

Custom Relations Part II: Complex Logic

Posted At : December 26, 2007 11:00 AM | Posted By : Steve

Related Categories: DataMgr

A friend of mine called me yesterday with an interesting problem because he was unable to get some data for a record using DataMgr because it would require complicated logic. He didn't want to drop DataMgr because then he would have to write this logic in several places.

The essence of my friend's problem is as follows (though the actual fact pattern differs). He has a "Members" table, a "PhoneNumbers" tables, and a "PhoneTypes" table. Other logic ensures that each member can only have one phone number listed for any given type of phone number.

His problem is that he needs to get the home phone number for a record. In most situations, DataMgr's **relation fields** would provide the answer. For example, he could use a relation field to get a list of phone numbers, but that isn't what he needs.

So, we first looked at what SQL we would write to get this information about one member:

```
SELECT Number
FROM PhoneNumbers
INNER JOIN PhoneTypes
ON PhoneNumbers.PhoneTypeID = PhoneTypes.PhoneTypeID
WHERE MemberID = 1
```

Now, we could use this code in a subquery if we replace the static "1" with a reference to the id of the Members table:

```
SELECT Number
FROM PhoneNumbers
INNER JOIN PhoneTypes
ON PhoneNumbers.PhoneTypeID = PhoneTypes.PhoneTypeID
WHERE MemberID = Members.MemberID
```

From here, it is a simple matter of using DataMgr's "custom" relation type. To do so using the XML definition:

```
<tables>
<table name="Members">
<field ColumnName="MemberID" CF_DataType="CF_SQL_INTEGER" PrimaryKey="true" Increment="true"/>
...
<field ColumnName="HomePhone">
<relation
type="custom"
sql="
SELECT Number
FROM PhoneNumbers
INNER JOIN PhoneTypes
ON PhoneNumbers.PhoneTypeID = PhoneTypes.PhoneTypeID
WHERE MemberID = Members.MemberID
"
/>
</field>
</table>
</tables>
```

To do so using setColumn():

```

<cfsavecontent variable="HomePhoneSQL">
SELECT Number
FROM PhoneNumbers
INNER JOIN PhoneTypes
ON PhoneNumbers.PhoneTypeID = PhoneTypes.PhoneTypeID
WHERE MemberID = Members.MemberID
</cfsavecontent>
<cfset sRelation = StructNew()>
<cfset sRelation["type"] = "custom">
<cfset sRelation["sql"] = HomePhoneSQL>
<cfset Application.DataMgr.setColumn(tablename="Members",ColumnName="HomePhone",Relation=sRelation)>

```

So now the recordset returned by calling `Application.DataMgr.getRecords("Members")` includes a column named "HomePhone" with the home phone number of the member.

The only problem with this solution is that the problem he has is actually a bit more complicated. In his situation, he needs to get the phone number for a member from an attendees table that indicates which members are attending certain classes. But that is a story for the next installation.

In the meantime, [DataMgr](#) is open source and free for any use.