



Universidad de Guayaquil

Facultad de Ciencias Matemáticas y Físicas

Carrera de Software

Materia: Inteligencia de Negocios

Manual del Proyecto del Primer Parcial

Grupo C:

-Duque Borbor Britany

-Peñafiel Del Rosario Gabriel

-Franco Emperador Daniel

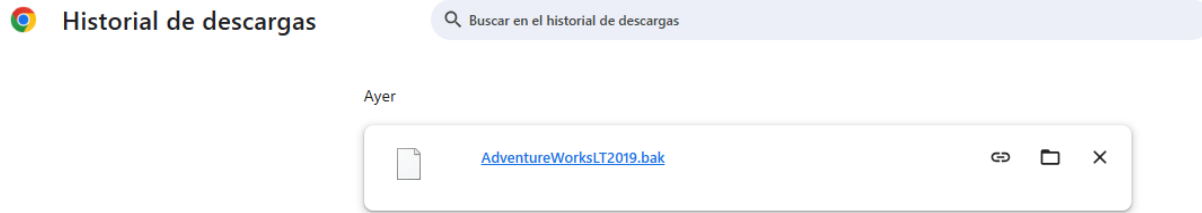
-Carbo Sandoval Kevin

-Rosario Delgado Bryan

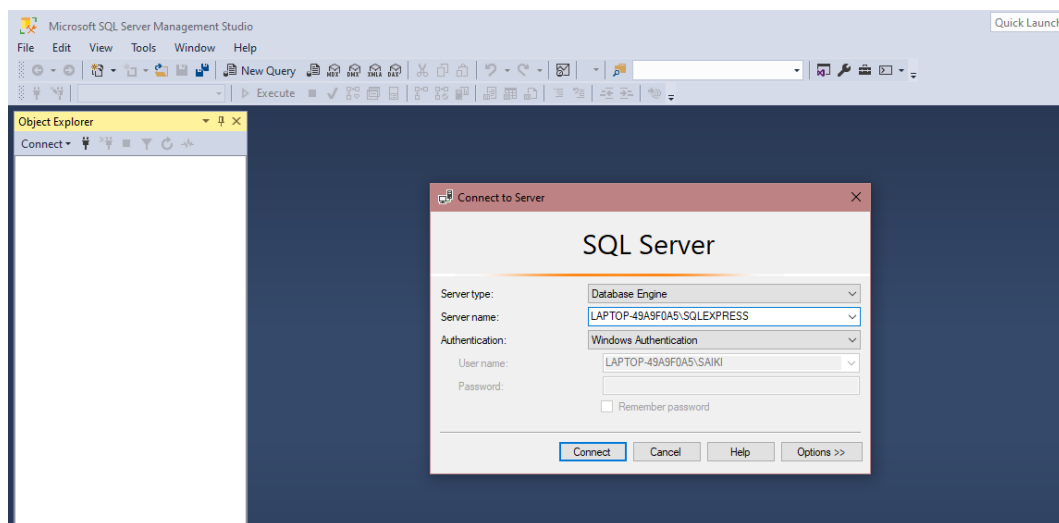
Curso: SOF-S-MA-7-2

Punto 1

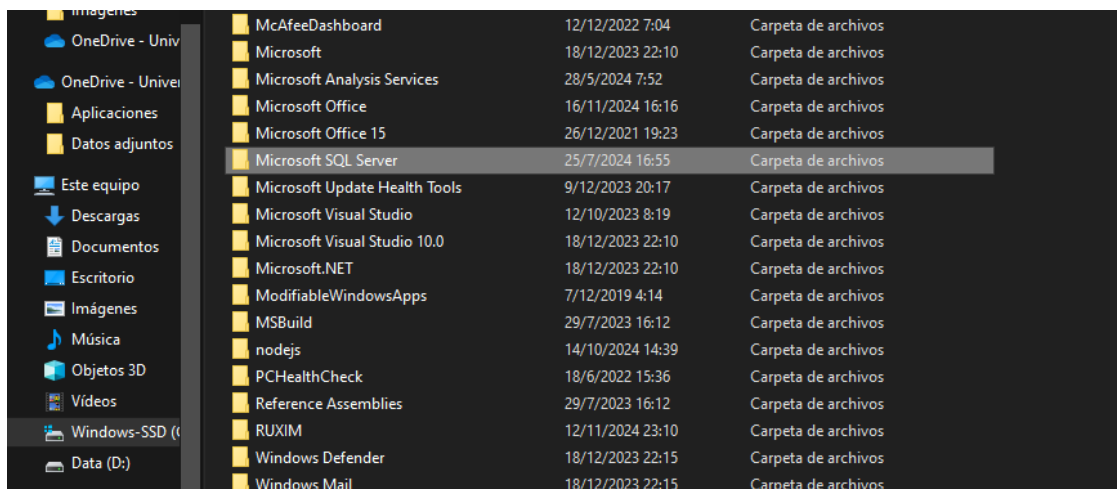
1.- Descargar la base de datos AdventureWorld 2019 con el link facilitado por el profesor.



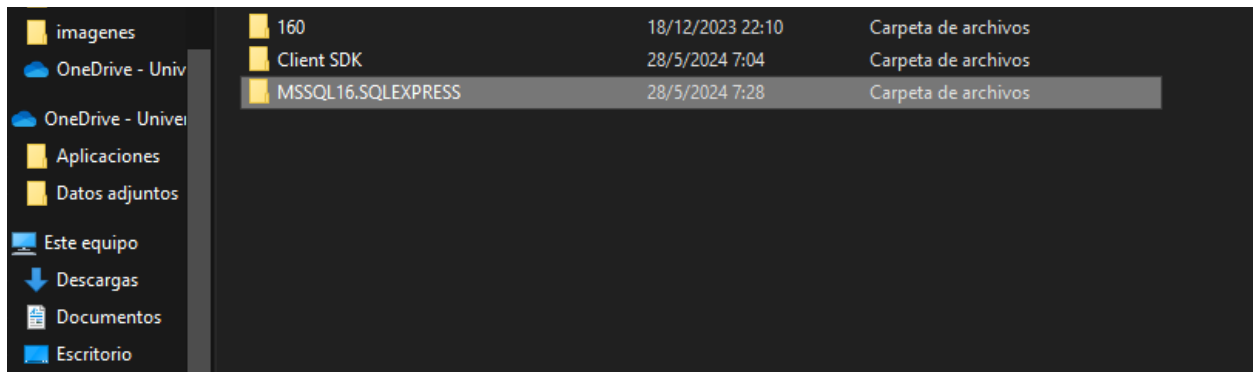
2.- Tener SQL server en el computador para instalar la base de datos descargada anteriormente.



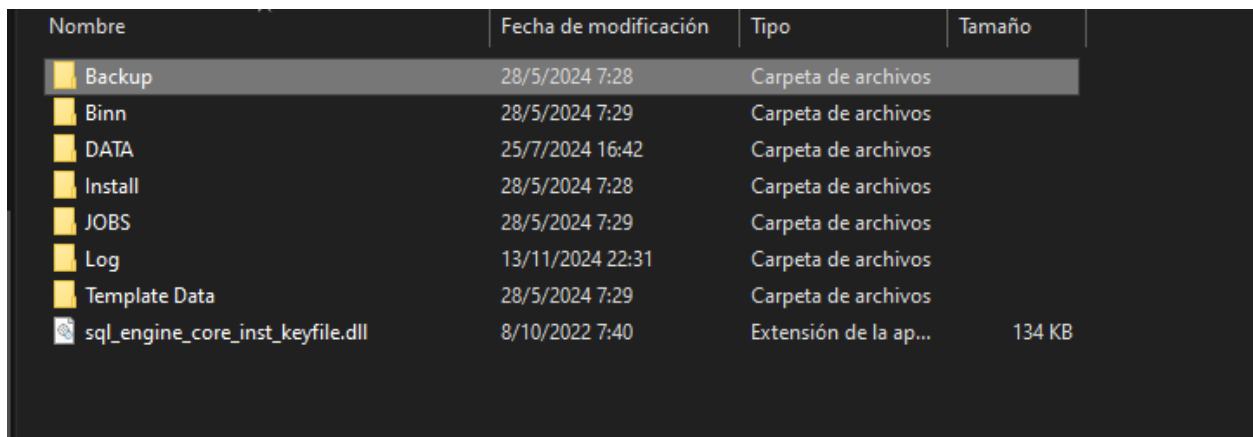
3.- Para instalar la base de datos en SQL server lo primero que hay que hacer es ir al explorador de archivos, luego entrar al disco c y buscar la carpeta archivos de programa, una vez dentro de esa carpeta buscar la carpeta “Microsoft SQL Server”.



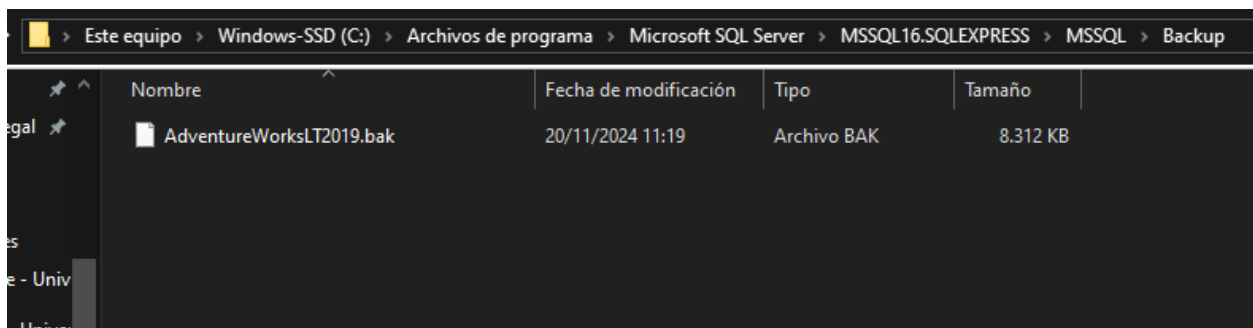
4.- El siguiente paso es entrar a la carpeta “Microsoft SQL Server” y buscar la siguiente carpeta.



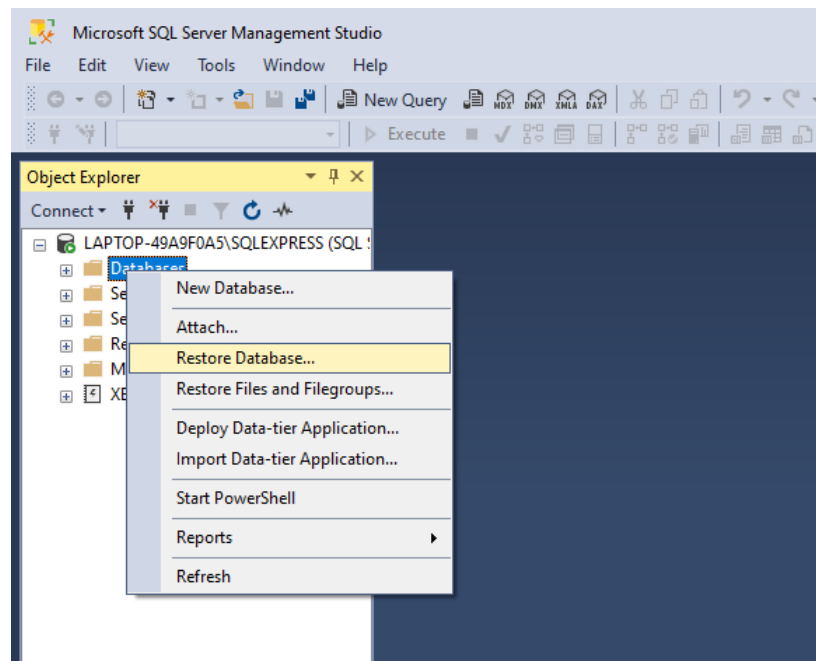
5.- Luego de entrar a la carpeta aparecerá una carpeta llamada “MSSQL” donde debemos entrar y encontrar la carpeta llamada “Backup”.



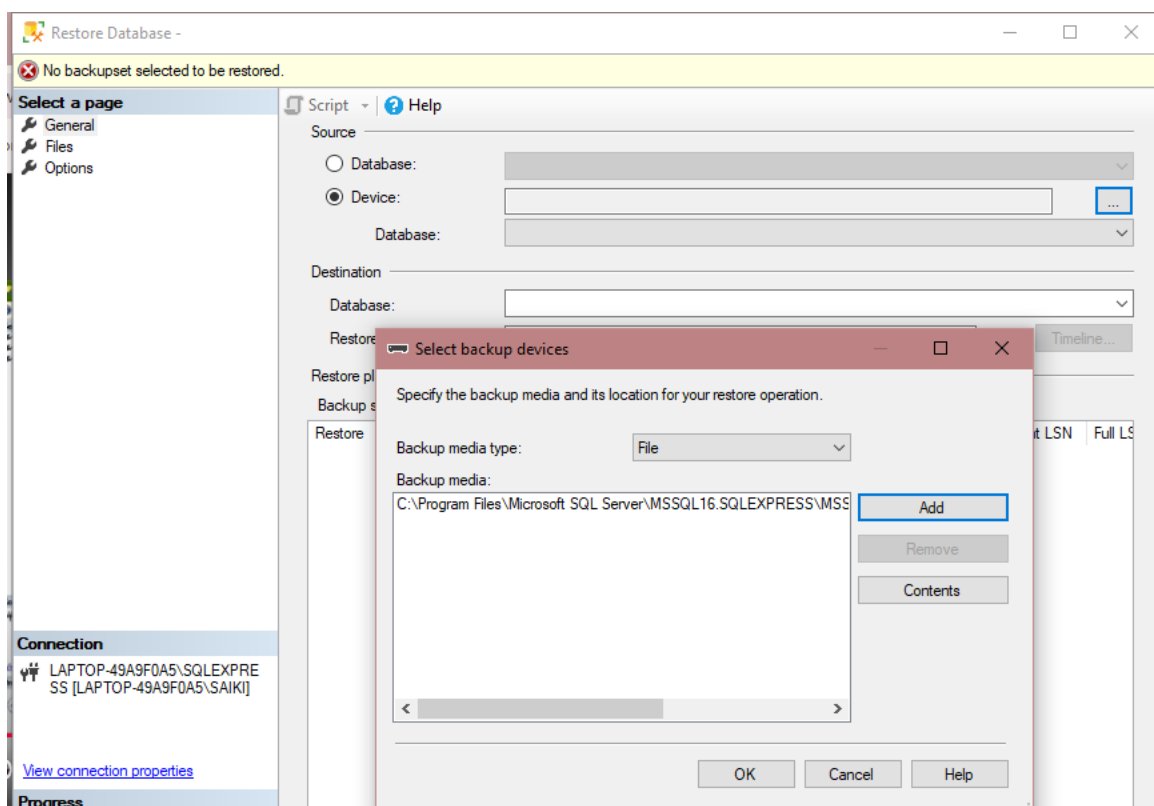
6.- Ahora debemos mover el archivo .bak que descargamos con la base de datos AdventureWorld 2019 a esa carpeta.



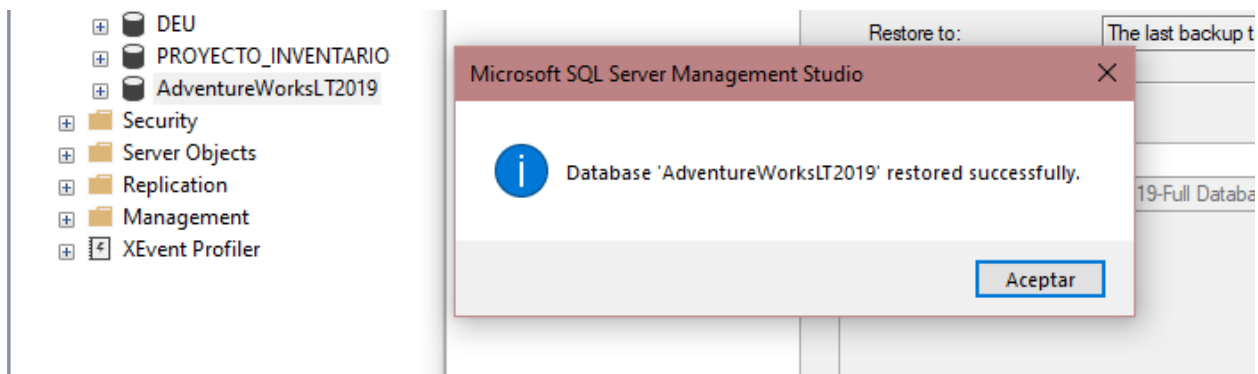
7.- Luego debemos abrir SQL server management studio, dar clic derecho en Databases y seleccionar restore database.



8.- Dentro de esa opción aparecerá la siguiente ventana donde en source debemos seleccionar device y al dar clic en los tres puntos de la derecha se abrirá otra ventana donde debemos elegir la opción add y luego de eso aparecerá el archivo .bak con la base de datos y la seleccionamos.

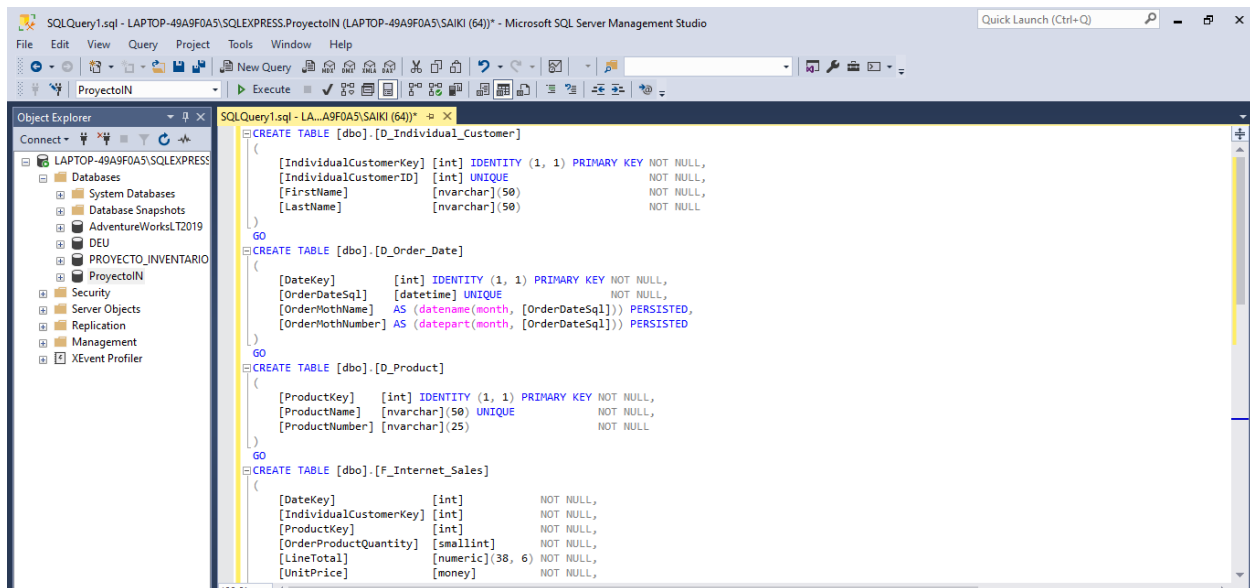


9.- Luego ponemos ok y con eso ya queda creada la base de datos en SQL Server.

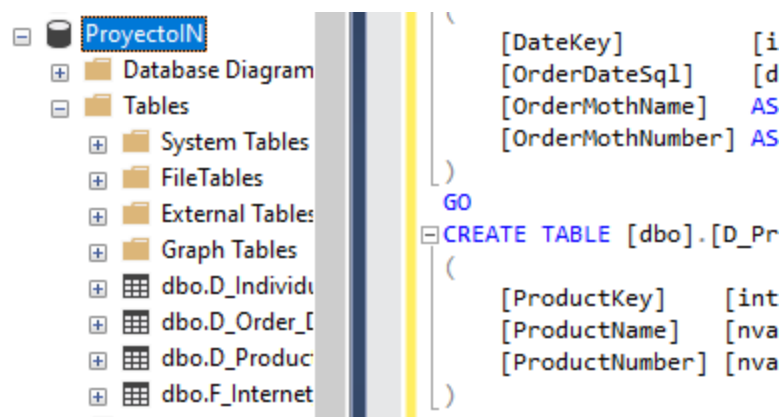


Punto 2

1.- Para crear la otra base de datos, se hará uso del siguiente script, que se creará dentro de otra base de datos llamada “ProyectoIN”

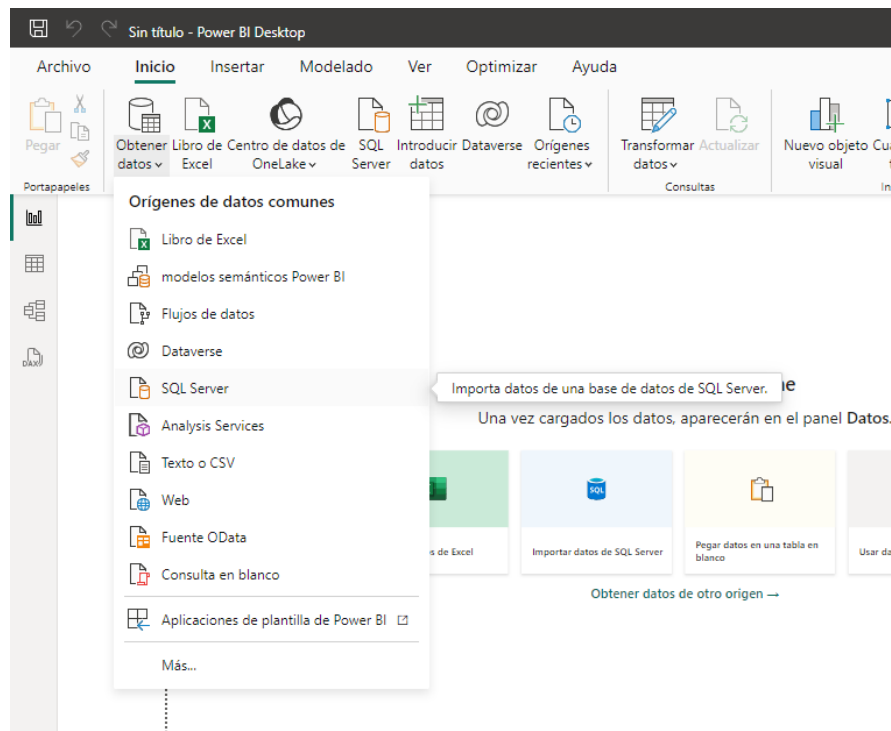


2.- Luego del uso del script ya quedarán todas las tablas creadas en la nueva base de datos

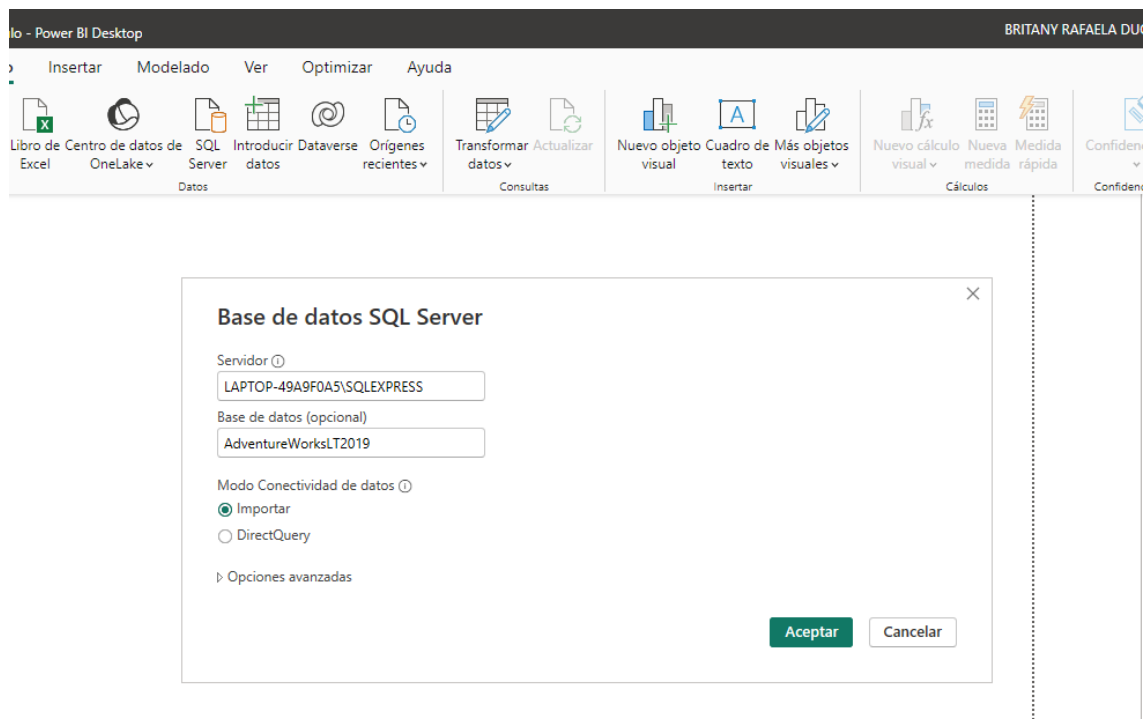


Punto 3

1.- Abrir Power BI y en obtener datos seleccionar SQL Server



2.- Luego aparecerá una ventana donde debemos llenar los datos de la base de datos que queremos importar.



3.- Luego debemos elegir las tablas que necesitamos para cargarlas en el otro modelo y elegir transformar datos.

Power BI Desktop - Sin título - Editor de Power Query

Archivo Inicio Transformar Agregar columna Vista Herramientas Ayuda

Consultas [4]

- SalesLT Customer
- SalesLT Product
- SalesLT SalesOrderDetail
- SalesLT ProductCategory

Origen([Schema="SalesLT",Item="ProductCategory"])[Data]

ProductCategoryID	ParentProductCategoryID	Name	rowguid
1	1	Bikes	CF8DA25C-DF71-47A7-B81B-64
2	2	Components	C657828D-D808-4ABA-91A3-AF
3	3	Clothing	10A7C342-CA82-48D4-8A38-46
4	4	Accessories	2BE38E36-D9A2-4EEF-B593-ED
5	5	Mountain Bikes	2D364ADE-264A-433C-B092-4F
6	6	Road Bikes	000310C0-BCC8-42C4-B0C3-45
7	7	Touring Bikes	02C5061D-ECDC-4274-B5F1-E9
8	8	Handlebars	3EF2C725-7135-4C85-9AE6-AE5
9	9	Bottom Brackets	A9E54089-8A1E-4CF5-8646-E38
10	10	Brakes	D43BA4A3-EF0D-426B-90EB-4B
11	11	Chains	E93A7231-F16C-4B0F-8C41-C73
12	12	Cranksets	4F644521-422B-4F19-974A-E3C
13	13	Dereailleurs	1830D70C-AA2A-40C0-A271-5B
14	14	Pedals	8F680A43-8C08-4F00-B2F5-E7

8 COLUMNAS, 41 FILAS Generación de perfiles de columnas basada en las 1000 primeras filas

VISTA PREVIA DESCARGADA A LAS 20:01

4.- Luego debemos ir eliminando columnas innecesarias para cada tabla

Power BI Desktop - Sin título - Editor de Power Query

Archivo Inicio Transformar Agregar columna Vista Herramientas Ayuda

Consultas [4]

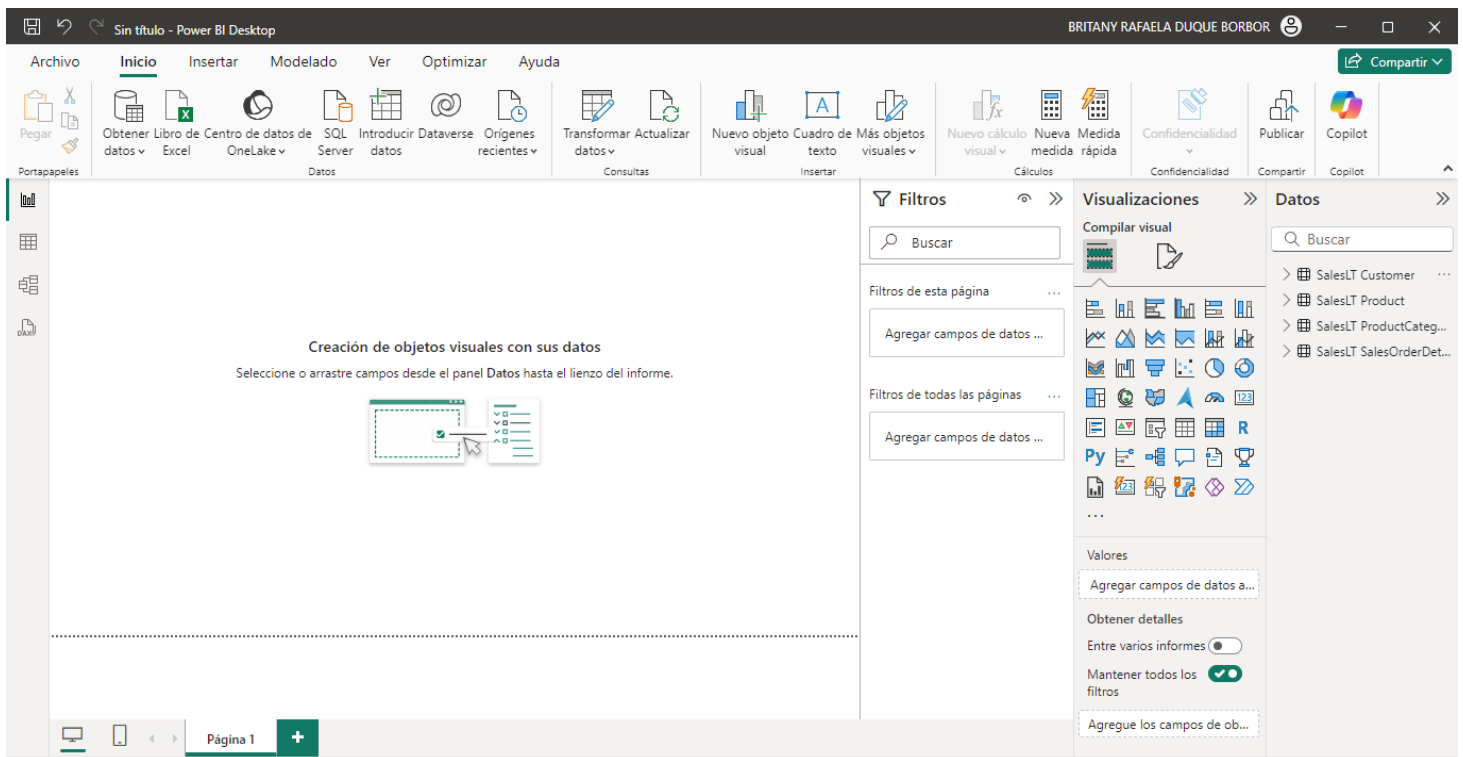
- SalesLT Customer
- SalesLT Product
- SalesLT SalesOrderDetail
- SalesLT ProductCategory

Table.RemoveColumns(SalesLT_Customer,{"NameStyle", "Title", "MiddleName", "Suffix", "CompanyName", "SalesPerson", ...})

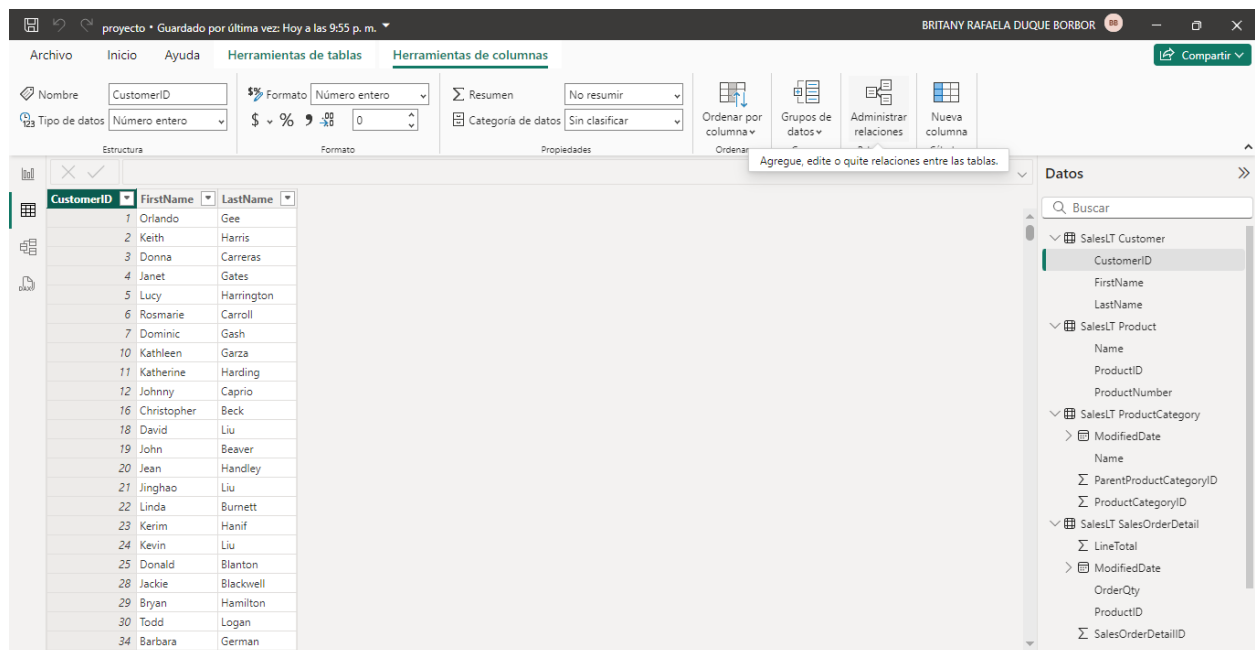
CustomerID	FirstName	LastName	rowguid
1	Orlando	Gee	
2	Keith	Harris	
3	Donna	Carreras	
4	Janet	Gates	
5	Lucy	Harrington	
6	Rosmarie	Carroll	
7	Dominic	Gash	
8	Kathleen	Garza	
9	Katherine	Harding	
10	Johnny	Caprio	
11	Christopher	Beck	
12	David	Liu	
13	John	Beaver	
14	Jean	Handley	
15	Jinghao	Liu	
16	Linda	Burnett	
17	Kerim	Hanif	
18	Kevin	Liu	
19	Donald	Blanton	
20	Jackie	Blackwell	
21	Bryan	Hamilton	
22	Todd	Logan	
23	Barbara	German	

1/7/2007 0:00:00 Table

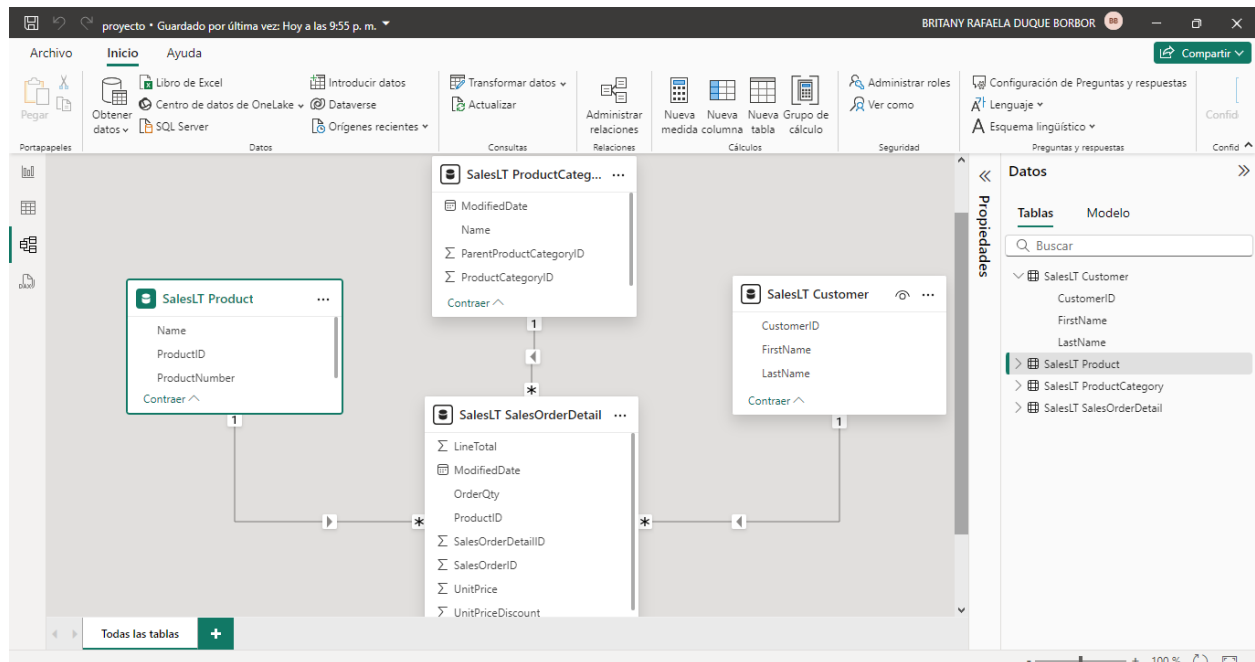
5.- Luego de haber guardado los cambios aparecerá lo siguiente



6.- Luego debemos modificar las relaciones entre tablas si es necesario.



7.- Luego en la vista del modelo podemos ver como queda el modelo estrella de nuestros datos.



Punto 4

Consultas SLICE:

1.- Se necesita saber las ventas totales realizadas por cada cliente, además de lo total gastado y el número de unidades compradas durante un periodo de 671 a 694.

```
SELECT
    IndividualCustomerKey,
    SUM(OrderProductQuantity) AS TotalQuantityBought,
    SUM(LineTotal) AS TotalSpent
FROM
    dbo.F_Internet_Sales
WHERE
    DateKey BETWEEN 671 AND 694
GROUP BY
    IndividualCustomerKey;
```

IndividualCustomerKey	TotalQuantityBought	TotalSpent
47	3	510000.000000
48	2	240000.000000
49	4	720000.000000
50	1	200000.000000
51	2	400000.000000
52	3	540000.000000
53	2	320000.000000
54	1	150000.000000

2.- Se solicita conocer el descuento promedio aplicado a cada producto durante el periodo de fechas comprendido entre DateKey = 634 y DateKey = 694. Este análisis debe incluir el descuento promedio por producto y estar ordenado de mayor a menor descuento aplicado.

```
SELECT
    ProductKey,
    AVG(DiscountUnitPrice) AS AverageDiscount
FROM
    dbo.F_Internet_Sales
WHERE
    DateKey BETWEEN 634 AND 694
GROUP BY
    ProductKey
ORDER BY
    AverageDiscount DESC;
```

ProductKey	AverageDiscount
201	0,00
202	0,00
203	0,00
204	0,00
205	0,00
206	0,00
207	0,00
208	0,00

Consultas DICE:

1.- Se requiere un análisis de las ventas para productos cuyo precio unitario sea mayor a 150,000, y que el número de unidades compradas por transacción sea mayor o igual a 2. Este análisis debe cubrir el rango de fechas DateKey = 634 a DateKey = 670, mostrando la cantidad total vendida y el monto total generado, agrupados por producto.

```
SELECT
    ProductKey,
    SUM(OrderProductQuantity) AS TotalQuantitySold,
    SUM(LineTotal) AS TotalSales
FROM
    dbo.F_Internet_Sales
WHERE
    DateKey BETWEEN 634 AND 670
    AND UnitPrice > 150000
    AND OrderProductQuantity >= 2
GROUP BY
    ProductKey
ORDER BY
    TotalSales DESC;
```

	ProductKey	TotalQuantitySold	TotalSales
1	211	4	800000.000000
2	230	4	720000.000000
3	233	3	540000.000000
4	209	3	540000.000000
5	237	3	510000.000000
6	228	3	480000.000000
7	232	2	400000.000000
-	---	-	-----

2.- Se requiere calcular el total de ventas realizadas por cada cliente para productos con un precio unitario mayor a 100,000, y donde el precio de descuento es igual a 0. El análisis debe cubrir el rango de fechas de DateKey = 634 a DateKey = 670.

```
SELECT
    IndividualCustomerKey,
    ProductKey,
    SUM(OrderProductQuantity) AS TotalQuantitySold,
    SUM(LineTotal) AS TotalSales
FROM
    dbo.F_Internet_Sales
WHERE
    DateKey BETWEEN 634 AND 670
    AND UnitPrice > 100000
    AND DiscountUnitPrice = 0
GROUP BY
    IndividualCustomerKey,
    ProductKey;
```

IndividualCustomerKey	ProductKey	TotalQuantitySold	TotalSales
11	202	1	300000.000000
12	203	3	450000.000000
13	204	2	400000.000000
14	205	1	120000.000000
15	206	4	600000.000000

Consultas PIVOT

1.- Se necesita calcular las ventas totales por producto y mostrar cómo se distribuyen en distintos rangos de precios (menos de 100,000, entre 100,000 y 500,000, y más de 500,000).

```
SELECT
    ProductKey,
    [LowRange] AS 'Menos de 100,000',
    [MidRange] AS 'De 100,000 a 500,000',
    [HighRange] AS 'Más de 500,000'
FROM (
    SELECT
        ProductKey,
        CASE
            WHEN UnitPrice < 100000 THEN 'LowRange'
            WHEN UnitPrice BETWEEN 100000 AND 500000 THEN 'MidRange'
            WHEN UnitPrice > 500000 THEN 'HighRange'
        END AS PriceRange,
        LineTotal
    FROM
        dbo.F_Internet_Sales
) AS SalesData
PIVOT (
    SUM(LineTotal)
    FOR PriceRange IN ([LowRange], [MidRange], [HighRange])
) AS PivotedProductSales
```

ProductKey	Menos de 100,000	De 100,000 a 500,000	Más de 500,000
201	NULL	200000.000000	NULL
202	NULL	300000.000000	NULL
203	NULL	450000.000000	NULL
204	NULL	400000.000000	NULL
205	NULL	120000.000000	NULL
206	NULL	600000.000000	NULL
207	NULL	250000.000000	NULL