

# Essbase (ASO): Clearing Data Using CrossJoin

Having been working on an ASO project for the last couple of months, I have learned a lot about Essbase and its related software. One of the things that gave me trouble at first was the syntax of CrossJoin in ASO's MDX language. More specifically, I was having trouble trying to nest multiple CrossJoin's together when I was trying to clear out a certain portion of data that included more than two dimensions. A CrossJoin is simple; it returns the cross-product of two sets from two different dimensions. But what if I want to return the cross product of four different dimensions? That one proved to be a little trickier

Before tackling a CrossJoin of four dimensions, let's look at a the basic syntax of the function. Below is an example from Oracle's documentation of a simple CrossJoin (it is using members from the Year & Market dimensions in the Sample Basic outline):



Notice that all 4 possible combinations of the members are returned by the CrossJoin. This is straightforward enough. The part I had the most trouble with occurred when I started trying to include multiple CrossJoin's in the statements. All of the statements & brackets seemed to jumble together. I figured it would be most helpful to provide examples of what I ended up using so that you can see the syntax:

Note: Since these CrossJoin's were used to delete data, all of the members being selected are Level 0. I'm also using the ASOsamp Sample ASO outline to demonstrate the functionality.

**Simple CrossJoin:**

- `CrossJoin({[Original Price]},{[Curr Year]})`

### **Double CrossJoin:**

- `CrossJoin(CrossJoin({[Original Price]},{[Curr Year]}),{[Jan]})`

### **Triple CrossJoin:**

- `CrossJoin(CrossJoin(CrossJoin({[Original Price]},{[Curr Year]}),{[Jan]}),{[Sale]})`

Above demonstrates a sample syntax that will use `CrossJoin` to bring together four different dimensions. However, it is only grabbing one member from each dimension. To grab multiple members from a dimension, there are family functions that can be used (`.Parent`, `.Children`, `.Ancestor()`, `.Cousin()`, `.FirstChild`, `.LastChild`, `.FirstSibling`, and `.LastSibling`). For this example, I'm using the `.Children` function.

### **Triple CrossJoin Using Children Function:**

- `CrossJoin(CrossJoin(CrossJoin({[Measures].Children},{[Curr Year]}),{[Jan]}),{[Sale]})`

Notice how `[Original Price]` was replaced with `[Measures].Children`. Rather than returning one member, this will return the following children from the "Measures" dimension:



Utilize these family functions to increase the `CrossJoin` function's returned set.

One more thing to note. Substitution variables can be included within a `CrossJoin`. For example, let's say I created the subvar `&CurrMonth`. I can replace `Jan` in the code, thus making the month variable.

### **Triple CrossJoin Using Substitution Variable:**

- `CrossJoin(CrossJoin(CrossJoin({[Measures].Children},{[Curr Year]}),{[&CurrMonth]}),{[Sale]})`

### MaxL Syntax to Clear Data from a Region:

Next, let's cover the MaxL syntax to clear data in the region specified by the CrossJoin. From the Essbase technical reference 11.1.2.3 pg. 894:



The entire **CUBE-AREA** portion must be enclosed in single quotes i.e. '**CUBE-AREA**', which will define what portion of the database is going to be cleared.

And here are some more detailed notes on the syntax (also from the Essbase technical reference 11.1.2.3 pg. 897):



Based on the information laid out above, the line that I used in my MaxL statement turned out to look like this:

- `alter database AS0samp.Sample clear data in region 'CrossJoin(CrossJoin(CrossJoin({[Measures].children},{[Curr Year]}),{[&CurrMonth]}),{[Sale]})' physical;`

Prior to reloading data, I use this command line to clear the database in this region. This is to make sure that there is no stray data leftover in the cube that might cause discrepancies later on.

Feel free to leave any tips/advice on a more efficient method of utilizing CrossJoin!