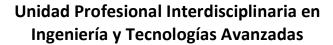


Instituto Politécnico Nacional





Ingeniería Telemática. Bases de Datos Distribuidas.

Integrantes: Lisardo René Morgado Reséndiz.

Moreno Galicia Jesús Antonio.

López Navarrete Sergio Hidekel.

Profesor: De La Cruz Sosa Carlos.

Grupo: 3TM3.

Fecha: Martes 07 de Junio de 2022.

Practica.
(Consultas BD COVID.)

Introducción.

Base de Datos.

Se encarga no solo de almacenar datos, sino también de conectarlos entre sí en una unidad lógica. En términos generales, una base de datos es un conjunto de datos estructurados que pertenecen a un mismo contexto y, en cuanto a su función, se utiliza para administrar de forma electrónica grandes cantidades de información.1 En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, por tanto, se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos. Hay programas denominados sistemas gestores de bases de datos, abreviado SGBD (del inglés Database Management System o DBMS), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos DBMS, así como su utilización y administración, se estudian dentro del ámbito de la informática. Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas; También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

Planes de ejecución.

Un plan de ejecución estimado es un plan de consultas SQL Server que es generado sin realmente correr la consulta (o procedimiento almacenado) para el cual el plan es creado. Está basado en una estimación de comportamiento esperado. Es útil para analizar cómo se comportaría una consulta sin realmente correrla. Esto es muy útil para propósitos de pruebas en ambientes donde el desempeño no debería ser afectado al correr el código real (por ejemplo, correr una sentencia SELECT con uniones complejas con tablas enormes), o cuando correr el código no es posible debido a los cambios en los datos que hace (por ejemplo, ejecutar un UPDATE). Su desventaja es que puede ser poco preciso en algunos escenarios. Un plan de ejecución real es el plan de consultas SQL Server que es generado después de que una consulta fuera ejecutada. Es más confiable, y está basado en la ejecución real, no estimados. También provee más información y estadísticas, por lo que es mucho más útil al resolver problemas. Hay muchos métodos disponibles para abrir un plan de consultas en SQL Server.

Desarrollo.

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.
- 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.
- 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 porcentaje 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.

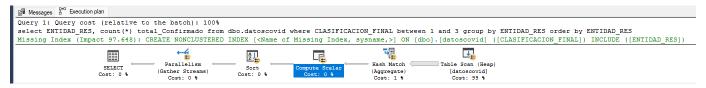
Consultas.

1. Listar los casos positivos por entidad de residencia

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



--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;



2. Listar los casos sospechosos por entidad

Parallelism

(Gather Streams)

Cost: 0 %

```
--1
select ENTIDAD_UM, ENTIDAD_RES, count (*) No_Total_Sospechosos
from dbo.datoscovid
where CLASIFICACION_FINAL =6
group by ENTIDAD_UM, ENTIDAD_RES
       order by ENTIDAD_UM;
 Messages 8° Execution plan

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

--1 select ENTIDAD_UM, ENTIDAD_ERS, count (*) No_Total_Sospechosos from dbo.datoscovid where CLASIFICACION_FINAL =6 group by ENTIDAD_UM, ENTIDAD_RES order by ENTIDAD_UM
Missing Index (Impact 98.7319): CREATE NONCLUSTERED INDEX (<Name of Missing Index, sysname,) ON (dbo).(datoscovid) ((CLASIFICACION_FINAL)) INCLUDE ((ENTIDAD_UM), (ENTIDAD_ERS))
                                                                                                                           唱
                                                                                                                                                        T<sub>2</sub>
                                                                       A J
                                                                                                Parallelism
                                                                                                                       Hash Match
                                                                                                                                                Table Scan (Heap)
                                                                       Sort
                                                                                          Compute Scalar
                                      (Gather Streams)
                                                                                                                                                   [datoscovid]
                                                                                                                       (Aggregate)
                                                                    Cost: 0 %
                                                                                             Cost: 0 %
--2
select ENTIDAD_UM, count (*) No_Total_Sospechosos
from dbo.datoscovid
where CLASIFICACION_FINAL =6
group by ENTIDAD_UM
order by ENTIDAD_UM;
 Messages Execution plan
 Query 1: Query cost (relative to the batch): 100%

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

Missing Index (Impact 99.0319): CREATE NONCLUSTERED INDEX (<Name of Missing Index, sysname,>) ON [dbo].[datoscovid] ([CLASIFICACION_FINAL]) INCLUDE ([ENTIDAD_UM])
                                                                                                                                     唱
                                                                                                                                                                      □
                                                                           A.J.
                LĢ
```

Compute Scalar

Cost: 0 %

Sort

Cost: 0 %

Hash Match (

(Aggregate)

Cost: 0 %

Table Scan (Heap)

[datoscovid]

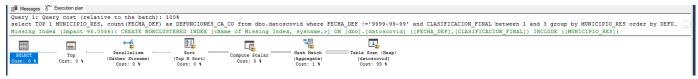
Cost: 100 %

3. <u>Listar el Top 5 de municipios por entidad con el mayor número de casos</u> reportados, indicando casos sospechosos y casos confirmados.

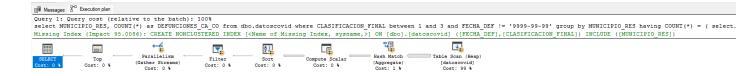
```
--1
         select ENTIDAD_RES, MUNICIPIO_RES, COUNT(*) as REPORTADOS, COUNT(case
         CLASIFICACION FINAL
         when 1 then CLASIFICACION FINAL
         when 2 then CLASIFICACION FINAL
         when 3 then CLASIFICACION FINAL
         end) as CONFIRMADO,
         COUNT(case CLASIFICACION_FINAL when 6 then CLASIFICACION_FINAL end) as
         SOSPECHOSOS
         from dbo.datoscovid
         group by ENTIDAD_RES, MUNICIPIO_RES
         order by ENTIDAD RES, REPORTADOS desc
 Messages Execution plan
 Query 1: Query cost (relative to the batch): 100% 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...
                                                                                       A J
                                                                                   Compute Scalar Table Scan (Heap)
                        Parallelism
                                                               Hash Match
                                         Sort
                                                    Compute Scalar
                      (Gather Streams)
Cost: 0 %
                                                                     (Aggregate)
Cost: 0 %
                                                                                                      [datoscovid]
Cost: 100 %
                                                                                     Cost: 0 %
         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): 100%
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.
Missing Index (Impact 47.2814): CREATE NONCLUSTERED INDEX (<Name of Missing Index, sysname,>) ON [dbo].[datoscovid] ([CLASIFICACION_FINAL]) INCLUDE ([ENTIDAD_RES]), [MUNICIPIO_RES])
                                      唱
                          ξJ
                                                    Parallelism
                                                               Hash Match
(Aggregate)
Cost: 0 %
                                                                           Table Scan (Heap)
                                                                             [datoscovid]
Cost: 50 %
                                                                           Table Scan (Heap
[datoscovid]
Cost: 50 %
                                                                 韫
                                                    Compute Scalar
Cost: 0 %
```

4. <u>Determinar el municipio con el mayor número de defunciones en casos confirmados.</u>

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



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



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

--1 select ENTIDAD_UM, ENTIDAD_RES, count (*) as DEFUNCIONES_AS_NEUMONIA 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): 100%

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

Missing Index (Impact 97.4348): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([NEUMONIA],[CLASIFICACION_FINAL], [FECHA_DEF]) INCLUDE ([ENTIDAD_UM], [ENTIDAD_UM], [ENTIDAD_UM]) **4** ₩ **■** Parallelism Hash Match Parallelism Table Scar (Repartition Streams) Cost: 1 % --2 select ENTIDAD_UM, count (*) as DEFUNCIONES_AS_NEUMONIA from dbo.datoscovid where CLASIFICACION_FINAL=6 and FECHA_DEF != '9999-99-99' and NEUMONIA=1 group by ENTIDAD UM Messages Execution plan Query 1: Query cost (relative to the batch): 100%

--2 select ENTIDAD_UM, count (*) as DEFUNCIONES_AS_NEUMONIA from dbo.datoscovid where CLASIFICACION_FINAL=6 and FECHA_DEF != '9999-99-99' and NEUMONIA=1 group by ENTIDAD_UM

Missing Index (Impact 97.7439): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([NEUMONIA],[CLASIFICACION_FINAL],[FECHA_DEF]) INCLUDE ([ENTIDAD_UM]) 4 352 **↓** å↓ Parallelism (Gather Streams) Cost: 0 % Table Scan [datoscovid] Cost: 99 % Compute Scalar Cost: 0 %

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

```
select ccs.*, dma.TDefuncionesMA as 'Defunciones de Marzo 2020 a Agosto 2020',
             ddm.TDefuncionesDM as 'Defunciones de Diciembre 2020 a Mayo 2021' from
              (select ENTIDAD RES,
             count(case CLASIFICACION FINAL when 1 then CLASIFICACION FINAL
             when 2 then CLASIFICACION FINAL
             when 3 then CLASIFICACION FINAL
             end) as CONFIRMADOS,
             count(case CLASIFICACION_FINAL when 6 then CLASIFICACION_FINAL end)as SOSPECHOSOS
             from dbo.datoscovid group by ENTIDAD_RES) ccs
             inner join
              (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
              (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;
Query 1: Query cost (relative to the batch): 100%
--1 select ccs.*, dma.TDefuncionesMA as 'Defunciones de Marzo 2020 a Agosto 2020', ddm.TDefuncionesDM as 'Defunciones de Diciembre 2020 a Mayo 2021' from (select ENTIDAD_RES, count(case CLASIFI...
Missing Index (Impact 31.677): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([FECHA_DEF]) INCLUDE ([ENTIDAD_RES])
                                                                                                         Table Scan (Heap)
[datoscovid]
Cost: 33 %
                                            4
                                                                                        7
                                                                   猖
                                                                                                               望

                                                                                         Hash Match
(Aggregate)
Cost: 0 %
                                                                                                               唱
                                                                                         Compute Scalar
Cost: 0 %
                                                                                                            (Aggregate)
             Select ENTIDAD_RES, count(case CLASIFICACION_FINAL when 6 then CLASIFICACION_FINAL
             end) as SOSPECHOSO,
             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_INGRESO between '2020-03-01' and '2020-
             08-31' then FECHA_DEF end ) as 'Defunciones de Marzo 2020 a Agosto 2020'
             from dbo.datoscovid
             group by ENTIDAD_RES
             order by ENTIDAD RES
      Query 1: Query cost (relative to the batch): 100%
--2 Select ENTIDAD_RES, count(case CLASIFICACION_FINAL when 6 then CLASIFICACION_FINAL end) as SOSPECHOSO, count(case CLASIFICACION_FINAL when 1 then CLASIFICACION_FINAL when 2 then CLASIFICACION_FINAL when 2 then CLASIFICACION_FINAL when 3 then CLASIFICACION_FINAL when 3 then CLASIFICACION_FINAL when 3 then CLASIFICACION_FINAL when 3 then CLASIFICACION_FINAL when 4 then CLASIFICACION_FINAL when 5 then CLASIFICACION_FINAL when 6 then CLASIFICACION_FINAL when 5 then CLASIFICACION_FINAL when 6 then CLASIFICACION_FINAL wh
                                                                                           唱
                                                                      4
                           Parallelism
                                                                                                                                   Table Scan (Heap)
                                                                                        (Aggregate)
Cost: 0 %
```

7. <u>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.</u>

```
select top 5 MUNICIPIO RES from dbo.datoscovid where EDAD < 13;</pre>
                                select MUNICIPIO_RES, NEUMONIA, DIABETES, EPOC, ASMA, INMUSUPR, HIPERTENSION,
                                OTRA COM, CARDIOVASCULAR, OBESIDAD, RENAL CRONICA
                                TABAQUISMO, OTRO CASO, edad, FECHA DEF
                                from dbo.datoscovid where edad <13 and FECHA DEF!='9999-99-99';
                   Query 1: Query cost (relative to the batch): 100% general control of the batch): 100% general control of the batch): 100% general control of the batch of the bat
                                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 DEFUNCIONES TO
                                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
Messages 6 Execution plan
Query 1: Query cost (relative to the batch): 100%
--2 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 wh.
                                                                               A J
                                                                                                                                      唱
                                                                                                                                                                    唱
                                                                                                          1
                         Hash Match
(Right Semi Join)
Cost: 1 %
                                              Parallelism
                                                                                                                                  Hash Match
                                                                                                                                                                                               Table Scan (Heap)
                      Top
Cost: 0 %
                                                                                                     Compute Scalar
                                           (Gather Streams)
Cost: 0 %
                                                                                                                                                                                                   [datoscovid]
Cost: 49 %
                                                                                                        Cost: 0 %
                                                                                                                                                                                                       7
                                                                                                                                                                                                                                 Table Scan (Heap)
                                                                                                                                                                                                      Filter
                                                                                                                                                                                                                            Compute Scalar
Cost: 0 %
```

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

--1

T-SQL PRINT

DECLARE @C2020 int, @C2021 int, @C2022 int;

select @C2020=1, @C2021=1, @C2022=1;

```
select @C2020= count(FECHA_DEF) from dbo.datoscovid where edad<18 and FECHA_DEF
         between '2020-01-01' and '2020-12-31'
         select @C2021= count(FECHA_DEF) from dbo.datoscovid where edad<18 and FECHA_DEF</pre>
         between '2021-01-01' and '2021-12-31'
         select @C2022= count(FECHA DEF) from dbo.datoscovid where edad<18 and FECHA DEF</pre>
         between '2022-01-01' and '2022-12-31'
         IF @C2020 > @C2021 and @C2020 > @C2022
         print 'En el año 2020 hay una mayor cantidad de defunciones menores de edad que en
         el año 2021 y 2022´
         print 'En el año 2020 NO hay una mayor cantidad de defunciones menores de edad que
         en el año 2021 y 2022´
Query 1: Query cost (relative to the batch): 0%
--1 DECLARE @C2020 int, @C2021 int, @C2022 int; select @C2020=1, @C2021=1, @C2022=1
Ouerv 2: Ouerv cost (relative to the batch): 33%
; select @C2020= count(FECHA DEF) from dbo.datoscovid where edad<18 and FECHA DEF between '2020-01-01' and '2020-12-31'
Missing Index (Impact 95.3634): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([FECHA_DEF]) INCLUDE ([EDAD])
                                   Bξ
                                                      4
                                                                                           1
  Parallelism
                                                                                        Table Scan
                              Stream Aggregate
                                                                    Stream Aggregate
             Compute Scalar
                                (Aggregate)
                                                 (Gather Streams)
                                                                      (Aggregate)
Cost: 1 %
                                                                                        [datoscovid]
Cost: 0 %
               Cost: 0 %
                                 Cost: 0 %
                                                    Cost: 0 %
                                                                                        Cost: 99 %
Query 3: Query cost (relative to the batch): 33%
select @C2021= count(FECHA_DEF) from dbo.datoscovid where edad<18 and FECHA_DEF between '2021-01-01' and '2021-12-31'
Missing Index (Impact 95.3633): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([FECHA_DEF]) INCLUDE ([EDAD])
                                   35
                                                                        Β≧
                                                                                           ↓
  Stream Aggregate
(Aggregate)
Cost: 0 %
                                                 Parallelism
(Gather Streams)
                                                                    Stream Aggregate
                                                                                        Table Scan
             Compute Scalar
Cost: 0 %
 SELECT
                                                                                        [datoscovid]
Cost: 99 %
                                                                      (Aggregate)
Cost: 1 %
Cost: 0 %
                                                    Cost: 0 %
Query 4: Query cost (relative to the batch): 33%
select @C2022= count(FECHA DEF) from dbo.datoscovid where edad<18 and FECHA DEF between '2022-01-01' and '2022-12-31'
Missing Index (Impact 95.3654): CREATE NONCLUSTERED INDEX (<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([FECHA DEF]) INCLUDE ([EDAD])
                                                                        Β}Ξ
                                                                                          ₽
                                   35
  Stream Aggregate
                                                   Parallelism
                                                                    Stream Aggregate
                                                                                        Table Scan
 SELECT
             Compute Scalar
                                (Aggregate)
Cost: 0 %
                                                 (Gather Streams)
Cost: 0 %
                                                                      (Aggregate)
Cost: 1 %
Cost: 0 %
               Cost: 0 %
Ouerv 5: Ouerv cost (relative to the batch): 0%
IF @C2020 > @C2021 and @C2020 > @C2022
  *?3
T F
                T-SQL
  COND
               PRINT
Cost: 0 %
              Cost: 0 %
```

```
--2
           Select 2020 as AÑO, COUNT( case when FECHA_DEF like '2020-%' then FECHA_DEF end ) as
           DEFUNCIONES_T
           from dbo.datoscovid
           where EDAD < 18
           UNION ALL
           Select 2021 as AÑO, COUNT( case when FECHA_DEF like '2021-%' then FECHA_DEF end ) as
           DEFUNCIONES_T
           from dbo.datoscovid
           where EDAD < 18
           UNION ALL
           Select 2022 as AÑO, COUNT( case when FECHA_DEF like '2022-%' then FECHA_DEF end ) as
           DEFUNCIONES_T
           from dbo.datoscovid
           where EDAD < 18
Query 1: Query cost (relative to the batch): 100%
--2 Select 2020 as AÑO, COUNT( case when FECHA_DEF like '2020-%' then FECHA_DEF end ) as DEFUNCIONES_T from dbo.datoscovid where EDAD < 18 UNION ALL Select 2021 as AÑO, COUNT( case when FECHA_D...
              #
                                                                         唱
                                                                                                    Table Scan (Heap)
[datoscovid]
Cost: 33 %
                              7
                                                           Parallelism
(Gather Streams)
Cost: 0 %
                                                                       Hash Match
(Aggregate)
Cost: 0 %
                                                                       Hash Match
(Aggregate)
Cost: 0 %
                                                                                                    Table Scan (Heap)
[datoscovid]
Cost: 33 %
                                                          ompute Scalar
Cost: 0 %
                                                                                      ompute Scalar
Cost: 0 %
                                          Compute Scalar
Cost: 0 %
                                                                       Hash Match
(Aggregate)
Cost: 0 %
```

9. <u>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.</u>

```
select * from dbo.datoscovid;
               declare @C_2021 float, @C_2020 float;
               select @C 2021=count(RESULTADO LAB)*0.6 from dbo.datoscovid
               where RESULTADO LAB =1 and FECHA ACTUALIZACION between '2021-01' and '2021-12-
               31';
               select @C_2020= 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 mas del 60 porciento'
               where @C_2021>@C_2020;
Query 1: Query cost (relative to the batch): 34% select * from dbo.datoscovid
                    4
  Table Scan
[datoscovid]
Cost: 100 %
Query 2: Query cost (relative to the batch): 33%; declare @C_2021 float, @C_2020 float; select @C_2021=count(RESULTADO_LAB)*0.6 from dbo.datoscovid where RESULTADO_LAB =1 and FECHA_ACTUALIZACION between '2021-01-01' and '2021-12-31' Missing Index (Impact 97.5624): CREATE NONCLUSIERED INDEX (<Name of Missing Index, sysname,>) ON [dbo].[datoscovid] ([RESULTADO_LAB], [FECHA_ACTUALIZACION])
                                                            3Σ
                                                                                                       B
                                                                                                                           4
                                       €
                   Parallelism
(Gather Streams
Cost: 0 %
                                                                                                                        Table Scan
[datoscovid]
Cost: 99 %
Query 3: Query cost (relative to the batch): 33%
; select 8C_2020 count (RESULTADO_LAB)*0.6 from dbo.datoscovid where RESULTADO_LAB=1 and FECHA_ACTUALIZACION between '2020-01-01' and '2020-12-32'
Missing Index (Impact 97.5624): CREATE NONCLUSTERED INDEX (<Name of Missing Index, sysname,)) ON (dbo].[datoscovid] ([RESULTADO_LAB], [FECHA_ACTUALIZACION])
                                                            3×E
                                                                            Parallelism
(Gather Stream
Cost: 0 %
                                                                                                    ream Aggrega
(Aggregate)
Cost: 1 %
                                                                                                                        Table Scan
[datoscovid]
Cost: 99 %
                                       Query 4: Query cost (relative to the batch): 0%; select 'VERDADERO' as 'En 2021 hay mas del 60 porciento' where @C_2021>@C_2020
                   7
  1
 SELECT
ost: 0 %
                                   Filter
Cost: 28 %
                                                   Constant Scan
Cost: 67 %
```

```
--2
                       Select ( Select count(*)
                       from dbo.datoscovid
                       where FECHA_INGRESO like '2021-%' and CLASIFICACION_FINAL = 3 ) *100/
                       COUNT(*) as CONFIRMADOS_2021
                       from dbo.datoscovid
                       where FECHA_INGRESO like '2021-%'
Messages & Excutton plan

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

Select (Select count(*) from dbo.datoscovid where FECHA_INGRESO like '2021-%' and CLASIFICACION_FINAL = 3 ) *100/ COUNT(*) as CONFIRMADOS_2021 from dbo.datoscovid where FECHA_INGRESO like '2021-

Missing Index (Impact 49.2987): CREATE NONCLUSTERED INDEX («Name of Missing Index, sysname,») ON [dbo].[datoscovid] ([CLASIFICACION_FINAL], [FECHA_INGRESO])
                                                                                                                                           Table Scan (Heap)
[datoscovid]
Cost: 50 %
                                                †P
                                                                                             Compute Scalar
Cost: 0 %
                                                                                                                        74
                   Compute Scalar
Cost: 0 %
  Nested Loops
(Inner Join)
Cost: 0 %
                                                                  Parallelism
(Gather Streams)
Cost: 0 %
                                                                                                                     Hash Match
(Aggregate)
Cost: 0 %
                                                                  Parallelism
(Gather Streams)
Cost: 0 %
                                                                                                                     Hash Match
(Aggregate)
Cost: 0 %
                                                                                                                                           Table Scan (Heap)
[datoscovid]
Cost: 50 %
                                                                                            Compute Scalar
Cost: 0 %
```

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

```
--1
           select * from dbo.datoscovid;
           declare @A0 int, @A19 int, @A40 int, @A60 int,@total int;
           select @A0=count(TIPO_PACIENTE) from dbo.datoscovid
           where TIPO PACIENTE=1 and EDAD < 18;</pre>
           Select @A19= count(TIPO PACIENTE) from dbo.datoscovid
           where TIPO PACIENTE=1 and EDAD between 19 and 40;
           select @A40= count(TIPO_PACIENTE)from dbo.datoscovid
           where TIPO_PACIENTE=1 and EDAD between 40 and 60;
           select @A60= count(TIPO_PACIENTE)from dbo.datoscovid
           where TIPO PACIENTE=1 and EDAD >60;
           select 'VERDADERO' as 'en este rango de edad hay mas casos recuperados' where
          @A0>@A19 or @A0>@A40 or @A0>@A60
                              or @A19>@A0 or @A19>@A40 or @A19>@A60
                                or @A40>@A0 or @A40>@A19 or @A40>@A60
                                or @A60>@A0 or @A60>@A19 or @A60>@A40
 Query 1: Query cost (relative to the batch): 20% --1 select * from dbo.datoscovid
    4
                           Table Scan
[datoscovid]
Cost: 100 %
Query 2: Query cost (relative to the batch): 20%; declare @AO int, @A19 int, @A40 int, @A60 int,@total int; select @AO=count(TIPO_PACIENTE) from dbo.datoscovid where TIPO_PACIENTE=1 and EDAD < 18
 Missing Index (Impact 96.1583): CREATE NONCLUSTERED INDEX (<Name of Missing Index, sysname,>) ON [dbo].[datoscovid] ([TIPO_FACIENTE]) INCLUDE ([EDAD])
                                                                                            4
                                                                                                                            T<sub>E</sub>
                                4
                                                                Parallelism
(Gather Streams)
Cost: 0 %
                                                                                        Hash Match
(Aggregate)
Cost: 0 %
                                                                                                                     Table Scan (Heap)
[datoscovid]
Cost: 100 %
 SELECT
Cost: 0 %
Query 3: Query cost (relative to the batch): 20%; Select @A19= count(TIPO PACIENTE) from dbo.datoscovid where TIPO PACIENTE=1 and EDAD between 19 and 40
 Missing Index (Impact 96.1706): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([TIPO_PACIENTE]) INCLUDE ([EDAD])
                                                                                                                            1
                                                                Ę
                            Parallelism
                                                                                        Hash Match
                                                                                                                    Table Scan (Heap)
 SELECT
Cost: 0 %
                                                         Compute Scalar
Cost: 0 %
                        (Gather Streams)
Cost: 0 %
                                                                                        (Aggregate)
Cost: 1 %
                                                                                                                        [datoscovid]
Cost: 99 %
 Query 4: Query cost (relative to the batch): 20%
; select @A40= count(TIPO_PACIENTE) from dbo.datoscovid where TIPO_PACIENTE=1 and EDAD between 40 and 60
Missing Index (Impact 96.1706): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([TIPO_PACIENTE]) INCLUDE ([EDAD])
                                                                                            唱
                                                                                                                             ↓
    Parallelism
                                                                                        Hash Match
                                                                                                                   Table Scan (Heap)
   SELECT
                                                         Compute Scalar
                        (Gather Streams)
Cost: 0 %
                                                                                        (Aggregate)
Cost: 1 %
                                                                                                                        [datoscovid]
Cost: 99 %
 Cost: 0 %
                                                            Cost: 0 %
 Query 5: Query cost (relative to the batch): 20%
 ; select @A60= count(TIPO_PACIENTE) from dbo.datoscovid where TIPO_PACIENTE=1 and EDAD >60
 Missing Index (Impact 96.1583): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[datoscovid] ([TIPO_PACIENTE]) INCLUDE ([EDAD])
                                                                                            唱
                                                                                                                             ₽
    Paralleliam
                                                                                        Hash Match
                                                                                                                    Table Scan (Hean)
   SELECT
                         (Gather Stream
Cost: 0 %
                                                                                                                        [datoscovid]
Cost: 100 %
 Cost: 0 %
                                                            Cost: 0 %
Query 6: Query cost (relative to the batch): 0%; select 'VERDADERO' as 'en este rango de edad hay mas casos recuperados' where %AO>%A19 or %AO>%A19 or %AO>%A60 or %A19>%A60 or %A19>%A60 or %A19>%A60 or %A19>%A60 or %A10>%A0 or %A10>%A
                                               7
  ompute Scalar
Cost: 2 %
                                            Filter
Cost: 76 %
```

```
--2
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
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'
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
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'
select 'mayor a 60' as Rango_Edad, COUNT(*) as Recuperados from dbo.datoscovid
where EDAD > 60 and FECHA DEF = '9999-99-99' ) as aux2 )
```

