## Title: Db2 Web Query for i Education Series: Module 6 – Using EZ-Report

**Assumptions:** You have installed Db2 Web Query using our "EZ-Install" package

Pre-Reqs: Modules 1-5

## Table of Contents

EZ-Install Package

This hands-on lab is based on having installed the latest version of Db2 Web Query with the "EZ-Install" package. This package includes all the Db2 Web Query software and pre-built sample reports and utilities, including:

- Db2 Web Query Auditing reports – for example, see who is running the longest or most reports
- Sample reports for the Systems Administrator – for example, track near real time utilization across multiple LPARs/Servers.
- Ready to run Tutorials – learn how to build reports with the InfoAssist authoring tool.
- Discover information about your current Query/400 environment with the included Query/400 Discovery tool and sample reports built over the output files that the tool creates.

For information on EZ-Install and how to request it, check out this blog entry at Ibm.biz/db2wqezinstall-info. Or you can always send an email to the IBM Db2 Web Query team at QU2@us.ibm.com

# 1 EZ-Report

## 1.1 Use cases for EZ-Report

Db2 Web Query has always had the ability to auto create a synonym and a report over Query/400 definitions. With EZ-Report we've added a new utility in the IBM Db2 Web Query Information Samples folder to easily create a synonym and report over an SQL Statement or a file. This makes it very easy to generate a report and synonym over:

- A business-oriented SQL Statement
- An IBM i Service using Access Client Solutions "insert from examples"
- A file/table

The underlying technology can also be useful for automating the conversion from other SQL based reporting or business intelligence tools. If you have more interest in conversion utilities reach out to us at QU2@us.ibm.com.

### 1.1.1 Access Client Solutions

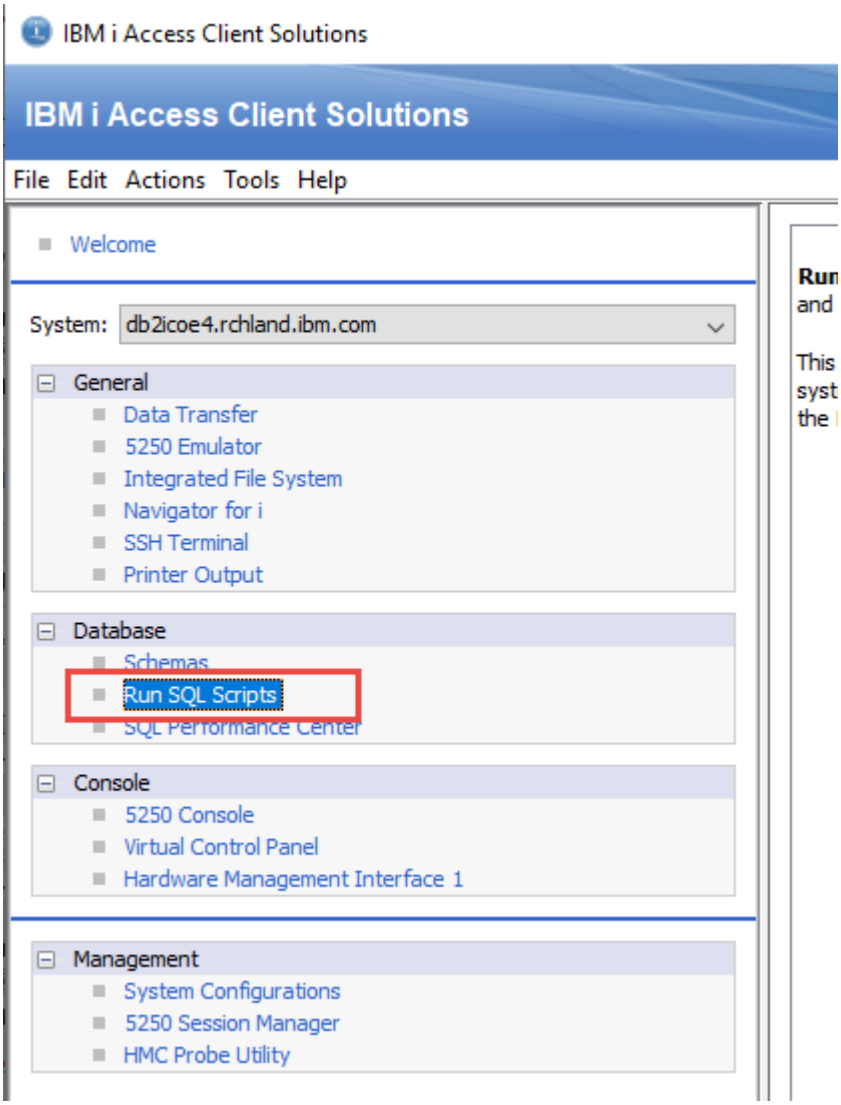
If developing query logic using SQL it would be advisable to do this through Access Client Solutions (ACS). You can download the no charge ACS tool by clicking here. ACS has many functions as you'll see, but we're mostly interested in the Run SQL Scripts component. This component allows you to use sample SQL as a starting point or write free form SQL and test it before entering it into EZ-Report. It is beyond the scope of this tutorial to cover all the cool stuff you can do in ACS but note that you could save your SQL in a text file on your PC or in the

IFS, or wrapper your SQL statement with a CREATE OR REPLACE VIEW that saves your SQL in a Db2 View object.

Previous to EZ-Report (and still available now), you could build a synonym over the uploaded SQL (into Db2 Web Query) text file, or over a Db2 view, and then build reports over that synonym. As you'll see, EZ-report is an easier way to auto generate the synonym and the report, eliminating a number of steps along the way. However, note that for long term use of SQL, you should consider how you are going to save and manage your SQL scripts. Saving them in text files or a view, or even a source physical file (all supported in ACS SAVE-AS options) is a good best practice for code management!

You can get to Run SQL Scripts from the main ACS pop up menu.




Suppose you would like a report like below, where you have a derived column for RUNNING TOTALS. To do this for ALL product types, that would be a very simple addition of a COMPUTE field with an express like "Running_Total = Running_Total + Revenue." However, that continues the running total across all product types (all rows in the data). Instead, what we would like to do is RESET the running total such at the running total calculation is contained within each Product Type. This is a more challenging aspect.

| PRODUCTTYPE | PRODUCTCATEGORY | REVENUE | RUNNING_TOTAL |
|---|---|---|---|
| Audio | Receivers | 35907113.00 | 35907113.00 |
| | Amplifiers/PreAmps/Tuners | 42374428.00 | 78281541.00 |
| | MP3 | 43491588.00 | 121773129.00 |
| | CD Players and Recorders | 53847459.00 | 175620588.00 |
| | Speakers | 84717053.00 | 260337641.00 |
| | Audio Systems | 122345680.00 | 382683321.00 |
| **Subtotal: Audio** | | **382683321.00** | |
| Camcorders | Digital8 Camcorders | 13614953.00 | 13614953.00 |
| | MiniDV Camcorders | 51539451.00 | 65154404.00 |
| | DVD Camcorders | 379376637.00 | 444531041.00 |
| **Subtotal: Camcorders** | | **444531041.00** | |
| Cameras | Digital Cameras | 184103667.00 | 184103667.00 |
| **Subtotal: Cameras** | | **184103667.00** | |
| Office | Organizers | 11712495.00 | 11712495.00 |
| | Handheld and PDA | 18533190.00 | 30245685.00 |
| **Subtotal: Office** | | **30245685.00** | |
| Video | VCR | 21688621.00 | 21688621.00 |
| | TV | 168799539.00 | 190488160.00 |
| | DVD | 329872045.00 | 520360205.00 |
| **Subtotal: Video** | | **520360205.00** | |

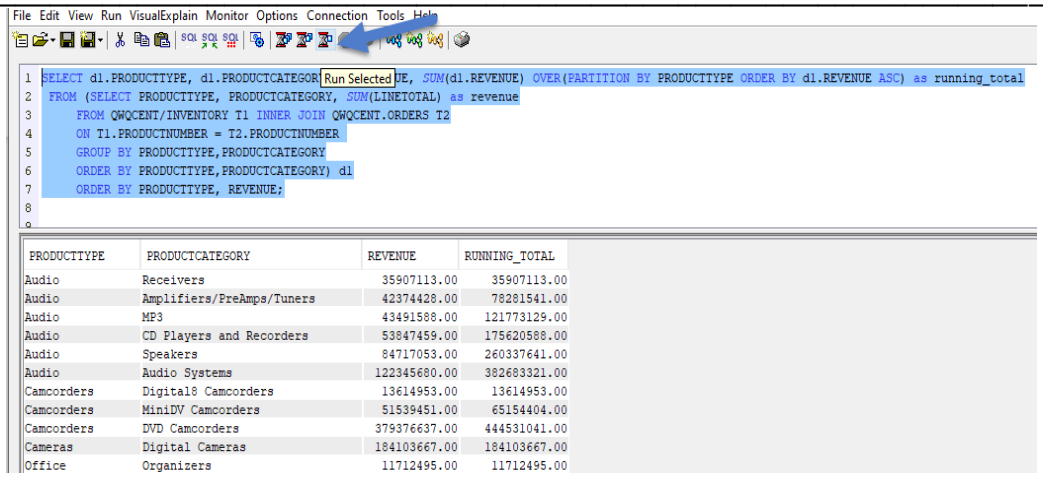
With a little bit of SQL magic, combined with EZ-Report, this can be done very quickly. In our case, we'll use some SQL language that only exists in 7.3 and above so this specific SQL will not run on 7.2 or below.

Enter this SQL statement in your Run SQL Scripts window:

```
SELECT d1.PRODUCTTYPE, d1.PRODUCTCATEGORY, d1.REVENUE, SUM(d1.REVENUE)
OVER(PARTITION BY PRODUCTTYPE ORDER BY d1.REVENUE ASC) as running_total
 FROM (SELECT PRODUCTTYPE, PRODUCTCATEGORY, SUM(LINETOTAL) as revenue
    FROM QWQCENT/INVENTORY T1 INNER JOIN QWQCENT.ORDERS T2
    ON T1.PRODUCTNUMBER = T2.PRODUCTNUMBER
    GROUP BY PRODUCTTYPE,PRODUCTCATEGORY
    ORDER BY PRODUCTTYPE,PRODUCTCATEGORY) d1
    ORDER BY PRODUCTTYPE, REVENUE;
```

If you're not familiar with SQL, this particular statement is taking advantage of a "sub-select" in blue to gather the fields of interest. The wrapper SELECT statement utilizes the new OLAP SQL language extensions that came out in 7.3, in particular the section in green. That "OVER(PARTITION…" clause allows you to calculate running total but WITHIN each Product Type. That's the MAGIC!

Place your cursor somewhere within the SQL Statement and **click on the RUN** button to test your SQL. Verify that you're getting running total within Product Type vs. all product types.

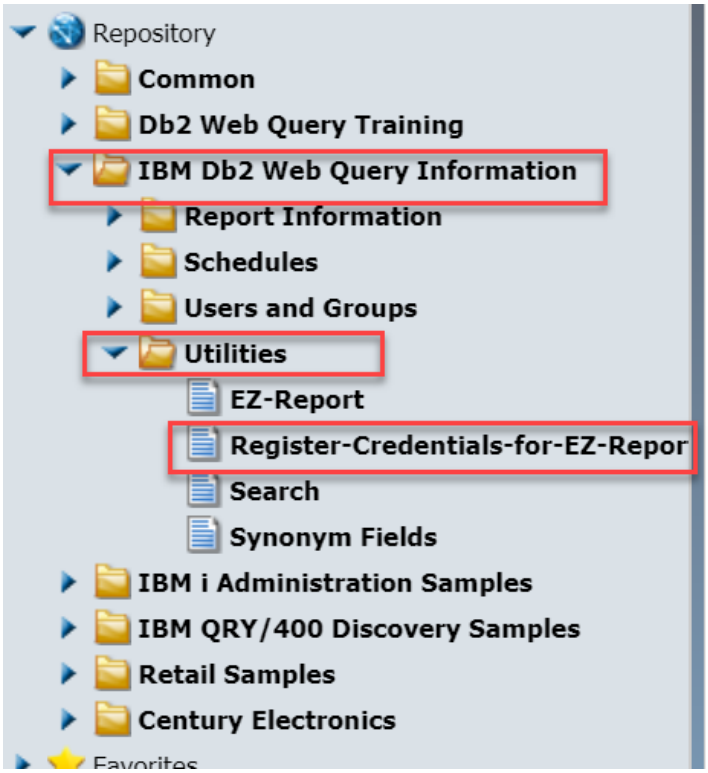



This is what we expect as it is resetting the running total at each product type change. Copy this SQL from ACS.

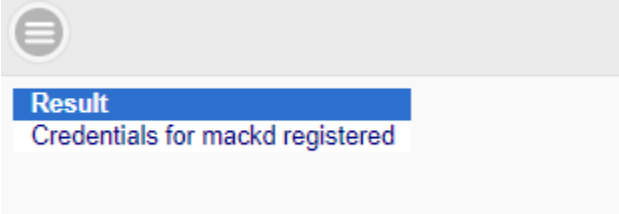
### 1.1.2 EZ-Report

Prior to the first time you run EZ-report you must register your credentials. This is a one-time activity.

If still signed into Db2 Web Query, navigate to the IBM I Db2 Web Query Information folder and open up the Utilities sub-folder. **Double click on Register-Credentials-for-EZ-Report.**



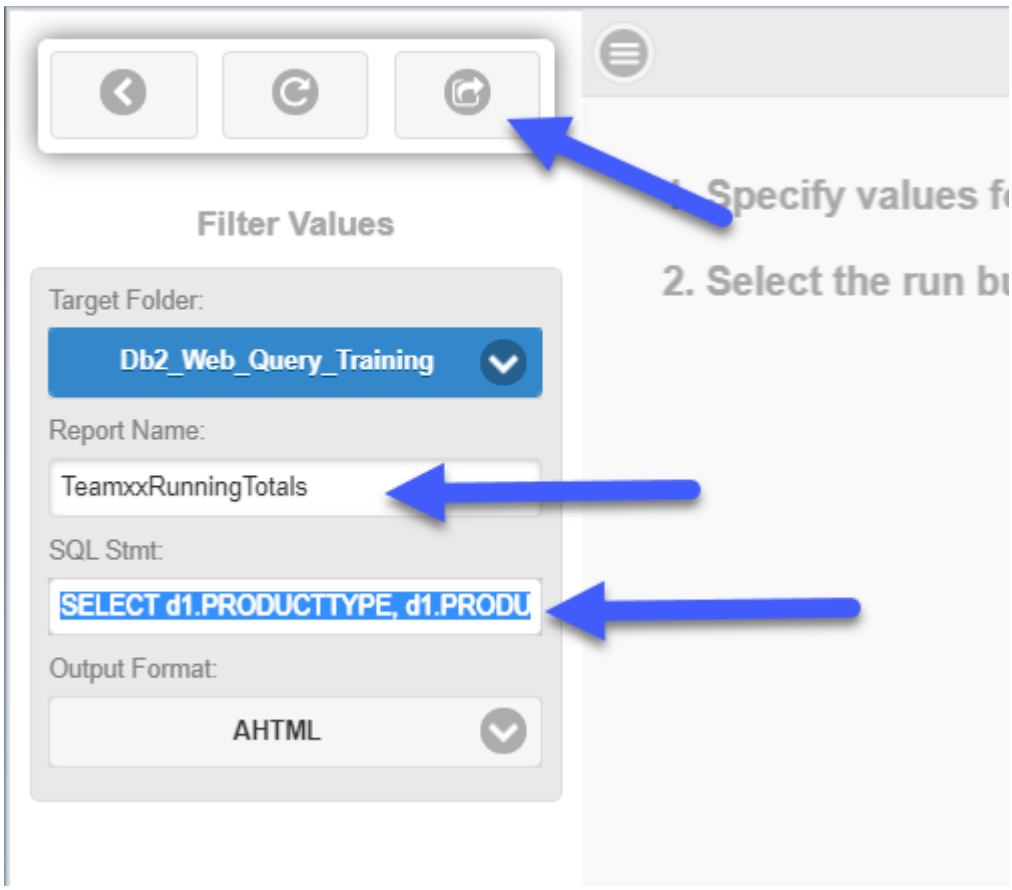
Enter your user ID and password that you used to sign into Db2 Web Query. **Click on the run button.**

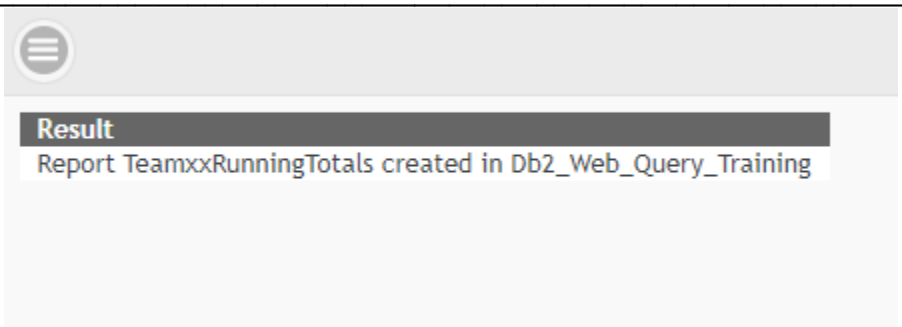



Navigate to the IBM i Db2 Web Query Information folder and open up the Utilities sub-folder if not already there. **Double click on EZ-Report**.

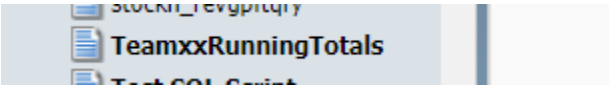
On the Target Folder drop down, **select the Db2 Web Query Training** folder. At this point, the drop down only allows you to select "top-level" folders, not subfolders. **Enter report name as TeamxxRunningTotal.** NOTE: EZ-Report does not allow any special characters or spaces in the title of the report.

In the SQL Stmt entry field, **paste your SQL statement** from ACS. Leave AHTML (HTML Analytical document) as the default output format. **Click on the RUN button**.






Assuming you got the "created" message**, right click** on the Db2 Web Query Training folder and **select REFRESH**. You should see your new report:




**Double click on your new report**. Note that we're almost there, but not quite done yet. We still want to add a subtotal of revenue by product type.

15 of 15 records, Page 1 of 1

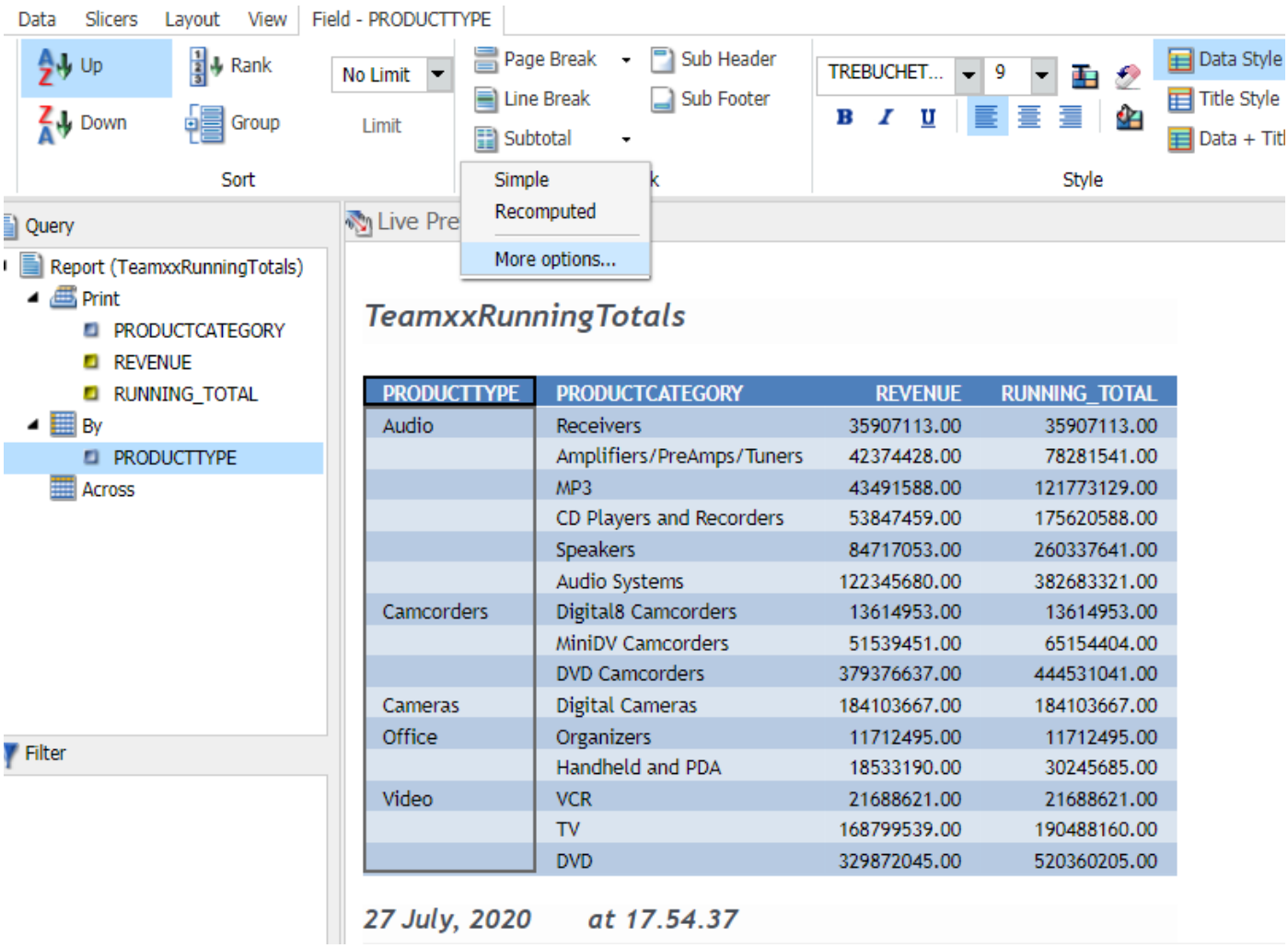
*TeamxxRunningTotals*

| PRODUCTTYPE | PRODUCTCATEGORY | REVENUE | RUNNING_TOTAL |
|---|---|---|---|
| Audio | Receivers | 35907113.00 | 35907113.00 |
| | Amplifiers/PreAmps/Tuners | 42374428.00 | 78281541.00 |
| | MP3 | 43491588.00 | 121773129.00 |
| | CD Players and Recorders | 53847459.00 | 175620588.00 |
| | Speakers | 84717053.00 | 260337641.00 |
| | Audio Systems | 122345680.00 | 382683321.00 |
| Camcorders | Digital8 Camcorders | 13614953.00 | 13614953.00 |
| | MiniDV Camcorders | 51539451.00 | 65154404.00 |
| | DVD Camcorders | 379376637.00 | 444531041.00 |
| Cameras | Digital Cameras | 184103667.00 | 184103667.00 |
| Office | Organizers | 11712495.00 | 11712495.00 |
| | Handheld and PDA | 18533190.00 | 30245685.00 |
| Video | VCR | 21688621.00 | 21688621.00 |
| | TV | 168799539.00 | 190488160.00 |
| | DVD | 329872045.00 | 520360205.00 |

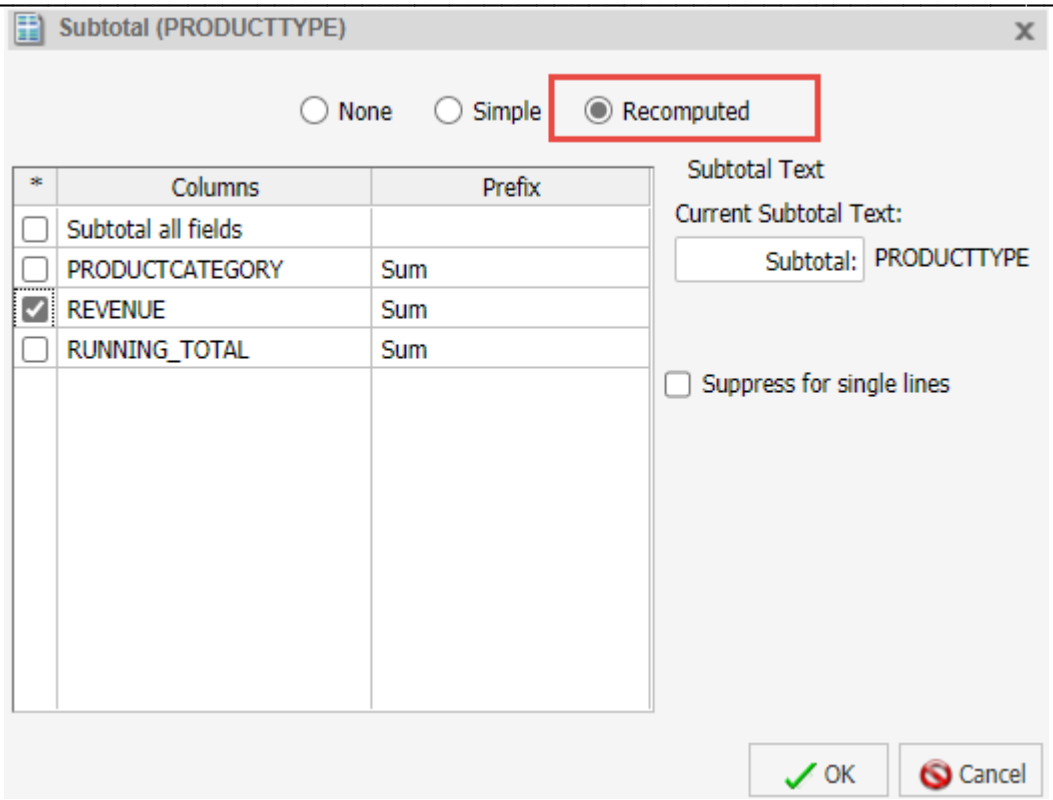
*27 July, 2020 at 17.53.22*

Right click on the TeamxxRunningTotals report and **choose EDIT**. Your report opens up in InfoAssist.

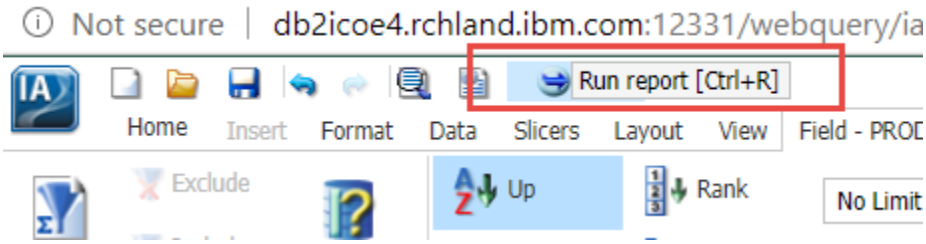
**Select the Product Type** column in your live preview menu to get the field/column tools in the ribbon bar. **Click on the down arrow** next to the Subtotals icon and **select More Options**:




On the subtotal pop up, click on the recomputed radio button. Also, DE-select subtotal all fields and instead check just the Revenue column.




**Click OK**. Now **click on the RUN** button and you should see our desired end goal!



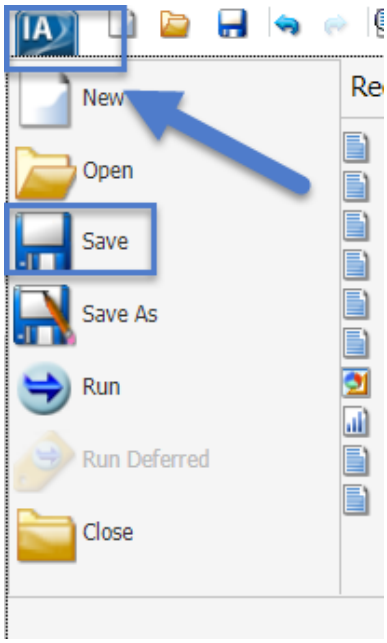
TeamxxRunningTotals[0]

15 of 15 records, Page 1 of 1

*TeamxxRunningTotals*

| PRODUCTTYPE | PRODUCTCATEGORY | REVENUE | RUNNING_TOTAL |
|---|---|---|---|
| Audio | Receivers | 35907113.00 | 35907113.00 |
| | Amplifiers/PreAmps/Tuners | 42374428.00 | 78281541.00 |
| | MP3 | 43491588.00 | 121773129.00 |
| | CD Players and Recorders | 53847459.00 | 175620588.00 |
| | Speakers | 84717053.00 | 260337641.00 |
| | Audio Systems | 122345680.00 | 382683321.00 |
| **Subtotal: Audio** | | **382683321.00** | |
| Camcorders | Digital8 Camcorders | 13614953.00 | 13614953.00 |
| | MiniDV Camcorders | 51539451.00 | 65154404.00 |
| | DVD Camcorders | 379376637.00 | 444531041.00 |
| **Subtotal: Camcorders** | | **444531041.00** | |
| Cameras | Digital Cameras | 184103667.00 | 184103667.00 |
| **Subtotal: Cameras** | | **184103667.00** | |
| Office | Organizers | 11712495.00 | 11712495.00 |
| | Handheld and PDA | 18533190.00 | 30245685.00 |
| **Subtotal: Office** | | **30245685.00** | |
| Video | VCR | 21688621.00 | 21688621.00 |
| | TV | 168799539.00 | 190488160.00 |
| | DVD | 329872045.00 | 520360205.00 |
| **Subtotal: Video** | | **520360205.00** | |

Save your report by **clicking on the large IA icon** and selecting SAVE:



There are many more ways to now use this report as is or edit it, or even edit the synonym. You'll learn more about InfoAssist in the next tutorial. But after your synonym and report are generated, you could:

- Change the color scheme
- Alter the header and footer
- Create a drill down to a DETAIL level report
- Add a prompt for Product Type
- Add a prompt to choose output type at Run Time
- Create a new CHART over this data

Schedule the report to run and send out via a distribution list or push it to the IFS or network drive.

Place in "FAVORITES" folder to be synchronized with your tablet device.

Optionally, alter the synonym that was auto generated:

- Join this data set to another
- Format the fields (add $ or Euro symbol to monetary fields)

By using the SQL statement here (using OLAP extensions), we are leveraging SQL to do a more complex reporting task, but also, we are creating a higher performing report by pushing the work to Db2 and letting it do the hard stuff! A WIN WIN for sure! OLAP extensions to the SQL language can be very powerful and useful tool for any business intelligence endeavor. To learn more about this, you can click here.

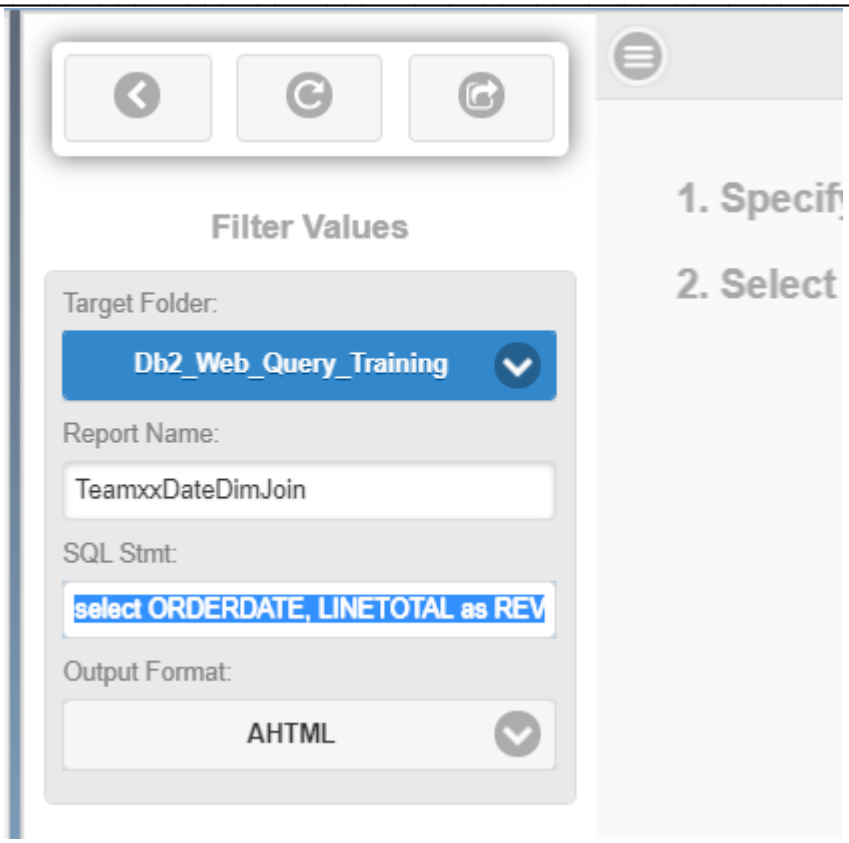
It should be pointed out that you can achieve this same report without requiring SQL. But the SQL combined with EZ-Report makes it…..EASIER!

### 1.1.3 Joining to Date Dimension Table

If you're not on IBM i 7.3 or above, then here's something you can try with EZ-Report. Db2 Web Query comes with a sample date dimension table, as well as a utility to create your own. The date dimension table is a great way to join your files with legacy dates (liked Packed 8 decimal) to it in order to surface any number of date attributes to the report writer. The above section on creating your first synonym showed you how you do that within the Synonym. But if I want a quick report choosing some date attributes then here is an example of just using a quick SQL statement to join the ORDERS table to the DATE_CONV table and feeding it into EZ-Report.

Using the following SQL statement, either copy/paste this into ACS (recommended to test it) or just jump directly to EZ-Report using the above section as your guide – and see how you do!

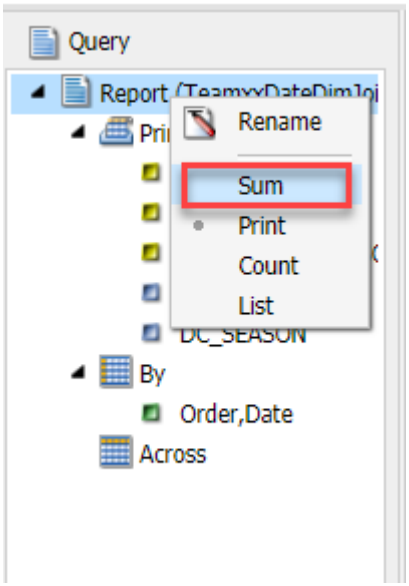
```
select ORDERDATE, LINETOTAL as REVENUE, DC_DOW as DAY_OF_WEEK, DC_DOW_ISO as
DAY_OF_WEEK_ISO, DC_DAY_NAME as DAY_NAME,DC_SEASON as SEASON
from qwqcent.orders INNER JOIN qwqcent.date_conv ON ORDERDATE = DC_DATE
WHERE DC_YEAR = '2019';
```




Be sure to specify Db2 Web Query Training and **name your report TeamxxDateDimJoin** when you run EZ-Report.

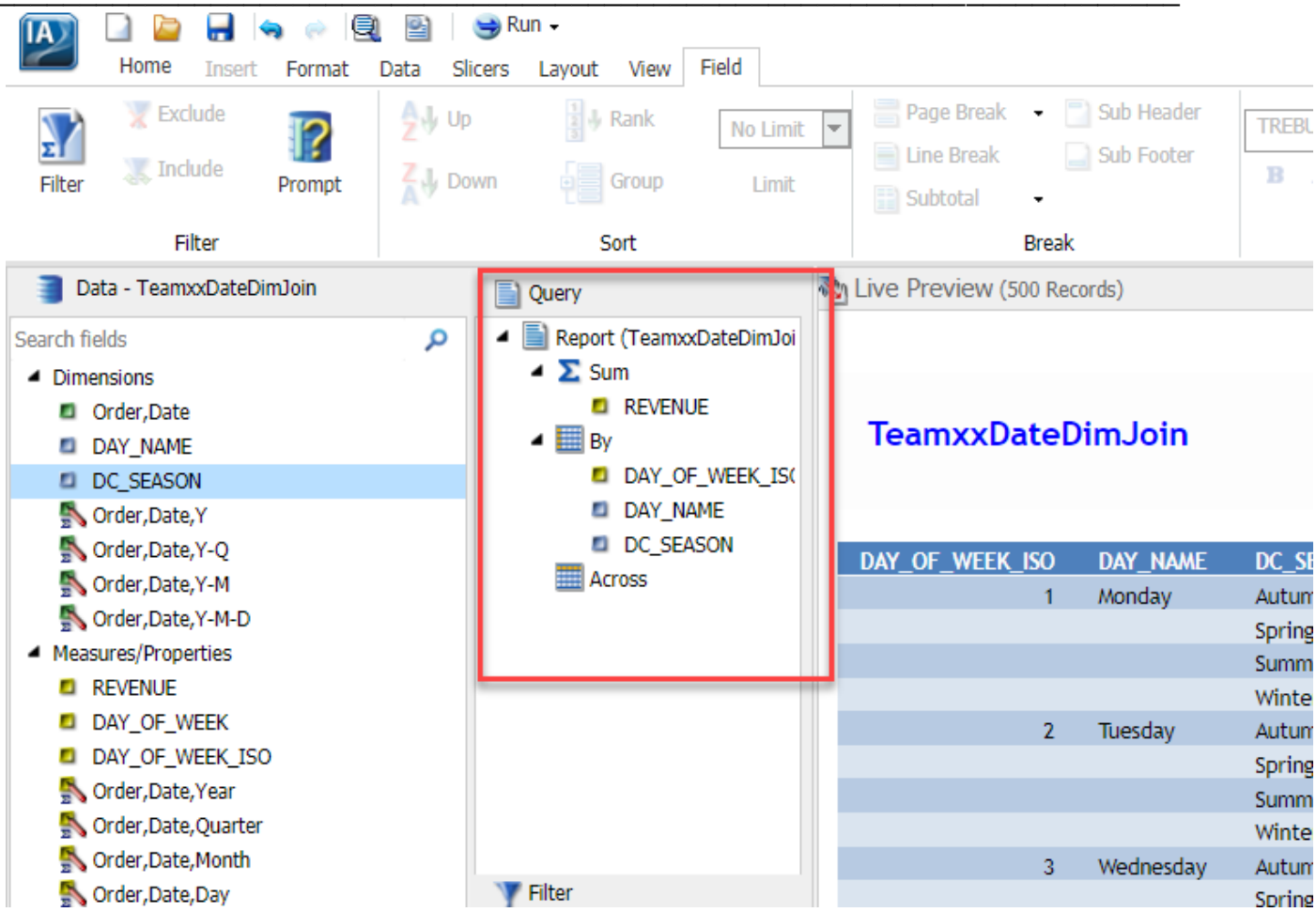
Once the report is generated remember to refresh the Db2 Web Query training folder, and double click on the new report. This is a really big report because it is detail level. Perhaps it would be better to edit this report and make it a summary report with filters. Open it up in InfoAssist.

Change the report type to SUMMARY by **right clicking** the report name in the query panel and **selecting SUM**.




Now that we have the data for every day in 2019 there was an order, what we really want to see our revenue by day of the week and season, with subtotals for each day.

**Delete and or move the fields around** to look like this in the query panel:

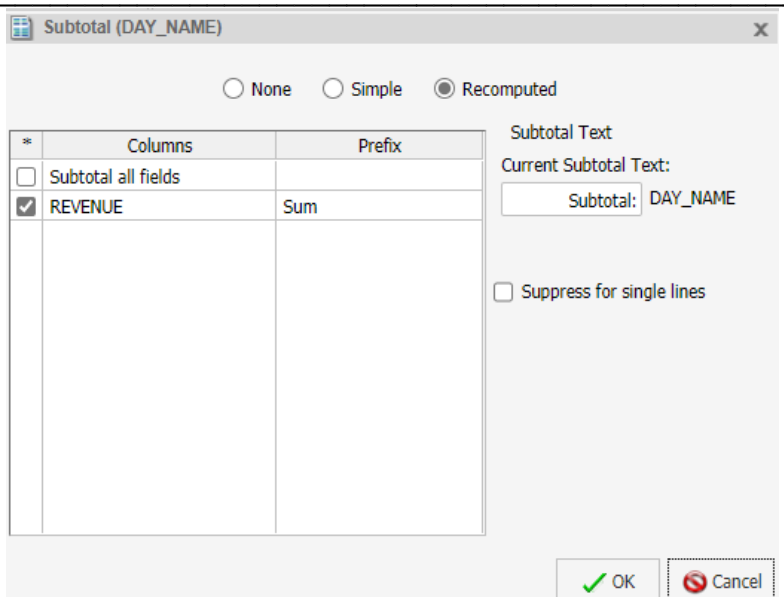



It would be nice to have the days to appear in logical order, which is why the DAY_OF_WEEK_ISO field is placed as the first field under the BY section, denoting it as the highest order SORT field. However, we don't want that to show up on our report. **Right click on DAY_OF_WEEK_ISO** and change the **visibility to HIDE** it.

The ISO week starts on Monday. If you wanted the day to start on Sunday, you would just use the DAY_OF_WEEK field instead of the ISO date!

One more thing – add a subtotal for each day. Either **right click on the DAY_NAME field in the query panel and select BREAK -> Subtotal -> more options - or click on the field in the Live Preview window and click the down arrow** on the subtotal icon then more options.

Choose the subtotal the revenue and select the recomputed radio button. Run your report:




Live Preview (500 Records)

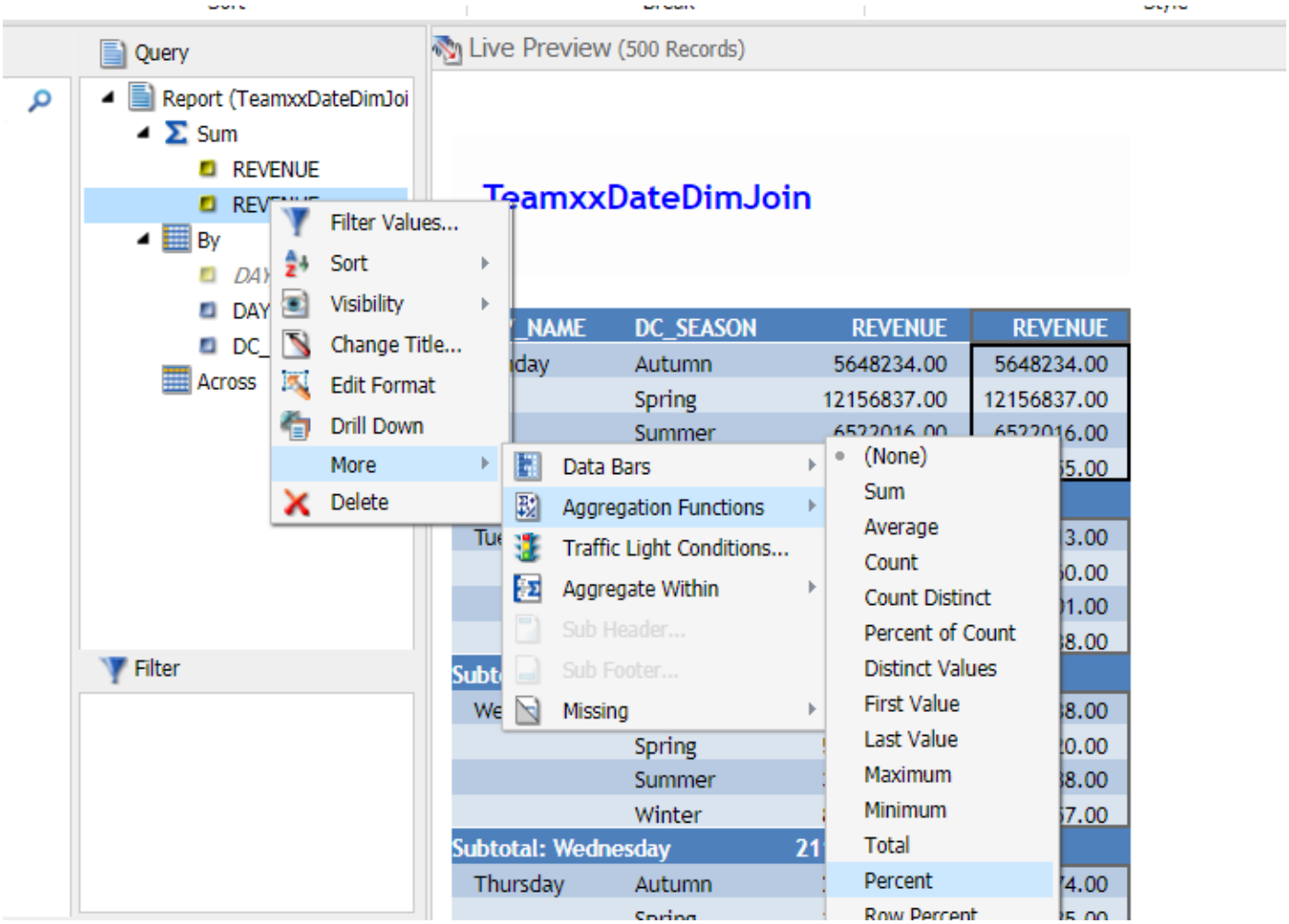
TeamxxDateDimJoin

| DAY_NAME | DC_SEASON | REVENUE |
|---|---|---|
| Monday | Autumn | 5648234.00 |
| | Spring | 12156837.00 |
| | Summer | 6522016.00 |
| | Winter | 12027855.00 |
| **Subtotal: Monday** | | **36354942.00** |
| Tuesday | Autumn | 3699313.00 |
| | Spring | 44604960.00 |
| | Summer | 23182701.00 |
| | Winter | 36406238.00 |
| **Subtotal: Tuesday** | | **107893212.00** |
| Wednesday | Autumn | 31728438.00 |
| | Spring | 56071420.00 |
| | Summer | 39806888.00 |
| | Winter | 83584357.00 |
| **Subtotal: Wednesday** | | **211191103.00** |
| Thursday | Autumn | 23215874.00 |

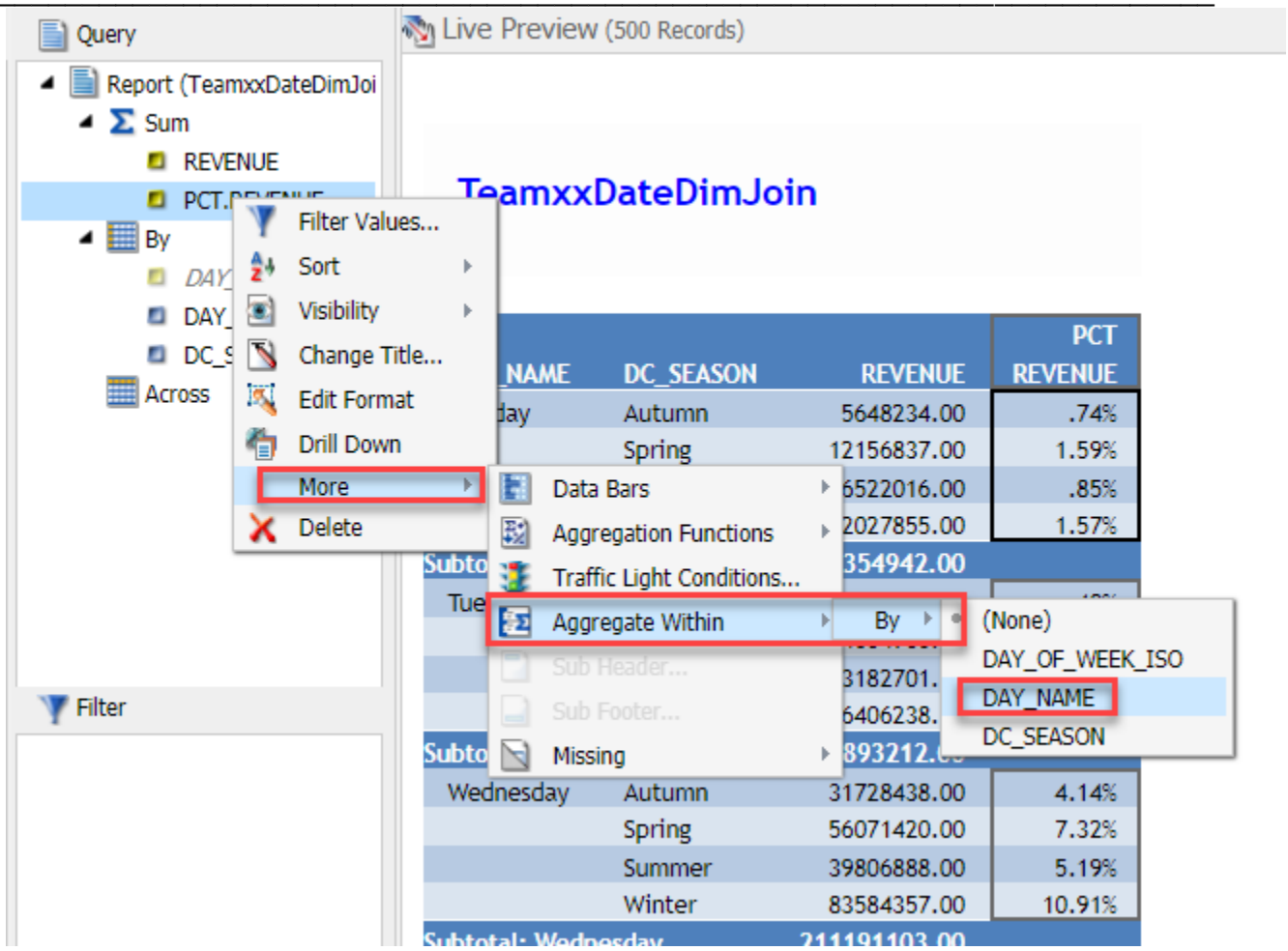
**Save your report.**

It might be fun to add an "as a percentage of" column, but similar to the running totals we want this to be as a percentage of within each day of the week. For this, we could also use 7.3 OLAP functions in our SQL statement, but this is fairly easy to do in the report we have.

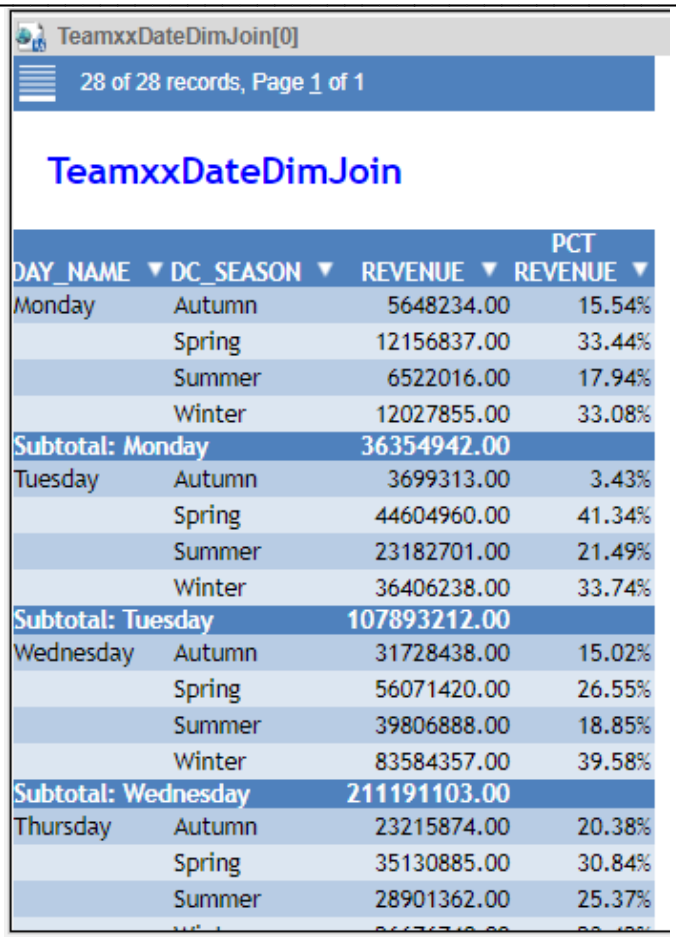
**Add revenue to the report a second time** in the query panel. **Right click on it and select more -> aggregation functions -> percent** as such:




Do this ONE MORE TIME to get the as a percentage of WITHIN your group of day name**. Right click again on the 2nd revenue field**, take **more -> AGGREGATE WITHIN -> and select Day_Name**.




Run the report and notice how our goal was achieved. We have a report that took some data joined with a date dimension table to get the date attributes we needed for this, and fed it into ez-report, then manipulated the generated report to look how we wanted it to, including the **as a percentage** of for each day name.



TeamxxDateDimJoin[0]

28 of 28 records, Page 1 of 1

TeamxxDateDimJoin

| DAY_NAME | DC_SEASON | REVENUE | PCT REVENUE |
|---|---|---|---|
| Monday | Autumn | 5648234.00 | 15.54% |
| | Spring | 12156837.00 | 33.44% |
| | Summer | 6522016.00 | 17.94% |
| | Winter | 12027855.00 | 33.08% |
| Subtotal: Monday | | 36354942.00 | |
| Tuesday | Autumn | 3699313.00 | 3.43% |
| | Spring | 44604960.00 | 41.34% |
| | Summer | 23182701.00 | 21.49% |
| | Winter | 36406238.00 | 33.74% |
| Subtotal: Tuesday | | 107893212.00 | |
| Wednesday | Autumn | 31728438.00 | 15.02% |
| | Spring | 56071420.00 | 26.55% |
| | Summer | 39806888.00 | 18.85% |
| | Winter | 83584357.00 | 39.58% |
| Subtotal: Wednesday | | 211191103.00 | |
| Thursday | Autumn | 23215874.00 | 20.38% |
| | Spring | 35130885.00 | 30.84% |
| | Summer | 28901362.00 | 25.37% |

To be clear, you do not need ez-report to build this report. Just like you built the synonym that joined to the date dimension (a.k.a., date conversion table) in the previous section, you could have done that and just built your report from there. In this case, it is simply an example of how you also have the alternative if you are comfortable with SQL and prefer that. The previous OLAP example is actually taking advantage of SQL to make a challenging reporting issue simpler! This example with the date dimension join, not so much.

Is it better to build a synonym such as in Chapter 5? Probably – for the long-term benefit of having a synonym with pre-defined joins, virtual fields, and other aspects that report authors can use over and over again. Think of EZ-report as a very quick way to get something done in a case where you specifically want to use SQL statements as the input.

# 2 Additional Information: Links

Links:
Ibm.biz/db2webqueryi - main website
Ibm.co/db2wqwiki - technical wiki
Db2webqueryi.blogspot.com – Doug Mack's blog
Ibm.biz/db2wqezinstall – EZ-install getting started page
Contact the Db2 Web Query Team at QU2@us.ibm.com