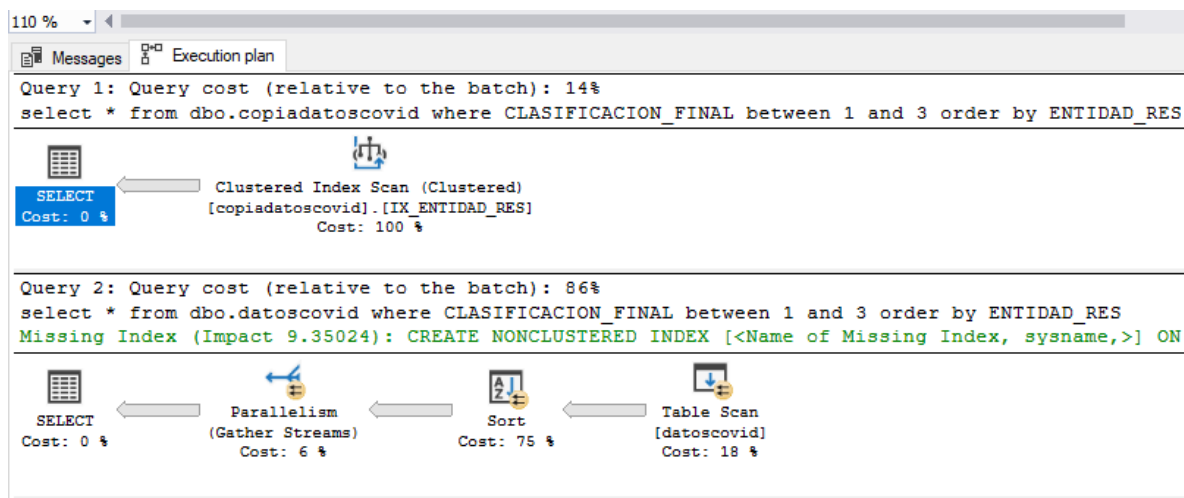```sql
-- 1 Listar los casos potivos por entidad de residencia
CREATE CLUSTERED INDEX IX_ENTIDAD_RES ON dbo.copiadatoscovid(ENTIDAD_RES)
-- solución 1
select *
from dbo.copiadatoscovid
where CLASIFICACION_FINAL between 1 and 3
order by ENTIDAD_RES

select *
from dbo.datoscovid
where CLASIFICACION_FINAL between 1 and 3
order by ENTIDAD_RES
```
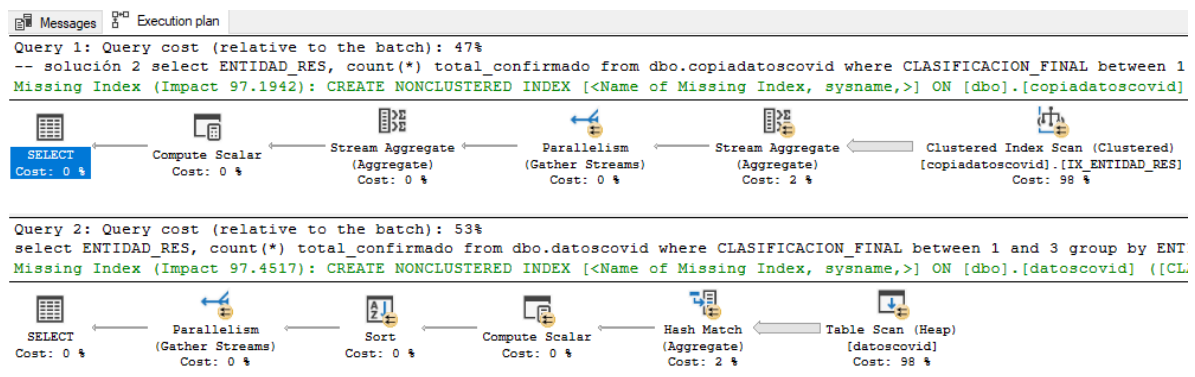
110 %

Messages | Execution plan

Query 1: Query cost (relative to the batch): 14%
select * from dbo.copiadatoscovid where CLASIFICACION_FINAL between 1 and 3 order by ENTIDAD_RES

SELECT
Cost: 0 %
← Clustered Index Scan (Clustered)
[copiadatoscovid].[IX_ENTIDAD_RES]
Cost: 100 %

Query 2: Query cost (relative to the batch): 86%
select * from dbo.datoscovid where CLASIFICACION_FINAL between 1 and 3 order by ENTIDAD_RES
Missing Index (Impact 9.35024): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON

SELECT
Cost: 0 %
← Parallelism
(Gather Streams)
Cost: 6 %
← Sort
Cost: 75 %
← Table Scan
[datoscovid]
Cost: 18 %

```sql
-- solución 2
select ENTIDAD_RES, count(*) total_confirmado
from dbo.copiadatoscovid
where CLASIFICACION_FINAL between 1 and 3
group by ENTIDAD_RES
order by ENTIDAD_RES

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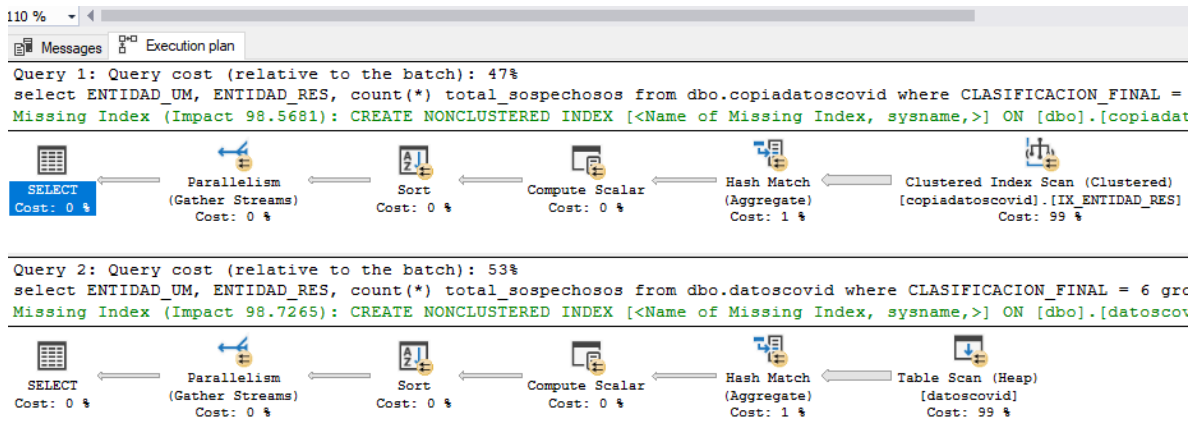
Messages | Execution plan

Query 1: Query cost (relative to the batch): 47%
-- solución 2 select ENTIDAD_RES, count(*) total_confirmado from dbo.copiadatoscovid where CLASIFICACION_FINAL between 1
Missing Index (Impact 97.1942): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[copiadatoscovid]

SELECT
Cost: 0 %
← Compute Scalar
Cost: 0 %
← Stream Aggregate
(Aggregate)
Cost: 0 %
← Parallelism
(Gather Streams)
Cost: 0 %
← Stream Aggregate
(Aggregate)
Cost: 2 %
← Clustered Index Scan (Clustered)
[copiadatoscovid].[IX_ENTIDAD_RES]
Cost: 98 %

Query 2: Query cost (relative to the batch): 53%
select ENTIDAD_RES, count(*) total_confirmado from dbo.datoscovid where CLASIFICACION_FINAL between 1 and 3 group by ENT
Missing Index (Impact 97.4517): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([CL

SELECT
Cost: 0 %
← Parallelism
(Gather Streams)
Cost: 0 %
← Sort
Cost: 0 %
← Compute Scalar
Cost: 0 %
← Hash Match
(Aggregate)
Cost: 2 %
← Table Scan (Heap)
[datoscovid]
Cost: 98 %

```sql
-- 2 Listar los casos sospechosos por entidad
-- Solucion 1
CREATE CLUSTERED INDEX IX_ENTIDAD_RES ON dbo.copiadatoscovid(ENTIDAD_RES)

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

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
```
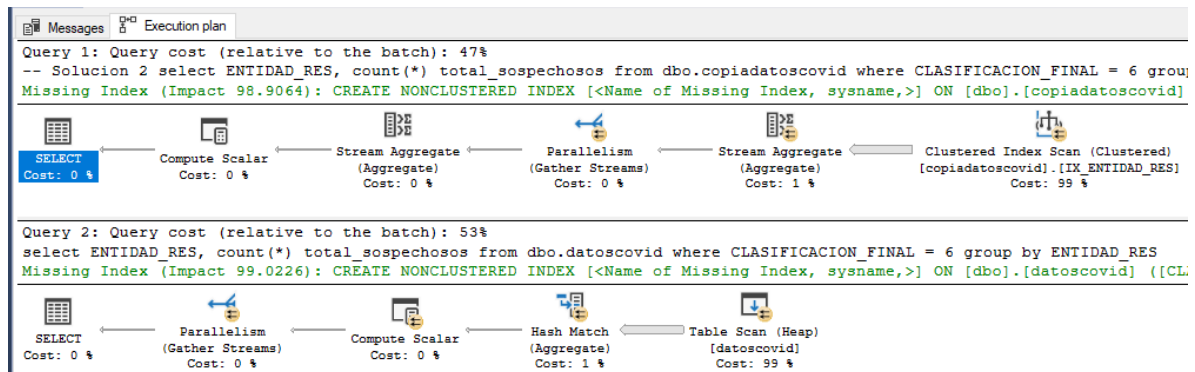
110 %

Messages   Execution plan

Query 1: Query cost (relative to the batch): 47%
select ENTIDAD_UM, ENTIDAD_RES, count(*) total_sospechosos from dbo.copiadatoscovid where CLASIFICACION_FINAL =
Missing Index (Impact 98.5681): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[copiadat

| SELECT Cost: 0 % | Parallelism (Gather Streams) Cost: 0 % | Sort Cost: 0 % | Compute Scalar Cost: 0 % | Hash Match (Aggregate) Cost: 1 % | Clustered Index Scan (Clustered) [copiadatoscovid].[IX_ENTIDAD_RES] Cost: 99 % |

Query 2: Query cost (relative to the batch): 53%
select ENTIDAD_UM, ENTIDAD_RES, count(*) total_sospechosos from dbo.datoscovid where CLASIFICACION_FINAL = 6 gro
Missing Index (Impact 98.7265): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscov

| SELECT Cost: 0 % | Parallelism (Gather Streams) Cost: 0 % | Sort Cost: 0 % | Compute Scalar Cost: 0 % | Hash Match (Aggregate) Cost: 1 % | Table Scan (Heap) [datoscovid] Cost: 99 % |

```sql
-- Solucion 2
select ENTIDAD_RES, count(*) total_sospechosos
from dbo.copiadatoscovid
where CLASIFICACION_FINAL = 6
group by ENTIDAD_RES

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

Messages   Execution plan

Query 1: Query cost (relative to the batch): 47%
-- Solucion 2 select ENTIDAD_RES, count(*) total_sospechosos from dbo.copiadatoscovid where CLASIFICACION_FINAL = 6 grou
Missing Index (Impact 98.9064): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[copiadatoscovid]

| SELECT Cost: 0 % | Compute Scalar Cost: 0 % | Stream Aggregate (Aggregate) Cost: 0 % | Parallelism (Gather Streams) Cost: 0 % | Stream Aggregate (Aggregate) Cost: 1 % | Clustered Index Scan (Clustered) [copiadatoscovid].[IX_ENTIDAD_RES] Cost: 99 % |

Query 2: Query cost (relative to the batch): 53%
select ENTIDAD_RES, count(*) total_sospechosos from dbo.datoscovid where CLASIFICACION_FINAL = 6 group by ENTIDAD_RES
Missing Index (Impact 99.0226): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([CL

| SELECT Cost: 0 % | Parallelism (Gather Streams) Cost: 0 % | Compute Scalar Cost: 0 % | Hash Match (Aggregate) Cost: 1 % | Table Scan (Heap) [datoscovid] Cost: 99 % |

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

go
CREATE CLUSTERED INDEX IX_MUNICIPIO_RES ON dbo.copiadatoscovid(MUNICIPIO_RES)
go

DROP INDEX IX_MUNICIPIO_RES ON dbo.copiadatoscovid
-- solución 1
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.copiadatoscovid
group by ENTIDAD_RES, MUNICIPIO_RES
order by ENTIDAD_RES, reportados desc


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
```
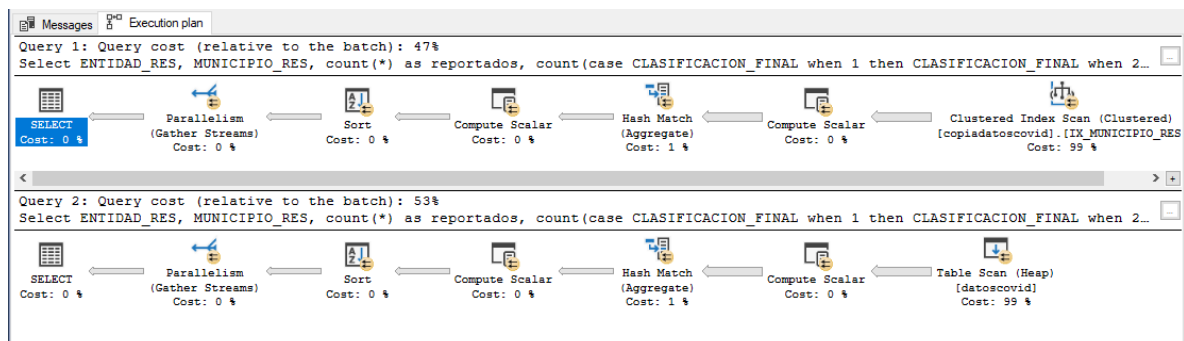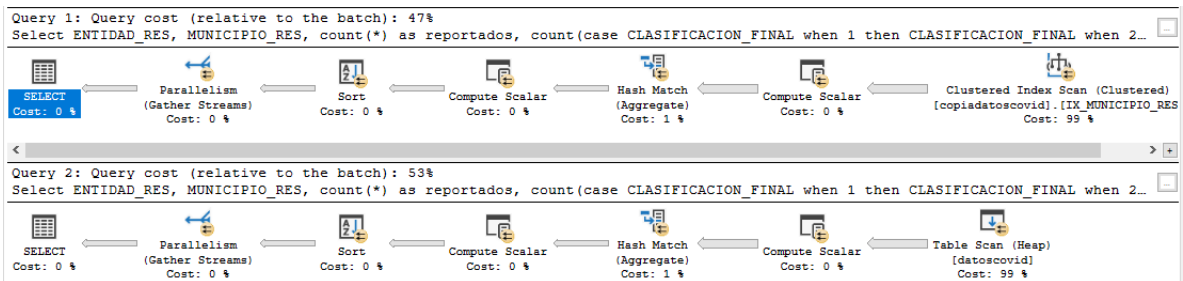
```sql
-- solución 2
select cc.ENTIDAD_RES, cc.MUNICIPIO_RES, cc.confirmado, cs.sospechoso
from (select ENTIDAD_RES, MUNICIPIO_RES, count(*) as sospechoso
        from dbo.copiadatoscovid where CLASIFICACION_FINAL = 6
           group by ENTIDAD_RES, MUNICIPIO_RES
        ) cs
inner join
(select ENTIDAD_RES, MUNICIPIO_RES, count (*) as confirmado
 from dbo.copiadatoscovid 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, cc.MUNICIPIO_RES

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, cc.MUNICIPIO_RES
```
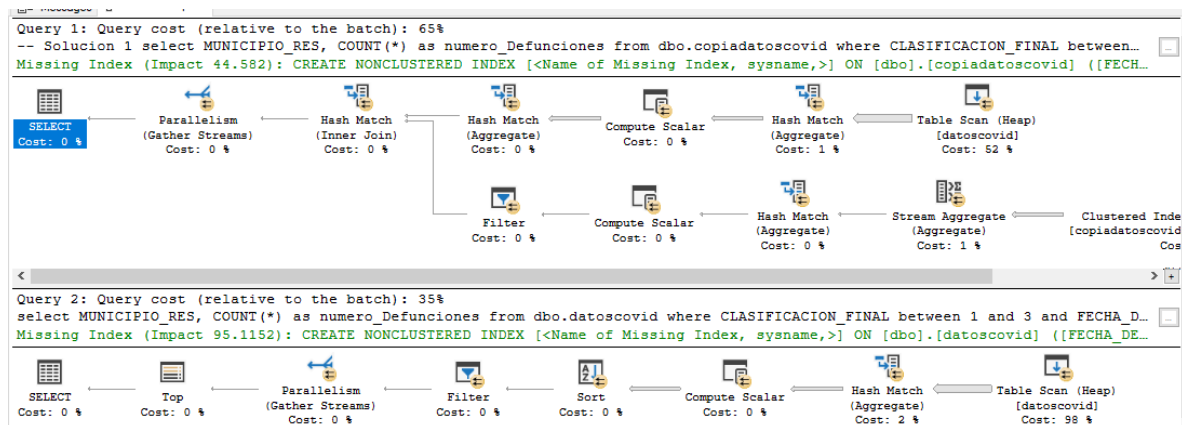

Query 1: Query cost (relative to the batch): 47%
Select ENTIDAD_RES, MUNICIPIO_RES, count(*) as reportados, count(case CLASIFICACION_FINAL when 1 then CLASIFICACION_FINAL when 2...

SELECT Cost: 0 % ← Parallelism (Gather Streams) Cost: 0 % ← Sort Cost: 0 % ← Compute Scalar Cost: 0 % ← Hash Match (Aggregate) Cost: 1 % ← Compute Scalar Cost: 0 % ← Clustered Index Scan (Clustered) [copiadatoscovid].[IX_MUNICIPIO_RES] Cost: 99 %

Query 2: Query cost (relative to the batch): 53%
Select ENTIDAD_RES, MUNICIPIO_RES, count(*) as reportados, count(case CLASIFICACION_FINAL when 1 then CLASIFICACION_FINAL when 2...

SELECT Cost: 0 % ← Parallelism (Gather Streams) Cost: 0 % ← Sort Cost: 0 % ← Compute Scalar Cost: 0 % ← Hash Match (Aggregate) Cost: 1 % ← Compute Scalar Cost: 0 % ← Table Scan (Heap) [datoscovid] Cost: 99 %

```sql
-- 4 Determinar el municipio con el mayor número de defunciones en casos
confirmados.
-- Solucion 1
select MUNICIPIO_RES, COUNT(*) as numero_Defunciones
from dbo.copiadatoscovid
where CLASIFICACION_FINAL between 1 and 3 and FECHA_DEF != '9999-99-99'
group by MUNICIPIO_RES
having COUNT(*) = ( select max(numero_Defunciones)
                            from (
                                    select MUNICIPIO_RES, COUNT(*) as
numero_Defunciones

                                    from dbo.datoscovid
                                    where CLASIFICACION_FINAL between 1 and 3
and FECHA_DEF != '9999-99-99'

                                    group by MUNICIPIO_RES ) as aux )


select MUNICIPIO_RES, COUNT(*) as numero_Defunciones
from dbo.datoscovid
where CLASIFICACION_FINAL between 1 and 3 and FECHA_DEF != '9999-99-99'
group by MUNICIPIO_RES
having COUNT(*) = ( select max(numero_Defunciones)
                            from (
                                    select MUNICIPIO_RES, COUNT(*) as
numero_Defunciones

                                    from dbo.datoscovid
                                    where CLASIFICACION_FINAL between 1 and 3
and FECHA_DEF != '9999-99-99'

                                    group by MUNICIPIO_RES ) as aux )
```
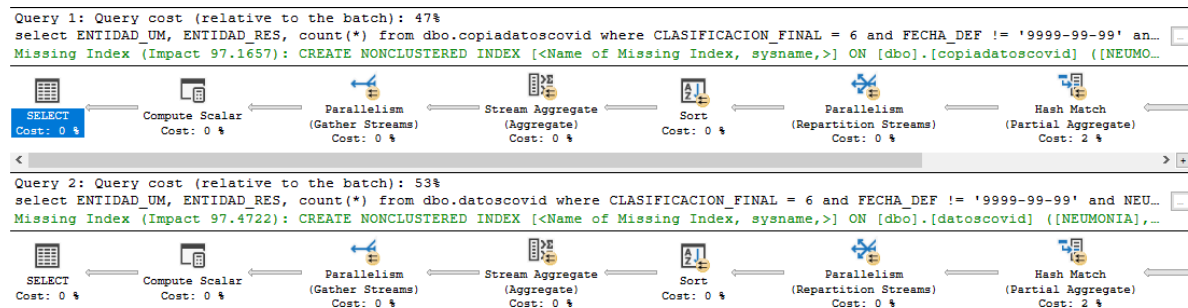
```sql
-- 5. Determinar por entidad, si de casos sospechosos hay defunciones reportadas
asociadas a neumonia.
-- Solucion 1
select ENTIDAD_UM, ENTIDAD_RES, count(*)
from dbo.copiadatoscovid
where CLASIFICACION_FINAL = 6 and FECHA_DEF != '9999-99-99' and NEUMONIA = 1
group by ENTIDAD_UM, ENTIDAD_RES


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
```
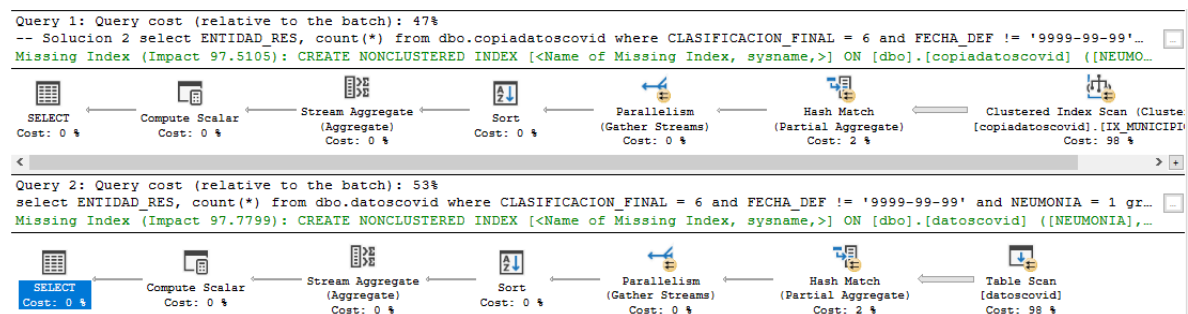
Query 1: Query cost (relative to the batch): 47%
select ENTIDAD_UM, ENTIDAD_RES, count(*) from dbo.copiadatoscovid where CLASIFICACION_FINAL = 6 and FECHA_DEF != '9999-99-99' an...
Missing Index (Impact 97.1657): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[copiadatoscovid] ([NEUMO...

SELECT — Compute Scalar — Parallelism (Gather Streams) — Stream Aggregate (Aggregate) — Sort — Parallelism (Repartition Streams) — Hash Match (Partial Aggregate)
Cost: 0 % — Cost: 0 % — Cost: 0 % — Cost: 0 % — Cost: 0 % — Cost: 0 % — Cost: 2 %

Query 2: Query cost (relative to the batch): 53%
select ENTIDAD_UM, ENTIDAD_RES, count(*) from dbo.datoscovid where CLASIFICACION_FINAL = 6 and FECHA_DEF != '9999-99-99' and NEU...
Missing Index (Impact 97.4722): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([NEUMONIA],...

SELECT — Compute Scalar — Parallelism (Gather Streams) — Stream Aggregate (Aggregate) — Sort — Parallelism (Repartition Streams) — Hash Match (Partial Aggregate)
Cost: 0 % — Cost: 0 % — Cost: 0 % — Cost: 0 % — Cost: 0 % — Cost: 0 % — Cost: 2 %

```sql
-- Solucion 2
select ENTIDAD_RES, count(*)
from dbo.copiadatoscovid
where CLASIFICACION_FINAL = 6 and FECHA_DEF != '9999-99-99' and NEUMONIA = 1
group by ENTIDAD_RES

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

Query 1: Query cost (relative to the batch): 47%
-- Solucion 2 select ENTIDAD_RES, count(*) from dbo.copiadatoscovid where CLASIFICACION_FINAL = 6 and FECHA_DEF != '9999-99-99'...
Missing Index (Impact 97.5105): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[copiadatoscovid] ([NEUMO...

SELECT — Compute Scalar — Stream Aggregate (Aggregate) — Sort — Parallelism (Gather Streams) — Hash Match (Partial Aggregate) — Clustered Index Scan (Cluste [copiadatoscovid].[IX_MUNICIPI
Cost: 0 % — Cost: 0 % — Cost: 0 % — Cost: 0 % — Cost: 0 % — Cost: 2 % — Cost: 98 %

Query 2: Query cost (relative to the batch): 53%
select ENTIDAD_RES, count(*) from dbo.datoscovid where CLASIFICACION_FINAL = 6 and FECHA_DEF != '9999-99-99' and NEUMONIA = 1 gr...
Missing Index (Impact 97.7799): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([NEUMONIA],...

SELECT — Compute Scalar — Stream Aggregate (Aggregate) — Sort — Parallelism (Gather Streams) — Hash Match (Partial Aggregate) — Table Scan [datoscovid]
Cost: 0 % — Cost: 0 % — Cost: 0 % — Cost: 0 % — Cost: 0 % — Cost: 2 % — Cost: 98 %

```sql
-- 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.
-- Solucion 1
go
CREATE CLUSTERED INDEX IX_ENTIDAD_RES ON dbo.copiadatoscovid(ENTIDAD_RES)
go
--------------------------------------BD Copia----------------------------------
Select ENTIDAD_RES, count(case CLASIFICACION_FINAL when 6 then CLASIFICACION_FINAL
end) as sospechoso,
    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 when  FECHA_DEF != '9999-99-99' then FECHA_DEF
end ) as total_Defunciones
from [dbo].[copiadatoscovid]
where FECHA_INGRESO between '2020-03-01' and '2020-08-31' or FECHA_INGRESO between
'2021-05-01' and '2021-12-31'
group by ENTIDAD_RES
order by ENTIDAD_RES
--------------------------------------------------BD Original----------------------
Select ENTIDAD_RES, count(case CLASIFICACION_FINAL when 6 then CLASIFICACION_FINAL
end) as sospechoso,
    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 when  FECHA_DEF != '9999-99-99' then FECHA_DEF
end ) as total_Defunciones
from dbo.datoscovid
where FECHA_INGRESO between '2020-03-01' and '2020-08-31' or FECHA_INGRESO between
'2021-05-01' and '2021-12-31'
group by ENTIDAD_RES
order by ENTIDAD_RES
```
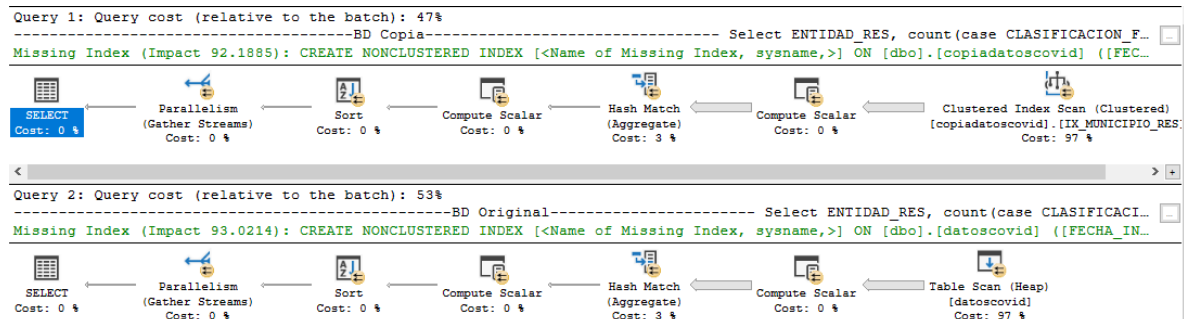
Query 1: Query cost (relative to the batch): 47%
--------------------------------BD Copia-------------------------------- Select ENTIDAD_RES, count(case CLASIFICACION_F...
Missing Index (Impact 92.1885): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[copiadatoscovid] ([FEC...

SELECT Cost: 0 % ← Parallelism (Gather Streams) Cost: 0 % ← Sort Cost: 0 % ← Compute Scalar Cost: 0 % ← Hash Match (Aggregate) Cost: 3 % ← Compute Scalar Cost: 0 % ← Clustered Index Scan (Clustered) [copiadatoscovid].[IX_MUNICIPIO_RES] Cost: 97 %

Query 2: Query cost (relative to the batch): 53%
--------------------------------BD Original-------------------- Select ENTIDAD_RES, count(case CLASIFICACI...
Missing Index (Impact 93.0214): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([FECHA_IN...

SELECT Cost: 0 % ← Parallelism (Gather Streams) Cost: 0 % ← Sort Cost: 0 % ← Compute Scalar Cost: 0 % ← Hash Match (Aggregate) Cost: 3 % ← Compute Scalar Cost: 0 % ← Table Scan (Heap) [datoscovid] Cost: 97 %

```sql
Select MUNICIPIO_RES, count(case CLASIFICACION_FINAL when 1 then CLASIFICACION_FINAL
                                                     when 2 then CLASIFICACION_FINAL

when 3 then CLASIFICACION_FINAL
                                        end) as confirmado,
                        COUNT( case when  FECHA_DEF != '9999-99-99' then
FECHA_DEF end ) as total_Defunciones
from dbo.copiadatoscovid
where EDAD < 13 and -- Falta la condicion para comorbilidad
group by MUNICIPIO_RES
order by confirmado desc


Select MUNICIPIO_RES, count(case CLASIFICACION_FINAL when 1 then CLASIFICACION_FINAL
                                                     when 2 then CLASIFICACION_FINAL

when 3 then CLASIFICACION_FINAL
                                        end) as confirmado,
                        COUNT( case when  FECHA_DEF != '9999-99-99' then
FECHA_DEF end ) as total_Defunciones
from dbo.datoscovid
where EDAD < 13 and -- Falta la condicion para comorbilidad
group by MUNICIPIO_RES
order by confirmado desc
```
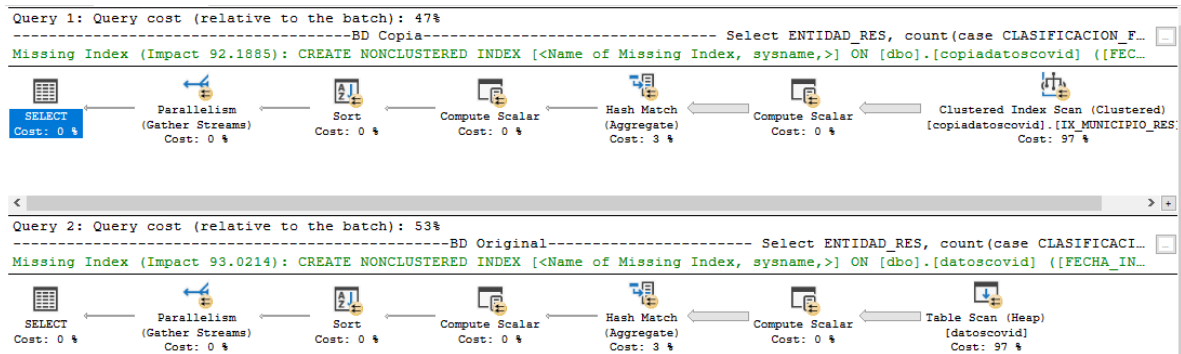
Query 1: Query cost (relative to the batch): 47%
------------------------------BD Copia-------------------------------- Select ENTIDAD_RES, count(case CLASIFICACION_F...
Missing Index (Impact 92.1885): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[copiadatoscovid] ([FEC...

SELECT   Parallelism   Sort   Compute Scalar   Hash Match   Compute Scalar   Clustered Index Scan (Clustered)
Cost: 0 %   (Gather Streams)   Cost: 0 %   Cost: 0 %   (Aggregate)   Cost: 0 %   [copiadatoscovid].[IX_MUNICIPIO_RES]
              Cost: 0 %   Cost: 3 %   Cost: 97 %

Query 2: Query cost (relative to the batch): 53%
--BD Original---------------------- Select ENTIDAD_RES, count(case CLASIFICACI...
Missing Index (Impact 93.0214): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([FECHA_IN...

SELECT   Parallelism   Sort   Compute Scalar   Hash Match   Compute Scalar   Table Scan (Heap)
Cost: 0 %   (Gather Streams)   Cost: 0 %   Cost: 0 %   (Aggregate)   Cost: 0 %   [datoscovid]
              Cost: 0 %   Cost: 3 %   Cost: 97 %
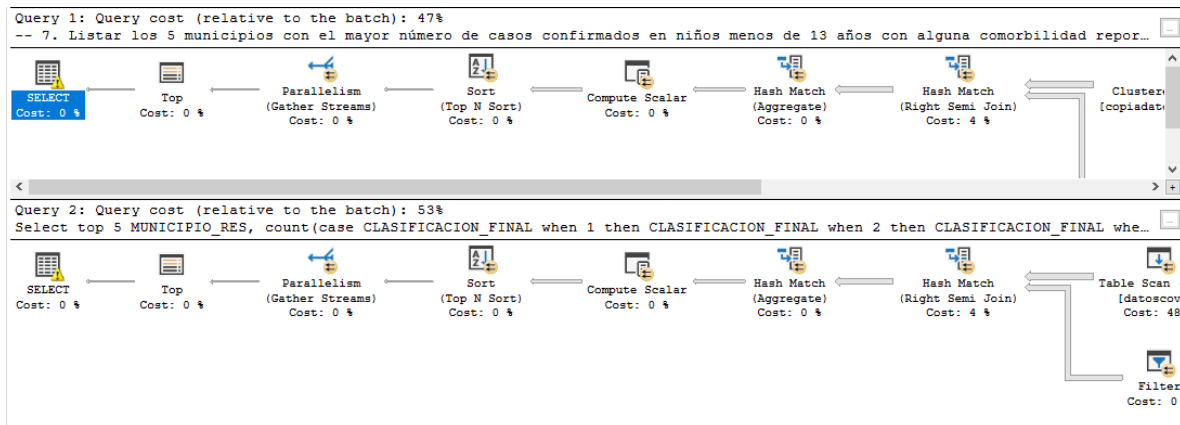
```sql
-- 7. Listar los 5 municipios con el mayor número de casos confirmados en niños
menos de 13 años con alguna comorbilidad reportada y
       --cuantos de esos casos fallecieron.
-- Solucion 1
Select top 5 MUNICIPIO_RES, count(case CLASIFICACION_FINAL when 1 then
CLASIFICACION_FINAL
                                                    when 2 then CLASIFICACION_FINAL

when 3 then CLASIFICACION_FINAL
                                        end) as confirmado,
                                    COUNT( case when  FECHA_DEF != '9999-99-99' then
FECHA_DEF end ) as total_Defunciones
from dbo.copiadatoscovid
where EDAD < 13  and ID_REGISTRO in ( select ID_REGISTRO
                                                        from
dbo.copiadatoscovid
                                                        where EDAD < 13 and (
(NEUMONIA = 1 and DIABETES = 1) or (NEUMONIA = 1 and ASMA = 1)

         or (NEUMONIA = 1 and HIPERTENSION = 1) or (DIABETES = 1 and ASMA = 1)

         or (DIABETES = 1 and HIPERTENSION = 1) or (ASMA = 1 and HIPERTENSION =
1)) )
group by MUNICIPIO_RES
order by confirmado desc


Select top 5 MUNICIPIO_RES, count(case CLASIFICACION_FINAL when 1 then
CLASIFICACION_FINAL
                                                    when 2 then CLASIFICACION_FINAL

when 3 then CLASIFICACION_FINAL
                                        end) as confirmado,
                                    COUNT( case when  FECHA_DEF != '9999-99-99' then
FECHA_DEF end ) as total_Defunciones
from dbo.datoscovid
where EDAD < 13  and ID_REGISTRO in ( select ID_REGISTRO
                                                        from dbo.datoscovid
                                                        where EDAD < 13 and (
(NEUMONIA = 1 and DIABETES = 1) or (NEUMONIA = 1 and ASMA = 1)

         or (NEUMONIA = 1 and HIPERTENSION = 1) or (DIABETES = 1 and ASMA = 1)

         or (DIABETES = 1 and HIPERTENSION = 1) or (ASMA = 1 and HIPERTENSION =
1)) )
group by MUNICIPIO_RES
order by confirmado desc
```

Query 1: Query cost (relative to the batch): 47%
-- 7. Listar los 5 municipios con el mayor número de casos confirmados en niños menos de 13 años con alguna comorbilidad repor...

Query 2: Query cost (relative to the batch): 53%
Select top 5 MUNICIPIO_RES, count(case CLASIFICACION_FINAL when 1 then CLASIFICACION_FINAL when 2 then CLASIFICACION_FINAL whe...

```sql
-- 8. Determinar si en el año 2020 hay una mayor cantidad de defunciones menores de
edad que en el año 2021 y 2022.
-- Solucion 1
Declare @cant2020 int, @cant2021 int, @cant2022 int
set @cant2020 = ( Select COUNT( case when  FECHA_DEF like '2020-%' then FECHA_DEF
end ) as total_Defunciones
                            from dbo.copiadatoscovid
                            where EDAD < 18 ) -- 1776
set @cant2021 = ( Select COUNT( case when  FECHA_DEF like '2021-%' then FECHA_DEF
end ) as total_Defunciones
                            from dbo.copiadatoscovid
                            where EDAD < 18 ) -- 1636
set @cant2022 = ( Select COUNT( case when  FECHA_DEF like '2022-%' then FECHA_DEF
end ) as total_Defunciones
                            from dbo.copiadatoscovid
                            where EDAD < 18 ) -- 314
IF @cant2020 > @cant2021 and @cant2020 > @cant2022
       print 'En el año 2020 hubo una mayor cantidad de defunciones en personas
menores de edad que en el año 2021 y 2022'
ELSE
       print 'En el año 2020 NO hubo una mayor cantidad de defunciones en personas
menores de edad que en el año 2021 y 2022'
-- Solucion 2
Select 2020 as Año, COUNT( case when  FECHA_DEF like '2020-%' then FECHA_DEF end )
as total_Defunciones
from dbo.copiadatoscovid
where EDAD < 18
UNION ALL
Select 2021 as Año, COUNT( case when  FECHA_DEF like '2021-%' then FECHA_DEF end )
as total_Defunciones
from dbo.copiadatoscovid
where EDAD < 18
UNION ALL
Select 2022 as Año, COUNT( case when  FECHA_DEF like '2022-%' then FECHA_DEF end )
as total_Defunciones
from dbo.copiadatoscovid
where EDAD < 18

Select 2020 as Año, COUNT( case when  FECHA_DEF like '2020-%' then FECHA_DEF end )
as total_Defunciones
from dbo.datoscovid
where EDAD < 18
UNION ALL
```
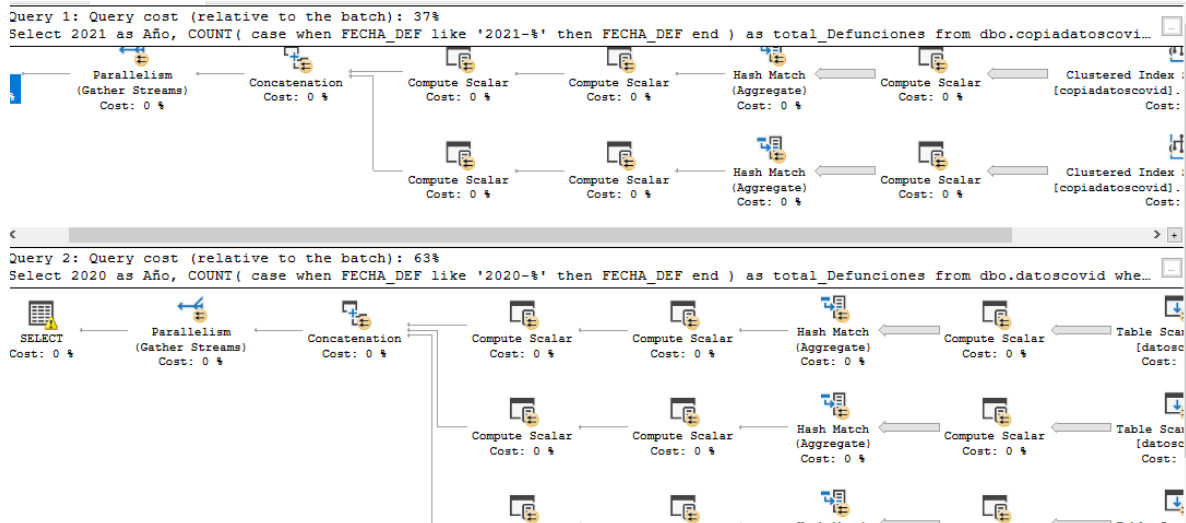
```sql
Select 2021 as Año, COUNT( case when  FECHA_DEF like '2021-%' then FECHA_DEF end )
as total_Defunciones
from dbo.datoscovid
where EDAD < 18
UNION ALL
Select 2022 as Año, COUNT( case when  FECHA_DEF like '2022-%' then FECHA_DEF end )
as total_Defunciones
from dbo.datoscovid
where EDAD < 18
```



```
Query 1: Query cost (relative to the batch): 37%
Select 2021 as Año, COUNT( case when FECHA_DEF like '2021-%' then FECHA_DEF end ) as total_Defunciones from dbo.copiadatoscovi...
```

```
Query 2: Query cost (relative to the batch): 63%
Select 2020 as Año, COUNT( case when FECHA_DEF like '2020-%' then FECHA_DEF end ) as total_Defunciones from dbo.datoscovid whe...
```

```sql
-- 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.
-- Solucion 1
select (aux.Confirmados_Laboratorio*100/aux.Reportados) as Confirmados_Laboratorio
from (
      Select count(*) as Reportados, COUNT( case when CLASIFICACION_FINAL between 1
and 3 then CLASIFICACION_FINAL end ) as Confirmados,
                                              COUNT( case when
CLASIFICACION_FINAL = 3 then CLASIFICACION_FINAL end ) as Confirmados_Laboratorio
      from dbo.copiadatoscovid
      where FECHA_INGRESO like '2021-%' ) as aux

select (aux.Confirmados_Laboratorio*100/aux.Reportados) as Confirmados_Laboratorio
from (
      Select count(*) as Reportados, COUNT( case when CLASIFICACION_FINAL between 1
and 3 then CLASIFICACION_FINAL end ) as Confirmados,
                                              COUNT( case when
CLASIFICACION_FINAL = 3 then CLASIFICACION_FINAL end ) as Confirmados_Laboratorio
      from dbo.datoscovid
      where FECHA_INGRESO like '2021-%' ) as aux
```
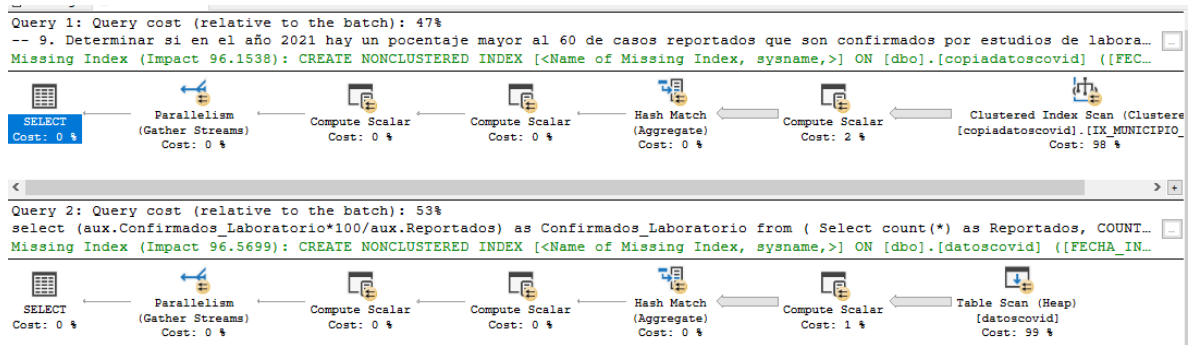
```
-- Solucion 2
Select ( Select count(*)
            from dbo.copiadatoscovid
            where FECHA_INGRESO like '2021-%' and CLASIFICACION_FINAL = 3 ) *100/
COUNT(*) as Confirmados_Laboratorio
from dbo.datoscovid
where FECHA_INGRESO like '2021-%'

Select ( Select count(*)
            from dbo.datoscovid
            where FECHA_INGRESO like '2021-%' and CLASIFICACION_FINAL = 3 ) *100/
COUNT(*) as Confirmados_Laboratorio
from dbo.datoscovid
where FECHA_INGRESO like '2021-%'
```
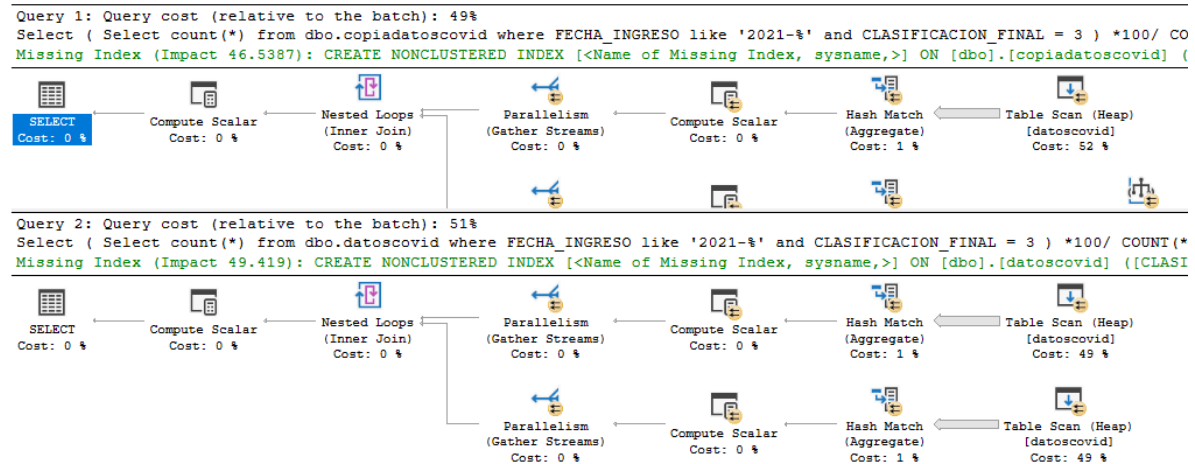
```sql
-- 10. Determinar en que rango de edad: menor de edad, 19 a 40, 40 a 60 o mayor de
60 hay mas casos reportados que se hayan recuperado.
-- Nota: Se desprecia la edad 18 y se toma en cuenta dos veces la edad de 40, porque
asi lo indica la consulta.
-- Solucion 1
SELECT *
from
        ( select 'menor de edad' as Rango_Edad, COUNT(*) as Recuperados
        from dbo.copiadatoscovid
        where EDAD < 18 and FECHA_DEF = '9999-99-99'
        UNION ALL
        select '19 a 40' as Rango_Edad, COUNT(*) as Recuperados
        from dbo.copiadatoscovid
        where EDAD between 19 and 40 and FECHA_DEF = '9999-99-99'
        UNION ALL
        select '40 a 60' as Rango_Edad, COUNT(*) as Recuperados
        from dbo.copiadatoscovid
        where EDAD between 40 and 60 and FECHA_DEF = '9999-99-99'
        UNION ALL
        select 'mayor a 60' as Rango_Edad, COUNT(*) as Recuperados
        from dbo.copiadatoscovid
        where EDAD > 60 and FECHA_DEF = '9999-99-99' ) as aux
where aux.Recuperados = (   SELECT MAX(aux2.Recuperados)
                                            from
                                                ( select 'menor de edad' as
Rango_Edad, COUNT(*) as Recuperados
                                                from dbo.copiadatoscovid
                                                where EDAD < 18 and FECHA_DEF =
'9999-99-99'
                                                UNION ALL
                                                select '19 a 40' as Rango_Edad,
COUNT(*) as Recuperados
                                                from dbo.copiadatoscovid
                                                where EDAD between 19 and 40 and
FECHA_DEF = '9999-99-99'
                                                UNION ALL
                                                select '40 a 60' as Rango_Edad,
COUNT(*) as Recuperados
                                                from dbo.copiadatoscovid
                                                where EDAD between 40 and 60 and
FECHA_DEF = '9999-99-99'
                                                UNION ALL
                                                select 'mayor a 60' as Rango_Edad,
COUNT(*) as Recuperados
                                                from dbo.copiadatoscovid
                                                where EDAD > 60 and FECHA_DEF =
'9999-99-99' ) as aux2 )


SELECT *
from
        ( select 'menor de edad' as Rango_Edad, COUNT(*) as Recuperados
        from dbo.datoscovid
        where EDAD < 18 and FECHA_DEF = '9999-99-99'
        UNION ALL
        select '19 a 40' as Rango_Edad, COUNT(*) as Recuperados
        from dbo.datoscovid
```

```sql
        where EDAD between 19 and 40 and FECHA_DEF = '9999-99-99'
        UNION ALL
        select '40 a 60' as Rango_Edad, COUNT(*) as Recuperados
        from dbo.datoscovid
        where EDAD between 40 and 60 and FECHA_DEF = '9999-99-99'
        UNION ALL
        select 'mayor a 60' as Rango_Edad, COUNT(*) as Recuperados
        from dbo.datoscovid
        where EDAD > 60 and FECHA_DEF = '9999-99-99' ) as aux
where aux.Recuperados = (  SELECT MAX(aux2.Recuperados)
                                    from
                            ( select 'menor de edad' as

Rango_Edad, COUNT(*) as Recuperados

                            from dbo.datoscovid
                            where EDAD < 18 and FECHA_DEF =

'9999-99-99'

                            UNION ALL
                            select '19 a 40' as Rango_Edad,
COUNT(*) as Recuperados

                            from dbo.datoscovid
                            where EDAD between 19 and 40 and
FECHA_DEF = '9999-99-99'

                            UNION ALL
COUNT(*) as Recuperados     select '40 a 60' as Rango_Edad,

                            from dbo.datoscovid
FECHA_DEF = '9999-99-99'     where EDAD between 40 and 60 and

                            UNION ALL
COUNT(*) as Recuperados     select 'mayor a 60' as Rango_Edad,

                            from dbo.datoscovid
'9999-99-99' ) as aux2 )    where EDAD > 60 and FECHA_DEF =
```

Messages   Execution plan

Query 1: Query cost (relative to the batch): 47%
SELECT * from (select 'menor de edad' as Rango_Edad, COUNT(*) as Recuperados from dbo.copiadatoscovid where EDAD < 18 and FEC…
Missing Index (Impact 12.4637): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[copiadatoscovid] ([FEC…

Query 2: Query cost (relative to the batch): 53%
SELECT * from (select 'menor de edad' as Rango_Edad, COUNT(*) as Recuperados from dbo.datoscovid where EDAD < 18 and FECHA_DE…
Missing Index (Impact 12.4676): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([FECHA_DE…