



INSTITUTO POLITECNICO NACIONAL

**UNIDAD PROFESIONAL INTERDISCIPLINARIA EN
INGENIERIA Y TECNOLOGIAS AVANZADAS**

FRAGMENTACIÓN

PRÁCTICA 3

**INGENIERIA TELEMÁTICA
BASE DE DATOS DISTRIBUIDAS**

PROFESOR: De la Cruz Sosa Carlos

ALUMNOS:

Legorreta Rodriguez Maria Fernanda

Macías Galván Arturo Daniel

Palacios Reyes Leslie Noemi

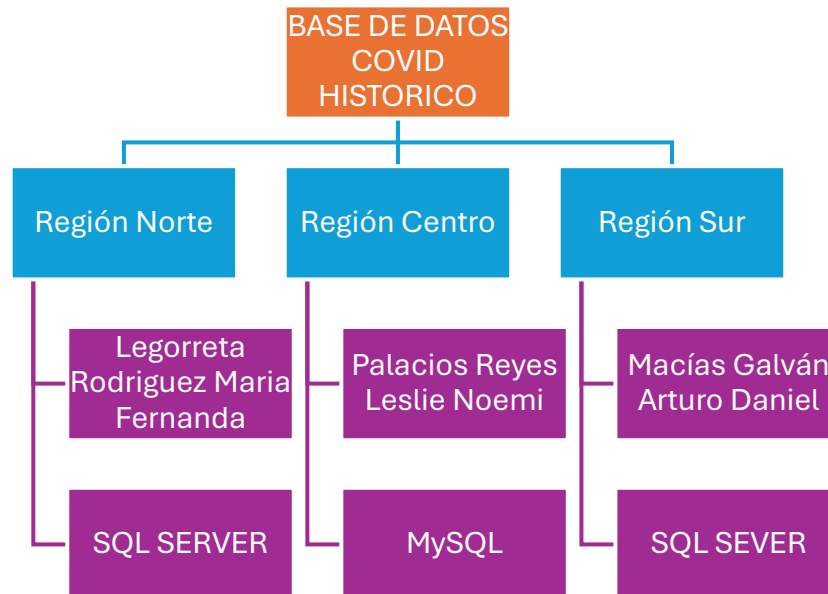
GRUPO: 3TM3

SEMESTRE 2025/2

Tabla de contenido

DIAGRAMA DE FRAGMENTACIÓN.....	3
SCRIPT SQL DE CADA BASE DE DATOS POR REGIÓN	5
Norte	5
Centro	6
Sur.....	7
ESTRATEGIAS DE CONECTIVIDAD PARA LA DISTRIBUCIÓN	9
MIGRACION DE DATOS	10
LISTADO DE NOMBRE DE SERVIDORES VINCULADOS	12
General.....	12
Legorreta Rodríguez María Fernanda	14
Macías Galván Arturo Daniel.....	15
Palacios Reyes Leslie Noemi	16
DESCRIPCIÓN DE LAS CONSULTAS MODIFICADAS:	17
Consulta 3:	17
Consulta 4:	17
Consulta 5:	17
Consulta 7:	17
SCRIPT DE CONSULTAS	17
RESULTADOS DE LAS CONSULTAS:.....	21
Consulta 3:	21
CONCLUSIONES	23
Legorreta Rodríguez María Fernanda.....	23
Macías Galván Arturo Daniel	23
Palacios Reyes Leslie Noemi	24

DIAGRAMA DE FRAGMENTACIÓN



Este diagrama nos muestra como a partir de una misma Base de Datos que cada integrante del equipo compartía, se hicieron las fragmentaciones necesarias por cada región usando como referencia la clasificación regional establecida por el INEGI en la Encuesta Nacional de Inclusión Financiera 2018 (INEGI, 2018).

INEGI (2018). *Encuesta Nacional de Inclusión Financiera 2018*. Recuperado de:

(Mexico - Encuesta Nacional de Inclusión Financiera 2018. <https://www.inegi.org.mx/rnm/index.php/catalog/434/variable/F12/V1179?name=REGION>).

Es importante aclarar que para que el servidor de Región Centro no se sobrecargará de registros, la región *Centro Sur y Oriente*, pertenecerán a región SUR, por lo que cada región en nuestra base de datos tendrá dos regiones oficiales del INEGI.

CATEGORIAS	
Valor	Categoría
1	Noroeste (Baja California, Baja California Sur, Chihuahua, Durango, Sinaloa, Sonora)
2	Noreste (Coahuila, Nuevo León, San Luis Potosí, Tamaulipas)
3	Occidente y Bajío (Aguascalientes, Guanajuato, Jalisco, Michoacán, Nayarit, Querétaro, Zacatecas, Colima)
4	Ciudad de México (Ciudad de México)
5	Centro Sur y Oriente (Estado de México, Hidalgo, Morelos, Puebla, Tlaxcala, Veracruz)
6	Sur (Campeche, Chiapas, Guerrero, Quintana Roo, Tabasco, Yucatán, Oaxaca)

Por lo que las regiones quedaron de la siguiente manera:

REGIÓN NORTE	REGIÓN CENTRO	REGIÓN SUR
<ul style="list-style-type: none">• Baja California• Baja California Sur• Chihuahua• Coahuila• Durango• Nuevo León• San Luis Potosí• Sinaloa• Sonora• Tamaulipas	<ul style="list-style-type: none">• Aguascalientes• Ciudad de México• Colima• Guanajuato• Jalisco• Michoacán• Nayarit• Querétaro• Zacatecas	<ul style="list-style-type: none">• Campeche• Chiapas• Estado de México• Guerrero• Hidalgo• Morelos• Oaxaca• Puebla• Quintana Roo• Tabasco• Tlaxcala• Veracruz• Yucatán

SCRIPT SQL DE CADA BASE DE DATOS POR REGIÓN

Norte

```
drop database NORTEp3
go

create database NORTEp3
go

use NORTEp3
go

CREATE TABLE NORTE(
    [FECHA_ACTUALIZACION] [nvarchar](15) NULL,
    [ID_REGISTRO] [varchar](15) NOT NULL,
    [ORÍGEN] [int] NULL,
    [SECTOR] [int] NULL,
    [ENTIDAD_UM] [nvarchar](15) NULL,
    [SEXO] [int] NULL,
    [ENTIDAD_NAC] [nvarchar](15) NULL,
    [ENTIDAD_RES] [nvarchar](15) NULL,
    [MUNICIPIO_RES] [nvarchar](15) NULL,
    [TIPO_PACIENTE] [int] NULL,
    [FECHA_INGRESO] [nvarchar](15) NULL,
    [FECHA_SINTOMAS] [nvarchar](15) NULL,
    [FECHA_DEF] [nvarchar](15) NULL,
    [INTUBADO] [int] NULL,
    [NEUMONIA] [int] NULL,
    [EDAD] [nvarchar](7) NULL,
    [NACIONALIDAD] [int] NULL,
    [EMBARAZO] [int] NULL,
    [HABLA LENGUA INDIG] [int] NULL,
    [INDIGENA] [int] NULL,
    [DIABETES] [int] NULL,
    [EPOC] [int] NULL,
    [ASMA] [int] NULL,
    [INMUSUPR] [int] NULL,
    [HIPERTENSION] [int] NULL,
    [OTRA_COM] [int] NULL,
    [CARDIOVASCULAR] [int] NULL,
    [OBESIDAD] [int] NULL,
    [RENAL_CRONICA] [int] NULL,
    [TABAQUISMO] [int] NULL,
    [OTRO_CASO] [int] NULL,
    [TOMA_MUESTRA_LAB] [int] NULL,
    [RESULTADO_LAB] [int] NULL,
    [TOMA_MUESTRA_ANTIGENO] [int] NULL,
    [RESULTADO_ANTIGENO] [int] NULL,
    [CLASIFICACION_FINAL] [int] NULL,
    [MIGRANTE] [int] NULL,
    [PAIS_NACIONALIDAD] [nvarchar](50) NULL,
```

```

        [PAIS_ORIGEN] [nvarchar](50) NULL,
        [UCI] [nvarchar](50) NULL
    );

INSERT INTO NORTE
SELECT * FROM covidHistorico.dbo.datoscovid
WHERE ENTIDAD_UM IN
('02','03','05','08','10','19','24','25','26','28');

```

Centro

```

drop database CENTROp3
go

create database CENTROp3
go

use CENTROp3
go

CREATE TABLE CENTRO(
    [FECHA_ACTUALIZACION] [nvarchar](15) NULL,
    [ID_REGISTRO] [varchar](15) NOT NULL,
    [ORIGEN] [int] NULL,
    [SECTOR] [int] NULL,
    [ENTIDAD_UM] [nvarchar](15) NULL,
    [SEXO] [int] NULL,
    [ENTIDAD_NAC] [nvarchar](15) NULL,
    [ENTIDAD_RES] [nvarchar](15) NULL,
    [MUNICIPIO_RES] [nvarchar](15) NULL,
    [TIPO_PACIENTE] [int] NULL,
    [FECHA_INGRESO] [nvarchar](15) NULL,
    [FECHA_SINTOMAS] [nvarchar](15) NULL,
    [FECHA_DEF] [nvarchar](15) NULL,
    [INTUBADO] [int] NULL,
    [NEUMONIA] [int] NULL,
    [EDAD] [nvarchar](7) NULL,
    [NACIONALIDAD] [int] NULL,
    [EMBARAZO] [int] NULL,
    [HABLA LENGUA_INDIG] [int] NULL,
    [INDIGENA] [int] NULL,
    [DIABETES] [int] NULL,
    [EPOC] [int] NULL,
    [ASMA] [int] NULL,
    [INMUSUPR] [int] NULL,
    [HIPERTENSION] [int] NULL,
    [OTRA_COM] [int] NULL,
    [CARDIOVASCULAR] [int] NULL,
    [OBESIDAD] [int] NULL,
    [RENAL_CRONICA] [int] NULL,
    [TABAQUISMO] [int] NULL,
    [OTRO_CASO] [int] NULL,

```

```

        [TOMA_MUESTRA_LAB] [int] NULL,
        [RESULTADO_LAB] [int] NULL,
        [TOMA_MUESTRA_ANTIGENO] [int] NULL,
        [RESULTADO_ANTIGENO] [int] NULL,
        [CLASIFICACION_FINAL] [int] NULL,
        [MIGRANTE] [int] NULL,
        [PAIS_NACIONALIDAD] [nvarchar](50) NULL,
        [PAIS_ORIGEN] [nvarchar](50) NULL,
        [UCI] [nvarchar](50) NULL
    );

INSERT INTO CENTRO
SELECT * FROM covidHistorico.dbo.datoscovid
WHERE ENTIDAD_UM IN
('01','06','09','11','13','14','16','17','18','22','32')

```

Sur

```

drop database SURp3
go

create database SURp3
go

use SURp3
go

CREATE TABLE SUR(
    [FECHA_ACTUALIZACION] [nvarchar](15) NULL,
    [ID_REGISTRO] [varchar](15) NOT NULL,
    [ORIGEN] [int] NULL,
    [SECTOR] [int] NULL,
    [ENTIDAD_UM] [nvarchar](15) NULL,
    [SEXO] [int] NULL,
    [ENTIDAD_NAC] [nvarchar](15) NULL,
    [ENTIDAD_RES] [nvarchar](15) NULL,
    [MUNICIPIO_RES] [nvarchar](15) NULL,
    [TIPO_PACIENTE] [int] NULL,
    [FECHA_INGRESO] [nvarchar](15) NULL,
    [FECHA_SINTOMAS] [nvarchar](15) NULL,
    [FECHA_DEF] [nvarchar](15) NULL,
    [INTUBADO] [int] NULL,
    [NEUMONIA] [int] NULL,
    [EDAD] [nvarchar](7) NULL,
    [NACIONALIDAD] [int] NULL,
    [EMBARAZO] [int] NULL,
    [HABLA LENGUA INDIG] [int] NULL,
    [INDIGENA] [int] NULL,
    [DIABETES] [int] NULL,
    [EPOC] [int] NULL,

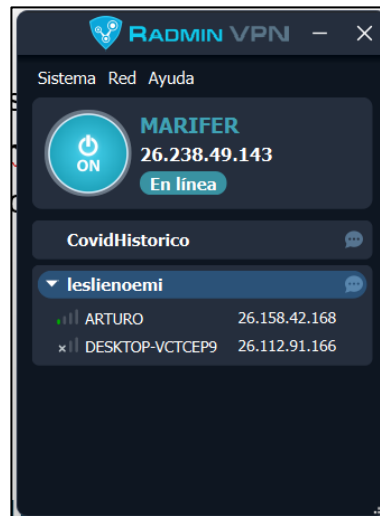
```

```
[ASMA] [int] NULL,  
[INMUSUPR] [int] NULL,  
[HIPERTENSION] [int] NULL,  
[OTRA_COM] [int] NULL,  
[CARDIOVASCULAR] [int] NULL,  
[OBESIDAD] [int] NULL,  
[RENAL_CRONICA] [int] NULL,  
[TABAQUISMO] [int] NULL,  
[OTRO_CASO] [int] NULL,  
[TOMA_MUESTRA_LAB] [int] NULL,  
[RESULTADO_LAB] [int] NULL,  
[TOMA_MUESTRA_ANTIGENO] [int] NULL,  
[RESULTADO_ANTIGENO] [int] NULL,  
[CLASIFICACION_FINAL] [int] NULL,  
[MIGRANTE] [int] NULL,  
[PAIS_NACIONALIDAD] [nvarchar](50) NULL,  
[PAIS_ORIGEN] [nvarchar](50) NULL,  
[UCI] [nvarchar](50) NULL  
);  
  
INSERT INTO SUR  
SELECT * FROM covidHistorico.dbo.datoscovid  
WHERE ENTIDAD_UM IN  
('04','07','12','15','20','21','23','27','29','30','31')
```


ESTRATEGIAS DE CONECTIVIDAD PARA LA DISTRIBUCIÓN

Para poder hacer la conectividad entre maquinas que contuvieran los nodos con los fragmentos de las bases de datos utilizamos la aplicación de **RadminVPN** para lograr que las IPs estuvieran en la misma red local sin necesidad de estar configurando los ODBC al conectarlo por la reasignación de IPs.

Se creo una red llamada “CovidHistorico” con su respectiva contraseña para restringir el acceso a nuestra red ‘local’ con nuestras IPs asignadas (que están mostradas en la imagen) lo que nos permitió conectar desde diferentes ubicaciones sin importar la red de conexión.

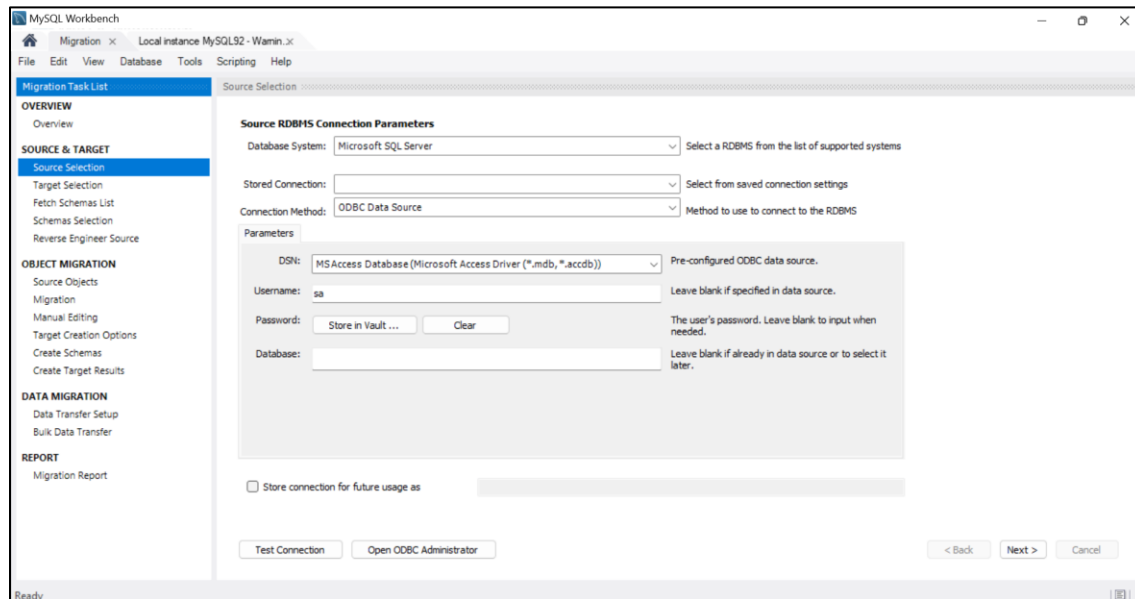


En esta tabla se muestran las direcciones IP asignadas para la conexión ‘local’ de nuestros nodos:

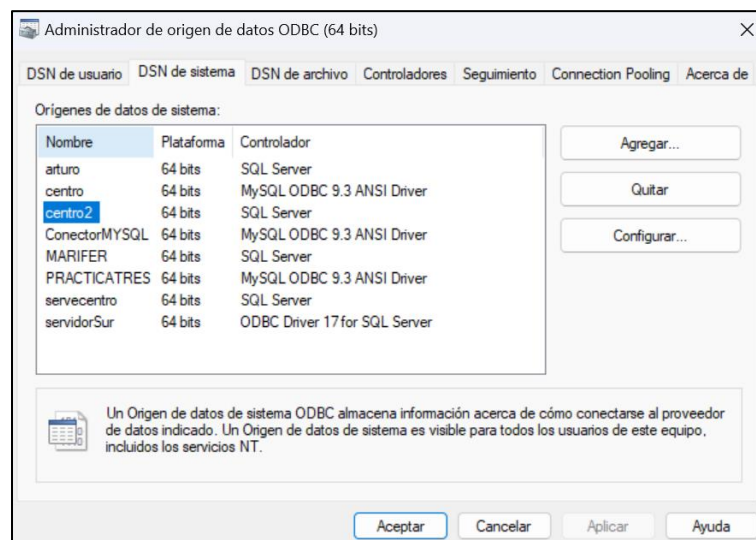
Alumno	IP
Maria Fernanda	26.238.49.143
Arturo Daniel	26.158.42.168
Leslie Noemi	26.112.91.166

MIGRACION DE DATOS

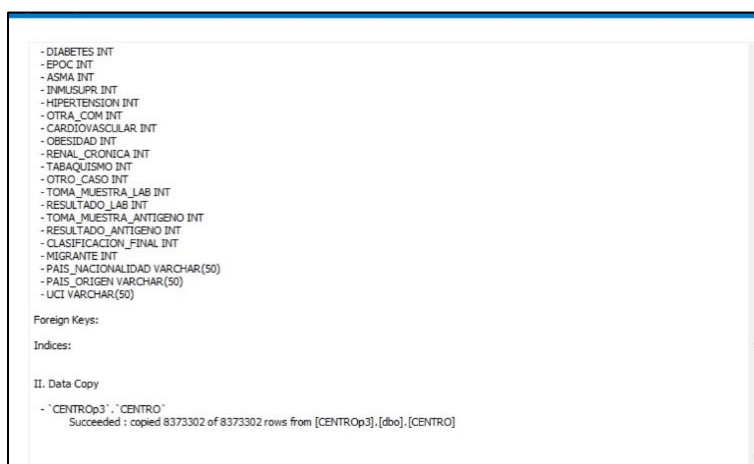
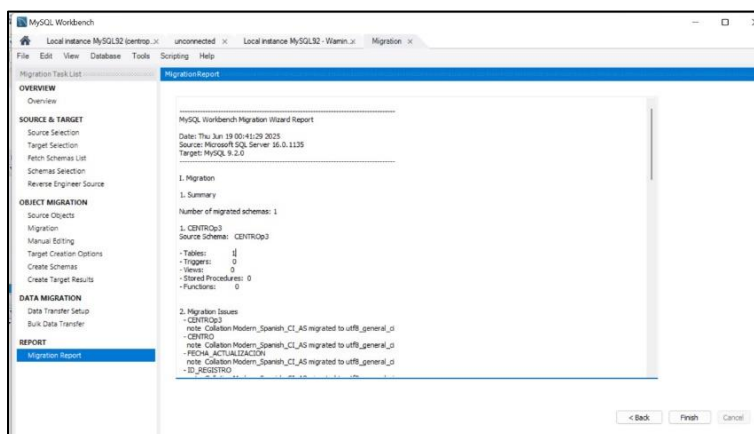
Para poder lograr la parte de la migración de datos en MySQL, se tuvo que hacer una configuración del MySQL Workbench, en donde tenemos que cambiar ciertos parámetros para hacer dicha migración, como se ve en la siguiente imagen.



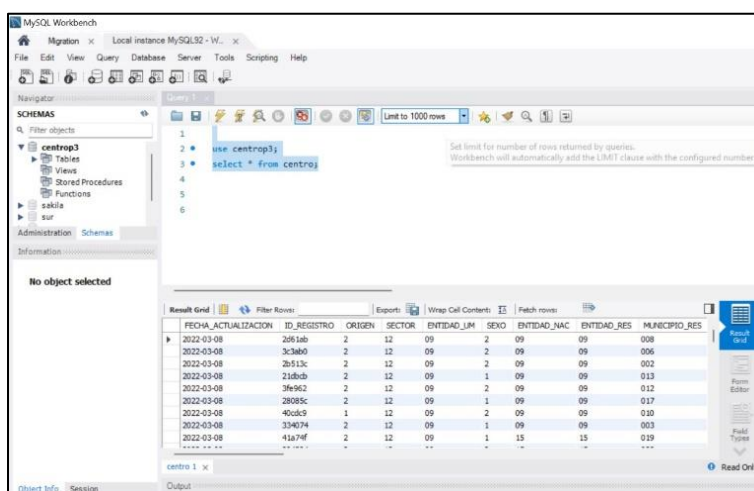
Pero antes de eso, tenemos que agregar un DNS del sistema, para poder almacenar información en otra plataforma.



Una vez configurado, la parte de migración, tendremos que esperar para que los datos migren correctamente y podamos tener un informe de migración, como el que tenemos a continuación.



Al final comprobamos que todo este correcto, a través de un script sencillo, para poder ver los datos de la base de datos.



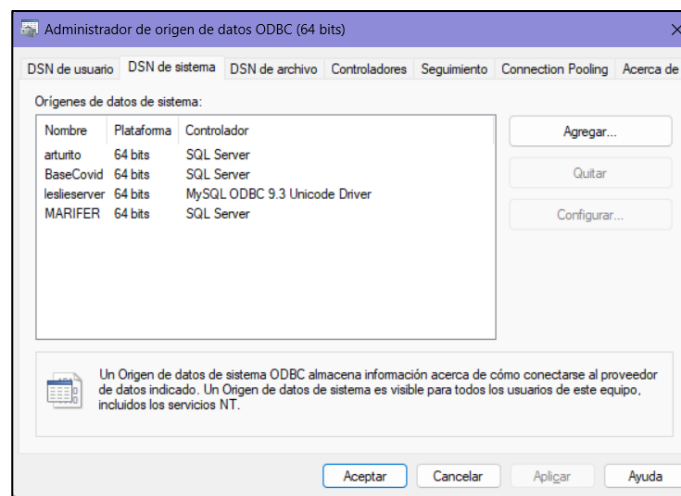
LISTADO DE NOMBRE DE SERVIDORES VINCULADOS

General

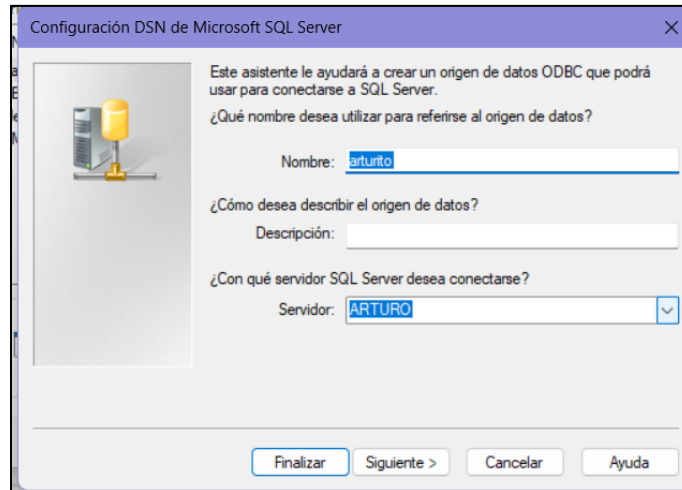
Conexión	María Fernanda (SQL SERVER)	Arturo Daniel (SQL SERVER)	Leslie Noemi (MY SQL)
Autor			
María Fernanda	LOCAL MARIFER	ARTURITOPRUEBA	SERVERLESLIE
Arturo Daniel	SERVMAFER	LOCAL ARTURO	SERVLESLIE
Leslie Noemi	MARYNORTE	ARTUSUR	LOCAL 26.112.91.166

Esta tabla de doble entrada nos muestra los nombres de los servidores vinculados (Linked servers) que usó cada integrante para poder hacer la conexión con los otros nodos, así como el método de acceso

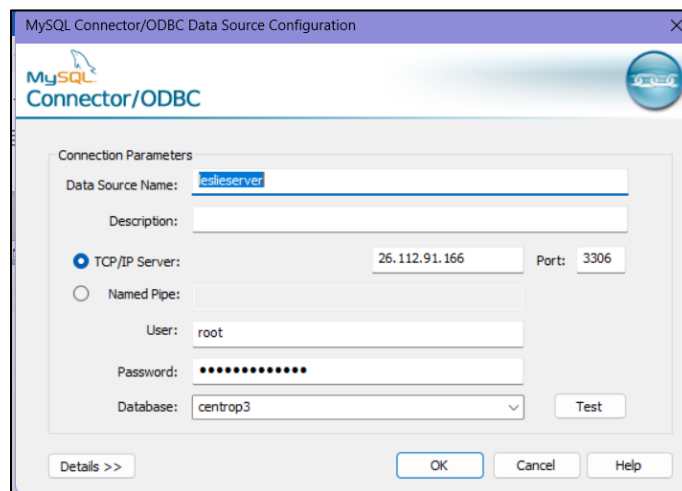
Muestra de alguna de las conexiones en ODBC donde se puede visualizar los DNS del sistema de los tres nodos.



A continuación, se presenta la configuración de los servidores, este primero es en SQL SERVER (Configuración DNS de Microsoft SQL Server).

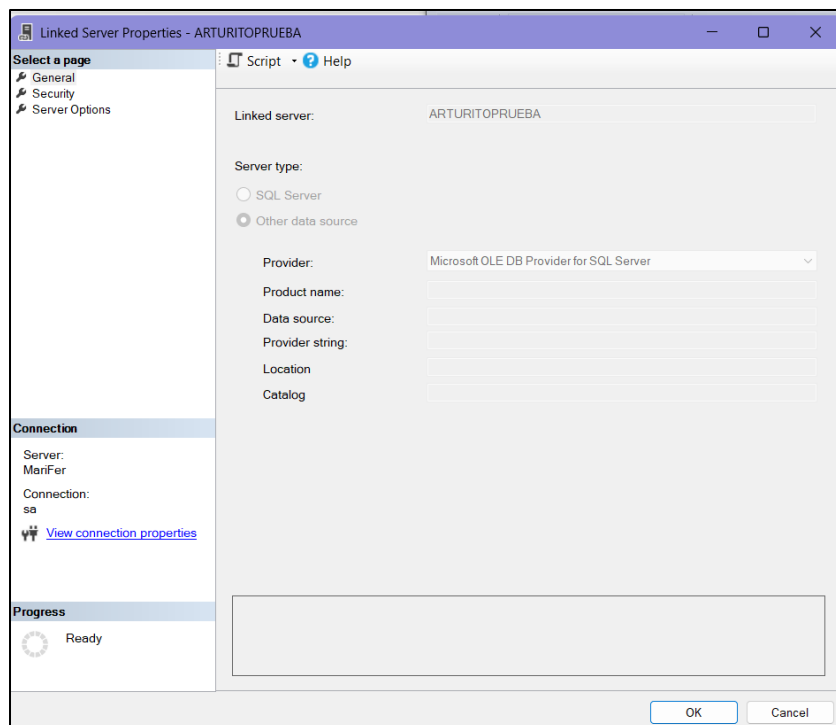
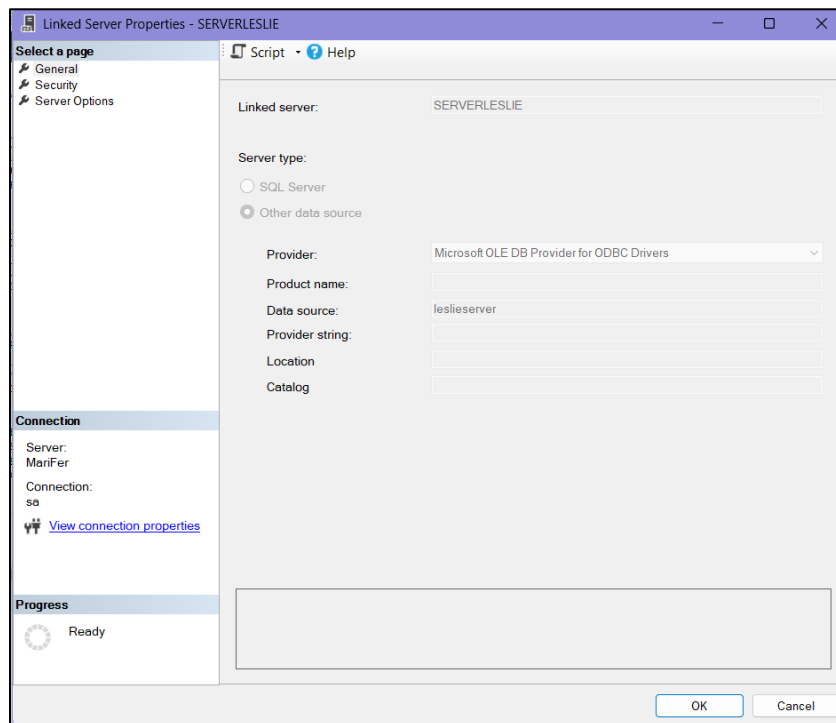


Esta es ODBC Data Source Configuration de MYSQL, la cual utilizamos desde los nodos de SQL SERVER para establecer la conexión conMYSQL:



La información siguiente especifica las propiedades de los Linked Server a los que nos conectamos cada uno de los integrantes del equipo:

Legorreta Rodríguez María Fernanda



Macías Galván Arturo Daniel

Linked Server Properties - SERVLESIE

Select a page: General, Security, Server Options

Script ? Help

Linked server: SERVLESIE

Server type:
☐ SQL Server
☒ Other data source

Provider: Microsoft OLE DB Provider for ODBC Drivers

Product name:

Data source: leslie

Provider string:

Location

Catalog

Connection
Server: Arturo
Connection: sa
[View connection properties](#)

Progress
Ready

OK Cancel

Linked Server Properties - SERVMAFER

Select a page: General, Security, Server Options

Script ? Help

Linked server: SERVMAFER

Server type:
☐ SQL Server
☒ Other data source

Provider: Microsoft OLE DB Provider for SQL Server

Product name:

Data source:

Provider string:

Location

Catalog

Connection
Server: Arturo
Connection: sa
[View connection properties](#)

Progress
Ready

OK Cancel

Palacios Reyes Leslie Noemi

Linked Server Properties - ARTUSUR

Select a page: General, Security, Server Options

Script Help

Linked server: ARTUSUR

Server type:
☐ SQL Server
☒ Other data source

Provider: Microsoft OLE DB Provider for SQL Server

Product name:

Data source:

Provider string:

Location:

Catalog:

Connection

Server: DESKTOP-VCTCEP9

Connection: sa

[View connection properties](#)

Progress

Ready

OK Cancel

Linked Server Properties - MARINORTE

Select a page: General, Security, Server Options

Script Help

Linked server: MARINORTE

Server type:
☐ SQL Server
☒ Other data source

Provider: Microsoft OLE DB Provider for SQL Server

Product name:

Data source:

Provider string:

Location:

Catalog:

Connection

Server: DESKTOP-VCTCEP9

Connection: sa

[View connection properties](#)

Progress

Ready

OK Cancel

DESCRIPCIÓN DE LAS CONSULTAS MODIFICADAS:

Consulta 3:

Listar el porcentaje de casos confirmados en cada una de las siguientes morbilidades a nivel nacional: diabetes, obesidad e hipertensión.

Consulta 4:

Listar los municipios que no tengan casos confirmados en todas las morbilidades.

Consulta 5:

Listar los estados con más casos recuperados con neumonía.

Consulta 7:

Para el año 2020 y 2021 cuál fue el mes con más casos registrados, confirmados, sospechosos, por estado registrado en la base de datos.

SCRIPT DE CONSULTAS

```
--CONSULTA 3:

SELECT morbilidad,
       100.0 * SUM(casos) / SUM(total_confirmados) AS porcentaje
FROM (
    -- NORTE
    SELECT 'diabetes' AS morbilidad, SUM(CASE WHEN diabetes = 1 THEN
1 ELSE 0 END) AS casos, COUNT(*) AS total_confirmados
    FROM nortep3.dbo.norte
    WHERE clasificacion_final IN (1, 2, 3)
    UNION ALL
    SELECT 'obesidad', SUM(CASE WHEN obesidad = 1 THEN 1 ELSE 0
END), COUNT(*)
    FROM nortep3.dbo.norte
    WHERE clasificacion_final IN (1, 2, 3)
    UNION ALL
    SELECT 'hipertensión', SUM(CASE WHEN hipertension = 1 THEN 1
ELSE 0 END), COUNT(*)
    FROM nortep3.dbo.norte
    WHERE clasificacion_final IN (1, 2, 3)

    -- SUR
    UNION ALL
    SELECT 'diabetes', SUM(CASE WHEN diabetes = 1 THEN 1 ELSE 0
END), COUNT(*)
```

```

FROM OPENQUERY(ARTURITOPRUEBA, '
    SELECT diabetes, clasificacion_final
    FROM SURp3.dbo.SUR
    WHERE clasificacion_final IN (1, 2, 3)
')
UNION ALL
SELECT 'obesidad', SUM(CASE WHEN obesidad = 1 THEN 1 ELSE 0
END), COUNT(*)
FROM OPENQUERY(ARTURITOPRUEBA, '
    SELECT obesidad, clasificacion_final
    FROM SURp3.dbo.SUR
    WHERE clasificacion_final IN (1, 2, 3)
')
UNION ALL
SELECT 'hipertensión', SUM(CASE WHEN hipertension = 1 THEN 1
ELSE 0 END), COUNT(*)
FROM OPENQUERY(ARTURITOPRUEBA, '
    SELECT hipertension, clasificacion_final
    FROM SURp3.dbo.SUR
    WHERE clasificacion_final IN (1, 2, 3)
')

-- CENTRO
UNION ALL
SELECT 'diabetes', SUM(CASE WHEN diabetes = 1 THEN 1 ELSE 0
END), COUNT(*)
FROM OPENQUERY(SERVERLESLIE, '
    SELECT diabetes, clasificacion_final
    FROM centrop3.centro
    WHERE clasificacion_final IN (1, 2, 3)
')
UNION ALL
SELECT 'obesidad', SUM(CASE WHEN obesidad = 1 THEN 1 ELSE 0
END), COUNT(*)
FROM OPENQUERY(SERVERLESLIE, '
    SELECT obesidad, clasificacion_final
    FROM centrop3.centro
    WHERE clasificacion_final IN (1, 2, 3)
')
UNION ALL
SELECT 'hipertensión', SUM(CASE WHEN hipertension = 1 THEN 1
ELSE 0 END), COUNT(*)
FROM OPENQUERY(SERVERLESLIE, '
    SELECT hipertension, clasificacion_final
    FROM centrop3.centro
    WHERE clasificacion_final IN (1, 2, 3)
')
) AS morbilidades
GROUP BY morbilidad;

```

--CONSULTA 4:

```
SELECT MUNICIPIO_RES
```

```

FROM (
    -- Nodo NORTE (local)
    SELECT MUNICIPIO_RES, HIPERTENSION, OBESIDAD, DIABETES,
    TABAQUISMO
    FROM nortep3.dbo.norte
    WHERE CLASIFICACION_FINAL IN (1, 2, 3)

    UNION ALL

    -- Nodo SUR
    SELECT MUNICIPIO_RES, HIPERTENSION, OBESIDAD, DIABETES,
    TABAQUISMO
    FROM OPENQUERY(ARTURITOPRUEBA, '
        SELECT MUNICIPIO_RES, HIPERTENSION, OBESIDAD, DIABETES,
    TABAQUISMO
    FROM SURp3.dbo.SUR
    WHERE CLASIFICACION_FINAL IN (1, 2, 3)
    ')

    UNION ALL

    -- Nodo CENTRO
    SELECT MUNICIPIO_RES, HIPERTENSION, OBESIDAD, DIABETES,
    TABAQUISMO
    FROM OPENQUERY(SERVERLESLIE, '
        SELECT MUNICIPIO_RES, HIPERTENSION, OBESIDAD, DIABETES,
    TABAQUISMO
    FROM centrop3.centro
    WHERE CLASIFICACION_FINAL IN (1, 2, 3)
    ')
) AS datos_unidos
GROUP BY MUNICIPIO_RES
HAVING SUM(CASE WHEN HIPERTENSION = 1 THEN 1 ELSE 0 END) = 0
    AND SUM(CASE WHEN OBESIDAD = 1 THEN 1 ELSE 0 END) = 0
    AND SUM(CASE WHEN DIABETES = 1 THEN 1 ELSE 0 END) = 0
    AND SUM(CASE WHEN TABAQUISMO = 1 THEN 1 ELSE 0 END) = 0;

```

--CONSULTA 5:

```

SELECT ENTIDAD_UM, COUNT(*) AS Casos_recuperados
FROM (
    -- Nodo NORTE (local)
    SELECT ENTIDAD_UM
    FROM nortep3.dbo.norte
    WHERE NEUMONIA = 1 AND FECHA_DEF = '9999-99-99'
    UNION ALL
    -- Nodo SUR ARTURO
    SELECT ENTIDAD_UM
    FROM OPENQUERY(ARTURITOPRUEBA, '
        SELECT ENTIDAD_UM
    FROM SURp3.dbo.SUR
    WHERE NEUMONIA = 1 AND FECHA_DEF = ''9999-99-99''
    ')

```

```

UNION ALL
-- Nodo CENTRO LESLIE
SELECT ENTIDAD_UM
FROM OPENQUERY(SERVERLESLIE, '
    SELECT ENTIDAD_UM
    FROM centrop3.centro
    WHERE NEUMONIA = 1 AND FECHA_DEF = ''9999-99-99''
')
) AS datos_unidos
GROUP BY ENTIDAD_UM
ORDER BY Casos_recuperados DESC;

--CONSULTA 7:

SELECT TOP 1 *
FROM (
    -- CENTRO (MySQL)
    SELECT sub.Año, sub.Mes, sub.ENTIDAD_RES, COUNT(*) AS
Total_Casos
    FROM OPENQUERY(CENTRO, '
        SELECT DATE_FORMAT(FECHA_INGRESO, ''%Y'') AS Año,
        DATE_FORMAT(FECHA_INGRESO, ''%m'') AS Mes,
        ENTIDAD_RES,
        CLASIFICACION_FINAL
        FROM centrop3.centro
        WHERE FECHA_INGRESO BETWEEN ''2020-01-01'' AND ''2021-12-
31''
        AND CLASIFICACION_FINAL IN (1, 2, 3, 6)
    ') AS sub
    WHERE sub.CLASIFICACION_FINAL IN (1, 2, 3, 6)
    GROUP BY sub.Año, sub.Mes, sub.ENTIDAD_RES

    UNION ALL

    -- NORTE (SQL Server)
    SELECT YEAR(FECHA_INGRESO) AS Año, MONTH(FECHA_INGRESO) AS Mes,
ENTIDAD_RES, COUNT(*) AS Total_Casos
    FROM OPENQUERY(MARINORTE, '
        SELECT FECHA_INGRESO, ENTIDAD_RES, CLASIFICACION_FINAL
        FROM NORTEp3.dbo.NORTE
        WHERE YEAR(FECHA_INGRESO) IN (2020, 2021)
        AND CLASIFICACION_FINAL IN (1, 2, 3, 6)
    ')
    GROUP BY YEAR(FECHA_INGRESO), MONTH(FECHA_INGRESO), ENTIDAD_RES

    UNION ALL

    -- SUR (SQL Server)
    SELECT YEAR(FECHA_INGRESO) AS Año, MONTH(FECHA_INGRESO) AS Mes,
ENTIDAD_RES, COUNT(*) AS Total_Casos
    FROM OPENQUERY(ARTUSUR, '
        SELECT FECHA_INGRESO, ENTIDAD_RES, CLASIFICACION_FINAL
        FROM SURp3.dbo.SUR

```

```

WHERE YEAR(FECHA_INGRESO) IN (2020, 2021)
      AND CLASIFICACION_FINAL IN (1, 2, 3, 6)
')
GROUP BY YEAR(FECHA_INGRESO), MONTH(FECHA_INGRESO), ENTIDAD_RES
) AS todos
ORDER BY Total_Casos DESC;

```

RESULTADOS DE LAS CONSULTAS:

Consulta 3:

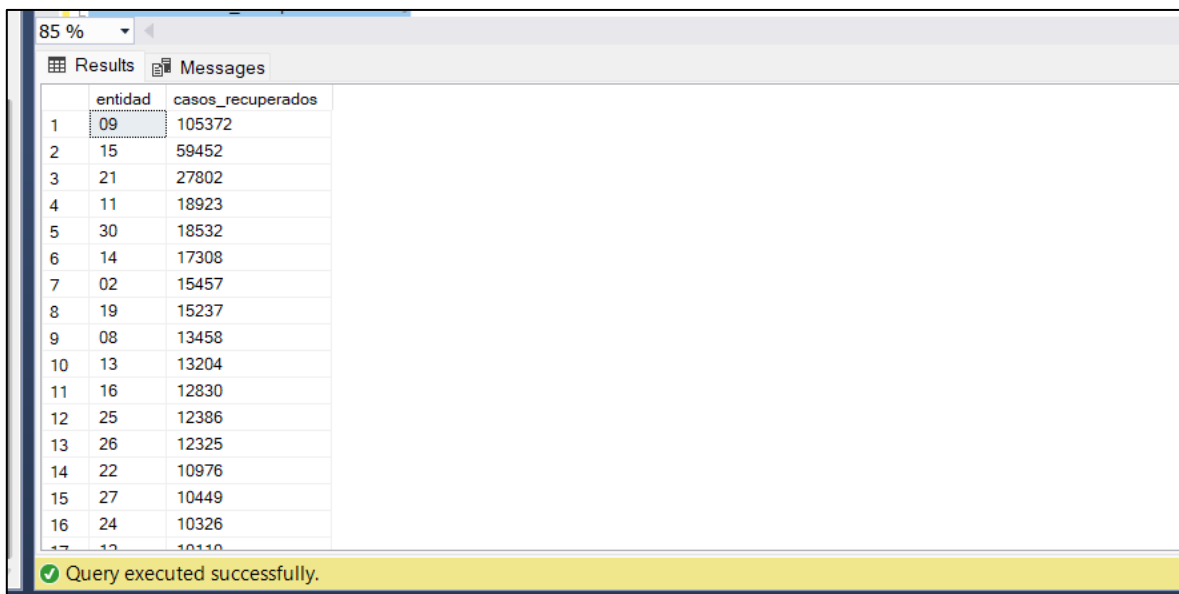
Results Messages		
	morbilidad	porcentaje
1	diabetes	9.617175382487
2	hipertensión	12.787018025657
3	obesidad	10.567170432354

Consulta 4:

Results Messages	
MUNICIPIO_RES	
1	566
2	218
3	275
4	444
5	317
6	481
7	527
8	451
9	328
10	529
11	371
12	512
13	491
14	521
15	408
16	374
17	473

Query executed successfully.

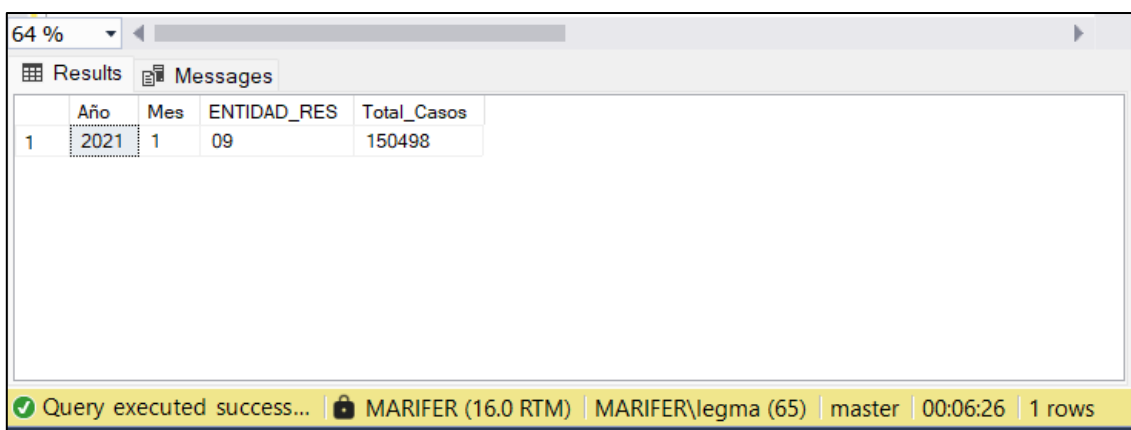
Consulta 5:



	entidad	casos_recuperados
1	09	105372
2	15	59452
3	21	27802
4	11	18923
5	30	18532
6	14	17308
7	02	15457
8	19	15237
9	08	13458
10	13	13204
11	16	12830
12	25	12386
13	26	12325
14	22	10976
15	27	10449
16	24	10326
17	12	10110

Query executed successfully.

Consulta 7:



	Año	Mes	ENTIDAD_RES	Total_Casos
1	2021	1	09	150498

Query executed success... | MARIFER (16.0 RTM) | MARIFER\legma (65) | master | 00:06:26 | 1 rows

CONCLUSIONES

Legorreta Rodríguez María Fernanda

Esta práctica me ayudó a comprender de forma muy clara lo útil y poderosa que puede ser una base de datos distribuida, sobre todo cuando se fragmenta estratégicamente según la región. Me pareció muy interesante ver cómo al repartir la carga entre diferentes servidores no solo se evita la saturación de uno solo, sino que también se gana eficiencia al limitar las búsquedas únicamente a la zona que nos interesa. Esto facilita el análisis y mejora bastante el rendimiento en consultas específicas. Además, el hecho de que hayamos logrado conectar nuestras computadoras usando una VPN fue una experiencia muy valiosa. Nunca había trabajado directamente con una red privada virtual, y me sorprendió lo mucho que se siente como si todos estuviéramos en la misma red local, aunque cada uno estuviera en su casa.

Una parte que me llamó mucho la atención fue la comparación entre OPENQUERY y las consultas de cuatro partes. Al principio pensaba que OPENQUERY iba a ser más rápido por ir directo al servidor, pero al hacer pruebas noté que en realidad tarda más, sobre todo cuando se consulta en varios linked servers al mismo tiempo. Por otro lado, las consultas de cuatro partes funcionaron mejor cuando se usaron directamente sobre servidores SQL Server, y fue notorio que optimizan más el tiempo de respuesta cuando todo está bien estructurado.

Macías Galván Arturo Daniel

Personalmente esta práctica me agrado mucho, debido a que la interconexión que se realizó en el equipo, aunque generó inconsistencias y desaciertos, el objetivo descrito fue cumplido. Principalmente la configuración de direcciones causó problemas, al igual que la partición, sin embargo, esta última fue por tiempo y recursos; no obstante, las consultas no generaron conflicto alguno pues solo fueron modificaciones a las consultas realizadas con anterioridad.

Como primera instancia, las consultas fueron creadas usando la sentencia *openquery* para la conexión con el servidor con el nodo MYSQL y el servidor MSSQL SQL SERVER, después con una investigación se llegó al acuerdo de usar el nombramiento de 4 partes para este último mostrando que la ejecución de esta manera mejora el rendimiento, no tanto como se esperaba, pero es una mejora considerable, con un aproximado de un 20% menos por cada subconsulta para un nodo de MSSQL SQL SERVER.

Palacios Reyes Leslie Noemi

Como parte del equipo, puedo concluir que La fragmentación de datos en nodos distribuidos por región como Centro(nodo de MYSQL), Norte y Sur representa una estrategia eficaz para el manejo de grandes volúmenes de información al utilizar consultas con OPENQUERY se logra integrar resultados desde distintas fuentes remotas lo que optimiza el rendimiento (pero no tanto como el proceso de 4 partes) general del sistema y reduce la carga sobre el nodo principal permitiendo además la especialización por región que favorece un análisis más preciso y segmentado así como una mayor escalabilidad ya que los datos no se concentran en un solo punto lo que facilita su crecimiento y mantenimiento al mismo tiempo mejora la eficiencia de las consultas al aplicar filtros directamente en el origen evitando transferencias innecesarias y favoreciendo la interoperabilidad entre distintas tecnologías como SQL Server y MySQL mediante el uso de Linked Server.