By: Christopher Singleton Business Intelligence Analyst

This is my data modeling conversion from the Northwind OLTP Operational Database to the DWNorthwind OLAP Data Warehouse. Please note that I made this picture really large so that I can plan my arrangements for OLTP to OLAP conversion.

Northwind (OLTP) Operational Database

Northwind Database (Large Full Scale Photo)

By: Christopher Singleton

Employees

Column Name	Data Type	Allow Nulls
EmployeeID	int	
LastName	nvarchar(20)	
FirstName	nvarchar(10)	
Title	nvarchar(30)	✓
TitleOfCourtesy	nvarchar(25)	✓
BirthDate	datetime	✓
HireDate	datetime	✓
Address	nvarchar(60)	✓
City	nvarchar(15)	✓
Region	nvarchar(15)	✓
PostalCode	nvarchar(10)	✓
Country	nvarchar(15)	✓
HomePhone	nvarchar(24)	✓
Extension	nvarchar(4)	✓
Photo	image	✓
Notes	ntext	✓
ReportsTo	int	✓
PhotoPath	nvarchar(255)	✓

Orders

Column Name	Data Type	Allow Nulls
OrderID	int	
CustomerID	nchar(5)	✓
EmployeeID	int	✓
OrderDate	datetime	✓
RequiredDate	datetime	✓
ShippedDate	datetime	✓
ShipVia	int	✓
Freight	money	✓
ShipName	nvarchar(40)	✓
ShipAddress	nvarchar(60)	✓
ShipCity	nvarchar(15)	✓
ShipRegion	nvarchar(15)	✓
ShipPostalCode	nvarchar(10)	✓
ShipCountry	nvarchar(15)	✓

Categories

Column Name	Data Type	Allow Nulls
CategoryID	int	
CategoryName	nvarchar(15)	
Description	ntext	✓
Picture	image	✓

Products

Column Name	Data Type	Allow Nulls
ProductID	int	
ProductName	nvarchar(40)	
SupplierID	int	✓
CategoryID	int	✓
QuantityPerUnit	nvarchar(20)	✓
UnitPrice	money	✓
UnitsInStock	smallint	✓
UnitsOnOrder	smallint	✓
ReorderLevel	smallint	✓
Discontinued	bit	

Suppliers

Column Name	Data Type	Allow Nulls
SupplierID	int	
CompanyName	nvarchar(40)	
ContactName	nvarchar(30)	✓
ContactTitle	nvarchar(30)	✓
Address	nvarchar(60)	✓
City	nvarchar(15)	✓
Region	nvarchar(15)	✓
PostalCode	nvarchar(10)	✓
Country	nvarchar(15)	✓
Phone	nvarchar(24)	✓
Fax	nvarchar(24)	✓
HomePage	ntext	✓

Customers

Column Name	Data Type	Allow Nulls
CustomerID	nchar(5)	
CompanyName	nvarchar(40)	
ContactName	nvarchar(30)	✓
ContactTitle	nvarchar(30)	✓
Address	nvarchar(60)	✓
City	nvarchar(15)	✓
Region	nvarchar(15)	✓
PostalCode	nvarchar(10)	✓
Country	nvarchar(15)	✓
Phone	nvarchar(24)	✓
Fax	nvarchar(24)	✓

Order Details

Column Name	Data Type	Allow Nulls
OrderID	int	
ProductID	int	
UnitPrice	money	
Quantity	smallint	
Discount	real	

EmployeeTerritories

Column Name	Data Type	Allow Nulls
EmployeeID	int	
TerritoryID	nvarchar(20)	

Territories

Column Name	Data Type	Allow Nulls
TerritoryID	nvarchar(20)	
TerritoryDescription	nchar(50)	
RegionID	int	

Region

Column Name	Data Type	Allow Nulls
RegionID	int	
RegionDescription	nchar(50)	

CustomerCustomerDemo

Column Name	Data Type	Allow Nulls
CustomerID	nchar(5)	
CustomerTypeID	nchar(10)	

CustomerDemographics

Column Name	Data Type	Allow Nulls
CustomerTypeID	nchar(10)	
CustomerDesc	ntext	✓

Shippers

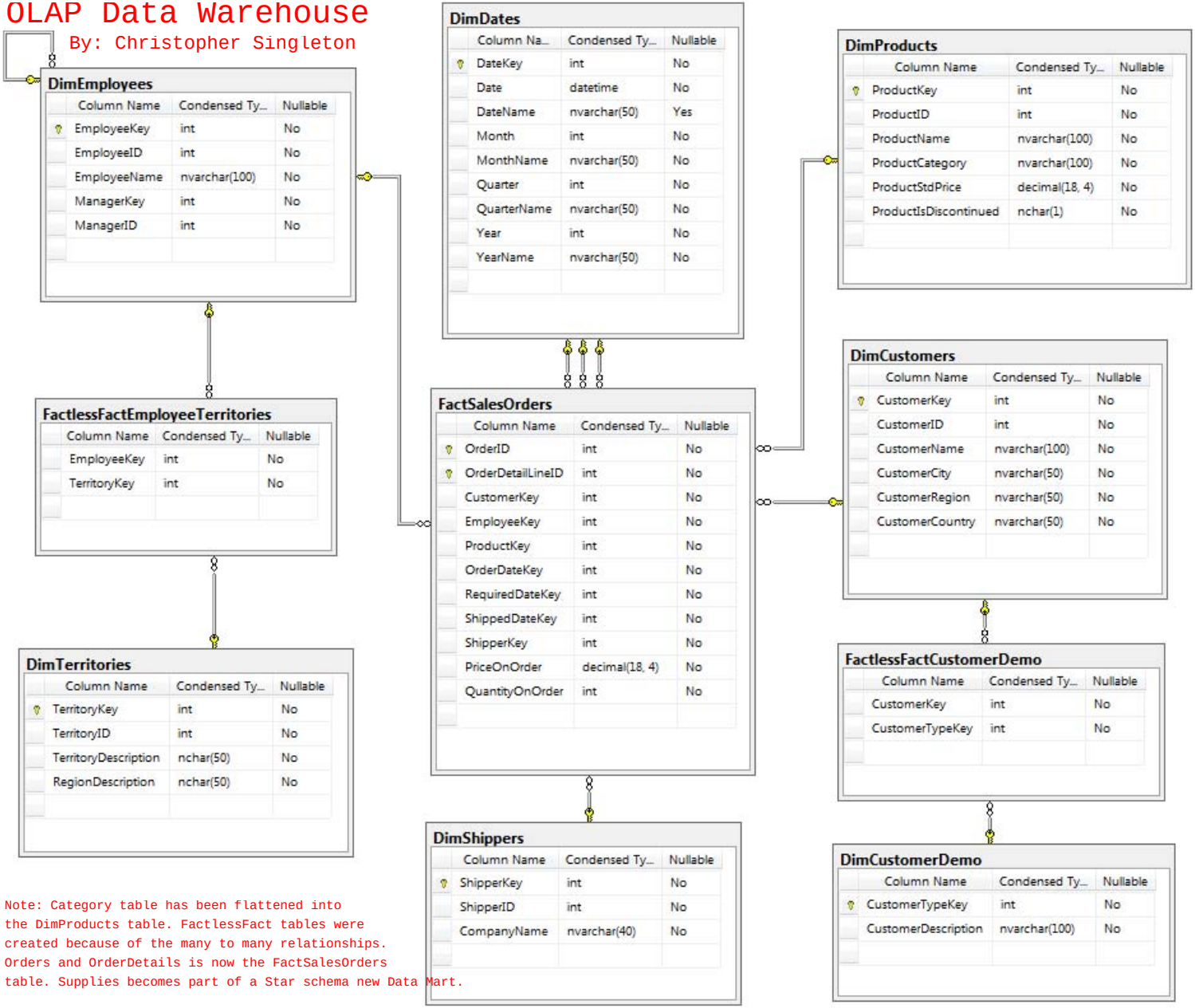
Column Name	Data Type	Allow Nulls
ShipperID	int	
CompanyName	nvarchar(40)	
Phone	nvarchar(24)	✓

DWNorthwind (OLAP) Data Warehouse

Note: [Factless Fact] tables were created because of the many to many relationships. Categories table has been flattened into the [Dim Products] table. [Order Details] table and Orders becomes [Fact Sales Orders]. Supplies can become another star schema Data Mart in the future.

OLAP Data Warehouse

By: Christopher Singleton

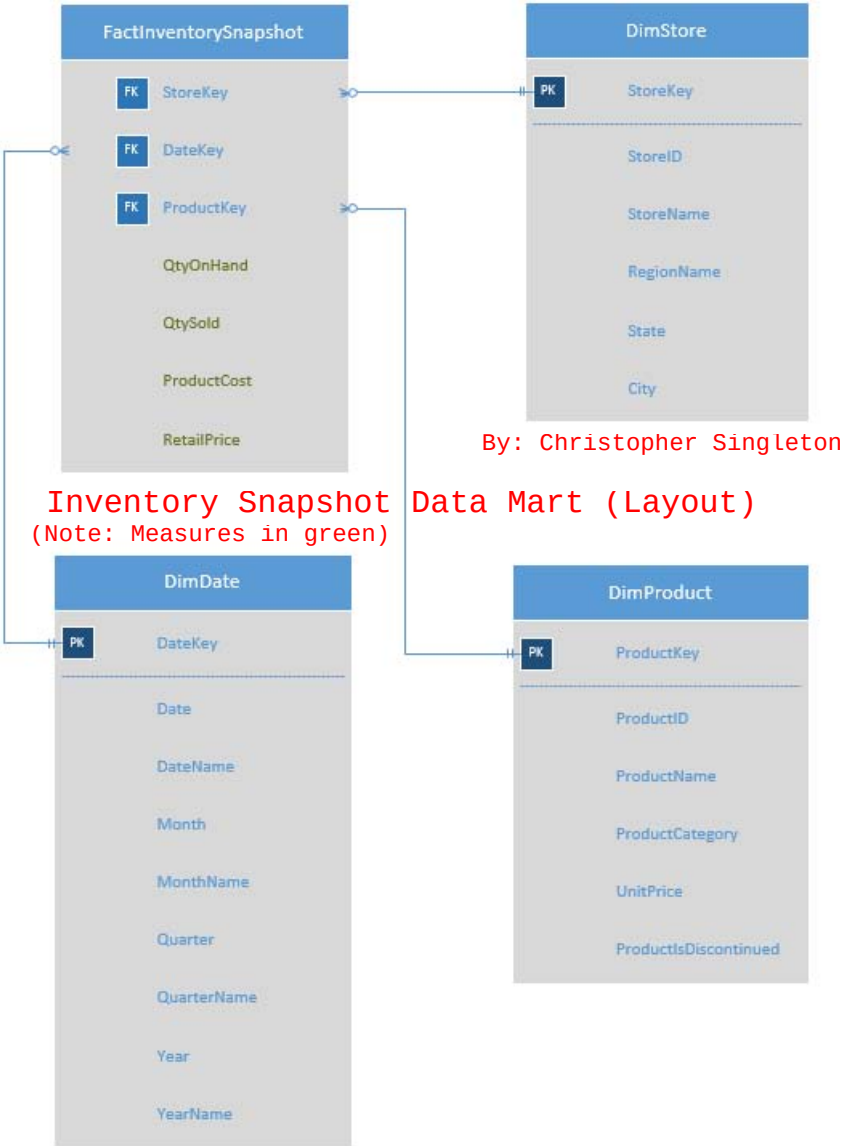


Note: Category table has been flattened into the DimProducts table. FactlessFact tables were created because of the many to many relationships. Orders and OrderDetails is now the FactSalesOrders table. Supplies becomes part of a Star schema new Data Mart.

Inventory Table Snapshot (Layout)

This is my inventory snap-shot Data Mart Layout from Visio. As you can see I allowed capability for more than one Star-schema Data Mart in the future. The measures are in green in the [Fact Inventory Snapshot] table, which allows the user to slice in the cube while calculating inventory at any particular given point in time.

The reason of why I came up with this very basic logical design is because it meets all the requirements in doing the calculations for (GMROI) Gross Margin Return On Investment. This design is also very easy to explain to the end user of what is going on and how it fits in with getting all the (KPI's) Key Performance Indicators. All the measures in the [Fact Inventory Snapshot] table can be calculated directly on the fly with the interconnecting tables that satisfy the minimum requirements in being able to retrieve all the data needed for GMROI. The end users can have their analysis of all measures by products or/and date. The [Dim Store] table allows for expandability in the future for other store locations, otherwise it is not needed nor would it change anything. The end user can easily get all the measures sliced by product and/or dates.


FactInventorySnapshot
FK StoreKey
FK DateKey
FK ProductKey
QtyOnHand
QtySold
ProductCost
RetailPrice
DimStore
PK StoreKey
StoreID
StoreName
RegionName
State
City
By: Christopher Singleton
Inventory Snapshot Data Mart (Layout)
(Note: Measures in green)
DimDate
PK DateKey
Date
DateName
Month
MonthName
Quarter
QuarterName
Year
YearName
DimProduct
PK ProductKey
ProductID
ProductName
ProductCategory
UnitPrice
ProductIsDiscontinued

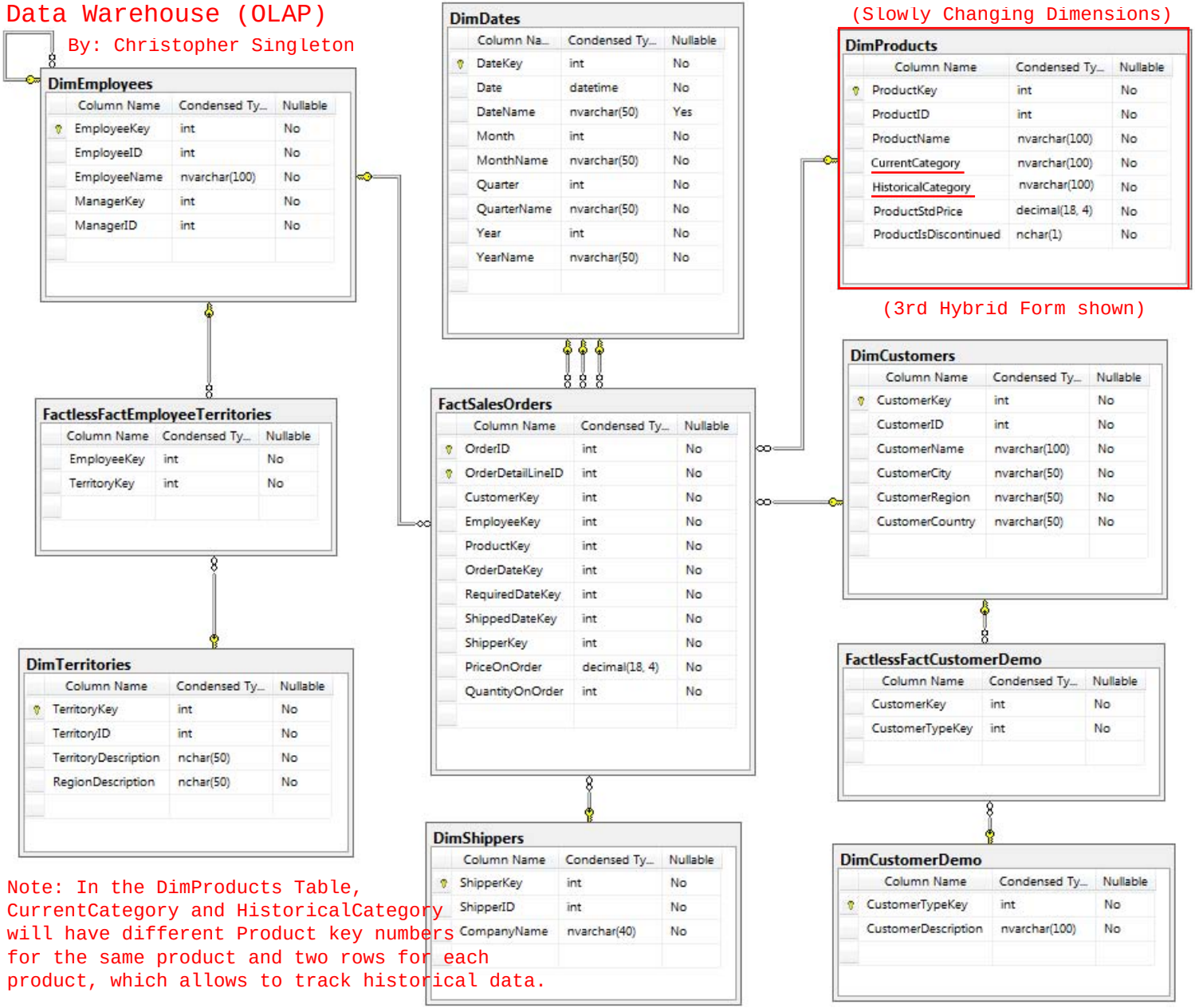
DWNorthwind(OLAP) DataWarehouse (SlowlyChangingDimensions)

[Product Category] has been changed to reflect how slowly changing dimensions can be handled as a hybrid type 3 conversion. Note: In [Dim Products] you now have a [Current Category] column and [Historical Category] column to show how product categorical dimensions can be tracked over changes in time. This means that a product can have two rows and a different product key on each row for the same product. One product key for the [Current Category] and the other for tracking the historical data of the same product. The [Historical Category] holds changes for previous years, while [Current Category] holds the current year data. Note also that composite keys were added where needed to facilitate in making it easier to query using SQL.

Data Warehouse (OLAP)

By: Christopher Singleton

(Slowly Changing Dimensions)



Note: In the DimProducts Table, CurrentCategory and HistoricalCategory will have different Product key numbers for the same product and two rows for each product, which allows to track historical data.