

UNIVERSIDAD PRIVADA DE TACNA



INGENIERIA DE SISTEMAS

TITULO:

SESION DE LABORATORIO 05

CURSO:

INTELIGENCIA DE NEGOCIOS

DOCENTE:

Ing. Patrick Cuadros Quiroga

ESTUDIANTE:

Escalante Maron, Nelia

(2014049551)

Índice

1. Desarrollo	2
2. Conclusiones	30

1. Desarrollo

I. INFORMACION GENERAL

1. OBJETIVO GENERAL

Proveer soporte para actividades de toma de decision basado en informacion empirica El objetivo principal vendria a ser la creacion de una Data Warehouse esta data base nos va a permitir disminuir la data base original siendo mas facil y accesible para una creacion de inteligencia de negocios, tambien vamos a crear nuestro Data Source, Data Flow con sus diferentes Queries de cada Dimension

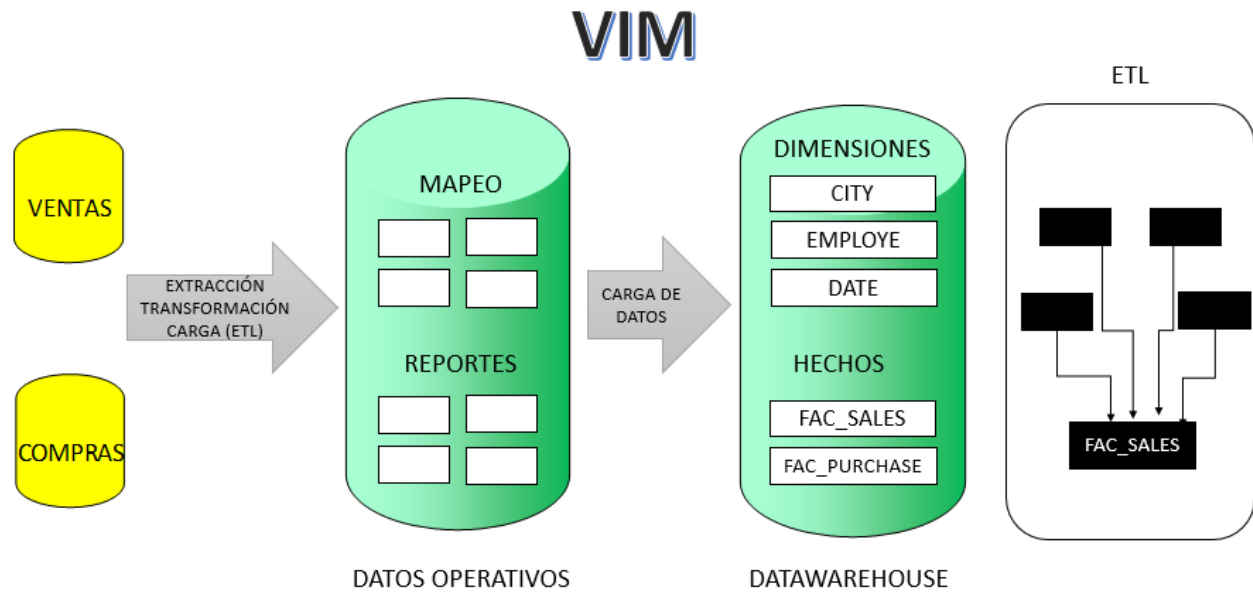
2. OBJETIVO ESPECIFICO

Desarrollar los siguientes puntos para este trabajo de la segunda unidad:

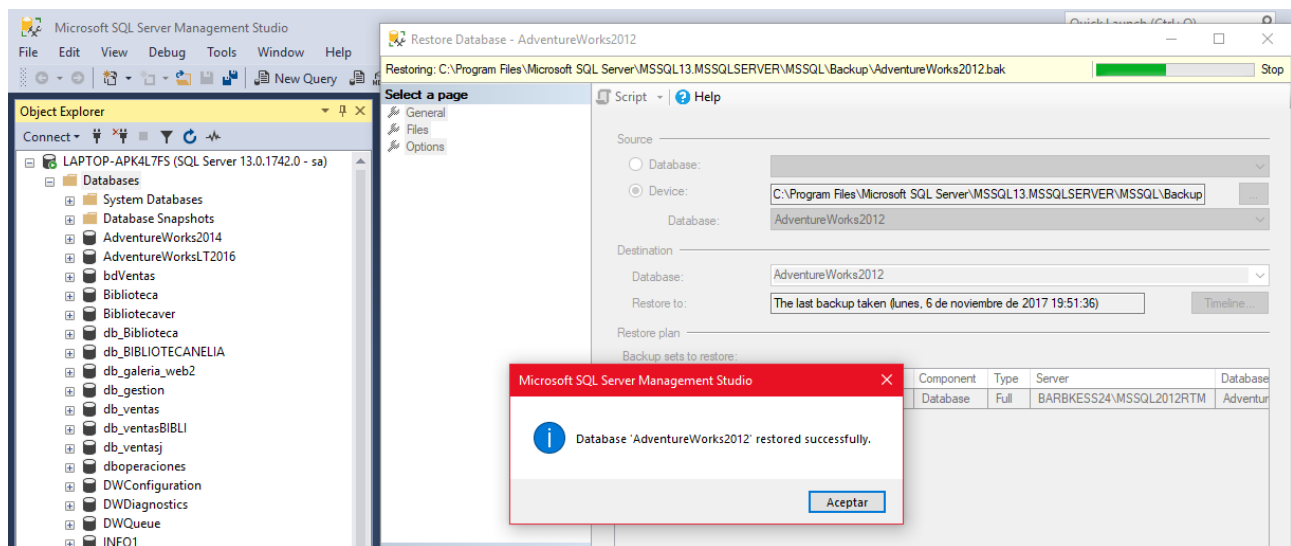
- Arquitectura de BI para su Empresa
- Diseño Dimensional Lógico y Fisico
- Detallar los Procesos de Negocios de su trabajo
- Requerimientos de Negocios
- Identificar las Dimensiones
- Identificar las Jerarquías
- Identificar los Hechos
- Identificar las Métricas
- Realizar el Modelado
- Crear el DW en un SGBD como SQL Server
- Desarrollo de ETL
- Análisis de los Datos
- Mapeo (Realizar un mapeo entre DW y Data Source)
- Conexiones a Base de Datos (Detallar las conexiones a datos)
- Desarrollo de DTSX (DataFlow, Queries)

II. DESARROLLO DEL PROYECTO(VIM)

1. ARQUITECTURA DE BI PARA SU EMPRESA



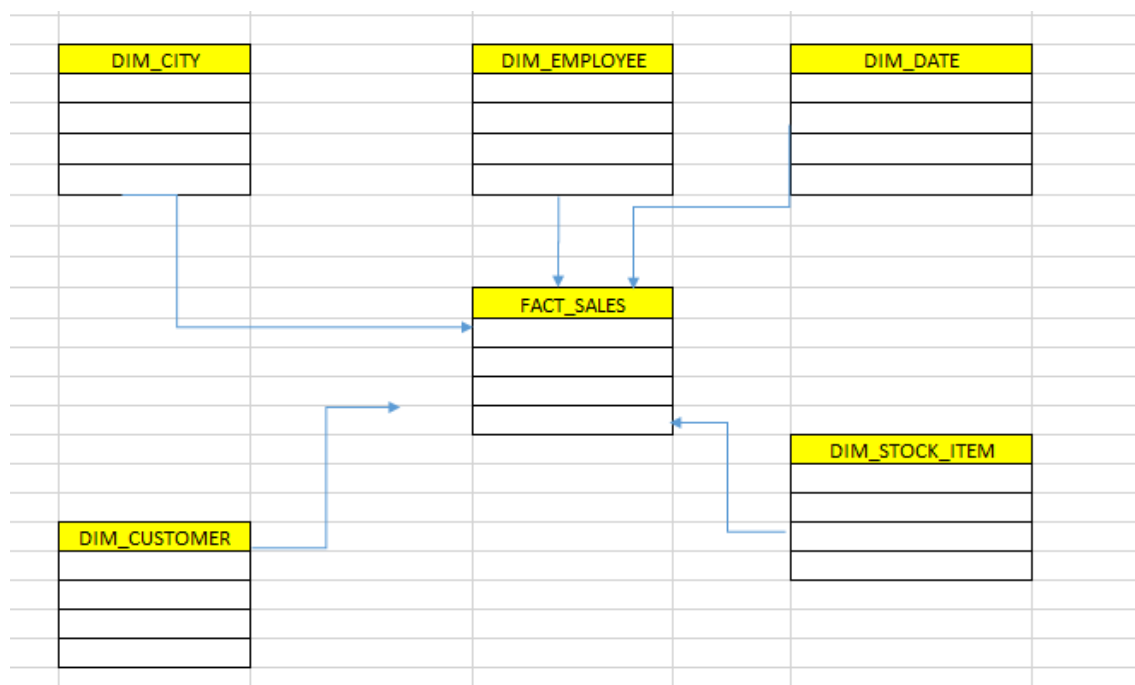
2. Restaurar Backup Workstationa2012



II. DESARROLLO DIMENSIONAL LOGICO Y FISICO - DISEÑO LOGICO - Proceso de Ventas

VENTAS						
	DIMENSIONES					
PROCESOS	Date	City	Customer	Employee	Stock Item	Supplier
Ventas(monto,cantidad)	X	X	X	X	X	
Compras(Monto,cantidad)						

Dimensiones	Tabla
DIM_EMPLOYEE	
DIM_CITY	Cities
DIM_CITY	State Provinces
DIM_DATE	
DIM_CUSTOMER	Customer Categories
DIM_CUSTOMER	Buying Groups
DIM_CUSTOMER	Special Deals
DIM_CUSTOMER	Stock Groups
DIM_CUSTOMER	Stock Item Stock Groups
DIM_STOCK_ITEM	Colors
Hechos	Tabla
FAC_SALES	Invoices
FAC_SALES	Invoice Lines
FAC_SALES	Delivery Methods
FAC_SALES	Orders
FAC_SALES	Orders Lines



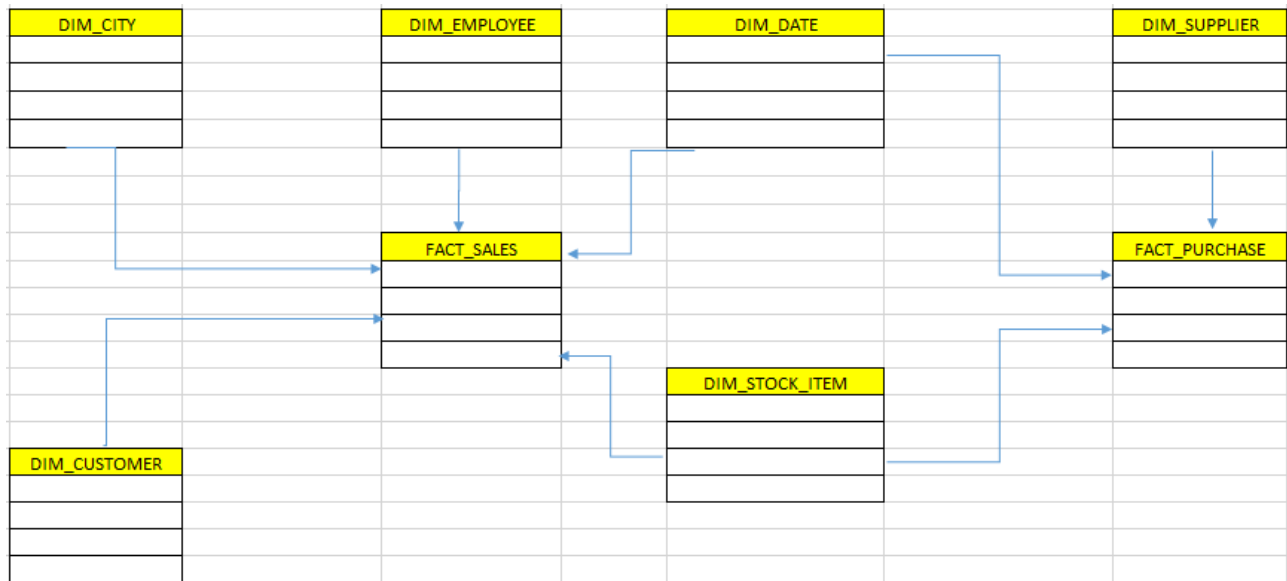
- Proceso de Ventas

COMPRAS						
	DIMENSIONES					
PROCESOS	Date	City	Customer	Employee	Stock Item	Supplier
Ventas(monto,cantidad)	X	X	X	X	X	
Compras(Monto,cantidad)	X				X	X

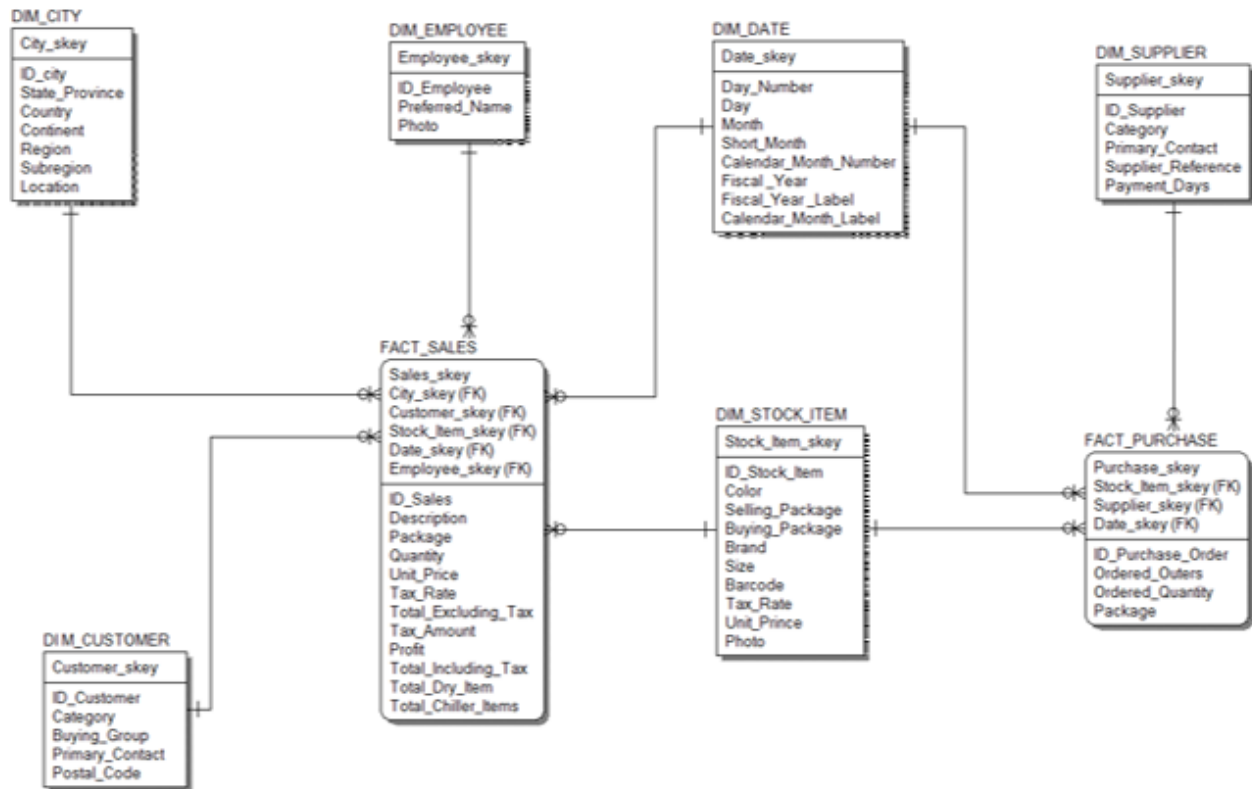
Dimensiones	Tabla
DIM_EMPLOYEE	Stock de Grupos
DIM_CITY	Stock de Articulos de Grupos
DIM_DATE	Stock de Articulos
DIM_CUSTOMER	Stock de Transacciones de Articulos
DIM_STOCK_ITEM	Colors
DIM_SUPPLIER	Suppliers
DIM_SUPPLIER	Supplier Categories

Hechos	Tabla
FAC_SALES	
FAC_PURCHASE	Purchase Orders

Hechos	Tabla
FAC_SALES	
FAC_PURCHASE	Purchase Orders



- Diseño Fisico



A. DETALLAR LOS PROCESOS DE NEGOCIOS DE SU TRABAJO

Lo que vamos a conocer en nuestro trabajo, del proyecto que vamos a presentar tenemos asignados a dos procesos, que va a ser ventas y compras, también tenemos en total 6 dimensiones, el primer proceso va a ser Ventas va a tener las siguientes dimensiones, Date, City, Customer, Employee, Stock Item, y el segundo proceso va a ser Compras va a tener las siguientes dimensiones, Date Stock Item, y Supplier.

DIMENSIONES						
PROCESOS	Date	City	Customer	Employee	Stock Item	Supplier
Ventas (Monto, Cantidad)	X	X	X	X	X	
Compras (Monto, Cantidad)	X				X	X

B. REQUERIMIENTOS DE NEGOCIOS

Para nuestra empresa VIM tenemos algunos requerimientos que queremos tener en cuenta.

REQUERIMIENTOS GENÉRICOS

- proveer un sistema intuitivo y fácil de usar que permita a los usuarios finales generar sus propios reportes y análisis.
- tener una sola versión de la información.
- proveer información de toda la compañía en un solo sistemas.
- que los usuarios puedan acceso la información desde cualquier lugar y en cualquier momento.

REQUERIMIENTOS FUNCIONALES

- El sistema deberá tener los mecanismos para controlar la seguridad de los datos por departamento, área, gerencia, así como una distribución organizacional jerárquica de la información.
- el sistema proveerá un mecanismo de notificaciones y alertas, con criterios y reglas configurables.
- el sistema permitirá la integración de diferentes fuentes de datos.
- el sistema debe ser intuitivo para que los usuarios finales puedan desarrollar sus propios reportes.

REQUERIMIENTOS DE NEGOCIO

- Definir la estrategia de negocio en procesos que sean medibles y controlables.
- Identificar la información necesaria para cada proceso de negocio, detallando su forma de análisis, seguimiento, técnicas de análisis y predicción (sí aplica), así como requerimientos tecnológicos para su cumplimiento.
- Priorizar los objetivos de la solución en función de su impacto en las metas del negocio.
- Diseñar una administración flexible que permita la adaptación de las estrategias del negocio a corto y mediano plazo basado en el aprendizaje continuo de la operación y seguimiento del negocio.

C. IDENTIFICAR LAS DIMENSIONES

Ahora vamos a identificar las dimensiones de nuestros dos procesos de esta forma en la cuál estamos viendo de Ventas y Compras, de las dos tenemos 6 en total 5 son para Ventas y 3 son para Compras, pero dos de las dimensiones de Ventas se unen a nuestro Hecho Compras

DIMENSIONES DE VENTAS Y COMPRAS	
DIM_CUSTOMER	
DIM_CITY	
DIM_EMPLOYEE	
DIM_DATE	
DIM_STOCK_ITEM	
DIM_SUPPLIER	

D. IDENTIFICAR LAS JERARQUIAS

Se crearon jerarquías para las dimensiones de:

- CUSTOMER (Cliente)

Esta tiene el siguiente esquema:

- Grupo de Compra
- Categoría
- Cliente

- CITY (Ciudad)

Esta tiene el siguiente esquema:

- Continente
- País
- Ciudad

- EMPLOYEE (Empleado)

Esta tiene el siguiente esquema:

- Grupo de Venta
- Categoría
- Employee

- DATE (Tiempo)

Esta tiene el siguiente esquema:

- Año
- Mes
- Día

- STOCK ITEM (Producto)

Esta tiene el siguiente esquema:

- Marca
- Producto
- Color

- SUPPLIER (Proveedor)

Esta tiene el siguiente esquema:

- Grupo de Compra
- Categoría
- Proveedor

E. IDENTIFICAR LOS HECHOS

Aquí tenemos nuestros Hechos con sus respectivas dimensiones dentro de cada Hecho, una tiene 5 que es FACTSALES este es el de ventas y el otro FACPURCHASE solo tiene dos una diferente y dos de la misma tabla de hechos de FACSLES que se unen a nuestro otro hecho

FACT_SALES
DIM_CUSTOMER
DIM_CITY
DIM_EMPLOYEE
DIM_DATE
DIM_STOCK_ITEM

FACT_PURCHASE
DIM_SUPPLIER
DIM_DATE
DIM_STOCK_ITEM

A. IDENTIFICAR LAS METRICAS

Las métricas que tiene nuestra empresa

COMPRAS						
	DIMENSIONES					
PROCESOS	Date	City	Customer	Employee	Stock Item	Supplier
Ventas(monto,cantidad)	X	X	X	X	X	
Compras(Monto,cantidad)	X				X	X

Métricas en ventas

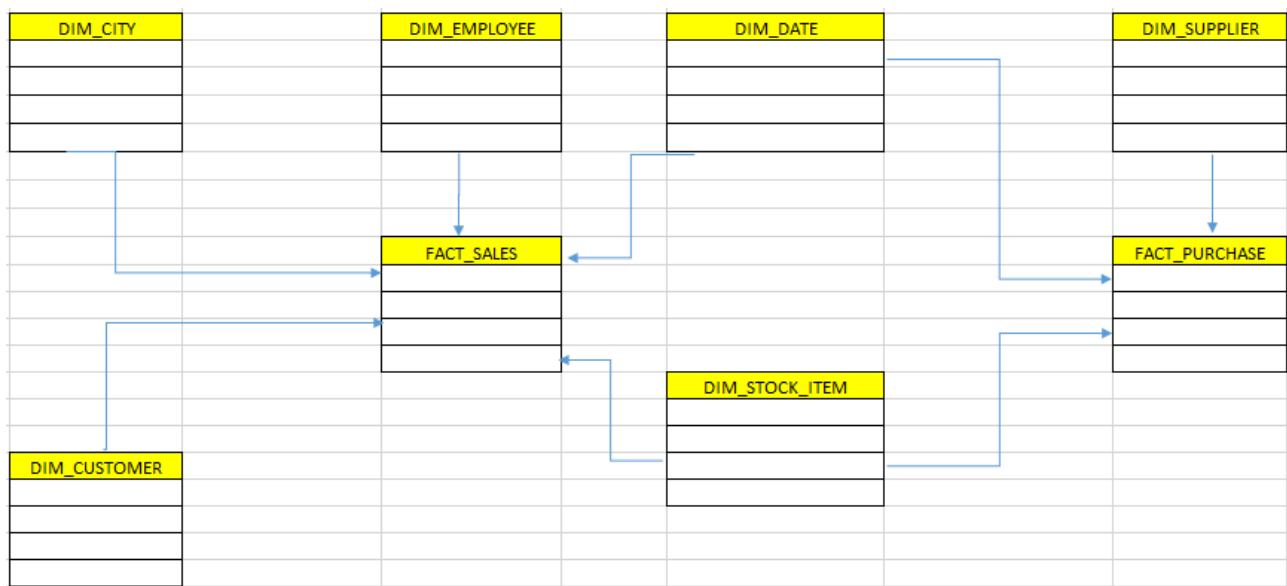
Monto	Cantidad
Monto de venta	Cantidad de stock
Monto de impuesto	Cantidad de venta
Monto perdidas	Cantidad de oferta

Métricas en compras

Monto	Cantidad
Monto de compras fallidas	Cantidad de productos comprados
Monto	Cantidad de

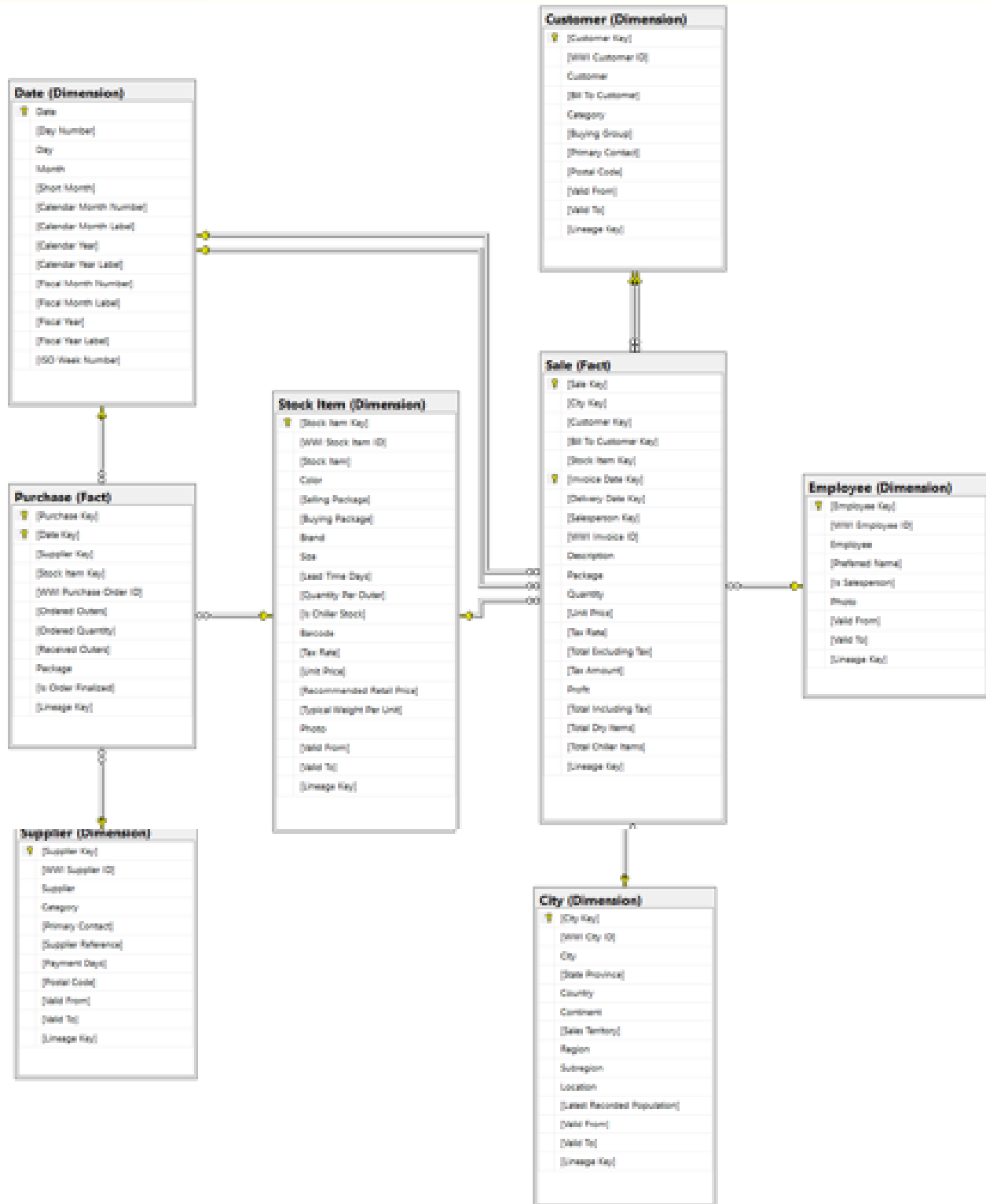
G. REALIZAR EL MODELADO

Para realizar el modelado de dimensiones y hecho primero se hizo un análisis de la base de datos transaccional luego pasamos seleccionar las tabas que vamos a utilizar para a realización de la data warehouse. Las tablas son las siguientes:



A. CREAR EL DW EN UN SGBD COMO SQL SERVER

Para la realización de nuestra data warehouse utilizamos el software de sql server.



3. DESARROLLO DE ETL

A. ANALISIS DE LOS DATOS (DETALLAR)

Para la realización de datos se hizo los siguientes query para cada dimension y hechos.

Query de customer

1

```
SELECT c.CustomerID, c.CustomerName as 'Customer', bt.CustomerName as  
'Bill to Customer', cc.CustomerCategoryName as 'Category',  
bg.BuyingGroupName as 'Buying Group', p.FullName as 'Primary Contact',  
c.DeliveryPostalCode as 'Postal Code',  
c.ValidFrom as 'Valid From', c.ValidTo as 'Valid to'  
FROM Sales.Customers c  
INNER JOIN Sales.CustomerCategories AS cc  
ON c.CustomerCategoryID = cc.CustomerCategoryID  
INNER JOIN Sales.Customers AS bt  
ON c.BillToCustomerID = bt.CustomerID  
INNER JOIN [Application].People AS p  
ON c.PrimaryContactPersonID = p.PersonID  
LEFT JOIN Sales.BuyingGroups AS bg  
ON c.BuyingGroupID = bg.BuyingGroupID  
ORDER BY CustomerID
```

Query Employee

```
SELECT p.PersonID as 'WWI Employee ID', p.FullName as 'Employee', p.PreferredName,  
p.IsSalesperson, p.Photo,  
p.ValidFrom, p.ValidTo  
FROM [Application].People p
```

Query city

```
SELECT c.CityID, c.CityName, sp.StateProvinceName, co.CountryName, co.Continent,  
sp.SalesTerritory, co.Region, co.Subregion,  
c.[Location], COALESCE(c.LatestRecordedPopulation, 0) as LatestRecordedPopulation,  
c.ValidFrom, c.ValidTo  
FROM [Application].Cities AS c  
INNER JOIN [Application].StateProvinces sp  
ON c.StateProvinceID = sp.StateProvinceID  
INNER JOIN [Application].Countries AS co  
ON sp.CountryID = co.CountryID
```

Query Stock Item

```
SELECT si.StockItemID,  
       si.StockItemName,  
       c.ColorName,  
       spt.PackageTypeName as 'Selling Package',  
       bpt.PackageTypeName as 'Buying Package',  
       si.Brand,  
       si.Size,  
       si.LeadTimeDays,  
       si.QuantityPerOuter,  
       si.IsChillerStock,  
       si.Barcode,  
       si.TaxRate,  
       si.UnitPrice,  
       si.RecommendedRetailPrice,  
       si.TypicalWeightPerUnit,  
       si.Photo,  
       si.ValidFrom,  
       si.ValidTo  
FROM Warehouse.StockItems AS si  
INNER JOIN Warehouse.PackageTypes AS spt  
ON si.UnitPackageID = spt.PackageTypeID  
INNER JOIN Warehouse.PackageTypes AS bpt  
ON si.OuterPackageID = bpt.PackageTypeID  
LEFT OUTER JOIN Warehouse.Colors AS c  
ON si.ColorID = c.ColorID
```

I

Query sale

```

SELECT
c.DeliveryCityID AS [City Key],
i.CustomerID AS [Customer Key],
i.BillToCustomerID AS [Bill To Customer Key],
    i.StockItemID AS [Stock Item Key],
CAST(i.InvoiceDate AS date) AS [Invoice Date Key],
CAST(i.ConfirmedDeliveryTime AS date) AS [Delivery Date Key],
    i.SalespersonPersonID AS [Salesperson Key],
i.InvoiceID AS [WWI Invoice ID],
i.[Description],
pt.PackageTypeName AS Package,
i.Quantity,
i.UnitPrice AS [Unit Price],
i.TaxRate AS [Tax Rate],
i.ExtendedPrice - i.TaxAmount AS [Total Excluding Tax],
i.TaxAmount AS [Tax Amount],
i.LineProfit AS Profit,
i.ExtendedPrice AS [Total Including Tax],
CASE WHEN si.IsChillerStock = 0 THEN i.Quantity ELSE 0 END AS [Total Dry Items],
CASE WHEN si.IsChillerStock <> 0 THEN i.Quantity ELSE 0 END AS [Total Chiller Items]
FROM Sales.Invoices AS i
INNER JOIN Sales.InvoiceLines AS il
ON i.InvoiceID = il.InvoiceID
INNER JOIN Warehouse.StockItems AS si
ON il.StockItemID = si.StockItemID
INNER JOIN Warehouse.PackageTypes AS pt
ON il.PackageTypeID = pt.PackageTypeID
INNER JOIN Sales.Customers AS c
ON i.CustomerID = c.CustomerID
INNER JOIN Sales.Customers AS bt

```

B. MAPEO (REALIZAR UN MAPEO ENTRE DW Y DATA SOURCE)

Comenzamos el mapeo que esta contiene cada datos que tiene una tabla par ello vamos a colocar cada tabla para ver los datos.

- FAC-PURCHASE

DATAWAREHOUSE						
Fact.Purchase						
Database	Table Schema	Table Name	Column Name	Type	Computed	Length
WideWorldImportersDW	Dimension	Purchase	Purchase Key	bigint	no	8
WideWorldImportersDW	Dimension	Date	Date Key	date	no	3
WideWorldImportersDW	Dimension	Supplier	Supplier Key	int	no	4
WideWorldImportersDW	Dimension	Item	Stock Item Key	int	no	4
WideWorldImporters	Purchasing	PurchaseOrders	PurchaseOrderID	int	no	4
WideWorldImporters	Purchasing	PurchaseOrderLines	OrderedOuters	int	no	4
WideWorldImporters				int	no	4
WideWorldImporters	Purchasing	PurchaseOrderLines	ReceivedOuters	int	no	4
WideWorldImporters				nvarchar	no	100
WideWorldImporters	Purchasing	PurchaseOrders	IsOrderFinalized	bit	no	1
WideWorldImporters				int	no	4

- DIM-stock-Item

DATAWAREHOUSE							
Dimension.[Stock Item]							
Database	Table Schema	Table Name	Column Name	Column_name	Type	Computed	Length
WideWorldImporters				Stock Item Key	int	no	4
WideWorldImporters	Warehouse	StockItems	StockItemID	WWI Stock Item ID	int	no	4
WideWorldImporters	Warehouse	StockItems	StockItemName	Stock Item	nvarchar	no	200
WideWorldImporters	Warehouse	Colors	ColorName	Color	nvarchar	no	40
WideWorldImporters				Selling Package	nvarchar	no	100
WideWorldImporters				Buying Package	nvarchar	no	100
WideWorldImporters	Warehouse	StockItems	Brand	Brand	nvarchar	no	100
WideWorldImporters	Warehouse	StockItems	Size	Size	nvarchar	no	40
WideWorldImporters	Warehouse	StockItems	LeadTimeDays	Lead Time Days	int	no	4
WideWorldImporters	Warehouse	StockItems	QuantityPerOuter	Quantity Per Outer	int	no	4
WideWorldImporters	Warehouse	StockItems	IsChillerStock	Is Chiller Stock	bit	no	1
WideWorldImporters	Warehouse	StockItems	Barcode	Barcode	nvarchar	no	100
WideWorldImporters	Warehouse	StockItems	TaxRate	Tax Rate	decimal	no	9
WideWorldImporters	Warehouse	StockItems	UnitPrice	Unit Price	decimal	no	9
WideWorldImporters	Warehouse	StockItems	RecommendedRetailPrice	Recommended Retail Price	decimal	no	9
WideWorldImporters	Warehouse	StockItems	TypicalWeightPerUnit	Typical Weight Per Unit	decimal	no	9
WideWorldImporters	Warehouse	StockItems	Photo	Photo	varbinary	no	-1
WideWorldImporters	Warehouse	StockItems	ValidFrom	Valid From	datetime2	no	8
WideWorldImporters	Warehouse	StockItems	ValidTo	Valid To	datetime2	no	8
				Lineage Key	int	no	4

- DIM-Supplier

DATAWAREHOUSE							
Dimension.Supplier							
Database	Table Schema	Table Name	Column Name	Column_name	Type	Computed	Length
WideWorldImporters				Supplier Key	int	no	4
WideWorldImporters	Purchasing	Suppliers	SupplierID	WWI Supplier ID	int	no	4
WideWorldImporters	Purchasing	Suppliers	SupplierName	Supplier	nvarchar	no	200
WideWorldImporters	Purchasing	SupplierCategories	SupplierCategoryName	Category	nvarchar	no	100
WideWorldImporters	Purchasing	Suppliers	PrimaryContactPersonID	Primary Contact	nvarchar	no	100
WideWorldImporters	Purchasing	Suppliers	SupplierReference	Supplier Reference	nvarchar	no	40
WideWorldImporters	Purchasing	Suppliers	PaymentDays	Payment Days	int	no	4
WideWorldImporters	Purchasing	Suppliers	PostalPostalCode	Postal Code	nvarchar	no	20
WideWorldImporters	Purchasing	Suppliers	ValidFrom	Valid From	datetime2	no	8
WideWorldImporters	Purchasing	Suppliers	ValidTo	Valid To	datetime2	no	8
WideWorldImporters				Lineage Key	int	no	4

- FAC-SALES

DATAWAREHOUSE								
FACT_SALES								
Database	Table Schema	Table Name	Column Name	Data Type	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	DATA_TYPE
WideWorldImportersDW	Fact	Sale	Sale Key	bigint				
WideWorldImportersDW	Fact	Sale	City Key	int	Dimension	City	City Key	int
WideWorldImportersDW	Fact	Sale	Customer Key	int	Dimension	Customer	Customer Key	int
WideWorldImportersDW	Fact	Sale	Bill To Customer Key	int	Dimension	Customer	Bill To Customer	nvarchar
WideWorldImportersDW	Fact	Sale	Stock Item Key	int	Dimension	Stock Item	Stock Item Key	int
WideWorldImportersDW	Fact	Sale	Invoice Date Key	date	Sales	Invoices	InvoiceDate	date
WideWorldImportersDW	Fact	Sale	Delivery Date Key	date	Sales	Invoices	ConfirmedDeliveryTime	datetime2
WideWorldImportersDW	Fact	Sale	Salesperson Key	int	Dimension	Employee	Employee Key	int
WideWorldImportersDW	Fact	Sale	WWI Invoice ID	int	Sales	InvoiceLines	InvoiceID	int
WideWorldImportersDW	Fact	Sale	Description	nvarchar	Dimension	Stock Item	Stock Item	nvarchar
WideWorldImportersDW	Fact	Sale	Package	nvarchar	Dimension	Stock Item	Selling Package	nvarchar
WideWorldImportersDW	Fact	Sale	Quantity	int	Sales	InvoiceLines	Quantity	int
WideWorldImportersDW	Fact	Sale	Unit Price	decimal	Dimension	Stock Item	Unit Price	decimal
WideWorldImportersDW	Fact	Sale	Tax Rate	decimal	Dimension	Stock Item	Tax Rate	decimal
WideWorldImportersDW	Fact	Sale	Total Excluding Tax	decimal				
WideWorldImportersDW	Fact	Sale	Tax Amount	decimal	Sales	InvoiceLines	TaxAmount	decimal
WideWorldImportersDW	Fact	Sale	Profit	decimal	Sales	InvoiceLines	LineProfit	decimal
WideWorldImportersDW	Fact	Sale	Total Including Tax	decimal	Sales	InvoiceLines	ExtendedPrice	decimal

- DIM-CITY

DATAWAREHOUSE								
DIM_city								
Database	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	DATA_TYPE	Table Schema	Table Name	Column Name	Data Type
VIM BD DW	Application	Cities	CityID	int	Dimension	City	City Key	int
VIM BD DW	Application	Cities	CityName	nvarchar	Dimension	City	WWI City ID	int
VIM BD DW	Application	StateProvinces	StateProvinceName	nvarchar	Dimension	City	State Province	nvarchar
VIM BD DW	Application	Countries	CountryName	nvarchar	Dimension	City	Country	nvarchar
VIM BD DW	Dimension	City	Continent	nvarchar	Dimension	City	Continent	nvarchar
VIM BD DW	Application	StateProvinces	SalesTerritory	nvarchar	Dimension	City	Sales Territory	nvarchar
VIM BD DW	Application	Countries Archive	Region	nvarchar	Dimension	City	Region	nvarchar
VIM BD DW	Application	Countries Archive	Subregion	nvarchar	Dimension	City	Subregion	nvarchar
VIM BD DW	Application	Cities	Location	geography	Dimension	City	Location	geography
VIM BD DW	Application	Cities	LatestRecordedPopulation	bigint	Dimension	City	Latest Recorded Population	bigint
VIM BD DW	Application	Cities	ValidFrom	datetime2	Dimension	City	Valid From	datetime2
VIM BD DW	Application	Cities	ValidTo	datetime2	Dimension	City	Valid To	datetime2

- DIM-EMPLOYEE

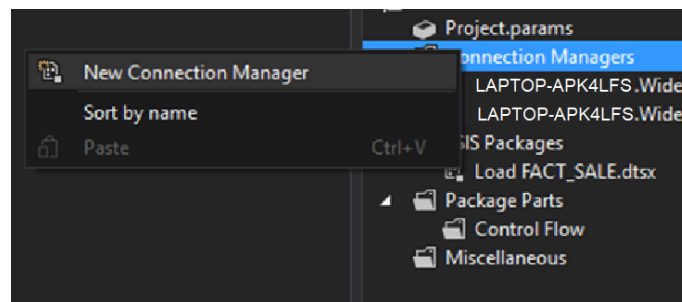
DATAWAREHOUSE								
DIM_EMPLOYEE								
Database DW	Table Schema	Table Name	Column Name	Data Type	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	DATA_TYPE
VIM BD DW	Dimension	Employee	Employee Key	int				
VIM BD DW	Dimension	Employee	WWI Employee ID	int	Application	People	PersonID	int
VIM BD DW	Dimension	Employee	Employee	nvarchar		People	FullName	nvarchar
VIM BD DW	Dimension	Employee	Preferred Name	nvarchar	Application	People	PreferredName	nvarchar
VIM BD DW	Dimension	Employee	Is Salesperson	bit	Application	People	IsSalesperson	bit
VIM BD DW	Dimension	Employee	Photo	varbinary	Application	People	Photo	varbinary
VIM BD DW	Dimension	Employee	Valid From	datetime2	Application	People	ValidFrom	datetime2
VIM BD DW	Dimension	Employee	Valid To	datetime2	Application	People	ValidTo	datetime2

- DIM-CUSTOMER

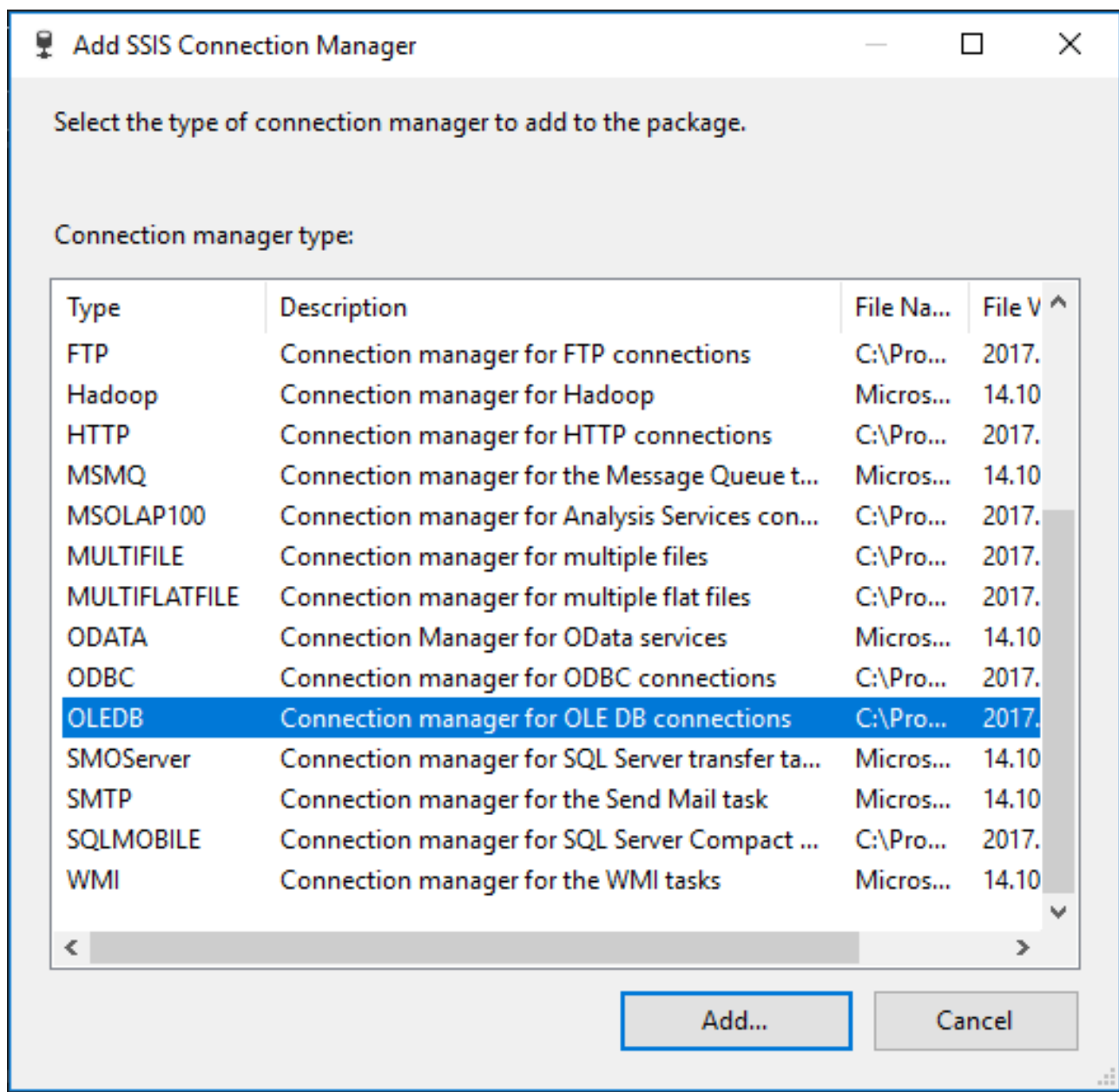
DATAWAREHOUSE								
DIM_CUSTOMER								
Database DW	Table Schema	Table Name	Column Name	Data Type	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	DATA_TYPE
WideWorldImporters	Dimension	Customer	Customer Key	int				
WideWorldImporters	Dimension	Customer	WWI Customer ID	int	Sales	Customers	CustomerID	int
WideWorldImporters	Dimension	Customer	Customer	nvarchar	Sales	Customers	CustomerName	nvarchar
WideWorldImporters	Dimension	Customer	Bill To Customer	nvarchar	Sales	Customers	BillToCustomerID	int
WideWorldImporters	Dimension	Customer	Category	nvarchar	Website	Customers	CustomerCategoryName	nvarchar
WideWorldImporters	Dimension	Customer	Buying Group	nvarchar	Website	Customers	BuyingGroupName	nvarchar
WideWorldImporters	Dimension	Customer	Primary Contact	nvarchar	Website	Customers	PrimaryContact	nvarchar
WideWorldImporters	Dimension	Customer	Postal Code	nvarchar	Sales	Customers	PostalPostalCode	nvarchar
WideWorldImporters	Dimension	Customer	Valid From	datetime2	Sales	Customers	ValidFrom	datetime2
WideWorldImporters	Dimension	Customer	Valid To	datetime2	Sales	Customers	ValidTo	datetime2
WideWorldImporters	Dimension	Customer	Lisense Key	int				

C. CONECCIONES A BASE DE DATOS (DETALLAR LAS CONECCIONES A DATOS)

Ahora vamos a crear la conexión de nuestra base de datos en nuestro programa de Data Flow, ponemos clic derecho y luego nueva conexión



Ahora nos pide conectar el tipo de conexión que vamos a tener, y nosotros trabajamos con OLEDB y luego le damos clic en ADD agregar



Luego vamos a poner el nombre de nuestra máquina, que será LAPTOP-APK4LFS, y lo ponemos

Connection Manager

Provider: Native OLE DB\SQL Server Native Client 11.0

Connection

Server name: LAPTOP-APK4L7F5 Refresh

Log on to the server

☒ Use Windows Authentication

☐ Use SQL Server Authentication

User name:

Password:

☐ Save my password

Connect to a database

☒ Select or enter a database name: WideWorldImporters

☐ Attach a database file: Browse...

Logical name:

Test Connection OK Cancel Help

Luego escogemos nuestra base de datos transaccional y le damos clic en OK

Connect to a database

☒ Select or enter a database name:

WideWorldImporters

☐ Attach a database file:

DWConfiguration

DWDiagnostics

DWQueue

master

model

msdb

ReportServer

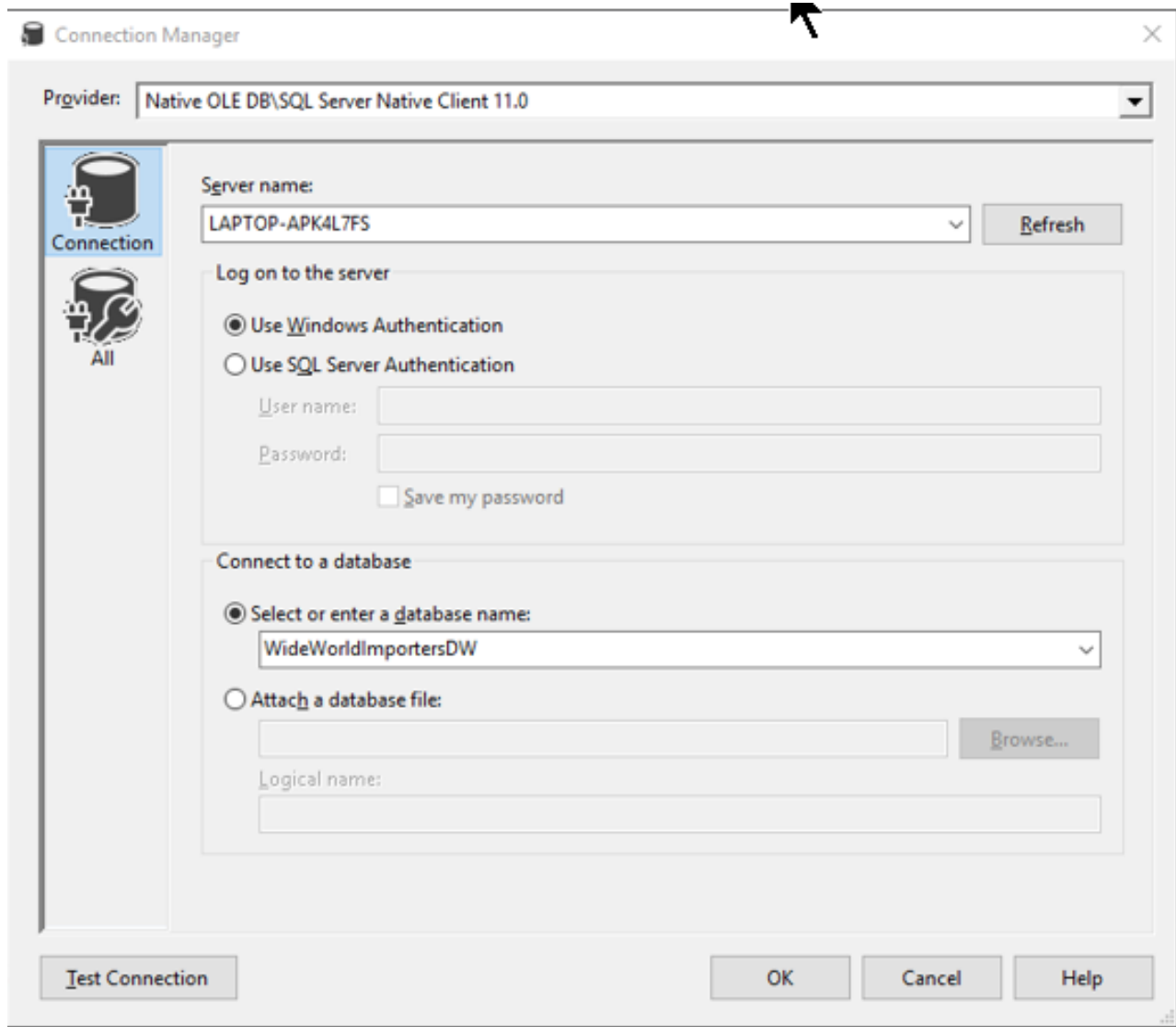
ReportServerTempDB

tempdb

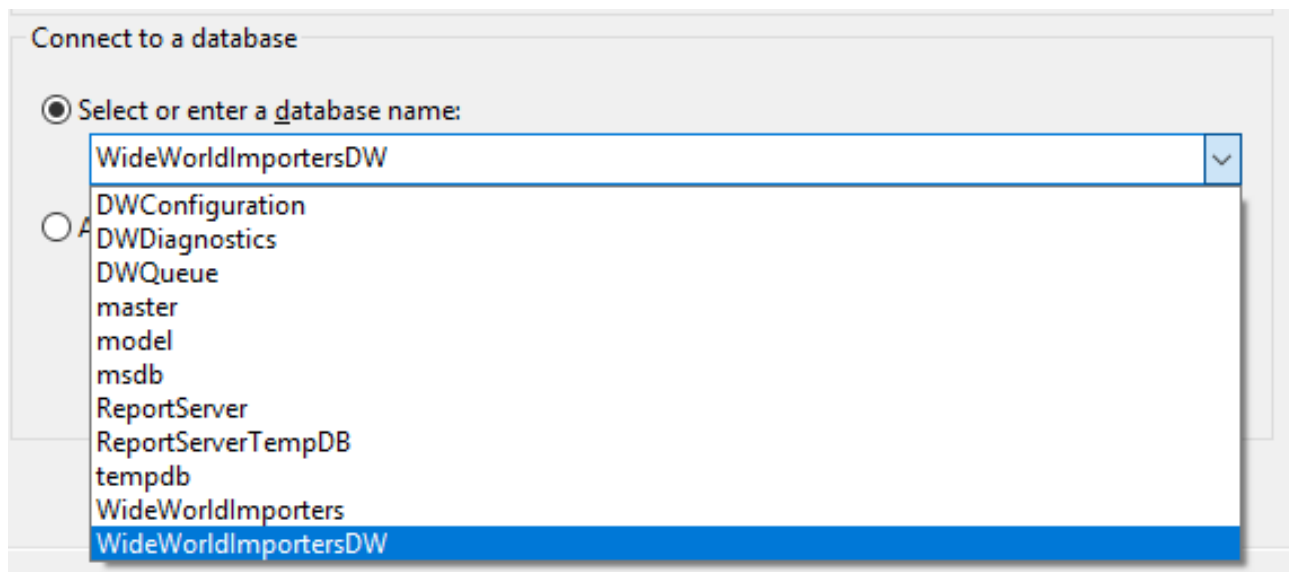
WideWorldImporters

WideWorldImportersDW

Luego vamos a crear otra nueva conexión, y luego ponemos el nombre de equipo, y esto es para la otra base de datos qué es el Datawarehouse

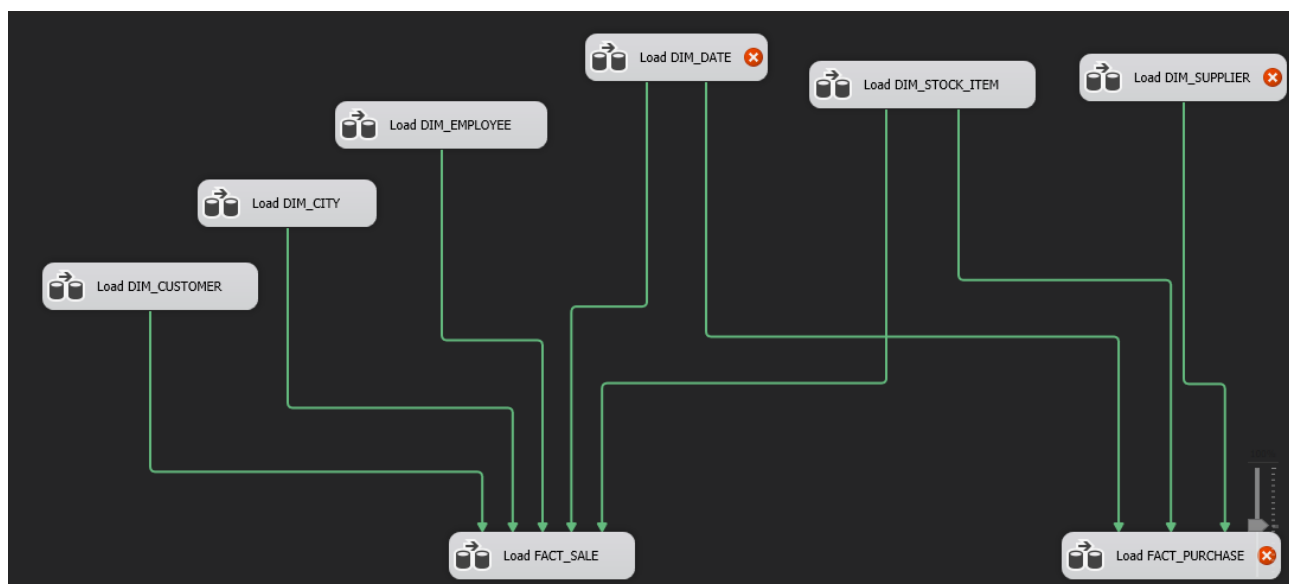


Luego escogemos la base de datos creada de nuestro DataWarehouse

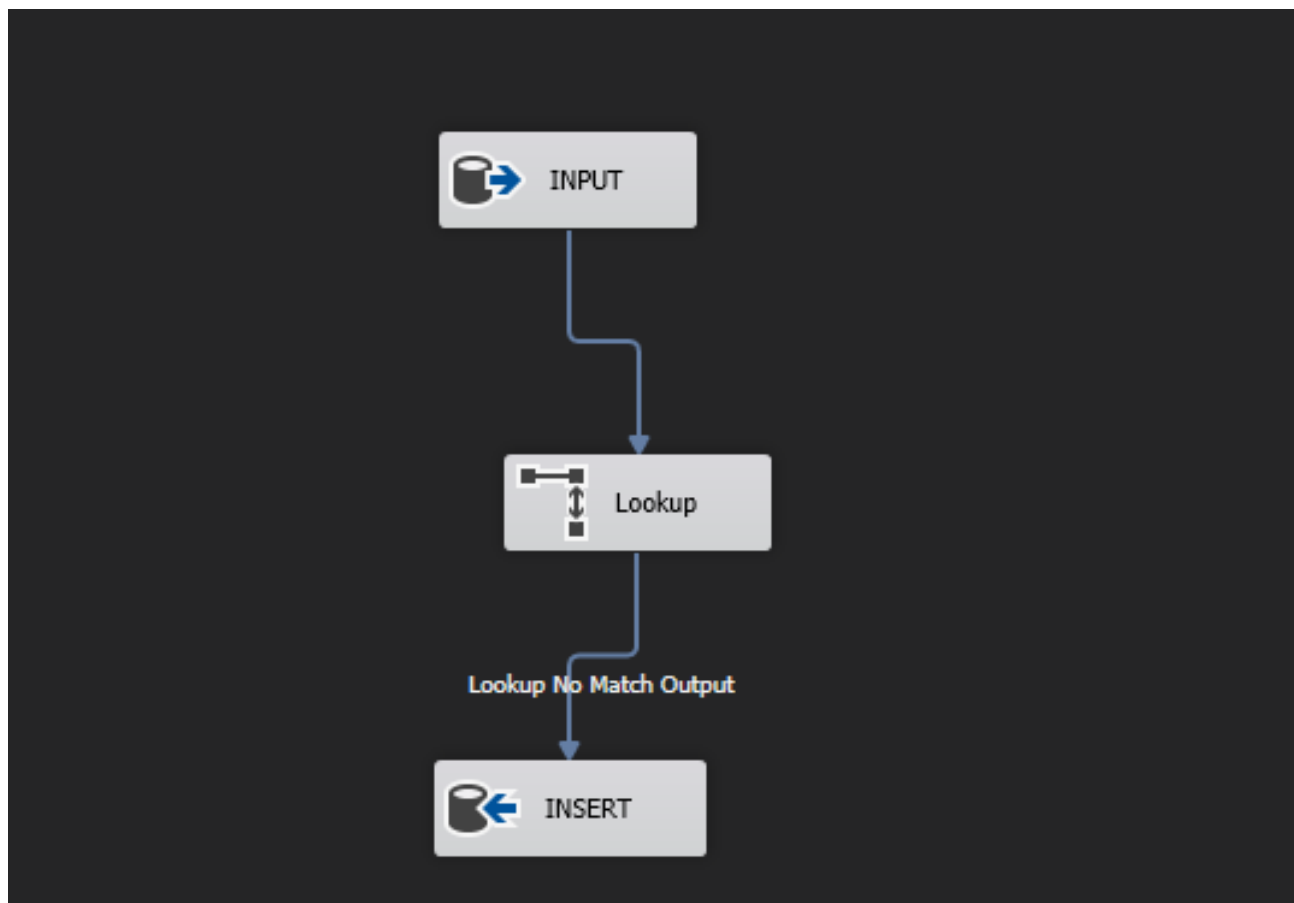


D. DESARROLLO DE DTSX (DATAFLOW, QUERIES)

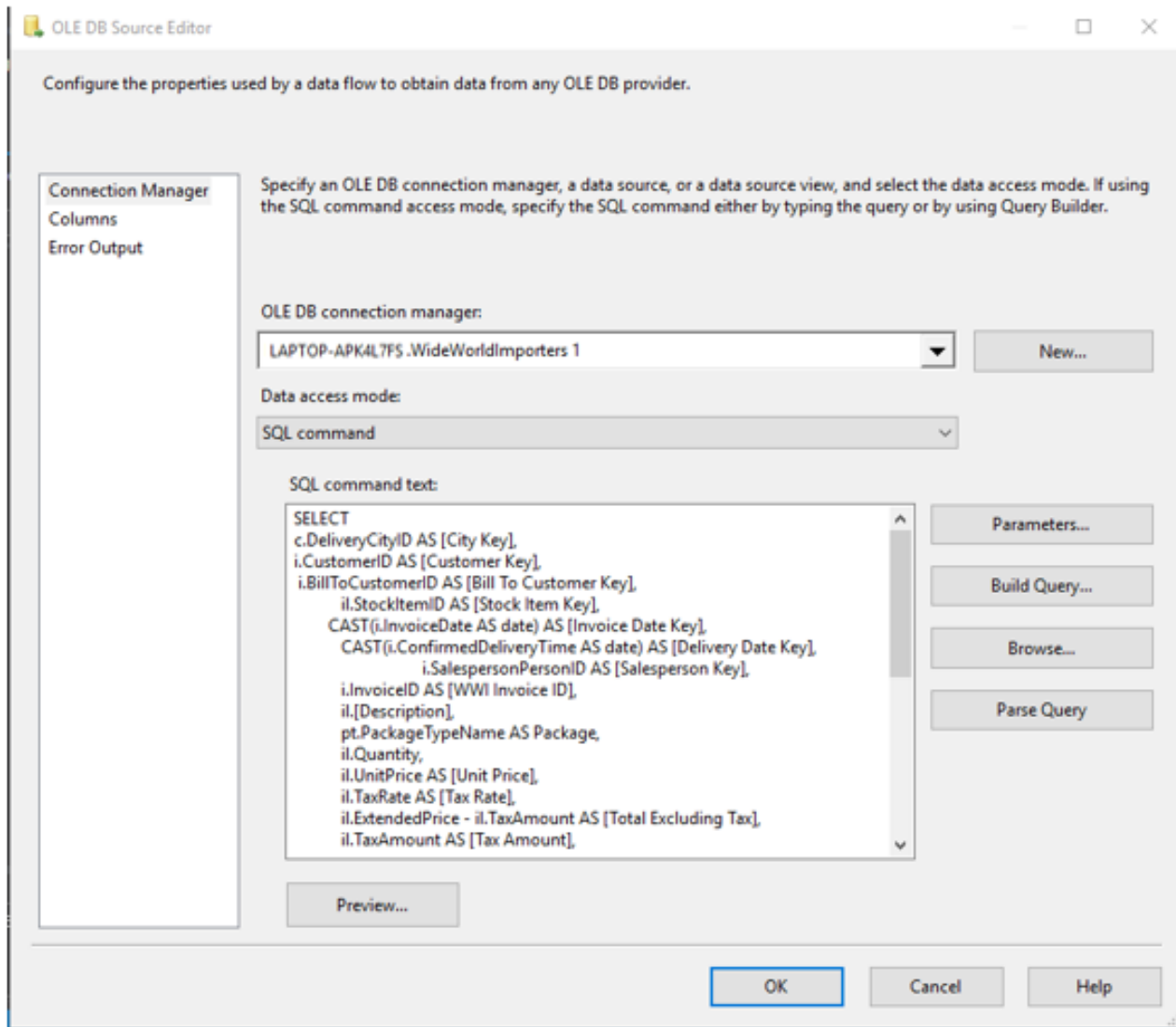
Esta es la creación de nuestro Control Flow, en si con sus dimensiones respectivas para cada hecho, que tenemos que es Fact-Sale, Fact-Purchase



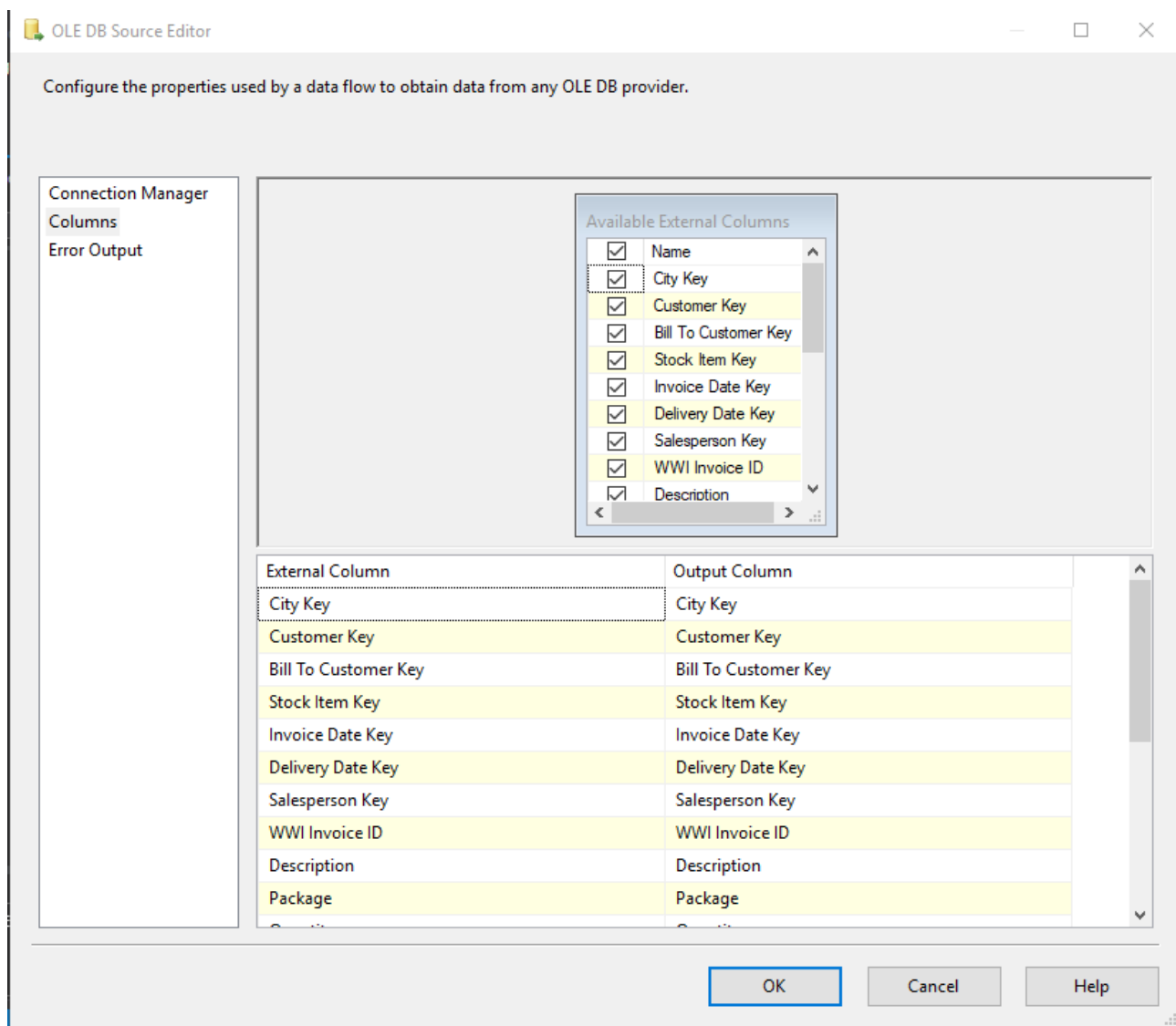
Ahora vamos a crear el Data Flow para cada uno de nuestras dimensiones y luego también para los hechos



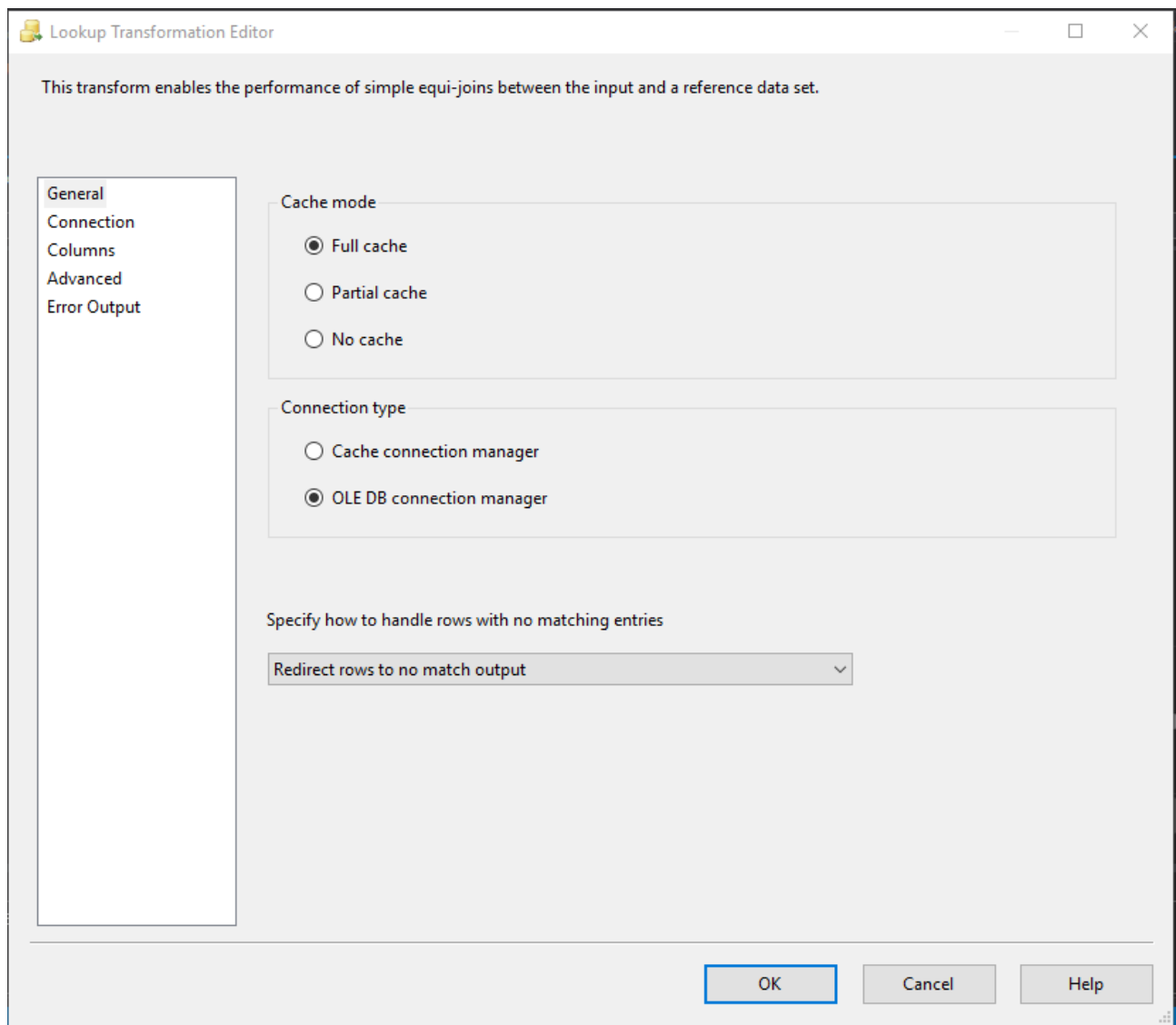
Ahora vamos a darle clic en el input, esto es donde vamos a poner nuestra query para llamar los datos pero primero tenemos que poner la conexión de nuestra base de datos transaccional



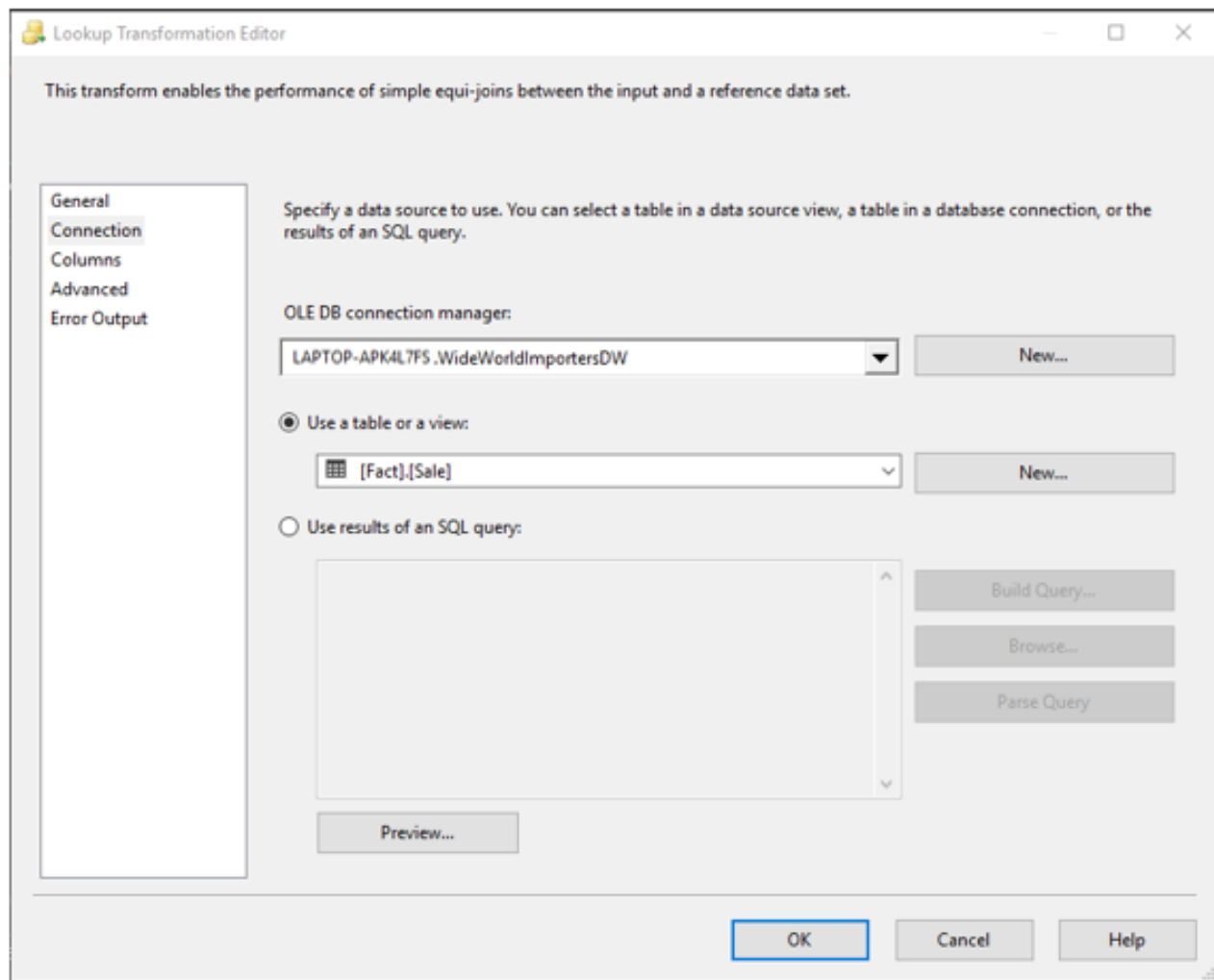
Ahora vamos a columnas y vemos que nos jalo todo los datos que hemos puesto, ahora vamos, y luego vamos a dar clic en ok



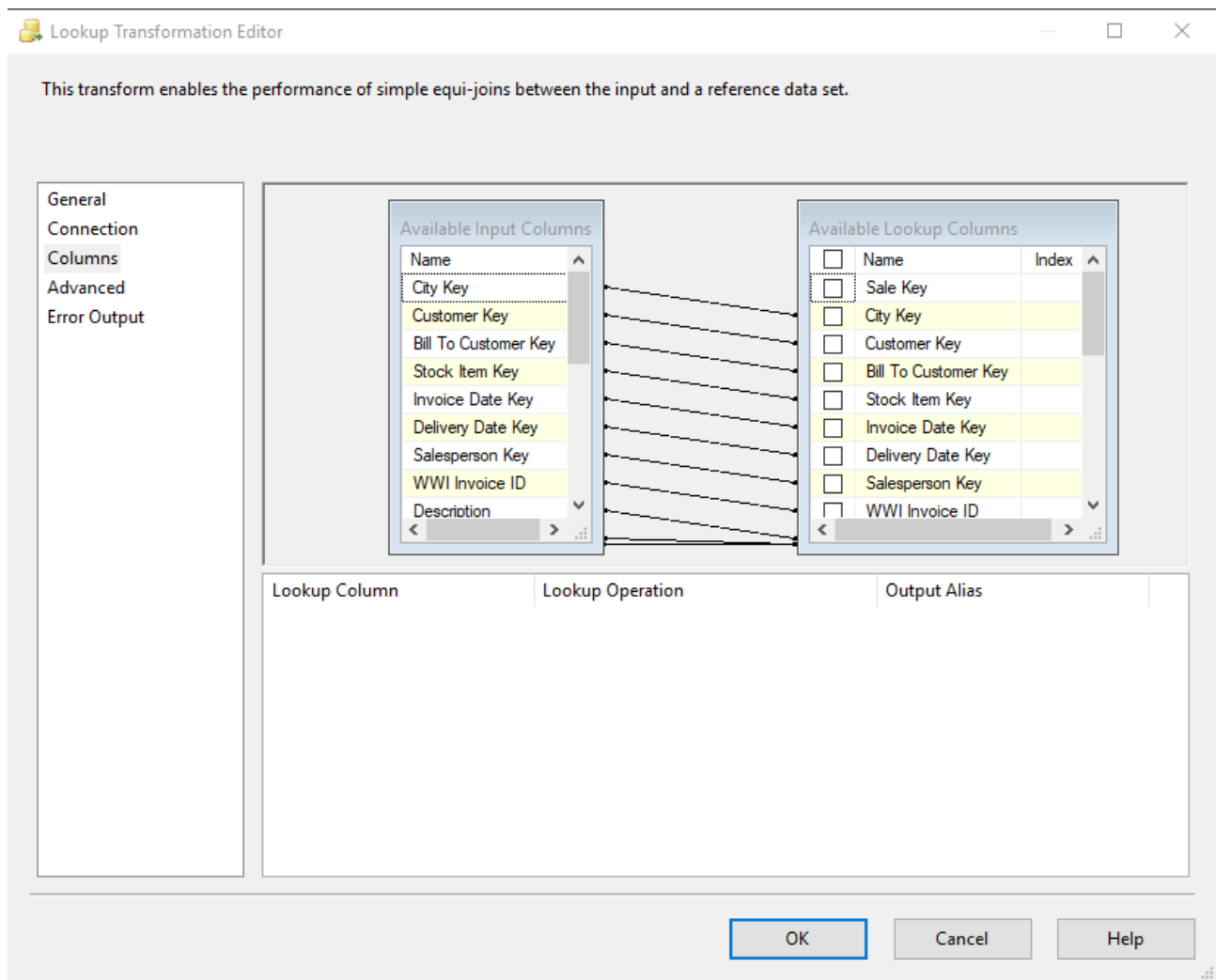
Luego ponemos en general, y ponemos abajo para acabar y luego ponemos en redirect row to no match output, ponemos para que te lo redireccione



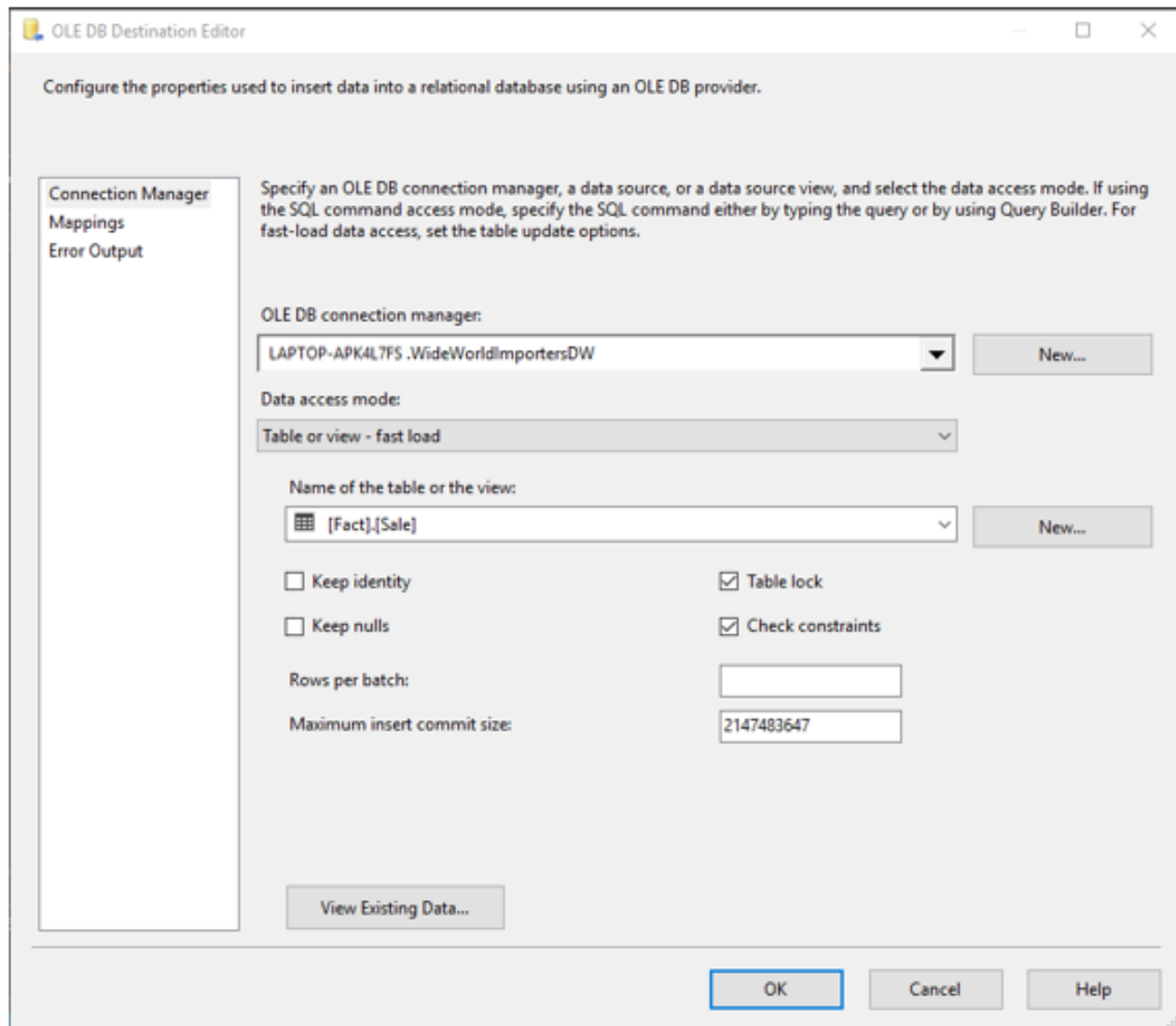
Luego ahora ponemos la conexión que sería DataWarehouse, y luego seleccionamos la tabla que estamos conectando y luego esta tabla tiene que estar vacía sin datos para que te lo pueda extraer los datos, en el insert



Luego ponemos en columnas y como vemos se relaciona normal



Luego ponemos la base de datos del DataWarehouse, luego escogemos la tabla de hecho Fact Sale



Luego ponemos en extracción de datos y como vemos nos jalo todo los datos y luego ponemos clic en close para que lo cierre

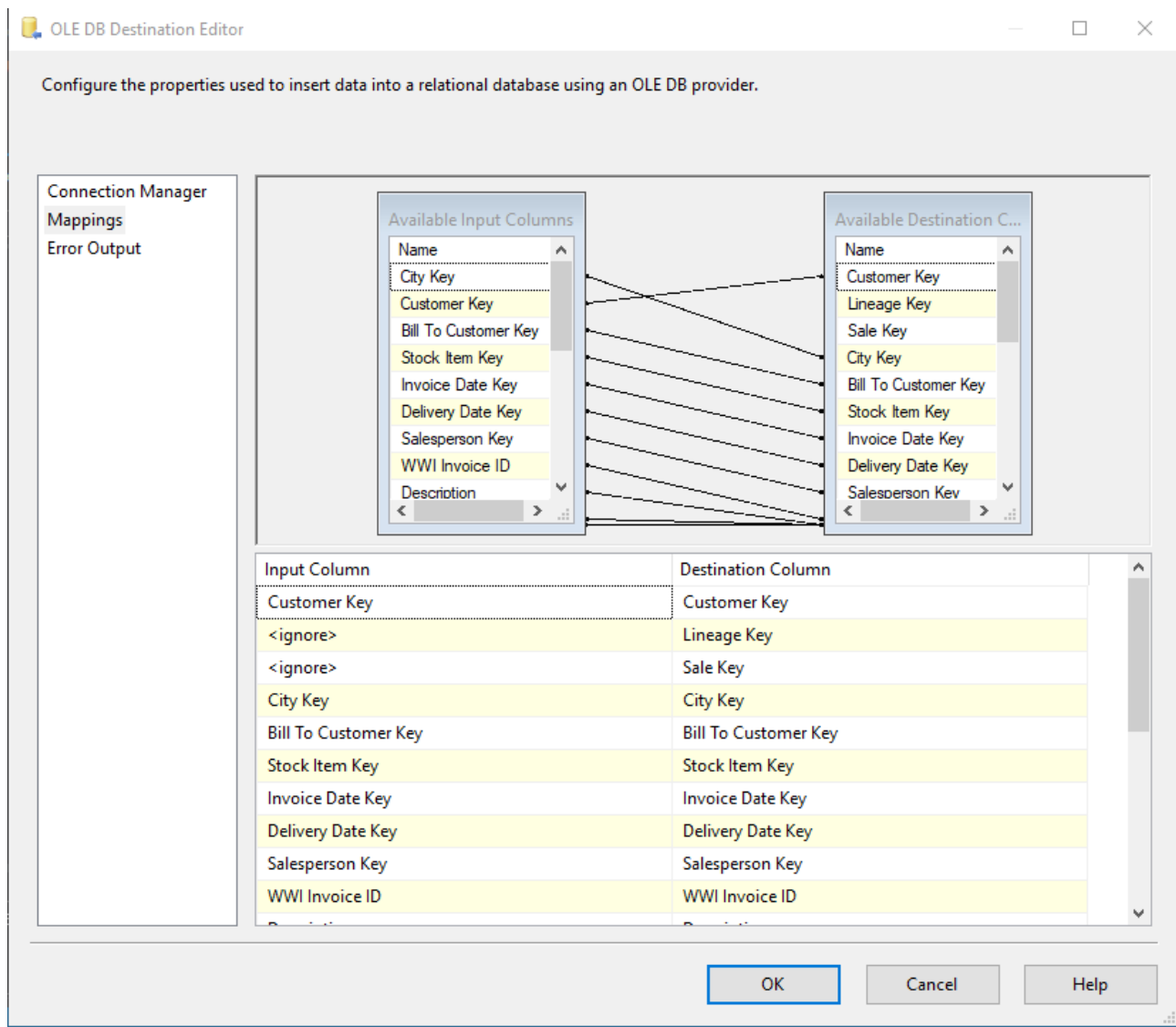
Preview Query Results

Query result (up to the first 200 rows):

Sale Key	City Key	Customer ...	Bill To Cust...	Stock Item...	Invoice Da...
49258	71135	0	0	194	22/10/20...
49265	41568	0	0	204	22/10/20...
49456	70409	0	0	202	22/10/20...
49372	48937	0	0	173	22/10/20...
49624	41981	0	0	168	24/10/20...
49635	70510	0	0	164	24/10/20...
49908	66274	0	0	198	25/10/20...
50034	44106	0	0	180	25/10/20...
50190	56014	0	0	195	26/10/20...
50487	79969	0	0	164	28/10/20...
50367	81342	0	0	180	28/10/20...
50379	91464	0	0	174	28/10/20...
50436	38583	0	0	192	28/10/20...
50715	85575	0	0	190	29/10/20...
50739	81481	0	0	168	29/10/20...

Close

Luego ponemos en mappings, y como vemos se relaciona solamente con cada tabla que ponemos, y le damos clic en ok



2. Conclusiones

- Como conclusión hemos visto como realizar una data warehouse y ETL para nuestro proyecto ya que viendo que es importante que primero realicemos nuestra data warehouse para luego poder seguir con nuestro ETL que nos va permitir mover los datos a múltiples fuentes.