PowerExcel Functions Manual

PowerExcel

This page is blank intentionally.

POWEREXCEL

FUNCTIONS MANUAL

T o p i c s

- The PowerExcel Functions

Descriptions, Syntax and Example

Table of Contents

This page is left blank intentionally.

PowerExcel Functions Manual

NOTE: Although repeatedly these functions are described reaching or otherwise working with an Olation database, **they work exactly the same with a PowerExcel database**, which, from a technical standpoint, is a type of Olation database.

1. OLAConnection

Function Description: This function will allow User 1 to send a spreadsheet to another user(s) so that User 2 (User3, et. al.) can, upon opening the spreadsheet, establish a connection (as defined with this function) to the specified Olation database..

Syntax: OLAConnection (Name, URL, Database, Windows Authentication

Name: Enter a name for a PowerExcel connection for User2—simple text entry.

URL: Enter the fully path of the URL required to reach the Olation Server where the database exists.

Database:: Enter the Database name that User2 will be enabled to reach.

Windows Authentication: Enter “1” or “True” if User2 will use Windows Authentication credentials to reach the Olation database; if “2” or “False” is entered, User 2 will be required to provide Username and Password information..

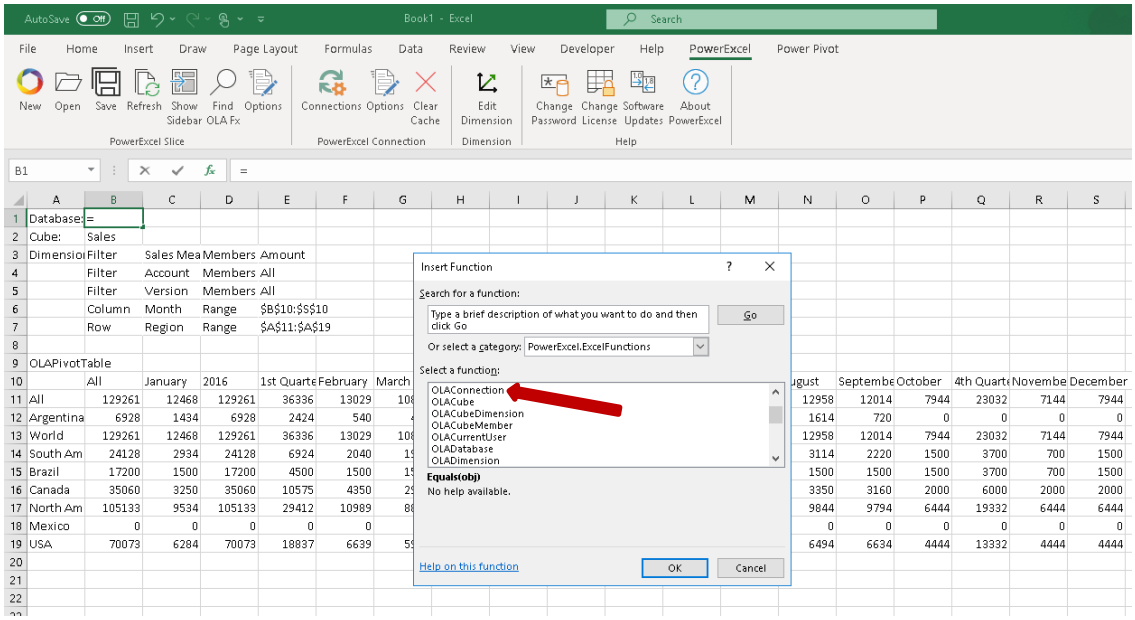
Remarks:

- The Olation Web Service must be running
- The Database must be opened and running in the specified server as identified by the URL.
- The ‘’Windows Authentication” parameter is compulsory.

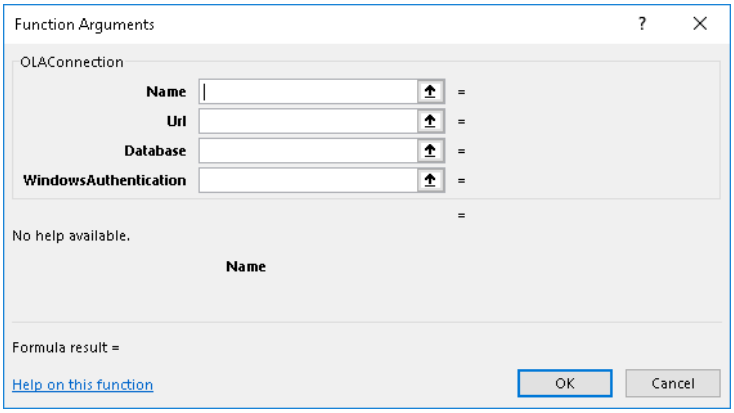
.

Example:

- User1 deletes the OLADatabase function in his or her PowerExcel Slice that establishes a valid connection to an Olation/PowerExcel database (in the following image, Cell B1)
- In the Excel formula bar, click on the Insert Function symbol (***fx***). The Insert Function window will appear.
- In the **Or select a category drop-down**, select **PowerExcel.ExcelFunctions**.
- Select **OLAConnection** (as shown in the following image). Click **OK**.



- Click **OK**. The Function Arguments window for **OLAConnection** appears, as shown in the following image.



- For **Name**, you can enter any text that you deem appropriate; for **URL**, enter the full Url path to the Olation database; enter the Database that the next user(s) will be enabled to reach; lastly, enter “1” or “True” to allow the recipient user to reach the Olation database via Windows Authentication credentials.
- Upon receiving the spreadsheet with the information filled in above, User2 (User3…et. al.) will be able to open the spreadsheet and see the same PowerExcel Slice used by User1.
- With the connection established, User2 (et. al.) will be able to create new PowerExcel Slices going forward.

2. OLACube

Function Description: This function will establish connection to and return the name of the source or target Cube by taking the parameters: (a) PowerExcel/Database Connection name or the cell reference that indicates the PowerExcel/Database Connection name; and the (b) Cube name or the cell reference that indicates the Cube name you want to connect to OR the index number corresponding to the Cube you want to return.

Syntax: OLACube(Connection,Cube)

OR

OLACube(Connection,CubeIndex)

Connection: Enter the PowerExcel connection that contains the information about the Olation server URL and the source database name.

Cube: Enter the name of the source/target Cube; or enter the cell reference that contains the name of the Cube that you wish to establish connection to.

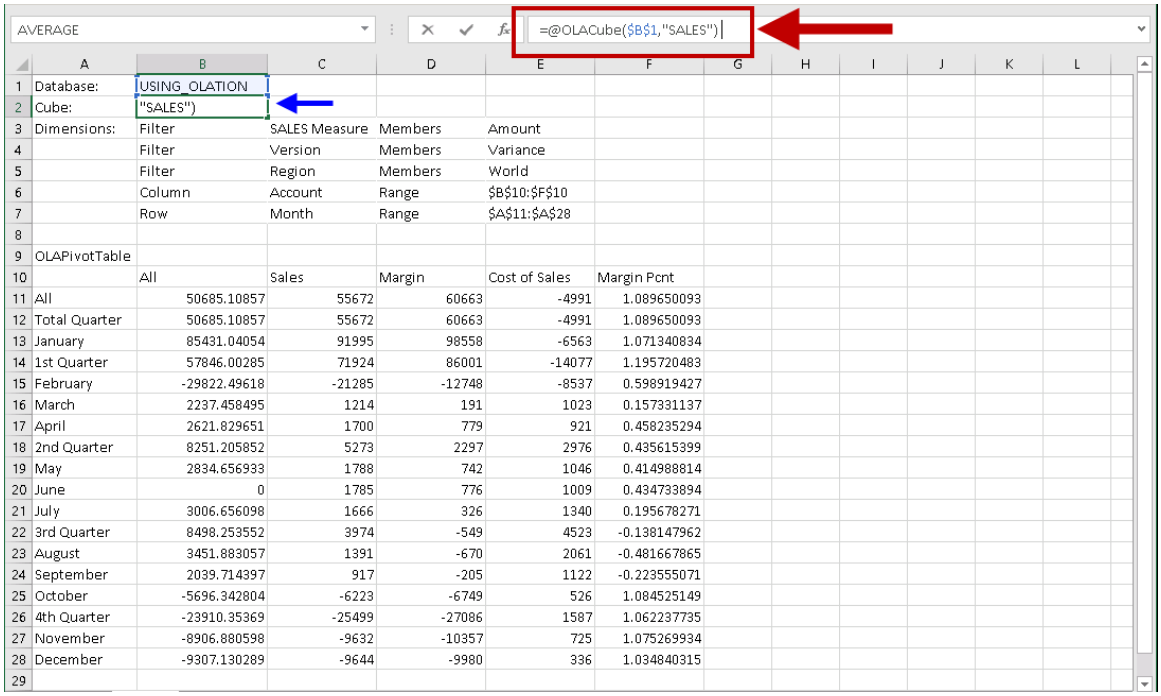
Cube Index: The index number corresponding to the Cube you want to return.

Remarks:

- The PowerExcel Connection must exist.
- The Olation Web Service must be running.
- The Database must be opened and running in the specified server.
- The 'Connection' and 'Cube' parameters are compulsory.
- Each Cube within the database is assigned an index number starting from 1, 2, 3… and so on. If there is no Cube assigned to that index number, then the function will return a #NAME? error.

It is worth noting that the **OLACube** function exists in all standard Slices. The example Slice below shows a PowerOLAP Pivot Table. When you click on the cell containing the **OLACube** formula, cell **B2**, the Excel formula bar shows the **OLACube** formula and its parameters.

- By clicking in the formula bar area (as can be seen in the screenshot, the mouse cursor is placed at the end of the formula), it will show the cell references corresponding to the **OLACube** function; in this example, the fact data is coming from the Cube called **"SALES"**, and it is using the PowerExcel connection/OLADatabase connection called "*USING_OLATION*"(**$B$1**).



	A	B	C	D	E	F
1	Database:	USING_OLATION				
2	Cube:	"SALES")				
3	Dimensions:	Filter	SALES Measure	Members	Amount	
4		Filter	Version	Members	Variance	
5		Filter	Region	Members	World	
6		Column	Account	Range	$B$10:$F$10	
7		Row	Month	Range	$A$11:$A$28	
8						
9	OLAPivotTable					
10		All	Sales	Margin	Cost of Sales	Margin Pcnt
11	All	50685.10857	55672	60663	-4991	1.089650093
12	Total Quarter	50685.10857	55672	60663	-4991	1.089650093
13	January	85431.04054	91995	98558	-6563	1.071340834
14	1st Quarter	57846.00285	71924	86001	-14077	1.195720483
15	February	-29822.49618	-21285	-12748	-8537	0.598919427
16	March	2237.458495	1214	191	1023	0.157331137
17	April	2621.829651	1700	779	921	0.458235294
18	2nd Quarter	8251.205852	5273	2297	2976	0.435615399
19	May	2834.656933	1788	742	1046	0.414988814
20	June	0	1785	776	1009	0.434733894
21	July	3006.656098	1666	326	1340	0.195678271
22	3rd Quarter	8498.253552	3974	-549	4523	-0.138147962
23	August	3451.883057	1391	-670	2061	-0.481667865
24	September	2039.714397	917	-205	1122	-0.223555071
25	October	-5696.342804	-6223	-6749	526	1.084525149
26	4th Quarter	-23910.35369	-25499	-27086	1587	1.062237735
27	November	-8906.880598	-9632	-10357	725	1.075269934
28	December	-9307.130289	-9644	-9980	336	1.034840315
29						

Cell References:

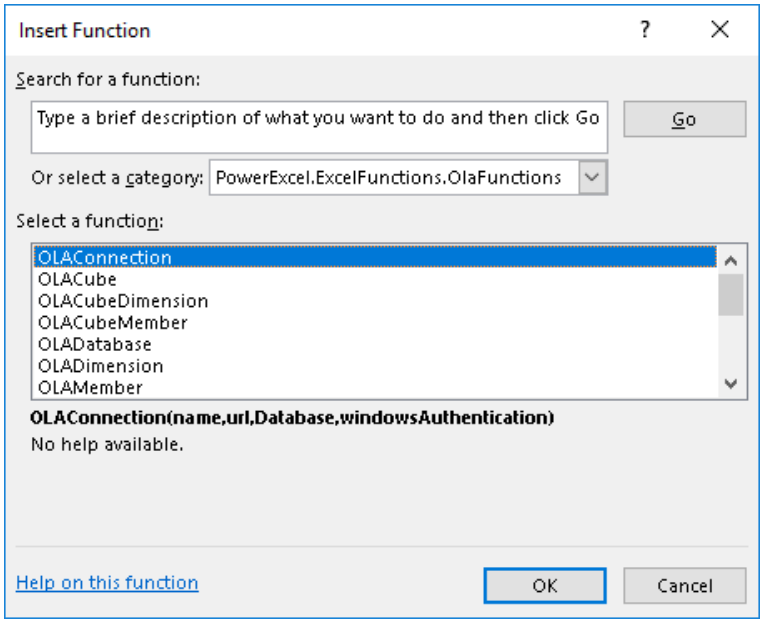
```
=OLACube($B$1,"SALES")
```

- $B$1– the Database name or the PowerExcel connection name/cell reference that contains the name of the Database or the PowerExcel connection, i.e., **USING_OLATION**
- "SALES"– the Cube in the Database

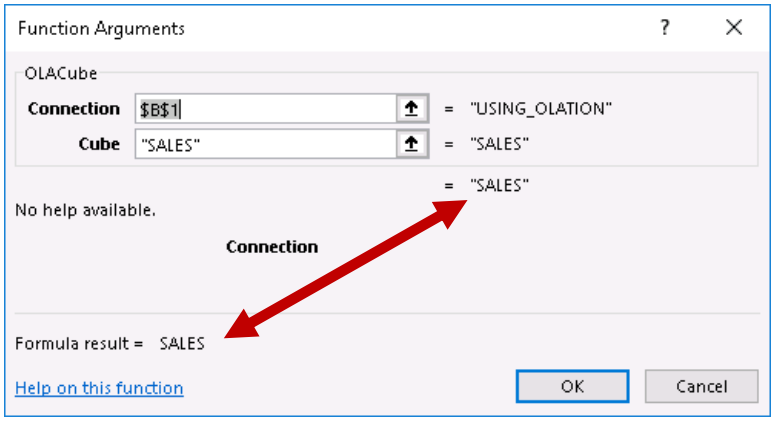
We will next provide examples that show the use of this function. The first example demonstrates use of the OLACube in an empty cell, i.e., outside the range of fact data returned by the cube.

Example 1: OLACube(Connection,Cube)

- Select a cell to the right of the field of data, e.g., Cell **H5**.
- In the Excel formula bar, click on the Insert Function symbol (***fx***). The Insert Function window will appear.
- In the **Or select a category drop-down**, select **PowerExcel.ExcelFunctions**..
- Select **OLACube** (as shown in the following image). Click **OK**.

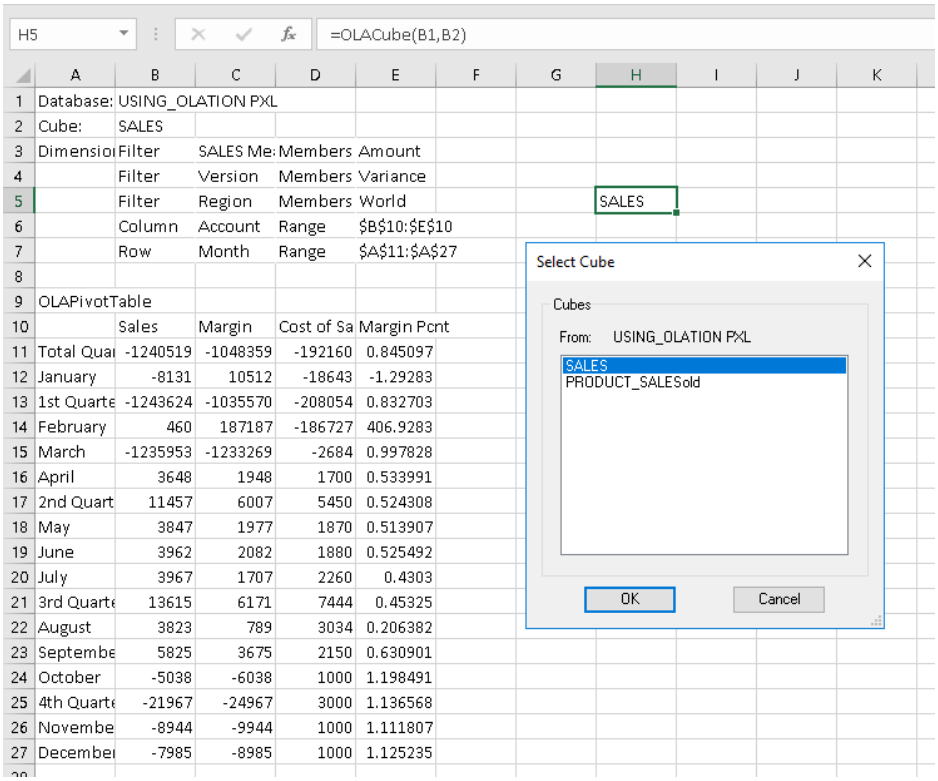


- Click **OK**. The Function Arguments window for **OLACube** appears.
- For Connection, you can reference Cell **B1** from the sample Slice (or type B1); for Cube, reference Cell **B2** (or type SALES), as shown below.



The return value of the formula function is, as expected (and quite obviously) **SALES**—as indicated above; also, upon clicking OK in this window, **SALES** will show in Cell **H5**.

That said, the use of this function can be demonstrated by, next, **double-clicking** on Cell **H5**. Note that a dialog pops up, **Select Cube**:

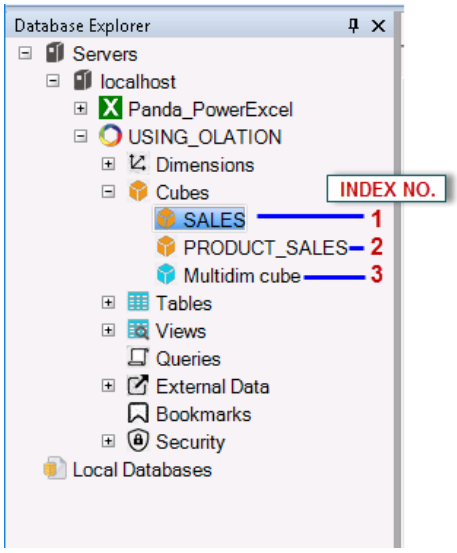


	A	B	C	D	E	F	G	H	I	J	K
1	Database:	USING_OLATION PXL									
2	Cube:	SALES									
3	Dimensio	Filter	SALES Me:	Members	Amount						
4		Filter	Version	Members	Variance						
5		Filter	Region	Members	World			SALES			
6		Column	Account	Range	$B$10:$E$10						
7		Row	Month	Range	$A$11:$A$27						
8											
9	OLAPivotTable										
10		Sales	Margin	Cost of Sa	Margin Pcnt						
11	Total Qua	-1240519	-1048359	-192160	0.845097						
12	January	-8131	10512	-18643	-1.29283						
13	1st Quarte	-1243624	-1035570	-208054	0.832703						
14	February	460	187187	-186727	406.9283						
15	March	-1235953	-1233269	-2684	0.997828						
16	April	3648	1948	1700	0.533991						
17	2nd Quart	11457	6007	5450	0.524308						
18	May	3847	1977	1870	0.513907						
19	June	3962	2082	1880	0.525492						
20	July	3967	1707	2260	0.4303						
21	3rd Quarte	13615	6171	7444	0.45325						
22	August	3823	789	3034	0.206382						
23	Septembe	5825	3675	2150	0.630901						
24	October	-5038	-6038	1000	1.198491						
25	4th Quarte	-21967	-24967	3000	1.136568						
26	Novembe	-8944	-9944	1000	1.111807						
27	Decembe	-7985	-8985	1000	1.125235						

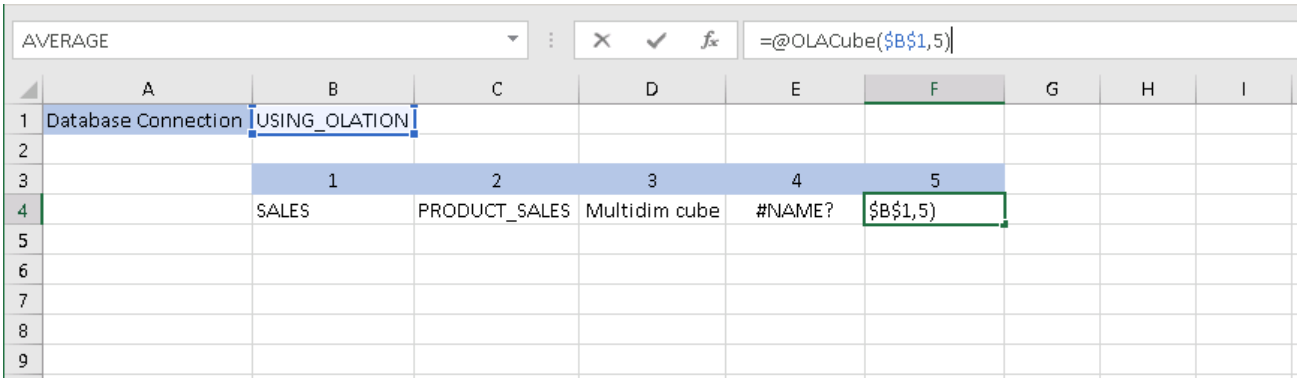
Note here that a Select Cube dialog appears—which can be useful for many reasons, including a visual indication of what cubes are available in the PowerExcel database.

Example 2: OLACube(Connection,CubeIndex)

For this example, use the **OLACube** function to identify all the Cubes that exist within a specific Database and in what order these Cubes are arranged. The index number assigned to each Cube is based on the order they are created into the Database (i.e., the first Cube created will be assigned the index number 1, the second Cube created will be assigned the index number 2, … and so on).



- First establish a connection to the target database: in Cell A1 type in **Database connection** (descriptive—i.e., non-formula-derived—cells are blue-highlighted to easily identify them), then go to cell B1 and use the OLADatabase function to establish a connection to the target database. In the example, we are using the Database connection: =@OLADatabase("USING_OLATION").
- In cells **B3 to F3**, type the numbers **1** to **5** (cells are highlighted in blue per above).
- Now, to use the **OLACube** function to determine the Cubes in the USING_OLATION database: in cell **B4**, type in **=@OLACube($B$1,1)**, with **$B$1** referencing the cell containing the Database connection and **1** corresponding to the index number. Press **Enter**. This will return the first Cube in the Database, i.e., *SALES*.
  Note: You can also use the Function button found beside the formula bar to define your **OLACube** formula.
- **Copy this formula** across cells **C4 to F4**, but change the index numbers with **2**, **3**, **4**… and so on. Once you hit Enter each time, you will see another Cube listed. If there are no more Cube assigned for an index number the function will return a #NAME? error. In this example (see next image), wthere are only 3 existing Cubes within the 'USING_OLATION' database, i.e., *SALES*, *PRODUCT_SALES* and *Multidim cube*, thus only indexes 1, 2 and 3 have corresponding cubes. Using index number 4 and 5 will return the #NAME? error.



3. OLACubeDimension

Function Description: This function returns the nth/indexed Dimension name of the specified Cube that exists within a specified PowerExcel Connection/Database Connection by taking the parameters: (a) PowerExcel/Database Connection name or the cell reference that indicates the PowerExcel/Database Connection name; and the (b) Cube name or cell reference that indicates the Cube name and (c) Dimension name or the cell reference that indicates the Dimension name you want to connect to /Index number corresponding to the Dimension you want to return.

Syntax: OLACubeDimension(Connection,Cube,Dimension)

OR

OLACubeDimension(Connection,Cube,DimensionIndex)

Connection: Enter the PowerExcel connection that contains the information about the Olation server URL and the source database name.

Cube: Enter the name of the source/target Cube; or enter the cell reference that contains the name of the source or target Cube you wish to establish connection to.

Dimension Name: Enter the Dimension name or the cell reference that contains the name of the Dimension that exists within the specified Database above.

Dimension Index: Enter the index number corresponding to the Dimension within the specified Cube that you want to return.

Remarks:

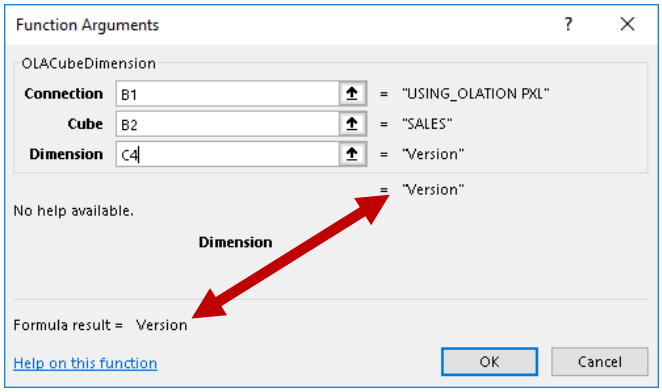
- The PowerExcel Connection must exist.
- The Olation Web Service must be running
- The Database must be opened and running in the specified server.
- The 'Connection' and 'Cube' parameters are compulsory.
- The last parameter can either be the 'Dimension name' or the 'Dimension Index' value
- Each Dimension within the Cube within the specified Database is assigned an index number starting from 1, 2, 3… and so on. If there is no Dimension assigned to that index number for the given Cube, then it will return a #NAME? error.
- The index number assigned to each Dimension is based on their order in the specified Cube.

Example 1:

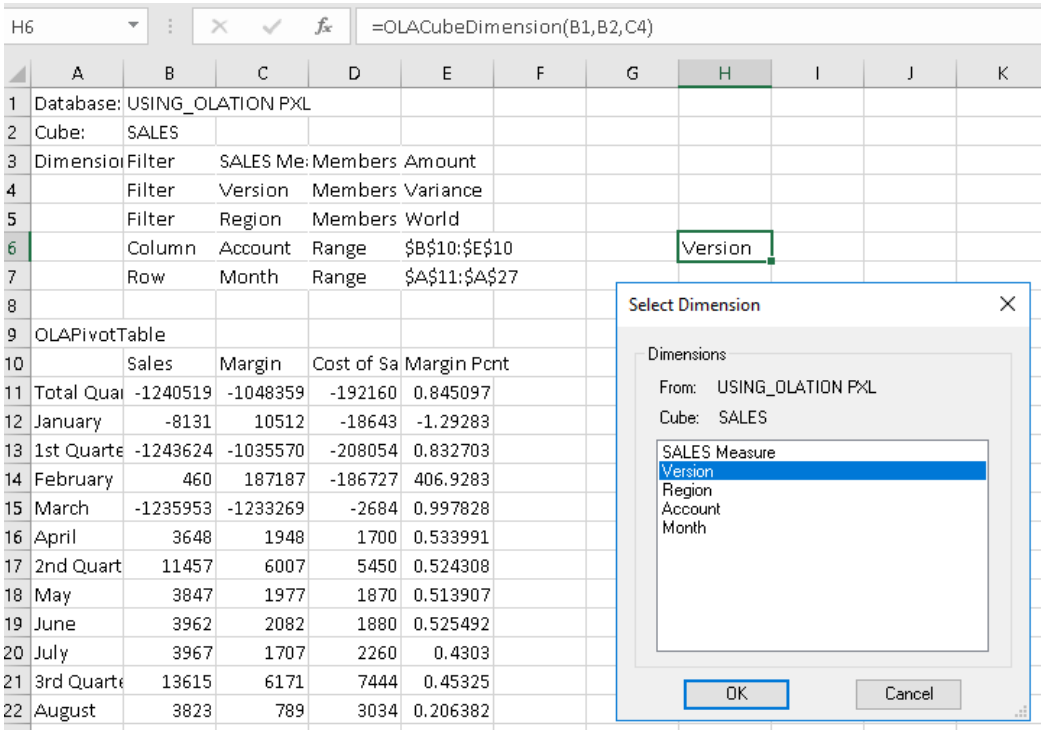
This first example, like the one for **OLACube**, will show how to make the function return a selection window for any of the Dimensions in the Cube

- Using an existing Slice, select a cell to the right of the field of data, e.g., Cell **H5**.
- In the Excel formula bar, click on the Insert Function symbol (***f*x**). The Insert Function window will appear.

- In the **Or select a category drop-down**, select **PowerExcel.ExcelFunctions.**
- Select **OLACubeDimension** Click **OK**.
- For Connection, you can reference Cell **B1** from the sample Slice (or type B1); for Cube, reference Cell **B2** (or type SALES), and then reference a Dimension name (or type it in) as shown below.



- Note that the Formula result is, as expected, “Version” (as indicated by the arrows, above)
- Click **OK**: the result (Version) will appear in the selected cell (e.g., H5).
- Next, **double-click** on that Cell. The Select Dimension window appears:

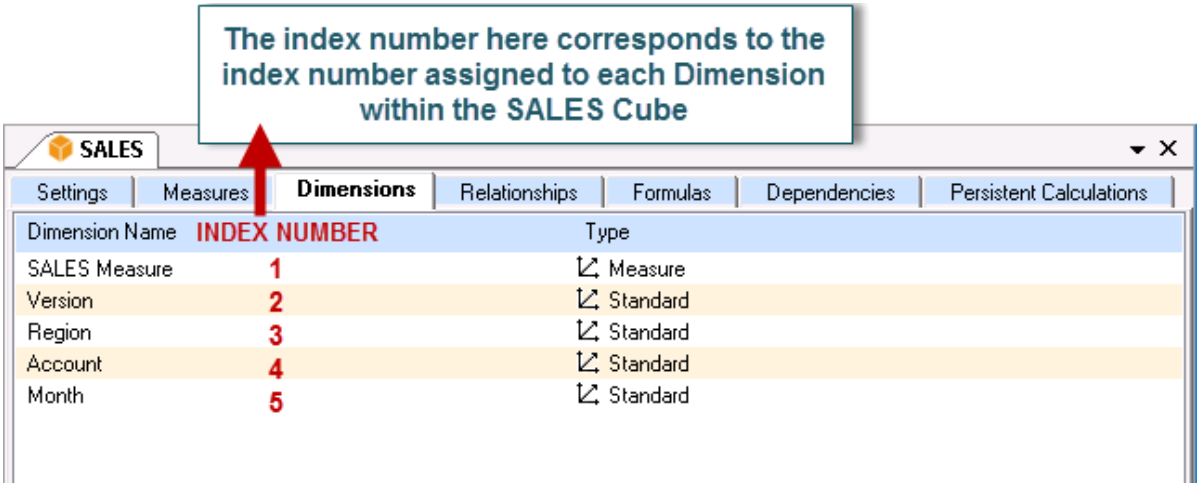


	A	B	C	D	E	F	G	H	I	J	K
1	Database:	USING_OLATION PXL									
2	Cube:	SALES									
3	Dimensio	Filter	SALES Me	Members	Amount						
4		Filter	Version	Members	Variance						
5		Filter	Region	Members	World						
6		Column	Account	Range	$B$10:$E$10						
7		Row	Month	Range	$A$11:$A$27						
8											
9	OLAPivotTable										
10		Sales	Margin	Cost of Sa	Margin Pcnt						
11	Total Quai	-1240519	-1048359	-192160	0.845097						
12	January	-8131	10512	-18643	-1.29283						
13	1st Quarte	-1243624	-1035570	-208054	0.832703						
14	February	460	187187	-186727	406.9283						
15	March	-1235953	-1233269	-2684	0.997828						
16	April	3648	1948	1700	0.533991						
17	2nd Quart	11457	6007	5450	0.524308						
18	May	3847	1977	1870	0.513907						
19	June	3962	2082	1880	0.525492						
20	July	3967	1707	2260	0.4303						
21	3rd Quarte	13615	6171	7444	0.45325						
22	August	3823	789	3034	0.206382						

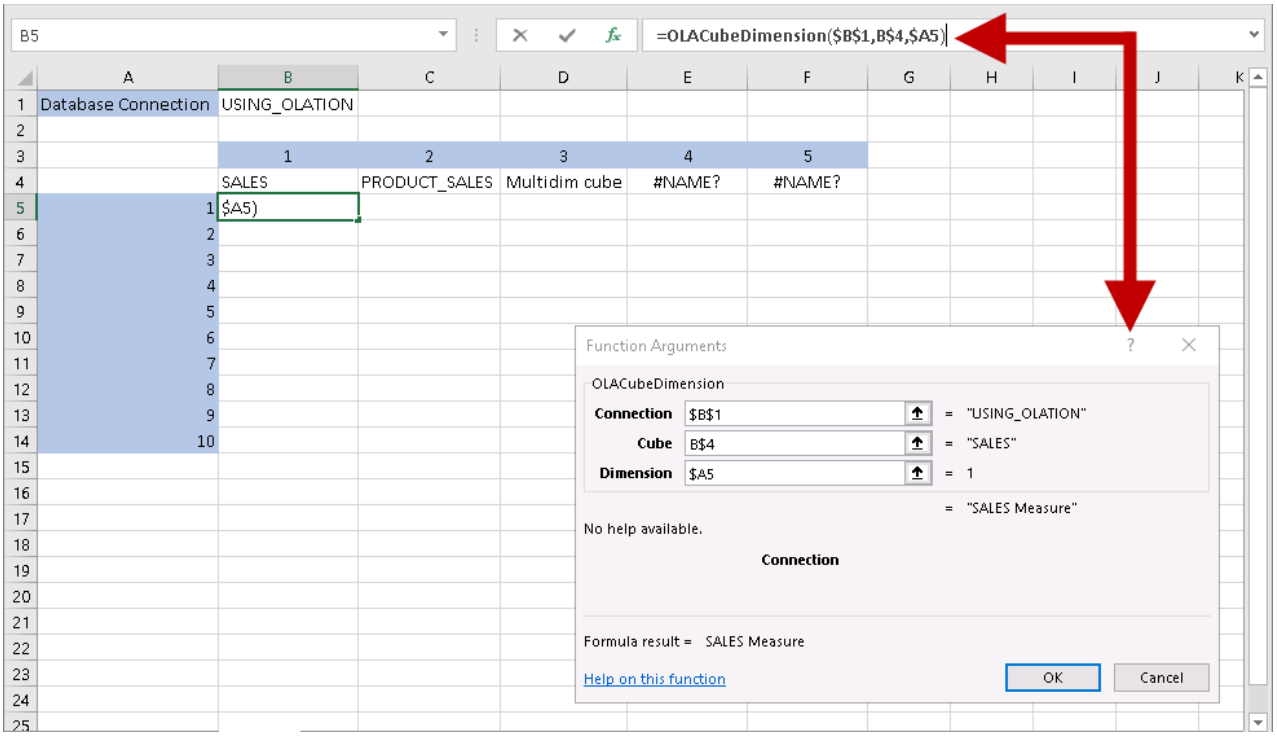
Example 2:

For this example, use the **OLACubeDimension** function to identify all the Dimensions that exist per Cube in the *USING_OLATION* database based on their index number. (Note that DimensionIndex will be the last argument in the function.)

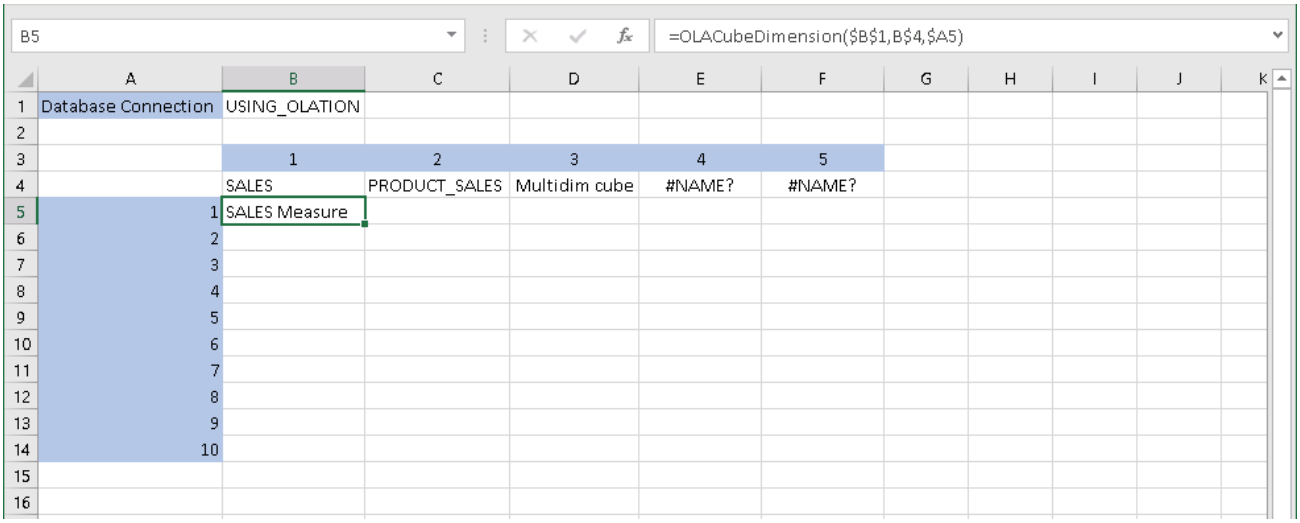
A sample screenshot below shows the *SALES* Cube and its component Dimensions. This screenshot indicates also how index numbers are assigned per Dimension based on how they are ordered within the Cube.



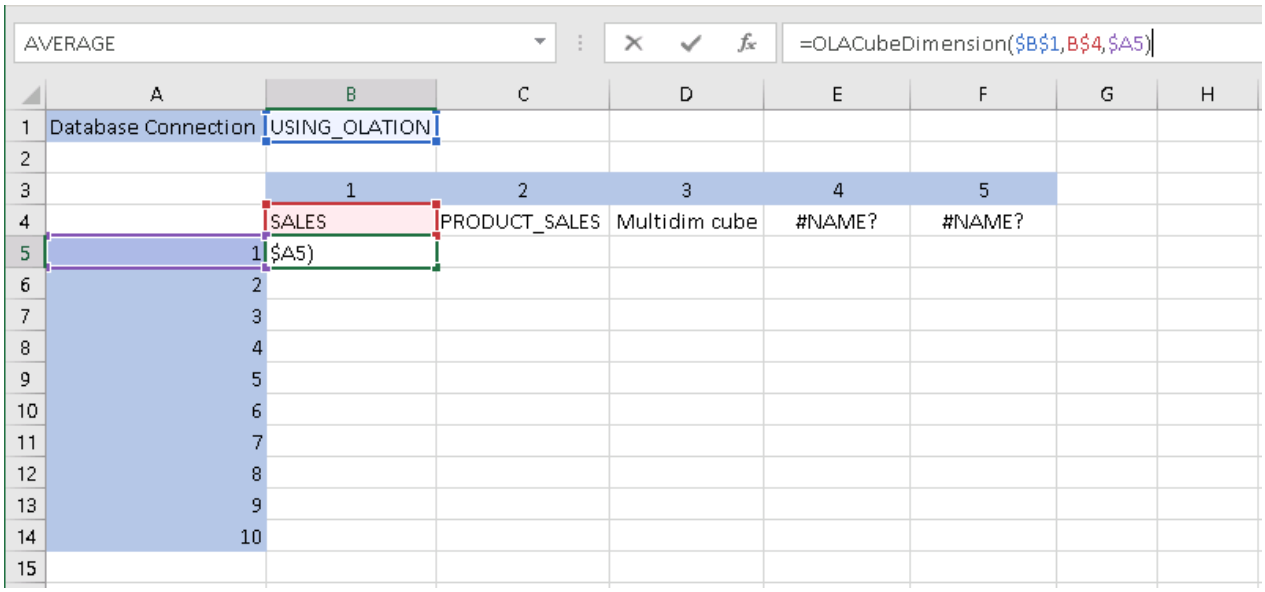
- First establish a connection to the target database. In cell A1 type in **Database connection** (in the next image, cells that are descriptive—i.e., non-formula-derived—are blue-highlighted for easy identification), then go to cell B1: use the OLADatabase function to establish a connection to the target database. In the example, we are using the Database connection: =@OLADatabase("USING_OLATION").
- In cells **B3 to F3**, type the numbers **1 to 5** (blue-highlighted, per above). Use the OLACube function to in Cells B4 to F5 to return the Cube names. Or you can just type in the Cube name/s, making sure that they are spelled correctly.
- In cells A5 to A14 type the numbers 1 to 10 and blue-highlight them (as they are descriptive, per above).
- Use the **OLACubeDimension** function to return the complete list of Dimension for each Cube: Click on cell **B5;** then click the **Function button**. The Insert Function dialog appears. In the Function Category drop-down menu, select **PowerExcel.ExcelFunctions** then select **OLACubeDimension** from the Function list and click **OK**. The Function Arguments dialog appears. This is where you will define the formula.

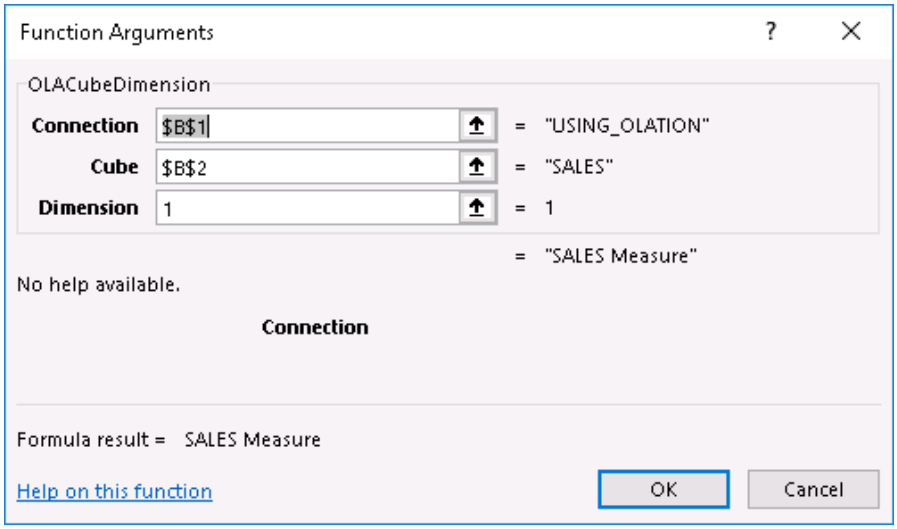


- In the Function Arguments dialog, click on the **Connection** field; then click on cell **B1**, which contains the Database connection reference. Notice that the Database connection name “USING_OLATION” appears beside the connection field.
  Note: Use an absolute reference (**$B$1**) so that the formulas can be copied across to other cells.
- Click on the **Cube** field, then click on cell **B4**, which contains the Cube reference. Note that the Cube name “SALES” appears beside the Cube field.
  Note: Use an absolute reference for this formula (**B$4**).
- Click on the Dimension field then type the index number (1). Or, an easier way would be to click on A5, making the column absolute (result is **$A5**).
- Click **OK**. The first Dimension is returned (i.e., **SALES Measure**), per the following image.



- To review the cell references:



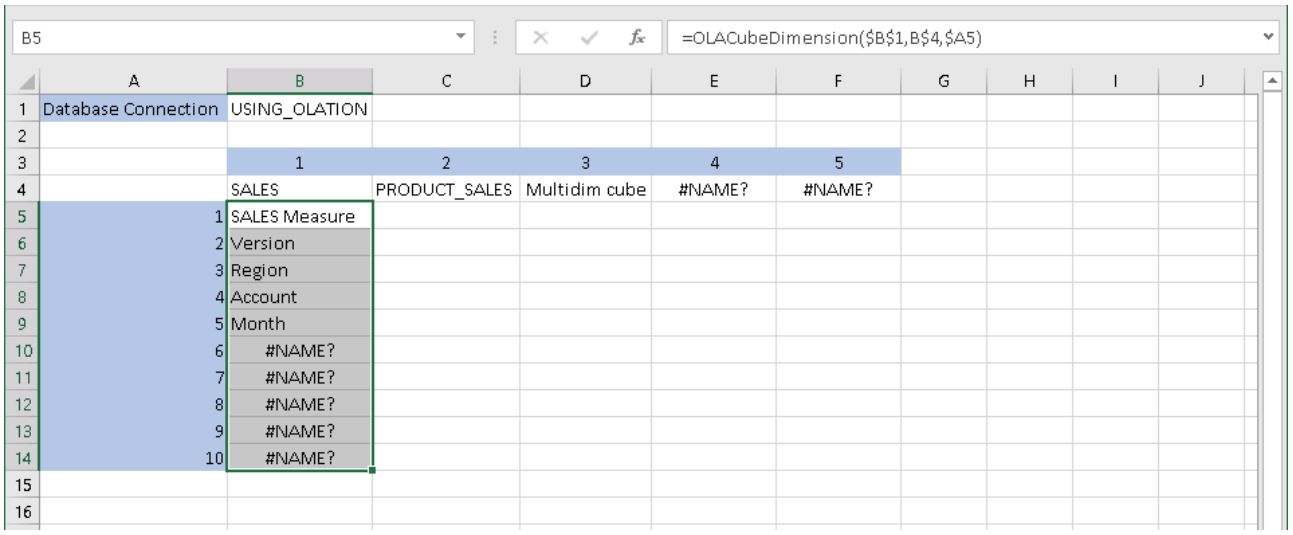


Cell References:

=OLACubeDimension($B$1,B$4,$A5)

- $B$1 – the Database name or the PowerExcel connection name/cell reference that contains the name of the Database or the PowerExcel connection, i.e., **USING_OLATION**
- B$4 – the Cube in the Database, i.e., **SALES**
- $A5 – the Dimension Index number, i.e., **1**

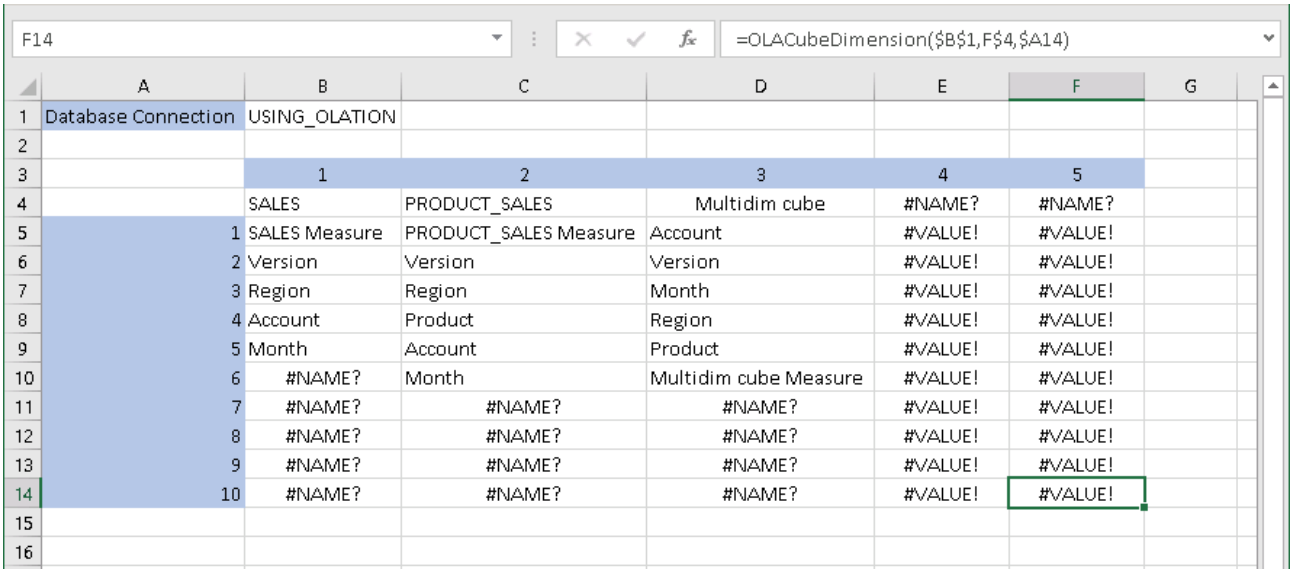
- Next, copy the formula to cells **B6 to B14** then click **F9** to refresh Excel. The **OLACubeDimension** function will return all the Dimensions that exist within the *SALES* Cube. If the Index value has no assigned Dimension, it will return a #NAME? error, as shown in the image below:



B5 =OLACubeDimension($B$1,B$4,$A5)

	A	B	C	D	E	F	G	H	I	J
1	Database Connection	USING_OLATION								
2										
3		1	2	3	4	5				
4		SALES	PRODUCT_SALES	Multidim cube	#NAME?	#NAME?				
5	1	SALES Measure								
6	2	Version								
7	3	Region								
8	4	Account								
9	5	Month								
10	6	#NAME?								
11	7	#NAME?								
12	8	#NAME?								
13	9	#NAME?								
14	10	#NAME?								
15										
16										

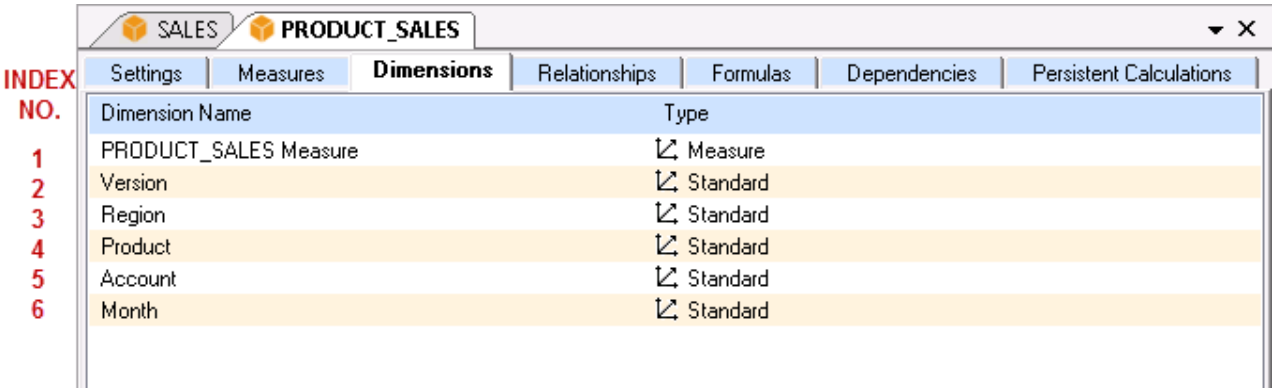
- Copy the formula across all other cells (**B5:F14**). Then click **Refresh**. The table is now updated.

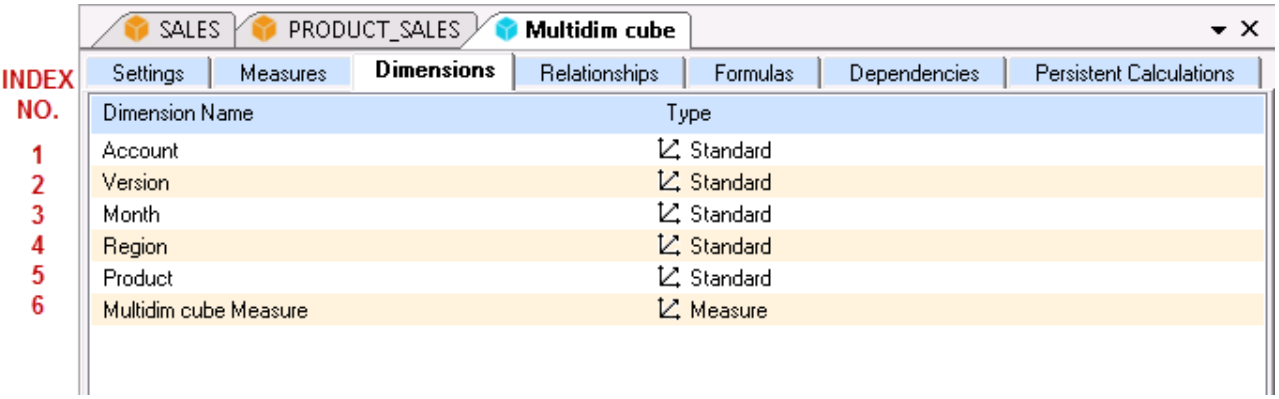


F14 =OLACubeDimension($B$1,F$4,$A14)

	A	B	C	D	E	F	G
1	Database Connection	USING_OLATION					
2							
3		1	2	3	4	5	
4		SALES	PRODUCT_SALES	Multidim cube	#NAME?	#NAME?	
5	1	SALES Measure	PRODUCT_SALES Measure	Account	#VALUE!	#VALUE!	
6	2	Version	Version	Version	#VALUE!	#VALUE!	
7	3	Region	Region	Month	#VALUE!	#VALUE!	
8	4	Account	Product	Region	#VALUE!	#VALUE!	
9	5	Month	Account	Product	#VALUE!	#VALUE!	
10	6	#NAME?	Month	Multidim cube Measure	#VALUE!	#VALUE!	
11	7	#NAME?	#NAME?	#NAME?	#VALUE!	#VALUE!	
12	8	#NAME?	#NAME?	#NAME?	#VALUE!	#VALUE!	
13	9	#NAME?	#NAME?	#NAME?	#VALUE!	#VALUE!	
14	10	#NAME?	#NAME?	#NAME?	#VALUE!	#VALUE!	
15							
16							

The following image shows in the Dimensions Name column the dimensions in the Multidem cube, which matches what is shown in Column 3 above.





4. OLACubeMember

Function Description: This function returns the nth/indexed Member for a specified Dimension that exists within a specified Cube. This function takes the parameters: (a)PowerExcel/Database Connection name or the cell reference that indicates the PowerExcel/Database Connection name; and the (b)Cube name or cell reference that indicates the Cube name; (c)Dimension name or the cell reference that indicates the Dimension name ; and the (d)Member name or the cell reference that indicates the Member you want to connect to /Index number corresponding to the Member you want to return.

Syntax: OLACubeMember(Connection,Cube,Dimension,Member)

OR

OLACubeMemberIndex(Connection,Cube,Dimension,MemberIndex)

Connection: Enter the PowerExcel connection which contains the information about the Olation server URL and the source database name.

Cube: Enter the name of the source/target Cube; or enter the cell reference that contains the name of the source or target Cube you wish to establish connection to.

Dimension Name: Enter the Dimension name or the cell reference that contains the name of the Dimension that exists within the specified Database above.

Member Name: Enter the Member name or the cell reference that contains the name of the Member that exists within the specified Dimension.

Member Index: Enter the index number corresponding to the Member within the specified Dimension that you want to return.

Remarks:

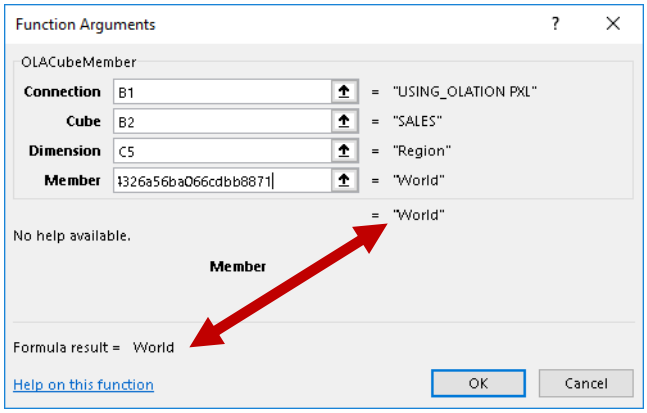
- The PowerExcel Connection must exist.
- The Olation Web Service must be running
- The Database must be opened and running in the specified server.
- The 'Connection', 'Cube' and 'Dimension' parameters are compulsory
- The last parameter can either be the 'Member name' or the 'Member Index' value
- All Members within the Dimension are each assigned an index number starting from 1, 2, 3… and so on. If there is no Member assigned to that index number for the given Dimension, then it will return a #NAME? error.
- The index number assigned to each Member is based on the order they are arranged in the specified Dimension.

Example 1:

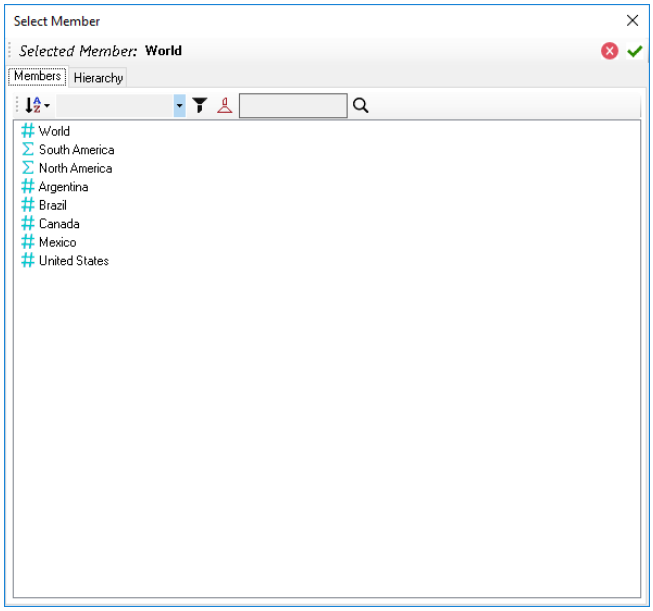
This first example, like the ones for **OLACube** and **OLACubeDimension**, will show how to make the function return a selection window for any Member of a Dimension in the Cube.

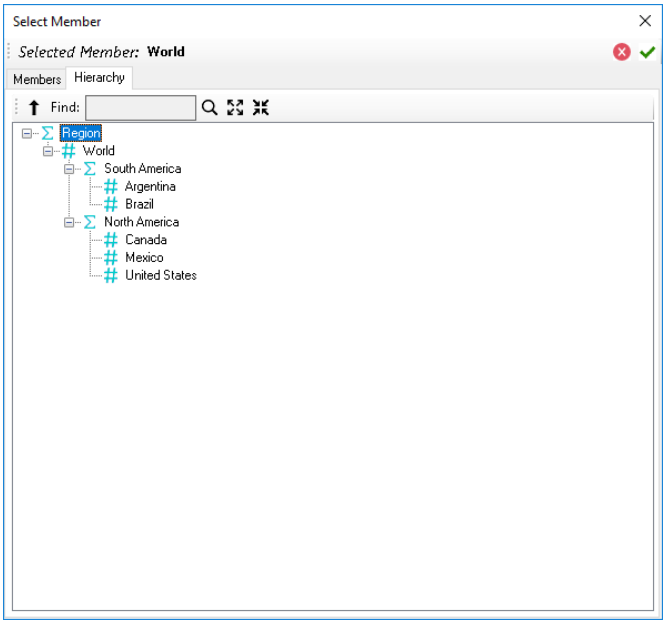
- Using an existing Slice, select a cell to the right of the field of data, e.g., Cell **H5**.

- In the Excel formula bar, click on the Insert Function symbol (***fx***). The Insert Function window will appear.
- In the **Or select a category drop-down**, select **PowerExcel.ExcelFunctions.**
- Select **OLACubeMember**. Click **OK**.
- For Connection, you can reference Cell **B1** from the sample Slice (or type B1); for Cube, reference Cell **B2** (or type SALES); next, reference a selected Dimension (e.g., ***Region***) and then reference a Member name (or type it in) as shown below. Note that if you reference a cell with a Member in it from the Filter area, a unique string will appear, as in the final argument below, which provides the result "World".



- Click **OK**: the result (World) will appear in the selected cell (e.g., H5).
- Next, **double-click** on that Cell. The Select Member window appears—note that it has two tabs (shown in two successive images below): one (Members) for selection of Members from a list, and another (Hierarchy) that shows the Members as they appear in a hierarchy within the Dimension:

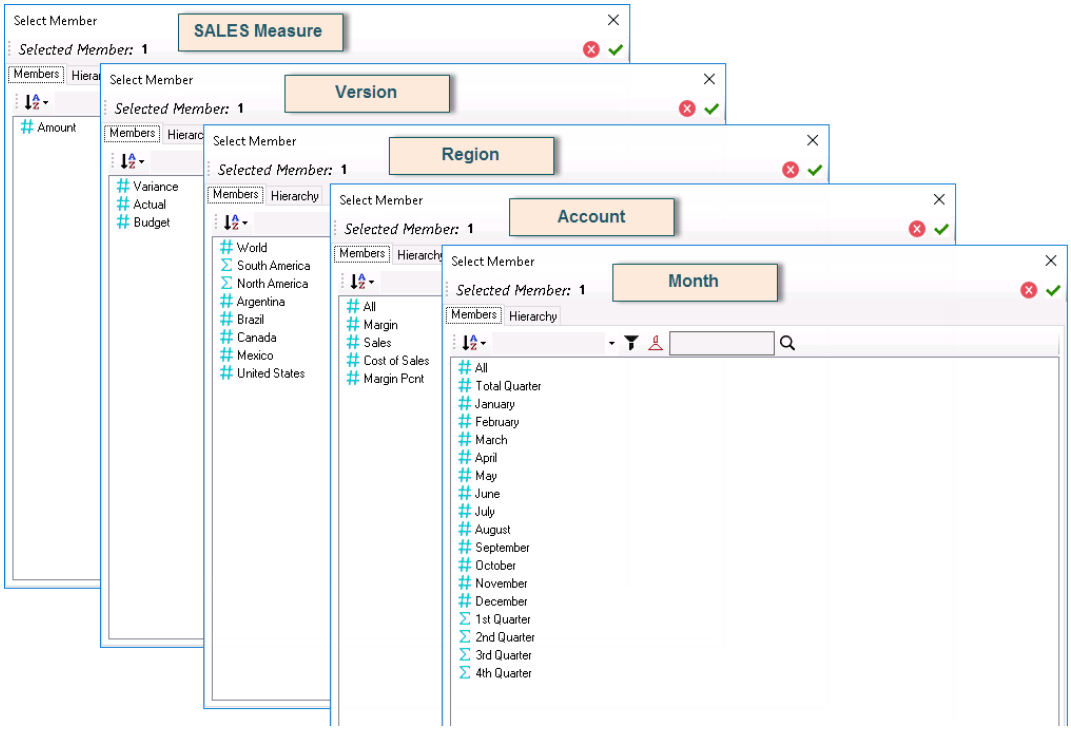




Example 2:

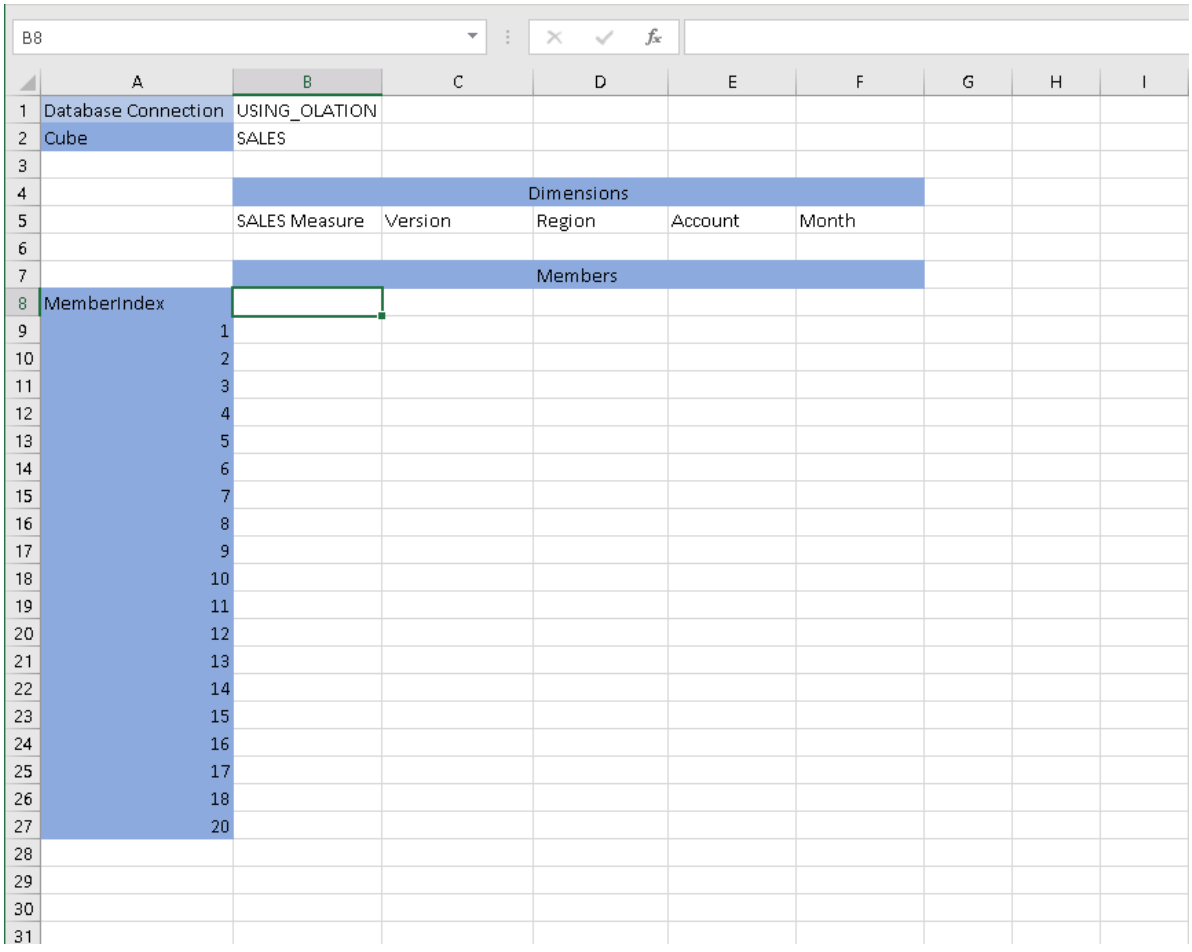
For this example, we will use the **OLACubeMember** function to identify and make a list of all the Members that exist for a component Dimension of a specific Cube and bring them down to Excel—all based on their index numbers. For this example, *USING_OLATION* is the source database and the focus is on the *SALES* Cube. Our focus will be on the *Months* Dimension.

A sample screenshot below shows the *SALES* Cube and its component Dimensions. This screenshot indicates also how index numbers are assigned per Dimension based on how they are ordered within the Cube.



A screenshot of the list of Members for each component Dimension of the SALES Cube

- First we will establish a connection to the target database. In cell A1 type in **Database connection** (cells that are descriptive—i.e., non-formula-derived—are blue-highlighted for easy identification), then in cell B1, use the OLADatabase function to establish a connection to the target database, in the example, we are using the Database connection: =@OLADatabase("USING_OLATION").
- In cell A2 type the caption **Cube** (again highlight this in blue since this is just a caption) then go to cell B2 then and use the OLACube function to return the source Cube (*SALES*). In the example, we defined the formula as: =@OLACube($B$1,"SALES").
- In cells **B5 to F5**, use the OLACubeDimension function to pull in the Dimensions that exist for the *SALES* Cube. In the example, we defined the formula in cell B5 as: =OLACubeDimension($B$1,$B$2,1).
  Copy the formula to cells **C5 to F5** and just change the last parameter (index value) with **2**, **3**… and so on.
- In cells **A9 to A28**, type the Member Index value 1 to 20. You can also add captions as in the following image.

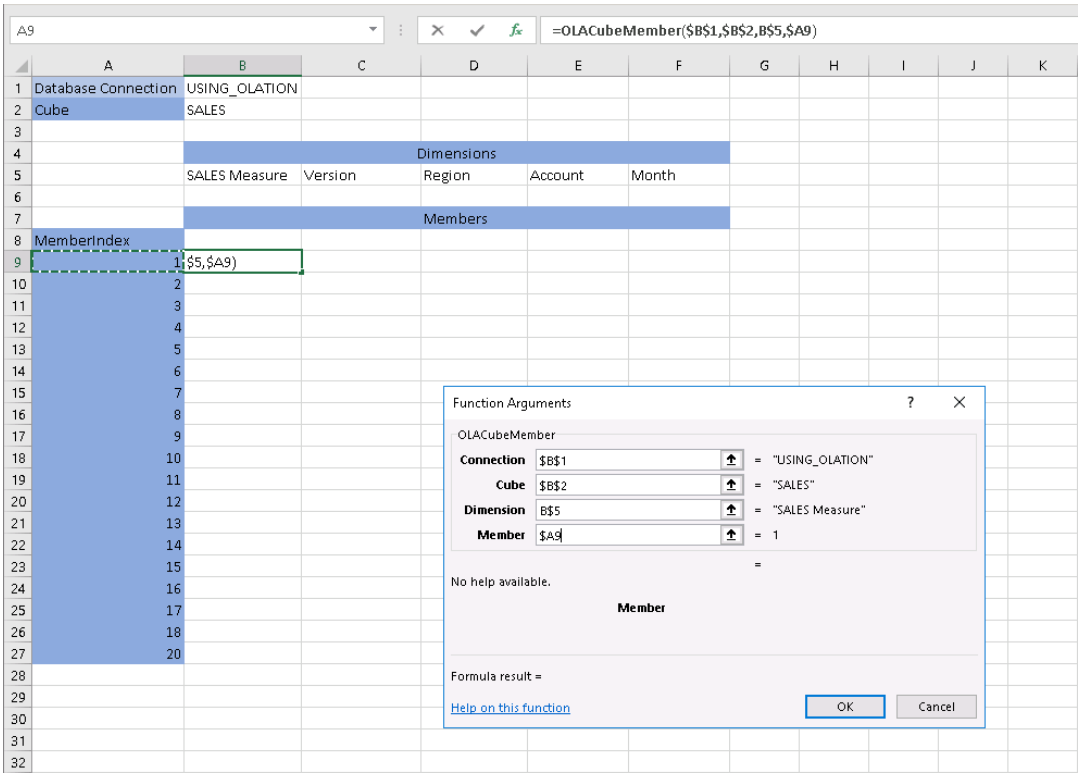


	A	B	C	D	E	F
1	Database Connection	USING_OLATION				
2	Cube	SALES				
3						
4		Dimensions				
5		SALES Measure	Version	Region	Account	Month
6						
7		Members				
8	MemberIndex					
9	1					
10	2					
11	3					
12	4					
13	5					
14	6					
15	7					
16	8					
17	9					
18	10					
19	11					
20	12					
21	13					
22	14					
23	15					
24	16					
25	17					
26	18					
27	20					
28						
29						
30						
31						

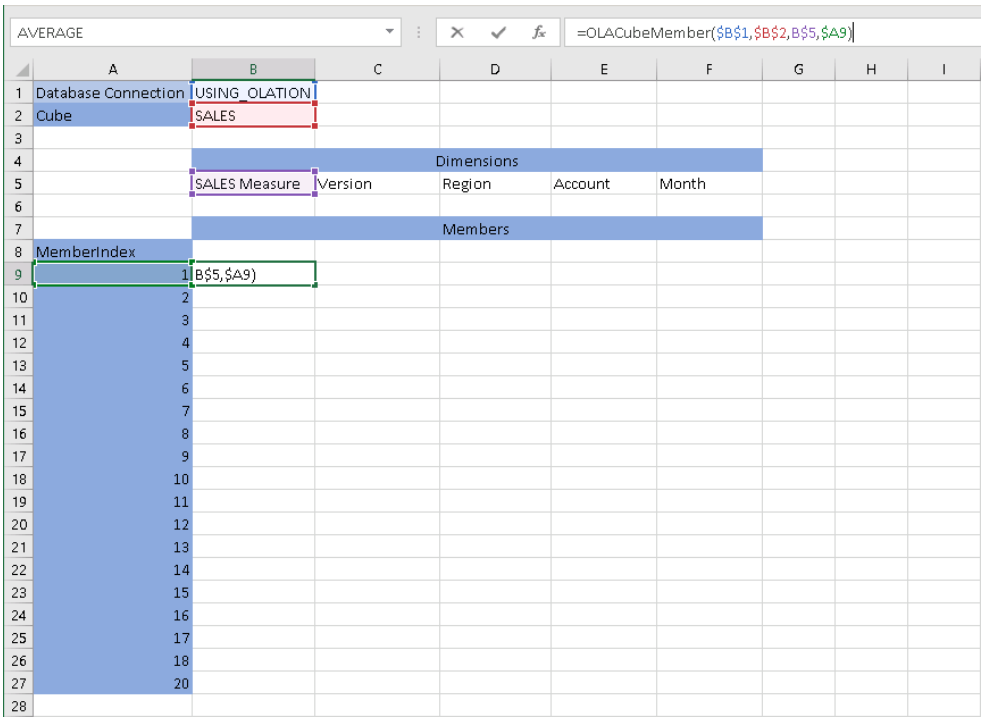
- Now, we are going to use the **OLACubeMember** function to create a list of Members for each Dimension that exist for the *SALES* Cube within the *USING_OLATION* database. Go

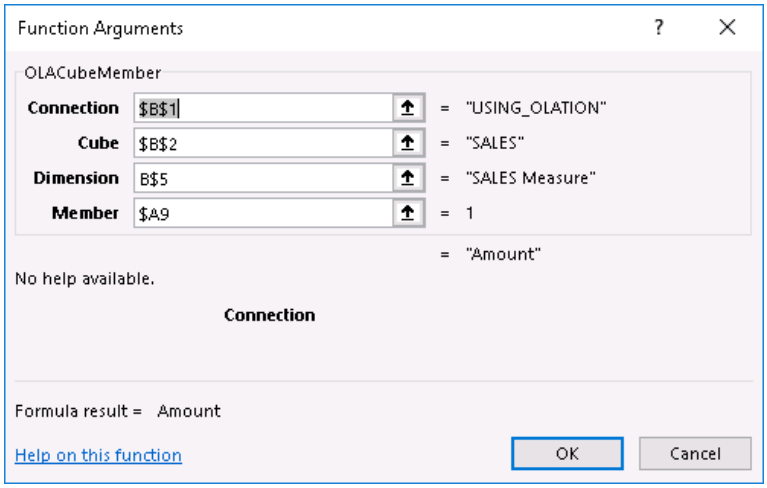
to cell **B9** then click next to the **Function** button located beside the formula bar. The Insert Function dialog box appears.

- In the category list, select **PowerExcel.ExcelFunctions**, click OLACubeMember and click OK.The Function Arguments dialog box appears. This is where you will define the **OLACubeMember** formula.
- In the Function Arguments dialog, click on the **Connection** field, then click on cell **B1** which contains the Database connection reference. Notice that the Database connection name “USING_OLATION” appears beside the connection field.
  Note: Use the absolute reference to easily copy the formulas across the other cells. We used absolute reference along rows and columns (result is **$B$1**)
- Click on the **Cube** field,m then click on the cell **B2** which is the cell that contains the Cube reference. Again, notice that the Cube name “SALES” appears beside the Cube field.
  Note: Again, use absolute reference for this formula. We used absolute reference along the rows and columns (result is **$B$2**).
- Click on the **Dimension** field then click on cell **B5** which is the cells that contains the Dimension reference. Again, notice that the Dimension name “SALES Measure” appears beside the Dimension field.
  Note: You can again use absolute reference for this formula. Let us use absolute reference along the rows (result is **B$5**)
- Lastly, click on the **Member** field then click on cell **A9** which will dictate the index value for our last parameter. Again, we will use the absolute reference along the columns (result is **$A9**).
- The Function Arguments dialog will look as follows:



- Click **OK**. This will return the Member **Amount**. Let us take a look at the cell references:



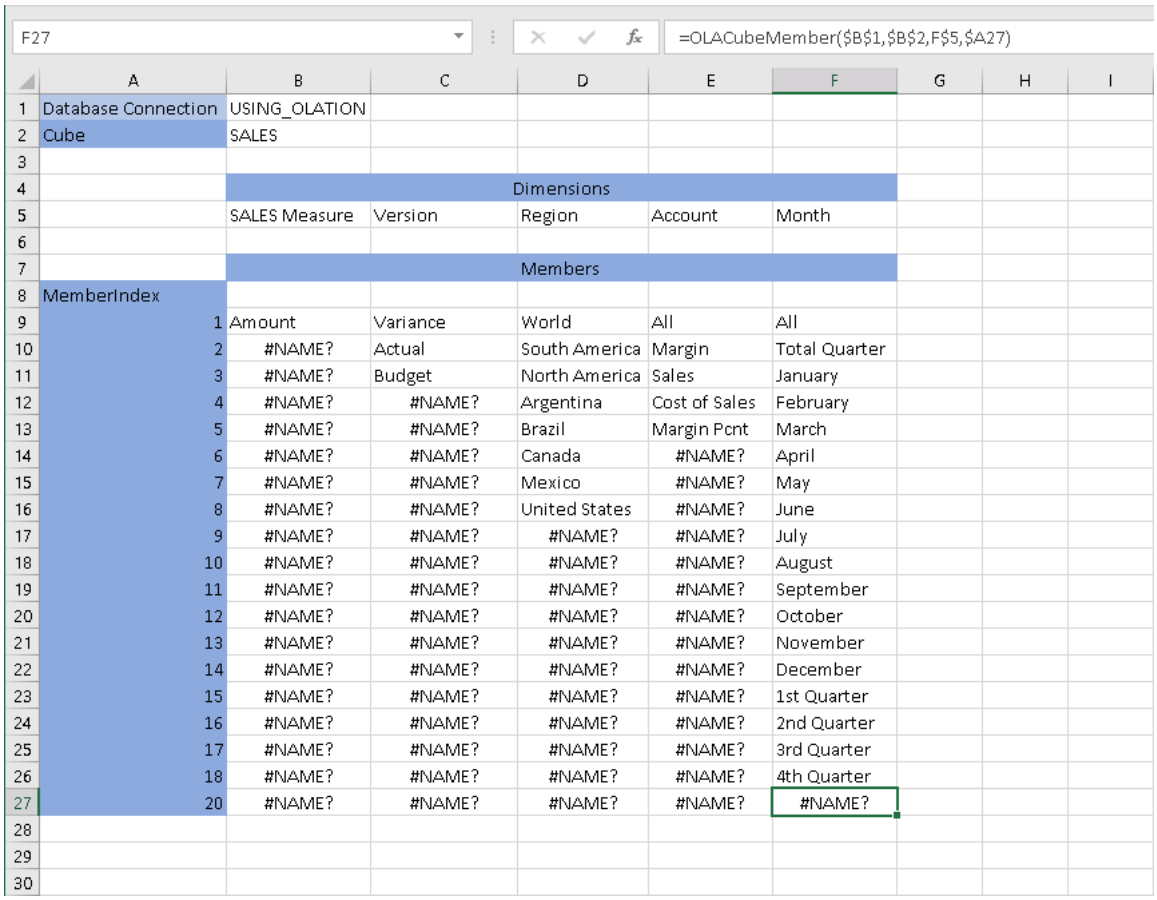


Cell References:

=OLACubeMember($B$1,$B$2,B$5,$A9)

- $B$1– the Database name or the PowerExcel connection name/cell reference that contains the name of the Database or the PowerExcel connection, i.e., **USING_OLATION**
- $B$2– the Cube in the Database, i.e., **SALES**
- B$5– the Dimension in the *SALES* Cube, i.e., **SALES Measure**
- $A9– the Member Index number or cell reference, i.e., **1**

- Now, copy this formula across the range **B9:F27**.



F27 =OLACubeMember($B$1,$B$2,F$5,$A27)

	A	B	C	D	E	F	G	H	I
1	Database Connection	USING_OLATION							
2	Cube	SALES							
3									
4		Dimensions							
5		SALES Measure	Version	Region	Account	Month			
6									
7		Members							
8	MemberIndex								
9	1	Amount	Variance	World	All	All			
10	2	#NAME?	Actual	South America	Margin	Total Quarter			
11	3	#NAME?	Budget	North America	Sales	January			
12	4	#NAME?	#NAME?	Argentina	Cost of Sales	February			
13	5	#NAME?	#NAME?	Brazil	Margin Pcnt	March			
14	6	#NAME?	#NAME?	Canada	#NAME?	April			
15	7	#NAME?	#NAME?	Mexico	#NAME?	May			
16	8	#NAME?	#NAME?	United States	#NAME?	June			
17	9	#NAME?	#NAME?	#NAME?	#NAME?	July			
18	10	#NAME?	#NAME?	#NAME?	#NAME?	August			
19	11	#NAME?	#NAME?	#NAME?	#NAME?	September			
20	12	#NAME?	#NAME?	#NAME?	#NAME?	October			
21	13	#NAME?	#NAME?	#NAME?	#NAME?	November			
22	14	#NAME?	#NAME?	#NAME?	#NAME?	December			
23	15	#NAME?	#NAME?	#NAME?	#NAME?	1st Quarter			
24	16	#NAME?	#NAME?	#NAME?	#NAME?	2nd Quarter			
25	17	#NAME?	#NAME?	#NAME?	#NAME?	3rd Quarter			
26	18	#NAME?	#NAME?	#NAME?	#NAME?	4th Quarter			
27	20	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?			
28									
29									
30									

- Press **F9** or refresh the Excel worksheet. You will see that it now returns a full list of the Members per Dimension that exist on the USING_OLATION database.

5. CurrentUser

Function Description: This function will return the name of the current user logged into the machine and accessing the PowerExcel application, taking the PowerExcel/Database Connection name or the cell reference that indicates the PowerExcel/Database Connection name as the only parameter.

Syntax: OLACurrentUser(Connection)

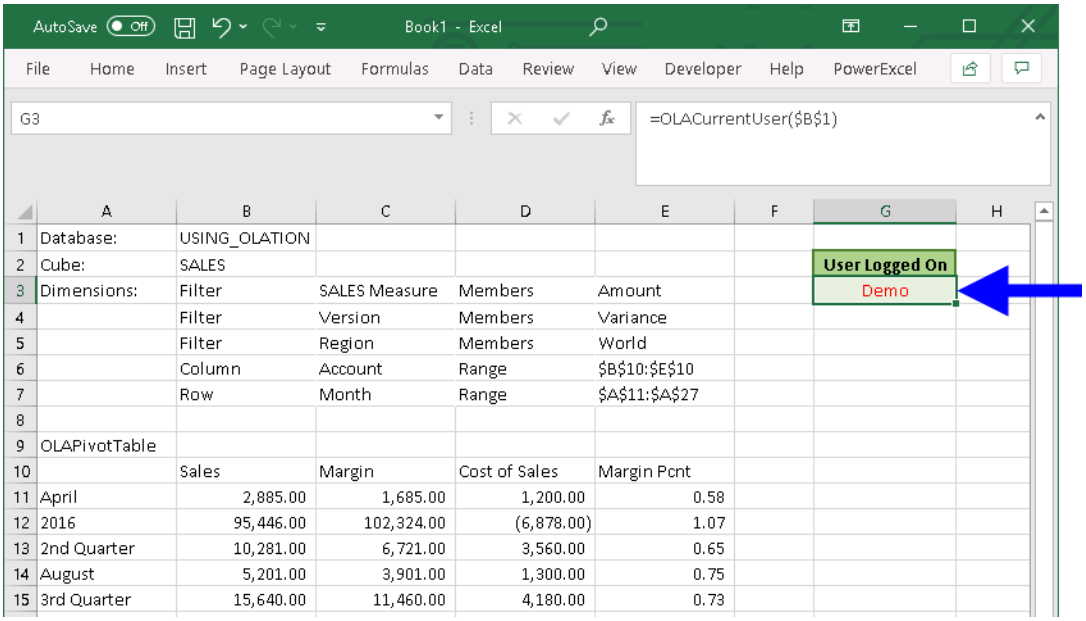
Connection: Enter the PowerExcel connection that contains the information about the Olation server URL and the source database name.

Remarks:

- The PowerExcel Connection must exist.
- The Olation Web Service must be running.
- The Database must be opened and running in the specified server.
- The 'Connection' parameter is compulsory.

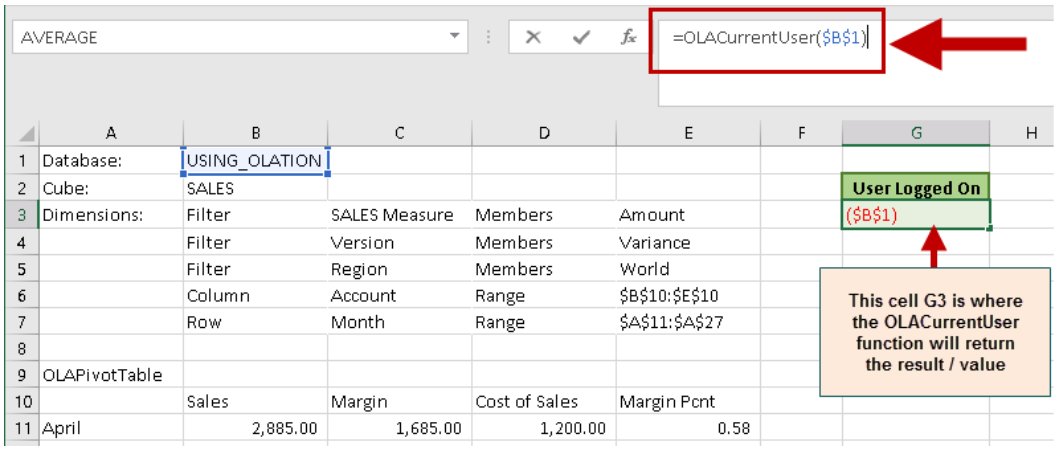
Example:

For this example, we will use the **OLACurrentUser** function to return the name of the user account currently logged on the machine where the PowerExcel Slice, e.g., *SALES Report*, is currently opened. Looking at the screenshot below, the active cell is cell **G3,** which contains the **OLACurrentUser** function: it returns the current user named ***Demo***.



	A	B	C	D	E	F	G	H
1	Database:	USING_OLATION						
2	Cube:	SALES					User Logged On	
3	Dimensions:	Filter	SALES Measure	Members	Amount		Demo	
4		Filter	Version	Members	Variance			
5		Filter	Region	Members	World			
6		Column	Account	Range	$B$10:$E$10			
7		Row	Month	Range	$A$11:$A$27			
8								
9	OLAPivotTable							
10		Sales	Margin	Cost of Sales	Margin Pcnt			
11	April	2,885.00	1,685.00	1,200.00	0.58			
12	2016	95,446.00	102,324.00	(6,878.00)	1.07			
13	2nd Quarter	10,281.00	6,721.00	3,560.00	0.65			
14	August	5,201.00	3,901.00	1,300.00	0.75			
15	3rd Quarter	15,640.00	11,460.00	4,180.00	0.73			

- By clicking in the formula bar area (as can be seen in the following screenshot, the mouse cursor is placed at the end of the formula), it will show the cell references corresponding to the **OLACurrentUser** function. Since this OLACurrentUser function only requires one parameter, the formula only shows the Connection parameter called "*USING_OLATION*"(**$B$1**).

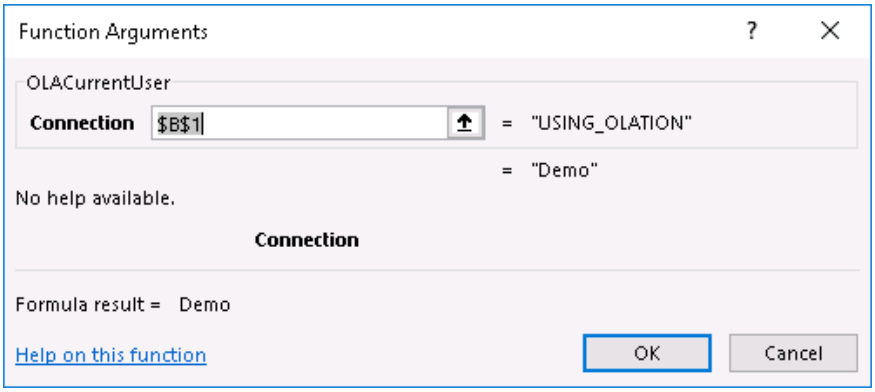


	A	B	C	D	E
1	Database:	USING_OLATION			
2	Cube:	SALES			
3	Dimensions:	Filter	SALES Measure	Members	Amount
4		Filter	Version	Members	Variance
5		Filter	Region	Members	World
6		Column	Account	Range	$B$10:$E$10
7		Row	Month	Range	$A$11:$A$27
8					
9	OLAPivotTable				
10		Sales	Margin	Cost of Sales	Margin Pcnt
11	April	2,885.00	1,685.00	1,200.00	0.58

Cell References:

=OLACurrentUserbe($B$1)

- $B$1– the Database name or the PowerExcel connection name/cell reference that contains the name of the Database or the PowerExcel connection, i.e., **USING_OLATION**



As previously mentioned, the return value of the formula for this example is the **Demo** user.

6. OLADatabase

Function Description: Establishes a connection to the source/target database via identifying the correct PowerExcel Connection that contains the necessary connection information of the source <Olation> Server and database.

Syntax: OLADatabase(Connection)

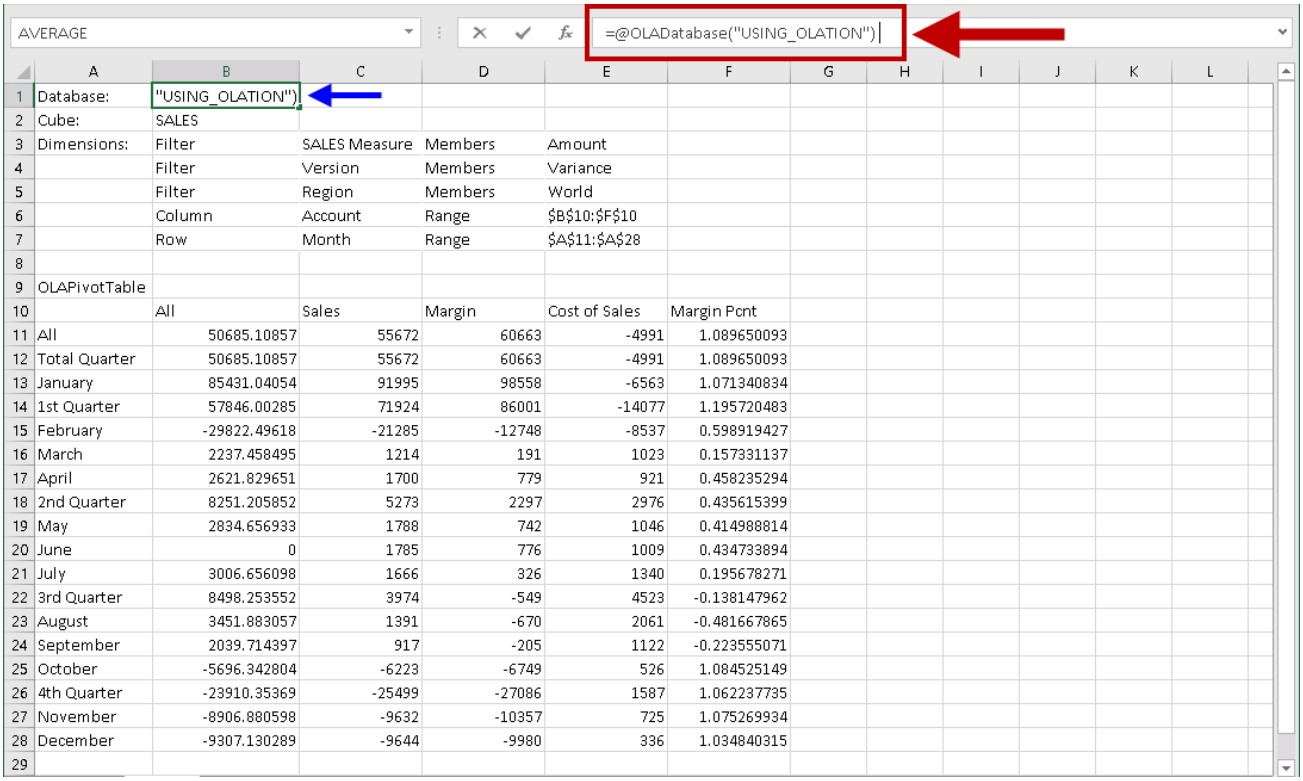
Connection: Enter the PowerExcel connection that contains the information about the Olation server URL and the source database name.

Remarks:

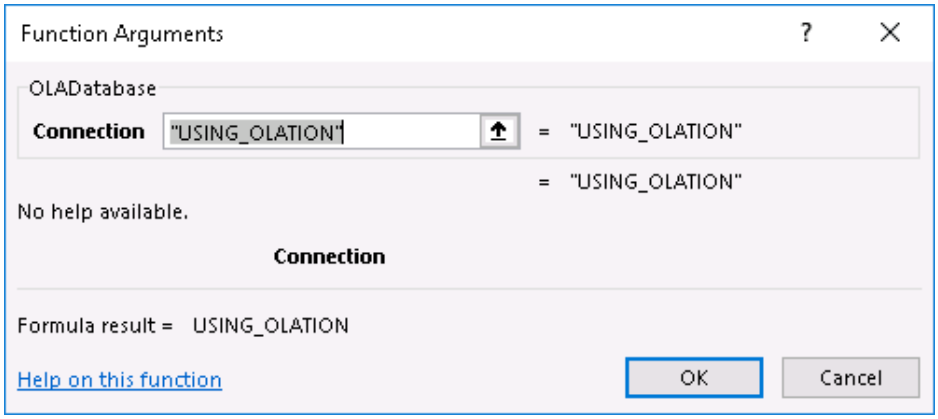
- The PowerExcel Connection must exist.
- The Olation Web Service must be running.
- The Database must be opened and running in the specified server.
- The 'Connection' parameter is compulsory.

Example:

The example Slice below shows a PowerExcel PivotTable with the **PowerExcel Connection/OLADatabase** named **"USING_OLATION"** located on cell B1.



	A	B	C	D	E	F
1	Database:	"USING_OLATION")				
2	Cube:	SALES				
3	Dimensions:	Filter	SALES Measure	Members	Amount	
4		Filter	Version	Members	Variance	
5		Filter	Region	Members	World	
6		Column	Account	Range	$B$10:$F$10	
7		Row	Month	Range	$A$11:$A$28	
8						
9	OLAPivotTable					
10		All	Sales	Margin	Cost of Sales	Margin Pcnt
11	All	50685.10857	55672	60663	-4991	1.089650093
12	Total Quarter	50685.10857	55672	60663	-4991	1.089650093
13	January	85431.04054	91995	98558	-6563	1.071340834
14	1st Quarter	57846.00285	71924	86001	-14077	1.195720483
15	February	-29822.49618	-21285	-12748	-8537	0.598919427
16	March	2237.458495	1214	191	1023	0.157331137
17	April	2621.829651	1700	779	921	0.458235294
18	2nd Quarter	8251.205852	5273	2297	2976	0.435615399
19	May	2834.656933	1788	742	1046	0.414988814
20	June	0	1785	776	1009	0.434733894
21	July	3006.656098	1666	326	1340	0.195678271
22	3rd Quarter	8498.253552	3974	-549	4523	-0.138147962
23	August	3451.883057	1391	-670	2061	-0.481667865
24	September	2039.714397	917	-205	1122	-0.223555071
25	October	-5696.342804	-6223	-6749	526	1.084525149
26	4th Quarter	-23910.35369	-25499	-27086	1587	1.062237735
27	November	-8906.880598	-9632	-10357	725	1.075269934
28	December	-9307.130289	-9644	-9980	336	1.034840315
29						



Cell References:

```
=@OLADatabase("USING_OLATION")
```

- "USING_OLATION" – this is the name of the PowerExcel connection or the Database name that we want to establish connection to.

7. OLADimension

Function Description: This function returns the specified Dimension or the Dimension name that corresponds to a specified Dimension Index number that exists in a specified PowerExcel/Database Connection.

Syntax: OLADimension(Connection,Dimension)

OR

OLADimension(Connection,DimensionIndex)

Connection: Enter the PowerExcel connection that contains the information about the Olation server URL and the source database name.

Dimension: Enter the name of the Dimension or the cell reference that contains the name of the Dimension you wish to return.

Dimension Index: The index number corresponding to the Dimension you want to return.

Remarks:

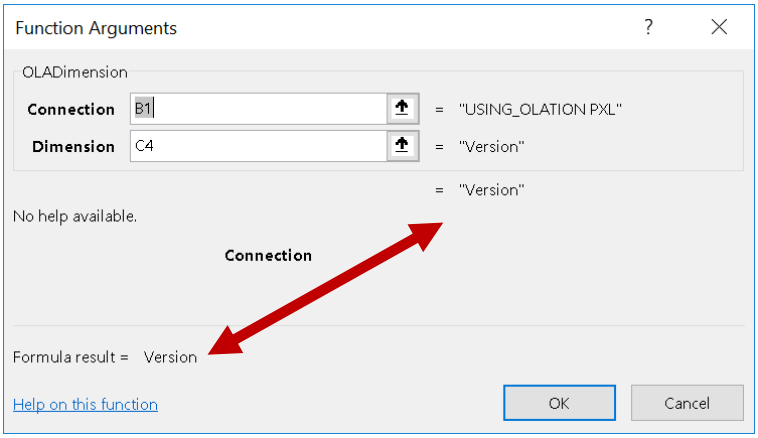
- The PowerExcel Connection must exist.
- The Olation Web Service must be running.
- The Database must be opened and running in the specified server.
- The 'Connection' parameter is compulsory.
- Each Dimension in the database is assigned an index number starting from 1, 2, 3… and so on. If there is no Dimension assigned to that index number, then it will return a #NAME? error.
- The index number assigned to the Dimension is based on the order they are created within the specified database.

Example 1: OLADimension(Connection,DimensionIndex)

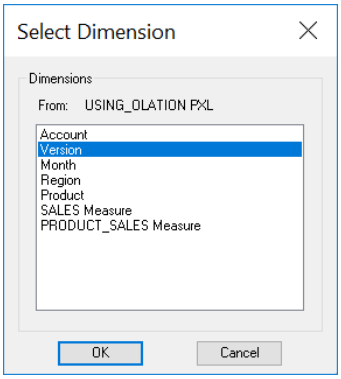
This first example, like the ones for **OLACube** and **OLACubeDimension** and **OLAPCubeMember**, will show how to make the function return a selection window for all Dimensions in the Database. (This is in contrast to OLACubeDimension, which concerns accessing the Dimensions in a selected Cube.)

- Using an existing Slice, select a cell to the right of the field of data, e.g., Cell **H5**.
- In the Excel formula bar, click on the Insert Function symbol (***fx***). The Insert Function window will appear.
- In the **Or select a category drop-down**, select **PowerExcel.ExcelFunctions**.
- Select **OLADimension**. Click **OK**.

- For Connection, you can reference Cell **B1** from the sample Slice (or type B1); next, reference a selected Dimension (e.g., ***Version***).

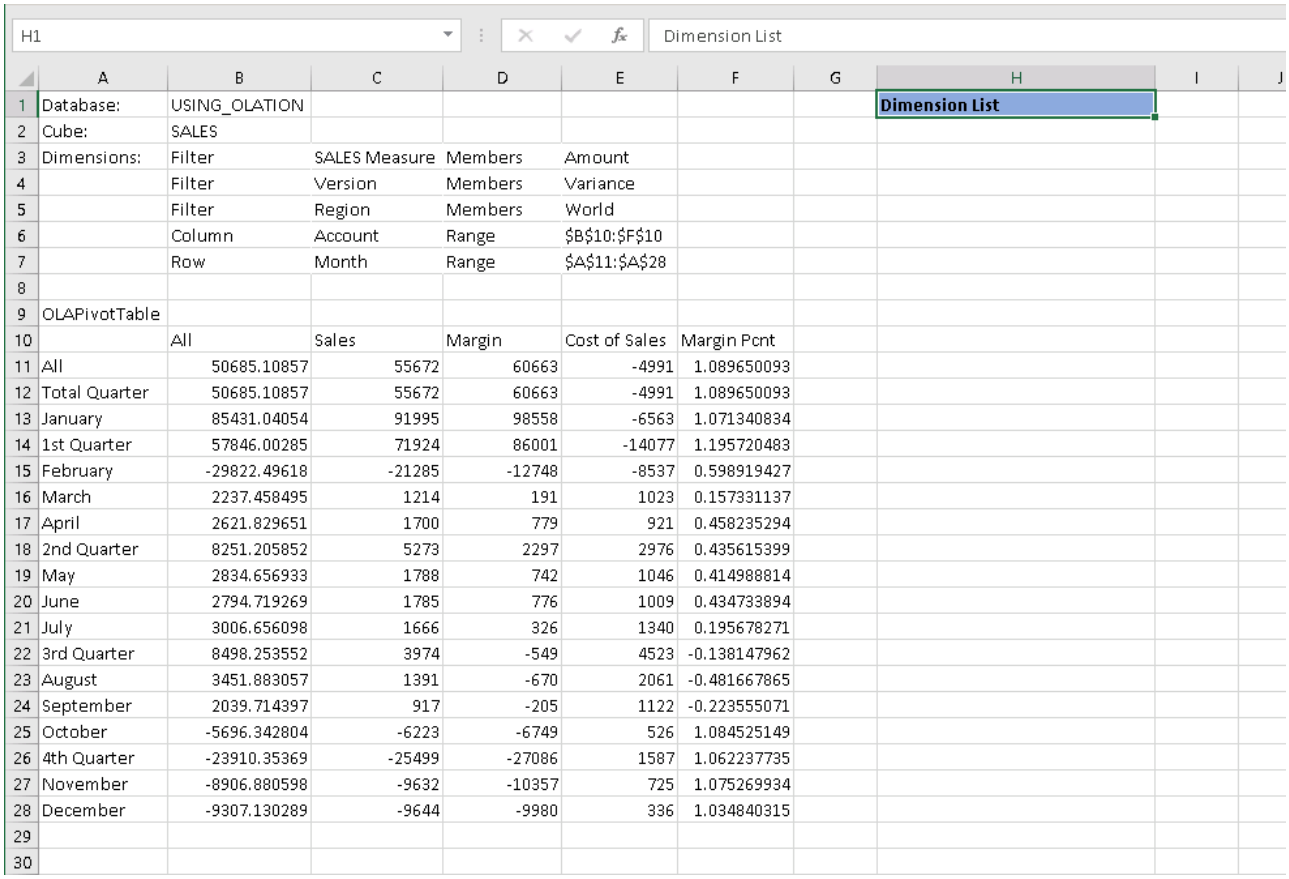


- Click **OK**: the result ("Version") will appear in the selected cell (e.g., H5).
- Next, **double-click** on that Cell. The Select Dimension window appears showing all Dimensions from the Database (i.e., not limited to any specified Cube), as shown below.



Example 2: OLADimension(Connection,DimensionIndex)

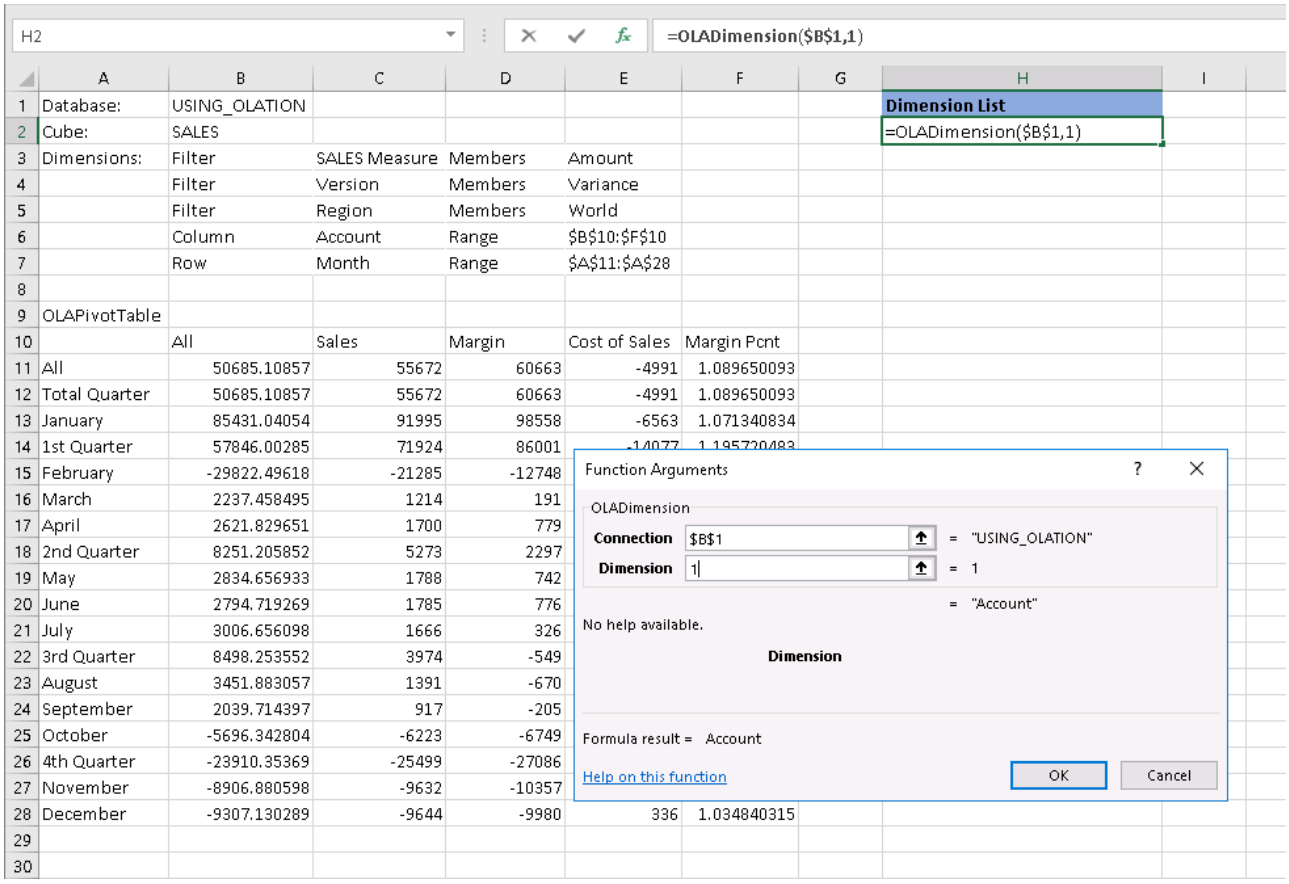
For example, we have a sample PowerExcel Slice and want to see a complete list of the Dimensions that exist in our source database and return those Dimensions in Excel.



H1 | Dimension List

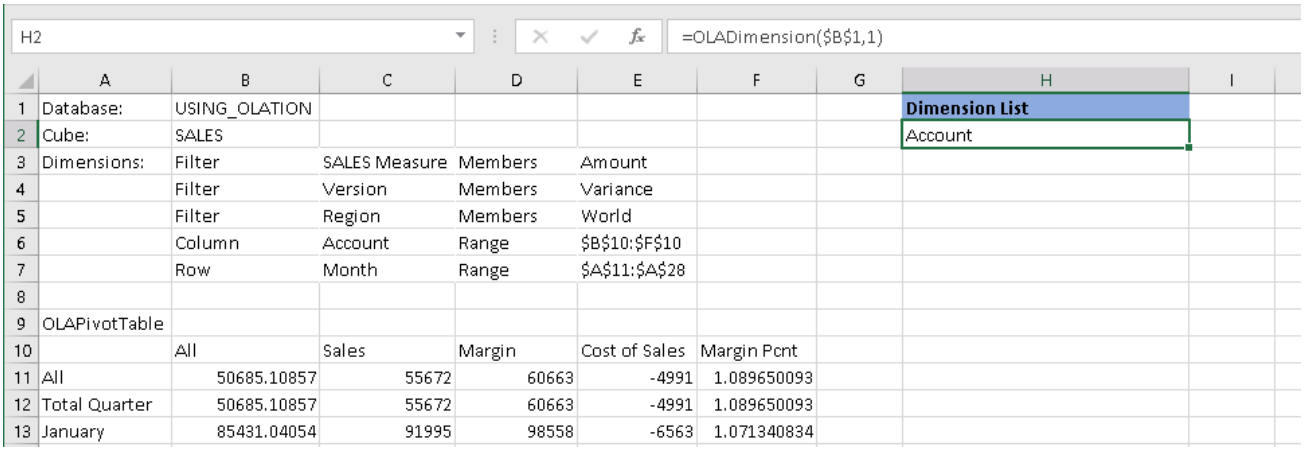
	A	B	C	D	E	F	G	H	I	J
1	Database:	USING_OLATION						**Dimension List**		
2	Cube:	SALES								
3	Dimensions:	Filter	SALES Measure	Members	Amount					
4		Filter	Version	Members	Variance					
5		Filter	Region	Members	World					
6		Column	Account	Range	$B$10:$F$10					
7		Row	Month	Range	$A$11:$A$28					
8										
9	OLAPivotTable									
10		All	Sales	Margin	Cost of Sales	Margin Pcnt				
11	All	50685.10857	55672	60663	-4991	1.089650093				
12	Total Quarter	50685.10857	55672	60663	-4991	1.089650093				
13	January	85431.04054	91995	98558	-6563	1.071340834				
14	1st Quarter	57846.00285	71924	86001	-14077	1.195720483				
15	February	-29822.49618	-21285	-12748	-8537	0.598919427				
16	March	2237.458495	1214	191	1023	0.157331137				
17	April	2621.829651	1700	779	921	0.458235294				
18	2nd Quarter	8251.205852	5273	2297	2976	0.435615399				
19	May	2834.656933	1788	742	1046	0.414988814				
20	June	2794.719269	1785	776	1009	0.434733894				
21	July	3006.656098	1666	326	1340	0.195678271				
22	3rd Quarter	8498.253552	3974	-549	4523	-0.138147962				
23	August	3451.883057	1391	-670	2061	-0.481667865				
24	September	2039.714397	917	-205	1122	-0.223555071				
25	October	-5696.342804	-6223	-6749	526	1.084525149				
26	4th Quarter	-23910.35369	-25499	-27086	1587	1.062237735				
27	November	-8906.880598	-9632	-10357	725	1.075269934				
28	December	-9307.130289	-9644	-9980	336	1.034840315				
29										
30										

- With an example Slice already created (as in the above image), In Cell **H1** (or any empty cell), type the caption '**Dimension List**'.
- Now, to use the **OLADimension** function to list all Dimensions that exist within the database (*USING_OLATION*, in the example): In Cell **H2** click Dimethe **Function** button located beside the formula bar. The Insert Function dialog box appears.
- In the Insert Function dialog box, select **PowerExcel.ExcelFunctions** as the category, then select **OLADimension** as the function. The Function Arguments dialog box appears.
- In the Function Arguments dialog, click in the **Connection** field, then click on Cell **B1**, which contains the Database connection reference. Notice that the Database connection name "USING_OLATION" appears beside the connection field.
  Note: You can use the absolute reference to easily copy the formulas across to other: **$B$1**.



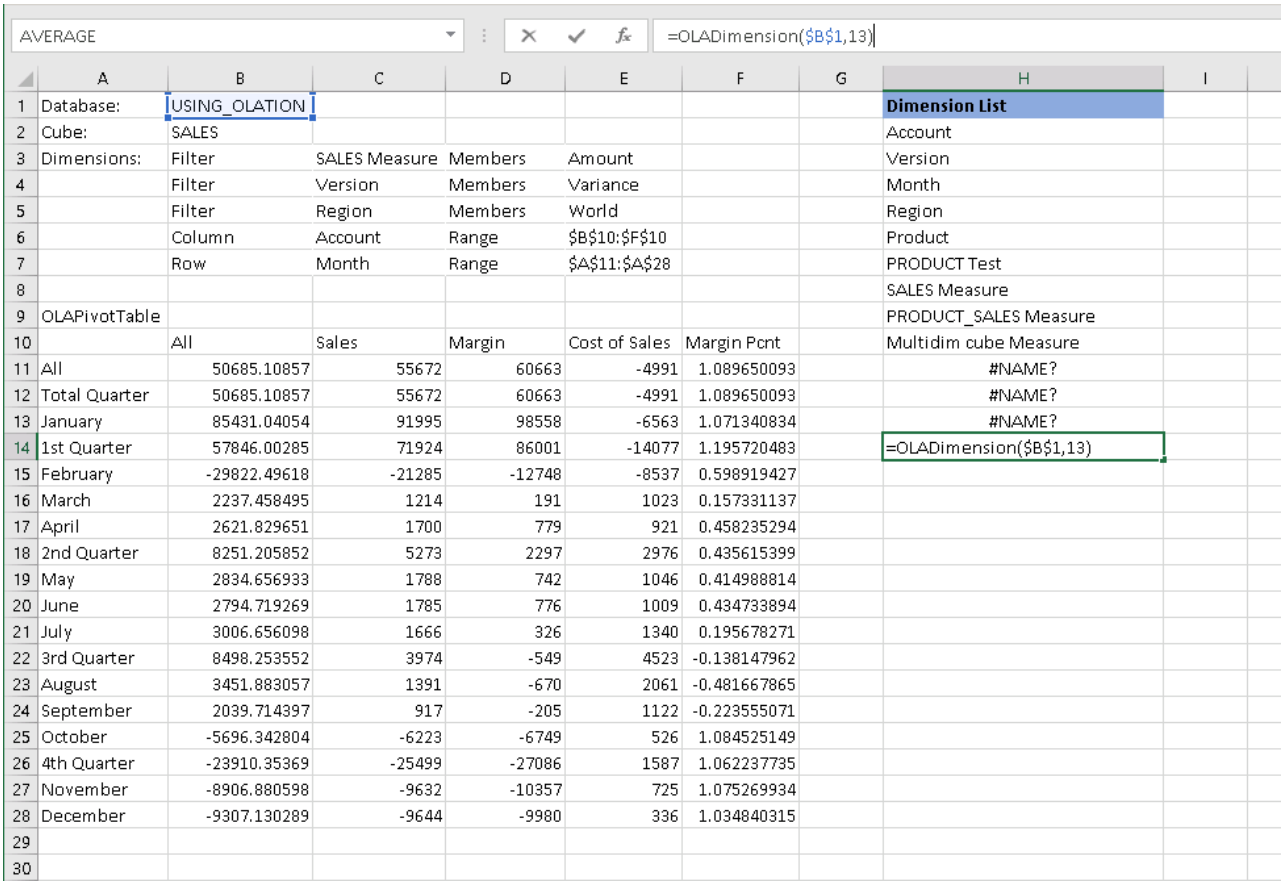
	A	B	C	D	E	F	G	H	I
1	Database:	USING_OLATION						Dimension List	
2	Cube:	SALES						=OLADimension($B$1,1)	
3	Dimensions:	Filter	SALES Measure	Members	Amount				
4		Filter	Version	Members	Variance				
5		Filter	Region	Members	World				
6		Column	Account	Range	$B$10:$F$10				
7		Row	Month	Range	$A$11:$A$28				
8									
9	OLAPivotTable								
10		All	Sales	Margin	Cost of Sales	Margin Pcnt			
11	All	50685.10857	55672	60663	-4991	1.089650093			
12	Total Quarter	50685.10857	55672	60663	-4991	1.089650093			
13	January	85431.04054	91995	98558	-6563	1.071340834			
14	1st Quarter	57846.00285	71924	86001	-14077	1.195720483			
15	February	-29822.49618	-21285	-12748					
16	March	2237.458495	1214	191					
17	April	2621.829651	1700	779					
18	2nd Quarter	8251.205852	5273	2297					
19	May	2834.656933	1788	742					
20	June	2794.719269	1785	776					
21	July	3006.656098	1666	326					
22	3rd Quarter	8498.253552	3974	-549					
23	August	3451.883057	1391	-670					
24	September	2039.714397	917	-205					
25	October	-5696.342804	-6223	-6749					
26	4th Quarter	-23910.35369	-25499	-27086					
27	November	-8906.880598	-9632	-10357					
28	December	-9307.130289	-9644	-9980	336	1.034840315			
29									
30									

- In the **Dimension** field, type the index number: 1
- 1 begin with index value **1** then click **OK**. Back in the Excel worksheet in cell **H2**, the *Account* dimension appears (as in the image below).



	A	B	C	D	E	F	G	H	I
1	Database:	USING_OLATION						Dimension List	
2	Cube:	SALES						Account	
3	Dimensions:	Filter	SALES Measure	Members	Amount				
4		Filter	Version	Members	Variance				
5		Filter	Region	Members	World				
6		Column	Account	Range	$B$10:$F$10				
7		Row	Month	Range	$A$11:$A$28				
8									
9	OLAPivotTable								
10		All	Sales	Margin	Cost of Sales	Margin Pcnt			
11	All	50685.10857	55672	60663	-4991	1.089650093			
12	Total Quarter	50685.10857	55672	60663	-4991	1.089650093			
13	January	85431.04054	91995	98558	-6563	1.071340834			

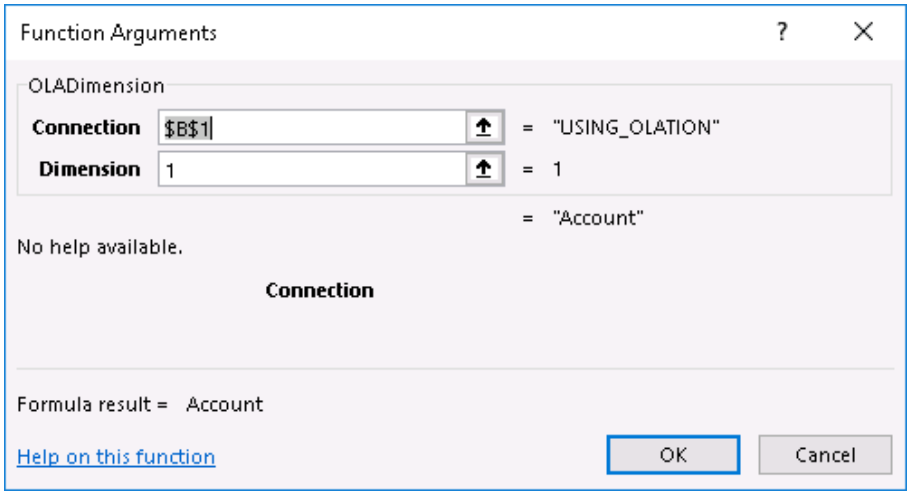
- **Copy the formula down column H** and just change the last parameter, which is the Dimension Index value with **2**, **3**, **4** and so on. The function will start to return #NAME? error when there are no more Dimension corresponding to an index number, meaning you reached the end of the Dimension list.



AVERAGE | =OLADimension($B$1,13)

	A	B	C	D	E	F	G	H	I
1	Database:	USING_OLATION						**Dimension List**	
2	Cube:	SALES						Account	
3	Dimensions:	Filter	SALES Measure	Members	Amount			Version	
4		Filter	Version	Members	Variance			Month	
5		Filter	Region	Members	World			Region	
6		Column	Account	Range	$B$10:$F$10			Product	
7		Row	Month	Range	$A$11:$A$28			PRODUCT Test	
8								SALES Measure	
9	OLAPivotTable							PRODUCT_SALES Measure	
10		All	Sales	Margin	Cost of Sales	Margin Pcnt		Multidim cube Measure	
11	All	50685.10857	55672	60663	-4991	1.089650093		#NAME?	
12	Total Quarter	50685.10857	55672	60663	-4991	1.089650093		#NAME?	
13	January	85431.04054	91995	98558	-6563	1.071340834		#NAME?	
14	1st Quarter	57846.00285	71924	86001	-14077	1.195720483		=OLADimension($B$1,13)	
15	February	-29822.49618	-21285	-12748	-8537	0.598919427			
16	March	2237.458495	1214	191	1023	0.157331137			
17	April	2621.829651	1700	779	921	0.458235294			
18	2nd Quarter	8251.205852	5273	2297	2976	0.435615399			
19	May	2834.656933	1788	742	1046	0.414988814			
20	June	2794.719269	1785	776	1009	0.434733894			
21	July	3006.656098	1666	326	1340	0.195678271			
22	3rd Quarter	8498.253552	3974	-549	4523	-0.138147962			
23	August	3451.883057	1391	-670	2061	-0.481667865			
24	September	2039.714397	917	-205	1122	-0.223555071			
25	October	-5696.342804	-6223	-6749	526	1.084525149			
26	4th Quarter	-23910.35369	-25499	-27086	1587	1.062237735			
27	November	-8906.880598	-9632	-10357	725	1.075269934			
28	December	-9307.130289	-9644	-9980	336	1.034840315			
29									
30									

- In the above example there are 9 existing Dimensions within the *USING_OLATION* database. If you click on any cell containing the **OLADimension** formula (as in the image above) then click on the Formula bar, you will see the cell/index references:

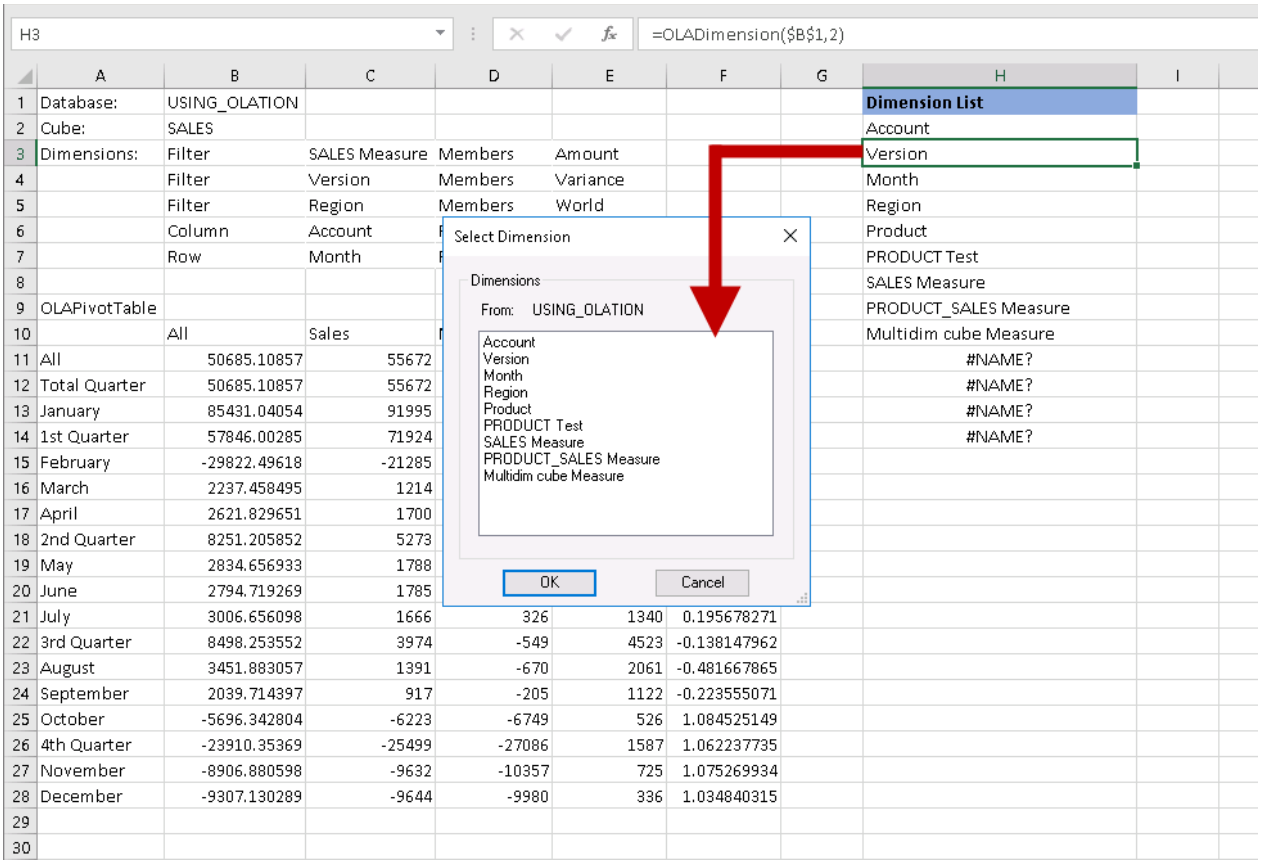


Cell References:

```
=OLADimension ($B$1,1)
```

- $B$1– the Database name or the PowerExcel connection name/cell reference that contains the name of the Database or the PowerExcel connection, i.e., **USING_OLATION**
- 1 –the Dimension Index number; the corresponding Member to this index value in this case is **Account**

- If you double-click on the cell that contains the **OLADimension** formula (as in the image below), the Select Dimension dialog will appear—this also shows a list all existing Dimensions within the source database.



	A	B	C	D	E	F	G	H	I
1	Database:	USING_OLATION						**Dimension List**	
2	Cube:	SALES						Account	
3	Dimensions:	Filter	SALES Measure	Members	Amount			Version	
4		Filter	Version	Members	Variance			Month	
5		Filter	Region	Members	World			Region	
6		Column	Account					Product	
7		Row	Month					PRODUCT Test	
8								SALES Measure	
9	OLAPivotTable							PRODUCT_SALES Measure	
10		All	Sales					Multidim cube Measure	
11	All	50685.10857	55672					#NAME?	
12	Total Quarter	50685.10857	55672					#NAME?	
13	January	85431.04054	91995					#NAME?	
14	1st Quarter	57846.00285	71924					#NAME?	
15	February	-29822.49618	-21285						
16	March	2237.458495	1214						
17	April	2621.829651	1700						
18	2nd Quarter	8251.205852	5273						
19	May	2834.656933	1788						
20	June	2794.719269	1785						
21	July	3006.656098	1666	326	1340	0.195678271			
22	3rd Quarter	8498.253552	3974	-549	4523	-0.138147962			
23	August	3451.883057	1391	-670	2061	-0.481667865			
24	September	2039.714397	917	-205	1122	-0.223555071			
25	October	-5696.342804	-6223	-6749	526	1.084525149			
26	4th Quarter	-23910.35369	-25499	-27086	1587	1.062237735			
27	November	-8906.880598	-9632	-10357	725	1.075269934			
28	December	-9307.130289	-9644	-9980	336	1.034840315			
29									
30									

8. OLAMember

Function Description: This function returns the specified Member within the specified Dimension that exists in a specified PowerExcel/Database Connection.

Syntax: OLADimension(Connection,Dimension,Member)

OR

OLADimension(Connection,Dimension,MemberIndex)

Connection: Enter the PowerExcel connection which contains the information about the Olation server URL and the source database name.

Dimension: Enter the name of the Dimension or the cell reference that contains the name of the Dimension where the Member to be returned exists.

Member: Enter the name of the Member or the cell reference that contains the name of the Member you wish to return.

Member Index: The index number corresponding to the Member you want to return.

Remarks:

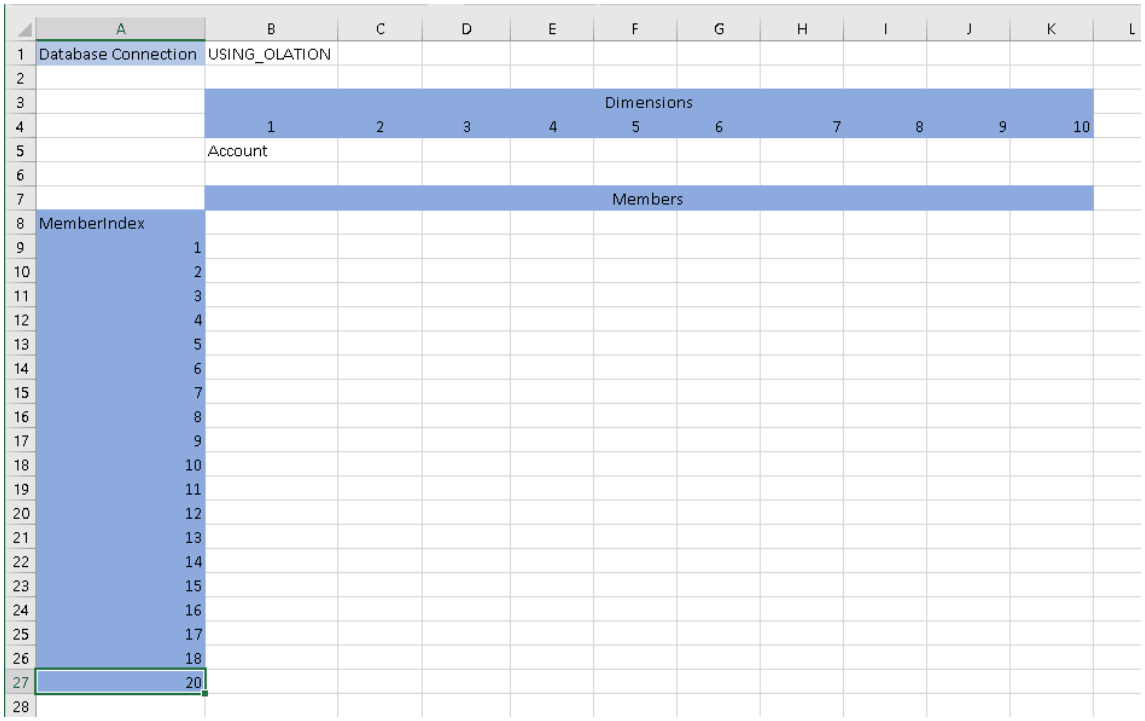
- The PowerExcel Connection must exist.
- The Olation Web Service must be running.
- The Database must be opened and running in the specified server.
- The 'Connection' and 'Dimension' parameters are compulsory.
- All component Members of Dimensions within the database are each assigned an index number starting from 1, 2, 3… and so on. If there is no Member assigned to that index number, then it will return a #NAME? error.

Example: OLAMember (Connection,Dimension,MemberIndex)

For this example, we will use the **OLAMember** function to identify and make a list of all the Members that exist for the Dimensions in our source database (*USING_OLATION*).

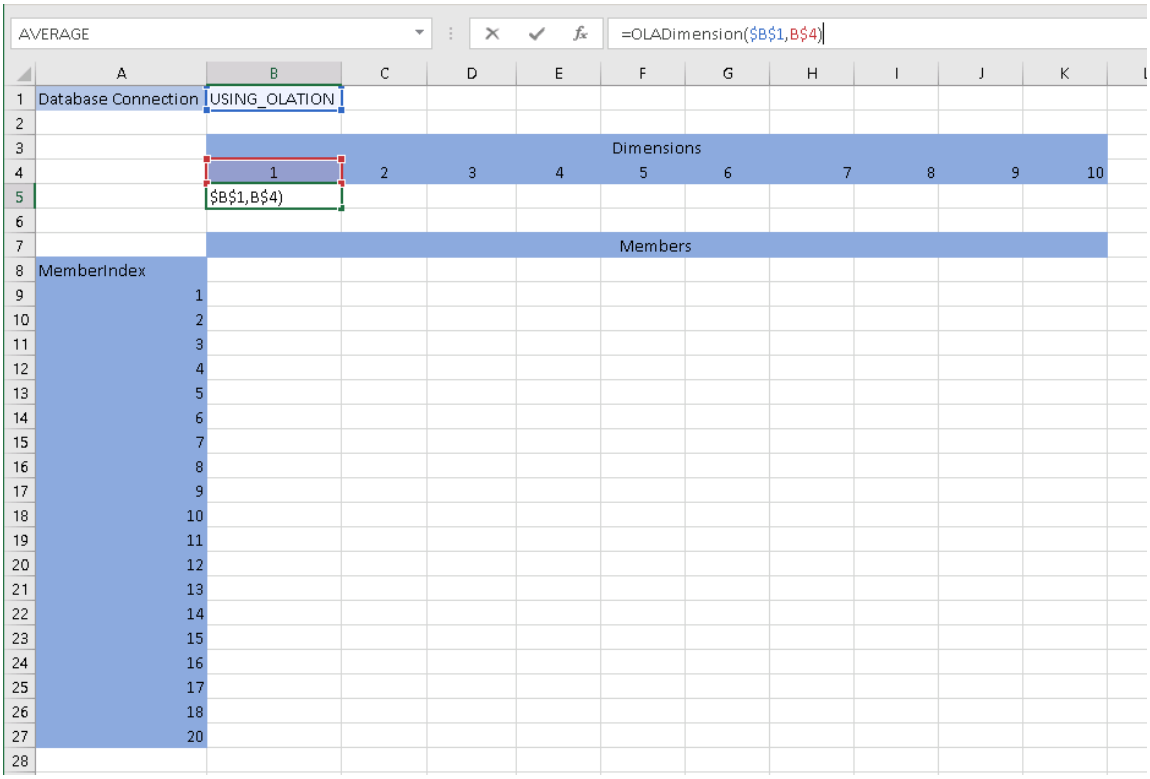
- First establish a connection a the target database. In cell A1 type in **Database connection** (cells that are descriptive—i.e., non-formula-derived—are blue-highlighted for easy identification), then go to cell B1 and use the OLADatabase function to establish a connection to the target database (in the example, the Database connection is =@OLADatabase("USING_OLATION").
- For ease in copying the functions across other cells, we will make use of index numbers and apply absolute references to the OLADimension and OLAMember formula functions. Row 4, starting cell B4, type numbers **1 to 10** (type in cells **B4 to K4**). In the row above enter the term **Dimensions**.

  In cells **A9 to A27**, type the numbers **1 to 20** and put the term **MemberIndex** in cell **A8** (all as shown in the following image.)



	A	B	C	D	E	F	G	H	I	J	K	L
1	Database Connection	USING_OLATION										
2												
3		Dimensions										
4		1	2	3	4	5	6	7	8	9	10	
5		Account										
6												
7		Members										
8	MemberIndex											
9	1											
10	2											
11	3											
12	4											
13	5											
14	6											
15	7											
16	8											
17	9											
18	10											
19	11											
20	12											
21	13											
22	14											
23	15											
24	16											
25	17											
26	18											
27	20											
28												

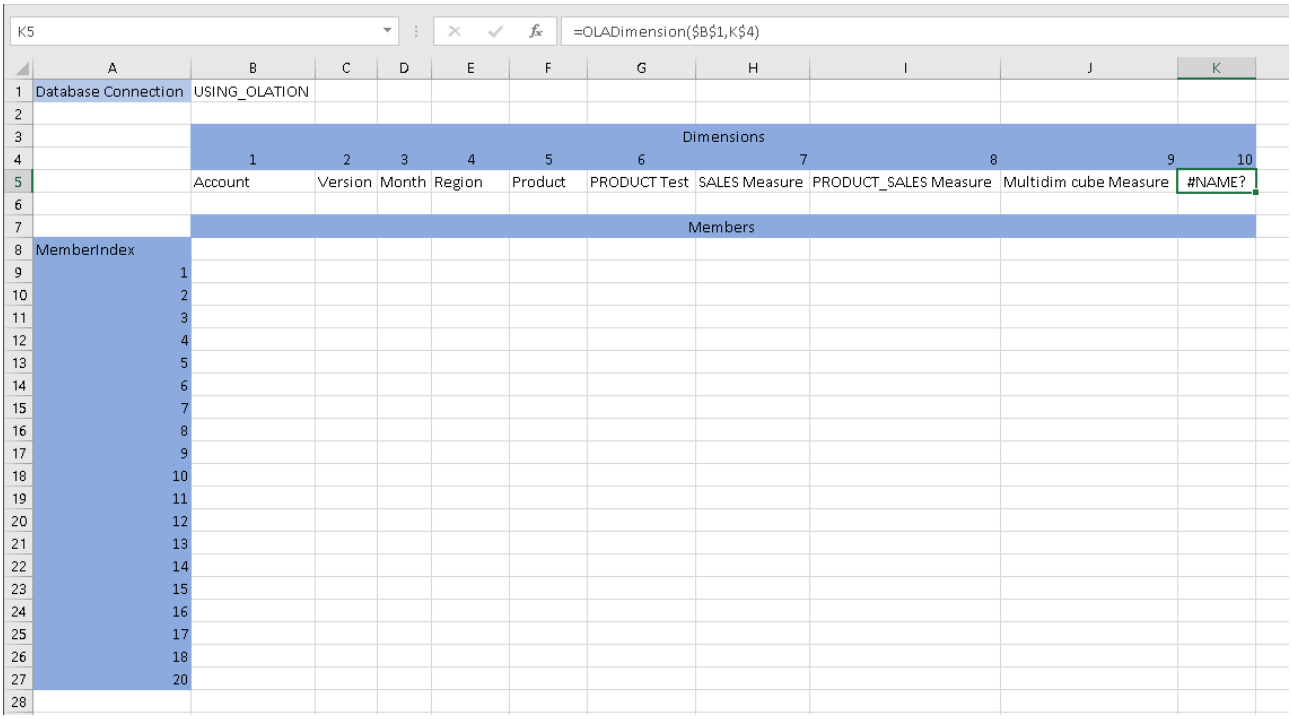
- In cells **B5 to K5**, use the **OLAMember** function to pull in the Dimensions that exist within the *USING_OLATION* database. In the example, define the formula in cell B5 as: =OLADimension($B$1,B$4).



AVERAGE | =OLADimension($B$1,B$4)

	A	B	C	D	E	F	G	H	I	J	K	L
1	Database Connection	USING_OLATION										
2												
3		Dimensions										
4		1	2	3	4	5	6	7	8	9	10	
5		$B$1,B$4)										
6												
7		Members										
8	MemberIndex											
9	1											
10	2											
11	3											
12	4											
13	5											
14	6											
15	7											
16	8											
17	9											
18	10											
19	11											
20	12											
21	13											
22	14											
23	15											
24	16											
25	17											
26	18											
27	20											
28												

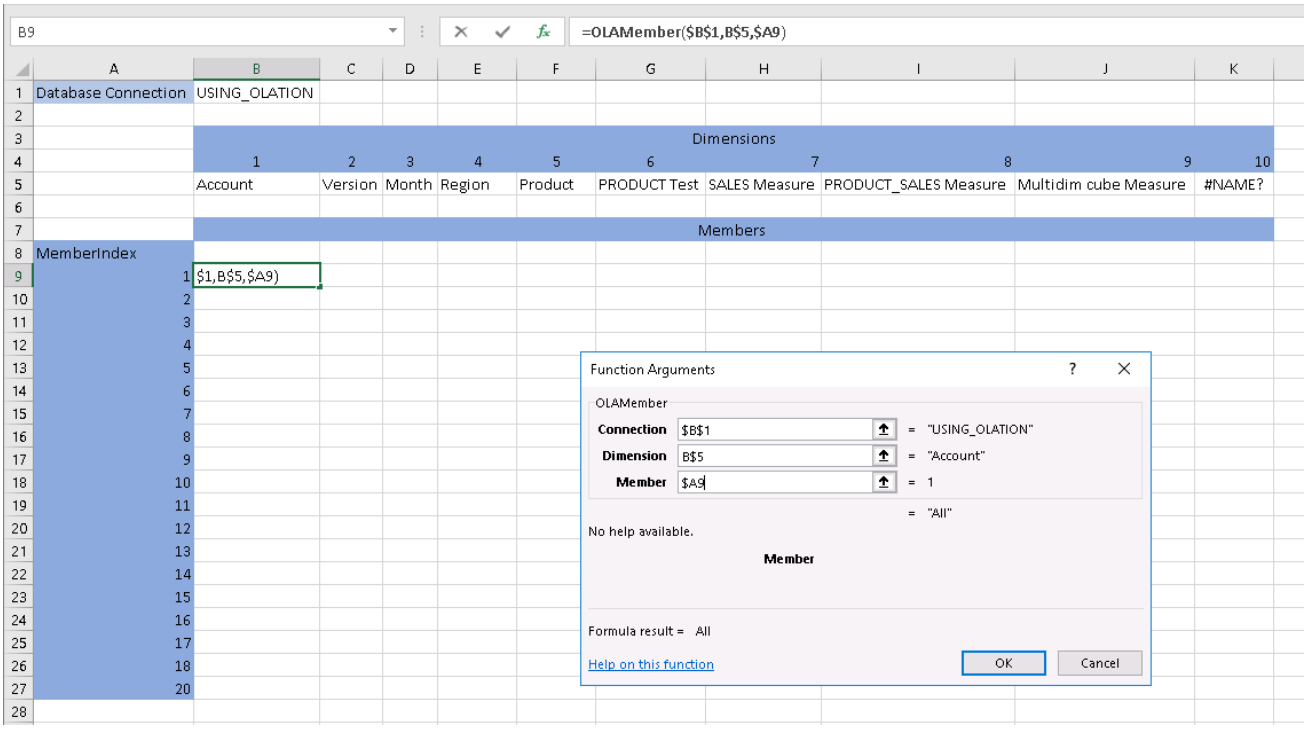
- **Copy the formula** to cells **C5** to **K5**. Click **F9** to refresh the Excel values.



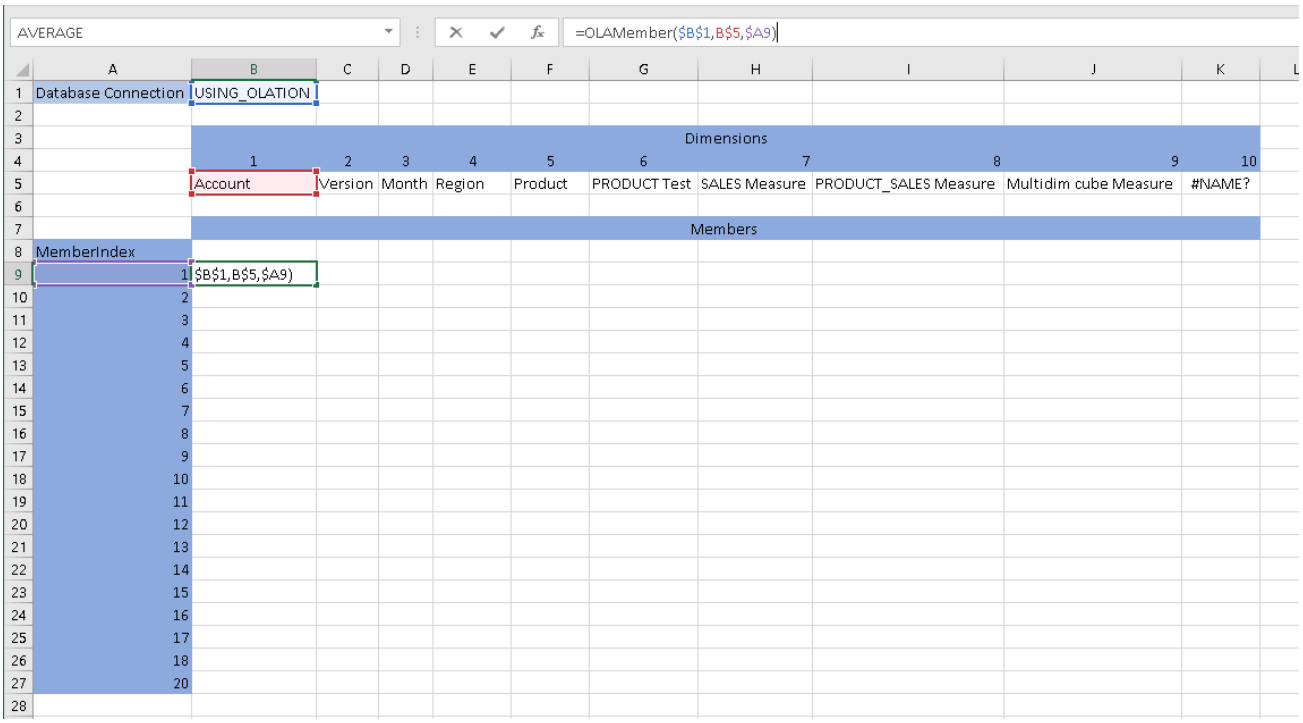
K5 =OLADimension($B$1,K$4)

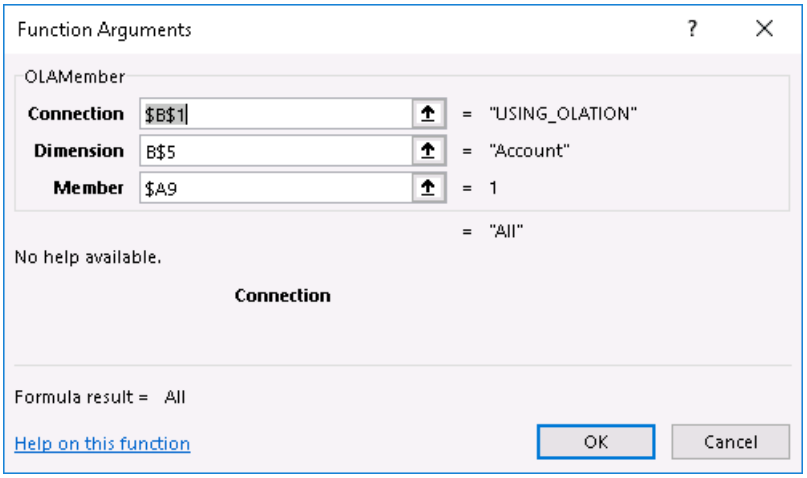
	A	B	C	D	E	F	G	H	I	J	K
1	Database Connection	USING_OLATION									
2											
3		Dimensions									
4		1	2	3	4	5	6	7	8	9	10
5		Account	Version	Month	Region	Product	PRODUCT Test	SALES Measure	PRODUCT_SALES Measure	Multidim cube Measure	#NAME?
6											
7		Members									
8	MemberIndex										
9	1										
10	2										
11	3										
12	4										
13	5										
14	6										
15	7										
16	8										
17	9										
18	10										
19	11										
20	12										
21	13										
22	14										
23	15										
24	16										
25	17										
26	18										
27	20										
28											

- Next use the **OLAMember** function to create a list of Members for each Dimension that exist within the *USING_OLATION* database: in Cell **B9** click next to the **Function** button located beside the formula bar. The Insert Function dialog box appears.
- In the category list, select **PowerExcel.ExcelFunctions.OlaFuncions**; click **OLAMember** and click **OK**. The Function Arguments dialog box appears. This is where you will define the **OLAMember** formula.
- In the Function Arguments dialog, click on the **Connection** field, then click on cell **B1**, which contains the Database connection reference. Notice that the Database connection name “USING_OLATION” appears beside the connection field.
  Note: You can use the absolute reference to easily copy the formulas across to other cells: **$B$1**.
- Click on the **Dimension** field, then click on cell **B5**, which contains the Dimension reference. Note that the Dimension name “Account” appears beside the Dimension field.
  Note: You can again use an absolute reference for this formula: **B$5**.
- Lastly, click on the **Member** field, then click on cell **A9**, which will dictate the index value for the last parameter. Again, use an absolute reference along the columns: **$A9**.
- The Function Arguments dialog will look as follows:



- Click **OK**. This will return the Member **All**. Note the cell references in the following image:





Cell References:

=OLAMember($B$1,B$5,$A9)

- $B$1– the Database name or the PowerExcel connection name/cell reference that contains the name of the Database or the PowerExcel connection, i.e., **USING_OLATION**
- B$5– the Dimension name having the index value of 1 in the *USING_OLATION* Database, i.e., **Account**
- $A9– the Member Index number or cell reference, i.e., **1**

- Now, copy this formula across the range **B9:K27**.

K27 =@OLAMember($B$1,K$5,$A27)

	A	B	C	D	E	F	G	H	I	J	K
1	Database Connection	USING_OLATION									
2											
3		Dimensions									
4		1	2	3	4	5	6	7	8	9	10
5		Account	Version	Month	Region	Product	PRODUCT Test	SALES Measure	PRODUCT_SALES Measure	Multidim cube Measure	#NAME?
6											
7		Members									
8	MemberIndex										
9	1	All	Variance	All	World	All	All	Amount	Amount	Amount	#VALUE!
10	2	Margin	Actual	Total Quarter	South America	Current Product	A	#NAME?	#NAME?	Text	#VALUE!
11	3	Sales	Budget	January	North America	New Product 1	B	#NAME?	#NAME?	#NAME?	#VALUE!
12	4	Cost of Sales	#NAME?	February	Argentina	New Product 2	C	#NAME?	#NAME?	#NAME?	#VALUE!
13	5	Margin Pcnt	#NAME?	March	Brazil	New Product 3	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
14	6	#NAME?	#NAME?	April	Canada	New Product 4	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
15	7	#NAME?	#NAME?	May	Mexico	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
16	8	#NAME?	#NAME?	June	United States	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
17	9	#NAME?	#NAME?	July	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
18	10	#NAME?	#NAME?	August	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
19	11	#NAME?	#NAME?	September	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
20	12	#NAME?	#NAME?	October	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
21	13	#NAME?	#NAME?	November	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
22	14	#NAME?	#NAME?	December	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
23	15	#NAME?	#NAME?	1st Quarter	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
24	16	#NAME?	#NAME?	2nd Quarter	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
25	17	#NAME?	#NAME?	3rd Quarter	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
26	18	#NAME?	#NAME?	4th Quarter	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
27	20	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#NAME?	#VALUE!
28											
29											

- Press **F9** or refresh the Excel worksheet. You will see that a full list of the Members appears, by Dimension, in the USING_OLATION database.

9. OLAPivotTable

Function Description: This function, when used as the means to bring data into a Slice, creates a sophisticated array of data that enables a user to pivot or re-arrange data, easily apply data constraints, and quicklycustomize the spreadsheet view.

Syntax: OLAPivotTable(Connection,Cube,Dimension1,Dimension2 Dimension3,…, DimensionN,CellLocation,True/False1,True/False2,True/False3)

Connection: The PowerExcel connection which contains the information about the Olation server URL and the source database name.

Cube: The name of the source/target Cube; or enter the cell reference that contains the name of the source or target Cube you wish to establish connection to.

Dimension1 to DimensionN: The related Dimension references.

CellLocation: the cell location where the OLAPivotTable will start to bring in data

True/False1: relates to checkbox Constrain Empty Rows

True/False2: relates to checkbox Delete Removed Rows

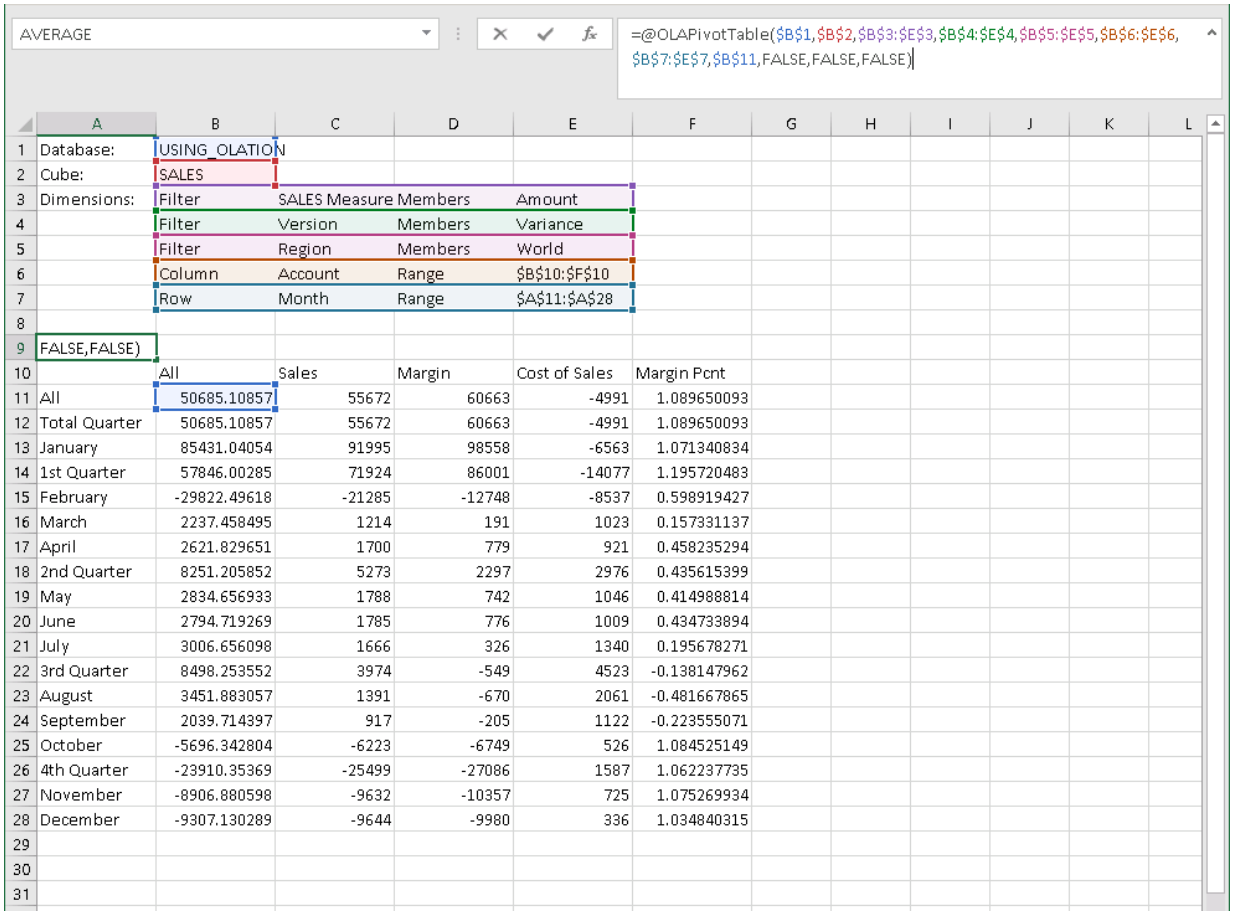
True/False3: relates to checkbox Expandable Members (under development)

Remarks:

- The PowerExcel Connection must exist.
- The Olation Web Service must be running.
- The Database must be opened and running in the specified server.
- The 'Connection', 'Cube' and 'Dimension' parameters are compulsory.
- CellLocation. TrueFalse1, TrueFalse2, TrueFalse3 are compulsory.

Example: OLAPivotTable(Connection,Cube,Dimension1,Dimension2,Dimension3,…, DimensionN,CellLocation,True/False1,True/False2,True/False3)

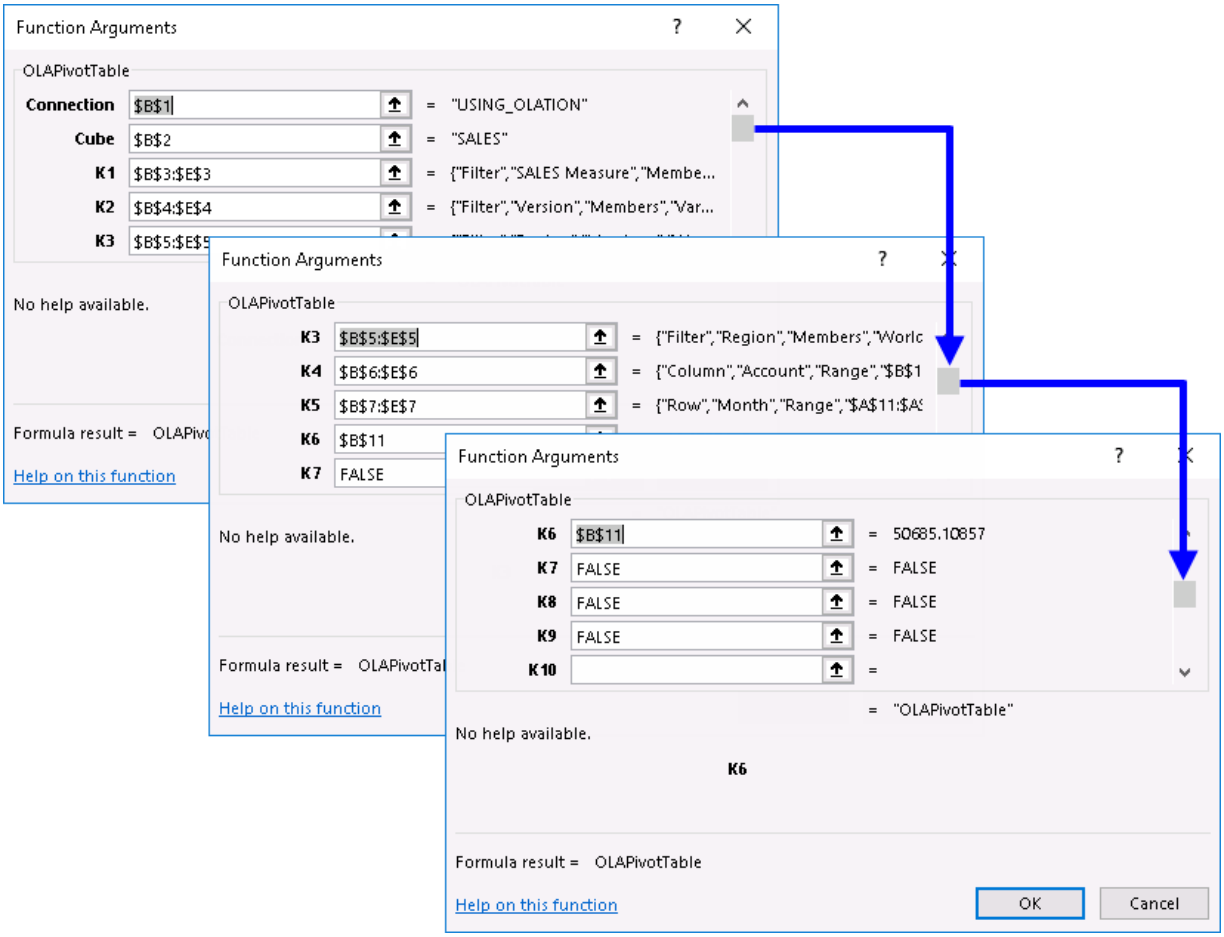
The Example Slice below shows a PowerExcel PivotTable. Let us look at the corresponding cell references:



AVERAGE

=@OLAPivotTable($B$1,$B$2,$B$3:$E$3,$B$4:$E$4,$B$5:$E$5,$B$6:$E$6,$B$7:$E$7,$B$11,FALSE,FALSE,FALSE)

	A	B	C	D	E	F	G	H	I	J	K	L
1	Database:	USING_OLATION										
2	Cube:	SALES										
3	Dimensions:	Filter	SALES Measure	Members	Amount							
4		Filter	Version	Members	Variance							
5		Filter	Region	Members	World							
6		Column	Account	Range	$B$10:$F$10							
7		Row	Month	Range	$A$11:$A$28							
8												
9	FALSE,FALSE)											
10		All	Sales	Margin	Cost of Sales	Margin Pcnt						
11	All	50685.10857	55672	60663	-4991	1.089650093						
12	Total Quarter	50685.10857	55672	60663	-4991	1.089650093						
13	January	85431.04054	91995	98558	-6563	1.071340834						
14	1st Quarter	57846.00285	71924	86001	-14077	1.195720483						
15	February	-29822.49618	-21285	-12748	-8537	0.598919427						
16	March	2237.458495	1214	191	1023	0.157331137						
17	April	2621.829651	1700	779	921	0.458235294						
18	2nd Quarter	8251.205852	5273	2297	2976	0.435615399						
19	May	2834.656933	1788	742	1046	0.414988814						
20	June	2794.719269	1785	776	1009	0.434733894						
21	July	3006.656098	1666	326	1340	0.195678271						
22	3rd Quarter	8498.253552	3974	-549	4523	-0.138147962						
23	August	3451.883057	1391	-670	2061	-0.481667865						
24	September	2039.714397	917	-205	1122	-0.223555071						
25	October	-5696.342804	-6223	-6749	526	1.084525149						
26	4th Quarter	-23910.35369	-25499	-27086	1587	1.062237735						
27	November	-8906.880598	-9632	-10357	725	1.075269934						
28	December	-9307.130289	-9644	-9980	336	1.034840315						
29												
30												
31												



Cell References:

```
=OLAPivotTable($B$1,$B$2,$B$3:$E$3,$B$4:$E$4,$B$5:$E$5,
$B$6:$E$6,$B$7:$E$7,$B$11,FALSE,FALSE,FALSE)
```

- $B$1– the Database name or the PowerExcel connection name/cell reference that contains the name of the Database or the PowerExcel connection, i.e., **USING_OLATION**
- $B$2 – the Cube in the Database, i.e., **SALES** Cube
- $B$3:$E$3 – the ***Amount*** Member in the *SALES Measure* Dimension [Filter reference]
- $B$4:$E$4 – the ***Variance*** Member in the *Version* Dimension [Filter reference]
- $B$5:$E$5 – the ***World*** Member in the *Region* Dimension [Filter reference]
- $B$6:$E$6 – the Range relevant to the *Account* Dimension [Column reference]. This will dictate what Members will be displayed along Columns.
- $B$7:$E$7 – the Range relevant to the *Month* Dimension [Row reference]. This will dictate what Members will be displayed along Rows.
- $B$11 – this the cell location where the OLAPivotTable will start to bring in data.

- FALSE – the checkbox Constrain Empty Rows is disabled
- FALSE – the checkbox Delete Removed Rows is disabled
- FALSE – the checkbox Expandable Members is disabled (under development)

- Now, delete a cell value corresponding to a fact data, for example value in cell B12. Notice that once you refresh the PowerExcel Slice, the value will be returned in the cell.
- Next, delete ALL fact data within the PowerExcel Slice; once again, upon hitting the Refresh button or F9, all the data will be returned in the Slice.

10.OLAPowerQuery

Function Description: The PowerExcel Power Query Table dynamically creates a Slice in Excel in a powerful format that leverages Excel's own Power Query capabilities. In sum, it gives the user the ability to apply any queries to the table, apply numerous filters, easily reorder the table, and record/automate/undo changes, among other capabilities.

The important thing to take note of when using the PowerExcel Power Query Table is that it is updated using the **Refresh** button found in the PowerExcel tab of the Excel ribbon. The F9 key WILL NOT refresh a PowerExcel Slice that has the Power Query Table as the Slice type.

Syntax: OLAPowerQuery(Connection,Cube,Dimension1,Dimension2, Dimension3,…, DimensionN,Table_ExternalData_1)

Connection: The PowerExcel connection which contains the information about the Olation server URL and the source database name.

Cube: The name of the source/target Cube; or enter the cell reference that contains the name of the source or target Cube you wish to establish connection to.

Dimension1 to DimensionN: The related Dimension references.

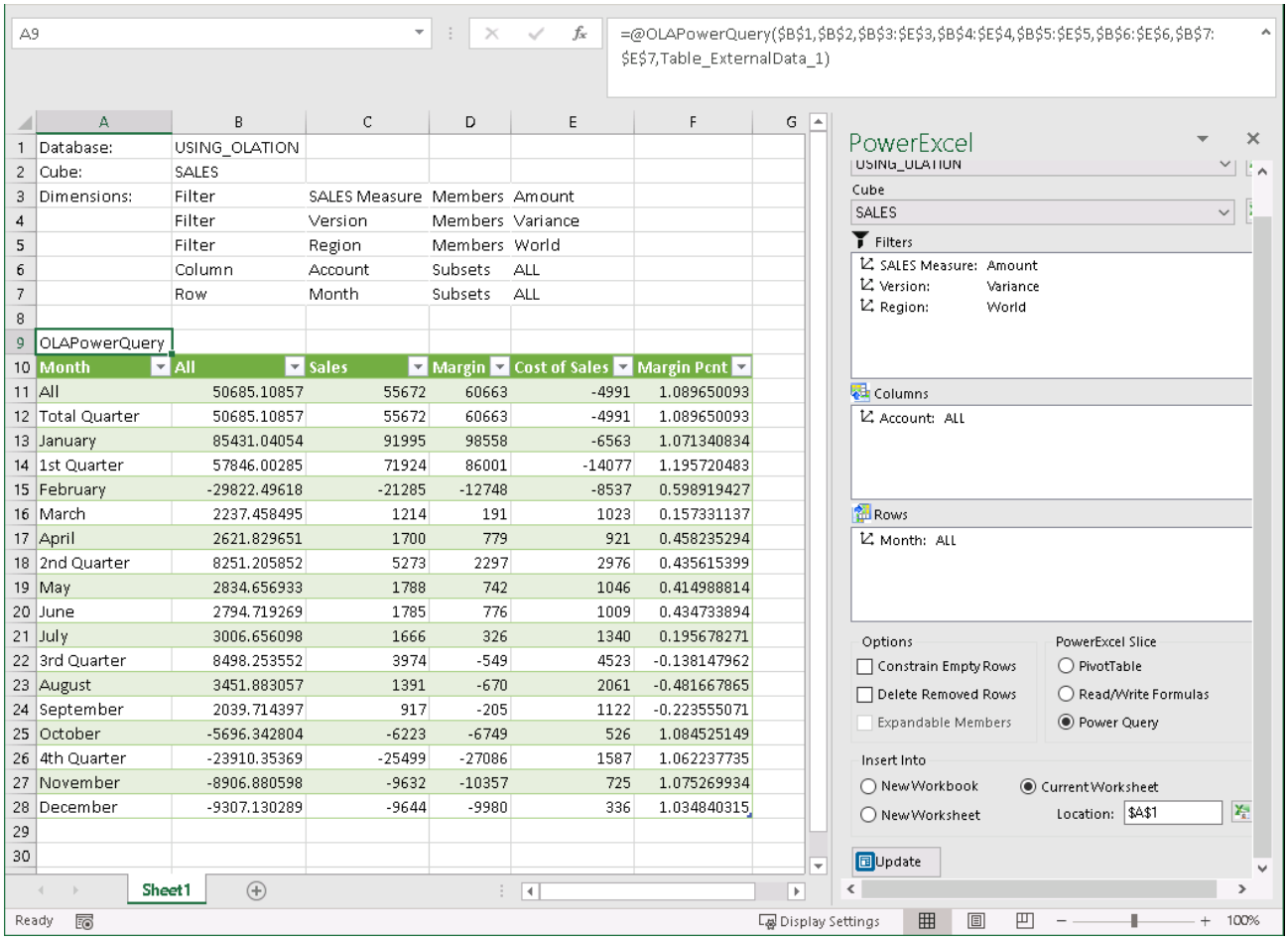
Table_ExternalData_1: as referenced by Excel, the range of cells where the data appears.

Remarks:

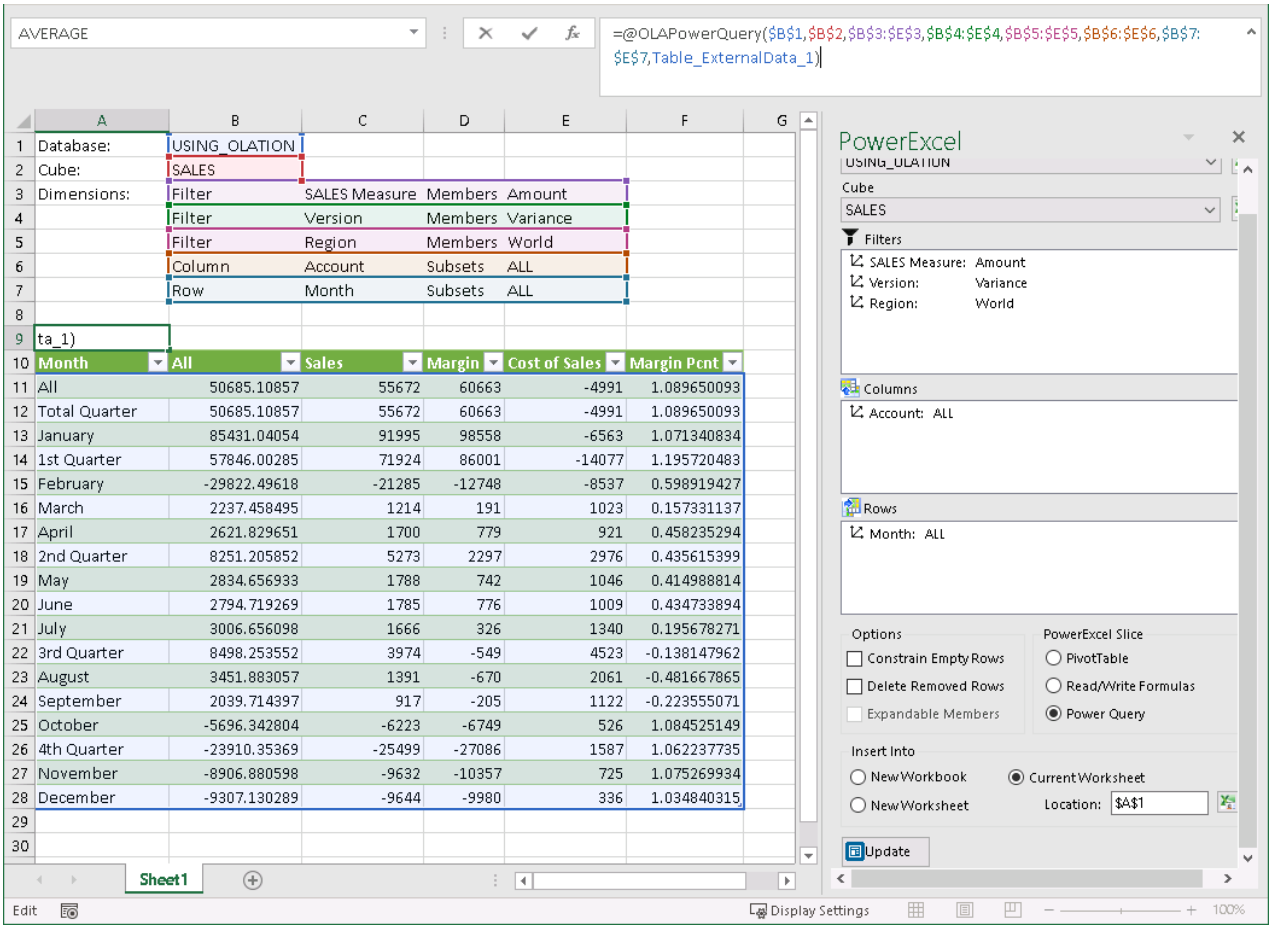
- The PowerExcel Connection must exist.
- The Olation Web Service must be running.
- The Database must be opened and running in the specified server.
- The 'Connection', 'Cube' and 'Dimension' parameters are compulsory
- Table_ExternalData_1 is compulsory.

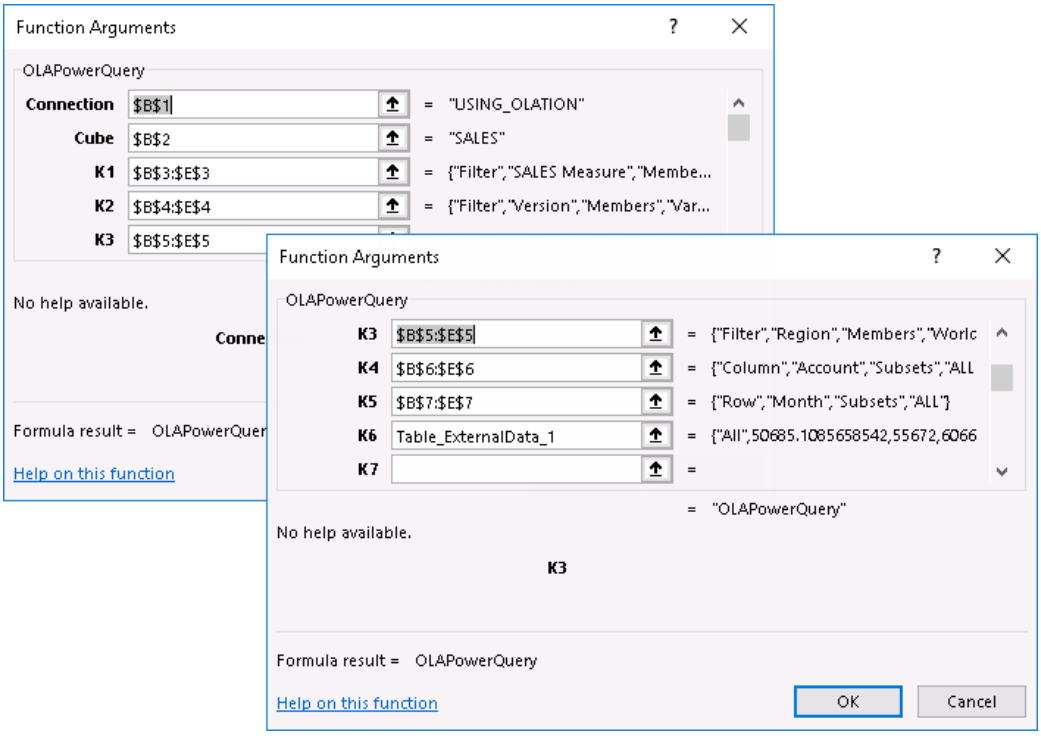
Example: OLAPowerQuery(Connection,Cube, Dimension1, Dimension2, Dimension3,…, DimensionN,Table_External_Data_1)

The Example Slice below shows a PowerExcel PowerQuery Table. Let us look at the corresponding cell references:



- Click on the **OLAPowerQuery** formula function on cell A9 then click on the Formula bar (in the screenshot below, the cursor is placed at the end of the formula).


AVERAGE
=@OLAPowerQuery($B$1,$B$2,$B$3:$E$3,$B$4:$E$4,$B$5:$E$5,$B$6:$E$6,$B$7:$E$7,Table_ExternalData_1)
Database: USING_OLATION
Cube: SALES
Dimensions: Filter SALES Measure Members Amount
Filter Version Members Variance
Filter Region Members World
Column Account Subsets ALL
Row Month Subsets ALL
Month All Sales Margin Cost of Sales Margin Pcnt
All 50685.10857 55672 60663 -4991 1.089650093
Total Quarter 50685.10857 55672 60663 -4991 1.089650093
January 85431.04054 91995 98558 -6563 1.071340834
1st Quarter 57846.00285 71924 86001 -14077 1.195720483
February -29822.49618 -21285 -12748 -8537 0.598919427
March 2237.458495 1214 191 1023 0.157331137
April 2621.829651 1700 779 921 0.458235294
2nd Quarter 8251.205852 5273 2297 2976 0.435615399
May 2834.656933 1788 742 1046 0.414988814
June 2794.719269 1785 776 1009 0.434733894
July 3006.656098 1666 326 1340 0.195678271
3rd Quarter 8498.253552 3974 -549 4523 -0.138147962
August 3451.883057 1391 -670 2061 -0.481667865
September 2039.714397 917 -205 1122 -0.223555071
October -5696.342804 -6223 -6749 526 1.084525149
4th Quarter -23910.35369 -25499 -27086 1587 1.062237735
November -8906.880598 -9632 -10357 725 1.075269934
December -9307.130289 -9644 -9980 336 1.034840315
Sheet1
PowerExcel
Cube
SALES
Filters
SALES Measure: Amount
Version: Variance
Region: World
Columns
Account: ALL
Rows
Month: ALL
Options
Constrain Empty Rows
Delete Removed Rows
Expandable Members
PowerExcel Slice
PivotTable
Read/Write Formulas
Power Query
Insert Into
New Workbook
New Worksheet
Current Worksheet
Location: $A$1
Update
Edit
Display Settings
100%


Function Arguments
OLAPowerQuery
Connection $B$1 = "USING_OLATION"
Cube $B$2 = "SALES"
K1 $B$3:$E$3 = {"Filter","SALES Measure","Membe...
K2 $B$4:$E$4 = {"Filter","Version","Members","Var...
K3 $B$5:$E$5
No help available.
Formula result = OLAPowerQuer
Help on this function
Function Arguments
OLAPowerQuery
K3 $B$5:$E$5 = {"Filter","Region","Members","Worlc
K4 $B$6:$E$6 = {"Column","Account","Subsets","ALL
K5 $B$7:$E$7 = {"Row","Month","Subsets","ALL"}
K6 Table_ExternalData_1 = {"All",50685.1085658542,55672,6066
K7 =
= "OLAPowerQuery"
No help available.
K3
Formula result = OLAPowerQuery
Help on this function
OK
Cancel

Cell References:

```
=OLAPowerQuery($B$1,$B$2,$B$3:$E$3,$B$4:$E$4,$B$5:$E$5,
$B$6:$E$6,$B$7:$E$7, Table_ExternalData_1)
```

- $B$1– the Database name or the PowerExcel connection name/cell reference that contains the name of the Database or the PowerExcel connection, i.e., **USING_OLATION**
- $B$2 – the Cube in the Database, i.e., **SALES** Cube
- $B$3:$E$3 – the ***Amount*** Member in the *SALES Measure* Dimension [Filter reference]
- $B$4:$E$4 – the ***Variance*** Member in the *Version* Dimension [Filter reference]
- $B$5:$E$5 – the ***World*** Member in the *Region* Dimension [Filter reference]
- $B$6:$E$6 – the Range relevant to the *Account* Dimension [Column reference]. This will dictate what Members will be displayed along the Columns.
- $B$7:$E$7 – the Range relevant to the *Month* Dimension [Row reference]. This will dictate what Members will be displayed along the Rows.
- Table_ExternalData_1 – the Range (aka, Table) where the data will appear.

- Delete a cell value corresponding to a fact data, for example the value in Cell B12. Press **F9**. Notice that the cell stays blank.
- Next, click the **Refresh** button found in the PowerExcel Tab of the Excel ribbon. The table data is now updated/refreshed, and the value re-appears on cell B12.
  This confirms that the F9 key does not work on the PowerQuery Table. **Use the REFRESH button when updating the PowerExcel PowerQuery Slice**.
- Next, try to delete ALL fact data within the PowerExcel Slice (i.e., **B11:F28**), and click the **Refresh button** found along the PowerExcel Tab of the Excel ribbon. The values return—which shows the dynamic connection to the source data.

11.OLARead

Description: PowerExcel's Read Formula function reads a value from a specific multidimensional data point in the PowerExcel model and returns it to a specified cell in an Excel worksheet. As opposed to PowerExcel ReadWrite formula, this OLARead function does not have a writeback capability, hence it can only 'Read' specific data from the source database.

Syntax: OLARead(Connection,Cube,Member1,Member2,…,MemberN)

Connection: The PowerExcel connection that contains the information about the Olation Server URL and the source database name.

Cube: The name of the source/target Cube; or the cell reference that contains the name of the source or target Cube you wish to establish a connection to.

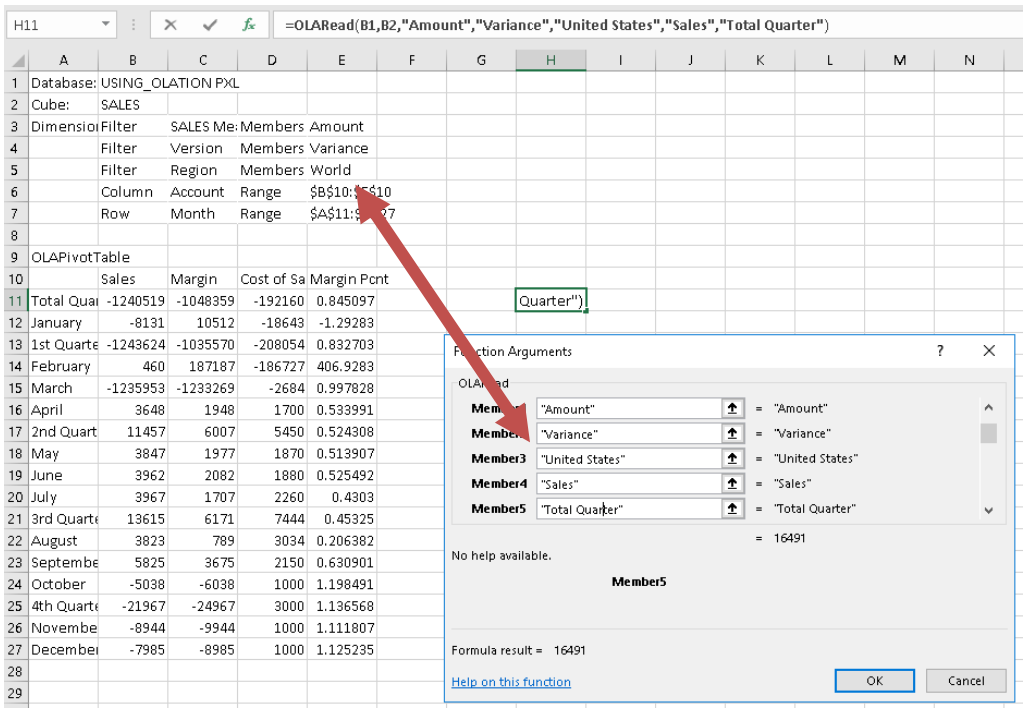
Member1 to MemberN: The related Member references.

Remarks:

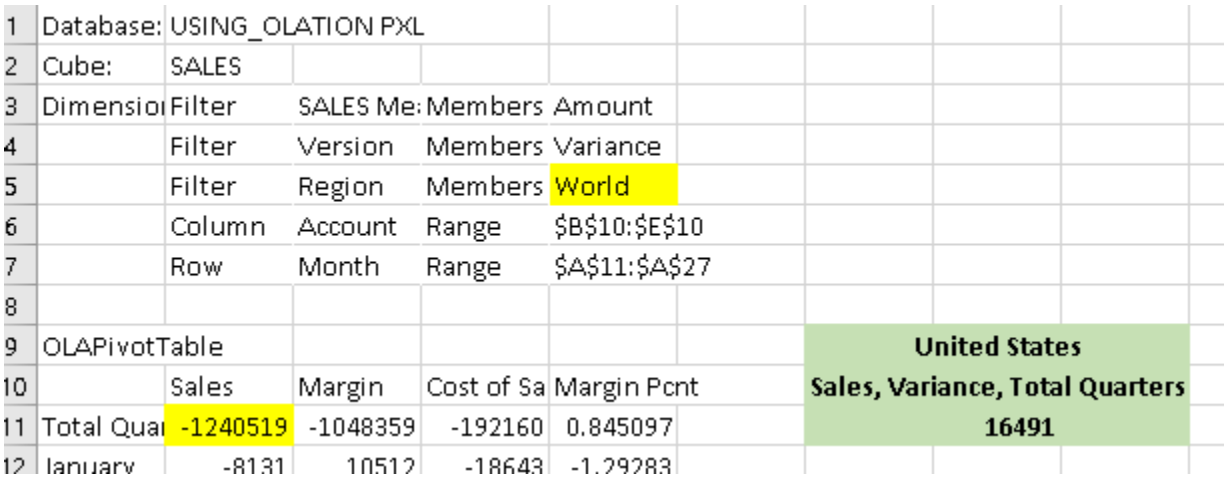
- The PowerExcel Connection must exist.
- The Olation Web Service must be running.
- The Database must be opened and running in the specified server.
- The 'Connection', 'Cube' and 'Member' parameters are compulsory.

Example1: OLARead(Connection,Cube,Member1,Member2,…,MemberN)

- Using an existing Slice, select a cell to the right of the field of data, e.g., Cell **H11**.
- In the Excel formula bar, click on the Insert Function symbol (***fx***). The Insert Function window will appear.
- In the **Or select a category drop-down**, select **PowerExcel.ExcelFunctions**.
- Select **OLARead** Click **OK**.
- For Connection, you can reference Cell **B1** from the sample Slice (or type B1); next, reference the Cube (Cell **B2**).
- Type in the names of Dimension Members:
    - For Sales Measure, type **Amount**.
    - For Version, type **Variance**.
    - For Region, type **United States** (Note: this is the key point of difference with the existing Slice, which shows World—as shown by the arrow in the following image).
    - For Account, type **Sales**.
    - For Month, type **Total Quarter**.



- Click **OK** in the Functions Arguments window, then hit F9 to update the worksheet. The figure **164911**—which is the precise data point described by the OLARead function--appears in the cell, as shown in the following image (labeled and highlighted green, to compare to the *World* figure for the same other Member details, highlighted yellow).

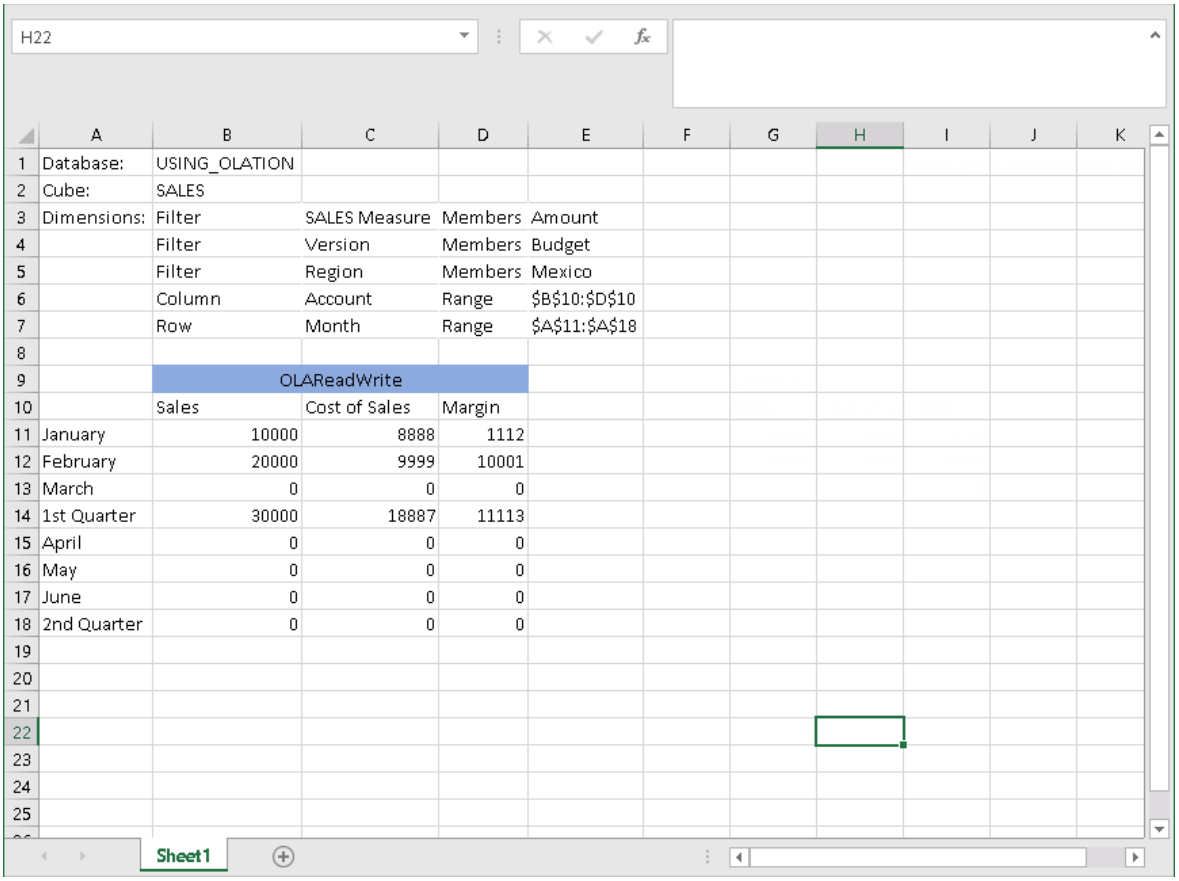


Example2: (OLAReadWrite VS OLARead)

Next we will compare how PowerExcel's OLAReadWrite (descried in the following sections) works vs an OLARead Function.

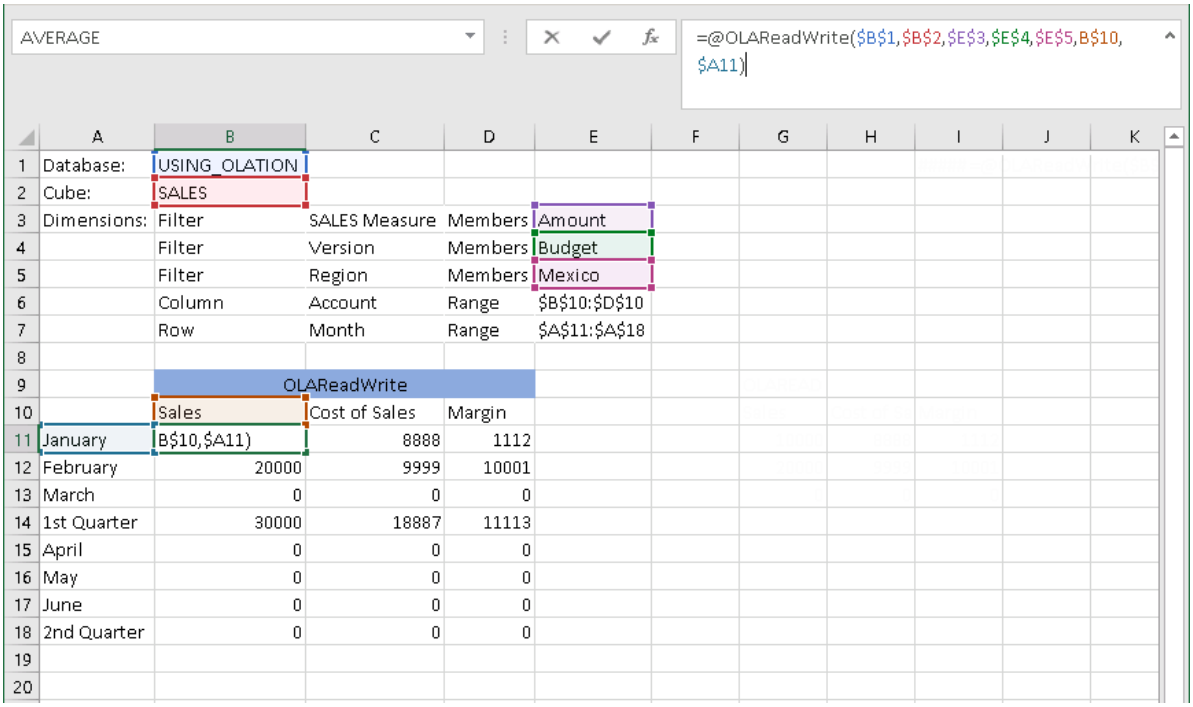
- Create a PowerExcel ReadWrite Slice with the orientation of your choosing (e.g., the following image will serve as a example).

Note: We placed a heading (in Row 9), "OLAReadWrite", at the top of the sample table so we can identify that the cells below are governed by this type of formula.



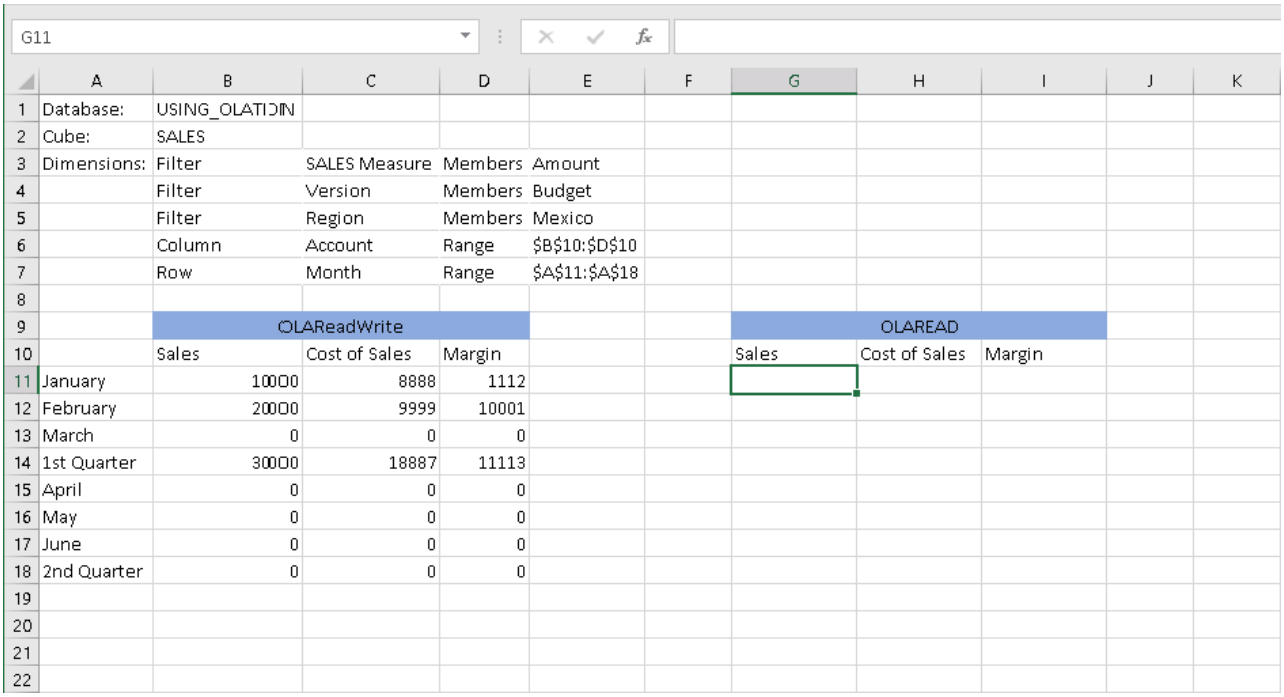
	A	B	C	D	E
1	Database:	USING_OLATION			
2	Cube:	SALES			
3	Dimensions:	Filter	SALES Measure	Members	Amount
4		Filter	Version	Members	Budget
5		Filter	Region	Members	Mexico
6		Column	Account	Range	$B$10:$D$10
7		Row	Month	Range	$A$11:$A$18
8					
9		OLAReadWrite			
10		Sales	Cost of Sales	Margin	
11	January	10000	8888	1112	
12	February	20000	9999	10001	
13	March	0	0	0	
14	1st Quarter	30000	18887	11113	
15	April	0	0	0	
16	May	0	0	0	
17	June	0	0	0	
18	2nd Quarter	0	0	0	

- Note that in a OLAReadWrite Slice, each cell contains an individual formula, as in the following image (Cell B11 has been clicked on). As we will see, an **OLARead Formula function** also appertains to an individual cell.



	A	B	C	D	E
1	Database:	USING_OLATION			
2	Cube:	SALES			
3	Dimensions:	Filter	SALES Measure	Members	Amount
4		Filter	Version	Members	Budget
5		Filter	Region	Members	Mexico
6		Column	Account	Range	$B$10:$D$10
7		Row	Month	Range	$A$11:$A$18
8					
9		OLAReadWrite			
10		Sales	Cost of Sales	Margin	
11	January	B$10,$A11)	8888	1112	
12	February	20000	9999	10001	
13	March	0	0	0	
14	1st Quarter	30000	18887	11113	
15	April	0	0	0	
16	May	0	0	0	
17	June	0	0	0	
18	2nd Quarter	0	0	0	
19					
20					

- Next to create an OLARead formula: in Cells **G10**, **H10** and **I10** and type, respectively, *Sales*, *Cost of Sales* and *Margin* (make sure to put a **single quote** at the beginning of each). In the screenshot, **OLARead** has been typed above and highlighted in blue.

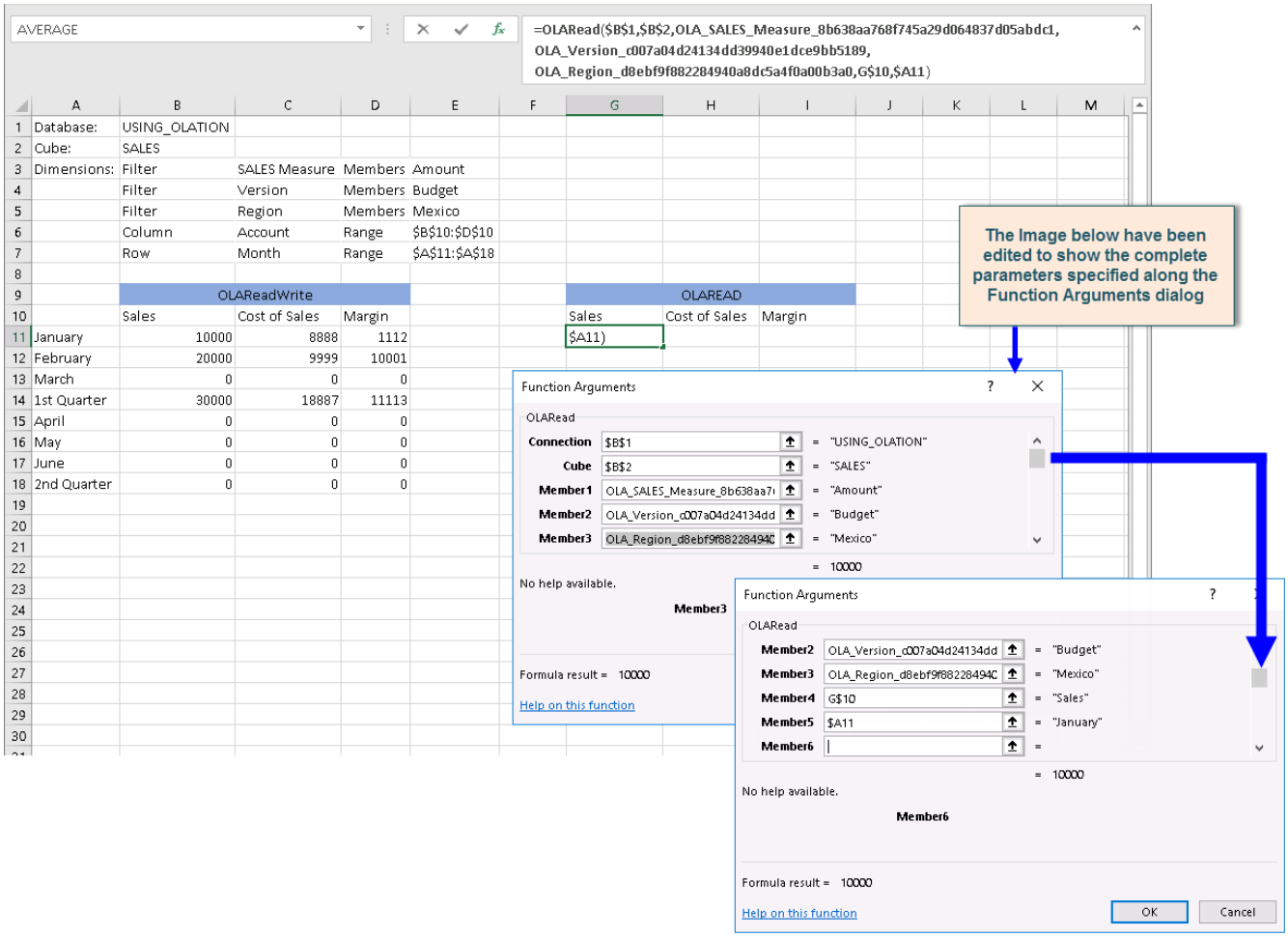


	A	B	C	D	E	F	G	H	I	J	K
1	Database:	USING_OLATION									
2	Cube:	SALES									
3	Dimensions:	Filter	SALES Measure	Members	Amount						
4		Filter	Version	Members	Budget						
5		Filter	Region	Members	Mexico						
6		Column	Account	Range	$B$10:$D$10						
7		Row	Month	Range	$A$11:$A$18						
8											
9		OLAReadWrite					OLAREAD				
10		Sales	Cost of Sales	Margin			Sales	Cost of Sales	Margin		
11	January	10000	8888	1112							
12	February	20000	9999	10001							
13	March	0	0	0							
14	1st Quarter	30000	18887	11113							
15	April	0	0	0							
16	May	0	0	0							
17	June	0	0	0							
18	2nd Quarter	0	0	0							
19											
20											
21											
22											

- Define the **OLARead** formula: in Cell **G11** click the **Function** button beside the formula bar. In the Insert Function dialog that appears, choose **PowerExcel.ExcelFunctions** as the category, select **OLARead** from the function list and click **OK**.

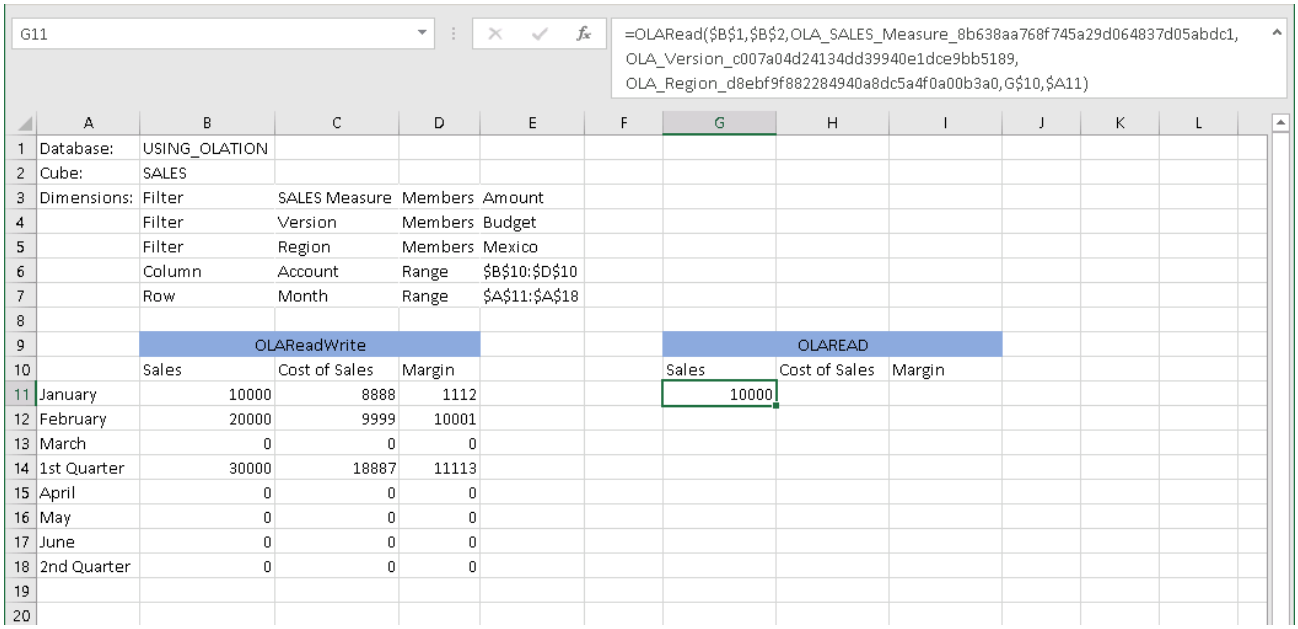
- In the Function Arguments dialog box, define the parameters:
    - Click on Connection field and then click on cell B1 (which is the cell reference for *USING_OLATION* database). Use an Absolute reference so the connection reference appears as **$B$1**.
    - Click on the Cube field then click on cell B2 (the cell reference for the *SALES* cube) Again, use absolute an reference, thus **$B$2**.
    - Click on the Member 1 field, then click on cell **E3** (the cell reference for the Filter Member *Amount* in the *SALES Measure* Dimension).
    - Click on the Member 2 field then click on cell **E4** (the cell reference for the Filter Member *Budget* in the *Version* Dimension).
    - Click on the Member 3 field then click on cell **E5** (the cell reference for the Filter Member *Mexico* in the *Region* Dimension).
    - Click on the Member 4 field, then click on cell **G10** (the cell reference for the Column Member *Sales* of the *Account* Dimension); use the absolute row reference, thus **G$10**.
    - Click on the Member 5 field, then click on cell **A11** (the cell reference for the Row Member *January* in the *Month* Dimension); use the absolute column reference, so the Member 5 reference **$A11**.

      Note: Notice that the corresponding Database, Cube and Member names appears in each corresponding field, as shown in the following image.



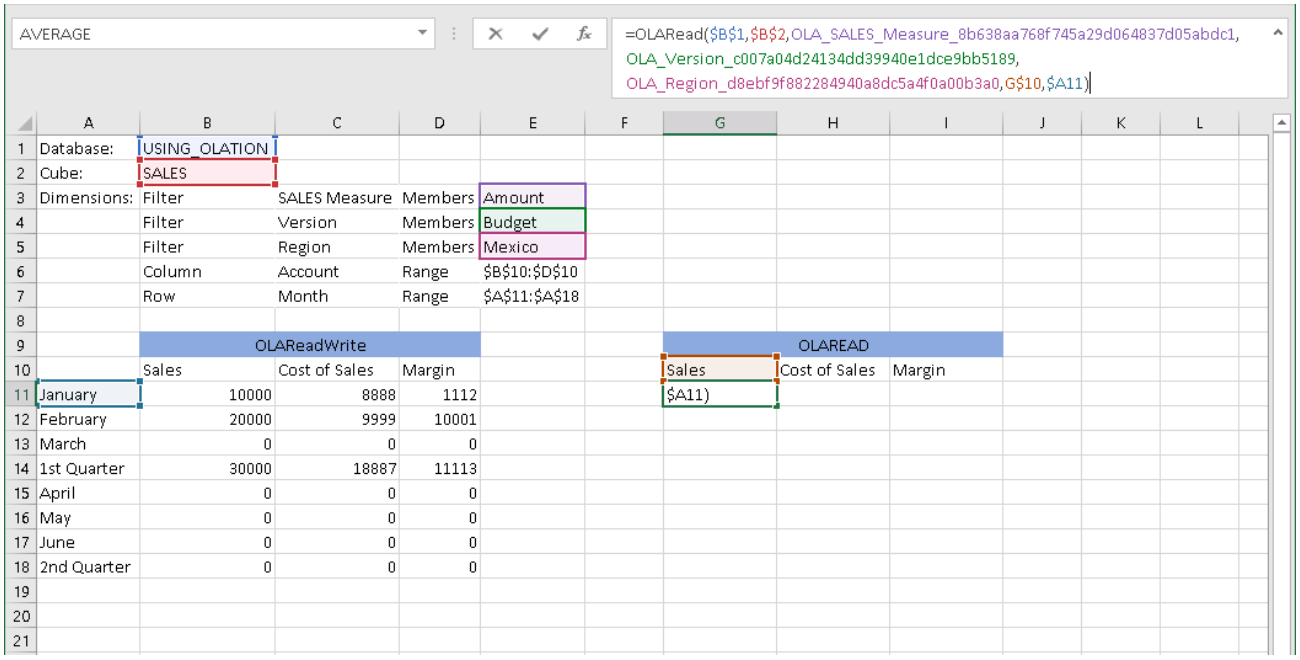
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Database:	USING_OLATION											
2	Cube:	SALES											
3	Dimensions:	Filter	SALES Measure	Members	Amount								
4		Filter	Version	Members	Budget								
5		Filter	Region	Members	Mexico								
6		Column	Account	Range	$B$10:$D$10								
7		Row	Month	Range	$A$11:$A$18								
8													
9		OLAReadWrite					OLAREAD						
10		Sales	Cost of Sales	Margin			Sales	Cost of Sales	Margin				
11	January	10000	8888	1112			$A11)						
12	February	20000	9999	10001									
13	March	0	0	0									
14	1st Quarter	30000	18887	11113									
15	April	0	0	0									
16	May	0	0	0									
17	June	0	0	0									
18	2nd Quarter	0	0	0									
19													
20													
21													
22													
23													
24													
25													
26													
27													
28													
29													
30													

- Click **OK**. Then click the **Refresh** button along the PowerExcel Tab of the Excel ribbon. Notice that the formula returns the value **10000**: it returns the same value as in OLAReadWrite section of the Slice (Cell B11).

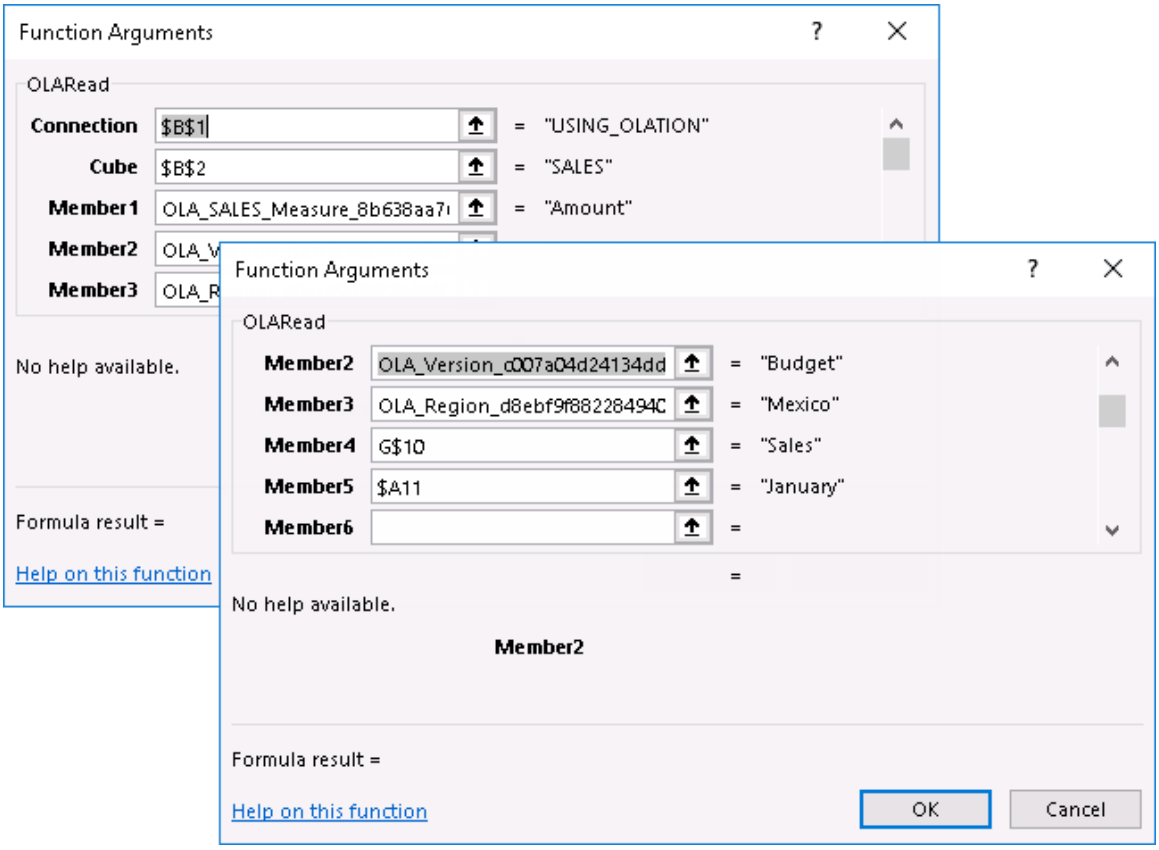


	A	B	C	D	E	F	G	H	I	J	K	L
1	Database:	USING_OLATION										
2	Cube:	SALES										
3	Dimensions:	Filter	SALES Measure	Members	Amount							
4		Filter	Version	Members	Budget							
5		Filter	Region	Members	Mexico							
6		Column	Account	Range	$B$10:$D$10							
7		Row	Month	Range	$A$11:$A$18							
8												
9		OLAReadWrite					OLAREAD					
10		Sales	Cost of Sales	Margin			Sales	Cost of Sales	Margin			
11	January	10000	8888	1112			10000					
12	February	20000	9999	10001								
13	March	0	0	0								
14	1st Quarter	30000	18887	11113								
15	April	0	0	0								
16	May	0	0	0								
17	June	0	0	0								
18	2nd Quarter	0	0	0								
19												
20												

- Now, click on the returned value of the **OLARead** formula of **10000** (cell G11), then click on the formula bar (notice that the cursor is at the end of the formula). This will show the corresponding cell references of the formula.



	A	B	C	D	E	F	G	H	I	J	K	L
1	Database:	USING_OLATION										
2	Cube:	SALES										
3	Dimensions:	Filter	SALES Measure	Members	Amount							
4		Filter	Version	Members	Budget							
5		Filter	Region	Members	Mexico							
6		Column	Account	Range	$B$10:$D$10							
7		Row	Month	Range	$A$11:$A$18							
8												
9		OLAReadWrite					OLAREAD					
10		Sales	Cost of Sales	Margin			Sales	Cost of Sales	Margin			
11	January	10000	8888	1112			$A11)					
12	February	20000	9999	10001								
13	March	0	0	0								
14	1st Quarter	30000	18887	11113								
15	April	0	0	0								
16	May	0	0	0								
17	June	0	0	0								
18	2nd Quarter	0	0	0								
19												
20												
21												

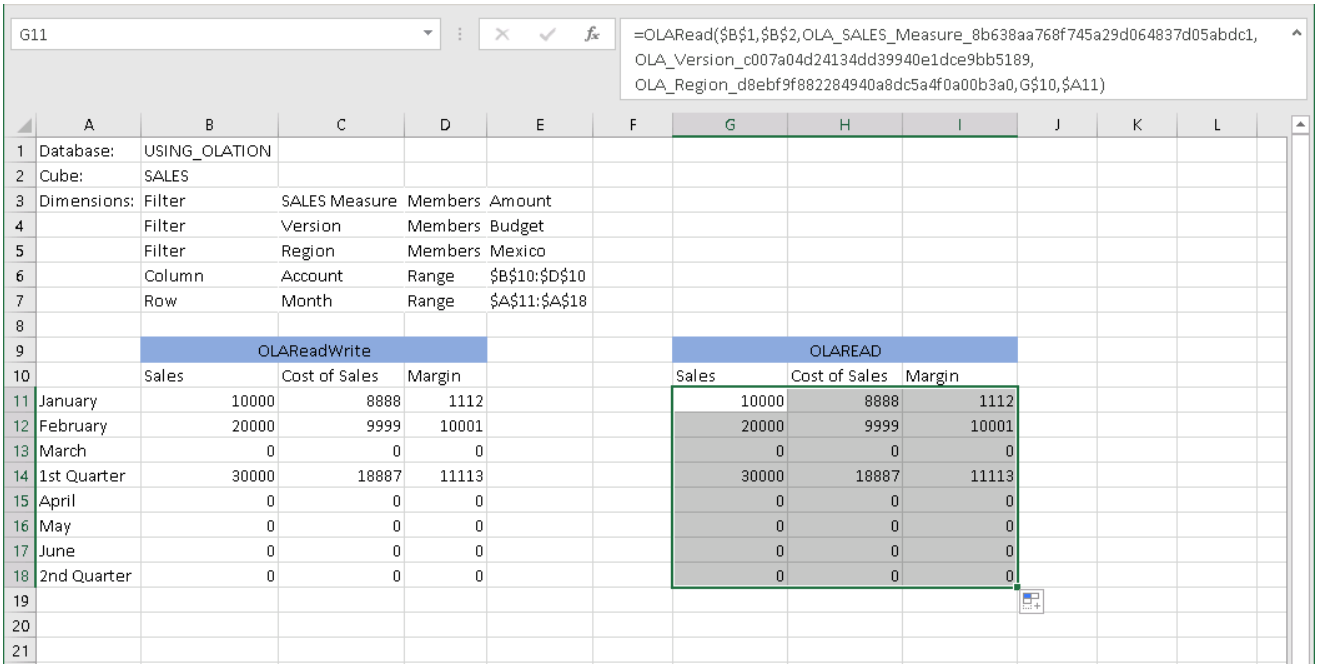


Cell References:

```
=OLARead($B$1,$B$2,OLA_SALES_Measure_8b638aa768f745a29d064837d05abdc1,OLA_Version_c007a04d24134dd39940e1dce9bb5189,OLA_Region_d8ebf9f882284940a8dc5a4f0a00b3a0,G$10,$A11)
```

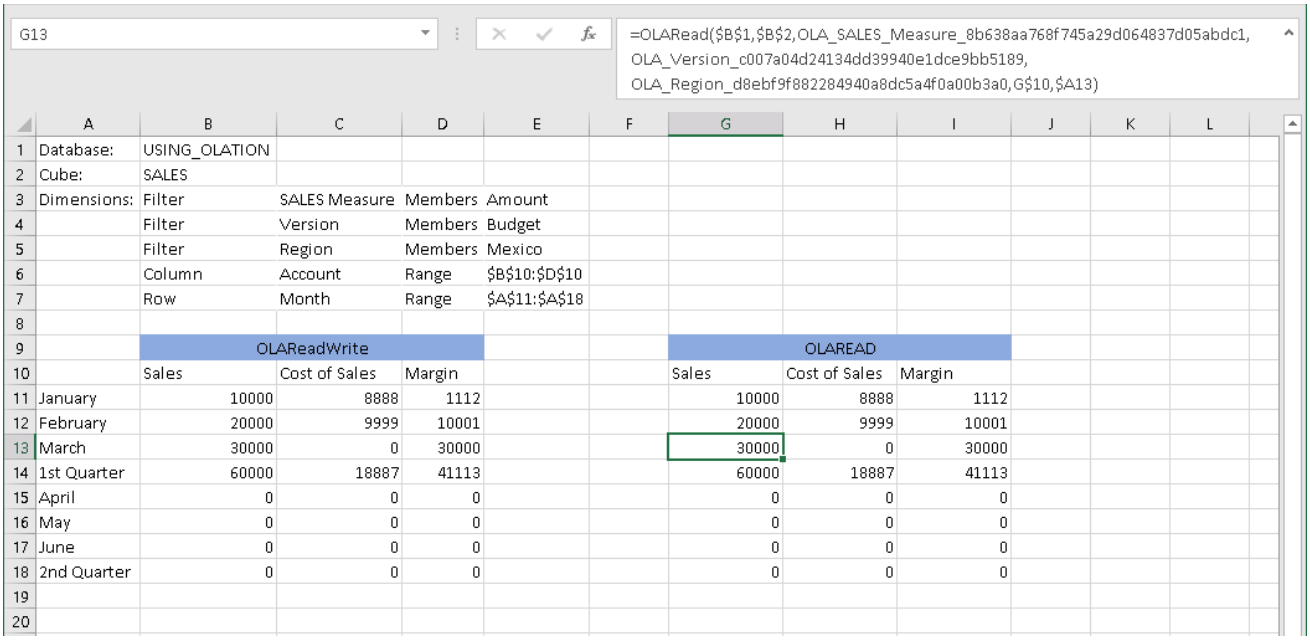
- $B$1– the Database name or the PowerExcel connection name/cell reference that contains the name of the Database or the PowerExcel connection, i.e., **USING_OLATION**
- $B$2 – the Cube in the Database, i.e., **SALES** Cube
- OLA_SALES_Measure_8b638aa768f745a29d064837d05abdc1 – the ***Amount*** Member in the *SALES Measure* Dimension [Filter reference]
- OLA_Version_c007a04d24134dd39940e1dce9bb5189 – the ***Budget*** Member in the *Version* Dimension [Filter reference]
- OLA_Region_d8ebf9f882284940a8dc5a4f0a00b3a0 – the ***Mexico*** Member in the *Region* Dimension [Filter reference]
- G$10 – the Column Member reference ***Sales*** from the *Account* Dimension [Column reference].
- $A11 – the Row Member reference ***January*** from the *Month* Dimension [Row reference].

- Copy the formula across and down to Cell **I18**. Click the **Refresh** button along the PowerExcel Tab. Notice that all cells return the same data values as in the PowerExcel ReadWrite section of the Slice.



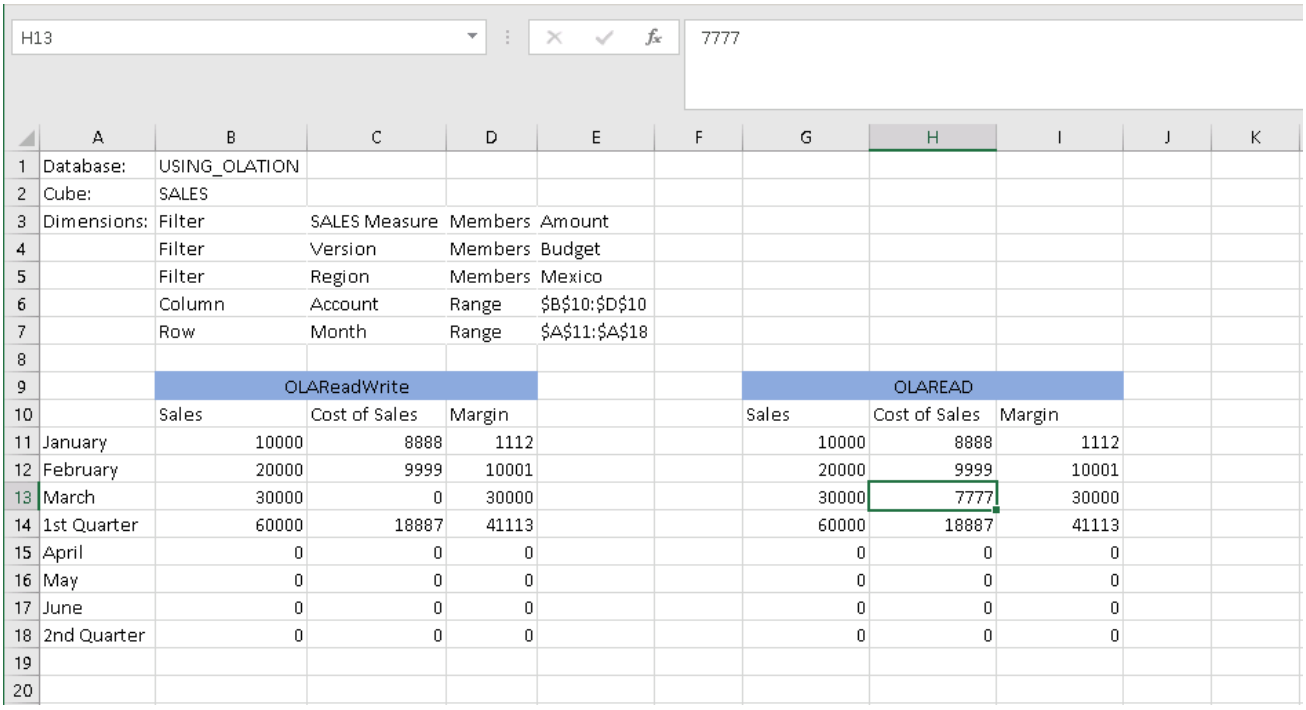
	A	B	C	D	E	F	G	H	I	J	K	L
1	Database:	USING_OLATION										
2	Cube:	SALES										
3	Dimensions:	Filter	SALES Measure	Members	Amount							
4		Filter	Version	Members	Budget							
5		Filter	Region	Members	Mexico							
6		Column	Account	Range	$B$10:$D$10							
7		Row	Month	Range	$A$11:$A$18							
8												
9		OLAReadWrite					OLAREAD					
10		Sales	Cost of Sales	Margin			Sales	Cost of Sales	Margin			
11	January	10000	8888	1112			10000	8888	1112			
12	February	20000	9999	10001			20000	9999	10001			
13	March	0	0	0			0	0	0			
14	1st Quarter	30000	18887	11113			30000	18887	11113			
15	April	0	0	0			0	0	0			
16	May	0	0	0			0	0	0			
17	June	0	0	0			0	0	0			
18	2nd Quarter	0	0	0			0	0	0			
19												
20												
21												

- Next in Cell **B13** of the ReadWrite Slice enter a new sales value—e.g., **30000**. Press **Enter** then hit the **Refresh** button along the PowerExcel Tab of the Excel ribbon. Notice that the aggregate values are subsequently updated in both the OLARead and the ReadWrite sections of the Slice.



	A	B	C	D	E	F	G	H	I	J	K	L
1	Database:	USING_OLATION										
2	Cube:	SALES										
3	Dimensions:	Filter	SALES Measure	Members	Amount							
4		Filter	Version	Members	Budget							
5		Filter	Region	Members	Mexico							
6		Column	Account	Range	$B$10:$D$10							
7		Row	Month	Range	$A$11:$A$18							
8												
9		OLAReadWrite					OLAREAD					
10		Sales	Cost of Sales	Margin			Sales	Cost of Sales	Margin			
11	January	10000	8888	1112			10000	8888	1112			
12	February	20000	9999	10001			20000	9999	10001			
13	March	30000	0	30000			30000	0	30000			
14	1st Quarter	60000	18887	41113			60000	18887	41113			
15	April	0	0	0			0	0	0			
16	May	0	0	0			0	0	0			
17	June	0	0	0			0	0	0			
18	2nd Quarter	0	0	0			0	0	0			
19												
20												

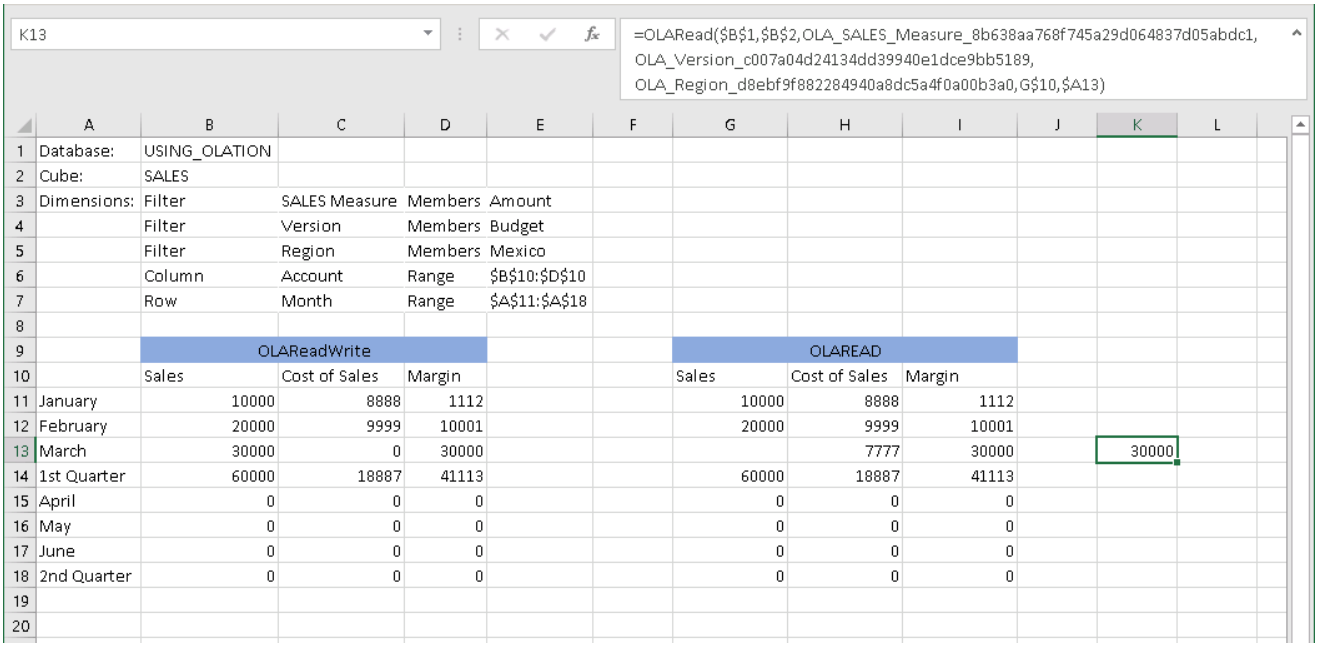
- Now try entering in the Read Slice (OLARead Slice) and observe the results. For example, in Cell **H13** enter a new *Cost of Sales* value—e.g., **7777**. This cell corresponds to the *Budget, Cost of Sales* value for the month of *March* and for the region *Mexico*. Press **Enter** then click the **Refresh** button.



H13 | 7777

	A	B	C	D	E	F	G	H	I	J	K
1	Database:	USING_OLATION									
2	Cube:	SALES									
3	Dimensions:	Filter	SALES Measure	Members	Amount						
4		Filter	Version	Members	Budget						
5		Filter	Region	Members	Mexico						
6		Column	Account	Range	$B$10:$D$10						
7		Row	Month	Range	$A$11:$A$18						
8											
9		OLAReadWrite					OLAREAD				
10		Sales	Cost of Sales	Margin			Sales	Cost of Sales	Margin		
11	January	10000	8888	1112			10000	8888	1112		
12	February	20000	9999	10001			20000	9999	10001		
13	March	30000	0	30000			30000	7777	30000		
14	1st Quarter	60000	18887	41113			60000	18887	41113		
15	April	0	0	0			0	0	0		
16	May	0	0	0			0	0	0		
17	June	0	0	0			0	0	0		
18	2nd Quarter	0	0	0			0	0	0		
19											
20											

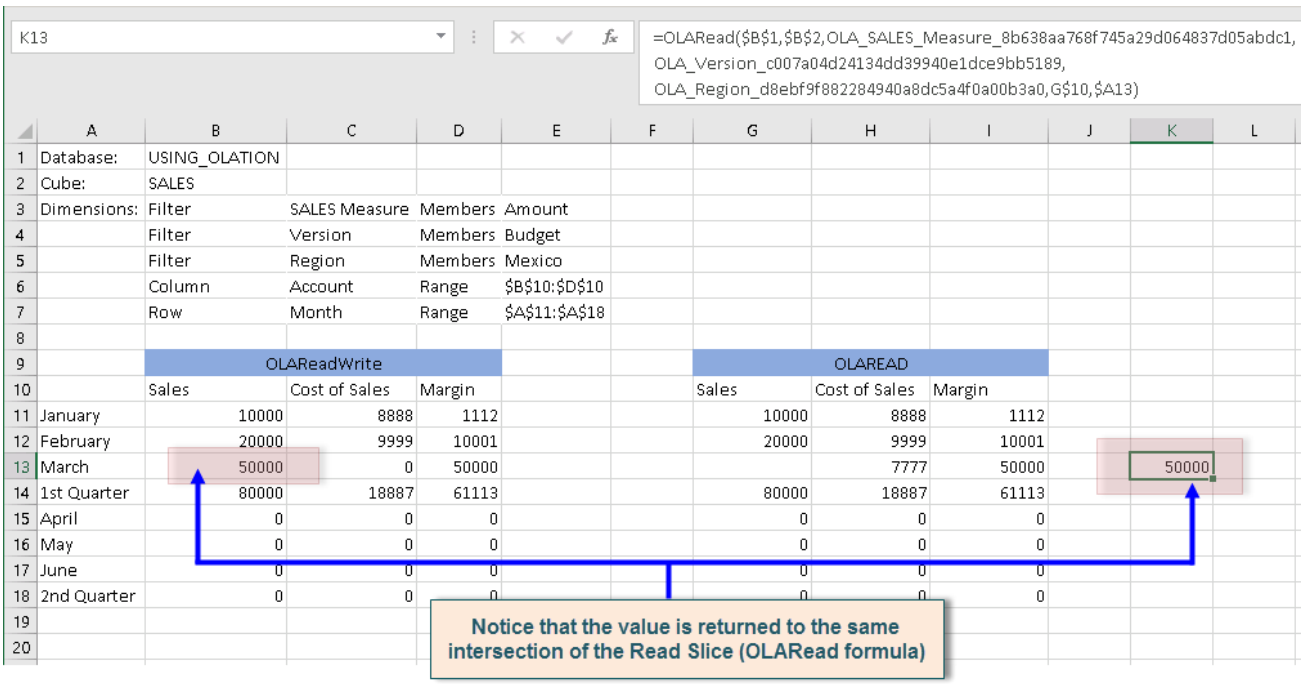
- Notice that although you can see 7777 appear on Cell H13, that value did not appear in the same intersection of the ReadWrite section of the Slice (Cell **C13**). This means that the newly entered *Cost of Sales* value is not committed/saved to the source/target database.
- Additionally, clicking on Cell H13, notice that the OLARead formula was overwritten and replaced with the numeric value of 7777. (It is worth noting: in the ReadWrite section of the Slice, however, typing a number on a Detail intersections will not remove the OLAReadWrite function.)
- Next, to move the OLARead formula to a different cell: for example, move the March Sales Read formula (cell **G13**) to a different cell—e.g., Cell **K13**. Although the formula was moved, the cell references remained the same (which is standard Excel behavior).



K13 =OLARead($B$1,$B$2,OLA_SALES_Measure_8b638aa768f745a29d064837d05abdc1, OLA_Version_c007a04d24134dd39940e1dce9bb5189, OLA_Region_d8ebf9f882284940a8dc5a4f0a00b3a0,G$10,$A13)

	A	B	C	D	E	F	G	H	I	J	K	L
1	Database:	USING_OLATION										
2	Cube:	SALES										
3	Dimensions:	Filter	SALES Measure	Members	Amount							
4		Filter	Version	Members	Budget							
5		Filter	Region	Members	Mexico							
6		Column	Account	Range	$B$10:$D$10							
7		Row	Month	Range	$A$11:$A$18							
8												
9		OLAReadWrite					OLAREAD					
10		Sales	Cost of Sales	Margin			Sales	Cost of Sales	Margin			
11	January	10000	8888	1112			10000	8888	1112			
12	February	20000	9999	10001			20000	9999	10001			
13	March	30000	0	30000				7777	30000		30000	
14	1st Quarter	60000	18887	41113			60000	18887	41113			
15	April	0	0	0			0	0	0			
16	May	0	0	0			0	0	0			
17	June	0	0	0			0	0	0			
18	2nd Quarter	0	0	0			0	0	0			
19												
20												

- Back in the ReadWrite section of the Slice, change the March Sales value at cell **B13** to **50000**. Press **Enter** then click the **Refresh**: the new value from the ReadWrite Slice is reflected at the same intersection point that is using the OLARead formula function (Cell K13).



K13 =OLARead($B$1,$B$2,OLA_SALES_Measure_8b638aa768f745a29d064837d05abdc1, OLA_Version_c007a04d24134dd39940e1dce9bb5189, OLA_Region_d8ebf9f882284940a8dc5a4f0a00b3a0,G$10,$A13)

	A	B	C	D	E	F	G	H	I	J	K	L
1	Database:	USING_OLATION										
2	Cube:	SALES										
3	Dimensions:	Filter	SALES Measure	Members	Amount							
4		Filter	Version	Members	Budget							
5		Filter	Region	Members	Mexico							
6		Column	Account	Range	$B$10:$D$10							
7		Row	Month	Range	$A$11:$A$18							
8												
9		OLAReadWrite					OLAREAD					
10		Sales	Cost of Sales	Margin			Sales	Cost of Sales	Margin			
11	January	10000	8888	1112			10000	8888	1112			
12	February	20000	9999	10001			20000	9999	10001			
13	March	50000	0	50000				7777	50000		50000	
14	1st Quarter	80000	18887	61113			80000	18887	61113			
15	April	0	0	0			0	0	0			
16	May	0	0	0			0	0	0			
17	June	0	0	0			0	0	0			
18	2nd Quarter	0	0	0			0	0	0			
19												
20												

Notice that the value is returned to the same intersection of the Read Slice (OLARead formula)

12.OLAReadWrite

Function Description: PowerExcel's Read/Write Formulas, when used as the means to bring data into a Slice, has the key advantage of returning business-model data (according to Dimensions selected and filtered) via individual, discrete cell-by-cell functions. In other words, each cell's value is governed by its own function (an OLAReadWrite function), rather than as part of a swath of cells, which is the case when the OLAP Pivot Table or the PowerQuery functions are used.

Syntax: OLAReadWrite (Connection, Cube, Member1, Member2,…,MemberN)

Connection: Enter the PowerExcel connection which contains the information about the Olation server URL and the source database name.

Cube: Enter the name of the source/target Cube; or enter the cell reference that contains the name of the source or target Cube you wish to establish connection to.

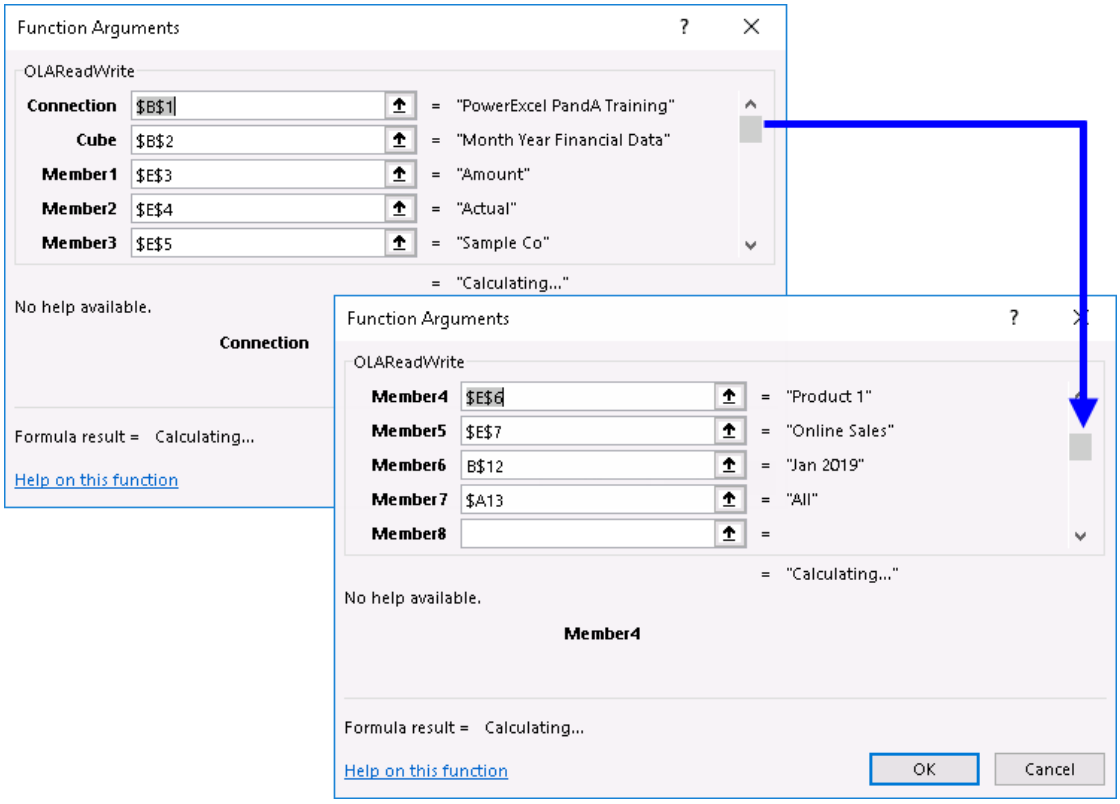
Member1 to MemberN: The related Member references.

Remarks:

- The PowerExcel Connection must exist.
- The Olation Web Service must be running.
- The Database must be opened and running on the specified server.
- The 'Connection', 'Cube' and 'Member' parameters are compulsory.

Example 1:

The function in this cell (reproduced below) has unique cell references; if you click in any other cell that returns values, you will see that the cell references are indeed different.



Cell References:

```
=OLAReadWrite($B$1,$B$2,$E$3,$E$4,$E$5,$E$6,$E$7,B$12,$A13)
```

The **Read/Write Formula** function points to—and returns the value from—a precise multidimensional data point in the PowerExcel model: thus, the formula in Cell G13 is returning a value from the intersection of (in this example):

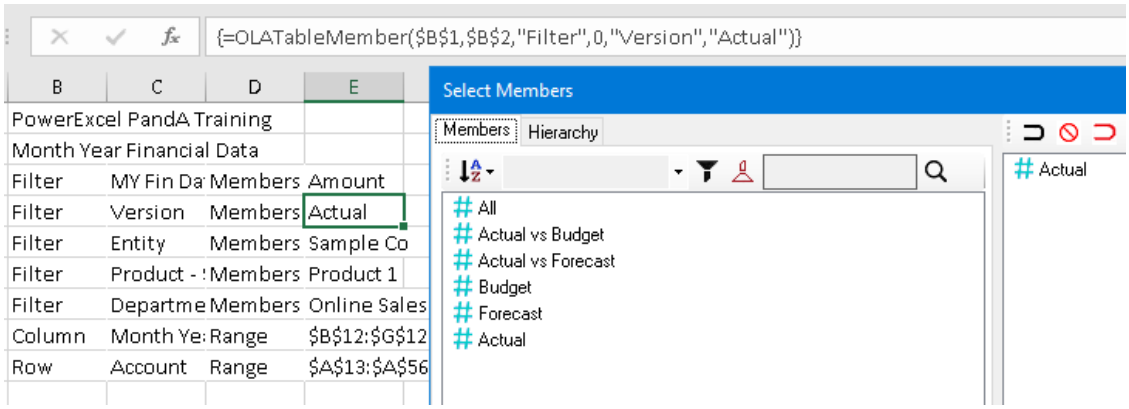
- $B$1 – the Database where the model is located, i.e., **PowerExcel PandA Training**
- $B$2 – the Cube in the Database, i.e., **Month Year Financial Data**
- $E$3 – the ***Amount*** Member in the *MY Fin Data Measure* Dimension
- $E$4 – the ***Actual*** Member in the *Version* Dimension
- $E$5 – the ***Sample Co*** Member in the *Entity* Dimension
- $E$6 – the ***Product 1*** Member in the *Product* – Services Dimension
- $E$7 – the ***Online Sales*** Member in the *Department* Dimension
- B$12 – ***Jan 2019*** from the *Month Year* Dimension [Column reference]
- $A13 – ***All*** from the *Account* Dimension [Row reference]

The interesting—and very important—fact is that that these cells are themselves "selectable"—meaning that either by using the PowerExcel pane to the right or by double-

click on, for example, those governed by an {OLATableMember…} function, you can select a different Member that will change all results in the field of data in Columns and Rows.

Example 2:

- As an example, you can double-click in Cell $E$4, and make a selection of a different version: if you do so and pick “Budget” (from the resulting PowerExcel Select Members window, as shown in the below image), then results for all the Members, including those in Columns and Rows, will show results for *Budget* rather than *Actual*.



	B	C	D	E
	PowerExcel PandA Training			
	Month Year Financial Data			
	Filter	MY Fin Da	Members	Amount
	Filter	Version	Members	Actual
	Filter	Entity	Members	Sample Co
	Filter	Product - !	Members	Product 1
	Filter	Departme	Members	Online Sales
	Column	Month Ye:	Range	$B$12:$G$12
	Row	Account	Range	$A$13:$A$56

For present, go back to (or keep the selection at) Actual. We will proceed what this “cell-by-cell return of values” enables us to do in the following steps.

- First delete Row 13, which shows data for the All Account Member—this is a formatting step, to show our data results more cleverly.

- Next, after the Rows have shifted upward 1 Row, click in Cell C13—*Feb 2019*, *Sales Income*.

- Highlight all cells across and down to G15 (Jun 2019, INCOME); “grab” those cells and drop them starting in Cell I13—the result will show as in the following image:

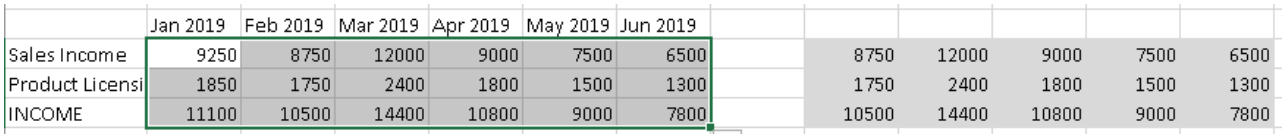
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Database:	PowerExcel PandA Training											
2	Cube:	Month Year Financial Data											
3	Dimensions:	Filter	MY Fin Da	Members	Amount								
4		Filter	Version	Members	Actual								
5		Filter	Entity	Members	Sample Co								
6		Filter	Product - !	Members	Product 1								
7		Filter	Departme	Members	Online Sales								
8		Column	Month Ye:	Range	$B$12:$G$12								
9		Row	Account	Range	$A$13:$A$56								
10													
11													
12		Jan 2019	Feb 2019	Mar 2019	Apr 2019	May 2019	Jun 2019						
13	Sales Income	9250							8750	12000	9000	7500	6500
14	Product Licensi	1850							1750	2400	1800	1500	1300
15	INCOME	11100							10500	14400	10800	9000	7800
16	DirectCosts	3515	3325	4560	3420	2850	2470						

- If you now double-click on Cell E4 and select *Budget*, then **hit F9 to update**—note, those cells that you moved to the right show all zeros (as in the next image)! That is because Excel maintained the references to the cells, and the selected Member in them, in their new position—and the *Budget* values for all parameters happen to be zero. In essence, you have proved that with PowerExcel, you can return values from a business model wherever you like, anywhere in a spreadsheet. This is enormously useful in creating precisely the report view you wish, from a multidimensional model of the sort PowerExcel features.

B13 =@OLAReadWrite($B$1,$B$2,$E$3,$E$4,$E$5,$E$6,$E$7,B$12,$A13)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Database:	PowerExcel PandA Training											
2	Cube:	Month Year Financial Data											
3	Dimensions:	Filter	MY Fin Da	Members	Amount								
4		Filter	Version	Members	Budget								
5		Filter	Entity	Members	Sample Co								
6		Filter	Product - :	Members	Product 1								
7		Filter	Departme	Members	Online Sales								
8		Column	Month Ye:	Range	$B$12:$G$12								
9		Row	Account	Range	$A$13:$A$55								
10													
11													
12		Jan 2019	Feb 2019	Mar 2019	Apr 2019	May 2019	Jun 2019						
13	Sales Income	0							0	0	0	0	0
14	Product Licensi	0							0	0	0	0	0
15	INCOME	0							0	0	0	0	0
16	DirectCosts	0	0	0	0	0	0						

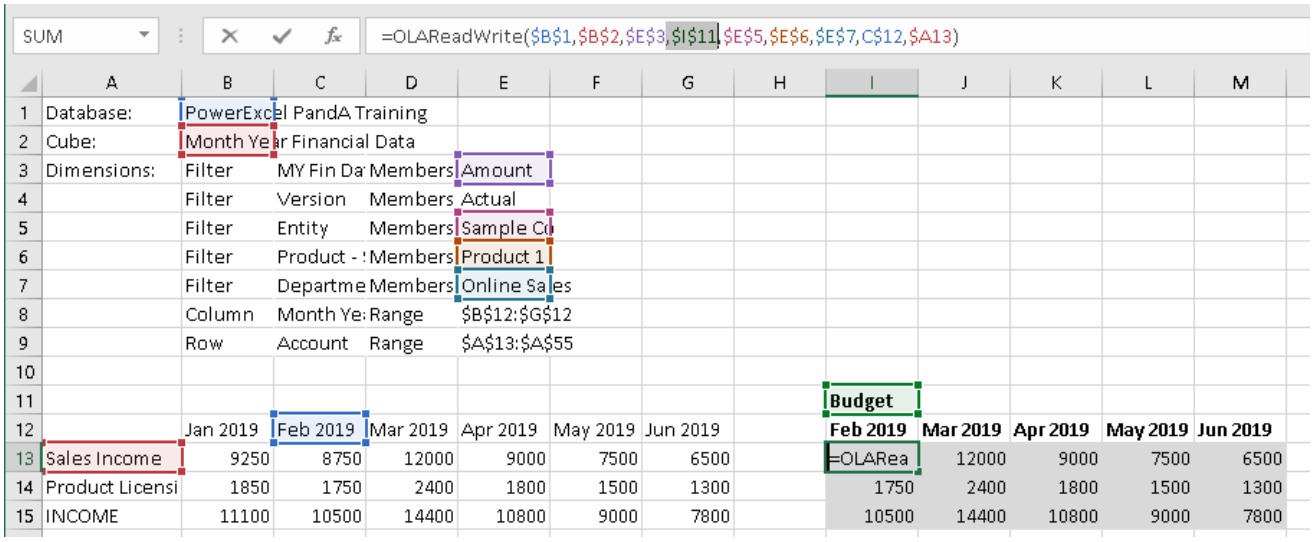
- For present purposes, revert back to the *Actual* Member (double-click on Cell E4, select Budget, etc.).
  Next, highlight Cells B13 to B15 (*Jan 2019*, *Sales Income* to *Jan 2019, INCOME*); click on the dot at the bottom right of Cell B15 and "drag" the highlight across to Column G.
- When you hit **F9 to upd**ate—you will see the that the data is the same as the date to the right, which you moved a few steps ago. Here Excel, as is its custom, ensured that what shows in Columns is the sequence (following *Jan 2020*) of *Feb 2020*, *Mar 2020*, etc., out to Column G, *Jun 2019*. (This is a fine example of Excel working the way a user expected!)



	Jan 2019	Feb 2019	Mar 2019	Apr 2019	May 2019	Jun 2019						
Sales Income	9250	8750	12000	9000	7500	6500	8750	12000	9000	7500	6500	
Product Licensi	1850	1750	2400	1800	1500	1300	1750	2400	1800	1500	1300	
INCOME	11100	10500	14400	10800	9000	7800	10500	14400	10800	9000	7800	

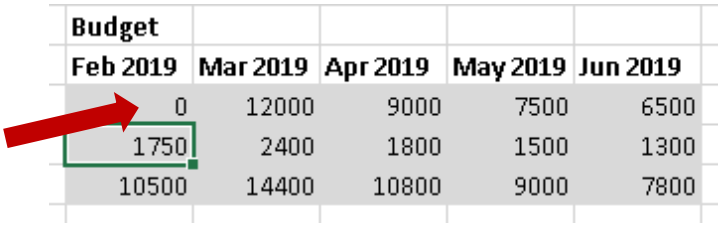
- This next step involves some housekeeping/formatting, which are always important in data presentation, and which here is necessary to arrive are our objective, which is to show *Budget* numbers to the right of the data showing *Actuals*:
  Copy the cells C12 to G12 (*Feb 2019* to Jun *2019*) and paste them into cell I12, and make those month headers **bold**.
  Then, type the word **Budget** into Cell I11, making it **bold** as well.

- Next, to change the key reference for the right “block” of cells, so that they show Budget figures:
  Click in Cell I13, then click within the function in the formula bar.
  Change the reference to *Actual* ($E$4) by highlighting it and pointing to Cell I11 (you can do this by pull the rectangle surrounding E4 down to I11).
  Before you press F9 to recalculate, your spreadsheet will look as follows:



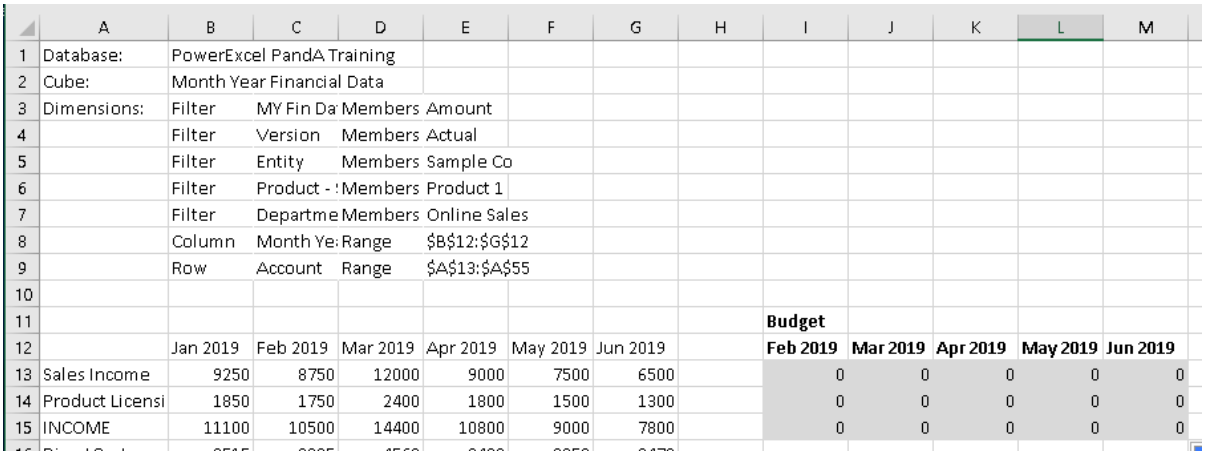
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Database:	PowerExcel	PandA Training										
2	Cube:	Month Ye	r Financial Data										
3	Dimensions:	Filter	MY Fin Da	Members	Amount								
4		Filter	Version	Members	Actual								
5		Filter	Entity	Members	Sample Co								
6		Filter	Product - !	Members	Product 1								
7		Filter	Departme	Members	Online Sales								
8		Column	Month Ye:	Range	$B$12:$G$12								
9		Row	Account	Range	$A$13:$A$55								
10													
11									Budget				
12		Jan 2019	Feb 2019	Mar 2019	Apr 2019	May 2019	Jun 2019		Feb 2019	Mar 2019	Apr 2019	May 2019	Jun 2019
13	Sales Income	9250	8750	12000	9000	7500	6500		=OLARea	12000	9000	7500	6500
14	Product Licensi	1850	1750	2400	1800	1500	1300		1750	2400	1800	1500	1300
15	INCOME	11100	10500	14400	10800	9000	7800		10500	14400	10800	9000	7800

- Hit the **Enter** key and then press **F9**—note that the value changes to zero (as per the arrow in the following image)! As we saw earlier, there is indeed a zero value for *Budget*, *Sales Income* for *Feb 2019* (for Amount, Sample Co, Product 1, Online Sales—i.e., the other referenced cells.)



Budget				
Feb 2019	Mar 2019	Apr 2019	May 2019	Jun 2019
0	12000	9000	7500	6500
1750	2400	1800	1500	1300
10500	14400	10800	9000	7800

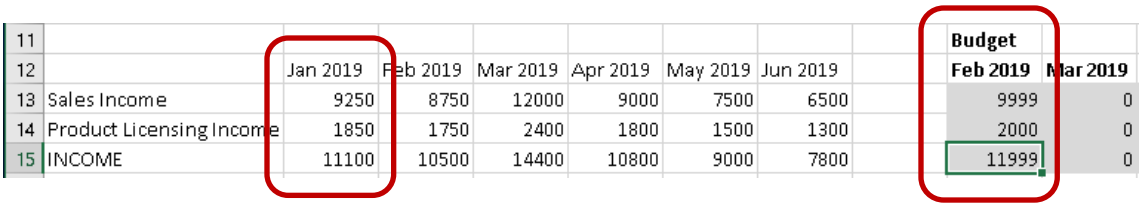
- Once again put your cursor in Cell **I13**; use dot at the bottom right and pull down to include I14 and I15; hit Enter and then F9 to update. These cells also show zero values. Lastly, highlight I13 through I15 and, as above, pull the highlighted area across, covering all of the data set through to Column M. Hit Enter and F9.
  The result set will show as in the following image.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Database:	PowerExcel PandA Training											
2	Cube:	Month Year Financial Data											
3	Dimensions:	Filter	MY Fin Da	Members	Amount								
4		Filter	Version	Members	Actual								
5		Filter	Entity	Members	Sample Co								
6		Filter	Product - !	Members	Product 1								
7		Filter	Departme	Members	Online Sales								
8		Column	Month Ye:	Range	$B$12:$G$12								
9		Row	Account	Range	$A$13:$A$55								
10													
11									Budget				
12		Jan 2019	Feb 2019	Mar 2019	Apr 2019	May 2019	Jun 2019		Feb 2019	Mar 2019	Apr 2019	May 2019	Jun 2019
13	Sales Income	9250	8750	12000	9000	7500	6500		0	0	0	0	0
14	Product Licensi	1850	1750	2400	1800	1500	1300		0	0	0	0	0
15	INCOME	11100	10500	14400	10800	9000	7800		0	0	0	0	0

As a final part of this exercise, we will do something that is covered more in depth in a later section: entering data into a PowerExcel model—here will type a couple of numbers into this spreadsheet, which will further demonstrate (and validate) the use of creating data sets in a spreadsheet by using the OLA Read Write function.

- Type example numbers—e.g., 9999 and 2000—in Cells I13 and I14 (*Sales Income* and *Product Licensing Income*, for *Feb 2019*). In this way we might be entering future budget numbers (in actuality, the entry template would be for a "plan" period, not the current months), with full, and nearby, knowledge of ongoing Actuals. (The Actual figures appear in Cells C13 and C14—circle to the left in the next image.)
  Hit F9.



11									Budget	
12		Jan 2019	Feb 2019	Mar 2019	Apr 2019	May 2019	Jun 2019		Feb 2019	Mar 2019
13	Sales Income	9250	8750	12000	9000	7500	6500		9999	0
14	Product Licensing Income	1850	1750	2400	1800	1500	1300		2000	0
15	INCOME	11100	10500	14400	10800	9000	7800		11999	0

- As shown above on the right, the two Budget numbers appear and even calculate automatically, delivering the *INCOME* number for *Feb 2019*. This calculation, defined once in the PowerExcel model—rather than in numerous cells individually, in numerous spreadsheets—demonstrates another advantage of using PowerExcel for business modeling.

13.OLATableMember

Function Description: This function covers a 'cell range' or a 'group of cells' that define the Dimension Name and the corresponding Member/s that will be displayed along the Filter area of a PowerExcel Slice. The 'cell range' covered by this function must be updated simultaneously and changes will only be committed by use of the CTRL+SHIFT+ENTER keys.

To change the Display Members of a particular Dimension, click on all the cells covered by the OLAPTableMember function, change the last parameter and enter the exact name of the new Display Member then press **Ctrl+Shift+Enter** keys to commit the changes. Notice that the update is reflected across all the cells covered by the function.

Syntax: OLATableMember (Connection, Cube, AXIS, Index, Dimension, Member)

Connection: Enter the PowerExcel connection which contains the information about the Olation server URL and the source database name.

Cube: Enter the name of the source/target Cube; or enter the cell reference that contains the name of the source or target Cube you wish to establish connection to.

AXIS: This indicates the area of the PowerExcel Slice where the data will appear (i.e., Filter, Column or Rows)

Index: 0
[NOTE: When Filter is indicated, Index will always be "0", and changing this number will not change the value result in PowerExcel.]

Dimension: Enter the Dimension name or the cell reference that contains the name of the Dimension that exists within the specified Database above.

Member: Enter the Member name or the cell reference that contains the name of the preferred display Member along the Filter area of the specified Dimension.

Remarks:

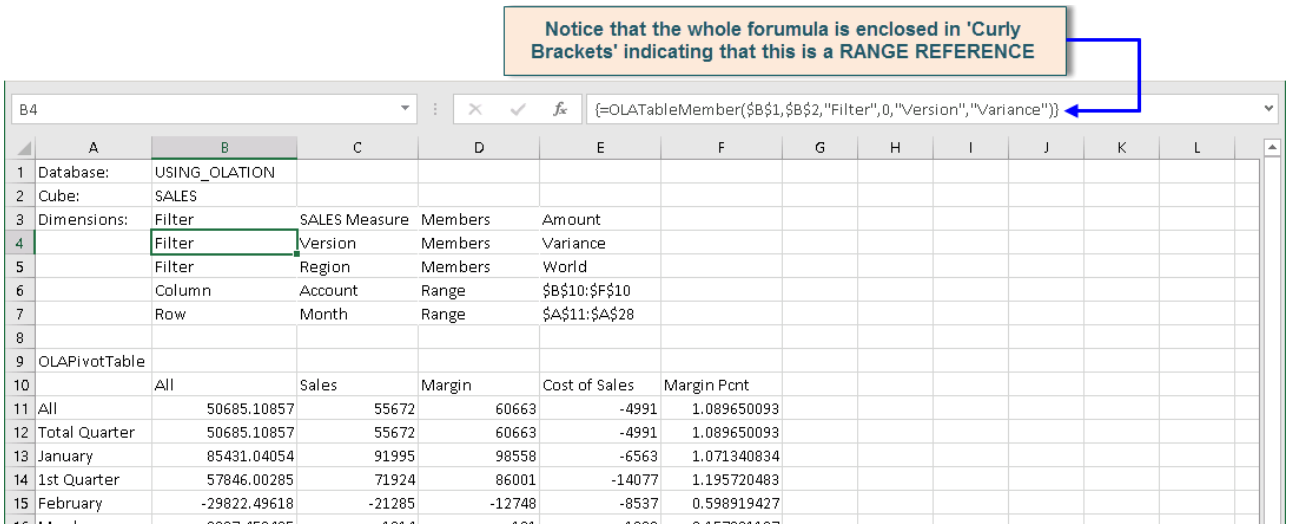
- The PowerExcel Connection must exist.
- The Olation Web Service must be running.
- The Database must be opened and running on the specified server.
- The Cube must exist within the specified database.
- The Dimension must exist within the specified database.
- The Member must exist within the specified Dimension.
- All the parameters are compulsory since the function covers a 'range of cells'
- It is imperative that 'All' cells governed by the OLATableMember function be updated in order to successfully commit the changes to the formula.

Example:

The example Slice below shows a PowerOLAP Pivot Table. The **OLATableMember** function is used to define the Members to display along the Filter area of the PowerExcel Slice. Additionally, the **OLATableMember** is a 'Range Reference', meaning it is a formula function that governs a group of cells. Hence, if you click on any cell containing the formula, you will notice that the formula is enclosed in 'Curly Brackets', an indication that this is a Range Reference: all cells covered by the same Range reference will render the same formula.

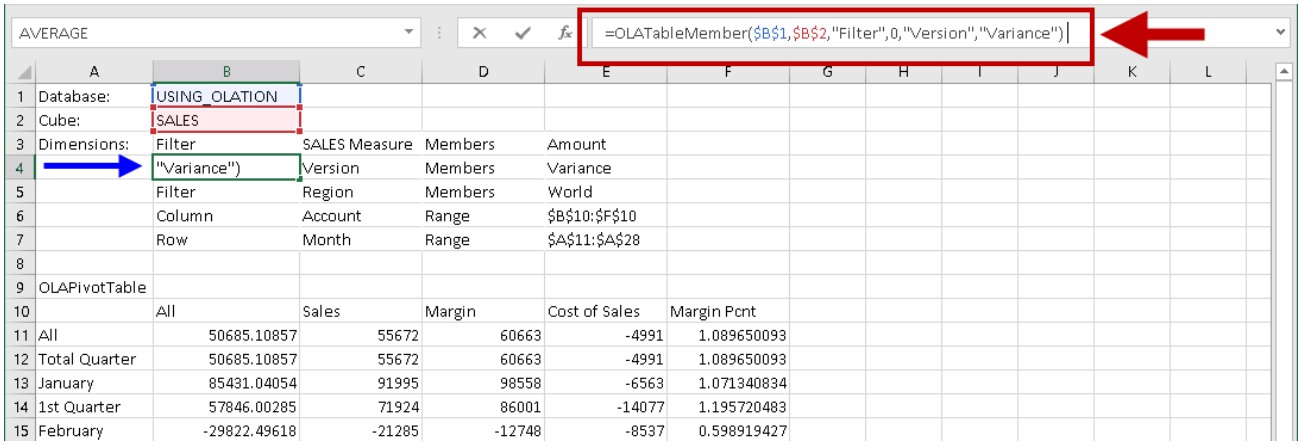
- As in the example screenshot below, the **OLATableMember** formulas can be found in the cells **B3:E3** (*SALES Measure* Dimension with the filter Member *Amount*), **B4:E4** (*Version* Dimension with the filter Member *Variance*) and **B5:E5** (*Region* Dimension with the filter Member *World*).
- When you click on the cell containing the **OLATableMember** formula (as in the example, the active cell is **B4**) notice that the **OLATableMember** formula that appears in the formula bar is enclosed in curly brackets. When you click through the cells that are part of the cell range—**C4**, **D4** and **E4**—notice that they will show the same formula as that seen in cell B4:

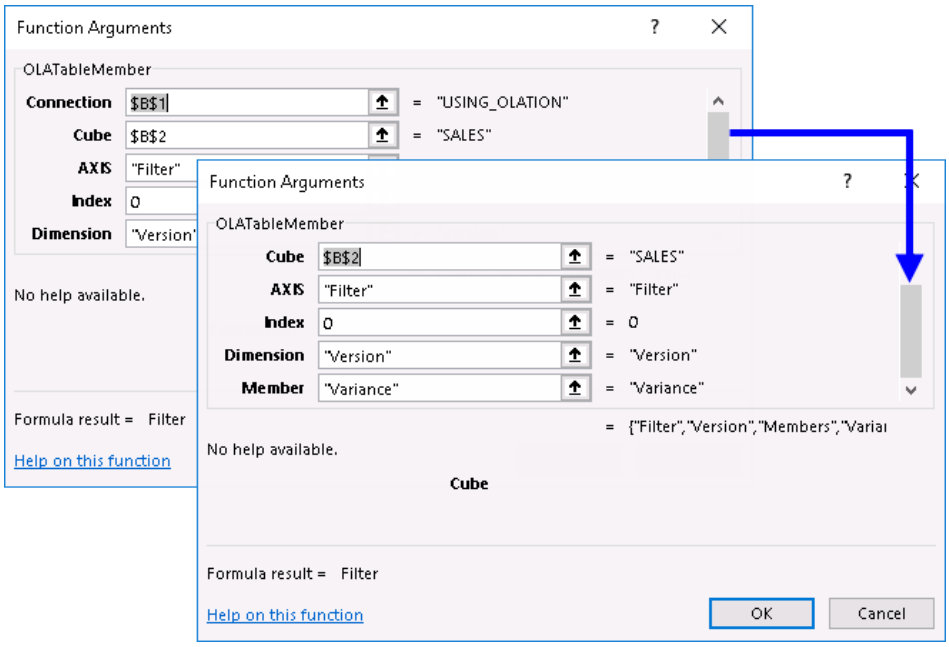
{=OLATableMember($B$1,$B$2,"Filter",0,"Version","Variance")}



	A	B	C	D	E	F
1	Database:	USING_OLATION				
2	Cube:	SALES				
3	Dimensions:	Filter	SALES Measure	Members	Amount	
4		Filter	Version	Members	Variance	
5		Filter	Region	Members	World	
6		Column	Account	Range	$B$10:$F$10	
7		Row	Month	Range	$A$11:$A$28	
8						
9	OLAPivotTable					
10		All	Sales	Margin	Cost of Sales	Margin Pcnt
11	All	50685.10857	55672	60663	-4991	1.089650093
12	Total Quarter	50685.10857	55672	60663	-4991	1.089650093
13	January	85431.04054	91995	98558	-6563	1.071340834
14	1st Quarter	57846.00285	71924	86001	-14077	1.195720483
15	February	-29822.49618	-21285	-12748	-8537	0.598919427

- By clicking in the formula bar area (in the next image, the mouse cursor is placed at the end of the formula), the cell references corresponding to the **OLATableMember** function will be outlined on the spreadsheet. The OLATableMember function returns **Filter**(B4), **Version**(C4), **Members**(D4) and **Variance**(E4)
- It is also referencing the Cube called "*SALES*"(**$B$2**) and is using the PowerExcel connection/OLADatabase connection called "*USING_OLATION*"(**$B$1**).
  Note: When you are on edit mode of a Range Reference type of formula, notice that the 'Curly Brackets' disappear. Click on Esc (Escape) to come out of the formula bar.





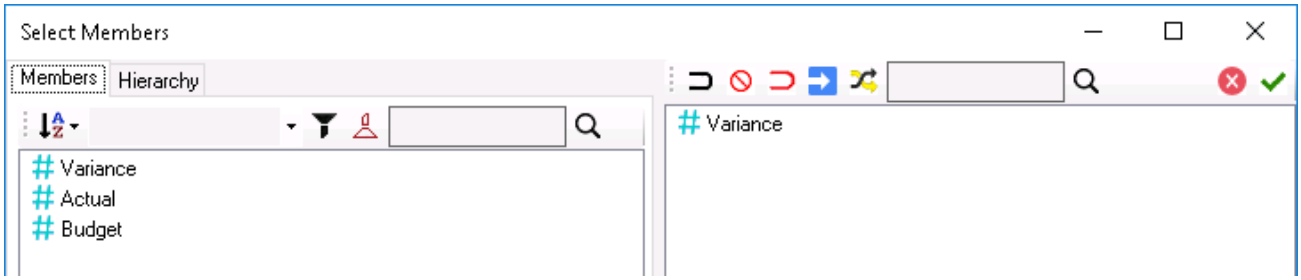
Cell References:

```
=OLATableMember($B$1,$B$2,"Filter",0,"Version","Variance")
```

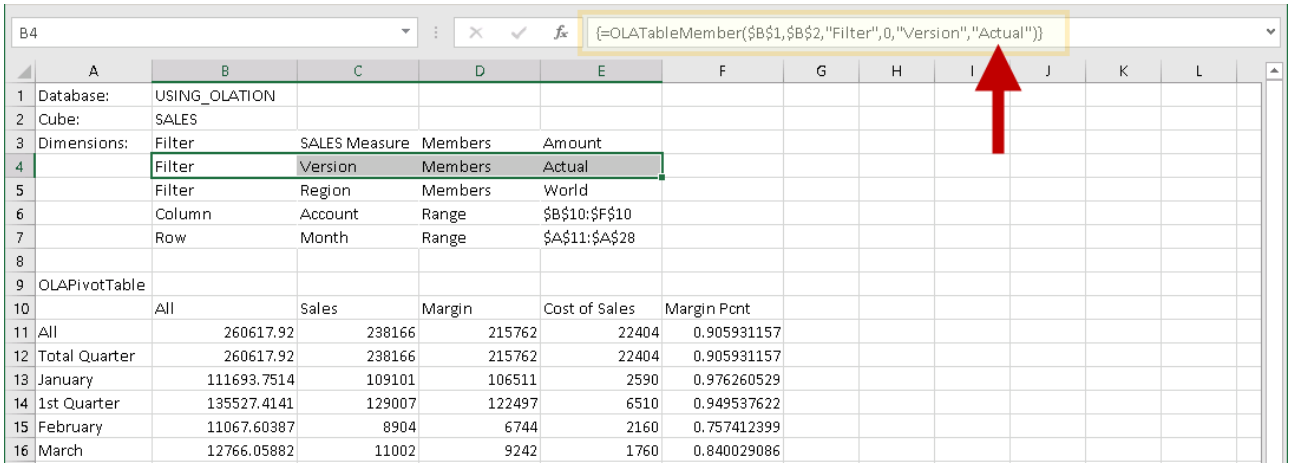
- $B$1– the Database name or the PowerExcel connection name/cell reference that contains the name of the Database or the PowerExcel connection, i.e., **USING_OLATION**
- $B$2– the Cube name or the cell reference that contains the name of the Cube, i.e., **SALES**
- "Filter"– This indicate that the data will appear along the Filter area of the Slice
- "0" – [Note: this is a constant when Filter is indicated, and there will be no change if another number is used]
- "Version"– the Dimension name that exists within the database placed along the Filter area of the Slice

- ”Variance”– the Member name that exists within the specified Dimension that will serve as the Filter Member

As an example: assume that you know the Members that exist for the *Version* Dimension: *Variance*, *Actual* and *Budget*, illustrated in the image below.



- Next, you want to change the current display or Filter Member (*Variance*) to *Actual*. Select all the cells governed by the **OLATableMember** formula for Version Dimension (**B4:E4**); then click on the formula bar and double-click on the **Member parameter** (the last parameter; in this example it is *Variance*). Next, type in the new filter Member, which is **Actual** (make sure that this is enclosed in double quotes). Press **Ctrl+Shift+Enter** keys to commit the formula change. Click the **Refresh** button along the PowerExcel Tab of the Excel ribbon, or press **F9**. The **OLATableMember** is now updated and the table shows the new fact data for *Actual*.



	A	B	C	D	E	F	G	H	I	J	K	L
1	Database:	USING_OLATION										
2	Cube:	SALES										
3	Dimensions:	Filter	SALES Measure	Members	Amount							
4		Filter	Version	Members	Actual							
5		Filter	Region	Members	World							
6		Column	Account	Range	$B$10:$F$10							
7		Row	Month	Range	$A$11:$A$28							
8												
9	OLAPivotTable											
10		All	Sales	Margin	Cost of Sales	Margin Pcnt						
11	All	260617.92	238166	215762	22404	0.905931157						
12	Total Quarter	260617.92	238166	215762	22404	0.905931157						
13	January	111693.7514	109101	106511	2590	0.976260529						
14	1st Quarter	135527.4141	129007	122497	6510	0.949537622						
15	February	11067.60387	8904	6744	2160	0.757412399						
16	March	12766.05882	11002	9242	1760	0.840029086						

- If you click across cells B4 to E4 you will see the new formula: **{=OLATableMember($B$1,$B$2,"Filter",0,"Version","Actual")}**

14.OLATableMembers

The previous function described is OLATableMember – note, first, that this function, **OLATableMembers**, differs only by an "s" at the end.

Function Description: This function covers a 'cell range' or a 'group of cells' that define the Dimension Name and the corresponding Member/s that will be displayed along either the Filter area or the Row area of a PowerExcel Slice. [NOTE: when used with the Row area, this function is used when either Dynamic Rows Labels or Constrain Empty Rows is enabled—both by means of a checkbox in the PowerExcel pane.] Individual Dimension members are indicated, rather than a 'cell range', and to change the Member/s displayed requires only that any Members be added or deleted in the cell referenced by the last argument.

Syntax: OLATableMembers (Connection, Cube, AXIS, Index, Dimension, Members)

Connection: Enter the PowerExcel connection which contains the information about the Olation server URL and the source database name.

Cube: Enter the name of the source/target Cube; or enter the cell reference that contains the name of the source or target Cube you wish to establish connection to.

AXIS: This indicates the area of the PowerExcel Slice where the data will appear (i.e., Filter, Column or Rows)

Index: 0
[NOTE: When Filter is indicated, Index will always be "0", and changing this number will not change the value result in PowerExcel.]

Dimension: Enter the Dimension name or the cell reference that contains the name of the Dimension that exists within the specified Database above.

Member: Enter the Member names in the cell reference that contains the name of the preferred display Members along the Filter or the Rows area of the specified Dimension.

Remarks:

- The PowerExcel Connection must exist.
- The Olation Web Service must be running.
- The Database must be opened and running on the specified server.
- The Cube must exist within the specified database.
- The Dimension must exist within the specified database.
- The Member must exist within the specified Dimension.
- All the parameters are compulsory since the function covers a 'range of cells'

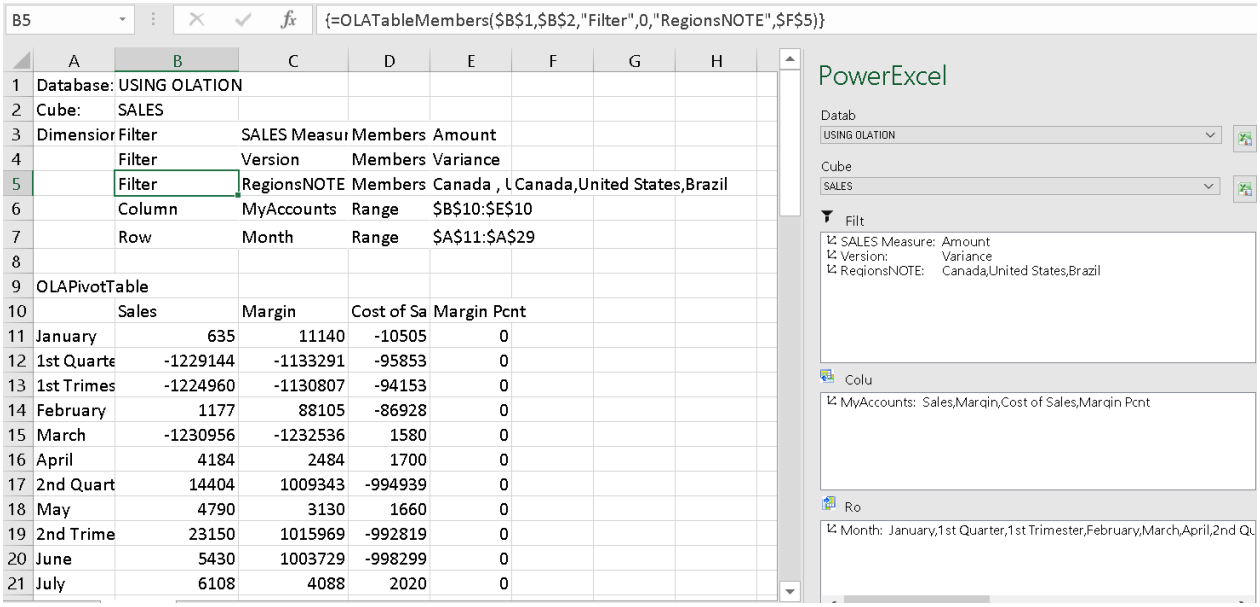
Example (for Filter area):

The example Slice below shows a PowerOLAP Pivot Table. The **OLATableMembers** function is used to define the Members to display along the Filter area of the PowerExcel Slice.

Additionally, the **OLATableMembers** is a 'Range Reference', meaning it is a formula function that governs a group of cells. Hence, if you click on any cell containing the formula, you will notice that the formula is enclosed in 'Curly Brackets', an indication that this is a Range Reference: all cells covered by the same Range reference will render the same formula.

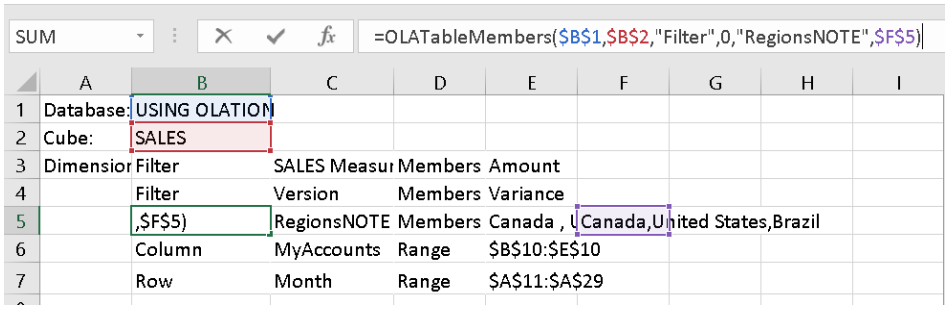
- As in the example screenshot below, the **OLATableMembers** formulas can be found in the cells **B5:E5** (*RegionsNOTE* Dimension with Members referenced in Cell $F$5).
- When you click on the cell containing the **OLATableMembers** formula (as in the example, the active cell is **B3**) notice that the **OLATableMembers** formula that appears in the formula bar is enclosed in curly brackets. When you click through the cells that are part of the cell range—**C5**, **D5** and **E5**—notice that they will show the same formula as that seen in cell B3:

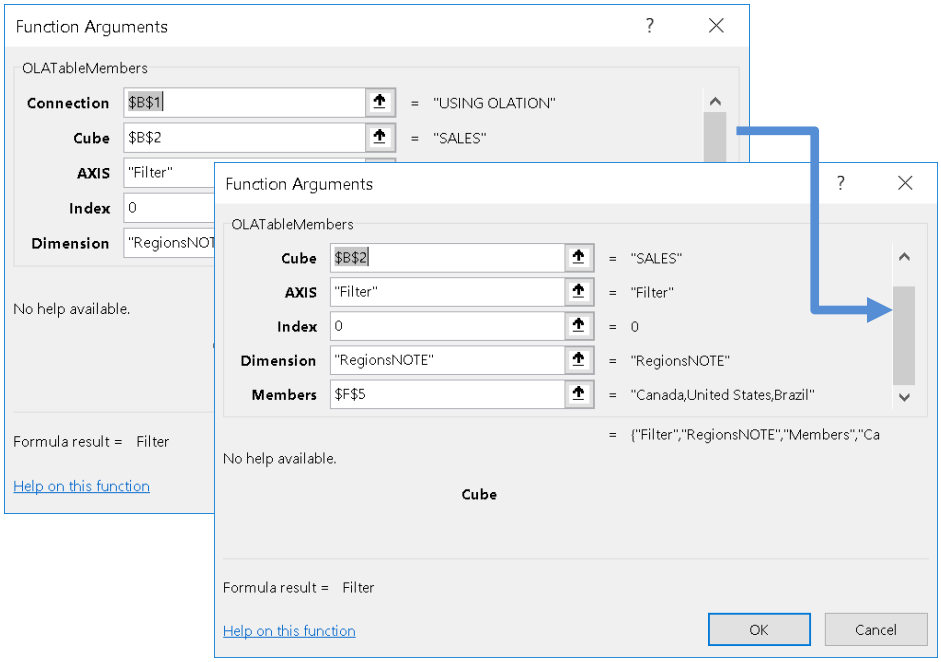
{=OLATableMembers($B$1,$B$2,"Filter",0,"RegionsNOTE",$F$5)}



	A	B	C	D	E	F	G	H
1	Database:	USING OLATION						
2	Cube:	SALES						
3	Dimensior	Filter	SALES Measu	Members	Amount			
4		Filter	Version	Members	Variance			
5		Filter	RegionsNOTE	Members	Canada , L	Canada,United States,Brazil		
6		Column	MyAccounts	Range	$B$10:$E$10			
7		Row	Month	Range	$A$11:$A$29			
8								
9	OLAPivotTable							
10		Sales	Margin	Cost of Sa	Margin Pcnt			
11	January	635	11140	-10505	0			
12	1st Quarte	-1229144	-1133291	-95853	0			
13	1st Trimes	-1224960	-1130807	-94153	0			
14	February	1177	88105	-86928	0			
15	March	-1230956	-1232536	1580	0			
16	April	4184	2484	1700	0			
17	2nd Quart	14404	1009343	-994939	0			
18	May	4790	3130	1660	0			
19	2nd Trime	23150	1015969	-992819	0			
20	June	5430	1003729	-998299	0			
21	July	6108	4088	2020	0			

- By clicking in the formula bar area (in the next image, the mouse cursor is placed at the end of the formula), the cell references corresponding to the **OLATableMembers** function will be outlined on the spreadsheet. The OLATableMembers function returns **Filter**(B5), **RegionsNOTE**(C4), **Members**(D4) and **$F$5**—this latter cell itself contains the Member/s refenced by the function (in this example: *Canada,United States,Brazil*).
- It is also referencing the Cube called "SALES"(**$B$2**) and is using the PowerExcel connection/OLADatabase connection called "USING_OLATION"(**$B$1**).
  Note: When you are on edit mode of a Range Reference type of formula, notice that the 'Curly Brackets' disappear. Click on Esc (Escape) to come out of the formula bar.





Cell References:

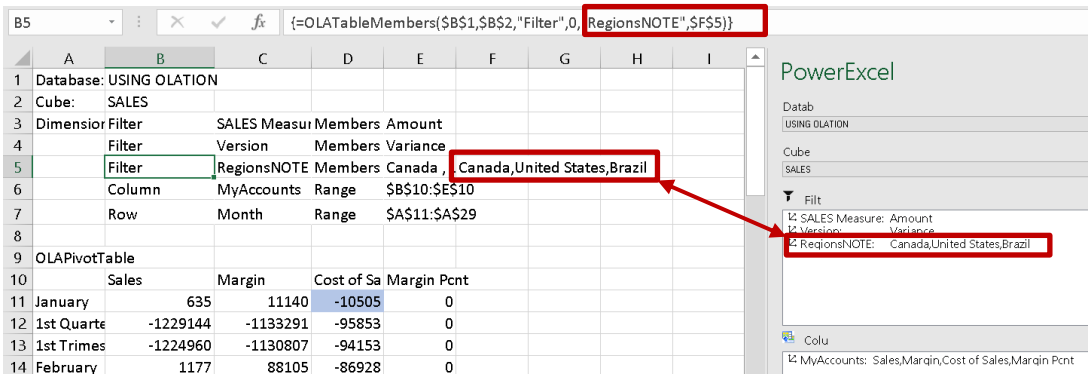
```
=OLATableMembers($B$1,$B$2,"Filter",0,"Version","
Variance")
```

- $B$1– the Database name or the PowerExcel connection name/cell reference that contains the name of the Database or the PowerExcel connection, i.e., **USING_OLATION**
- $B$2– the Cube name or the cell reference that contains the name of the Cube, i.e., **SALES**
- ”Filter”– This indicate that the data will appear along the Filter area of the Slice
- “0”- [Note: this is a constant when Filter is indicated, and there will be no change if another number is used.]
- ”RegionsNOTE”– the Dimension name that exists within the database placed along the Filter area of the Slice

- $F$5– the Member names that exists within the specified Dimension that are referenced by that Cell: those Members values will be added for the values returned to the spreadsheet.

Example 1: Use of **OLATableMembers function with Filter**

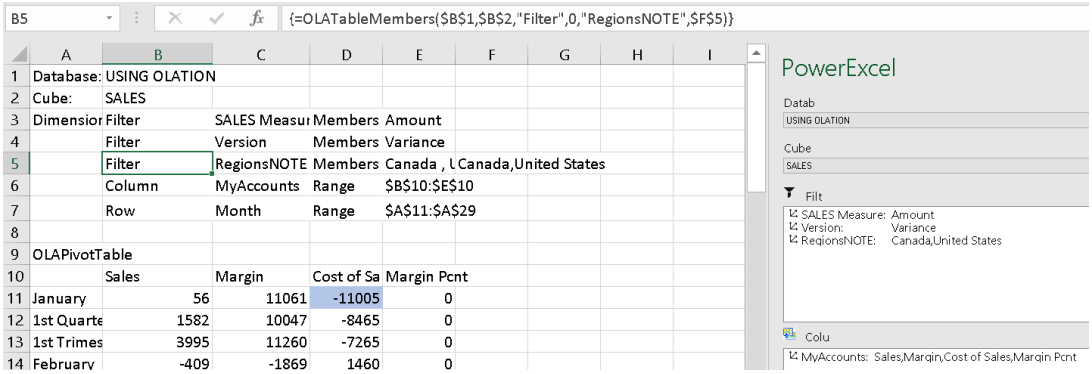
This last bullet point is key: when the OLATableMembers function is used with the Filters area of the speadhseet, values returned will be added for the Members specified. In this spreadsheet example, the values for *Cost of Sales*, *Variance*, in *January* is **-10505** (Cell D11, in**)**—that is, for the RegionsNOTE (i.e., the regions) members of *Canada*, *United States* and *Brazil* added together. These Members exist in Cell **$F$5** and are refenced in the last argument of the OLATableMembers function—all as shown in the following image. (Not coincidentally, those are the Members that appear in the PowerExcel pane on the right, as the Filter selections for RegionsNote.)



- Next, assume that you wish to change the spreadsheet to show values returned for only *United States* and *Canada*: you can of course use the PowerExcel pane to the right—you would double-click on the dimension, *RegionNote*, and then select those two countries, so that their values would be added.

  OR

- You can now use the OLATableMembers function: simply click in the Cell referenced by the parameter at the end of the function, in this case Cell **$F$5**. Delete Brazil (and the comma that precedes it).
- Hit **Enter**, then **F9**.
  The value for *United States* and *Canada*, added together, appears in cell **D11** (that is, -**11005**).

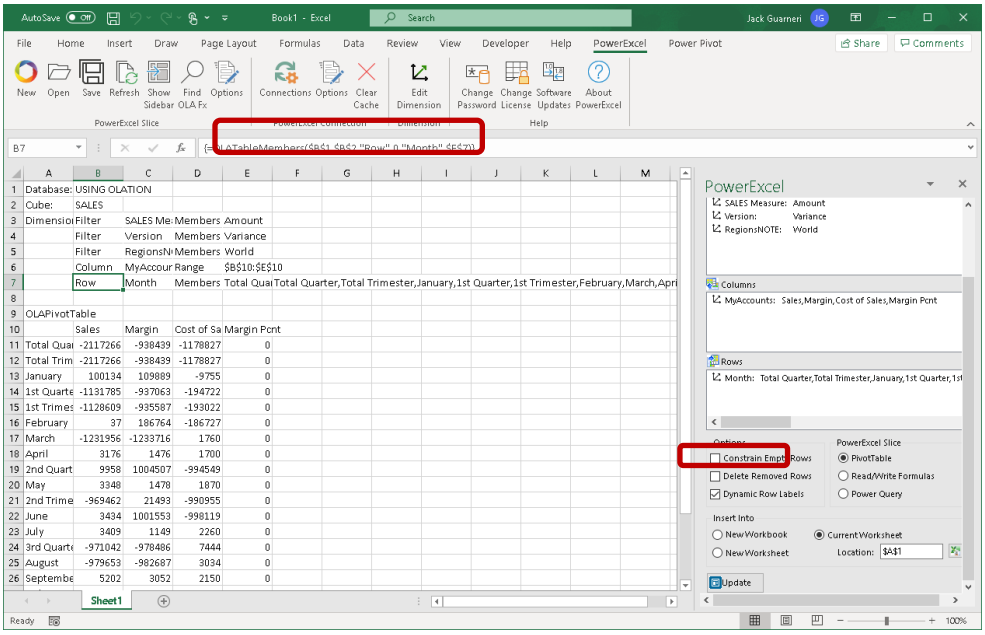


Example 2: Use of OLATableMembers function with **Dynamic Row Labels enabled**

Until now, we have shown the OLATableMembers function used for the Filters area. Another use of the function occurs when either of the following is enabled by its corresponding checkbox in the PowerExcel pane: (a) Dynamic Row Labels, or (b) Constrain Empty Rows.

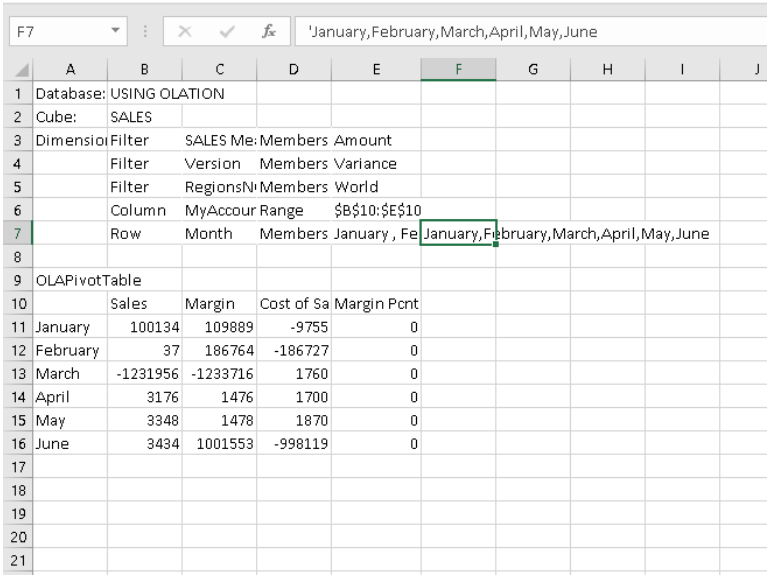
For present purposes we will show what happens when Dynamic Row Labels is enabled:

- Create a Slice using the PivotTable function, like the one in the following image. When you initially click in the cell that shows Row (**B7** in the example), what appears in the formula bar is an OLATableRange function—this function will be explained at a later point in this document. (As an aside: this function also covers a 'cell range' or a 'group of cells': you will see the same formula in Cells C7, D7 and E7).
- Locate/enable the **Dynamic Row Labels checkbox** (boxed in the image) in the PowerExcel pane to the right. Click **Update**. The Slice will look as follows:



- Note that now with your curser where the word Row appears, an **OLATableMembers** function appears in the formula bar (also shown in the preceding image).

- The same function will appear as you move your cursor right (to Cells C7, D7, E7).
- As in Example 1, the referenced Members (here for the *Months* dimension) are indicated in the last parameter of the function: **Cell $F$7**. All the Months indicated there—as a consequence of **Dynamic Row Labels** being enabled—are shown in Rows in the Slice. As a consequence, one can remove or add Members by typing them in, or by deleting them (just as in Example 1), and the Rows will reflect the new list of Months. In the following image, the first six months of the year have been left (or typed in). After hitting **Enter** and **F9**, the Slice will now show only those 6 months in Rows.



	A	B	C	D	E	F	G	H	I	J
1	Database:	USING OLATION								
2	Cube:	SALES								
3	Dimensio	Filter	SALES Me	Members	Amount					
4		Filter	Version	Members	Variance					
5		Filter	RegionsN	Members	World					
6		Column	MyAccour	Range	$B$10:$E$10					
7		Row	Month	Members	January , Fe	January,F	ebruary,March,April,May,June			
8										
9	OLAPivotTable									
10		Sales	Margin	Cost of Sa	Margin Pcnt					
11	January	100134	109889	-9755	0					
12	February	37	186764	-186727	0					
13	March	-1231956	-1233716	1760	0					
14	April	3176	1476	1700	0					
15	May	3348	1478	1870	0					
16	June	3434	1001553	-998119	0					
17										
18										
19										
20										
21										

15.OLATableRange

Function Description: This function covers a 'cell range' or a 'group of cells' that define the Dimension and corresponding Members that will be displayed along the Rows or Columns of a PowerExcel Slice. The 'cell range' covered by this function must be updated simultaneously and changes will only be committed by use of the **CTRL+SHIFT+ENTER** keys. This function applies to PowerExcel PivotTable or PowerExcel ReadWrite Slice outputs when they are initially created.

To change the Display Members along the Row or Column of a PowerExcel Slice, click on all the cells covered by the OLAPTableRange function, change the 'range function argument' or the last parameter and enter the 'new target range'. Press the **Ctrl+Shift+Enter** keys to commit the changes. Notice that the update is reflected across all the cells covered by the function.

Syntax: OLATableRange (Connection, Cube, AXIS, Index, Dimension, Range)

Connection: Enter the PowerExcel connection that contains the information about the Olation server URL and the source database name.

Cube: Enter the name of the source/target Cube; or enter the cell reference that contains the name of the source or target Cube.

AXIS: This indicates the area of the PowerExcel Slice where the data will appear (i.e., Filter, Column or Rows)

Index: 0 [NOTE: When there are no "stacked Dimensions" in Row or Column, "0" will always show; If there are stacked Dimensions, the number will reflect the order of the Rows or Columns, starting with "1" as the "topmost" in the stack, and continuing.]

Dimension: Enter the Dimension name or the cell reference that contains the name of the Dimension that exists within the specified Database above.

Range: Enter the range corresponding to the target Display Members along the row or column

Remarks:

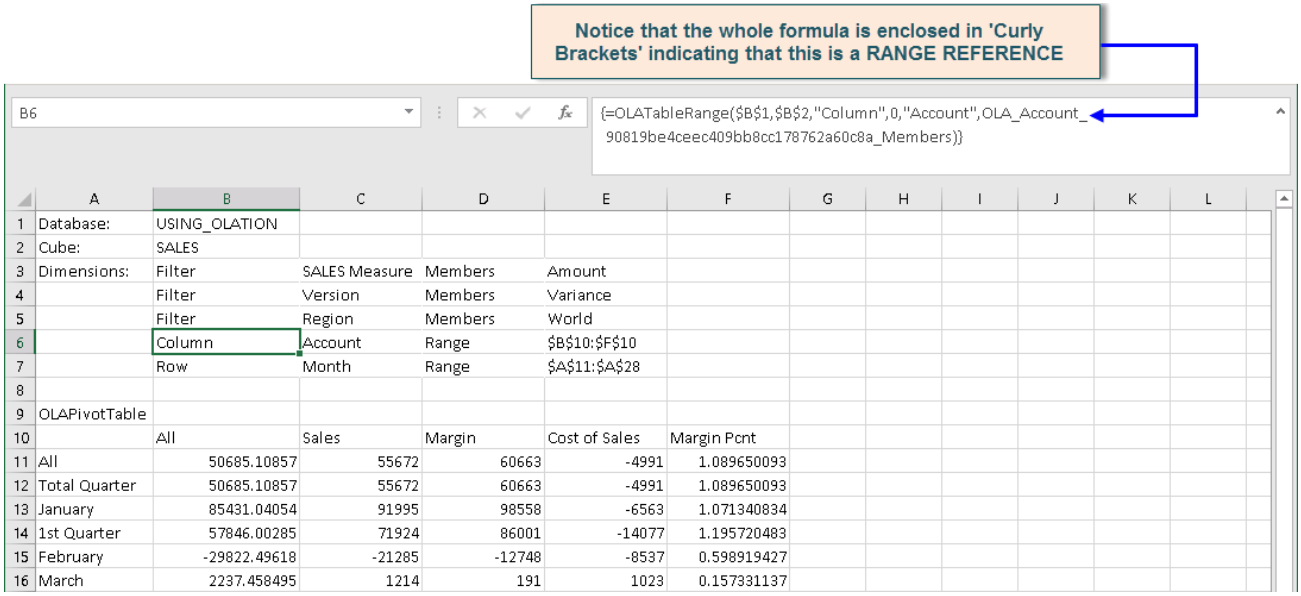
- The PowerExcel Connection must exist.
- The Olation Web Service must be running.
- The Database must be opened and running in the specified server.
- The Cube must exist within the specified database.
- The Dimension must exist within the specified Database.
- All the parameters are compulsory since the function covers a 'range of cells'.
- It is imperative that 'All' cells governed by the OLATableRange function be updated in order to successfully commit the changes to the formula.

Example:

The example Slice below shows a PowerOLAP Pivot Table. The **OLATableRange** function is used to define what Members to display along the Row or Column area of a PowerExcel Slice. Additionally, the **OLATableRange** is a 'Range Reference', meaning it is a formula function that governs a group of cells. Hence, if you click on any cell containing the formula, you will notice that the formula is enclosed in 'Curly Brackets', an indication that this is a Range Reference; all cells covered by the same Range reference will return the exact same formula.

- As in the example screenshot below, the **OLATableRange** formula for Columns can be found in the cells **B6:E6** (*Account* Dimension on Columns). Clicking through all the cells governed by this specific Range Reference formula will return the exact same formula.
- When you click on the cell containing the **OLATableRange** formula (the example, the active cell is **B6)**, notice that the **OLATableRange** formula that appears in the formula bar is enclosed in curly brackets. When you click through the cells that are part of the cell range: **C6**, **D6** and **E6**, notice that it will show the same formula as that seen in cell B6:

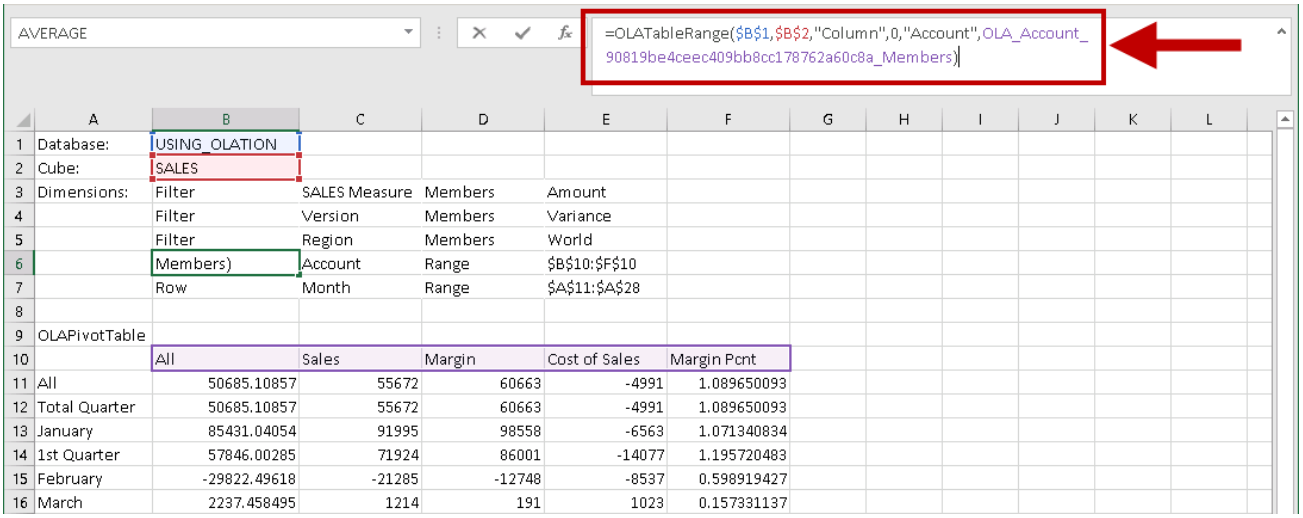
{=OLATableRange($B$1,$B$2,"Column",0,"Account",OLA_Account_90819be4ceec409bb8cc178762a60c8a_Members)}



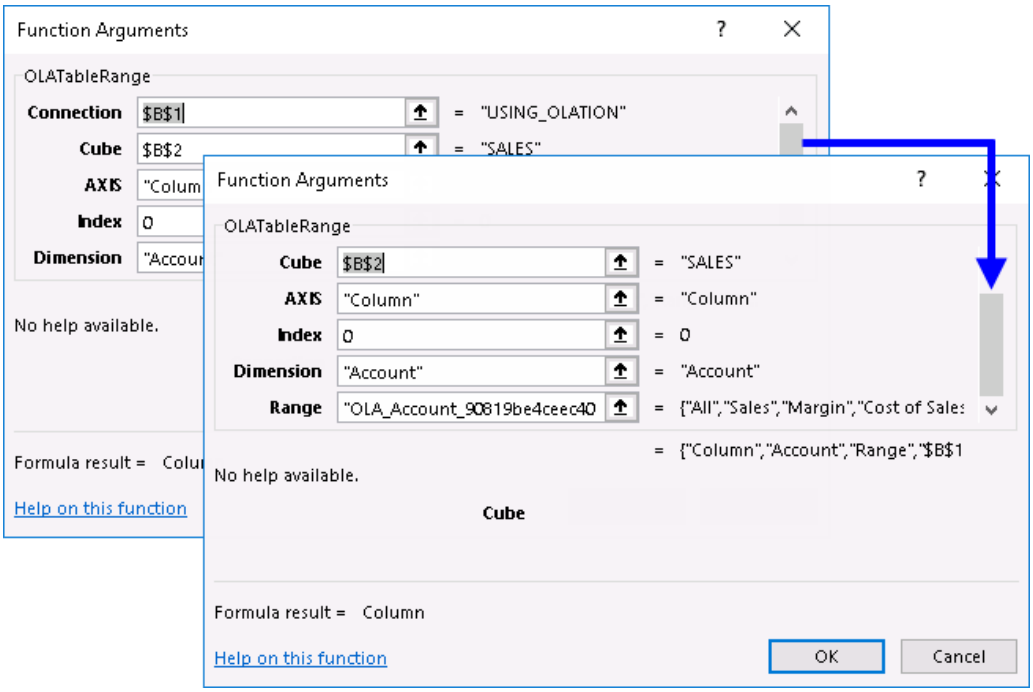
	A	B	C	D	E	F
1	Database:	USING_OLATION				
2	Cube:	SALES				
3	Dimensions:	Filter	SALES Measure	Members	Amount	
4		Filter	Version	Members	Variance	
5		Filter	Region	Members	World	
6		Column	Account	Range	$B$10:$F$10	
7		Row	Month	Range	$A$11:$A$28	
8						
9	OLAPivotTable					
10		All	Sales	Margin	Cost of Sales	Margin Pcnt
11	All	50685.10857	55672	60663	-4991	1.089650093
12	Total Quarter	50685.10857	55672	60663	-4991	1.089650093
13	January	85431.04054	91995	98558	-6563	1.071340834
14	1st Quarter	57846.00285	71924	86001	-14077	1.195720483
15	February	-29822.49618	-21285	-12748	-8537	0.598919427
16	March	2237.458495	1214	191	1023	0.157331137

- By clicking in the formula bar area (in the next image, the mouse cursor is placed at the end of the formula), the cell references corresponding to the **OLATableRange** function will be outlined on the spreadsheet. **Column**(B6), **Account**(C6), **Range**(D6) and **$B$10:$F$10**(E6).
- It is also referencing the Cube *"SALES"*(**$B$2**) NS the PowerExcel connection/Database connection called *"USING_OLATION"*(**$B$1**) and references to the "cell range" along the column which corresponds to display Members for the Account Dimension (**OLA_Account_90819be4ceec409bb8cc178762a60c8a_Members**).

  Note: When you are on edit mode of a Range Reference type of formula, notice that the 'Curly Brackets' disappear. Click on Esc (Escape) to come out of the formula bar.



	A	B	C	D	E	F
1	Database:	USING_OLATION				
2	Cube:	SALES				
3	Dimensions:	Filter	SALES Measure	Members	Amount	
4		Filter	Version	Members	Variance	
5		Filter	Region	Members	World	
6		Members)	Account	Range	$B$10:$F$10	
7		Row	Month	Range	$A$11:$A$28	
8						
9	OLAPivotTable					
10		All	Sales	Margin	Cost of Sales	Margin Pcnt
11	All	50685.10857	55672	60663	-4991	1.089650093
12	Total Quarter	50685.10857	55672	60663	-4991	1.089650093
13	January	85431.04054	91995	98558	-6563	1.071340834
14	1st Quarter	57846.00285	71924	86001	-14077	1.195720483
15	February	-29822.49618	-21285	-12748	-8537	0.598919427
16	March	2237.458495	1214	191	1023	0.157331137



Cell References:

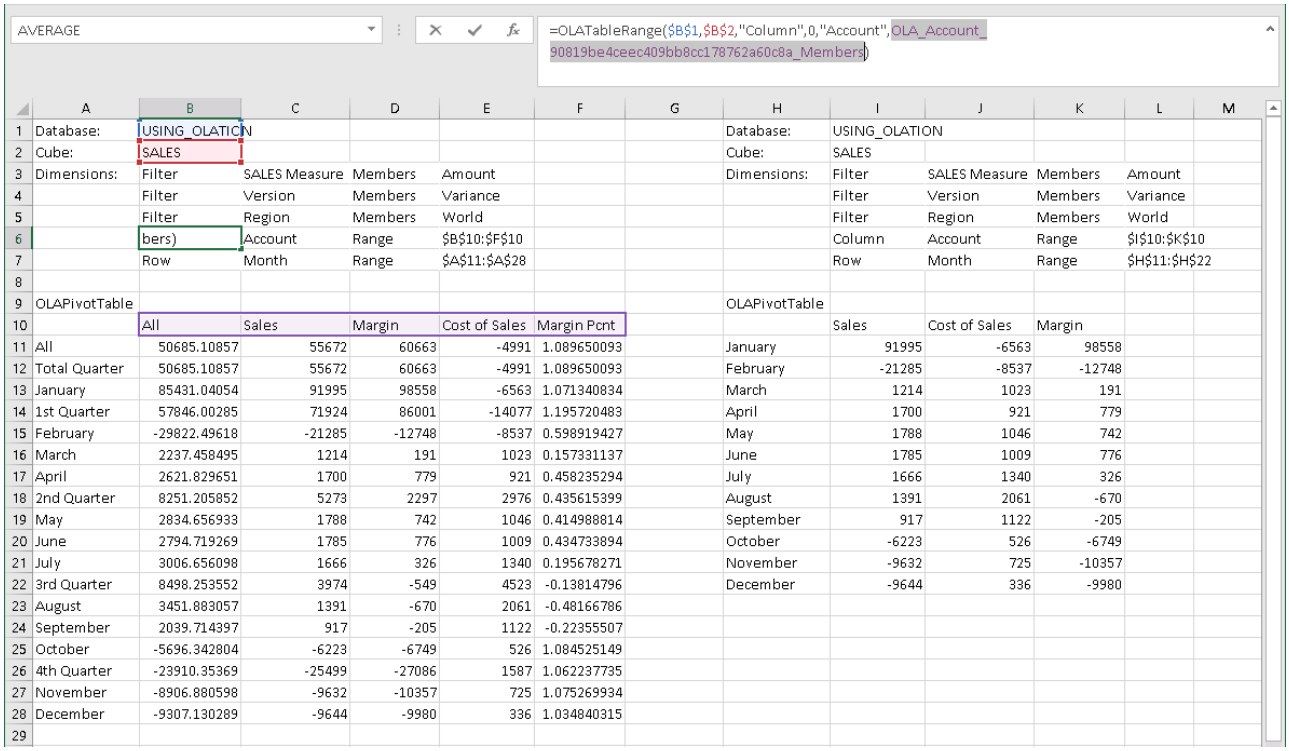
={OLATableRange($B$1,$B$2,"Column",0,"Account",OLA_Account_90819be4ceec409bb8cc178762a60c8a_Members)}

- $B$1– the Database name or the PowerExcel connection name/cell reference that contains the name of the Database or the PowerExcel connection, i.e., **USING_OLATION**
- $B$2– the Cube name or the cell reference that contains the name of the Cube, i.e., **SALES**
- "Column"– this indicates the data will appear along the Column area of the Slice

- 0– this indicates the index number of the order of Dimensions, thus “0” when there are no stacked Dimensions in Rows or Columns to order [otherwise, “1”, “2”, etc.
- ”Account”– the Dimension name that exists within the database
- OLA_Account_ae313fdff64047288a76921b05908dac_Members– this range corresponds to the target display Members that will dictate what Account members will be displayed along the column area of the Slice output

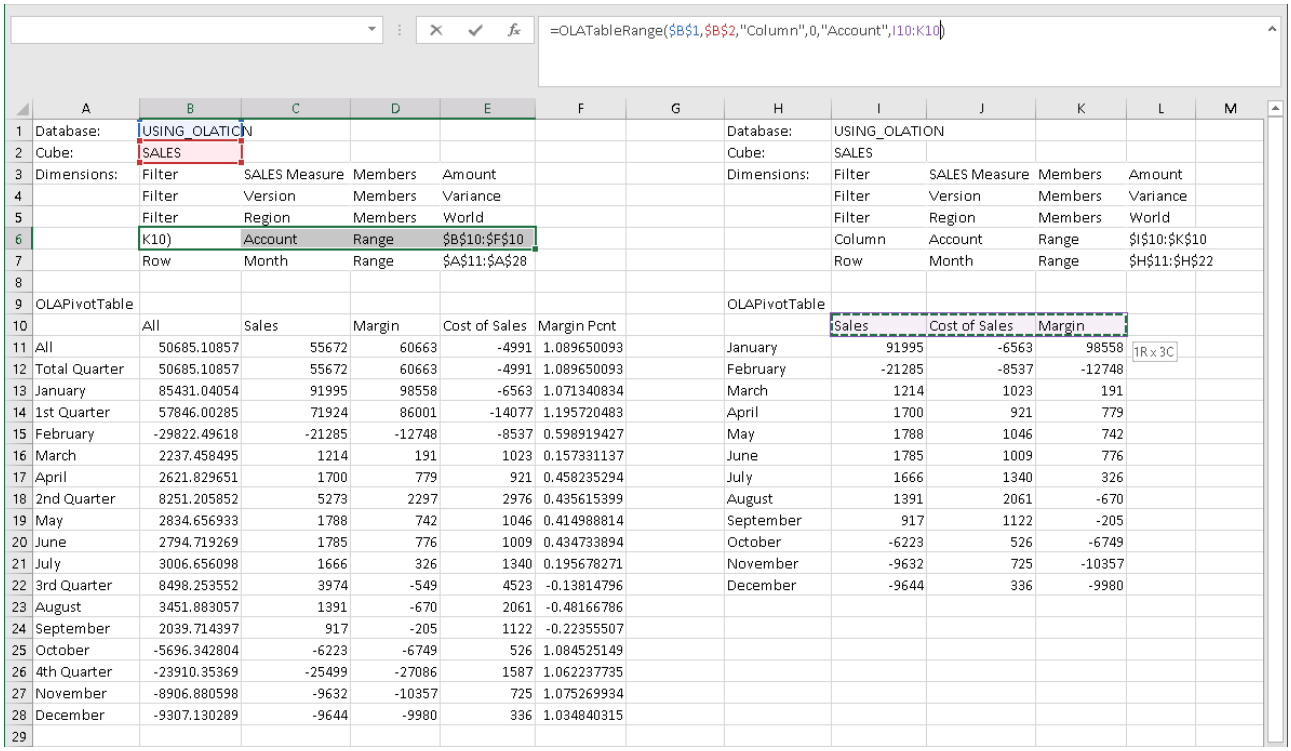
As an example: assume we have created two PivotTables within a worksheet; we want to have the same set of Account Members displayed in our first PivotTable based on those appearing in the second PivotTable.

- While on the first PivotTable, select all cells governed by the **OLATableRange** formula for Account Dimension (**B6:E6**); then click on the formula bar and double-click on the Range parameter (the last parameter). You will notice that the last parameter/Range parameter is highlighted at this point.



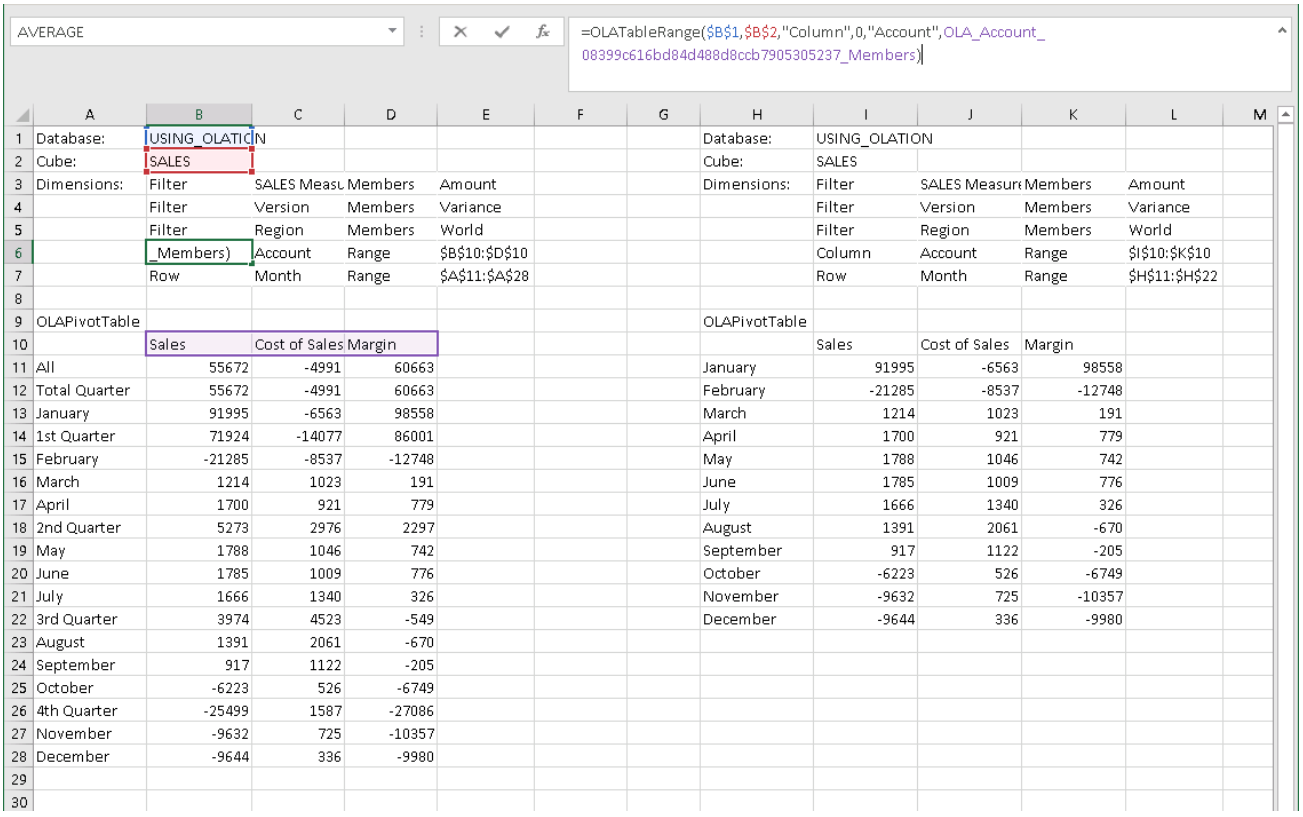
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Database:	USING_OLATION						Database:	USING_OLATION				
2	Cube:	SALES						Cube:	SALES				
3	Dimensions:	Filter	SALES Measure	Members	Amount			Dimensions:	Filter	SALES Measure	Members	Amount	
4		Filter	Version	Members	Variance				Filter	Version	Members	Variance	
5		Filter	Region	Members	World				Filter	Region	Members	World	
6		bers)	Account	Range	$B$10:$F$10				Column	Account	Range	$I$10:$K$10	
7		Row	Month	Range	$A$11:$A$28				Row	Month	Range	$H$11:$H$22	
8													
9	OLAPivotTable							OLAPivotTable					
10		All	Sales	Margin	Cost of Sales	Margin Pcnt			Sales	Cost of Sales	Margin		
11	All	50685.10857	55672	60663	-4991	1.089650093		January	91995	-6563	98558		
12	Total Quarter	50685.10857	55672	60663	-4991	1.089650093		February	-21285	-8537	-12748		
13	January	85431.04054	91995	98558	-6563	1.071340834		March	1214	1023	191		
14	1st Quarter	57846.00285	71924	86001	-14077	1.195720483		April	1700	921	779		
15	February	-29822.49618	-21285	-12748	-8537	0.598919427		May	1788	1046	742		
16	March	2237.458495	1214	191	1023	0.157331137		June	1785	1009	776		
17	April	2621.829651	1700	779	921	0.458235294		July	1666	1340	326		
18	2nd Quarter	8251.205852	5273	2297	2976	0.435615399		August	1391	2061	-670		
19	May	2834.656933	1788	742	1046	0.414988814		September	917	1122	-205		
20	June	2794.719269	1785	776	1009	0.434733894		October	-6223	526	-6749		
21	July	3006.656098	1666	326	1340	0.195678271		November	-9632	725	-10357		
22	3rd Quarter	8498.253552	3974	-549	4523	-0.13814796		December	-9644	336	-9980		
23	August	3451.883057	1391	-670	2061	-0.48166786							
24	September	2039.714397	917	-205	1122	-0.22355507							
25	October	-5696.342804	-6223	-6749	526	1.084525149							
26	4th Quarter	-23910.35369	-25499	-27086	1587	1.062237735							
27	November	-8906.880598	-9632	-10357	725	1.075269934							
28	December	-9307.130289	-9644	-9980	336	1.034840315							
29													

- Next, go to the second PivotTable and select the preferred new range: in the second PivotTable this would be **I10:K10**. Press **Ctrl+Shift+Enter** keys. The change will be saved across the cell range **B6:E6** in the first PivotTable.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Database:	USING_OLATION						Database:	USING_OLATION				
2	Cube:	SALES						Cube:	SALES				
3	Dimensions:	Filter	SALES Measure	Members	Amount			Dimensions:	Filter	SALES Measure	Members	Amount	
4		Filter	Version	Members	Variance				Filter	Version	Members	Variance	
5		Filter	Region	Members	World				Filter	Region	Members	World	
6		K10)	Account	Range	$B$10:$F$10				Column	Account	Range	$I$10:$K$10	
7		Row	Month	Range	$A$11:$A$28				Row	Month	Range	$H$11:$H$22	
8													
9	OLAPivotTable							OLAPivotTable					
10		All	Sales	Margin	Cost of Sales	Margin Pcnt			Sales	Cost of Sales	Margin		
11	All	50685.10857	55672	60663	-4991	1.089650093		January	91995	-6563	98558	1R x 3C	
12	Total Quarter	50685.10857	55672	60663	-4991	1.089650093		February	-21285	-8537	-12748		
13	January	85431.04054	91995	98558	-6563	1.071340834		March	1214	1023	191		
14	1st Quarter	57846.00285	71924	86001	-14077	1.195720483		April	1700	921	779		
15	February	-29822.49618	-21285	-12748	-8537	0.598919427		May	1788	1046	742		
16	March	2237.458495	1214	191	1023	0.157331137		June	1785	1009	776		
17	April	2621.829651	1700	779	921	0.458235294		July	1666	1340	326		
18	2nd Quarter	8251.205852	5273	2297	2976	0.435615399		August	1391	2061	-670		
19	May	2834.656933	1788	742	1046	0.414988814		September	917	1122	-205		
20	June	2794.719269	1785	776	1009	0.434733894		October	-6223	526	-6749		
21	July	3006.656098	1666	326	1340	0.195678271		November	-9632	725	-10357		
22	3rd Quarter	8498.253552	3974	-549	4523	-0.13814796		December	-9644	336	-9980		
23	August	3451.883057	1391	-670	2061	-0.48166786							
24	September	2039.714397	917	-205	1122	-0.22355507							
25	October	-5696.342804	-6223	-6749	526	1.084525149							
26	4th Quarter	-23910.35369	-25499	-27086	1587	1.062237735							
27	November	-8906.880598	-9632	-10357	725	1.075269934							
28	December	-9307.130289	-9644	-9980	336	1.034840315							
29													

- Click the **Refresh** button in the PowerExcel Tab of the Excel ribbon. Notice that the first PivotTable is now updated.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Database:	USING_OLATION						Database:	USING_OLATION				
2	Cube:	SALES						Cube:	SALES				
3	Dimensions:	Filter	SALES Measu	Members	Amount			Dimensions:	Filter	SALES Measur	Members	Amount	
4		Filter	Version	Members	Variance				Filter	Version	Members	Variance	
5		Filter	Region	Members	World				Filter	Region	Members	World	
6		_Members)	Account	Range	$B$10:$D$10				Column	Account	Range	$I$10:$K$10	
7		Row	Month	Range	$A$11:$A$28				Row	Month	Range	$H$11:$H$22	
8													
9	OLAPivotTable							OLAPivotTable					
10		Sales	Cost of Sales	Margin					Sales	Cost of Sales	Margin		
11	All	55672	-4991	60663				January	91995	-6563	98558		
12	Total Quarter	55672	-4991	60663				February	-21285	-8537	-12748		
13	January	91995	-6563	98558				March	1214	1023	191		
14	1st Quarter	71924	-14077	86001				April	1700	921	779		
15	February	-21285	-8537	-12748				May	1788	1046	742		
16	March	1214	1023	191				June	1785	1009	776		
17	April	1700	921	779				July	1666	1340	326		
18	2nd Quarter	5273	2976	2297				August	1391	2061	-670		
19	May	1788	1046	742				September	917	1122	-205		
20	June	1785	1009	776				October	-6223	526	-6749		
21	July	1666	1340	326				November	-9632	725	-10357		
22	3rd Quarter	3974	4523	-549				December	-9644	336	-9980		
23	August	1391	2061	-670									
24	September	917	1122	-205									
25	October	-6223	526	-6749									
26	4th Quarter	-25499	1587	-27086									
27	November	-9632	725	-10357									
28	December	-9644	336	-9980									
29													
30													

16.OLATableSubset

Function Description: This function covers a 'cell range' or a 'group of cells' that define the Dimension and corresponding Members that will be displayed along the Rows or Columns of a PowerExcel Slice. The 'cell range' covered by this function must be updated simultaneously and changes will only be committed by use of the **CTRL+SHIFT+ENTER** keys. This function applies to the PowerExcel Power Query Table.

To change the Display Members along the column or row of a PowerExcel Slice, click on all the cells covered by the OLAPTableSubset function, change to the preferred 'Subset name' or the last parameter and enter the 'new target Subset'. Press the **Ctrl+Shift+Enter** keys to commit the changes. Notice that the update is reflected across all the cells covered by the function.

Syntax: OLATableSubset (Connection, Cube, AXIS, Index, Dimension, Subset)

Connection: The PowerExcel connection that contains the information about the Olation server URL and the source database name.

Cube: The name of the source/target Cube; or enter the cell reference that contains the name of the source/target Cube you wish to establish a connection to.

AXIS: This indicates the area of the PowerExcel Slice where the data will appear (i.e., Filter, Column or Rows)

Index: 0 [NOTE: When there are no "stacked Dimensions" in Row or Column, "0" will always show; If there are stacked Dimensions, the number will reflect the order of the Rows or Columns, starting with "1" as the "topmost" in the stack, and continuing.]

Dimension: The Dimension name or the cell reference that contains the name of the Dimension that exists within the specified Database above.

Subset: The Subset name corresponding to the target subset of Members to be displayed along the row or column.

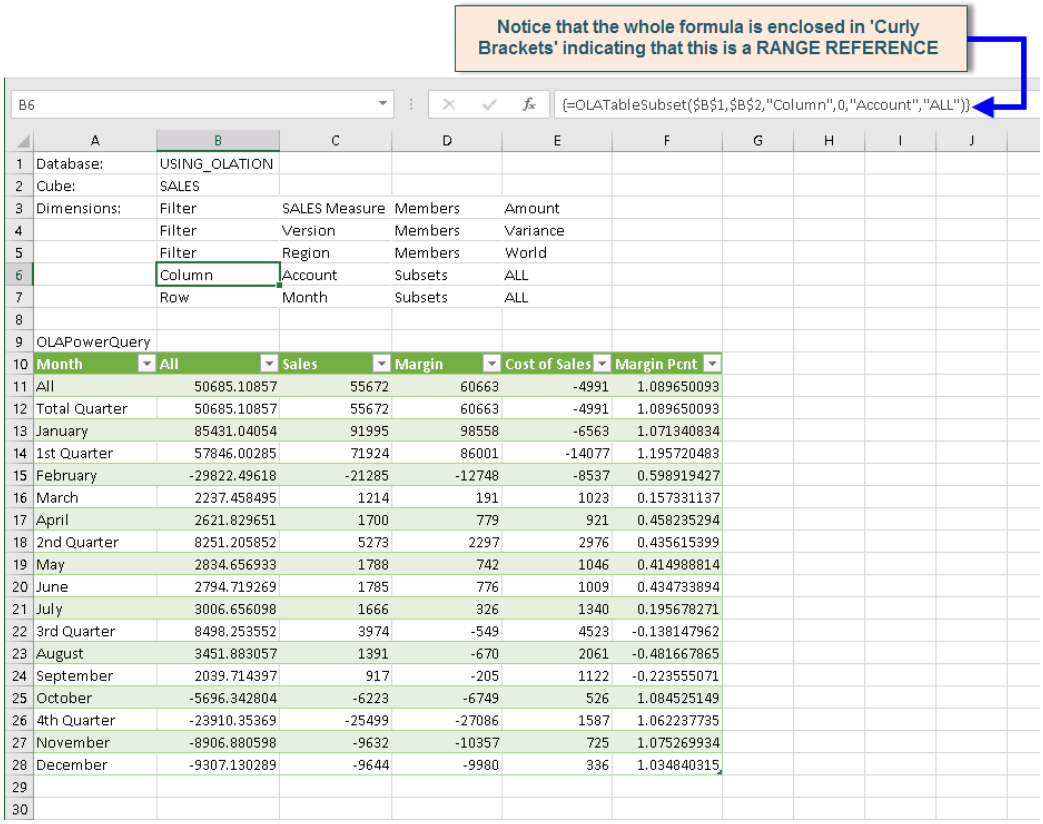
Remarks:

- The PowerExcel Connection must exist.
- The Olation Web Service must be running.
- The Database must be opened and running in the specified server.
- The Cube must exist within the specified database.
- The Dimension must exist within the specified Database.
- The Subset must exist within the specified Dimension.
- All the parameters are compulsory because the function covers a 'range of cells'
- It is imperative that 'All' cells governed by the OLATableRange function be updated in order to successfully commit the changes to the formula.

Example:

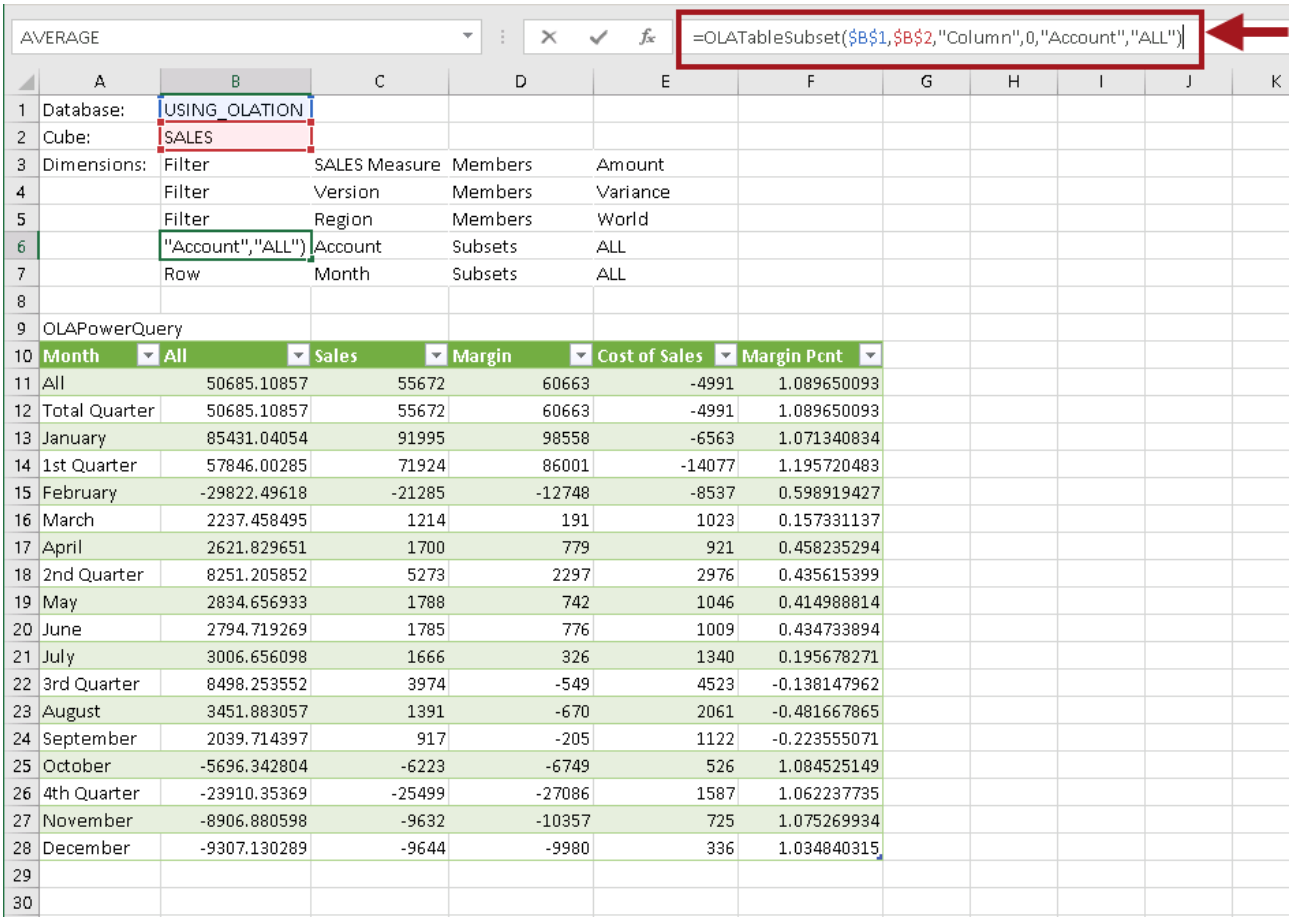
- The example Slice below shows a Power Query Table. The **OLATableSubset** function is used in the PowerExcel Slice to define what Members to display along the Row or Column by picking a pre-defined Subset of Members.
- The OLATableSubset function is a 'Range Reference', meaning it is a formula function that governs a group of cells. Hence, if you click on any cell containing the formula, you will notice that the formula is enclosed in 'Curly Brackets', an indication that this is a Range Reference. And all cells covered by the same Range reference will display the same formula.
- As in the example screenshot below, the **OLATableSubset** formula for Columns can be found in the cells **B6:E6** (*Account* Dimension on Columns) and for Rows in cells **B7:E7** (*Month* Dimension on Rows). If we look at the **OLATableSubset** formula for *Account* Dimension, clicking through all the cells governed by this specific Range Reference formula will display the same formula.
- When you click on the cell containing the **OLATableSubset** formula (in the example, the active cell is at cell **B6**), notice that the **OLATableSubset** formula that appears in the formula bar is enclosed in curly brackets. When you click through the cells that are part of the cell range (**C6**, **D6** and **E6**), notice that it will show the exact formula as in cell **B6**:

{=OLATableSubset($B$1,$B$2,"Column",0,"Account","ALL")}



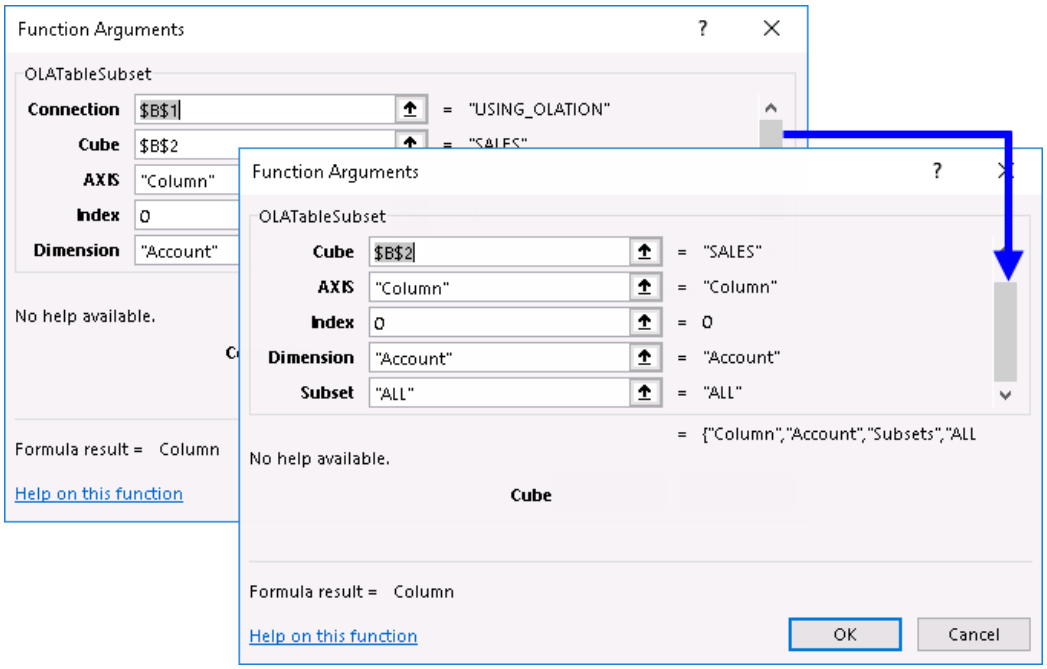
	A	B	C	D	E	F	G	H	I	J
1	Database:	USING_OLATION								
2	Cube:	SALES								
3	Dimensions:	Filter	SALES Measure	Members	Amount					
4		Filter	Version	Members	Variance					
5		Filter	Region	Members	World					
6		Column	Account	Subsets	ALL					
7		Row	Month	Subsets	ALL					
8										
9	OLAPowerQuery									
10	Month	All	Sales	Margin	Cost of Sales	Margin Pcnt				
11	All	50685.10857	55672	60663	-4991	1.089650093				
12	Total Quarter	50685.10857	55672	60663	-4991	1.089650093				
13	January	85431.04054	91995	98558	-6563	1.071340834				
14	1st Quarter	57846.00285	71924	86001	-14077	1.195720483				
15	February	-29822.49618	-21285	-12748	-8537	0.598919427				
16	March	2237.458495	1214	191	1023	0.157331137				
17	April	2621.829651	1700	779	921	0.458235294				
18	2nd Quarter	8251.205852	5273	2297	2976	0.435615399				
19	May	2834.656933	1788	742	1046	0.414988814				
20	June	2794.719269	1785	776	1009	0.434733894				
21	July	3006.656098	1666	326	1340	0.195678271				
22	3rd Quarter	8498.253552	3974	-549	4523	-0.138147962				
23	August	3451.883057	1391	-670	2061	-0.481667865				
24	September	2039.714397	917	-205	1122	-0.223555071				
25	October	-5696.342804	-6223	-6749	526	1.084525149				
26	4th Quarter	-23910.35369	-25499	-27086	1587	1.062237735				
27	November	-8906.880598	-9632	-10357	725	1.075269934				
28	December	-9307.130289	-9644	-9980	336	1.034840315				
29										
30										

- By clicking in the formula bar area (as can be seen in the image below, the mouse cursor is placed at the end of the formula), it will show the cell references corresponding to the **OLATableSubset** function. The **OLATableSubset** function returns data for **Column**(B6), **Account**(C6), **Subsets**(D6) and **ALL**(E6).
- It is also referencing to the Cube called "SALES"(**$B$2**) and is using the PowerExcel connection/Database connection called "USING_OLATION"(**$B$1**).
  Note: When you are on edit mode of a Range Reference type of formula, notice that the 'Curly Brackets' disappear.



AVERAGE | =OLATableSubset($B$1,$B$2,"Column",0,"Account","ALL")

	A	B	C	D	E	F	G	H	I	J	K
1	Database:	USING_OLATION									
2	Cube:	SALES									
3	Dimensions:	Filter	SALES Measure	Members	Amount						
4		Filter	Version	Members	Variance						
5		Filter	Region	Members	World						
6		"Account","ALL")	Account	Subsets	ALL						
7		Row	Month	Subsets	ALL						
8											
9	OLAPowerQuery										
10	Month	All	Sales	Margin	Cost of Sales	Margin Pcnt					
11	All	50685.10857	55672	60663	-4991	1.089650093					
12	Total Quarter	50685.10857	55672	60663	-4991	1.089650093					
13	January	85431.04054	91995	98558	-6563	1.071340834					
14	1st Quarter	57846.00285	71924	86001	-14077	1.195720483					
15	February	-29822.49618	-21285	-12748	-8537	0.598919427					
16	March	2237.458495	1214	191	1023	0.157331137					
17	April	2621.829651	1700	779	921	0.458235294					
18	2nd Quarter	8251.205852	5273	2297	2976	0.435615399					
19	May	2834.656933	1788	742	1046	0.414988814					
20	June	2794.719269	1785	776	1009	0.434733894					
21	July	3006.656098	1666	326	1340	0.195678271					
22	3rd Quarter	8498.253552	3974	-549	4523	-0.138147962					
23	August	3451.883057	1391	-670	2061	-0.481667865					
24	September	2039.714397	917	-205	1122	-0.223555071					
25	October	-5696.342804	-6223	-6749	526	1.084525149					
26	4th Quarter	-23910.35369	-25499	-27086	1587	1.062237735					
27	November	-8906.880598	-9632	-10357	725	1.075269934					
28	December	-9307.130289	-9644	-9980	336	1.034840315					
29											
30											



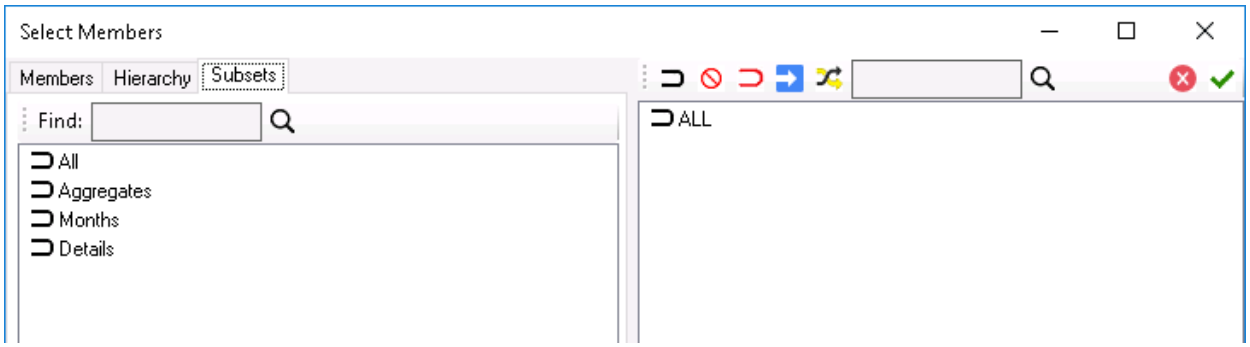
Cell References:

```
={OLATableSubset($B$1,$B$2,"Column",0,"Account","ALL")
```

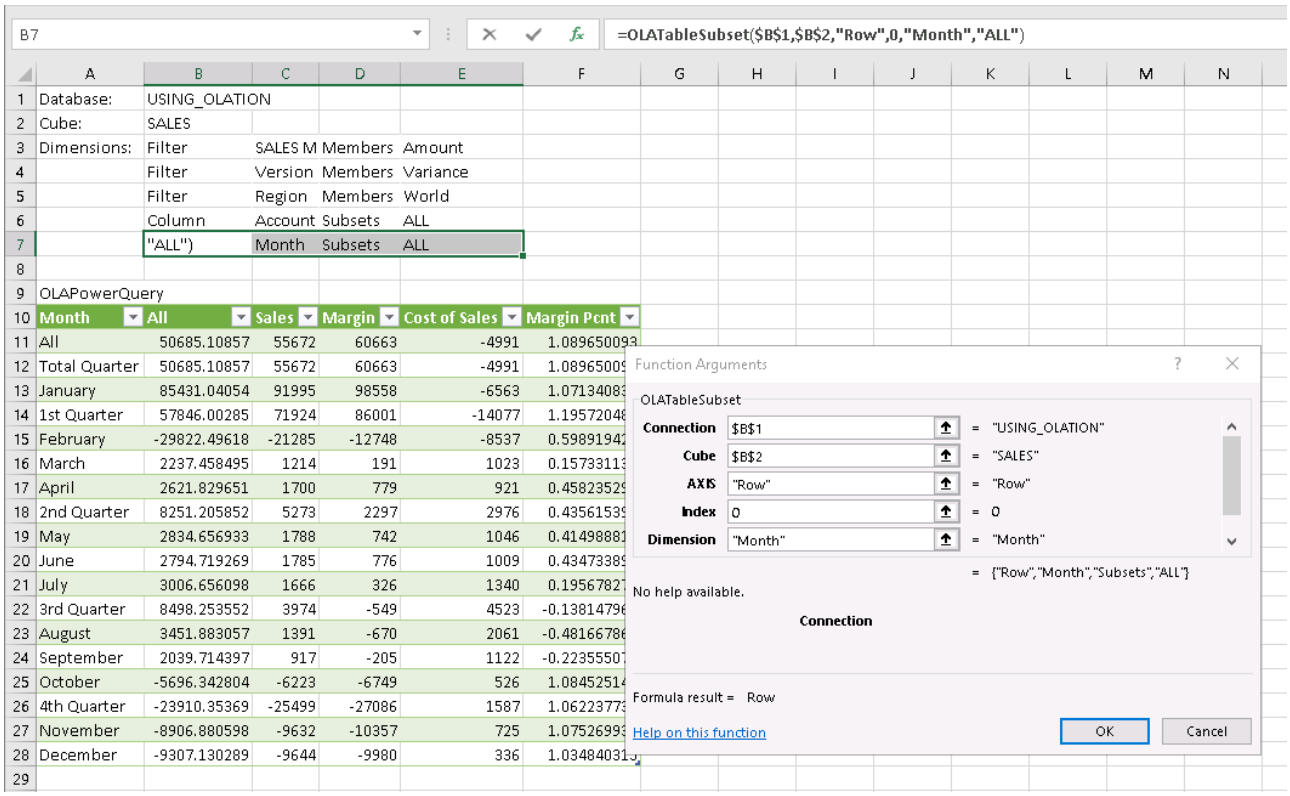
- $B$1– the Database name or the PowerExcel connection name/cell reference that contains the name of the Database or the PowerExcel connection, i.e., **USING_OLATION**
- $B$2– the Cube name or the cell reference that contains the name of the Cube, i.e., **SALES**
- "Column"– this indicates the data will appear along the Column area of the Slice
- 0– this indicates the index number: "0" if there are no stacked Dimensions in Columns or Rows; if either are stacked, the index number will be "1" for the topmost Dimension, "2" for the next, etc.
- "Account"– the Dimension name that exists within the database
- "ALL"– this range corresponds to the target display Members that will dictate the Account members displayed along the column of the Slice output

To change the Subset displayed by changing the Subset parameter (note that this example concerns the Rows, for the *Month* Dimension, in order to use a Subset in this dimension).

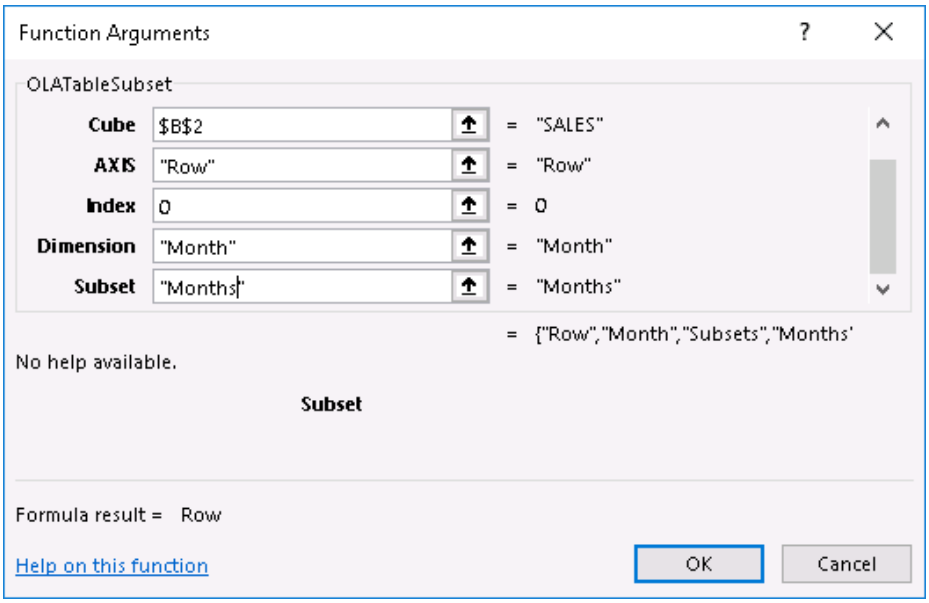
Assuming we know that these are the Subsets that exist within the Month Dimension:



- Click and Highlight the cells **B7 to E7** (OLATableSubset formula for the Month Dimension along the Rows), then click on the **Function** button. The Functions Arguments dialog box appears:
  Notice that it displays the particulars of the cell references.



- Scroll down to the Subset parameter and change it to a new preferred subset—e.g., **"Months"**—then click **OK**.
  Note: Make sure that the parameter is enclosed in double quotes.



- Click **OK**. Then click **Refresh** the PowerExcel Slice ribbon. The PowerExcel Power Query Slice is now updated to show the new display Members along the rows as shown below:

B7

	A	B	C	D	E	F	G	H
1	Database:	USING_OLATION						
2	Cube:	SALES						
3	Dimensions	Filter	SALES M	Members	Amount			
4		Filter	Version	Members	Variance			
5		Filter	Region	Members	World			
6		Column	Account	Subsets	ALL			
7		Row	Month	Subsets	Months			
8								
9	OLAPowerQuery							
10	Month	All	Sales	Margin	Cost of Sales	Margin Pcnt		
11	January	85431.04054	91995	98558	-6563	1.071340834		
12	February	-29822.49618	-21285	-12748	-8537	0.598919427		
13	March	2237.458495	1214	191	1023	0.157331137		
14	April	2621.829651	1700	779	921	0.458235294		
15	May	2834.656933	1788	742	1046	0.414988814		
16	June	2794.719269	1785	776	1009	0.434733894		
17	July	3006.656098	1666	326	1340	0.195678271		
18	August	3451.883057	1391	-670	2061	-0.481667865		
19	September	2039.714397	917	-205	1122	-0.223555071		
20	October	-5696.342804	-6223	-6749	526	1.084525149		
21	November	-8906.880598	-9632	-10357	725	1.075269934		
22	December	-9307.130289	-9644	-9980	336	1.034840315		
23								

17.OLAWrite

Description: The OLAWrite Formula function allows a user to write values back to a precise multidimensional data point in an Olation database model from a PowerExcel Slice.

Syntax: OLAWrite (Connection, Cube, Value1, Value2,…,ValueN)

Connection: The PowerExcel connection that contains the information about the Olation server URL and the source database name.

Cube: The name of the source/target Cube; or the cell reference that contains the name of the source or target Cube you wish to establish a connection to.

Value1 to Value N: The related Member references.

Last Value parameter: The write value or the cell reference that contains the data value to be written back to a specific data point to a target Database.

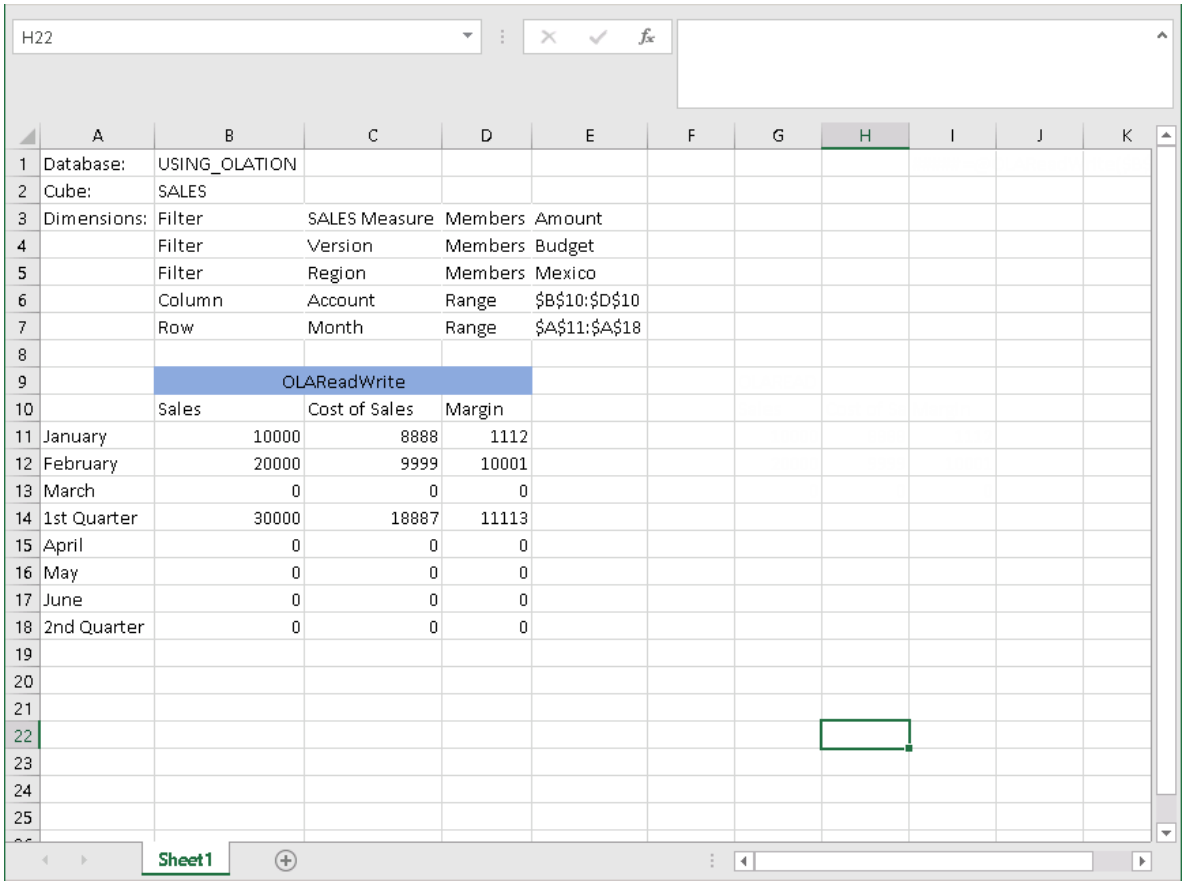
Remarks:

- The PowerExcel Connection must exist.
- The Olation Web Service must be running.
- The Database must be opened and running in the specified server.
- The 'Connection' and 'Cube' parameters are compulsory.
- The Value parameters prior to the 'last value parameter' refer to Member references that identify the specific data point where the data value should be written to.
- The 'Last Value parameter' is either the actual write value or the cell that contains the data value or number which will be written back to the target data point within a database.

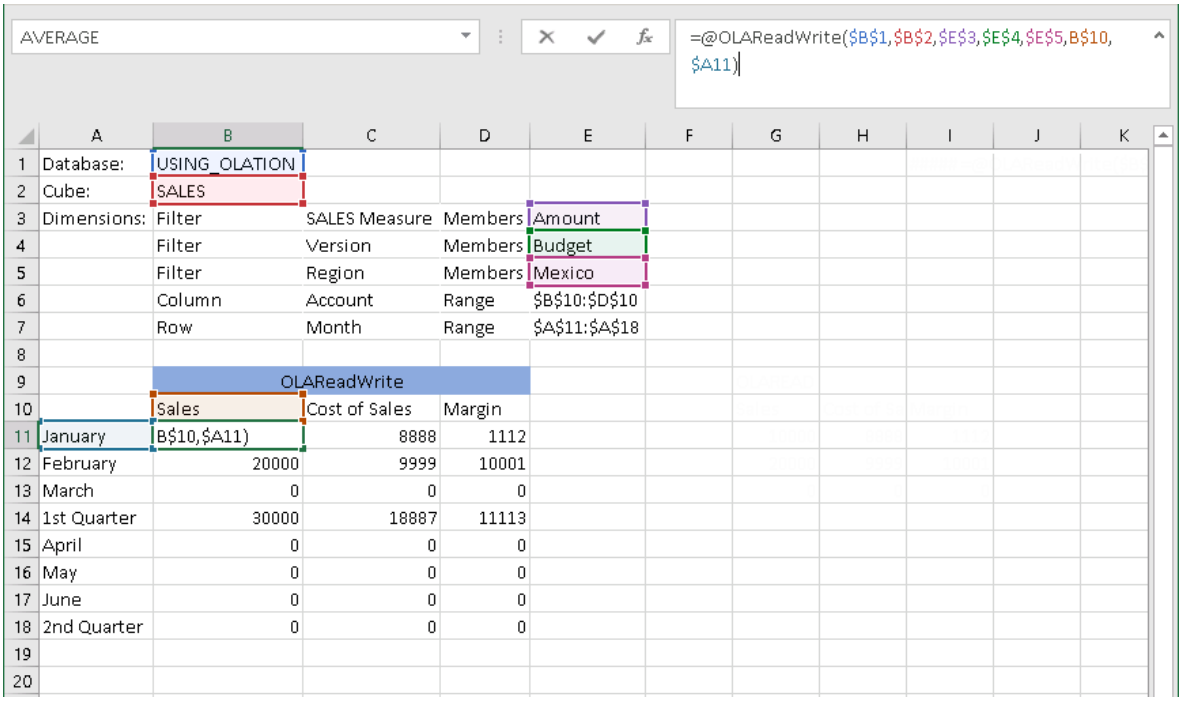
Example 1:

First we will show how to create an OLAWrite formula to write a value to a specific intersection in a target database--for this example, to a data point in the *SALES* Cube of the *USING_OLATION* database.

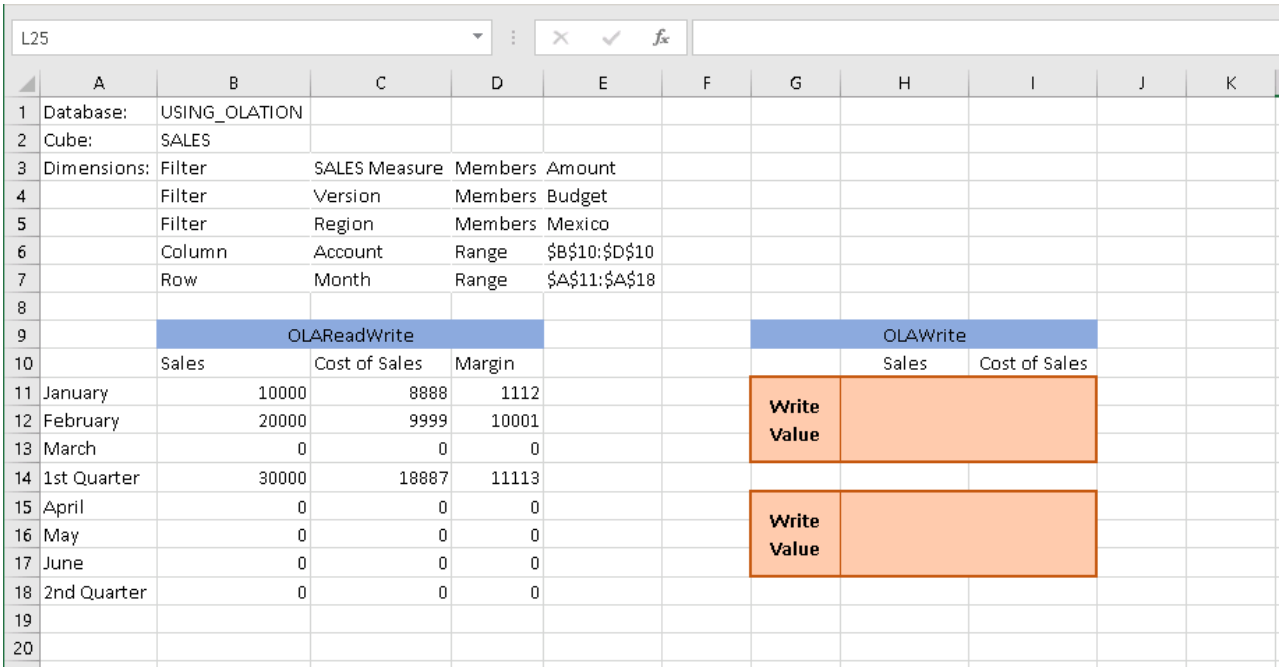
- First create a PowerExcel ReadWrite Slice with the orientation of your choosing (e.g., the following image will serve as an example).
  Note: We placed a heading (in Row 9), "OLAReadWrite", at the top of the sample table so we can identify that the cells below are governed by this type of formula.



- Note that in an OLAReadWrite Slice, each cell contains an individual formula function, as in the following image (Cell B11 has been clicked on). As we will see, an **OLAWrite Formula function** also appertains to an individual cell.



- Next to create an OLAWrite formula—in this case, to write a new *Sales* value for the month of *March*.
  (In the next image, the area where the OLAWrite formulas will go is shaded orange for easy identification.)

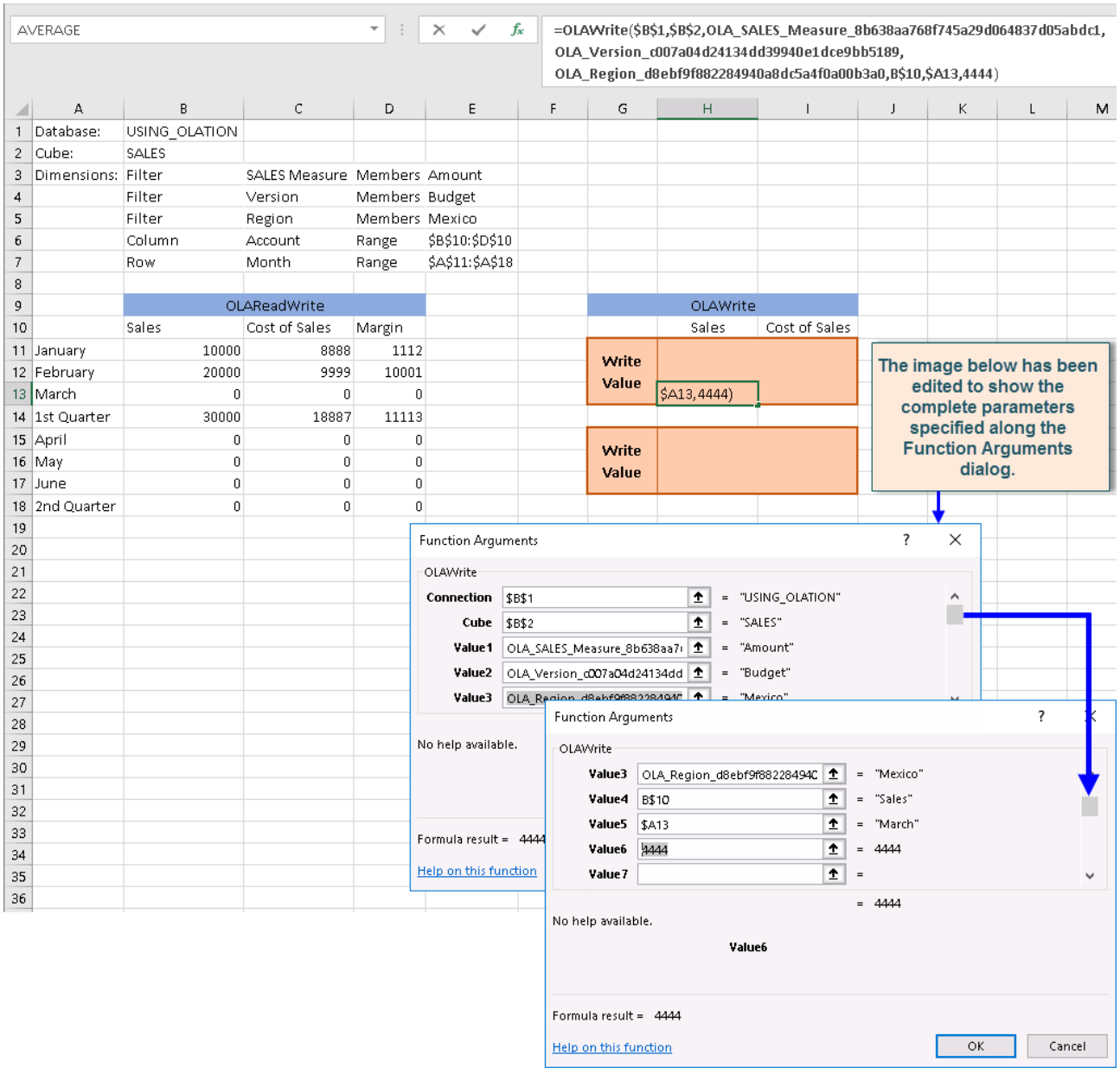


	A	B	C	D	E	F	G	H	I	J	K
1	Database:	USING_OLATION									
2	Cube:	SALES									
3	Dimensions:	Filter	SALES Measure	Members	Amount						
4		Filter	Version	Members	Budget						
5		Filter	Region	Members	Mexico						
6		Column	Account	Range	$B$10:$D$10						
7		Row	Month	Range	$A$11:$A$18						
8											
9		OLAReadWrite					OLAWrite				
10		Sales	Cost of Sales	Margin				Sales	Cost of Sales		
11	January	10000	8888	1112			Write Value				
12	February	20000	9999	10001							
13	March	0	0	0							
14	1st Quarter	30000	18887	11113							
15	April	0	0	0			Write Value				
16	May	0	0	0							
17	June	0	0	0							
18	2nd Quarter	0	0	0							
19											
20											

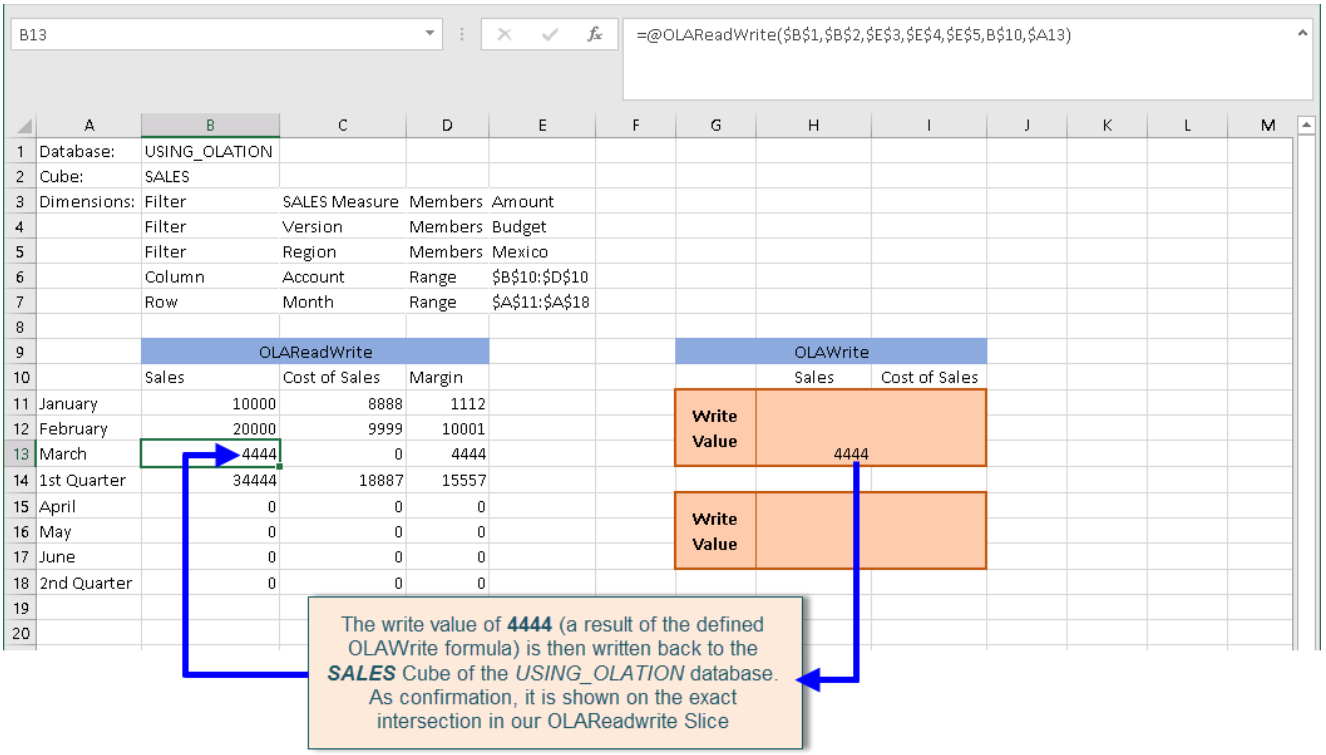
- Define the **OLAWrite** formula: in Cell **H13** click the **Function** button beside the formula bar. In the Insert Function dialog that appears, choose **PowerExcel.ExcelFunctions** as the category, select **OLAWrite** from the function list and click **OK**.
- In the Function Arguments dialog box, define the parameters:
  - Click on Connection field and then click on cell **B1** (the cell reference for *USING_OLATION* database). Use an Absolute reference so the connection reference appears as **$B$1**.
  - Click on the Cube field, then click on cell **B2** (the cell reference for the *SALES* cube); again, use an absolute reference (**$B$2**).
  - Click on the Value 1 field, then click on cell **E3** (the cell reference for the Filter Member *Amount* in the *SALES Measure* Dimension).
  - Click on the Value 2 field, then click on cell **E4** (the cell reference for the Filter Member *Budget* of the *Version* Dimension).
  - Click on the Value 3 field, then click on cell **E5** (the cell reference for the Filter Member *Mexico* of the *Region* Dimension).
  - Click on the Value 4 field, then click on cell **B10** (which is the cell reference for the Column Member *Sales* of the *Account* Dimension); use the absolute row reference, thus **B$10**.
  - Click on the Value 5 field, then click on cell **A13** (the cell reference for the Row Member *March* of the *Month* Dimension); use the absolute column reference, thus **$A13**.

- Lastly, click on the Value 6 field, then type a new *Sales* value for the month of *March*—for example, **4444**. This is our write value.

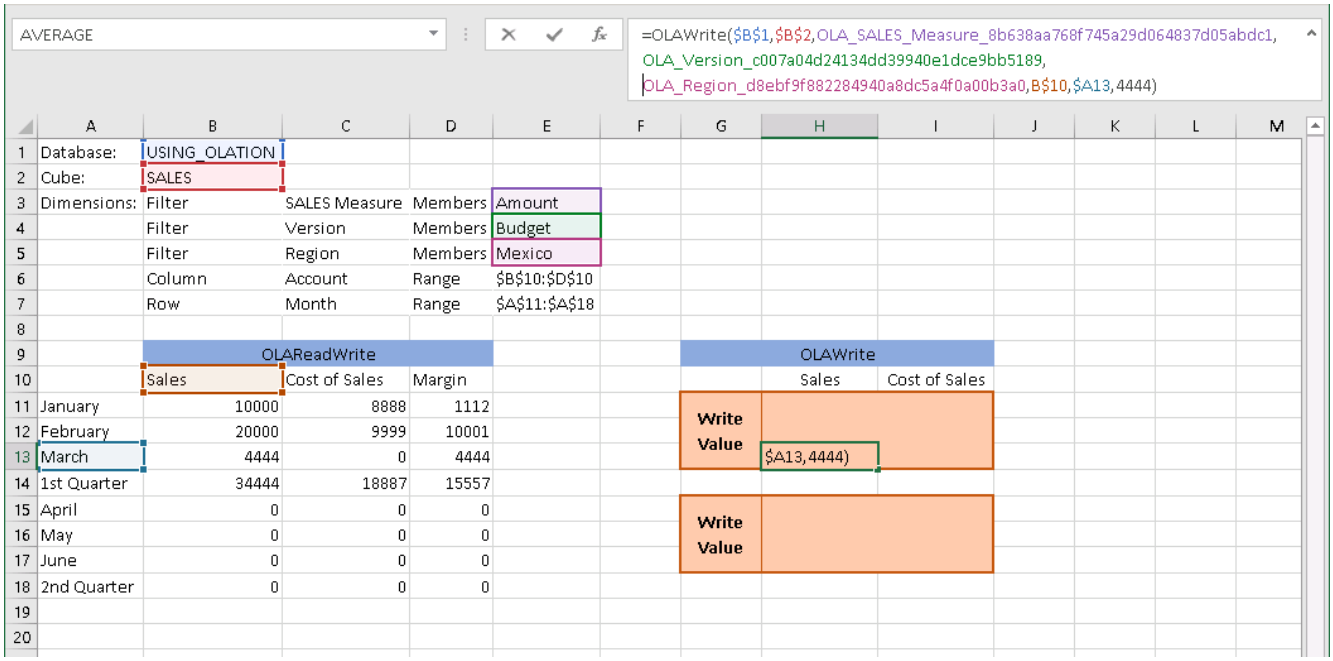
  Note: Notice that the corresponding Database, Cube and other references appear in each corresponding field.

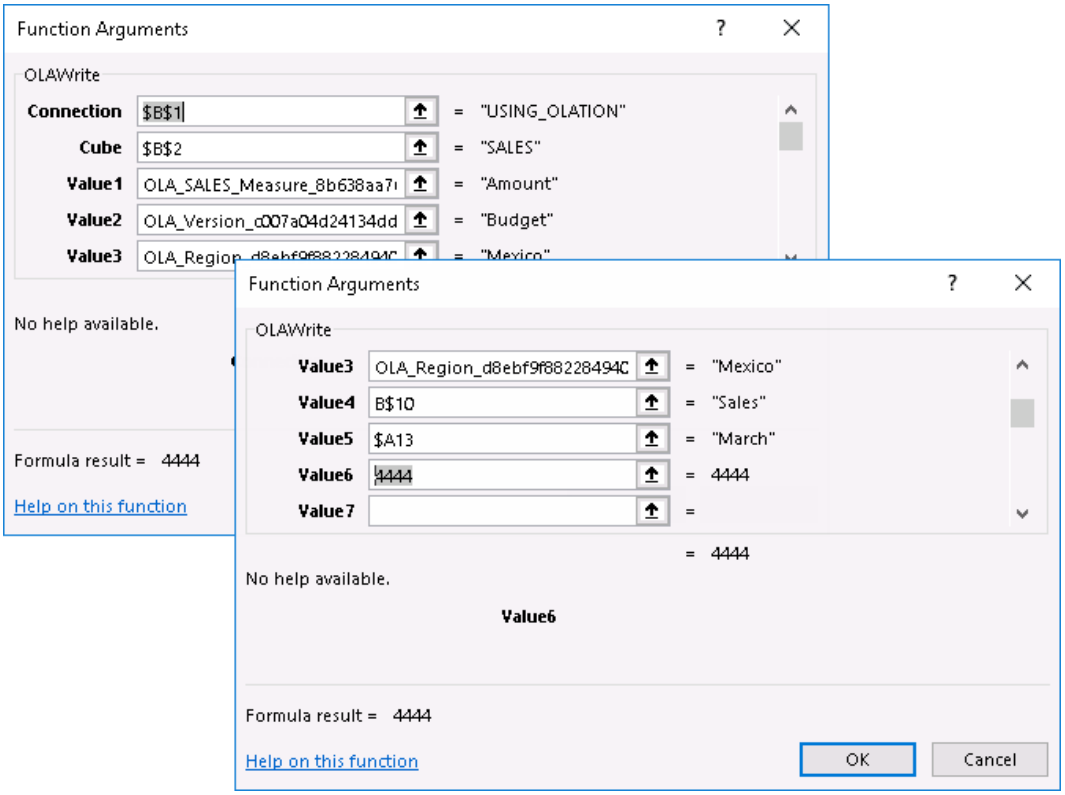


- Click **OK**. Press **Enter**, then click the **Refresh** button along the PowerExcel Tab of the Excel ribbon. Notice the intersection in the ReadWrite section of the Slice, Cell **B13**, now has the value **4444**.



- Click on the **OLAWrite** formula in cell **H13,** then click on the formula bar (notice that the cursor is at the end of the formula). This will show the corresponding cell references of the formula.



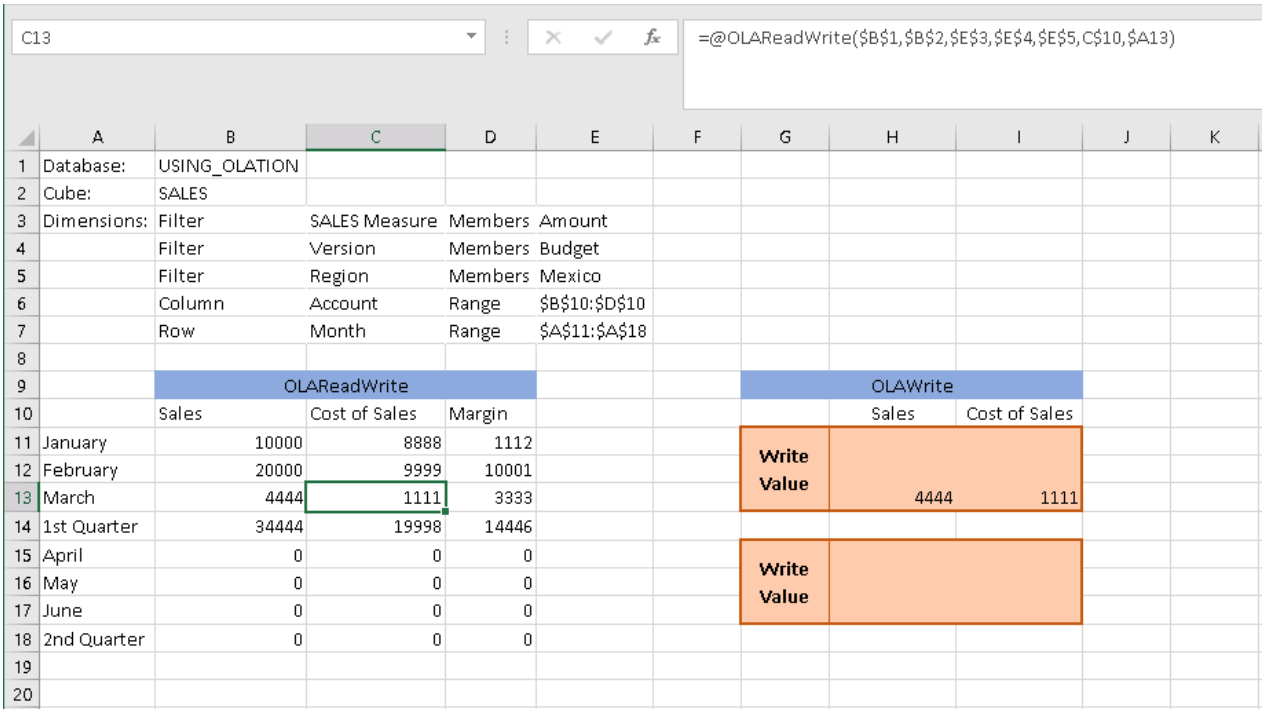


Cell References:

```
=OLARead($B$1,$B$2,OLA_SALES_Measure_8b638aa768f745a29d064837d05abdc1,OLA_Version_c007a04d24134dd39940e1dce9bb5189,OLA_Region_d8ebf9f882284940a8dc5a4f0a00b3a0,B$10,$A13,4444)
```

- $B$1– the Database name or the PowerExcel connection name/cell reference that contains the name of the Database or the PowerExcel connection, i.e., **USING_OLATION**
- $B$2 – the Cube in the Database, i.e., **SALES** Cube
- OLA_SALES_Measure_8b638aa768f745a29d064837d05abdc1 – the ***Amount*** Member in the *SALES Measure* Dimension [Filter reference]
- OLA_Version_c007a04d24134dd39940e1dce9bb5189 – the ***Budget*** Member in the *Version* Dimension [Filter reference]
- OLA_Region_d8ebf9f882284940a8dc5a4f0a00b3a0 – the ***Mexico*** Member in the *Region* Dimension [Filter reference]
- B$10 – the Column Member reference ***Sales*** from the *Account* Dimension [Column reference].
- $A13 – the Row Member reference ***March*** from the *Month* Dimension [Row reference].
- 4444 – the new Sales value for March to be written back to the database.

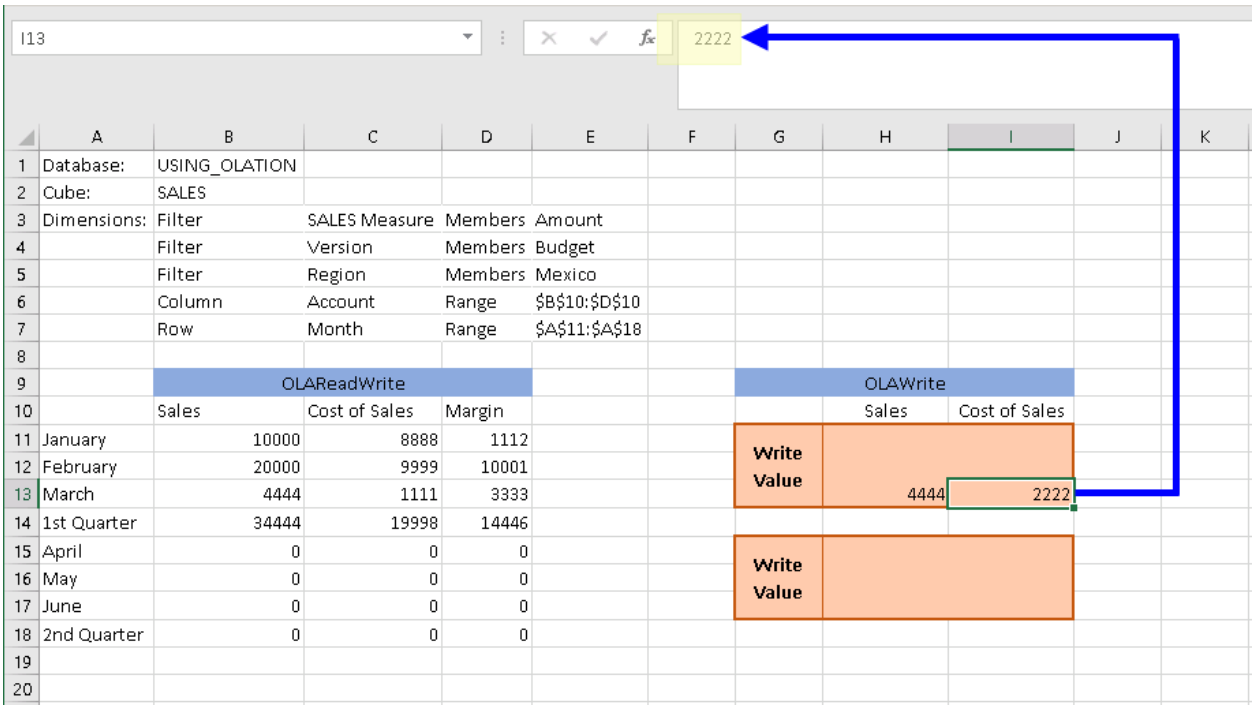
- Next, write the same formula in cell **I13**, but this time, change the month reference to **Cost of Sales** (**C$10**) and the write value to **1111**. Press **Enter** then click the **Refresh** button along the PowerExcel Tab of the Excel ribbon.



C13 =@OLAReadWrite($B$1,$B$2,$E$3,$E$4,$E$5,C$10,$A13)

	A	B	C	D	E	F	G	H	I	J	K
1	Database:	USING_OLATION									
2	Cube:	SALES									
3	Dimensions:	Filter	SALES Measure	Members	Amount						
4		Filter	Version	Members	Budget						
5		Filter	Region	Members	Mexico						
6		Column	Account	Range	$B$10:$D$10						
7		Row	Month	Range	$A$11:$A$18						
8											
9		OLAReadWrite					OLAWrite				
10		Sales	Cost of Sales	Margin				Sales	Cost of Sales		
11	January	10000	8888	1112			Write Value				
12	February	20000	9999	10001							
13	March	4444	1111	3333				4444	1111		
14	1st Quarter	34444	19998	14446							
15	April	0	0	0			Write Value				
16	May	0	0	0							
17	June	0	0	0							
18	2nd Quarter	0	0	0							
19											
20											

- Once again, the value is saved back to the *USING_OLATION* database: the new *Cost of Sales* value appears at the same intersection in the PowerExcel ReadWrite section of the Slice (cell **C13**).
- Notice also that the relative aggregate points are updated as well (*1st Quarter* values and *Margin* values).
- When you click over the cells with an OLAWrite formula, the corresponding formulas will appear in the formula bar. However, unlike the OLAReadWrite formula, where typing a value will not erase the formula, if you type in a value in the cell with an OLAWrite formula, the formula will be deleted and replaced with the value you have entered.
  As an example, in Cell **I11**, type **2222** and click **Enter**.



	A	B	C	D	E	F	G	H	I	J	K
1	Database:	USING_OLATION									
2	Cube:	SALES									
3	Dimensions:	Filter	SALES Measure	Members	Amount						
4		Filter	Version	Members	Budget						
5		Filter	Region	Members	Mexico						
6		Column	Account	Range	$B$10:$D$10						
7		Row	Month	Range	$A$11:$A$18						
8											
9		OLAReadWrite					OLAWrite				
10		Sales	Cost of Sales	Margin				Sales	Cost of Sales		
11	January	10000	8888	1112			Write Value				
12	February	20000	9999	10001							
13	March	4444	1111	3333				4444	2222		
14	1st Quarter	34444	19998	14446							
15	April	0	0	0			Write Value				
16	May	0	0	0							
17	June	0	0	0							
18	2nd Quarter	0	0	0							
19											
20											

- Note that when you click on the cell, the OLAWrite formula is gone and the formula bar shows, simply, the value 2222.