

**FIS/HRIS Data Warehouse Cooperative Learning Group**  
**03-05-03**  
**Random Tips**

2. Select data from one object (we'll select LFM Name from Person). Then select data from a second object that isn't already connected to the first object by a single line (we'll select Job Title from Jobs). Close the attribute list from the second object and look at the relationship lines between the two objects. If they aren't highlighted in blue, you'll get the "can't join data" error message when you submit your query. To avoid the message, click once on each line between the two objects to highlight them blue, then submit the query.

It used to be that if you merely opened an object to look at the attributes in it, the software would show that object as selected (the name of the object is in Bold and there is a bullet to the left of it). Even though no attributes were selected from that object, the only way to submit the query without clearing it out and starting over was to select the relationship line to connect that extra object. With the latest release of the data warehouse models, that problem should have gone away.

3. By clicking in the Function column across from an attribute, you can apply an Aggregate Function to the attribute. These group any selected attributes so that you don't see duplicated data and display the results of the function. When you use these, you usually don't want to try to get much detail. The function applies a mathematic operation on a group of records. The results will change based on which additional attributes you have selected. They don't operate on NULL values, which are blank fields.

**AVERAGE** (averages the value of all non-null, that is, not blank, values)  
In Jobs, qualify the EClass attribute = CA. Select the Regular Hourly Rate attribute and choose Average from the function column across from it.

**MAXIMUM**  
Change the function for the Regular Hourly Rate to Maximum.

**MINIMUM**  
Change the function for the Regular Hourly Rate to Minimum.

**SUM**  
Change the function for the Regular Hourly Rate to Sum.

**COUNT**  
Start a new query. In Person, qualify Middle Name = Ellen (or try your own middle name!), and submit the query to see how many records come back. Go back to the query and select Count from the function column across from Middle Name and submit the query again, and see that you get the same Count. Now go back to the query and select None from the function column for Middle Name, and select Count for PIDM and submit the query. See that you again get the same results. Now also select the Last Name. See

how it counts the number of occurrences of that middle name for each last name. The more attributes you select, the less the function aggregates the data.

Start a new query. In Person, qualify Middle name IS NULL. Now submit the query and see how many records come back. Now Count on Middle Name and submit the query. See that no records are found? That's because the function doesn't work on NULL values. Change Middle Name to a function of None, and Count instead on PIDM, which must be populated on every record. See that it does come up with the correct count.

DISTINCT means the operation eliminates duplicates, and then uses each remaining unique value once.

4. If you are tired of always creating the exact same calculated column in a report, or if you wish you could qualify on that calculated column, you might want to create a calculated attribute. Once you create one, it can be hard to get it removed from your model, so be very sure before you save it.

Click in the Function column across from the attribute above where you want the new attribute to appear, and choose Calculation from the list. Give your attribute a name, review the data type, choosing Number or Character as appropriate, and uncheck "Save Definition". The calculation is entered in the Expression box, for example: Jul Dr + Aug Dr + Sep Dr, or Area Code || Phone Number. The two vertical lines tell the software to run the two attributes together with no space between them, like 5417370616.

So that you don't have to recreate your calculated attribute all the time, without making it a permanent part of your model, use it in a query and then save the query. Each time you use the query subsequently, the calculated attribute will be available.