**Part 1 – Introduction EntertainmentAgency**

This database is designed to manage entertainers, agents, customers, and bookings. You would use a similar design to handle event bookings or hotel reservations.

The database handles scheduling of entertainers with customers. We list all the styles of music that each entertainer plays. We also have a table that contains the musical preferences of each customer.

You can see that the Musical\_Preferences table contains a column to rank the customer preferences using a sequence number. In this database, a 1 indicates the customer’s first preference, a 2 the second preference, and so on. There is also a column in the Entertainer\_Styles table that lists for each style that an entertainer can play the relative strength of that style. For example, customer Zachary Johnson has specified a preference for Rhythm and Blues, Jazz, and Salsa in that order. Entertainer Jazz Persuasion says they focus on Rhythm and Blues, Salsa, and Jazz in that order.

* Download from Chamilo and execute the scripts Documenten > Scripts > EntertainmentAgency.sql
* Create the Database Diagram



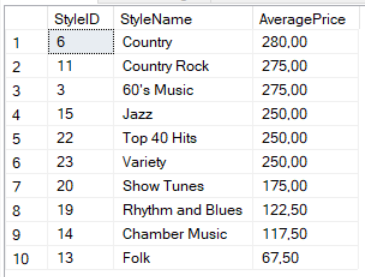
**Part 2 – 6 SQL Queries to get to know EntertainmentAgency**

Find the appropriate SQL queries to solve these questions (year = year of the startdate)

-- Query 1

-- What is the average price per musical style based on EntPricePerDay.

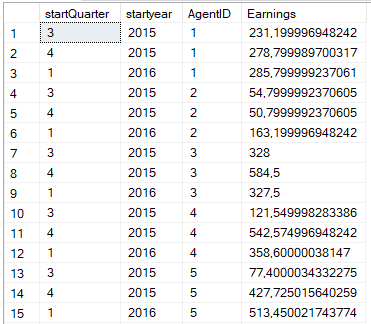
-- Only take into account StyleStrength = 1. Order by average price in descending way.



-- Query 2

-- What are the earnings of each agent per quarter

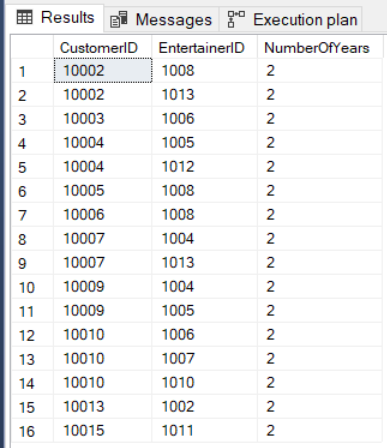
-- Only take into account the commissionrate of the agent and the contractprice



-- Query 3

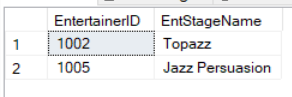
-- What are the customers that have booked the same entertainer every year (every year = every year bookings are registered).

-- The image below shows only a part of the resultset



-- Query 4

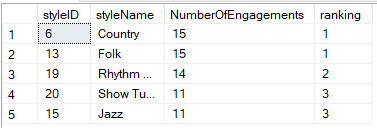
-- We are looking for a Rhythm and Blues group. Which entertainer is the most popular one?



-- Query 5

-- What is the TOP 3 of most booked musical styles

-- Only take into account StyleStrength = 1.



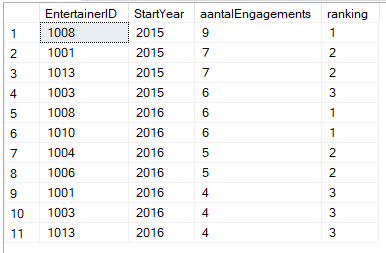
-- Query 6

-- Give for each year the top 3 of most popular entertainers (= entertainers with most engagements for that year)

-- first step: count van de engagements per entertainer per year

-- second step: dense\_rank

-- third step: top 3



**Part 3 – Design EntertainmentAgencyDW**

Design the starschema for a simple DW based on the EntertainmentAgency OLTP database

The following requirements must be met

1. 1 fact = 1 engagement. Precalculated fields are:
   * NumberOfDays of the engagement (= number of days between StartDate and EndDate. Don't forget to do + 1)
   * NumberOfHours of the engagement (= number of hours between StartTime and StopTime)
   * ContractPrice of the engagement
   * CommissionAgent (= the commission the agent received)
2. Time aspects
   * One should be able to make reports based on name of the month (english or dutch), year, day of the week (english or dutch), quarter, …
3. One should be able to make reports based on Customers, Agents, Entertainers and Musical\_Styles
   * Customers is the only slowly changing dimension.
   * One should be able to include in a report information about CustFirstName, CustLastName, CustCity, CustZipCode and CustState
   * One should be able to include in a report information about AgtFirstName, AgtLastName, AgtCity, AgtZipCode and AgtState
   * One should be able to include in a report information about EntStageName, EntCity, EntZipCode and EntState
   * One should be able to include in a report information about Musical\_Styles. Assume that the musical style of an engagement matches the main style (stylestrength = 1) of the entertainer

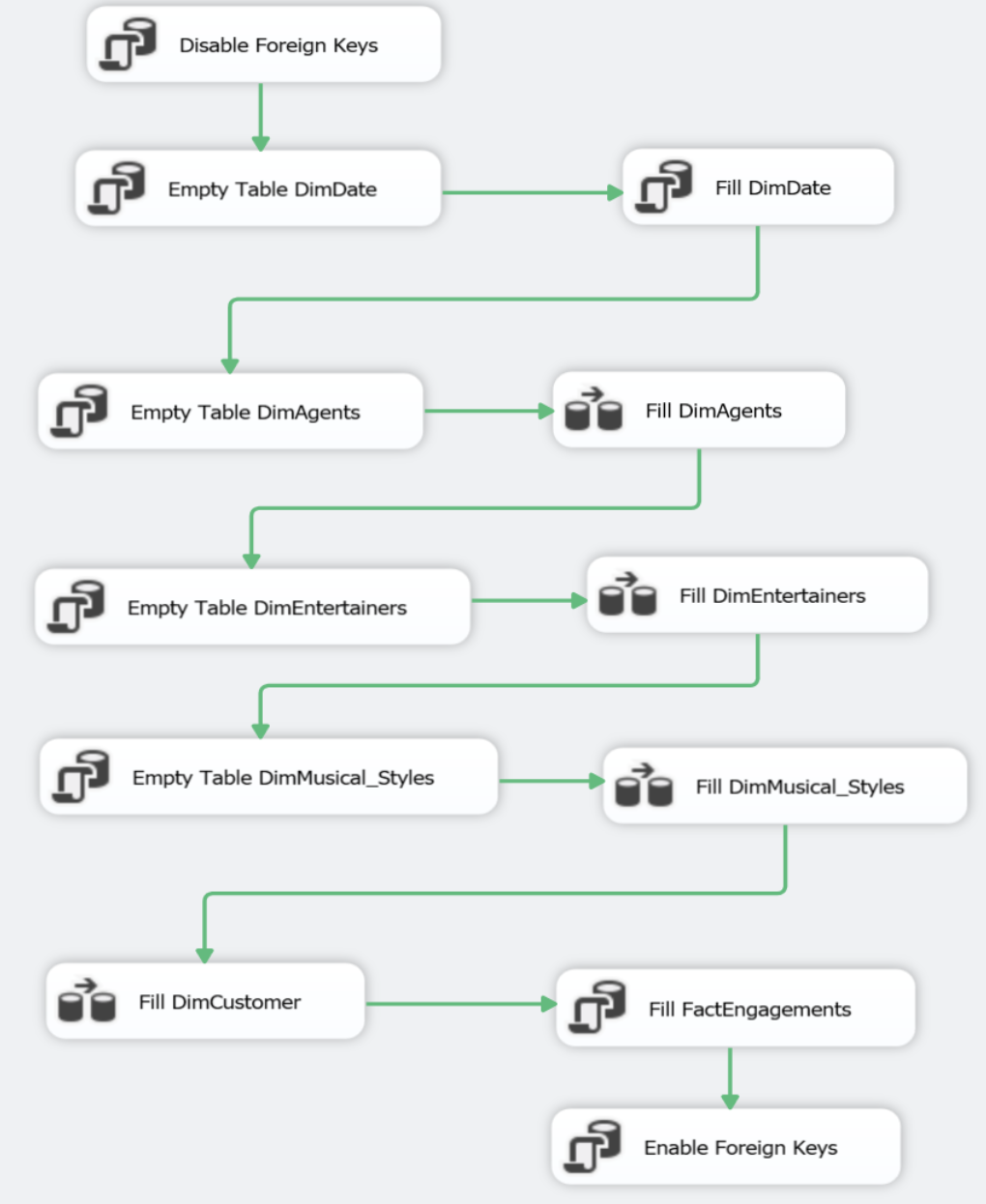
**Part 4 – Create the script for EntertainmentAgencyDW**

1. Download from Chamilo the file Documenten > Scripts Schema\_DimDate.sql. In this file you find the code to create DimDate
2. Add extra code to create the dimensions and the fact table of EntertainmentAgencyDW
   * DimAgents
   * DimEntertainers
   * DimMusical\_Styles
   * DimCustomers. Specify CustomerKey as an identity value with seed = 1 and increment = 1.
   * FactEngagements.   
     **Make sure to add the Foreign Key constraints!**
     + StartDateKey in FactEngagements is a foreign key to DateKey in DimDate
     + EndDateKey in FactEngagements is a foreign key to DateKey in DimDate
     + AgentKey in FactEngagements is a foreign key to AgentKey in DimAgents
     + CustomerKey in FactEngagements is a foreign key to CustomerKey in DimCustomer
     + EntertainerKey in FactEngagements is a foreign key to EntertainerKey in DimEntertainers
     + Musical\_StyleKey in FactEngagements is a foreign key to Musical\_StyleKey in DimMusical\_Styles

**Part 5 – Create ETL pipeline to fill EntertainmentAgencyDW using SSIS**

!Use the document AdventureWorks Intro – Startschema - Filling if you don't know how to create and fill EntertainmentAgencyDW!

This is the pipeline that should be created



1. Create the necessary views in the OLTP database EntertainmentAgency.

A view is needed to fill

* + DimAgents
  + DimEntertainers
  + DimMusical\_Styles

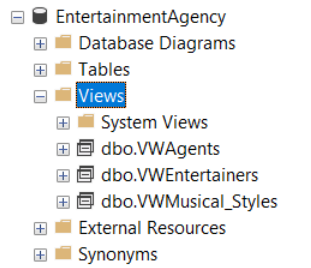
1. Create the view VWAgents in SQL Server Management Studio in the OLTP database EntertainmentAgency

CREATE OR ALTER VIEW VWAgents AS

SELECT AgentID As AgentKey, AgtFirstName, AgtLastName, AgtCity, AgtState, AgtZipCode

FROM Agents

1. Check if this view is available in EntertainmentAgency and is filled with the data of the agents
2. Create fully analog the views VWEntertainers and VWMusical\_Styles
3. Check if the views are available in EntertainmentAgency and are filled with the correct dta



1. Using SSIS as ETL Tool
2. Start Visual Studio 2019
3. Click on Create a new Project

Afbeelding met tekst

Automatisch gegenereerde beschrijving

1. Type integration
2. Click on Integration Services Project

Afbeelding met tekst

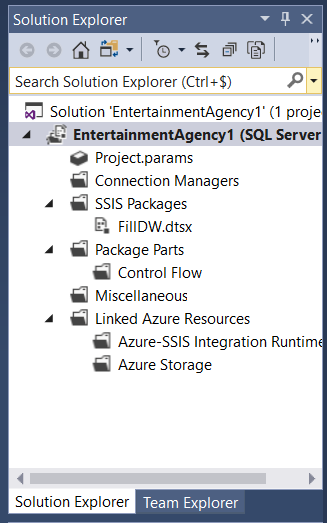
Automatisch gegenereerde beschrijving

1. Project Name = EntertaintmentAgencyProject + Click on Create

Afbeelding met tekst

Automatisch gegenereerde beschrijving

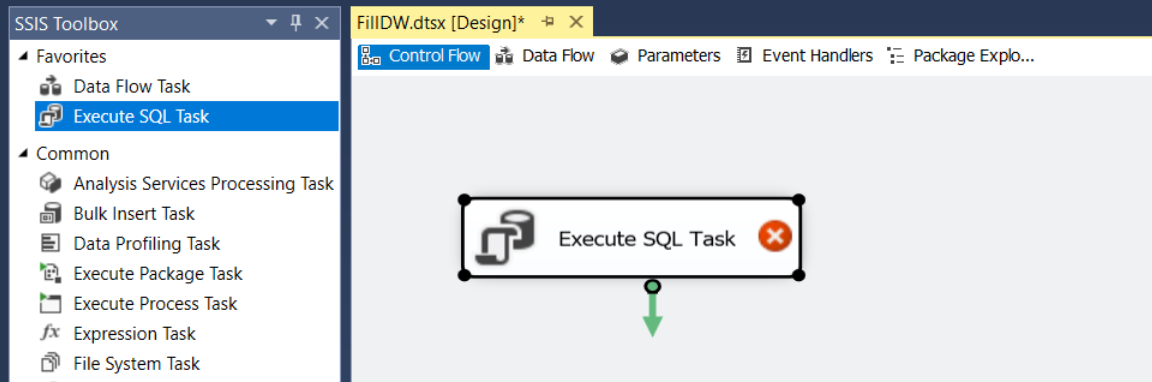
1. In the Solution Explorer (at the right side) rename package.dtsx to FillDW.dtsx



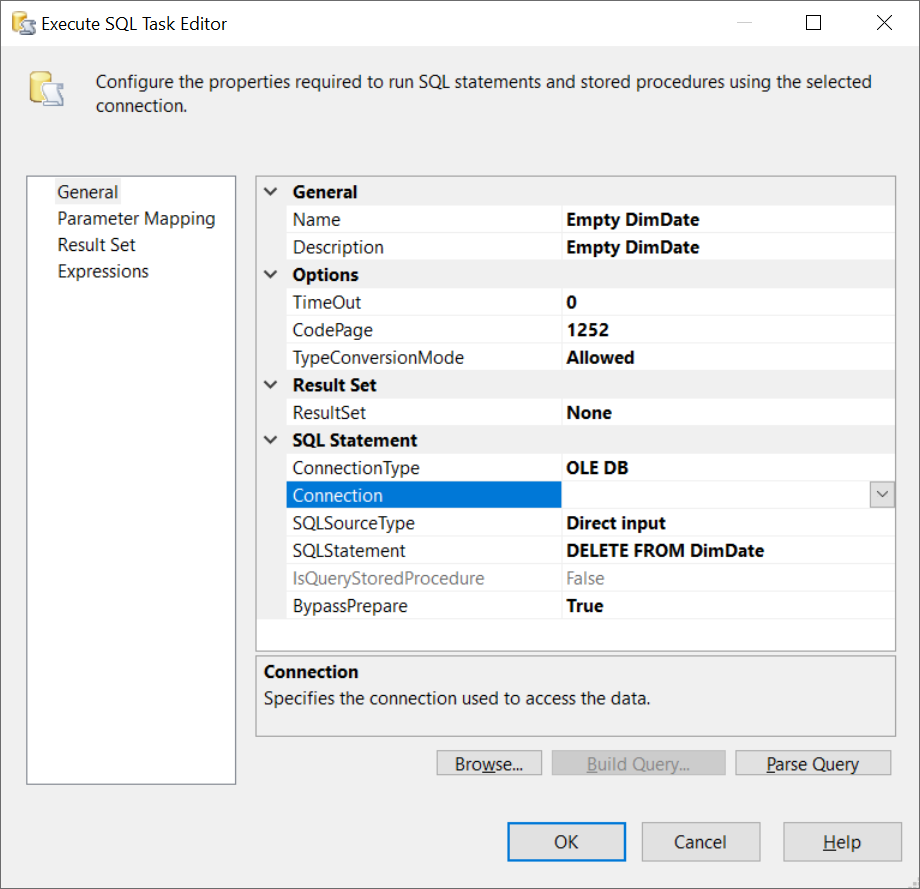
1. Fill DimDate  Fill dimension using a script.



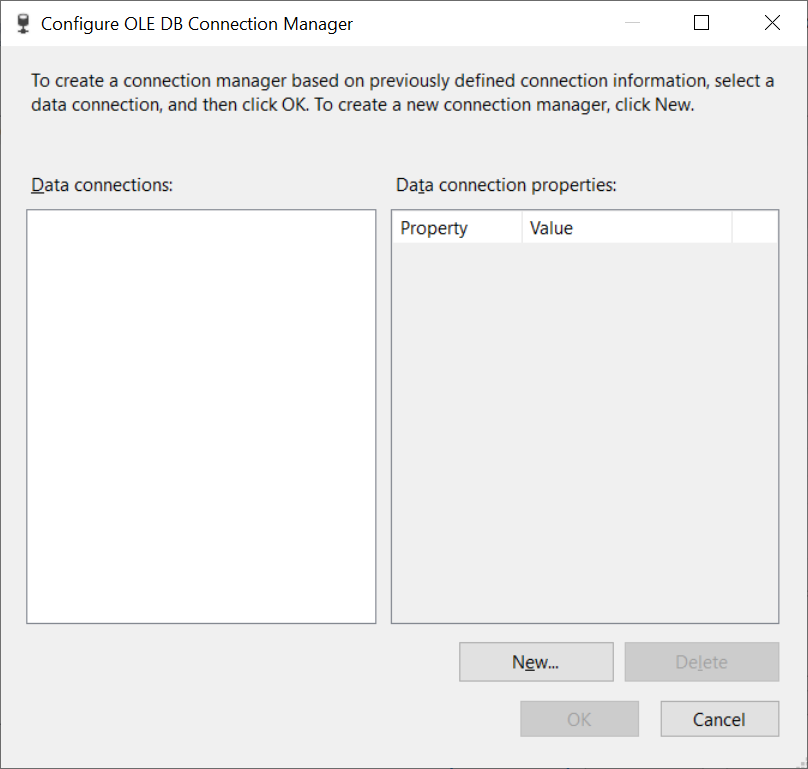
1. Use the script you find on Chamilo: Documenten > Scripts > Fill\_DimDate.sql
2. Drag and drop 'Execute SQL Task' from the SSIS Toolbox (at the left) to the Control Flow area



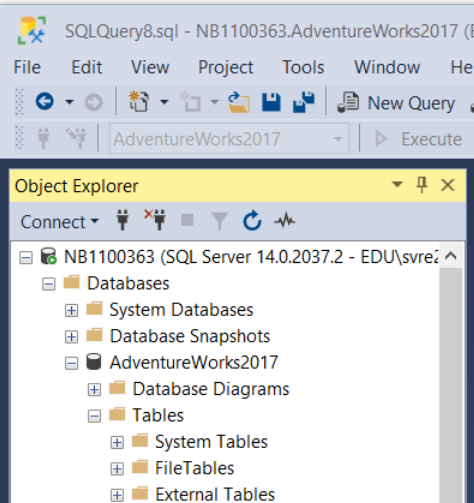
1. We will use this SQL Task to delete all records from DimDate before each transfer.
2. Right click on the 'Execute SQL Task' and choose Edit… to configure the 'Empty DimDate' task

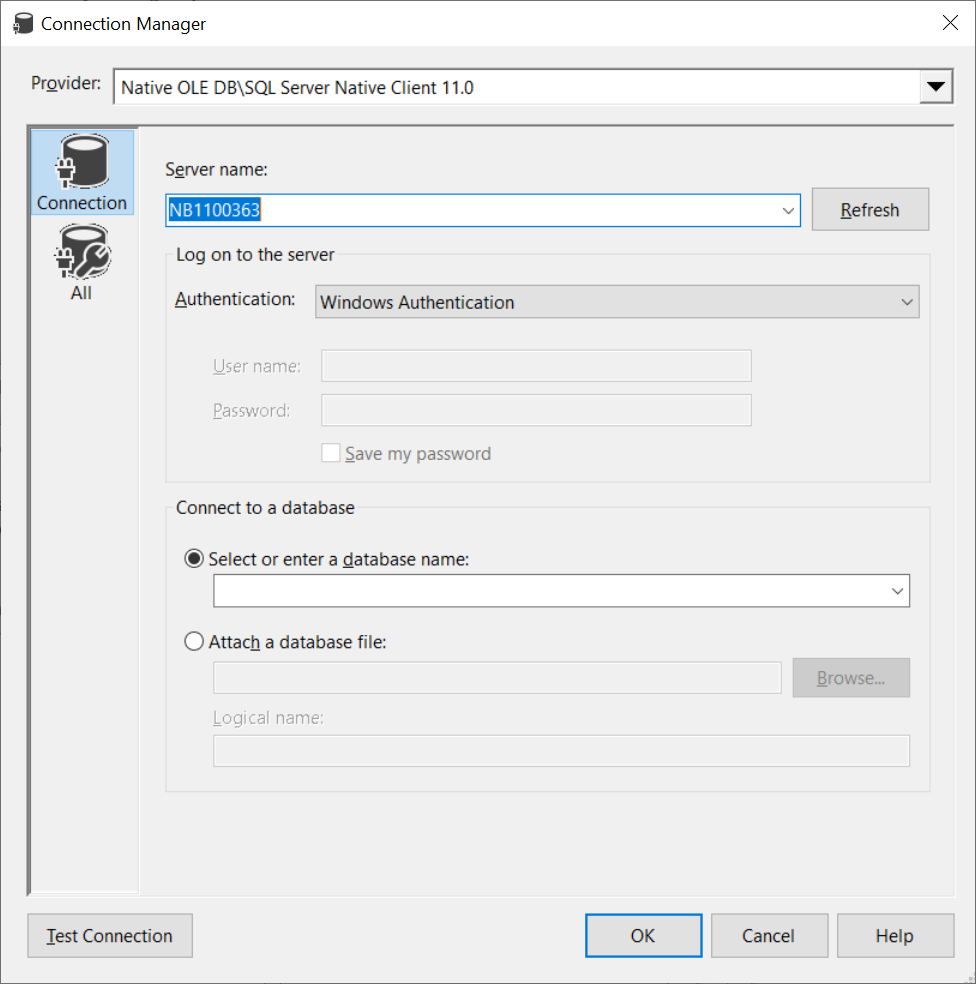


1. The Connection is still missing and should be configured
   * Click on the arrow next to Connection
   * Choose New connection
   * Click New…



* + Fill in the Server name. You don't have to use the dropdownlist, just fill in the Server name. You can find the Server name in Microsoft SQL Server Management Studio

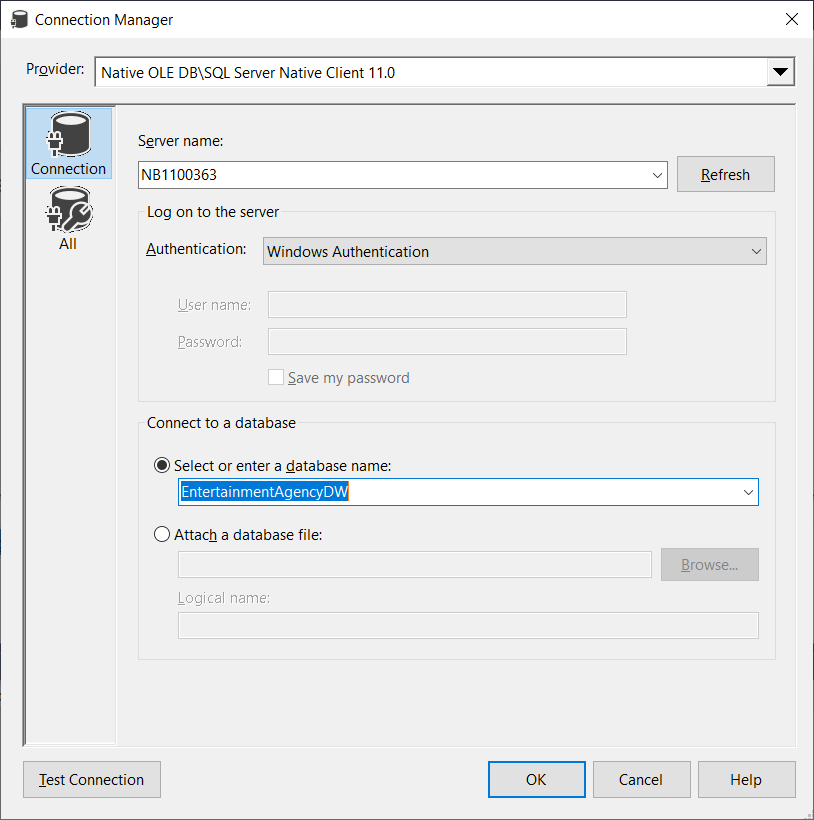




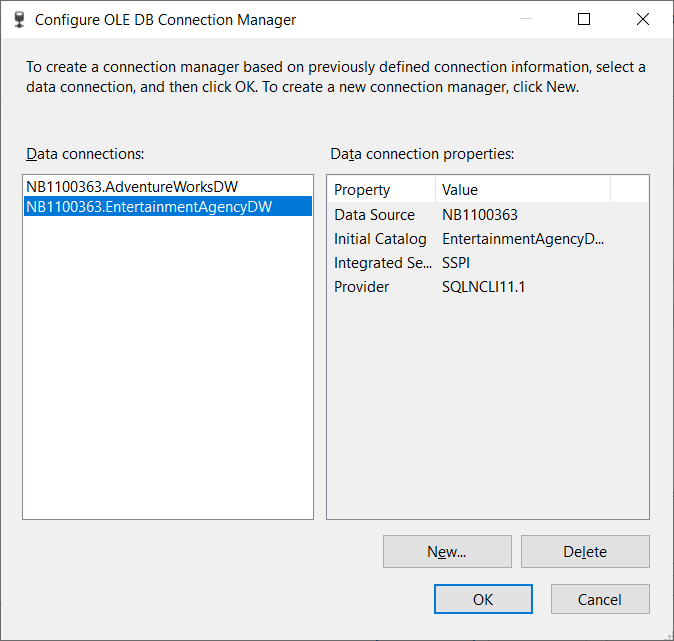
* + Select the correct database in the dropdownlist 'Select or enter a database name'. Because this Connection will be used to connect to EntertainmentAgencyDW, this is of course the name of the database we select.
  + Click Test Connection



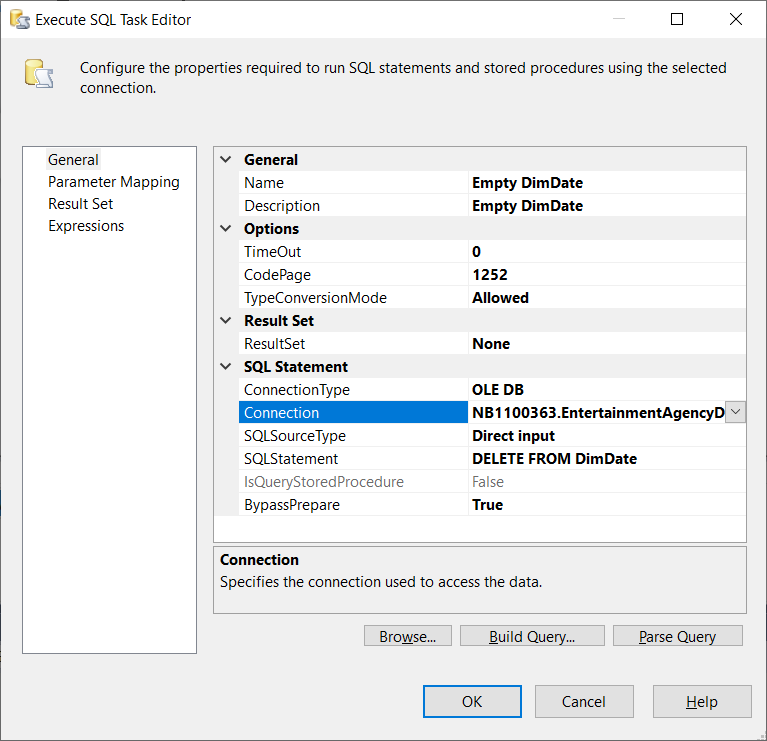
* + Click OK to close this window



* + Click OK



* + Click OK

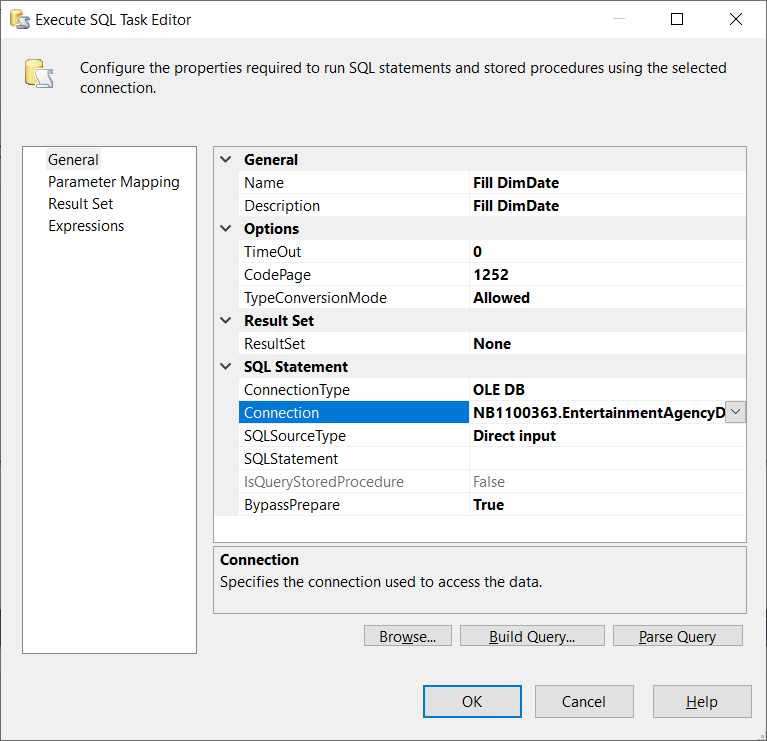


* + Click OK

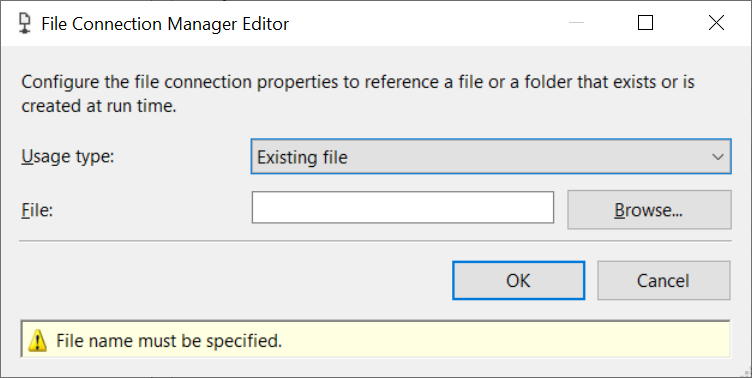
1. Drag and drop 'Execute SQL Task' from the SSIS Toolbox (at the left) to the Control Flow area
2. We will use this SQL Task to run the script Fill\_DimDate to fill DimDate.
3. Connect both tasks



1. Right click on the 'Execute SQL Task' and choose Edit… to configure the 'Fill DimDate' task



1. The FileConnection is still missing and should be configured
   * Click on the arrow next to FileConnection
   * Choose New connection

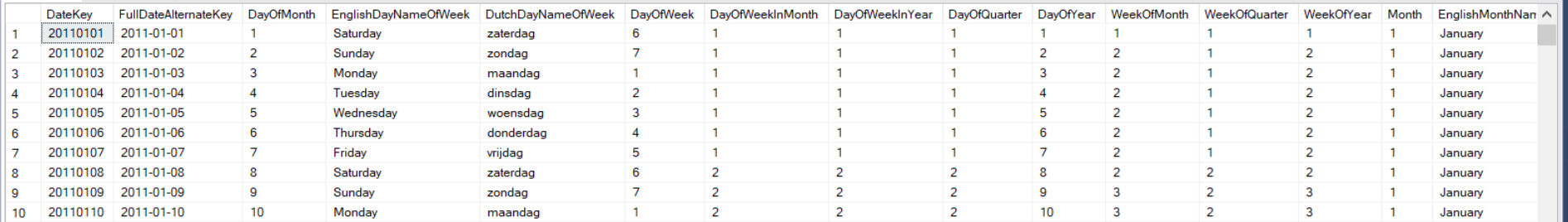


* + Click Browse… and select Fill\_DimDate.sql
  + Click OK

1. Before you continue, check if the current flow is already working by clicking on Start 
2. Click on Stop 



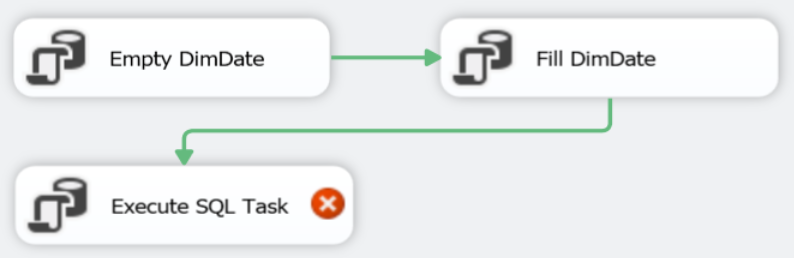
1. Go to Microsoft SQL Server Management Studio and check if EntertainmentAgencyDW > DimDate contains the correct data



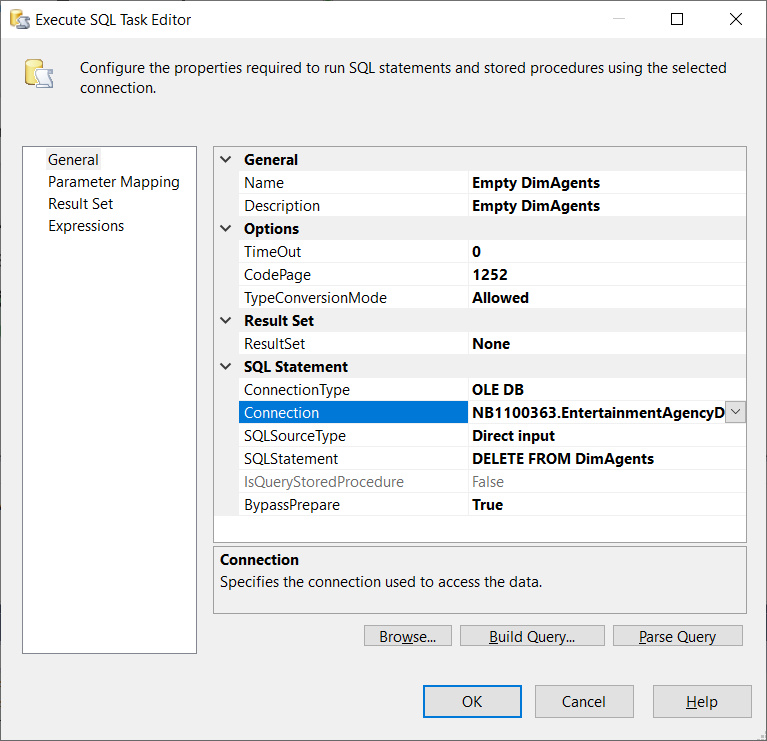
1. Fill DimAgents  Fill dimension using a view.



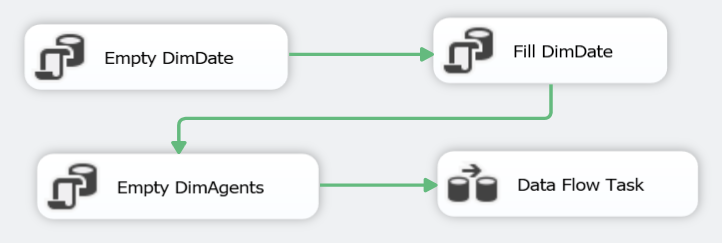
1. Drag and drop 'Execute SQL Task' from the SSIS Toolbox (at the left) to the Control Flow area.  
   Connect with the previous tasks



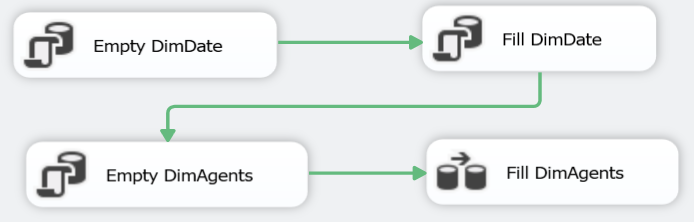
1. We will use this SQL Task to delete all records from DimAgents before each transfer.
2. Right click on the 'Execute SQL Task' and choose Edit… to configure the 'Empty DimAgents' task



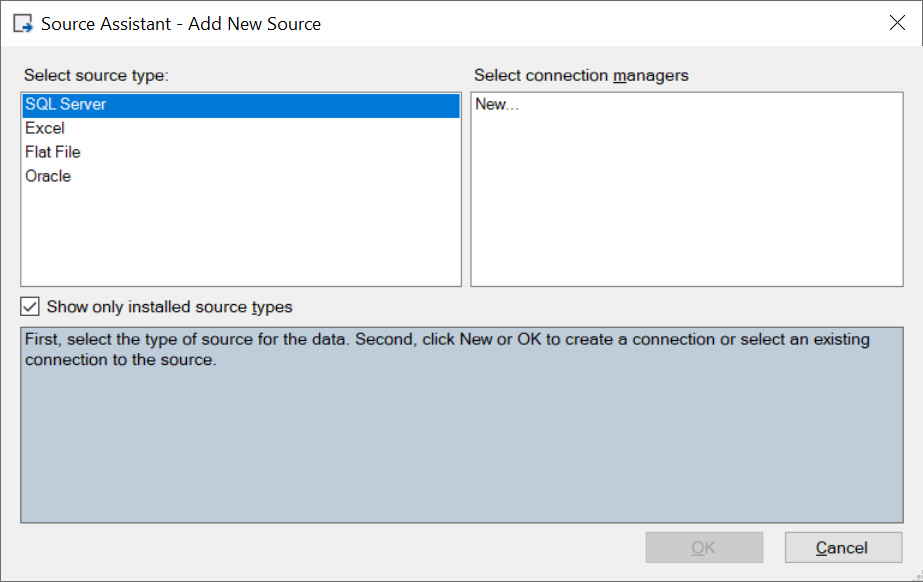
1. To 'refill' DimAgents drag and drop 'Data Flow Task' from the SSIS Toolbox (at the left) to the Control Flow area.   
   Connect with the previous tasks.



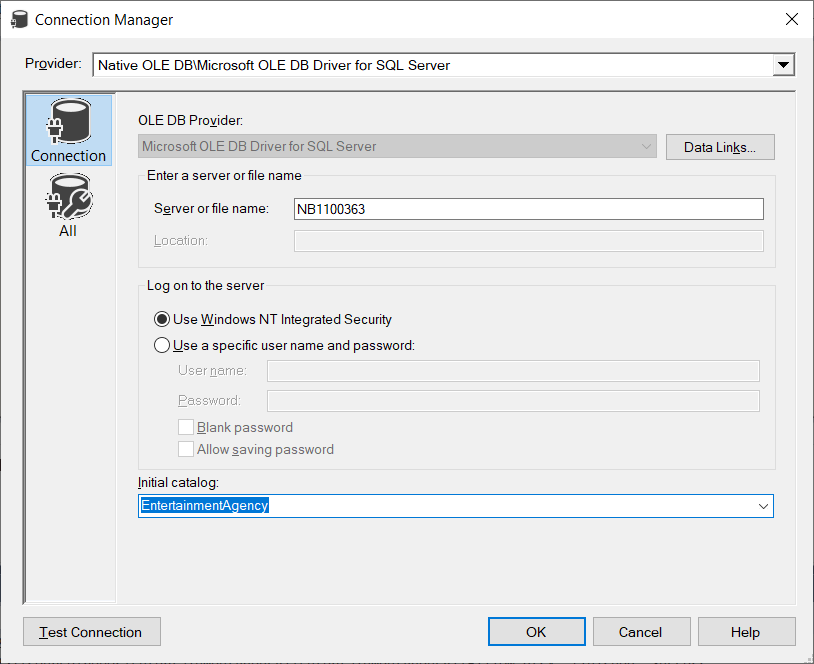
1. Right click on Data Flow Task and choose Rename, to rename the 'Data Flow Task' to 'Fill DimAgents'



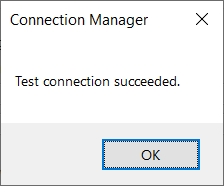
1. Right click on 'Fill DimAgents' and choose Edit… to configure the Fill DimAgents' task.
2. Notice that you are now in the 'Data Flow' area
3. Drag and drop 'Source Assistant' from the SSIS Toolbox (at the left) to the Data Flow area



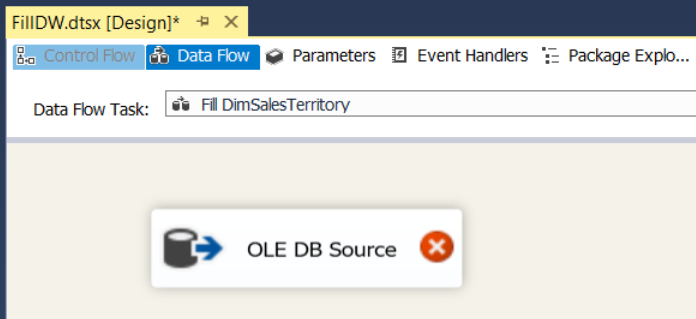
1. Select New… in the Select connection managers at the right side. Click OK.
2. We need to connect to the OLTP database EntertainmentAgendy and we don't have a connection to this database yet, that's why we need a new connection. We do have a connection with the DWH EntertainmentAgencyDW, but that's another database.
3. Enter the Server name, the same as you did to establish the connection with the DWH
4. Choose the correct database in the dropdownlist Initial catalog



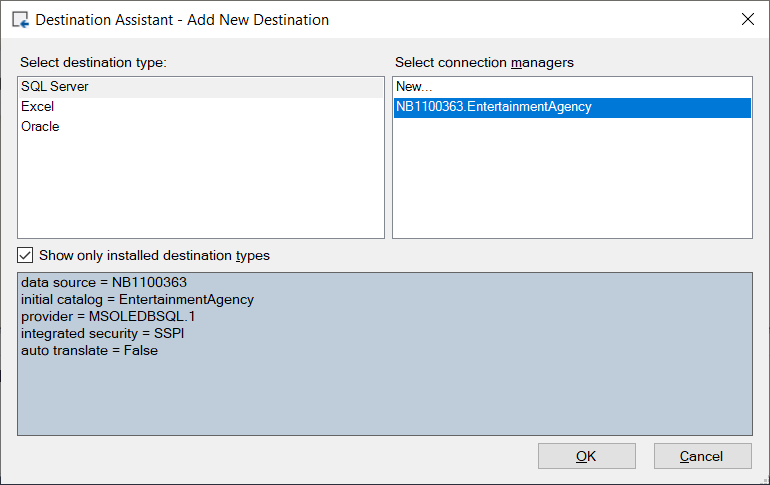
1. Click Test Connection



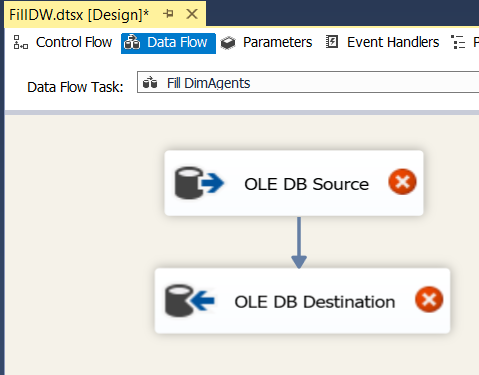
1. Click OK
2. Click OK



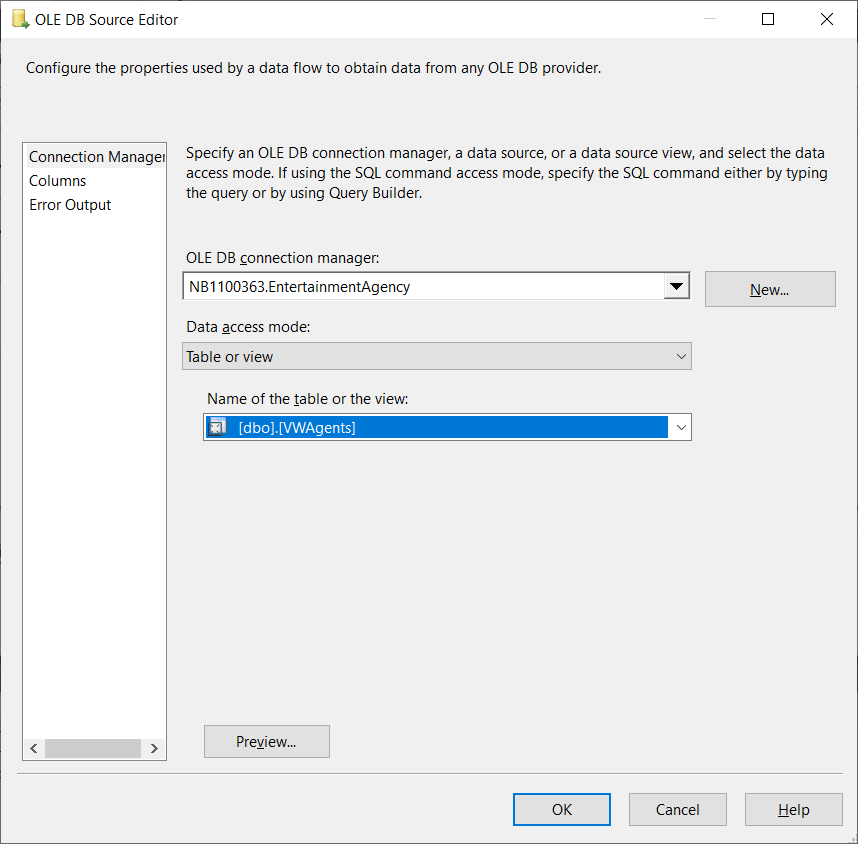
1. Drag and drop 'Destination Assistant' from the SSIS Toolbox (at the left) to the Data Flow area
2. Select the connection manager at the right side



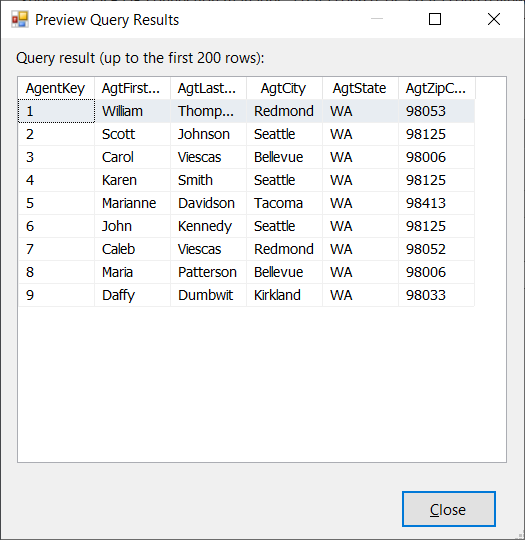
1. Click OK
2. Connect the Source Assistant and the Destination Assistant



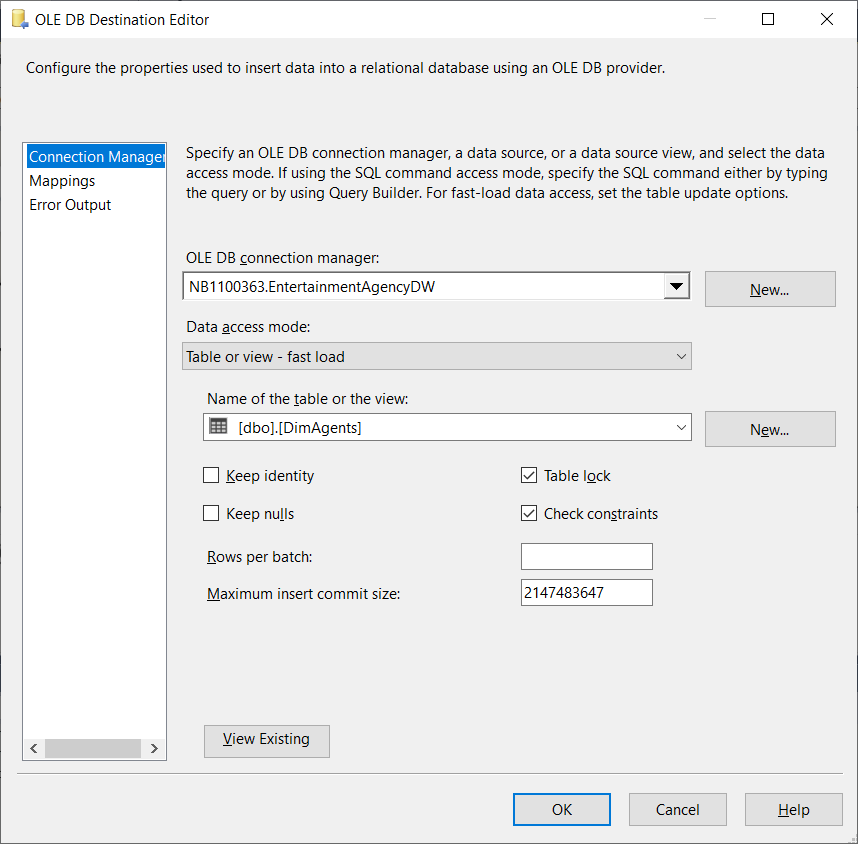
1. Right click on 'OLE DB Source' and choose Edit… to configure the source.
2. Choose VWAgents as the correct data source



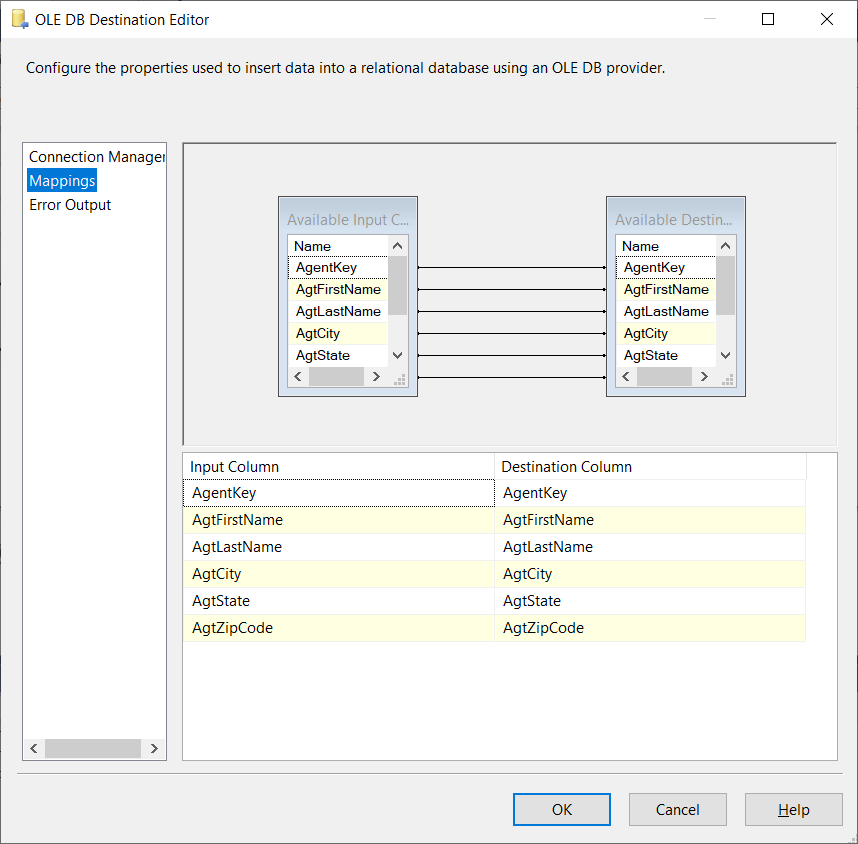
1. Click Preview to check the data



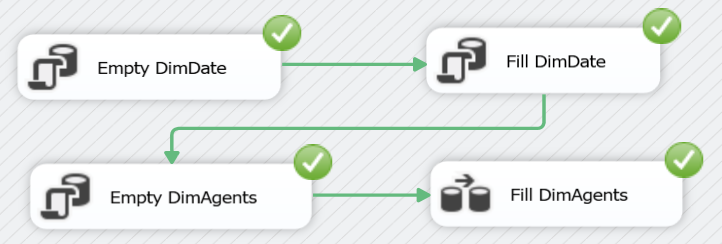
1. Click Close
2. Click OK
3. Right click on 'OLE DB Destination' and choose Edit… to configure the destination.
4. Choose the appropriate values for OLE DB connection manager (= the connection manager for the DWH!!) and for the table



1. Click on Mappings on the left to check if the Mappings between the view VWAgents in EntertainmentAgency and the table DimAgents are correct
2. Click OK



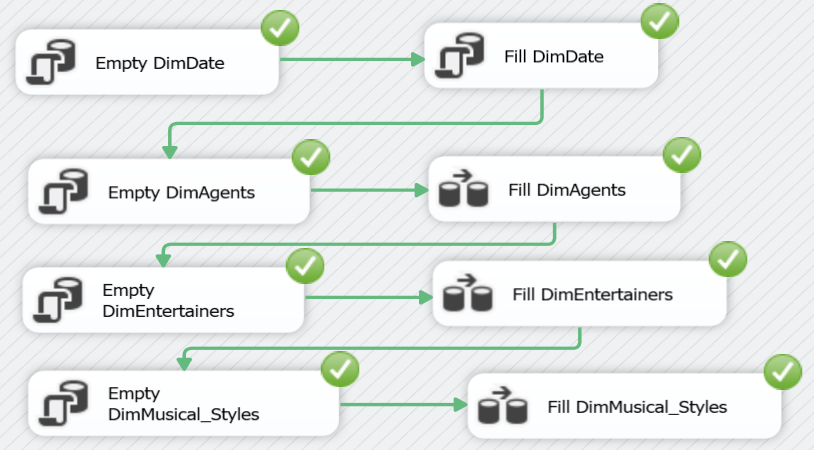
1. Go back to the Control Flow area and run the provisional pipeline



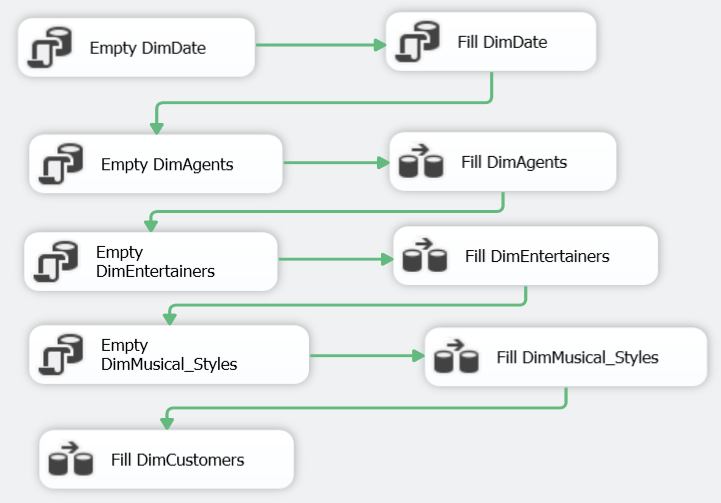
1. Go to Microsoft SQL Server Management Studio and check if EntertainmentAgencyDW > DimDate and EntertainmentAgencyDW > DimAgents contain the correct data



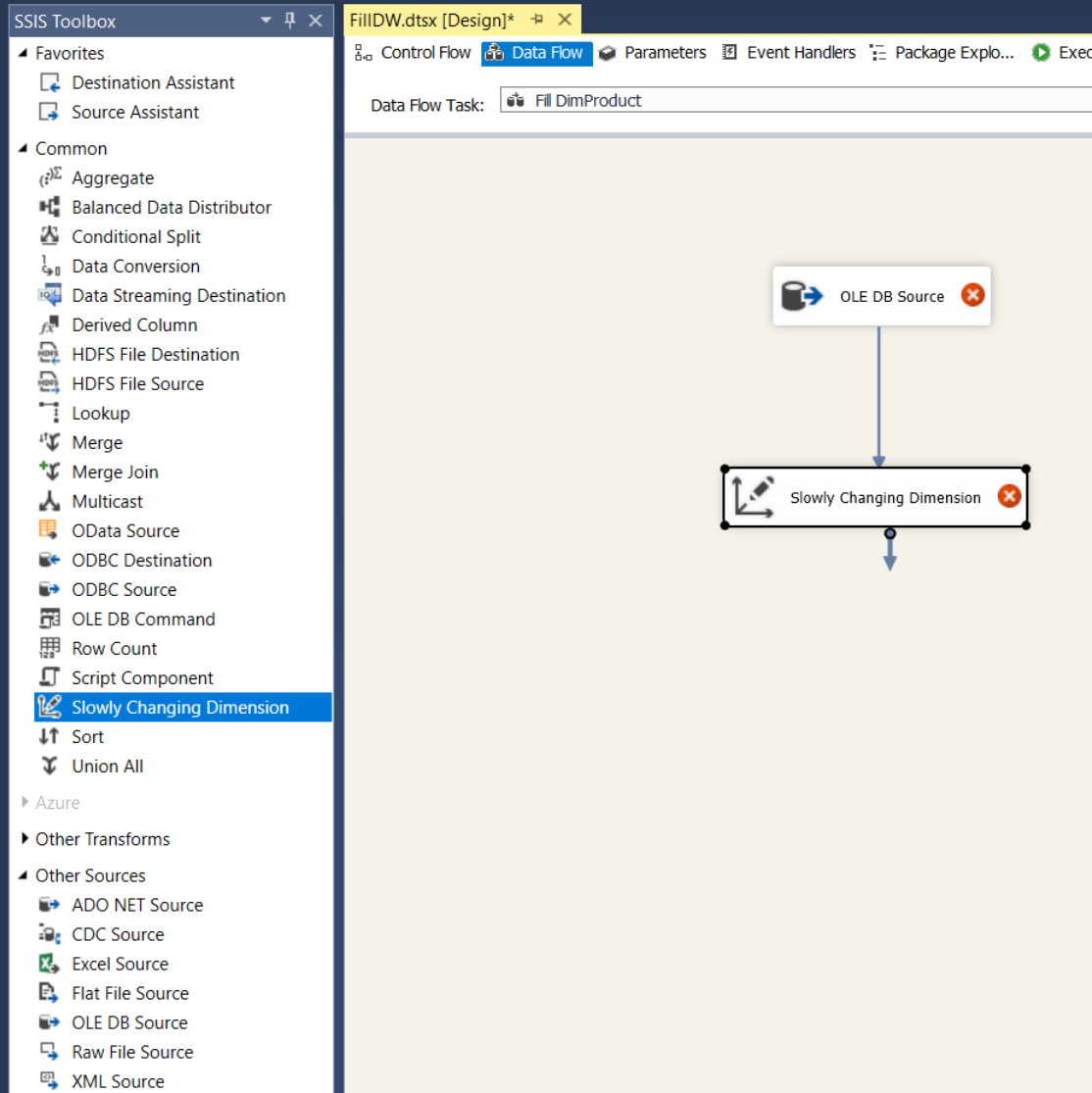
1. Fill DimEntertainers  Fill dimension using a view.
2. Repeat the previous steps to Empty and Fill DimEntertainers
3. Run the provisional pipeline
4. Go to Microsoft SQL Server Management Studio and check if EntertainmentAgencyDW > DimDate and EntertainmentAgencyDW > DimAgents and EntertainmentAgencyDW > DimEntertainers
5. Fill DimMusical\_Styles 🡪 Fill dimension using a view.
6. Repeat the previous steps to Empty and Fill DimMusical\_Styles
7. Run the provisional pipeline
8. Go to Microsoft SQL Server Management Studio and check if EntertainmentAgencyDW > DimDate and EntertainmentAgencyDW > DimAgents and EntertainmentAgencyDW > DimEntertainers and EntertainmentAgencyDW > DimMusical\_Styles contain the correct data



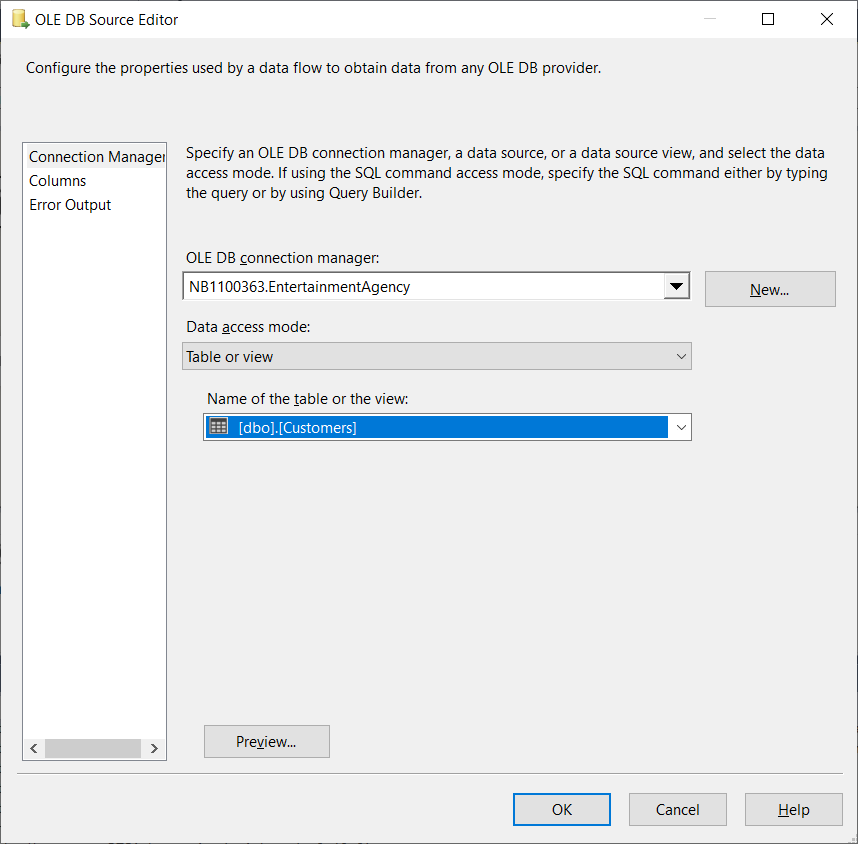
1. Fill DimCustomers 🡪 Slowly changing dimension
2. Drag and drop 'Data Flow Task' from the SSIS Toolbox (at the left) to the Control Flow area.
3. Right click on Data Flow Task and choose Rename, to rename the 'Data Flow Task' to 'Fill DimCustomers'.
4. Connect to the previous tasks



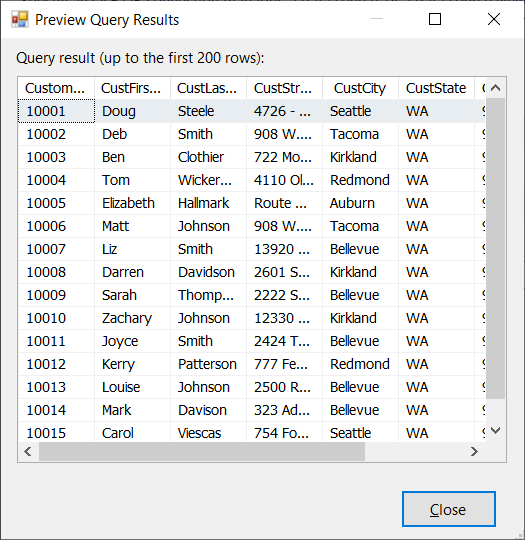
1. We don't need a DELETE task since we want to build customer history!
2. Right click on Fill DimCustomers and choose Edit… to configure the 'Fill DimCustomers' task
3. Notice that you are now in the 'Data Flow' area
4. Drag and drop 'OLE DB Source' and 'Slowly Changing Dimension' from the SSIS Toolbox (at the left) to the Data Flow area
5. Connect both



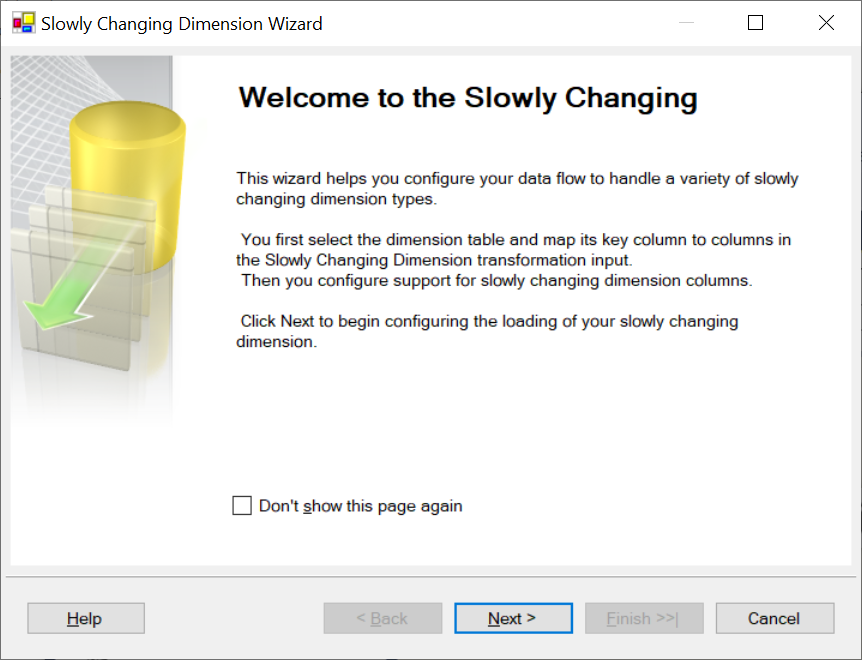
1. Right click on OLE DB Source and choose Edit… to specify the Customer table from the OLTP database EntertainmentAgency as the OLE DB Source



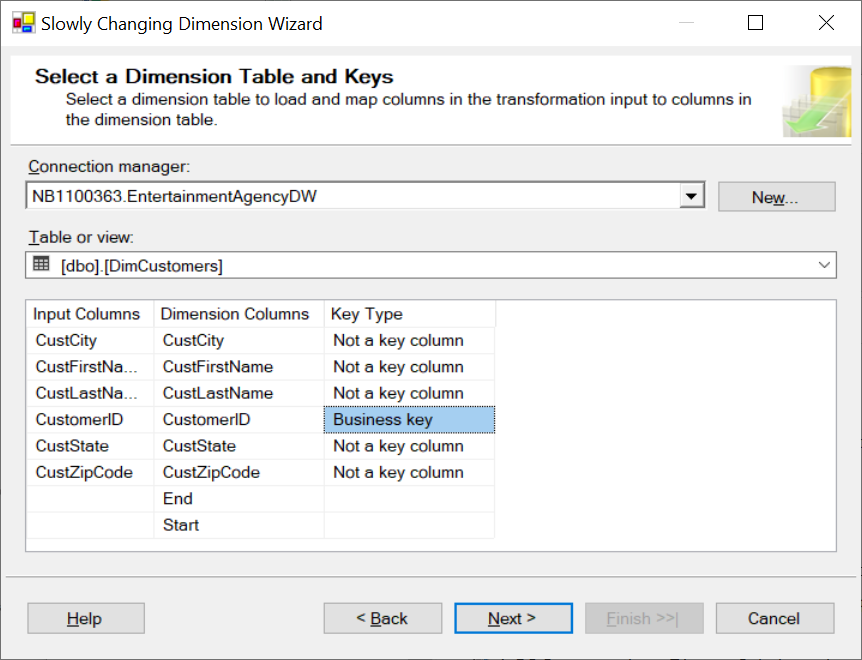
1. Click Preview…



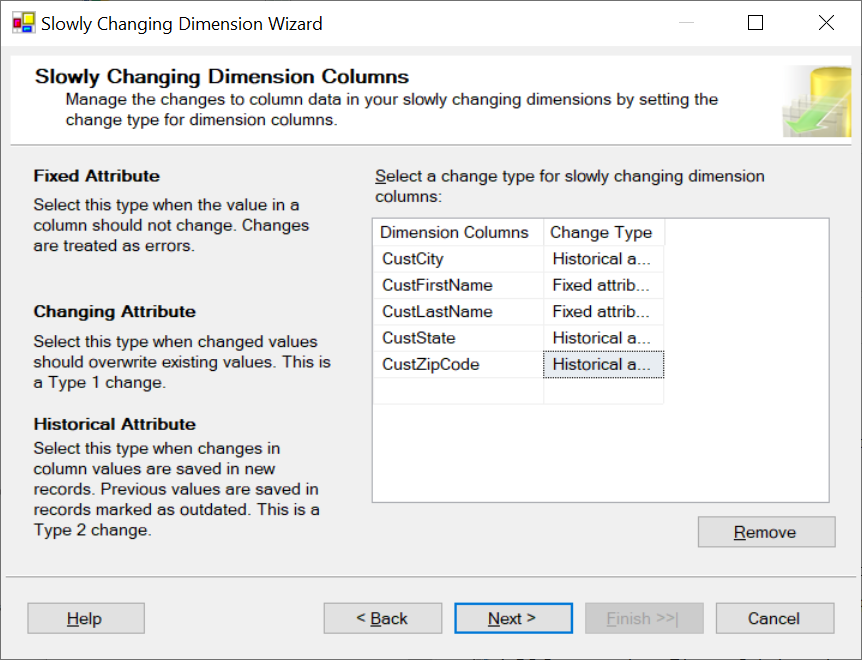
1. Click Close
2. Click OK
3. Right click on Slowly Changing Dimension and choose Edit… to start the wizard for the Slowly Changing Dimenion task



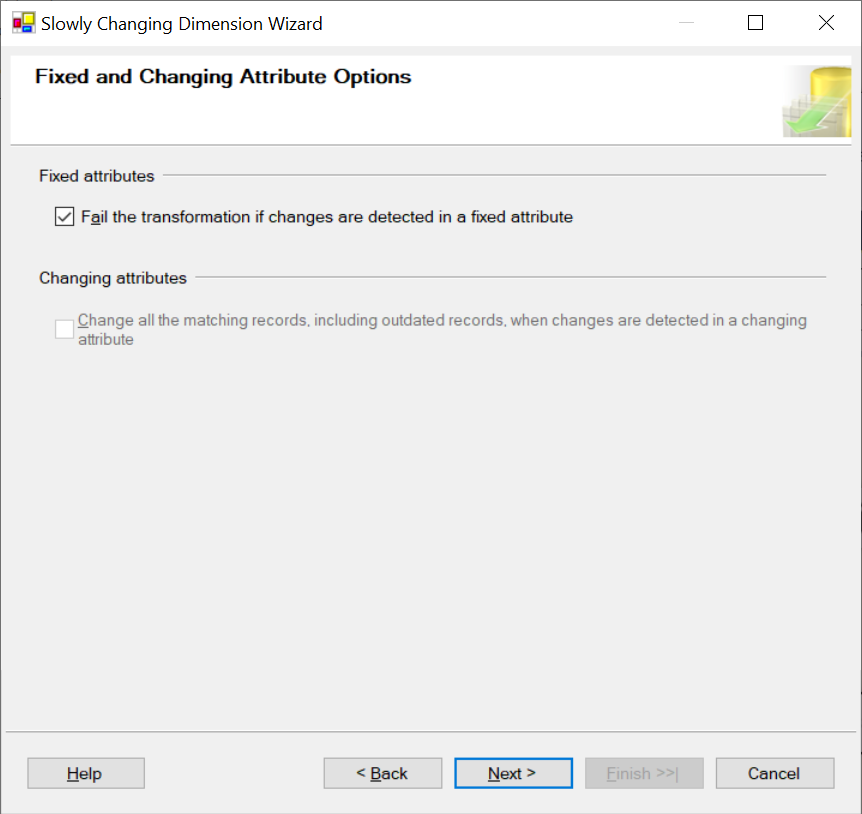
1. Click Next
2. Choose the appropriate values for OLE DB connection manager (= the connection manager for the DWH!!) and for the table  
   Change the Key Type next to CustomerID to Business key



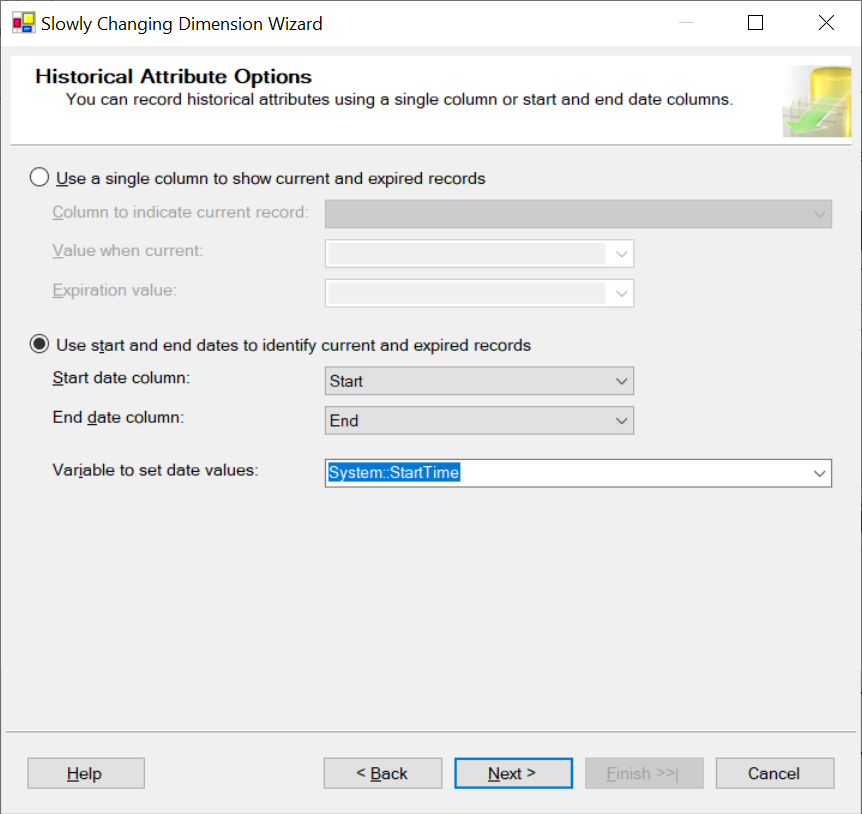
1. Click Next
2. Select a change type for slowly changing dimension columns



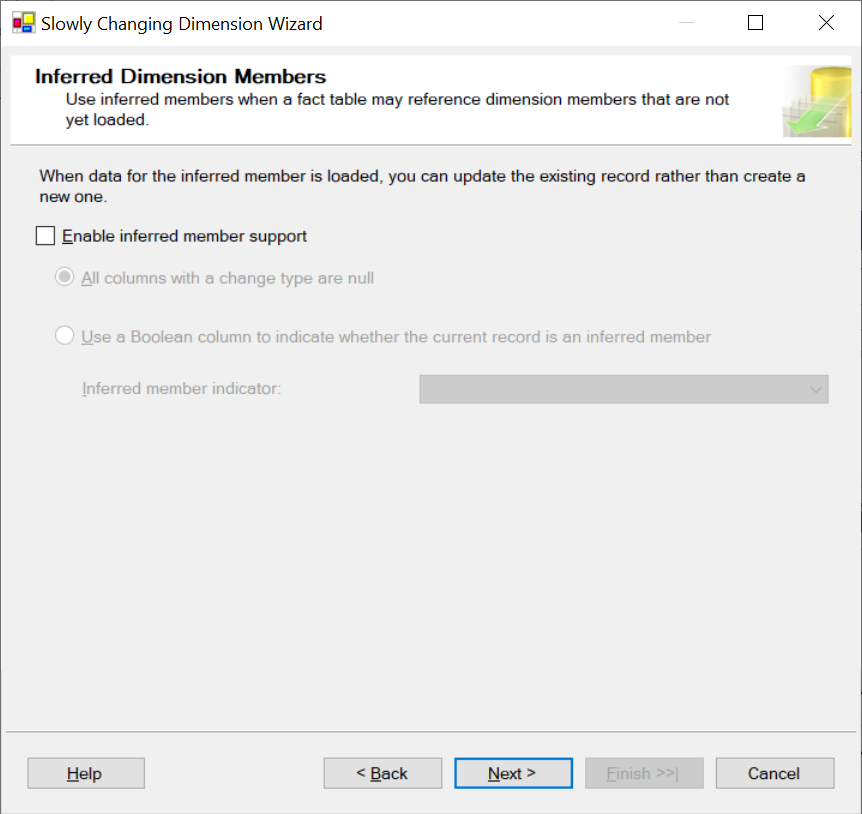
1. Click Next



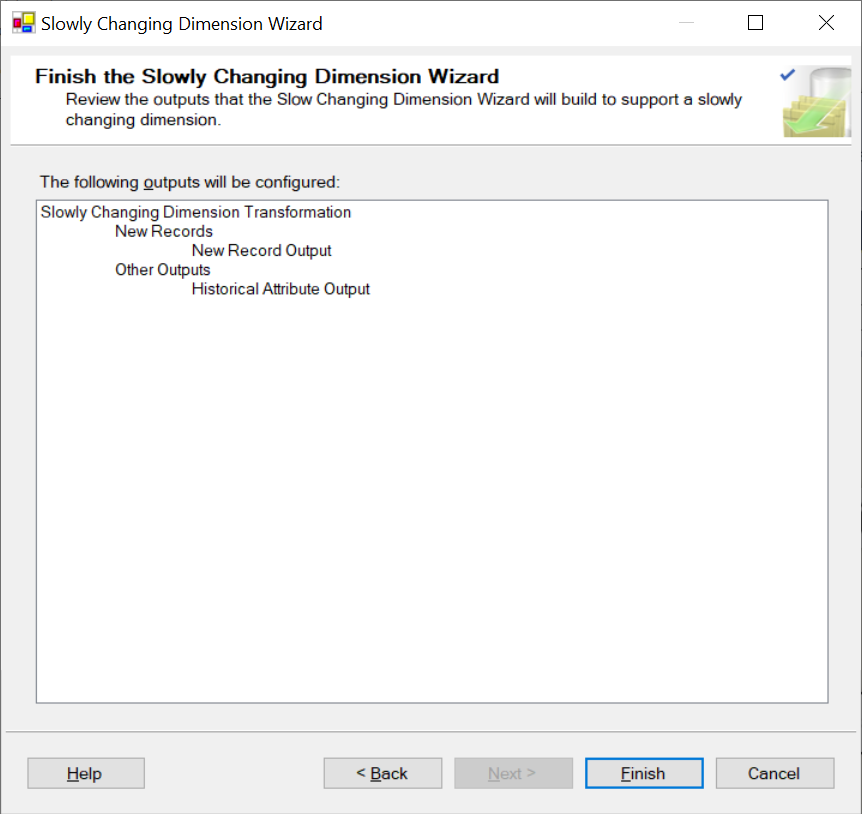
1. Click Next
2. Change the Variable to set data values



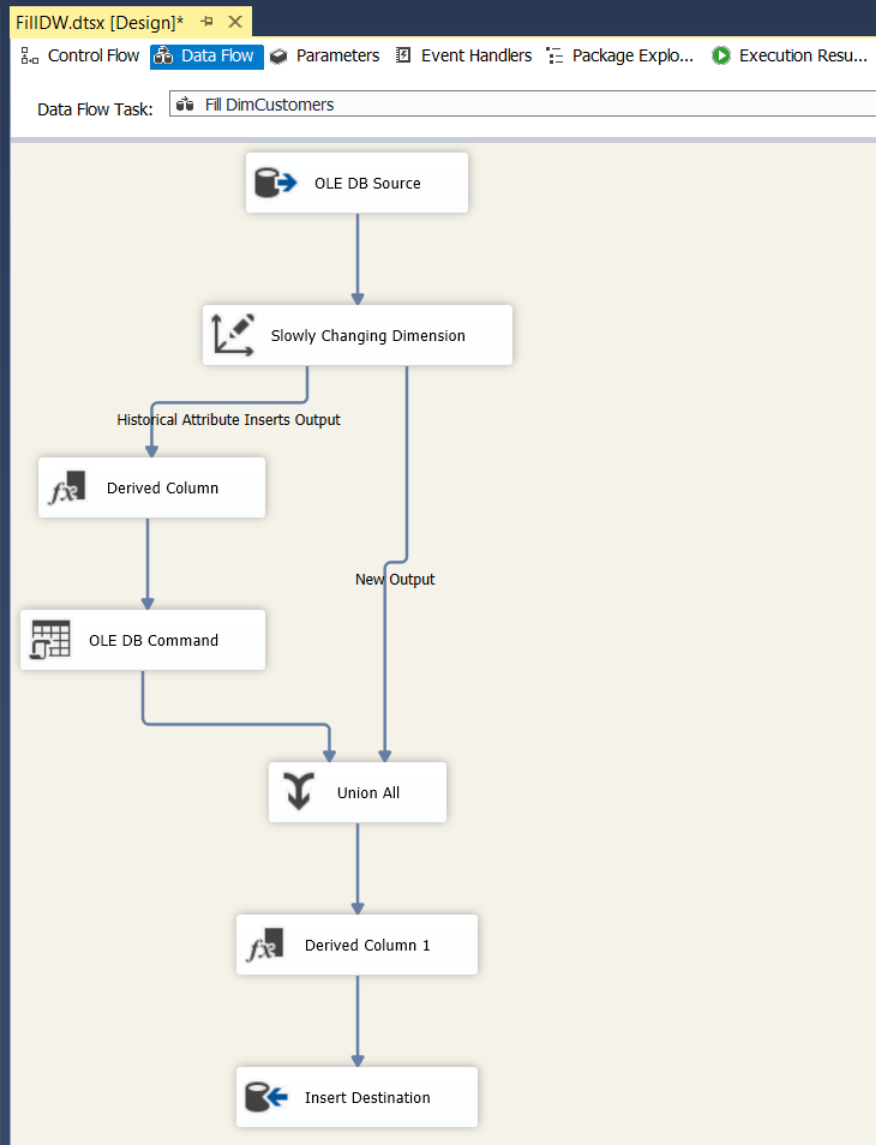
1. Click Next



1. Click Next

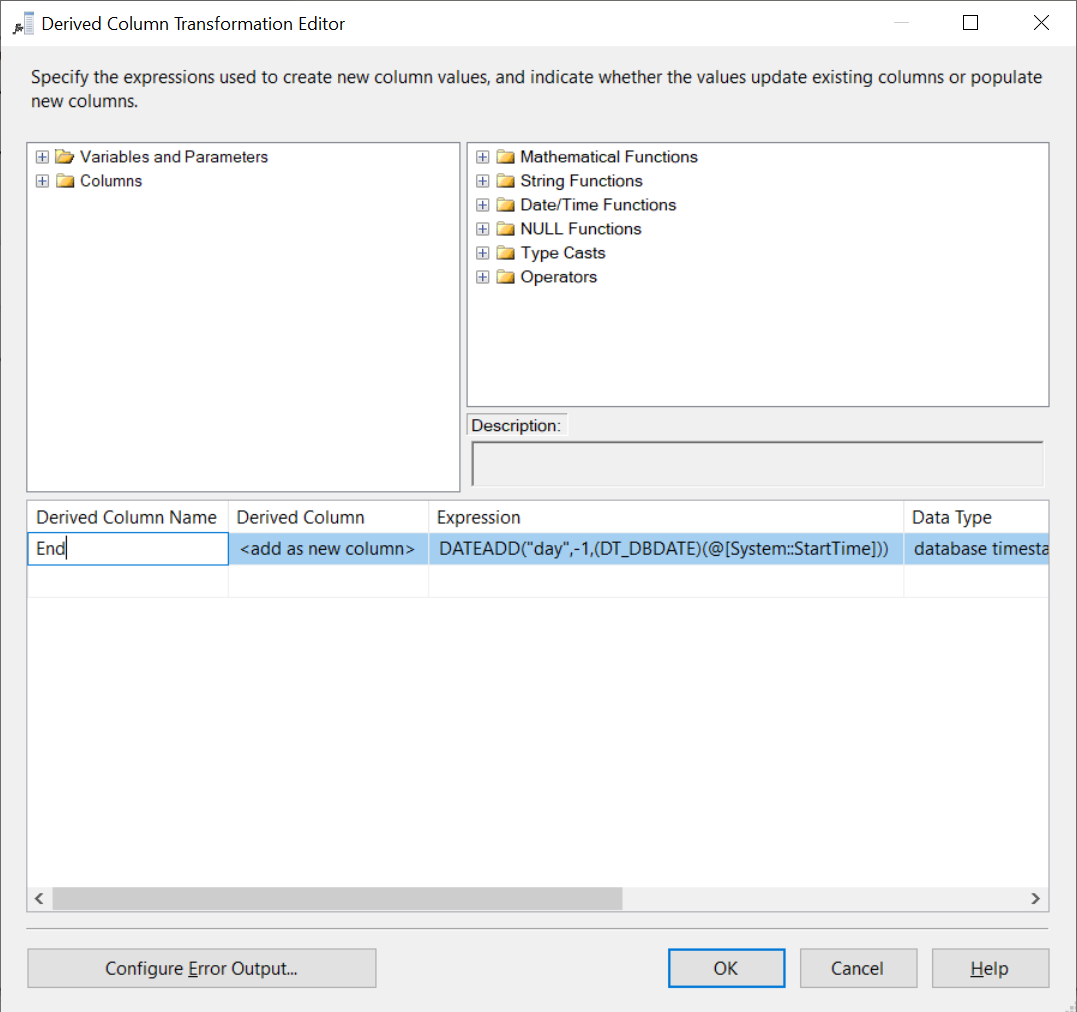


1. Click Finish

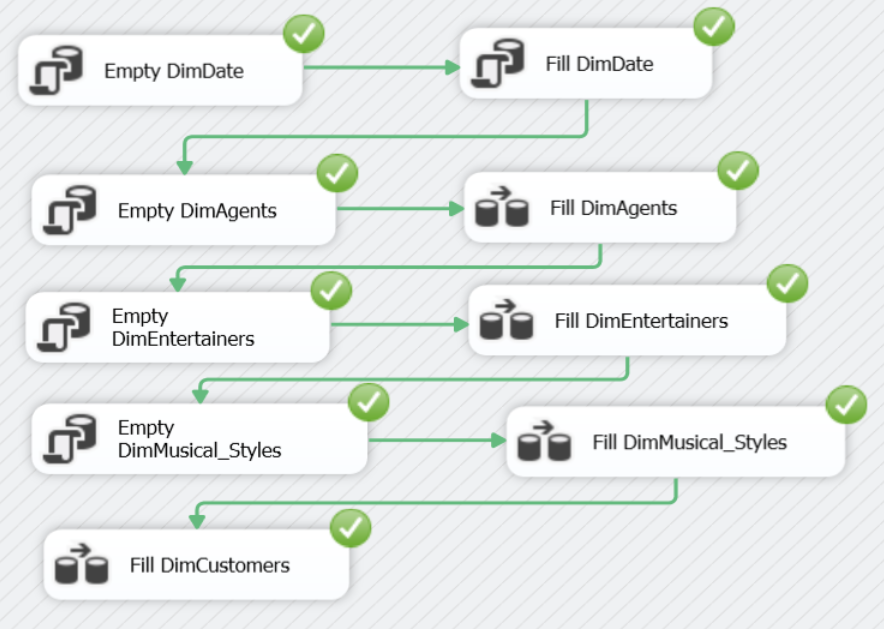


1. By default SSIS sets the End date equal to the new start date. It seems more appropriate to set the end date to yesterday (the day before the SSIS runs) and the new start date to today.
2. Right click on 'Derived Column' and choose Edit… to change the Expression field as follows:

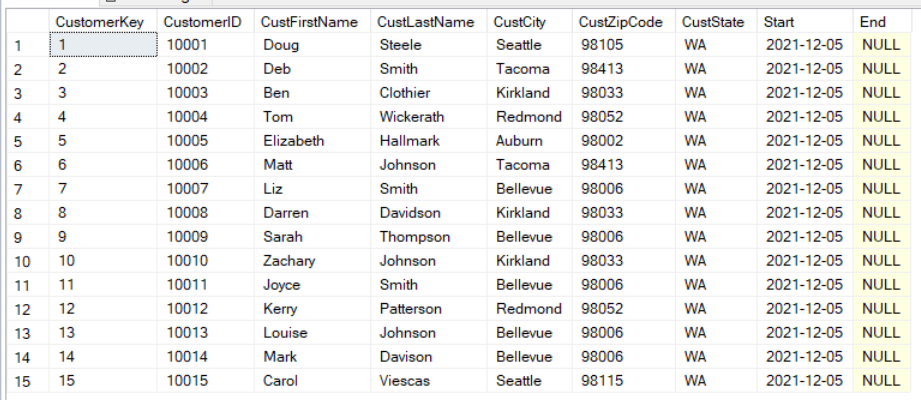
DATEADD("day",-1,(DT\_DBDATE)(@[System::StartTime]))



1. Click OK
2. Go back to the Control Flow area and run the provisional pipeline



1. Go to Microsoft SQL Server Management Studio and check if EntertainmentAgencyDW > DimDate and EntertainmentAgencyDW > DimAgents and EntertainmentAgencyDW > DimEntertainers and EntertainmentAgencyDW > DimMusical\_Styles and EntertainmentAgencyDW > DimCustomers contain the correct data
2. DimCustomers is filled but for Start = today.



1. From a business point of view this is essentially wrong. Current customer data is valid since the start of the company (since we have no history yet). So execute in EntertainmentAgencyDW:

UPDATE DimCustomers set Start =

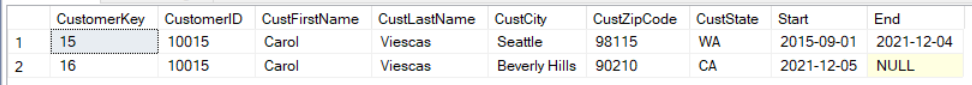
(SELECT MIN(StartDate) from EntertainmentAgency.dbo.Engagements);



1. Check what happens if e.g. the city of a customer is updated and you rerun the package.  
   * In the OLTP database  
     UPDATE Customers   
     SET CustCity = 'Beverly Hills', CustZipCode = '90210', CustState = 'CA'  
     WHERE CustomerID = 10015;
   * Rerun the package
   * In the DWH  
     SELECT \* FROM DimCustomers WHERE CustomerID = 10015;  
     🡪 There are now two lines of this customer.

Notice that the CustomerID is the same for both lines, but the CustomerKey is different for both lines

Notice the Start and End dates => the Customer with CustomerKey = 15 is valid from 2015-09-01 until 2021-12-04 and the customer with CustomerKey = 16 is valid from 2021-12-05  
By CustomerID we know it's the same customer, but the customer moved to another state



1. Fill FactEngagements 🡪 fill fact table
2. This is how the query looked like in AdventureWorksDW. We will have to replace each part to make it work for EntertainmentAgencyDW

INSERT INTO FactSales(SalesOrderLineNumber, ProductKey, SalesTerritoryKey,

OrderDateKey, OrderQuantity, UnitPrice, ExtendedAmount)

SELECT d.SalesOrderDetailID, p.ProductKey, h.TerritoryID, CAST(FORMAT(h.OrderDate,'yyyyMMdd') as int), d.OrderQty, d.UnitPrice,

d.OrderQty \* d.UnitPrice

FROM AdventureWorks2019.Sales.SalesOrderHeader h join

AdventureWorks2019.Sales.SalesOrderDetail d on h.SalesOrderID = d.SalesOrderID

join DimProduct p on d.ProductID = p.ProductID

WHERE

/\* Slowly Changing Dimension dimproduct \*/

h.OrderDate >= p.start and (p.[end] is null or h.orderdate <= p.[end])

AND

/\* only add new lines + make sure it runs from an empty FactSales table \*/

**d.SalesOrderDetailID > (SELECT ISNULL(MAX(SalesOrderLineNumber),0) from FactSales)**

1. Replace the first part of this query to match FactEngagements

INSERT INTO FactSales(SalesOrderLineNumber, ProductKey, SalesTerritoryKey,

OrderDateKey, OrderQuantity, UnitPrice, ExtendedAmount) 🡪

INSERT INTO FactEngagements(EngagementKey, StartDateKey, EndDateKey, NumberOfDays, StartTime, StopTime, NumberOfHours, CustomerKey, AgentKey, EntertainerKey,

Musical\_StyleKey, ContractPrice, CommissionAgent)

1. What becomes the new value for
   * EngamentKey 🡪 … (analogous to what we've used in the query for AdventureWorksDW)
   * StartDateKey 🡪 CAST(format(e.startdate,'yyyyMMdd') as int)
   * EndDateKey 🡪
   * NumberOfDays 🡪 DATEDIFF(…)
   * StartTime 🡪
   * StopTime 🡪
   * NumberOfHours 🡪 DateDIFF(hour, e.StartTime, e.StopTime) + CASE WHEN e.StopTime<e.StartTime THEN 24 ELSE 0 END (make sure you understand this!!)
   * CustomerKey 🡪 … (analogous to what we've used in the query for AdventureWorksDW)
   * AgentKey 🡪 AgentID
   * EntertainerKey 🡪 …
   * Musical\_StyleKey 🡪 …
   * ContractPrice 🡪 …
   * CommissionAgent 🡪 … (contractprice \* commissionRate of the agent)
2. Taking into account the previous step: what becomes the new value for

SELECT d.SalesOrderDetailID, p.ProductKey, h.TerritoryID, CAST(FORMAT(h.OrderDate,'yyyyMMdd') as int), d.OrderQty, d.UnitPrice, d.OrderQty \* d.UnitPrice 🡪

SELECT DISTINCT

…,

CAST(format(e.startdate,'yyyyMMdd') AS int),

…,

DATEDIFF(…),

…,

…,   
DateDIFF(hour, e.StartTime, e.StopTime) + CASE WHEN e.StopTime<e.StartTime THEN 24 ELSE 0 END,

…,

…,

…,

…,

…,

…

1. What becomes the new value for

FROM AdventureWorks2019.Sales.SalesOrderHeader h join

AdventureWorks2019.Sales.SalesOrderDetail d on h.SalesOrderID = d.SalesOrderID

join DimProduct p on d.ProductID = p.ProductID

1. What becomes the new value for

WHERE

/\* Slowly Changing Dimension dimproduct \*/

h.OrderDate >= p.start and (p.[end] is null or h.orderdate <= p.[end])

1. What becomes the new value for

AND

/\* only add new lines + make sure it runs from an empty FactSales table \*/

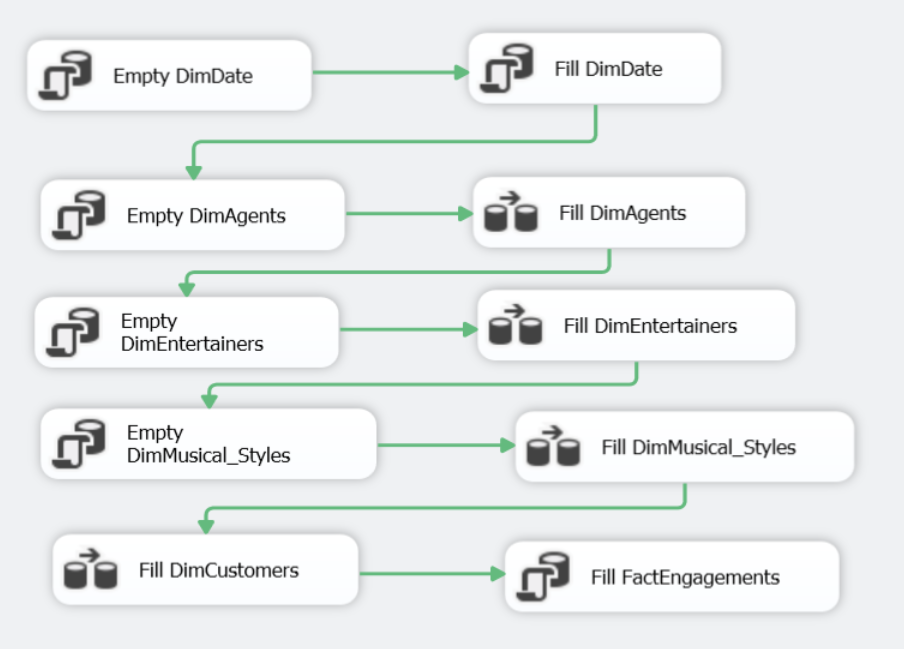
**d.SalesOrderDetailID > (SELECT ISNULL(MAX(SalesOrderLineNumber),0) from FactSales)**

1. Add to the previous WHERE – clause the following part

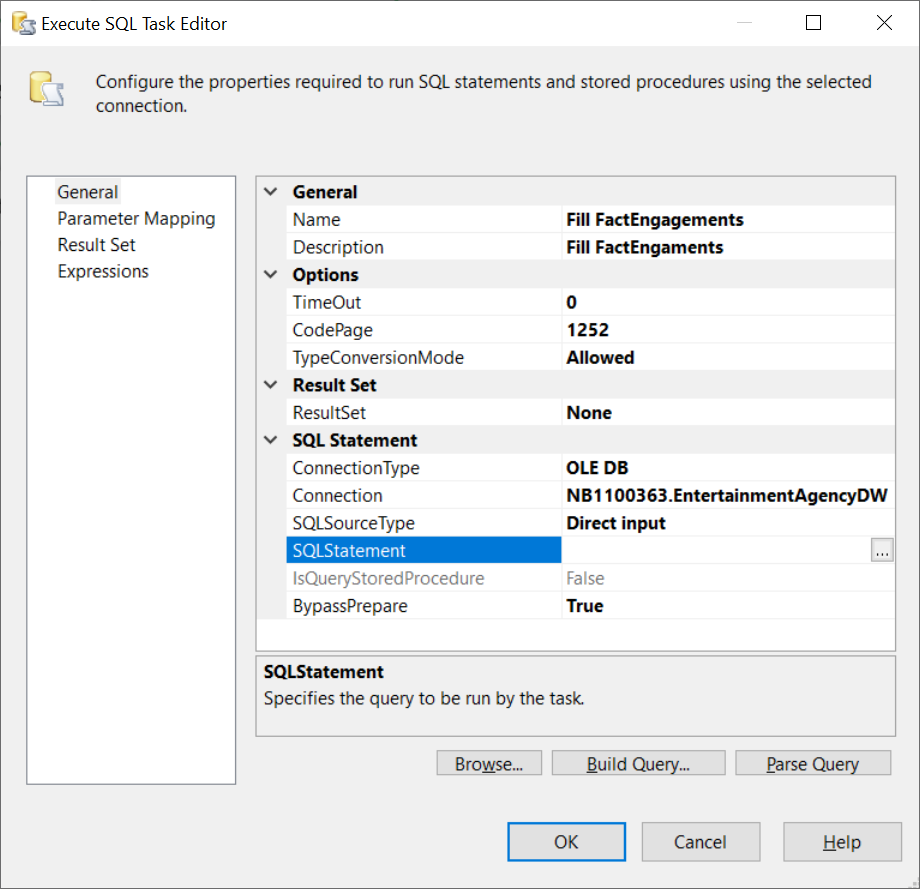
AND /\* pick dominant style \*/

s.StyleStrength = 1

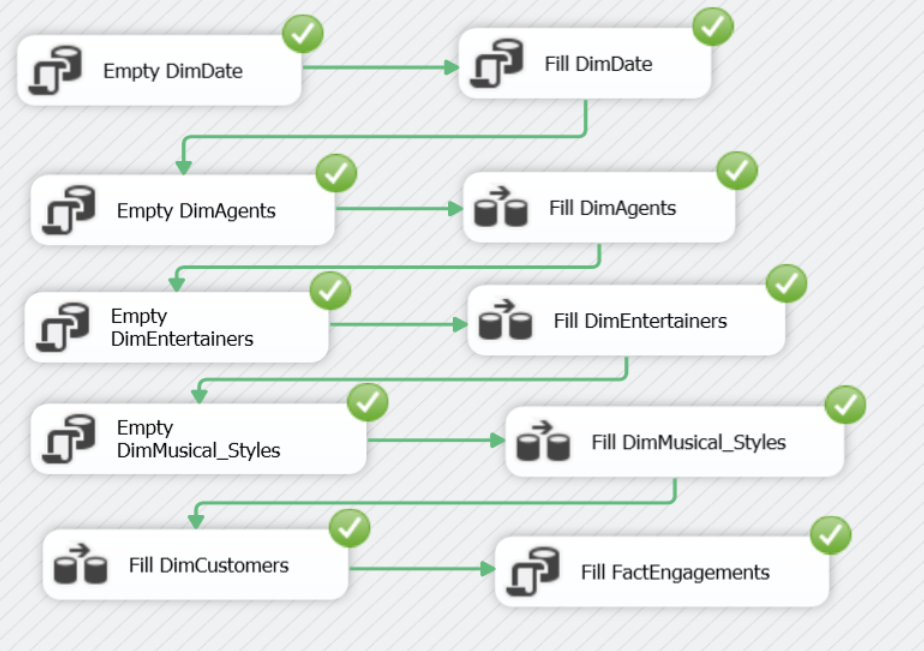
1. Drag and drop 'Execute SQL Task' from the SSIS Toolbox (at the left) to the Control Flow area.
2. Connect to the previous tasks



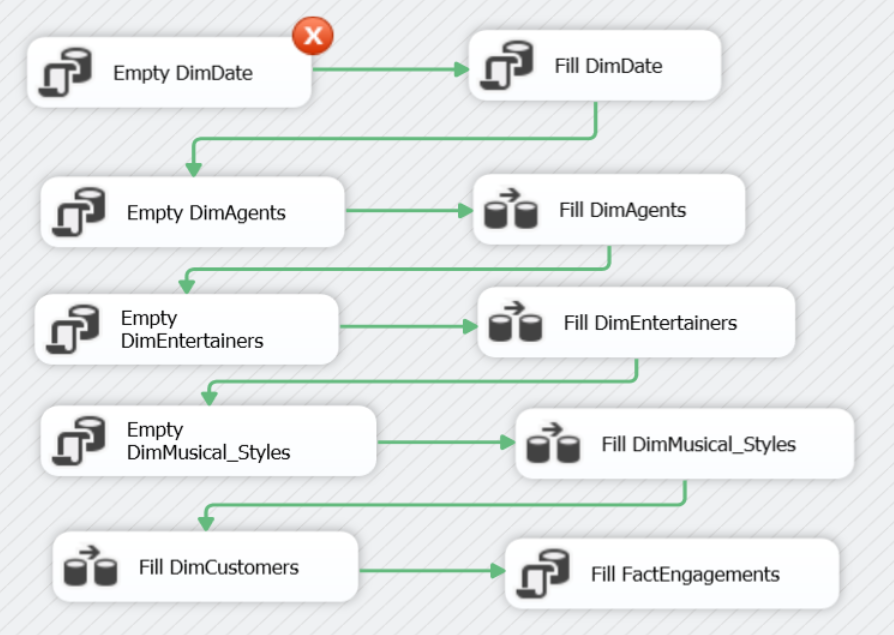
1. Right click on Execute SQL Task and choose Edit… to configure the Execute SQL Task



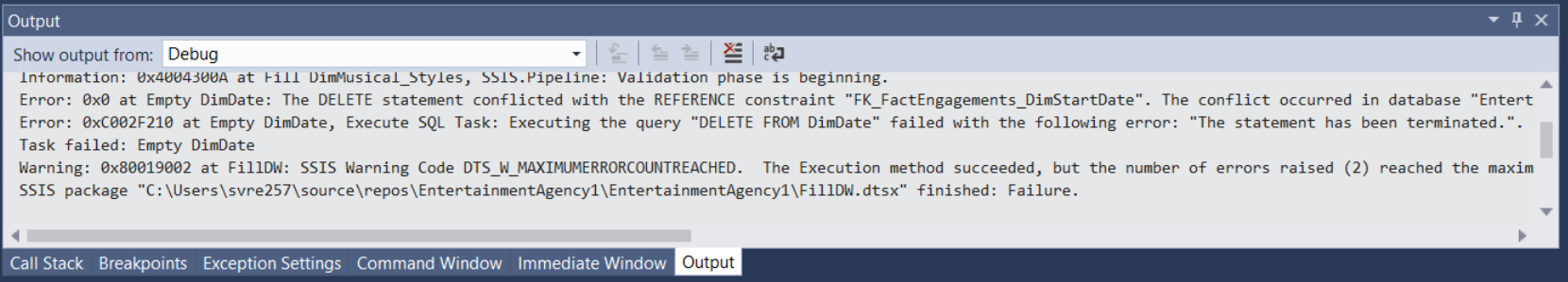
1. Click on the … next tot SQL Statement and copy paste the entire, previous SQL statement
2. Click OK x 2
3. Run the provisional pipeline.



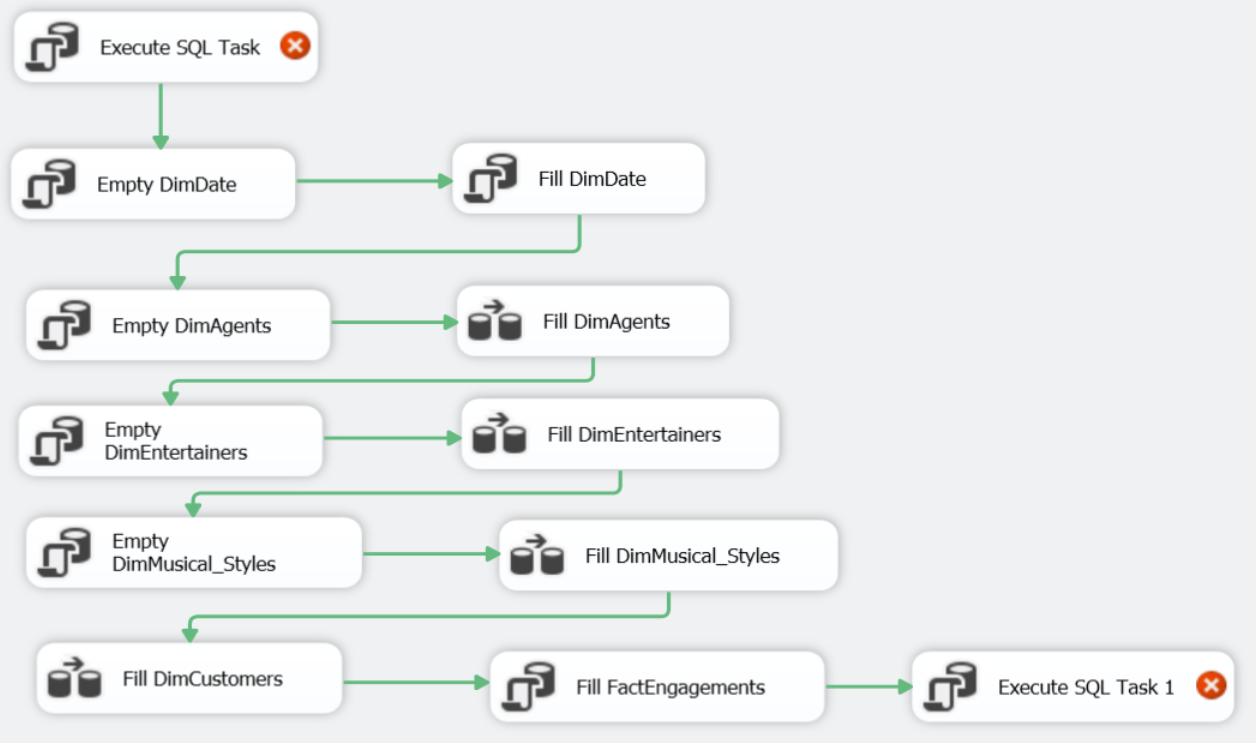
1. Run the provisional pipeline again.



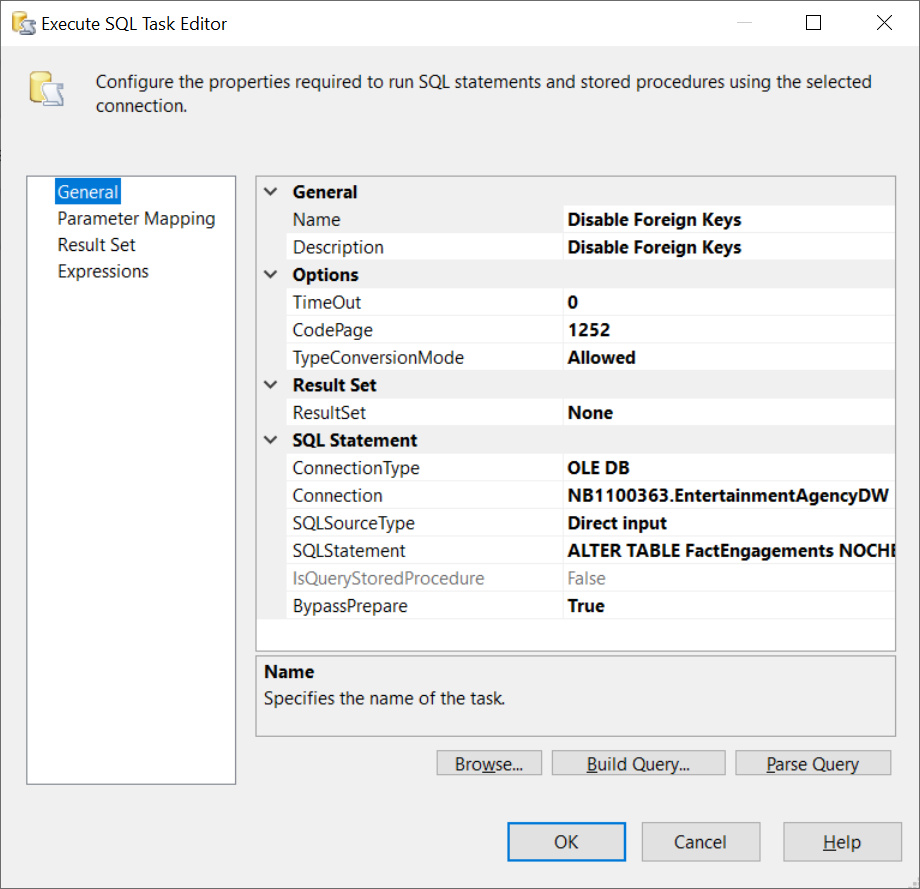
1. We get an error message because we are trying to delete DimDate while there are foreign key constraints between DimDate and FactEngagements. The previous time we ran the pipeline, there wasn't a problem because FactEngagments was still empty

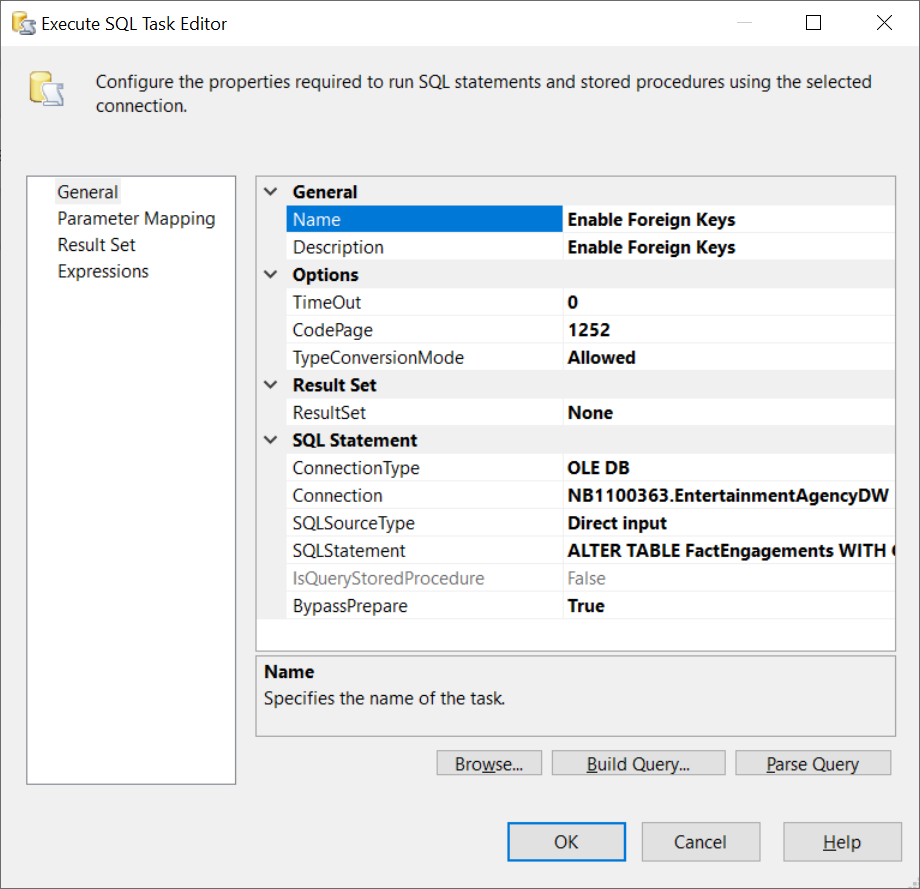


1. One way to cope with this is to temporarily disable the constraints at the start of the fill operation and enable them again at the end.
2. Add two Execute SQL Tasks and connect to the previous tasks.

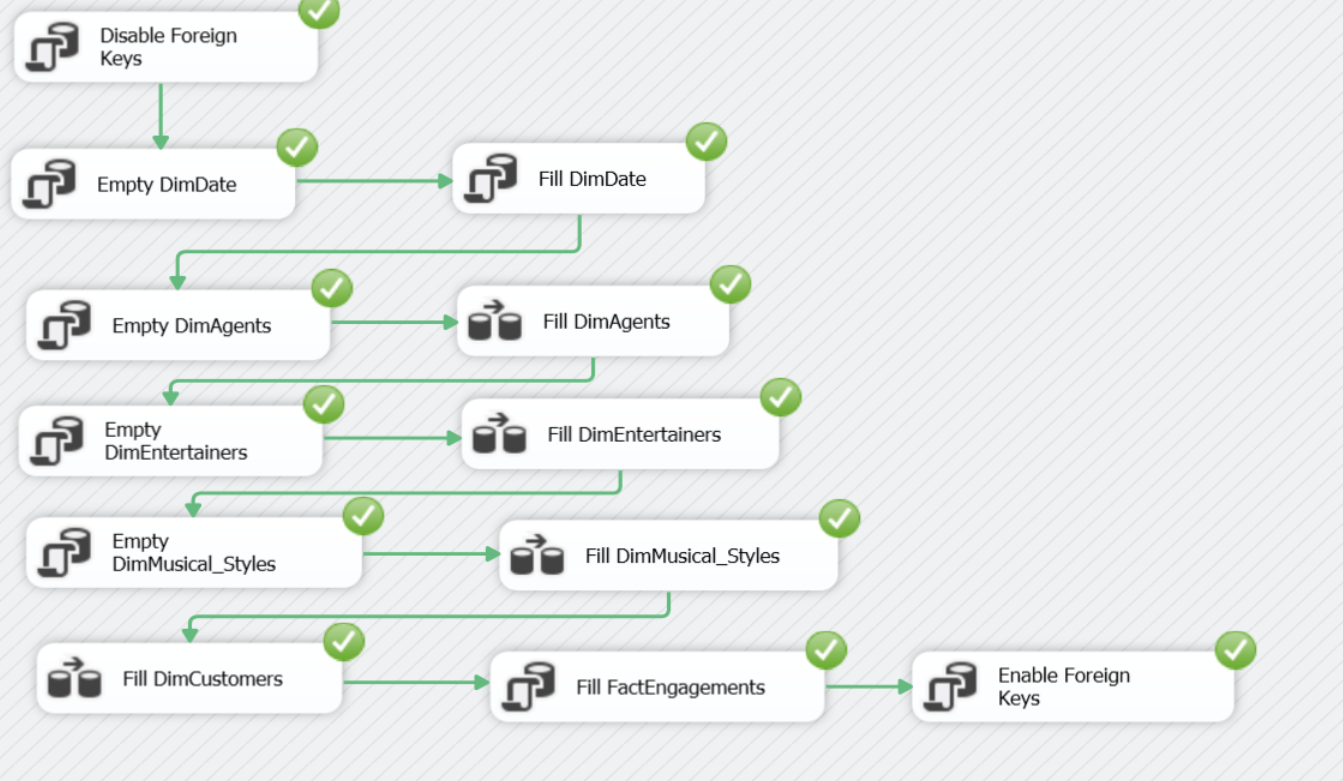


1. Rightclick on both tasks and choose Edit… to configure the tasks
   * At the start of the pipeline: Disable Foreign Keys  
     SQL command = ALTER TABLE FactEngagements NOCHECK CONSTRAINT ALL
   * At the end of the pipeline: Enable Foreign Keys  
     SQL command = ALTER TABLE FactEngagements WITH CHECK CHECK CONSTRAINT ALL; (2 times CHECK!)





1. Run the final pipeline.



1. As a final test we can add a new engagement in the OLTP database EntertainmentAgency for customer 10015.

INSERT INTO Engagements (StartDate, EndDate, StartTime, StopTime, ContractPrice, CustomerID, AgentID, EntertainerID)

VALUES (GETDATE() + 5, GETDATE() + 6, '13:00:00', '16:00:00', 1000, 10015, 4, 1010)

Rerun the package.

Check if the corresponding FactEngagements line in the datawarehouse will be linked to the correct CustomerKey in DimCustomers.

SELECT \*

FROM EntertainmentAgencyDW.dbo.FactEngagements

WHERE ContractPrice = 1000



Remove the record inserted into Engagements in order to get the previous database, otherwise the sample solutions in the extra exercises (Exercises EntertainmentAgency) won't be correct any more.

DELETE FROM Engagements WHERE ContractPrice = 1000

**Part 6 – 6 SQL Queries on EntertainmentAgencyDW**

These are the same queries from the start. Try to solve them now using EntertainmentAgencyDW

-- Query 1

-- What is the average price per musical style based on EntPricePerDay.

-- Only take into account StyleStrength = 1. Order by average price in descending way.

-- Query 2

-- What are the earnings of each agent per quarter

-- Only take into account the commissionrate of the agent and the contractprice

-- Query 3

-- What are the customers that have booked the same entertainer every year (every year = every year bookings are registered).

-- The image below shows only a part of the resultset

-- Query 4

-- We are looking for a Rhythm and Blues group. Which entertainer is the most popular one?

-- Query 5

-- What is the TOP 3 of most booked musical styles

-- Only take into account StyleStrength = 1.

-- Query 6

-- Give for each year the top 3 of most popular entertainers (= entertainers with most engagements for that year)