## Practical 2: Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sqlserver / Power BI.

**Step 1 : Data Extraction :**

The data extraction is first step of ETL. There are 2 Types of Data Extraction

1. Full Extraction : All the data from source systems or operational systems gets extracted to staging area. (Initial Load)

2. Partial Extraction : Sometimes we get notification from the source system to update specific date. It is called as Delta load.

Source System Performance: The Extraction strategies should not affect source system performance.

**Step 2 : Data Transformation :**

The data transformation is second step.After extracting the data there is big need to do the transformation as per the target system.I would like to give you some bullet points of Data Transformation.

- Data Extracted from source system is in to Raw format. We need to transform it before loading in to target server.
- Data has to be cleaned, mapped and transformed

- There are following important steps of Data Transformation :

1.**Selection :** Select data to load in target

2.**Matching :** Match the data with target system

3.**Data Transforming :** We need to change data as per target table structures

**Real life examples of Data Transformation :**

- Standardizing data : Data is fetched from multiple sources so it needs to be standardized as per the target system.
- Character set conversion : Need to transform the character sets as per the target systems. (Firstname and last name example)
- Calculated and derived values: In source system there is first val and second val and in target we need the calculation of first val and second val.
- Data Conversion in different formats : If in source system date in in DDMMYY format and in target the date is in DDMONYYYY format then this transformation needs to be done at transformation phase.

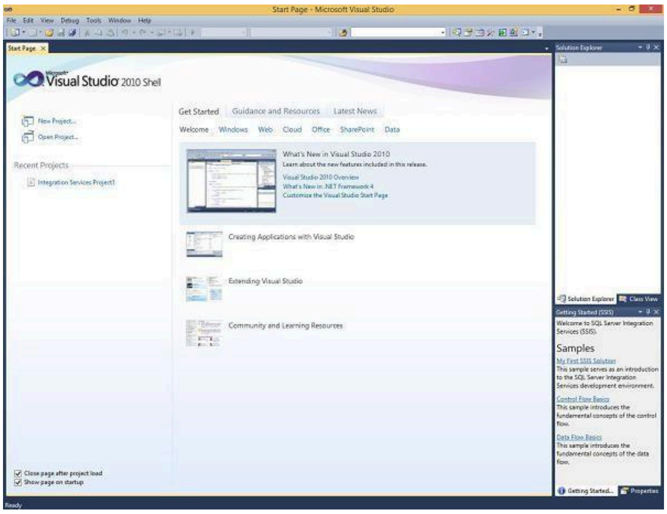
## Step 3 : Data Loading

- Data loading phase loads the prepared data from staging tables to main tables.

ETL process in SQL Server:

Following are the steps to open BIDS\SSDT.

**Step 1** − Open either BIDS\SSDT based on the version from the Microsoft SQL Server programs group. The following screen appears.


Start Page - Microsoft Visual Studio
File Edit View Debug Tools Window Help
Start Page
Microsoft Visual Studio 2010 Shell
New Project...
Open Project...
Recent Projects
Integration Services Project1
Get Started
Guidance and Resources
Latest News
Welcome Windows Web Cloud Office SharePoint Data
What's New in Visual Studio 2010
Visual Studio 2010 Overview
What's New in .NET Framework 4
Customize the Visual Studio Start Page
Creating Applications with Visual Studio
Extending Visual Studio
Community and Learning Resources
☑ Close page after project load
☑ Show page on startup
Solution Explorer
Solution Explorer
Class View
Getting Started (SSIS)
Samples
My First SSIS Solution
Control Flow Basics
Data Flow Basics
Getting Started...
Properties
Ready

Learn about the new features included in this release.

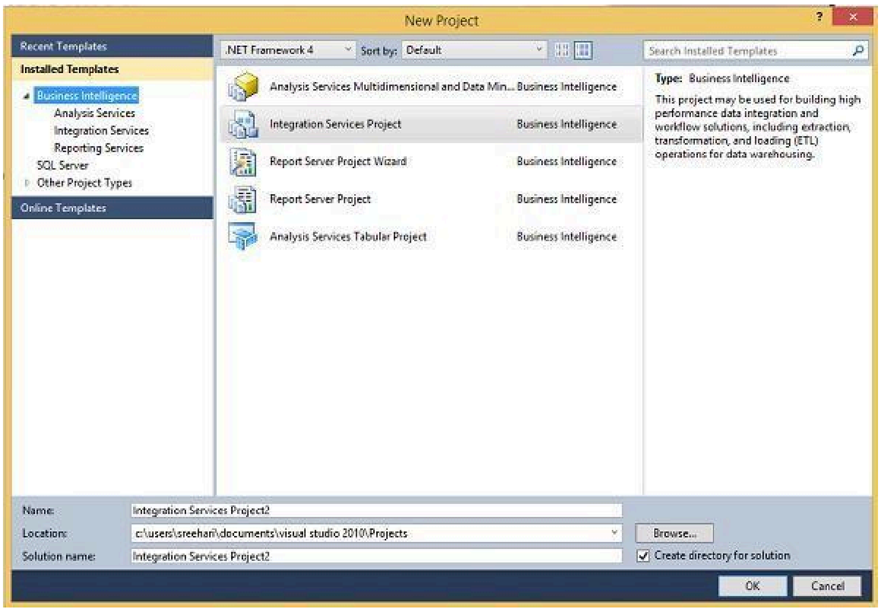
Welcome to SQL Server Integration Services (SSIS).

This sample serves as an introduction to the SQL Server Integration Services development environment.

This sample introduces the fundamental concepts of the control flow.

This sample introduces the fundamental concepts of the data flow.

**Step 2** − The above screen shows SSDT has opened. Go to file at the top left corner in the above image and click New. Select project and the following screen opens.



| | |
|---|---|
| Analysis Services Multidimensional and Data Min... | Business Intelligence |
| Integration Services Project | Business Intelligence |
| Report Server Project Wizard | Business Intelligence |
| Report Server Project | Business Intelligence |
| Analysis Services Tabular Project | Business Intelligence |

Type: Business Intelligence

This project may be used for building high performance data integration and workflow solutions, including extraction, transformation, and loading (ETL) operations for data warehousing.

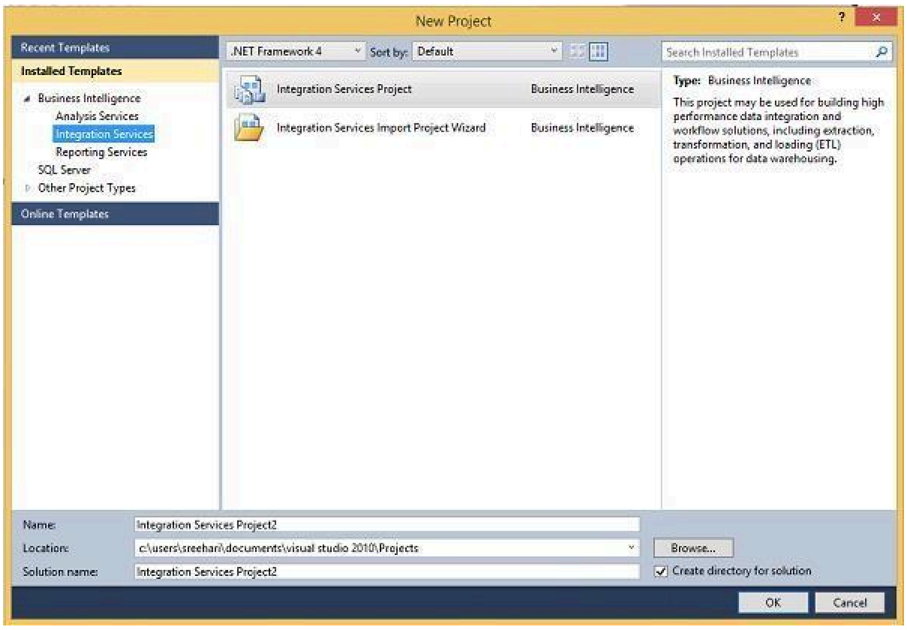
Name: Integration Services Project2

Location: c:\users\sreehari\documents\visual studio 2010\Projects

Solution name: Integration Services Project2

☑ Create directory for solution

**Step 3** − Select Integration Services under Business Intelligence on the top left corner in the above screen to get the following screen.



| | |
|---|---|
| Integration Services Project | Business Intelligence |
| Integration Services Import Project Wizard | Business Intelligence |

Type: Business Intelligence

This project may be used for building high performance data integration and workflow solutions, including extraction, transformation, and loading (ETL) operations for data warehousing.

Name: Integration Services Project2

Location: c:\users\sreehari\documents\visual studio 2010\Projects

Solution name: Integration Services Project2

☑ Create directory for solution

**Step 4** − In the above screen, select either Integration Services Project or Integration Services Import Project Wizard based on your requirement to develop\create the package.

## Modes

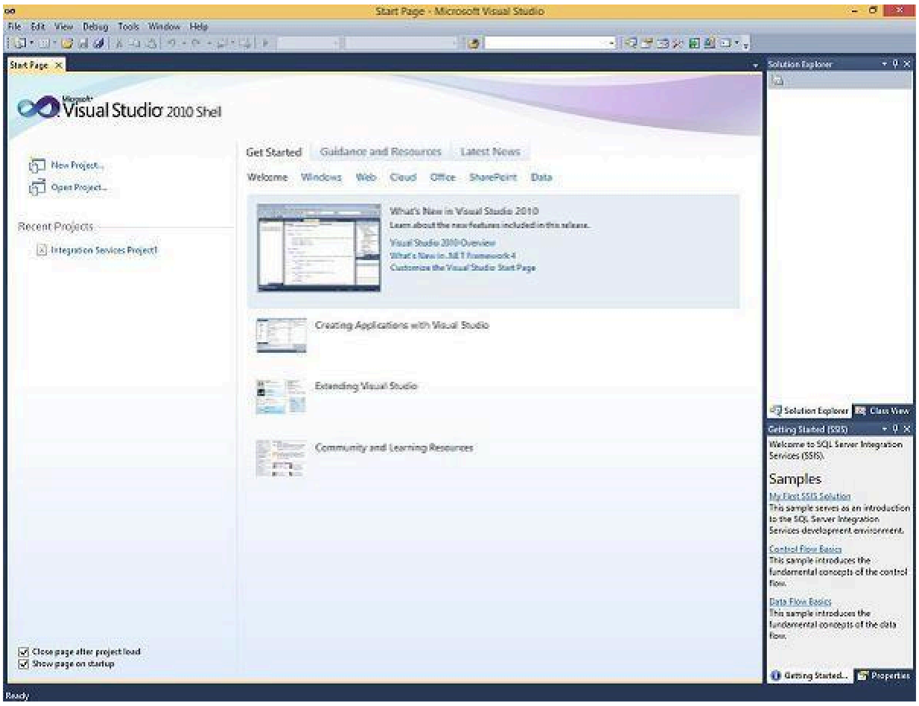
There are two modes − Native Mode (SQL Server Mode) and Share Point Mode.

## Models

There are two models − Tabular Model (For Team and Personal Analysis) and Multi Dimensions Model (For Corporate Analysis).

The BIDS (Business Intelligence Studio till 2008 R2) and SSDT (SQL Server Data Tools from 2012) are environments to work with SSAS.

**Step 1** − Open either BIDS\SSDT based on the version from the Microsoft SQL Server programs group. The following screen will appear.



Start Page - Microsoft Visual Studio
File Edit View Debug Tools Window Help
Start Page
Visual Studio 2010 Shell
New Project...
Open Project...
Recent Projects
Integration Services Project1
Get Started
Guidance and Resources
Latest News
Welcome Windows Web Cloud Office SharePoint Data
What's New in Visual Studio 2010
Visual Studio 2010 Overview
What's New in .NET Framework 4
Customize the Visual Studio Start Page
Creating Applications with Visual Studio
Extending Visual Studio
Community and Learning Resources
Solution Explorer
Class View
Getting Started (SSIS)
Samples
My First SSIS Solution
Control Flow Basics
Data Flow Basics
Getting Started...
Properties
Ready

Learn about the new features included in this release.

☑ Close page after project load
☑ Show page on startup

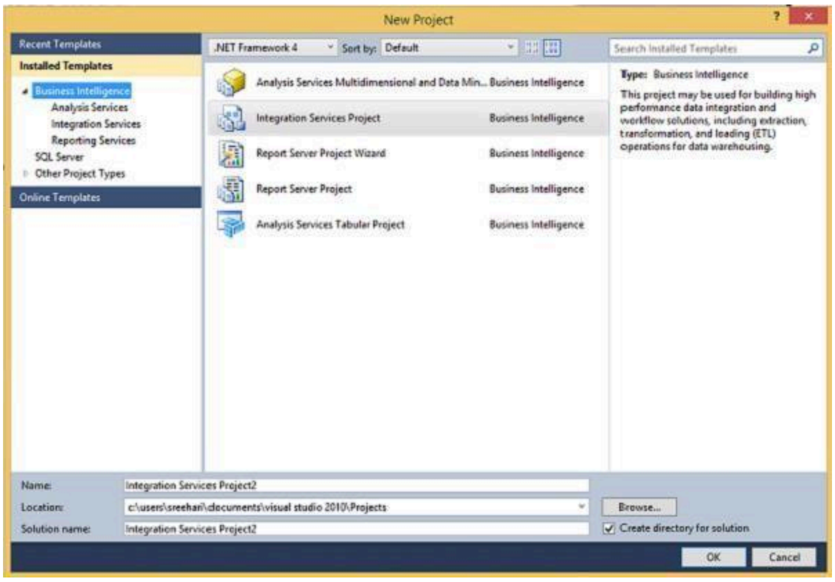
Welcome to SQL Server Integration Services (SSIS).

My First SSIS Solution: This sample serves as an introduction to the SQL Server Integration Services development environment.

Control Flow Basics: This sample introduces the fundamental concepts of the control flow.

Data Flow Basics: This sample introduces the fundamental concepts of the data flow.

**Step 2** − The above screen shows SSDT has opened. Go to file on the top left corner in the above image and click New. Select project and the following screen opens.



.NET Framework 4 · Sort by: Default

| | |
|---|---|
| Analysis Services Multidimensional and Data Min... | Business Intelligence |
| Integration Services Project | Business Intelligence |
| Report Server Project Wizard | Business Intelligence |
| Report Server Project | Business Intelligence |
| Analysis Services Tabular Project | Business Intelligence |

Type: Business Intelligence

This project may be used for building high performance data integration and workflow solutions, including extraction, transformation, and loading (ETL) operations for data warehousing.

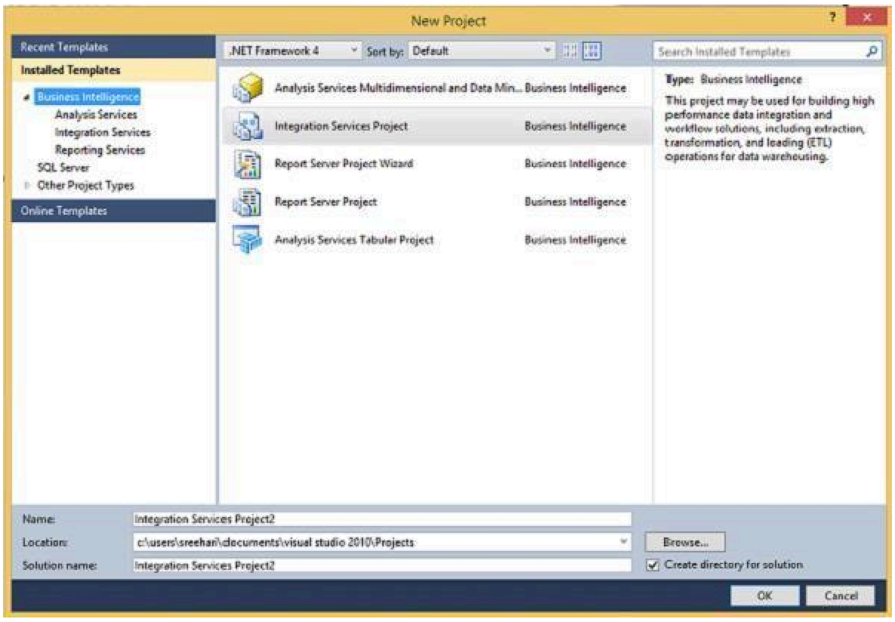
Name: Integration Services Project2

Location: c:\users\sreehari\documents\visual studio 2010\Projects

Solution name: Integration Services Project2

☑ Create directory for solution

**Step 3** − Select Analysis Services in the above screen under Business Intelligence as seen on the top left corner. The following screen pops up.



| | |
|---|---|
| Analysis Services Multidimensional and Data Min... | Business Intelligence |
| Integration Services Project | Business Intelligence |
| Report Server Project Wizard | Business Intelligence |
| Report Server Project | Business Intelligence |
| Analysis Services Tabular Project | Business Intelligence |

Type: Business Intelligence

This project may be used for building high performance data integration and workflow solutions, including extraction, transformation, and loading (ETL) operations for data warehousing.

Name: Integration Services Project2

Location: c:\users\sreehari\documents\visual studio 2010\Projects

Solution name: Integration Services Project2

☑ Create directory for solution

**Step 4** − In the above screen, select any one option from the listed five options based on your requirement to work with Analysis services.

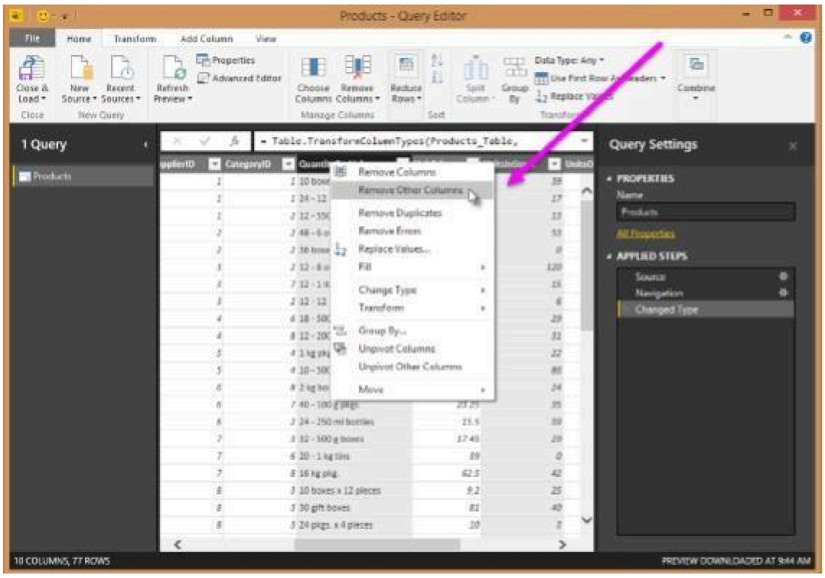
## ETL Process in Power BI

### 1) Remove other columns to only display columns of interest

In this step you remove all columns except **ProductID**, **ProductName**, **UnitsInStock**, and **QuantityPerUnit**

Power BI Desktop includes Query Editor, which is where you shape and transform your data connections. Query Editor opens automatically when you select **Edit** from Navigator. You can also open the Query Editor by selecting Edit Queries from the Home ribbon in Power BI Desktop. The following steps are performed in Query Editor.

1. In **Query Editor**, select the **ProductID, ProductName**, **QuantityPerUnit,** and **UnitsInStock** columns (use **Ctrl+Click** to select more than one column, or **Shift+Click** to select columns that are beside each other).
2. Select **Remove Columns** > **Remove** Other Columns from the ribbon, or right-click on a column header and click Remove Other Columns.



```
= Table.TransformColumnTypes(Products_Table,
```

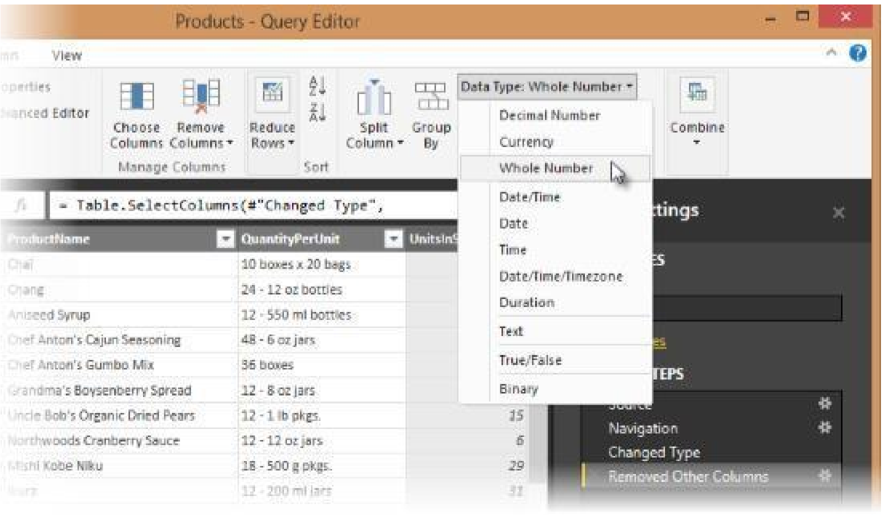
### 3. Change the data type of the UnitsInStock column

When Query Editor connects to data, it reviews each field and to determine the best data type. For the Excel workbook, products in stock will always be a whole number, so in this step you confirm the **UnitsInStock** column's datatype is Whole Number.

1. Select the **UnitsInStock** column.

2. Select the **Data Type drop-down button** in the **Home ribbon**.

3. If not already a Whole Number, select **Whole Number** for data type from the drop down (the Data Type: button also displays the data type for the current selection).



```
= Table.SelectColumns(#"Changed Type",
```

| ProductName | QuantityPerUnit | UnitsIn |
|---|---|---|
| Chai | 10 boxes x 20 bags | |
| Chang | 24 - 12 oz bottles | |
| Aniseed Syrup | 12 - 550 ml bottles | |
| Chef Anton's Cajun Seasoning | 48 - 6 oz jars | |
| Chef Anton's Gumbo Mix | 36 boxes | |
| Grandma's Boysenberry Spread | 12 - 8 oz jars | |
| Uncle Bob's Organic Dried Pears | 12 - 1 lb pkgs. | 15 |
| Northwoods Cranberry Sauce | 12 - 12 oz jars | 6 |
| Mishi Kobe Niku | 18 - 500 g pkgs. | 29 |
| Ikura | 12 - 200 ml jars | 31 |

### 3. **Expand the Order_Details table**

The Orders table contains a reference to a Details table, which contains the individual products that were included in each Order. When you connect to data sources with multiples tables (such as a relational database) you can use these references to build up your query

In this step, you expand the **Order_Details** table that is related to the Orders table, to combine the **ProductID**, **UnitPrice**, and **Quantity** columns from **Order_Details** into the **Orders table**. This is a representation of the data in these tables:

The Expand operation combines columns from a related table into a subject table. When the query runs, rows from the related table (**Order_Details**) are combined into rows from the subject table (**Orders**).

After you expand the Order_Details table, three new columns and additional rows are added to the Orders table, one for each row in the nested or related table.

1. In the Query View, scroll to the Order_Details column.

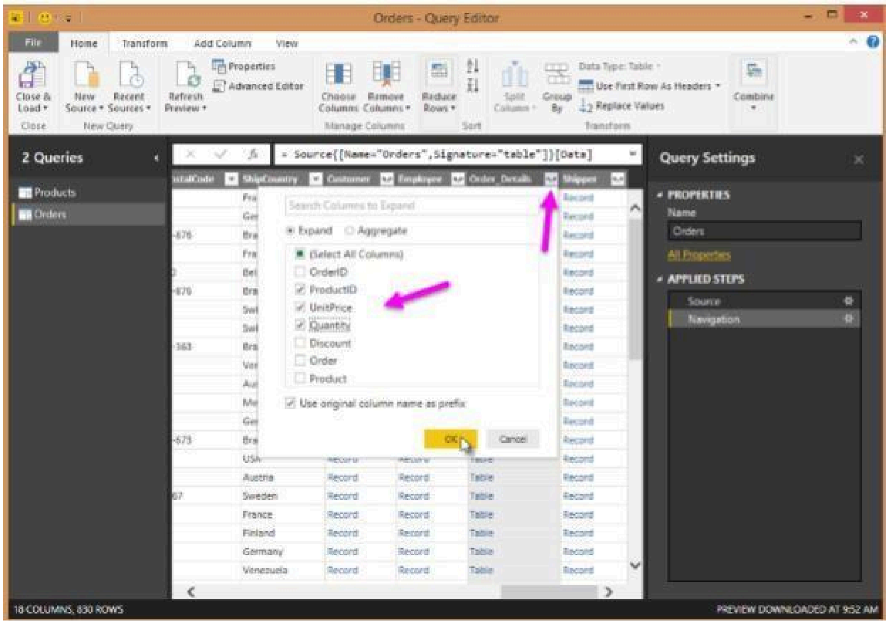
2. In the Order_Details column, select the expand icon ( ).

3. In the Expand drop-down:

   a. Select (Select All Columns) to clear all columns.

   b. Select ProductID, UnitPrice, and Quantity.

   c. Click OK.



= Source{[Name="Orders",Signature="table"]}[Data]

○ Search Columns to Expand

● Expand ○ Aggregate

■ (Select All Columns)
□ OrderID
☑ ProductID
☑ UnitPrice
☑ Quantity
□ Discount
□ Order
□ Product

☑ Use original column name as prefix

Name: Orders

APPLIED STEPS: Source, Navigation

18 COLUMNS, 830 ROWS

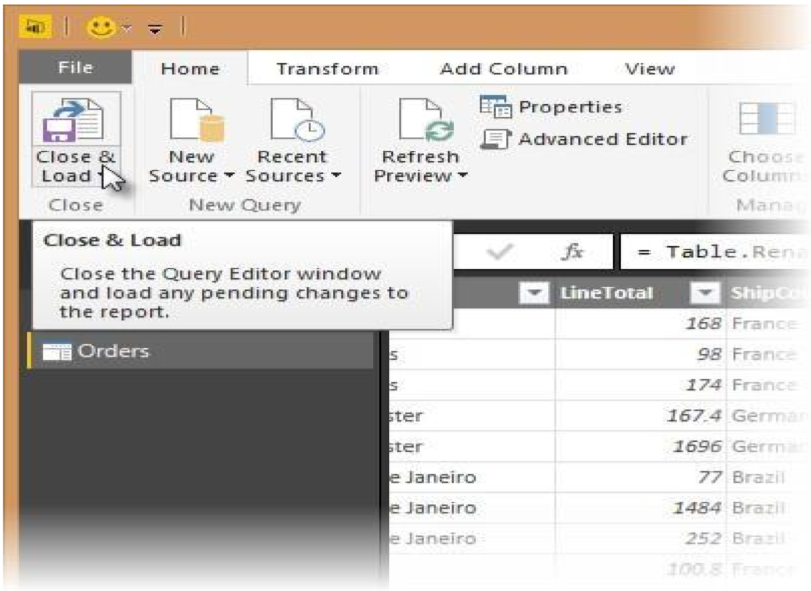
PREVIEW DOWNLOADED AT 9:52 AM

## 4. Calculate the line total for each Order_Details row

Power BI Desktop lets you to create calculations based on the columns you are importing, so you can enrich the data that you connect to. In this step, you create a Custom Column to calculate the line total for each Order_Details row.

Calculate the line total for each Order_Details row:

1. In the Add Column ribbon tab, click Add Custom Column.



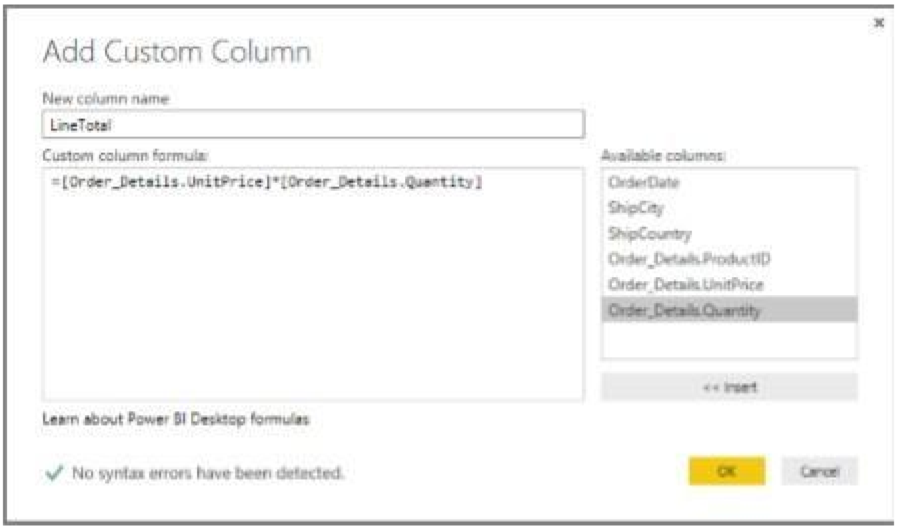
Close & Load

Close the Query Editor window and load any pending changes to the report.

= Table.Ren

| | LineTotal | ShipCo |
|---|---|---|
| | 168 | France |
| s | 98 | France |
| s | 174 | France |
| ster | 167.4 | German |
| ster | 1696 | German |
| e Janeiro | 77 | Brazil |
| e Janeiro | 1484 | Brazil |
| e Janeiro | 252 | Brazil |
| | 100.8 | France |

2. In the Add Custom Column dialog box, in the Custom Column Formula textbox, enter [Order_Details.UnitPrice] * [Order_Details.Quantity].
3. In the New column name textbox, enter LineTotal.
4. Click OK.



Add Custom Column

New column name: LineTotal

Custom column formula:

```
=[Order_Details.UnitPrice]*[Order_Details.Quantity]
```

Available columns:

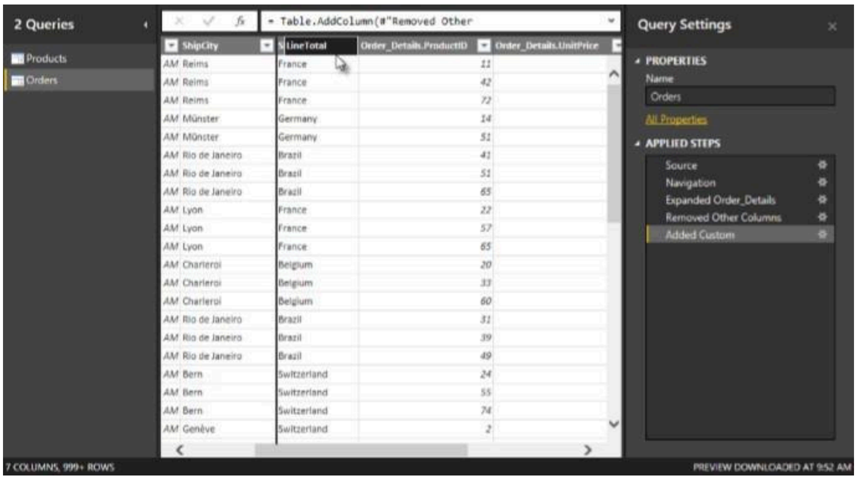
- OrderDate
- ShipCity
- ShipCountry
- Order_Details.ProductID
- Order_Details.UnitPrice
- Order_Details.Quantity

✓ No syntax errors have been detected.

## 5. Rename and reorder columns in the query

In this step you finish making the model easy to work with when creating reports, by renaming the final columns and changing their order.

1. In Query Editor, drag the LineTotal column to the left, after ShipCountry.



```
= Table.AddColumn(#"Removed Other
```

| ShipCity | LineTotal | Order_Details.ProductID | Order_Details.UnitPrice |
|---|---|---|---|
| AM Reims | France | 11 | |
| AM Reims | France | 42 | |
| AM Reims | France | 72 | |
| AM Münster | Germany | 14 | |
| AM Münster | Germany | 51 | |
| AM Rio de Janeiro | Brazil | 41 | |
| AM Rio de Janeiro | Brazil | 51 | |
| AM Rio de Janeiro | Brazil | 65 | |
| AM Lyon | France | 22 | |
| AM Lyon | France | 57 | |
| AM Lyon | France | 65 | |
| AM Charleroi | Belgium | 20 | |
| AM Charleroi | Belgium | 33 | |
| AM Charleroi | Belgium | 60 | |
| AM Rio de Janeiro | Brazil | 31 | |
| AM Rio de Janeiro | Brazil | 39 | |
| AM Rio de Janeiro | Brazil | 49 | |
| AM Bern | Switzerland | 24 | |
| AM Bern | Switzerland | 55 | |
| AM Bern | Switzerland | 74 | |
| AM Genève | Switzerland | 2 | |

Name: Orders

APPLIED STEPS: Source, Navigation, Expanded Order_Details, Removed Other Columns, Added Custom

7 COLUMNS, 999+ ROWS

PREVIEW DOWNLOADED AT 9:52 AM

2. Remove the Order_Details. prefix from the Order_Details.ProductID, Order_Details.UnitPrice and Order_Details.Quantity columns, by double-clicking on each column header, and then deleting that text from the column name.

## 6. Combine the Products and Total Sales queries

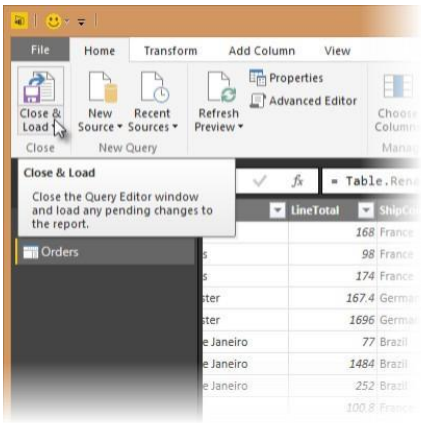
Power BI Desktop does not require you to combine queries to report on them. Instead, you can create Relationships between datasets. These relationships can be created on any column that is common to your datasets

we have Orders and Products data that share a common 'ProductID' field, so we need to ensure there's a relationship between them in the model we're using with Power BI Desktop. Simply specify in Power BI Desktop that the columns from each table are related (i.e. columns that have the same values). Power BI Desktop works out the direction and cardinality of the relationship for you. In some cases, it will even detect the relationships automatically.

In this task, you confirm that a relationship is established in Power BI Desktop between the Products and Total Sales queries

Step 1: Confirm the relationship between Products and Total Sales

1. First, we need to load the model that we created in Query Editor into Power BI Desktop. From the Home ribbon of Query Editor, select Close & Load.



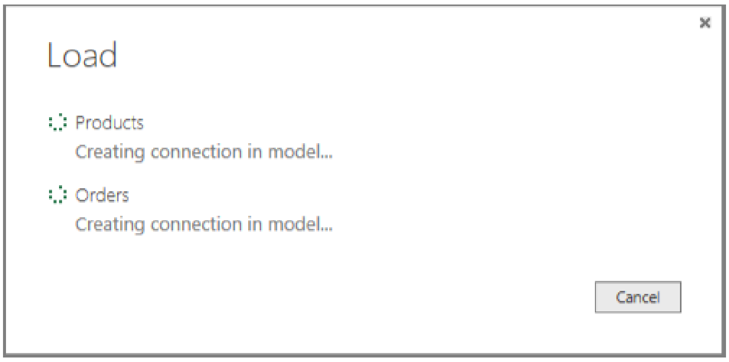
Close & Load

Close the Query Editor window and load any pending changes to the report.

```
= Table.Ren
```

| | LineTotal | ShipCi |
|---|---|---|
| | 168 | France |
| s | 98 | France |
| s | 174 | France |
| ter | 167.4 | Germa |
| ter | 1696 | Germa |
| e Janeiro | 77 | Brazil |
| e Janeiro | 1484 | Brazil |
| e Janeiro | 252 | Brazil |
| | 100.8 | France |

2. Power BI Desktop loads the data from the two queries.

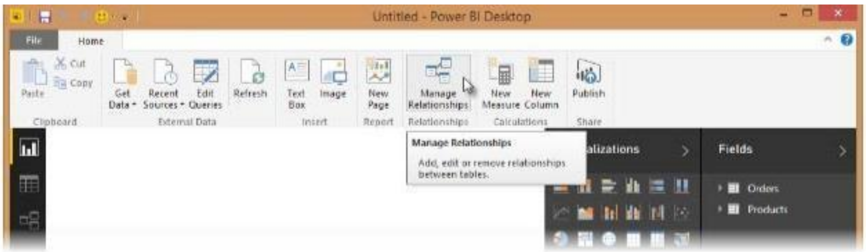


Load

Products
Creating connection in model...

Orders
Creating connection in model...

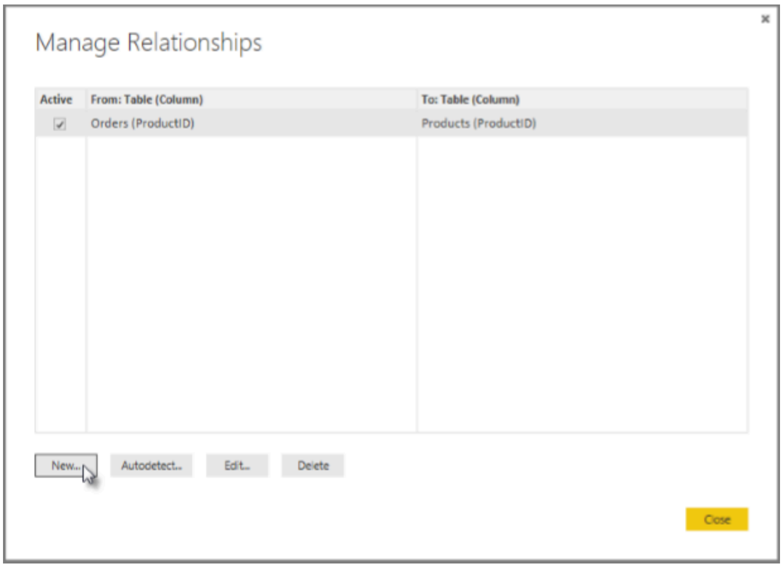
3. Once the data is loaded, select the Manage Relationships button Home ribbon.



Manage Relationships

Add, edit or remove relationships between tables.

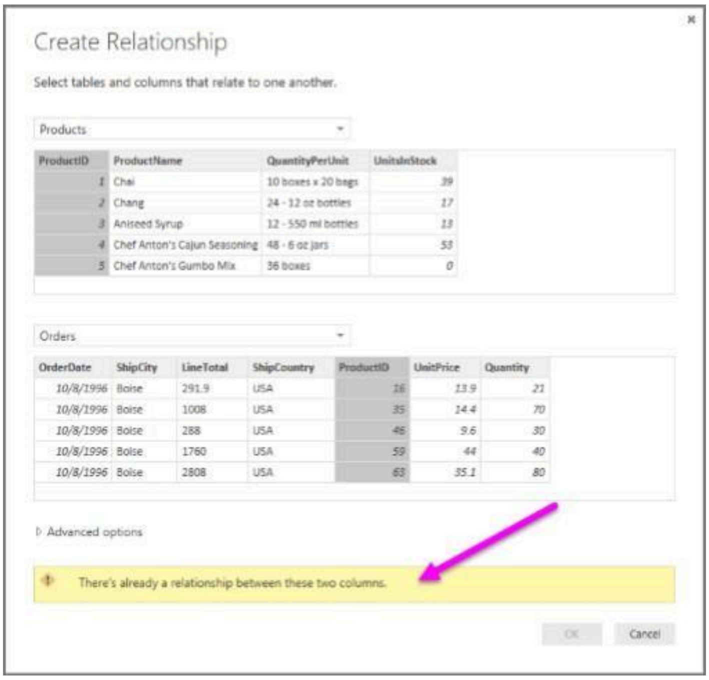
4. Select the New… button



Manage Relationships

| Active | From: Table (Column) | To: Table (Column) |
|---|---|---|
| ☑ | Orders (ProductID) | Products (ProductID) |

5. When we attempt to create the relationship, we see that one already exists! As shown in the Create Relationship dialog (by the shaded columns), the ProductsID fields in each query already have an established relationship.



Create Relationship

Select tables and columns that relate to one another.

Products

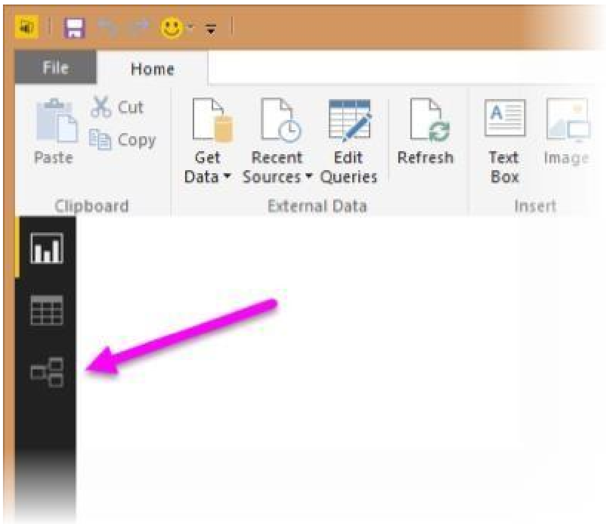
| ProductID | ProductName | QuantityPerUnit | UnitsInStock |
|---|---|---|---|
| 1 | Chai | 10 boxes x 20 bags | 39 |
| 2 | Chang | 24 - 12 oz bottles | 17 |
| 3 | Aniseed Syrup | 12 - 550 ml bottles | 13 |
| 4 | Chef Anton's Cajun Seasoning | 48 - 6 oz jars | 53 |
| 5 | Chef Anton's Gumbo Mix | 36 boxes | 0 |

Orders

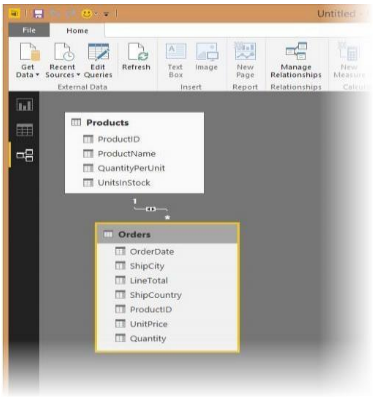
| OrderDate | ShipCity | LineTotal | ShipCountry | ProductID | UnitPrice | Quantity |
|---|---|---|---|---|---|---|
| 10/8/1996 | Boise | 291.9 | USA | 16 | 13.9 | 21 |
| 10/8/1996 | Boise | 1008 | USA | 35 | 14.4 | 70 |
| 10/8/1996 | Boise | 288 | USA | 46 | 9.6 | 30 |
| 10/8/1996 | Boise | 1760 | USA | 59 | 44 | 40 |
| 10/8/1996 | Boise | 2808 | USA | 63 | 35.1 | 80 |

There's already a relationship between these two columns.

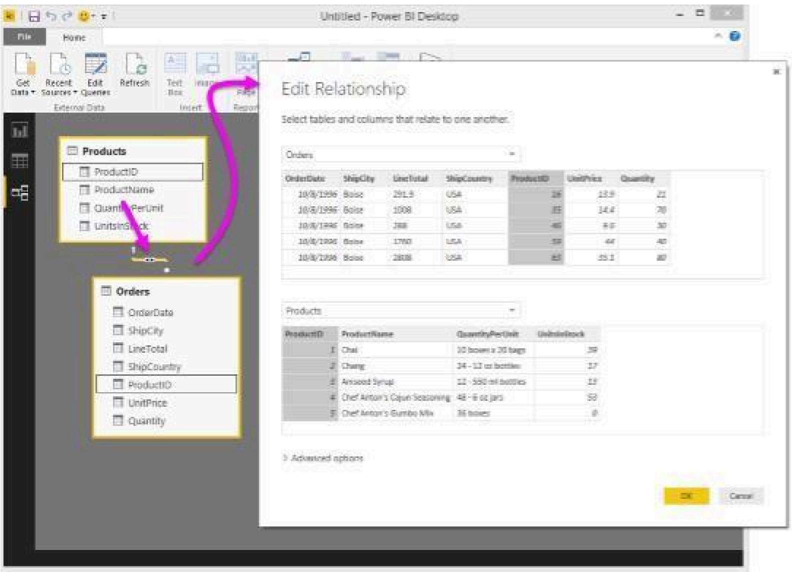
5. Select Cancel, and then select Relationship view in Power BI Desktop.



6. We see the following, which visualizes the relationship between the queries.



7. When you double-click the arrow on the line that connects the to queries, an Edit Relationship dialog appears.



Edit Relationship

Select tables and columns that relate to one another.

Orders

| OrderDate | ShipCity | LineTotal | ShipCountry | ProductID | UnitPrice | Quantity |
|---|---|---|---|---|---|---|
| 10/8/1996 | Boise | 291.9 | USA | 16 | 13.9 | 21 |
| 10/8/1996 | Boise | 1008 | USA | 35 | 14.4 | 70 |
| 10/8/1996 | Boise | 288 | USA | 46 | 9.6 | 30 |
| 10/8/1996 | Boise | 1760 | USA | 59 | 44 | 40 |
| 10/8/1996 | Boise | 2808 | USA | 63 | 35.1 | 80 |

Products

| ProductID | ProductName | QuantityPerUnit | UnitsInStock |
|---|---|---|---|
| 1 | Chai | 10 boxes x 20 bags | 39 |
| 2 | Chang | 24 - 12 oz bottles | 17 |
| 3 | Aniseed Syrup | 12 - 550 ml bottles | 13 |
| 4 | Chef Anton's Cajun Seasoning | 48 - 6 oz jars | 53 |
| 5 | Chef Anton's Gumbo Mix | 36 boxes | 0 |

> Advanced options

OK Cancel

8. No need to make any changes, so we'll just select Cancel to close the Edit Relationship dialog.