



WHERESCAPE RED TERADATA TUTORIALS

6.8.5.0

WhereScape RED Teradata Tutorials

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of agreement.

Copyright Notice

Copyright © 2002-2016 WhereScape Software Limited. All rights reserved. This document may be redistributed in its entirety and in this electronic or printed form only without permission; all other uses of this document and the information it contains require the explicit written permission of WhereScape Software limited.

Due to continued product development this information may change without notice. WhereScape Software Limited does not warrant that this document is error-free.

Trademarks

WhereScape and WhereScape RED are trademarks or registered trademarks of WhereScape Software Limited. Other brands or product names are trademarks or registered trademarks of their respective companies.

WhereScape USA, Inc

1915 NW AmberGlen Parkway
Suite 400, Beaverton
Oregon 97006
United States
T: 503-466-3979
F: 503-466-3978

WhereScape Limited

P.O.Box 56569, Auckland 1446
12-16 Tapora Street
Quay Park
Auckland 1010, New Zealand
T: +64-9-358-5678
T: US toll free 1-877-237-3980
F: +64-9-358-5679

WhereScape Europe

Reading Enterprise Centre
Earley Gate, Whiteknights Rd
Reading RG6 6BU
United Kingdom
T: +44-118-914-4509
F: +44-118-914-4508

WhereScape Asia Pte. Ltd

300 Tampines Avenue 5
#09-02 Singapore 529653
T: +65-6679-5728

CONTENTS

Tutorial Overview	1
<hr/>	
The First Step	2
Logging In	3
Tutorial 1 - Basic Star Schema Fact Table	4
<hr/>	
1.1 Purpose and Roadmap	5
1.1.1 Prerequisite Actions	8
1.2 Adding an Odbc Data Source	9
1.3 Creating a Connection	13
1.4 Loading Source Tables	20
1.4.1 Loading Source Tables using Schemas	26
1.5 Building Dimensions	31
1.6 Creating Dimension Views	42
1.7 Defining the Staging Table	48
1.8 Including Dimension Links	56
1.9 Creating a Fact Table	66
1.10 Switching to Diagrammatic View	72
1.11 Producing Documentation	77

CHAPTER 1

TUTORIAL OVERVIEW

In Tutorial 1 we will create a basic Star Schema Fact Table.
Additional Tutorials may be available on the WhereScape website.

IN THIS CHAPTER

The First Step.....	2
Logging In.....	3

THE FIRST STEP

The first step

To get started you need to follow the steps in the **WhereScape Installation and Administration Guide** to create the required environment. The basic steps in this process are:

The basic steps in this process are:

- 1 Install the WhereScape product suite.
- 2 Create a database user under which to load the WhereScape metadata repository.
- 3 Install the WhereScape metadata repository.

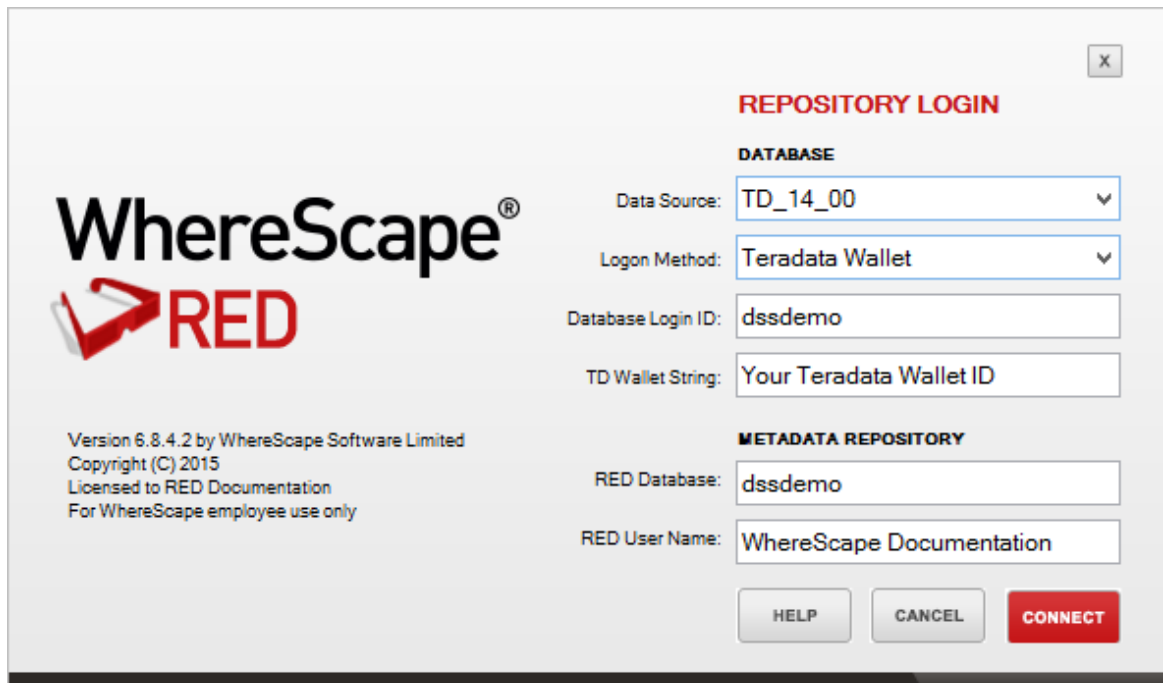
You are now ready to **log on** (see "**Logging In**" on page 3) to the repository you have created.

LOGGING IN

Having completed the first step, and using WhereScape RED, you can now log on to the repository you have created .

To log in:

- 1 Click WhereScape RED from the Start menu. The Access Control screen displays. See sample screen below:



The screenshot shows the 'REPOSITORY LOGIN' dialog box. On the left is the WhereScape RED logo and version information: 'Version 6.8.4.2 by WhereScape Software Limited Copyright (C) 2015 Licensed to RED Documentation For WhereScape employee use only'. On the right, there are two sections: 'DATABASE' and 'METADATA REPOSITORY'. The 'DATABASE' section includes a 'Data Source' dropdown menu set to 'TD_14_00', a 'Logon Method' dropdown menu set to 'Teradata Wallet', a 'Database Login ID' text box containing 'dssdemo', and a 'TD Wallet String' text box containing 'Your Teradata Wallet ID'. The 'METADATA REPOSITORY' section includes a 'RED Database' text box containing 'dssdemo' and a 'RED User Name' text box containing 'WhereScape Documentation'. At the bottom are three buttons: 'HELP', 'CANCEL', and 'CONNECT'.

- 2 The **Data Source** is the ODBC connection for your database. This connection must have been established prior to logon. Refer to the Installation and Administrator Guide if the connection doesn't exist.
- 3 The **Logon Method** is the Method chosen to login. Select either DB User/Password or Teradata Wallet method from the drop-down menu.
- 4 The **Database Login ID** is the User Name where the user has been granted access to the user under which the metadata repository has been loaded.
- 5 The **TD Wallet String** is the string replacing the user name and password for your connection. Teradata TD Wallet is a Teradata product part of the TTU (Tools and Utilities). Refer to Teradata documentation if you don't have a TD Wallet created already.
- 6 The **RED Database** is the User Name for the metadata repository.
- 7 The **RED User Name** is the name that will be associated with any procedures, tables, etc, and scheduled jobs that are created within RED. Normally this would be your full name.
- 8 Click **OK**. The Builder screen displays.

CHAPTER 2

TUTORIAL 1 - BASIC STAR SCHEMA FACT TABLE

IN THIS CHAPTER

1.1 Purpose and Roadmap.....	5
1.2 Adding an Odbc Data Source.....	9
1.3 Creating a Connection	13
1.4 Loading Source Tables	20
1.5 Building Dimensions	31
1.6 Creating Dimension Views	42
1.7 Defining the Staging Table.....	48
1.8 Including Dimension Links	56
1.9 Creating a Fact Table.....	66
1.10 Switching to Diagrammatic View	72
1.11 Producing Documentation	77

1.1 PURPOSE AND ROADMAP

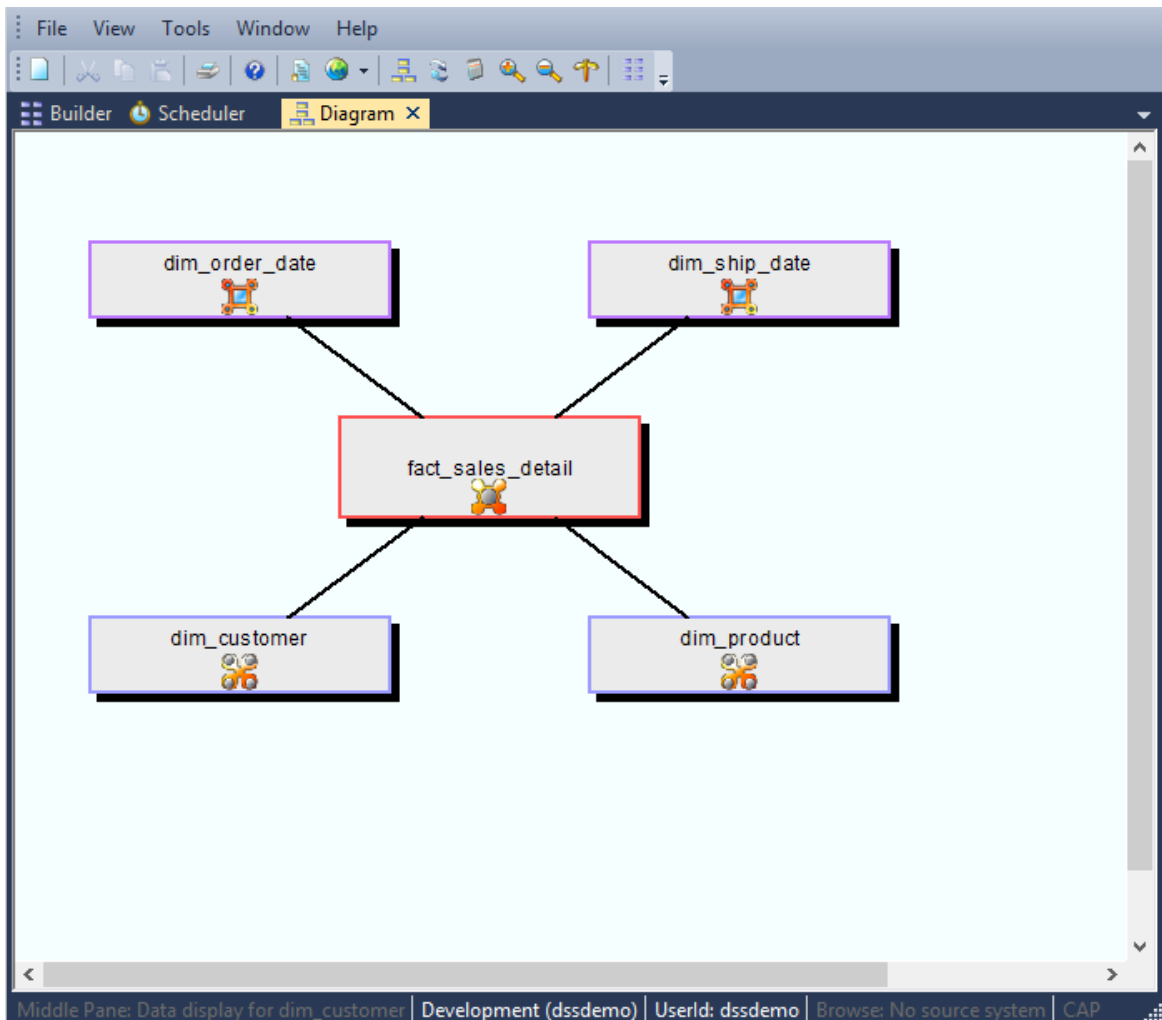
Purpose

This tutorial is designed to introduce you to basic development using WhereScape RED. At the end of the tutorial you will have built a simple dimensional analysis area of a data warehouse, with identity column surrogate keys.

Note: Surrogate keys using identity columns are not always the recommended approach in Teradata. This first tutorial uses them for simplicity. It is important to note that WhereScape RED can be used with and without surrogate keys and when used with surrogate keys, they can be derived from identity columns or using any other user defined approach.

The tutorial will build the star-schema shown below. This star-schema comprises a central fact table, fact_sales_detail, joined to four dimension tables. Data will be loaded from tables in another Teradata database.

In the process of creating this star-schema you will learn to create load, stage, fact and dimension tables. You will also see how data flows from the data source, through the different tables enroute to its fact or dimension table destination.



Tutorial Roadmap

This tutorial works through a number of steps. These steps and the relevant section within the chapter are summarized below to assist in guiding you through the tutorial.

Step in Tutorial	Section
Create a new odbc data source for the RED tutorial source system	Adding an Odbc Data Source
Create a new RED connection object for the RED tutorial source system	Creating a Connection
Create and load the load tables for <ul style="list-style-type: none"> • Customer • Product • Order_line • Order_header 	Loading Source Tables

Step in Tutorial	Section
Create (and update from load tables) the following dimension tables <ul style="list-style-type: none"> • Dim_customer • Dim_product 	Building Dimensions
Create dimensions for <ul style="list-style-type: none"> • Dim_order_date • Dim_ship_date These are views on the dim_date table	Creating Dimension Views
Create the stage_sales_detail table <ul style="list-style-type: none"> • Create stage table using columns from load_order_line and load_order_header • Specify join condition • Include links to the following dimensions (dim_customer, dim_product, dim_order_date, dim_ship_date) 	Defining the Staging Table Including Dimension Links
Create the fact_sales_detail table	Creating a Fact Table
View the WhereScape generated documentation	Switching to Diagrammatic View Producing Documentation

1.1.1 PREREQUISITE ACTIONS

Before commencing this tutorial, please perform the following tasks:

- 1 Install RED metadata
- 2 Install Tutorial data

For more information, refer to the **WhereScape Installation and Administration Guide**.

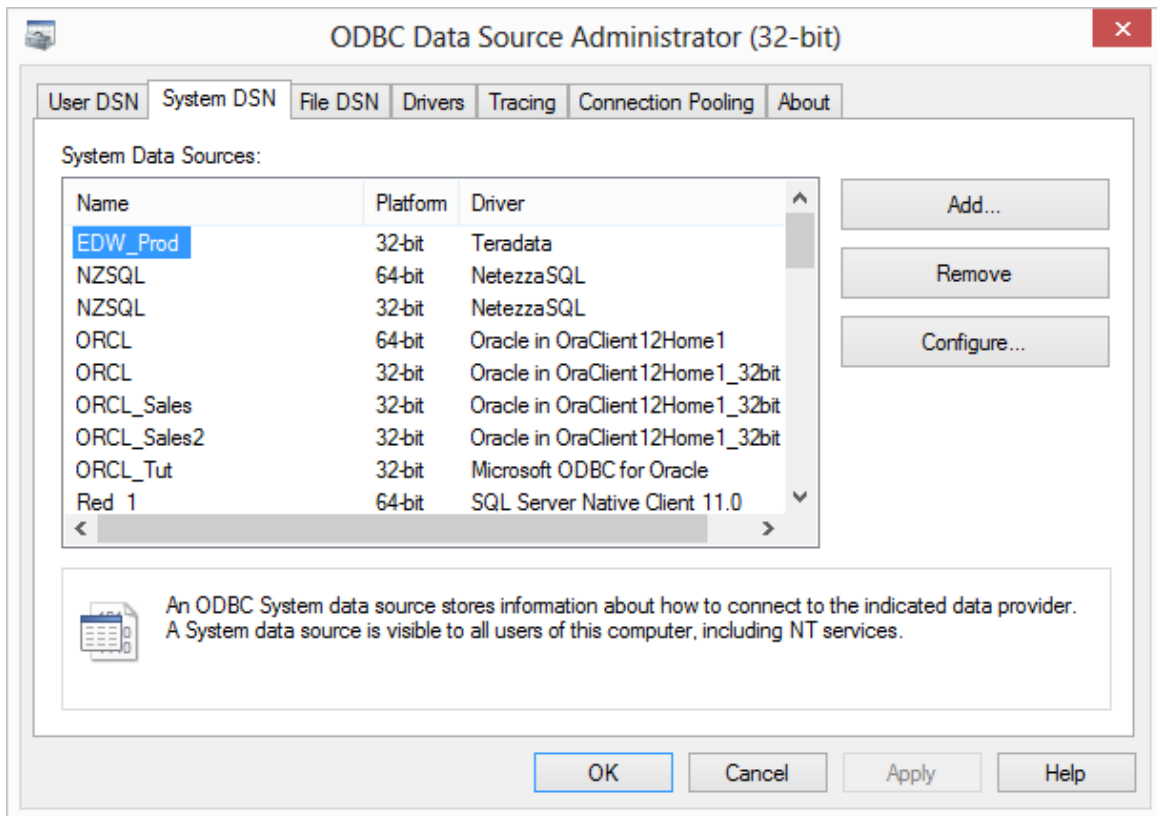
1.2 ADDING AN ODBC DATA SOURCE

We need to add an Odbc DSN called TutorialTDAT, which will be used to connect to the Tutorial Source data.

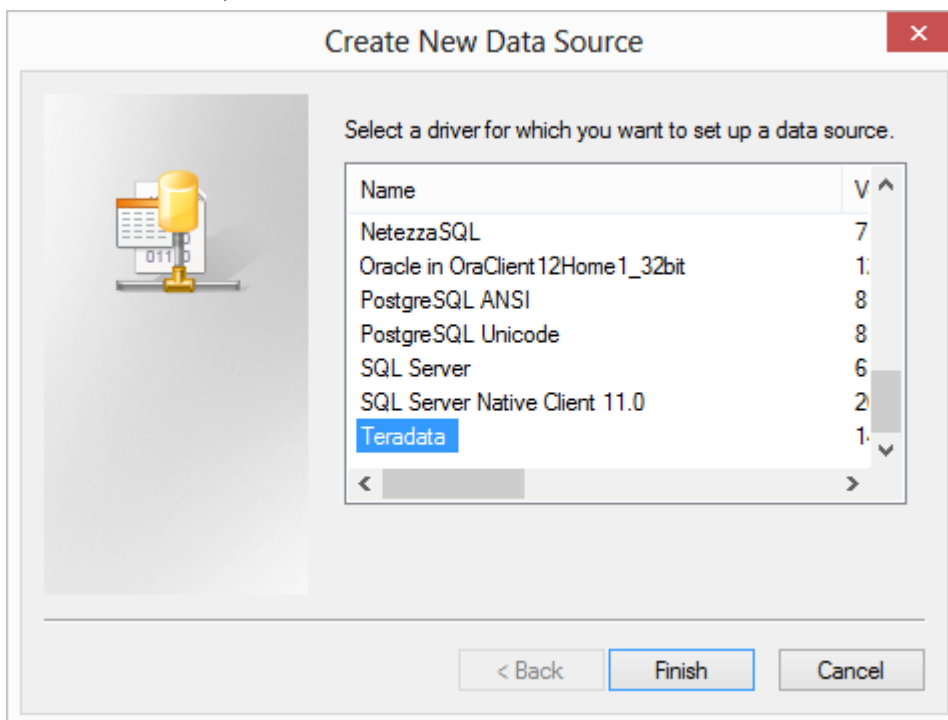
- 1 Open **WhereScape Administrator** and select **PC Setup > Validate / Add Odbc Source**.



- 2 Right-click in the middle work pane and click **Add Odbc entry**. **WhereScape Administrator** will open ODBC Administrator. Choose the **System DSN** tab and click **Add** to add a System Data Source.



- 3 Select a driver for your Data Source, choose **Teradata** and click **Finish**.

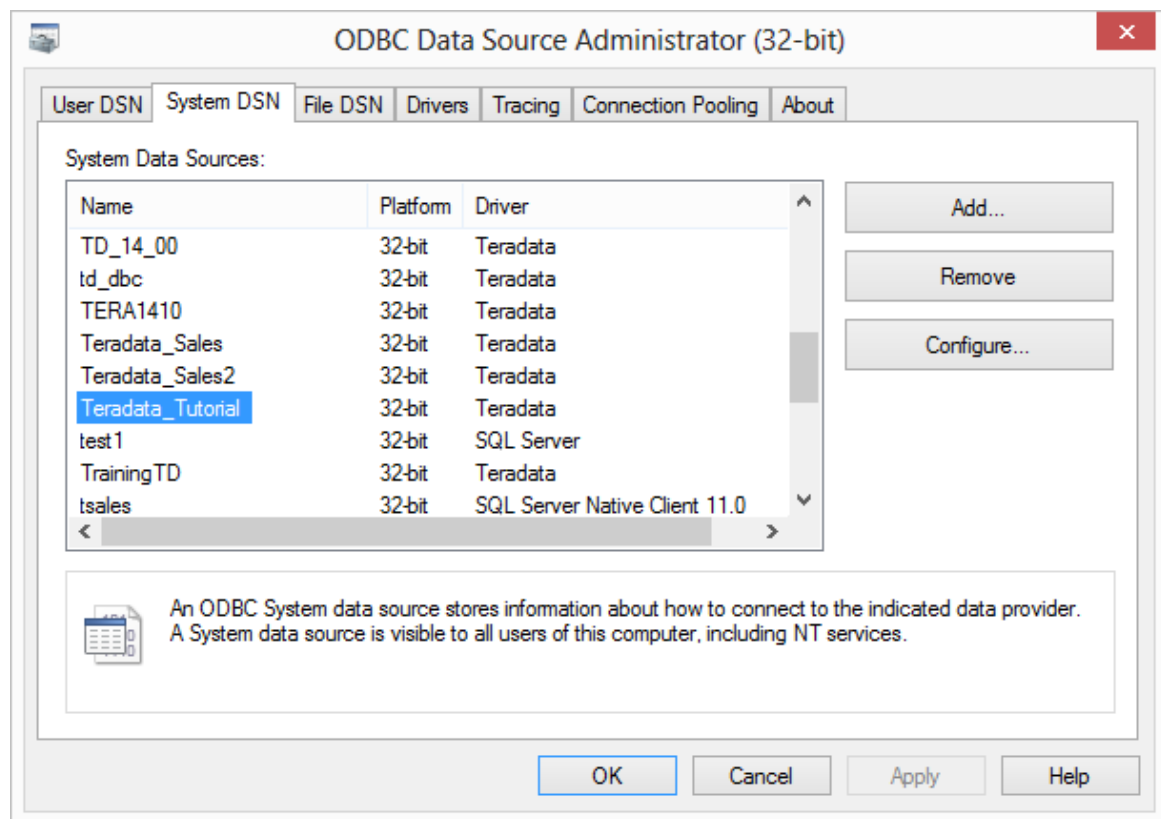


- 4 Enter a Data Source **Name** of **Teradata_Tutorial**. As the Teradata Server is environment specific, enter the name of your Teradata Server.
- 5 Enter either User name and Password or your TD Wallet String. **Click Ok.**

The screenshot shows the 'ODBC Driver Setup for Teradata Database' dialog box. It is divided into several sections:

- Data Source:** Contains a 'Name' field with the value 'Teradata_Tutorial' and an empty 'Description' field. To the right are 'OK', 'Cancel', and 'Help' buttons.
- Teradata Server Info:** Contains a 'Name or IP address' dropdown menu with the value 'TD_14_00'.
- Authentication:** Contains a 'Use Integrated Security' checkbox (unchecked). Below it is a 'Mechanism' dropdown menu. A 'Parameter' field is followed by a 'Change...' button. The 'Username' field contains 'tutorial'. There are two radio buttons: 'Password' (unchecked) and 'Teradata Wallet String' (checked). The 'Teradata Wallet String' field contains 'type your TD Wallet String|' and has a tooltip that reads '\$tdwallet(type your TD Wallet String)'. There is also a 'Change...' button next to this field.
- Optional:** Contains 'Default Database' and 'Account String' text boxes, followed by an 'Options >>' button.
- Session Character Set:** A dropdown menu with 'ASCII' selected.

- 6 Click OK to close.



- 7 In the left pane, click on **Validate / Add Odbc Source** again to refresh the middle pane and display the new ODBC Source Teradata_Tutorial.

1.3 CREATING A CONNECTION

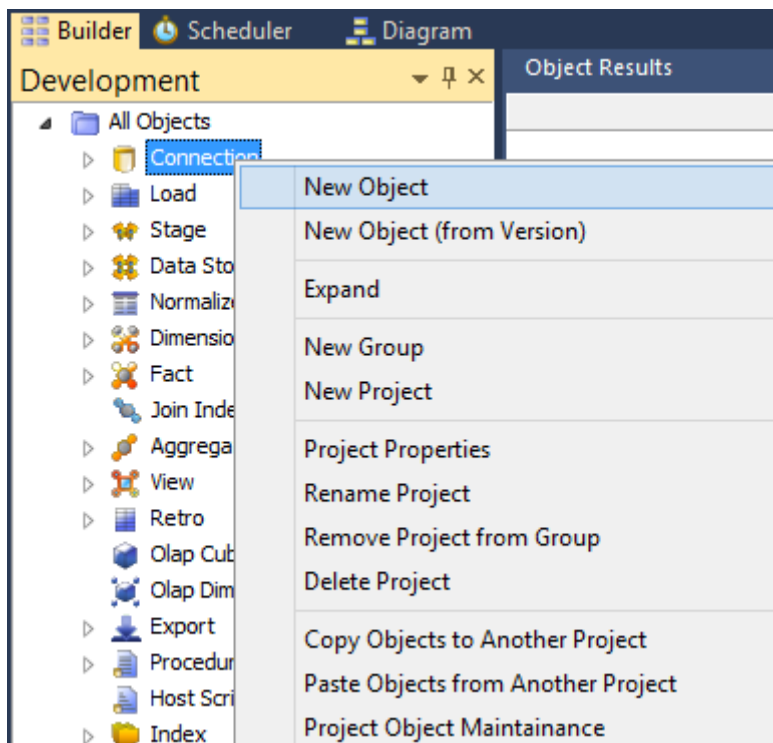
In order to populate the metadata repository, a connection needs to be created for the tutorial source system.

This section describes how to create this new connection.

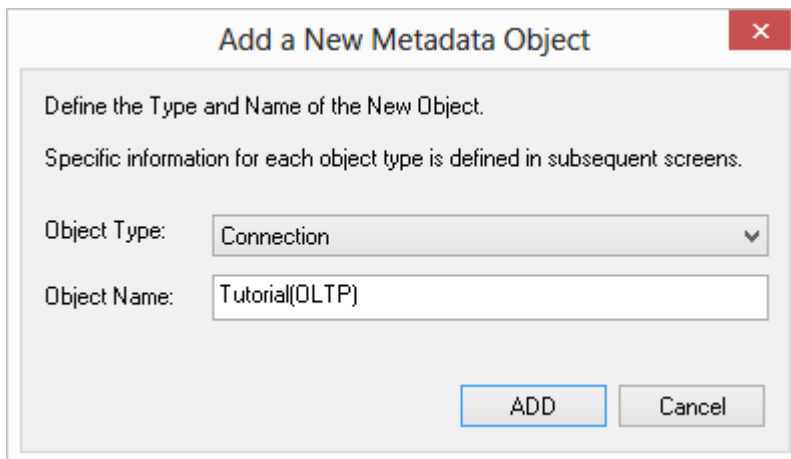


TIP: In order to use the drag and drop features there must always be a connection to the data warehouse itself.

- 1 In the left pane of the RED Builder, click on and highlight the **Connection object group**. (This selects the object group to be worked on.)
 - From the right mouse menu select **New Object**.



- 2 A dialog box displays with the Object Type defaulted to Connection.
 - Name your connection **Tutorial(OLTP)** and click **ADD**.



Add a New Metadata Object [X]

Define the Type and Name of the New Object.

Specific information for each object type is defined in subsequent screens.

Object Type:

Object Name:

- 3 The Connection Definition dialog displays next. Complete the details as below and click **OK**:
 - The **ODBC source** is Teradata_Tutorial.
 - The **Database ID (SID)** can be left blank.
 - The **Database Link ID** can be left blank.
 - The **Extract User ID** and **password** are the user name and password for a user who has select access to the tutorial database (or user) where the source tables reside.
 - If the **ODBC User Default** is set to Teradata Wallet, enter the User ID who has select access to the tutorial database and the relevant Teradata Wallet String.
 - The **Administrator User ID** and **password** can be left blank for the tutorial.
 - The **New Table Default Load Type** enables you to set the default load type at connection level for ODBC and Database Connections. Set to 'Database link load'.
 - Leave **Data Type Mapping** set as 'Default'.

Connection Tutorial(OLTP)

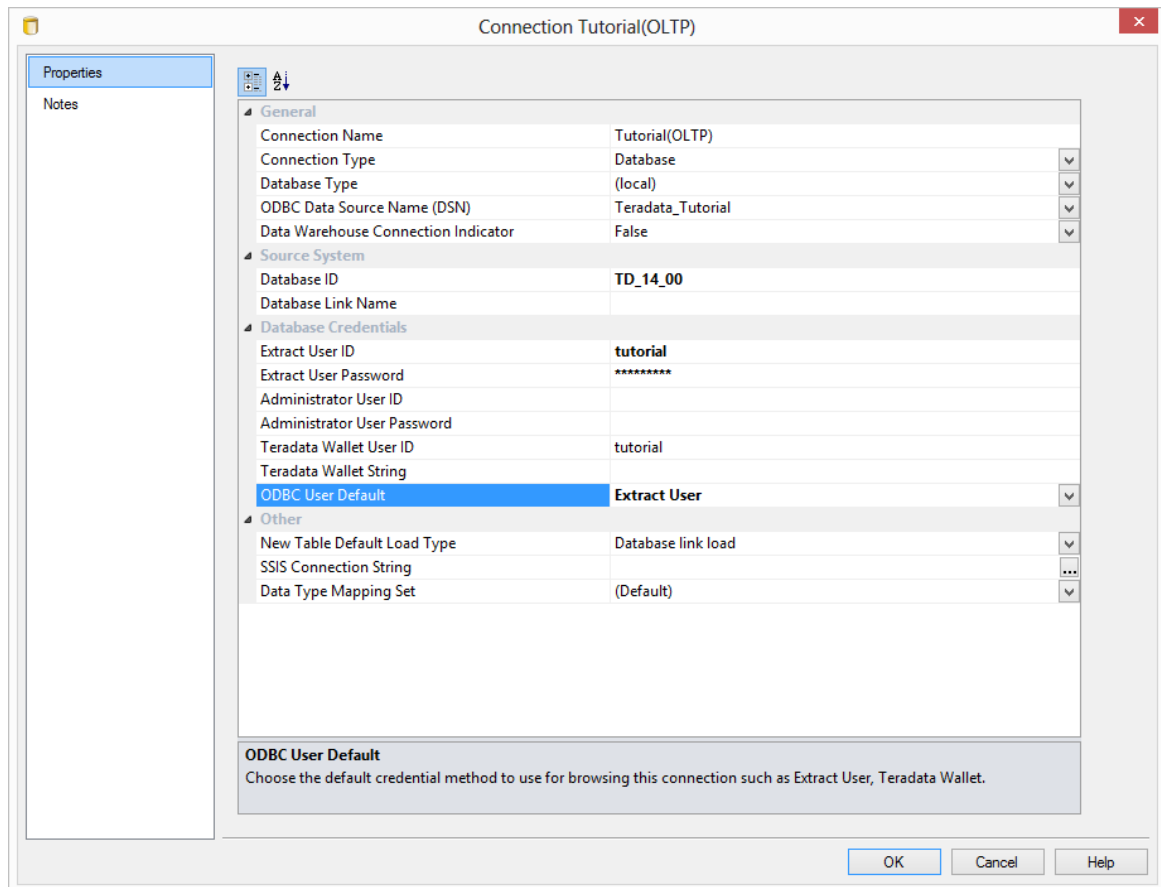
Properties

Notes

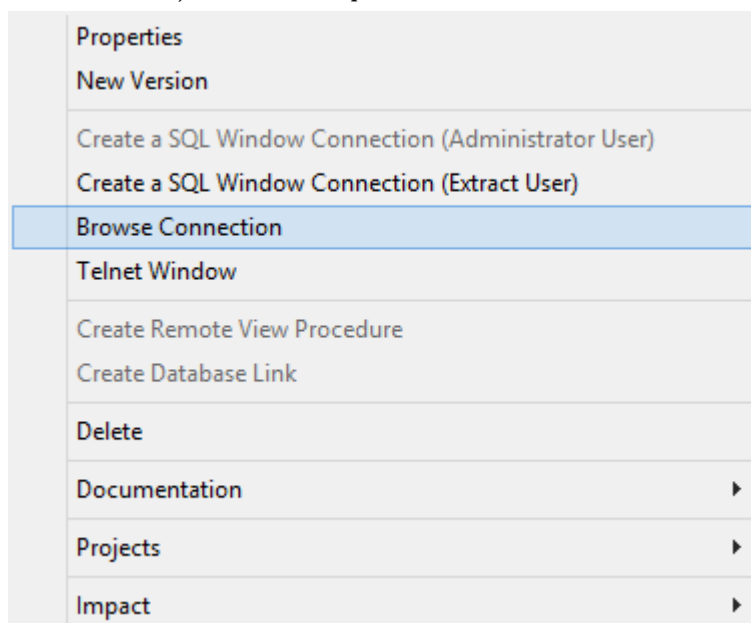
General	
Connection Name	Tutorial(OLTP)
Connection Type	Database
Database Type	(local)
ODBC Data Source Name (DSN)	Teradata_Tutorial
Data Warehouse Connection Indicator	False
Source System	
Database ID	TD_14_00
Database Link Name	
Database Credentials	
Extract User ID	
Extract User Password	
Administrator User ID	
Administrator User Password	
Teradata Wallet User ID	tutorial
Teradata Wallet String	type your TD Wallet String
ODBC User Default	Teradata Wallet
Other	
New Table Default Load Type	Database link load
SSIS Connection String	...
Data Type Mapping Set	(Default)

Teradata Wallet String
Enter a Teradata Wallet reference string as password for the user. Do not include the Teradata Wallet function Stdwallet() – specify only the wallet string. [Optional].

OK Cancel Help



- 4 To confirm that you have connected to the system correctly, right-click on the new connection object in the left pane and select **Browse Connection**.



- 5 Ensure the **Schema** is set to the database (or user) where tutorial tables have been created and click **OK**.

Note: If the Extract User ID is selected, the Schema will have the user ID and the password encrypted. If the TD Wallet has been selected instead, the Browse Connection Window will show the TD Wallet String by default on the password window and the TD Wallet String check-box will be ticked.

The screenshot shows the 'List Source Tables Connection' dialog box. The 'Connection' dropdown is set to 'Tutorial(OLTP)'. The 'User ID' field contains 'tutorial' and the 'Password' field is masked with '*****'. There is an unchecked checkbox labeled 'Password is TD\Wallet string.'. The 'Filter' section contains a 'Schema' field with 'tutorial', a 'Name' dropdown with '(None)', and an 'Object Types' section with 'Tables' and 'Views' checked, and 'System Objects' unchecked. The 'Group' and 'Project' dropdowns are both set to '(All)'. The 'Data Type Mapping Set' dropdown is set to '(Default)'. At the bottom, there are three buttons: 'Refresh Current', 'OK', and 'Cancel'.

List Source Tables Connection ✕

Connection: Tutorial(DLTP) ▼

User ID: tutorial

Password: (TD Wallet String will be shown here by default)

Password is TDWallet string.

Filter

Schema: tutorial

Name: (None) ▼

Object Types

Tables Views System Objects

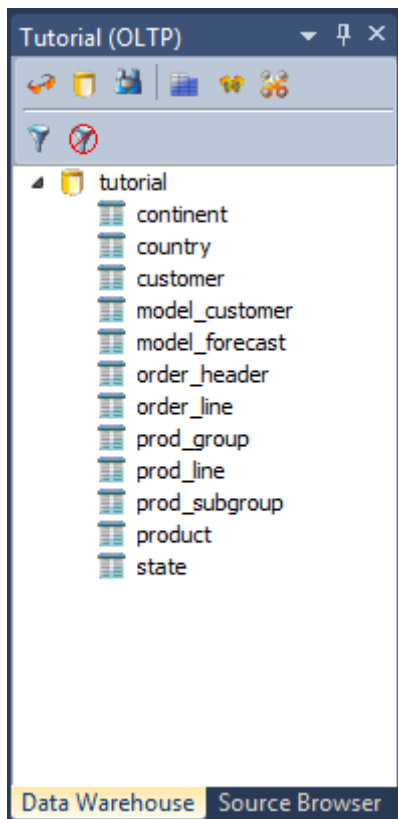
Group: (All) ▼

Project: (All) ▼

Data Type Mapping Set: (Default) ▼

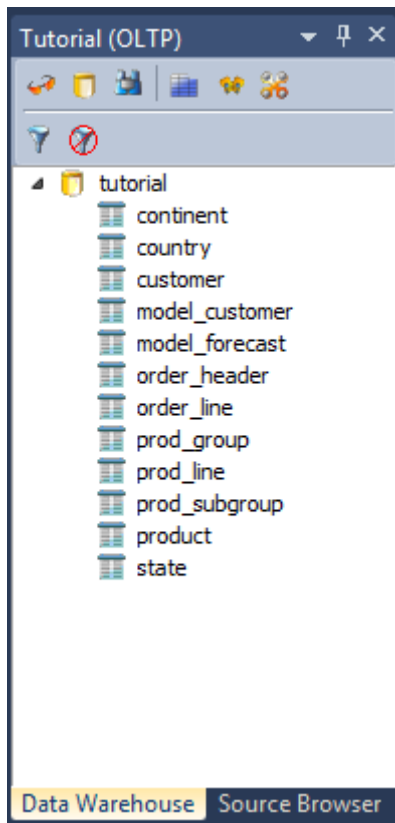
Refresh Current OK Cancel

- 6 A third pane on the right displays the tables contained in the tutorial source system:



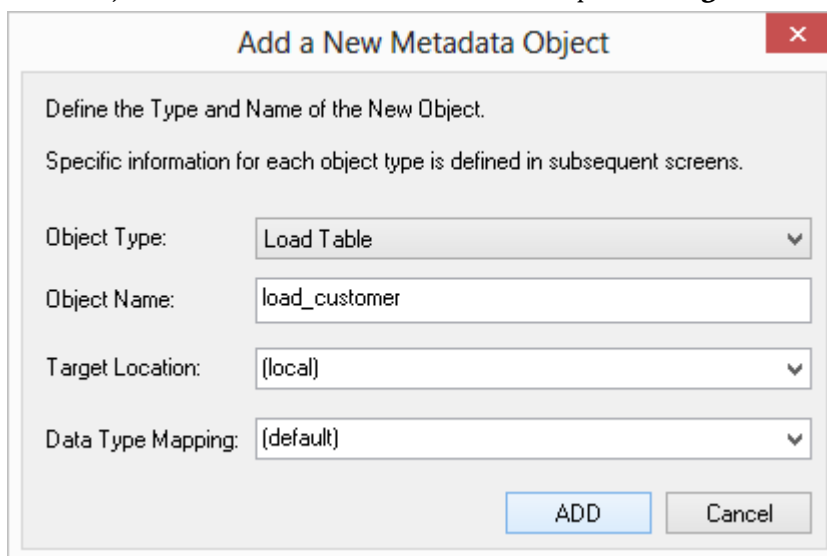
You have created a database connection for the source system **Tutorial (OLTP)** and are now ready to proceed to the next step - **Loading Source Tables** (see "**1.4 Loading Source Tables**" on page 20)

- 2 Ensure the Object Tree is expanded in the right pane.



- 3 Click on **customer** and drag this table into the middle pane, placing it anywhere in the pane. A dialog box displays with the name of the object defaulted to **load_customer**.

 - If you want to place this table on a different schema, use the Target Location drop-down list to select the right schema for your table. To see more details about placing tables in different schemas, see further instructions on the next topic *Loading Source Tables using schemas*.



- 4 Click **Add** and the following table definition will display:

Load load_customer

Properties

Storage

Override Create DDL

Source

Notes

Load Table Name: load_customer

Unique Short Name: (maximum 22 characters) load_customer

Description:

Connection: Tutorial(DLTP)

Load Type: Database link load

Database Link:

Script Name: (None)

Pre-Load Action: Truncate

Pre-Load SQL:

Post Load Procedure: (None)

Timestamps

Metadata Structure Changed:	Database Created:	Database Altered:
2014-10-30 00:45:53.820000	2014-10-30 00:46:23.970000	2014-10-30 00:46:23.970000

OK Cancel Help

Note: For the purposes of this tutorial, all the necessary details have been created automatically, so click **OK**. See the Loading Data chapter for explanations of the load table fields and options.

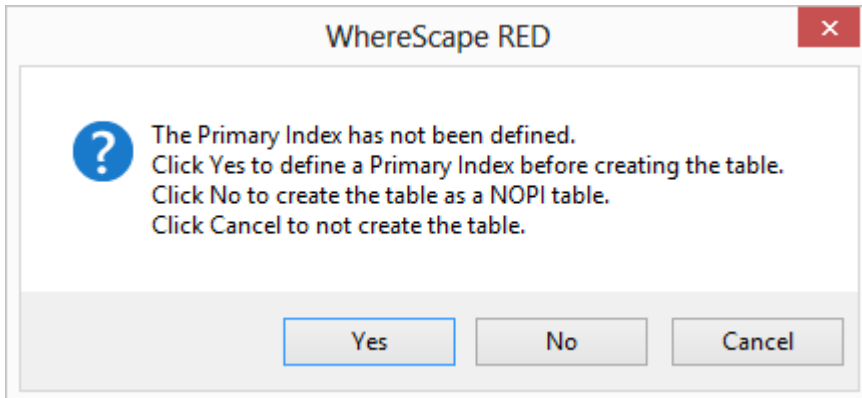
- 5 A dialog box displays showing that the load table **load_customer** has been defined and asks if you want to create and load the table. Click **Create and Load**.

Create Database Table

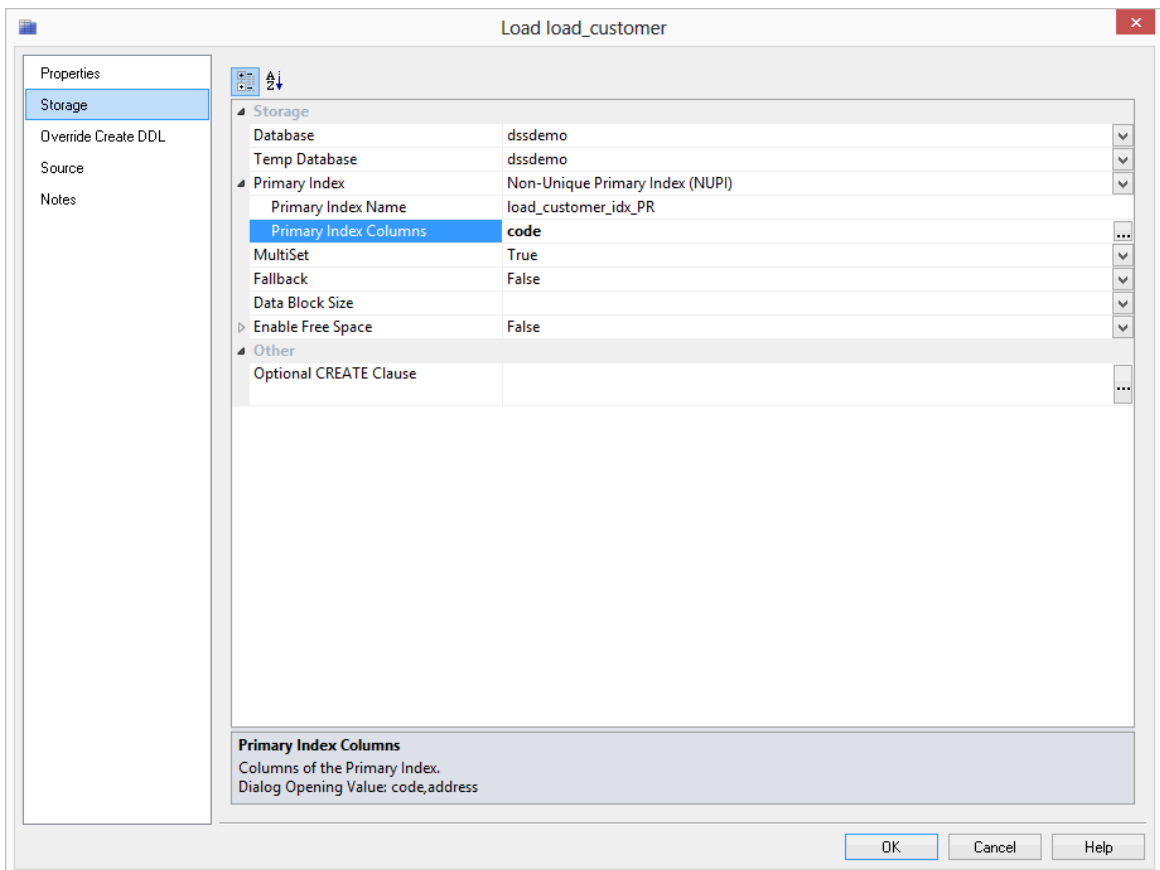
Load load_customer has been defined

Create Create and Load Close

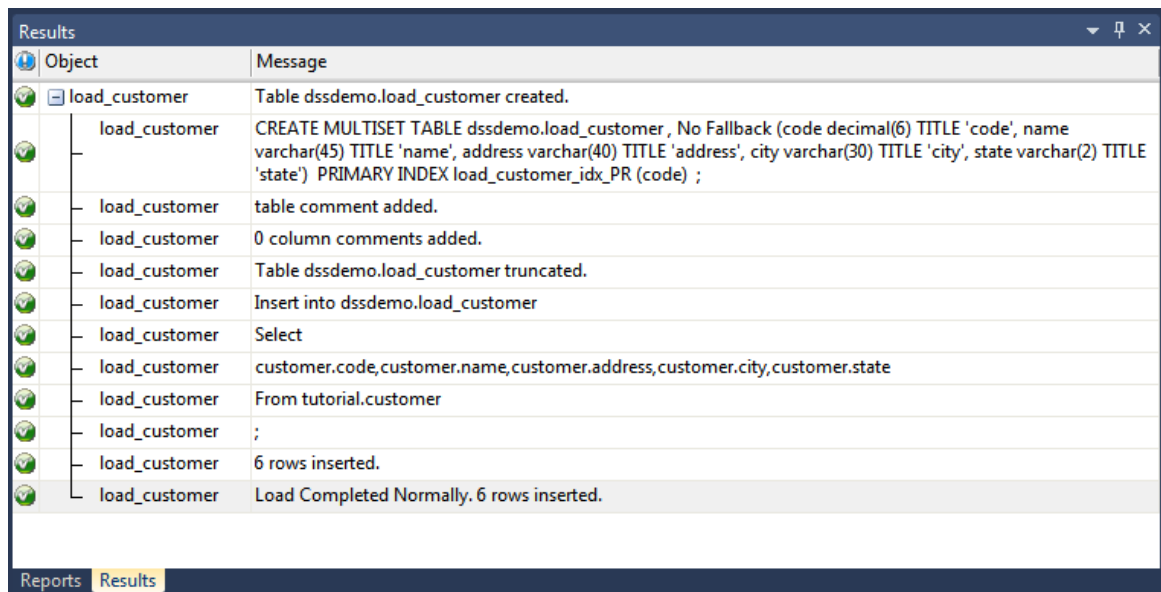
- 6 Click **Yes** to define a Primary Index.



- 7 Type **code** as the on the Primary Index Columns field. This will create the physical tables in the data warehouse and load the data. Click **OK** on the Storage screen.



- 8 The physical tables will now be created in the data warehouse and the data will be loaded. The results will be posted in the results pane.



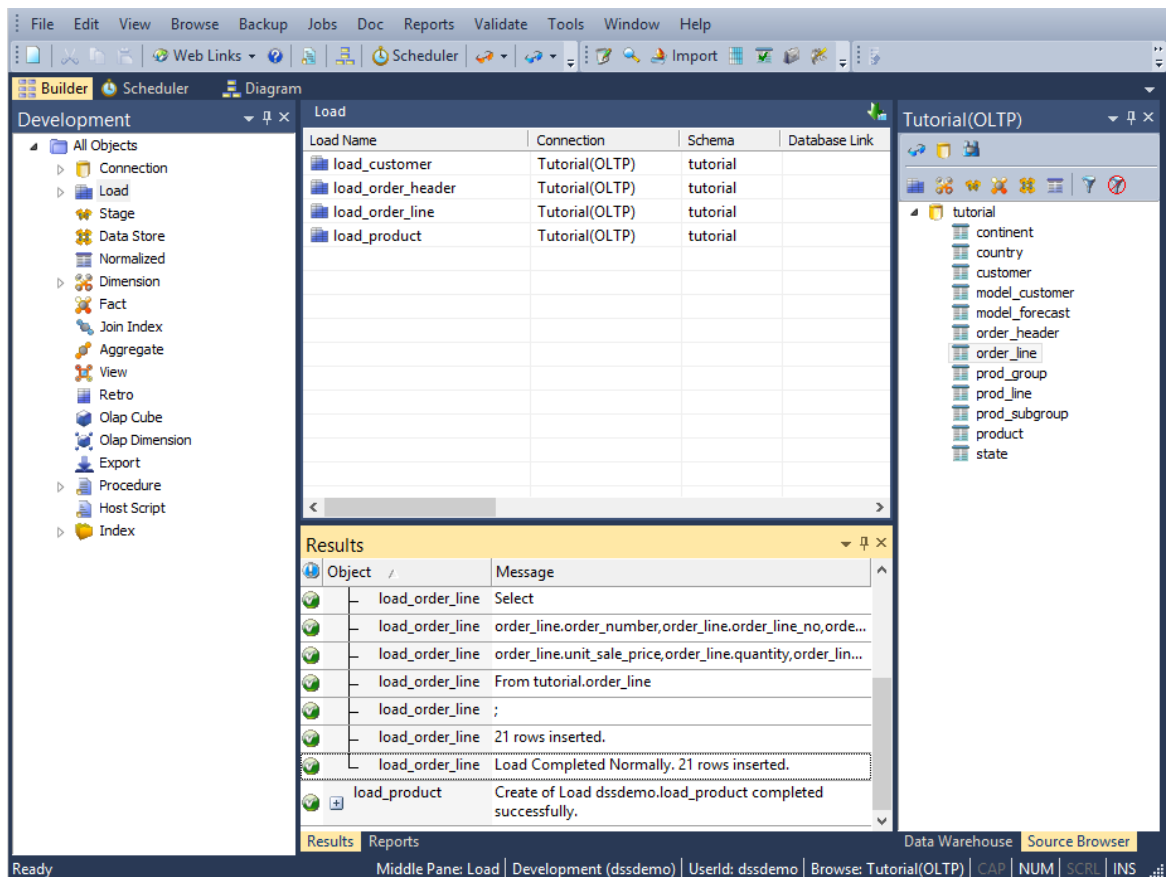
Object	Message
load_customer	Table dssdemo.load_customer created.
load_customer	CREATE MULTiset TABLE dssdemo.load_customer , No Fallback (code decimal(6) TITLE 'code', name varchar(45) TITLE 'name', address varchar(40) TITLE 'address', city varchar(30) TITLE 'city', state varchar(2) TITLE 'state') PRIMARY INDEX load_customer_idx_PR (code) ;
load_customer	table comment added.
load_customer	0 column comments added.
load_customer	Table dssdemo.load_customer truncated.
load_customer	Insert into dssdemo.load_customer
load_customer	Select
load_customer	customer.code,customer.name,customer.address,customer.city,customer.state
load_customer	From tutorial.customer
load_customer	;
load_customer	6 rows inserted.
load_customer	Load Completed Normally. 6 rows inserted.

- 9 Notice that the Load Table object group in the left pane now has a dependent/child.
- 10 Repeat this process (steps 1 - 7) for the source tables **product**, **order_header**, and **order_line**. When selecting the Primary Index for each table, choose:
- **code** for product table
 - **order_number** for order_header table
 - **order_number** for order_line table



TIP: Remember to double click on the left pane Load Table object group between loading each of the source tables to ensure that you are reassigning the target, rather than adding to the columns in the middle pane.

11 Your screen should look something like this:



You are now ready to proceed to the next step - **Building Dimensions** (see "1.5 Building Dimensions" on page 31)

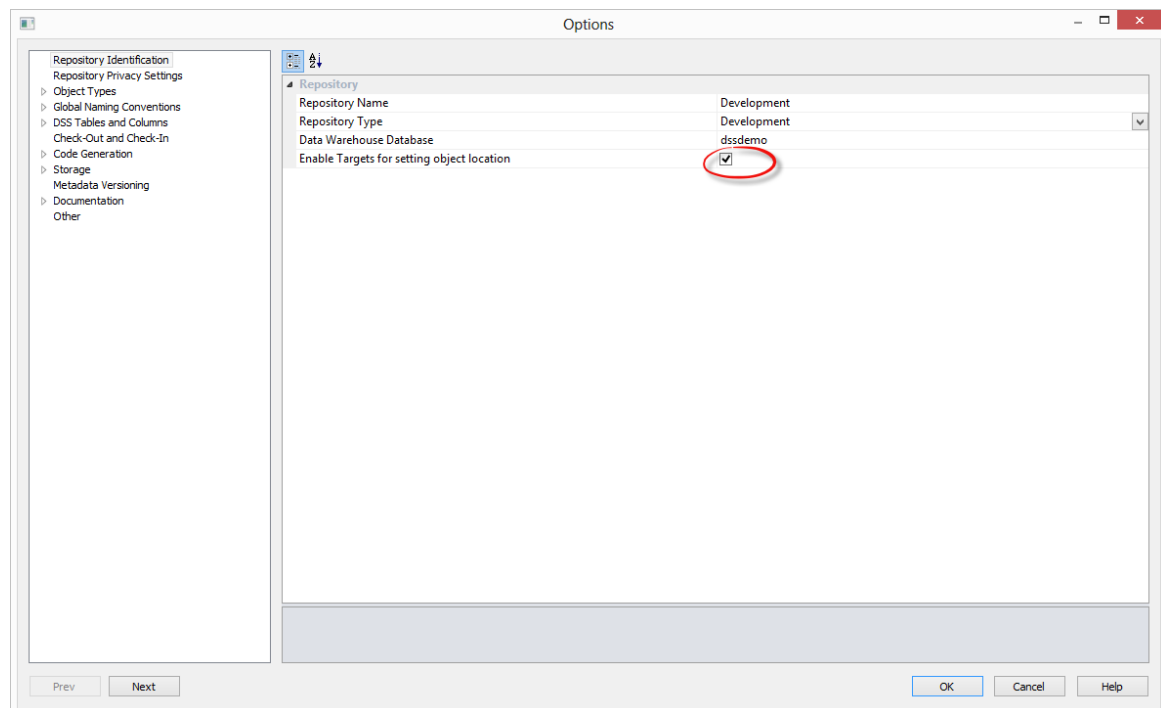
1.4.1 LOADING SOURCE TABLES USING SCHEMAS

Tables can be placed in different target locations on a connection level which enables simpler handling of deployments and database storage changes. To place tables in different target locations users should follow the below steps:

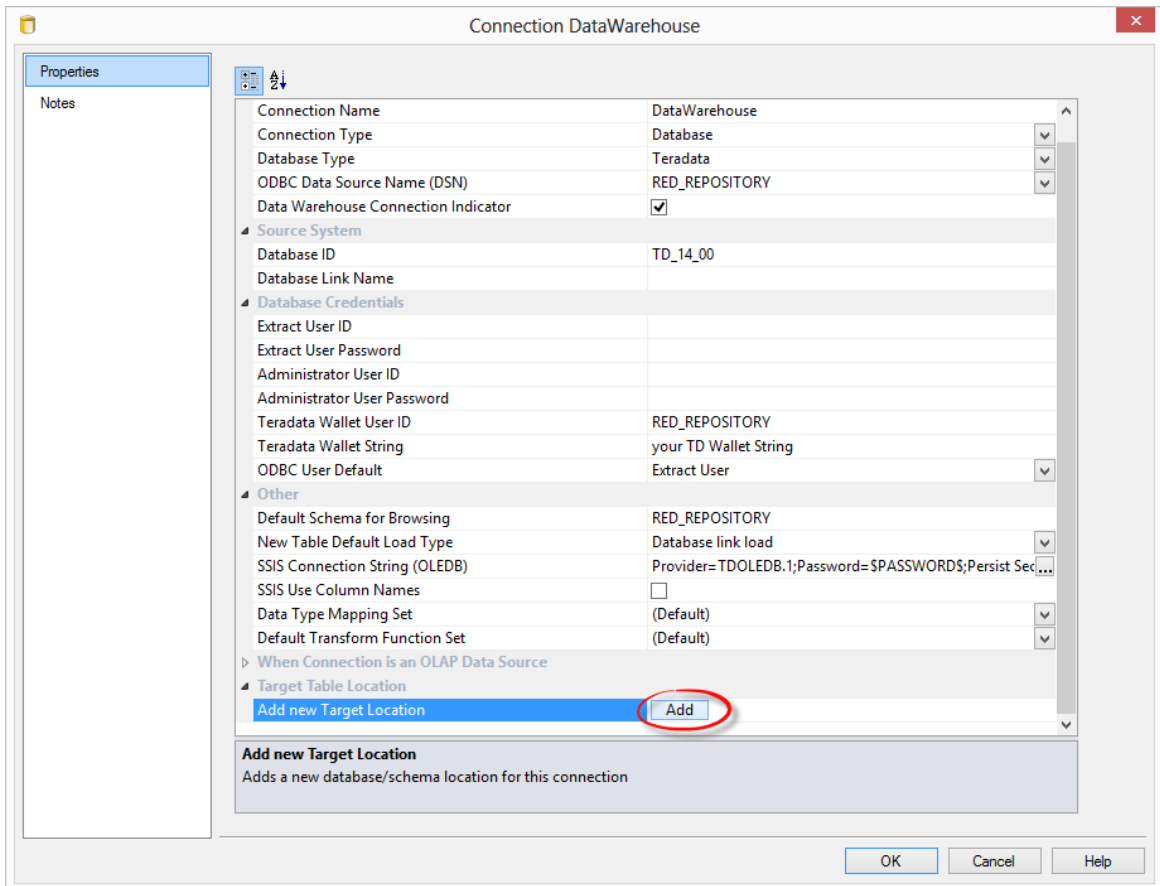
Overview

- Ensure the relevant target databases exist in Teradata. Create any databases that do not exist.
- Set the **Enable Targets for setting object location** option in the <PRODUCT> Repository Identification options.
- Add one or more target locations to the Data Warehouse connection in <PRODUCT> for each database you intend to use.
- Configure the Data Warehouse connection in <PRODUCT> to browse all required databases by default.
- Set the default target locations for **new tables** in the <PRODUCT> **Table Location** options.
- When defining a new table in <PRODUCT>, check and ensure the correct target is set on the **Storage** tab.

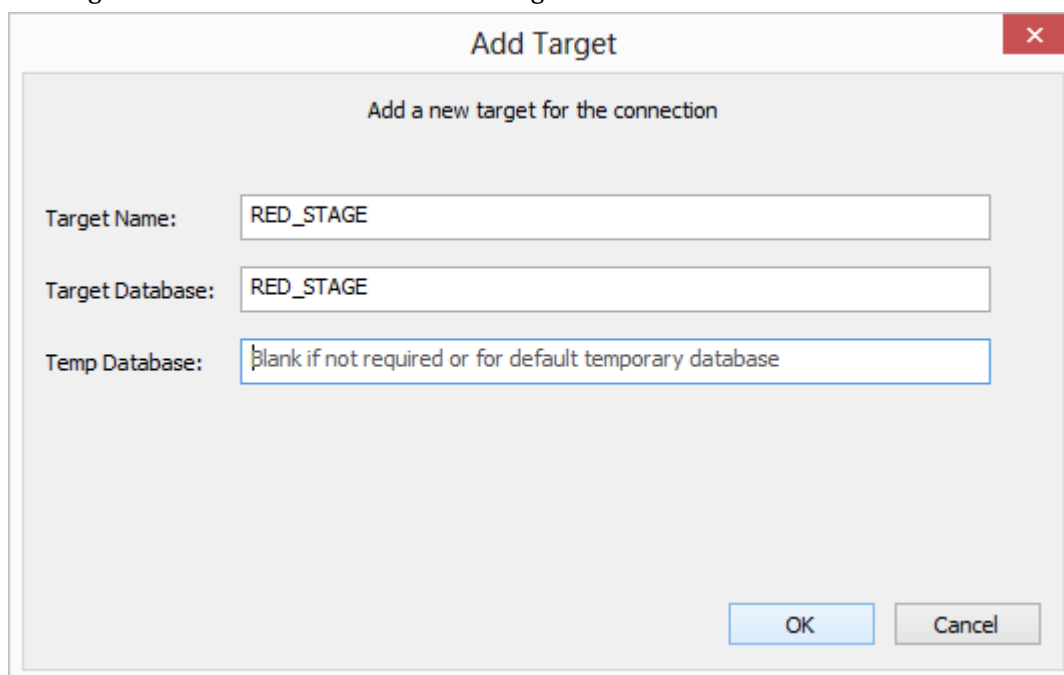
- 1 After logging in to <COMPANY> <PRODUCT>, make sure the **Enable Targets for setting object location** option is set in the **Tools->Options->Repository Identification** settings.



- 2 Add one or more target locations to the Data Warehouse connection in <PRODUCT> for each target database you want to use:
 - Click the **Add** button to add the required target locations for this connection.

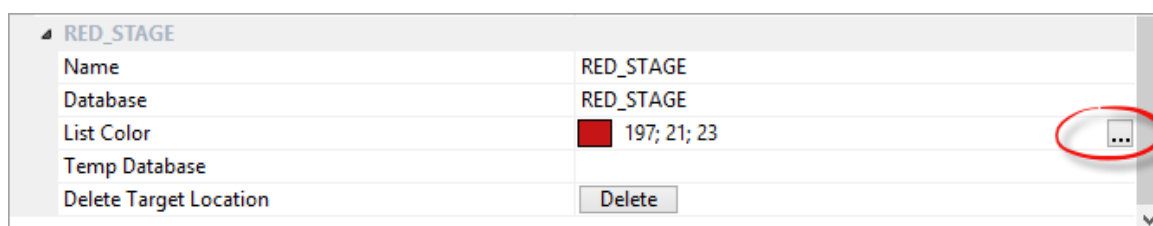


- 3 Give the new **Target Database** a name and then enter the target's database. It is best to set the target name to the same name as the target database.



The image shows a dialog box titled "Add Target" with a close button (X) in the top right corner. Below the title is the instruction "Add a new target for the connection". There are three input fields: "Target Name:" with the value "RED_STAGE", "Target Database:" with the value "RED_STAGE", and "Temp Database:" with the placeholder text "blank if not required or for default temporary database". At the bottom right, there are two buttons: "OK" and "Cancel".

- 4 Expand the target locations to change target database location colors or to delete target databases.

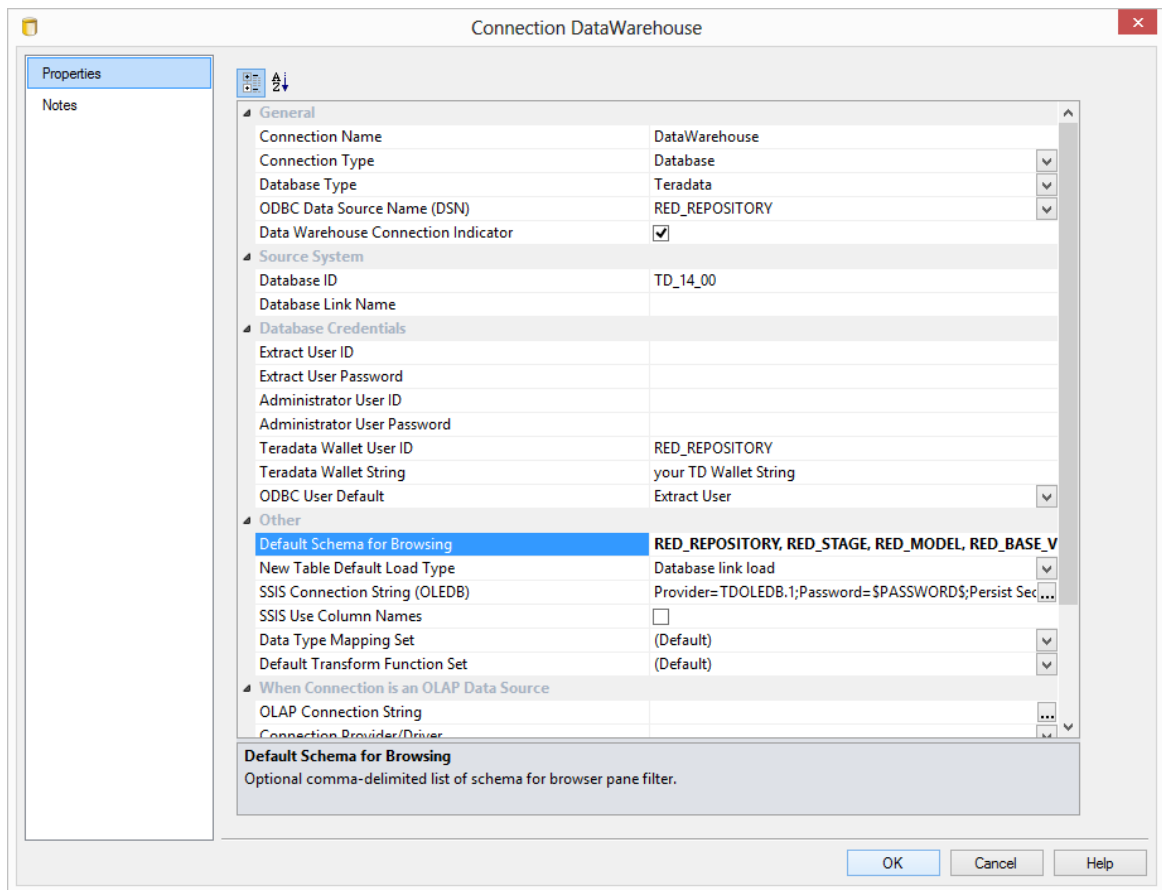


The image shows a table with the following data:

RED_STAGE	
Name	RED_STAGE
Database	RED_STAGE
List Color	■ 197; 21; 23
Temp Database	
Delete Target Location	Delete

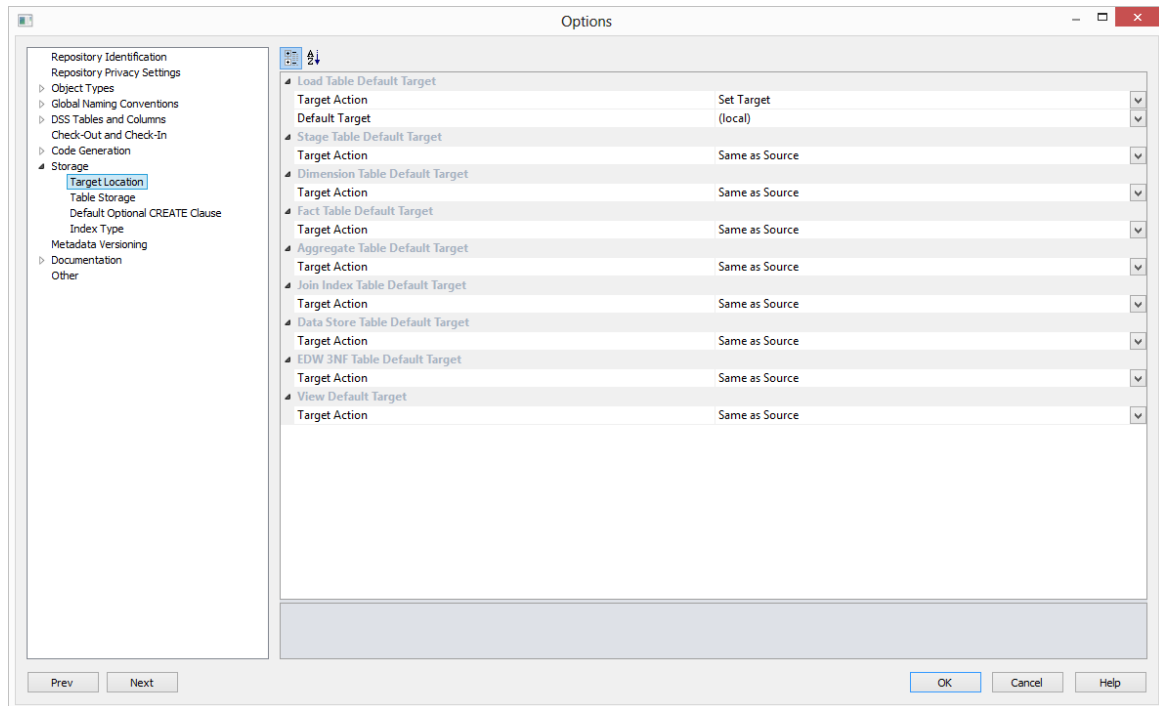
A red circle highlights a menu icon (three dots) on the right side of the table, next to the "List Color" row.

- 5 Still in the **DataWarehouse** connection, add the new databases to the **Default Schema for Browsing** field separated by commas.



NOTE: While browsing this connection, <PRODUCT> will then display a list with all the target databases and their associated objects on the right-hand browser pane.

- 6 You are also able to set the default target location for **New Tables** in **Tools/Options**. This default target location is only applied when a new table is created, not for existing tables.
- Select between the **Set Target** option to set a default target location for new tables (or to use the table's storage) or
 - **Same as Source** to place new tables in the same database as their source.



- 7 Follow the usual steps for creating objects by using the drag and drop functionality. As you are defining a new table in <PRODUCT>, ensure the correct target location options are set in the **Target Location** options before creating the table in the database. When using drag and drop, it is also possible to change the target database location of a particular object as you create it. After a table has been created, it is also possible to change its target location on the **Storage** tab of the table's Properties screen.

NOTE: When upgrading from a <PRODUCT> version previous to 6.8.2.0 and moving existing objects to a target location, all procedures that reference those objects will need to be rebuilt. Any **FROM** clauses will also need to be manually regenerated in order for the table references to be updated to the new [TABLEOWNER] form.

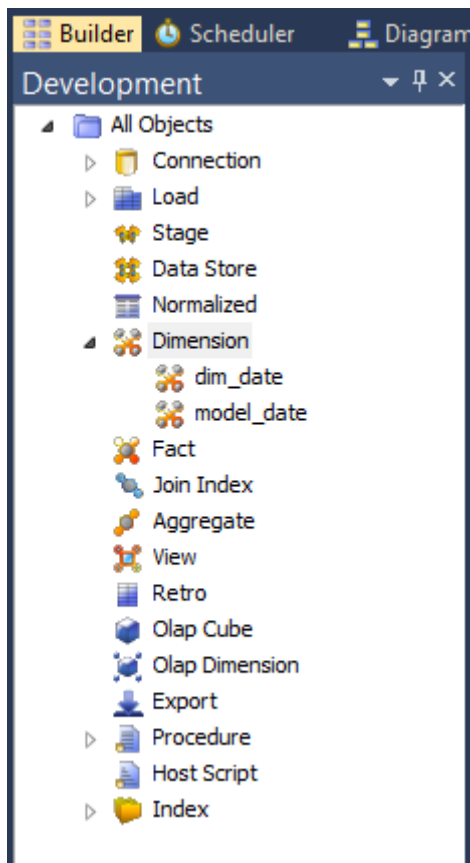
1.5 BUILDING DIMENSIONS

The necessary source tables have been loaded into the data warehouse. Now the dimensions of the data warehouse can be built.

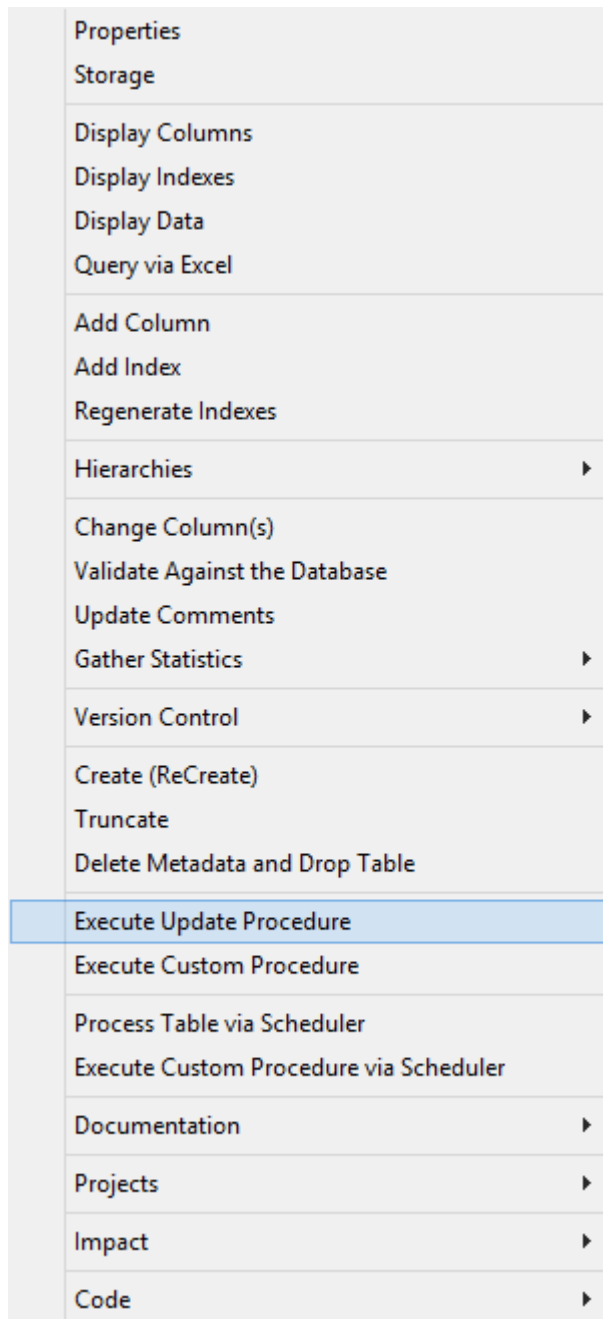
When building dimensions you will be prompted as to how you would like the dimension managed. WhereScape RED generates code for normal, slowly changing and date ranged dimensions.

You will also be prompted for the business (or natural) key of the dimension. This is needed so WhereScape RED knows when to add new dimensional records.

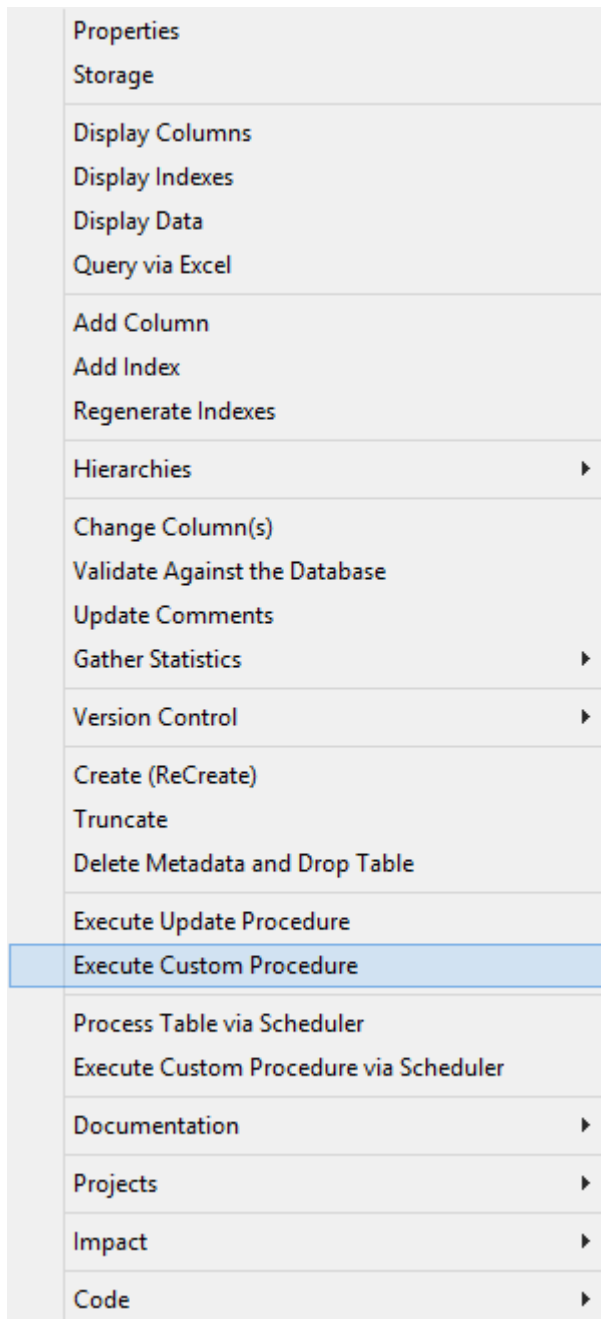
- 1 Double-click on the **Dimension Table object group** to expand the dimension folder in the left pane.



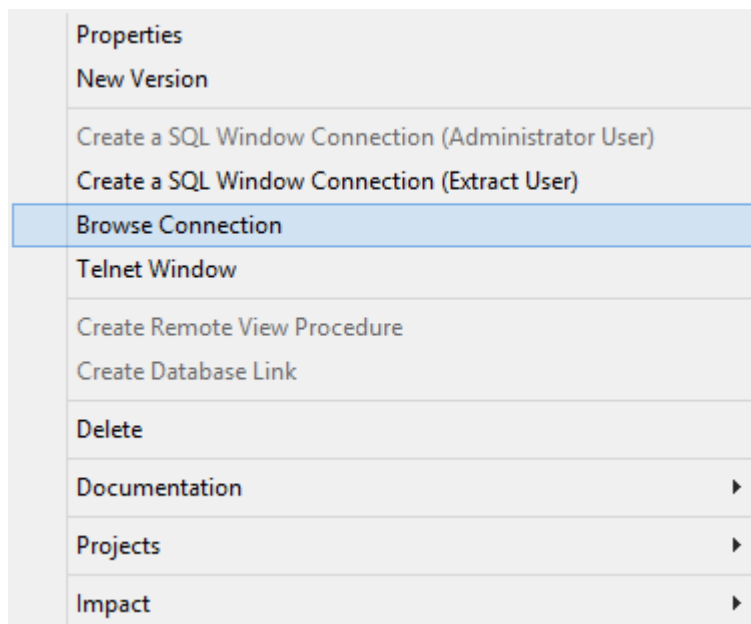
- 2 Right-click on **dim_date** and select **Execute Update Procedure**.



- 3 Right-click on **dim_date** and select **Execute Custom Procedure**.



- 4 Right-click on the **DataWarehouse** connection in the left pane and select **Browse Connection**.



- 5 The Source Tables Connection will display either your encrypted password or your TD Wallet string depending on the connection properties database credentials' option chosen. Click **OK**.

List Source Tables Connection

Connection: DataWarehouse

User ID: dssdemo

Password: TD Wallet String

Password is TD Wallet string.

Filter

Schema: dssdemo

Name: (None)

Object Types

Tables Views System Objects

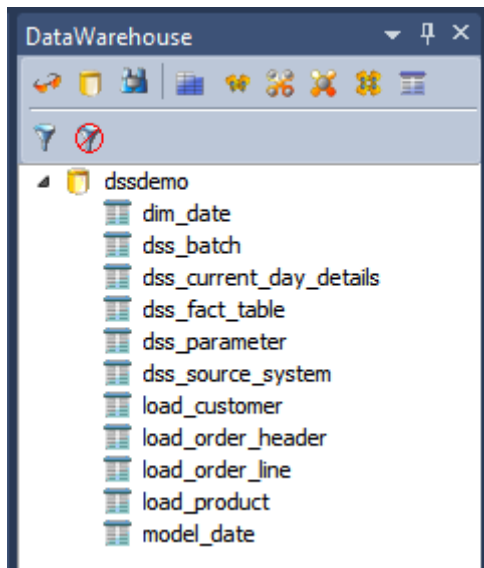
Group: (All)

Project: (All)

Data Type Mapping Set: (Default)

Refresh Current OK Cancel

- 6 This shows the data warehouse tables and views in the right pane.



- 9 A Dimension Type dialog is displayed.
Click **Normal**.

Dimension Type

Four methods are provided for managing dimensions. Please select the desired method.

1. Normal. The dimension is updated based on a business key, with new records being added if required. All columns except the business key can change. Normal
2. Slowly changing. Changes in the values of selected columns result in new dimensional records being created. In all other respects the same as Type 1. Slowly Changing
3. Previous data retained. The previous values of selected columns are stored in additional columns. In all other respects the same as Type 1. Previous values
4. Date Ranged. The source system provides a date ranged business key. Similar to Type 2 except that we deal with the record as a whole and the dates are provided. Date Ranged

- 10 A table definition displays with all the necessary defaults completed.

- Make one change - Select **(Build Procedure...)** from the Update Procedure drop-down list box - this will generate procedures to get surrogate (artificial) keys based on the business key and to update the dimension.
- Click **OK**.

Dimension dim_customer

Properties

Storage

Override Create DDL

Language Mapping

Purpose

Concept

Grain

Examples

Usage

Notes

Table Name: Table Type:

Unique Short Name: (maximum 22 characters)

Business Display Name (EUL):

Description:

Update Procedure:

Custom Procedure:

Get Key Function: Mnemonic (EUL):

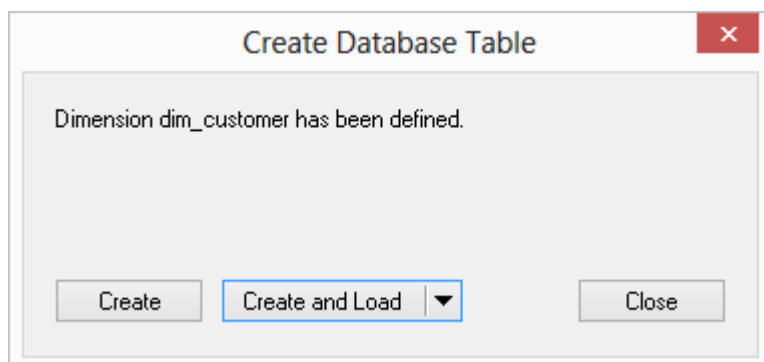
Timestamps

Metadata Structure Changed:

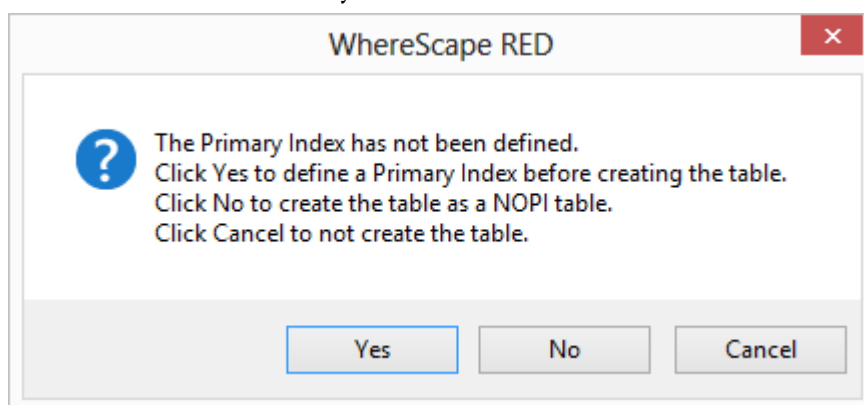
Database Created:

Database Altered:

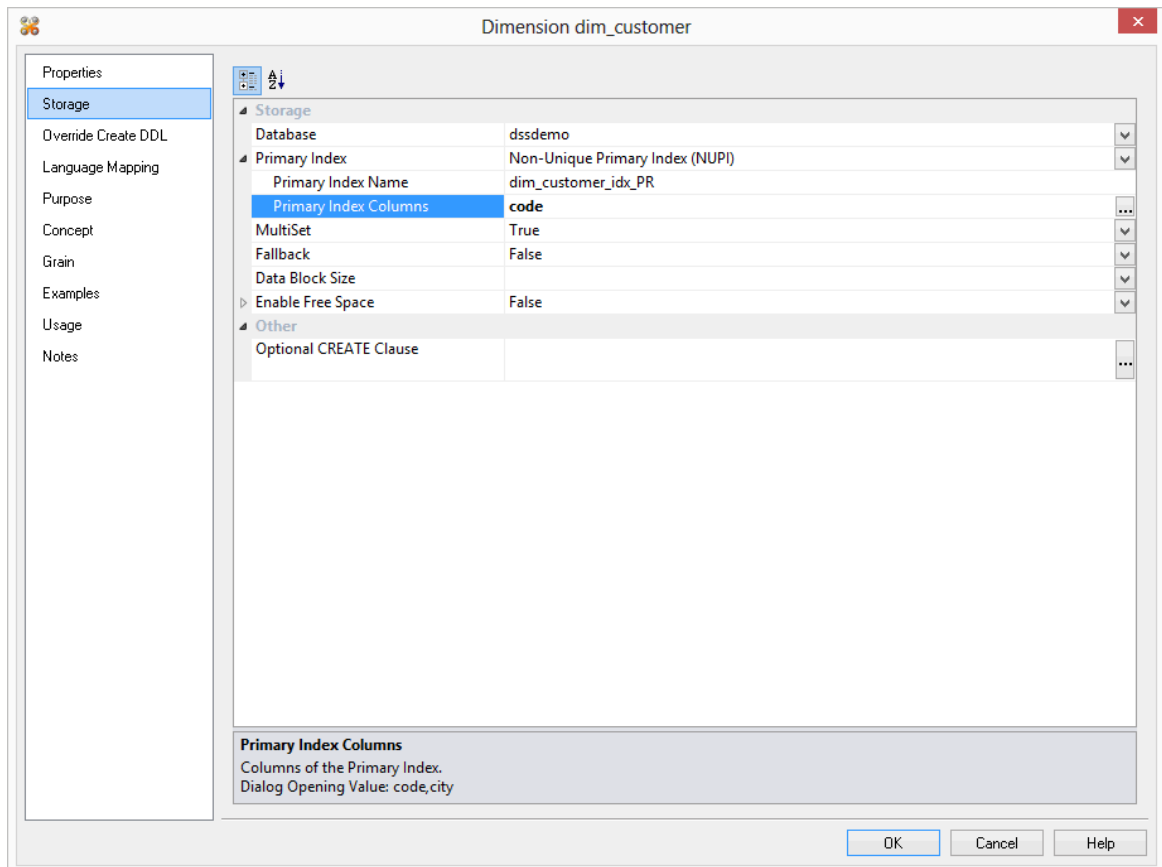
- 11 A dialog box displays confirming that the dimension table **dim_customer** has been defined and asking if you want to create and load the table.
Click **Create and Load**.



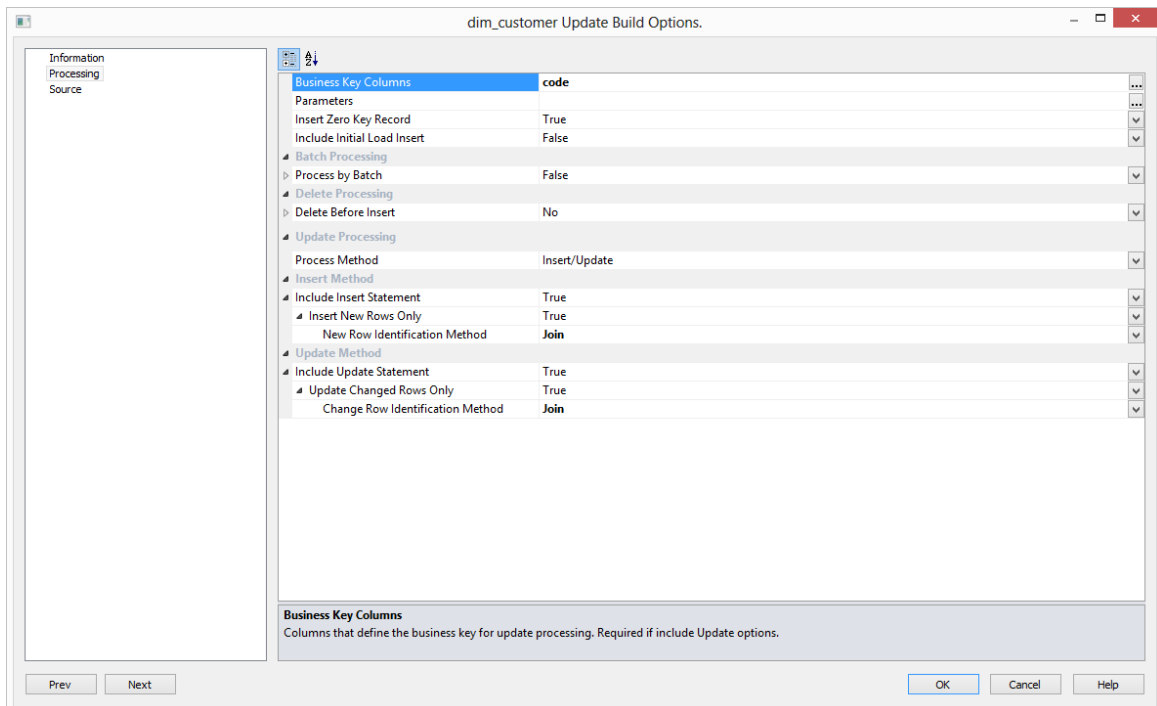
- 12 Click **Yes** to define a Primary Index.



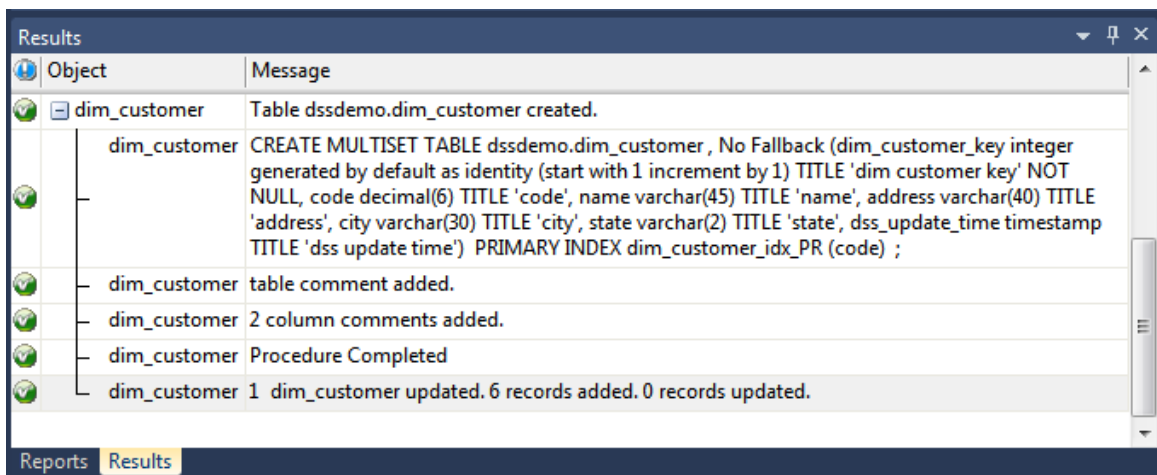
13 Type **code** as the Primary Index Column and click **OK** on the Storage screen.



- 14 Define the Business Key as below when the next dialog appears. The business (natural) key is the unique identifier for the dimensional record. Type **Code** and click **OK**.



- 15 Notice that the results will be posted in the Results pane.



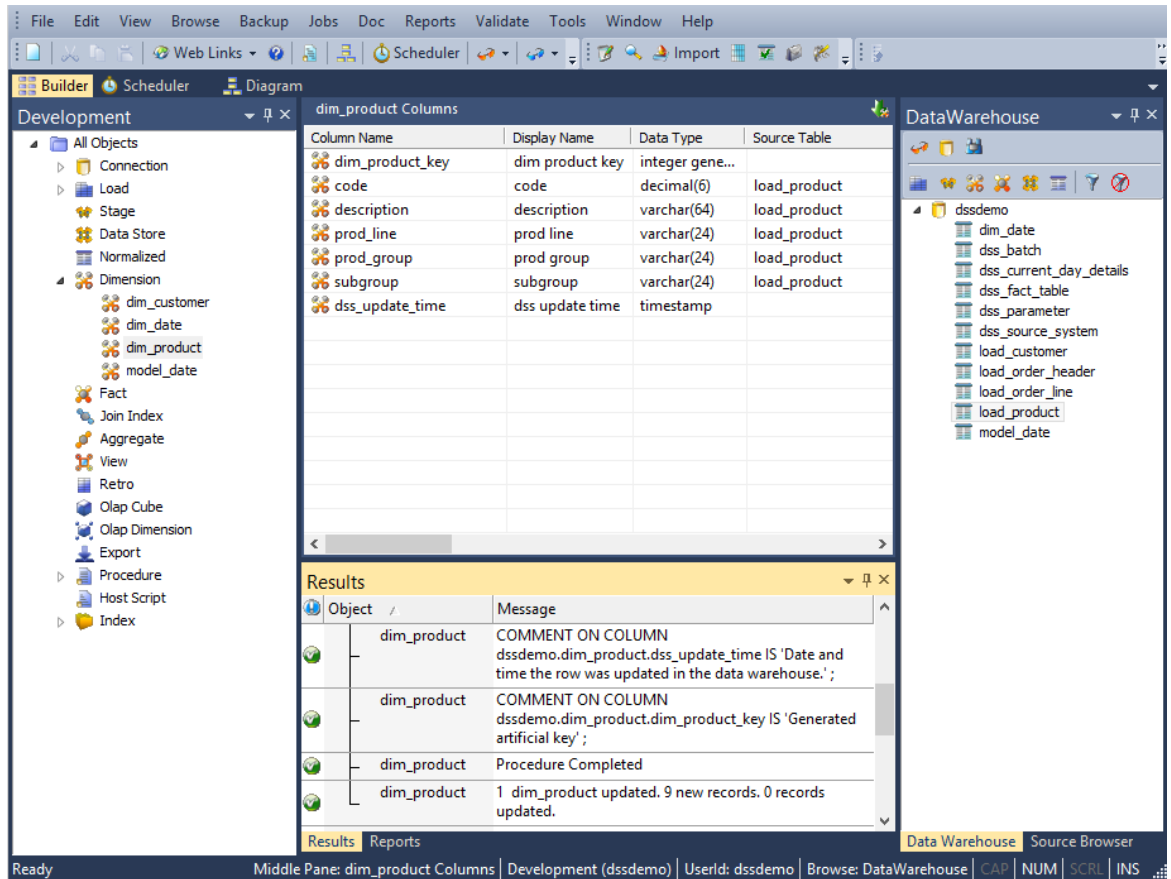
- 16 Repeat this same process (steps 7 through 16) for the load table **load_product**. The Primary Index and Business key will be **code**.



TIP: Remember to double-click on the left pane Dimension Table object group between loading each of the above dimension tables.

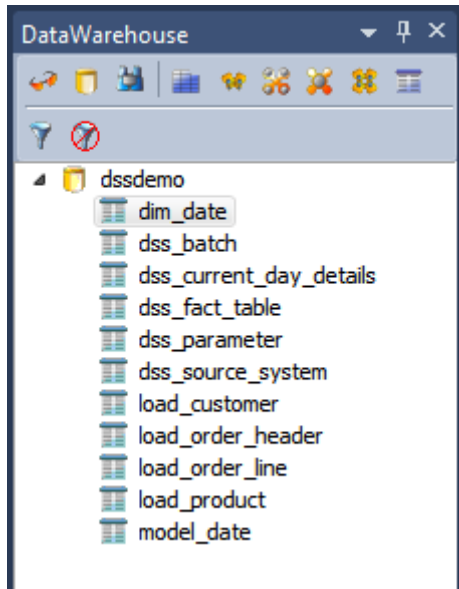
17 Double click **Dimension** in the left pane.

18 The WhereScape RED screen should look like this:



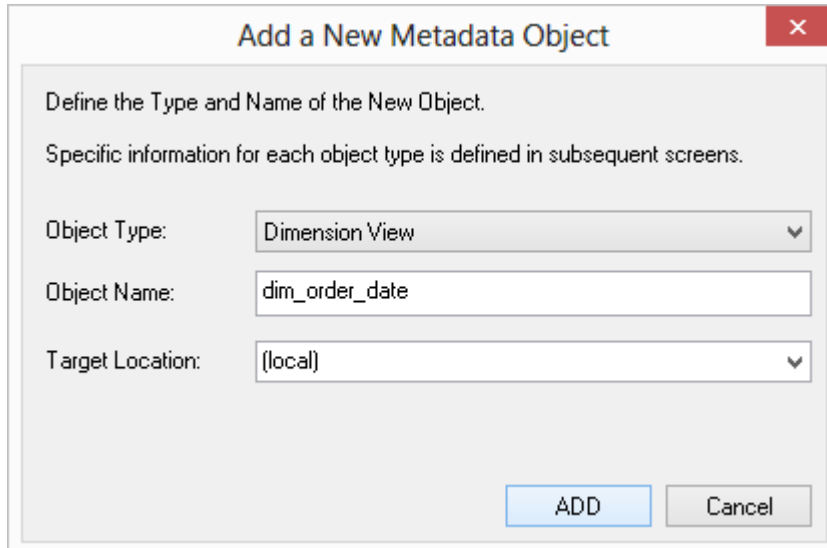
You are now ready to proceed to the next step - **Creating Dimension Views** (see "1.6 Creating Dimension Views" on page 42)

- 2 Click and drag **dim_date** from the right pane into the middle pane.



- 3 The dialog box that displays defaults the object type to a dimension view, and names the view **view_date**.

We want to create two dimension views from the same source, **dim_date**, so we need to change this dimension view name to one that is more meaningful, specifically **dim_order_date**.



- 4 Make this change and click **ADD**.

- 5 Make the following changes to the Column Definition dialog and click **OK**.

View Column Definition

The column names for the view being created can be modified by filling in the following form. If the Default button is pressed nothing will be changed.

Remove Column Prefix: ...> Add Column Prefix:

Remove Business Display Prefix: ...> Add Business Display Prefix:

Change Column Names for Specific Columns

Old Column Name:		New Column Name:
<input type="text" value="dim_date_key"/>	...>	<input type="text" value="dim_order_date_key"/>
<input type="text" value="calendar_date"/>	...>	<input type="text" value="order_date"/>
<input type="text"/>	...>	<input type="text"/>
<input type="text"/>	...>	<input type="text"/>
<input type="text"/>	...>	<input type="text"/>

- On the `dim_order_date` Properties dialog, change the View Type to **Dimension View** and click **OK**.

The screenshot shows the 'View dim_order_date' dialog box with the following fields and values:

- View Name: `dim_order_date`
- View Type: **Dimension View**
- Unique Short Name: `dim_order_date`
- Business Display Name (EUL): `dim_order_date`
- Description: (Empty text area)
- Update Procedure: (None)
- Custom Procedure: (None)
- Distinct Data Select:
- From/Where or Where Clause: (Empty text area)
- Table Locking Mode: **LOCK ROW FOR ACCESS**
- Mnemonic (EUL): (Empty text field)
- Timestamps:
 - Metadata Structure Changed: `2014-10-30 01:02:55.850000`
 - Database Created: `2014-10-30 01:03:50.330000`
 - Database Altered: `2014-10-30 01:03:50.330000`

Buttons: **Rebuild**, **Regenerate**, **OK**, **Cancel**, **Help**

- Click **Create View**.

The screenshot shows the 'Create Database View' dialog box with the following content:

Create Database View

View `dim_order_date` has been defined

Create View **Close**

- Repeat steps 1-7 to create the dimension view **dim_ship_date**. On the column Definition dialog, make the following changes:

View Column Definition

The column names for the view being created can be modified by filling in the following form. If the Default button is pressed nothing will be changed.

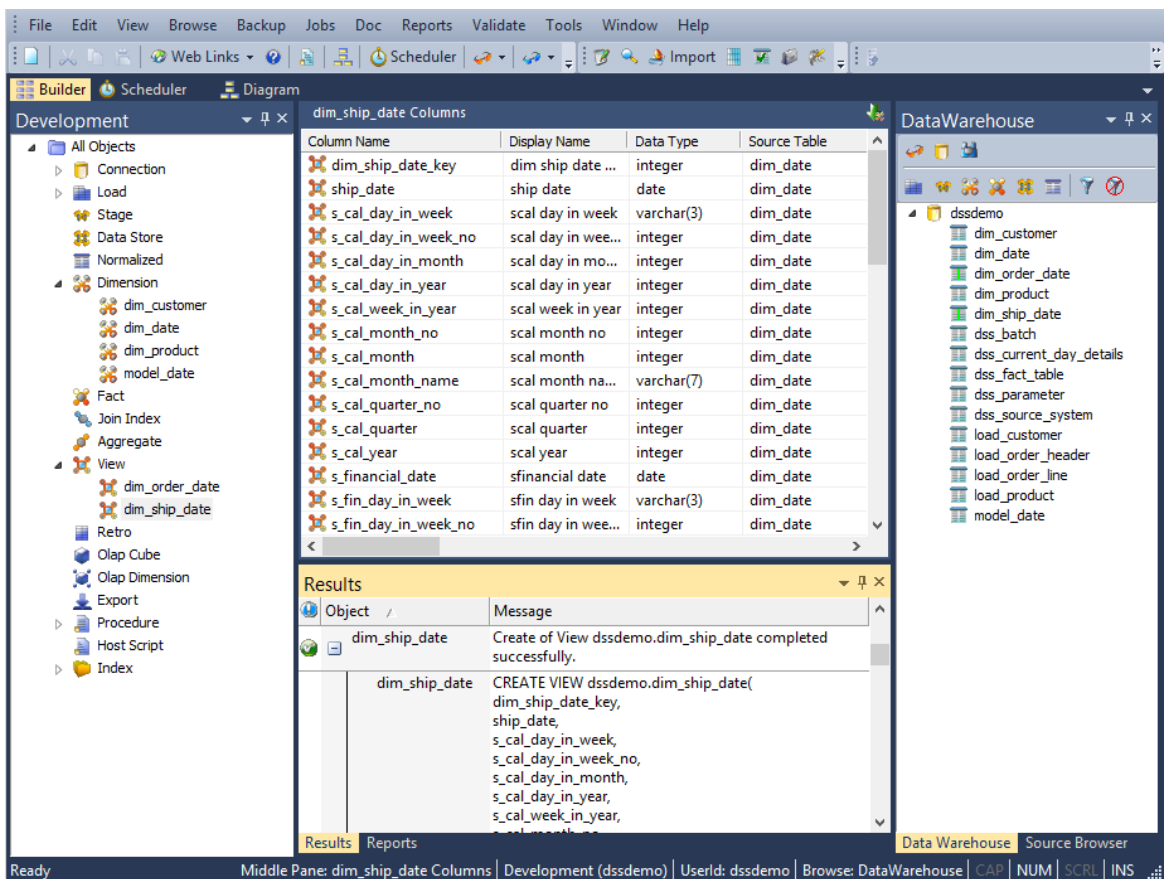
Remove Column Prefix:	--->	Add Column Prefix:
<input type="text"/>		<input type="text" value="s_"/>
Remove Business Display Prefix:	--->	Add Business Display Prefix:
<input type="text"/>		<input type="text" value="s"/>

Change Column Names for Specific Columns

Old Column Name:		New Column Name:
<input type="text" value="dim_date_key"/>	--->	<input type="text" value="dim_ship_date_key"/>
<input type="text" value="calendar_date"/>	--->	<input type="text" value="ship_date"/>
<input type="text"/>	--->	<input type="text"/>
<input type="text"/>	--->	<input type="text"/>
<input type="text"/>	--->	<input type="text"/>

OK Default

- Click in the right pane and press F5 to refresh the Data Warehouse table view in the right pane. Your screen should look something like this:



You are now ready to proceed to the next step - *Defining the Staging Table* (see "1.7 Defining the Staging Table" on page 48).

- Click and drag the load_order_header table from the right hand pane data warehouse schema. Drop it in the middle pane.
A dialog box displays defaulting the name of the object to stage_order_header.
To make it a more meaningful name, change the name of the object to **stage_sales_detail** and click **ADD**.

Add a New Metadata Object

Define the Type and Name of the New Object.
Specific information for each object type is defined in subsequent screens.

Object Type: Stage Table

Object Name: stage_sales_detail

Target Location: (local)

ADD Cancel

- A table definition displays with all the required defaults completed. Click **OK**

Stage stage_sales_detail

Properties

Storage

Override Create DDL

Notes

Table Name: stage_sales_detail Table Type: Stage

Unique Short Name: (maximum 22 characters) stage_sales_detail

Description:

Update Procedure: (None) Rebuild Regenerate

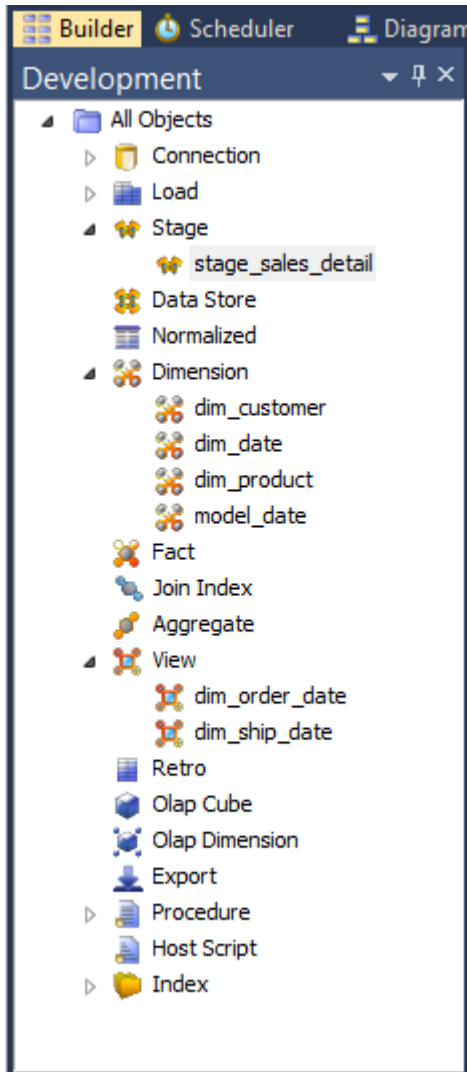
Custom Procedure: (None)

Timestamps

Metadata Structure Changed: 2014-12-03 03:26:57.450000 Database Created: Database Altered:

OK Cancel Help

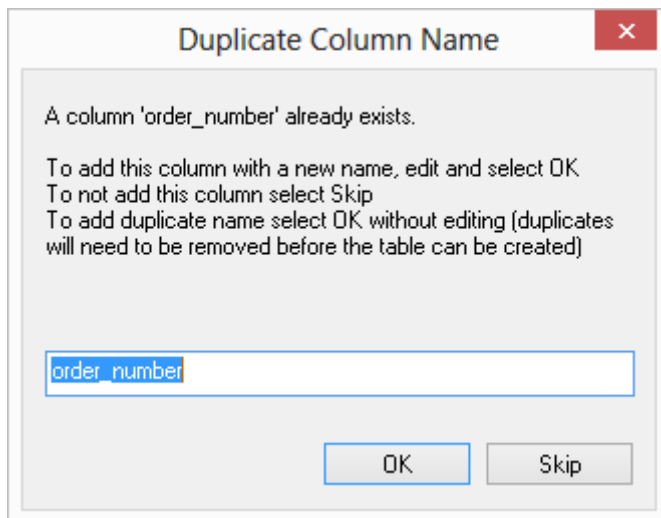
- 4 The Stage Table object in the left pane now has a dependent/child.



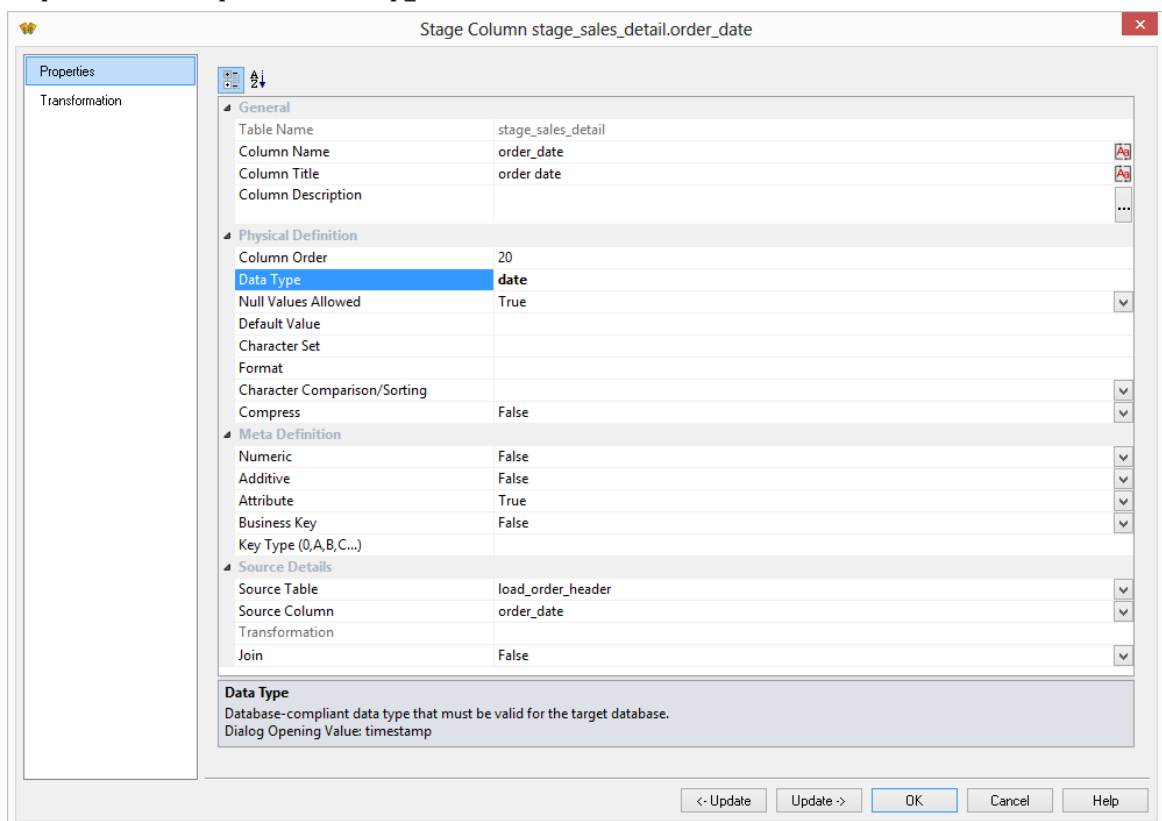
- 5 To add the remaining information from the second load table, click on stage_sales_detail in the left pane. Next drop load_order_line from the right pane and into the middle pane. A message is displayed with options to create a 'New Table' or to 'Add Columns'. Click **Add Columns**.



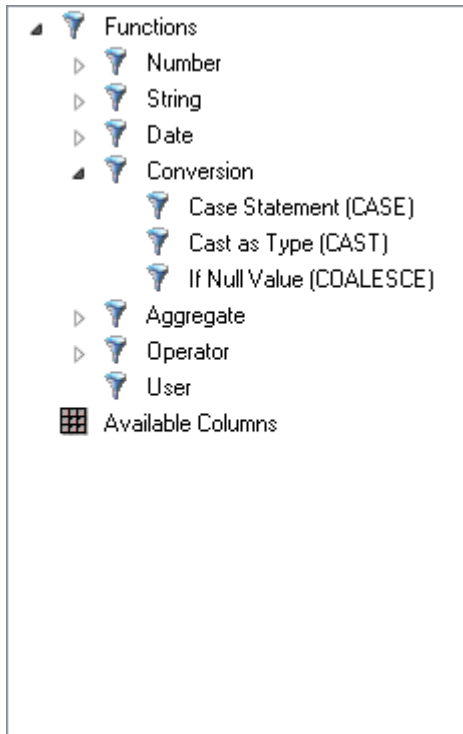
- WhereScape RED detects duplicate columns. As both `load_order_header` and `load_order_line` have the `order_number` field, the following is displayed. Click **Skip** to exclude the second instance of `order_number` - this combines data from two load tables (`load_order_header` and `load_order_line`) into one stage table.



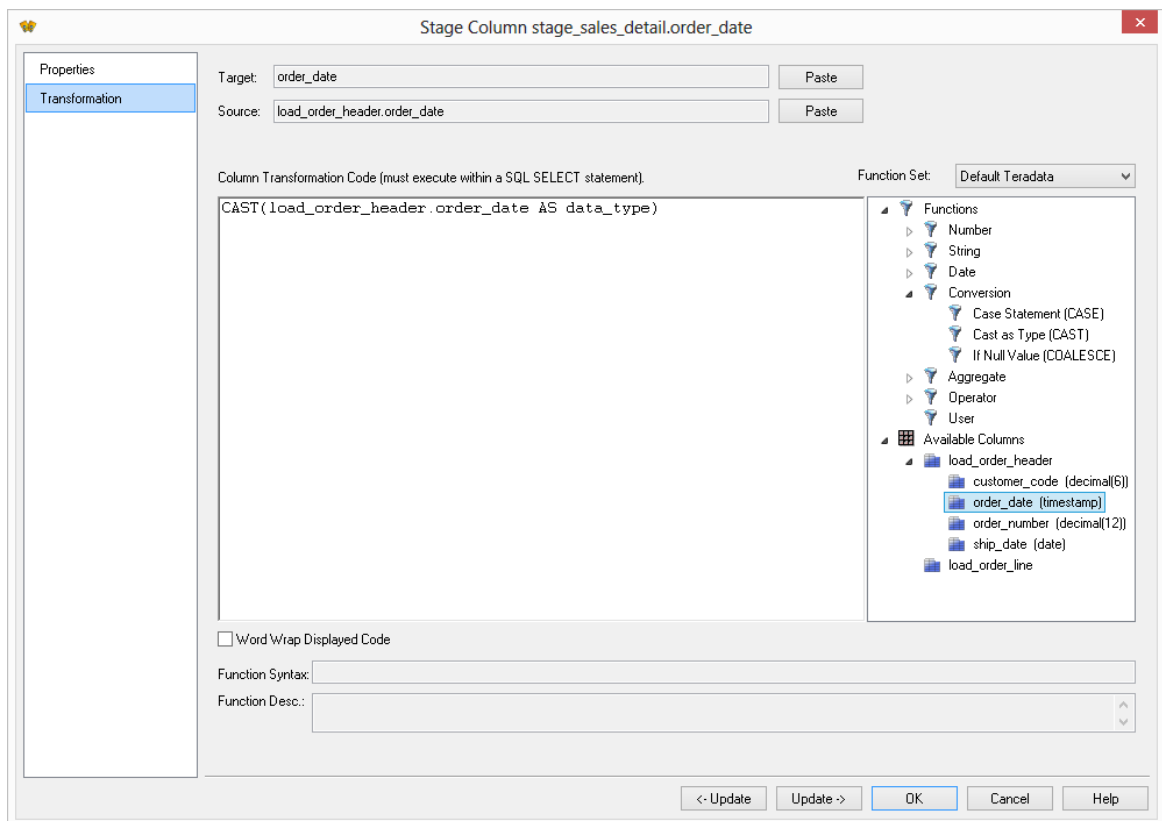
- In the middle pane, double-click on `order_date` to bring up the Properties screen for that column and change the Data Type to `date`.
- Repeat this same process for `ship_date`.



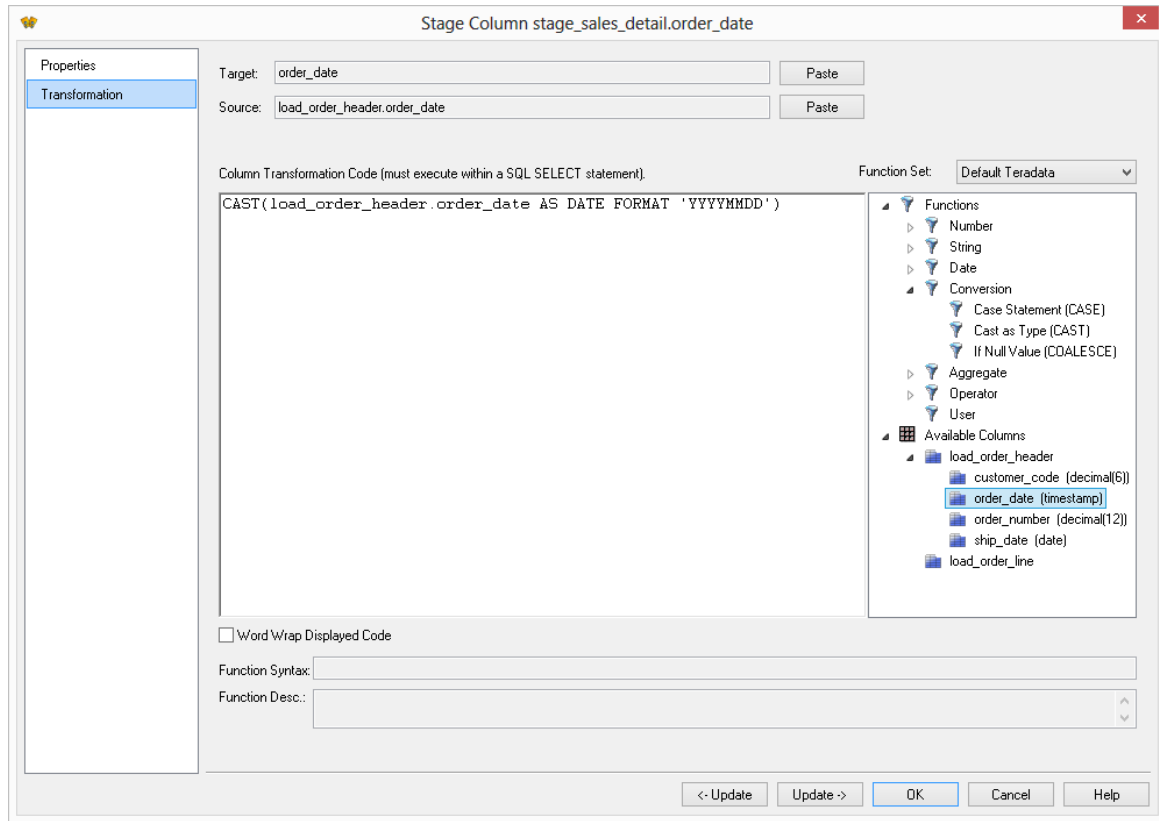
- 8 Click on the 'Transformation' tab, double-click on Functions, double-click on Conversion and then double-click on Cast as Type (CAST) to add the Cast function to the 'Column Transformation code' in the middle pane.



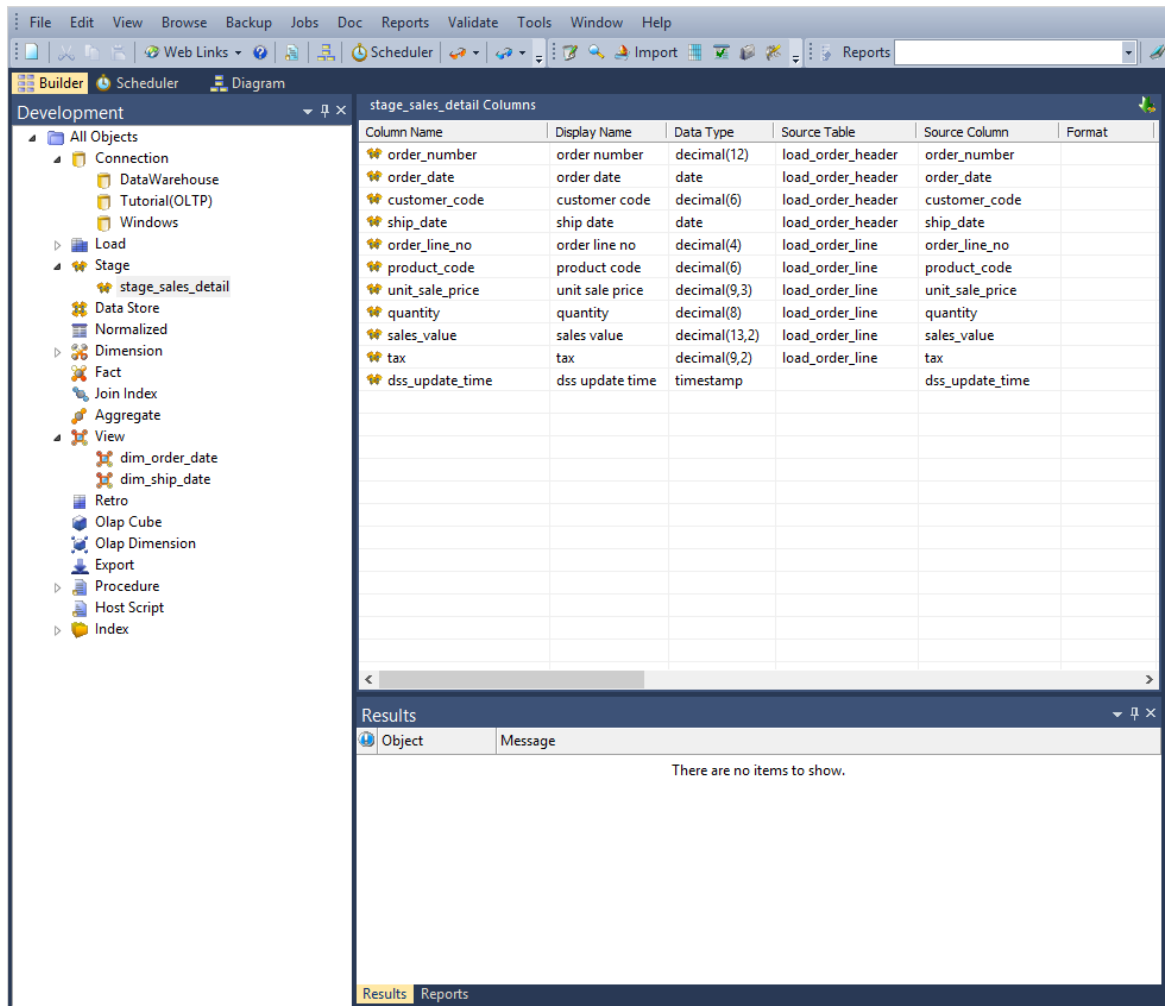
- 9 Double-click on Available Columns, load_order_header and then order_date to enter the correct column in the 'Column Transformation Code' in the middle pane.



10 Finally, replace 'data_type' with DATE FORMAT 'YYYYMMDD' and click OK.



Your screen should look something like this:



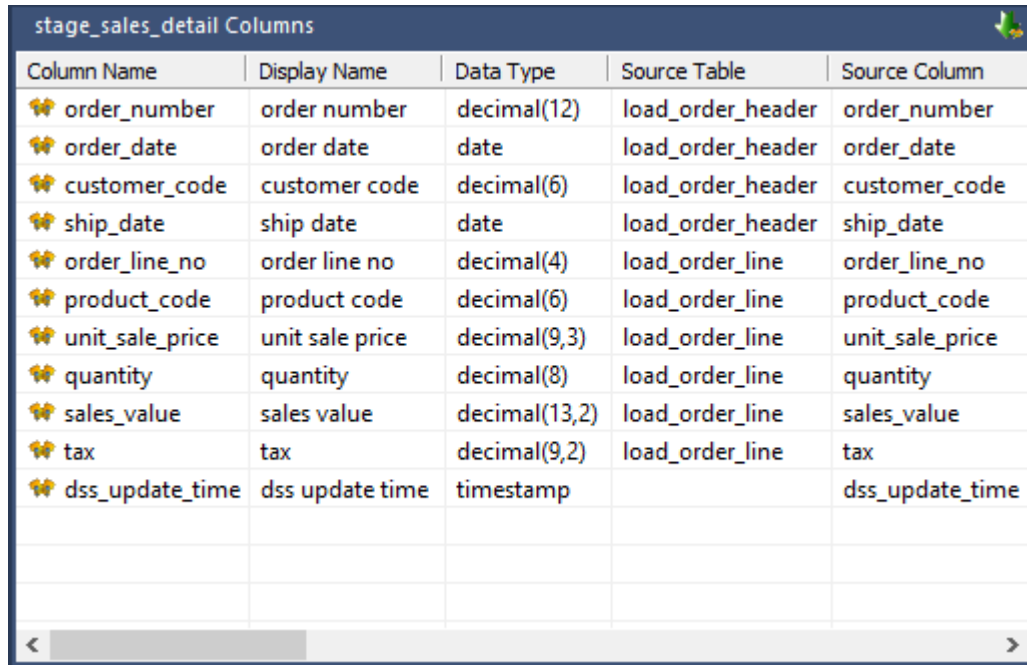
In the middle pane under Source Table you can see the source of each of the columns.

You are now ready to proceed to the next step - ***Including Dimension Links*** (see "1.8 ***Including Dimension Links***" on page 56).

1.8 INCLUDING DIMENSION LINKS

The dimension links that allow us to create the fact-like star schema now need to be included:

- 1 In the left pane, click on the **stage_sales_detail** table in the Stage Table object group. The middle pane should display the contents of this stage table.



Column Name	Display Name	Data Type	Source Table	Source Column
🔥 order_number	order number	decimal(12)	load_order_header	order_number
🔥 order_date	order date	date	load_order_header	order_date
🔥 customer_code	customer code	decimal(6)	load_order_header	customer_code
🔥 ship_date	ship date	date	load_order_header	ship_date
🔥 order_line_no	order line no	decimal(4)	load_order_line	order_line_no
🔥 product_code	product code	decimal(6)	load_order_line	product_code
🔥 unit_sale_price	unit sale price	decimal(9,3)	load_order_line	unit_sale_price
🔥 quantity	quantity	decimal(8)	load_order_line	quantity
🔥 sales_value	sales value	decimal(13,2)	load_order_line	sales_value
🔥 tax	tax	decimal(9,2)	load_order_line	tax
🔥 dss_update_time	dss update time	timestamp		dss_update_time

- 2 Drag each of the following dimensions from the right pane into the stage table in the middle pane:
 - **dim_customer**
 - **dim_product**
 - **dim_order_date**
 - **dim_ship_date**

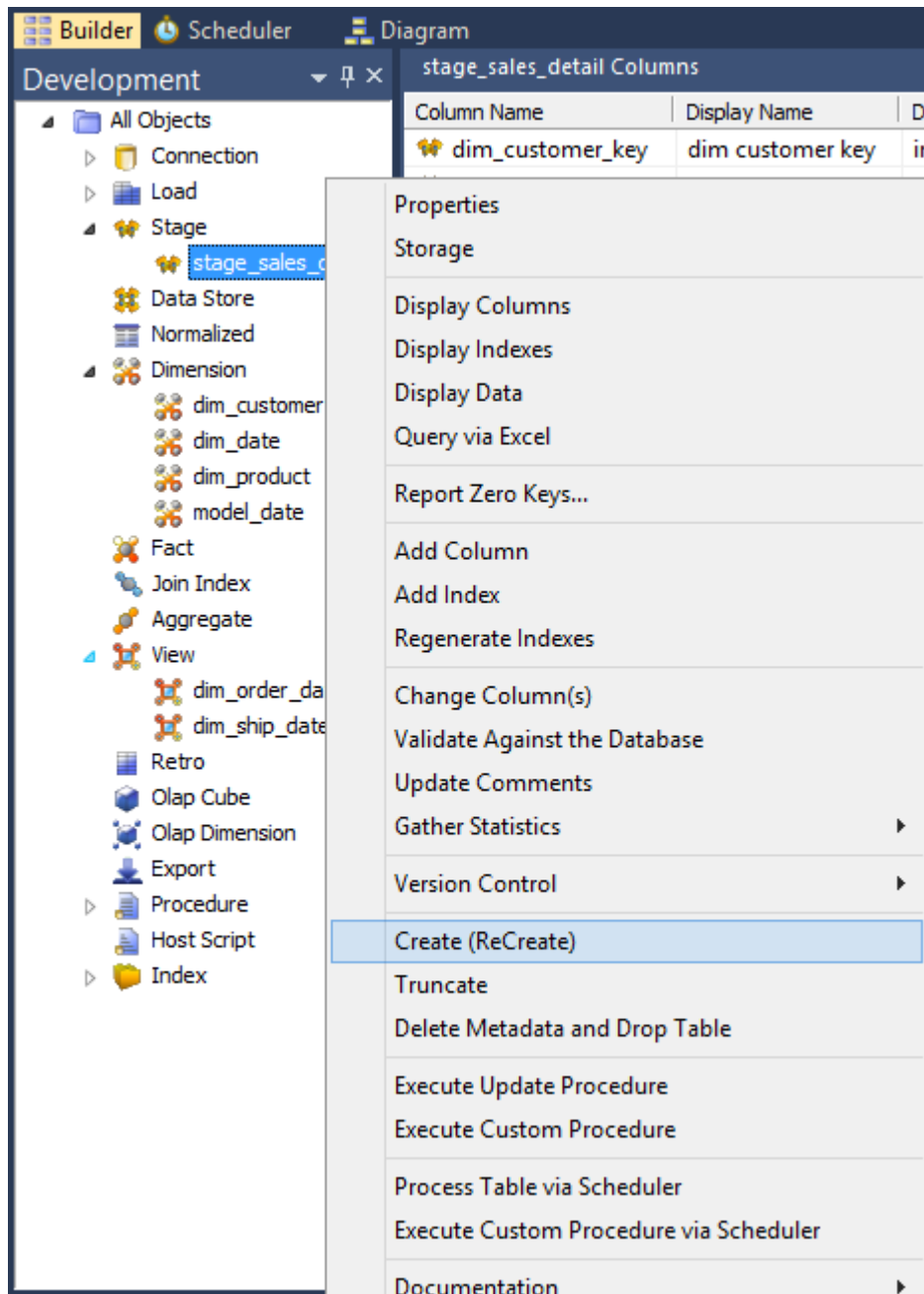
- 3 This adds the dimension keys from each dimension to the stage table. Your screen should look something like this:

The screenshot shows the WhereScape Builder interface. The main pane displays a table of columns for the 'stage_sales_detail' stage. The table has columns for Column Name, Display Name, Data Type, Source Table, and Source Column. The Results pane shows comments for the 'dim_ship_date' column.

Column Name	Display Name	Data Type	Source Table	Source Column
dim_customer_key	dim customer key	integer	dim_customer	dim_customer_key
dim_product_key	dim product key	integer	dim_product	dim_product_key
dim_order_date_key	dim order date key	integer	dim_order_date	dim_order_date_key
dim_ship_date_key	dim ship date key	integer	dim_ship_date	dim_ship_date_key
order_number	order number	decimal(12)	load_order_header	order_number
order_date	order date	date	load_order_header	order_date
customer_code	customer code	decimal(6)	load_order_header	customer_code
ship_date	ship date	date	load_order_header	ship_date
order_line_no	order line no	decimal(4)	load_order_line	order_line_no
product_code	product code	decimal(6)	load_order_line	product_code
unit_sale_price	unit sale price	decimal(9,3)	load_order_line	unit_sale_price
quantity	quantity	decimal(8)	load_order_line	quantity
sales_value	sales value	decimal(13,2)	load_order_line	sales_value
tax	tax	decimal(9,2)	load_order_line	tax
dss_update_time	dss update time	timestamp		dss_update_time

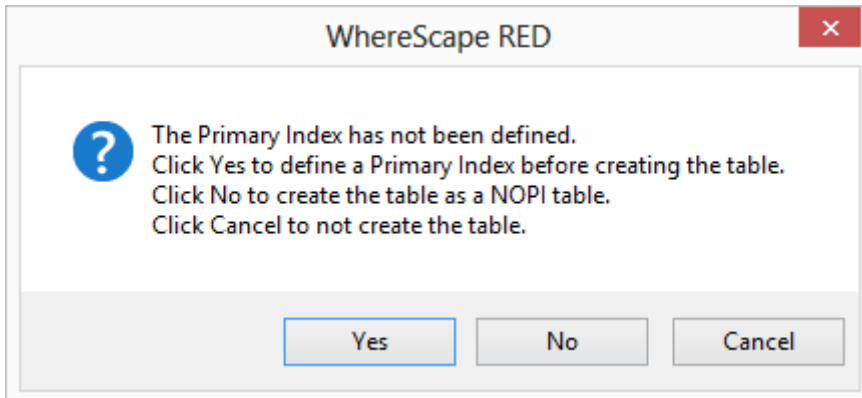
Object	Message
dim_ship_date	COMMENT ON COLUMN dssdemo.dim_ship_date.s_holiday_flag IS 'Flag to indicate that the day in question is a holiday. Y=holiday, N=normal.';
dim_ship_date	COMMENT ON COLUMN dssdemo.dim_ship_date.s_fin_quarter_no IS 'Financial quarter number (1-4).';
dim_ship_date	COMMENT ON COLUMN dssdemo.dim_ship_date.s_week_day_flag IS 'Flag to indicate if a week day Y=week day N=week end.';
dim_ship_date	COMMENT ON COLUMN dssdemo.dim_ship_date.s_moving_fin_quarter IS 'Flag to indicate days that constitute a quarter counting back from the current financial day. Set as per current fin. day.';

- The stage table metadata has been defined, but the stage table has not been created. To create the stage table in the data warehouse, right-click on **stage_sales_detail** in the left pane and select **Create (ReCreate)**.

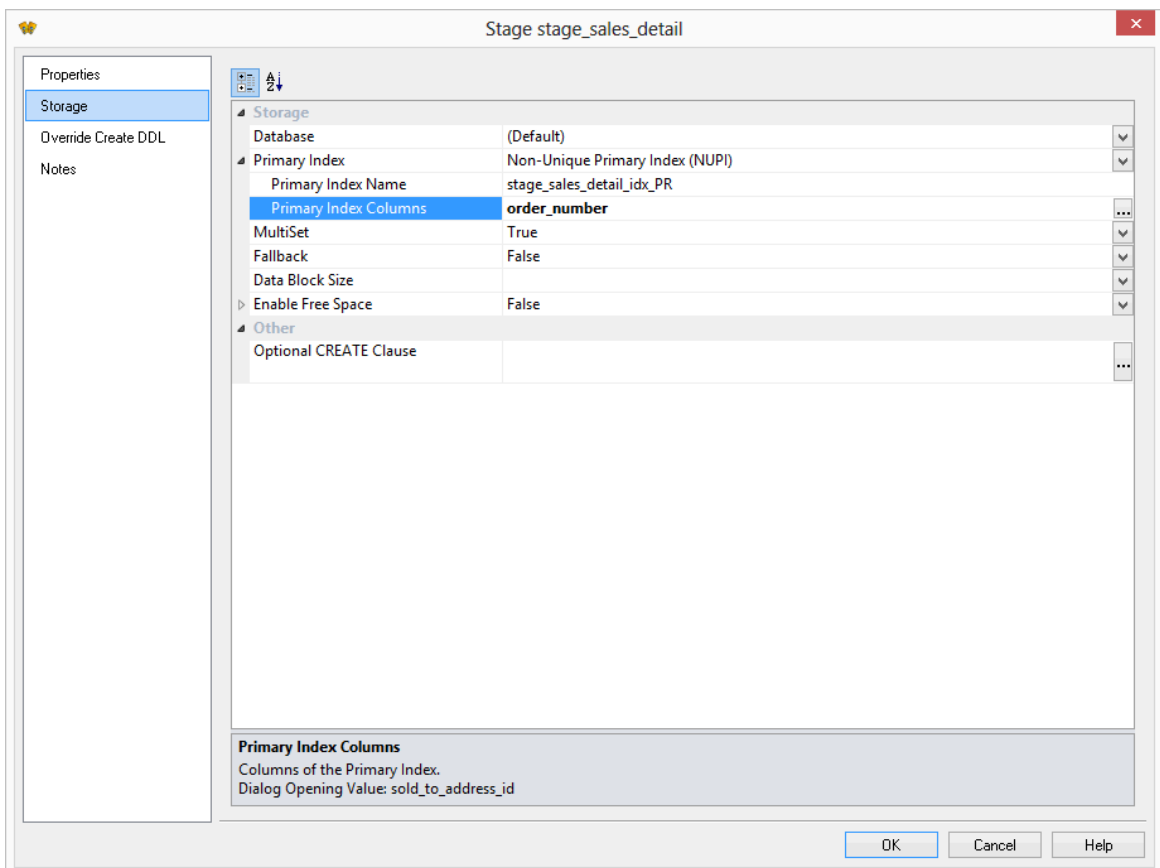


Note: The table must exist in the data warehouse before we can proceed to the next step. If the table has not been physically created then the procedure in step 5 will fail to compile.

- 5 Click **Yes** to define a Primary Index.



- 6 Type **order_number** as the Primary Index and click **OK** on the storage screen.



7 Right-click on **stage_sales_detail**, choose **Code** and then **Build Update Procedure**.

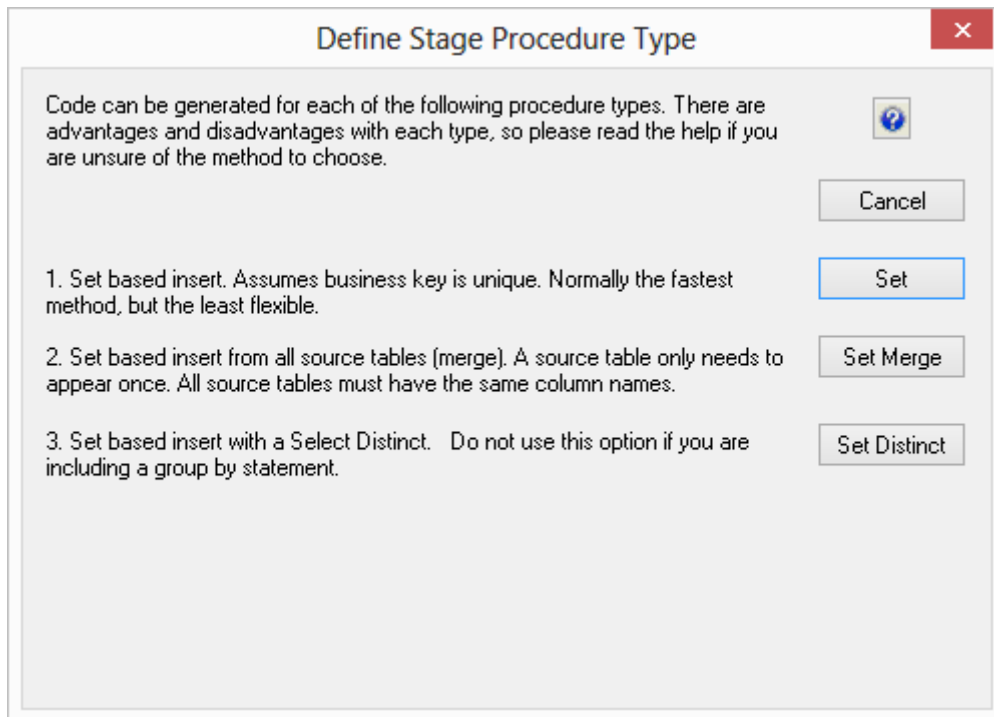
The screenshot shows the 'Development' environment with a tree view on the left and a 'stage_sales_detail Columns' table on the right. A context menu is open over the 'stage_sales_detail' stage, listing various actions. The 'Code' option is selected, and its sub-menu is visible, with 'Build Update Procedure' highlighted.

Column Name	Display Name	Data Type	Source Table	Source C
dim_customer_key	dim customer key	integer	dim_customer	dim_cu:
dim_product_key	dim product key	integer	dim_product	dim_pro
dim_order_date_key	dim order date key	integer	dim_order_date	dim_orc
		er	dim_ship_date	dim_shi
		nal(12)	load_order_header	order_n
			load_order_header	order_d
		nal(6)	load_order_header	custom
			load_order_header	ship_da
		nal(4)	load_order_line	order_li
		nal(6)	load_order_line	product
		nal(9,3)	load_order_line	unit_sal
		nal(8)	load_order_line	quantity
		nal(13,2)	load_order_line	sales_va
		nal(9,2)	load_order_line	tax
		stamp		dss_upc

Context Menu Options:

- Properties
- Storage
- Display Columns
- Display Indexes
- Display Data
- Query via Excel
- Report Zero Keys...
- Add Column
- Add Index
- Regenerate Indexes
- Change Column(s)
- Validate Against the Database
- Update Comments
- Gather Statistics
- Version Control
- Create (ReCreate)
- Truncate
- Delete Metadata and Drop Table
- Execute Update Procedure
- Execute Custom Procedure
- Process Table via Scheduler
- Execute Custom Procedure via Scheduler
- Documentation
- Projects
- Impact
- Code**
 - Build Update Procedure**

- 8 Choose the **Set** based procedure generation from the stage procedure type dialog box.



- 9 Click **OK** on the Parameters dialog.

10 On the Source Table Mapping screen, highlight the two tables in the left pane and click **Outer Join**.

- Select order_number from the load_order_header empty drop-down box at the bottom of the screen.
- Again select order_number from the load_order_line drop-down list box. This will create a join statement in the right pane.
- Click **OK**.

Source Table Mapping

Define the joins (or edit the from and where clause).
To define a Join select two tables and press the join type. Then select the join columns from the column lists presented.

Source Tables:

- [load_order_header]
- [load_order_line]

From and Where Clause:

```
FROM [load_order_header] load_order_header  
LEFT OUTER JOIN [load_order_line] load_order_line  
ON load_order_header.order_number = load_order_line.order_number
```

Outer Join Simple Join ANSI join code generated

Select the columns that join the two tables. Select the column from the Master Table first.

[load_order_header] [load_order_line]

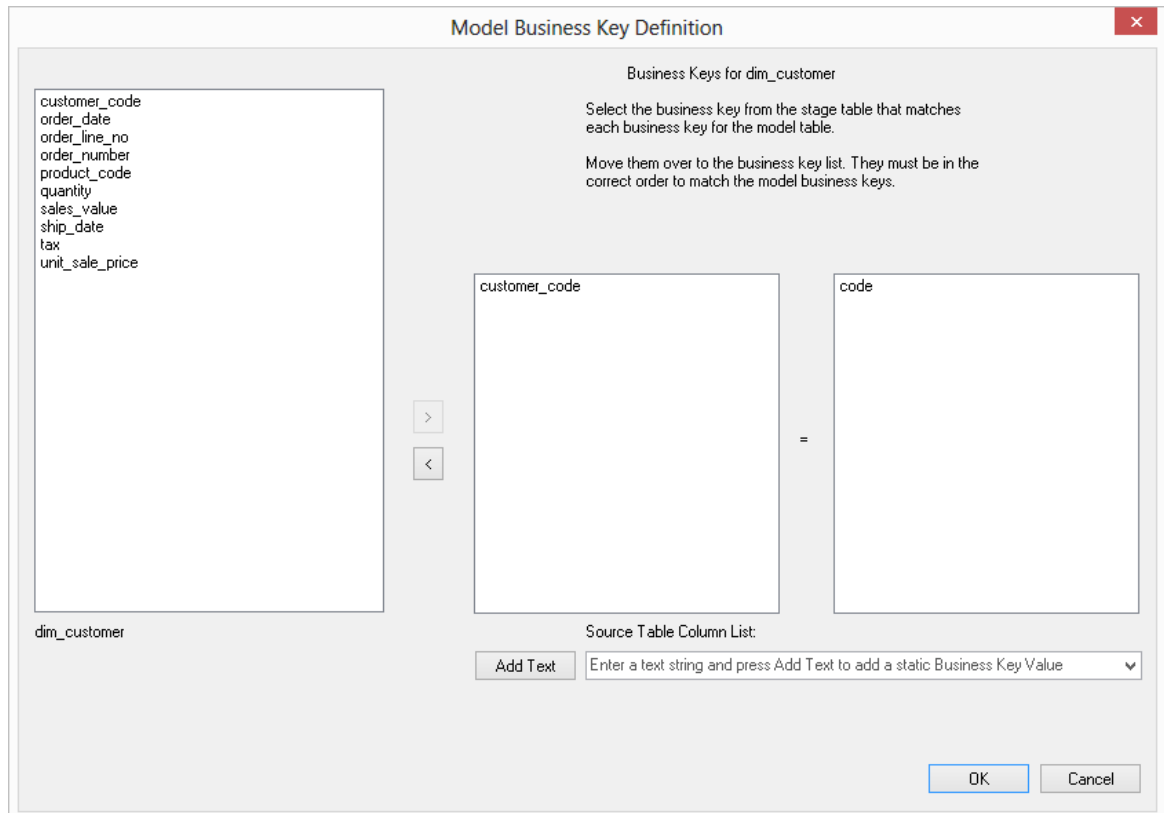
Word Wrap Displayed Code

OK Cancel

11 You need to match the dimension business keys with the business keys in the stage table. This associates the correct dimensional record to each stage table record.

A dialog box displays for each dimensional join:

- For the **dim_customer**, select **customer_code**.
- Click **>** and **OK**.

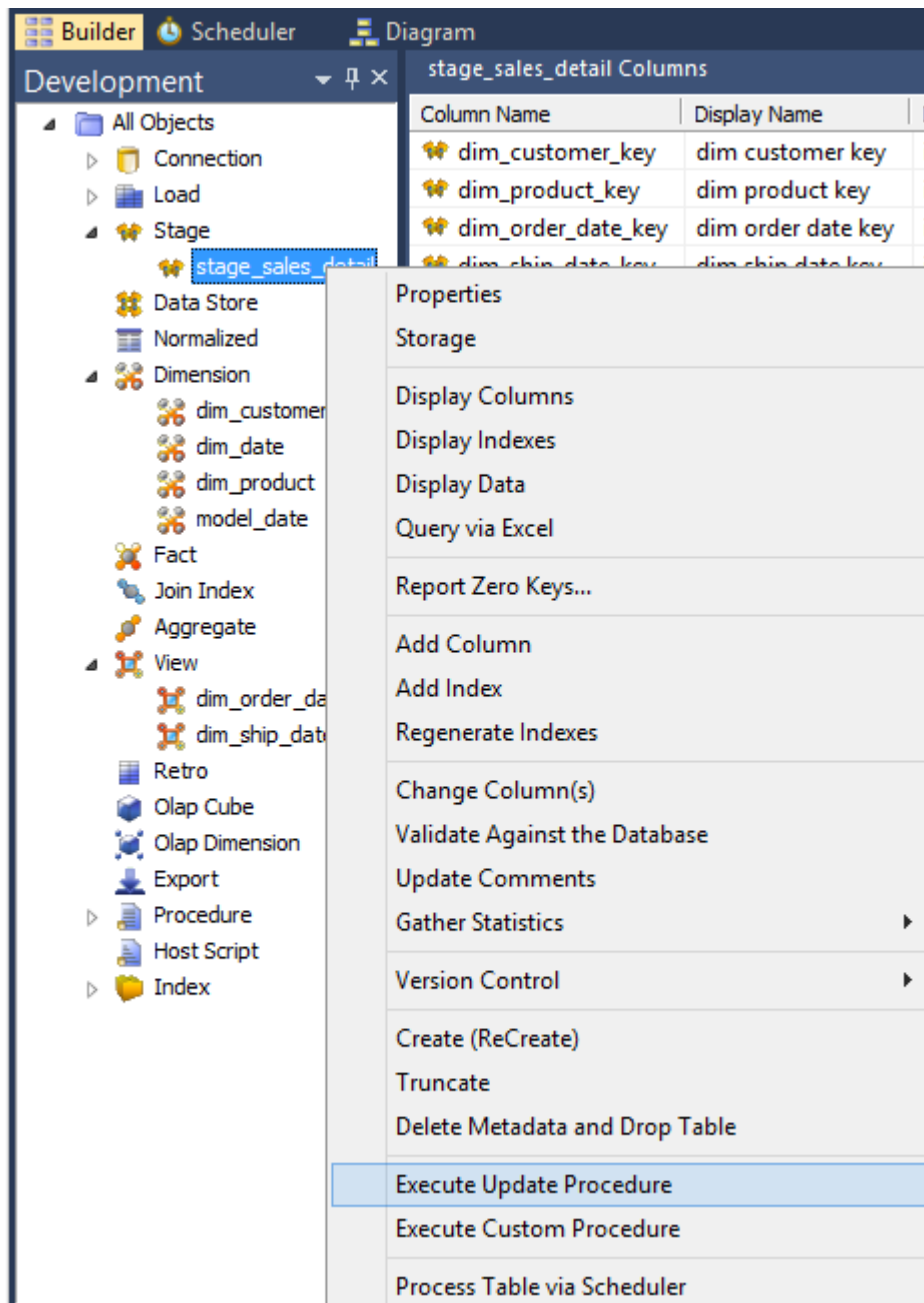


12 For **dim_product**, select **product_code**. Click **>** and **OK**.

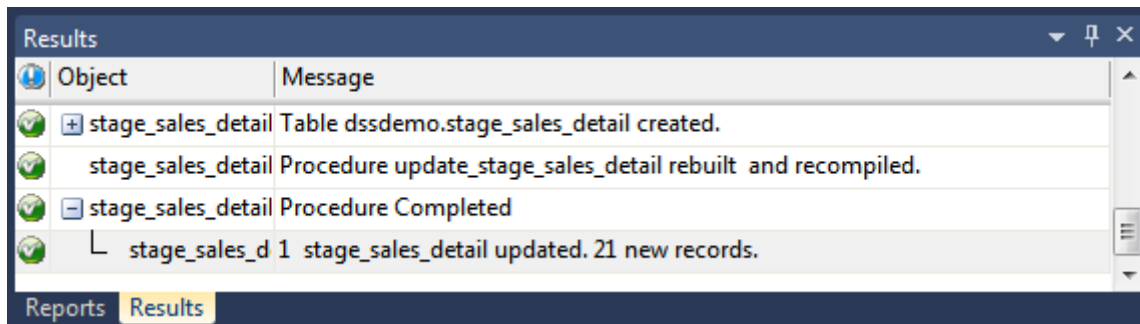
- The business key for dim_order_date has the same column name in the stage table and the dimension view, allowing WhereScape RED to automatically move **order_date** to the left hand side.
- Click **OK** to progress to dim_ship_date, where ship_date has also been automatically chosen.
- Click **OK** again.

13 The final step is the population of the stage table.

- Click on **stage_sales_detail** in the left pane, right-click and select **Execute Update Procedure**.



- 14 The output from the stage table being updated can now be seen in the **Results** window:

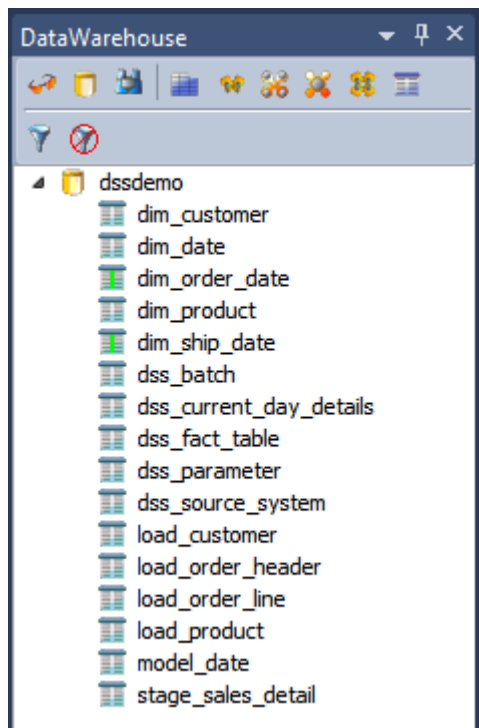


You are now ready to proceed to the next step - **Creating a Fact Table** (see "1.9 Creating a **Fact Table**" on page 66).

1.9 CREATING A FACT TABLE

In this step you will create a fact table.

- 1 Click in the right pane and press F5 to refresh the Data Warehouse table view in the right pane.



- The `fact_sales_detail` table Properties screen will appear. Select **(Build Procedure...)** in the update procedure drop-down and click **OK**.

The screenshot shows the 'Fact fact_sales_detail' Properties dialog box. On the left is a navigation pane with the following items: Properties (selected), Storage, Override Create DDL, Language Mapping, Purpose, Concept, Grain, Examples, Usage, and Notes. The main area contains the following fields and controls:

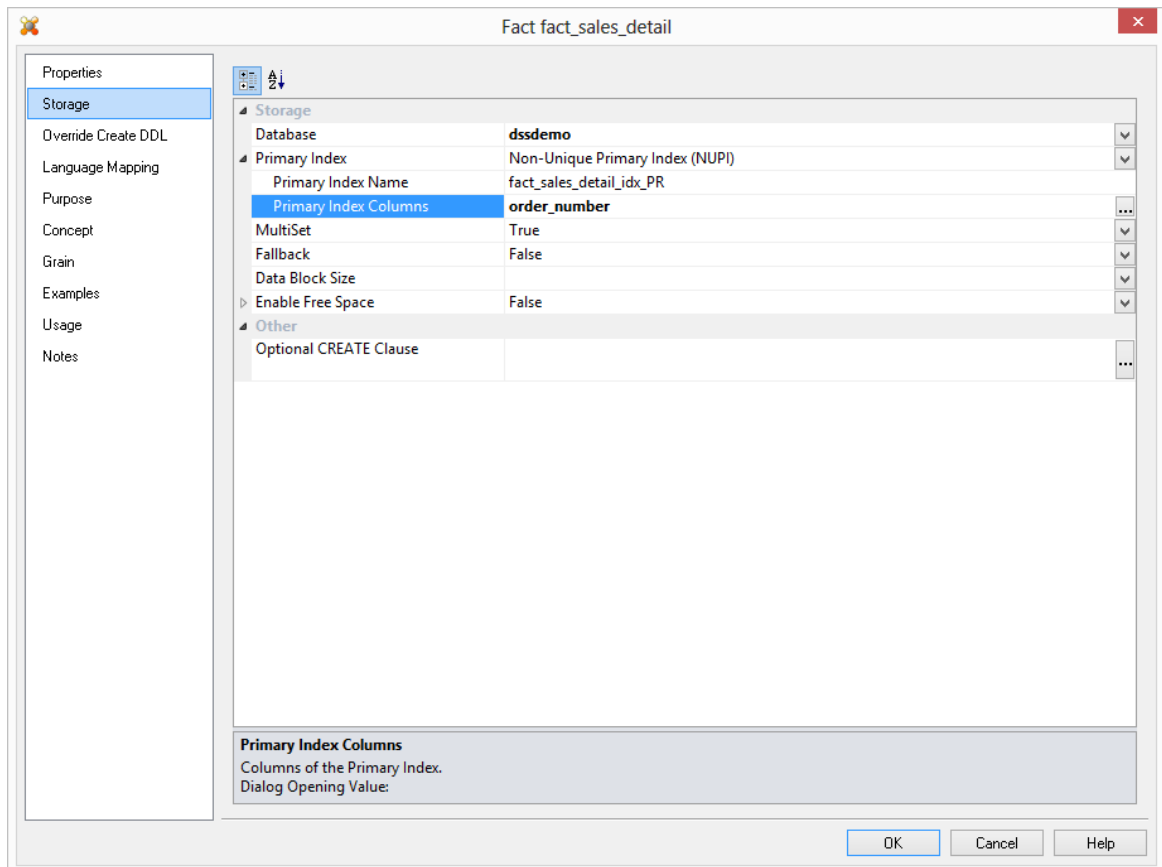
- Table Name: `fact_sales_detail`
- Table Type: `Detail`
- Unique Short Name: `fact_sales_detail` (maximum 22 characters)
- Business Display Name (EUL): `fact_sales_detail`
- Description: (empty text area)
- Update Procedure: `(Build Procedure...)` (dropdown menu)
- Custom Procedure: `(None)` (dropdown menu)
- Buttons: `Rebuild`, `Set Based Update`
- Get Key Function: `(None)` (dropdown menu) with `Edit` button
- Mnemonic (EUL): (empty text field)
- Timestamps section:
 - Metadata Structure Changed: `2014-10-30 01:19:51.720000`
 - Database Created: `2014-10-30 01:20:14.640000`
 - Database Altered: `2014-10-30 01:20:14.640000`
- Buttons at the bottom: `OK`, `Cancel`, `Help`

- Select **Create and Load** to create and load the table now:

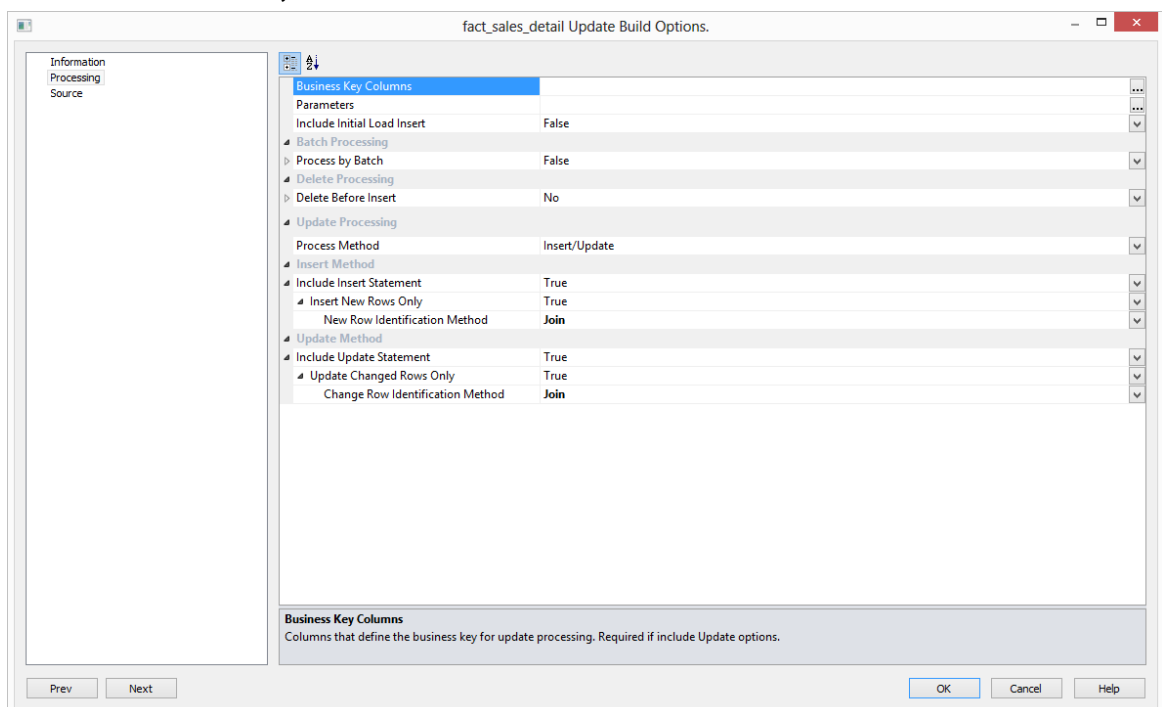
The screenshot shows the 'Create Database Table' dialog box. The title bar reads 'Create Database Table'. The main area contains the message: 'Fact fact_sales_detail has been defined'. At the bottom, there are three buttons: `Create`, `Create and Load` (highlighted with a blue border), and `Close`.

- Click **Yes** to define a Primary Index on the next dialog.

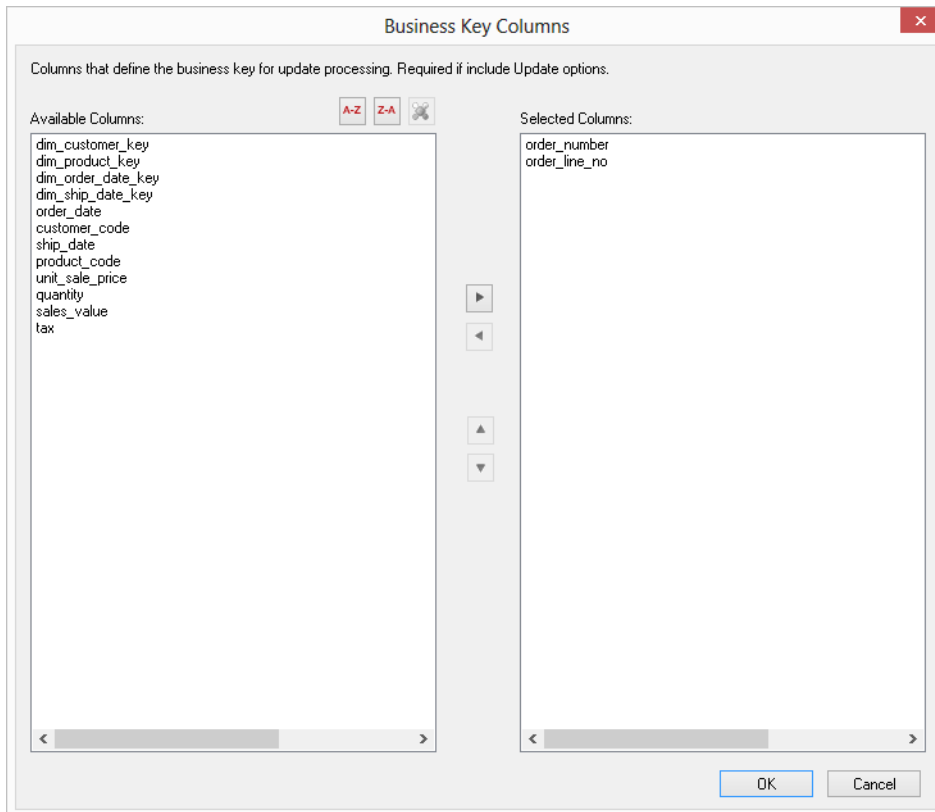
- 7 Type **order_number** as the Primary Index and click **OK**.



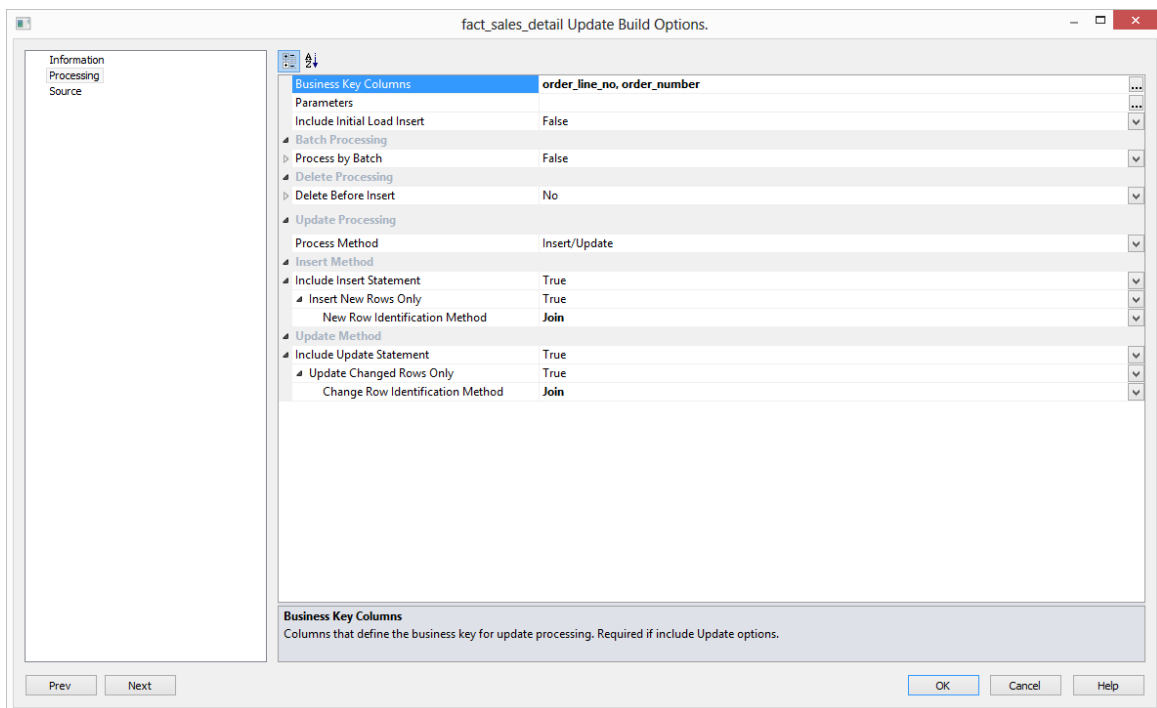
- 8 Select the **Business Key** for the fact table by clicking on the ellipsis button on the right hand side of the Business Key Columns field.



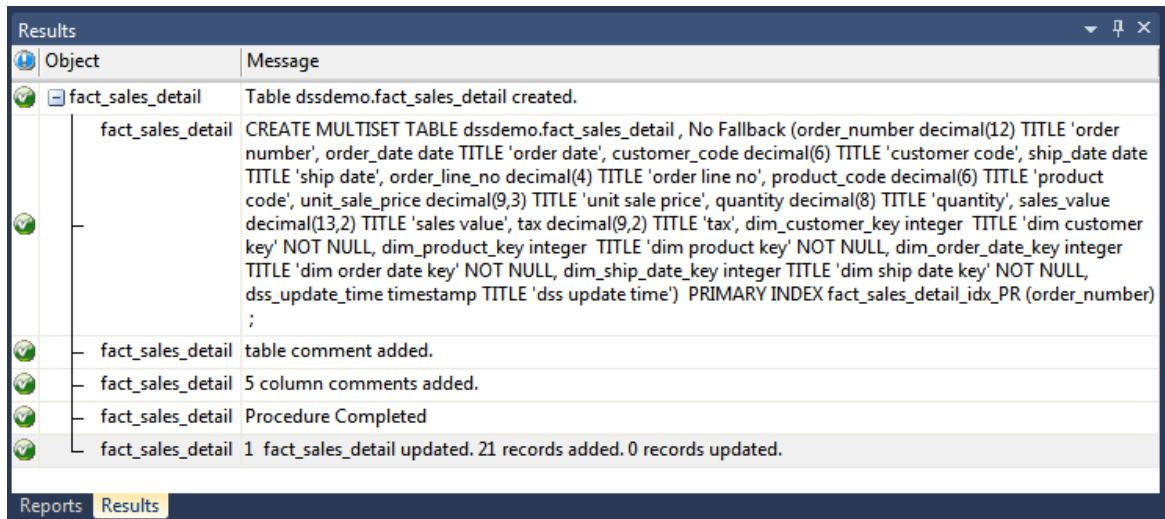
- Choose **order_number** and **order_line_no**.
Click > and then **OK**.



- Click **OK** on the **Update Build Options** screen.



- 11 The output from the fact table being created and updated can now be seen in the **Results** window.

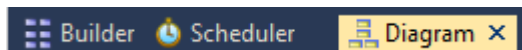


You are now ready to proceed to the next step - **Switching to Diagrammatic View** (see "1.10 Switching to Diagrammatic View" on page 72).

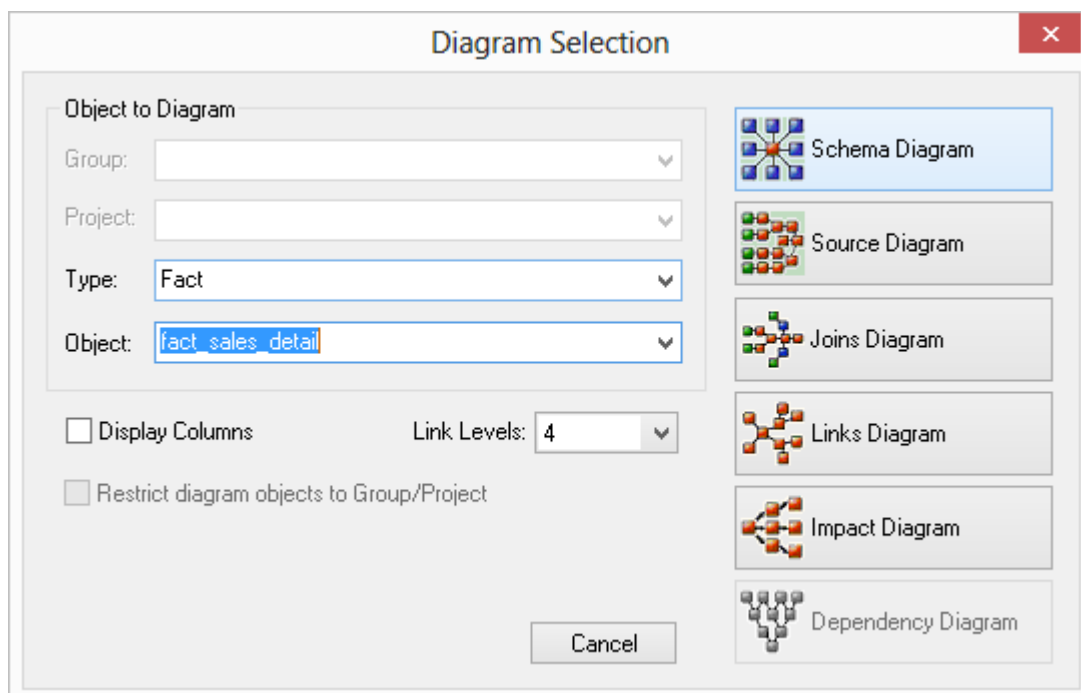
1.10 SWITCHING TO DIAGRAMMATIC VIEW

WhereScape RED provides the ability to diagrammatically view the data warehouse you have created.

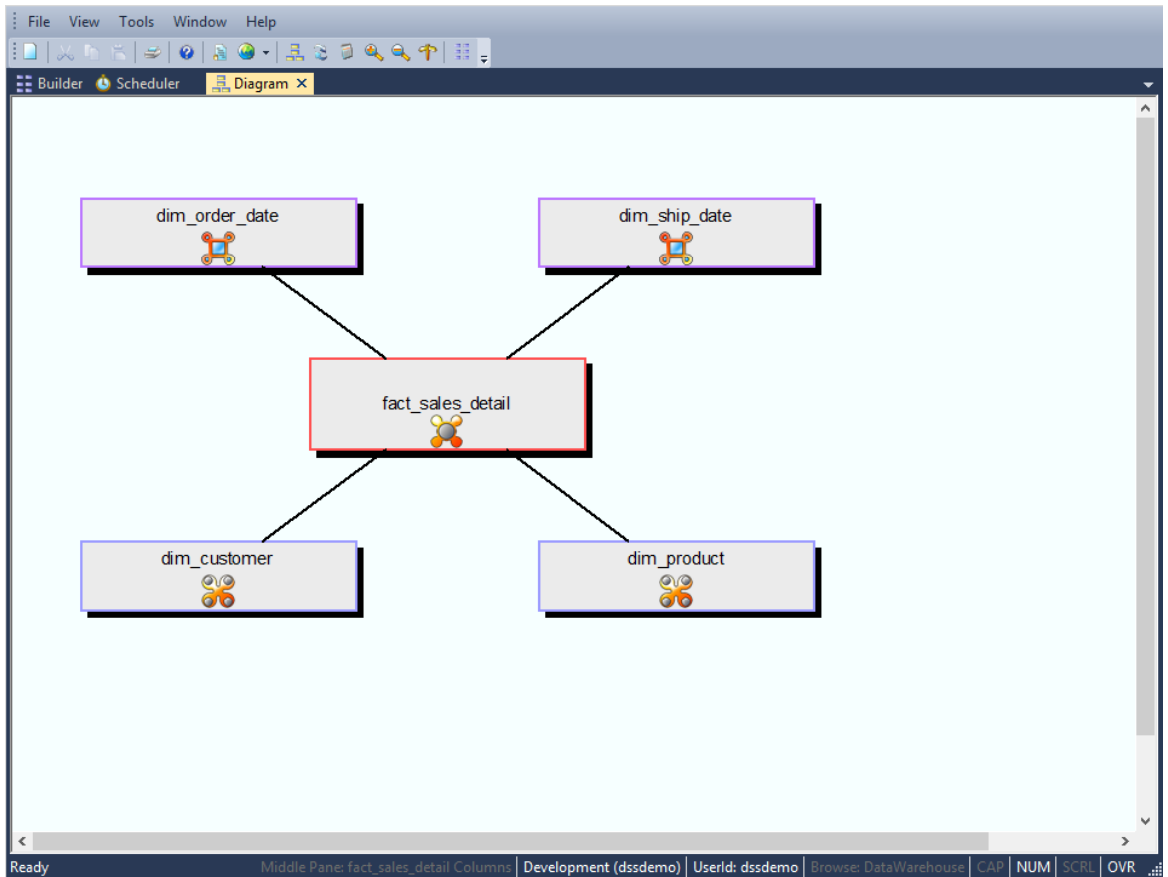
- 1 Click on the Diagram tab to display the **Diagram Selection** dialog.




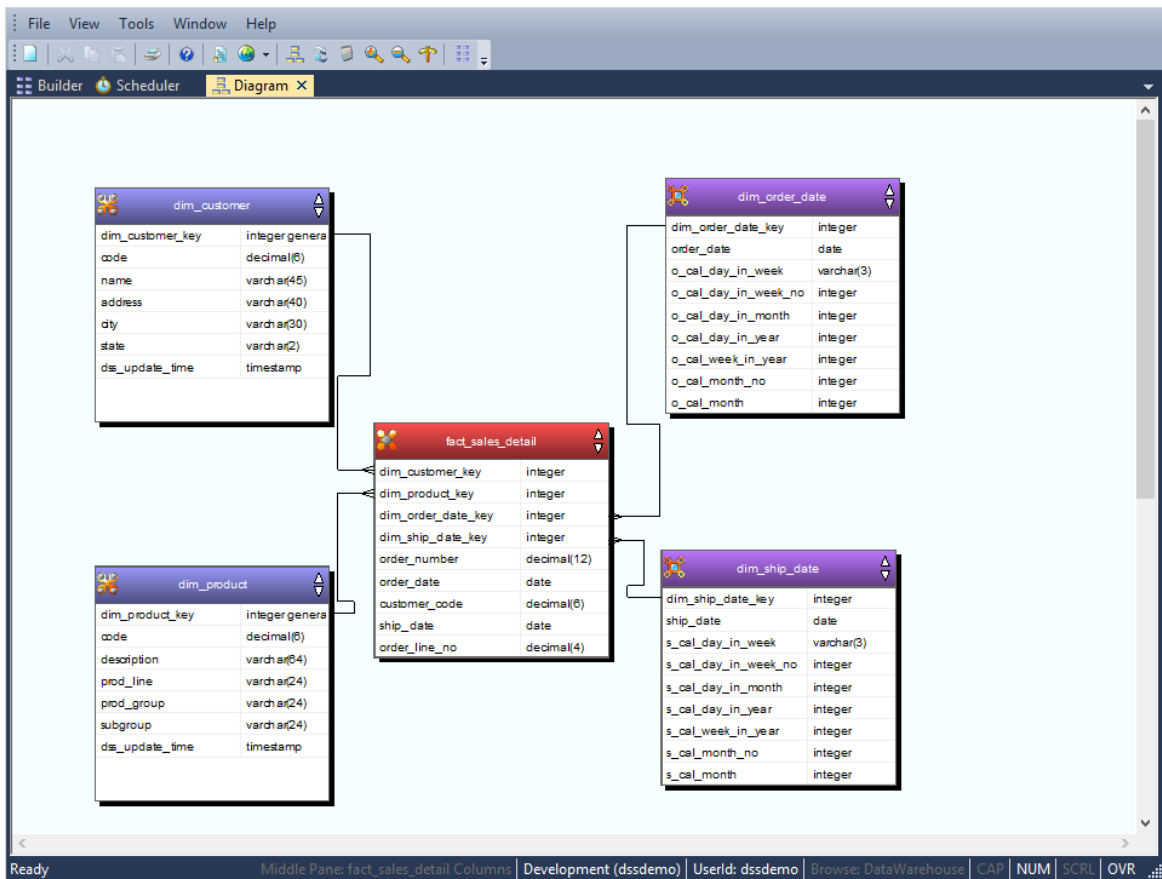
- 2 Select an object **Type** of **Fact** to narrow the selection list and then select **fact_sales_detail**. Click on the **Schema Diagram** button to display a star schema diagram.




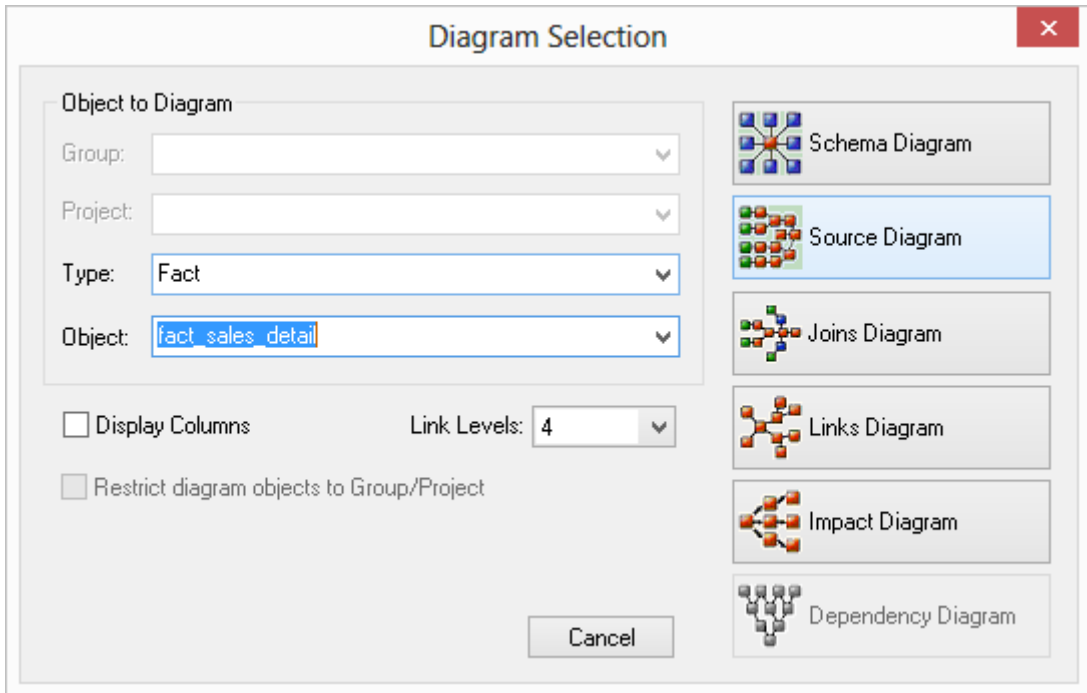
3 The diagram will be displayed.



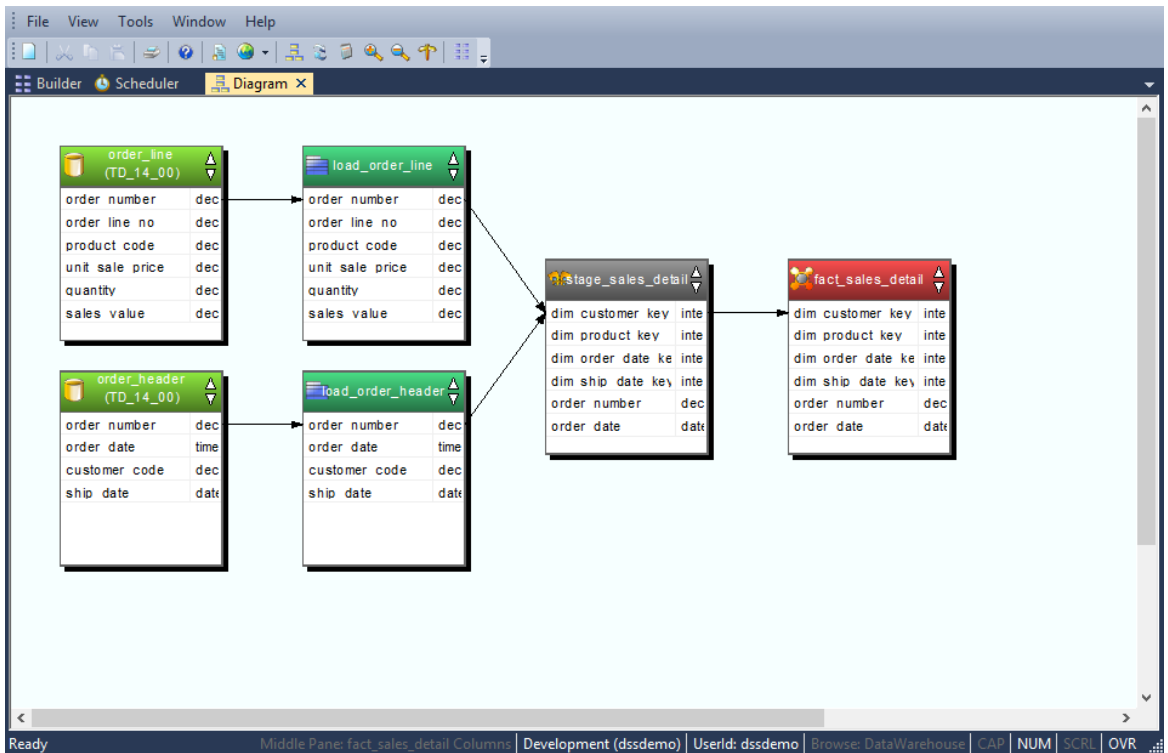
- 4 To view a more detailed diagram, click on the Toggle button  You will need to use the Zoom In button to see the diagram more clearly.



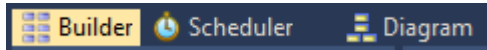
TIP: To view the source tracking of the **fact_sales_detail** table, click once more on the  button, choose the **fact_sales_detail** table and then click on the **Source Diagram** button.



5 The diagram will be displayed.



- 6 To close the diagrammatic view, click on the **X** on the diagram tab, or alternatively, return to the Builder section by clicking the Builder tab.

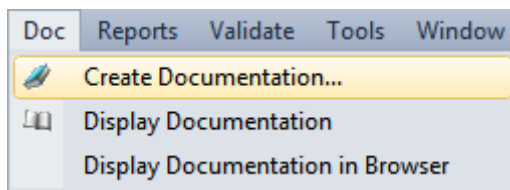


You are now ready to proceed to the next step - ***Producing Documentation*** (see "***1.11 Producing Documentation***" on page 77).

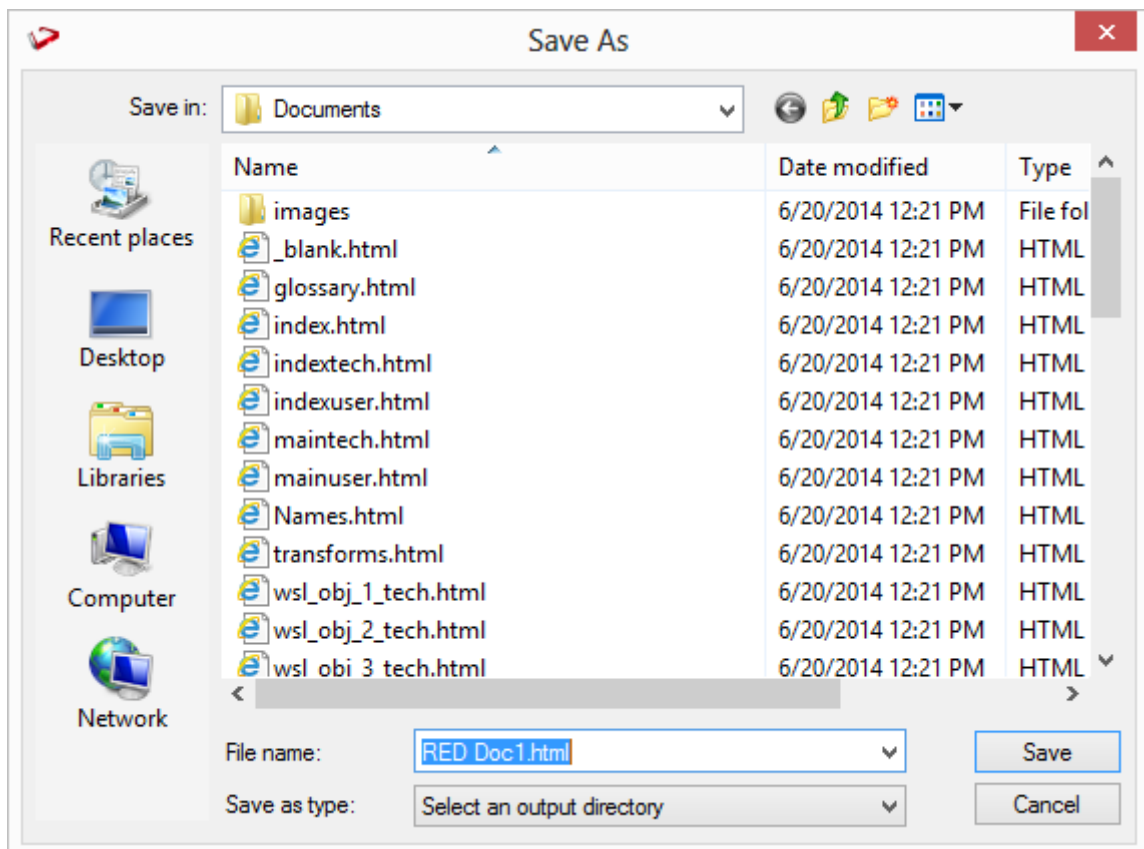
1.11 PRODUCING DOCUMENTATION

WhereScape RED also provides the ability to produce user and technical documentation. This is obviously of more value if the descriptive data has been entered against the columns and tables in the data warehouse, which we have not done during this tutorial.

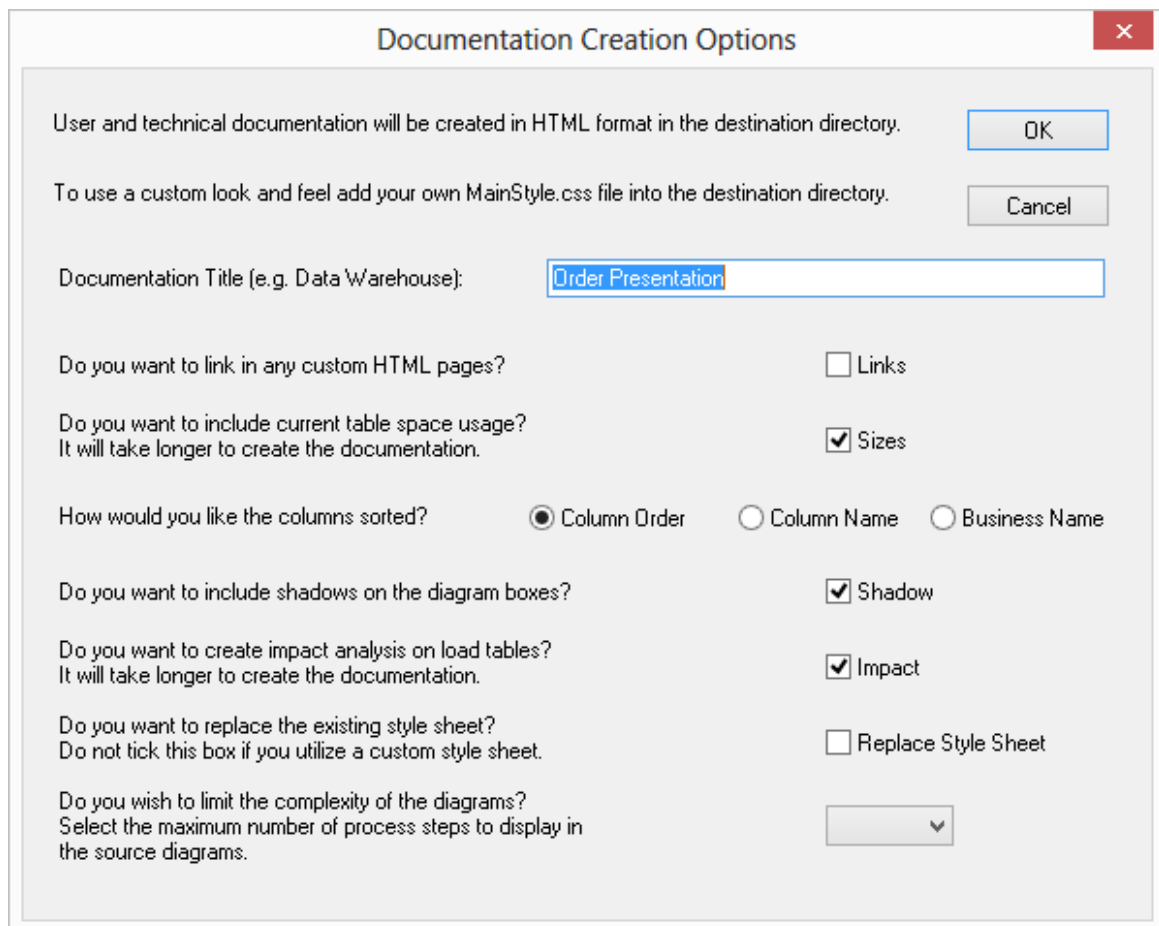
- 1 To view the documentation for the components of the data warehouse, select **Doc > Create Documentation**.



- 2 Select a file path (directory) under which to save the HTML files that will be produced. Click **Save**.



- The next screen allows for the inclusion of a banner and user defined links. Leave the **Links** option un-ticked and click **OK** to proceed.



Documentation Creation Options

User and technical documentation will be created in HTML format in the destination directory. OK

To use a custom look and feel add your own MainStyle.css file into the destination directory. Cancel

Documentation Title (e.g. Data Warehouse):

Do you want to link in any custom HTML pages? Links

Do you want to include current table space usage?
It will take longer to create the documentation. Sizes

How would you like the columns sorted? Column Order Column Name Business Name

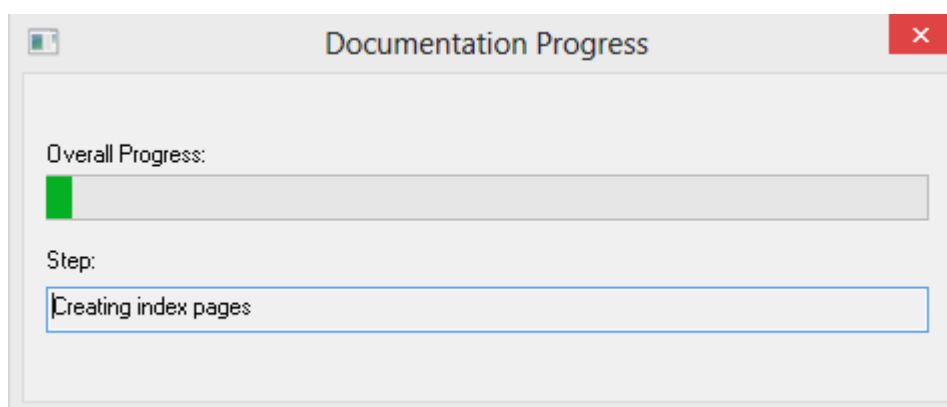
Do you want to include shadows on the diagram boxes? Shadow

Do you want to create impact analysis on load tables?
It will take longer to create the documentation. Impact

Do you want to replace the existing style sheet?
Do not tick this box if you utilize a custom style sheet. Replace Style Sheet

Do you wish to limit the complexity of the diagrams?
Select the maximum number of process steps to display in the source diagrams.

- The documentation will run.



Documentation Progress

Overall Progress:

Step:



TIP: To view the documentation, select **Doc > Display Documentation** on the main menu tabs

