

# Propuesta de fragmentación

Bases de Datos Distribuidas

3TM3

Equipo 3:

Ramirez Martinez Brayan Gustavo

## Fragmentación

Gracias a la sugerencia del profesor decidí implementar la fragmentación a partir de SalesTerritory y de la columna TerritoryID, ya que proporciona un “límite” más visible para cada fragmento, en este caso, los fragmentos son de fácilmente separables en Norteamérica, Europa y Asia Pacífico, esto además de las tablas que no se fragmentan pero que se siguen necesitando (esto lo observamos gracias a las 10 consultas sobre las que se trabajó), dichos fragmentos cumplen con las reglas de fragmentación por que todos los elementos están repartidos en los distintos fragmentos, la tabla original puede reconstruirse insertando los datos de los fragmentos en las tablas originales y no se encuentran datos repetidos en las fragmentos.

## Predicados para customer

PrCustomer={

P1: TerritoryID = 1

...

P10: TerritoryID=10

}

MCustomer= {

M1:  $P1 \wedge P2 \wedge P3 \wedge P4 \wedge P5 \wedge P6 \wedge \neg(P7) \wedge \neg(P8) \wedge \neg(P9) \wedge \neg(P10)$

M2:  $\neg(P1) \wedge \neg(P2) \wedge \neg(P3) \wedge \neg(P4) \wedge \neg(P5) \wedge \neg(P6) \wedge P7 \wedge P8 \wedge \neg(P9) \wedge \neg(P10)$

M3:  $\neg(P1) \wedge \neg(P2) \wedge \neg(P3) \wedge \neg(P4) \wedge \neg(P5) \wedge \neg(P6) \wedge \neg(P7) \wedge \neg(P8) \wedge P9 \wedge \neg(P10)$

M4:  $\neg(P1) \wedge \neg(P2) \wedge \neg(P3) \wedge \neg(P4) \wedge \neg(P5) \wedge \neg(P6) \wedge \neg(P7) \wedge \neg(P8) \wedge \neg(P9) \wedge \neg(P10)$

}

Frag1\_Customer =TerritoryID1-6(Customer)

SELECT\*FROM Customer where TerritoryID BETWEEN 1 and 6

Frag2\_Customer =TerritoryID=7,8,10(Customer)

SELECT\*FROM Customer where TerritoryID BETWEEN 7 and 10 and TerritoryID!=9

Frag3\_Customer =TerritoryID=9(Customer)

SELECT\*FROM Customer where TerritoryID = 9

Frag4\_Customer =TerritoryID>10

SELECT\*FROM Customer where TerritoryID > 10

### Fragmentación Horizontal de SalesOrderHeader

Frag1\_SOH= SalesOrderHeader  $\times$  Frag1\_Customer

SELECT \* into NorthAmerica.SalesOrderHeader FROM Sales.SalesOrderHeader  
WHERE CustomerID IN

(SELECT CustomerID FROM Sales.Customer WHERE TerritoryID BETWEEN 1 AND 6)

Frag2\_SOH= SalesOrderHeader  $\times$  Frag2\_Customer

SELECT \* into NorthAmerica.SalesOrderHeader FROM Sales.SalesOrderHeader  
WHERE CustomerID IN (SELECT CustomerID FROM Sales.Customer WHERE  
TerritoryID BETWEEN 7 AND 10 and TerritoryID!=9)

Frag3\_SOH= SalesOrderHeader  $\times$  Frag3\_Customer

SELECT \* into NorthAmerica.SalesOrderHeader FROM Sales.SalesOrderHeader  
WHERE CustomerID IN (SELECT CustomerID FROM Sales.Customer WHERE  
TerritoryID =10)

Frag4\_SOH= SalesOrderHeader  $\times$  Frag4\_Customer

SELECT \* into NorthAmerica.SalesOrderHeader FROM Sales.SalesOrderHeader  
WHERE CustomerID IN (SELECT CustomerID FROM Sales.Customer WHERE  
TerritoryID>10)

### Fragmentación Horizontal de SalesOrderDetail con base en SalesOrderHeader

Frag1\_SOD= SalesOrderDetail  $\times$  Frag1\_SOH

select

(sod.SalesOrderID,sod.SalesOrderDetailID,sod.CarrierTrackingNumber,sod.OrderQty,s  
od.ProductID,sod.SpecialOfferID,sod.UnitPrice,sod.UnitPriceDiscount,sod.LineTotal,so  
d.rowguid,sod.ModifiedDate) into NorthAmerica.Sales.SalesOrderDetail from

AdventureWorks2019.Sales.SalesOrderDetail as sod left join (select \* from NorthAmerica.Sales.SalesOrderHeader) as oh on sod.SalesOrderID=oh.SalesOrderID

Frag2\_SOD= SalesOrderDetail  $\times$  Frag2\_SOH

select

(sod.SalesOrderID,sod.SalesOrderDetailID,sod.CarrierTrackingNumber,sod.OrderQty,sod.ProductID,sod.SpecialOfferID,sod.UnitPrice,sod.UnitPriceDiscount,sod.LineTotal,sod.rowguid,sod.ModifiedDate) into Europe.Sales.SalesOrderDetail from AdventureWorks2019.Sales.SalesOrderDetail as sod left join (select \* from Europe.Sales.SalesOrderHeader) as oh on sod.SalesOrderID=oh.SalesOrderID

Frag3\_SOD= SalesOrderDetail  $\times$  Frag3\_SOH

select

(sod.SalesOrderID,sod.SalesOrderDetailID,sod.CarrierTrackingNumber,sod.OrderQty,sod.ProductID,sod.SpecialOfferID,sod.UnitPrice,sod.UnitPriceDiscount,sod.LineTotal,sod.rowguid,sod.ModifiedDate) into AsiaPacific.Sales.SalesOrderDetail from AdventureWorks2019.Sales.SalesOrderDetail as sod left join (select \* from AsiaPacific.Sales.SalesOrderHeader) as oh on sod.SalesOrderID=oh.SalesOrderID

## Configuración de los fragmentos en las instancias SQL

A partir de lo analizado previamente se decide la creación de tres fragmentos (se tiene un cuarto, pero no se implementa ya que queda fuera del uso de las consultas propuestas) estos fragmentos se colocaron en dos instancias de SQL Server,

Los fragmentos de Europa y Norte América se asignan en la instancia 2:

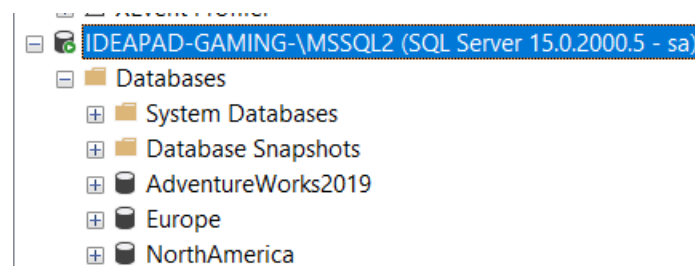


Figure 1 Fragmentos de la instancia 2

El fragmento restante (Asia pacifico) se asigna en la instancia 1, la cual es accesible mediante un servidor vinculado:

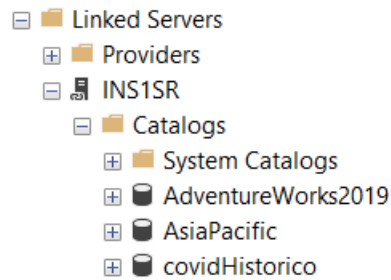


Figure 2 Vista desde el servidor vinculado

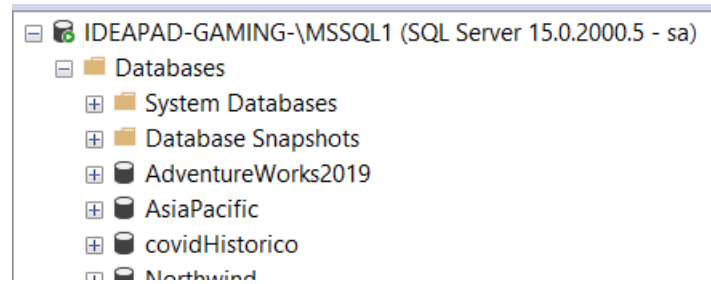


Figure 3 Vista de la base en la parte local\

Adicional a esto considero importante constatar que las tablas que no se fragmentan (Esquema production, Sales.SpecialOffer y Sales.SpecialOfferProduct) y que son usadas en algunas de las consultas se encuentran alojadas en la base AdventureWorks2019 para simplificar su consulta.

#### Configuración del servidor vinculado:

En esta sección configurar el servidor vinculado fue sencillo, esto debido a que solo se requirió de Agregarle un nombre al servidor, establecer de forma explicita el contexto de seguridad (usuario y contraseña) y finalmente probar la conexión,

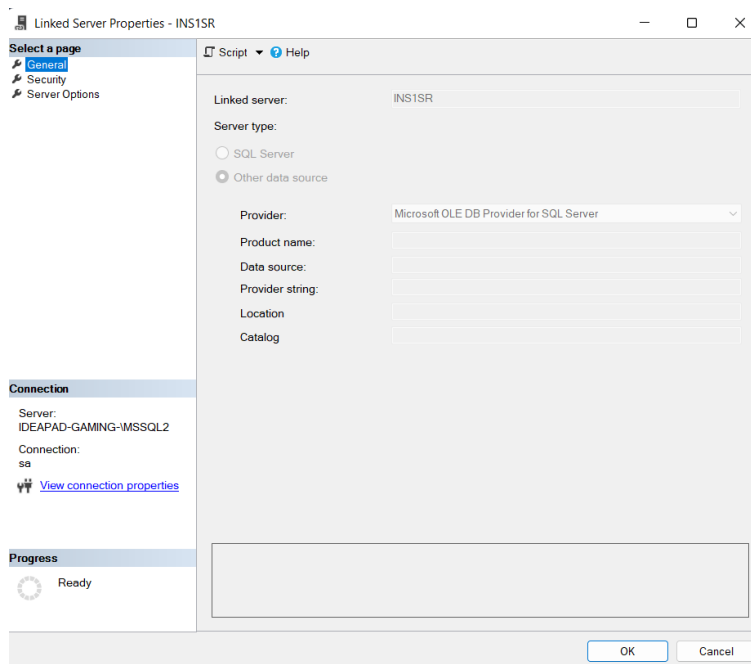


Figure 4 Propiedades generales del servidor vinculado

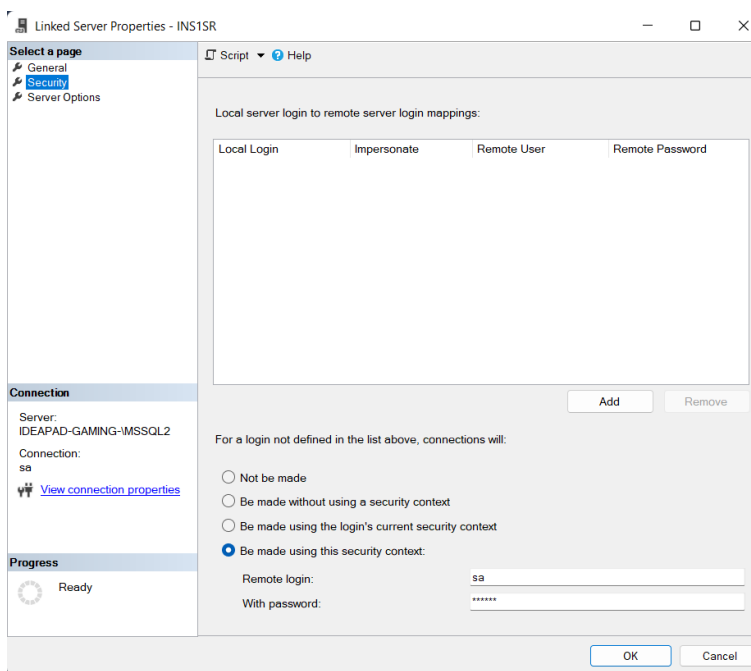


Figure 5 Propiedades de seguridad del servidor

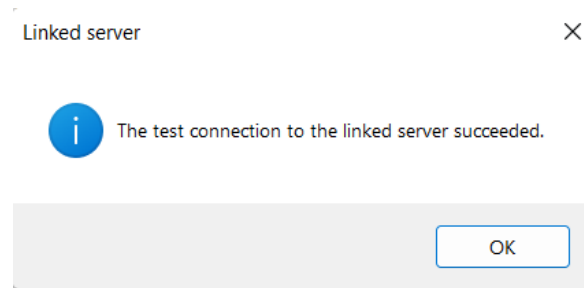


Figure 6 Conexión exitosa al servidor

Finalmente podemos observar que se puede acceder a los datos del servidor, y más especifico las tablas que se configuraron para la región de Asia Pacifico:

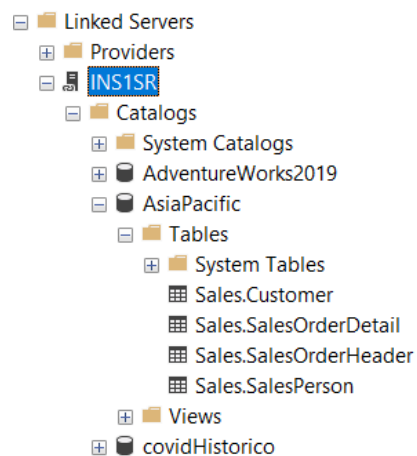


Figure 7 Tablas vistas a través del servidor vinculado

## Consultas:

### Consultas listadas:

1. La información de los clientes de almacenarse por región, considerando las regiones de acuerdo con el atributo group de SalesTerritory
2. Listar datos del empleado que atendió más ordenes por territorio
3. Listar los datos del cliente con más ordenes solicitadas en la región "North America"
4. Listar el producto más solicitado en la región "Europe"
5. Listar las ofertas que tienen más productos de la categoría "Bikes"
6. Listar los 3 productos menos solicitados en la región "Pacific"
7. Actualizar la subcategoría de los productos con ProductID del 1 al 4 a la subcategoría válida para el tipo de producto.
8. Listar los productos que no estén disponibles a la venta.
9. Listar los clientes del territorio 1 al 4 que no tengas asociado un valor en PersonID

10. Listar los clientes del territorio 1 que tengan ordenes en otro territorio

Soluciones propuestas

2:

```
select * from
(select top 1 SalesPersonID IMV, count(SalesOrderID) TV from Sales.SalesOrderHeader
where TerritoryID=1 and SalesPersonID>1 group by SalesPersonID order by TV desc) mv
inner join
(select * from Sales.SalesPerson) sp
on mv.IMV=sp.BusinessEntityID
```

(Esta solución se repite cambiando el numero del territorio en especifico que se desee recuperar)

3:

```
select * from
(select top 1 SalesPersonID IMV, count(SalesOrderID) TV from NorthAmerica.Sales.SalesOrderHeader
group by SalesPersonID order by TV desc) mv
inner join
(select * from NorthAmerica.Sales.SalesPerson) sp
on mv.IMV=sp.BusinessEntityID
```

4:

```
Select TOP 1 WITH TIES ProductID, SUM(SalesOrderDetail.OrderQty) as Ventas
from Europe.Sales.SalesOrderDetail Group by SalesOrderDetail.ProductID Order by Ventas desc
```

5:

```
Select * from Sales.SpecialOfferProduct as sop inner join
(Select ProductID from Production.Product as p inner join
(Select *from Production.ProductSubcategory where ProductSubcategory.ProductCategoryID=1)as Cat
on p.ProductSubcategoryID=Cat.ProductSubcategoryID) as psubid on sop.ProductID=psubid.ProductID
```

6:

```
Select TOP 3 WITH TIES SalesOrderDetail.ProductID, SUM(SalesOrderDetail.OrderQty) as Ventas
from AsiaPacific.Sales.SalesOrderDetail
Group by SalesOrderDetail.ProductID
Order by Ventas asc
```

7:

```

|create procedure sp_frag7
as
select * from Production.Product where ProductID between 1 and 4
|update Production.Product
set ProductSubcategoryID = 11
where ProductID=1
|update Production.Product
set ProductSubcategoryID = 17
where ProductID=2
|update Production.Product
set ProductSubcategoryID = 17
where ProductID=3
|update Production.Product
set ProductSubcategoryID = 11
where ProductID=4
go

```

8:

```

|select * from Production.Product where SellEndDate is not null

```

9:

```

Use NorthAmerica
SELECT * FROM NorthAmerica.sales.CUSTOMER
WHERE TerritoryID BETWEEN 1 AND 4 and TerritoryID!=2 and TerritoryID!=3 AND PersonID IS NULL

```

10:

No llegué a una solución.