Assignment 03:
High-Level Dimensional Modeling

Part 1: Overview

This assignment will introduce you to the high-level dimensional-modeling process. The goal of this process is to turn functional business requirements into dimensional data warehouse (DDS) specifications based on the Kimball technical architecture. Upon completing this lab activity you learn:

- Additional ways of profiling data using the SQL Query language as to identify master data and business processes
- The process of high-level dimensional modeling, including:
    - Create a high-level dimensional model diagram (Kimball: Figure 7-3, p. 304)
    - Create an attribute and metrics list (Kimball: Figure 7-2, p. 294)
    - Keeping track of issues

Goals

Specifically the goals of this assignment are to:

- Understand the goals of the high-level dimensional modeling process and practice its steps
- Master the act of profiling data and transforming functional requirements into a technical specifications for a Kimball (DDS) data warehouse architecture
- Understand the value of the high-level modeling worksheet as a technical documentation tool, which can be later used to determine how to properly build tables in our DDS

Effort

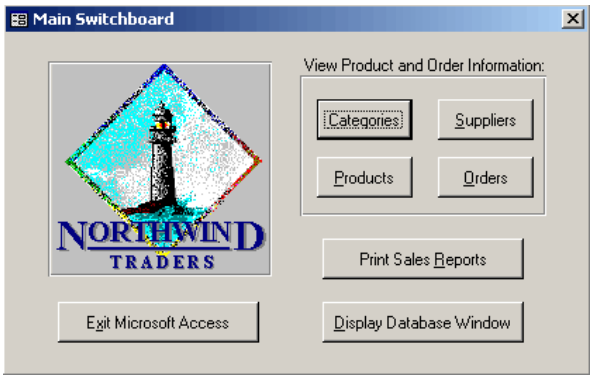
This assignment can be done individually or with a partner. If you work with a partner, do not simply divide up the work. Collaborate with each other throughout the exercise as if you were working on the same data warehousing team.

Technical Requirements

To complete this assignment you will need the following:

- Access to the course **ist-cs-dw1.ad.syr.edu** SQL Server, specifically the Northwind Traders database. You should connect to this server before starting the assignment.
- The High-Level Dimensional-Modeling Excel workbook, available in the same place where you got this assignment.
- Microsoft Excel 2007 or later for editing the workbook.

Recall: The Northwind Traders Case Study

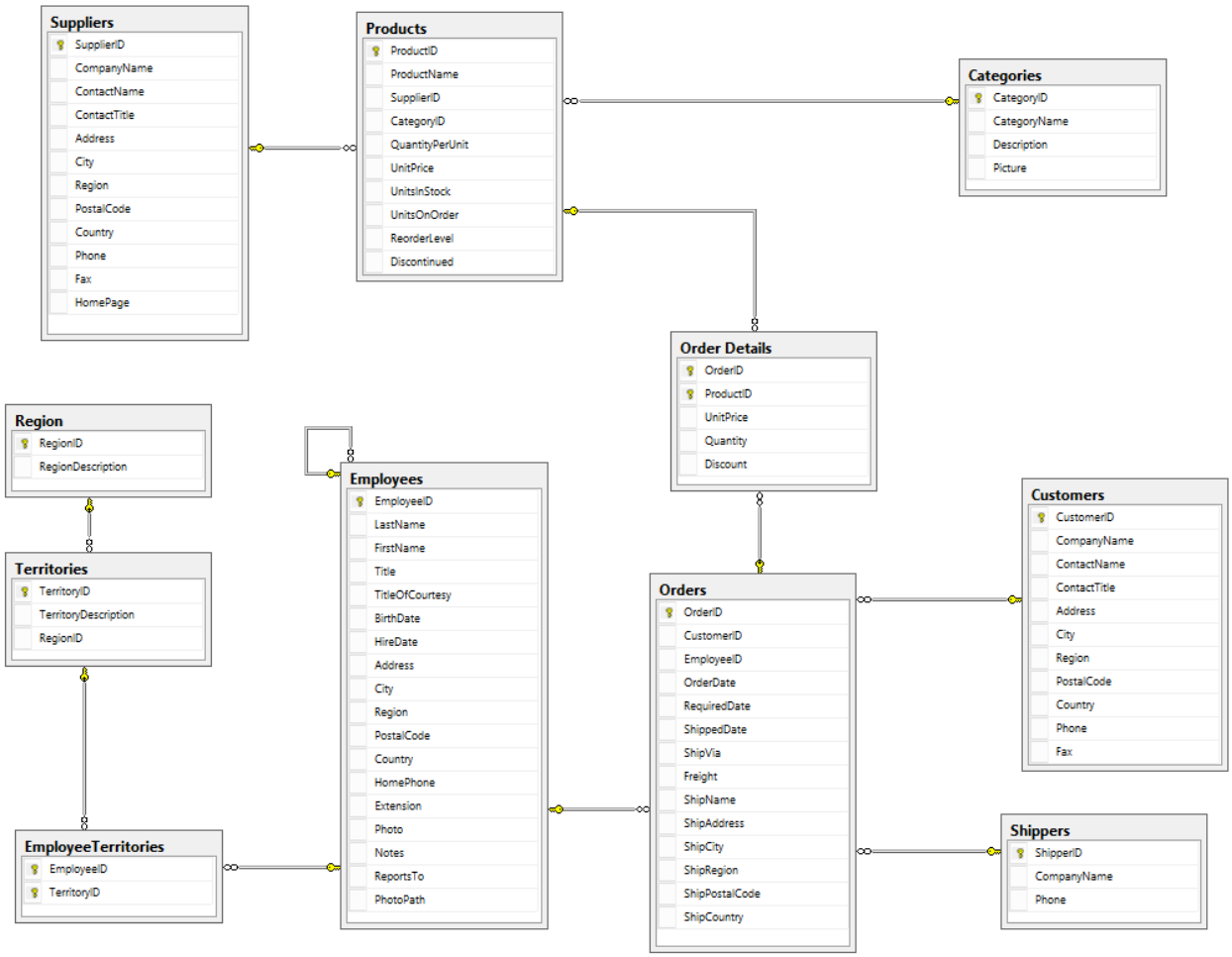


Northwind Traders is a fictitious importer and exporter of specialty foods from around the world. It was created by Microsoft as a sample operational database to use with its database products, such as Microsoft Access and SQL Server.

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.



Data Warehouse Functional Business Requirements

As part of the company's business intelligence initiative, the data warehousing team has established the following four functional business requirements:

1. ***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.
2. ***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
3. ***Product inventory analysis.*** Management requires a means to track inventory, on order, and reorder levels of products by supplier or category. Inventory levels should be snapshotted daily and recorded into the warehouse for analysis.
4. ***Sales coverage analysis.*** An analysis of the employees and the sales territories they cover.

Part 2: Walk-Through

In this part of the assignment, we will work together to create a high-level design for the first functional business requirement: sales reporting. Along the way we'll profile our dimensional data and get a feel for our facts using SQL queries against the Northwind database.

Getting Started

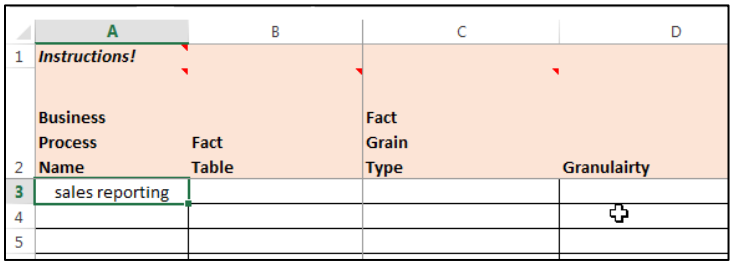
- Connect to your SQL Server using **SQL Server Management Studio**, and open a query window for the **Northwind** database.
- Open the High-Level-Dimensional-Modeling Excel workbook, to the **Bus Matrix** page.

Kimball's Four-Step Modeling Process

Kimball's four-step modeling process walks us through setting the fact grain and identifying the useable dimensions. The **Bus Matrix** page 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:



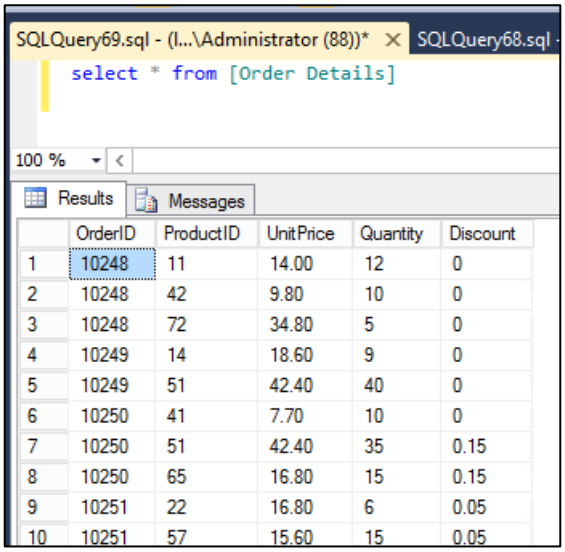
Kimball's Step 2: Declare the Grain

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:

	A	B	C	D
1	***Instructions!***			
2	**Business Process Name**	**Fact Table**	**Fact Grain Type**	**Granulairty**
3	sales reporting	sales_fact	Transaction	one row per order detail
4				
5				

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 →



	OrderID	ProductID	UnitPrice	Quantity	Discount
1	10248	11	14.00	12	0
2	10248	42	9.80	10	0
3	10248	72	34.80	5	0
4	10249	14	18.60	9	0
5	10249	51	42.40	40	0
6	10250	41	7.70	10	0
7	10250	51	42.40	35	0.15
8	10250	65	16.80	15	0.15
9	10251	22	16.80	6	0.05
10	10251	57	15.60	15	0.05

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 the 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 master data associated with the business process. In the Kimball technical architecture, these will become the dimensions in our dimensional data store. How do find the dimensions? The best way to do this when your source data are 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 →

```
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
```

100 %

Results | Messages

	ProductName	OrderCount	TotalQuantity
1	Alice Mutton	37	978
2	Aniseed Syrup	12	328
3	Boston Crab Meat	41	1103
4	Camembert Pierrot	51	1577
5	Carnarvon Tigers	27	539

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 are 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 →

```
select c.CategoryName, p.ProductName
    from Products p join Categories c
        on p.CategoryID=c.CategoryID
    order by c.CategoryName, p.ProductName
```

100 %

Results | Messages

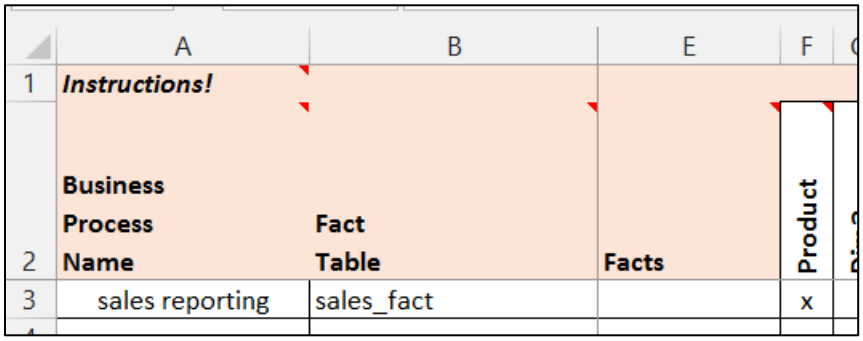
	CategoryName	ProductName
7	Beverages	Lakkalikööri
8	Beverages	Laughing Lumberjack Lager
9	Beverages	Outback Lager
10	Beverages	Rhönbräu Klosterbier
11	Beverages	Sasquatch Ale
12	Beverages	Steeleye Stout
13	Condiments	Aniseed Syrup
14	Condiments	Chef Anton's Cajun Seasoning
15	Condiments	Chef Anton's Gumbo Mix

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.

There are cases where some other business process might need Suppliers or Categories, and therefore it would make sense to combine them into a single dimension. This is the fundamental idea behind *snowflaking*.

Once you've identified a useful dimension, it's time to add it to our **Bus Matrix** like so. In this example we've added the **Product** dimension.



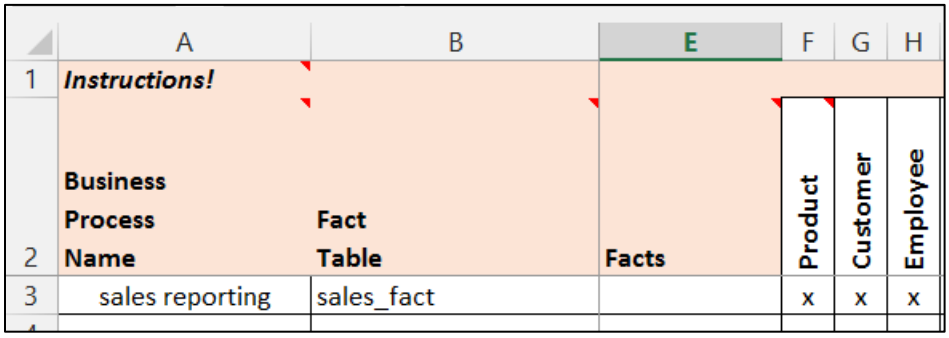
	A	B	E	F
1	***Instructions!***			
2	**Business Process Name**	**Fact Table**	**Facts**	**Product**
3	sales reporting	sales_fact		x

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 via foreign key. Look for other dimensions that could be useful in our model, and when in doubt you can always check its roll-up capability with some SQL, like we did with **Product**.

Important Tip: There should always be a *many-to-one* relationship between the business process table and the master data that make up your dimension. One row in the dimension should appear many times in the business process. For example, one product appears many times on different orders.

Fast-forward through some more data profiling, and here's a screenshot of the dimensions I've discovered so far:



	A	B	E	F	G	H
1	***Instructions!***					
2	**Business Process Name**	**Fact Table**	**Facts**	**Product**	**Customer**	**Employee**
3	sales reporting	sales_fact		x	x	x

Important Tip: The **x** at the intersection of dimension and business process indicates there will be a foreign key in our DDS connecting the business process to the dimension table. Our goal is to reuse dimensions like **Product, Customer,** and so on across other business processes. This is called *conforming dimensions.*

The Date and Time Dimensions

One key dimension essential to the DDS architecture is the date (and sometimes time) dimensions. To identify date and time dimensions, look for dates stored in the tables that sources the business process.

In our case if you run an SQL Query on the **Orders** table, you'll see **Order Date** and **Shipped Date**. So we'll add both to our model:

	A	B	E	F	G	H	I	J
1	***Instructions!***							
2	**Business Process Name**	**Fact Table**	**Facts**	**Product**	**Customer**	**Employee**	**Order Date**	**Shipped Date**
3	sales reporting	sales_fact		x	x	x	x	x
4								

Important Tip: These are not two different dimensions. They are the same dimension, but we need two foreign keys back to the same table. This is referred to as a *role-play dimension*.

Kimball's Step 4: Identity the Facts

After you establish your business process and wrap up your dimensions, it's time to identify the facts. Facts are the quantifiable values that we measure across attributes in the dimension. I know. It's a mouthful, so here's a couple of examples.

- How many of a specific product category were sold? **Category** is the attribute of the **Product** dimension, and **how many** is the measurement and, therefore, the fact.
- Which customers have ordered the most? **Customer** is the dimension, and **Sold Amount** is the measurement (fact).

From merely identifying the fact grain of the model you probably already have a few facts in mind (they can be found in the business process table), 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 thing to recognize is not all facts appear among your source data. Some of the facts you'll need are *derived facts***.** We do a little math on some of the source data values. We include the facts we want in the **Bus Matrix** but explain how they are derived in the **Attributes and Metrics** worksheet. For now, we'll add the following facts to our **Bus Matrix** and complete it.

- Quantity (of product sold)
- Unit price
- Discount amount (unit price * discount)
- Sold amount (Quantity * Unit Price less discount Amount for each item on order)
- Freight amount (split evenly among items on the order: Freight/sum of quantity)

Important Tip: That last one is a bit tricky. The **Freight** value is found in the **Orders** table and is not broken out per item on the order. We have made a *data governance* decision to evenly divide the freight by the number of items in the order. In the real world we don't make this decision—the business users decide how it should be handled.

	A	B	C	D	E	F	G	H	I	J
1	***Instructions!***									
2	**Business Process Name**	**Fact Table**	**Fact Grain Type**	**Granulairty**	**Facts**	**Product**	**Customer**	**Employee**	**Order Date**	**Shipped Date**
3	sales reporting	sales_fact	Transaction	one row per order detail	Quantity, Unit Price , Discount Amount, Sold Amount, Freight Amount	x	x	x	x	x

Attributes and Metrics

Now that you've completed the **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 business process (fact tables) 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.

The idea behind the attributes and metrics is to define your facts and outline the important attributes in your dimensions.

Completing the **Attributes and Metrics** page in the workbook 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 **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 semiadditive, make note in the description.
- If your fact is derived, be sure to explain how it is derived in the description.
- Don't assume anything. Use this page in the workbook to make your intentions clear.

Part 3: On Your Own

After you have finished up the sales reporting, it's time to move on to the other three business processes:

1. ~~***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.~~
2. ***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
3. ***Product inventory analysis.*** Management requires a means to track inventory, on order, and reorder levels of products by supplier or category. Inventory levels should be snapshotted daily and recorded into the warehouse for analysis.
4. ***Sales coverage analysis.*** An analysis of the employees and the sales territories they cover.

In this part, you will repeat the process outlined in Part 2 of the assignment for the remaining three business processes.

When you are finished you should have the following in your **High- Level Dimensional-Modeling workbook:**

1. A completed **Bus Matrix** with all four business processes in it. No dimensions should repeat. To reuse a dimension for another business process, include an X at its intersection.
2. A completed **Attributes and Metrics** list. Specifically you should define facts and derived facts and important dimension attributes.
3. Along the way if you encounter issues or unknowns, record them under the **Issues List** tab so you remember to address them at some point in the process.

Tip: Keep in mind you can model only the data you have. If it's not in your external world source data (in this case, it's Northwind Traders), then you cannot include it in your data warehouse!

Turning It In

Please turn in your completed **High-Level Dimensional-Modeling worksheet.** Make sure your name, NetID, and date appear somewhere at the top of the **Bus Matrix** page.

If you worked with a partner, please indicate that in your assignment by including your partner's name and NetID. You should both submit the assignment individually.