
Title: Db2 Web Query for i Education Series: Module 4 – Build your first Synonym

Assumptions: You have installed Db2 Web Query using our “EZ-Install” package

Pre-Reqs: Modules 1-3

Table of Contents

1 *Getting Started with Metadata*..... 2

 1.1.1 Synonyms/Metadata Overview 2

 1.1.2 Creating Synonyms 2

 1.1.3 Joining Data 20

 1.1.4 Defining Dimensions for Auto Drill Downs 27

2 *Additional Information: Links*.....34

1 Getting Started with Metadata

1.1.1 Synonyms/Metadata Overview

Before you continue on with this section, please read through the chapter on metadata in the "Nuts and Bolts" Redbook: <http://ibm.biz/db2wq-nutsandbolts>.

A key concept in Db2 Web Query is that you do not build reports directly over your files/tables. You build reports over a representation of the data source. In Db2 Web Query, this representation of your data source are called SYNONYMS. The data source can certainly be files/tables, but also SQL statements (as SQL Views or scripts), Stored Procedures, uploaded Excel files, and even Query/400 definitions. With Standard Edition, you can also access data in non Db2 relational databases like MS SQLServer.

There are many benefits to using synonyms, including defining file relationships with joins (while forcing the join definition), creating virtual (or derived or calculated) fields using an expression builder and many of the 100+ built in functions or Db2 functions. In essence, synonyms allow you to simplify the complex database structures you may have in place such that the report author does not have to have the depth of database knowledge that you otherwise would need when using tools like Query/400. The ability to define virtual/derived fields in the metadata synonym provides this perspective of "single version of the truth" so that calculated fields are defined ONCE even if that data element is used in 100 different reports.

You can also create hierarchical data relationships with related data elements. This enables auto drill downs and "chained" parameters in your reports.

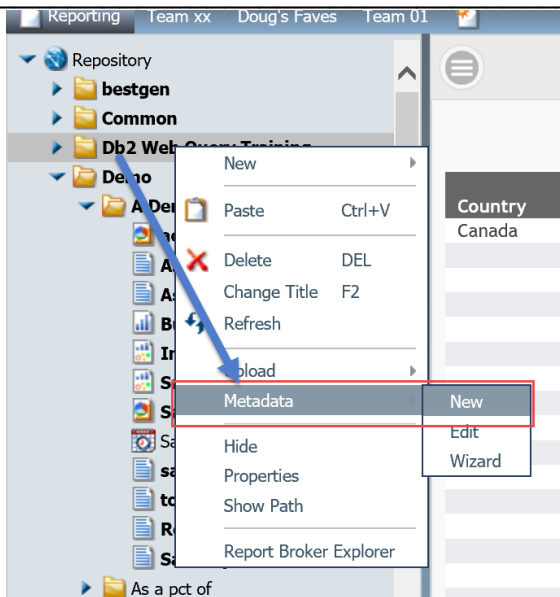
1.1.2 Creating Synonyms

We will use the sample database to show you how to create synonyms (we need to, as we don't have your data). But the process would be the same for your data.

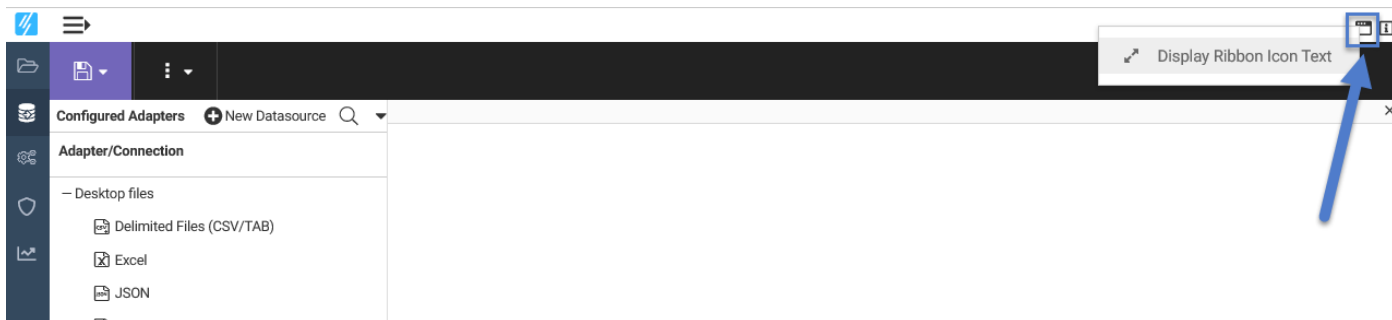
In order to work with metadata (called synonyms) from the BI Portal, you either need to be a Db2 Web Query Administrator or have that person add you to the "DBA" group of any top-level folder you want to work in (in this case, perhaps the Db2 Web Query Training folder you worked with previously). If you are not an admin, and don't have DBA rights, you cannot create/edit metadata and won't even see the metadata options on the BI Portal folder tree.

The following steps will create unique synonyms for five tables in the sample database, library QWQCENT. Then you'll join those files into a single synonym for use by reports.

Sign into the BI Portal if you are not already signed in. Right click on the Db2 Web Query Training folder and select METADATA -> NEW.

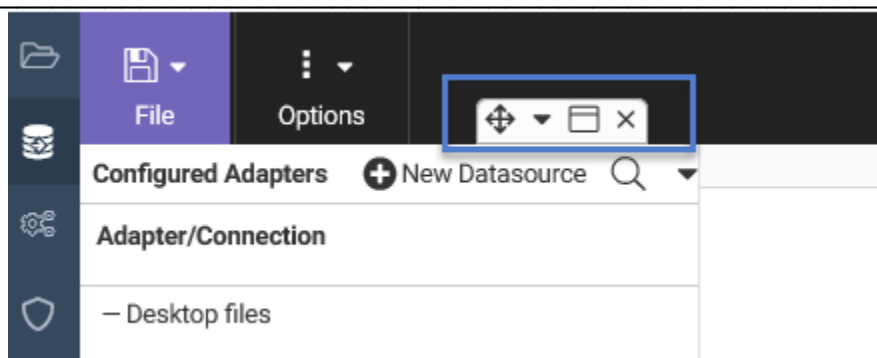


The web console for working with data opens. You may want to add text to the toolbar ribbon at the top. **Click on Toggle Ribbon Icon Text** at the upper right hand screen and **select Display Ribbon Icon Text**.



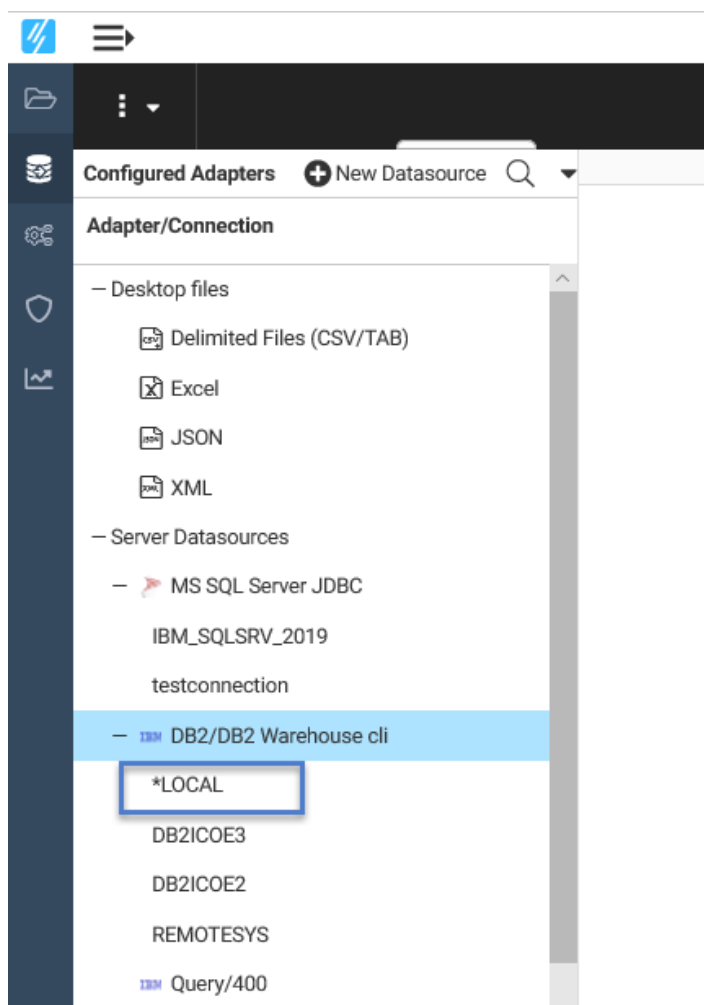
One other usability hint. In the above screen shot you only see the one panel for Configured Adapters. But you'll see as you start working in this user interface you could have multiple panels open side by side.

If you hover the upper right-hand portion of any panel you'll see a little pop-up to be able to control that panel (hide it, move it, maximize or close it).



Leave that as is for now.

You should see the ADAPTERs section available for you to select the adapter to create your synonym(s) over. In Db2 Web Query, we use a CLI adapter for Db2 databases. Make sure the Db2/Db2 Warehouse CLI section is open. **Double click on *LOCAL** (*LOCAL represents the Db2 for i database in the same LPAR/Server where Db2 Web Query is installed).



Note that the default object type is TABLES (SQL term for FILE). You might want to click on the down arrow in the Object Type drop down box and look at some of the other types of things you can build synonyms over. For now, leave this as Tables.

Available Objects for DB2/DB2 Warehouse (*LOCAL)

Filter

? Object Type: Tables

? Library:

? Object Name:

> Miscellaneous

> Customize data

? Create: ☐ Create ☐ Existing

? Application: db2_web_query_training ... ? Prefix: ? Suffix:

Row Limit: 50

	Default Synonym Name	Table Name	Library/Schema	Type
☐	bindata	BINDATA	QGPL	TABLE
☐	centoutput	CENTOUTPUT	QGPL	TABLE
☐	century_electronics_adhocrequest	CENTURY_ELECTRONICS_ADHOCREQUEST	QGPL	TABLE
☐	century_electronics_rev_cogs	CENTURY_ELECTRONICS_REV_COGS	QGPL	TABLE

Enter in QWQCENT in the library name text entry box, and select NEXT:

Filter

? Object Type: Tables

? Library: QWQCENT

? Object Name:

Sample: wrd%

Search

Later you will learn about EZ-report and the Query/400 definition capabilities to auto-create a synonym and report. But for now, we want to create a "join" or "cluster" synonym containing five files/tables of interest that will be joined together.

Here is where you'll select the files/tables of interest. Make sure there is a check box next to each of these tables:

DATE_CONV

INVENTORY
ORDERS
STORES
PLANTS

Also, make sure you add "Teamxx_" into the PREFIX text box:

Filter

Object Type

Tables

Library

qwqcent

Supply value to avoid system-wide search, sample: S999

Object Name

Sample: wrd%

Miscellaneous Parameters

Customize data type mappings

Create:

☐ Cluster Synonym
 ☒ Base Synonyms

Application

db2_web_query_training

Prefix

teamxx_

Suffix

Row Limit

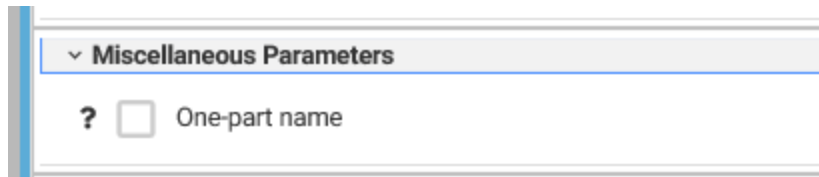
50

☐	Default Synonym Name	Table Name	Library/Schema	Type
☒	date_conv	DATE_CONV	QWQCENT	TABLE
☐	date_dim_t	DATE_DIM_T	QWQCENT	TABLE
☐	hr	HR	QWQCENT	TABLE
☒	inventory	INVENTORY	QWQCENT	TABLE

The reason for using a prefix in this case is to simply identify your synonyms as being your team's synonyms should multiple people be doing this on your system. In reality, you would probably want to use a prefix to associate the synonyms with a specific set of data, or an application such as "Finance_" or "SALES_" tables. The underscore separates the prefix from the table name for readability purposes.

The suffix is also a nice way to identify what type of data the synonym is built over. For instance, if you were building a synonym over a Query/400 definition, you might use "_qry400", or if it were a Microsoft SQLServer table, something like "_SQLSRV."

There is probably one other important thing to note on this screen here that has to do with library list management. **Open the Miscellaneous Parameters** information and note the One Part Name check box.



By default, it is not checked. What this means is that a TWO-PART name will be created in the synonym by default. In other words, the file/table (in this case) will have its library name hard coded in the synonym. A ONE PART name, if that box is checked, will leave the library name out of the synonym and therefore give you flexibility to have reports accessing a library by the specific user's library list (or set by something called a Run Time Environment which you can read about on the wiki at <http://ibm.co/db2wqwiki>).

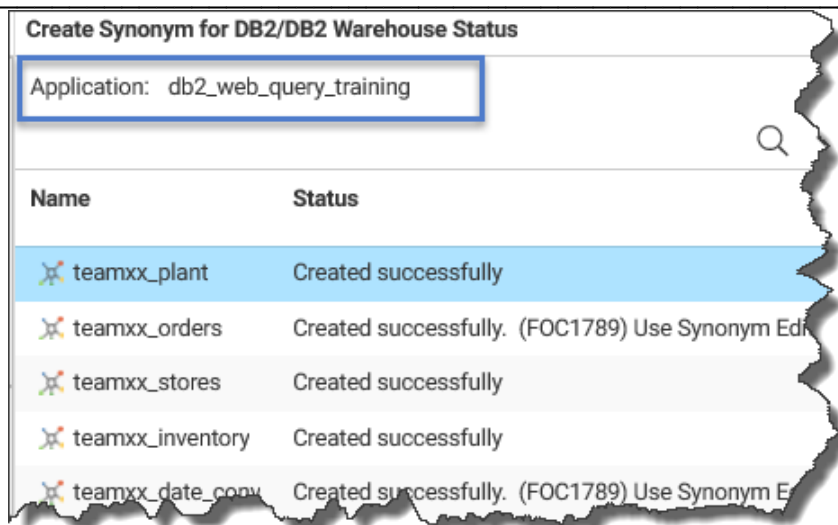
Best practice is to use one-part names if you want library list to drive access to the file/table that might exist in multiple libraries (like CUST1 and CUST2 libraries that contain the same file/table structures but unique data). Keep it unchecked if you want reports to always go to the file/table with the library name hard coded in the synonym.

For now, leave it unchecked as there is only one of these sample database libraries on your system.

Click on ARROW button at the top of the window.



You should see "created successfully" messages.

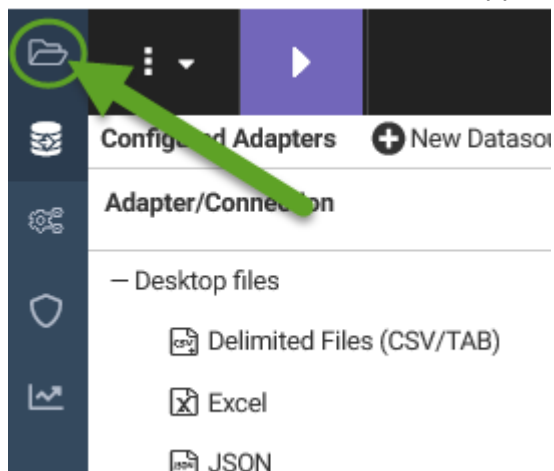


There is quite a bit of explaining about metadata that goes beyond this “test drive” but, one important concept is that of application folders. If you recall from above on the BI Portal tree, there are many folders underneath the “repository” icon on the left-hand side. The highest level folders, such as the one you created called Db2 Web Query Training, are considered to be **TOP-LEVEL** folders (sometimes also known as domains). As noted above, you store reporting objects (reports, graphs, dashboards, html pages, schedules. Etc.) in Top-level folders, and you can control access at that level.

Synonyms, on the other hand, are stored in **APPLICATION** folders. When you first create a Top-level folder, an application folder of the same name is also created. In the above screen shot, note that the application folder is “Db2 Web Query Training” because we right clicked on that Top-level folder.

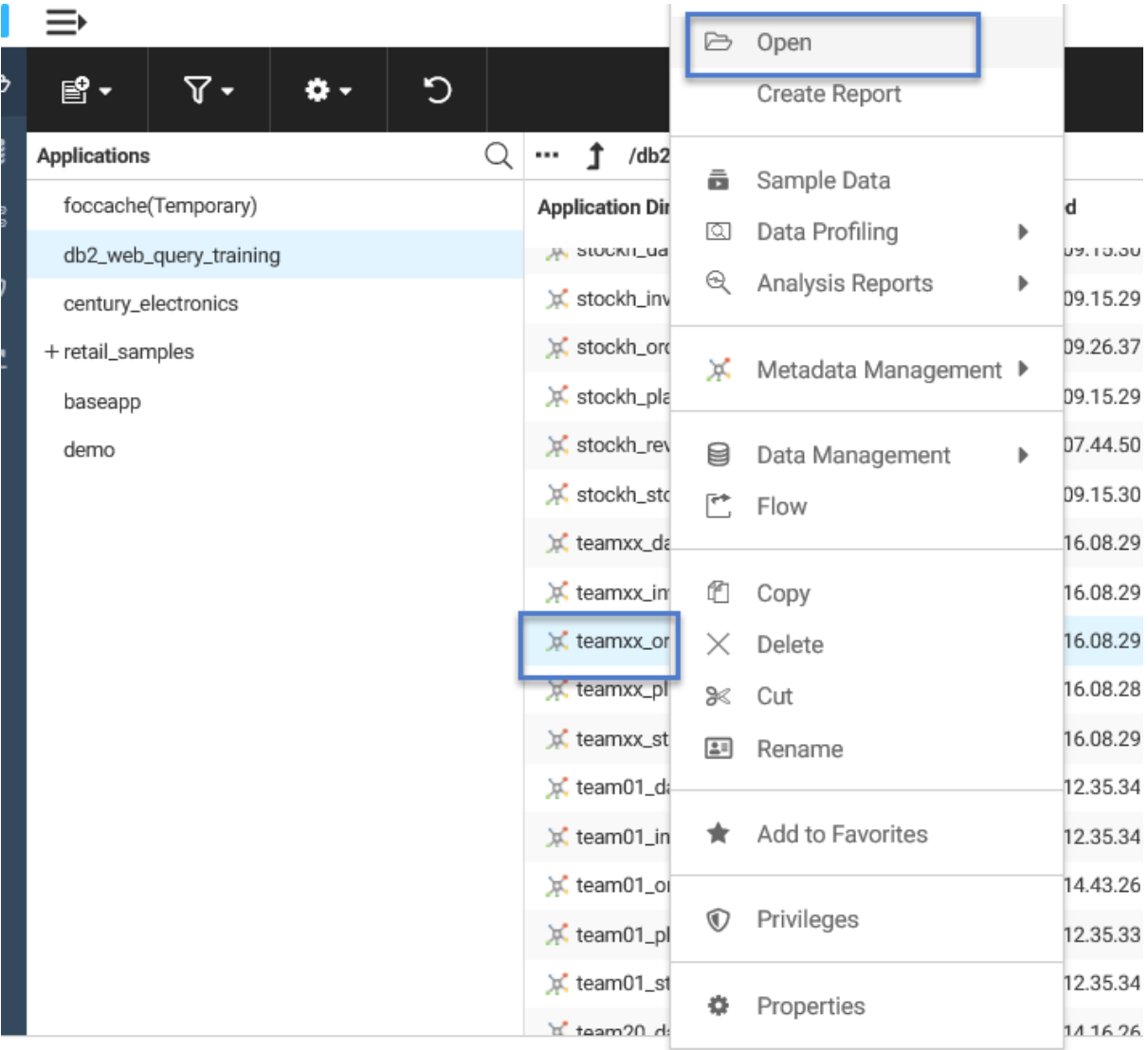
There may be times where you want to consider creating synonyms in a standalone application folder. For instance, perhaps many reports in many different top-level folders need to access a customer master table. Instead of creating many synonyms for the customer master table, you might want to think about creating a standalone application folder and putting more globally used synonyms there, and then just linking that application folder to whichever top-level folders need that for their reports (you do that through the properties of the top-level folder).

Click on the folder icon to see the application folder view.



Double click on the **Db2 Web Query Training application folder**. You should see the synonyms you just created with prefix "Teamxx_."

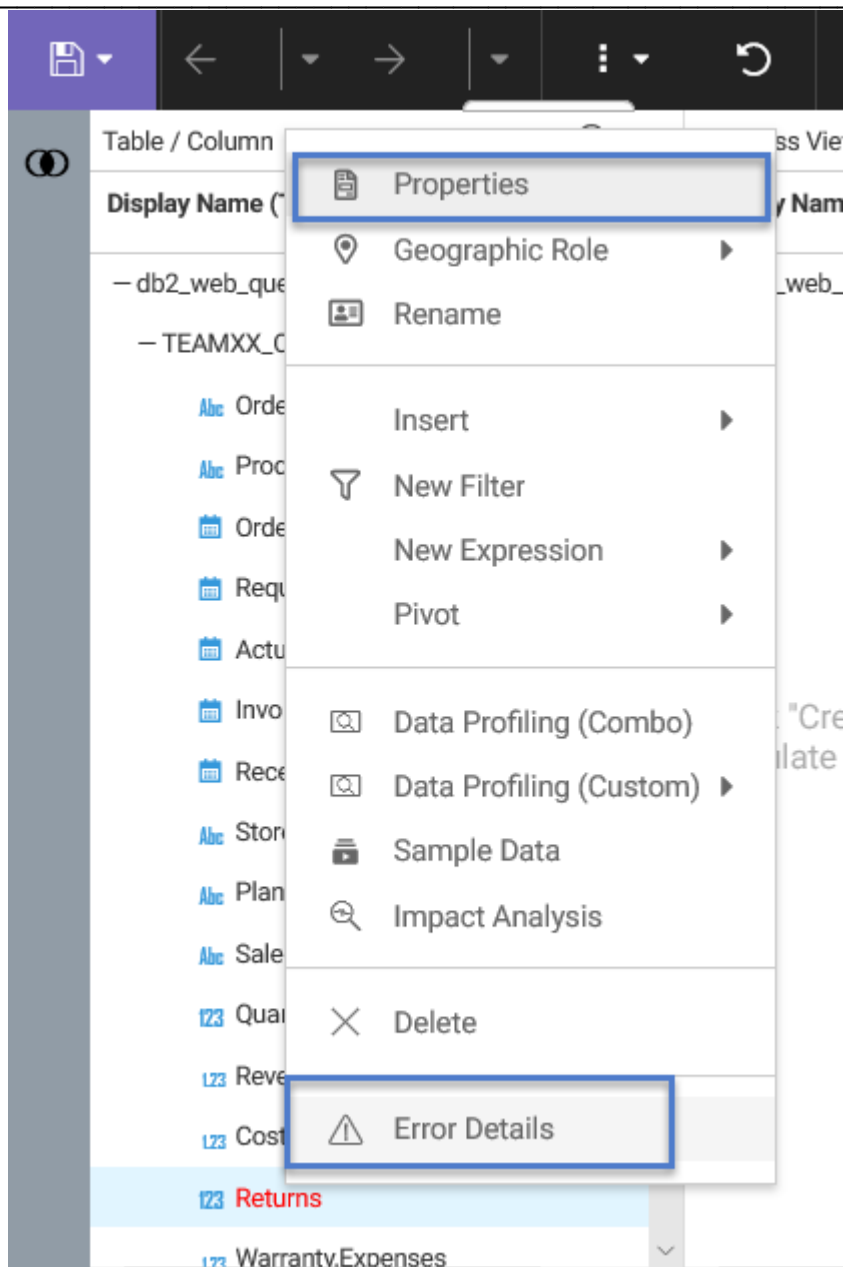
Right click on TEAMxx_ORDERS. You will select OPEN to open the synonym editor, but before you do, take a look at some of the other options on that pop-up menu:



Now click on OPEN if you haven't yet.

Display Name (Title)	Expr	Display Name (Title)	Expr	Order Number	Product Number	Order Date
db2_web_query_training/teamxx_orders		db2_web_query_training/teamxx_orders		28003	2005	2020/
TEAMXX_ORDERS				28003	3004	2020/
Order,Number				28003	4022	2020/
Product,Number				28003	5002	2020/
Order,Date				28003	5004	2020/
Requested,Ship Date				28004	1004	2020/
Actual,Ship Date				28004	1014	2020/
Invoice,Date				28004	1031	2020/
Receive,Date				28004	2001	2020/
Store,Code				28004	2004	2020/
Plant,Code				28004	3001	2020/
Sales Rep				28005	1005	2020/
Quantity				28005	1042	2020/
Revenue				28005	2001	2020/
Cost of,Goods Sold				28005	3004	2020/
Returns				28005	4014	2020/
Warranty,Expenses						

Here you will see the columns/fields in the ORDERS table. Note RETURNS has some red associated with it, implying there is some kind of error. Returns happens to be a reserved word so we want to change that. **Right click on Returns** and you can see at the bottom of the pop-up menu the error details. Feel free to click on that to have a look, but then **click on PROPERTIES**.



Change the Fieldname to "Returns_Quantity". It might seem a little confusing – you are not really changing the underlying field name in the database. Again, metadata is just a representation of the underlying data source. But you might want to note on the properties pop up window some of the other elements there. It is here where you can dictate default formatting of the field so you don't have to do that in every single report. Things like a "\$" sign for monetary values, or how to handle negative numbers, or if you want commas separating the 1000's markers in numeric fields, etc. Enhancing the metadata over time to add formatting is a nice feature.

Scroll down in that window and **select APPLY**.

Properties

General

? FIELDNAME

Return_Quantity

×

? TITLE

Returns

? Usage Format

I11

? Type

Integer

? Length

11

? Options

? Negative

Default - Minus sign at the left: -6148

? Comma

Default - Suppress: 41376

? L - Leading zeroes

☐

? S - Print blank for zero

☐

? Abbreviated display

Default: 1234567890123

? Percent sign

Default: 0.75

The name used to reference this element of data in a re...

An alternative report column title for the field

Describes how to format a field when displaying it in a r...

Is the data type

Is a length specification

Display options

Negative numbers display options

Comma display options

Adds leading zeroes

If the data value is zero, prints a blank in its place

Displays 12345 as 12.3K and 12345678901 as 12.3B.

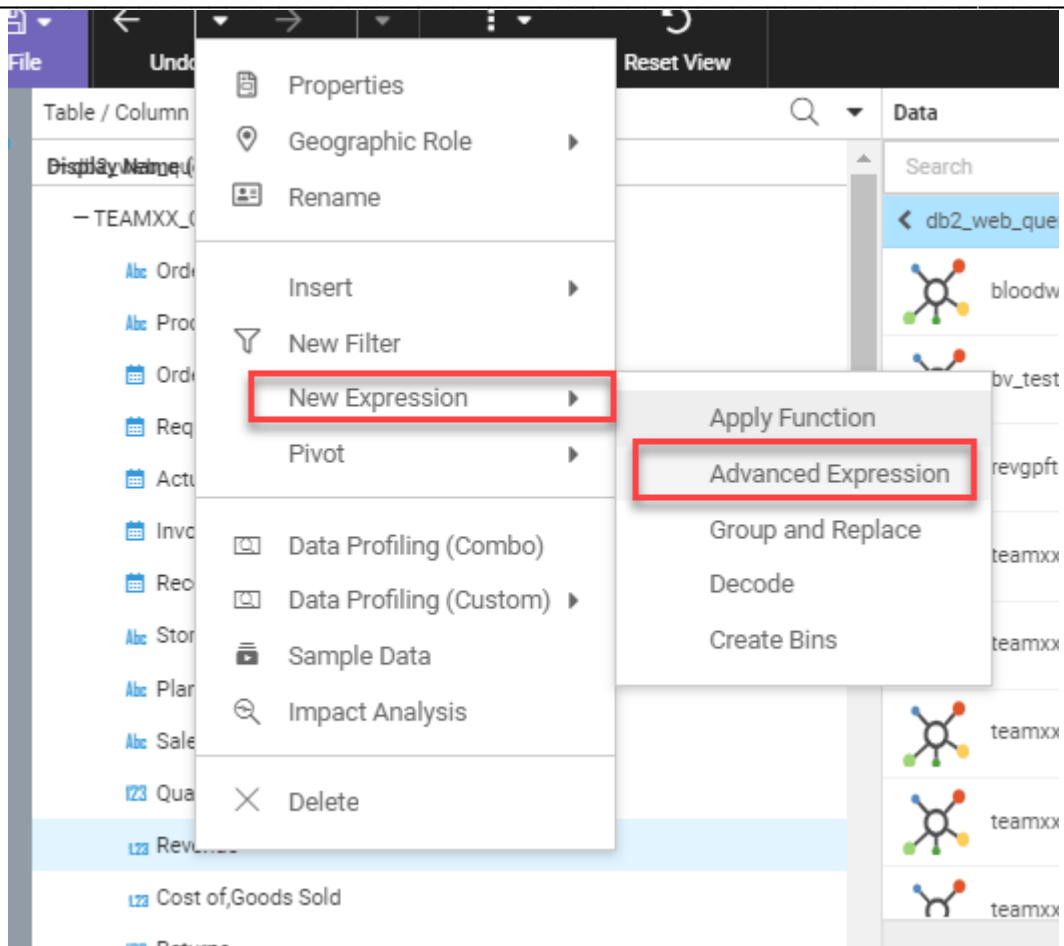
Displays a percent sign along with numeric data

Let's add a "virtual" or derived field. The beauty of synonyms are that they are a representation of the data source but can also contain derived fields. If you use a field like "commission pay rate" in 100 different reports, you only need to define the rule about this in ONE PLACE – in the synonym. Should that pay rate change, you can change it in the synonym and all of those 100 reports do not have to be touched. This is building integrity, or "one version of the truth" around calculated fields.

In Db2 Web Query there are two types of virtual/derived fields. **DEFINE** fields get calculated every time a row is read in the report execution. **COMPUTE** fields read ALL rows and then do the calculation. The difference might seem subtle but can be an important distinction in both performance and getting the right value you want. You really can't calculate a "as a percentage of revenue" type calculation unless you've read through all the records and calculated total revenues!

Let's create a Gross_Profit DEFINE field. This will be a simple calculation of revenue – cost of goods sold.

Right click on the REVENUE field **select New Expression then Advanced Expression**.



In the Expression Builder display you have an interface to apply rules to your new field. You can use any of the 100 or so BUILT IN functions, Db2 SQL Functions, as well as the keypad to apply arithmetic or if-then-else type logic.

Suppose your business defines a shipment status that you want to use in reports. If the SHIP_DATE is 2 days after PLANNED_SHIP_DATE then the new derived field contains "LATE" but if its 2 days earlier than PLANNED_SHIP_DATE it should be "Early", else "ON-TIME." Defining this one time in the synonym allows any report built over that synonym to use this field. If the business says "it should only be 1 day before or after PLANNED_SHIP_DATE" you change it in one place, not in each and every report.

Enter the field name as GROSS_PROFIT, the title (which will be your column heading in your report) as Gross,Profit (the comma puts Gross on one line and Profit on another line below it in a column heading). **Select the box with 3 dots** right of the FORMAT field.



Select DECIMAL in the drop-down box for TYPE and **make it a length of 18 with 2 decimal places**. Also, while here, let's add a currency symbol as a default formatting option. **Click on the currency box and select US Dollars and floating \$ sign**.

Properties

General		
? Usage Format	P18.2:C	Describes how to format a field when displaying it in a report
? Type	Decimal	Is the data type
? Length	18	Is a length specification
? Number of decimal places	2	Is the number of digits that follow the decimal point
? Options		Display options
? Negative	Default - Minus sign at the left: -6148	Negative numbers display options
? Comma	Default - Suppress: 41376	Comma display options
? L - Leading zeroes	☐	Adds leading zeroes
? S - Print blank for zero	☐	If the data value is zero, prints a blank in its place
? Abbreviated display	Default: 1234567890123	Displays 12345 as 12.3K and 12345678901 as 12.3B.
? Percent sign	Default: 0.75	Displays a percent sign along with numeric data
? Currency	☒	Extended Currency Support
? Currency Code	USD - USA dollar	Describes currency ISO code of the field
? Currency Symbol Position	LEFT_FLOAT: \$123.45	Describes currency display options of the field

Your screen should look like the above. **Click on OK.**

Advanced Expression

Name Title Format

☐ post-aggregation

Name

- + Character
- + Numeric
- + Date/Date-Time
- + Format Conversion
- + Geography
- + Data Source and Decoding
- + System
- + Miscellaneous
- + DBMS pass-through

Note that the left-hand panel defaults to a list of built-in functions. Feel free to expand some of the folders to look at the various built-in functions. For more information on functions, refer to the product manual on the Db2 Web Query wiki (<http://ibm.biz/db2wqwiki>) DOCUMENTATION link.

Let's build our expression for Gross_Profit. Gross_Profit is a simple arithmetic calculation using existing fields in the data source. **Click on the 3 box icon** to see the fields in the synonym.

Advanced Expression

Name Title

☐ post-aggregation

Name

- + Character

If not already expanded, open up the Teamxx_Orders section on the left hand side to see all the measurement fields. **Double click on Revenue (or you might see it as LINETOTAL),** select the minus "-" sign on the key-pad, then **double click on Cost,of Goods,Sold.**

Advanced Expression

Name: Title: Format:

☐ post-aggregation &

Name

Order,Number
Requested,Ship Date
Actual,Ship Date
Invoice,Date
Receive,Date
Sales Rep
Quantity
Revenue
Cost of,Goods Sold
Returns
Warranty,Expenses

Revenue - "Cost of,Goods Sold"

IF THEN ELSE EQ NE GT GE

⏏ ↺ ↻ 🗑️ ⬆️ ⬆️

To check out your expression, **click on the Validate button**. If satisfied, **click on OK**. Note the new field in your list of fields within the teamxx_orders synonym:

Table / Column	Expr	Displ
Order,Date		db
Requested,Ship Date		
Actual,Ship Date		
Invoice,Date		
Receive,Date		
Store,Code		
Plant,Code		
Sales Rep		
Quantity		
Revenue		
Cost of,Goods Sold		
Returns		
Warranty,Expenses		
Shipping,Cost		
Gross,Profit		Re
+ Indexes		

While we're here, let's create a couple of date attributes by "decomposing" an existing date field. Later we'll join to a date conversion (or date dimension) table, but it's also nice to very quickly create some date attributes from an existing "true date" date field (if your dates are not true DATE data type this won't work unless you create a DEFINE field that converts that to a true date field). You can refer to the Nuts and Bolts redbook (<http://ibm.biz/db2wq-nutsandbolts>) for date conversion function examples.

In this case we can use Order,Date field. **Scroll up to Order,Date, right click and select New Expression and Decompose Date.**

The screenshot shows the IBM Data Studio interface. At the top, there's a toolbar with icons for saving, running, and navigating. Below the toolbar, a table structure is visible with columns for 'Display Name (Title)' and 'Expression'. The table is named 'db2_web_query_training/teamxx_orders'. A right-click context menu is open over the 'Order,Date' field. The menu options include 'Properties', 'Geographic Role', 'Rename', 'Insert', 'New Filter', 'New Expression', 'Pivot', 'Data Profiling (Combo)', 'Data Profiling (Custom)', 'Sample Data', 'Impact Analysis', and 'Delete'. The 'New Expression' option is highlighted with a blue box. A secondary menu is open from 'New Expression', showing options like 'Apply Function', 'Advanced Expression', 'Group and Replace', 'Decode', 'Create Bins', and 'Decompose Date...'. The 'Decompose Date...' option is also highlighted with a blue box. A blue arrow points to the 'Order,Date' field in the table structure. A text overlay on the right says 'Click "Create Default" icon to populate th'.

You'll be presented with a dialog box to choose up to 8 date attributes (such as YEAR, Month, or combinations like Year-Month). The bottom 4 are selected by default. Go ahead and **check the top 4 and click on OK**.

NOTE: It's possible that the dates were auto decomposed for you already and you won't see the Decompose Date option. This is a setting in the synonym editor. You can check by just scrolling down in the field panel and see if you see fields like Order,Date,Year and Order,Date,Month.

Decompose Date Field - Order,Date

×

Leave a check next to the fields you want added to the synonym file.

Fields to add

Q

☐	Name	Format	Title
☒	<i>f_x</i> ORDERDATE_YEAR	YY	Order,Date,Year
☒	<i>f_x</i> ORDERDATE_QUARTER	Q	Order,Date,Quarter
☒	<i>f_x</i> ORDERDATE_MONTH	M	Order,Date,Month
☒	<i>f_x</i> ORDERDATE_DAY	D	Order,Date,Day
☐	<i>f_x</i> ORDERDATE_YEAR_MONTH	YYMM	Order,Date,Year,Month

OK

Cancel

You'll now have 8 new derived fields you can use in reports that represent pulled out date elements from a date field (true dates only)!

Warranty,Expenses

Shipping,Cost

Gross,Profit

Order,Date,Year

Order,Date,Quarter

Order,Date,Month

Order,Date,Day

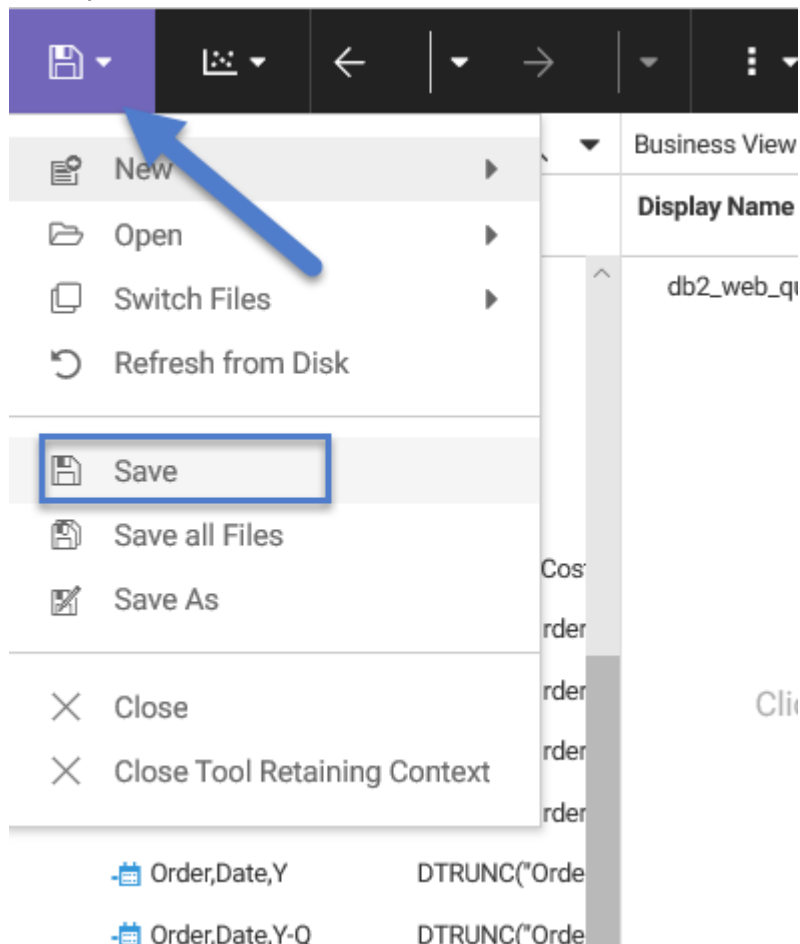
Order,Date,Y

Order,Date,Y-Q

Order,Date,Y-M

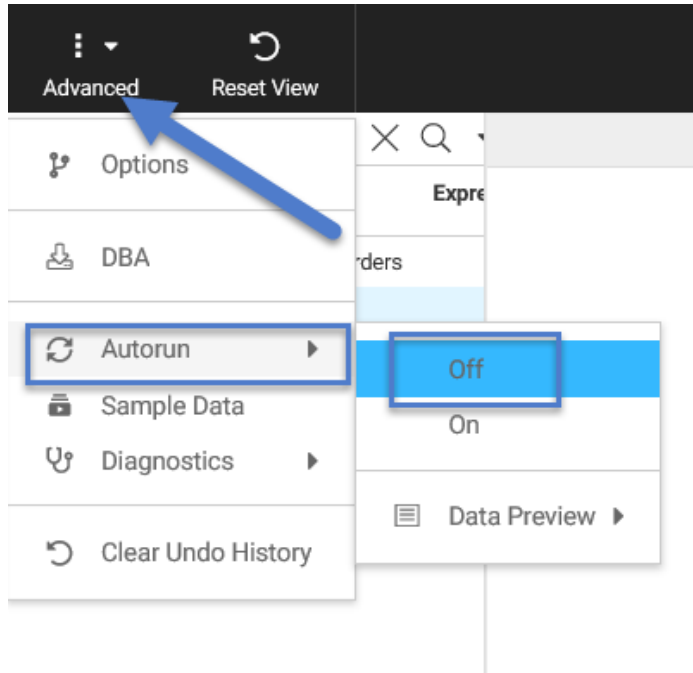
Order,Date,Y-M-D

Save your work!



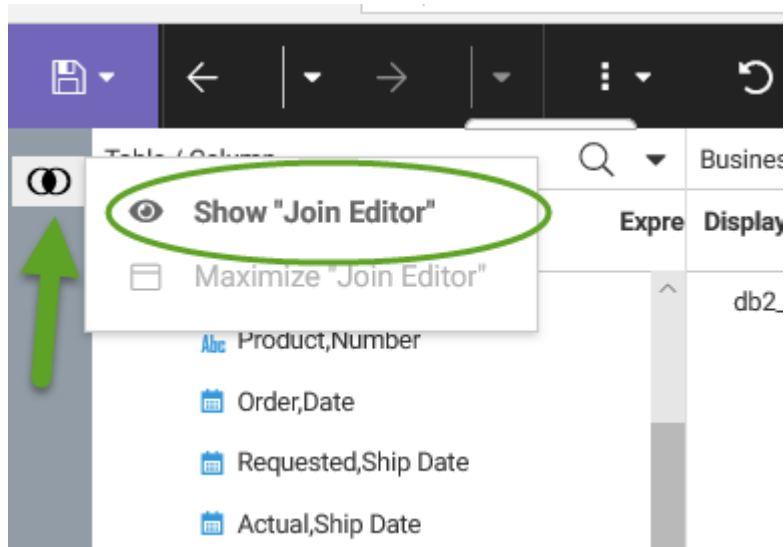
1.1.3 Joining Data

Before we create some joins, it might be a good idea to speed up this process by turning off the auto "sample data" that occurs with every change to the synonym. To turn off auto-run, **click on the tool in the ribbon bar with the 3 vertical dots, and select AUTO-RUN and OFF.**

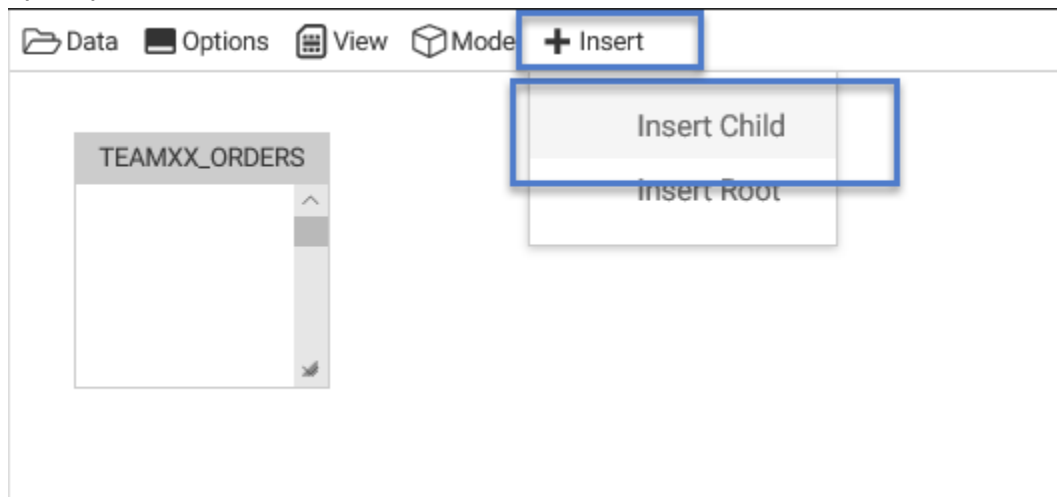


Now onto the joining of files!

Click on the Join Editor tool in the ribbon at the top.



What we want to do is add in the Inventory, Stores, PLANT and DATE_CONV tables to this synonym. **Click on +INSERT, then Insert Child:**



Make sure that you select the Db2 Web Query Training folder and then **select your Teamxx_inventory, plant, and stores tables.**

What you are doing is adding these synonyms into teamxx_orders synonym. If these files are often used together in reports, now a report author can just choose to build their report using this rather than have to define joins in their report.

Note that Db2 Web query found what it assumes are the primary and foreign key fields (for example, "Product,Number" = "Product,Number").

Click on OK

Insert Child

? Filtering: ☒ Foreign -> Primary ☒ Matching names ☐ Primary -> Foreign ☐ Fuzzy matching names ☐ Other (or Unmatche

— Applications

- foccache - Session Directory: Fil...
- db2_web_query_training
- century_electronics
- + retail_samples
- baseapp - Default Directory: File...
- demo

Parent/Child	Confidence	Status	Condition	Desc
stockn_orders	90%		"Store,Code" = "Store,Code"	Order
stockh_plant	100%		"Plant,Code" = "Plant,Code"	Plant
stockh_stores	100%		"Store,Code" = "Store,Code"	Store
teamxx_inventory	100%	☒	"Product,Number" = "Product,Number"	Inver
teamxx_plant	100%	☒	"Plant,Code" = "Plant,Code"	Plant
teamxx_stores	100%	☒	"Store,Code" = "Store,Code"	Store
team01_inventory	100%		"Product,Number" = "Product,Number"	Inver
team01_orders	90%		"Order,Date" = "Order,Date"	Order
team01_orders	90%		"Product,Number" = "Product,Number"	Order

OK Cancel

We see that Db2 Web Query has added the 3 additional tables and automatically joined them, on the join predicates that it is proposing. In other words, it found common field names in the files that it chose to join on. You could override this, but it did a pretty nice job. You can click on any of the lines joining the tables and hover of the line to see join conditions, or you can right click and change join conditions, or even select the edit parent link to get an editor to make more extensive changes. For instance, it is not unusual to see a scenario of needing 2 or more join columns. Or situations where you want to join on different size fields. For example, you want to join the 4 first characters of an 8 character field with a 4 character field in the child table. The Edit Parent Link is where you can get creative about how your files are joined!

The screenshot shows a database tool interface with a join configuration. At the top, there's a toolbar with icons for Data, Options, View, Mode, Delete All Joins, and Insert. Below this, three tables are shown: TEAMXX_ORDERS, TEAMXX_STORES, and TEAMXX_PLANT. TEAMXX_ORDERS is at the top, with fields Order, Numl, Product, Nu, and Order, Date. TEAMXX_STORES is on the left, with fields Store, Code, Store, Name, and Country. TEAMXX_PLANT is on the right, with fields Plant, Code, Plant Name, and Plant Coun. Arrows indicate joins between these tables. A context menu is open over the join, showing options: Edit Parent Link, Join Multiple, Join Unique (highlighted), Inner Join (highlighted), Left Outer Join, Right Outer Join, Full Outer Join, Delete Link, Sample Data with parent key, and Data Profiling (Custom). Below the tables, there's a section for data for db2_web_query_training/teamxx_orders, showing columns Sales Rep, Quantity, and Revenue. A row is visible with values AJ Van de Kam, 319, and 41151.00.

But what happened to our Date_Conv file? Because there was no obvious matching fields, the synonym representing that table wasn't even presented to us as an option to bring into the teamxx_orders synonym.

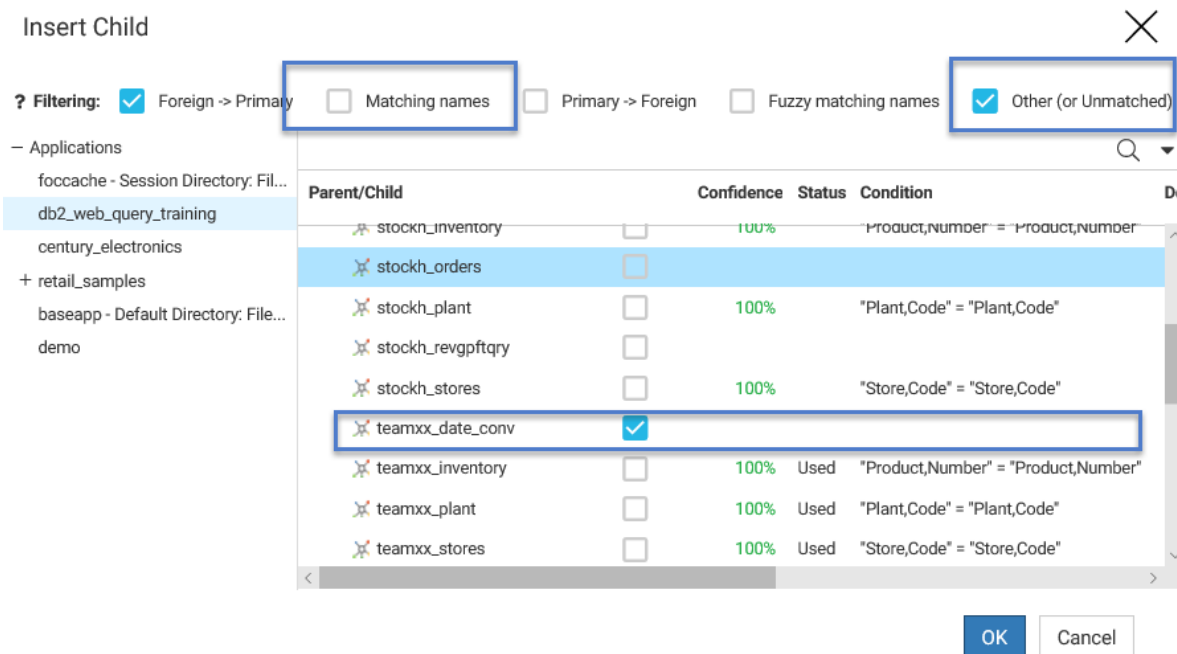
There is quite a bit of detail about date processing and how to leverage a table like this for easily dealing with date fields in your database that are not "true" dates. Storing dates in Packed 8 Decimal, Character, or even 7 (or 6) character formats without a century field type legacy dates? Probably – we've seen many of these "legacy" dates and the data dimension table contains a field to join on for just about everything we've experienced!

To read more about date processing, refer to the chapter in the Nuts and Bolts Redbook at <http://ibm.biz/db2wq-nutsandbolts>.

Click on the + INSERT and INSERT CHILD options one more time.

In this case, there is no easily identifiable (matching) primary-foreign key. Date_Conv didn't even show up in the above list of synonyms to add to teamxx_orders. What we need to do is **DESELECT the matching names, SELECT the Other (or Unmatched)** check box, and then we'll see our teamxx_Date_Conv synonym

Check the teamxx_date_conv synonym and click OK.



Db2 Web Query didn't find a common field name (with similar data type) to auto link the two so it opens the join "expression" editor. **Click on the open space** box under teamxx_orders which will pop up the list of fields. **Select ORDER,DATE.**

Edit Join from TEAMXX_ORDERS to TEAMXX_DATE_CONV

Configure Join

Inner

Left Outer

Right Outer

Full Outer

TEAMXX_ORDERS

TEAMXX_DATE_CONV

Click 'Edit' to select fields

Order,Number

Product,Number

Order,Date

Requested,Ship...

Actual,Ship Date

Invoice,Date

Receive,Date

Store,Code

Plant,Code

Session

Clear All

Suggestions

Sample Data

Join Profiler

Data with Parent Key for TEAMXX_DATE_CONV. Limit 50 rows. Double the Row Limit

TEAMXX_ORDERS Order Number	TEAMXX_ORDERS Product Number	TEAMXX_ORDERS Order Date	TEAMXX_ORDERS Requested Ship Date	TEAMXX_ORDERS Actual Ship Date
28003	2005	2020/10/17	2021/01/20	2020/11/21
28003	3004	2020/10/17	2021/01/18	2021/01/16
28003	4022	2020/10/17	2020/11/27	2020/12/14
28003	5002	2020/10/17	2020/11/17	2020/12/04
28003	5004	2020/10/17	2020/11/26	2020/12/01
28004	1004	2020/10/22	2021/01/26	2021/02/03
28004	1014	2020/10/22	2020/12/04	2020/11/26
28004	1031	2020/10/22	2020/11/30	2020/12/06
28004	2001	2020/10/22	2020/12/03	2020/12/04
28004	2004	2020/10/22	2020/12/04	2020/12/04

Now **click on the open space under teamxx_date_conv** and a list of fields from that synonym appear. **Select DC_DATE**. As you learn more about the date dimension table, you'll find out that in YOUR case you may be joining a Packed 8 decimal field containing a date to the field on the date dimension table of the same data type). Once you've done this, you have a TON of date attributes (Q/M/Y, day of week, week of year, etc.) available to you in this cluster (join) synonym.

WAY COOL!

Note also that the default of an inner join here, which is good in this circumstance. Also, if you click on the down arrow between the two fields we just selected, you can see "=" was the default but you can also choose other expressions to relate the two tables.

Page 25

© 2020 IBM Corporation

Edit Join from TEAMXX_ORDERS to TEAMXX_DATE_CONV

Configure Join
☐ Multiple
Sample Data
Join Profiler

Inner
Left Outer
Right Outer
Full Outer

TEAMXX_ORDERS
TEAMXX_DATE_CONV

Order,Date
=
Click 'Edit' to select fields
2020/10/17
28003

+
+ Expression

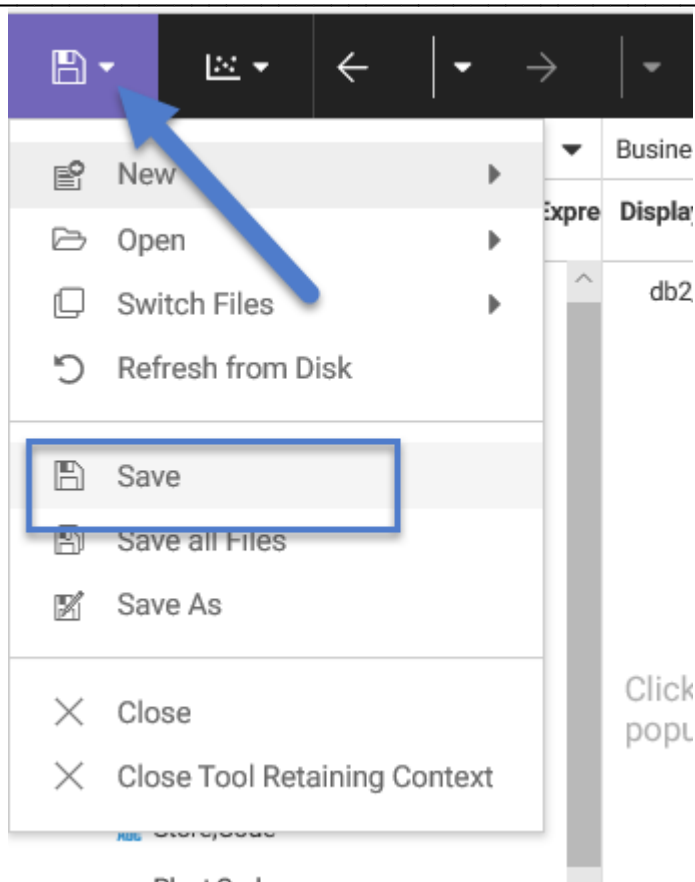
DC_DATE
DC_JDE_JULIAN_DATE
DC_MDYY_DEC
DC_MDYY_ZONED
DC_MDYY_CHAR
DC_YYMD_DEC
DC_YYMD_ZONED
DC_YYMD_CHAR
DC_MDY_DEC

TEAMXX_ORDERS	TEAMXX_DATE_CONV
Order Date	On Nun
2020/10/17	28003
17	28003
17	28003
17	28003
17	28003
22	28004
22	28004
22	28004
22	28004
22	28004
22	28004

If you **click on the EXPRESSION** button you open up the expression builder. Here is where you can do more sophisticated join conditions, using the keypad we've seen before or the 100+ different built in (or Db2) functions. For instance, what if you had a 4 character field you needed to join to the first 4 characters of an 8 character field. You could use the SUBSTR function and build that into your join definition.

Hit CANCEL for now to close your window. **Close the Join Editor** ("X" in upper corner).

This is a good time to save your work. Back on the main synonym editor screen, **select the FILE and SAVE options:**

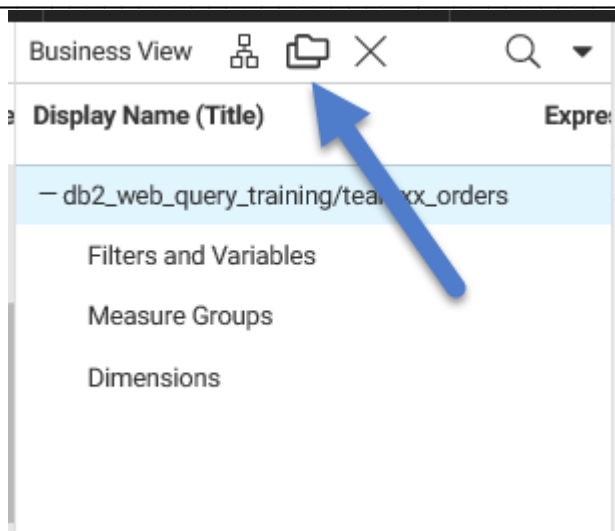


You've just built your first "cluster join" synonym, i.e., a single synonym that represents 5 underlying files/tables joined together. Report authors will not have to deal with defining their joins in each and every report that needs to use this set of data, nor will they need to know how the files/tables should be joined.

1.1.4 Defining Dimensions for Auto Drill Downs

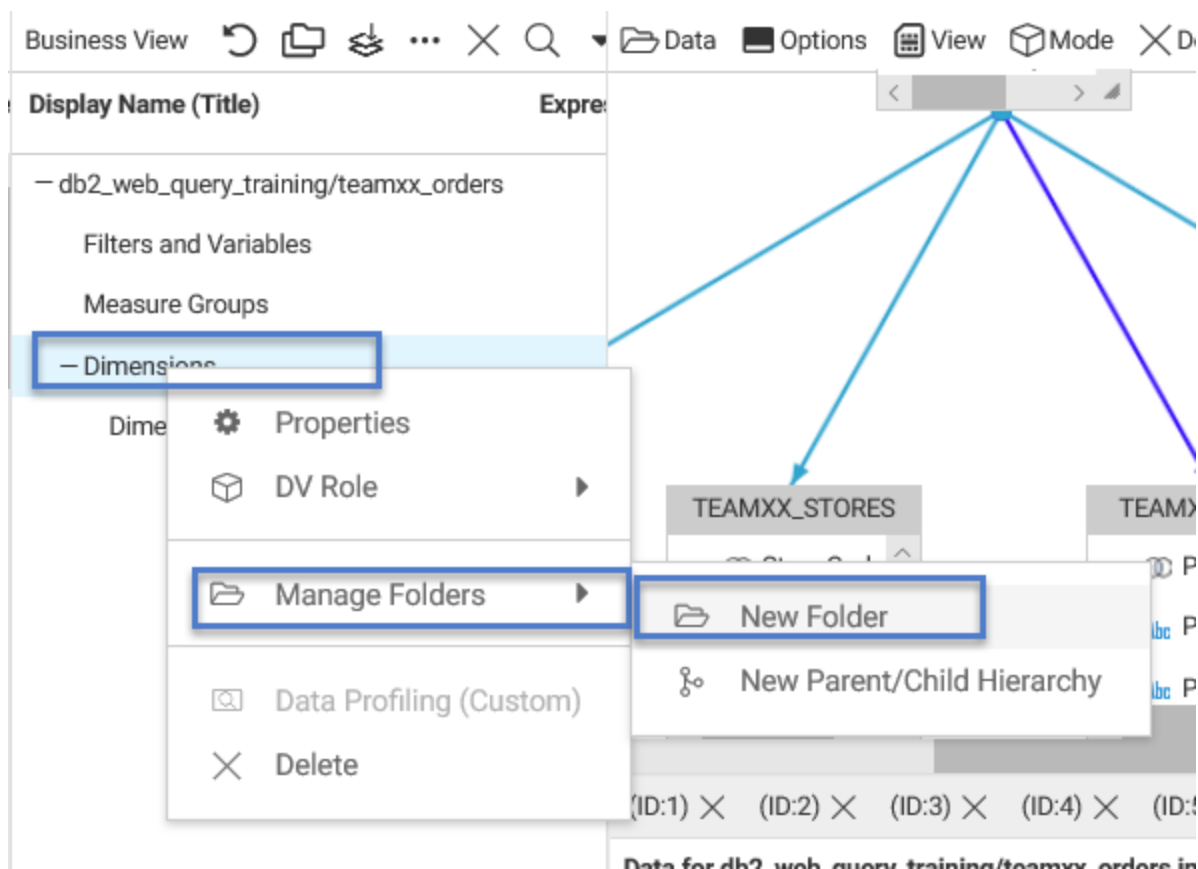
There is much more to metadata and functions within the synonym editor. You can also define hierarchical relationships within your data that enables certain types of reports. OLAP and auto-drill enabled reports use this relationship definition for drill downs. They can also be used in slicer reports or for chaining dynamic parameters (if a user select United States for the country parameter, the drop down box for REGIONS should only show regions within the United States).

For dimensional hierarchies we'll be using the Business Views panel in the synonym editor. **Click on folders icon** to have webquery create some folder structure to store your hierarchies.

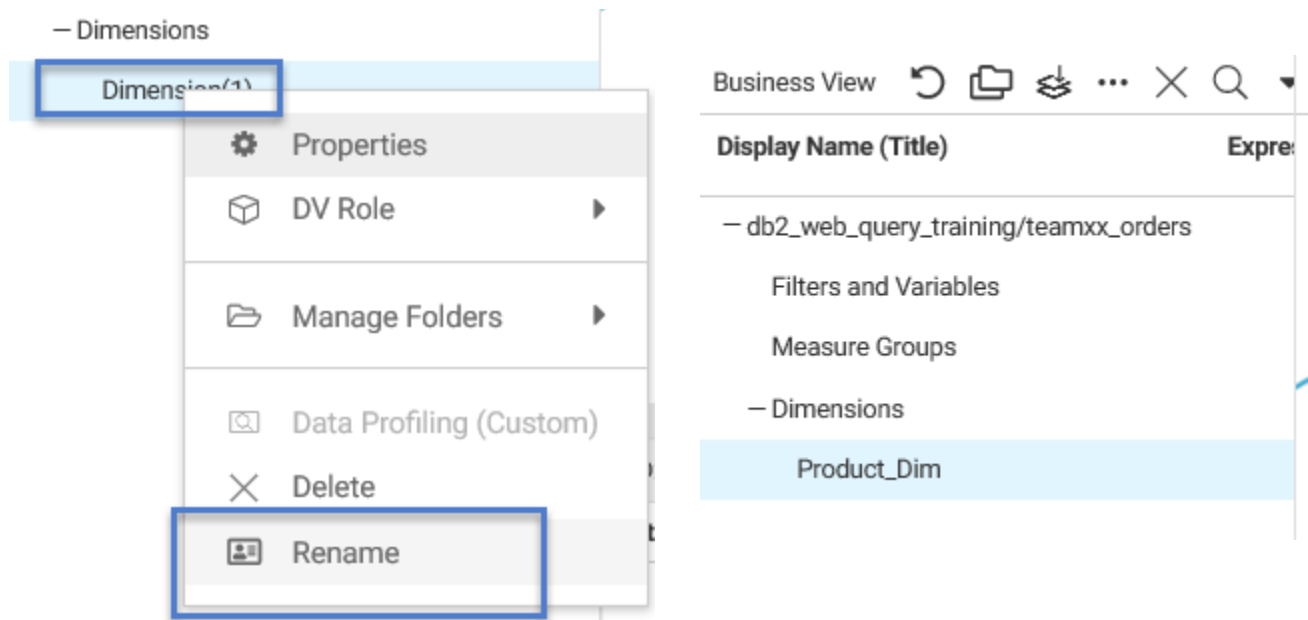


What we are going to do is set up four dimensional hierarchies.

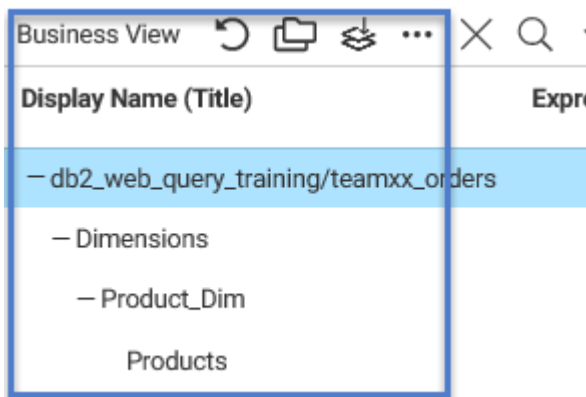
Right click on Dimensions and **select Manage Folders** and **New Folder**.



Let's rename our new folder. **Right click on Dimensions(1)** and **select Rename**. Give it the name Product_Dim.



We need to create a subfolder to contain the hierarchy underneath Product_Dim. **Right click on Product_Dim, select Manage Folders, and New Folder. Right click on Heirarchy(1) and rename it "Products."** Your folders should look like this:



In the field/column list panel on the left hand side, navigate to the Inventory table. Expand it and **use CTRL-click to select Product Type, Product Category, and Model.** Once selected, **drag them to the business view panel under Products.**

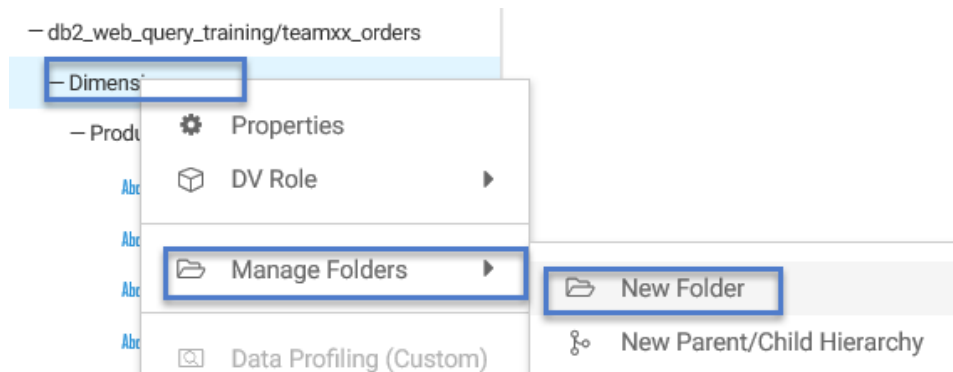
Table / Column	Expression	Business View
Display Name (Title)	Expression	Display Name (Title)
Receive,Date,Y-Q	DTRUNC("Re	– db2_web_query_training/teamxx_orders
Receive,Date,Y-M	DTRUNC("Re	– Dimensions
Receive,Date,Y-M-D	DTRUNC("Re	– Product_Dim
Gross,Profit	Revenue - "C	– Products
– TEAMXX_INVENTORY		Abc Product Type
Abc Product,Number		Abc Product Category
Abc Product Type		Abc Model
Abc Product Category		
Abc Product Name		
Abc Model		
Quantity,In Stock		

Now **select Product Name and drag to be underneath Model** in the Business View. Your PRODUCTS hierarchy should look like this:

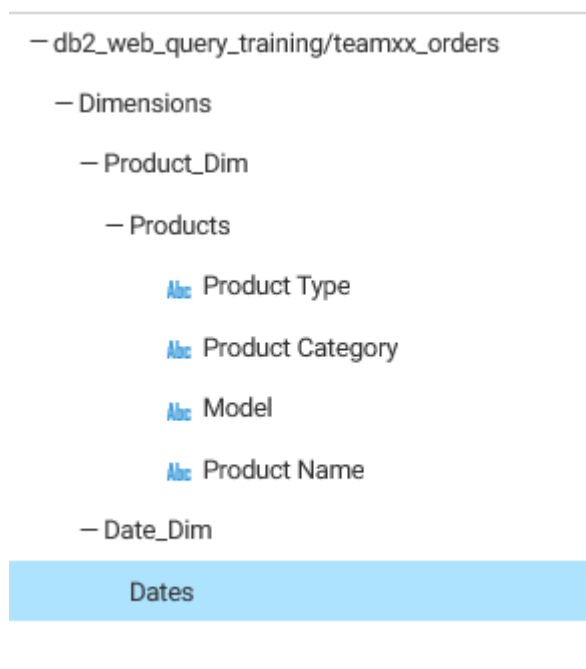
Business View
Display Name (Title)
– db2_web_query_training/teamxx_orders
– Dimensions
– Product_Dim
– Products
Abc Product Type
Abc Product Category
Abc Model
Abc Product Name

What we're doing is ordering this such that if you're looking at Product Type, an auto drill will report will allow the drill down into Product Category, and then Model, and finally Product Name. Reports like OLAP reports or "auto-drill" reports leverage this!

Add a 2nd hierarchy. Go back to the **DIMENSIONS level and right click, select Manage Folders and New folder. Rename the new folder to Date_Dim.**



Now **right click** on Date_Dim and select manage folder and new folder again. Rename the new folder to Dates.

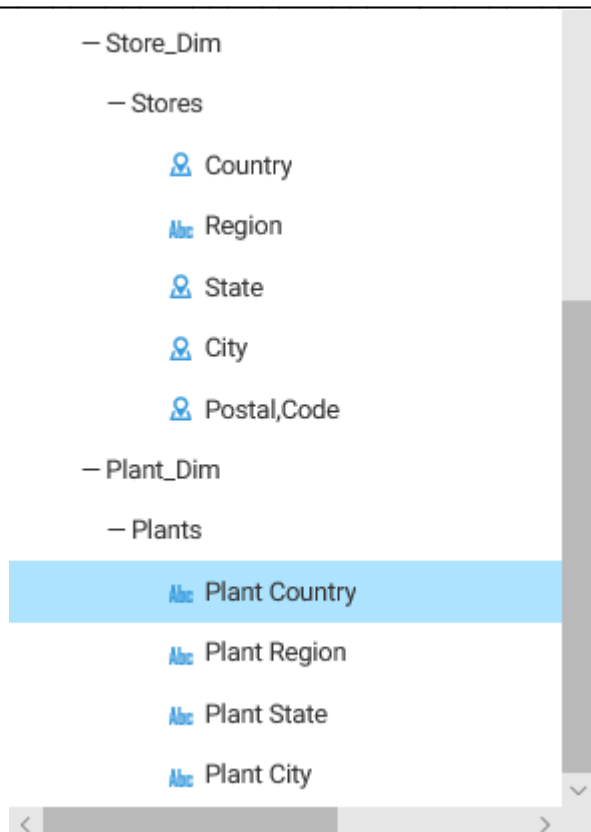


Remember that Order,Date that we decomposed? Navigate to the ORDERS table within the synonym's list of fields, find the first four decomposed values of Order,Date. **Select all four using CTRL-click and move to the right underneath your new Date_Dim.**

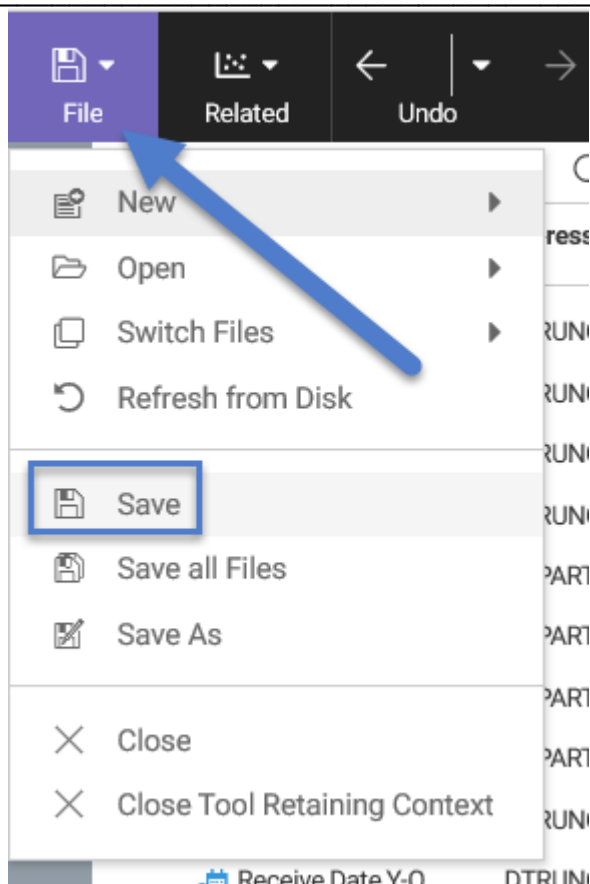
Table / Column ...		Business View	
Display Name (Title)	Express	Display Name (Title)	Express
Revenue		db2_web_query_training/teamxx_orders	
Cost of,Goods Sold		Dimensions	
Returns		Product_Dim	
Warranty,Expenses		Products	
Shipping,Cost		Product Type	
Order,Date,Year	DTP/	Product Category	
Order,Date,Quarter	DTP/	Model	
Order,Date,Month	DTP/	Product Name	
Order,Date,Day	DT/	Date_Dim	
Order,Date,Y	DTRI	Dates	
Order,Date,Y-Q	DTRI		
Order,Date,Y-M	DTRI		
Order,Date,Y-M-D	DTRI		

Now see if you can do the same thing for a STORES hierarchy (from the STORES table) and a PLANT hierarchy (from the PLANT table). Refer to the below screen shot. Remember to right click on DIMENSION level to create a new folder, then rename it with the _Dim suffix. Then right click on the new dimension folder and create another new folder to contain the hierarchy. Then use drag and drop to place under the appropriate dimension/hierarchy.

Your Store and Plants hierarchy should look like this:



Remember to save your synonym:



Exit from the Synonym Editor.

In this chapter the BI Portal metadata interface was shown. There is also a windows client called Developer Workbench. This is a separate PC installation process. Similar functions exist there, although the interface is slightly different. To see the process of creating and editing a synonym using Developer Workbench, refer to chapter 4 of the Nuts and Bolts Redbook (<http://ibm.biz/db2wq-nutsandbolts>).

CONGRATULATIONS! You've just created your first synonym.

2 Additional Information: Links

Links:

[Ibm.biz/db2webqueryi](http://ibm.biz/db2webqueryi) - main website

[Ibm.biz/db2wqwiki](http://ibm.biz/db2wqwiki) - technical wiki

Db2webqueryi.blogspot.com - Db2 Web Query Blog

Ibm.biz/db2wqezinstall - EZ-install getting started page

Contact the Db2 Web Query Team at QU2@us.ibm.com