

## OLAP Drill-through Table Considerations

M. Michelle Buchecker, SAS Institute, Inc.

### ABSTRACT

When creating an OLAP cube, you have the option of specifying a drill-through table, also known as a Show Details table. This quick tip discusses the implications of using your detail table as your drill-through table and explores some viable alternatives.

### INTRODUCTION

Creating an OLAP cube is easy. Creating a good OLAP cube is difficult. One aspect of creating a cube is the ability to allow the consumer to view the details that went into creating the summarized numbers. This view is made possible with a drill-through table. However, the drill-through table can be problematic and could cause confusion for your consumer.

This paper looks at these problems and how to mitigate them.

### What Is a Drill-through Table?

An **OLAP cube** contains large volumes of data stored in summary form. A statistic associated with a variable, typically a numeric variable, is called a **measure**. The classification variables that you want to summarize by are called **levels**. The levels are then organized into hierarchies.

In this view, we are looking at the Sum of Total\_Retail\_Price for Golf equipment and Golf Clothes. The view is broken down by Age Group, and then within the 31-45 year age group it is further broken down by gender.

Product Group		+ Golf	+ Golf Clothes
Measures		Sum of Total_Retail...	Sum of Total_Retail...
Customer Age Group	Customer Gender		
+ 15-30 years		\$737,971.09	\$255,317.13
- 31-45 years	+ Female	\$325,298.66	\$124,492.21
	+ Male	\$360,667.12	\$113,338.27
<b>31-45 years</b>		\$685,965.78	\$237,830.48
+ 46-60 years		\$700,862.52	\$227,919.94
+ 61-75 years		\$448,157.52	\$147,087.75

Display 1. OLAP Cube

We can see that Females spent over \$10,000 more on Golf Clothes in that age group, but they spend \$35,000 less on Golf equipment. A drill-through table is defined for the cube. Therefore, if we right-click on the \$325,298.66 of the Golf equipment for Females, one of the options is **Drill Through Details**.

Product Group		+	-	Golf	+	-	Golf Clothes
Measures		Sum of Total_Retail...			Sum of Total_Retail...		
Customer Age Group	Customer Gender						
+	+	15-30 years			\$737,971.09		\$255,317.13
-	+	31-45 years	+	Female	\$325,298.66		\$124,402.21
	+		+	Male	\$360,667.2		
31-45 years		\$685,965.2					
+	+	46-60 years			\$700,862.5		
+	+	61-75 years			\$448,157.5		

Copy
  
Drill Through Details
  
Conditional Highlight
  
Totals
  
Create Percent Of
  
Table View

**Display 2. Right-click Data Value**

A pop-up window appears that displays all columns from the defined drill-through table, and the rows that match the drill-through criteria. In this instance a WHERE clause of Product\_Group='Golf' and Gender='Female' and Age\_Group='31-45 years' is applied.

Orion Star Customer Order Cube-Detail for (Female,(Golf,Sum of Total_Retail_Price)) (read-only)							
	Customer_ID	Employee_ID	Product_ID	Date_ID	Company	CostPrice_Per_Unit	Customer_
1	2520	120925	240200100097	03JAN2006	Orion UK	124.5	
2	59770	120920	240200100060	03JAN2006	Orion UK	26.65	
3	70553	99999999	240200100118	04JAN2006	Logistics	89.55	
4	70553	99999999	240200100180	04JAN2006	Logistics	62.9	
5	40411	120450	240200100053	04JAN2006	Orion Germany	44.95	
6	20643	121051	240200100234	04JAN2006	Orion USA	24.2	
7	17021	120450	240200100180	06JAN2006	Orion Germany	62.9	
8	55391	99999999	240200100046	06JAN2006	Logistics	9.15	
9	51751	120134	240200100136	08JAN2006	Orion Australia	463.55	
10	51751	120134	240200100186	08JAN2006	Orion Australia	0.65	
11	51751	120134	240200100235	08JAN2006	Orion Australia	49	
12	93330	120364	240200100240	08JAN2006	Orion France	13.3	
13	12844	120922	240200100069	08JAN2006	Orion UK	4.55	
14	16908	99999999	240200100229	09JAN2006	Logistics	78.5	
15	15888	99999999	240200100007	11JAN2006	Logistics	11.8	
16	12844	120926	240200100077	12JAN2006	Orion UK	90.7	
17	12844	120926	240200100220	12JAN2006	Orion UK	42.75	
18	89459	120838	240200100095	13JAN2006	Orion Spain	52.55	
19	57409	120131	240200100134	16JAN2006	Orion Australia	4.75	
20	78765	120837	240200100043	16JAN2006	Orion Spain	69.4	
21	79614	120535	240200100129	16JAN2006	Orion Holland	6.1	
22	86499	99999999	240200100007	18JAN2006	Logistics	11.8	
23	28984	120925	240200100198	18JAN2006	Orion UK	29.55	

**Display 3. Drill-through Table Data**

## DANGERS AHEAD

If an OLAP cube is built from either a Detail table or a Star Schema, the OLAP cube is a static, summarized copy of data at the time the cube was built or last updated. This could result in the drill-through table having more recent data than the OLAP cube. There are also other risks that we need to ponder.

### UNDERLYING DATA CAN GET UPDATED WITHOUT CUBE BEING UPDATED

Lets assume that a cube is built from a Detail Table named **Orders** at 3:00 PM on 09SEP2013. At 12:01 AM 10SEP2013 a job runs and populates the Orders table with sales from the previous day. If a user views the cube, they are looking at the data as of 3:00 PM on 09SEP2013. If the user clicks **Drill Through Details**, they can see the data from the 12:01 AM 10SEP2013 job. It is very likely that the user then examines the resulting drill-through data and realizes that it does not match the cube data and calls you.

### DRILL-THROUGH TABLE CAN BE ONLY A SINGLE TABLE

Now, assume that a cube is built from a Star Schema that has one fact table and four dimension tables. As in the Golf equipment example above, SAS wants to submit a WHERE clause of Product\_Group='Golf' and Gender='Female' and Age\_Group='31-45 years'. The problem is that the measures variables are in the fact table, Product\_Group is in one dimension table, and Gender and Age\_Group are in a separate dimension table.

OLAP Cubes only permit a single table to be listed as a --through table, so we cannot point to all of the tables that make up the star schema.

### CONFIDENTIAL DATA MAY BE EXPOSED

As we saw in the Golf example above, **all** columns from the defined drill-through table are displayed unless otherwise controlled. This could have implications if one of those columns contains sensitive data such as Social Security number or HIPAA confidential information.

### UNDERLYING TABLE CAN GET DELETED

The OLAP cube that is built from a Detail Table or a Star Schema is a copy of the data. As a result, the cube no longer relies on the underlying detail table or star schema tables. It is possible that someone might delete the table, or move it to a different location. The cube itself is fine, until you try to Drill Through to Details.

## MITIGATING THOSE FACTORS

Here are some options for solving those problems.

### UNDERLYING DATA CAN GET UPDATED WITHOUT CUBE BEING UPDATED

#### Create a Separate, Static Table

When we built the cube built from a Detail Table named **Orders** at 3:00 PM on 09SEP2013, at that time you could create a **separate** copy of the detail table. For example, create a separate copy of the Orders table and call it **Orders DT**. You can now use that as your drill-through table. Now you are guaranteeing that the same data that populated the cube is what is being used for the drill-through table.

Pros: No concerns that the data does not match.

Cons: The disk space of the detail data is now doubled. Also, when you need to update the cube, you must re-create the drill-through table.

## Create a View to the Original Table

Alternatively, you could create a view to the original table that contains a WHERE clause based on the date that you created the cube. For example

```
Proc sql;
Create view orion.order_DT as
Select *
From orion.order
Where order_date <= '09SEP2013:15:00:00'dt;
```

Now you are assuring that the same data that went in to the cube is what is being used for the drill-through table.

Pros: No need to duplicate the data.

Cons: There must be a variable in your table that has a datetime stamp. Also, normal caveats about views are in place and the view instructions execute every time it is accessed. Therefore, it could be slow to bring up the drill-through data.

## DRILL-THROUGH TABLE CAN BE ONLY A SINGLE TABLE

OLAP Cubes only permit a single table to be listed as a drill-through table, so we cannot point to all of the tables that make up the star schema. If you want to have a drill-through table for the cube, you must manually join the fact and dimension tables together and create either an output table or a view.

Pros: This is your only choice if you want a drill-through table. Remember, drill-through tables are **optional**. Your cube does not have to have one.

Cons: If you are creating a table of the join, it will double the disk space needed for the input data. Also, when you need to update the cube, you must re-create the drill-through table. If you create a view of the join, it executes every time it is accessed. Which means every time a user clicks on Drill Through Details, SAS has to join all the tables. Every time!

## CONFIDENTIAL DATA MAY BE EXPOSED

### Ensure That Your Metadata Permissions Are Properly Set

Remember that **all** columns from the defined drill-through table are displayed unless otherwise controlled. We can focus on the otherwise controlled part. When a user goes to view a cube, their metadata credentials are passed to the SAS Metadata Server. When the user clicks **Drill Through Details** those metadata credentials are checked against the defined drill-through table. If the user does not have Read permission on the table, then the pop-up window is not displayed. More importantly, if the user has Read permission on the table, but does not have Read permission for a column within the table, then that column does not appear in the pop-up window.

Pros: Permissions hide confidential information from those who should not be seeing it.

Cons: It requires the SAS Platform Administrator to go into the necessary columns of each drill-through table and deny Read permission for the Users and or Groups. But, you should be doing this anyway.

### Create a Table / View with Specific Columns

Alternatively, you could create a table or view to the original table that contains only the columns that you want in the pop-up window.

Pros: As the OLAP Cube builder, you have full control over the columns that appear in the pop-up window and do not have to talk to your Platform Admin.

Cons: If you are creating a table, then you have doubled the disk space needed for the input data. Also, when you need to update the cube, you must re-create the drill-through table. If you create a view, the instructions execute every time the view is accessed.

## UNDERLYING TABLE CAN GET DELETED

You can set permissions that restrict who can delete the table. For people who have permission, just tell them to stop deleting that table.

## CONCLUSION

Clearly there is no “one size fits all” approach for defining a drill-through table. You should determine which of the above options is the best suited for your needs. A combination of options may be the best approach.

## RECOMMENDED TRAINING

The following related training is available from SAS

- [\*Designing, Tuning, and Maintaining SAS OLAP Cubes\*](#)
- [\*SAS OLAP Environment: Administration\*](#)

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at the following address:

Michelle Buchecker  
SAS Institute  
180 N. Stetson Ave, Suite 1600  
Chicago, IL 60601  
[Michelle.Buchecker@sas.com](mailto:Michelle.Buchecker@sas.com)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.