



**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 that 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

Date	INDEX	Pg. No.	Sign.
	<p>Theory-1 : Loading</p> <p>Practical-1: 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 Adventure works, North wind, food mart etc.) (IT Lab)</p> <p>Example-1: Import the legacy data from different sources such as Excel. (IT Lab)</p> <p>Example-2: Show Implementation of Classification algorithm in R -(Homework)</p> <p>Example -3: Import the legacy data from different sources such as Sql Server. (Homework)</p> <p>Practical-1: Familiarizing Quantum GIS: Installation of QGIS, datasets for both Vector and Raster data, Maps. (IT Lab)</p>		
	<p>Theory-2: Extraction</p> <p>Practical-2: Perform the Extraction Transformation and Loading (ETL) -process to construct the database in the Sql server. (IT Lab)</p> <p>Example-1: Perform the Extraction. (IT Lab)</p> <p>Example-2: (A) Create the cube with suitable dimension and fact tables based on ROLAP, MOLAP and HOLAP model. (Homework)</p> <p>Example -3: Perform Transformation. (Homework)</p>		
	<p>Theory-3: Data staging</p> <p>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. (IT Lab)</p> <p>Example-1: Create the cube with suitable dimension and fact tables based on ROLAP. (IT Lab)</p> <p>Example-2: Perform the data clustering using clustering algorithm in R Programming. (Homework)</p> <p>Example -3: Create the cube with suitable dimension and fact tables based on MOLAP. (Homework)</p>		
	<p>Theory-4: ETL</p> <p>Practical-4: a. Create the ETL map and setup the schedule for execution. –</p> <p>b. Execute the MDX queries to extract the data from data warehouse. (IT Lab)</p> <p>Example-1: Execute the MDX queries to extract the data from the Excel. (IT Lab)</p> <p>Example-2: Perform the Linear regression on the given data warehouse data. (Homework)</p> <p>Example-3: Execute the MDX queries to extract the data from the SQL server. (Homework)</p>		
	<p>Theory-5: Data ware house</p> <p>Practical-5: a. Import the data warehouse data in Microsoft 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. (IT Lab)</p> <p>Example-1: Import the data warehouse data in Microsoft Excel and create the Pivot table. (IT Lab)</p> <p>Example-2: Show prediction Using Linear Regression. (Homework)</p> <p>Example-3: Import the data warehouse data in Microsoft Excel and create the PivotChart. (Homework)</p>		
	<p>Theory-6: Data ware house data</p> <p>Practical-6: Apply the what – if Analysis for data visualization. Design and generate necessary reports based on the data warehouse data. (IT Lab)</p> <p>Example-1: Show waterfall graph on data in power bi. (IT Lab)</p> <p>Example-2: perform the logistic regression on the given data warehouse data. (Homework)</p> <p>Example-3: Show use of table and matrix. (Homework)</p>		

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



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.

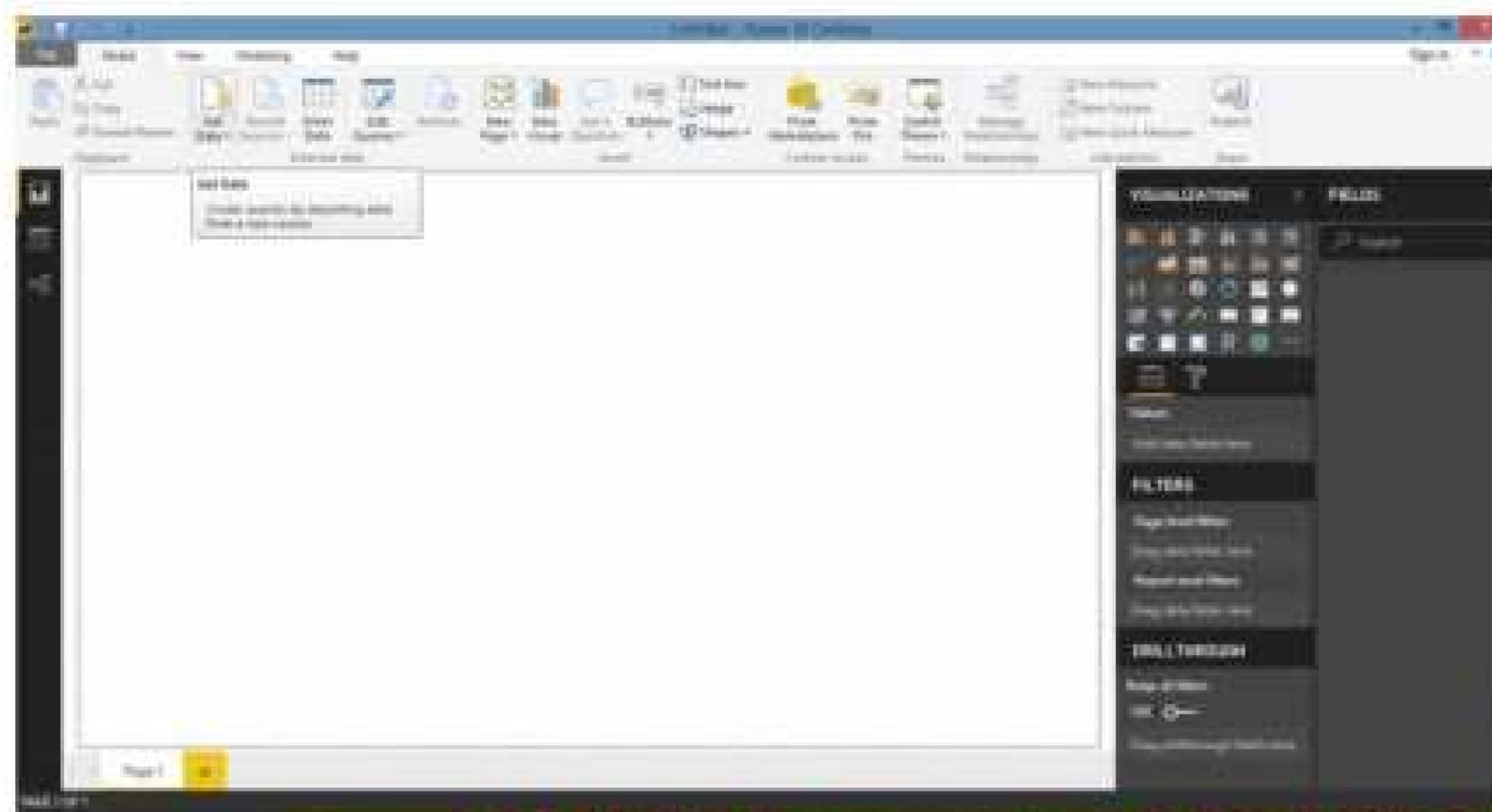
Challenges and Considerations

- **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.

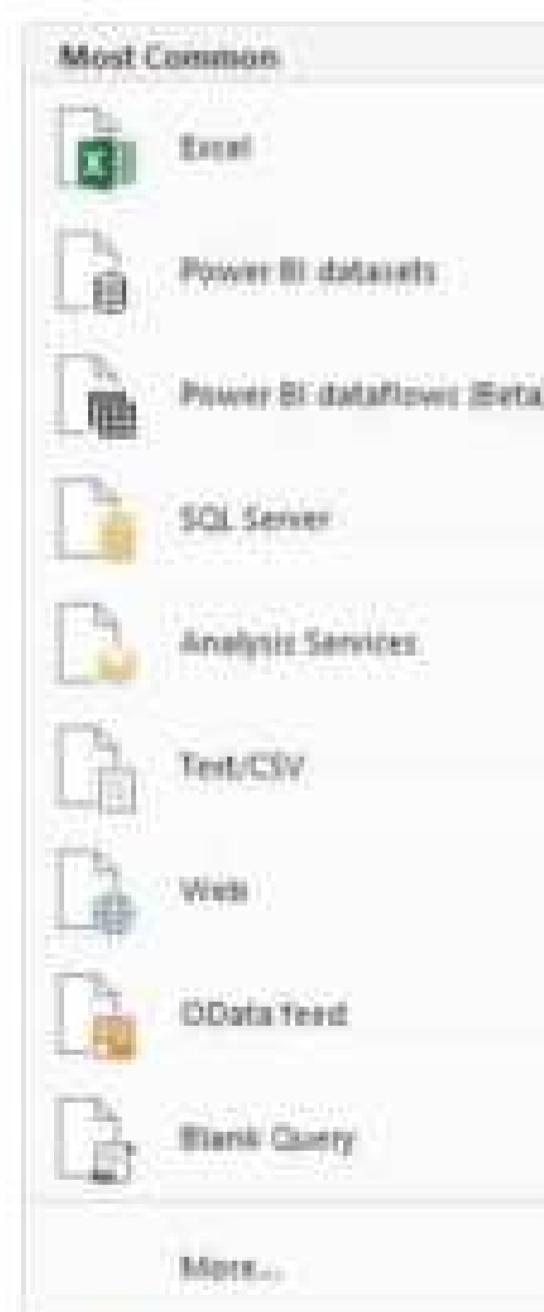


PRACTICAL 1

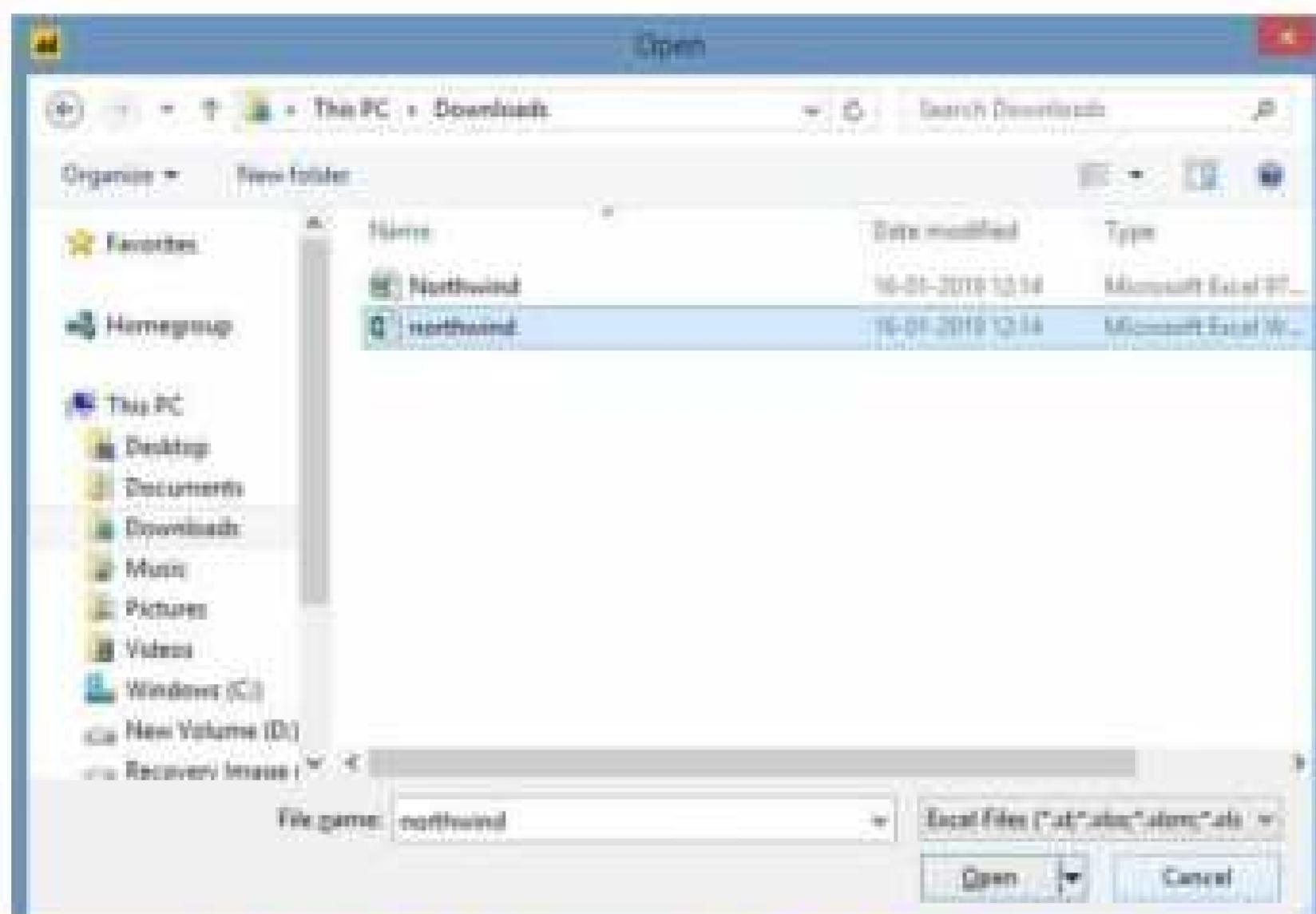
- a. Import the legacy data from different sources such as (Excel , SqlServer, Oracle etc.) and load in the target system. (You can download sample database such as Adventureworks, Northwind, foodmart etc.)
Step 1: Open Power BI



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

नियमित नोह उत्तम सेवाधर्म



Navigator

Display Options: Show formulas Table, show formulas Sheet

No items selected for preview

Close

Co...

Navigator

Display Options: Show formulas Table, show formulas Sheet 1

Table [Customer] (1)

#	Customer Company	Customer Name	Address	Telephone
1	Customer 1	Walter	Italy	
2	Customer 2	William Malone	America	
3	Customer 3	Asia	France	
4	Customer 4	Lee	Germany	
5	Customer 5	W.Brown	Spain	
6	Customer 6	Patricia Werner	Portugal	
7	Customer 7	Witt	Hong Kong	
8	Customer 8	Kurt	England	
9	Customer 9	Markus	USA	
10	Customer 10	Stephan	Spain	
11	Customer 11	Christina	USA	
12	Customer 12	Werner	Spain	
13	Customer 13	Yves	Portugal	
14	Customer 14	Elaine	USA	
15	Customer 15	Janet	Spain	
16	Customer 16	Steve	Germany	
17	Customer 17	Frederick	Portugal	
18	Customer 18	Robert	USA (Mobile)	
19	Customer 19	Patricia	Germany	
20	Customer 20	Ugo	Germany	
21	Customer 21	Winn	Germany	
22	Customer 22	Yann	Germany	

The data in the process has been truncated due to size limit.

Close

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

OK

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



निष्ठानोह उत्तम सेवाधर्म

Navigator

Navigator

Orders					
OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShipVia
102489	VINET	3	04-07-1996 00:00:00	01-08-1996	1
102490	TOMSP	6	05-07-1996 00:00:00	16-08-1996	1
102491	HANAR	8	06-07-1996 00:00:00	03-08-1996	1
102492	VICTR	3	08-07-1996 00:00:00	05-08-1996	1
102493	MATRD	4	09-07-1996 00:00:00	06-08-1996	1
102494	HANAR	3	10-07-1996 00:00:00	24-07-1996	1
102495	CHOPP	3	12-07-1996 00:00:00	09-08-1996	1
102496	FRESCU	9	12-07-1996 00:00:00	09-08-1996	1
102497	WELLU	3	13-07-1996 00:00:00	11-08-1996	1
102498	HILAR	4	16-07-1996 00:00:00	13-08-1996	1
102499	TRINH	1	17-07-1996 00:00:00	14-08-1996	1
102500	CENRC	4	18-07-1996 00:00:00	15-08-1996	1
102501	OTTIK	4	19-07-1996 00:00:00	16-08-1996	1
102502	QUICK	4	19-07-1996 00:00:00	16-08-1996	1
102503	BATTY	8	27-07-1996 00:00:00	24-08-1996	1
102504	ERNSH	3	28-07-1996 00:00:00	25-08-1996	1
102505	POLAQ	8	29-07-1996 00:00:00	26-08-1996	1
102506	BLONP	2	29-07-1996 00:00:00	27-08-1996	1
102507	WARTH	3	29-07-1996 00:00:00	26-08-1996	1
102508	FRANK	4	29-07-1996 00:00:00	26-08-1996	1
102509	GROSR	3	30-07-1996 00:00:00	27-08-1996	1
102510	WHITE	3	31-07-1996 00:00:00	24-08-1996	1
102511	WARTH	3	01-08-1996 00:00:00	29-08-1996	1

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

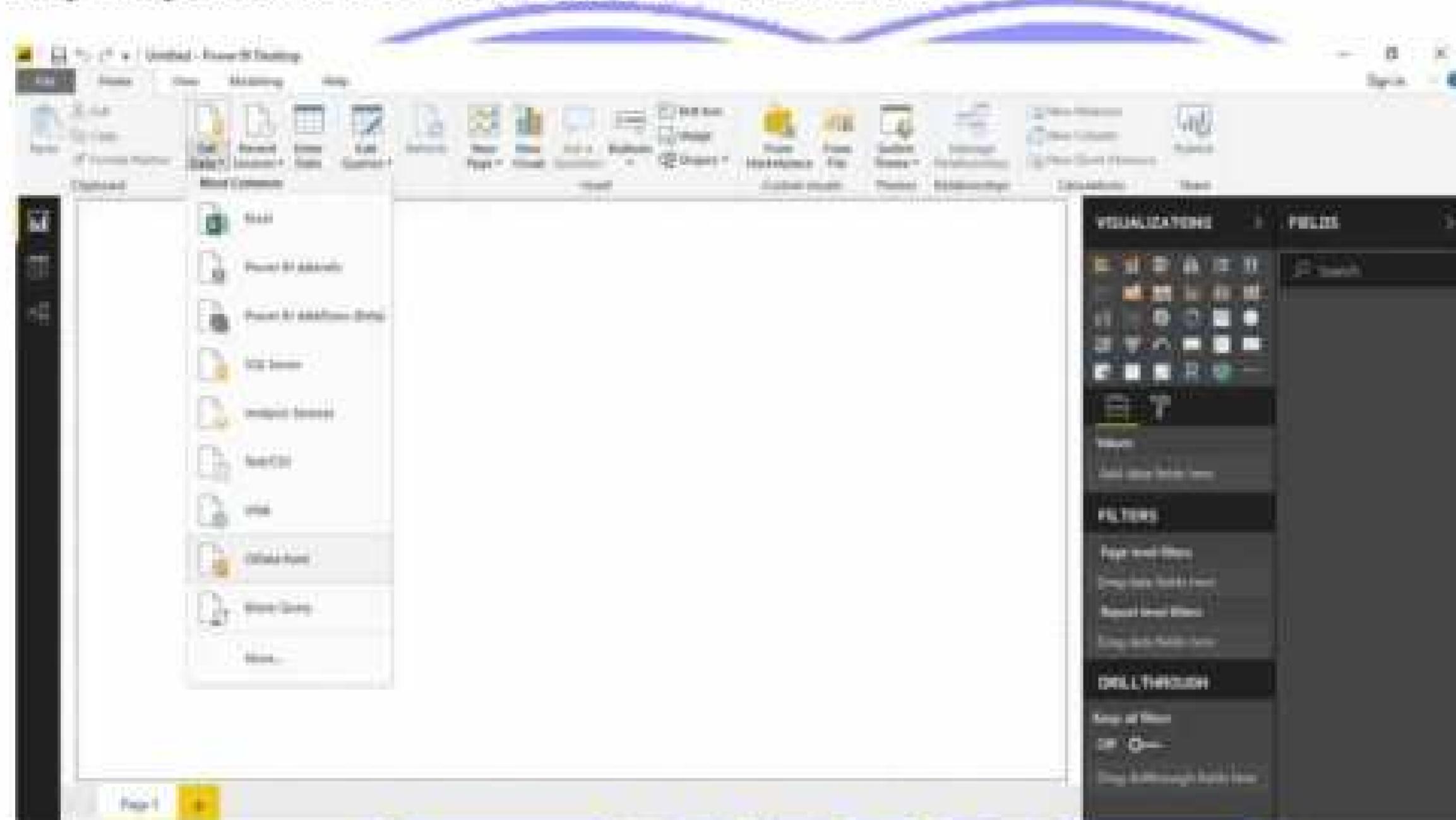
- **Performance Issues:** Large data volumes can impact system performance; optimizing queries and indexing helps mitigate this.
- **Data Integrity:** Ensure extracted data remains consistent and unaltered during transfer.
- **Security Concerns:** Implement encryption and secure connections to protect 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



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



Navigator

Display Options: http://microsoftdata.org/SQLNorthwind/Products

- Alphabetical List of products
- Categories
- Category_Sales_Net_1997
- Current_Product_List
- Customer_and_Distributor_By_City
- CustomerDemographics
- Customers
- Employees
- Invoices
- Order_Details
- Order_Status_Extended
- Order_Statuses
- Orders
- Orders_Own
- Product_Sales_Net_1997
- Products
- Products_Above_Average_Price
- Products_By_Categories
- Regions

Products

ProductID	ProductName	SupplierID	CategoryID	Unit
1	Chai	1	1	20 boxes x 20 bags
2	Chang	1	1	15 kg (5 kg bags)
3	Mt. St. Helen's	2	1	200 ml bottles
4	Chef Anton's Cajun Seasoning	2	2	9 kg per carton
5	Chef Anton's Gumbo Mix	2	2	2 kg bags
6	Grandma's Boysenberry Spread	3	2	12 - 450 g jars
7	Lakewood Organic Dried Pears	3	7	32 - 100 g pkgs
8	Rustic上去的 Cranberry Sauce	3	7	12 - 300 g jars
9	Alfredo's Spaghetti Sauce	4	3	100 kg bags
10	Maria's Italian Rice	4	3	300 g cans
11	Haus	4	3	20 boxes x 1 kg bags
12	Gusto Oysters	5	4	12 kg bags
13	Canned Mango (La枫糖)	5	4	100 g cans
14	Kusto	5	4	100 g cans
15	Tuna	6	2	2 kg cans
16	Golden Moose	6	2	2 kg cans
17	Pavlova	7	3	3 kg bags
18	Alvin's Marmalade	7	4	6 kg bags
19	Uncle Bob's Organic Dried Pears	8	2	12 - 450 g jars
20	Mr. Holmes' Organic Tomatoes	8	2	100 kg bags
21	Smoked Sardines	9	2	100 g cans
22	South's Organic	9	2	100 g cans
23	Uncle Bob's Organic Dried Pears	9	2	12 - 450 g jars
24	Uncle Bob's Organic Tomatoes	9	2	100 kg bags

Select Related Tables: [Select] [OK] [Cancel]



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

Home | Refresh | Add Column | View | Help

File New Source... Data Data Source Settings Manage Parameters Refresh Manage Home | Advanced Editor | Advanced Table | Change Columns | Remove Columns | Help | Remove Rows | Sort | Filter | Group By | Use First Row as Headers | Load & Home Values | Close & Load | Close & Save As | Transform

Queries [1]

Products

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitsInStock
1	Chai	1	1	20 boxes x 20 bags	10
2	Chang	1	1	15 kg (5 kg bags)	15
3	Mt. St. Helen's	2	1	200 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	9 kg per carton	10
5	Chef Anton's Gumbo Mix	2	2	2 kg bags	10
6	Grandma's Boysenberry Spread	3	2	12 - 450 g jars	10
7	Lakewood Organic Dried Pears	3	7	32 - 100 g pkgs	10
8	Rustic上去的 Cranberry Sauce	3	7	12 - 300 g jars	10
9	Alfredo's Spaghetti Sauce	4	3	100 kg bags	10
10	Maria's Italian Rice	4	3	300 g cans	10
11	Haus	4	3	20 boxes x 1 kg bags	10
12	Gusto Oysters	5	4	12 kg bags	10
13	Canned Mango (La枫糖)	5	4	100 g cans	10
14	Kusto	5	4	100 g cans	10
15	Tuna	6	2	2 kg cans	10
16	Golden Moose	6	2	2 kg cans	10

14 COLUMNS, 21 ROWS

REFRESH | TRANSFORM | LOAD & HOME VALUES | CLOSE & LOAD | CLOSE & SAVE AS

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



The screenshot shows the Microsoft Power Query Editor interface. In the center, there is a table with 16 columns and 21 rows. A context menu is open over the first four columns, with 'Remove Other Columns' highlighted. The 'QUERY SETTINGS' pane on the right shows the query name as 'Product' and the applied steps as 'Source' and 'Navigation'. The status bar at the bottom indicates '16 COLUMNS, 21 ROWS'.

16 COLUMNS, 21 ROWS

PREVIEW DOWNLOADED AT 00:00

Product

Remove Other Columns

1.0 lbs

2.0 lbs

3.0 lbs

4.0 lbs

5.0 lbs

6.0 lbs

7.0 lbs

8.0 lbs

9.0 lbs

10.0 lbs

11.0 lbs

12.0 lbs

13.0 lbs

14.0 lbs

15.0 lbs

16.0 lbs

17.0 lbs

18.0 lbs

19.0 lbs

20.0 lbs

21.0 lbs

22.0 lbs

23.0 lbs

24.0 lbs

25.0 lbs

26.0 lbs

27.0 lbs

28.0 lbs

29.0 lbs

30.0 lbs

31.0 lbs

32.0 lbs

33.0 lbs

34.0 lbs

35.0 lbs

36.0 lbs

37.0 lbs

38.0 lbs

39.0 lbs

40.0 lbs

41.0 lbs

42.0 lbs

43.0 lbs

44.0 lbs

45.0 lbs

46.0 lbs

47.0 lbs

48.0 lbs

49.0 lbs

50.0 lbs

51.0 lbs

52.0 lbs

53.0 lbs

54.0 lbs

55.0 lbs

56.0 lbs

57.0 lbs

58.0 lbs

59.0 lbs

60.0 lbs

61.0 lbs

62.0 lbs

63.0 lbs

64.0 lbs

65.0 lbs

66.0 lbs

67.0 lbs

68.0 lbs

69.0 lbs

70.0 lbs

71.0 lbs

72.0 lbs

73.0 lbs

74.0 lbs

75.0 lbs

76.0 lbs

77.0 lbs

78.0 lbs

79.0 lbs

80.0 lbs

81.0 lbs

82.0 lbs

83.0 lbs

84.0 lbs

85.0 lbs

86.0 lbs

87.0 lbs

88.0 lbs

89.0 lbs

90.0 lbs

91.0 lbs

92.0 lbs

93.0 lbs

94.0 lbs

95.0 lbs

96.0 lbs

97.0 lbs

98.0 lbs

99.0 lbs

100.0 lbs

101.0 lbs

102.0 lbs

103.0 lbs

104.0 lbs

105.0 lbs

106.0 lbs

107.0 lbs

108.0 lbs

109.0 lbs

110.0 lbs

111.0 lbs

112.0 lbs

113.0 lbs

114.0 lbs

115.0 lbs

116.0 lbs

117.0 lbs

118.0 lbs

119.0 lbs

120.0 lbs

121.0 lbs

122.0 lbs

123.0 lbs

124.0 lbs

125.0 lbs

126.0 lbs

127.0 lbs

128.0 lbs

129.0 lbs

130.0 lbs

131.0 lbs

132.0 lbs

133.0 lbs

134.0 lbs

135.0 lbs

136.0 lbs

137.0 lbs

138.0 lbs

139.0 lbs

140.0 lbs

141.0 lbs

142.0 lbs

143.0 lbs

144.0 lbs

145.0 lbs

146.0 lbs

147.0 lbs

148.0 lbs

149.0 lbs

150.0 lbs

151.0 lbs

152.0 lbs

153.0 lbs

154.0 lbs

155.0 lbs

156.0 lbs

157.0 lbs

158.0 lbs

159.0 lbs

160.0 lbs

161.0 lbs

162.0 lbs

163.0 lbs

164.0 lbs

165.0 lbs

166.0 lbs

167.0 lbs

168.0 lbs

169.0 lbs

170.0 lbs

171.0 lbs

172.0 lbs

173.0 lbs

174.0 lbs

175.0 lbs

176.0 lbs

177.0 lbs

178.0 lbs

179.0 lbs

180.0 lbs

181.0 lbs

182.0 lbs

183.0 lbs

184.0 lbs

185.0 lbs

186.0 lbs

187.0 lbs

188.0 lbs

189.0 lbs

190.0 lbs

191.0 lbs

192.0 lbs

193.0 lbs

194.0 lbs

195.0 lbs

196.0 lbs

197.0 lbs

198.0 lbs

199.0 lbs

200.0 lbs

201.0 lbs

202.0 lbs

203.0 lbs

204.0 lbs

205.0 lbs

206.0 lbs

207.0 lbs

208.0 lbs

209.0 lbs

210.0 lbs

211.0 lbs

212.0 lbs

213.0 lbs

214.0 lbs

215.0 lbs

216.0 lbs

217.0 lbs

218.0 lbs

219.0 lbs

220.0 lbs

221.0 lbs

222.0 lbs

223.0 lbs

224.0 lbs

225.0 lbs

226.0 lbs

227.0 lbs

228.0 lbs

229.0 lbs

230.0 lbs

231.0 lbs

232.0 lbs

233.0 lbs

234.0 lbs

235.0 lbs

236.0 lbs

237.0 lbs

238.0 lbs

239.0 lbs

240.0 lbs

241.0 lbs

242.0 lbs

243.0 lbs

244.0 lbs

245.0 lbs

246.0 lbs

247.0 lbs

248.0 lbs

249.0 lbs

250.0 lbs

251.0 lbs

252.0 lbs

253.0 lbs

254.0 lbs

255.0 lbs

256.0 lbs

257.0 lbs

258.0 lbs

259.0 lbs

260.0 lbs

261.0 lbs

262.0 lbs

263.0 lbs

264.0 lbs

265.0 lbs

266.0 lbs

267.0 lbs

268.0 lbs

269.0 lbs

270.0 lbs

271.0 lbs

272.0 lbs

273.0 lbs

274.0 lbs

275.0 lbs

276.0 lbs

277.0 lbs

278.0 lbs

279.0 lbs

280.0 lbs

281.0 lbs

282.0 lbs

283.0 lbs

284.0 lbs

285.0 lbs

286.0 lbs

287.0 lbs

288.0 lbs

289.0 lbs

290.0 lbs

291.0 lbs

292.0 lbs

293.0 lbs

294.0 lbs

295.0 lbs

296.0 lbs

297.0 lbs

298.0 lbs

299.0 lbs

300.0 lbs

301.0 lbs

302.0 lbs

303.0 lbs

304.0 lbs

305.0 lbs

306.0 lbs

307.0 lbs

308.0 lbs

309.0 lbs

310.0 lbs

311.0 lbs

312.0 lbs

313.0 lbs

314.0 lbs

315.0 lbs

316.0 lbs

317.0 lbs

318.0 lbs

319.0 lbs

320.0 lbs

321.0 lbs

322.0 lbs

323.0 lbs

324.0 lbs

325.0 lbs

326.0 lbs

327.0 lbs

328.0 lbs

329.0 lbs

330.0 lbs

331.0 lbs

332.0 lbs

333.0 lbs

334.0 lbs

335.0 lbs

336.0 lbs

337.0 lbs

338.0 lbs

339.0 lbs

340.0 lbs

341.0 lbs

342.0 lbs

343.0 lbs

344.0 lbs

345.0 lbs

346.0 lbs

347.0 lbs

348.0 lbs

349.0 lbs

350.0 lbs

351.0 lbs

352.0 lbs

353.0 lbs

354.0 lbs

355.0 lbs

356.0 lbs

357.0 lbs

358.0 lbs

359.0 lbs

360.0 lbs

361.0 lbs

362.0 lbs

363.0 lbs

364.0 lbs

365.0 lbs

366.0 lbs

367.0 lbs

368.0 lbs

369.0 lbs

370.0 lbs

371.0 lbs

372.0 lbs

373.0 lbs

374.0 lbs

375.0 lbs

376.0 lbs

377.0 lbs

378.0 lbs

379.0 lbs

380.0 lbs

381.0 lbs

382.0 lbs

383.0 lbs

384.0 lbs

385.0 lbs

386.0 lbs

387.0 lbs

388.0 lbs

389.0 lbs

390.0 lbs

391.0 lbs

392.0 lbs

393.0 lbs

394.0 lbs

395.0 lbs

396.0 lbs

397.0 lbs

398.0 lbs

399.0 lbs

400.0 lbs

401.0 lbs

402.0 lbs

403.0 lbs

404.0 lbs

405.0 lbs

406.0 lbs

407.0 lbs

408.0 lbs

409.0 lbs

410.0 lbs

411.0 lbs

412.0 lbs

413.0 lbs

414.0 lbs

415.0 lbs

416.0 lbs

417.0 lbs

418.0 lbs

419.0 lbs

420.0 lbs

421.0 lbs

422.0 lbs

423.0 lbs

424.0 lbs

425.0 lbs

426.0 lbs

427.0 lbs

428.0 lbs

429.0 lbs

430.0 lbs

431.0 lbs

432.0 lbs

433.0 lbs

434.0 lbs

435.0 lbs

436.0 lbs

437.0 lbs

438.0 lbs

439.0 lbs

440.0 lbs

441.0 lbs

442.0 lbs

443.0 lbs

444.0 lbs

445.0 lbs

446.0 lbs

447.0 lbs

448.0 lbs

449.0 lbs

450.0 lbs

451.0 lbs

452.0 lbs

453.0 lbs

454.0 lbs

455.0 lbs

456.0 lbs

457.0 lbs

458.0 lbs

459.0 lbs

460.0 lbs

461.0 lbs

462.0 lbs

463.0 lbs

464.0 lbs

465.0 lbs

466.0 lbs

467.0 lbs

468.0 lbs

469.0 lbs

470.0 lbs

471.0 lbs

472.0 lbs

473.0 lbs

474.0 lbs

475.0 lbs

476.0 lbs

477.0 lbs

478.0 lbs

479.0 lbs

480.0 lbs

481.0 lbs

482.0 lbs

483.0 lbs

484.0 lbs

485.0 lbs

486.0 lbs

487.0 lbs

488.0 lbs

489.0 lbs

490.0 lbs

491.0 lbs

492.0 lbs

493.0 lbs

494.0 lbs

495.0 lbs

496.0 lbs

497.0 lbs

498.0 lbs

499.0 lbs

500.0 lbs

501.0 lbs

502.0 lbs

503.0 lbs

504.0 lbs

505.0 lbs

506.0 lbs

507.0 lbs

508.0 lbs

509.0 lbs

510.0 lbs

511.0 lbs

512.0 lbs

513.0 lbs

514.0 lbs

515.0 lbs

516.0 lbs

517.0 lbs

518.0 lbs

519.0 lbs

520.0 lbs

521.0 lbs

522.0 lbs

523.0 lbs

524.0 lbs

525.0 lbs

526.0 lbs

527.0 lbs

528.0 lbs

529.0 lbs

530.0 lbs

531.0 lbs

532.0 lbs

533.0 lbs

534.0 lbs

535.0 lbs

536.0 lbs

537.0 lbs

538.0 lbs

539.0 lbs

540.0 lbs

541.0 lbs

542.0 lbs

543.0 lbs

544.0 lbs

545.0 lbs

546.0 lbs

547.0 lbs

548.0 lbs

549.0 lbs

550.0 lbs

551.0 lbs

552.0 lbs

553.0 lbs

554.0 lbs

555.0 lbs

556.0 lbs

557.0 lbs

558.0 lbs

559.0 lbs

560.0 lbs

561.0 lbs

562.0 lbs

563.0 lbs

564.0 lbs

565.0 lbs

566.0 lbs

567.0 lbs

568.0 lbs

569.0 lbs

570.0 lbs

571.0 lbs

572.0 lbs

573.0 lbs

574.0 lbs

575.0 lbs

576.0 lbs

577.0 lbs

578.0 lbs

579.0 lbs

580.0 lbs

581.0 lbs

582.0 lbs

583.0 lbs

584.0 lbs

585.0 lbs

586.0 lbs

587.0 lbs

588.0 lbs

589.0 lbs

590.0 lbs

591.0 lbs

592.0 lbs

593.0 lbs

594.0 lbs

595.0 lbs

596.0 lbs

597.0 lbs

598.0 lbs

599.0 lbs

600.0 lbs

601.0 lbs

602.0 lbs

603.0 lbs

604.0 lbs

605.0 lbs

606.0 lbs

607.0 lbs

608.0 lbs

609.0 lbs

610.0 lbs

611.0 lbs

612.0 lbs

613.0 lbs

614.0 lbs

615.0 lbs

616.0 lbs

617.0 lbs

618.0 lbs

619.0 lbs

620.0 lbs

621.0 lbs

622.0 lbs

623.0 lbs

624.0 lbs

625.0 lbs

626.0 lbs

627.0 lbs

628.0 lbs

629.0 lbs

630.0 lbs

631.0 lbs

632.0 lbs

633.0 lbs

634.0 lbs

635.0 lbs

636.0 lbs

637.0 lbs

638.0 lbs

639.0 lbs

640.0 lbs

641.0 lbs

642.0 lbs

643.0 lbs

644.0 lbs

645.0 lbs

646.0 lbs

647.0 lbs

648.0 lbs

649.0 lbs

650.0 lbs

651.0 lbs

652.0 lbs

653.0 lbs

654.0 lbs

655.0 lbs

656.0 lbs

657.0 lbs

658.0 lbs

659.0 lbs

660.0 lbs

661.0 lbs

662.0 lbs

663.0 lbs

664.0 lbs

665.0 lbs

666.0 lbs

667.0 lbs

668.0 lbs

669.0 lbs

670.0 lbs

671.0 lbs

672.0 lbs

673.0 lbs

674.0 lbs

675.0 lbs

676.0 lbs

677.0 lbs

678.0 lbs

679.0 lbs

680.0 lbs

681.0 lbs

682.0 lbs

683.0 lbs

684.0 lbs

685.0 lbs

686.0 lbs

687.0 lbs

688.0 lbs

<



2. Change the data type of the UnitsInStock column

- a) Select the UnitsInStock column.

The screenshot shows the Microsoft Power Query Editor interface. A red arrow points to the 'Data Type' dropdown button in the ribbon, which is set to 'Decimal Number'. A yellow box highlights this button with the text 'Check if the data type of selected column is a Whole number'. The 'UnitsInStock' column is selected in the table preview area. The 'QUERY SETTINGS' pane on the right shows the 'Name' as 'Products' and the 'Applied Steps' section with three steps: 'Source', 'Navigation', and 'Add Selected Columns'.

- 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).



The screenshot shows the Power Query Editor interface. On the right, a dropdown menu for 'Data Type' is open, with 'Whole Number' selected. The main area displays a table with columns: 'ProductID', 'ProductName', and 'UnitsInStock'. The 'UnitsInStock' column contains numerical values like 10, 20, 30, etc. A preview window at the bottom right shows the data with the 'UnitsInStock' column now containing whole numbers.

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

This screenshot shows the same Power Query Editor interface as the previous one, but with a yellow highlight around the 'UnitsInStock' column header in the table. The 'Data Type' dropdown is still open on the right, showing 'Whole Number' as the selected option. The preview window at the bottom right shows the data with the 'UnitsInStock' column now containing whole numbers.

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.

The screenshot shows the Microsoft Power BI desktop application. At the top, there is a header bar with the college logo and the text "SHRI G.P.M. DEGREE COLLEGE" and "Department of Computer". Below the header is a ribbon menu with tabs like Home, Visualizations, Modeling, and Data. The main workspace is a grid-based canvas where data visualizations can be built. On the left side, there is a "Recent Sources" pane containing a list of recent data sources, with "ORDERS" highlighted. To the right of the canvas is a "Visualizations" pane, which contains various chart and report icons, and a "Fields" pane listing fields such as Product, ProductID, ProductName, QuantityPerUnit, and UnitPrice. A yellow arrow points from the text "Once You have loaded a data source, you can click on Recent Sources to select desired table (Orders)." to the "ORDERS" entry in the recent sources list.

3. Expand the Orders table

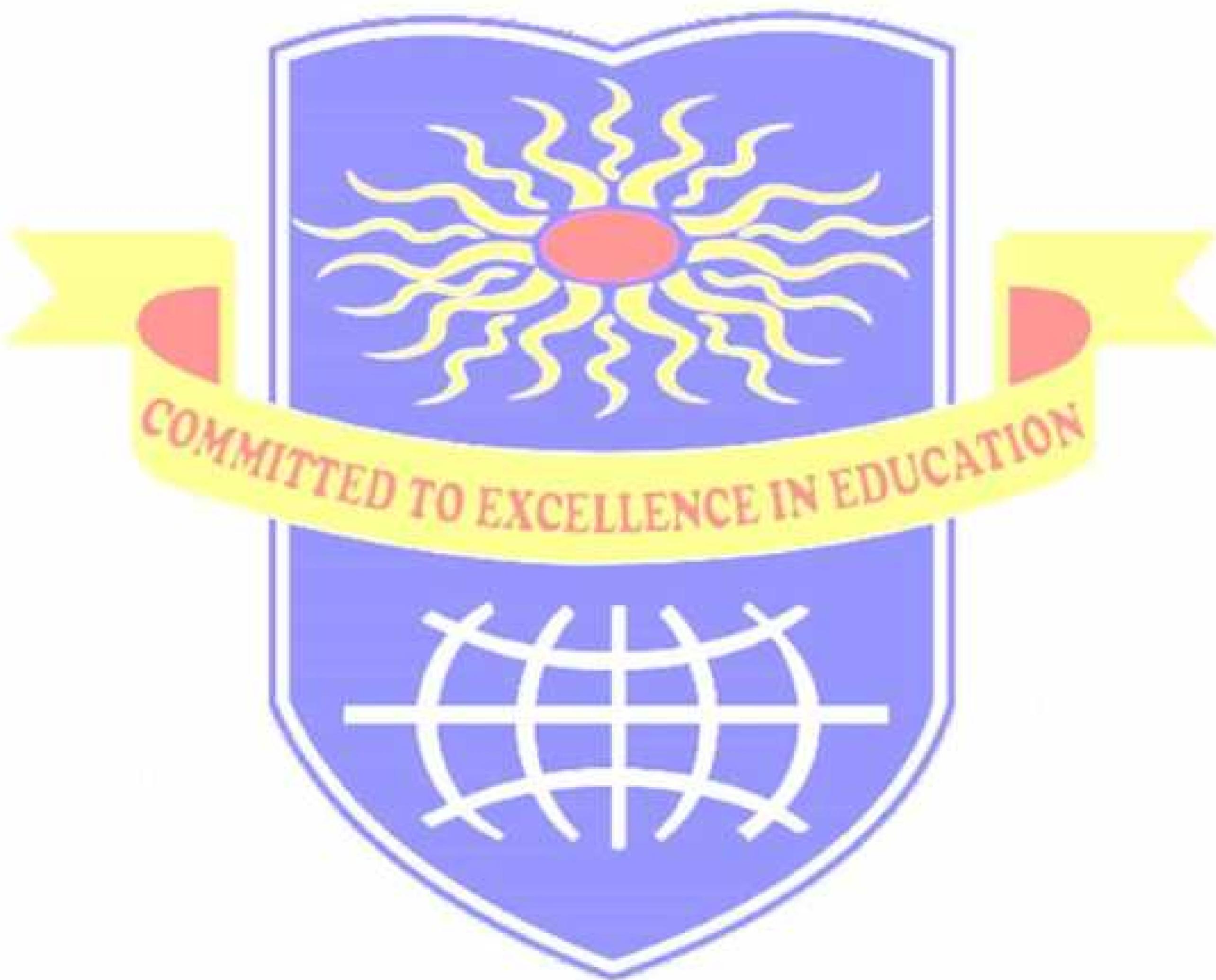
Once You have loaded a data source, you can click on Recent Sources to select desired table (Orders).

This screenshot is identical to the one above, showing the Microsoft Power BI desktop application. It displays the "ORDERS" table in the "Recent Sources" pane. The "Visualizations" and "Fields" panes are also visible on the right side of the interface. A yellow arrow points from the text "Once You have loaded a data source, you can click on Recent Sources to select desired table (Orders)." to the "ORDERS" entry in the recent sources list.



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

Click on Edit.



नियमित नोह उत्तम सेवाधर्म



Navigator

Display Options: [http://services.odata.org/V3/Northwind/Northwind.svc](#)

- [Alphabetical_list_of_products](#)
- [Categories](#)
- [Category_Sales_for_1997](#)
- [Current_Product_Lists](#)
- [Customer_and_Supplier_by_Cities](#)
- [CustomerDemographics](#)
- [Customers](#)
- [Employees](#)
- [Invoices](#)
- [Order_Details](#)
- [Order_Details_Extended](#)
- [Order_Subtotals](#)
- [Orders](#)
- [Orders_Cries](#)
- [Product_Sales_for_1997](#)
- [Products](#)
- [Products_Above_Average_Prices](#)
- [Products_by_Categories](#)
- [Regions](#)

Orders

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

Select Related Tables

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.



The screenshot shows the Power Query Editor interface. A dialog box titled 'Combine Queries' is open, indicating the process of merging two queries. The 'Order_Details' query is designated as the target. The 'Add Column' ribbon tab is currently selected. In the dialog, three columns from the source query ('ProductID', 'UnitPrice', and 'Quantity') are chosen for concatenation into a new column. The 'OK' button is highlighted, suggesting the next step is to confirm the combination.

After clicking on OK following screen appears with combined columns

The screenshot shows the Power BI Desktop interface after the 'OK' button was clicked. The 'Order' query now includes a new column named 'Order_Details', which contains the concatenated values from the 'ProductID', 'UnitPrice', and 'Quantity' columns of the 'Order_Details' query. The main data grid displays the combined information, and the 'OK' button is no longer visible, confirming the successful merge.

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:

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

Untitled - Power Query Editor

File Home Refresh Add Column View Help

Column Tools Columns Inv. Curbs Functions

Conditional Column Merge Columns 100 Properties +

Format Address Standard Headers 100+ Headers +

Date Time Duration

General From Text From Number From Date & Time

Queries (2)

Products

Chains

My Chain Product

1.1 Other Inventory

1.2 Other Invent. Curbs

1.3 Headers

1.4

1.5

1.6

1.7

1.8

1.9

1.10

1.11

1.12

1.13

1.14

1.15

1.16

1.17

1.18

1.19

1.20

1.21

1.22

1.23

1.24

1.25

1.26

1.27

1.28

1.29

1.30

1.31

1.32

1.33

1.34

1.35

1.36

1.37

1.38

1.39

1.40

1.41

1.42

1.43

1.44

1.45

1.46

1.47

1.48

1.49

1.50

1.51

1.52

1.53

1.54

1.55

1.56

1.57

1.58

1.59

1.60

1.61

1.62

1.63

1.64

1.65

1.66

1.67

1.68

1.69

1.70

1.71

1.72

1.73

1.74

1.75

1.76

1.77

1.78

1.79

1.80

1.81

1.82

1.83

1.84

1.85

1.86

1.87

1.88

1.89

1.90

1.91

1.92

1.93

1.94

1.95

1.96

1.97

1.98

1.99

1.100

1.101

1.102

1.103

1.104

1.105

1.106

1.107

1.108

1.109

1.110

1.111

1.112

1.113

1.114

1.115

1.116

1.117

1.118

1.119

1.120

1.121

1.122

1.123

1.124

1.125

1.126

1.127

1.128

1.129

1.130

1.131

1.132

1.133

1.134

1.135

1.136

1.137

1.138

1.139

1.140

1.141

1.142

1.143

1.144

1.145

1.146

1.147

1.148

1.149

1.150

1.151

1.152

1.153

1.154

1.155

1.156

1.157

1.158

1.159

1.160

1.161

1.162

1.163

1.164

1.165

1.166

1.167

1.168

1.169

1.170

1.171

1.172

1.173

1.174

1.175

1.176

1.177

1.178

1.179

1.180

1.181

1.182

1.183

1.184

1.185

1.186

1.187

1.188

1.189

1.190

1.191

1.192

1.193

1.194

1.195

1.196

1.197

1.198

1.199

1.200

1.201

1.202

1.203

1.204

1.205

1.206

1.207

1.208

1.209

1.210

1.211

1.212

1.213

1.214

1.215

1.216

1.217

1.218

1.219

1.220

1.221

1.222

1.223

1.224

1.225

1.226

1.227

1.228

1.229

1.230

1.231

1.232

1.233

1.234

1.235

1.236

1.237

1.238

1.239

1.240

1.241

1.242

1.243

1.244

1.245

1.246

1.247

1.248

1.249

1.250

1.251

1.252

1.253

1.254

1.255

1.256

1.257

1.258

1.259

1.260

1.261

1.262

1.263

1.264

1.265

1.266

1.267

1.268

1.269

1.270

1.271

1.272

1.273

1.274

1.275

1.276

1.277

1.278

1.279

1.280

1.281

1.282

1.283

1.284

1.285

1.286

1.287

1.288

1.289

1.290

1.291

1.292

1.293

1.294

1.295

1.296

1.297

1.298

1.299

1.2000

1.2001

1.2002

1.2003

1.2004

1.2005

1.2006

1.2007

1.2008

1.2009

1.2010

1.2011

1.2012

1.2013

1.2014

1.2015

1.2016

1.2017

1.2018

1.2019

1.2020

1.2021

1.2022

1.2023

1.2024

1.2025

1.2026

1.2027

1.2028

1.2029

1.2030

1.2031

1.2032

1.2033

1.2034

1.2035

1.2036

1.2037

1.2038

1.2039

1.2040

1.2041

1.2042

1.2043

1.2044

1.2045

1.2046

1.2047

1.2048

1.2049

1.2050

1.2051

1.2052

1.2053

1.2054

1.2055

1.2056

1.2057

1.2058

1.2059

1.2060

1.2061

1.2062

1.2063

1.2064

1.2065

1.2066

1.2067

1.2068

1.2069

1.2070

1.2071

1.2072

1.2073

1.2074

1.2075

1.2076

1.2077

1.2078

1.2079

1.2080

1.2081

1.2082

1.2083

1.2084

1.2085

1.2086

1.2087

1.2088

1.2089

1.2090

1.2091

1.2092

1.2093

1.2094

1.2095

1.2096

1.2097

1.2098

1.2099

1.2100

1.2101

1.2102

1.2103

1.2104

1.2105

1.2106

1.2107

1.2108

1.2109

1.2110

1.2111

1.2112

1.2113

1.2114

1.2115

1.2116

1.2117

1.2118

1.2119

1.2120

1.2121

1.2122

1.2123

1.2124

1.2125

1.2126

1.2127

1.2128

1.2129

1.2130

1.2131

1.2132

1.2133

1.2134

1.2135

1.2136

1.2137

1.2138

1.2139

1.2140

1.2141

1.2142

1.2143

1.2144

1.2145

1.2146

1.2147

1.2148

1.2149

1.2150

1.2151

1.2152

1.2153

1.2154

1.2155

1.2156

1.2157

1.2158

1.2159

1.2160

1.2161

1.2162

1.2163

1.2164

1.2165

1.2166

1.2167

1.2168

1.2169

1.2170

1.2171

1.2172

1.2173

1.2174

1.2175

1.2176

1.2177

1.2178

1.2179

1.2180

1.2181

1.2182

1.2183

1.2184

1.2185

1.2186

1.2187

1.2188

1.2189

1.2190

1.2191

1.2192

1.2193

1.2194

1.2195

1.2196

1.2197

1.2198

1.2199

1.2200

1.2201

1.2202

1.2203

1.2204

1.2205

1.2206

1.2207

1.2208

1.2209

1.2210

1.2211

1.2212

1.2213

1.2214

1.2215

1.2216

1.2217

1.2218

1.2219

1.2220

1.2221

1.2222

1.2223

1.2224

1.2225

1.2226

1.2227

1.2228

1.2229

1.2230

1.2231

1.2232

1.2233

1.2234

1.2235

1.2236

1.2237

1.2238

1.2239

1.2240

1.2241

1.2242

1.2243

1.2244

1.2245

1.2246

1.2247

1.2248

1.2249

1.2250

1.2251

1.2252

1.2253

1.2254

1.2255

1.2256

1.2257

1.2258

1.2259

1.2260

1.2261

1.2262

1.2263

1.2264

1.2265

1.2266

1.2267

1.2268

1.2269

1.2270

1.2271

1.2272

1.2273

1.2274

1.2275

1.2276

1.2277

1.2278

1.2279

1.2280

1.2281

1.2282

1.2283

1.2284

1.2285

1.2286

1.2287

1.2288

1.2289

1.2290

1.2291

1.2292

1.2293

1.2294

1.2295

1.2296

1.2297

1.2298

1.2299

1.2300

1.2301

1.2302

1.2303

1.2304

1.2305

1.2306

1.2307

1.2308

1.2309

1.2310

1.2311

1.2312

1.2313

1.2314

1.2315

1.2316

1.2317

1.2318

1.2319

1.2320

1.2321

1.2322

1.2323

1.2324

1.2325

1.2326

1.2327

1.2328

1.2329

1.2330

1.2331

1.2332

1.2333

1.2334

1.2335

1.2336

1.2337

1.2338

1.2339

1.2340

1.2341

1.2342

1.2343

1.2344

1.2345

1.2346

1.2347

1.2348

1.2349

1.2350

1.2351

1.2352

1.2353

1.2354

1.2355

1.2356

1.2357

1.2358

1.2359

1.2360

1.2361

1.2362

1.2363

1.2364

1.2365

1.2366

1.2367

1.2368

1.2369

1.2370

1.2371

1.2372

1.2373

1.2374

1.2375

1.2376

1.2377

1.2378

1.2379

1.2380

1.2381

1.2382

1.2383

1.2384

1.2385

1.2386

1.2387

1.2388

1.2389

1.2390

1.2391

1.2392

1.2393

1.2394

1.2395

1.2396

1.2397

1.2398

1.2399

1.2400

1.2401

1.2402

1.2403

1.2404

1.2405

1.2406

1.2407

1.2408

1.2409

1.2410

1.2411

1.2412

1.2413

1.2414

1.2415

1.2416

1.2417

1.2418

1.2419

1.2420

1.2421

1.2422

1.2423

1.2424

1.2425

1.2426

1.2427

1.2428

1.2429

1.2430

1.2431

1.2432

1.2433

1.2434

1.2435

1.2436

1.2437

1.2438

1.2439

1.2440

1.2441

1.2442

1.2443

1.2444

1.2445

1.2446

1.2447

1.2448

1.2449

1.2450

1.2451

1.2452

1.2453

1.2454

1.2455

1.2456

1.2457

1.2458

1.2459

1.2460

1.2461

1.2462

1.2463

1.2464

1.2465

1.2466

1.2467

1.2468

1.2469

1.2470

1.2471

1.2472

1.2473

1.2474

1.2475

1.2476

1.2477

1.2478

1.2479

1.2480

1.2481

1.2482

1.2483

1.2484

1.2485

1.2486

1.2487

1.2488

1.2489

1.2490

1.2491

1.2492

1.2493

1.2494

1.2495

1.2496

1.2497

1.2498

1.2499

1.2500

1.2501

1.2502

1.2503

1.2504

1.2505

1.2506

1.2507

1.2508

1.2509

1.2510

1.2511

1.2512

1.2513

1.2514

1.2515

1.2516

1.2517

1.2518

1.2519

1.2520

1.2521

1.2522

1.2523

1.2524

1.2525

1.2526

1.2527

1.2528

1.2529

1.2530

1.2531

1.2532

1.2533

1.2534

1.2535

1.2536

1.2537

1.2538

1.2539

1.2540

1.2541

1.2542

1.2543

1.2544

1.2545

1.2546

1.2547

1.2548

1.2549

1.2550

1.2551

1.2552

1.2553

1.2554

1.2555

1.2556

1.2557

1.2558

1.2559

1.2560

1.2561

1.2562

1.2563

1.2564

1.2565

1.2566

1.2567

1.2568

1.2569

1.2570

1.2571

1.2572

1.2573

1.2574

1.2575

1.2576

1.2577

1.2578

1.2579

1.2580

1.2581

1.2582

1.2583

1.2584

1.2585

1.2586

1.2587

1.2588

1.2589

1.2590

1.2591

1.2592

1.2593

1.2594

1.2595

1.2596

1.2597

1.2598

1.2599

1.2600

1.2601

1.2602

1.2603

1.2604

1.2605

1.2606

1.2607

1.2608

1.2609

1.2610

1.2611

1.2612

1.2613

1.2614

1.2615

1.2616

1.2617

1.2618

1.2619

1.2620

1.2621

1.2622

1.2623

1.2624

1.2625

1.2626

1.2627

1.2628

1.2629

1.2630

1.2631

1.2632

1.2633

1.2634

1.2635

1.2636

1.2637

1.2638

1.2639

1.2640

1.2641

1.2642

1.2643

1.2644

1.2645

1.2646

1.2647

1.2648

1.2649

1.2650

1.2651

1.2652

1.2653

1.2654

1.2655

1.2656

1.2657

1.2658

1.2659

1.2660

1.2661

1.2662

1.2663

1.2664

1.2665

1.2666

1.2667

1.2668

1.2669

1.2670

1.2671

1.2672

1.2673

1.2674

1.2675

1.2676

1.2677

1.2678

1.2679

1.2680

1.2681

1.2682

1.2683

1.2684

1.2685

1.2686

1.2687

1.2688

1.2689

1.2690

1.2691

1.2692

1.2693

1.2694

1.2695

1.2696

1.2697

1.2698

1.2699

1.2700

1.2701

1.2702

1.2703

1.2704

1.2705

1.2706

1.2707

1.2708

1.2709

1.2710

1.2711

1.2712

1.2713

1.2714

1.2715

1.2716

1.2717

1.2718

1.2719

1.2720

1.2721

1.2722

1.2723

1.2724

1.2725

1.2726

1.2727

1.2728

1.2729

1.2730

1.2731

1.2732

1.2733

1.2734

1.2735

1.2736

1.2737

1.2738

1.2739

1.2740

1.2741

1.2742

1.2743

1.2744

1.2745

1.2746

1.2747

1.2748

1.2749

1.2750

1.2751

1.2752

1.2753

1.2754

1.2755

1.2756

1.2757

1.2758

1.2759

1.2760

1.2761

1.2762

1.2763

1.2764

1.2765

1.2766

1.2767

1.2768

1.2769

1.2770

1.2771

1.2772

1.2773

1.2774

1.2775

1.2776

1.2777

1.2778

1.2779

1.2780

1.2781

1.2782

1.2783

1.2784

1.2785

1.2786

1.2787

1.2788

1.2789

1.2790

1.2791

1.2792

1.2793

1.2794

1.2795

1.2796

1.2797

1.2798

1.2799

1.2800

1.2801

1.2802

1.2803

1.2804

1.2805

1.2806

1.2807

1.2808

1.2809

1.2810

1.2811

1.2812

1.2813

1.2814

1.2815

1.2816

1.2817

1.2818

1.2819

1.2820

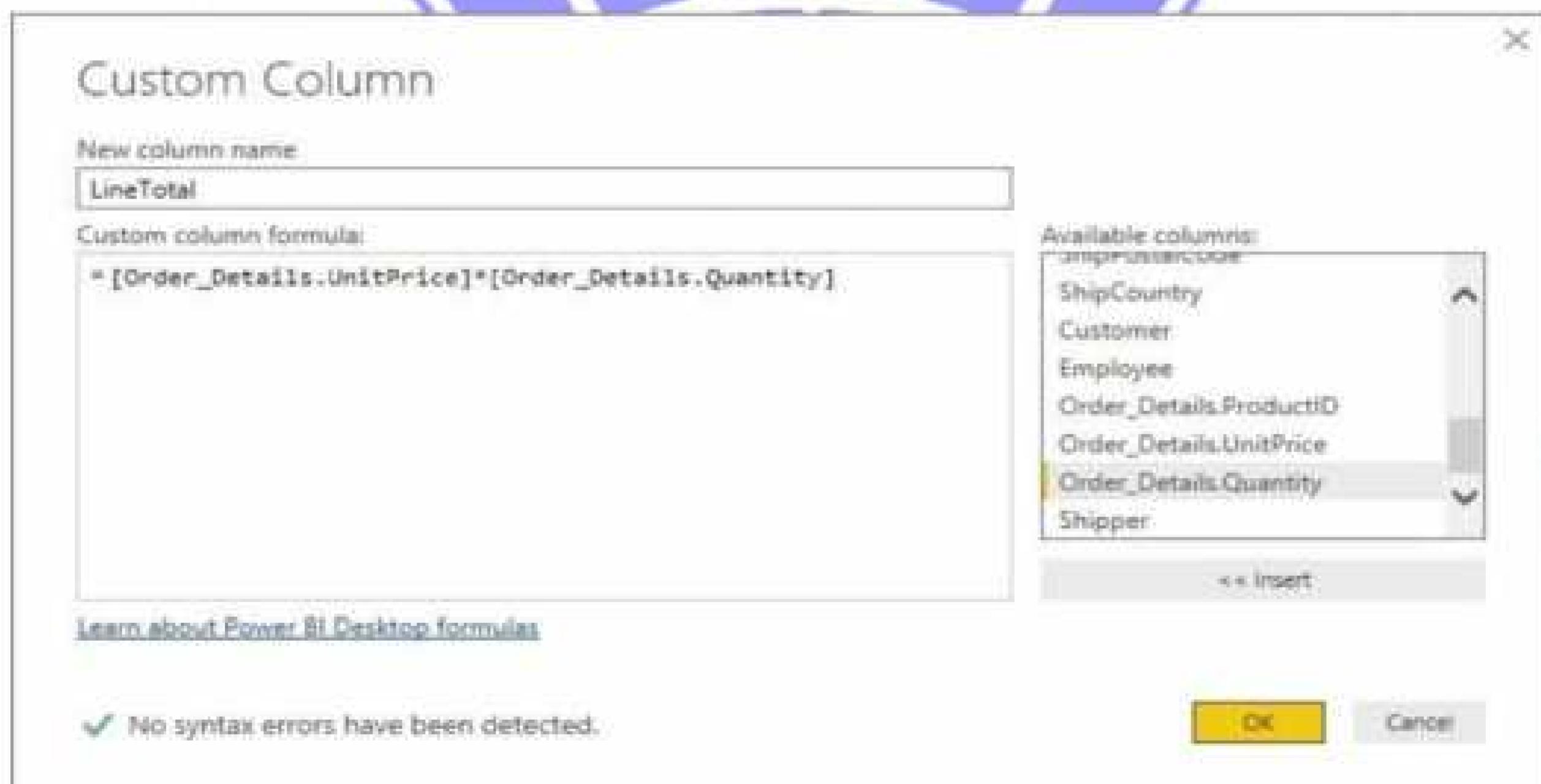
1.2821

1.2822

1.2823

<

- 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.





The screenshot shows the Microsoft Power Query Editor interface. A query named 'Orders' is open. The data table contains three columns: 'LineID', 'ShipCountry', and 'LineTotal'. The 'LineTotal' column is currently selected, indicated by a yellow background. The 'QUERY SETTINGS' pane on the right shows the query name as 'Orders' and the applied steps as 'Added Custom'.

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.

The screenshot shows the Microsoft Power Query Editor interface again. The 'Orders' query is open, and the data table now has four columns: 'LineID', 'ShipCountry', 'ShipName', and 'LineTotal'. The 'LineTotal' column is no longer highlighted. The 'QUERY SETTINGS' pane on the right remains the same.



- 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.

The screenshot shows the Power Query Editor interface with the 'Orders' query selected. The 'ProductID' column header is highlighted, and the 'Order_Details.' prefix is being deleted. The 'QUERY SETTINGS' pane on the right shows the 'Name' set to 'Orders' and the 'Applied Steps' list containing 'Source', 'Imported Order Details', 'Added Custom', 'Renamed Columns', and 'All previous'.

Customer	Region	OrderID	UnitPrice	Quantity	Shipped
Customer 1	Region 1	10	14	12	Never
Customer 2	Region 2	20	16	18	Never
Customer 3	Region 3	30	18.0	9	Never
Customer 4	Region 4	40	13.0	8	Never
Customer 5	Region 5	50	42.0	40	Never
Customer 6	Region 6	60	7.0	20	Never
Customer 7	Region 7	70	40.0	10	Never
Customer 8	Region 8	80	10.0	15	Never
Customer 9	Region 9	90	10.0	8	Never
Customer 10	Region 10	100	13.0	10	Never
Customer 11	Region 11	110	13.0	20	Never
Customer 12	Region 12	120	13.0	20	Never
Customer 13	Region 13	130	13.0	40	Never
Customer 14	Region 14	140	2.0	20	Never
Customer 15	Region 15	150	21.0	40	Never
Customer 16	Region 16	160	10	20	Never

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.



Products

ProductID	ProductName	Unit	UnitsInStock
1	Chai	10	120
2	Sauces	10	100
3	Baked Beans	10	150
4	Marmalade	10	100
5	Pates	10	100
6	Spices	10	100
7	Cheese	10	100
8	Tea	10	100
9	Meat Pies	10	100
10	Quiches	10	100
11	Strudels	10	100
12	Pastry	10	100
13	Brownies	10	100
14	Prunes	10	100
15	Butter	10	100

Sales

OrderID	CustomerID	EmployeeID	OrderDate
1	1	1	2013-01-01
2	2	1	2013-01-01
3	3	1	2013-01-01
4	4	1	2013-01-01
5	5	1	2013-01-01
6	6	1	2013-01-01
7	7	1	2013-01-01
8	8	1	2013-01-01
9	9	1	2013-01-01
10	10	1	2013-01-01
11	11	1	2013-01-01
12	12	1	2013-01-01
13	13	1	2013-01-01
14	14	1	2013-01-01
15	15	1	2013-01-01
16	16	1	2013-01-01
17	17	1	2013-01-01
18	18	1	2013-01-01
19	19	1	2013-01-01
20	20	1	2013-01-01
21	21	1	2013-01-01
22	22	1	2013-01-01
23	23	1	2013-01-01
24	24	1	2013-01-01
25	25	1	2013-01-01
26	26	1	2013-01-01
27	27	1	2013-01-01
28	28	1	2013-01-01
29	29	1	2013-01-01
30	30	1	2013-01-01
31	31	1	2013-01-01
32	32	1	2013-01-01
33	33	1	2013-01-01
34	34	1	2013-01-01
35	35	1	2013-01-01
36	36	1	2013-01-01
37	37	1	2013-01-01
38	38	1	2013-01-01
39	39	1	2013-01-01
40	40	1	2013-01-01
41	41	1	2013-01-01
42	42	1	2013-01-01
43	43	1	2013-01-01
44	44	1	2013-01-01
45	45	1	2013-01-01
46	46	1	2013-01-01
47	47	1	2013-01-01
48	48	1	2013-01-01
49	49	1	2013-01-01
50	50	1	2013-01-01
51	51	1	2013-01-01
52	52	1	2013-01-01
53	53	1	2013-01-01
54	54	1	2013-01-01
55	55	1	2013-01-01
56	56	1	2013-01-01
57	57	1	2013-01-01
58	58	1	2013-01-01
59	59	1	2013-01-01
60	60	1	2013-01-01
61	61	1	2013-01-01
62	62	1	2013-01-01
63	63	1	2013-01-01
64	64	1	2013-01-01
65	65	1	2013-01-01
66	66	1	2013-01-01
67	67	1	2013-01-01
68	68	1	2013-01-01
69	69	1	2013-01-01
70	70	1	2013-01-01
71	71	1	2013-01-01
72	72	1	2013-01-01
73	73	1	2013-01-01
74	74	1	2013-01-01
75	75	1	2013-01-01
76	76	1	2013-01-01
77	77	1	2013-01-01
78	78	1	2013-01-01
79	79	1	2013-01-01
80	80	1	2013-01-01
81	81	1	2013-01-01
82	82	1	2013-01-01
83	83	1	2013-01-01
84	84	1	2013-01-01
85	85	1	2013-01-01
86	86	1	2013-01-01
87	87	1	2013-01-01
88	88	1	2013-01-01
89	89	1	2013-01-01
90	90	1	2013-01-01
91	91	1	2013-01-01
92	92	1	2013-01-01
93	93	1	2013-01-01
94	94	1	2013-01-01
95	95	1	2013-01-01
96	96	1	2013-01-01
97	97	1	2013-01-01
98	98	1	2013-01-01
99	99	1	2013-01-01
100	100	1	2013-01-01

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

Apply query changes

Orders
Loading data to model...

Cancel

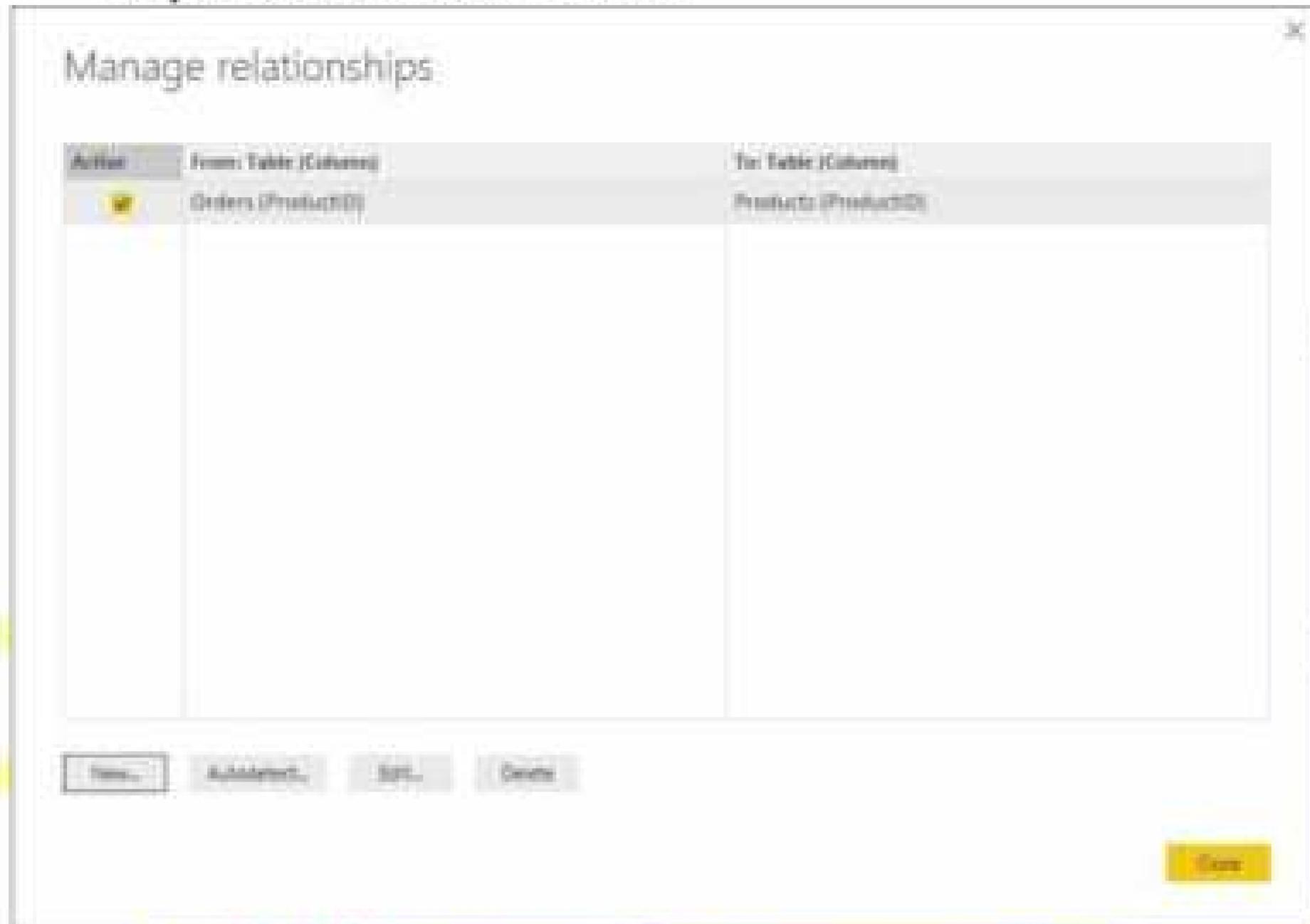
Step 3: Once the data is loaded, select the Manage Relationships button Home ribbon

Manage Relationships

Add, edit, or remove relationships between tables.



Step 4. Select the New... button



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.

निष्ठानोह उत्तम सेवाधर्म



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	Ur
X-Stop	Taucherstraße 10	Cunewalde	null	01307	Germany	595.2	10	
X-Stop	Taucherstraße 10	Cunewalde	null	01307	Germany	150	31	
X-Stop	Taucherstraße 10	Cunewalde	null	01307	Germany	40	33	

Cardinality

One to many (1:n)

Cross filter direction

Single

Make this relationship active

Assume referential integrity

Apply security filters in both directions

There's already a relationship between these two columns.

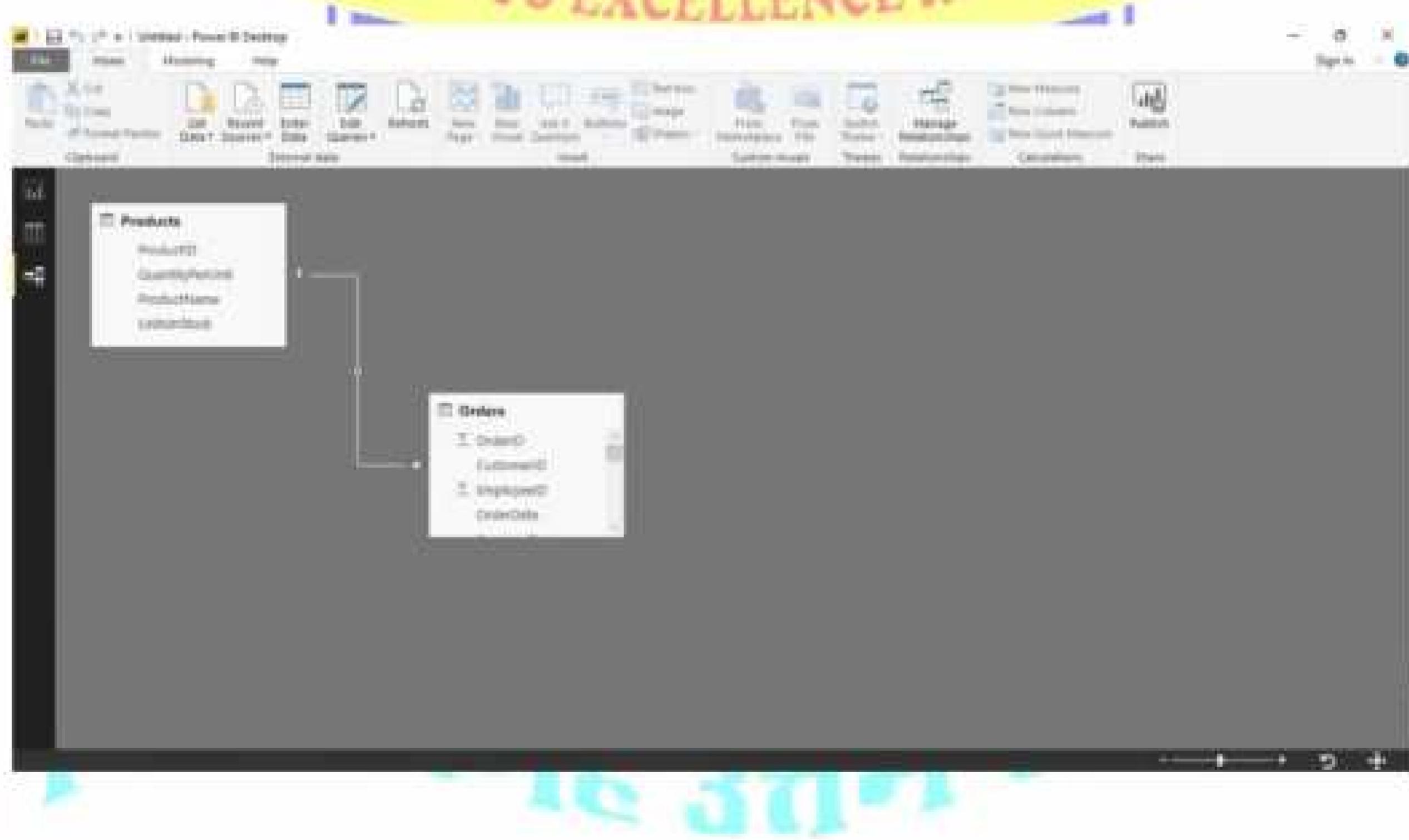
OK Cancel

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

निस्लेन्ह उत्तम सेवाधर्म



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 two queries, an Edit Relationship dialog appears.



Edit relationship

Select tables and columns that are related.

Orders

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	F
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
2	10 boxes x 20 bags	Chai	39
2	24 - 12 oz bottles	Chang	17
3	12 - 550 ml bottles	Aniseed Syrup	15

Cardinality

Many to one (*:1)

Cross filter direction

Single

Make this relationship active

Apply security filter in both directions

Assume referential integrity

OK Cancel

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

निस्लेन्ह उत्तम सेवाधर्म

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

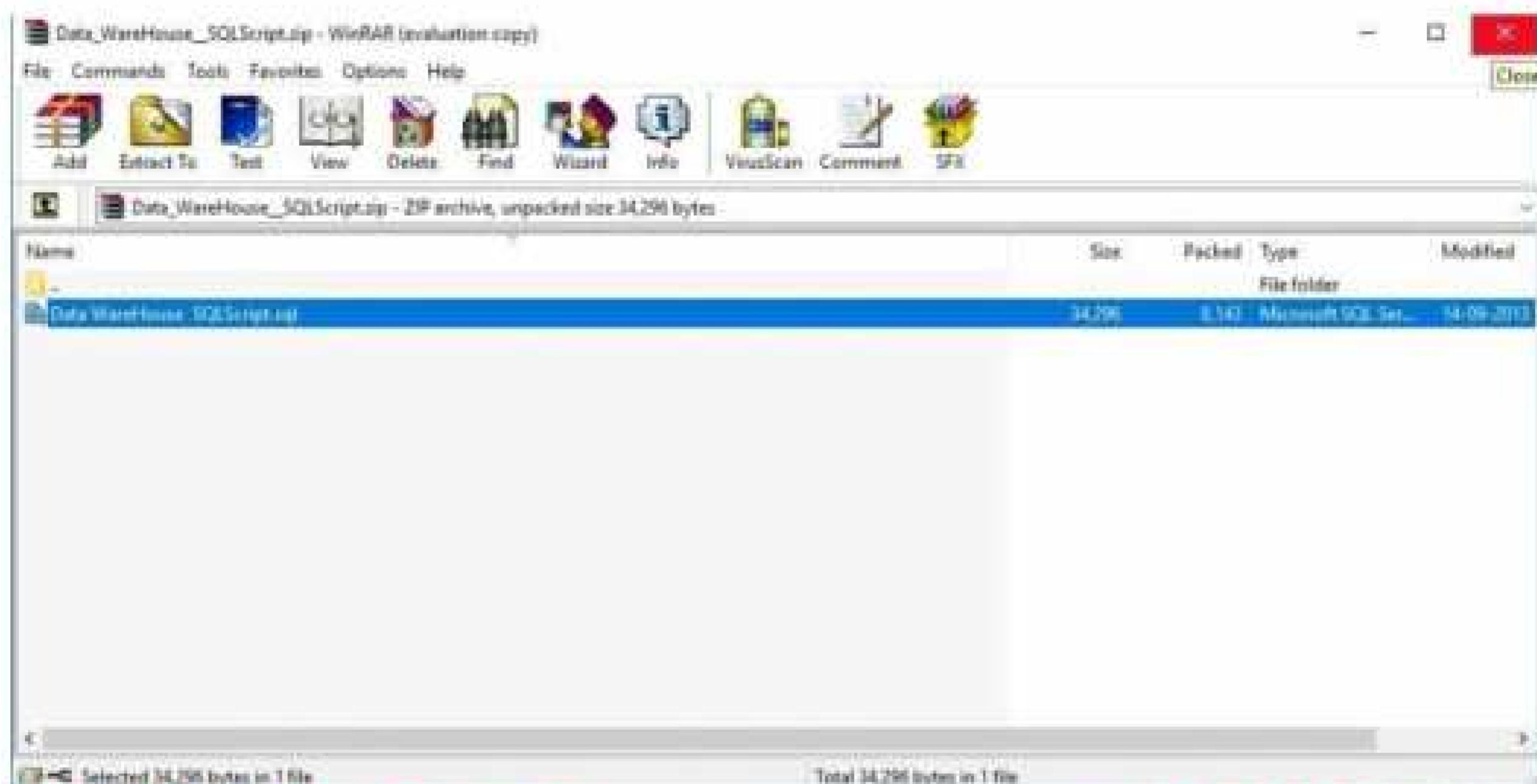
Let us execute our T-SQL Script to create data warehouse with fact tables, dimensions and populate them with appropriate test values.

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/652408/Create-First-Data-Warehouse>



निष्ठानीह उत्तम सेवाधर्म



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**



OMNIBUS
COMMITTED TO EXCELLENCE IN EDUCATION
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.

निषिद्ध नोह उत्तम सेवाधर्म



File Database Taskgroup (Laptop-GPMDEGCOLLEGE) - Microsoft SQL Server Management Studio

File Home Databases Tasks Help

Object Explorer

Connect To... File Edit View Design Tools Window Help

File Databases System Databases Master Model msdb tempdb Security Server Objects Replication AlwaysOn High Availability Management Integration Services Catalogs SQL Server Agent

Tasks

File Databases Laptops (1).mdf (16) N

```
--Create customer identifier table in Sales database which will hold customer personal details.
--Create table Customer
CustomerID int primary key identity;
CustomerName varchar(50) not null;
CustomerAddress varchar(50);
MobileNumber varchar(20);
Email varchar(50);

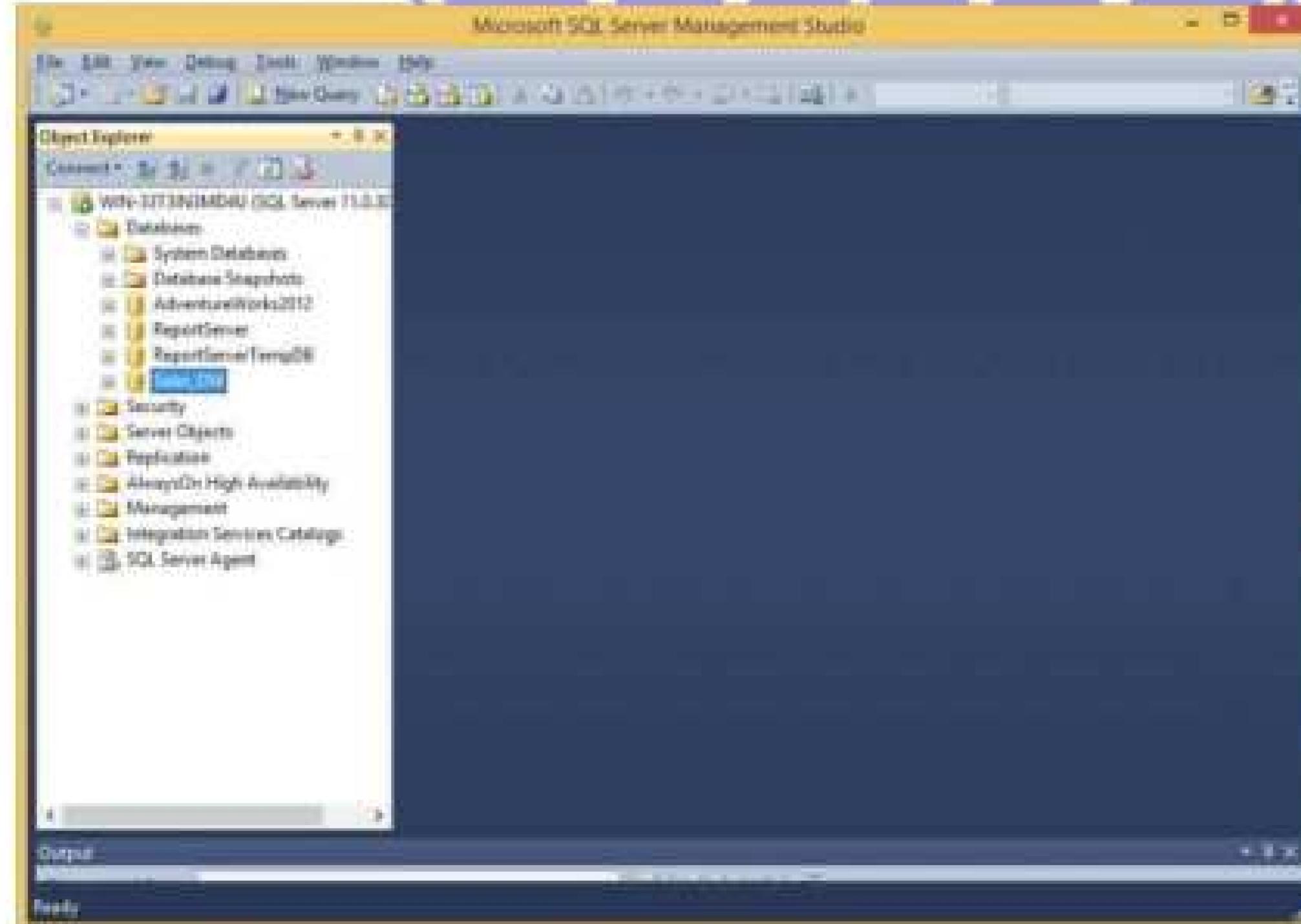
--Insert the Customer information with sample values
Insert into Customer (CustomerID, CustomerName, MobileNumber)
values ('100-001', 'Merry Puff', '987');
('100-002', 'Sally Sazer', '976');
('100-003', 'Raman Shetty', '965');
('100-004', 'Richard Shutter', '954');
('100-005', 'Tina Perron', '943');

--Create basic table of Product description table without considering any Category or SubCategory
--Create table Product
```

File Home Databases Tasks Help

File Databases Laptops (1).mdf (16) N

- 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.



ପାଧ୍ୟ



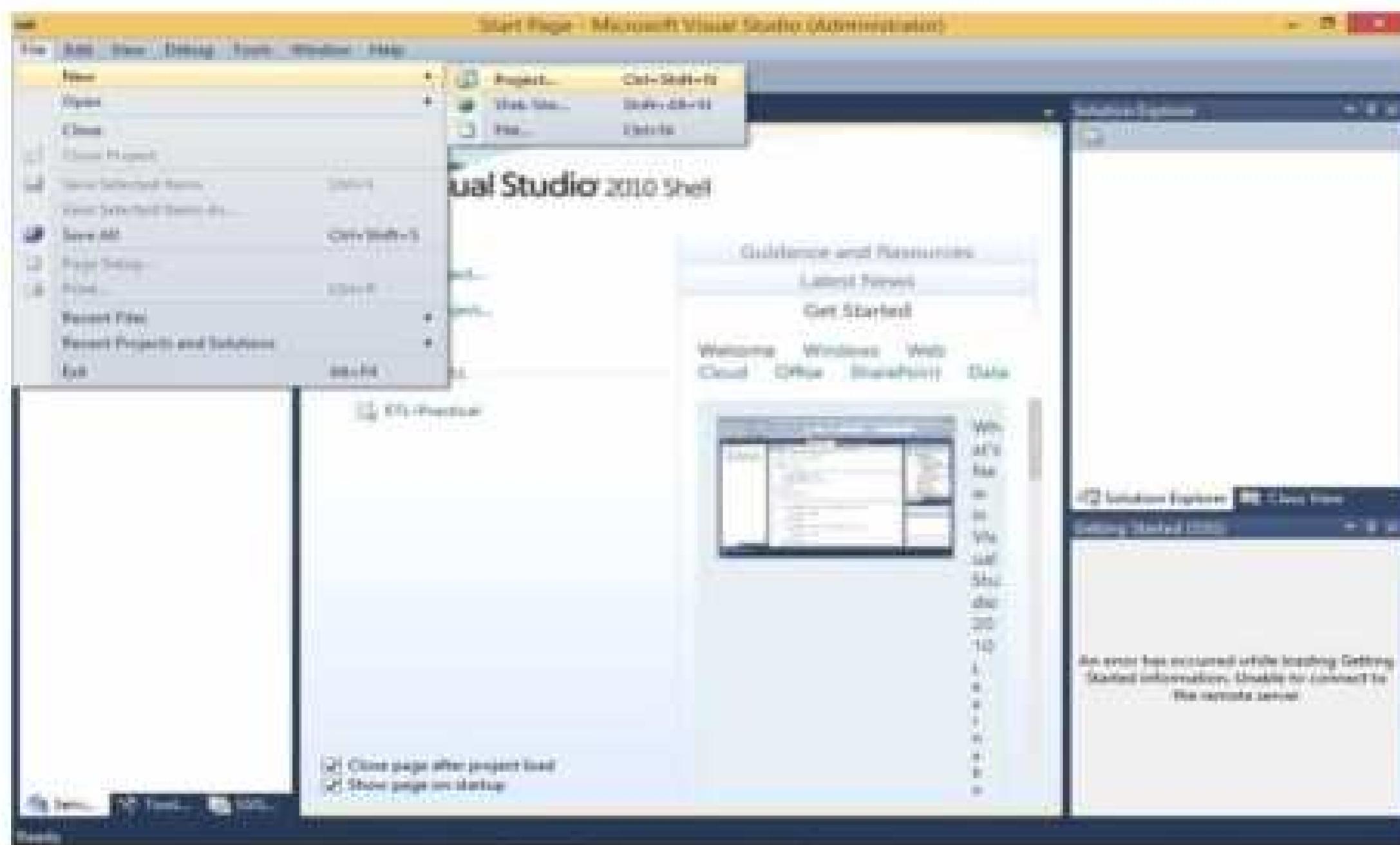
Step 2: Start SSDT environment and create New Data Source

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



Click on File → New → Project

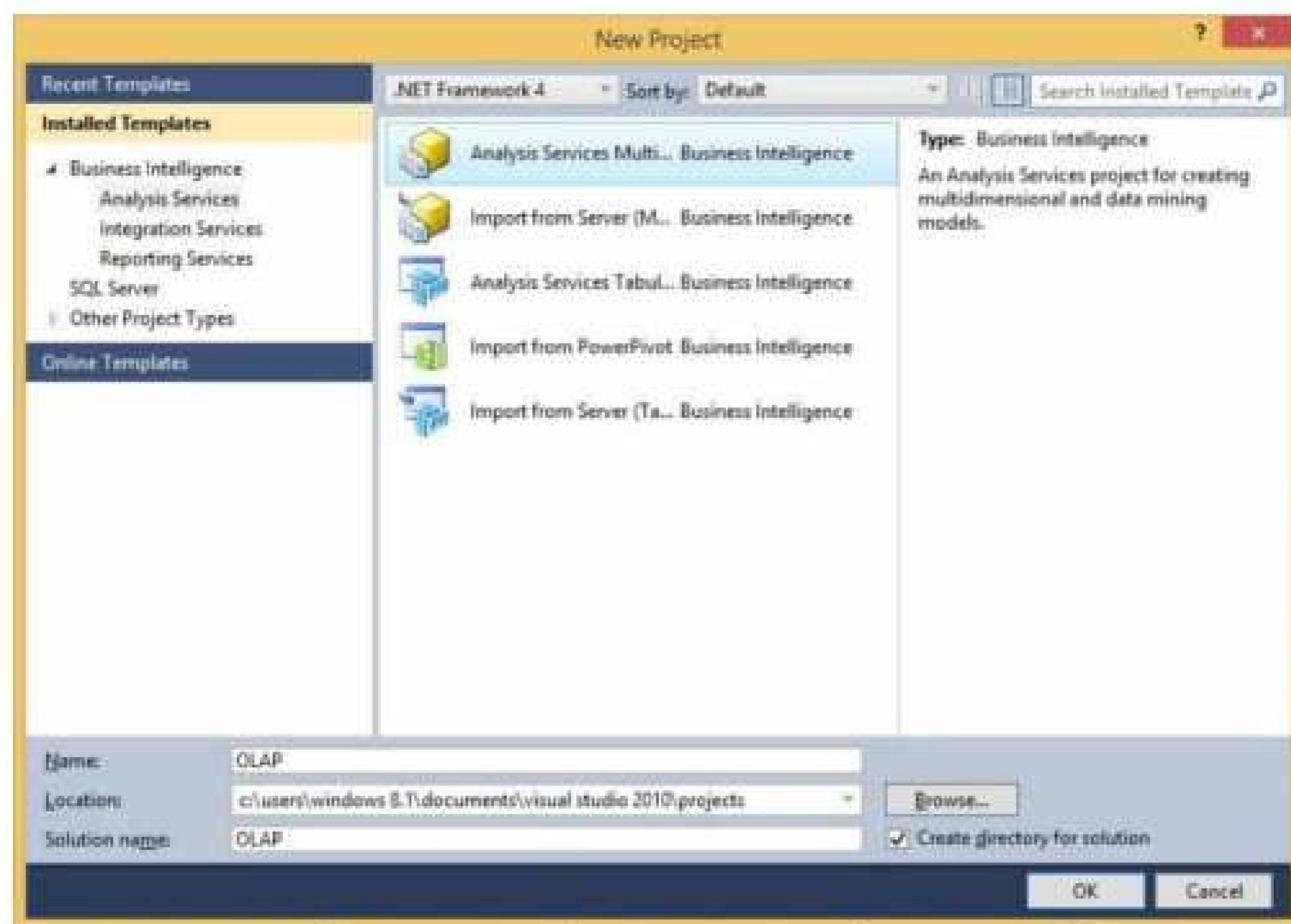
नियंत्रणों उत्तम सेवाधर्म



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

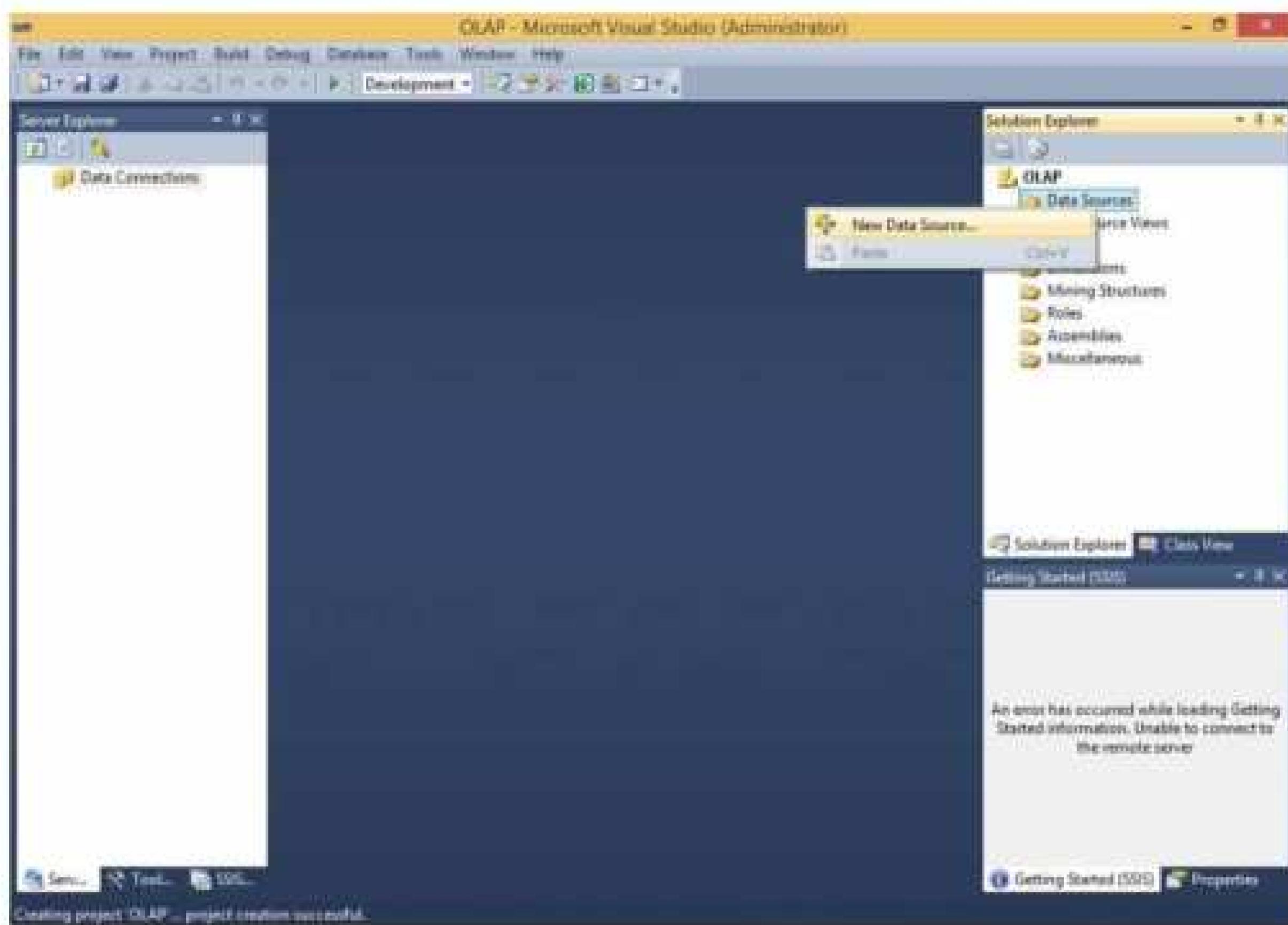


निष्ठानोह उत्तम सेवाधर्म

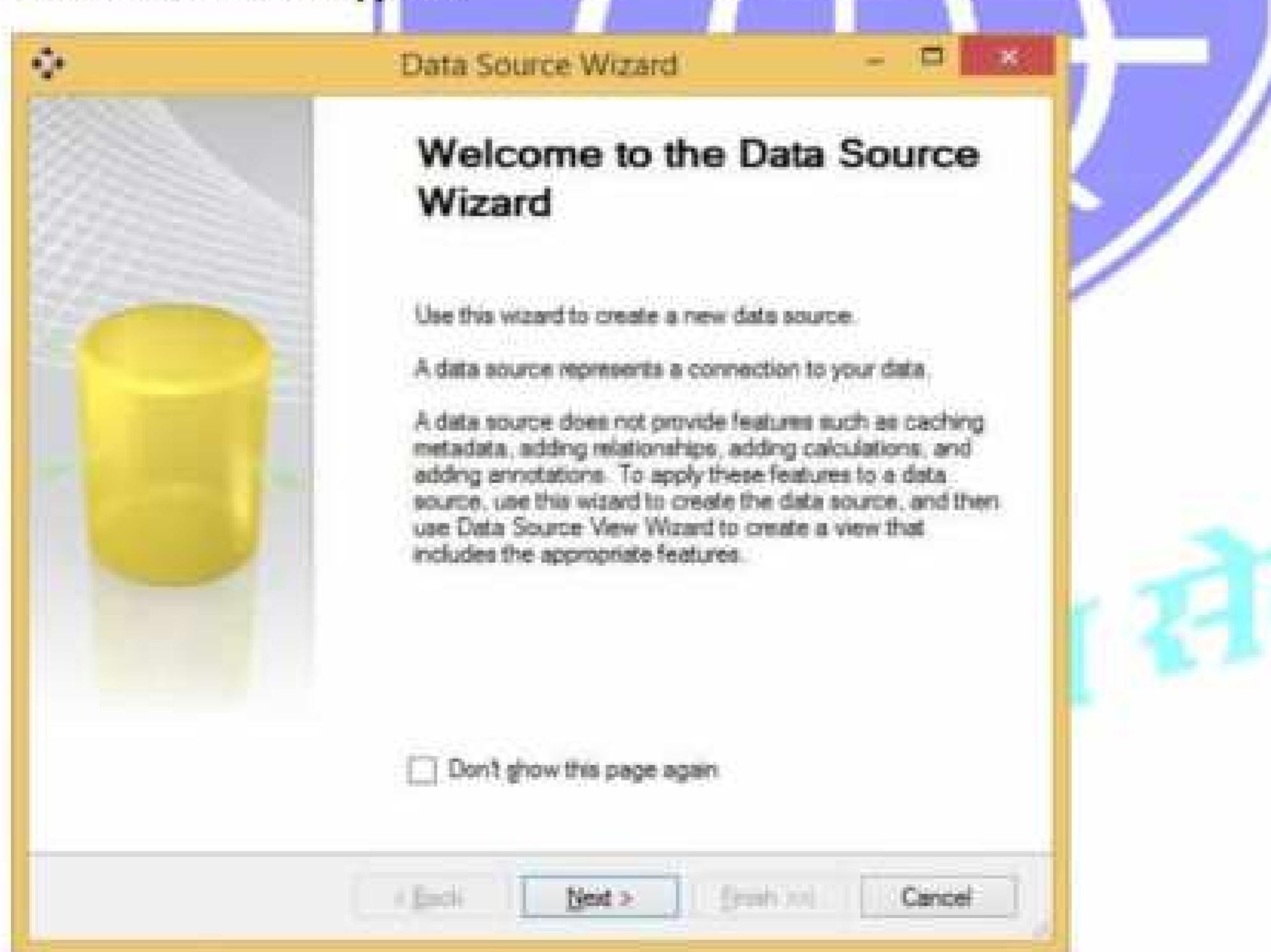


Right click on Data Sources in solution explorer → New Data Source

नियंत्रित करने हें उत्तम संवाधन

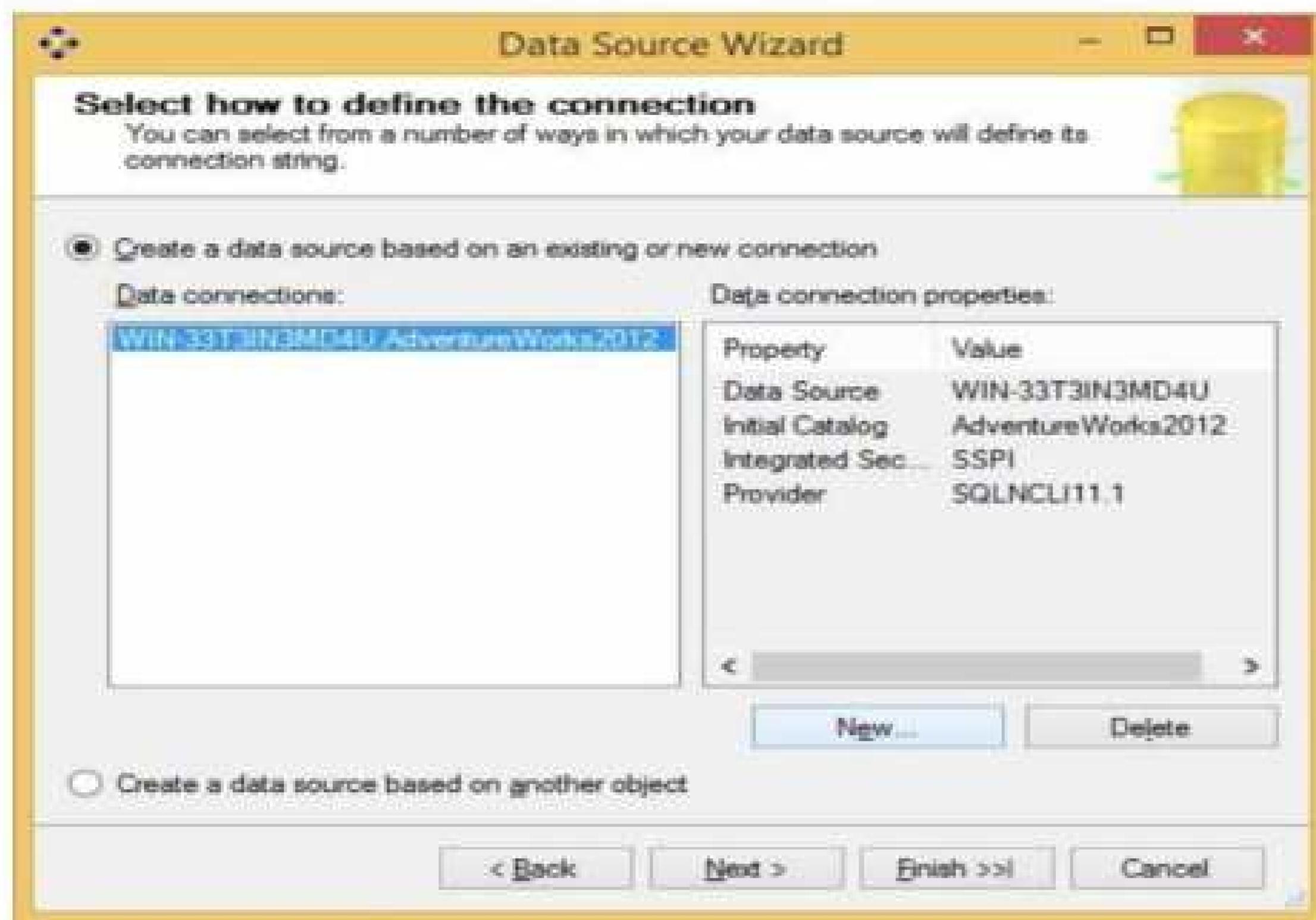


Data Source Wizard appears





Click on New



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)

नियंत्रित करने हेतु सबसे अच्छा



Connection Manager

Provider: Native OLE DB\SQL Server Native Client 11.0

Connection

All

Server name: WIN-33T3IN3MD4U Refresh

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:
SalesDW

Attach a database file:
Logical name:
Browse...

Test Connection OK Cancel Help

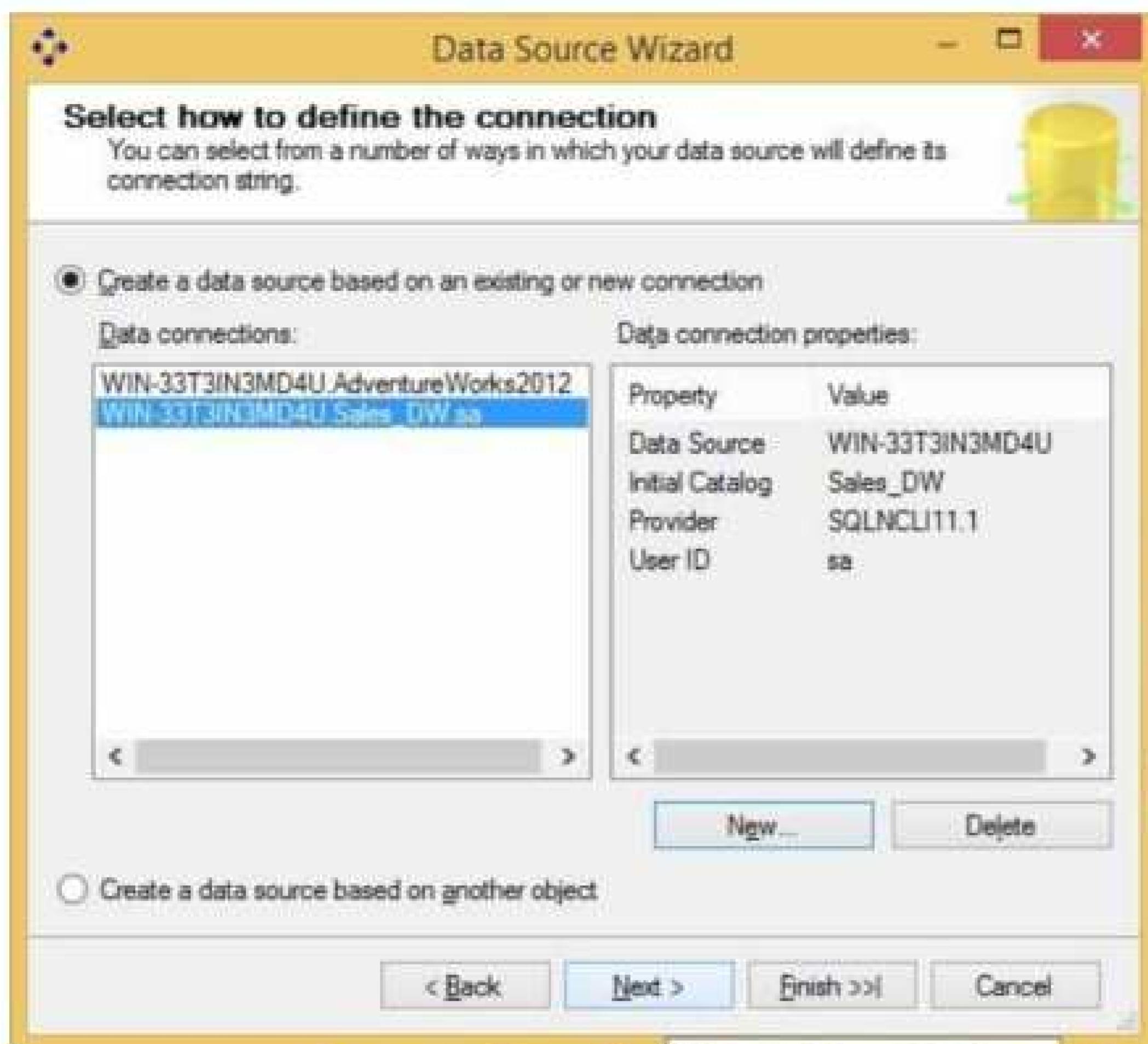
Connection Manager

Test connection succeeded.

OK

Click Next

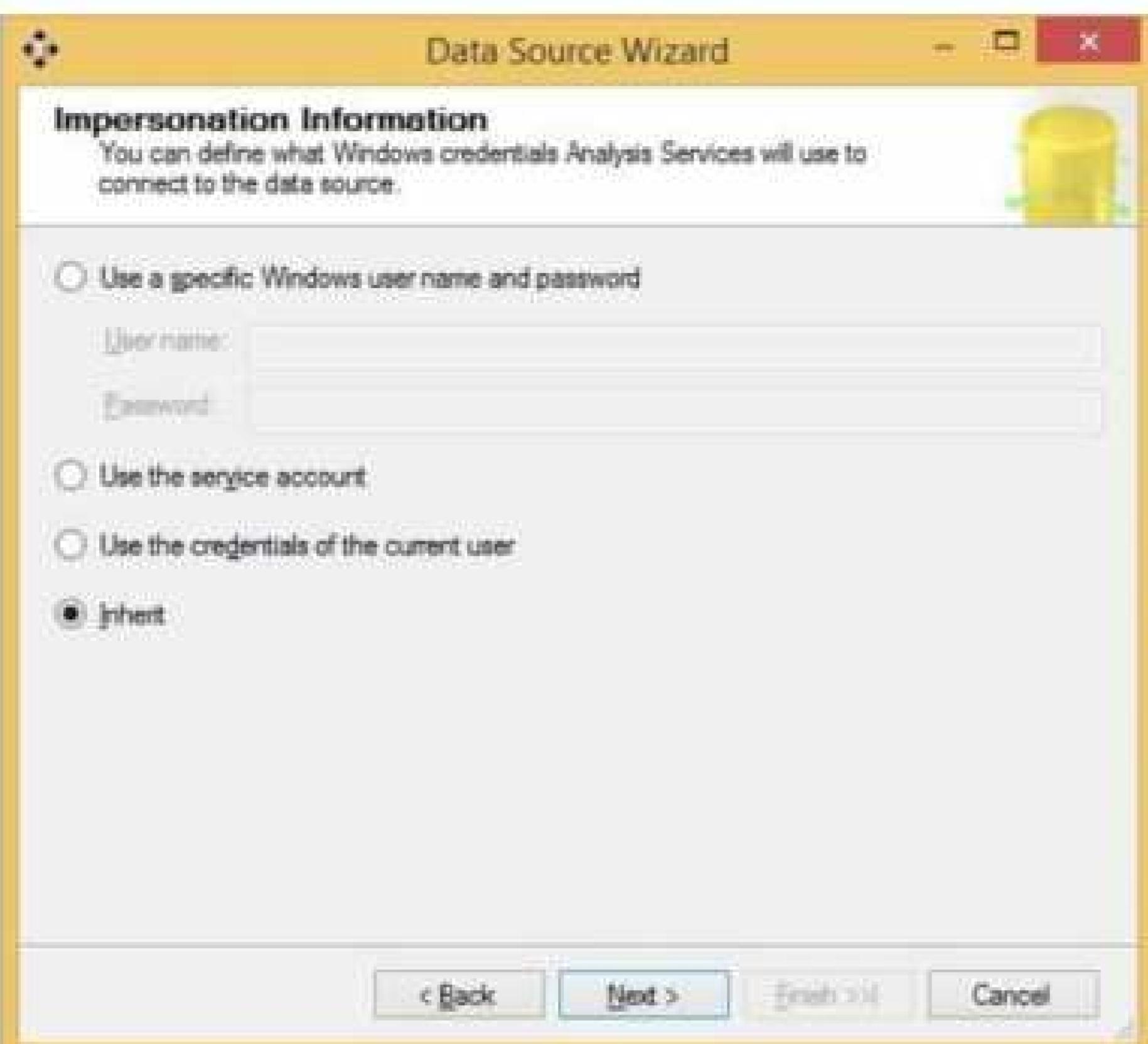
नियमित नोहे उत्तम सेवाधर्म



Select Inherit → Next



नियमित नोट उत्तम सेवाधर्म

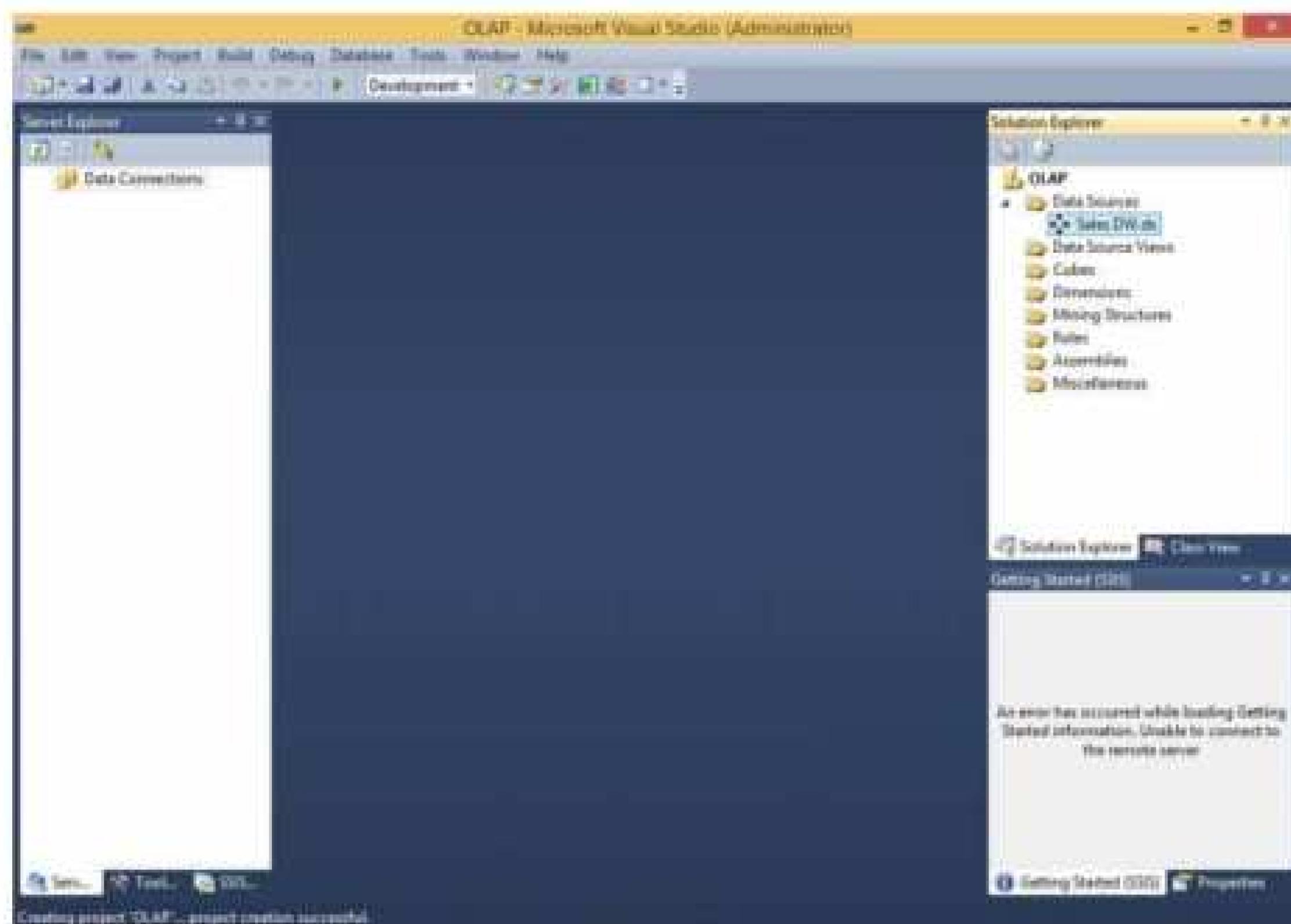


Click Finish





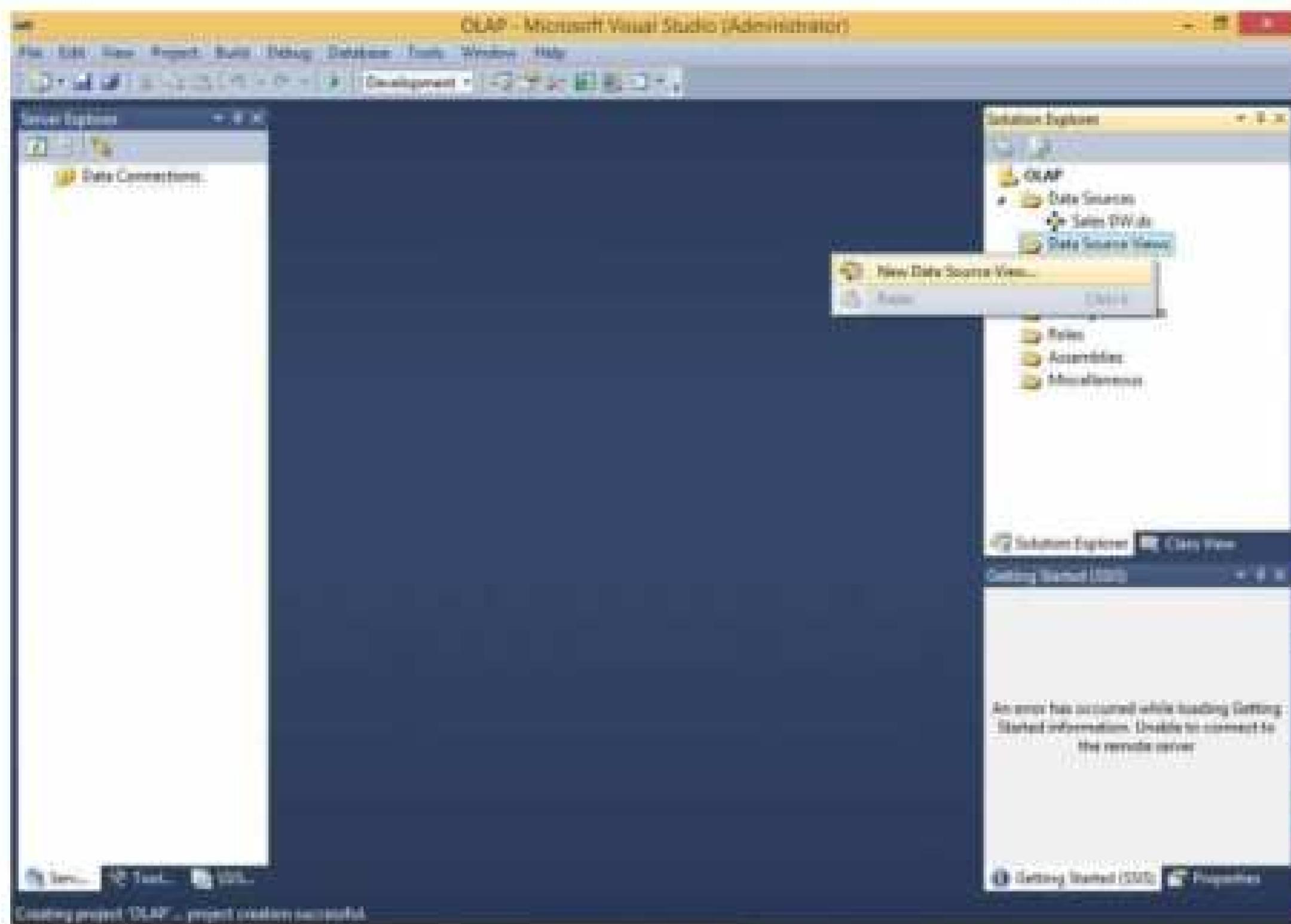
Sales_DW.ds gets created under Data Sources in Solution Explorer



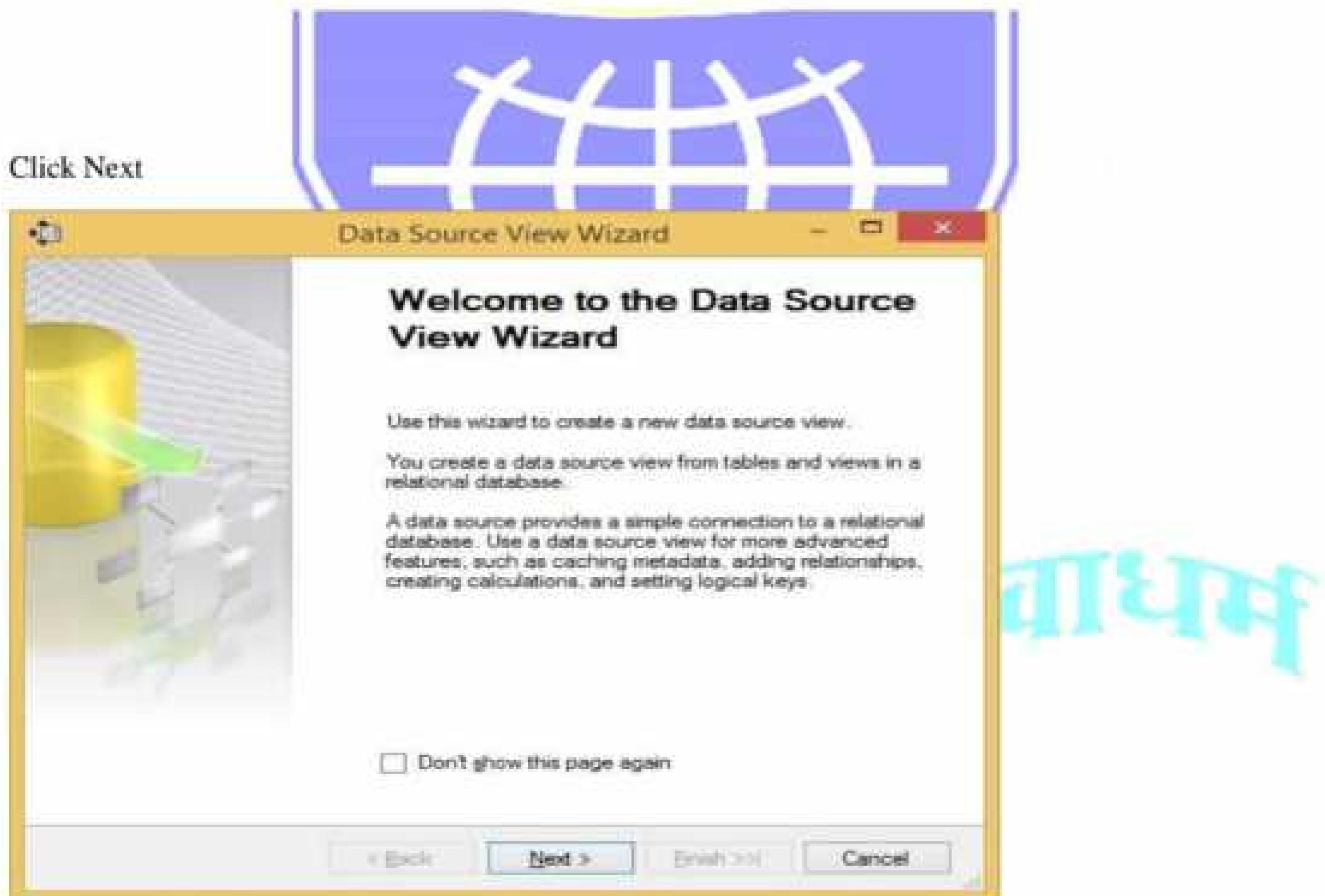
Step 3: Creating New Data Source View

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

नियंत्रित करने ही उत्तम सेवाधर्म

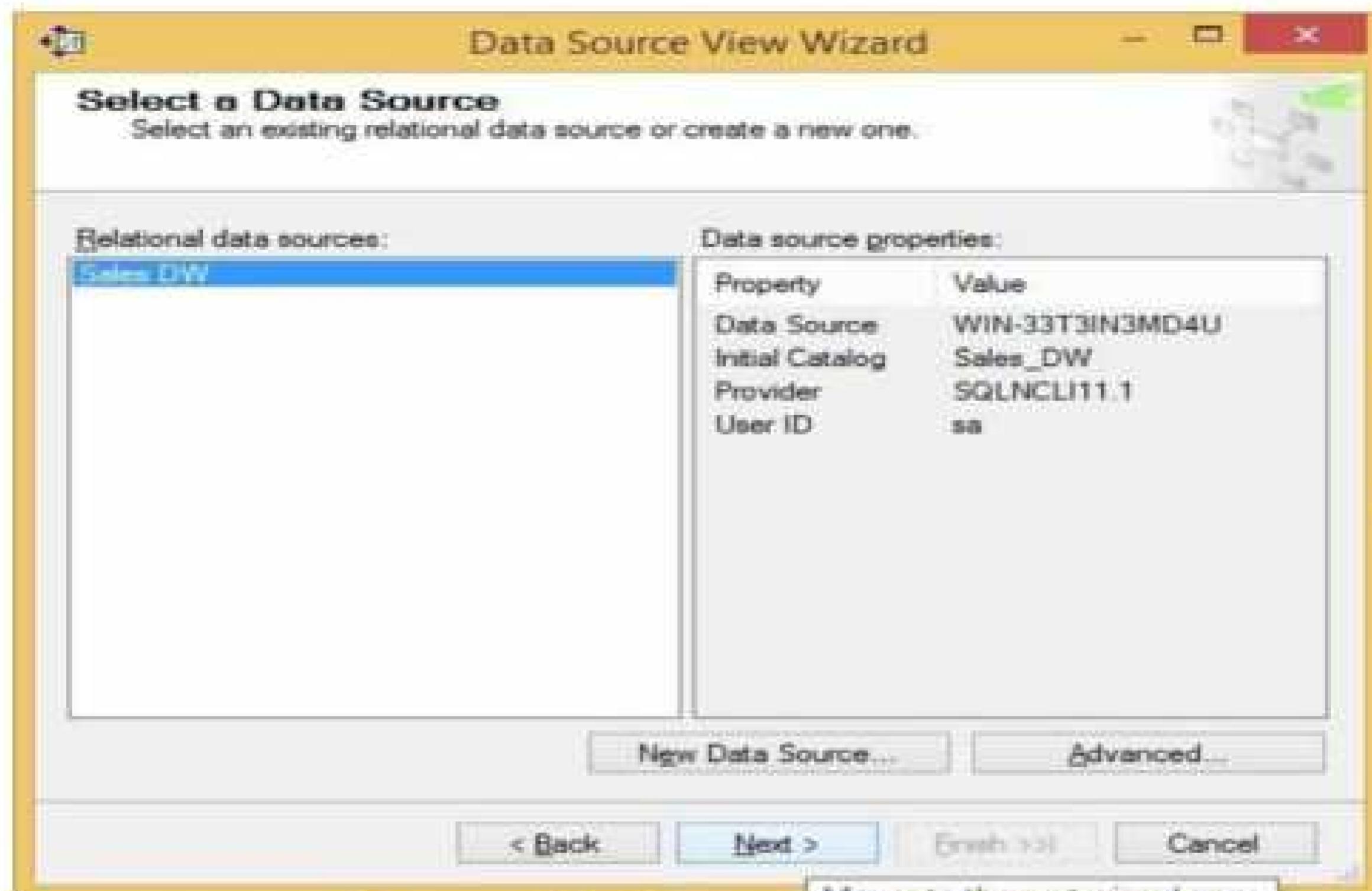


Click Next





Click Next



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

नियंत्रित करने हें उत्तम संवाधक



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:

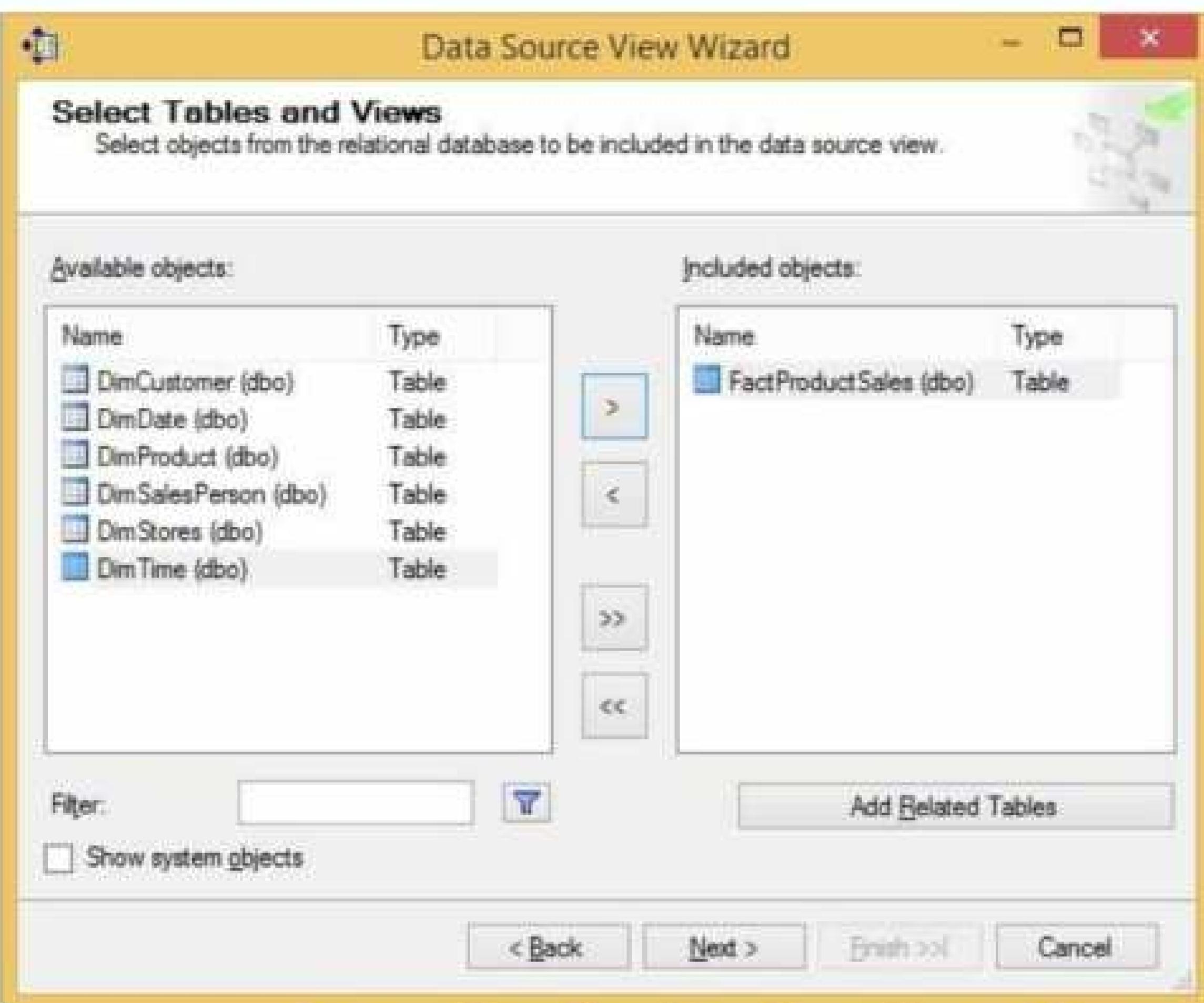
Add Related Tables

Show system objects

< Back Next > Finish | Cancel



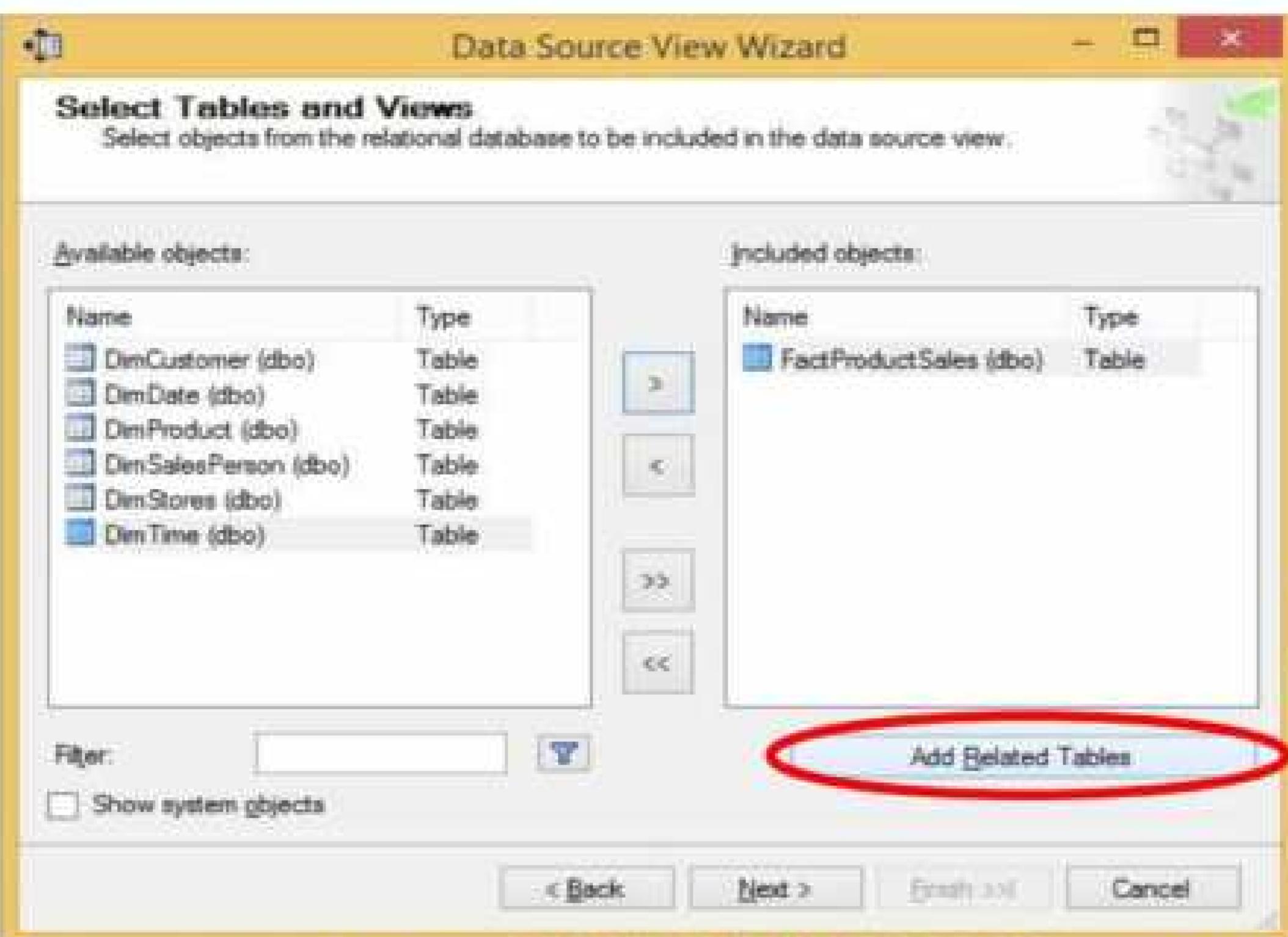
निष्ठानोह उत्तम सेवाधर्म



Click on Add Related Tables



विस्तृत विज्ञान एवं संगणक
विज्ञान विभाग



Click Next



नियमित नोहे उत्तम सेवाधर्म



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:

Preview:

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

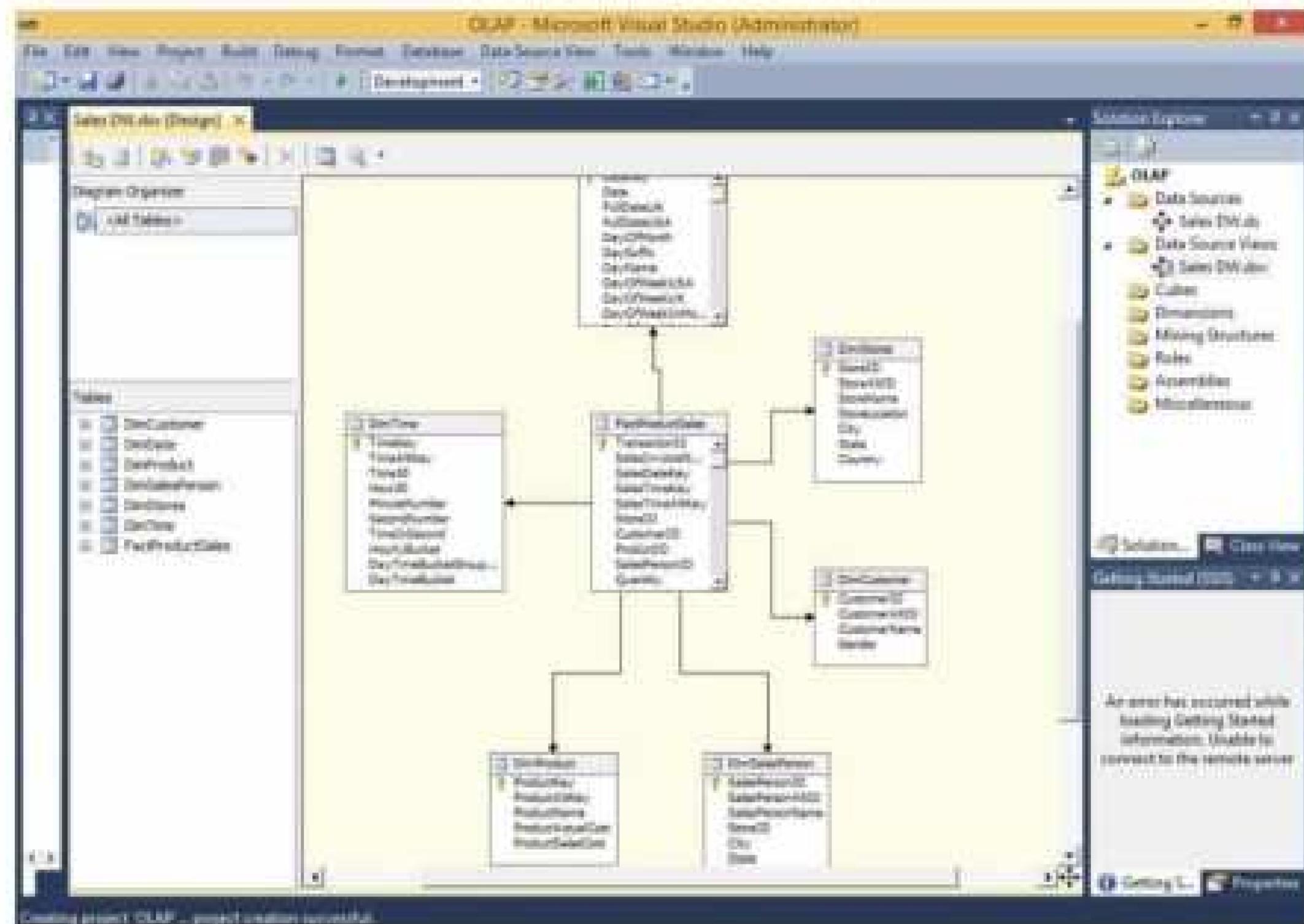
I completed the wizard



ग्राधकी



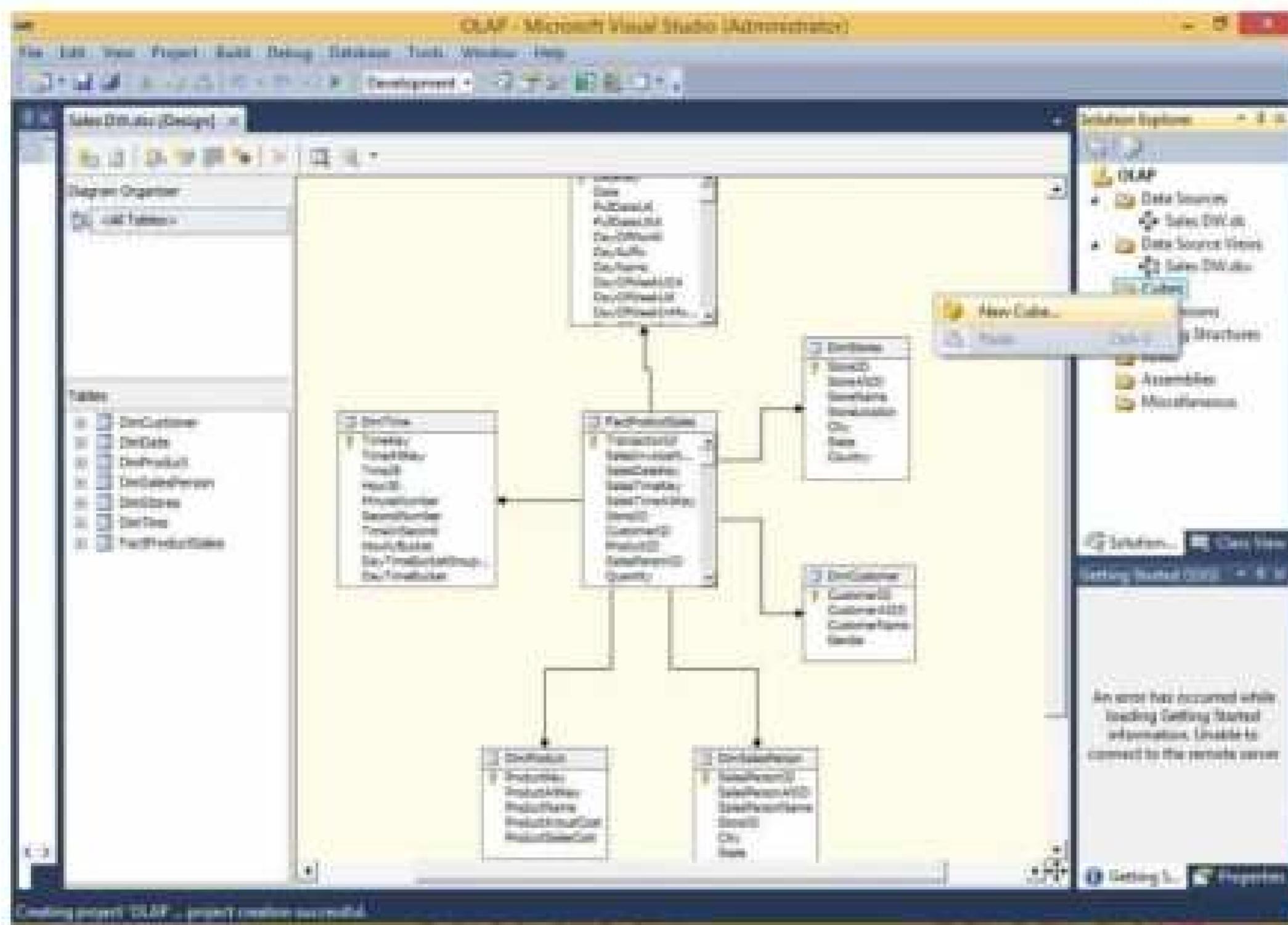
Sales DW.dsv appears in Data Source Views in Solution Explorer.



Step 4: Creating new cube

Right click on Cubes → New Cube

नियंत्रित करने हें उत्तम संवाधम्



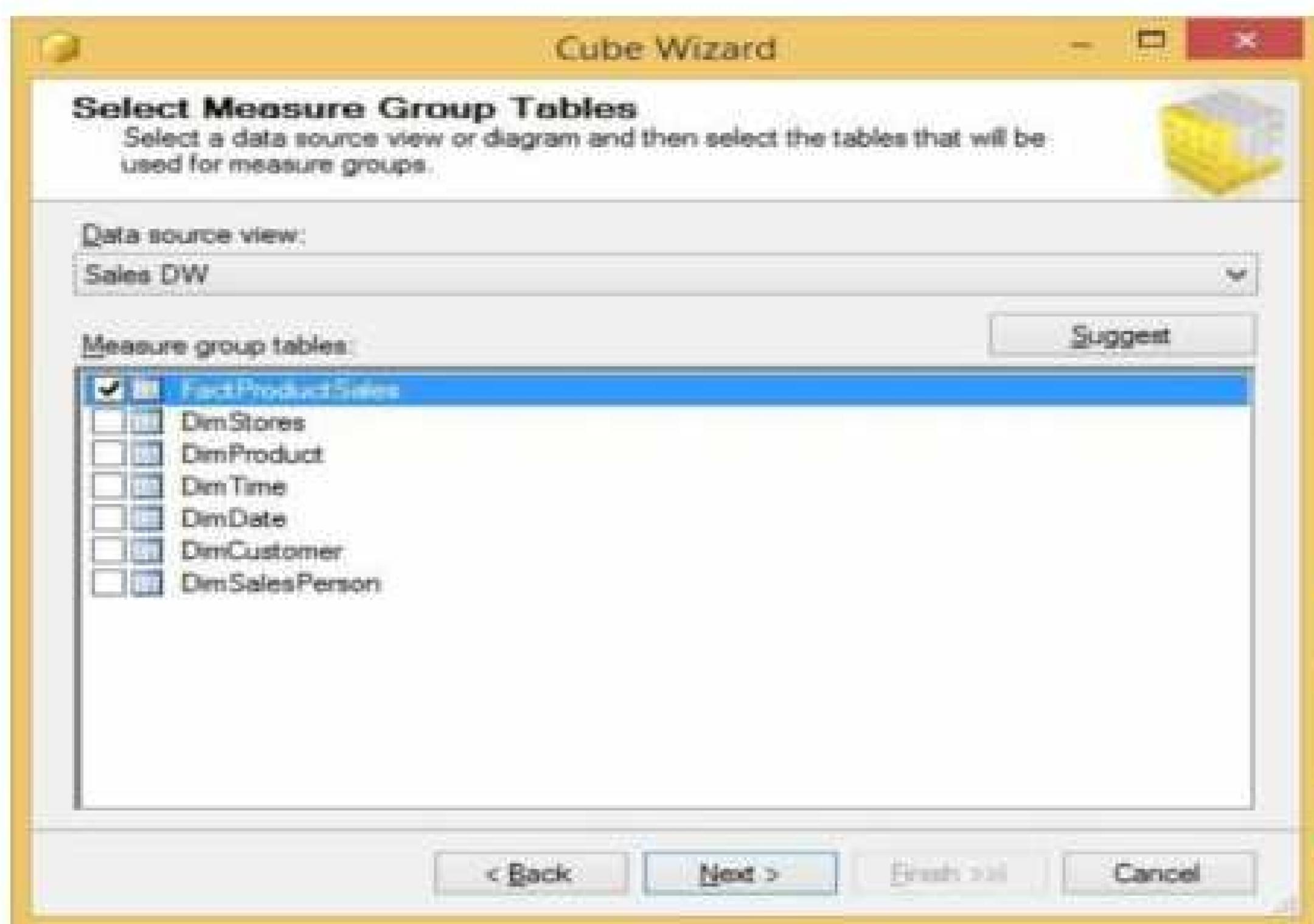
Select Use existing tables in Select Creation Method → Next



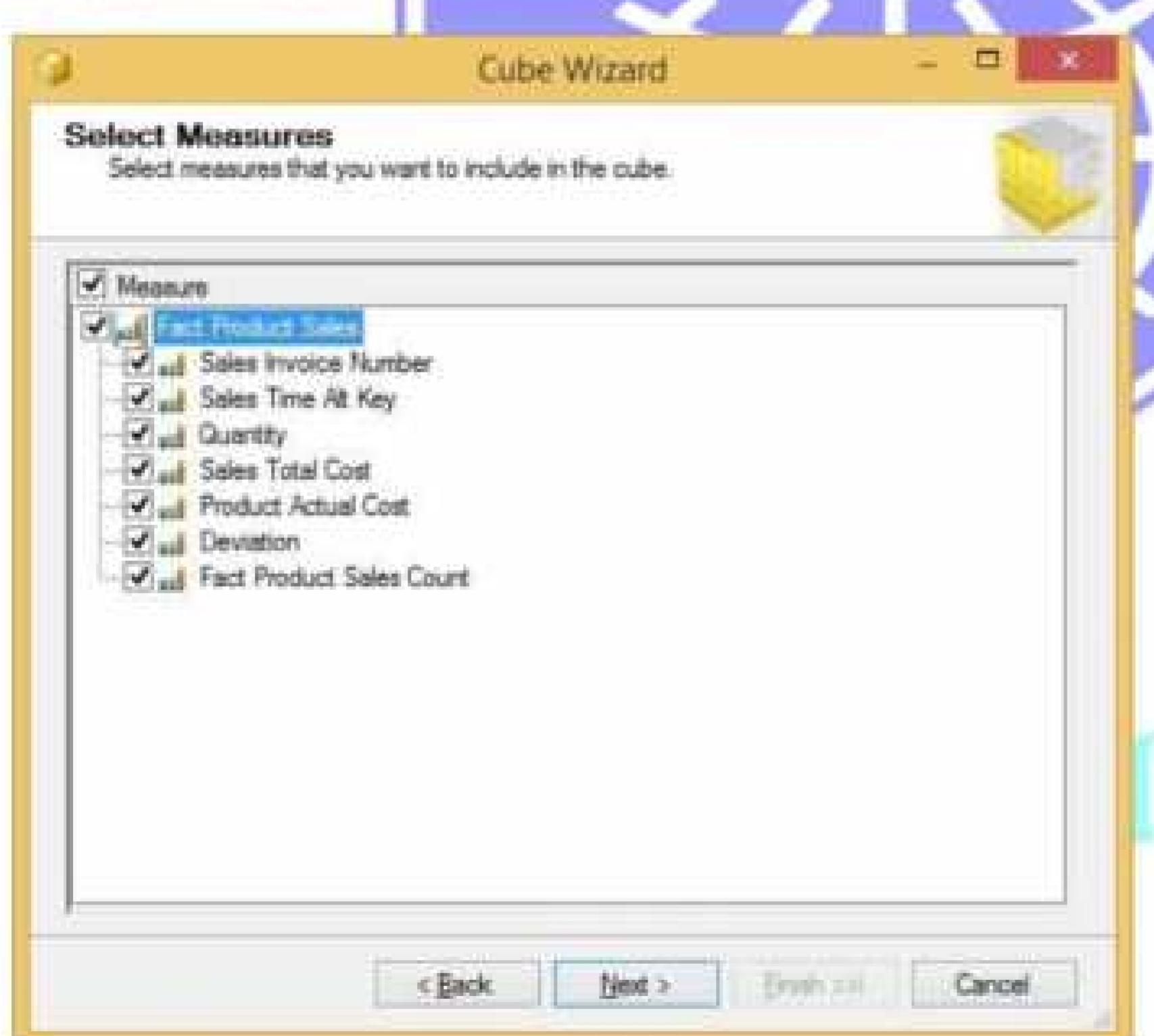
In Select Measure Group Tables → Select FactProductSales → Click Next



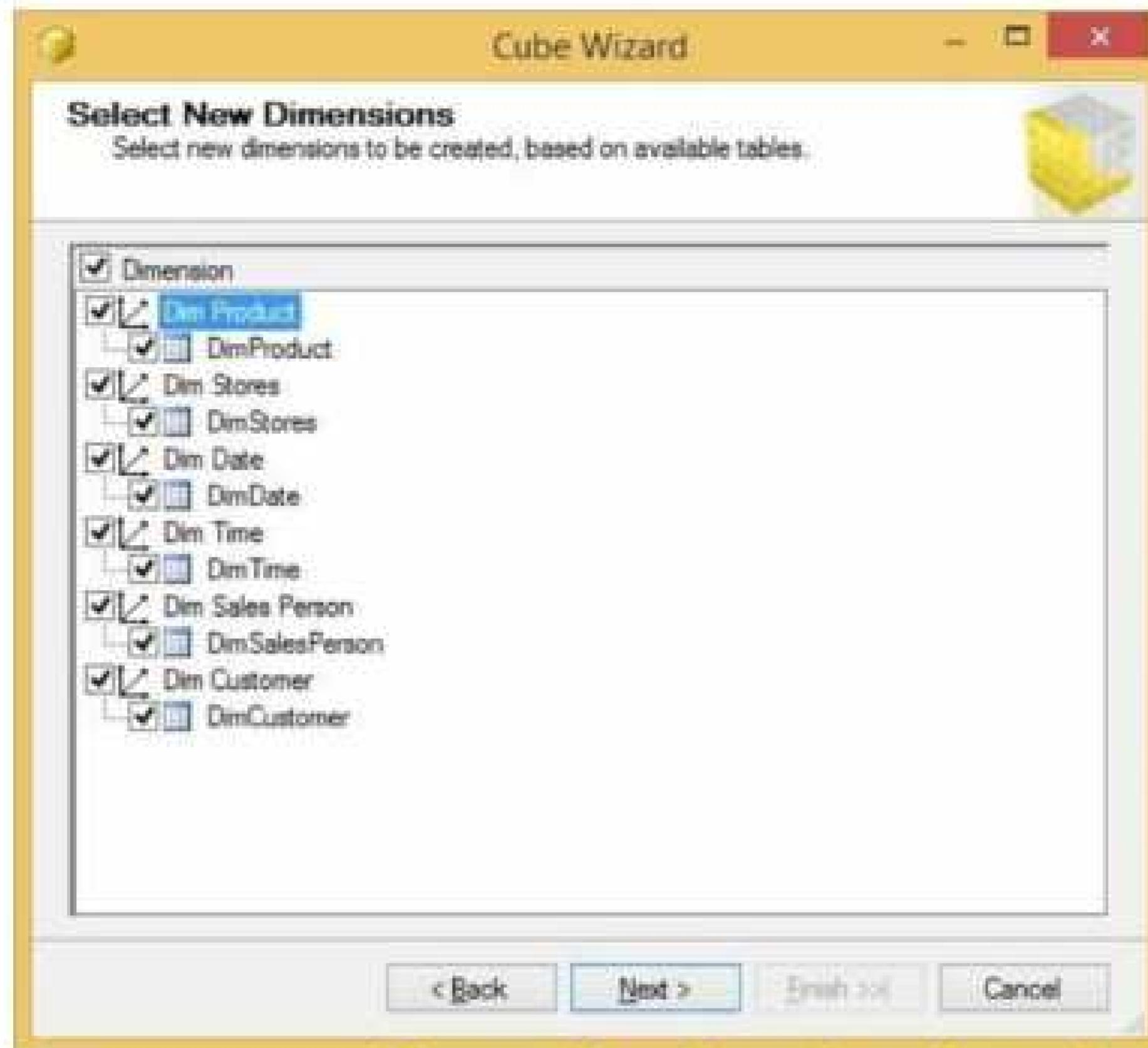
नियमित नोट उत्तम सेवाधर्म



In Select Measures → check all measures → Next



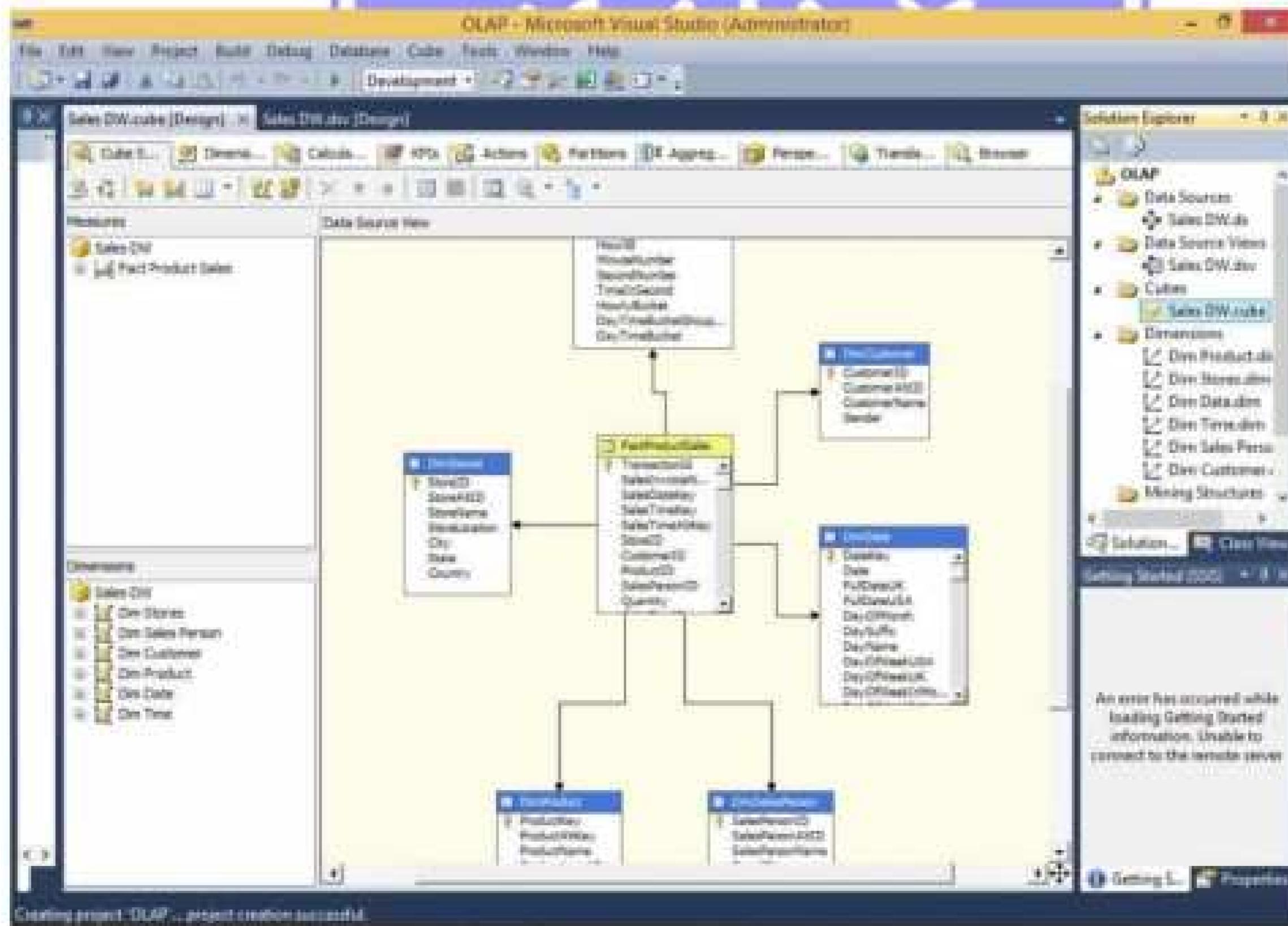
In Select New Dimensions → Check all Dimensions → Next



Click on Finish



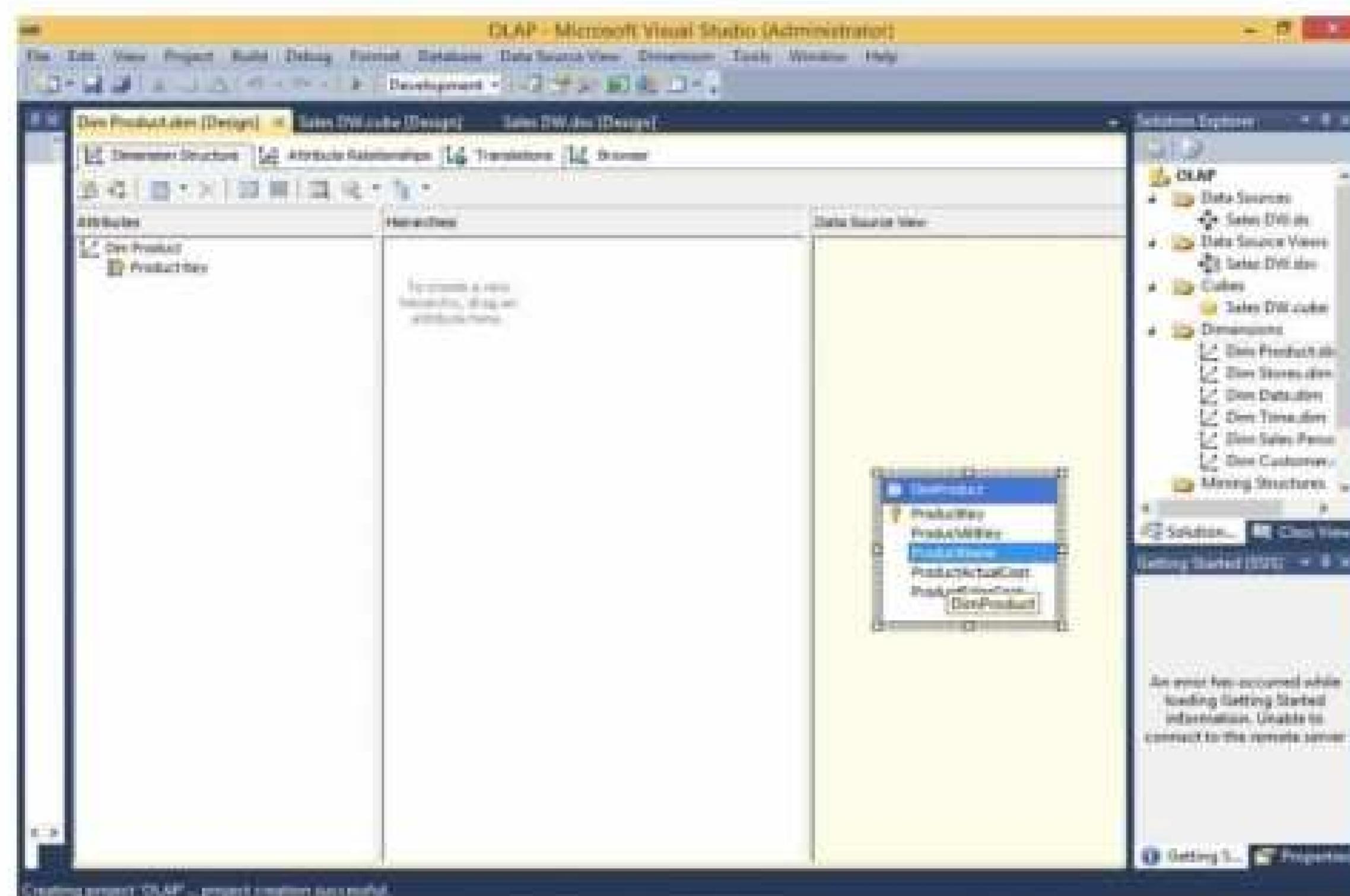
Sales_DW.cube is created





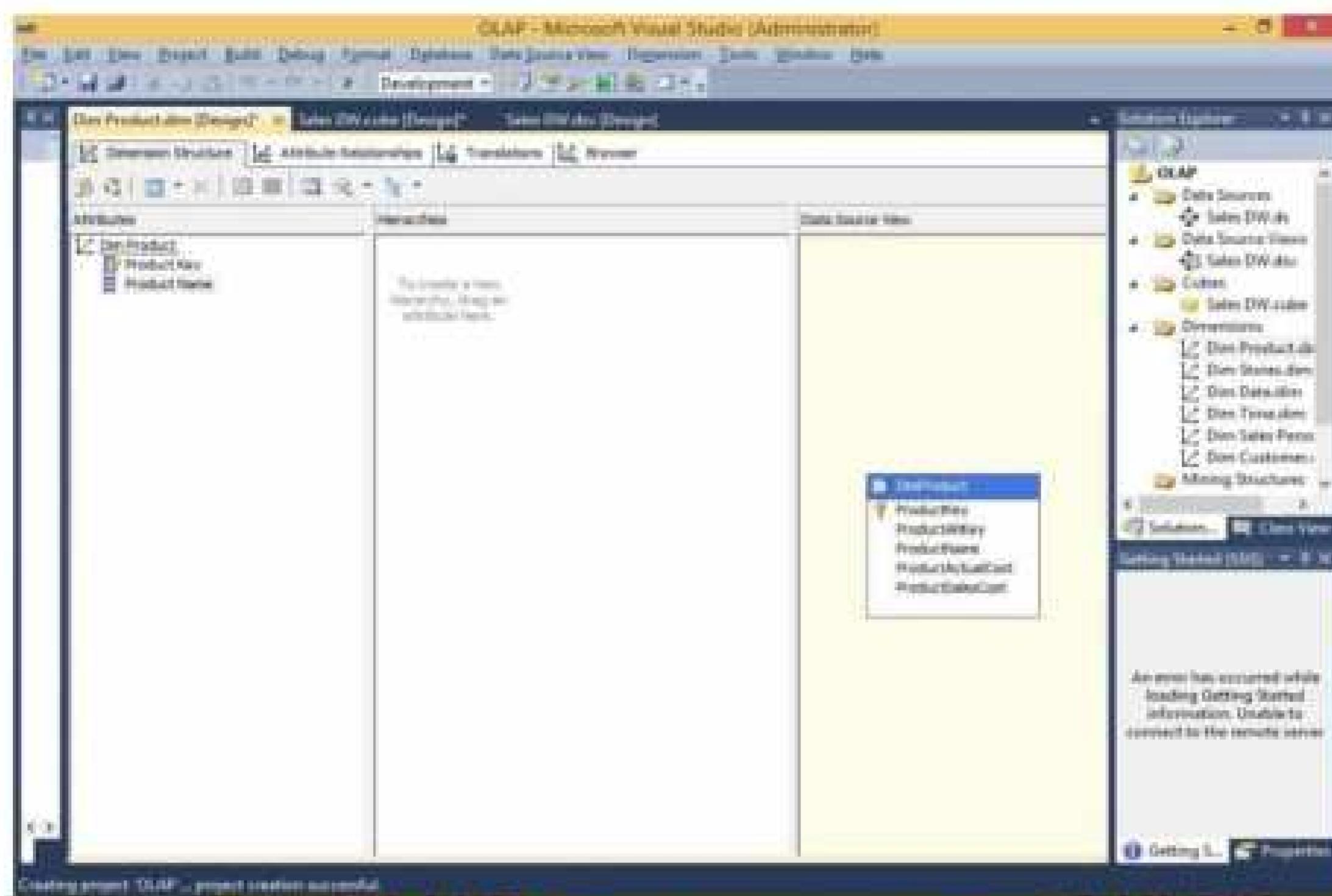
Step 5: Dimension Modification

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

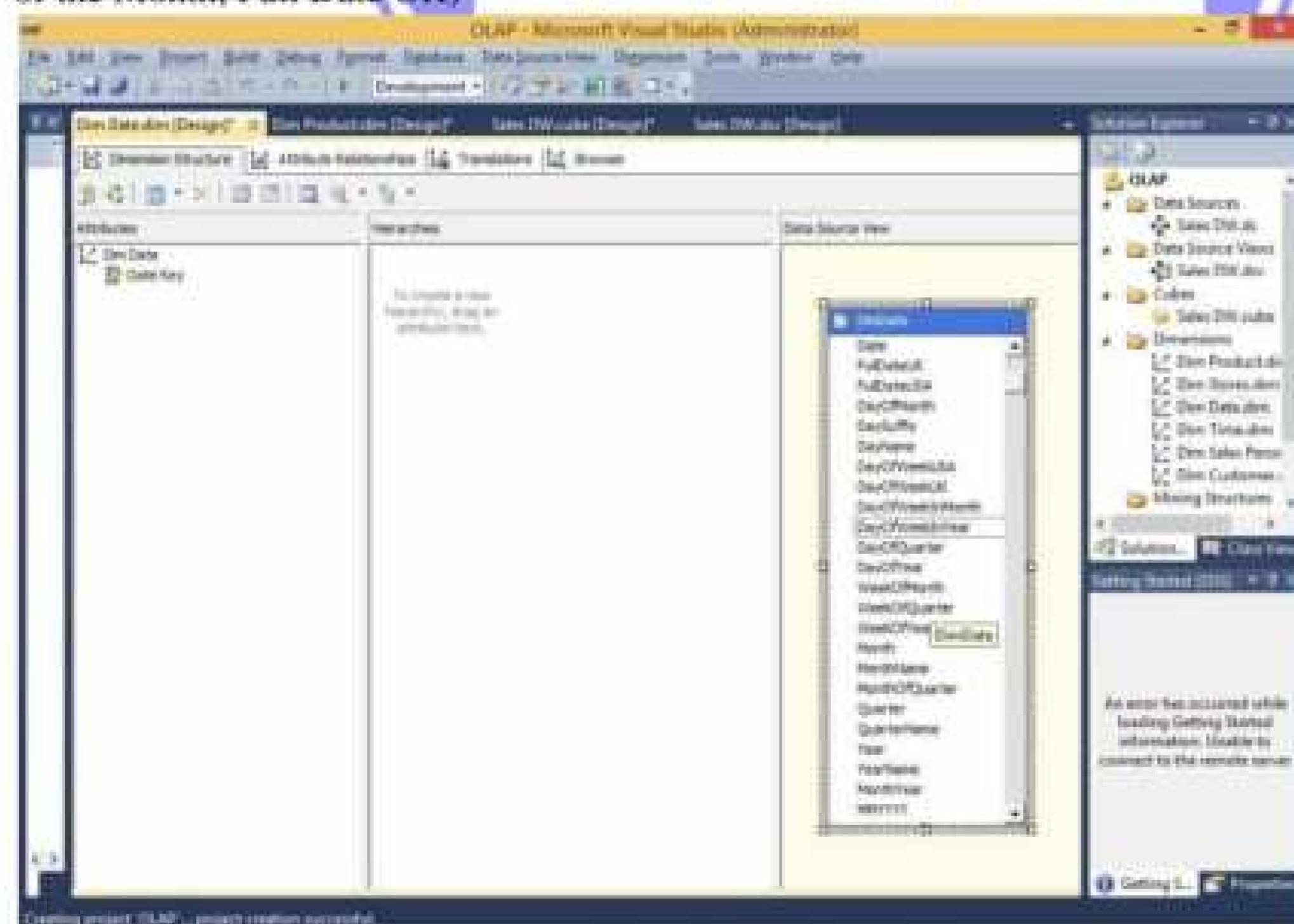
निष्ठानोह उत्तम सेवाधर्म

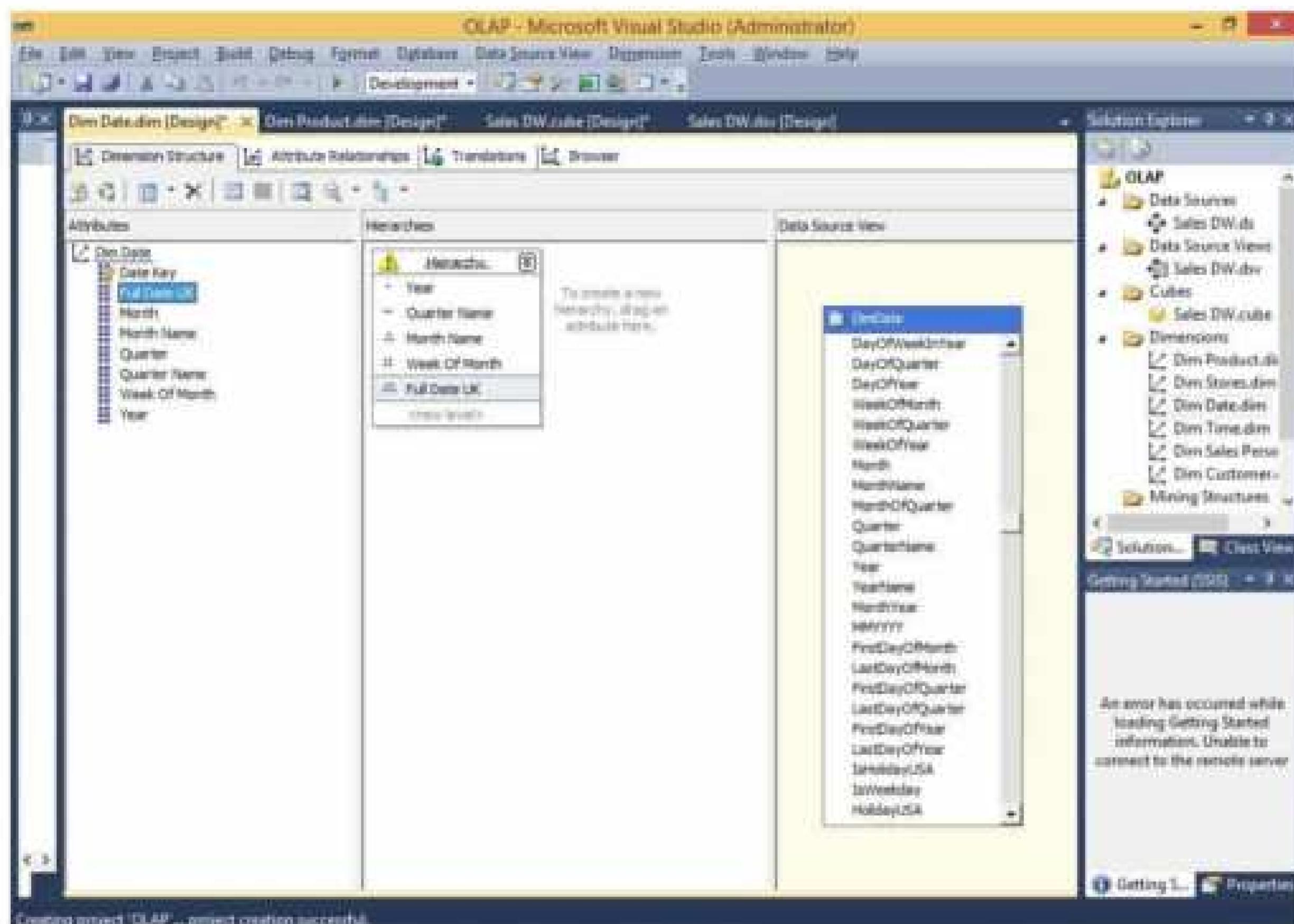


Step 6: Creating Attribute Hierarchy in Date Dimension

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)

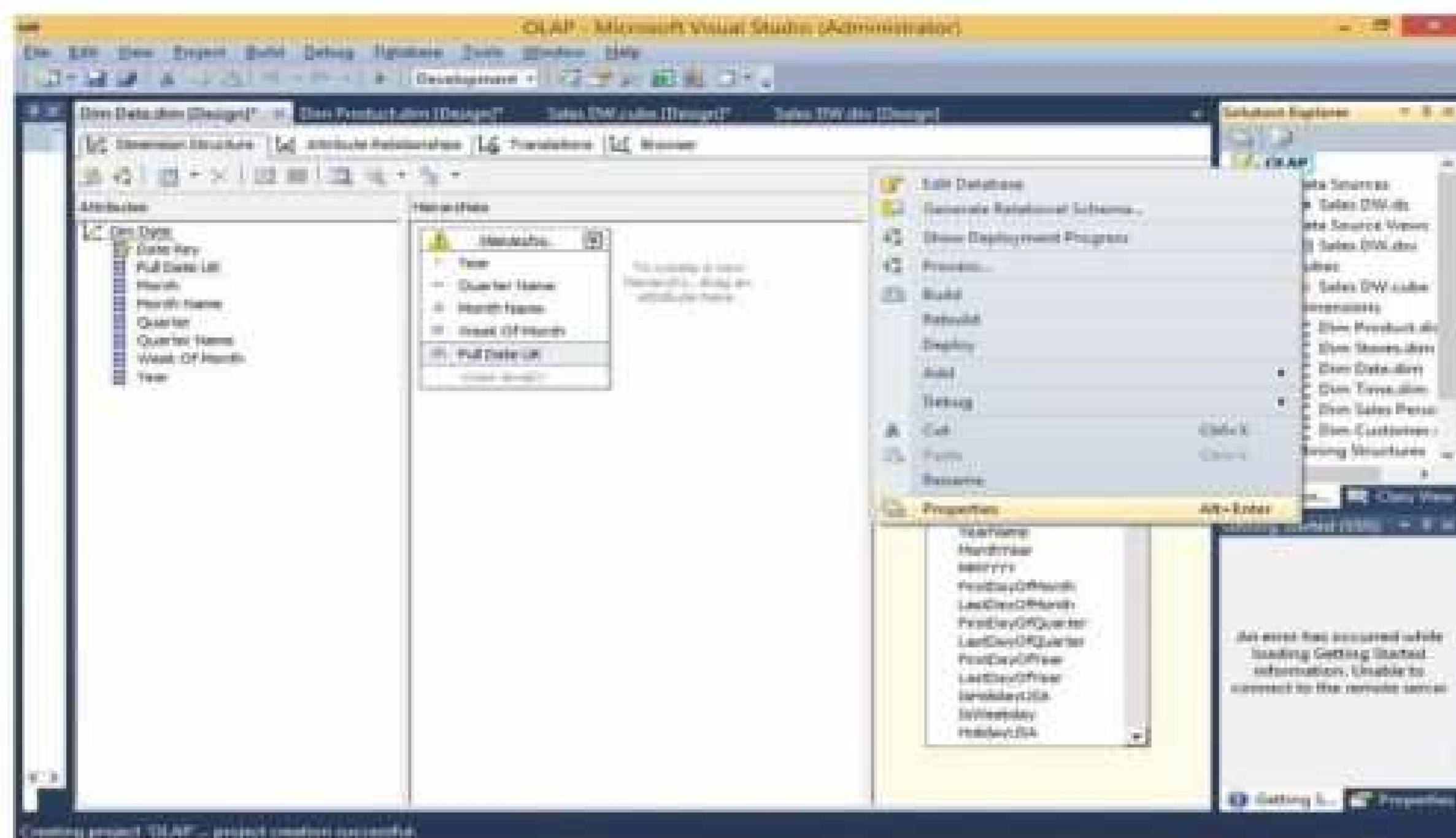




Step 7: Deploy Cube

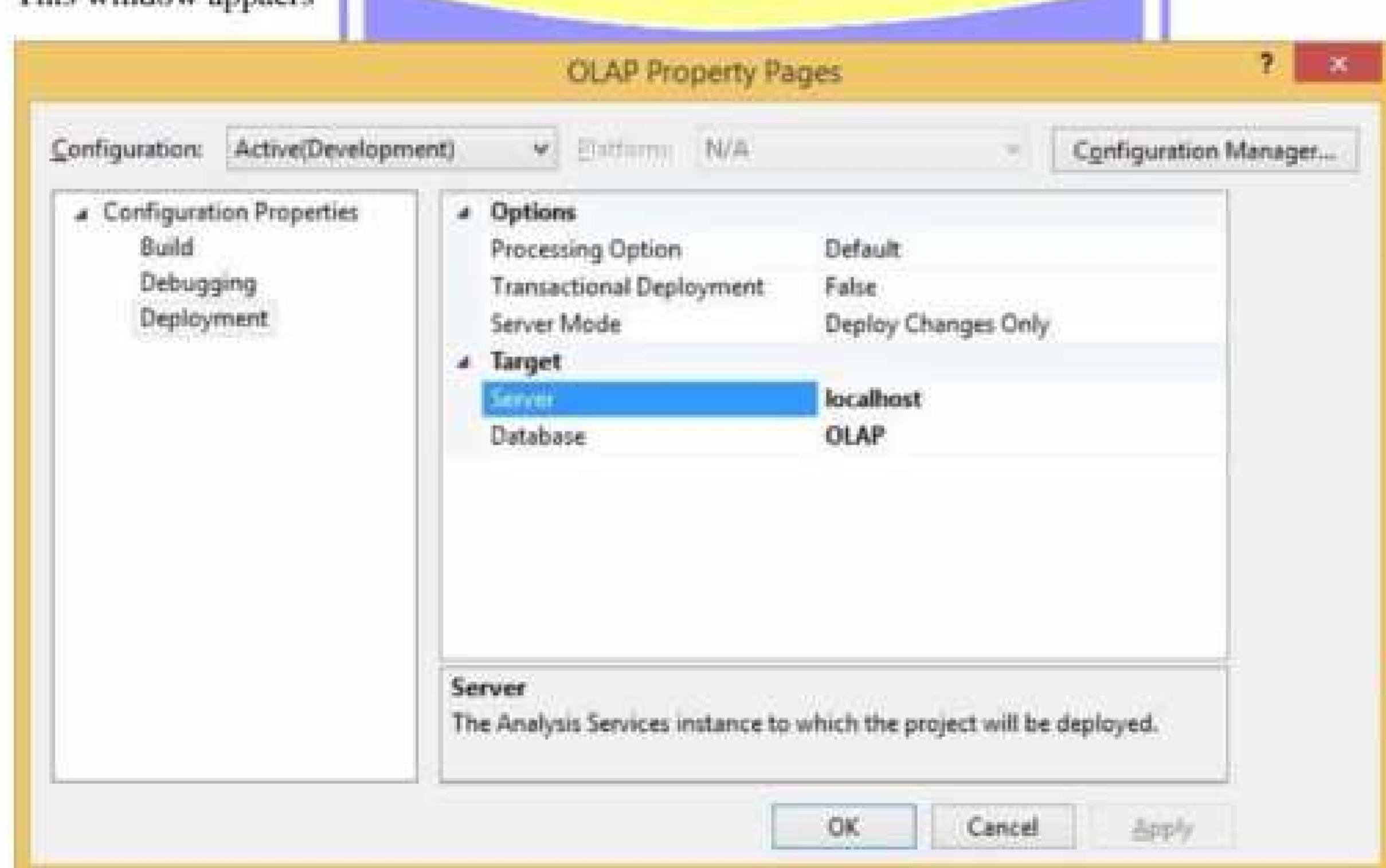
Right click on Project name → Properties

निस्लिनोह उत्तम सेवाधर्म

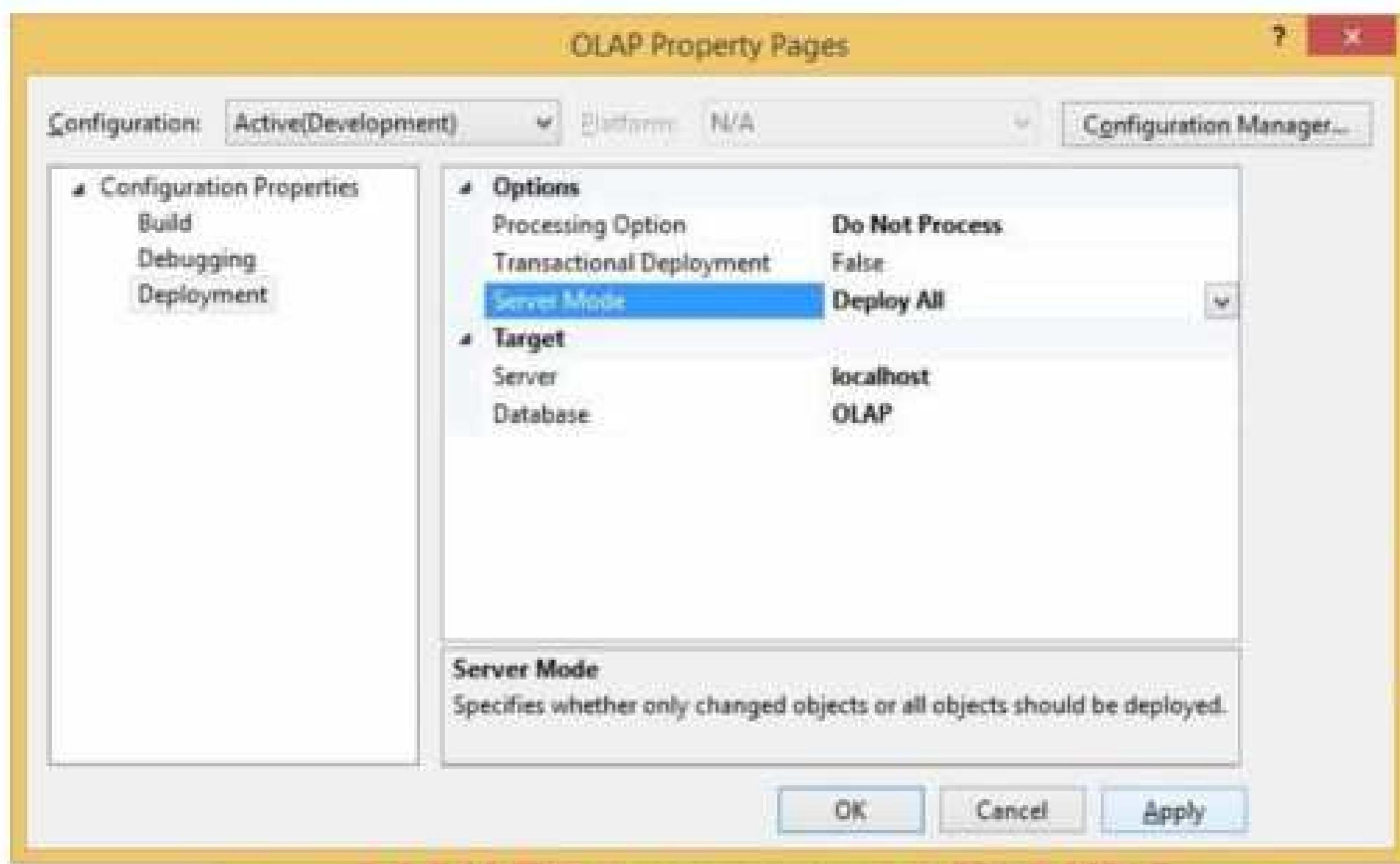


This window appears

*** LED TO EXCELLENCE IN EDUCATION ***

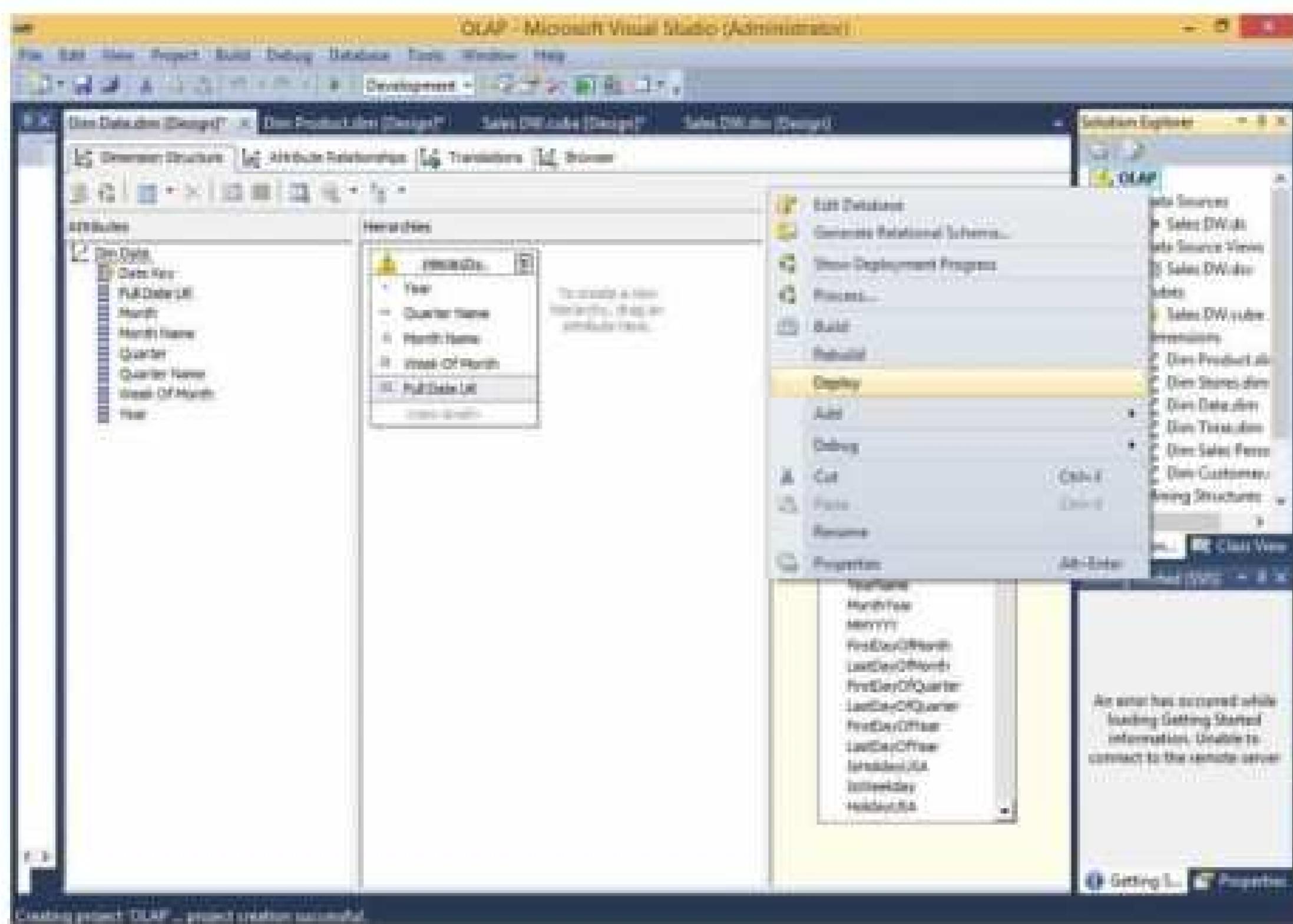


Do following changes and click on Apply & ok



Right click on project name → Deploy

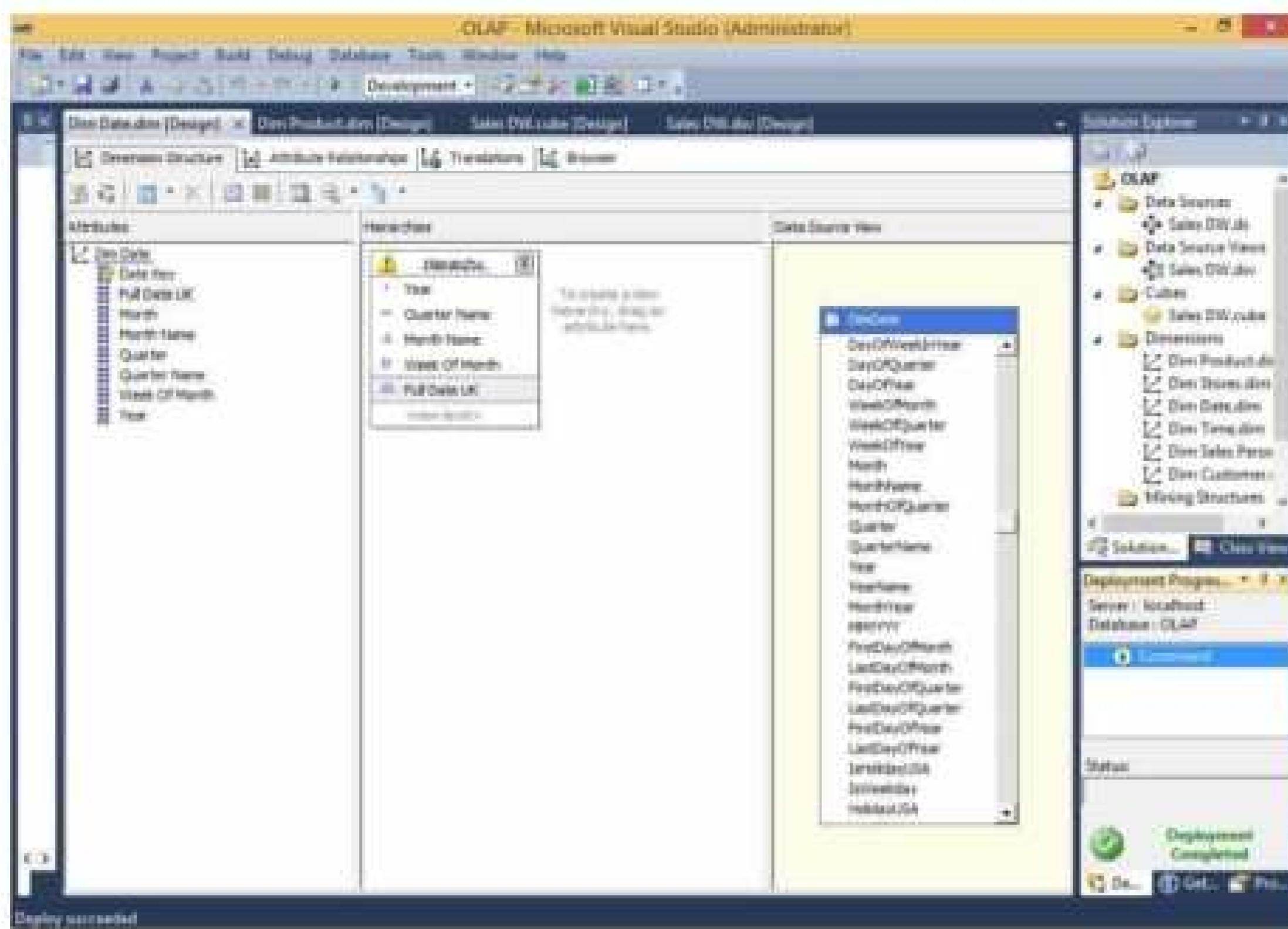
निष्ठानोह उत्स सेवाधर्म



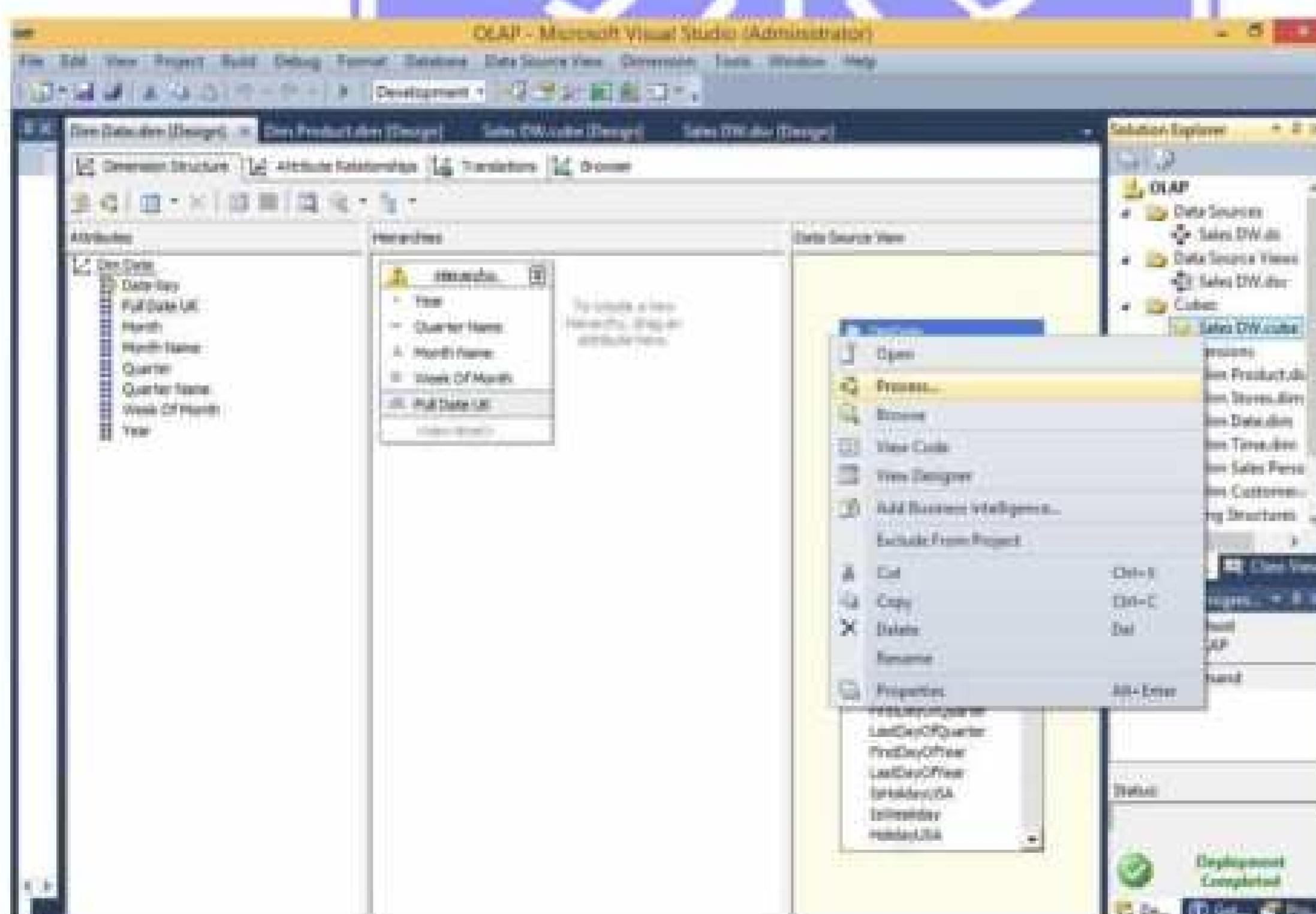
Deployment successful



निष्पत्ति नोह उत्तम सेवाधर्म

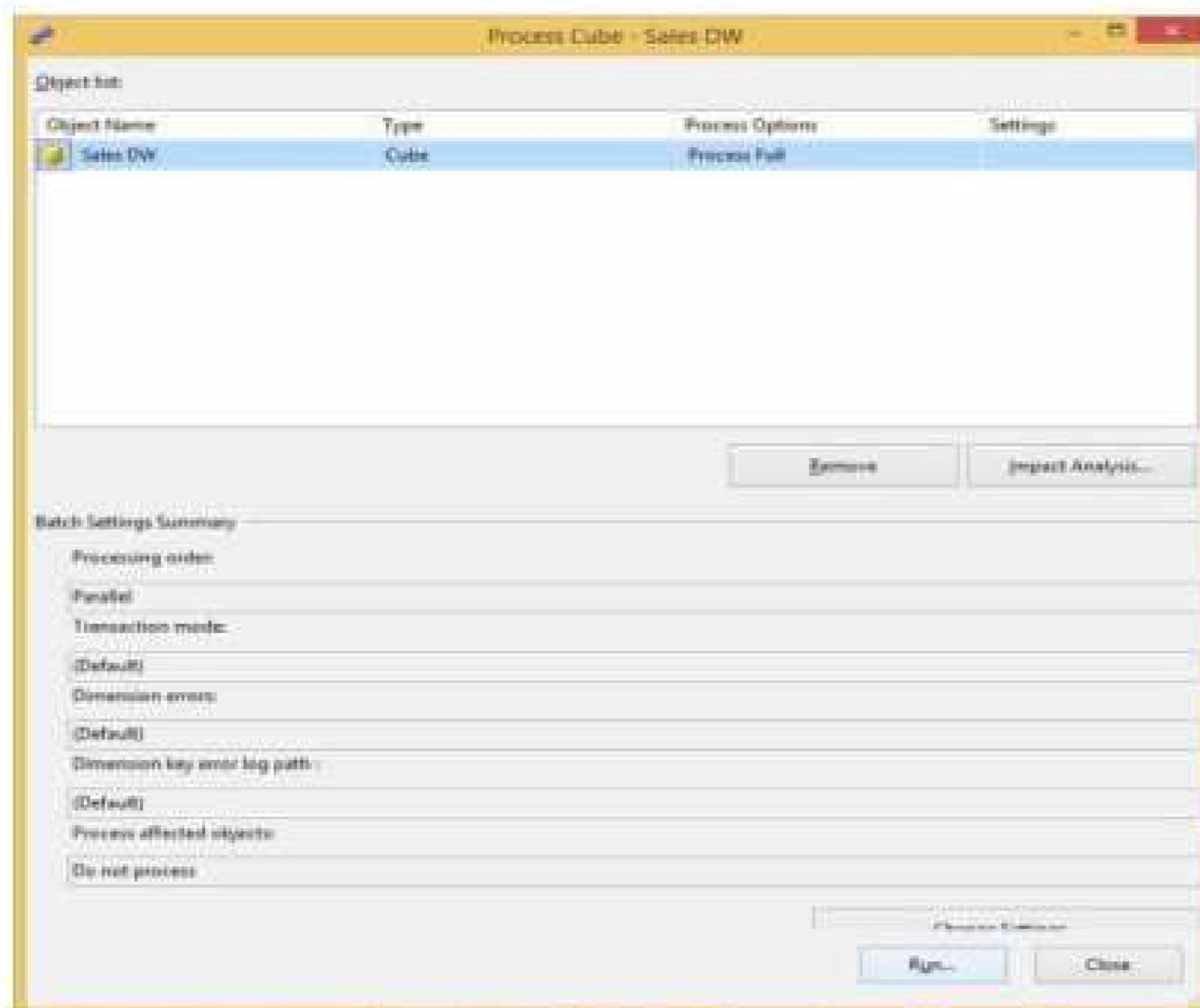


To process cube right click on Sales_DW.cube → Process

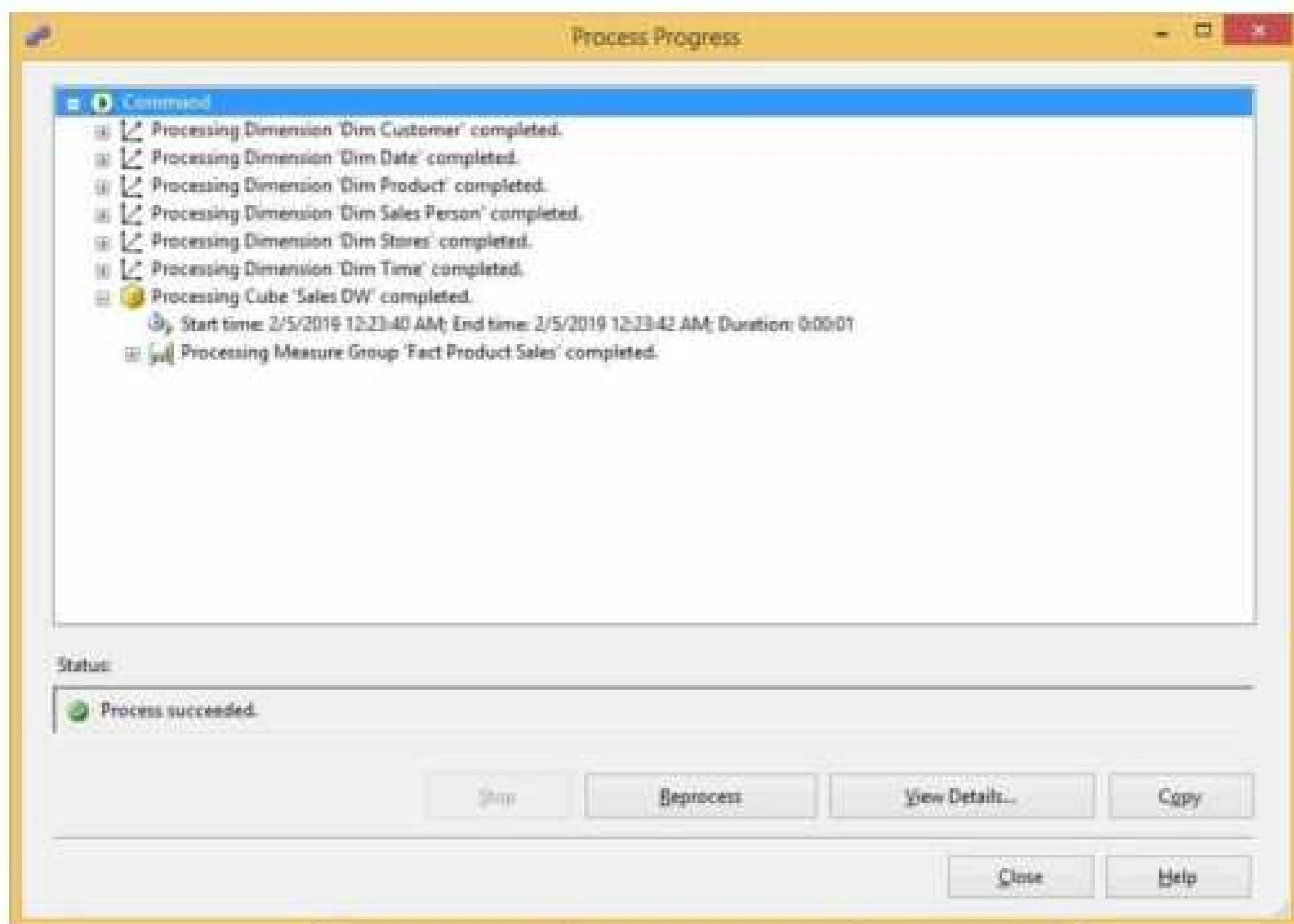




Click run



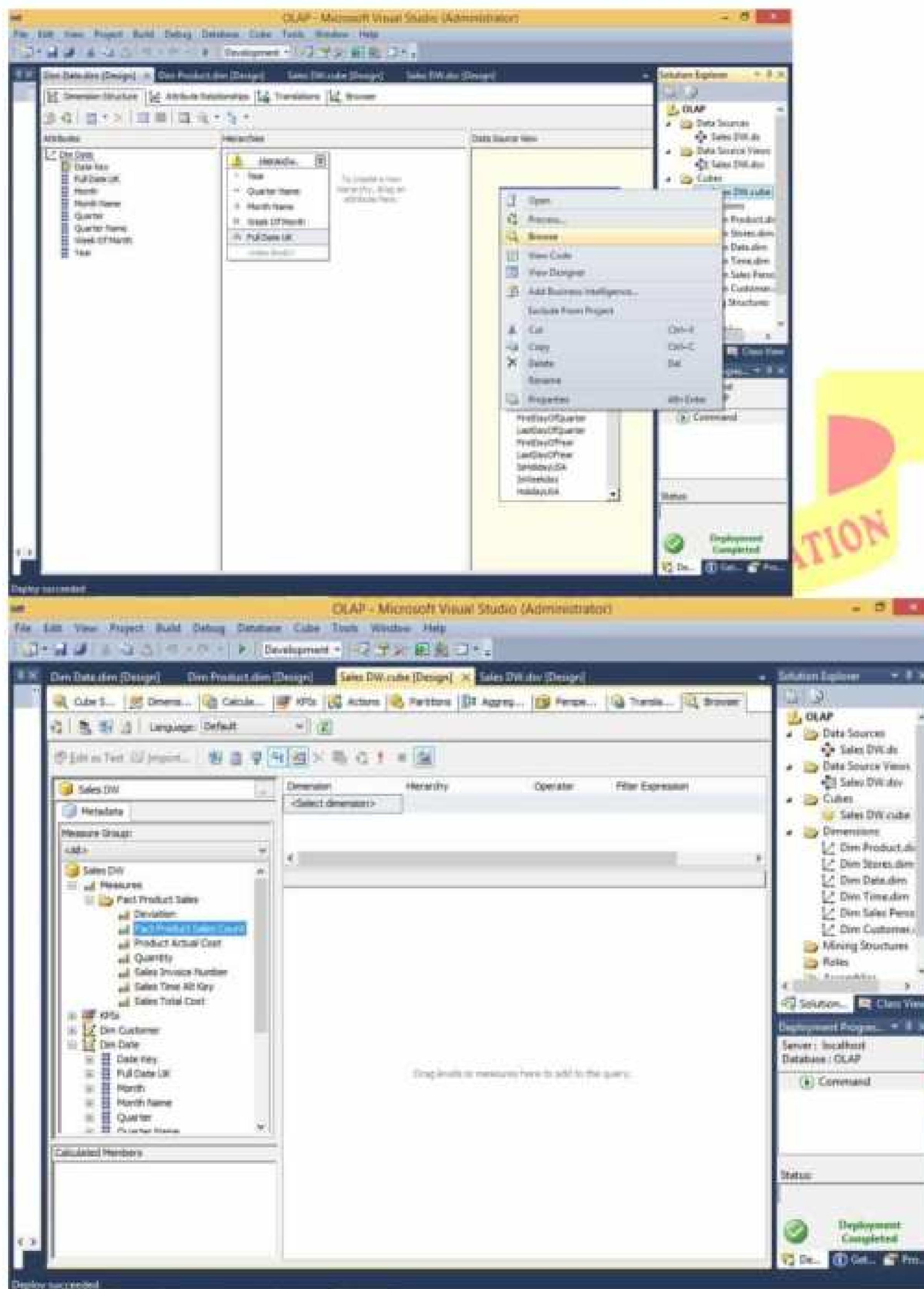
निष्ठानेह उत्तम सेवाधर्म



Browse the cube for analysis in solution explorer



नियमितीय उत्तम सेवाधर्म



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:

```
SELECT [Measures].[Sales] ON COLUMNS
FROM [Adventure Works]
WHERE [Category] = 'Food' AND [Year] = 2024;
```
- **Result Analysis:** Analyze the query results to generate actionable insights, support reporting, and enable dynamic data visualizations.



PRACTICAL 4 b

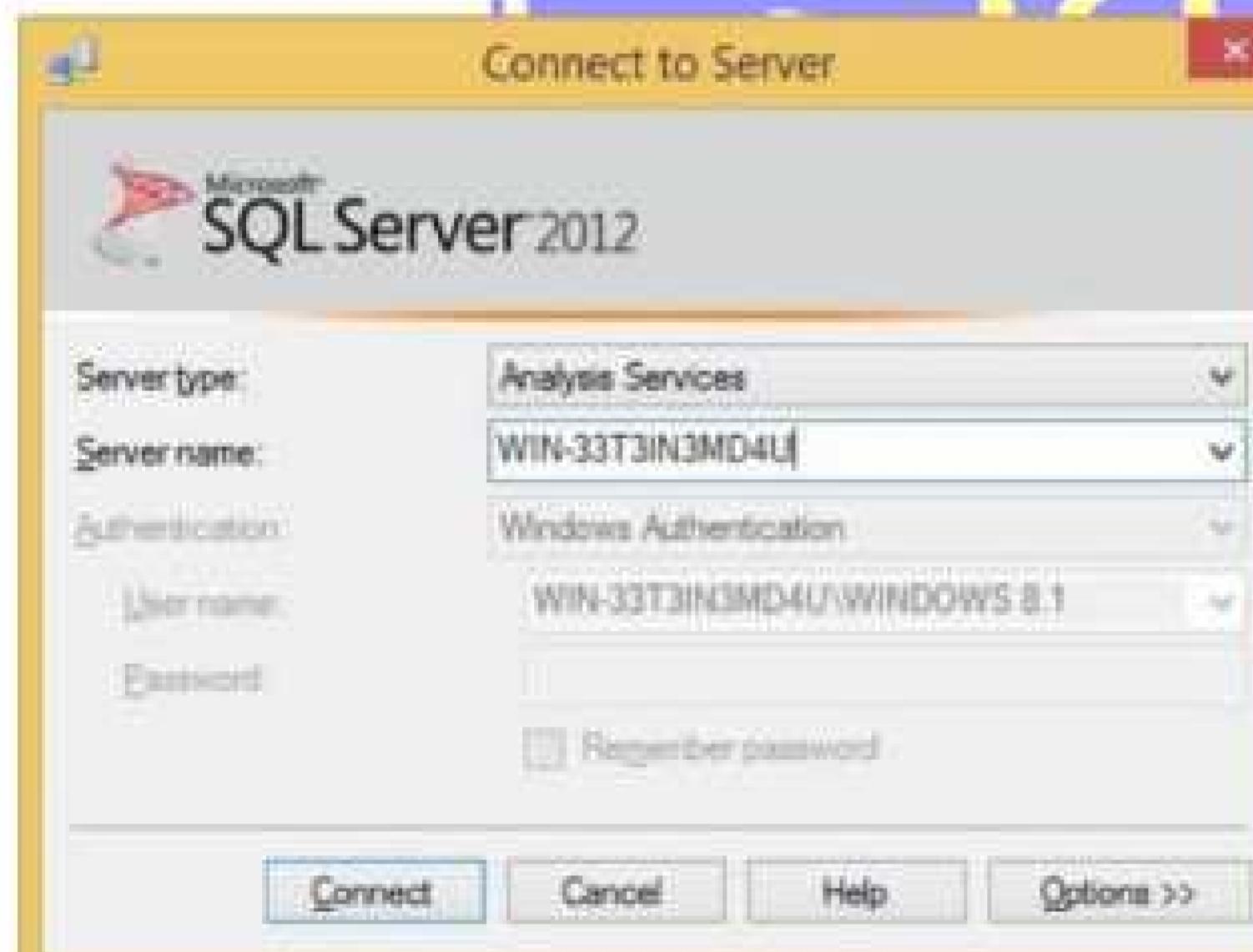
Execute the MDX queries to extract the data from the datawarehouse.

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

Server type: Analysis Services

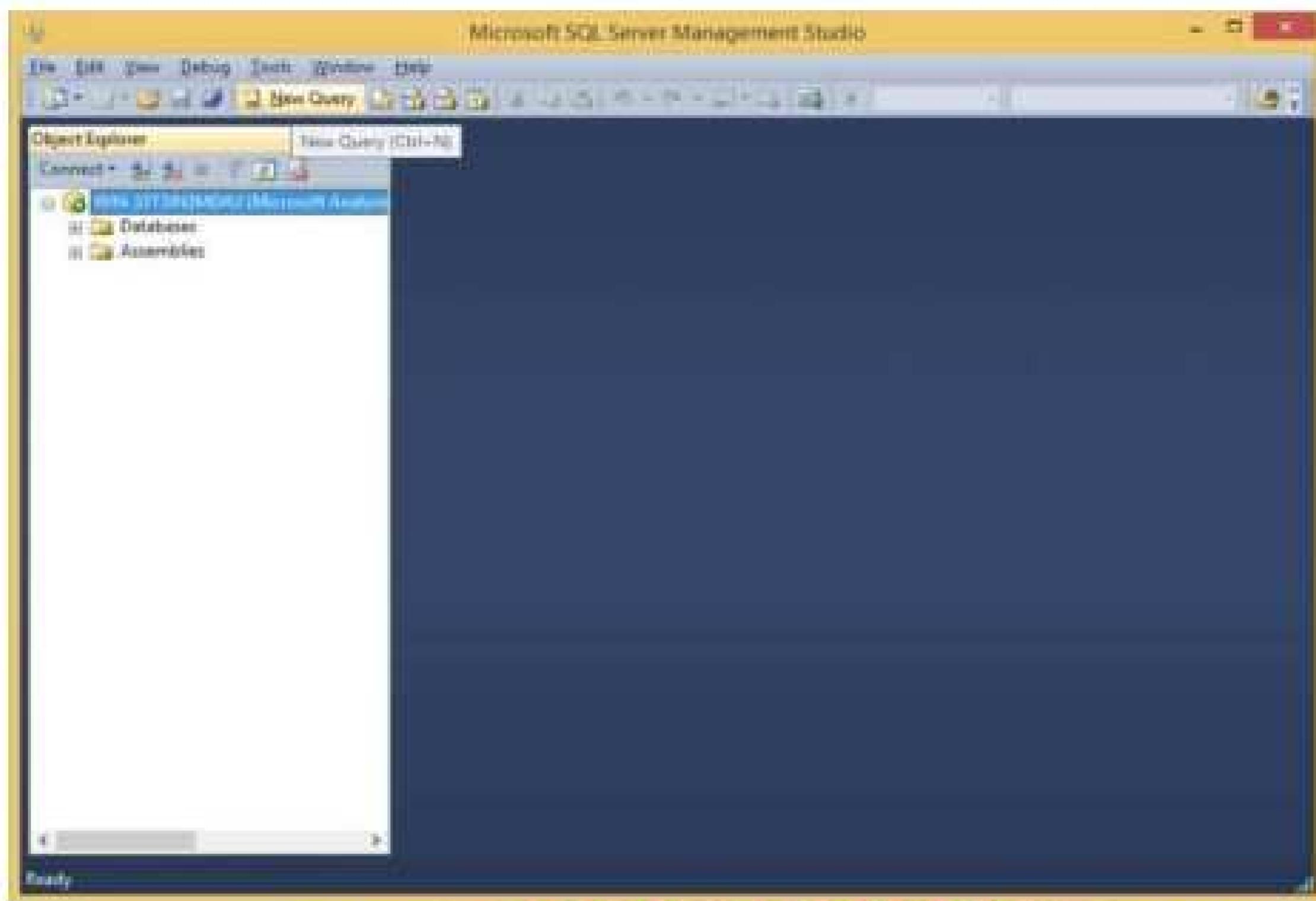
Server Name: (according to base machine)

Click on connect



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

निसिलेन्हे उत्तम सेवाधर्म



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

नियंत्रित करने हें उत्तम सेवाधर्म



MDXQuery1.mds - WIN-ETTINMDAUOLAP (WIN-ETTINMDAU\WINDOWS 8.1) - Microsoft SQL Server Management Studio

File Edit View Insert Object Debug Tools Window Help

MDXQuery1.mds ~ (DW\WINDOWS 8.1) >

Object Explorer > [] > MDXQuery1.mds ~ (DW\WINDOWS 8.1) > Cube: Sales DW

Metadata Functions

Measure Groups:

Sales DW

all Measures

Fact Product Sales

all Dimension

Fact Product Sales Count

Product Actual Cost

Quantity

Sales Invoice Number

Sales Time Alt Key

Sales Total Cost

KPIs

Dim Customer

Dim Date

Dim Product

Dim Sales Person

Dim Store

Dim Time

Messages Results

Sales Time Alt Key

3621639

Query executed successfully

WIN-ETTINMDAU\WINDOWS 8.1 OLAP 05:00:21

Priority: 1 Le: 1 Coll: 0 Ch: 0 PG: 0



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

नियमितीय उत्तम सेवाधर्म



MDXQuery1.mds - WIN-3TSN3MDOLAP\hmv\DATA\MDOLAP\MDX\DETAILS\MDOLAP\WINDOWS 8.1\Microsoft SQL Server Management Studio - 10:30 AM

File Edit View Insert Design Tools Window Help

OLAP

MDXQuery1.mds - [DW\WINDOWS 8.1] X

Code:

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

Measure Group:

Sales DW

Measures

- Fact Product Sales
- Deviation
- Fact Product Sales Count
- Product Actual Cost
- Quantity
- Sales Invoice Number
- Sales Time As Key
- Sales Total Cost

Dim

- KPIs
- Dim Customer
- Dim Date
- Dim Product
- Dim Sales Person
- Dim Store
- Dim Time

Ready

Query executed successfully.

WIN-3TSN3MDOLAP\hmv\DATA\MDOLAP\WINDOWS 8.1\Microsoft SQL Server Management Studio - 10:30 AM

OLAP: 10:30:01



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

निषिद्धत्वे ह उत्तम सेवाधर्म



MDXQuery1.mds - WIN-3T3RNDM4U\OLAP (WIN-3T3RNDM4U\WINDOWS 8.1) - Microsoft SQL Server Management Studio

File Edit View Insert Design Tools Window Help

OLAP

Object Explorer

Connections: 3

MDXQuery1.mds - D4U\WINDOWS 8.1

Cube: Sales DW

Measure Group: Sales DW

Measures:

- + Fact Product Sales
 - + Deviation
 - + Fact Product Sales Count
 - + Product Actual Cost
 - + Quantity
 - + Sales Invoice Number
 - + Sales Time All Key
 - + Sales Total Cost
- + KPIs
- + Dim Customer
- + Dim Date
- + Dim Product
- + Dim Sales Person
- + Dim Store
- + Dim Time

Results

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

100 %

Ready

Query executed successfully.

WIN-3T3RNDM4U\WINDOWS 8.1\OLAP 05:00:01

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

MDXQuery1.mds - WIN-3T3RNDM4U\OLAP (WIN-3T3RNDM4U\WINDOWS 8.1) - Microsoft SQL Server Management Studio

File Edit View Insert Design Tools Window Help

OLAP

Object Explorer

Connections: 3

MDXQuery1.mds - D4U\WINDOWS 8.1

Cube: Sales DW

Measure Group: Sales DW

Measures:

- + Fact Product Sales
 - + Deviation
 - + Fact Product Sales Count
 - + Product Actual Cost
 - + Quantity
 - + Sales Invoice Number
 - + Sales Time All Key
 - + Sales Total Cost
- + KPIs
- + Dim Customer
- + Dim Date
- + Dim Product
- + Dim Sales Person
- + Dim Store
- + Dim Time

Results

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

100 %

Ready

Query executed successfully.

WIN-3T3RNDM4U\WINDOWS 8.1\OLAP 05:00:01



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

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled "MDXQuery1.mds - WIN-3T3INIMDOLAP (Mktv-331-9N3MD4U\WINDOWS 8.1)* - Microsoft SQL Server Management Studio" displays the following MDX query:

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

The results pane shows the following data:

	Sales Total Cost
2013	1231.5
2014	Null
Unknown	Null

At the bottom of the query window, a message says "Query executed successfully".

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

निसिलेन्हे उत्तम संवाधमं



MDXQuery1.mds - WIN-3T3UNMDAOLAP\WIN-3T3UNMDA\WINDOWS & IP - Microsoft SQL Server Management Studio

File Edit View Query Design Tools Window Help

OLAP

MDXQuery1.mds - [SALES DW]

Code

Save DW

Metadata Functions

Measure Group

Sales DW

all Measures

Fact Product Sales

all Dimension

all Fact Product Sales Cost

all Product Actual Cost

all Quantity

all Sales Invoice Number

all Sales Time All Key

all Sales Total Cost

all KPIs

Dim Customer

Dim Date

Date Key

Full Date UK

Month

Month Name

Quarter

Quarter Name

Week Of Month

Query executed successfully

WIN-3T3UNMDA\WIN-3T3UNMDA\WINDOWS & IP OLAP 00:00:01

Ready

1 of 1 Col 16 CR 16 F16

Results

Sales Total Cost	2013
1231.5	



select [Measures].[Sales Total Cost] on columns from [Sales DW]
Where [Dim Date].[Year].[Year].&[2013]

निष्ठानोह उत्तम सेवाधर्म



MDXQuery1.mds - WIN-3T3N9M04OLAP (WIN-3T3N9M04\WINDOW\$) - Microsoft SQL Server Management Studio

File Edit View Insert Design Tools Window Help

OLAP

Object Explorer

Connect to... OLAP MDXQuery1.mds - [Data] [Windows & TV]

Cube: Sales DW

Measure Group: Sales Total Cost

Members Functions

Measure Group: Sales Total Cost

Members

Year

2013

2014

Unknown

Month

Quarter

Week Of Month

Month Name

Full Date UK

Date Key

Dim Data

Dim Customer

KPIs

Sales Total Cost At Key

From [Sales DW]

Where [Dim Date].[Year].<[2013]>

select {[Measures].[Sales Total Cost]} on columns

Results

Sales Total Cost

1231.0

Ready

WIN-3T3N9M04U WIN-3T3N9M04\WINDOW\$ OLAP 00:00:01

LA 1 Ch 1 Ch 1 RD



प्रिंसिपल ने हृतम संवाधम्

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:**
 - o Utilize Excel's Data tab to establish a connection to the data warehouse using options like ODBC, SQL Server, or other database connectors.
 - o Configure the connection to securely access the required tables or views from the warehouse.
- 2. Creating a PivotTable:**
 - o 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.



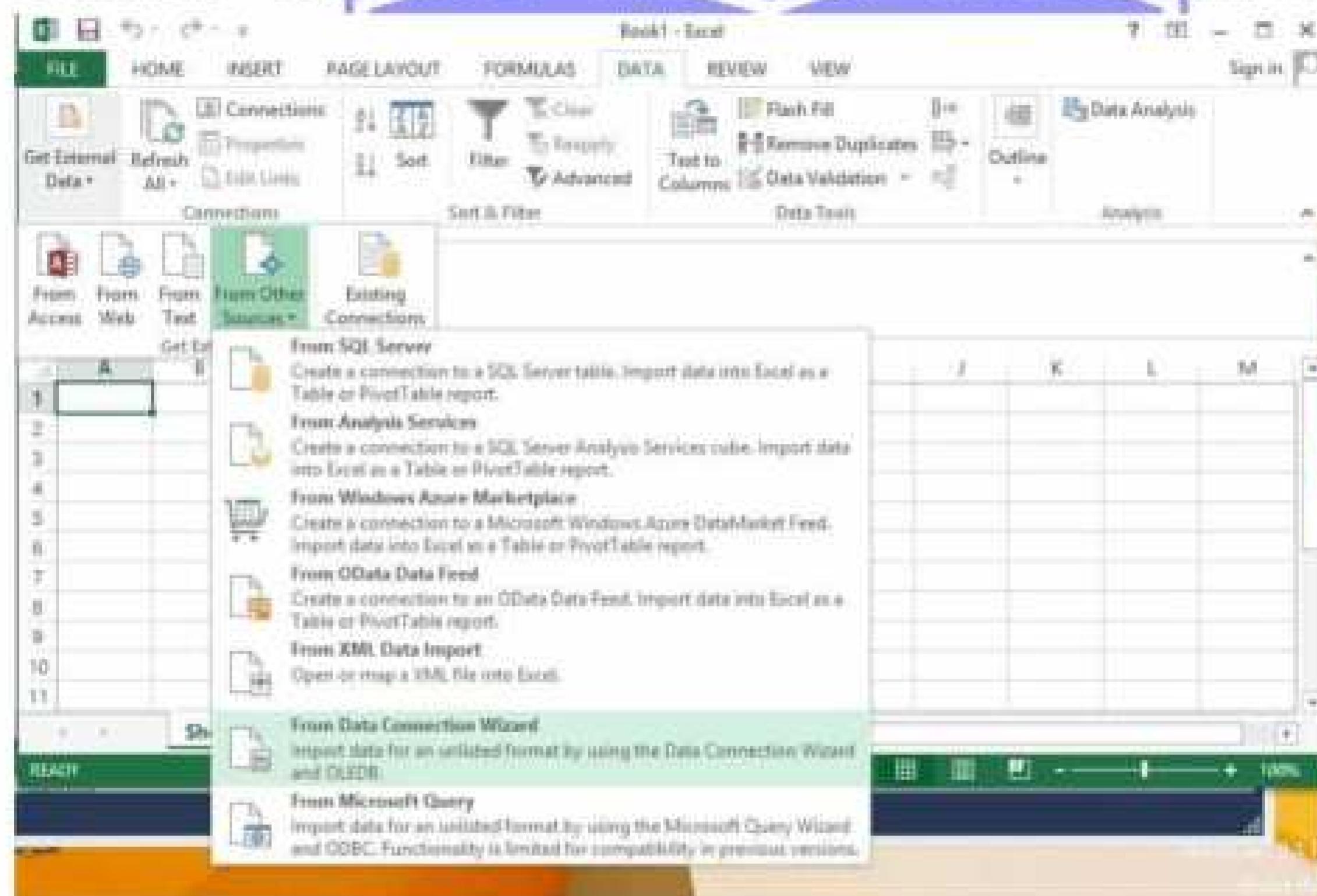
PRACTICAL 5 a

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



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

नियंत्रित करने ही उत्तम सेवाधर्म



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



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

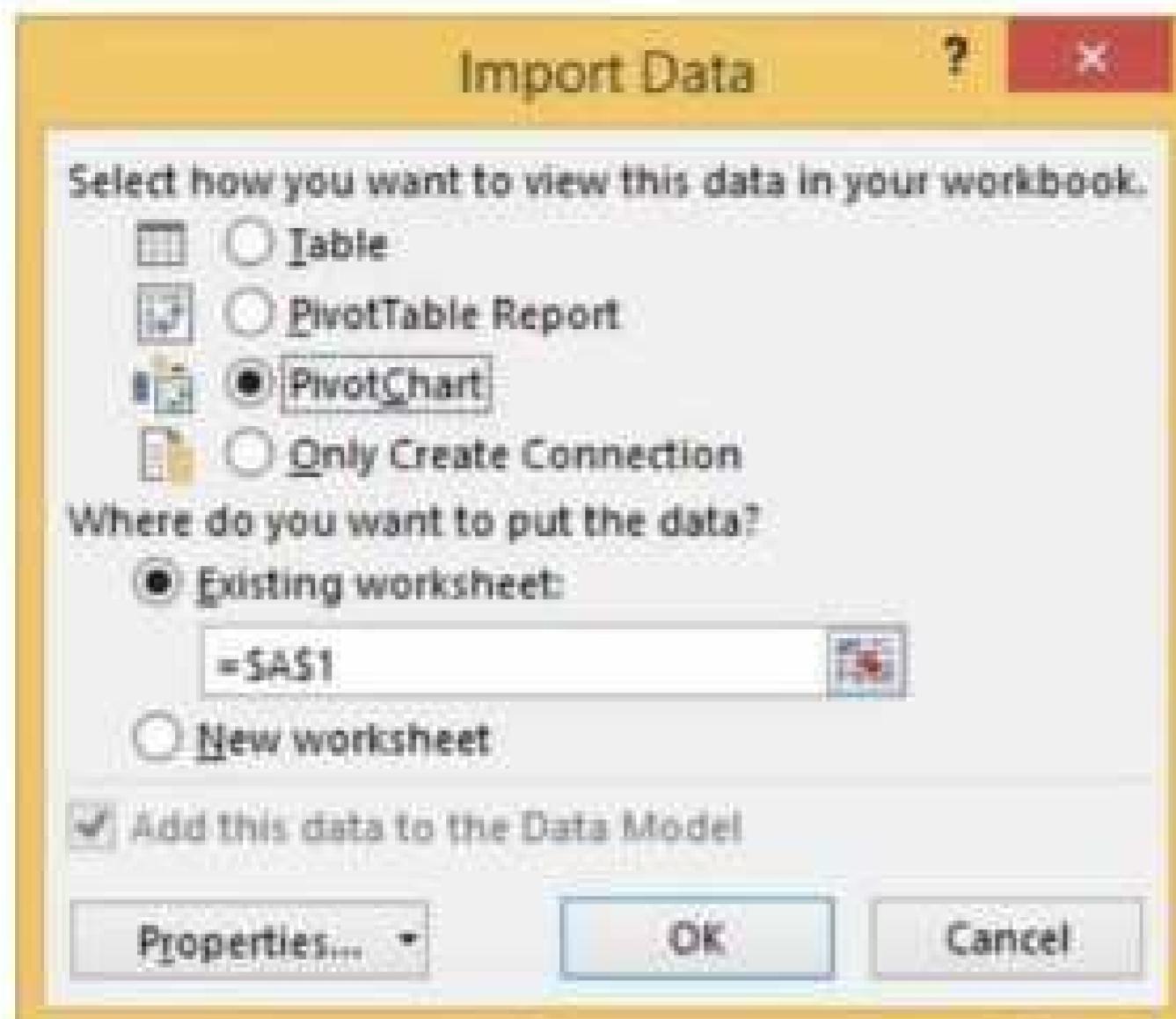


Step 5: In save data connection files browse path and click on Finish



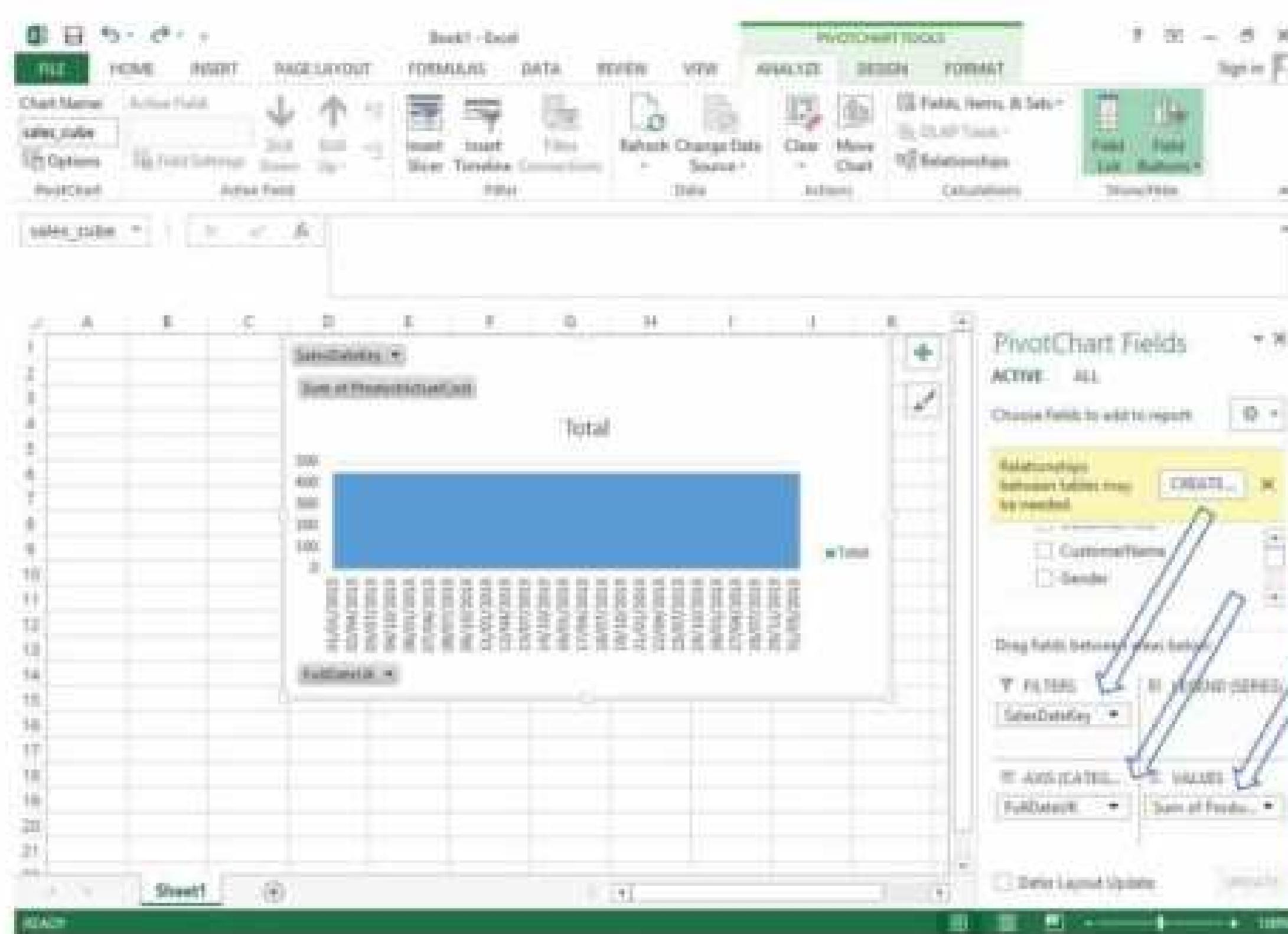


Step 6: In import data select Pivot Chart and click on OK

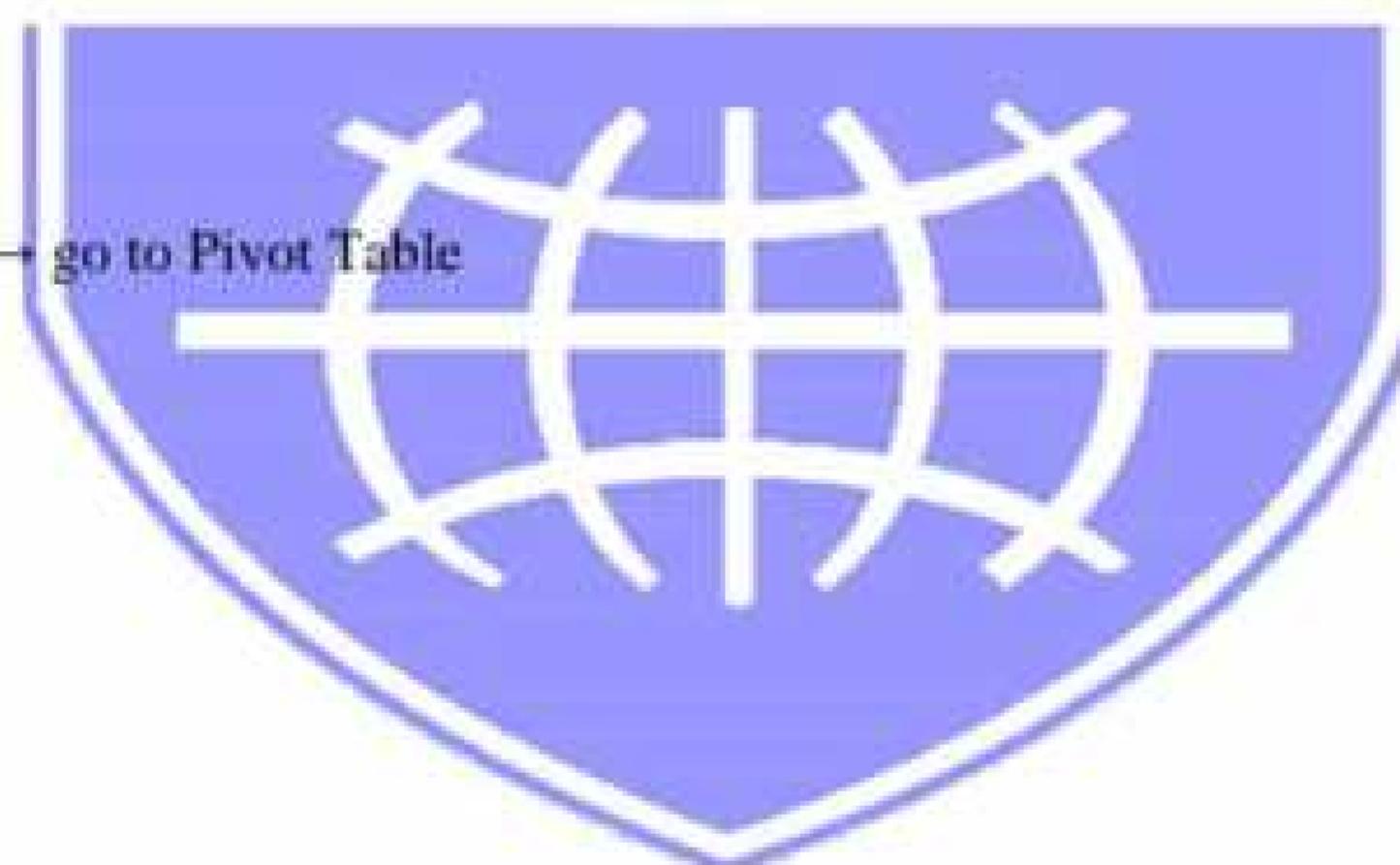


Step 7: In fields put SalesDateKey in filters, FullDateUK in axis and Sum of ProductActualCost in values

नियमित नोह उत्तम सेवाधर्म



Step 8: In Insert Tab → go to Pivot Table



निष्ठानोह उत्तम सेवाधर्म



The screenshot shows a Microsoft Excel spreadsheet. At the top, the ribbon menu is visible with tabs like FILE, HOME, INSERT, PAGE LAYOUT, FORMULAS, DATA, REVIEW, and INFO. Below the ribbon, there's a toolbar with icons for PivotTable, Recommended PivotCharts, PivotTables, PivotCharts, and other data analysis tools. A large PivotTable is displayed in the main area, showing data for various categories like ProductID, ProductName, and Sales. To the right of the PivotTable is a PivotChart, which is a bar chart titled 'Total'. The bars are blue, and the chart has a legend indicating the categories. The overall interface is clean and professional, typical of a college setting.



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



नियमित नोह उत्तम सेवाधर्म



Create PivotTable

Choose the data that you want to analyze

Select a table or 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

Screenshot of Microsoft Excel showing a PivotTable setup.

The PivotTable Fields pane on the right shows:

- ACTIVE: ALL
- Relationships between tables may be needed. **CREATE...**
- ProductKey (selected)
- ProductCategory
- ProductName

The PivotTable Fields pane also includes sections for Filters, Rows, and Values, with various fields listed.

The main worksheet area displays a PivotTable with the following data:

Row Labels	Sum of SalesPersonID	Sum of ProductKey
=Bill Gates	78	56
M	78	56
=Emma Watson	78	56
F	78	56
=Henry Ford	78	56
M	78	56
=Muskan Shukla	78	56
F	78	56
=Richard Thrushin	78	56
M	78	56
Grand Total	78	56

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%]



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.

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.



The screenshot shows a Microsoft Excel spreadsheet titled "Untitled - Excel". The data is organized into two main sections:

	Total Books	% sold for highest price
	100	60

	No of Books	Unit Profit
Highest	60	30
Lowest	40	20
Total Price		3600

The "Data" tab is selected in the ribbon. A tooltip for the "What-If Analysis" button is visible, stating: "Create different groups of values or scenarios, and switch between them."

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

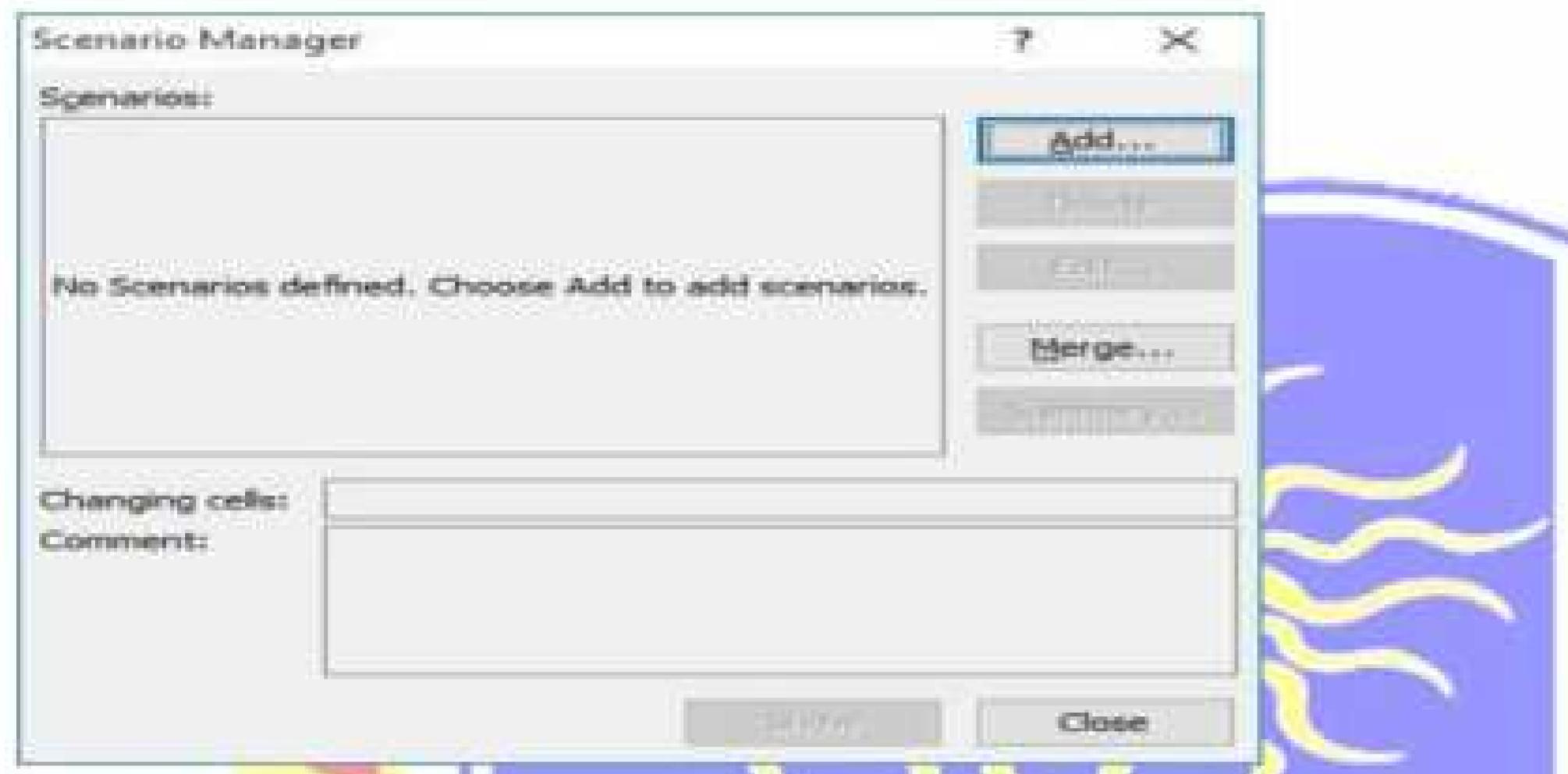
The screenshot shows the same Excel spreadsheet as above, but with the "Scenario Manager" dialog box open. The dialog box has the following fields:

- Scenarios: [Empty]
- Data Table: [Empty]
- Description: Create different groups of values or scenarios, and switch between them.

The "Data" tab is still selected in the ribbon.

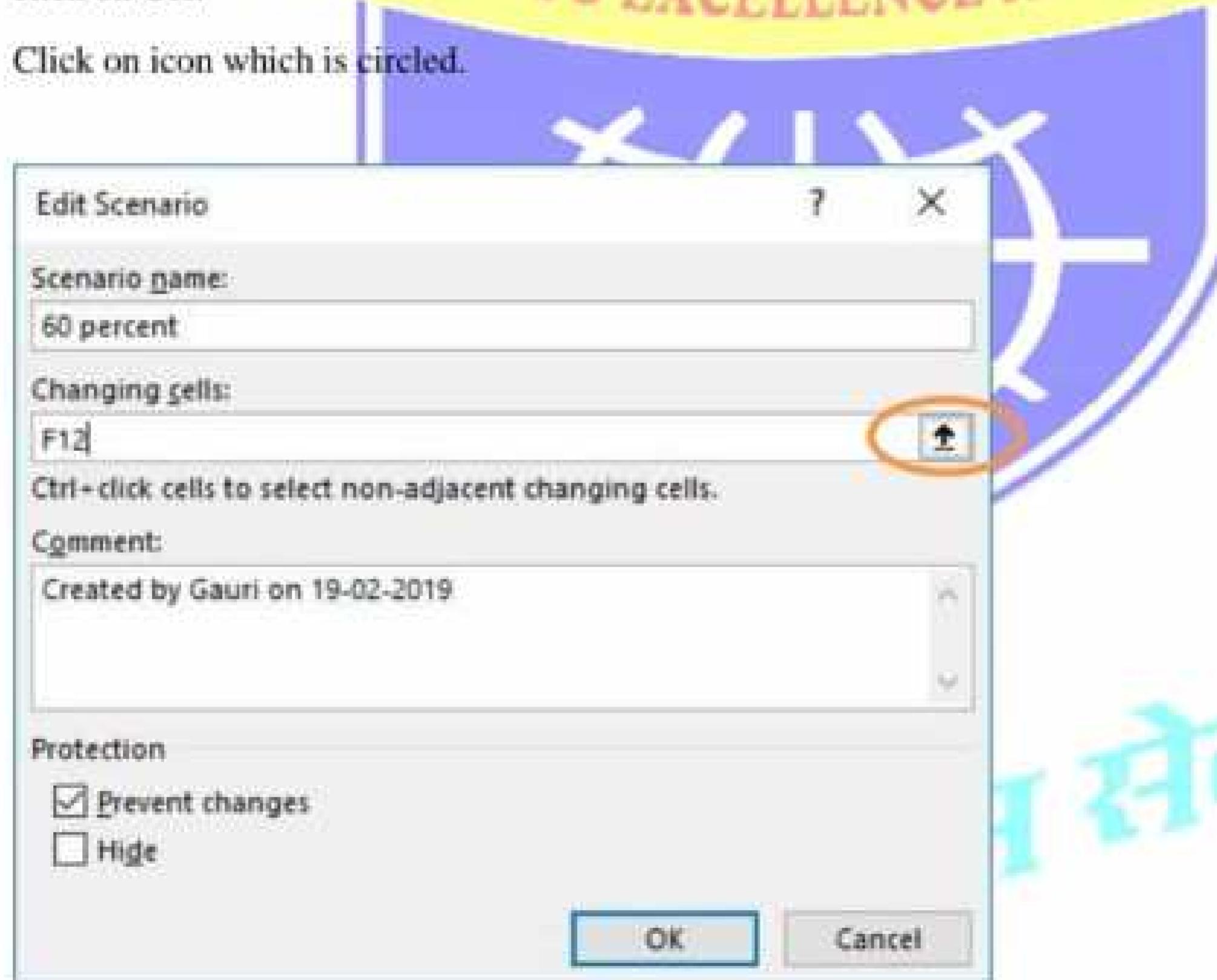


The Scenario Manager Dialog box appears. Step 3: Add a scenario by clicking on Add.



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.



Select F10 cell.



Add Scenario - Changing cells:

?

X

SFS10



Click back on the icon again and then click OK

Edit Scenario

Scenario name:
60 percent

Changing cells:
SFS10

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

Comment:
Created by Gauri on 19-02-2019

Protection

Prevent changes
 Hide

OK Cancel



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

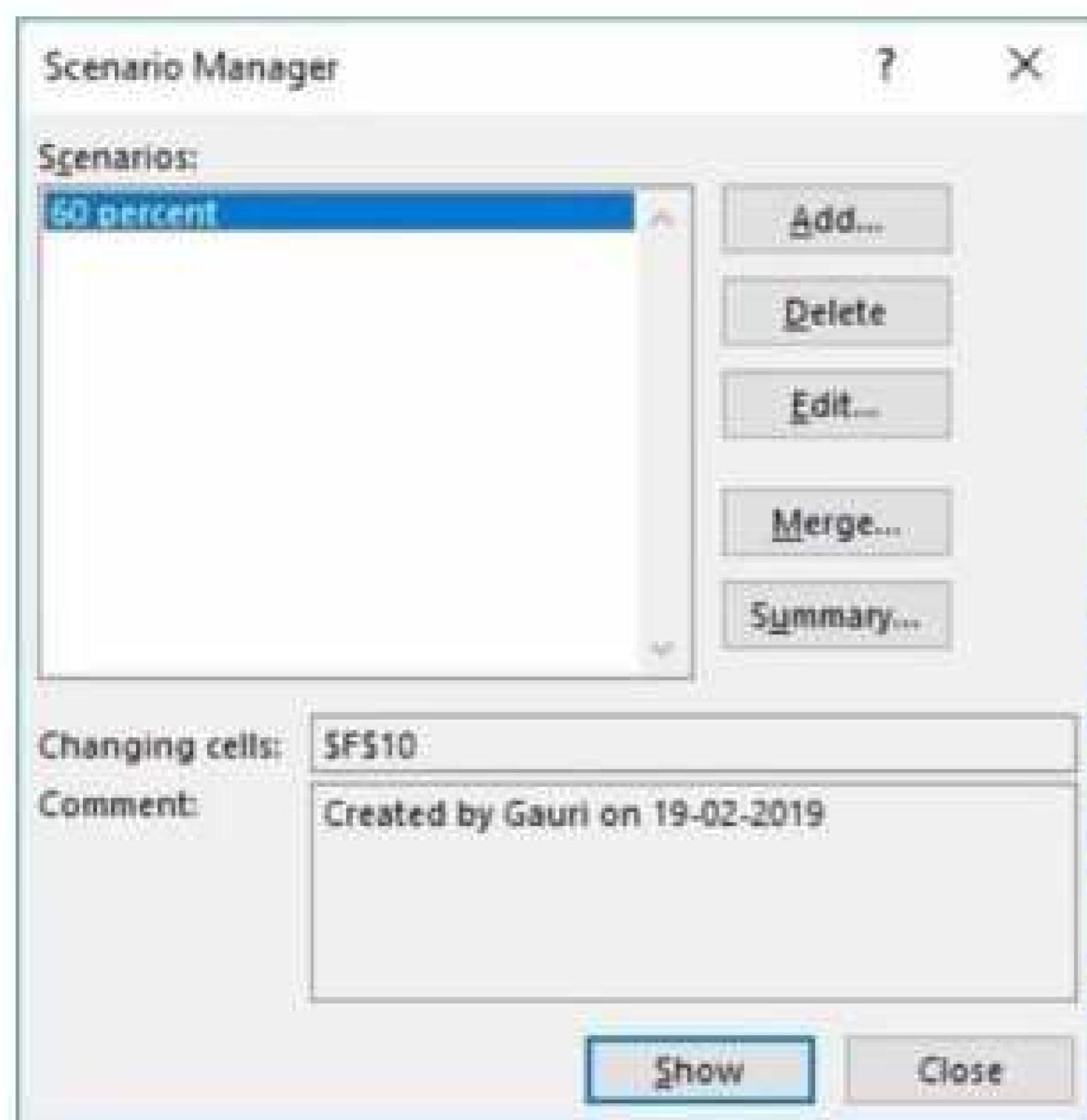
Scenario Values

Enter values for each of the changing cells.

1: SFS10 0.6

Add OK Cancel

नियमितीय उत्तम सेवाधर्म



Step 6: To apply scenarios click on Show



निष्ठानेह उत्तम सेवाधर्म



Screenshot of Microsoft Excel showing a scenario manager dialog box. The worksheet contains data for book sales and profit.

total books	% sold for highest price
100	60

highest lowest	no of books	unit profit
60	60	5.5
40	40	3.5
		total price
		100

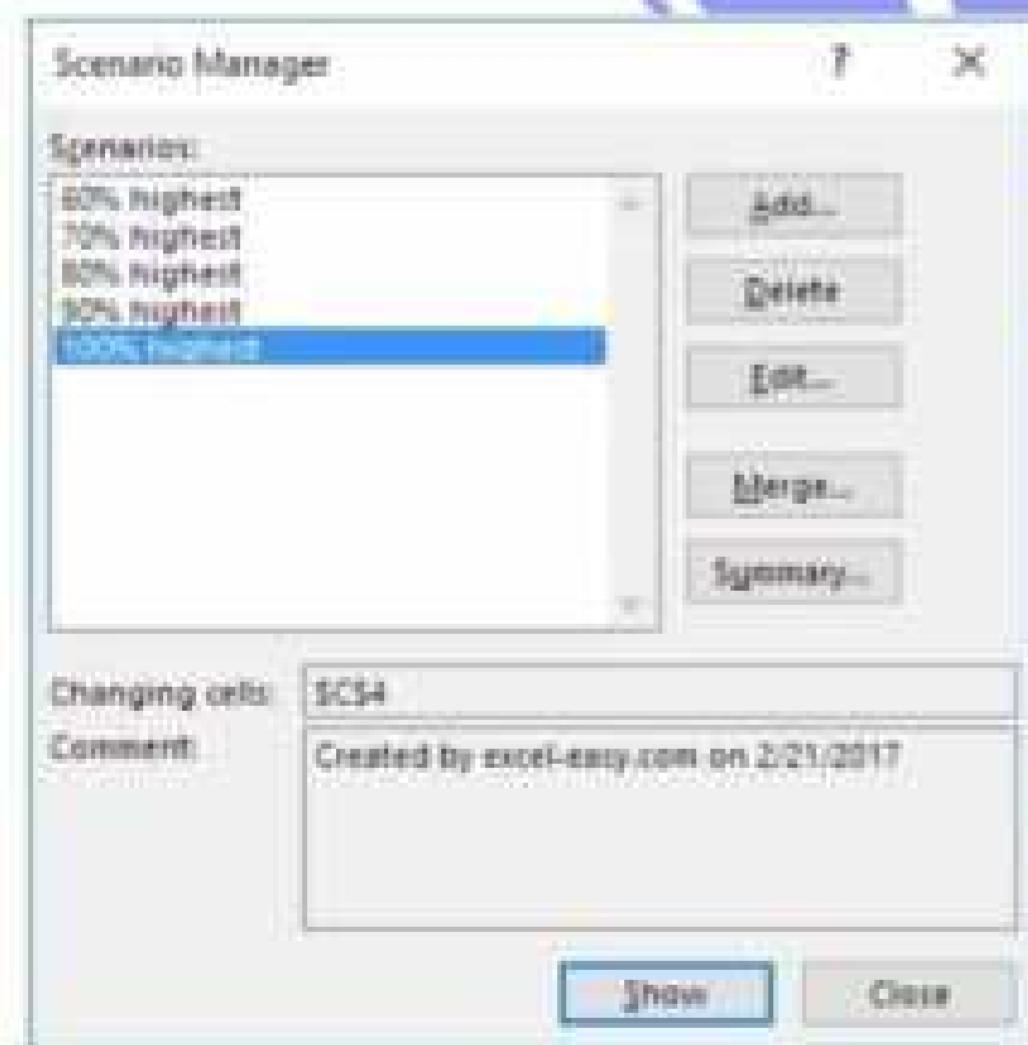
The Scenario Manager dialog box shows the following scenarios:

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

Buttons in the dialog box include Add, Delete, Edit, Merge, and Summary. The "Summary" button is highlighted.

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

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



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**
 - o **Collection & Cleaning:** Gather a representative dataset and clean it by handling missing values, removing outliers, and correcting errors.
 - o **Feature Selection & Engineering:** Identify key features that influence the classification outcome and transform raw data into a suitable format for analysis.
2. **Dataset Partitioning**
 - o **Training Set:** Allocate a portion of the data to train the classification model, ensuring it learns the underlying patterns.
 - o **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	985.0	882.8	1071.0



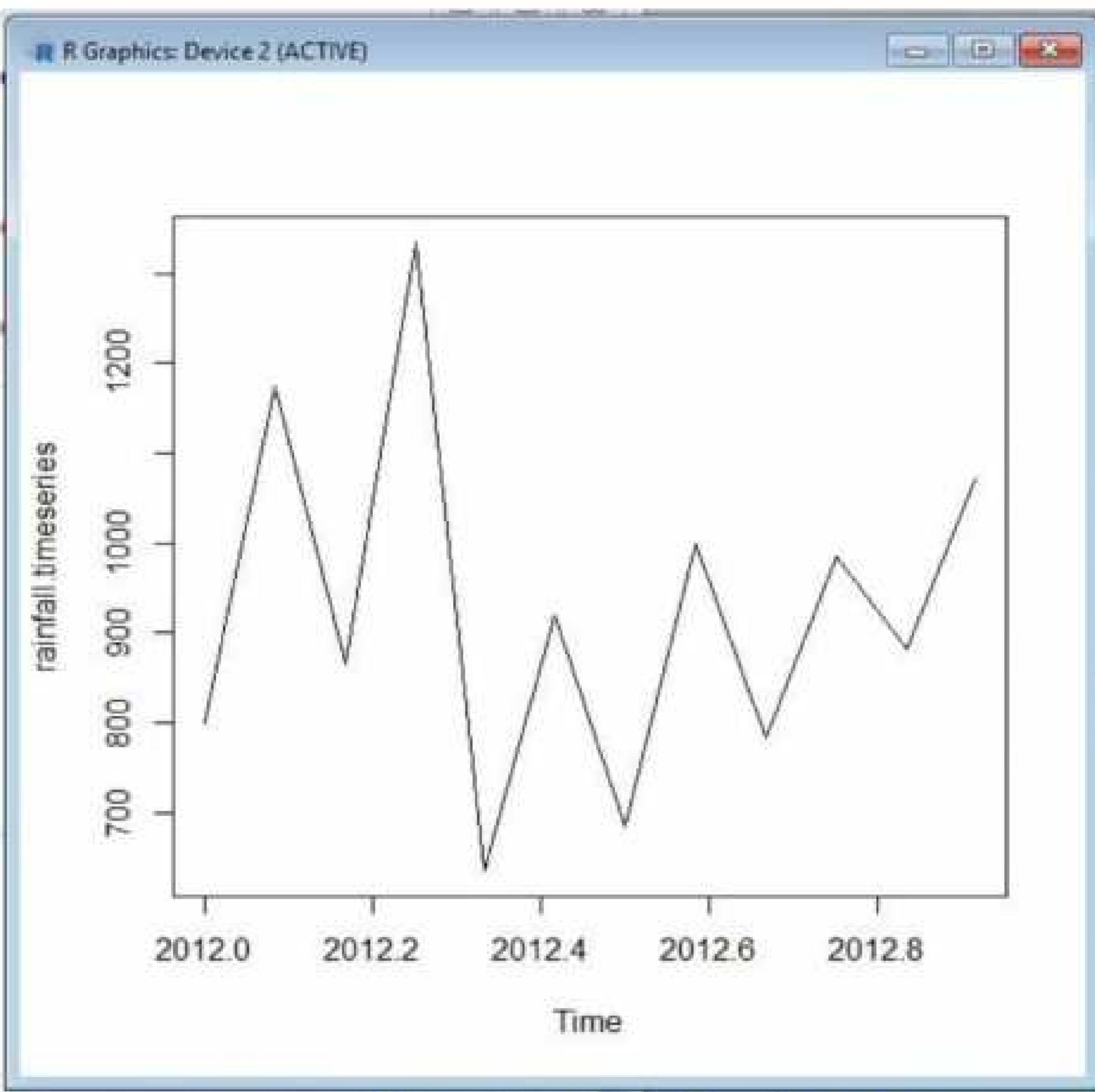
File Edit View Insert Package Window Help



Type 'q()' to quit R.

```
> # Get the data points in form of a R vector.  
> rainfall <- c(799,1174.0,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.0,1071)  
> # 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 1174.0 865.1 1334.6 635.4 918.5 685.5 998.6 784.2 985.0  
Nov Dec  
2012 882.0 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)  
>
```





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**
 - o **Data Collection & Cleaning:** Gather a comprehensive dataset and clean it by removing noise, handling missing values, and eliminating outliers.
 - o **Normalization & Scaling:** Standardize or normalize features to ensure that no single feature dominates the clustering due to scale differences.
2. **Feature Selection and Extraction**
 - o **Feature Selection:** Identify key variables that capture the essence of the data, ensuring that the chosen features enhance the distinction between clusters.
 - o **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
```

```
newins <- iris
```

```
newirisSSpecies <- NULL
```

```
(kc <- kmeans(newiris.3))
```

K-means clustering with 3 clusters of sizes 21, 26, 33

©1997 by Prentice-Hall

	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	5.175750	3.634282	1.472727	0.2737273

Clustering vector:

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"       "proto"        "withiness"      "tot.withinss"  
[6] "betweeness"   "size"          "iter"         "ifault"        ".
```

#Compare the Species label with the clustering result

```
table(iris$Species,iris$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)
```



SHRI G.P.M. DEGREE COLLEGE
Department of Computer

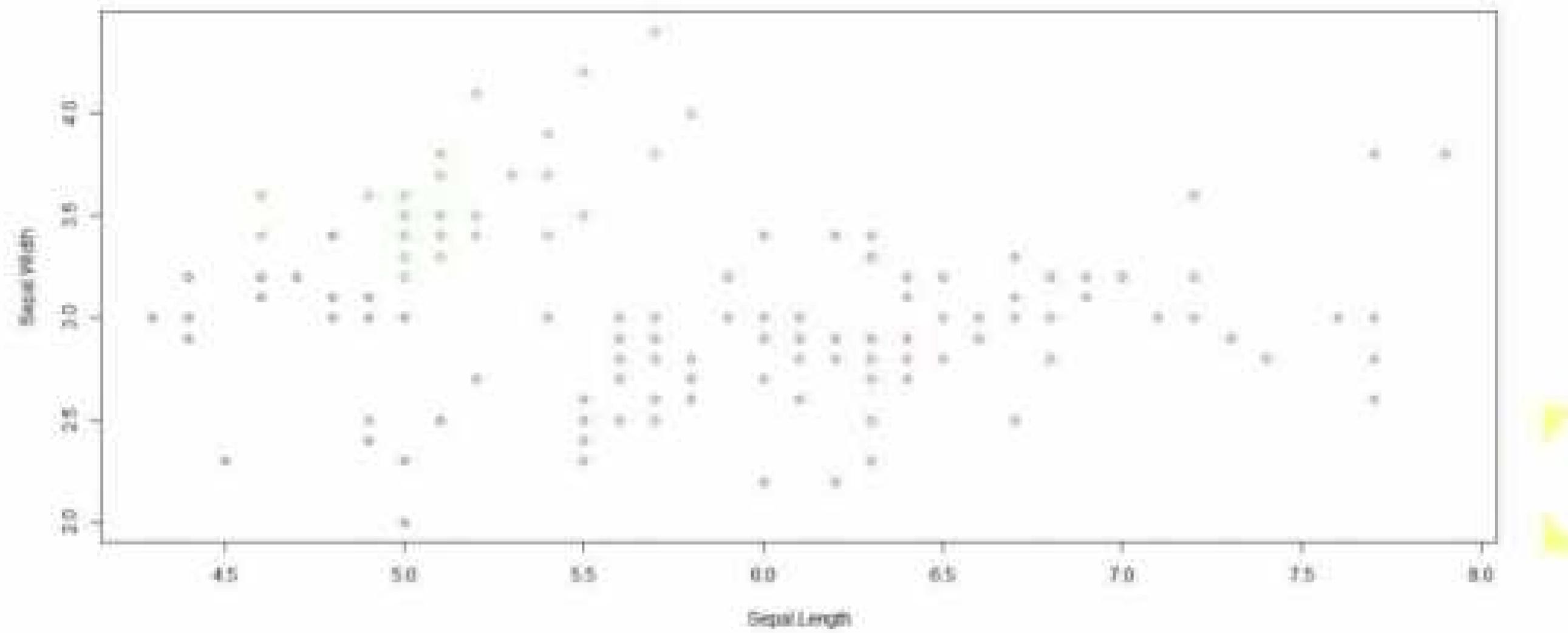
Wise... Dynamic... Innovative... Motivational... Professionalism

(1) Run (ctrl+F5) - (2) Capture Device 2 (ACTIVE)

(3) File - Home - Page - Window



Reset Viewport (F5)



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.
```

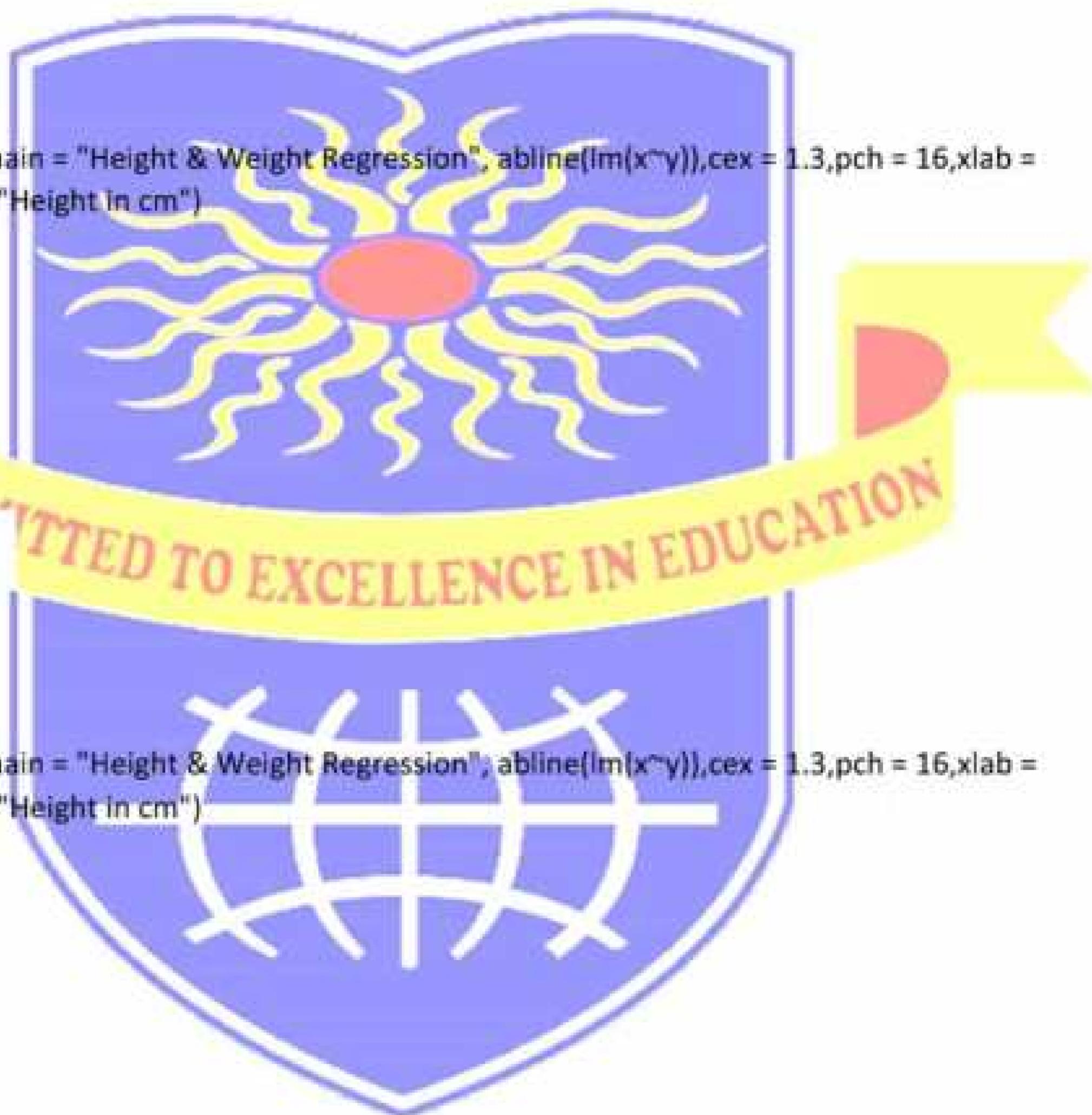
```
dev.off()
```

```
null device  
1
```

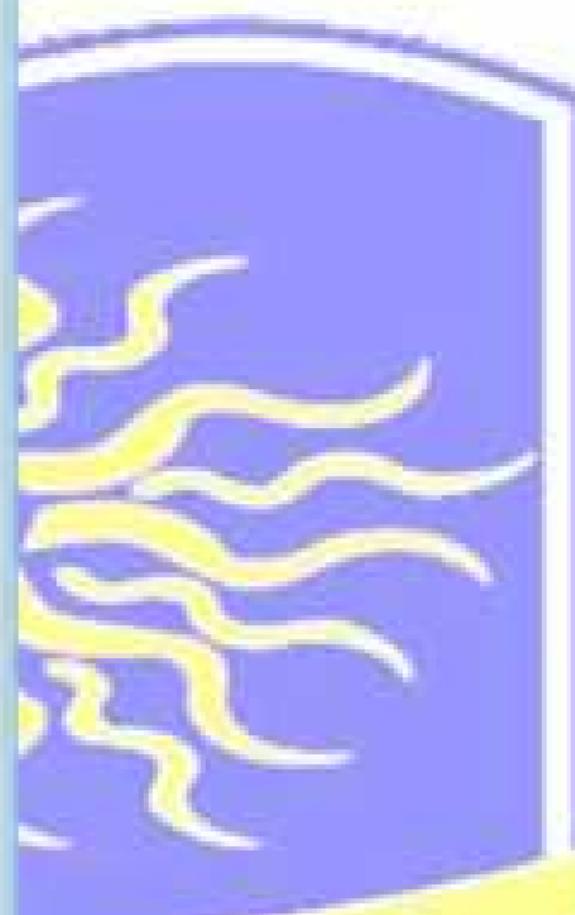
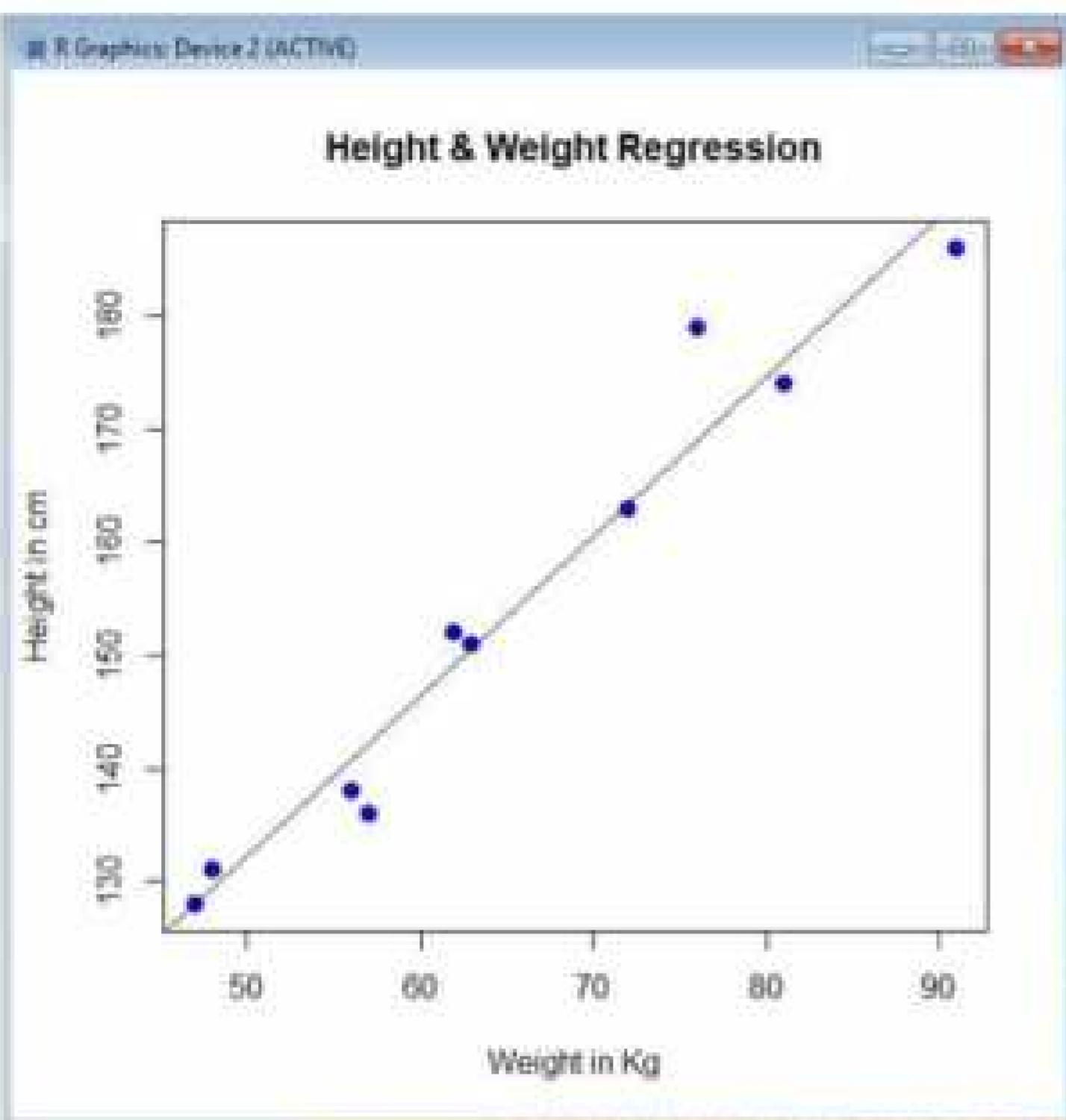
```
# Plot the chart.
```

```
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:



निष्ठानोह उत्तम सेवाधर्म



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/Week5%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 24 ...  
 $ 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 1 0 0 1 0 ...  
> table(quality$PoorCare)
```



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.zip'
```

```
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_packages

> 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 FALSE TRUE TRUE
```

```
[28] TRUE TRUE 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  
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 FALSE 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_packages
```

```
> 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

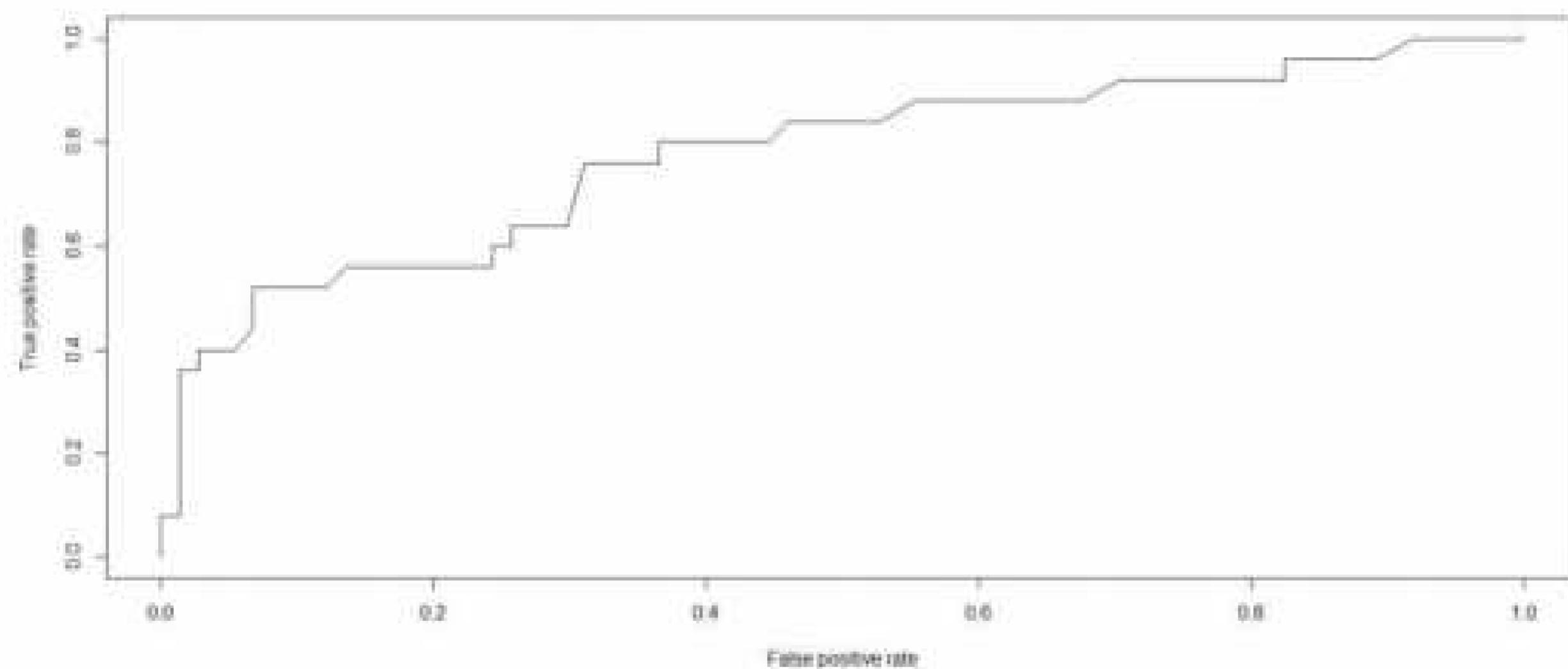
2: package 'gplots' was built under R version 3.5.2

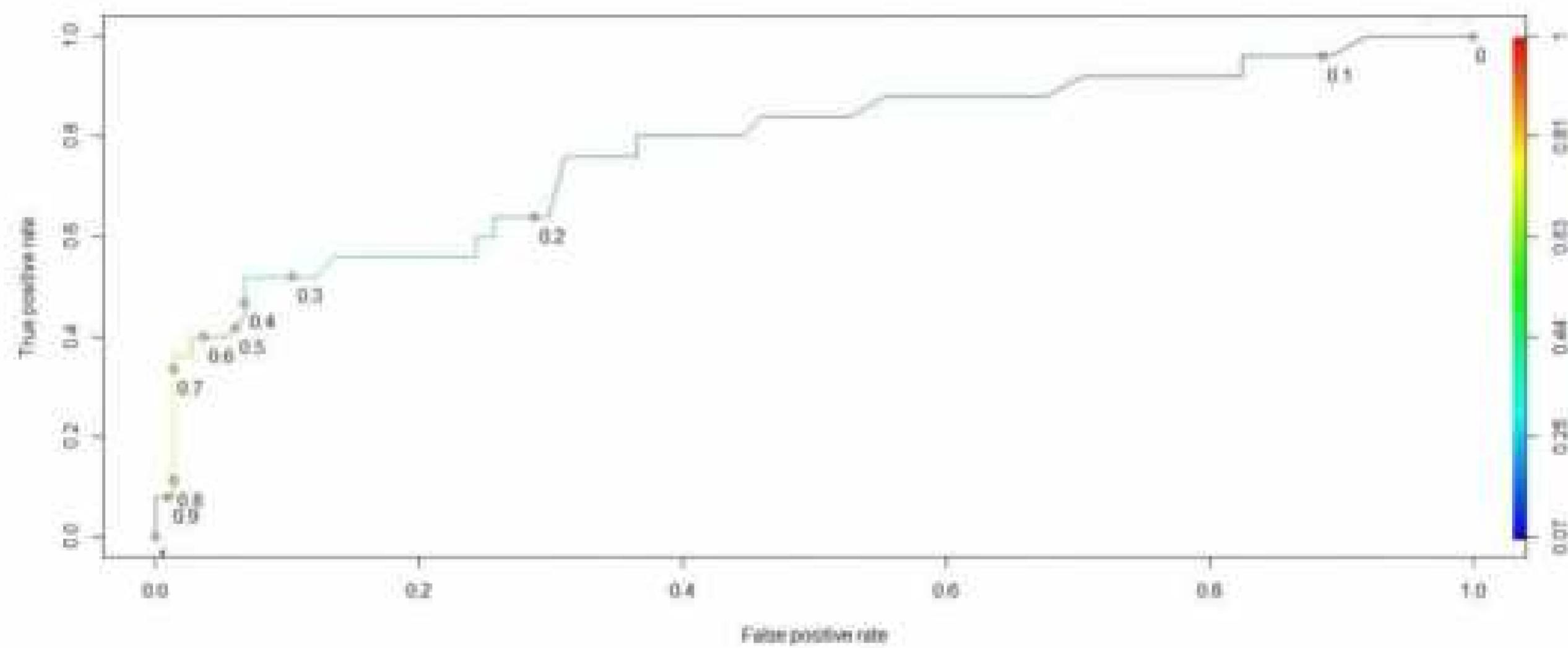
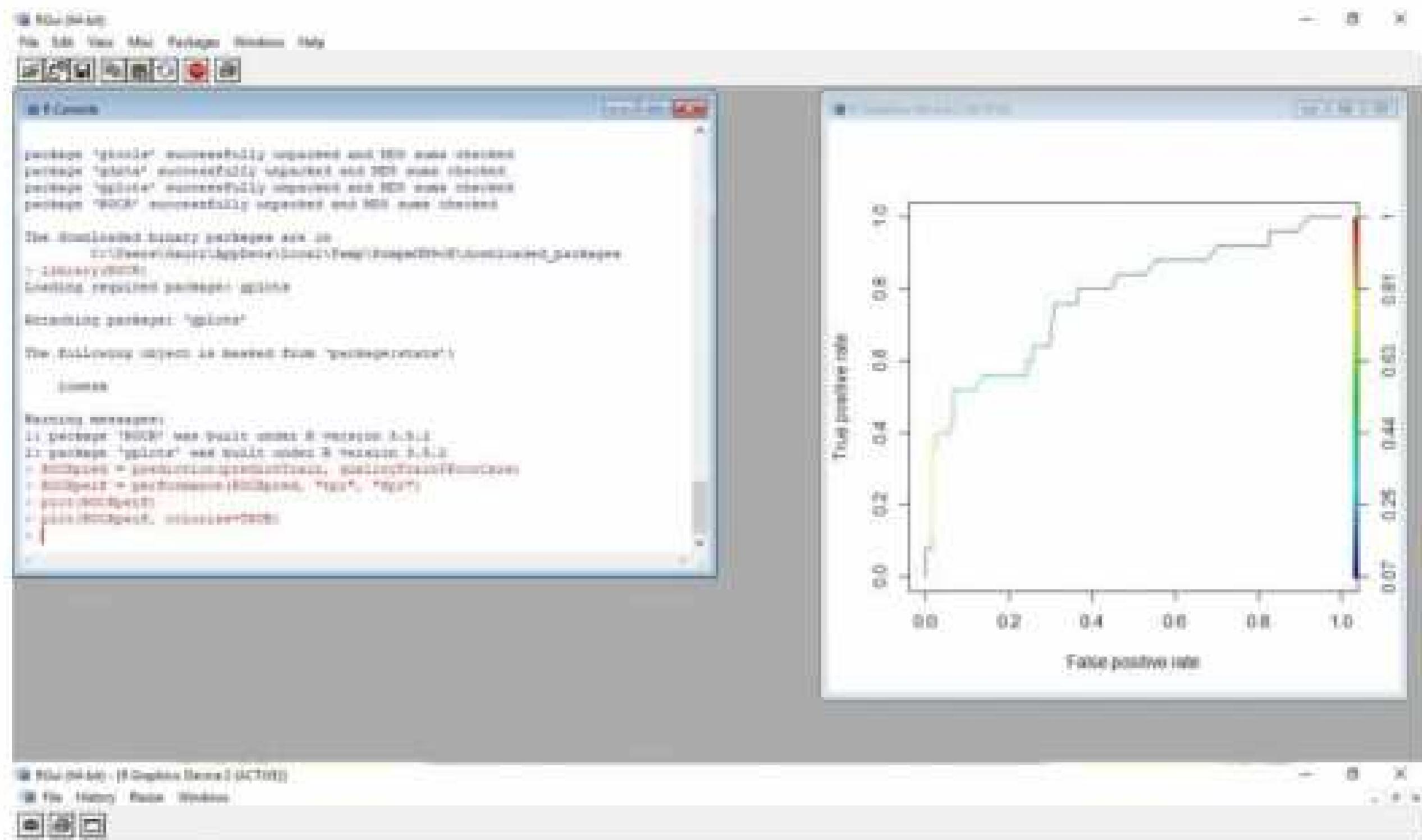
```
> ROCRpred = prediction(predictTrain, qualityTrain$PoorCare)
> ROCRperf = performance(ROCRpred, "tpr", "fpr")
> plot(ROCRperf)
> plot(ROCRperf, colorize=TRUE)
> plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))
>
```

(0) File (ctrl+F10) - (Properties Device 2 (ACTIVE))

(1) File History Force Windows

□ □ □





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%]
-----------------------	---------------------------	-------------------------------------	-----------------------------------