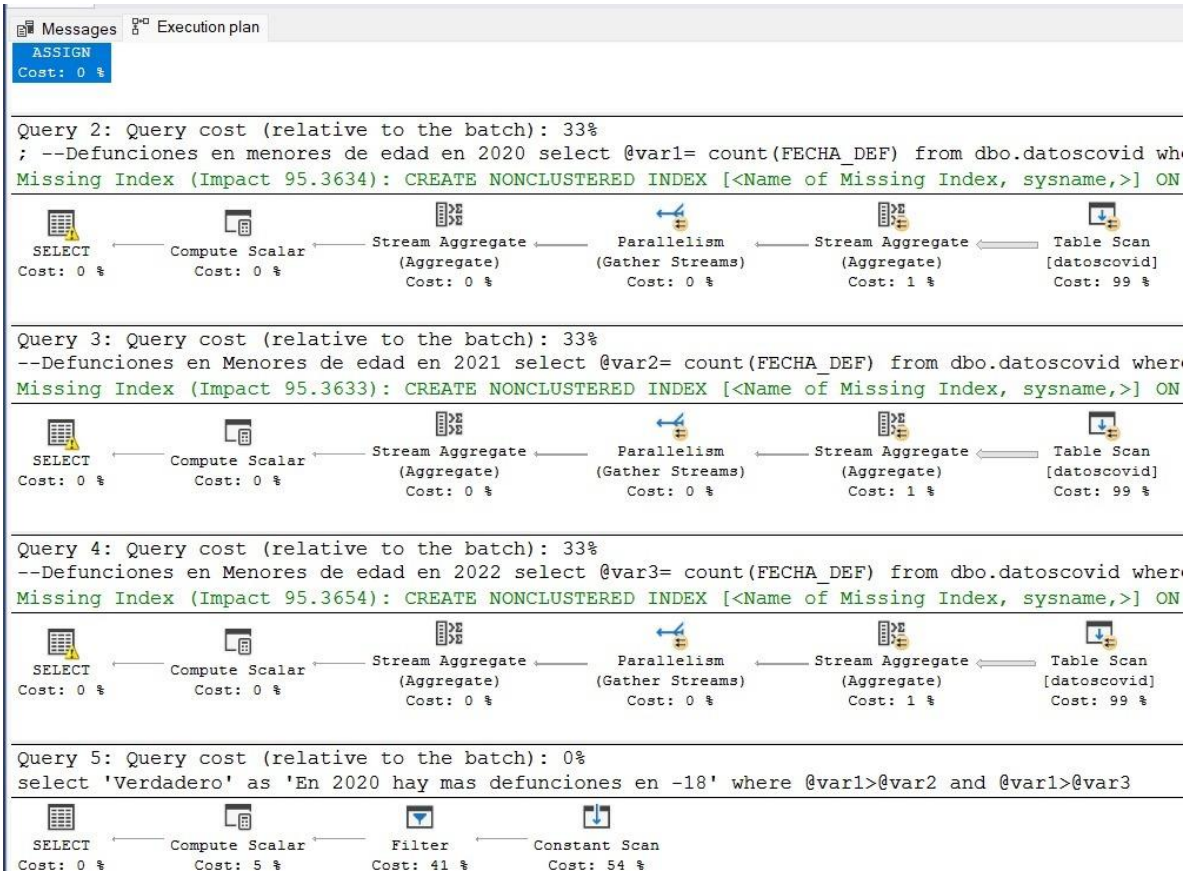


```
--8

DECLARE @var1 int, @var2 int, @var3 int;

select @var1=1, @var2=1, @var3=1;

--Defunciones en menores de edad en 2020

select @var1= count(FECHA_DEF) from dbo.datoscovid where edad<18 and FECHA_DEF between '2020-01-01'
and '2020-12-31'

--Defunciones en Menores de edad en 2021

select @var2= count(FECHA_DEF) from dbo.datoscovid where edad<18 and FECHA_DEF between '2021-01-01'
and '2021-12-31'

--Defunciones en Menores de edad en 2022

select @var3= count(FECHA_DEF) from dbo.datoscovid where edad<18 and FECHA_DEF between '2022-01-01'
and '2022-12-31'

select 'Verdadero' as 'En 2020 hay mas defunciones en -18' where @var1>@var2 and @var1>@var3;
```

Resultado de la Consulta.

	En 2020 hay mas defunciones en -18
1	Verdadero

Plan de ejecución.



En esta consulta tenemos tres querys diferentes ejecutándose, primero declaramos 3 variables de tipo

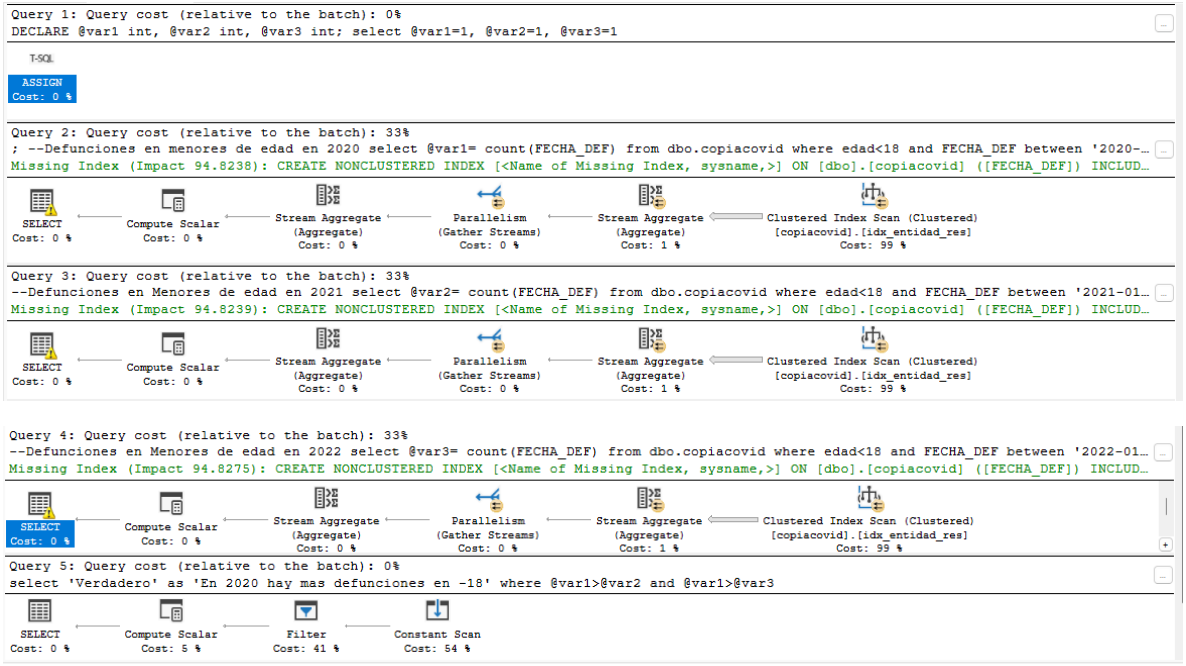
entero donde estas almacenaran el conteo de los difuntos. Las 3 consultas contienen un table scan que lo hacen de la tabla datoscovid.

Las tres cuentan con un operador stream aggregate, lo que hace es agregar un operador físico y espera a que agregues datos en las filas, tiene un costo del 1% porque optimiza la consulta y además espera a que termine las comparaciones para agregarlas a la tabla.

Solución con índice Agrupado.

	En 2020 hay mas defunciones en -18
1	Verdadero

Plan de Ejecución.

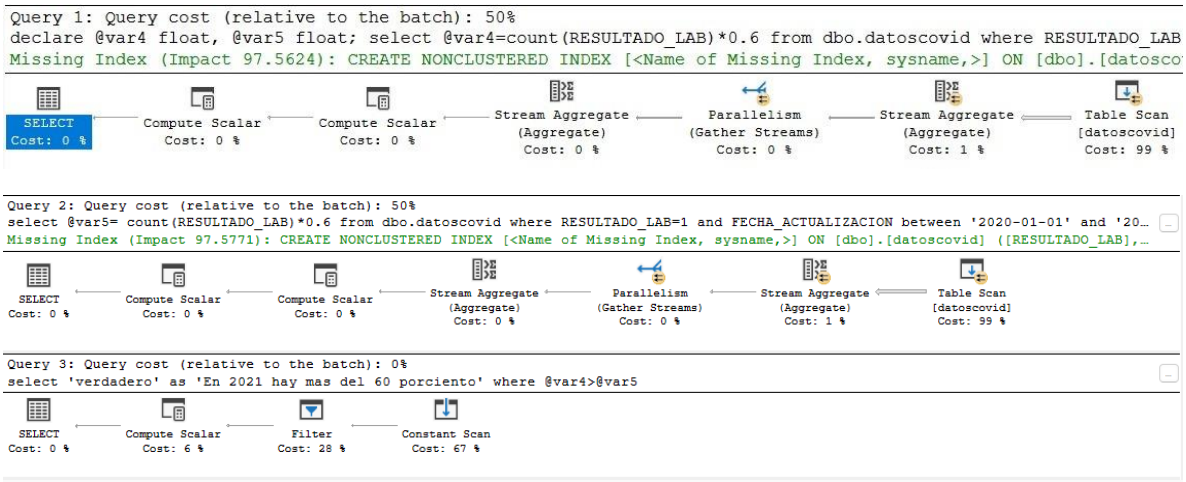


```
--9

declare @var4 float, @var5 float

select @var4=count(RESULTADO_LAB)*0.6 from dbo.datoscovid where RESULTADO_LAB =1  and
FECHA_ACTUALIZACION between '2021-01-01' and '2021-12-31'

select @var5= count(RESULTADO_LAB)*0.6 from dbo.datoscovid where RESULTADO_LAB=1 and
FECHA_ACTUALIZACION between '2020-01-01' and '2020-12-32'

select 'verdadero' as 'En 2021 hay más del 60 porciento' where @var4>@var5;
```

Resultado de la Consulta.

En 2021 hay mas del 60 porciento

Plan de ejecución.



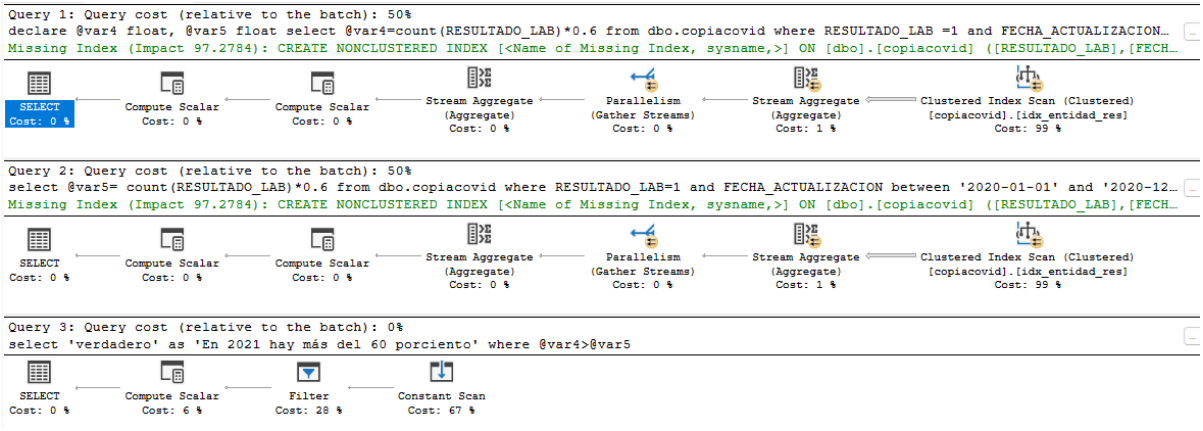
En la forma que se realiza la consulta, se obtienen distintos planes de ejecución. Al usarse variables en donde se almacenan los resultados de consultas realizadas se obtienen dos planes de ejecución parecidos, los cuales utilizan un table scan para analizar todos los datos de la tabla, posteriormente la función de Stream Aggregate (aggregate) calcula valores de resumen para grupos de filas, el parallelism produce un solo flujo de salida, y una vez más se calculan valores de resumen para poder calcular nuevos valores a través de otras filas con compute scalar y finalmente se asignan a una variable.

Para la parte final de la consulta donde se comparan las variables el plan de ejecución muestra un Constant Scan con un coste de 67% que consulta una tabla de constantes, después nos encontramos un filtro con un coste de 28%, este se encarga de realizar las comparaciones entre las variables, finalmente el compute scalar calcula nuevos valores a partir de otros para mostrar la consulta.

Solución con índice agrupado.

En 2021 hay más del 60 porciento

Plan de ejecución.

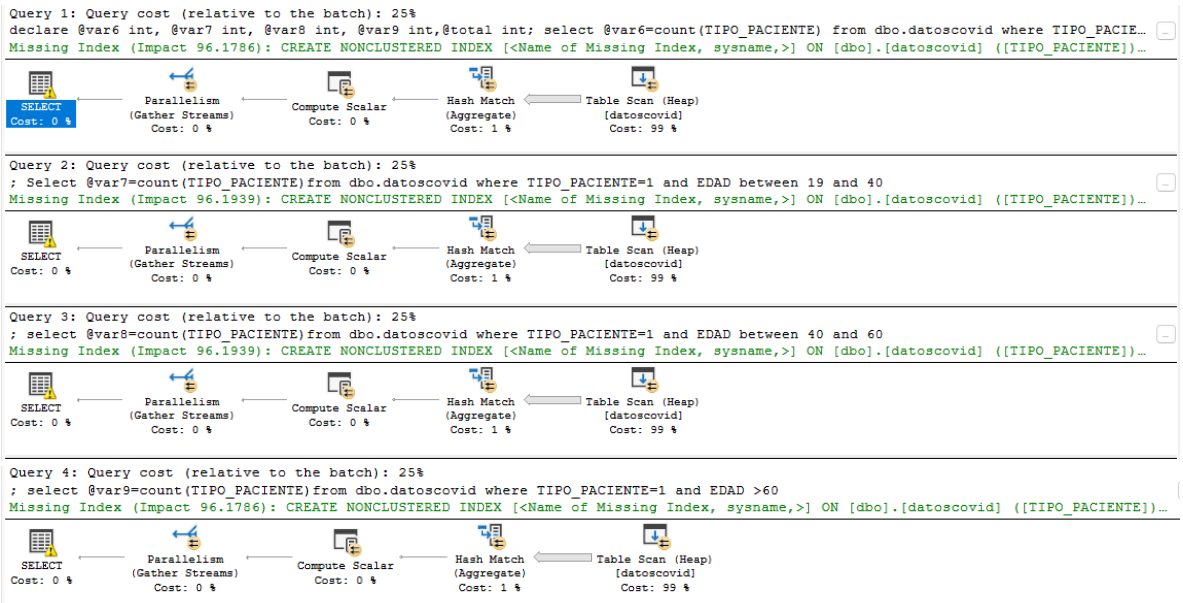


```
--10

declare @var6 int, @var7 int, @var8 int, @var9 int,@total int;

select @var6=count(TIPO_PACIENTE) from dbo.datoscovid where TIPO_PACIENTE=1 and EDAD < 18;

Select @var7= count(TIPO_PACIENTE)from dbo.datoscovid where TIPO_PACIENTE=1 and EDAD between 19 and
40;

select @var8= count(TIPO_PACIENTE)from dbo.datoscovid where TIPO_PACIENTE=1 and EDAD between 40 and
60;

select @var9= count(TIPO_PACIENTE)from dbo.datoscovid where TIPO_PACIENTE=1 and EDAD >60;

IF @var6>@var7 and @var6>@var8 and @var6>@var9
begin
print 'Casos reportados y recuperados: Menores de edad'
end else
if @var7>@var6 and @var7>@var8 and @var7>@var9
begin
print 'Casos reportados y recuperados: Entre 19 y 40 años'
end else
if @var8>@var6 and @var8>@var7 and @var8>@var9
begin
print 'Casos reportados y recuperados: entre 40 y 60 años'
end else
if @var9>@var6 and @var9>@var7 and @var9>@var8
begin
print 'Casos reportados y recuperados: Mayores de 60 años'
end;
```

Resultado de la Consulta.

```
Casos reportados y recuperados: Entre 19 y 40 años

Completion time: 2022-06-07T11:39:10.2259971-05:00
```

Plan de ejecución.



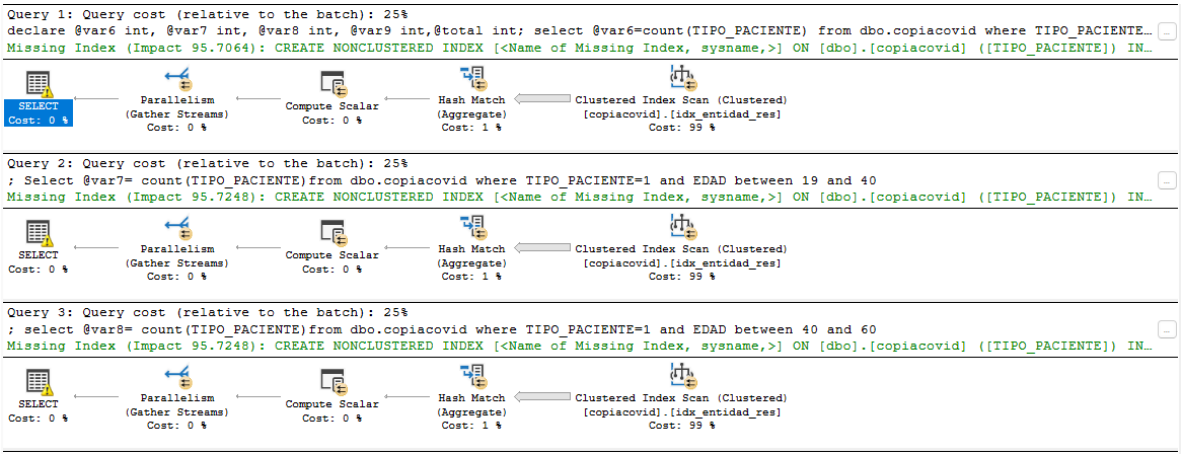
Declaramos 3 variables para almacenar el conteo de los casos, después el table scan hace ese escaneo de las tablas, el hash match optimiza los cálculos y las comparaciones que tiene que hacer para que sean más rápidas y mostrar los resultados.

El computer scalar optimiza los cálculos y el parallelism manda esos datos al select para que sean mostrados.

Resultado con índice agrupado.

```
Casos reportados y recuperados: Entre 19 y 40 años

Completion time: 2022-06-16T10:19:47.4434328-05:00
```

Plan de ejecución.



PLANES DE EJECUCIÓN ALTERNATIVOS.

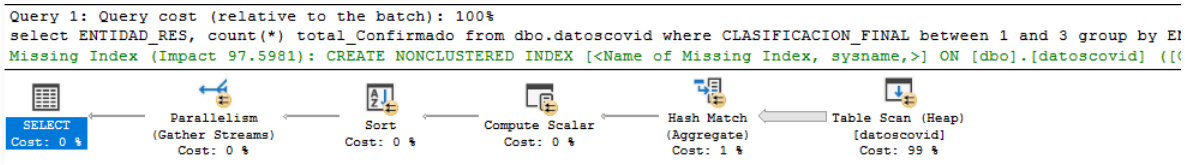
--1

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

Resultado de la Consulta.

ENTIDAD_RES	total_Confirmado
01	61705
02	130194
03	101286
04	33489
05	143702
06	54225
07	34746
08	125965
09	1366421
10	67737
11	277113
12	97719
13	89808
14	234286
15	528556
16	93199
17	66703
18	57733
19	309540
20	117996

Plan de ejecución.



En esta consulta tenemos una operación Table scan que analiza todas las filas de la tabla para devolver las que necesita el usuario, se hace de este modo ya que no hay un índice que sea útil para agilizar el proceso de búsqueda, esta operación conlleva el 99% del total de la consulta.

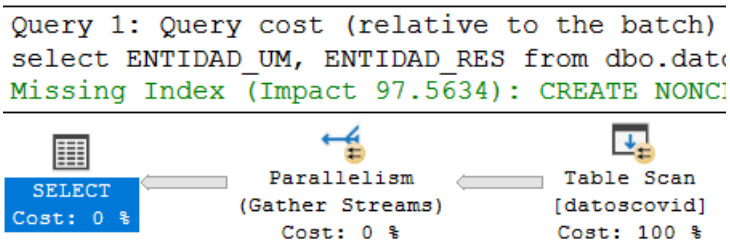
A continuación, tenemos una operación hash match (aggregate) la cual consume 1% del coste total de la consulta, aquí ayuda a que la parte del count para que sea más eficiente y rápida. Posteriormente tenemos un compute scalar que apoya a la parte de calcular valores a partir de otros registros, un sort que ordenará los datos de la consulta y un parallelism que une todos los datos para ser mostrados.

2— select ENTIDAD_UM, ENTIDAD_RES from dbo.datoscovid where CLASIFICACION_FINAL=6;

Resultado de la Consulta.

	ENTIDAD_UM	ENTIDAD_RES
1	09	09
2	31	31
3	19	19
4	01	01
5	02	02
6	19	19
7	07	07
8	19	19
9	01	01
10	15	15

Plan de ejecución.



Esta consulta tiene un table scan que tiene el costo del 100% ya que escanea los datos filtrados y no los ordena de ningún modo y el parallelism manda los datos que solicita la consulta.

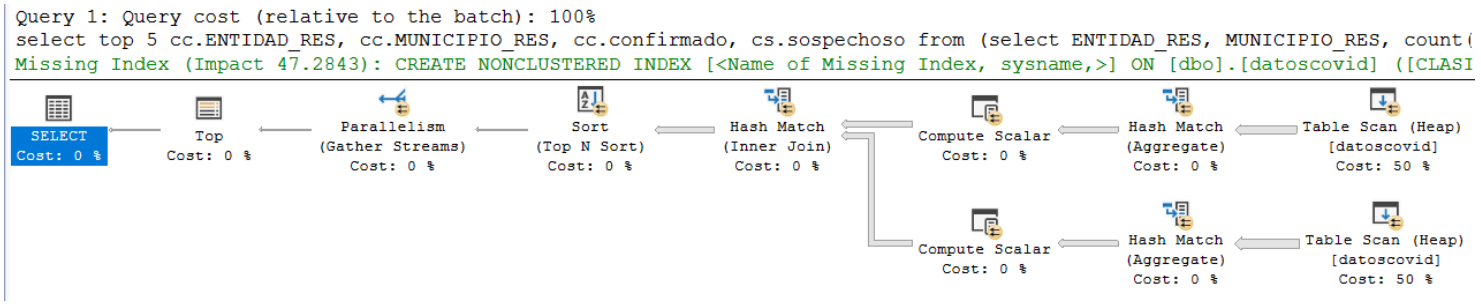
El select es el que muestra el resultado.

```
--3

select top 5 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;
```

Resultado de la Consulta.

	ENTIDAD_RES	MUNICIPIO_RES	confirmado	sospechoso
1	01	006	1682	62
2	01	003	1574	32
3	01	011	535	381
4	01	010	224	55
5	01	007	1289	99

Plan de Ejecución.



Aquí tenemos dos procesos hijos donde los dos tiene un table scan de la tabla datos covid, como son dos se dividen el costo con 50% cada uno. los dos procesos cuentan con un hash match donde al estarlos ordenando por la entidad de residencia y el municipio de residencia.

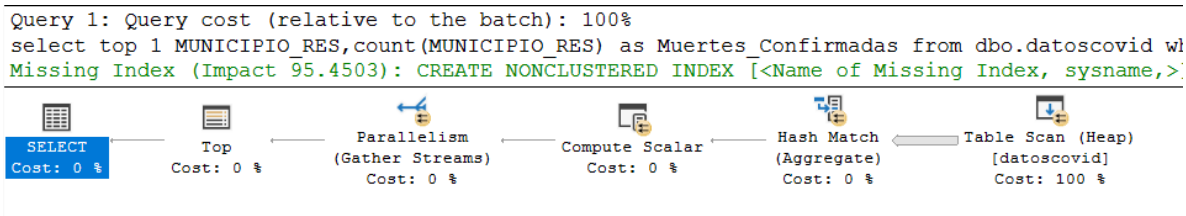
Despues tenemos el hash match del inner join, este aparece ya que al hacer la unión de las tablas para mostrar la consulta y es por esto por lo que se divide en dos procesos ya que al ser unión de tablas hace las consultas en ambas y hace el orden en que deben ir.

```
--Solucion 4

select top 1 MUNICIPIO_RES,count(MUNICIPIO_RES) as Muertes_Confirmadas from dbo.datoscovid where
FECHA_DEF!='9999-99-99' group by MUNICIPIO_RES
```

Resultado de la Consulta.

MUNICIPIO_RES	Muertes_Confirmadas
114	10328

Plan de ejecución.



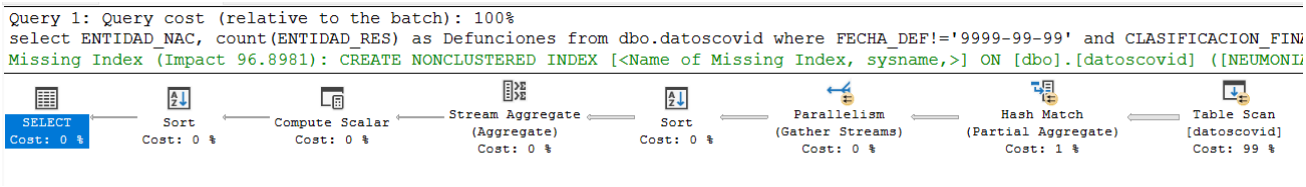
Este resultado cuenta con un table scan donde escanea la tabla y con el hash match hace las comparaciones y los filtrados que requiere para mostrar el resultado, el parallelism toma los diferentes flujos y genera una sola salida, finalmente la instrucción TOP muestra el dato que esta por encima del resto de los resultados.

```
--Solucion 5

select ENTIDAD_NAC, count(ENTIDAD_RES) as Defunciones from dbo.datoscovid where FECHA_DEF!='9999-99-
99' and CLASIFICACION_FINAL=6 and NEUMONIA=1 group by ENTIDAD_NAC order by Defunciones
```

Resultado de la Consulta.

ENTIDAD_NAC	Defunciones
06	7
23	9
03	12
04	22
22	27
18	31
99	35
31	40
32	43
01	46
29	55
10	69
28	74
17	81
24	81
19	93
02	100
27	109
13	117
26	121
08	162
12	180
11	189
16	208

Plan de ejecución.



En esta consulta tenemos un table scan de datos COVID que tiene un costo del 99%, este tiene el costo ya que escanea todas las filas y columnas de la tabla para mostrar la información que quiere ver el usuario y así agilizar el proceso de la búsqueda de datos

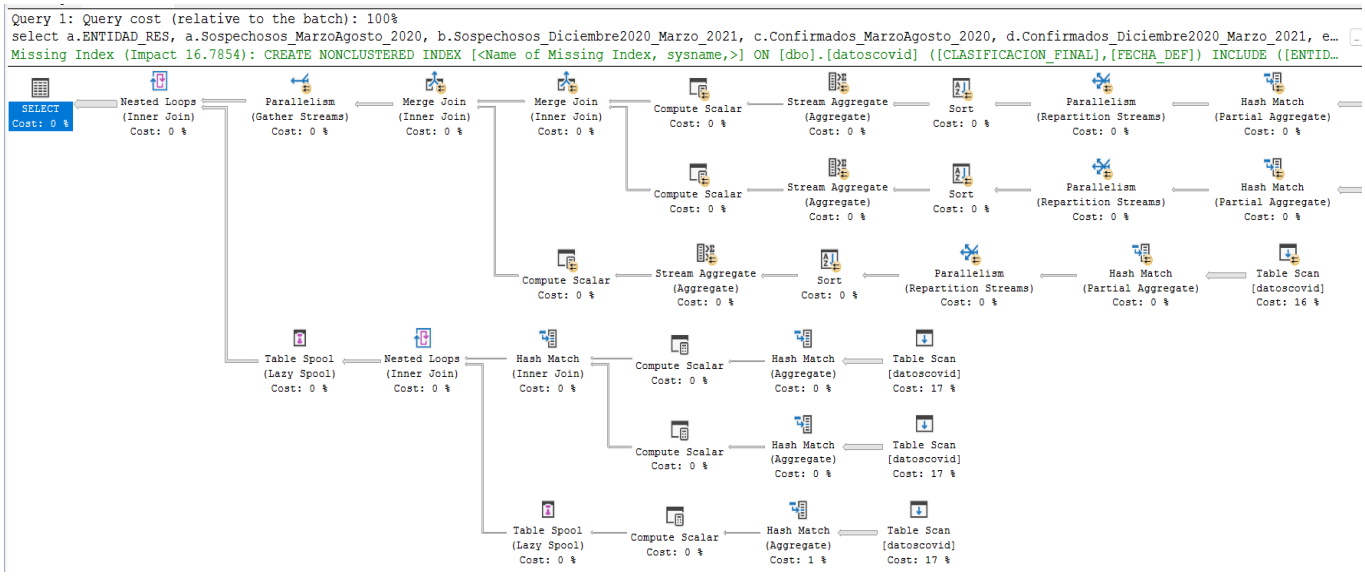
Tenemos un hash match debido a la parte del count que contiene y esto lo hace para poder agilizar más la consulta, por eso tiene el 1% y se complementa con el table scan. Después tenemos un paralellims que ordena los datos para ser mostrados, un sort que se presenta al tener un order by y un group by.

```
--Solucion 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.


select a.ENTIDAD_RES, a.Sospechosos_MarzoAgosto_2020, b.Sospechosos_Diciembre2020_Marzo_2021,
c.Confirmados_MarzoAgosto_2020, d.Confirmados_Diciembre2020_Marzo_2021,

e.Defunciones_MarzoAgosto_2020, f.Defunciones_Diciembre2020_Marzo_2021

from (

select ENTIDAD_RES, count (CLASIFICACION_FINAL) as Sospechosos_MarzoAgosto_2020 from dbo.datoscovid

where CLASIFICACION_FINAL=6 and FECHA_DEF between '2020-03-01' and '2020-08-01'

group by ENTIDAD_RES) a JOIN

(select ENTIDAD_RES, count (CLASIFICACION_FINAL) as Sospechosos_Diciembre2020_Marzo_2021 from
dbo.datoscovid

where CLASIFICACION_FINAL=6 and FECHA_DEF between '2020-12-01' and '2021-03-01'

group by ENTIDAD_RES) b ON

a.ENTIDAD_RES = b.ENTIDAD_RES JOIN

(select ENTIDAD_RES, count (CLASIFICACION_FINAL) as Confirmados_MarzoAgosto_2020 from dbo.datoscovid

where CLASIFICACION_FINAL between 1 and 3 and FECHA_DEF between '2020-03-01' and '2020-08-01'

group by ENTIDAD_RES) c ON

b.ENTIDAD_RES=c.ENTIDAD_RES JOIN

(select ENTIDAD_RES, count (CLASIFICACION_FINAL) as Confirmados_Diciembre2020_Marzo_2021 from
dbo.datoscovid

where CLASIFICACION_FINAL between 1 and 3 and FECHA_DEF between '2020-12-01' and '2021-03-01'

group by ENTIDAD_RES) d ON

c.ENTIDAD_RES=d.ENTIDAD_RES JOIN

(select ENTIDAD_RES, count (FECHA_DEF) as Defunciones_MarzoAgosto_2020 from dbo.datoscovid

where FECHA_DEF between '2020-03-01' and '2020-08-01' and FECHA_DEF!='9999-99-99' group by
ENTIDAD_RES) e ON

d.ENTIDAD_RES=e.ENTIDAD_RES JOIN

(select ENTIDAD_RES, count (FECHA_DEF) as Defunciones_Diciembre2020_Marzo_2021 from dbo.datoscovid

where FECHA_DEF between '2020-12-01' and '2021-03-01' and FECHA_DEF!='9999-99-99' group by
ENTIDAD_RES ) f ON

e.ENTIDAD_RES = f.ENTIDAD_RES Order by a.ENTIDAD_RES;
```

Resultado de la Consulta.

ENTIDAD_RES	Sospechosos_MarzoAgosto_2020	Sospechosos_Diciembre2020_Marzo_2021	Confirmados_MarzoAgosto_2020	Confirmados_Diciembre2020_Marzo_2021	Defunciones_MarzoAgosto_2020	Defunciones_Diciembre2020_Marzo_2021
01	20	15	293	946	387	1312
02	89	64	3376	3181	4115	3819
03	1	4	246	535	342	669
04	22	1	743	139	910	214
05	47	107	1036	2375	1465	3082
07	309	26	1145	269	1629	355
08	29	70	1489	1378	2158	2032
09	1551	771	10192	14948	13908	18031
10	10	20	351	807	520	1021

Plan de ejecución.



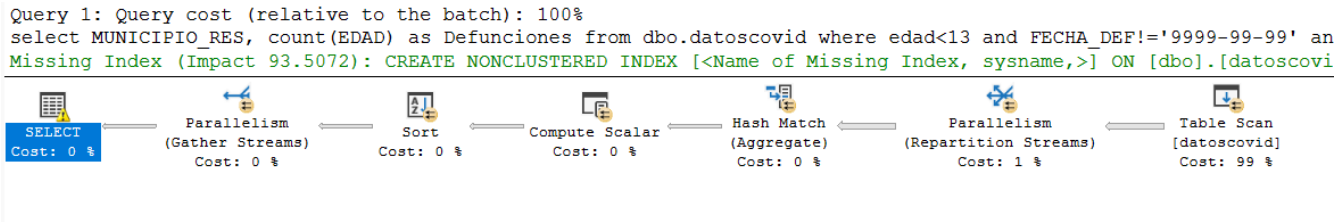
--Solución 7 Listar los 5 municipios con el mayor número de casos confirmados en niños menos de 13 años con alguna comorbilidad reportada y cuantos de esos casos fallecieron.

```
select MUNICIPIO_RES, count(EDAD) as Defunciones from dbo.datoscovid

where edad<13 and FECHA_DEF!='9999-99-99' and ASMA=1 and NEUMONIA=1 group by MUNICIPIO_RES order by Defunciones desc
```

Resultado de la Consulta.

MUNICIPIO_RES	Defunciones
020	4
032	1
002	1
001	1
023	1
012	1
058	1
004	1
038	1
033	1
082	1
104	1
024	1
017	1
046	1
278	1
044	1

Plan de ejecución.



Contamos con un table scan que hace el escaneo total de la tabla, después tenemos un parallelism que une los datos de otras tablas junto con el hash match que agiliza las operaciones aritméticas que esta realiza como los son el count para ser más eficiente y rápida.

Después tenemos un computer scalar apoya en la parte de los valores aritméticos y un sort para hacer ordenar los datos y valores que salgan en las operaciones.

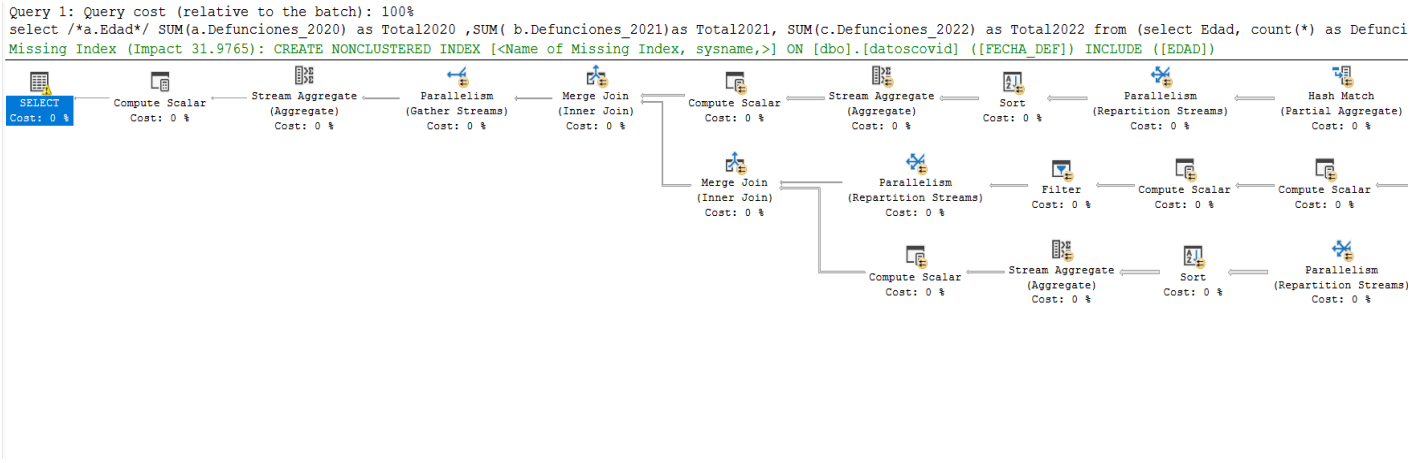
--8. Determinar si en el año 2020 hay una mayor cantidad de defunciones menores de edad que en el año 2021 y 2022.

```
select /*a.Edad*/ SUM(a.Defunciones_2020) as Total2020 ,SUM( b.Defunciones_2021)as Total2021,
SUM(c.Defunciones_2022) as Total2022

from (select Edad, count(*) as Defunciones_2020 from dbo.datoscovid where FECHA_DEF like '2020%' and
FECHA_DEF!='9999-99-99' group by EDAD) a

JOIN

(select Edad, count(*) as Defunciones_2021 from dbo.datoscovid where FECHA_DEF like '2021%' and
FECHA_DEF!='9999-99-99' group by EDAD) b ON

a.EDAD=b.EDAD

JOIN

(select Edad, count(*) as Defunciones_2022 from dbo.datoscovid where FECHA_DEF like '2022%' and
FECHA_DEF!='9999-99-99' group by EDAD) c ONb.EDAD=c.EDAD where a.EDAD < 18
```

Resultado de la Consulta.

Total2020	Total2021	Total2022
1721	1663	342

Plan de ejecución.



Aquí tenemos a 2 procesos hijos ya que están siendo separados por los casos confirmados, casos sospechosos y casos confirmados de diciembre 2020 y mayo del 2021. Los tres hijos contienen un table scan con un costo del 33% cada uno ya que al ser 3 se divide el total del 100%.

De la misma manera los 2 contienen un hash match optimiza los procesos hijos y no se demoren mucho en realizar las operaciones.

Después tenemos un hash match (inner join), este aparece ya que une las diferentes consultas que se hacen en los procesos hijos y las muestras en una sola.

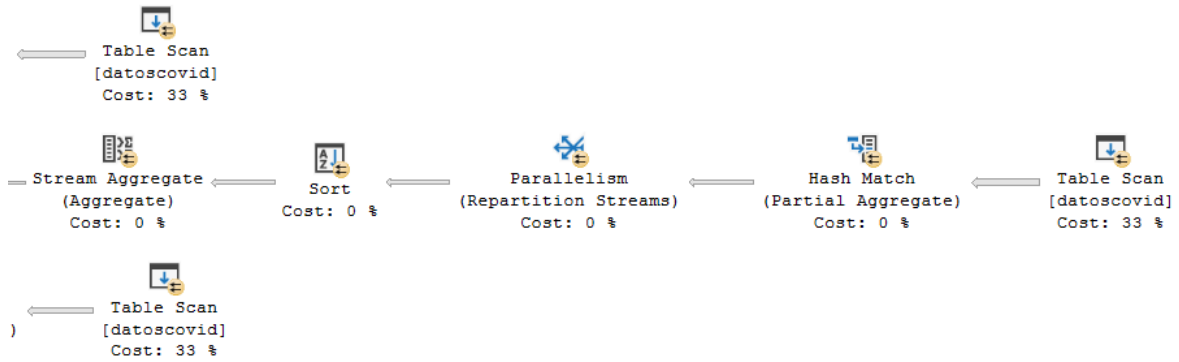
9. Determinar si en el año 2021 hay un pocentaje mayor al 60 de casos reportados que son confirmados por estudios de laboratorio en comparación al año 2020

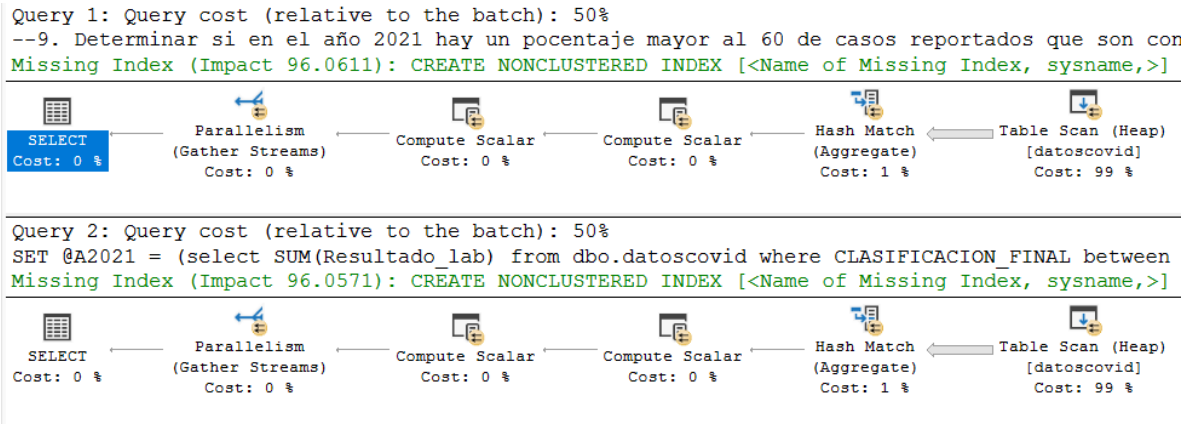
```
Declare @A2020 int;

Declare @A2021 int;

Declare @x     real;

Declare @Porc1 real;

Declare @Porc2 FLOAT;

SET @A2020 = (select SUM(Resultado_lab) from dbo.datoscovid where CLASIFICACION_FINAL between 1 and 3 and FECHA_INGRESO like '2020%')

SET @A2021 = (select SUM(Resultado_lab) from dbo.datoscovid where CLASIFICACION_FINAL between 1 and 3 and FECHA_INGRESO like '2021%')

SET @x = @A2020

SET @Porc1= @A2021/@x

SET @Porc2= (@A2020/100)*60

IF @Porc1 <= @Porc2

        BEGIN

        PRINT 'El porcentaje es menor al 60% del año 2020'

        END     ELSE

        BEGIN

        PRINT 'El porcentaje es Mayor al 60% del año 2020'

        END
```

Resultado de la Consulta.

```
El porcentaje es menor al 60% del año 2020

Completion time: 2022-06-07T09:26:49.7287149-05:00
```

Plan de ejecución.





En esta consulta tenemos tres querys diferentes ejecutándose, primero declaramos 3 variables de tipo entero donde estas almacenaran el conteo de los difuntos. Las 3 consultas contienen un table scan que lo hacen de la tabla datoscovid.

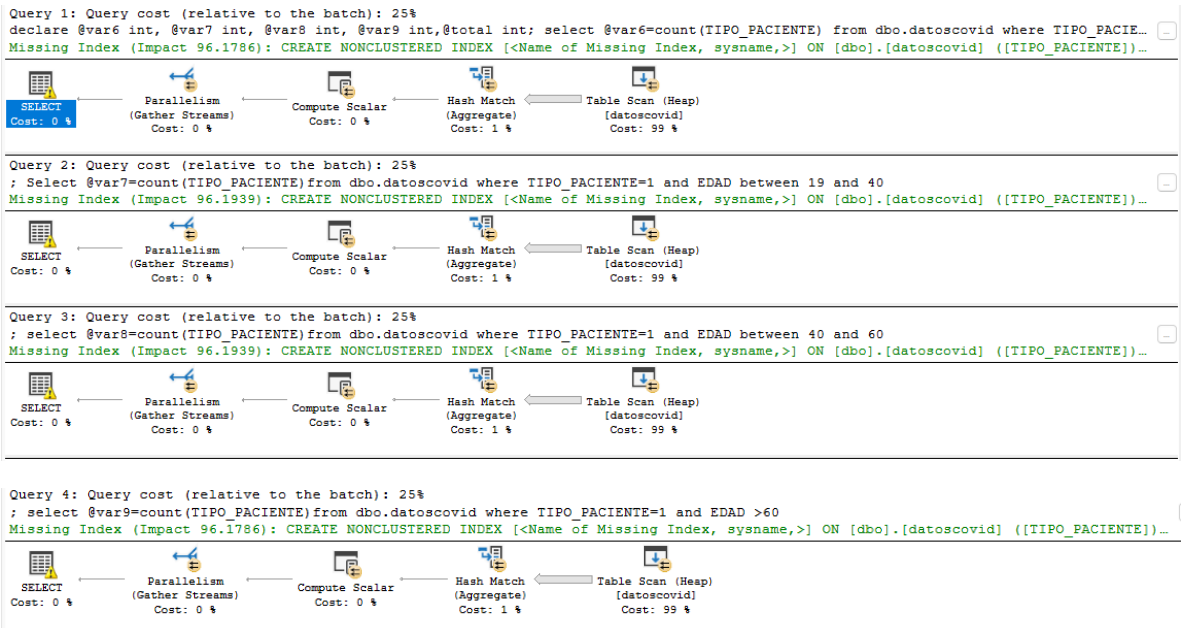
Las tres cuentan con un operador stream aggregate, lo que hace es agregar un operador físico y espera a que agregues datos en las filas, tiene un costo del 1% porque optimiza la consulta y además espera a que termine las comparaciones para agregarlas a la tabla. Se agrega un transact que sería un if que hará la comparación de los resultados.

```
--10
declare @var6 int, @var7 int, @var8 int, @var9 int,@total int;
select @var6=count(TIPO_PACIENTE) from dbo.datoscovid where TIPO_PACIENTE=1 and EDAD < 18;
Select @var7=count(TIPO_PACIENTE)from dbo.datoscovid where TIPO_PACIENTE=1 and EDAD between 19 and
40;
select @var8=count(TIPO_PACIENTE)from dbo.datoscovid where TIPO_PACIENTE=1 and EDAD between 40 and
60;
select @var9=count(TIPO_PACIENTE)from dbo.datoscovid where TIPO_PACIENTE=1 and EDAD >60;

IF @var6>@var7 and @var6>@var8 and @var6>@var9
begin
print 'Casos reportados y recuperados: Menores de edad'
end else
if @var7>@var6 and @var7>@var8 and @var7>@var9
begin
print 'Casos reportados y recuperados: Entre 19 y 40 años'
end else
if @var8>@var6 and @var8>@var7 and @var8>@var9
begin
print 'Casos reportados y recuperados: entre 40 y 60 años'
end else
if @var9>@var6 and @var9>@var7 and @var9>@var8
begin
print 'Casos reportados y recuperados: Mayores de 60 años'
end
```

Resultado de la Consulta.

```
Casos reportados y recuperados: Entre 19 y 40 años

Completion time: 2022-06-07T11:39:10.2259971-05:00
```

Plan de ejecución.



Declaramos 3 variables para almacenar el conteo de los casos, después el table scan hace ese escaneo de las tablas, el hash match optimiza los cálculos y las comparaciones que tiene que hacer para que sean más rápidas y mostrar los resultados.

El computer scalar optimiza los cálculos y el parallelism manda esos datos al select para que sean mostrados.