The Dimensional Modeling Process

# Overview

This lab will introduce the Dimensional Modeling process. Upon completing this lab activity you learn:

* Techniques for profiling data using the SQL Query language.
* The process of high-level dimensional modeling, including:
  + Create a high-level dimensional model diagram (Kimball: fig. 7-3 pp. 304).
  + Create an attribute and metrics list (Kimball: fig. 7-2 pp. 294).
  + Keep track of issues
* The process of detailed dimensional modeling, including:
  + Create a detailed bus matrix (Kimball: fig. 7-7 pp. 315).
  + Documenting the detailed modeling process (Kimball: fig. 7-6 pp. 313).

## Lab Requirements

To complete this lab you will need the following:

* Access to the Northwind Database on Microsoft SQL Server 2012. This should be available through your iSchool vSphere login.
* You should connect to your SQL server database before starting this lab.
* The High-Level and Detailed dimensional modeling Excel Workbooks, available in the same place where you got this lab.
* Microsoft Excel 2007 or higher for editing the worksheets

## Grading

This lab may be handed in as part of a problem set.

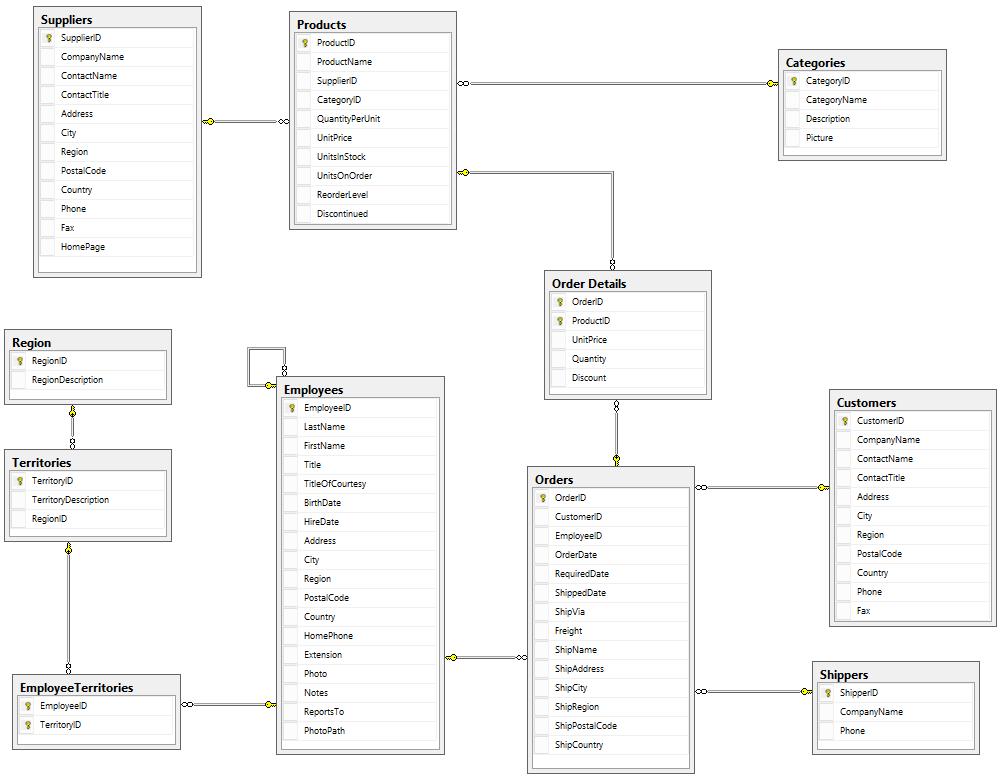
# http://www.geeksengine.com/img/article/northwind/northwind.gifPart 1: The Northwind Traders Case Study

Northwind traders is a fictitious importer and exporter of specialty foods from around the world. It was created by Microsoft as a sample operational database to use with their database products, such as Microsoft Access and SQL Server.

In our class, we’ll use this database as a case study for building a data warehouse. Over time you’ll need to get very intimate with the Northwind table design and source data as we complete our build out.

## The Northwind Data Model

Below is a screen shot of the internal model for the Northwind database. Use this diagram as a reference for understanding the structure of the Northwind data and building your dimensional model designs.



**NOTE:** You can view this database diagram on-line. It’s under the *Database Diagrams* section of your *Northwind* database and is accessible through *SQL Server Management Studio*.

## Data Warehouse Business Requirements

As part of the company’s Business Intelligence initiative, Northwind would like create data marts with the following goals:

* ***Sales reporting.*** Senior management would like to be able to track sales by customer, employee, product and supplier, with the goal of establishing which products are the top sellers which employees place the most orders, and who are the best suppliers.
* ***Order Fulfillment and Delivery****.* There is a need to analyze the order fulfillment process to see if the time between when the order is placed and when it is shipped can be improved
* ***Product Inventory Analysis.*** Management requires a means to track inventory, On Order, and Re-Order levels of products by supplier or category. Inventory levels should be snapshotted daily and recorded into the warehouse for analysis.
* ***Sales Coverage Analysis.*** An Analysis of the employees and the sales territories they cover.

As part of the business requirements, the following Enterprise Bus Matrix was created.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dimension ⭢**  **Bus. Process ⭣** | **Order Date** | **Shipped Date** | **Customers** | **Employees** | **Shippers** | **Products** | **Suppliers** | **Territory** |
| **Sales Reporting** | X | X | X | X |  | X | X |  |
| **Order Fulfillment** | X | X | X | X | X |  |  |  |
| **Inventory Analysis** | X |  |  |  |  | X | X |  |
| **Sales Coverage** |  |  |  | X |  |  |  | X |

# Part 2: Walk-Thru – Sales Reporting

In this part of the lab, we will work together to create a high-level and detailed design for the sales reporting business process. Along the way we’ll profile our dimensional data and get a feel for our facts using SQL Queries against the Northwind database.

## Part 2.a High Level Design

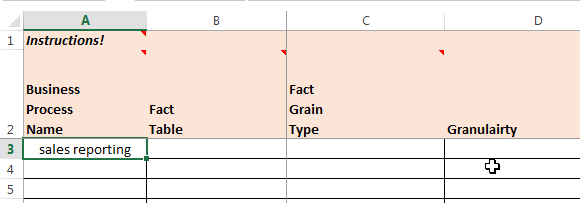
* Connect to your SQL Server using **SQL Server Management Studio** and open the **Northwind** database.
* Open the High-Level-Dimensional-Modeling Excel Workbook, to the **Detailed Bus Matrix** worksheet.

## Kimball’s 4 Step Modeling process

Kimball’s 4 step modeling process walks us through setting the fact grain and identifying the useable dimensions. The Detailed Bus Matrix worksheet is designed to walk you through this process.

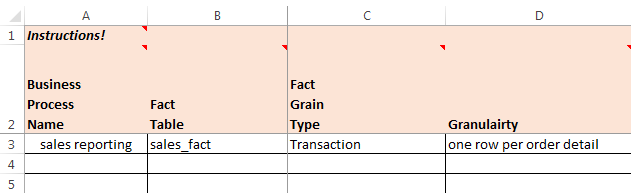
### Kimball’s Step 1: Business Process

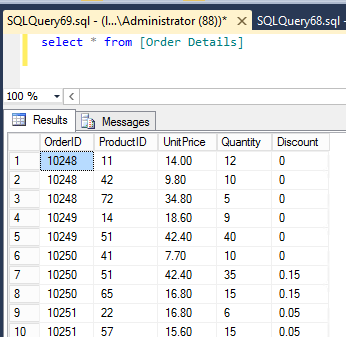
Our business process is **sales reporting**, so we place that in our worksheet:



### Kimball’s Step 2: Declare the grain

Our next step is to determine the level of grain for the fact table. What does it mean to be a single row in the sales reporting fact table? Well, if you read through the requirements (I know they’re vague) you can determine that each row represents the *sale of a product*, or a *line item on an order*. This is a ***transaction*** type fact, and so we update as follows:



At this point you might be wondering: what does order detail look like and how to we know it is what we need? This is where *data profiling* comes into play. Let’s take a look.

**DO THIS:** Switch to your SQL Server, and from **SQL Server Management Studio**, open a new Query window (**Ctrl + N)** and type select \* from [Order Details] Then press [F5] to execute. You should see results like this 🡪

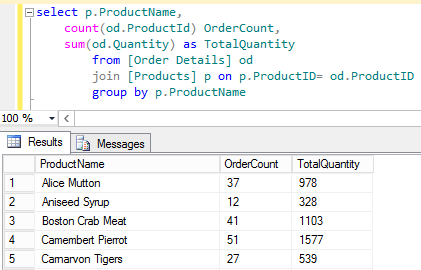
Each row in this query output represents the purchase of a product, which according to the requirements is what we need. Furthermore, many products can be part of one order (for example look at OrderID 10248) and therefore through order we can get back to other dimensions like customer and employee.

**NOTE:** In real life you won’t strike gold so easily. You’ll have to look at several tables before you can get a clear picture of your fact table grain.

### Kimball’s Step 3: Identify the dimensions

Next we identify the dimensions, but don’t take the enterprise bus matrix’s word for it. You should explore all possible dimensions and vet each one for yourself and with your stakeholders. The best way to do this when your source data is in a relational database is to *look at the table dependencies for the source of your fact data.*

For example if you review the *database diagram on page 2* of the lab you’ll see that the **Order Details** table connects directly to the **Products** table via a foreign key in a *many to one* relationship. Because it appears on multiple orders, **Product** fits the candidacy of a dimension. Once again we can verify this dimension works for us and “rolls up” a couple of our known facts by writing some SQL.

**DO THIS:** Switch to your SQL Server, and from **SQL Server Management Studio**, open a new Query window (**Ctrl + N)** and type   
select p.ProductName,

count(od.ProductId) OrderCount,

sum(od.Quantity) as TotalQuantity

from [Order Details] od

join [Products] p

on p.ProductID= od.ProductID

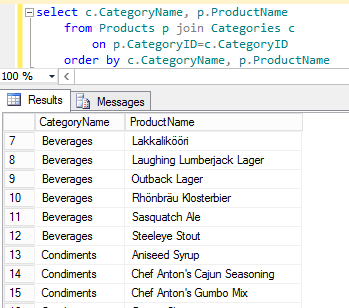
group by p.ProductName

Then press [F5] to execute. You should see results like this 🡪

What you’re seeing is a list of Products, along with a count of orders for which that product appears, and a total quantity sold for that product. (There’s a lot of details in this SQL statement, so feel free to ask your instructor for an explanation should you need it.)

**Important Tip:** You should always exercise caution when profiling live systems. Executing SQL queries against production data is usually not a wise decision as you may impact performance negatively. It is important to seek the advice of a Database Administrator prior to embarking your data profiling adventure!

#### Outrigger Dimensions and Hierarchies

You’ve probably noticed the **Products** table connects to the **Categories** and **Suppliers** tables in a *many to one* relationship (and thus these two tables are *dependent upon**the* ***Products*** *table*). This means there’s many products in a single category and several products for a single supplier. Situations like this help you discover *hierarchies* you can use in your dimension. Here’s the SQL we use to do this:

**DO THIS:** Switch to your SQL Server, and from **SQL Server Management Studio**, open a new Query window (**Ctrl + N)** and type   
select c.CategoryName, p.ProductName

from Products p join Categories c

on p.CategoryID=c.CategoryID

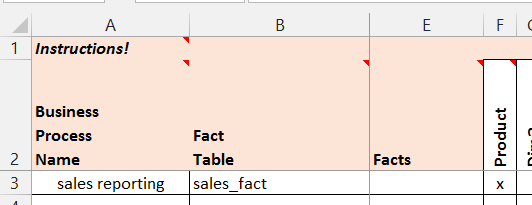
order by c.CategoryName, p.ProductName

Then press [F5] to execute. You should see results like this 🡪

Note our use of the order by clause in the SQL statement. This is important as it helps us visually identify the data hierarchy.

In this case, if we determine the hierarchy is useful we can consolidate the attributes we need from it into the product dimension. This makes more sense than including a separate dimension for Category.

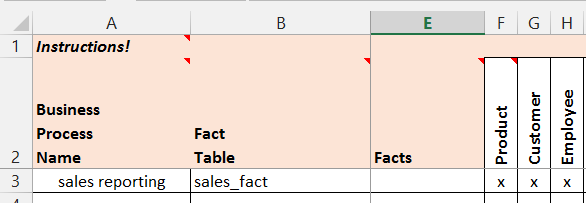
Once you’ve identified a useful dimension, it’s time to add it to our **Detailed Bus Matrix** like so:



#### Rinse, Lather and Repeat

Next we should go back and evaluate the other dependencies among our data once again by looking at the tables connected to our **Orders** and **Order Details** tables. Look for other dimensions which could be useful in our model, and when in doubt you can always check its roll-up capability with some SQL.

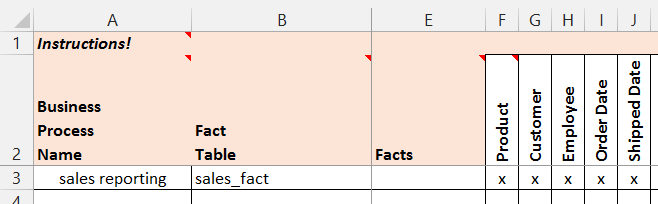
Here’s a screenshot of the dimensions I’ve discovered so far:



**Important Tip:** It’s important to recognize that dimensional modeling is not a formal that can be automated. There’s a lot of art that goes with the science. (To quote the pirate’s code these are more like guidelines than hard and fast rules ☺).

#### What about the time dimension?

To identify date and time dimensions, look for dates stored in the tables associated with your fact grain. In our case if you run an SQL query on the **Orders** table you’ll see **Order Date** and **Shipped Date**. We’ll add both to our model:

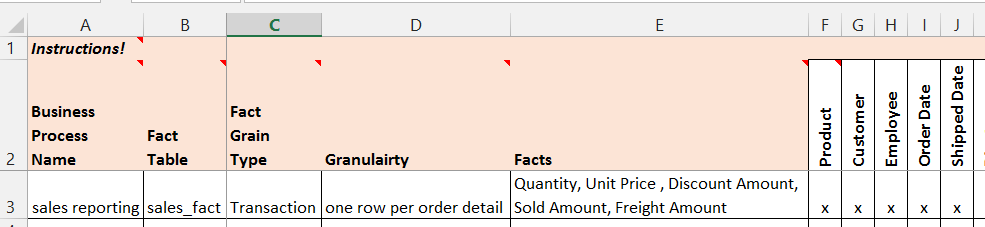


### Kimball’s Step 4: Identity the facts

After you wrap up your dimensions, it’s time to identify the facts. From identifying the fact grain of the model you probably already have a few facts in mind, but now’s the time to really nail down the facts you need in your model. Like everything else in this step a lot will depend on your requirements.

One important this to recognize is not all facts appear among your source data. Some of the facts you’ll need are *derived*by doing a “little math” on some of the source data values. We include the facts we want in the **Detailed Bus Matrix** but explain how they are derived in the **Attributes and Metrics** worksheet. For now, we’ll add the following facts to our **Detailed Bus Matrix** and complete it.

* Quantity – (of product sold)
* Unit Price
* Discount amount (unit price \* discount)  
  Sold Amount ( Quanity \* Unit Price less discount Amount for each item on order)
* Freight Amount (split evenly among items on the order)



## Attributes & Metrics

Now that you’ve completed the Detailed bus Matrix for your business process, it’s time to move down a level of detail in the process. In this next step we will circle back through our dimensions and facts and put together a quick list of **Attributes and Metrics** that we require for our dimensional model this list allows you to get more specific about the needs of your dimensional model.

Completing the **Attributes and Metrics** worksheet is self-explanatory and therefore I will leave it as an exercise for you. As you complete this part, keep the following in mind:

* Start with the dimensions you’ve identified in your **Detailed Bus Matrix**.
* You can profile for useful dimensional attributes with a SQL query like this:   
  select \* from [table\_name]
* Don’t forget to explore any hierarchies among your dimensions, as discussed in the previous section.
* Time dimensions are fairly standard. You only need to be detailed about any unique definitions in your time dimensions.
* If your fact is semi-additive, make note in the description.
* If your fact is derived, be sure to explain how it is derived in the description.

When you’re done. Save your worksheet before moving on to the next step.

## Part 2.b Detailed Design

Congratulations! If you made it this far, you’ve completed your first High-Level dimensional design. In this next part we will iterate over our dimensional model ***one more time*** to create our **Detailed Dimensional Design**. The goals of this process are to:

1. Create a formal table design, including tables, keys, data types, and indexes so we can create tables and indexes required for our star schemas (ROLAP).
2. Identify data sources of our dimensional model so that we can architect and implement the ETL process in a future phase.

### Introducing the Detailed Dimensional Modeling Workbook

The **Detail Dimensional Modeling Workbook** we will use comes from the Kimball Consulting Website <http://www.kimballgroup.com>. The Workbook contains Excel macros which automate some diagram creation a basic SQL generation. ***We will not use these features, however you are welcome to try them out***.

**DO THIS:** You should start by **opening the** **Excel Workbook** and reading the section titled **How to use this tool** under the **Home** worksheet, and then read the **ReadMe** tab. This will give you an overview of how to use this workbook.

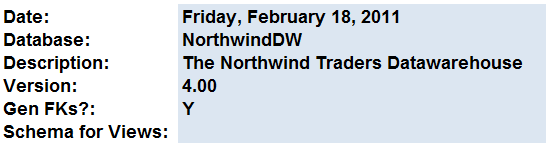
#### Getting Started

First let’s setup the workbook.

**DO THIS:** Click on the **Home** worksheet, and complete the fields as follows:

* Database: **NorthwindDW**
* Description: **The Northwind Traders Data Warehouse**
* Gen FK’s?: **Y**
* Schema For Views:  **(leave blank)**

Here’s a screenshot of the completed worksheet.

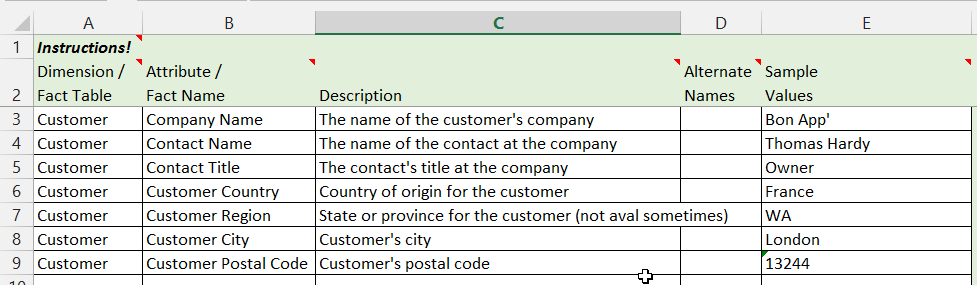


We’re now ready to start our detailed dimensional design.

### Completing the detailed design for the Customer dimension

In this next step, we will complete the detailed dimensional design for the **Customer** dimension of our **Sales** **Reporting** data mart from part 2.a. The other dimensions will be left as an exercise for you. If you need assistance, refer to the sample detailed design you instructor has provided for you.

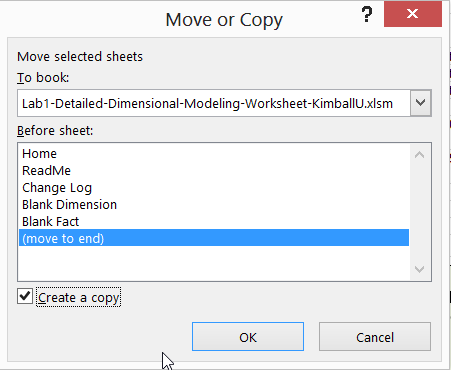
To complete the design we will need to refer to the **Attributes & Measures** from the high level design. A screen shot has been included for reference.



The process you’ll follow to design a dimension or fact table is outlined in **5 Steps:**

1. Create a new dimension (or fact) worksheet in the workbook.
2. Complete the table definition part of the worksheet.
3. Complete the basic column information.
4. Complete the target table information.
5. Complete the source data information.

#### Step 1: Create a new dimension worksheet

Let’s document the details of this dimension.

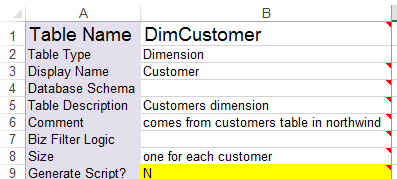
**DO THIS:** Make a copy of the **Blank Dimension** worksheet. ***Right-click*** on it and select ***Move or Copy*** from the menu. When the dialog appears, click ***(move to end)*** and check the ***Create a copy*** checkbox before clicking ***OK***.

You will now have a **Blank Dimension (2)** worksheet. ***Right-click*** on it and select ***rename*** from the menu, then type in **Customer**.

You should now have your first dimension: 

#### Step 2: Complete the table definition

Our next step is to complete the **table definition** for the customers dimension table. Please complete it as follows:



The name of table

Must match the worksheet tab name for the diagram to work

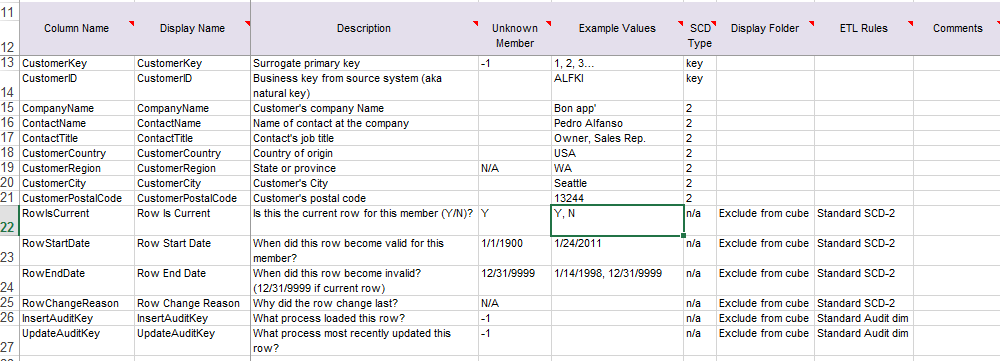
These fields should be completed as necessary

#### Step 3: Complete the basic column information

Next we complete the basic column definitions using the data from our **Attributes & Measures** from the high level design. Here’s an explanation of the first 9 columns:

* **Column Name** 🡪 Physical name of column in the table
* **Display Name** 🡪 Logical name of column in the table (should match the physical name)
* **Description** 🡪 Explanation of the column, for documentation purposes
* **Unknown Member** 🡪 What should be used for an unknown value (in place of NULL)
* **Example Values** 🡪 What do sample values look like?
* **SCD Type** 🡪 Slowly changing dimension type: key (does not change), 1,2,3, or n/a
* **Display Folder** 🡪 Provides grouping for similar attributes / facts in a cube design.
* **ETL Rules** 🡪 Any special ETL rules, if known at this time.

Fill out your column information to match mine:



**NOTE:** Included in this detailed design are techniques for dealing with type-2 SCD’s and an audit dimension (everything from row 22 and higher in the screenshot). Both of these techniques are covered in the ETL chapters of our course. For now, we can leave these in our design. We’ll revisit them later.

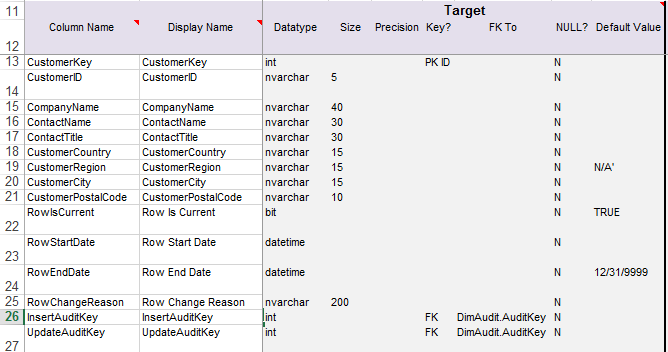
#### Step 4: Target (dimensional table) table definition

In this next step, you’ll define the table definition for our dimension table in our ROLAP star schema. You work here follows along with normal relational table design definitions.

The columns you’ll need to complete in this step for each attribute are:

* **Datatype, Size, Precision** – the SQL Server datatype (including size and precision, where appropriate) of the attribute. A good rule of thumb is to check the source data type for reference. It should be noted that data types vary from DBMS to DBMS. SQL server datatype reference can be found at <http://msdn.microsoft.com/en-us/library/ms187752.aspx>.
* **Key?** – Should be blank if not a key or labeled PK = primary key, PK ID = primary key (with surrogate), or FK = foreign key.
* **FK To** – When you label an attribute as FK, you need to include a dimension table and its primary key as the referencing column.
* **NULL**? – Whether or not the attribute permits null values. This should only be permitted in very rare circumstances. The better design decision is to provide a default value in place of NULL.
* **Default Value** – A value which should be stored in the event there is no value.

Here’s a screenshot of my completed target definition for customer:

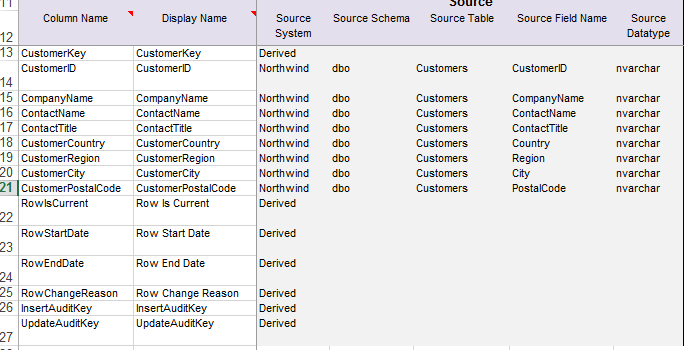


#### Step 5: Source definition

In this final step you’ll complete the source definition, which will assist us as we complete the ETL implementation in a subsequent step. Here’s an explanation of the columns you’ll need to complete in this step:

* **Source System** – List the source system for the attribute. ***Derived*** implies the attribute is calculated.
* **Source Schema** – If the attribute comes from a specific schema, list it here.
* **Source Table** – State the table the attribute comes from on the source system.
* **Source Field Name** – The column or columns which supply the attribute. If the column is a calculation, specify that here (ex. OrderQty\*Price).

Here’s a screenshot of my completed target definition for customer:



### Next Steps

Finish your detailed design by repeating steps 1-5 for your other dimensions and fact table for the sales reporting process. You’ll probably have to do a bit of profiling to complete the process. Ask questions if you have them, or if working outside of class time, log your issues to the issues worksheet.

# Part 3: On Your Own – Order Fulfillment

After you have finished up the sales reporting design, its time to move on to order fulfillment.

In this part, you will repeat the process outlined in part 2 of the lab for the order fulfillment business process. Here’s a set of instructions and guidelines for you to follow:

1. In your high-level modeling Excel workbook:
   1. Revise your **Detailed Bus Matrix** worksheetto include this new business processes. When evaluating your dimensions, don’t just take the Enterprise Bus Matrix at its word. Remember to observe the data relationships of the Northwind tables and profile in SQL for other useful dimensions and attributes.
   2. Based on your findings, update the **Attributes & Metrics** worksheet for your new business processes. Remember in the *Enterprise Data Warehouse* we strive for **conformed dimensions** and **consolidated fact tables** so keep a keen eye out for ways you can reuse what you’ve modeled already.
   3. If you encounter issues, unknowns, or things you just cannot figure out, record them to your issues list.
2. Once you’ve completed the first part move on to the detailed modeling Excel workbook:
   1. Update any changes to your existing fact and dimension tables. Don’t add another customer dimension if there’s one already there. Remember we strive for **conformed dimensions** and **consolidated fact tables.**
   2. Add new fact and dimensions table designs that you have, basing them on your **Attributes & Metrics** worksheet from the first step.
   3. In both cases make sure to be complete in your documentation process, including the target table design for your star schema and information regarding your data source so that you have what is required for the ETL design.
   4. Any issues you encounter, such as not knowing how to source your data, should be placed in the issues list (in the other workbook).