Custom Relations Part I: IsNull

Posted At : December 20, 2007 7:00 AM | Posted By : Steve
Related Categories: DataMgr

I just got an email from someone using DataMgr asking how to use DataMgr to filter by whether or not a field is NULL. This seems like functionality that DataMgr should have (and hopefully will in a future version), but right now it doesn't. It doesn't seem right that this should be the end of the road. Fortunately, you can convince DataMgr to do things it wasn't built to do.

The answer lies in DataMgr's **relation fields** (if you haven't used them, take a read). It seems like they wouldn't be of any use here, as none of the existing types of relation fields has anything to do with whether or not a field is NULL. Fortunately, the "custom" type can be used for any purpose.

In this case, we need to use the isnull() function of the database to determine if a field is null and have a field with that value. This can be done in the XML definition:

```
<field ColumnName="IsMyFieldNull">
<relation type="custom" sql="abs(IsNull(MyField))" CF_DataType="CF_SQL_BOOLEAN" />
</field>
```

Or it can be done via setColumn():

```
<cfset sRelation = StructNew()>
<cfset sRelation["type"] = "custom">
<cfset sRelation["sql"] = "abs(IsNull(MyField))">
<cfset sRelation["CF_DataType"] = "CF_SQL_BOOLEAN">
<cfset Application.DataMgr.setColumn(tablename="MyTable",ColumnName="IsMyFieldNull",Relation=sRelation)>
```

Note that typically the "CF_Datatype" attribute would be part of the field definition, but this time it is part of the relation definition. This is necessary so that DataMgr doesn't try to treat this field as a real field in the database.

Now DataMgr will return IsMyFieldNull as a field in the recordset resulting from:

```
<cfset qRecords = Application.DataMgr.getRecords("MyTable")>
```

Note that the above would work even without the CF_Datatype definition in the relation. The CF_Datatype definition is required, however, if you want to filter by the field.

To filter by the field, simply treat it as any other field. So, if I wanted to get only records where "MyField" is null:

```
<cfset sData = StructNew()>
<cfset sData.IsMyFieldNull = 1>
<cfset qRecords = Application.DataMgr.getRecords("MyTable",sData)>
```

If I wanted a field that indicated the opposite value (that is, true when a field is not null), I could set the sql attribute to "abs(NOT IsNull(MyField))" (replacing "MyField" with the name of the field being tested, of course).

That's it!

Next time, how to get data that requires complex SQL from multiple tables.

DataMgr is open source and free for any use.