EXCELLENCE IN EDUCATION

निर्मलस्नेह उत्तम सेवाधर्म

SHRI G.P.M. DEGREE COLLEGE OF SCIENCE & COMMERCE.

SHRI G.P.M. DEGREE COLLEGE OF SCIENCE & COMMERCE

(COMMITTED TO EXCELLENCE IN EDUCATION)

CERTIFICATE

This is to certify thar Mr/Ms ____

Student of class BSc-IT [Roll No:____] has completed the required number of practical's in the subject of Business-Intelligence as prescribed by the University of Mumbai under my supervision during the academic year 2023-2024.

Prof. In Charge

Course Co-coordinator

External Examiner

Principal

Date:________________

College Seal

Prof. Name :	**Class /SEM : T.Y. B.Sc. - IT / Sem – VI (2023-2024)**
Course Code : USIT6P3	**Subject Name : Business Intelligence**

	Theory-7: Classification **Practical-7:** Perform the data classification using classification algorithm **(IT Lab)** **Example-1:** Show use of slicer on data. **(IT Lab)** **Example-2:** Perform the data clustering using clustering algorithm in R – Programing. **(Homework)** **Example-3:** Use filters on data. **(Homework)**		
	Theory-7: Classification **Practical-7:** Perform the data classification using classification algorithm **(IT Lab)** **Example-1:** Show use of slicer on data. **(IT Lab)** **Example-2:** Perform the data clustering using clustering algorithm in R -Programing. **(Homework)** **Example-3:** Use filters on data. **(Homework)**		
	Theory-8: Clustering **Practical-8:** Perform the data clustering using clustering algorithm. **(IT Lab)** **Example-1:** Transform less structured data in power bi. **(IT Lab)** **Example-2:** Use merge query in power bi. **(Homework)**		
	Theory-9: Linear regression **Practical-9:** Perform the Linear regression on the given data ware house data.- **(IT Lab)** **Example-1:** Optimize models for reporting. **(IT Lab)** **Example-2:** Show map visualization. **(Homework)**		
	Theory-10: logistic regression **Practical-10:** Perform the logistic regression on the given data ware house -data. **(IT Lab)** **Example-1:** Perform ETL transformation on the above data by converting the attribute Name from lowercase to uppercase. **(IT Lab)** **Example-2:** What is pinning on data set? **(IT Lab)** **Example-3:** publish a report to the web from power bi. **(Homework)**		

Practical-1: Import the legacy data from different sources such as (Excel ,- Sql Server, Oracle etc.) and load in the target system. (You can download sample database such as Adventure works, North wind, food mart etc.)

Aims:

1. To understand and implement the process of loading legacy data from various sources such as Excel, SQL Server, and Oracle into a target system efficiently.

Learning Objectives:

1. Understand the importance of data migration and loading techniques.
2. Gain hands-on experience in importing data from different data sources.
3. Learn how to transform and load data into a target database.
4. Identify common challenges in data migration and methods to overcome them.

Tool & Technologies used:

1. Power BI is a powerful business intelligence tool used for data visualization and analysis.

Theory-1: Loading

Loading is the process of importing legacy data from different sources such as Excel, SQL Server, Oracle, and other database systems into a target system. This is a crucial step in data migration and ETL (Extract, Transform, Load) processes, ensuring seamless data integration for further processing and analysis.

Process of Loading

The loading phase consists of several key steps:

1. **Data Extraction:** Extract data from different legacy sources while maintaining data integrity.
2. **Data Transformation:** Perform necessary transformations such as data cleaning, validation, and mapping to match the target system's schema.
3. **Data Loading:** Transfer the transformed data into the target system, ensuring minimal downtime and data consistency.

Types of Loading

- **Full Load:** A one-time transfer of all data from the source system to the target system.
- **Incremental Load:** Only new or updated records are loaded periodically to optimize performance.
- **Batch Loading:** Data is loaded in predefined chunks to manage system resources efficiently.
- **Real-time Loading:** Continuous streaming of data to support real-time analytics and reporting.

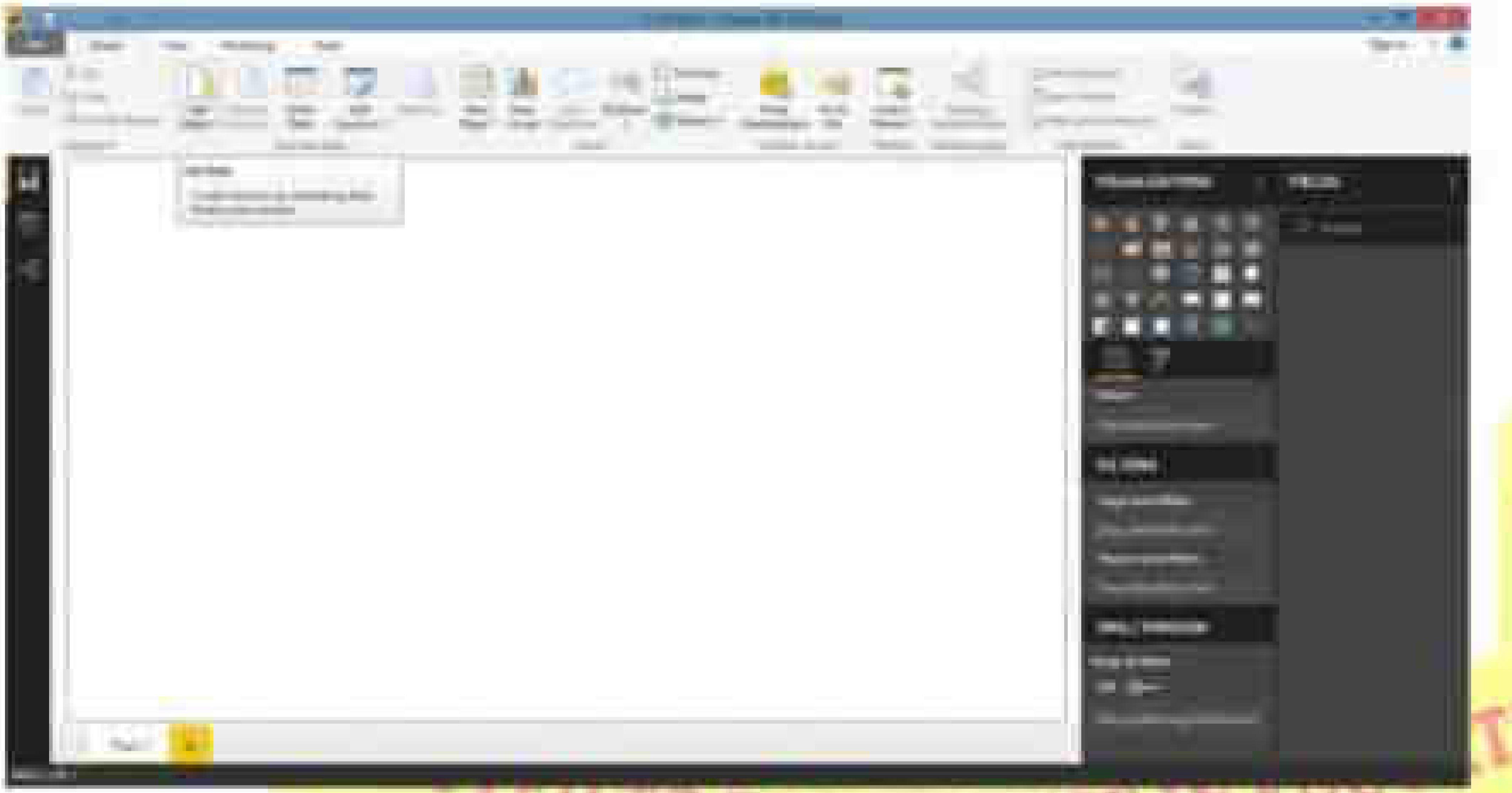
Challenges and Considerations

- **Data Compatibility:** Ensure the legacy data format aligns with the target system's structure.
- **Performance Optimization:** Efficient indexing and batch processing help improve load speed.
- **Error Handling:** Implement logging and rollback mechanisms to handle failures and ensure data consistency.
- **Security Compliance:** Maintain data confidentiality by implementing encryption and access control.

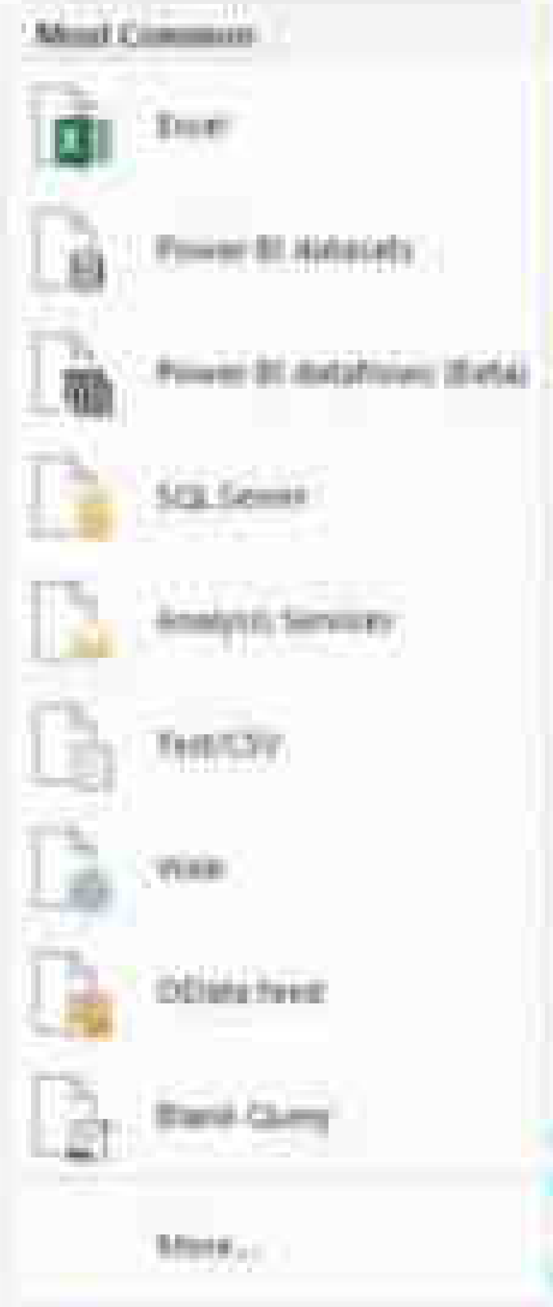
PRACTICAL 1

a.Import the legacy data from different sources such as (Excel , SqlServer, Oracle etc.) and load in the target system. (You can download sample database such as Adventureworks, Northwind, foodmart etc.)

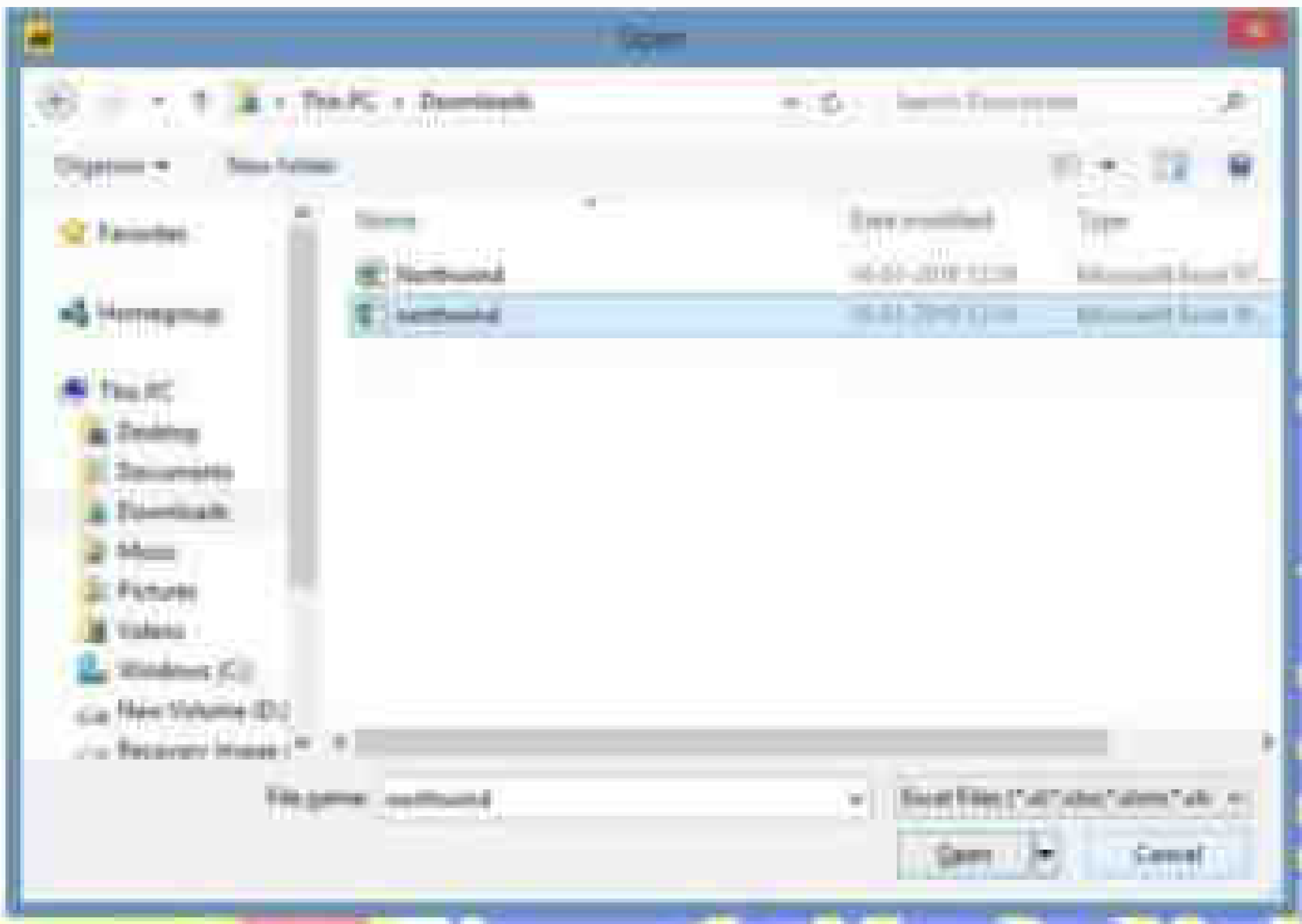
Step 1: Open Power BI



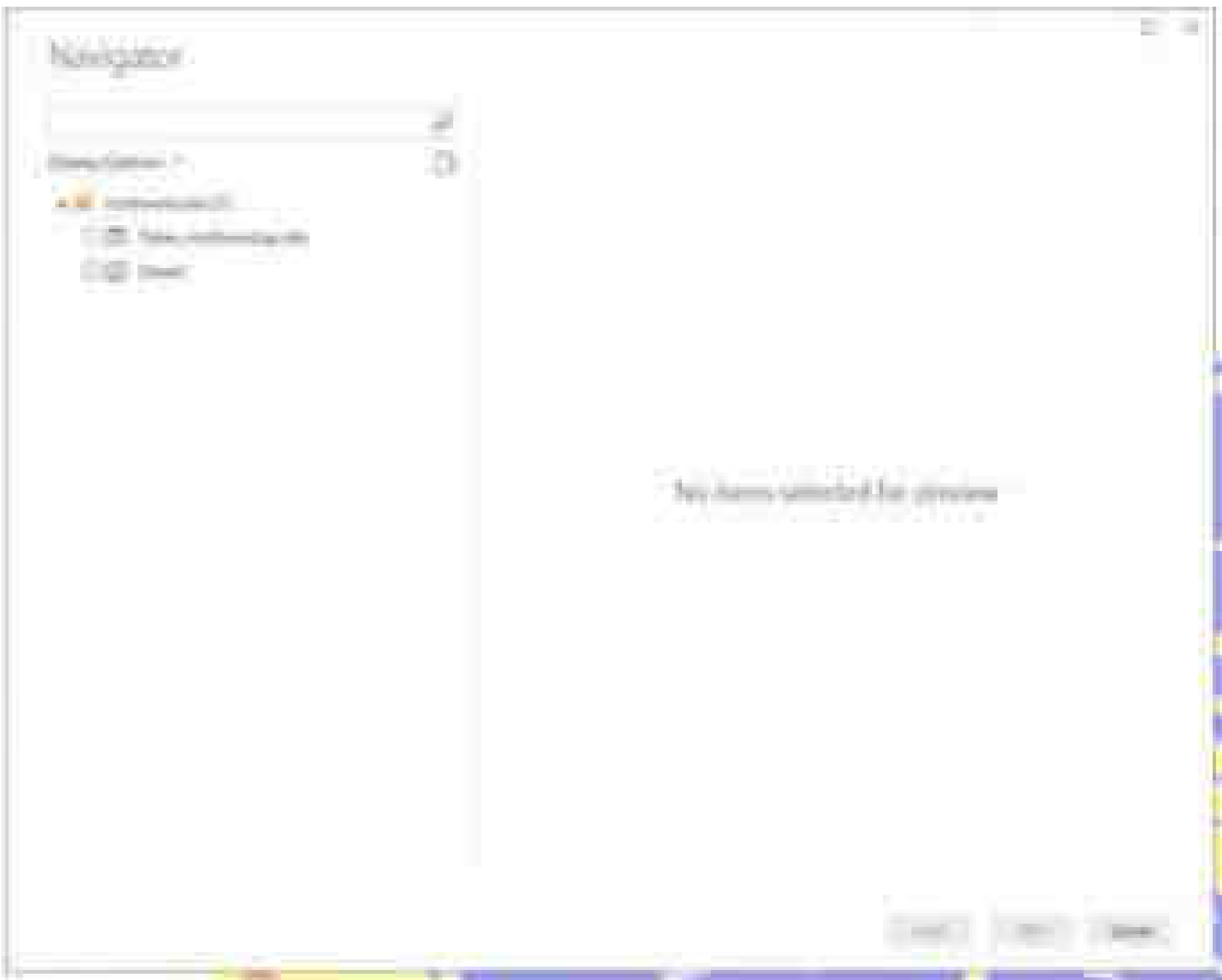
Step 2: Click on Get data following list will be displayed → select Excel

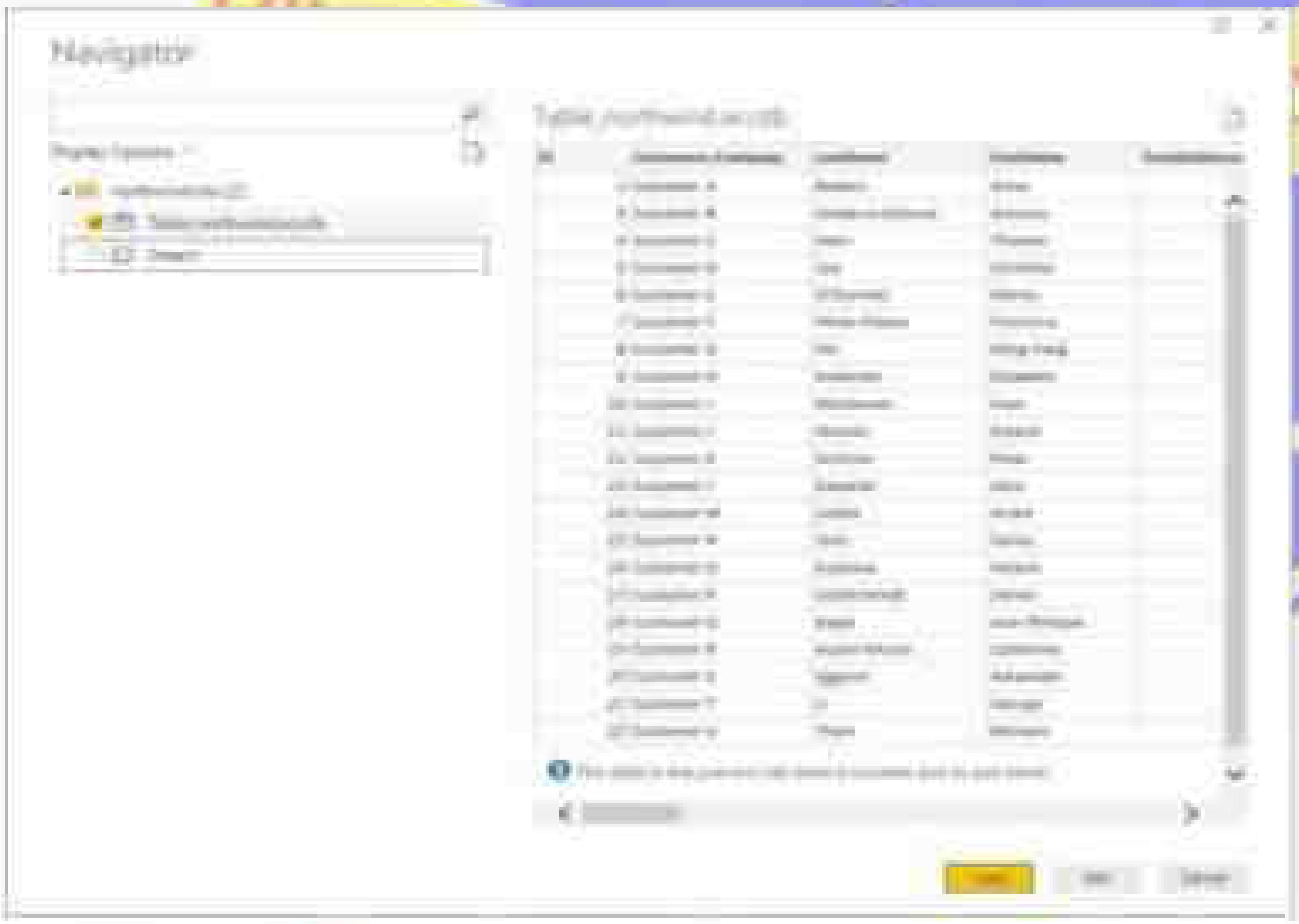


Step 3: Select required file and click on Open, Navigator screen appears

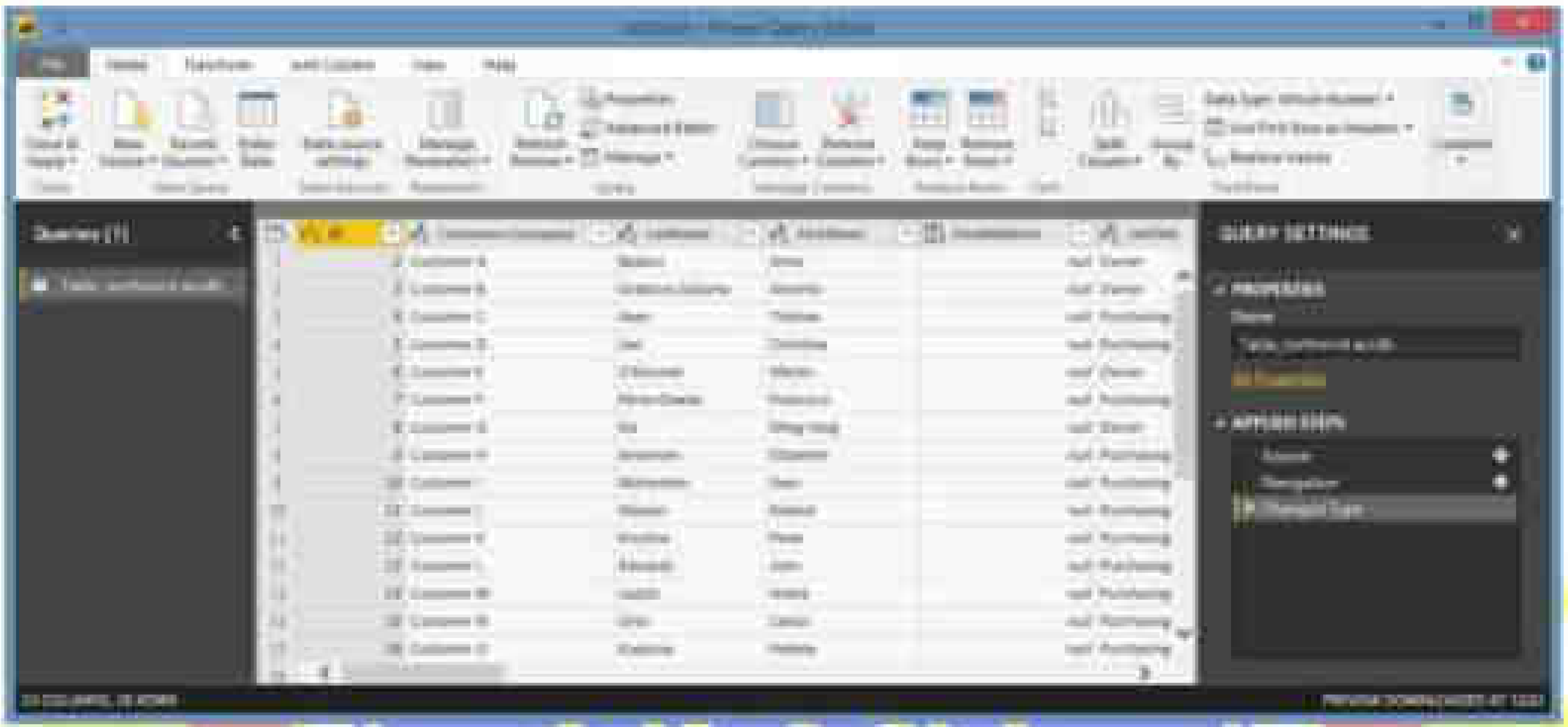


Step 4: Select file and click on edit

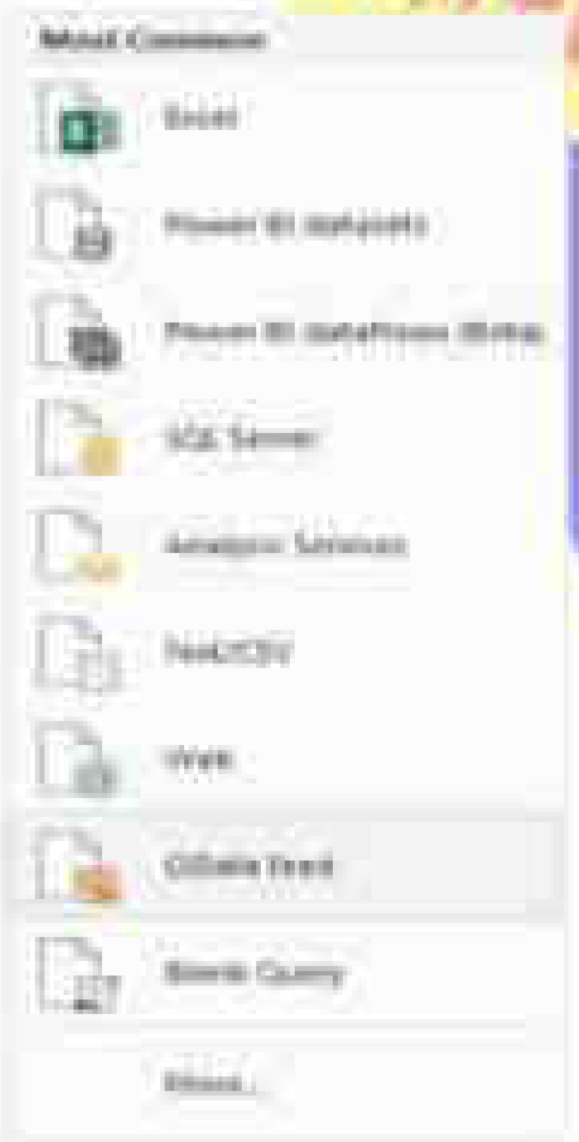




Step 5: Power query editor appears

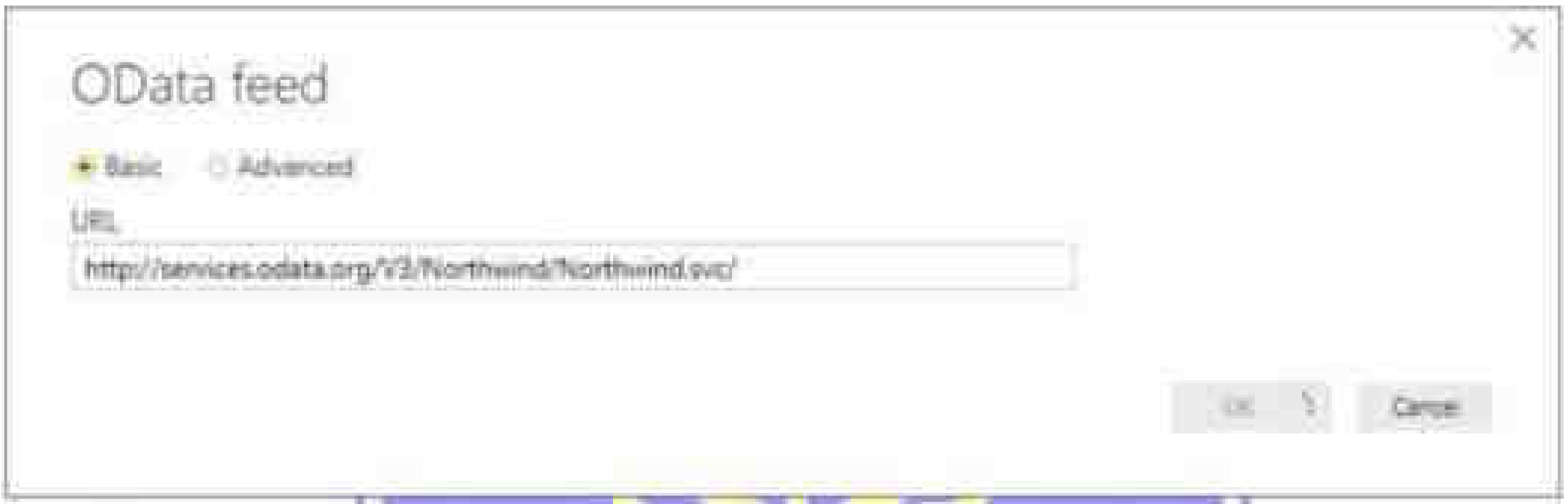


Step 6: Again, go to Get Data and select OData feed



Step 7:

Paste url as http://services.odata.org/V3/Northwind/Northwind.svc/ Click on ok



OData feed

● Basic ○ Advanced

URL

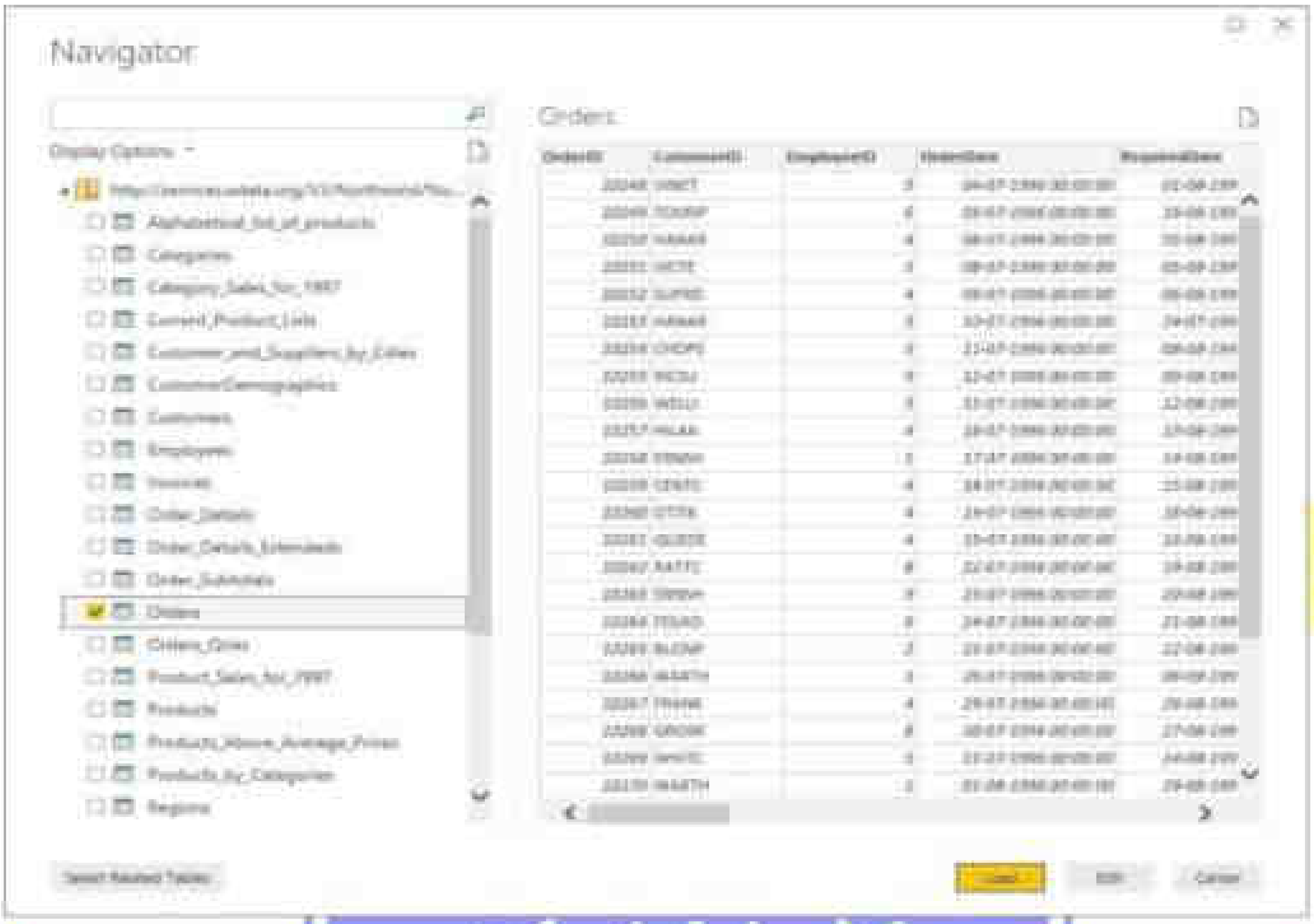
http://services.odata.org/V3/Northwind/Northwind.svc/

Step 8: Select orders table

And click on edit

Note: If you just want to see preview you can just click on table name without clicking on checkbox

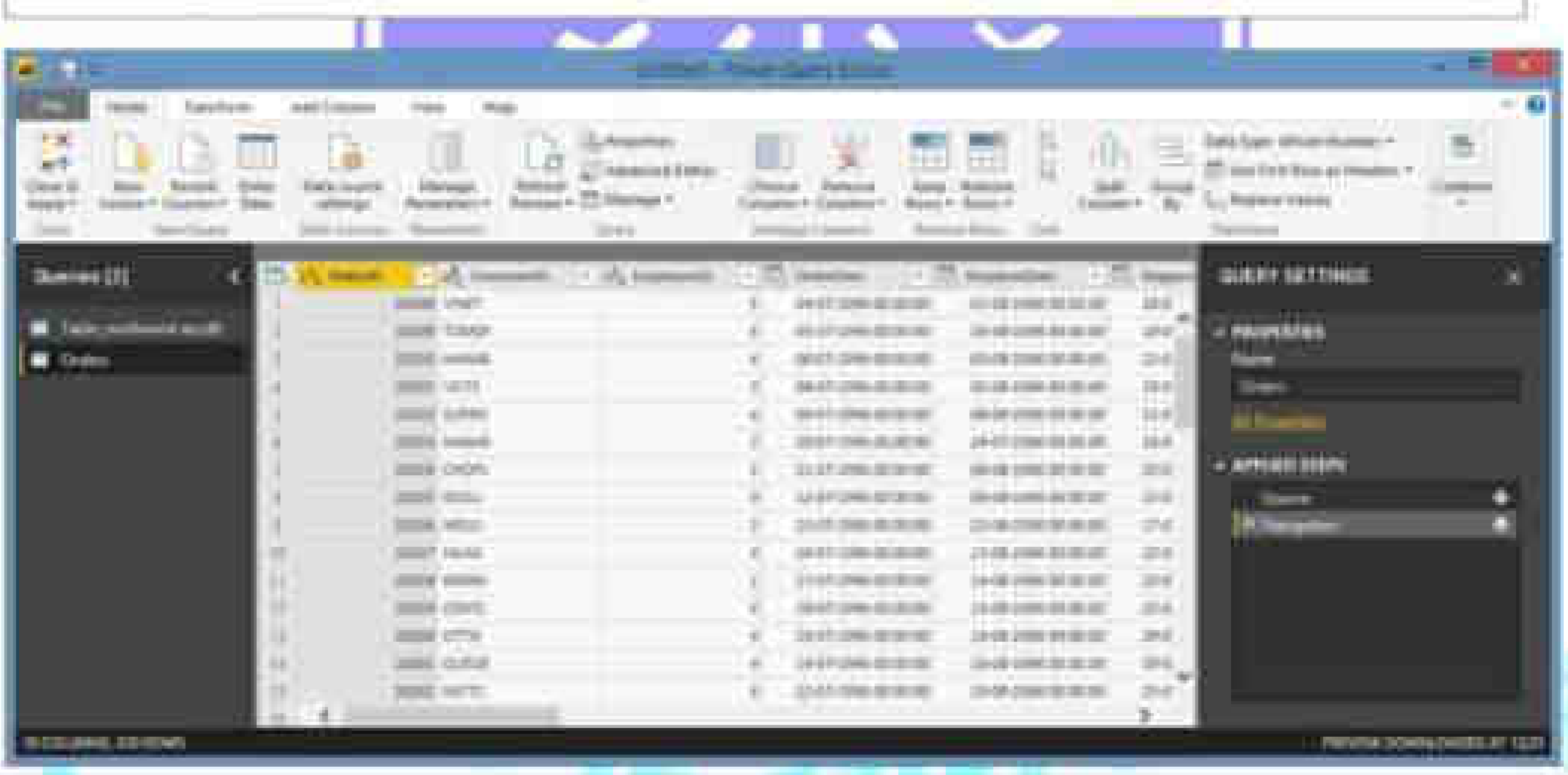
Click on edit to view table



http://services.odata.org/V3/Northwind/No...

- ☐ Alphabetical_list_of_products
- ☐ Categories
- ☐ Category_Sales_for_1997
- ☐ Current_Product_Lists
- ☐ Customer_and_Suppliers_by_Cities
- ☐ CustomerDemographics
- ☐ Customers
- ☐ Employees
- ☐ Invoices
- ☐ Order_Details
- ☐ Order_Details_Extendeds
- ☐ Order_Subtotals
- ☑ Orders
- ☐ Orders_Qries
- ☐ Product_Sales_for_1997
- ☐ Products
- ☐ Products_Above_Average_Prices
- ☐ Products_by_Categories
- ☐ Regions

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
10248	VINET	5	04-07-1996 00:00:00	01-08-199
10249	TOMSP	6	05-07-1996 00:00:00	16-08-199
10250	HANAR	4	08-07-1996 00:00:00	05-08-199
10251	VICTE	3	08-07-1996 00:00:00	05-08-199
10252	SUPRD	4	09-07-1996 00:00:00	06-08-199
10253	HANAR	3	10-07-1996 00:00:00	24-07-199
10254	CHOPS	5	11-07-1996 00:00:00	08-08-199
10255	RICSU	9	12-07-1996 00:00:00	09-08-199
10256	WELLI	3	15-07-1996 00:00:00	12-08-199
10257	HILAA	4	16-07-1996 00:00:00	13-08-199
10258	ERNSH	1	17-07-1996 00:00:00	14-08-199
10259	CENTC	4	18-07-1996 00:00:00	15-08-199
10260	OTTIK	4	19-07-1996 00:00:00	16-08-199
10261	QUEDE	4	19-07-1996 00:00:00	16-08-199
10262	RATTC	8	22-07-1996 00:00:00	19-08-199
10263	ERNSH	9	23-07-1996 00:00:00	20-08-199
10264	FOLKO	6	24-07-1996 00:00:00	21-08-199
10265	BLONP	2	25-07-1996 00:00:00	22-08-199
10266	WARTH	3	26-07-1996 00:00:00	06-09-199
10267	FRANK	4	29-07-1996 00:00:00	26-08-199
10268	GROSR	8	30-07-1996 00:00:00	27-08-199
10269	WHITC	5	31-07-1996 00:00:00	14-08-199
10270	WARTH	1	01-08-1996 00:00:00	29-08-199



Queries [2]
- [illegible]
- Orders

Name: Orders

APPLIED STEPS
- Source
- Navigation

	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	Shipped
1	10248	VINET	5	04-07-1996 00:00:00	01-08-1996 00:00:00	16-0
2	10249	TOMSP	6	05-07-1996 00:00:00	16-08-1996 00:00:00	10-0
3	10250	HANAR	4	08-07-1996 00:00:00	05-08-1996 00:00:00	12-0
4	10251	VICTE	3	08-07-1996 00:00:00	05-08-1996 00:00:00	15-0
5	10252	SUPRD	4	09-07-1996 00:00:00	06-08-1996 00:00:00	11-0
6	10253	HANAR	3	10-07-1996 00:00:00	24-07-1996 00:00:00	16-0
7	10254	CHOPS	5	11-07-1996 00:00:00	08-08-1996 00:00:00	23-0
8	10255	RICSU	9	12-07-1996 00:00:00	09-08-1996 00:00:00	15-0
9	10256	WELLI	3	15-07-1996 00:00:00	12-08-1996 00:00:00	17-0
10	10257	HILAA	4	16-07-1996 00:00:00	13-08-1996 00:00:00	22-0
11	10258	ERNSH	1	17-07-1996 00:00:00	14-08-1996 00:00:00	23-0
12	10259	CENTC	4	18-07-1996 00:00:00	15-08-1996 00:00:00	25-0
13	10260	OTTIK	4	19-07-1996 00:00:00	16-08-1996 00:00:00	29-0
14	10261	QUEDE	4	19-07-1996 00:00:00	16-08-1996 00:00:00	30-0
15	10262	RATTC	8	22-07-1996 00:00:00	19-08-1996 00:00:00	25-0

Learning Outcomes:

Course Outcomes:

Conclusion:

Viva Questions:

1. **Define data loading.**
2. **What is incremental loading?**
3. **Why validate data during loading?**
4. **Mention one data loading challenge.**

For Faculty use:

Correction Parameters	Formative Assessment[40%]	Timely Completion of Practical[40%]	Attendance Learning Attitude[20%]

Practical-2: Perform the Extraction Transformation and Loading (ETL) -process to construct the database in the Sql server.

Aims:

1. To implement a complete ETL process by extracting data from multiple sources, transforming it as necessary, and loading it into a SQL Server database.
2. To build a reliable and optimized database using ETL techniques for improved data management and reporting.

Learning Objectives:

1. Understand the ETL methodology and its role in data warehousing.
2. Gain proficiency in extracting data from diverse sources such as Excel, SQL Server, and Oracle.
3. Learn to perform data transformations including cleansing, formatting, and aggregation.
4. Master the process of loading transformed data into SQL Server and validating its integrity.

Tool & Technologies used:

1. Power BI is a powerful business intelligence tool used for data visualization and analysis.

Theory-2: Extraction

Extraction is the first phase of the ETL (Extract, Transform, Load) process, which is essential for constructing a database in SQL Server. This step involves retrieving data from multiple sources such as relational databases, flat files, APIs, and cloud storage systems to ensure that accurate and relevant data is available for further processing.

Process of Extraction

The extraction process includes the following steps:

1. **Identifying Data Sources:** Determine and analyze the structure and format of legacy data sources.
2. **Data Retrieval:** Extract data using various methods such as SQL queries, API calls, or file parsing.
3. **Data Staging:** Store the extracted data temporarily in a staging area to maintain integrity before transformation.
4. **Data Validation:** Check data completeness, consistency, and correctness to ensure reliability.

Types of Extraction

- **Full Extraction:** Extracts all data from the source system at once, typically used for initial loads.
- **Incremental Extraction:** Only new or modified data is extracted periodically, reducing system load.
- **Real-time Extraction:** Continuous extraction to support real-time analytics and processing.

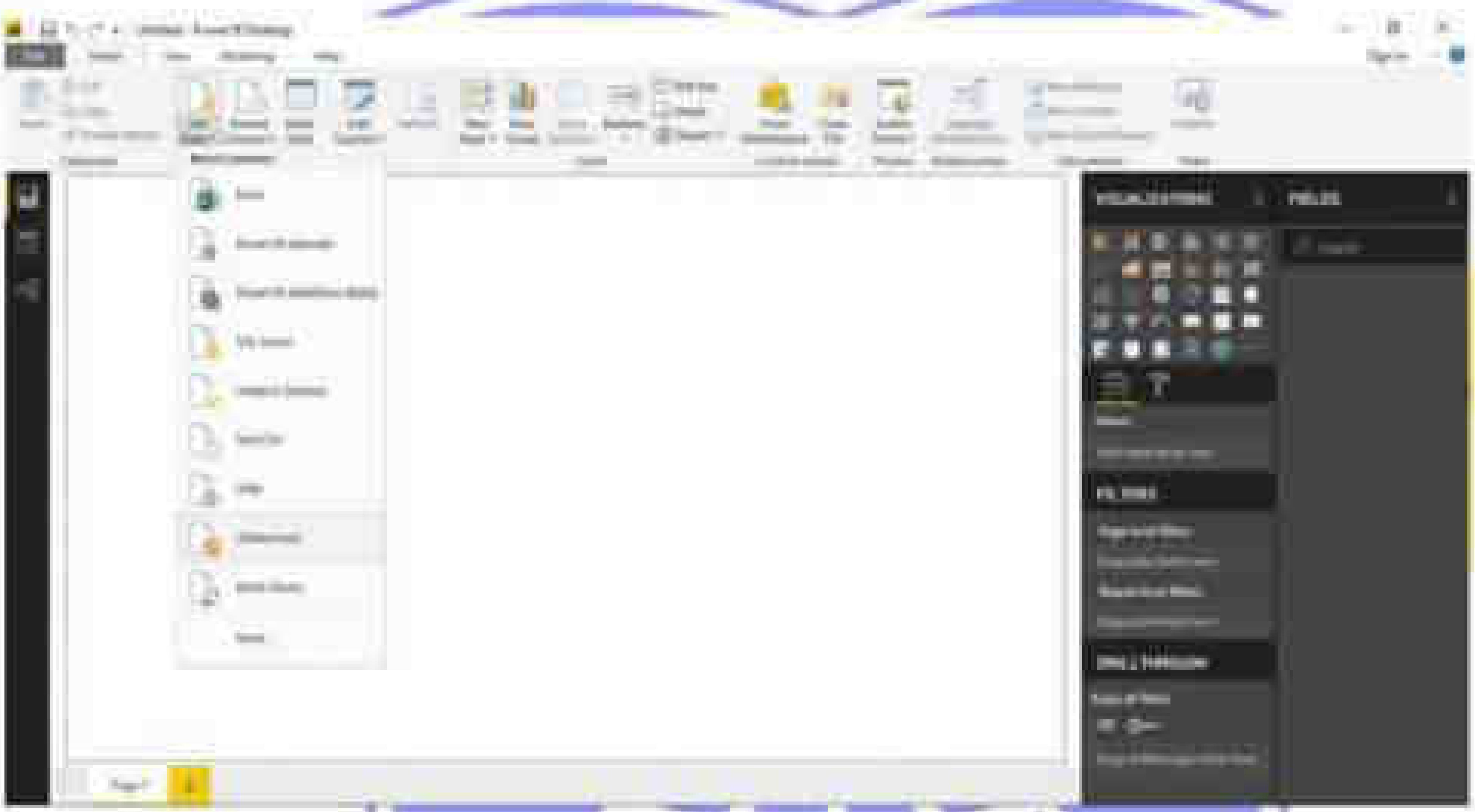
Challenges and Considerations

- **Performance Issues:** Large data volumes can impact system performance; optimizing queries and indexing helps mitigate this.
- **Data Integrity:** Ensure extracted data remains consistent and unaltered during transfer.
- **Security Concerns:** Implement encryption and secure connections to protect sensitive data.
- **Handling Data Anomalies:** Implement error detection and correction mechanisms to avoid loading faulty data.

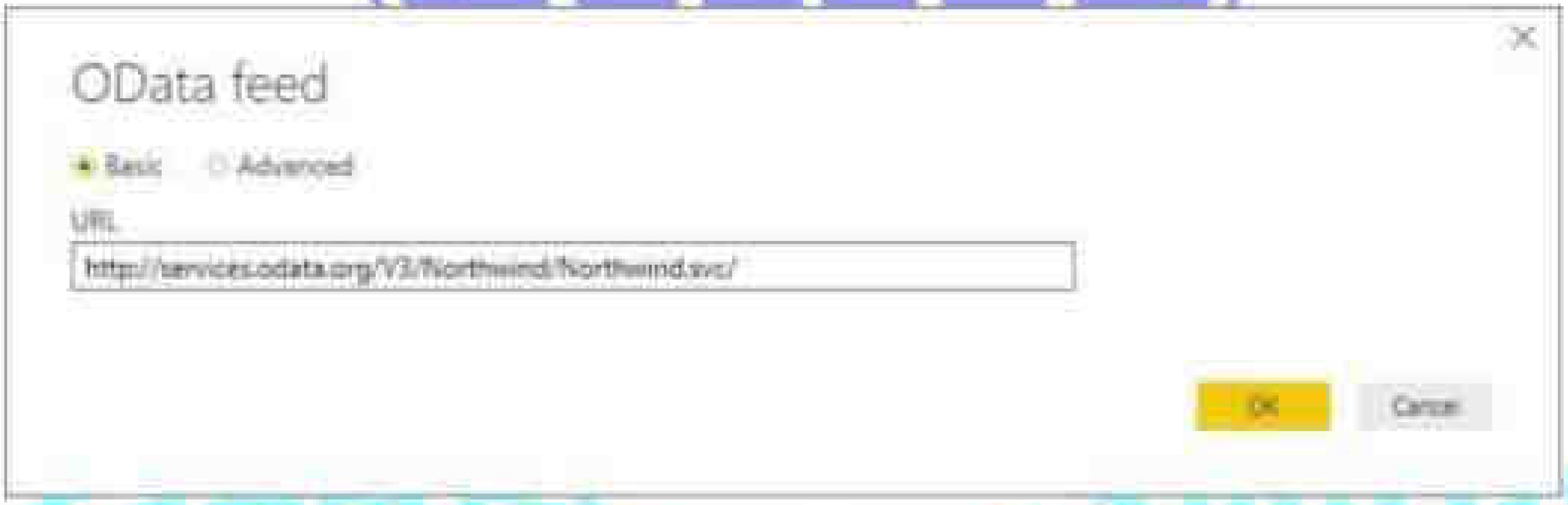
PRACTICAL 2

Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Power BI.

Step 1: Open Power BI. Click on Get Data → OData Feed



Paste Url : http://services.odata.org/V3/Northwind/Northwind.svc/ And Click OK



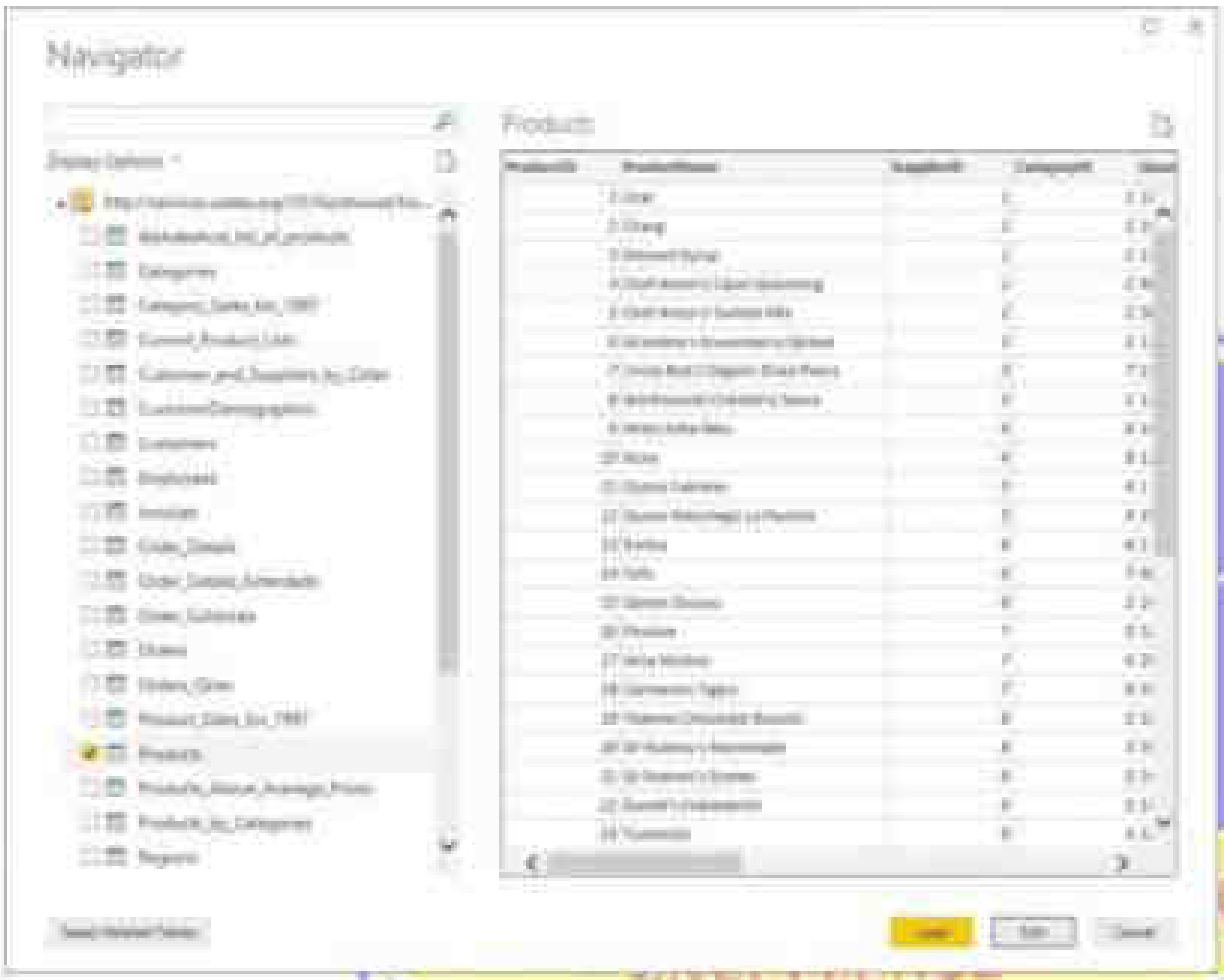
OData feed

● Basic ○ Advanced

URL

http://services.odata.org/V3/Northwind/Northwind.svc/

Step 2: Click on Check Box of Products table and then click on Edit



Navigator

Products

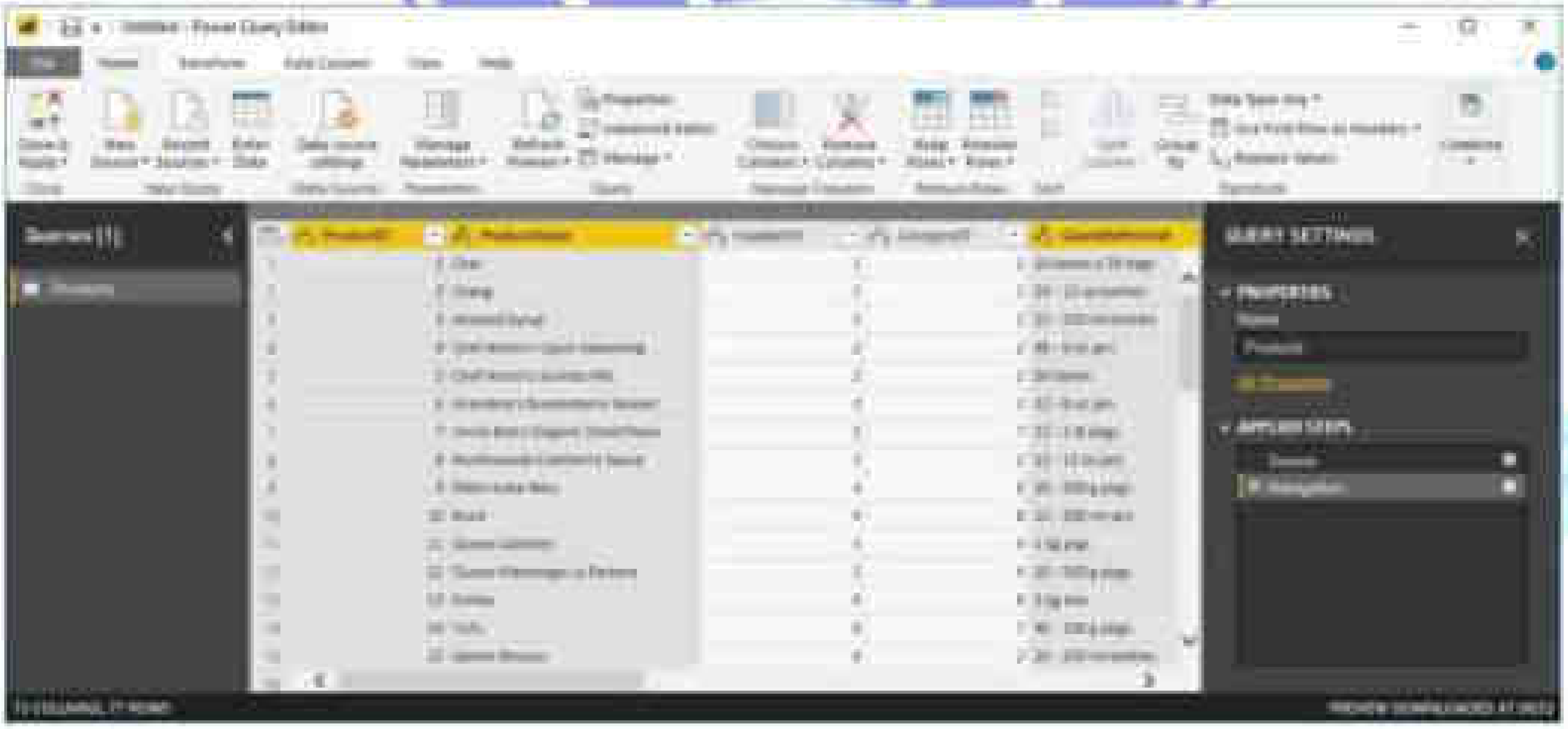
Display Options

- http://services.odata.org/V3/Northwind/Nor...
- □ Alphabetical_list_of_products
- □ Categories
- □ Category_Sales_for_1997
- □ Current_Product_Lists
- □ Customer_and_Suppliers_by_Cities
- □ CustomerDemographics
- □ Customers
- □ Employees
- □ Invoices
- □ Order_Details
- □ Order_Details_Extendeds
- □ Order_Subtotals
- □ Orders
- □ Orders_Qries
- □ Product_Sales_for_1997
- ☑ Products
- □ Products_Above_Average_Prices
- □ Products_by_Categories
- □ Regions

ProductID	ProductName	SupplierID
1	Chai	1
2	Chang	1
3	Aniseed Syrup	1
4	Chef Anton's Cajun Seasoning	2
5	Chef Anton's Gumbo Mix	2
6	Grandma's Boysenberry Spread	3
7	Uncle Bob's Organic Dried Pears	3
8	Northwoods Cranberry Sauce	3
9	Mishi Kobe Niku	4
10	Ikura	4
11	Queso Cabrales	5
12	Queso Manchego La Pastora	5
13	Konbu	6
14	Tofu	6
15	Genen Shouyu	6
16	Pavlova	7
17	Alice Mutton	7
18	Carnarvon Tigers	7
19	Teatime Chocolate Biscuits	8
20	Sir Rodney's Marmalade	8
21	Sir Rodney's Scones	8
22	Gustaf's Knäckebröd	9
23	Tunnbröd	9

1) Remove other columns to only display columns of interest

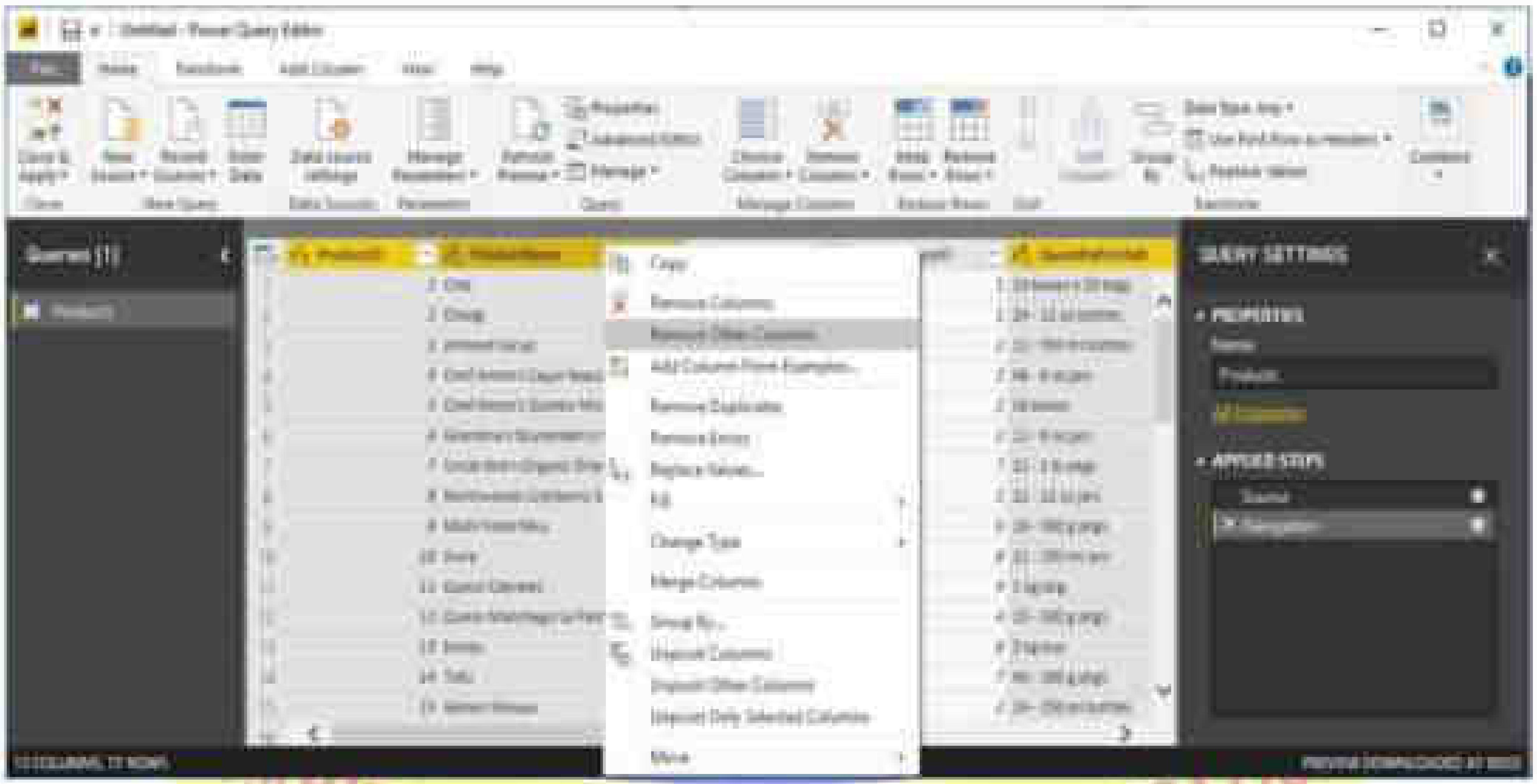
In Query Editor, select the ProductID, ProductName, QuantityPerUnit, and UnitsInStock columns (use Ctrl+Click to select more than one column, or Shift+Click to select columns that are beside each other).



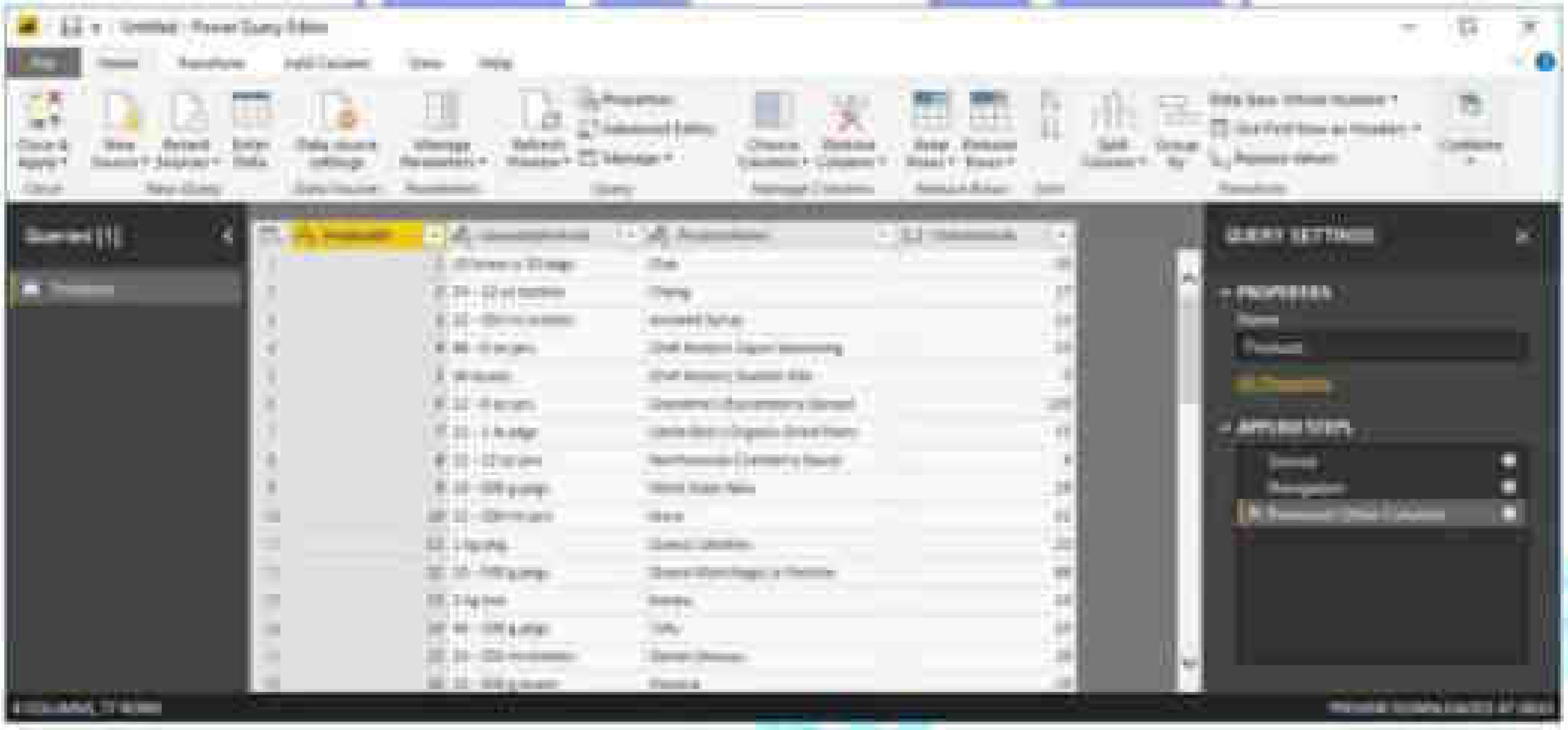
Untitled - Power Query Editor

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit
1	1	Chai	1	1	10 boxes x 20 bags
2	2	Chang	1	1	24 - 12 oz bottles
3	3	Aniseed Syrup	1	2	12 - 550 ml bottles
4	4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars
5	5	Chef Anton's Gumbo Mix	2	2	36 boxes
6	6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars
7	7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.
8	8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars
9	9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.
10	10	Ikura	4	8	12 - 200 ml jars
11	11	Queso Cabrales	5	4	1 kg pkg.
12	12	Queso Manchego La Pastora	5	4	10 - 500 g pkgs.
13	13	Konbu	6	8	2 kg box
14	14	Tofu	6	7	40 - 100 g pkgs.
15	15	Genen Shouyu	6	2	24 - 250 ml bottles

Select Remove Columns > Remove Other Columns from the ribbon, or rightclick on a column header and click Remove Other Columns



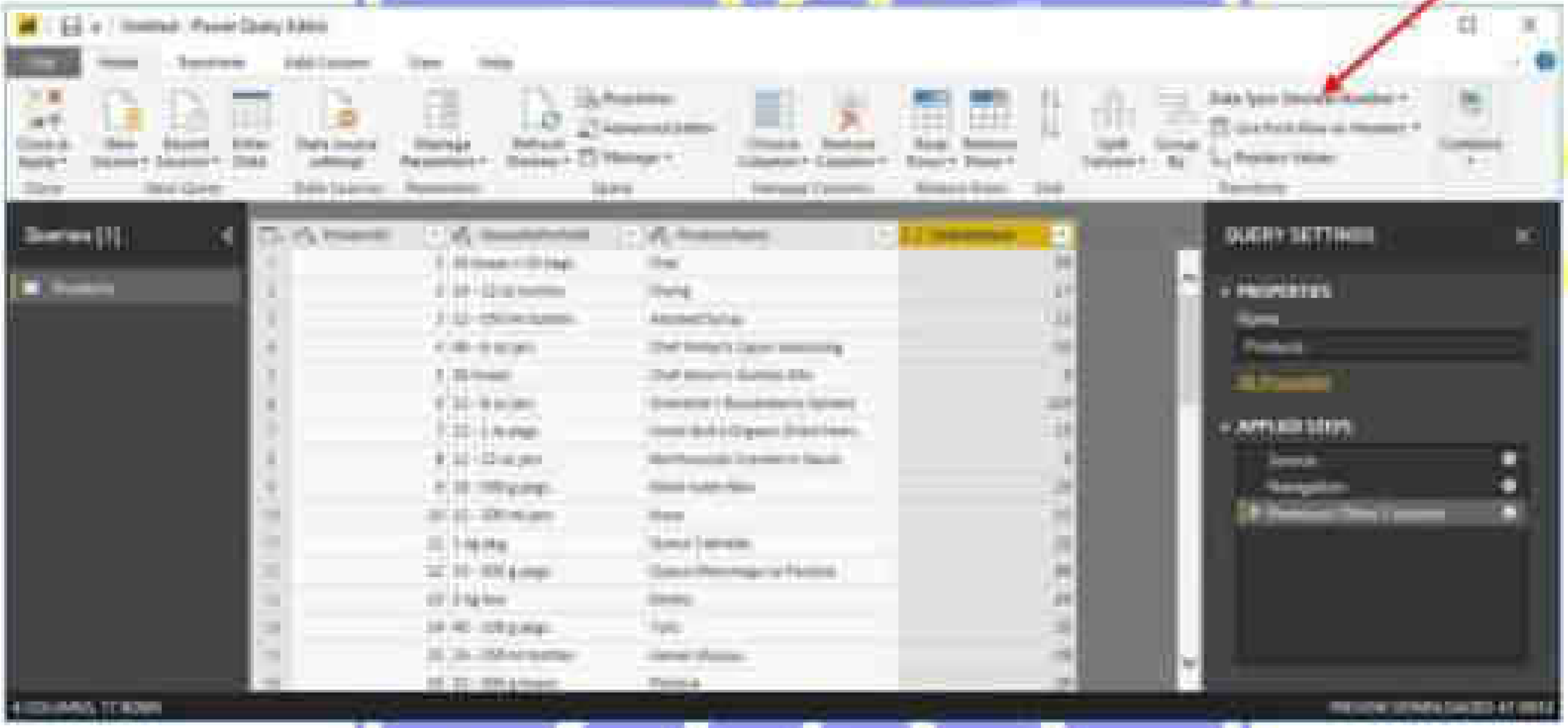
After selecting Remove Other Columns only selected four columns are displayed other columns are discarded.



2. Change the data type of the UnitsInStock column

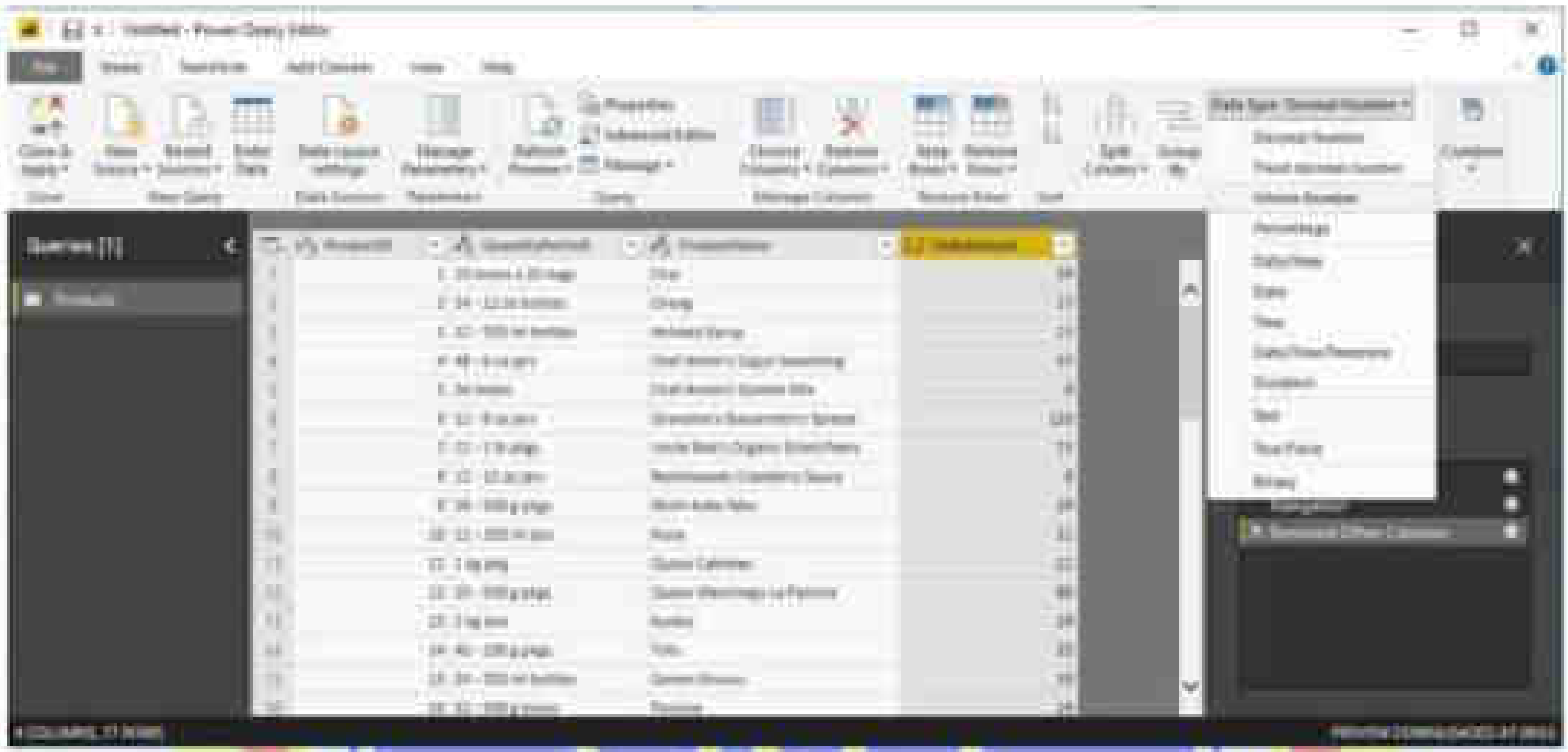
a) Select the UnitsinStock column.

Check if the data type of selected column is a Whole number

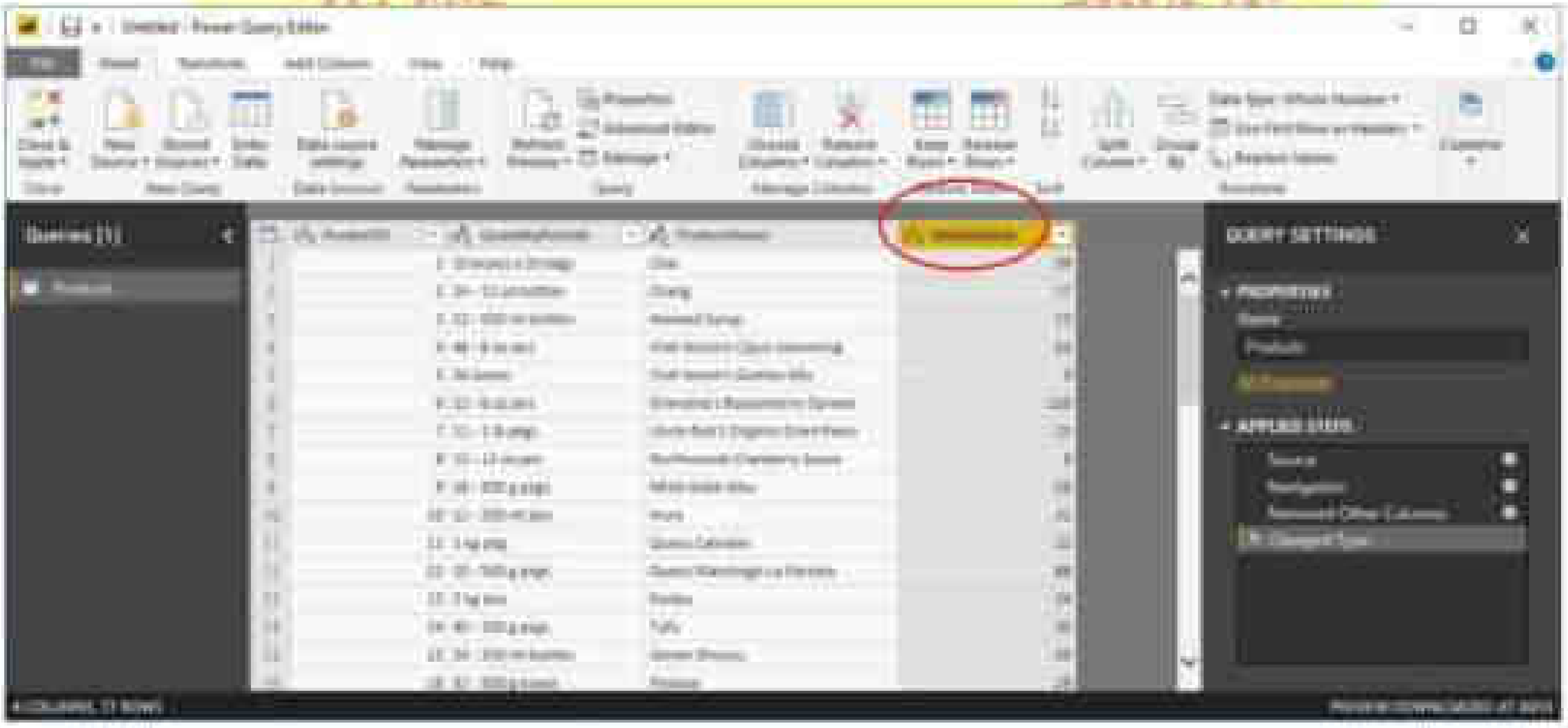


b) Select the Data Type drop-down button in the Home ribbon.

c) 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).

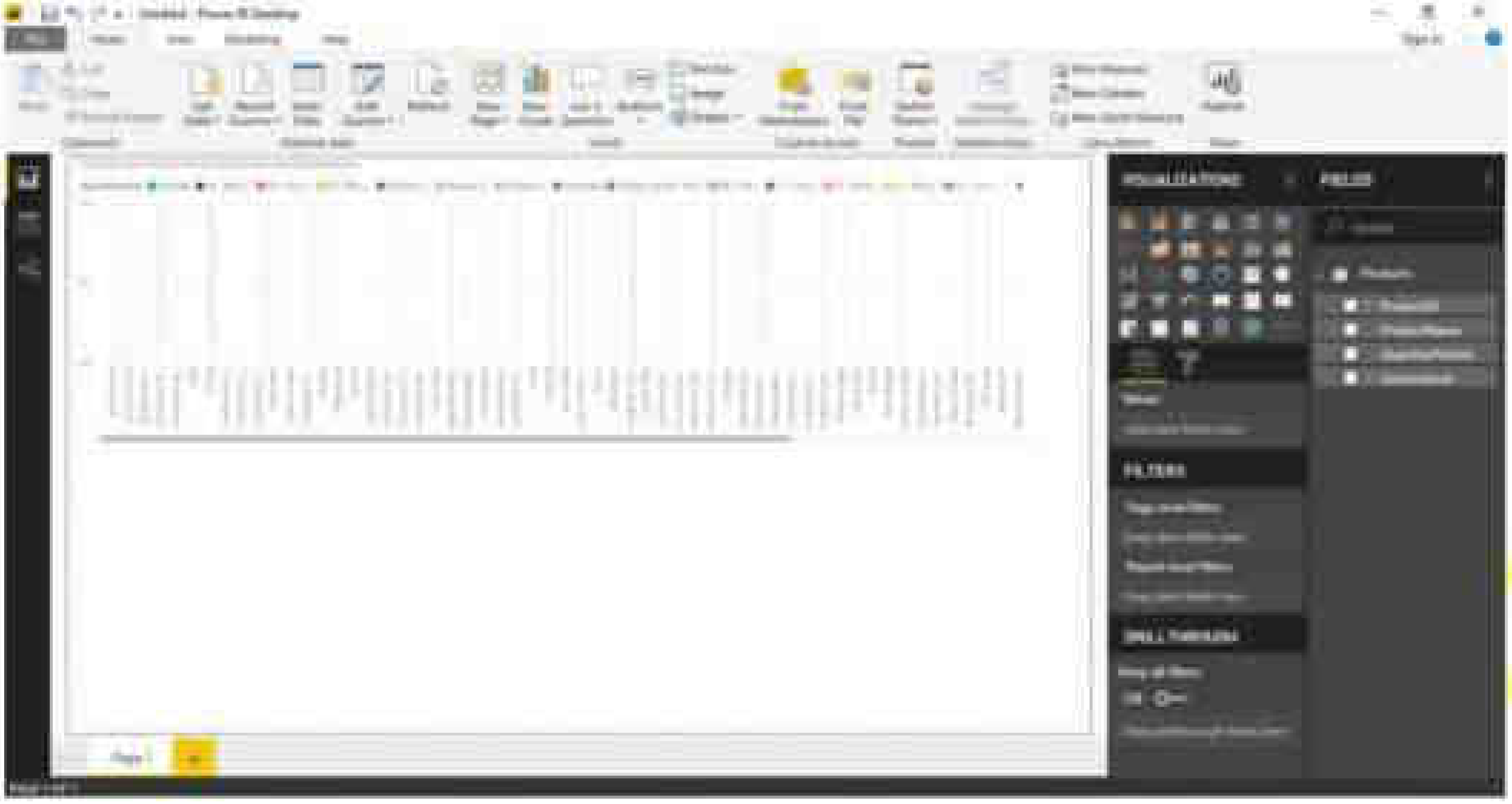


After clicking on Whole number, you can see the changed Datatype in column header of UnitsInStock.



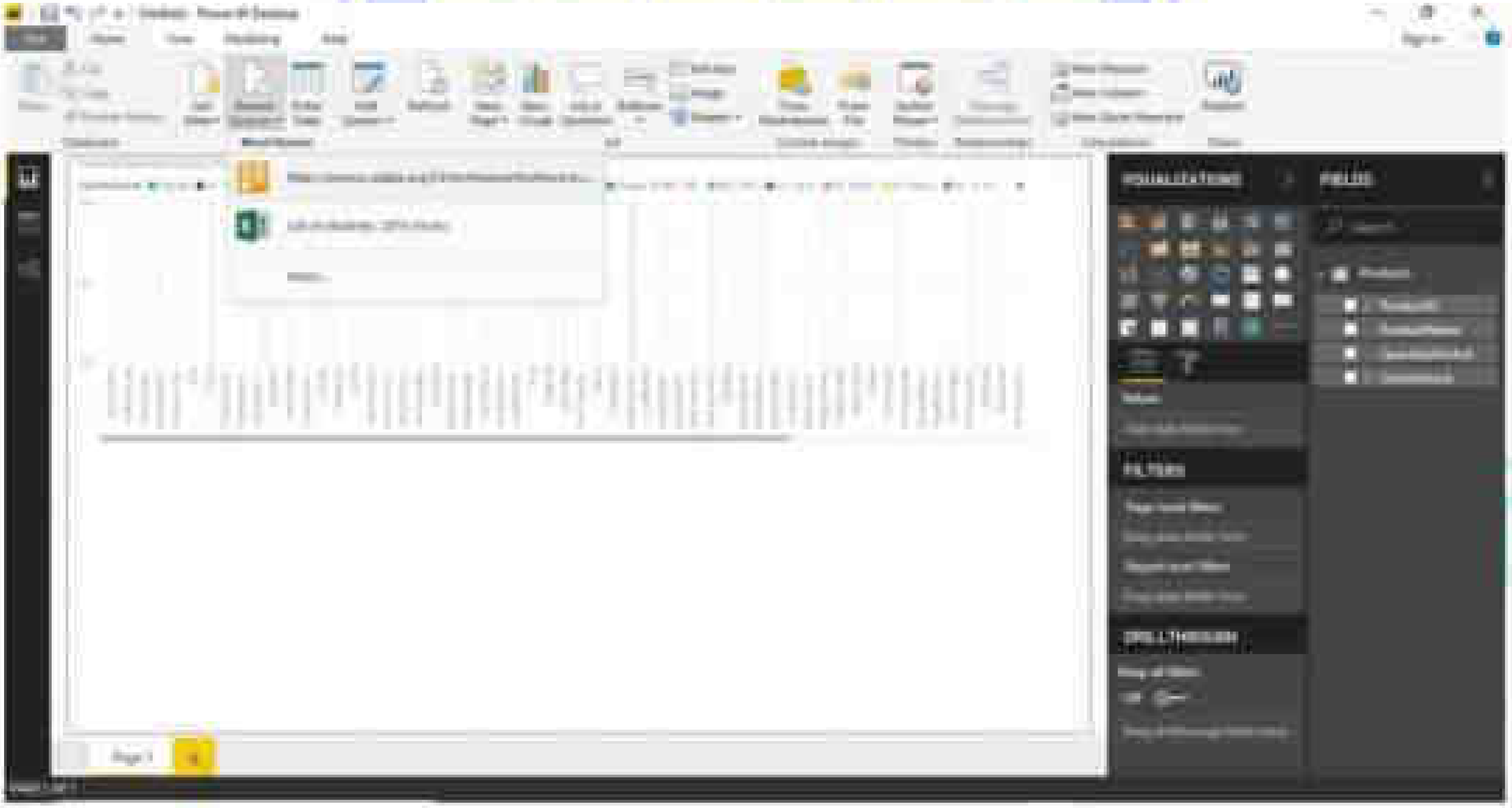
After above step, close query editor and click on Yes to save changes.

Now you can view fields of Products table on right side, check all the fields of table to get representation in charts form.



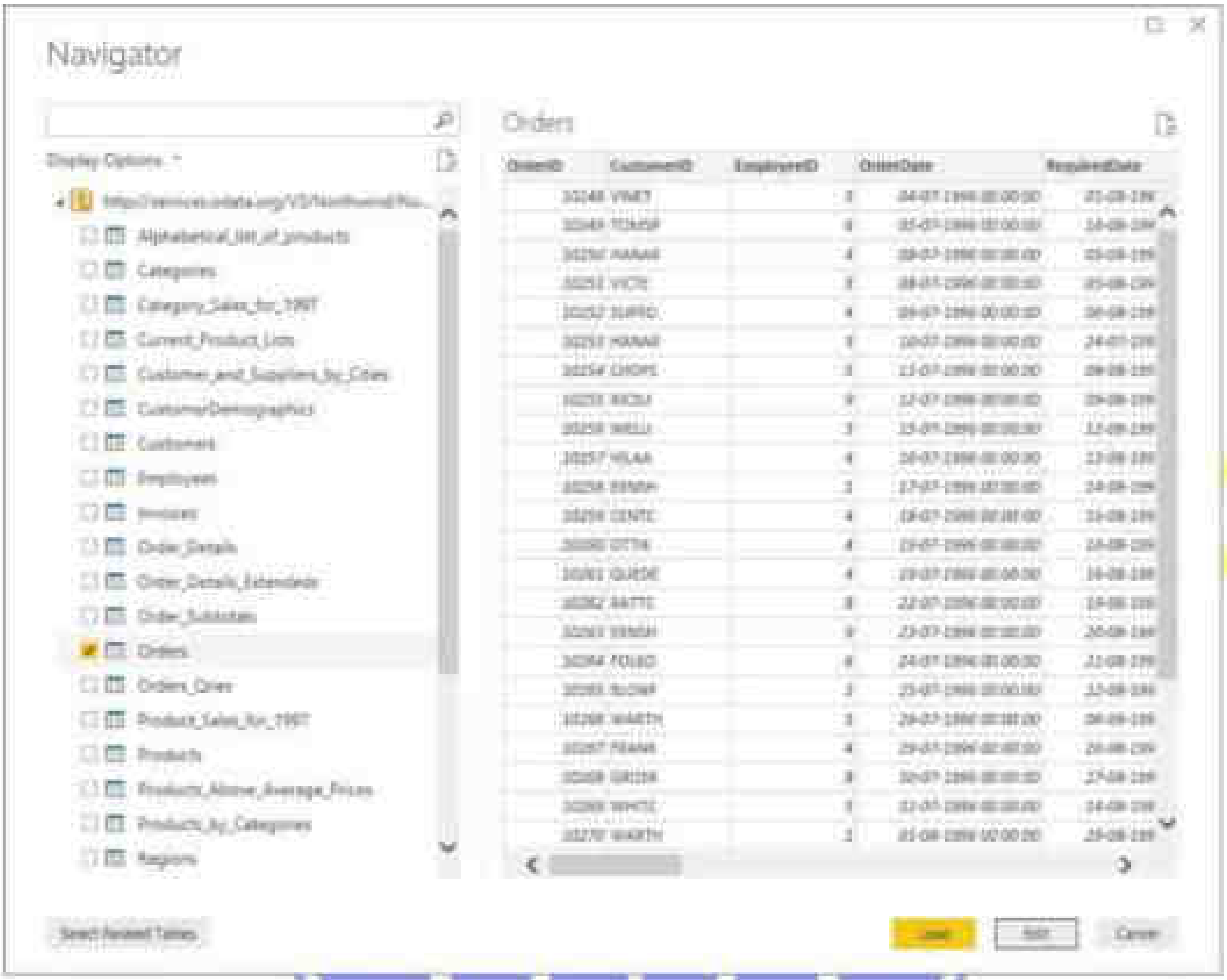
3. Expand the Orders table

Once You have loaded a data source, you can click on Recent Sources to select desired table (Orders).



After selecting the URL, Navigator window will appear from which you can select Orders table.

Click on Edit.



Orders

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate
10248	VINET	5	04-07-1996 00:00:00	01-08-199
10249	TOMSP	6	05-07-1996 00:00:00	16-08-199
10250	HANAR	4	08-07-1996 00:00:00	05-08-199
10251	VICTE	3	08-07-1996 00:00:00	05-08-199
10252	SUPRD	4	09-07-1996 00:00:00	06-08-199
10253	HANAR	3	10-07-1996 00:00:00	24-07-199
10254	CHOPS	5	11-07-1996 00:00:00	08-08-199
10255	RICSU	9	12-07-1996 00:00:00	09-08-199
10256	WELLI	3	15-07-1996 00:00:00	12-08-199
10257	HILAA	4	16-07-1996 00:00:00	13-08-199
10258	ERNSH	1	17-07-1996 00:00:00	14-08-199
10259	CENTC	4	18-07-1996 00:00:00	15-08-199
10260	OTTIK	4	19-07-1996 00:00:00	16-08-199
10261	QUEDE	4	19-07-1996 00:00:00	16-08-199
10262	RATTC	8	22-07-1996 00:00:00	19-08-199
10263	ERNSH	9	23-07-1996 00:00:00	20-08-199
10264	FOLKO	6	24-07-1996 00:00:00	21-08-199
10265	BLONP	2	25-07-1996 00:00:00	22-08-199
10266	WARTH	3	26-07-1996 00:00:00	06-09-199
10267	FRANK	4	29-07-1996 00:00:00	26-08-199
10268	GROSR	8	30-07-1996 00:00:00	27-08-199
10269	WHITC	5	31-07-1996 00:00:00	14-08-199
10270	WARTH	1	01-08-1996 00:00:00	29-08-199

Query Editor Window will appear

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

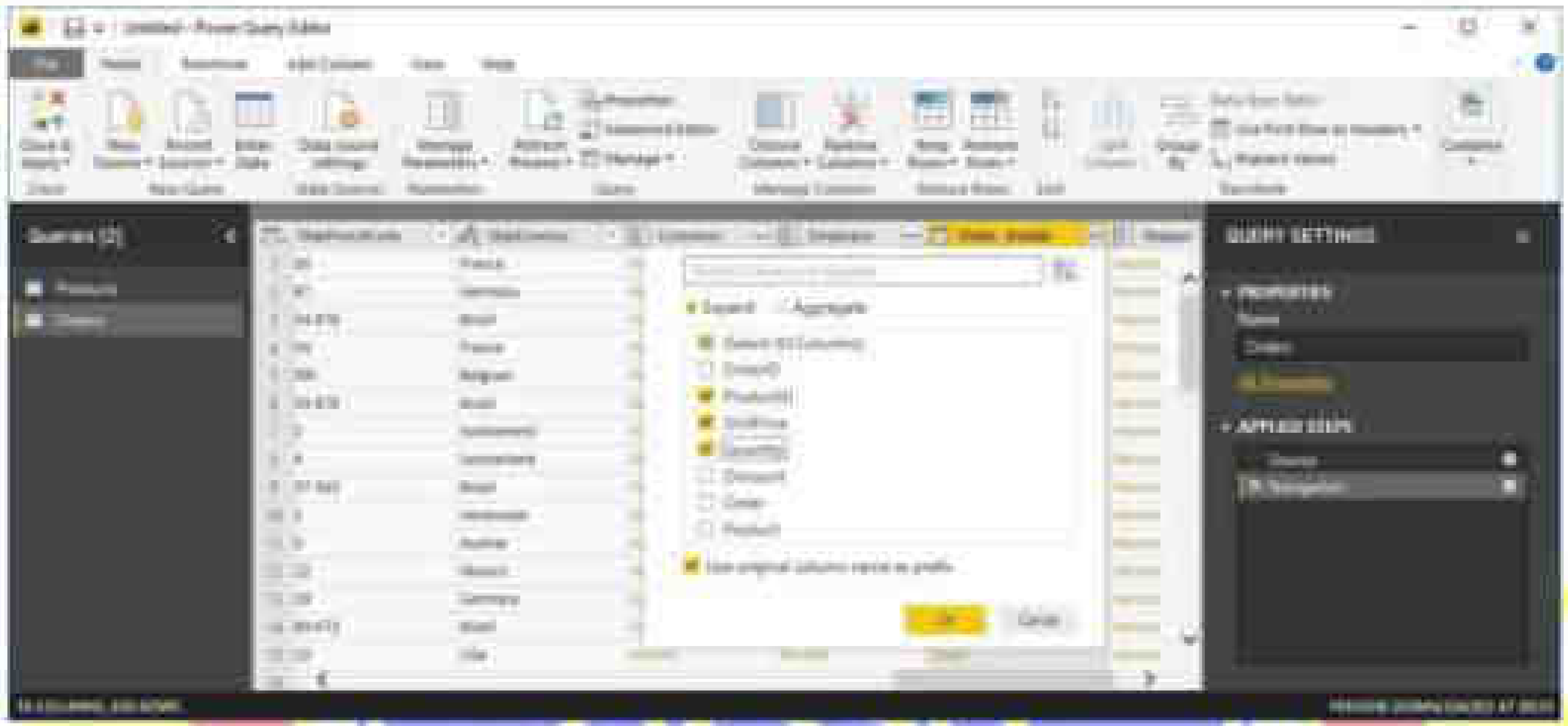
2. In the Order_Details column, select the expand icon .

3. In the Expand drop-down:

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

b. Select ProductID, UnitPrice, and Quantity.

c. Click OK.



● Expand ○ Aggregate

☑ (Select All Columns)

□ OrderID

☑ ProductID

☑ UnitPrice

☑ Quantity

□ Discount

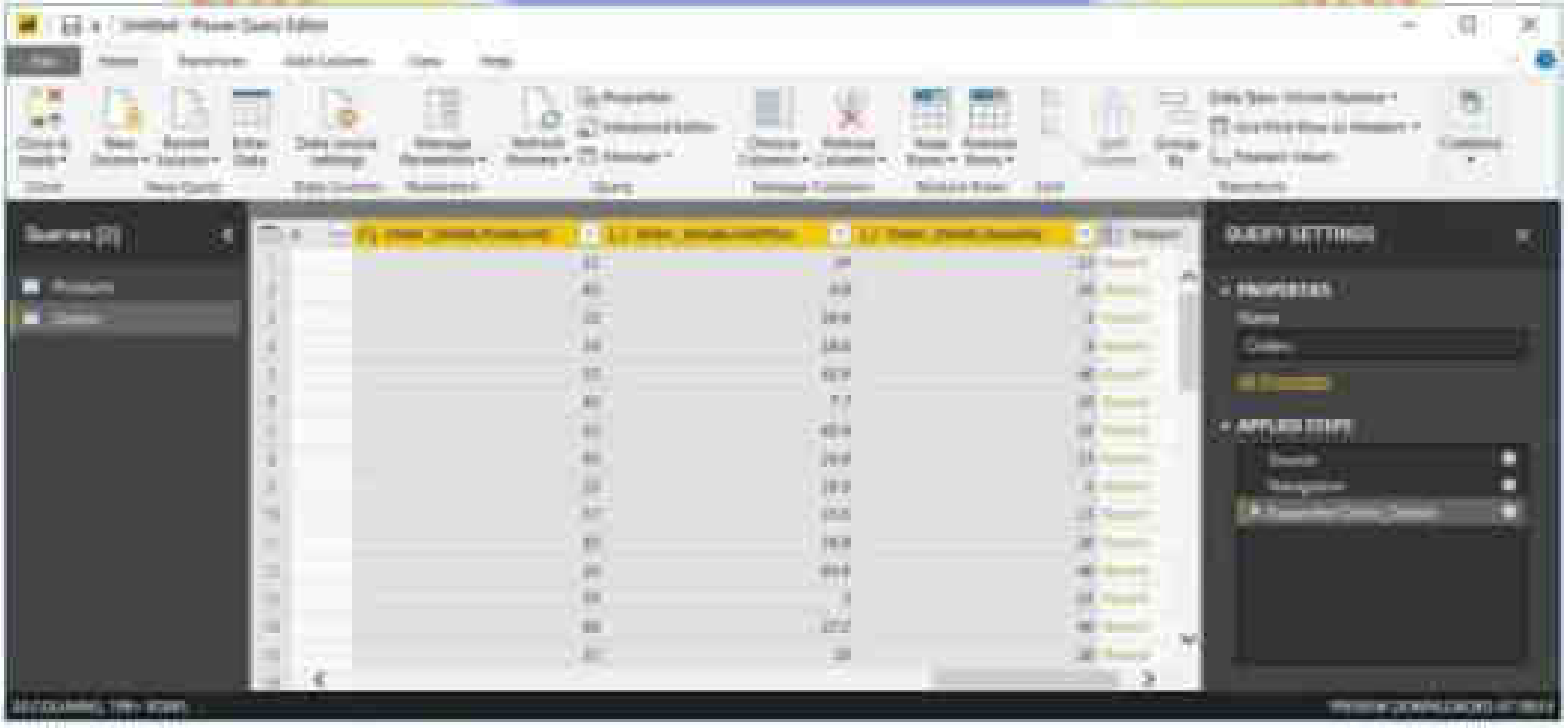
□ Order

□ Product

☑ Use original column name as prefix

OK Cancel

After clicking on OK following screen appears with combined columns

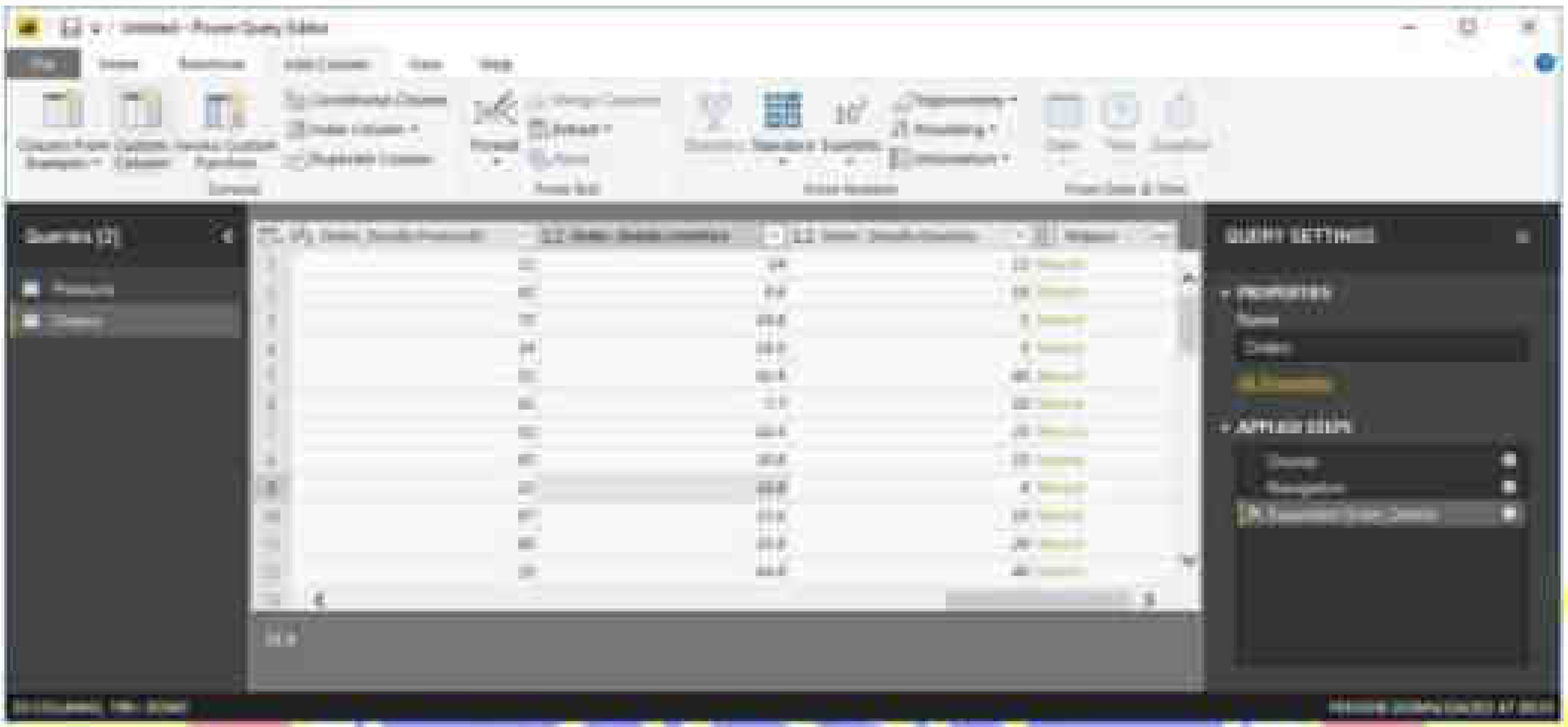


4. Calculate the line total for each Order_Details row

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:

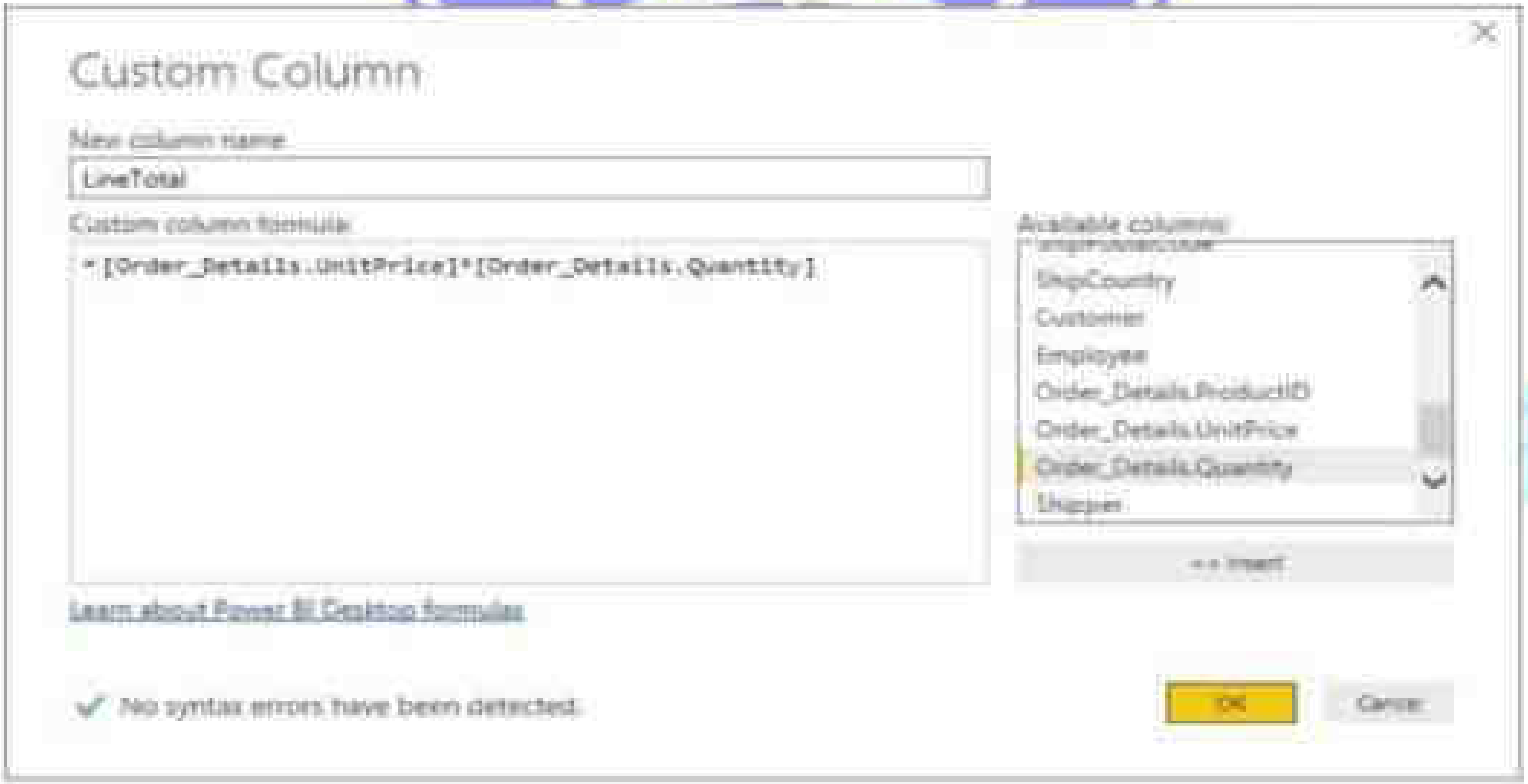
a) In the Add Column ribbon tab, click Add Custom Column.



b) In the Custom Column dialog box, in the Custom Column Formula textbox, enter [Order_Details.UnitPrice] *
[Order_Details.Quantity] by selecting from available columns and click on insert for each column.

c) In the New column name textbox, enter LineTotal.

d) Click OK.



Custom Column

New column name

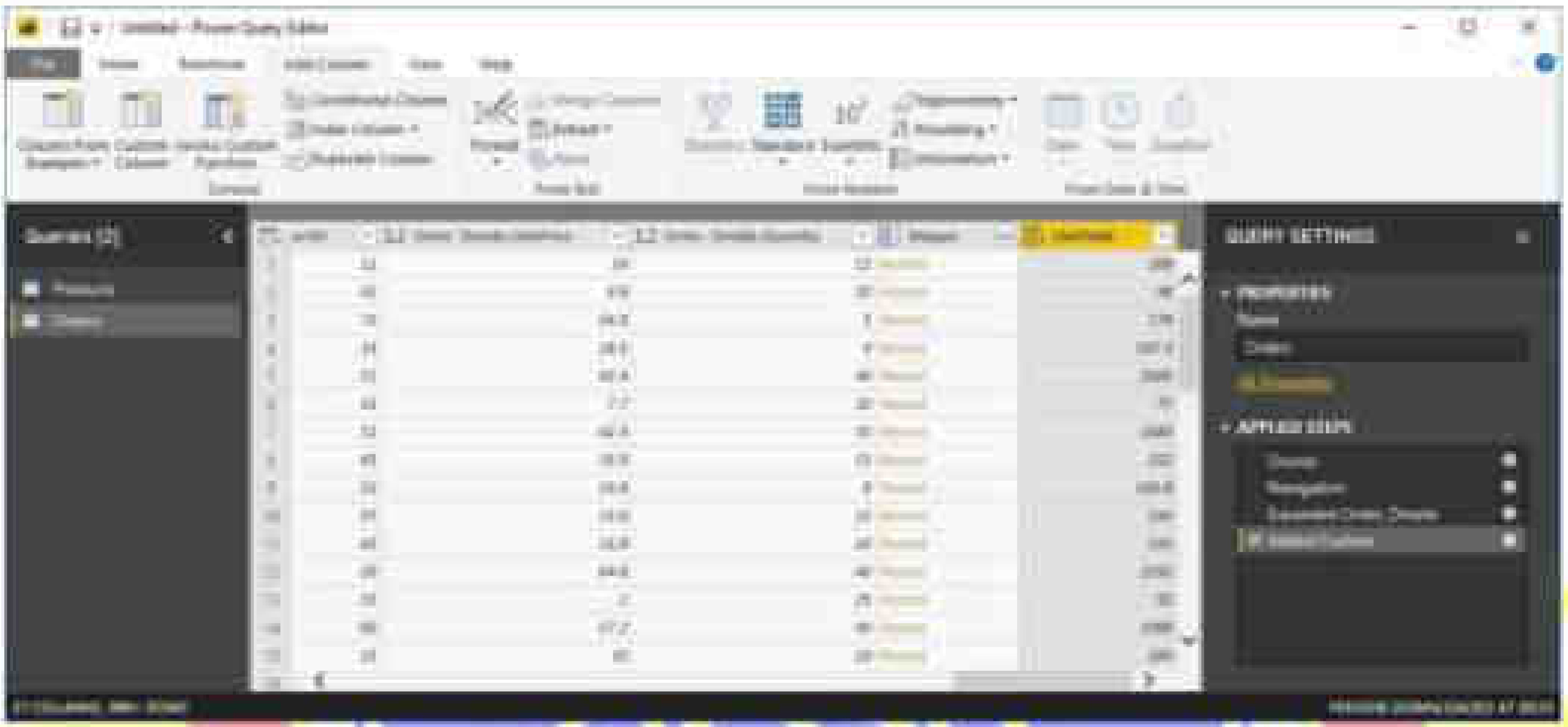
LineTotal

Custom column formula:

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

Available columns:

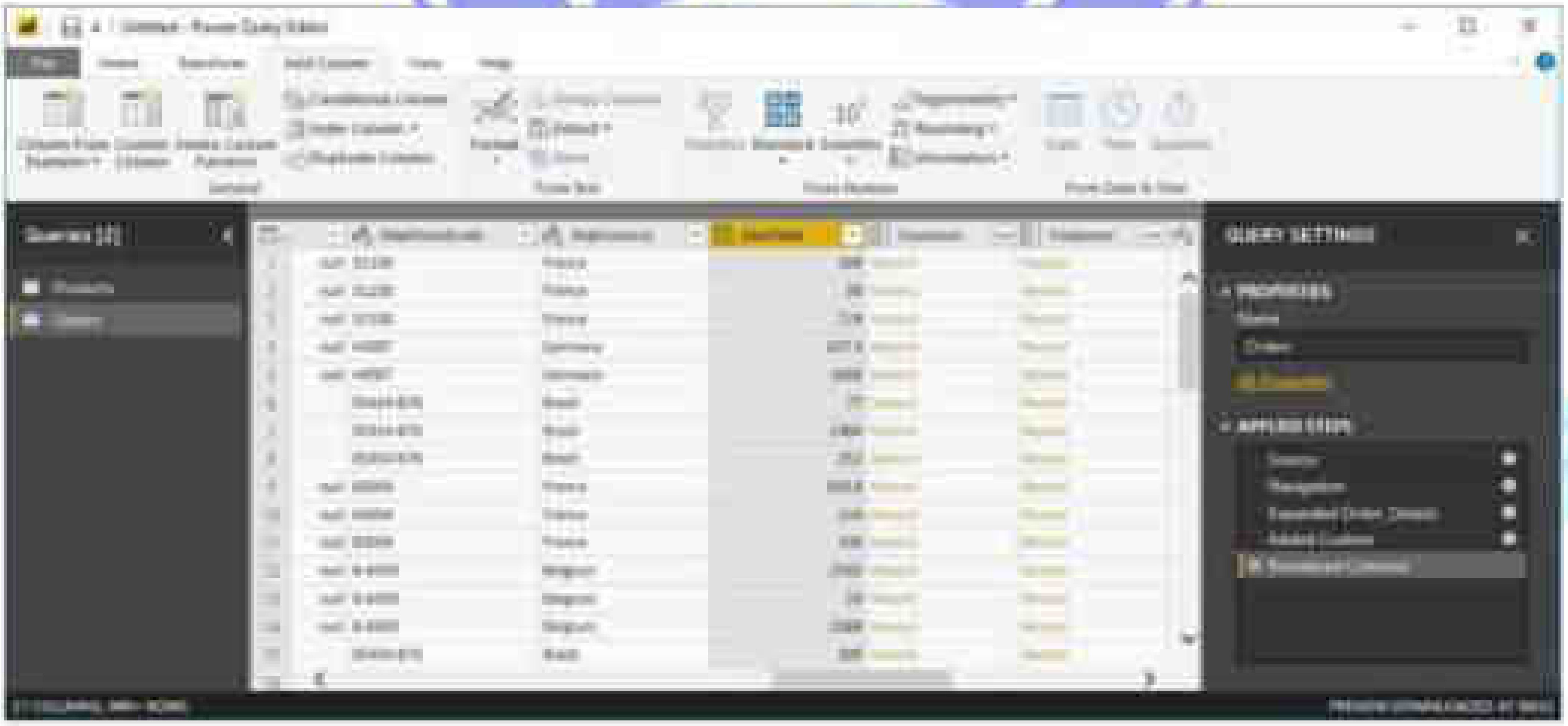
✓ No syntax errors have been detected.



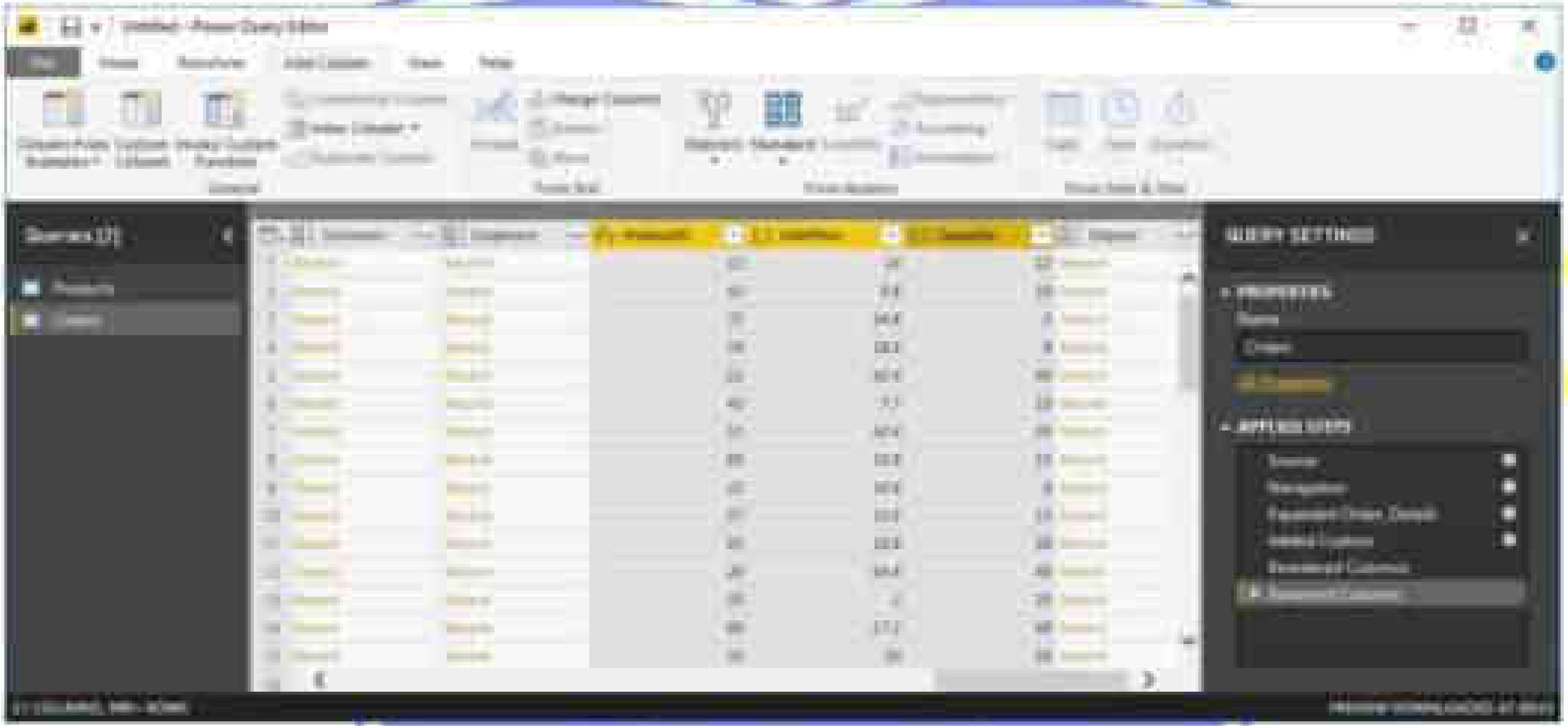
5. Rename and reorder columns in the query

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

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



b) 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.



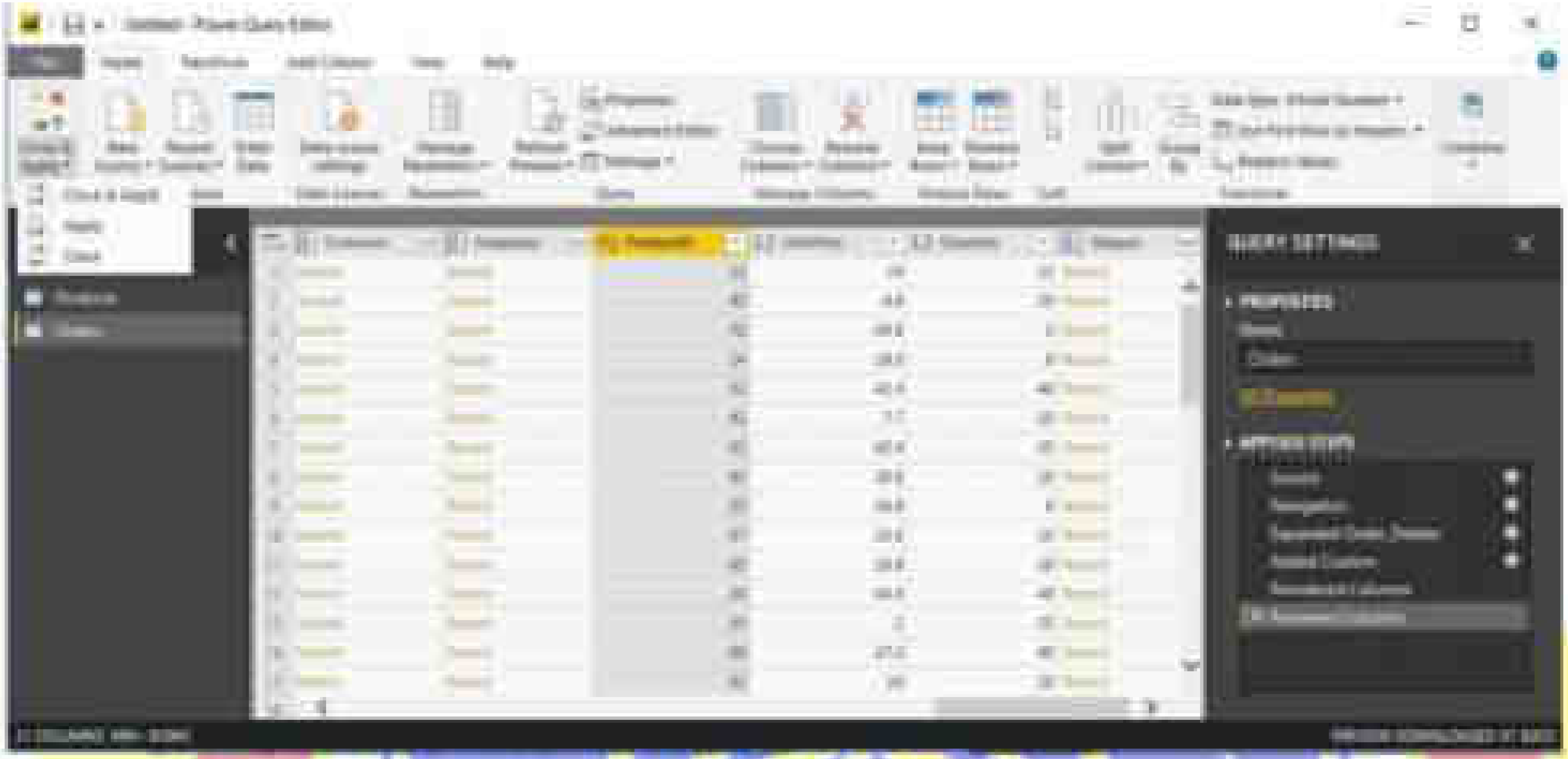
6. Combine the Products and Total Sales queries

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.

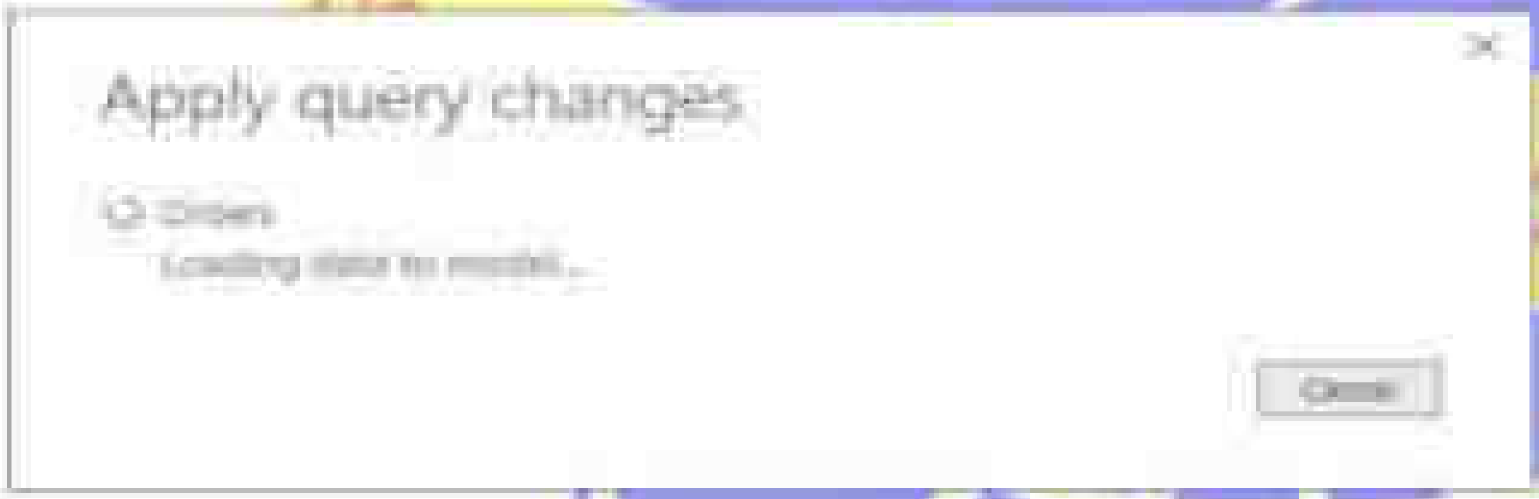
We have Orders and Products data that share a common 'ProductID' field, so we need to ensure there's a relationship between them in the model we're using with Power BI Desktop. Simply specify in Power BI Desktop that the columns from each table are related (i.e. columns that have the same values). Power 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 & Apply.



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

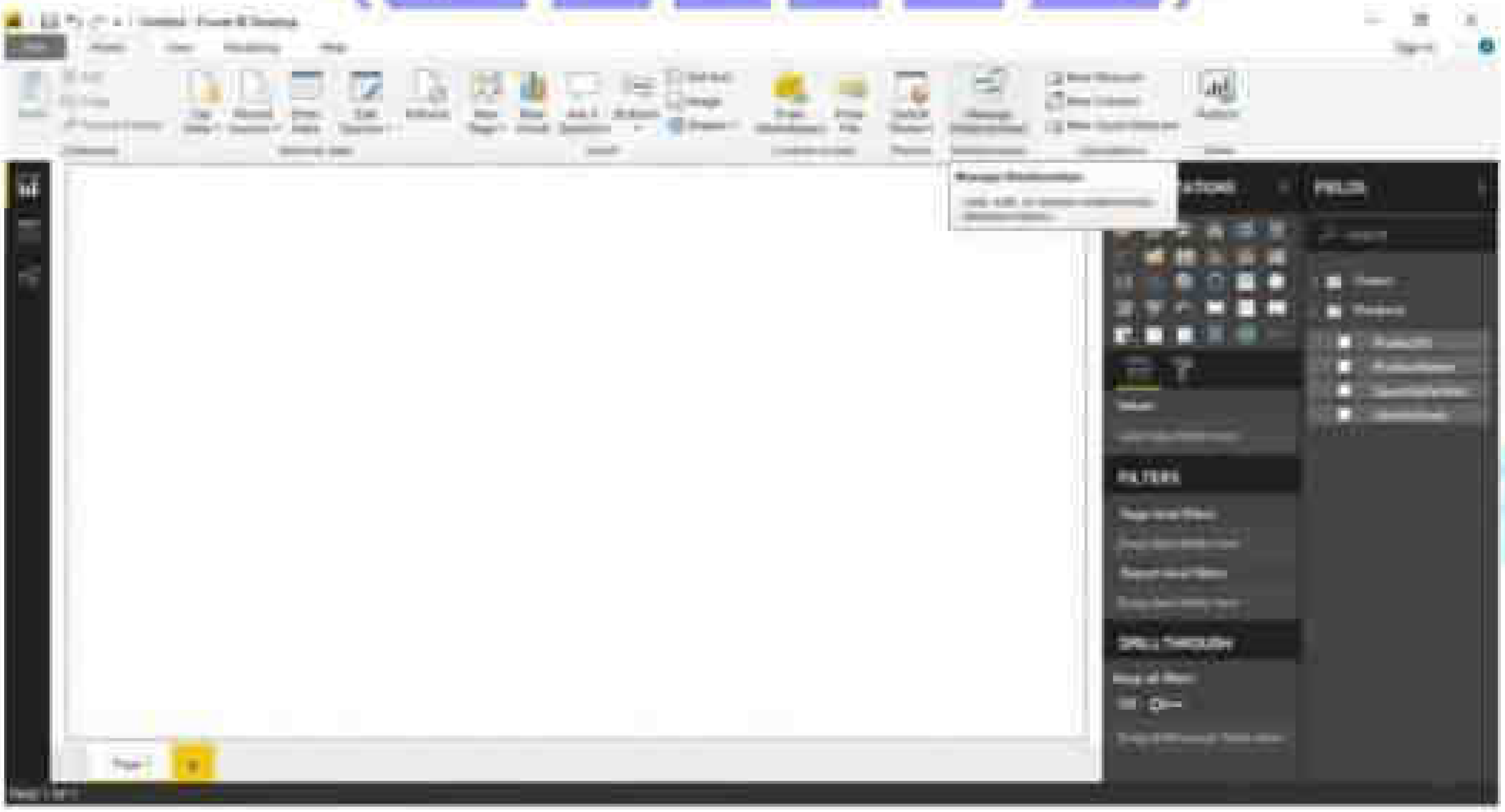


Apply query changes

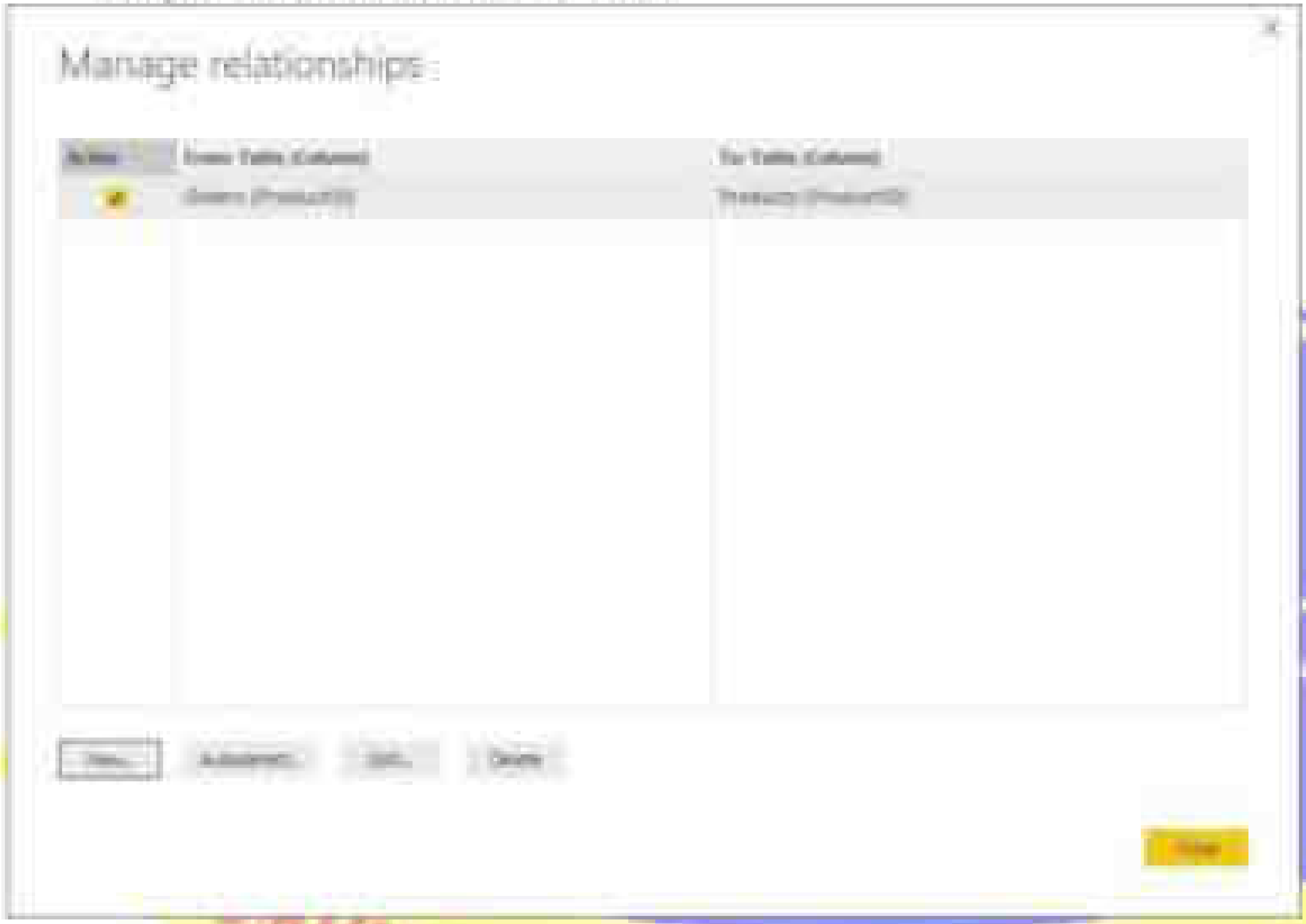
Orders
Loading data to model...

Cancel

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



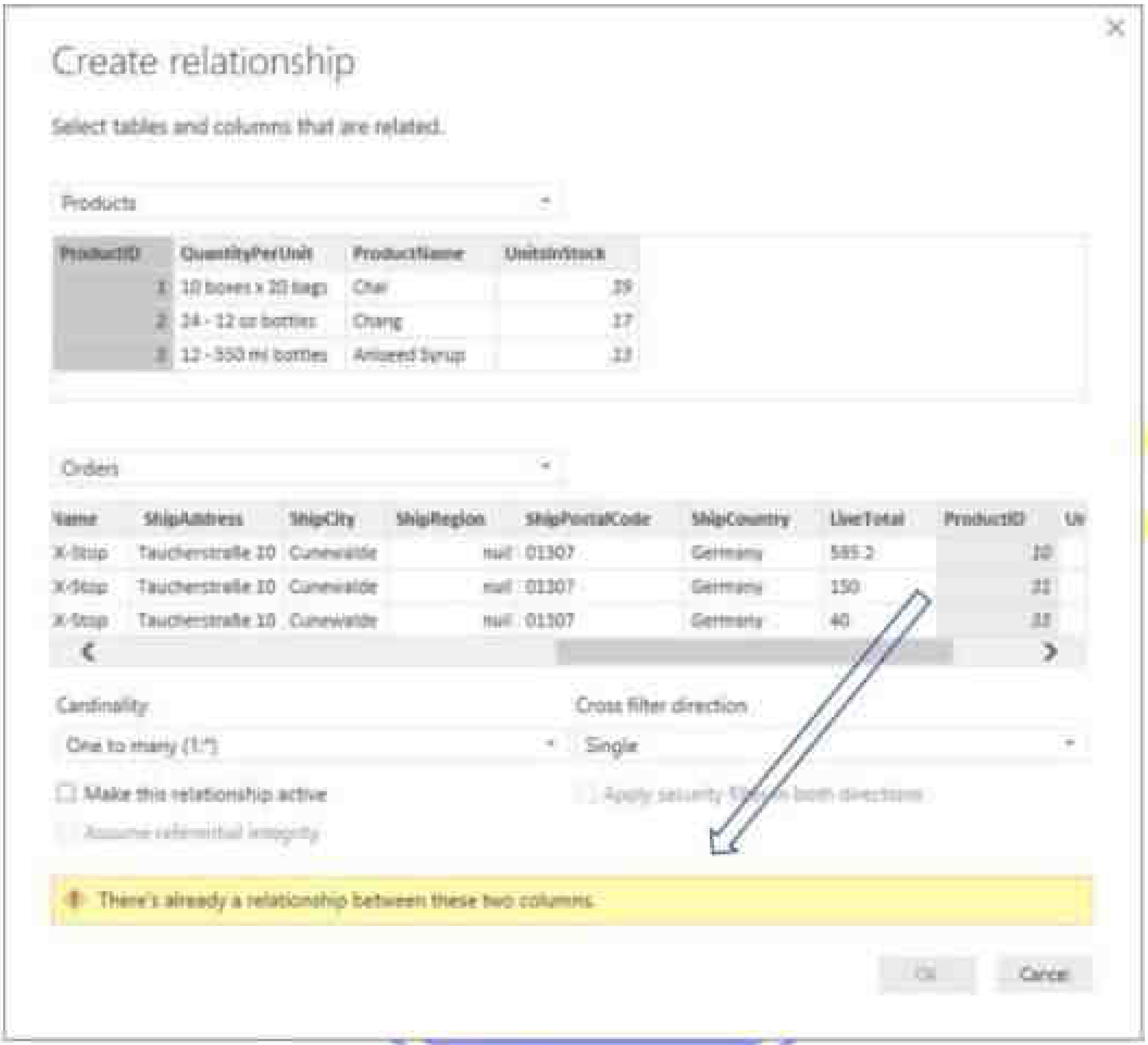
Step 4. Select the New... button



Manage relationships

Active	From: Table (Column)	To: Table (Column)
✓	Orders (ProductID)	Products (ProductID)

Step 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 are related.

Products

ProductID	QuantityPerUnit	ProductName	UnitsInStock
1	10 boxes x 20 bags	Chai	39
2	24 - 12 oz bottles	Chang	17
3	12 - 550 ml bottles	Aniseed Syrup	13

Orders

Name	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	ShipCountry	LineTotal	ProductID	Un
X-Stop	Taucherstraße 10	Cunewalde	null	01307	Germany	585.2	10	
X-Stop	Taucherstraße 10	Cunewalde	null	01307	Germany	150	22	
X-Stop	Taucherstraße 10	Cunewalde	null	01307	Germany	40	22	

Cardinality: One to many (1:*)

Cross filter direction: Single

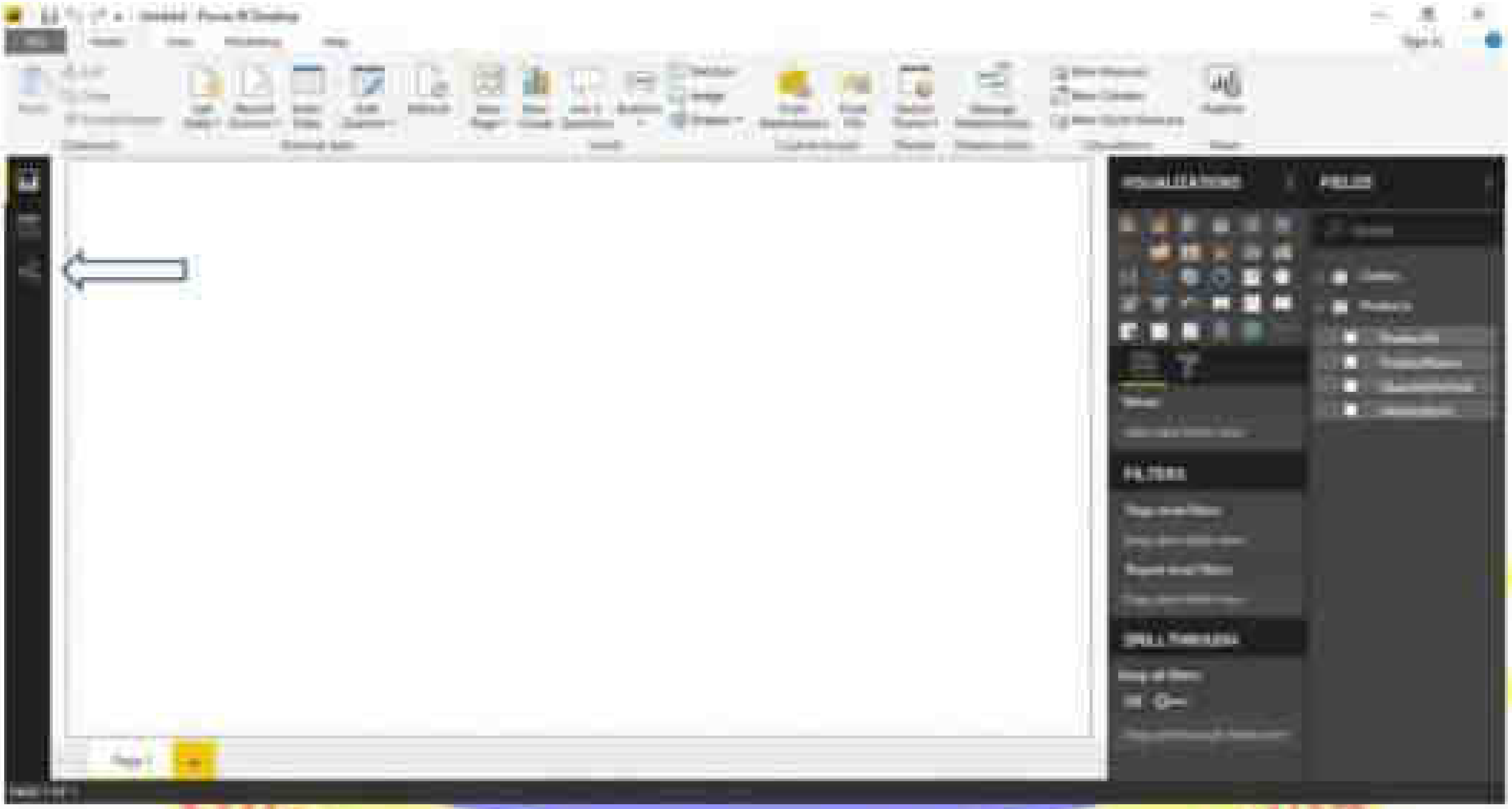
□ Make this relationship active

□ Apply security filter in both directions

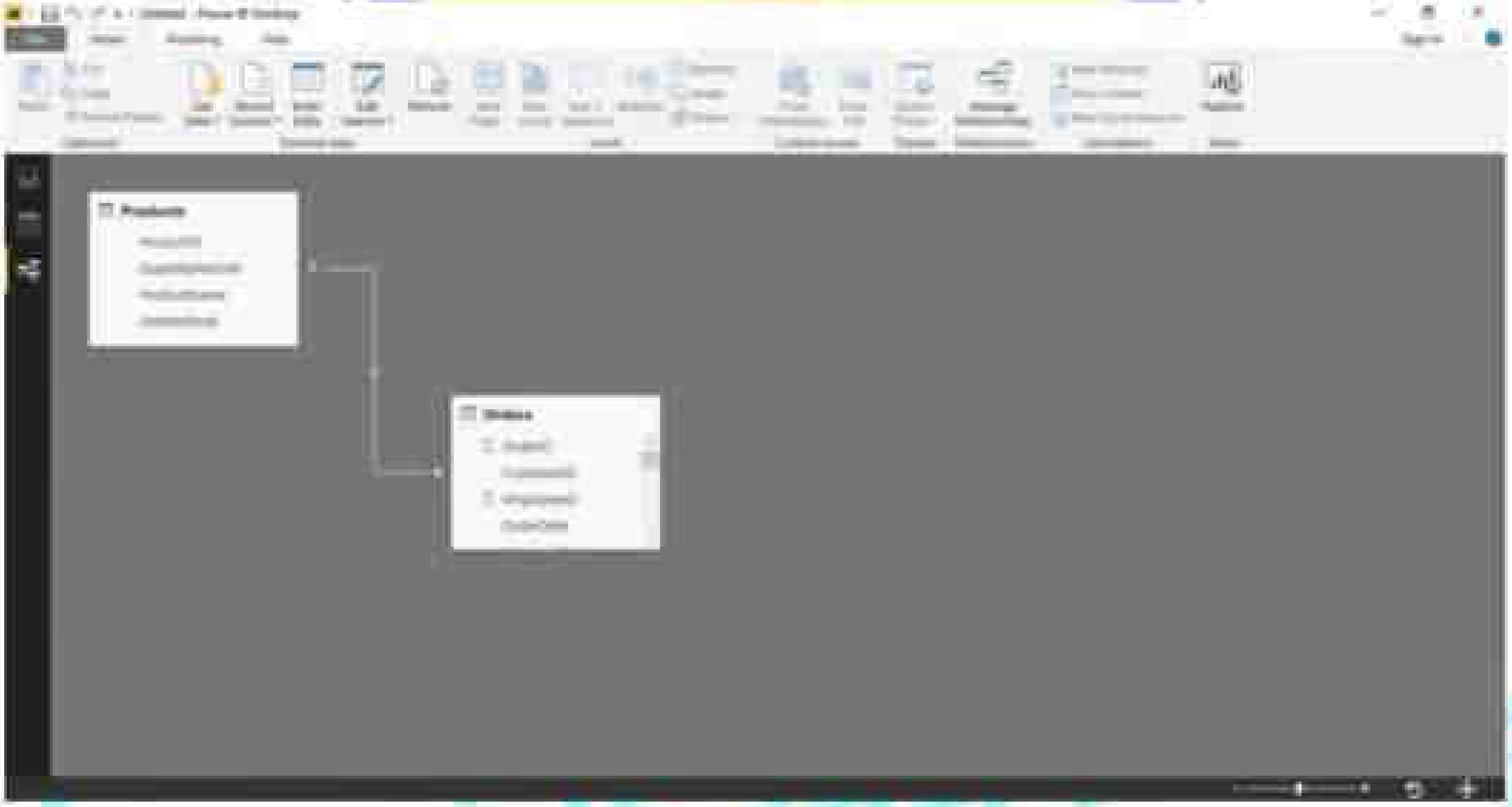
□ Assume referential integrity

There's already a relationship between these two columns.

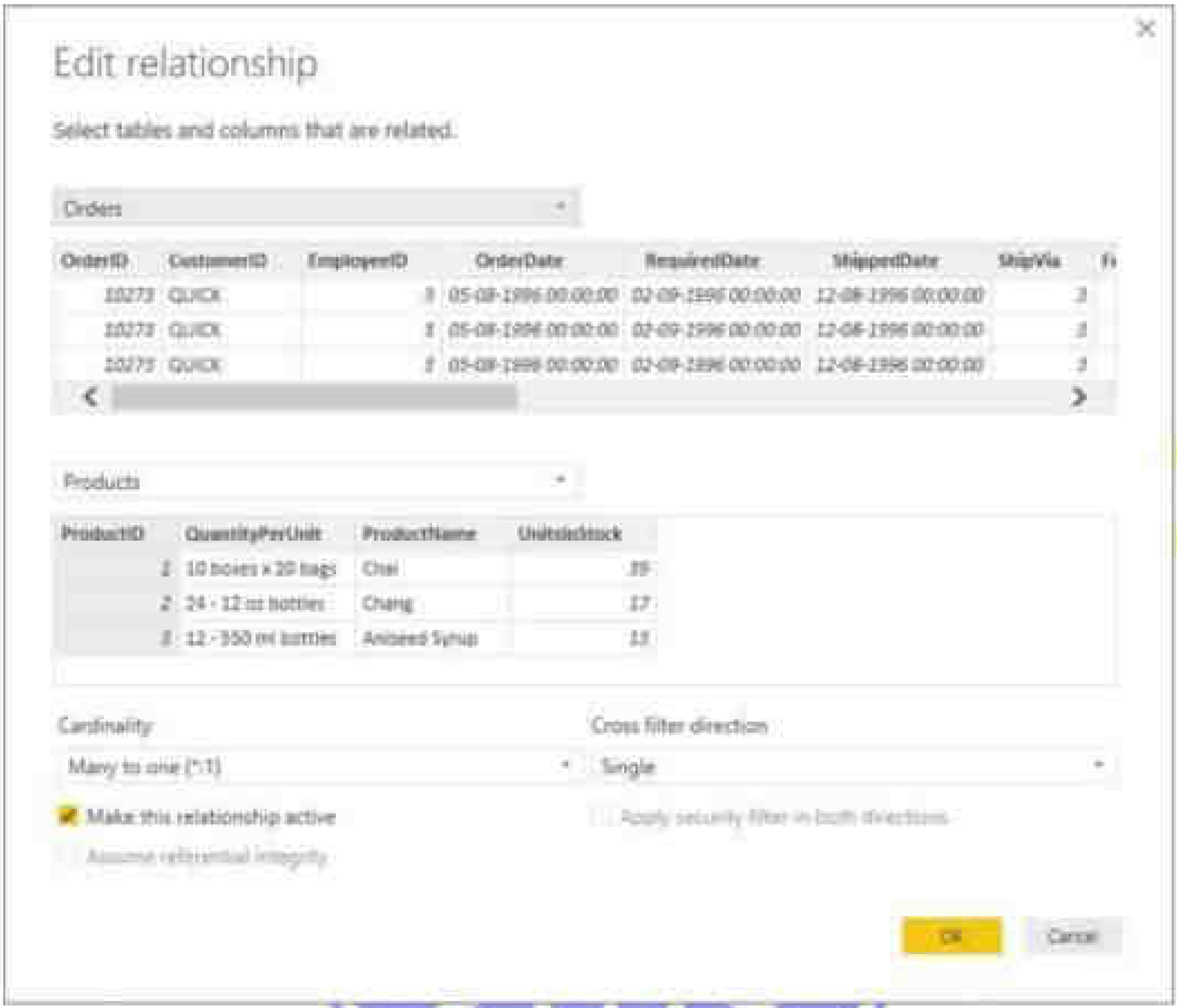
Step 6: Select Cancel, and then select Relationship view in Power BI Desktop.



Step 7: We see the following, which visualizes the relationship between the queries.



Step 8: 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 are related.

Orders

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Fr
10273	QUICK	3	05-08-1996 00:00:00	02-09-1996 00:00:00	12-08-1996 00:00:00	3	
10273	QUICK	3	05-08-1996 00:00:00	02-09-1996 00:00:00	12-08-1996 00:00:00	3	
10273	QUICK	3	05-08-1996 00:00:00	02-09-1996 00:00:00	12-08-1996 00:00:00	3	

Products

ProductID	QuantityPerUnit	ProductName	UnitsInStock
1	10 boxes x 20 bags	Chai	39
2	24 - 12 oz bottles	Chang	17
3	12 - 550 ml bottles	Aniseed Syrup	13

Cardinality: Many to one (*:1)

Cross filter direction: Single

☑ Make this relationship active

□ Apply security filter in both directions

□ Assume referential integrity

Step 9: No need to make any changes, so we'll just select Cancel to close the Edit Relationship dialog.

Learning Outcomes:

Course Outcomes:

Conclusion:

Viva Questions:

1. **What is extraction?**
2. **Name an extraction tool.**
3. **What is incremental extraction?**
4. **Why is extraction important?**

For Faculty use:

Correction Parameters	Formative Assessment[40%]	Timely Completion of Practical[40%]	Attendance Learning Attitude[20%]

Practical-3: a. Create the Data staging area for the selected database.
b. Create the cube with suitable dimension and fact tables based on ROLAP, MOLAP and HOLAP model.

Aims:

1. To create a dedicated data staging area for the selected database, enabling efficient data cleansing, transformation, and preparation for analysis.
2. To design and construct a multidimensional cube with appropriate dimension and fact tables using ROLAP, MOLAP, and HOLAP models for enhanced data analysis.

Learning Objectives:

1. Understand the role of data staging in the ETL and data warehousing process.
2. Gain hands-on experience in setting up a staging area to extract, clean, and transform raw data.
3. Learn the fundamentals of cube design, including the creation of dimension and fact tables.
4. Explore the differences and use cases for ROLAP, MOLAP, and HOLAP models in multidimensional analysis.

Tool & Technologies used:

1. Power BI is a powerful business intelligence tool used for data visualization and analysis.

Theory-3: Data staging

Data staging is a crucial phase in data warehousing where extracted data is temporarily stored, cleansed, and transformed before being loaded into the target database. This ensures data quality, integrity, and optimized performance for analytical processing.

Creating a Data Staging Area

1. **Database Selection:** Choose a suitable database system (e.g., SQL Server, Oracle) to store the staged data.
2. **Schema Design:** Define staging tables to hold raw, intermediate, and transformed data.
3. **Data Loading:** Extracted data from multiple sources is loaded into the staging area for preprocessing.
4. **Data Cleansing:** Perform data validation, deduplication, and format standardization.
5. **Transformation:** Apply necessary business rules, data aggregation, and normalization before transferring data to the data warehouse.

Creating a Cube with Suitable Dimensions and Fact Tables

1. **Choosing the OLAP Model:**
   - **ROLAP (Relational OLAP):** Stores data in relational databases and processes queries dynamically.
   - **MOLAP (Multidimensional OLAP):** Stores pre-aggregated data in multidimensional cubes for faster access.
2. **Defining Fact Tables:**
   - Contains measurable business data (e.g., sales, revenue, transaction count).
   - Linked with dimension tables using foreign keys.
3. **Defining Dimension Tables:**
   - Stores descriptive attributes (e.g., time, product, customer, location).
   - Supports data slicing, dicing, and drill-down operations.
4. **Building the Cube:**
   - Organize fact and dimension tables within the chosen OLAP model.
   - Precompute aggregations to enhance query performance.

PRACTICAL 3 b

Create the cube with suitable dimension and fact tables based on

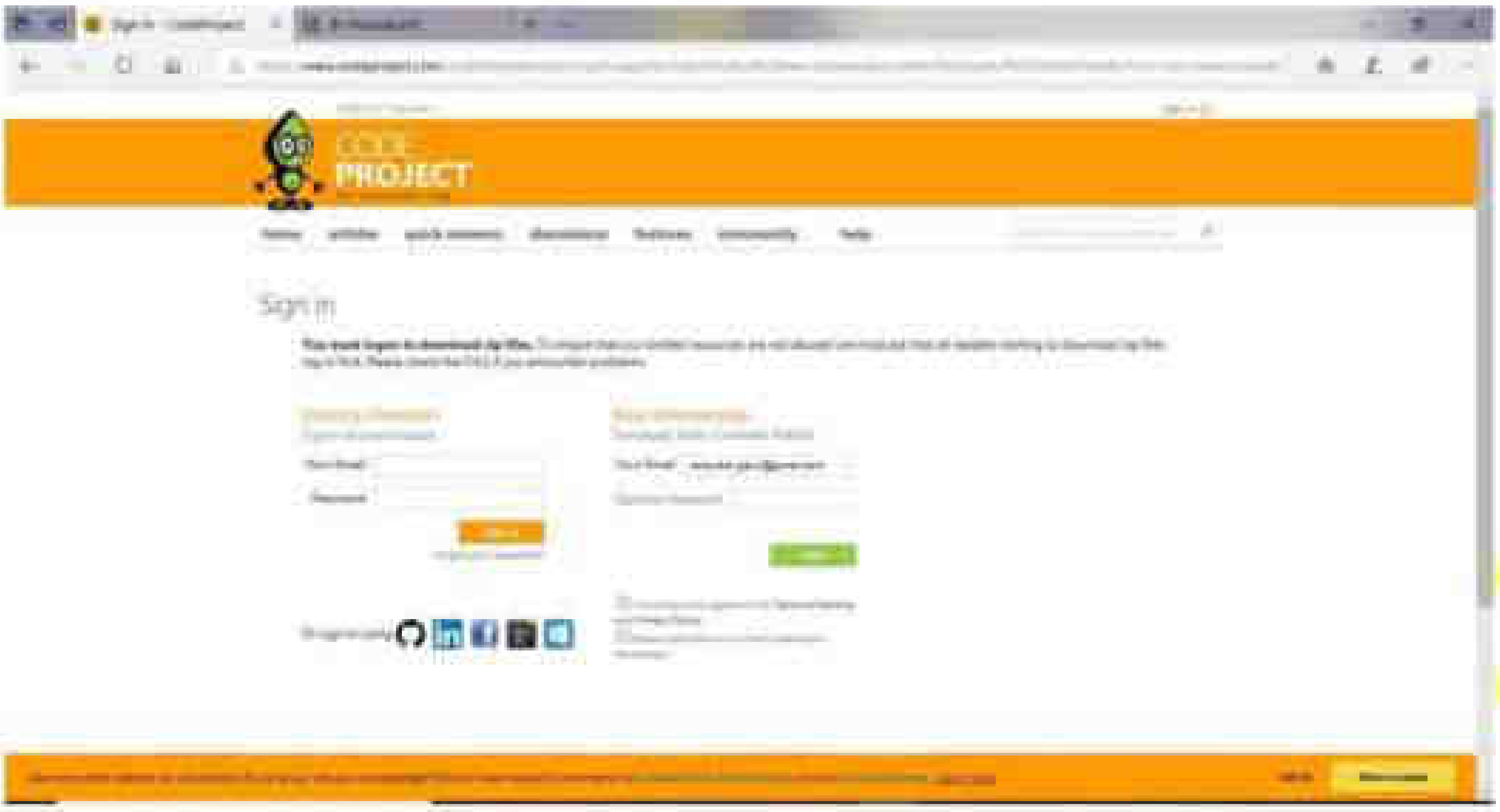
OLAP

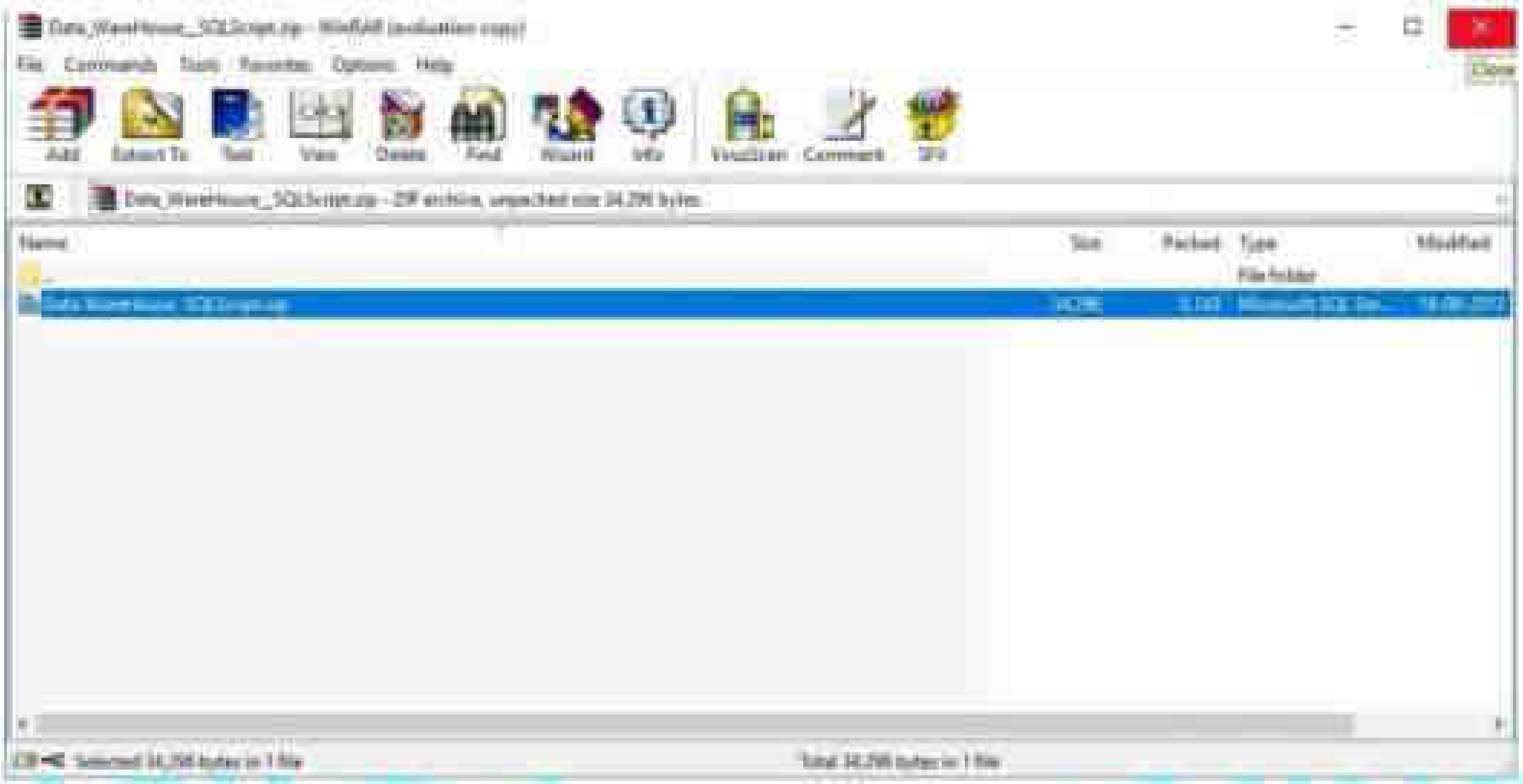
Step 1: Creating Data Warehouse

Let us execute our T-SQL Script to create data warehouse with fact tables, dimensions and populate them with appropriate test values.

Download T-SQL script attached with this article for creation of Sales Data Warehouse or download from this article "Create First Data Warehouse" and run it in your SQL Server.

Downloading "Data_WareHouse__SQLScript.zip" from the article
https://www.codeproject.com/Articles/652108/Create-First-Data-WareHouse





Data_WareHouse_SQLScript.zip - WinRAR (evaluation copy)

Data_WareHouse_SQLScript.zip - ZIP archive, unpacked size 34,298 bytes

Name	Size	Packed	Type	Modified
..			File folder	
Data_WareHouse_SQLScript.sql	34,298	5,133	Microsoft SQL Ser...	18-08-2012

Selected 34,298 bytes in 1 file

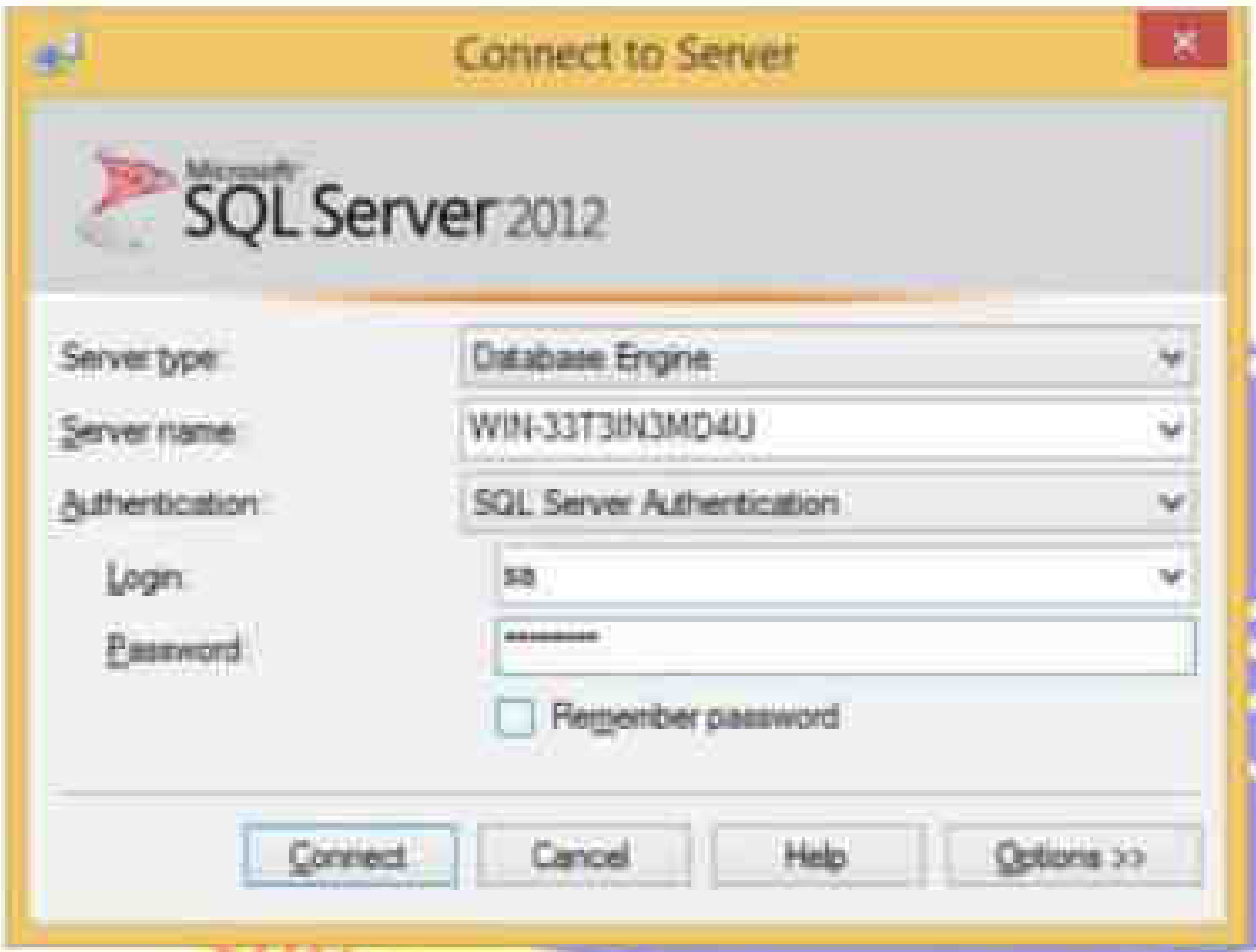
Total 34,298 bytes in 1 file

After downloading extract file in folder.

Follow the given steps to run the query in SSMS (SQL Server Management Studio).

1. Open SQL Server Management Studio 2012

2. Connect Database Engine



Server type: Database Engine

Server name: WIN-33T3IN3MD4U

Authentication: SQL Server Authentication

Login: sa

Password: ********

□ Remember password

Password for sa : admin123 (as given during installation)

Click Connect.

3. Open New Query editor

4. Copy paste Scripts given below in various steps in new query editor window one by one

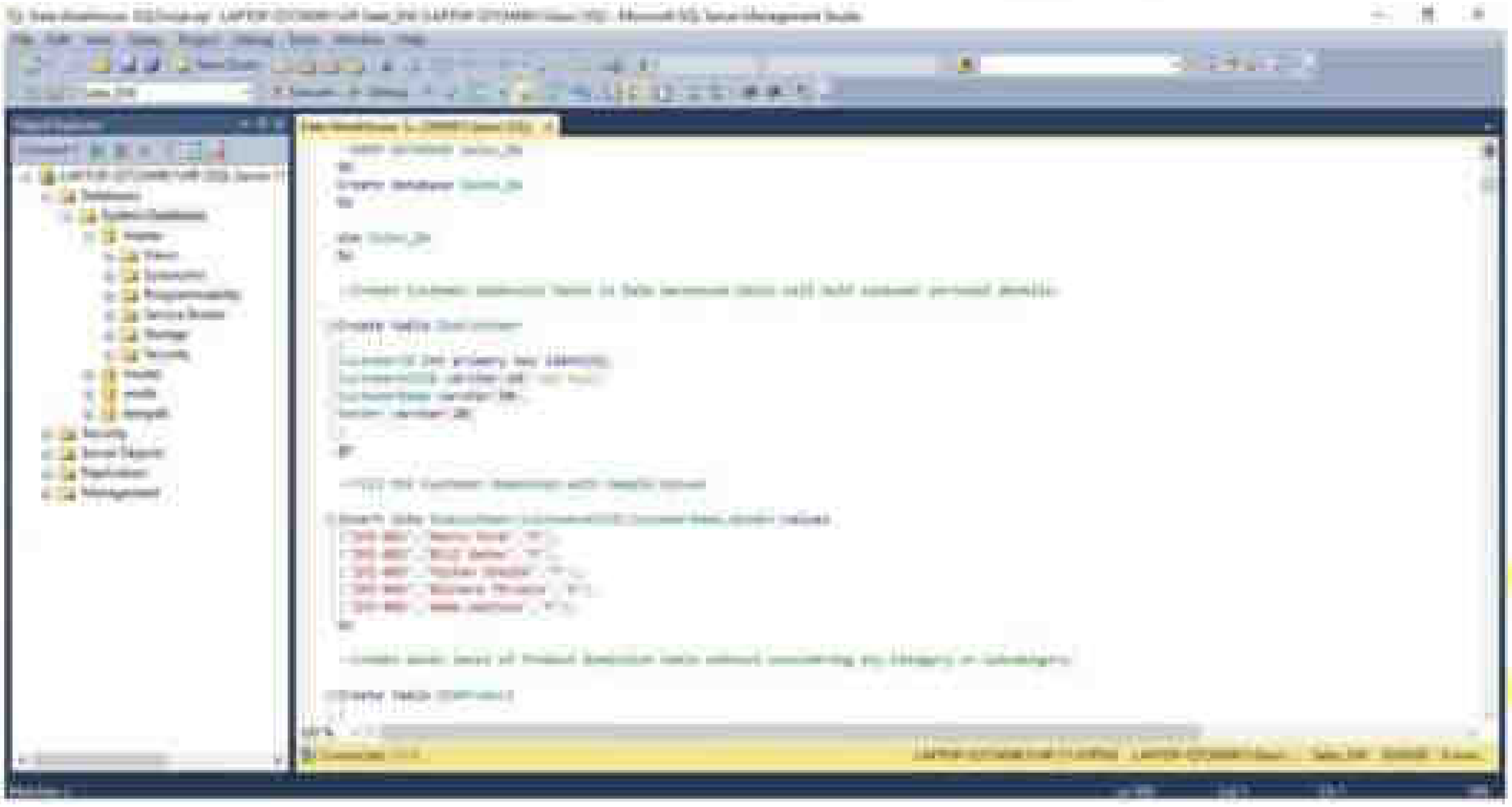
5. To run the given SQL Script, press F5

6. It will create and populate "Sales_DW" database on your SQL Server

OR

1. Go to the extracted sql file and double click on it.

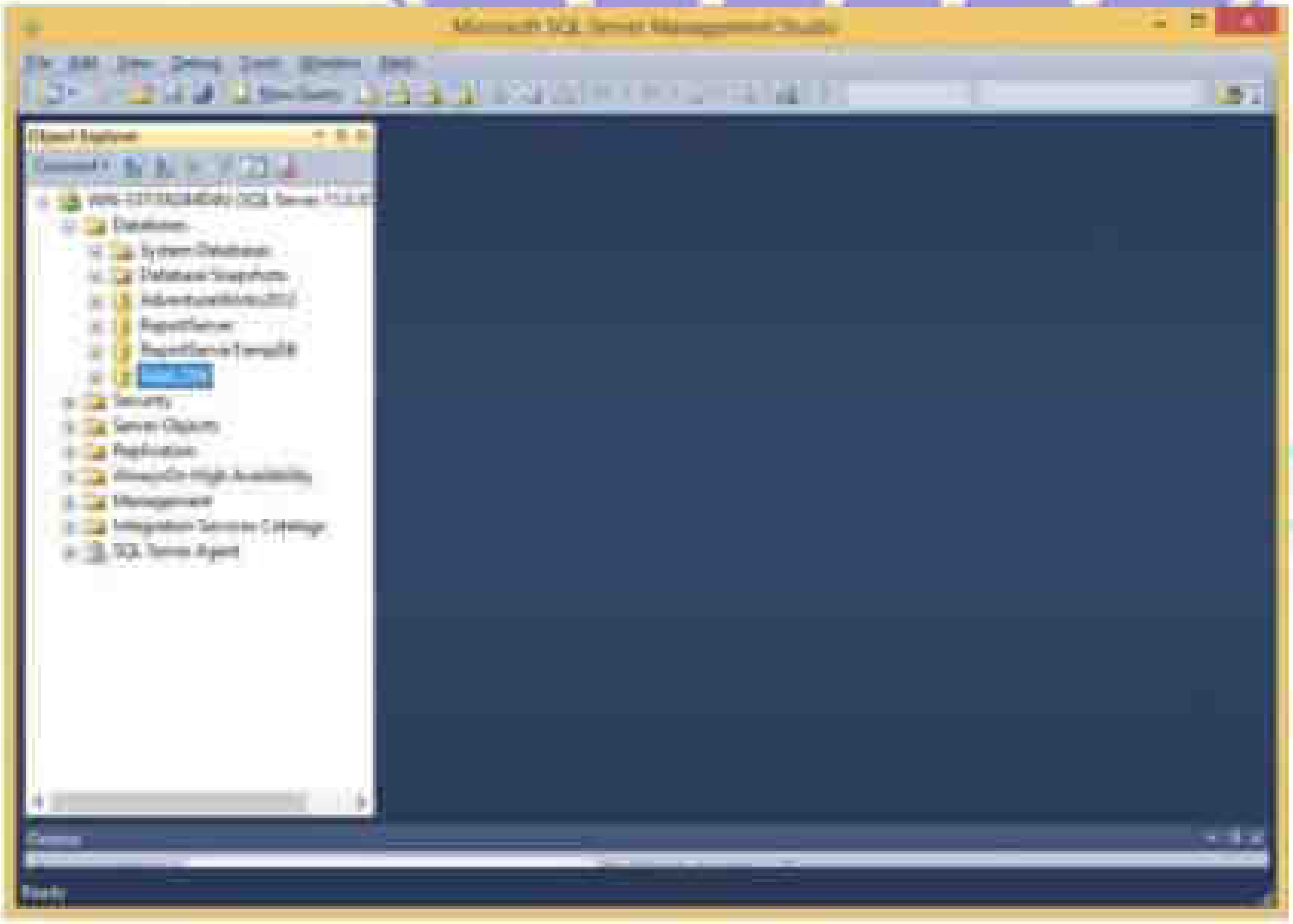
2. New Sql Query Editor will be opened containing Sales_DW Database.



```
--DROP DATABASE Sales_DW
Go
Create database Sales_DW
Go

Use Sales_DW
Go

--Create Customer dimension table in Data Warehouse which will hold customer personal details.

Create table DimCustomer
(
CustomerID int primary key identity,
CustomerAltID varchar(10) not null,
CustomerName varchar(50),
Gender varchar(20)
)
go

--Fill the Customer dimension with sample Values

Insert into DimCustomer(CustomerAltID,CustomerName,Gender) values
('IMI-001','Henry Ford','M'),
('IMI-002','Bill Gates','M'),
('IMI-003','Muskan Shaikh','F'),
('IMI-004','Richard Thrubin','M'),
('IMI-005','Emma Wattson','F');
Go

--Create basic level of Product Dimension table without considering any Category or Subcategory

Create table DimProduct
(
```

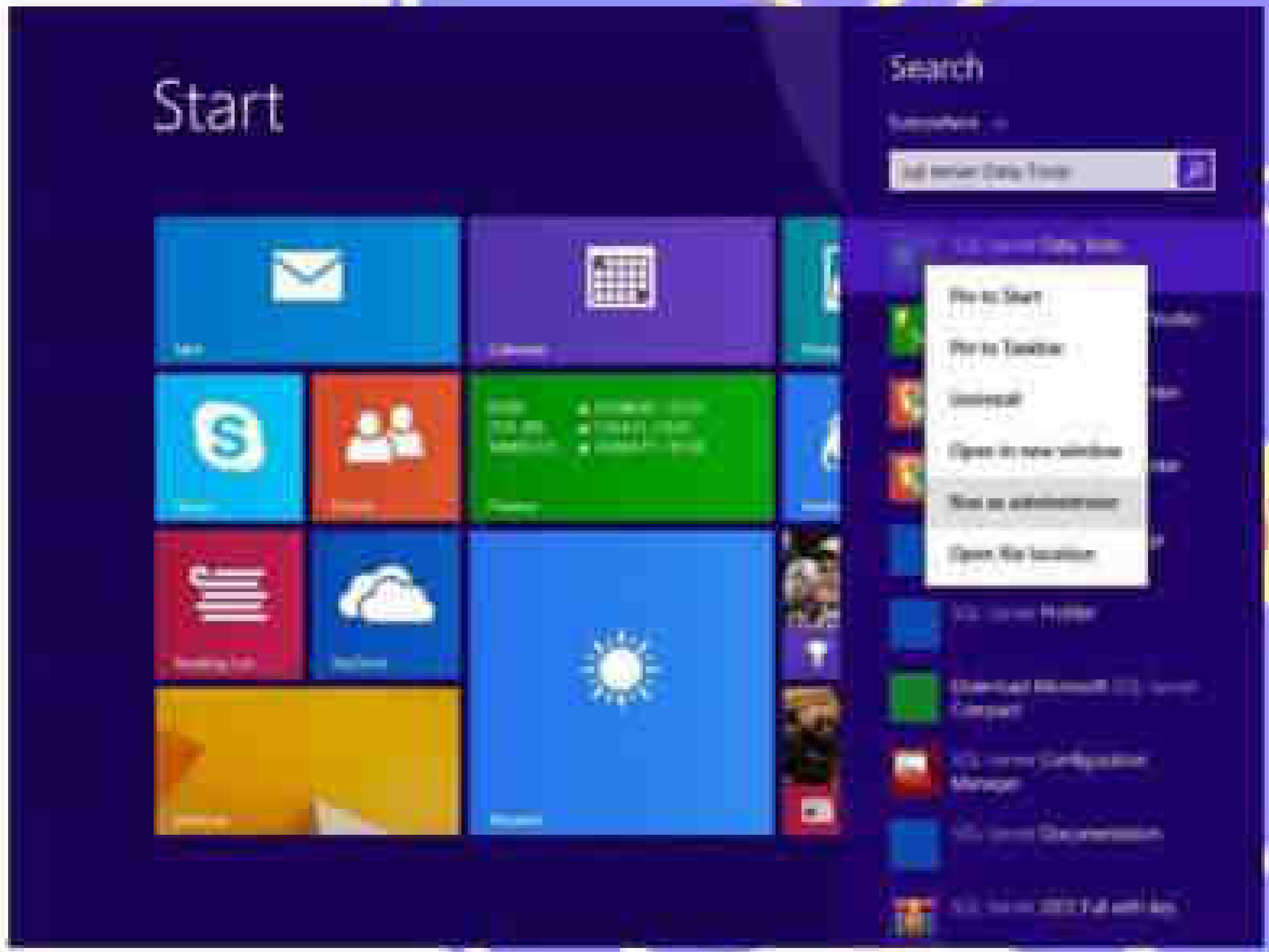
3. Click on execute or press F5 by selecting query one by one or directly click on Execute.

4. After completing execution save and close SQL Server Management studio & Reopen to see Sales_DW in Databases Tab.

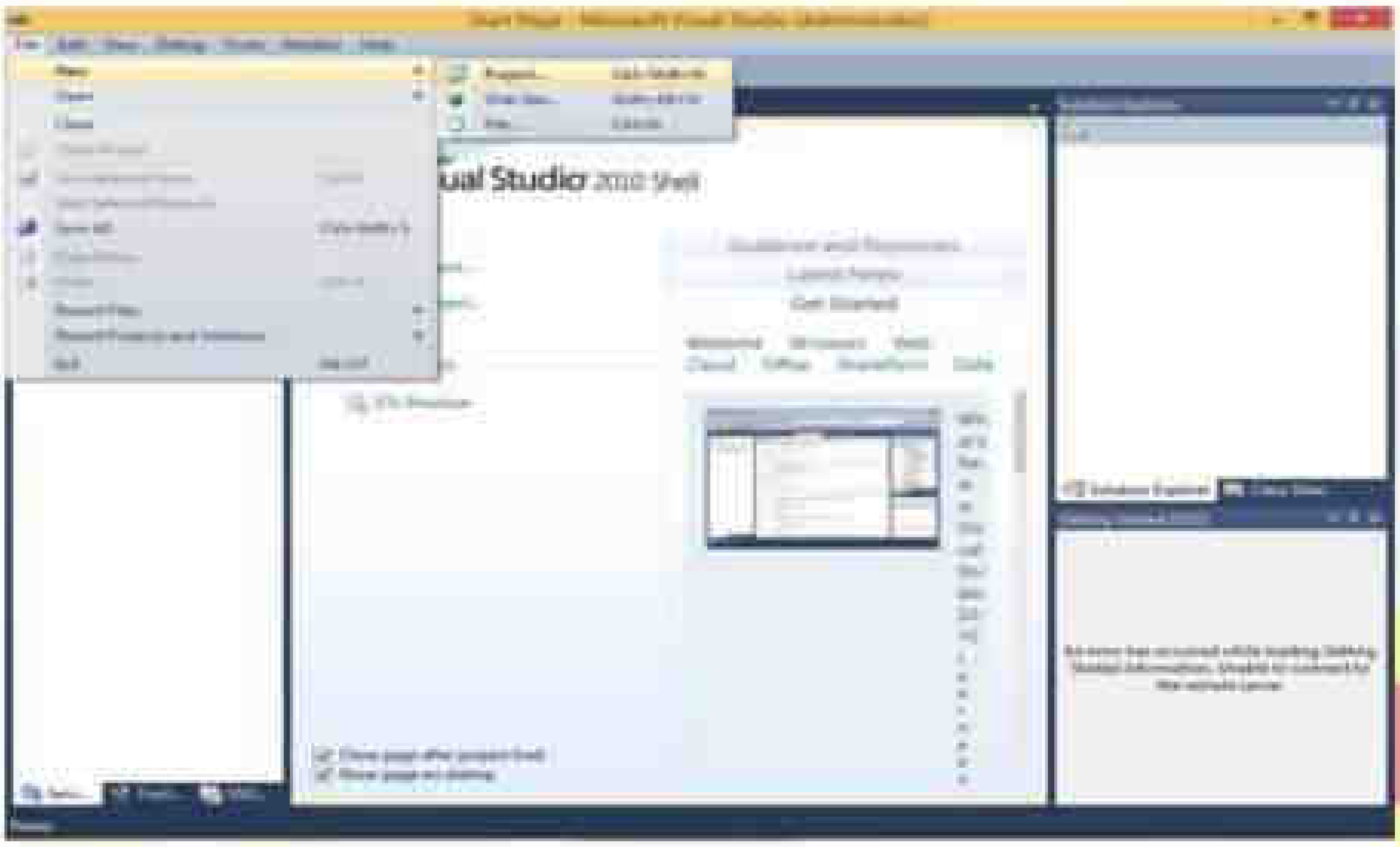


Step 2: Start SSDT environment and create New Data Source

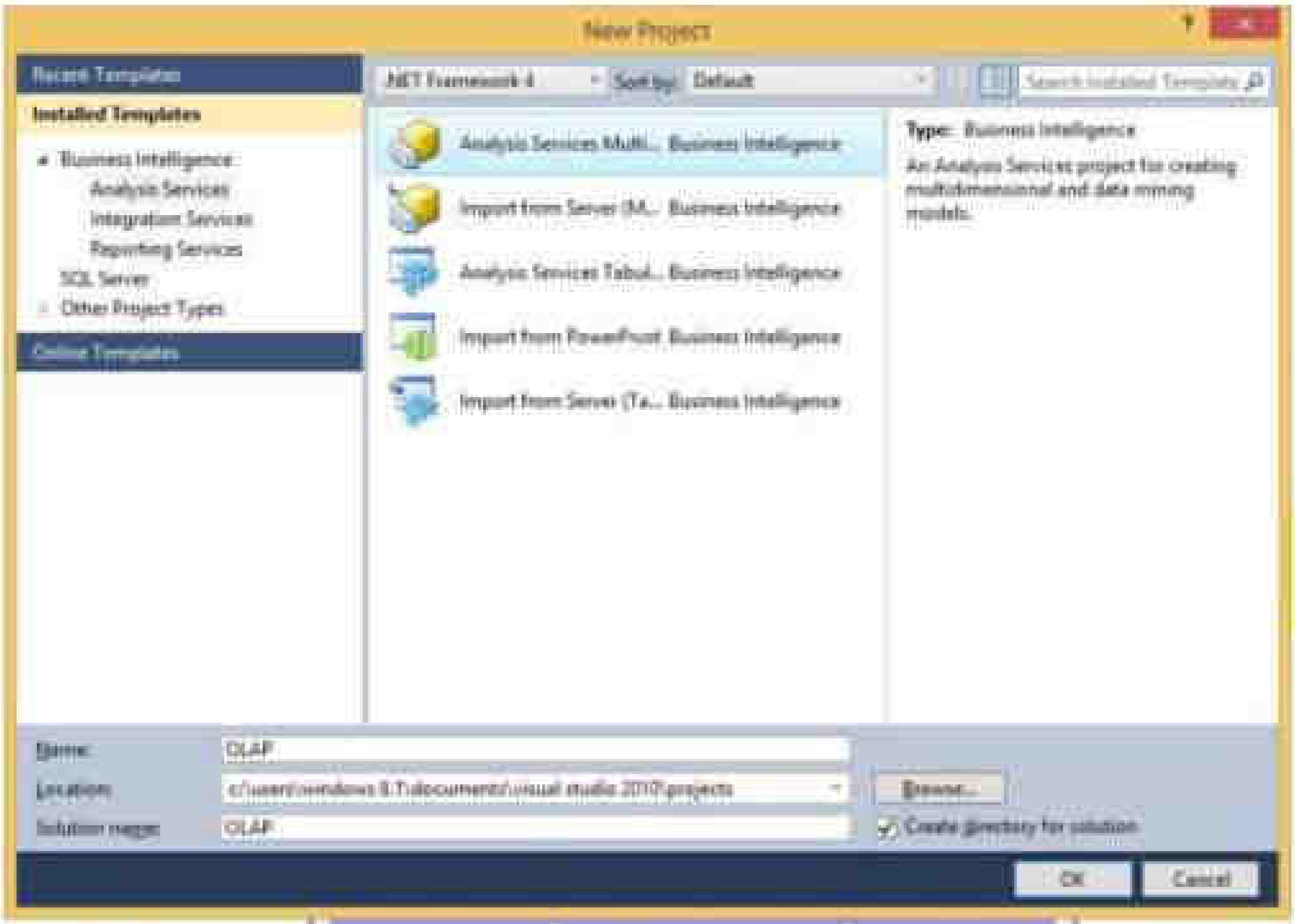
Go to Sql Server Data Tools --> Right click and run as administrator



Click on File → New → Project



In Business Intelligence → Analysis Services Multidimensional and Data Mining models → appropriate project name → click OK


New Project
Recent Templates
Installed Templates
Business Intelligence
Analysis Services
Integration Services
Reporting Services
SQL Server
Other Project Types
Online Templates
.NET Framework 4
Sort by: Default
Search Installed Templates
Analysis Services Multi... Business Intelligence
Import from Server (M... Business Intelligence
Analysis Services Tabul... Business Intelligence
Import from PowerPivot Business Intelligence
Import from Server (Ta... Business Intelligence
Browse...
OK
Cancel

Type: Business Intelligence

An Analysis Services project for creating multidimensional and data mining models.

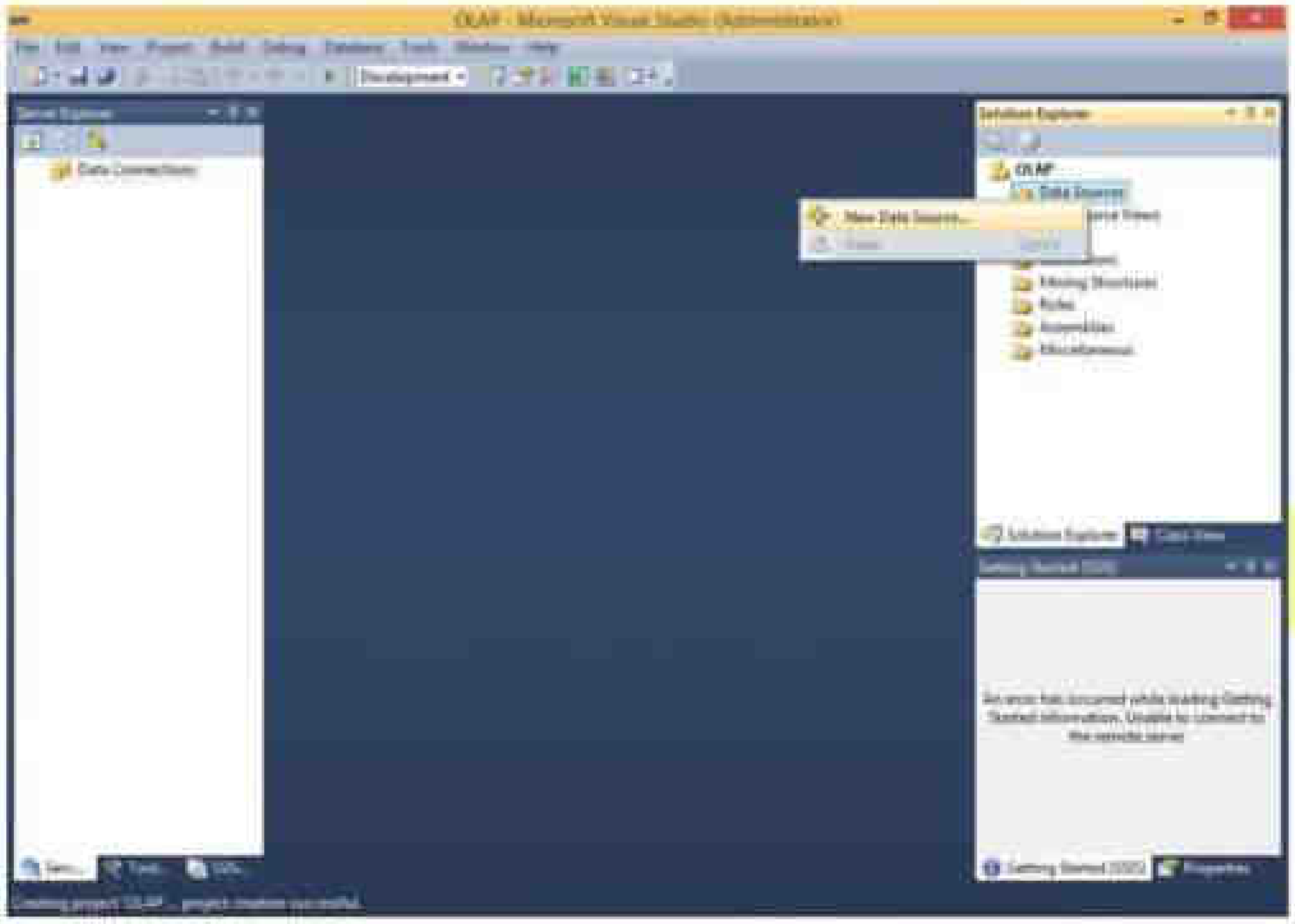
Name: OLAP

Location: c:\users\windows 8.1\documents\visual studio 2010\projects

Solution name: OLAP

☑ Create directory for solution

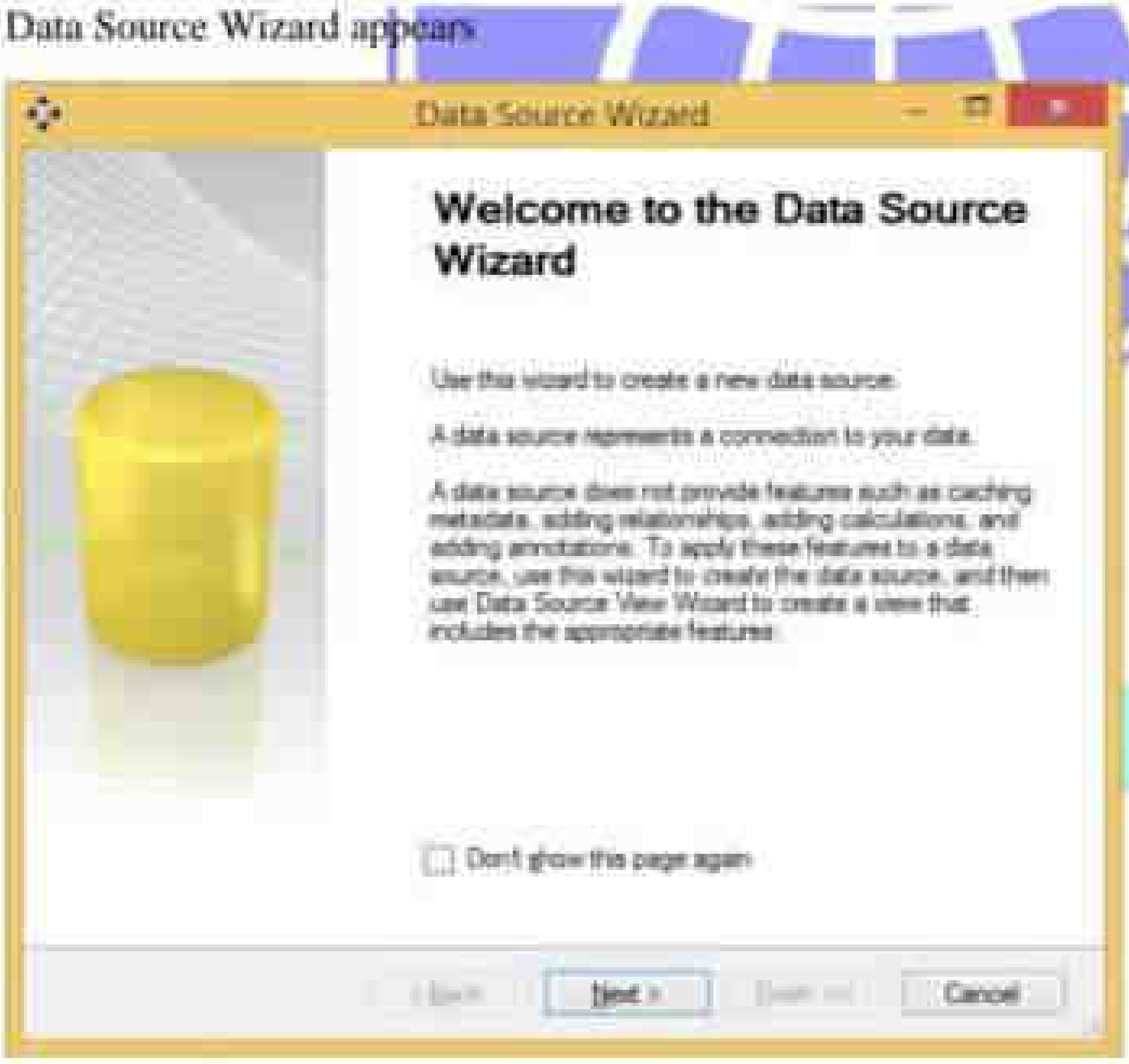
Right click on Data Sources in solution explorer → New Data Source



An error has occurred while loading Getting Started information. Unable to connect to the remote server.

Creating project 'OLAP'... project creation successful.

Data Source Wizard appears



Welcome to the Data Source Wizard

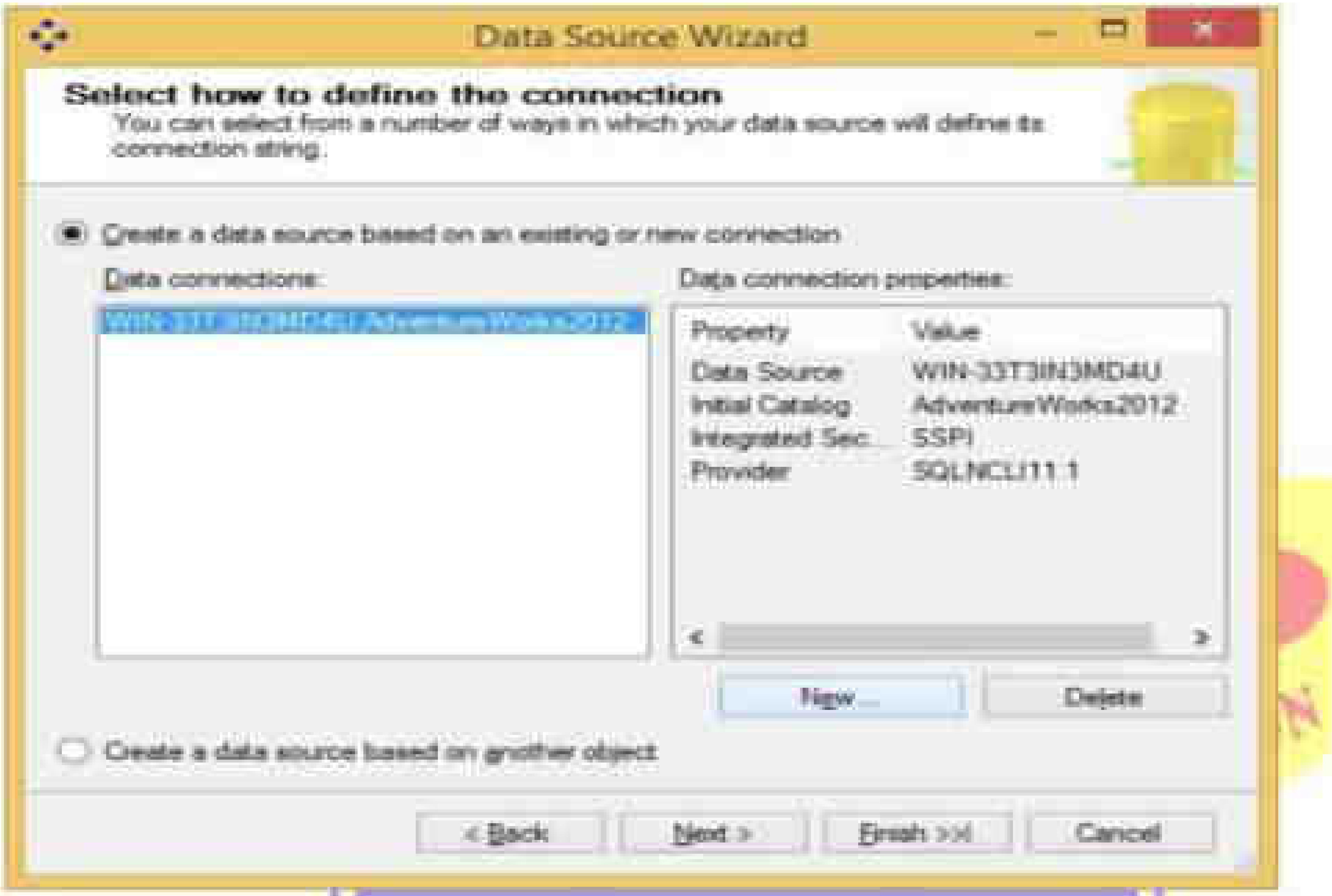
Use this wizard to create a new data source.

A data source represents a connection to your data.

A data source does not provide features such as caching metadata, adding relationships, adding calculations, and adding annotations. To apply these features to a data source, use this wizard to create the data source, and then use Data Source View Wizard to create a view that includes the appropriate features.

□ Don't show this page again

Click on New



Data Source Wizard

Select how to define the connection

You can select from a number of ways in which your data source will define its connection string.

● Create a data source based on an existing or new connection

Data connections:

WIN-33T3IN3MD4U.AdventureWorks2012

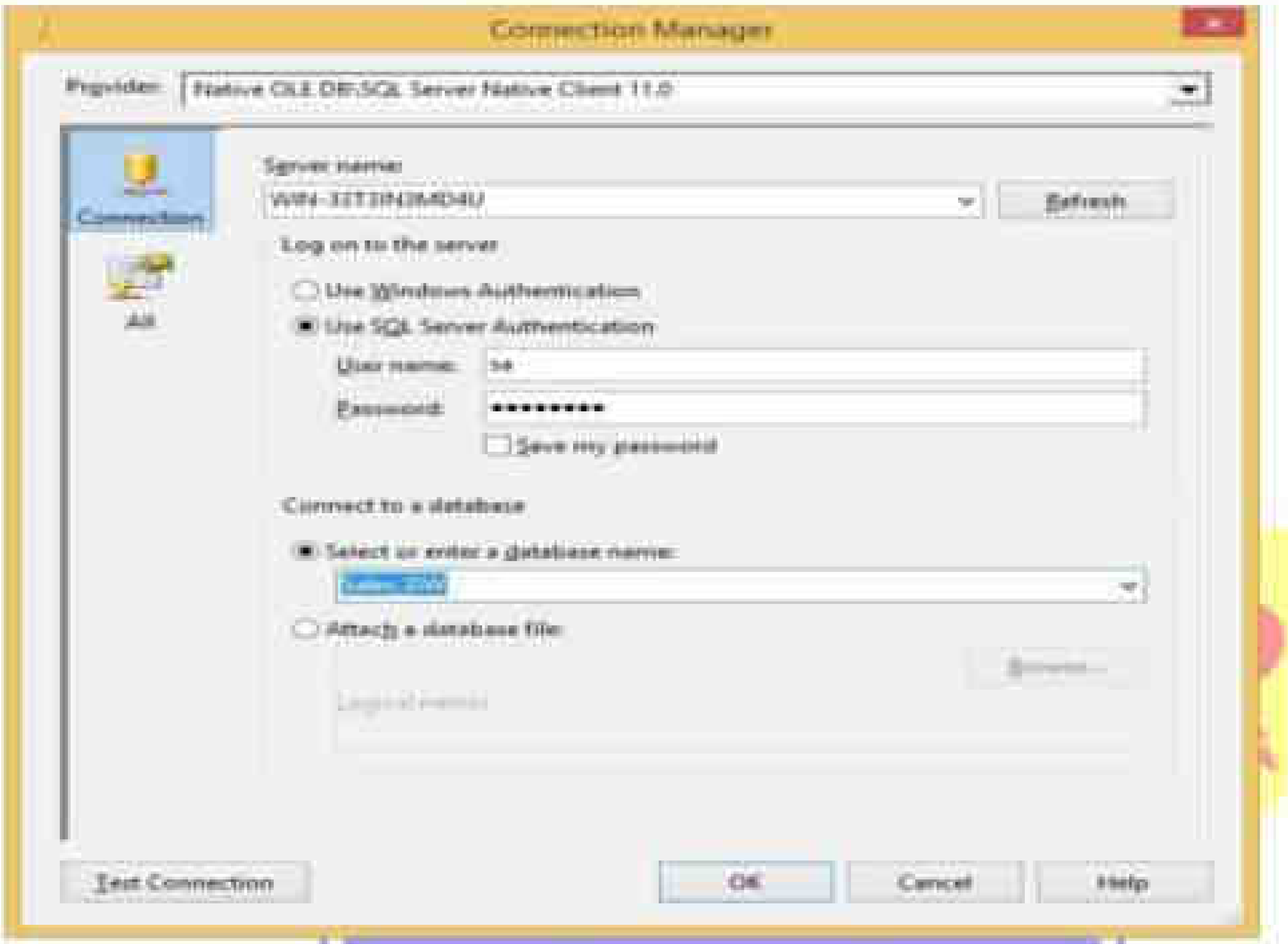
Data connection properties:

Property	Value
Data Source	WIN-33T3IN3MD4U
Initial Catalog	AdventureWorks2012
Integrated Sec...	SSPI
Provider	SQLNCLI11.1

○ Create a data source based on another object

Select Server Name → select Use SQL Server Authentication → Select or enter a database name (Sales_DW)

Note : Password for sa : admin123 (as given during installation of SQL 2012 full version)



Connection Manager

Provider: Native OLE DB\SQL Server Native Client 11.0

Server name: WIN-33T3IN3MD4U

Log on to the server

○ Use Windows Authentication

● Use SQL Server Authentication

User name: sa

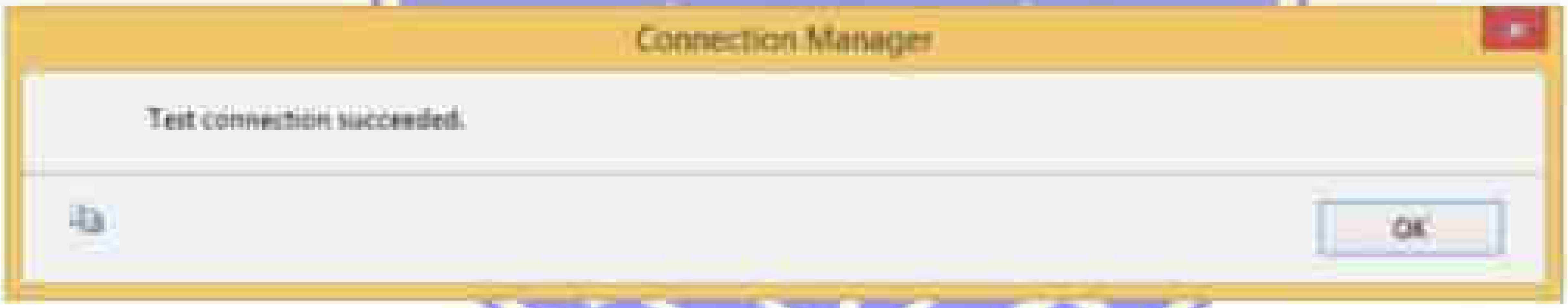
Password: •••••••••

□ Save my password

Connect to a database

● Select or enter a database name:

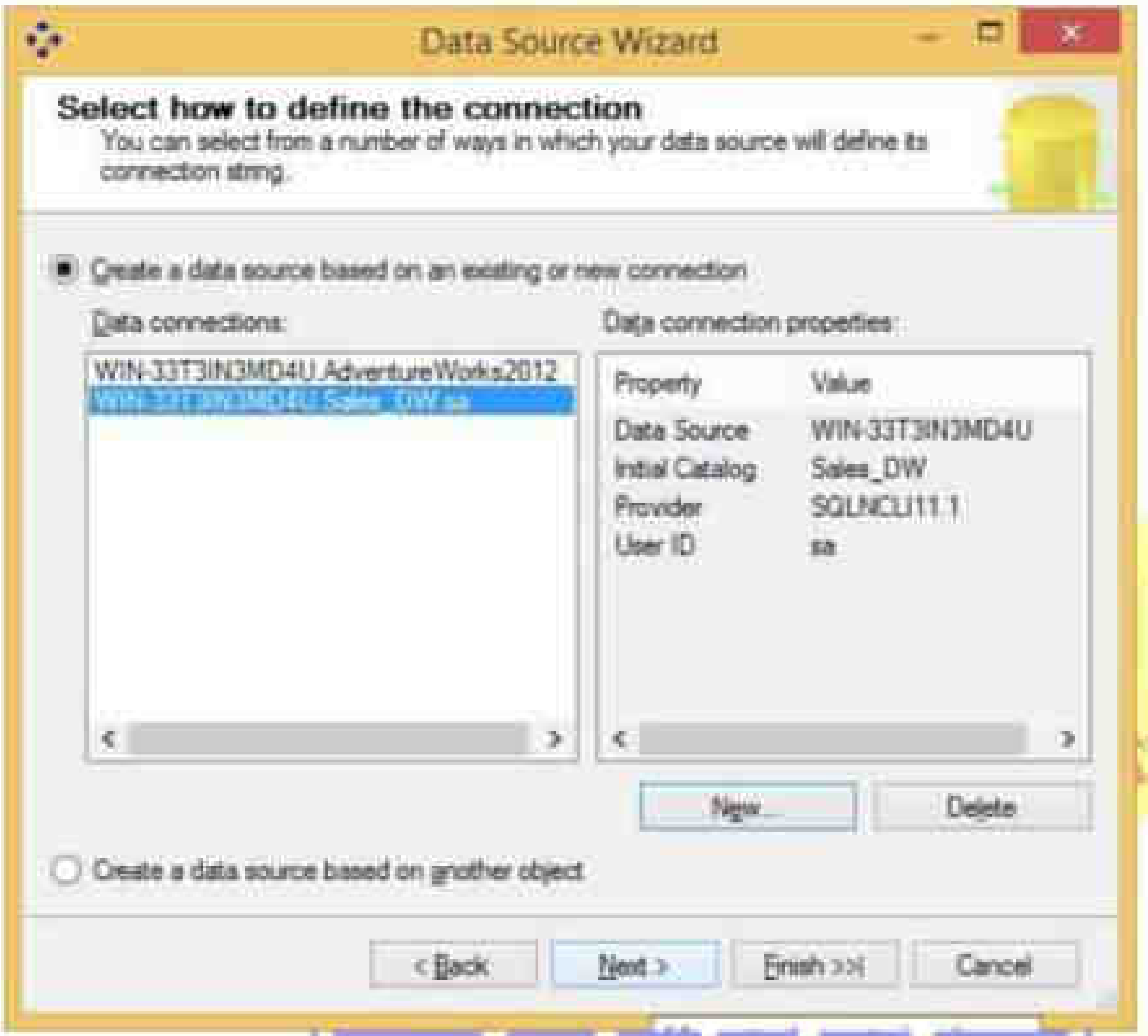
○ Attach a database file:



Connection Manager

Test connection succeeded.

Click Next



Data Source Wizard

Select how to define the connection

You can select from a number of ways in which your data source will define its connection string.

● Create a data source based on an existing or new connection

Data connections:

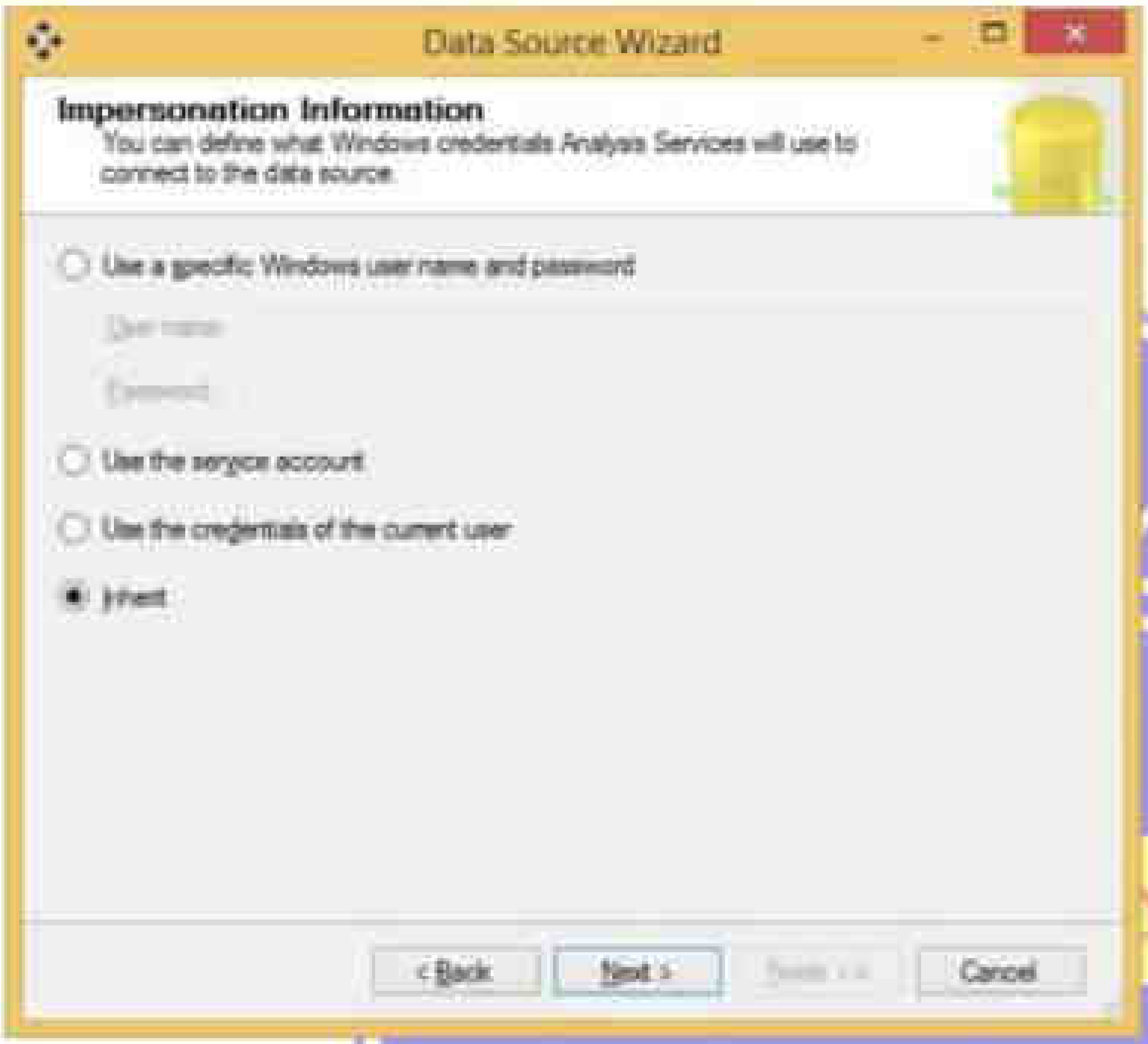
WIN-33T3IN3MD4U.AdventureWorks2012
WIN-33T3IN3MD4U.Sales_DW.sa

Data connection properties:

Property	Value
Data Source	WIN-33T3IN3MD4U
Initial Catalog	Sales_DW
Provider	SQLNCLI11.1
User ID	sa

○ Create a data source based on another object

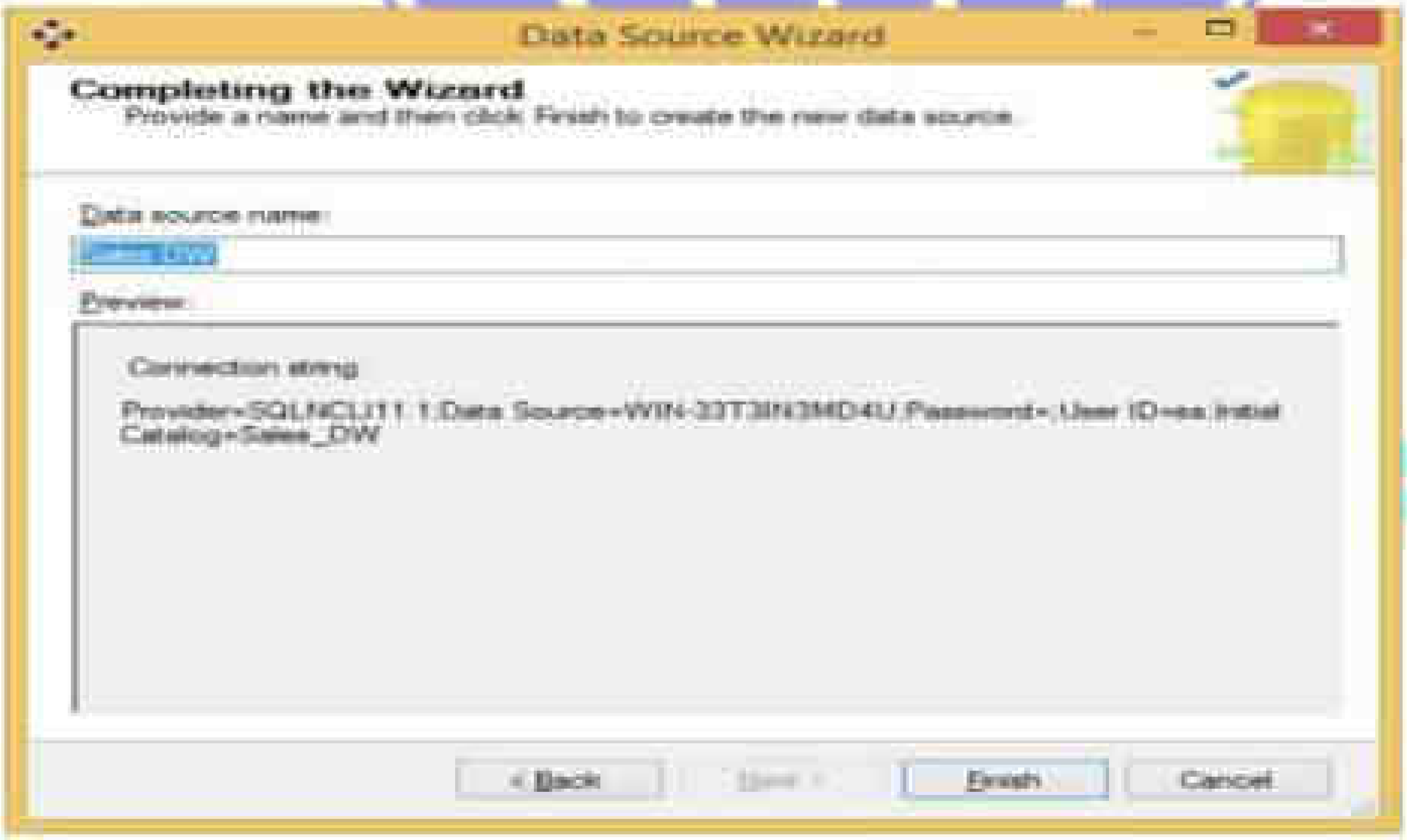
Select Inherit → Next



Impersonation Information
You can define what Windows credentials Analysis Services will use to connect to the data source.

○ Use a specific Windows user name and password
User name:
Password:
○ Use the service account
○ Use the credentials of the current user
● Inherit

Click Finish



Completing the Wizard
Provide a name and then click Finish to create the new data source.

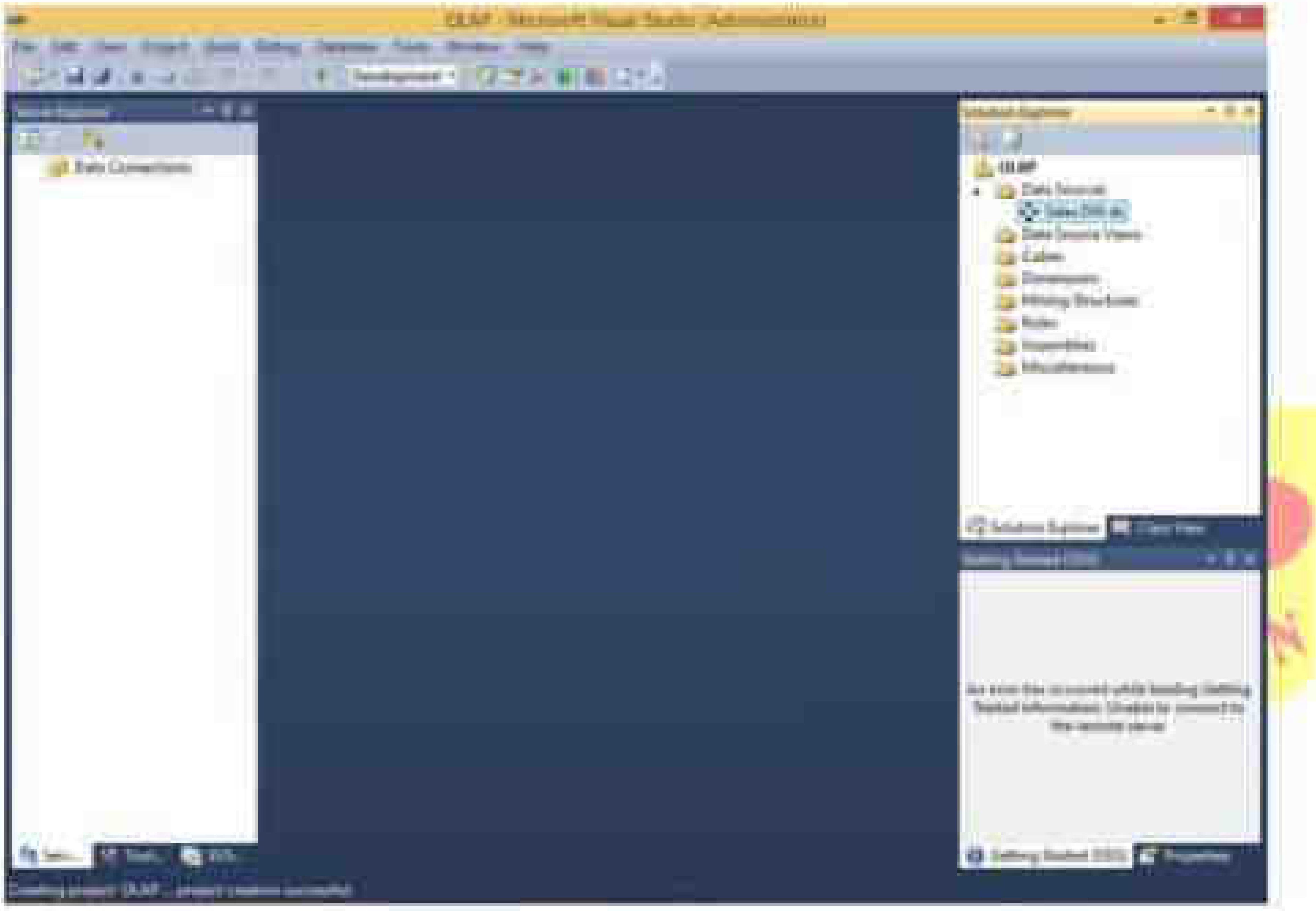
Data source name:
Sales DW

Preview:

Connection string

```
Provider=SQLNCLI11.1;Data Source=WIN-33T3N3MD4U;Password=;User ID=sa;Initial Catalog=Sales_DW
```

Sales_DW.ds gets created under Data Sources in Solution Explorer

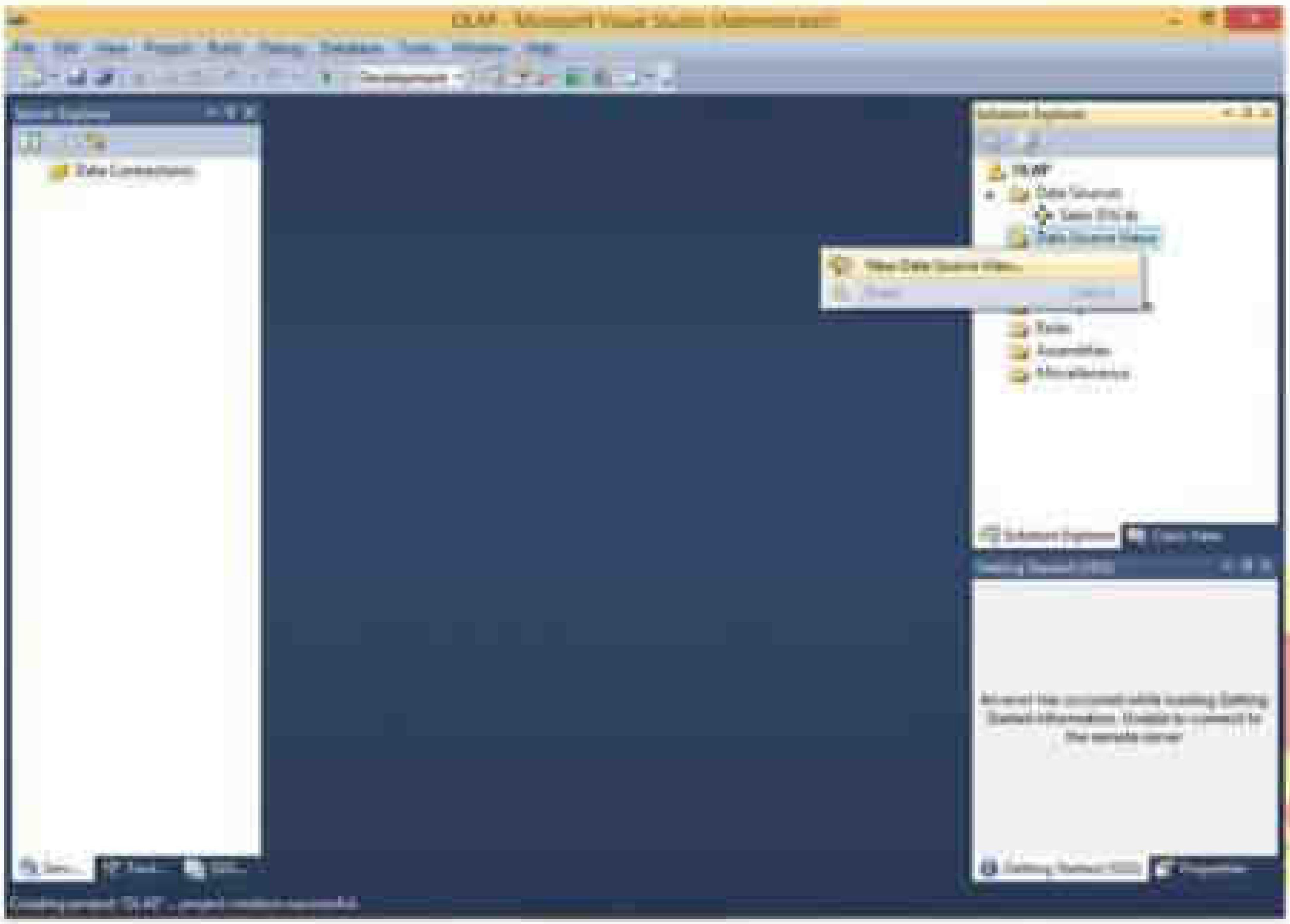


An error has occurred while loading Getting Started information. Unable to connect to the remote server.

Creating project 'OLAP'... project creation successful

Step 3: Creating New Data Source View

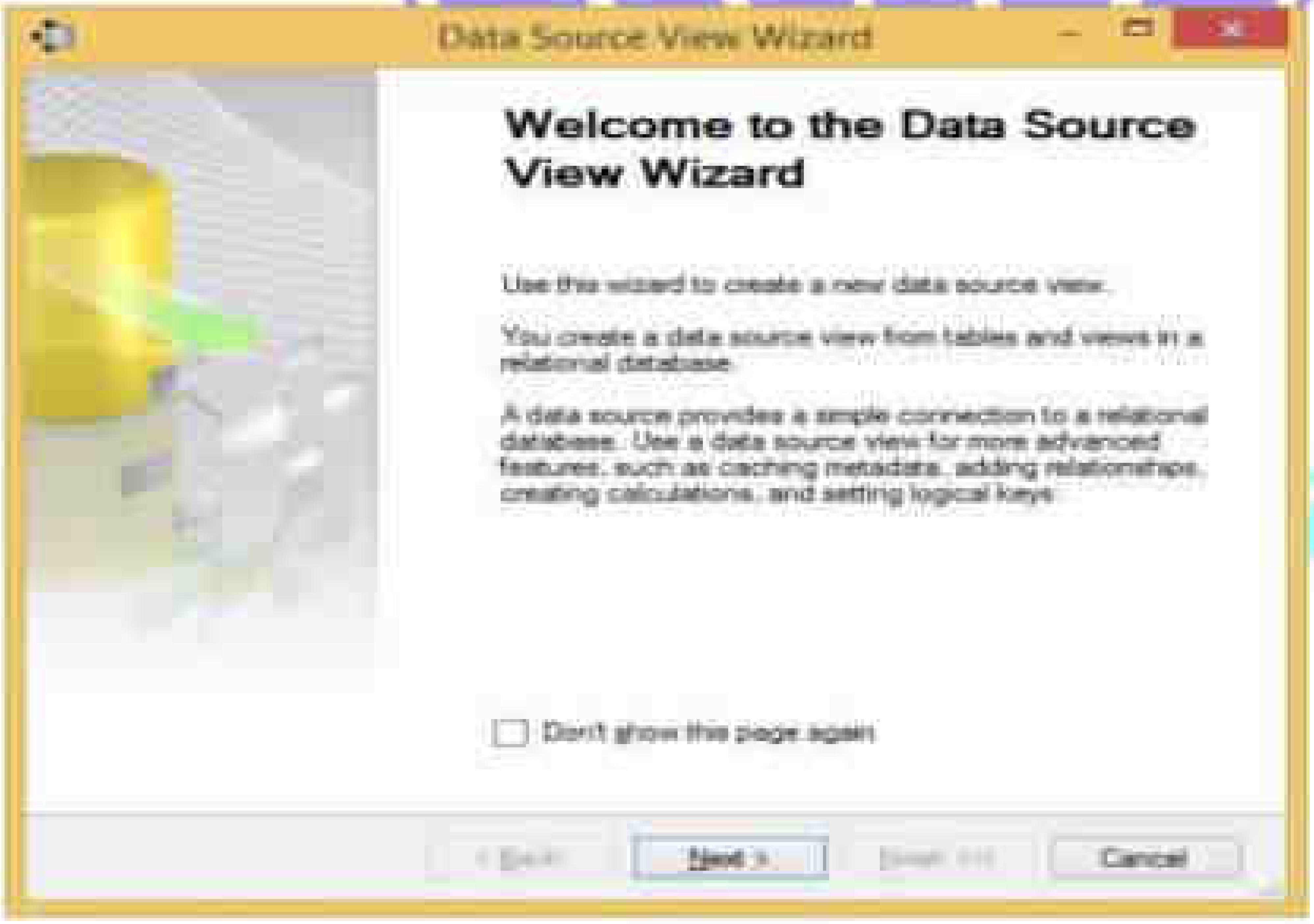
In Solution explorer right click on Data Source View → Select New Data Source View



An error has occurred while loading Getting Started information. Unable to connect to the remote server

Creating project 'OLAP'... project creation successful.

Click Next



Welcome to the Data Source View Wizard

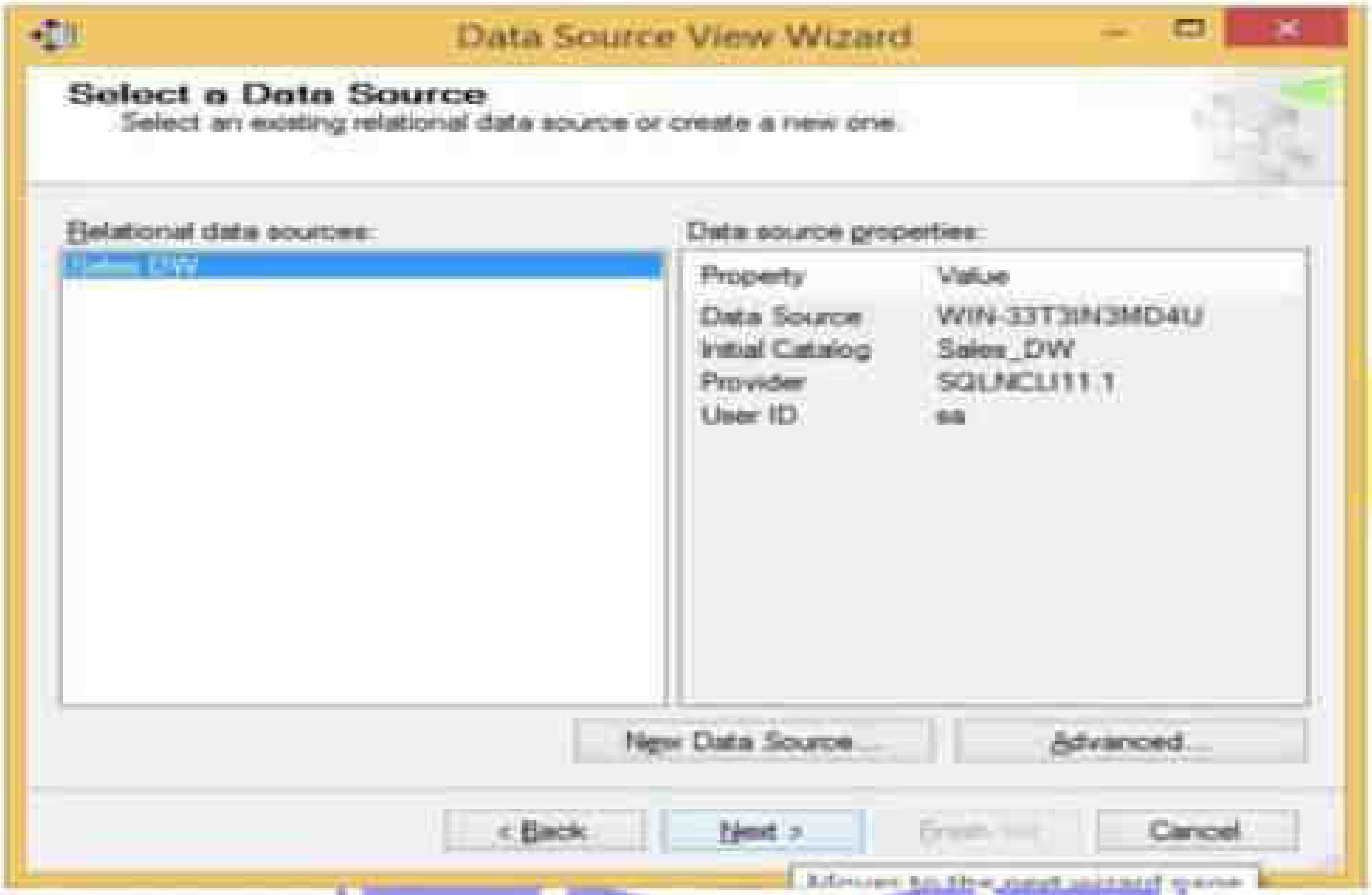
Use this wizard to create a new data source view.

You create a data source view from tables and views in a relational database.

A data source provides a simple connection to a relational database. Use a data source view for more advanced features, such as caching metadata, adding relationships, creating calculations, and setting logical keys.

☐ Don't show this page again

Click Next



Data Source View Wizard

Select a Data Source

Select an existing relational data source or create a new one.

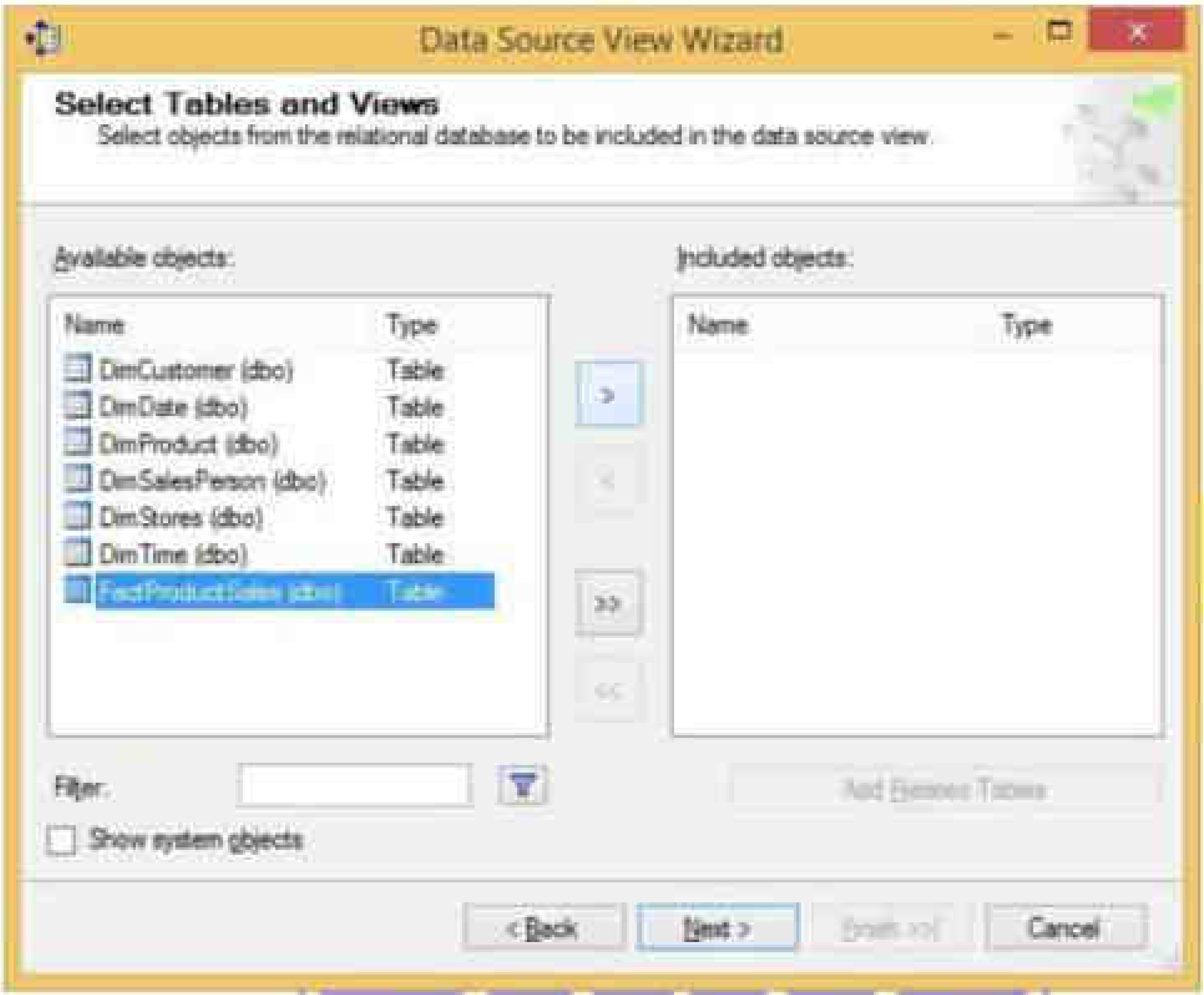
Relational data sources:

Sales DW

Data source properties:

Property	Value
Data Source	WIN-33T3IN3MD4U
Initial Catalog	Sales_DW
Provider	SQLNCLI11.1
User ID	sa

Select FactProductSales(dbo) from Available objects and put in Includes Objects by clicking on >



Data Source View Wizard

Select Tables and Views

Select objects from the relational database to be included in the data source view.

Available objects:

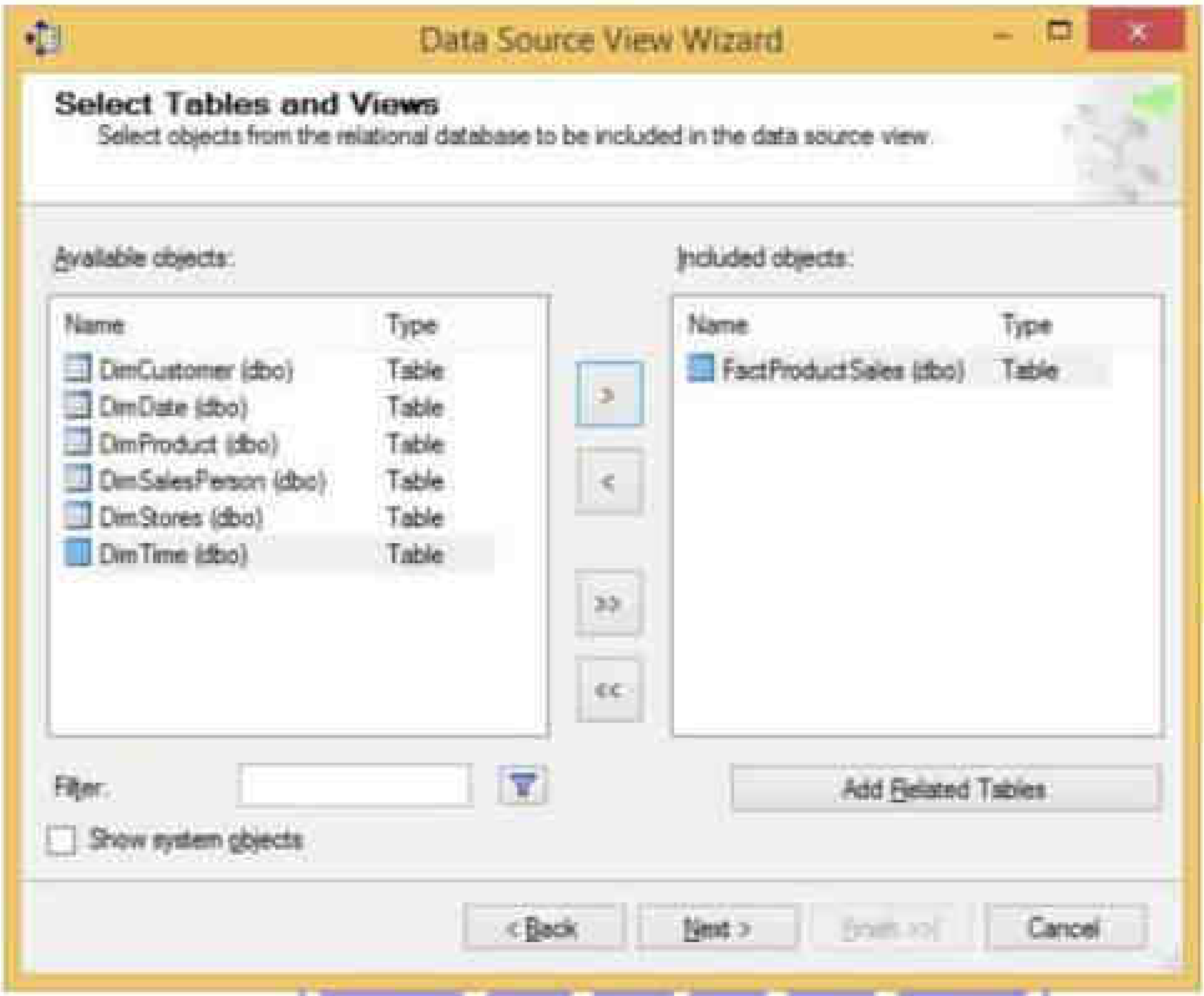
Name	Type
DimCustomer (dbo)	Table
DimDate (dbo)	Table
DimProduct (dbo)	Table
DimSalesPerson (dbo)	Table
DimStores (dbo)	Table
DimTime (dbo)	Table
FactProductSales (dbo)	Table

Included objects:

Name	Type

Filter:

□ Show system objects



Select Tables and Views

Select objects from the relational database to be included in the data source view.

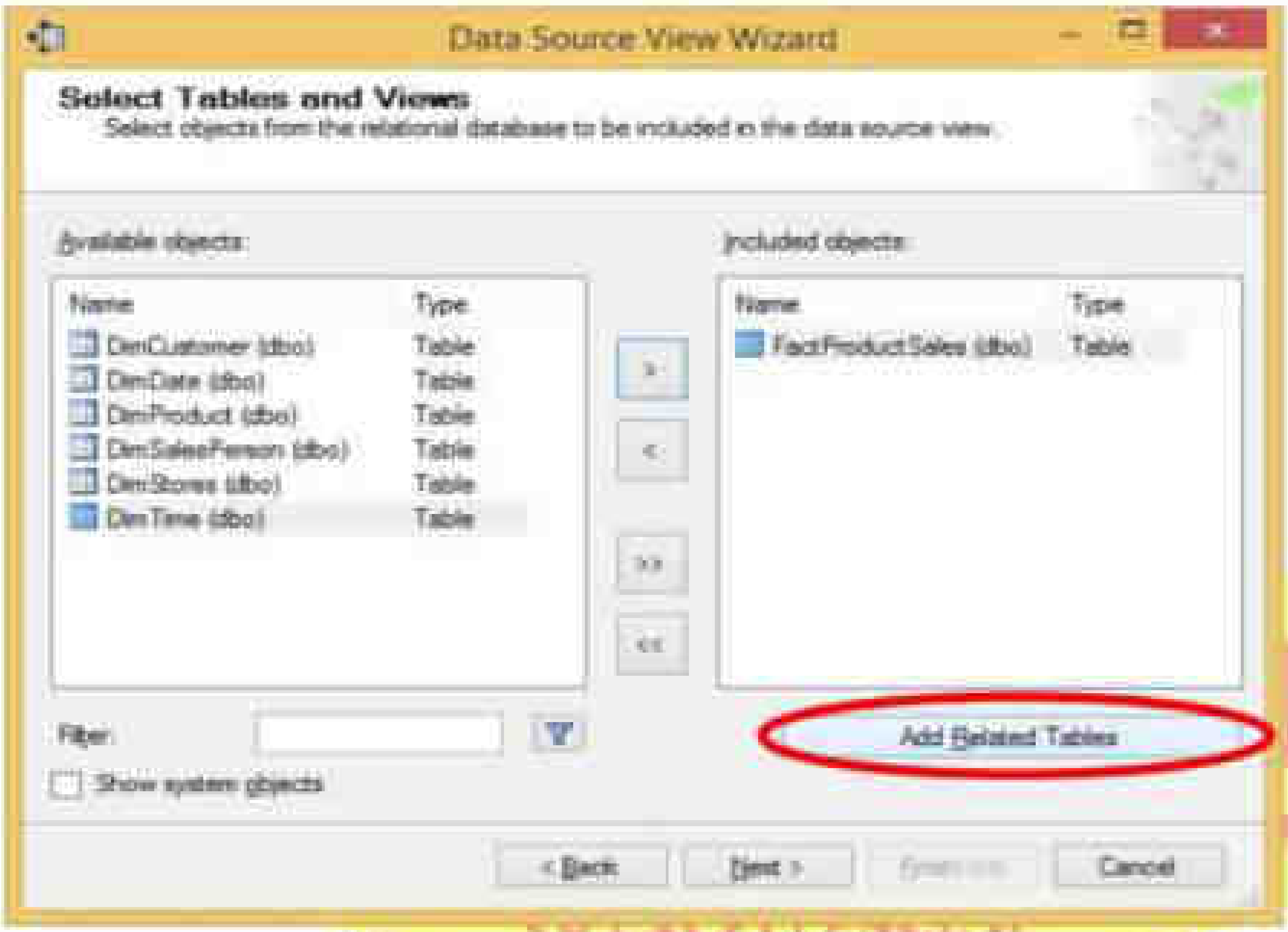
Available objects:

Name	Type
DimCustomer (dbo)	Table
DimDate (dbo)	Table
DimProduct (dbo)	Table
DimSalesPerson (dbo)	Table
DimStores (dbo)	Table
DimTime (dbo)	Table

Included objects:

Name	Type
FactProductSales (dbo)	Table

Click on Add Related Tables



Data Source View Wizard

Select Tables and Views

Select objects from the relational database to be included in the data source view.

Available objects:

Name	Type
DimCustomer (dbo)	Table
DimDate (dbo)	Table
DimProduct (dbo)	Table
DimSalesPerson (dbo)	Table
DimStores (dbo)	Table
DimTime (dbo)	Table

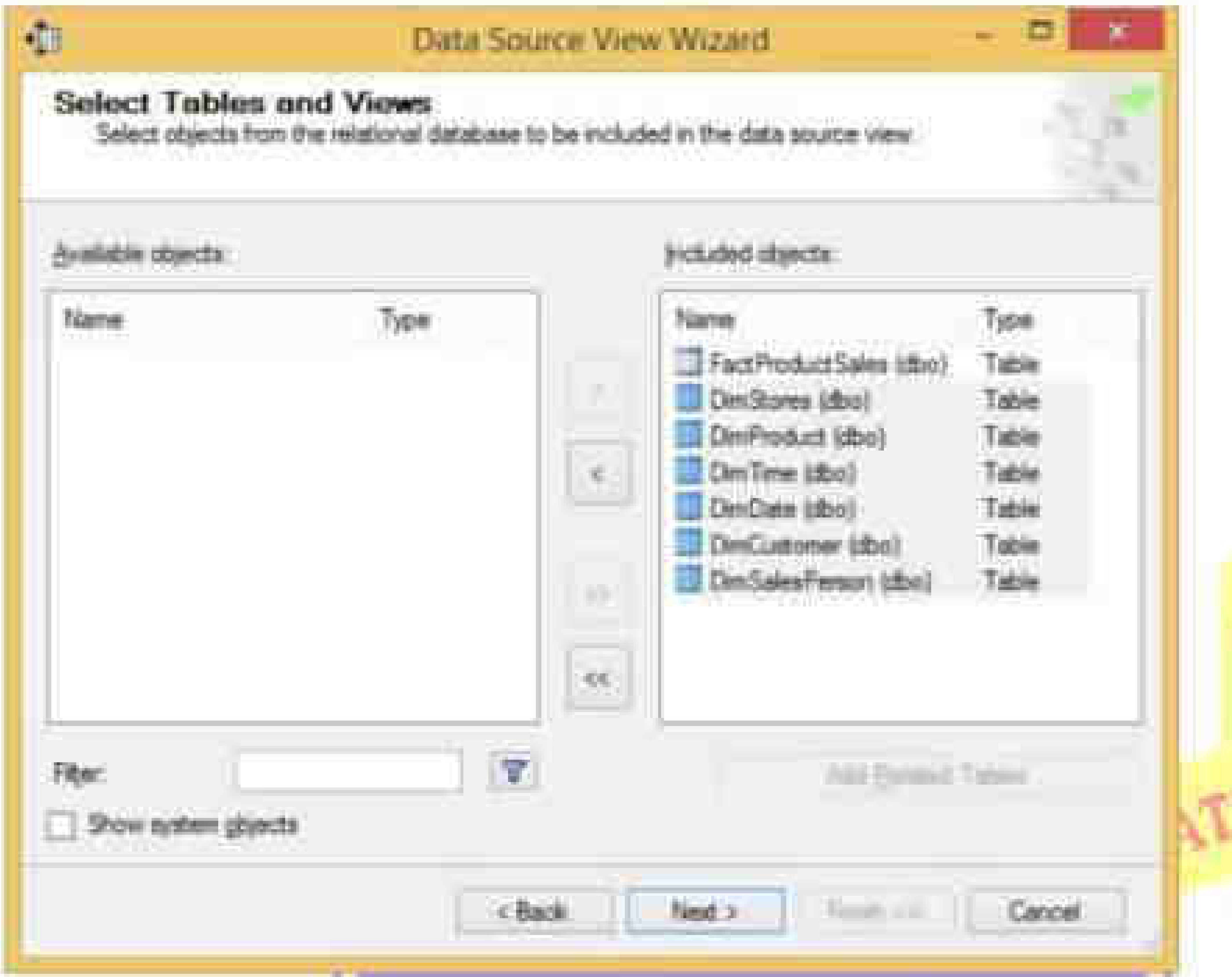
Included objects:

Name	Type
FactProductSales (dbo)	Table

Filter:

□ Show system objects

Click Next



Data Source View Wizard

Select Tables and Views

Select objects from the relational database to be included in the data source view.

Available objects:

Name	Type

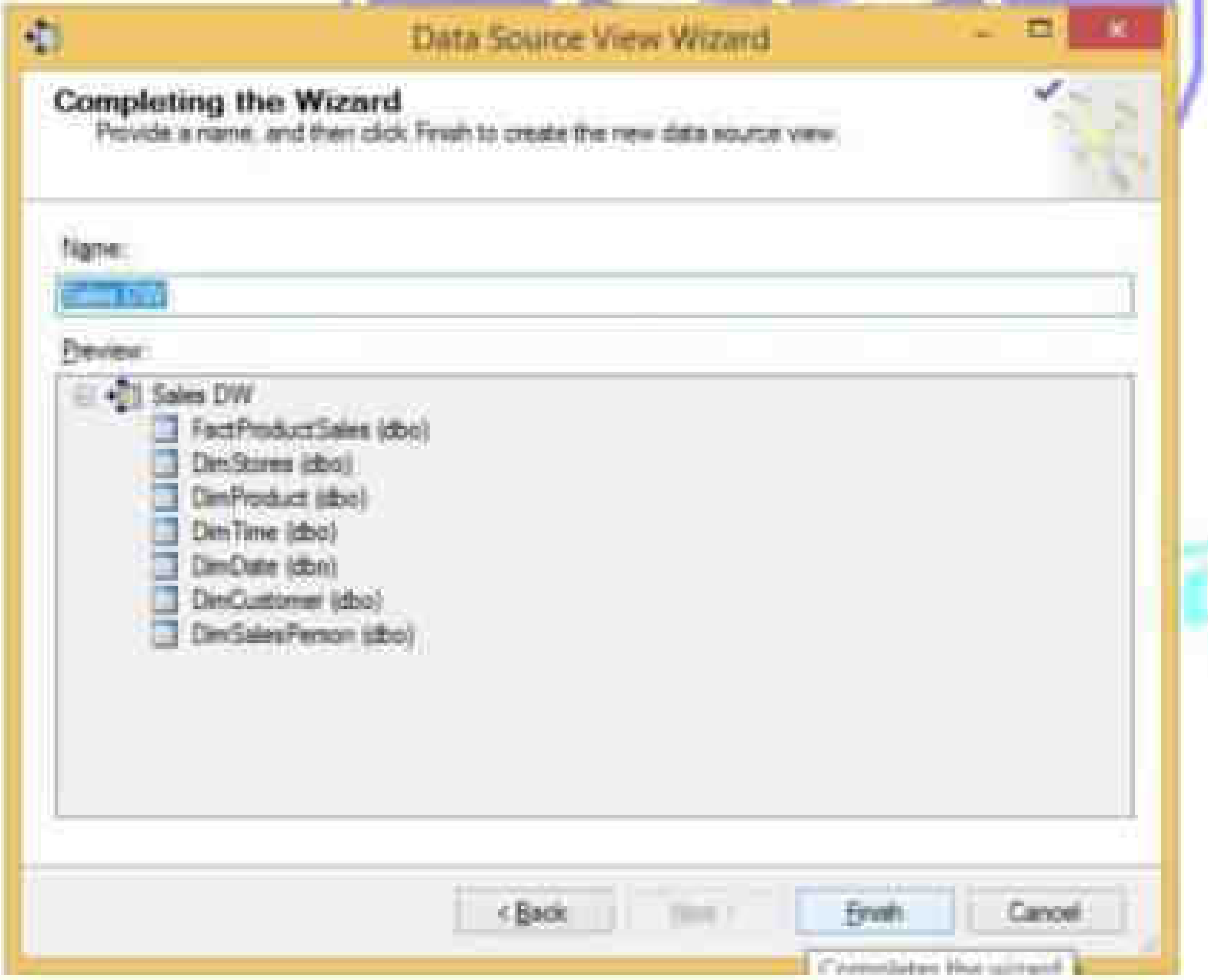
Included objects:

Name	Type
FactProductSales (dbo)	Table
DimStores (dbo)	Table
DimProduct (dbo)	Table
DimTime (dbo)	Table
DimDate (dbo)	Table
DimCustomer (dbo)	Table
DimSalesPerson (dbo)	Table

Filter:

□ Show system objects

Click Finish



Data Source View Wizard

Completing the Wizard

Provide a name, and then click Finish to create the new data source view.

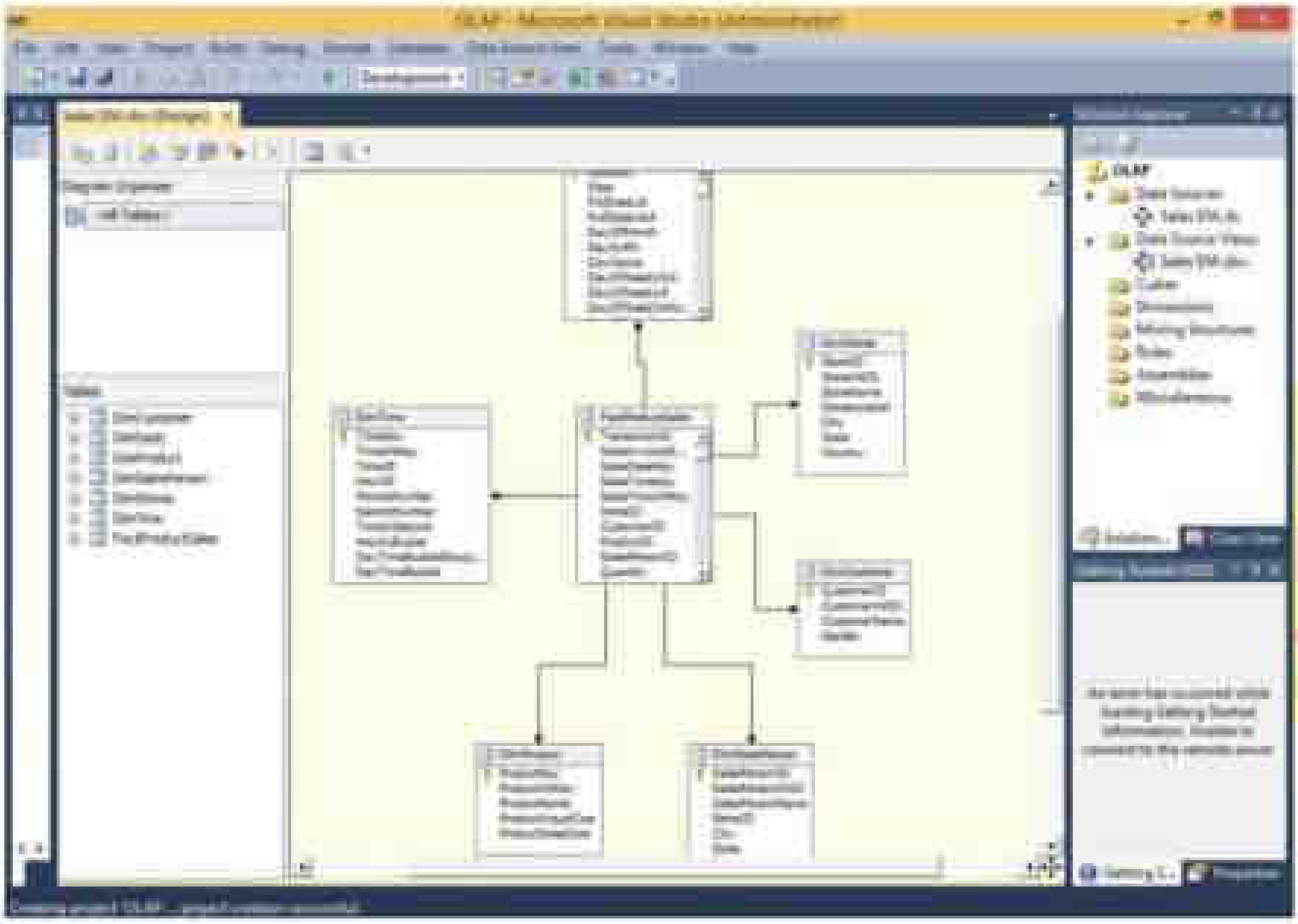
Name:

Sales DW

Preview:

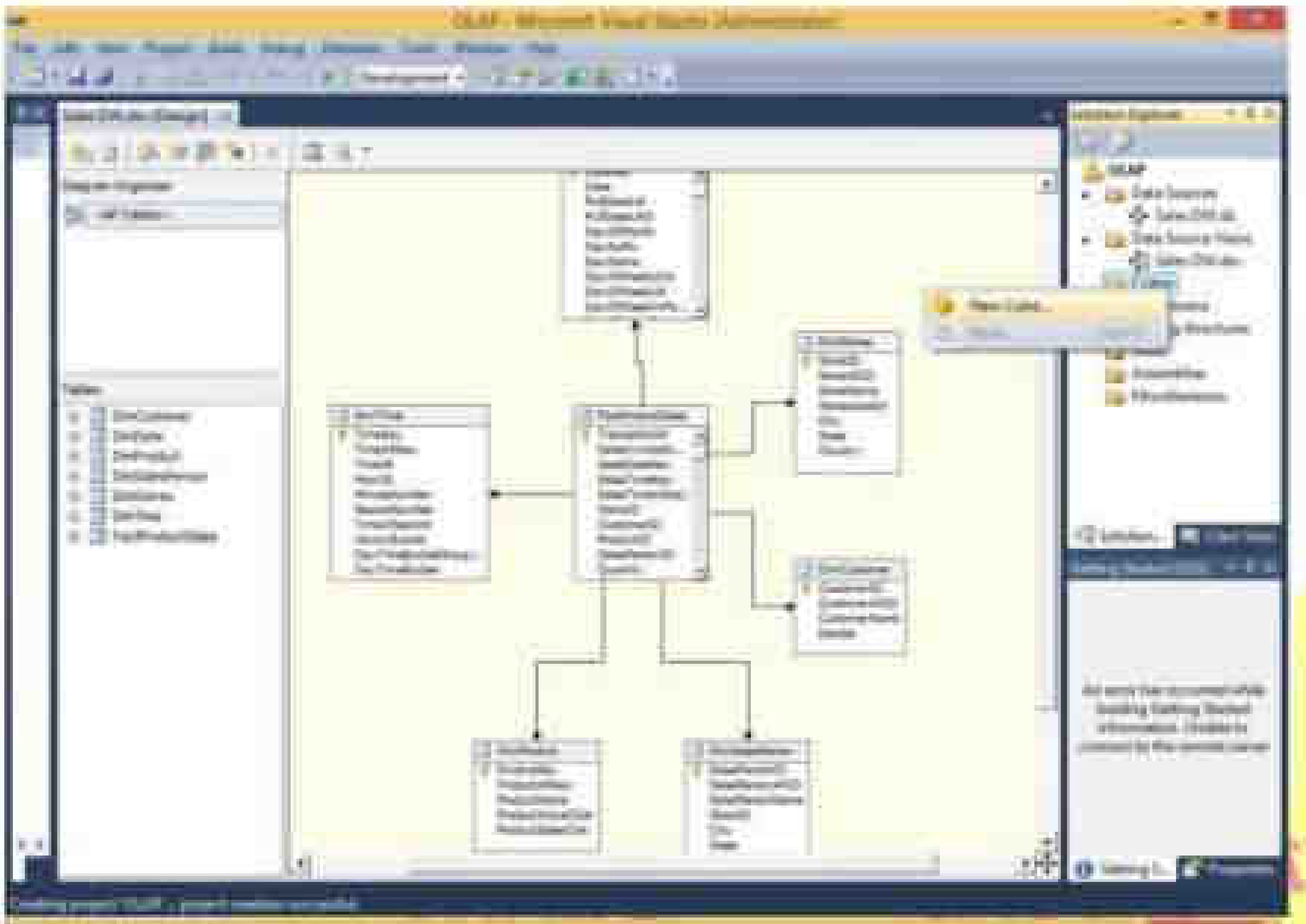
```
Sales DW
    FactProductSales (dbo)
    DimStores (dbo)
    DimProduct (dbo)
    DimTime (dbo)
    DimDate (dbo)
    DimCustomer (dbo)
    DimSalesPerson (dbo)
```

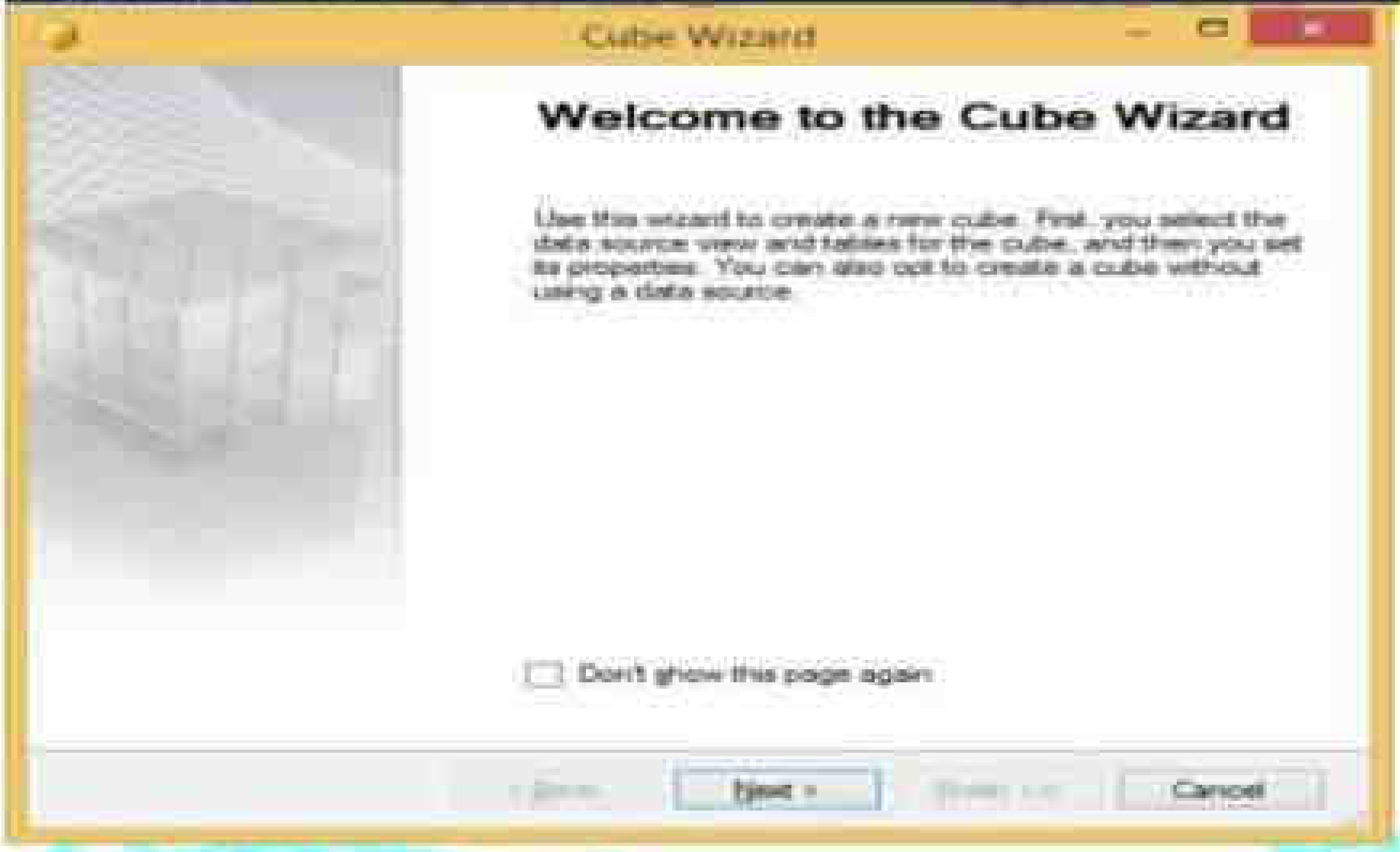
Sales DW.dsv appears in Data Source Views in Solution Explorer.



Step 4: Creating new cube

Right click on Cubes → New Cube





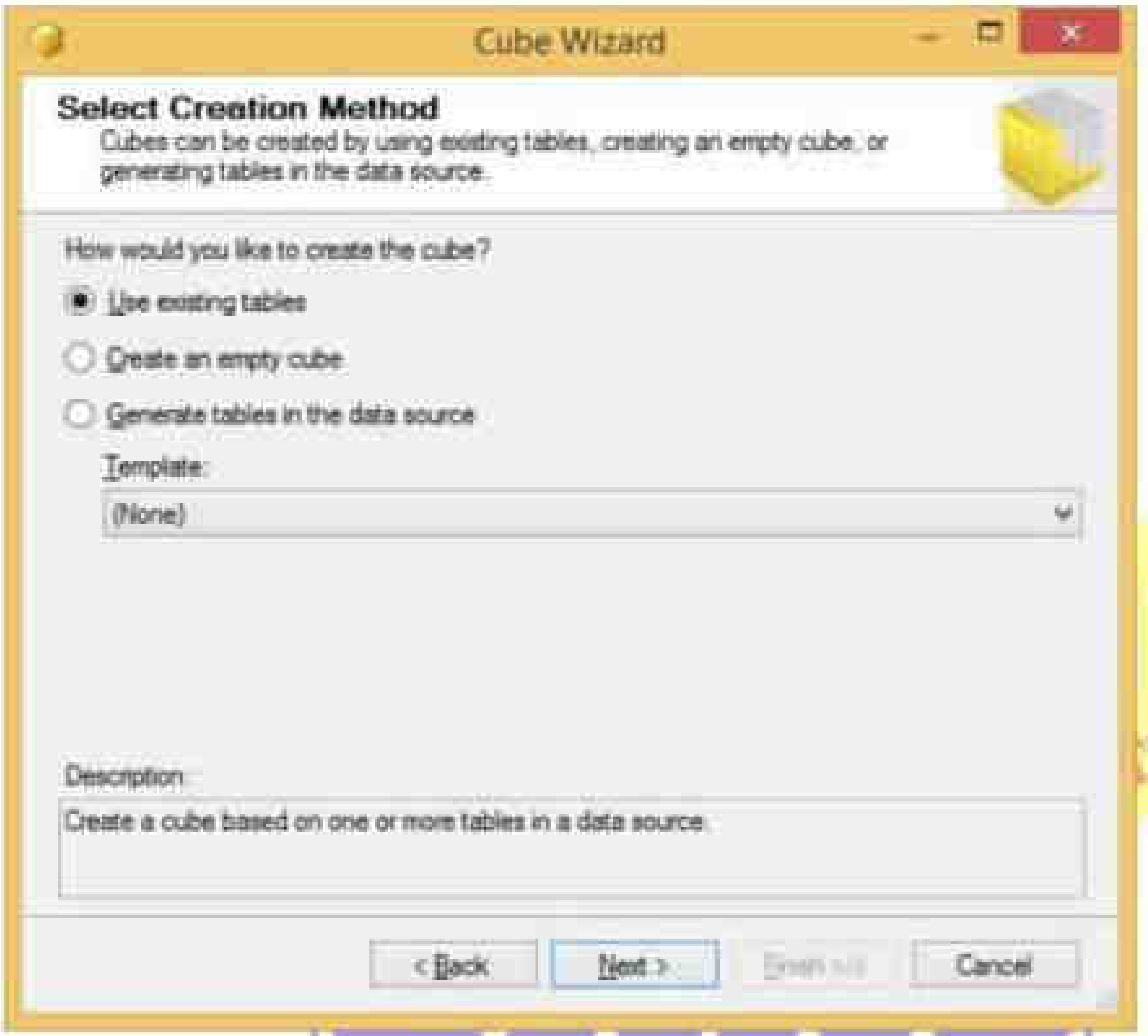
Welcome to the Cube Wizard

Use this wizard to create a new cube. First, you select the data source view and tables for the cube, and then you set its properties. You can also opt to create a cube without using a data source.

□ Don't show this page again

< Back | Next > | Finish >>| | Cancel

Select Use existing tables in Select Creation Method → Next



Cube Wizard

Select Creation Method

Cubes can be created by using existing tables, creating an empty cube, or generating tables in the data source.

How would you like to create the cube?

● Use existing tables

○ Create an empty cube

○ Generate tables in the data source

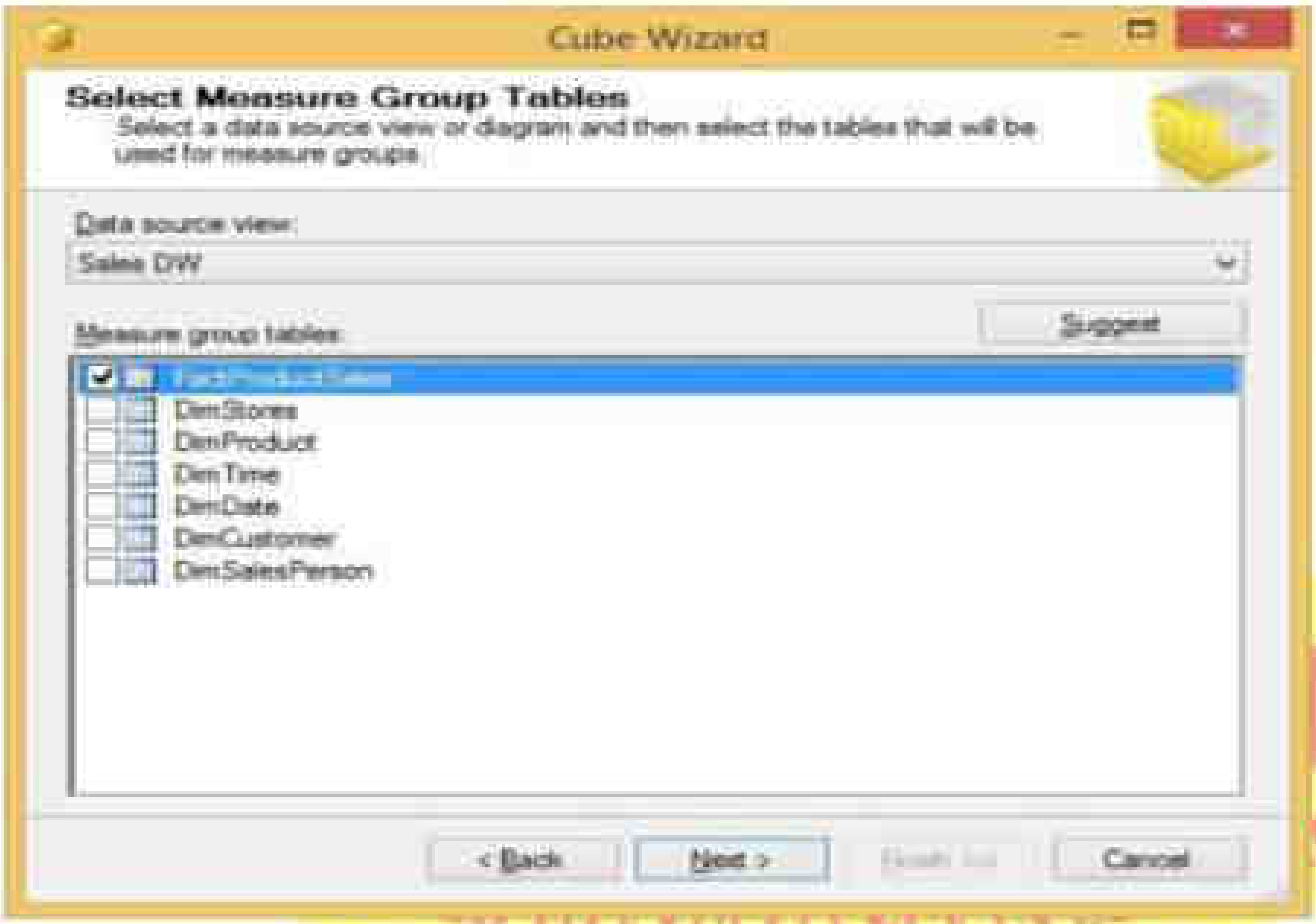
Template:

(None)

Description:

Create a cube based on one or more tables in a data source.

In Select Measure Group Tables → Select FactProductSales → Click Next



Select Measure Group Tables

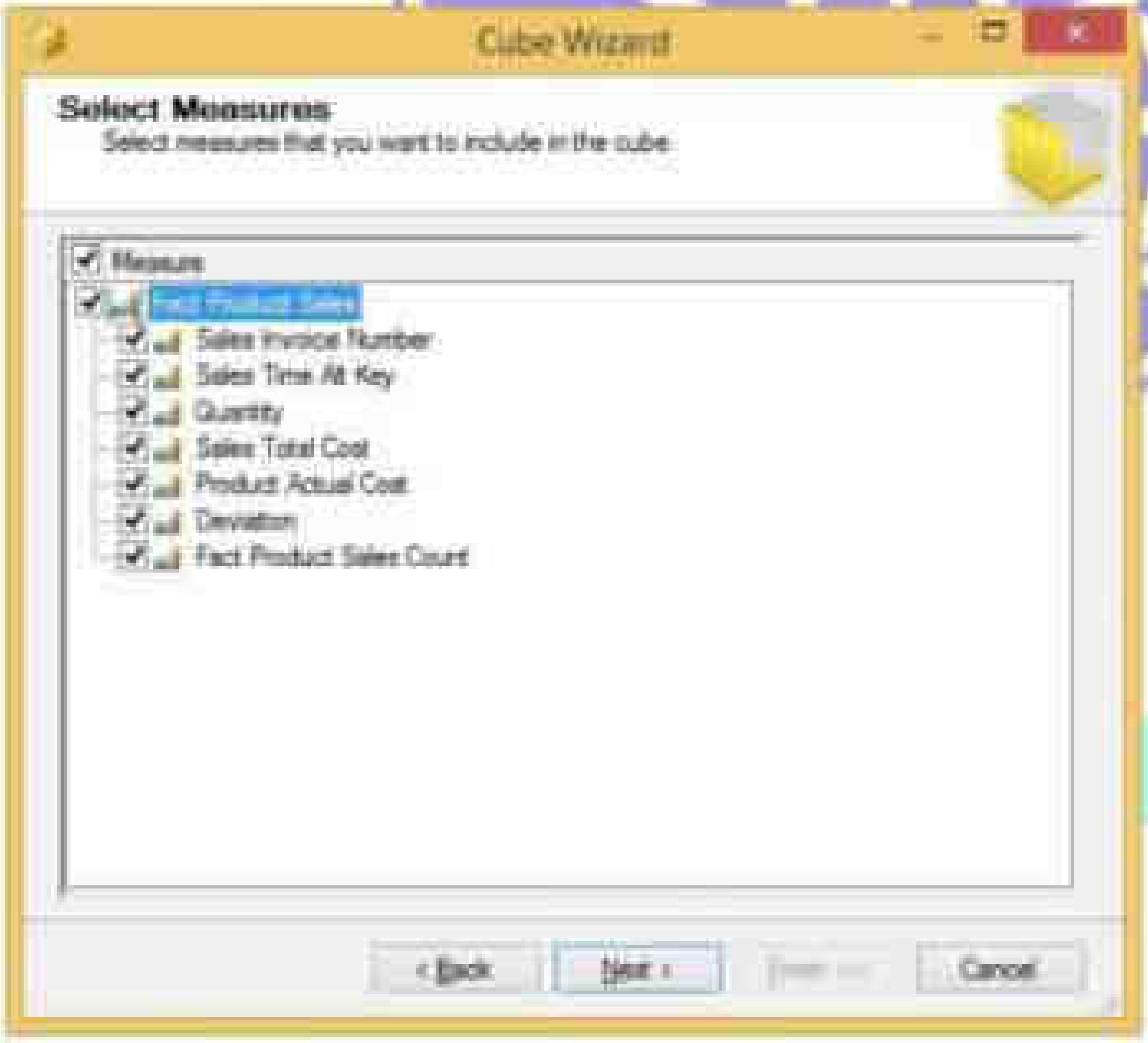
Select a data source view or diagram and then select the tables that will be used for measure groups.

Data source view:
Sales DW

Measure group tables:

- ☑ FactProductSales
- □ DimStores
- □ DimProduct
- □ DimTime
- □ DimDate
- □ DimCustomer
- □ DimSalesPerson

In Select Measures → check all measures → Next

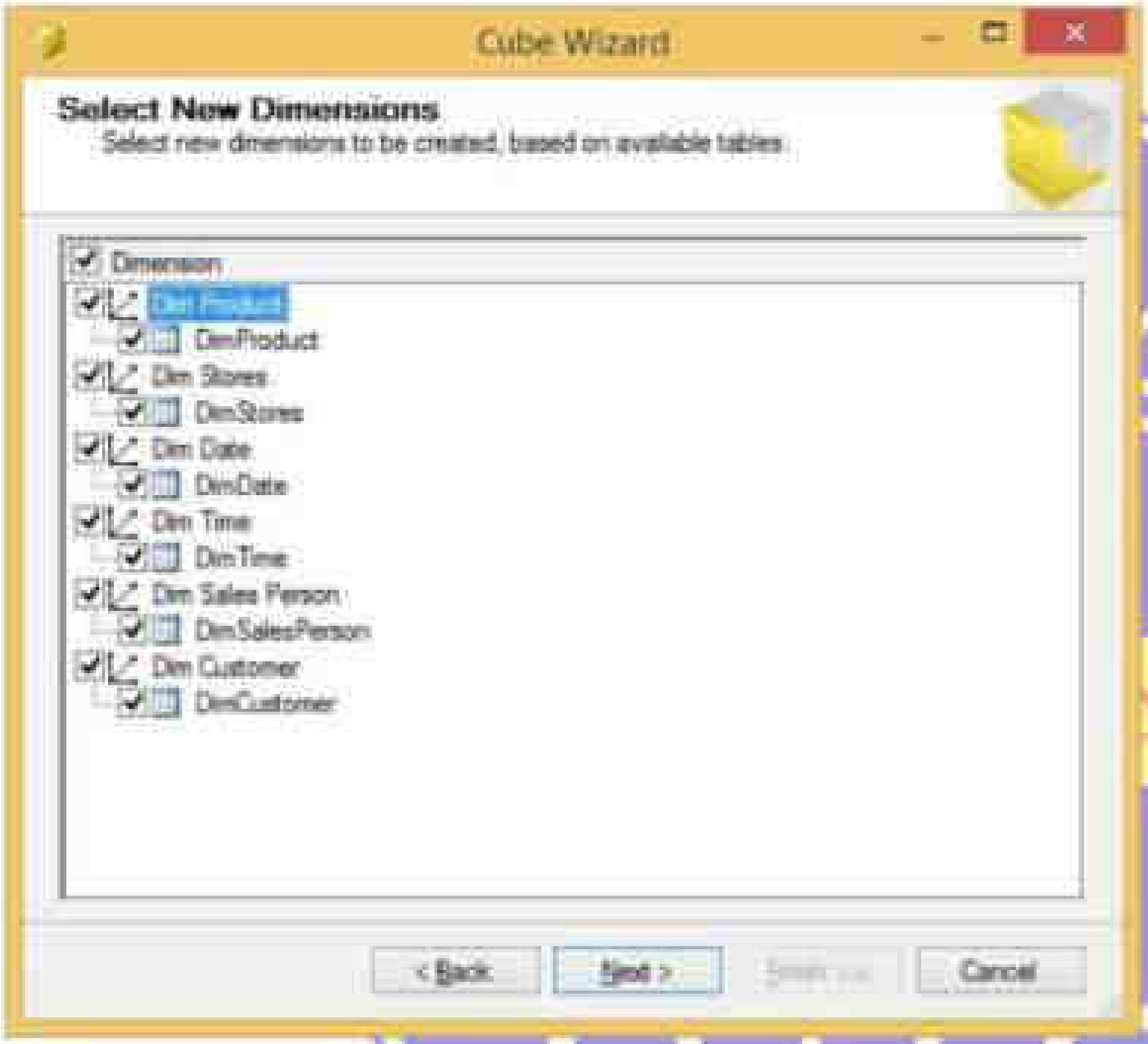


Select Measures

Select measures that you want to include in the cube.

- ☑ Measure
  - ☑ Fact Product Sales
    - ☑ Sales Invoice Number
    - ☑ Sales Time Alt Key
    - ☑ Quantity
    - ☑ Sales Total Cost
    - ☑ Product Actual Cost
    - ☑ Deviation
    - ☑ Fact Product Sales Count

In Select New Dimensions → Check all Dimensions → Next

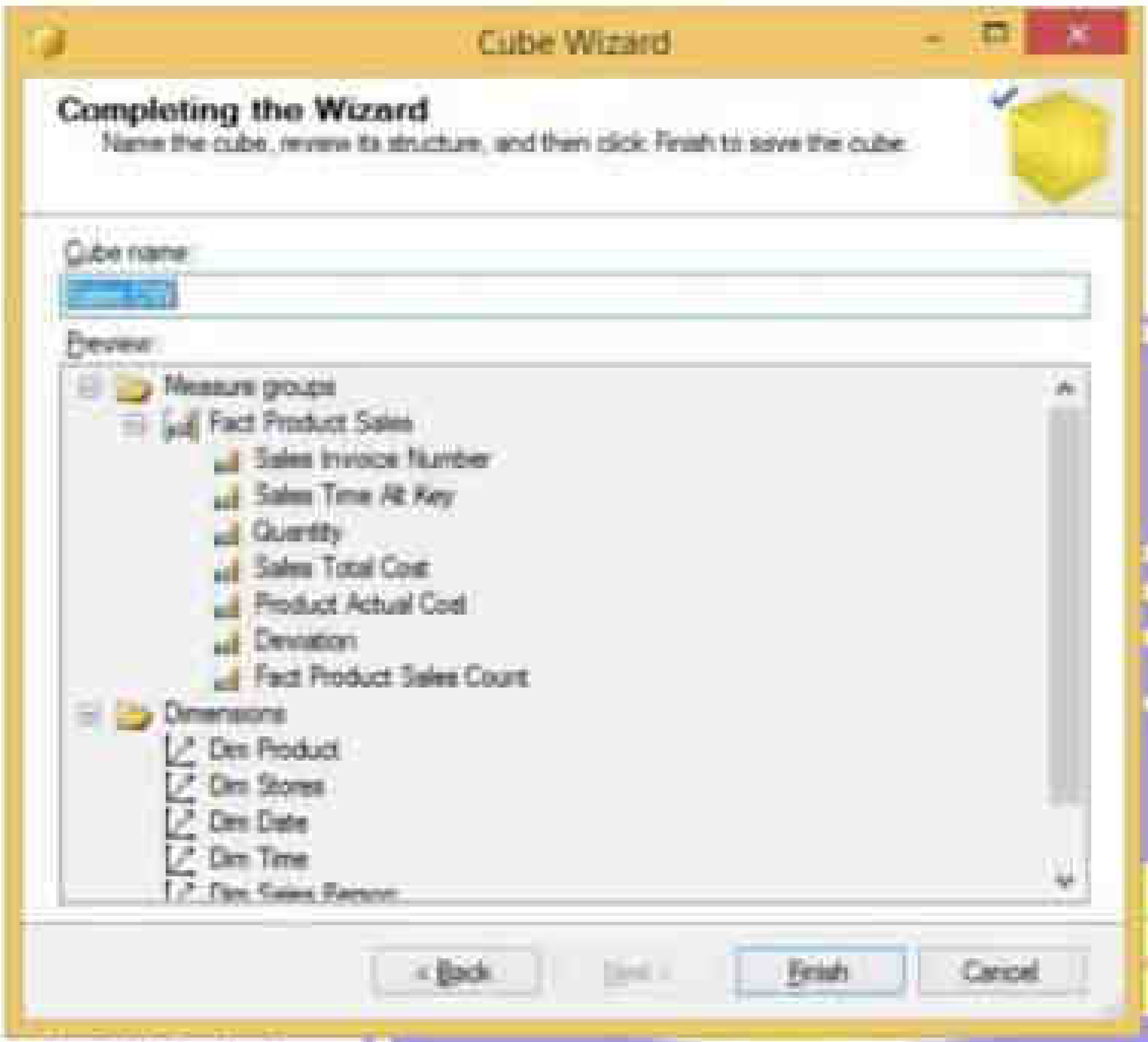


Select New Dimensions

Select new dimensions to be created, based on available tables

- ☑ Dimension
  - ☑ Dim Product
    - ☑ DimProduct
  - ☑ Dim Stores
    - ☑ DimStores
  - ☑ Dim Date
    - ☑ DimDate
  - ☑ Dim Time
    - ☑ DimTime
  - ☑ Dim Sales Person
    - ☑ DimSalesPerson
  - ☑ Dim Customer
    - ☑ DimCustomer

Click on Finish



Cube Wizard

Completing the Wizard

Name the cube, review its structure, and then click Finish to save the cube.

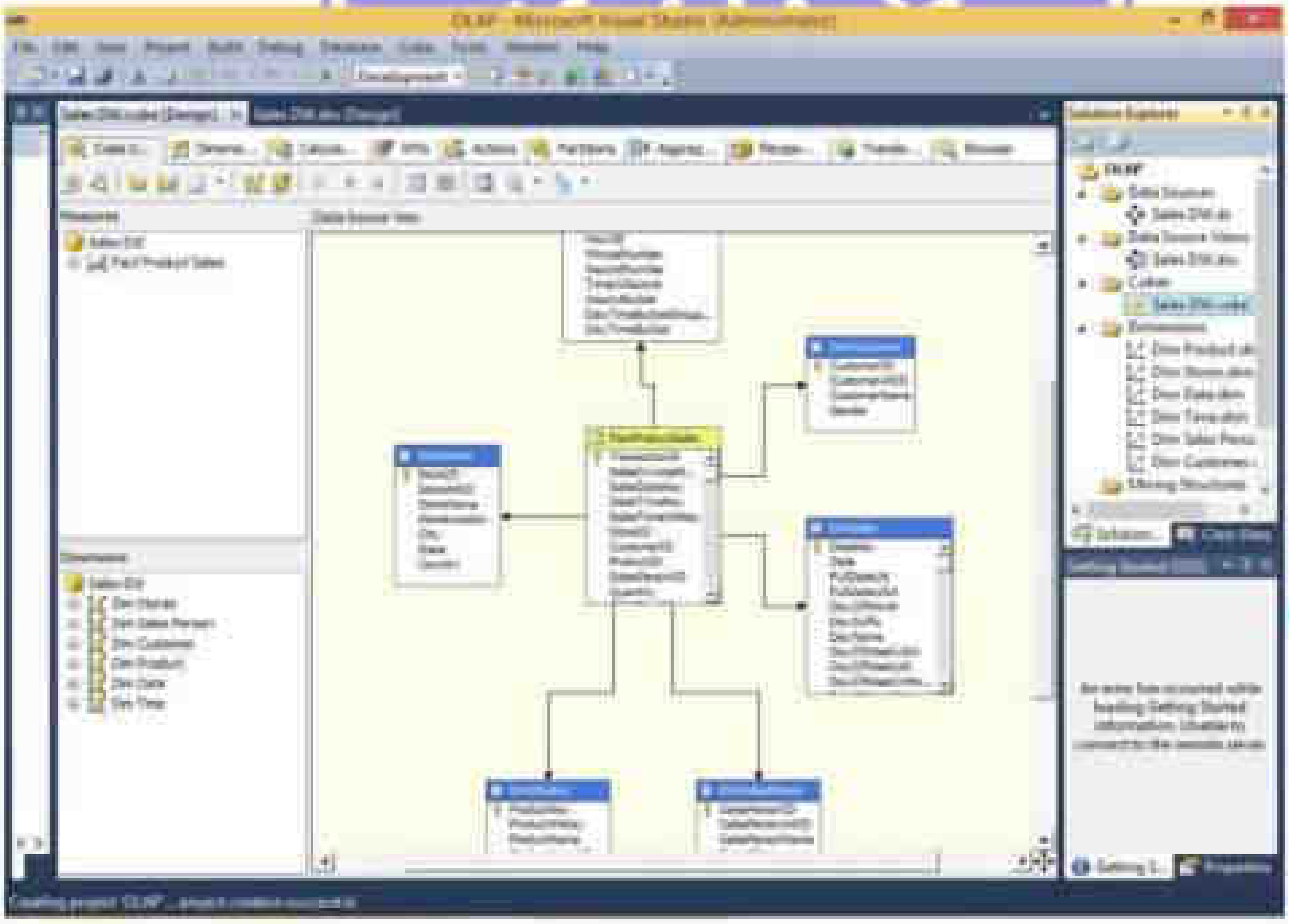
Cube name:

Sales DW

Preview:

- Measure groups
  - Fact Product Sales
    - Sales Invoice Number
    - Sales Time Alt Key
    - Quantity
    - Sales Total Cost
    - Product Actual Cost
    - Deviation
    - Fact Product Sales Count
- Dimensions
  - Dim Product
  - Dim Stores
  - Dim Date
  - Dim Time
  - Dim Sales Person

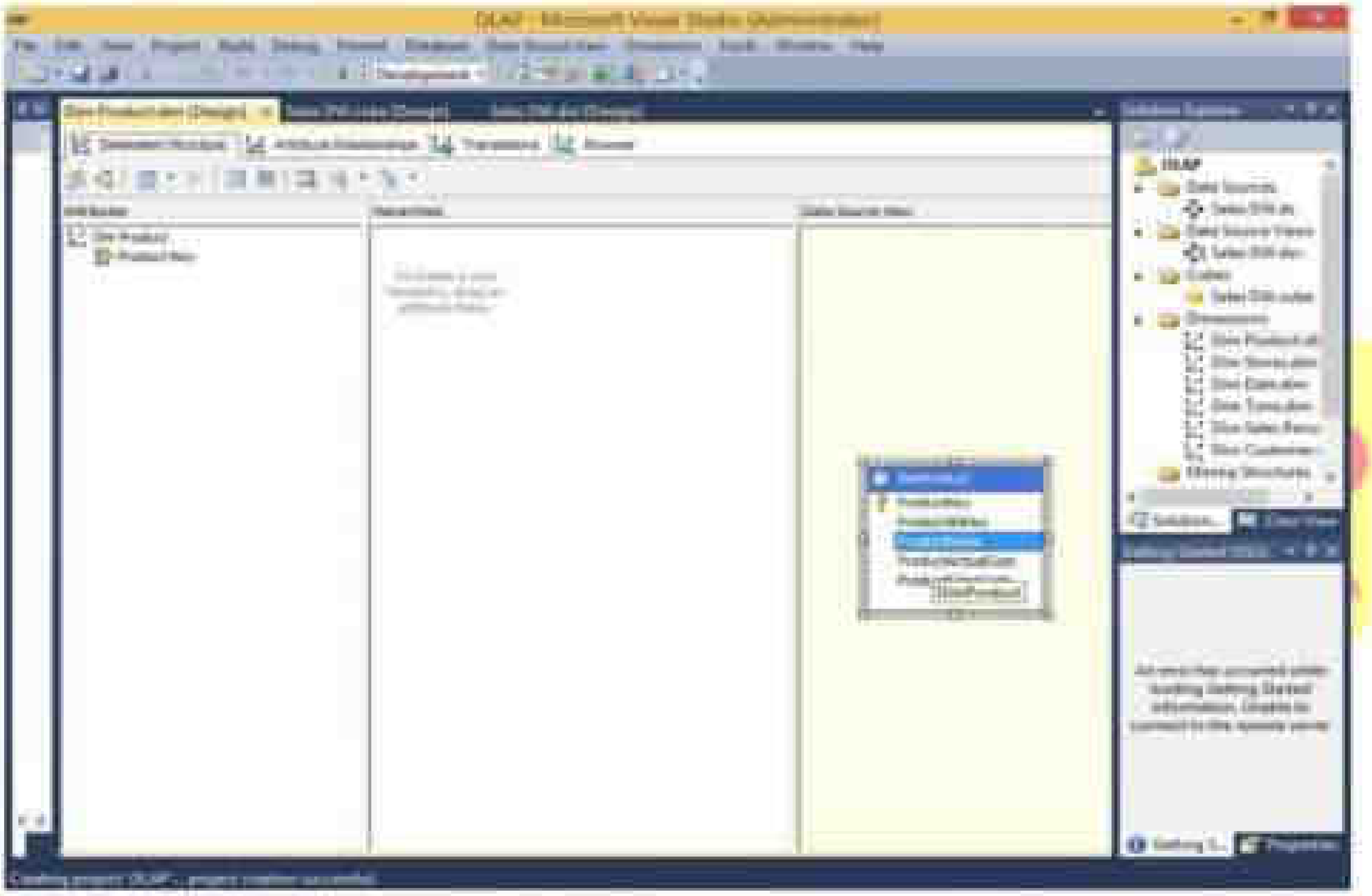
Sales_DW.cube is created



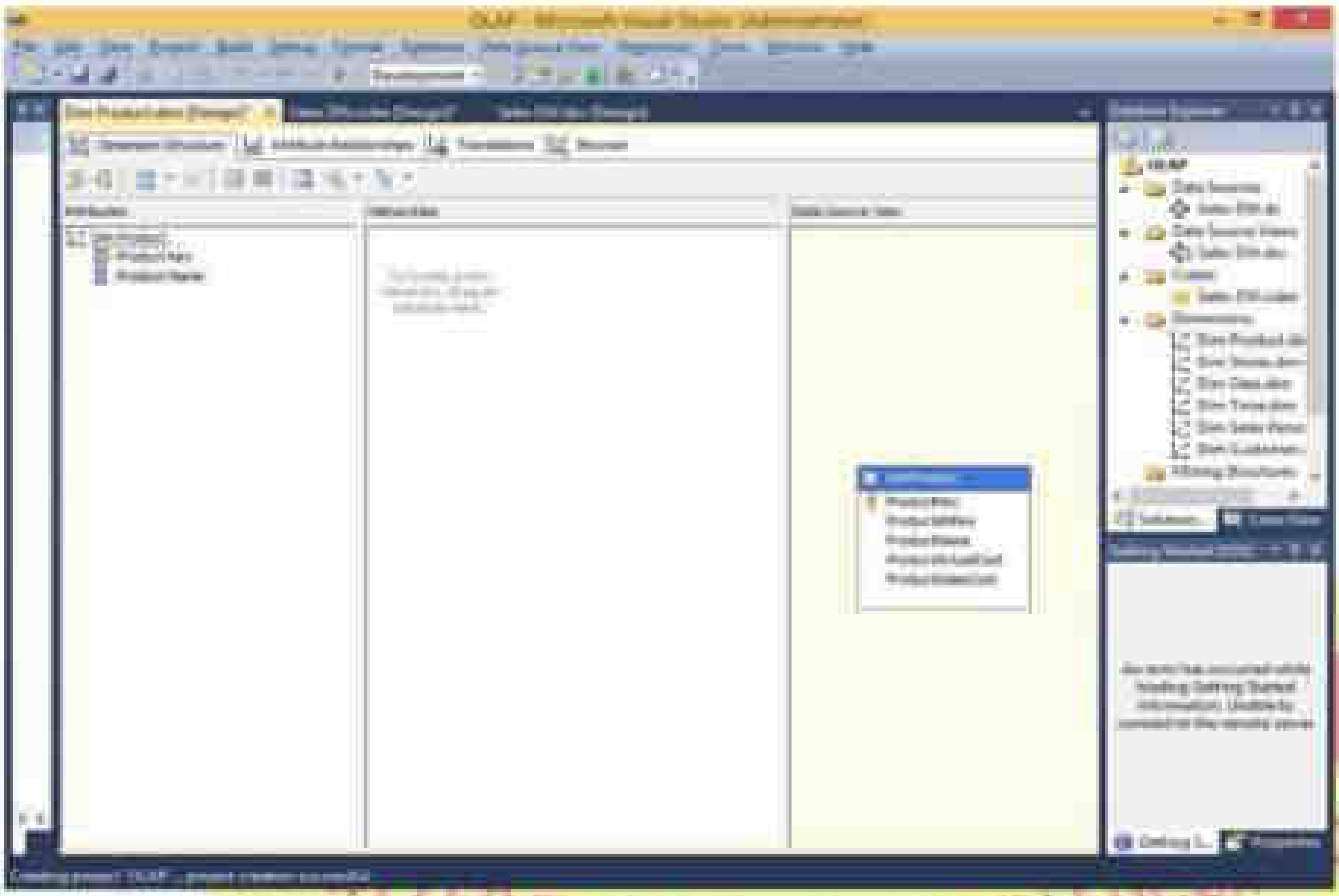
OLAP - Microsoft Visual Studio (Administrator)

Step 5: Dimension Modification

In dimension tab → Double Click Dim Product.dim



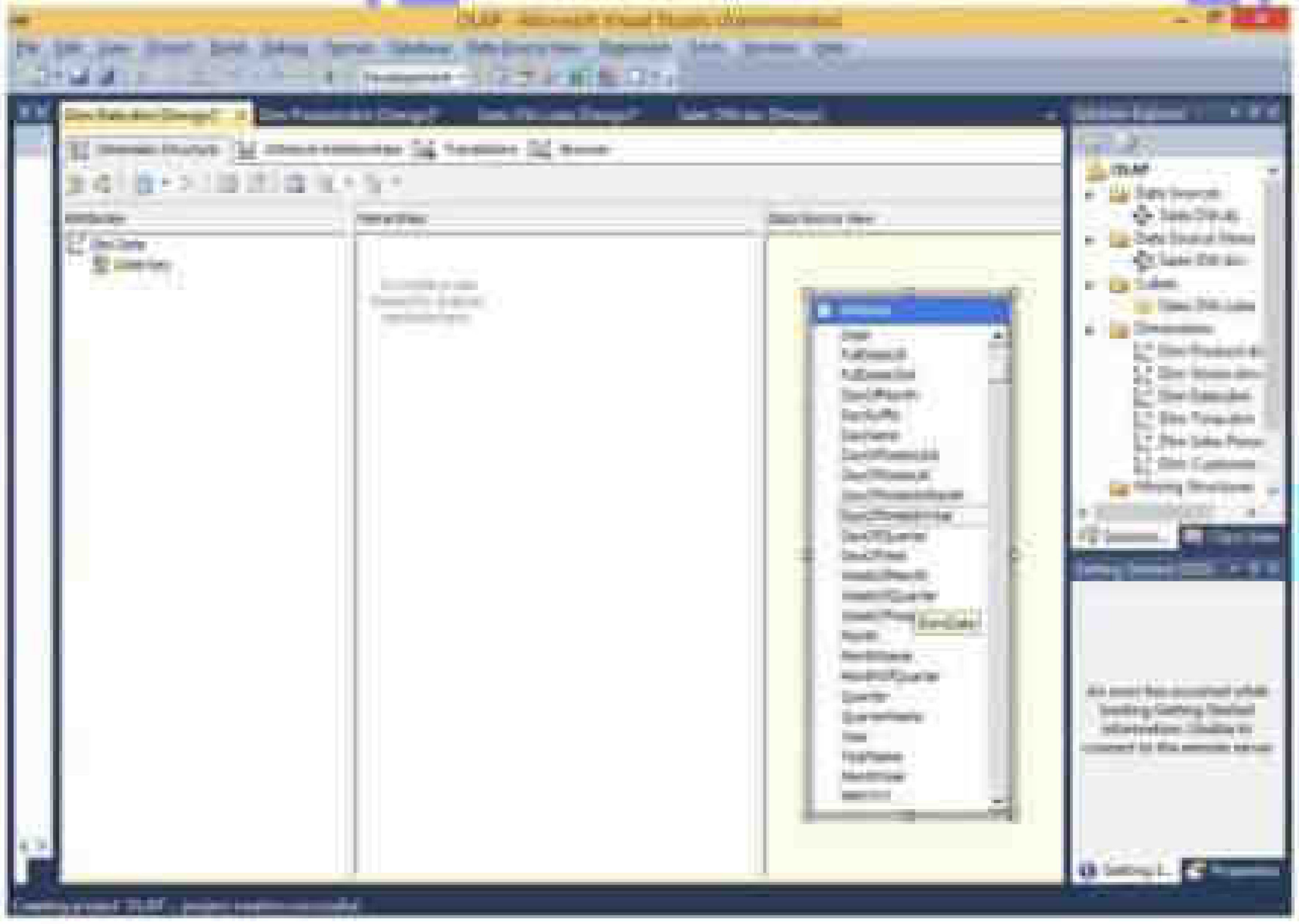
Drag and Drop Product Name from Table in Data Source View and Add in Attribute Pane at left side

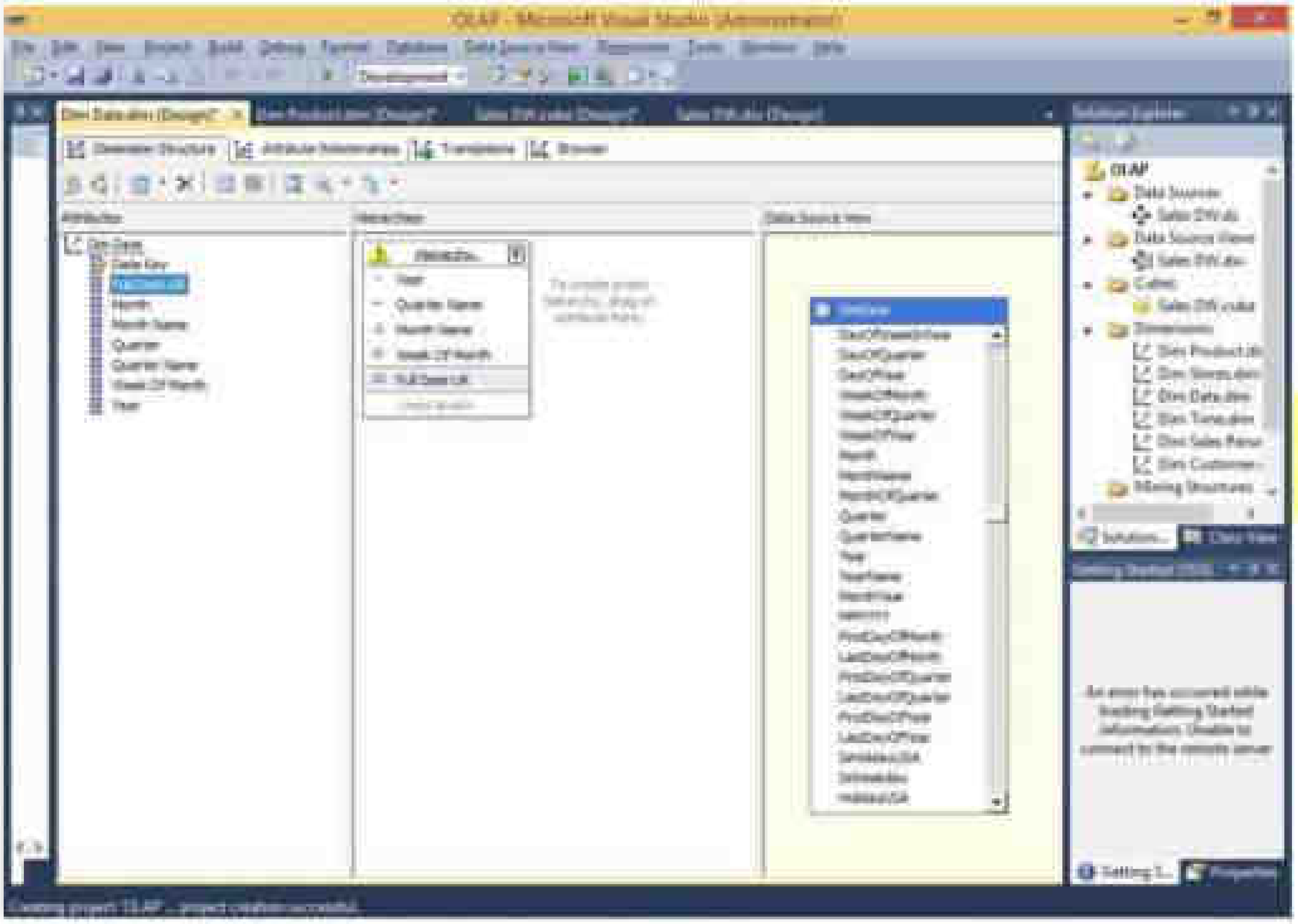


Step 6: Creating Attribute Hierarchy in Date Dimension

Double click On Dim Date dimension -> Drag and Drop Fields from Table shown in Data Source View to Attributes-> Drag and Drop attributes from leftmost pane of attributes to middle pane of Hierarchy.

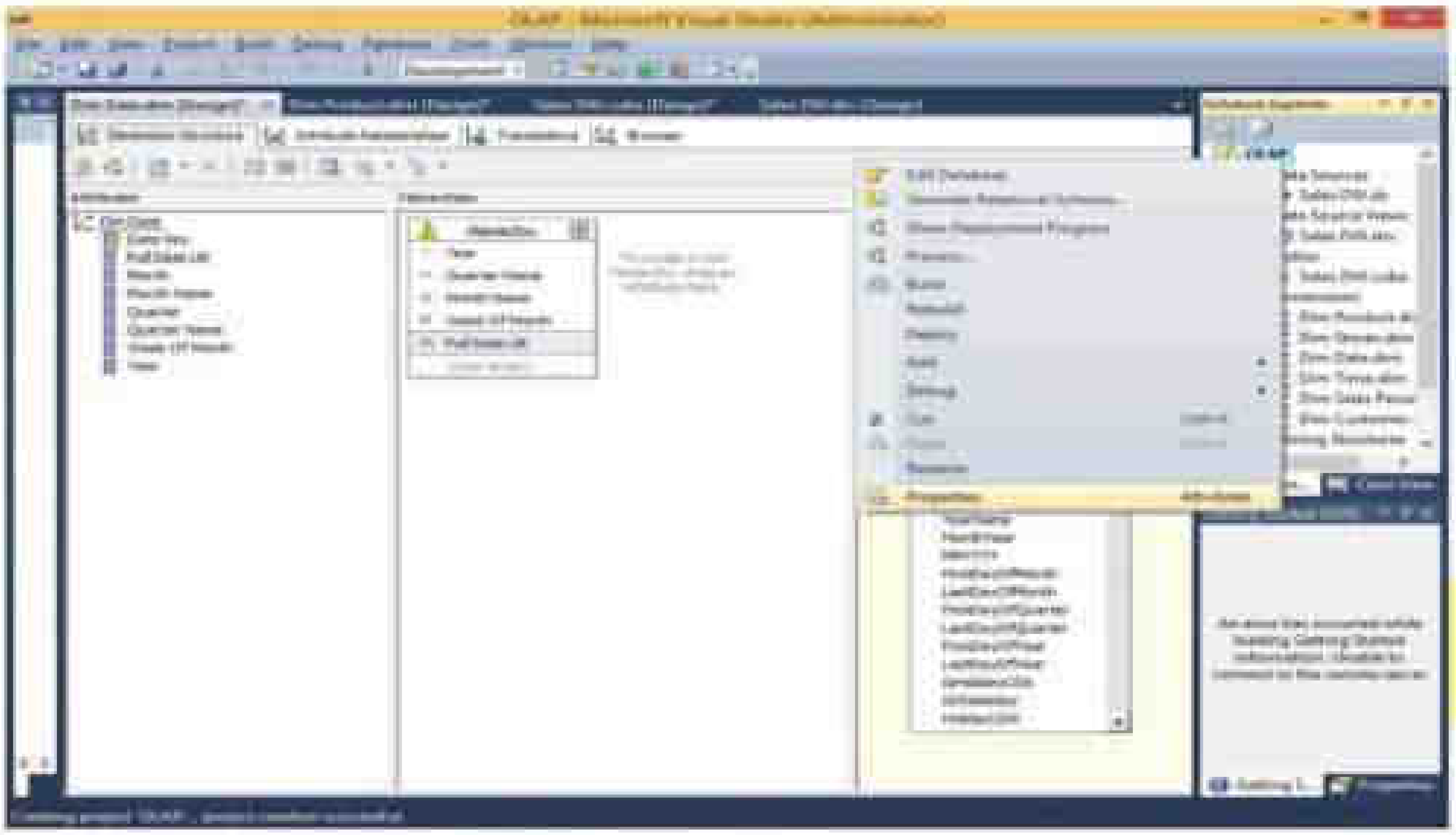
Drag fields in sequence from Attributes to Hierarchy window (Year, Quarter Name, Month Name, Week of the Month, Full Date UK)



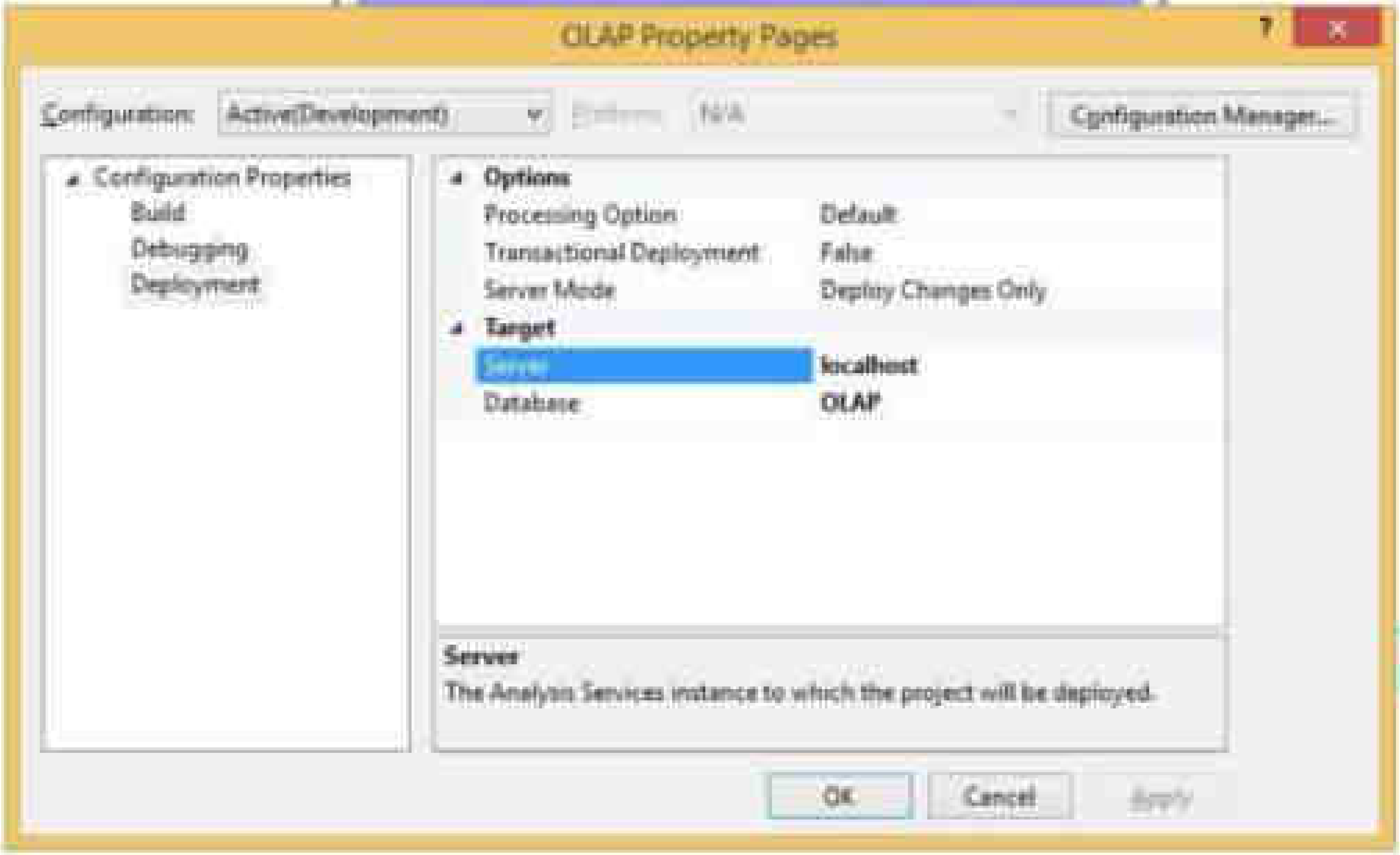


Step 7: Deploy Cube

Right click on Project name → Properties



This window appaers


OLAP Property Pages

Configuration: Active(Development) Platform: N/A Configuration Manager...

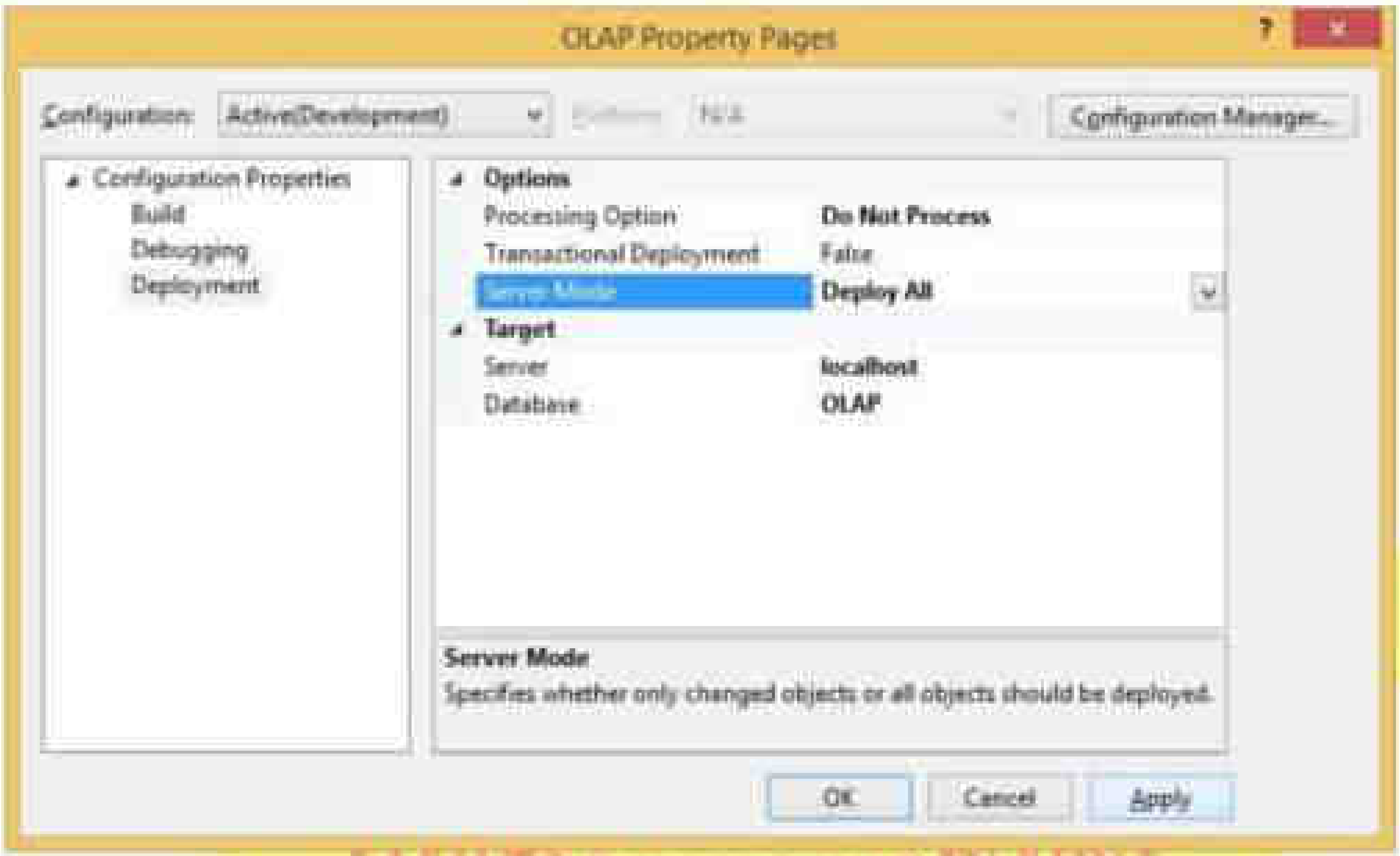
- Configuration Properties
  - Build
  - Debugging
  - Deployment

Options
Processing Option: Default
Transactional Deployment: False
Server Mode: Deploy Changes Only

Target
Server: localhost
Database: OLAP

Server
The Analysis Services instance to which the project will be deployed.

Do following changes and click on Apply & ok



Options

Processing Option	Do Not Process
Transactional Deployment	False
Server Mode	Deploy All

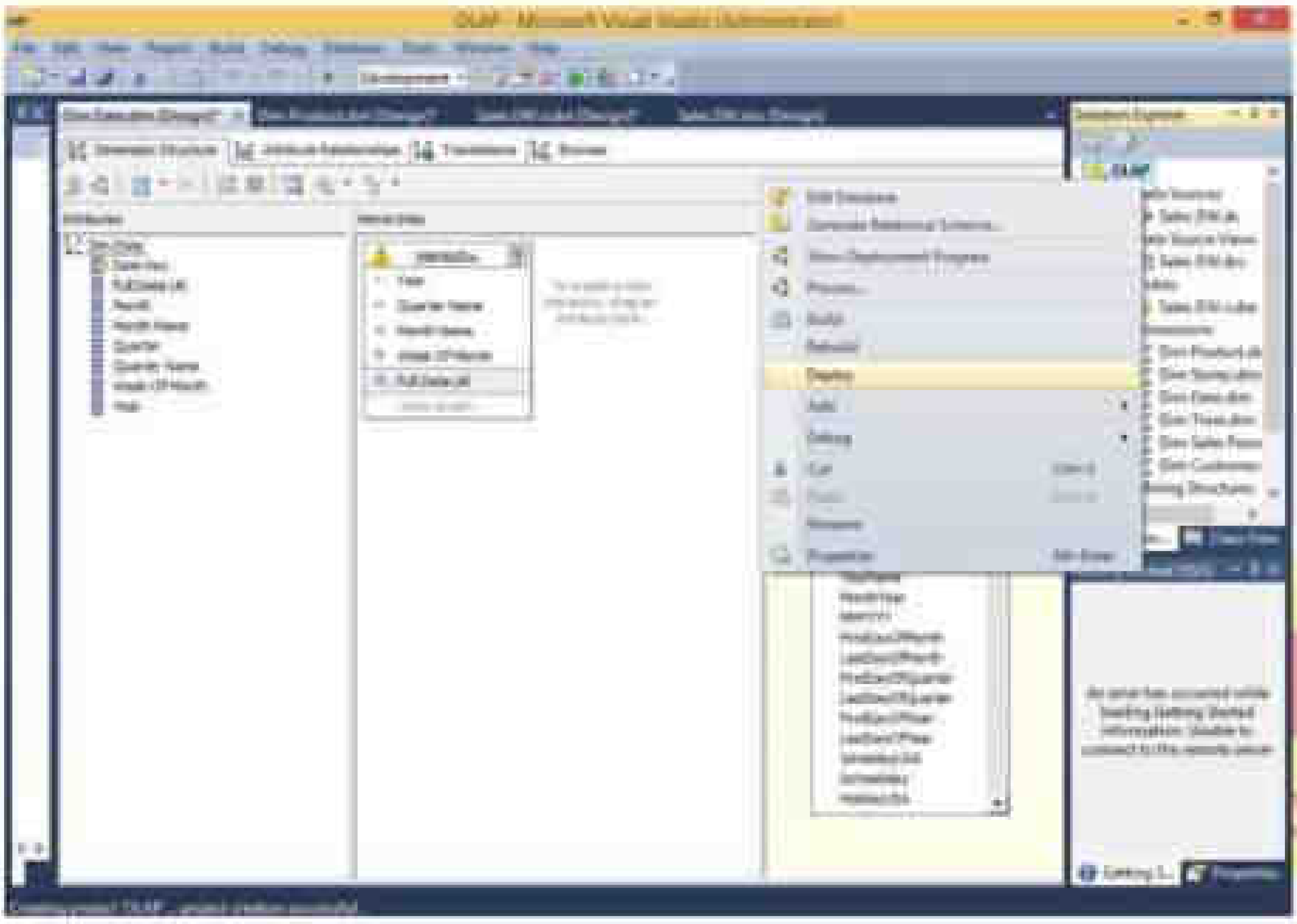
Target

Server	localhost
Database	OLAP

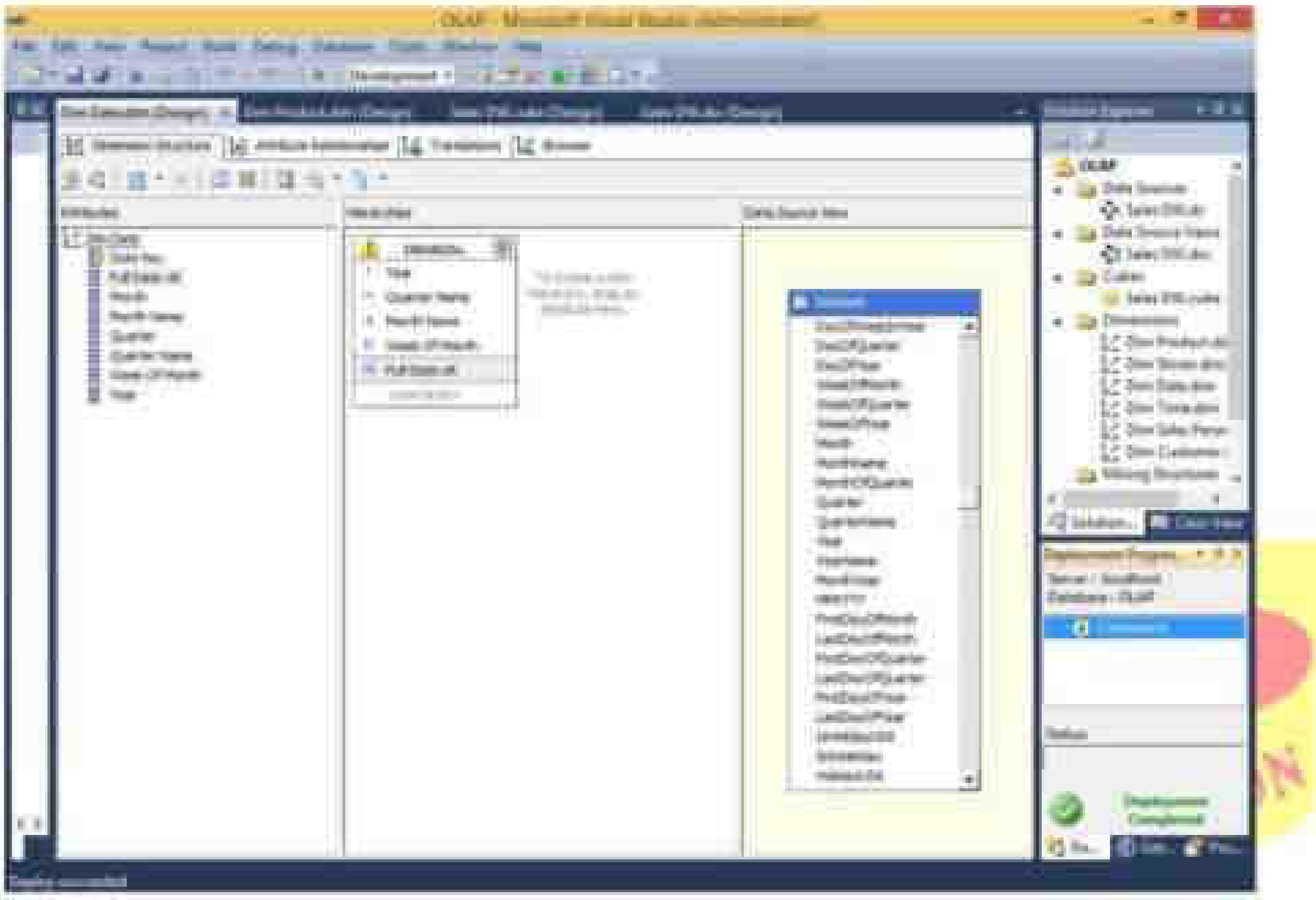
Server Mode

Specifies whether only changed objects or all objects should be deployed.

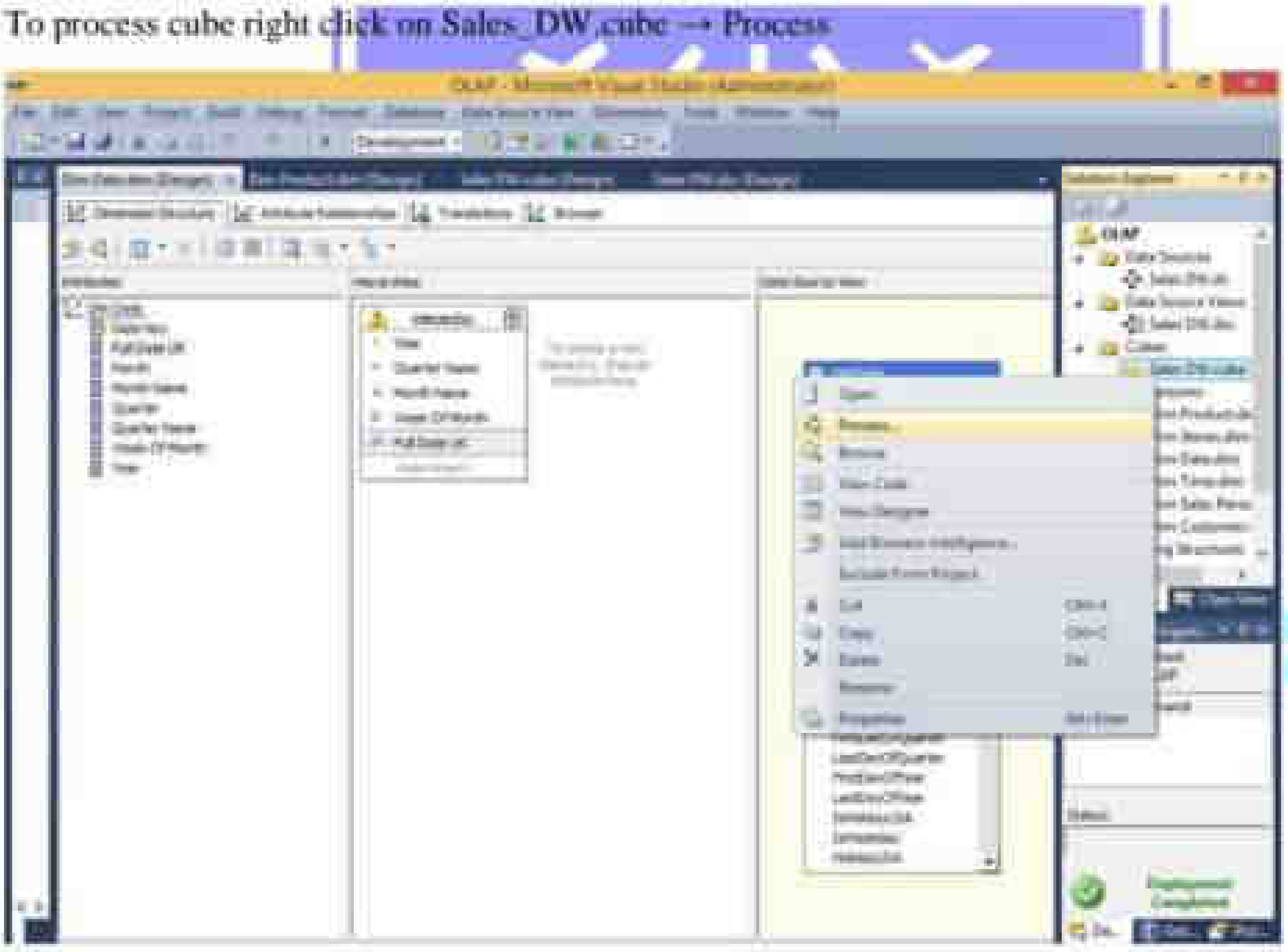
Right click on project name → Deploy



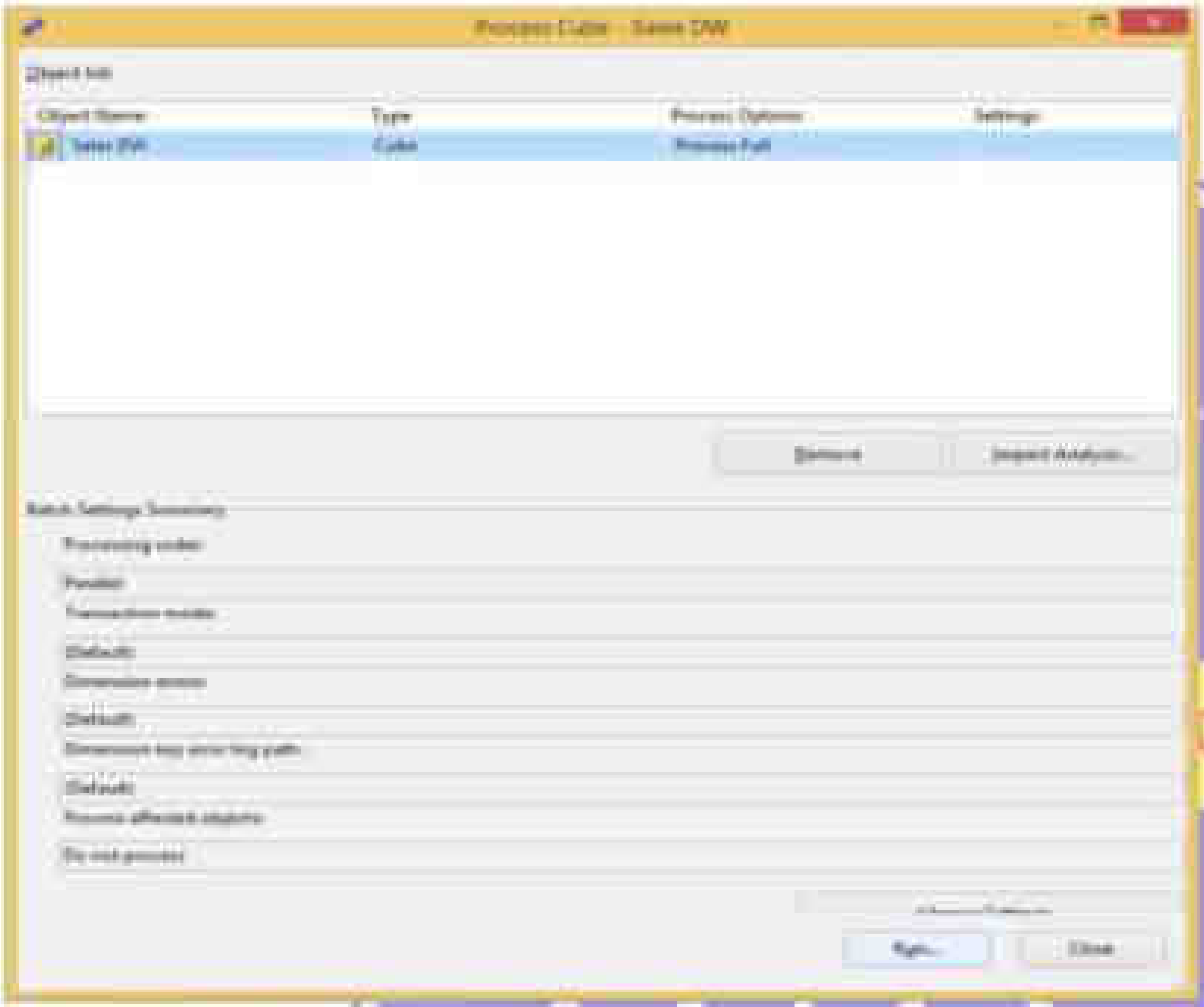
Deployment successful



To process cube right click on Sales_DW.cube → Process



Click run



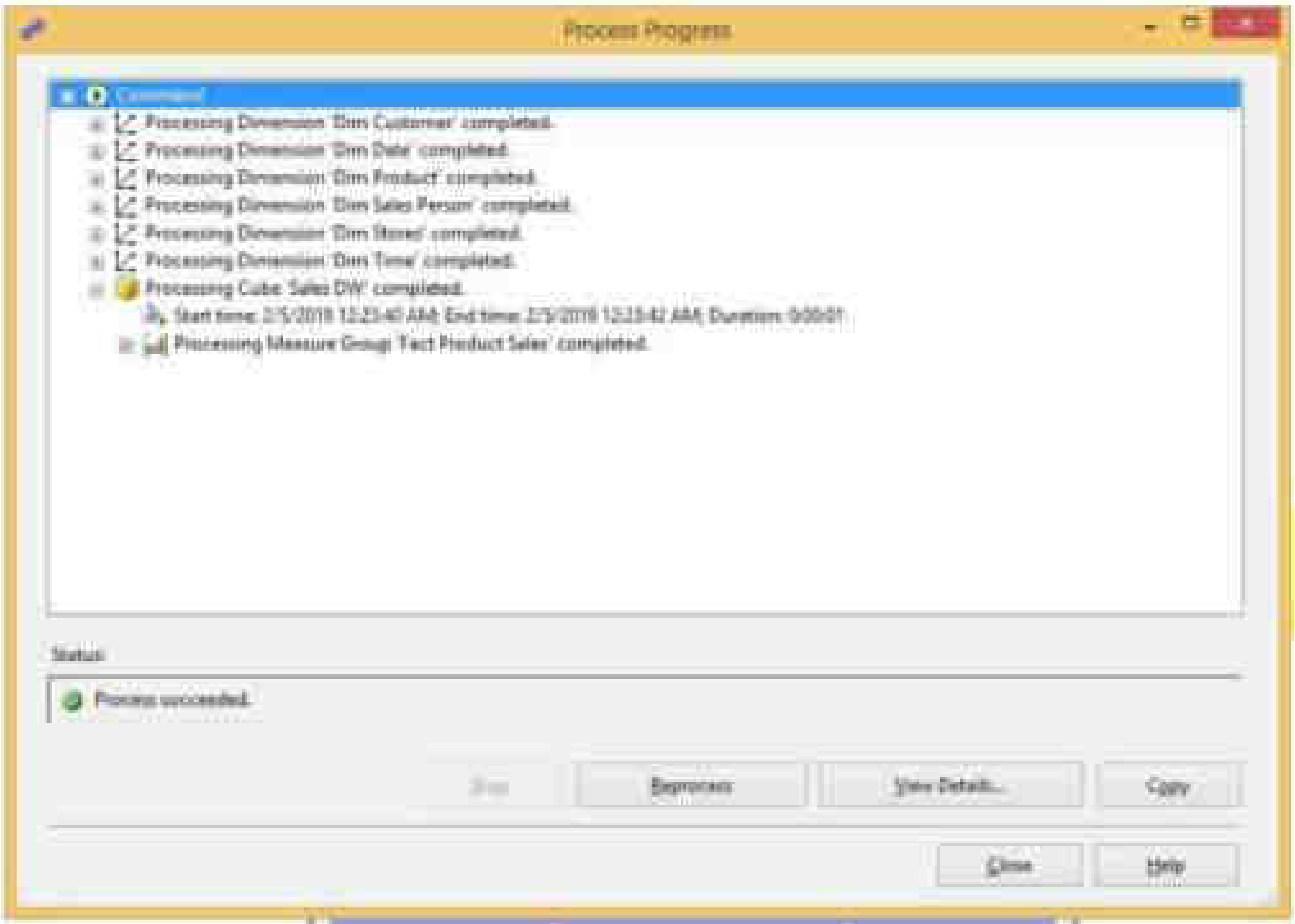
Process Cube - Sales DW

Object list

Object Name	Type	Process Options	Settings
Sales DW	Cube	Process Full	

Batch Settings Summary

Processing order:
Parallel
Transaction mode:
(Default)
Dimension errors:
(Default)
Dimension key error log path:
(Default)
Process affected objects:
Do not process

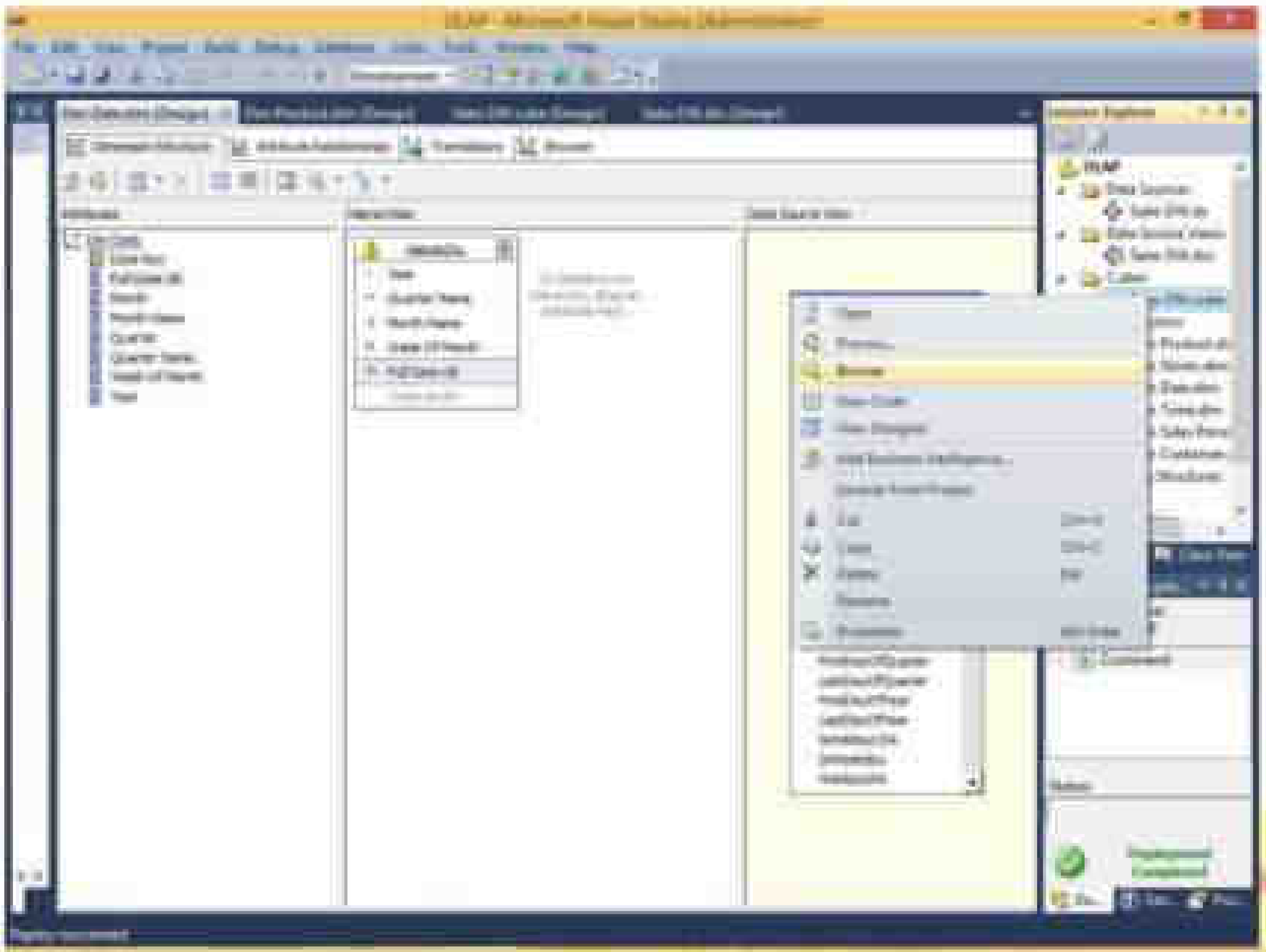


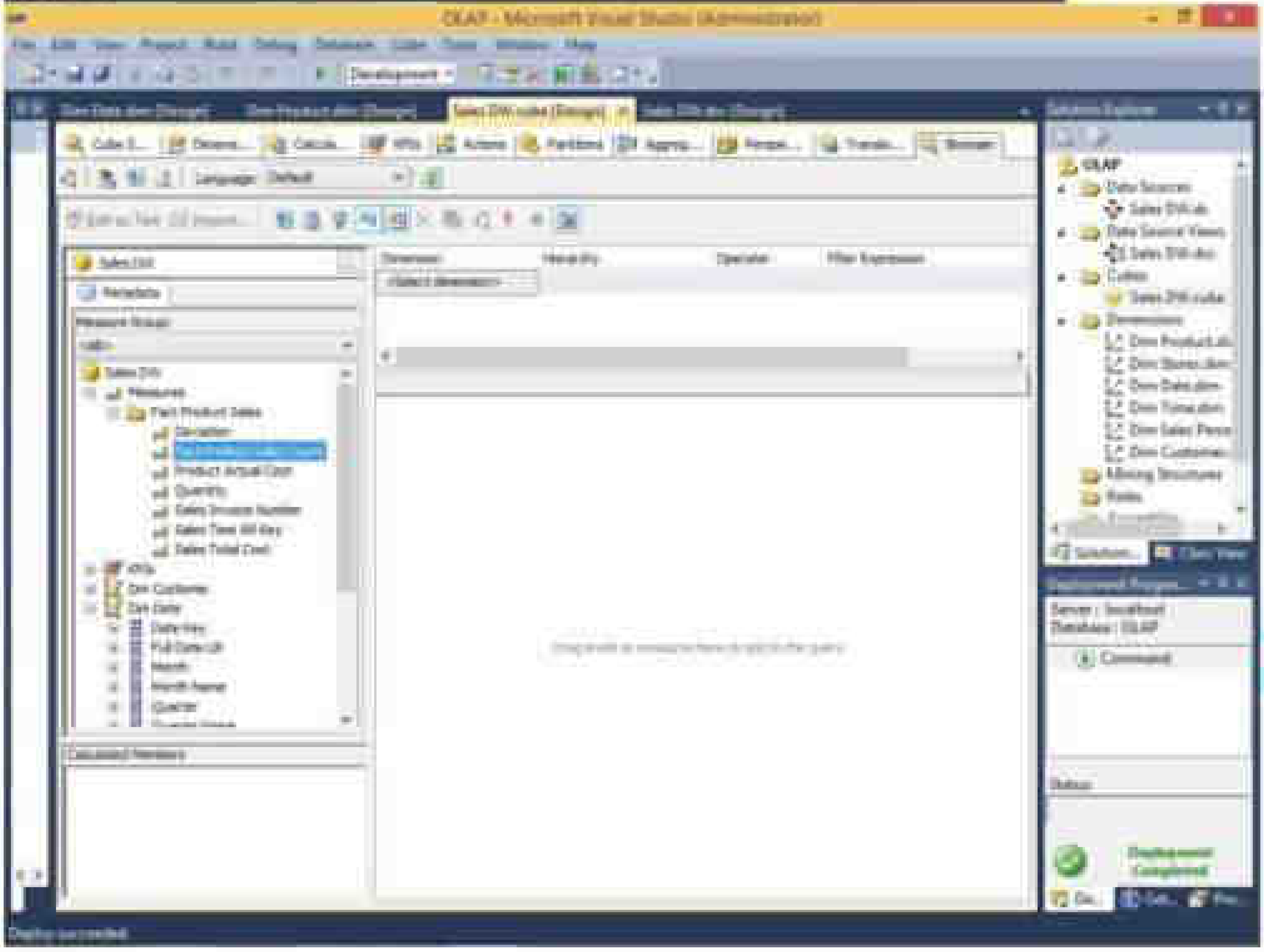
```
Command
  Processing Dimension 'Dim Customer' completed.
  Processing Dimension 'Dim Date' completed.
  Processing Dimension 'Dim Product' completed.
  Processing Dimension 'Dim Sales Person' completed.
  Processing Dimension 'Dim Stores' completed.
  Processing Dimension 'Dim Time' completed.
  Processing Cube 'Sales DW' completed.
    Start time: 2/5/2019 12:23:40 AM; End time: 2/5/2019 12:23:42 AM; Duration: 0:00:01
    Processing Measure Group 'Fact Product Sales' completed.
```

Status:

Process succeeded.

Browse the cube for analysis in solution explorer


OLAP - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Database Cube Tools Window Help
Development
Dim Date.dim [Design]
Dim Product.dim [Design]
Sales DW.cube [Design]
Sales DW.dsv [Design]
Dimension Structure
Attribute Relationships
Translations
Browser
Attributes
Hierarchies
Data Source View
Dim Date
Date Key
Full Date UK
Month
Month Name
Quarter
Quarter Name
Week of Month
Year
Hierarchy
Year
Quarter Name
Month Name
Week Of Month
Full Date UK
Open
Process...
Browse
View Code
View Designer
Add Business Intelligence...
Cut
Copy
Delete
Rename
Properties
Solution Explorer
OLAP
Data Sources
Sales DW.ds
Data Source Views
Sales DW.dsv
Cubes
Sales DW.cube
Dimensions
Dim Product.dim
Dim Stores.dim
Dim Date.dim
Dim Time.dim
Dim Sales Person
Dim Customer
Mining Structures
Status
Deployment Completed
Deploy succeeded


OLAP - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Database Cube Tools Window Help
Development
Dim Date.dim [Design]
Dim Product.dim [Design]
Sales DW.cube [Design]
Sales DW.dsv [Design]
Cube S...
Dimens...
Calcula...
KPIs
Actions
Partitions
Aggreg...
Perspe...
Transla...
Browser
Language: Default
Edit as Text
Import...
Sales DW
Metadata
Measure Group:
<All>
Sales DW
Measures
Fact Product Sales
Deviation
Product Actual Cost
Quantity
Sales Invoice Number
Sales Time Alt Key
Sales Total Cost
KPIs
Dim Customer
Dim Date
Date Key
Full Date UK
Month
Month Name
Quarter
Calculated Members
Dimension
Hierarchy
Operator
Filter Expression
<Select dimension>
Drag levels or measures here to add to the query.
Solution Explorer
OLAP
Data Sources
Sales DW.ds
Data Source Views
Sales DW.dsv
Cubes
Sales DW.cube
Dimensions
Dim Product.dim
Dim Stores.dim
Dim Date.dim
Dim Time.dim
Dim Sales Person
Dim Customer
Mining Structures
Roles
Solution...
Class View
Deployment Progress
Server : localhost
Database : OLAP
Command
Status
Deployment Completed
Deploy succeeded

Learning Outcomes:

Course Outcomes:

Conclusion:

Viva Questions:

1. **Define staging.**
2. **Why is staging important?**
3. **Name one staging component.**
4. **How to validate data in staging?**

For Faculty use:

Correction Parameters	Formative Assessment[40%]	Timely Completion of Practical[40%]	Attendance Learning Attitude[20%]

Practical-4: a. Create the ETL map and setup the schedule for execution.
– b. Execute the MDX queries to extract the data from data ware house.

Aims:

1. To design an ETL map outlining the flow of data from source systems to the data warehouse.
2. To set up and schedule the ETL process for automated execution.
3. To execute MDX queries for extracting and analyzing data from the data warehouse.

Learning Objectives:

1. Understand the principles of ETL mapping and how it drives data integration.
2. Gain hands-on experience in scheduling ETL jobs to ensure timely data processing.
3. Learn to write and execute MDX queries to extract multidimensional data for reporting and analysis.
4. Evaluate the effectiveness and performance of the ETL process and query results.

Tool & Technologies used:

1. Power BI is a powerful business intelligence tool used for data visualization and analysis.

Theory-4: ETL

ETL (Extract, Transform, Load) is a core process in data warehousing that integrates data from multiple sources into a centralized repository. This theory focuses on two key aspects: designing an ETL map with an execution schedule and using MDX queries to extract data from the data warehouse.

Creating the ETL Map and Scheduling Execution

1. **Designing the ETL Map**
   - **Source Identification:** Identify all relevant data sources (databases, files, APIs) and understand their formats and structures.
   - **Mapping Data Flow:** Create a visual map that outlines the flow of data from source systems to the staging area, detailing transformation steps and the final load into the data warehouse.
   - **Defining Transformation Rules:** Document the business rules needed for data cleansing, conversion, aggregation, and enrichment.
   - **Error Handling:** Establish procedures for logging errors, issuing alerts, and rolling back transactions if issues occur during the ETL process.
2. **Scheduling the ETL Process**
   - **Determining Frequency:** Decide whether the ETL jobs should run in batch mode (e.g., nightly, hourly) or in real time, based on data freshness requirements and system load.
   - **Automation:** Utilize ETL automation tools (such as SQL Server Integration Services, Informatica, or Apache NiFi) to schedule and execute jobs reliably.

    - **Monitoring and Alerts:** Set up monitoring dashboards and notification systems to track job performance and address any failures promptly.

Executing MDX Queries to Extract Data from the Data Warehouse

1. **Understanding MDX:**
    - MDX (Multidimensional Expressions) is a query language designed for OLAP (Online Analytical Processing) systems. It enables the retrieval and analysis of multidimensional data stored in cubes.
    - MDX queries help extract aggregated, detailed, or time-series data across various dimensions like product, geography, and time.
2. **Running MDX Queries:**
    - **Connection:** Establish a secure connection to the OLAP data warehouse.
    - **Query Construction:** Build MDX queries to retrieve relevant data. For example, a query to extract total sales by product category for the year 2024 might look like this:
    - **Result Analysis:** Analyze the query results to generate actionable insights, support reporting, and enable dynamic data visualizations.

PRACTICAL 4 b

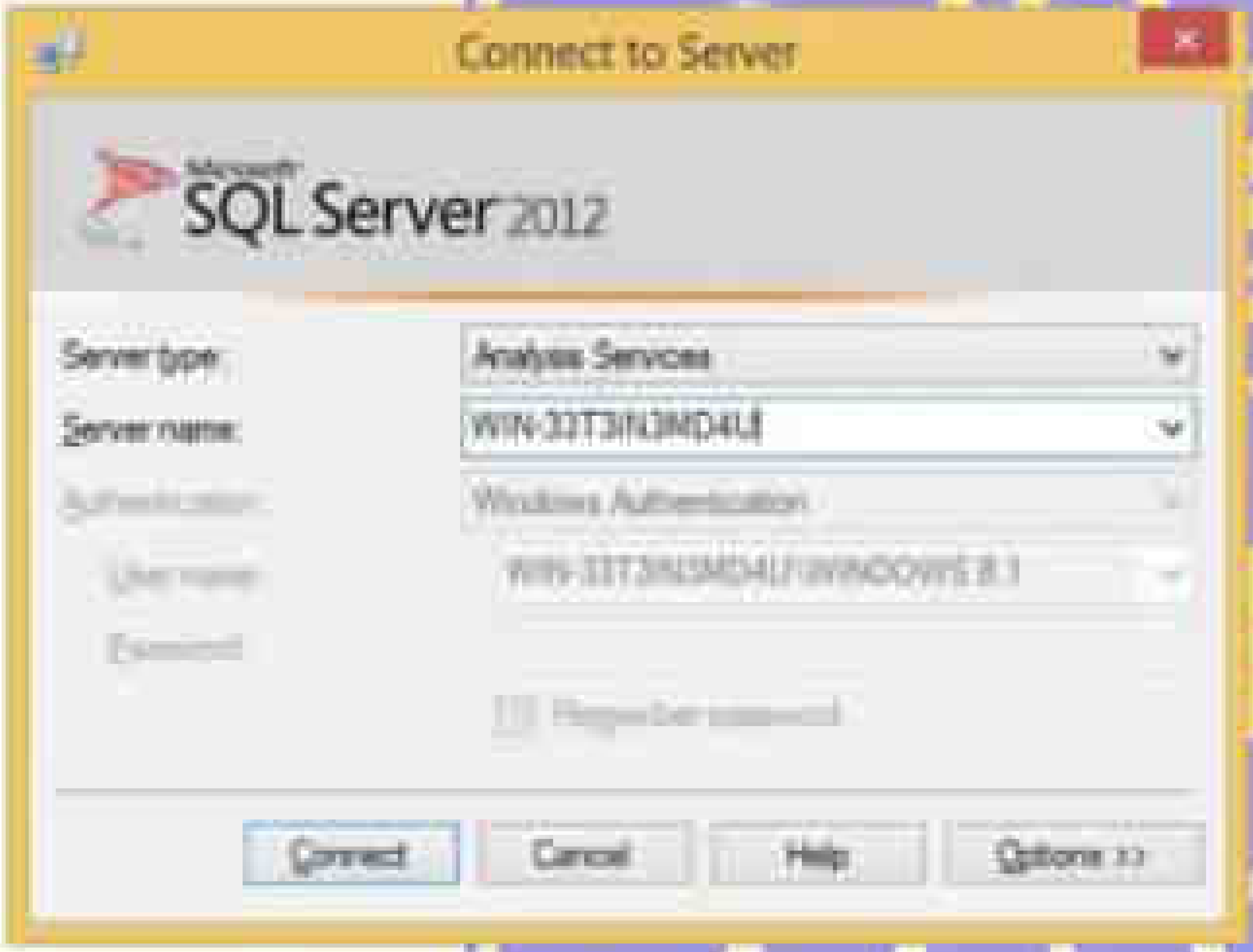
Execute the MDX queries to extract the data from the datawarehouse.

Step 1: Open SQL Server Management Studio and connect to Analysis Services.

Server type: Analysis Services

Server Name: (according to base machine)

Click on connect



Server type: Analysis Services

Server name: WIN-33T3IN3MD4U

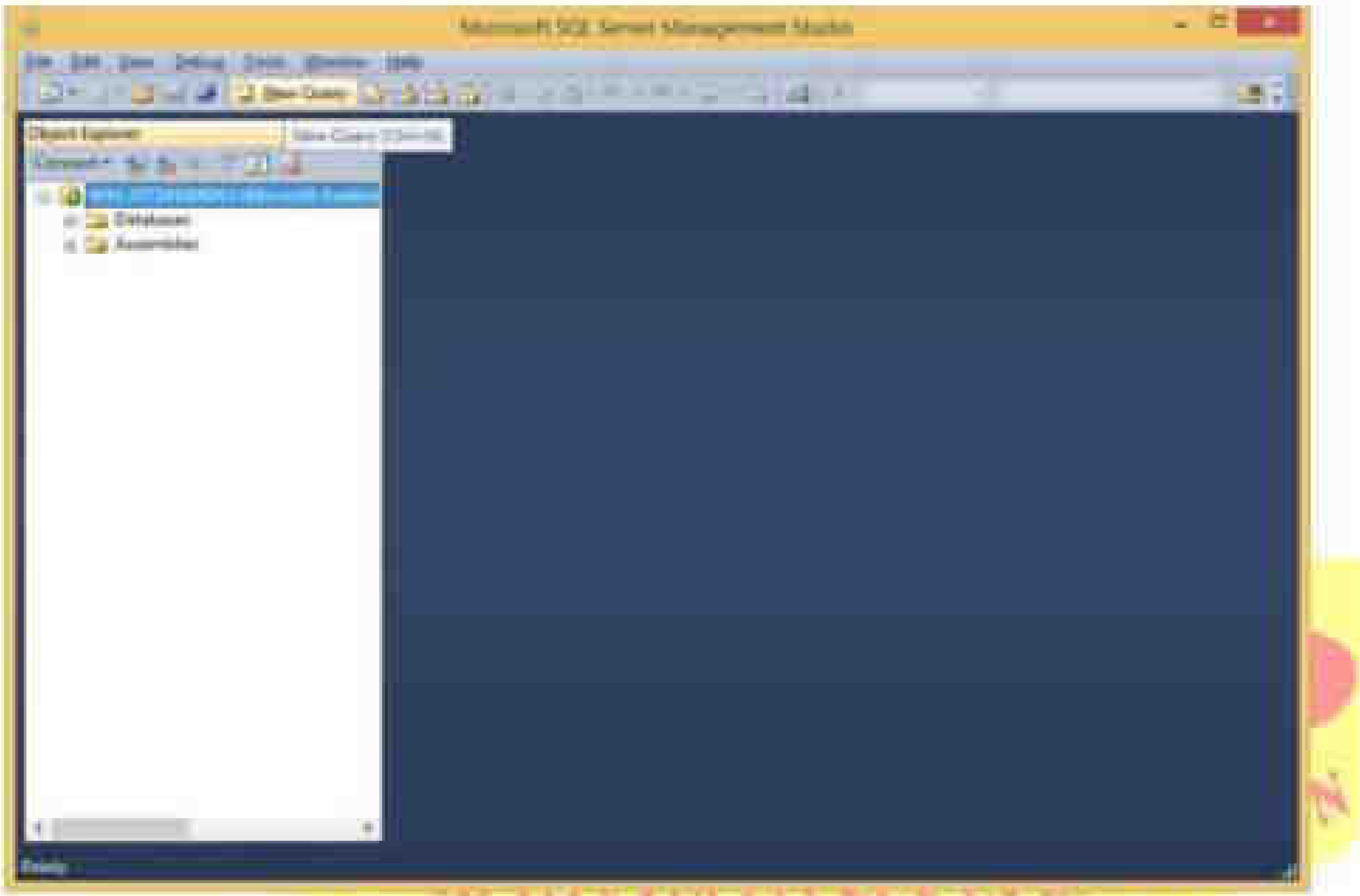
Authentication: Windows Authentication

User name: WIN-33T3IN3MD4U\WINDOWS 8.1

Password:

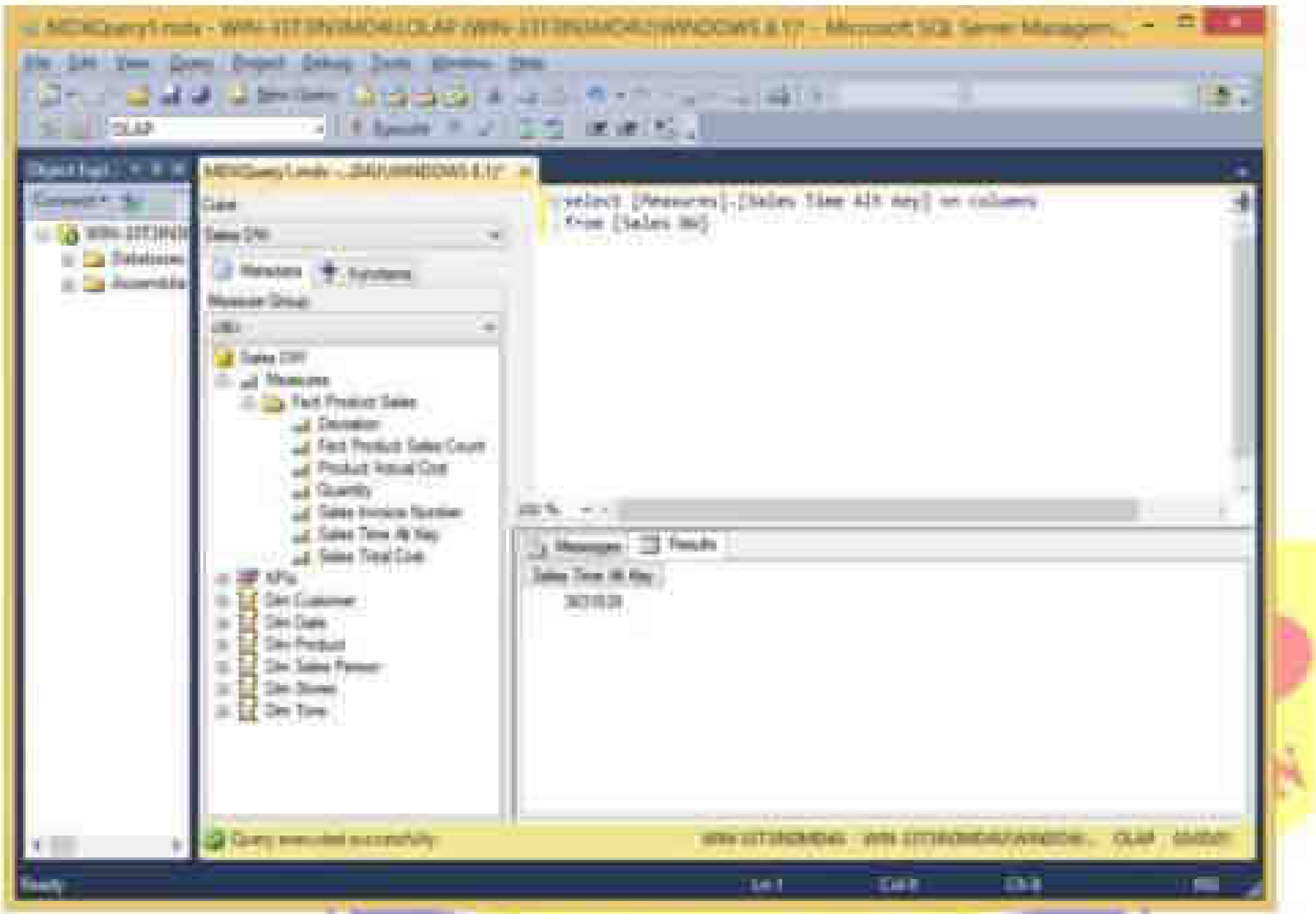
□ Remember password

Step 2: Click on New Query & type following query based on Sales_DW


Microsoft SQL Server Management Studio
File Edit View Debug Tools Window Help
New Query
Object Explorer
New Query (Ctrl+N)
Connect
Databases
Assemblies
Ready

select [Measures].[Sales Time Alt Key] on columns

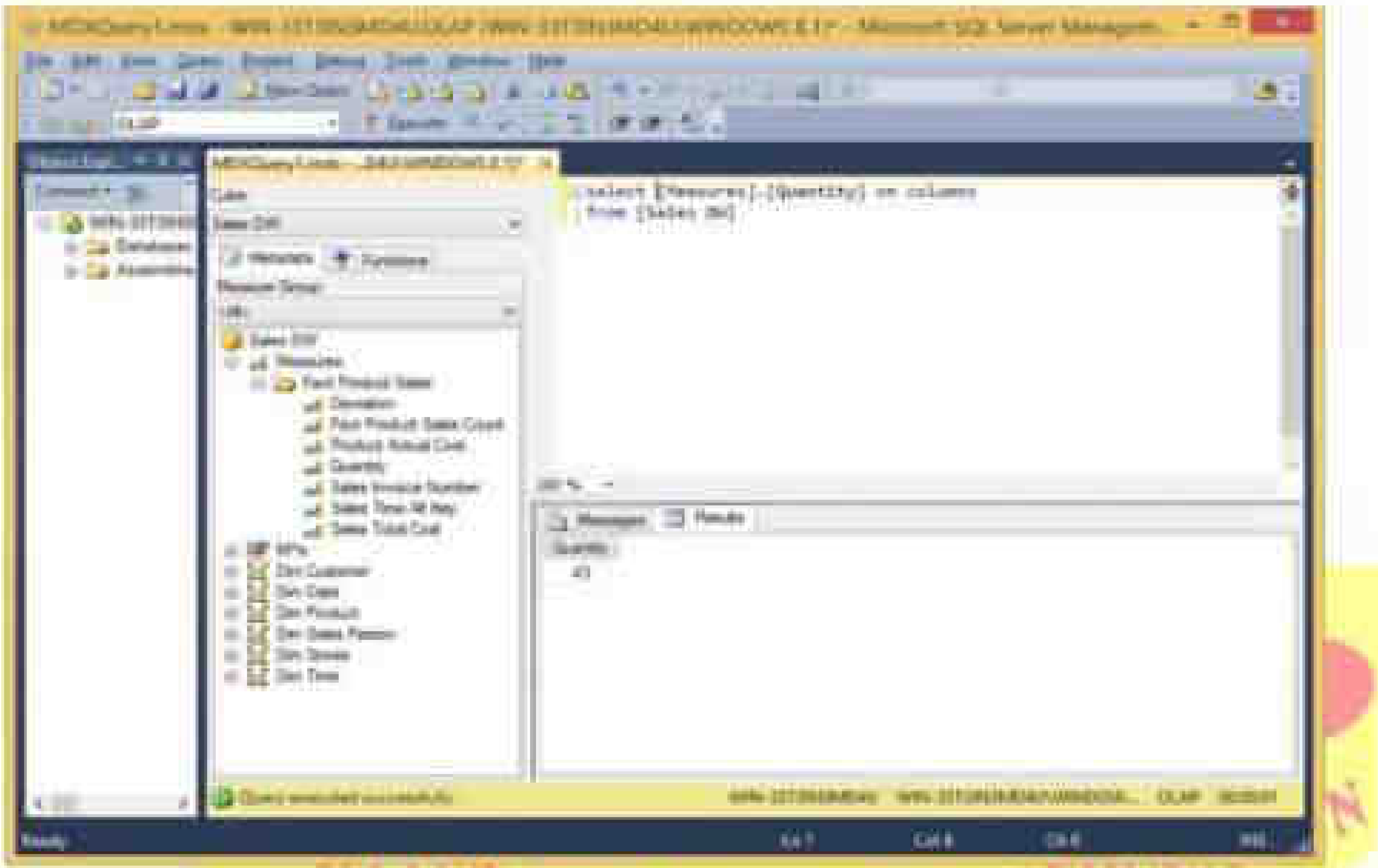
from [Sales DW] Click on execute



```
select [Measures].[Sales Time Alt Key] on columns
from [Sales DW]
```

Sales Time Alt Key
[illegible]

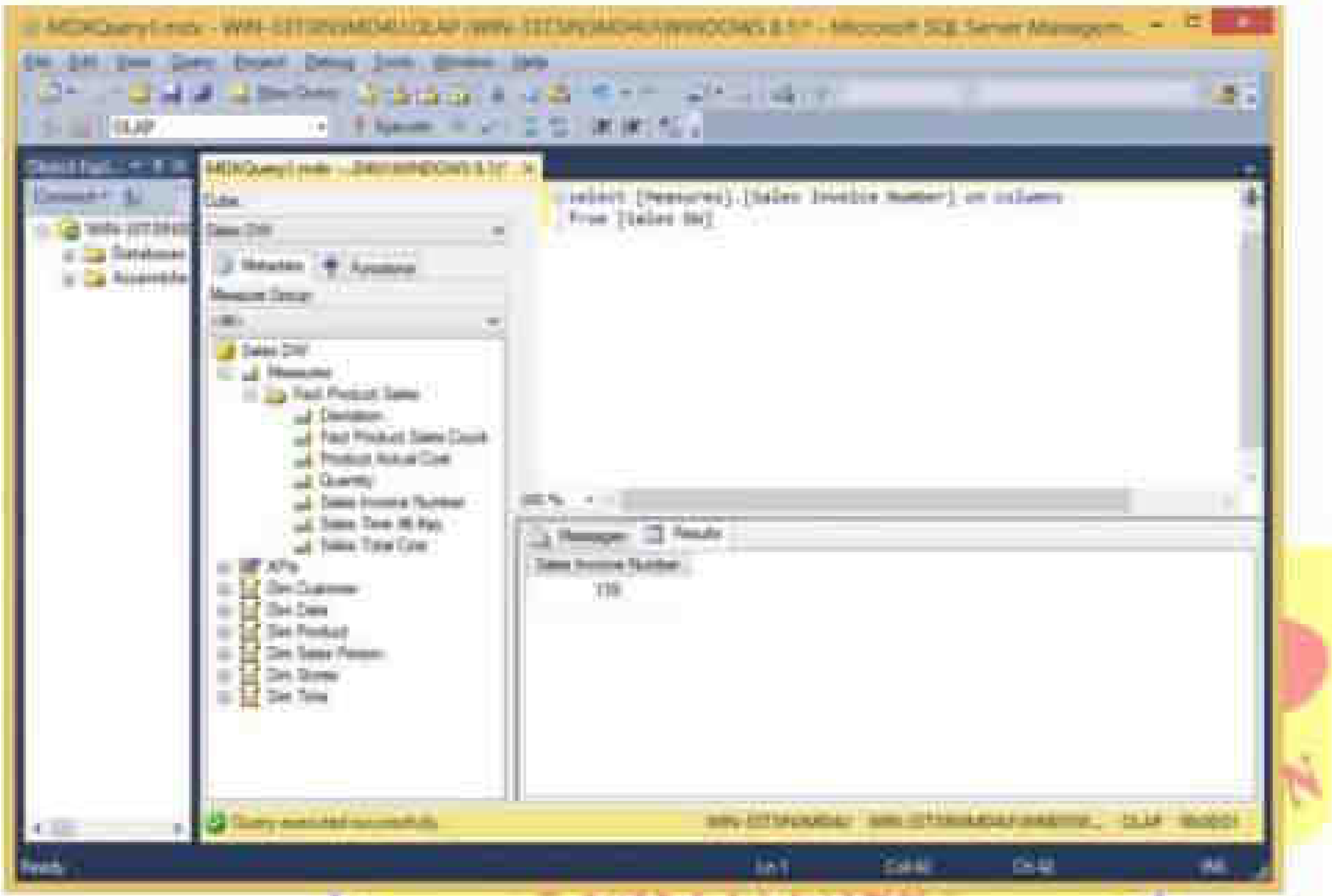
select [Measures].[Quantity] on columns from [Sales DW]



```
select [Measures].[Quantity] on columns
from [Sales DW]
```

```
Quantity
43
```

select [Measures].[Sales Invoice Number] on columns from [Sales DW]

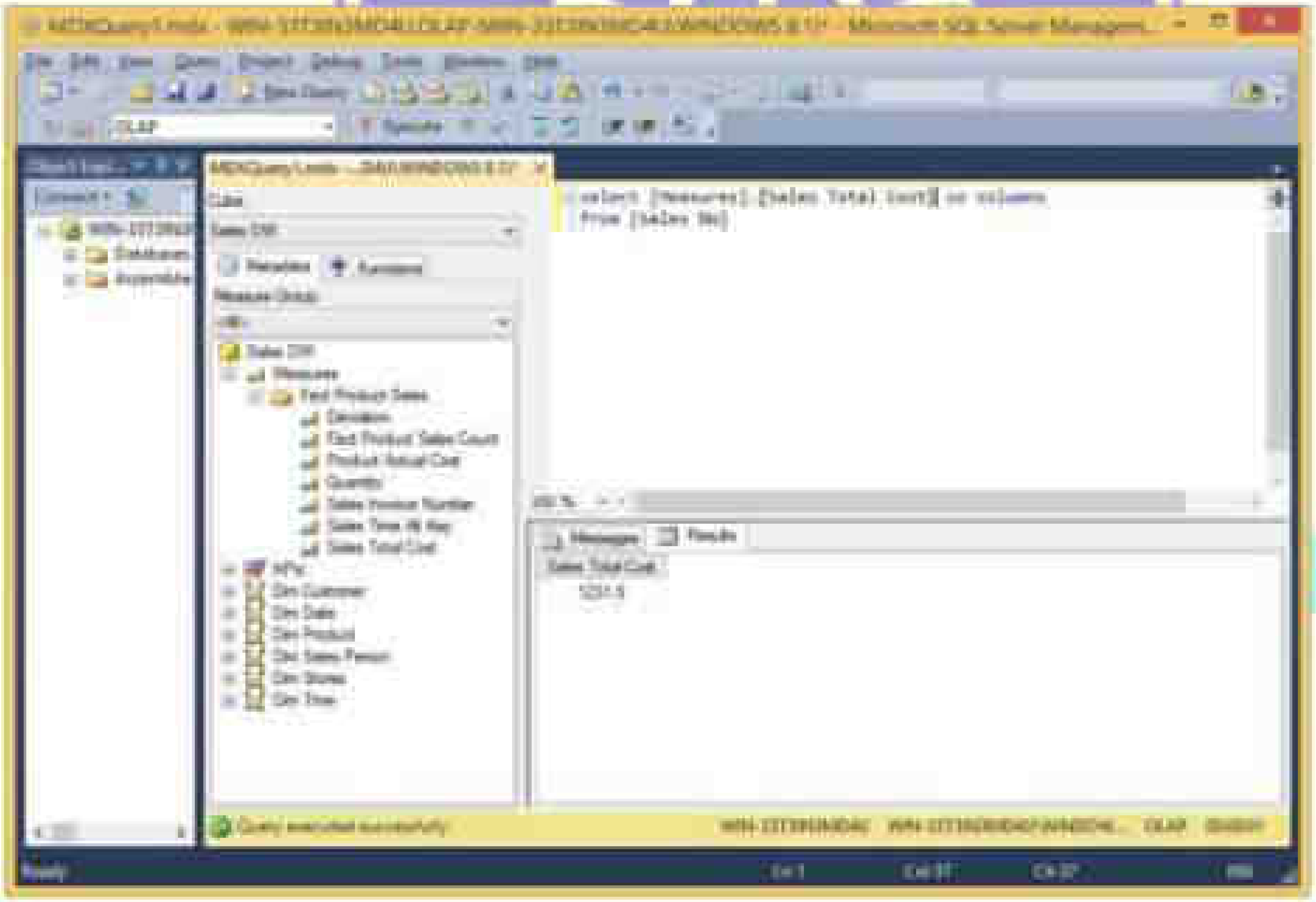


```
select [Measures].[Sales Invoice Number] on columns
from [Sales Dw]
```

Sales Invoice Number
119

Query executed successfully.

select [Measures].[Sales Total Cost] on columns from [Sales DW]

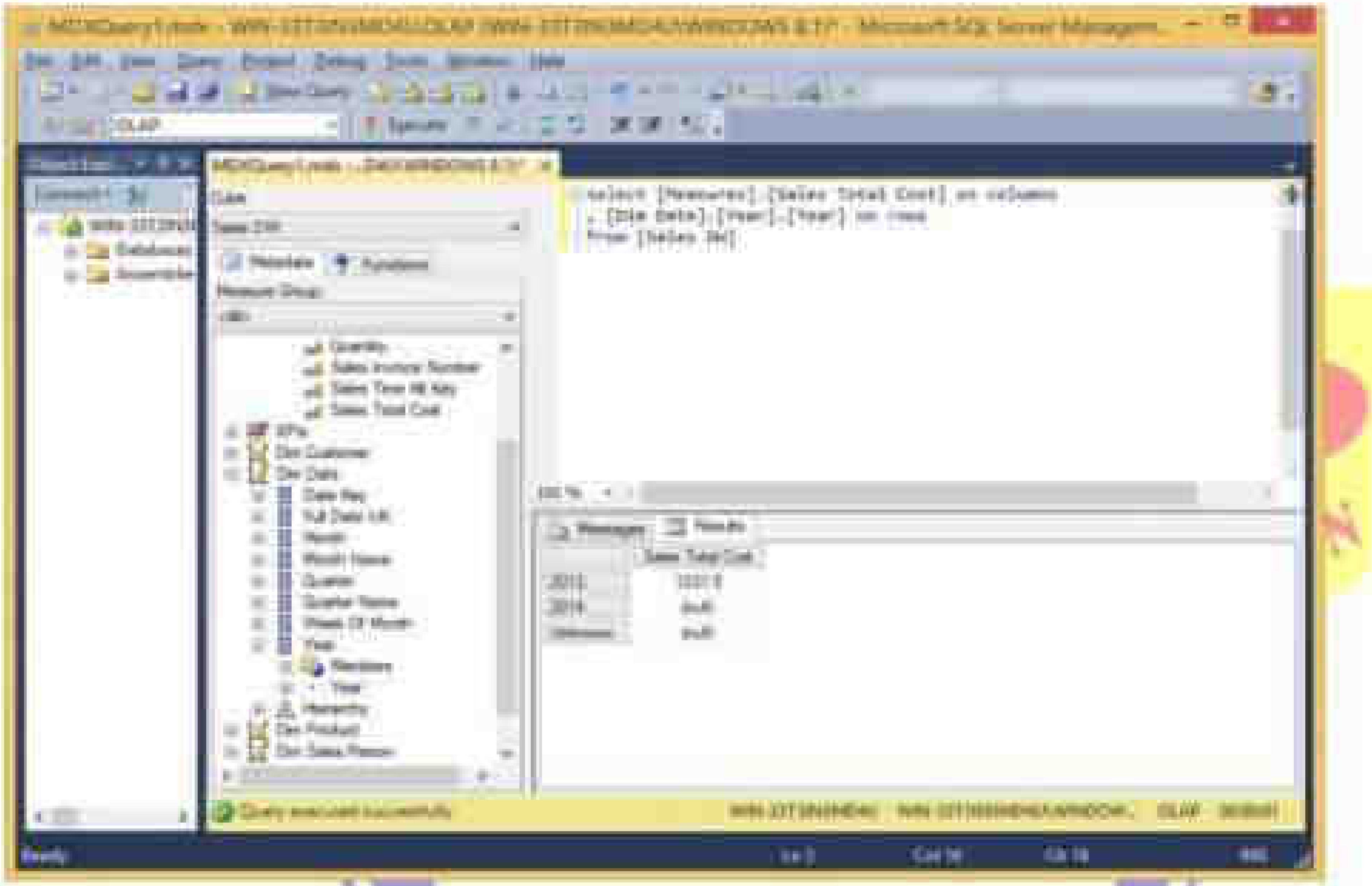


```
select [Measures].[Sales Total Cost] on columns
from [Sales Dw]
```

Sales Total Cost
1251.5

Query executed successfully.

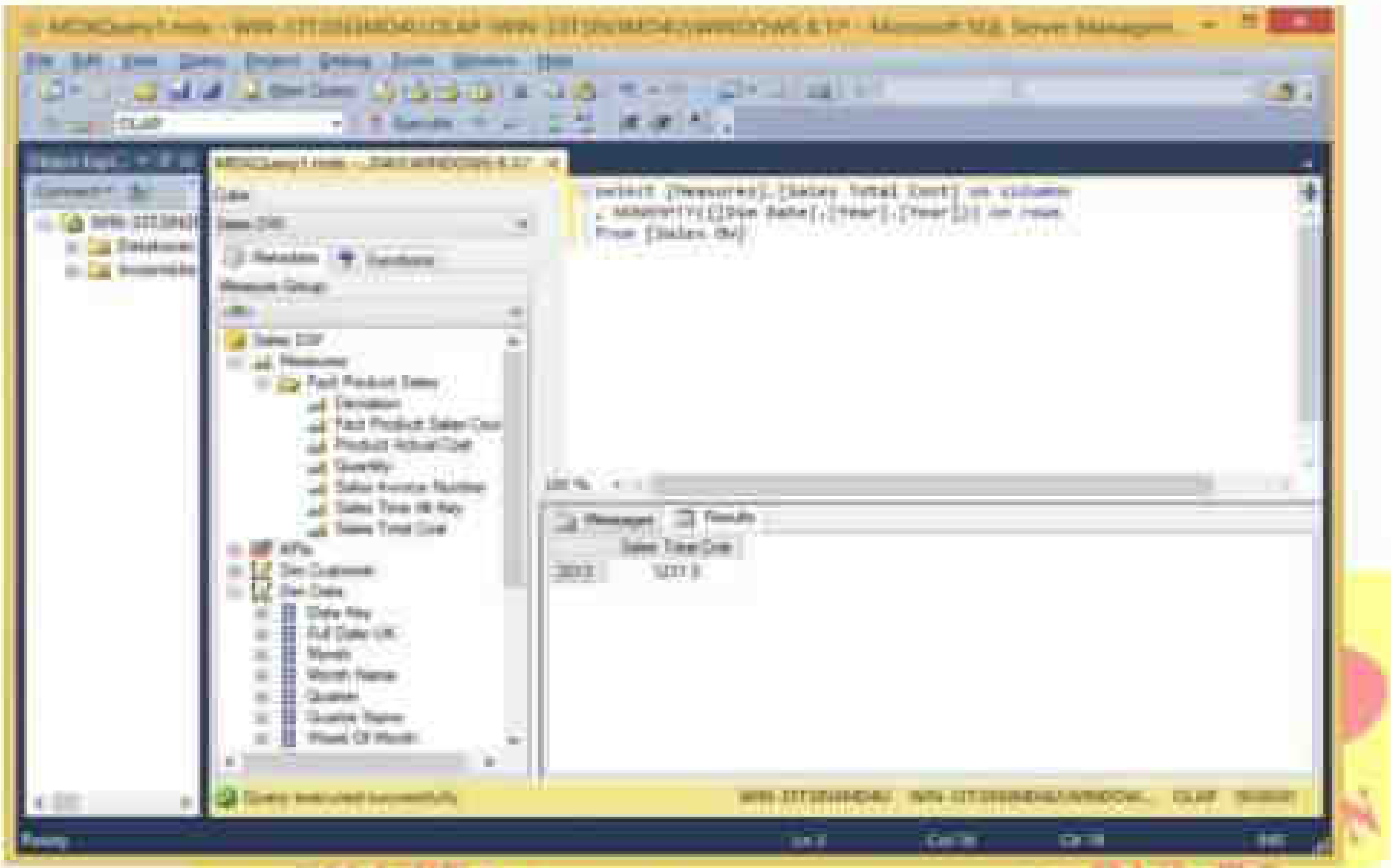
```
select [Measures].[Sales Total Cost] on columns
, [Dim Date].[Year].[Year] on rows  from [Sales DW]
```



```
select [Measures].[Sales Total Cost] on columns
, [Dim Date].[Year].[Year] on rows
from [Sales DW]
```

	Sales Total Cost
2013	[illegible]
2014	(null)
Unknown	(null)

```
select [Measures].[Sales Total Cost] on columns , NONEMPTY({[Dim
Date].[Year].[Year]}) on rows  from [Sales DW]
```

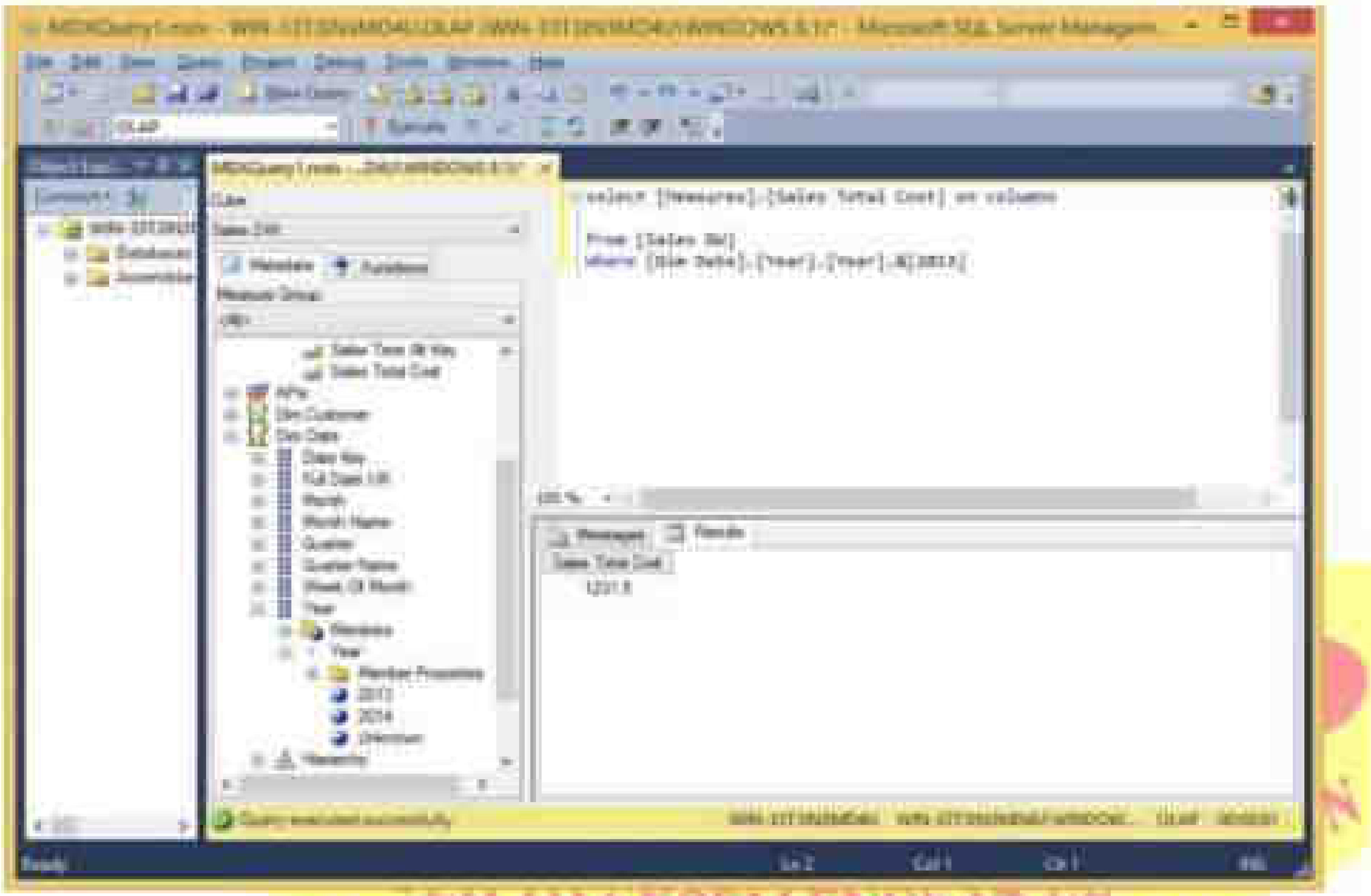


```
select [Measures].[Sales Total Cost] on columns
, NONEMPTY([Dim Date].[Year].[Year]) on rows
From [Sales DW]
```

```
        Sales Total Cost
2013    5217.5
```

select [Measures].[Sales Total Cost] on columns from [Sales DW]

Where [Dim Date].[Year].[Year].&[2013]



```
select [Measures].[Sales Total Cost] on columns

from [Sales DW]
where [Dim Date].[Year].[Year].&[2013]
```

```
Sales Total Cost
    1231.8
```

Query executed successfully.

Learning Outcomes:

Course Outcomes:

Conclusion:

Viva Questions:

1. **What is ETL?**
2. **Name the three steps in ETL.**
3. **Why is ETL important?**
4. **Name an ETL tool.**

For Faculty use:

Correction Parameters	Formative Assessment[40%]	Timely Completion of Practical[40%]	Attendance Learning Attitude[20%]

Practical-5:

a. Import the data ware house data in Micros Excel and create the Pivot table and PivotChart.

b. Import the cube in Microsoft Excel and create- the Pivot table and Pivot Chart to perform data analysis.

Aims:

1. To import data from a data warehouse into Microsoft Excel and create PivotTables and PivotCharts for effective data analysis.
2. To import a multidimensional cube into Excel and use its interactive features to analyze data through PivotTables and Pivot Charts.

Learning Objectives:

1. Understand how to connect Excel to a data warehouse and cube.
2. Learn to create and customize PivotTables and Pivot Charts to summarize and visualize data.
3. Develop skills in analyzing large datasets and deriving actionable insights using Excel's data analysis tools.

Tool & Technologies used:

1. Power BI is a powerful business intelligence tool used for data visualization and analysis.

Theory-5: Data ware house

Data warehousing consolidates data from various sources into a centralized repository, enabling efficient analysis and reporting. Microsoft Excel serves as a versatile tool for importing data, creating PivotTables, and generating PivotCharts, which empower users to explore and visualize complex datasets.

A. Importing Data Warehouse Data into Microsoft Excel

1. **Data Importation:**
   - Utilize Excel's Data tab to establish a connection to the data warehouse using options like ODBC, SQL Server, or other database connectors.
   - Configure the connection to securely access the required tables or views from the warehouse.
2. **Creating a PivotTable:**
   - Once the data is imported, go to the Insert tab and select PivotTable to initiate a new PivotTable based on the imported data.

- Organize the data by dragging and dropping fields into Rows, Columns, Values, and Filters, enabling a multidimensional view of the data.

3. **Generating a PivotChart:**
    - With the PivotTable in place, use the PivotChart feature to create a visual representation of your summarized data.
    - Customize the chart type (bar, line, pie, etc.) to effectively highlight key trends and insights.

B. Importing the OLAP Cube into Microsoft Excel

1. **Connecting to the Cube:**
    - In Excel, select Data > Get Data > From Other Sources > From Analysis Services to connect directly to an OLAP cube.
    - Input the necessary server and database credentials to access the multidimensional data.
2. **Building a PivotTable from the Cube:**
    - Excel will automatically present the cube's dimensions and measures.
    - Create a PivotTable by dragging cube fields into the respective areas, leveraging the inherent hierarchical structure for detailed analysis.
3. **Creating a PivotChart for Cube Data:**
    - After setting up the PivotTable, insert a PivotChart to visualize the aggregated cube data.
    - Utilize interactive features like drill-downs to explore underlying data and uncover hidden insights.

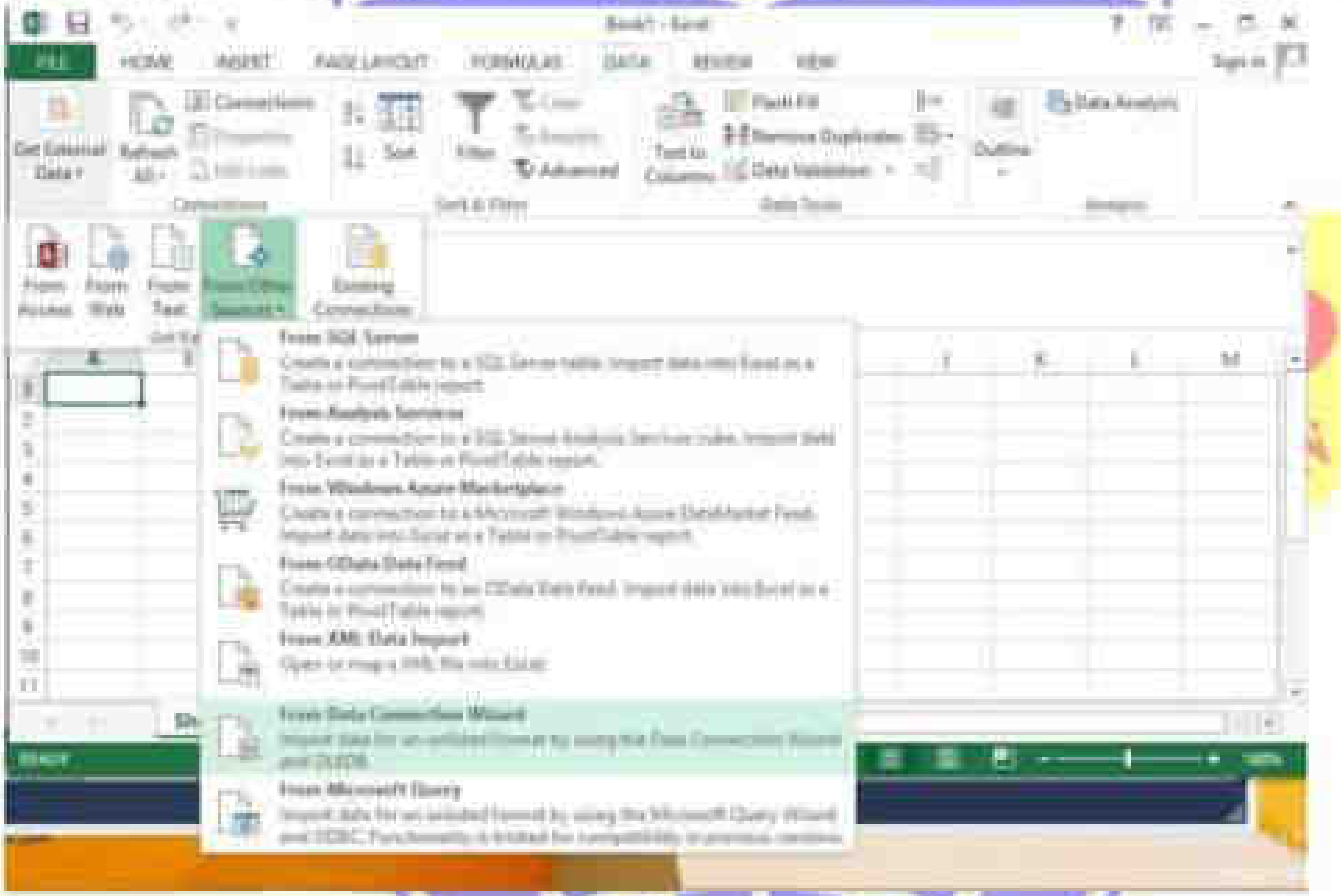
PRACTICAL 5 a

Import the datawarehouse data in Microsoft Excel and create the Pivot table and Pivot Chart

(Ms Office Professional is used to make sure Power View is enabled for visualization.)

Step 1: Open Excel 2013 (Professional)

Go to Data tab → Get External Data → From Other Sources → From Data Connection Wizard



From SQL Server
Create a connection to a SQL Server table. Import data into Excel as a Table or PivotTable report.

From Analysis Services
Create a connection to a SQL Server Analysis Services cube. Import data into Excel as a Table or PivotTable report.

From Windows Azure Marketplace
Create a connection to a Microsoft Windows Azure DataMarket Feed. Import data into Excel as a Table or PivotTable report.

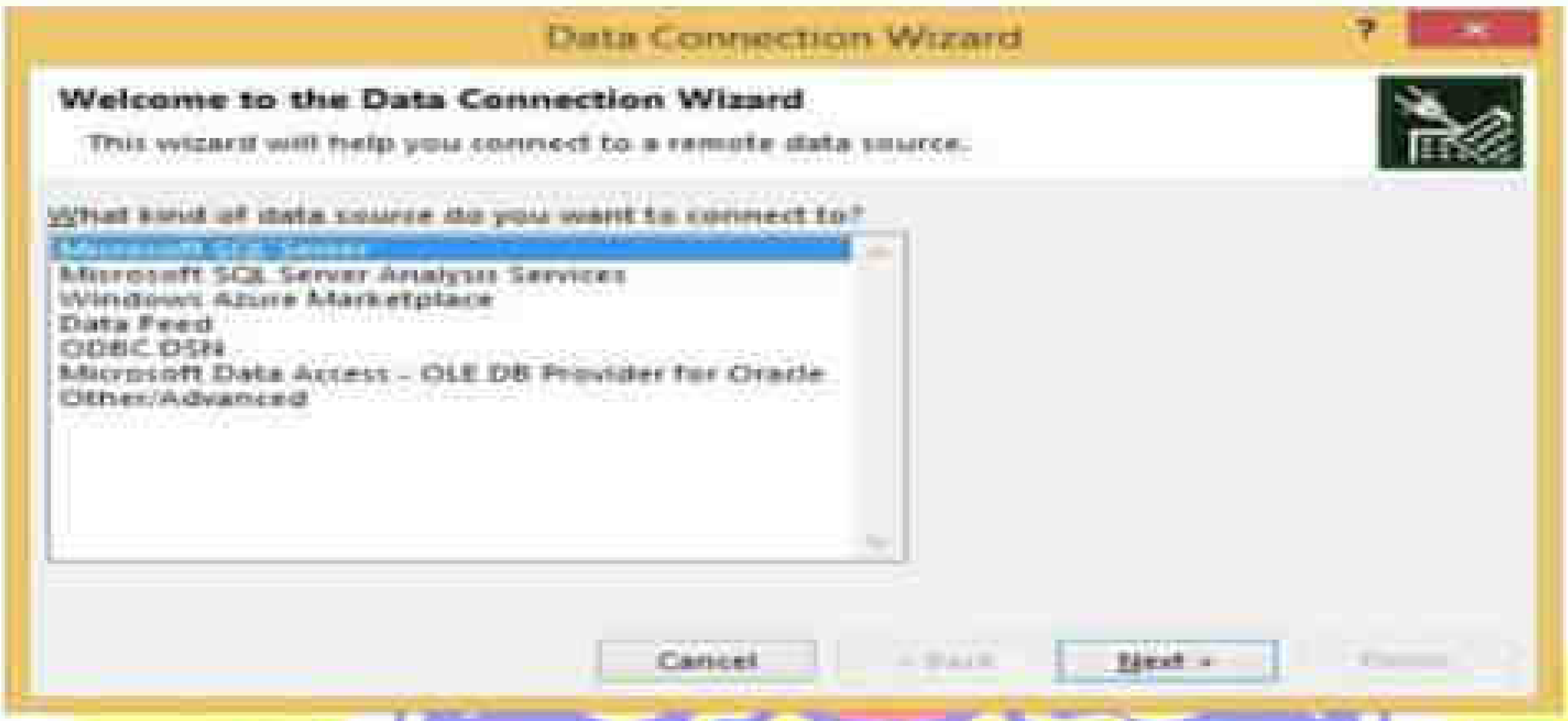
From OData Data Feed
Create a connection to an OData Data Feed. Import data into Excel as a Table or PivotTable report.

From XML Data Import
Open or map a XML file into Excel.

From Data Connection Wizard
Import data for an unlisted format by using the Data Connection Wizard and OLEDB.

From Microsoft Query
Import data for an unlisted format by using the Microsoft Query Wizard and ODBC. Functionality is limited for compatibility in previous versions.

Step 2: In Data Connection Wizard → Select Microsoft SQL Server → Click on Next



Welcome to the Data Connection Wizard

This wizard will help you connect to a remote data source.

What kind of data source do you want to connect to?

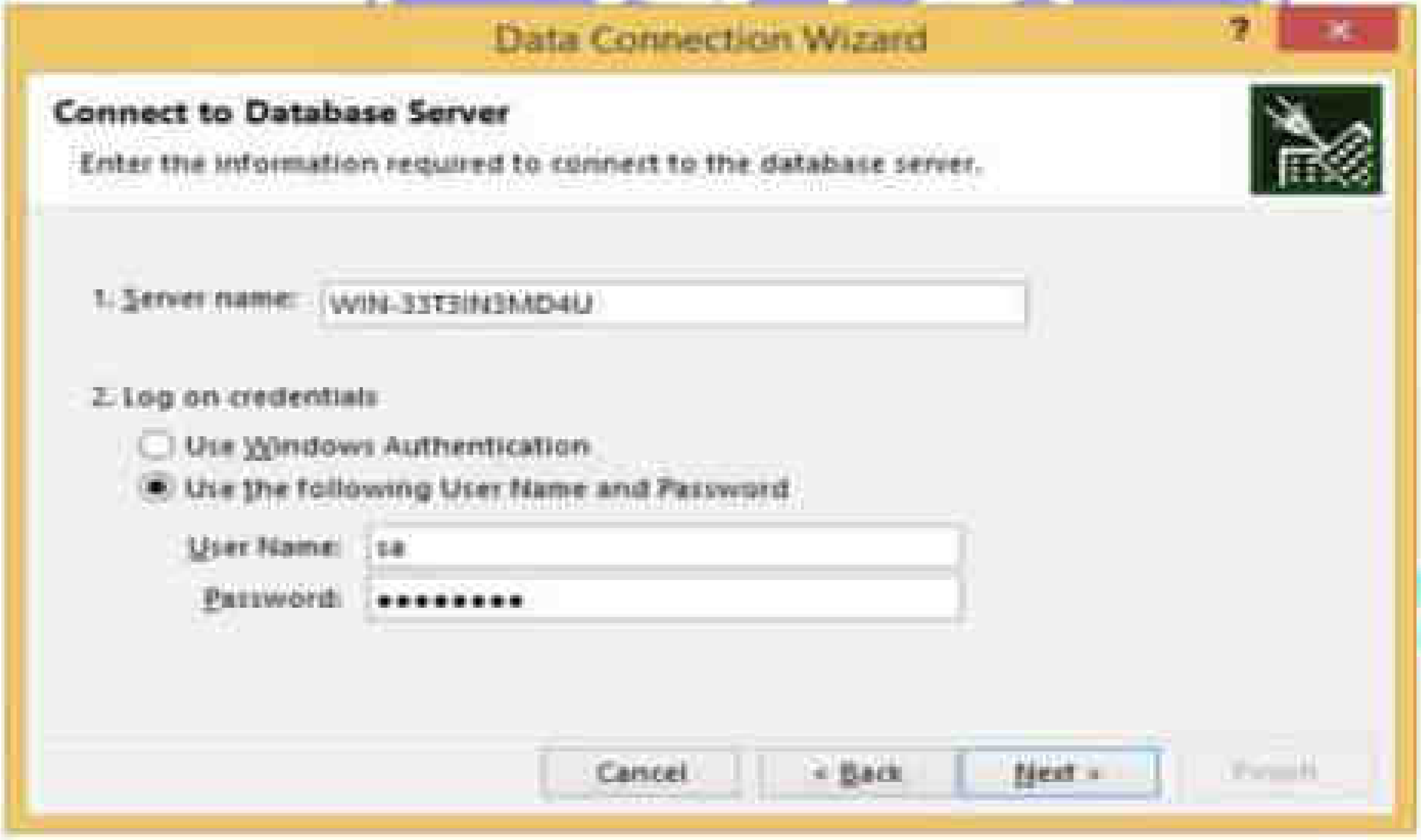
- Microsoft SQL Server
- Microsoft SQL Server Analysis Services
- Windows Azure Marketplace
- Data Feed
- ODBC DSN
- Microsoft Data Access – OLE DB Provider for Oracle
- Other/Advanced

Step 3: In connect to Database Server provide Server name(Microsoft SQL Server Name)

Provide password for sa account as given during installation of SQL Server 2012 full version)

Password: admin123

Click on Next



Connect to Database Server

Enter the information required to connect to the database server.

1. Server name: WIN-33T3IN3MD4U

2. Log on credentials

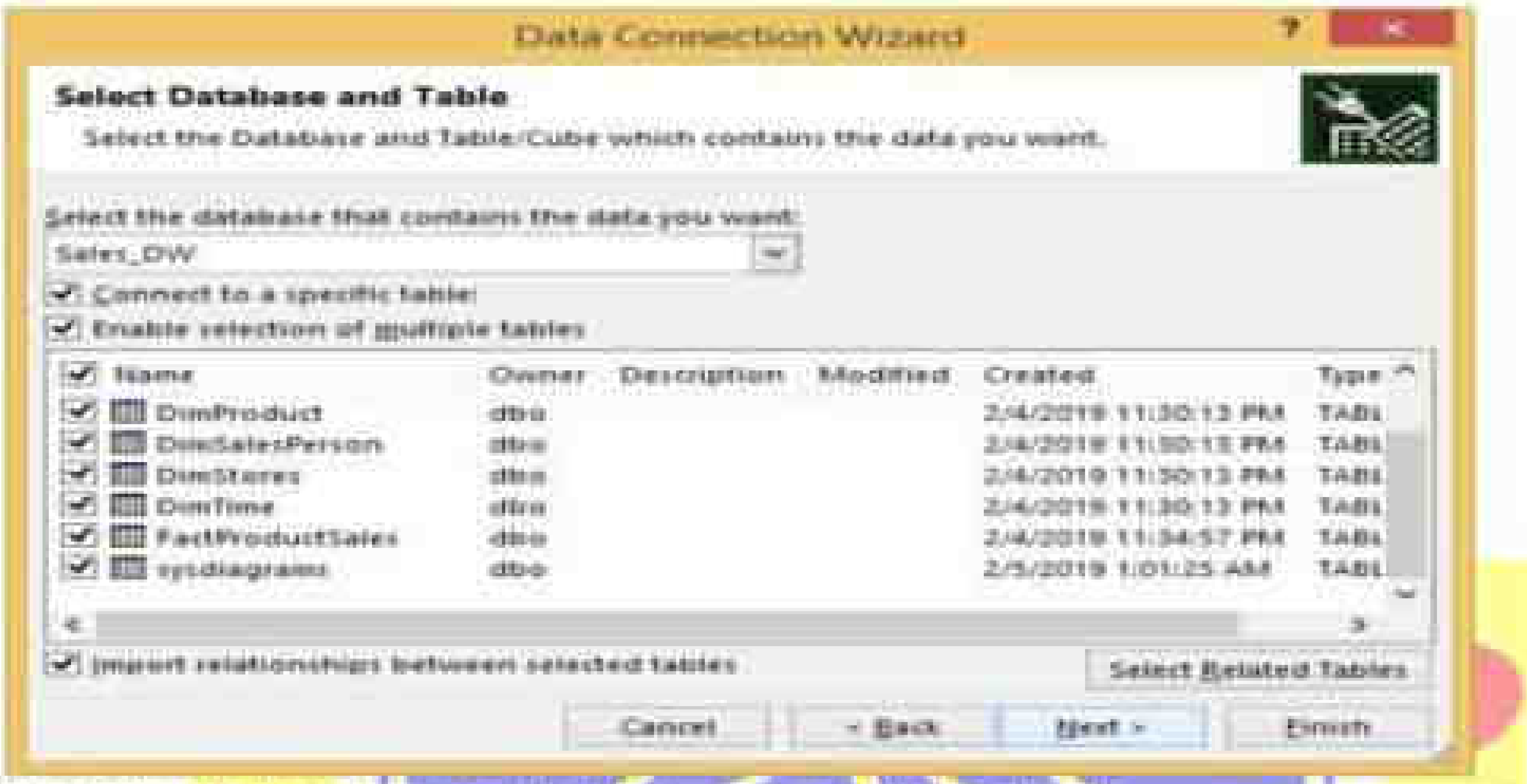
○ Use Windows Authentication

● Use the following User Name and Password

User Name: sa

Password: ●●●●●●●●●

Step 4: In Select Database and Table→ Select Sales_DW (already created in SQL) → check all dimensions and import relationships between selected tables



Data Connection Wizard

Select Database and Table

Select the Database and Table/Cube which contains the data you want.

Select the database that contains the data you want:

Sales_DW

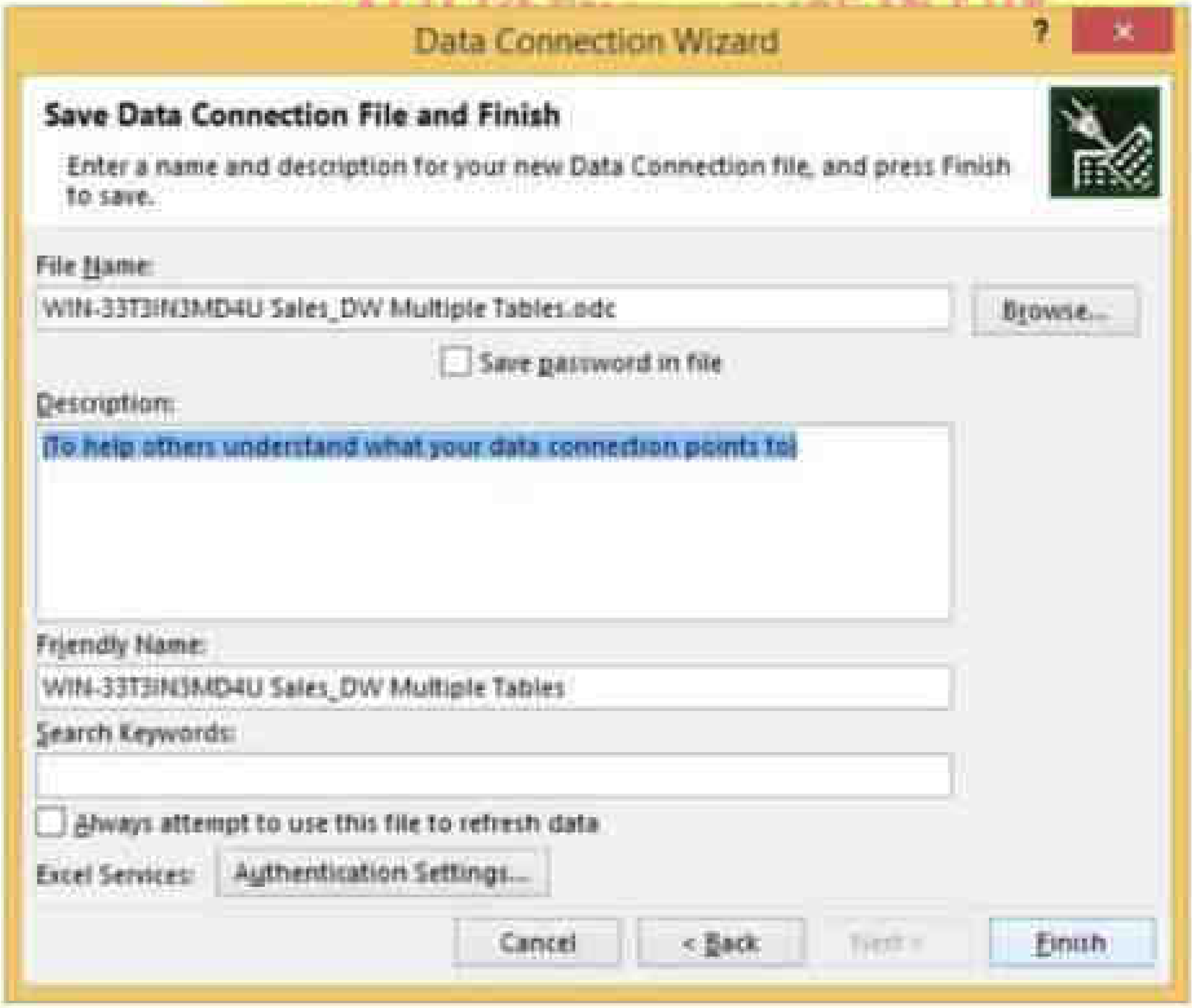
☑ Connect to a specific table:

☑ Enable selection of multiple tables

☑ Name	Owner	Description	Modified	Created	Type
☑ DimProduct	dbo			2/4/2019 11:30:13 PM	TABL
☑ DimSalesPerson	dbo			2/4/2019 11:30:13 PM	TABL
☑ DimStores	dbo			2/4/2019 11:30:13 PM	TABL
☑ DimTime	dbo			2/4/2019 11:30:13 PM	TABL
☑ FactProductSales	dbo			2/4/2019 11:34:57 PM	TABL
☑ sysdiagrams	dbo			2/5/2019 1:01:25 AM	TABL

☑ Import relationships between selected tables

Step 5: In save data connection files browse path and click on Finish



Data Connection Wizard

Save Data Connection File and Finish

Enter a name and description for your new Data Connection file, and press Finish to save.

File Name:

WIN-33T3IN3MD4U Sales_DW Multiple Tables.odc

□ Save password in file

Description:

(To help others understand what your data connection points to)

Friendly Name:

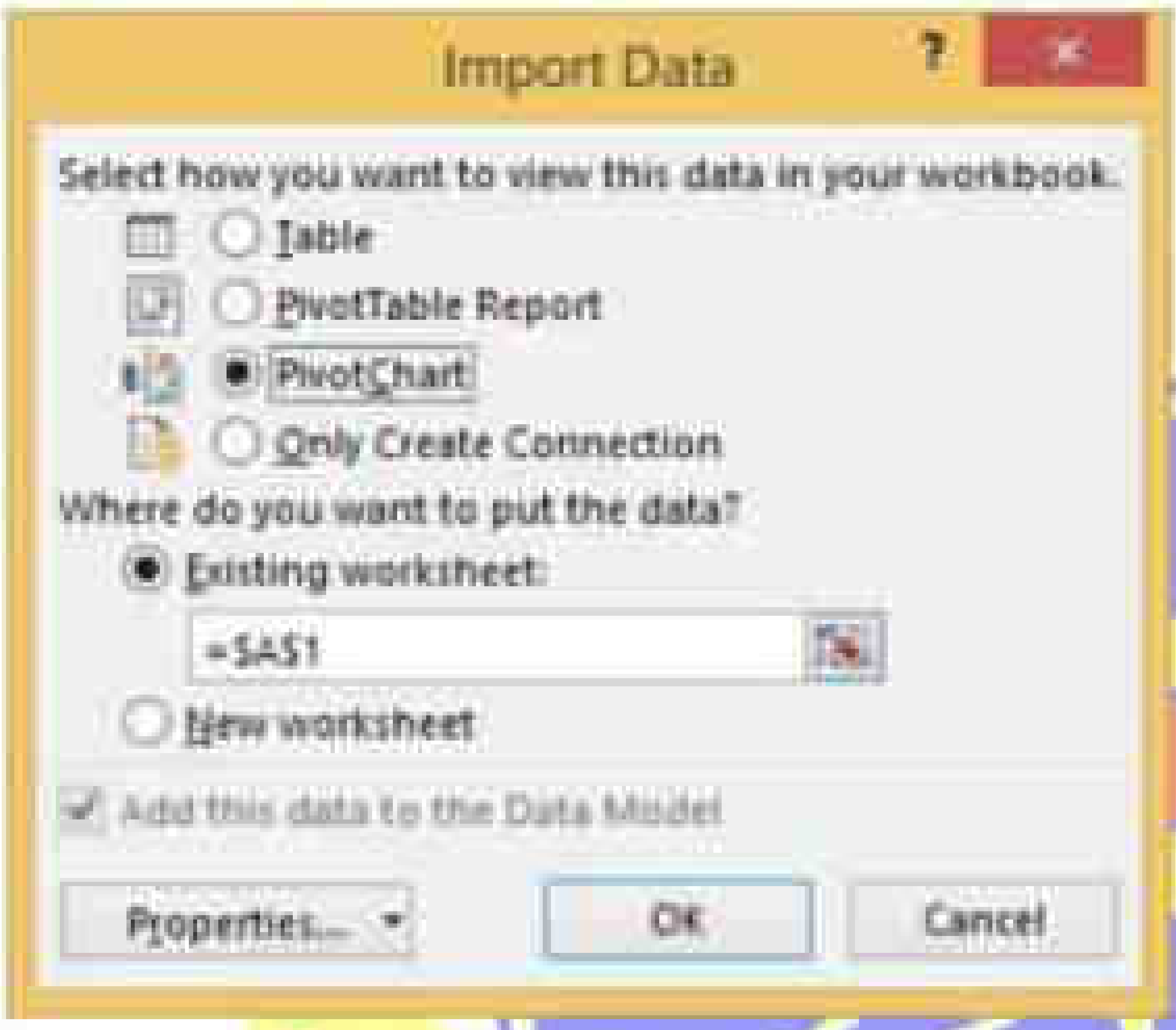
WIN-33T3IN3MD4U Sales_DW Multiple Tables

Search Keywords:

□ Always attempt to use this file to refresh data

Excel Services:

Step 6: In import data select Pivot Chart and click on OK



Import Data

Select how you want to view this data in your workbook.

- ○ Table
- ○ PivotTable Report
- ● PivotChart
- ○ Only Create Connection

Where do you want to put the data?

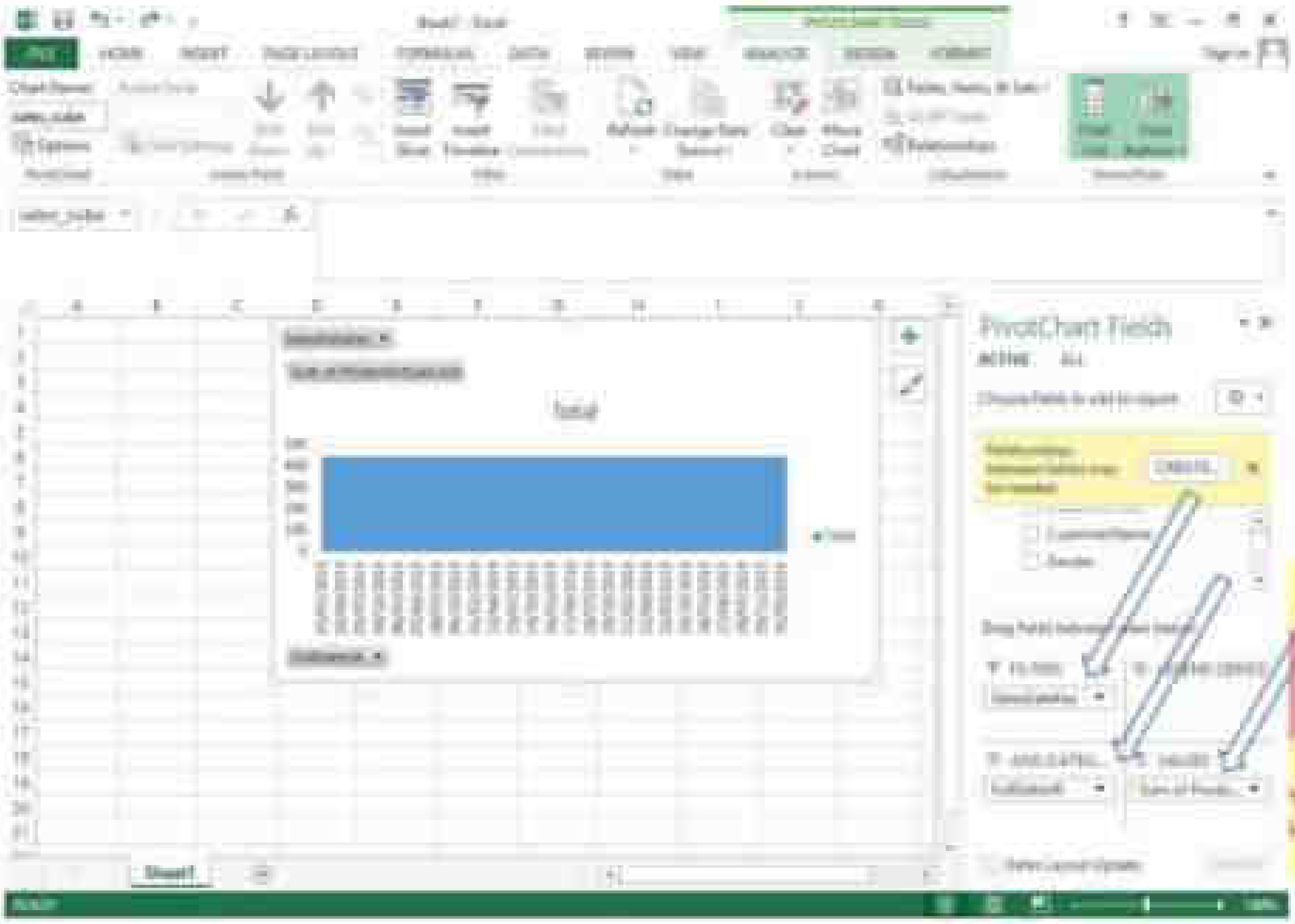
- ● Existing worksheet:

  =$A$1

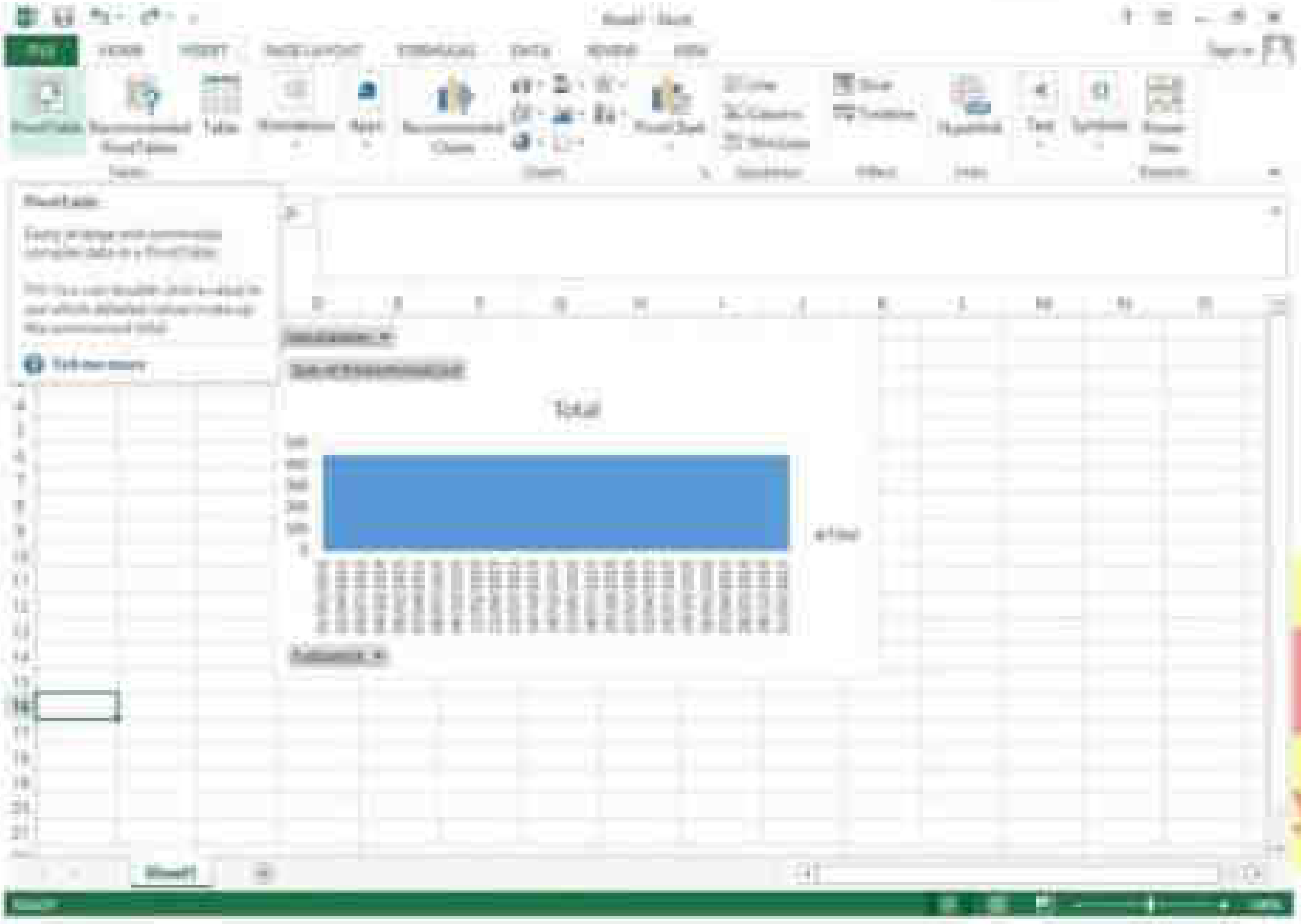
- ○ New worksheet

☑ Add this data to the Data Model

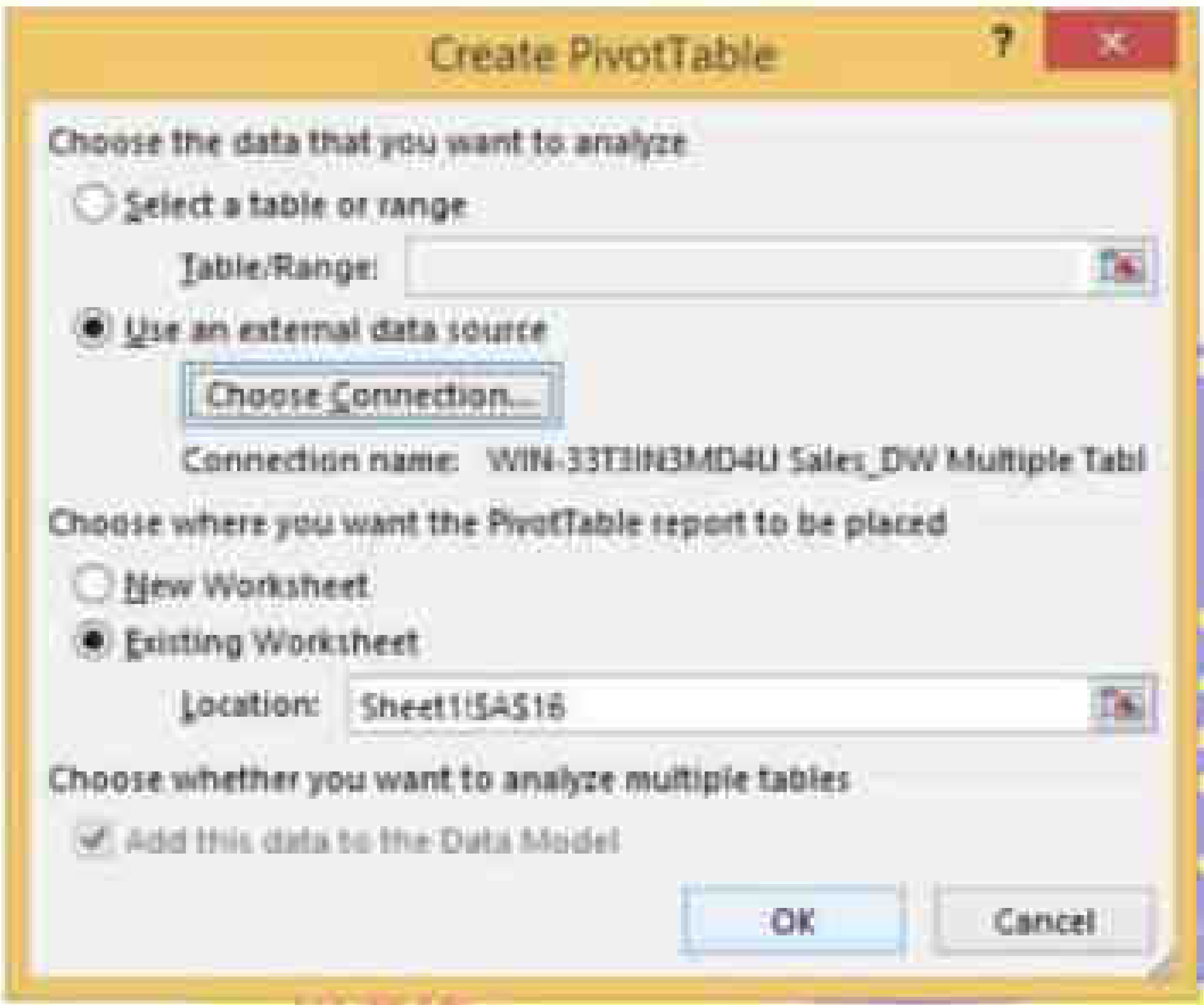
Step 7: In fields put SalesDateKey in filters, FullDateUK in axis and Sum of ProductActualCost in values



Step 8: In Insert Tab → go to Pivot Table



Step 9: Click on Choose Connection to select existing connection with Sales_DW and click on open



Create PivotTable

Choose the data that you want to analyze

○ Select a table or range

Table/Range:

● Use an external data source

Choose Connection...

Connection name: WIN-33T3IN3MD4U Sales_DW Multiple Tabl

Choose where you want the PivotTable report to be placed

○ New Worksheet

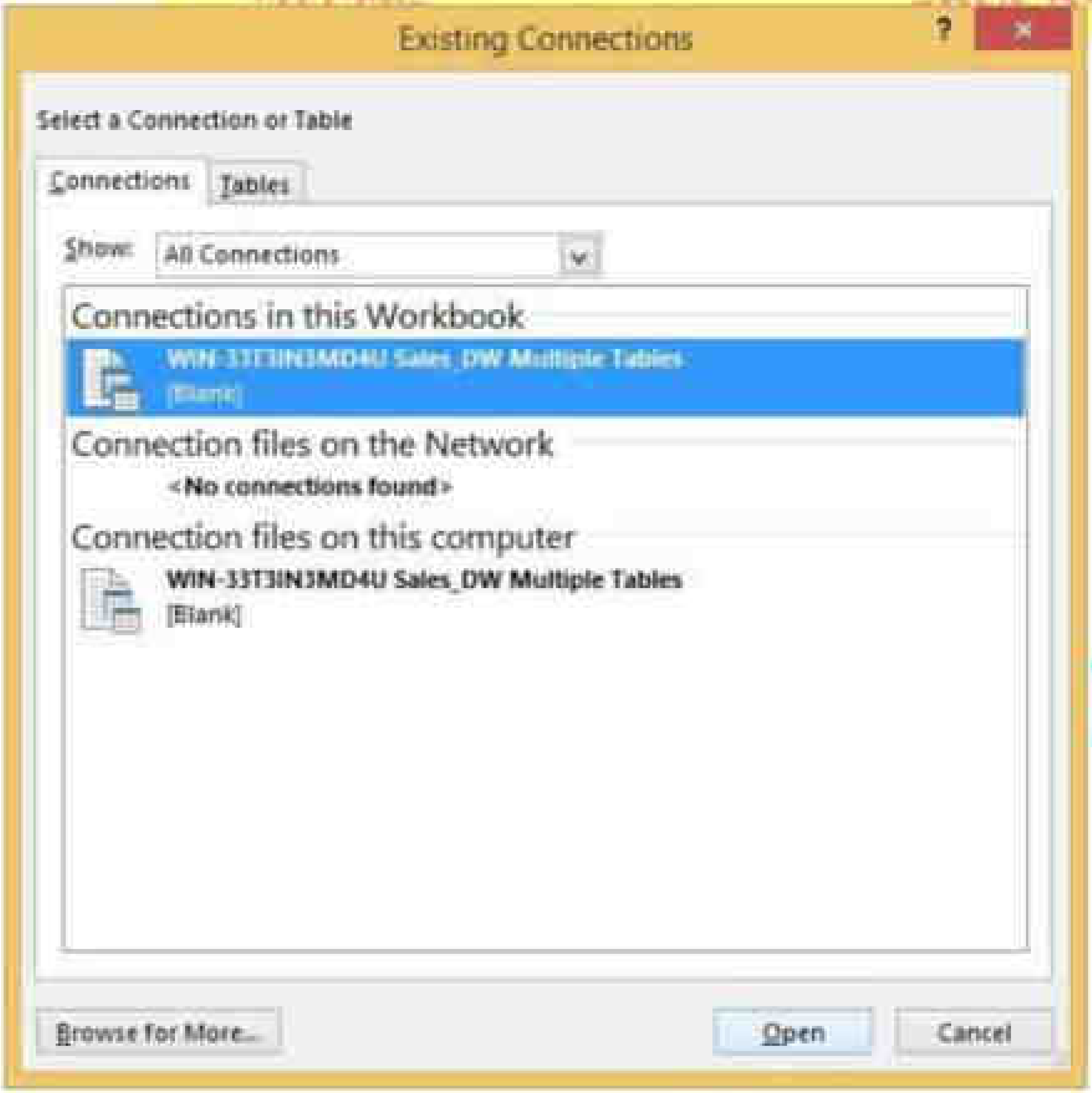
● Existing Worksheet

Location: Sheet1!$A$16

Choose whether you want to analyze multiple tables

☑ Add this data to the Data Model

OK Cancel



Existing Connections

Select a Connection or Table

Connections Tables

Show: All Connections

Connections in this Workbook

WIN-33T3IN3MD4U Sales_DW Multiple Tables
[Blank]

Connection files on the Network

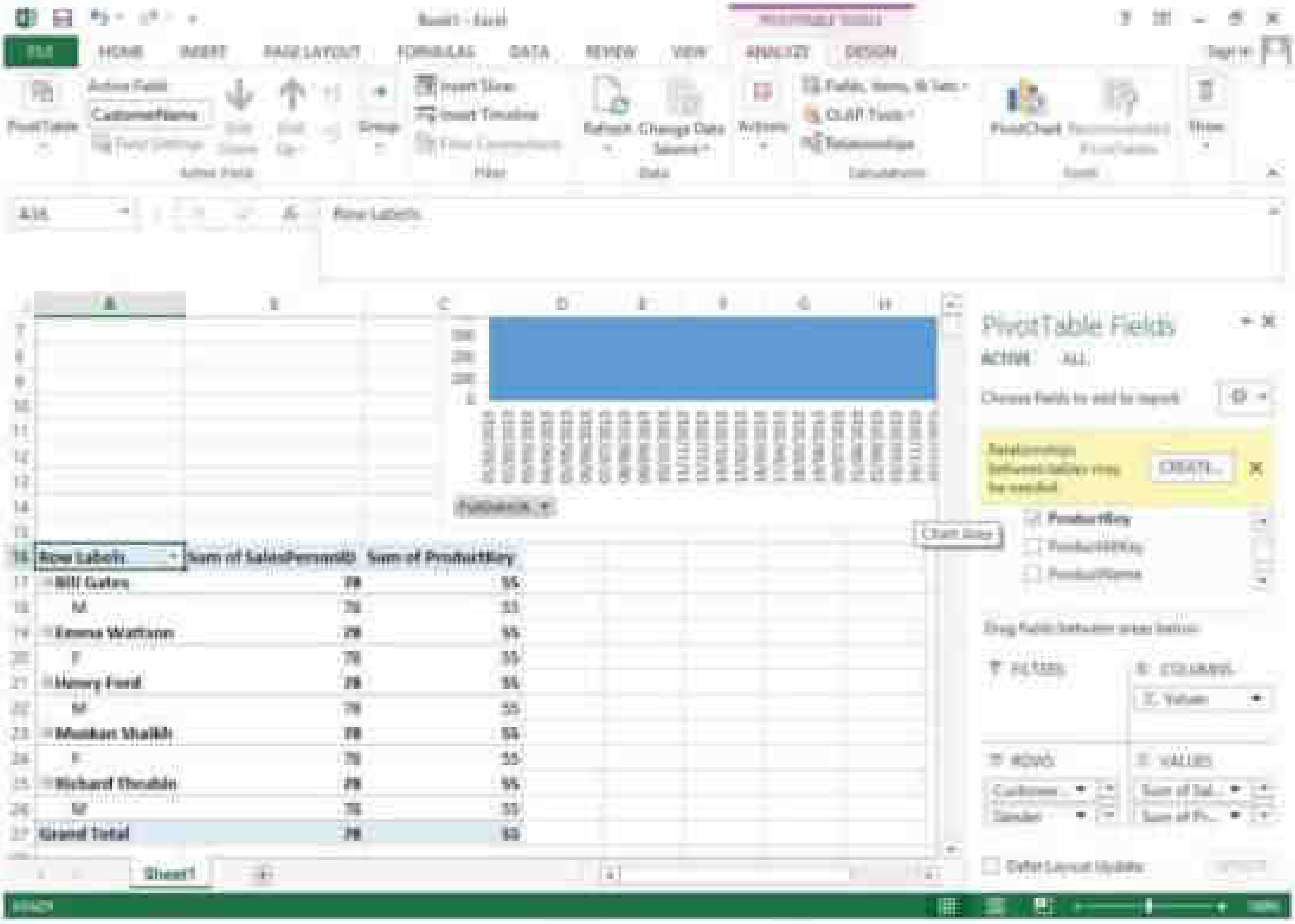
<No connections found>

Connection files on this computer

WIN-33T3IN3MD4U Sales_DW Multiple Tables
[Blank]

Browse for More... Open Cancel

Pivot table and Pivot chart is created



Row Labels

Row Labels	Sum of SalesPersonID	Sum of ProductKey
Bill Gates	78	55
M	78	55
Emma Watson	78	55
F	78	55
Henry Ford	78	55
M	78	55
Muskan Shaikh	78	55
F	78	55
Richard Thrubin	78	55
M	78	55
Grand Total	78	55

Learning Outcomes:

Course Outcomes:

Conclusion:

Viva Questions:

1. **What is a data warehouse?**
2. **Name a key feature of a data warehouse.**
3. **Why are data warehouses used?**
4. **What is the difference between OLTP and OLAP?**

For Faculty use:

Correction Parameters	Formative Assessment[40%]	Timely Completion of Practical[40%]	Attendance Learning Attitude[20%]

SHRI GPM DEGREE COLLEGE OF SCIENCE & COMMERCE

Department of Computer

Affiliated to University of Mumbai

Vision.. Innovation.. Solution.. Presentation..

Practical-6: Apply the what – if Analysis for data visualization. Design and generate necessary reports based on the data ware house data.

Aims:

1. To apply what-if analysis techniques to simulate various business scenarios using data warehouse data.
2. To design and generate dynamic reports that help in understanding potential outcomes and supporting decision-making.

Learning Objectives:

1. Understand the concept and benefits of what-if analysis in a data warehousing context.
2. Learn to configure and manipulate what-if parameters to simulate changes in key business metrics.
3. Gain hands-on experience in designing interactive dashboards and reports that reflect hypothetical scenarios.
4. Develop skills in interpreting data-driven insights to inform business strategies.

Tool & Technologies used:

1. Power BI is a powerful business intelligence tool used for data visualization and analysis.

Theory-6: Data ware house data

Data warehouses consolidate information from diverse sources, creating a centralized repository for strategic decision-making. Applying what-if analysis enables analysts to simulate different scenarios, assess potential impacts, and visualize outcomes. This approach transforms raw data into actionable insights, guiding strategic planning and operational improvements.

What-If Analysis for Data Visualization

- **Definition:**
  What-if analysis involves modifying key input variables to explore alternative outcomes. It helps identify risks, forecast trends, and understand the sensitivity of various business metrics.
- **Techniques and Tools:**
  Utilize tools like Microsoft Excel, Power BI, or Tableau to perform scenario analysis. These tools support dynamic simulations using data models, adjustable parameters, and interactive dashboards.
- **Application Areas:**

- **Financial Forecasting:** Model revenue or expense changes under different economic conditions.
- **Sales Projections:** Analyze the impact of pricing strategies or market fluctuations on sales volumes.
- **Resource Allocation:** Predict outcomes of different staffing or inventory levels.

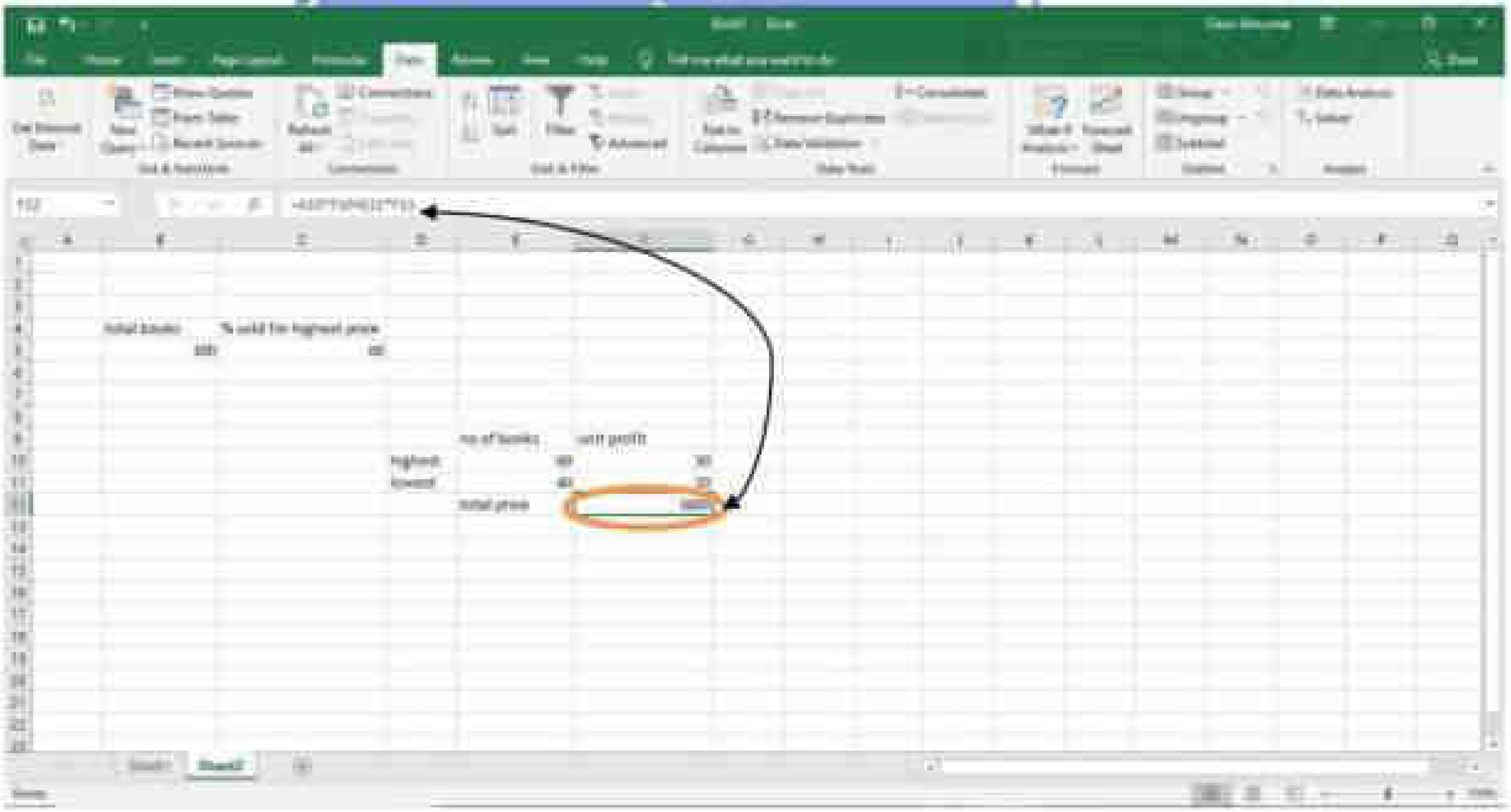
Designing and Generating Reports

1. **Data Preparation and Integration:**
   - Import data warehouse records into Excel or a BI tool using secure connections (e.g., ODBC, SQL Server connectors).
   - Cleanse and aggregate the data to ensure accuracy for analysis.
2. **Report Layout and Structure:**
   - **Interactive Dashboards:** Create dashboards that integrate PivotTables, PivotCharts, and slicers to allow dynamic filtering.
   - **Custom Metrics:** Incorporate key performance indicators (KPIs) tailored to the business context, such as profit margins, growth rates, or cost variances.

PRACTICAL 6

Apply the what – if Analysis for data visualization. Design and generate necessary reports based on the data warehouse data.

A book store and have 100 books in storage. You sell a certain % for the highest price of $50 and a certain % for the lower price of $20.



	B	C	D	E	F
4	total books	% sold for highest price			
5	100	60			
9				no of books	unit profit
10			highest	60	50
11			lowest	40	20
12				total price	[illegible]

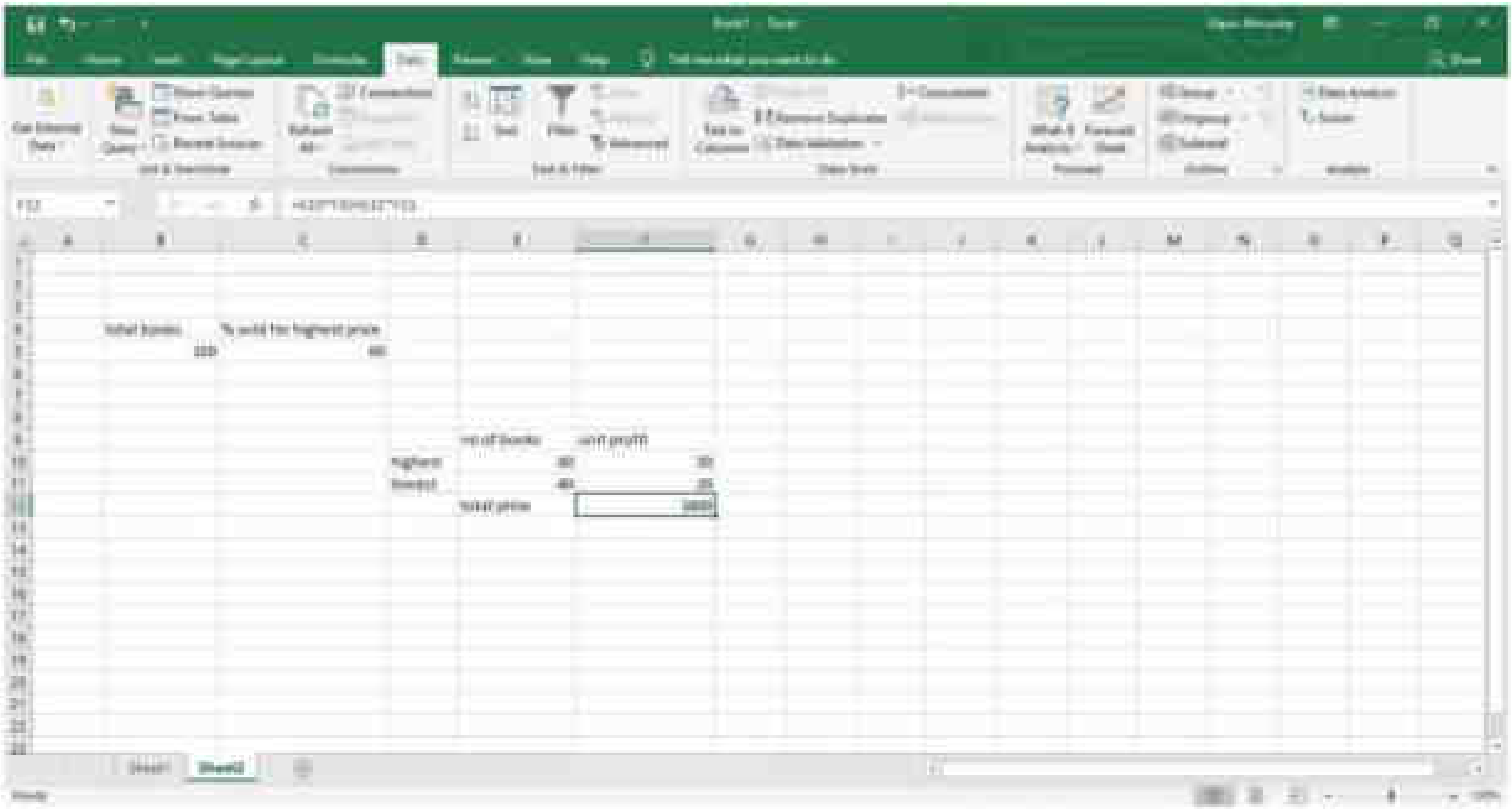
If you sell 60% for the highest price, cell D10 calculates a total profit of 60 * 50 + 40 * 20 = 3800.

Create Different Scenarios But what if you sell 70% for the highest price? And what if you sell 80% for the highest price? Or 90%, or even 100%? Each different percentage is a different scenario. You can use the Scenario Manager to create these scenarios.

Note: To type different percentage into cell C4 to see the corresponding result of a scenario in cell D10 we use what if analysis.

What-if analysis enables you to easily compare the results of different scenarios.

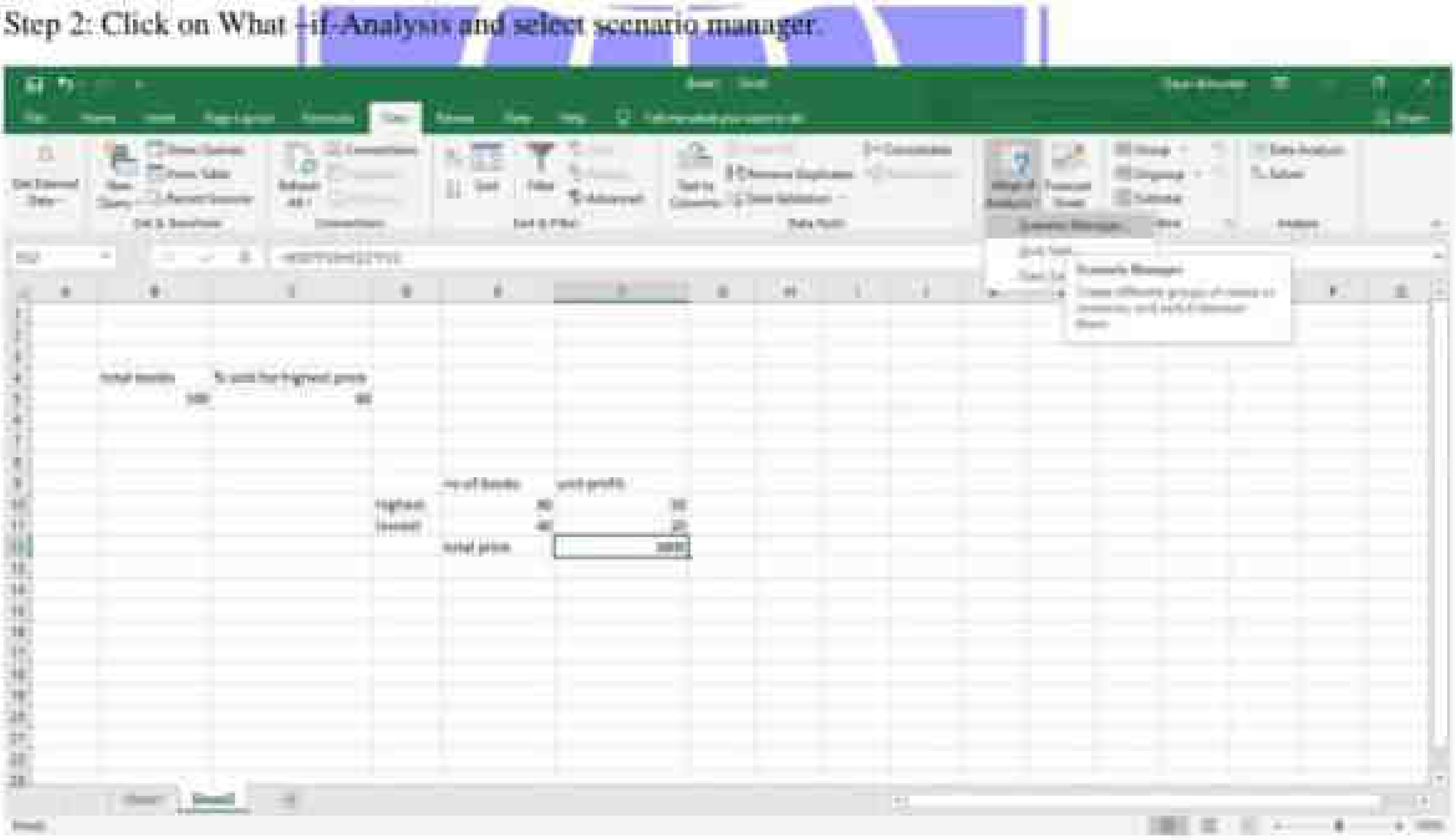
Step 1: In Excel, On the Data tab, in the Data tools group, click What-If Analysis



F12 =E10*F10+E11*F11

	B	C	D	E	F
4	total books	% sold for highest price			
5	100	60			
9				no of books	unit profit
10			highest	60	50
11			lowest	40	20
12				total price	3800

Step 2: Click on What –if-Analysis and select scenario manager.

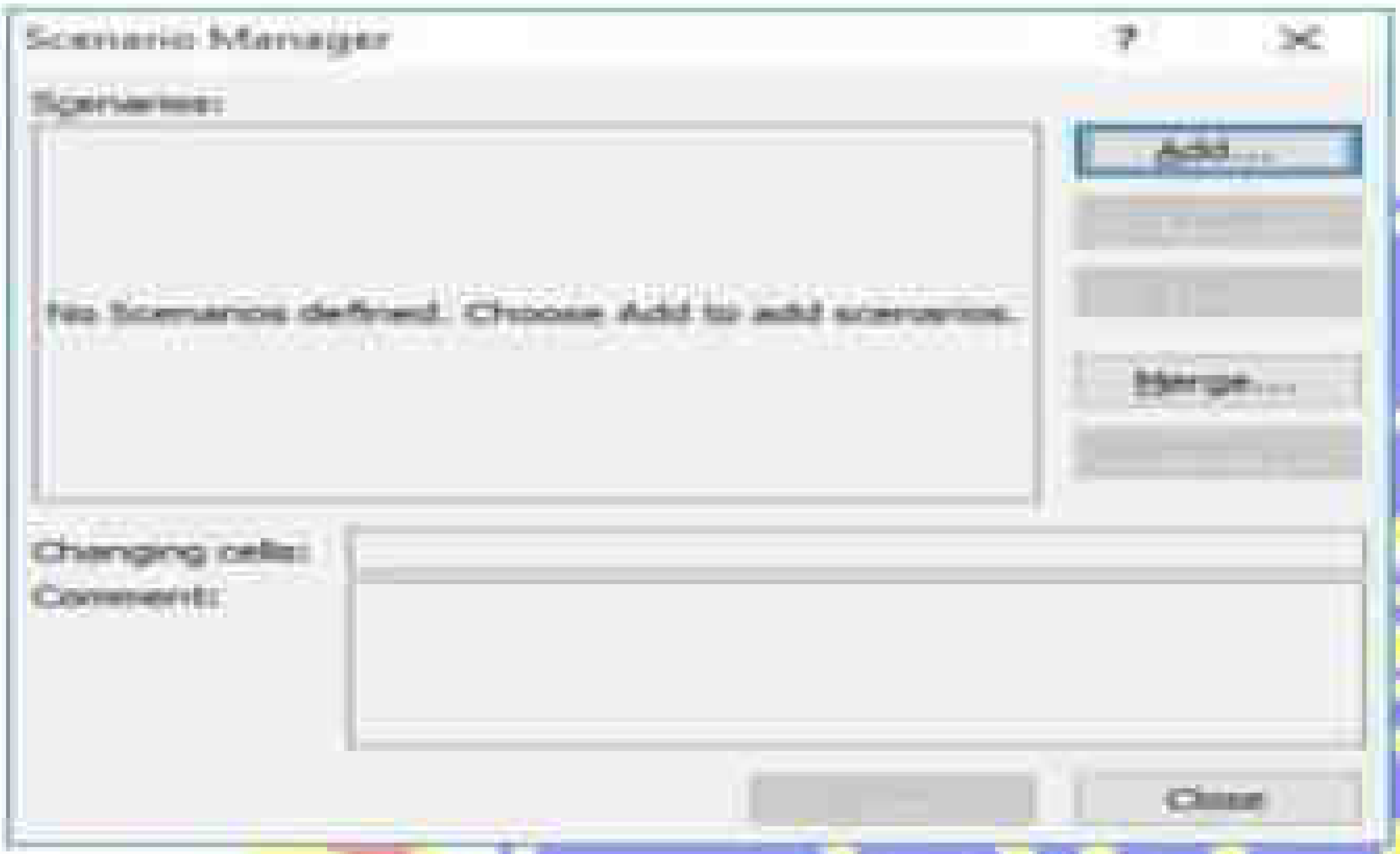


Scenario Manager
Create different groups of values, or scenarios, and switch between them.

F12 =E10*F10+E11*F11

	B	C	D	E	F
4	total books	% sold for highest price			
5	100	60			
9				no of books	unit profit
10			highest	60	50
11			lowest	40	20
12				total price	3800

The Scenario Manager Dialog box appears. Step 3: Add a scenario by clicking on Add.

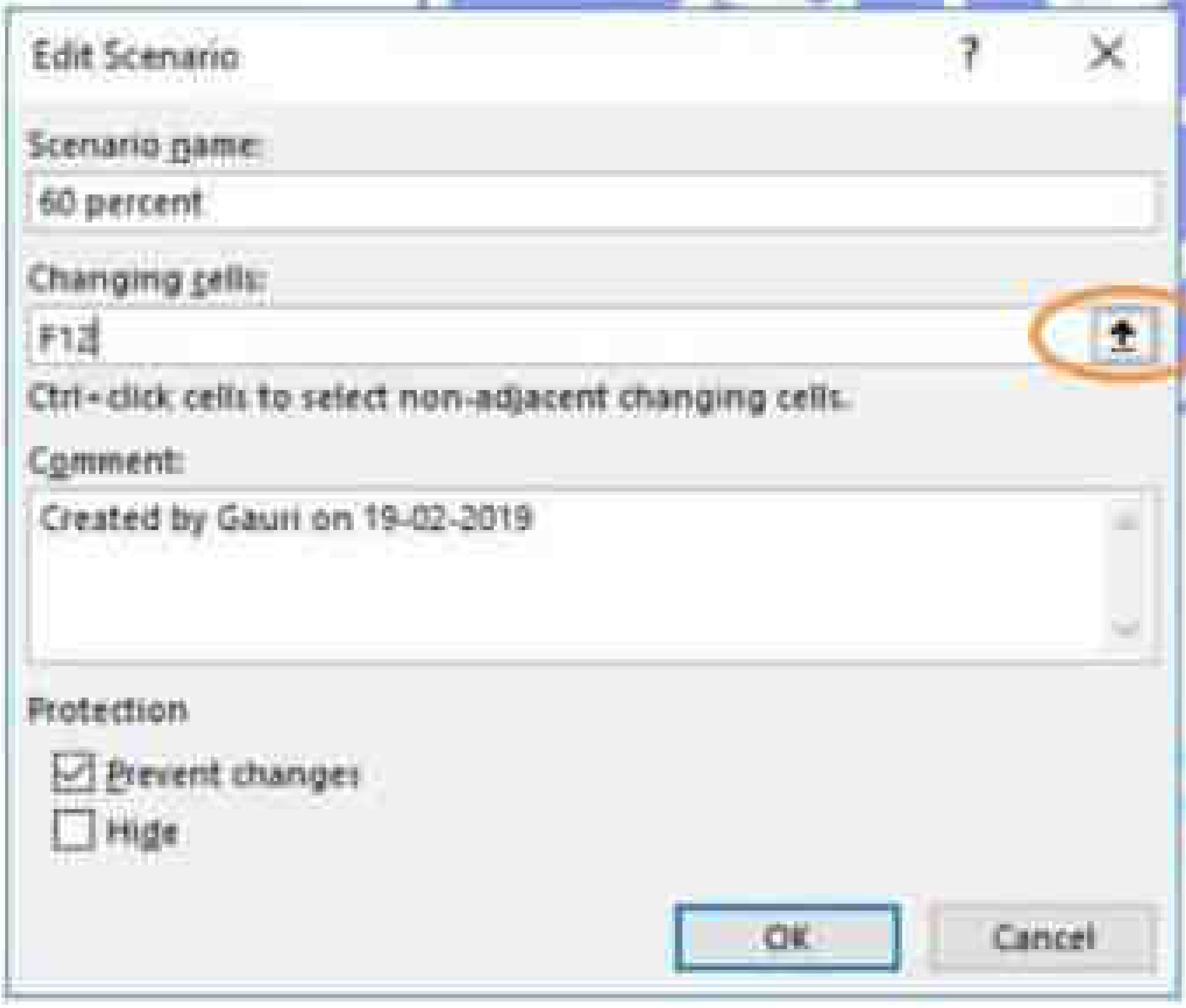


No Scenarios defined. Choose Add to add scenarios.

Changing cells:
Comment:

Step 4: Type a name (60percent), select cell F10 (% sold for the highest price) for the Changing cells and click on OK.

Click on icon which is circled.



Scenario name:
60 percent

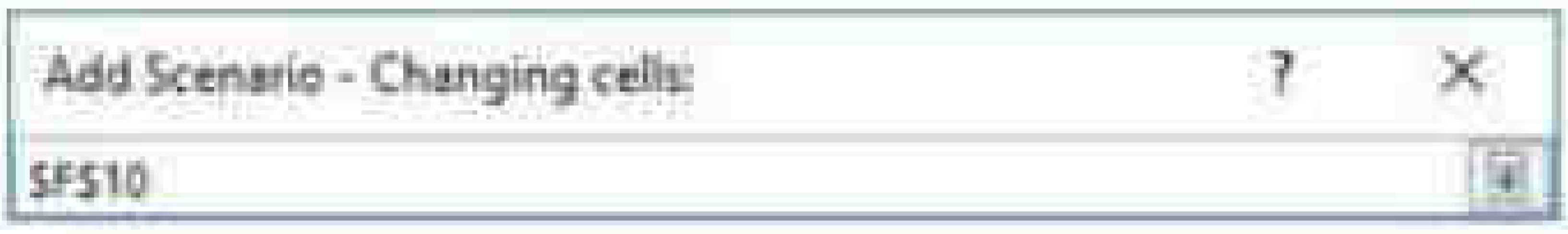
Changing cells:
F12

Ctrl+click cells to select non-adjacent changing cells.

Comment:
Created by Gauri on 19-02-2019

Protection
☑ Prevent changes
□ Hide

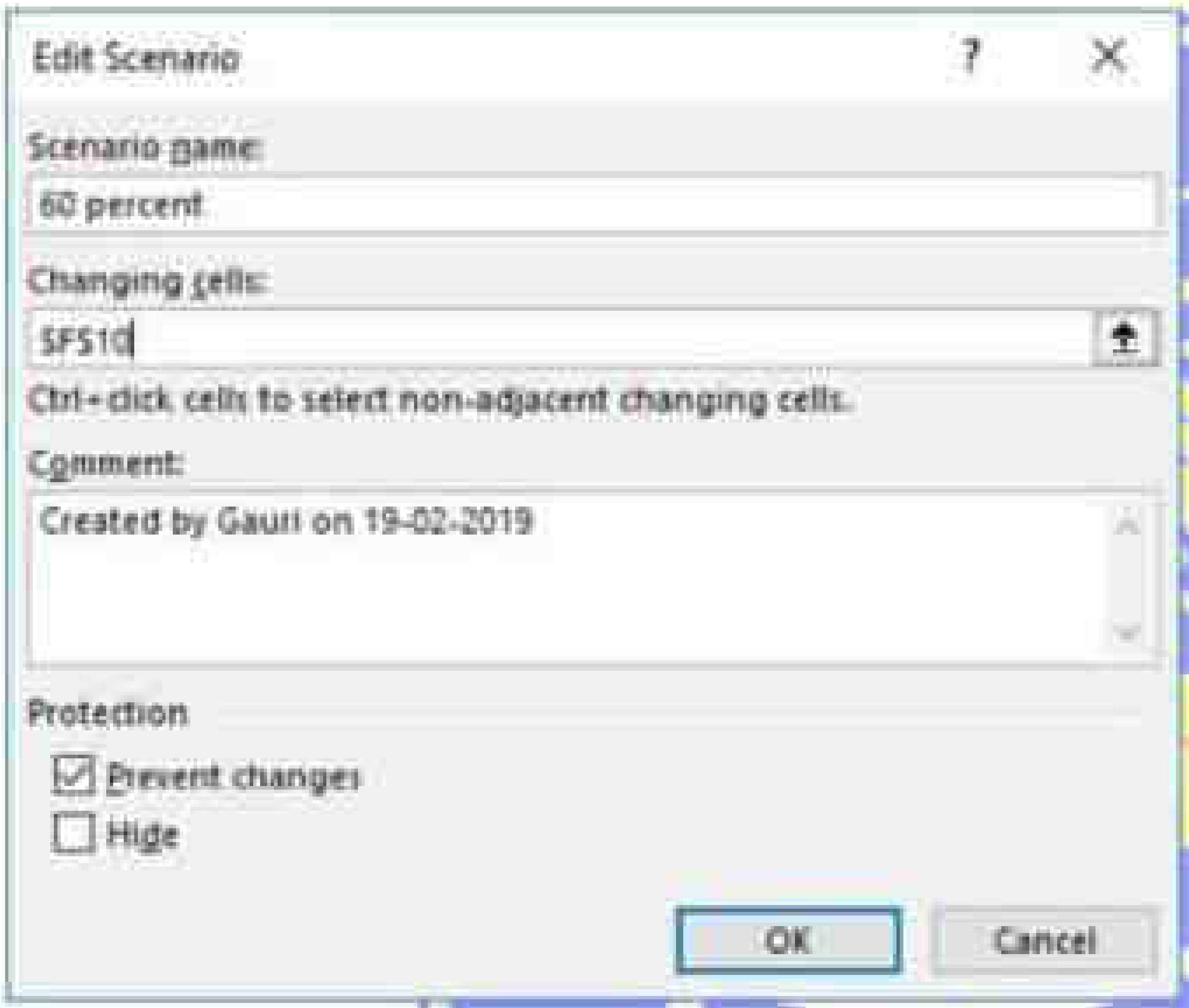
Select F10 cell.



Add Scenario - Changing cells:

$F$10

Click back on the icon again and then click OK



Edit Scenario

Scenario name: 60 percent

Changing cells: $F$10

Ctrl+click cells to select non-adjacent changing cells.

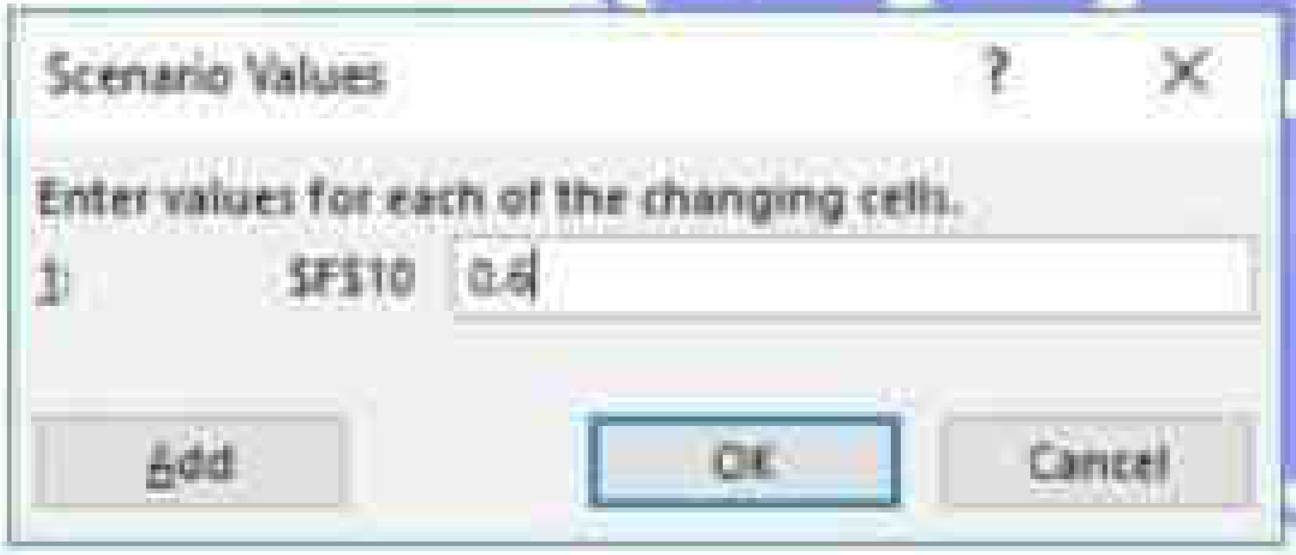
Comment: Created by Gauri on 19-02-2019

Protection

☑ Prevent changes

□ Hide

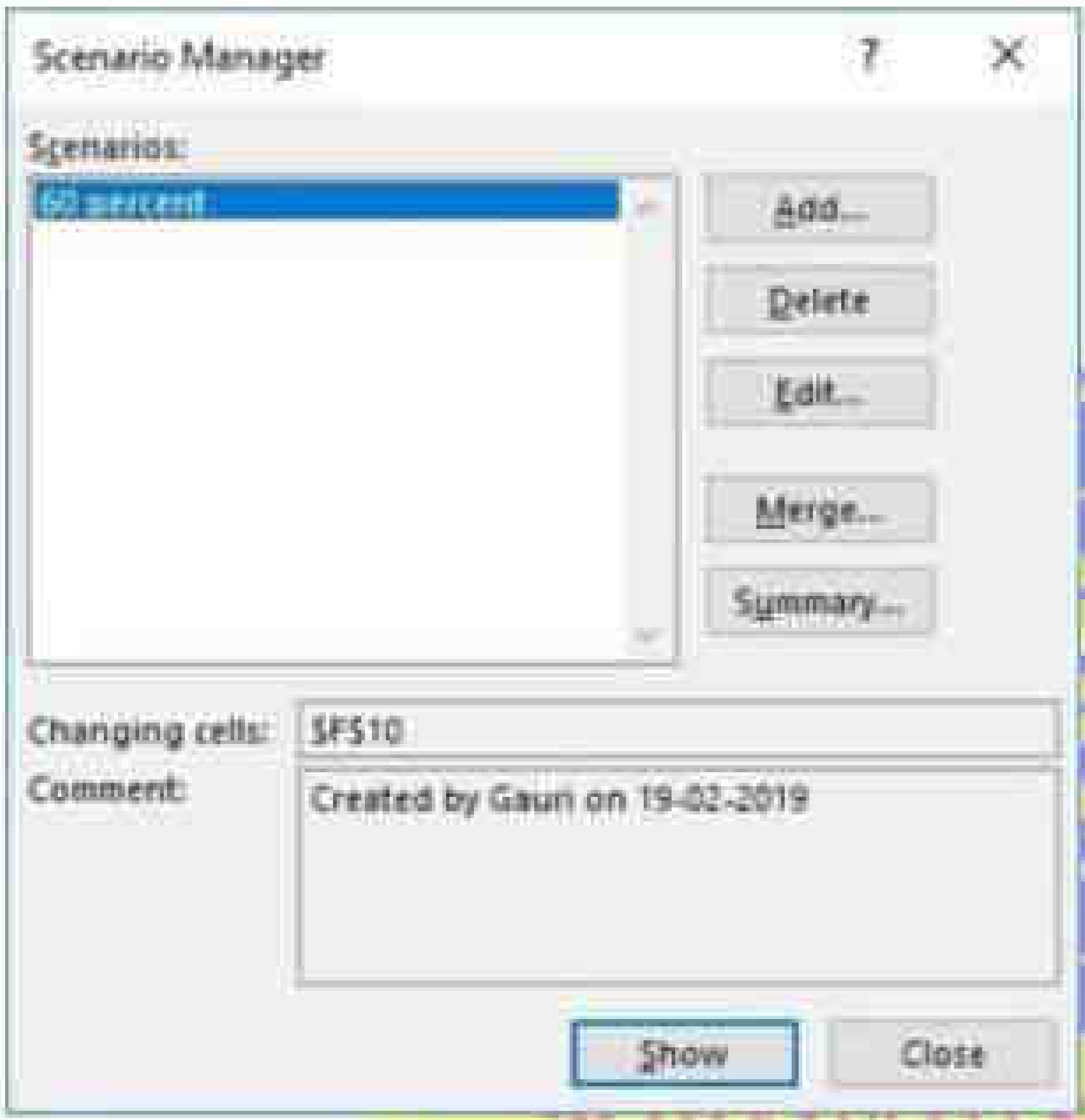
Step 5: Enter the corresponding value 0.6 and click on OK again.



Scenario Values

Enter values for each of the changing cells.

1: $F$10 0.6

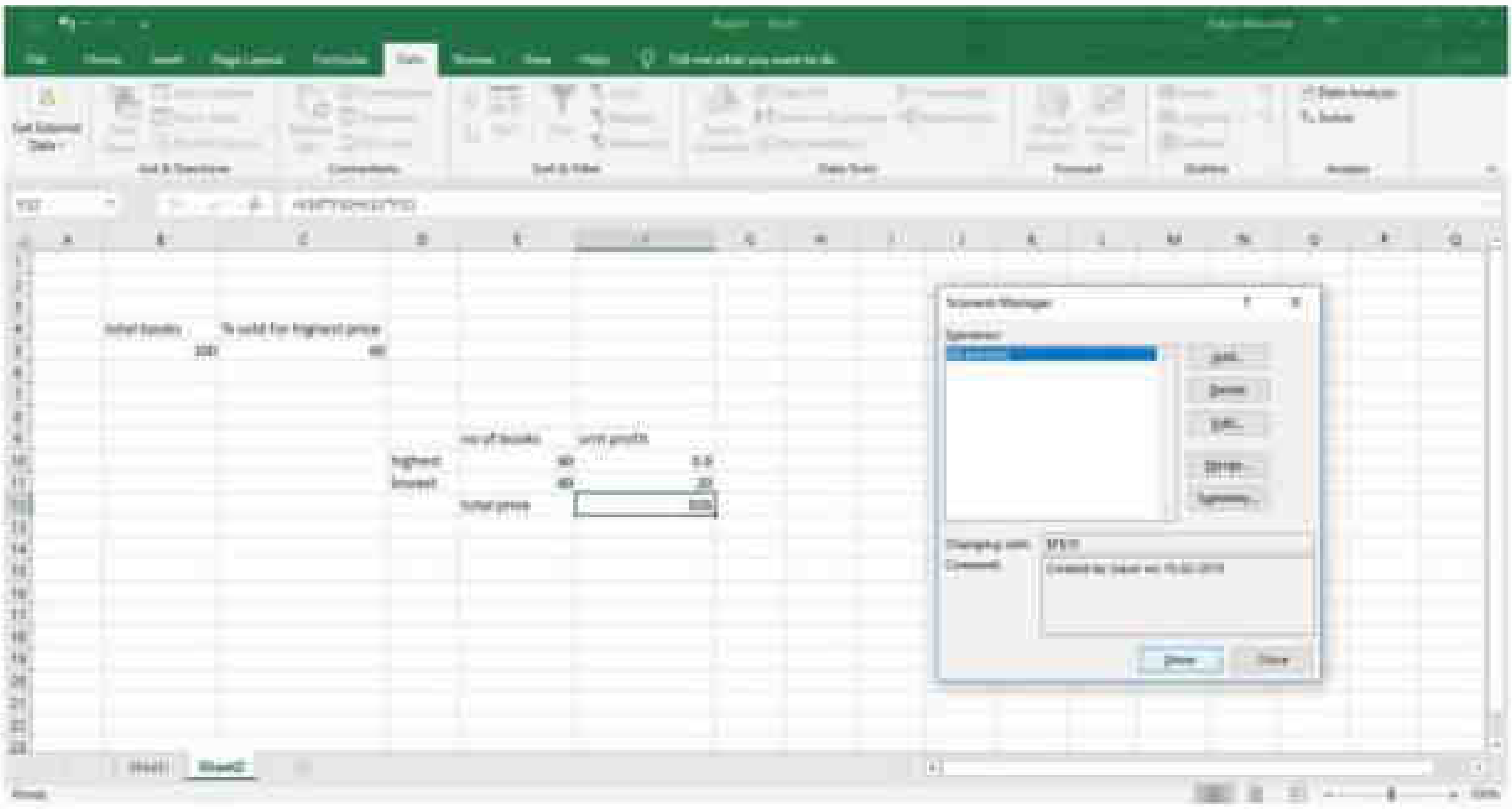


Scenarios: 60 percent

Changing cells: $F$10

Comment: Created by Gauri on 19-02-2019

Step 6: To apply scenarios click on Show.



total books | % sold for highest price
100 | 60

	no of books	unit profit
highest	60	3.5
lowest	40	20
	total price	600

Scenario Manager

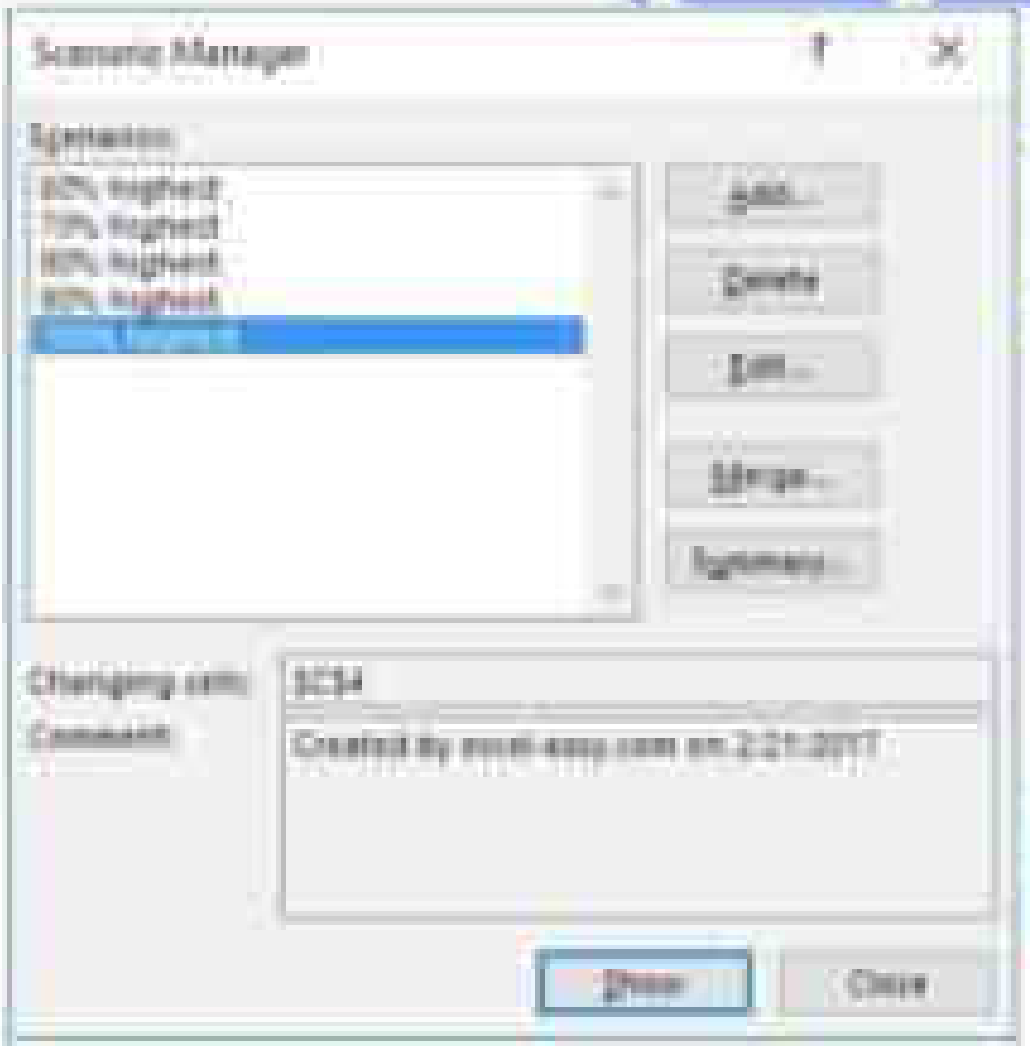
Scenarios:

Changing cells: $F$11

Comment: Created by [illegible] on [illegible]

Step 7: Next, add 4 other scenarios (70%, 80%, 90% and 100%)

Finally, your Scenario Manager should be consistent with the picture below:



Scenario Manager

Scenarios:

60% highest
70% highest
80% highest
90% highest
100% highest

Changing cells: $C$4

Comment: Created by excel-easy.com on 2-21-2017

Learning Outcomes:

Course Outcomes:

Conclusion:

Viva Questions:

1. **What is data warehousing?**
2. **Define data warehouse data.**
3. **Why is data warehousing important?**
4. **Data warehouse vs. database?**

For Faculty use:

Correction Parameters	Formative Assessment[40%]	Timely Completion of Practical[40%]	Attendance Learning Attitude[20%]

Practical-7: Perform the data classification using classification algorithm

Aims:

1. To apply classification algorithms for categorizing data into predefined classes.
2. To understand how data classification supports predictive analytics and decision-making.

Learning Objectives:

1. Comprehend the fundamental concepts and techniques in classification within data mining and machine learning.
2. Gain hands-on experience in implementing classification algorithms using programming libraries.
3. Evaluate model performance through metrics such as accuracy, precision, recall, and F1 score.
4. Develop skills in data preprocessing, model training, and interpretation of classification results.

Tool & Technologies used:

1. Power BI is a powerful business intelligence tool used for data visualization and analysis.

Theory-7: Classification

Data classification is a supervised machine learning technique aimed at categorizing data into predefined classes or labels. By analyzing historical, labeled data, classification algorithms learn patterns and relationships, enabling them to accurately predict the class of new, unseen instances.

Process of Data Classification

1. **Data Preparation**
   - **Collection & Cleaning:** Gather a representative dataset and clean it by handling missing values, removing outliers, and correcting errors.
   - **Feature Selection & Engineering:** Identify key features that influence the classification outcome and transform raw data into a suitable format for analysis.
2. **Dataset Partitioning**
   - **Training Set:** Allocate a portion of the data to train the classification model, ensuring it learns the underlying patterns.
   - **Test Set:** Reserve another portion to validate and evaluate the model's performance, thereby preventing overfitting.
3. **Algorithm Selection**

- Choose an appropriate classification algorithm based on the nature of the data and problem requirements. Common choices include:
  - **Decision Trees:** Offer intuitive, rule-based classification.
  - **Naive Bayes:** Uses probabilistic reasoning for efficient classification.
  - **Support Vector Machines (SVM):** Effective in high-dimensional feature spaces.
  - **K-Nearest Neighbors (KNN):** A simple, instance-based approach.

4. **Model Training and Evaluation**
   - **Training:** Input the training data into the selected algorithm to develop a predictive model.
   - **Evaluation:** Use performance metrics—such as accuracy, precision, recall, and F1 score—to assess how well the model classifies new data using the test set.
5. **Optimization and Deployment**
   - **Parameter Tuning:** Employ techniques like cross-validation and grid search to optimize model parameters and enhance performance.
   - **Deployment:** Once validated, deploy the model to classify incoming data in real-time or batch processes, ensuring it continuously supports decision-making with up-to-date predictions.

PRACTICAL 7

Perform the data classification using classification algorithm.

OR

Data Analysis using Time Series Analysis

Software required: R 3.5.1

Time series is a series of data points in which each data point is associated with a timestamp. A simple example is the price of a stock in the stock market at different points of time on a given day. Another example is the amount of rainfall in a region at different months of the year. R language uses many functions to create, manipulate and plot the time-series data. The data for the time series is stored in an R object called time-series object. It is also a R data object like a vector or data frame.

The time series object is created by using the ts() function.

Syntax

The basic syntax for ts() function in time series analysis is − timeseries.object.name <- ts(data, start, end, frequency)

Following is the description of the parameters used −

- data is a vector or matrix containing the values used in the time series.
- start specifies the start time for the first observation in time series.
- end specifies the end time for the last observation in time series.
- frequency specifies the number of observations per unit time. Except the parameter "data" all other parameters are optional

Consider the annual rainfall details at a place starting from January 2012. We create an R time series object for a period of 12 months and plot it.

Code to run in R

```
# Get the data points in form of a R vector.
rainfall <- c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)


# Convert it to a time series object.
rainfall.timeseries <- ts(rainfall,start = c(2012,1),frequency = 12)
```

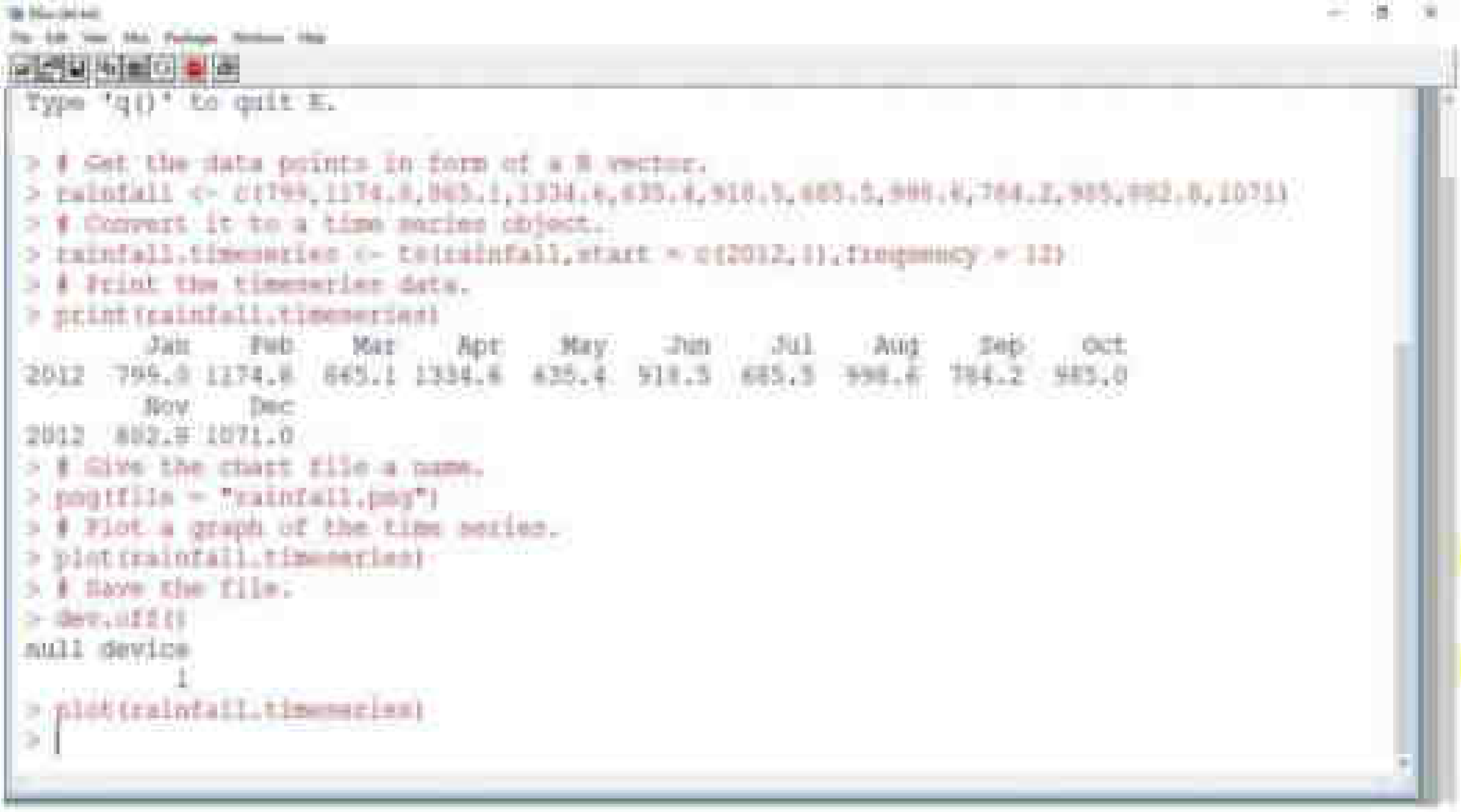
```
# Print the timeseries data.  print(rainfall.timeseries)

# Give the chart file a name.  png(file =
"rainfall.png")

# Plot a graph of the time series.
plot(rainfall.timeseries)

# Save the file.
dev.off()
```

After this again plot to get chart plot(rainfall.timeseries)

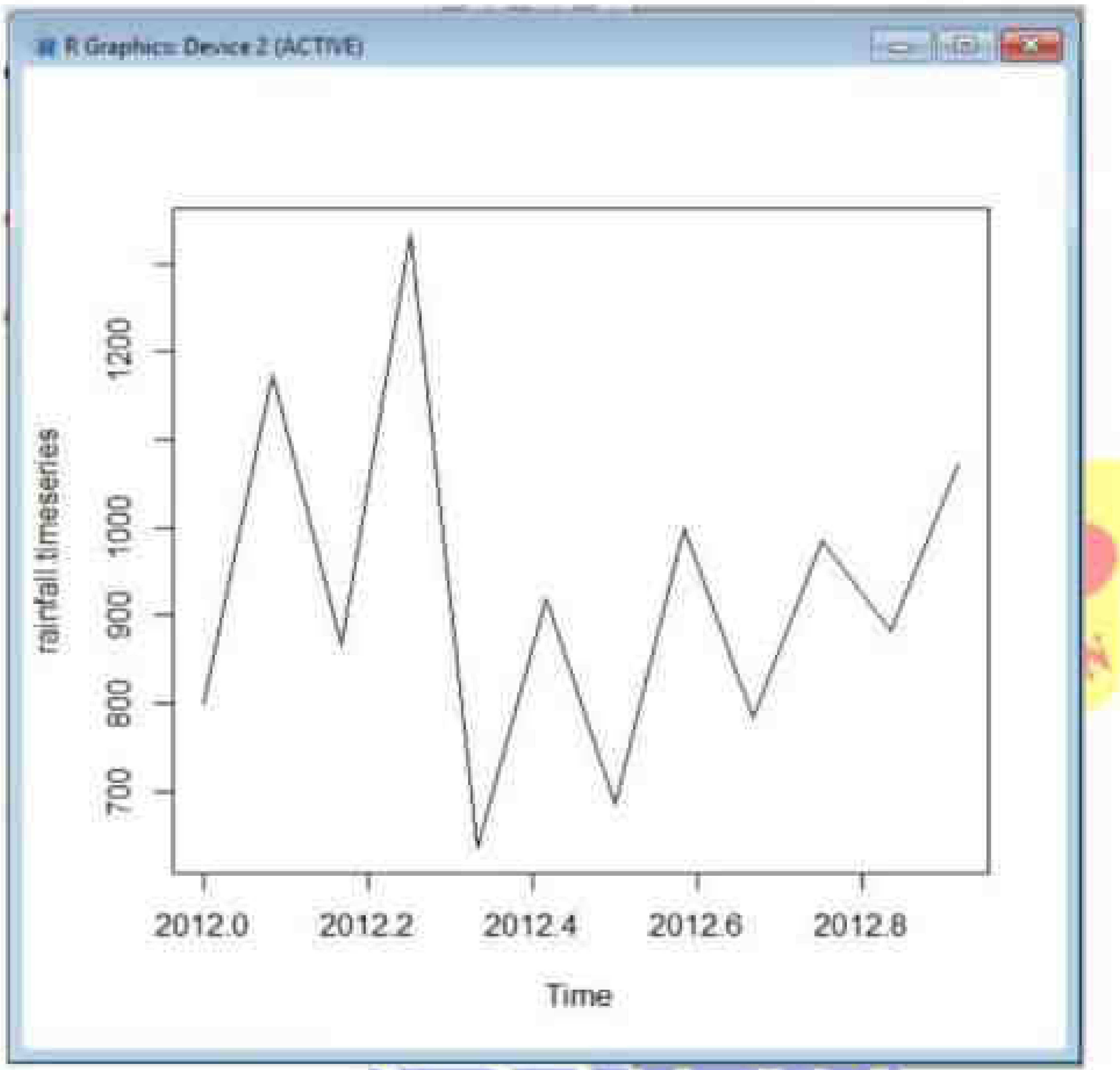
Output:

When we execute the above code, it produces the following result and chart −

```
Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep 2012  799.0  1174.8
865.1  1334.6  635.4  918.5  685.5  998.6  784.2        Oct    Nov    Dec 2012  985.0  882.8 1071.0
```



```
Type 'q()' to quit R.

> # Get the data points in form of a R vector.
> rainfall <- c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)
> # Convert it to a time series object.
> rainfall.timeseries <- ts(rainfall,start = c(2012,1),frequency = 12)
> # Print the timeseries data.
> print(rainfall.timeseries)
        Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
2012  799.0 1174.8  865.1 1334.6  635.4  918.5  685.5  998.6  784.2  985.0
        Nov    Dec
2012  882.8 1071.0
> # Give the chart file a name.
> png(file = "rainfall.png")
> # Plot a graph of the time series.
> plot(rainfall.timeseries)
> # Save the file.
> dev.off()
null device
          1
> plot(rainfall.timeseries)
> |
```


R Graphics: Device 2 (ACTIVE)
rainfall timeseries
700
800
900
1000
1200
2012.0
2012.2
2012.4
2012.6
2012.8
Time

Learning Outcomes:

Course Outcomes:

Conclusion:

Viva Questions:

1. **What is classification?**
2. **What is supervised classification?**
3. **Why is classification important?**
4. **Example of a classification task?**

For Faculty use:

Correction Parameters	Formative Assessment[40%]	Timely Completion of Practical[40%]	Attendance Learning Attitude[20%]

Practical-8: Perform the data clustering using clustering algorithm.

Aims:

1. To understand the fundamentals of clustering and its application in grouping similar data points.
2. To discover inherent patterns and segments within datasets using unsupervised learning techniques.

Learning Objectives:

1. Comprehend the basic concepts of unsupervised learning and clustering methods.
2. Gain hands-on experience with popular clustering algorithms such as K-means and hierarchical clustering.
3. Learn how to preprocess, analyze, and visualize data for effective clustering.
4. Evaluate the quality of clusters using metrics like silhouette scores.

Tool & Technologies used:

1. Power BI is a powerful business intelligence tool used for data visualization and analysis.

Theory-8: Clustering

Clustering is an unsupervised learning technique that groups similar data points into clusters, uncovering inherent structures in the dataset without predefined labels. This approach is useful for segmenting data, discovering patterns, and reducing data complexity in areas such as market segmentation, anomaly detection, and image analysis.

Process of Data Clustering

1. **Data Preparation**
   - **Data Collection & Cleaning:** Gather a comprehensive dataset and clean it by removing noise, handling missing values, and eliminating outliers.
   - **Normalization & Scaling:** Standardize or normalize features to ensure that no single feature dominates the clustering due to scale differences.
2. **Feature Selection and Extraction**
   - **Feature Selection:** Identify key variables that capture the essence of the data, ensuring that the chosen features enhance the distinction between clusters.
   - **Dimensionality Reduction:** Apply techniques such as Principal Component Analysis (PCA) if necessary to reduce dimensionality while preserving important information.
3. **Algorithm Selection**

   - **K-Means:** A widely used algorithm that partitions data into k clusters by minimizing the variance within each cluster.
   - **Hierarchical Clustering:** Builds a tree of clusters (dendrogram) and does not require specifying the number of clusters upfront.
   - **DBSCAN:** Groups data points based on density, which is effective for discovering clusters of arbitrary shapes and handling noise.

4. **Model Training and Evaluation**
   - **Clustering Execution:** Run the selected algorithm on the prepared dataset. For example, with K-Means, choose an initial value for k and iterate until the cluster centroids stabilize.
   - **Cluster Validation:** Evaluate the quality of the clusters using metrics such as silhouette scores, Davies-Bouldin index, or within-cluster sum of squares (WCSS).
   - **Parameter Tuning:** Adjust parameters (like the number of clusters in K-Means or the neighborhood radius in DBSCAN) to refine cluster quality.

PRACTICAL 8

Perform the data clustering using clustering algorithm.

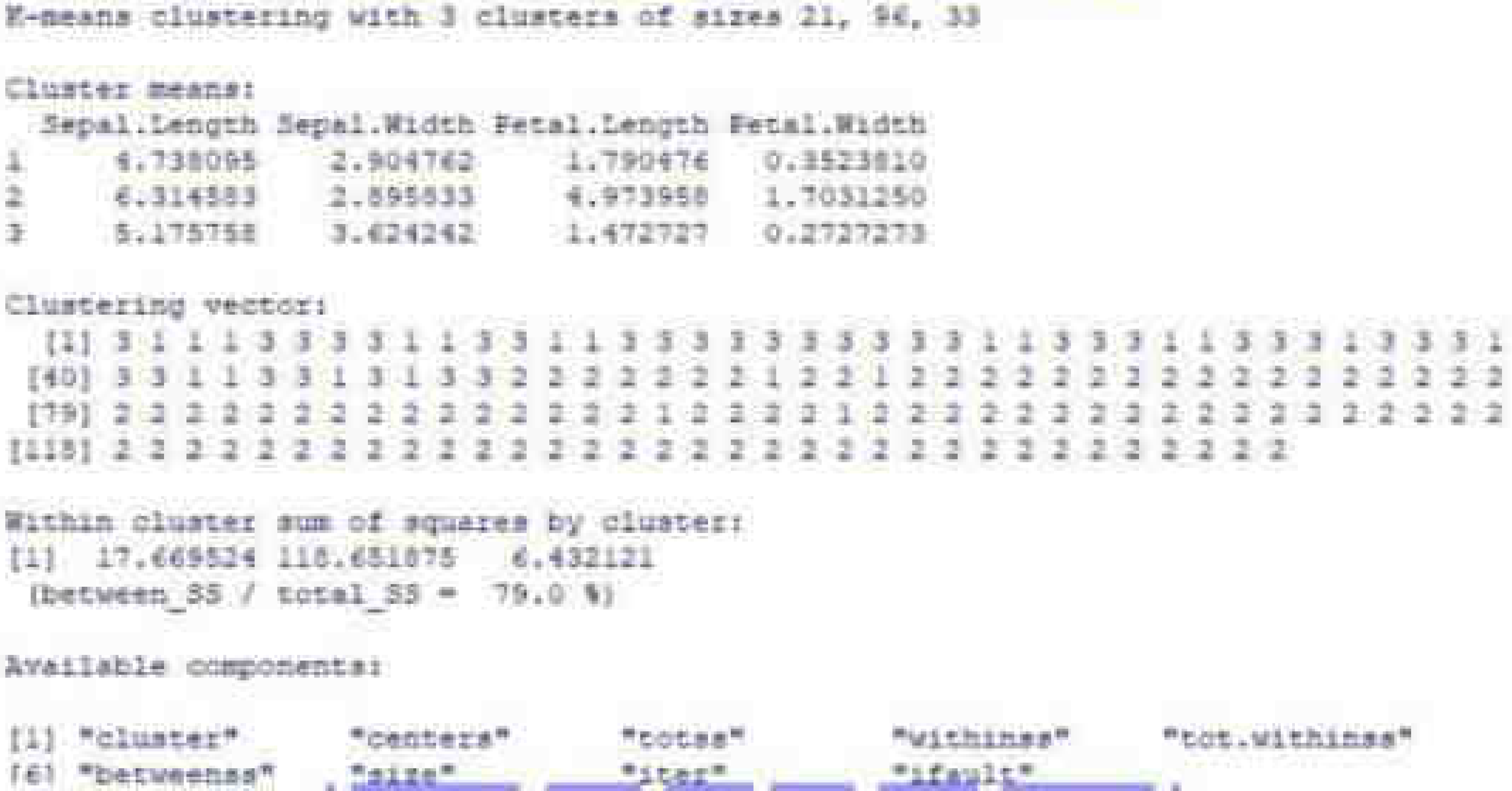
k-means clustering using R

#apply K means to iris and store result

newiris <- iris

newiris$Species <- NULL

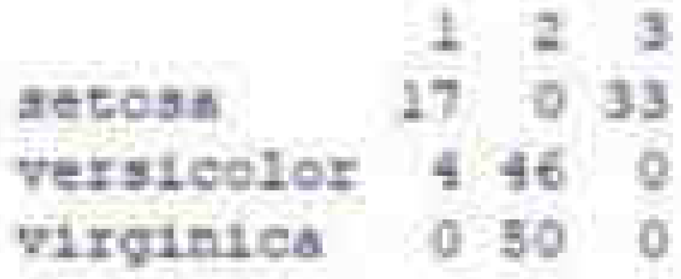
(kc <- kmeans(newiris,3))



```
K-means clustering with 3 clusters of sizes 21, 96, 33

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     4.738095    2.904762     1.790476   0.3523810
2     6.314583    2.895833     4.973958   1.7031250
3     5.175758    3.624242     1.472727   0.2727273

Clustering vector:
  [1] 3 1 1 1 3 3 3 3 1 1 3 3 1 1 3 3 3 3 3 3 3 3 3 3 1 1 3 3 3 1 1 3 3 3 1 3 3 3 1
 [40] 3 3 1 1 3 3 1 3 1 3 3 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [79] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[118] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

Within cluster sum of squares by cluster:
[1]  17.669524 118.651875   6.432121
 (between_SS / total_SS =  79.0 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

#Compare the Species label with the clustering result

table(iris$Species,kc$cluster)



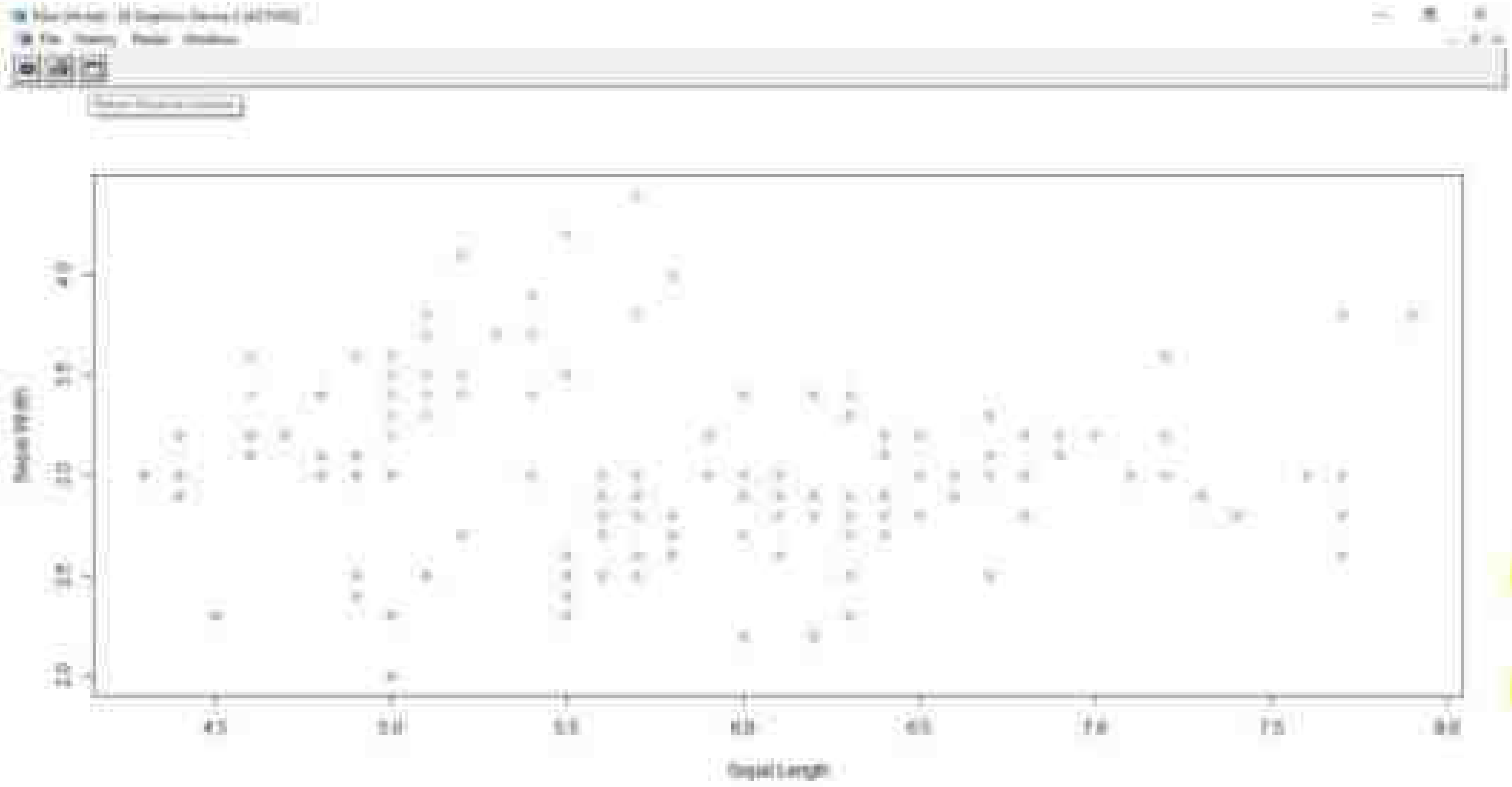
```
              1  2  3
  setosa     17  0 33
  versicolor  4 46  0
  virginica   0 50  0
```

#Plot the clusters and their centers

plot(newiris[c("Sepal.Length","Sepal.Width")],col=kc$cluster)
points(kc$centers[,c("Sepal.Length","Sepal.Width")],col=1:3,pch=8,cex=2) dev.off()

#Plot the clusters and their centre

plot(newiris[c("Sepal.Length","Sepal.Width")],col=kc$cluster)


R Graphics: Device 2 (ACTIVE)
File History Resize Windows
Sepal Width
Sepal Length
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

Learning Outcomes:

Course Outcomes:

Conclusion:

Viva Questions:

1. **What is clustering?**
2. **Name a clustering algorithm.**
3. **What is the purpose of clustering?**
4. **Difference between clustering and classification?**

For Faculty use:

Correction Parameters	Formative Assessment[40%]	Timely Completion of Practical[40%]	Attendance Learning Attitude[20%]

Practical-9: Perform the Linear regression on the given data ware house data.

Aims:

1. To apply linear regression techniques on data extracted from a data warehouse.
2. To forecast relationships between variables and derive predictive insights for informed decision-making.

Learning Objectives:

1. Understand the fundamental concepts and assumptions of linear regression.
2. Gain practical experience in preprocessing and cleaning data for regression analysis.
3. Learn to implement linear regression models using Python and evaluate model performance using metrics such as R-squared, Mean Absolute Error (MAE), and Mean Squared Error (MSE).
4. Interpret the regression coefficients to assess the impact of independent variables on the target variable.

Tool & Technologies used:

1. Power BI is a powerful business intelligence tool used for data visualization and analysis.

Theory-9: Linear regression

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. It helps identify trends, make predictions, and derive insights from data warehouse information. This technique is widely used for forecasting, performance analysis, and decision-making in business intelligence.

Steps to Perform Linear Regression

1. Data Preparation and Extraction

- Extract structured data from the data warehouse using SQL queries or data connectors.
- Identify the target variable (dependent) and relevant predictors (independent variables).
- Perform data cleaning by handling missing values, removing duplicates, and standardizing formats.

2. Feature Selection and Preprocessing

- Analyze feature correlations to select the most relevant independent variables.
- Normalize or scale variables if they have different units to improve model accuracy.
- Split the dataset into training and testing subsets (e.g., 80% for training, 20% for testing).

3. Model Evaluation and Interpretation

- Assess the model's performance using metrics such as:
  - **R-squared (R^2):** Measures how well the model explains the variability in data.
  - **Mean Squared Error (MSE) and Root Mean Squared Error (RMSE):** Evaluate prediction accuracy.
  - **Residual Analysis:** Check for normal distribution and homoscedasticity of residuals.

4. Prediction and Business Application

- Apply the trained model to new warehouse data for forecasting and decision-making.
- Use insights from the model to optimize business strategies, such as sales predictions, resource allocation, and trend analysis.

PRACTICAL 9

Perform the Linear regression on the given data warehouse data.

Input Data

Below is the sample data representing the observations –

Values of height

151, 174, 138, 186, 128, 136, 179, 163, 152, 131

Values of weight

63, 81, 56, 91, 47, 57, 76, 72, 62, 48

lm() Function :

This function creates the relationship model between the predictor and the response variable.

Syntax :

The basic syntax for lm() function in linear regression is − lm(formula,data)

Following is the description of the parameters used :-

- formula is a symbol presenting the relation between x and y.
- data is the vector on which the formula will be applied.

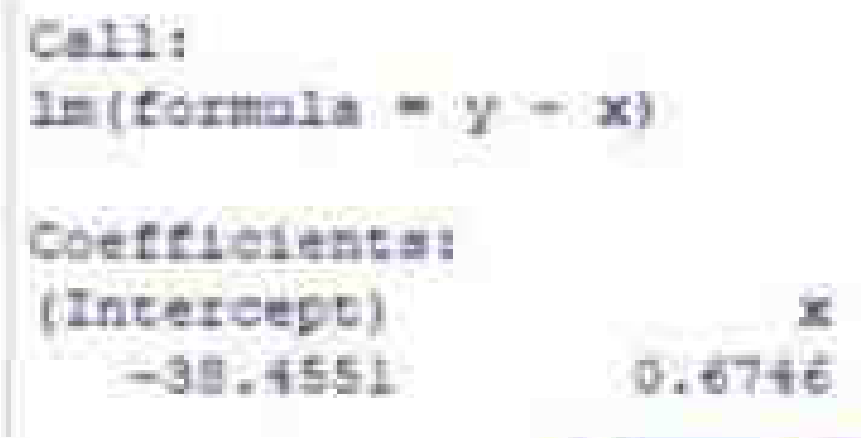
A. Create Relationship Model & get the Coefficients # Values of height x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

Values of width y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62,
48)

Apply the lm() function.

relation <- lm(y~x) print(relation)

OUTPUT:

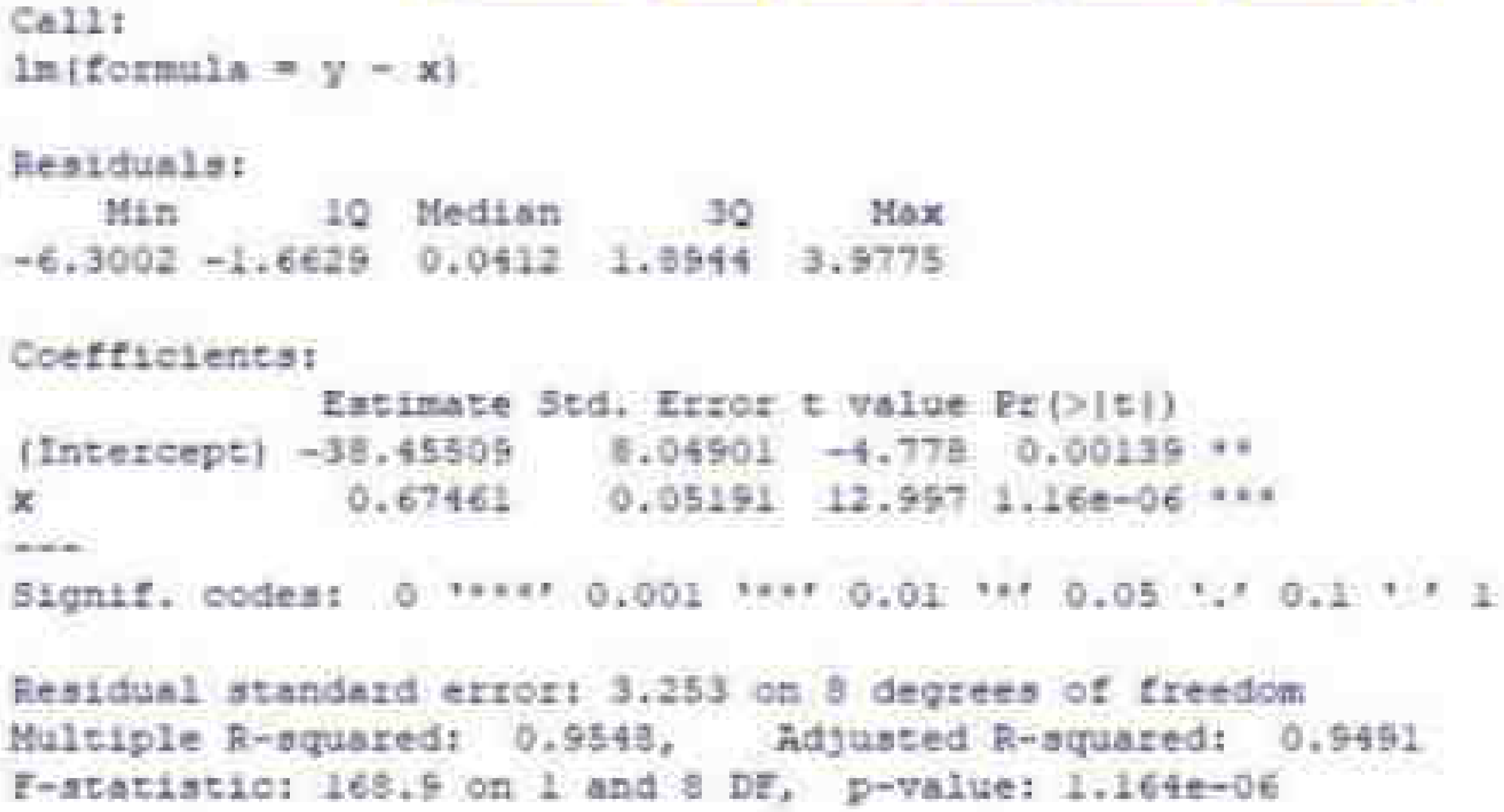


```
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
   -38.4551       0.6746
```

B. Get the Summary of the Relationship # Values of height x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

Values of width y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

Apply the lm() function. relation <- lm(y~x) print(summary(relation)) OUTPUT:



```
Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-6.3002 -1.6629  0.0412  1.8944  3.9775

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -38.45509    8.04901  -4.778  0.00139 **
x             0.67461    0.05191  12.997 1.16e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.253 on 8 degrees of freedom
Multiple R-squared:  0.9548,    Adjusted R-squared:  0.9491
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.164e-06
```

predict() Function

Syntax

The basic syntax for predict() in linear regression is – predict(object, newdata)

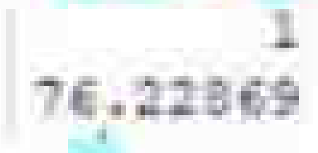
Following is the description of the parameters used –

- object is the formula which is already created using the lm() function.
- newdata is the vector containing the new value for predictor variable.

C. Predict the weight of new persons

```
# The predictor vector.
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

# The response vector.  y <- c(63, 81, 56, 91, 47, 57, 76,
72, 62, 48)

# Apply the lm() function.
relation <- lm(y~x)

# Find weight of a person with height 170.
a <- data.frame(x = 170)  result <-
predict(relation,a)  print(result)
```

OUTPUT:

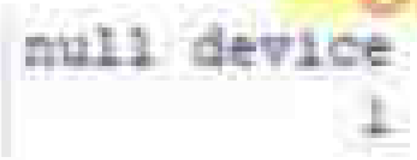


```
       1
76.22869
```

D. Visualize the Regression Graphically # Create the predictor and response variable. x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48) relation <- lm(y~x)

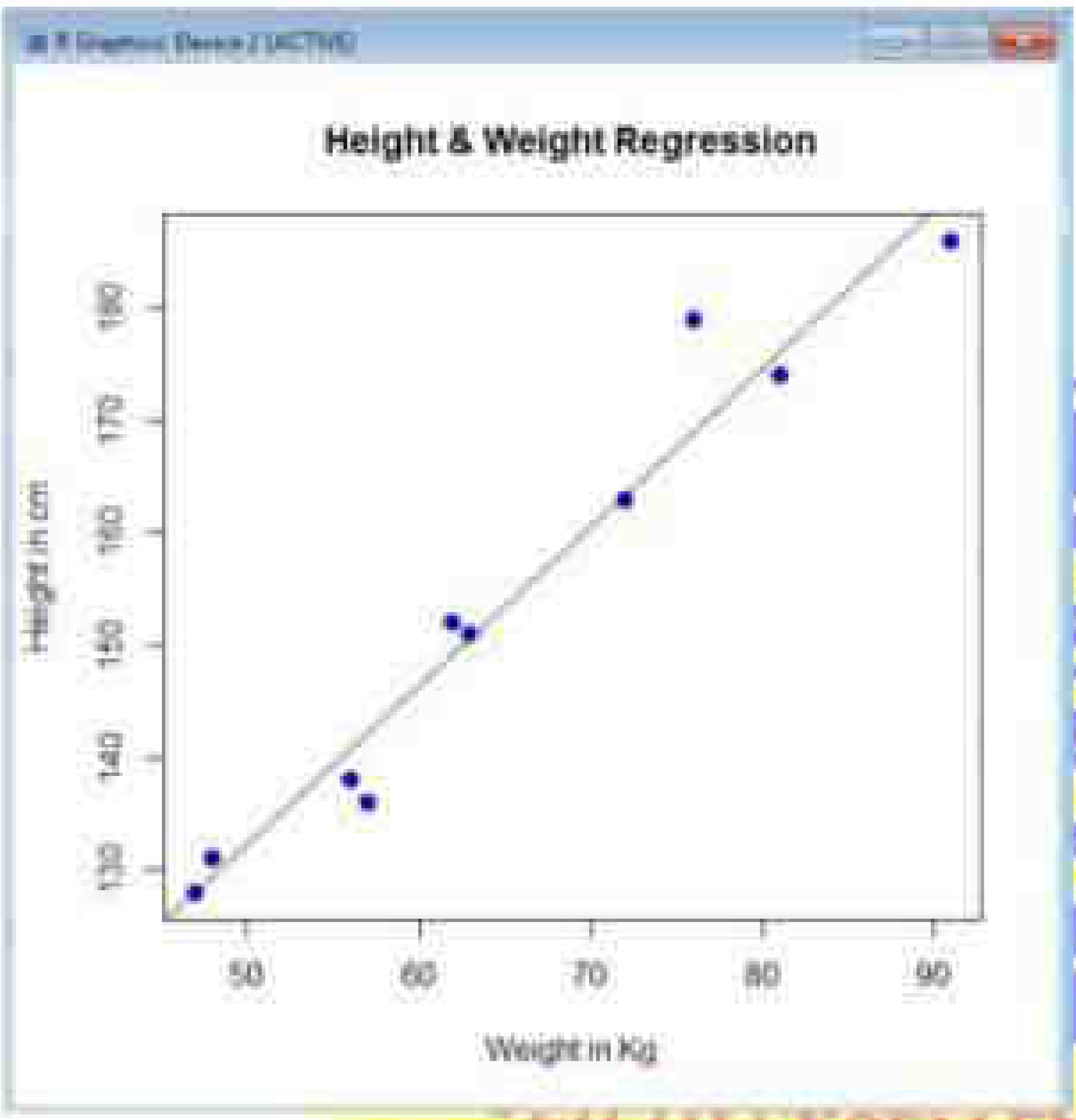
```
# Give the chart file a name.

png(file = "linearregression.png")


# Plot the chart.

plot(y,x,col = "blue",main = "Height & Weight Regression", abline(lm(x~y)),cex = 1.3,pch = 16,xlab =
"Weight in Kg",ylab = "Height in cm")


# Save the file.

dev.off()
```



```
null device
          1
```

```
# Plot the chart.

plot(y,x,col = "blue",main = "Height & Weight Regression", abline(lm(x~y)),cex = 1.3,pch = 16,xlab =
"Weight in Kg",ylab = "Height in cm")
```

OUTPUT:


R Graphics: Device 2 (ACTIVE)
Height & Weight Regression
Height in cm
130
140
150
160
170
180
Weight in Kg
50
60
70
80
90

Learning Outcomes:

Course Outcomes:

Conclusion:

Viva Questions:

1. **What is linear regression?**
2. **Name one assumption of linear regression.**
3. **What is the purpose of the slope coefficient?**
4. **How is the R-squared value interpreted?**

For Faculty use:

Correction Parameters	Formative Assessment[40%]	Timely Completion of Practical[40%]	Attendance Learning Attitude[20%]

Practical-10: Perform the logistic regression on the given data ware house -data.

Aims:

1. To apply logistic regression techniques on data extracted from a data warehouse for binary classification tasks.
2. To predict the probability of a binary outcome based on multiple predictor variables.

Learning Objectives:

1. Understand the fundamentals and assumptions of logistic regression.
2. Gain hands-on experience in preparing and preprocessing data for logistic regression analysis.
3. Evaluate model performance using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.
4. Interpret model coefficients to derive insights for decision-making.

Tool & Technologies used:

1. Power BI is a powerful business intelligence tool used for data visualization and analysis.

Theory-10: logistic regression

Logistic regression is a statistical and machine learning technique used for classification problems, where the target variable is categorical (e.g., binary classification: yes/no, success/failure). Unlike linear regression, logistic regression models the probability that an instance belongs to a particular category using the logistic (sigmoid) function.

Steps to Perform Logistic Regression

1. Data Preparation and Extraction

- Extract structured data from the data warehouse using SQL queries or data connectors.
- Identify the dependent variable (categorical outcome) and independent variables (predictors).
- Perform data cleaning by handling missing values, standardizing formats, and removing duplicates.

2. Feature Selection and Preprocessing

- Select the most relevant independent variables using correlation analysis.
- Convert categorical variables into numerical format using encoding techniques (e.g., one-hot encoding).
- Normalize or scale numerical variables if required.
- Split the dataset into training and testing subsets (e.g., 80% training, 20% testing).

3. Model Evaluation and Performance Metrics

- **Accuracy Score:** Measures overall correctness of predictions.
- **Confusion Matrix:** Displays true positives, false positives, true negatives, and false negatives.
- **Precision, Recall, and F1-Score:** Evaluate classification performance.
- **ROC Curve & AUC Score:** Analyze the model's ability to distinguish between classes.

4. Prediction and Business Application

- Apply the trained model to new warehouse data for predictive analysis.
- Use logistic regression to classify outcomes such as customer churn, fraud detection, risk assessment, and marketing segmentation.

PRACTICAL 10

Perform the logistic regression on the given data warehouse data.

To perform this you need to download quality.csv file from following link:

https://github.com/TarekDib03/Analytics/tree/master/Week3%20-%20Logistic%20Regression/Data

```
#provide path of file where it is saved on your machine quality <-
read.csv('C:/Users/Gauri/Downloads/quality.csv')
> #analysing the quality dataset
> str(quality)
'data.frame':   131 obs. of  14 variables:
 $ MemberID           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ InpatientDays      : int  0 1 0 0 8 2 16 2 2 4 ...
 $ ERVisits           : int  0 1 0 1 2 0 1 0 1 2 ...
 $ OfficeVisits       : int  18 6 5 19 19 9 8 8 4 0 ...
 $ Narcotics          : int  1 1 3 0 3 2 1 0 3 2 ...
 $ DaysSinceLastERVisit: num  731 411 731 158 449 ...
 $ Pain               : int  10 0 10 34 10 6 4 5 5 2 ...
 $ TotalVisits        : int  18 8 5 20 29 11 25 10 7 6 ...
 $ ProviderCount      : int  21 27 16 14 24 40 19 11 28 21 ...
 $ MedicalClaims      : int  93 19 27 59 51 53 40 28 20 17 ...
 $ ClaimLines         : int  222 115 148 242 204 156 261 87 98 66 ...
 $ StartedOnCombination: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ AcuteDrugGapSmall  : int  0 1 5 0 0 4 0 0 0 0 ...
 $ PoorCare           : int  0 0 0 0 0 1 0 0 1 0 ...
> table(quality$PoorCare)

 0  1
```

```
98 33

> 98/131

[1] 0.7480916


> install.packages("caTools")

Installing package into 'C:/Users/Gauri/Documents/R/win-library/3.5'

(as 'lib' is unspecified)

--- Please select a CRAN mirror for use in this session --- also installing the

dependency 'bitops'


trying URL
'http://mirror.its.dal.ca/cran/bin/windows/contrib/3.5/bitops_1.0-6.zip' Content type

'application/zip' length 38894 bytes (37 KB) downloaded 37 KB


trying URL
'http://mirror.its.dal.ca/cran/bin/windows/contrib/3.5/caTools_1.17.1.1.zi p'

Content type 'application/zip' length 329665 bytes (321 KB) downloaded 321 KB


package 'bitops' successfully unpacked and MD5 sums checked
package 'caTools' successfully unpacked and MD5 sums
checked


The downloaded binary packages are in

C:\Users\Gauri\AppData\Local\Temp\RtmpmUN9oK\downloaded_package s

> library(caTools) Warning

message:

package 'caTools' was built under R version 3.5.2

> set.seed(88)
> split = sample.split(quality$PoorCare, SplitRatio = 0.75) >

> split

  [1]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE
FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE

 [28]  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE
TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE
FALSE  TRUE  TRUE FALSE FALSE  TRUE

 [55]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
TRUE  TRUE  TRUE  TRUE  TRUE

 [82]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE

[109]  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE

> qualityTrain = subset(quality, split == TRUE) > qualityTest =

subset(quality, split == FALSE)

> nrow(qualityTrain)

[1] 99

> nrow(qualityTest)

[1] 32

> QualityLog = glm(PoorCare ~ OfficeVisits + Narcotics,data=qualityTrain, family=binomial)

> summary(QualityLog)


Call:

glm(formula = PoorCare ~ OfficeVisits + Narcotics, family = binomial,     data = qualityTrain)


Deviance Residuals:

    Min       1Q   Median       3Q      Max

-2.06303  -0.63155  -0.50503  -0.09689   2.16686


Coefficients:
```

```
         Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.64613    0.52357  -5.054 4.33e-07 ***
OfficeVisits  0.08212    0.03055   2.688  0.00718 **
Narcotics     0.07630    0.03205   2.381  0.01728 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 111.888  on 98  degrees of freedom
Residual deviance:  89.127  on 96  degrees of freedom
AIC: 95.127

Number of Fisher Scoring iterations: 4

> predictTrain = predict(QualityLog, type="response")
> summary(predictTrain)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.06623 0.11912 0.15967 0.25253 0.26765 0.98456
> tapply(predictTrain, qualityTrain$PoorCare, mean)
0         1
0.1894512 0.4392246
> table(qualityTrain$PoorCare, predictTrain > 0.5)

    FALSE TRUE
0     70    4
1     15   10
> 10/25
```

```
[1] 0.4

> 70/74

[1] 0.9459459

> table(qualityTrain$PoorCare, predictTrain > 0.7)


   FALSE TRUE

0     73   1

1     17   8

> 8/25

[1] 0.32

> 73/74

[1] 0.9864865

> table(qualityTrain$PoorCare, predictTrain > 0.2)


   FALSE TRUE

0      54  20

1       9  16

> 16/25

[1] 0.64

> 54/74

[1] 0.7297297

> install.packages("ROCR")

Installing package into 'C:/Users/Gauri/Documents/R/win-library/3.5'

(as 'lib' is unspecified) also installing the

dependencies 'gtools', 'gdata', 'gplots'


trying URL
```

```
'http://mirror.its.dal.ca/cran/bin/windows/contrib/3.5/gtools_3.8.1.zip' Content type

'application/zip' length 325812 bytes (318 KB) downloaded 318 KB


trying URL
'http://mirror.its.dal.ca/cran/bin/windows/contrib/3.5/gdata_2.18.0.zip' Content type

'application/zip' length 1260728 bytes (1.2 MB) downloaded 1.2 MB


trying URL
'http://mirror.its.dal.ca/cran/bin/windows/contrib/3.5/gplots_3.0.1.1.zip' Content type 'application/zip'

length 656764 bytes (641 KB) downloaded 641 KB


trying URL
'http://mirror.its.dal.ca/cran/bin/windows/contrib/3.5/ROCR_1.0-7.zip' Content type

'application/zip' length 201823 bytes (197 KB) downloaded 197 KB
```

package 'gtools' successfully unpacked and MD5 sums checked

package 'gdata' successfully unpacked and MD5 sums checked

package 'gplots' successfully unpacked and MD5 sums checked

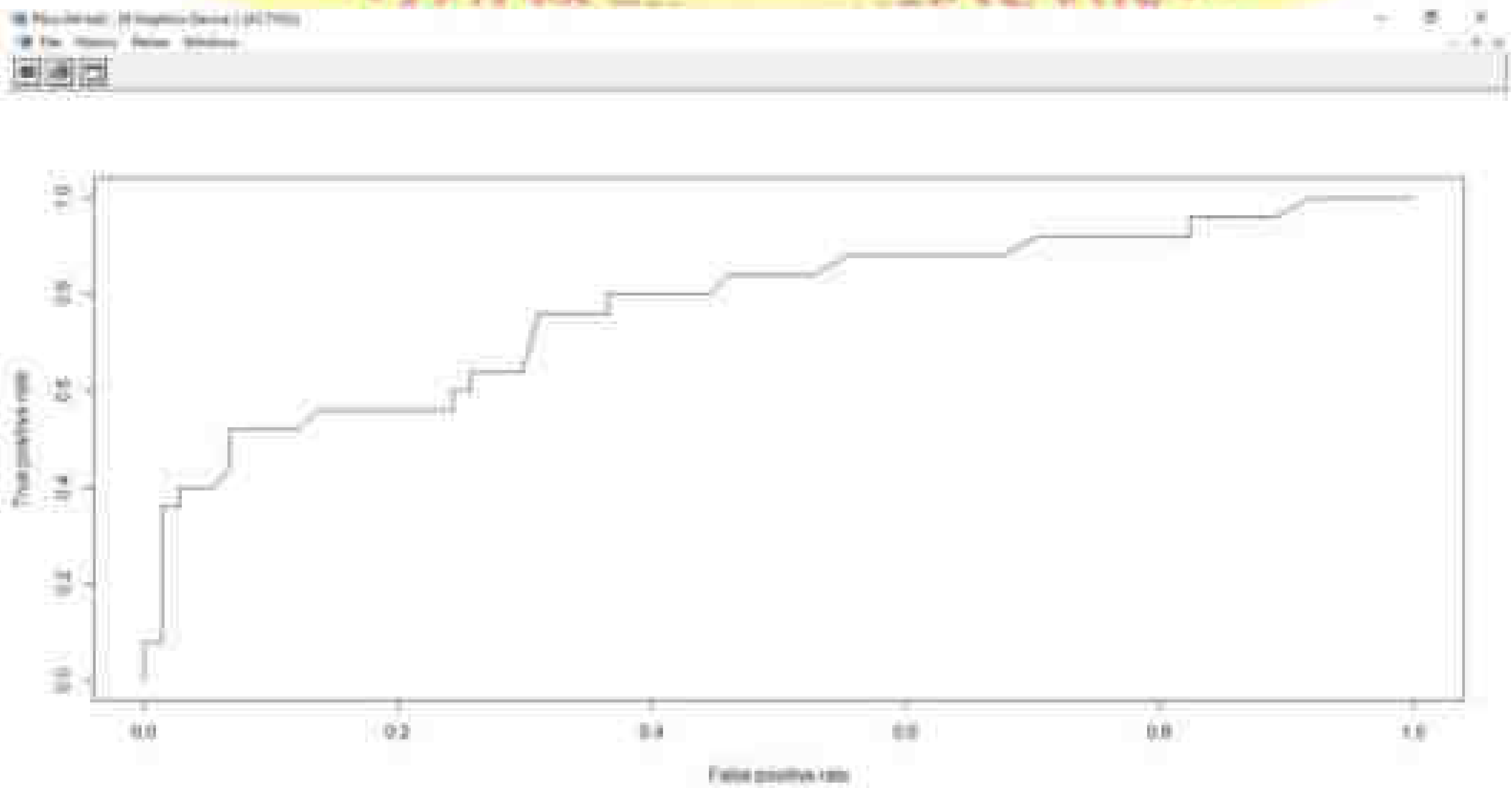
package 'ROCR' successfully unpacked and MD5 sums checked

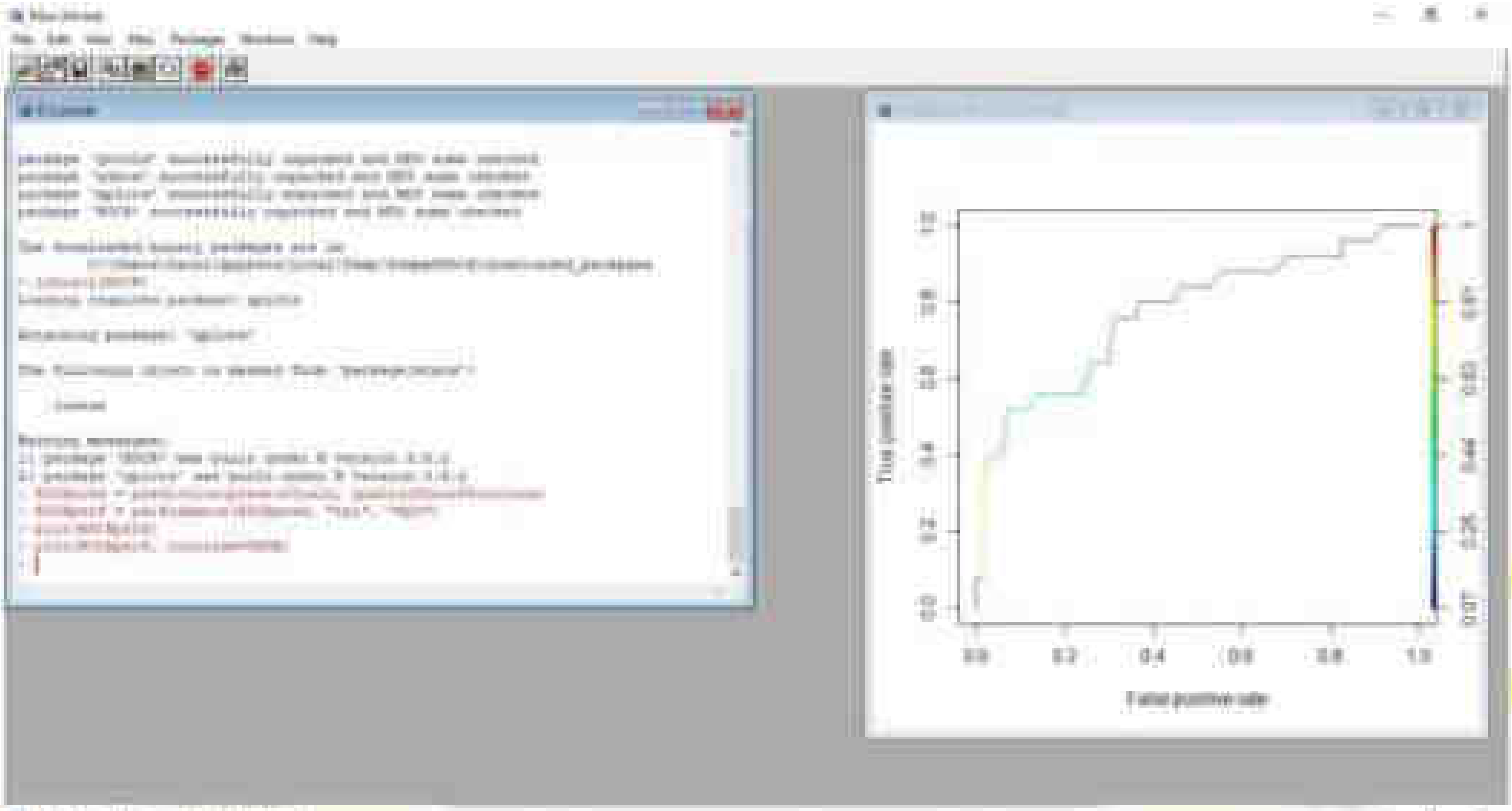
```
The downloaded binary packages are in

C:\Users\Gauri\AppData\Local\Temp\RtmpmUN9oK\downloaded_package s

> library(ROCR)
Loading required package: gplots
```

Attaching package: 'gplots'

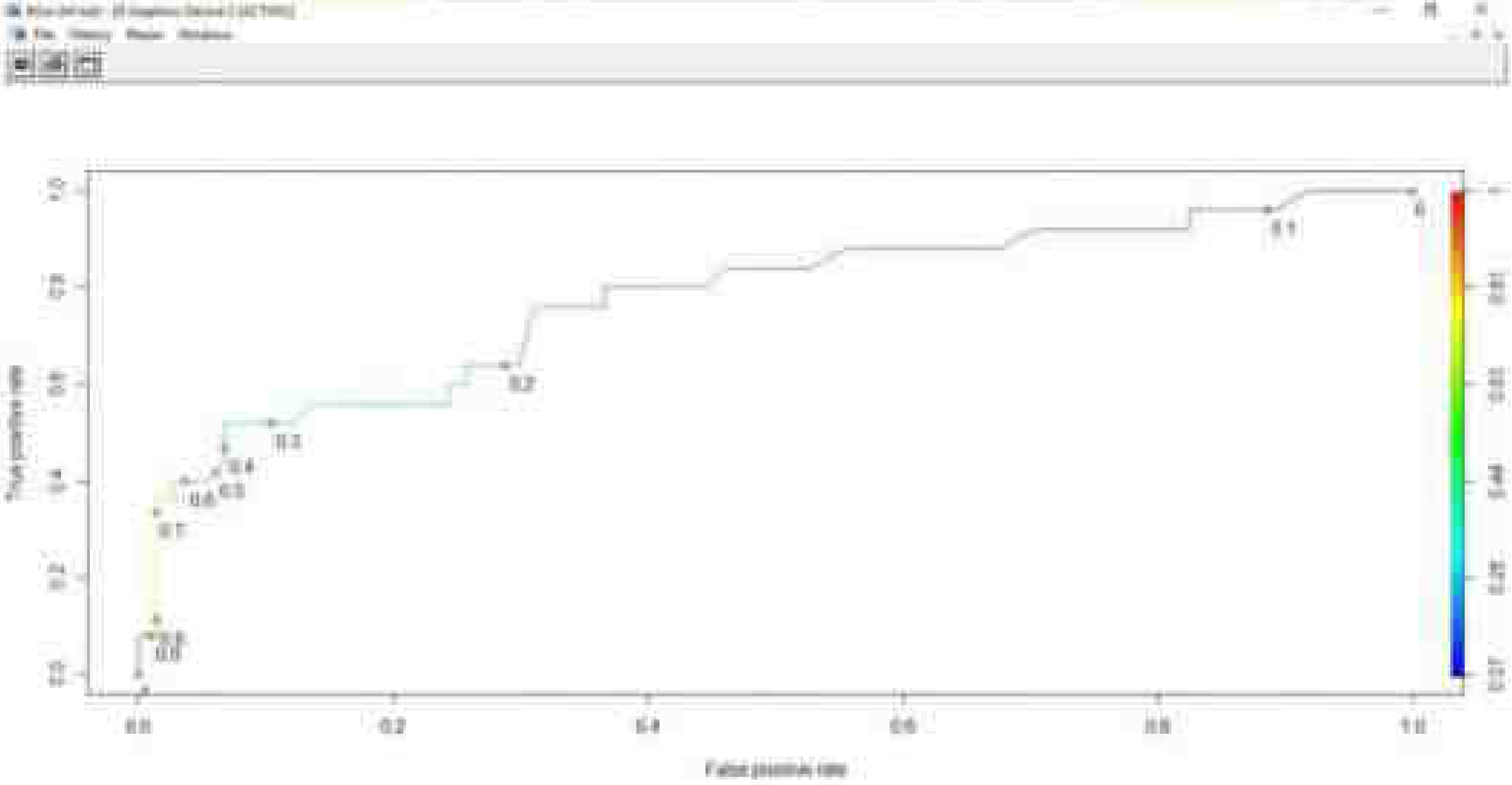
```
The following object is masked from 'package:stats':
```

```
lowess

Warning messages:

1: package 'ROCR' was built under R version 3.5.2

2: package 'gplots' was built under R version 3.5.2

> ROCRpred = prediction(predictTrain, qualityTrain$PoorCare)

> ROCRperf = performance(ROCRpred, "tpr", "fpr")

> plot(ROCRperf)

> plot(ROCRperf, colorize=TRUE)

> plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))

>
```





```
package 'gtools' successfully unpacked and MD5 sums checked
package 'gdata' successfully unpacked and MD5 sums checked
package 'gplots' successfully unpacked and MD5 sums checked
package 'ROCR' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        [illegible]
> library(ROCR)
Loading required package: gplots

Attaching package: 'gplots'

The following object is masked from 'package:stats':

    lowess

Warning messages:
1: package 'ROCR' was built under R version 3.5.2
2: package 'gplots' was built under R version 3.5.2
> ROCRpred = prediction(predictTrain, qualityTrain$PoorCare)
> ROCRperf = performance(ROCRpred, "tpr", "fpr")
> plot(ROCRperf)
> plot(ROCRperf, colorize=TRUE)
> 
```



Learning Outcomes:

Course Outcomes:

Conclusion:

Viva Questions:

1. **Define logistic regression.**
2. **Purpose of the logit function?**
3. **Outcome variable type?**
4. **Example use case?**

For Faculty use:

Correction Parameters	Formative Assessment[40%]	Timely Completion of Practical[40%]	Attendance Learning Attitude[20%]