



Global Knowledge®

Expert Reference Series of White Papers

Extending Role Security in Analysis Services for SQL Server

Extending Role Security in Analysis Services for SQL Server

Bill Kenworthy, Global Knowledge Instructor

Abstract

The goal of this paper is to describe a method that allows users to browse an Analysis Services cube, limiting the user's access by the user account used to connect to the cube. The method makes use of a many-to-many relationship established in the underlying online transaction processing (OLTP) database. Instead of maintaining many roles and assigning users to roles, the technique described uses a table in the data warehouse to link users or groups to the parts of a dimension to which they are allowed access. In a large organization, this simplifies security maintenance greatly.

Introduction

Given an OLTP schema, implement a data warehouse schema with tables that connect a user account using a many-to-many relationship from the user account to a dimension in the table. A cube is then created from the data warehouse which contains the many-to-many relationship. A role is created to implement dimensional security to limit what members of the dimension the user is able to view.

The Steps

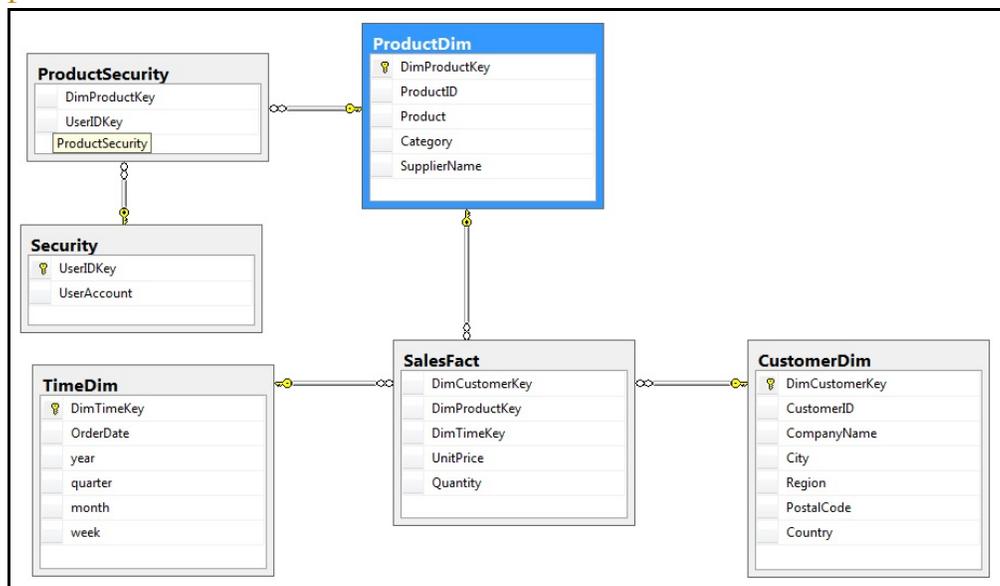


Diagram 1. Data warehouse schema

Diagram 1 shows the data warehouse schema and the relationship between the Security table and the Product dimension. This database was created by extracting appropriate data from the Northwind database, which is one of the sample databases supplied with SQL Server in the past. Northwind is a small, simple database that is useful for testing and prototyping. The Northwind database is still available from CodePlex as a downloadable script. After creating a simple data warehouse with a single-fact table and three dimensions (time, customer, and product)—two tables were added to implement the security model. Windows user accounts are listed in the Security table and linked to the Product dimension through the ProductSecurity table.

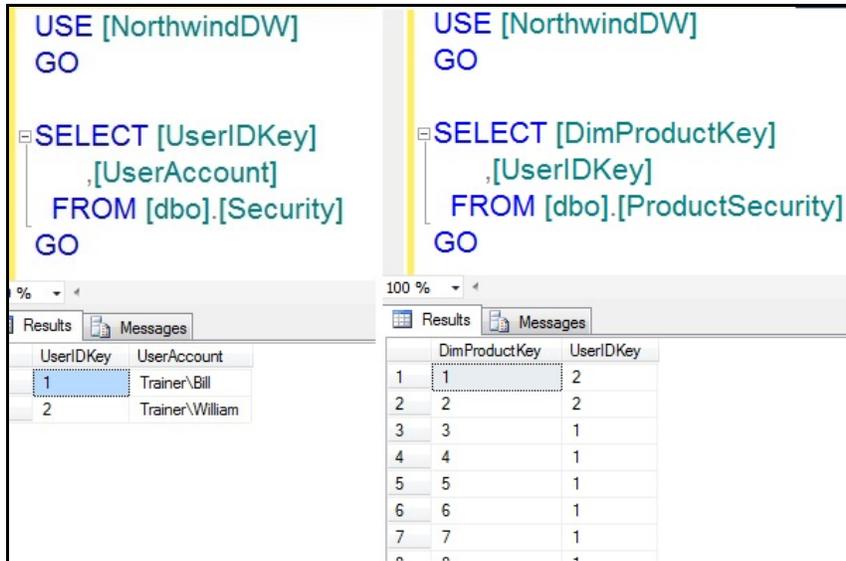


Diagram 2. Contents of the two security tables, Security and Product Security

The contents of my security table have two user accounts, Bill and William. The ProductSecurity table will link the user accounts to the dimension surrogate key, DimProductKey, in ProductDim. Bill is mapped to all products except to Beverages, whereas William is mapped to just Beverages. The simplicity employed here will make it easier to verify the results when I apply the role in the Cube browser.

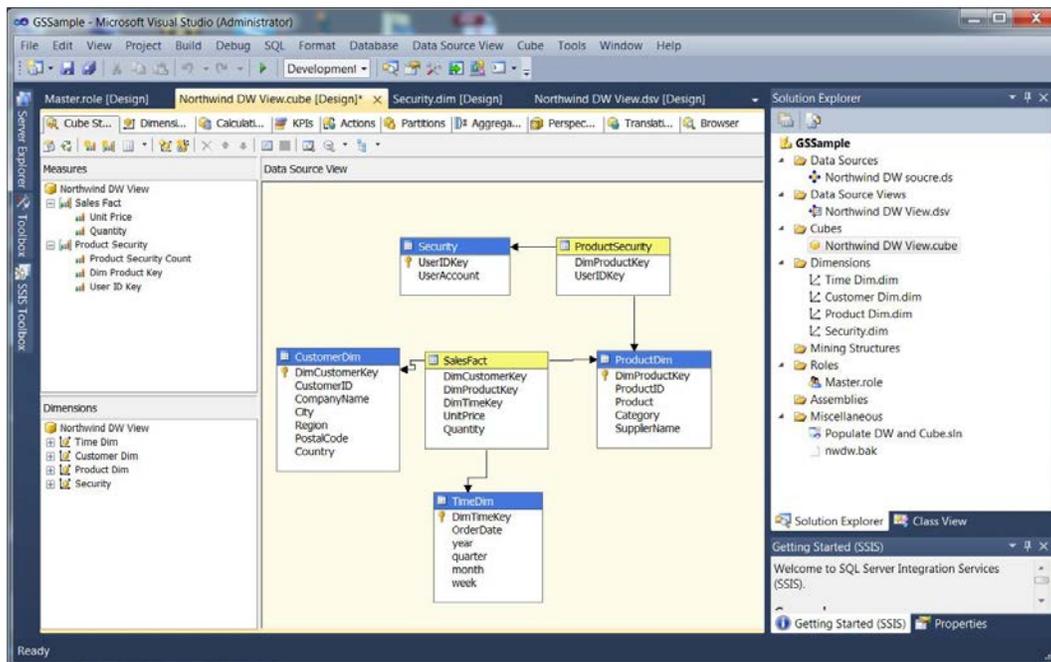


Diagram 3. The data source view for the cube

The data source view brings in all the tables from my simple data warehouse. Because of the many-to-many relationship between the ProductDim and Security tables, I will build the first part of the project using the Cube wizard and the three-dimensional tables (CustomerDim, ProductDim, and TimeDim).

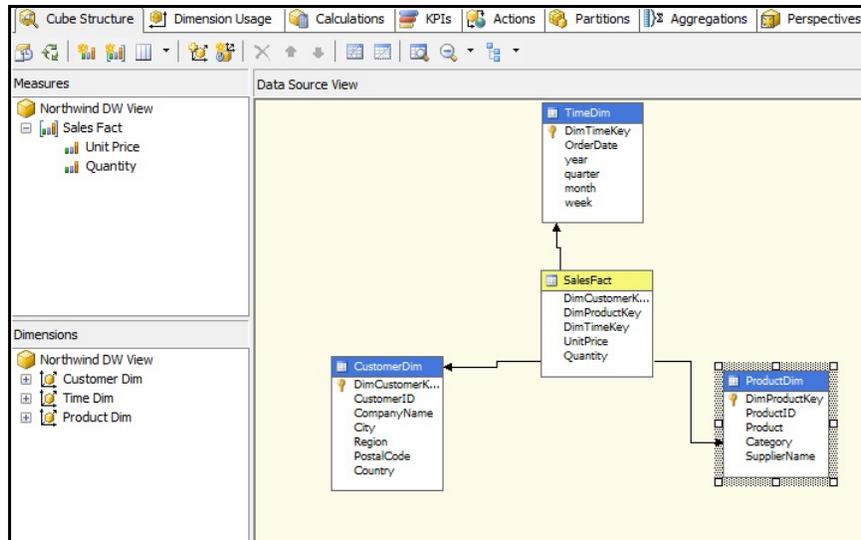


Diagram 4. The Basic cube with four tables

The Basic cube will need a measure group that links the ProductDim and Security tables. The relationship between ProductDim and Security is a many-to-many relationship in the underlying data source view, so a measure group is created that is mapped to the ProductSecurity linking table.

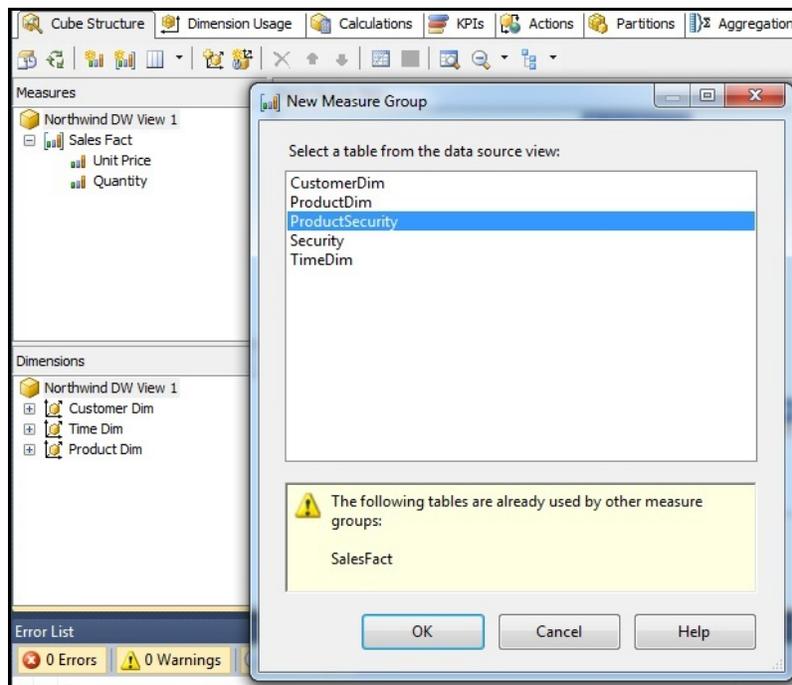


Diagram 5. Creating the second measure group

Once the Product Security measure group and Security dimension are part of the cube structure, we must define the dimension usage in the Dimension Usage editor, which is the second tab from the left in the cube editor.

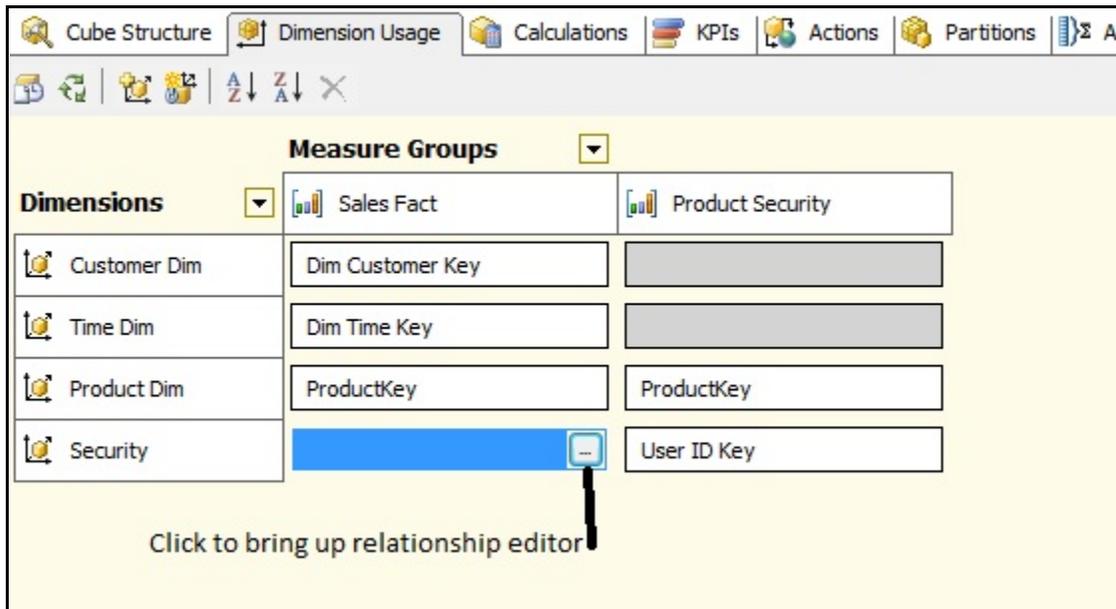


Diagram 8. Add a relationship between the Security table and the Sales Fact measure

Both measure groups are in place. A relationship must be added to connect the measure groups in a many-to-many relationship, which ties the user ID in the security table to the product key in Product Dim. The Product Security measure group serves as the linking element in the many-to-many relationship. Click on the ellipsis button to bring up the relationship editor.

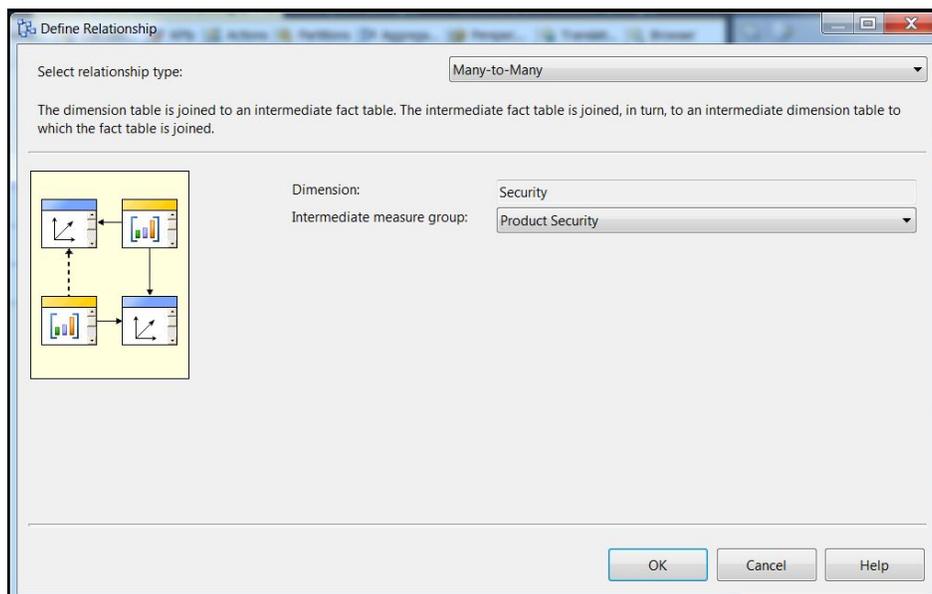


Diagram 9. Defining a many-to-many relationship

The interface is simple. Choose “Many-to-Many” as the relationship type and choose “Product Security” as the intermediate measure group. This will then link the user ID in the security table to the Sales Fact table through the Product Security measure group. The Product Security measure group is in place simply to allow this relationship to be defined. The cube should process and deploy at this time. You can browse the cube in the cube editor browser. You should implement some dimensional hierarchies; this step is left up to the reader.

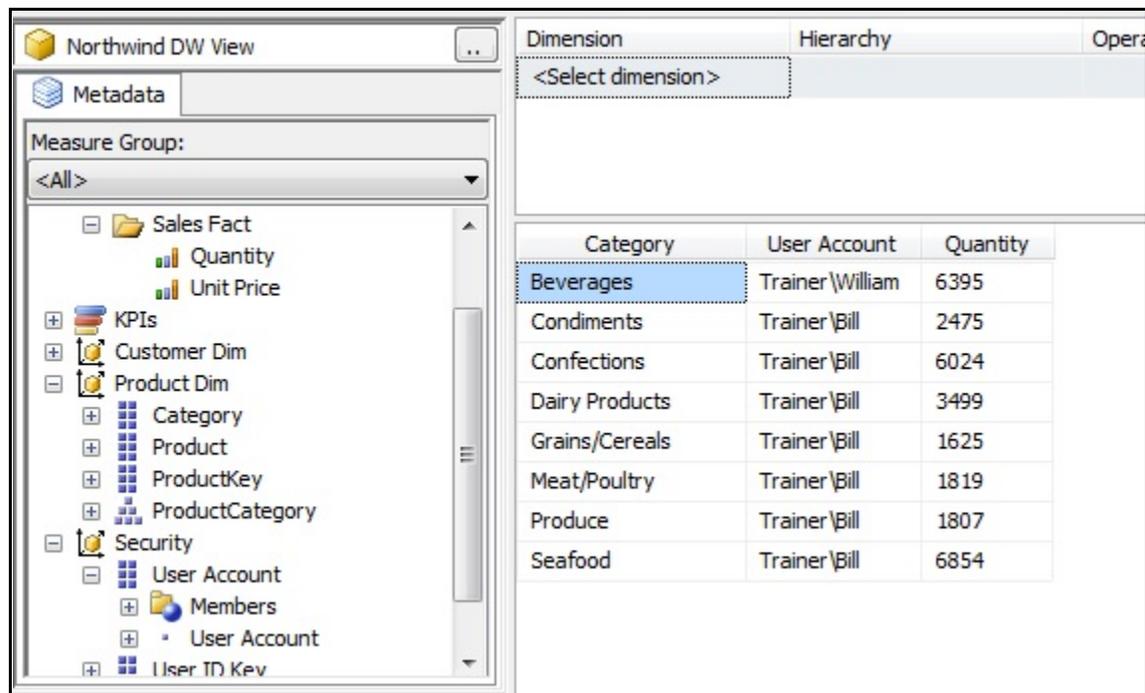


Diagram 10. Browsing the cube as a check that it is functioning before moving on

Create a role. I renamed mine as Master.Role. In the Cube tab of the Role Editor, set the access attribute to Read. Then move to the Dimension Data table. In that tab, choose Security from the Dimension combo box selector, and enter the MDX statement in the Allowed member set text box, as shown in Diagram 11. Check the box at the bottom of the interface, Enable Visual Tools.

The MDX Statement

```
STRTOMEMBER("[Security].[User Account].&[" + UserName + ""]")
```

Concatenates the output of the UserName function with the appropriate dimensional context and return this string as the definition of a member of the security dimension. Dimensional security has less overhead than Cell Data security, resulting in better performance browsing.

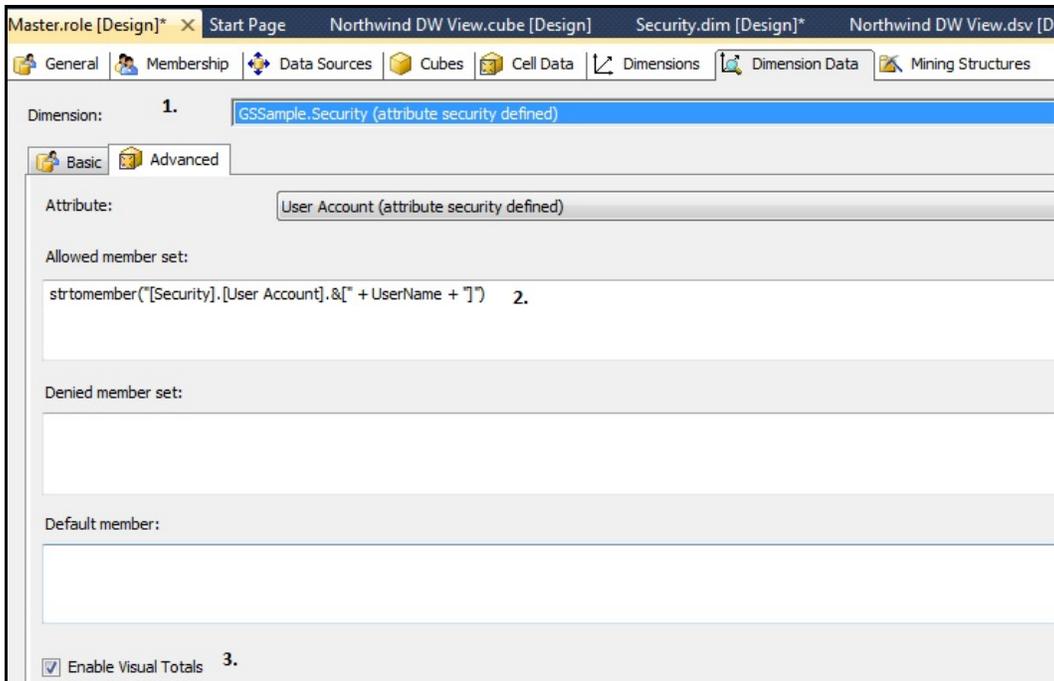


Diagram 11. Master.Role, the dimensional security

Once the cube has been deployed and processed successfully, you can move to the cube browser, set the security context to the role that has been created, and note that access to the product dimension is limited by the relationship that is defined in the security tables in the data source view.

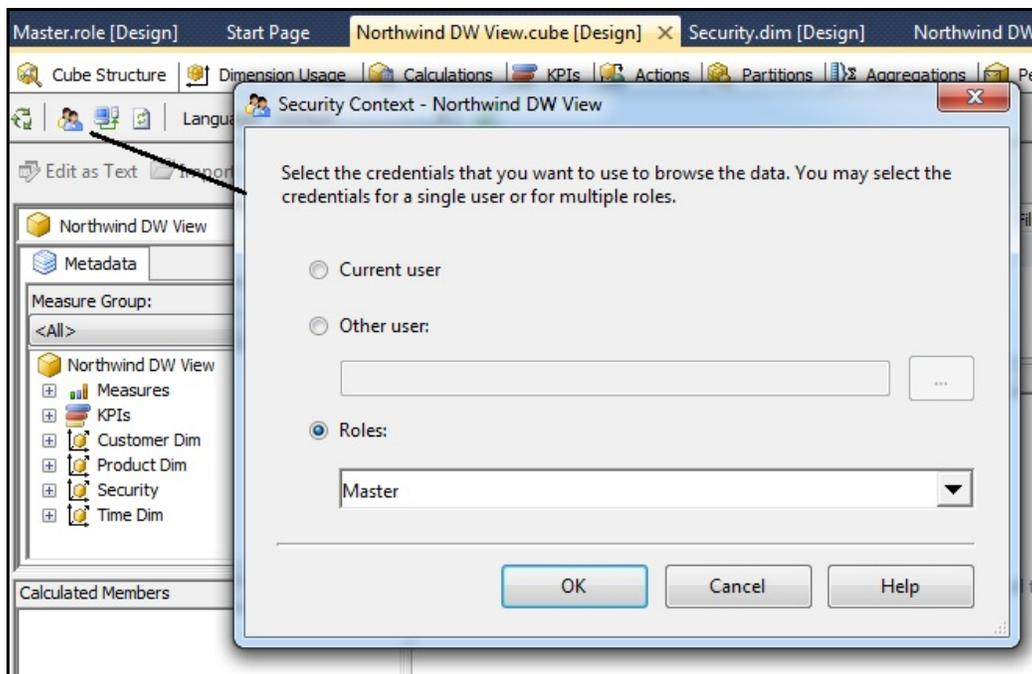


Diagram 12. Set the browser security context

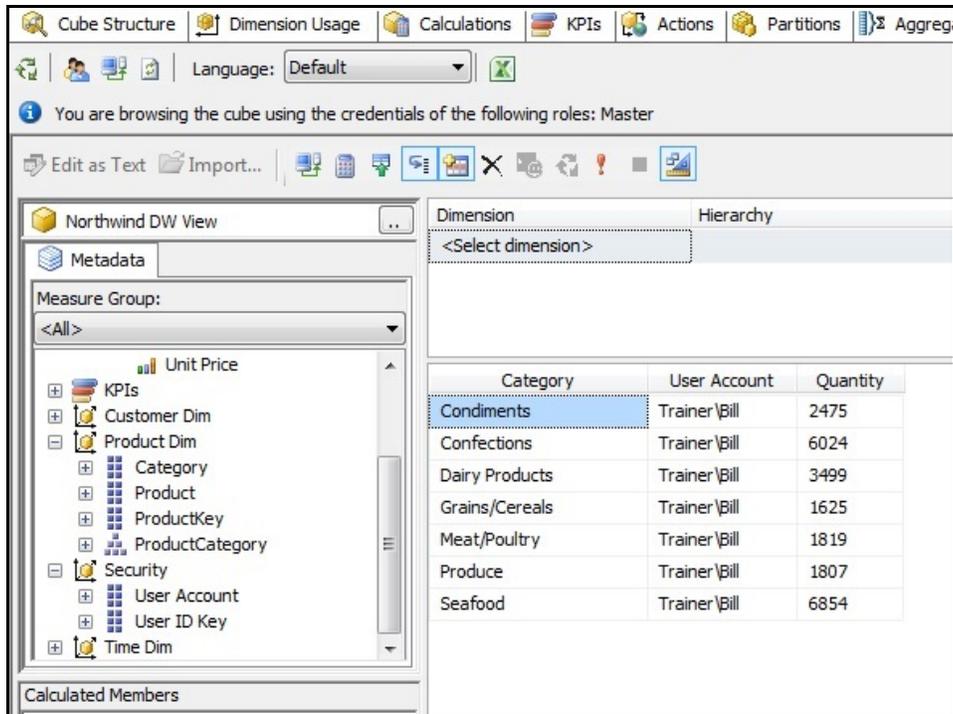


Diagram 13. View the effect of security logged in as Bill (no Beverages are displayed)

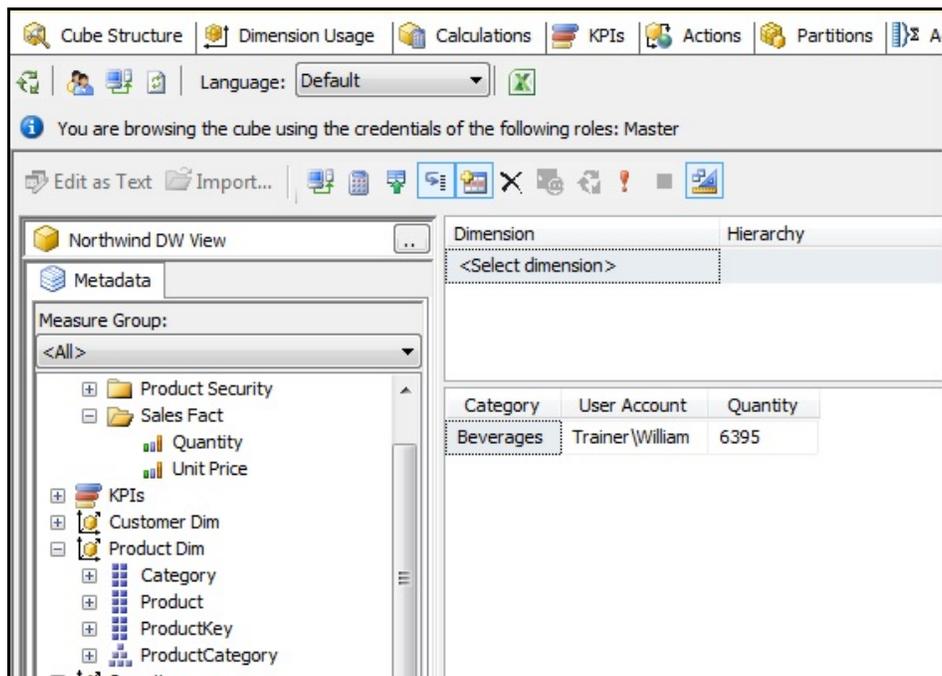


Diagram 14. View the effect of security logged in as William

Only one role is necessary; the MDX supplies the user account context dynamically. To test this role, it requires that the user log out and log back in to test that the security model is working properly.

Conclusion

The role security model can be challenging to maintain in a large enterprise environment. It is not uncommon to have security in place on an OLTP system prior to rolling out a data warehouse. Using this technique, existing tables with user account information can be used directly to secure cube contents rather than converting to role based security in the cube. By using and implementing a many-to-many relationship in a cube, you can more easily secure your Analysis Services and access sensitive information contained within it.

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge through training.

[Implementing Data Models and Reports with Microsoft SQL Server 2012 \(M10778\)](#)

[Designing Business Intelligence Solutions with Microsoft SQL Server 2012 \(M20467\)](#)

Visit www.globalknowledge.com or call **1-800-COURSES (1-800-268-7737)** to speak with a Global Knowledge training advisor.

About the Author

Bill Kenworthy is a business intelligence developer who's been using Microsoft SQL Server since version 4.2. Bill works in the fulfillment industry, providing solutions for data consolidation and consistency, reporting, and analysis. He has been teaching for Global Knowledge for over 10 years and developing database applications for more than 20 years.