# DataMgr 2.2 Has Math Now

Posted At : February 20, 2008 7:30 AM | Posted By : Steve
Related Categories: DataMgr

The second Alpha of **DataMgr** 2.2 is now available. In addition to the new features from **DataMgr 2.2 Alpha 1**, I have added three relation fields and one new method and removed a significant restriction for join tables. These changes should make it even easier to use DataMgr to develop your applications and keep your applications database agnostic.

The new method (covered momentarily) is getNowSQL(). The new relation fields are "has", "math", and "now" (hence the title! ;).

### saveRelationList() and "list" Relation Type

Thanks to a suggestion from Sal Valverde a join table for a many-to-many relationship (the table indicated in the saveRelationList method or the "join-table" for a list relation type) is no longer required to a have a compound primary key. Formerly, a join table must have had a primary key made up of two fields (holding the primary key values from each joining table). Now it can have a separate primary key (though it must, of course, still have a field for the primary key of each table that it joins). Thanks Sal!

### getNowSQL()

The new method is easy to use. It simply returns a string representing the SQL that indicates the current date and time. For example, it would return "getDate()" for MS SQL and "Now()" for MS Access.

### "now" Relation Type

The now relation type is a simple one. It simply creates a column that always return the current date and time for every column. I'll confess that I am not sure how I would use this yet, but it was an easy addition.

XML:

```
<field ColumnName="MyField">
<relation type="now" />
</field>
```

setColumn():

```
<cfset sRelation = StructNew()>
<cfset sRelation.type = "now">
<cfset Application.DataMgr.setColumn(tablename="MyTable",ColumnName="MyField",Relation=sRelation)>
```

### "math" Relation Type

The math relation allows you to perform basic mathematical operations between two fields from within DataMgr. The operators allows are: +-*/, for addition, subtraction, multiplication, and division respectively.

Among the uses for this, one that I have considered is the ability to calculate an order total in a cart. Each line item could have a total cost that is a multiplication of the quantity by price. The order itself could then add these values for the pre-tax total and take that number and multiply it by the tax rate (pulled from another relation field). All of this could now be done from within DataMgr without writing any SQL.

XML:

```
<field ColumnName="MyField">
<relation type="math" field1="OrderTotal" field2="TaxRate" operator="*" />
</field>
```

setColumn():

```
<cfset sRelation = StructNew()>
<cfset sRelation.type = "math">
<cfset sRelation.field1= "OrderTotal">
<cfset sRelation.field2= "TaxRate">
<cfset sRelation.operator= "*">
<cfset Application.DataMgr.setColumn(tablename="MyTable",ColumnName="MyField",Relation=sRelation)>
```

### "has" Relation Type

I have noticed recently that I often need to know if a record has one thing or another. Hence the "has" relation type. It returns a boolean value indicating the existence of a positive value in the indicated field.

This breaks down by field type as follows:

- string: True if the field has length. False for NULL or empty string.
- date: True for any non-NULL value.
- boolean: True for true. False for false and NULL.
- numeric: True for any number over zero.

This allows an indication, for example, of all users who do (or do not) have an email address.

XML:

```
<field ColumnName="MyField">
<relation type="has" field="Email" />
</field>
```

setColumn():

```
<cfset sRelation = StructNew()>
<cfset sRelation.type = "has">
<cfset sRelation.field= "Email">
<cfset Application.DataMgr.setColumn(tablename="MyTable",ColumnName="MyField",Relation=sRelation)>
```

All of these relation types can point use other relation fields and can be used by other relation fields.

In the meantime, feel free to **download DataMgr 2.2 Alpha** from the repository. If you have a database I can test against, let me know.

**DataMgr** is open source and free for any use.