

DataMgr "All" for Many-to-Many Relationships

Posted At : July 11, 2008 7:00 AM | Posted By : Steve

Related Categories: DataMgr

In my [last post](#), I explained how I use an "all" field in combination with a many-to-many relationship to indicate that one record is related to any/all records in the related table (even if new records are added later). I have been using that technique for about three years now with very good results.

In fact, this worked so well for me that I added a feature into [DataMgr 2.2](#) to handle it.

(What is DataMgr?)

So, if I wanted to use the previous example in DataMgr using loadXml(), I would define a "permissions" column:

```
<field ColumnName="permissions">
  <relation
    type="list"
    table="articles2permissions"
    field="permission_id"
    join-field="article_id"
  />
</field>
```

The code for doing the same using setColumn() would be:

```
<cfscript>
sRelation = StructNew();
sRelation["type"] = "list";
sRelation["table"] = "articles2permissions";
sRelation["field"] = "permission_id";
sRelation["join-field"] = "article_id";
DataMgr.setColumn(tablename="articles",ColumnName="permissions",Relation=sRelation);
</cfscript>
```

Now I could pass in a "permissions" key to get all of the articles with any of the given permissions:

```
<cfset sData = {permissions="1,2"}>
<cfset qArticles = DataMgr.getRecords(tablename="articles",data=sData)>
```

If I want to make sure that any articles with allPermissions of 1 get included as well (regardless of what permissions are passed in), I would make the following changes:

loadXml():

```
<field ColumnName="permissions">
  <relation
    type="list"
    table="articles2permissions"
    field="permission_id"
    join-field="article_id"
    all-field="allPermissions"
  />
</field>
```

setColumn():

```

<cfscript>
sRelation = StructNew();
sRelation["type"] = "list";
sRelation["table"] = "articles2permissions";
sRelation["field"] = "permission_id";
sRelation["join-field"] = "article_id";
sRelation["all-field"] = "allPermissions";
DataMgr.setColumn(tablename="articles",ColumnName="permissions",Relation=sRelation);
</cfscript>

```

The code to get the records remains the same:

```

<cfset sData = {permissions="1,2"}>
<cfset qArticles = DataMgr.getRecords(tablename="articles",data=sData)>

```

In this example the "allPermissions" field is an actual field in the "articles" table. It could also have been another relation field (like "permissions") in the "articles" table. That is probably best left for another entry though.

If you find the "all" approach useful and you use DataMgr, it is one more thing you can do without having to write any SQL.

DataMgr is open source and free for any use. Download 2.2 Alpha from the [DataMgr page](#) on my site.