

Northwind Database Portfolio

Taylor Jackson

Colorado State University - Global

MIS 541: Data Warehousing in Enterprise Environments

Dr. Richard Morse

March 7th, 2021

Northwind Database Portfolio

The Northwind database captures transactions within the company, along with data regarding its inventory, employees, and so on. In other words, the database represents the business process of a sale of their products and creating the inventory of those products. This process will be used to develop the dimensional model. Developing the dimensions model will require identifying the grain, dimension and fact tables, and the attributes of both. In addition to these steps, a set of business questions will be defined.

Business Relations

As mentioned, this database represents a series of transactions which will later be defined as a fact table. Whereas the dimensions cover the components that make this transaction possible (for example, products and employees). The business process as a complete cycle would be defined as the following: The organization acquires products from a supplier and sells this product to a customer.

With this understanding there are several business questions that should be considered. These questions are vital to strong business decisions and should be answered by queries to the database (the creation of which is outlined below). Important business questions include an understanding of how sales are by month, year, product, and even by employee.

These queries provide insights to the Northwind company and can be used in their BI (business intelligence) processes. It is the key to making decisions moving forward. For example, if a product is only accountable for .05% of sales last year, it may be worth discontinuing. Another example is employee performance evaluations, how are sales teams' members doing? Who deserves a raise this year?

Dimensional Model

This business process is used to build a dimensional model. As the center of the business process, the sales transaction is represented as the fact table. The dimensions represented in this dimensional model are customers, employees, products (and product categories), as well as shippers.

The fact table Orders, represents a single order per row (this is the defined grain of the fact table). This is easily implemented due to this table's attributes and is a vital step in the dimensional model creation; "In debugging thousands of dimensional designs over the years, the most frequent error is not declaring the grain of the fact table at the beginning of the design process" (Kimball and Ross, 2013, p. 70).

The fact table attributes include the primary key the order_id and several foreign keys: customer_id, employee_id, shipper_id. To allow further analysis of the order fact because it is limited by a single order by row grain. To this point, the granularity of this fact table is quite low. Rather than granularity being the entire order description, or listing all products within the order, the fact table is expanded into dimensions that can be analysis to come to higher level determinations.

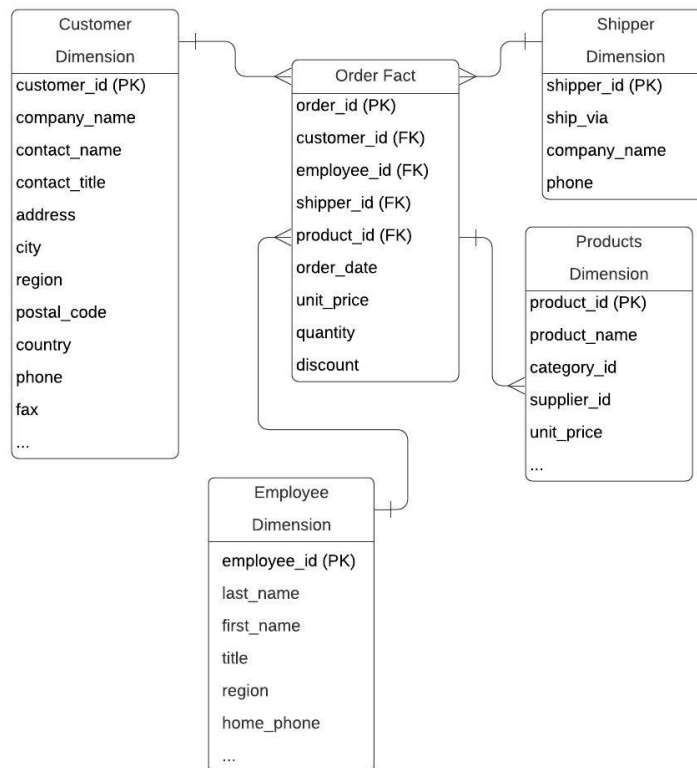
The other dimensions vary in attributes depending on their value and purpose. For example, some have several attributes and no variables while others have many variables. To identify the attributes and variables of all dimensions and fact tables in the Northwind, see Table 1.

Table 1
Northwind Database Characteristics

Dimension/Fact Table	Attributes	Variables
Customers (dimensional table)	customer_id	company_name, contact_name, contact_title, address, city, region, postal_code, country, phone, fax
Employees (dimensional table)	employee_id	last_name, first_name, title, region, home_phone
Orders (fact table)	order_id, customer_id, employee_id, shipper_id, product_id	order_date, unit_price, quantity, discount
Products (dimensional table)	product_id	product_name, category_id, supplier_id, unit_price
Shippers (dimensional table)	shipper_id	ship_via, company_name, phone

To further understand the Northwind Database dimensions model design, see Figure 1 below. This figure provides the star schema to be utilized based on the details outlined above. Furthermore, it provides the cardinalities of each relationship.

Figure 1
Dimensional Model



SQL Statements

The SQL commands used to generate and load data into the new Northwind Database are outlined in Table 2. Furthermore, the commands used to review and understand the newly added data in the database are outlined in Table 2 as well.

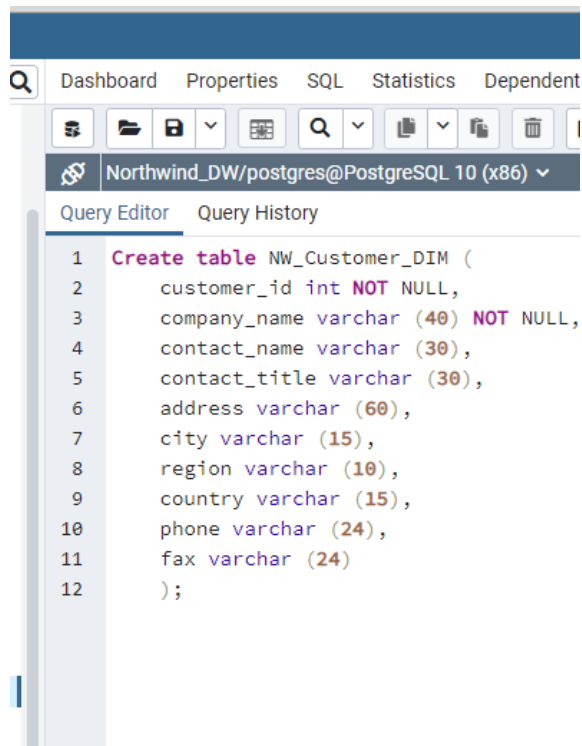
Table 2
SQL Commands

SQL Commands	Purpose
CREATE TABLE [table] ();	Create the dimension and fact tables
ALTER TABLE ONLY [table] ADD CONSTRAINT [] PRIMARY KEY ([]);	Add constraints to the created tables (primary and foreign)
ALTER TABLE ONLY [] ADD CONSTRAINT [] FOREIGN KEY ([]) REFERENCES [],	
CREATE EXTENSION []; CREATE SERVER [] FOREIGN DATA WRAPPER [] OPTIONS (host '[]', dbname'[]'); CREATE USER MAPPING FOR CURRENT_USER SERVER [] OPTIONS (user '[]', password '[]');	Load data from another database into the created tables. Requires foreign data wrapper to add foreign data tables.
CREATE SCHEMA [];	
IMPORT FOREIGN SCHEMA public FROM SERVER [] INTO [];	
SELECT [] INTO [] FROM [];	Load the connected foreign tables data into the created tables in the new database.

Dimension Model Tables

Below, the created tables and SQL queries performed are demonstrated. The first figures (2-5) show the tables being created. These tables are based on the dimensional model outlined in Figure 1. Figures 6 and 7 show constraints being added using the ALTER TABLE SQL command. Note that the added tables are visible in Figures 6 and 7 to show they were successfully created.

*Figure 2
Customer Table*

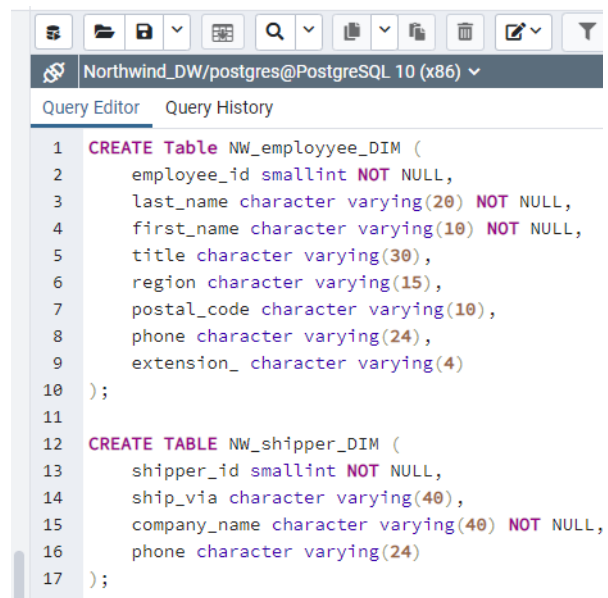


```

1  Create table NW_Customer_DIM (
2      customer_id int NOT NULL,
3      company_name varchar (40) NOT NULL,
4      contact_name varchar (30),
5      contact_title varchar (30),
6      address varchar (60),
7      city varchar (15),
8      region varchar (10),
9      country varchar (15),
10     phone varchar (24),
11     fax varchar (24)
12 );

```

*Figure 3
Employee and Shipper Table*



```

1  CREATE Table NW_employee_DIM (
2      employee_id smallint NOT NULL,
3      last_name character varying(20) NOT NULL,
4      first_name character varying(10) NOT NULL,
5      title character varying(30),
6      region character varying(15),
7      postal_code character varying(10),
8      phone character varying(24),
9      extension_ character varying(4)
10 );
11
12 CREATE TABLE NW_shipper_DIM (
13     shipper_id smallint NOT NULL,
14     ship_via character varying(40),
15     company_name character varying(40) NOT NULL,
16     phone character varying(24)
17 );

```

Figure 4
Products Table

```

17 );
18
19 CREATE TABLE NW_products_DIM (
20     product_id smallint NOT NULL,
21     product_name character varying(40) NOT NULL,
22     category_id smallint,
23     supplier_id smallint,
24     unit_price real
25 );
26

```

Figure 5
Order Table

Northwind_DW/postgres@PostgreSQL 10 (x86) ▾

Query Editor Query History

```

1 CREATE TABLE NW_orders_Fact (
2     order_id smallint NOT NULL,
3     customer_id character varying(40) NOT NULL,
4     employee_id smallint NOT NULL,
5     shipper_id smallint NOT NULL,
6     product_id smallint NOT NULL,
7     order_date date,
8     unit_price smallint,
9     quantity smallint,
10    discount smallint
11 );

```

Figure 6
Adding Primary Keys

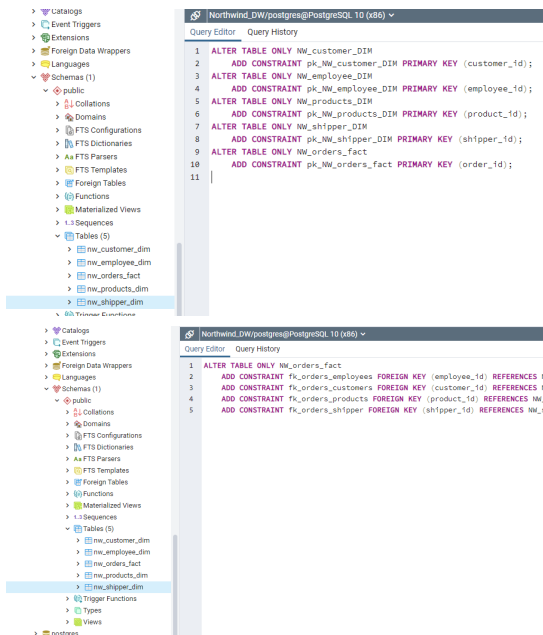
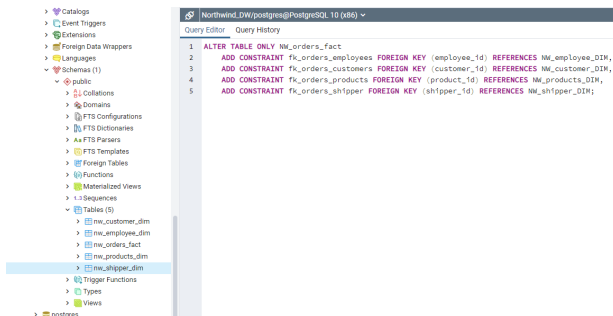


Figure 7
Adding Foreign Keys



The following figures used the SQL statements listed in Table 2 related to viewing and understanding the tables and loaded data. See Figures 8-11 for the populated dimension tables and the first tens rows of each table.

Figure 8
First 10 rows Employees

Northwind_DW/postgres@PostgreSQL 10 (x86) ▾

Query Editor Query History

```
1 SELECT * FROM nw_employee_dim LIMIT 10;
```

Data Output Explain Messages Notifications

	employee_id	last_name	first_name	title	region
1	1	Davolio	Nancy	Sales Representative	WA
2	2	Fuller	Andrew	Vice President, Sales	WA
3	3	Leverling	Janet	Sales Representative	WA
4	4	Peacock	Margaret	Sales Representative	WA
5	5	Buchanan	Steven	Sales Manager	[null]
6	6	Suyama	Michael	Sales Representative	[null]
7	7	King	Robert	Sales Representative	[null]
8	8	Callahan	Laura	Inside Sales Coordinator	WA
9	9	Dodsworth	Anne	Sales Representative	[null]

Figure 9
First 10 rows Products

Northwind_DW/postgres@PostgreSQL 10 (x86) ▾

Query Editor Query History

```
1 SELECT * FROM nw_products_dim LIMIT 10;
```

Data Output Explain Messages Notifications

	product_id	product_name	supplier_id	unit_price
1	1	Chai	8	18
2	2	Chang	1	19
3	3	Aniseed Syrup	1	10
4	4	Chef Anton's Cajun Seasoning	2	22
5	5	Chef Anton's Gumbo Mix	2	21.35
6	6	Grandma's Boysenberry Spr...	3	25
7	7	Uncle Bob's Organic Dried P...	3	30
8	8	Northwoods Cranberry Sauce	3	40
9	9	Mishi Kobe Niku	4	97
10	10	Ikura	4	31

Figure 10
First 10 rows Customers

Northwind_DW/postgres@PostgreSQL 10 (x86) ▾

Query Editor Query History

```
1 SELECT * FROM nw_customers_dim LIMIT 10;
```

Data Output Explain Messages Notifications

	customer_id	contact_name	contact_title	address	city
1	ALFKI	Maria Anders	Sales Representative	Obere Str. 57	Berlin
2	ANATR	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.
3	ANTON	Antonio Moreno	Owner	Mataderos 2312	México D.F.
4	AROUT	Thomas Hardy	Sales Representative	120 Hanover Sq.	London
5	BERGS	Christina Berglund	Order Administrator	Berguvägen 8	Luleå
6	BLAUS	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim
7	BLONP	Frédérique Citeaux	Marketing Manager	24, place Kléber	Strasbourg
8	BOLID	Martin Sommer	Owner	C/ Araquil, 67	Madrid
9	BONAP	Laurence Leblan	Owner	12, rue des Bouchers	Marseille
10	BOTTM	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.	Tsawassen

Figure 11
First 10 rows Shippers

Northwind_DW/postgres@PostgreSQL 10 (x86) ▾

Query Editor Query History

```
1 SELECT * FROM nw_customers_dim LIMIT 10;
```

Data Output Explain Messages Notifications

	customer_id	contact_name	contact_title	address	city	re
1	ALFKI	Maria Anders	Sales Representative	Obere Str. 57	Berlin	[in]
2	ANATR	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	[in]
3	ANTON	Antonio Moreno	Owner	Mataderos 2312	México D.F.	[in]
4	AROUT	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	[in]
5	BERGS	Christina Berglund	Order Administrator	Berguvägen 8	Luleå	[in]
6	BLAUS	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim	[in]
7	BLONP	Frédérique Citeaux	Marketing Manager	24, place Kléber	Strasbourg	[in]
8	BOLID	Martin Sommer	Owner	C/ Araquil, 67	Madrid	[in]
9	BONAP	Laurence Leblan	Owner	12, rue des Bouchers	Marseille	[in]
10	BOTTM	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.	Tsawassen	BC

See figures 12-15 for the total row count of each table after data was loaded using foreign data wrapper (see Table 2 for more details).

Figure 12
Row Count Employees

Northwind_DW/postgres@PostgreSQL 10 (x86) ▾

Query Editor Query History

```
1 SELECT
2 COUNT(*)
3 FROM nw_employee_dim;
```

Data Output Explain Notifications

	count
1	9

Figure 13
Row Count Products

Northwind_DW/postgres@PostgreSQL 10 (x86) ▾

Query Editor Query History

```
1 SELECT
2 COUNT(*)
3 FROM
4 nw_products_dim;
```

Data Output Explain Messages

	count
1	77

Figure 14
Row Count Customers

Northwind_DW/postgres@PostgreSQL 10 (x86) ▾

Query Editor Query History

```
1 SELECT
2 COUNT(*)
3 FROM nw_customer_dim;
```

Data Output Explain Notifications

	count
1	91

Figure 15
Row Count Shippers

Northwind_DW/postgres@PostgreSQL 10 (x86) ▾

Query Editor Query History

```
1 SELECT
2 COUNT(*)
3 FROM nw_shipper_dim;
```

Data Output Explain Messages

	count
1	6

Conclusions and Learning Experience

Dimensional Models are designed to optimize storage and data retrieval, as a result, it is vital that the model is created with efficiency in mind. This portfolio assignment forced this perspective and required a switch from a relational database design mindset. This included treating dimensions as actual dimensions rather than fact tables.

Personally, the biggest challenge of this project was the concept of loading data from another dataset in PostgreSQL. This was not something that I have ever had to do or seen done. This was a valuable lesson as it can be very useful in the future when working with dimension models. The second challenge was - as mentioned - the perspective shift from a relational database to a dimensional model. While the differentiation is relatively simple, there were several components that were missed through the portfolio milestones.

With these experiences, I would approach a similar project with greater thought and planning. Just as highlighted in the ETL process, understanding the business problem and case is vital before creating a dimensional model. For this project, I was lacking a stronger understanding of the needs of the model.

Furthermore, building data warehouse systems requires routine review. For example, throughout this assignment there were several milestones (just as with organizational projects). While it is important that a project continues to move forward, reflecting on the existing work is vital too. In this case it was supported by instructor feedback, however, in an organizational project this can easily be overlooked but can prevent major issues in the model later on.

References

- Create Foreign Data Wrapper. (2021, February). Retrieved February 14, 2021, from <https://www.postgresql.org/docs/9.3/sql-createforeigndatawrapper.html>
- Kimball, R. & Ross, M. (2013). The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling (3rd Edition). *John Wiley & Sons, Inc.*